Veronte Autopilot (Up to HW 4.5 / SW 6.4)

Release 5.42.28

Embention

2023-07-26

CONTENTS

1 Quick Start **1**
- 1.1 First Steps . . . 1
  - 1.1.1 Kit Elements . . . 2
  - 1.1.2 Veronte Pipe . . . 2
- 1.2 Model Simulation . . . 5
  - 1.2.1 Loading a Template to the Autopilot . . . 5
  - 1.2.2 X-Plane and Veronte Pipe Setup . . . 7
    - 1.2.2.1 Veronte Pipe . . . 7
    - 1.2.2.2 X-Plane . . . 7
  - 1.2.3 Executing a Simulation . . . 8
- 1.3 Autopilot Radio Pairing . . . 15
- 1.4 Calibrating Servos . . . 16
  - 1.4.1 Servos Output . . . 16
  - 1.4.2 SU Matrix . . . 18
  - 1.4.3 System Trim . . . 20
- 1.5 External Transmitter Configuration . . . 23
  - 1.5.1 PPM . . . 25
  - 1.5.2 Output . . . 26
- 1.6 Pre-flight Checklist . . . 27
  - 1.6.1 Main Checklist . . . 27
  - 1.6.2 Other Items . . . 33

2 Hardware Installation **35**
- 2.1 Veronte Autopilot . . . 35
  - 2.1.1 Aircraft Mounting . . . 35
    - 2.1.1.1 Enclosure . . . 35
    - 2.1.1.2 OEM . . . 36
    - 2.1.1.3 Mechanical Mounting . . . 37
    - 2.1.1.4 Vibration Isolation . . . 37
    - 2.1.1.5 Location . . . 38
    - 2.1.1.6 Orientation . . . 38
    - 2.1.1.7 Connector Layout . . . 38
    - 2.1.1.8 Mating Connector . . . 39
    - 2.1.1.9 Antenna Integration . . . 39
    - 2.1.1.10 Pressure lines . . . 40
  - 2.1.2 Electrical . . . 40
    - 2.1.2.1 Power . . . 40
    - 2.1.2.2 Veronte I/O Signals . . . 41
    - 2.1.2.3 Flight Termination System (FTS) . . . 43
    - 2.1.2.4 Joystick . . . 43

2.1.3 Performances . . . 45
2.1.4 Annex 1: Connector colour code . . . 46
2.1.5 Annex 2: Connection examples . . . 48
2.1.5.1 Ground Station . . . 48
2.1.5.2 Multicopter . . . 49
2.1.5.3 Fixed Wing . . . 50
2.1.5.4 Helicopter . . . 51
2.1.6 Operation . . . 52
2.1.7 Platforms . . . 53
2.1.8 Safety . . . 53
2.2 4x Veronte Autopilot . . . 54
2.2.1 Aircraft Mounting . . . 54
2.2.1.1 Enclosure . . . 54
2.2.1.2 Vibration Isolation . . . 54
2.2.1.3 Location . . . 55
2.2.1.4 Orientation . . . 55
2.2.1.5 Connector Layout . . . 56
2.2.1.6 Mating Connectors . . . 57
2.2.1.7 Antenna Integration . . . 58
2.2.1.8 Pressure lines . . . 58
2.2.2 Electrical . . . 59
2.2.2.1 Power . . . 59
2.2.2.2 Redundant Connector Pinout . . . 59
2.2.2.3 Arbiter Connector Pinout . . . 65
2.2.3 Performances . . . 72
2.2.4 Annex 1: Connection Examples . . . 73
2.2.4.1 Ground Station . . . 73
2.2.5 Annex 2: Connector colour code . . . 73

3 Pipe Configuration **79**
3.1 Installation . . . 79
3.1.1 System Requeriments . . . 79
3.1.2 Windows . . . 79
3.1.3 MAC . . . 83
3.1.4 Linux . . . 84
3.2 Update . . . 88
3.3 Preferences . . . 91
3.3.1 General . . . 91
3.3.2 Connections . . . 95
3.3.3 Encryption . . . 100
3.3.3.1 Remove Encryption . . . 103
3.3.4 System Units . . . 103
3.3.4.1 Creating Custom Units . . . 104
3.3.4.2 Units . . . 105
3.3.5 Map . . . 106
3.3.5.1 Terrain Height . . . 107
3.3.5.2 Download ALtitude Information . . . 107
3.3.5.3 Custom Terrain File . . . 108

4 Workspace **111**
4.1 Workspace Toolbar . . . 111
4.1.1 Open Details . . . 112
4.1.2 Load . . . 113
4.1.3 Import/Export Workspaces . . . 115

4.2 Map Display . . . 115
4.2.1 Map provider . . . 115
4.2.2 Custom image . . . 115
4.3 Widgets . . . 116
4.3.1 Gimbal . . . 116
4.3.2 Gauge Display . . . 118
4.3.2.1 Gauge Selection . . . 118
4.3.2.2 Gauge Configuration . . . 120
4.3.2.3 Alarm Creation . . . 121
4.3.3 Advanced Primary Flight Display (PFD) . . . 123
4.3.4 Stick . . . 126
4.3.5 Terrain . . . 128
4.3.6 Autotune Tool . . . 129
4.3.7 Line of Sight . . . 130
4.3.8 ADS-B decoder . . . 131
4.3.9 GNSS status . . . 132
4.4 Main Interface . . . 134

5 Veronte Panel **135**
5.1 Sensors Calibration . . . 135
5.2 Veronte Icon . . . 138
5.3 Veronte Panel . . . 139

6 Veronte Configuration **141**
6.1 File Management . . . 141
6.1.1 Import Configurations . . . 141
6.1.1.1 CHECK PDI . . . 145
6.1.2 Export Configurations . . . 147
6.1.3 Offline Configurations . . . 148
6.1.4 Version Control . . . 149
6.1.5 Update Onboard Software . . . 149
6.1.6 Safe Mode . . . 150
6.1.7 CONFIGURATION FILES (.VER) . . . 151
6.1.8 PDI FILES . . . 152
6.2 Setup Toolbar . . . 152
6.2.1 Veronte . . . 152
6.2.2 Connections . . . 153
6.2.2.1 ADC . . . 153
6.2.2.1.1 Application example . . . 154
6.2.2.2 Arbiter - SuC . . . 154
6.2.2.3 CAN . . . 155
6.2.2.3.1 CAN Configuration . . . 155
6.2.2.3.2 CAN Telemetry . . . 157
6.2.2.3.2.1 TX Messages . . . 158
6.2.2.3.2.2 RX Messages . . . 160
6.2.2.3.3 CAN Messages . . . 160
6.2.2.3.3.1 Variable . . . 161
6.2.2.3.3.2 Checksum . . . 164
6.2.2.3.3.3 Matcher . . . 166
6.2.2.3.3.4 Skip . . . 167
6.2.2.3.3.5 Parse ASCII . . . 167
6.2.2.3.3.6 Position . . . 168
6.2.2.3.4 Arbiter CAN protocol . . . 168
6.2.2.3.4.1 CAN parameters . . . 168

6.2.2.3.4.2 CAN message structure . . . 169
6.2.2.3.4.3 Single message structure . . . 170
6.2.2.3.4.4 External autopilot simulation . . . 172
6.2.2.3.4.5 Can protocol for 4-autopilot configuration . . . 172
6.2.2.4 GPIO/PWM . . . 175
6.2.2.4.1 GPIO . . . 175
6.2.2.4.2 PWM . . . 176
6.2.2.4.3 Add . . . 177
6.2.2.5 Serial . . . 179
6.2.3 Devices . . . 182
6.2.3.1 Actuators . . . 182
6.2.3.1.1 Physical . . . 182
6.2.3.1.1.1 Calibration Interface . . . 182
6.2.3.1.1.2 Configuration Wizard . . . 183
6.2.3.1.2 Logical . . . 187
6.2.3.1.2.1 $\boldsymbol{SU}$ & $\boldsymbol{US}$ Matrices . . . 187
6.2.3.1.2.2 $\boldsymbol{SU}$ of a Flying Wing . . . 190
6.2.3.1.2.3 $\boldsymbol{SU}$ of a V-Tail Aircraft . . . 190
6.2.3.1.2.4 $\boldsymbol{SU}$ of a Quadcopter . . . 191
6.2.3.1.2.5 $\boldsymbol{SU}$ of a VTOL Aircraft . . . 192
6.2.3.1.3 Saturation . . . 192
6.2.3.1.3.1 Standard Saturation . . . 193
6.2.3.2 Communications . . . 198
6.2.3.2.1 4G . . . 199
6.2.3.2.1.1 ESIM . . . 199
6.2.3.2.1.2 SIM . . . 199
6.2.3.2.2 Comstats . . . 200
6.2.3.2.3 Iridium . . . 201
6.2.3.3 Payload . . . 202
6.2.3.3.1 Camera . . . 202
6.2.3.3.1.1 Veronte Gimbal Wizard . . . 204
6.2.3.3.2 Gimbal/Tracker . . . 207
6.2.3.3.3 Transponder . . . 208
6.2.3.3.4 ADS-B . . . 211
6.2.3.4 Sensors . . . 216
6.2.3.4.1 Accelerometer . . . 216
6.2.3.4.1.1 Integer var sensor . . . 217
6.2.3.4.1.2 Decimal var sensor . . . 218
6.2.3.4.1.3 Internal . . . 219
6.2.3.4.2 Altimeter . . . 221
6.2.3.4.3 GNSS . . . 222
6.2.3.4.3.1 GNSS 1 & 2 Configuration . . . 222
6.2.3.4.3.2 Configuration . . . 222
6.2.3.4.3.3 SBAS . . . 224
6.2.3.4.3.4 Message Rate . . . 225
6.2.3.4.3.5 Estimation Error . . . 226
6.2.3.4.3.6 Advanced . . . 227
6.2.3.4.3.7 GNSS Compass . . . 228
6.2.3.4.3.8 GNSS External . . . 231
6.2.3.4.3.9 Configuration . . . 232
6.2.3.4.3.10 Estimation Error . . . 233
6.2.3.4.3.11 NTRIP . . . 233
6.2.3.4.3.12 Registration . . . 233
6.2.3.4.3.13 Configuration . . . 233

6.2.3.4.4 Gyroscope . . . 236
6.2.3.4.4.1 Integer var sensor . . . 237
6.2.3.4.4.2 Decimal var sensor . . . 238
6.2.3.4.4.3 Internal . . . 239
6.2.3.4.5 I2C Devices . . . 241
6.2.3.4.6 Magnetometer . . . 242
6.2.3.4.6.1 Calibration . . . 242
6.2.3.4.6.2 Configuration . . . 243
6.2.3.4.6.3 Integer var sensor . . . 244
6.2.3.4.6.4 Decimal var sensor . . . 245
6.2.3.4.6.5 Internal . . . 246
6.2.3.4.6.6 External . . . 247
6.2.3.4.7 Obstacle Detection . . . 248
6.2.3.4.7.1 External . . . 248
6.2.3.4.7.2 Obsens . . . 249
6.2.3.4.7.3 Range sensors . . . 251
6.2.3.4.8 Pressure . . . 252
6.2.3.4.9 RPM . . . 253
6.2.3.4.10 Speed Down . . . 254
6.2.3.4.11 Ultrasounds . . . 255
6.2.3.4.11.1 Sensor . . . 255
6.2.3.4.11.2 Base . . . 256
6.2.3.4.11.3 Transmitter . . . 257
6.2.3.4.11.4 Navigation . . . 257
6.2.3.5 Stick Configuration . . . 259
6.2.3.5.1 Arcade trim . . . 259
6.2.3.5.2 Local Sources . . . 260
6.2.3.5.2.1 Transmitter . . . 260
6.2.3.5.2.2 PPM . . . 260
6.2.3.5.2.3 Exponential . . . 262
6.2.3.5.2.4 Trim . . . 262
6.2.3.5.2.5 Output . . . 263
6.2.3.5.2.6 Virtual Stick . . . 264
6.2.3.5.3 Test Stick . . . 266
6.2.3.5.4 Stick . . . 267
6.2.3.5.4.1 Wizard Stick . . . 271
6.2.3.5.4.2 Mask Servos . . . 272
6.2.3.5.4.3 Transmitter Inputs . . . 274
6.2.3.6 Others . . . 275
6.2.3.6.1 Payload . . . 275
6.2.3.6.1.1 Payload Operation . . . 277
6.2.3.6.2 SARA . . . 281
6.2.3.6.3 XPC Uint8 . . . 282
6.2.3.6.3.1 Tunnel . . . 282
6.2.3.6.4 XPC CAP . . . 284
6.2.4 Control . . . 285
6.2.4.1 Phases . . . 285
6.2.4.1.1 Guidance . . . 285
6.2.4.1.1.1 Guidance Variables . . . 285
6.2.4.1.1.2 Taxi . . . 286
6.2.4.1.1.3 VTOL . . . 288
6.2.4.1.1.4 Climb . . . 291
6.2.4.1.1.5 Cruise . . . 294
6.2.4.1.1.6 Hold . . . 297

6.2.4.1.1.7 Landing . . . 298
6.2.4.1.1.8 Yaw . . . 301
6.2.4.1.1.9 Rendezvous . . . 302
6.2.4.1.2 Loop . . . 305
6.2.4.1.2.1 PID Controller . . . 305
6.2.4.1.2.2 Adaptative-Predictive Control . . . 308
6.2.4.1.2.3 Gain Scheduling . . . 309
6.2.4.1.2.4 Fixed Control . . . 312
6.2.4.1.2.5 PID Settings . . . 313
6.2.4.1.2.6 Exporting PIDs To Other Phases . . . 315
6.2.4.1.3 Arcade . . . 315
6.2.4.1.3.1 Arcade Mode Settings . . . 315
6.2.4.1.4 TC Panel . . . 316
6.2.4.1.4.1 PID Gains . . . 317
6.2.4.1.4.2 Cruise Guidance . . . 319
6.2.4.2 Runway . . . 323
6.2.4.2.1 Runway . . . 323
6.2.4.2.2 Spot . . . 326
6.2.4.3 Envelope . . . 326
6.2.4.4 Smooth . . . 328
6.2.4.5 Modes . . . 330
6.2.4.6 Arcade Axis . . . 331
6.2.5 Navigation . . . 334
6.2.5.1 Wind Estimation . . . 334
6.2.6 Automations . . . 336
6.2.6.1 Events . . . 336
6.2.6.1.1 Phase . . . 336
6.2.6.1.2 Variable . . . 336
6.2.6.1.3 Alarm . . . 338
6.2.6.1.4 Mode . . . 339
6.2.6.1.5 Button . . . 339
6.2.6.1.6 Route . . . 340
6.2.6.1.7 Timer . . . 341
6.2.6.1.8 Polygon . . . 342
6.2.6.2 Actions . . . 345
6.2.6.2.1 Change Active Sensor . . . 345
6.2.6.2.2 Custom TX . . . 346
6.2.6.2.3 Envelope . . . 347
6.2.6.2.4 Go To . . . 347
6.2.6.2.5 Track . . . 348
6.2.6.2.5.1 Position . . . 349
6.2.6.2.5.2 By cam . . . 350
6.2.6.2.5.3 Follow Leader . . . 351
6.2.6.2.6 Terrain Obstacle . . . 352
6.2.6.2.7 Arcade Trim . . . 352
6.2.6.2.8 Navigation . . . 353
6.2.6.2.9 Mode . . . 354
6.2.6.2.10 Output . . . 355
6.2.6.2.11 Phases . . . 357
6.2.6.2.12 Select Arcade Axis . . . 358
6.2.6.2.13 Periodical . . . 359
6.2.6.2.14 Run Operation . . . 361
6.2.6.2.15 User Log . . . 361
6.2.6.2.16 Variable . . . 362

6.2.6.3 Other Options . . . 366
6.2.7 Variables . . . 366
6.2.7.1 System Variables . . . 366
6.2.7.1.1 Names . . . 366
6.2.7.2 Telemetry . . . 371
6.2.7.2.1 Data Link . . . 371
6.2.7.2.2 Onboard Log . . . 374
6.2.7.2.3 User Log . . . 374
6.2.7.2.4 Fast Log . . . 375
6.2.7.3 Sniffer . . . 376
6.2.7.4 List of Variables . . . 378
6.2.7.4.1 32 VAR . . . 378
6.2.7.4.2 BIT . . . 385
6.2.7.4.3 16 VAR . . . 386
6.2.8 Panel . . . 388
6.2.8.1 Short Phases . . . 388
6.2.8.2 Checklists . . . 389
6.2.8.3 Confirmation . . . 391
6.2.8.4 Notifications . . . 393
6.2.9 HIL . . . 395
6.2.9.1 Autopilot Configuration . . . 396
6.2.9.2 Adding New variable . . . 397
6.2.9.3 Veronte Pipe communications . . . 399
6.2.10 Blocks . . . 400
6.2.11 Radio . . . 404
6.3 Installation . . . 407
6.3.1 System Position . . . 407
6.4 License & Safe Mode . . . 408
6.4.1 License . . . 408
6.4.2 Safe Mode . . . 410
6.5 Overview . . . 411
6.6 Side Panel Options . . . 411

7 Mission configuration **413**
7.1 Mission Setup . . . 413
7.1.1 Introducction . . . 413
7.1.2 Terrain profile and Magfield . . . 414
7.1.3 Marks . . . 417
7.2 Mission Tools . . . 418
7.3 Mission toolbar . . . 419

8 Operation **421**
8.1 LOG Record . . . 421
8.1.1 Create PDF . . . 422
8.2 Veronte Panel . . . 423
8.3 Flight Options . . . 424
8.4 In-flight Settings . . . 424
8.5 In-flight mission edition . . . 425

9 Post Flight **427**
9.1 Tour Play . . . 427
9.2 Data Export . . . 429
9.2.1 Data link . . . 429
9.2.2 LOG . . . 431

10 Simulation **435**
10.1 Professional HIL . . . 435
10.1.1 Mounting . . . 435
10.1.1.1 Cable Connection . . . 435
10.1.1.2 Veronte Pipe Configuration . . . 436
10.1.1.2.1 Autopilot Configuration . . . 436
10.1.1.2.2 Veronte Pipe Communications . . . 438
10.1.2 X-Plane 10 Settings . . . 438
10.1.2.1 X-Plane 10 Configuration . . . 438
10.1.2.1.1 Aircraft Model Installation . . . 439
10.1.2.1.2 X-Plane 10 Setup . . . 439
10.1.2.2 Low Performance Computer Configuration . . . 443
10.1.3 X-Plane 11 Settings . . . 443
10.1.3.1 X-Plane 11 Configuration . . . 443
10.1.3.1.1 Aircraft Model Installation . . . 444
10.1.3.1.2 X-Plane 11 Setup . . . 444
10.1.4 Operation . . . 445
10.2 SIL Simulink . . . 447
10.2.1 Autopilot Simulation . . . 448
10.2.2 Sensors Simulation & Input Examples . . . 452
10.2.3 Monitoring Telemetry with Simulink . . . 460
10.2.4 Complete Simulation . . . 462
10.2.5 Connecting Simulink & Veronte Pipe . . . 464
10.2.6 Dealing with PDI files . . . 467
10.3 3D Simulation . . . 468
10.3.1 Veronte Pipe Configuration . . . 469

11 Examples **473**
11.1 4G Communication . . . 473
11.1.1 4G Communication with Veronte Autopilot . . . 473
11.2 Configurations . . . 479
11.2.1 Device Configurations . . . 479
11.2.1.1 Magnetometer Honeywell HMR2300-232 . . . 479
11.2.1.2 Radar Altimeter Smartmicro . . . 481
11.2.1.3 Volz Servo . . . 485
11.2.2 Plataform Configurations . . . 492
11.2.2.1 Fixed Wing-Mentor . . . 492
11.2.2.1.1 Servo Configuration . . . 492
11.2.2.1.2 Mission Phases . . . 495
11.2.2.1.2.1 Takeoff . . . 495
11.2.2.1.2.2 Climbing . . . 496
11.2.2.1.2.3 Cruise . . . 498
11.2.2.1.2.4 Hold . . . 500
11.2.2.1.2.5 Landing . . . 501
11.2.2.1.2.6 Flare . . . 502
11.2.2.1.3 Automations . . . 503
11.2.2.1.3.1 Takeoff to Climb . . . 504
11.2.2.1.3.2 Climbing to Cruise . . . 505
11.2.2.1.3.3 Radio Error . . . 506
11.2.2.1.3.4 Low Battery . . . 507
11.2.2.1.3.5 Landing to Flare . . . 508
11.2.2.1.3.6 Stick Auto . . . 509
11.2.2.2 Flying Wing-W210 . . . 511
11.2.2.2.1 Servo Configuration . . . 511

11.2.2.2.1.1 Servos Output . . . 511
11.2.2.2.1.2 SU Matrix . . . 513
11.2.2.2.1.3 System Trim . . . 515
11.2.2.2.2 Mission Phases . . . 517
11.2.2.2.2.1 Standby . . . 517
11.2.2.2.2.2 Hand-Launch . . . 518
11.2.2.2.2.3 Catapult . . . 520
11.2.2.2.2.4 Separation . . . 521
11.2.2.2.2.5 Climbing . . . 522
11.2.2.2.2.6 Cruise . . . 524
11.2.2.2.2.7 Hold . . . 526
11.2.2.2.2.8 Landing . . . 526
11.2.2.2.2.9 Flare . . . 528
11.2.2.2.3 L200 Catapult . . . 530
11.2.2.2.3.1 Configuration . . . 530
11.2.2.2.3.2 Actuator Trimming . . . 530
11.2.2.2.3.3 Launch Automation . . . 531
11.2.2.3 Quadcopter-M400 . . . 535
11.2.2.3.1 Servo Configuration . . . 535
11.2.2.3.1.1 Servos Output . . . 535
11.2.2.3.1.2 SU Matrix . . . 537
11.2.2.3.1.3 System Trim . . . 538
11.2.2.3.2 Mission Phases . . . 539
11.2.2.3.2.1 Standby . . . 539
11.2.2.3.2.2 Takeoff . . . 539
11.2.2.3.2.3 Hover . . . 541
11.2.2.3.2.4 Cruise . . . 542
11.2.2.3.2.5 Landing . . . 544
11.2.2.3.2.6 VHOLD . . . 546
11.2.2.3.2.7 Return to Home . . . 547
11.2.2.4 Helicopter T-Rex600 . . . 551
11.2.2.4.1 Servo Configuration . . . 551
11.2.2.4.1.1 Servos Output . . . 551
11.2.2.4.1.2 SU Matrix . . . 553
11.2.2.4.1.3 System Trim . . . 555
11.2.2.4.2 Mission Phases . . . 557
11.2.2.4.2.1 Stanby . . . 557
11.2.2.4.2.2 Engine Off . . . 558
11.2.2.4.2.3 Ignite . . . 561
11.2.2.4.2.4 Ready . . . 563
11.2.2.4.2.5 Takeoff . . . 565
11.2.2.4.2.6 Hover . . . 566
11.2.2.4.2.7 Cruise . . . 567
11.2.2.4.2.8 Landing . . . 569
11.2.2.4.2.9 Hold . . . 570
11.2.2.5 Hybrid . . . 573
11.2.2.5.1 Servo Configuration . . . 573
11.2.2.5.1.1 Quadcopter servos . . . 573
11.2.2.5.1.2 Servos Output . . . 573
11.2.2.5.1.3 SU Matrix . . . 575
11.2.2.5.1.4 Plane Servos . . . 576
11.2.2.5.1.5 Servos Output . . . 576
11.2.2.5.1.6 System Trim . . . 578
11.2.2.5.2 Mission Phases . . . 583

11.2.2.5.2.1 Standby . . . 583
11.3 Automations . . . 586
11.3.1 Veronte Panel Buttons . . . 586
11.3.1.1 Phase Change Buttons . . . 586
11.3.1.2 Generic Button . . . 588
11.3.2 Phase Changing . . . 591
11.3.2.1 Takeoff to Climb Change . . . 591
11.3.2.2 Climbing to Cruise Change . . . 592
11.3.2.3 Landing to Flare Change . . . 593
11.3.3 Failsafe . . . 594
11.3.3.1 Radio Error . . . 595
11.3.3.2 Low Bettery . . . 597
11.3.3.3 GPS Signal lost . . . 597
11.3.4 Photogrammetry . . . 598
11.3.5 Others . . . 600
11.3.5.1 Stick Auto . . . 600
11.3.5.2 Automatic Landing (Multicopterss) . . . 601
11.4 Lidar Integration . . . 603
11.4.1 Lidar Configuration . . . 603
11.4.2 Veronte configuration - ACD In . . . 603
11.4.3 Veronte Configuration - altimeter . . . 604
11.5 Veronte Tracker . . . 605
11.5.1 Veronte Ground Configuration . . . 605
11.5.2 Veronte Air Configuration . . . 612
11.5.3 Tracking Operation . . . 613
11.6 Autotune . . . 617
11.6.1 Modes Configuration . . . 618
11.6.2 Workspace Configuration . . . 619
11.7 USB joystick integration . . . 622
11.7.1 Adding a joystick . . . 622
11.7.2 Defining a virtual stick . . . 623

12 Troubleshooting **625**
12.1 Veronte not detected . . . 625
12.2 PFD Attitude . . . 626
12.3 Altitude Error . . . 627
12.3.1 Atmosphere Calibration . . . 627
12.3.2 Terrain Altitude Meshes . . . 629
12.4 HIL Simulation Fails . . . 629
12.5 Telemetry Overflow Error . . . 630
12.6 Communications Incompatibility . . . 630
12.7 No enough information of terrain height . . . 631
12.8 License Expired . . . 632

13 Acronyms & Definitions **635**
13.1 Acronyms . . . 635
13.2 Definitions . . . 636

14 Axes Convention **637**

15 Feedback **639**
15.1 Feedback Panel . . . 639

16 4x Redundant Autopilot **641**
16.1 Introducction . . . 641

16.2 Setup 642
16.2.1 Radio 642
16.2.2 CAN bus 642
16.2.2.1 Variable value message: 643
16.2.2.2 Status message: 643
16.2.2.3 Variable value message: 643
16.2.2.4 Status message: 644
16.2.3 Control 644

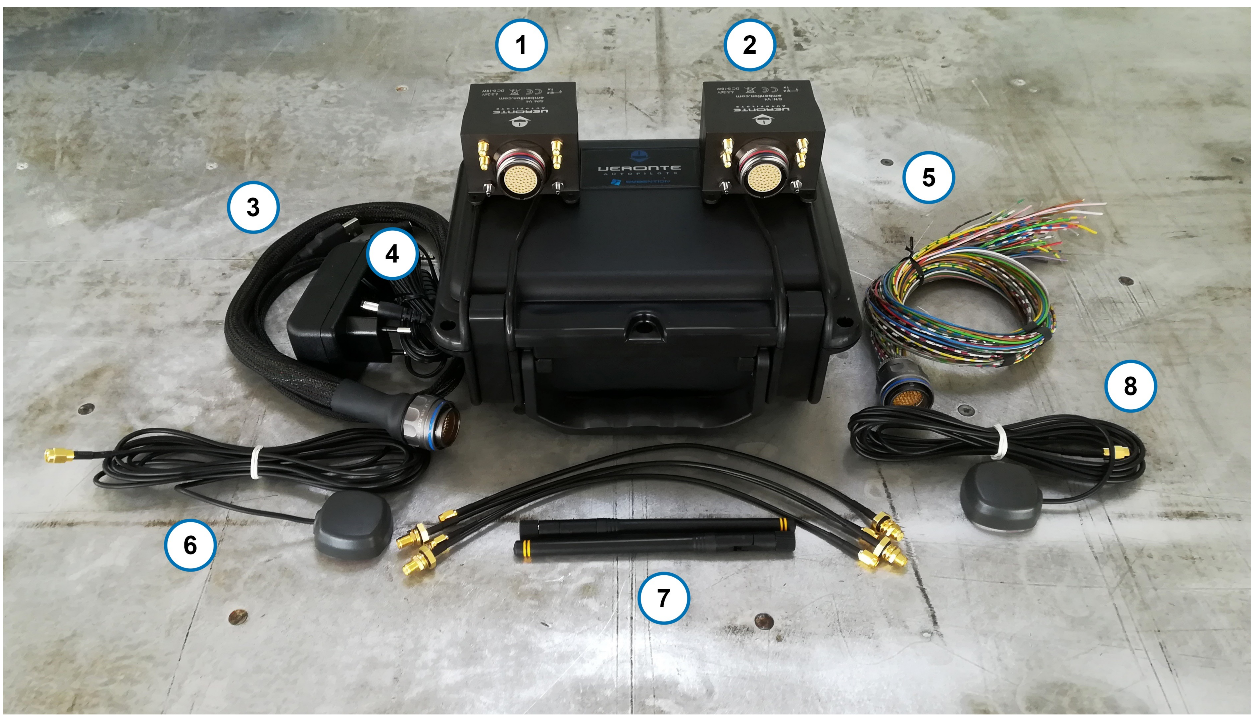
CHAPTER

ONE

QUICK START

1.1 First Steps

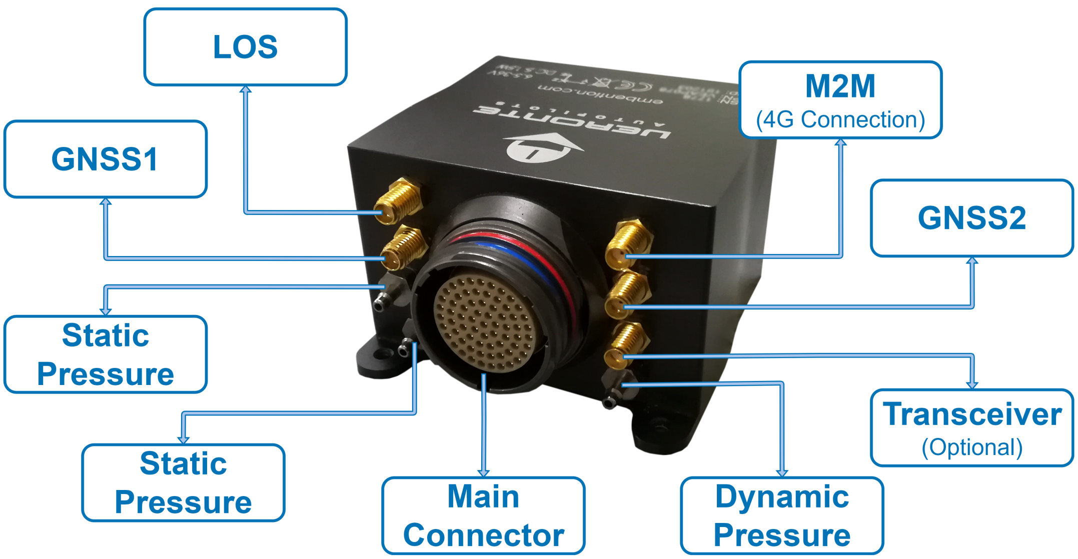
This section provides basic information of the connections of the autopilot and how to have it ready-to-use connected to the provided software **Veronte Pipe**.



Autopilot Kit

1.1.1 Kit Elements

1. Veronte Autopilot
2. Veronte Autopilot
3. Autopilot Harness CS, including USB & Joystick connectors and 50cm wire leads
4. 12V Power Supply (European plug)
5. Autopilot Harness including 50cm wire leads
6. GNSS Antenna (Standard, SSMA Male connector)
7. 2x LOS Antenna SMA Male connector + SMA/SSMA adapters
8. GNSS Antenna (Standard, SSMA Male connector)

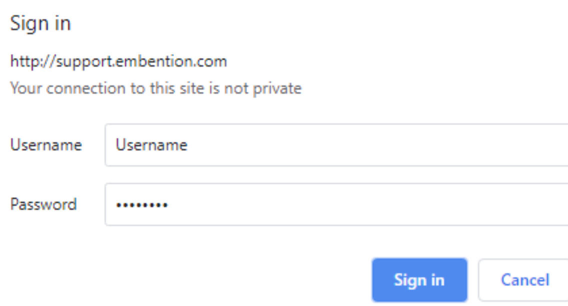


Veronte 4.5 Connections

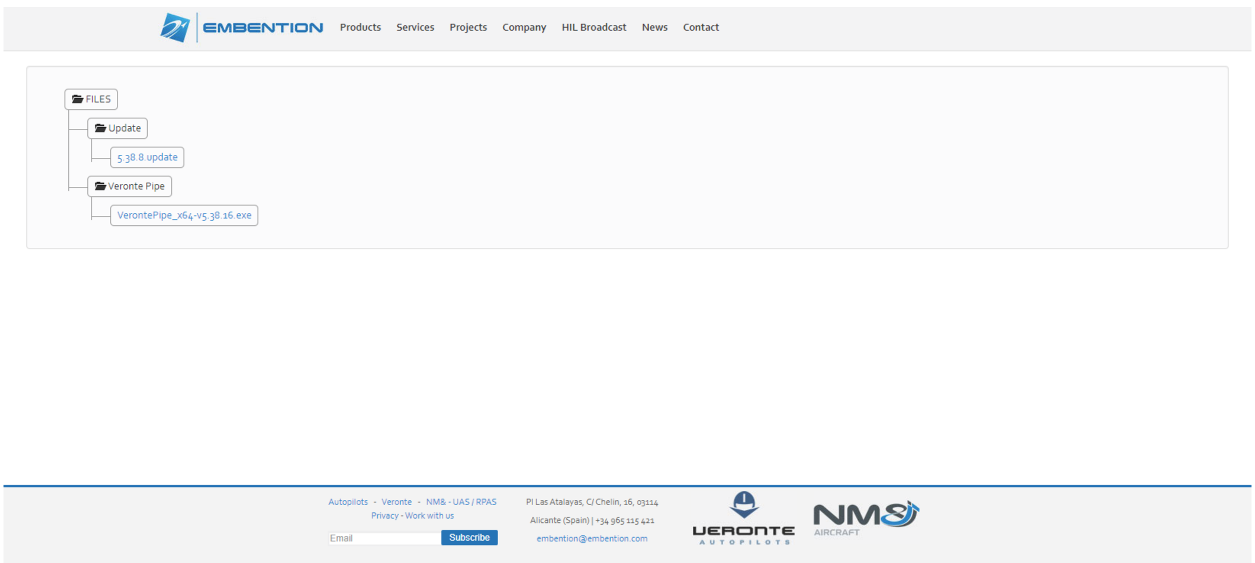
For more information about the autopilot's hardware (vibration isolation, orientation and location on the platform, pressure lines, etc) check *Hardware Installation*.

1.1.2 Veronte Pipe

Once the Veronte is delivered, a shared folder between the Customer and Embention is automatically created. The user will receive an email from the Support Team containing the information needed to access. If the email is not received within 72h, please contact with support@embention.com and our Support Team will be happy to help you.



FTP Access



FTP folder content

The shared folder contains the last version of Veronte Pipe and firmware. Please, download and install the executable file provided. Veronte units are normally shipped with the last version of firmware.

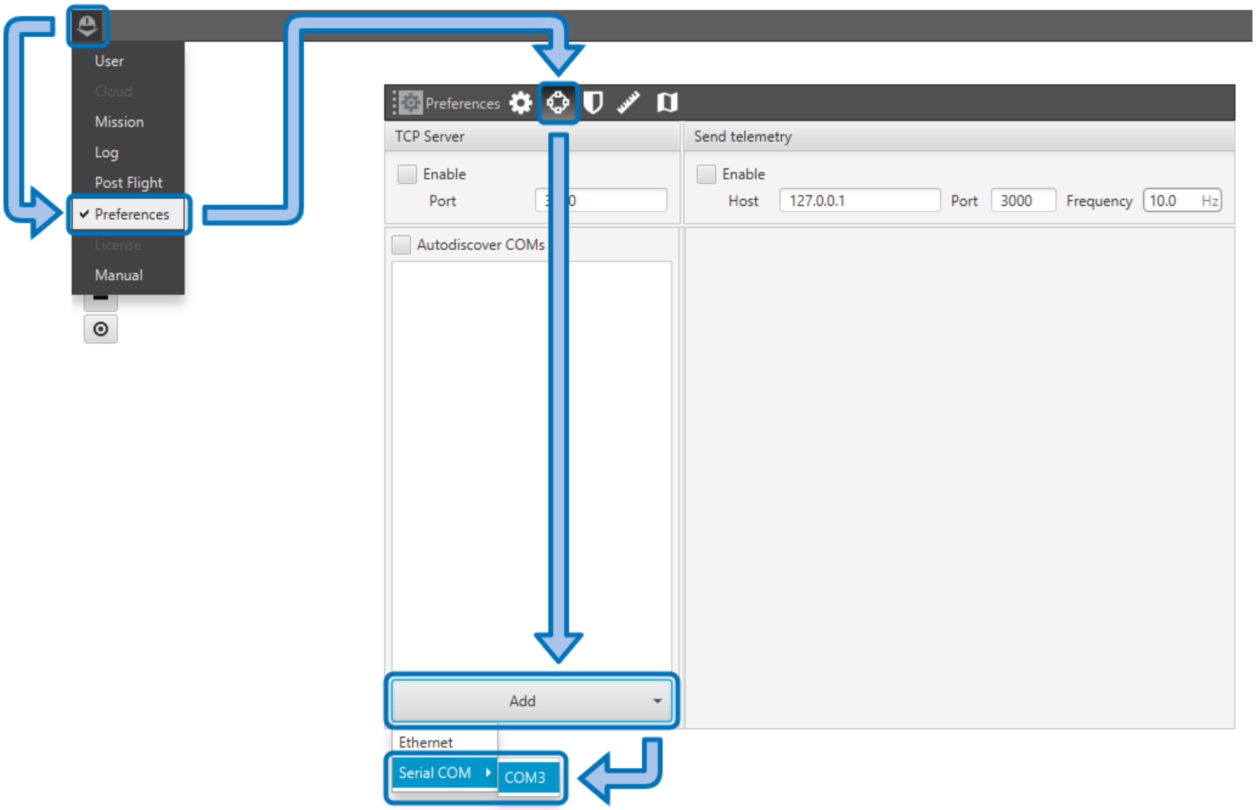
imgs/quick_start_guide/pipe_version.png

Pipe Installation

For more information about Veronte Pipe intallation process, system requirements and OS, please check the *Pipe Configuration* section.

Once Veronte Pipe is installed it is necessary to configure the connection to the autopilot.

Click on Main Menu and then select Preferences in the pull-down menu. Now click on Add -> Serial COM. If only one unit is connected to the computer via USB, only one COM port will be available.



Connections Menu

This step is only performed once with every new unit. Once the configuration is done, the Veronte autopilot will be automatically detected by Veronte Pipe and it will show up in the right bar.

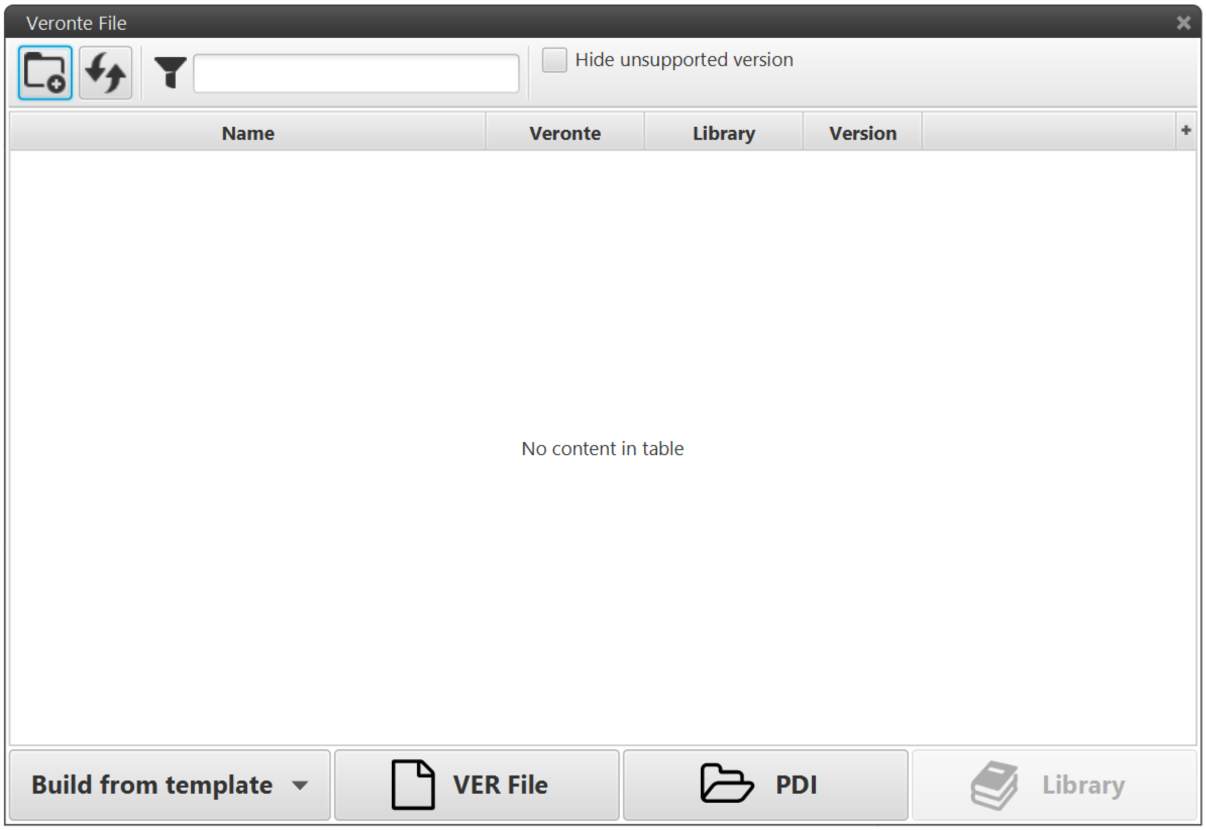
For more information about the Connection's menu check *Connections*.

1.2 Model Simulation

This section will allow the user to load a pre-existing model/configuration to the autopilot and perform a mission simulation with the combined help of the software X-Plane. The content is organized in 3 sub-sections: how to load the template into Veronte autopilot, setting XPlane and Veronte Pipe so they can communicate, and how to run a simulation.

1.2.1 Loading a Template to the Autopilot

When the autopilot is connected, in the side panel click on  and then click on Import.



Import a Configuration File

The **Import Configuration** option displays a window (shown in the previous figure) where the user can select a file to be uploaded. The option of interest for the scope of this section is **Build from template**. The software is fitted with a set of default configurations (templates) so the user can have reference of what a fixed wing, quadcopter or ground configuration should look like.

There are three other options to select a configuration file. For more information about the logical calibration of the servos check *Import Configurations*.

The configuration file is loaded in the software (Veronte Pipe), but it is not loaded on the autopilot until the Save button  is pressed.

1.2.2 X-Plane and Veronte Pipe Setup

Integration of Veronte Autopilot within the X-Plane simulator enables the user to perform a Harware-In-the-Loop (HIL) simulation. HIL simulation enables testing an effective model of the vehicle by adding the complexity of the plant under control.

1.2.2.1 Veronte Pipe

Once a template has been loaded into the autopilot, the HIL simulation parameters need to be set. On the side panel, click on  and then click on HIL. A pop-up window will appear (see the figure below) and it has to be filled in as follows:

IP	127.0.0.1
Receive Port	49005
Send Port	49000

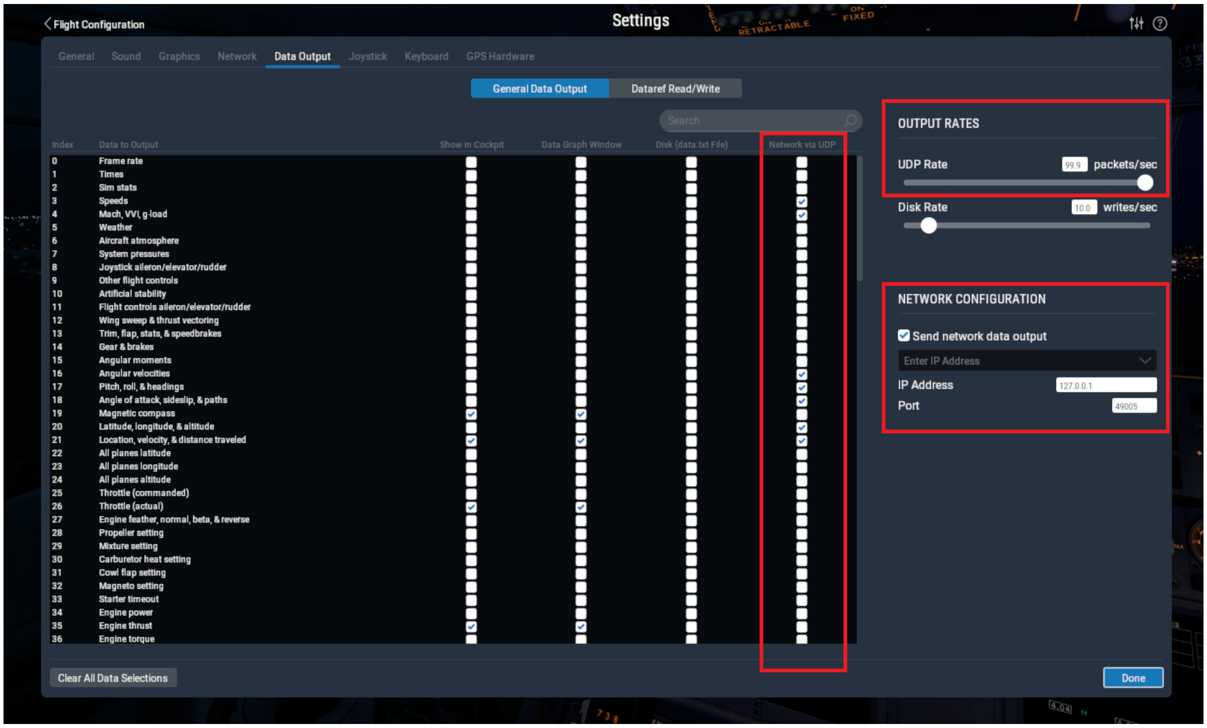
Veronte Pipe – HIL Configuration

1.2.2.2 X-Plane

X-Plane 10 or 11 are compatible with Veronte Pipe to perform HIL simulations. Here, necessary settings on X-Plane 11 to communicate with Veronte Pipe are detailed. If X-Plane 10 is used, the user can check *X-Plane 10 Settings*.

On X-Plane's main menu, go to Settings and then to Data Output. Here, the data tha will be shared with Veronte Pipe is defined:

- Variables 3, 4, 16, 17, 18, 20 and 21 should be selected on the column *Network via UDP*.
- UDP rate should be set to its maximum value of 99 packets/sec.
- Network configuration should be set to *IP Address* 127.0.0.1 and *Port* 49005.



X-Plane 11 I/O configuration

X-Plane 11 provides a tool called Plane Maker where the user can create the platform model. Once the vehicle model has been created, it can be integrated on the X-Plane 11 simulator by following next steps:

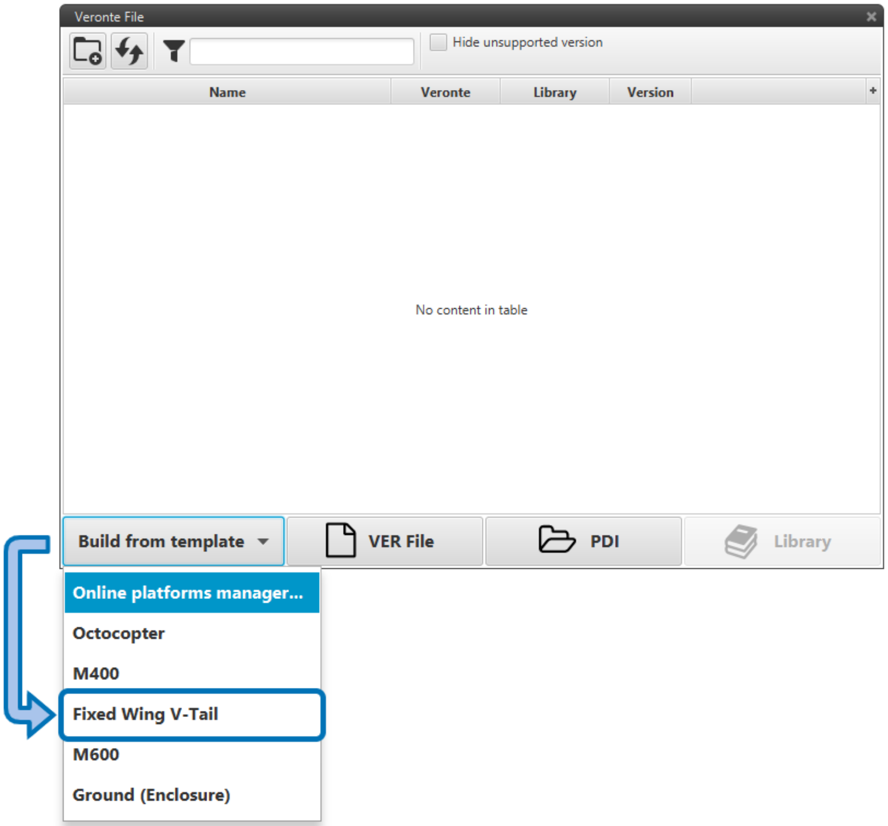
- Copy the model folder to the *Aircraft* folder within the X-Plane 11 installation directory.
- Copy the content of the *Airfoils* folder available on the aircraft model folder to the *Airfoils* directory within the X-Plane 11 installation directory.

1.2.3 Executing a Simulation

Before starting a simulation with one of the built-in templates of Veronte Pipe, the user needs to take into account the following:

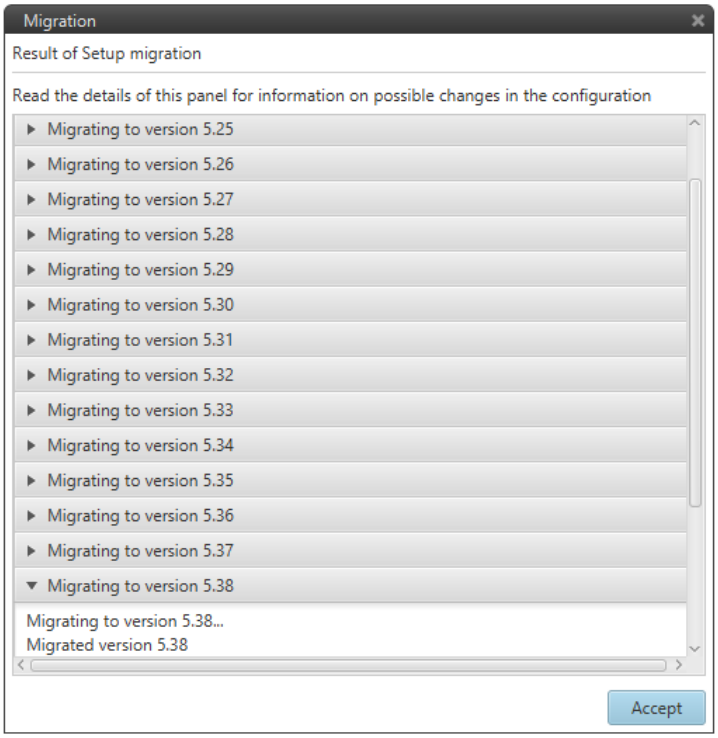
- The missions defined on this configurations are set at Alicante's airport (LEAL), with runway 10 as the default one.
- The flight envelope and all the flight control settings and transitions are tailored for the template model. Therefore, the user might need to modify some parameters in order to have their own model flying as desired.

Let us supose the user has created a fixed wing model with a V-tail configuration on Plane Maker. Then, the most suitable template on Veronte Pipe would be Fixed Wing V-Tail:



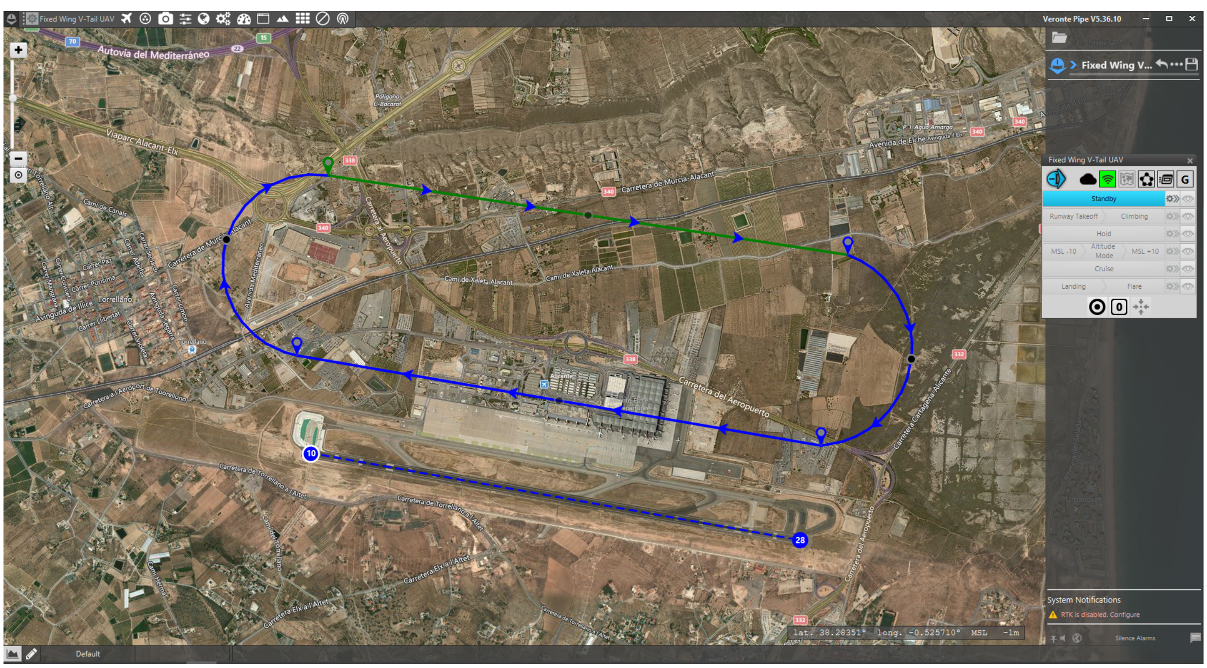
Fixed wing V-Tail template

A configuration merge and Migration windows will pop-up (see figure below). The user should accept both of them as they are. For more information on this topic go to *Import Configurations*.



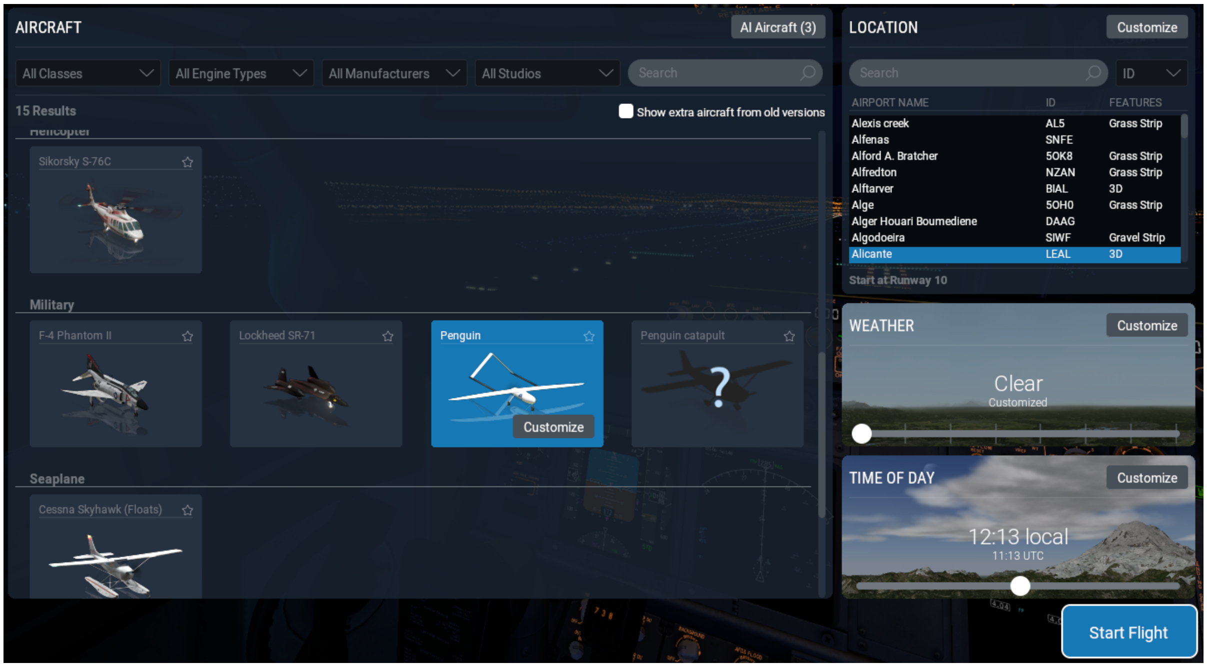
Pop-up window to end the import process

The Fixed Wing V-Tail template mission should look like this:



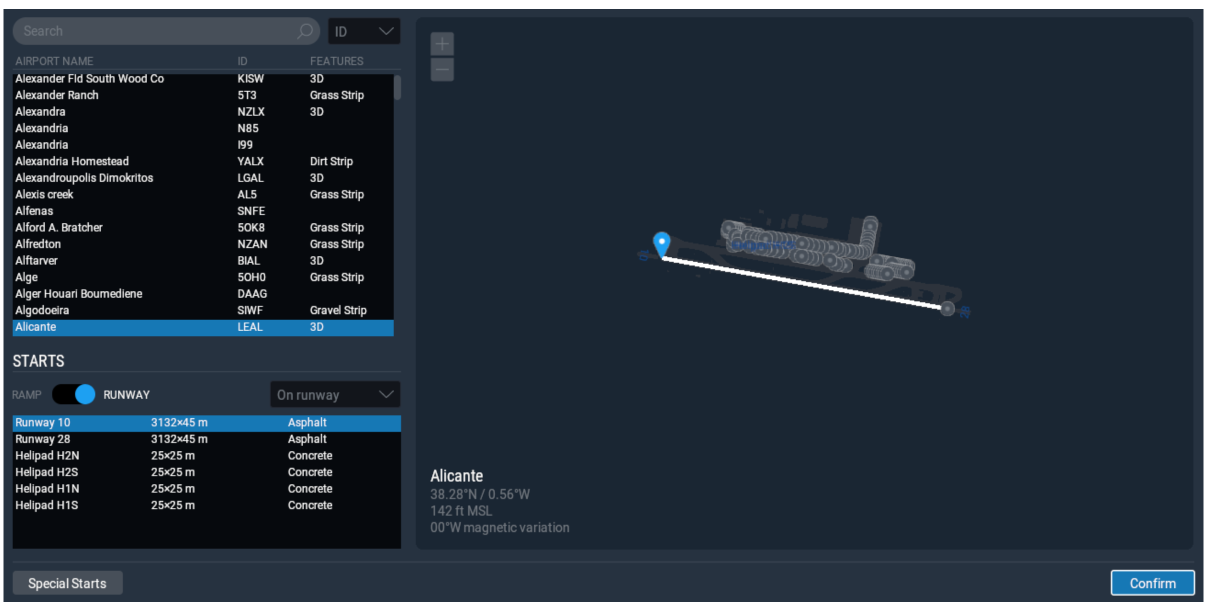
Template's mission

Now, the user needs to open X-Plane 11 and click on New Flight. On that menu the user will select the custom-made model and Alicante as the desired location (see the figure below):



X-Plane 11: new flight

One should click on *Customize* in order to select the desired runway for take-off (see the figure below):

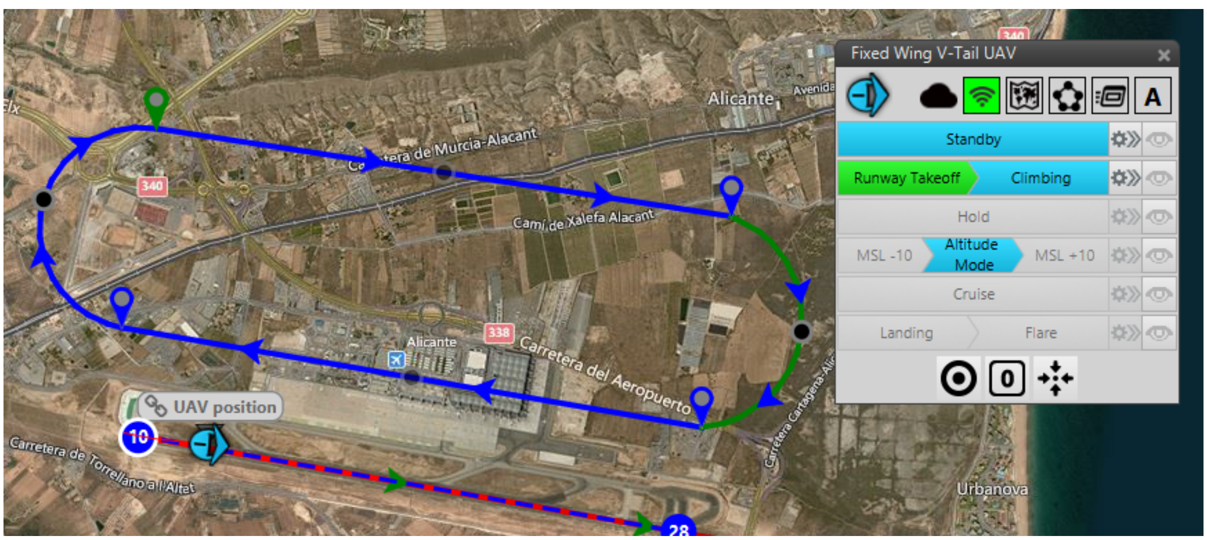


X-Plane 11: runway selection

After that the flight can already be started (click on *Start Flight*). The aircraft will appear at the header of runway 10 of Alicante's airport. Once that is done, connection between X-Plane 11 and Veronte Pipe can be stablished. Go to Pipe, click on , then HIL and click on Start. If every step was correctly done, GPS is simulated (from X-Plane) and then the UAV must be visible on Veronte Pipe map.

Autopilot visible on Veronte Pipe

Now the user can pass the system to Standby phase to start the flight. To do so click on the panel and the word *Standby* will turn green. After that, click on *Runway Takeoff* and the aircraft will carry on autonomously until reaching a Cruise phase. The transition between phases is defined in the Automations menu. For more infomation regarding that subject check *Automations*.

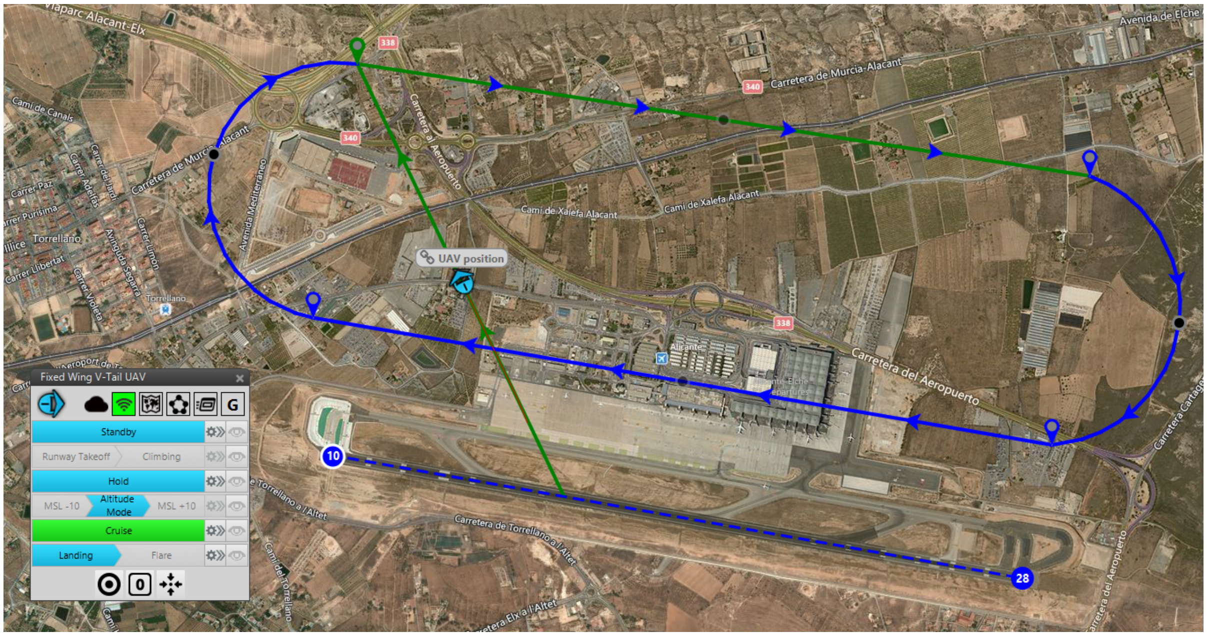


Take-off phase

The transition from take-off phase to climbing phase happens when the Indicated Air Speed (IAS) is greater than 20 m/s. The climbing phase (see figure below) automatically traces a climbing path.

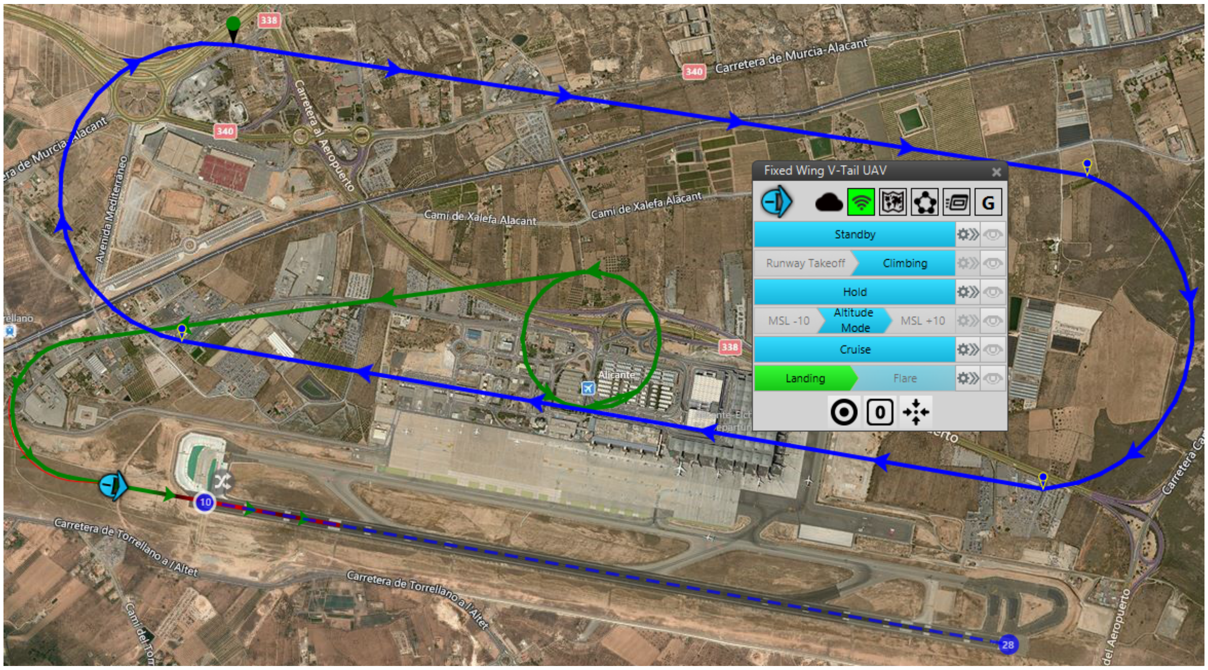
Climbing phase

But transittioning to cruise phase happens when only one condition is met: above ground level (AGL) altittude of more than 100 m.



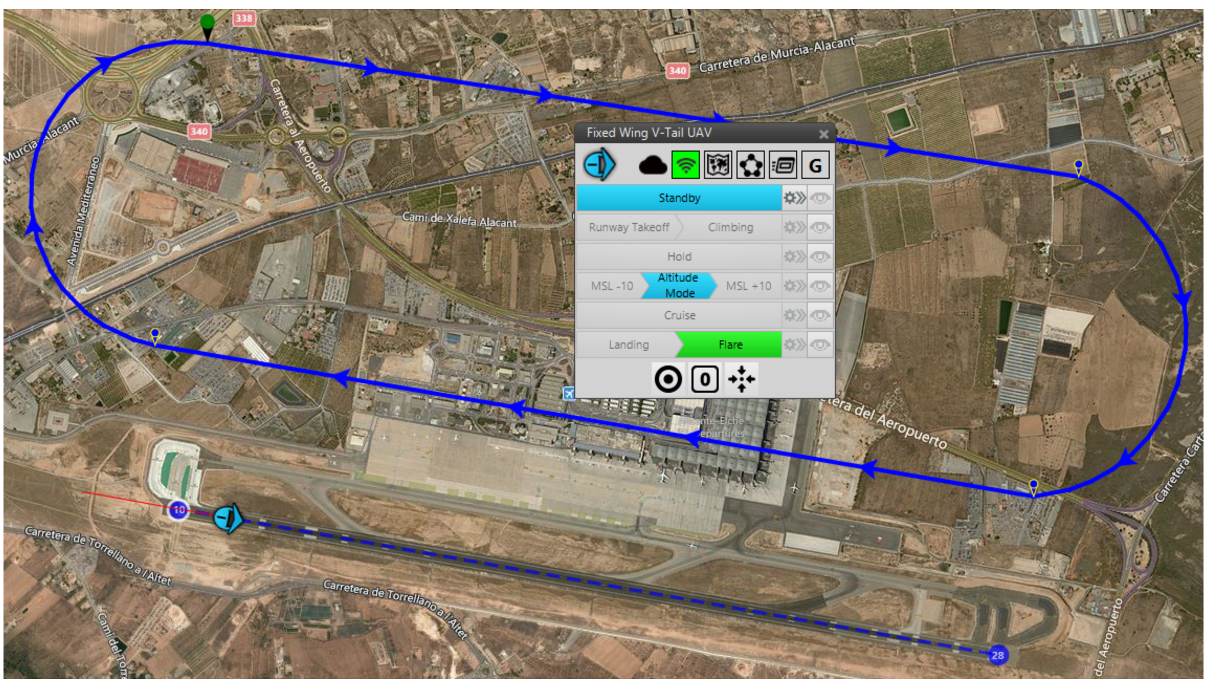
Cruise phase

Once in cruise mode the aircraft will follow the mission displayed above in blue, being in creen the present route. Unless you define an event to motivate the entry of landing phase (a certaing flight time, a certain altitude, etc), this template does not have that part automated. Upon clicking on *Landing*, a landing path is automatically traced.



Landing phase

The condition to change from landing to flare phase is double: having an AGL smaller than 10 m **and** having passed by a waypoint. For more information on mission planning/design check Mission Configuration.



Flare phase

After that, the aircraft will have succesfully landed and gone back to Standby phase. If the user need to adapt the flight phases to its model, one needs to go to the Control menu, and then go to phases. There, parameters as take-off speed, cruise speed, among others, can be modified. For more information phases definition :ref:`Phases <phases>PNG All the stepsPNGtailed above should serve as a starting point for the user to become more acquainted with HIL simulations. With the latter he or she will improve the definition and operation of the mission and the performance and control of the platform.

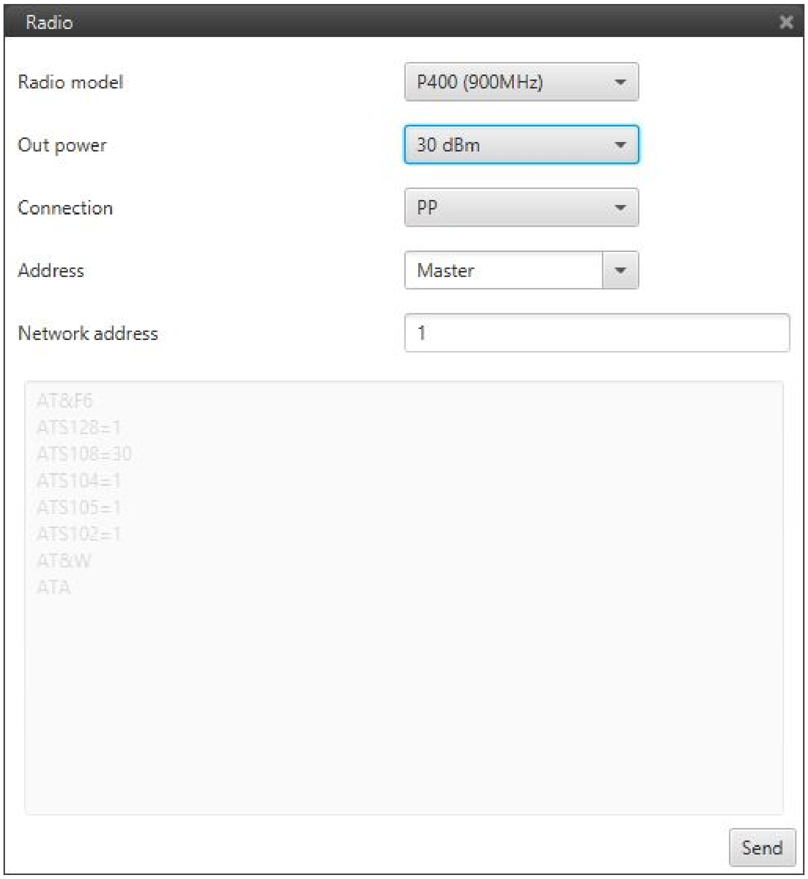
1.3 Autopilot Radio Pairing

This section includes the information of how to pair the **GND** and **AIR** autopilots. Also, the configuration of the parameters of the radio module inside each unit will be covered. The steps described in here must be first performed on the **GND** unit, and then on the **AIR** one.

First display the Setup Toolbar: click on  and then click on Setup. This toolbar allows the user to modify the main features of the autopilot.

Setup Toolbar

Now click on  Radio. The following menu pops up:



Radio Configuration Menu

All these parameters have to be set accordingly to the radio module installed in the Veronte autopilot. Contact Embention if you need the details of your specific radio module.

- **Radio Module:** each autopilot has only one of the following radio modules P400/900, P900 and P2400. Select here the radio module installed.
- **Output power:** set the desired power output.
- **Connection:** Point to Point or Point Multi Point. That will depend on your mission and on the number of autopilots required in the same LOS network.
- **Address:** Master or Slave. Select the role for each autopilot by selectig master or slave. By default the **AIR** unit is defined as slave and ground units as master.
- **Network address:** the network address is a number that must be equal between autopilots using the same network. First the master´s address should be filled in so when the slave's is set to the same value, the **GND** autopilot will find the **AIR** autopilot and bind a connection with it.

By pressing Send the user sets **all parameters** .

1.4 Calibrating Servos

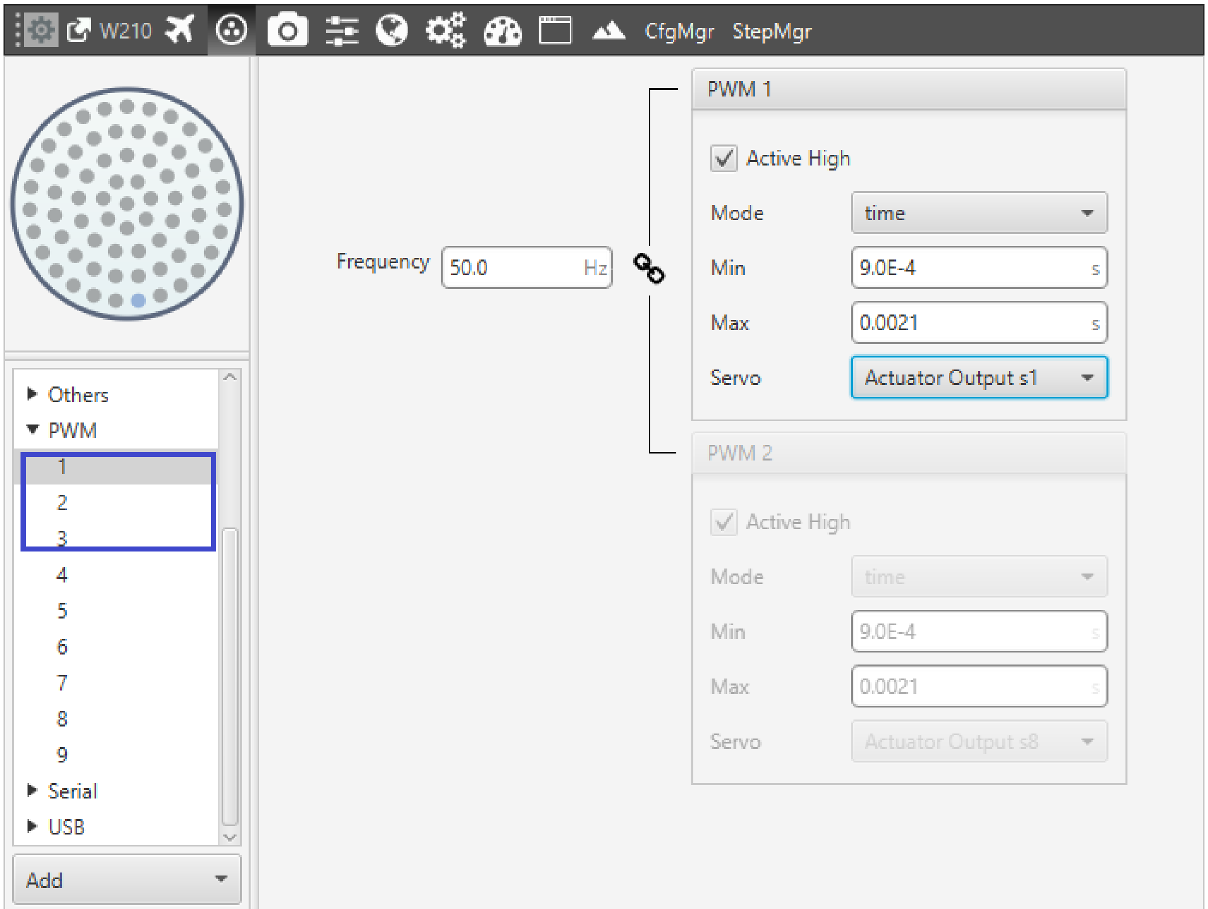
This section includes the information of how to calibrate the servos that control the attitude of the platform. Configuration of the servos includes: assignment of PWM pins to each servo, controller output to servo output relation, and trimming of the servos.

1.4.1 Servos Output

The first step of the process is the servos/actuators configuration. Let´s consider a flying wing as our platform.

The controls in this case are the two control surfaces (elevons) and throttle. Each one of these control variables corresponds to a PWM pin of the connector and they must be positioned in the same order in an **S** vector representing the **Actuator/Servo Outputs**.

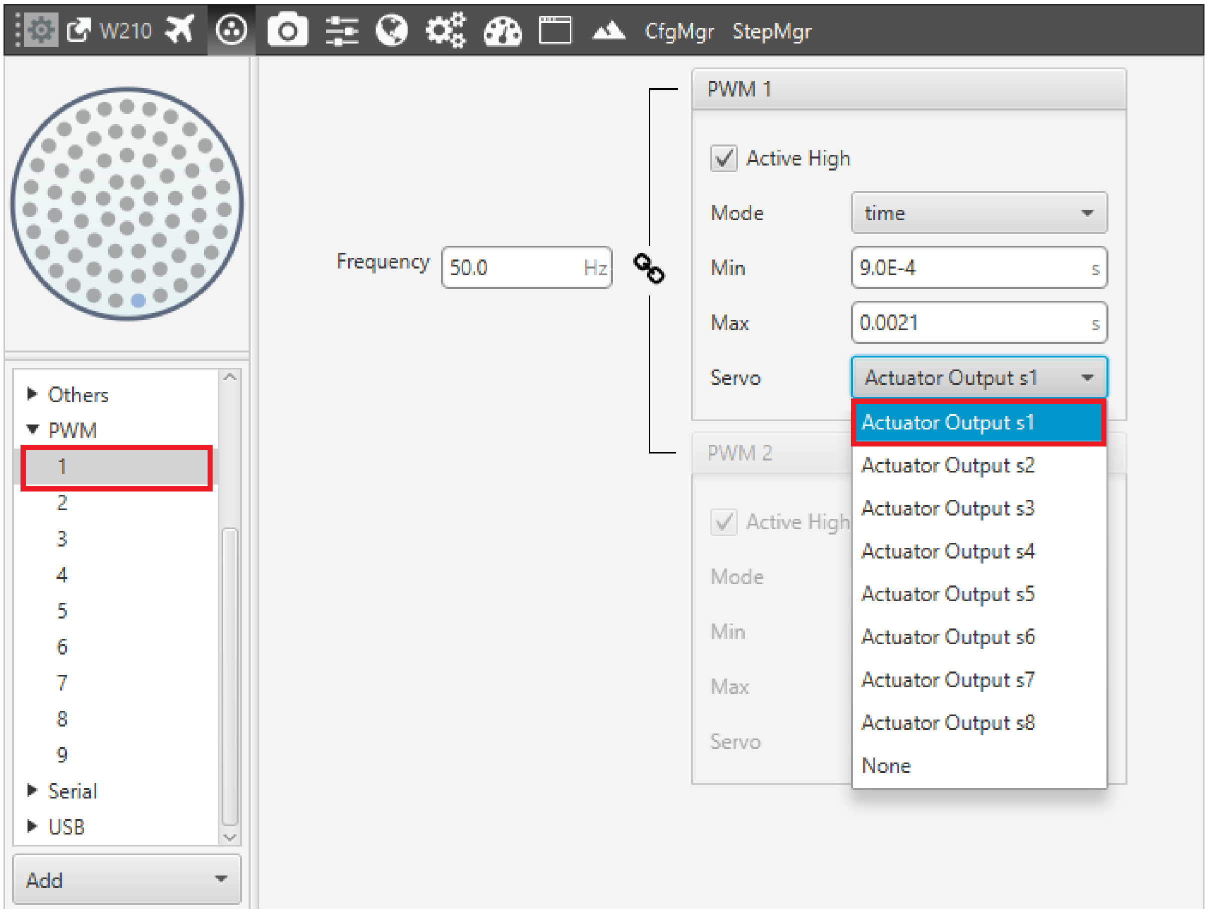
To assign the actuator outputs go the side panel , click on Setup, then go to Connections  and open the PWM tab.



Output-PWM pin links

In this case, only 3 pins are used:

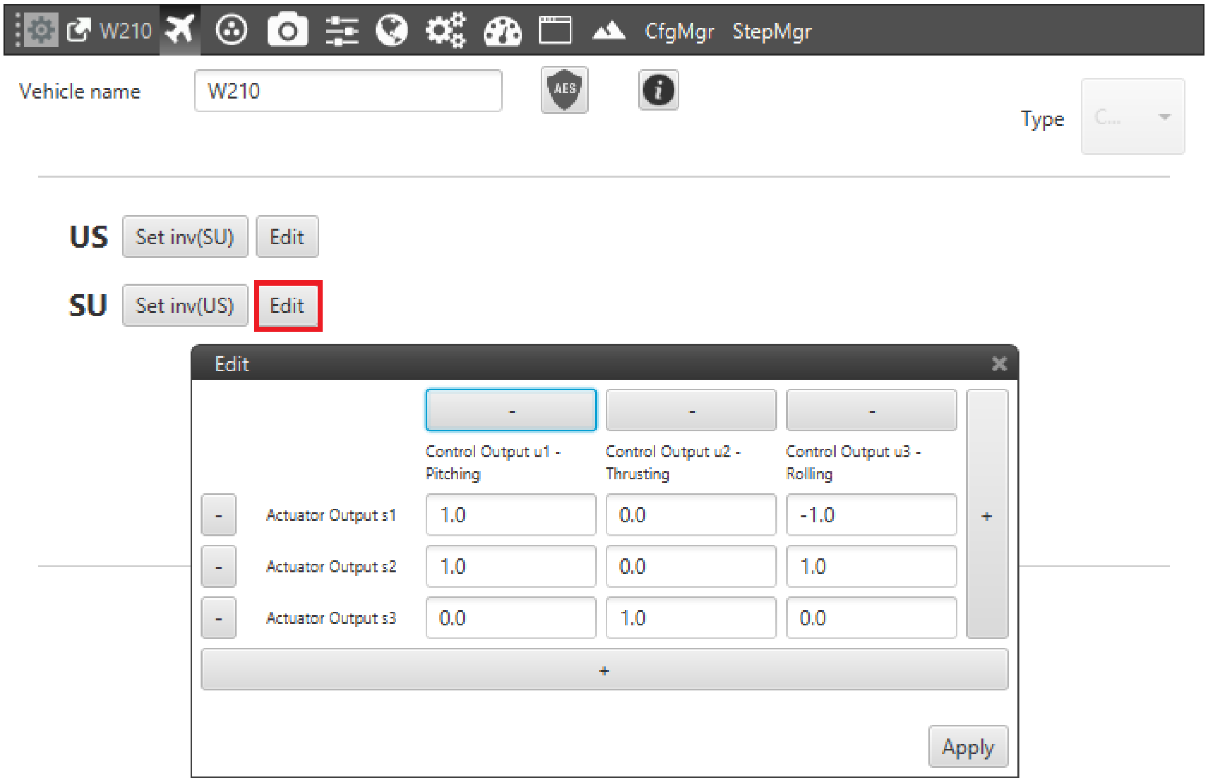
- Output 1 – Actuator Output s1 (**Elevon 1**)
- Output 2 – Actuator Output s2 (**Elevon 2**)
- Output 3 – Actuator Output s3 (**Throttle**)



Output 1 – PWM pin 1 configuration

1.4.2 SU Matrix

At this point, the **S** vector is defined and the **SU** matrix can be edited. At the Setup menu, go to Devices , then open the Actuators tab, and click on Logical. By clicking on **Edit** it is possible to configure the relation between the **Controller Outputs** (**U** vector) and the servo movements (**S** vector).



SU matrix editing

A flying wing is configured as follow:

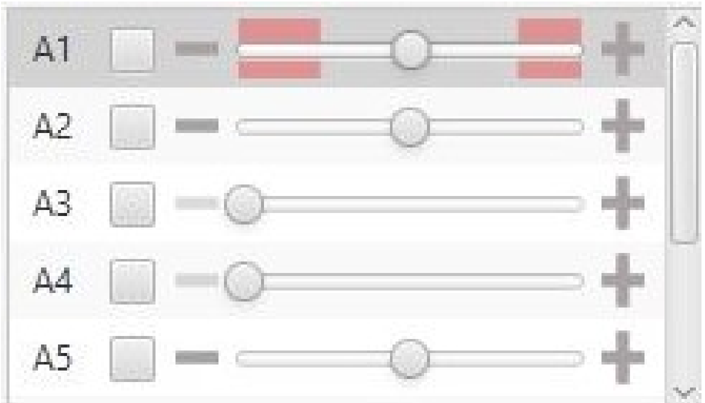
1. **Pitch Control:** control surfaces must be moved in the same direction to modify the pitching angle. The contribution of the actuators has same magnitude and direction.
2. **Thrust Control:** the Actuator Output 3 is the only one that allows a thrusting change.
3. **Roll Control:** in this case, the contribution of the actuators must be set with the same magnit:ref:`Logical <logical>`ude and reverse direction in order to perform a rotation around the body axis of the aircraft.

> **Warning:** The panel above considers the reference system of the aircraft. It should match the Autopilot´s one. In case it would not, it can be edited by clicking on the corresponding axis in order to reverse its direction.

For more information about the logical calibration of the servos check *Logical*.

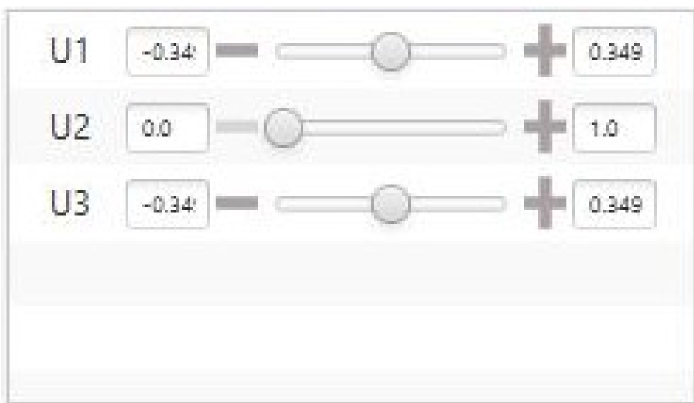
1.4.3 System Trim

As a final step, the system has to be trimmed. At the Setup menu, go to Devices , then open the Actuators tab, and click on Physical. The options that appear on the screen are presented as follows:



Servos Configuration Display

- **Servos:** this menu contains the servos of the platform. The signal to the system will only be sent if the checkbox next to the servo number (A1, A2. . .) is marked.



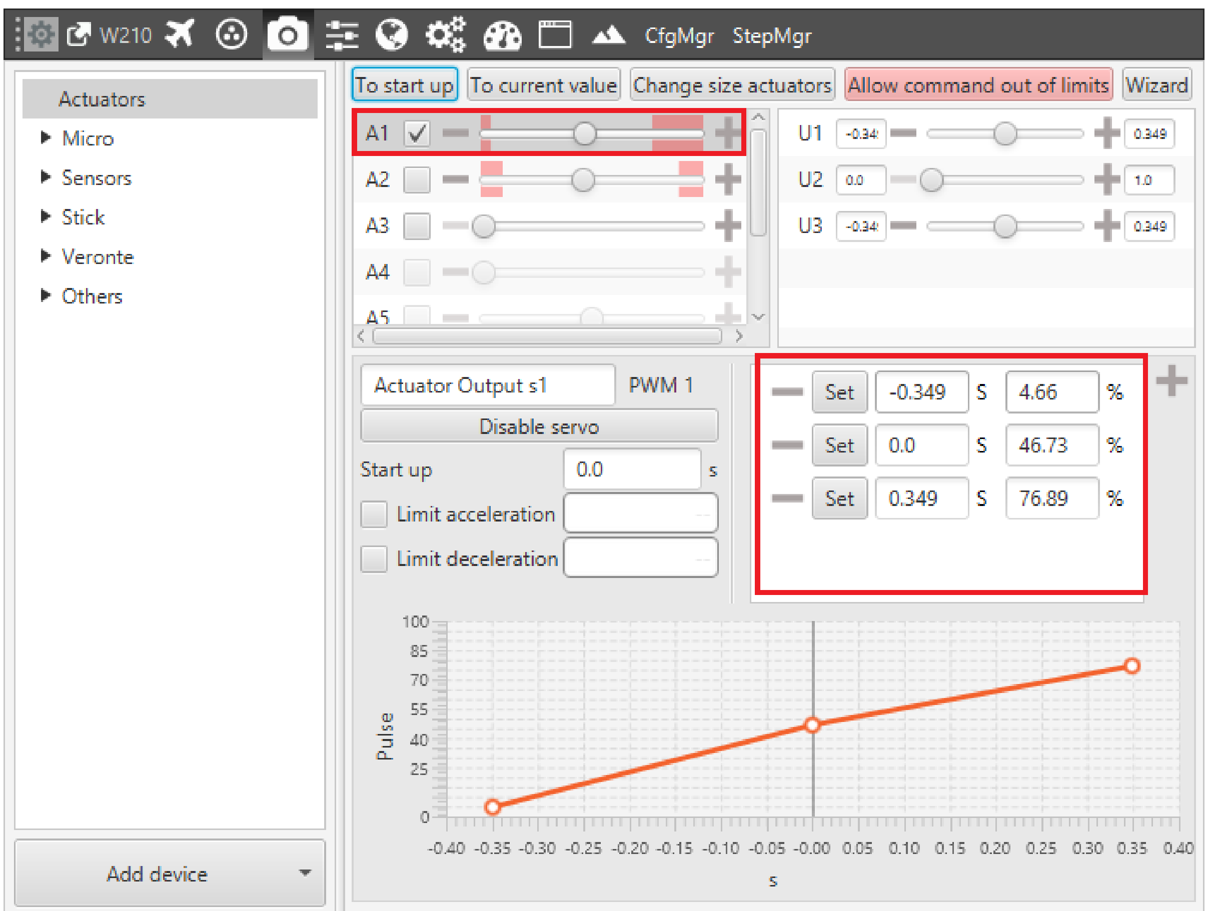
Control Signals Configuration Display

- **Control Signals:** this menu contains the variables (**U** vector) that represent the control signals generated by the system.

Servo PWM Configuration Display

- **Servo Position – PWM:** this option is used to set the transformation from a control position to a PWM signal. The example above is: a 20º degrees deflection (0.349 rad) of the right elevon corresponds to an 81'5 % pulse to be sent to the corresponding servo.
- The other options that appear on the screen are:
  - **To start up:** set initial values.
  - **To current value:** set the current value.
  - **Change size actuators:** change the number of actuators.
  - **Allow command out of limits:** allow manual control over the actuators limits established.
  - **Wizard:** recommended on first system configuration. It guides the user for configuring actuator limits and performance.
  - **Disable/Enable servos:** menu for enabling servos with their limits or just to disable them.
  - **Start up servo position:** set the pulse value for the Start up.
  - **Increasing/Decreasing Rate Limit:** set a limit for increasing/decreasing value of the servo.

The trim can be performed by moving the servos in three different positions: zero position, minimum and maximum deflection angle (angles are usually limited physically). These positions must be inserted and saved in the software by clicking on **Set** when the actuator is in the desired position. Otherwise, position can be introduced manually.



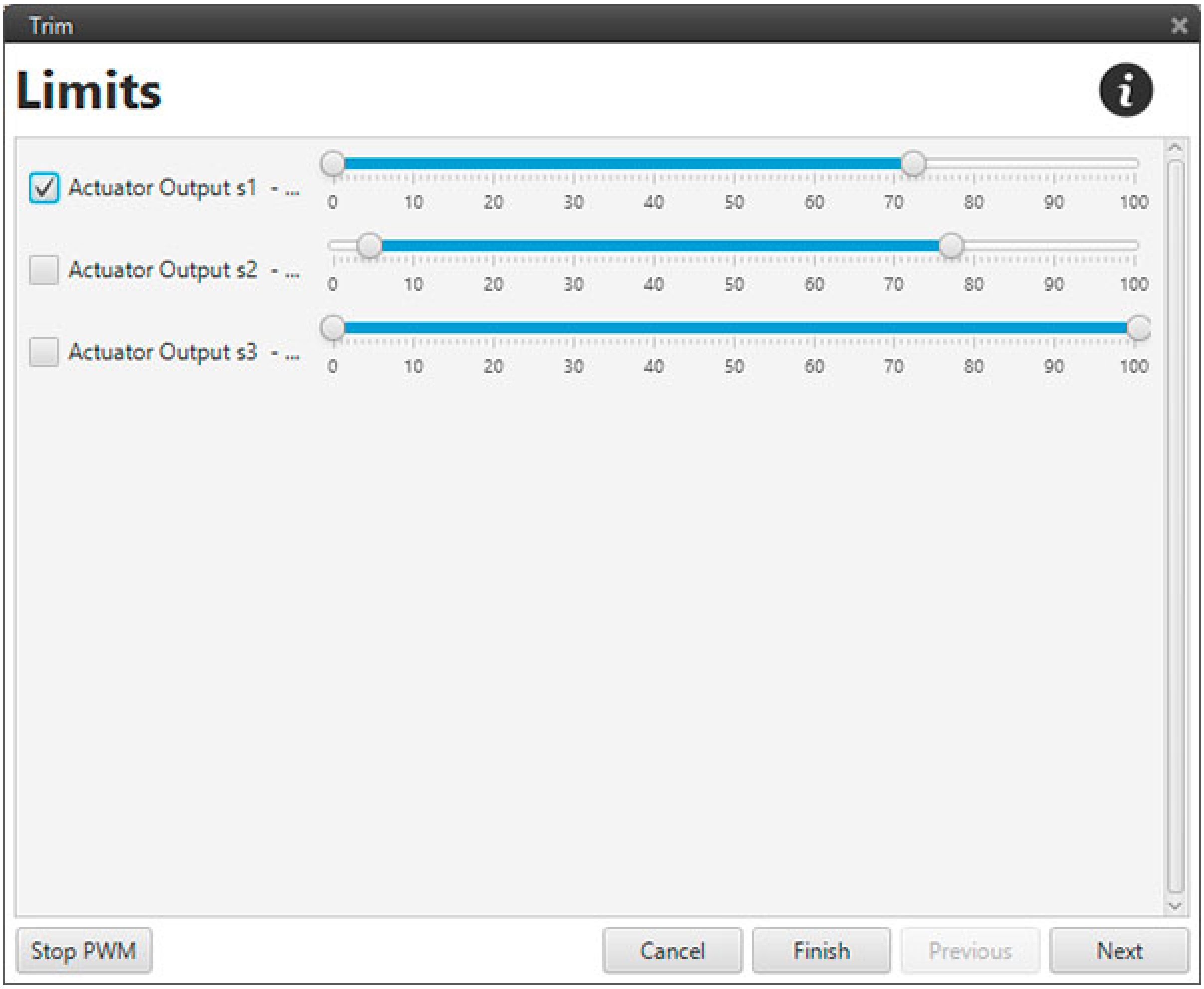
Actuator 1 Trimming

The picture above shows the setting of the elevon number 1:

- **Minimum:** -0.34906 [rad] deflection; 4.66% PWM output.
- **Zero position:** 0.0[rad] deflection; 46.73% PWM output.
- **Maximum:** 0.34906 [rad] deflection; 76.89% PWM output.

Warning: The actuators can be moved directly from Veronte Pipe only when the system is in an Initial phase (when there is no phase selected in Veronte Panel). During the actuator run, if the desired position is in the **Out of range** zone (red zone), it is possible to click on **Allow command out of limits** in order to move completely the actuator and find the correct position.

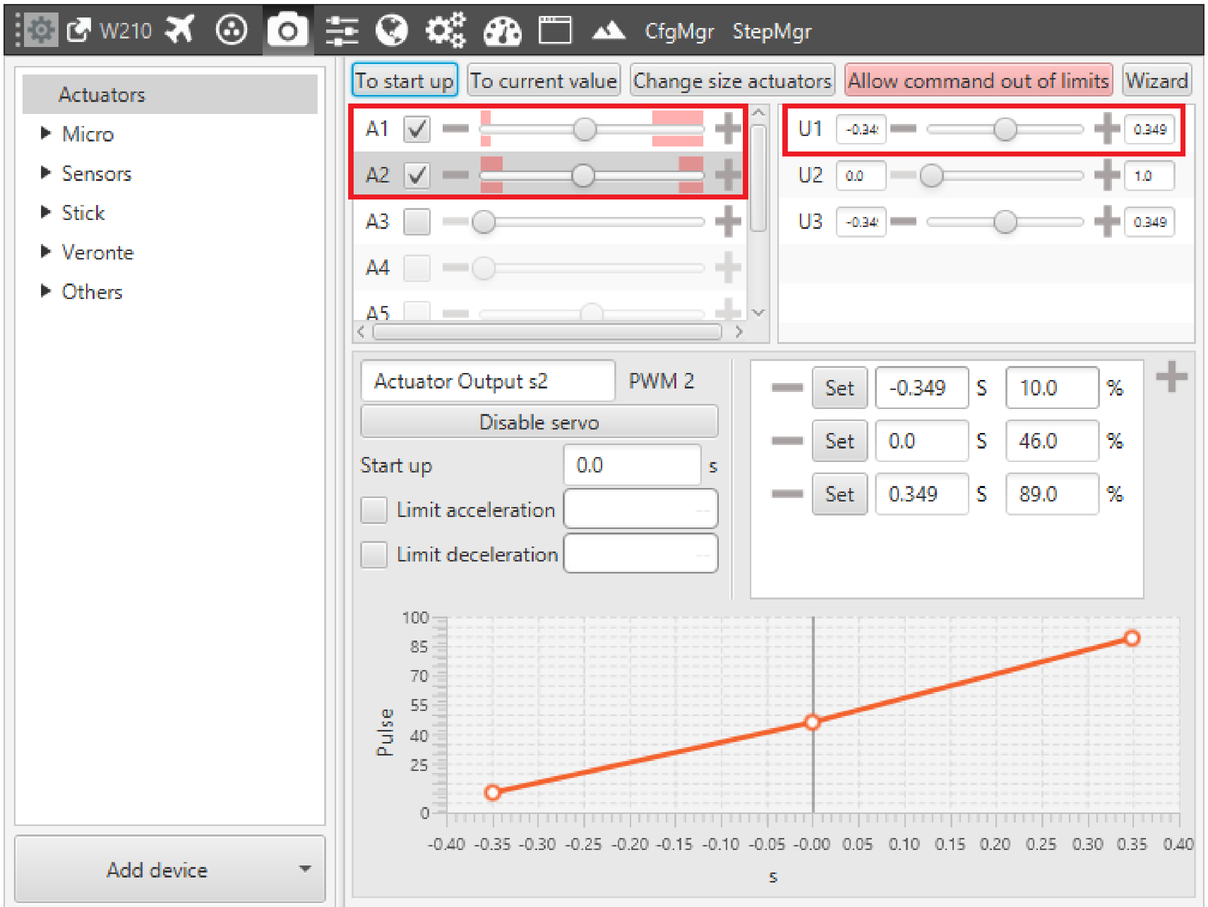
This procedure can be performed in the same way by using the **Wizard**. This tool allows moving actuators limits easily and finding the correct range, as shonw below.



Trim wizard tool

In order to perform a final check, it is possible to select the desired channel and test pitch, roll and thrust control.

The image below shows a pitching output testing. By moving the U1 control, surfaces must change the position according to the reference system: positive corresponds to nose down and negative to nose up.



Pitching test

For more information about the physical calibration of the servos check *Physical*.

1.5 External Transmitter Configuration

This section includes the information of how to connect an external PPM transmitter to control the platform fitted with the autopilot.

To use the joystick in the system, connect the PPMout of the trainer port to a digital input of Veronte and configure that digital input as the radio input in Pipe.

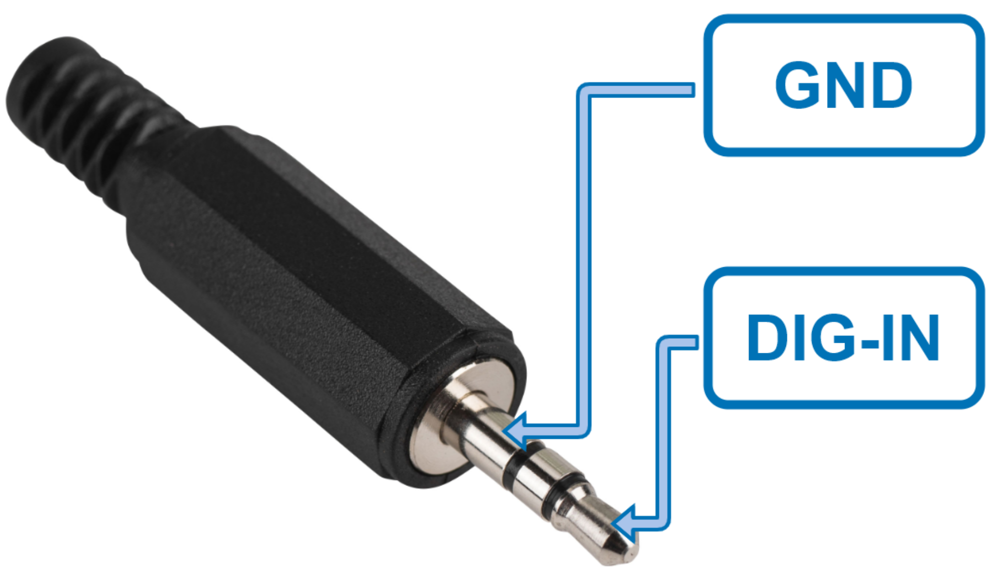
If the PPM level is 3.3V, pins 1-8, 10-17 and 55-58 pins can be used.

Veronte is compatible with standard Pulse Positon Modulation (PPM) signals between 8 and 16 channels.

Jeti DC-16 Transmitter

> **Warning: Caution!!** PPM signal must be into the Veronte voltage ranges. Some joysticks may need an adaptation board, please ask our team to check compatibility.

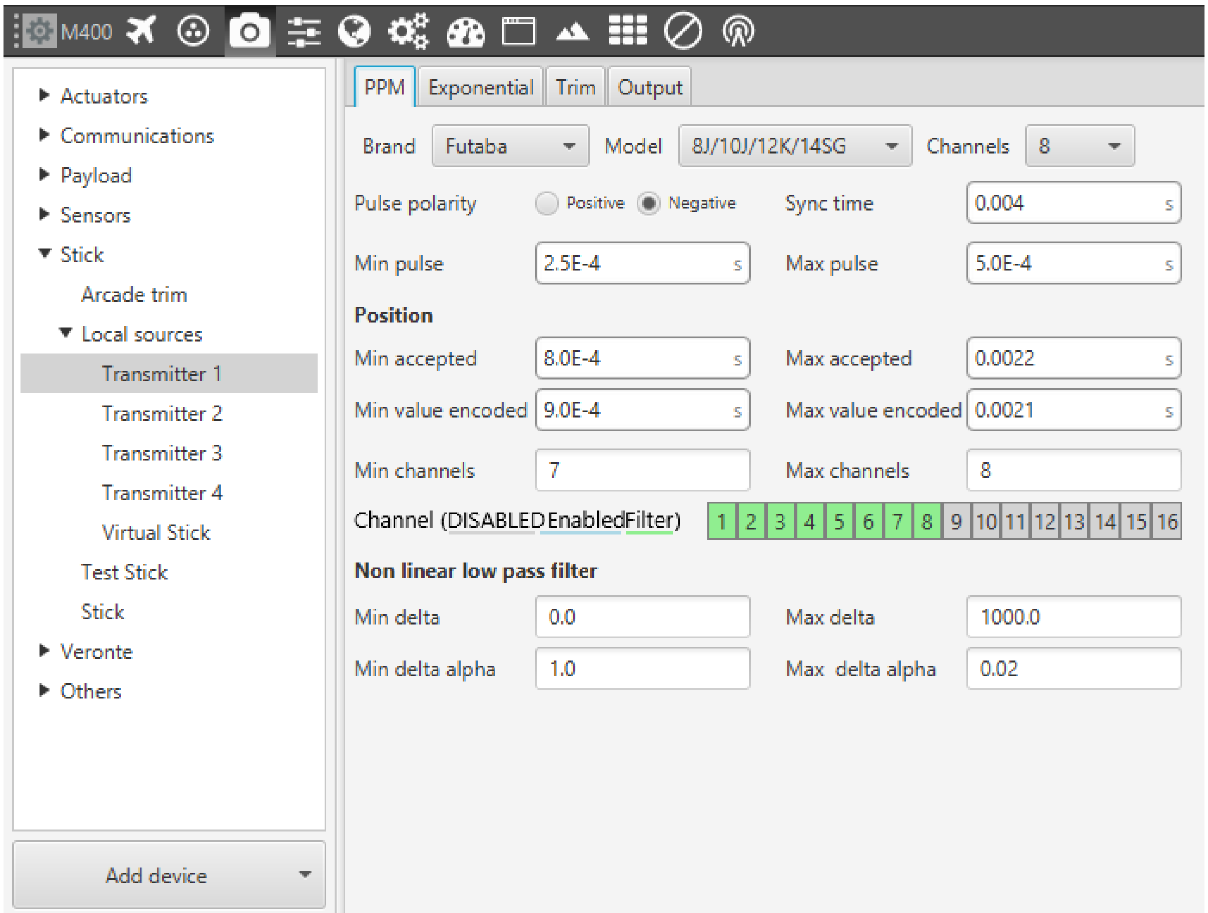
Veronte connector for CS is provided with 3.5mm stereo plug connector as follows:



Male Jack connector (3.5mm) pinout - CS Harness

In the side panel , click on Setup, go to Devices , the open the Stick tab, and finally go to the Local sources tab. A menu opens up with 4 tabs on top: PPM, Exponential, Trim and Output.

1.5.1 PPM



Stick Transmitter - PPM Configuration Parameters

Veronte Pipe can help the user by configuring the PPM panel in order to read the PPM output from the most famous transmitter brands:

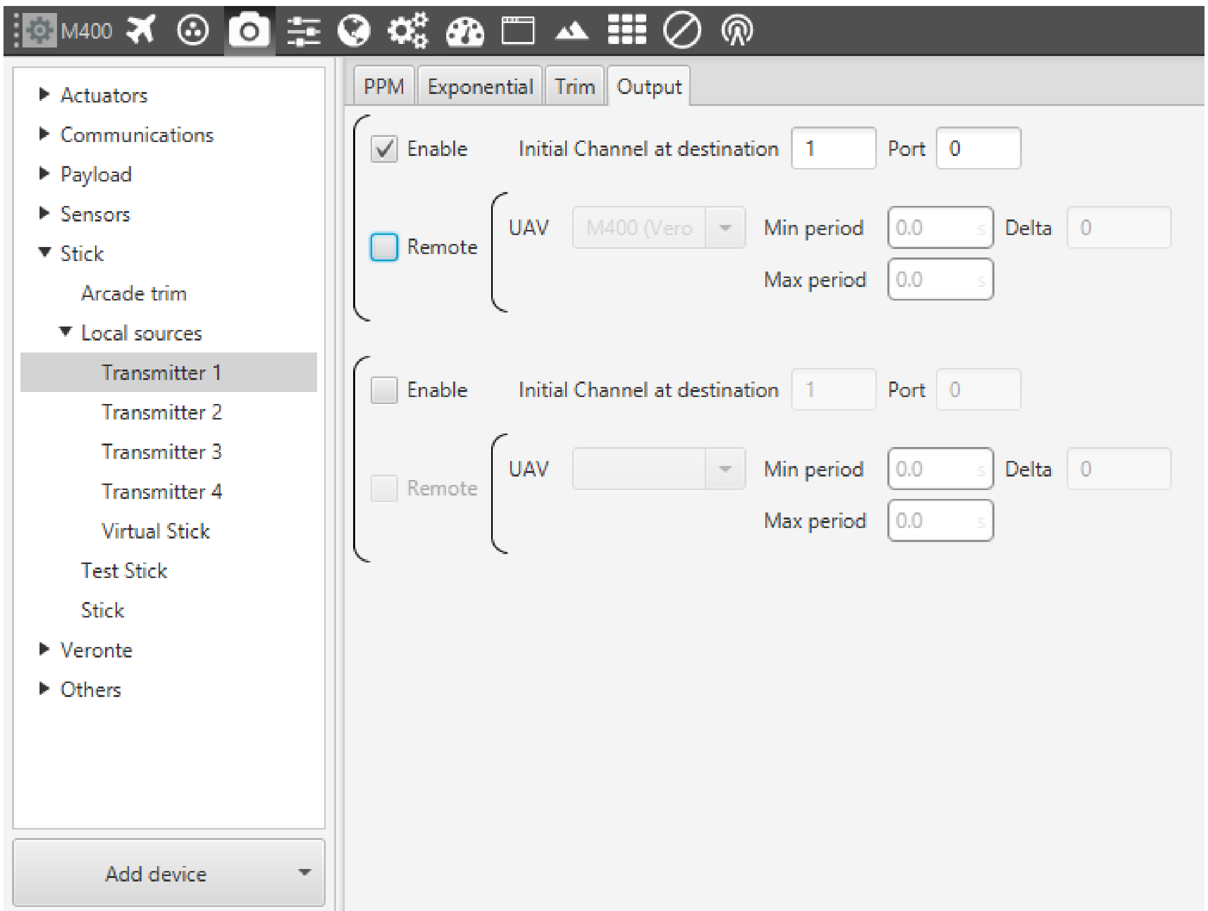
- **Brand, Model & Channels**:
  - Futaba.
    - Model 8J/10J/12K/14SG with 8 channels.
    - Model 12K/14SG with 12 channels.
    - Model T18SZ with 8 channels.
  - Jeti.
    - DC 16 / DC 24 with 16 channels.

- FrSky.
  - Taranis 9XD with 8 channels.
  - X12S with 8 channels.

If the User's transmitter is not present in the previous table, it is also possible to configure the PPM parameters in order to match with the transmitter's PPM output. Please check the Stick Configuration section for more information. In the same section, it's possible to find further information about Exponential and Trim configurations.

1.5.2 Output

Once the stick has been configured, the commands that arrive at the ground autopilot have to be sent to the air one.



Stick Transmitter - Output

Toggling the option *Enable*, the user indicates at which channel of the **AIR** autopilot will be sent the first channel received in the **GND** autopilot. The channels arrive at the platform by order and without spaces between them i.e, if the external transmitter is sending the **GND** the channels 1,2,3,4 and 6, then the **AIR** autopilot will arrive channels 1,2,3,4 and 5, where channel 5 of the **AIR** will actually be channel 6 of the external transmitter.

The alternate option *Remote* permits the delivery of the commands to the platform by indicating the address on UAV (when *Broadcast* is selected, the commands are sent to all the air autopilots linked to the ground one).

1.6 Pre-flight Checklist

This section provides the necessary information on how to implement/prepare a pre-flight checklist. The latter will be sensitive to every vehicle and its mission. It is fundamental for the user to know the set of sensors and/or events that must be evaluated before allowing the vehicle to enter a ready-to-fly phase.

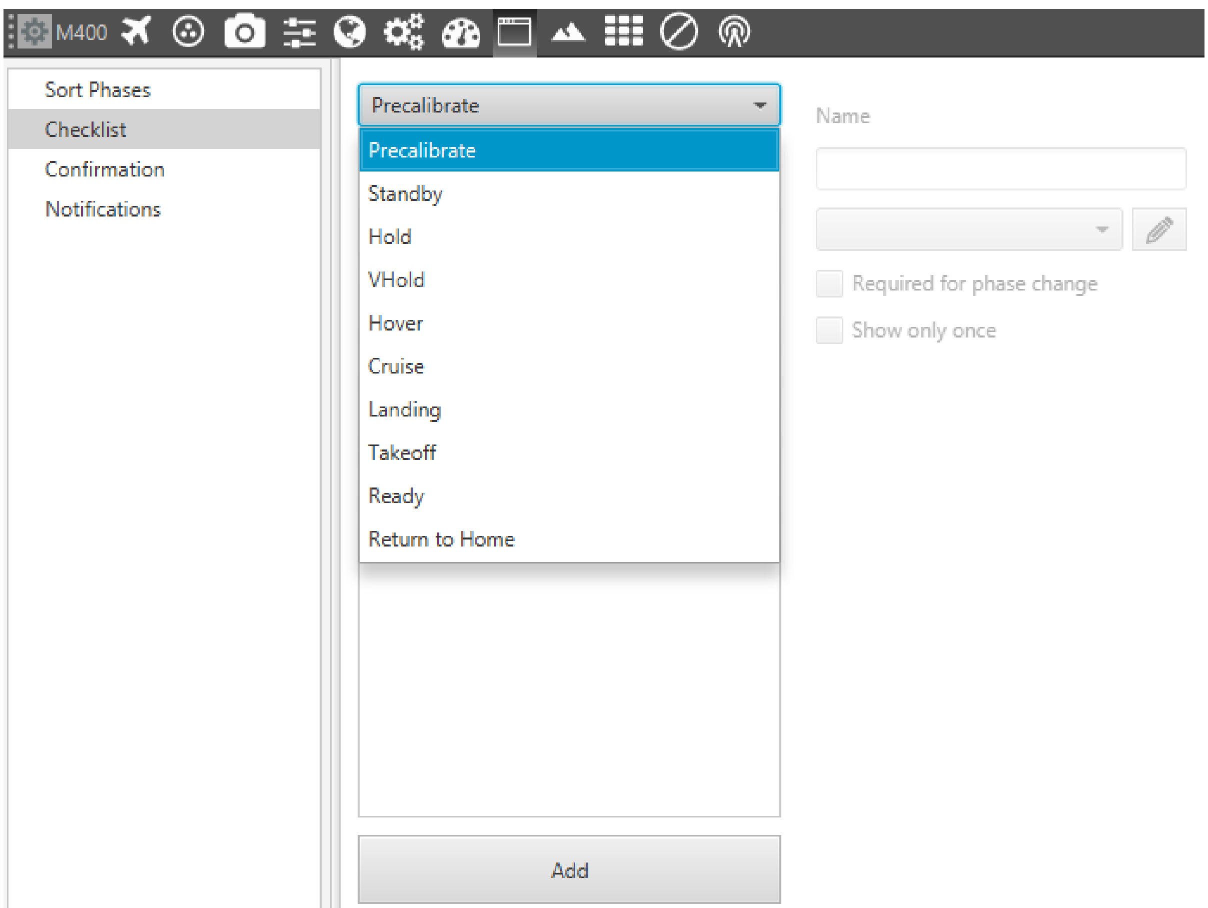
However, there is a set of critical parameters that should always be on the user's checklist. There are others worth-considering too. Both of them will be introduced hereafter with detail on the implementation in Veronte Pipe.

1.6.1 Main Checklist

Within the software, it is possible to define a checklist for every phase created. Such checklist can be tailored so that if one item is not satisfactorily fulfiled upon phase changing, the vehicle won't be able to swap to the desired/programmed phase.

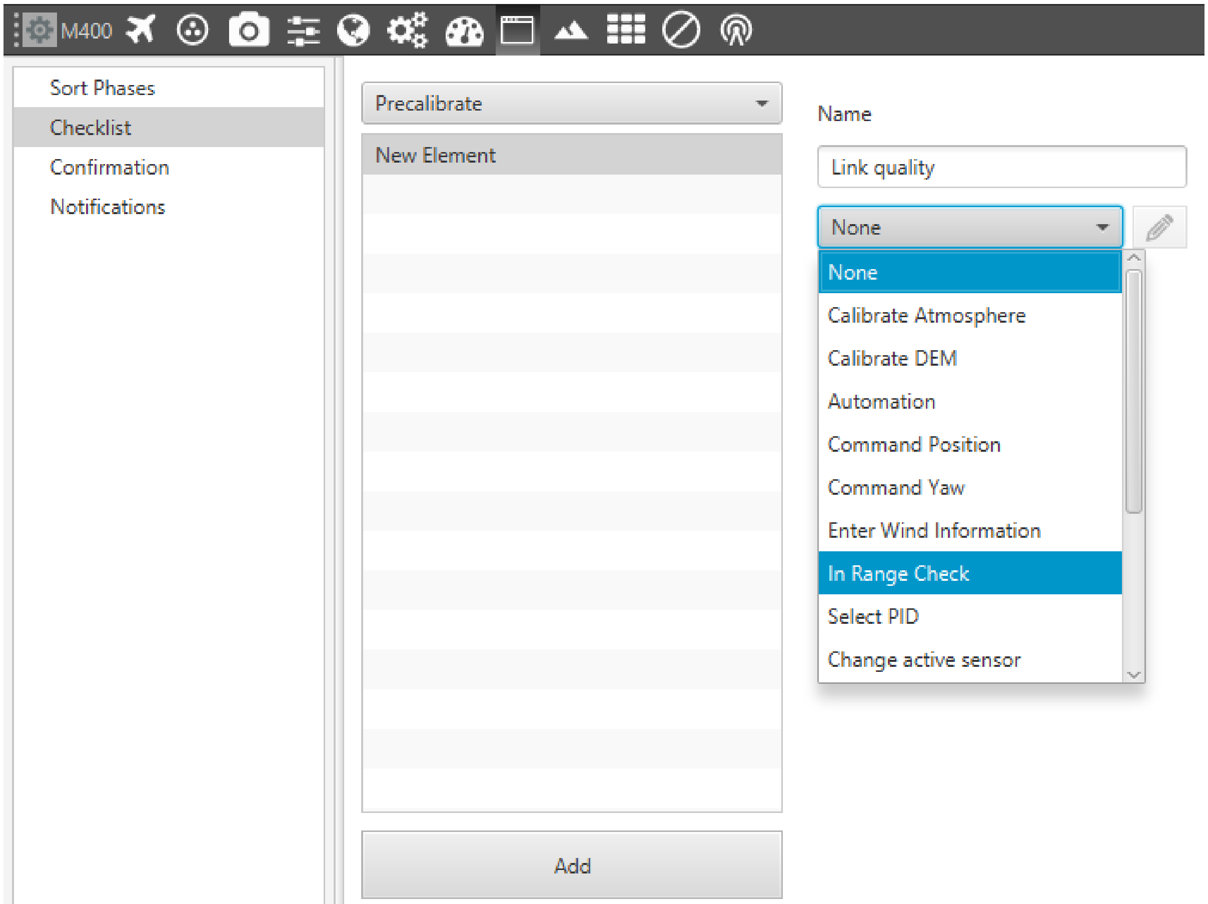
These checklist are also helpful for the user to check relevant vairables when a phase change occurs. But for what it is concerned in this section, a checklist needs to be defined on the Precalibrate phase, i.e. the initial default phase of the autopilot.

Go to , then click on Setup. On the Setup toolbar then go to Panel . And finally click on Checklist. There the user should select the Precalibrate phase.



Panel Checklist – Precalibrate Phase

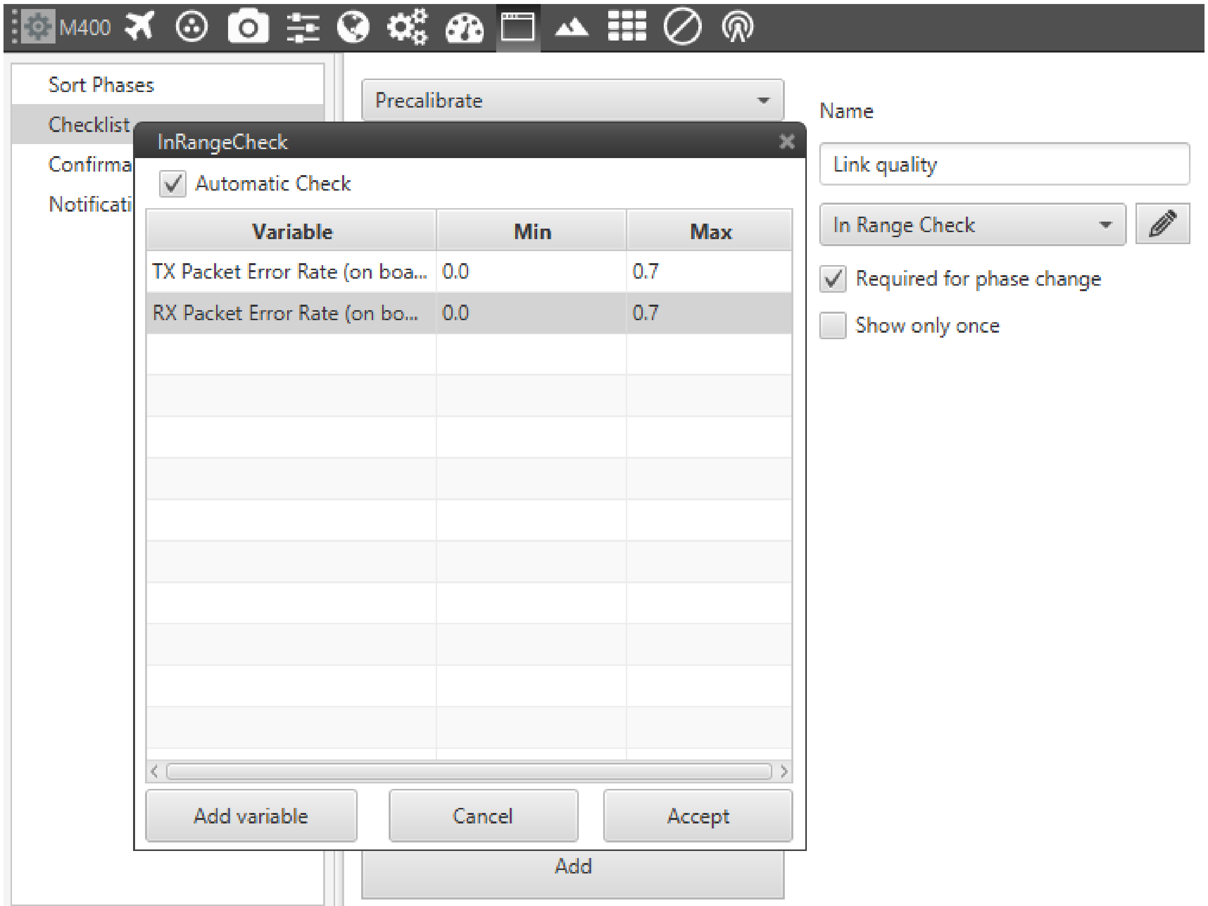
When creating a new element the user can select among many options: from commanding a certain action (position, yaw, etc), calibrating a sensor, to check that a variable or group of variables are within a range of values (see figure below). Then there are also the options "Required for phase change" and "Show only once", which are self-explanatory. "Required for phase change" should be selected for every item as the user does not want to start the flight without this checklist completely verified.



Options for New Element

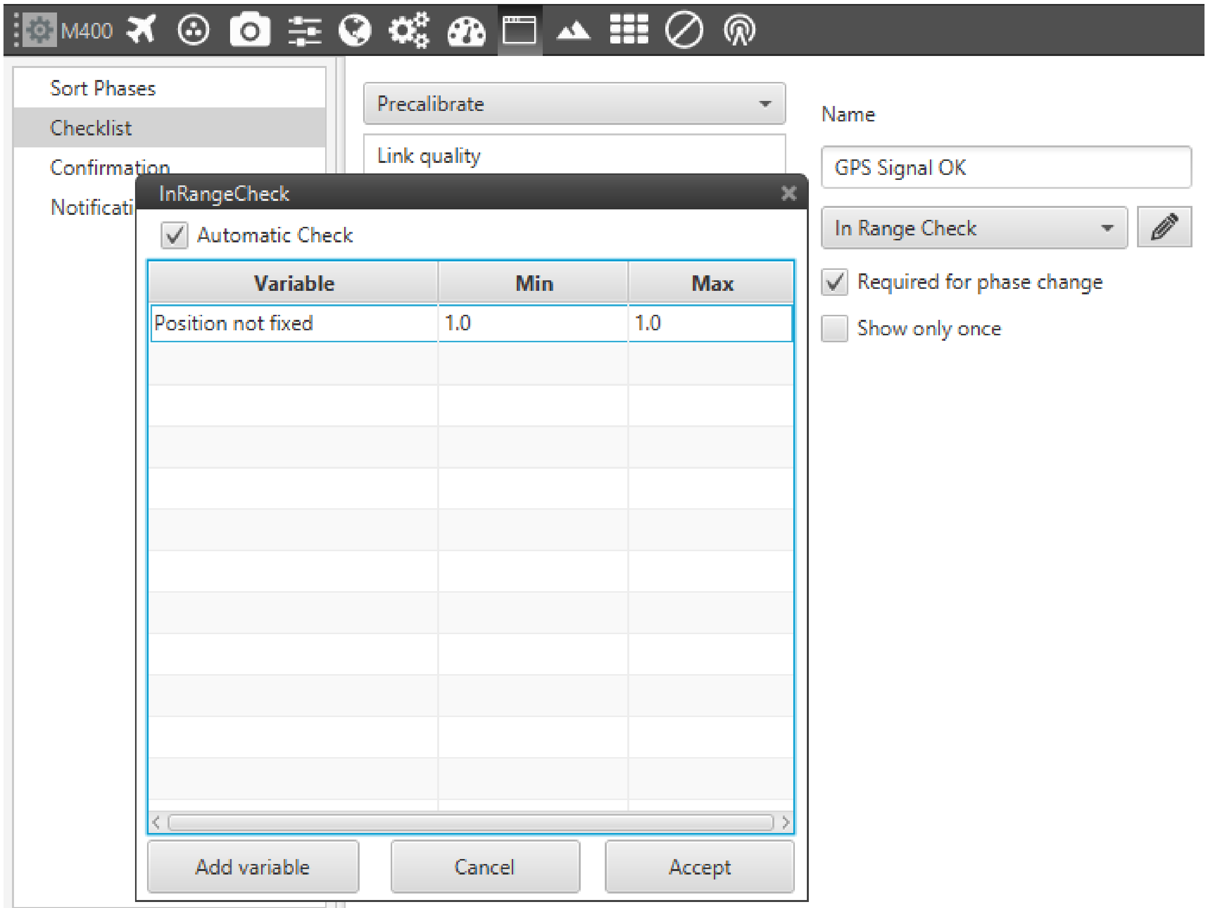
The main checklist is composed of:

- **Link quality**: radio connection from the **GND** unit to the **AIR** unit can be evaluated with 2 built-in variables. They measure the transmitting and receiving error rate, i.e. the percentage of messages that are not received or transmitted correctly. What is set is that this variables' values shoud not be bigger than 0.7 (option "In Range Check") and the option "Automatic Check" should be toggled so this is reviewed automatically (see the figure below).



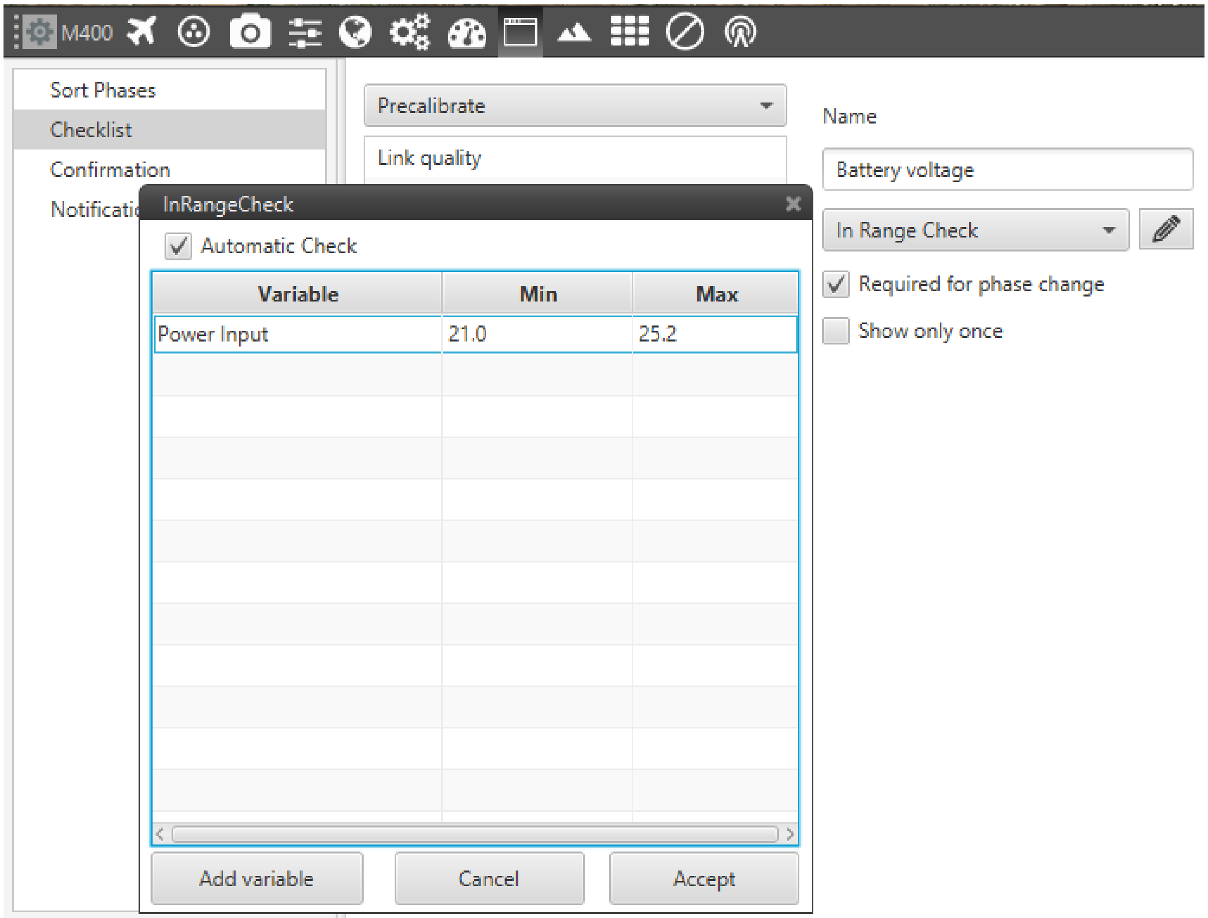
Link Quality Parameters

- **GPS signal OK**: if the GPS module is not working correctly there is a boolean variable called “Position not fixed” that will be turned to 0. Therefore option “In Range Check” is selected with a value between 1 and 1, i.e. the boolean variable should be *true*. Option “Automatic Check” should also be toggled.



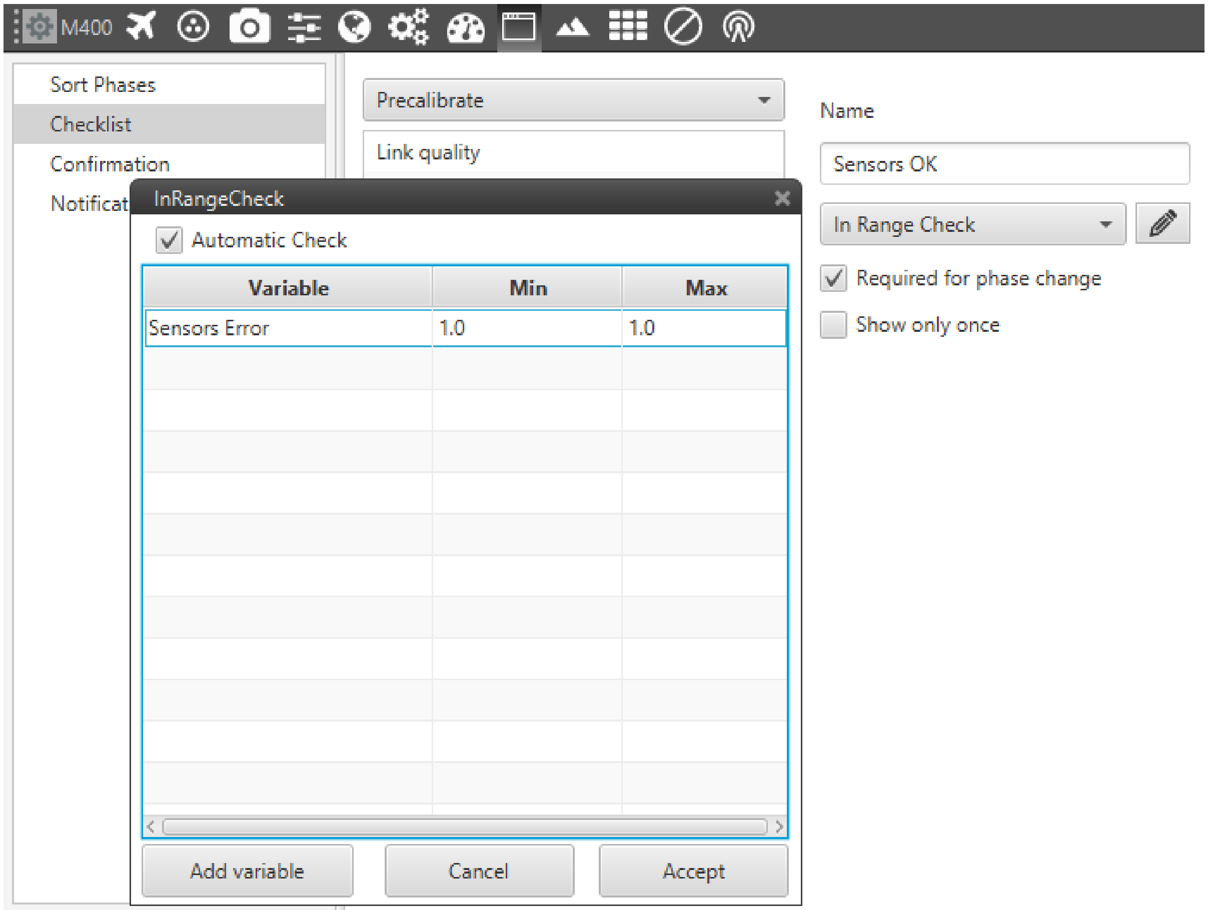
GPS Signal Parameters

- **Battery voltage / fuel volume**: a system variable shoud be used to monitor the voltage of the battery of the amount of fuel left on the tank of the vehicle. Then, option "In Range Check" should be selected setting the range of possible values, e.g. 21 V and 25.2 V for 6 Cell LiPo Battery. Option "Automatic Check" should also be toggled.



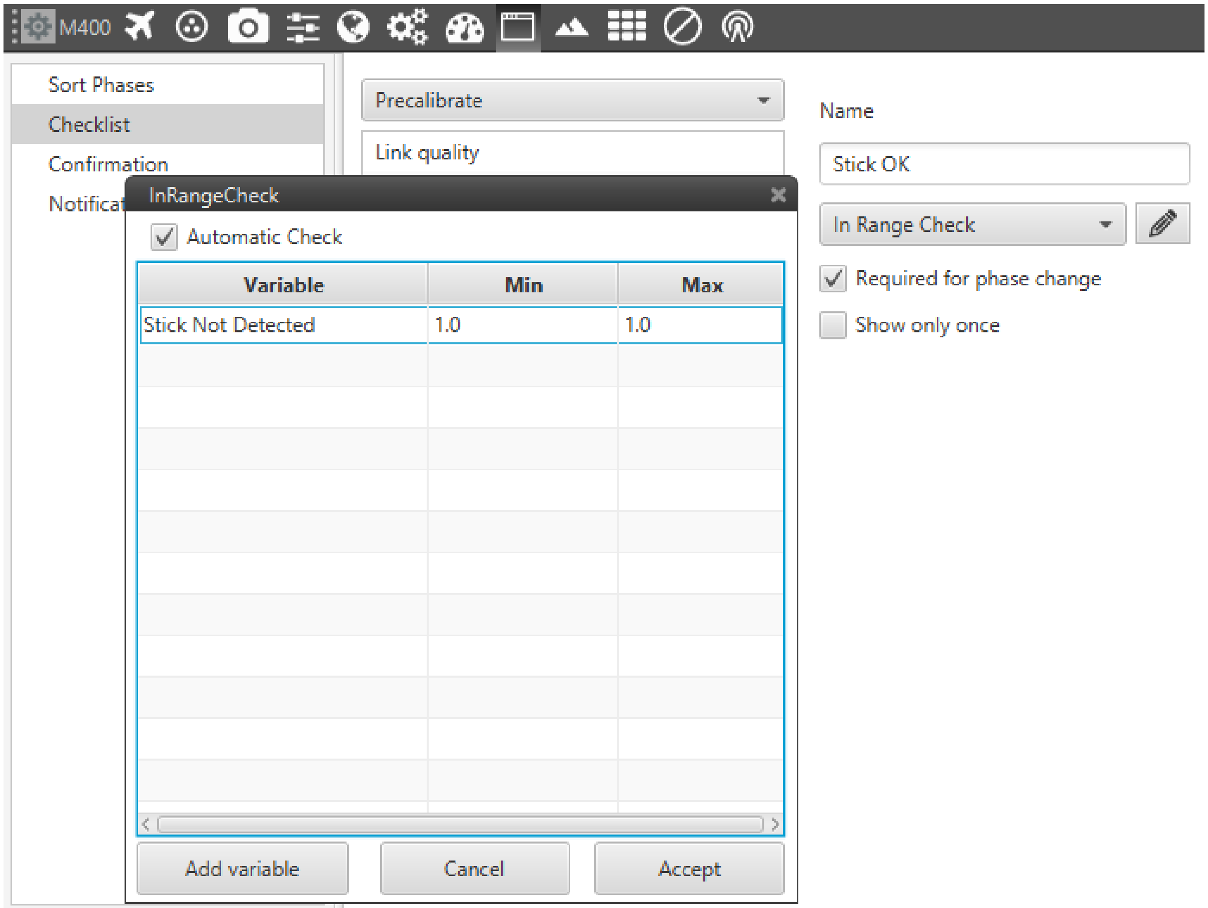
Battery Voltage Parameters

- **Sensors OK**: there is a boolean variable that checks all the autopilot's avialable sensors called "Sensors Error". If any of the sensors is not working then this variable will be 0. Therefore option "In Range Check" is selected with a value between 1 and 1, i.e. the boolean variable should be *true*. Option "Automatic Check" should also be toggled.



Sensors Checkup Parameters

- **Stick OK**: there is a boolean variable that checks that there is an external transmitter connected to the **GND** autopilot. The latter is essential so the mission can be carried in "Manual mode" if ever necessary. The variable is called "Stick not detected" and if there's no external transmitter connected it will be 0. Therefore option "In Range Check" is selected with a value between 1 and 1, i.e. the boolean variable should be *true*. Option "Automatic Check" should also be toggled.



External Transmitter Parametters

1.6.2 Other Items

Other parameters worth-considering to be on the user's checklist:

- **Terrain Mesh OK**
- **Mission OK**
- **Automations OK**

This quick start guide provides the necessary information to have Veronte Autopilot set up and functional to operate its mission.

Veronte 4.5

This guide covers the following points:

- **First steps:** veronte autopilot connections and software installation.
- **HIL simulation:** how to perform a simulation with an existing configuration using veronte autopilot.
- **Internal radio module pairing:** how to pair of the ground segment (**GND**) and onboard (**AIR**) autopilots.
- **Outputs configuration:** how to calibrate the sevos for a good control of the surfaces/devices that determine the attitude of the platform.
- **Transmitter:** how to set up an external transmitter to control the vehicle when necessary.
- **Pre-flight checklist:** set of sensors, communications, and other events that should be verified before starting the mission.

CHAPTER

TWO

HARDWARE INSTALLATION

2.1 Veronte Autopilot

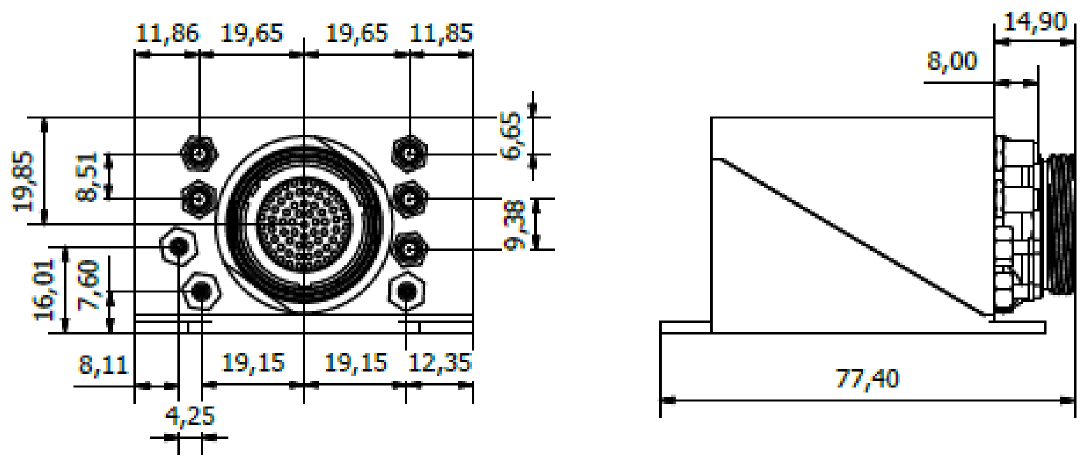
2.1.1 Aircraft Mounting

There are two versions of Veronte autopilot: with or without enclosure.

2.1.1.1 Enclosure

Veronte is provided using an anodized aluminium enclosure with enhanced EMI shielding and IP protection. A high reliability connector is also provided in this version. The total approximate weight is 190 g.

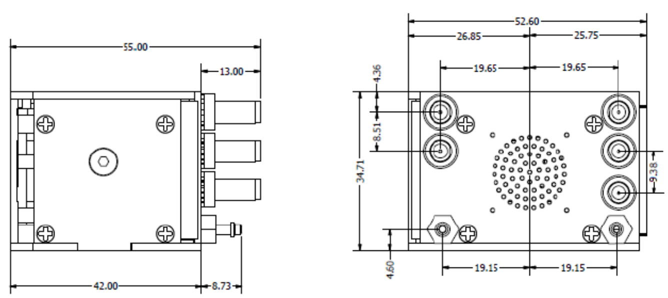
Veronte with aluminium enclosure



Veronte dimensions

2.1.1.2 OEM

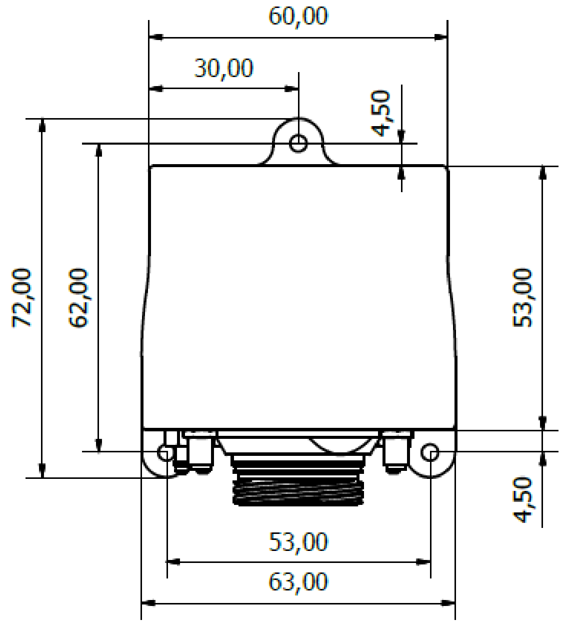
Veronte can be provided in OEM version too.The total approximate weight is 90g.



OEM dimensions

2.1.1.3 Mechanical Mounting

M3 screws are recommended for mounting. In saline environments such as coastal and oceanic, the screw material must be stainless steel.

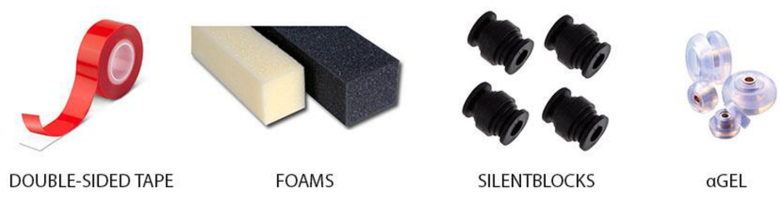


Veronte Top View

2.1.1.4 Vibration Isolation

Although Veronte ultimately rejects noise and high-frequency modes of vibration with electronic filters and internal mechanical filters, there might be situation where external isolation components might be needed.

Veronte can be mounted in different ways in order to reject the airframe vibration. The simplest could be achieved by just using double-sided tape on the bottom side of Veronte. Other ways may use some external structure which could be rigidly attached to the airframe and softly attached to Veronte (e.g. foam, silent blocks, gel, etc)



Veronte mounts

The user should take into account that wiring should be loose enough so vibrations may not be transmitted to Veronte.

In cases where Veronte isolation is not viable, it is possible to use soft engine mounts. It is also recommended when there are other sensible payloads like video cameras or for high vibration engines.

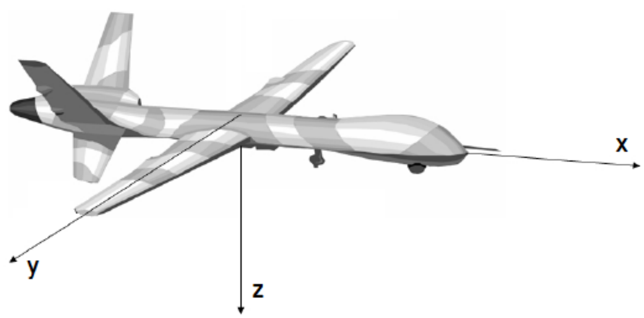
2.1.1.5 Location

The location of Veronte has no restrictions. You only need to configure its relative position with respect to the centre of mass of the aircraft and the GNSS antenna. The configuration of the location of Veronte can be easily configured using Veronte Pipe Software.

2.1.1.6 Orientation

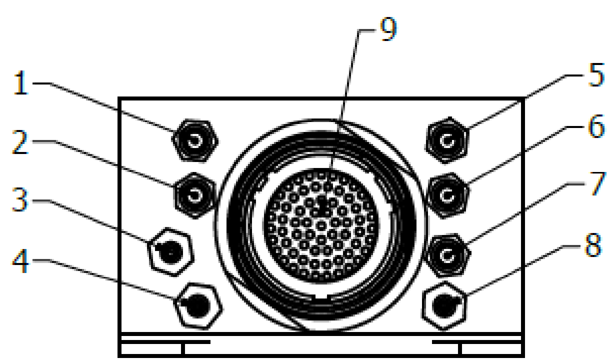
The orientation of Veronte has no restrictions either. You only need to configure Veronte axes with respect to the aircraft body axes by means of a rotation matrix or a set of correspondences between axes. The configuration of the orientation of Veronte can be easily configured using Veronte Pipe Software.

Veronte axes are printed on the box and aircraft coordinates are defined by the standard aeronautical conventions.



Aircraft Mounting

2.1.1.7 Connector Layout



Veronte Connectors

Index	Connector
1	LOS SSMA connector
2	GNSS1 SSMA connector
3	Static pressure port (Fitting 5/64in)
4	Static pressure port (Fitting 5/64in)
5	M2M SSMA connector
6	GNSS2 SSMA connector
7	TPDR (Transponder) SSMA connector
8	Dynamic pressure port (Fitting 5/64in)
9	68-pin connector

For the pressure ports, mating with clamped 2mm internal diameter flexible tubing is recommended.

The two static pressure ports must be used for sensor redundancy.

2.1.1.8 Mating Connector

Index	Connector	Mating Connector
1	RF antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
2,6	GNSS antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.Active Antenna GNSS: Gain min 15dB (to compensate signal loss in RF Cable) max 50dB, maximum noise figure 1.5dB, power supply 3.3V max current 20 mA
5	M2M antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
7	TPDR antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
9	Connector HEW.LM.368.XLNP	Mating connector P/N: FGW.LM.368.XLCT Mating harness is available on demand.

2.1.1.9 Antenna Integration

The system uses different kinds of antennas to operate that must be installed on the airframe. Here you can find some advice for obtaining the best performance and for avoiding antenna interferences.

Antenna Installation
Maximize separation between antennas as much as possible.
Keep them far away from alternators or other interference generators.
Always isolate antenna ground panel from the aircraft structure.
Make sure the antenna is securely mounted.
Always use high-quality RF wires minimising the wire length.
Always follow the antenna manufacturer manual.
SSMA connections shall be tightened applying 1Nm of torque
For all-weather aircraft, insert SSMA lightning protectors.

GNSS Antenna
Antenna top side must point the sky.
Install it on a top surface with direct sky view.
Never place metallic / carbon parts or wires above the antenna.
It is recommended to install it on a small ground plane.
For all-weather aircraft, insert SSMA lightning protectors.

2.1.1.10 Pressure lines

Veronte has two pressure input lines, one for static pressure to determine the absolute pressure and one for pitot in order to determine the dynamic pressure.

Absolute pressure connection on the aircraft is mandatory while pitot port can be obviated in some aircraft, pitot port absence must be configured on Veronte Pipe software.

Pressure Intake
Pressure intakes must be located in order to prevent clogging.
Never install pressure intakes on the propeller flow.
Design pressure tubing path in order to avoid tube constriction.

Static Pressure
It is not recommended to use inside fuselage pressure if it is not properly vented.

Pitot Tube
Pitot tube must be installed facing the airflow in the direction of the “x” axis of the aircraft.
It is recommended to install it near the aircraft axis in order to avoid false measures during manoeuvres.
For low-speed aircraft, it is recommended at least 6,3mm tubes for preventing rain obstruction.

2.1.2 Electrical

2.1.2.1 Power

Veronte can use unregulated DC (**6.5V to 36V**). Pins used for power and ground are the same for both Ground and Air configurations.

LiPo batteries between 2S and 8S can be used without regulation needs. Remaining battery level can be controlled by the internal voltage sensor and by configuring the voltage warnings on the Pipe software.

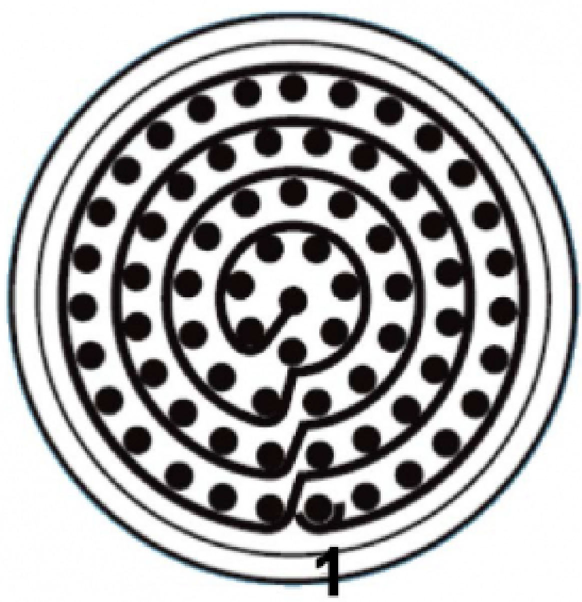
For higher voltage installations, voltage regulators must be used. For dimensioning voltage regulators take into account that a blocked servo can activate regulator thermal protection.

Warning: Caution!! Power Veronte out of the given range can cause irreversible damage to the system. Please read carefully the manual before powering the system.

Veronte and servos can be powered by the same or different batteries. In case of having more than one battery on the system, a single point ground union is needed to ensure a good performance. The ground signal should be isolated from other noisy ground references (e.g. engines). If all grounds need to be connected, the connection should be made on the negative pole of the battery.

It is recommendable to use independent switches for autopilot and motor/actuators. During the system initialization, the PWM signal will be set to low level (0V), please make sure that actuators/motor connected support this behaviour before installing a single switch for the whole system.

2.1.2.2 Veronte I/O Signals



68 pin connector for Veronte Autopilot (frontal view)

Pin	Signal	Type	Comments
1	I/O1	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
2	I/O2	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
3	I/O3	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
4	I/O4	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
5	I/O5	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
6	I/O6	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
7	I/O7	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
8	I/O8	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
9	GND	GROUND	Ground signal for actuators 1-8
10	I/O9	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
11	I/O10	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
12	I/O11	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
13	I/O12	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
14	I/O13	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
15	I/O14	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
16	I/O15	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
17	I/O16	I/O	PWM/DIGITAL output /DIGITAL input signal (0-3.3V)
18	GND	GROUND	Ground signal for actuators 9-16
19	RS 232 TX	Output	RS 232 Output
20	RS 232 RX	Input	RS 232 Input

Pin	Signal	Type	Comments
21	GND	GROUND	Ground signal for buses
22	Analog 4	Input Analog	Input 0-3V
23	Analog 5	Input Analog	Input 0-3V
24	GND	GROUND	Ground signal for buses
25	CanA P	I/O	CANbus interface. It supports data rates up to 1 Mbps
26	CanA N	I/O	Twisted pair with a 120 Zo recommended
27	GND	GROUND	Ground signal for buses
28	CANB_P	I/O	CANbus interface. It supports data rates up to 1 Mbps
29	CANB_N	I/O	Twisted pair with a 120 Zo recommended
30	GND	GROUND	Ground signal for buses
31	I2C_CLK	Output	Clk line for I2C bus
32	I2C_DATA	I/O	Data line for I2C bus
33	GND	GROUND	Ground for 3.3 V power supply
34	3.3V	POWER	3.3 V-100 mA power supply
35	GND	GROUND	Ground for 5 V power supply
36	5V	POWER	5 V – 100 mA power supply
37	GND	GROUND	Ground for analog signals
38	ANALOG_1	Input	Analog input 0-3 V
39	ANALOG_2	Input	Analog input 0-3 V
40	ANALOG_3	Input	Analog input 0-3 V
41	GND	GROUND	Ground for FTS signals
42	FTS_OUT	Output	Deadman (i.e. Heart-Beat, Alive…) signal from comicro, monitors main MPU encoding
43	FTS2_OUT	Output	Output is a !SystemOK (0 when Ok, VCC when in Error) Bit Low (0 volts) if no failure d
44	GND	GROUND	Ground signal for safety buses
45	V_ARB_TX	Output	Veronte comicro UART output to activate safety mechanism
46	V_ARB_RX	Input	Veronte comicro UART output to activate safety mechanism
47	GND	GROUND	Ground signal comicro power supply
48	V_ARB_VCC	POWER	Veronte comicro power (6.5 to 36 V)
49	FTS3_OUT_MPU	Output	MPU alive voting signal, to use with 4xVeronte. It is a Square Wave at [100,125] Hz
50	OUT_RS485_P	Output	Non-inverted output from RS485 bus
51	OUT_RS485_N	Output	Inverted output from RS485 bus
52	IN_RS845_N	Input	Inverted input from RS485 bus
53	IN_RS845_P	Input	Non-inverted output from RS485 bus
54	RS-485_GND	GND	Ground for RS-485 bus
55	EQEP_A	Input	Encoder quadrature input A (0-3.3 V)
56	EQEP_B	Input	Encoder quadrature input B (0-3.3 V)
57	EQEP_S	Input	Encoder strobe input (0-3.3 V)
58	EQEP_I	Input	Encoder index input A (0-3.3 V)
59	GND	GROUND	Ground for encoders
60	V_USB_DP	I/O	Veronte USB data line
61	V_USB_DN	I/O	Veronte USB data line
62	V_USB_ID	I/O	Veronte USB ID line
63	FTS_OUT_MPU	Output	Abort mission voting signal from MPU, to use with 4xVeronte. Bit Low (0 volts) if missi
64	FTS2_OUT_MPU	Output	Abort mission voting signal 2 from MPU, to use with 4xVeronte. Bit Low (0 volts) if mis
65	GND	GROUND	Veronte ground input
66	ND	GROUND	Veronte ground input
67	VCC	POWER	Veronte power supply (6.5 to 36 V)
68	VCC	POWER	Veronte power supply (6.5 to 36 V)

2.1.2.3 Flight Termination System (FTS)

In order to use the FTS of Veronte the user has to analyze Pins 42 and 43 (FTS ones). With pin 43 it can be studied with a !SystemOK bit if the system is working properly or not. Check below the information provided by each pin:

Pin 42: Deadman signal (i.e. Heart-Beat, Alive…) encoding its product-level bit. Deadman signal is a Square Wave at [100,125] Hz. It can be higher at rebooting (about 300-400Hz) but should never be less than 100 Hz.

Pin 43: Outputs a !SystemOK (0 when Ok, VCC when in Error) Bit. Low (0 volts) if no failure detected. High (3.3 volts) on error detected. This pin goes high if the deadman signal sent by the MPU (main processor unit) is lower than 63Hz. That means there is a critical error.

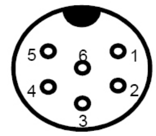
2.1.2.4 Joystick

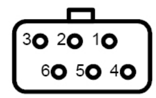
To use the joystick in the system, connect the PPMout of the trainer port to a digital input of Veronte and configure that digital input as the radio input in Pipe.

If the PPM level is 3.3V, pins 1-8, 10-17 and 55-58 pins can be used.

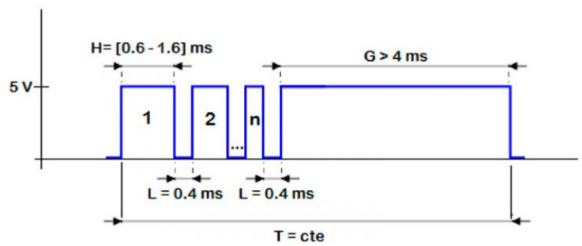
Veronte is compatible with standard Pulse Positon Modulation (PPM) signals, Futaba radios between 8 and 12 channels are recommended.

Futaba T10 Joystick

Pin	Designation	Connector
SHIELD	GROUND	
1	$V_{ENCODER}$	
2	PPM_{OUT}	
3	PPM_{IN}	
4	V_{ENC2}	
5	$V_{BATTERY}$	
6	UNKNOWN	

Pin	Designation	Connector
1	NC	
2	GROUND	
3	PPM_{OUT}	
4	$V_{BATTERY}$	
5	$V_{ENCODER}$	
6	PPM_{IN}	

Futaba T10 pinout

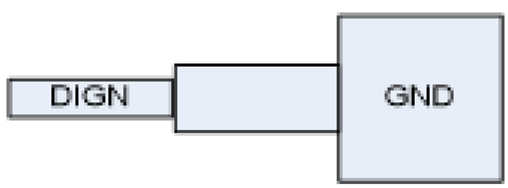


PPM signal

As default, channel 8 is reserved for manual / auto switch. High level is used for automatic flight and low level for manual control. This channel can be configured on Veronte Pipe.

> **Warning: Caution!!** PPM signal must be into the Veronte voltage ranges. Some joysticks may need an adaptation board, please ask our team to check compatibility.

Veronte connector for CS is provided with 3.5mm stereo plug connector as follows:



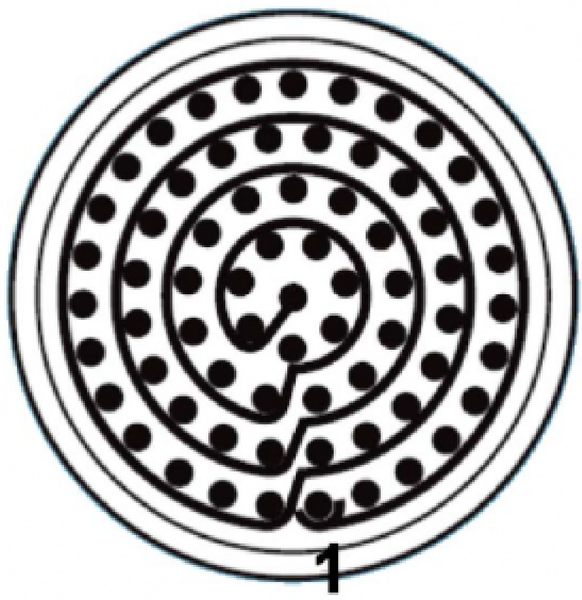
PPM pinout

PPM connector

2.1.3 Performances

Variable	Value
Weight (with enclosure and connector)	190 g
Weight (OEM)	90 g
Power Input	6.5 V to 36 V
Minimum Temperature	-40 ºC
Maximum Temperature	+55ºC (No convection, ask for increased limits (up to 71ºC))
Max. Internal Temperature	+85ºC
Minimum Pressure	0 kPa
Maximum Pressure	104 kPa
Maximum Dynamic pressure	6 kPa (Ask for increased limits (up to 50kPa))
Protection Rating	IP67 enclosure version
Acceleration Limits (3 axes)	±2 g to ±16 g (for sustained maneuvers, transitional higher accelerations are possible (e.g. catapult launch). Ask for increased limits.)
Angular Velocity Limits (3 axes)	±125 deg/s to ± 2000 deg/s (for sustained maneuvers, transitional higher angular velocities are possible. Ask for increased limits.)
Magnetic Field Limits (3 axes)	±4 to ±16 Gauss
GNSS	72 channels, GPS L1C/A, GLONASS L1OF, BeiDou B1I
Datalink	410 to 480 MHz licensed or FHSS/902-928MHz FHSS/2.4 to 2.483 GHz ISM Band/869.5-869.75 MHz ISM Band

2.1.4 Annex 1: Connector colour code



Connector HEW.LM.368.XLNP

PIN	Color code	PIN	Color code
1	White	35	Gray
2	Brown	36	Pink
3	Green	37	Blue
4	Yellow	38	Red
5	Gray	39	Black
6	Pink	40	Violet
7	Blue	41	Gray – Pink
8	Red	42	Red – Blue
9	Black	43	White – Green
10	Violet	44	Brown – Green
11	Gray – Pink	45	White – Yellow
12	Red – Blue	46	Yellow – Brown
13	White – Green	47	White – Gray
14	Brown – Green	48	Gray – Brown
15	White – Yellow	49	White – Pink
16	Yellow – Brown	50	Pink – Brown
17	White – Gray	51	White – Blue
18	Gray – Brown	52	Brown – Blue
19	White – Pink	53	White – Red
20	Pink – Brown	54	Brown – Red
21	White – Blue	55	White – Black
22	Brown – Blue	56	Brown – Black
23	White – Red	57	Gray – Green
24	Brown – Red	58	Yellow – Green
25	White – Black	59	Pink – Green

continues on next page

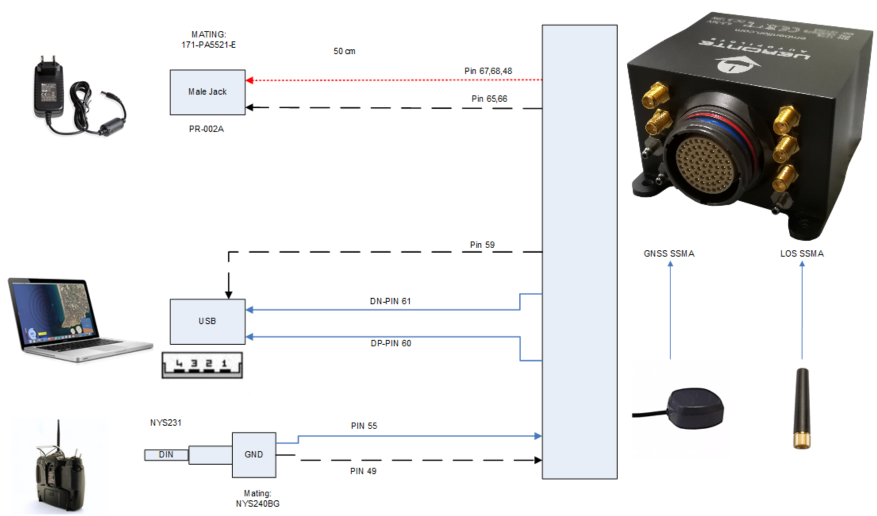
Table 2 – continued from previous page

PIN	Color code	PIN	Color code
26	Brown – Black	60	Yellow – Pink
27	Grey – Green	61	White
28	Yellow – Green	62	Brown
29	Pink – Green	63	Green
30	Yellow – Pink	64	Yellow
31	White	65	Grey
32	Brown	66	Pink
33	Green	67	Blue
34	Yellow	68	Red

Warning: The colour code is repeated due to the amount of pins. Check the pin number before connecting. Pin number increases following the black line of the picture above.

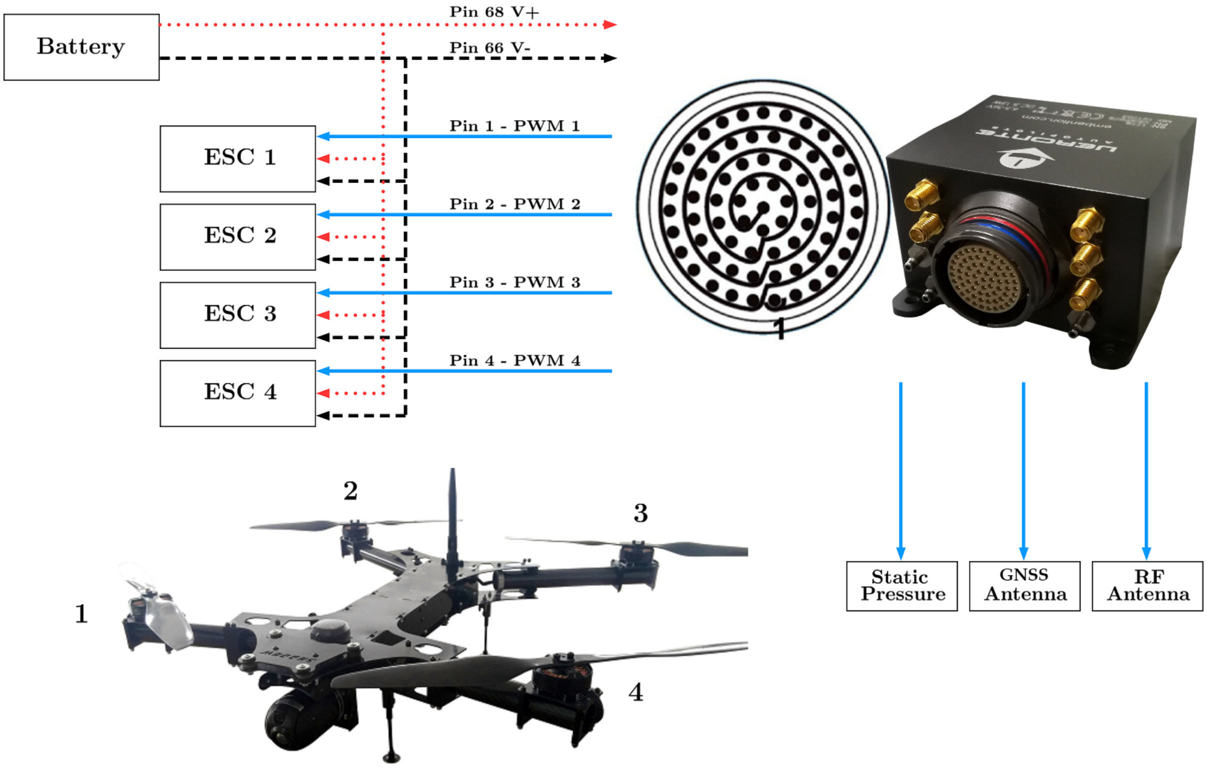
2.1.5 Annex 2: Connection examples

2.1.5.1 Ground Station



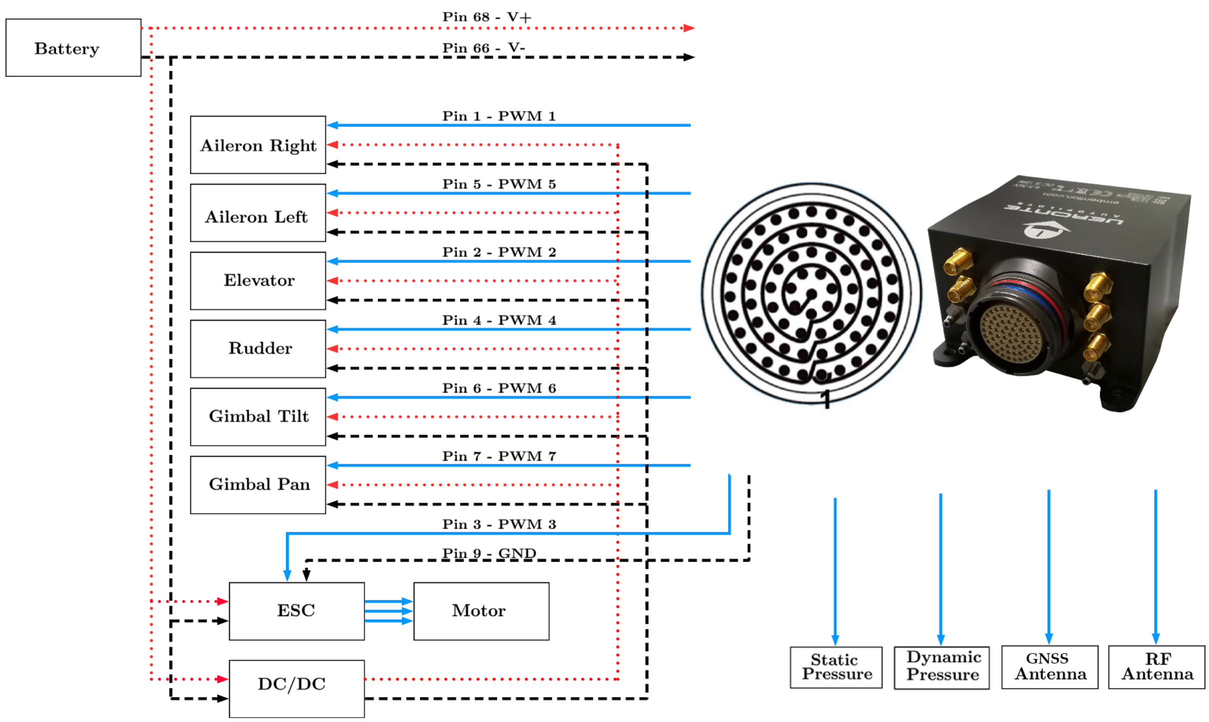
Ground Station

2.1.5.2 Multicopter



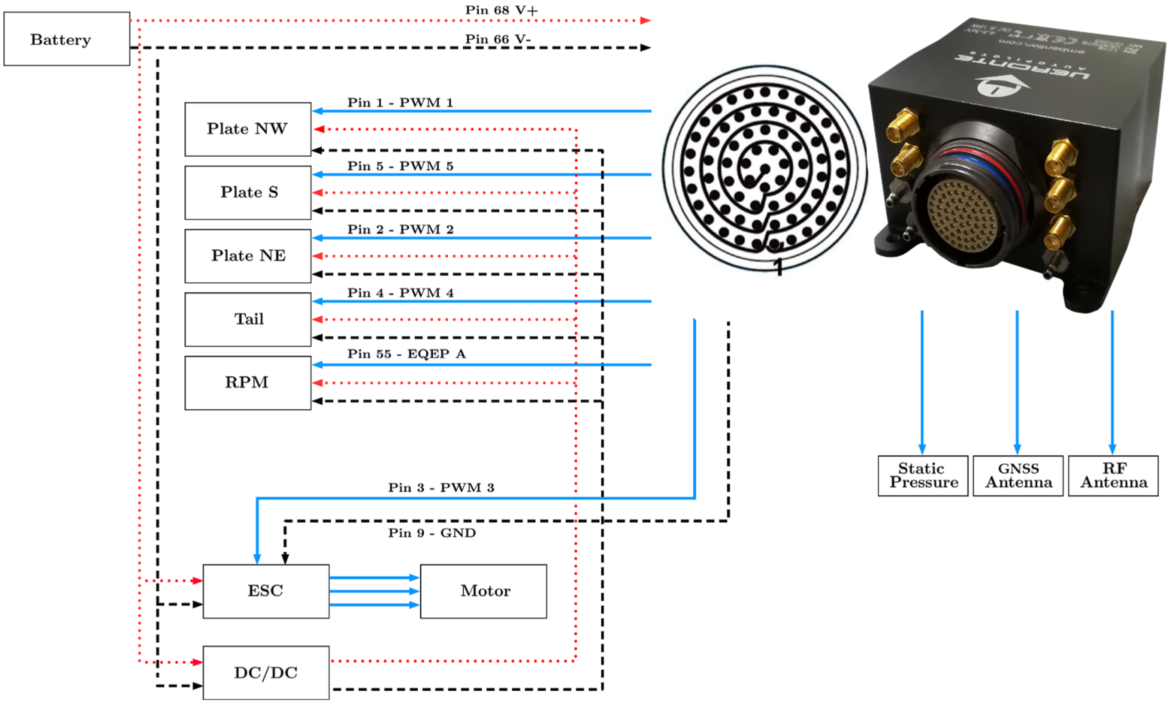
Multicopter

2.1.5.3 Fixed Wing



Fixed Wing

2.1.5.4 Helicopter



Helicopter

Veronte Autopilot

Veronte Autopilot is a miniaturized high-reliability avionics system for advanced control of unmanned systems. This control system embeds a state-of-the-art suite of sensors and processors together with **LOS and BLOS M2M datalink** radio, all with reduced size and weight.

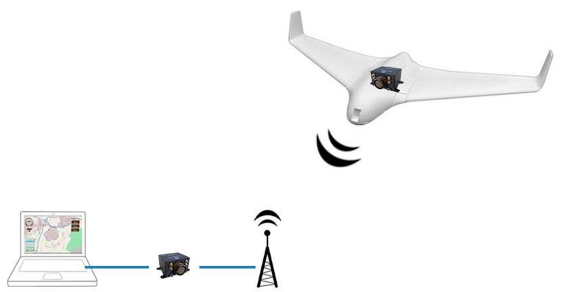
2.1.6 Operation

The unique **Plug ´n Fly** control system, Veronte Autopilot ads fully autonomous control capabilities to any unmanned system for complete operation, compatible with: UAV, Drone, RPAS, USV, UGV…

- **Highly configurable:** Veronte control system is fully configurable; payload, platform layout, control phases, control channels… even the user interface layout can be user-defined.
- **Custom routines:** User selectable automatic actions, activated on system event or periodically.
  - **Actions:** phase change, activate payload, move servo, go to, onboard log, parachute release…
  - **Events:** waypoint arrival, inside/outside polygon, alarm, variable range, button…
- **Telemetry & log:** Embedded datalink for system monitoring and telecommand and customizable user log in both onboard and control station, all with user-defined variables and frequency record.
- **External sensor:** Support for external sensor connection: magnetometer, radar, LIDAR, RPM, temperature, fuel level, battery level, weather…
- **Payload & Peripheral:** Transponder, secondary radios, satcom transceivers, camera gimbals, motor drivers, photo cameras, flares, parachute release systems, tracking antennas, pass through RS232, RS485 & CAN tunnel…

2.1.7 Platforms

The Veronte Autopilot is designed to control any unmanned vehicle, either aircraft such as: multirotors, helicopters, airplanes, VTOL, blimps… as well as ground vehicles, surface vehicles or many others. Custom flight phases and control channels provide support for any aircraft layout and performance by using the same software and hardware for: UAS, RPAS, Drone, USV / ASV, UGV…



Veronte FCS overview

Veronte contains all the electronics and sensors needed in order to properly execute all the functions needed to control the UAV. A Veronte-based FCS contains the following elements:

- Veronte (Air): it executes in real time all the guidance, navigation and control algorithms for the carrying airframe, acting on the control surfaces and propulsion system and processing the signals from different sensors: accelerometers, gyroscopes, magnetometer, static pressure, dynamic pressure, GPS (EGNOS/Galileo compatible).
- Veronte (Ground): apart from linking to other flying Veronte units and supporting manual and arcade modes with conventional joysticks, it can also control a directional antenna in order to expand the maximum range. It communicates to Veronte Pipe (software for ground segment mission management).
- Veronte Pipe: software for mission management at the ground segment. It monitors flying vehicles in real time and can also reproduce past missions in an offline manner. It is also the graphical user interface where commands and flight plans are produced.

2.1.8 Safety

Veronte autopilot includes the following features in order to provide your UAS with the best safety performances:

- Redundant IMU.
- Redundant GNSS receiver.
- Redundant Pressure sensor.
- Dual core principal microprocessor + dissimilar safety microcontroller (comicro).

Independent power supply for main system and safety microcontroller.

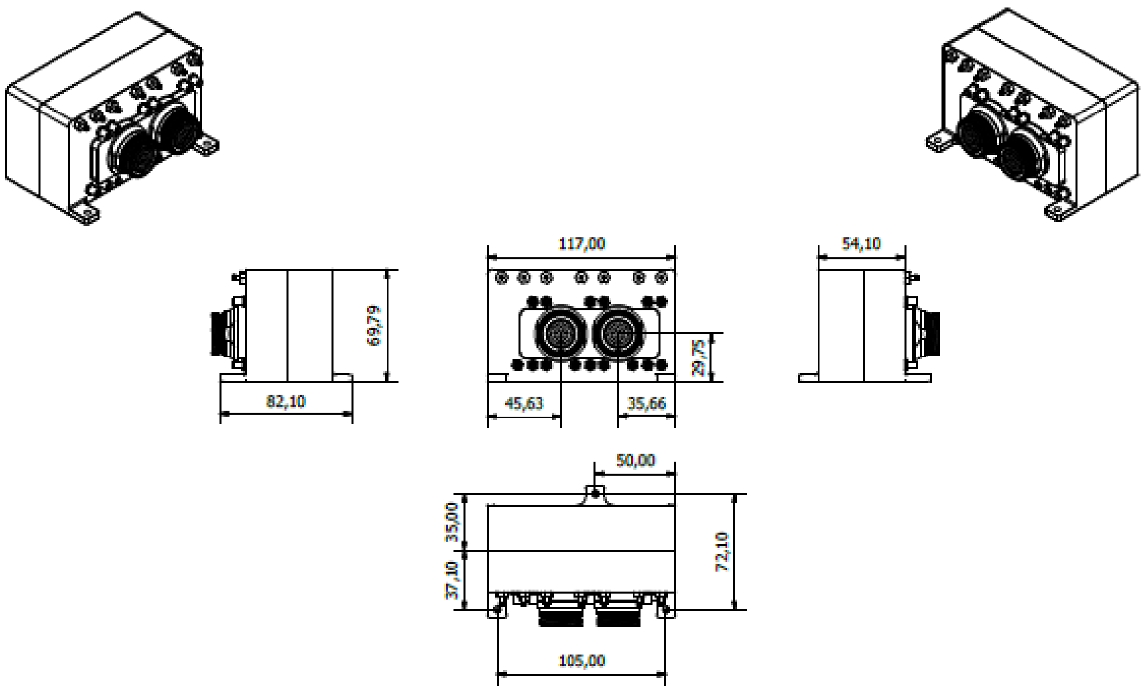
In case any malfunction occurs in the microprocessor, the comicro can activate different safety mechanism by means of 2 digital outputs and 1 serial port.

2.2 4x Veronte Autopilot

2.2.1 Aircraft Mounting

2.2.1.1 Enclosure

4xVeronte is provided using an anodized-aluminium enclosure with enhanced EMI shielding and IP protection. The approximate total weight including radio modules is 750g. The following figure show the dimensions of the enclosure. M4 screws are recommended for mounting.



4xVeronte dimensions (mm)

2.2.1.2 Vibration Isolation

Although Veronte rejects noise and modes of vibration with internal electronic and mechanical filters, an external vibration isolation might be needed depending of the vehicle.

Veronte can be mounted in different ways in order to reject the airframe vibration if needed. One way to avoid vibration would be the use of some external structure which could be rigidly attached to the airframe and softly attached to Veronte (e.g. foam, silent blocks, etc.)

The user should take into account that wiring should be loose enough so vibrations may not find another way to enter the aircraft system.

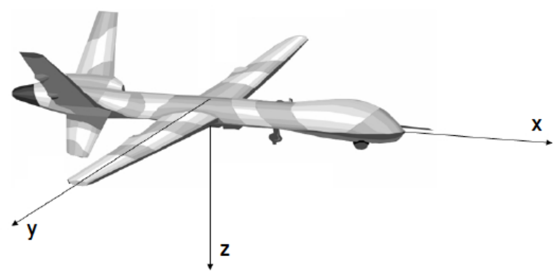
2.2.1.3 Location

The location of 4xVeronte has no restrictions. You only need to configure its relative position with respect to the centre of mass of the aircraft and the GNSS antenna. The configuration of the location of Veronte can be easily configured using Veronte Pipe Software.

2.2.1.4 Orientation

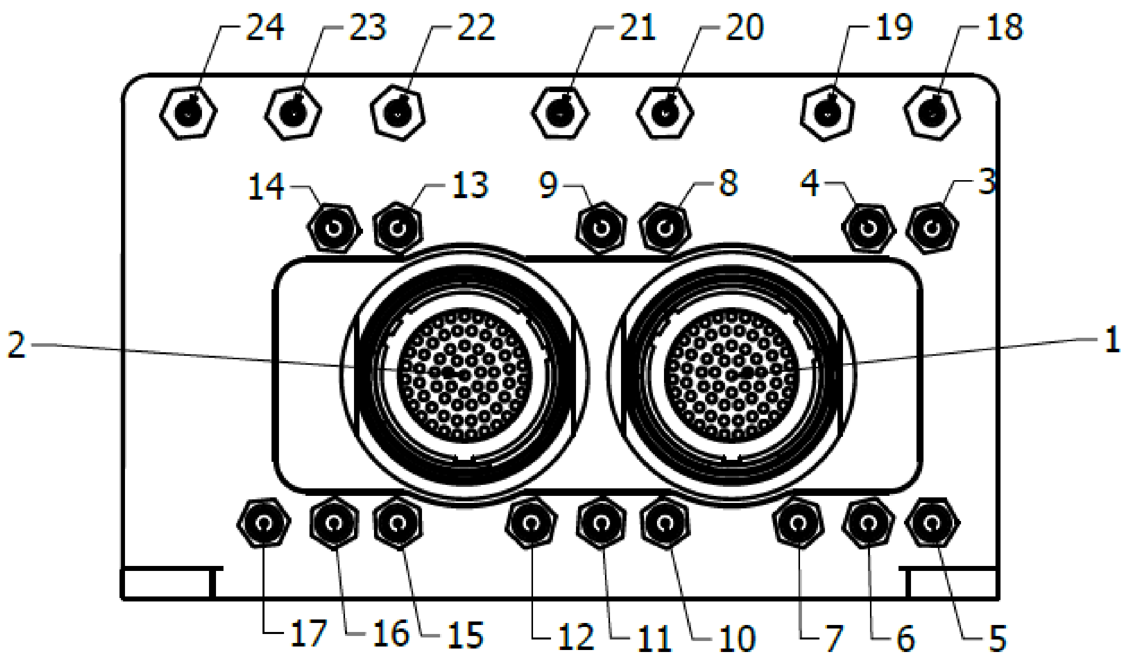
The orientation of 4xVeronte has no restrictions either. You only need to configure Veronte axes with respect to the aircraft body axes by means of a rotation matrix or a set of correspondences between axes. The configuration of the location of Veronte can be easily configured using Veronte Pipe Software.

Veronte axes are printed on the box and aircraft reference frame are defined by the standard flight dynamics conventions.



Aircraft Mounting

2.2.1.5 Connector Layout



4xVeronte Connectors

Index	Connector
1	Redundant (Critical) connector
2	Arbiter (Optional) connector
3	LOS SSMA connector for Veronte 3
4	GNSS1 SSMA connector for Veronte 3
5	M2M SSMA connector for Veronte 3
6	GNSS2 SSMA connector for Veronte 3
7	TPDR (Transponder) SSMA connector for Veronte 3
8	LOS SSMA connector for Veronte 2
9	GNSS1 SSMA connector for Veronte 2
10	M2M SSMA connector for Veronte 2
11	GNSS2 SSMA connector for Veronte 2
12	TPDR (Transponder) SSMA connector for Veronte 2
13	LOS SSMA connector for Veronte 1
14	GNSS1 SSMA connector for Veronte 1
15	M2M SSMA connector for Veronte 1
16	GNSS2 SSMA connector for Veronte 1
17	TPDR (Transponder) SSMA connector for Veronte 1
18	Dynamic pressure port (Fitting 5/64in) for Veronte 3
19	Static pressure port (Fitting 5/64in) for Veronte 3
20	Dynamic pressure port (Fitting 5/64in) for Veronte 2
21	Static pressure port (Fitting 5/64in) for Veronte 2
22	Dynamic pressure port (Fitting 5/64in) for Veronte 1
23	Static pressure port (Fitting 5/64in) for Veronte 1
24	Static pressure port (Fitting 5/64in) for all Verontes

For pressure ports, mating with clamped 2mm internal diameter flexible tubing is recommended.

The static pressure port for all Verontes sets the 4xVeronte internal cage pressure.

2.2.1.6 Mating Connectors

Index	Connector	Mating Connector
3,8,13	RF antenna/(SSMA Jack Female)	SSMA male Plug , low-loss cable is recommended.
4,9,14,16,11,6	GNSS antenna/(SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended./Active Antenna GNSS: Gain min 15dB (to compensate signal loss in RF Cable) max 50dB, maximum noise figure 1.5dB, power supply 3.3V max current 20 mA
10,12,14	M2M antenna/(SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.
1	Redundant Connector/Connector HEW.LM.368.XLNP	Mating connector P/N: FGW.LM.368.XLCT/Mating harness is available on demand.
2	Arbiter Connector/Connector HEW.LM.368.XLNP	Mating connector P/N: FGW.LM.368.XLCT/Mating harness is available on demand.
7,12,17	TPDR antenna (SSMA Jack Female)	SSMA male Plug, low-loss cable is recommended.

2.2.1.7 Antenna Integration

The system uses different kinds of antennas to operate that must be installed on the airframe. Here you can find some advice for obtaining the best performance and for avoiding antenna interferences.

Antenna Installation
Maximize separation between antennas as much as possible.
Keep it far away from alternators or other interference generators.
Always isolate antenna ground panel from the aircraft structure.
Make that the antenna is securely mounted.
Always use high-quality RF wires minimising the wire length.
Always follow the antenna manufacturer manual.
SSMA connections shall be tightened applying 1Nm of torque
For all-weather aircraft, insert SSMA lightning protectors.

GNSS Antenna
Antenna top side must point the sky.
Install it on a top surface with direct sky view.
Never place metallic / carbon parts or wires above the antenna.
It is recommended to install it on top of a ground plane.
For all-weather aircraft, insert SSMA lightning protectors.

2.2.1.8 Pressure lines

4xVeronte has 6 pressure input lines, 3 for static pressure to determine the absolute pressure and 3 for pitot in order to determine the dynamic pressure on each internal autopilot.

Absolute pressure connection on the aircraft is mandatory while pitot port can be obviated in some aircrafts. Pitot port absence must be configured on Veronte Pipe software.

Pressure Intake
Pressure intakes must be located in order to prevent clogging.
Never install pressure intakes on the propeller flow.
Design pressure tubing path in order to avoid tube constriction.

Static Pressure
It is not recommended to use inside fuselage pressure if it is not properly vented.

Pitot Tube
It is recommended to install it near the aircraft axis in order to avoid false measures during manoeuvres.
For low-speed aircraft it is recommended at least 6,3mm tubes for preventing rain obstruction.
Pitot tube must be installed facing the airflow in the direction of the “x” axis of the aircraft.

2.2.2 Electrical

2.2.2.1 Power

4xVeronte can use unregulated DC (6.5V to 36V) for the internal Veronte autopilots and also for the arbiter.

LiPo batteries between 2S and 8S can be used without voltage regulation. Remaining battery can be controlled by the internal voltage sensor and by configuring the voltage warnings on the PC application.

For higher voltage installations, voltage regulators must be used. For dimensioning voltage regulators take into account that a blocked servo can activate regulator thermal protection.

> **Warning: Caution!!** Power Veronte out of the given range can cause irreversible damage to the system. Please read carefully the manual before powering the system.

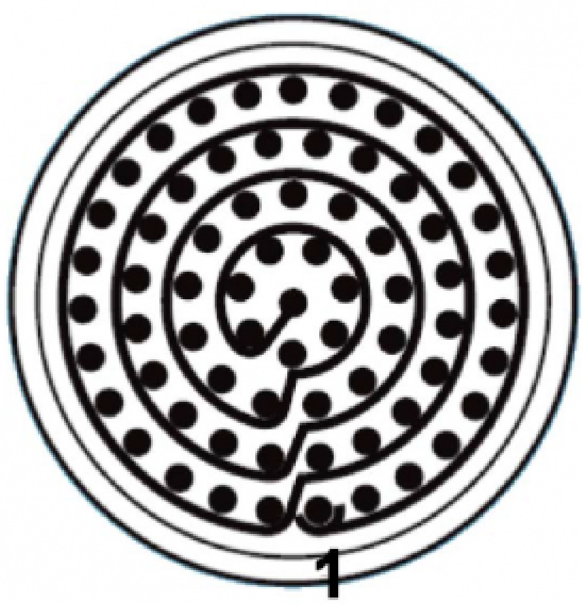
Veronte and servos can be powered by the same or different batteries. In case there are more than one battery on the system, a single point ground union it is needed to ensure a good performance. The ground signal should be isolated from other system ground references (e.g. engines).

It is recommendable to use independent switches for autopilot and motor / actuators. During the system initialization, PWM signal will be fixed to low level (0V), please make sure that actuators / motor connected support this behaviour before installing a single switch for the whole system.

Despite the names, all GND connectors share the same line, meaning they can be used for any ground connection required.

Harness ended in Blue matting connector will refer to Redundant connector, and the one ended in Yellow (reverse polarity) will refer to Arbiter connector.

2.2.2.2 Redundant Connector Pinout



68-pin redundant connector for 4xVeronte Autopilot

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
1	I/O1	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
2	I/O2	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
3	I/O3	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
4	I/O4	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
5	I/O5	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
6	I/O6	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
7	I/O7	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
8	I/O8	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
9	GND	GROUND	•	GROUND SIGNAL FOR ACTUATORS 1-8
10	I/O9	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)

continues on next page

Table 3 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
11	I/O10	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
12	I/O11	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
13	I/O12	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
14	I/O13	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
15	I/O14	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
16	I/O15	I/O	A	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
17	I/O16	I/O	B	MUXED PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
18	GND	GROUND	•	GROUND SIGNAL FOR ACTUATORS 9-16
19	RS_232_TX	OUTPUT	A	MUXED RS-232 OUTPUT
20	RS_232_RX	INPUT	A	REDUNDANT RS-232 INPUT
21	V2_USB_DP	I/O	•	VERONTE 2 USB DATA LINE
22	ANALOG_4	INPUT	B	REDUNDANT ANALOG INPUT 0-36V

continues on next page

Table 3 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
23	ANALOG_5	INPUT	B	REDUNDANT ANALOG INPUT 0-36V
24	V2_USB_DN	I/O	•	VERONTE 2 USB DATA LINE
25	CANA_P	I/O	•	CANbus interface. It supports data rates up to 1 Mbps. Twisted pair with a 120 Zo recommended
26	CANA_N	I/O	•	
27	GND	GROUND	•	GROUND SIGNAL FOR BUSES
28	CANB_P	I/O	•	CANbus interface. It supports data rates up to 1 Mbps. Twisted pair with a 120 Zo recommended
29	CANB_N	I/O	•	
30	V2_USB_ID	I/O	•	VERONTE 2 USB ID LINE
31	I2C_CLK	OUTPUT A	•	MUXED CLK LINE FOR I2C BUS
32	I2C_DATA	I/O	A	MUXED DATA LINE FOR I2C BUS
33	GND	GROUND	•	GROUND FOR 3.3V POWER SUPPLY
34	3.3V	POWER	B	3.3V-100mA POWER SUPPLY
35	GND	GROUND	•	GROUND FOR 5V POWER SUPPLY
36	5V	POWER	B	5V-100mA POWER SUPPLY
37	GND	GROUND	•	GROUND FOR ANALOG SIGNALS
38	ANALOG_1	INPUT	A	REDUNDANT ANALOG INPUT 0-36V
39	ANALOG_2	INPUT	A	REDUNDANT ANALOG INPUT 0-36V

continues on next page

Table 3 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
40	ANALOG_3	INPUT	A	REDUNDANT ANALOG INPUT 0-36V
41	GND	GROUND	•	GROUND SIGNAL FOR BUSES
42	V3_USB_DP	I/O	•	VERONTE 3 USB DATA LINE
43	V3_USB_DN	I/O	•	VERONTE 3 USB DATA LINE
44	GND	GROUND	•	GROUND SIGNAL FOR BUSES
45	UART_TX	OUTPUT	B	MUXED UART OUTPUT
46	UART_RX	INPUT	B	REDUNDANT UART INPUT
47	GND	GROUND	•	GROUND SIGNAL FOR BUSES
48	GND	GROUND	•	GROUND SIGNAL FOR BUSES
49	V3_USB_ID	I/O	•	VERONTE 3 USB ID LINE
50	OUT_RS485_P	OUTPUT	B	NON-INVERTED OUTPUT FOR MUXED RS-485 BUS
51	OUT_RS845_N	OUTPUT	B	INVERTED OUTPUT FOR MUXED RS-485 BUS
52	IN_RS845_N	INPUT	•	INVERTED INPUT FOR MUXED RS-485 BUS
53	IN_RS485_P	INPUT	•	NON-INVERTED INPUT FOR MUXED RS-485 BUS
54	RS-485_GND	GROUND	•	GROUND FOR RS-485 BUS
55	EQEP_A	INPUT	A FOR VERONTE 1&2 B FOR VERONTE 3	ENCODER QUADRATURE REDUNDANT INPUT A (0-5V)

continues on next page

Table 3 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
56	EQEP_B	INPUT		ENCODER QUADRATURE REDUNDANT INPUT B (0-5V)
57	EQEP_S	INPUT		ENCODER STROBE REDUNDANT INPUT (0-5V)
58	EQEP_I	INPUT		ENCODER INDEX REDUNDANT INPUT (0-5V)
59	GND3	GROUND	•	VERONTE 3 GROUND INPUT
60	V1_USB_DP	I/O	•	VERONTE 1 USB DATA LINE
61	V1_USB_DN	I/O	•	VERONTE 1 USB DATA LINE
62	V1_USB_ID	I/O	•	VERONTE 1 USB ID LINE
63	GND	GROUND	•	GROUND SIGNAL FOR BUSES
64	VCC3	POWER	•	VERONTE 3 POWER SUPPLY (6.5 to 36V)
65	GND2	GROUND	•	VERONTE 2 GROUND INPUT
66	GND1	GROUND	•	VERONTE 1 GROUND INPUT
67	VCC2	POWER	•	VERONTE 2 POWER SUPPLY (6.5 to 36V)
68	VCC1	POWER	•	VERONTE 1 POWER SUPPLY (6.5 to 36V)

2.2.2.3 Arbiter Connector Pinout

Although being the same part, Arbiter connector and the Redundant connector are polarized differently to avoid wiring swapping.

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
1	EXTERNAL FCU I/O1	I/O	A	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
2	EXTERNAL FCU I/O2	I/O	B	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
3	EXTERNAL FCU I/O3	I/O	A	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
4	EXTERNAL FCU I/O4	I/O	B	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
5	EXTERNAL FCU I/O5	I/O	A	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
6	EXTERNAL FCU I/O6	I/O	B	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.

continues on next page

Table 4 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
7	EXTERNAL FCU I/O7	I/O	A	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
8	EXTERNAL FCU I/O8	I/O	B	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
9	EXTERNAL FCU I/O9	I/O	A	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
10	EXTERNAL FCU I/O10	I/O	B	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
11	EXTERNAL FCU I/O11	I/O	A	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
12	EXTERNAL FCU I/O12	I/O	B	EXTERNAL FCU PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V). Input current must be limited to 25mA.
13	VCC2	POWER	•	VERONTE 2 POWER SUPPLY (6.5 to 36V)

continues on next page

Table 4 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
14	EXTERNAL FCU ANALOG INPUT 1	OUTPUT	A	EXTERNAL FCU ANALOG INPUT (0-36V). This is the analog signal corresponding to Analog signal 1 on Redundant connector.
15	EXTERNAL FCU ANALOG INPUT 2	OUTPUT	A	EXTERNAL FCU ANALOG INPUT (0-36V). This is the analog signal corresponding to Analog signal 2 on Redundant connector.
16	EXTERNAL FCU ANALOG INPUT 3	OUTPUT	A	EXTERNAL FCU ANALOG INPUT (0-36V). This is the analog signal corresponding to Analog signal 3 on Redundant connector.
17	EXTERNAL FCU ANALOG INPUT 4	OUTPUT	B	EXTERNAL FCU ANALOG INPUT (0-36V). This is the analog signal corresponding to Analog signal 4 on Redundat connector.
18	FLIGHT TERMINATION SIGNAL STAGE B	OUTPUT	B	OPEN DRAIN OUTPUT FROM VOTING STAGE B (Sensed by arbiter)
19	EXTERNAL FCU TO PAYLOAD UART SIGNAL	INPUT	A	EXTERNAL FCU UART OUTPUT. (0-3.3V) THIS SIGNAL WILL BE AN INPUT FOR THE RS_232 OUTPUT MULTIPLEXER. MULTIPLEXED OUTPUT ON REDUNDANT CONNECTOR PIN 19

continues on next page

Table 4 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
20	PAYLOAD TO EXTERNAL FCU UART SIGNAL	OUTPUT	A	EXTERNAL FCU UART INPUT. (0-3.3V) THIS SIGNAL WILL BE THE OUTPUT OF THE RS_232 MULTIPLEXER. MULTIPLEXED INPUT ON REDUNDANT CONNECTOR PIN 20
21	PAYLOAD TO EXTERNAL FCU RS-485_P	OUTPUT	•	RS-485_P OUTPUT SIGNAL FROM PAYLOAD (REDUNDANT CONNECTOR PIN 53)TO FCU RS-485_P INPUT
22	PAYLOAD TO EXTERNAL FCU RS-485_N	OUTPUT	•	RS-485_N OUTPUT SIGNAL FROM PAYLOAD (REDUNDANT CONNECTOR PIN 52)TO FCU RS-485_N INPUT
23	EXTERNAL FCU TO PAYLOAD RS-485_P	INPUT	B	RS-485_P OUTPUT SIGNAL FROM EXTERNAL FCU (REDUNDANT CONNECTOR PIN 50)TO PAYLOAD RS-485_P INPUT
24	EXTERNAL FCU TO PAYLOAD RS-485_N	INPUT	B	RS-485_N OUTPUT SIGNAL FROM EXTERNAL FCU (REDUNDANT CONNECTOR PIN 51)TO PAYLOAD RS-485_N INPUT
25	CANA_P	I/O	•	CANbus interface. It supports data rates up to 1 Mbps. Recommended cable is a twisted pair with a 120 Zo.
26	CANA_N	I/O	•	

continues on next page

Table 4 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
27	VCC1	POWER	•	VERONTE 1 POWER SUPPLY (6.5 to 36V)
28	CANB_P	I/O	•	CANbus interface. It supports data rates up to 1 Mbps. Recommended cable is a twisted pair with a 120 Zo.
29	CANB_N	I/O	•	
30	ARBITER_RS485_OUT_P	OUTPUT	ARBITER	NON-INVERTED OUTPUT FOR ARBITER'S RS-485 BUS
31	ARBITER_RS485_OUT_N	OUTPUT	ARBITER	INVERTED OUTPUT FOR ARBITER'S RS-485 BUS
32	ARBITER_RS485_IN_N	INPUT	ARBITER	INVERTED INPUT FOR ARBITER'S RS-485 BUS
33	ARBITER_RS485_IN_P	INPUT	ARBITER	NON-INVERTED INPUT FOR ARBITER'S RS-485 BUS
34	ARBITER_ARINC_TXA	OUTPUT	ARBITER	ARBITER'S ARINC POSITIVE OUTPUT
35	ARBITER_ARINC_TXB	OUTPUT	ARBITER	ARBITER'S ARINC NEGATIVE OUTPUT
36	ARBITER_ARINC_RXA	INPUT	ARBITER	ARBITER'S ARINC POSITIVE INPUT
37	ARBITER_ARINC_RXB	INPUT	ARBITER	ARBITER'S ARINC NEGATIVE INPUT
38	GND	GROUND	•	GROUND SIGNAL FOR BUSES
39	ARBITER_I2C_SCL	OUTPUT	ARBITER	CLK LINE FOR ARBITER'S I2C BUS
40	ARBITER I2C_DATA	I/O	ARBITER	DATA LINE FOR ARBITER'S I2C BUS
41	ARBITER_RS232B_RX	INPUT	ARBITER	ARBITER RS-232 INPUT B

continues on next page

Table 4 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
42	ARBITER_RS232B_TX	OUTPUT	ARBITER	ARBITER RS-232 OUTPUT B
43	ARBITER_RS232A_RX	INPUT	ARBITER	ARBITER RS-232 INPUT A
44	ARBITER_RS232A_TX	OUTPUT	ARBITER	ARBITER RS-232 OUTPUT A
45	GND	GROUND	•	GROUND SIGNAL FOR ANALOG SIGNALS
46	ARBITER ANALOG_INPUT_1	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
47	ARBITER ANALOG_INPUT_2	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
48	ARBITER ANALOG_INPUT_3	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
49	ARBITER ANALOG_INPUT_4	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
50	ARBITER ANALOG_INPUT_5	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
51	ARBITER ANALOG_INPUT_6	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
52	ARBITER ANALOG_INPUT_7	I/O	ARBITER	ARBITER ANALOG INPUT (0-36V)
53	FLIGHT TERMINATION SIGNAL STAGE A	OUTPUT	A	OPEN DRAIN OUTPUT FROM VOTING STAGE A (Sensed by arbiter)
54	ARB_GPIO9	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
55	ARB_GPIO10	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
56	WD_EXT	I	ARBITER	WATCHDOG SIGNAL FROM EXTERNAL AUTOPILOT TO ARBITER (0-3.3V)

continues on next page

Table 4 – continued from previous page

PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
57	EXT_DETECT	I	ARBITER	CONNECT TO GND IF EXTERNAL FCU IS CONNECTED. OTHERWISE, LEAVE OPEN
58	GND	GROUND	•	GROUND SIGNAL FOR GPIO
59	ARB_GPIO1	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
60	ARB_GPIO2	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
61	ARB_GPIO3	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
62	ARB_GPIO4	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
63	ARB_GPIO5	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
64	ARB_GPIO6	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
65	ARB_GPIO7	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)
66	ARB_GPIO8	I/O	ARBITER	ARBITER'S PWM/DIGITAL OUTPUT/DIGITAL INPUT SIGNAL (0-3.3V)

continues on next page

Table 4 – continued from previous page

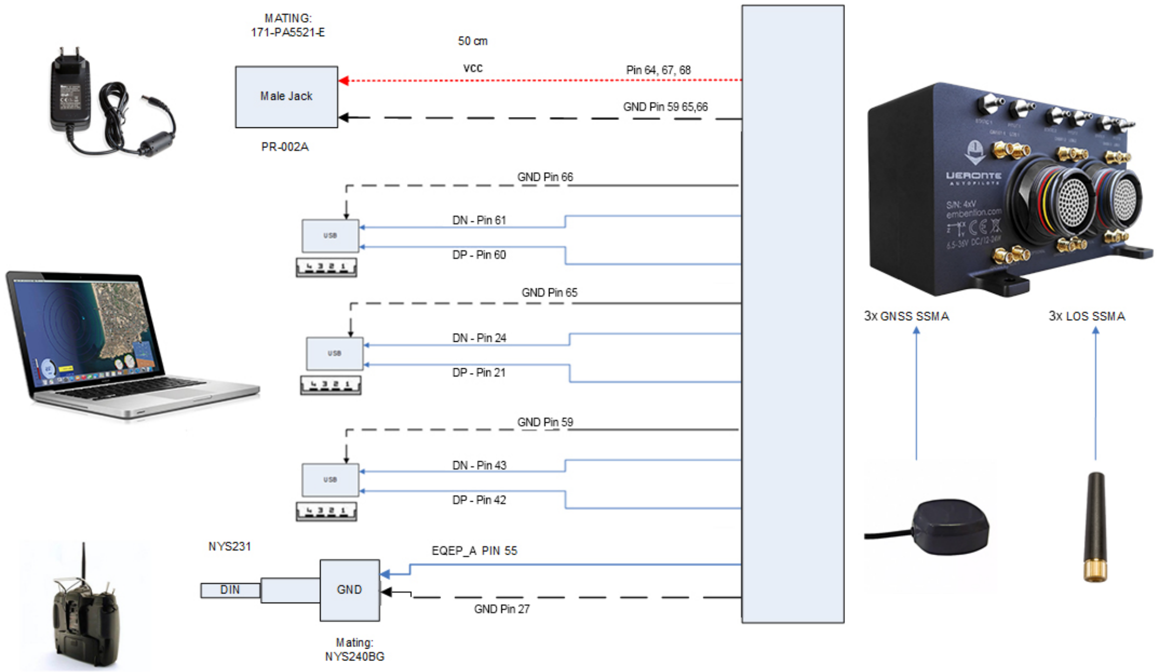
PIN	SIGNAL	TYPE	INTERNAL POWER DOMAIN	COMMENTS
67	GND_ARBITER	GROUND	•	ARBITER GROUND INPUT
68	VCC_ARBITER	POWER	•	ARBITER POWER SUPPLY (6.5 to 36V)

2.2.3 Performances

Variable	Value
Weight (radio included)	750 g
Power Input	6.5 V to 36 V
Minimum Temperature	-40 ºC
Maximum Temperature	+55ºC (No convection, ask for increased limits (up to 71ºC))
Max. Internal Temperature	+85ºC
Minimum Pressure	0 kPa
Maximum Pressure	104 kPa
Maximum Dynamic pressure	6 kPa (Ask for increased limits (up to 50kPa))
Protection Rating	IP67 enclosure version
Acceleration Limits (3 axes)	$\pm$2 g to $\pm$24 g (for sustained maneuvers, transitional higher accelerations are possible (e.g. catapult launch). Ask for increased limits.)
Angular Velocity Limits (3 axes)	$\pm$125 deg/s to $\pm$ 2000 deg/s (for sustained maneuvers, transitional higher angular velocities are possible. Ask for increased limits.)
Magnetic Field Limits (3 axes)	$\pm$4 to $\pm$16 Gauss
GNSS	72 channels, GPS L1C/A, GLONASS L1OF, BeiDou B1I
Datalink	410 to 480 MHz licensed or FHSS/902-928MHz FHSS/2.4 to 2.483 GHz ISM Band/869.5-869.75 MHz ISM Band
Special Datalinks on request	920 – 925 MHz, Singapore regulation compliance/869.5-869.75 MHz ISM Band

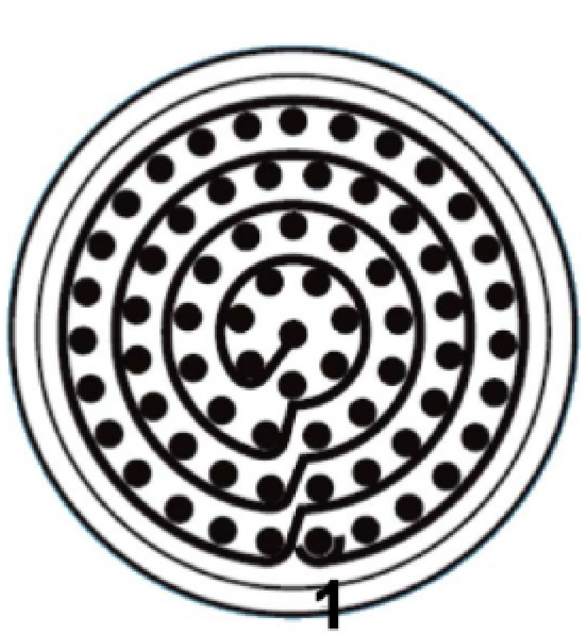
2.2.4 Annex 1: Connection Examples

2.2.4.1 Ground Station



Connection Example

2.2.5 Annex 2: Connector colour code

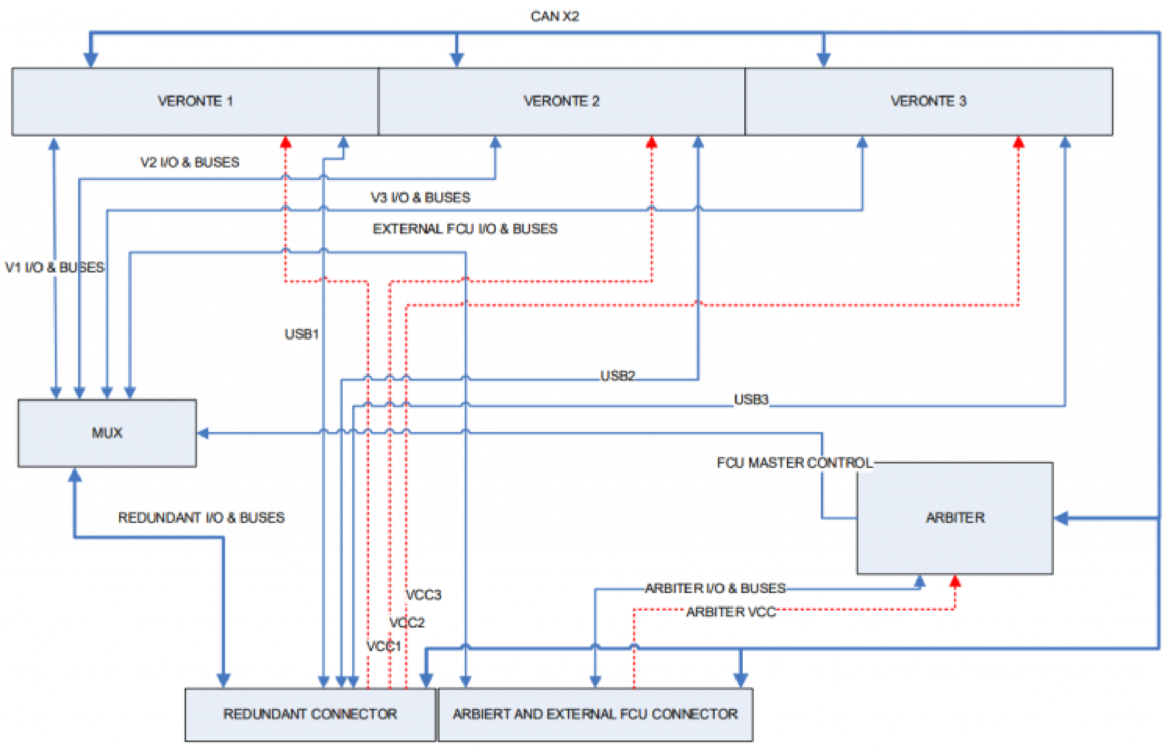


Connector HEW.LM.368.XLNP

PIN	Color code	PIN	Color code
1	White	35	Gray
2	Brown	36	Pink
3	Green	37	Blue
4	Yellow	38	Red
5	Gray	39	Black
6	Pink	40	Violet
7	Blue	41	Gray – Pink
8	Red	42	Red – Blue
9	Black	43	White – Green
10	Violet	44	Brown – Green
11	Gray – Pink	45	White – Yellow
12	Red – Blue	46	Yellow – Brown
13	White – Green	47	White – Gray
14	Brown – Green	48	Gray – Brown
15	White – Yellow	49	White – Pink
16	Yellow – Brown	50	Pink – Brown
17	White – Gray	51	White – Blue
18	Gray – Brown	52	Brown – Blue
19	White – Pink	53	White – Red
20	Pink – Brown	54	Brown – Red
21	White – Blue	55	White – Black
22	Brown – Blue	56	Brown – Black
23	White – Red	57	Gray – Green
24	Brown – Red	58	Yellow – Green
25	White – Black	59	Pink – Green
26	Brown – Black	60	Yellow – Pink
27	Grey – Green	61	White
28	Yellow – Green	62	Brown
29	Pink – Green	63	Green
30	Yellow – Pink	64	Yellow
31	White	65	Grey
32	Brown	66	Pink
33	Green	67	Blue
34	Yellow	68	Red

Warning: The colour code is repeated due to the amount of pins. Check the pin number before connecting. Pin number increases following the black line of the picture above.

4xVeronte Autopilot is a **triple redundant** version of the Veronte Autopilot. It includes three complete Veronte Autopilot modules together with a dissimilar arbiter for detecting system failures and selecting the module in charge of the control. The autopilot selected as the master will be the one controlling the actuators and communicating with the payloads, as seen in the following block diagram.



4xVeronte Overview

Each Veronte autopilot contains all the electronics and sensors in order to properly execute all the functions needed to control the UAV. Veronte executes in real time all the guidance, navigation and control algorithms for the carrying airframe, acting on the control surfaces and propulsion system and processing the signals from different sensors: accelerometers, gyroscopes, magnetometer, static pressure, dynamic pressure, GNSS and external sensors.

Additional I/O port is available for the connection of an **external control** system in case it is required and include it in the redundant scheme. It provides the system with full dissimilarity for high demanding environments as required by civil aviation authorities.

Datalink communications can be also redundant, being possible to install inside the autopilot 3 radios of different frequencies. For example, it allows you to have two radios working in the 900MHz frequency and one in the 2.4GHz, so in case there is any issue in the 900MHz band the module connected to the 2.4GHz band will take control. In addition, an external radio can be controlled as a critical device using the serial port in the redundant connector.

All three modules are managed by a dissimilar microprocessor. This arbiter includes voting algorithms for managing the module in charge of vehicle control. This microprocessor compares data from all modules in real time and processes it for discarding any autopilot module showing an undesired performance.

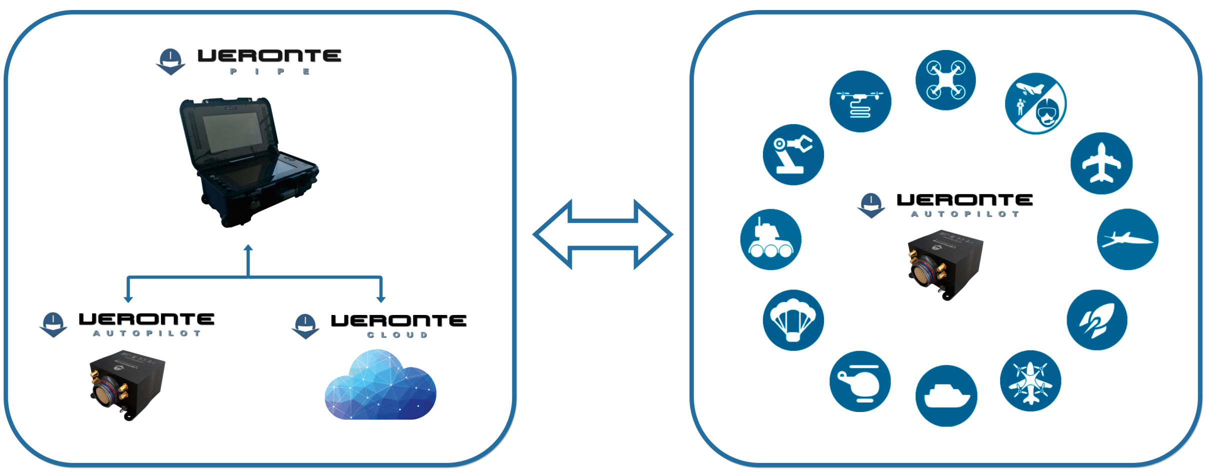
4xVeronte also includes two separate flight termination voting logics, completely dissimilar and implemented with simple hardware, with the purpose of giving the internal three Veronte Autopilots a way to decide by consensus if a flight termination signal should be activated or not.

This redundant device allows platforms to perform **sensitive flight mission** and transport **valuable payload** with advanced safety conditions and high reliabilty. By installing a triple redundant core it is possible to extend the MTBF (Mean Time Between Failures) in the system. This control module is also suitable for both, fail-safe and fail-operational missions. Extending the operability of the system.

Veronte Autopilot Kit

This section provides the information about the Hardware of Veronte Autopilots. The content includes enclosure and mating connectors references, general informations about electrical connections and Veronte specifications.

Veronte is the main element in our Flight Control System for UAS. As shown in the following diagram, Veronte AP is used both in the ground segment **(GND)** and onboard **(AIR)**.



System Overview

Veronte includes the required electronics and sensors in order to be able to properly execute all the needed functions for the UAS control. A Veronte-based FCS consists of the following components:

- **Veronte AIR** - It executes GNC algorithms in real time in order to accomplish the planned mission and handle the payload.
- **Veronte GND** - Link between Veronte Pipe and the AIR Units. It supports manual and arcade modes with conventional joysticks, it's capable to control a directional antenna in order to expand the maximum range and it

can be equipped with external physical interfaces in order to operate the system without Veronte Pipe.

- **Veronte Pipe** - Software dedicated to mission planning, configuration and operation. It allows the user to monitor the connected UAS in real time, to interact with them and to replay previous missions for post-flight analysis.

CHAPTER

THREE

PIPE CONFIGURATION

3.1 Installation

3.1.1 System Requeriments

Minimum Hardware Specification for installation:

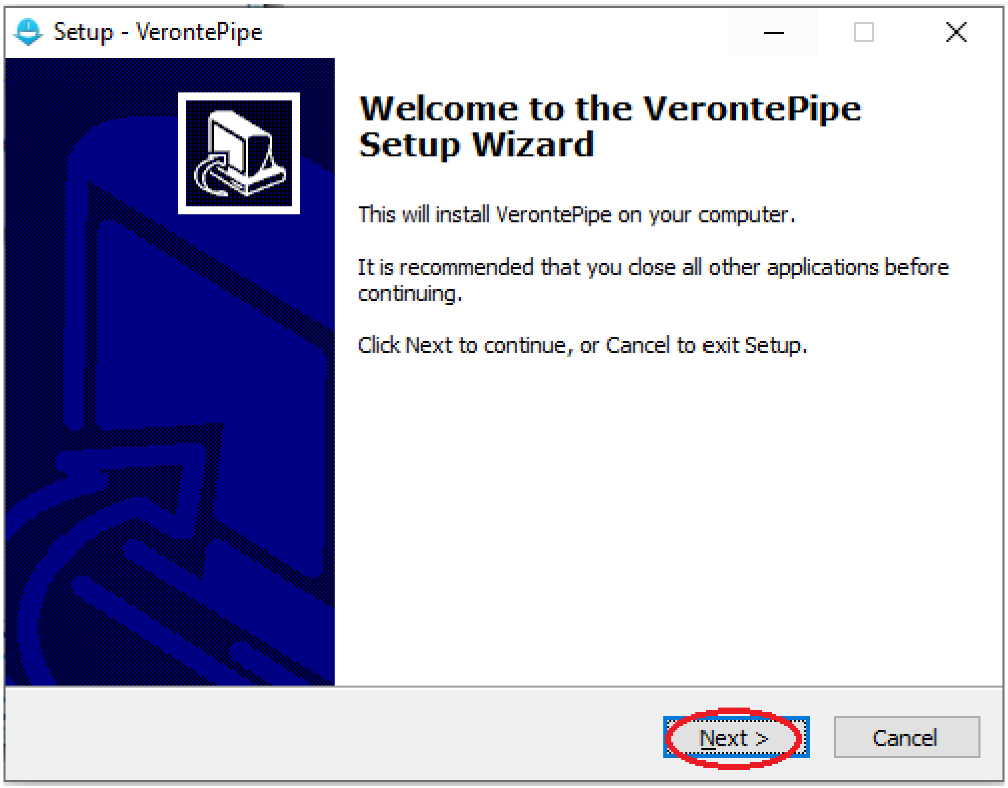
- Intel Core i5 Processor.
- 8 GB RAM
- USB port
- 500 MB free hard drive

Veronte supports the main Operating Systems (Windows, Linux and MacOS X). Contact Embention and we will provide you with the software that better fits your requirements. Also, you must have updated the latest version of java.

3.1.2 Windows

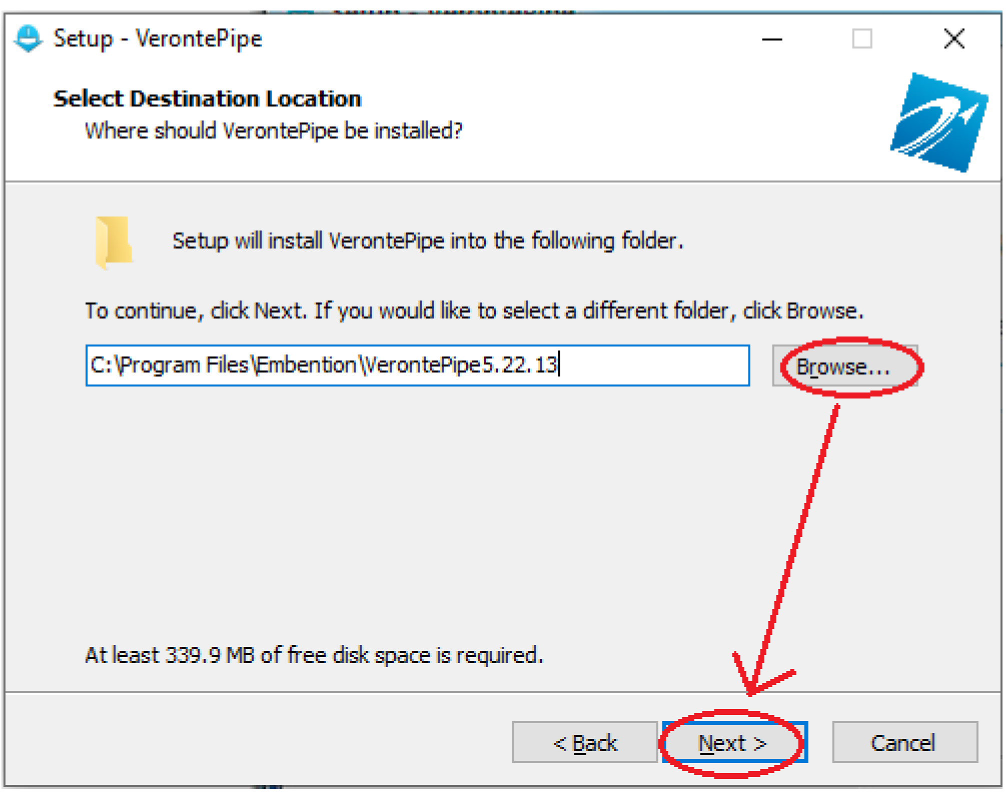
To install Veronte Pipe on Windows just execute "Veronte_Pipe.exe" and follow the indications.

1. Click on Next.



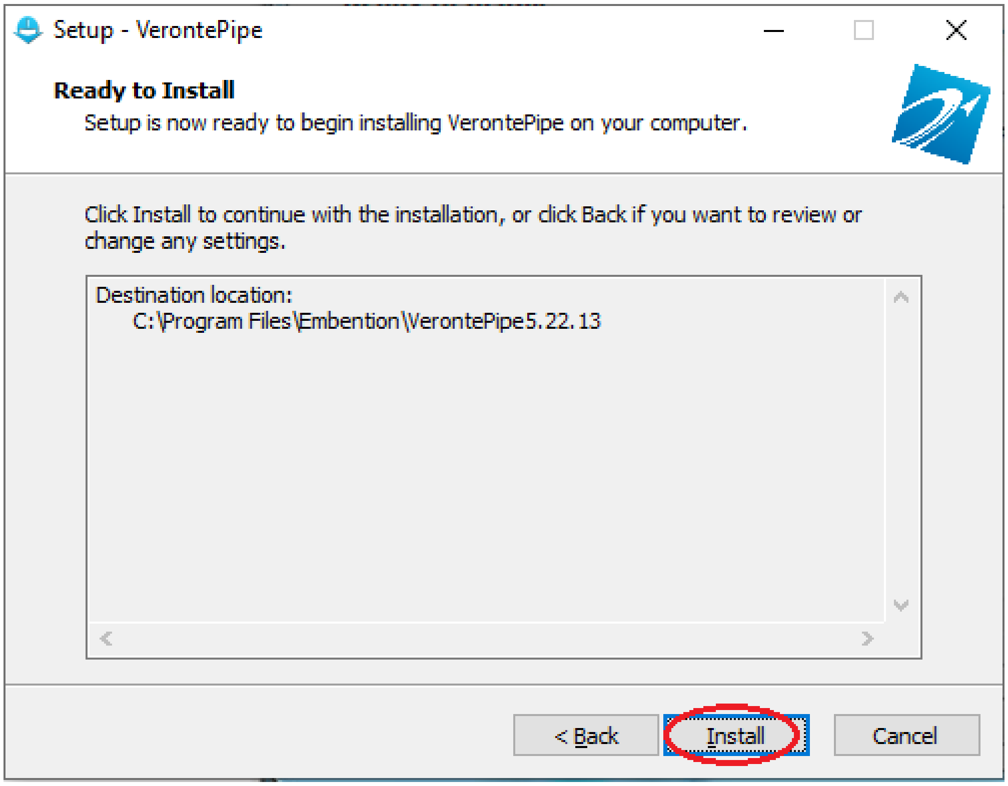
Windows Installation Step 01

2. Select the directory where you want to Install and click on next.



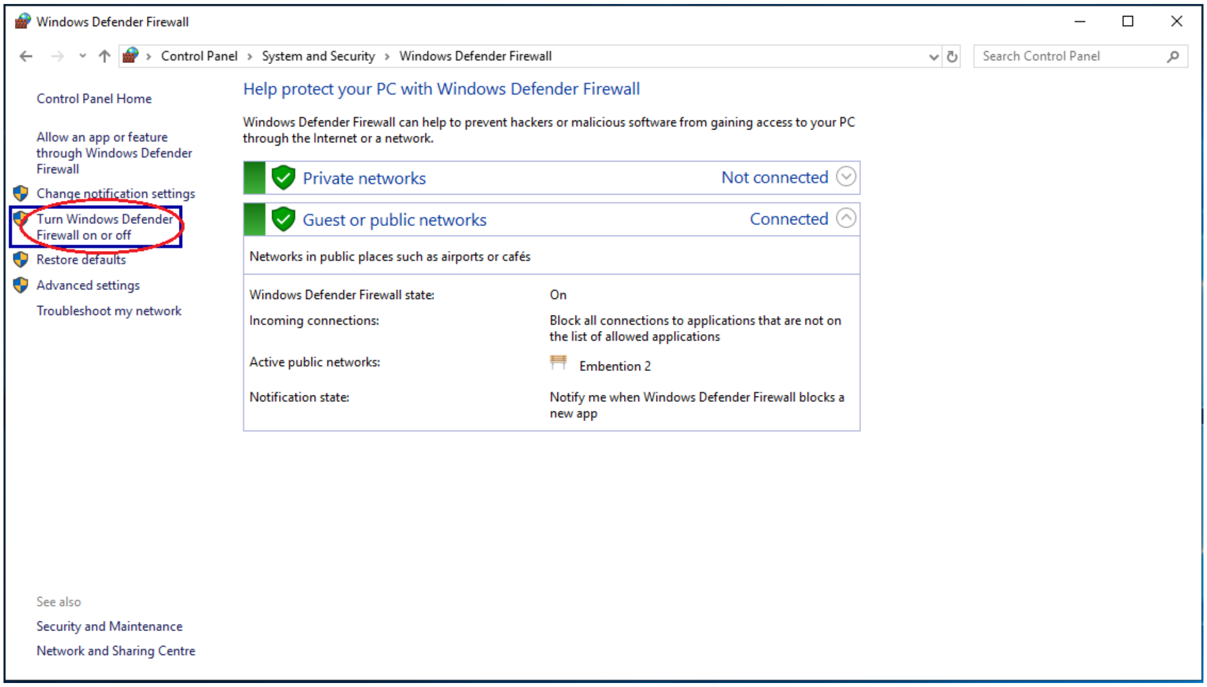
Windows Installation Step 02

3. Finally, click on Install

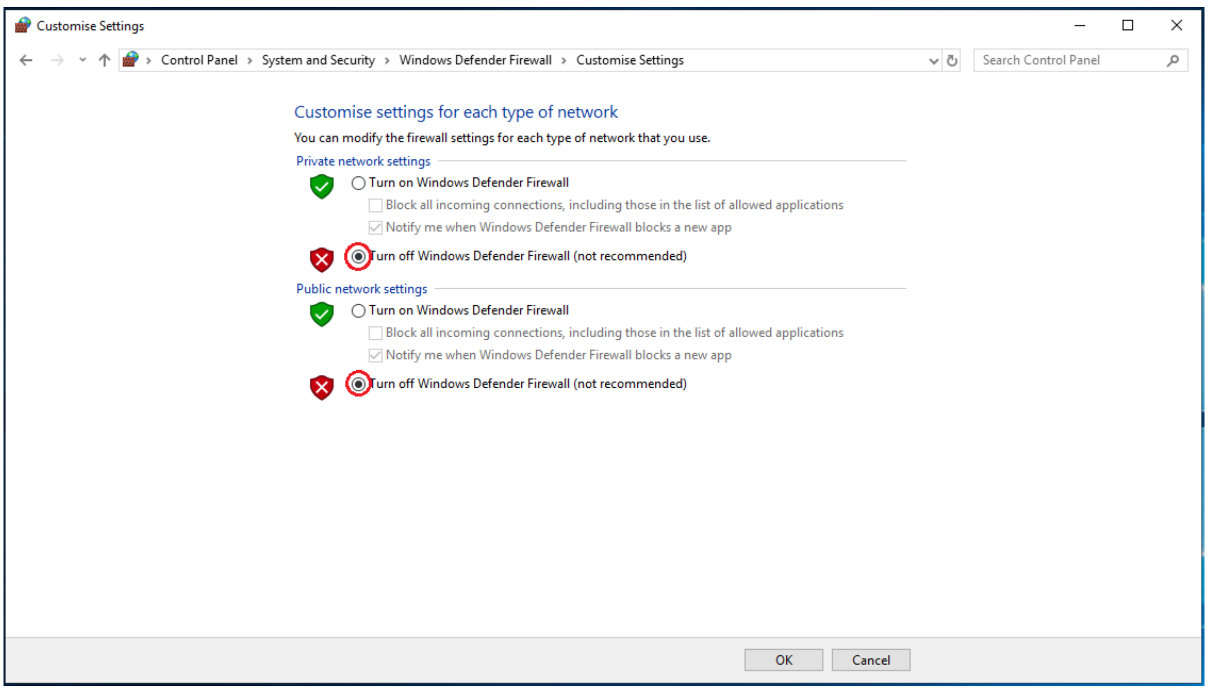


Windows Installation Step 03

Warning: If you have any problems with the installation, please disable antivirus and firewall of windows. Disabling the antivirus depends on your antivirus software. To disable the firewall, go to "Control Panel" and "Firewall of windows", then click on Turn on and turn off windows firewall and finally click on.



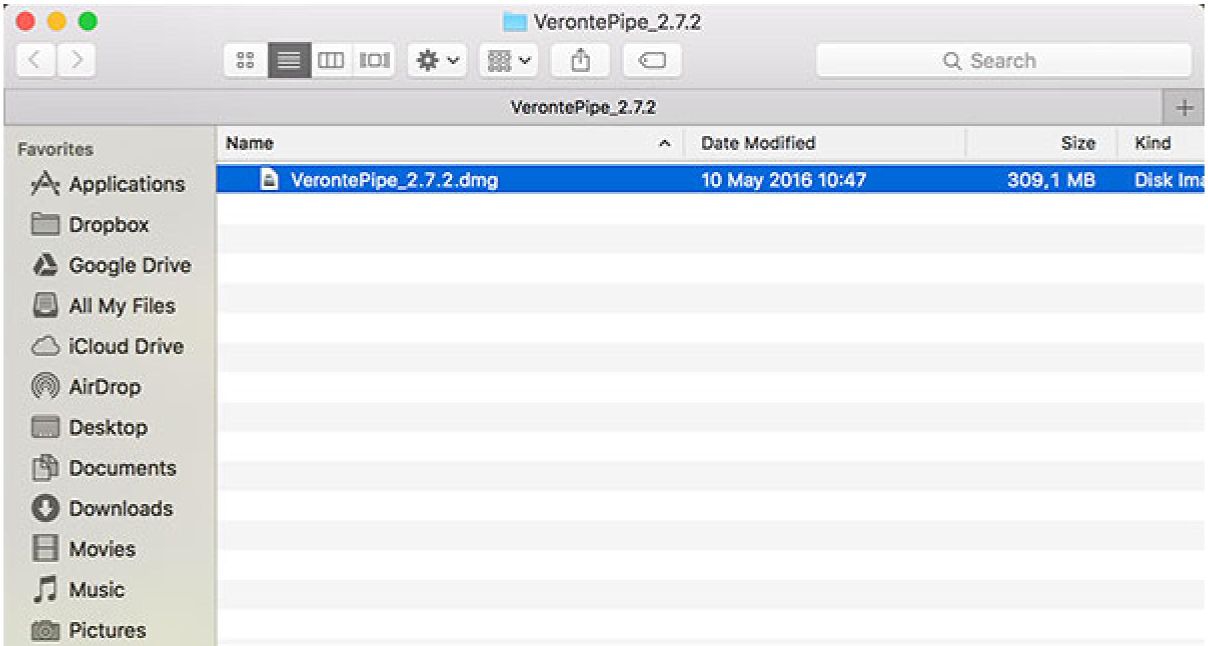
Turn on/off Firewall of Windows



Turn on/off Firewall of Windows

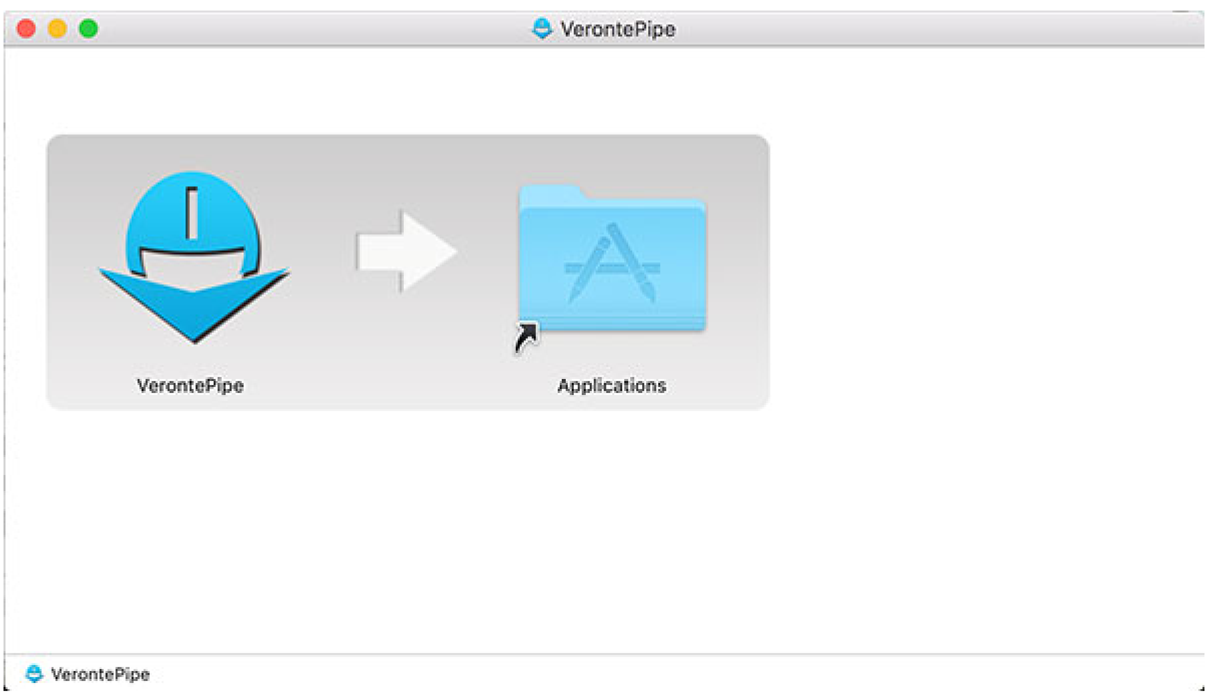
3.1.3 MAC

To install Veronte Pipe on Mac just double click on "VerontePipe_x.x.x.dmg" (where "x.x.x" is the version number), in order to mount the image.



Mac Installation Step 01

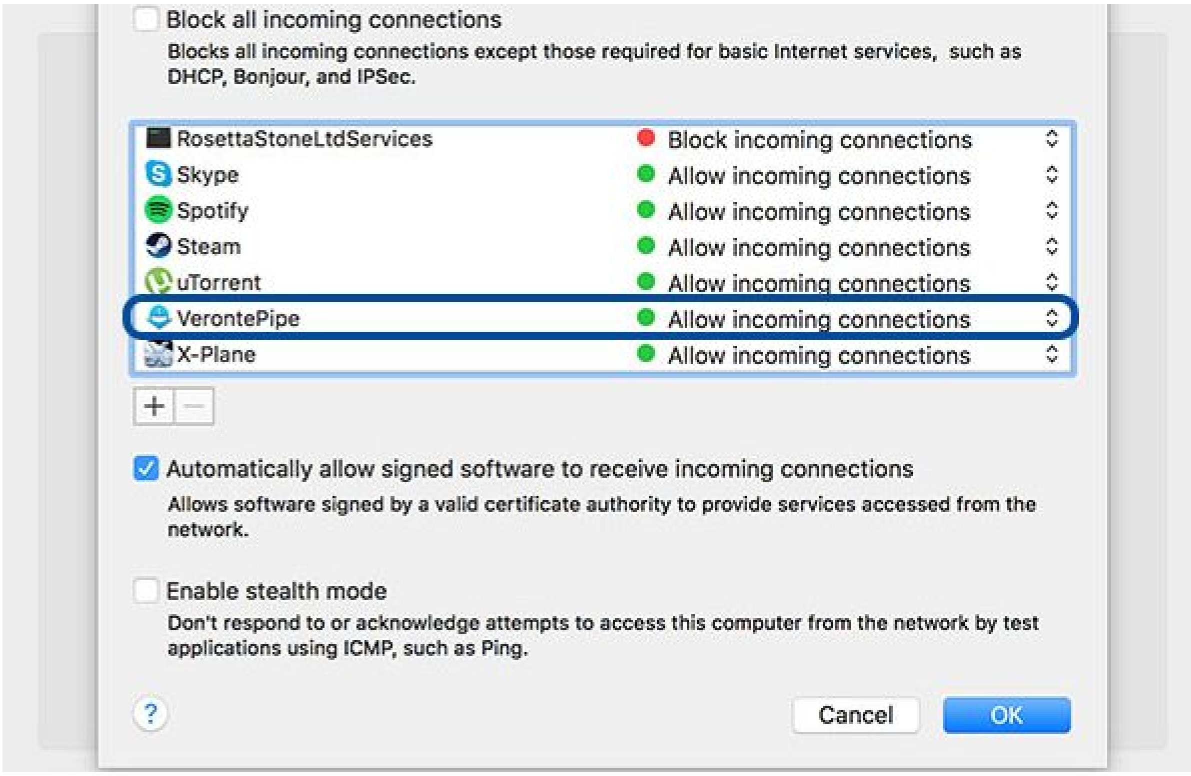
In the "VerontePipe" panel just follow the indications and move the Veronte Pipe application in the Applications folder to install it. The software is now ready to operate.



Mac Installation Step 02

Warning: If you have any problems during connection of VerontePipe, please check the firewall configuration of the application and disable it. To disable the firewall, go to "Security and Privacy" panel and be sure the program "allows incoming connections".

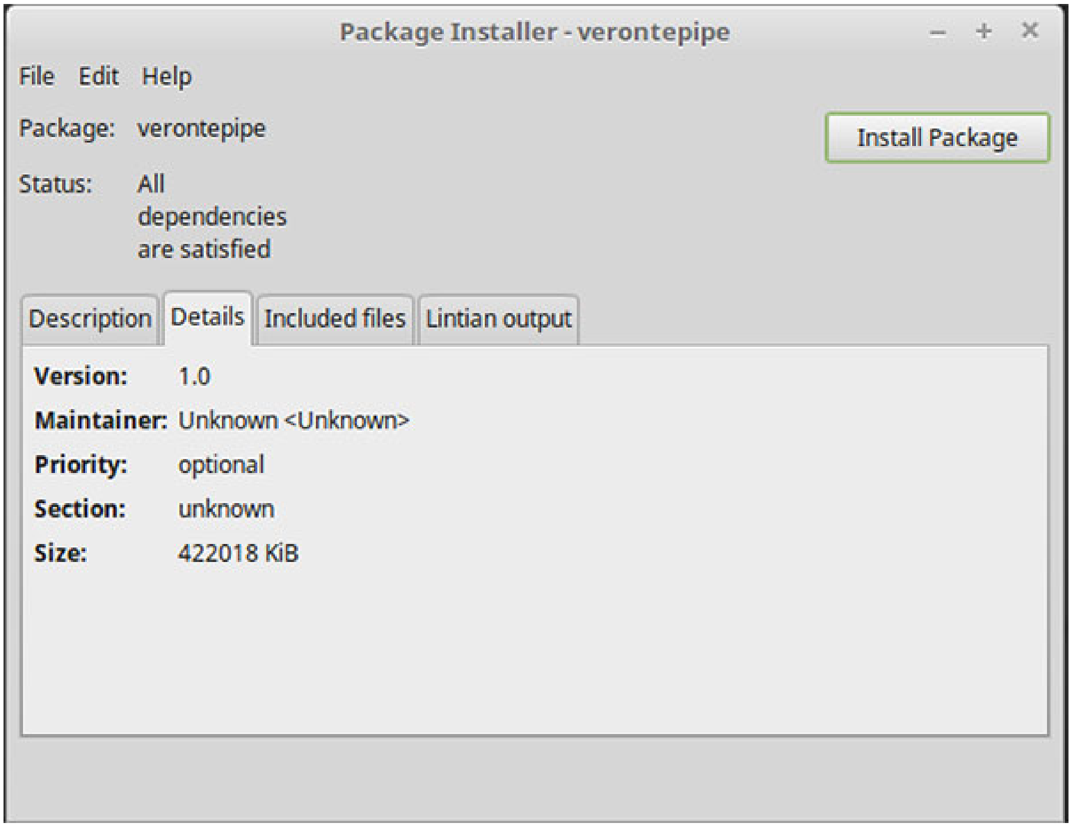
Deactivating the antivirus can be useful too. Disabling the antivirus depends on your antivirus software.



Turn on/off Firewall on Mac

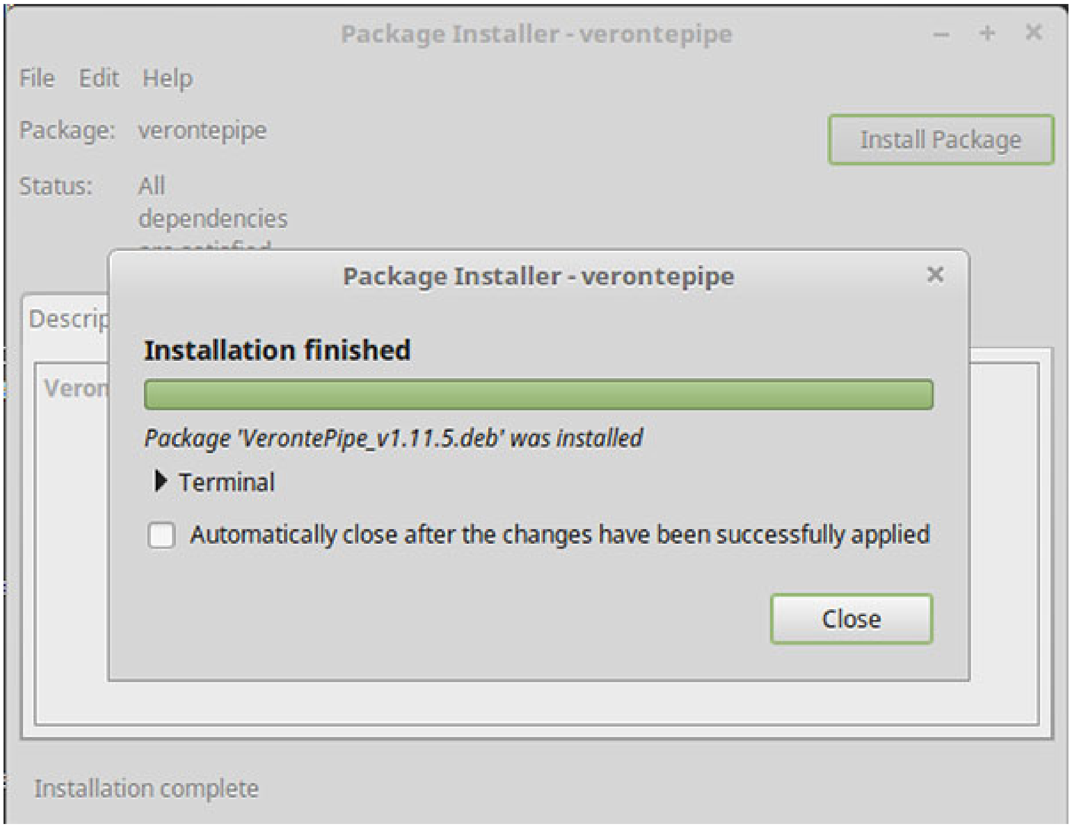
3.1.4 Linux

1. To install Veronte Pipe on Linux (Ubuntu in this case) just double click on "VerontePipe_x.x.x.deb" (where "x.x.x" is the version number), in order to open the Package Installer.



Linux install step 1

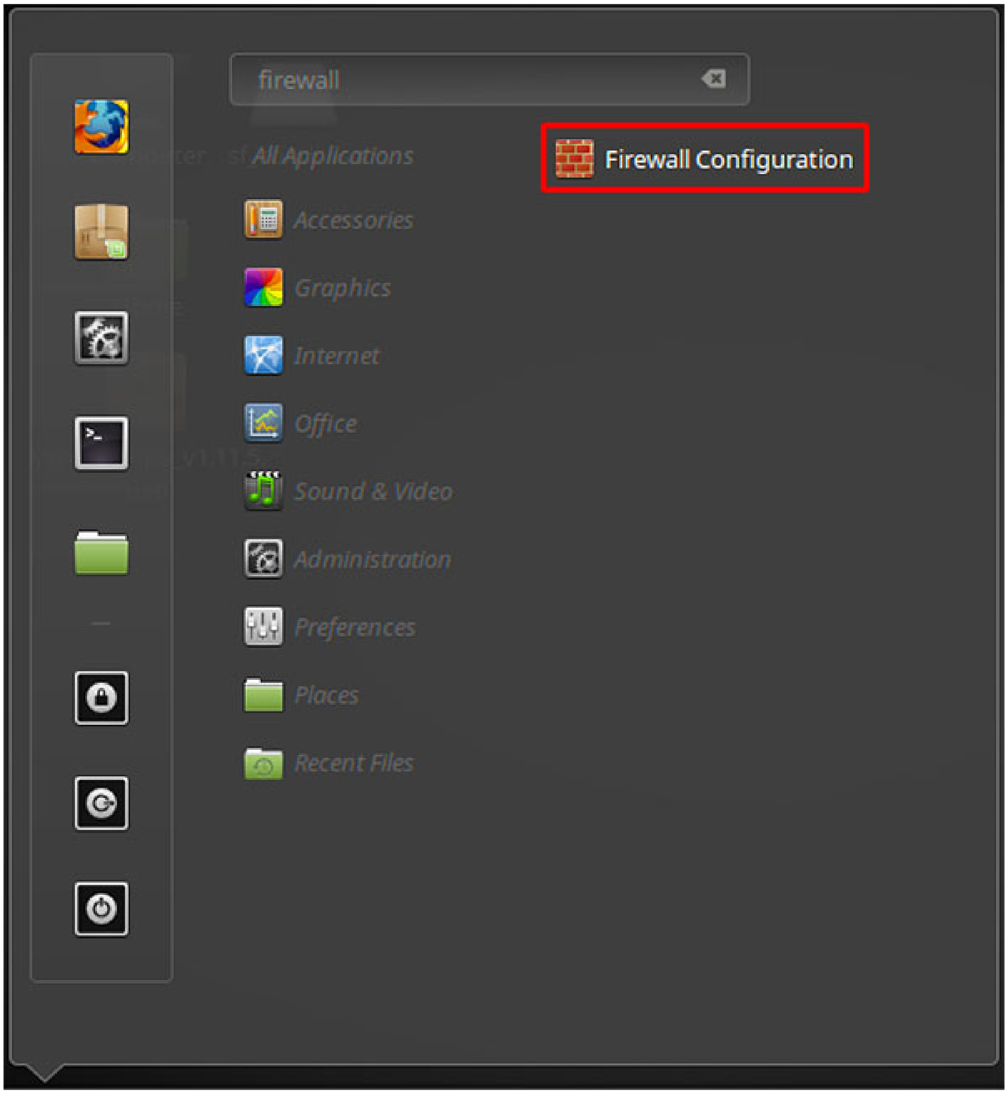
2. Then click on Install Package and wait the end of the installation process.



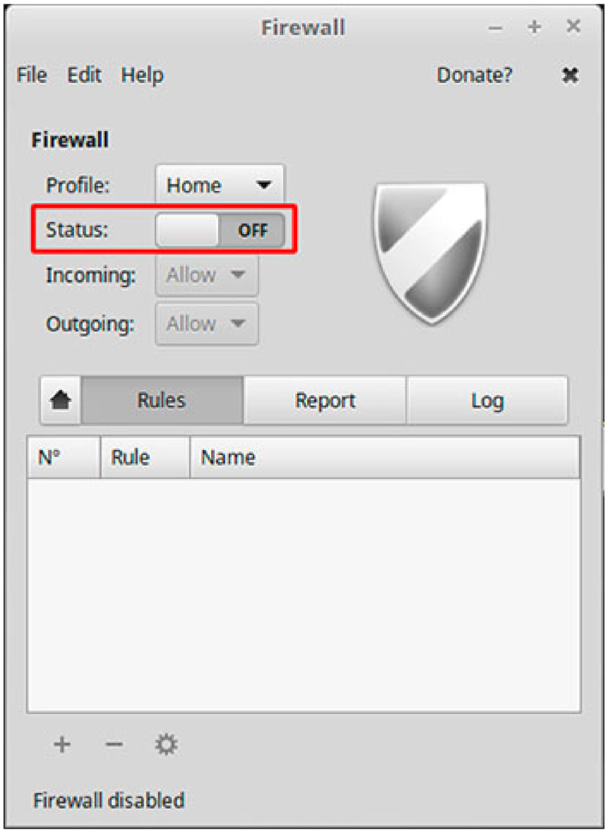
Linux install step 2

Warning: If you have any problems during connection of VerontePipe, please check the firewall configuration of the application and disable it. To disable the firewall, go to "Firewall Configuration" panel and be sure the firewall Status is set on "Off".

Deactivating the antivirus can be useful too. Disabling the antivirus depends on your antivirus software.



Firewall Configuration step 1



Firewall Configuration step 2

3.2 Update

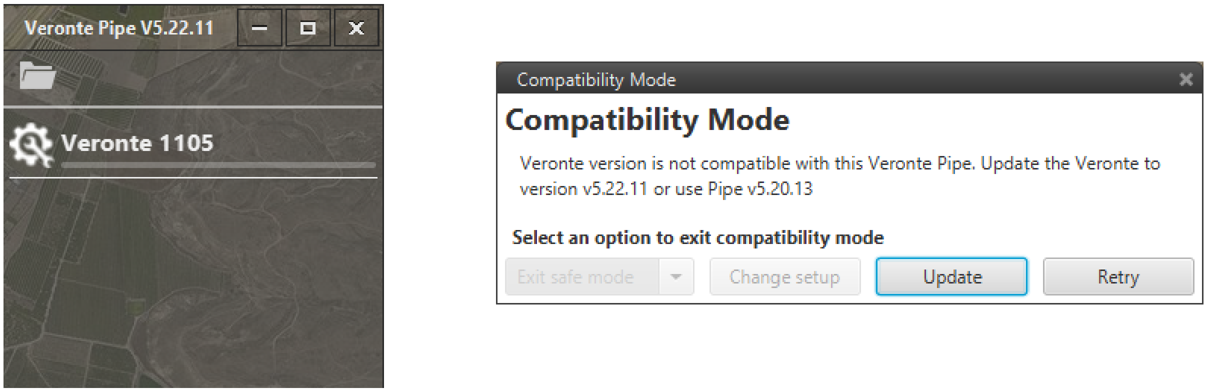
A setup wizard will be displayed in order to guide the user during the update process.

Warning:

- Although newer versions are usually compatible with older ones, when upgrading the system, updates must be done in the correct order. It is mandatory to update **Veronte Pipe** first, then **Veronte Onboard** and finally **Veronte on the Control Station**. Otherwise, part of the system could become unreachable.
- Never turn off Veronte during the update process. It could cause irreversible damage to the unit.
- When VerontePipe is updated be careful with managing .ver files. If the file was saved in a newer VerontePipe version, it will not be open in the actual one.

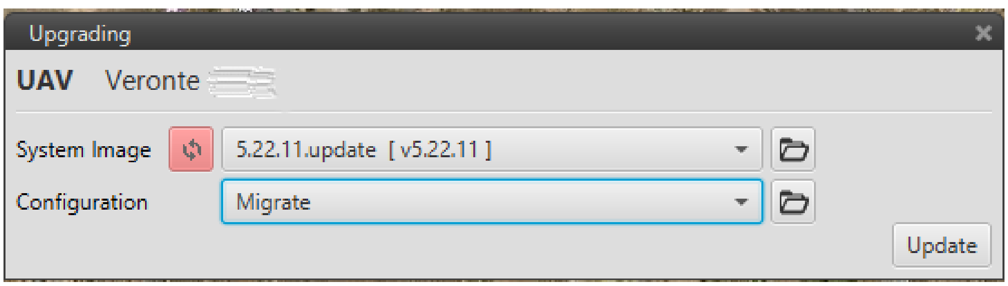
Current VerontePipe	.ver last saving	Compatibility
5.28.x	5.28.x	OK
5.28.x	5.22.x	OK
5.22.x	5.28.x	NO

When having the last version of Veronte Pipe installed in the computer, now is the time to update the software of the autopilots. In order to do that, click on the unit you wish to update from the sidebar. Then the window shown on the right of the following figure will appear.



Compabtibility Mode

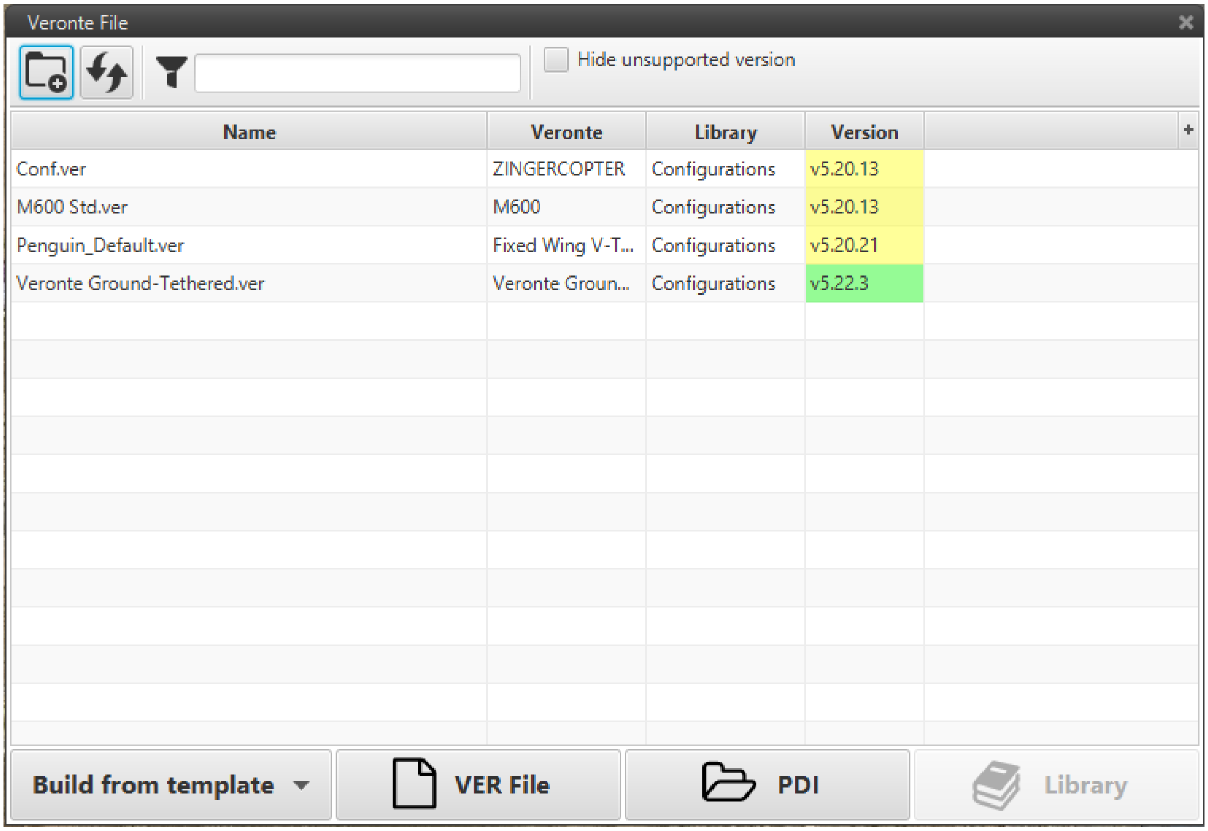
When clicking on Update, the next dialogue will appear on the screen.



Update Veronte Autopilot

User can select:

- **System image:** With internet connection, the upgrading process will download the last version of the onboard software (in this case v2.12.7). User can also select a different system image from the PC.
- **Setup:** Default option of the upgrading is the migration of the Setup. By clicking on the folder icon, user can open the following window and choose a way to upload a different Setup: Template, Select from PC or from Library.



Veronte File Selection

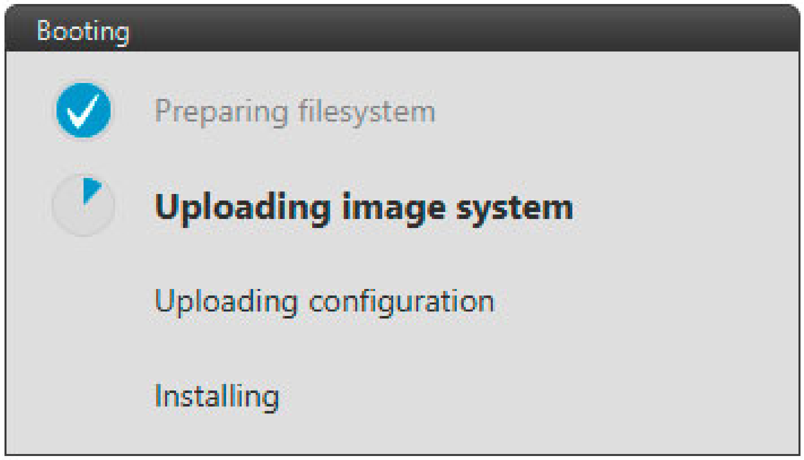
The previous window shows info about Pipe version used to save the configuration. Cell colour indicates the .ver compatibility:

- **Green:** Full compatibility.
- **Yellow:** Compatibility, but it needs migration.
- **Red:** No compatibility (.ver version is older than Pipe).

When the booting is completely configured, the user can launch the Update by clicking on the correspondent button and waiting until the end of the process.

Warning:

- During the update, the system will reboot so never perform an update during an operation.
- Make sure you choose the right Veronte Update file for the selected Veronte Autopilot.



Booting Process

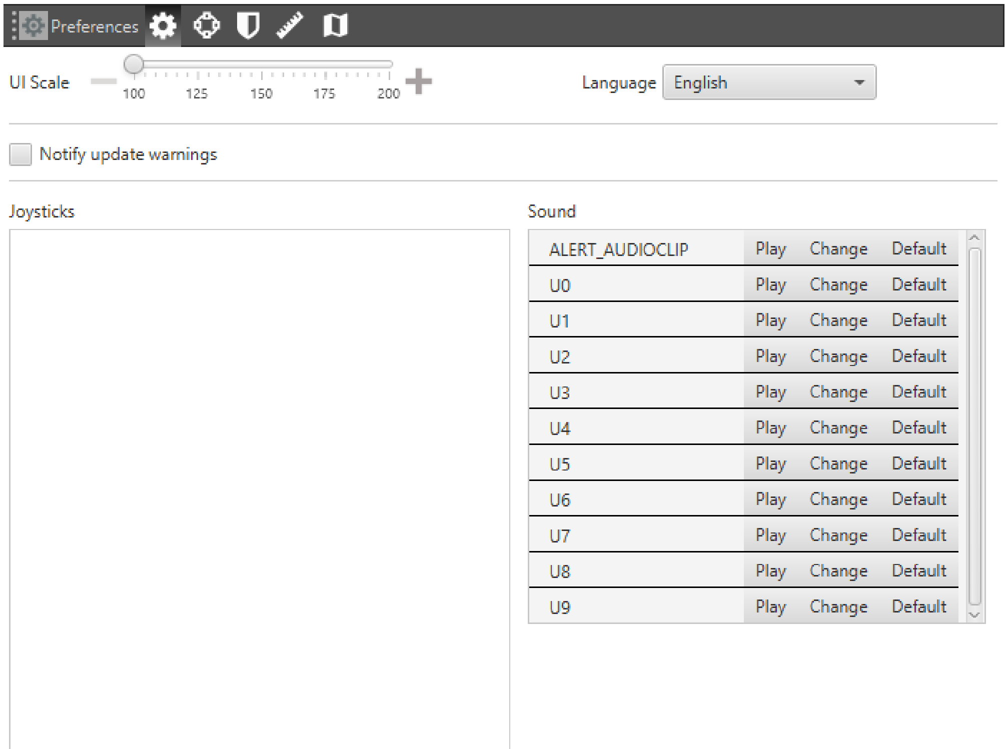
3.3 Preferences

3.3.1 General

If you click in:

General Tool

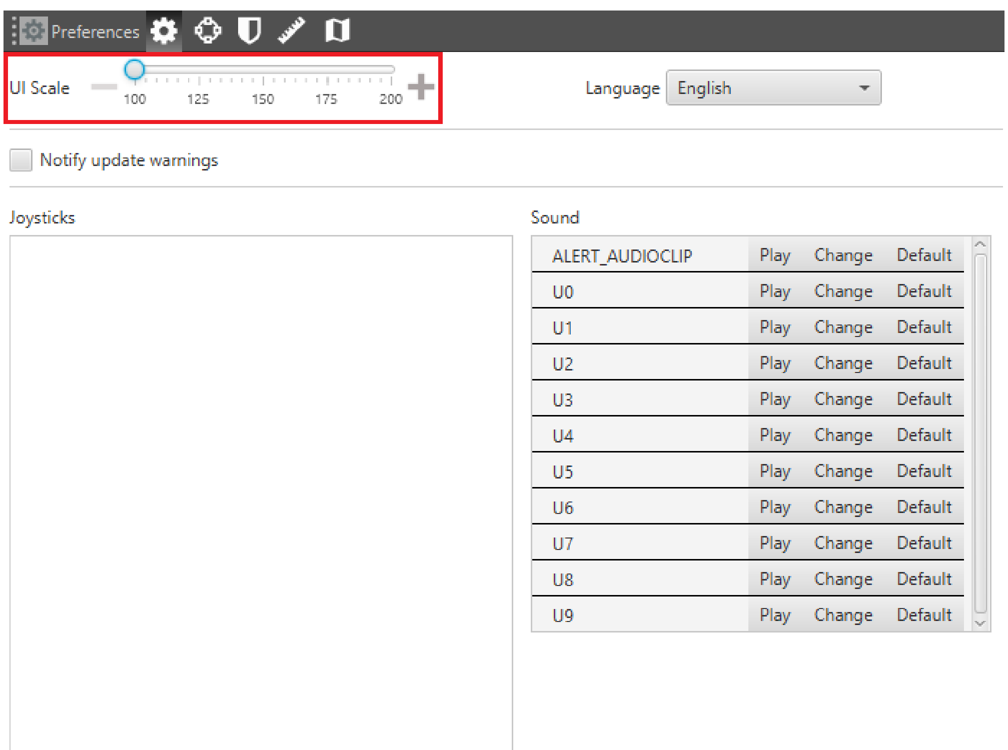
This menu allows the user to modify the general configuration for Veronte Pipe.



Software Setup

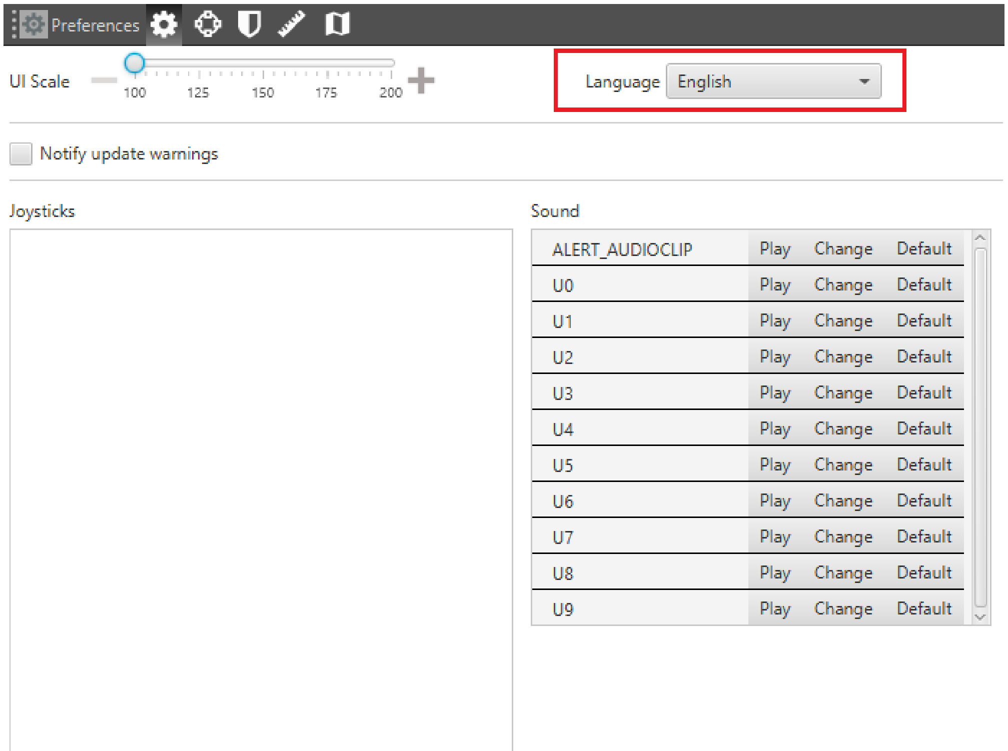
Where we will found:

- **UI Scale:** UI Scale permits to set the interface scale for adapting the application screen to the screen size on the system.



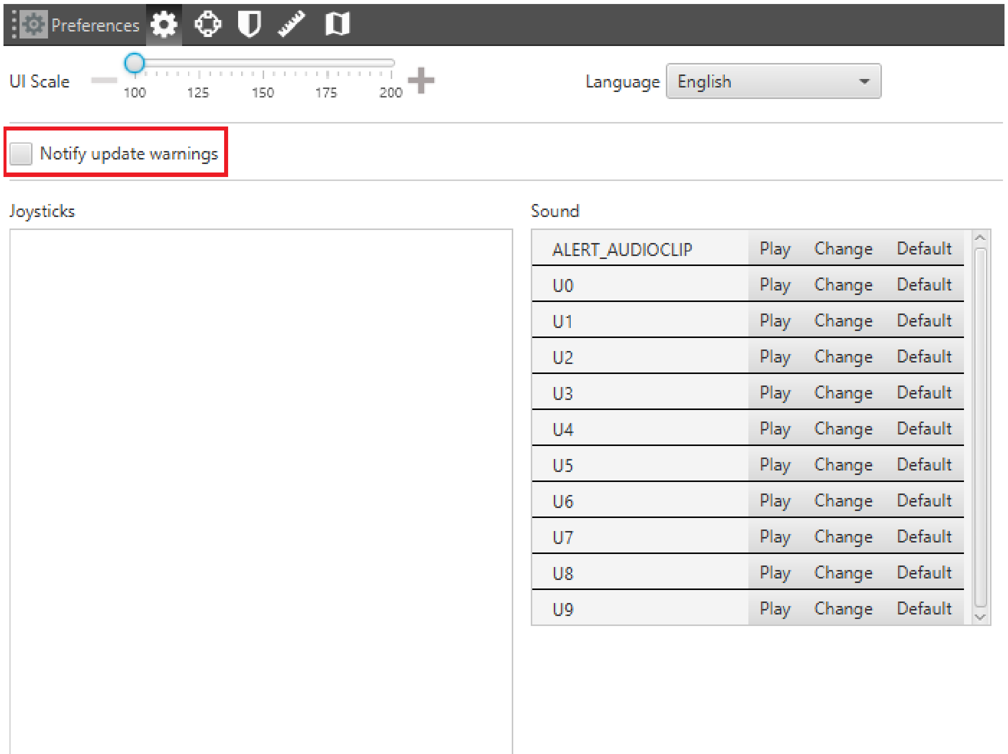
UI Scale

- **Language:** you can choose the language that you want.



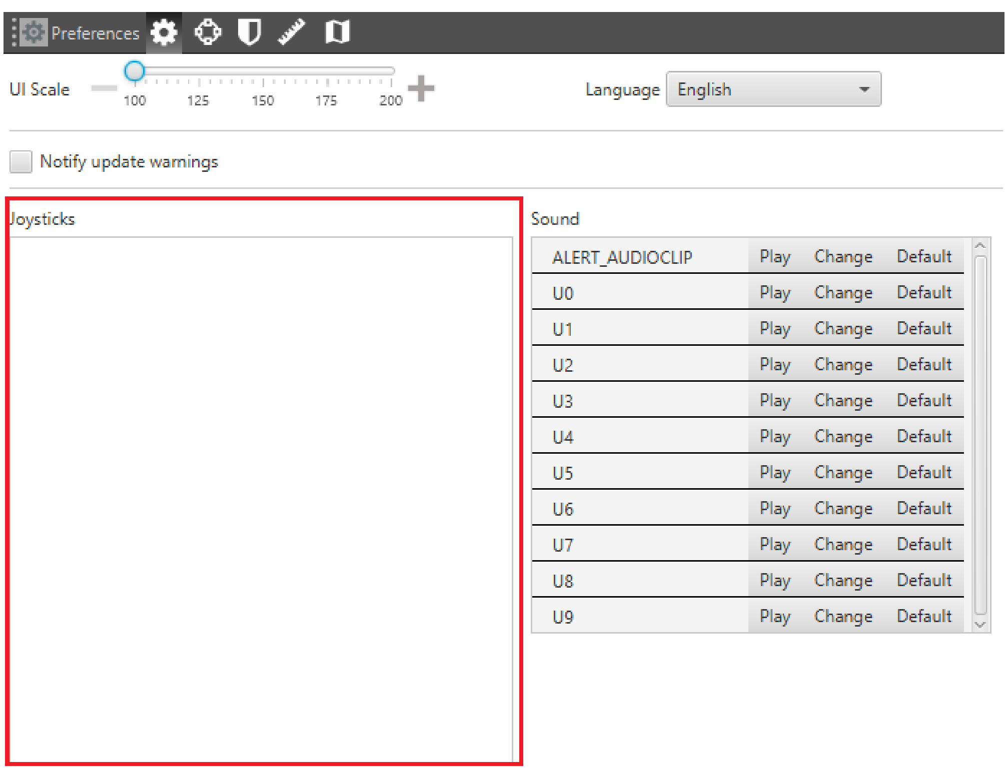
Language

- **Update notification:** the Notify update warnings checkbox allows the user to receive an alert when a new update has been released for Veronte.



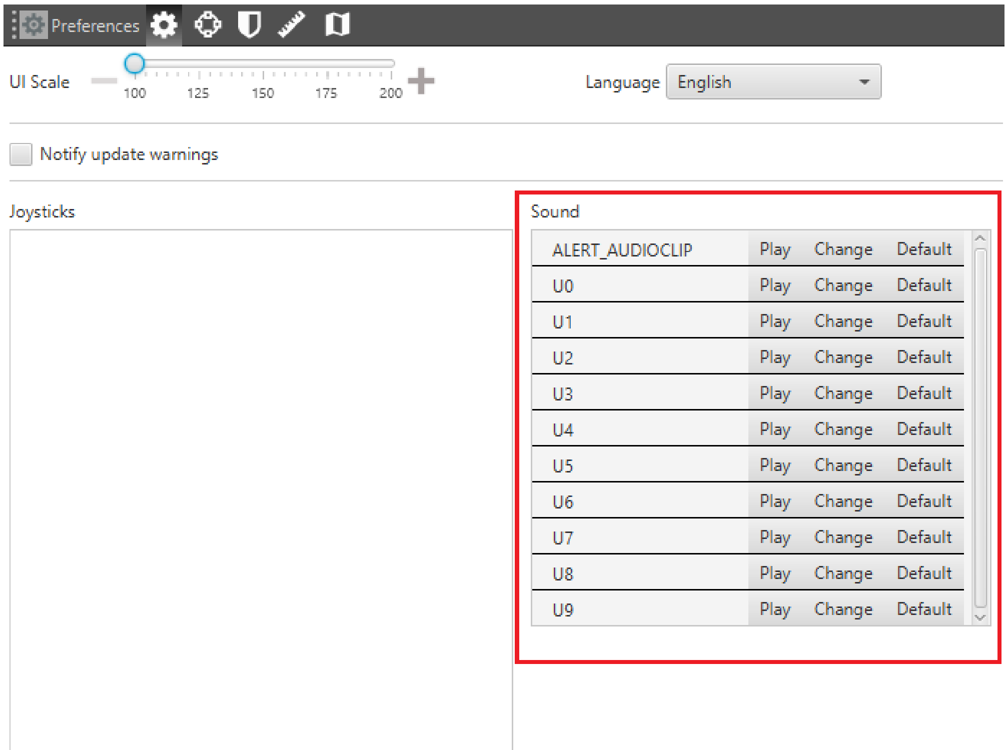
Update Notification

- **Joysticks:** this box shows you the joysticks that you have installed in your system.



Joysticks

- **Alert audioclips:** Alert Audioclips are used to manage audio files used on the application. They can be associated with system alerts on the Workspace configuration. There are nine audio files plus the alert_audioclip.
  - To substitute an audio file for another one, the Change button displays a browser to select a mp3/wav file stored in the computer.
  - It is possible to return to the standard alert by clicking Default.

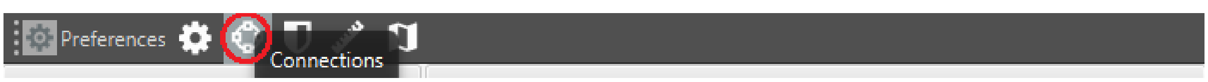


Alert Audioclips

To configure the alerts on a certain scenario visit *Gauge Display* on Workspace section.

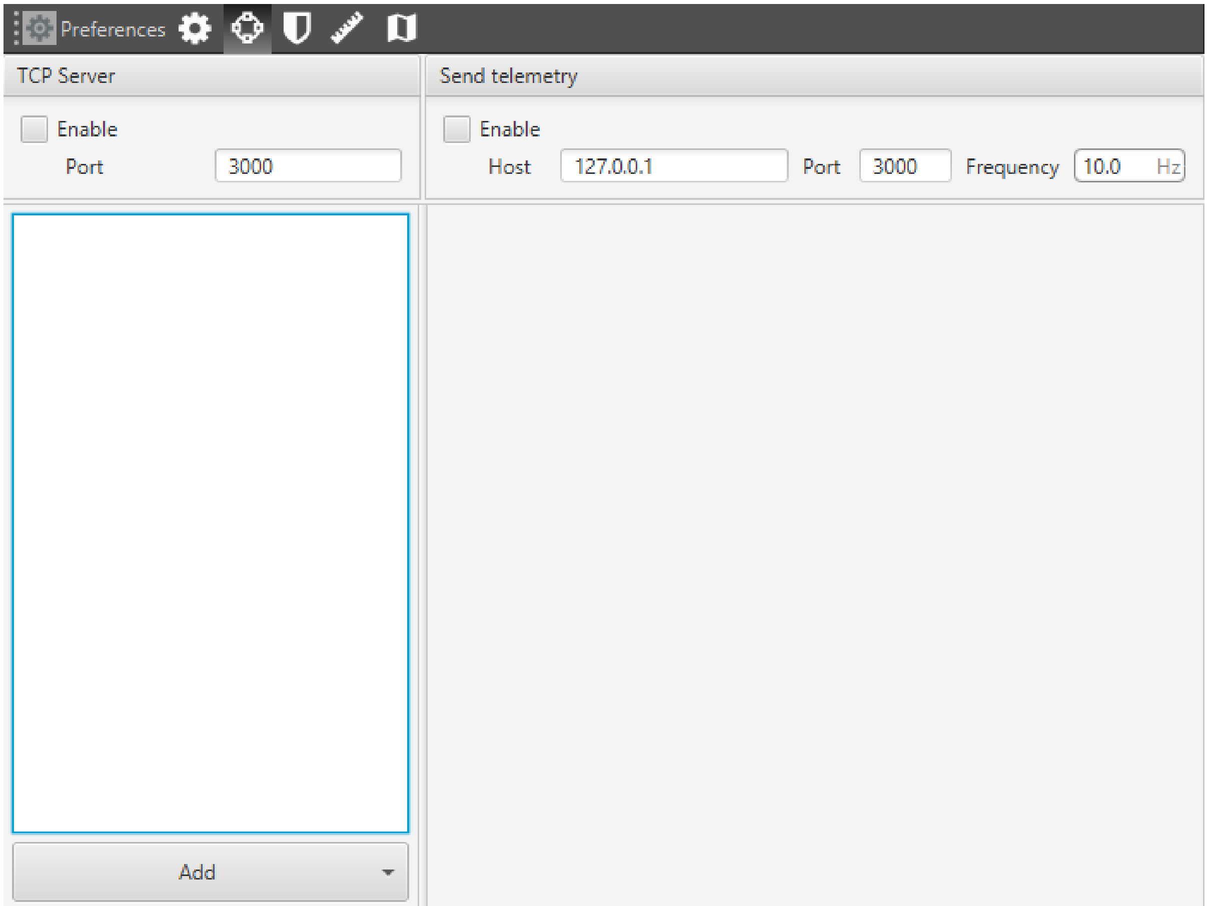
3.3.2 Connections

If you click in:



Connections Tool

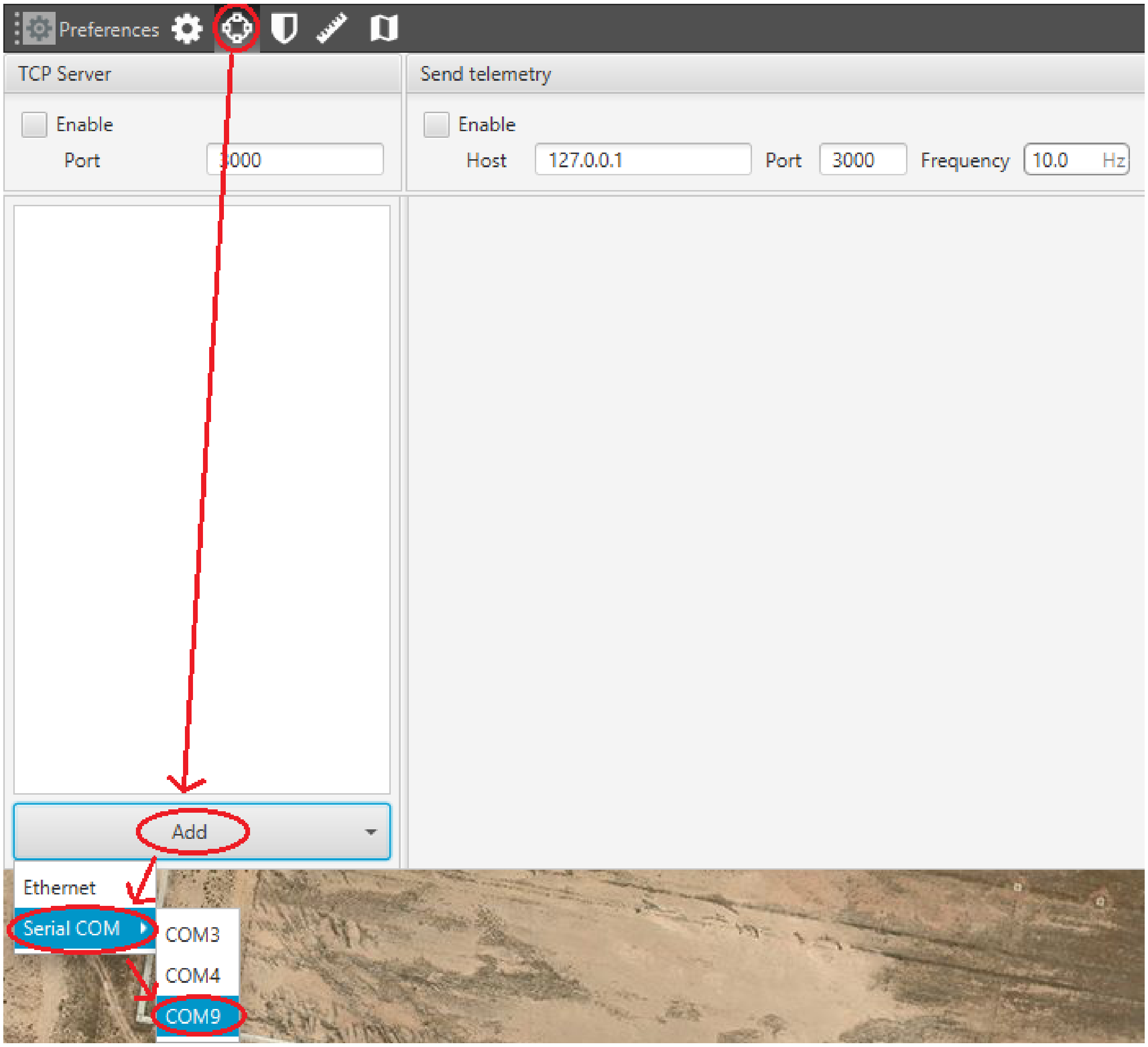
You can open the connections menu:



Connections Menu

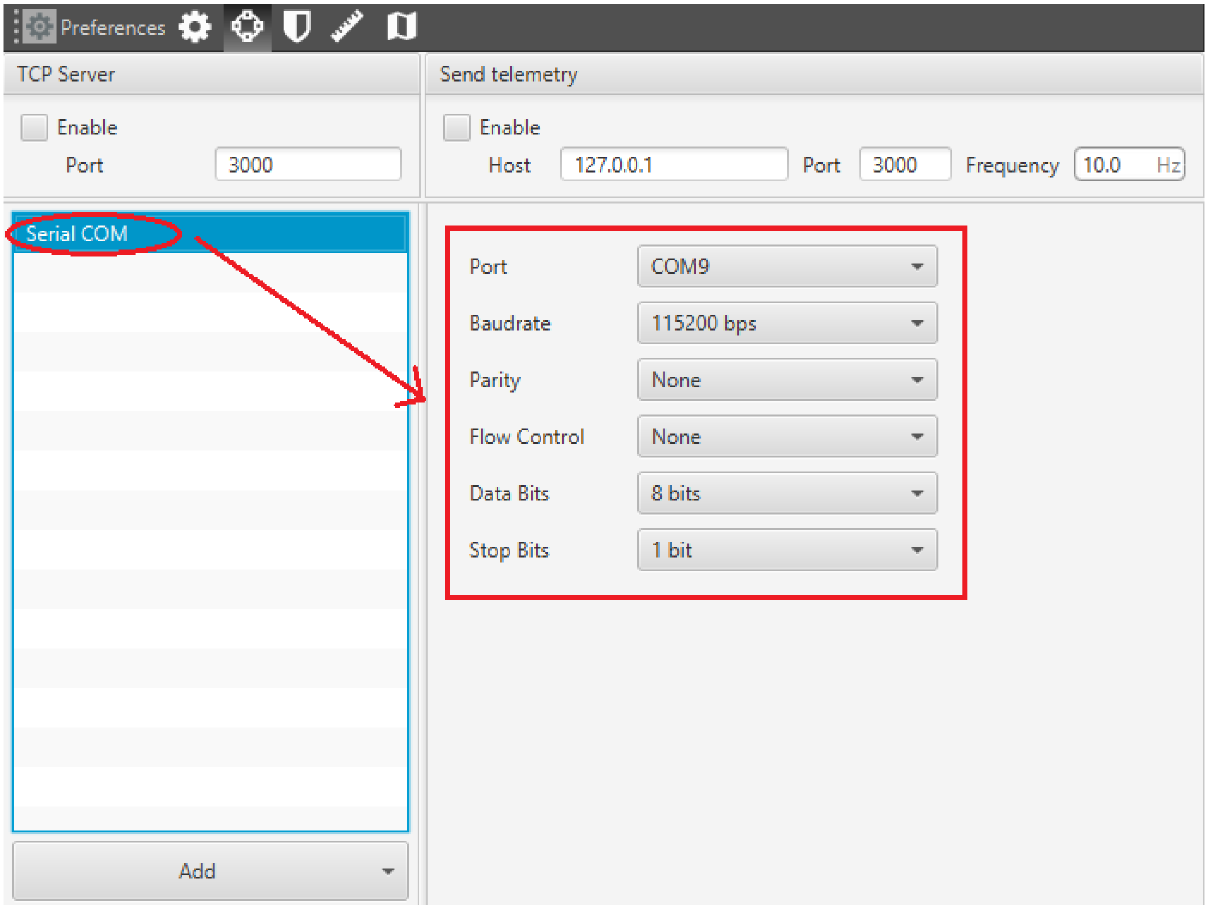
Where you can configure:

- **New veronte Units:** When using Veronte Pipe for the first time, it is necessary to configure the connection with Veronte Units. Once it has been done, Veronte Units will be automatically detected by Veronte Pipe.



New Veronte

In case of using a different interface of connection, it may require changing some parameters.



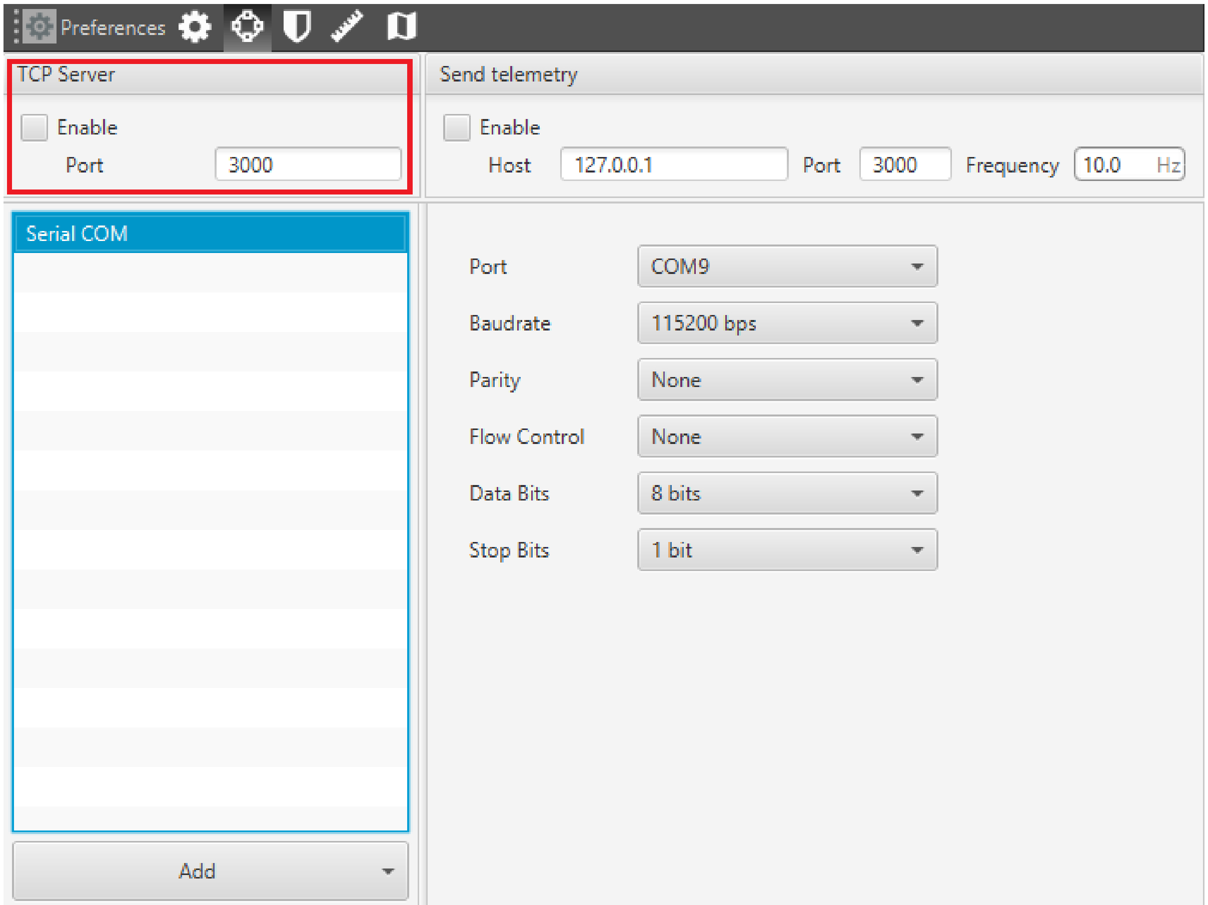
New Veronte Configuration

The following parameters can be edited in the menu shown above:

- **Baudrate:** this field specifies how fast data is sent.
- **Parity:** this field is a way of low-level error checking. It can be in odd or even.
- **Data Bits:** this field defines the bits number of the message.
- **Stop:** number of the stop bit.

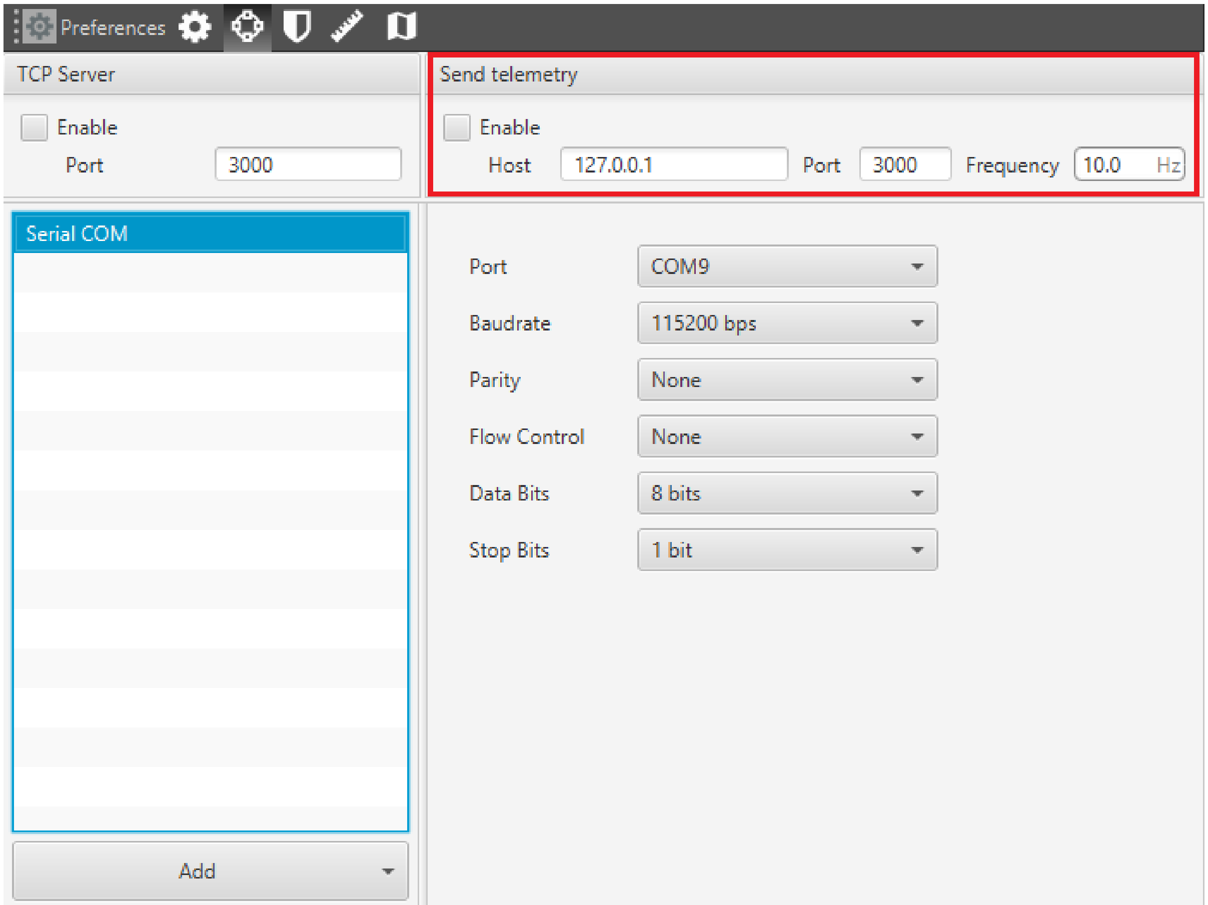
Type	Bits
Start	1
Data	4-8
Parity	1-2
Stop	1-2

- **TCP Server:** Pipe gets information from the configured Port of the TCP (Transmission Control Protocol) Server.



TCP Server

- **Send-telemetry:** Pipe sends the telemetry document through the configured parameters.



Send Telemetry

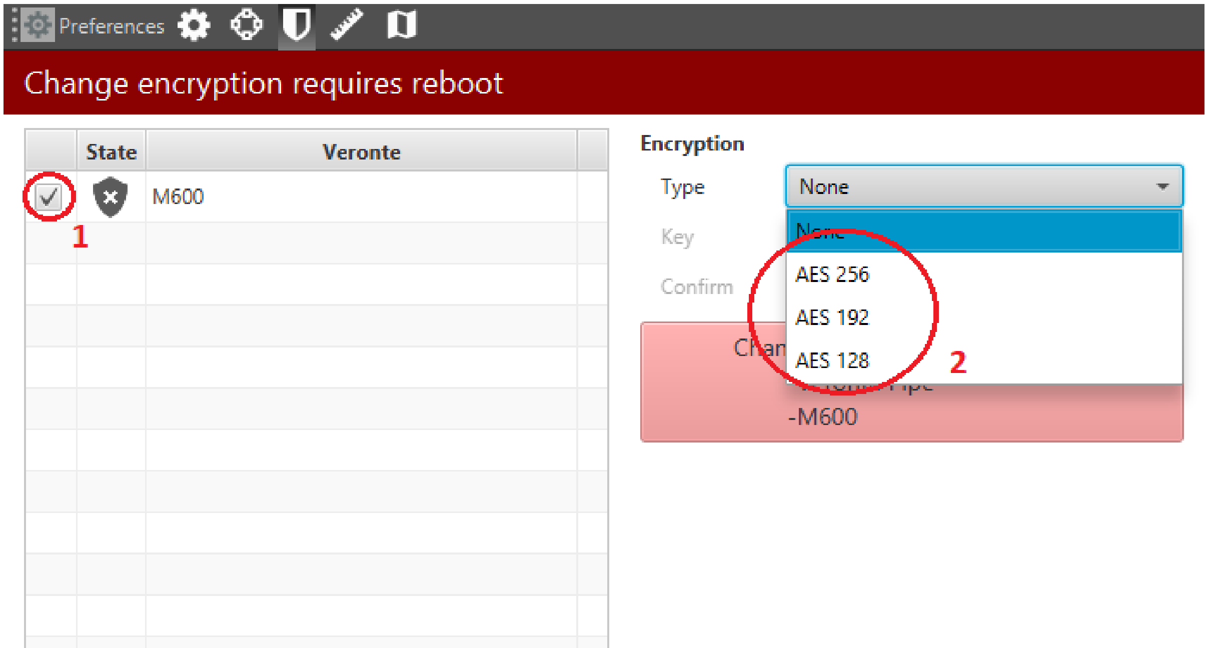
3.3.3 Encryption

If you click in:

Encryption Tool

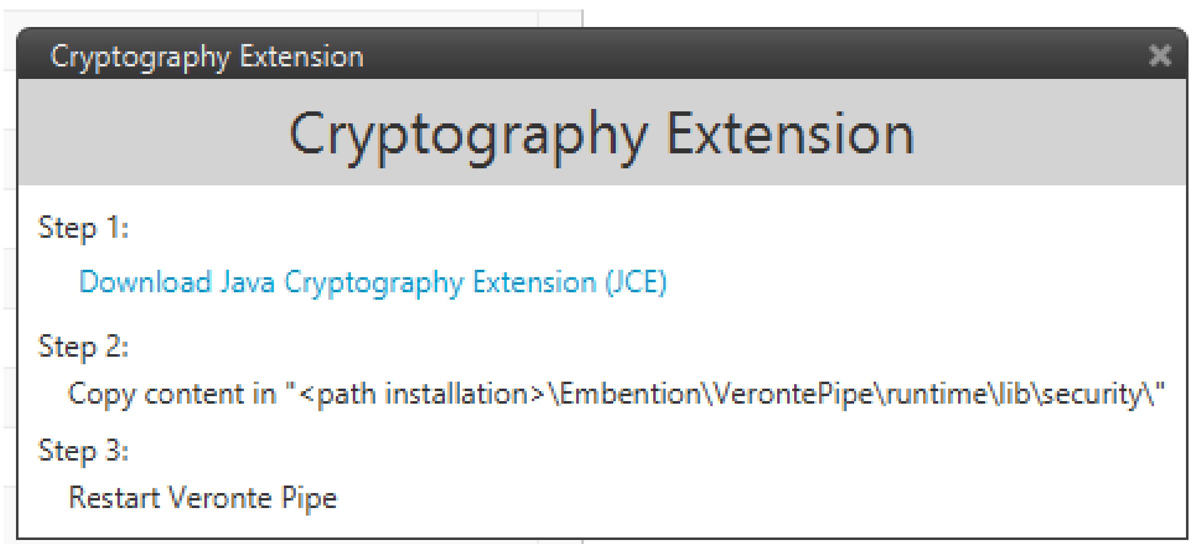
Veronte Pipe contains the option to encrypt the data transmitted from the autopilots and from Pipe itself. To configure this feature go to **Preferences** (Veronte Pipe) – **Encryption**.

1. Select the System that will be encrypted on the left side of the menu. By default, the unit set to encrypt is Veronte Pipe.
2. Select an Encryption Type. There exist three different encryption types, depending on the key size: 128, 192 and 256 bits. If one of the last two is selected, an additional step is required.



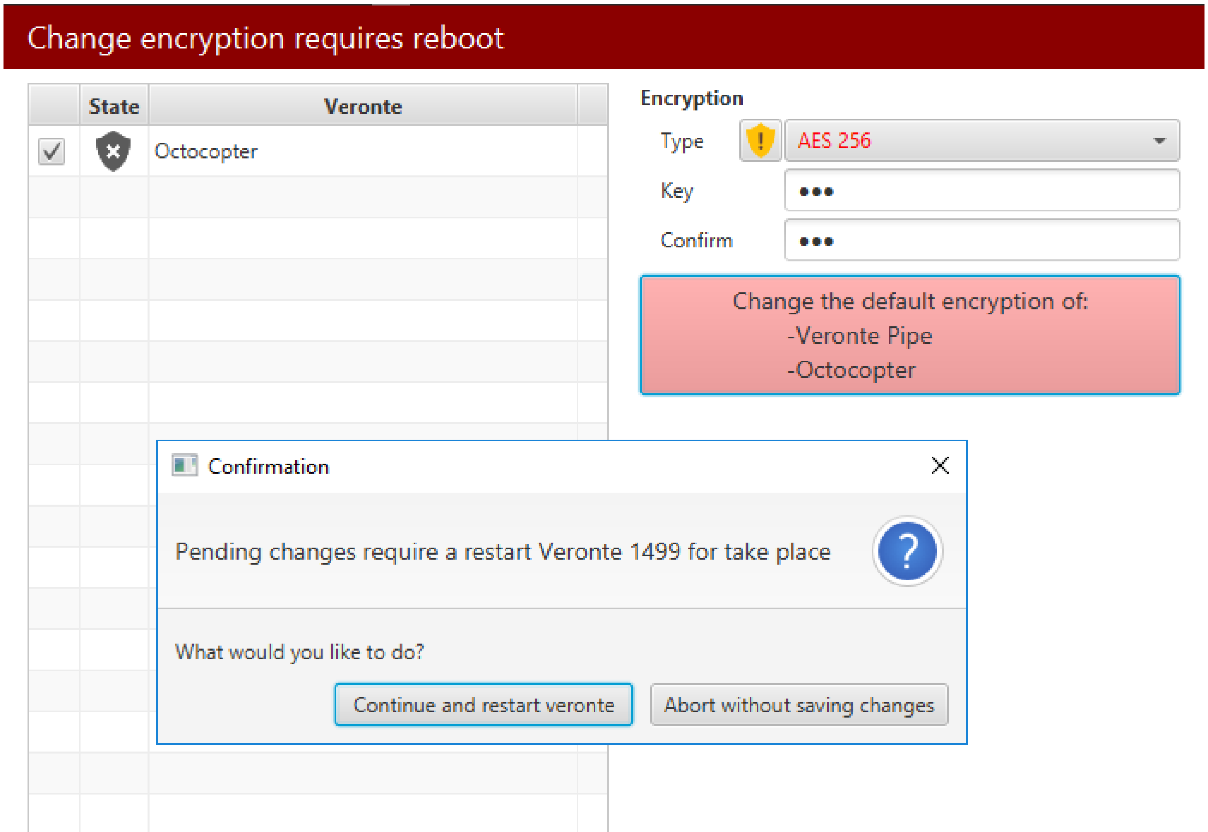
Encryption Menu

Warning: with 192 and 256 an additional step is required. Press The Shield to open the window that provides the instructions to ensure the functioning of AES (Advanced Encryption Standard).



Advanced Encryption Standard

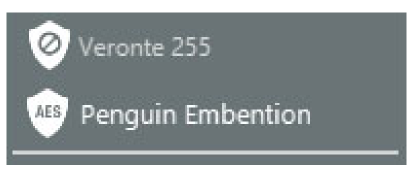
1. Specify a Key, this will be used to encrypt the data and can be used later to get the information back to a readable format.
2. Press the button Change the default encryption, a pop-up window will appear.
3. Restart the encrypted autopilots.



Veronte Restarting

Once autopilots are restarted they will show up in the right bar of Veronte Pipe. If both autopilot and Pipe are encrypted nothing will change in the view. If only the autopilots are encrypted (or the autopilots are connected to different encrypted Veronte Pipe), they will show up as shown in the following figure.

Both autopilots encrypted



Only Ground autopilot encrypted

If the autopilot icon appears with the label "Veronte X" (where X is the ID of the autopilot), it is not possible accessing to its configuration and user can not use it.

3.3.3.1 Remove Encryption

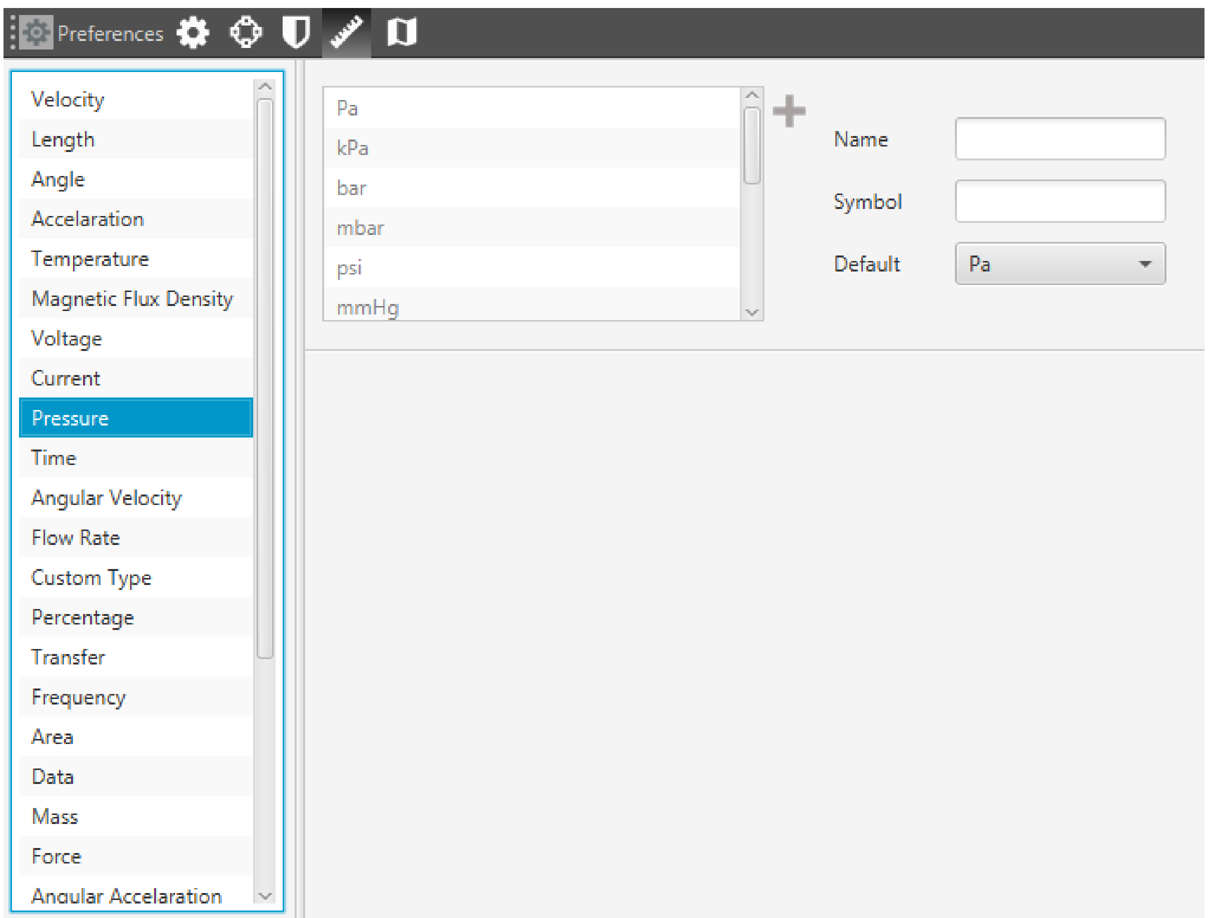
To remove encryption temporarily and be able to use the autopilot it is necessary to left-click on the autopilot label and insert the encryption type and password in the following window.

Remove temporarily encryption

To remove definitely the encryption it is sufficient to go in the Encryption Panel and selecting the encryption type None. The software will ask a last time for the password and, after the autopilots restarting, the encryption will be removed.

3.3.4 System Units

This panel shows all the system units available for the system variables. They are arranged according to the type of variable (velocity, acceleration, temperature, etc). It can be found in **Preferences** (Veronte Pipe) – **Units**.

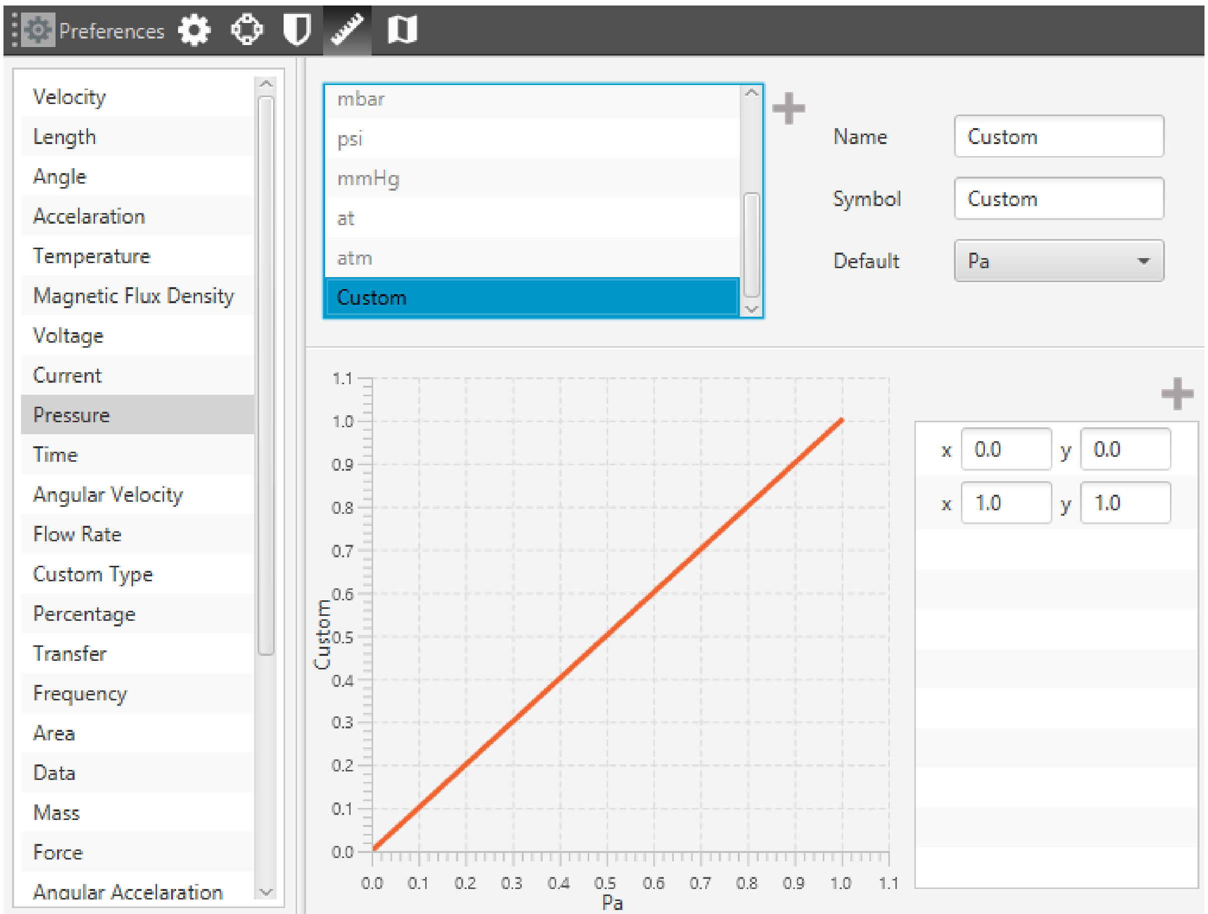


System Units

Veronte Pipe works by default with units of the International System, but these ones can be changed by selecting others from the ones shown when pressing Default option.

3.3.4.1 Creating Custom Units

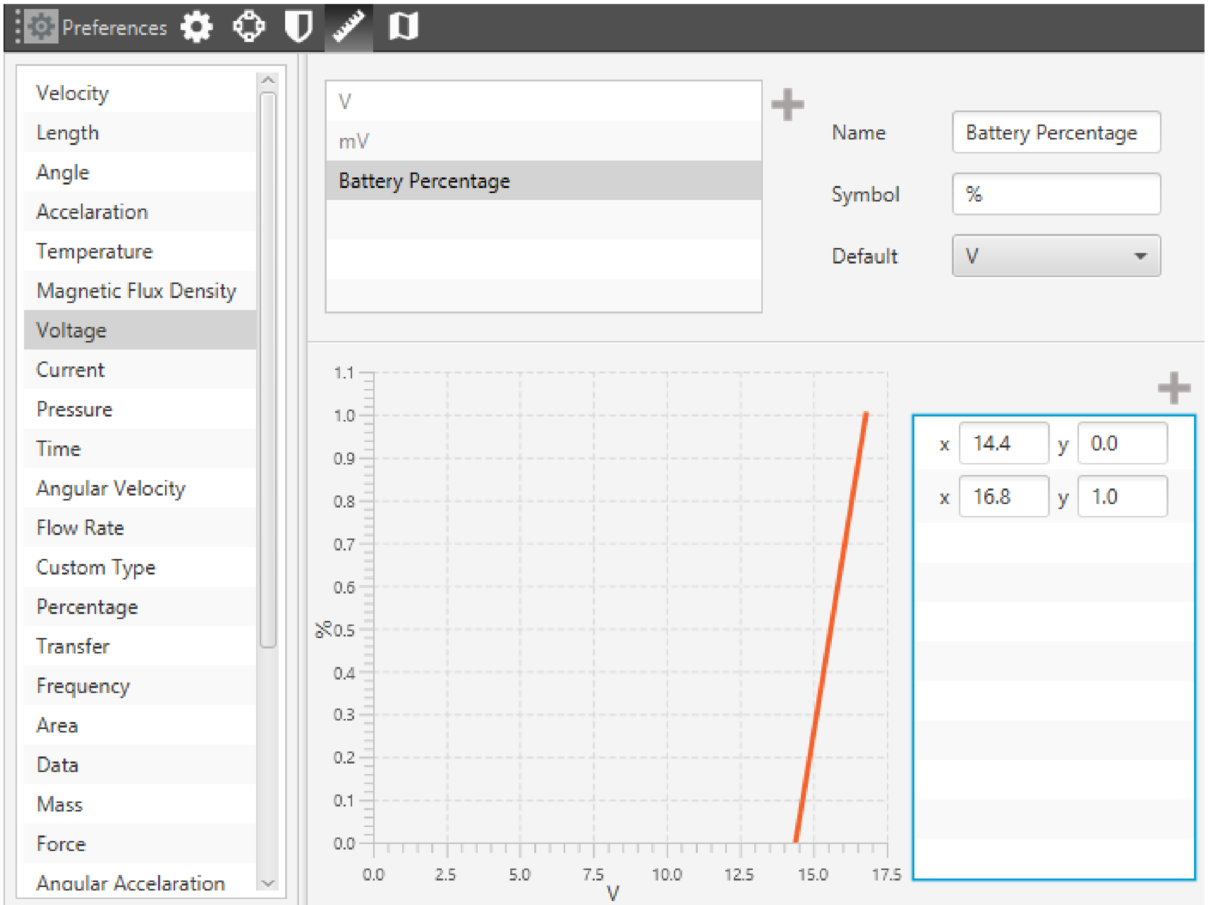
1. Click on "+" button, and a new variable will appear in the menu.
2. Introduce the conversion from the default unit to the new one. The conversion formula is created by introducing as many points as desired in the graph shown in the next figure.



Custom Unit Definition

An example is added to improve the understanding:

Considering the voltage of a LiPo battery, it could be useful to have a unit that indicates the remaining percentage of battery. For example, if the battery used is a 4S, the fully charged voltage is 16.8 V, while 14.4 will be the one when the battery does not provide the amount of energy needed to fly. So 16.8 V will correspond to 100% and 14.4 to 0%. Taking that into account the conversion graph will be the one shown in the next figure.



Voltage To Percentaje Conversion

> **Warning:** In this panel, it can be selected the default system units. It means the first displayed unit will be the selected one, but it will be possible to change it temporarily in all Pipe windows.

3.3.4.2 Units

The table below shows all the available units in VerontePipe.

Variable Type	Units
Velocity	[m/s] [kt] [km/h] [mph] [ft/s] [ft/min]
Length	[m] [km] [mi] [NM] [yd] [ft] [in] [cm] [mm]
Angle	rad[-;]°[-180;180] rad[0;2] °[0;360] [° ‘ “] [° ‘ “](N/S) [° ‘ “](E/W)
Acceleration	[m/s^2] [ft/s^2] [in/s^2] [g]
Temperature	[K] [°C] [°F]
Magnetic Flux Density	[T] [mG] [gauss]
Voltage	[V] [mV]
Current	[A] [mA]
Pressure	[Pa] [kPa] [bar] [mbar] [psi] [mmHg] [at][atm]
Time	[s] [min] [h] [s] [ms] [Time]
Angular Velocity	[rad/s] [rad/m] [rad/h] [rps] [rpm] [rph] [°/s]
Flow Rate	[m^3/s] [gal/s] [gal/h] [l/s] [l/h]
Custom Type	[- -]
Percentage	[x1] [%]
Transfer	[pkts/s]
Frequency	[Hz] [mHz] [kHz]
Area	[m^2] [cm^2] [mm^2] [km^2] [mile^2] [ft^2] [yd^2]
Data	[bit] [byte][KB][GB]
Mass	[kg] [g] [tonnes] [lbs] [oz]
Force	[N] [kN] [lbf] [pdl]
Angular Acceleration	[rad/s^2] [rad/m^2] [rad/h^2]
Baudrate	[Bd] [kBd] [MBd]
Pressure Variance	[Pa^2]
Magfield Variance	[T^2]
Velocity Variance	[(m/s)^2] [(cm/s)^2] [(mm/s)^2]
Numeral System	[bin] [octal] [dec] [hex]
Pressure Square Error Rate	[Pa^2/s]

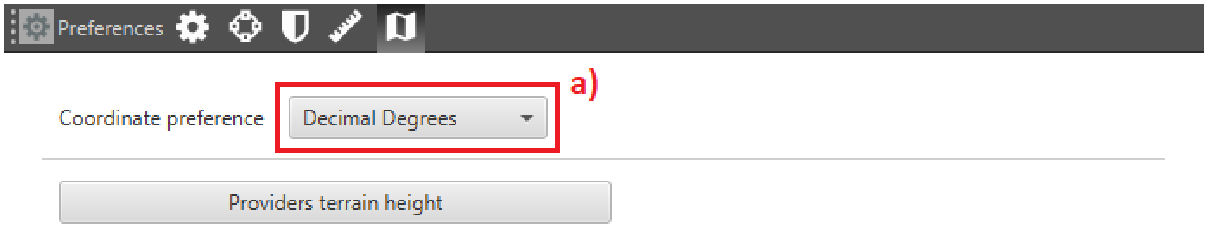
3.3.5 Map

If you click in:

Map Tool

In the map menu, the user can choose the coordinate preference (from MGRS, Decimal Degrees and Degrees) and download the Terrain Height.

3.3.5.1 Terrain Height



Map Menu

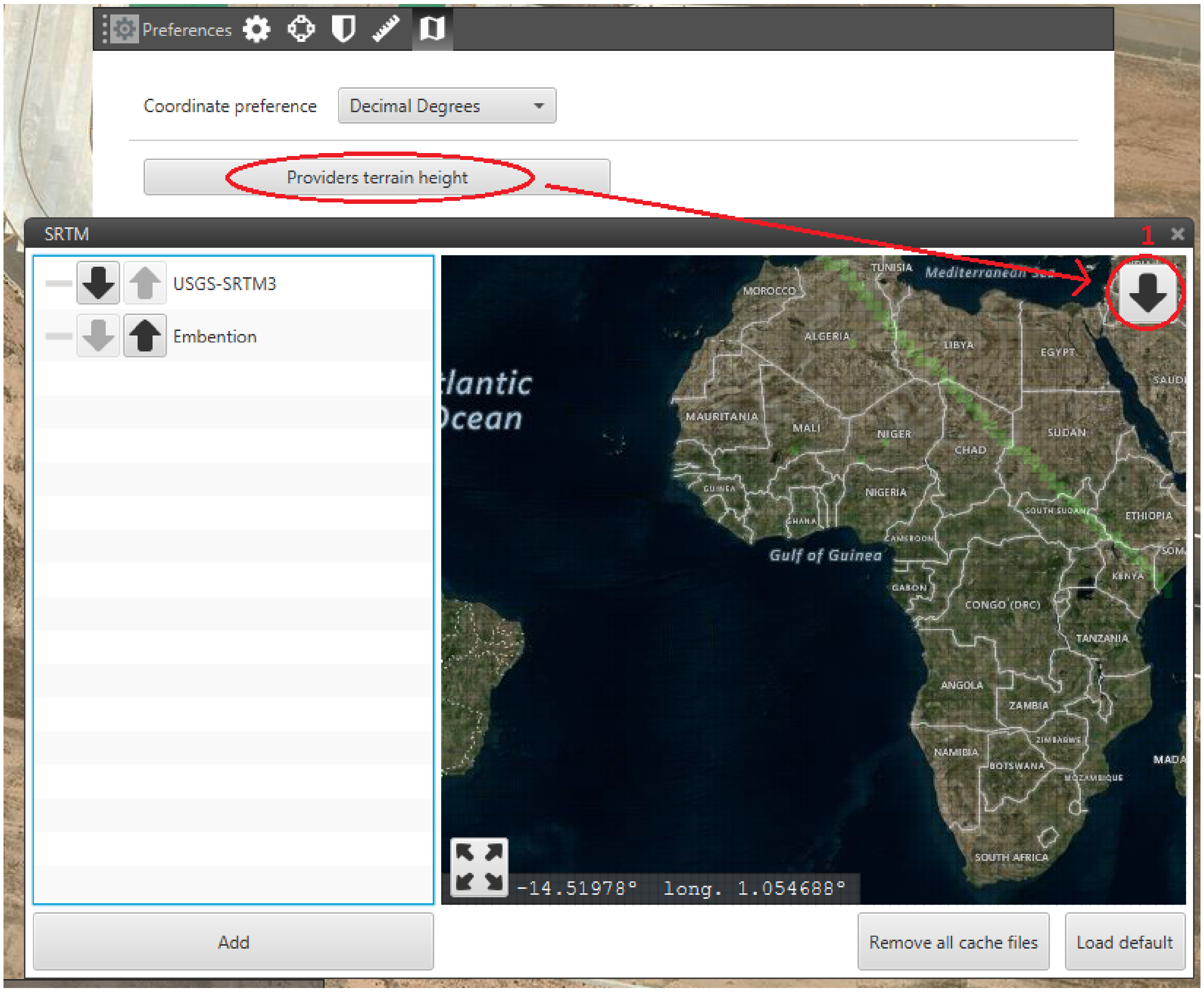
You can select the units of the coordinates clicking on **a)**

In order to perform the altitude estimation, Veronte installs a GIS (Geographic Information System) consisted of two meshes. When creating a mission, the terrain altitude on the mission area is automatically downloaded from the internet.

3.3.5.2 Download ALtitude Information

If the flight operation is carried out in an area where there is no internet connection, the altitude information can be downloaded a priori. To download the information:

1. Select **Providers terrain height** and press button **1**.



Terrain Height

1. Click on a quadrant to download the information of that zone into the system.

There are two providers of altitude information in Pipe:

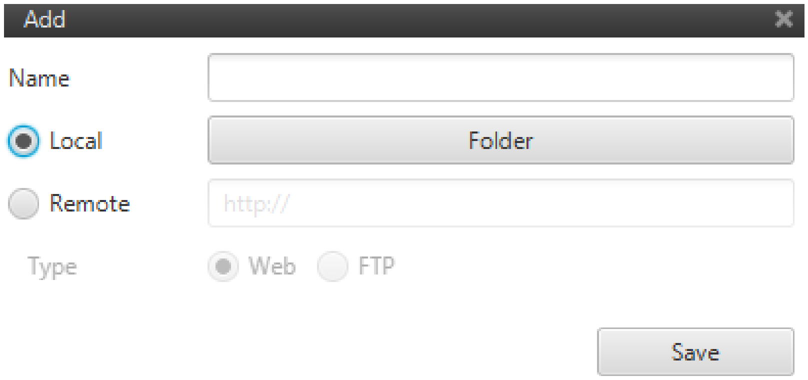
- The first one is the **SRTM3** (shuttle radar topographic mission 3) which contains the altitude information for the territories outside the United States (the number 1 is for USA).
- The other one is a server created by **Embention** with altitude information of areas that do not appear on the SRTM such as Iceland.

3.3.5.3 Custom Terrain File

Pipe also provides the option of introducing a custom terrain file.

Clicking on **Add** appears a menu with two options to charge a terrain file in the system:

- **Local:** allows the user to select a file from the computer.
- **Remote:** gives the option to introduce a URL (Un*iform Resource Identifier) of an altitude provider from both a web or FTP (File Transfer Protocol) server.



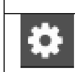
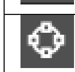
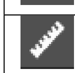
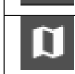
Custom Terrain Altitude Source

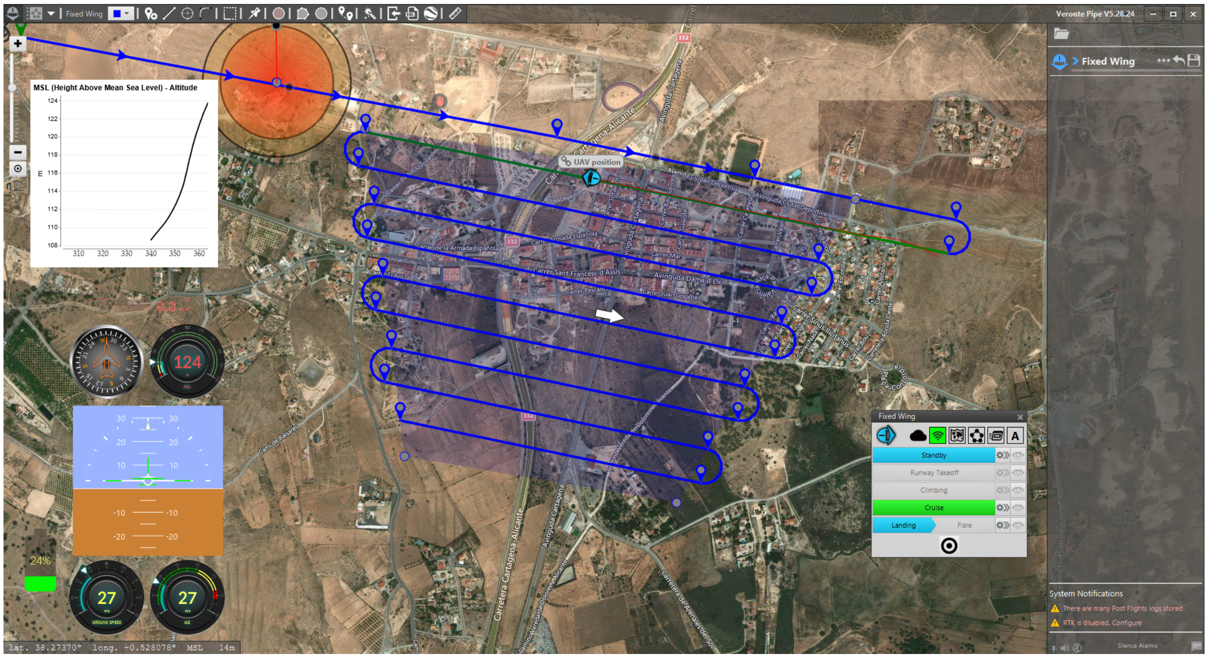
This section contains all preferences settings for Veronte Pipe (software). To access them, the user has to use the Preferences Toolbar.

Click on  (Main Menu), and then select Preferences in the pull-down menu, a new toolbar with different options will appear.

Preference Toolbar

The different elements of the setup toolbar are detailed in the following table.

	Item	Description
	General	Displays general configurable fields
	Connections	Open connections configuration
	Encryption	Displays encryption configuration
	Units	Open system units configuration
	Map	Coordinate preference and height terrain providers



Veronte System Overview

Veronte Pipe software is designed to operate any unmanned vehicle using Veronte as the onboard autopilot. With Veronte Pipe, users have an easy-to-use application with real-time response and telemetry with safety features.

It has been developed using software standard model of IEEE STD 830-1998, Recommended Practice for Software Requirements Specifications (SRS) and STANAG 4671 documentation, subpart about UAV Control Stations adapted to Veronte system.

Veronte Pipe includes:

- **Telemetry:** Real-time onboard UAV metrics, such as sensors, actuators and control states.
- **Telecommand:** Support for all synchronous operator control commands that can be sent to the flight segment, e.g. operational mode switch, mission management, payload control.
- **Mission design:** User defined, predefined mapping and drop missions configuration as well as inflight mission edit .
- **Mission analysis:** Post-flight viewer, reproduce all recorded data from a previous flights and generate plots and reports.
- **Configuration:** Edit vehicle settings, such as servo trim, interface/port management, modes and automations.
- **Multiple Users:** One or more operators can work simultaneously .

CHAPTER

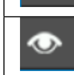
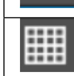
FOUR

WORKSPACE

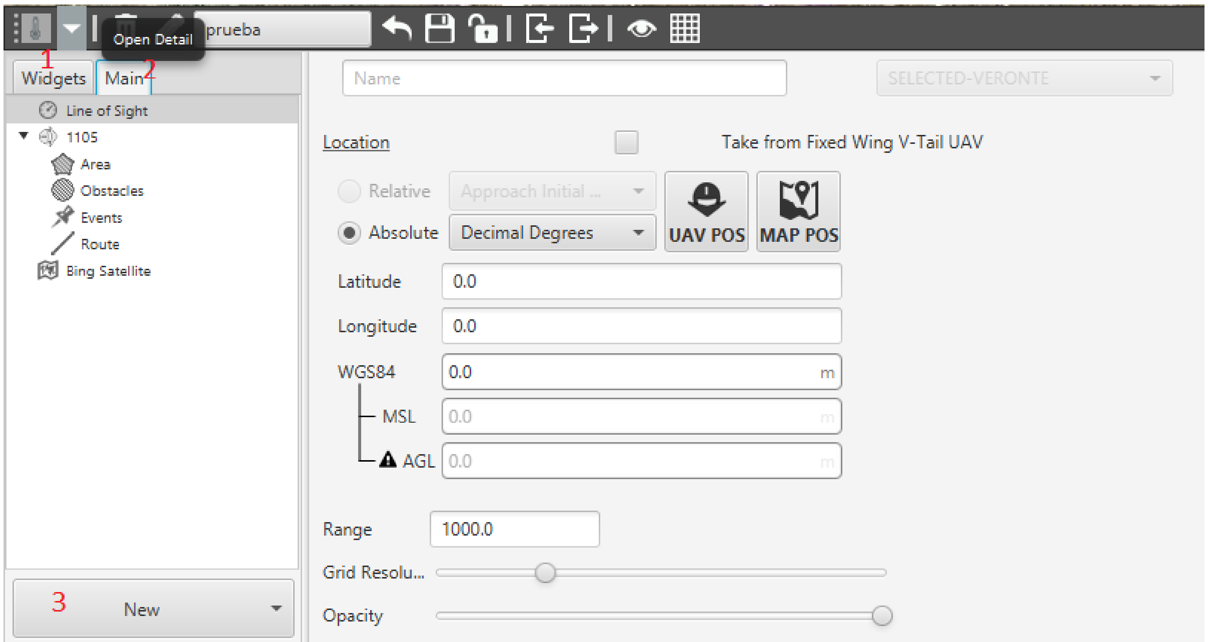
4.1 Workspace Toolbar

Workspace settings allow the user to customize any information to be displayed on the screen for monitoring the operation. Custom workspaces can be created, set any workspace as default in order to open it automatically on system start. All these options are available in the Workspace Toolbar, whichis shown on the image below.

Workspace Toolbar

	**Open Details**	Displays a panel where Widgets and Map are configured.
	**Remove**	Remove selected workspace configuration.
	**Rename**	Change the name of the Workspace.
	**Load**	Select the workspace to be displayed or create a new one.
	**Discard Changes**	Discard all changes.
	**Save Workspace**	Save workspace configuration.
	**Lock display position**	Lock/Unlock display position.
	**Open**	Open a workspace configuration.
	**Save as**	Save workspace configuration as.
	**Show/Hide**	Display or hide workspace elements.
	**Regroup**	Displays all widgets on the main screen.

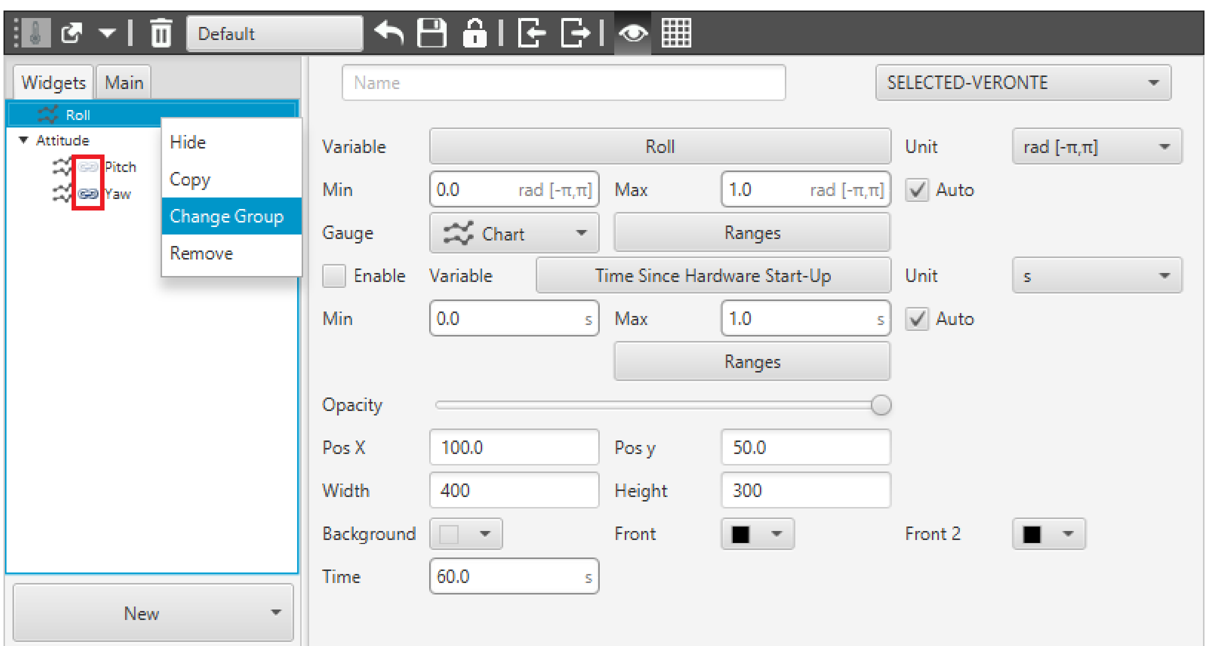
4.1.1 Open Details



'Open Details' Menu

1. **Widgets:** Displays a list with all the displays superposed to the map (widgets).
2. **Main:** Displays a list with all the displays bounded to the map.
3. **New:** Allows the addition of new displays which can be configured in **Widgets** and **Main**.

By right clicking on an item in **Widgets** or **Main**, the following dropdown list appears:

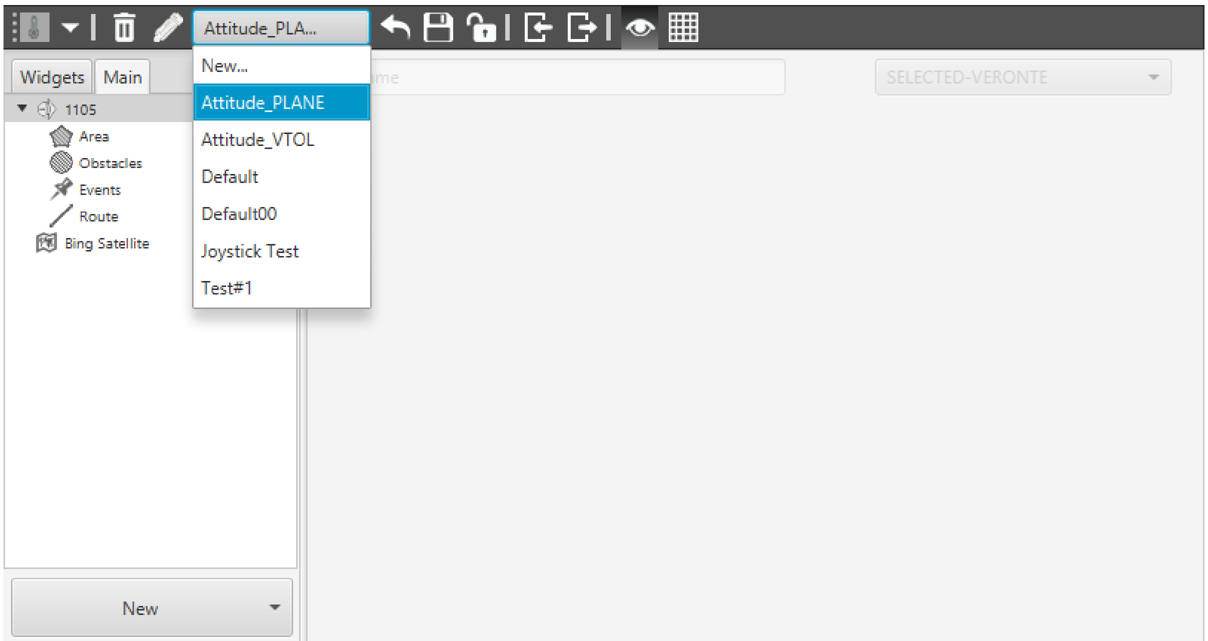


- **Hide:** Occults the selected element.
- **Copy:** Stores in the clipboard the selected element.

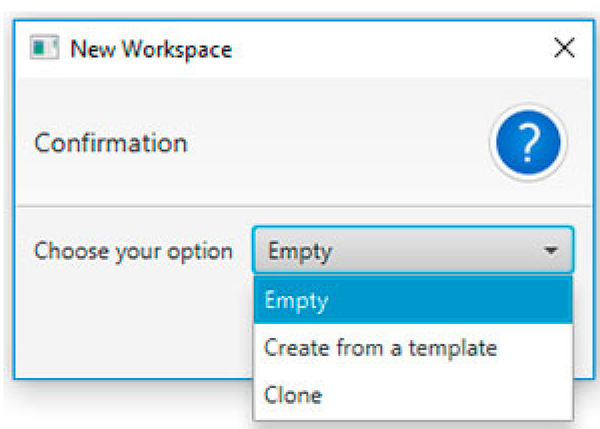
- **Change Group:** Opens a window to type the group to place the element in. If there is no group created or the name typed does not belong to any group, this option will create one and store the element in it.
- **Remove:** Deletes the selected element.

The option highlighted in red is the option to lock the chart. If all the elements of the group are marked they will move together, if one is not marked it could be moved with respect to the other group elements.

4.1.2 Load



By clicking on 'Load' icon a dropdown list appears with the current workspaces. Workspaces can be created by clicking on **New…**



New Workspace

The workspaces can be created from **Empty**, **Create from template** or **Clone**.

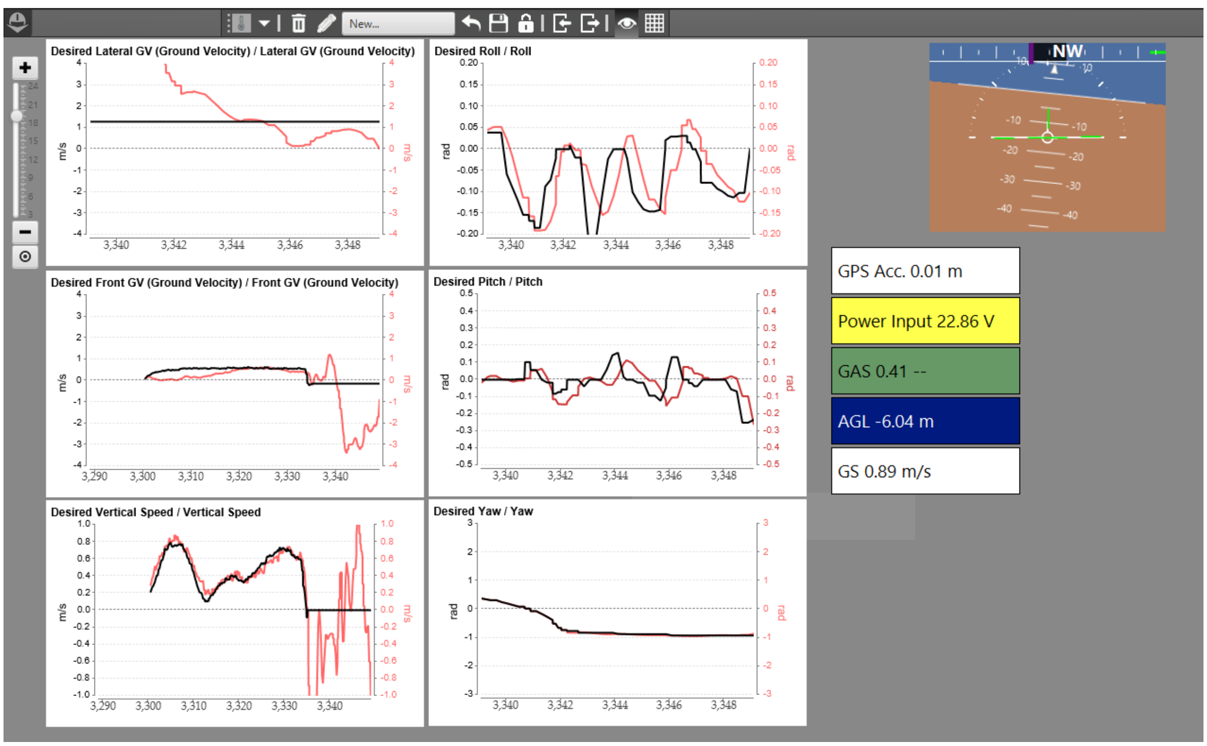
Workspaces can be modified too. The system allows the user to change the appearance and when it is completed, it can be saved by clicking on  present in the toolbar.

> **Warning:** Please note that Workspaces are not stored in Veronte Autopilot. Workspaces are located in the computer in which Veronte Pipe is installed.

In VerontePipe there are some templates available. When the workspace toolbar is open, the following templates can actually be opened:

- Attitude_Adv (Advanced)
- Attitude_PLANE
- Attitude_VTOL
- Default
- Joystick Test
- Map
- Output Test
- Sensors
- Stick Inputs

Following a picture of one of them:



4.1.3 Import/Export Workspaces

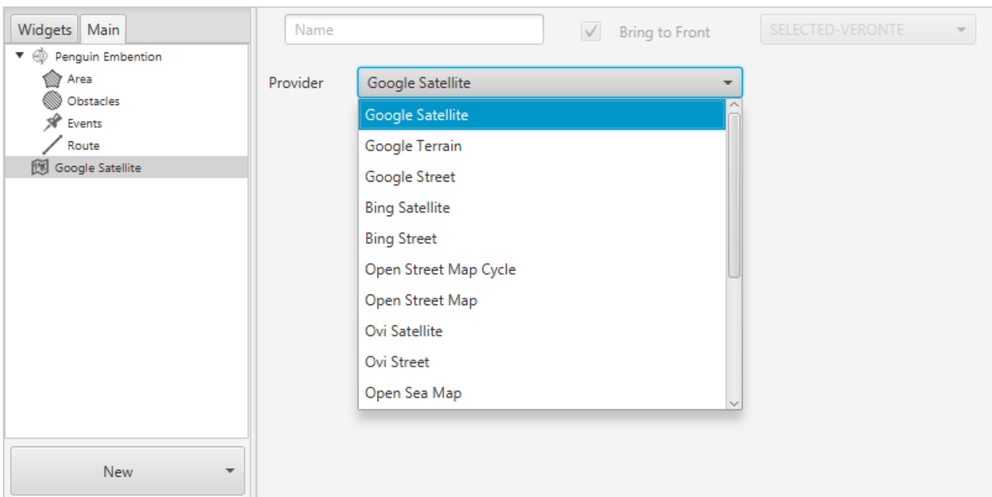
It is possible to save the current workspace on a **.wsn** file, that can be loaded later in a different configuration.

 and  are the buttons used to import and export a workspace respectively.

4.2 Map Display

4.2.1 Map provider

The map widget configures the background map that appears in Veronte screen. In the list shown in the following figure, it is possible to select the map provider from diverse options.



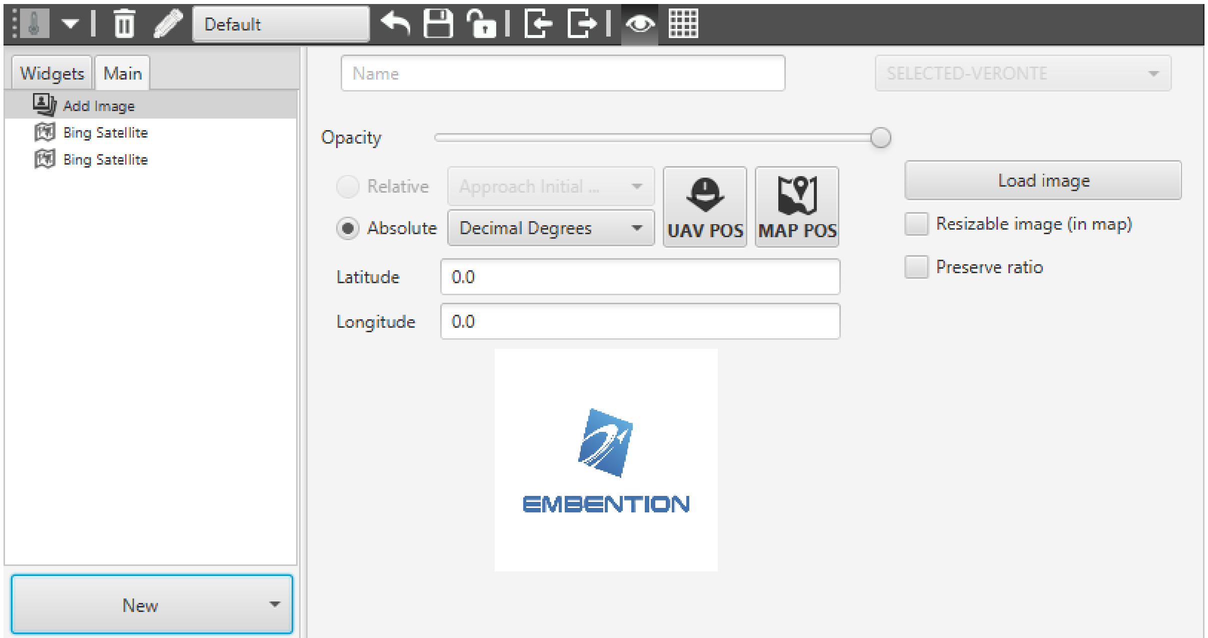
Custom Map Example

4.2.2 Custom image

Custom maps can be displayed in Veronte Pipe. It allows to include as many images as desired that will be displayed over the map.

Background Image Example

To insert an image within the map click on **New**, **Add images** and then select the desired image file. Once the image has been loaded, it is possible to configure its position and appearance in the image manager.



Background Image Positioning and Manager

Item	Description
Name	Define widget's custom name.
Opacity	Change widget opacity.
Relative	Define image's relative position coordinates.
Absolute	Define image's absolute position coordinates (UTM, MGRS, Decimal degrees…).
UAV POS	Changes image's position to the current selected UAV position.
MAP POS	Allows the user to click on the map and change image's position to the position clicked
Load image	Opens a window to select and place an image on the map.
Resizable image (in map)	When selected, allows change image dimensions.
Preserve ratio	When selected, image dimensions' ratio is conserved.

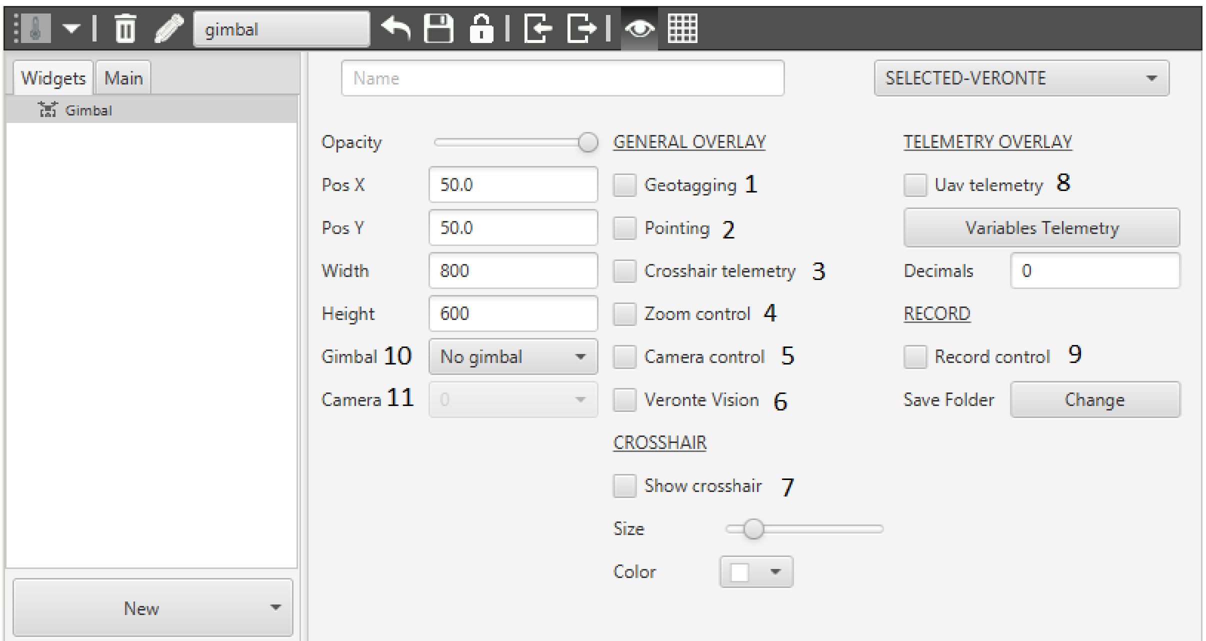
4.3 Widgets

4.3.1 Gimbal

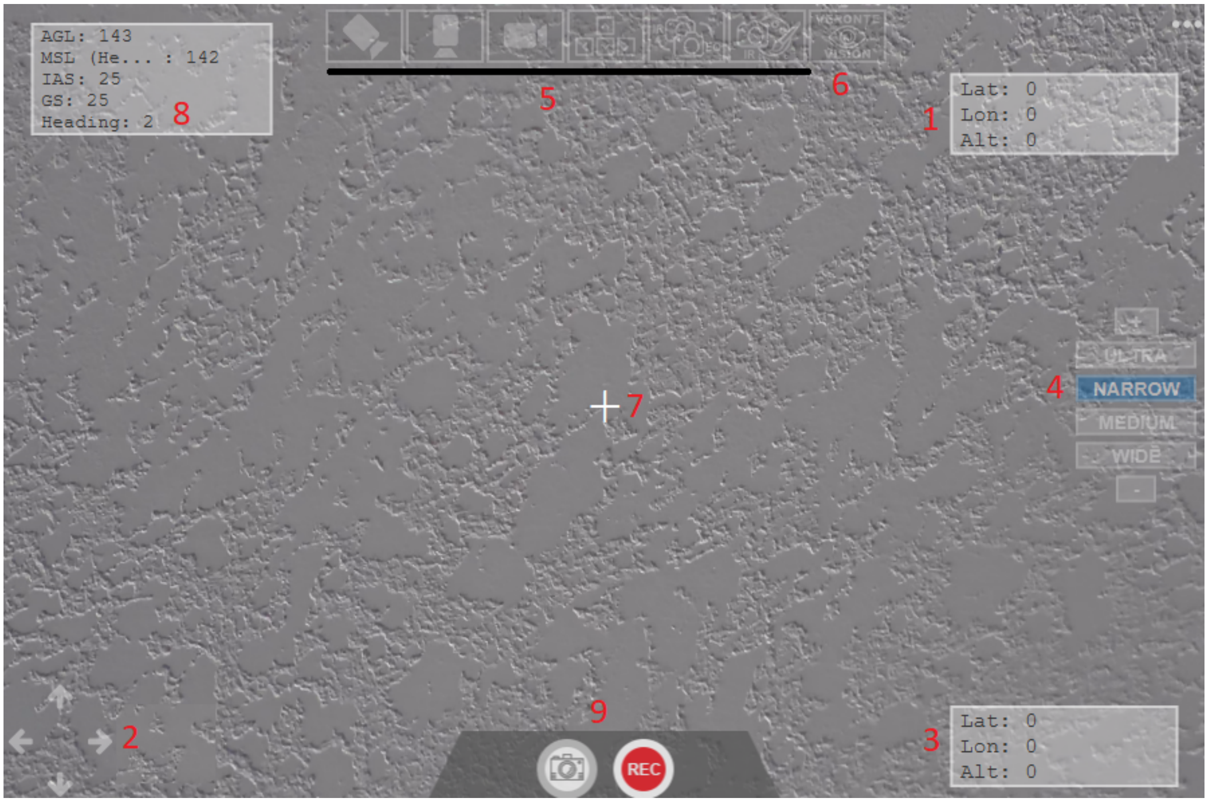
This widget allows the user to display a camera on screen and command a Gimbal if there is any device connected. The options configurable on this widget are the following ones:

1. **Geotagging.** Coordinates and height from mouse point selected. If Veronte Vision is in use, coordinates and height are from the tracking point.
2. **Pointing.** Manually move the camera through a digital stick.
3. **Crosshair.** Shows the coordinates from the picture central pixel (crosshair).
4. **Zoom control.** Top and bottom buttons for gradual zooming (+ and – icons) and 4 predefined zoom levels: wide, medium, narrow and ultra.

5. **Camera control.** Buttons placed on the top part of the widget for:
    - Three predefined positions (45º, Zenithal, Frontal – Navigation)
    - Allow keyboard camera control
    - Camera switch EO – IR
    - Colour palette switch for IR camera.
6. **Veronte Vision.** Enables Veronte Vision software for detection and tracking.
7. **Crosshair.** Show/hide, resize and colour modification.
8. **Telemetry Overlay.** Enables a screen with telemetry information into Gimbal Widget.
9. **Record control.** Photography and video recording, choosing in which folder to be stored.
10. **Gimbal.** Choose controlled Gimbal.
11. **Camera.** Choose controlled Camera.



Gimbal Widget Configuration Menu



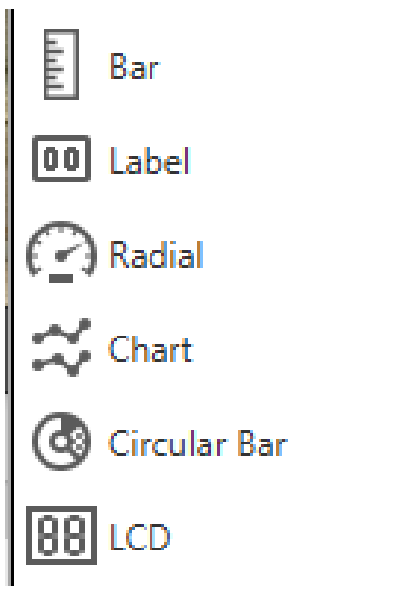
Gimbal Widget Display

4.3.2 Gauge Display

4.3.2.1 Gauge Selection

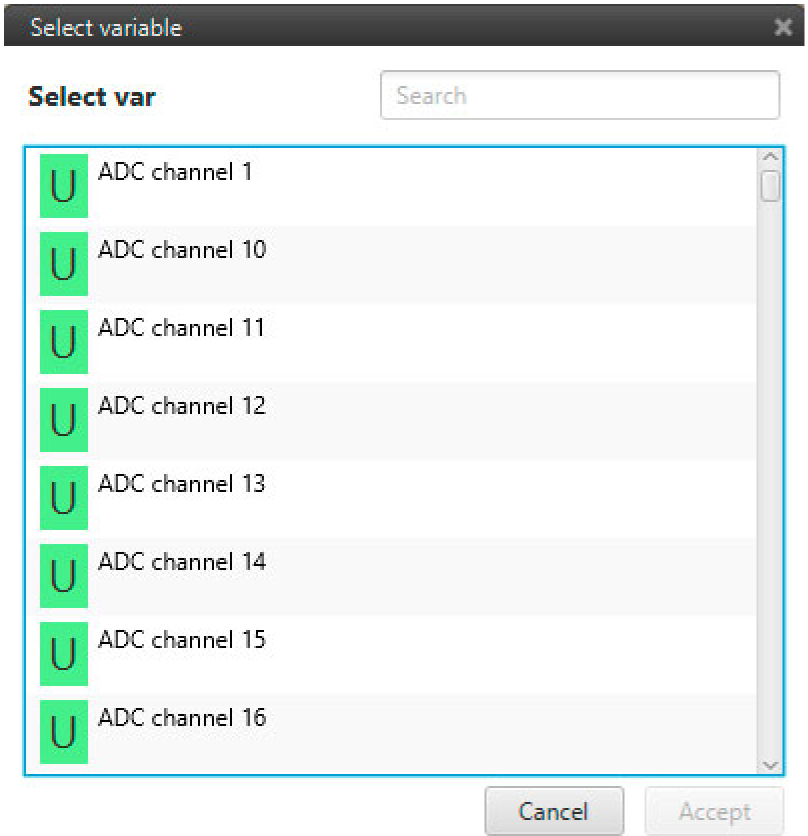
This option is used to display a variable on the workspace. In order to display a new variable:

1. Go to **Workspace** and click on **New**.



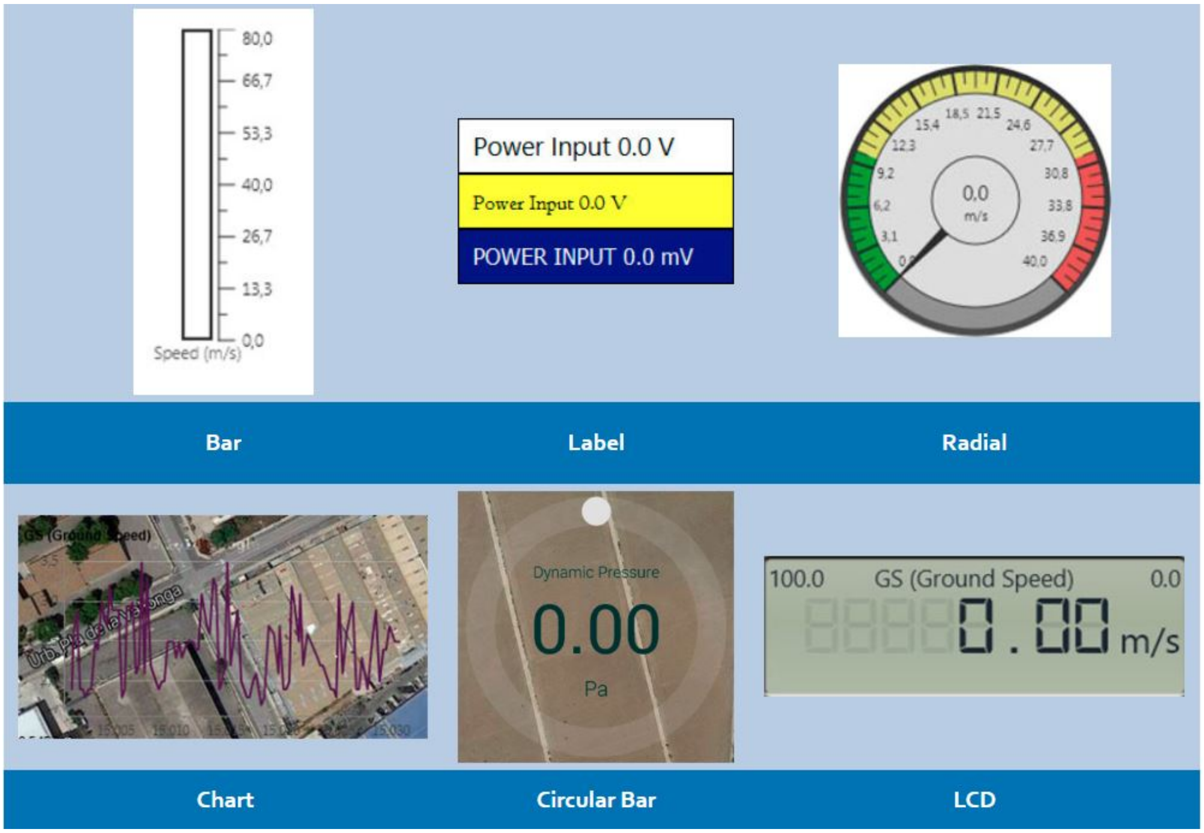
Gauge Selection Menu

2. When a gauge is selected, the system will ask for a variable to display. The variable selected can be changed easily by clicking on **Variable**, appearing then the following window.



Variable Selection

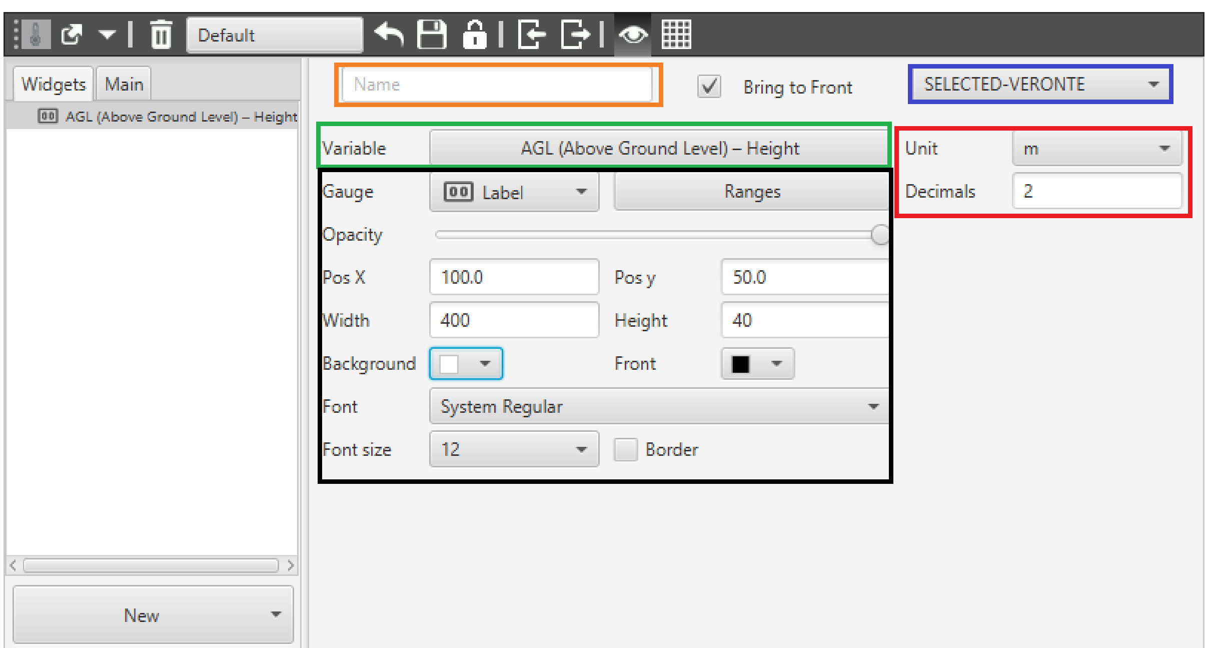
Once the variable to display has been selected, the way it is shown in the interface is also configurable. Here are some of the different gauge types that can be chosen.



Gauge Types

4.3.2.2 Gauge Configuration

Each gauge type can be configured in its panel and some parameters can be changed:



Gauge configuration

- Gauge name. It can be changed by typing a new one. (Orange)
- Selected Veronte. The dropdown menu allows to select another linked Veronte unit. (Blue)
  - If Selected Veronte is the option taken, the gauge will display the information corresponding to the autopilot which has been clicked on the Side Panel.
- Variable units and decimals. (Red)

Warning: Changing the unit of measure of the variable does not imply that the selected Veronte will gather the corresponding data in the selected unit. It only converts the unit.

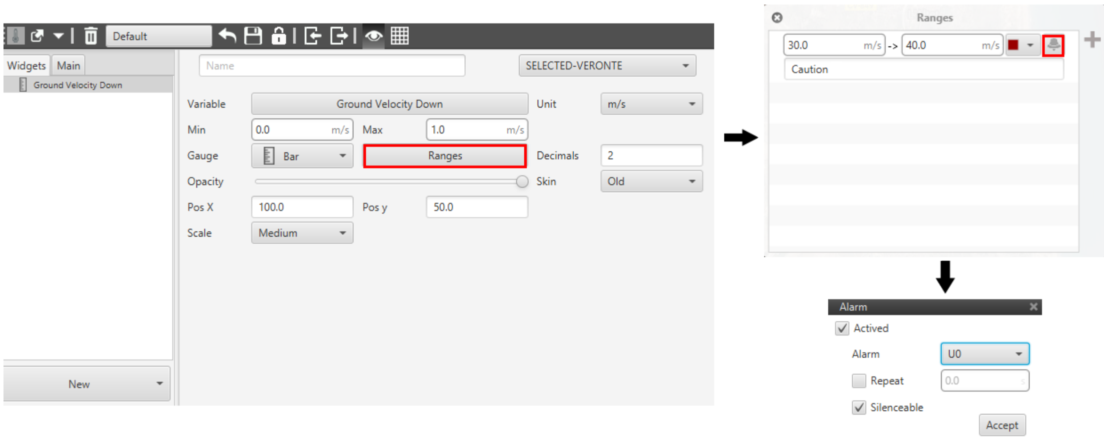
- Displayed variable. (Green)
- Gauge type, dimensions and position. (Black)
  - **Gauge:** Allows the user to change the gauge type (Bar, Label, Radial…)
  - **Ranges:** Allows the user to define an interval by clicking on the '+' button. For this interval several customizations can be made:
    - Colour: Change the color of the interval which will be shown graphically in the gauge (does not apply to graph gauges)
    - Alarm: By clicking the bell icon an alarm can be created. For more information about alarm customization please refer to 'Alarm Creation' section.
    - Text: Add a text to be displayed (only in the gauge types that allow this addition of text)

One of the options that appear for all the kind of gauges is the customization of the colour according to the value of the variable shown.

Warning: Variables will not be collected from the autopilot unless  button is pressed or selected from the *Telemetry* panel.

4.3.2.3 Alarm Creation

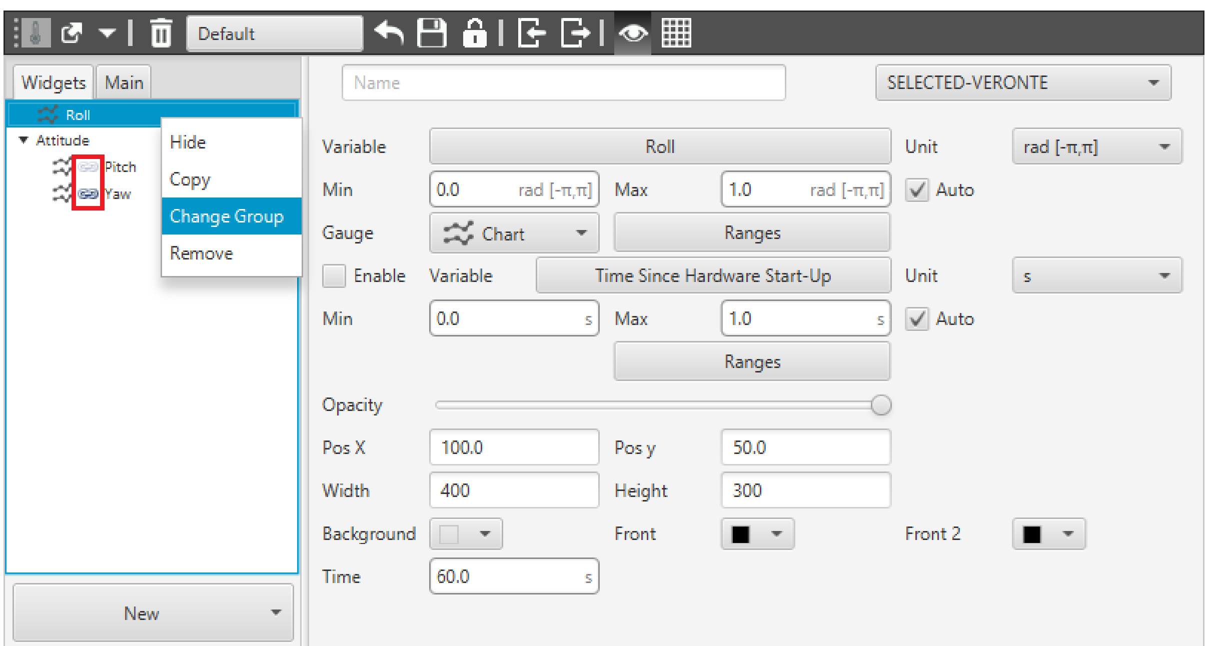
1. Select the option **Range** and a new window will be displayed.
2. In the new window, the user can create different ranges, each one having a certain colour and an associated alarm of the ones configured on **Pipe Configuration-Preferences-General-Alert_Audioclips**.
3. Click on the Alarm icon and configure it.



Gauge – Range

> **Warning:** In the case of displaying a boolean on the workspace (GNSS Navigation Down for example), in the Range menu will appear automatically the two states, each one with a colour (green, red), and also with the possibility of selecting an audio clip for each one of them.

When displaying a chart, it is useful to overlap with transparencies two or more in the same figure to observe their evolution. To do that, Pipe provides the *Change Group* option. Right clicking on the graph and selecting this item displays a window where the group name can be introduced. If two charts are in the same group, their relative position will be kept unchanged (they will move together).

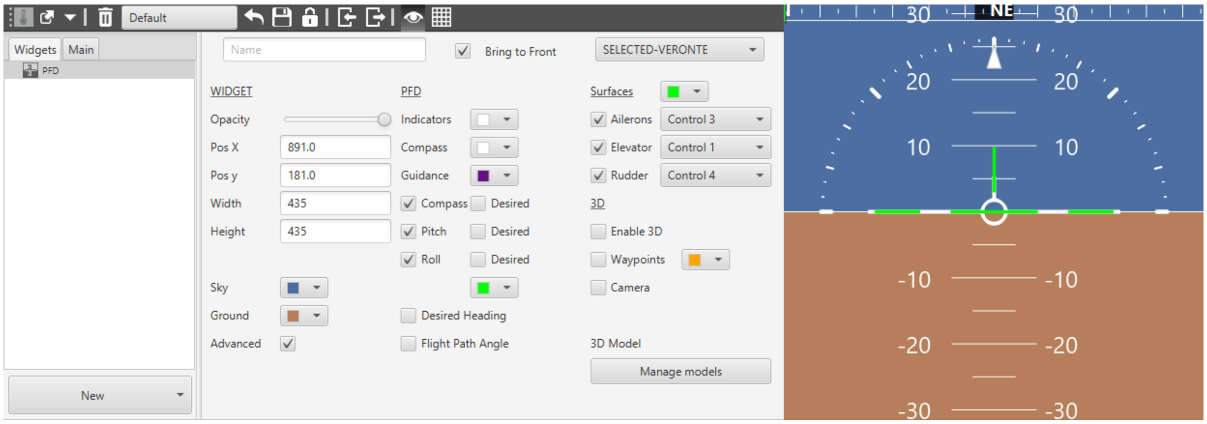


Variable – Group

The option highlighted in red is the option to lock the chart. If all the elements of the group are marked they will move together, if one is not marked it could be moved with respect to the other group elements.

4.3.3 Advanced Primary Flight Display (PFD)

The **primary flight display (PFD)**, also known as "artificial horizon", represents graphically the attitude of the aircraft (roll, pitch and yaw). This display is highly configurable in colours and size. It is possible to select between the 2D and 3D visualization, and also the surface deflection can be represented.



PFD Configuration

Item	Description
Name	Define widget's custom name.
Bring to front	Places the current widget above every other display.
SELECTED-VERONTE	Dropdown menu to select which Veronte will the widget gather data from.

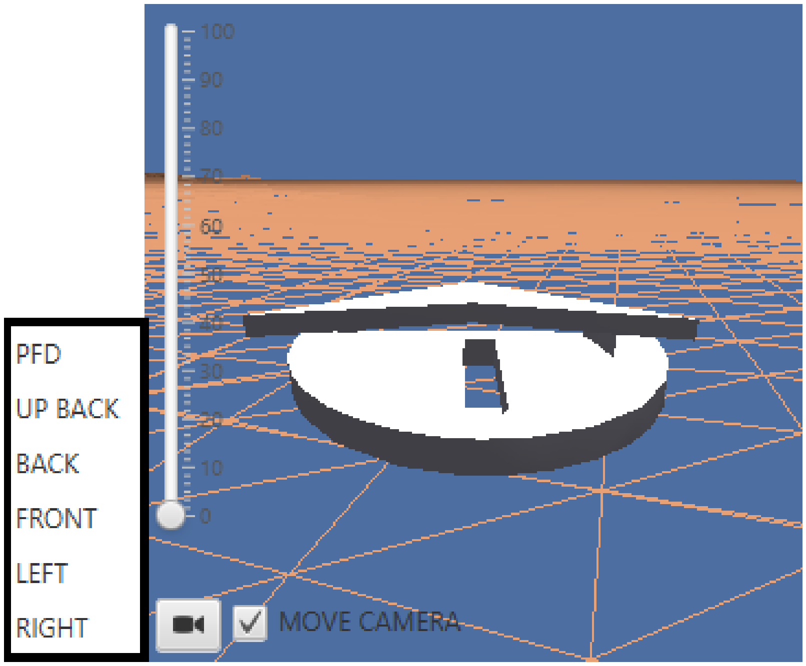
Widget	
Opacity	Change widget opacity.
Pos X/Y	Define widget's position in the screen's X/Y axis.
Width/Height	Define widget's width/height.
Sky/Ground	Select sky's/ground's color.
Advanced	When marked, displays all the configuration options available.

PFD	
Indicators/Compass/Guidance	Change display color of the selected feature.
Compass/Pitch/Roll	When marked, displays the selected feature.
Desired	When marked, it displays the desired Compass/Pitch/Roll, a color change option is available.
Desired Heading/Flight Path angle	When marked, displays the selected feature.

Surfaces	
Ailerons/Elevator/Rudder	When marked it displays the selected control surface, the control channel can also be selected.

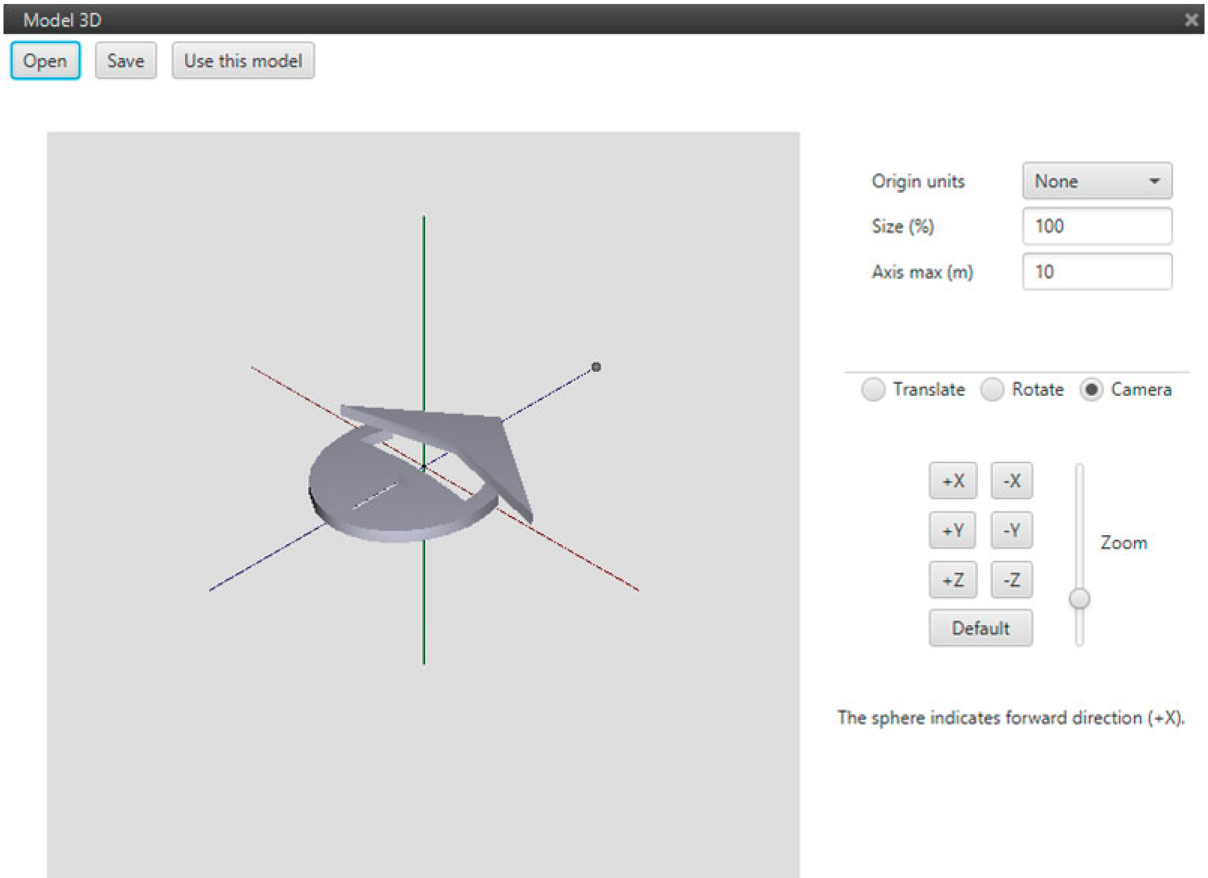
3D	
Enable 3D	When marked, it enables the 3D feature
Waypoints	Displays mission waypoints in the PFD.
Camera	Displays camera view if 'Enable 3D' feature is marked.

- By clicking the camera icon, the camera view can be selected: *PFD* (internal view at the cabin), *UP BACK*, *BACK*, *FRONT*, *LEFT*, *RIGHT*.
- When an external view is selected, it will be displayed a slide to change the distance between the camera and the aircraft, and a *MOVE CAMERA* checkbox, which allows to move around the view by dragging the mouse on the widget.



3D PDF visualization

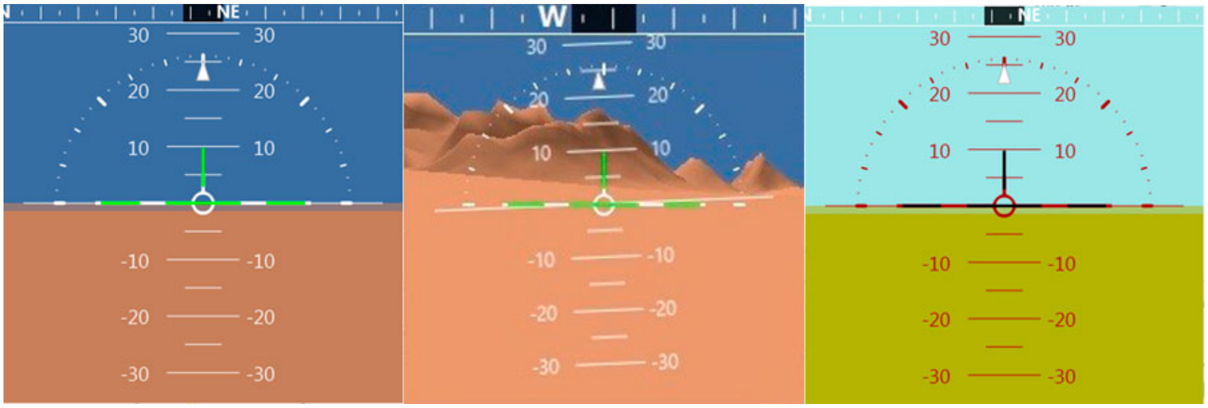
3D Model	
Manage models	Changes the 3D model display.



3D Model

- **Open:** it allows to select an STL file to be loaded.
- **Save:** save user changes in an STL file.
- **Use this model:** select the current model to be displayed on the widget.
- **Origin units:** user can select origin file units. The model is real-scaled according to the terrain shown on the widget.
- **Size:** percentage scale factor.
- **Axis max:** axis length in meters.
- **Translate/rotate/camera:** translate and rotate modify model orientation. Camera changes the view displayed in Model 3D window. X,Y,Z buttons produce changes according to this axis. Default button resets any modification.

Some PFD display configurations are shown as an example:



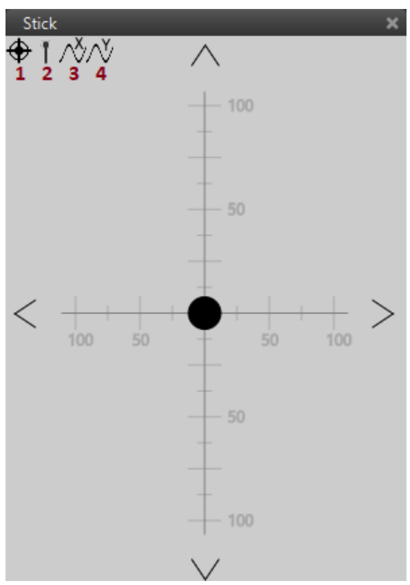
PFD Example

4.3.4 Stick

Virtual sticks are created to simulate a radio controller that controls the channels of the aircraft directly from the computer.

In the stick display, there are four icons and their function are:

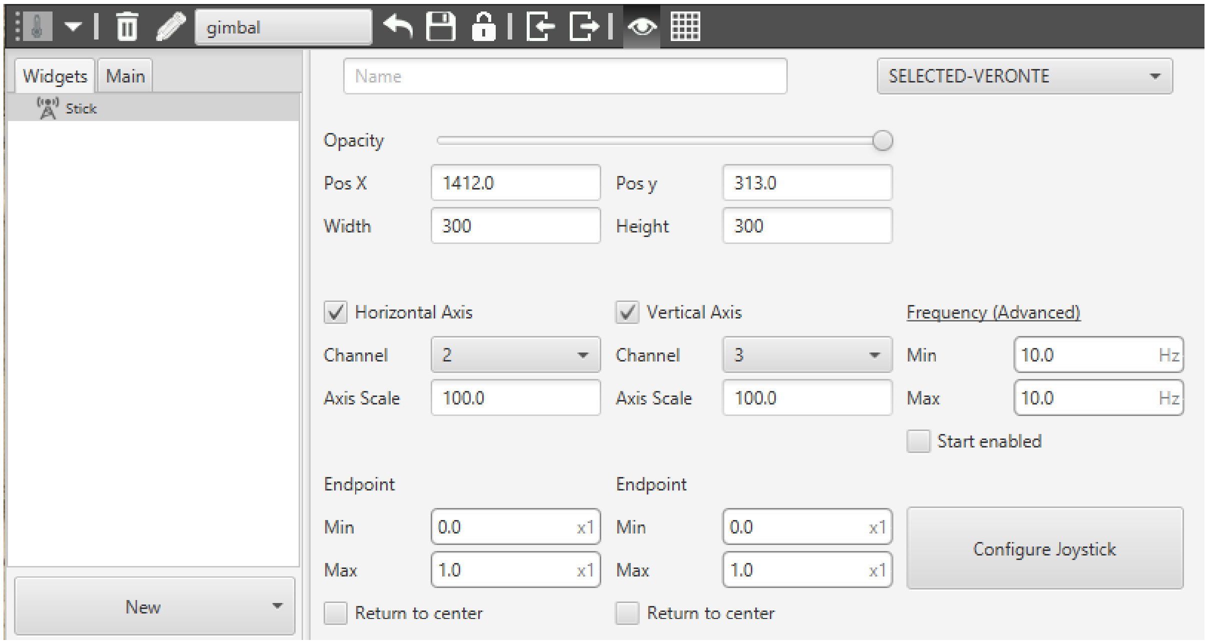
1. Return the stick to the centre
2. Activate/Deactivate the virtual stick
3. Move the stick in the X direction in a sinusoidal way
4. Move the stick in the Y direction in a sinusoidal way



Stick Display Configuration

Last two are normally used in order to test the stick or servos.

Warning: If the Stick is not active it will make no effect on the system. Activate it by clicking on the second icon (some waves will appear near the icon and the stick will be actived). In order to deactivate it, only click on the antenna icon one more time.



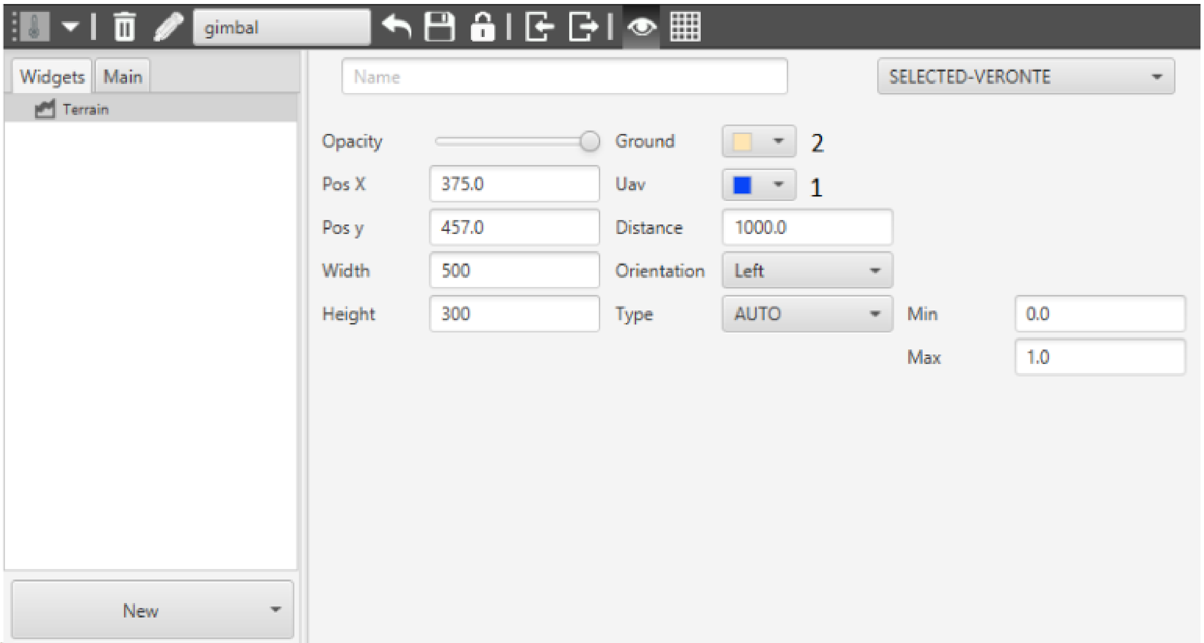
Stick Configuration Menu

In the dropdown on the top right corner, the user can select the Veronte to be used in this widget.

Item	Description
Name	Define widget's custom name.
Opacity	Change widget opacity.
Pos X/Y	Define widget's position in the screen's X/Y axis.
Width/Height	Define widget's widt/height
Channel	Select which channel is controlled by each axis.
Axis Scale	Scale to show in the axis of the stick.
Endpoint Min & Max	Establish the minimum and maximum values reached by the stick.
Frequency Min	Sets the minimum quantity of messages sent when there is no movement.
Frequency Max	Sets the maximum quantity of messages that can be sent.
Return center	When it is selected the stick automatically returns to middle position on stick release.
Configure Joystick	Configure external USB joystick for camera control.

4.3.5 Terrain

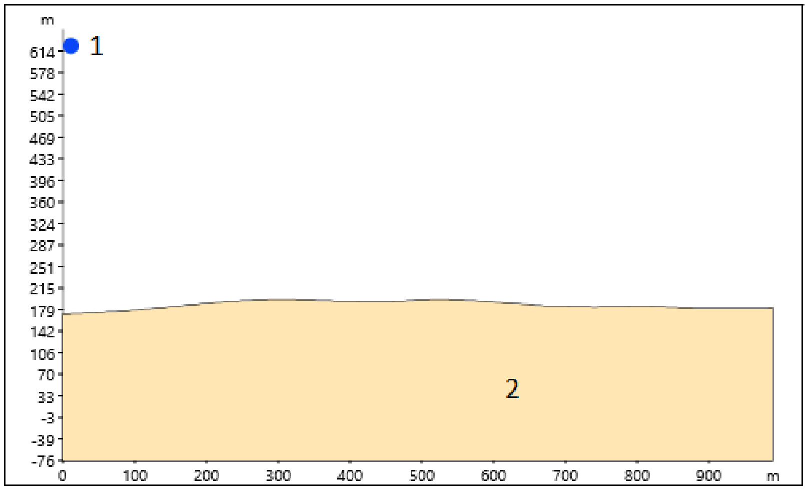
Terrain display shows the terrain profile on the platform direction. Visualization configuration options are as follows:



Terrain Profile Configuration

In the dropdown on the top right corner, the user can select the Veronte to be used in this widget.

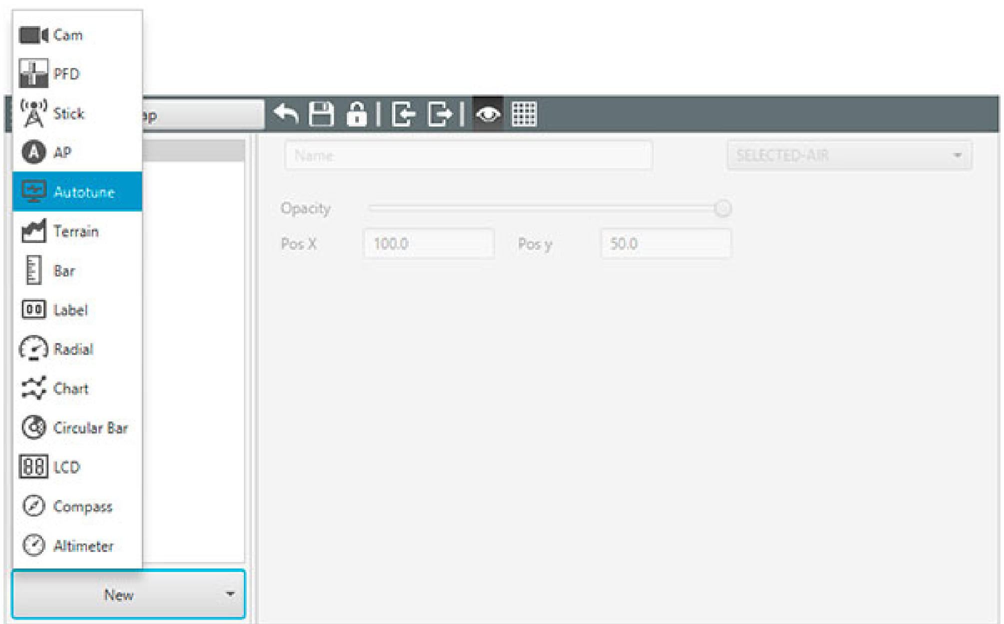
Item	Description
Name	Define widget's custom name.
Opacity	Change widget opacity.
Pos X/Y	Define widget's position in the screen's X/Y axis.
Width/Height	Define widget's width/height.
Ground	Select ground's color.
Uav	Select UAV's color.
Distance	Change distance range.
Orientation	Define UAV ahead orientation (from left to right or the opposite)
Type Min/Max	Select where the program catch the variables.



Terrain Profile Widget Display

4.3.6 Autotune Tool

In the Workspace panel, it is possible to select the Autotune Tool. It is necessary to configure this window in order to be able to find automatically the gains of the PID controllers.

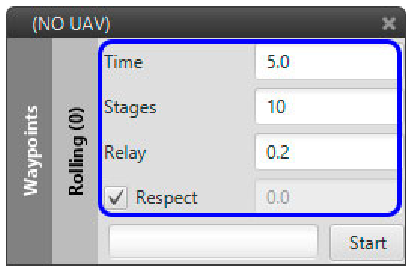


Autotune tool in the Workspace

When the window is opened, it is possible to select the loop for the autotuning.

Autotune loop selecting

This window shows the actual flight phase in the left bar and the control names. For each control, the available loops are showed and they are enumerated from 0 to 2 (from the intern one to the extern one). If some control loop does not have a PID controller configured, it appears with transparency and it can not be autotuned (red).



Autotune parameters

When the loop is selected, the window changes and some parameters (blue) are showed and they can be edited:

- **Time:** It is the period of time in which the Autotune is performed [s].
- **Stages:** The number of stages of the Fast Fourier Transform (a value between 5 and 10 is allowed).
- **Relay:** This is the amplitude of the Relay function (R). The value has to be chosen in accordance with the proportional gain in the PID of the autotuned variable.
- **Respect:** A mean value of the variable has to be selected. If the respect is select, the autotune will start from the last value of the variable. If not, the value can be edited by the user and the Autotune Relay function will start from this value and it will go from -R to R.

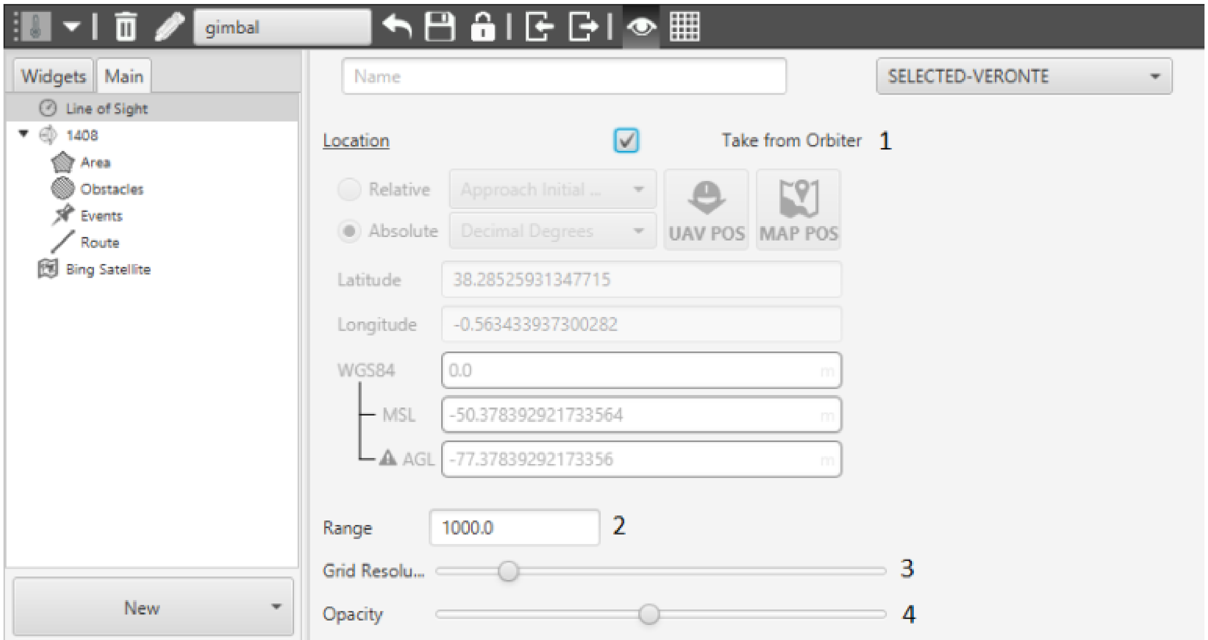
When all parameters are set, it is possible to click on Start and the autotuning process will begin and the blue bar will move until the end of the process. The left bar of the window allows checking the flight Phase and the selected loop.

In the section *Examples - Autotune* of this manual, an autotune example is explained in detail.

4.3.7 Line of Sight

Line of Sight is a widget with which the user can display a grey area around Veronte, showing the Line of Sight it has on every moment at 360º. The configurable parameters are:

1. **Location.** Choose from showing the LOS area from the UAV (checkbox) or any particular location which is fixed (menu).
2. **Range.** The radius of the area covered in meters.
3. **Grid resolution.** A slider that increases/decreases the number of points calculated in the grid. More points equal more resolution but more calculations as well.
4. **Opacity.** A slider that allows the change in LOS area opacity.

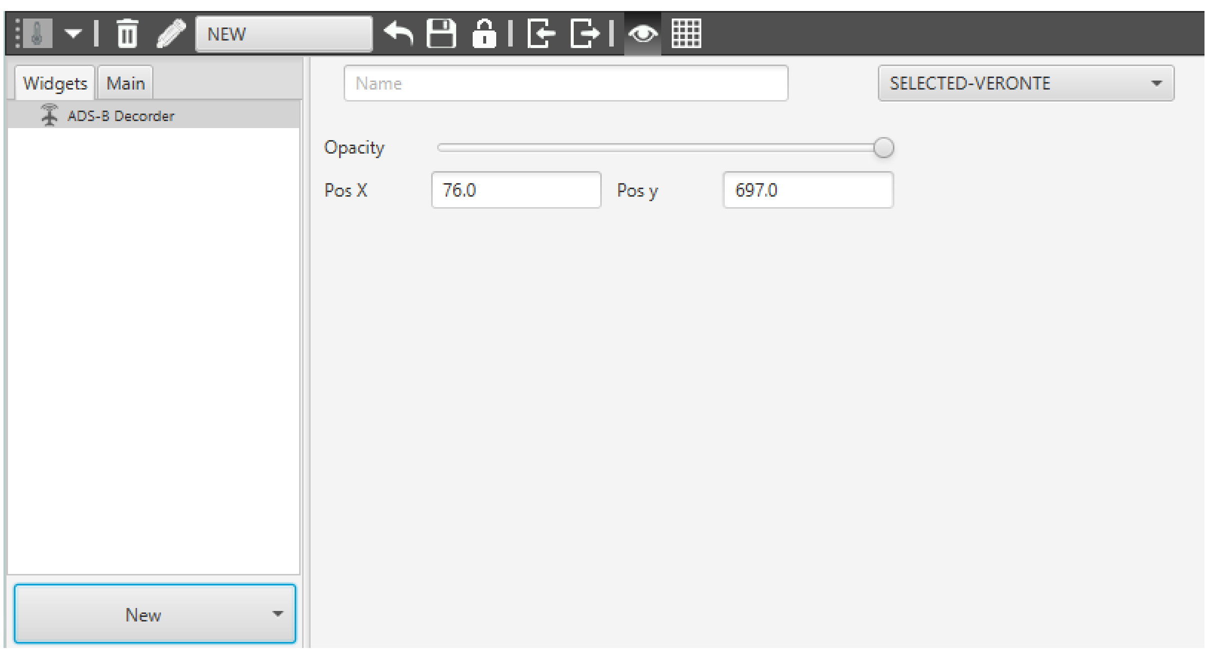


Line of Sight Configuration Menu

Line of Sight Simulation

4.3.8 ADS-B decoder

The ADS-B Decoder widget is a visual information menu page where the user can get all ADS-B information displayed in one window while also having real-time traffics on the screen.



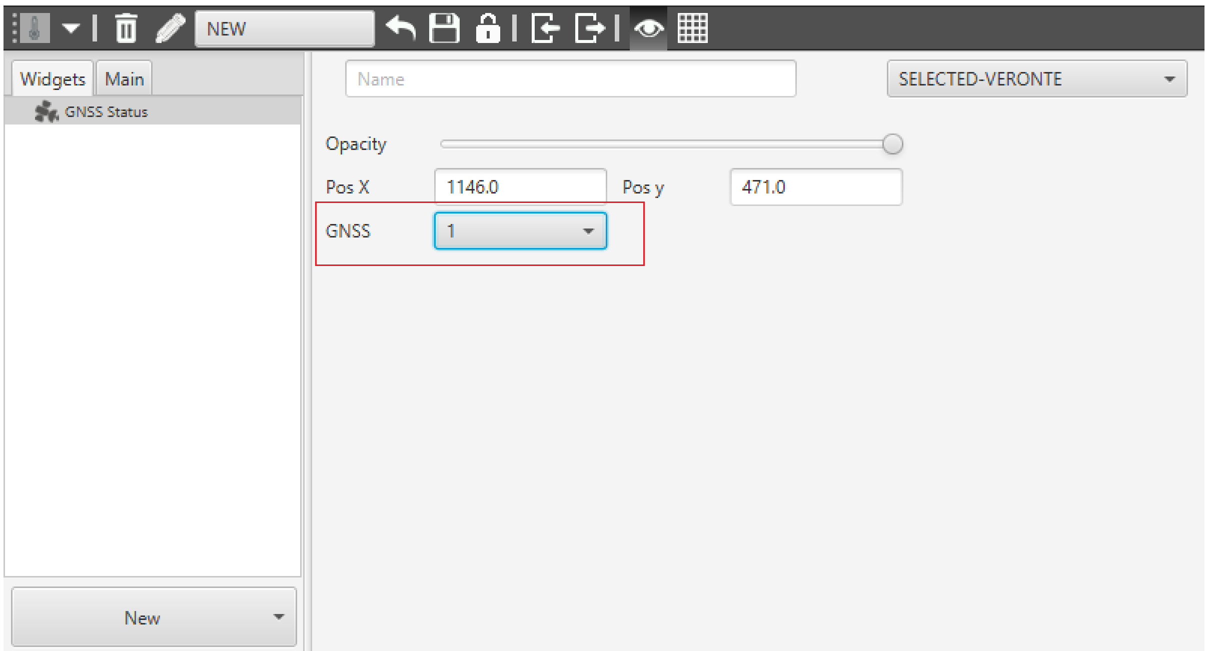
ADS-B Decoder Menu

Item	Description
Name	Define widget's custom name.
SELECTED-VERONTE	Dropdown menu to select which Veronte will the widget gather data from.
Opacity	Change widget opacity.
Pos X/Y	Define widget's position in the screen's X/Y axis.

ADS-B Real Session

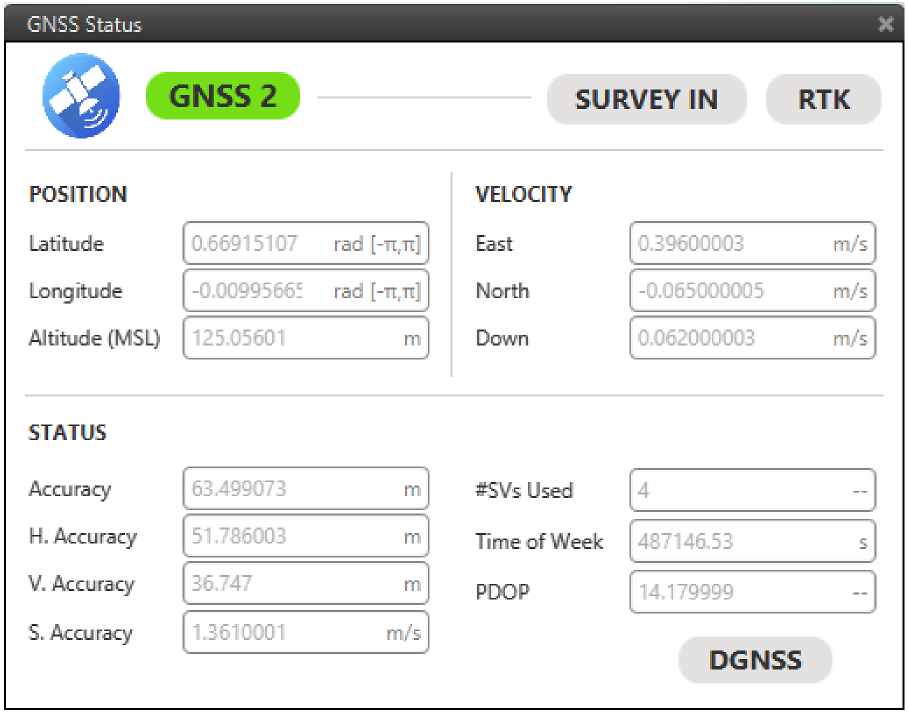
4.3.9 GNSS status

The GNSS Status widget is a visual information menu where the user can get all information from the GNSS configured in one same window. It can be chosen which GNSS is displayed:



GNSS Status Configuration Menu

Item	Description
Name	Define widget's custom name.
SELECTED-VERONTE	Dropdown menu to select which Veronte will the widget gather data from.
Opacity	Change widget opacity.
Pos X/Y	Define widget's position in the screen's X/Y axis.
GNSS	Select GNSS to be displayed.



GNSS Status Display

The information shown in the widget is as follows:

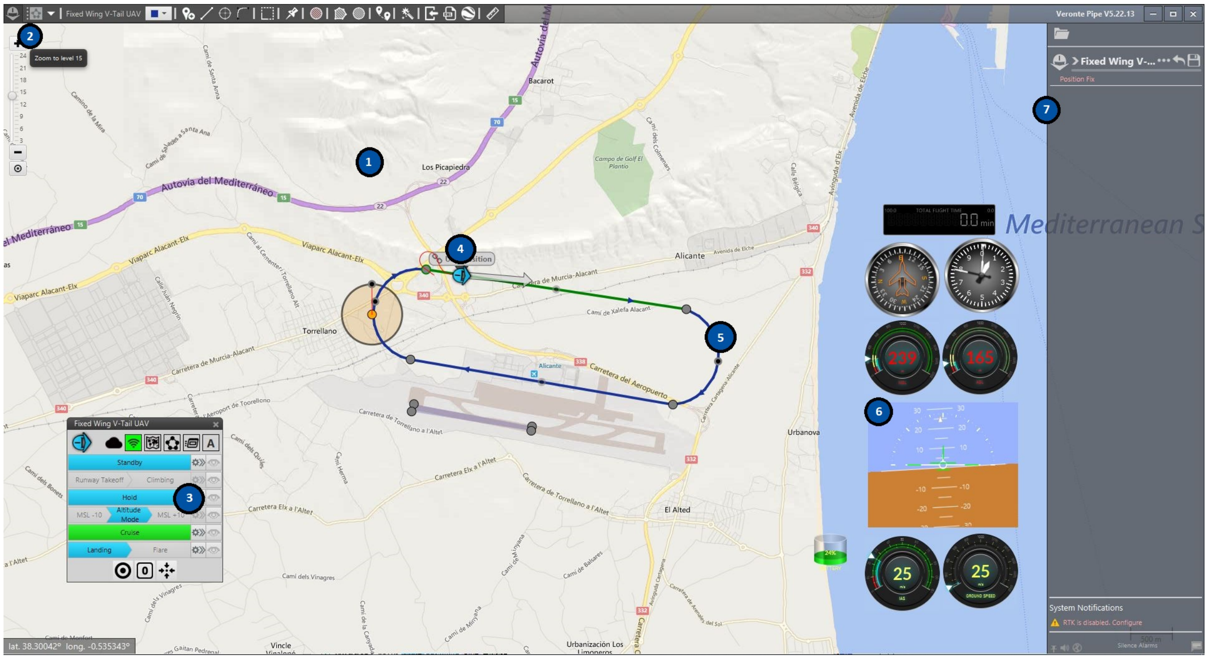
- **GNSS 1/2.** Navigation bit status (red/green).
- **Survey In.** RTK conditions being accomplished if green. Grey if finished or not achieved.
- **RTK.** Becomes green when SurveyIn bit goes grey (off) and RTK is enabled. Grey otherwise.
- **DGNSS.** Green when Differential Positioning is enabled (GNSS Compass). Grey if not.

Veronte Pipe offers several applications to display parameters and variables in real time of the flight mission.

These are called widgets, they can be created by clicking on 'New' under the  icon in the workspace toolbar.

4.4 Main Interface

Workspace on Veronte Pipe is distributed as shown in the following figure.



Workspace Main Interface

Each section has the following functions:

1. **Main Display:** Displays a selectable background map or a plain colour together with the most important mission data. To change it visit *Map Display*
2. **Main Menu:**
    - **User:** Manage user preferences.
    - **Cloud:** Provides access to the Cloud options.
    - **Workspace:** Select the way flight information is displayed.
    - **Mission:** Create and edit missions.
    - **Log:** View operation data log and introduce custom events.
    - **Post Flight:** Tools for recorded data analysis.
    - **Preferences:** Configure Pipe and Veronte autopilot.
    - **License:** Manage license preferences.
    - **Manual:** Shows help information available.
3. **Veronte Panel:** Veronte information and telecommand buttons.
4. **Veronte Position:** Veronte location on the map.
5. **Mission:** Defined mission on Veronte.
6. **Telemetry:** Configurable drag & drop flight information displays.
7. **Side Panel:** Shows linked Veronte information.

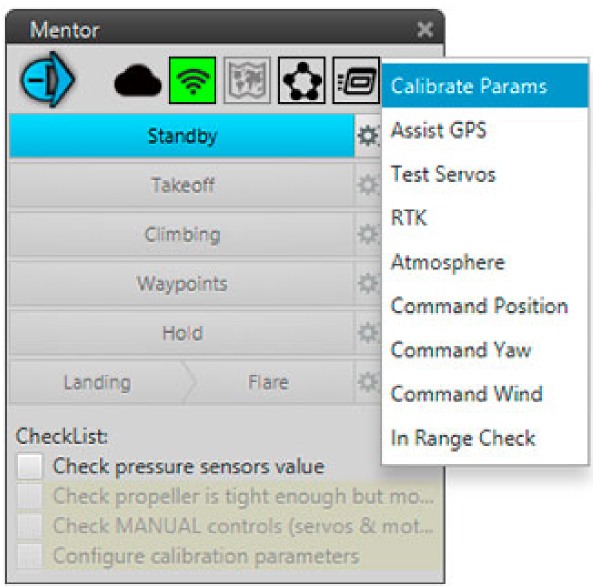
CHAPTER

FIVE

VERONTE PANEL

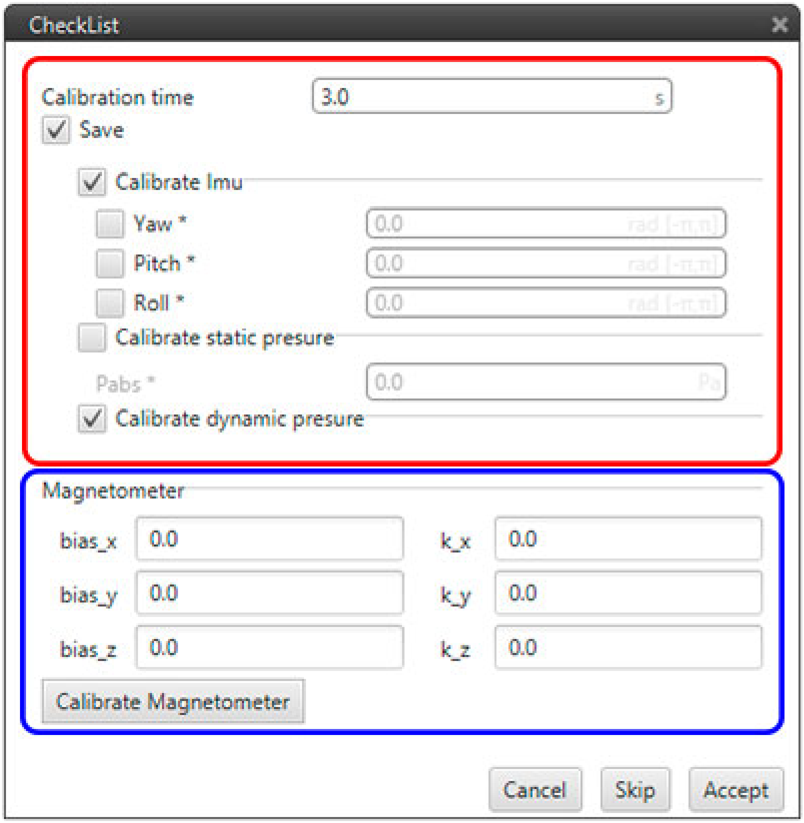
5.1 Sensors Calibration

The internal sensors fitted in Veronte Autopilot can be calibrated through the menu that can be displayed from the Veronte Panel.



Veronte Panel – Parameters Calibration

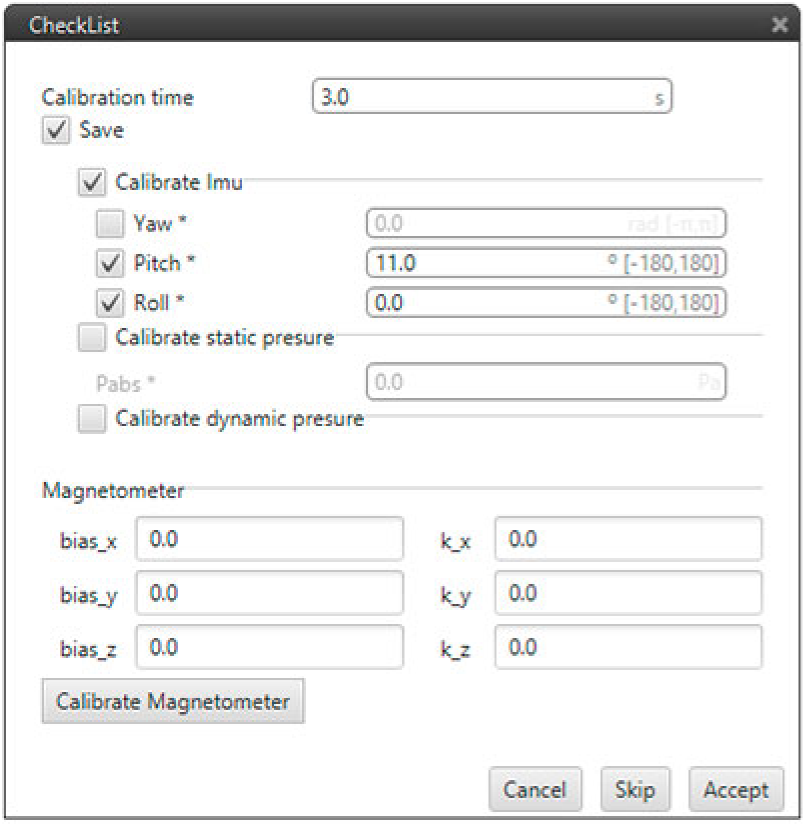
The Calibrate Params option will open the calibration menu shown in the following figure.



Sensors Calibration Menu

This menu allows calibrating the IMU, the Dynamic pressure sensor and the Static pressure sensor. When sensors are marked, the Save option is selected and the button "Accept" is clicked, the calibration is done, so the user should proceed with caution and mark only the sensors that he wants to calibrate in the red section.

- The IMU is calibrated according to the aircraft current position, so the UAV has to be located in a leveled position when performing the calibration. The three axes of the platform can be calibrated separately and at the desired angle. For example, the following figure represents the typical IMU calibration of Mentor which has 11° [deg] of pitch angle when positioned horizontally on the ground.

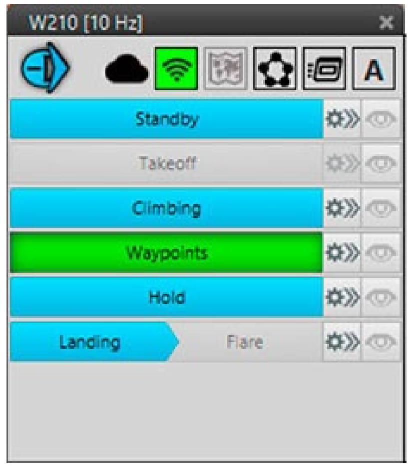


IMU Calibration of Mentor

- The **Static pressure** sensor is used to obtain the altitude. If the pressure at the flight area is known, introducing this value and calibrating the sensor will improve the height estimation.
- The **Dynamic pressure** sensor (Pitot tube), used to estimate the velocity with respect to the air, has to be calibrated when the orifice of the tube is covered, so the calibration is done from a 0 m/s lecture.

The blue section of the panel allows the user to access to the Magnetometer calibration panel. For more info about the magnetometer, see section *Magnetometer*.

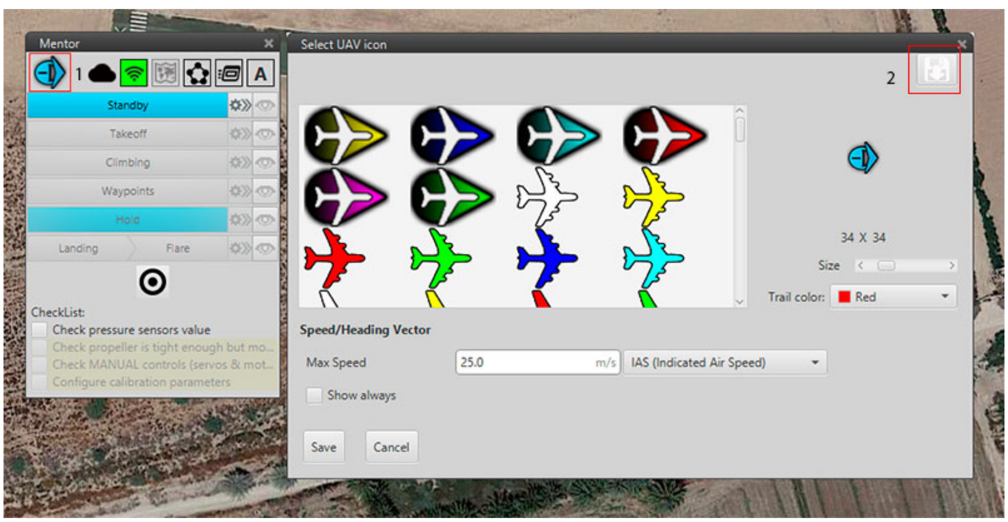
Veronte Panel is the display that shows the different phases of the flight operation and permits the user to change from one to another. Besides, it can be configured to allow the user to change different parameters of a phase during a real flight (gains, speed, altitude and so on). The panel also shows the checklist that contains a set of actions that must be done before entering into a new phase. In order to configure



Veronte Panel

5.2 Veronte Icon

The system also makes it possible to change the icon that indicates the position of the autopilot over the map.



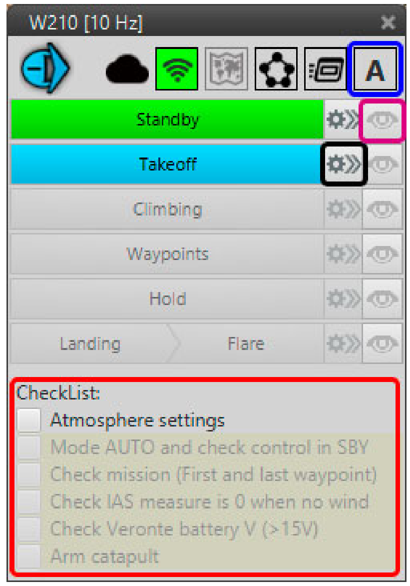
Veronte icon selection

In the Veronte Panel, pressing 1 will display a window where the user can change the icon that marks the current position of the autopilot (once it has GNSS coverage). There exist in the system a list of icons, but it is also possible to introduce new designs stored in the computer using option 2.

The trail that indicates the path that the autopilot has covered is also customizable by the user (color).

5.3 Veronte Panel

Double click on any Veronte to display its Veronte Panel. This item is the main interface to control the flight operations.



Veronte Panel checklist

The current phase is marked in green, select one of the available blue phases to change manually. When entering a new phase, all required **Checklist elements** (red) must be completed.

There are two options when entering a new phase:

- Clicking on the phase name (blue box) will make the aircraft enter the phase with the preconfigured parameters.
- If the **Options button** (black) is selected, the system will enter the phase but a window will appear allowing the user to change the parameters of that phase. Phase parameters can also be configured on the control tab on the setup menu.

The **Mode button** (blue) indicates the actual flight mode (A-Auto, M-Manual…) and allows to change it. Finally, the View button (purple) displays the phase route on the screen.

Dependencies between phases and automatic phase transitions are configured on the automations panel.

CHAPTER

SIX

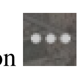
VERONTE CONFIGURATION

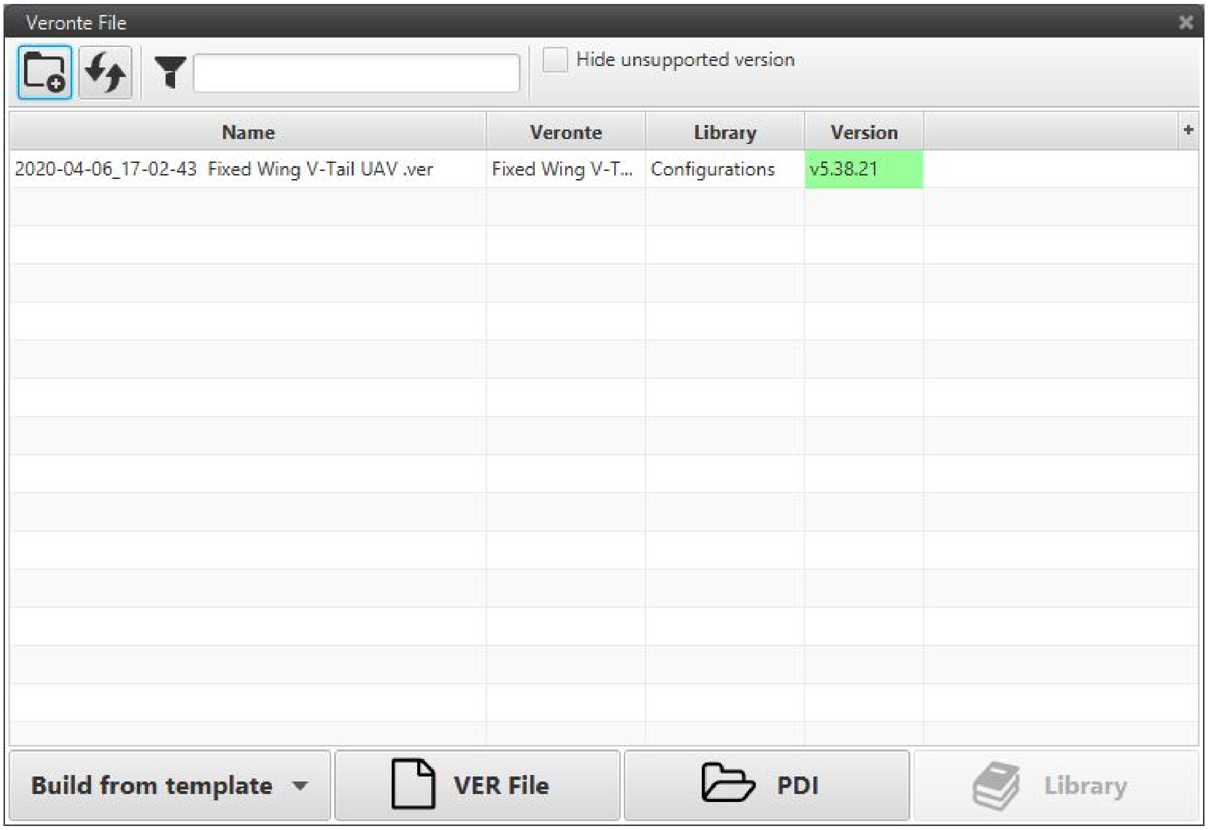
6.1 File Management

6.1.1 Import Configurations

In this section is explained the process to load a configuration file into a Veronte Unit.

Import Configuration

Power and connect the autopilot to Veronte Pipe, in the side panel, click on  and then select Import. The **Import Configuration** option displays a window used to select the file to be uploaded.

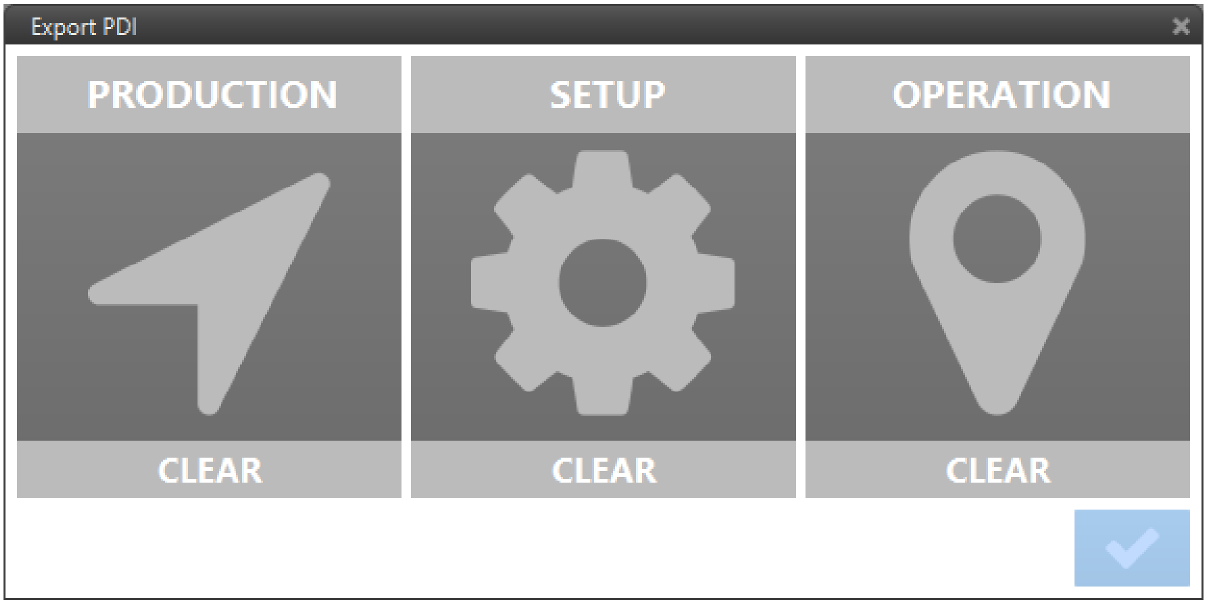


- **Build from template:** By selencting this option the user can import a configuration file from a set of default configurations that can be used as a starting point. There are templates for multicopters, fix wings, Ground station among others.
- **VER File:** Veronte Pipe will request the location of the .ver file to be loaded.

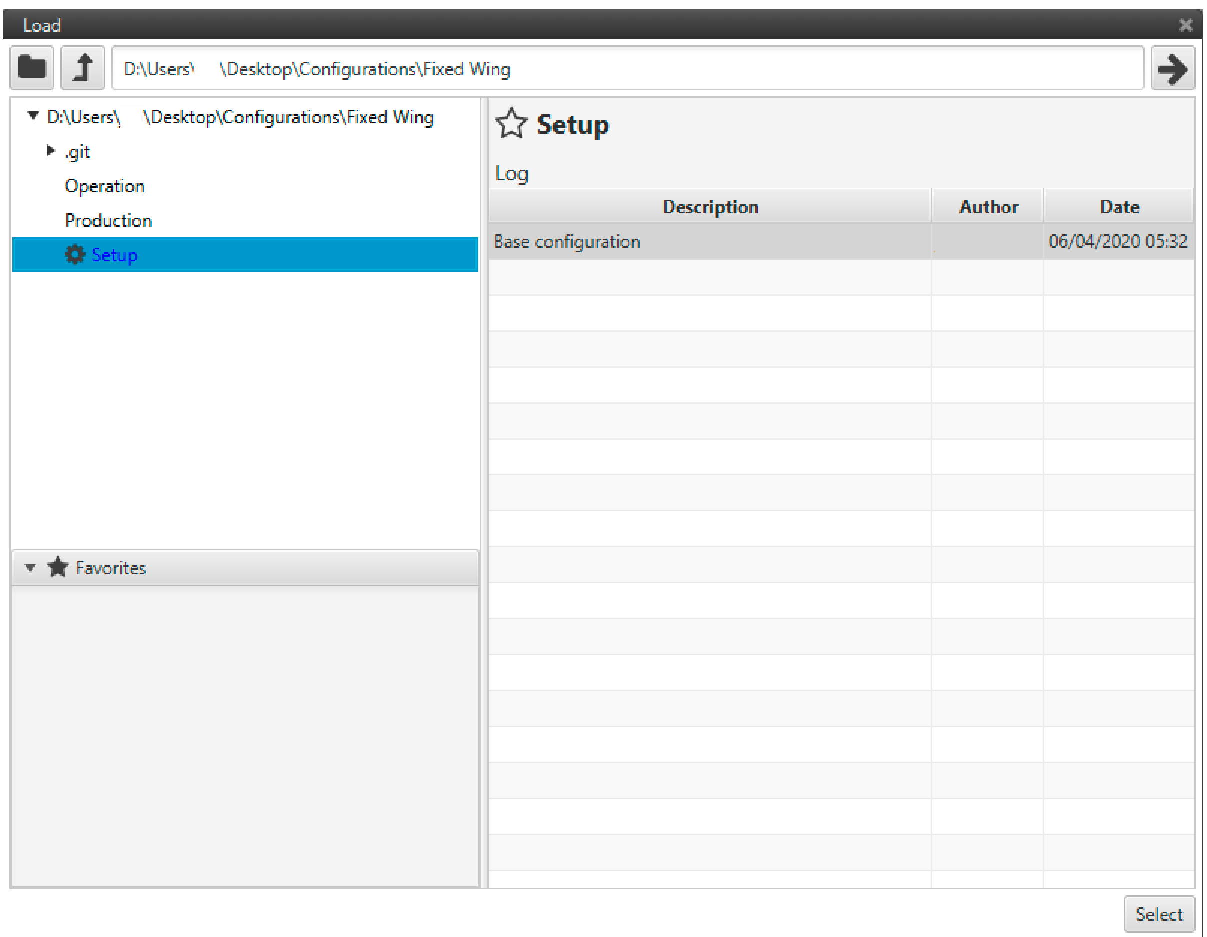
> **Warning:** Overwriting Calibration when loading a .ver file will overwrite the factory calibration of the sensors.

- **PDI:** By selecting PDI The user will be able to select the locations of each folder and load them.

> **Warning:** Loading Production files will overwrite the factory calibration of the sensors.



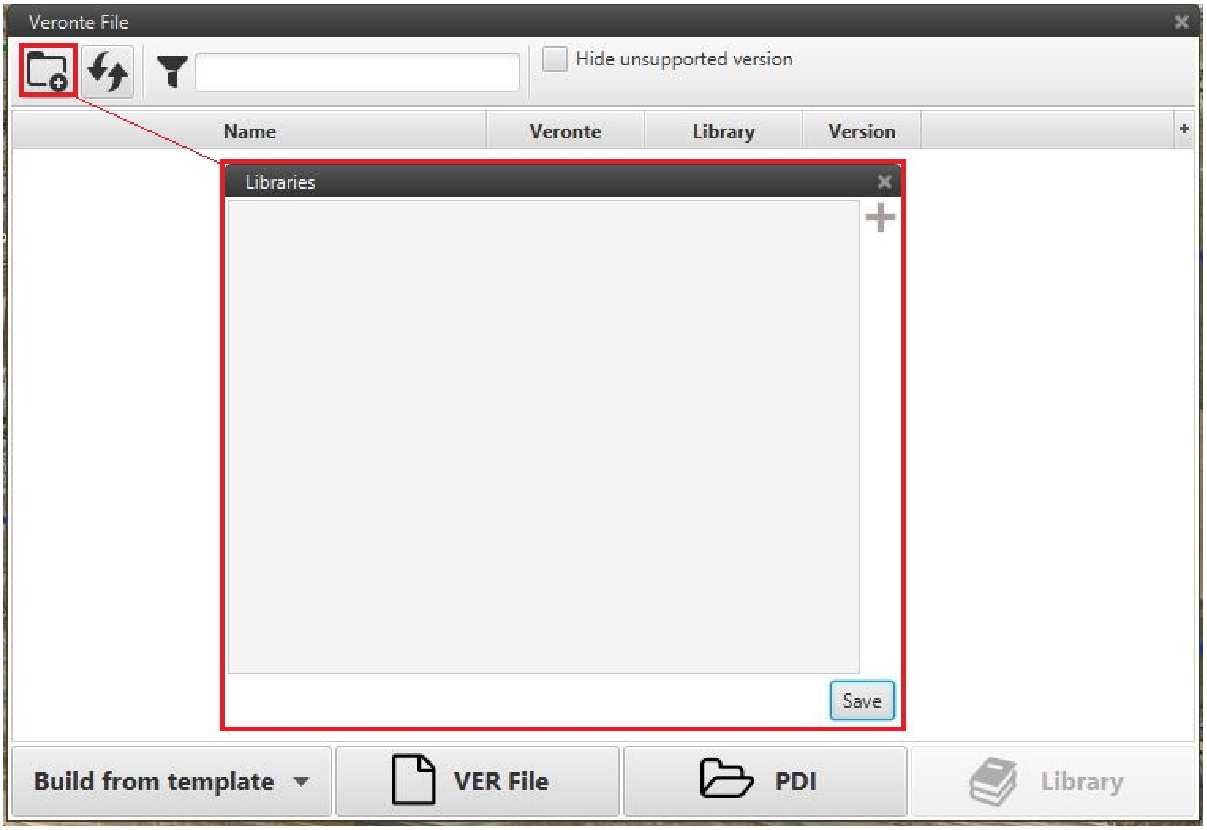
Import a Configuration File



Import a Configuration File

- **Library:** Loads a configuration from the library.

In order to include a folder as a part of the library select *Manage Libraries*. By incluing a folder to the library any .ver files in that folder will be displayed.



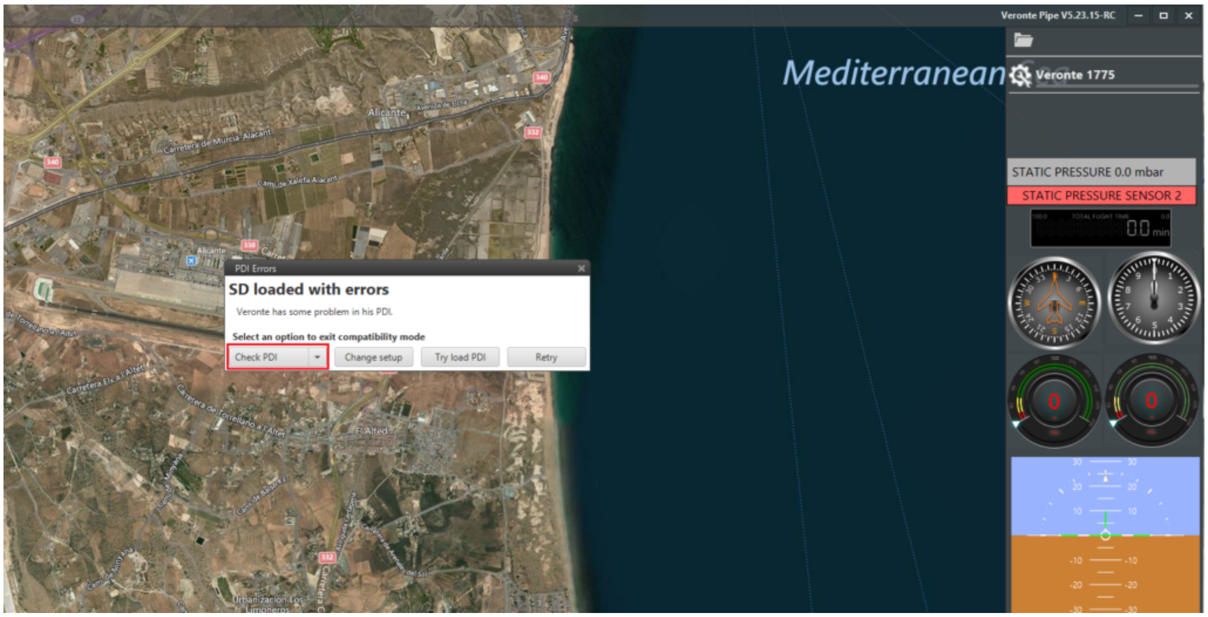
Library Menu

Warning: For each one of the configurations of the library appears its Version. About the Version, it is only possible to load configuration files whose version is the same or lower than the one of the software embedded on the autopilot. The colours are a way to show the compatibility: **Green** means fully compatible, **Yellow** means migration/PN adaptation and **Red** means not compatible.

Warning: Press *Save*,  , to load the configuration on the autopilot. Otherwise, it will be only loaded on the system.

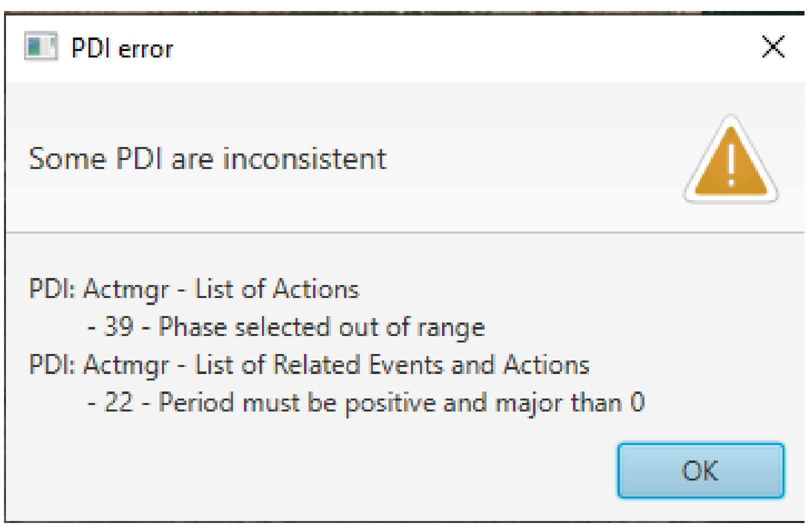
6.1.1.1 CHECK PDI

After Importing PDI files or migrating a configuration to a newer version, Veronte might enter in "SD loaded with error" state, which is a similar state to *Safe Mode*. When clicking on the unit affected it can be seen the following window appearing.



PDI Error Window

The option provided to the user are Check PDI (which generates an error display window), Change Setup (explained below), Try load PDI with which a new configuration can be loaded and Retry. When Checking PDI the window that appears is as follows:

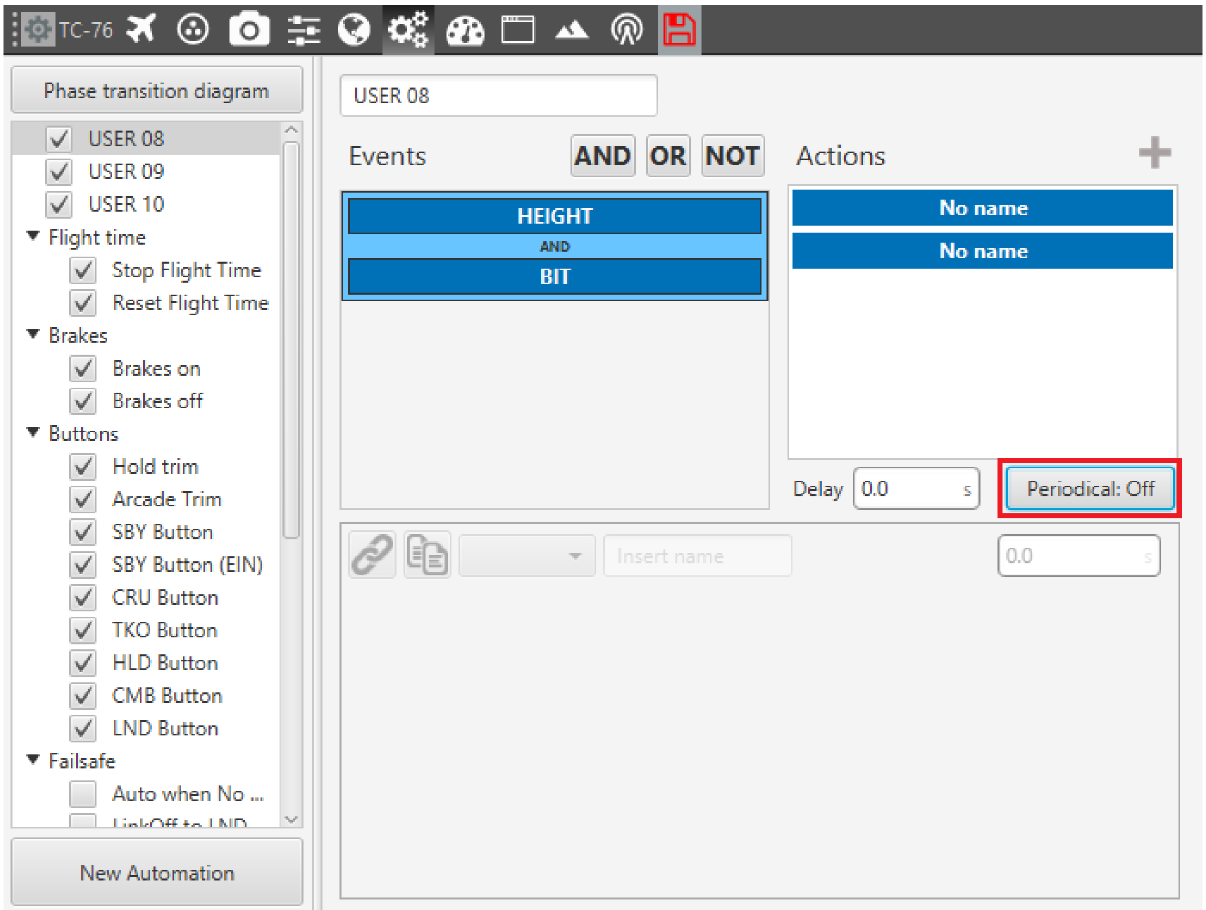


PDI Error Display

Now the user must find and correct the errors shown in the list displayed. For that, access Setup through "Change Setup" in PDI Error window.

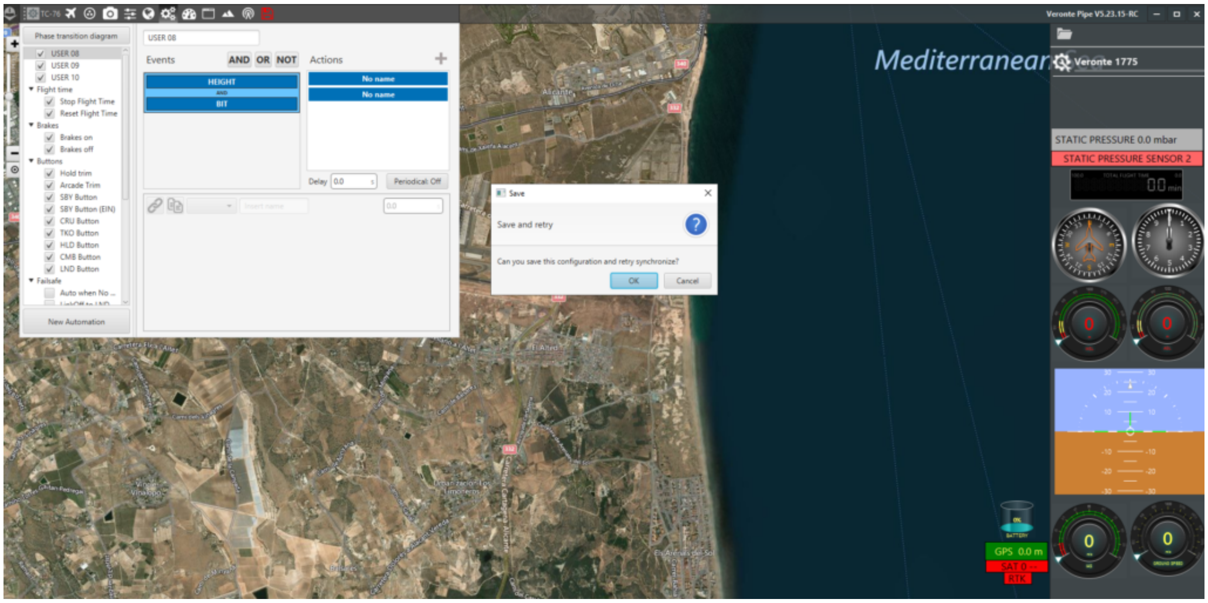
Check PDI setup

In this particular example, one of the errors reported was a Period in Events/Action (Automations tab) which has been disabled.



Correction done in PDI Check

Once all reported errors have been solved, press on the Red Save button on the Setup toolbar to finish the PDI Check.



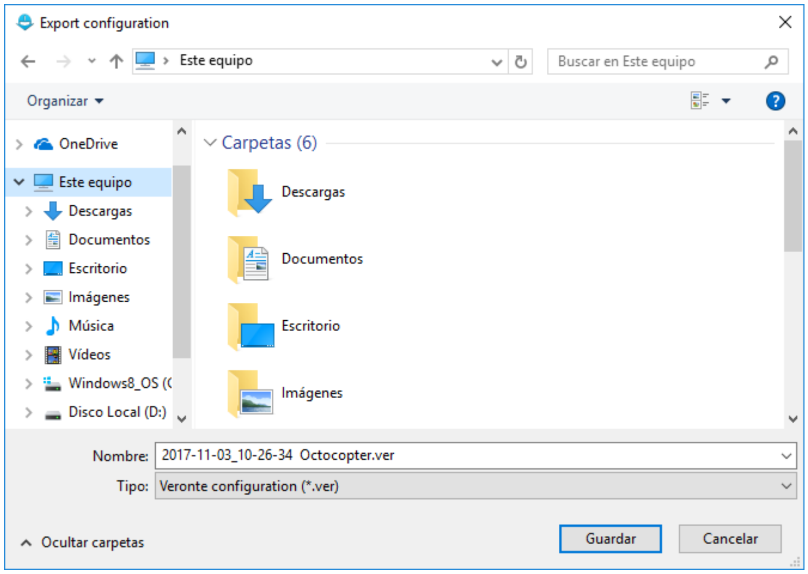
Save Setup after PDI Check

6.1.2 Export Configurations

When exporting a configuration, the file saved is a compression of a set of files with extension .bin, containing each one of them a part of the setup (mission, terrain information, servos, platform, communications and so on). This tool allows users to save configuration files from the autopilot or offline configuration.

In the side panel, click on  and then choose the kind of export to be done:

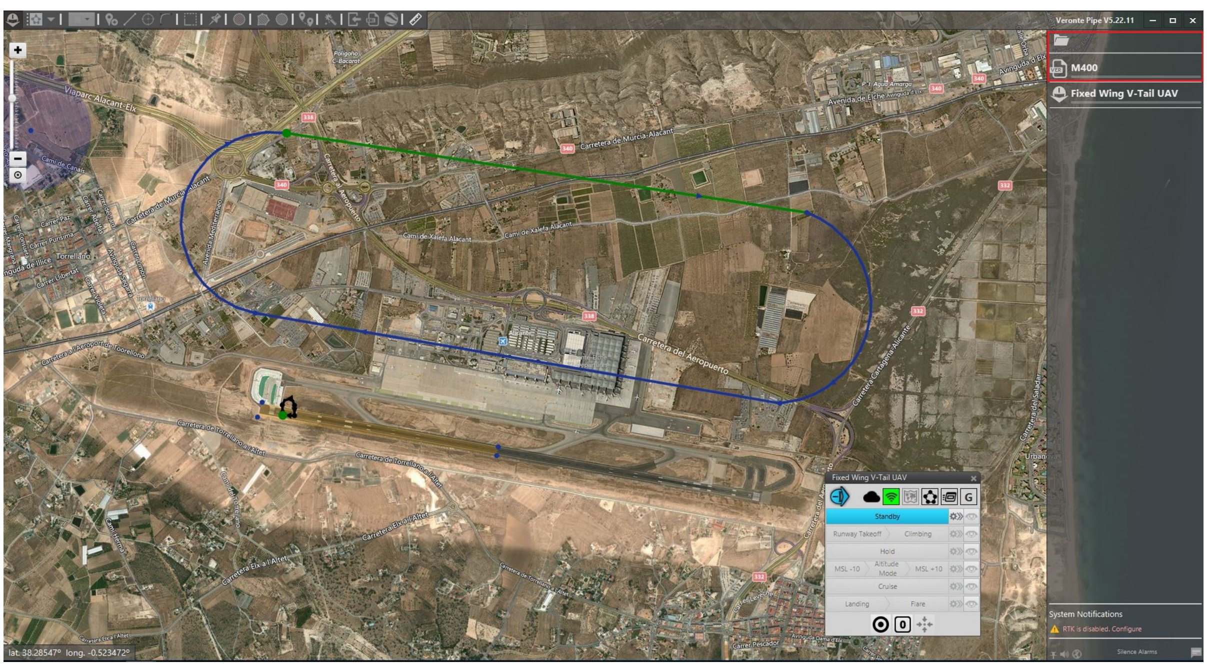
- **Export PDI.** Same as we have seen in VERONTE CONFIGURATION – *File Management*, a 3-screen window will appear to let you choose where to save all XML files.
- **Export .ver.** A new window will be displayed to select the file destination and set the name. By default, the name of the file will be the autopilot name plus the current date, but it is fully customizable by the user.



Exporting a configuration file

6.1.3 Offline Configurations

It is possible to load and modify a configuration when the autopilots are not connected. That configuration file can be changed and exported, and it is also possible to overwrite that one importing a new PDI file/.ver configuration.



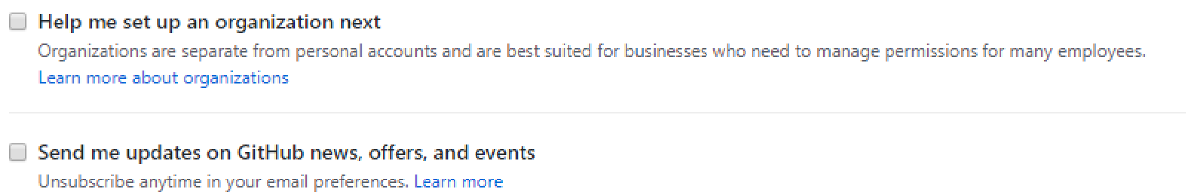
Offline Configuration

The other features available working with a configuration file are **Mission** and **Setup** configuration. This configuration file can be uploaded to the Autopilot following the steps described in the section *Import Configurations*.

6.1.4 Version Control

Embention offers the possibility to have a tool for version control over a customer's configuration. It is performed through the platform GitHub, where it is necessary to sign in before accessing the service. The steps to do it are:

1. Access GitHub.
2. Proceed through Step 1 with user data.
3. In Step 2, choose Free plan and decide upon the below options

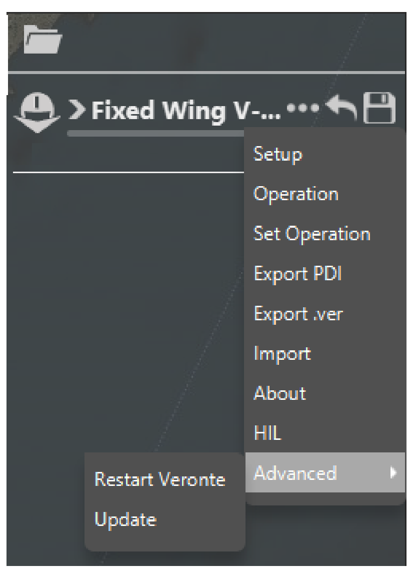


Step 2 Options

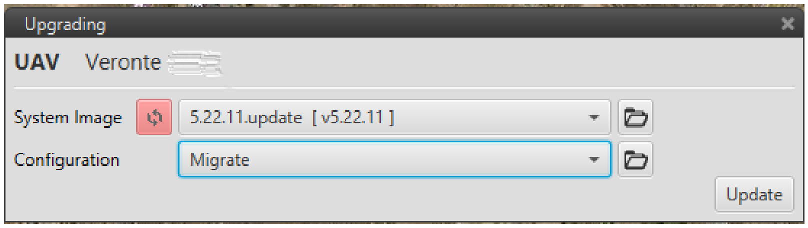
4. During Step 3, we recommend to "skip this step", but GitHub can be configured through these options freely.
5. Finally, send your Username and email of access to Embention so we can provide you with the relevant permissions for the correspondent repositories.

6.1.5 Update Onboard Software

To display the Update dialogue click on  and choose **Advanced – Update** on the side menu.



Advanced - Update Menu



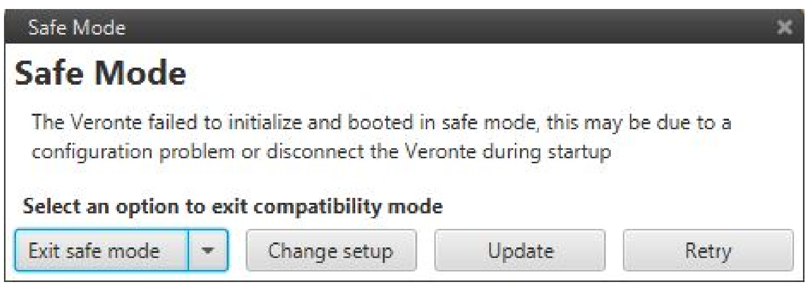
Update Veronte Autopilot

The different parameters to configure when updating the onboard software are detailed here:

- **System image:** here is where the version to be updated is selected. For instance, using *Pipe 5.22* the onboard version will be *5.22.X* (with 'X' being the last option available). If the .update file is stored in the computer it can be uploaded by clicking on the folder icon.
- **Configuration:** this option is used to decide which configuration will be loaded when the update process ends. There are different options: migrate will keep the current configuration for the new version and the folder icon will display the *Import Configurations*.

6.1.6 Safe Mode

Whenever there is an issue with connection, powering or configuration, the unit enters in Safe Mode and the following window is displayed.



Safe Mode

The options available are:

- **Exit Safe Mode.** The Veronte tries to initialise again in order to exit Safe Mode.
- **Change setup:** this option is used to load a configuration file into Veronte. The user has to select a configuration file, once selected a new window will be displayed in Veronte Pipe showing the version and identification of the configuration file and the autopilot where the file is going to be loaded.

Upload Configuration - Safe Mode

With this tool, the configuration file is loaded directly on the autopilot. Now Pipe is only a tool to load the file from the computer to the autopilot and the configuration parameters will not be shown in the software window before being loaded on the autopilot.

- **Update:** this option allows the user to Update the Veronte Unit, this is explained in *Update Onboard Software*.
- **Retry:** the unit will try to boot again.

This section explains the management of configuration files. Veronte Pipe allows users to import or export configuration files and Update Veronte Units from the Side Panel

Side Panel Options

The setup toolbar of Veronte Pipe is the section that contains the parameters to configure the system, which is formed by the autopilots (Air and Ground) and the software (Veronte Pipe).

6.1.7 CONFIGURATION FILES (.VER)

The .ver file contains the entire Veronte configuration and it means:

- **Aircraft configuration** (Sensors, Servos, Automations, Control, etc.)
- **Mission** (Terrain info, Waypoints, etc.)

The best way to manage .ver files is to keep a default version with the correct aircraft configuration which must contain no mission. In this way, it is possible to load the default config. when the user wants to create a new mission and, when it is complete, to save it with a different file name (normally the mission name or the mission site).

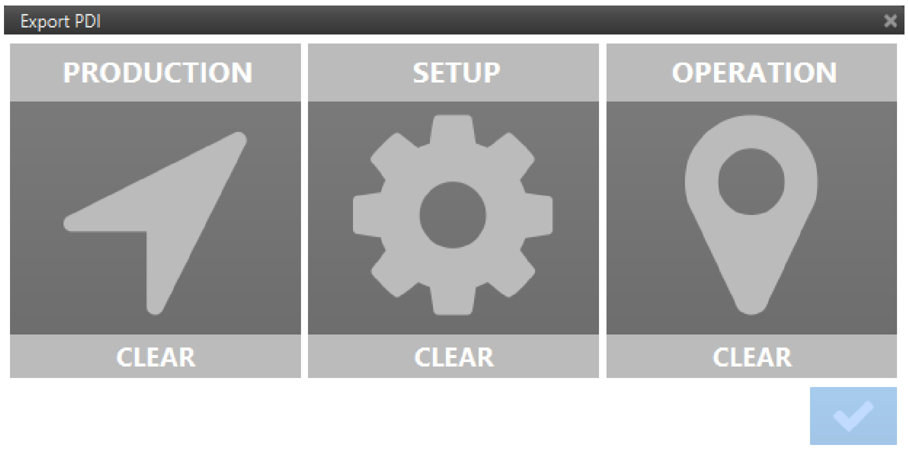
When the user wants to do the 'X' mission is sufficient to load the 'X' mission.ver file in the autopilot and the aircraft will be ready to perform the desired route.

6.1.8 PDI FILES

PDI files is a way of managing the configuration from Veronte, allowing the user for a modular control with improved version management. PDI files are split in:

- **Production.** Contains the configuration of the sensors, the calibration parameters, the active devices…
- **Setup.** Contains the configuration of the flight phases, guidance commands, control loops…
- **Operation.** Contains the operation values that are set before the operation starts (Cruise Speed, Altitudes…)

The optimal way is to work with a repository for version control, so each time the user exports the configuration there is a track of the changes done and the saving goes together with a commit message for later revision.



PDI Files management menu

6.2 Setup Toolbar

6.2.1 Veronte

This tab consists of the identification of the autopilot, where the user can define the name of the configuration in **Vehicle Name**.

Another option available is **Encryption**, the user can click on the shield icon, see section *Encryption* for more information.

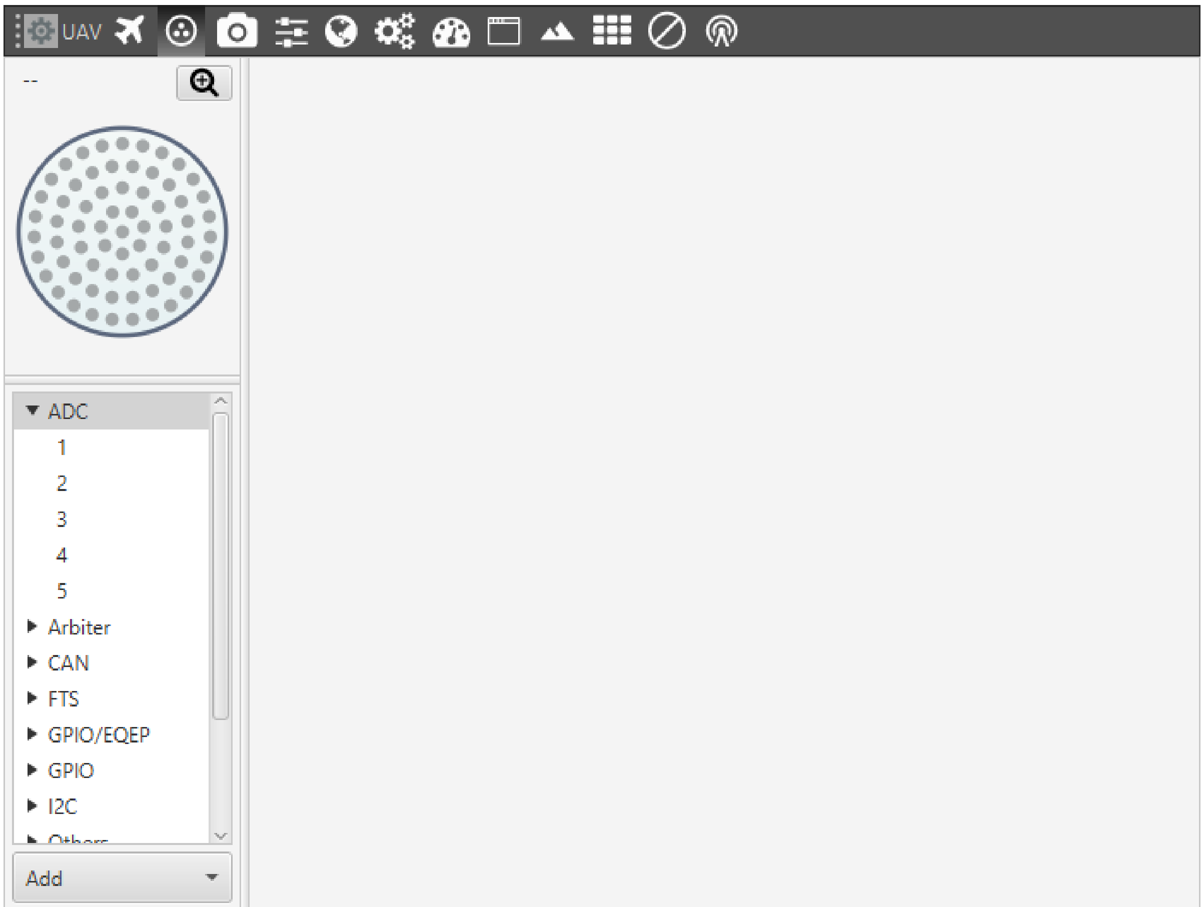
Autopilot Identification Data

6.2.2 Connections

6.2.2.1 ADC

ADC stands for Analog-to-Digital Converter. This connection is used by analog sensors. These sensors provide a voltage readout that needs to be converted into the actual measured vairable, e.g. temperature, fuel volume, etc.

Veronte autopilots are equipped with **5 connections** of this kind. To set them up, the user needs to go to the *Connections menu* and click on *ADC* on the left-hand side panel. Every *ADC* connection that is set requires an integer variable associated where the voltage readout will be stored. The maximum voltage of the ADC connection is 3 V.



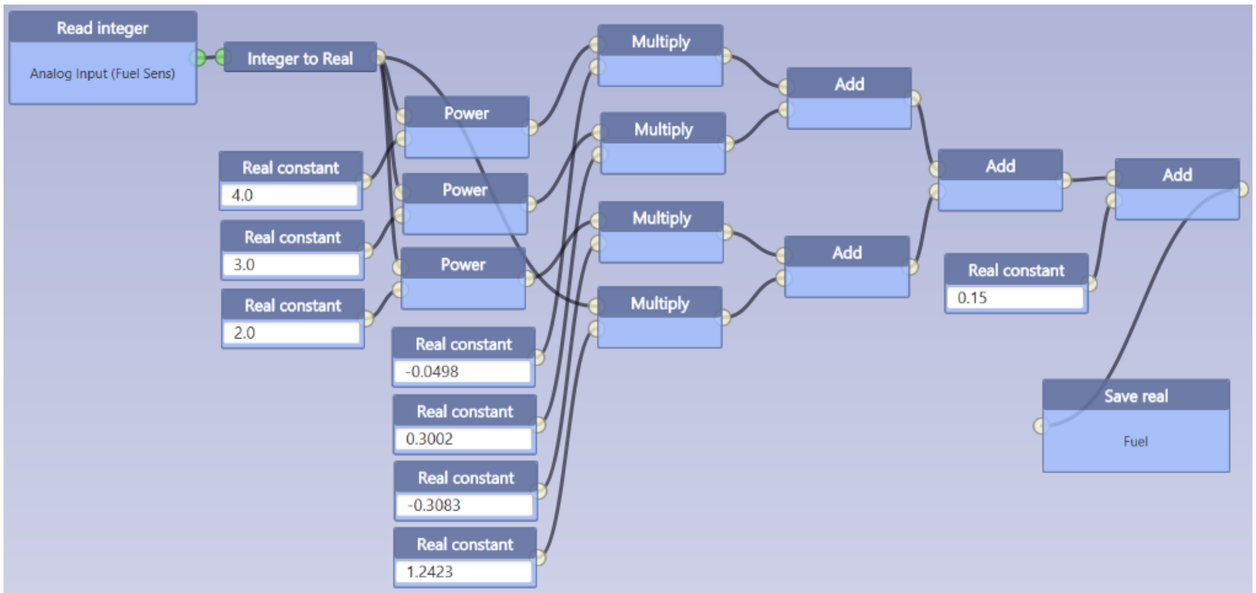
ADC Menu

To convert the input ADC value to the physical variable it represents the user needs to create a new operation – see more information in *Blocks*.

6.2.2.1.1 Application example

Let us consider a Fuel Level Sensor whose datasheet provides a direct relation of the voltage readout and the fuel volume (in L) through the polynomial $y = -0.0498x^4 + 0.3002x^3 - 0.3083x^2 + 1.2423x + 0.15$, where y is the fuel volume and x is the voltage of the sensor.

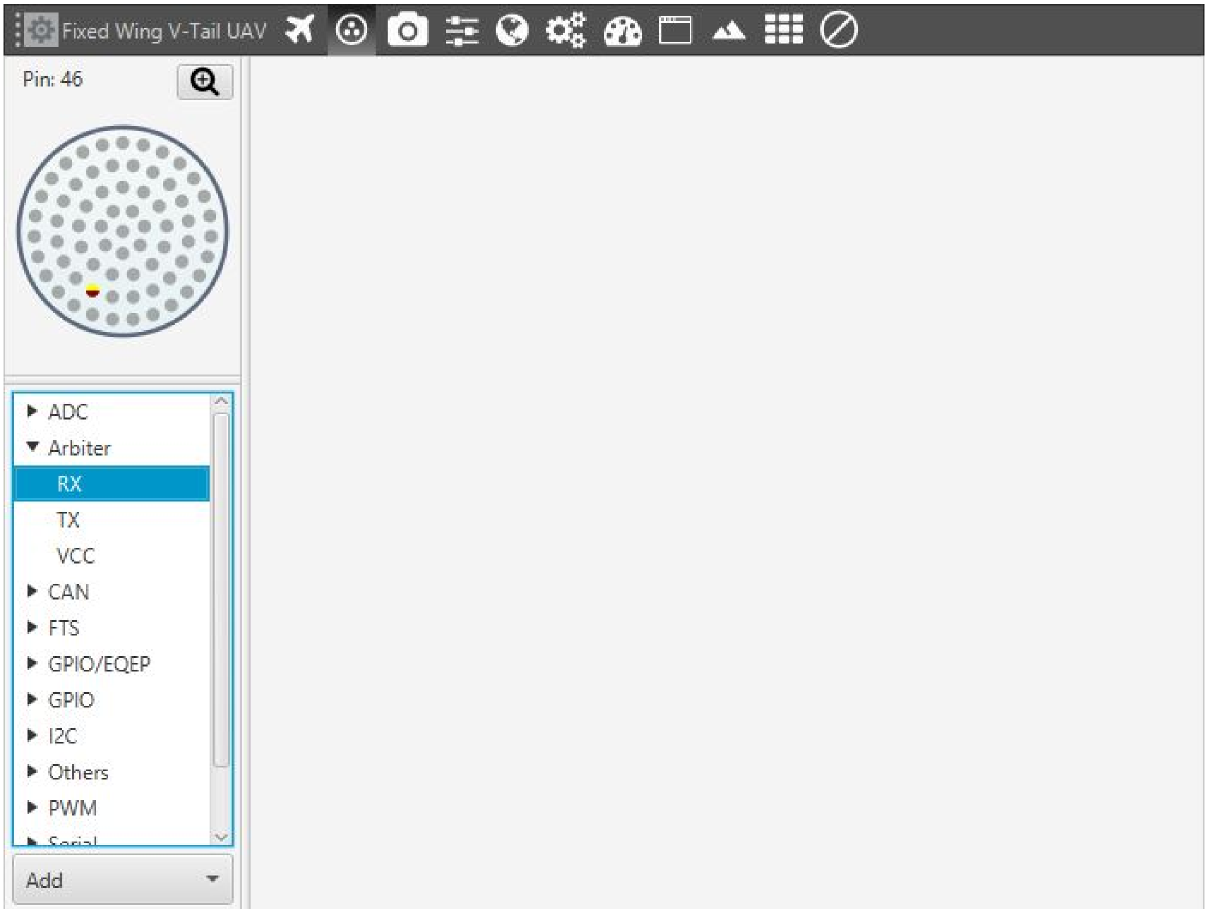
Creating a new operation in the *Blocks menu*, the above can be reproduced – see Figure below. Note that the ADC varialbe is first converted from integer to real, and then the polynomial is applied. The fuel remaining in the tank is saved in a *user variable*, which can be used for displaying or warning puproses.



ADC conversion for a Fuel Level Sensor

6.2.2.2 Arbiter - SuC

Pins 45 and 46 are dedicated pins to allow the uart communication with the Safety micro Controller (SuC). This microcrollers is in charge of monitoring the state of the main microcontroller and providing the Flight Termination Signals (FTS).



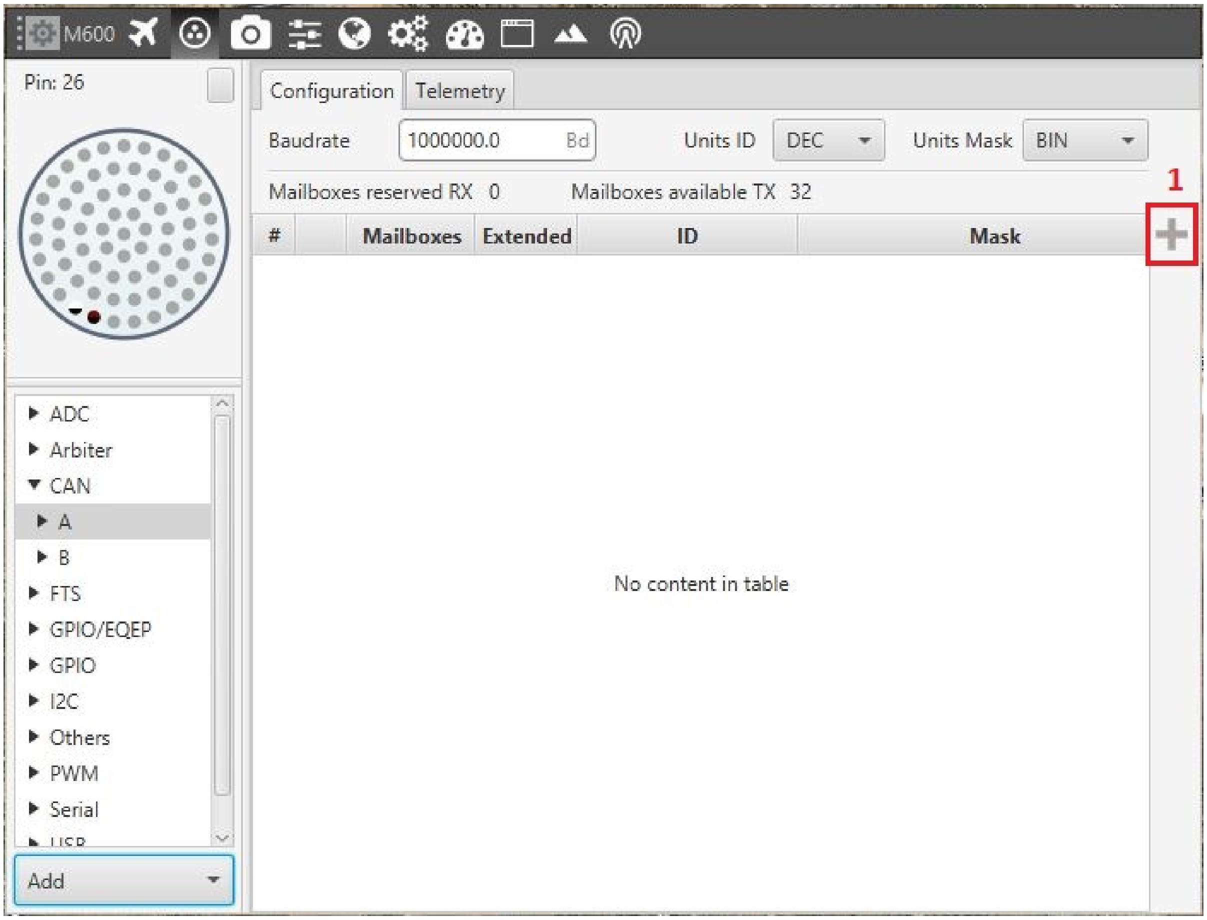
Connections – Safety microcontroller (SuC)

6.2.2.3 CAN

6.2.2.3.1 CAN Configuration

In this menu, it is possible to modify the Baudrate and the Mailboxes from the CAN bus.

When the user has to configure Veronte to receive data from the CAN Bus, it is mandatory configuring a Mailbox. In order to add a mailbox select (1).



CAN – Mailboxes

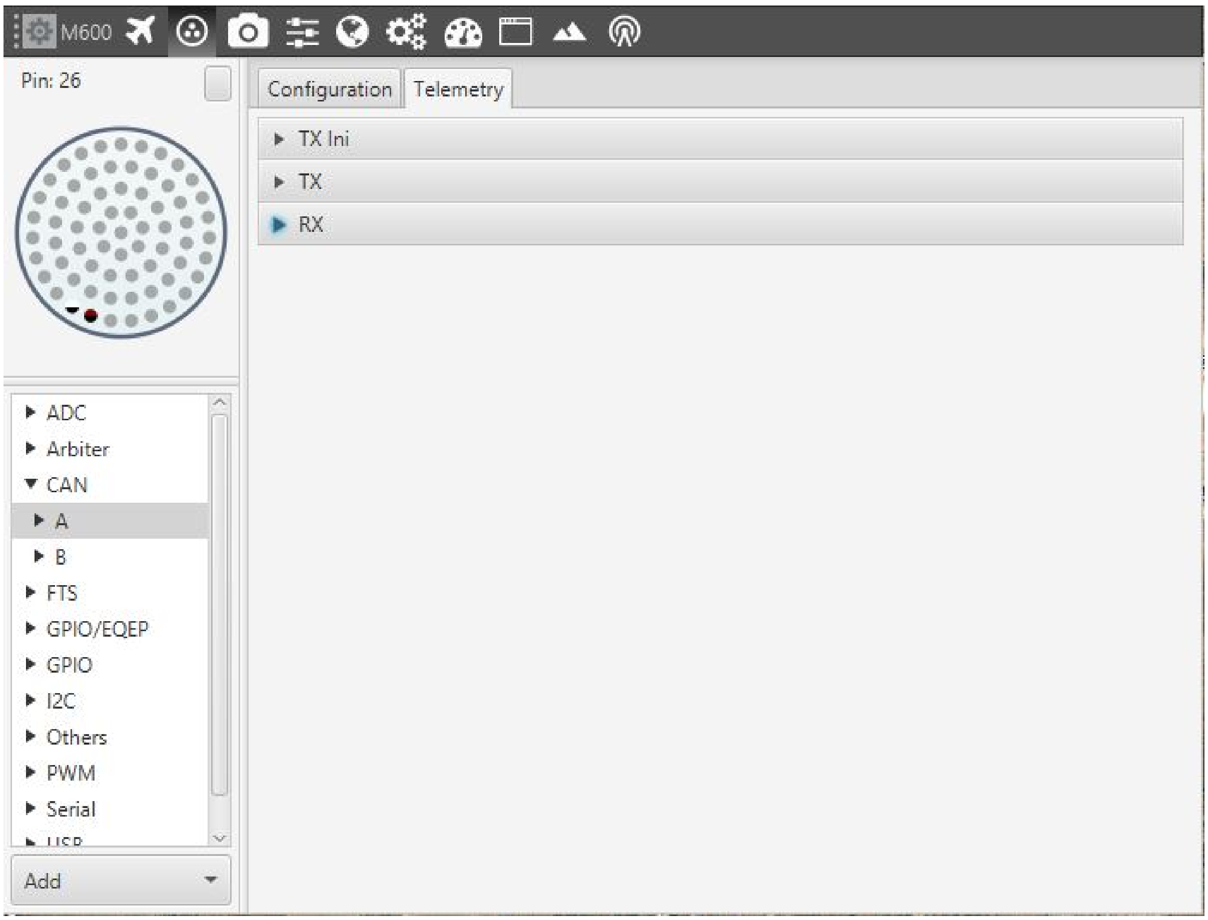
Mailboxes are used to save the data received, being possible to configure more than one. Veronte allows up to 32 mailboxes. The options available when adding a new mailbox are:

- **Mailboxes:** allows the user to identify a Mailbox with a determined configuration.
- **Extended:** frame format with 29 identifier bits.
- **ID:** identifier 11 or 29-bits (Extended), used to identify RX messages. The value set has to be decimal.
- **Mask:** This filter is configured for reception messages; received data will be stored on mailboxes where message ID coincides with mailbox ID. Mask adds some flexibility on the reception, when comparing message with mailbox data, only the value of binary digits configured as 1 on the mask will be taken into account. (e.g, for a configuration MASK: **11** 000 and ID:**10** 110 all incoming messages addressed to **10** XXX will be received in this mailbox).

6.2.2.3.2 CAN Telemetry

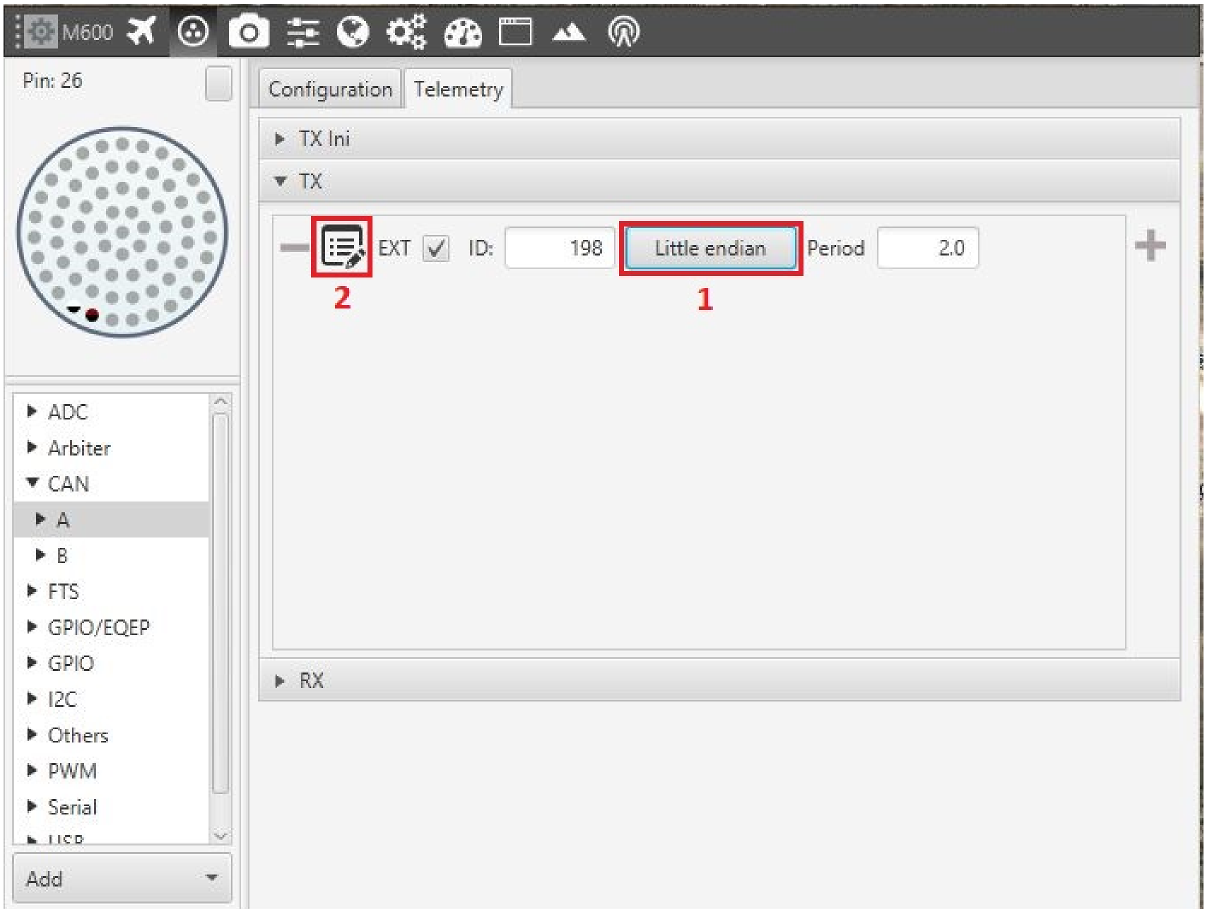
Interactions between system variables and CAN variables are managed from the telemetry configuration. In this section the following elements can be configured:

- **TX Ini:** used to configure transmitted messages that are only sent once at the beginning of the operation
- **TX:** used to configure transmitted messages through CAN bus.
- **RX:** used to configure the interaction between the variables read from the CAN bus and the variables of the system i.e, how they are stored.



CAN – Telemetry

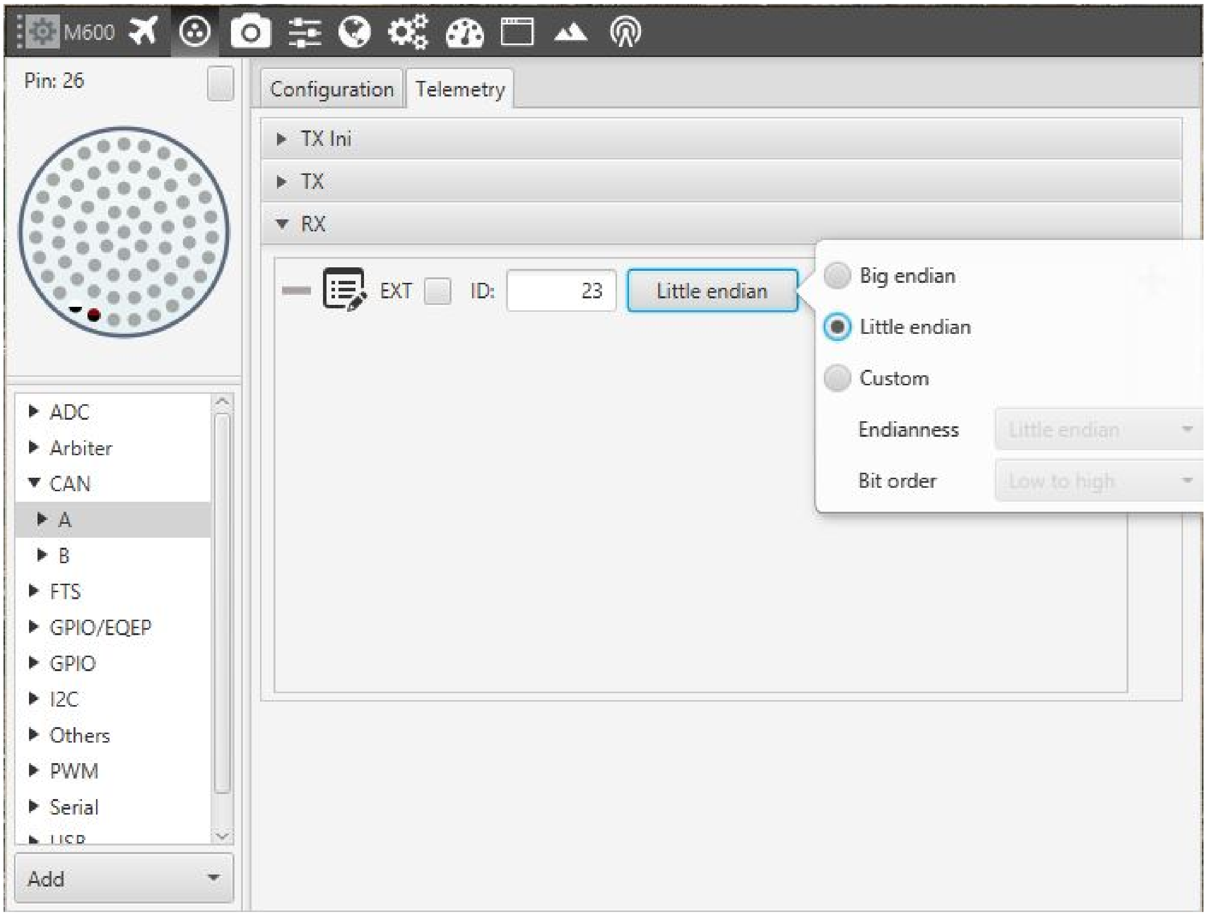
6.2.2.3.2.1 TX Messages



CAN – Telemetry - TX

In order to add a message to be sent press "+" and a new element will be added into the panel.

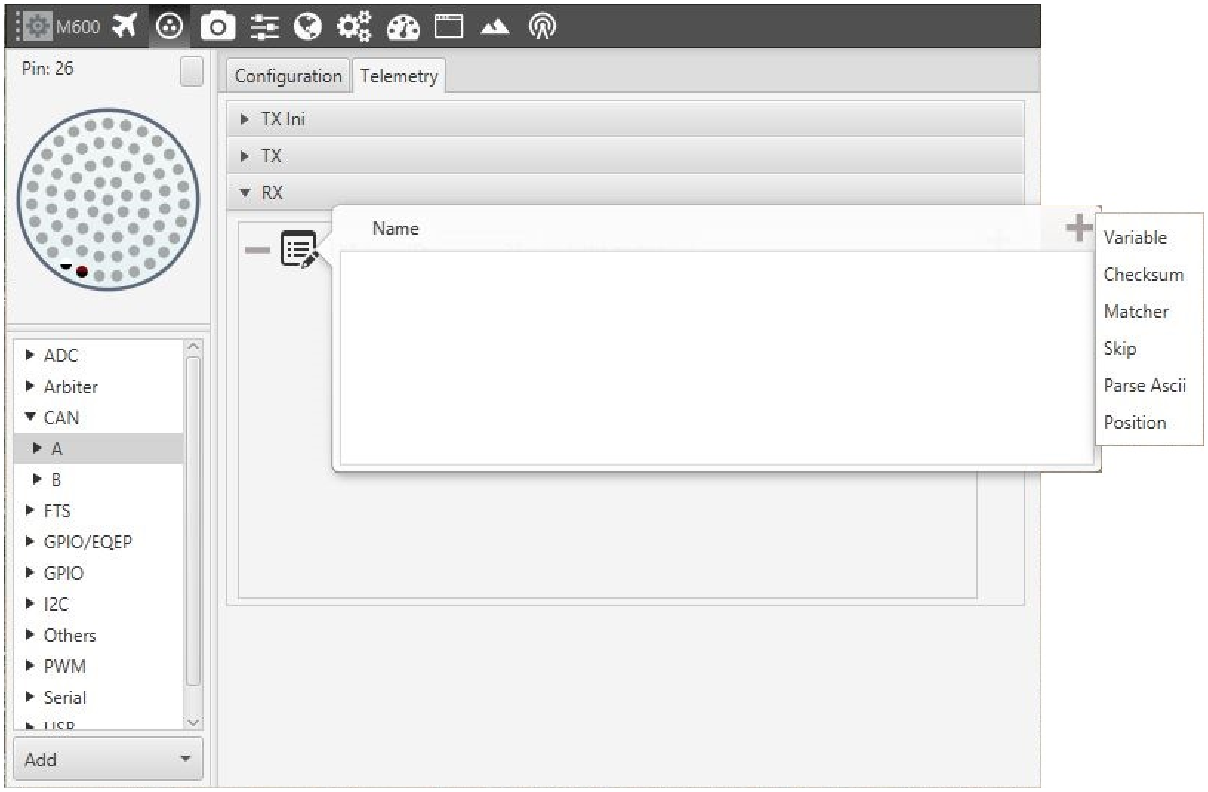
- **EXT:** enables the frame format with 29 identifier bits.
- **ID:** identifier 11 or 29-bits (Extended), used to identify TX messages. The value set has to be decimal as it was written on Configuration.
- **Period:** time in seconds between TX messages delivery.
- **Button 1:** open a new window to configure the endianness of the message, which indicates how the bytes that it contains are written:
  - **Big endian:** set the value from left to right
  - **Little endian:** set the value from right to left
  - **Mixed endian:** Any devices have this format. If you need to configure, please contact us.



CAN – Telemetry – RX Endianess

- **Button 2:** displays the menu to configure how the bits of the message are divided and sent.

There are six different elements that can be added when setting up a CAN message: *Variable*, *Checksum*, *Matcher*, *Skip*, *Parce Ascii* and *Position*.



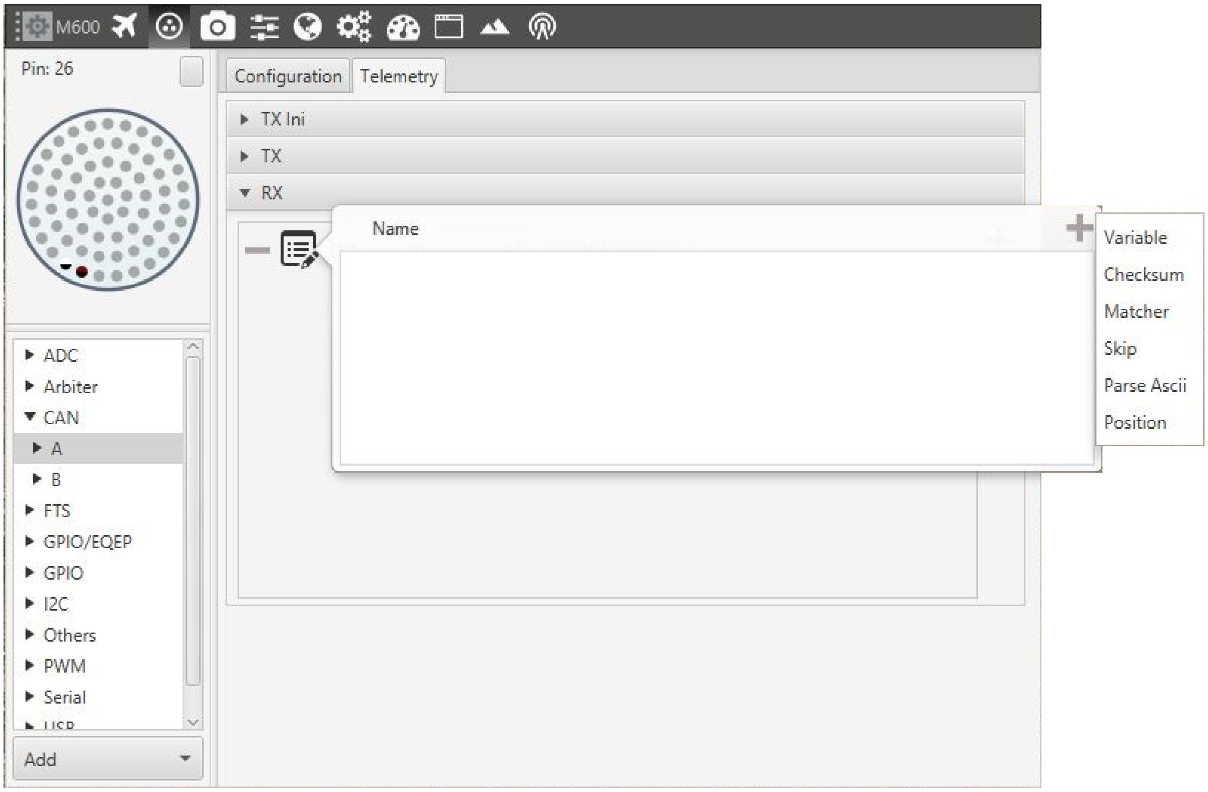
CAN – Telemetry – RX Message Confguration

6.2.2.3.2.2 RX Messages

Once a message is "captured" by Pipe (tab Configuration) and stored in the mailbox, its information can be read and saved in the software. To perform that, press "+" to create a new message and follow the steps explained for TX Messages, except by Period which is exclusive for TX.

6.2.2.3.3 CAN Messages

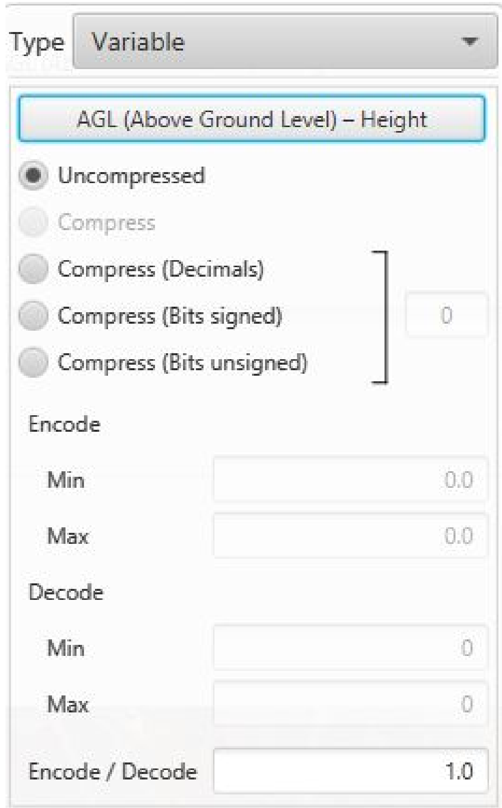
When using and configuring CAN messages either TX (transmitted) or RX (received), the following types of message are available.



Telemetry Panel

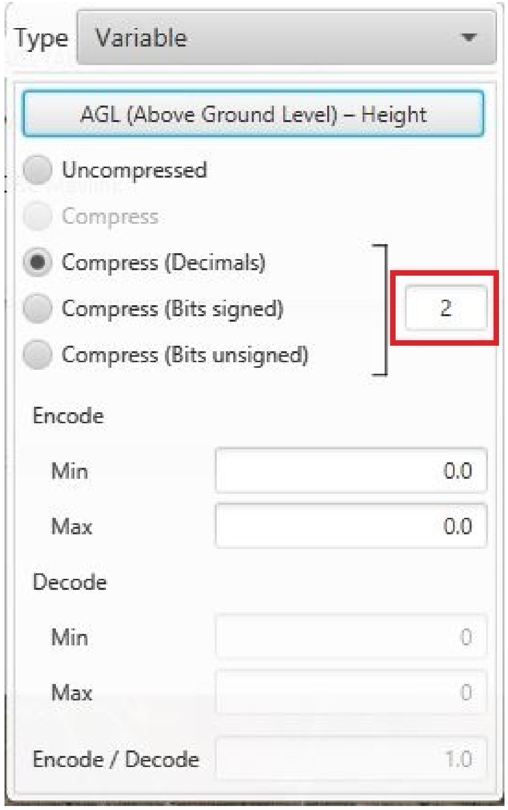
6.2.2.3.3.1 Variable

Used to store certain bits in a system variable (RX) or to send a certain variable through the CAN bus (TX).



Variable Configuration Menu

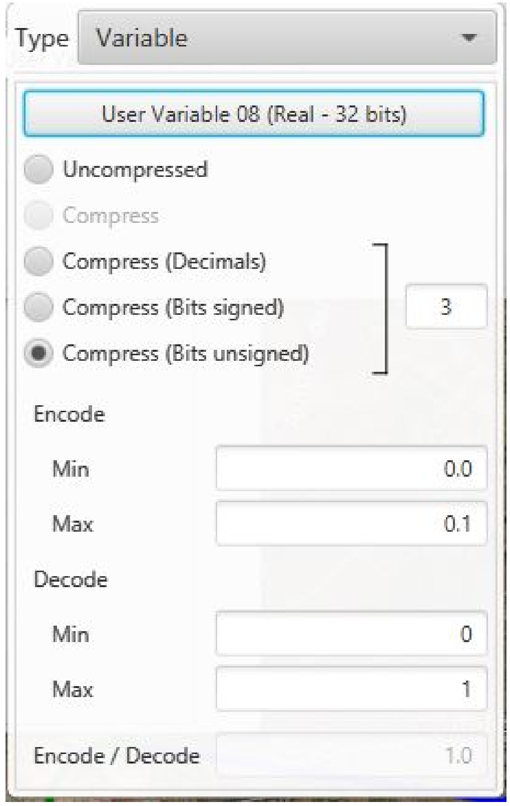
- **Type of Compression.** The first step is to configure which kind of compression will be used for this variable.
  - **Uncompressed.** The variable is taken in its full length, with no value modification. If it is a 32bits variable, it will take all 32bits from the CAN message directly.
  - **Compress.** The three subcategories of compressed Variable allow the user to input a digit, which is explained in the following lines:
    * **Compress (Decimals).** Sets the number of decimals to use i.e, if the AGL has a value of 500.43453, it will be compressed to 500.43.
    * **Compress (Bits signed).** Sets the number of bits, and these can be signed i.e, negative values are admitted. It is necessary that the user configures Encode/Decode options.
    * **Compress (Bits unsigned).** Sets the number of bits which will not accept negative values. It is necessary that the user configures Encode/Decode options.



Compress Decimal Selection

- **Encode/Decode.** These values are used to apply a scaling factor after the transformation from binary to decimal value.

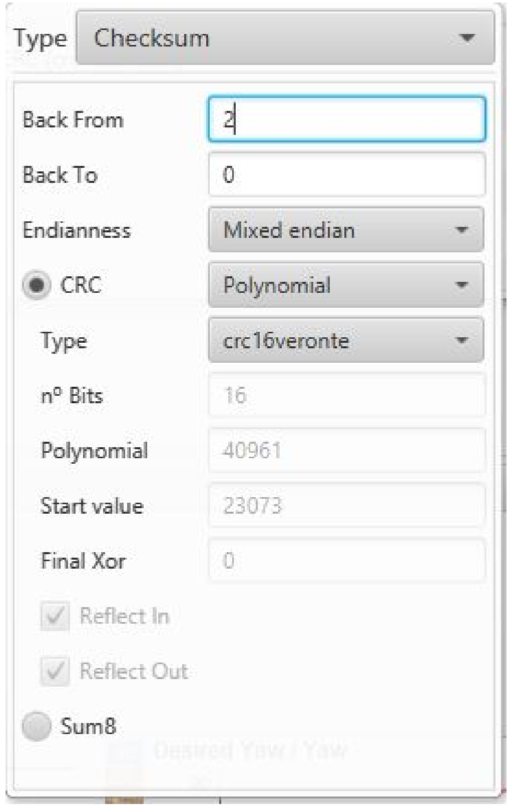
When reading data from a CAN bus it is common to have information about the message layout. In that case, let's consider that the first 3 bits correspond to a certain variable. If that variable is always positive, unsigned has to be selected, if it could be positive or negative select signed. The fields Encode/Decode are used to make the transformation from binary to decimal, corresponding Encode to the decimal value and Decode to the binary one. Considering the example shown in the following figure, the binary number is divided by 10 to make the transformation to decimal (and decimal multiplied by 10 to go to binary), so the variable represented here will go from 0 to 0.7 (as 3 unsigned bits can represent numbers from 0 to 7).



Variable Identification Example

6.2.2.3.3.2 Checksum

Sometimes, control codes are needed for preventing random errors in transmission, where a bits frame is operated and the result is sent to the receiver to check it. To do so the CheckSum option is used.



CheckSum

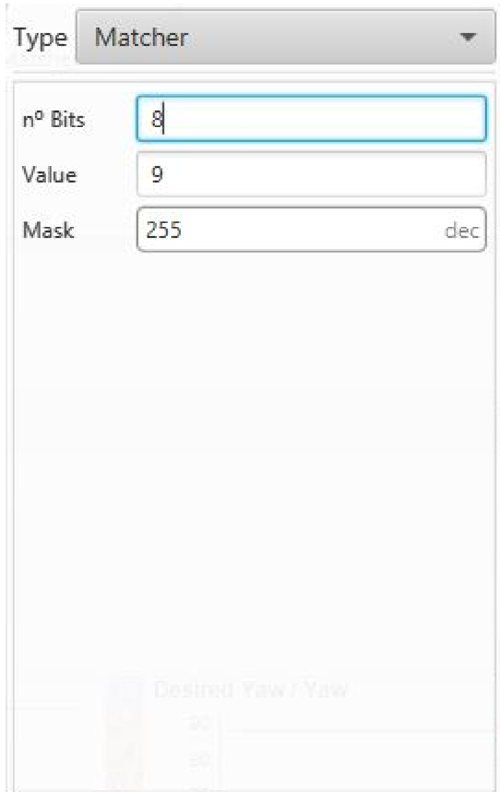
- **Back From:** Indicates that the CRC will be computed from the indicated byte (inclusive).
- **Back To:** Indicates that the CRC will be computed to the indicated byte (exclusive).

Note: Byte 0 it is referred to the first byte of the Checksum block. i.e for a message with 6 bytes where 2 bytes are the CRC. Back from = 4 and Back To = 0

- **Endianness:** Indicates how the bytes that it contains are read:
  - **Big endian:** Set the value from left to right
  - **Little endian:** Set the value from right to left
  - **Mixed endian:** For some special devices.
- **Type:** User can choose the type of CRC that will be applied.
- **CRC type.**
  - **Polynomial.** Select from a list of predefined Embention Veronte CRC. Last option is Custom, where fields as nº Bits, Polynomial, Start Value, Final Xor, Reflect In and Out can be defined. Check Polynomial CRC online for more information.
  - **Module.** Applies Module CRC.
  - **Mavlink.** EMbention has implemented the Mavlink checksum, used only for Mavlink protocol communications.
- **Sum 8.** The Sum 8 checksum is the 2's complement of the sum of all bytes. The checksum value, when added to the sum of all bytes, produces a zero.

6.2.2.3.3.3 Matcher

This option is used to send a constant value through the CAN bus in TX or wait for a particular value in RX.



Matcher

- **Nº Bits:** number of bits in which the matcher is performed.
- **Value:** sent/received value for the above nº of bits.
- **Mask:** it is automatically set when the nº of bits is assigned.

For example, a matcher of 8 bits with a value of 9 will be reading/sending: 0000 1001 .

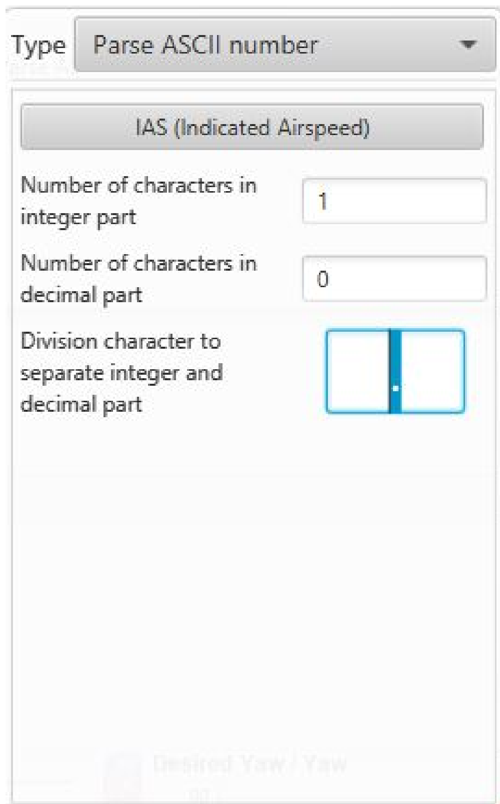
6.2.2.3.3.4 Skip

This option is used to discard a certain number of bits from the message (the maximum number of bits that can be skipped with a single "Skip" are 32). This tool can be used when there are variables incoming that are from no interest for the user, not loading unnecessary information into the system.

6.2.2.3.3.5 Parse ASCII

Parsing ASCII is used when the protocol required is of this kind. Only for RX. The variable set at the top is where the ASCII will be saved.

ASCII protocol is used for transforming a character array into decimal values. For such task, the user needs to define the number of characters in the integer and the decimal parts, as well as defining which is the division character. See how to introduce all this information in the picture below.

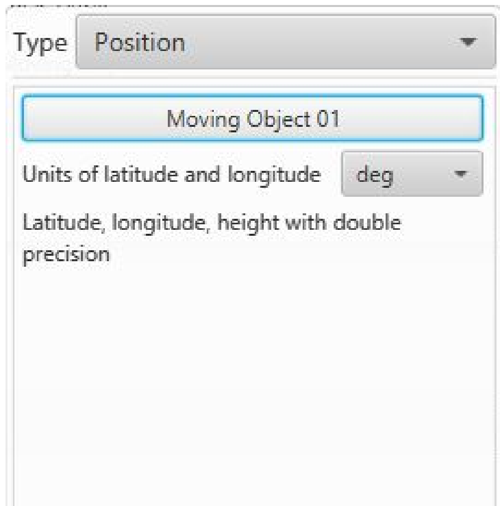


Parse ASCII Menu

6.2.2.3.3.6 Position

Position is used to input/output a data set with a particular format. When created, the user can only choose from Moving Object variables.

The window display below is the configurable menu, where it can be chosen between degrees and radians as units. The information stored is the WGS84 coordinates in the following order: Latitude, Longitude and Height. All of them are stored with double precision.



Position Menu

6.2.2.3.4 Arbiter CAN protocol

This section describes the CAN Bus protocol actually in use for the communication between the **Arbiter** and the Autopilots in **Redundant Veronte Autopilot**. Veronte is able to use both Standard CAN and Extended CAN messages. In this first integration step, the messages are configured following the Standard CAN communication protocol.

6.2.2.3.4.1 CAN parameters

The following Standard CAN parameters are described according to the Veronte Pipe configuration:

- **Baudrate:** 1 [Mbps].
- **TX Ini:** in this section is set the Start Message. **Period** has a value of 1 [s].
- **TX:** this section has to be configured as it is shown below. **Period** has to be set to 0.1 [s]
- **RX:** None.

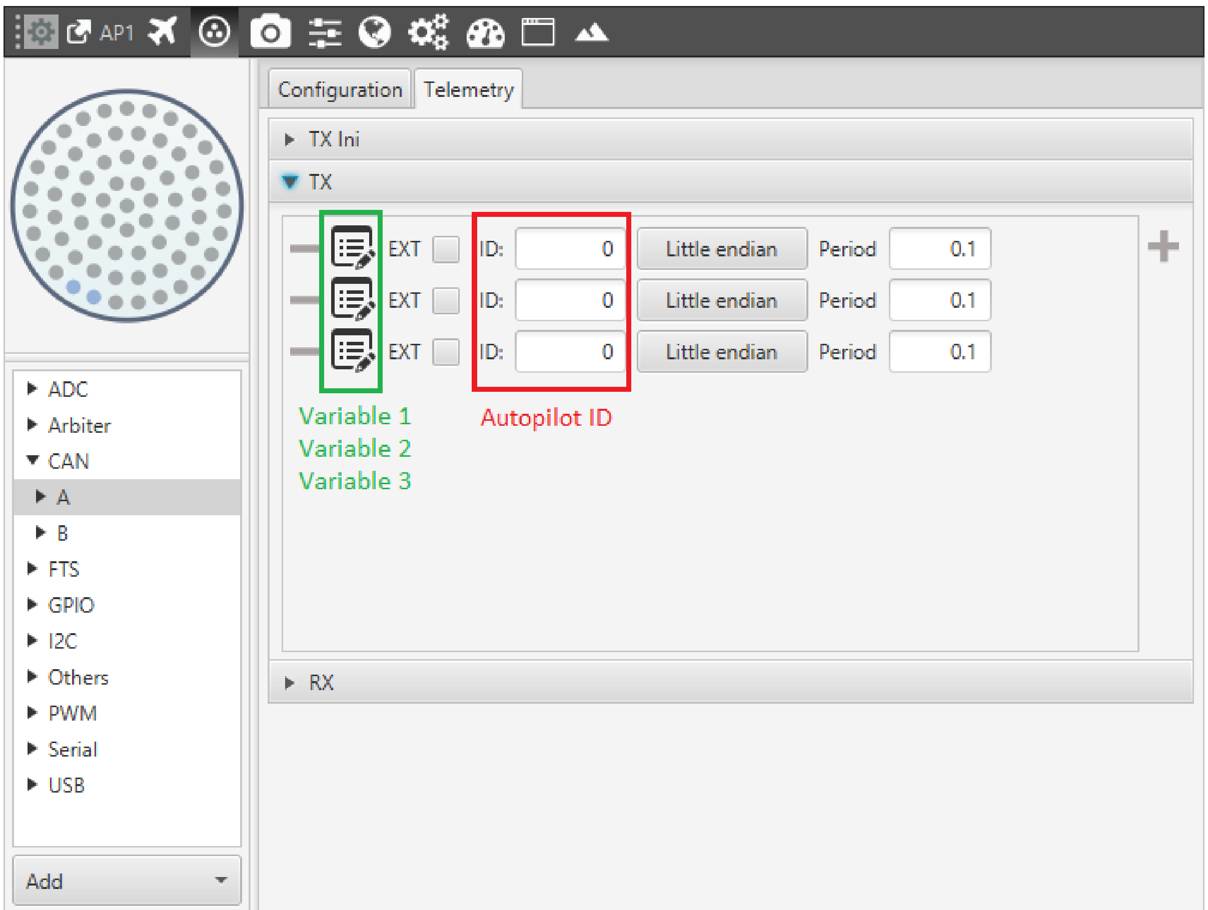
6.2.2.3.4.2 CAN message structure

The message that each Autopilot (Veronte or External) connected to the Arbiter must send through CAN Bus has to be as follows:

Table 1: Can Message Structure

Name	Variable 1	Variable 2	Variable 3	Start Message (TX Ini)
Endianness	Little Endian	Little Endian	Little Endian	Little Endian
ID	Autopilot ID	Autopilot ID	Autopilot ID	Autopilot ID

- **Variable X:** variable sent to the Arbiter. In this first integration phase, they are Roll, Pitch and Yaw angles.



Variable X – Veronte AP 1

- **Start Message:** the arbiter will start to arbitrate only when it reaches the start message from all Autopilots. The start message must be like the following:

Table 2: Start Message Structure

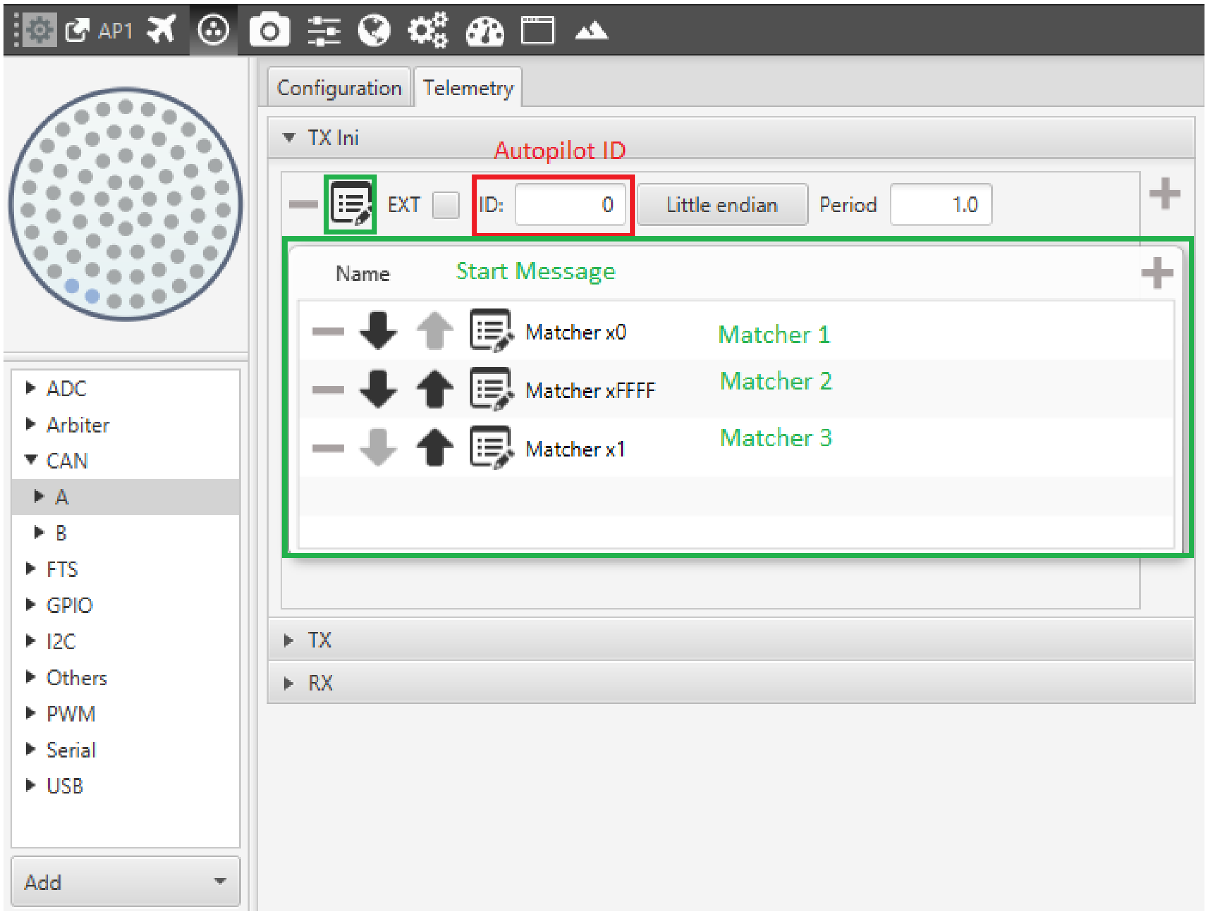
Name:	Matcher 1	Matcher 2	Matcher 3
Bits Number	16 bits	16 bits	1 bit
Value	Autopilot ID	65535	1

Where:

Table 3: Autopilot ID Number

Name:	Veronte AP 1	Veronte AP 2	Veronte AP 3	External AP
Autopilot ID	0	1	2	3

The following image shows this configuration using Veronte Pipe and Veronte AP 1.



Start Message – Veronte AP 1

- **Endianness:** In this case, all the messages are configured with a Little Endian structure.
- **ID:** The ID of the Autopilot that is sending the message (value from 0 to 3).

6.2.2.3.4.3 Single message structure

The structure of the messages including the single variables sent from the Autopilots must be configured like the following:

Table 4: Variable Message Structure

Matcher 1	Matcher 2	Variable
16 bits	16 Bits	32 bits

Where:

- **Matcher 1:** Autopilot Identifier. In this case, the value must be from 0 to 3.

Table 5: Autopilot ID Number

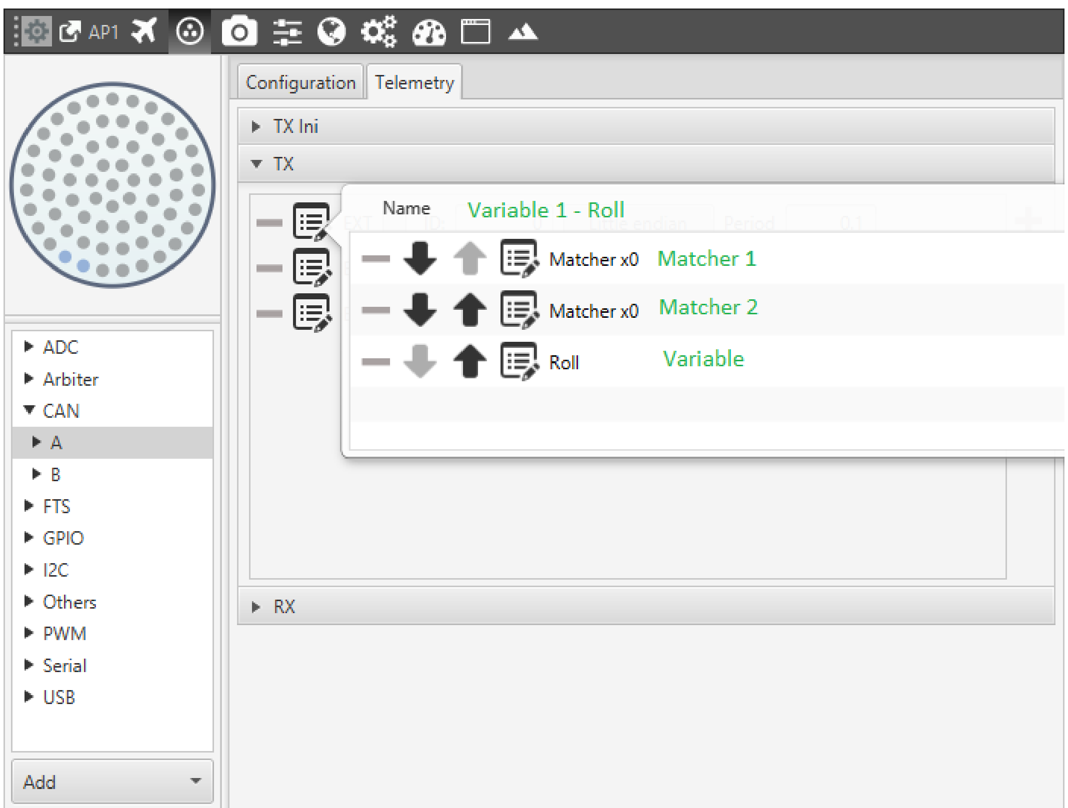
Veronte Autopilot 1	Veronte Autopilot 2	Veronte Autopilot 3	External Autopilot
0	1	2	3

- **Matcher 2:** Value ranging from 0 to 2 meaning the variable ID.
- **Variable:** the variable that the Arbiter will receive and monitor. In this first integration step the variables are:

Table 6: Variable ID

Variable	Description	Range [rad]	Variable ID	Bits	Type
Roll	Roll Angle measured from Veronte Autopilot in real time in Body Axes.	(-,)	0	32	Float
Pitch	Pitch Angle measured from Veronte Autopilot in real time in Body Axes.	(-,)	1	32	Float
Yaw	Yaw Angle measured from Veronte Autopilot in real time in Body Axes.	(0, 2)	2	32	Float

The following figure shows the configuration for Variable 1 – Roll, using Veronte AP 1.



Variable 1 – Roll

The variables can be modified in the Autopilot CAN configuration using Veronte Pipe. To do this it is sufficient to change the variables in the single message in both the Autopilot configurations.

6.2.2.3.4.4 External autopilot simulation

When the External Autopilot is not connected to the Redundant network, it is possible to simulate its presence allowing the Arbiter to start the arbitration. To do this, it is sufficient to add a message sent from one of the active autopilots through the CAN Bus that contains the Start message of the External Autopilot (ID: 3). The following example represents this kind of message sent from Veronte Autopilot 1 (ID: 0):

Message	Autopilot	ID	Variable ID	Type	Value	Description
1	0	0	Variable	Roll Angle	Roll Angle	variable of Veronte Autopilot 1
2	0	1	Variable	Pitch Angle	Pitch Angle	variable of Veronte Autopilot 1
3	0	2	Variable	Yaw Angle	Yaw Angle	variable of Veronte Autopilot 1
4	– TX Ini	0	65535	Start Message	1	Start message of Veronte Autopilot 1
5	– TX Ini	3	65535	Start Message	1	Start message of External Autopilot

External Autopilot Structure

Once the Arbiter starts to arbitrate, the External Autopilot will be automatically discarded and the process will be carried out using the Veronte Autopilots only.

6.2.2.3.4.5 Can protocol for 4-autopilot configuration

When the system in configured integrating four Autopilots (including then: three Veronte Autopilots, one Arbiter and one External Autopilot) the CAN Protocol structure does not change. In this case, the arbiter will expect to receive the Start Message from 4 Autopilots with different ID value that will go from 0 to 3.

The user must take into account that:

- the ID order is not important (Veronte 1 can be configured using ID:2 and vice-versa).
- the Arbiter does not need any information about the Autopilot type (External or Veronte).
- when the number of the Autopilots changes, the Arbiter software has to be modified.
- when the number of the monitored variables changes, the Arbiter software has to be modified

The following tables represent a possible configuration of the messages sent from four Autopilots using three monitored variables.

- Veronte Autopilot 1:

Message	Autopilot	ID	Variable ID	Type	Value	Description
1	0	0	Variable	Roll Angle	Roll Angle	variable of Veronte Autopilot 1
2	0	1	Variable	Pitch Angle	Pitch Angle	variable of Veronte Autopilot 1
3	0	2	Variable	Yaw Angle	Yaw Angle	variable of Veronte Autopilot 1
4	– TX Ini	0	65535	Start Message	1	Start message of Veronte Autopilot 1

Message Structure AP1

- Veronte Autopilot 2:

Message	Autopilot	ID	Variable ID	Type	Value	Description
1	1	0	Variable	Roll Angle	Roll Angle	variable of Veronte Autopilot 2
2	1	1	Variable	Pitch Angle	Pitch Angle	variable of Veronte Autopilot 2
3	1	2	Variable	Yaw Angle	Yaw Angle	variable of Veronte Autopilot 2
4	– TX Ini	1	65535	Start Message	1	Start message of Veronte Autopilot 2

Message Structure AP2

- Veronte Autopilot 3:

Message	Autopilot	ID	Variable ID	Type	Value	Description
1	2	0	Variable	Roll Angle	Roll Angle	variable of Veronte Autopilot 3
2	2	1	Variable	Pitch Angle	Pitch Angle	variable of Veronte Autopilot 3
3	2	2	Variable	Yaw Angle	Yaw Angle	variable of Veronte Autopilot 3
4	– TX Ini	2	65535	Start Message	1	Start message of Veronte Autopilot 3

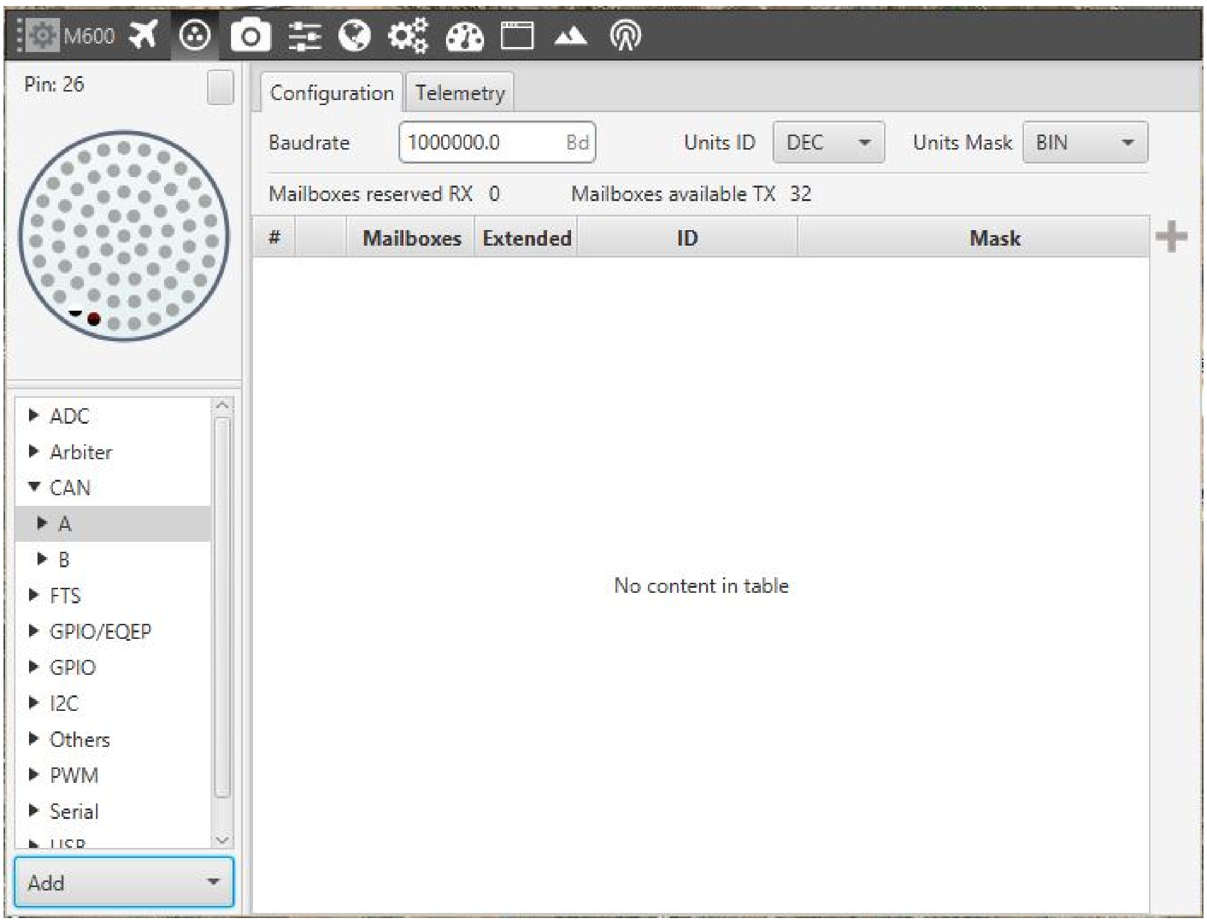
Message Structure AP3

- External Autopilot:

Message	Autopilot	ID	Variable ID	Type	Value	Description
1	3	0	Variable	Roll Angle	Roll Angle	variable of External Autopilot
2	3	1	Variable	Pitch Angle	Pitch Angle	variable of External Autopilot
3	3	2	Variable	Yaw Angle	Yaw Angle	variable of External Autopilot
4	– TX Ini	3	65535	Start Message	1	Start message of External Autopilot

Message Structure External

Controller Area Network is a message-based protocol. Veronte includes two CAN bus connections (A-B), being both configurable.

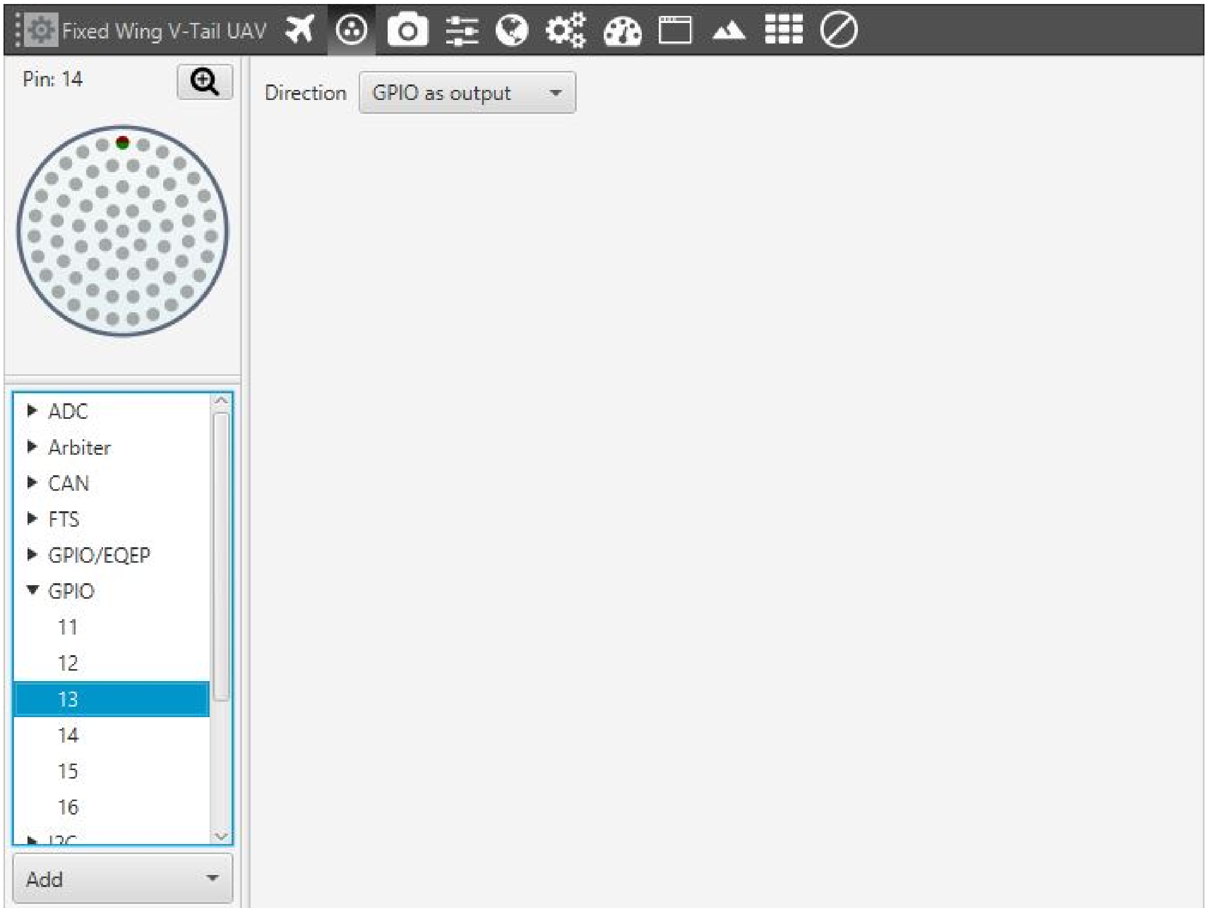


Connections – CAN

This connection allows the user to communicate with other devices. Each of them will be on the same bus, being possible to configure Veronte to transmit (TX) or receive (RX) data from other devices.

6.2.2.4 GPIO/PWM

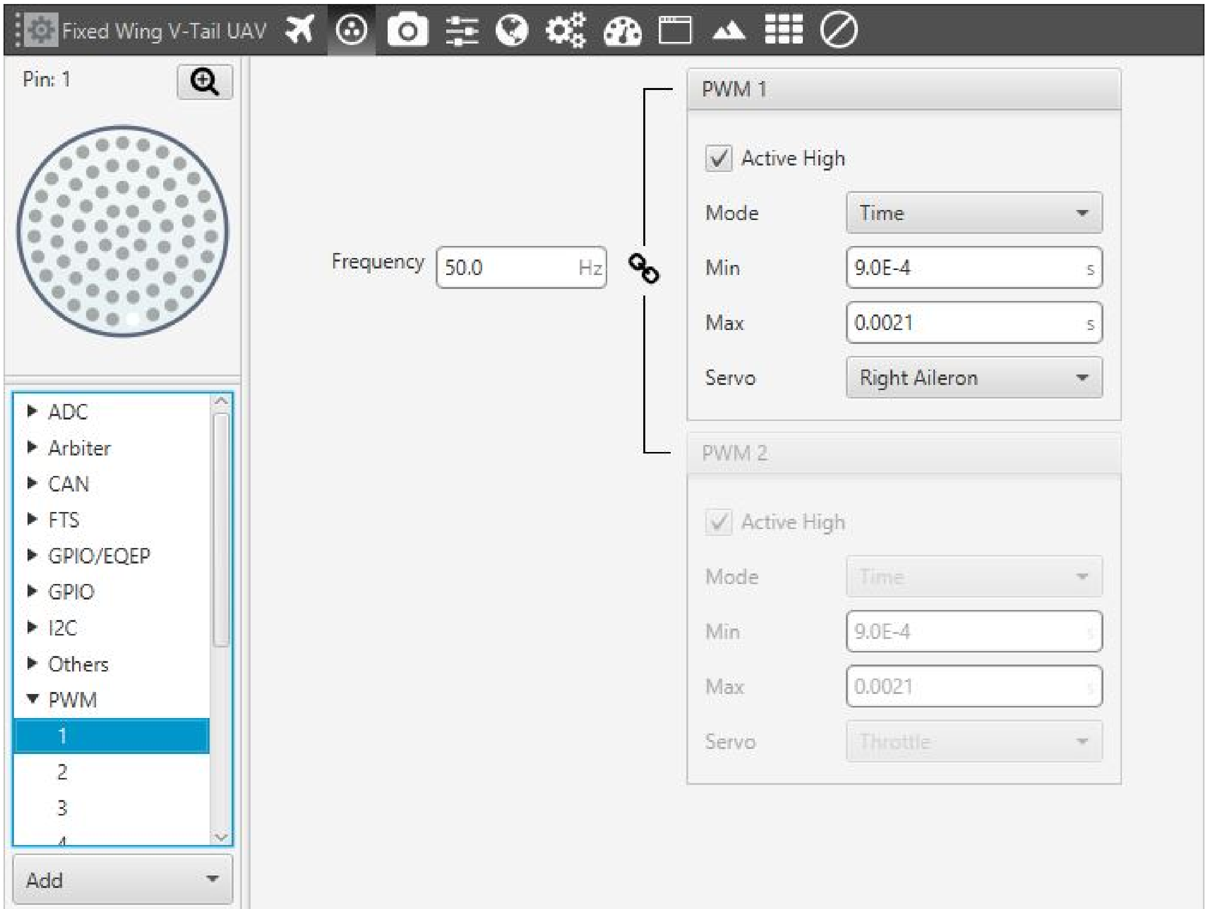
Output pins produce PWM or GPIO signals that are used to move the different servos and actuators of the platform.



GPIO Menu

6.2.2.4.1 GPIO

A GPIO (General Purpose Input/Output) is a generic pin that can be configured as an input or output pin. When this option is configured as an output pin, the value sent will be different from the one sent if it was a PWM. GPIO pins admit up to 4 different states: ON (a continuous signal of value 1, made by 3.3V), OFF (a continuous signal of value 0, made by Ground), PULSE ON (a single pulse of value 1, with a width specified in seconds) and PULSE OFF (a single pulse of value 0, with a width specified in seconds). The configuration of the pin output value is done with an action *Output* in *Automations*.



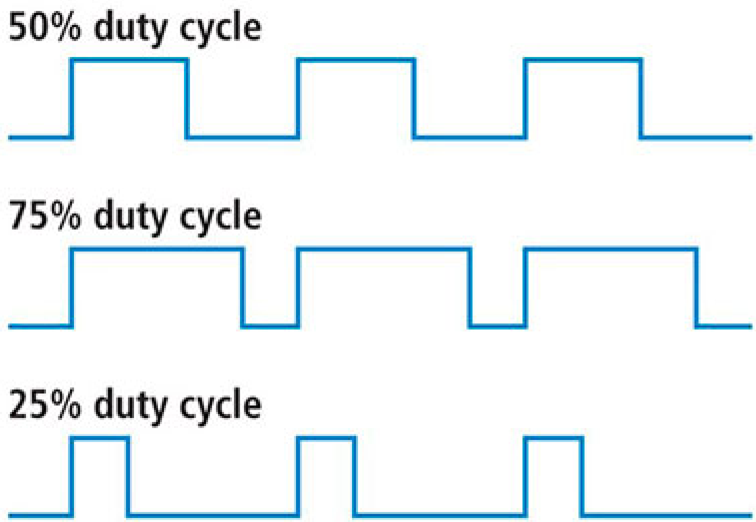
PWM Menu

6.2.2.4.2 PWM

The acronym PWM corresponds to Pulse Width Modulation. Veronte sends a pulse with a certain width that is received by the servo/actuator, and according to the width of such pulse, it changes its behaviour. A wide pulse will correspond to a big movement and a narrow one to a small movement.

The **Min** and **Max** parameters are the pulse width values that will make the servo/actuator go to its lowest and highest position. As an example let's consider the servo of an aircraft elevator, a pulse sent by Veronte of 0.9 ms will correspond with the lowest point of the servo range (-30 degrees for example). On the other hand, a pulse of 2.1 ms will make the servo go to its top position (for example 30 degrees).

The option **duty cycle** (select mode/duty cycle) is a different way of indicating the pulse width. Now the value indicated is a percentage which corresponds to the relation between the pulse width over the total period of the sent signal. So a 100% duty cycle will correspond to a signal with a constant value of 1, while a 0% duty cycle implies a constant signal with value 0. Between this two extremes, the pulse width can vary as in the examples shown in the following figure.



Duty Cycle

The option **Servo** is used to select which servo of the platform will be wired to that pin of Veronte connector, so the signal sent through that pin will go to it. The name of the "Actuator Output **S_** " can be changed to other more identifiable according to the real actuator such as right or left aileron, see section *System Variables*.

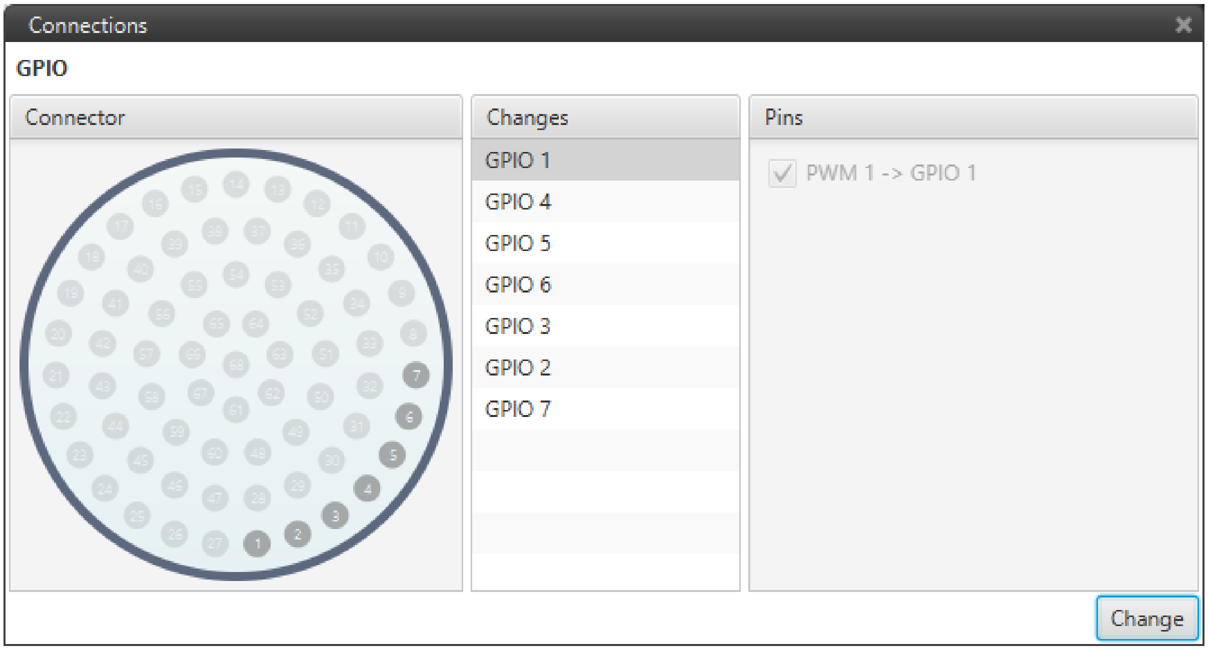
The option **Frequency** determines the period of the pulses sent by the autopilot. The PWM is built in pairs inside the autopilot, and that is why the frequency is indicated in pairs, i.e when the frequency of PWM 1 is changed, the one of PWM 2 also changes. The following table shows the PMW pairs as configured in Veronte autopilot.

PWM Pairs	
PWM 1	PWM 2
PWM 3	PWM 4
PWM 5	PWM 6
PWM 7	PWM 8
PWM 9	PWM 10
PWM 11	PWM 12
PWM 13	PWM 14
PWM 15	PWM 16

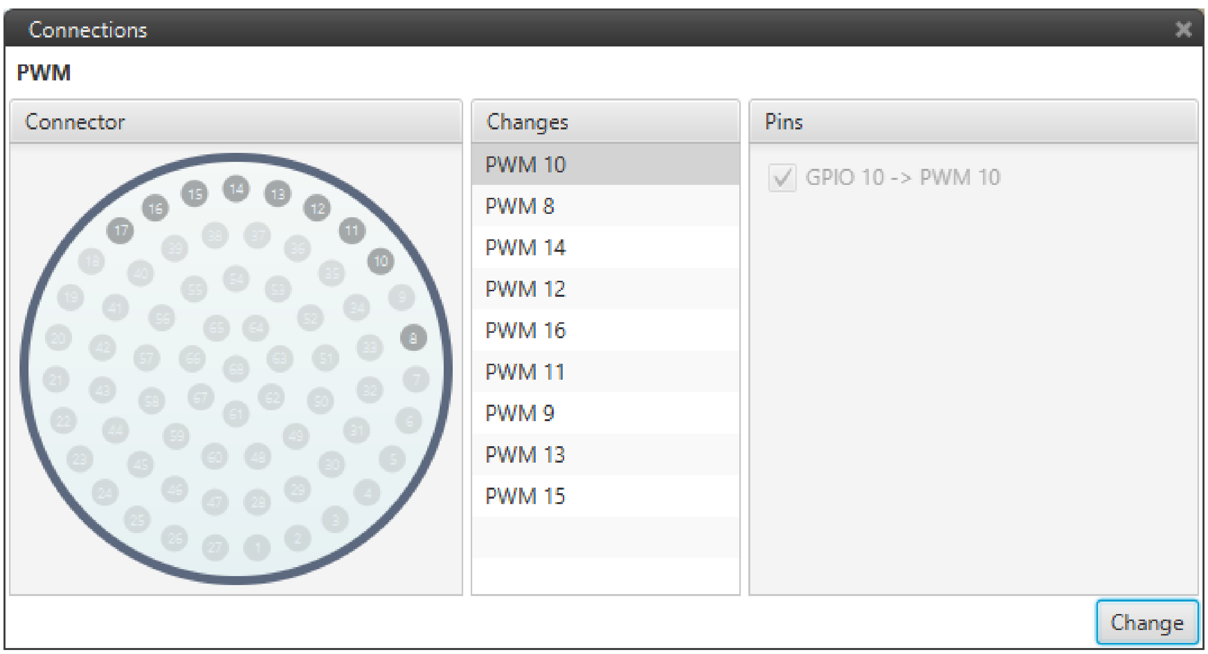
To sum up, a PWM is a signal which consists of a series of pulses having a width determined by a percentage over a range specified by the parameters **Min** and **Max**. On the other hand, the GPIO is a signal with a constant value (1,0) or with a single pulse (1,0).

6.2.2.4.3 Add

Veronte admits up to 16 I/O PWM/GPIO signals. To configure a pin as PWM or GPIO, click on **Add** and select **PWM** or **GPIO**. The following windows will be displayed according to the option selected.



GPIO Selection

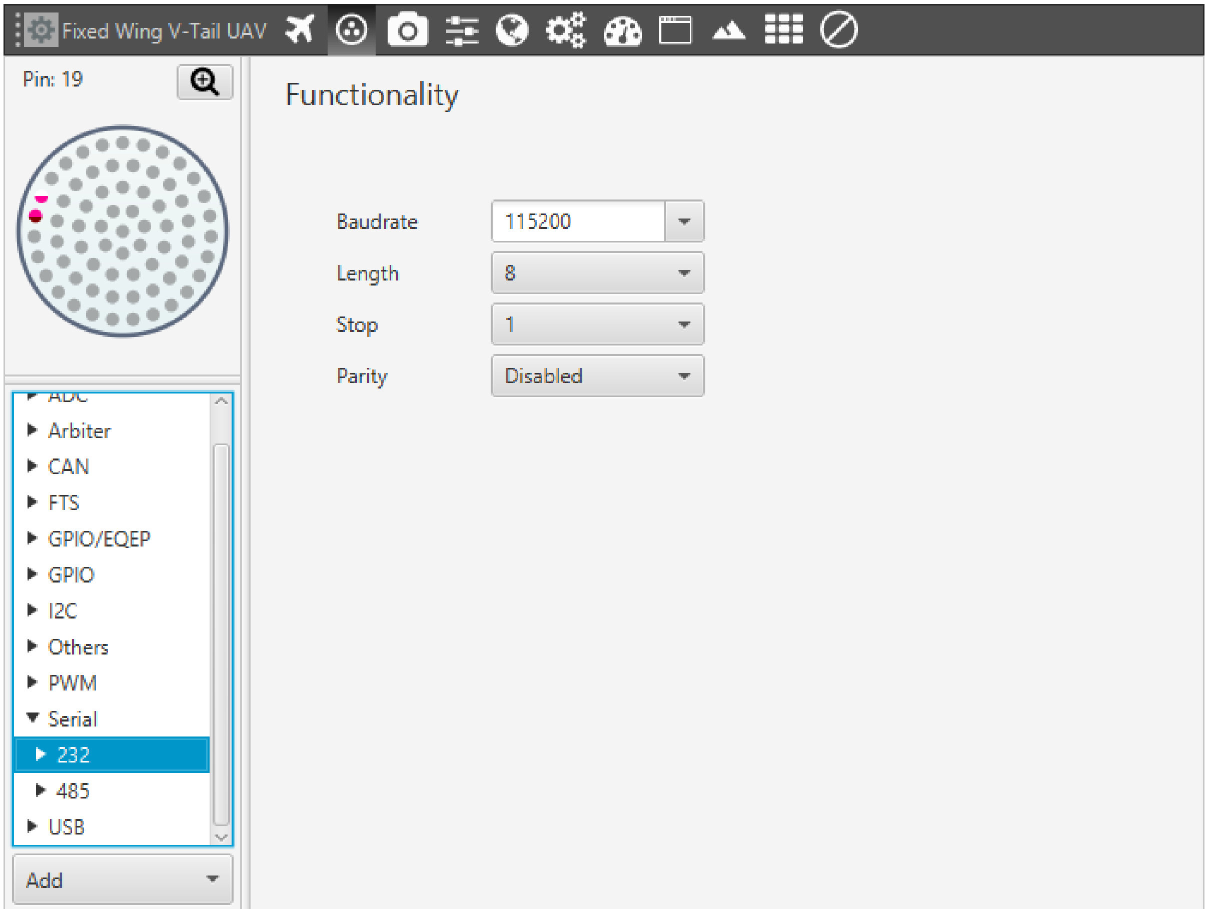


PWM Selection

As can be seen, pins are interchangeable. Once selected the desired pin, click on **Change**.

6.2.2.5 Serial

Two serial interfaces are available with Veronte Autopilots, RS-232 and RS-485, and more can be added by using the Can Expander Board. Each one of the serial interfaces is associated with a set of pins, which are displayed when an interface is selected.



Connections – Serial

The following fields can be configured:

- **Baud rate:** This field specifies how fast data is sent over a serial line.
- **Length:** This field defines the number of data bits in each character.
- **Stop:** Stop bits sent at the end of every character.
- **Parity:** is a method of detecting errors in transmission. When parity is used with a serial port, an extra data bit is sent with each data character, arranged so that the number of 1 bits in each character, including the parity bit, is always odd or always even.

Type	Start	Data	Parity	Stop
Bits	1	4-8	1-2	1-2

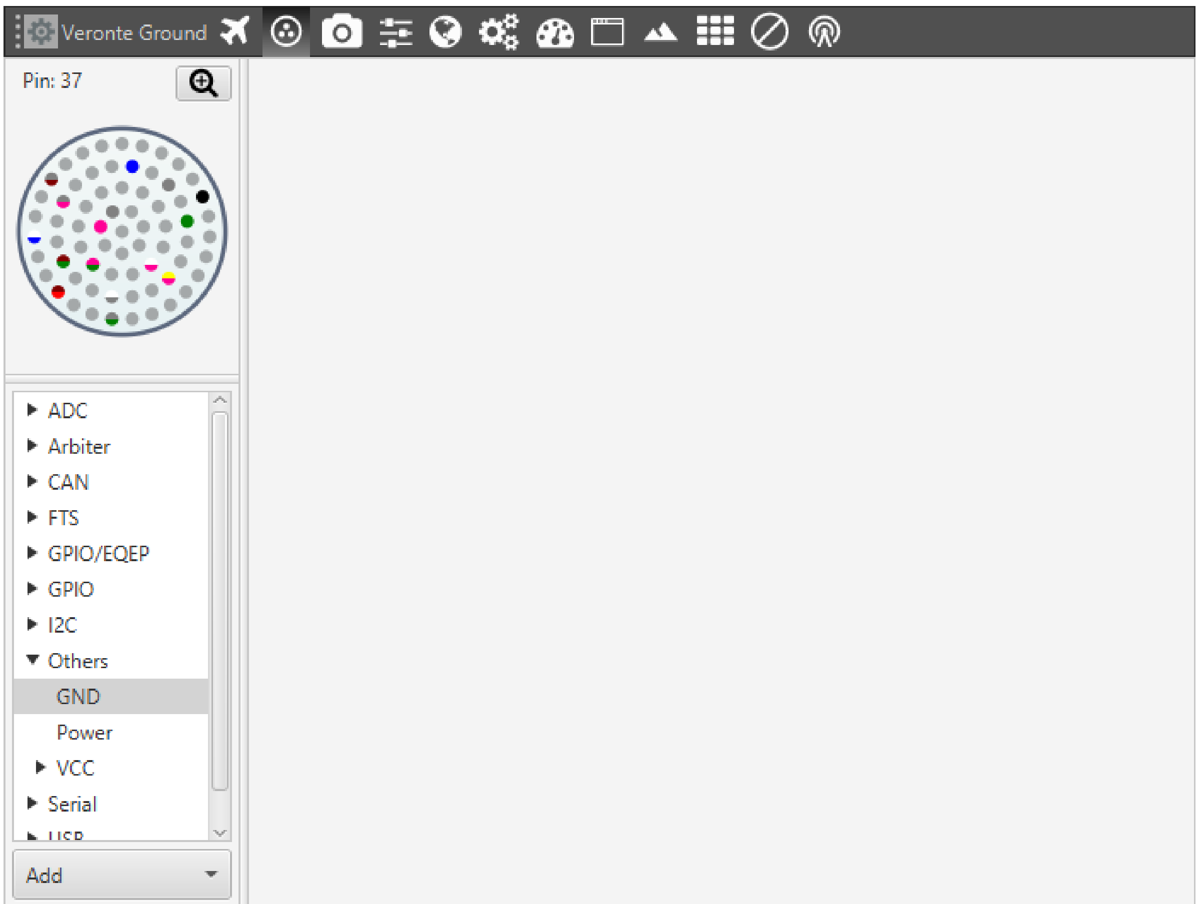
Note: All these settings are already specified for a given device,therefore, Veronte should match with them in order to be able to communicate.

Compatibility table:

<table>
<tr><td>Port name</td><td>RS-232</td><td colspan="2">RS-422 | RS-485</td></tr>
<tr><td>Transfer Type</td><td>Full duplex</td><td colspan="2">Full duplex | Half/Full duplex</td></tr>
<tr><td>Maximum distance</td><td>15 meters at 9600 bps</td><td>1200 meters at 9600 bps</td><td>1200 meters at 9600 bps</td></tr>
<tr><td>Topology</td><td>Point to point</td><td>Point to point</td><td>Multi point</td></tr>
<tr><td>Max number of devices</td><td>1</td><td>1-10 in receive mode</td><td>32</td></tr>
</table>

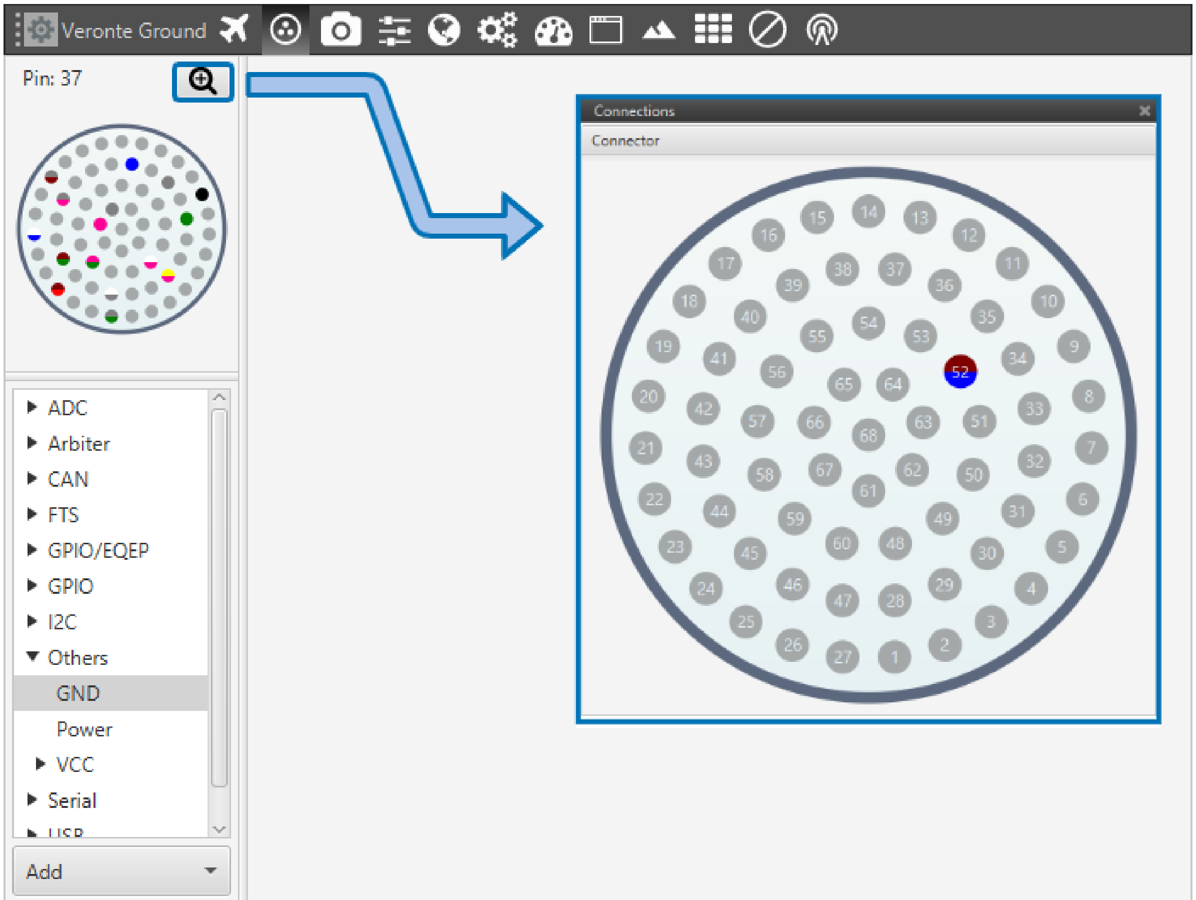
Refer to *examples* for a complete implementation of a serial device communication.

Here the user can configure the Input/Output ports of the autopilot. Veronte Autopilots have 68 pins, which are shown on the Pipe menu **Setup -> Connections** – see the Figure below. When selecting a port its pin is displayed on the panel.



Setup – Connections – GND Pins Displayed

To know about a particular pin's colour scheme, the user shoulg click the button on the top right corner above the 68 pin connector picture – see the Figure below.



Connection 68 Pin Colour Scheme

Warning: The colour code is referred to the single Veronte. For 4x Veronte refer to 4x Hardware Installation.

4x Hardware Installation – Electrical .

Finally, depending on the configurable port selected the user will need to provide different parameters. The following table shows the type of connections available for configuration in Veronte Autopilot.

Field	Description
ADC	Analog-to-Digital Converter.
Arbiter-SuC	Safety micro controller
CAN	Configurable Controller Area Network bus A & B.
GPIO/EQEP	Enhanced Quadrature Encoder Pulse Input.
GPIO	Veronte input/output signals.
I2C	I2C (Inter-Integrated Circuit) bus.
PWM	Pulse Width Modulation configuration.
Serial	Configurable Serial ports RS232 & RS485.

Each connection is associated with a specific pin number. For more details see the section *Hardware Installation – Electrical* .

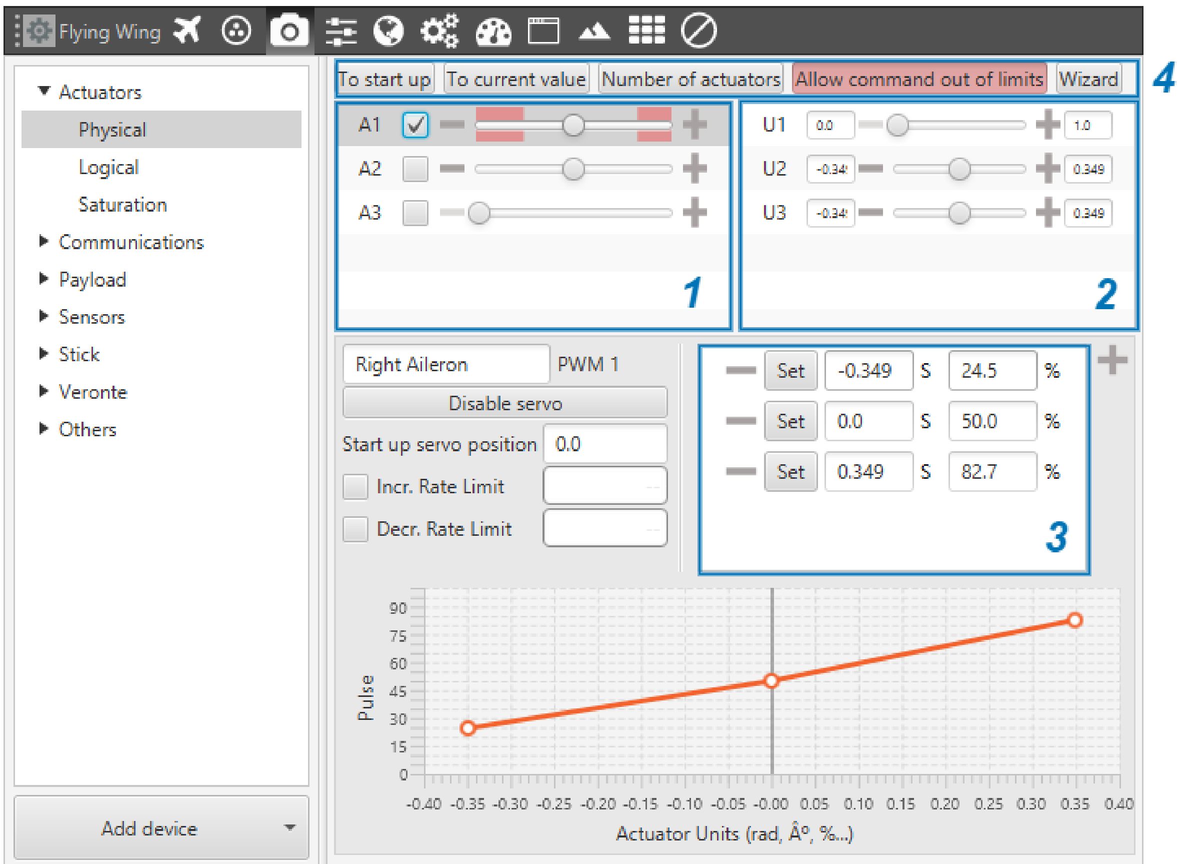
6.2.3 Devices

6.2.3.1 Actuators

6.2.3.1.1 Physical

6.2.3.1.1.1 Calibration Interface

The menu available on **Devices -> Actuators -> Physical** allows the calibration of all connected actuators. On this panel it is possible to set actuator position for each control signal/output, allowing to configure the maximum and minimum values and its **custom** performance. On the Figure below the configuration of a tailless aircraft , i.e. flying wing , is used as an example.



Physical Actuators Menu

The parameters of each one of the servos can be configured in two different ways: directly on the menu shown in the previous Figure, or through the *wizard button*. This last option is the recommended upon first configuration, leaving the other option for adjustments and advanced settings.

The options of the calibration menu are presented next:

1. **Servos (actuators):** this menu contains the servos of the platform. Moving the scrolling bar will change the servo position, but the signal to the system will only be sent if the checkbox next to the servo number (A1, A2…) is marked. The manual movement of the servos can only be done in the "Initial" phase (when there is no phase selected in Veronte Panel).

2. **Control signals:** this menu contains the variables that represent the control signals/outputs U generated by the

system. Considering the flying wing, the controls are thrusting (U1), pitching (U2) and rolling (U3), so there are 3 different controls in total. The mapping from the controls to the servo positions is indicated within the **SU matrix**, which is set in *Devices - Actuators - Logical*. When moving the scrolling bar of one of the control channels, the corresponding movement on the servos will also be represented.

3. **Servo position – PWM:** this option is used to set the mapping from servo position S to PWM signal. For example, a 20º degrees deflection (0.349 rad) of the right elevon corresponds to an 82'7 % pulse to be sent to the corresponding servo. The mapping is expressed through the graph, where the user can introduce as many points as desired.

4. The other tools that appear on the menu are:

   - **To start up:** set initial values of the actuators.
   - **To current value:** set actuators to the current value – requires the physical platform to be connected.
   - **Change size actuators:** change the total number of available actuators.
   - **Allow command out of limits:** allow out-of-limits motion for the selected actuator.
   - **Wizard:** explained below.
   - **Disable/Enable servos:** allows the user to enable or disable a configured actuator.
   - **Start up servo position:** sets the pulse value for the *start up configuration*.
   - **Increasing/Decreasing Rate limit:** sets a rate limit for increasing/decreasing motions of the servo.

The procedure to follow when configuring a servo/actuator is:

- Firstly, set the scrolling bar to its middle position.
- Secondly, click on Enable: this avoids the problem of enabling a servo when it is in an extreme position, which can produce a malfunction.
- Set the desired maximum S_{max} and minimum S_{min} values of the actuator S (a restrictive approximate value works).

Note: S_{max} and S_{min} can be chosen to match the actual physical value they represent, e.g. the angle of delfection of an aileron. But they can also be chosen not to match a physical variable, and it won't affect the autopilots behaviour as these S values are the representation of PWM signals. And the latter are the ones that correspond to an actual control surface/device movement.

Take a motor, for instance. The minimum PWM signal corresponds to a 0 rpm state, but a maximum PWM signal corresponds to finite number of rpms. In this case it's easier to assume $S_{max} = 1$ and $S_{min} = 0$. The same could be done for the ailerons. If they are deflected the same amount in both directions then $S_{max} = 1$ and $S_{min} = -1$.

- Finally, connect the physical actuator and adjust its minimum and maximum values.

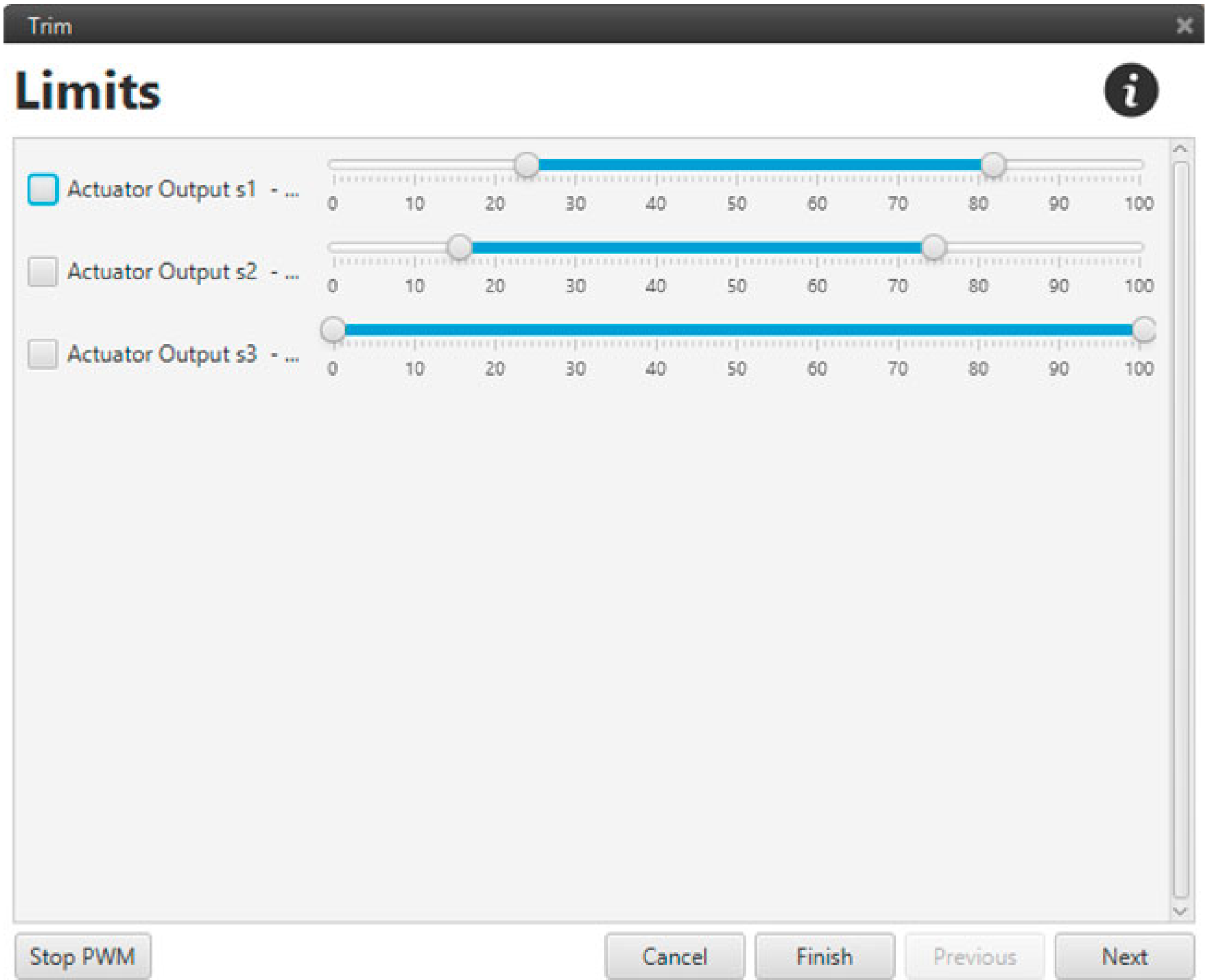
6.2.3.1.1.2 Configuration Wizard

Configuration wizard is recommended on first system configuration. It guides the user for configuring actuator limits and performance. **This calibration should be performed in order to define the relationship between the autopilot output and real servo position.**

In order to set servo positions, it is possible to enable servo movement from the user interface. Check the box on the left of each servo in order to enable servo movement, once enabled, servos will move according to the bar. On the bottom left part of the windows, there is the Stop PWM, with which all checkboxes are disabled.

If the user wants to finish the wizard and keep the results configured up to that point, just click on Finish on the bottom right part.

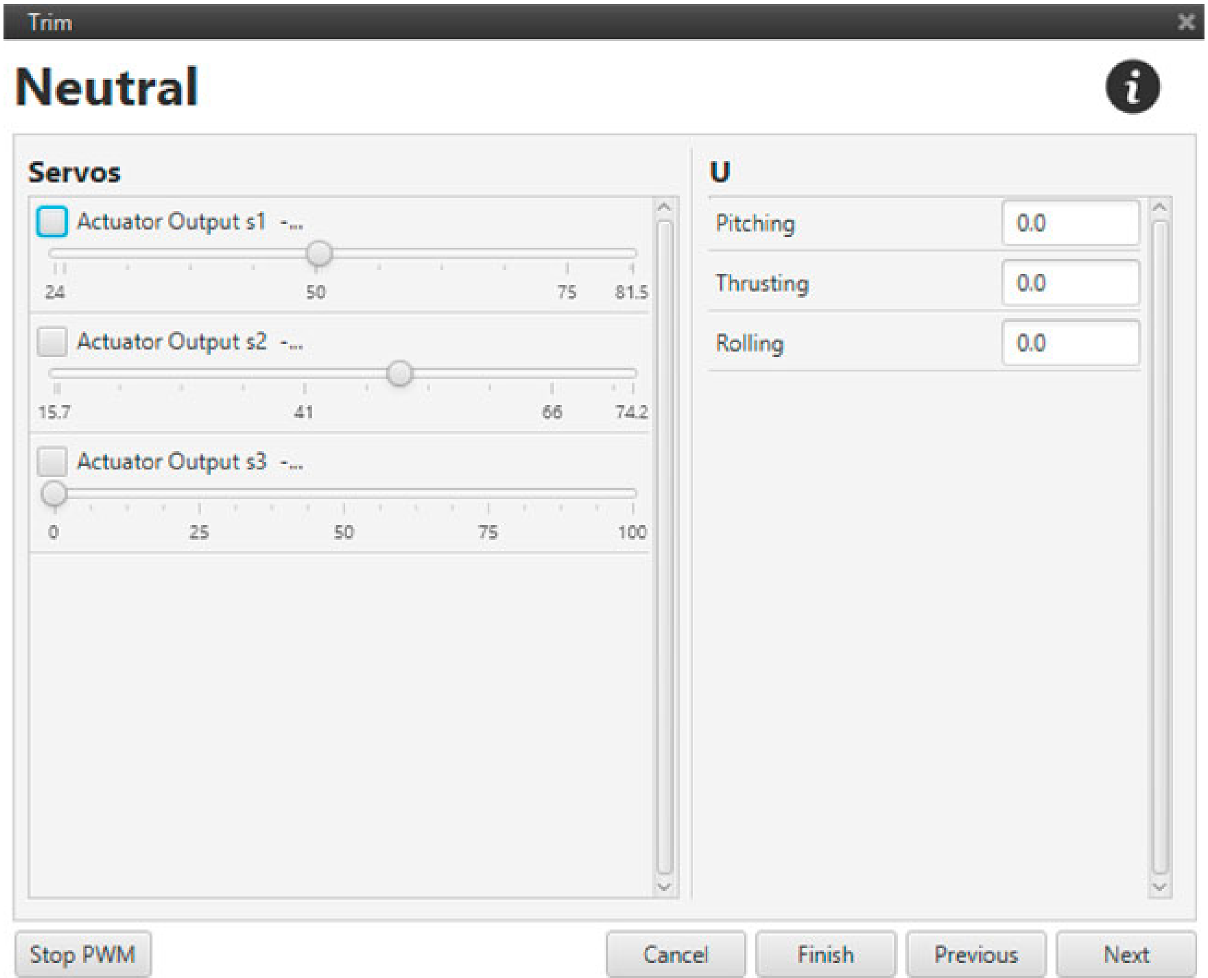
- **Limits:** Set maximum and minimum pulse value for the mechanical limits on the actuator.



Trim Wizard – Limits

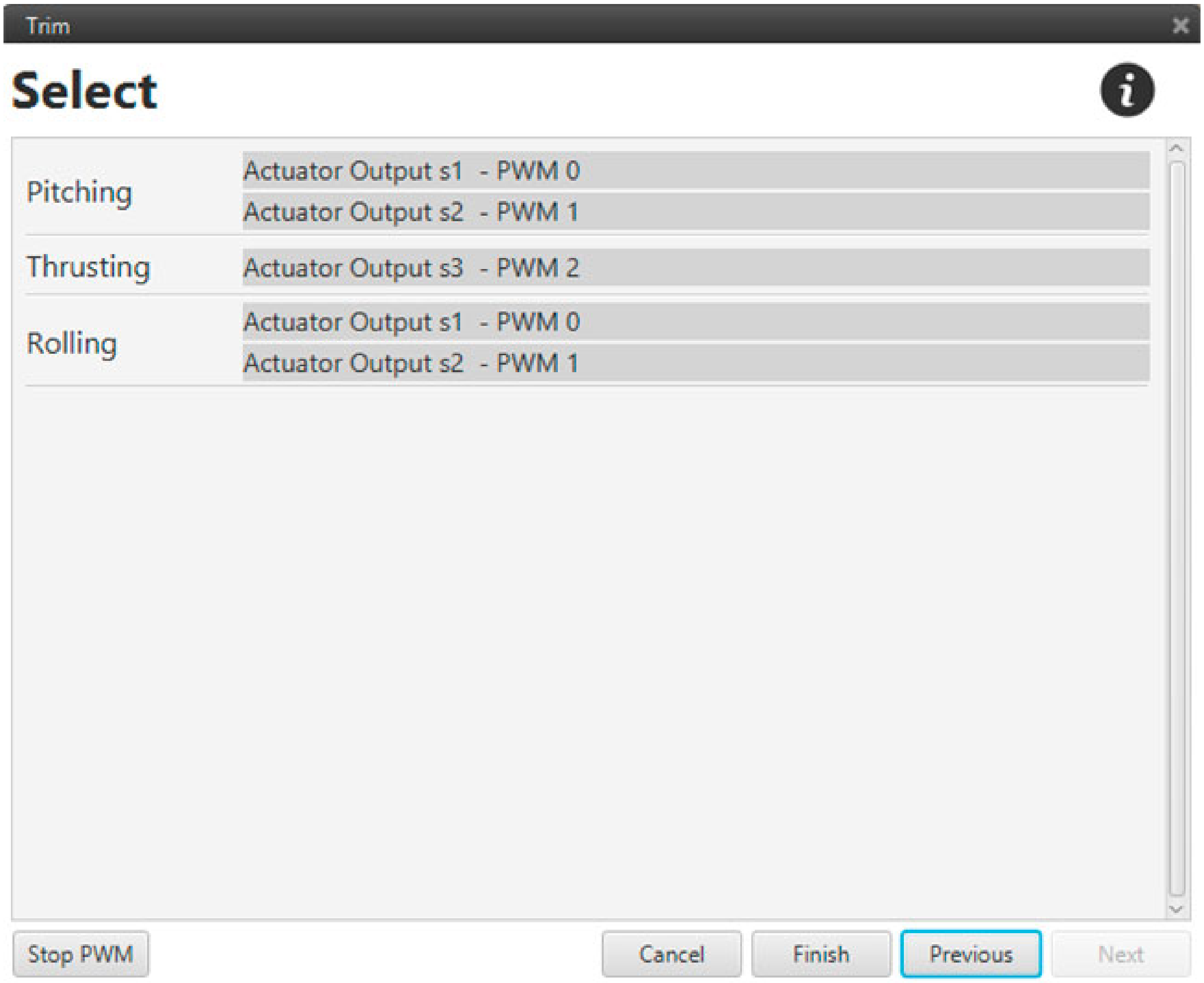
The autopilot will never command the actuator outside these limits, preventing mechanical damages in the aircraft.

Neutral: Set neutral position for actuators and control channels. This position is set as a reference in the system and will remain as the default position on system startup. On the right, there is the column for U, where the neutral value for that particular control output is chosen.



Trim Wizard – Trim Wizard – Neutral

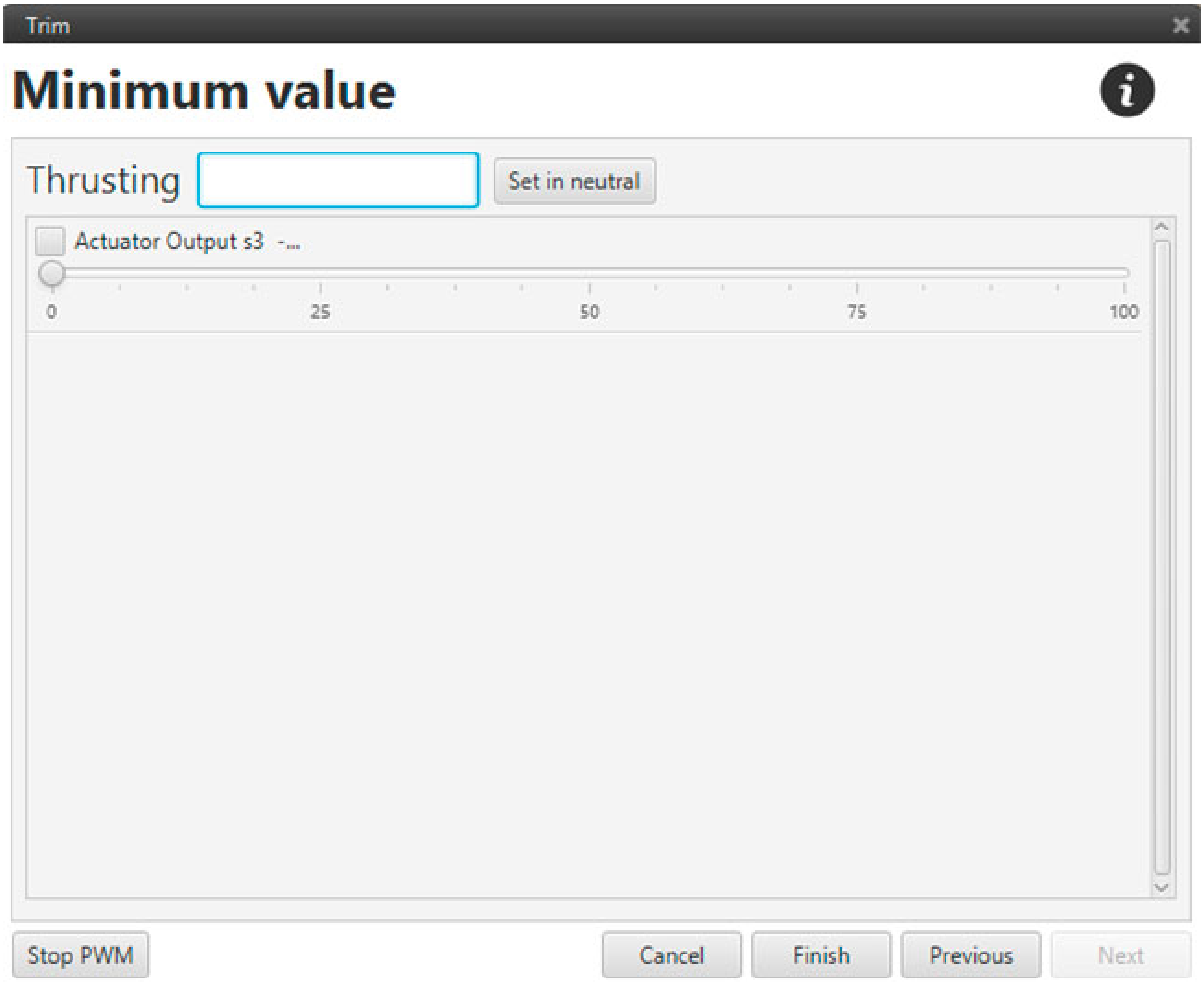
- **Select:** Select a servo to trim. The servos are sorted depending on which is affected by each control input.



Trim Wizard – Trim Wizard – Select Output

Once a servo is trimmed for a defined control channel, it will not be displayed for trim in other control channels.

- **Value:** Enter the maximum and minimum actuator position for the control variable. These values correspond to the components of S, for example, a maximum deflection of 20 degrees will correspond to a 75 % pulse and a minimum of -20 a 25%. These limits don't have to be the same as the mechanical limits of the actuator establish in the first screen of the wizard.



Trim Wizard – Trim Wizard – Min/Max Values

When the process is finished, a summary display appears, showing the user the graphs relating all the information provided, as the on the bottom part of the main display.

6.2.3.1.2 Logical

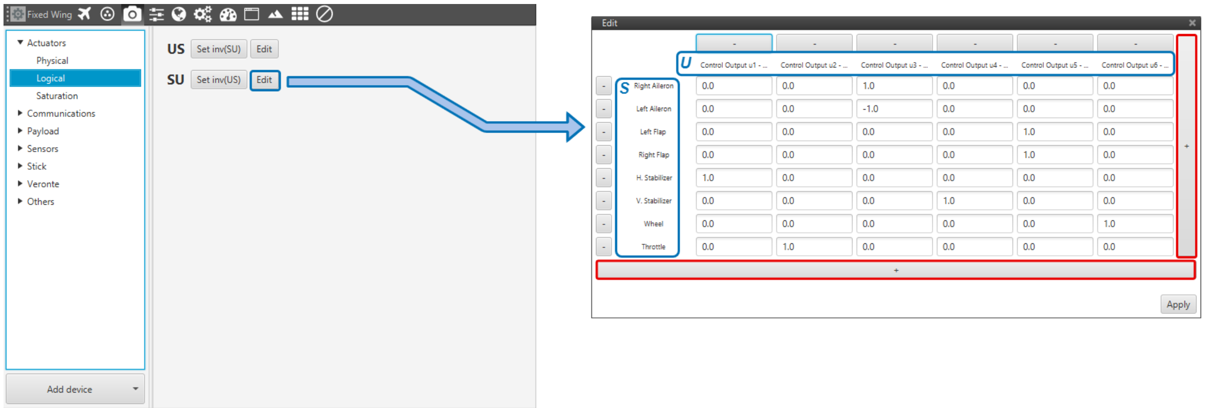
6.2.3.1.2.1 SU & US Matrices

SU and US are two matrices (inverse of one another, respectively) which contain the relationship between actuator outputs S and control outputs U, i.e. the influence of each control channel on each actuator output. The option of having a configurable SU matrix allows Veronte autopilots to control any type of vehicle, independently of how its control surfaces/devices are set and adjusted.

U is a vector which contains the control outputs of the platform, e.g. **pitch**, **roll**, **yaw**, **throttle**, etc. The values of U do not represent a physical variable. They are instead fictitious variables which are used in the control algorithm. What is actually applied to the system are the actuators movements, i.e. the PWM signals sent to the servos, which are mapped in the S vector – see *Devices - Actuators - Physical* for more information. The relation between S and U is essential for the right attitude control of the platform.

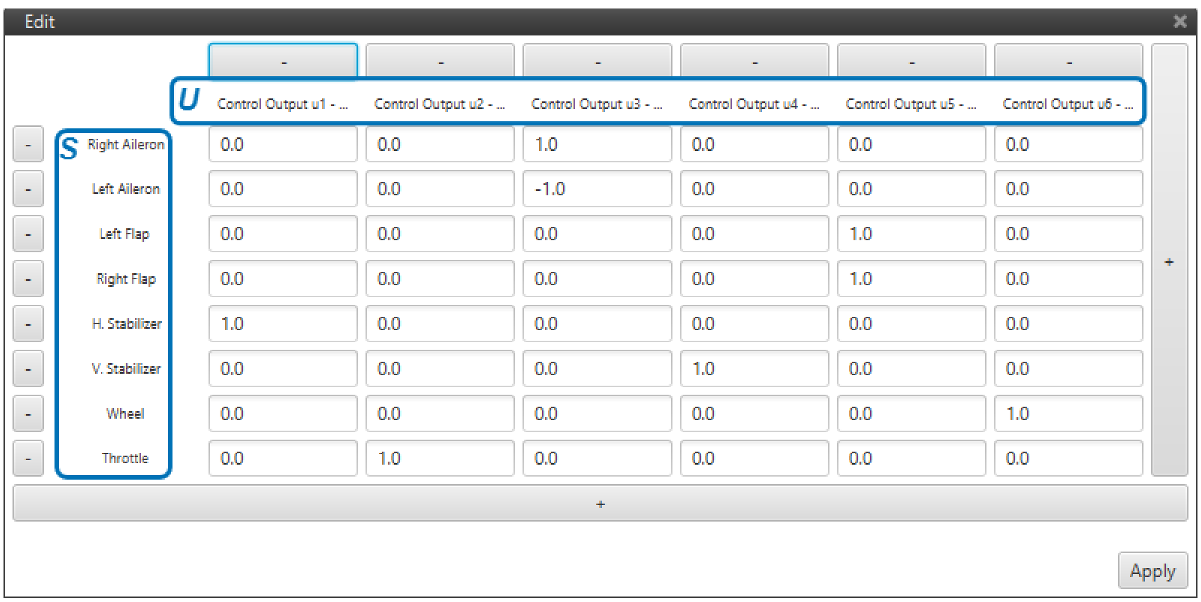
In order to define SU, the user needs to go to **Devices -> Actuators -> Logical** and click on *Edit* – see Figure below. A pop-up window will open containing the matrix. Control outputs U are placed on the columns and actutator outputs

S on the rows. Clicking on the '+' sign allows the user to add a new U or S, adding a new column or row will appear, respectively.



SU Matrix Menu

Definition of US is also avaialble, but it is recommended to go with the first as it is more intuitive to define. Let's comsider the SU of a fixed-wing aircraft – see the Figure below. On the S vector there are 8 entries, i.e. 8 actuators; and on the U vector there are 6 control outputs.



SU Matrix for a Fixed-Wing aircraft

As we see in the Figure above, the user defines which actuators influence which control channels. For instance, only the actuator **throttle**, e.g the engine – influences the control output U_2: **thrusting**; and only the **horizontal stabilizer** influences U_1: **pitching**.

Warning: Regarding the selection of the parameters of SU matrix:

The order of magnitude of the parameters should be respected at least for every row, i.e. every control channel, as long as there are no coupled control channels U.

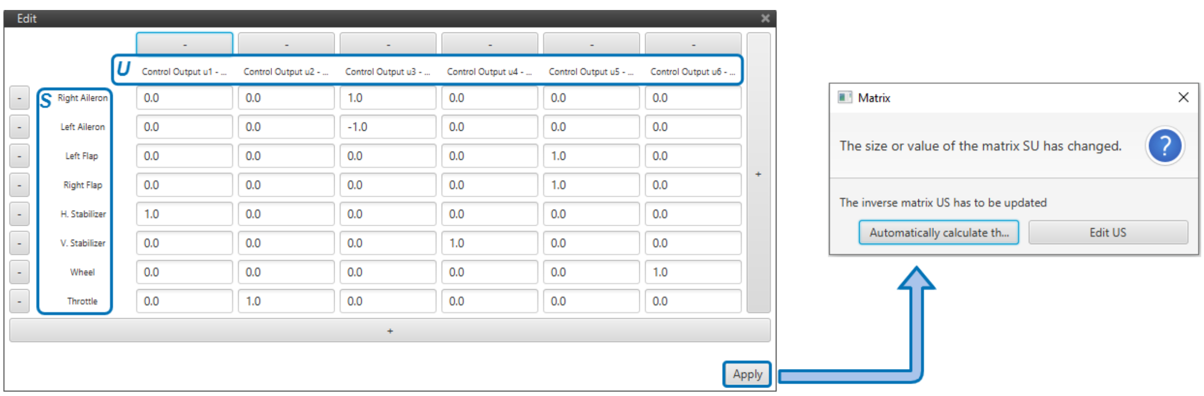
Note: Good practice recommendations:

- Unitary values are recommended. Doing so, U will be equal to S. And if S has been defined according to a physical value – e.g. deflected angle, then control outputs can be easier to understand.
- The order of magnitude and the value of the SU parameters will not influence control algorithm calculations. But it will affect the control parameters, i.e. the control gains.
- It is recommended to keep the same order of magnitude for the whole matrix. That will allow an easier set up of a scaled version of the platform. Keeping the same SU and knowing the scaling factor, then the new control gains should be the old ones multiplied by that scaling factor. This practice can also be useful for transition to similar platforms.
- The SU vector should be defined accordingly in order to follow the sign convention for aerial navigation: a positive roll lowers the right wing, a positive pitch moves the nose up and a positive yaw moves the nose the right.

Important: Regarding SU parameters of opposite sign:

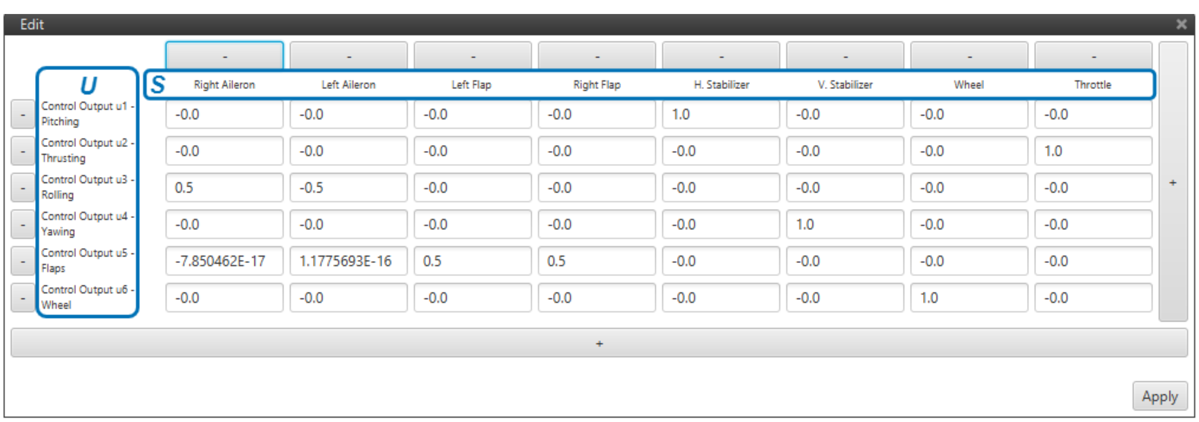
In the above Figure one can see that for control output U_3 : **rolling**, the right aileron indicates 1 and the left aileron -1. In other words, one moves opposite to the other. But this is a mere convention. Everything depends on the ailerons PWM to servo movement curves – defined in *Devices -> Actuators -> Physical*.

Coming back to the matrix SU menu, the user should click on *Apply* once it has finished. A pop-up window will appear (see Figure below) warning the user about the US matrix needing to be updated: it can be automatically updated from the SU, or introduced manually. It is advised to go with the first option.



SU to ***US***

The US matrix of the fixed-wing aircraft is shown in the Figure below:

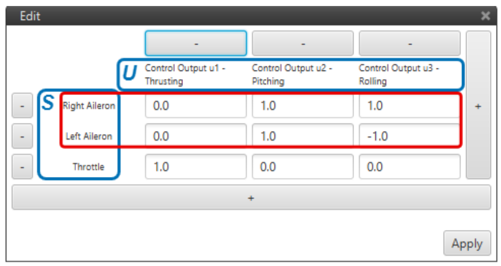


$\boldsymbol{US}$ Matrix for a Fixed-Wing aircraft

Note that the fixed-wing example shown above has only one S only. Now, some more practical examples of real configurations – with more complex SU matrices – will be presented in order to complement the above explanation.

6.2.3.1.2.2 SU of a Flying Wing

Flying wings are tailless aircrafts, i.e. their only control devices are ailerons and a motor. So both the longitudinal (**pitching**) and lateral (**rolling**) control of the platform have to be controlled only with the ailerons – known as elevons in this kind of configurations. Therefore, two actuator channels S (left and right ailerons) will have to incluence two control outputs U. When working like elevators, the elevons move together; when working like ailerons, one moves up as the other one moves down – this dual behaviour ir highlited in red in the Figure below.

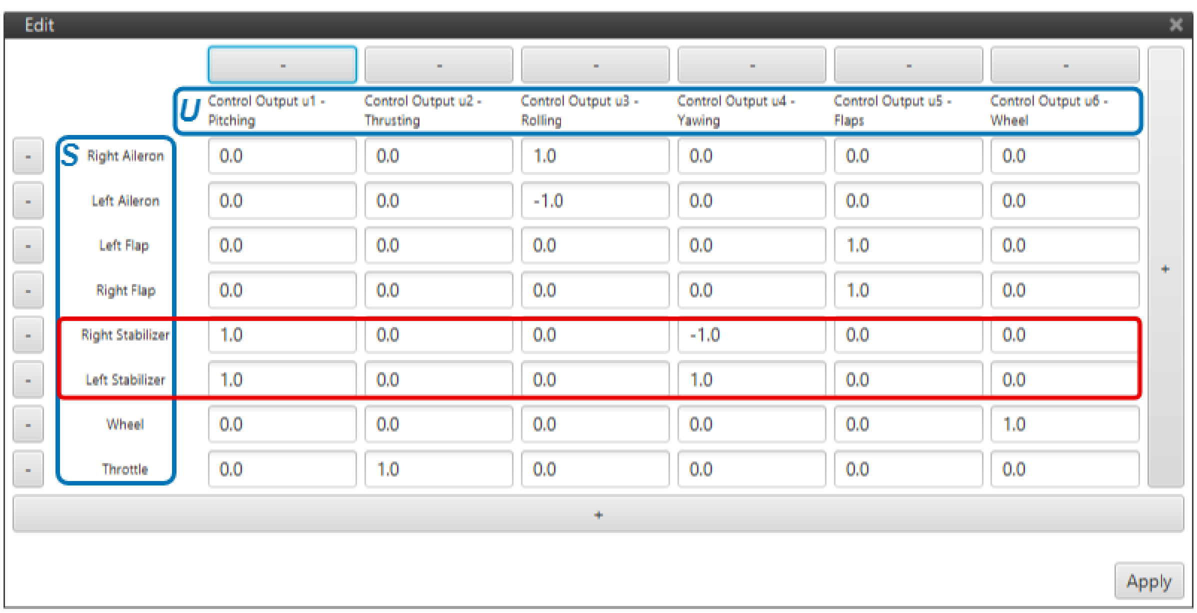


SU Matrix of a Flying Wing

6.2.3.1.2.3 SU of a V-Tail Aircraft

The V-tail aircraft presented next has the same platform structure as the one shown above except for one difference: the tail of the aircraft is at an angle (v-shaped). That implies that **pitching** and **yawing** are controlled by two control surfaces, while for a regular congfiguration they are controlled by one separate control surface each. These control surfaces are know as ruddervators. When working like elevators, the ruddervators move together; when working like rudders, one moves up as the other one moves down – this dual behaviour ir highlited in red in the Figure below.

Regarding the rest of the actuators S, each of them only influence one single control output U– see the SU matrix below.



SU Matrix of a V-Tail Aircraft

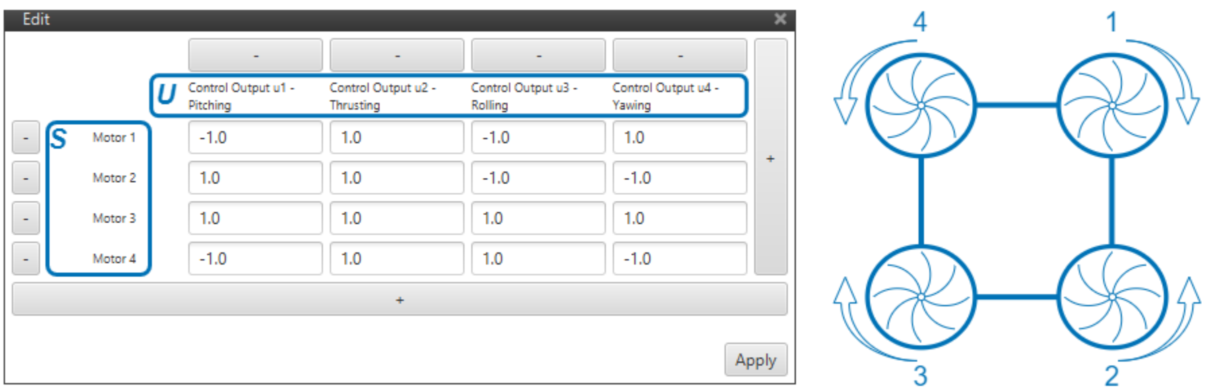
6.2.3.1.2.4 SU of a Quadcopter

When it comes to a multicopter the SU matrix definition is not straigthforward. Here we consider a quadcopter because it is the simplest of cases. Therefore, as its name suggests, it has 4 motors. The convention of the motor order in a quadcopter is the following: motor 1 is placed on the front right of the quadcopter, and the rest of them are placed following a clock-wise oder – see the Figure below.

When defining the SU matrix of a quadcopter with symmetrical geometry:

- Control output U: **thrusting** will require all 4 motors to increise their rpms the same amount.
- Control output U: **pitching** will require for motors 1 and 4 to increase their rpms one particular amount, and motors 2 and 3 to decrease their rpms that same amount.
- Control output U: **rolling** will require for motors 1 and 2 to increase their rpms one particular amount, and motors 3 and 4 to decrease their rpms that same amount.
- Control output U: **yawing** will require for motors 1 and 3 to increase their rpms one particular amount, and motors 2 and 4 to decrease their rpms that same amount.

In multicopters, specially for symmetrical geomtries, it is interesting to introduce the same influence values for the **pitching** and **rolling** control channels. That is so because when tunning the control gains (control parameters) they are usually very similar in both control channels.

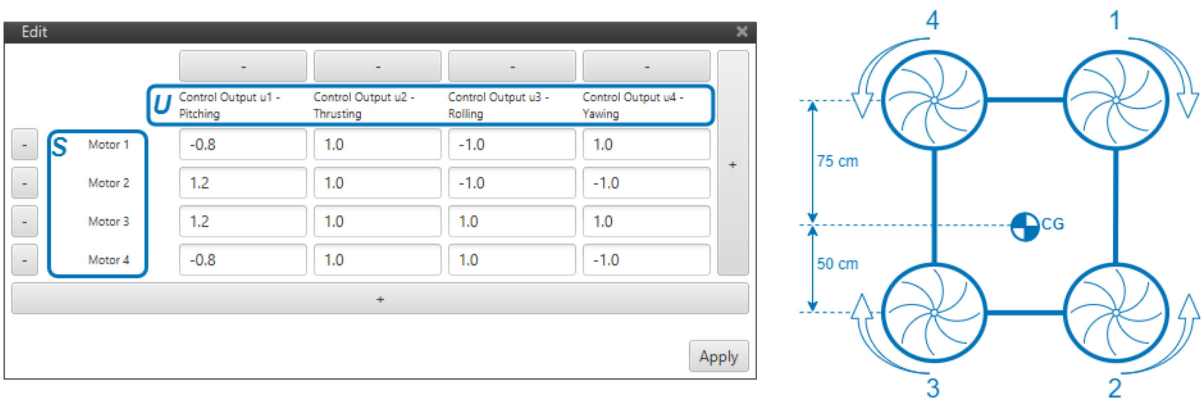


SU Matrix of a Quadcopter

If the considered quadcopter wouldn't have symmetrical geometry, the SU matrix needs to account for the lack of symmetry. In the Figure below, one can see that motors 1 and 4 are further ahead of the center of gravity (CG) of the platform than motors 2 and 3.

For a hover maneouvre, the desired **pitching** moment is 0. Assuming the latter, i.e. the sum of all existing moments equal to 0, one finds the following relation $T_{back} = 1.5 \cdot T_{front}$, where T represents the force generated by the motors.

Therefore, one needs to define an influence value for the motors in the front (1 and 4) and then, multiply that same value by 1.5 for the motors in the back (2 and 3). One could choose the values 1 and 1.5, but in order for the **pitching** and **rolliing** values to be similar, 0.8 and 1.2 are a better option – and they also satisfy the above relation.



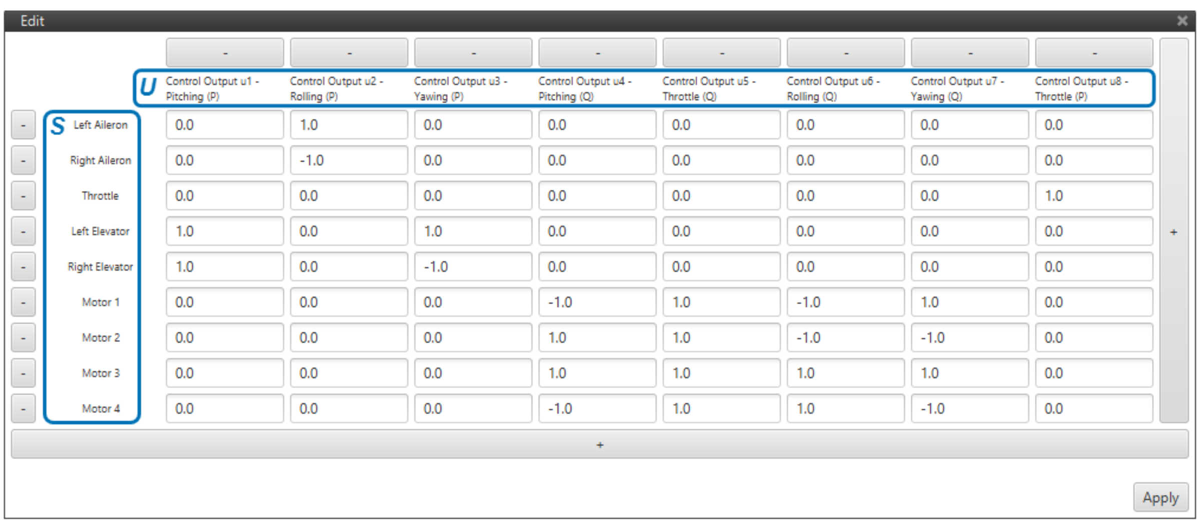
	Control Output u1 - Pitching	Control Output u2 - Thrusting	Control Output u3 - Rolling	Control Output u4 - Yawing
Motor 1	-0.8	1.0	-1.0	1.0
Motor 2	1.2	1.0	-1.0	-1.0
Motor 3	1.2	1.0	1.0	1.0
Motor 4	-0.8	1.0	1.0	-1.0

SU Matrix of a Quadcopter (non-symmetrical geometry)

6.2.3.1.2.5 SU of a VTOL Aircraft

Finally, a vertical take-off and landing (VTOL) aircraft configuration is covered. Main features of the aircraft are: V-shaped tail, a wing with ailerons only, one motor for plane mode, and 4 motors for quadcopter mode (VTOL). The recommended strategy for this kind of platform is to create 4 control channels U for the plane mode and 4 control channels U for the quadcopter modes.

The SU matrix of the VTOL aircraft looks like:

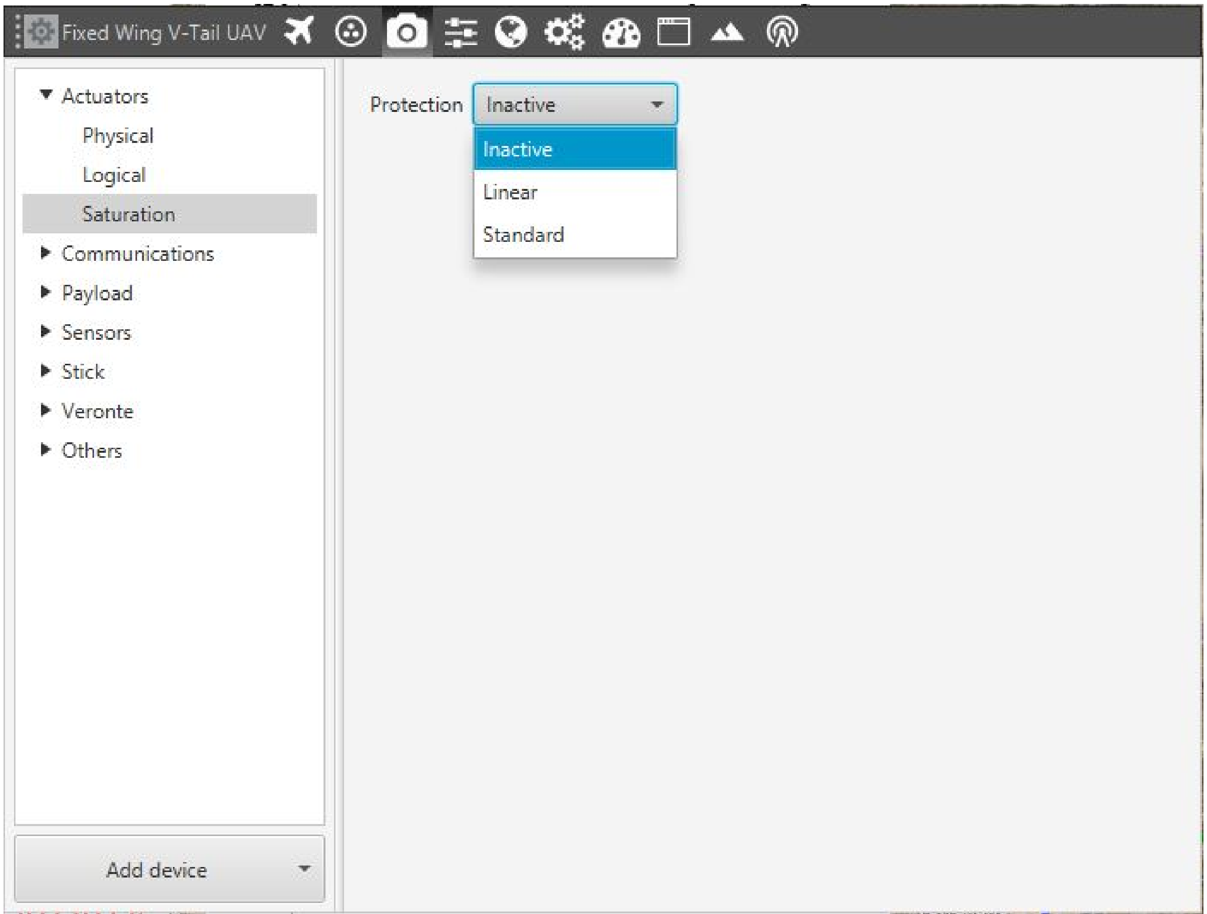


	Control Output u1 - Pitching (P)	Control Output u2 - Rolling (P)	Control Output u3 - Yawing (P)	Control Output u4 - Pitching (Q)	Control Output u5 - Throttle (Q)	Control Output u6 - Rolling (Q)	Control Output u7 - Yawing (Q)	Control Output u8 - Throttle (P)
Left Aileron	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Right Aileron	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
Throttle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Left Elevator	1.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Right Elevator	1.0	0.0	-1.0	0.0	0.0	0.0	0.0	0.0
Motor 1	0.0	0.0	0.0	-1.0	1.0	-1.0	1.0	0.0
Motor 2	0.0	0.0	0.0	1.0	1.0	-1.0	-1.0	0.0
Motor 3	0.0	0.0	0.0	1.0	1.0	1.0	1.0	0.0
Motor 4	0.0	0.0	0.0	-1.0	1.0	1.0	-1.0	0.0

SU Matrix of a VTOL Aircraft

6.2.3.1.3 Saturation

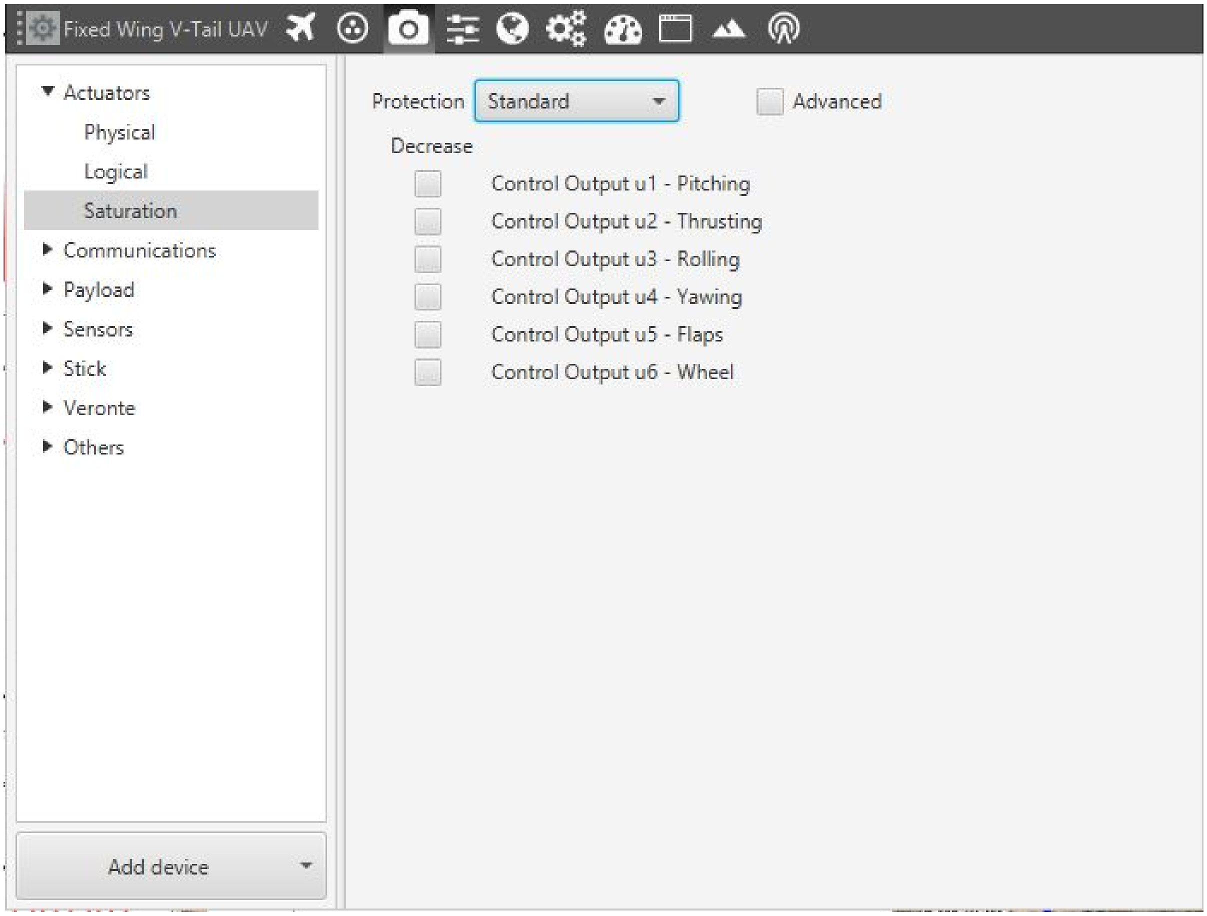
In this menu, the user can configure the behaviour of the platform when one or more of its actuators is/are in saturation state. The three options available are **Inactive** (the system does not respond to saturation), **Linear** (system affects all the actuators on the same way) and **Standard**, which can be found below.



Saturation Option for Actuators

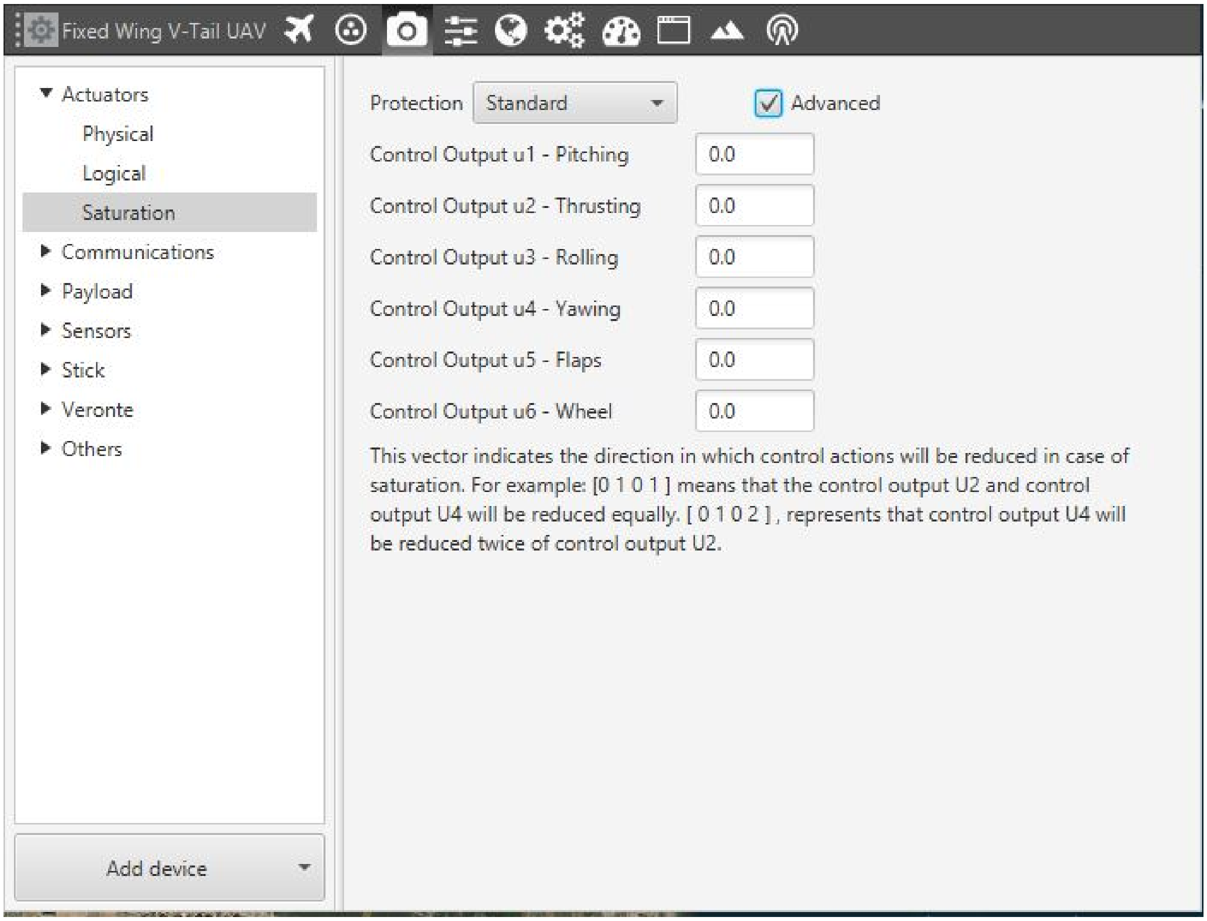
6.2.3.1.3.1 Standard Saturation

The system allows the user to choose which of the Controls is going to be affected if saturation is reached at any actuator. It can be chosen from 1 to all of them (which will be linear action).



Standard Saturation Menu

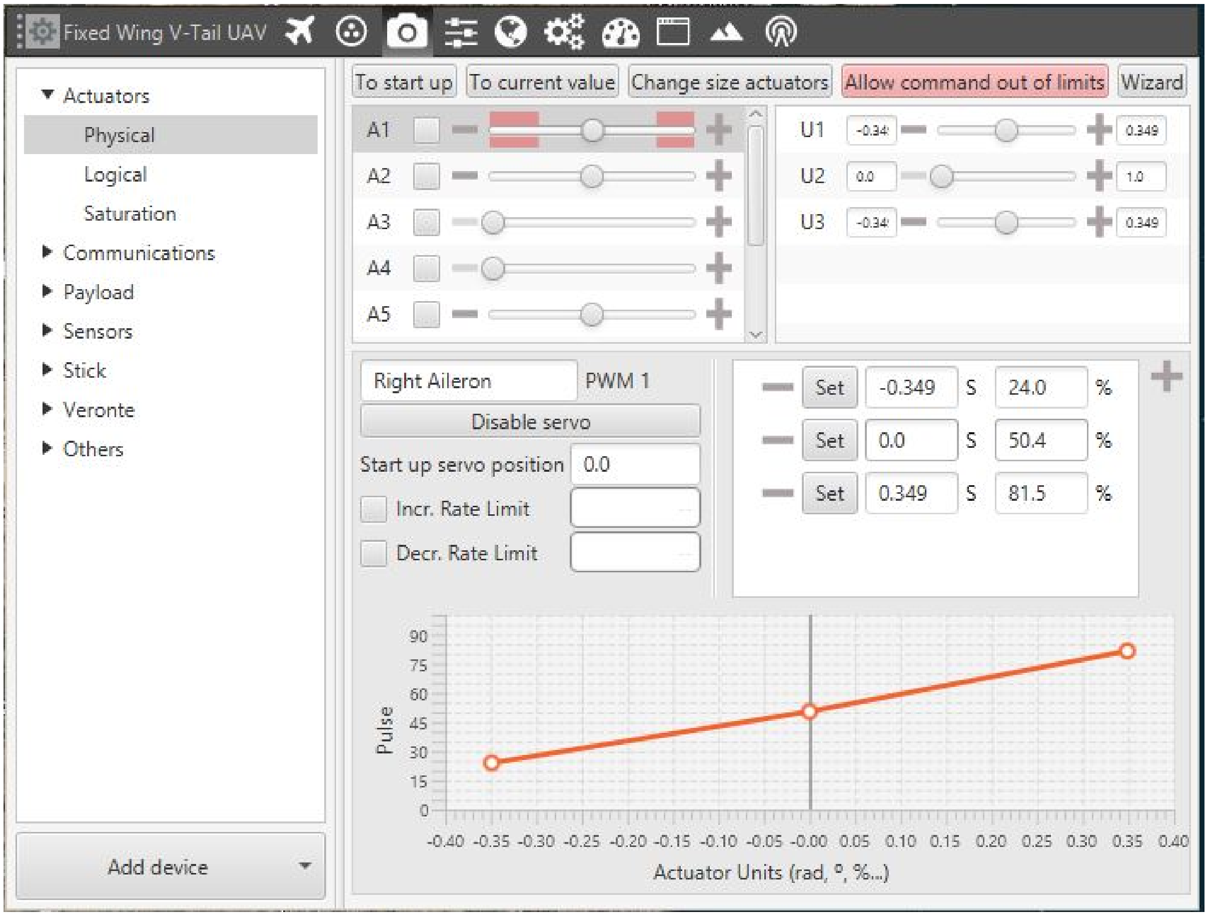
If clicking in the advance checkbox, a vector including all control outputs is generated, allowing for proportional control over the system when saturation happens. This tool is set for the user to have wider control over this feature if needed.



Advance Standard Saturation Mode

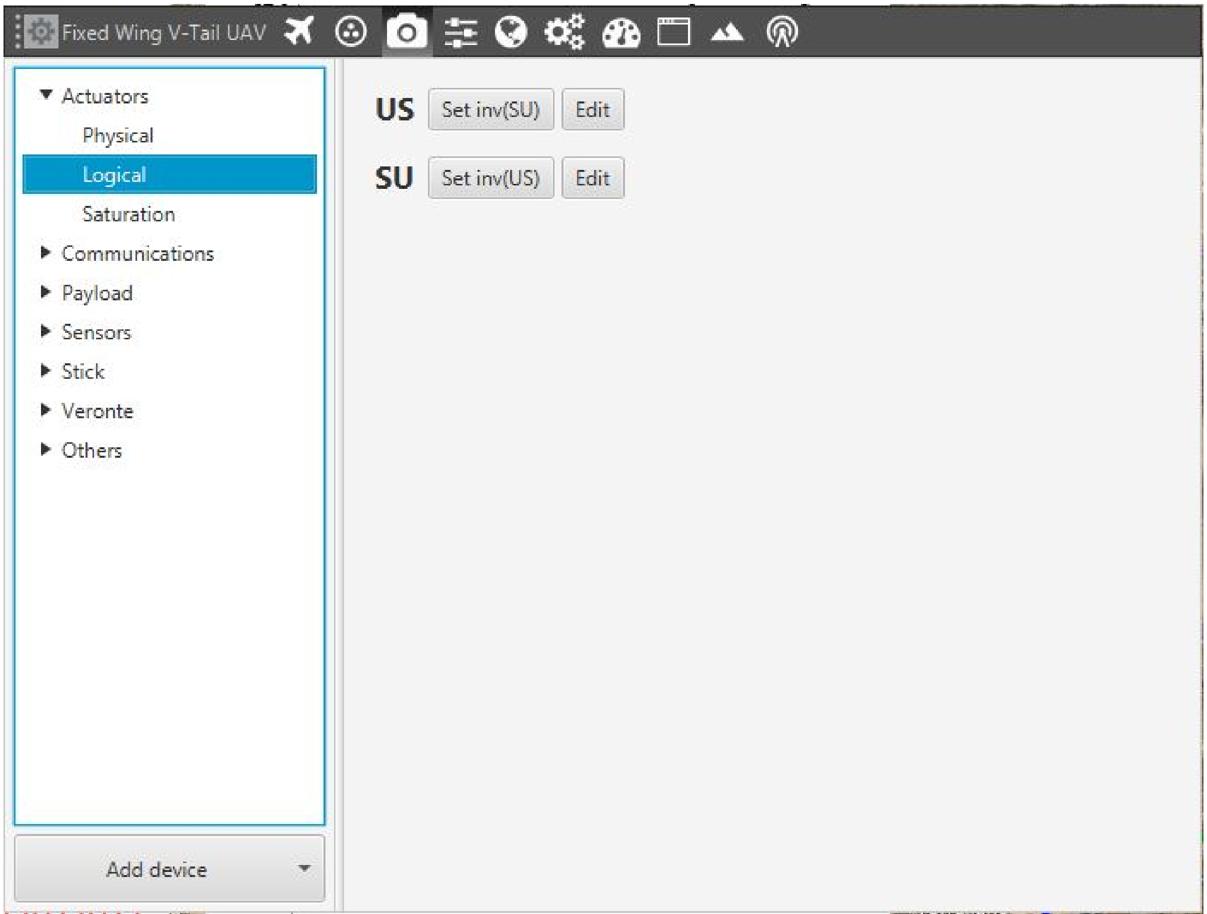
Inside this menu, the user can access the configuration for the servos.

The first menu allows for actuators physical configuration, establishing maximum and minimum values, enabling or disabling actuators, giving them a corresponding angle associated to a pulse or using a Wizard display to do everything at once.



Physical Actuators Configuration

The second menu is the logical, where the US/SU matrices are configured. This defines the influence of each control channel on each control actuator.



Logical Actuators Configuration

The last configuration menu available is the Saturation one, where the user can choose between Saturation kind of protection and some more internal parameters.

imgs/veronte_configuration/Actuators-Saturation.png

Saturation Actuators Configuration

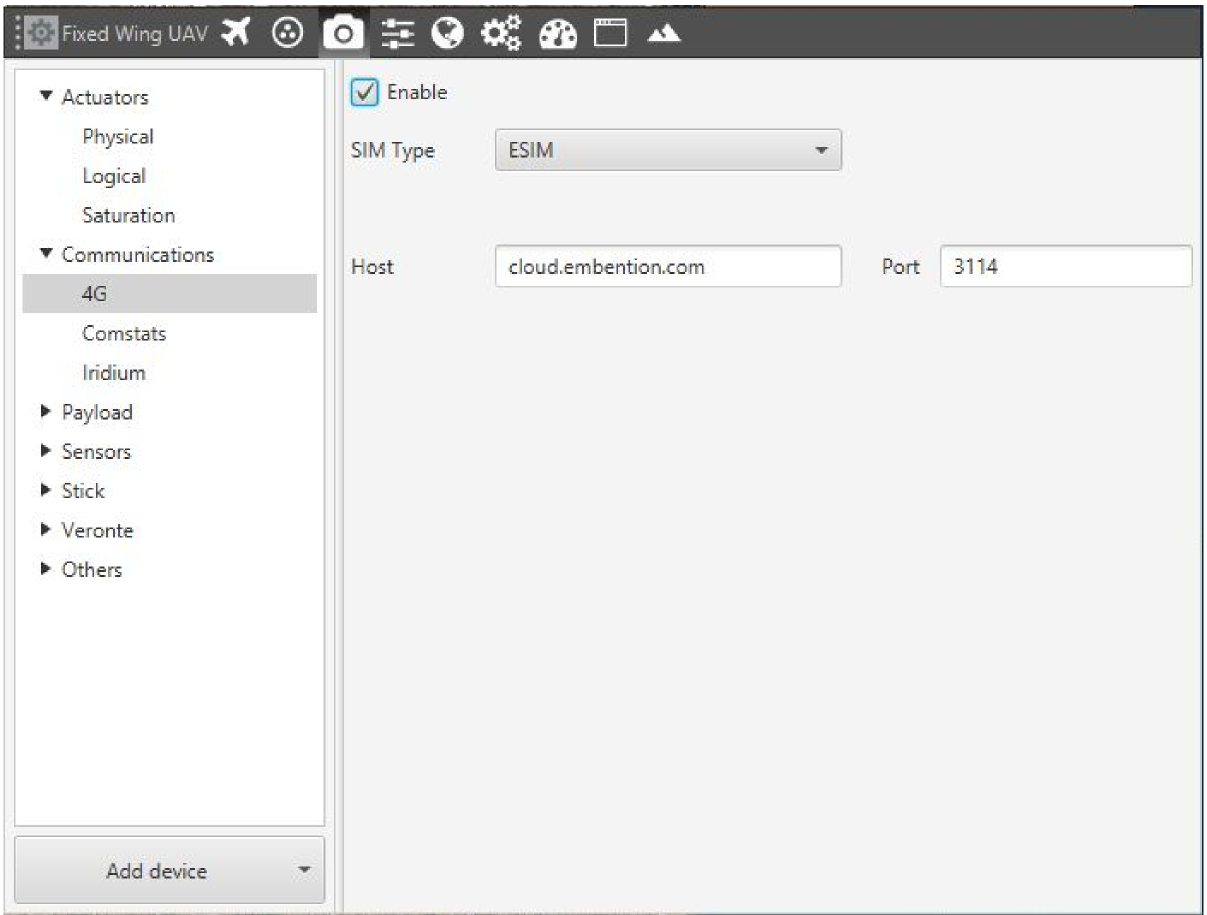
6.2.3.2 Communications

Communications menu covers the configuration for the **4G** service, **Comstats** and **Iridium**, in three different menus which are showed and explained below:

6.2.3.2.1 4G

6.2.3.2.1.1 ESIM

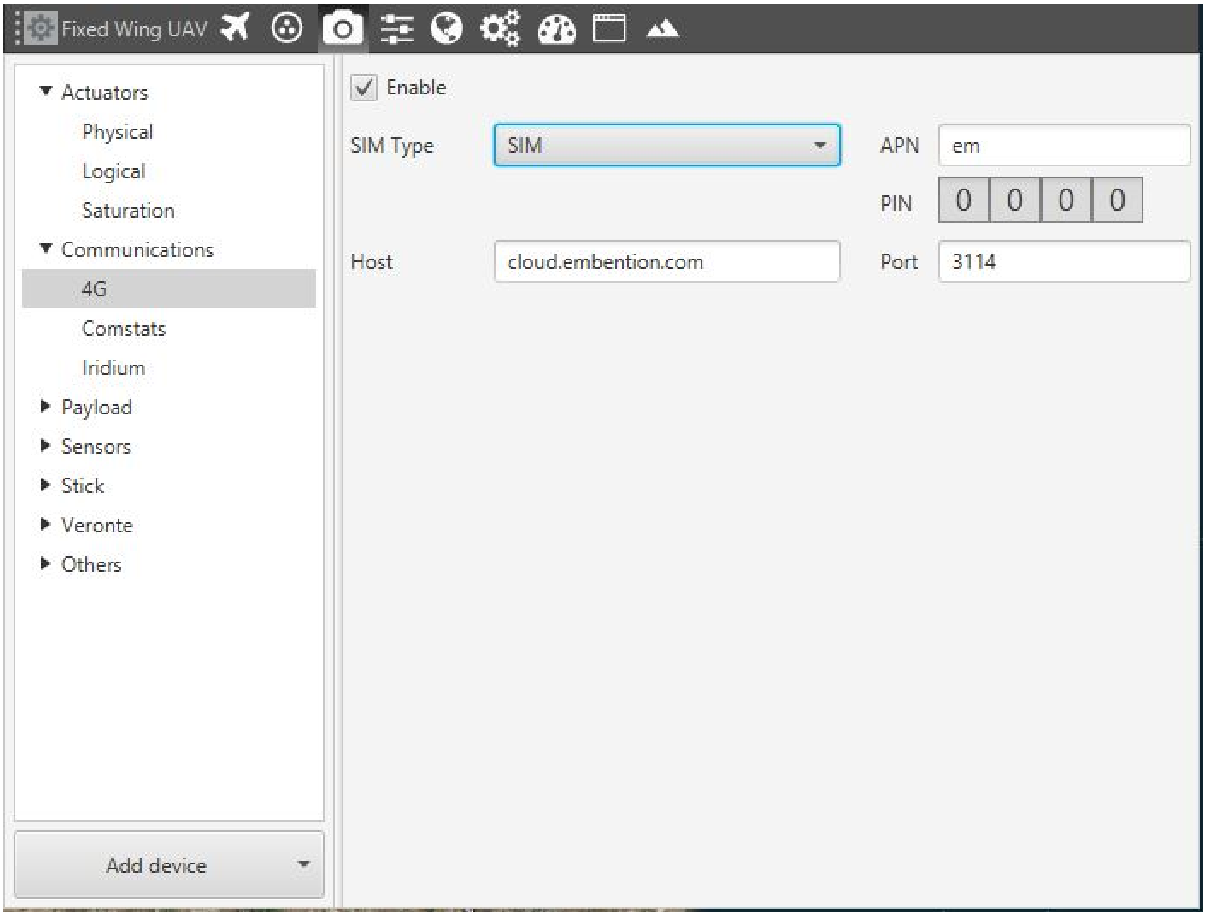
If the user has decided to use the card from Embention (directly added to Veronte autopilot), this option brings already configured the data needed to connect to the cloud. **Host** and **Port** are to be configured as in the following picture.



ESIM Configuration for 4G

6.2.3.2.1.2 SIM

If the SIM embedded in Veronte is the customer's card (contact Embention for using your own SIM Card), the menu differs from the ESIM one, asking for the **PIN** of the card (4 digits code) and its **APN** (Access Point Name), data provided by the company issuing the card. Host and Port can also be changed if the cloud system differs from Embention Cloud, but the communication protocol does not change.

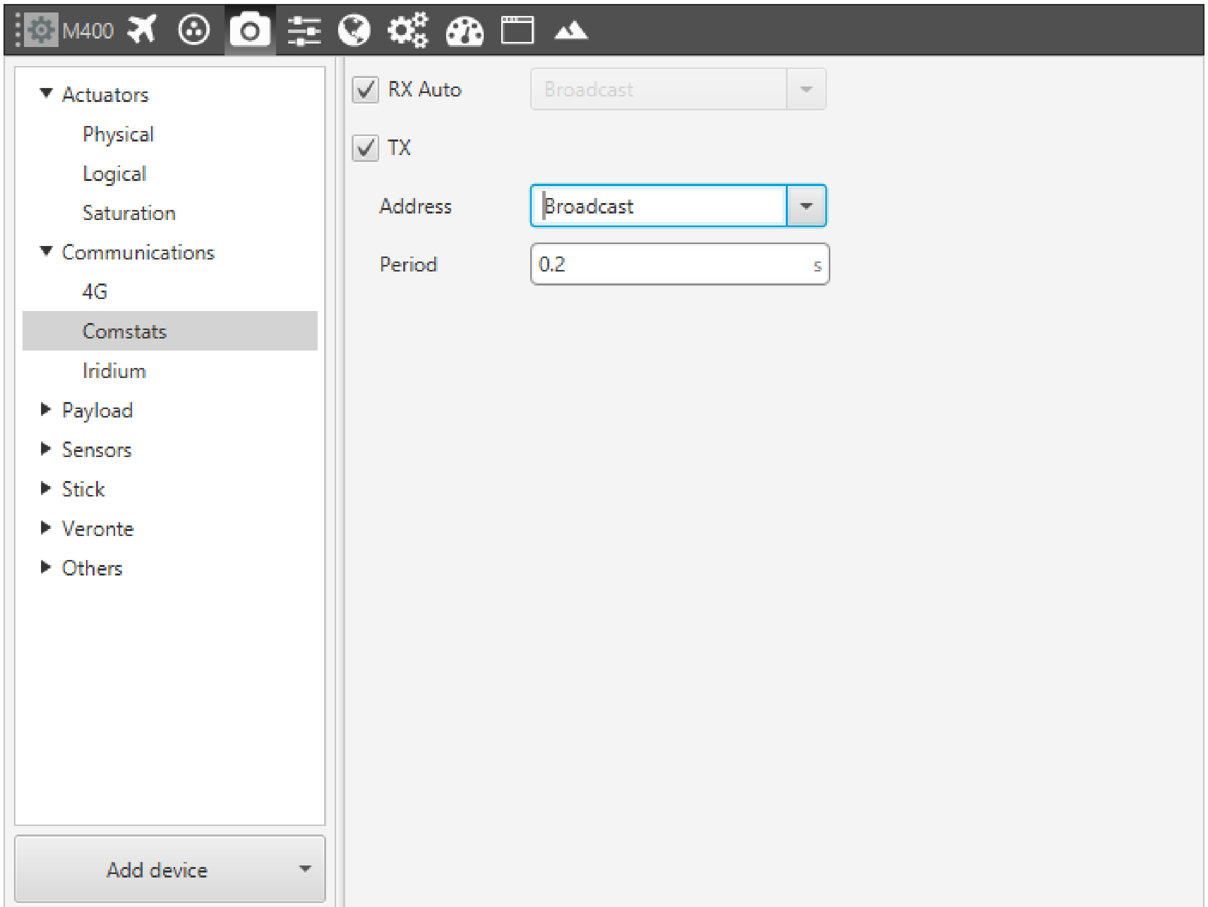


SIM Configuration for 4G

6.2.3.2.2 Comstats

In the Comstats menu, the user can configure this type of communications. There are two clickable buttons with which:

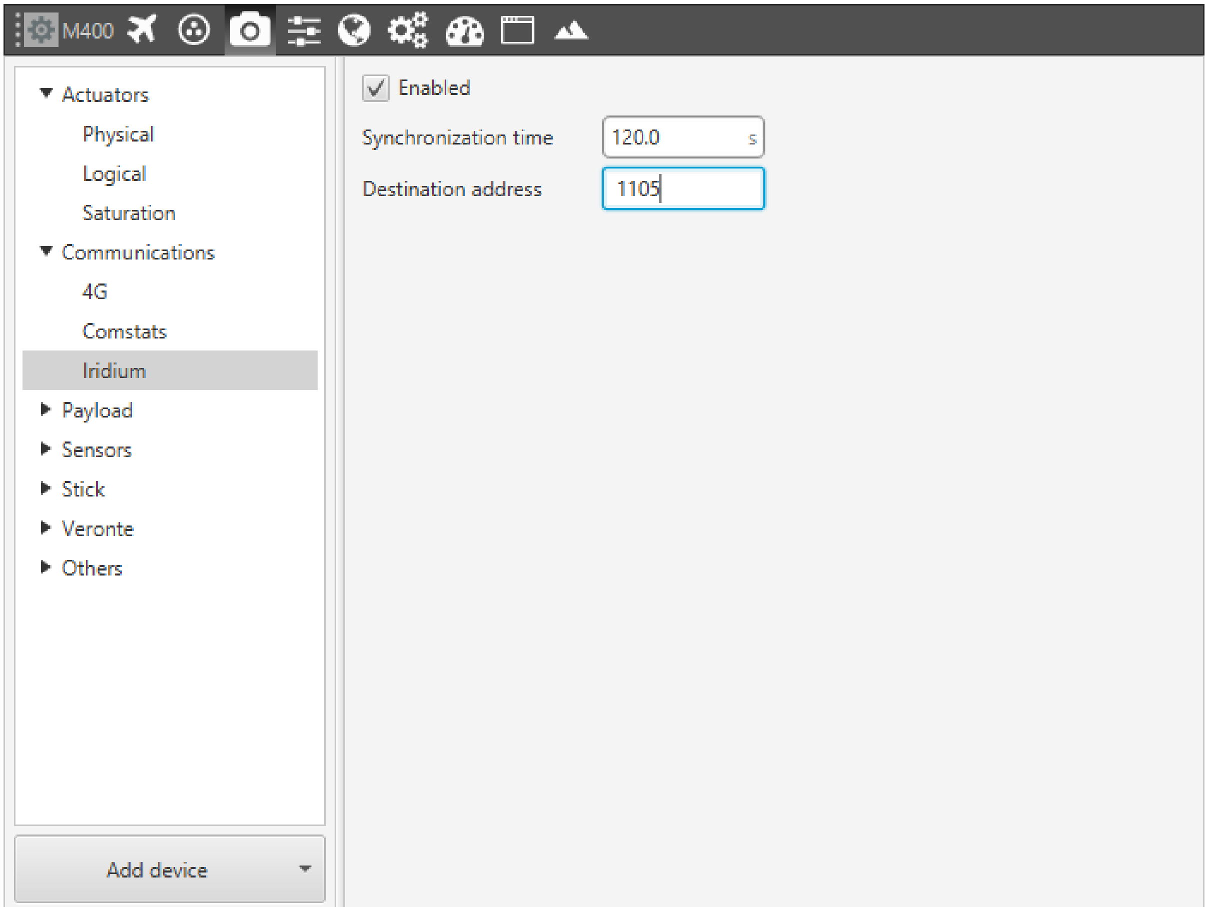
- **RX Auto.** Enabling this option will use the first remote AP found and shall be only one remote. If it is not select the user can choose between Broadcast and Pipe selected.
- **TX.** When enabling it, the user needs to define which platform is going to be monitored by the system, establishing its address and the period of transmission.



Comstats Configuration Menu

6.2.3.2.3 Iridium

If your system is ready for an Iridium communication, it can be activated from this menu. The user only needs to define the **Synchronization time** for the system and the **Destination address** for the unit to be connected with this service.

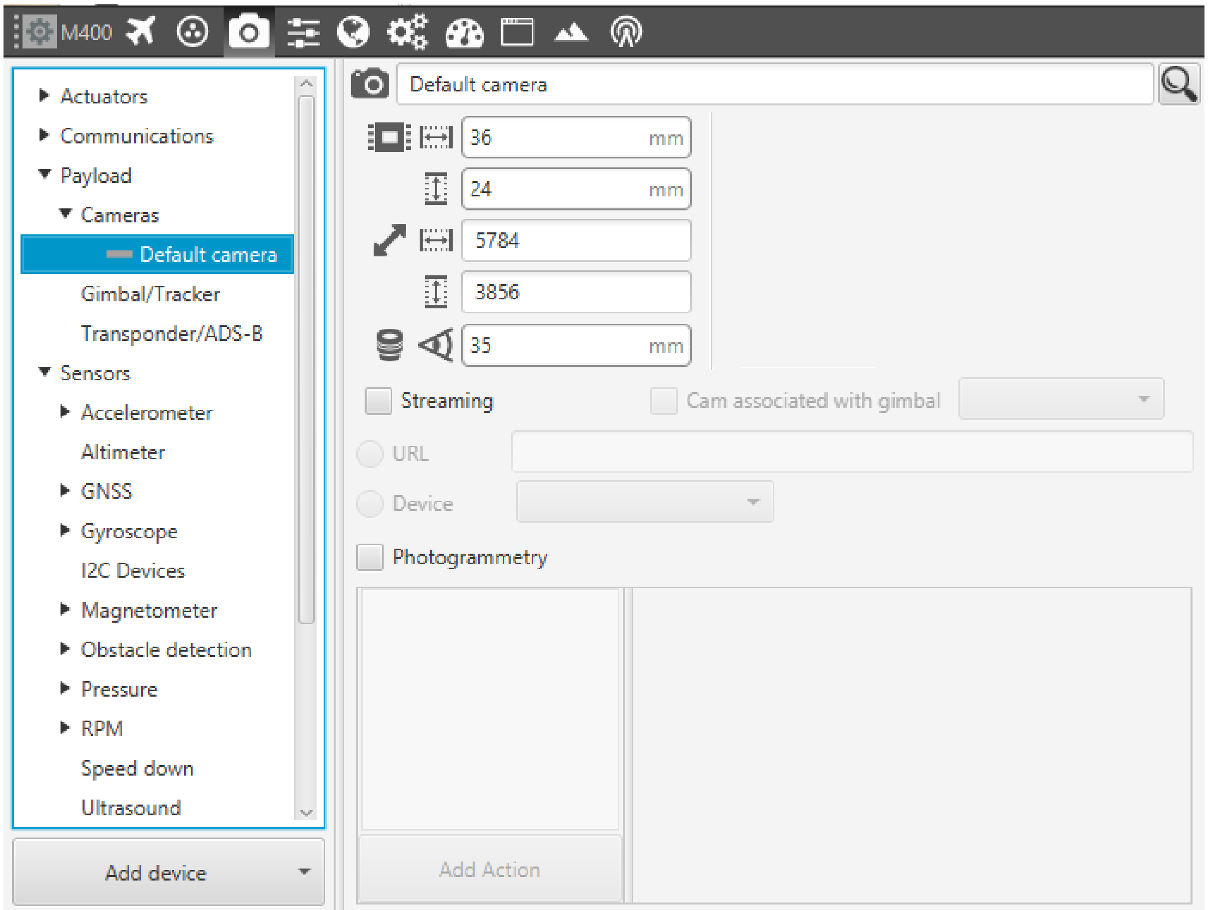


Iridium Configuration Menu

6.2.3.3 Payload

6.2.3.3.1 Camera

When adding a Device, the user can choose adding a Camera. This will create a default Camera under **Devices - Payload - Camera**.



Camera Menu - Configuration Parameters

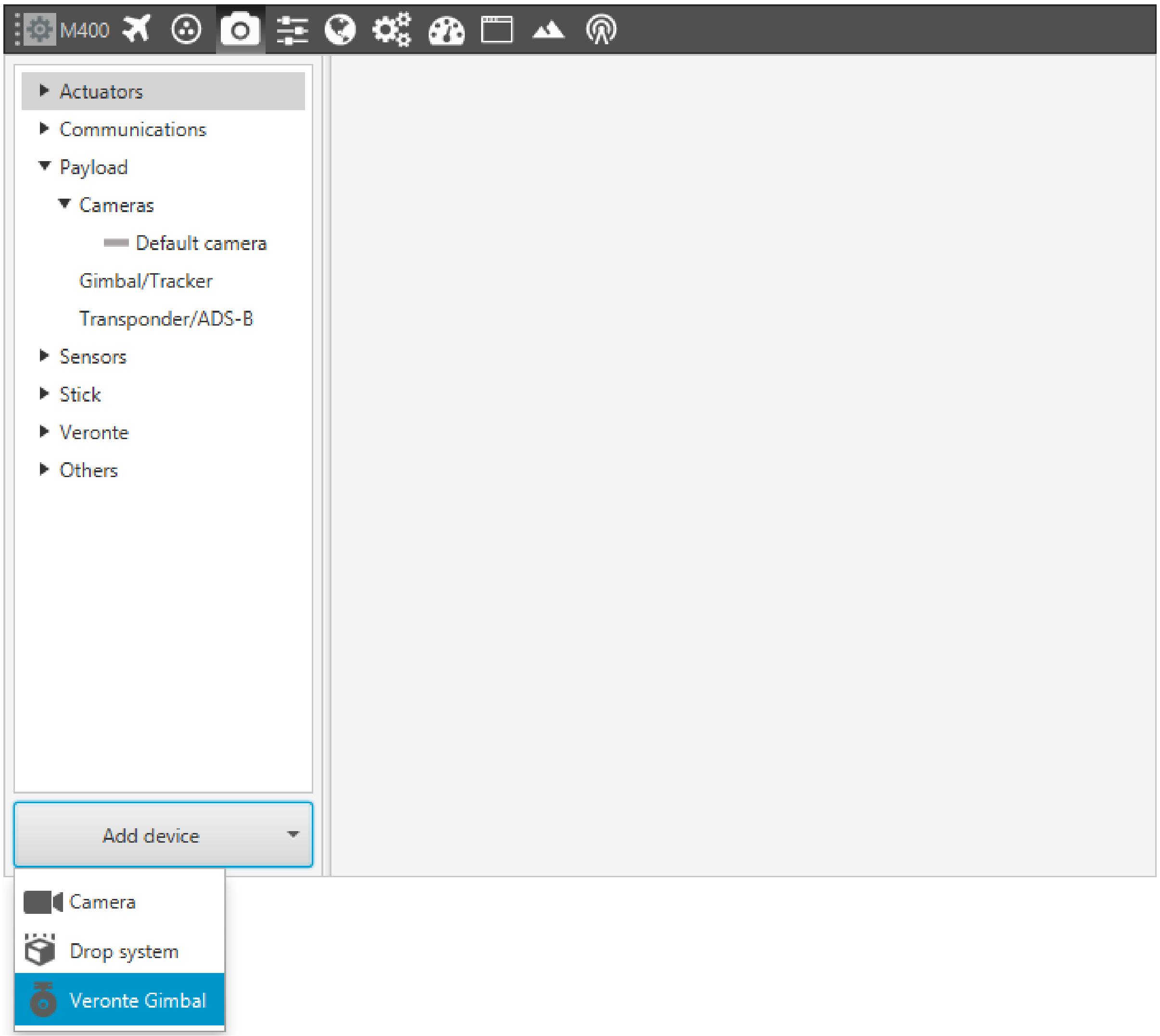
- **Name**. Put a name to the camera define or use the lens on the right to choose from the predefined list of cameras. Making this will automatically establish the following values (which in other way would have to be define manually):
  - **Sensor**. Defines the camera sensor width and height in mm.
  - **Resolution**. Defines the camera resolution width and height.
  - **Lens**. Defines the focal length from the camera in mm.
- **Streaming**. Section used to define a streaming service with which to configure a payload system to be able to have its video in Veronte Pipe, particularly in Workspace - Gimbal widget.
    - **URL**. Cameras whose protocol goes through Ethernet are configured introducing its URL in this field.

  If it is from a Gimbal device, it is important to be sure to configure the field *Cam asssociated with gimbal* if the user wants to move the Gimbal from the Widget.

    - **Device**. If the Camera is a system connected directly to the computer it will be selected from here. Generally speaking, Device 0 is linked to the Screen, Device 1 to the internal camera (if portable computer).
    - **Photogrammetry**. The actions undertaken in a Photogrammetry mission can be defined here, following the same possibilities as in Automations - Actions. A maximum of 4 Actions can be defined.

6.2.3.3.1.1 Veronte Gimbal Wizard

Veronte Pipe has an option which allows the user to configure from a basic menu all needed parameters for a Veronte Gimbal (in any of its four models).

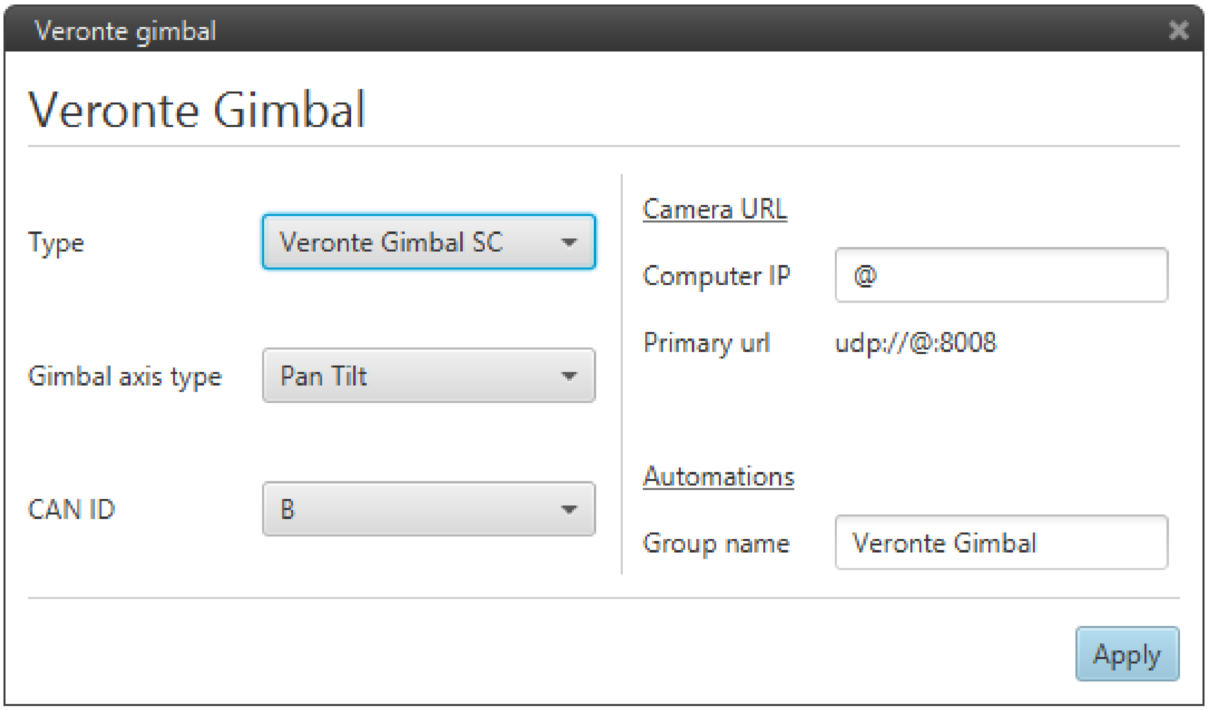


Camera Menu - Veronte Gimbal Wizard

The parameters to be configured are:

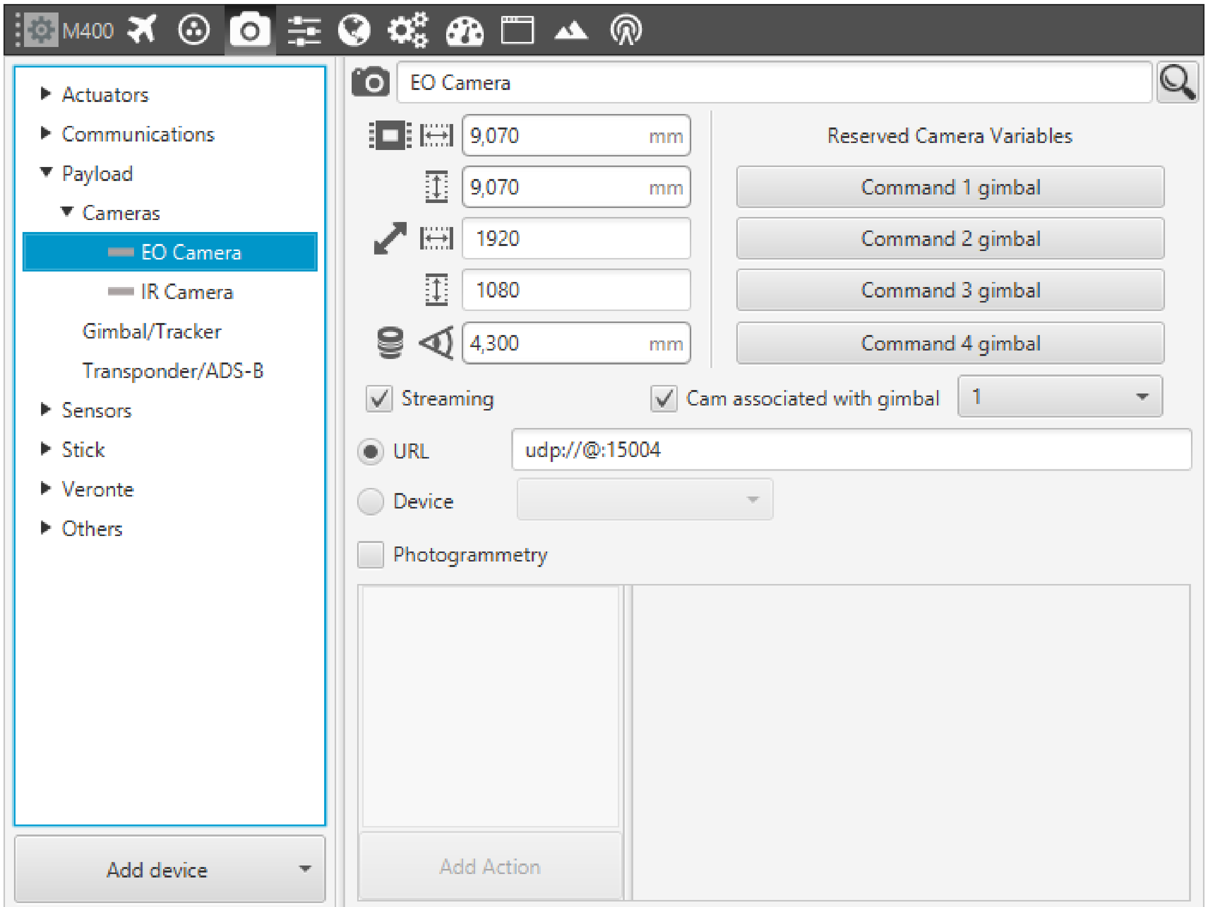
- **Type**. Defines the Veronte Gimbal model which is being configured:
  - Veronte Gimbal
  - Veronte Gimbal SC
  - Veronte Gimbal PRO
  - Veronte Gimbal PRO SC
- **Gimbal axis type**. Defines the angles that Veronte will be controlling from the payload system from a Combination of Pan (Z axis, the same as Yaw), Tilt (Y axis, the same as Pitch) and Roll. The two options are as seen below:

  - Pan & Tilt.
  - Roll & Tilt.
- **CAN ID**. Chooses the CAN which will be configured for using the Gimbal between A and B.
- **Camera URL**. Set the computer IP for the wizard to create the corresponding links between Gimbal and PC.
- **Automations**. Defines the name of the group of automations which will be created after the Wizard.



Camera Menu - Veronte Gimbal Wizard - Configuration Parameters

When clicking on apply, the wizard will automatically generate the CAN Messages, the cameras needed (either EO or IO depending on the model), the Gimbal configuration and a set of Automations for its use.



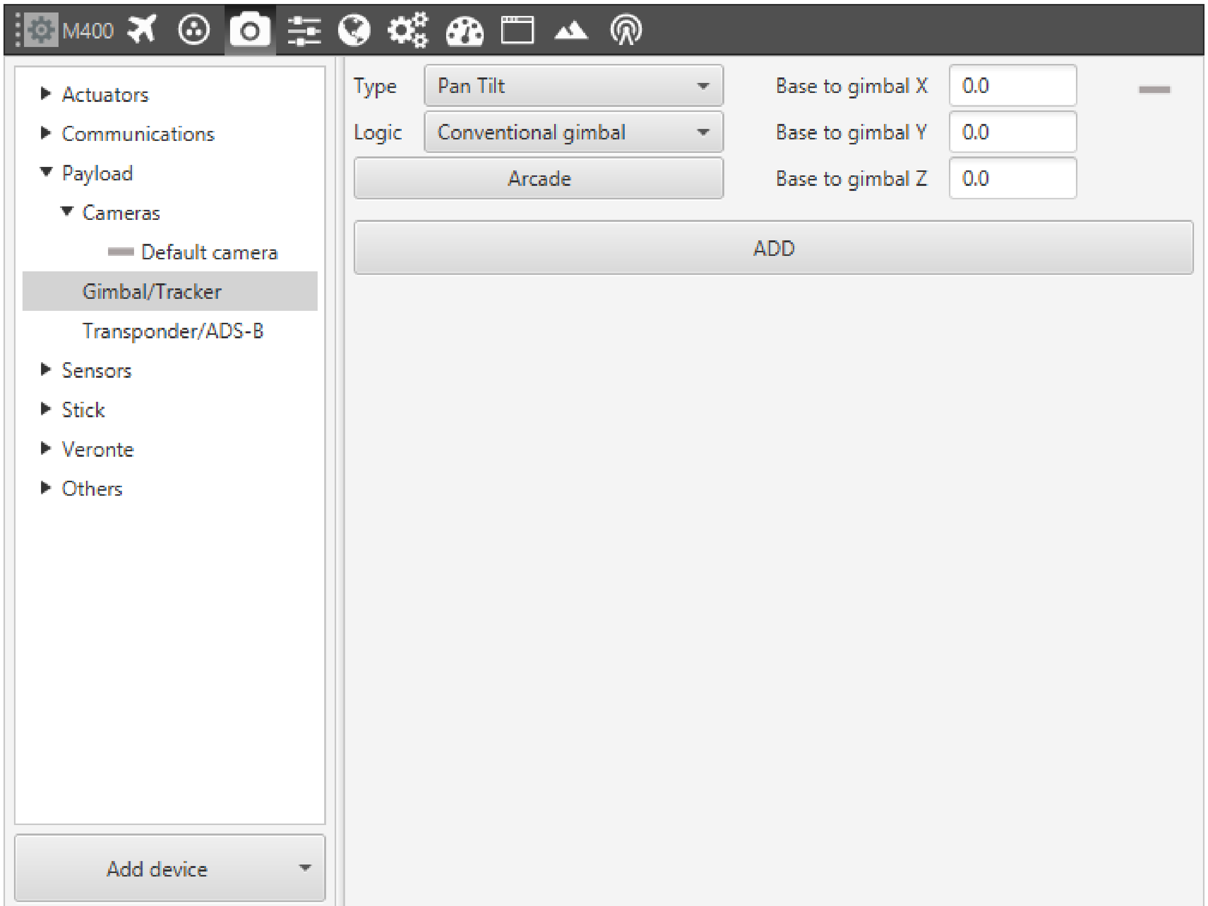
Camera Menu - Veronte Gimbal Wizard Results

Warning: If the camera has not been properly configured or the user wants to delete the previous configured model there are 4 menus to clear:

- **CAN A/B - Telemmetry - TX**
- **Devices - Payload - Cameras & Gimbal/Tracker**
- **Automations - Group Defined**

6.2.3.3.2 Gimbal/Tracker

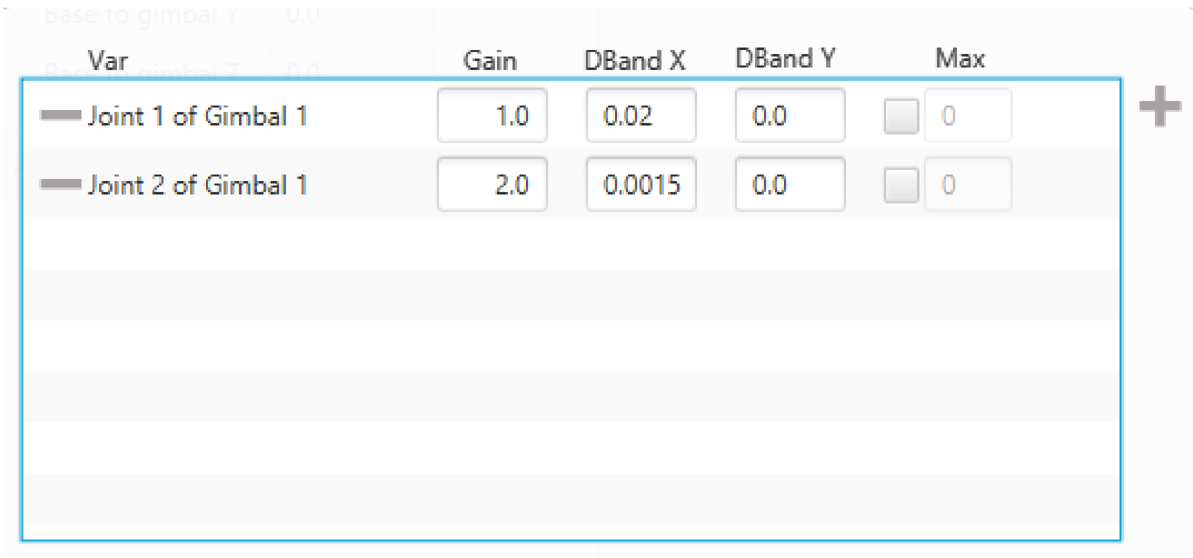
This menu allows the user to configure a Gimbal Camera or a Tracker Antenna. From here the user only needs to define the movements the system has (from predefined combinations of Pan, Tilt and Roll), its logic, a distance vector and the number of systems connected (up to a maximum of 2).



Gimbal/Tracker Menu - Configuration Parameters

- **Type**. Defines the angles that Veronte will be controlling from the payload system from a Combination of Pan (Z axis, the same as Yaw), Tilt (Y axis, the same as Pitch) and Roll. The three options are as seen below:
  - Pan & Tilt.
  - Pan, Roll & Tilt.
  - Roll & Tilt.
- **Logic**. Defines the kind of payload system configured.
  - **Conventional gimbal**. This option writes over the variables Joint 1-3 Gimbal 1-3 which are used later to configure the control and stabilization of the camera from Veronte.
  - **Self-stabilized gimbal**. The payload system only needs movement inputs and the variables mentioned will have no output.
- **Base to gimbal Vector**. Defines the vector thats joins the Veronte Autopilot controlling the payload system and the payload system itself, in Veronte body axes.

- **Arcade**. Configures the Arcade control of the payload system. For a new variable to be controlled, click on the top right "+" icon.
  - **Var**. Displays the variable over which the Arcade control will have its effect.
  - **Gain**. Value which multiplies the Var. to obtain the control value.
  - **DBand X**. Creates a non-effect area in which the Joystick won't input control values. DBand X is applied to both left and right on the X axis. Value used to stop the stick from inputting control when at rest.
  - **DBand Y**. Same as DBand X but applied to the Y axis.
  - **Max**. Defines a maximum value for the control output. Useful if it has been defined a big Gain.



Gimbal/Tracker Arcade - Configuration Parameters

Any further configuration will strongly depend on the payload system selected and its control signals. Check the manuals section for Veronte Gimbal manual as a reference.

6.2.3.3.3 Transponder

Veronte is compatible with:

- Sagetech
- uAvionix
  - Mode S Transponder ping 200Sr/200S
  - Mode S Transponder ping 20S

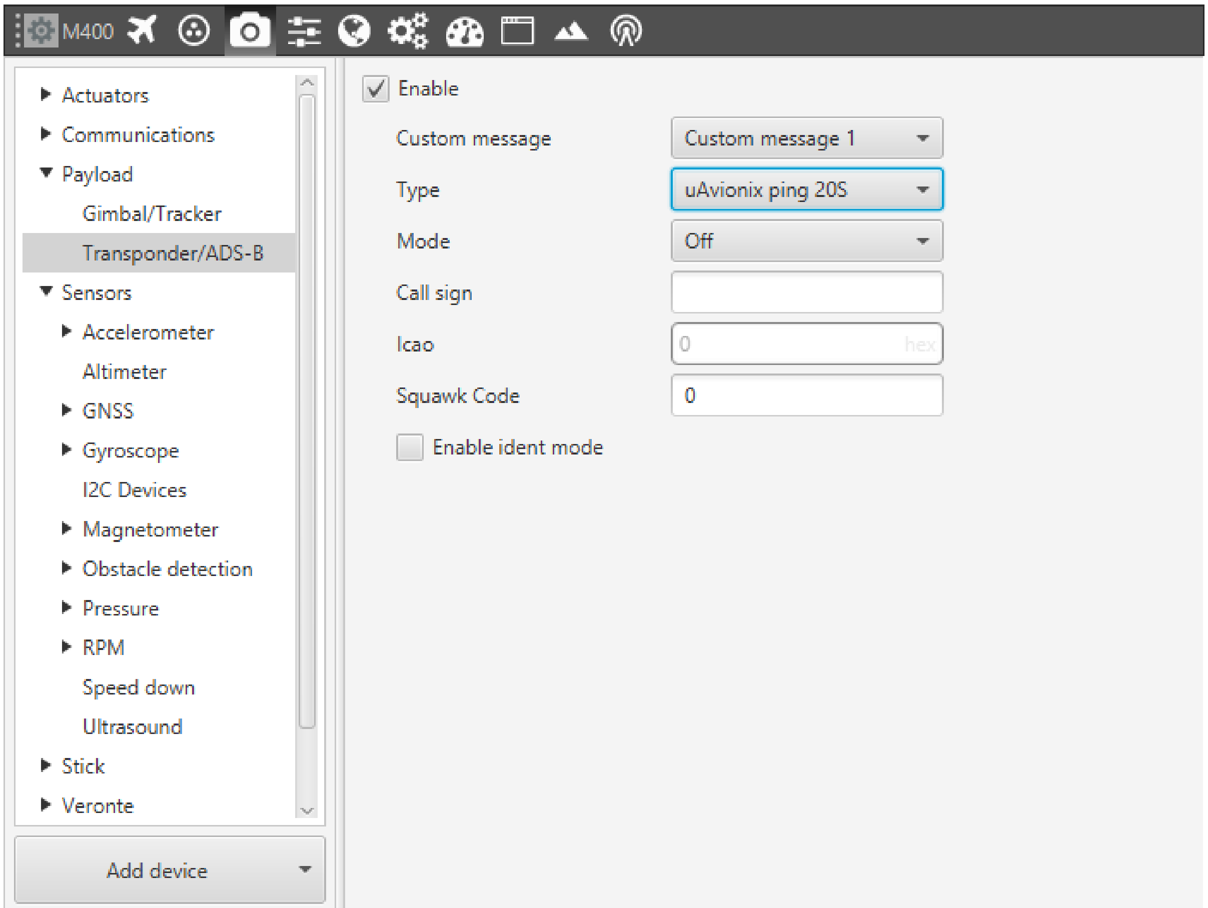
The configuration displayed below is for a Transponder.

1. The first step to configure it is to go to **Device – Payload – Transponder/ADS-B** and check the enable button. Once there, select the **Custom message** to be used for the information exchange, since it will automatically be filled with the information that the Transponder needs.
2. Choose the model already mentioned (uAvionix ping20S)
3. Switch to the desired **Mode**
   - **OFF:** Transponder switched off

- **STANDBY:** Transponder will not respond to interrogation.
- **ON:** Replies to interrogations with 4-digit squawk code.
- **ALT:** Replies to interrogation with altitude information.
- **Enable Ident Mode:** UAV identification under ATC request (only available for ON and ALT modes)

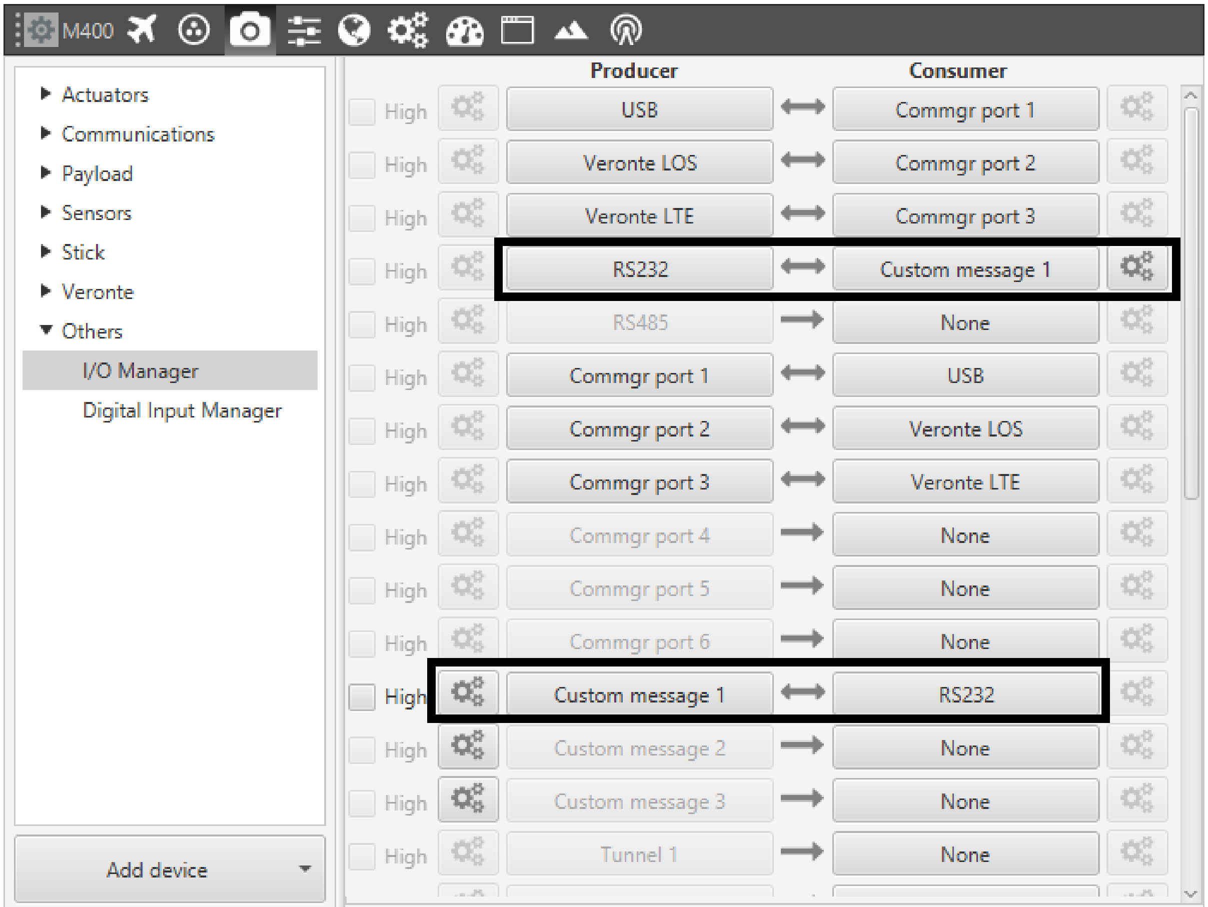
4. Introduce the **Squawk Code** provided.

There is an example configuration in the following picture from uAvionix ping 20S.



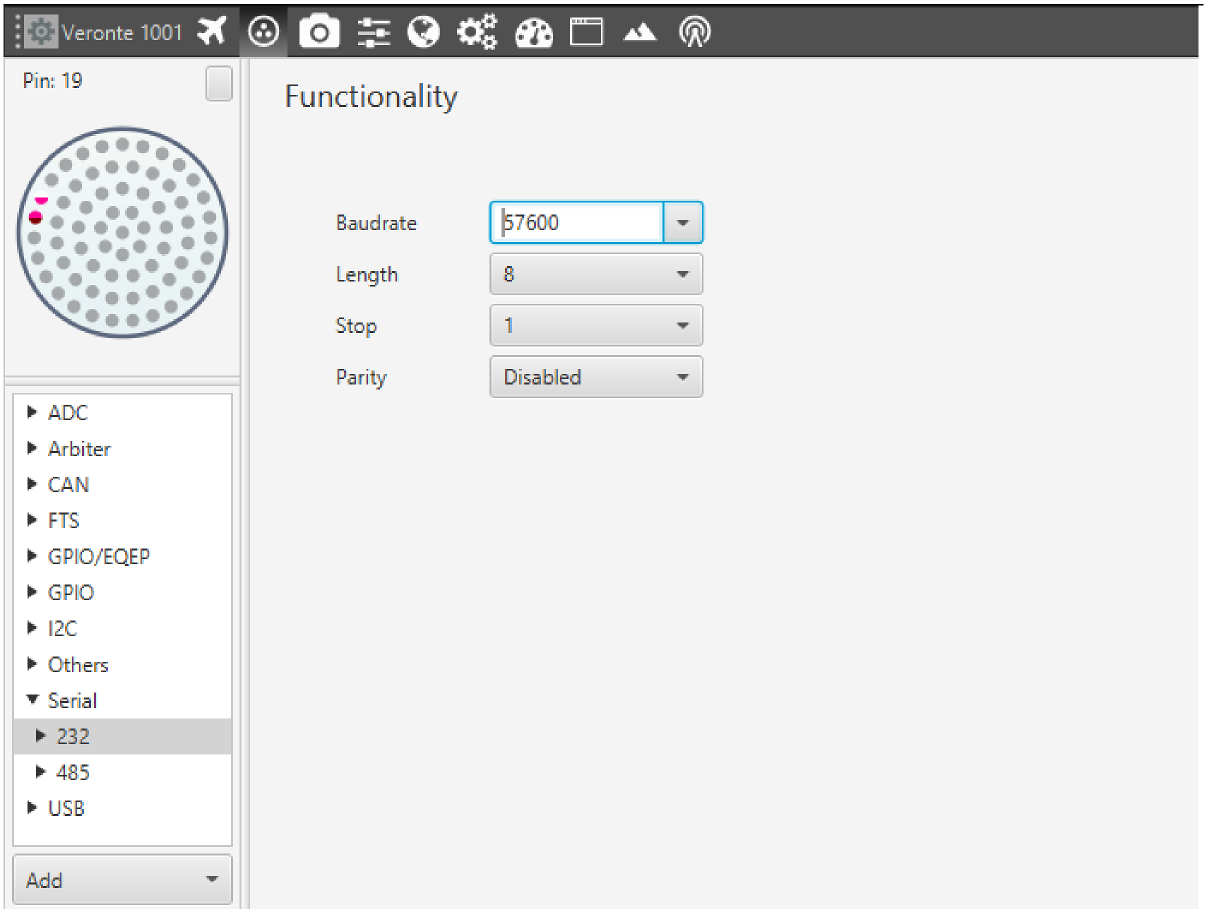
Transponder Configuration Parameters

Once the transponder is configured, go to **Devices – Others – I/O Manager**. Create a bind-bidirectional communication RS232/RS485 (in the example, through RS232) and assign to it the Custom Message selected before.



RS232 Message Reception

Note that either RS232 or RS485 must be configured with the corresponding Baudrate from the user transponder in **Connections – Serial – 232/485**.



RS232 Configuration Menu

6.2.3.3.4 ADS-B

Veronte is compatible with:

- Sagetech
- uAvionix
  - ADS-B Transceiver ping 1090/1090i
  - ADS-B Transceiver ping 2020/2020i
  - ADS-B Receiver ping Rx

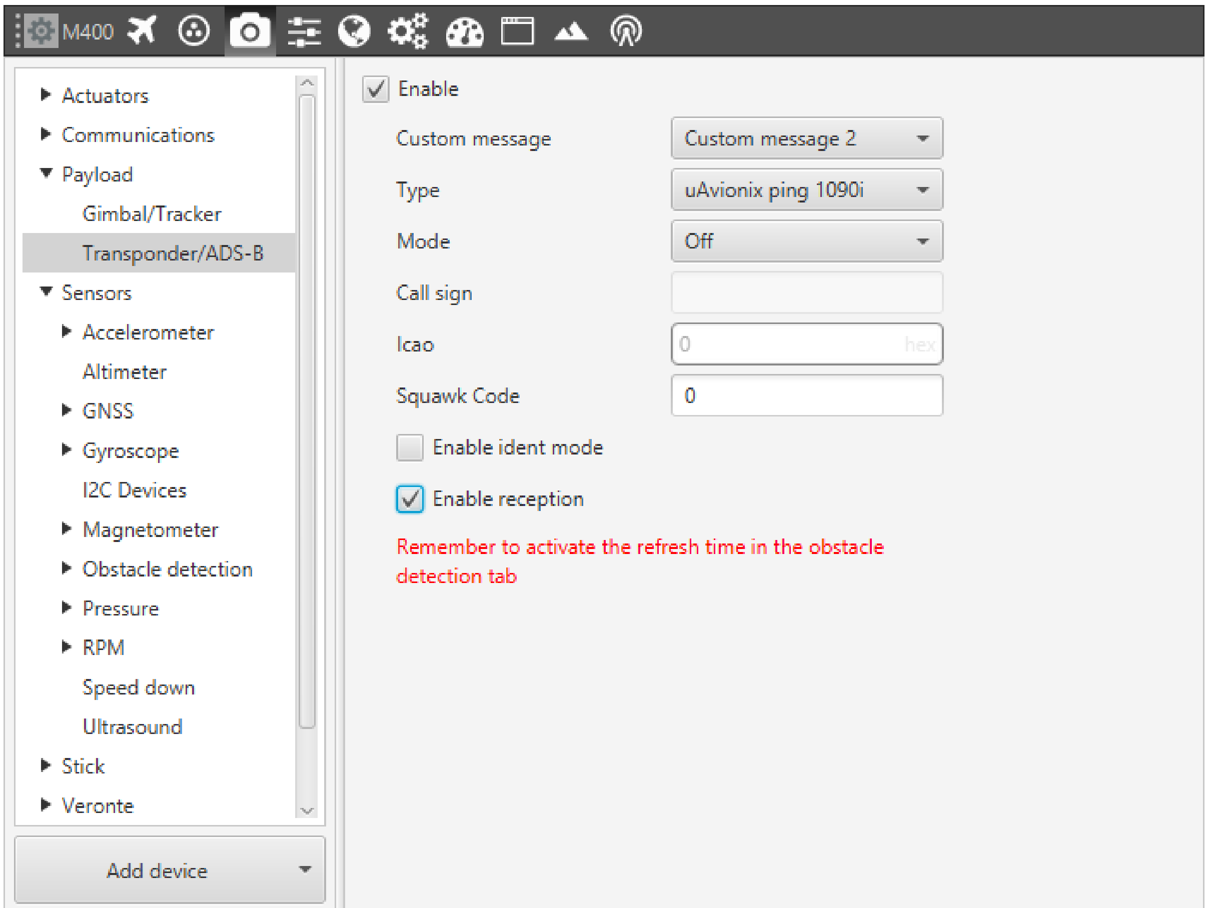
The configuration displayed below is for an ADS-B.

1. The first step to configure it is to go to **Device – Payload – Transponder/ADS-B** and check the enable button. Once there, select the **Custom message** to be used for the information exchange, since it will automatically be filled with the information that the ADS-B needs.
2. Choose the model already mentioned (uAvionix ping20S)
3. Switch to the desired **Mode**
   - **OFF:** Transponder switched off
   - **STANDBY:** Transponder will not respond to interrogation.

- **ON:** Replies to interrogations with 4-digit squawk code.
- **ALT:** Replies to interrogation with altitude information.
- **1090ES:** ADS-B transmit is always enabled when a 6-digit ICAO code is entered.
- **Enable Ident Mode:** UAV identification under ATC request (only available for ON and ALT modes)

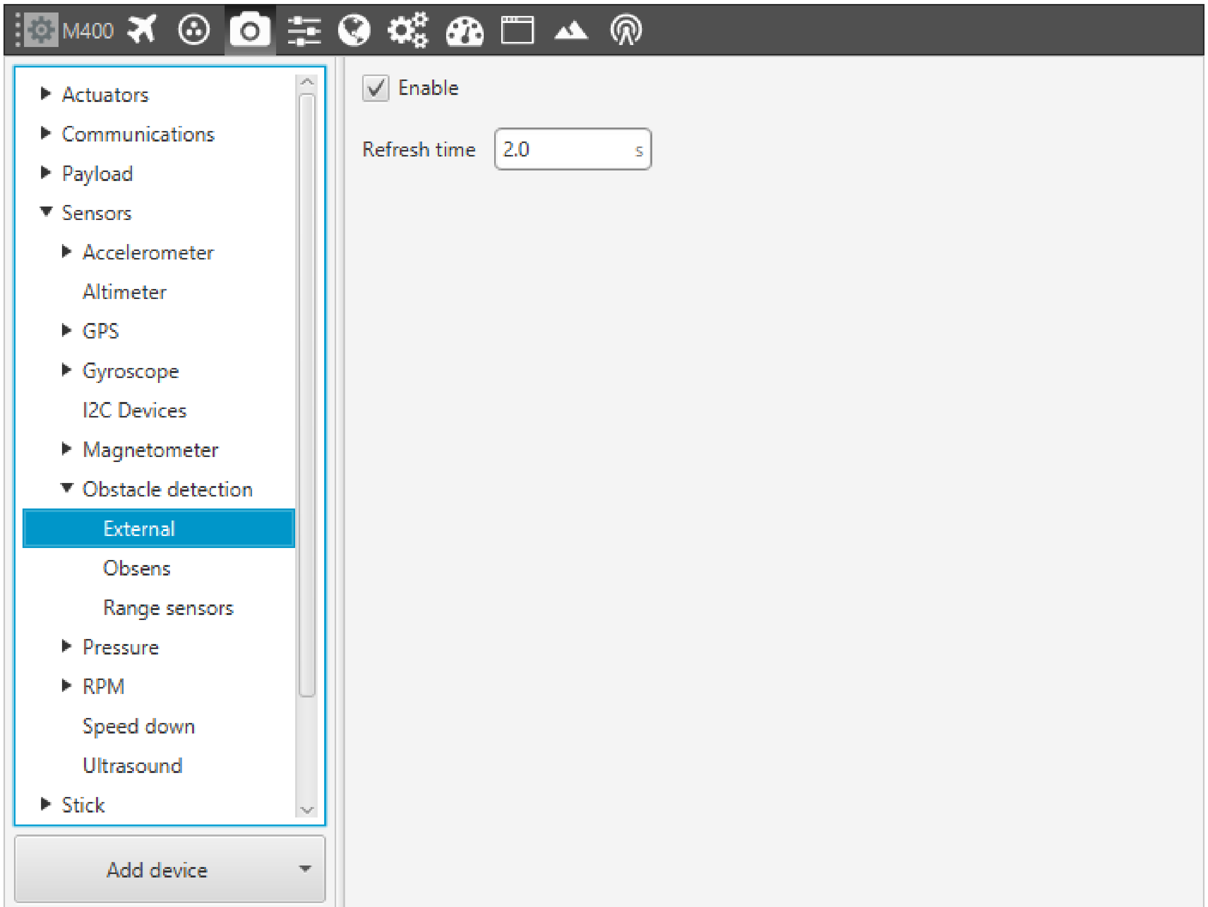
4. Introduce the **Squawk Code** provided.
5. Enable reception.

There is an example configuration in the following picture from uAvionix ping 1090/1090i.



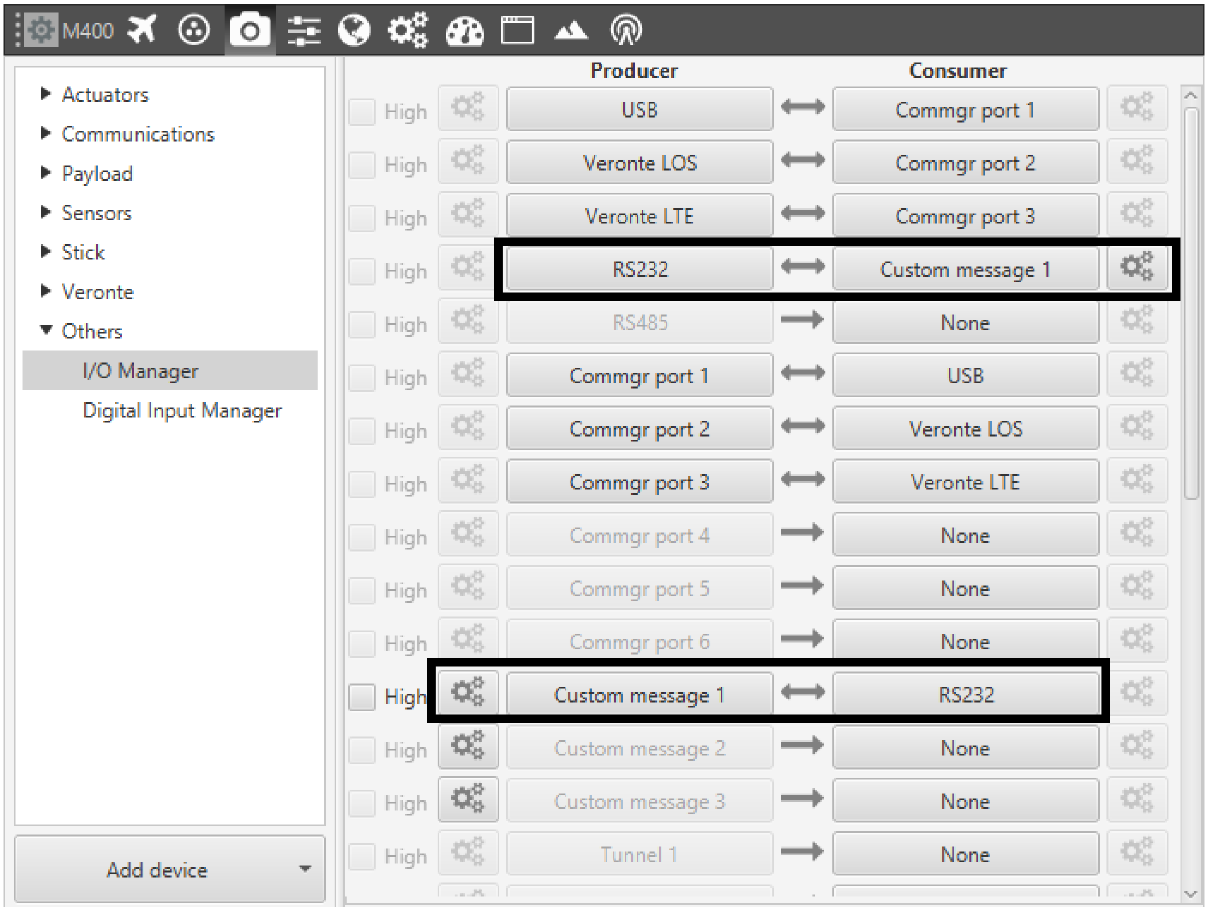
ADS-B Configuration Parameters

The last step you need to do is to enable the External Object Detection in Devices – Sensors – Obstacle Detection.



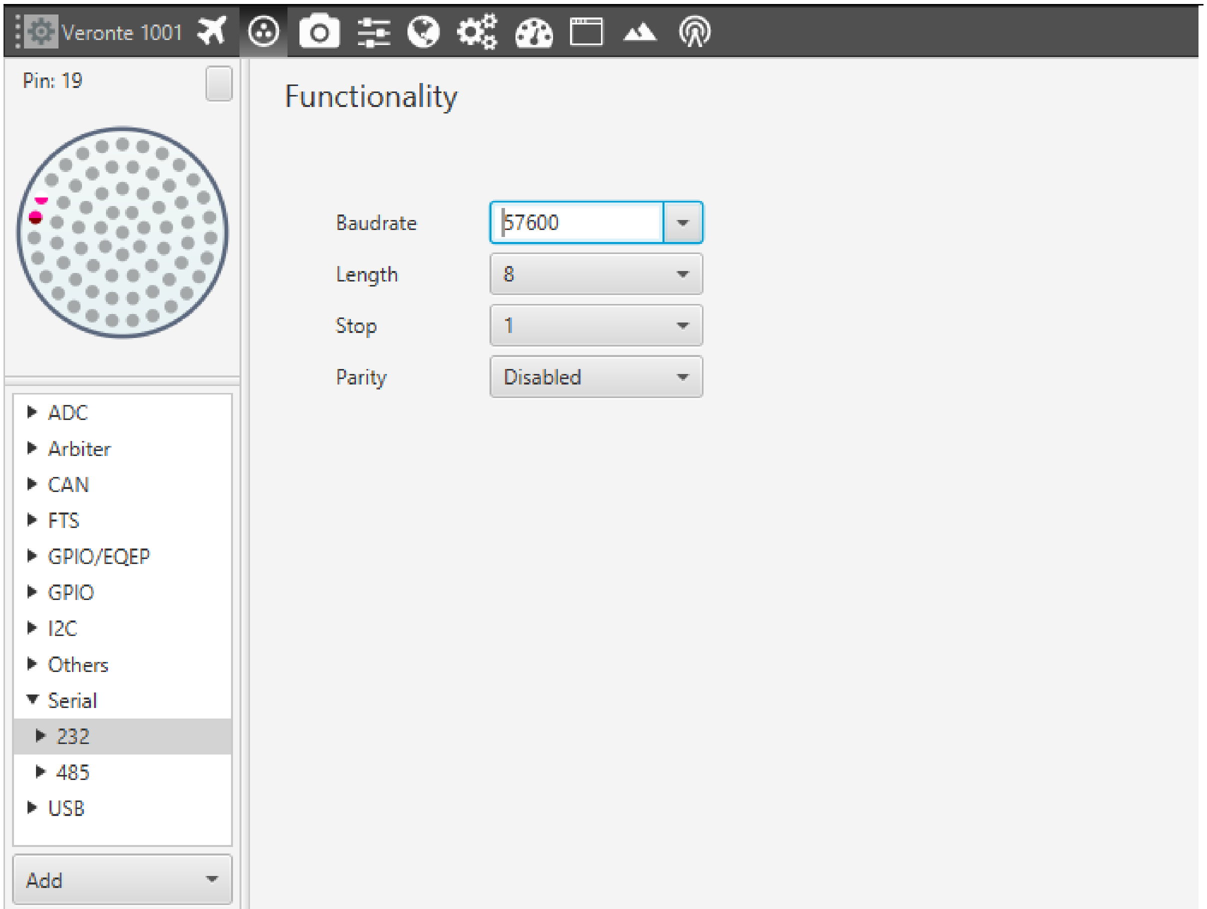
External Obstacle Detection Menu

Once the transponder is configured, go to **Devices – Others – I/O Manager**. Create a bind-bidirectional communication RS232/RS485 (in the example, through RS232) and assign to it the Custom Message selected before.



Transponder Configuration - Serial Communication

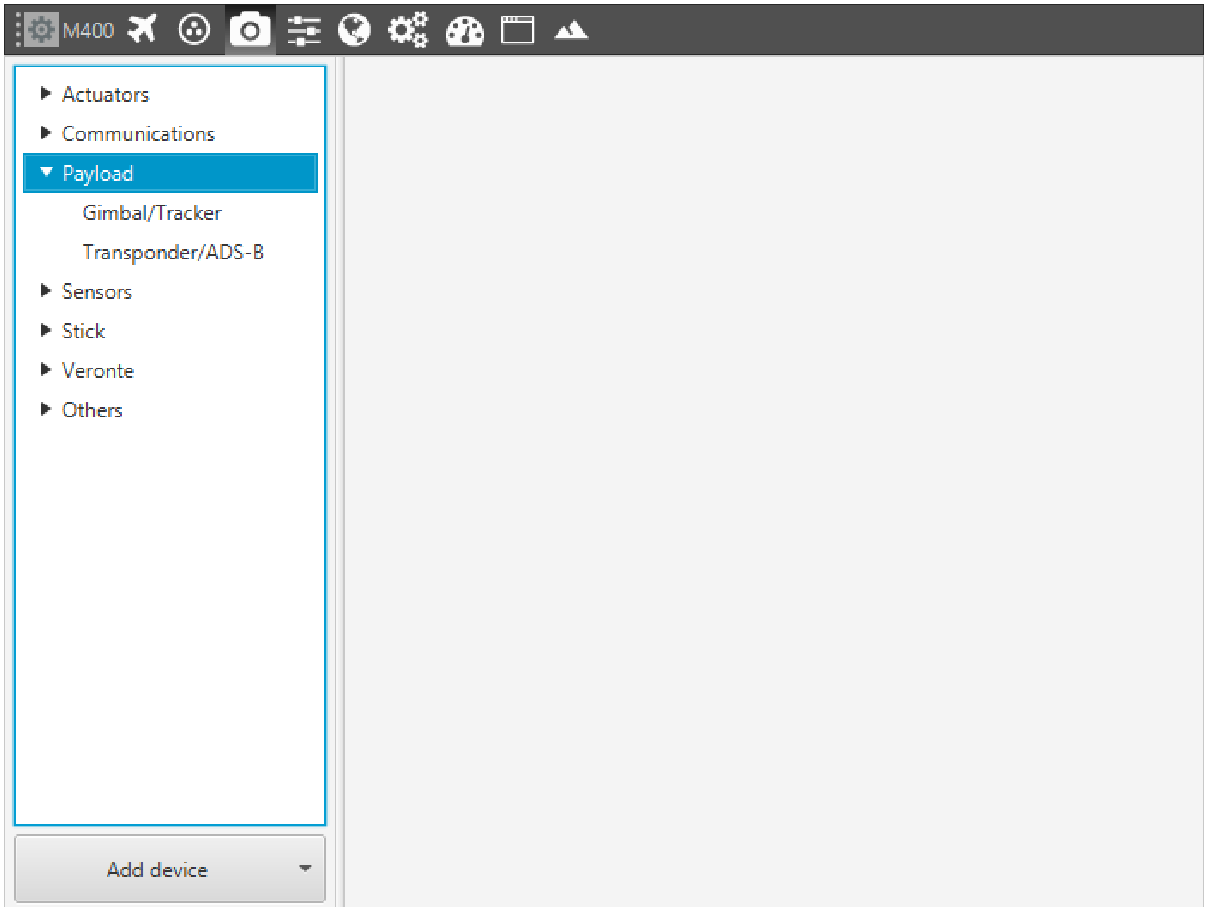
Note that either RS232 or RS485 must be configured with the corresponding Baudrate from the user transponder in **Connections – Serial – 232/485**.



Transponder Configuration - Serial Configuration

In order to see the information from the ADS-B, aside from having visual objects on the screen, the user can use a Widget (refer to Workspace – *ADS-B Decoder*)

The following menu displays the possible Payload that can be configured with Veronte. Each window will allow the user to configure different parameters from the available variety of payloads.



Payload Menu

6.2.3.4 Sensors

6.2.3.4.1 Accelerometer

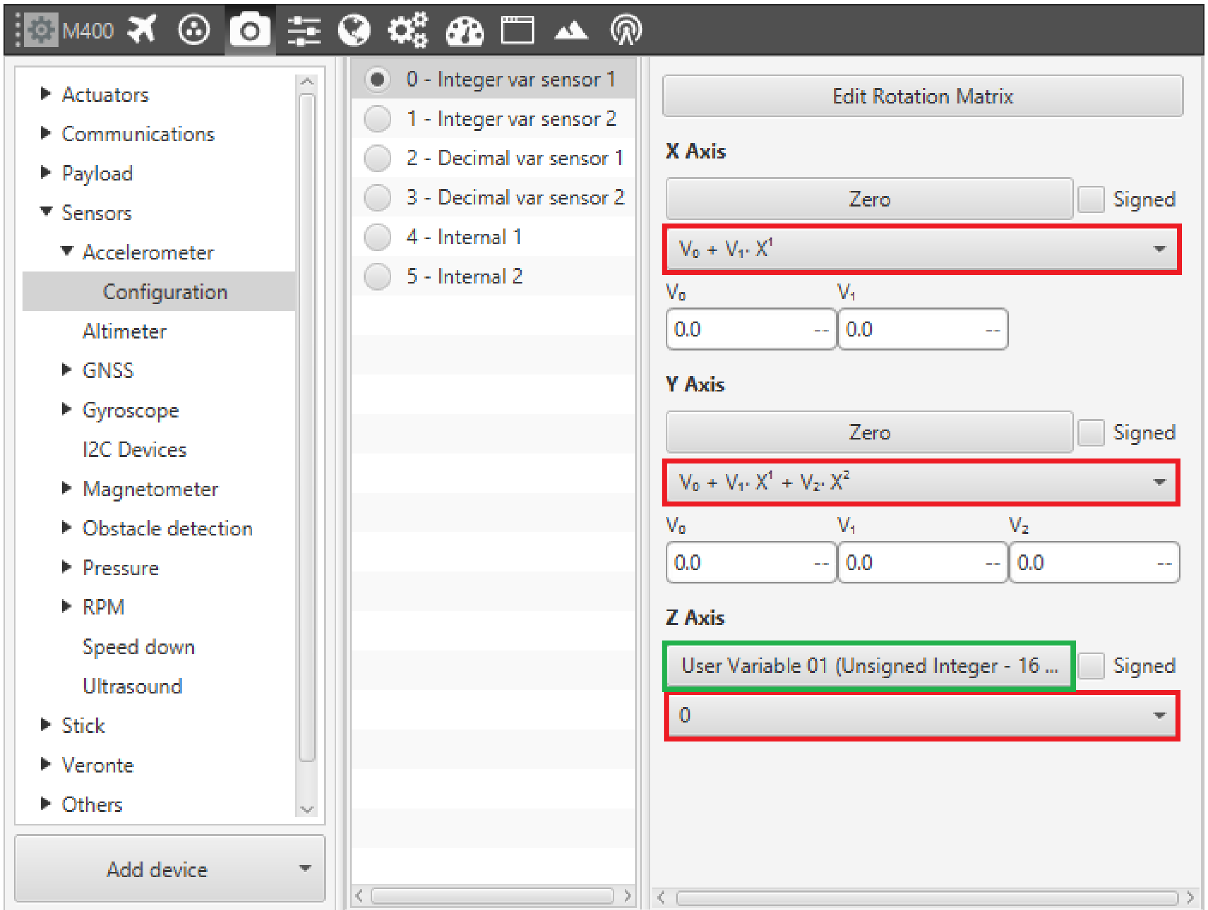
Veronte incorporates two Inertial Measurement Units (IMUs) that allows the Veronte System to measure different variables and that are the main navigation data source. From the IMU, the user can configure the Accelerometer and Gyroscope. The first one is explained below.

The user can choose between 3 types of source for the accelerometer.

- **Integer var sensor 1-2**. Veronte uses a integer value provided by an external sensor.
- **Decimal var sensor 1-2**. Veronte uses a decimal value provided by an external sensor.
- **Internal 1-2**. Veronte uses the internal sensor.

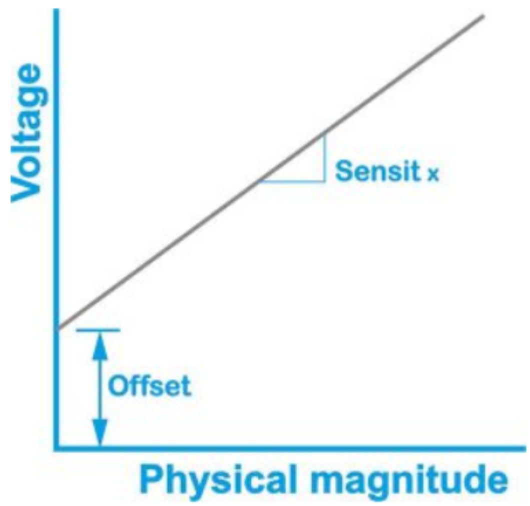
6.2.3.4.1.1 Integer var sensor

In this menu it is possible to configure integer variables provided by an external sensor.



Integer var Accelerometer Menu - Configuration Parameters

When Integer var sensor 1 or 2 are selected, the previous panel will be shown. In this panel, the user selects the variable that has been stored in a user variable (Green Box) and the operations that will be carried on (Red Box). It is possible to use the signal through a linear or quadratic relation. The following image shows an example of a linear relation.

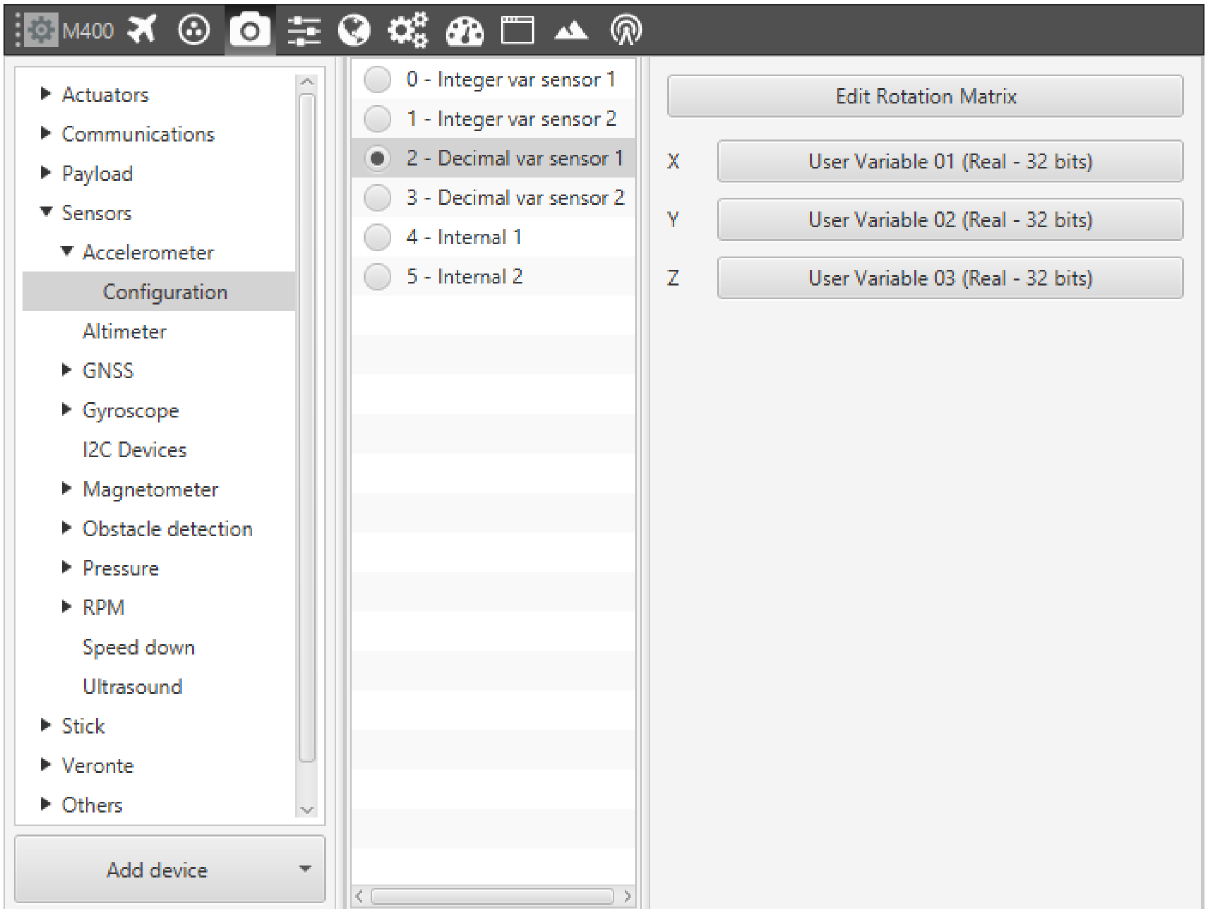


Linear relation of 2 Variables

The process of configuration has to be done using **Custom Messages**. This is to be configured in **Devices - Others - Digital - I/O Manager**. The configuration will depends on the device in use and its communication protocol.

6.2.3.4.1.2 Decimal var sensor

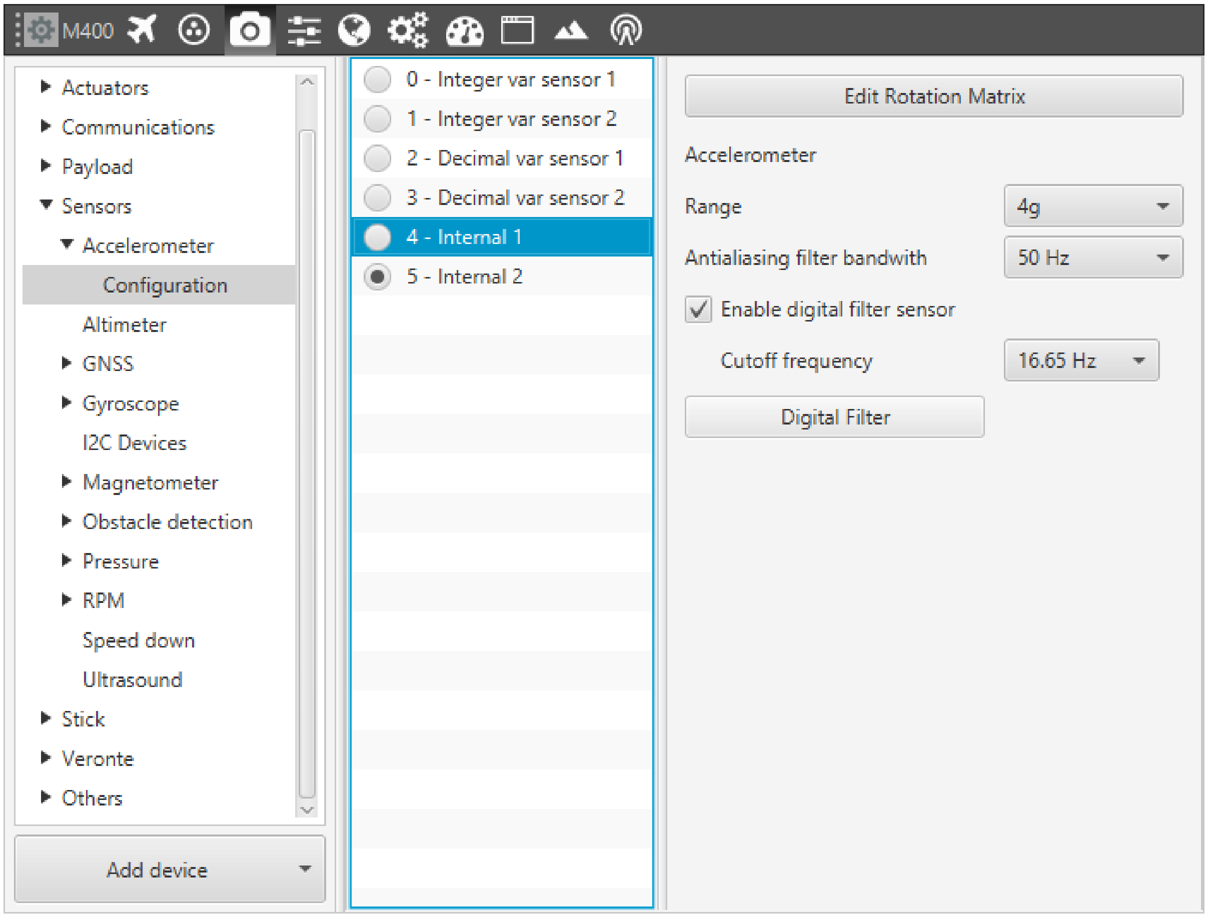
In this menu, the user selects real variables for each axis (X,Y,Z), these do not requiere a signal treatment. The process of configuration is similar to the one carried out when configuring a Integer Variable.



Decimal var Accelerometer Menu - Configuration Parameters

6.2.3.4.1.3 Internal

This last menu available displays the posible parameters that can be configured for the internal Accelerometer.



Accelerometer Menu - Configuration Parameters

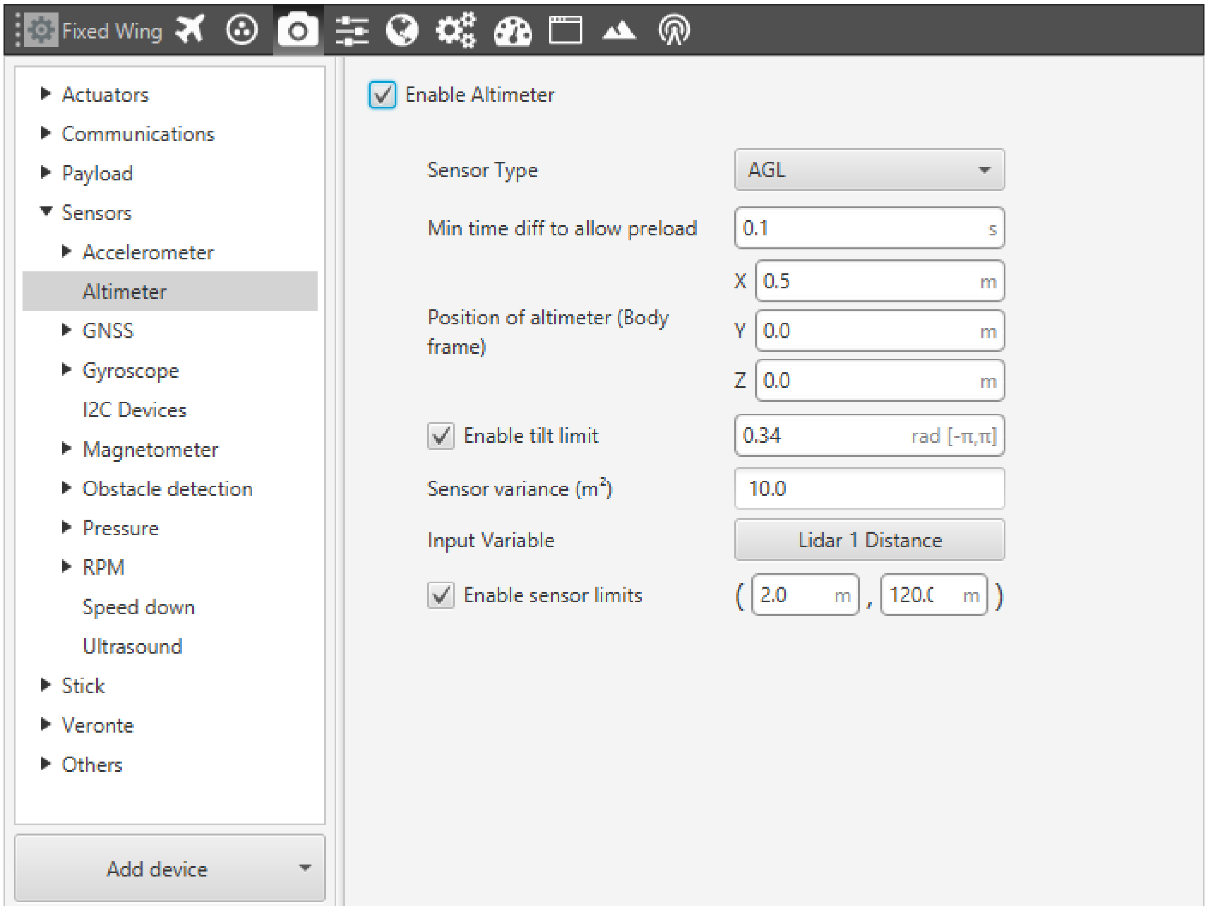
Warning: **Edit Rotation Matrix** brings the position of the accelerometer inside the Veronte Autopilot, it must **NOT** be changed under any circumstance.

In this menu it is possible to set different options regarding range and filters from the accelerometer. The parameters that can be modified are:

- **Range**. Sets the maximum range of performance, high ranges implies less precision while small ranges might mean the system saturates. Values allowed are 2g, 4g, 8g and 16g.
- **Antialiasing filter bandwith**. It is the bandwidth of the antialiasing **low pass filter**. The options available are 50Hz, 100Hz, 200Hz and 400Hz, the greater the value selected the worse the filtering will be.
- **Enable digital filter**. Enables a low pass filter which its cutoff frequency is configured from the options 16.65Hz, 66.6Hz, 133.2Hz and 740.0Hz. This is a **hardware filter**, included directly in the Accelerometer.
- **Digital filter**. Enables a low pass filter which its cutoff frequency is configured manually, allowing the user to input any desired value in Hz. It is a **software filter**, applied after the hardware filter from the point before.

6.2.3.4.2 Altimeter

The following figure shows the configuration menu for an external altimeter like LIDAR, Sonar, etc.

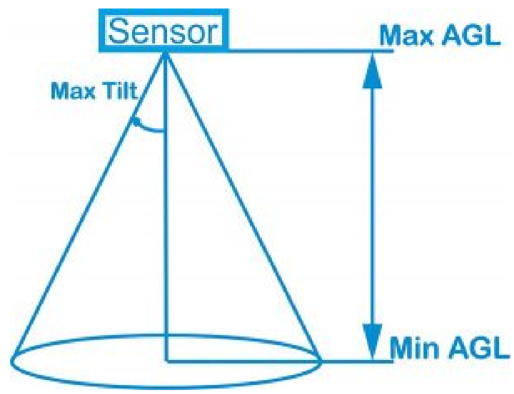


Altimeter Menu - Configuration Parameters

- **Sensor Type**. The system configured in this menu could be used to measure the altitude of the aircraft or to detect the distance from the vehicle to a certain object. This option allows the change between these 2 operation modes
  - AGL for altitude measurement
  - Distance for measurement to object detection.
- **Min time diff to allow preload**. Establishes the maximum frequency at which this sensor values will enter the Navigation filter. I.e. if it is executed at 50 Hz and we set (as in the picture 0.1 s) 10 Hz, there will be loops when the value is the one stored before.
- **Position of altimeter (Body frame)**. Parameter to indicate the distance from the altimeter to the centre of gravity of the platform. This is used to take into account the weight of the altimeter in the aircraft control.
- **Enable tilt limit**: The altimeter is normally installed in a fixed position having a constant direction with respect to the platform. Taking a LIDAR as an example, it is used to measure altitude so it has to point towards the ground, in a direction parallel to the Z body axis. When the vehicle is not level on its X or Y axis (has a pitch or roll angle different from zero), the LIDAR will not point in a direction perpendicular to the ground, and the measurement taken will not be the real altitude of the aircraft. This option is a safe condition to discard the measure of an altimeter when its tilt angle exceeds a certain value.

- **Sensor variance**. It is a measurement related to sensor error, whose units are squared meters. It indicates the weight that this measure will have in the sensor fusion algorithm that combines the information of different sensors to generate an altitude estimation. For example, if the altimeter installed is a LIDAR, whose precision is quite high, the value of sensor variance has to be low. Actually, the value of the sensor variance is fixed and depends on each sensor, commonly a LIDAR has an error of 0.5 meters more or less, so the variance will be 0.25. Now it depends on the application if the user wants to put this value or a higher one, that will imply that the altitude information depends not almost only on the LIDAR reading.
- **Input Variable**. User chooses the variable in which the sensor data is stored to be processed by Veronte Pipe. There are 5 Lider variables in case it is configured.
- **Enable sensor limits**. It is the range in which the sensor measurement is taken to be processed by Veronte Pipe. Any outer value will be discarded by the system.

The following figure shows a diagram with the values of maximum and minimum sensor limits altitude, and the maximum tilt angle.



Altimeter Menu - Sensor Limits

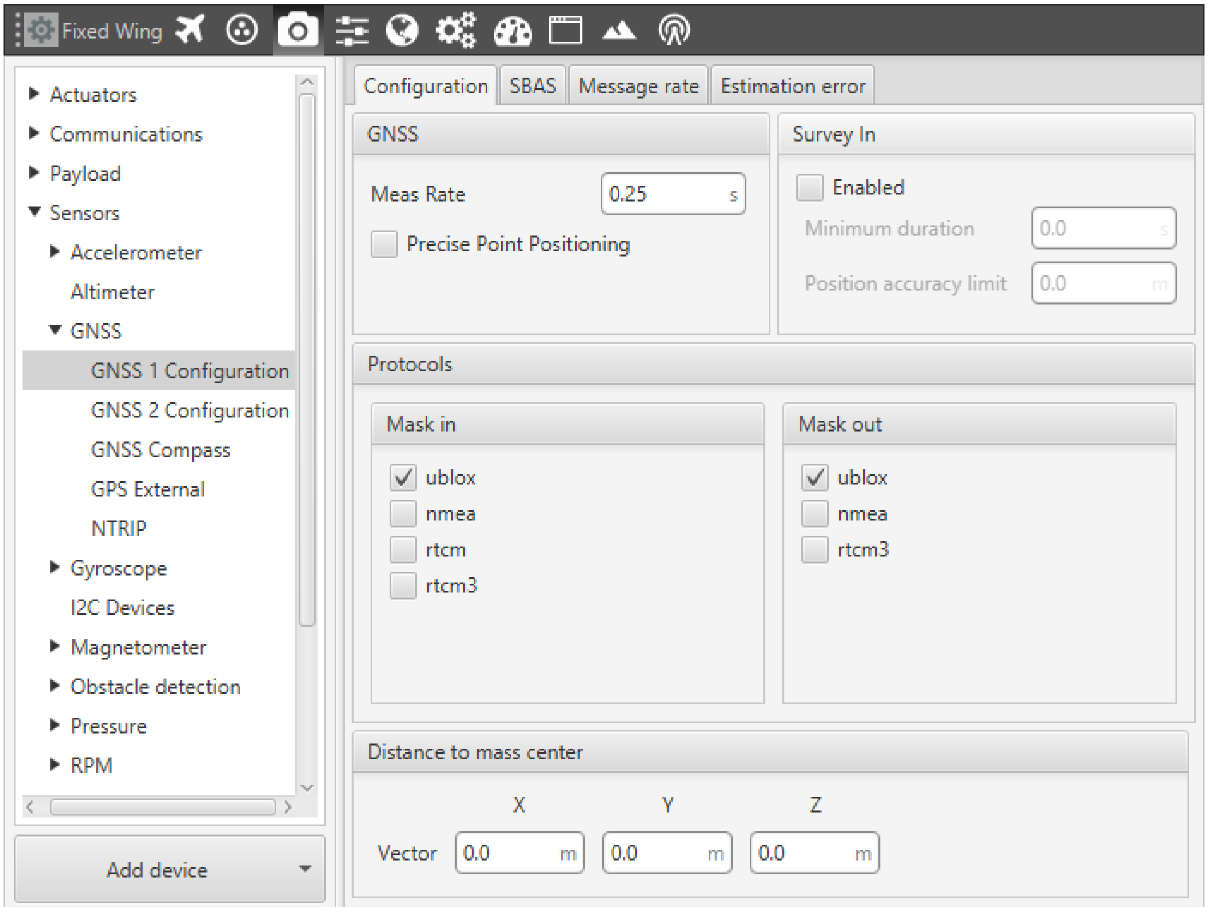
For configuring a Lidar, go to Examples - *Lidar Integration*.

6.2.3.4.3 GNSS

6.2.3.4.3.1 GNSS 1 & 2 Configuration

6.2.3.4.3.2 Configuration

This menu contains the parameters needed to configure the GNSS receiver 1-2 located in Veronte Autopilot.



GNSS - Configuration Menu

The following parameters are configurable in this menu:

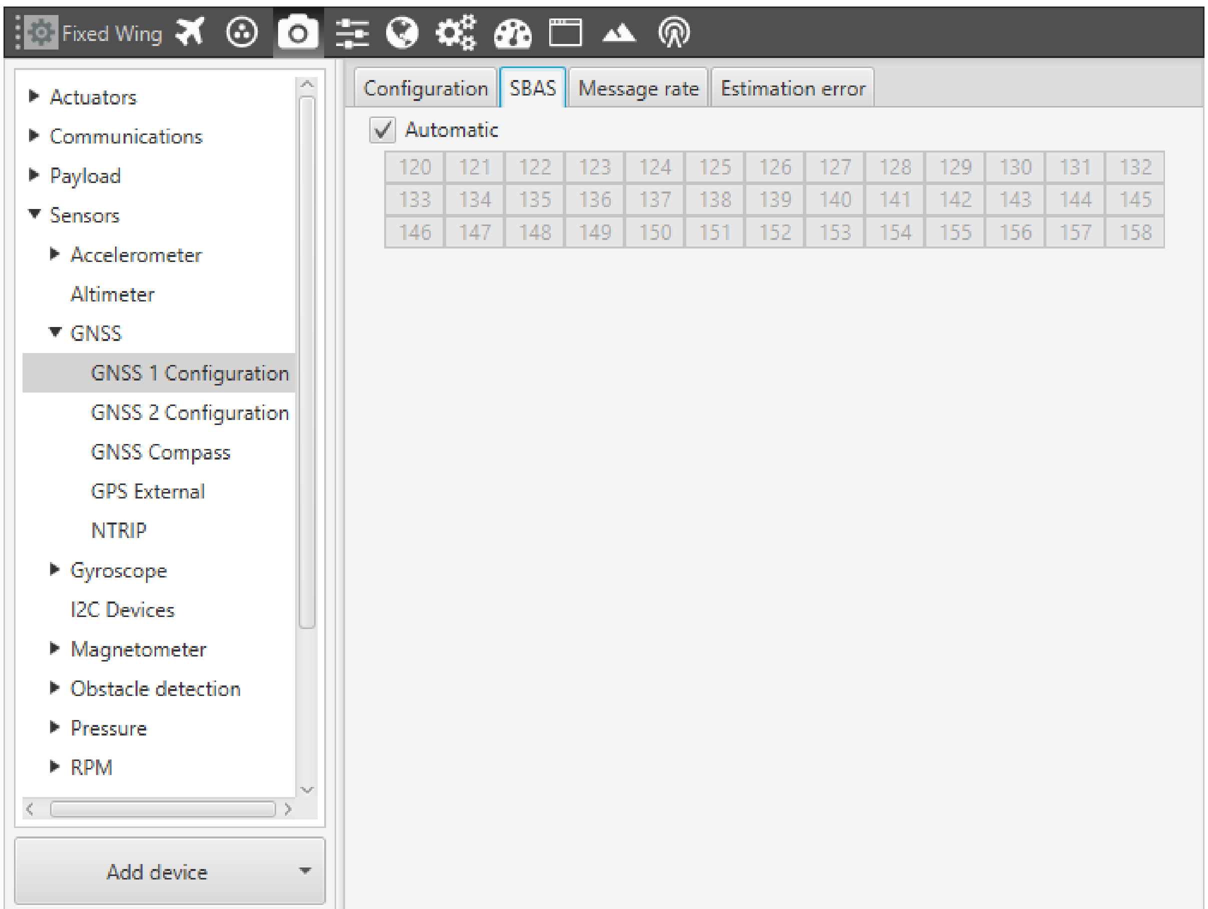
- **GNSS**. Data values that can be configured.
  - **Meas Rate**. Defines the minimum time between data acquisition.
  - **Precise Point Positioning (PPP)**. This option is a global precise positioning service, PPP is able to provide position solutions at centimetre to decimetre level after a few minutes with unobstructed sky view.
- **Survey in**. Determines a stationary receiver's position by building a weighted mean of all valid 3D position solutions. This mode should be activated in a Veronte Ground to enable GNSS Differential mode and send corrections to Veronte Air. Two requirements for stopping the procedure must be specified. Survey in procedure will end when both requirements are met.
  - **Minimum duration**. Defines a minimum amount of observation time regardless of the actual number of valid fixes that were used for the position calculation. Reasonable values range from one day for high accuracy requirements to a few minutes for coarse position determination
  - **Position accuracy limit**. Defines a limit on the spread of positions that contribute to the calculated mean.
- **Protocols**. Allows the user to select the different comunication protocols as input or output. One port can handle several protocols at the same time (e.g. NMEA and UBX).
  - Protocols will be changed when using the GNSS Compass Wizard (see in Devices - GNSS - GNSS Compass).

- **Distance to center of mass**. It is used to set the relative distance to the center of mass from the GNSS antenna in aircraft body axis. This parameter has to be set correctly in order to get a correct value when using GNSS Compass.

More information about protocols and configuration can be found here .

6.2.3.4.3.3 SBAS

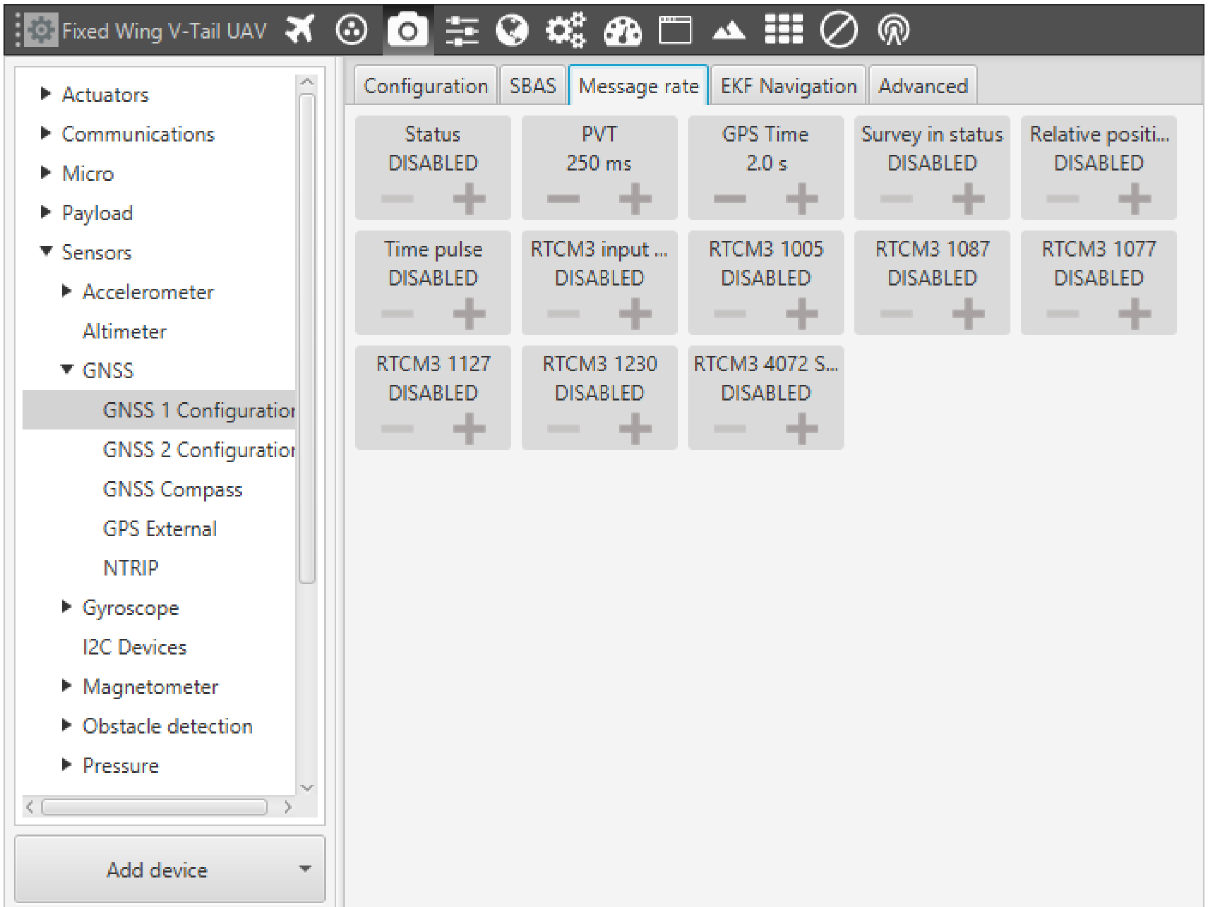
SBAS stands for Satellite Based Augmentation System. It is a set of geostationary satellites that are used to check the status of the signals sent by the GPS Satellites, and to improve the tracking via the correction of atmospheric disturbations, orbit deviations, clock errors and so on. In Veronte Pipe, it is possible to select the satellites that will be used for this purpose (selecting the number that appears in the table of the following figure), or make the software to automatically choose them according to the location of the platform.



GNSS - SBAS Menu

6.2.3.4.3.4 Message Rate

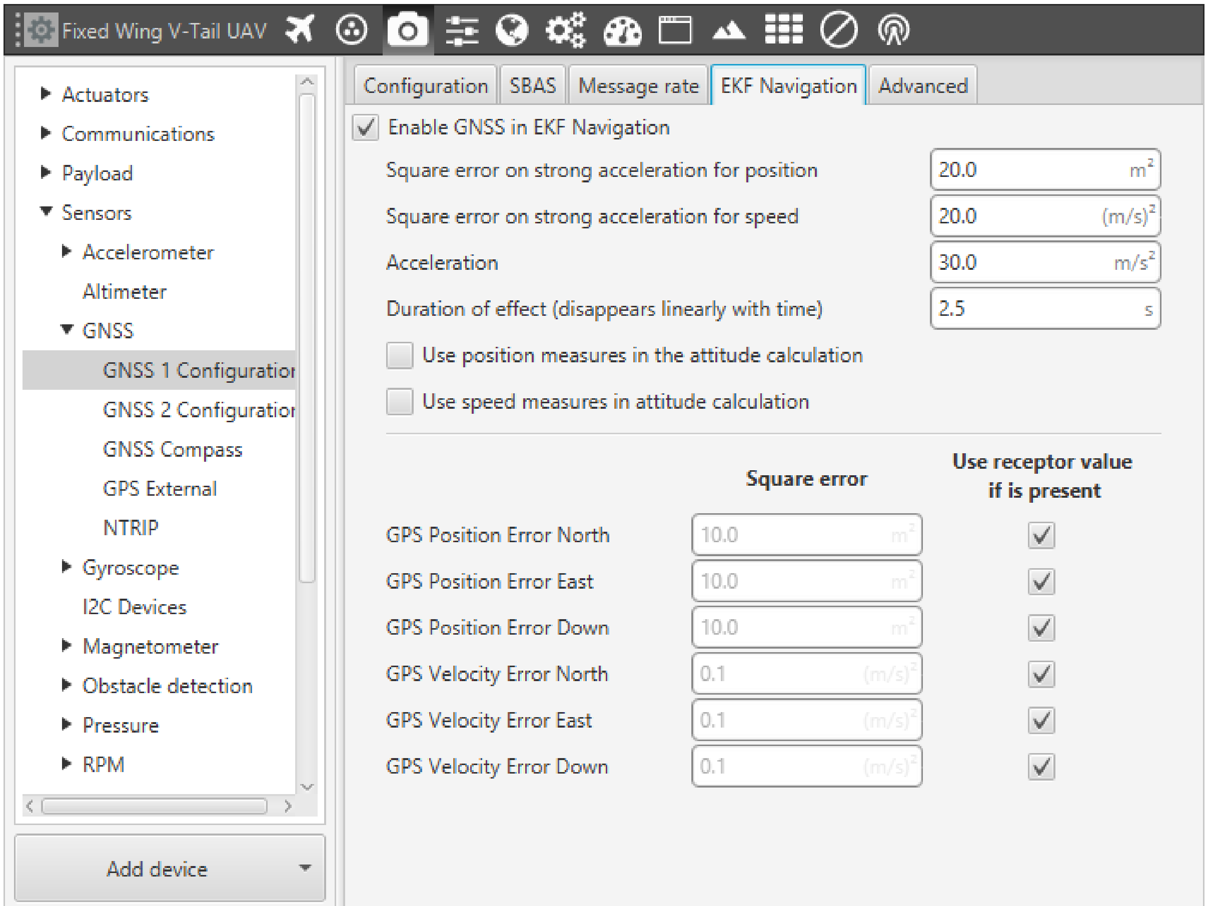
The **Message rate** options are used to set the time between the messages received at the autopilot. Each one of the different messages can be configured separately: ECEF (earth centred, earth fixed reference system), LLH (latitude, longitude and height), Speed, GPS Time, SV Status (state of the GPS satellite) and so on. This messages will be changed when using the GNSS Compass Wizard (see in Devices - GNSS - GNSS Compass).



GNSS - Message Rate

For information about RTCM 3 message list can be found here .

6.2.3.4.3.5 Estimation Error



GNSS - Estimation Error

- **Enable check**: Enables/Disables the usage GNSS solution in the Extended Kalman Filter. Veronte will keep receiving GNSS information from that module, but it will not be considered.
- **Use position measures in the attitude calculation**: When enabled, the position data from the GNSS solution is considered for the attitude estimation.
- **Use speed measures in the attitude calculation**: When enabled, the speed data from the GNSS solution is considered for the attitude estimation.

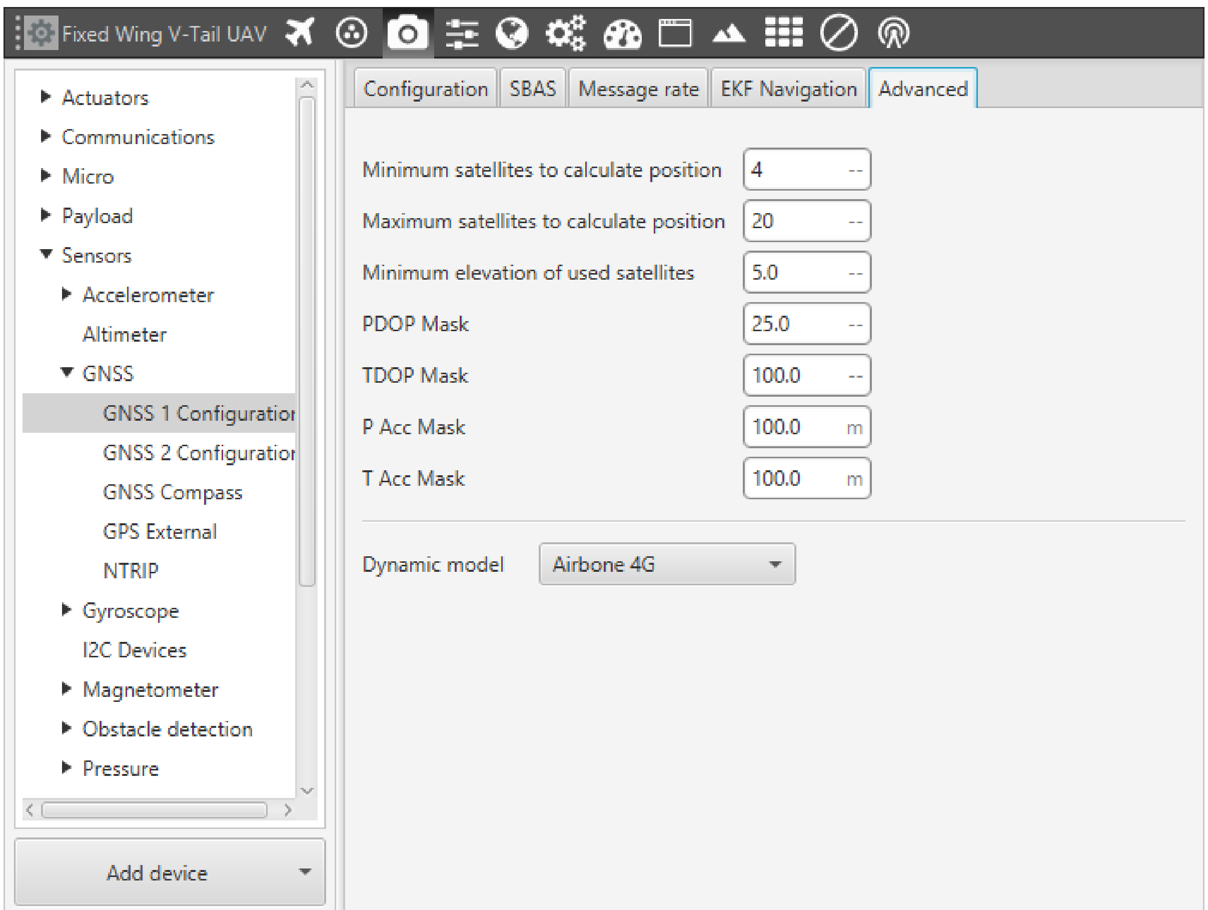
The variances considered in the EKF for the GNSS solution are by default the values provided by the GNSS receiver but can be modified for more complex scenarios.

- **GNSS North Position**: Variance for the North component of the position solution.
- **GNSS East Position**: Variance for the East component of the position solution.
- **GNSS Down Position**: Variance for the Down component of the position solution.
- **GNSS North Velocity**: Variance for the North component of the velocity solution.
- **GNSS East Velocity**: Variance for the East component of the velocity solution.
- **GNSS Down Velocity**: Variance for the Down component of the velocity solution.

Under strong accelerations the information provided by the GNSS does not represent the fast change of position and velocity therefore different variance must be considedred in the EKF.

- **Square error on strong acceleration for position**: Under a strong acceleration the variance for the position solution it is changed to the specified value.
- **Square error on strong acceleration for speed**: Under a strong acceleration the variance for the speed solution it is changed to the specified value.
- **Acceleration**: Threshold definition. When this threshold is exceeded, strong acceleration variances are considered.
- **Duration of effect**: Time needed to restore the default variances of the GNSS solution .

6.2.3.4.3.6 Advanced



GNSS - Estimation Error

> **Warning:** Modifying these parameters can cause problems during the acquisition of GNSS positioning.

- **Minimum satellites number**: Minimum number of satellites needed to have position fixed.
- **Maximum satellites number**: Maximum number of satellites needed to have position fixed

- **Minimum satellite elevation**: Minimum elevation of a satellite to be considered. Value in degrees.
- **PDOP mask**: Maximum Position dilution of precision to consider the solution.
- **TDOP mask**: Maximum Time dilution of precision to consider the solution.
- **P Acc mask**: Maximum Position accuracy to consider the solution.
- **T Acc mask**: Maximum Time accuracy to consider the solution.

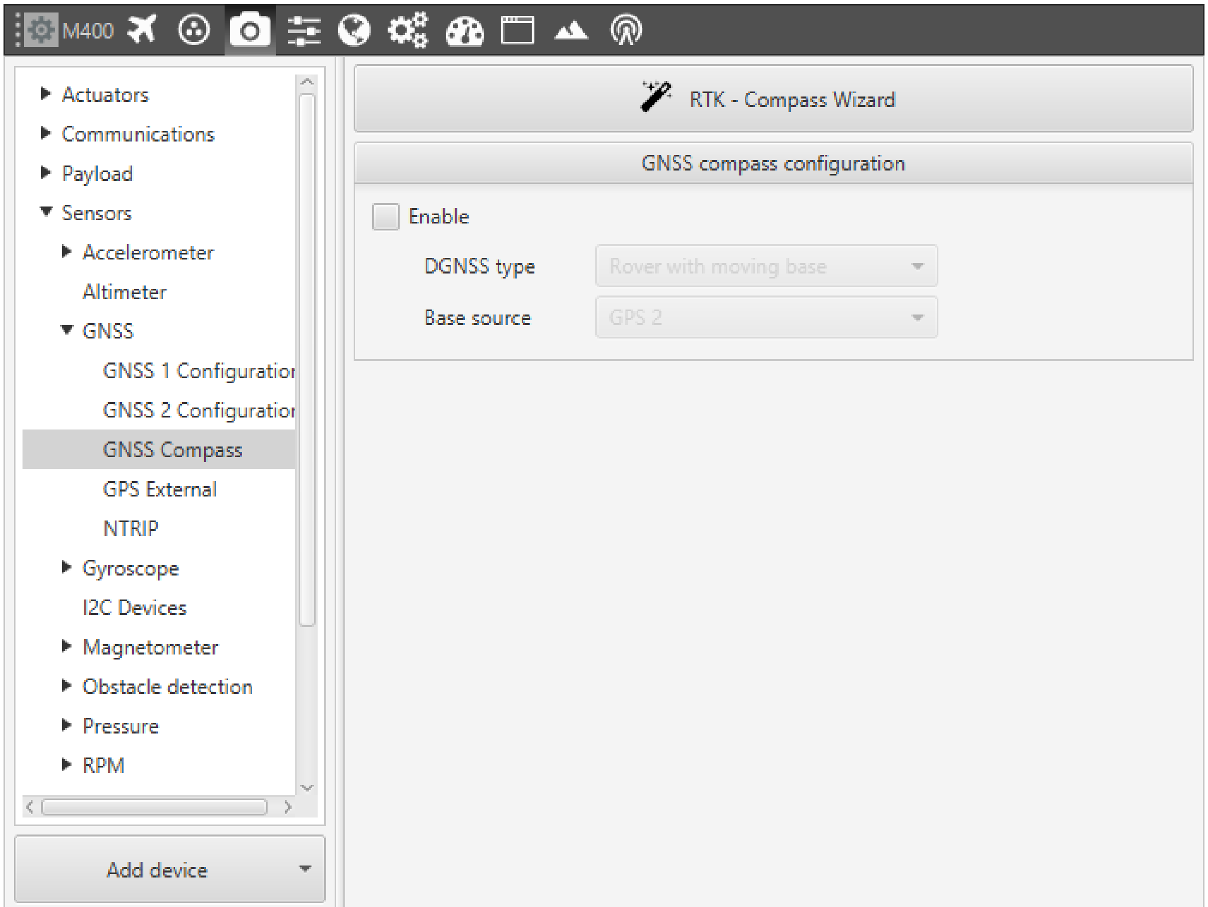
Dynamic Model

The embedded receiver supports different dynamic platform models to adjust the GNSS navigation engine to the expected application environment. The settings improve the receiver's interpretation of the measurements and thus provide a more accurate position output. Setting the receiver to an unsuitable platform model for the given application environment is likely to result in a loss of receiver performance and position accuracy.

Platform	Description
Portable	Applications with low acceleration.
Stationary	Stationary applications. Velocity restricted to 0 m/s. Zero dynamics assumed
Pedestrian	Applications with low acceleration and speed. Low acceleration assumed.
Automotive	Used for applications with equivalent dynamics to those of a car. Low vertical acceleration assumed.
Sea	Recommended for applications at sea, with zero vertical velocity. Zero vertical velocity assumed. Sea level assumed.
Airborne 1G	Used for applications with a higher dynamic range and greater vertical acceleration than a car.
Airborne 2G	Recommended for typical airborne environments.
Airborne 4G	Recommended for extremely dynamic environments.

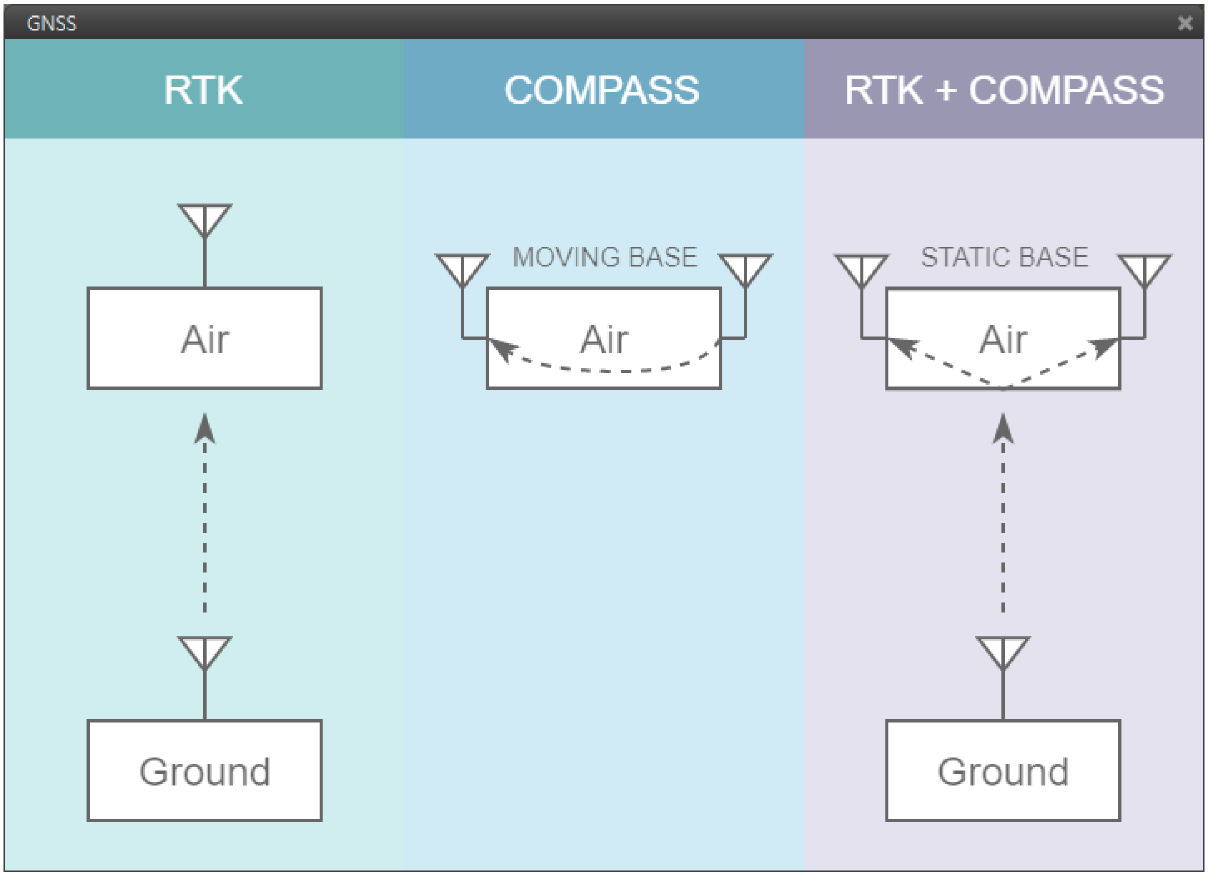
6.2.3.4.3.7 GNSS Compass

GNSS Compass is a menu where the user has access to the RTK - Compass Wizard (an interface which helps the user configuring everything related to RTK - GNSS Compass) and some other parameters which could be changed afterwards or if the user has knowledge enough.



GNSS Compass - Configuration Menu

By clicking on RTK - Compass Wizard button, the user will access the configuration menu where three options will be displayed.

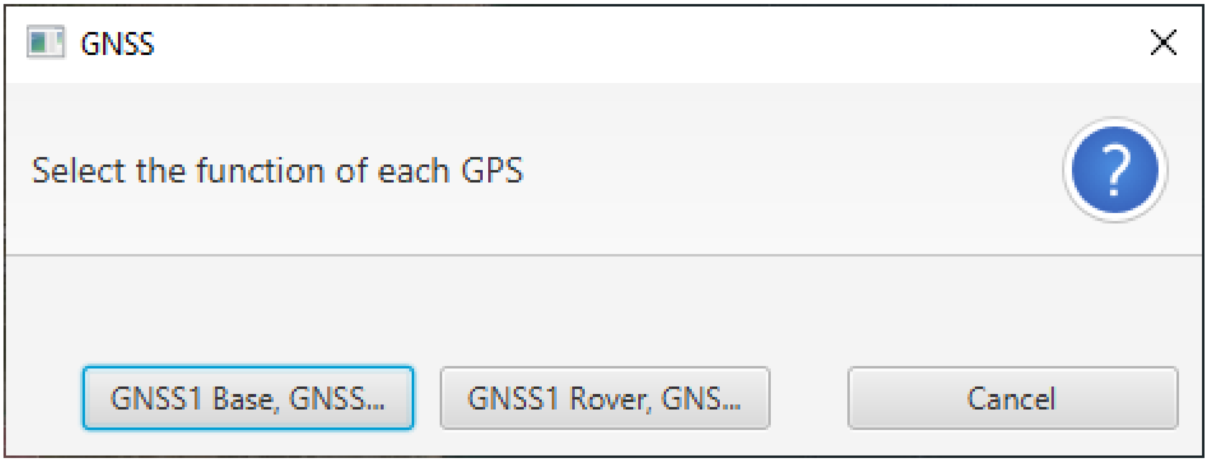


GNSS Compass - RTK & Compass Wizard

- **RTK**. Stands for Real Time Kinematics and it is a satellite navigation technique used to **enhance the precision** of position data derived from satellite-based positioning systems. It requires from **2 GNSS receivers** placed **in different autopilots** to work.
- **Compass**. The GNSS compass provides accurate dual antenna **GNSS based heading** that is not subject to magnetic interference. It requires from **2 GNSS receivers** placed **in the same autopilot** to work.
- **RTK + Compass**. A hybrid combination where both tool are employed at the same time in a system where the **AIR unit** must have **2 GNSS receivers** and the **GND**, at least, **must have 1**.

When accessing the GNSS Compass feature, the user will be asked which configuration is preferred in accordance to the function of each receiver:

- GNSS 1 Base and GNSS 2 Rover.
- GNSS 1 Rover and GNSS 2 Base.

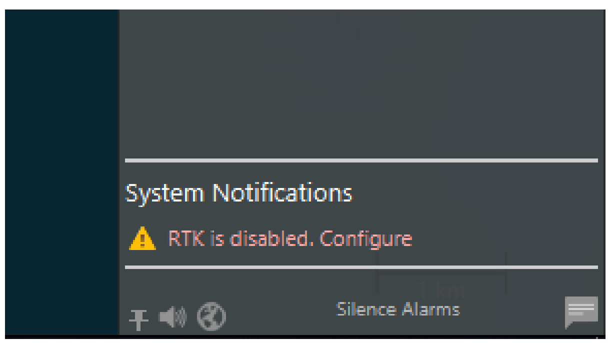


GNSS Compass - Compass Base/Rover

These same parameters can be manually changed afterwars in GNSS Compass Configuration. After using the wizard, the Enable check will be marked. The options enabled this way are:

- **DGNSS Type**. Where the option available are "Rover with moving base" and "Rover with rover (static base)". Consider the parameter established will be the one coming from the Wizard configuration.
- **Base source**. Where option are GPS 1 or 2.

The RTK - Compass Wizard does also appear as a **red notification** on the bottom right side of Veronte Pipe.



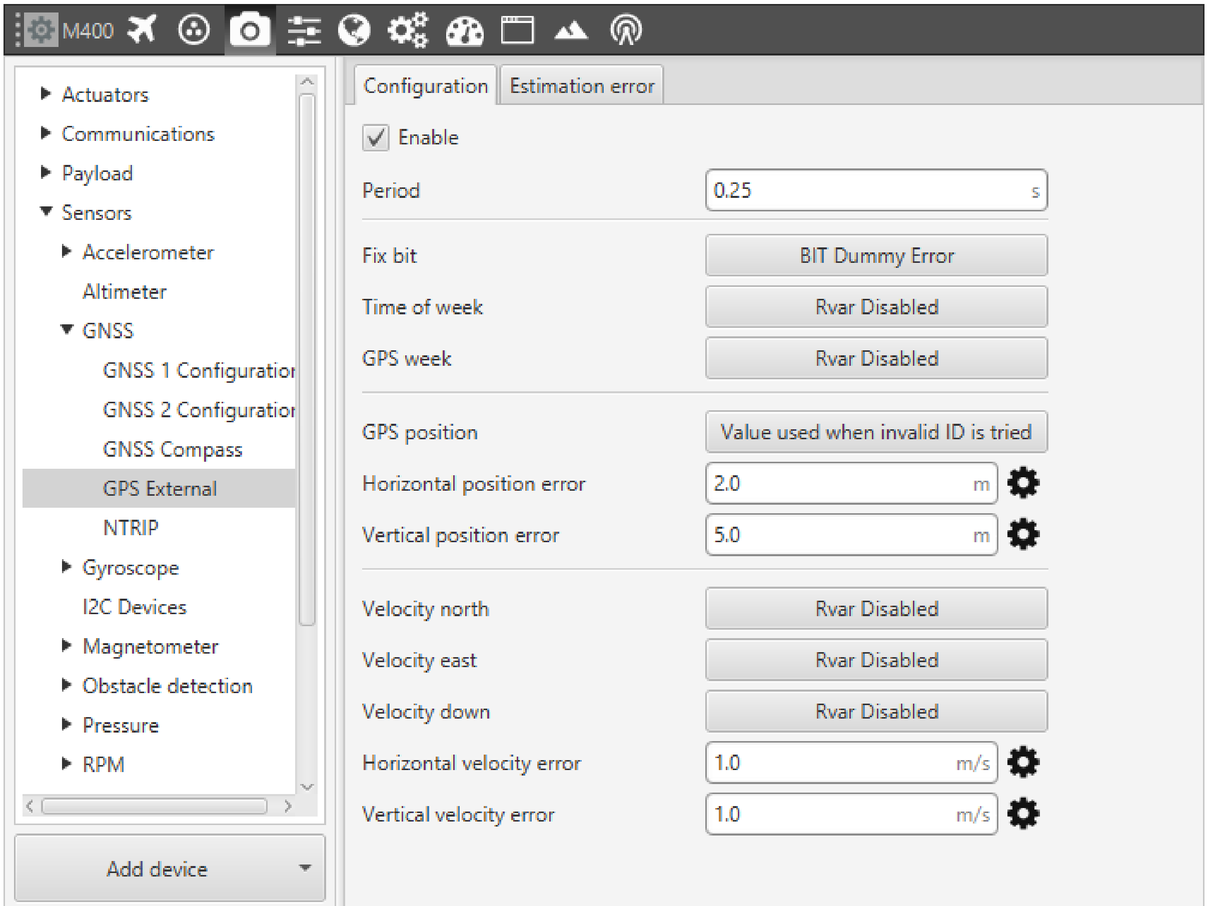
GNSS Compass - Configuration Notification

By clicking on it, the Wizard will ask the user which AP is the AIR unit and which one is the GND (in case RTK functionality is expected) and afterwards, the GNSS Compass wizard will be displayed and use in the same way.

6.2.3.4.3.8 GNSS External

If the reception of GNSS information is done through an external system, the user must configure it in this menu, so that system can be included in the navigation filters. After properly configuring the communication protocol on the corresponding channel (RS 232, RS485, CAN…) the GNSS External variables of interest must be filled in this interface:

6.2.3.4.3.9 Configuration



GNSS External - Configuration Menu

Check GNSS External device communication protocol before filling this menu.

- **Period**. Defines the period of incoming information from the external system.
- **Fix bit**. Data provided by the external device which is important to know the status of the positioning.
- **Time of week**. Variable extracted from the communication protocol defining the time of the week.
- **GPS Week**. Variable extracted from the communication protocol defining the week.
- **GPS Position**. Variable defininf latitude, longitude and height from GNSS. Usually Moving Object variables are used in Pipe.
- **Horizontal/Vertical Position Error**. Defined by the GNSS External device provider.
- **Velocity north/east/down**. Variables extracted from the communication protocol defining GNSS velocity measured.
- **Horizontal/Vertical velocity error**. Defined by the GNSS External device provider.

6.2.3.4.3.10 Estimation Error

See section Setup - Devices - Sensors - GNSS - *GNSS 1 & 2 Configuration*.

6.2.3.4.3.11 NTRIP

RTCM functionality provides precise positioning trough NTRIP (Networked Transport of RTCM via Internet Protocol). This protocol is a standard for streaming differential GPS (DGPS) data over the Internet.

6.2.3.4.3.12 Registration

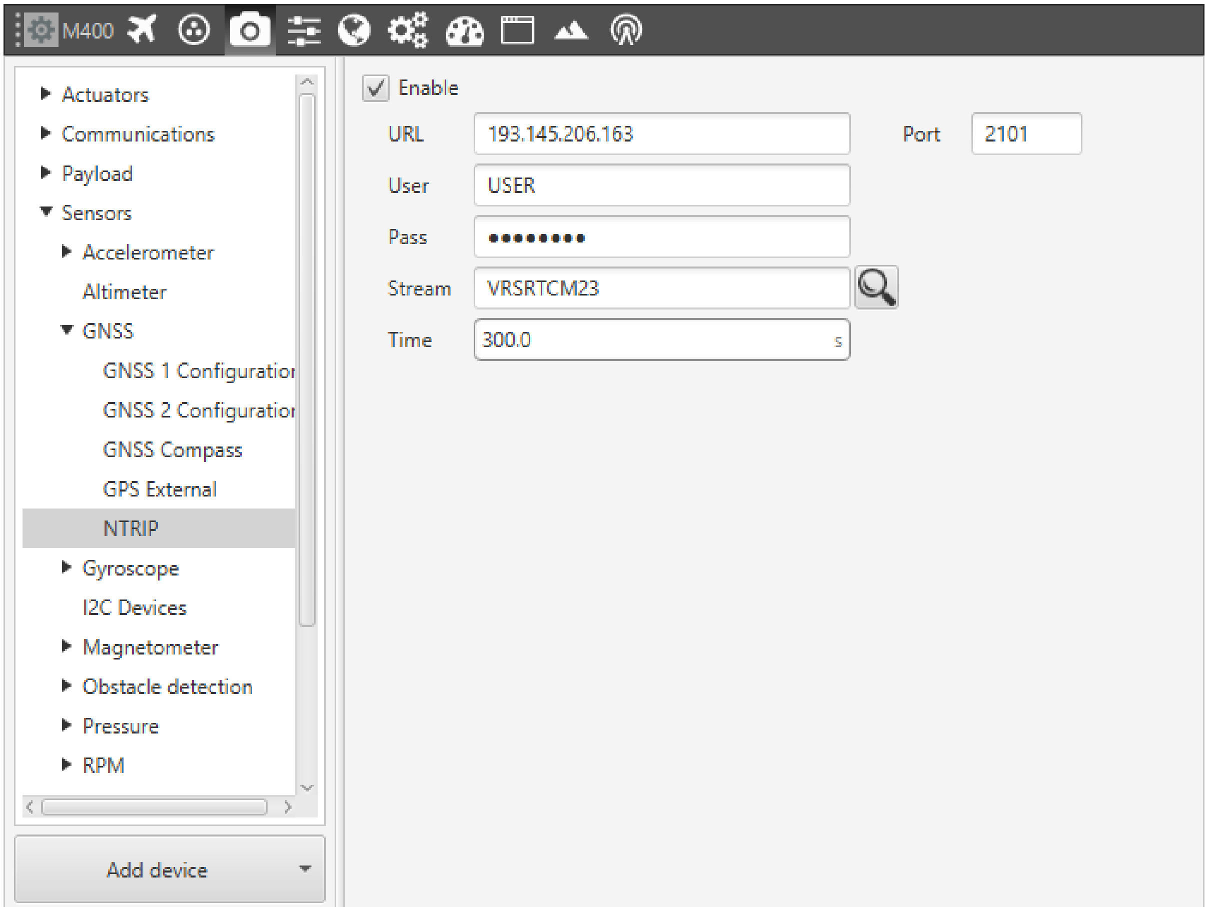
RTCM activation requires from a previous registration in one of the NTRIP stream providers available. Select the stream provider according to the location of your operation. Some examples of NTRIP providers are IGS.IP or EUREF.

Once registered, you will receive the following data from the NTRIP provider: URL, Port, User, Password

6.2.3.4.3.13 Configuration

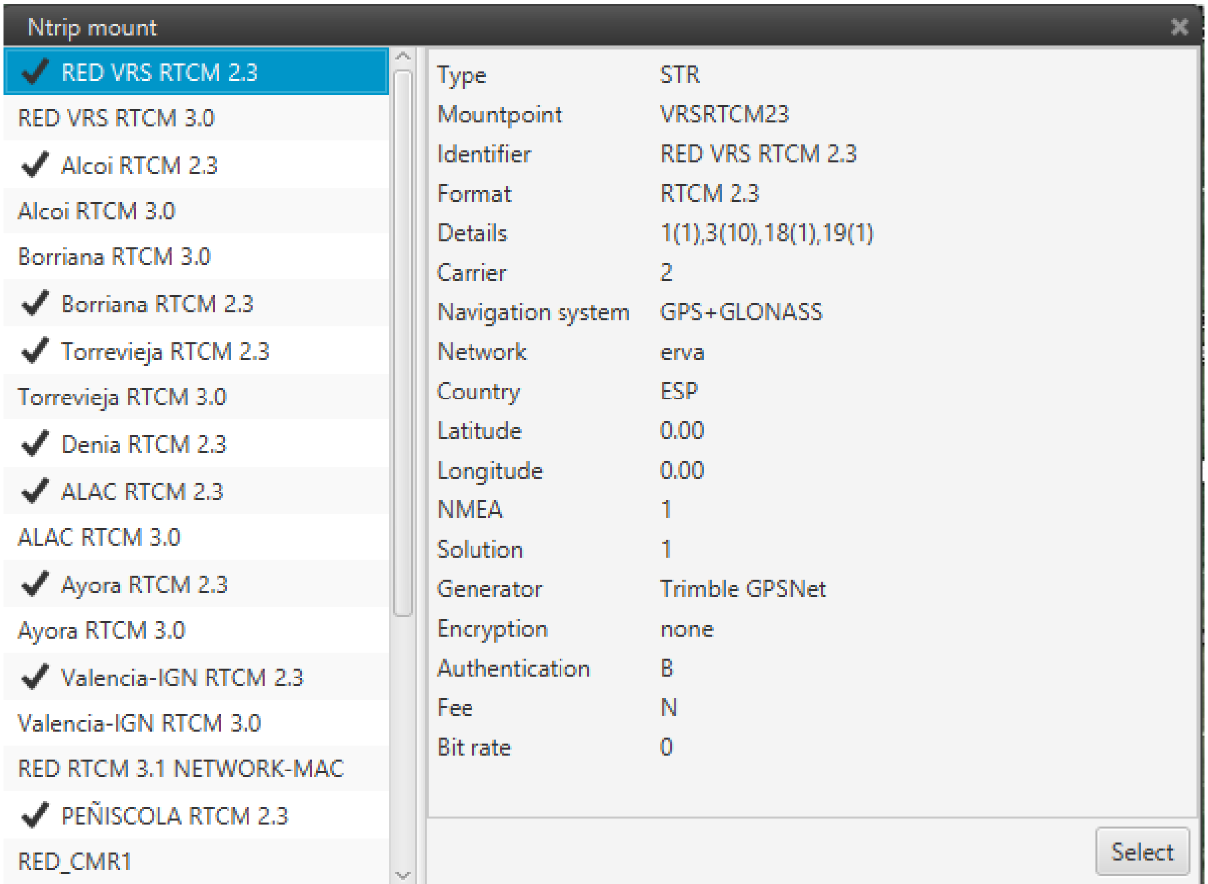
In order to activate precise positioning functionality in Veronte, NTRIP user data must be loaded in THE Veronte Autopilot. Go to the Setup menu in the Veronte Autopilot unit (Setup - Devices - Sensors - GNSS - NTRIP) and follow these steps:

1. Enable NTRIP
2. Enter NTRIP server data and user as follows



NTRIP - Configuration Menu

1. Click on the **Lens Icon** for connecting to the server
2. From the displayed list select the desired stream and click on select.

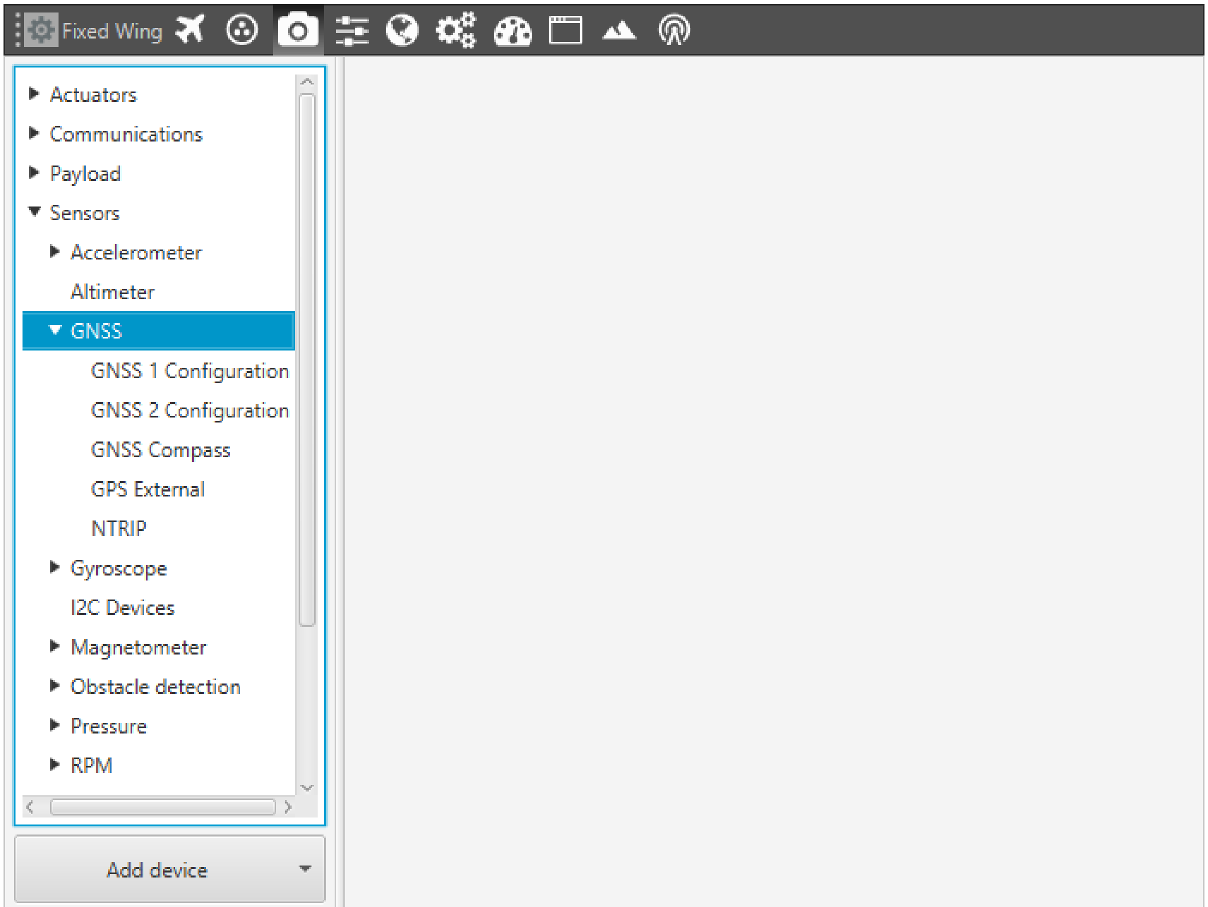


NTRIP - Stream List

Warning:

- The system is compatible with RTCM protocol 2.3. Compatible streams are marked with **Check**. It is recommended to select a virtual stream covering the area of operation.
- RTCM positioning requires an internet connection.

In this menu the user can modify all parameters which are relevant to GNSS sensors.



GNSS - Options available

Veronte has embedded 2 GNSS modules which are configured through GNSS 1-2 Configuration tabs. They can be configured as well for precision aoolications, such as RTK (high precision with correction from a base) or GNSS Compass (Compass information from differential positioning). External positioning such as GPS External or NTRIP have their own configuration tab as well.

6.2.3.4.4 Gyroscope

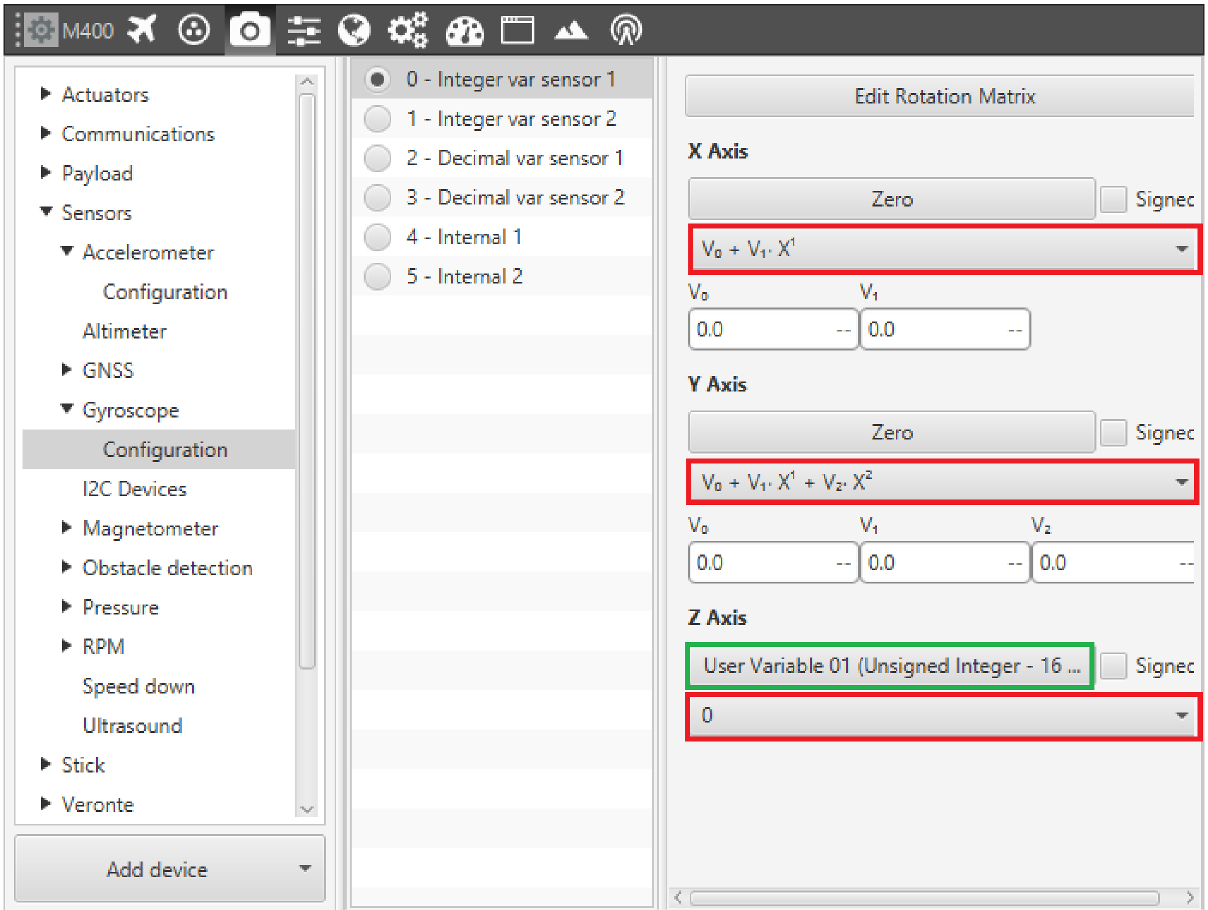
The gyroscope from the IMU can be configured as explained in the menus shown below.

The user can choose between 3 types of source for the gyroscope.

- **Integer var sensor 1-2**. Veronte uses a integer value provided by an external sensor.
- **Decimal var sensor 1-2**. Veronte uses a decimal value provided by an external sensor.
- **Internal 1-2**. Veronte uses the internal sensor.

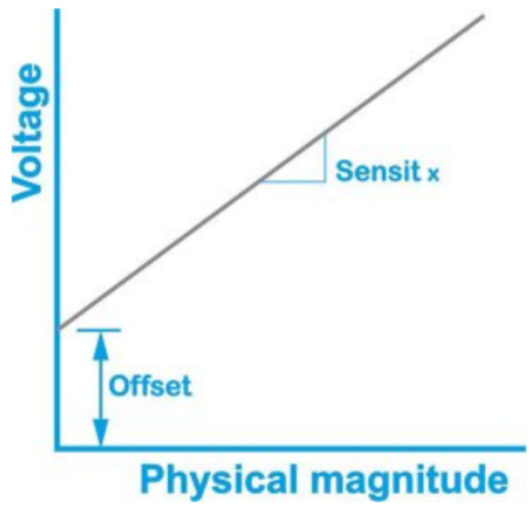
6.2.3.4.4.1 Integer var sensor

In this menu it is possible to configure integer variables provided by an external sensor.



Integer var Gyroscope Menu - Configuration Parameters

When Integer var sensor 1 or 2 are selected, the previous panel will be shown. In this panel, the user selects the variable that has been stored in a user variable (Green Box) and the operations that will be carried on (Red Box). It is possible to use the signal through a linear or quadratic relation. The following image shows an example of a linear relation.

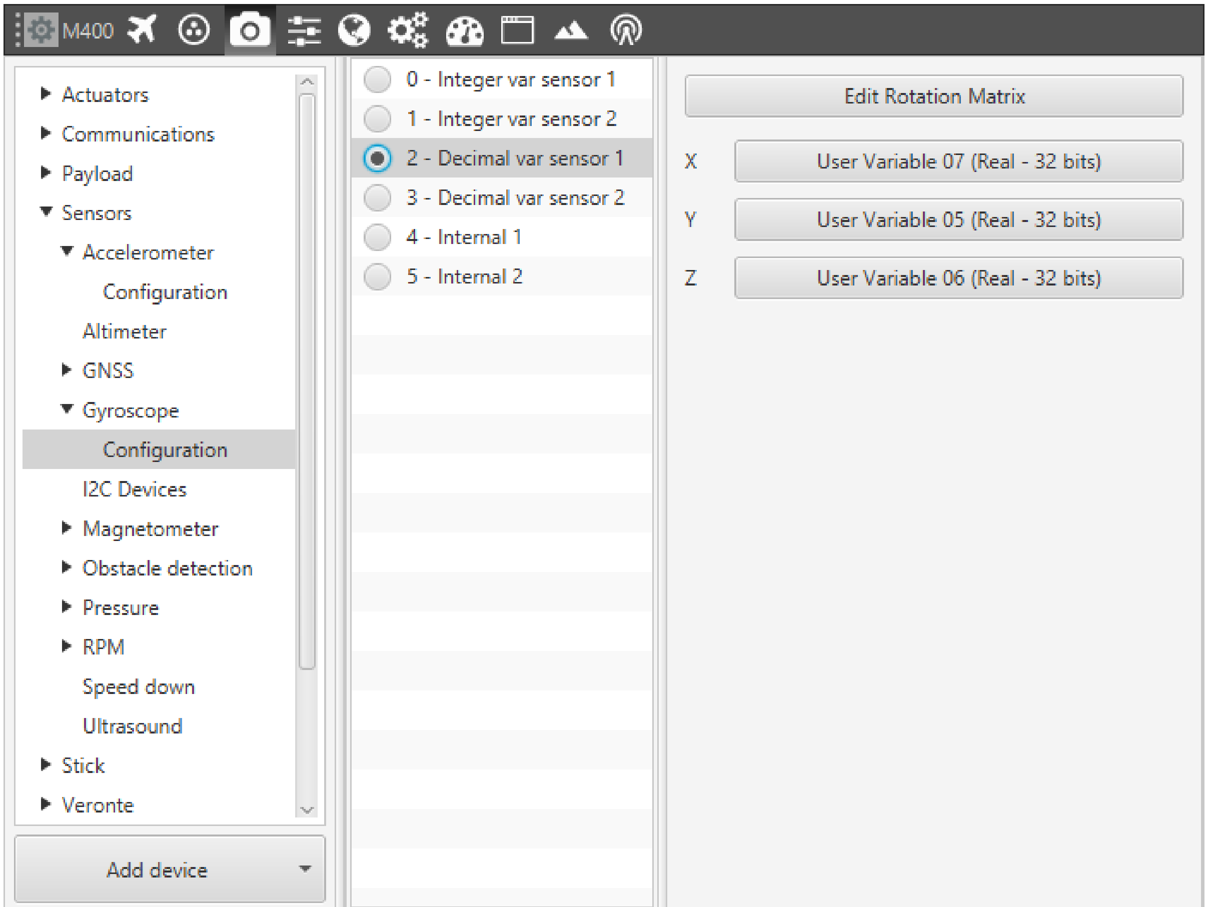


Linear relation of 2 Variables

The process of configuration has to be done using **Custom Messages**. This is to be configured in *Devices - Others - Digital - I/O Manager*. The configuration will depends on the device in use and its communication protocol.

6.2.3.4.4.2 Decimal var sensor

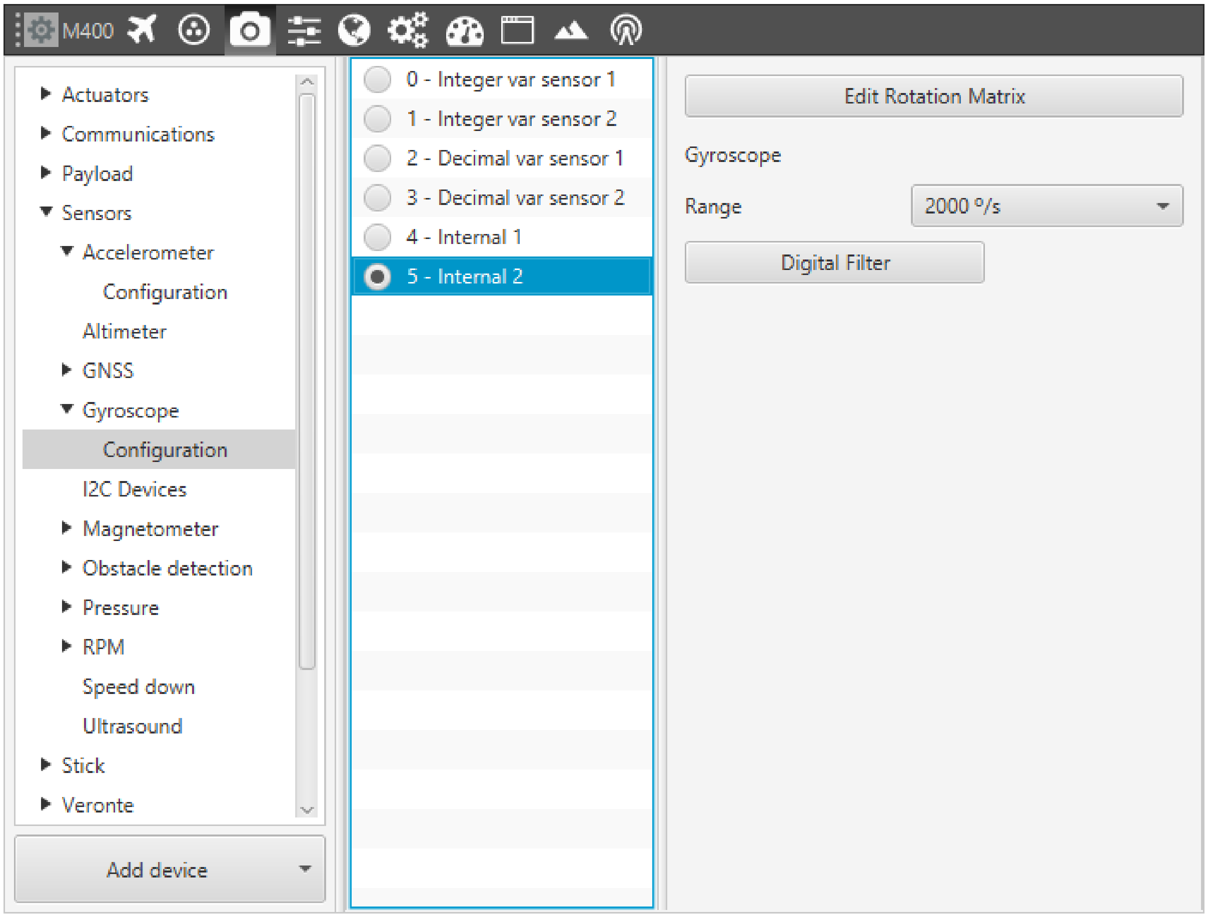
In this menu, the user selects real variables for each axis (X,Y,Z), these do not requiere a signal treatment. The process of configuration is similar to the one carried out when configuring a Integer Variable.



Decimal var Gyroscope Menu - Configuration Parameters

6.2.3.4.4.3 Internal

This last menu available displays the posible parameters that can be configured for the internal Gyroscope.



Gyroscope Menu - Configuration Parameters

> **Warning:** **Edit Rotation Matrix** brings the position of the gyroscope inside the Veronte Autopilot, it must **NOT** be changed under any circumstance.

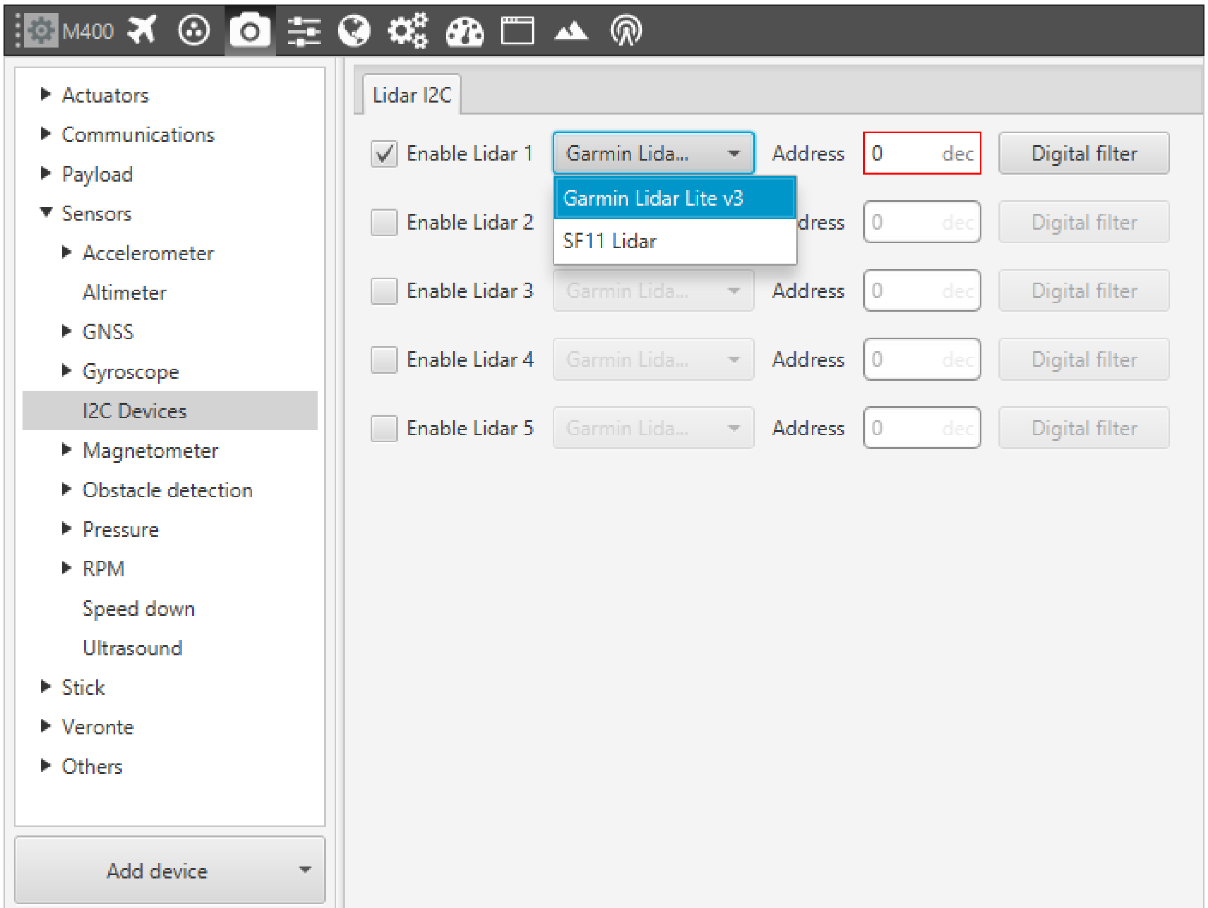
In this menu it is possible to set different options regarding range and filters from the gyroscope. The parameters that can be modified are:

- **Range**. Sets the maximum range of performance, high ranges implies less precision while small ranges might mean the system saturates. Values allowed are 125°/s, 250°/s, 500°/s, 1000°/s and 2000°/s.
- **Digital filter**. Enables a low pass filter which its cutoff frequency is configured manually, allowing the user to input any desired value in Hz. It is a **software filter**.

6.2.3.4.5 I2C Devices

The I2C bus allows the connection of several devices with different addresses to the same line. In our case, Veronte allows up to 5 Lidar devices to be connected to the system.

The configuration menu can be seen below:



I2C - Configuration Menu

After enabling the needed number of Lidar devices, configurable parameters are:

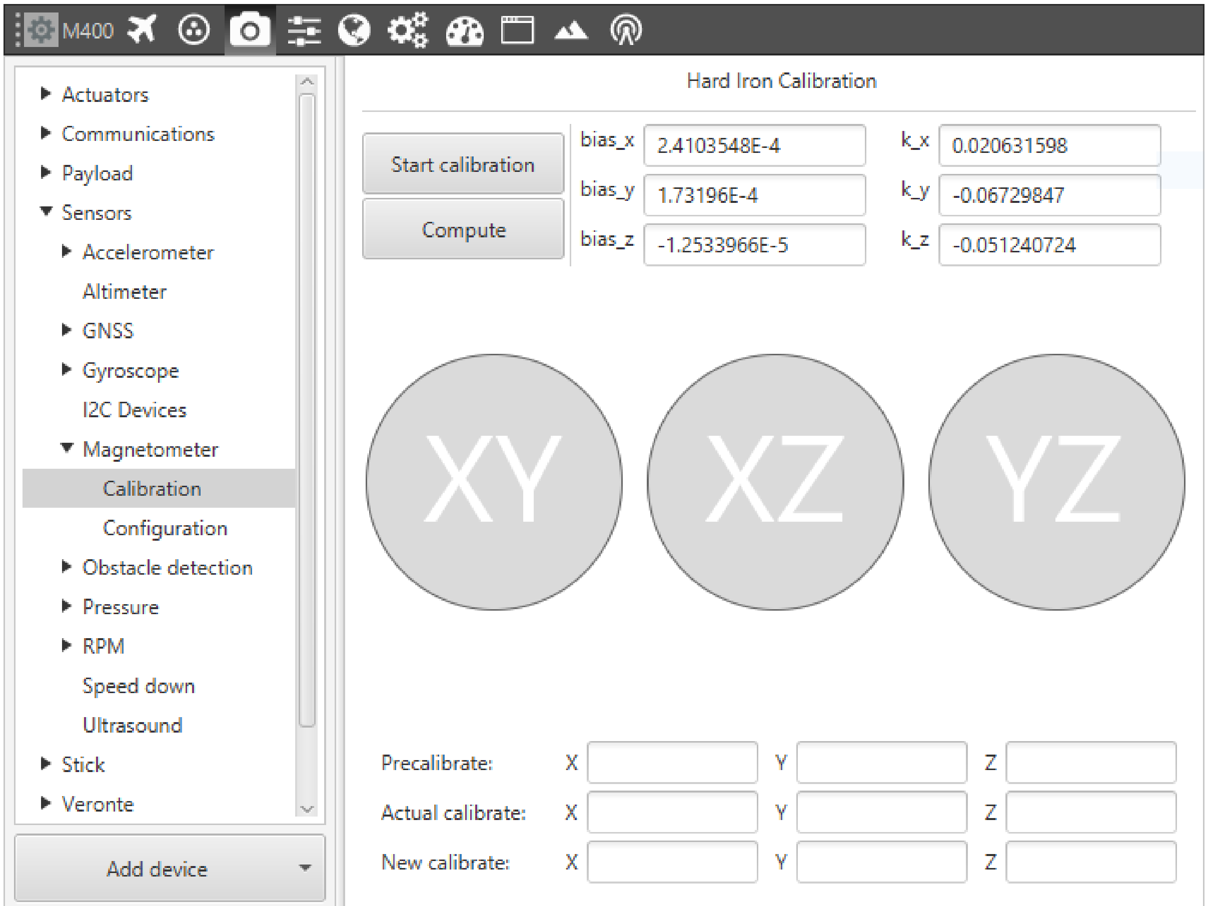
- **Type of Lidar**. Veronte is compatible with Garmin Lidar Lite v3 and SF11 Lidar.
- **Address**. With an accepted value between 16 - 239, this is the origin address from the lidar being configured.
- **Digital filter**. nables a low pass filter which its cutoff frequency is configured manually, allowing the user to input any desired value in Hz. It is a **software filter**.

The Lidar number need to be kept in order to properly configure the Altimeter menu afterwards in Setup - Devices - Sensors - *Altimeter*.

6.2.3.4.6 Magnetometer

6.2.3.4.6.1 Calibration

Magnetometer calibration should be performed once Veronte has been installed on the platform so the magnetic field during the operation is similar to the one measured during the calibration.

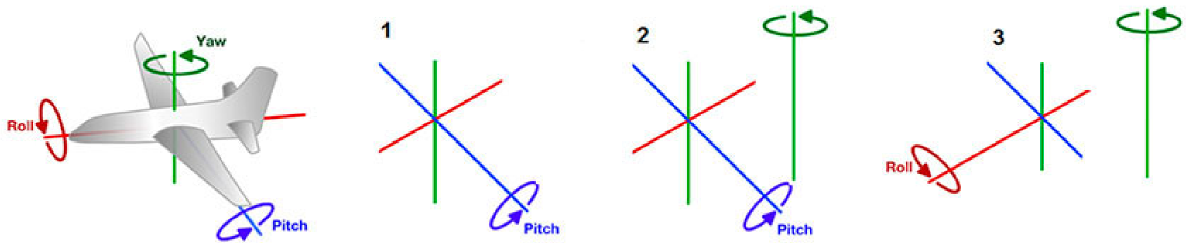


Magnetometer - Calibration Menu

In order to start calibration, press on the **Start Calibration** button so the system can capture magnetometer data. During the calibration, the system must be oriented in all possible directions so enough data can be captured. Once enough data has been captured, **Compute Data** sets the calibration.

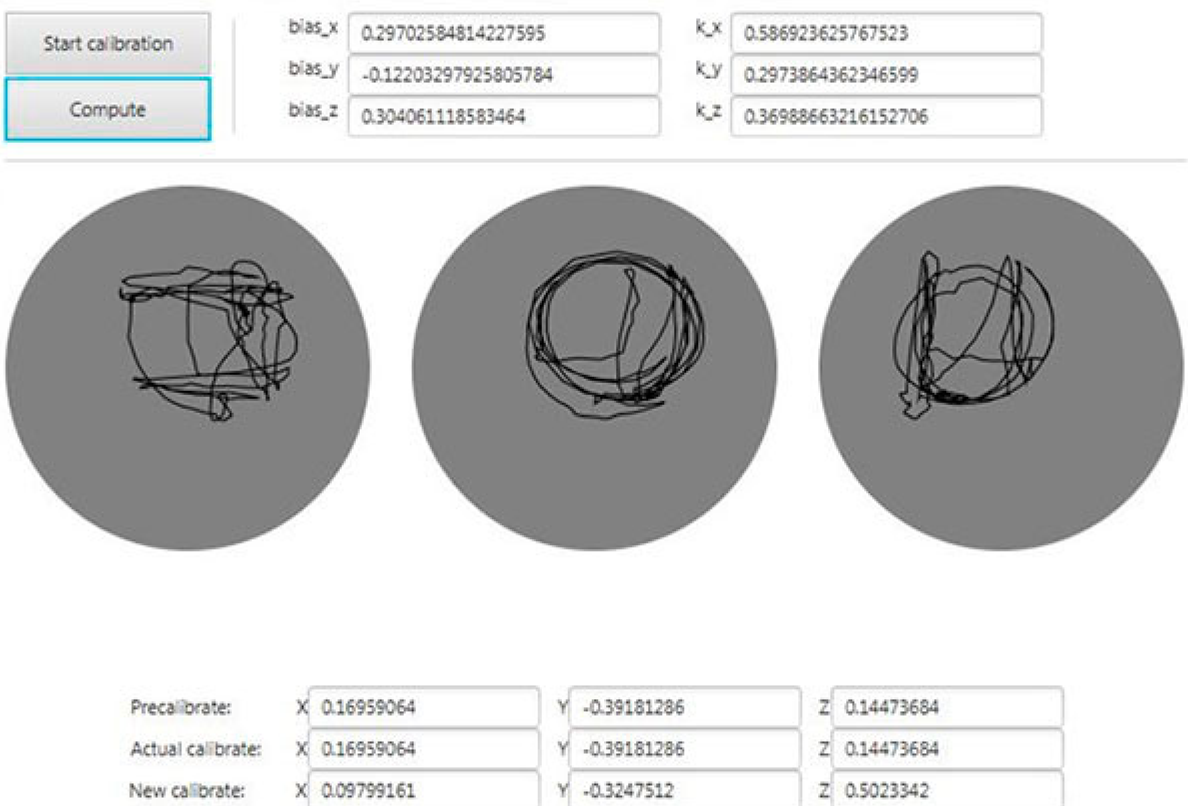
The procedure for acquiring enough data for performing the calibration is:

1. Hold the platform with your hands on the Y axis and rotate it parallel to the ground.
2. While the platform is rotating, rotate also yourself so the platform turns in two axes simultaneously.
3. Turn the platform 90 degrees within your hands and repeat the operation.



Magnetometer - Calibration Procedure

Once three circles have been drawn on the screen, captured data will be enough for saving the calibration data. The following image shows an example of the calibration result:



Magnetometer - Calibration Values

6.2.3.4.6.2 Configuration

Veronte incorporates an internal magnetometer that allows the Veronte System to measure the magnetic field.

The user can choose between 4 types of source for the accelerometer.

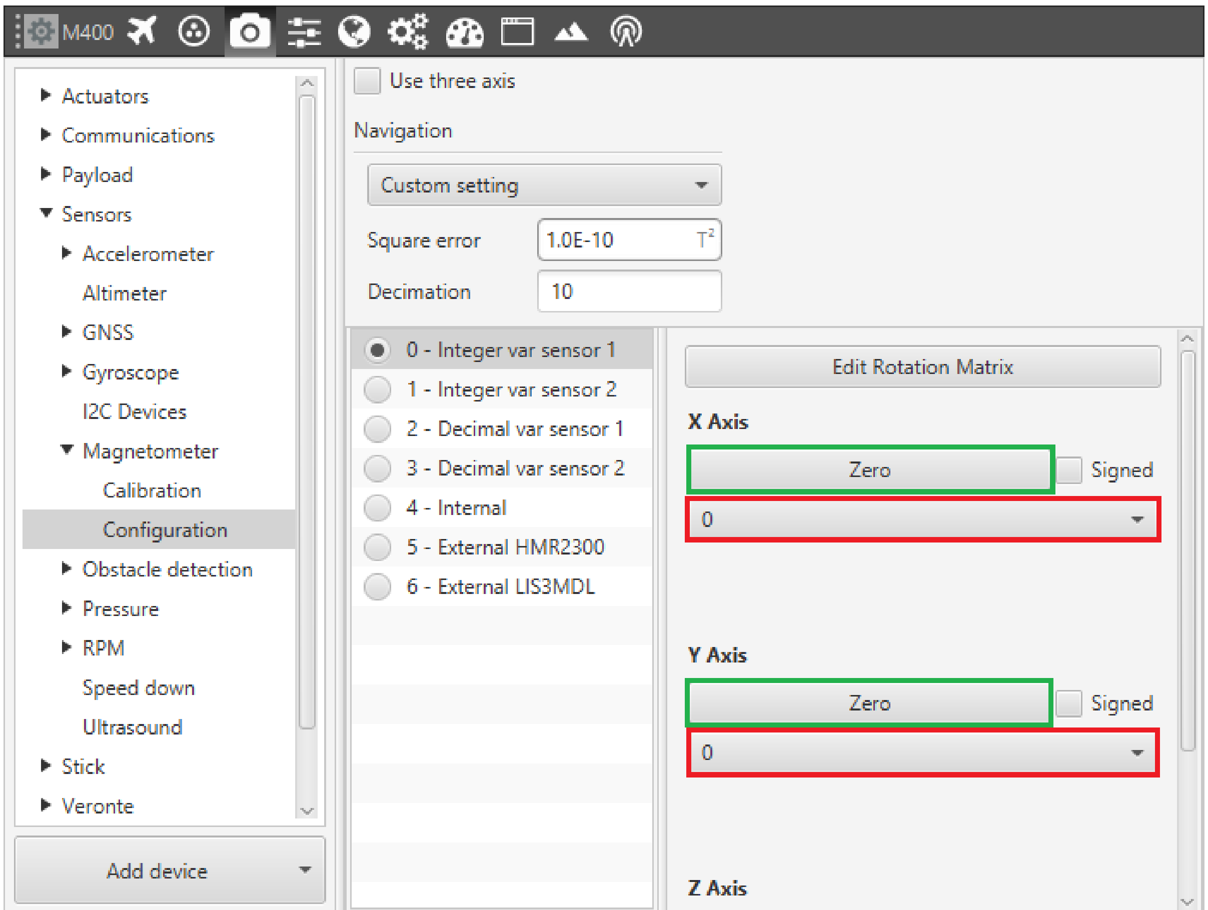
- **Integer var sensor 1-2**. Veronte uses a integer value provided by a no-integrated external sensor.
- **Decimal var sensor 1-2**. Veronte uses a decimal value provided by a no-integrated external sensor.
- **Internal** Veronte uses the internal sensor.

- **External HMR2300/LIS3MDL**. Veronte uses the information from one of the compatible external magnetometers.

There are 3 parameters that are configured independently from the Magnetometer selected. These are the one in the Navigation section: - **State**. User can choose from **Disabled** for magnetometer not entering the Navigation filters (not being used but not turning it off) or **Custom settings**, which will take into account all the following parameters. - **Square error**. Means the influence of the parameter on the Navigation filters. The greater the less effect it will have. - **Decimation**. Defines the bunch of data from which 1 value will be stored. For example, if decimation is 10, every 10 measurements 1 will be taken into account. This procedure is used to reduce the number of samples.

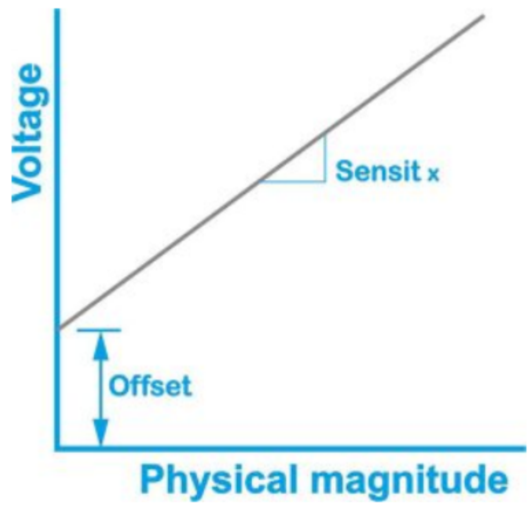
6.2.3.4.6.3 Integer var sensor

In this menu it is possible to configure integer variables provided by an external sensor.



Integer var Magnetometer Menu - Configuration Parameters

When Integer var sensor 1 or 2 are selected, the previous panel will be shown. In this panel, the user selects the variable that has been stored in a user variable (Green Box) and the operations that will be carried on (Red Box). It is possible to use the signal through a linear or quadratic relation. The following image shows an example of a linear relation.

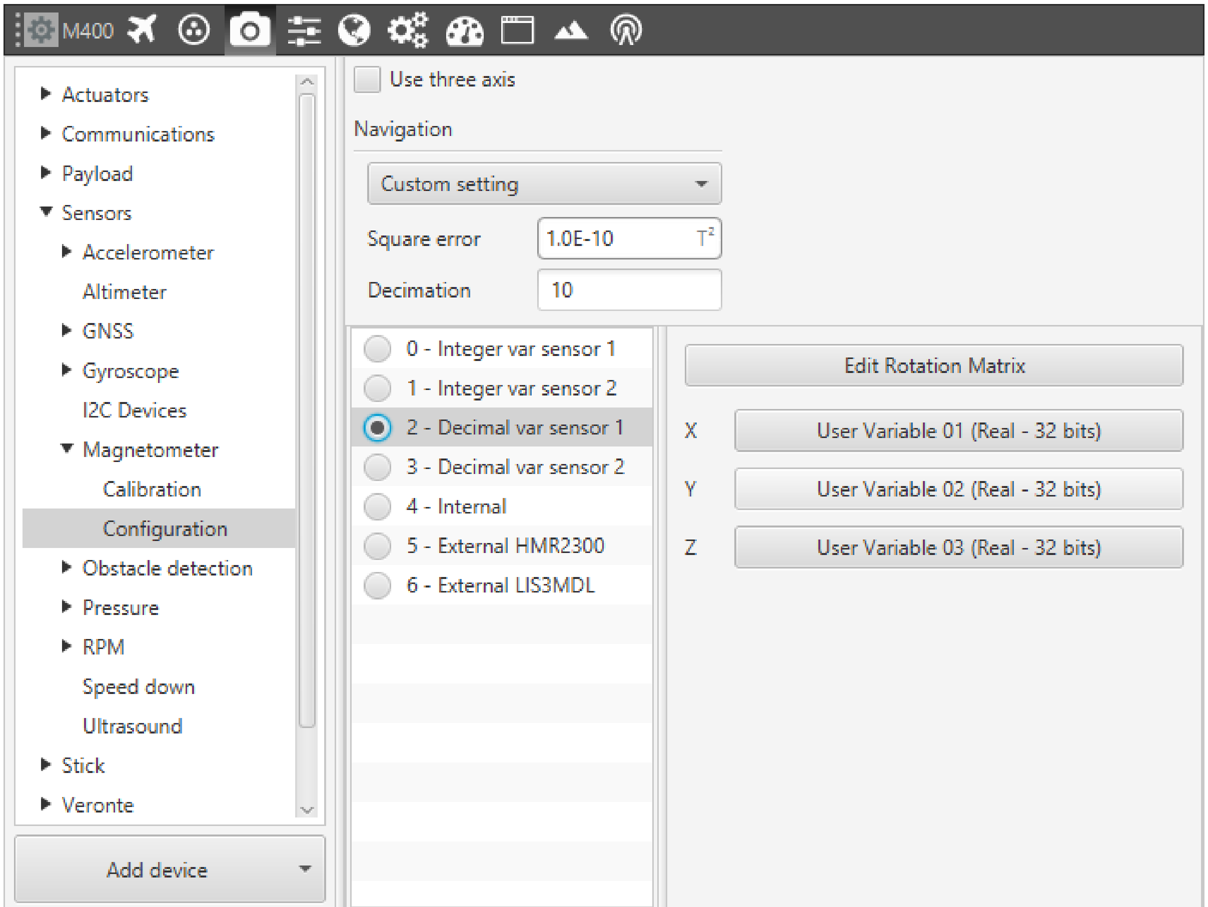


Linear relation of 2 Variables

The process of configuration has to be done using **Custom Messages**. This is to be configured in *Devices - Others - Digital - I/O Manager*. The configuration will depends on the device in use and its communication protocol.

6.2.3.4.6.4 Decimal var sensor

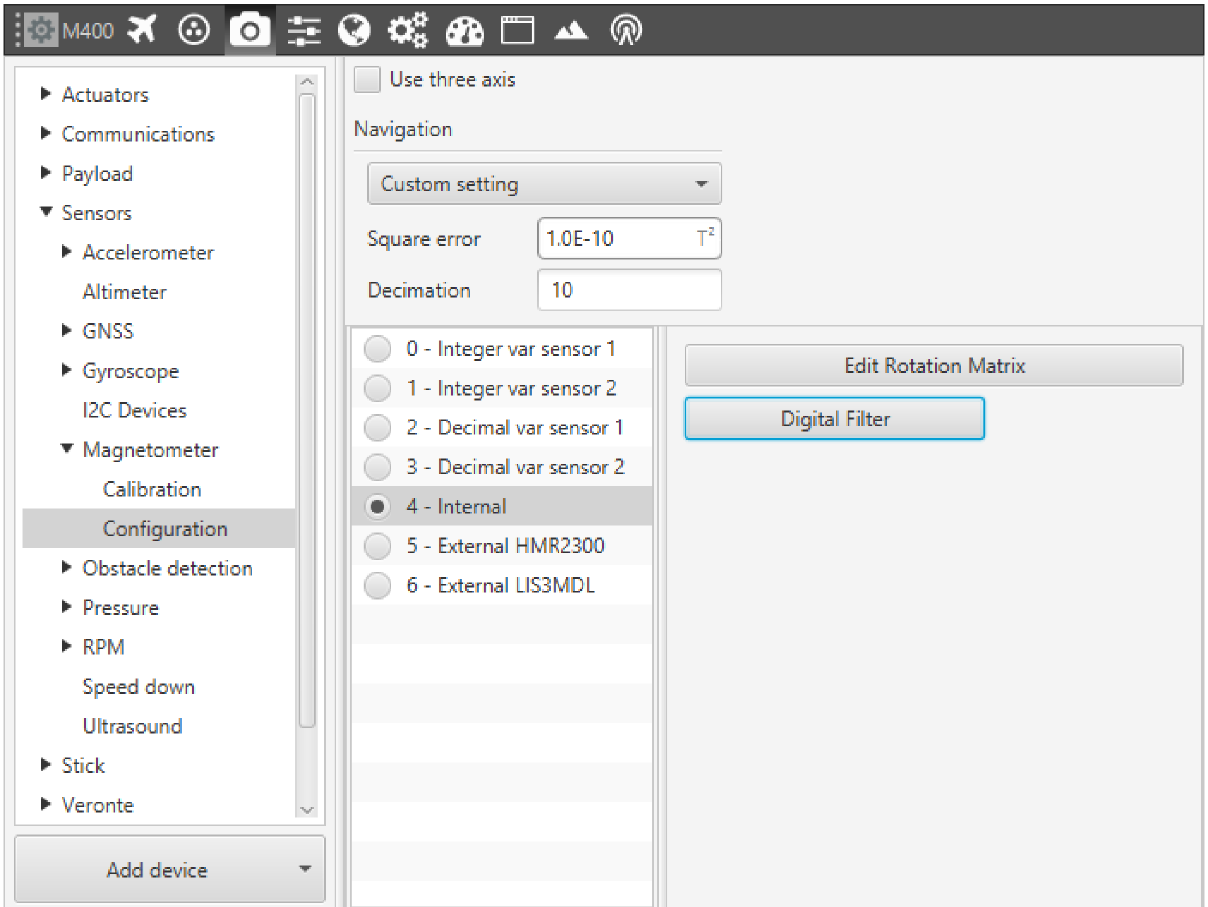
In this menu, the user selects real variables for each axis (X,Y,Z), these do not requiere a signal treatment. The process of configuration is similar to the one carried out when configuring a Integer Variable.



Decimal var Magnetometer Menu - Configuration Parameters

6.2.3.4.6.5 Internal

This menu available displays the posible parameters that can be configured for the internal Magnetometer.



Internal var Magnetometer Menu - Configuration Parameters

> **Warning:** **Edit Rotation Matrix** brings the position of the magnetometer inside the Veronte Autopilot, it must **NOT** be changed under any circumstance.

In this menu it is possible to set different options regarding filters.

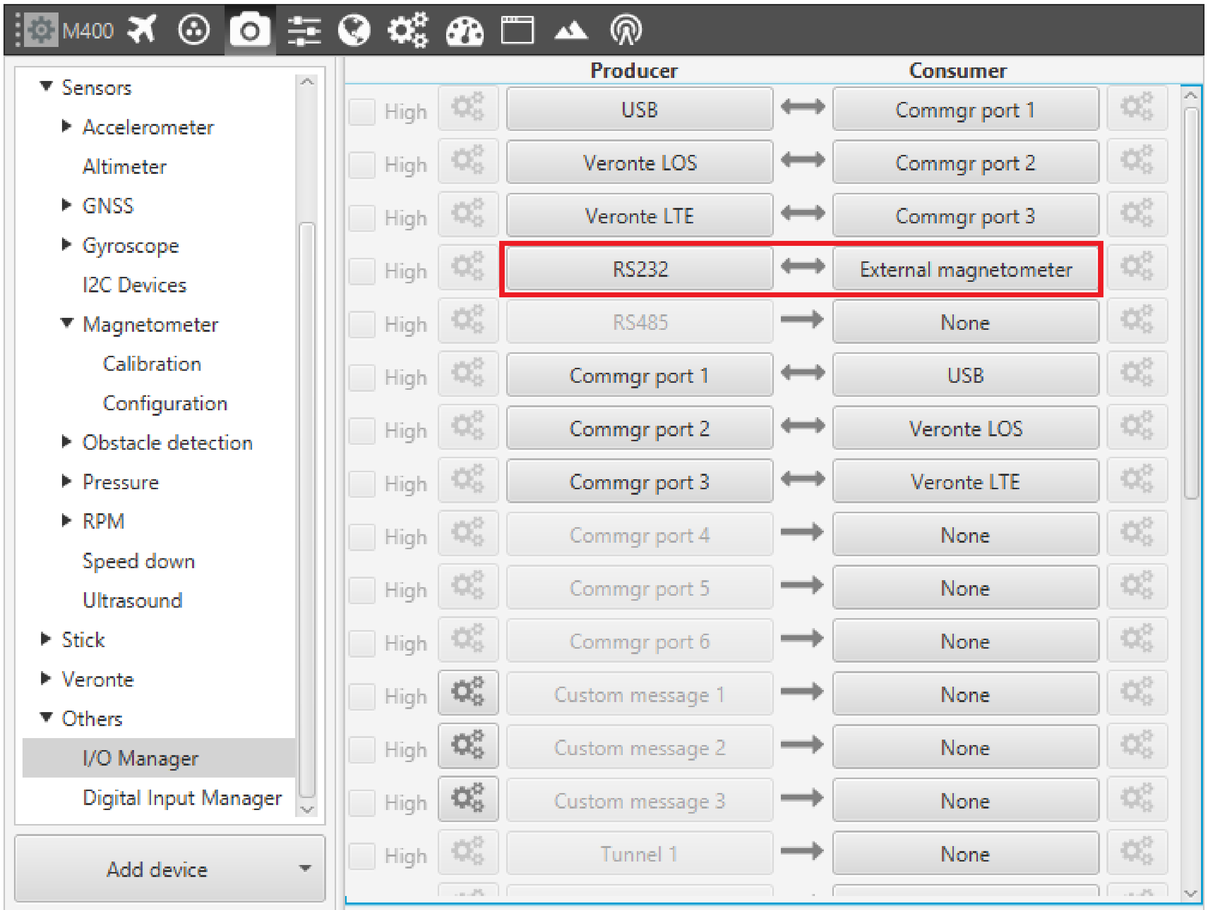
- **Digital filter**. Enables a low pass filter which its cutoff frequency is configured manually, allowing the user to input any desired value in Hz. It is a **software filter**.

6.2.3.4.6.6 External

Veronte has been designed to have compatibility with the external magnetometers HMR200 and LIS3MDL. The first one has no filters configurable, only the option Edit Rotation Matrix to set the orientation of it.

The LIS3MDL option is the sensor Veronte has inside it, but mounted externally to avoid the possible interferences from being close to ekectronic components. It has de Edit Rotation Matrix and the Digital Filter.

On the other hand, the connection to the serial port is configured in *Devices - Others - I/O Manager*



External Magnetometer - Channel Configuration

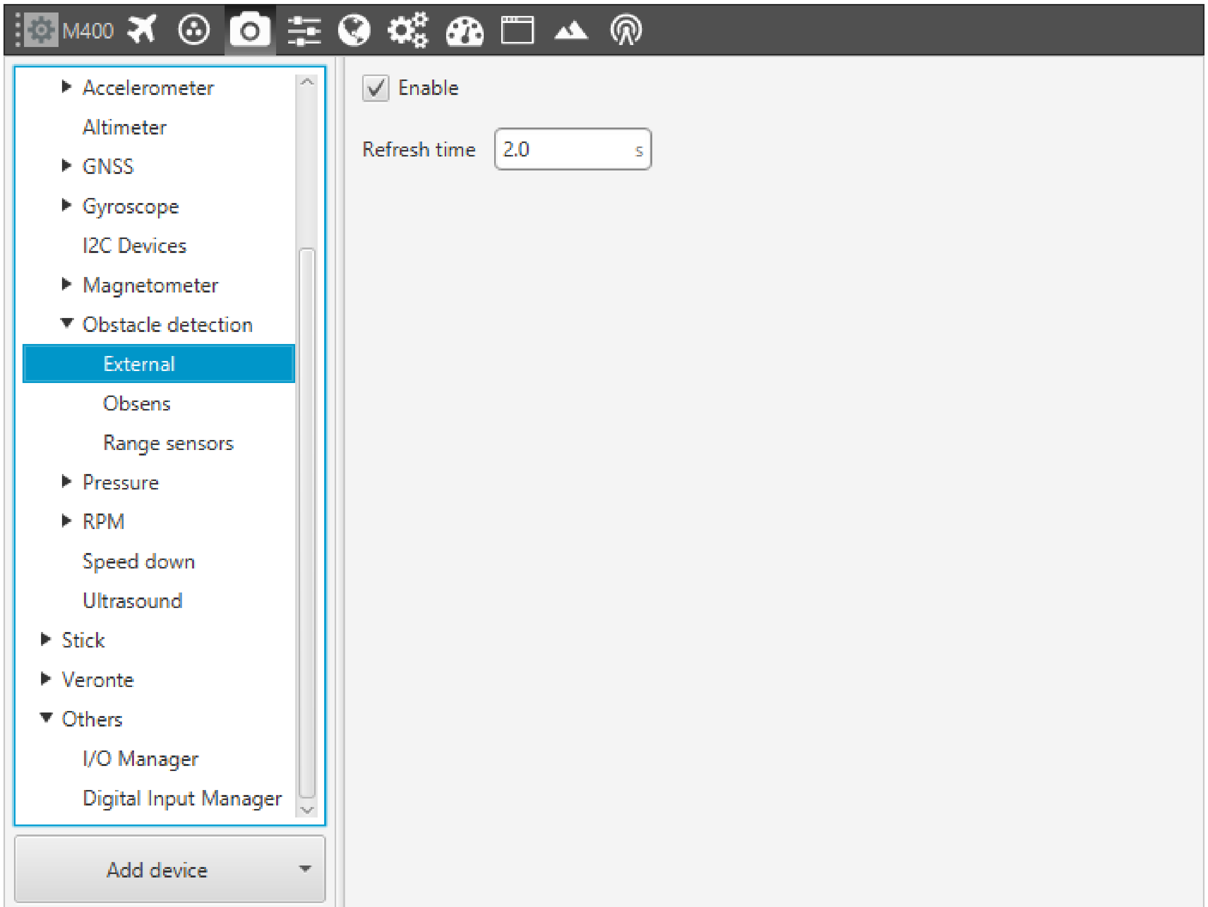
6.2.3.4.7 Obstacle Detection

Obstacle detection allows the creation of obstacles that the platform will consider for its Navigation. It could be an external source of obstacles or the sensors connected to Veronte with the selection of the sensors range.

6.2.3.4.7.1 External

If a user connects an Obstacle provider system to Veronte, it has to be enabled from this menu.

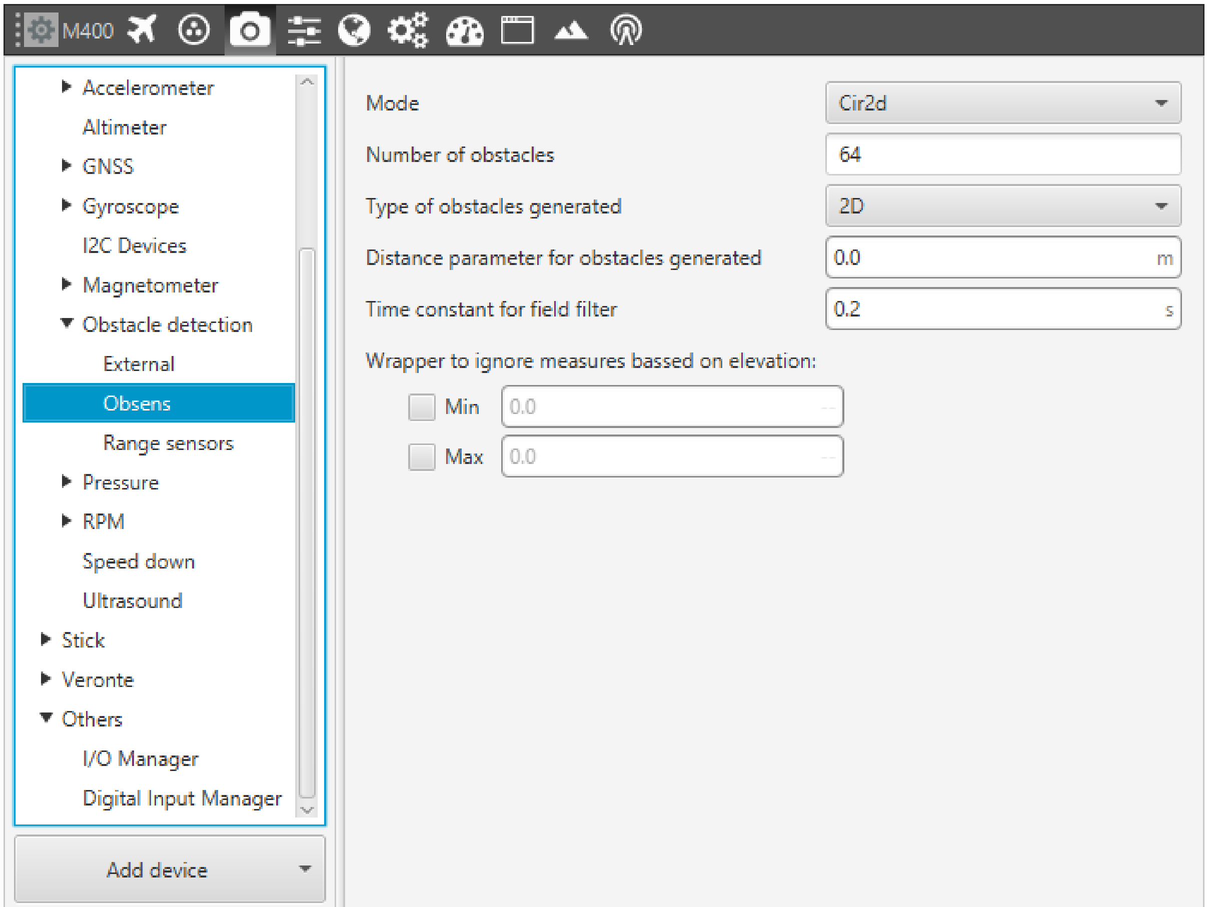
The **Refresh time** is the time it takes Veronte to check again the obstacle status from the external device. This is used to know when an obstacle has disappear from range or a new obstacle is inside it.



Obstacle Detection - External

6.2.3.4.7.2 Obsens

This panel contains the configuration menu to set the obstacles creation.



Obstacle Detection - Obsens

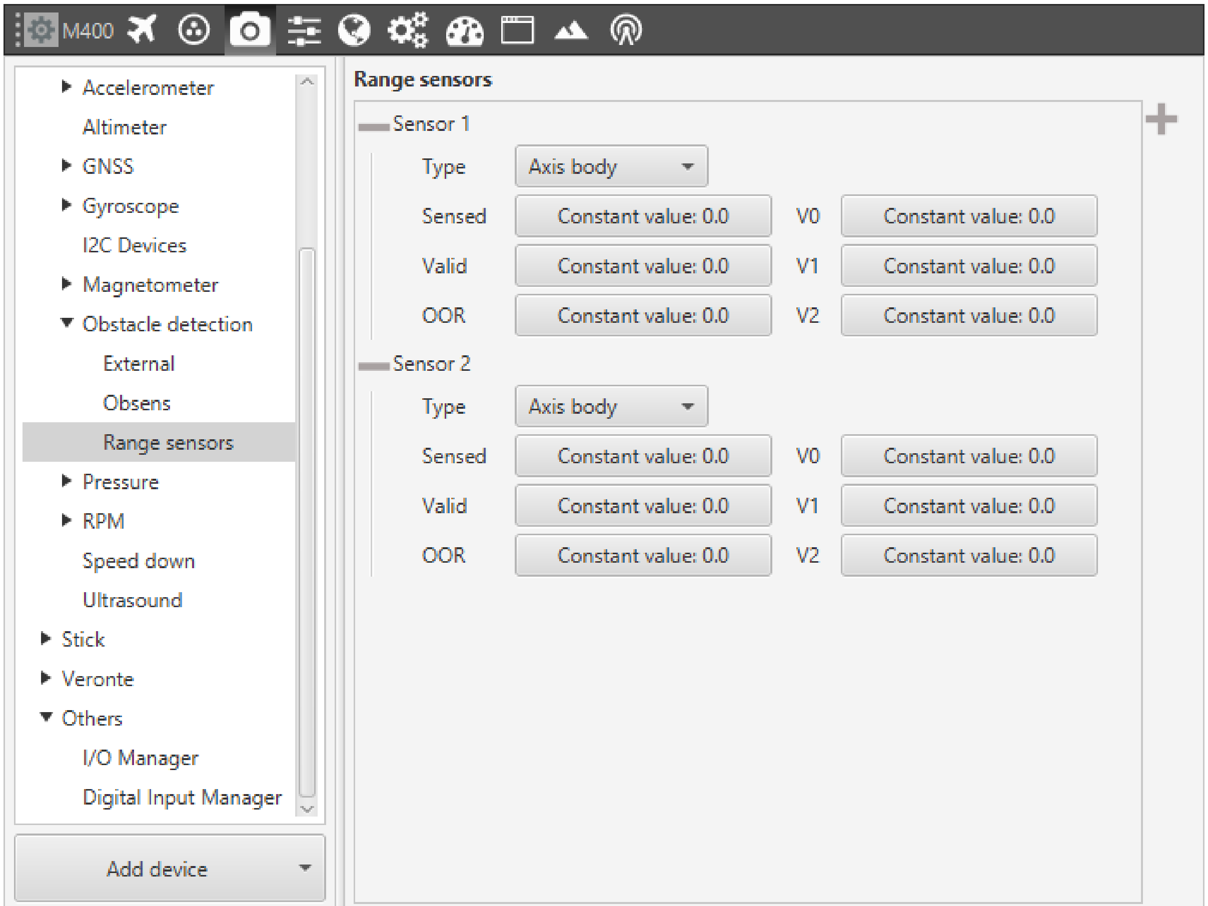
In this panel, it can be configured the following Obsens parameters:

- Mode: Cir2d (circular – 2 dimensions), Raw and off mode can be selected.
  - **Raw**. The sensors create the obstacle where it is located. At next step, if the obstacle is not present, the area is cleared.
  - **Cir2d**. Allows to input the number of divisions the area around the platform will have (considering a circular area). If one area is blocked due to an obstacle presence, even the sensors lost that obstacle, the area will remain blocked until enough distance has been reached.
  - **Off**. The obstacle detection is not enabled.
- **Type of obstacles generated**. 2D or 3D obstacles can be generated.
- **Distance parameter for obstacle generated**. Defines de radius of the obstacles generated with Obsens.
- **Time constant for field filter**. Filter applied to smooth the transition when obstacles are affecting the control.
- **Wrapper**. A minimum and a maximum elevation wrapper can be activated in order to limit the obstacle generation in a limited cone angle.

When configuring this sensor, obstacles will be created automatically, working similarly to the ones created in the mission toolbar.

6.2.3.4.7.3 Range sensors

The following figure shows the menu to configure the sensors which are used during the obstacle detection.



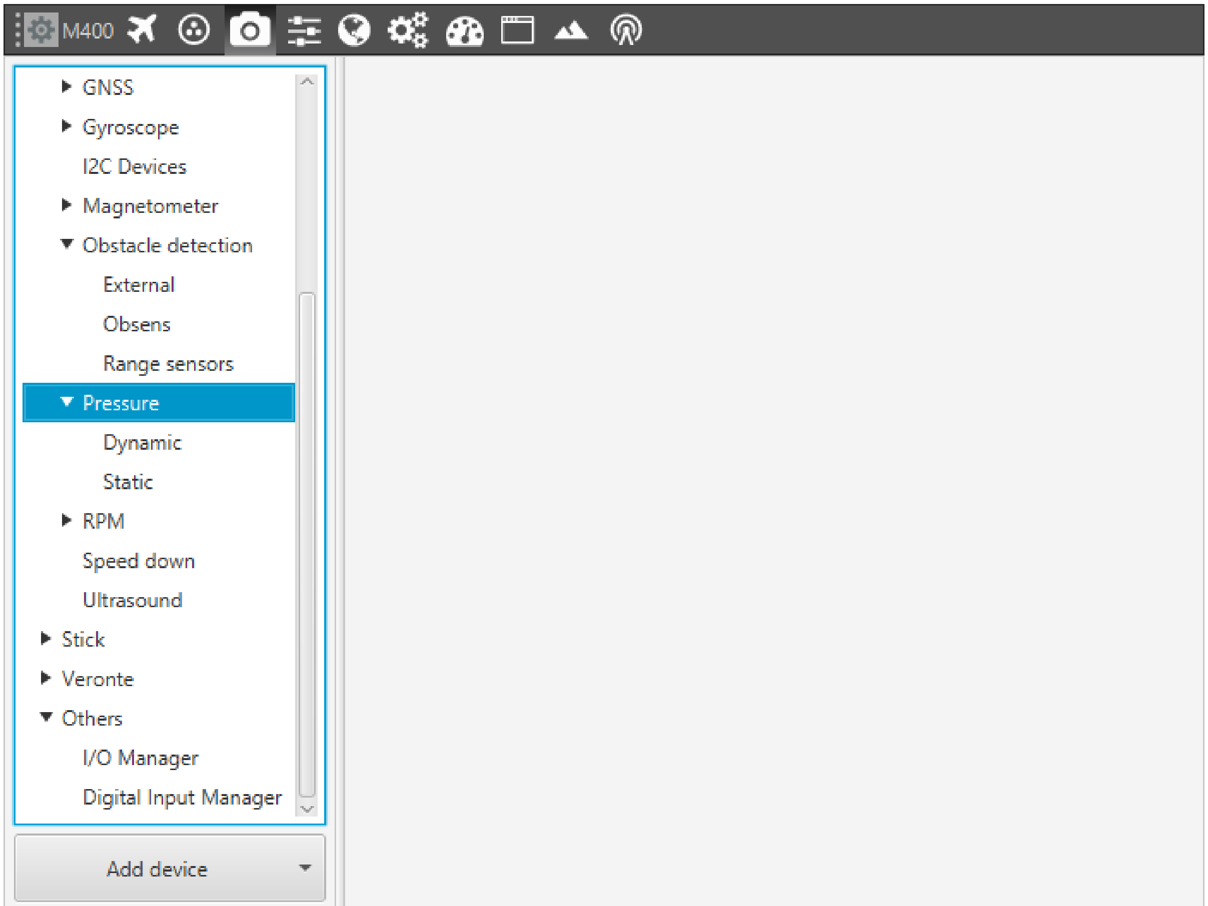
Obstacle Detection - Range Sensors

In this menu can be configured the following parameters:

- **Type**. Allows the user to select the kind of orientation which will be used: Axis body, Angles body or Angles NED
- **V0, V1,V2**. Once an orientation is selected, it is possible to edit where the sensor is pointing to by editing these values.
- **Sensed**. To select the variable which is measured.
- **Valid**. Usually sensors incorporates a varible to know if the data measured is valid. If that variable has been stored, it is possible to select it.
- **OOR**. Allows the user to select the variable that indicates if the data measured is out of range.

6.2.3.4.8 Pressure

In this menu the user can modify all parameters which are relevant to Pressure sensors.



Pressure - Options available

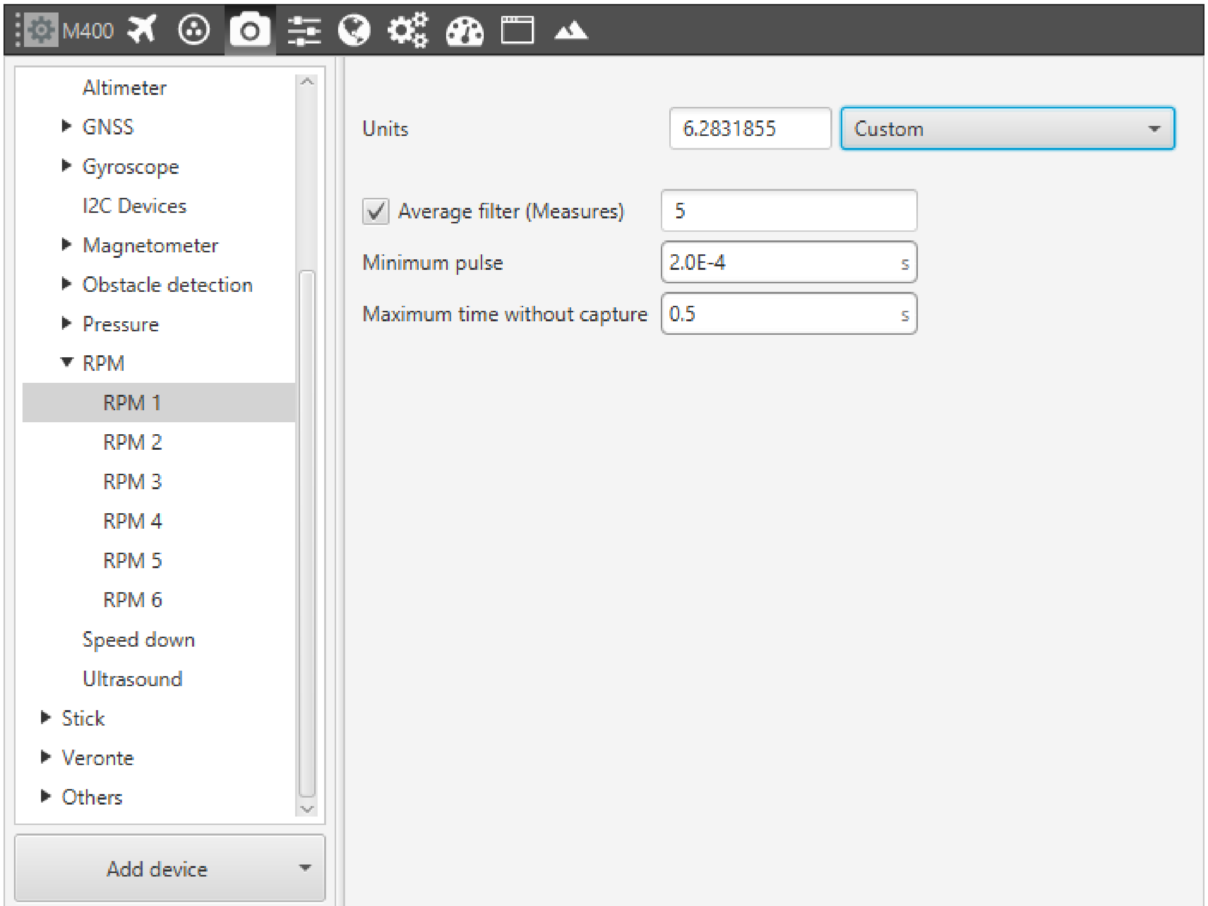
Veronte has two pressure input lines, one for static pressure to determine the absolute pressure and one for pitot in order to determine the dynamic pressure.

Absolute pressure connection on the aircraft is mandatory while pitot port can be obviated in some aircrafts, pitot port absence must be configured on Veronte Pipe software.

6.2.3.4.9 RPM

Users can connect in *Devices - Other - Digital Input Manager* up to 6 RPM signal.

The following menu allows the user to configure these sensors.

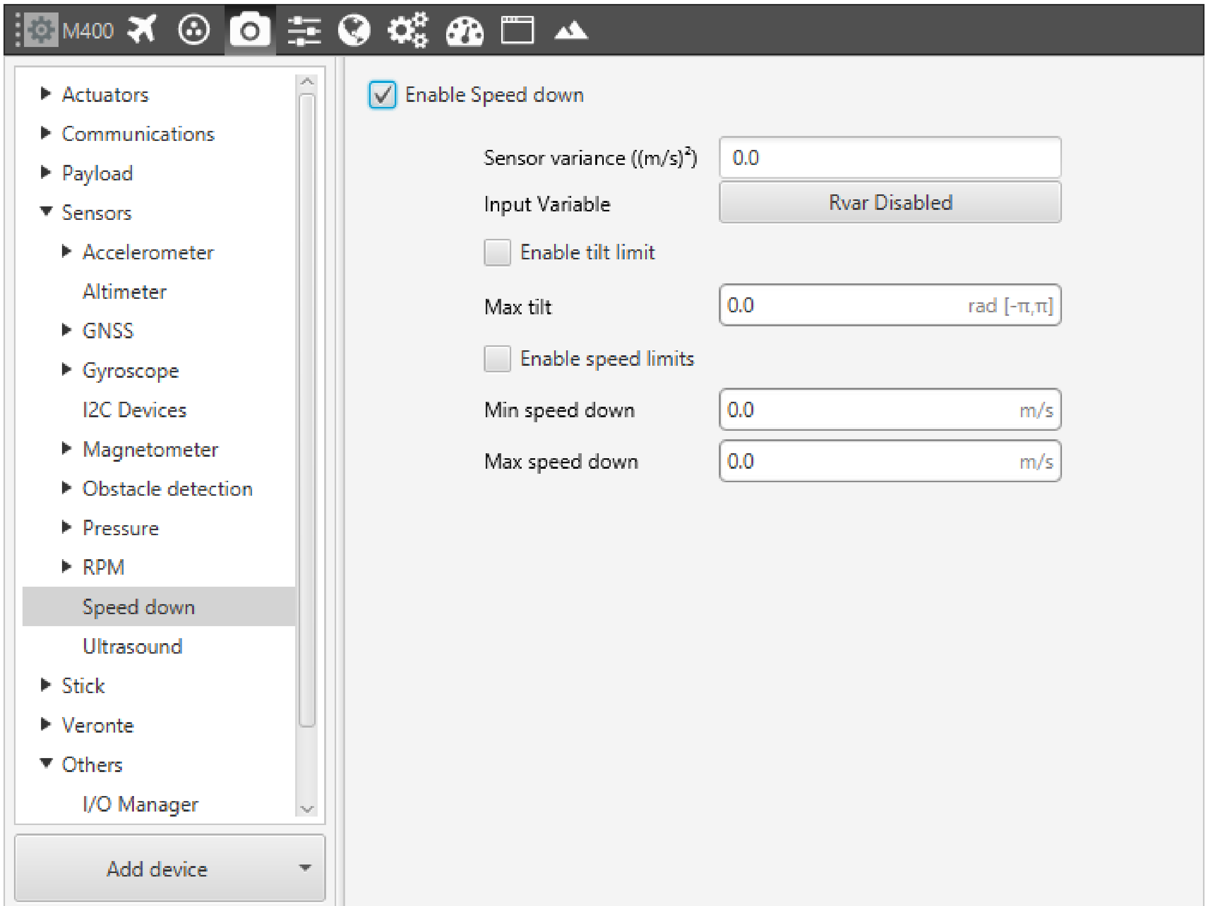


RPM - Menu Parameters

- **Units**. Sensor conversion factor. It can be Custom, Radians per pulse, Pulse per cycle.
- **Average filter (Measures)**. It is a filter to avoid voltage spikes.
- **Minimum pulse**. Minimum time to detect a lap.
- **Maximum time without capture**. The maximum period of time allowed without capturing.

6.2.3.4.10 Speed Down

The speed down sensor measures the velocity in the Z axis of the platform. Works in the same way as the altimeter, but in this case, instead of a position reading, the magnitude measured is velocity.



Speed Down - Menu Parameters

- **Sensor variance**. Variance of the error of the sensor in meters per second squared.
- **Input variable**. It indicates the variable that contains the information from the sensor.
- **Tilt limits**. The sensor measures the variable in a direction perpendicular to the longitudinal axis of the platform, so when it is tilted the reading will not be reliable. This option allows the definition of a tilt limit, so that if the limit is reached, the sensor reading will be discarded.
- **Speed limits**. Defines the limits of the speed measured by the sensor.

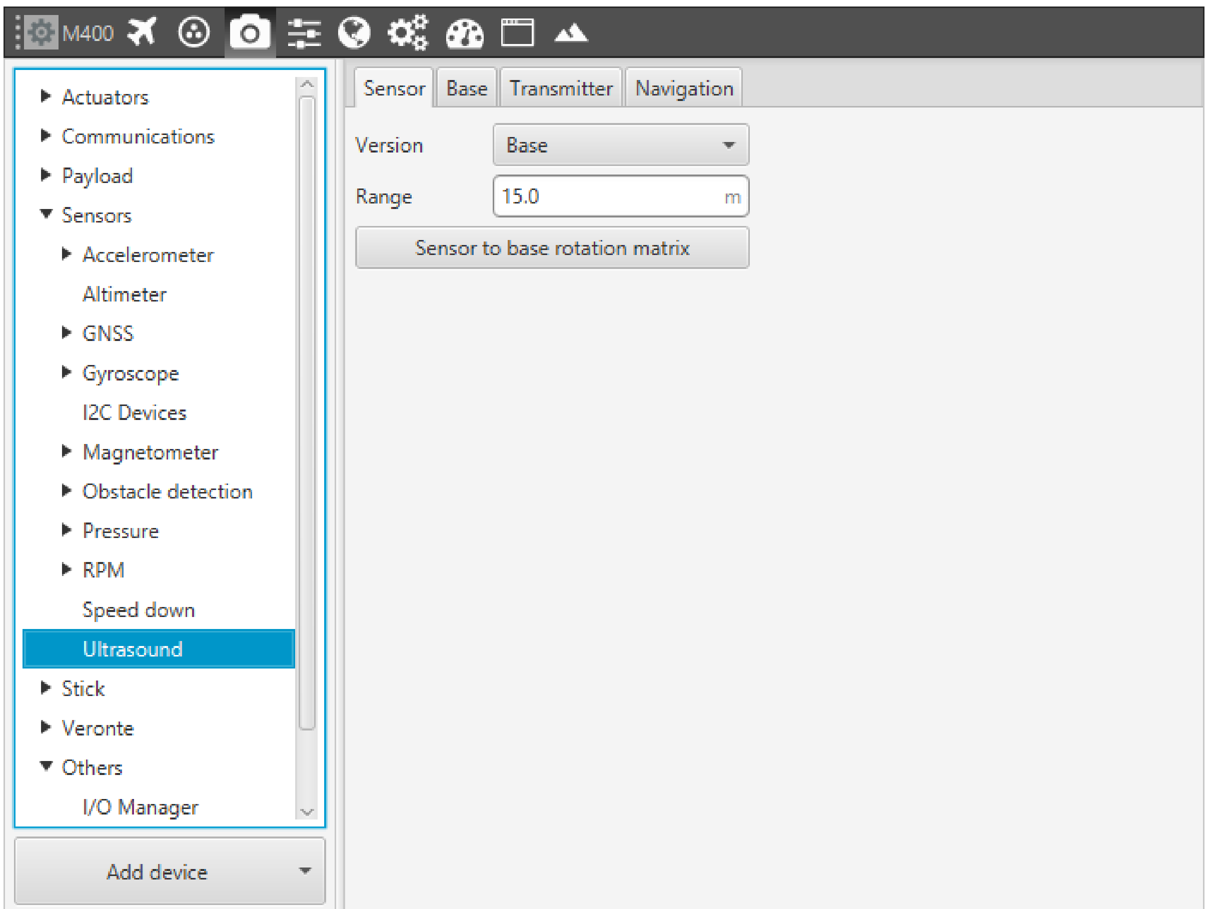
6.2.3.4.11 Ultrasounds

An ultrasound sensors computes Veronte position by measuring the time the signal sent out takes to return. The following panel allows the user to configure this sensor. This panel is used to configure an **Internest** system with Veronte Autopilot.

6.2.3.4.11.1 Sensor

In this menu the user is allowed to choose which Internest version is to be used, its range and the rotation matrix:

- **Version**. To be chosen between *Internest Base* and *Internest Explorer*.
- **Range**. Defines de distances at which Internest values will start to be valid.
- **Sensor to base rotation matrix**. Matrix to rotate the system and make it coincident with the Veronte Autopilot.

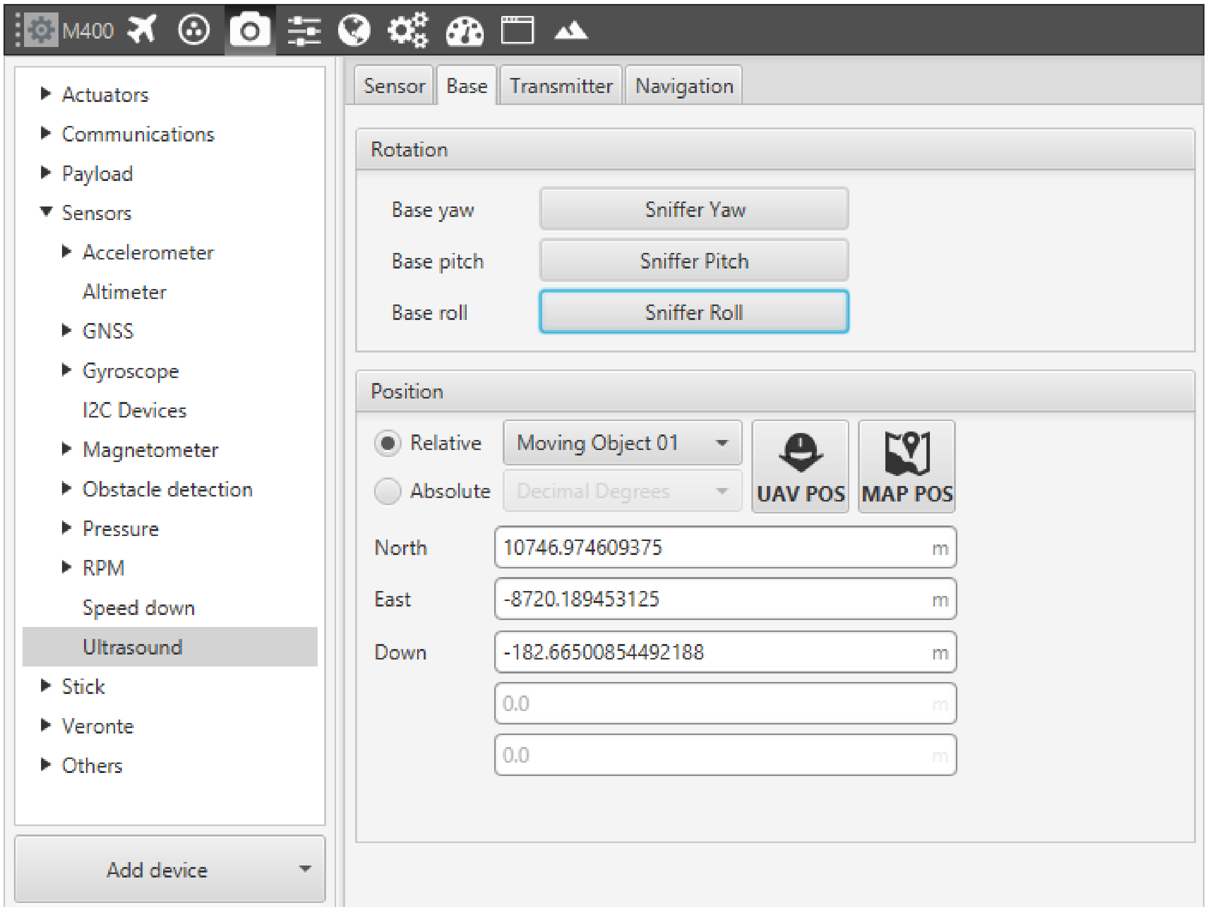


Ultrasound - Sensor Menu

6.2.3.4.11.2 Base

In the base menu the information *Sniffered* from the Base system is placed.

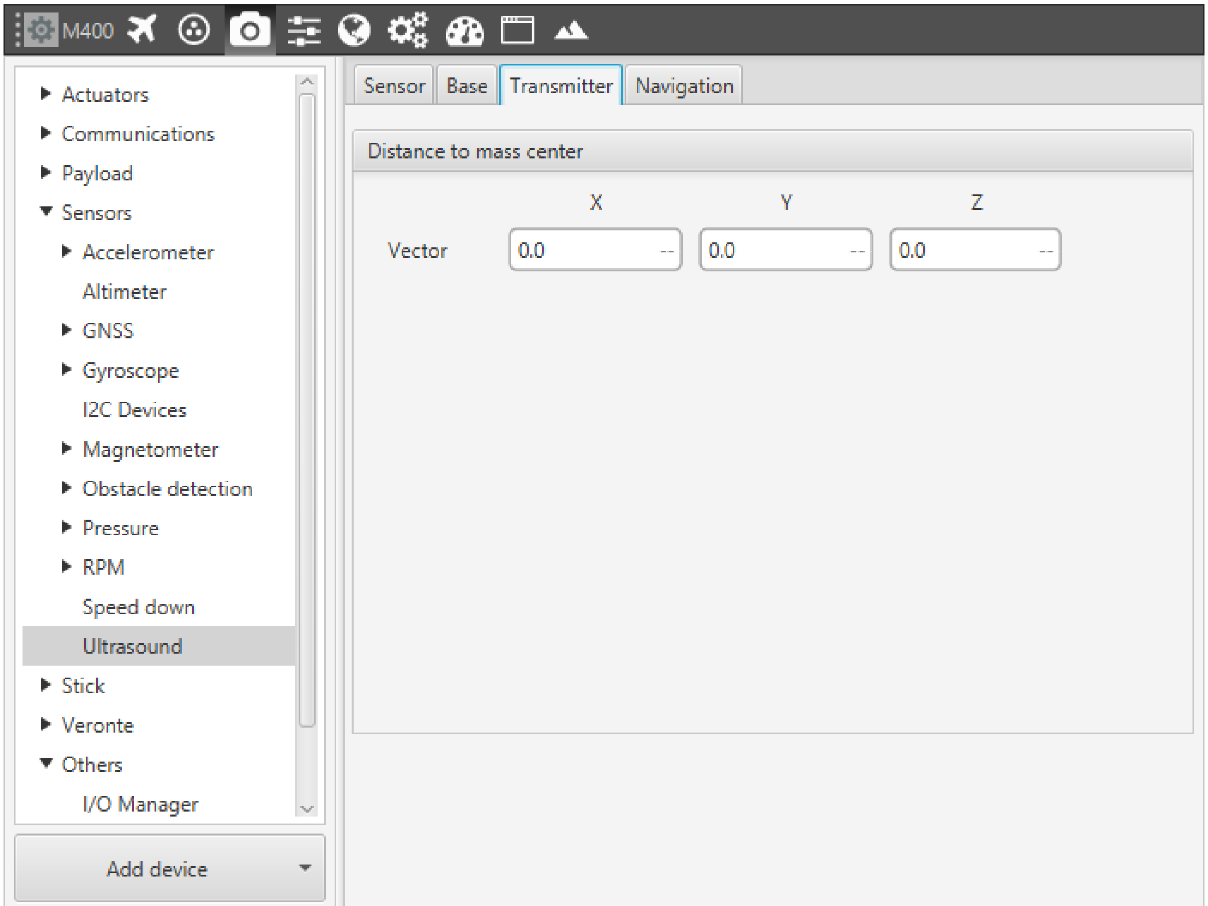
- **Rotation**. Input the 3 navigation angles from the Base platform.
- **Position**. Define it as Relative to Moving Object, which is the Base position sniffered.



Ultrasound - Base Menu

6.2.3.4.11.3 Transmitter

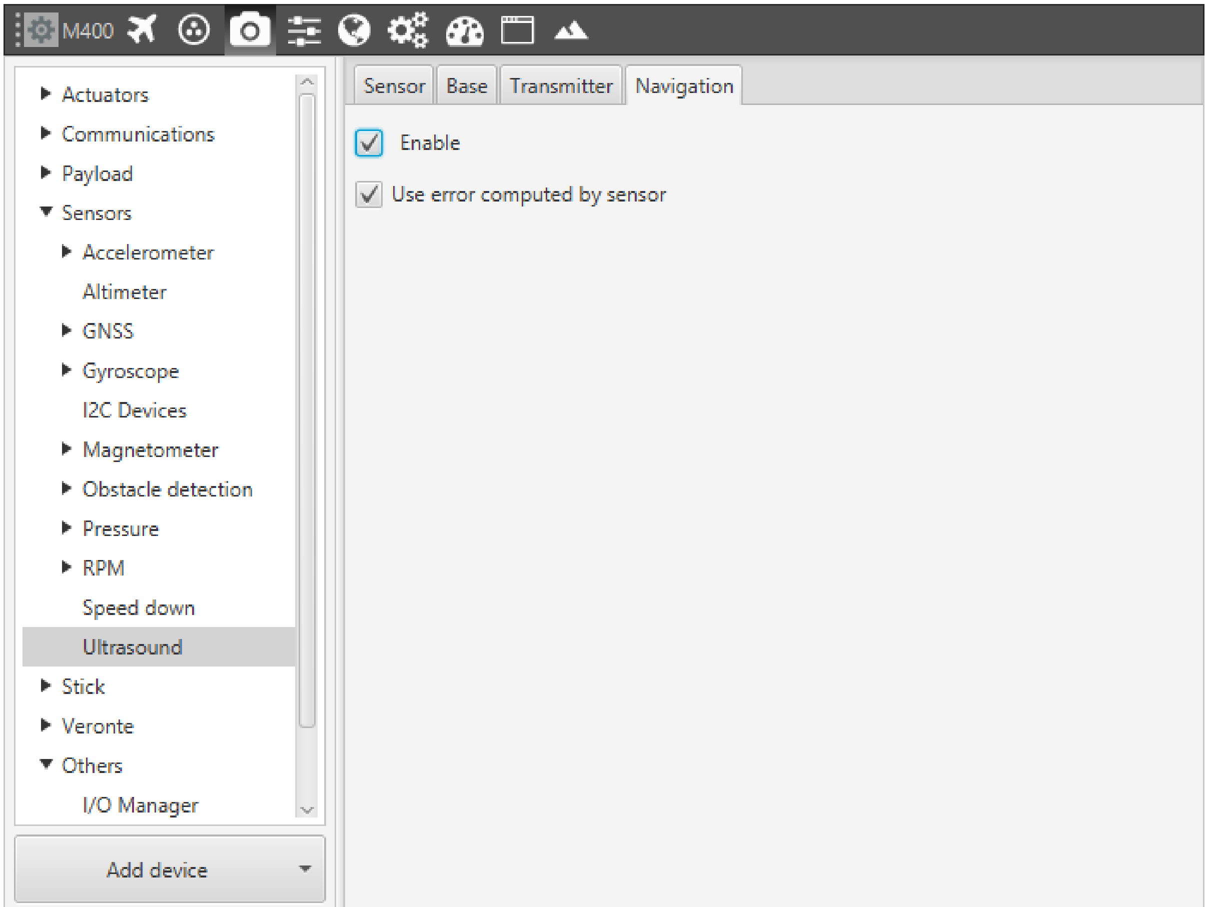
Defines the distance between the Internest system and the mass center from the platform.



Ultrasound - Transmitter Menu

6.2.3.4.11.4 Navigation

It has two checks configurable, one for enabling/disabling the Internest effect in navigation and a second one to use the error computed by the sensors if it affects the navigation.



Ultrasound - Navigation Menu

Sensors section permits to configure any sensor connected or internal from Veronte.

Accelerometer	Configure the accelerometer parameters
Altimeter	Enable the Altimeter into navigation if a sensor is connected to Veronte
GNSS	Configure GNSS receivers, RTK, Compass, External source and NTRIP
Gyroscope	Configure the gyroscope parameters
I2C devices	Configure Lidar sensors connected through I2C
Magnetometer	Configure the internal magnetometer or an external one
Obstacle Detection	Enables the obstacle detection and configured it
Pressure	Menu including both dynamic and static pressure configurations
RPM	Configure RPM sensors parameters
Speed Down	Enable the navigation using a Speed down sensor
Ultrasound	Menu to configure Internest with Veronte

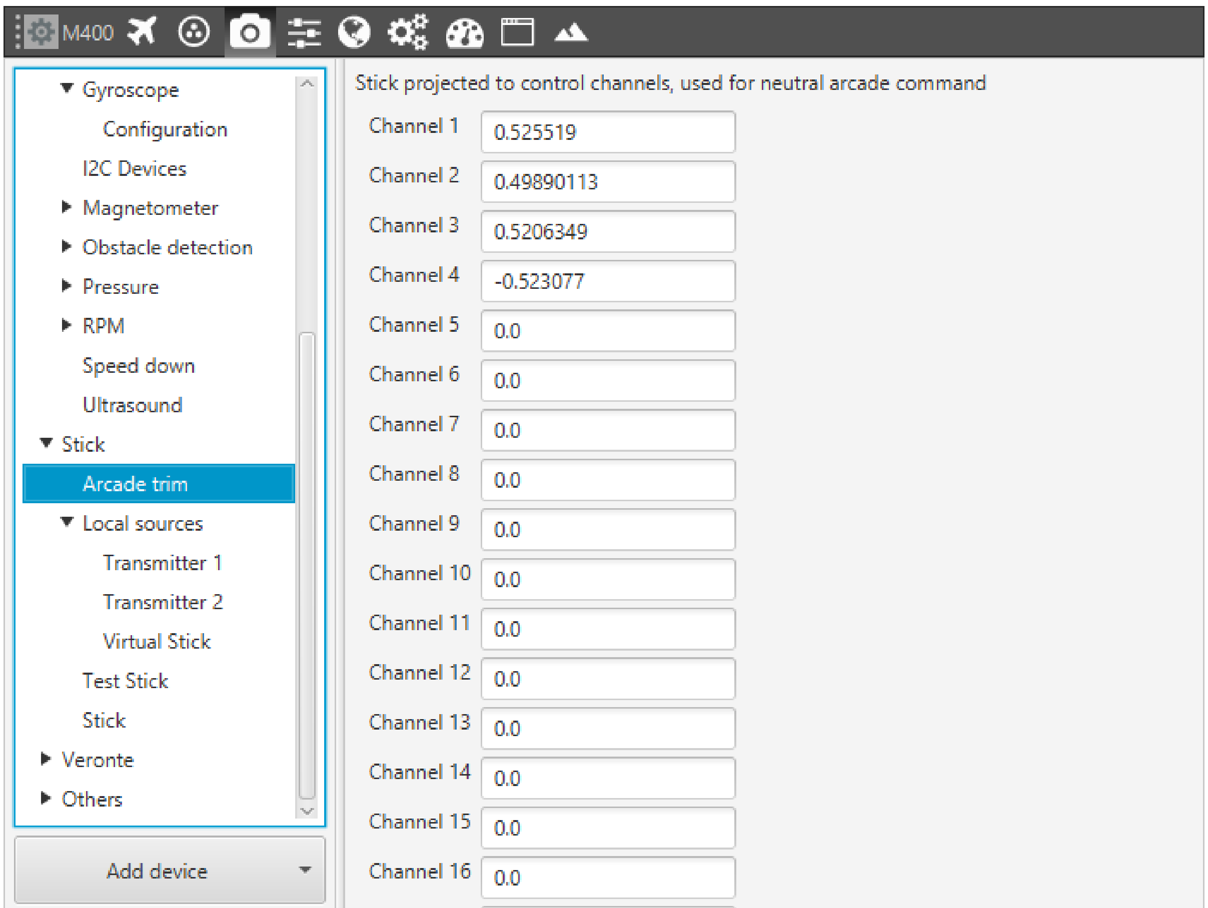
6.2.3.5 Stick Configuration

6.2.3.5.1 Arcade trim

The arcade trim is used to set the **zero-stick position** for the *Arcade Mode*.

In such mode, when obtaining the *Control output* $\boldsymbol{U}$ from *Stick input* $\boldsymbol{R}$ the final value is not the one that enters the navigation algorithm. The variable that is otherwise employed is $\boldsymbol{D}$, called "*Stick input d*", which is computed as $D = U - U_0$, being U_0 the arcade trim. In this way, when the sticks are trimmed at a certain position, the movement from that point will be the value of R that - after transformations - will generate the U.

The values of the trim vector U_0 can be introduced in **Devices - Actuators - Stick - Arcade trim** as shown in the figure below.



Stick Arcade Trim - Configuration Parameters

Another way to perform the arcade trim is to create the automation as described in Automations - Actions - Arcade trim. Once that is done, it is sufficient to move the stick main levers to their center position and to click on the *Trim Arcade button*. Then wait for 2-3 seconds and return the levers to their default positions.

> **Warning:** The Arcade mode has to be trimmed before flight. If not trimmed, the zero level will be different from the desired one.

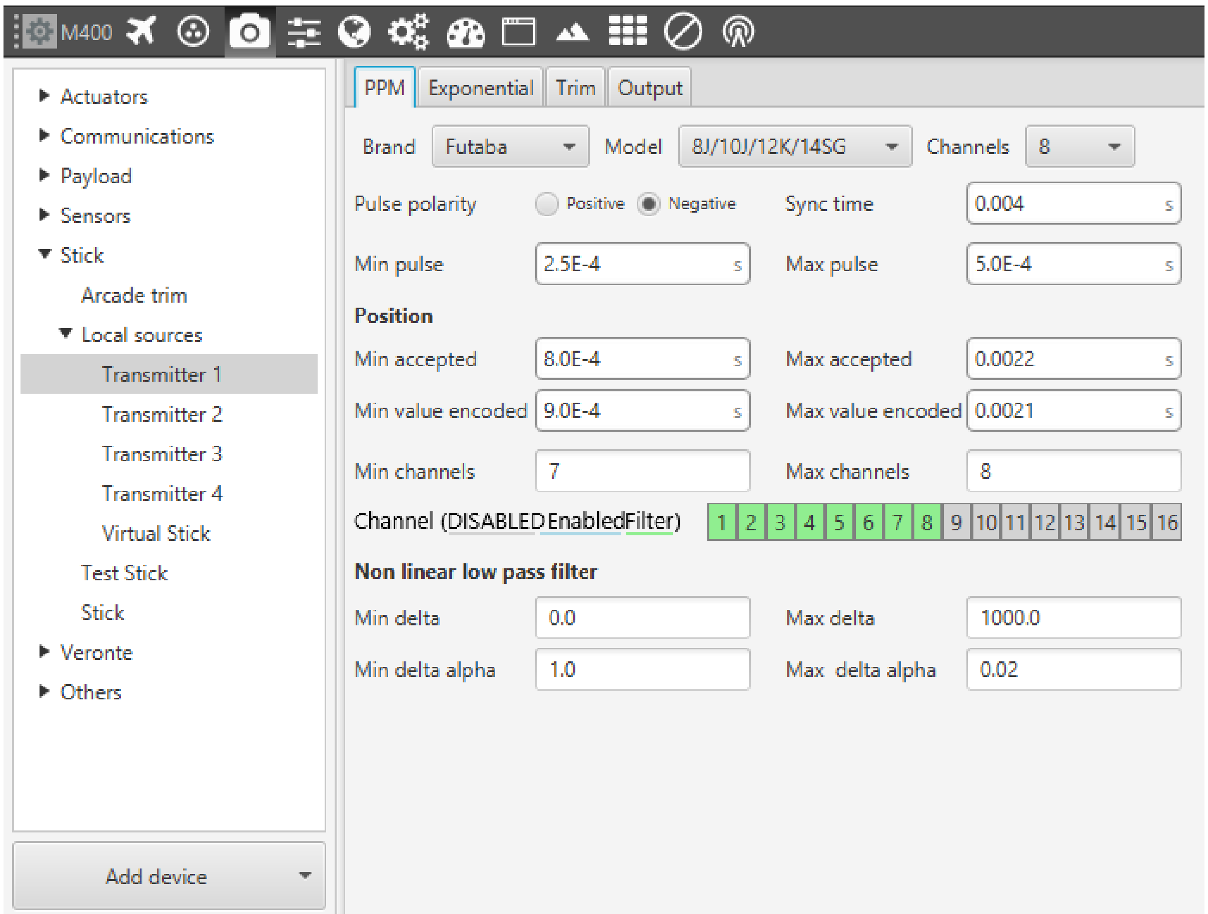
6.2.3.5.2 Local Sources

6.2.3.5.2.1 Transmitter

The wired connected transmitters are configured through the following menus.

6.2.3.5.2.2 PPM

This tab provides the options to configure a Pulse Postion Modulation (PPM) radio controller to control the platform fitted with the autopilot.



Stick Transmitter - PPM Configuration Parameters

- **Brand, Model & Channels**. Veronte Pipe has been configured to provide the user with the expected parameters to configure different transmitters models:
  - Futaba.
    * Model 8J/10J/12K/14SG with 8 channels.
    * Model 12K/14SG with 12 channels.
    * Model T18SZ with 8 channels.
  - Jeti.

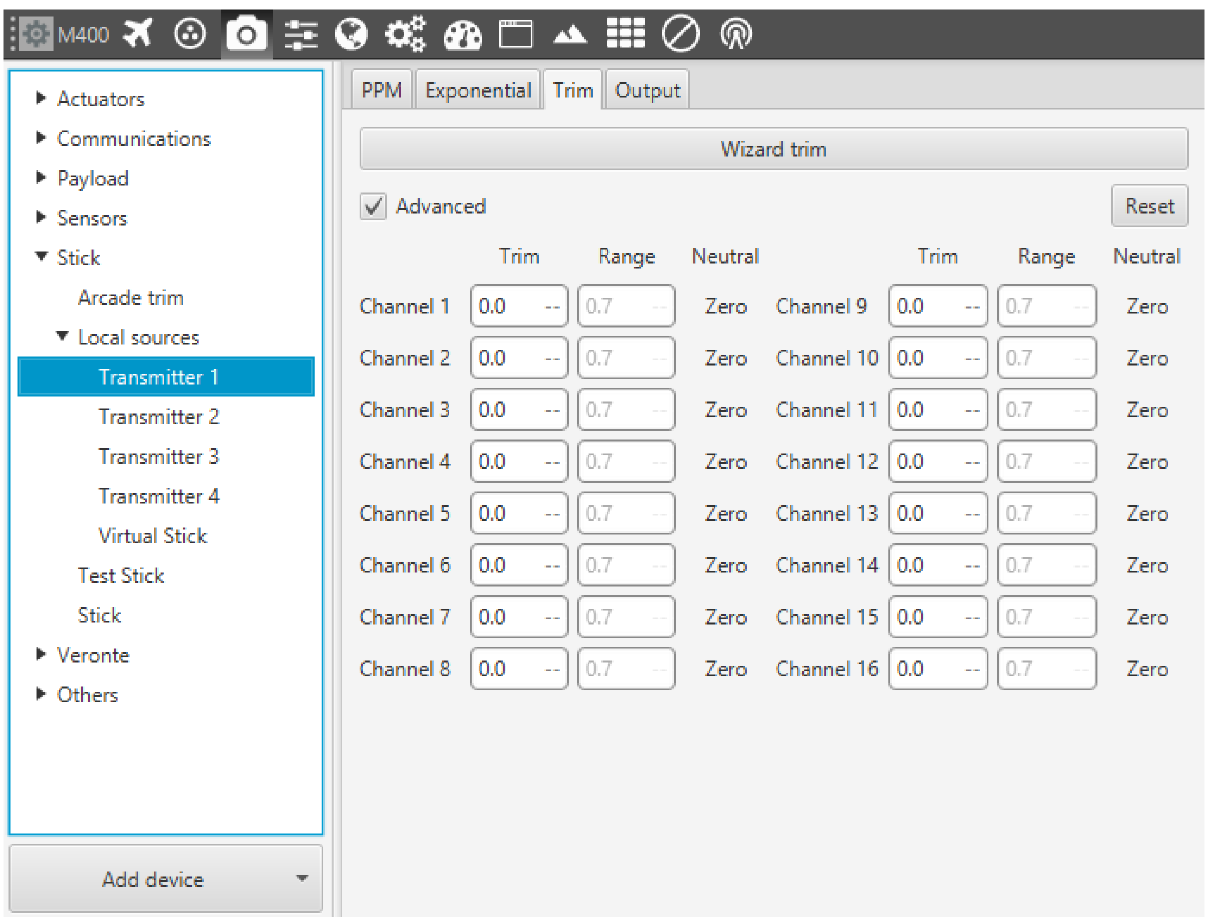
    * DC 16 / DC 24 with 16 channels.

  - FrSky.

    * Taranis 9XD with 8 channels.
    * X12S with 8 channels.

- **Polarity**. Indicates the pulse polarity. The image below shows a positive signal.
- **Sync time**. Minimum time on the PPM output till the next frame. It tells the receiver to reset its channel counter.
- **Minimum/Maximum pulse**. Pulse length, it depends on the system and it is a constant value (usually 0.2-0.5 ms).
- **Position – Minimum/Maximum accepted**. Pulse length accepted for each channel. Standard for R/C servos uses a pulse of 1 ms for the maximum position at one end, 1.5 ms for the midpoint and 2 ms for the maximum position at the opposite end.
- **Minimum/Maximum encoded**. If there is noise and the signal is varying around the minimum/maximum values accepted, Veronte will encode those values to the ones set here. For instance, a pulse length between 0.8-0.9 ms will be considered as one of 0.9 ms.
- **Channels**. Sets the number of channels accepted (minimun and maximun). Besides, it is possible to Disable/Enable/Filter each channel individually.
- **Non linear low pass filter**. Default parameters are recommended.

When some data is out of the established ranges, Veronte will discard that frame to avoid a possible malfunction. The next image clarifies the parameters introduced previously.

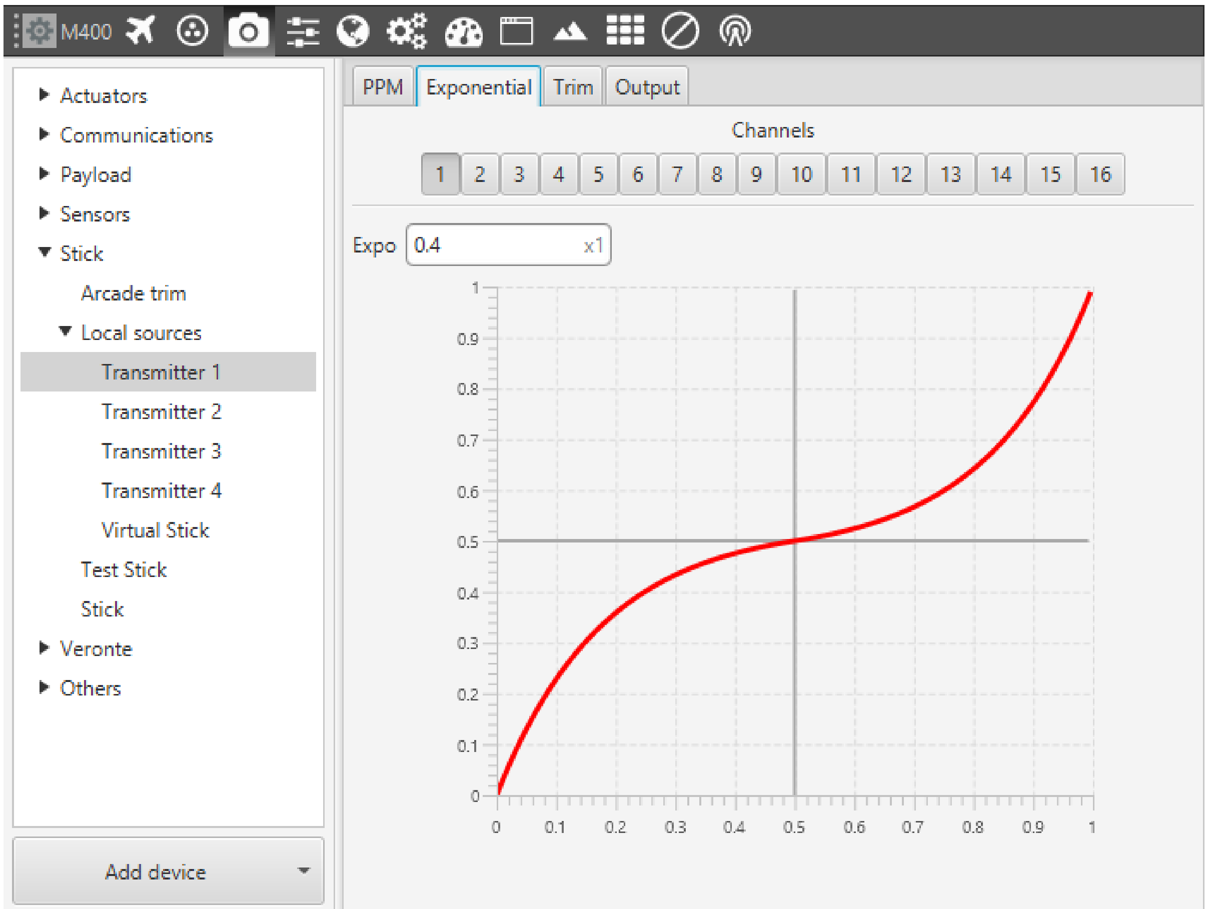


Transmitter - PPM Signal

6.2.3.5.2.3 Exponential

The second tab in allows the user to define an exponential stick response for every channel.

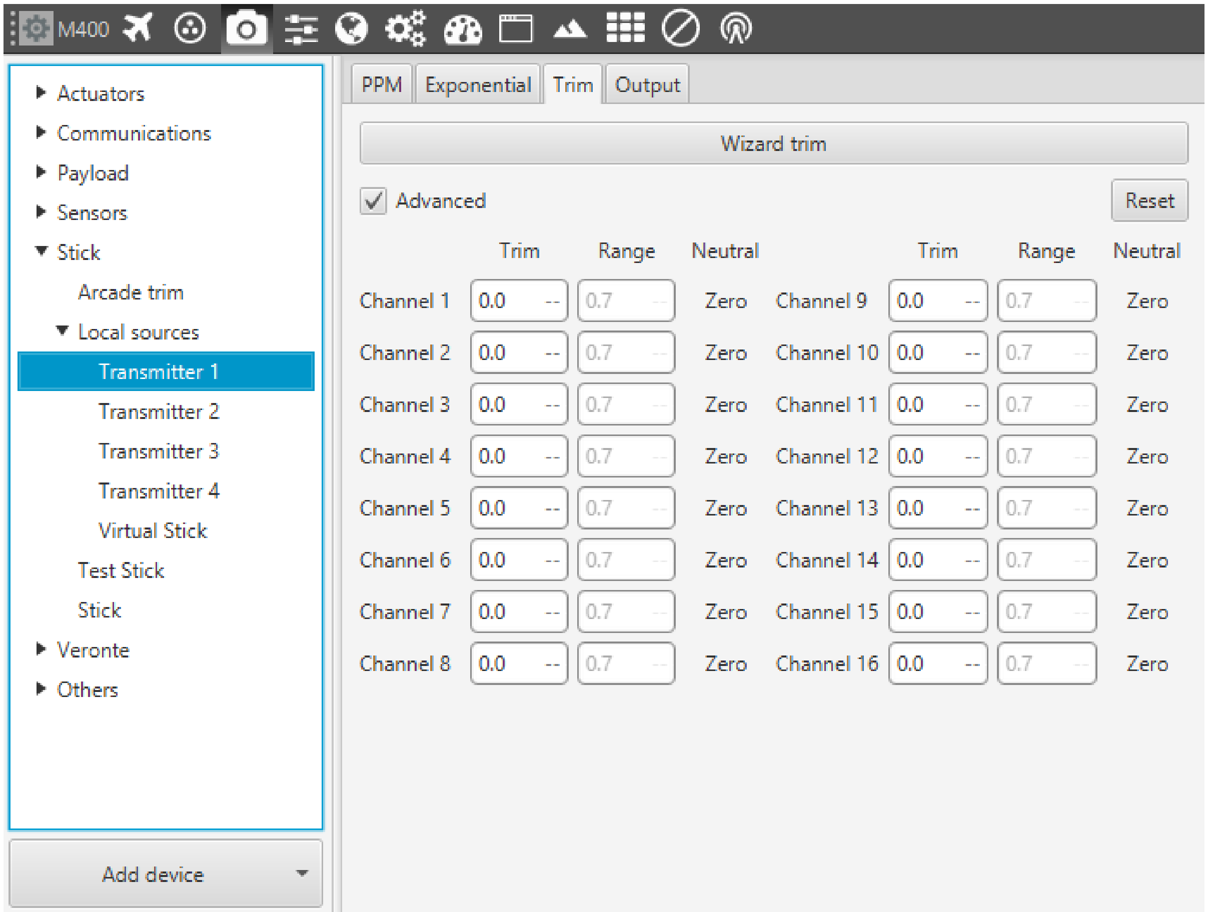
Allowed inputs go from 0 to 1 and there is a graph showing the generated response curve, as can be seen in the following figure.



Transmitter - Exponential

6.2.3.5.2.4 Trim

The third tab available is the Trim option. On the upper part, there is a *Wizard trim button*. The latter will guide the user through some pop-up windows.

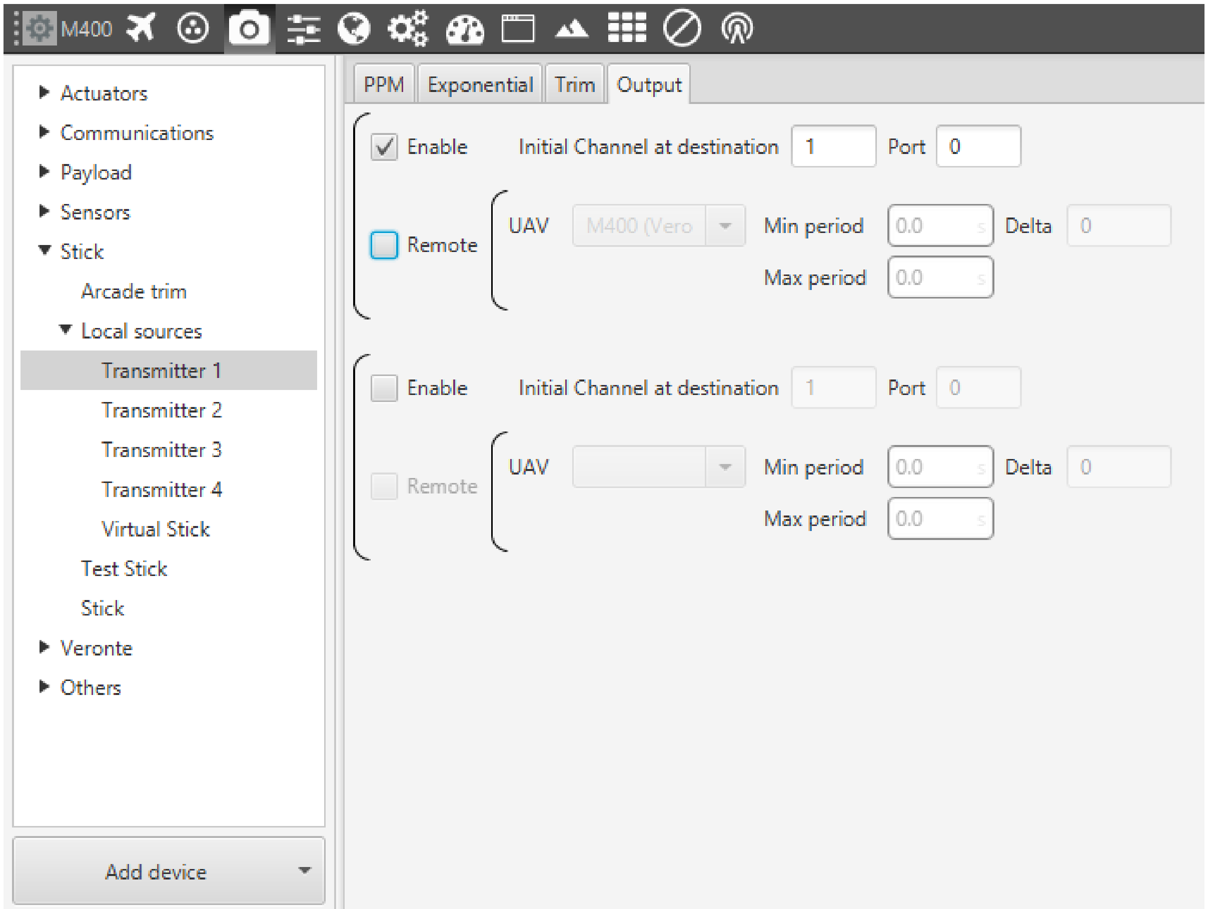


Transmitter - Trim

If that option is not chosen, ththe user can toggle the Advanced option and set the expected trim values manually. The user should have a deep knowledge on its transmitter if this option is selected. Finally, on the right hand side, the *Reset button* puts every parameter back to 0.

6.2.3.5.2.5 Output

Lastly there is the Output tab. Once the stick has been configured, the commands that arrive at the ground autopilot have to be sent to the air one. Here the user sets the receiving port and process the incoming commands.



Transmitter - Output

Once enable is flagged, the user indicates to which channel of the air autopilot (**AIR**) will be sent the first channel received in the ground segment (**GND**). The channels arrive at the platform in order and without spaces between them i.e, if at the **GND** channels 1,2,3,4 and 6 are enabled, the **AIR** will recieve channels 1,2,3,4 and 5. Therefore channel 6 of the stick will be channel 5 in the **AIR** configuration.

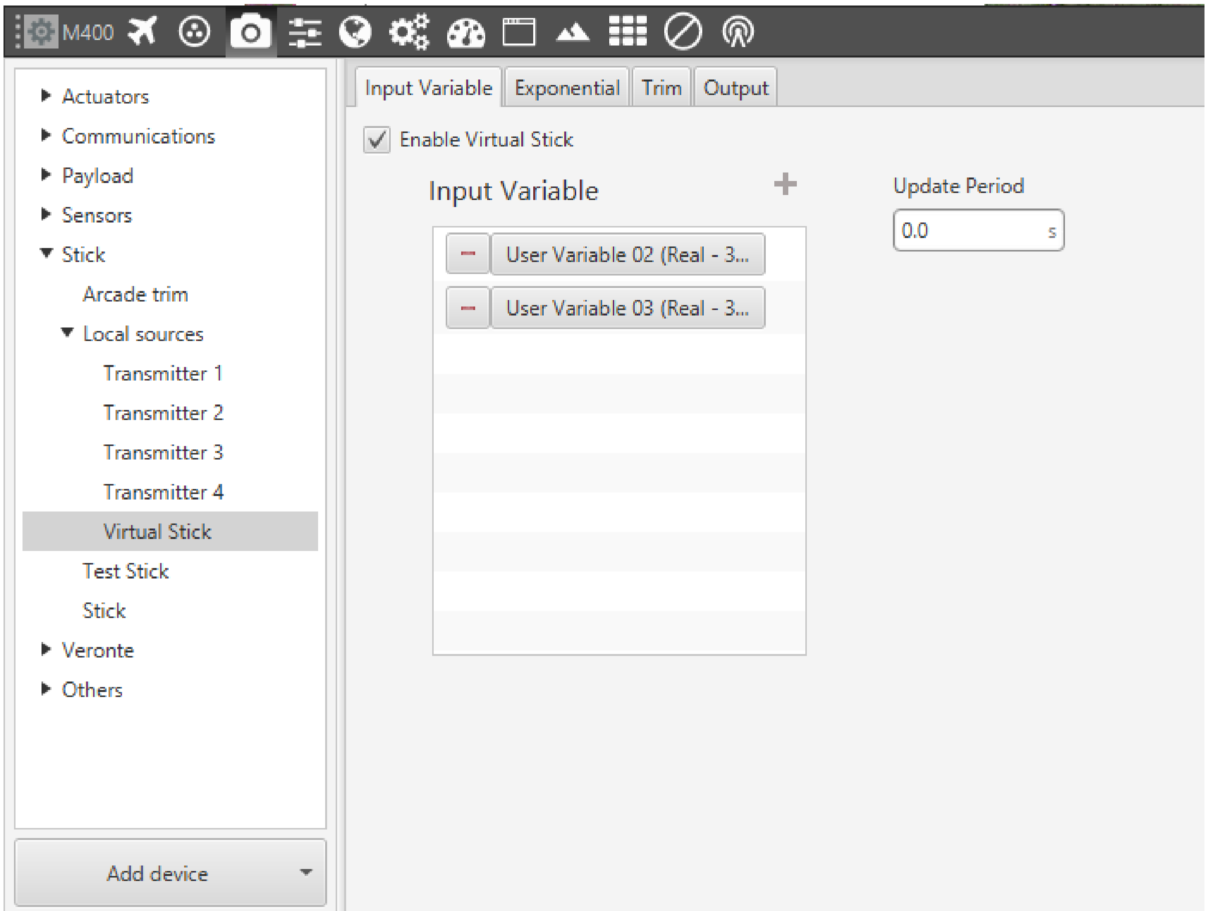
The option remote allows the delivery of the commands to the platform by indicating the address of the UAV. There is also the option *Broadcast*, where the commands are sent to all the air autopilots linked to the **GND**).

6.2.3.5.2.6 Virtual Stick

If the information of a stick is received through a different channel than PPM, the user can create a virtual stick to process that information and input it to the system.

In order to do so, enable the Virtual Stick, place the variables containing the stick information and configure the update frequency required.

The tabs Exponential, Trim and Output are the same as the Transmitter ones, so refer to *Transmitter* for more information.



Virtual Stick - Configuration Parameters

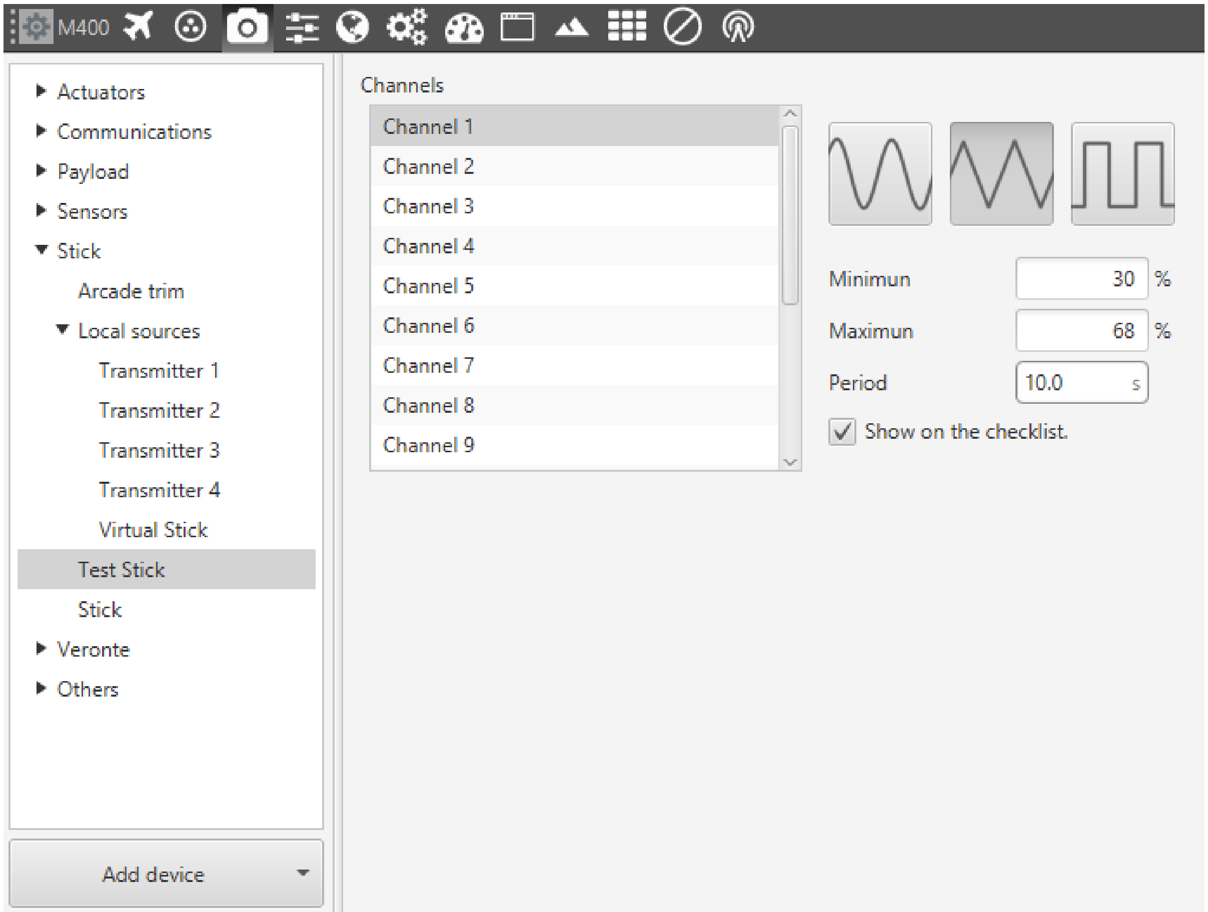
This part of the configuration toolbar allows the user to set **up to four transmitters** and **one virtual stick**. The autopilot's capabilities allows it to receive information from four different transmitters at the same time plus transforming some values into a virtual stick.

The content presented in the next two sections covers:

- Setting of the transmitter's parameters.
- Definition of exponential response-curves for the desired channels.
- Trimming of the channels' neutral position.
- Setting of the data receiving port on the autopilot.
- Definition of a virtual stick.

6.2.3.5.3 Test Stick

This menu is used to generate stick inputs that are introduced in the system. This is a way to check how the system behaves when a stick commands enters in the autopilot. For each channel, the user can set a continuos movement formed for example by a sine wave, spike-like funcion or a rectangular function. There are a set of parameters to be modified for the signals.



Test Stick - Configuration Parameters

- **Minimum & Maximum**: limit values that will be sent to the system for the stick test.
- **Period**: function period.
- **Show on the checklist**: configured parameters can be shown on the checklist in order to test the system prior to change flight phase - more information on *Checklists*. To activate the automatic movement, activate the virtual stick configured - see *Workspace - Widgets - Stick*.

> **Warning:** This test should be performed carefully if control surfaces/devices are engaged because the stick input should be carefully defined.

6.2.3.5.4 Stick

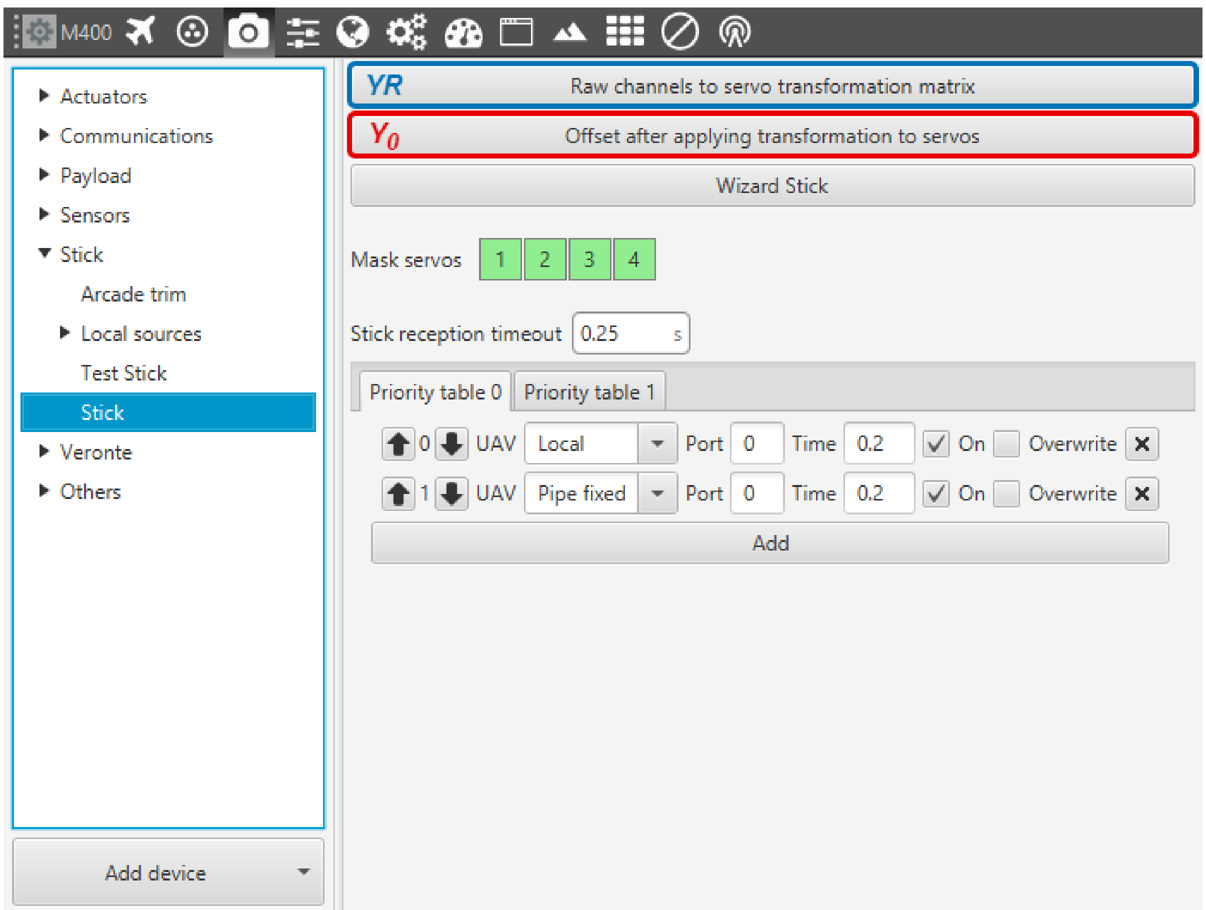
The following content will help the user to configure the stick parameters for manual and arcade modes – more information about these modes in *Modes*.

The movement that the pilot makes on the stick produces variations on a vector called R of length n, where n goes from 1 to the total number of employed transmitter channels. The values reached by the components of R are limited between 0 and 1. These stick movements need to be processed to produce the input signals that will go into the control algorithm, in the case of arcade mode; or directly into the servos for manual mode.

The process begins by mapping each one of the sticks inputs to PWM signals into a vector called Y of length m, where m goes from 1 to the total number of actuators. The full definition of Y is $Y = YR \cdot R + Y_0$, where :

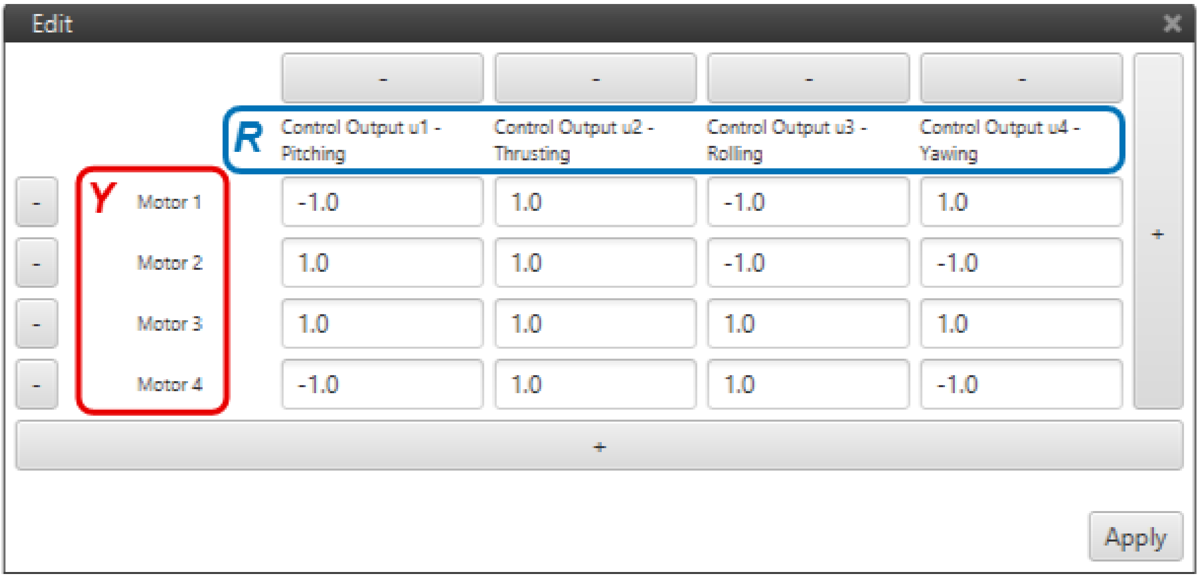
- $\boldsymbol{YR}$ is a matrix that transforms raw stick inputs R to PWM sigals Y.
- $\boldsymbol{Y_0}$ is an offset vector, which corrects the Y vector.

The above parameters are defined at beginning of the **Stick** menu (as seen in the Figure below). There is a *Wizard Stick button* that sets YR and Y_0, which has a dedicated section below.



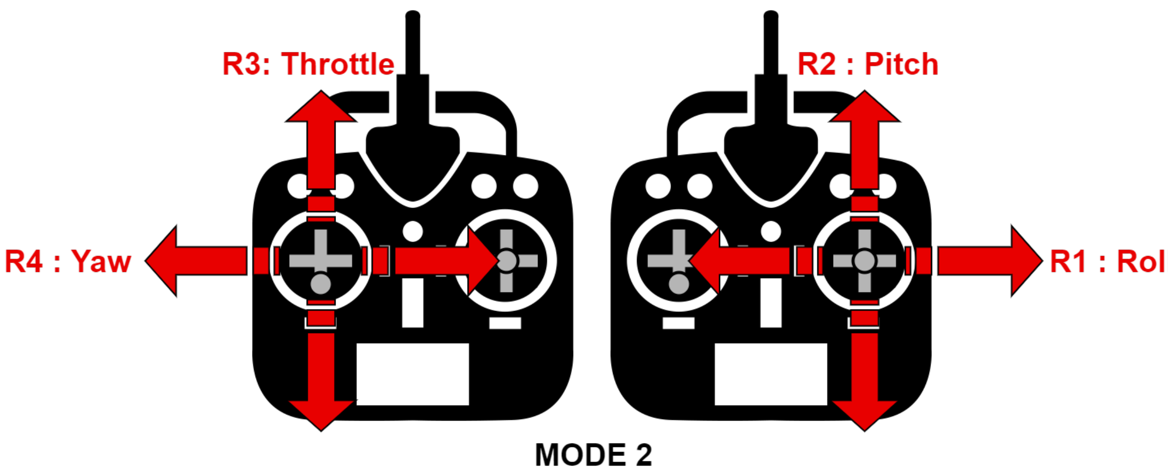
Stick - Configuration Parameters

Let's consider a quadocopter. With such platform applying thrust produces an increase in the rotation speed of the four engines, and only one stick channel (throttle) is required to achieve that. So here the mapping will be that an increase in one component of R has to produce an increase in four components of Y. Doing the same with the rest of stick channels (pitch, yaw, roll) will generate matrix YR:



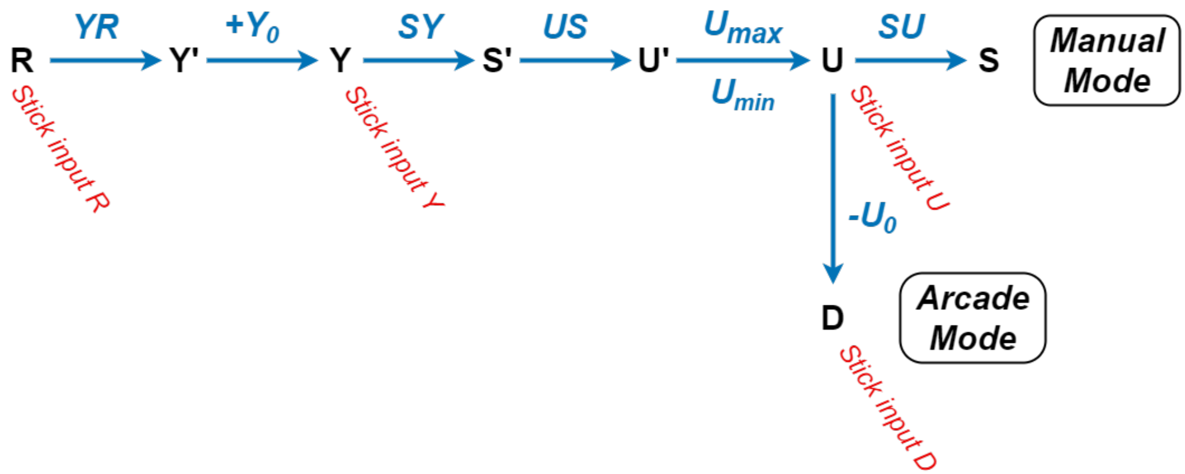
Stick - $\boldsymbol{YR}$ Matrix for a Quadcopter

Keep in mind that each stick channel (throttle, roll, yaw, pitch, etc) depends on the *mode* of the transmitter, i.e. the sticks' physical layout. There are 4 different modes and there is no prevailing type, it really depends on the pilot. In the following Figure a **Mode 2** channel layout is depicted:



Stick - Channels Layout

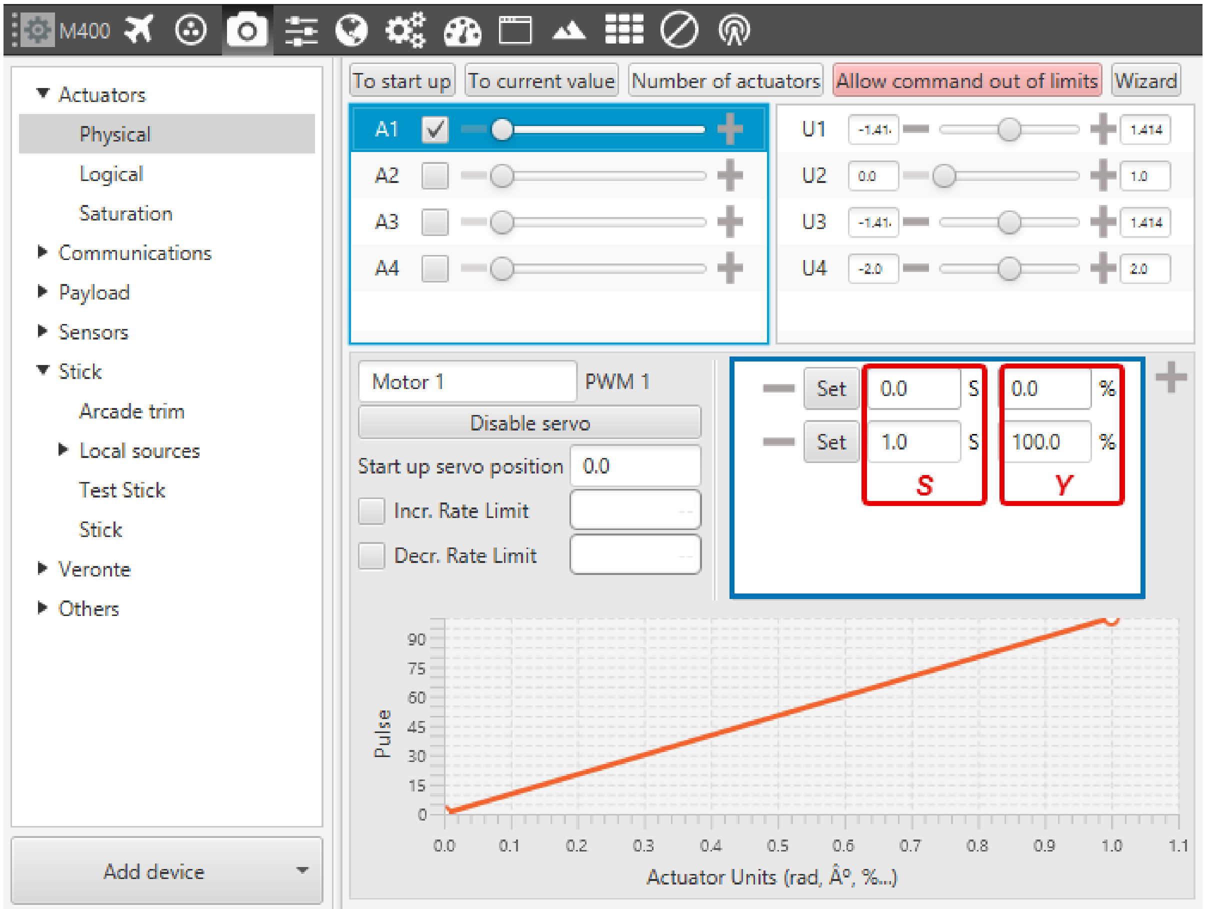
The offset Y_0 is calculated by setting R and Y with a specific value. Take the neutral position of a quadcopter: the sticks are $R_{neut} = [0.5; 0.5; 0; 0.5]$ (the third value is 0 as it corresponds to the neutral position of channel 3, i.e. the throttle) and the actuators PWM signals $Y_{neut} = [0; 0; 0; 0]$. Then, the solution for the offset vector will be: $Y_0 = Y_{neut} - YR \cdot R_{neut}$. The neutral position information will be asked when using the *Wizard Stick button*.



Stick - Transformation Diagram

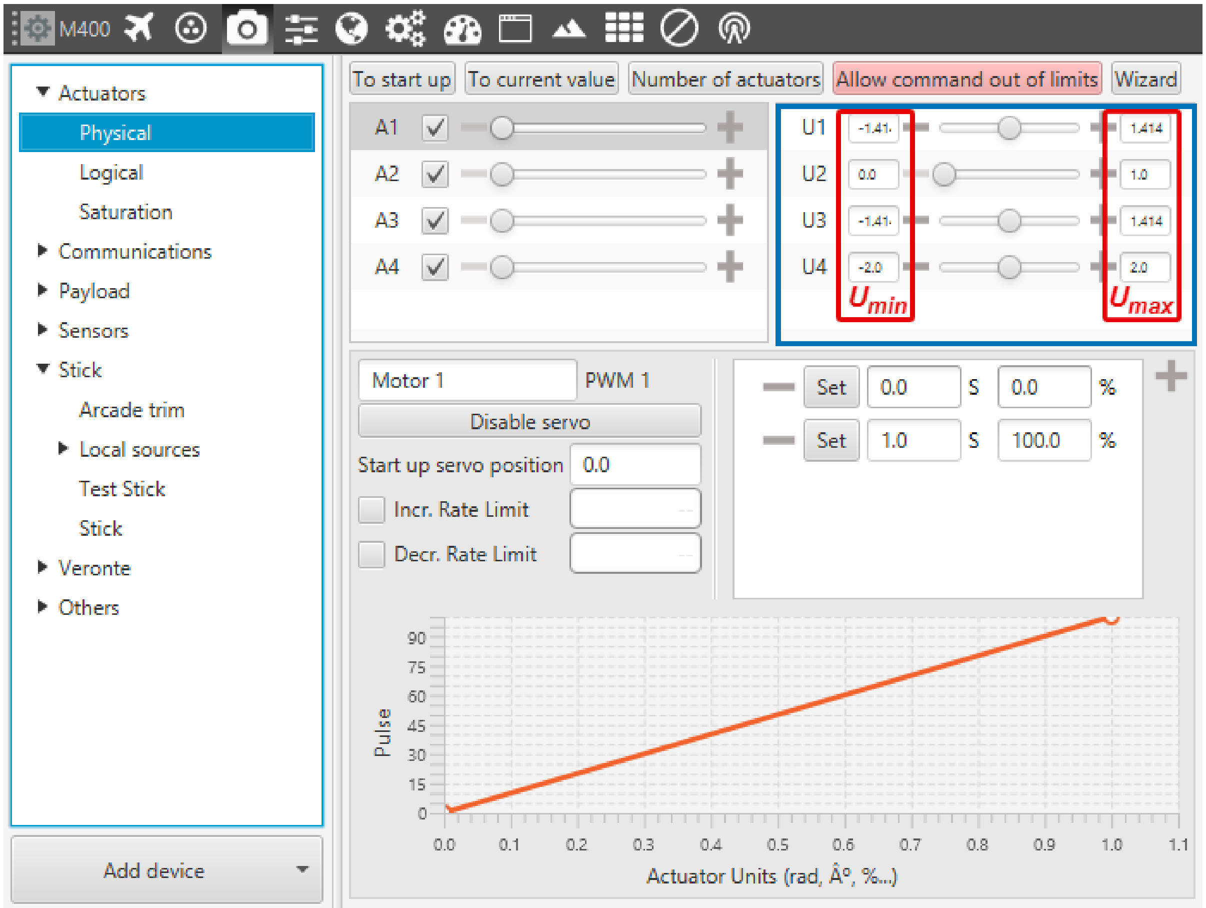
The above diagram depicts the whole transformation process of the incoming raw input R vector. It also shows in red the naming of some important variables in Veronte Pipe. Step by step, the process is:

1. From R one gets Y' using YR, as defined above.
2. From Y' one gets Y adding Y_0.
3. From Y, using the trim of the servos, one gets the actuator output S'. PWM to actuator output mapping SY is defined in *Devices - Actuators - Physical*. The mapping consists in a piecewise-linear function relating each actuator output S to a certain amount of PWM signal S:



Linear Mapping SY of a quadcopter engine

4. From S', using the logical transformation matrix US (defined in **Devices -> Actuators ->** *Logical*), one gets U'.
5. From U', and delimiting its values if they are higher or lower than the limit values U_{max} and U_{min}, one gets U. The limit values of U are found in *Devices - Actuators - Physical*.



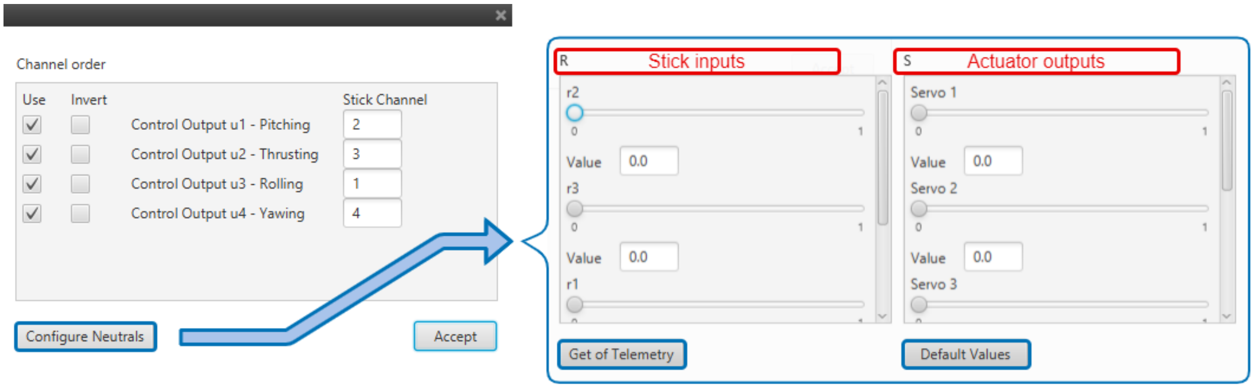
Limit values of U

6. From U, substracting the arcade trim of the transmitter U_0 one gets vector D, which goes into the guidance control of the modes where arcade mode is enabled. More information on the arcade trim on *Devices - Stick - Arcadetrim*.
7. Lastly, if manual mode is selected, one last transformation is required. From U, applying SU, one gets the final S: actuator outputs.

6.2.3.5.4.1 Wizard Stick

To make use of this assitant tool, logical matrices US and SU must have been defined, otherwise it will not work.

When clicking on the button (see Figure below), the user is asked to associate each control output U of the platform with a stick channel R. Doing so, matrix YR is automatically calculated.



Stick - Wizard Stick Configuration Parameters

In addition, to obtain Y_0 the user needs to click on *Configure Neutrals* – otherwise it is considered as $Y_0 = 0$. There, the neutral values of the actuator outputs S_{neut} (i.e. servo positions, engine rpms, etc) and the neutral values of the stick R_{neut} can be introduced manually. If *Default Values* button is clicked, S_{neut} is taken from *Devices - Actuators - Physical*, where its start-up position is defined.

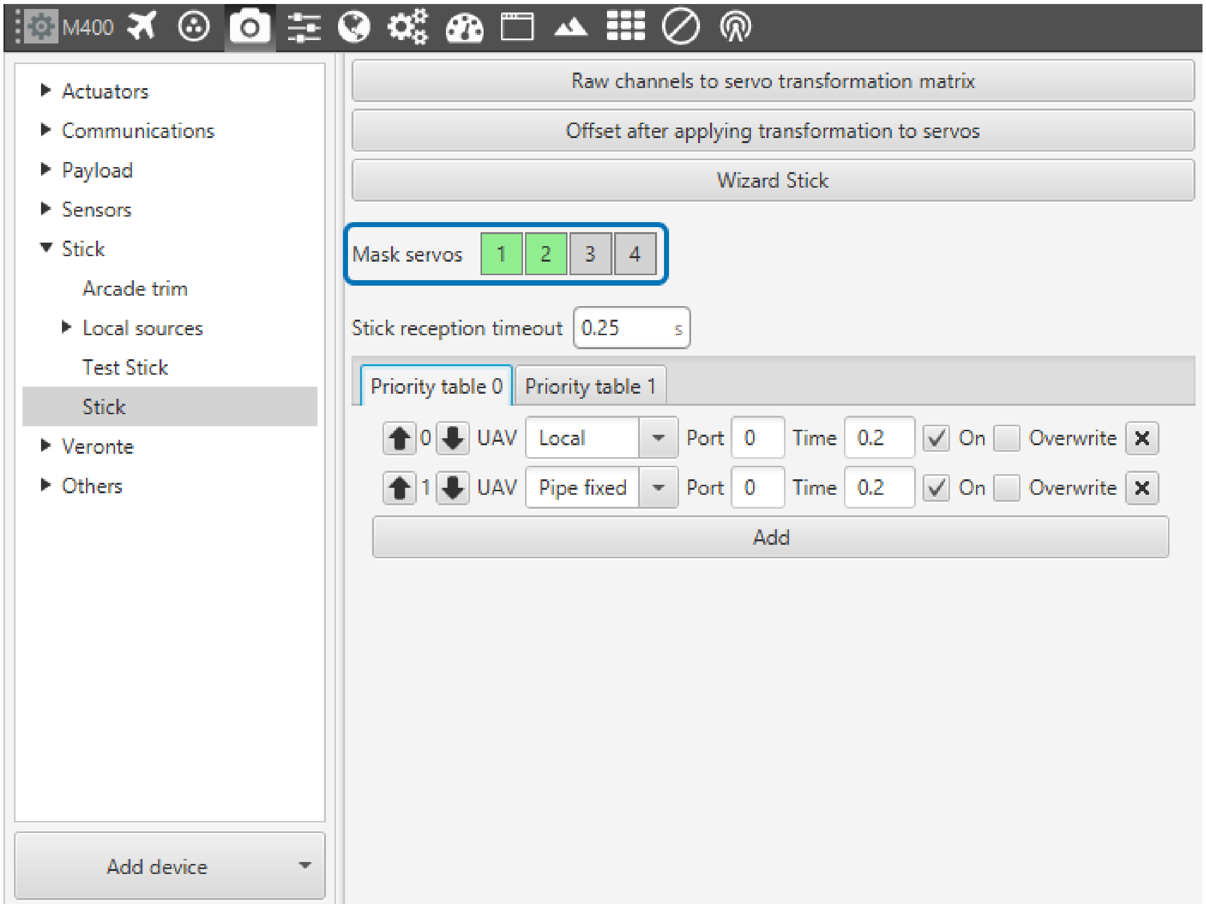
If *Get of Telemetry* is clicked, R_{neut} is taken from the connected transmitter. The user should move the sticks to their netural position and then click on the button. Finally, to exit the assistant tool click on *Accept*.

Note: The assistant tool might not work for some configurations, such as **hybrid platforms**.

For those platforms where a same channel is used both for the plane and the quadcopter configuration, the stick configuration has to be done manually.

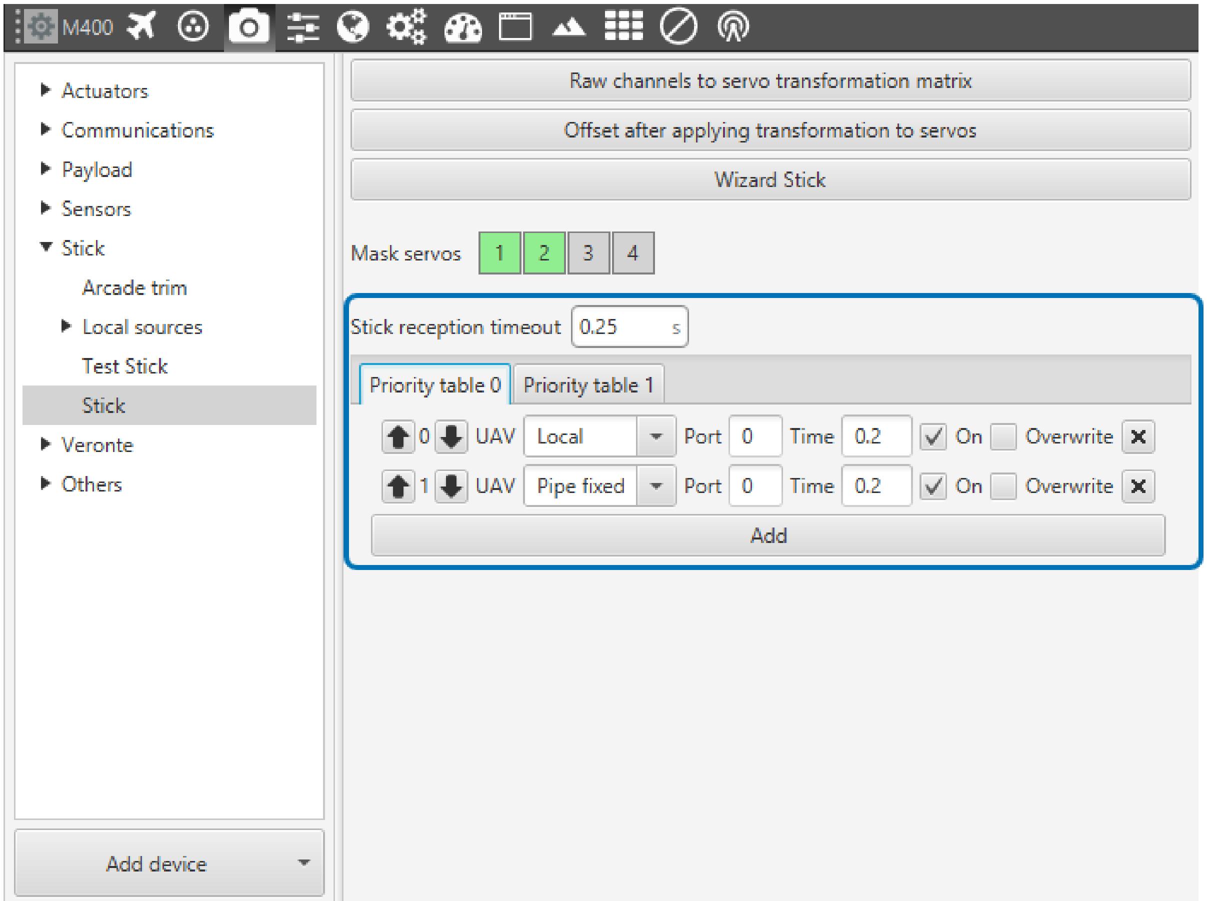
6.2.3.5.4.2 Mask Servos

This option is used to select which servos will "listen" to the commands sent by the stick. If the servo is selected (green box), it means the servo will be moved upon stick command. On the other side, if a servo is not selected (grey box), it will ignore the commands sent from the stick.



Stick - Mask Servos

6.2.3.5.4.3 Transmitter Inputs



Stick - Transmitter Inputs

It is possible to set multiple transmitter inputs with the respective priority, from top to bottom.

- **UAV**: defines the source of where the stick information is taken from. *Local* represents the actual selected autopilot (i.e. the transmitter is connected to it); *Pipe fixed* means the information is coming from the virtual stick; *Broadcast* means the information comes from all linked autopilots; and one particular autopilot, which needs to be visible in Pipe.
- **Port**: from each source it is posible to have more than one stick information, e.g. two transmitters can be connected to the same autopilot. The port is an identifier to distinguish them.
- **Time**: defines the time to consider the source inactive. Therefore the incoming stick information will be always the one from the source with higher priority and active. Once it is considered inactive the following active source will send its stick information.
- **ON**: enables the source.
- **Overwrite**: if marked, the stick information will be overwritten by this source when it becomes the active source.

In this section, the stick configuration on Veronte Pipe is explained. The next table details what is going to be covered:

Arcade trim	Neutral position value for the sticks
Local sources	External transmitter configuration (up to 2) and 1 virtual stick
Test Stick	Testing of the transmitter configured channels
Stick	Configuration of the stick matrices, offsets and transmitter priority

All these menus help the user in configuring the transformation of stick inputs into control outputs. Summarized, the following process is followed:

- Stick inputs $\boldsymbol{R}$ are named "*Stick input* r_n", where n goes from 1 to the maximum number of employed channels of the transmitter. R values go through the YR transformation matrix (see **Devices -> Stick ->** *Stick* menu), which converts raw stick iputs to servo PWM signals. And become $\boldsymbol{Y}$, which is named "*Stick input* y_m", where m goes from 1 to the total number of actuators of hte platform.
- Y PWM signals are corrected by adding an offset matrix Y_0. After that, Y values are converted into $\boldsymbol{S}$ values, known as "*Actuator outputs* S_m". The transformation is performed according to the relation defined in **Devices -> Actuators ->** *Physical* for each actuator.
- S values then go through the matrix US, defined in **Devices -> Actuators ->** *Logical*, and become $\boldsymbol{U}$, the "*Control outputs* U_n".

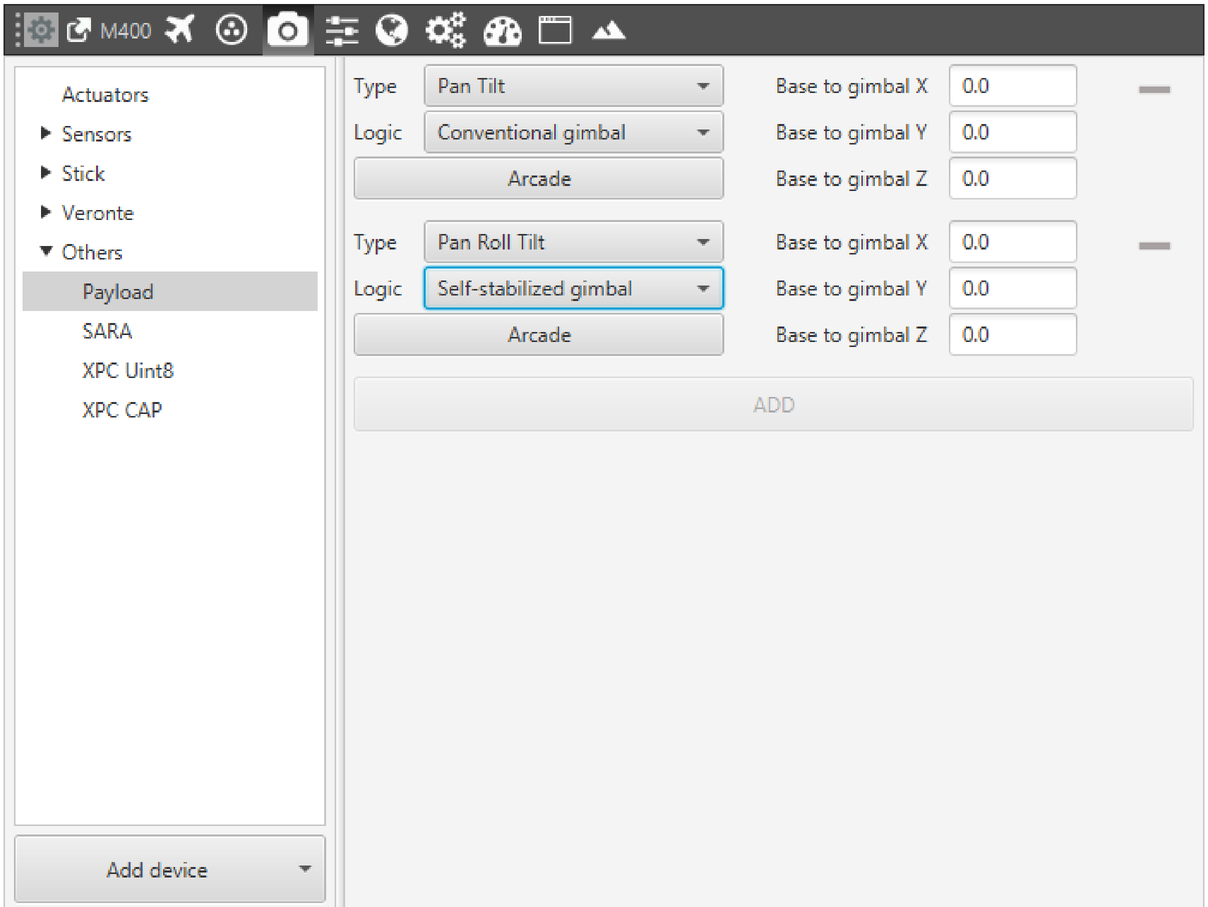
Note: If the YR matrix has not been properly defined, Y values will be greater than 1 but the actuator output will have the maximum S available.

A broader and more detailed explanation of the above can be found in *Stick*.

6.2.3.6 Others

6.2.3.6.1 Payload

This panel allows users to configure the gimbal settings, including the number of gimbal axis, servos used for gimbal connection and distance from the autopilot to the Gimbal.



Payload Configuration Menu

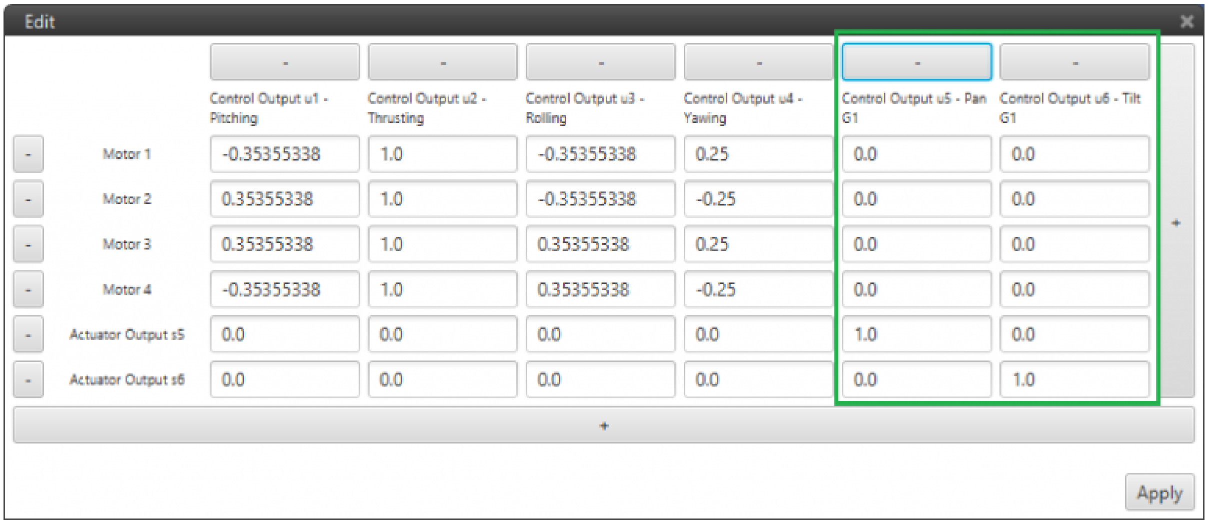
- **Type:** user can install a Pan -Tilt, Roll-Tilt or Pan-Roll-Tilt Gimbal, depending on the degrees of freedom of the Gimbal Device.
- **Logic:** Conventional gimbal or Self-stabilized gimbal.
- **Base to gimbal X, Y, Z:** relative position between the Gimbal and Veronte.

It is possible to configure two gimbals at the same time, press **ADD** to incorporate one into the Panel.

6.2.3.6.1.1 Payload Operation

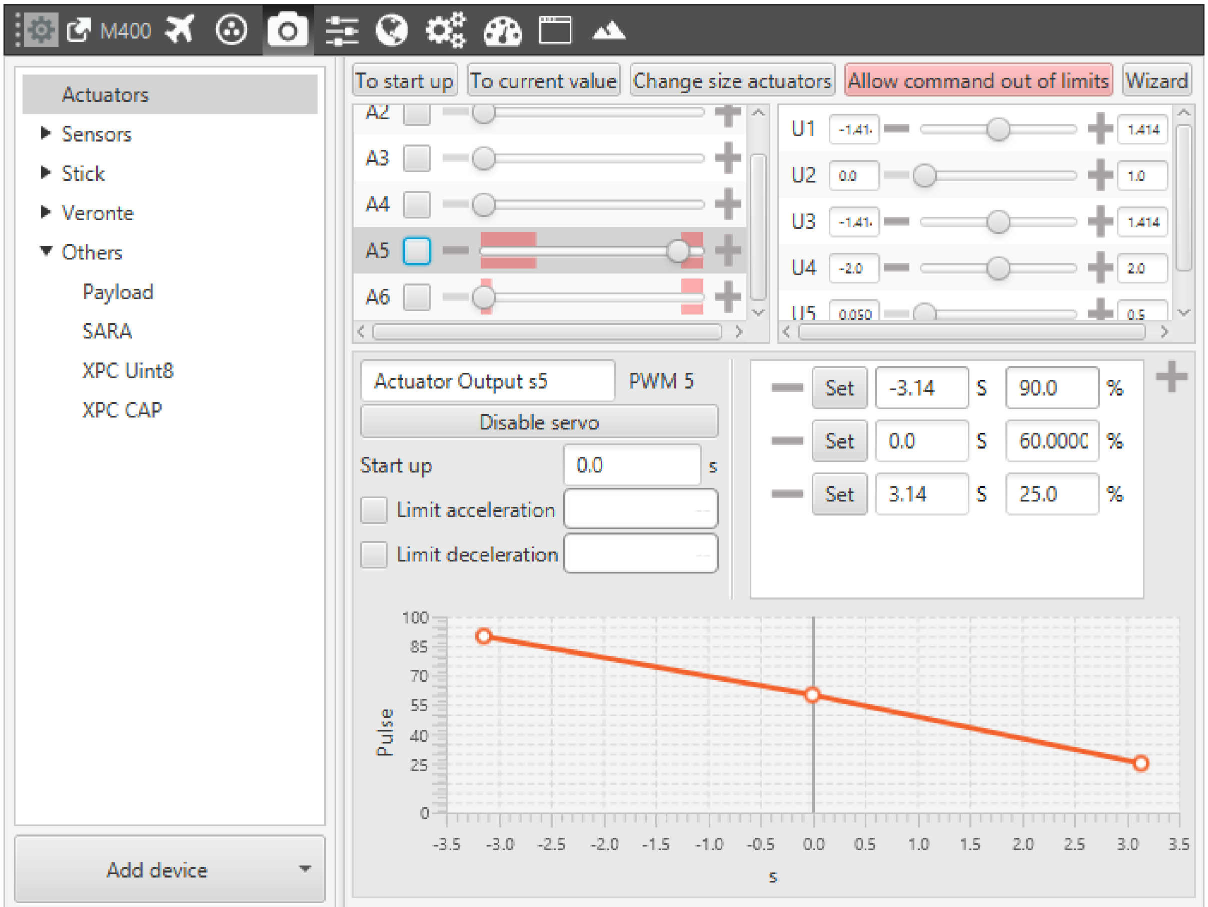
In order to configure a complete Payload Operation, the following steps have to be done. All steps are based on a Conventional Gimbal with two axes of rotation (Pan and Tilt).

1. Configure the US matrix selecting the control outputs (U) and actuator outputs (S). This has to be done according to the Gimbal used.



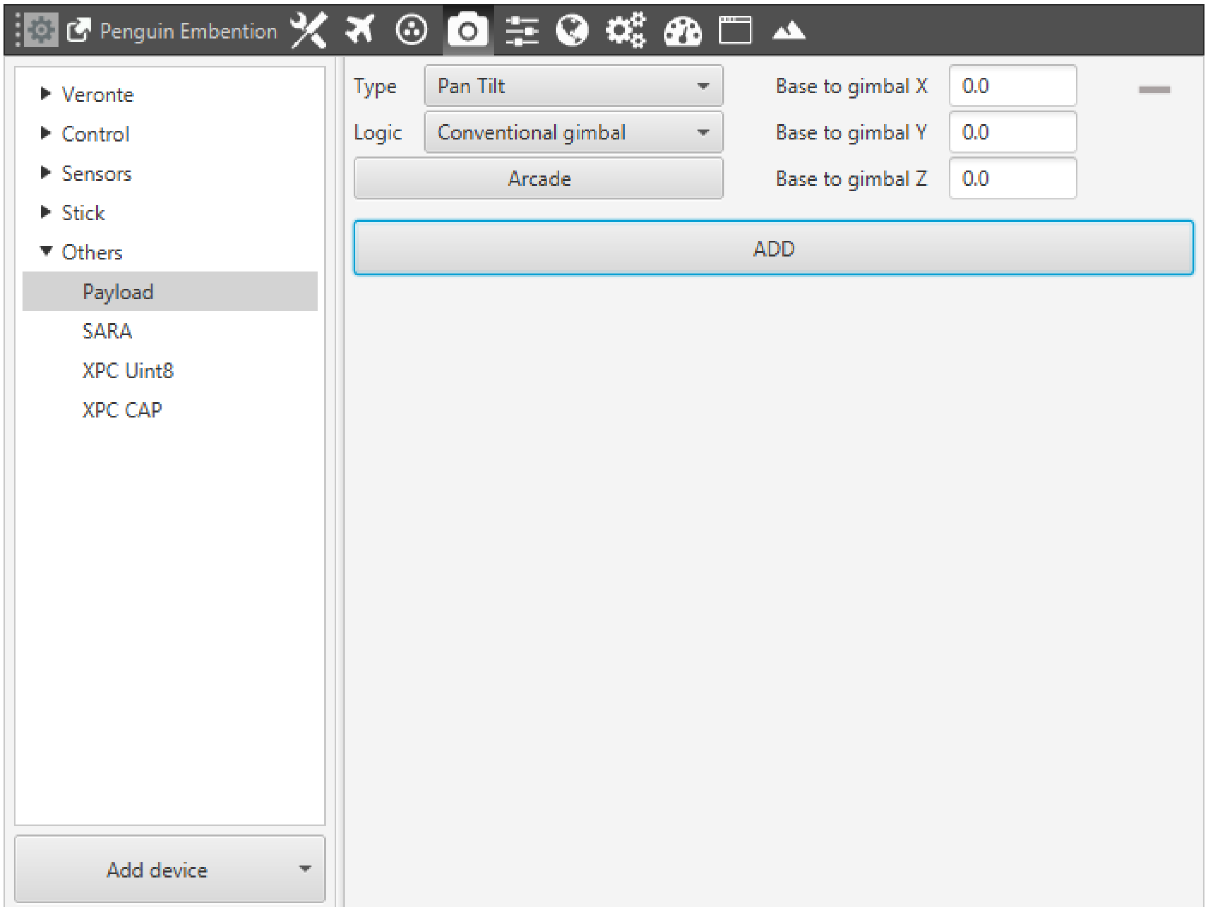
SU Matrix

2. Configure the relationship between PWM signals and actuator outputs (S), it depends on the device used, prior to configure this check performances. It may requiere to add PWM Outputs, in its correspondent panel, Connections.



SU Matrix

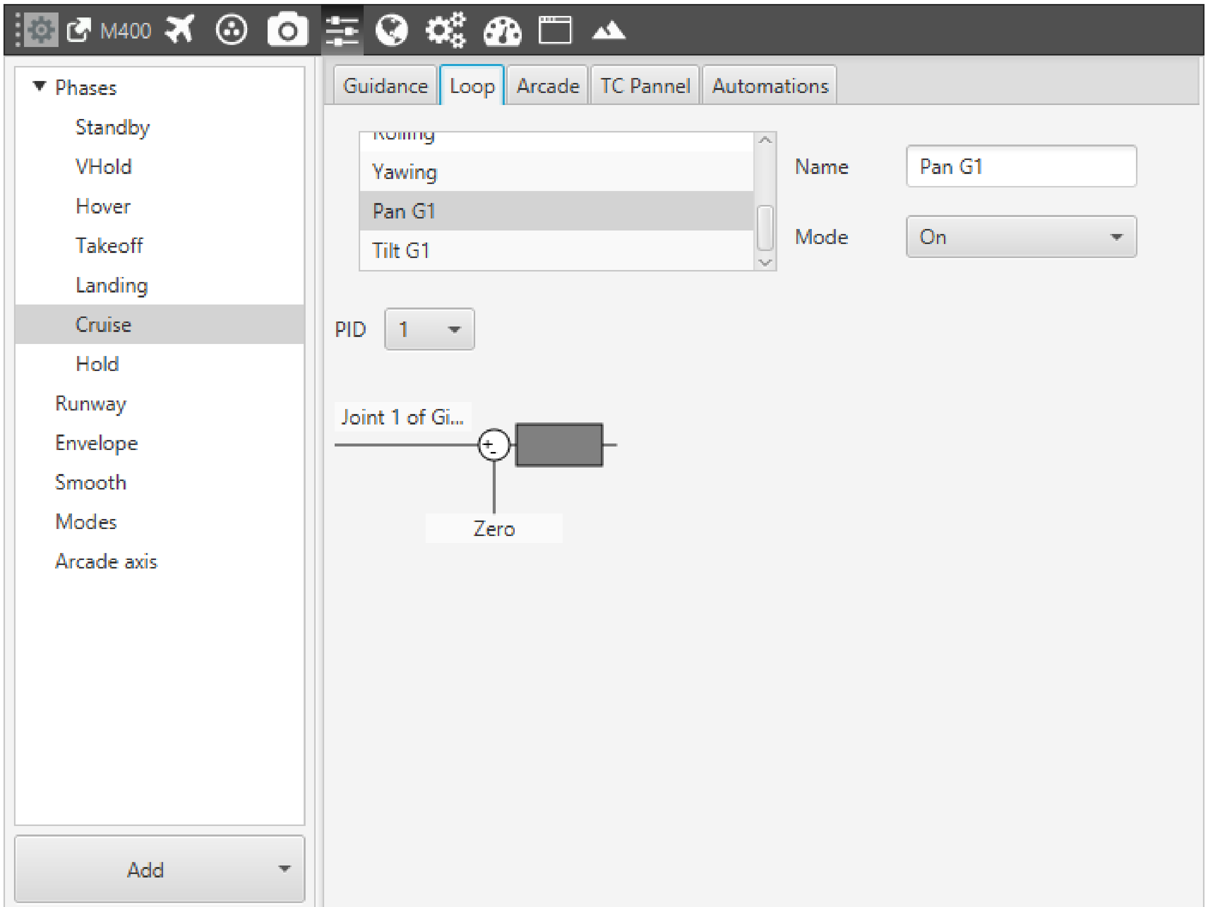
3. Configure the Payload Device.



Payload Configuration

4. Next step is the configuration of the control loops for each axe: Pan and Tilt. Control loops can be either close loop or open loop. It depends on if there is any variable available for feedback.

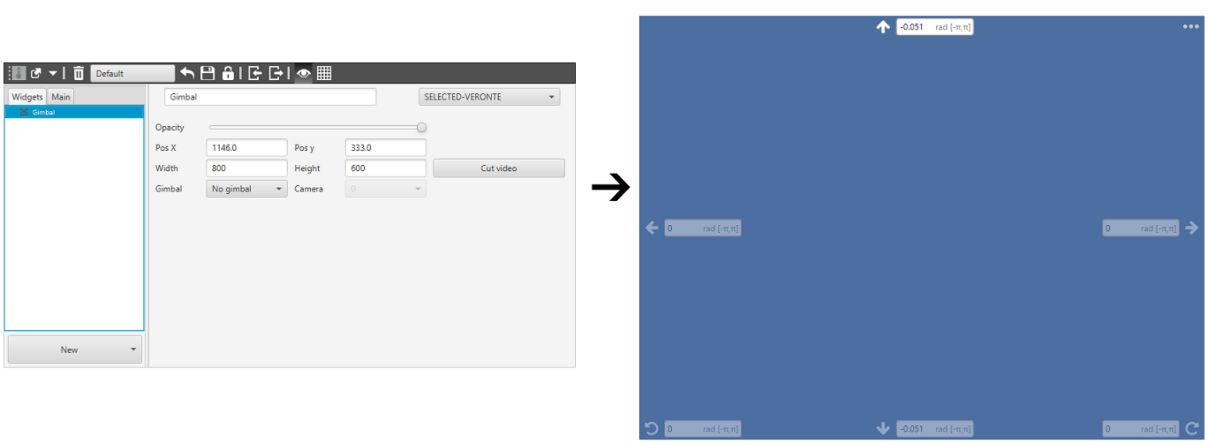
    - Pan control: select **Joint 1 of gimbal 1** as entry variable (desired Pan angle) and if exists a feedback variable select it, otherwise select the variable **Zero**, this sets a open loop.
    - Tilt control: select **Joint 2 of gimbal 1** as entry variable (desired Tilt angle) and if exists a feedback variable select it, otherwise select the variable **Zero**, this sets a open loop.



Control Loop

- A typical setting for the control loop incorpores: a proportional and integral gain.

5. The last step consists in configuring the widget that displays all the information related to the gimbal. Generally, this devices incorpores a Camera being possible to stream the signal in this widget. To configure this widget go to the Workspace Panel and add the widget of Gimbal. In this panel is possible to select a **Gimbal** and **Camera** if there is any device available.

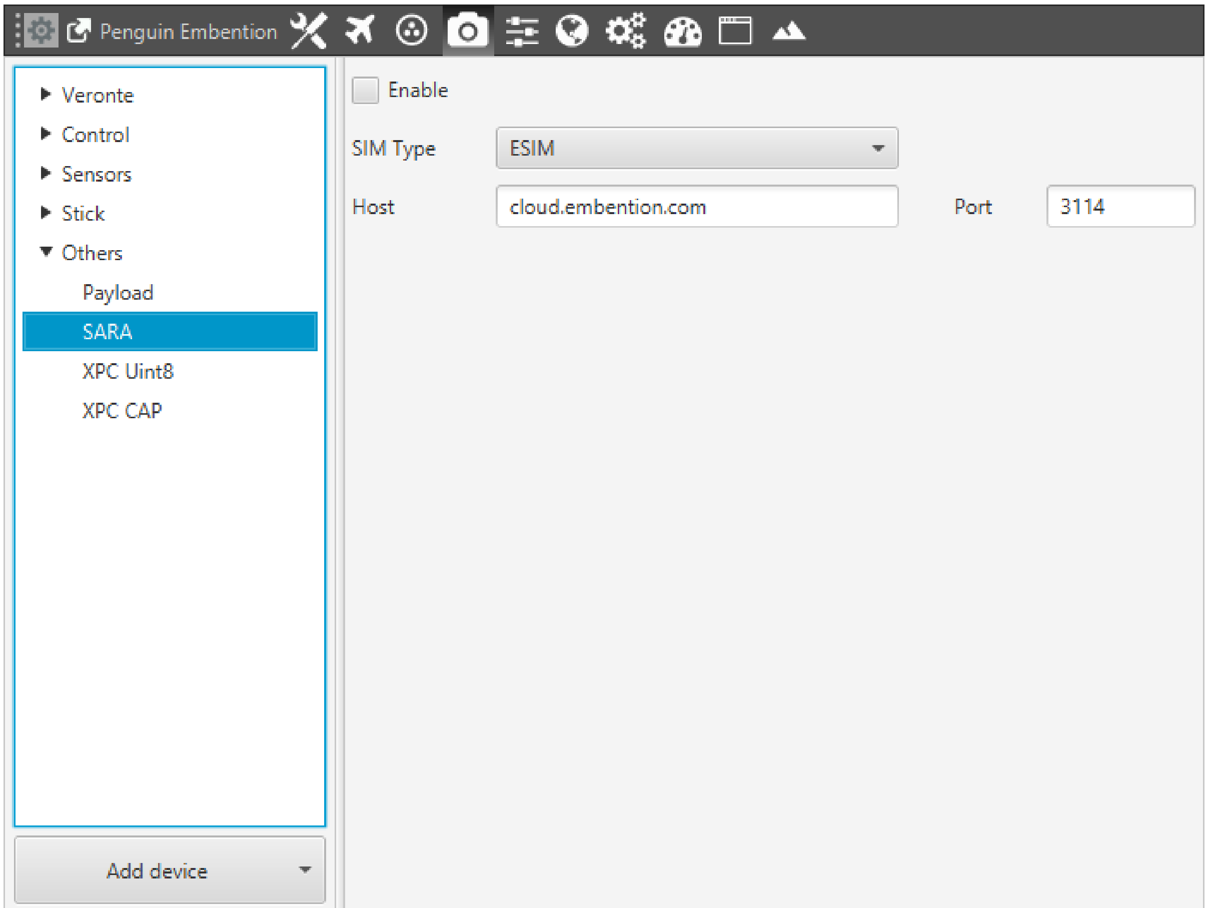


Widget Configuration

When configuring this widget the window above will be shown, this allows the user to control the gimbal. The movements are controlled with the arrows shown on the window, the user can press on them to move the gimbal or introduce a value.

6.2.3.6.2 SARA

In this panel it is possible to enable the 4G connection. This consisists in a step of configuration when using Cloud capabilities, this feature allows user to command Veronte Autopilots beyond the light of sight.



SARA Configuration Menu

- **SIM Type:** It is possible select either a ESIM or a SIM card.
- **Host:** Host Address.

Once configured, the user can access the cloud services. See section Users and Access Levels for more information.

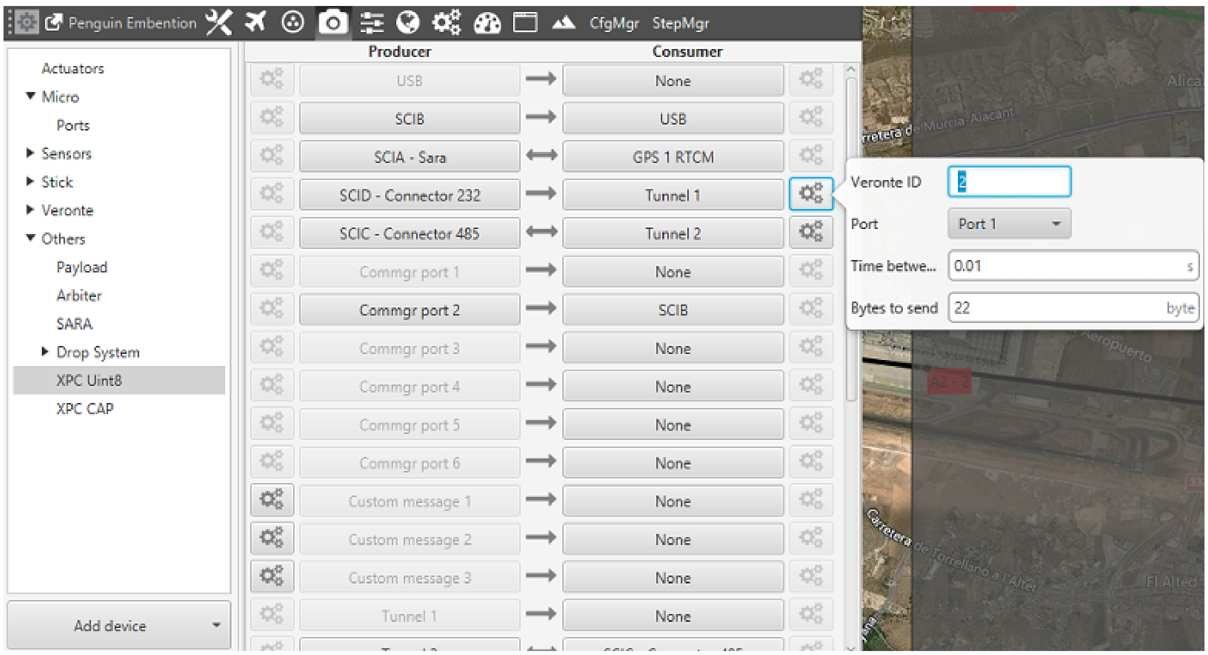
6.2.3.6.3 XPC Uint8

6.2.3.6.3.1 Tunnel

It is possible to configure a Tunnel which is a bidirectional bridge between Veronte Units that communicate to each other sharing information about an external device connected to the Serial or Digital port.

Imagine that we would like to have a button connected to the air autopilot to launch a parachute. It is not possible to physically connect the button because the air autopilot is in the flying platform, so we need a different option. Here is where the tunnel becomes useful. We could connect the button throught the Serial or Digital port to the Ground Autopilot, and then with the tunnel send the signal to the air one. With this configuration it would be like if the button were physically connected to the aircraft.

Let's consider the following image.



XPC Uint8 Configuration Menu

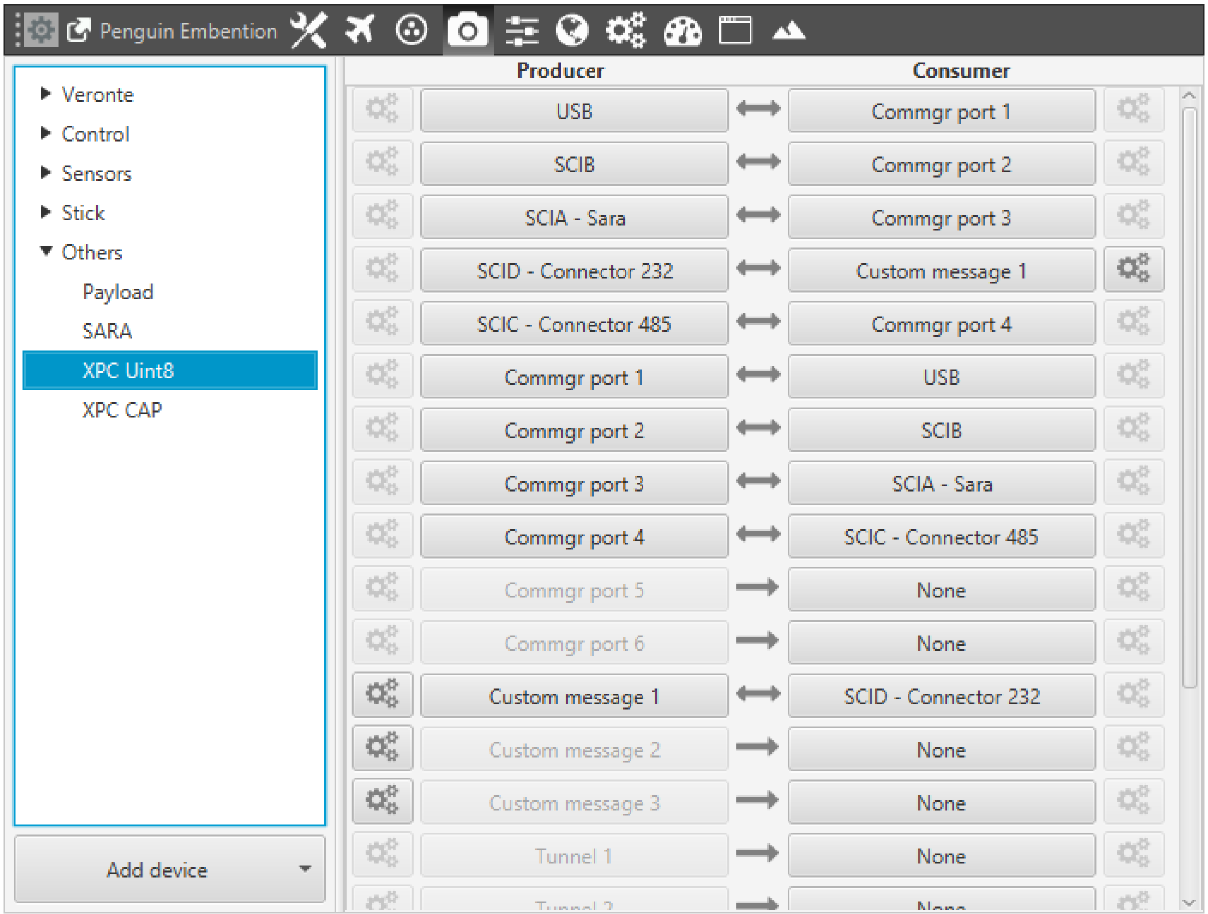
In the image above there is a device connected to the SCID Connector 232 (Producer) and there is a Tunnel (Consumer) which sends that information to other Veronte with a determined ID. On the other hand, Veronte Air has to be configured to receive the signal sent by other device. In that case the Producer will be Tunnel and Consumer will be the port where the device is connected.

The option available when configuring Tunnel as Consumer are:

- **Veronte ID:** identifies the Autopilot which will receive the information.
- **Port:** number of port is used to avoid mistakes and identify a Tunnel when using more than one.
- **Time between messages.**
- **Bytes to send:** sets the message size to send.

When configuring the unit that receives the information, only is necessary to configure the "Consumer" of that a information, generally a serial port.

In this panel the user can stablish the relationship between a determined signal with a I/O port. This allows users to configure an external sensors, messages between Veronte Units (Tunnel) and custom messages.



XPC Uint8 Configuration Menu

Firstly, users have to configure the **Producer** selecting the I/O port or information to use. Later, users have to configure the Consumer by clicking on an element, a new window will be displayed to select an item. The relationship between them can be unidirectional (Blind) or bidirectional (Blind Bidirectional), the last enables a port to receive or send information.

The following I/O ports are available:

Field	Description
USB	USB Port
SCIB	Radio
SCIA	4G Connection
SCID	Serial Port 232
SCIC	Serial Port 485
Commgr port	COM Manager Port

It is worth noting that only SCID/SCIC ports can be used to configure an element, the other ports have to be maintained by default.

Finally, it is possible to configure the next elements:

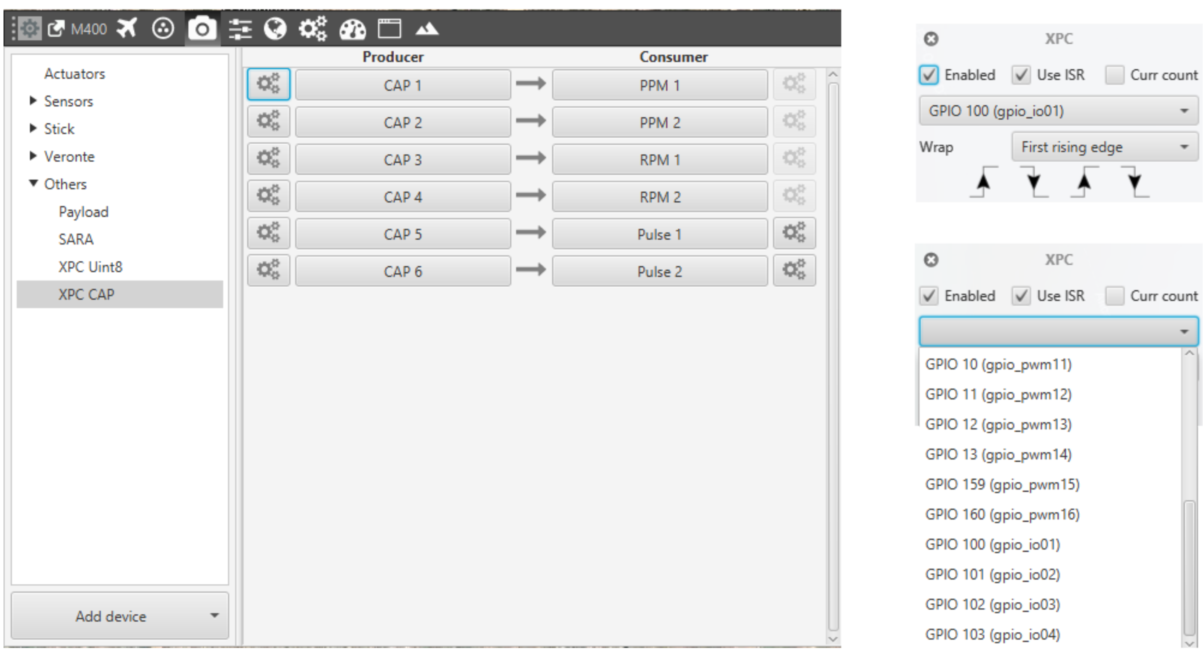
Field	Description
Tunnel	Creates a bidirectional brigde between two devices, see section *Tunnel*
Custom Message	This allows user to send/receiver a custom message.see section Custom Messages
GPS RTCM	Differential GPS.
Magneto	External magnetometer sensor, see section *Magnetometer*
Ultrasound	External ultrasound sensor, see section *Ultrasounds*
Splitter	Used to split a signal into 2.

More information about each element can be found in the links provided before.

6.2.3.6.4 XPC CAP

PWM outputs can work as Digital Input or Output, these pins are used to configure some custom sensors such as a Stick PPM, RPM Sensor or a Pulse Sensor. In addition to these pins, Enhanced Quadrature Encoder Pulse Inputs (EQEP) are also used, see section 2.1.2 for more information about Veronte I/O.

Sensors using a Digital In are configured in this menu.



XPC CAP Configuration Menu

The process to configure a device can be done as follows:

- Select a **Producer**.
- Click on a button to select the type of **Consumer**, it is possible to choose among a Stick PPM, RPM Sensor or Pulse Sensor.
- Configure the **Producer**, press on the configuration button and a new window will be displayed, see image above.
- In the new window is possible to select the pin where the device is connected. Pins available are PWM outputs, they are named as **gpio_pwmX**, and EQEP pins (GPIO 100, 101, 102, 103) these corresponds with pins 55, 56,

57 and 58, see section 2.1.2. When using the connector provided by Embention the Radio Digital In is connected to the pin 55 (GPIO 100) with pin 49 as Ground (see Ground Station)

- Wrap options allows to configure how the information is treated, for example if it will read the first rising edge or the falling. Clicking on the arrows it can be configured as desired.

Some sensors have their own configuration menu, visit them for more information: *Stick* and *RPM*.

This section of the manual contains the information about external sensors configuration. The following devices are available:

Field	Description
Payload	Gimbal configuration
SARA	4G Connection configuration
XPC-Uint8	In this panel users can configure how elements communicate each other, enabling to configure external sensors or messages.
XPC-CAP	This panel enables users to configure a joystick, pulse or rpm sensor.

Devices Panel permits to configure any device (payload, sensors…) connected to Veronte and the internal Veronte ones.

Type	Description
Actuators	To configure the actuators limits, their logic and saturation
Communications	4G, Comstats and Iridium configurations
Payload	Menus for different Cameras/Transponder configurations
Sensors	To set sensor device
Stick	To configure Stick
Veronte	To set the orientation and mass centre of the aircraft
Others	To set other devices and I/O signals

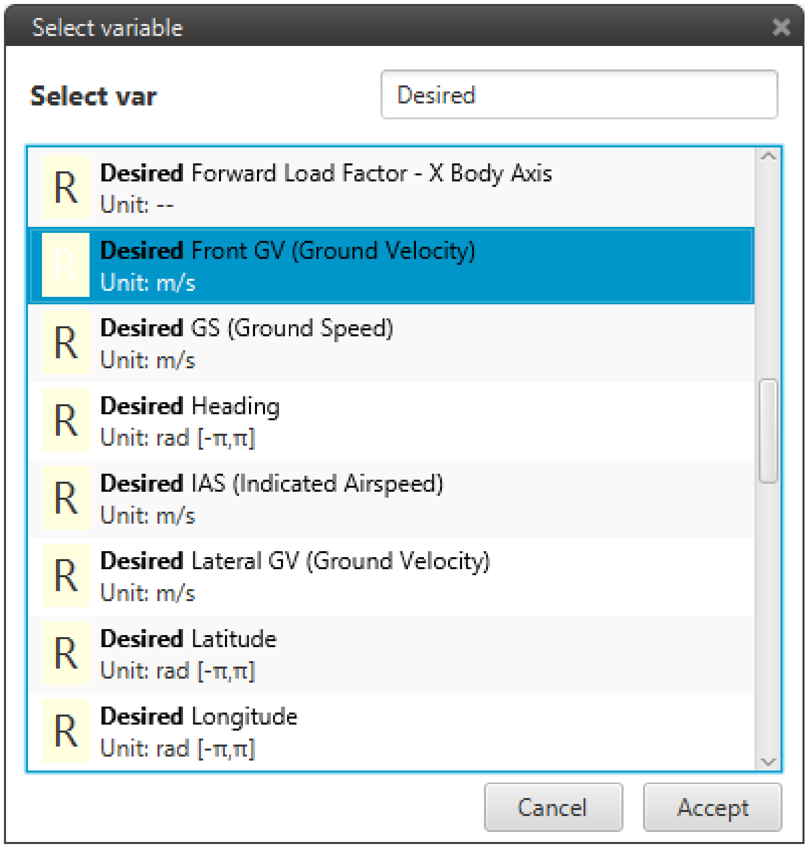
6.2.4 Control

6.2.4.1 Phases

6.2.4.1.1 Guidance

6.2.4.1.1.1 Guidance Variables

The guidances contained within Veronte generate a series of variables that are later used in the control system as the input of the PIDs. In the Hold guidance is the user the one that selects the desired variables to be generated. These variable can be added by clicking on the "+" button, there are different variables to select. Generally, variables named as Desired are used in this Guidance.

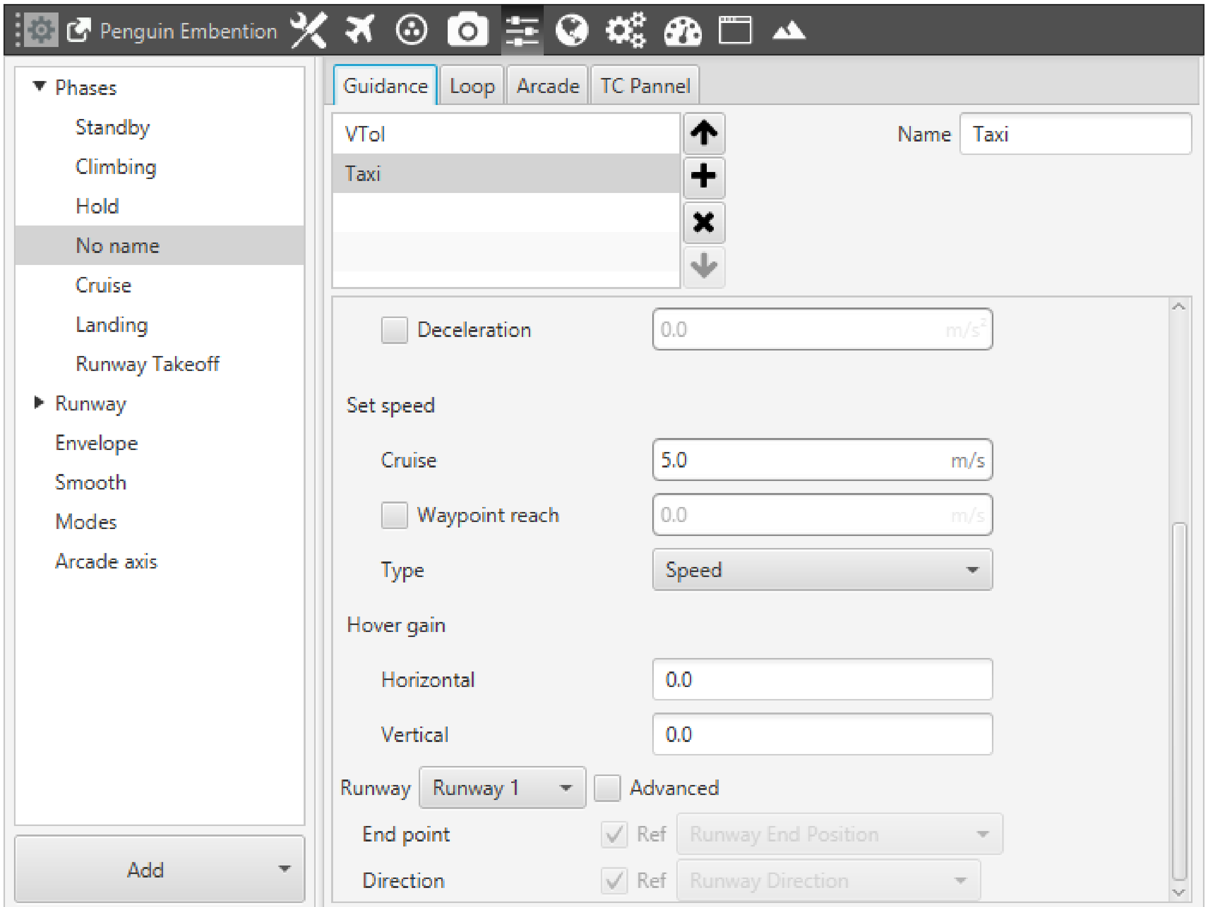


Variables Selection

After that these variables are used in the Control Loops.

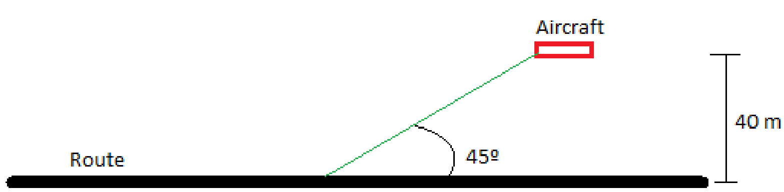
6.2.4.1.1.2 Taxi

Taxi guidance is used to create a linear path along the runway that is followed by the aircraft. This command is normally used in the take-off phase, where the airplane is wanted to keep the direction of the runway while is accelerating until the lift-off point.



TAXI Guidance

- **Line attraction:** is a parameter that determines how the platform is attracted to the line that determines the path that has to follow. The value is the distance perpendicular to the path at which the vehicle will try to go back to the line with a heading of 45 degrees with respect to it. When it is closer than this distance the heading angle is lower, and bigger when the distance is higher. So a small value will make the aircraft go to the line at high angles during more time (because the distance of 45 degrees is now small), and a big value implies less attraction because the platform will go to the line at smaller angles than 45 degrees from a bigger distance.



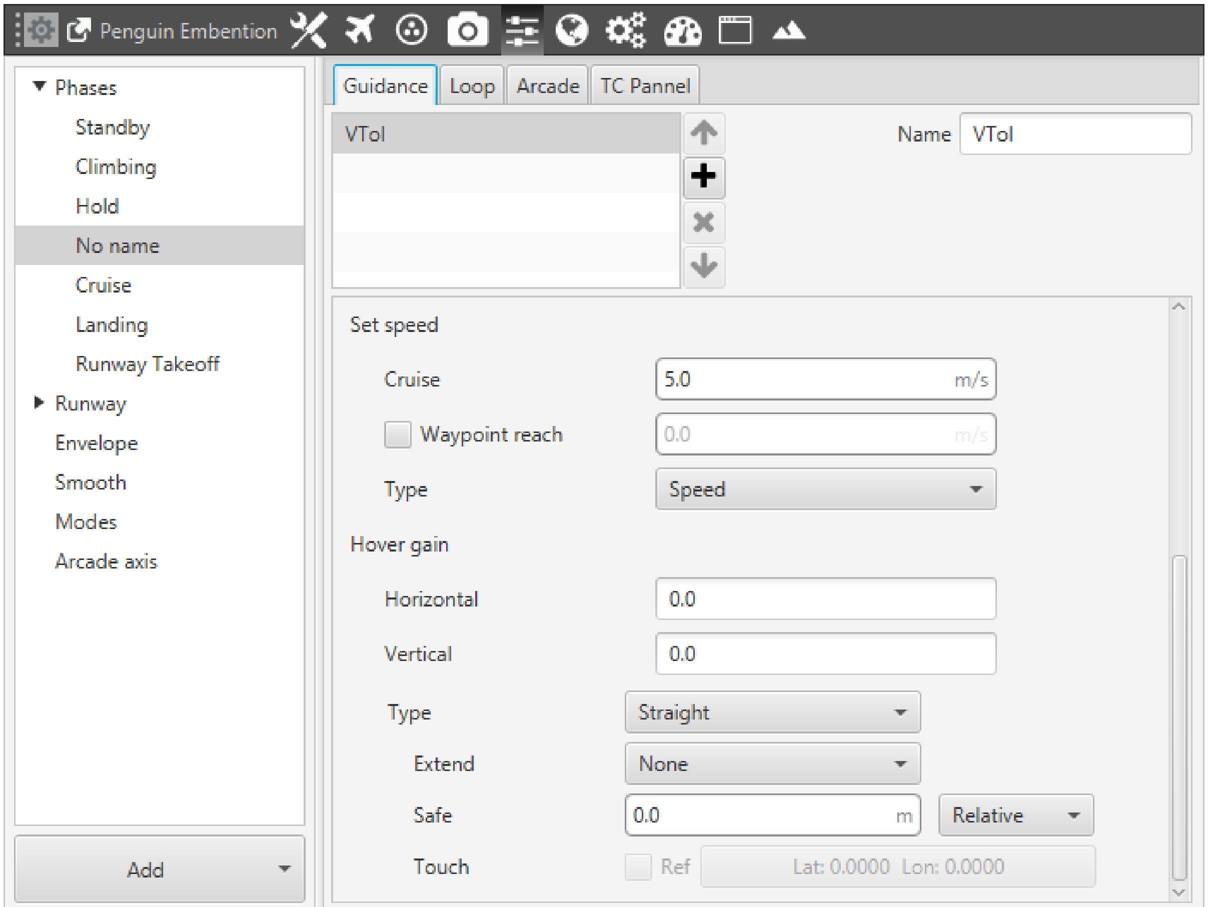
Angle and distance

In this phase the path is not directly indicated by the user as in the Cruise (which is defined in the Mission menu) but there is still a trajectory whose parameters are detalied later in this section, so this value is as important as it is for the Waypoint following route. The common values of the line attraction are between 20 and 40 for airplanes, and 15 for multicopters. This parameter only has to be changed by advanced users.

- **Acceleration proportional gain:** this parameters is releated with a new control system that Embention is developing in which elevator and thurst work side by side in both the pitching and thrusting.
- **Set height mode:** the height mode indicates how the aircraft will perform the route.
  - **2D mode:** if this mode it is selected, the platform will follow the predifined route without taking into account the altitude of the waypoints, it will keep the altitude that it has at the moment it enters in the cruise guidance.
  - **2.5D mode:** the vehicle goes from the altitude at which it enters in the mode, to the beginning of the route in a diagonal trajectory (it follows a 3D trajectory that connects the two points).
  - **3D mode:** is used in multicopters only, in this case the vehicle will climb vertically to the altitude of the first point of the route and then it will begin it.
- **Set limit acceleration:** establish a limit for acceleration and deceleration.
- **Set speed:** this option sets the speed that the vehicle will have during this phase. It can be **IAS** (indicated airspeed) or **Speed** (Ground Speed). Normally, IAS is used for airplanes and Speed for multicopters. The option **Waypoint reach** is used to indicate the speed at which the platform will reach the waypoints, so it will travel along the path with the speed indicated in the option **Cruise**, then it will decelerate or accelerate to the speed indicated in Waypoint Reach and then it will go back to the cruise speed.
- **Runway:** here it is seleceted a runway previously configured, see section *Runway*. Besides, it is possible to use the advanced mode and select a different end point or direction.

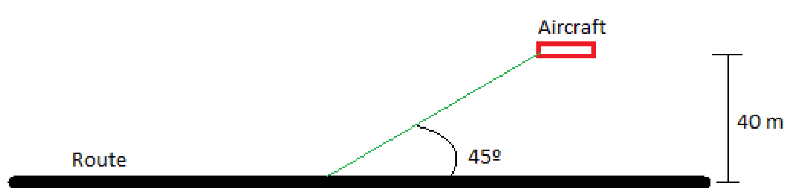
6.2.4.1.1.3 VTOL

VTOL guidance (vertical take-off and landing) is used in multicopters for the take-off and landing operations. This guidance consists on the creation of a vertical line that starts at the point where the platform enters in this guidance.



VTOL Guidance

- **Line Attraction:** is a parameter that determines how the platform is attracted to the line that determines the path that has to follow. The value is the distance perpendicular to the path at which the vehicle will try to go back to the line with a heading of 45 degrees with respect to it. When it is closer than this distance the heading angle is lower, and bigger when the distance is higher. So a small value will make the aircraft go to the line at high angles during more time (because the distance of 45 degrees is now small), and a big value implies less attraction because the platform will go to the line at smaller angles than 45 degrees from a bigger distance.



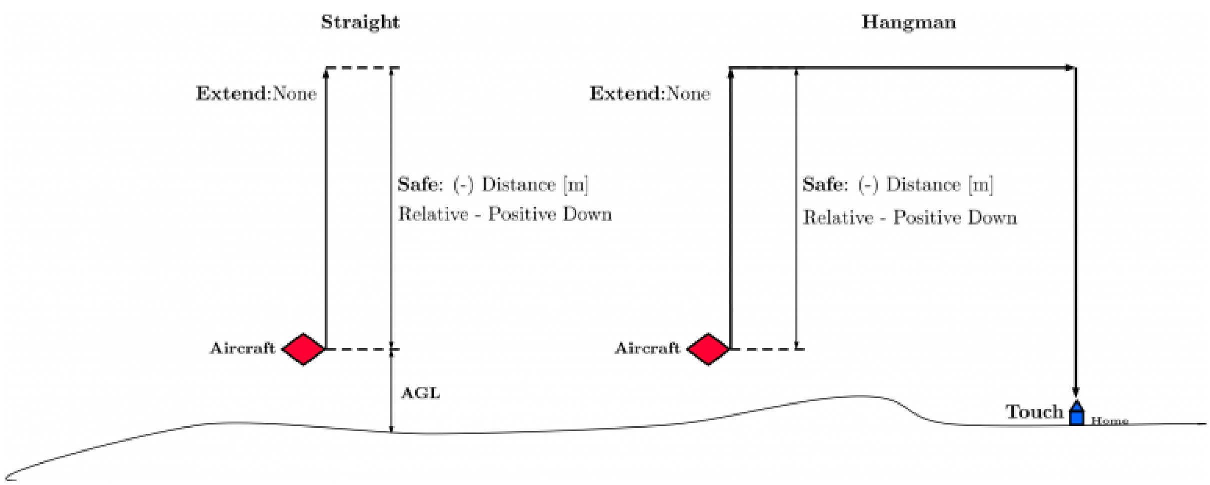
Angle and distance

In this phase the path is not directly indicated by the user as in the cruise (which is defined in the Mission menu) but there is still a trajectory whose parameters are detalied later in this section, so this value is as important as it is for the Waypoint following route. The common values of the line attraction are between 20 and 40 for airplanes, and 15 for multicopters. This parameter only has to be changed by advanced users.

- **Acceleration proportional gain:** this parameters is releated with a new control system that Embention is developing in which elevator and thurst work side by side in both the pitching and thrusting.
- **Set height mode:** the height mode indicates how the aircraft will perform the route.
  - **2D mode:** if this mode it is selected, the platform will follow the predifined route without taking into account the altitude of the waypoints, it will keep the altitude that it has at the moment it enters in the cruise guidance.
  - **2.5D mode:** the vehicle goes from the altitude at which it enters in this mode, to the beginning of the route in a diagonal trajectory (it follows a 3D trajectory that connects the two points).
  - **3D mode:** is used in multicopters only, in this case the vehicle will climb vertically to the altitude of the first point of the route and then it will begin it.
- **Set limit acceleration:** limit the values of the acceleration and deceleration during this guidance.
- **Set Speed:** this option sets the speed that the vehicle will have during this guidance. It could be IAS (indicated airspeed) or Speed (Ground Speed). Normally, the IAS is used for airplanes and the Speed for multicopters. The option Waypoint reach is used to indicate the speed at which the platform will reach the waypoints, so it will travel along the path with the speed indicated in the option Cruise, then it will decelerate or accelerate to the speed indicated in Waypoint Reach and then it will go back to the cruise speed.
- **Hover Gain:** those gains are used by a multicopter when it is hovering over a certain location. When the vehicle is moved from that hover point because of a wind gust, it will try to go back to that location using the gains specified here.
- **Type:** this parameters are used to indicate the route followed by the multicoter during the take-off and landing. The path Straight consists on a vertical line from the point where the vehicle enters in this phase. In the case of a take-off, the line goes from the ground to an altitude indicated by the user. The second option, Hangman, the path consists on a vertical and horizontal line.
- **Extend:** when Up or Down are selected, the value set in Safe will be discard, and the platform will ascend or descend, until a next change.
- **Safe:** this parameter defines the altitude the aircraft reach, it can be Relative (starting from the initial point of the route – current platform position) or an Absolute altitude (MSL,AGL or WGS84). As an example, in a take-off operation, an altitude of -10000 meters can be indicated as the final point of the route, so it is sure that the multicopter will keep climbing until another phase is commended (via automation or manually). The same procedure is done in the landing, indicating a big relative distance (for example 100 meters from the starting point) so it is sure that the vehicle reaches the ground, and an automation is set to stop the platform when it touches the surface.
- **Touch:** additional parameter to be configured when the type Hangman is selected. It defines a point that the aircraft has to reach. For instance, after go Up/Down the value set, the aircraft will perform an horizontal movement according to the point defined. Finally, when the aircraft is over the point, it will descend till reach that point. Usually, this option is used when there are obstacles in the area and performing this movement the platform can avoid them and landing safety.

Note: when the option relative is selected, a positive value will made the aircraft descend. Therefore, this value is Positive down.

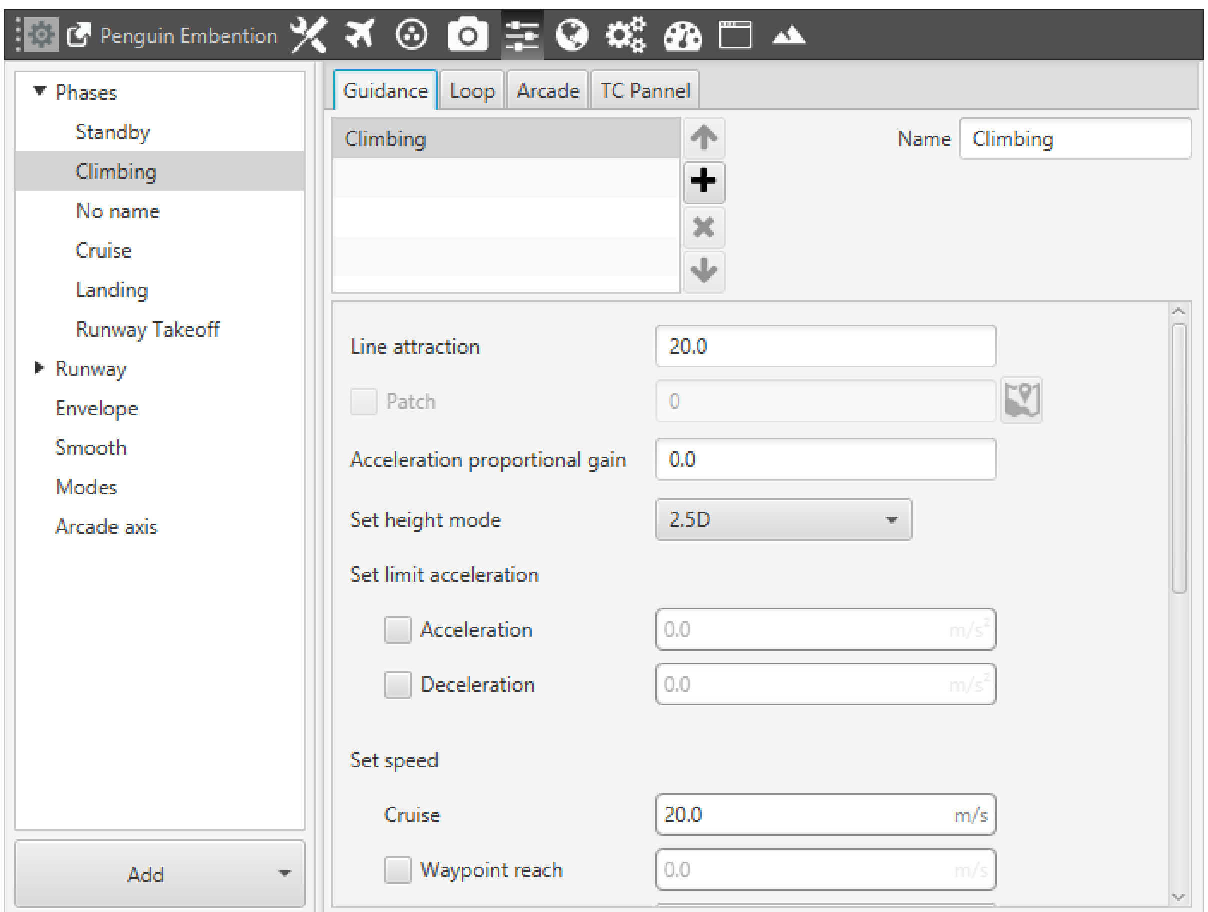
The following image gives an overview of some parameters introduced:



Parameters Overview

6.2.4.1.1.4 Climb

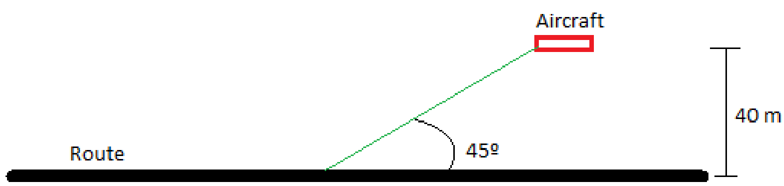
Climb guidance is used to make the aircraft climb from the start of the phase to another altitude. Commonly, this guidance is used after the take off to climb from the ground to the cruise altitude, but can be used for the proposit desired by the user.



Climbing Guidance

All the parameters that appear in the previous figure (and can be modified) are the ones that rule the climb of the aircraft. They are detailed as follows:

- **Line attraction:** is a parameter that determines how the platform is attracted to the line that determines the path that has to follow. The value is the distance perpendicular to the path at which the vehicle will try to go back to the line with a heading of 45 degrees with respect to it. When it is closer than this distance the heading angle is lower, and bigger when the distance is higher. So a small value will make the aircraft go to the line at high angles during more time (because the distance of 45 degrees is now small), and a big value implies less attraction because the platform will go to the line at smaller angles than 45 degrees from a bigger distance.



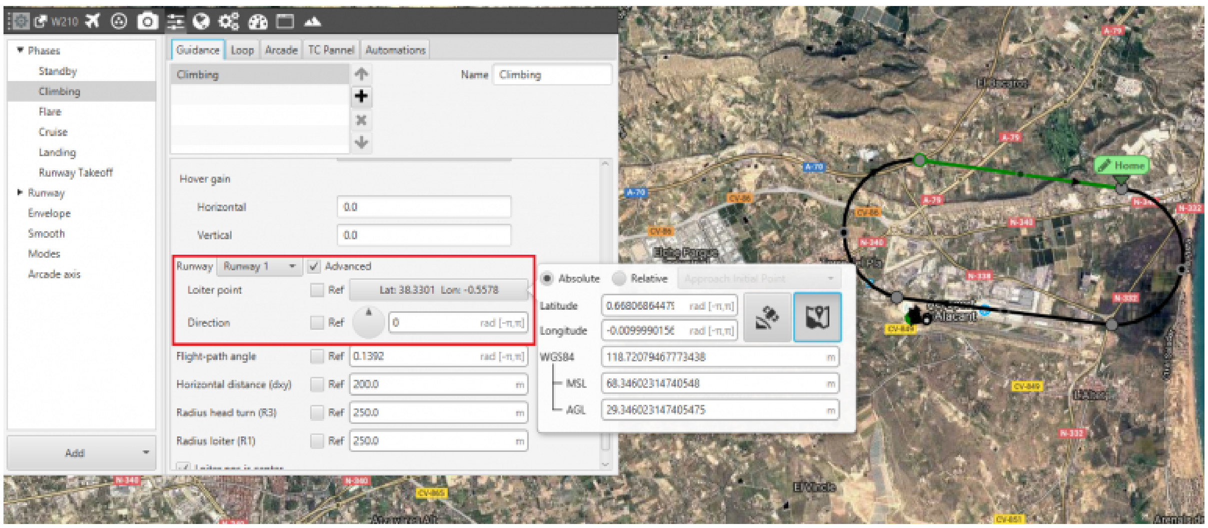
Angle and distance

In this phase the path is not directly indicated by the user as in the cruise (which is defined in the Mission menu) but there is still a trajectory whose parameters are detalied later in this section, so this value is as important as it is for the Waypoint following route. The common values of the line attraction are between 20 and 40 for airplanes, and 15 for multicopters. This parameter only has to be changed by advanced users.

- **Acceleration proportional gain:** this parameters is releated with a new control system that Embention is developing in which elevator and thurst work side by side in both the pitching and thrusting.
- **Set height mode:** the height mode indicates how the aircraft will perform the route.
  - **2D mode:** if this mode it is selected, the platform will follow the predifined route without taking into account the altitude of the waypoints, it will keep the altitude that it has at the moment it enters in the cruise guidance.
  - **2.5D mode:** the vehicle goes from the altitude at which it enters in this mode, to the beginning of the route in a diagonal trajectory (it follows a 3D trajectory that connects the two points).
  - **3D mode:** is used in multicopters only, in this case the vehicle will climb vertically to the altitude of the first point of the route and then it will begin it.
- **Set limit acceleration:** here are set the limits for deceleration and acceleration of the platform when it is climbing. Normally the limits set here are due to structural stability and to avoid extreme movements of the vehicle.
- **Set Speed:** this option sets the speed that the vehicle will have during the climb. It could be IAS (indicated airspeed) or Speed (Ground Speed). Normally, the IAS is used for airplanes and the Speed for multicopters. The option Waypoint reach is used to indicate the speed at which the platform will reach the waypoints, so it will travel along the path with the speed indicated in the option Cruise, then it will decelerate or accelerate to the speed indicated in Waypoint Reach and then it will go back to the cruise speed.
- **Hover Gain:** those gains are used by the multicopters when it is hovering over a certain location. When the vehicle is moved from that hover point because of a wind gust, it will try to go back to that location using the

gains specified here.

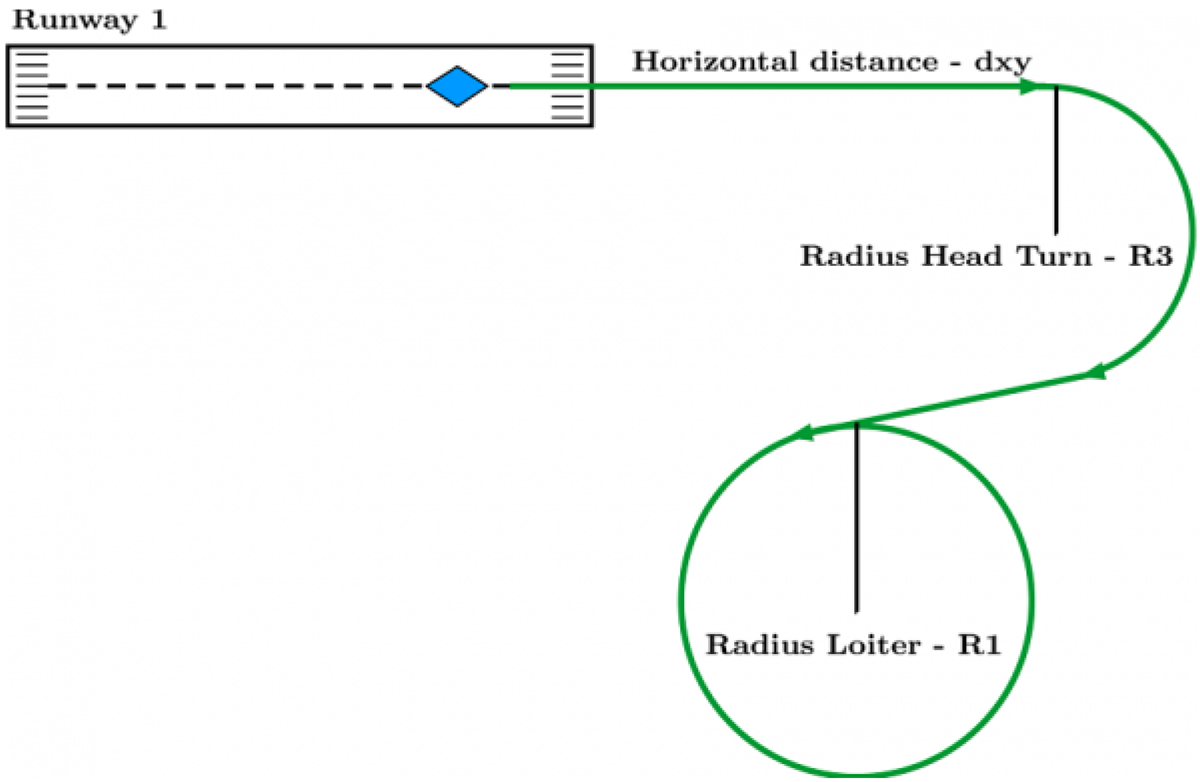
- **Route:** here are set the parameters that define the route followed by the aircraft when is climbing until a certain altitude. The climb is done in two segments. First a straight flight and then a circular route that allows the climb to be done in a reduced area.
  - **Runway:** here is selected the runway which previously has been edited, see section *Runway* It is possible to use the advanced mode and select a different loiter point (defines the altitude) or direction.
  - **Flight Path Angle:** angle at which the aircraft will climb.
  - **Horizontal Distance:** is the distance from the point where the aircraft enters in the phase which contains the climbing guidance, to the start of the circular climbing path.
  - **Radius Head Turn R3:** radius of the turn made to head the airplane towards the loiter direction.
  - **Radius loiter R1:** radius of the loiter ascending (helix) made by the aircraft to reach an altitude suitable.
- **Height:** this is determined by the **Loiter point**, by default, this value is definied with the runway (Runway Loiter). It can be changed when defining the runway or using **Advanced** options, see image below.



Climbing Guidance

As it can be seen, it possible to change the Loiter Point and Direction (Heading). If **Ref** is marked, it is possible to select any reference point/direction integrated into Veronte, by default these values are set to Runway Loiter and Runway Direction.

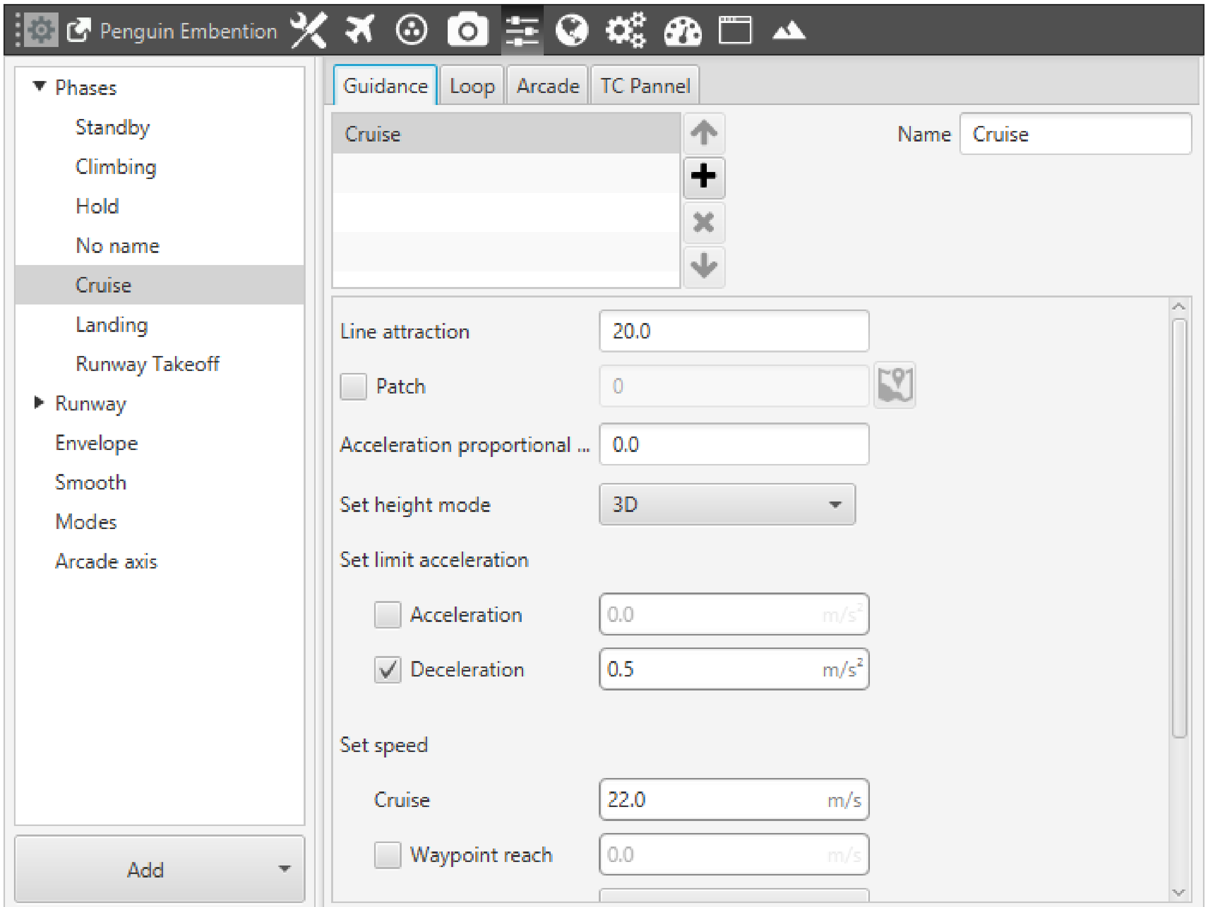
The following figure details schematically some parameters detailed previously.



Route Parameters

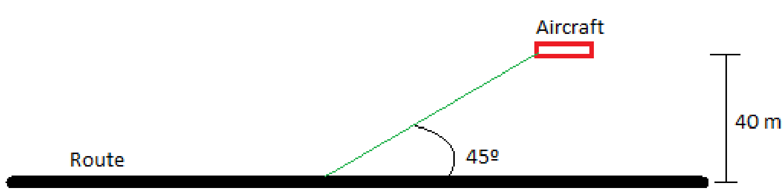
6.2.4.1.1.5 Cruise

This phase is used to make the aircraft follow a determined route created by the user in the Mission menu. This is the principal use of this guidance mode, but it can also be used to make the aircraft go to a certain location (waypoint) without indicating the full route, so in general is a guidance used to command a movement to be done by the platform. Some examples will be detailed later, by firstly all the parameters that define the cruise guidance are detailed.



Cruise Guidance

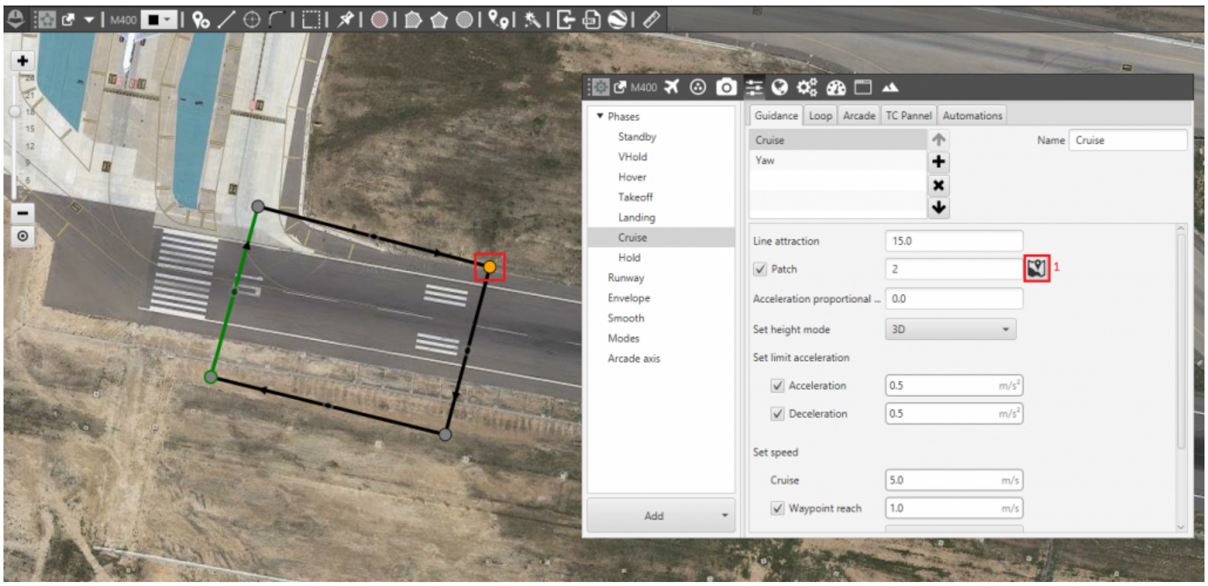
- **Line attraction**: is a parameter that determines how the platform is attracted to the line that determines the path that has to follow. The value is the distance perpendicular to the path at which the vehicle will try to go back to the line with a heading of 45 degrees with respect to it. When it is closer than this distance the heading angle is lower, and bigger when the distance is higher. So a small value will make the aircraft go to the line at high angles during more time (because the distance of 45 degrees is now small), and a big value implies less attraction because the platform will go to the line at smaller angles than 45 degrees from a bigger distance.



Angle and distance

In the this phase the path is not directly indicated by the user as in the cruise (which is defined in the Mission menu) but there is still a trajectory whose parameters are detalied later in this section, so this value is as important as it is for the Waypoint following route. The common values of the line attraction are between 20 and 40 for airplanes, and 15 for multicopters. This parameter only has to be changed by advanced users.

- **Patch:** in Pipe, a path is defined as the line between two waypoints. This option is used to force the aircraft to go to a certain patch when entering in the cruise phase. For example, it the Mission created by the used has 4 lines (a square), this option could be used to force the aircraft to take one of the lines (path) as the first line to cover during its operation.



Patch

In the previous figure, if the line that starts at the yellow waypoint wants to be taken as the first one of the route, clicking on 1 will allow the user to select that patch (a patch is selected by choosing its first waypoint, taking into account the direction). The numeration of the patchs is related with the order in which they were created. If the waypoint selected in the “Patch” option is alone (is not in a line), the aircraft will go to this point continuosly, so it will loiter around it. To sum up, if the path selected is part of a route with more than one line, the platform will first go to this line and the it will continue with the route. On the other hand, if the path selected is a single point, the vehicle will loiter around it (or hover in the case of a multicopter).

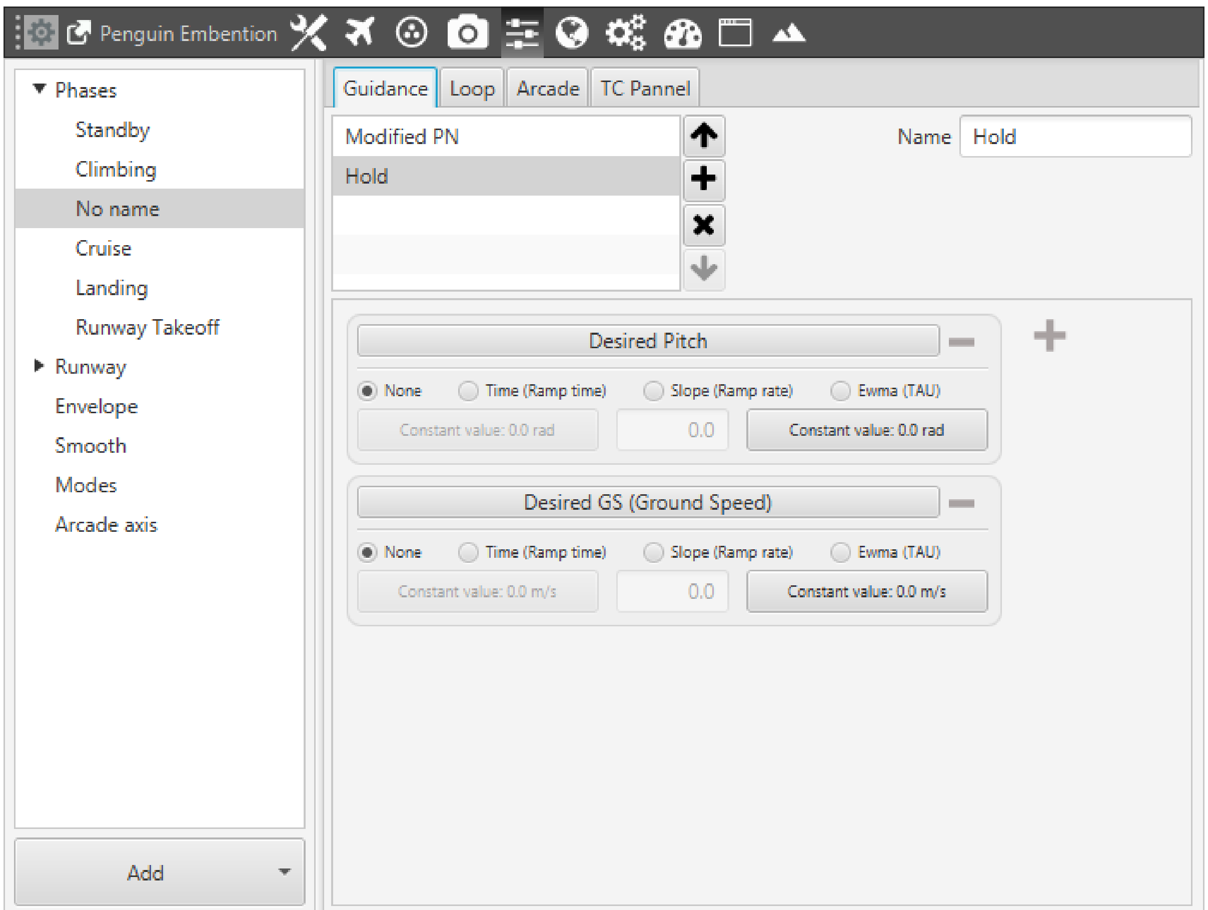
- **Acceleration proportional gain:** this parameters is releated with a new control system that Embention is developing in which elevator and thurst work side by side in both the pitching and thrusting.
- **Set height mode:** the height mode indicates how the aircraft will perform the route.
  - **2D mode:** if this mode it is selected, the platform will follow the predifined route without taking into account the altitude of the waypoints, it will keep the altitude that it has at the moment it enters in the cruise guidance.
  - **2.5D mode:** the vehicle goes from the altitude at which it enters in this mode, to the beginning of the route in a diagonal trajectory (it follows a 3D trajectory that connects the two points).
  - **3D mode:** is used in multicopters only, in this case the vehicle will climb vertically to the altitude of the first point of the route and then it will begin it.
- **Set limit acceleration:** the acceleration and deceleration can be limited to increase the safety of the operation.
- **Set speed:** this option sets the speed that the vehicle will have during the climb. It could be **IAS** (indicated airspeed) or **Speed** (Ground Speed). Normally, the IAS is used for airplanes and the Speed for multicopters. The

option **Waypoint reach** is used to indicate the speed at which the platform will reach the waypoints, so it will travel along the path with the speed indicated in the option **Cruise**, then it will decelerate or accelerate to the speed indicated in Waypoint Reach and then it will go back to the cruise speed.

- **Hover Gain:** those gains are used by the multicopters when it is hovering over a certain location. When the vehicle is moved from that hover point because of a wind gust, it will try to go back to that location using the gains specified here.

6.2.4.1.1.6 Hold

Hold guidance is used to introduce in the control system a desired variable with a certain value especified by the user. It is different than the other guidances, where for example the user determines a path to follow and the desired heading that makes the aircraft follow that path is not determined by the user, is determined by the path.



Hold Guidance

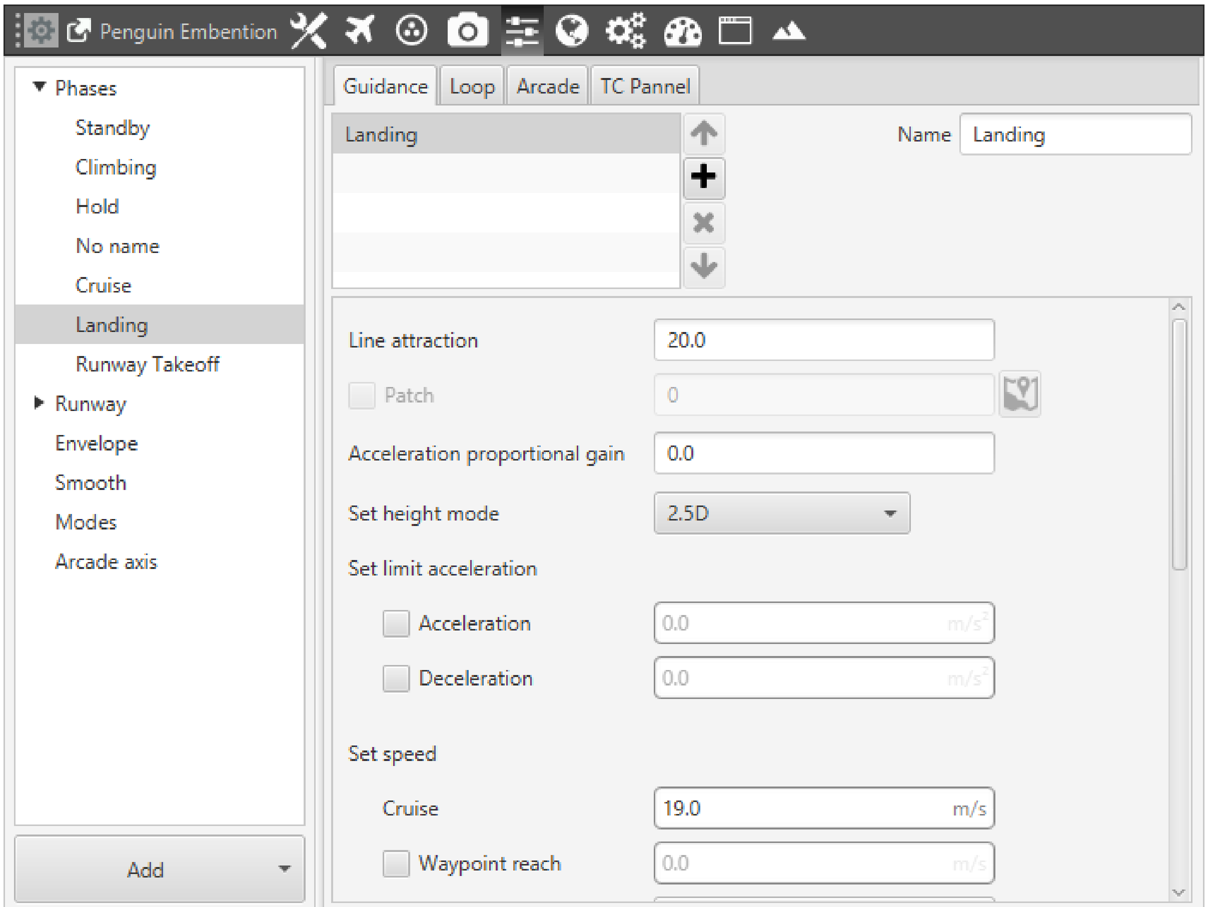
There are 4 ways in Pipe to determine how the variable is introduced in the controller in a Hold Guidance.

- **None:** a simple value is specified. As soon as the aircraft enters in the phase that has this hold guidance, the desired variable will take the value specified in this option. It is possible to define a constant value for the desired variable, or take the value that another variable has at the instant when the aircraft enters in that phase. For example, to make the aircraft keep the heading that it has when changing to a phase, select Desired Heading as the hold variable, and choosing heading.
- **Time (Ramp Time):** in this case, the desired variable introduced in the controller will take first the value (or the variable) introduced. After a certain time established, the variable reached the value (or variable) indicated.

- **Slope (Ramp Rate):** the same case as before, but know instead of the time between the two values of the ramp, the slope of that ramp can be indicated.
- **Ewma (TAU):** the variation between the start and final points is exponential.

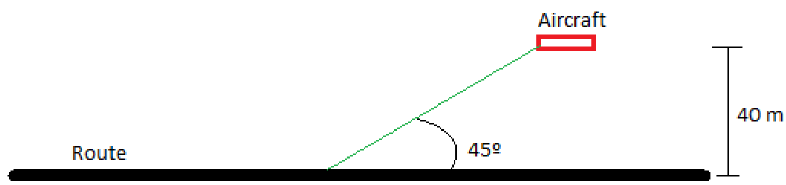
6.2.4.1.1.7 Landing

Landing guidance is used to create the route that the airplane will follow to land in a certain runway (defined by the user).



Landing Guidance

- **Line attraction:** as in the case of the climbing, is a parameter that determines how the aircraft is attracted to the line that it has to follow. The value is the distance perpendicular to the path at which the vehicle will try to go back to the line with a heading of 45 degrees with respect to it. When it is closer than this distance the heading angle is lower, and bigger when the distance is higher. So a small value will make the aircraft go to the line at high angles during more time (because the distance of 45 degrees is now small), and a big value implies less attraction because the platform will go to the line at smaller angles than 45 degrees from a bigger distance.



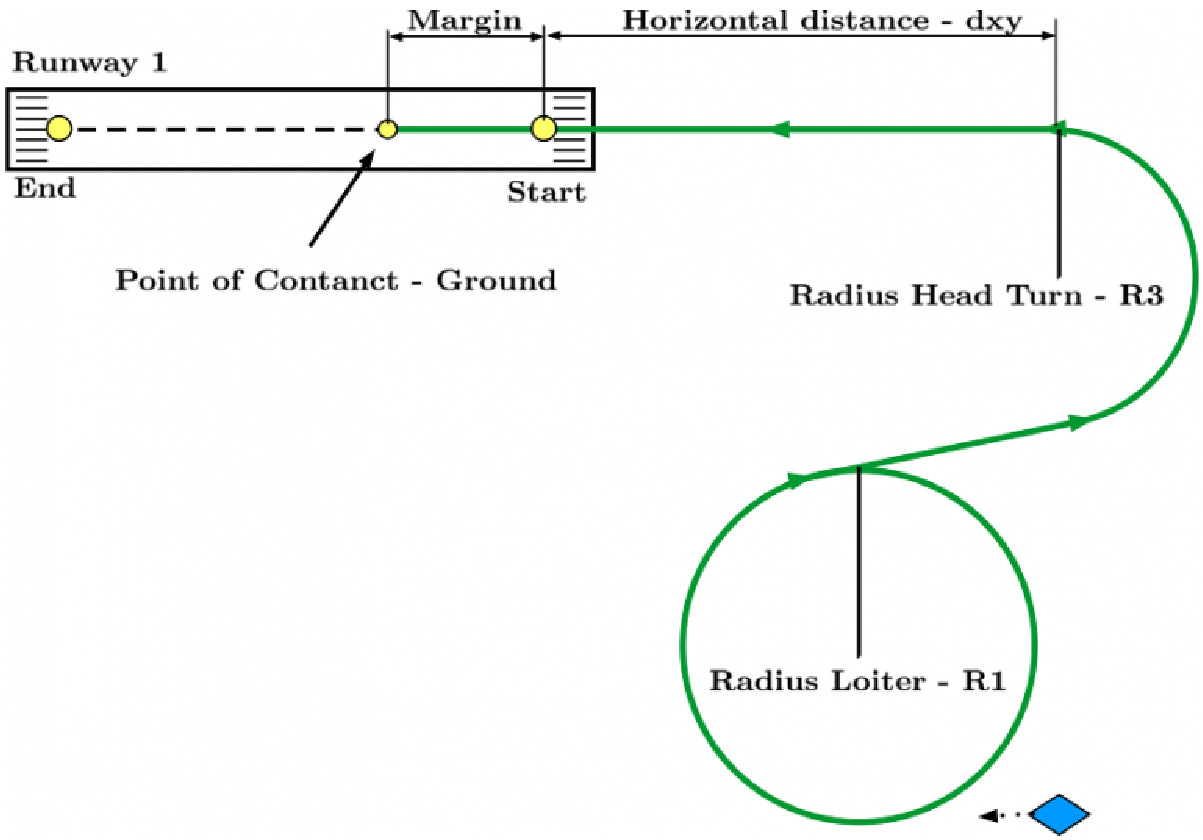
Angle and distance

In the this phase the path is not directly indicated by the user as in the cruise (which is defined in the Mission menu) but there is still a trajectory whose parameters are detalied later in this section, so this value is as important as it is for the Waypoint following route. The common values of the line attraction are between 20 and 40 for airplanes, and 15 for multicopters. This parameter only has to be changed by advanced users.

- **Acceleration proportional gain:** this parameters is releated with a new control system that Embention is developing in which elevator and thurst work side by side in both the pitching and thrusting.
- **Set height mode:** the height mode indicates how the aircraft will perform the route.
  - **2D mode:** if this mode it is selected, the platform will follow the predifined route without taking into account the altitude of the waypoints, it will keep the altitude that it has at the moment it enters in the cruise guidance.
  - **2.5D mode:** is used in multicopters only, in this case the vehicle will climb vertically to the altitude of the first point of the route and then it will begin it.
  - **3D mode:** the vehicle goes from the altitude at which it enters in cruise mode, to the beginning of the route in a diagonal trajectory (it follows a 3D trajectory that connects the two points).
- **Set speed:** this option sets the speed that the vehicle will have during the climb. It can be IAS (indicated airspeed) or Speed (Ground Speed). Normally, IAS is used for airplanes and Speed for multicopters. The option Waypoint reach is used to indicate the speed at which the platform will reach the waypoints, so it will travel along the path with the speed indicated in the option Cruise, then it will decelerate or accelerate to the speed indicated in Waypoint Reach and then it will go back to the cruise speed.
- **Route:** the route section of the landing guidance defines the route that the aircraft takes to go from the point when this phase is commanded, to the point where it touches the ground. The landing route has two parts: the first one consists on a descending loiter used to descend from the cruise altitude to an altitude where the manoeuvre of heading the airplane towards the runway direction and make it touch ground can be performed, the second part is the path that takes the aircraft to ground (to the runway). If the airplane starts the landing phase at an altitude at which it can perform the landing manoeuvre without the descending loiter, this one will not be performed, so the descending circular route will automatically be done when the aircraft is at a great height. The parameters that define the landing route are presented as follows:
  - **Loi Force:** force aircraft to perform loiter manoeuvre.
  - **Runway:** select a runway o spot previously defined, see section ref 6.7.2 . Besides, it is possible to use the advanced mode and select a different loiter/touch point and direction.
  - **Loiter Pos:** center point of the loiter descending route.

- **Flight Path Angle:** angle at which the aircraft will descend.
- **Horizontal Distance:** is the distance from the point where the aircraft is alligend with the runway to the start of the it.
- **Radius Head Turn R3:** radius of the turn made to head the airplane towards the runway direction.
- **Radius loiter R1:** radius of the loiter descending made by the aircraft to reach an altitude suitable to perform the landing manoeuvre.

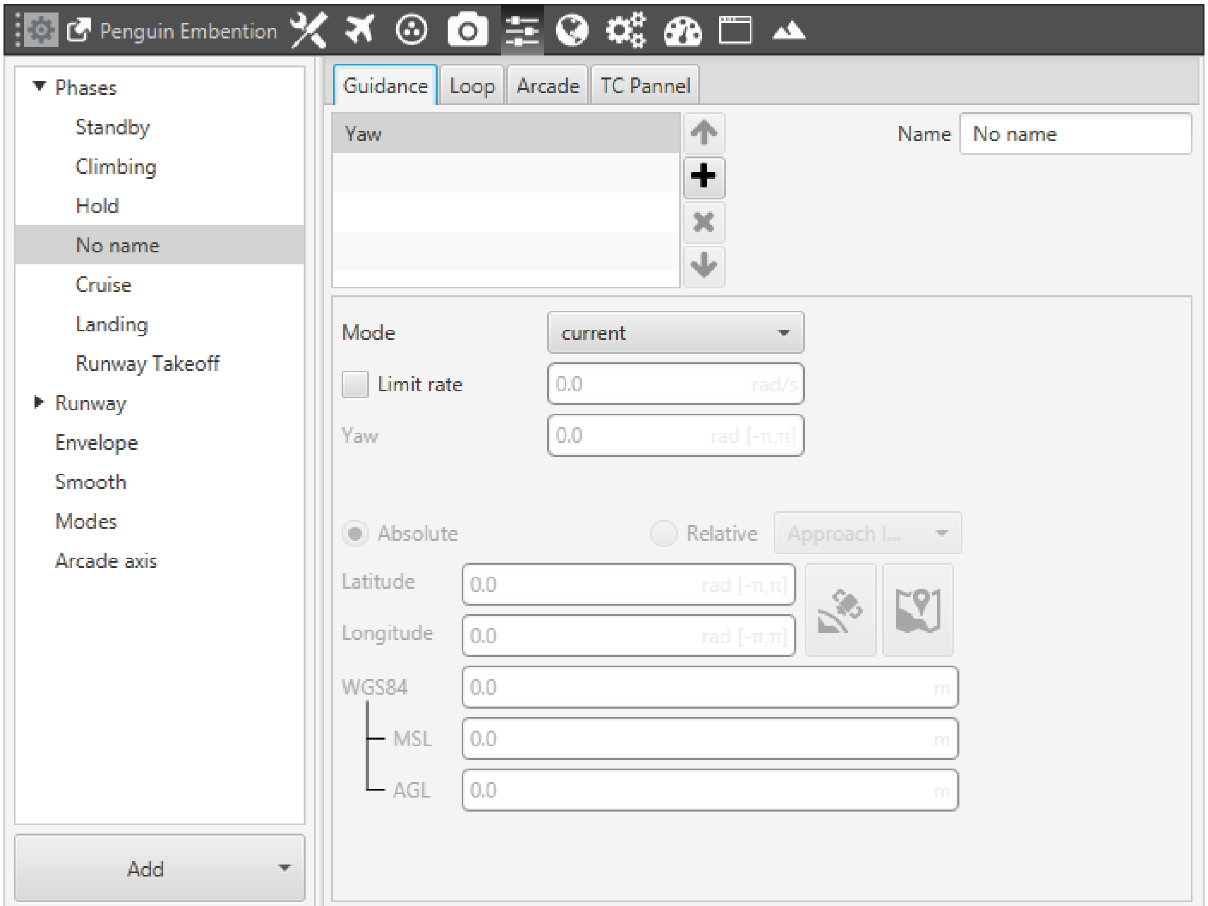
The following figure details schematically all the parameters detailed previously.



Landing Route

6.2.4.1.1.8 Yaw

Yaw guidance is used in multicopters to indicate the behaviour of the platform in the yaw axis. This option is normally used during the cruise phase of the multicopters, because the route can be carried out with the aircraft without rotating in the yaw axis, or rotate it to point its longitudinal axis parelel to the path.



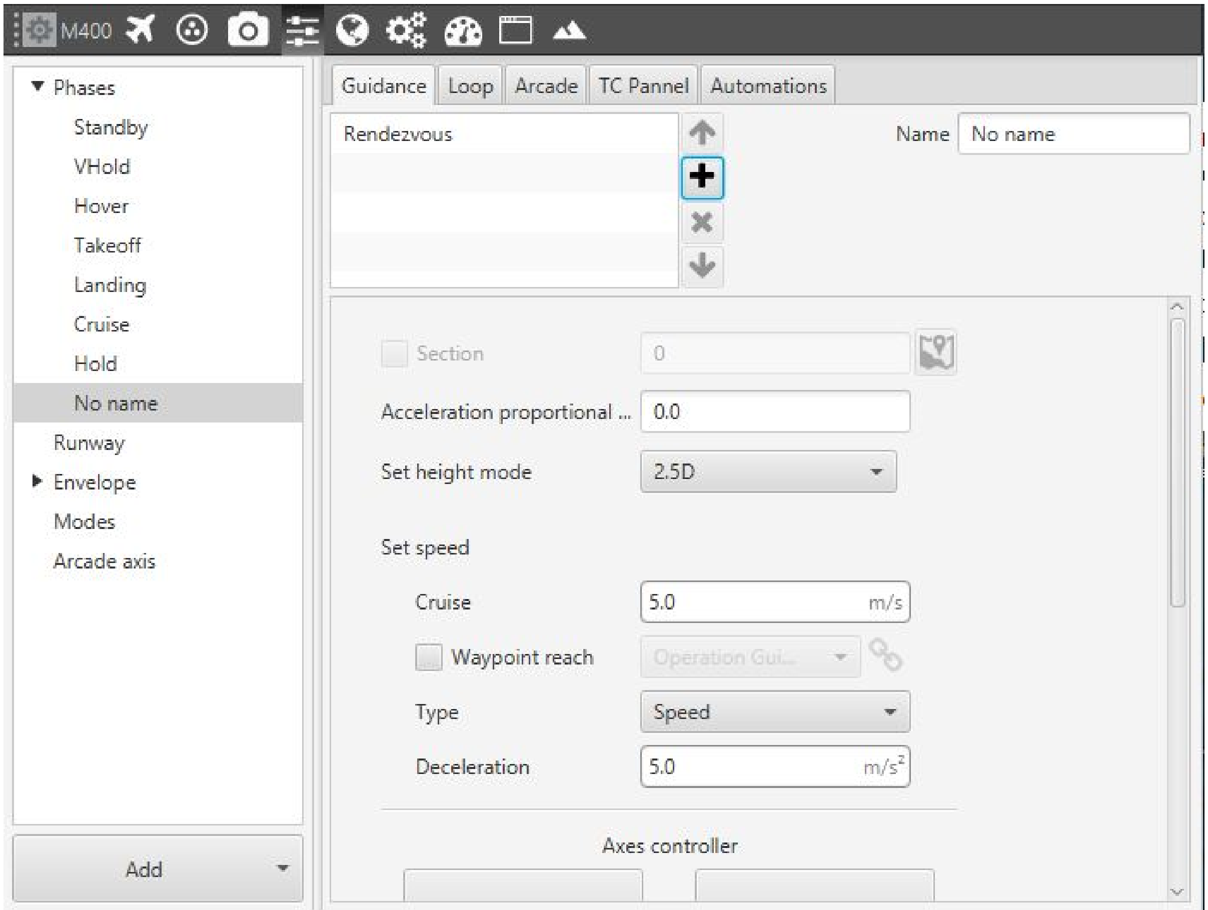
Yaw Guidance

The modes available in the yaw guidance are:

- **Current:** the multicopter will keep the yaw angle it has when entering in the phase that contains this guidance. Desired Yaw = Current Yaw
- **Fixed:** the yaw is kept at a constant value indicated by the user.
- **Heading:** Heading represents velocity vector direction and, when it is very small, its estimation is more complex and direction is changing constantly. Because of this, the approximation Yaw = Heading is introduced when the estimated velocity is near 0.
- **Position:** the yaw of the multicopter will be rotated so its longitudinal axis is always focusing a point defined by the user in the menu shown in the previous figure (absolute or relative).

6.2.4.1.1.9 Rendezvous

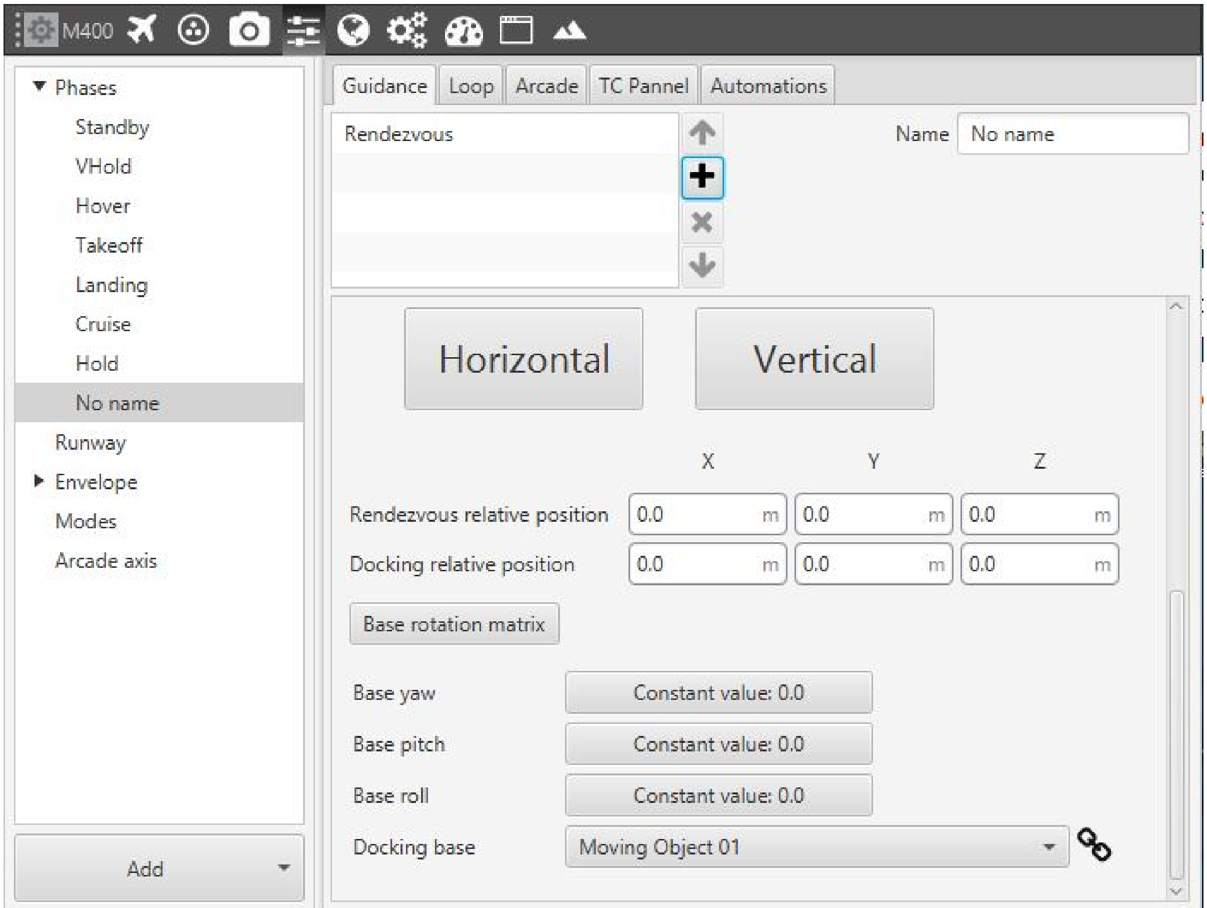
Rendezvous guidance is used to create a meeting point where the Air unit will approach a second unit (either Air or Ground) within a determined offset. This guidance updates constantly the vehicle attitude in order to track, with the shortest path, the position of the second unit (named as Ground hereafter). This guidance works for both static and moving Ground. Rendezvous navigation is ready for taking Internest input to improve the precision from its guidance, being the most suitable kind for Internest integration.



Rendezvous Guidance – Menu 1

- **Acceleration proportional gain:** this parameter is related with a new control system that Embention is developing in which elevator and thurst work side by side in both the pitching and thrusting.
- **Set height mode:** the height mode indicates how the aircraft will perform the route.
  - **2D mode:** if this mode it is selected, the platform will follow the predefined route without taking into account the altitude of the waypoints, it will keep the altitude that it has at the moment it enters in this guidance.
  - **2.5D mode:** is used in multicopters only, in this case, the vehicle will climb vertically to the altitude of the first point of the route and then it will begin it.
  - **3D mode:** the vehicle goes from the altitude at which it enters in this mode, to the beginning of the route in a diagonal trajectory (it follows a 3D trajectory that connects the two points).
- **Set speed:** this option sets the speed that the vehicle will have during the phase. It could be IAS (indicated airspeed) or Speed (Ground Speed). Normally, the IAS is used for airplanes and the Speed for multicopters. The option Waypoint reach is used to indicate the speed at which the platform will reach the waypoints, so it will

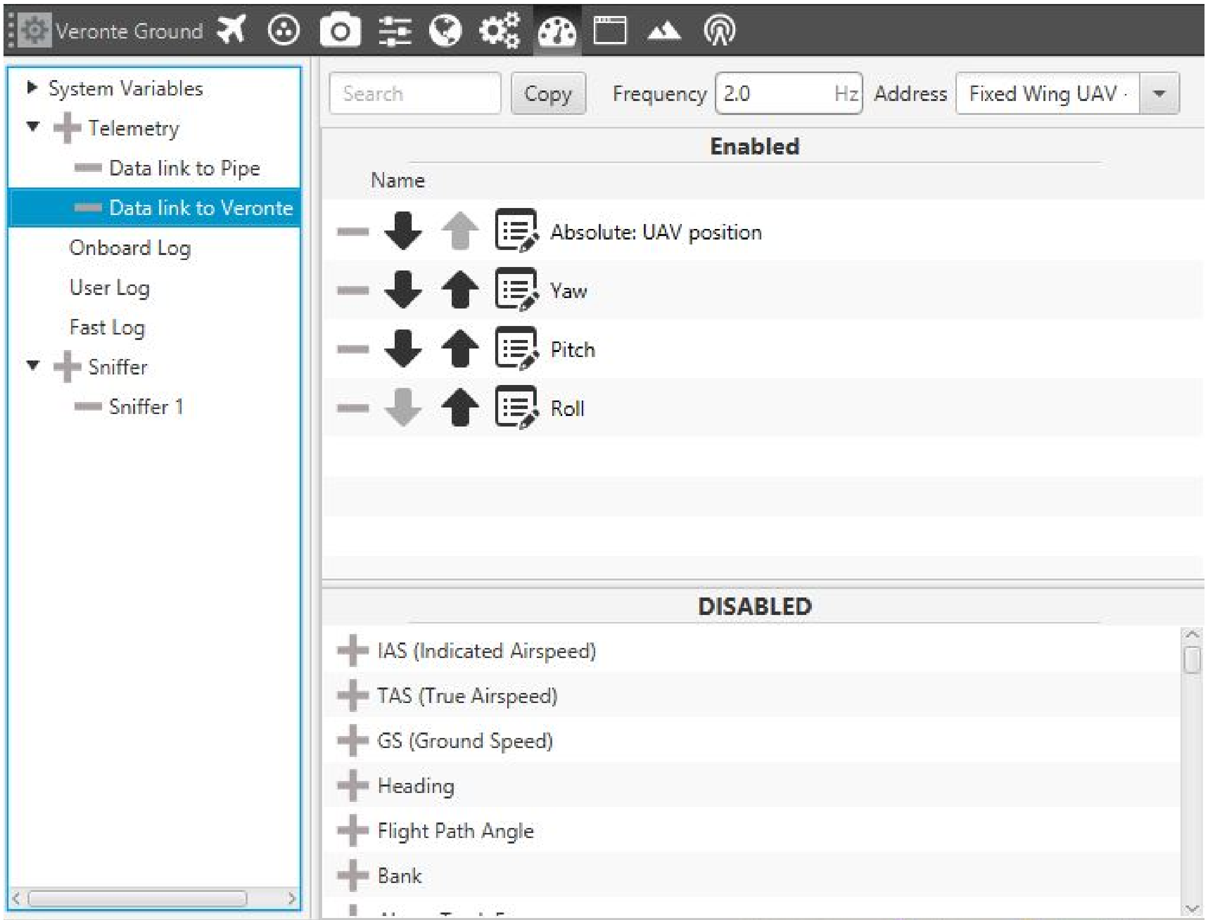
travel along the path with the speed indicated in the option Cruise, then it will decelerate or accelerate to the speed indicated in Waypoint Reach and then it will go back to the cruise speed.



Rendezvous Guidance – Menu 2

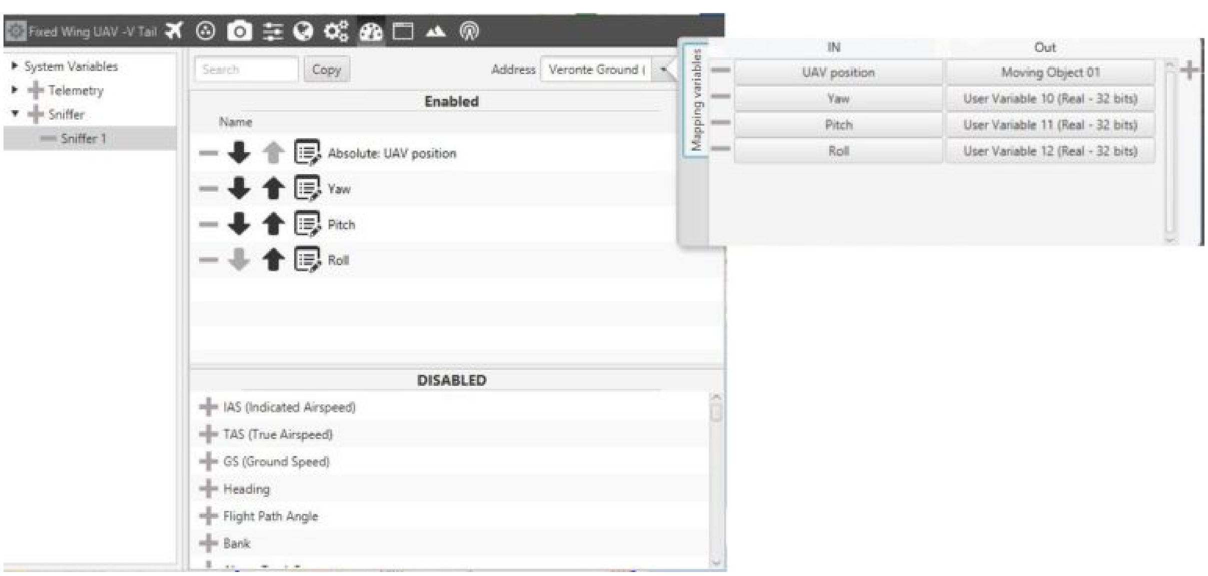
- **Axes Controller (Horizontal and Vertical):** configuration of the PIDs from the system for both axes.
- **Docking base:** defines the position of the GNSS antenna connected to Veronte Ground.
- **Base yaw, pitch & roll:** defines the attitude from the base body. These values affect the navigation by orienting the Air unit to be equal to the attitude from the Ground one. To be configured with telemetry (example below).

In order to configure the Moving Object which is assigned to Docking Base and the Base attitude, the next procedure must be followed.



Ground Telemetry

In the Ground unit, go to Variables *Telemetry* – Data link to Veronte and add the variables Absolute: UAV position, Yaw, Pitch & Roll. A recommended frequency is 10 Hz. On address, point to the Air Unit (it is needed to have both units connected through the radio in order to be able to see them on the menu).



Air Sniffer

For the Air unit, go to *Sniffer* – Add a new Sniffer and configure the same Variables (keeping the same order) than in the Ground unit (in this case Absolute: UAV position, Yaw, Pitch & Roll). Address the Ground unit and in the gear next to it, configure the 4 incoming variables as System Variables. Assign UAV Position to Moving Object and the 3 variables from attitude to 3 different User Variables (keeping the same order as well).

In order to be able to see the Moving Object position in Pipe interface, go to Air unit configuration ref Variables – Telemetry – Data Link to Pipe and add Moving Object to the list of variables.

- **Docking relative position:** 3D point used to configure the offset for the approaching vehicle to the Docking base. This will be the difference from GNSS position that defines the landing point.
- **Rendezvous relative position:** 3D point used to configure the meeting point for the Air Veronte. This point will be tracked by the vehicle and, once reached, it will start travelling to Docking relative position. For VTOL, X and Y components must be equal. For both Docking and Rendezvous the axes are set according to Veronte orientation.

When defining a guidance system, we refer to a set of commands sent to the platform controller in order to make it carry out a certain task. This task could be follow a line, climb, land, hold one of its states at a certain value and so on.

In Veronte Pipe, it is possible to combine a series of guidances to create custom flight phases that will make the aircraft perform in a given way. For example, to create a Take Off phase, the guidances to be included could be a **Runway**, which defines a line over the runway to make the aircraft follow it, and a **Hold** that will keep the roll and pitch angle at zero to keep the aircraft level when it is accelerating on the runway.

Each Guidance contains a different set of parameters to be configured. All of them are presented as follows and its parameters are explained in detail.

Value	Description
Taxi	Creates a linear path along the runway that is followed by the aircraft.
VTOL	Vertical take-off and landing.
Climb	Makes the aircraft climb from the start of the phase to another altitude.
Hold	Introduces in the control system a desired variable with a value especified by the user.
Cruise	Makes the aircraft follow a determined route created by the user.
Landing	Creates the route that the airplane will follow to land.
Yaw	Indicates the behaviour of the platform in the yaw axis.

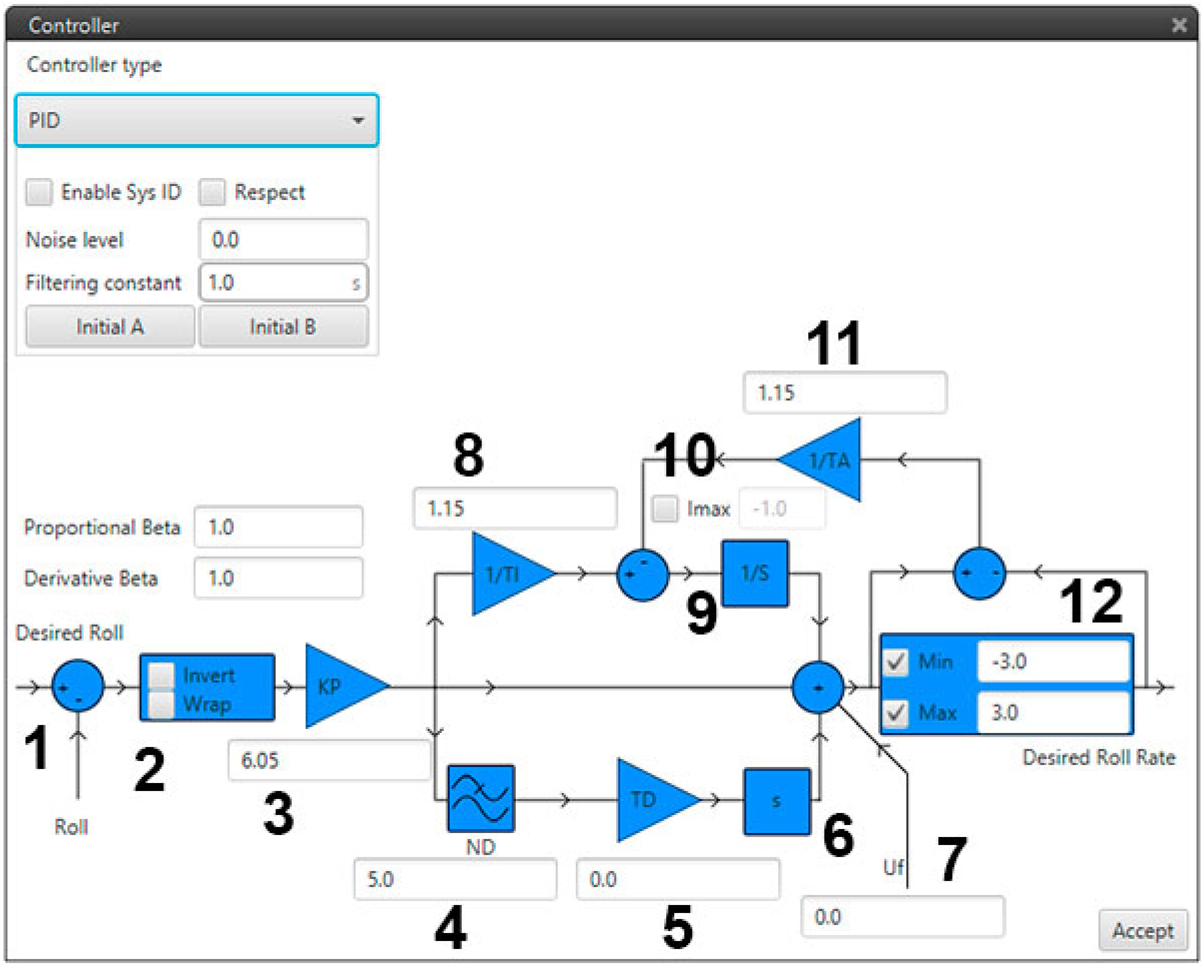
Note: in section *Configurations* there are some examples of how to combine these guidances to create phases for common platforms (Plane, Quadcopter, etc.).

6.2.4.1.2 Loop

6.2.4.1.2.1 PID Controller

Classical PID controller. The values for each one of the elements (proportional, integral, derivative and so on) can be changed directly on the screen.

Focusing on the PID structure the next figure and table describe all its elements and what means each one of them.



PID Architecture

Value	Description
1	Measure
2	Invert: Change error sign/Wrap: Wrap to pi [-,]/It is used in some angular variables (radians) for avoiding numerical errors on the – to change and keep continuity of the error signal
3	Proportional gain
4	Discrete filter parameter
5	Derivative time parameter
6	Derivative gain
7	Constant value added to output (Feedforward Control)
8	Integral gain
9	Inverse integral time parameter
10	The maximum value of integral admitted
11	Anti-windup parameter
12	Output bounds

Output values for PID controller refer to virtual control channels, units must coincide with servo trim configuration settings.

PID diagram represents the following PID model:

$$C = K_p\left(1 + \frac{1}{T_i} IF(z) + \frac{T_d}{\frac{T_d}{N} + DF(z)}\right)$$

PID Mathematical Model

- Kp=proportional gain
- Ti=Integrator time
- Td=Derivative time
- N=Derivative filter constant

For the derivation and integration models, Backward Euler and Trapezoidal (respectively) models have been integrated:

- Backward Euler:

$$DF(z) = T_s \frac{z}{z-1}$$

PID Mathematical Model - Backward Euler

- Trapezoidal

$$IF(z) = \frac{T_s}{2} \frac{z+1}{z-1}$$

PID Mathematical Model - Trapezoidal

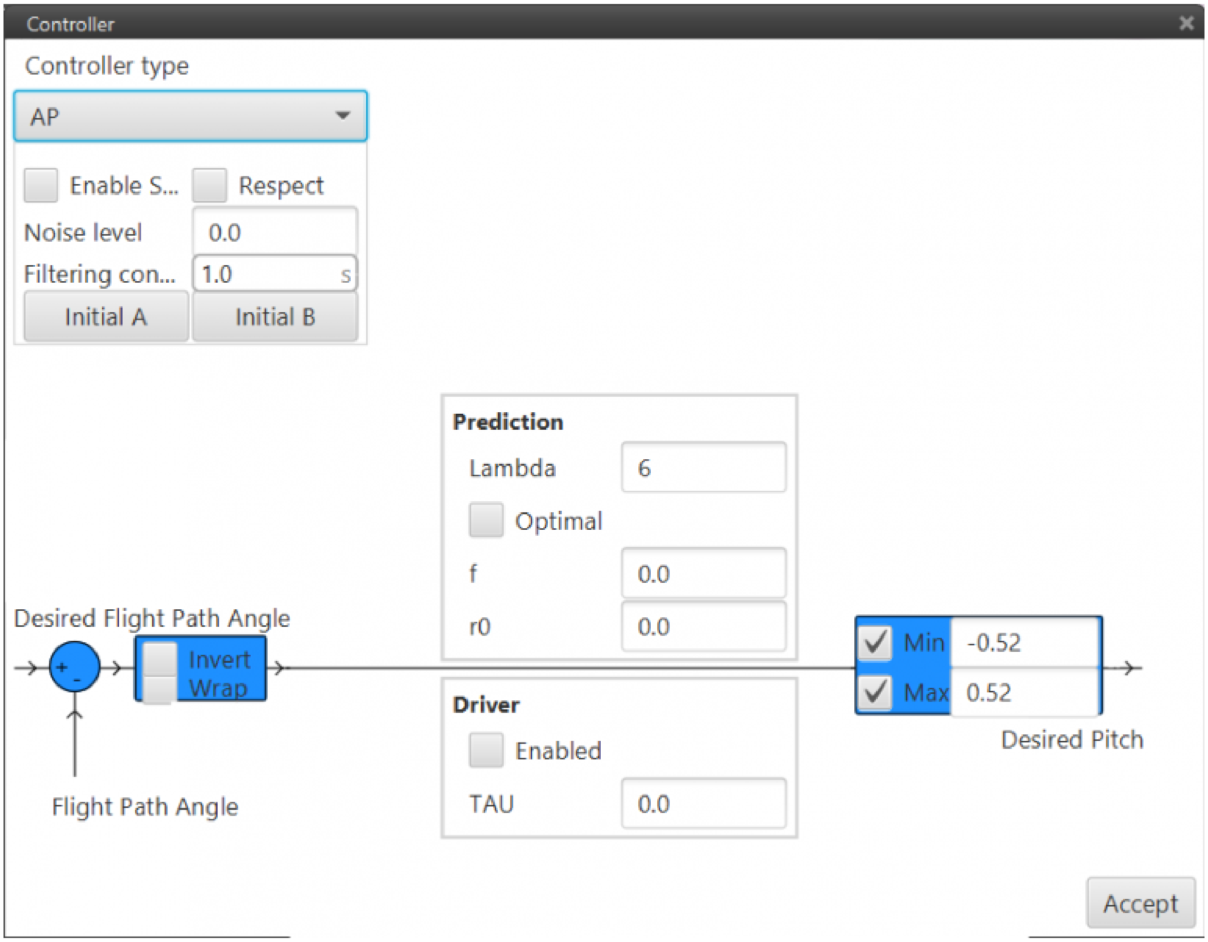
= Td/N where is the time constant on a first order low pass filter (LPF). In Laplace notation:

$$LPF(s) = \frac{1}{\tau s + 1}$$

PID Mathematical Model - Laplace

6.2.4.1.2.2 Adaptative-Predictive Control

This controller incorpores an algorithm for self-tuning i.e, the controller settings are ajusted automatically.



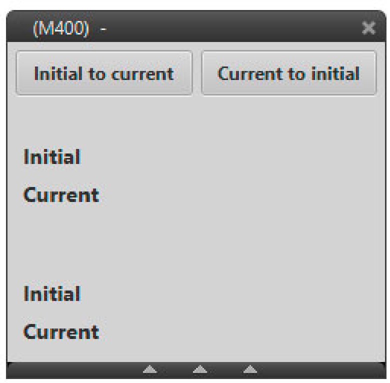
PID Mathematical Model - Adaptative Predictive controller configuration

The next table describes all the parameters available and what means each one of them.

Field	Description
Enable Sys ID	Activates the option of system identification. The plant is modified continuously by predicting a new one according to a set of parameters. This option must be activated when working in Adaptative – Predictive control.
Noise Level	Supposed noise level during System Identification
Filtering Constant	Constant value for the filter
Initial A & Initial B	Used to establish the initial plants constants for the system identification process
Lambda	Parameter that defines the control aggressiveness. High value means less aggressive control, low lambda means more aggressive control
Optimal	Select this option to use the Optimal AP algorithm
f	Sampling input period
ro	Control output period
Enable/Disable Driver	Mark this option to use the Driver Block
TAU	Time constant for the desired default trajectory

In addiction, it is possible to display an AP panel in the workspace that allows the user to perform the following actions:

- **Initial to Current:** to set the initial platform model to the current one.
- **Current to Initial:** to save the actual model to the initial one



PID Mathematical Model - AP Tool in Workspace

6.2.4.1.2.3 Gain Scheduling

This aproach uses a group of linear controllers for different operation points. Each controller adjusts its gains automatically according to scheduling variables.

Veronte incorporates 4 approaches for this type of control:

- **Table Scheduler:** gain scheduling controller with variation of the parameters according to a table interpolation. In this option, the values of the PID controller vary according to a variable that is selected when this option is marked. For example, if the IAS is selected as the scheduling parameter, depending on the velocity of the aircraft the parameters of the PID will change.

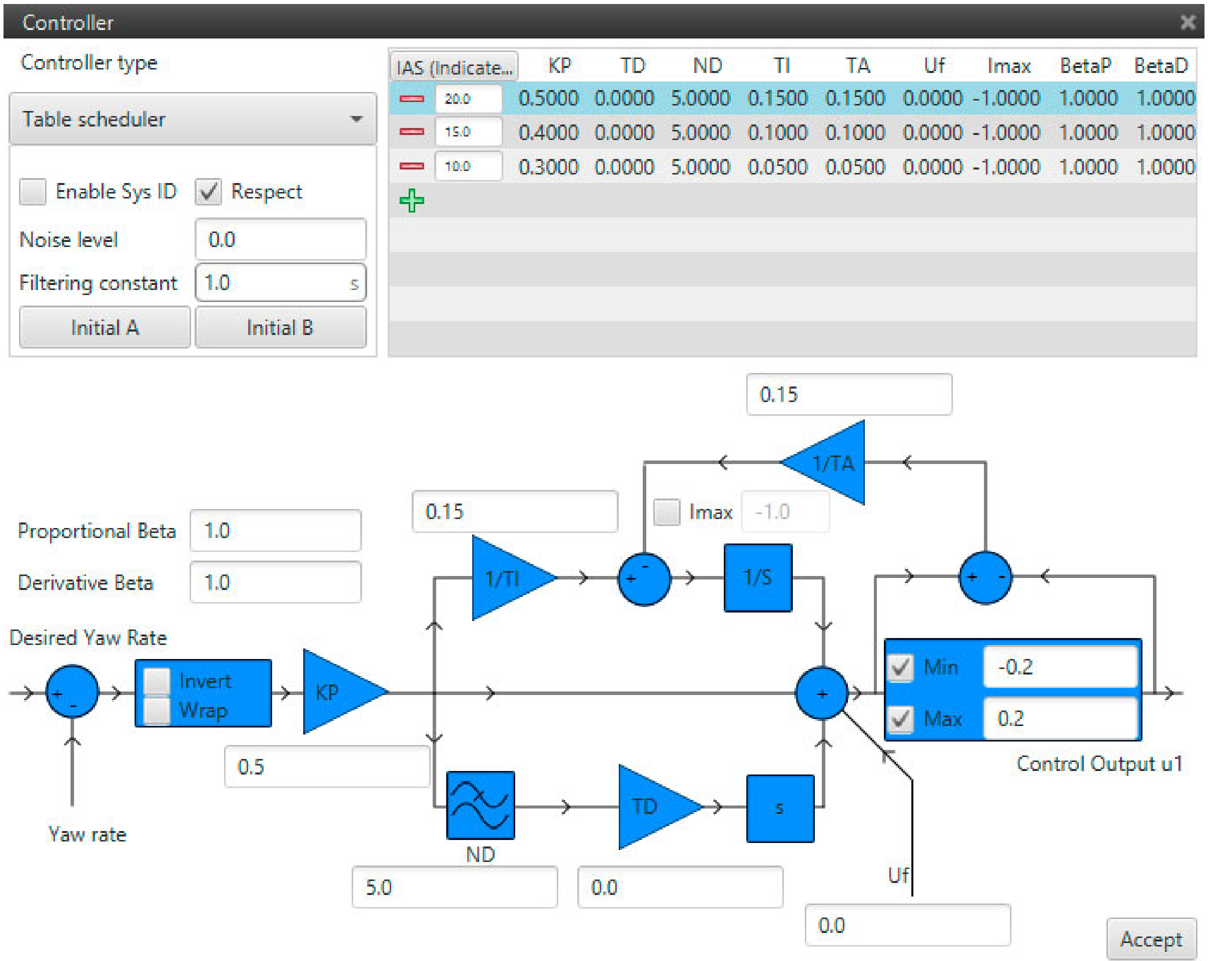


Table Scheduler controller

- **Inverse Scheduler:** gain scheduling controller with variation of the parameters according to the selected variable using inverse proportionality.
- **Proportional Scheduler:** gain scheduling controller with variation of the parameters according to the selected variable using direct proportionality.
- **Quadratic Scheduler:** gain scheduling controller with variation of the parameters according to the selected variable using quadratic proportionality.

For the last three approaches, the only gain that changes according to a certain variable is the proportional one. Now there is not a table interpolation but a mathematical expression, that can be inverse (KP1/V1=cte), proportional (KP1 * V1=cte) and quadratic (KP1 *V1^2=cte). The nominal point given by the expressions will remain fixed. Therefore, the proportional gain will change when V1 varies in order to mantain the same value.

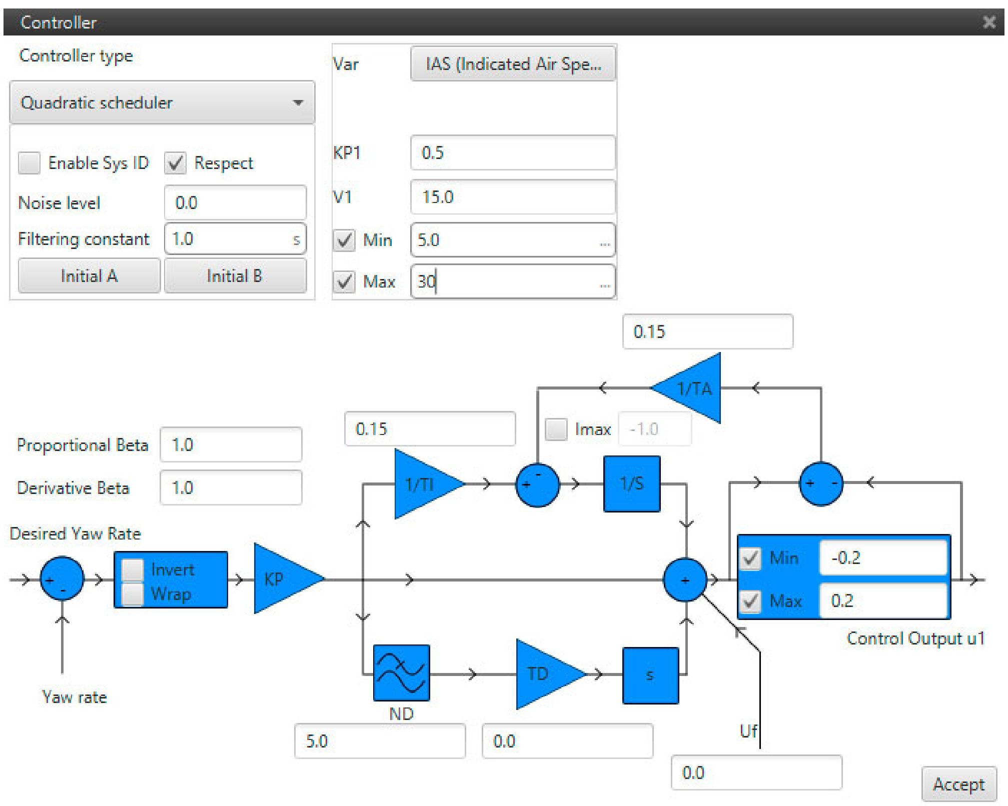
	C	Kp	Result (Kp)
Inverse	$\frac{Kp}{V_i}$	C * V	$Kp_i \frac{V}{V_i}$
Proportional	$Kp_i * V_i$	$\frac{C}{V}$ (V > 0)	$Kp_i \frac{V_i}{V}$
Quadratic	$Kp_i * V_i * V_i$	$\frac{C}{V*V}$ (V^2> 0)	$Kp_i \frac{V_i^2}{V^2}$

Scheduler table

In the previous expressions, shall be clear that limits are applied to the variable (V) and not to the gain (Kp).

The following figure shows the quadratic case, where the proportional gain changes according to the IAS, having as nominal point (KP1 and V1) 0.5 and 15 m/s respectively.

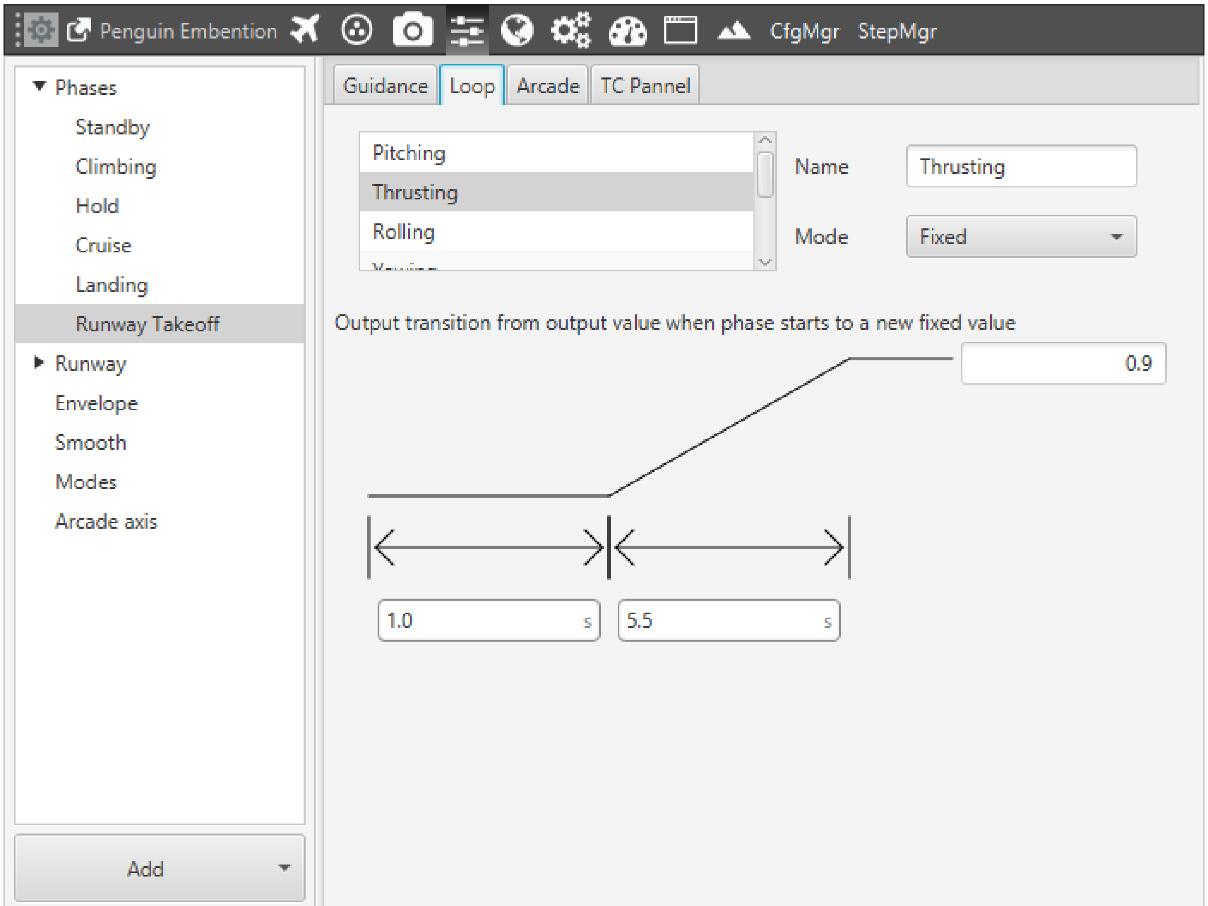
The values Min and Max are used to establish a limit of the scheduler. Below and above those limits, the system works as a conventional PID with the gain indicated in KP1.



Quadratic Scheduler controller

6.2.4.1.2.4 Fixed Control

When **Fixed** mode is selected, the Open Loop Control parameters are set to a fixed value.



Fixed Value Settings

Three values must be entered:

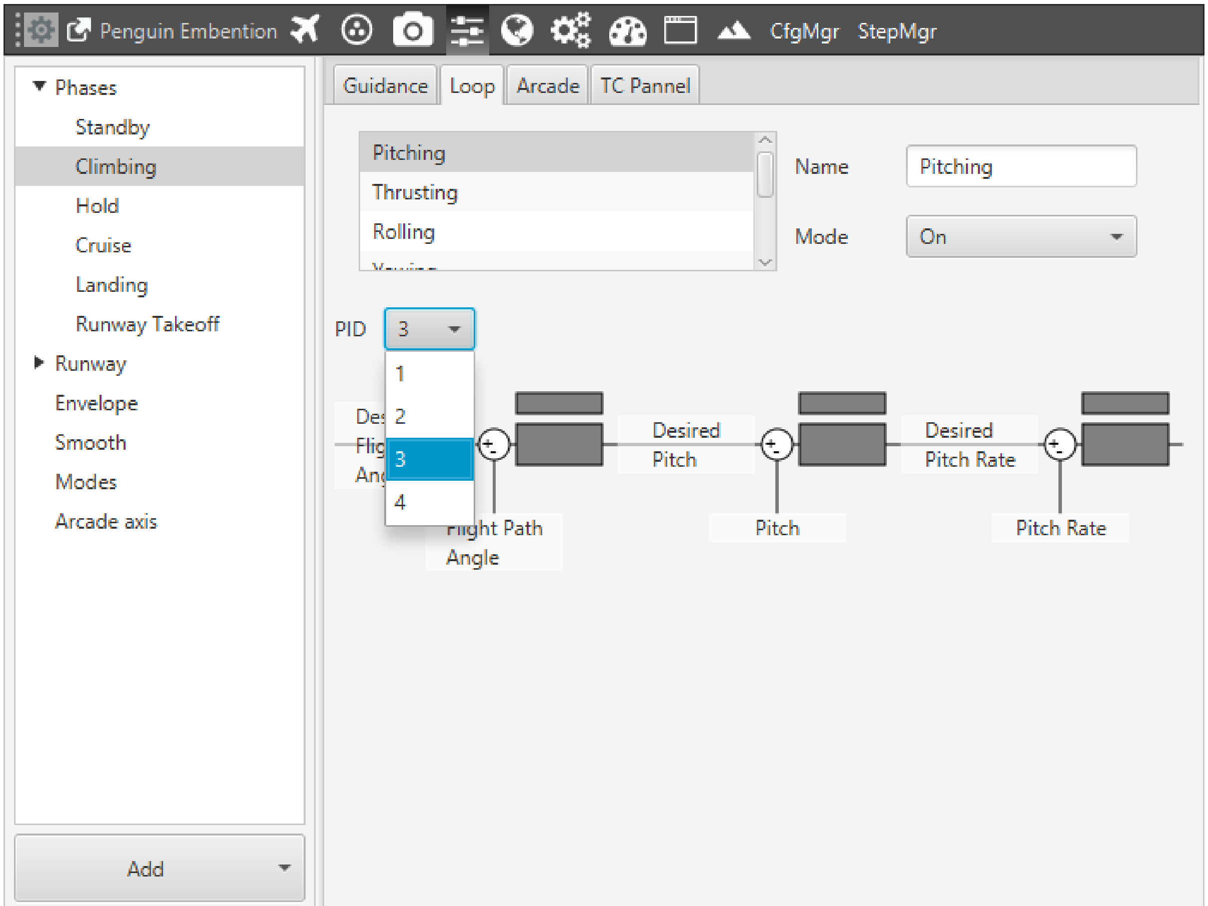
- Remaining time in the starting conditions.
- Transition time
- Variable final value.

In each one of the mission phases, it is possible to configure a controller for each control channel defined on Veronte Configuration. There are three different options for the control status.

Value	Description
Off	Disables the controller.
On	Enables the Closed Loop Control.
Fixed	Sets the Open Loop Control parameters to a fixed value.

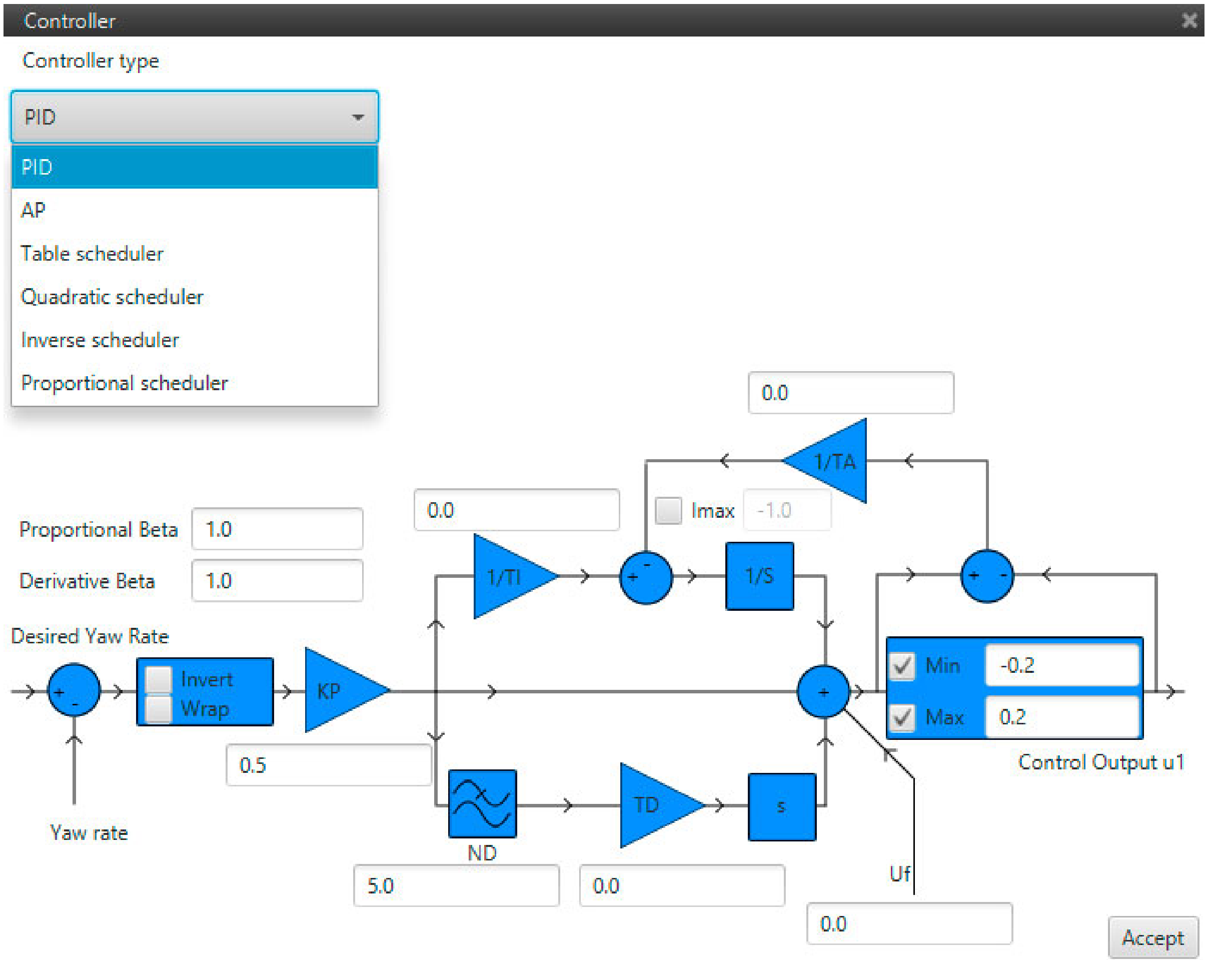
6.2.4.1.2.5 PID Settings

When creating the controller for a channel, it is possible to select up to four different loops connected in series.



Loops Diagram

For each block, it is possible to configure the controller type and its parameters by selecting it on the pull-down menu shown in the following figure.



Loop Configuration

There are six different options that can be implemented for a controller on Veronte Pipe:

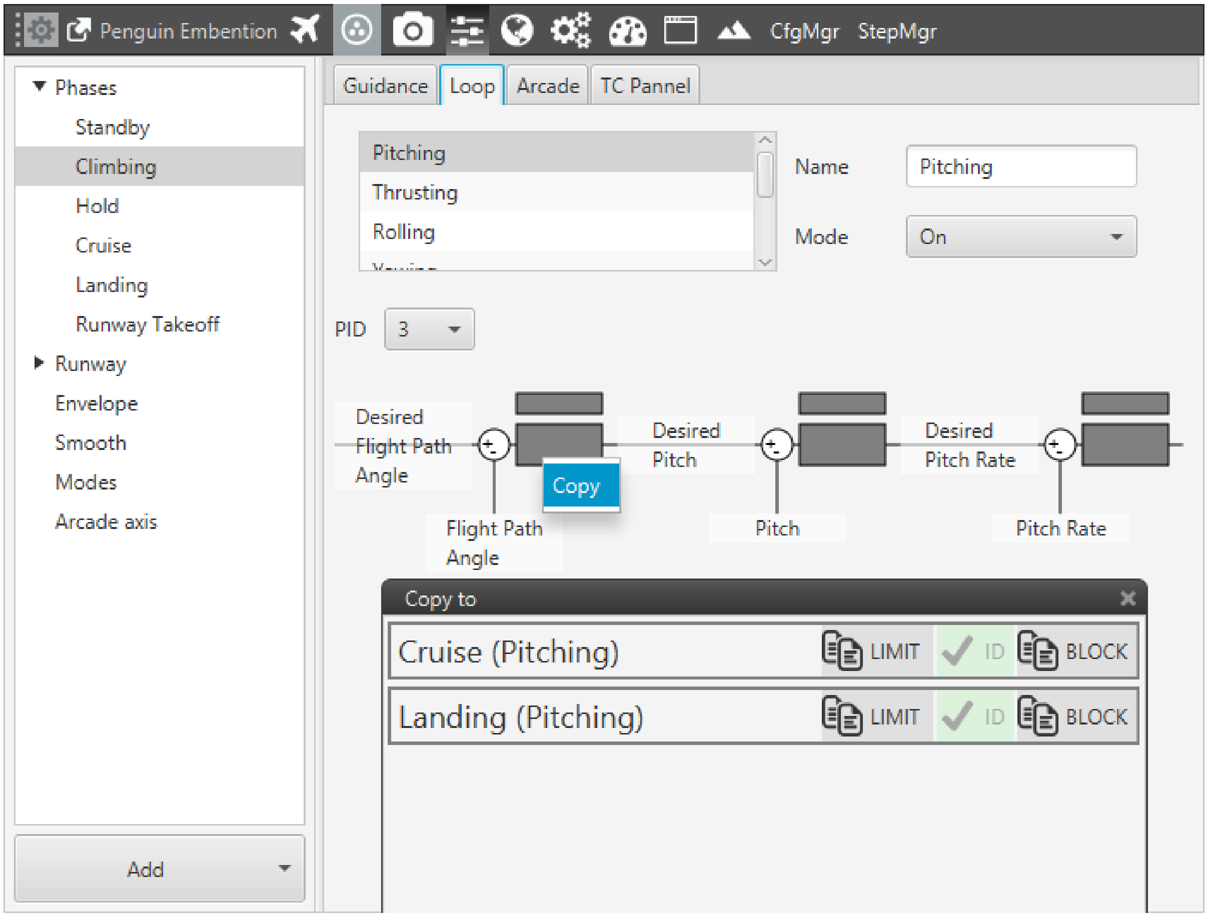
PID	Classical PID controller.
AP	Adaptative-Predictive control.
Gain scheduling controller	Gain scheduling controller with variation of the parameters:Table Scheduler, Inverse Scheduler, Proportional Scheduler, Quadratic Scheduler

In addition to the controller type, there are another set of parameters that can be changed in the window of each block.

- **Enable Sys ID:** activates the option of system identification. The plant is modified continuously by predicting a new one according to a set of parameters. This option must be activated when working in Adaptative-.Predictive control.
- **Respect:** this option is used for transition between phases. When activated the output of the controller is respected (is kept) when going from one phase to another. Normally, this is used in the internal control loops (rates) only to avoid great steps in the control parameters during transitions. If Respect is active in the external loops, the control will maintain attitude angles and heading/fly path angles during phase changing (for example), and this kind of control could be too much aggressive for the platform (depending on phases configuration). The Respect option **can be activated in PID controllers only.**
- **Noise Level & Filtering constant:** parameters used for the AP controller.
- **Initial A & B:** these buttons are used to establish the initial plants for the system identification process.

6.2.4.1.2.6 Exporting PIDs To Other Phases

Once it is considered that the PID is tuned, the user can easily export that PID in order to use it in other phases. To do so, just select **Copy** by right-clicking on the desired PID and click on the **Block** cell of the desired phase. That would copy only the gains of the PID (P, D and I), if the user also wants to copy the **Limits**, that column has to be checked.

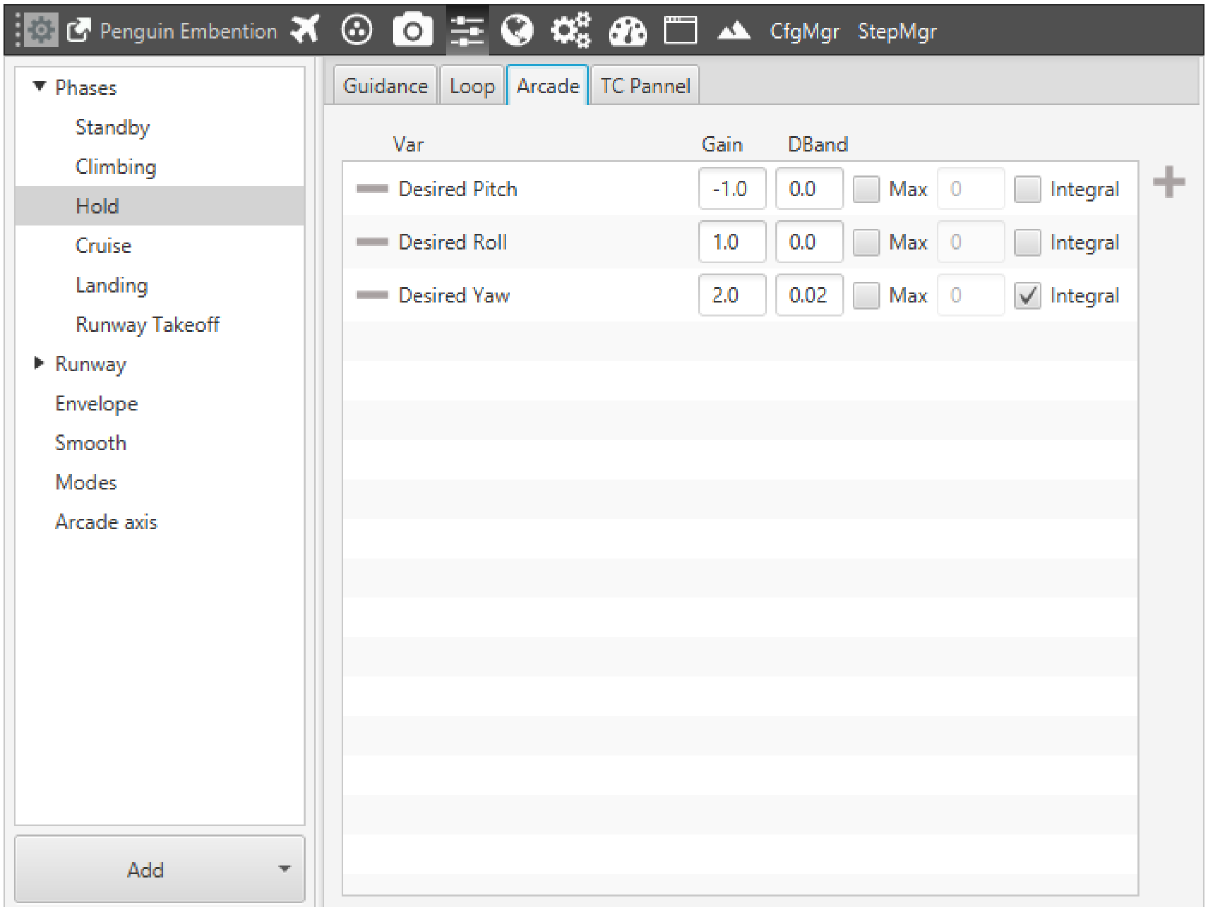


PID Exporting

6.2.4.1.3 Arcade

6.2.4.1.3.1 Arcade Mode Settings

The arcade mode could be considered as an aided manual control where the pilot sends a control command and not a PWN signal for the actuators. This control command is used by the Hold or Yaw guidance algorithm to provide the input for the controller. So, in this case, is not directly the autopilot which generates the PID inputs, but an optimal combination of its commands and the ones from the pilot.



Arcade Configuration Menu

Considering that the pilot's stick can reach values approximately between 0 and 1, the **gain** is applied to the stick position before entering into the controller so the complete range of the controlled variable is covered.

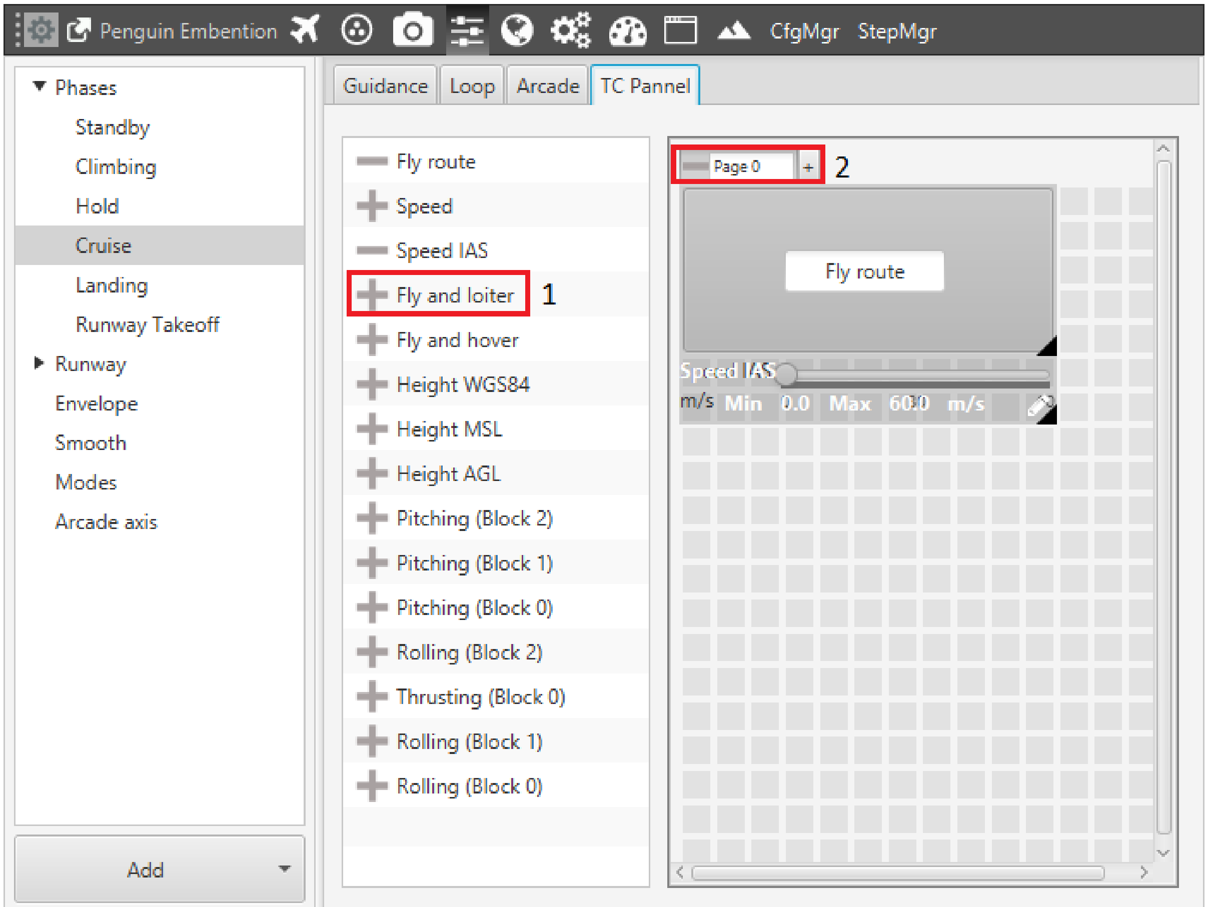
The **DBand** option (Dead Band) creates a zone where the movement of the stick is not sent to the system, that is used because of the noise present on the stick position. So if the pilot has the stick in the neutral position, the actual value would be 0.001 for example and not zero. If the dead band is set to +-0.01, that command will not be sent to the system avoiding then a possible malfunction.

Finally, when the **integral** option is marked the variable increases continuously on joystick hold, and when is unchecked the control variable is reset after joystick release.

The Arcade mode has to be trimmed before the flight and each time the radio controller is changed. For more info about the Trim Arcade process, please visit section *Arcade Trim*.

6.2.4.1.4 TC Panel

This menu sets the variables that will be accessible during the flight from the Veronte Panel. Considering as an example a typical navigation phase, the parameters that can be changed from the TC Panel appear in the following figure.

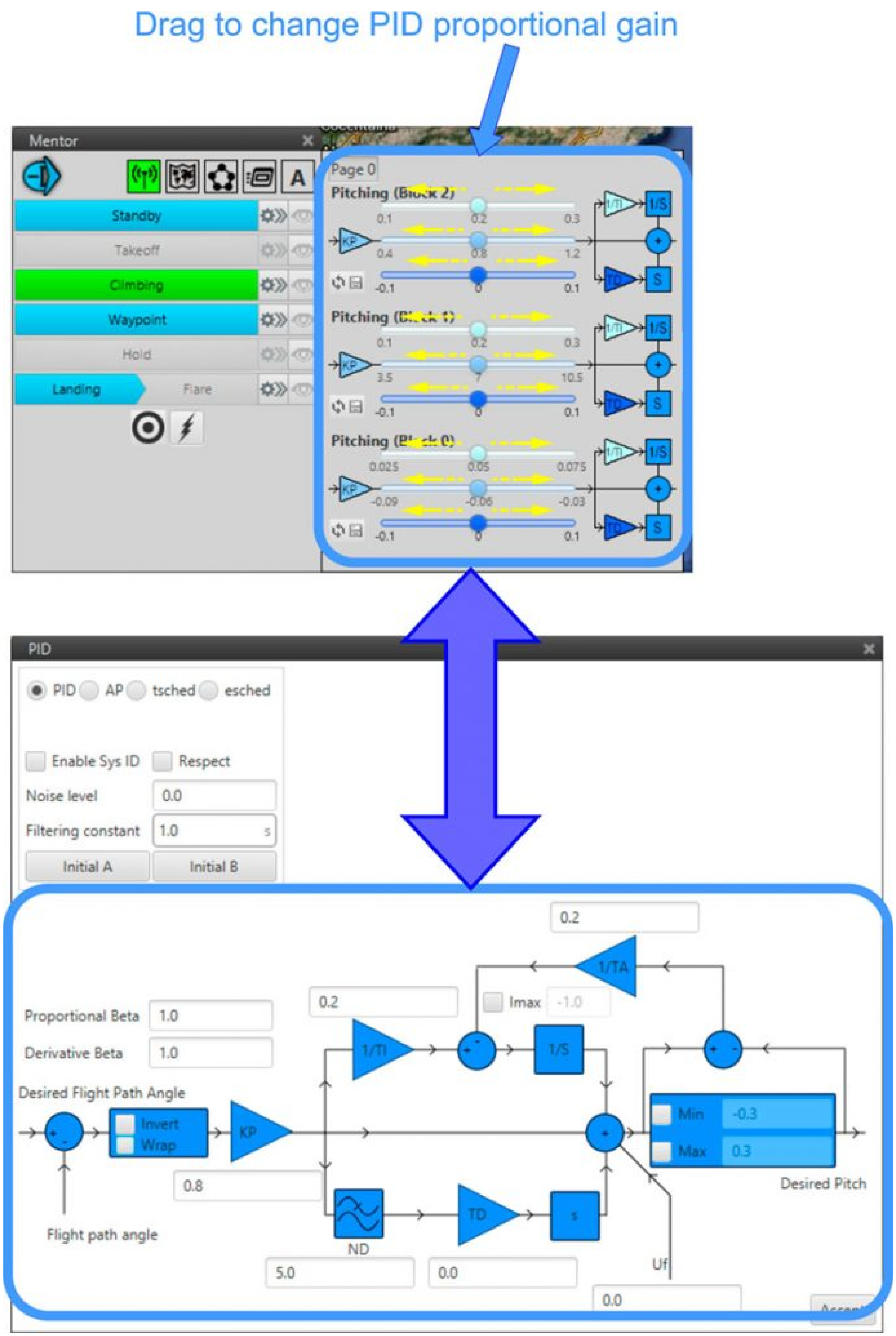


TC Panel

- Number **1** is the button used to add a variable to the TC Panel of a phase (the "-" button is used to remove an element).
- Number **2** indicates the TC Panel page where the current element is being added. Pipe allows user to set more than one page in order to arrange all the variables in an easily accessible way.

6.2.4.1.4.1 PID Gains

The parameters that are used the most for the TC Panel are the gains of the PID controller, because that provides the operator with an easy way to adjust the gains on a real flight. Each control configured in the Loop Panel will be shown on the TC Panel.


Drag to change PID proportional gain
Mentor
Standby
Takeoff
Climbing
Waypoint
Hold
Landing
Flare
Page 0
Pitching (Block 2)
Pitching (Block 1)
Pitching (Block 0)
PID
PID AP tsched esched
Enable Sys ID
Respect
Noise level
0.0
Filtering constant
1.0
Initial A
Initial B
Proportional Beta
1.0
Derivative Beta
1.0
Desired Flight Path Angle
Invert
Wrap
Imax
-1.0
Min
-0.3
Max
0.3
Desired Pitch
Flight path angle
ND
Uf
0.2
0.8
5.0
0.0

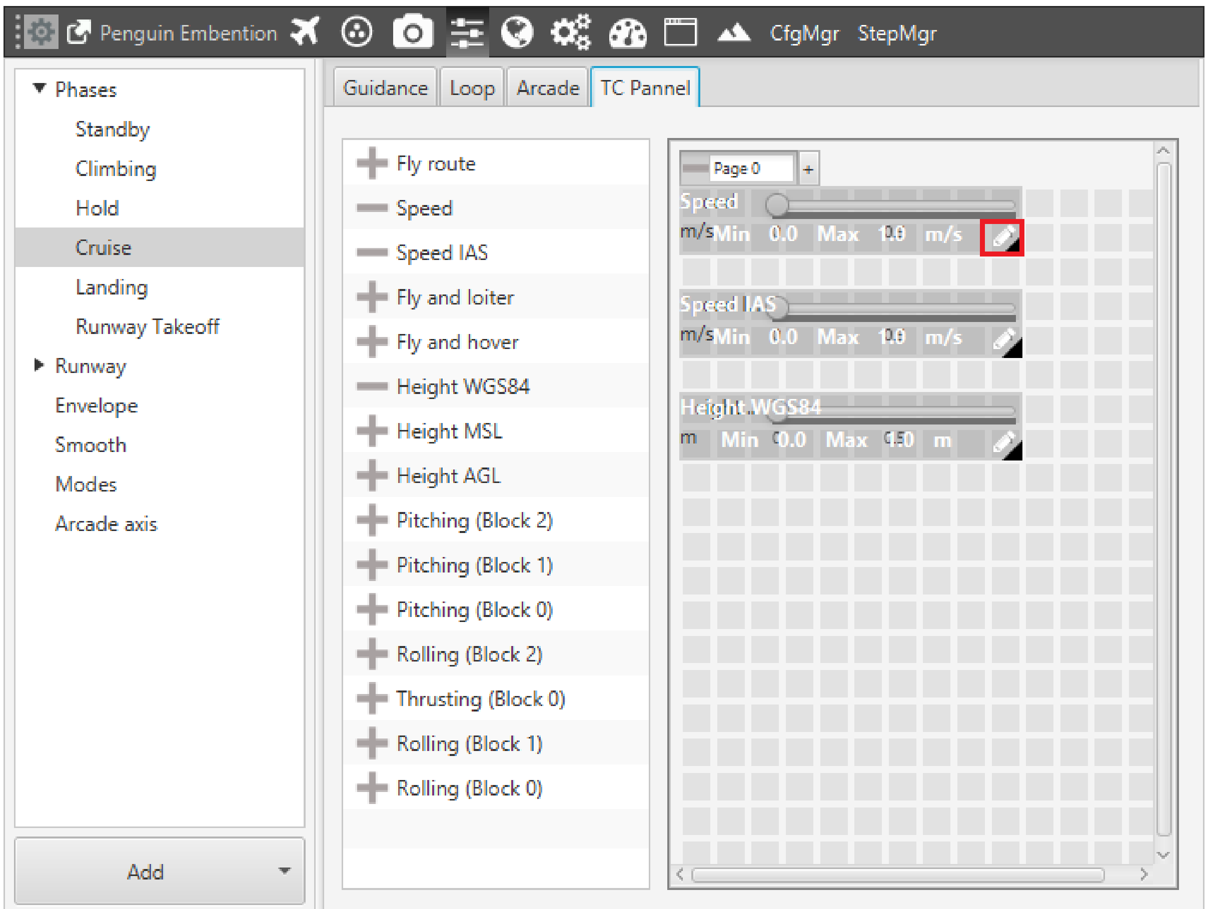
Set PID Parameters

6.2.4.1.4.2 Cruise Guidance

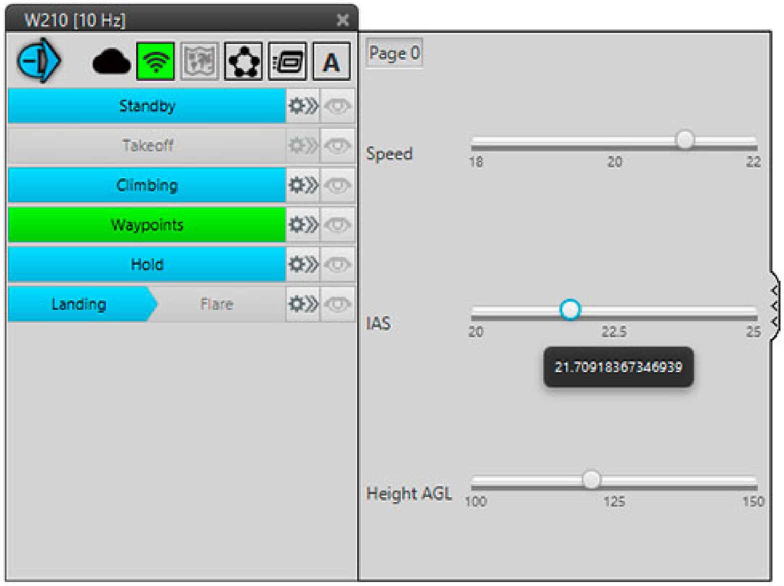
In addition, if the Guidance of the phase is Cruise, the TC Panel can also hold the option of creating different types of interactions with the UAV:

- **Desired variable change:** Speed, Speed IAS, Height AGL, Height MSL, Height WGS 84.
- **Route:** Fly route, Fly and loiter, Fly and hover and Follow Leader.

The variable changing allows the user to modify the desired variable in real time during the flight. In the configuration of the TC Panel, a maximum and a minimum of the variable has to be set (red box).



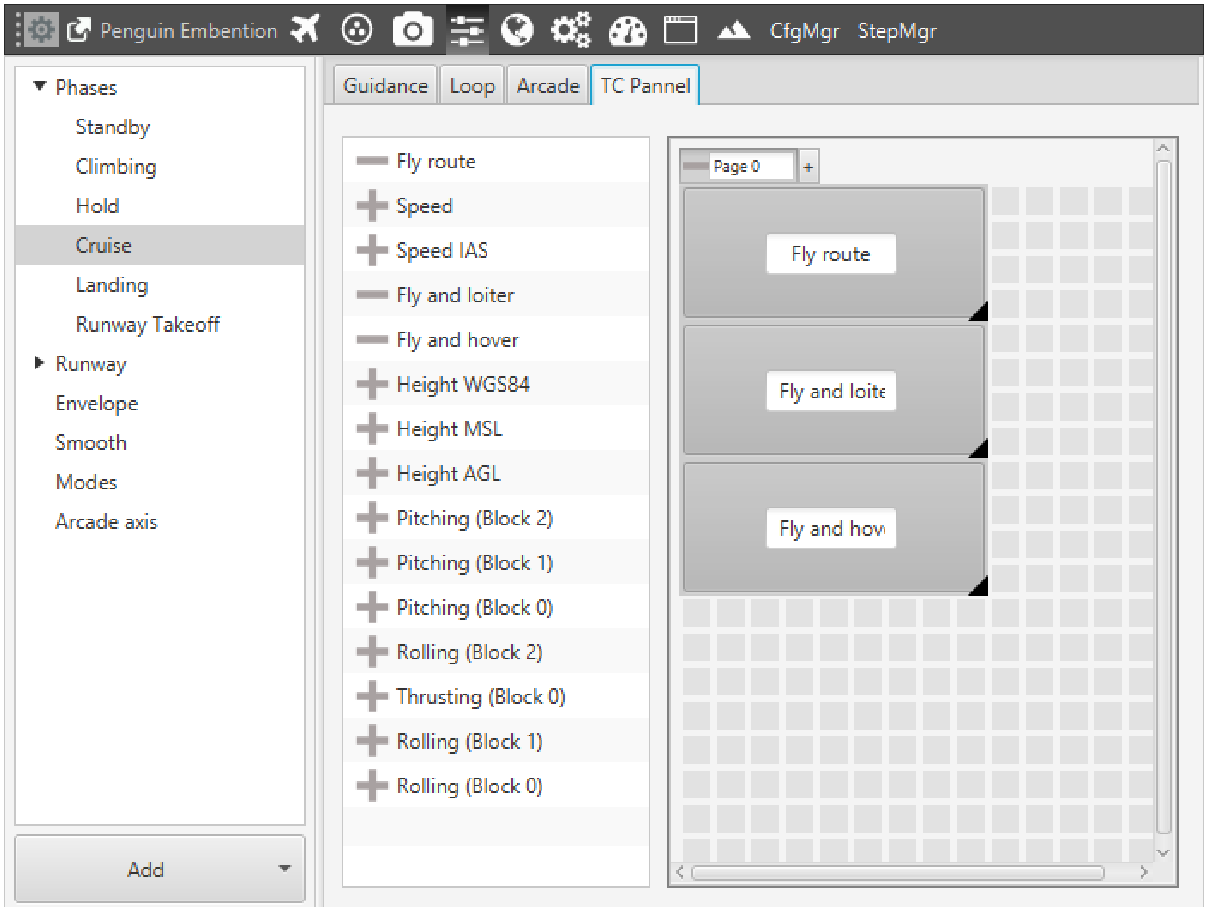
TC Panel variables change set



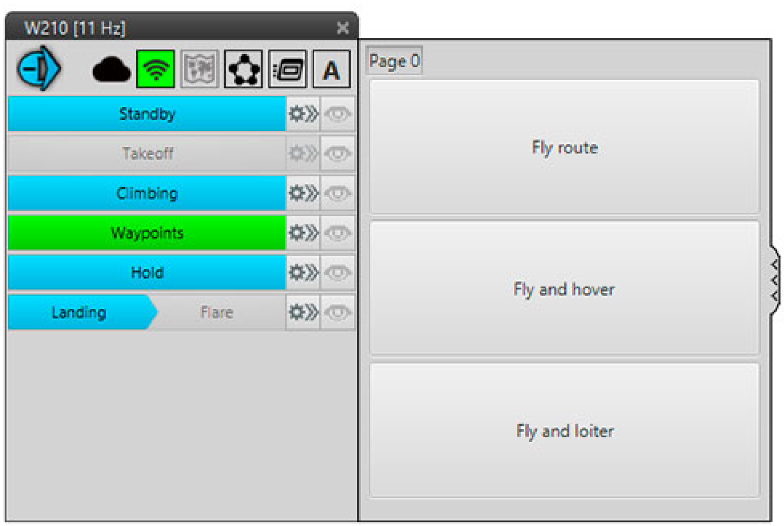
TC Panel during Waypoint phase (Variables)

The route tools can be used when the user wants to change the actual route:

- **Fly route:** the platform will create a new route to reach the selected waypoint.
- **Fly and loiter:** the platform (Plane) will create a new route to reach the selected point and start loitering. In order to set the loiter radius it is necessary to click and maintain to enlarge or reduce the circle.
- **Fly and hover:** the platform (Multicopter) will create a new route to reach the selected point and start hovering.
- **Follow liader:** the platform (Multicopter) will follow an moving object.



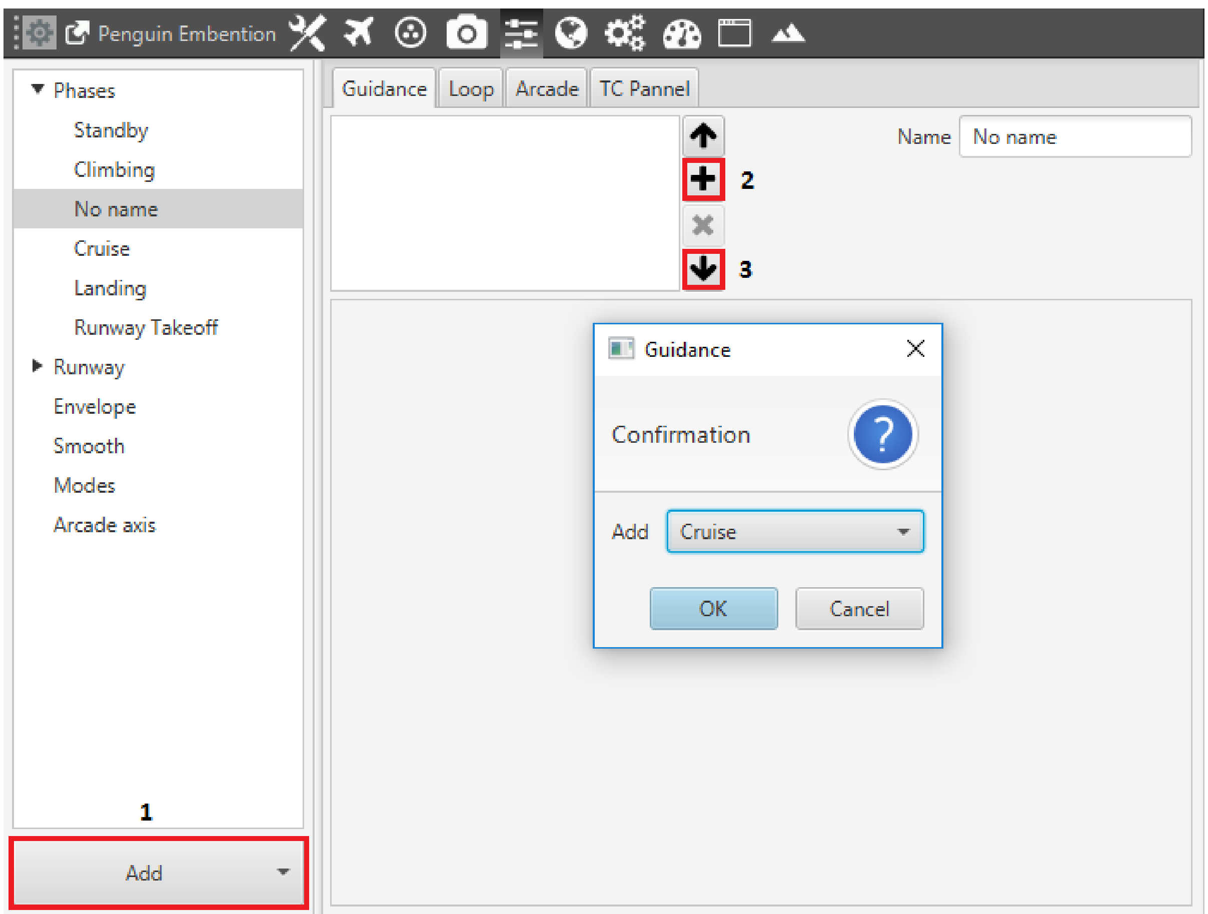
TC Panel routes change set



TC Panel during Waypoint phase (Routes)

Fly and loiter example

The **Phases** menu contains the guidance and control commands used to rule the vehicle. It is possible to create different phases, each one of them with a series of guidance commands and with different control systems. Commonly, the phases correspond with different stages of the flight operation, for example, in an airplane the phases could be: Standby, Take Off, Climb, Cruise and Land. So it is clear from that in each one of them the commands generated to control the vehicle have to be different.



Control Phases Configuration Menu

When pulling down the phases option, the different phases created appear.

- To create a new phase click on **Add**, and select **New Phase** (**1**). The new phase appears by default with the label **No Name** and with any guidance defined.
- Pressing the **Add** button (**2**), a new Guidance is added to the current phase. The section :ref:`Guidance <guidance>`contains the information about all the guidances available on Pipe.
- Lastly, the button (**3**) is used to order the guidances when there is more than one of the same type.

The second tab of the Control menu contains the **Control – Loop** system for each phase. The control system refers to the type of controller used for each one of the control channels of the platform (Pitching, Yawing, Rolling and Thrusting). More information about this section is found in *Loop*.

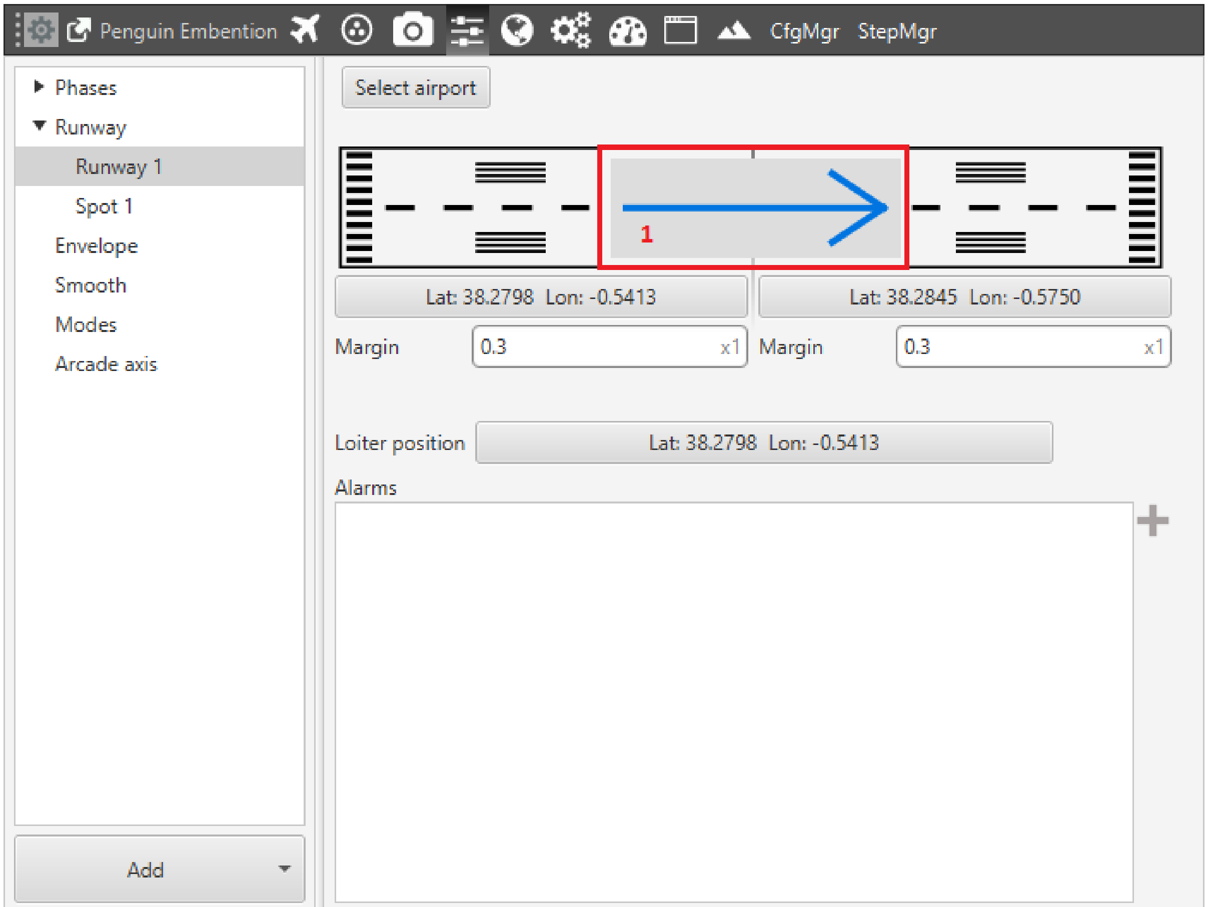
Arcade refers to the aided mode of controlling a platform, where the control system aids the pilot to improve the piloting experience. All the information about this option is available in section *Arcade*.

Finally, the TC Panel tab is used to include in Veronte Panel a series of variable to be changed in real time (while the platform is flying). Refer to *TC Panel* for more information about this.

6.2.4.2 Runway

6.2.4.2.1 Runway

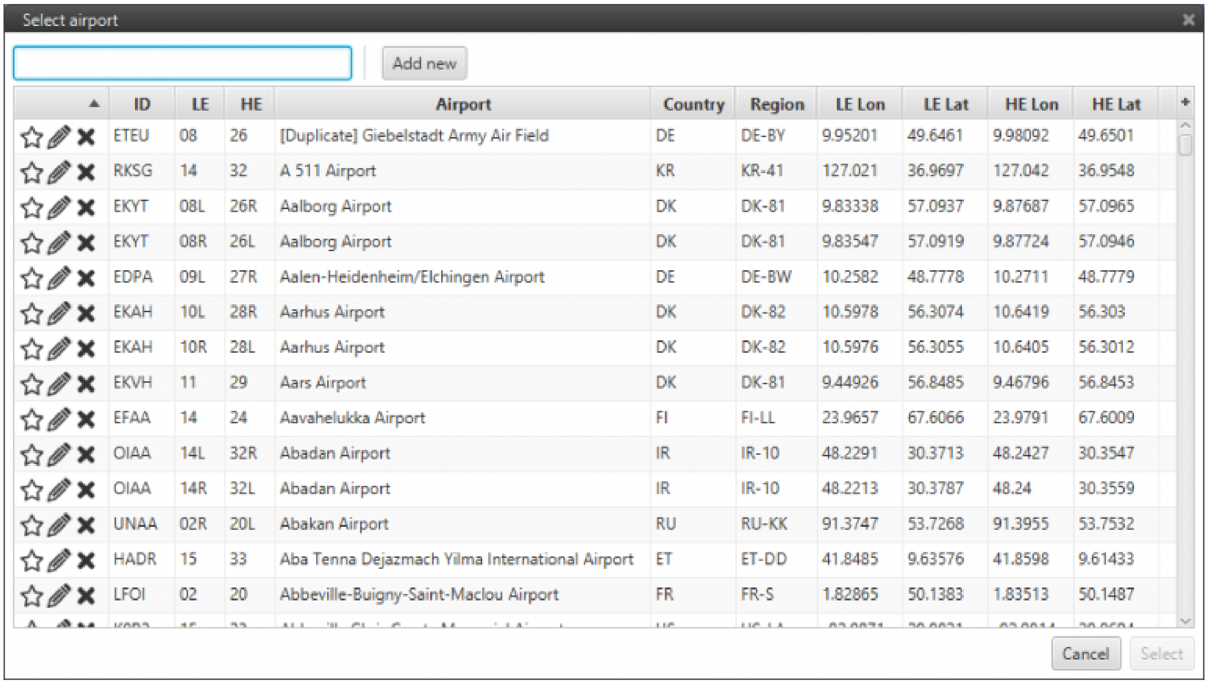
This menu allows the user to configure a **Runway** and **Spot** which are used during the **Phases** configuration. It is possible to configure more than one previously.



Runway Configuration Menu

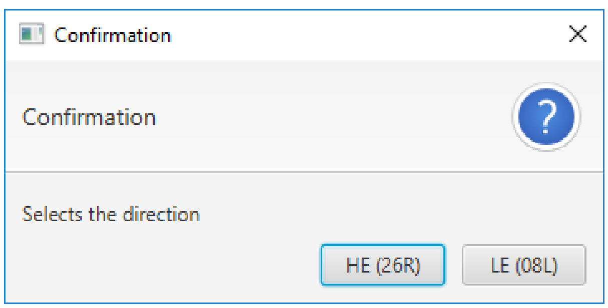
- **Margin:** percentage of the runway distance at which the airplane will try to touch the ground.
- **Loiter position:** this option defines the loiter point of reference (Runway Loiter) and the altitude that the aircraft will reach during the climb.
- **Alarms:** When an alarm is selected, e.g fuel lost. The aircraft will perform the actions associated to that alarm in the selected runway.

Veronte Pipe incorporates a database with different Airports. When the option **Select Airport** is selected the following panel is displayed.



Aiport Selectiong Menu

Once the user selects the desired airport, the different coordinates will be introduced automatically. In addition, the direction of the runway is asked.

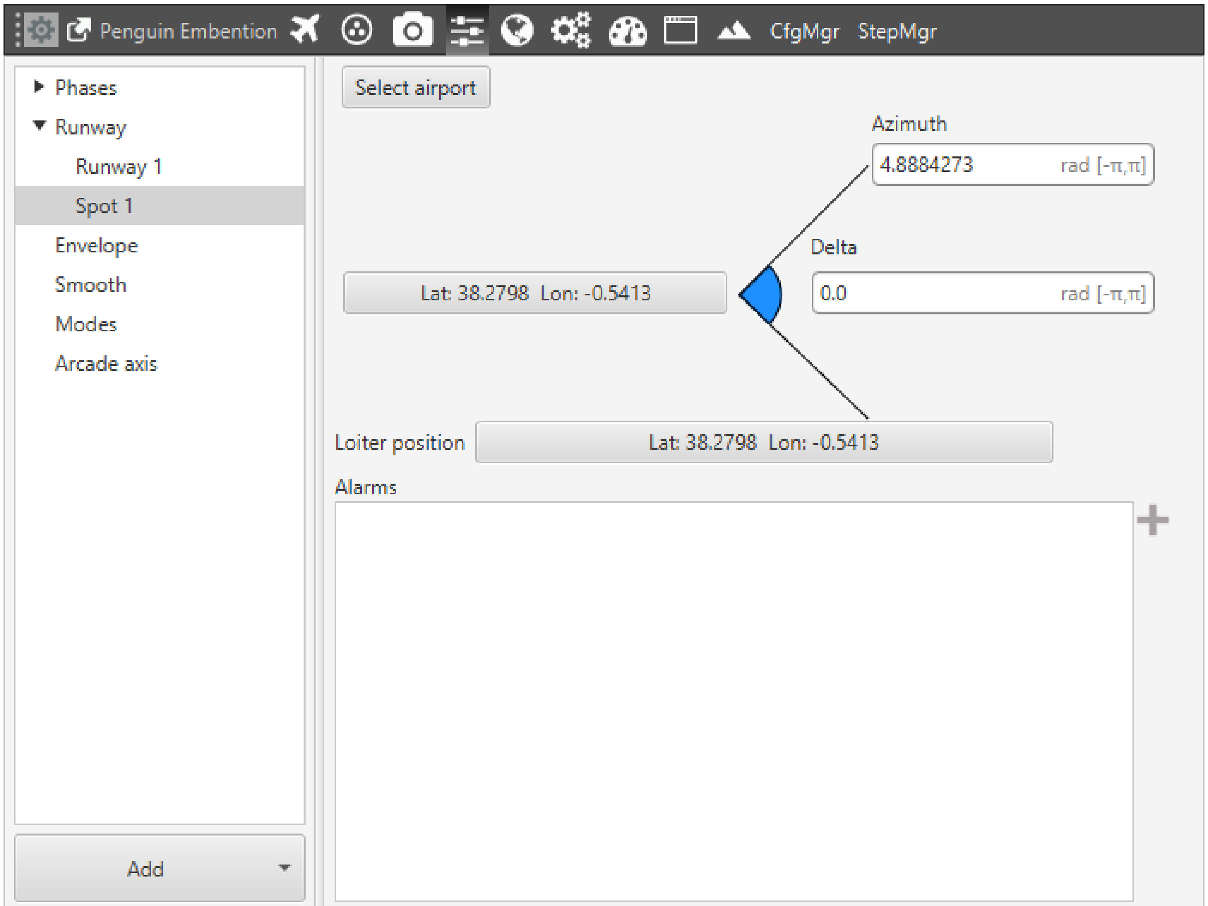


Runway Confirmation

On the other hand, it is possible to introduce the coordinates manually and the direction of the runway can be changed in the configuration menu by pressing 1.

6.2.4.2.2 Spot

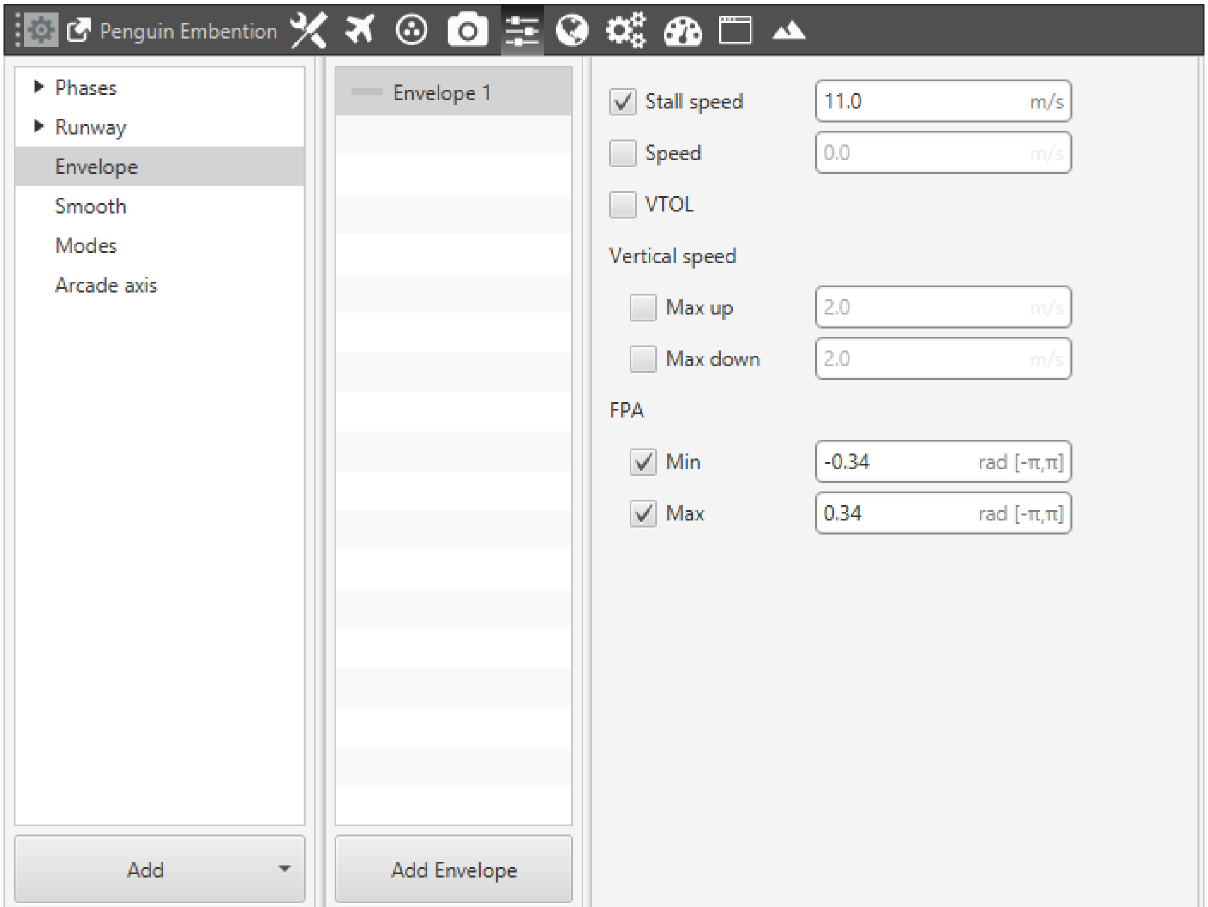
This option reffers to a kind of runway where a initial point and its azimuth is defined by **Select airport** option or the user information. Besides, it is necessary to define a **Delta** angle as is shown in the image. The aircraft will land or take off using the best orientation computed inside the area limited by the parameters introduced.



Spot Configuration Menu

6.2.4.3 Envelope

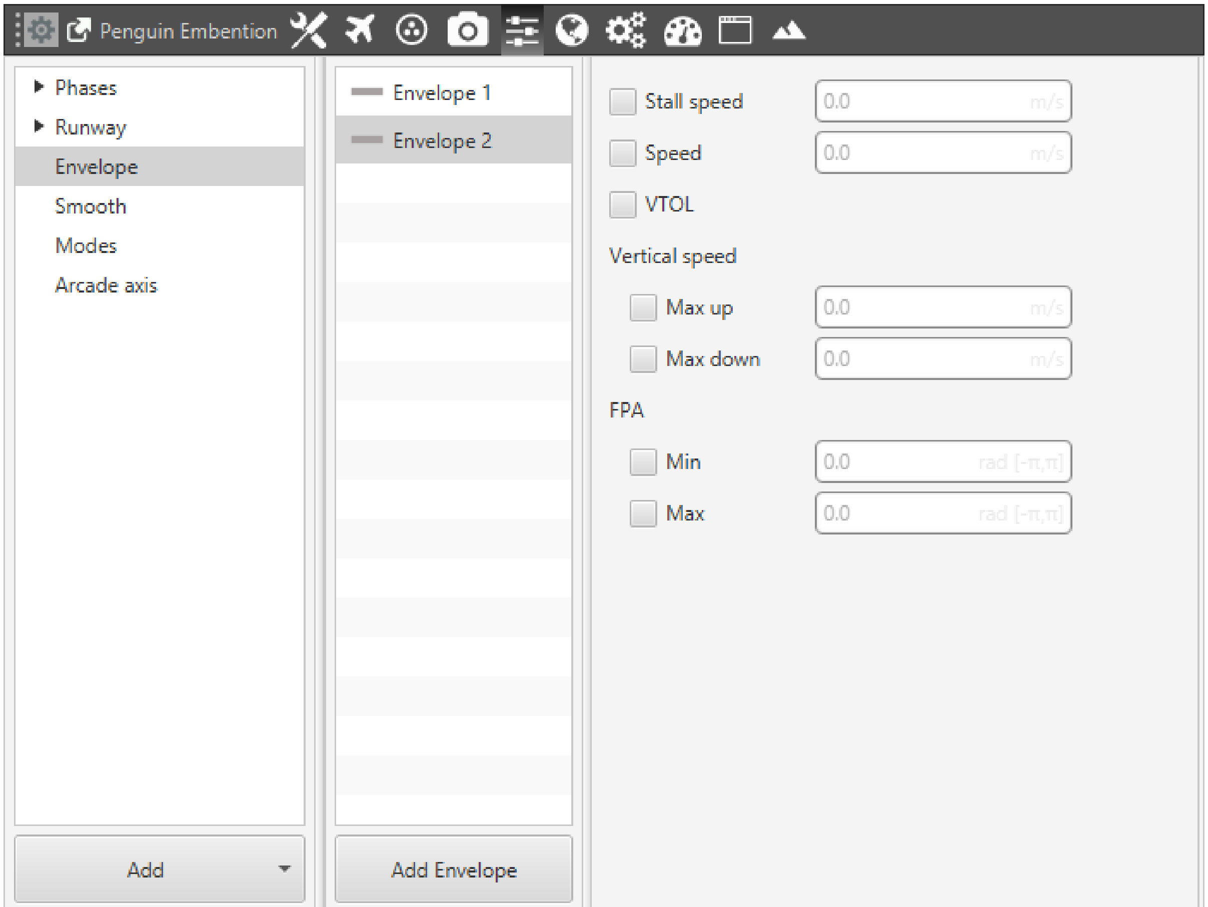
Menu to configure the flight envelope of the aircraft. Here are set the limits that will not be exceeded during the operation.



Envelope

- **Stall speed:** lower limit for the aircraft velocity. Below this level, the aircraft is not able to create enough lift to keep the flight and starts to fall. The value indicated here has effect over the "Cruise" guidance, but is overrided if there is a Hold command on the IAS, so the user must be careful with the velocity commands.
- **Speed:** minimum ground speed of the platform. In case of strong wind, these parameters set the minimum GS that the aircraft can reach, for lower values than this one the thrust will be automatically increased to gain speed and avoid a point where the platform is stopped in the air.
- **VTOL:** this option is used to limit the maximum upwards and downwards velocities in the case of multicopters.
- **FPA:** maximum and minimum values for the flight path angle (angle of climb or descent).

With the last VerontePipe update (from v2.11.1) is possible to insert multiple envelopes (useful for hybrid configurations for example). The change between envelopes can be performed using *Automations*.



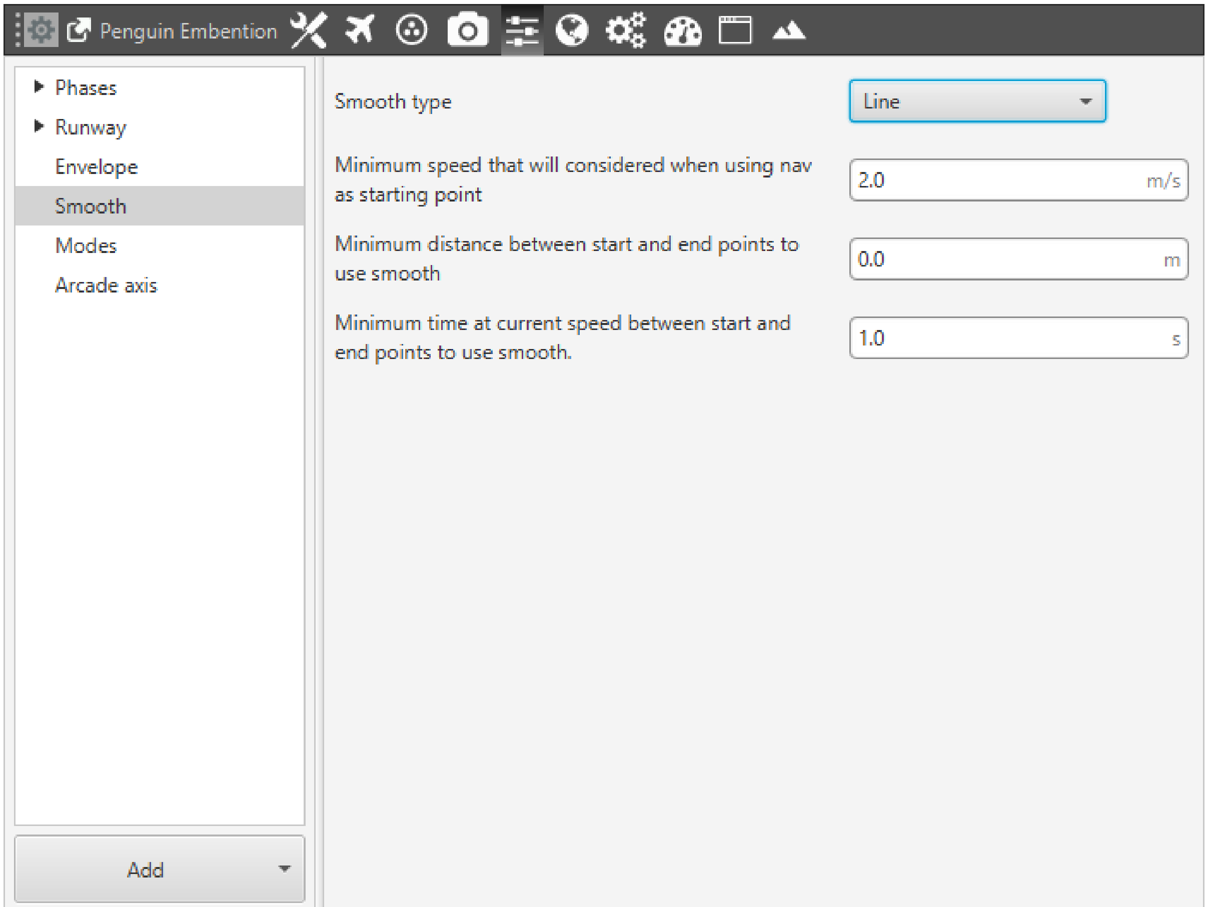
Multiple envelope configuration

6.2.4.4 Smooth

The Smooth menu establish the parameters for the smoothing of the aircraft trajectory when changing from a phase to another, i.e to go from the final point of a phase to the start of the new one.

There are two different ways to smooth the path:

- **Line** creates a linear track between the two points of the smoothed trajectory being possible only to set the position of the start and final points.
- **OGH** creates a cubic curve where is possible to set the position and velocity that the aircraft will have at those points.



Smooth Configuration Menu

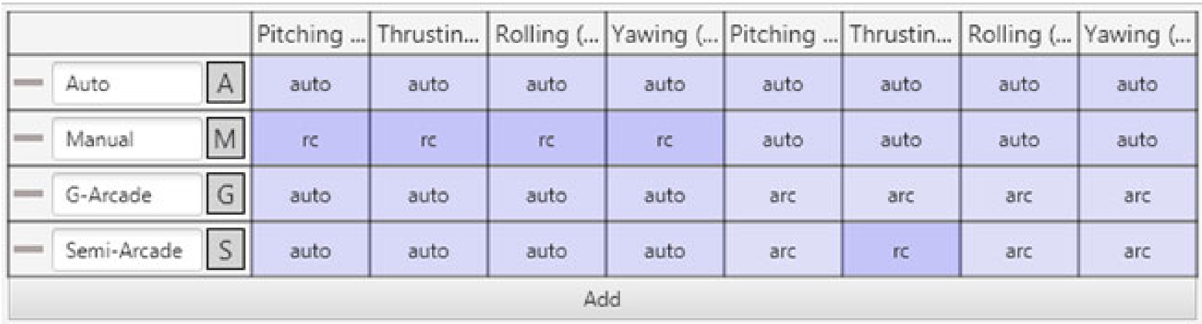
The other options that appear in this menu are parameters to limit when the smooth will be used or not.

- The first one establish the minimum velocity at which the smooth will start working,
- The second one is the minimum distance between the start and final points of the trajectory to begin with the smoothing.
- The last option is the minimum time between points to use smooth, considering that the aircraft flies at the current speed.

All these options are safety features to avoid a malfunction of the system when the final and start points of the phases are close to each other.

6.2.4.5 Modes

The flight modes determine who is in charge of controlling each one of the aircraft control channels. There are 5 different control modes and it is possible to combine them to create custom flight modes.

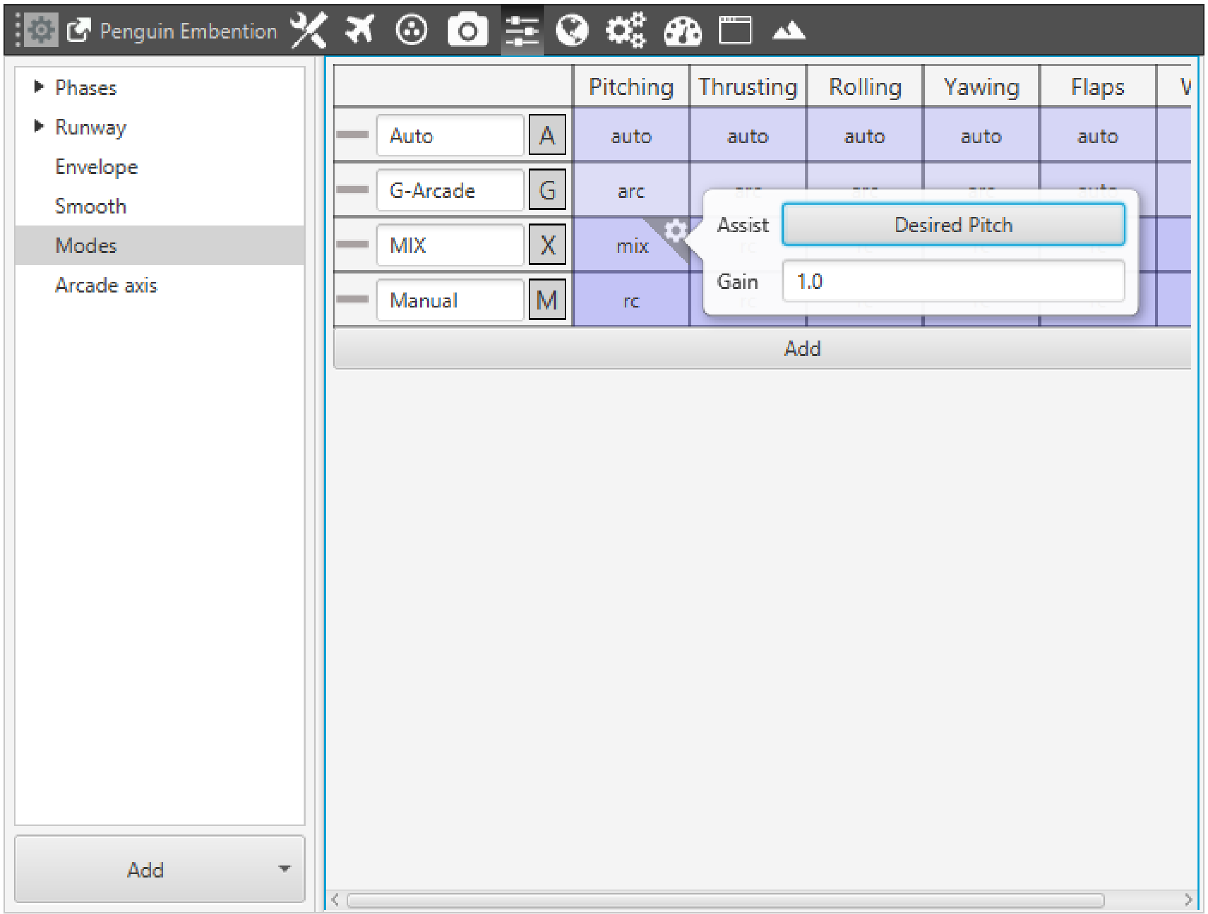


		Pitching ...	Thrustin...	Rolling (...	Yawing (...	Pitching ...	Thrustin...	Rolling (...	Yawing (...
Auto	A	auto	auto	auto	auto	auto	auto	auto	auto
Manual	M	rc	rc	rc	rc	auto	auto	auto	auto
G-Arcade	G	auto	auto	auto	auto	arc	arc	arc	arc
Semi-Arcade	S	auto	auto	auto	auto	arc	rc	arc	arc
Add									

Modes editing panel

The options available are:

- **Automatic:** the control channel is controlled totally by the autopilot.
- **RC:** the control is totally carried out manually. The movements on the pilot stick imply directly movements on the servo linked to that control channel.
- **ARC:** the autopilot aids the radio controller during the flight, i.e it could be considered as a mix between automatic and manual. The movements on the pilot stick are the input values on the control system, so the pilot commands a desired pitch, roll, IAS, heading and so on, and is the control system who is in charge of making the platform follow those commands.
- **Mix:** in this mode, it is possible to select in which step of the controller will enter the pilot command. For example, the pitching of an aircraft is commonly controlled with 3 PID being: flight path angle, pitch and pitch rate. In the arcade mode the pilot command will be a desired flight path angle that enters as input of the whole control system, but in the Mix mode is possible to select where we want the command to enter, so the pilot command could be pitch (entering in the second PID directly) or pitch rate (entering directly on the third PID). The control system will take this input as a disturbance that it wants to discard because the final objective is to match the input of the first PID (a desired flight path angle in this case), so the Mix mode can be used to make small corrections when the aircraft is following a route for example, where we want it to move slightly towards a certain direction by introducing a value directly on the roll PID. The following figure shows how to select the Mix mode and the parameters that have to be configured: the variable to control (Desired Pitch in this case) and the gain applied on the stick command (it goes from -0.5 to 0.5 so in this case, a gain of 1 implies a Desired Pitch that can go from -0.5 rad to 0.5 rad).



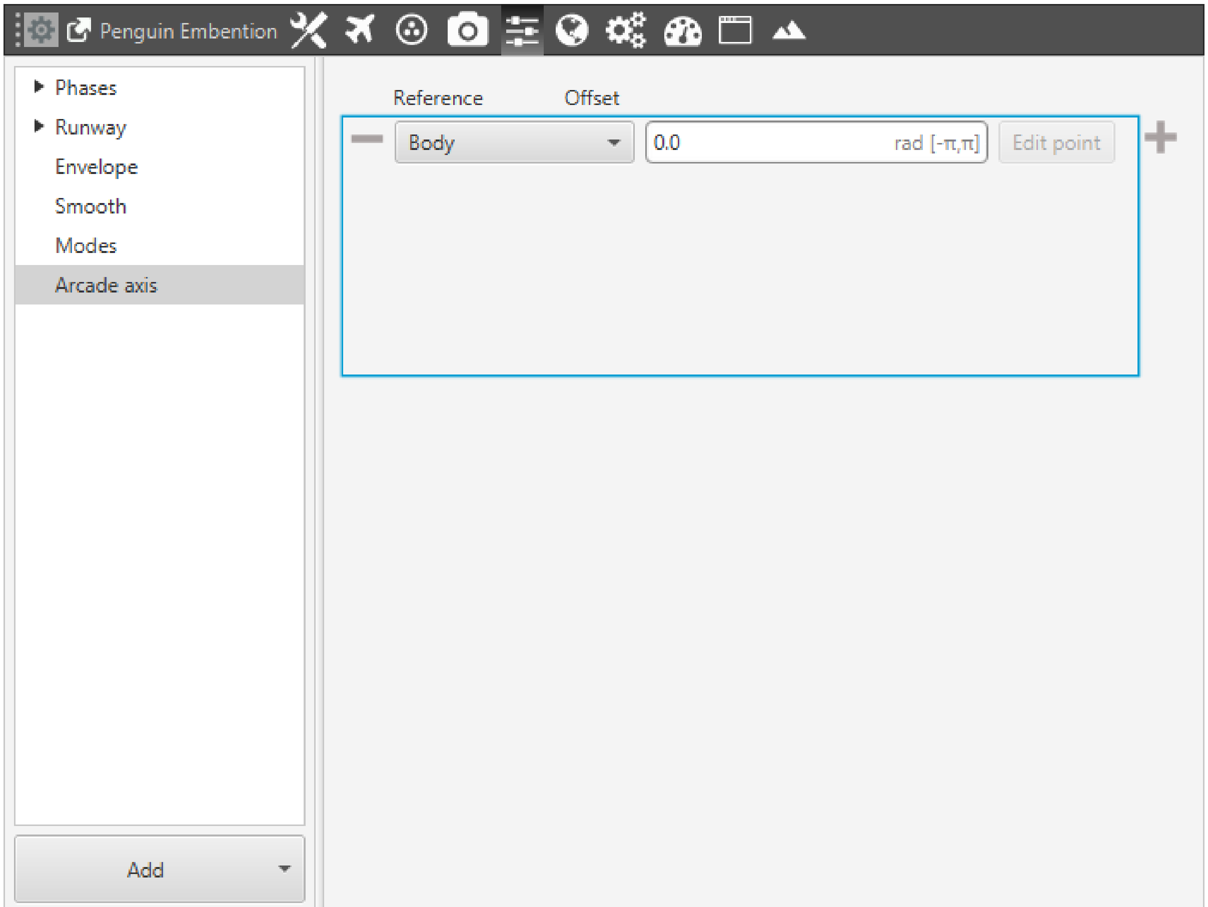
MIX Mode

- **Autotune**: this mode has to be select when the user wants to find control gains automatically.

It is every common to find an automatic mode where all the dynamics are controlled by the autopilot. Likewise, the manual mode is completely controlled by the remote controller (**rc**). To change any of this options, click on the cell you would like to change and the next option will be set.

6.2.4.6 Arcade Axis

The Arcade Axis menu enables the option of changing the center of the system axes. This option is useful, for example, when the pilot wants all movement to be made with respect him (Ground axes). In this way, if the pilot command a turn right, the aircraft will turn to the right of the pilot, instead the right of the aircraft (Body axes).



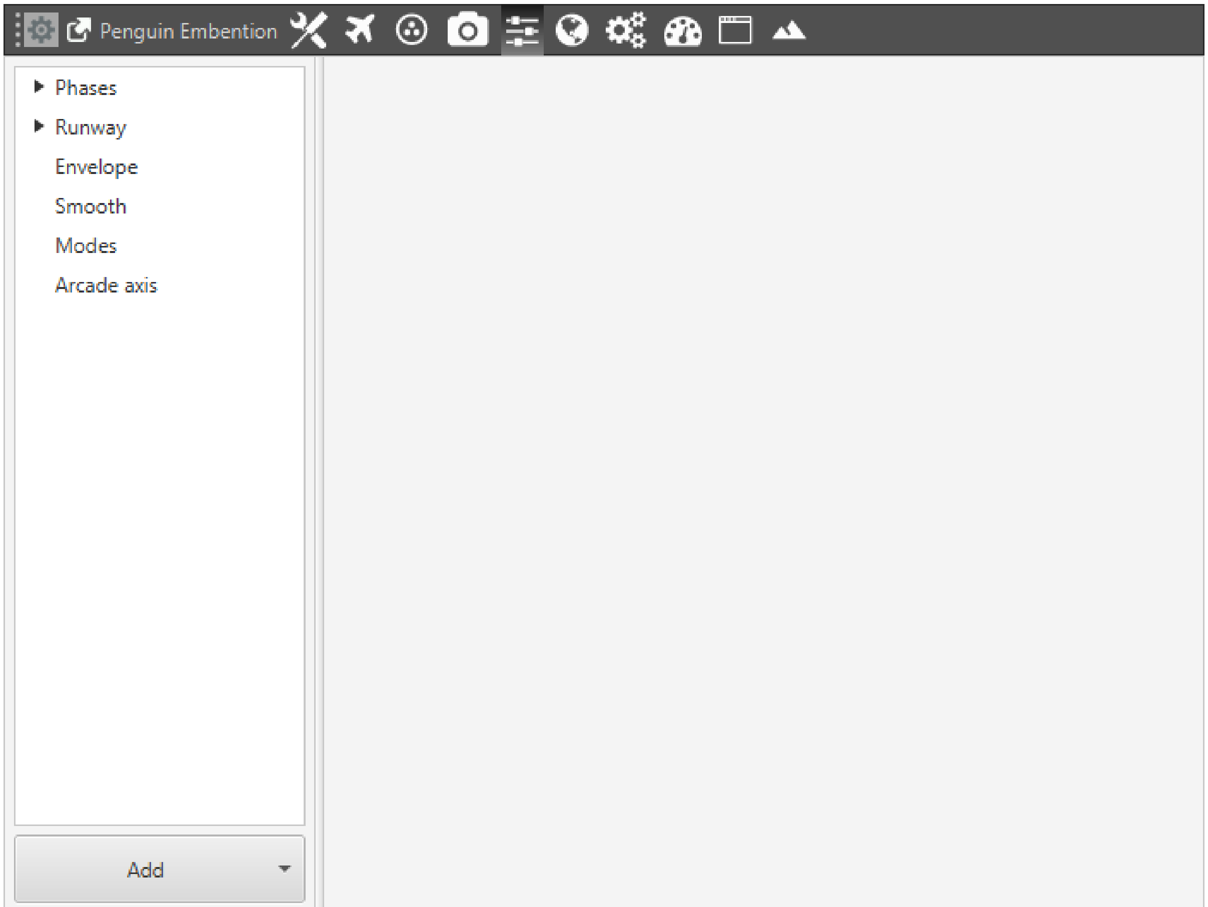
Arcade Axis Configuration Menu

It is possible to add as many axes system as desired, being able to choose between the following types:

- **Body:** fix the axes in the UAV. It is standard for the pilot.
- **Ground:** fix the axes in the Veronte GND.
- **Point:** fix the axes in a point that user defines.

An automation can be used to select an Arcade Axis in flight, see section *Select Arcade Axis*.

In this panel all the parameters related to the control of the platform can be found. There are 6 section, each one showing a different menu of configuration.



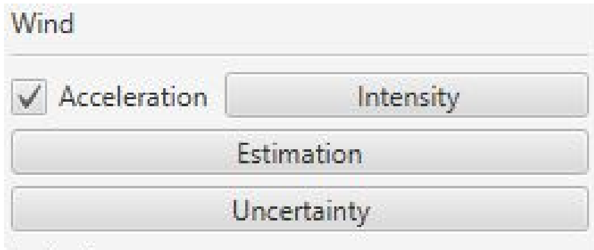
Control Configuration Menu

- **Phases:** in this section are created the Flight Phases that will control the aircraft at different satages of the operation. This section includes the Guiadances that are necessary, the type of Control used for each phase, the Arcade configuration when is used and the TC Panel for configuring in-flight parameters.
- **Runway:** this section allows user to configure a Runway or Spot used during the flight operation.
- **Envelope:** here are definied the flight limits, stall speed and the maximum climb and descent angles.
- **Smooth:** in this section is definied the trajectory of the aircraft when going from the final point of a phase to the beginning of the new one.
- **Modes:** allows the creation of custom flight modes, where each one of the control channels can be assigned a mode individually.
- **Arcade Axis:** this option is used to create axis systems referred to a certain point or direction.

6.2.5 Navigation

6.2.5.1 Wind Estimation

From the Navigation window can be configured the parameters that affects the wind estimation algorithm.



Wind estimation parameters

For proper estimation, the system needs to gather as much information of the wind as possible so missions with a trajectory involving changes on the of directions will result in better wind estimation compared with a straight trajectory mission.

Estimation Vector

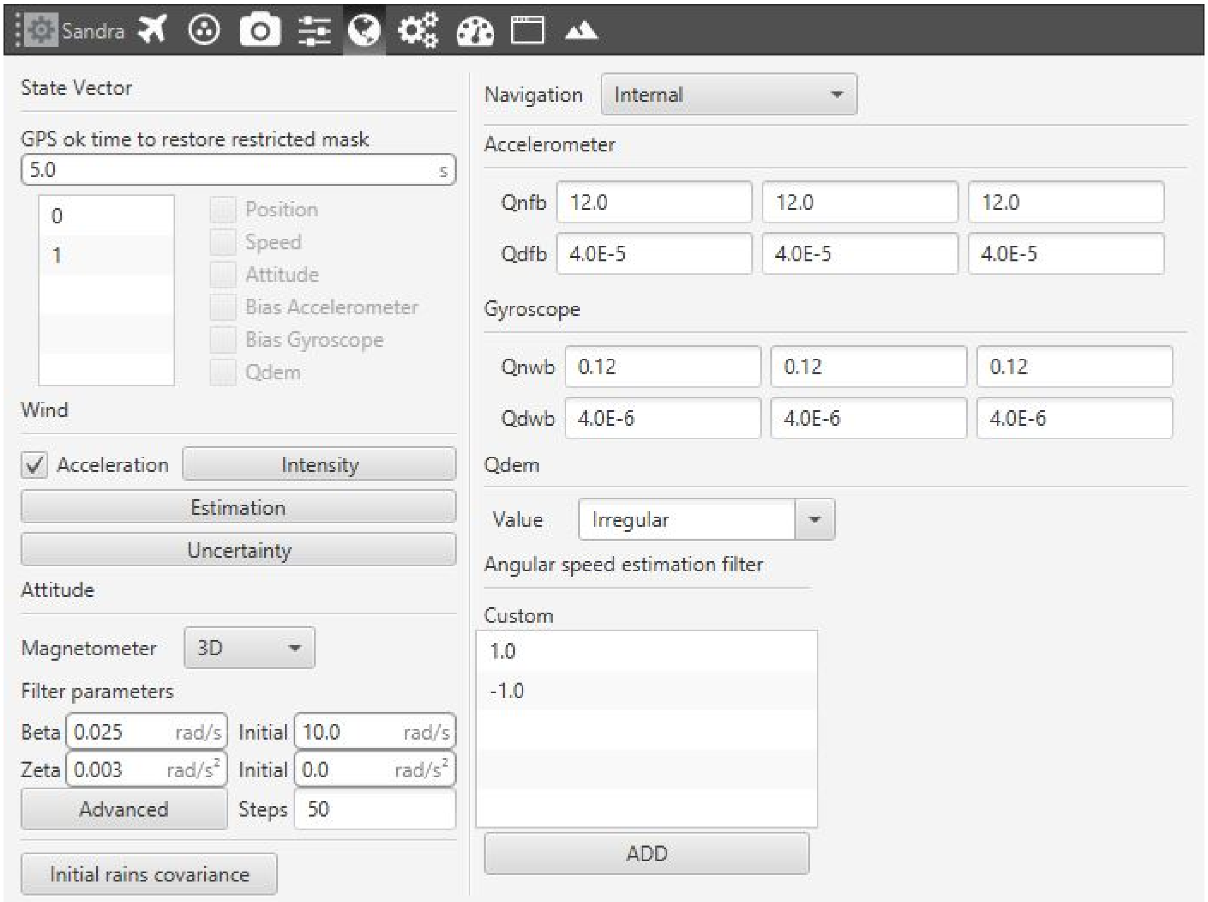
Intensity matrix

Uncertainty matrix

The computed result is displayed in the variables: **Wind Velocity Down, Wind Velocity East, Wind Velocity North.**

Warning: The values that appear here should only be changed by advanced users.

This menu contains the parameters used in the Kalman Filter algorithm to fuse the information provided by the different sensors. This data is used in the navigation system to generate the commands sent to the aircraft.



Navigation Parameters

Veronte integrates a navigation system which can operate with GPS and without GPS coverage. In the navigation with GPS, the system uses it to make the aircraft fly a route or towards a certain waypoint. It is possible to control the aircraft position (longitude and latitude) and the altitude. This is the navigation used by default, the one that the system uses when everything is working properly.

In case the GPS signal is lost, the navigation can easly measures the attitude angles with a greater precision than using a simple IMU. With these measures, it is possible for the system to control the pitch, roll and yaw and then keep a safe attitude when the GPS signal is lost, avoiding any possible malfunction. It is recommended to create an automation to change to a phase where the attitude angles are controlled, in the case of a loss of GPS signal. For more information visit *Automations*.

Note: the yaw can be measured in the navigation without GPS only if the magnetometer is activated in the navigation window.

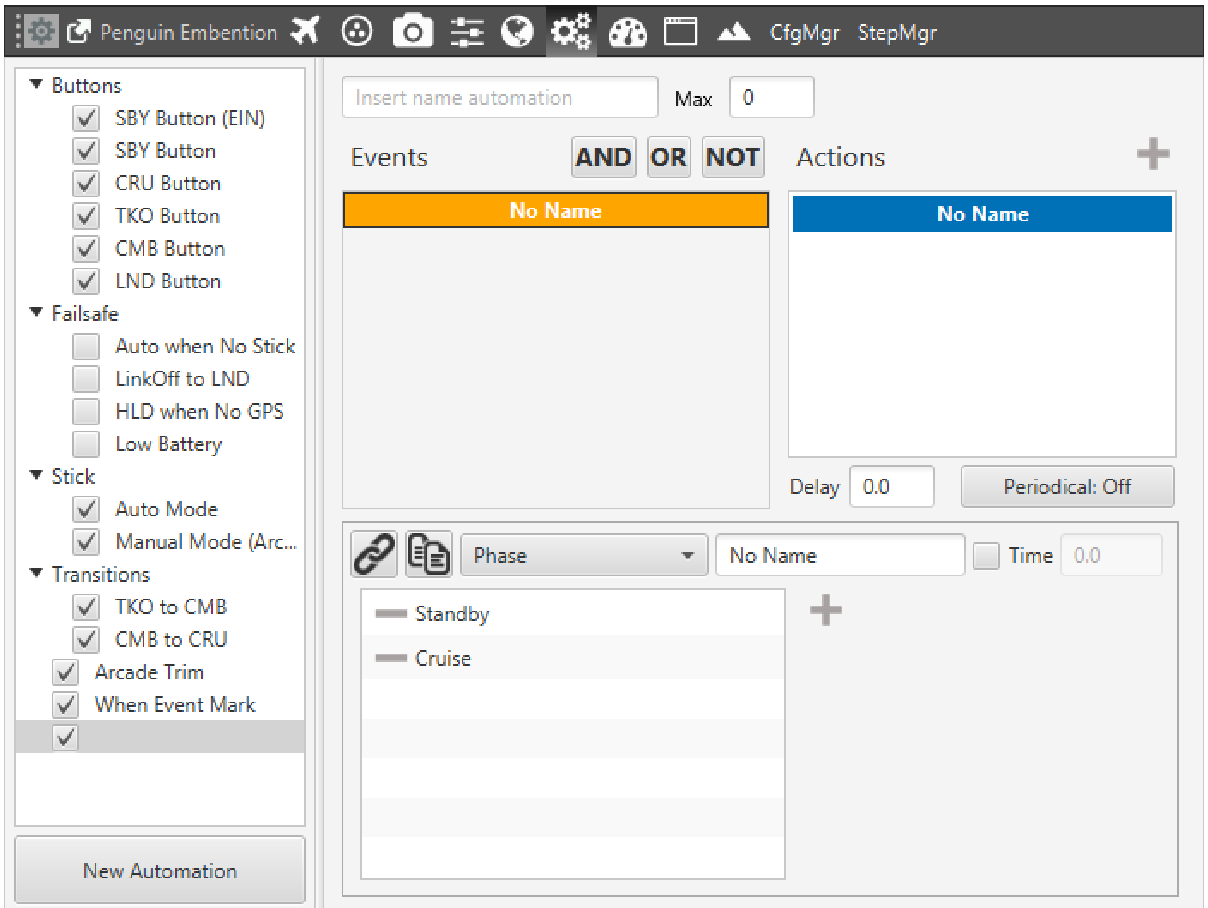
Warning: The values that appear here should only be changed by advanced users. If you are not familiar with the Kalman Filter algorithm and Sensor Fusion do not change the default parameters.

6.2.6 Automations

6.2.6.1 Events

6.2.6.1.1 Phase

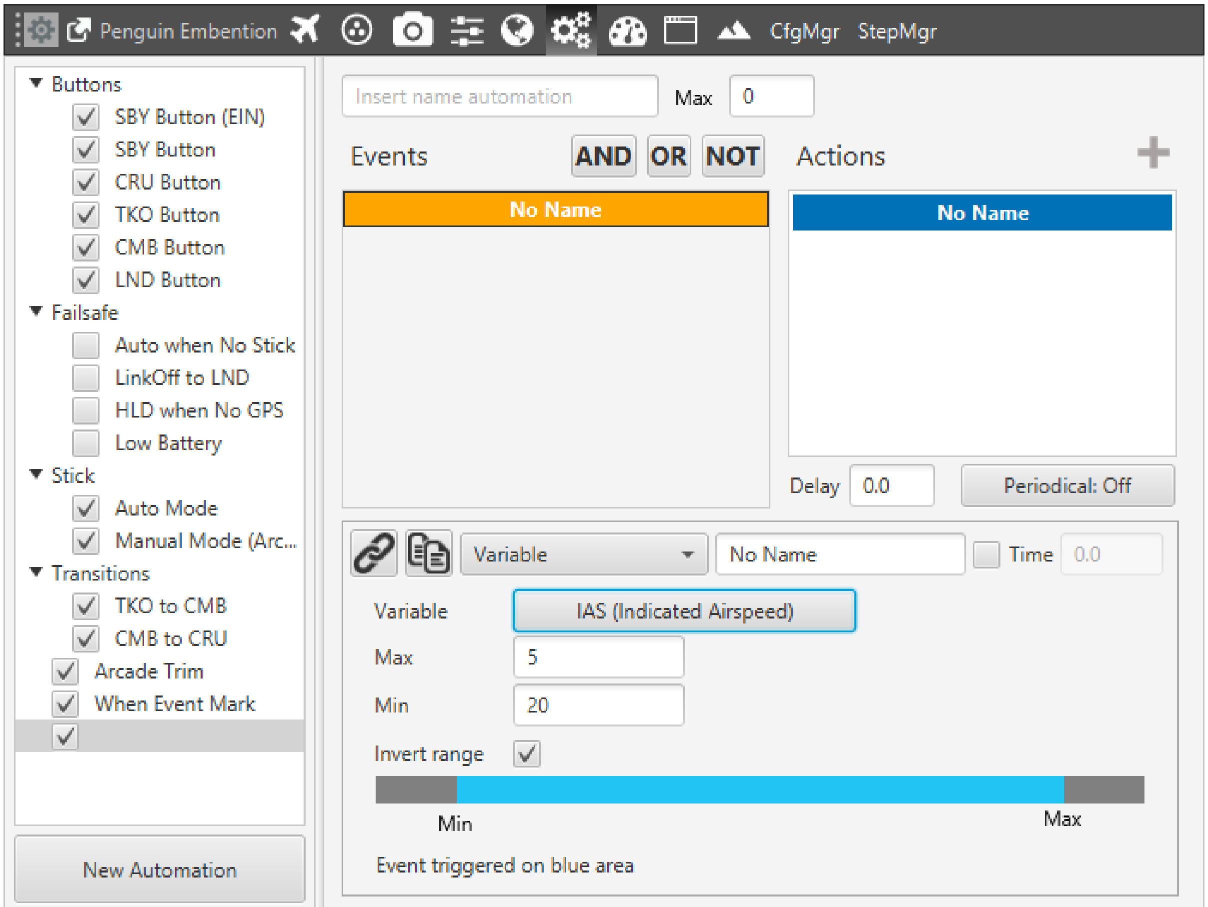
The event is triggered when the aircraft is in the phases selected by clicking on the "+" button, not in all of them at the same time, being in one of them is enough to trigger the action.



Event – Phase

6.2.6.1.2 Variable

This event is triggered when a variable selected is between a range established.



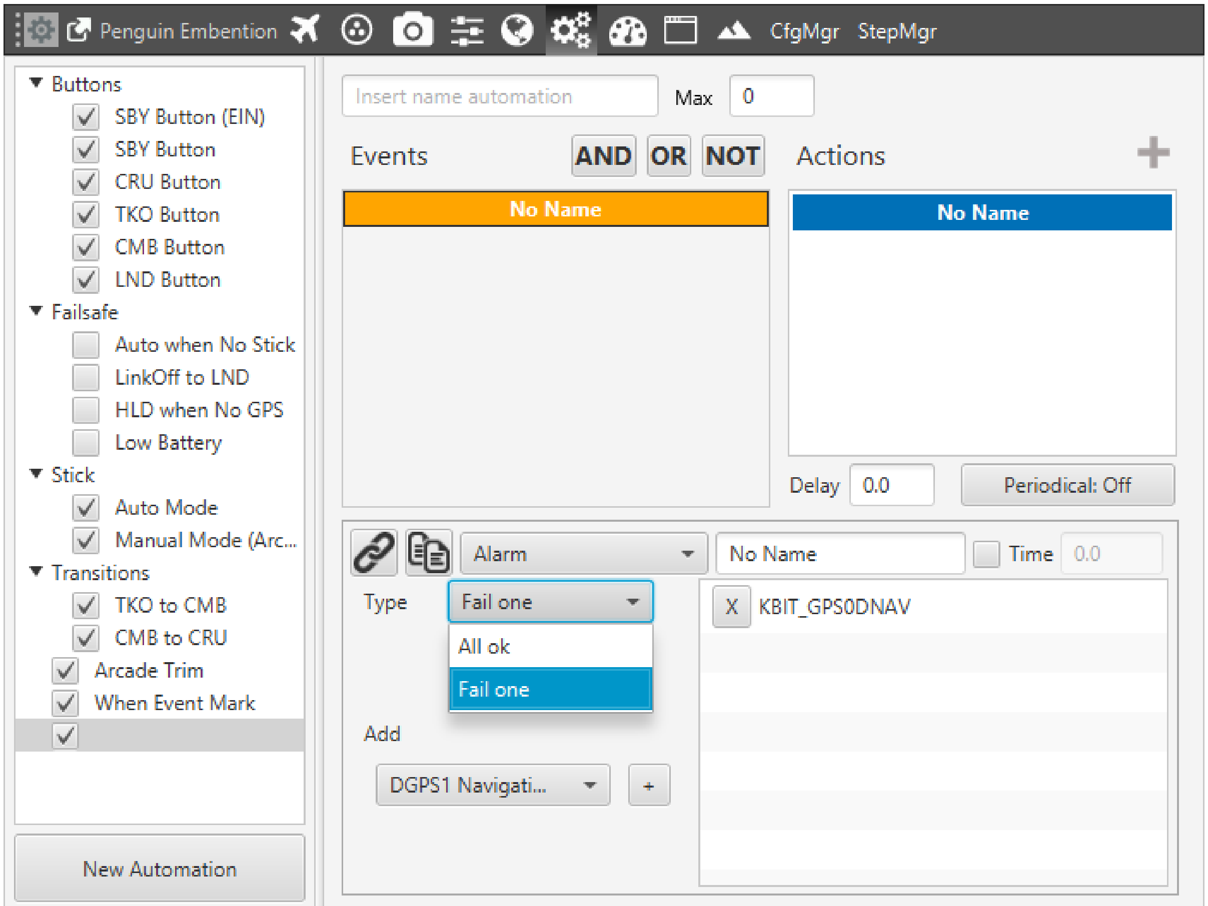
Event – Variable

- The variable to be evaluated is selected on **Variable**.
- **Max-Min:** maximum and minimum values of the threshold are established here.
- The option **invert range** will change the interval (the blue area will be gray, and the gray one will be blue).

As an example consider the event of the figure. With that parameters, the event is triggered when the IAS is between 5 and 20 meters per second. If the **invert range** option is unchecked, the event will be triggered when the IAS is lower that 5 m/s or greater than 20 m/s.

6.2.6.1.3 Alarm

When one of the elements selected in the **Add** menu fails or when all of them are working correctly, the event is triggered.



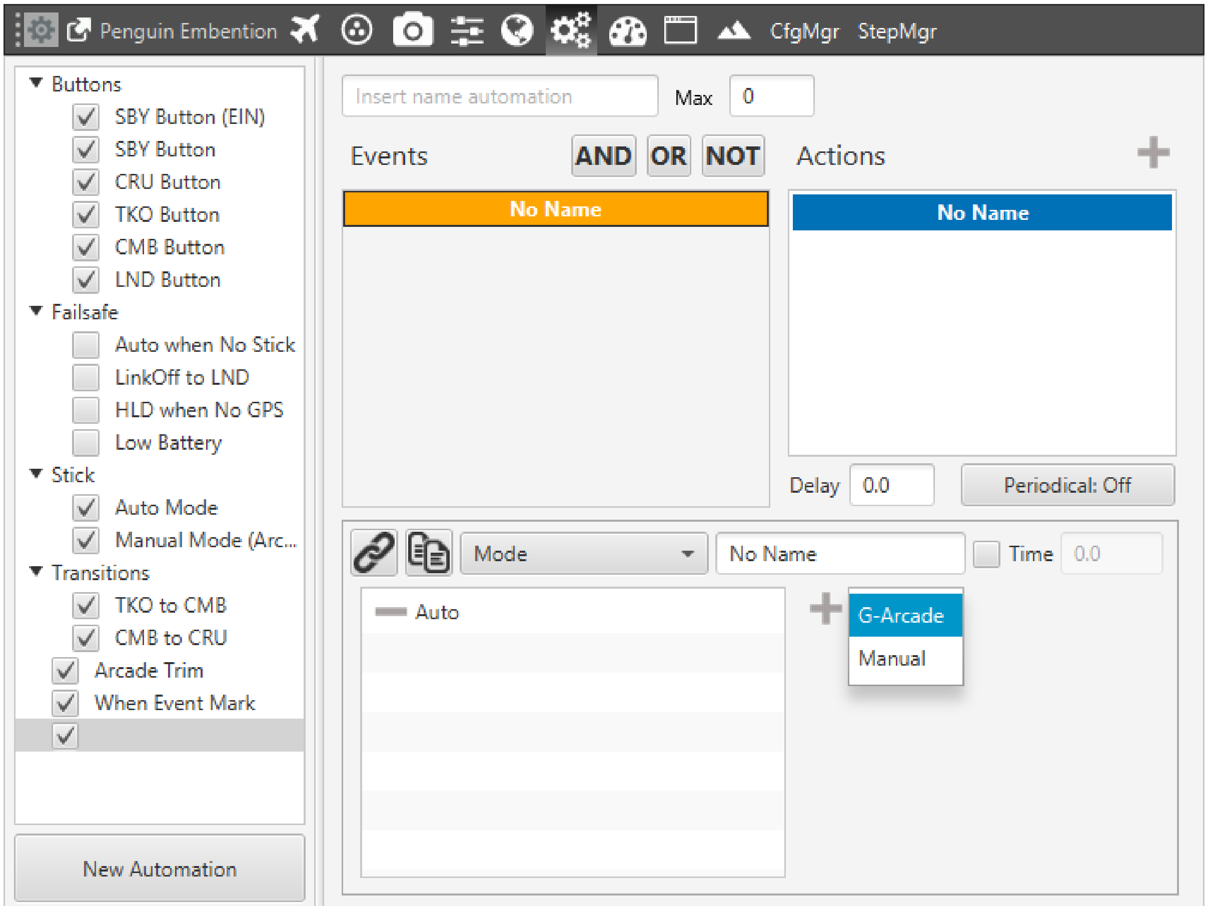
Event – Alarm

- The type options are **Fail** one and **All ok**.
- In the **Add** pull_down menu there are a series of parameters to include in the event.

A common alarm event is the **GPS Navigation Down**, that is triggered when there is not GPS signal in the autopilot.

6.2.6.1.4 Mode

The event is triggered when the aircraft is in one of the modes selected.

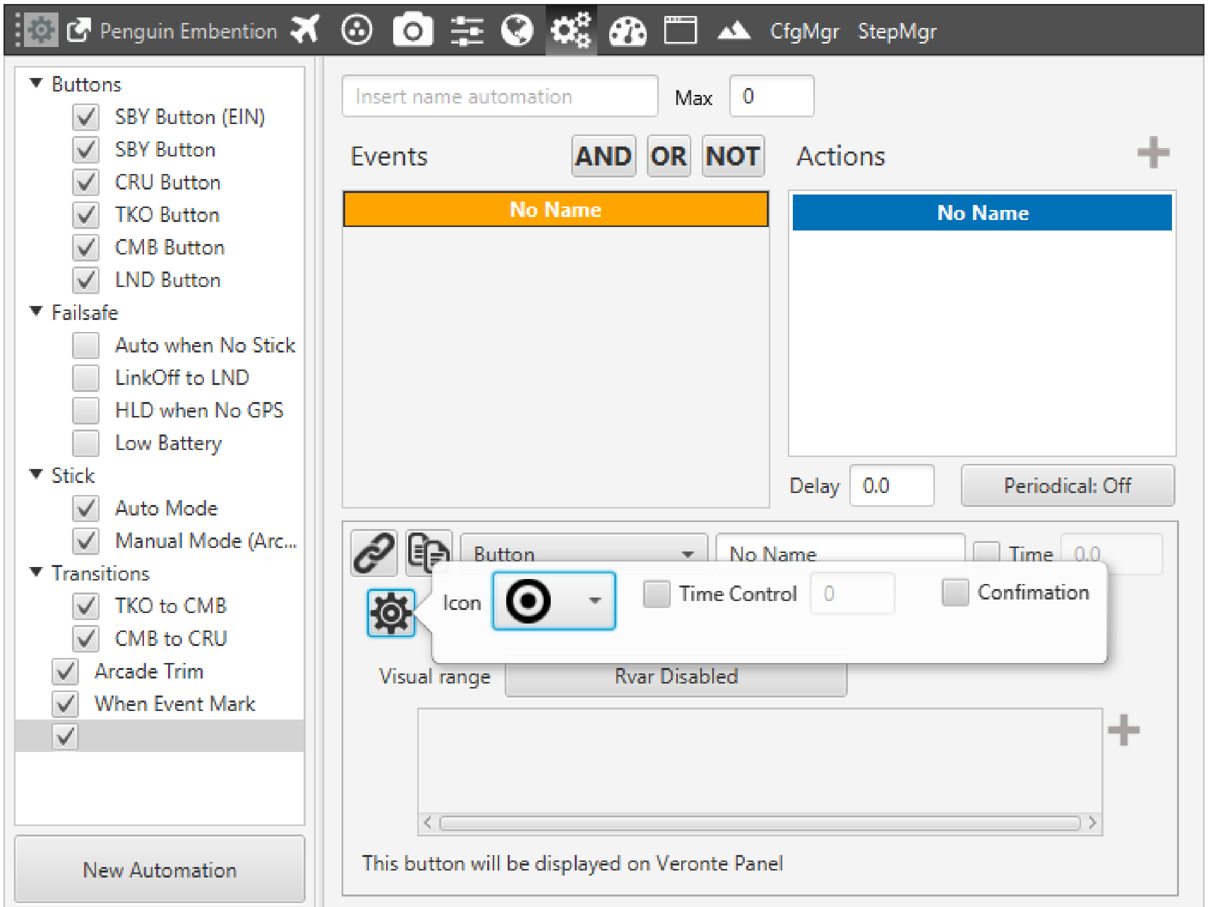


Event – Mode

These modes have been created previously. See section *Modes*, for more information about creating modes.

6.2.6.1.5 Button

This option creates a button that will trigger the event when it is clicked.



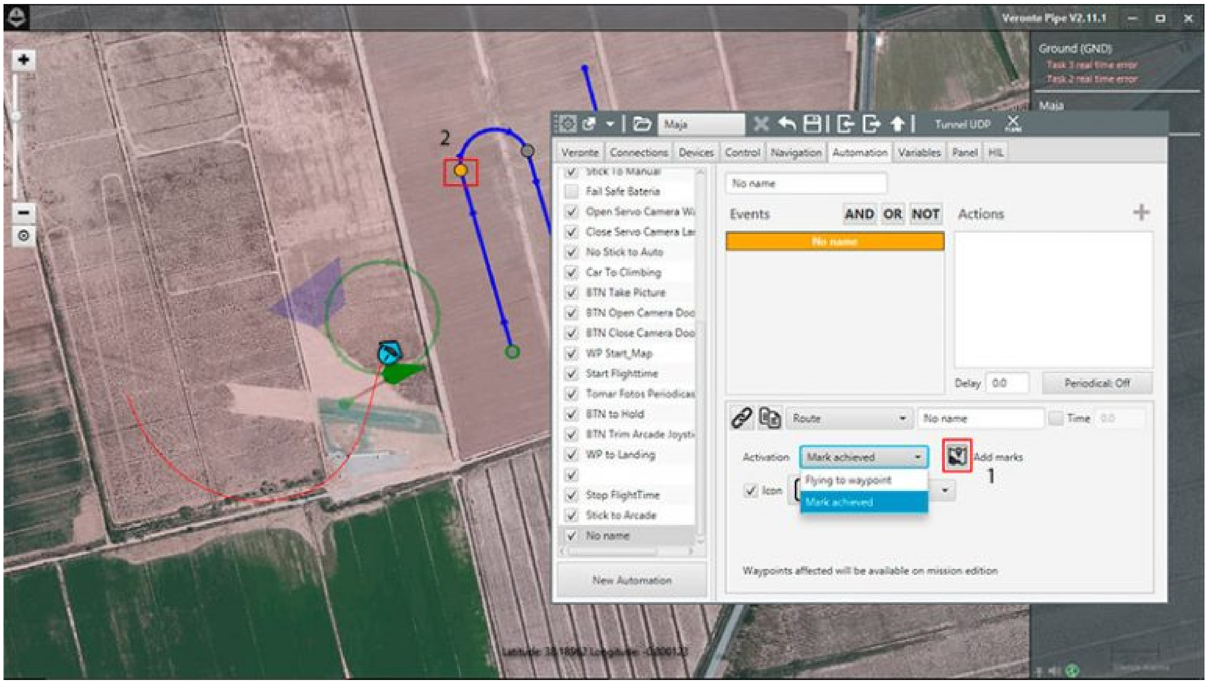
Event – Button

- **Visual range** option is used to make the button change its color according to the value of a variable. To do that, select a variable and then indicate as many points as desired, each one with its corresponding value and color.

If a button event triggers and action that consists of a change to a determined phase, the button will be the one of the Veronte Panel with the name of that phase on it. If the button event is linked to a different action (servo movement, variable…), it will appear in the lower part of Veronte Panel with the Icon selected by the user. The Confirmation option will display a pop-up window that asks for confirmation after pushing the button, so it is a safety measure. The **time** control option is used to trigger the action when the button is being pushed during the time specified in this option.

6.2.6.1.6 Route

This event is related with the waypoints defined by the user in the Mission menu.



Event – Route

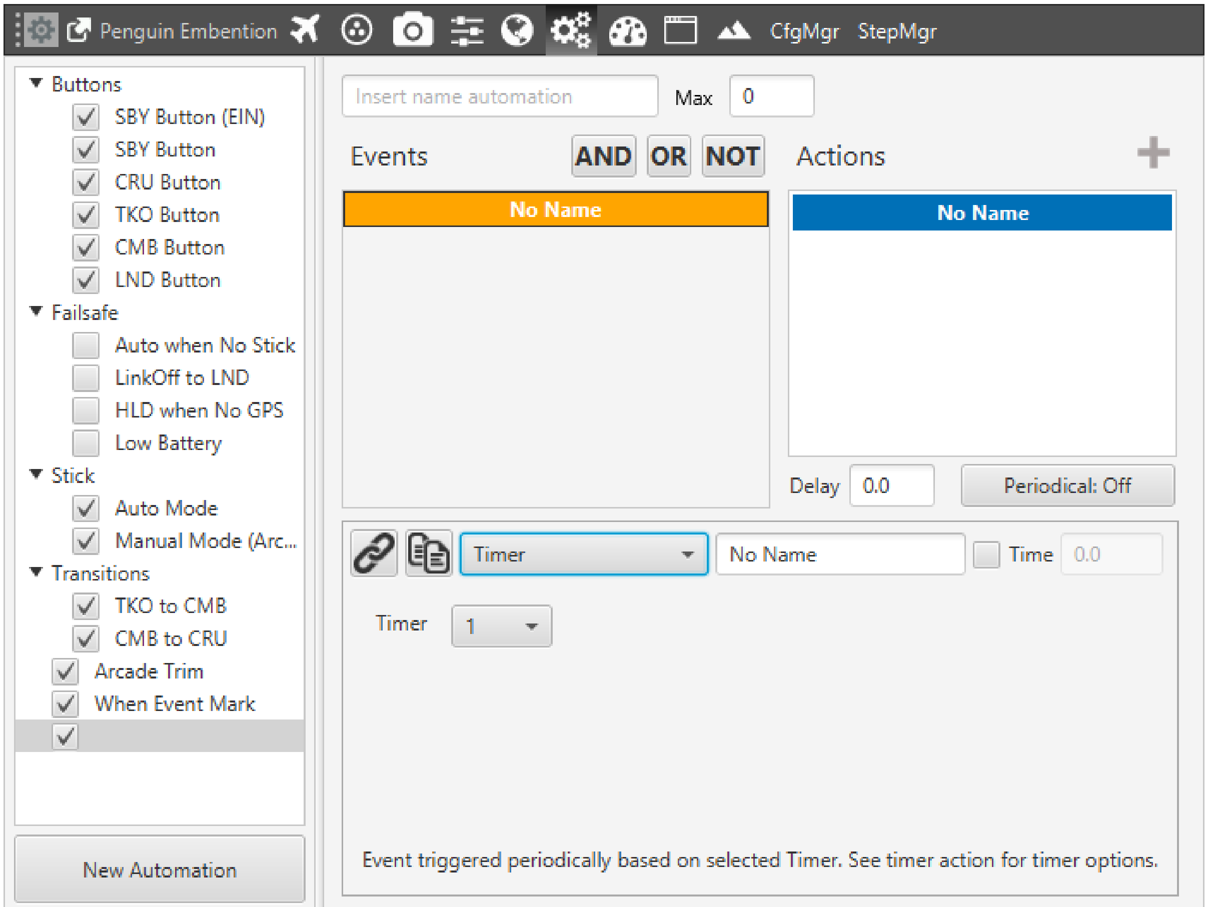
There are two modes in the event type.

- **Mark achieved:** triggers the action when the vehicle has reached the selected waypoint.
- **Flying to waypoint:** triggers the action when the platform is flying towards that waypoint (is the next waypoint of the route).

Clicking on **1** allows the selection of the waypoint among the ones created by the user (in this case **2** has been selected). It is possible to change the appearance of the waypoint to an image selected in the **icon** option, so the user can identify easily the waypoint linked to that automation.

6.2.6.1.7 Timer

This event will check the status of the timer selected in the menu. That timer should have been configured before on the action side of another automation ref (Action type *Periodical*).

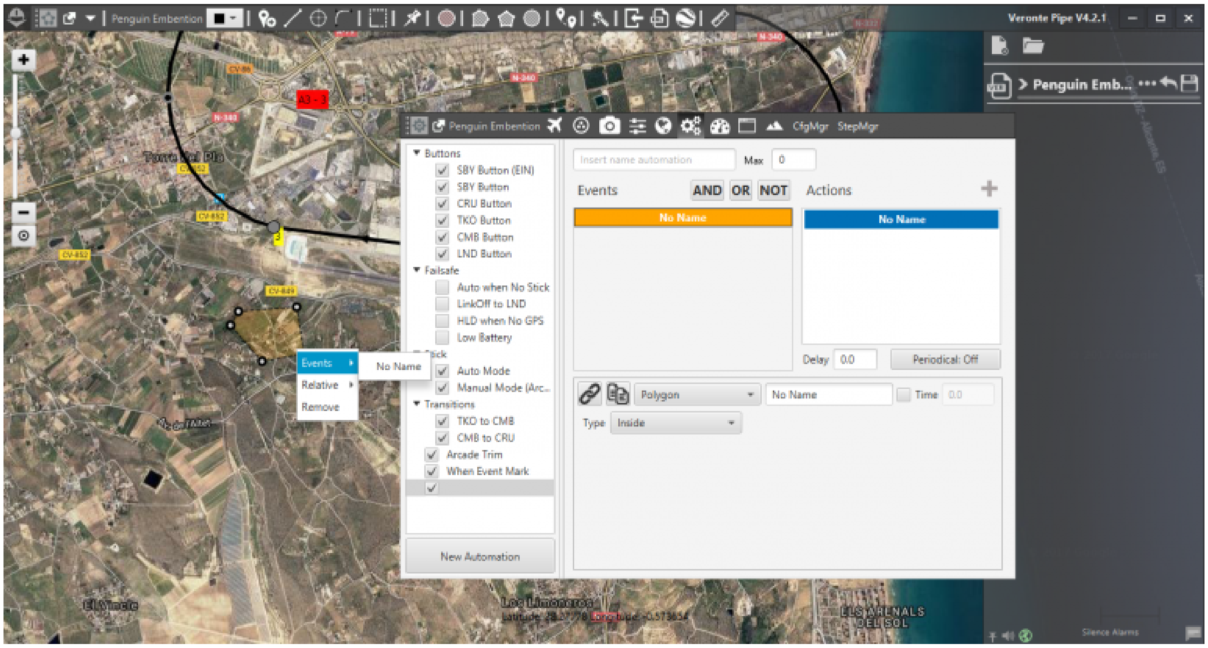


Event – Timer

For example: if it is desired to take a photo 10 seconds after the takeoff, one automation should have the event of Phase take off, with the correspondent *Periodical* action that will start a timer that lasts 10 seconds. Then with another automation, indicating in the event the timer created, an action is created to take a photo when the timer event is triggered. In Timer is selected the number that identifies the timer that is evaluated in this event.

6.2.6.1.8 Polygon

The event is triggered when the aircraft is inside or outside a polygon defined in the Mission menu.



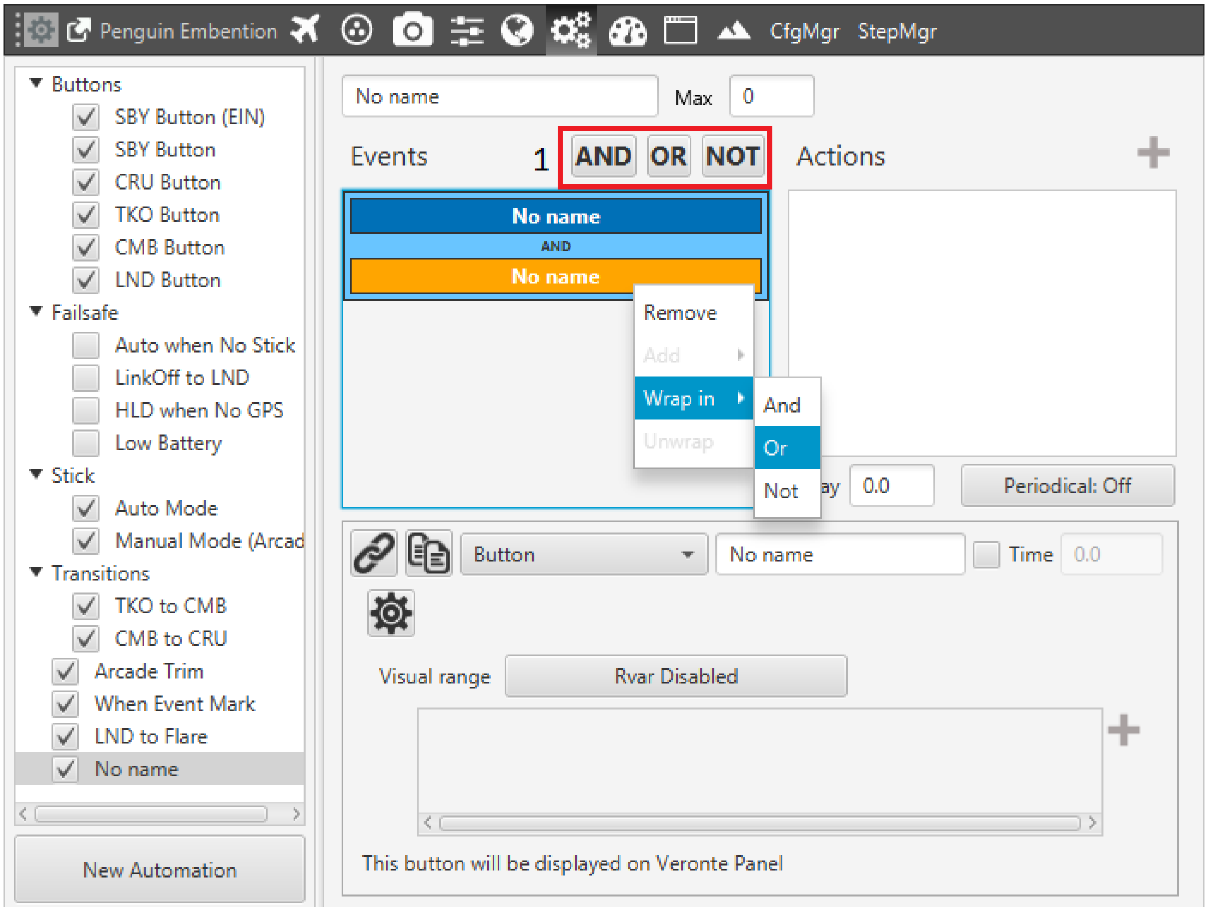
Event – Polygon

- When the event has been labeled (“Event Name (No Name)” in this case) and saved, it is then possible to link it to a polygon drawn on the map in the Mission menu.
- Right-clicking on the polygon will open a menu, and in *Events* will appear all the polygon-type events defined on the system. Click on it to link the polygon and the event.

An example about using this automation can be found in Circular Area.

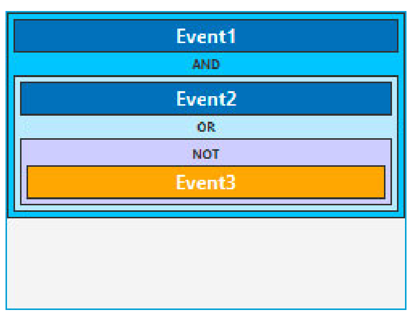
An event is something that has to be accomplished to trigger the actions. All the events can be combined to create a custom event, using the boolean operations provided by the software (AND,OR, NOT).

When entering a new Event or Action it is possible to choose from one of the previously created on the system or to create a new one.



Automations Panel – Events

When there is only one event, clicking on the boolean command (**1**) will create another event linked to the other one according to that operation. Right-clicking on an event and selecting **Wrap in** allows the creation of an operation as if it was inside brackets, i.e it will be evaluated first. Let's consider the following event group as an example.



Events

The first operation that is evaluated is the NOT, then the OR between Event2 and the result of the NOT, and finally the AND between Event1 and the result of the OR.

The following table depicts the meaning of each one of the boolean operators.

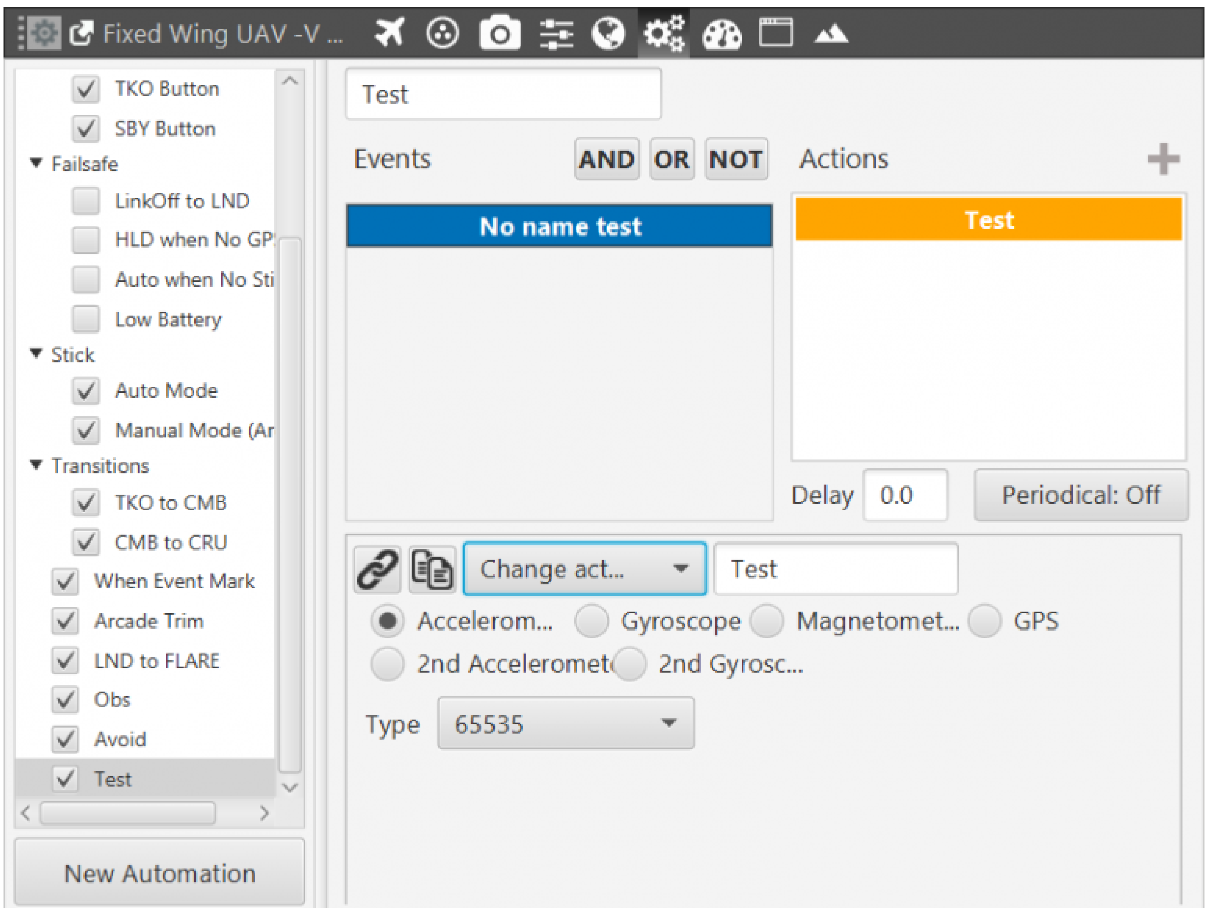
Logics	Description
AND	All events grouped on an AND should be accomplished simultaneously in order to activate the automation.
OR	One of the events in the group should be accomplished for activating the automation.
NOT	The event will be active meanwhile the event or event group is not accomplished.

When creating a new event, it possible to select different types of events, these are explained in the next sections.

6.2.6.2 Actions

6.2.6.2.1 Change Active Sensor

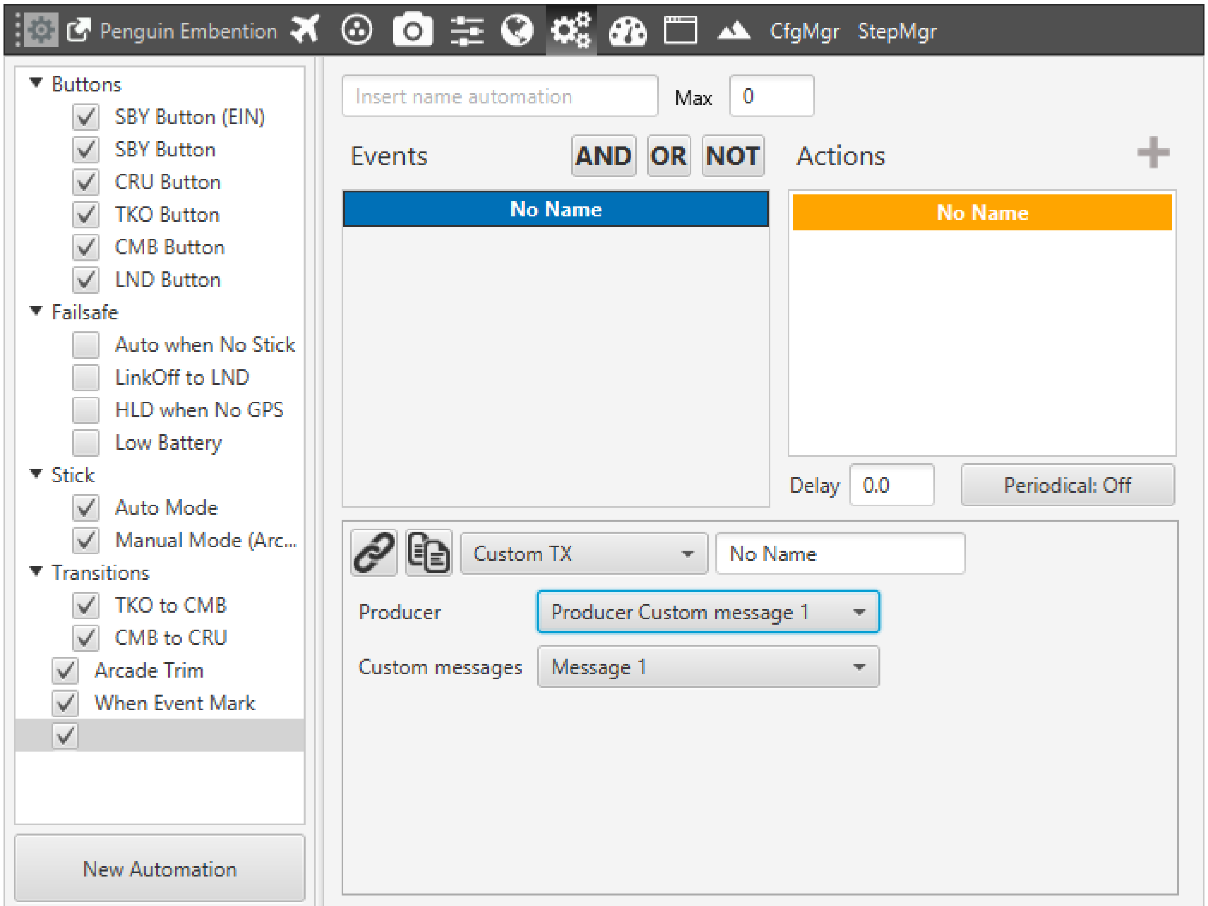
This option allows the change of the sensor used as accelerometer, gyroscope and magnetometer, between the internal and external ones. In addition, it is also possible to disable the selected sensor.



Action – Change Active Sensor

6.2.6.2.2 Custom TX

When this action is triggered, a previously configured message is sent trough the serial port of the autopilot. The message has to be configured in Custom messagess section.



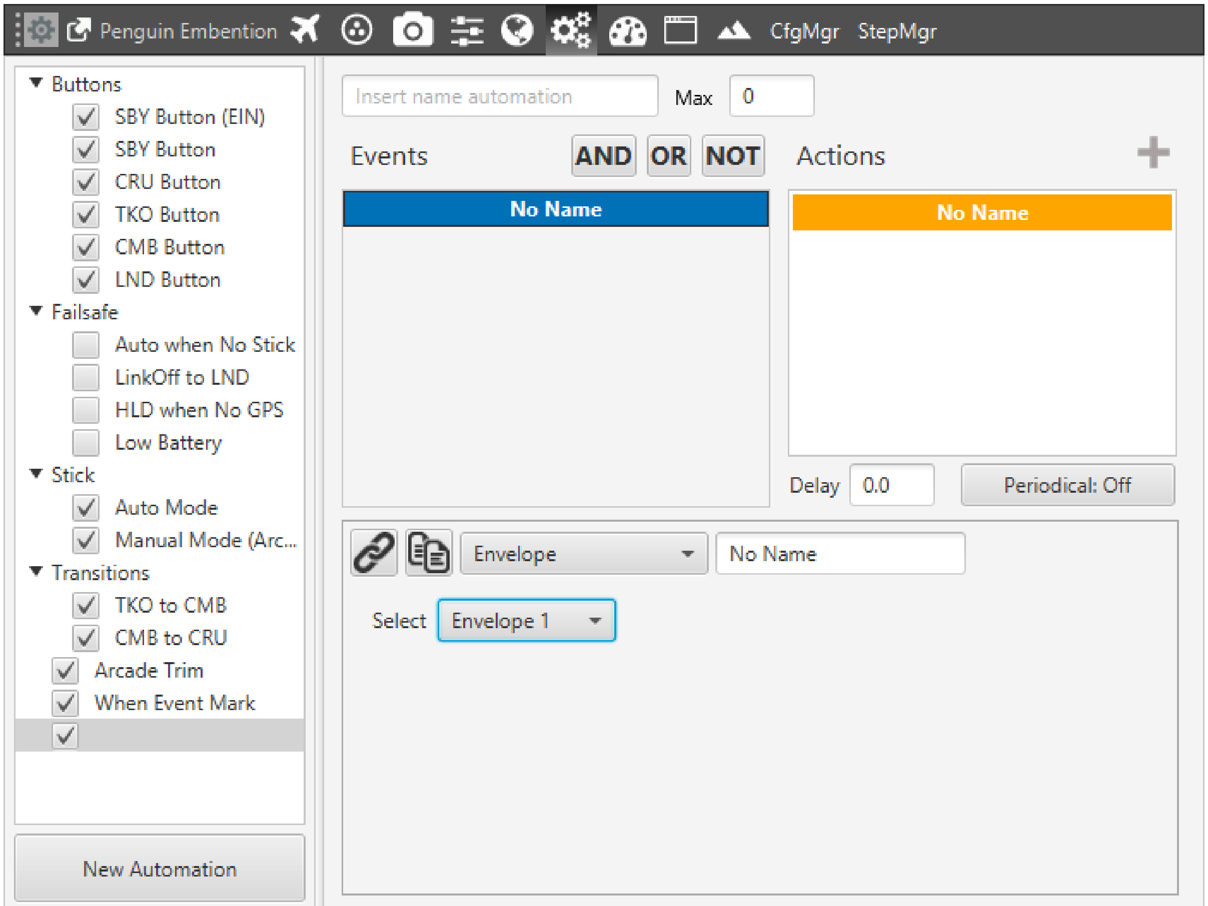
Action – Custom TX

The two parameters to configure in this action are:

- **Producer:** which is linked to a type of serial port SCIC, SCID (XPC Unit8 panel).
- **Custom message:** that will be sent (configured in the XPC Uint8 panel and Custom messages option).

6.2.6.2.3 Envelope

This action is used to change the envelope during the flight mission. Envelopes are created in the Control Panel, see section *Envelope*.

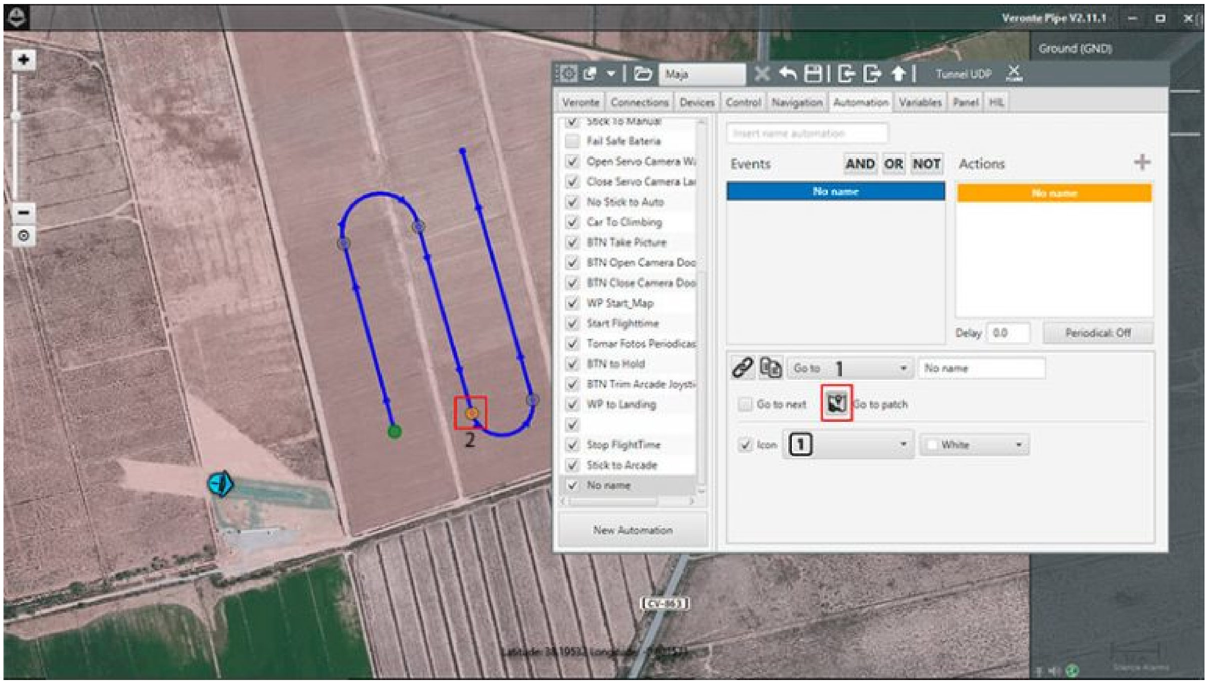


Action – Envelope

This is useful with a hybrid platform, being possible to change the envelope when the aircraft changes its configuration. Envelopes are selected from those created previously.

6.2.6.2.4 Go To

This action is used to make the aircraft go to a path (or waypoint if it is alone) of the route created by the user with the *Mission toolbar*.



Action – Go To

- Clicking in **1** allows the user to select a waypoint on the map (2 for example).

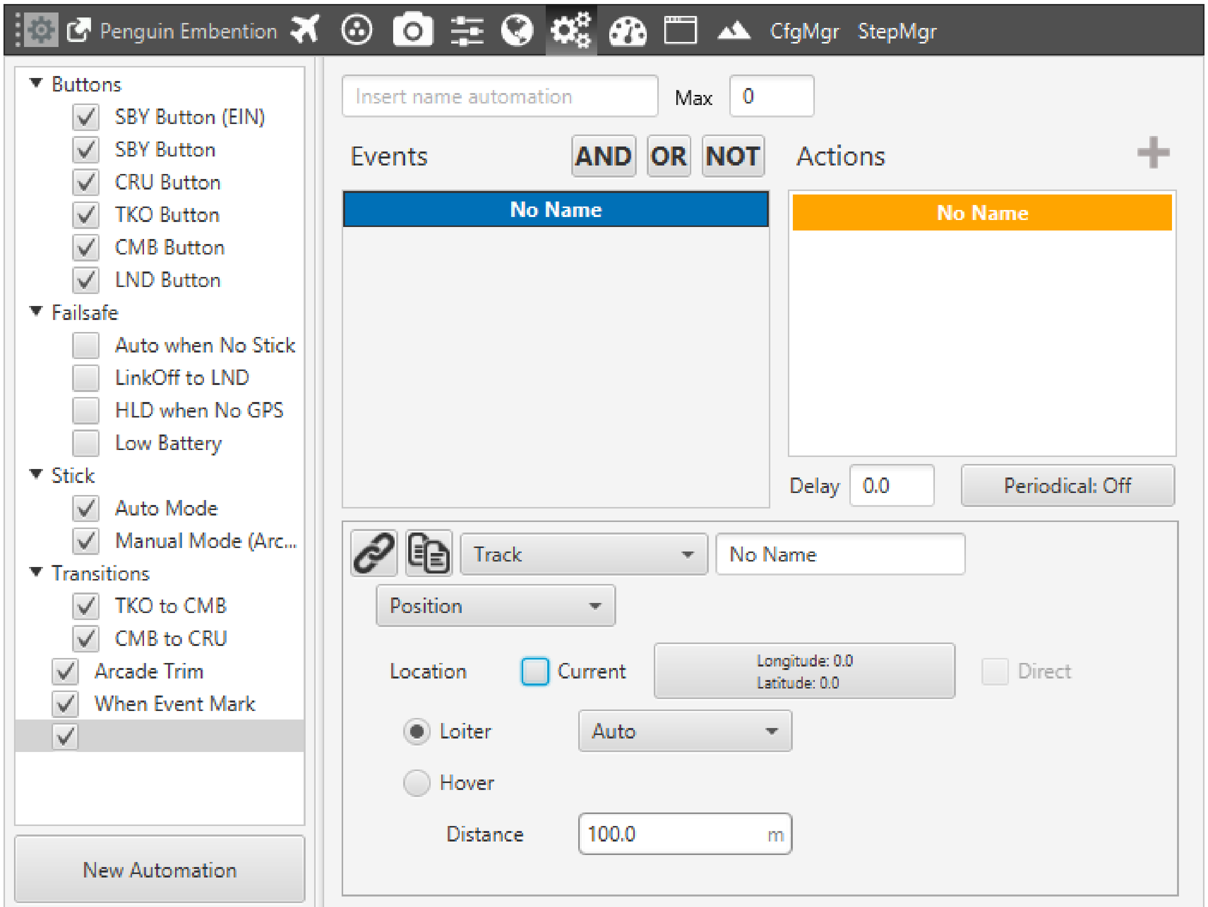
Once the action is triggered, the vehicle will go to that patch (or waypoint). If the patch is on a route, the vehicle will follow the selected patch and then it will continue the route going to its adjacent. On the other hand, if the option **Go to next** is selected, once the event is accomplished the aircraft will forget about the current path that is following and it will go to the next one. If the event happens again it will "jump" another patch and go to the following.

For example, considering the route that appears in the previous figure, if the automation **Go to next** is triggered by a button, when the aircraft is in the first patch (the one that starts on the green waypoint) and the button is pressed it will go to the next curved patch. If the button is pressed again before reaching that patch, the aircraft will go to the next straight line (parallel to the first one) without going over the curved patch.

It is possible to change the appearance of the waypoint to an image selected in the **icon** option, so the user can identify easily the waypoint linked to that automation.

6.2.6.2.5 Track

This action is used to configure a hover/loiter route (depending if it is a multicopter or an airplane) for the platform. Besides, there exists an option to fly the aircraft by the movement of a camera installed on it (fly **By cam**) and an option to follow a moving object.



Action – Track

There are our different options for the Track action, selecting **Disabled** no action will have effect on the guidance. The others are explained below.

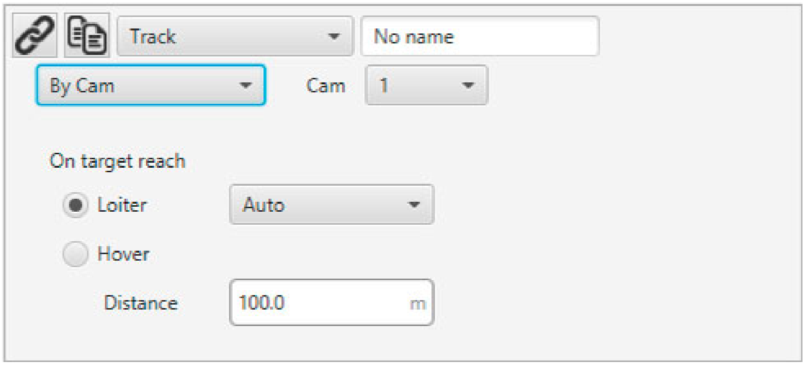
6.2.6.2.5.1 Position

The aircraft will loiter/hover in a selected point.

- Selecting **Current** will make the platform to hover over the position that the vehicle has when this action is triggered, or loiter around that point in a circular route with a radius indicated in **Distance**.
- It is also possible to select the direction of the loiter (**Auto**, **Clockwise** and **Anticlockwise**).
- On the other hand, the box (Longitude, Latitude)in the figure allows the user to select the point where the hover/loiter will be performed.

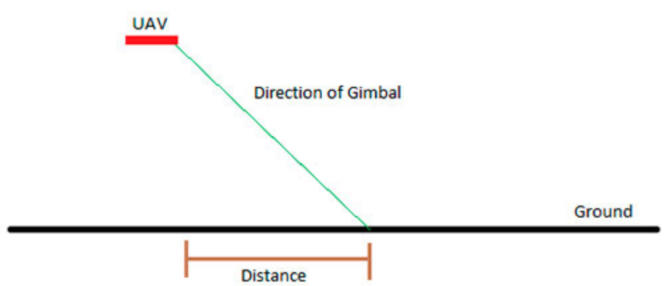
6.2.6.2.5.2 By cam

The direction of flight is controlled by the movement of a camera.



By CAM

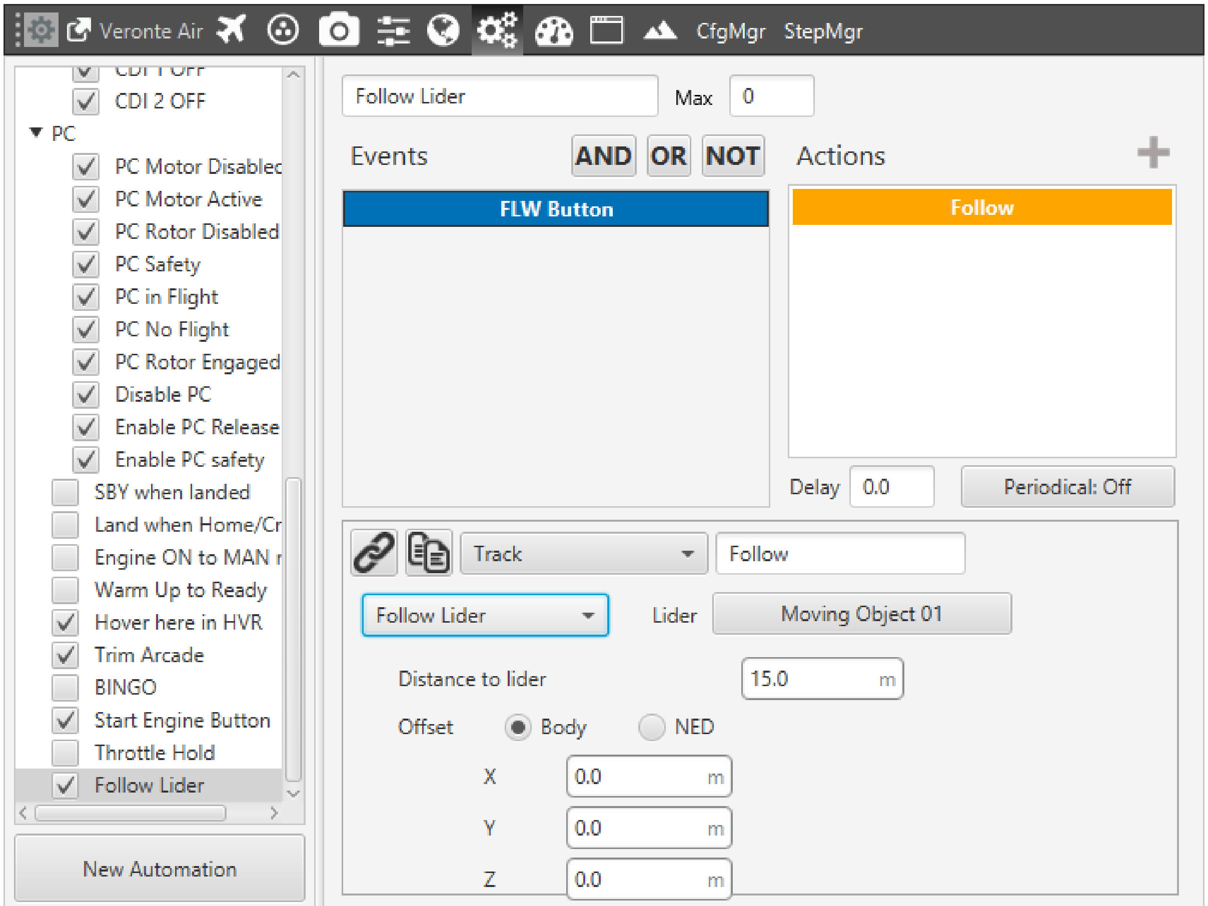
Let's consider that there is a virtual stick on the workspace that controls the movement of a gimbal that contains the camera. The horizontal movement of the stick that controls the yaw axis of the gimbal will act directly on the yaw axis of the aircraft. Considering that the longitudinal axis of camera and vehicle are lined up, the aircraft will point all the time to the same direction as the camera. On the other hand, the vertical movement of the stick will control the vertical axis of the gimbal. When the direction of the gimbal intersects with the ground, the vehicle will stop (or loiter) when being at a certain distance away from that intersection point (that distance is indicated by the user in **Distance**). If the vehicle is hovering because the distance is reached, once the gimbal vertical axis is risen again the vehicle will move again in the direction indicated by the gimbal. The following figure explains what has been said in this paragraph.



By CAM Distance

6.2.6.2.5.3 Follow Leader

The platform (Multicopter) will follow an moving object.



Follow Leader

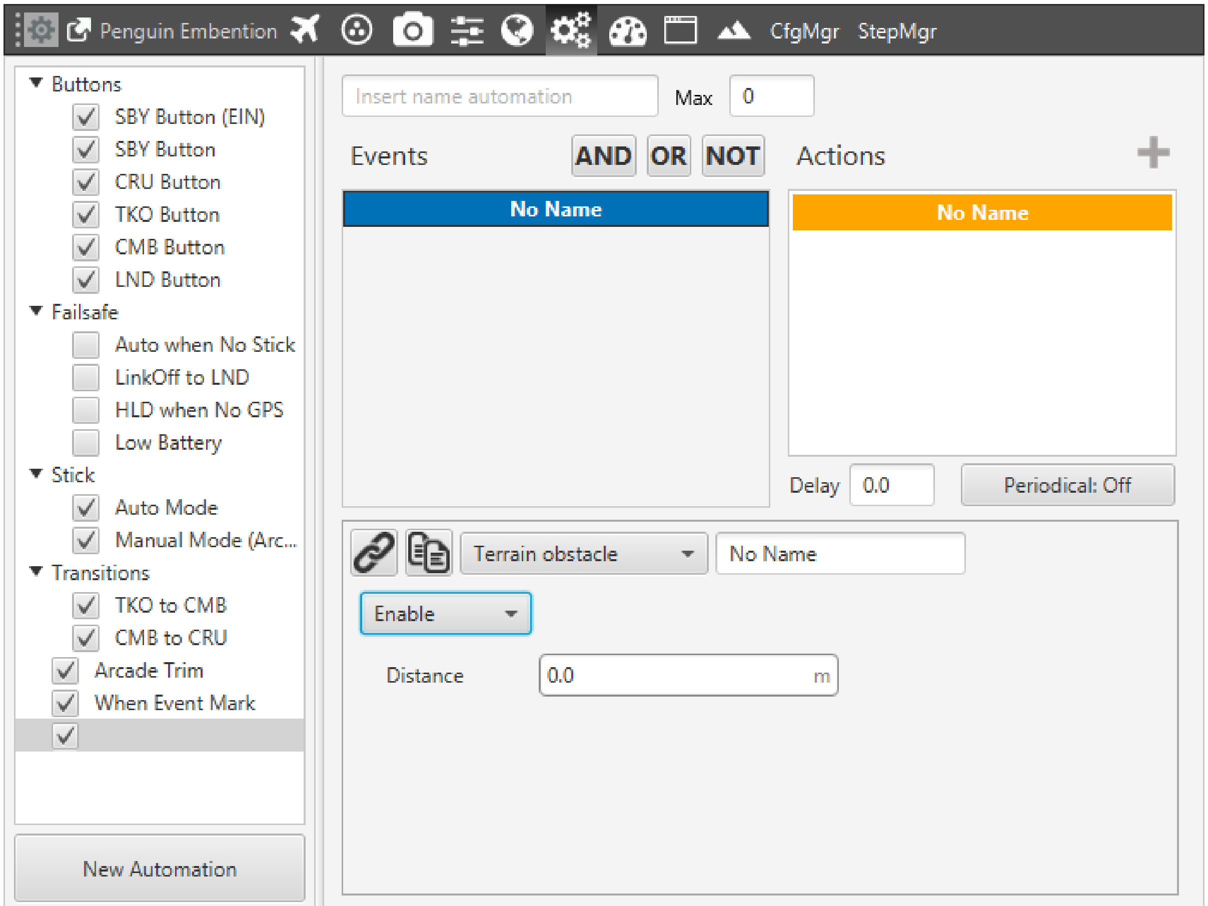
- **Leader:** here is selected the moving object i.e, the object to follow.
- **Distance:** distance to lider over trajectory.
- **Offset:** user can establish offset parameters related to trajectory in Body or NED coordinates.

To configure correctly this automation, user has to follow the next steps:

- Configure Telemetry Air and Ground.
- Configure the automation as desired.

6.2.6.2.6 Terrain Obstacle

This option is used to make the aircraft climb when is reaching an altitude of zero meters, for example, when flying towards a mountain. This option is not activated all the time because it will not allow the aircraft to land.

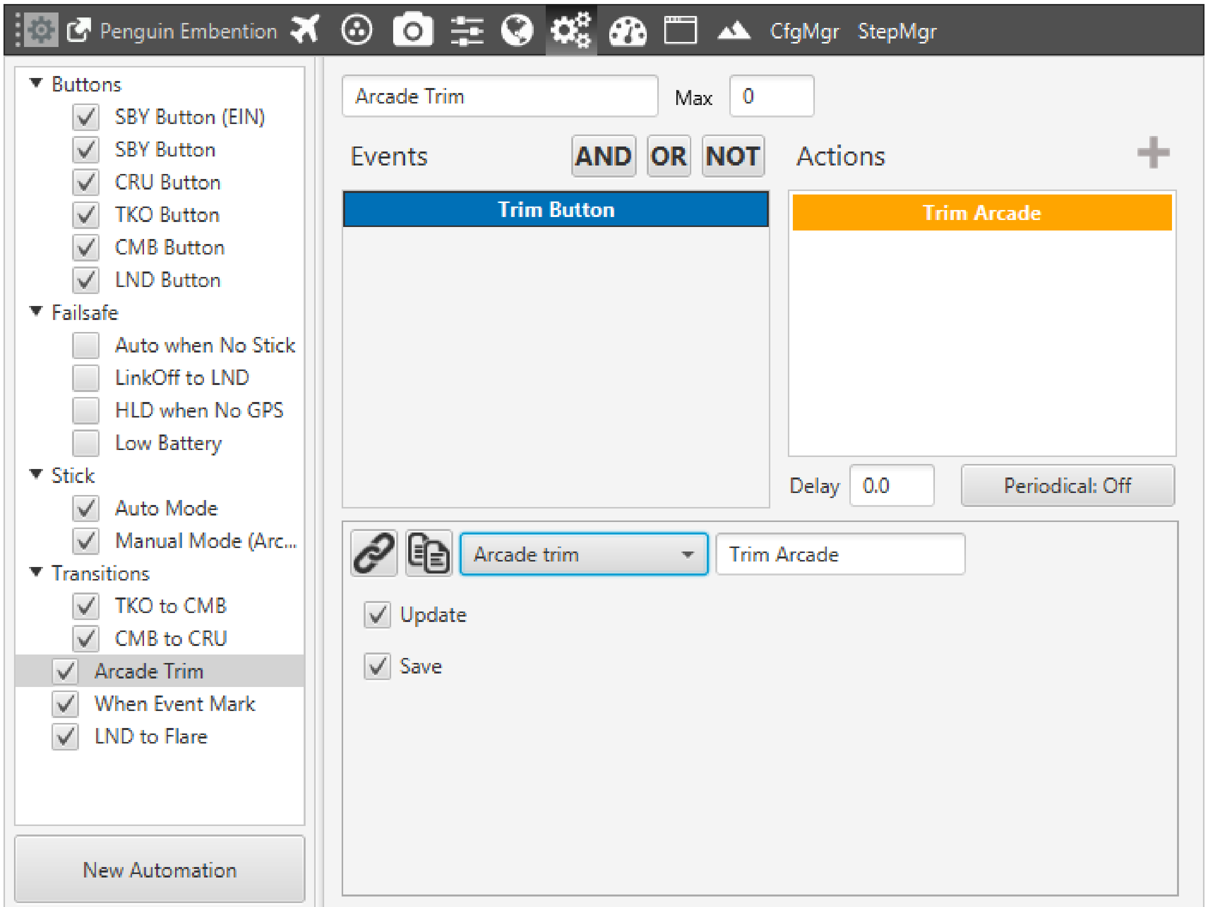


Action – Terrain Obstacle

- **Distance:** establish how the aircraft will climb, it can be said to be a repulsion value. High values made the platform ascent quickly. This effect is more noticeable when the aircraft is close to the ground.

6.2.6.2.7 Arcade Trim

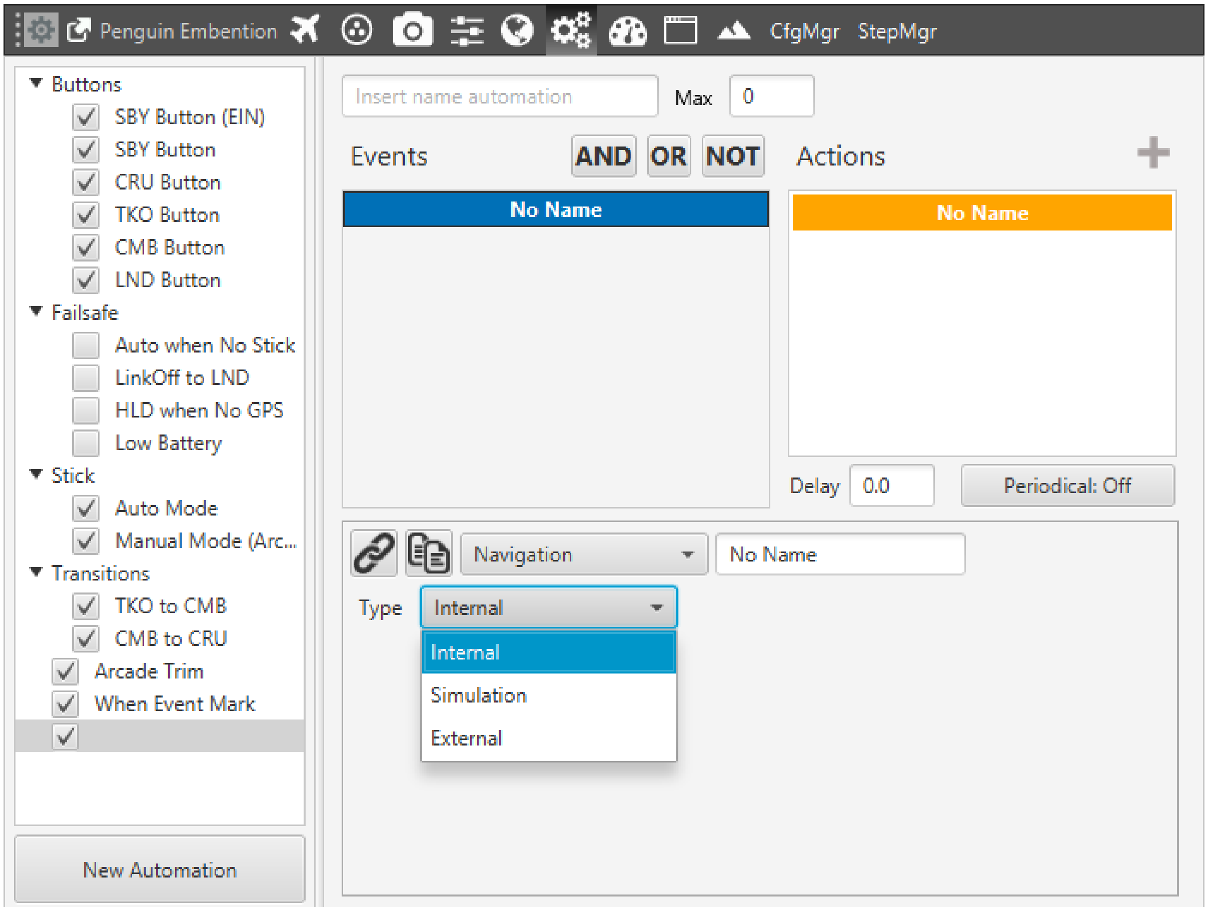
This action trims the radio controller, i.e sets as zero the current sticks positions. As seen in te picture below, it consists on simply creating a *Button* on the **Events** side and adding an **Action** which is called *Arcade trim*. The latter action is already configured to copy the current stick position into the the trim vector.



Action – Arcade Trim

6.2.6.2.8 Navigation

This action is used to change the navigation mode used by the aircraft. By default, the UAV uses a sensor fusion Internal algorithm, but for example, if the GPS falls, this algorithm produces bad results so it would be convenient to change to another type if that happens (External). The navigation without GPS will make the aircraft fly stable but it will not be possible to command a path to follow during that time, so it can be used as a safety mode to avoid a malfunction of the system when the GPS signal is lost.



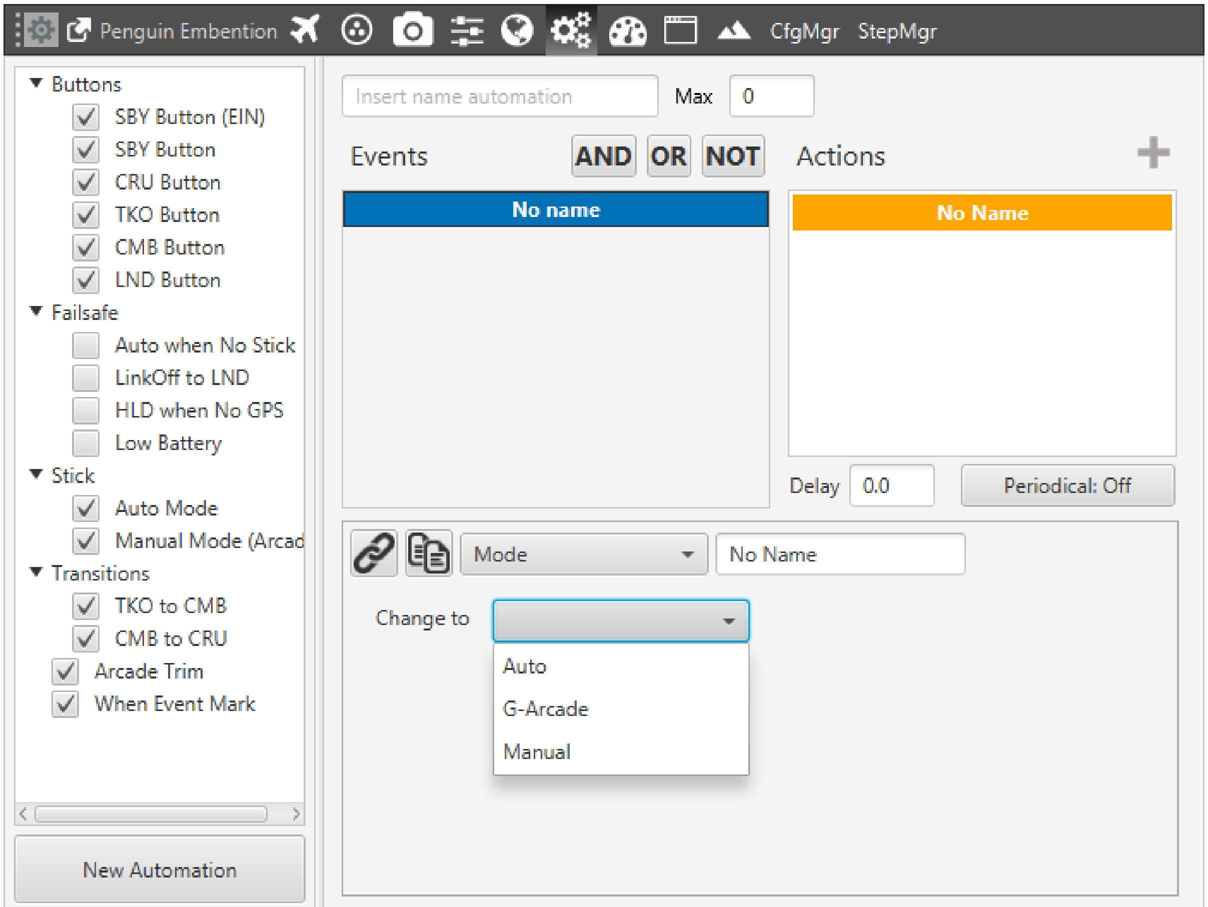
Action – Navigation

The options available are:

- **Internal:** uses internal data for navigation. Data (position, attitude, etc.) is processed into Veronte Unit from sensor measures.
- **External:** uses external data for navigation. Data (position, attitude, etc.) is provided by an external device.

6.2.6.2.9 Mode

The flight mode is changed to the one specified in this option.

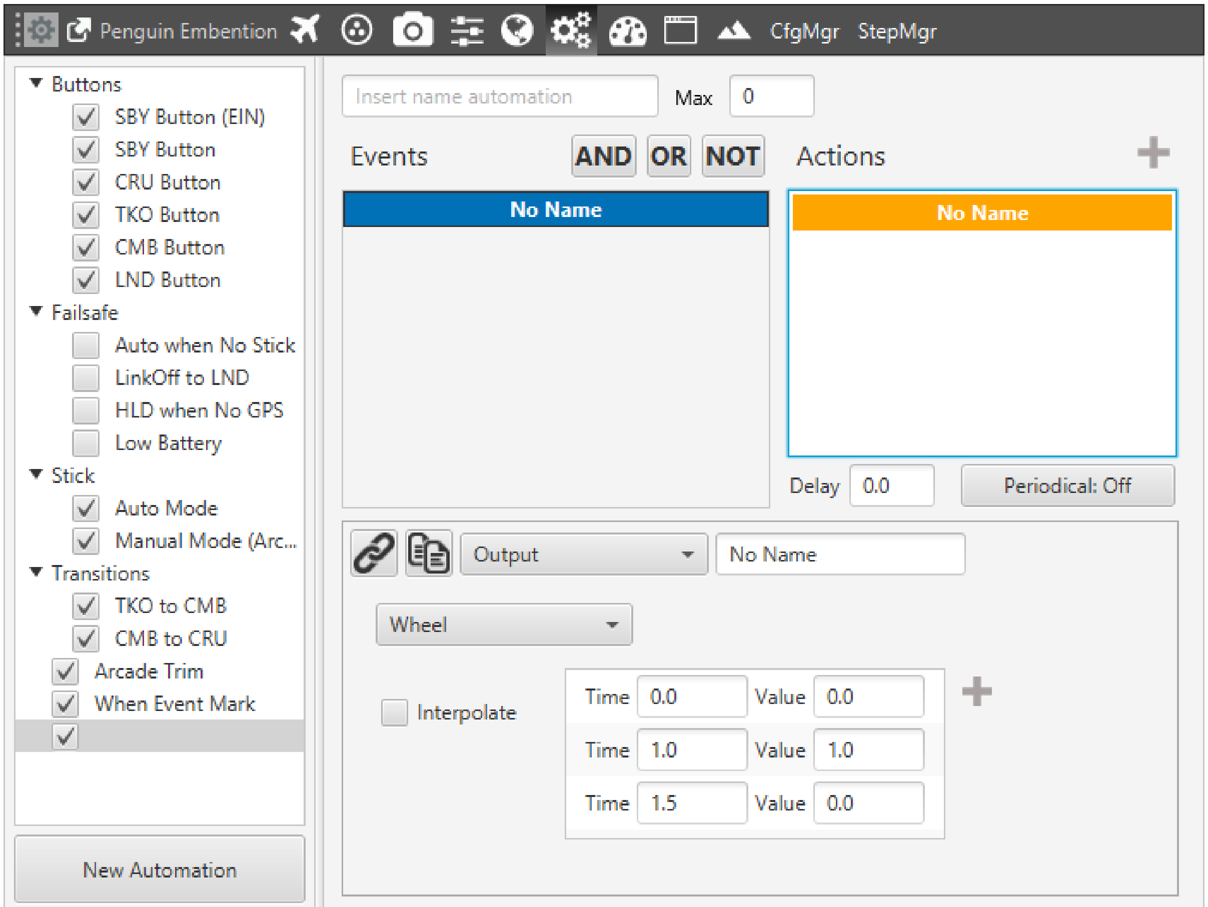


Action – Mode

These modes have been created previously. See *Modes section* for more information about creating modes.

6.2.6.2.10 Output

This action is used to set an actuator at a certain position.

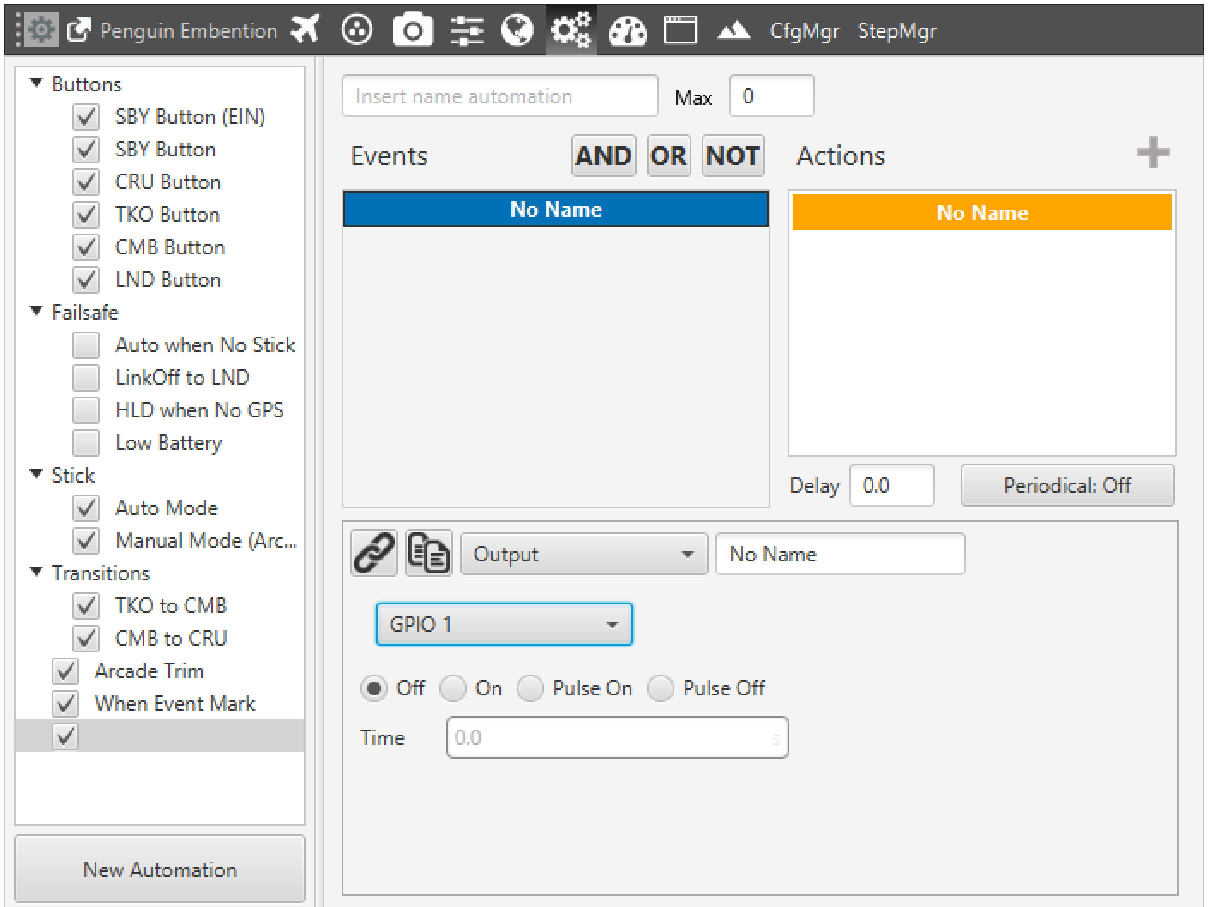


Action – Mode

The **desired output** is selected in the menu, and then a set of points with the time and value of the actuator can be indicated with the button "**+**".

Taking the previous figure as an example, when the action is triggered a series of commands are sent to the output **Wheel**. In this case, this pin is connected to a camera, which takes a photo when the trigger is active during 0.5 seconds.

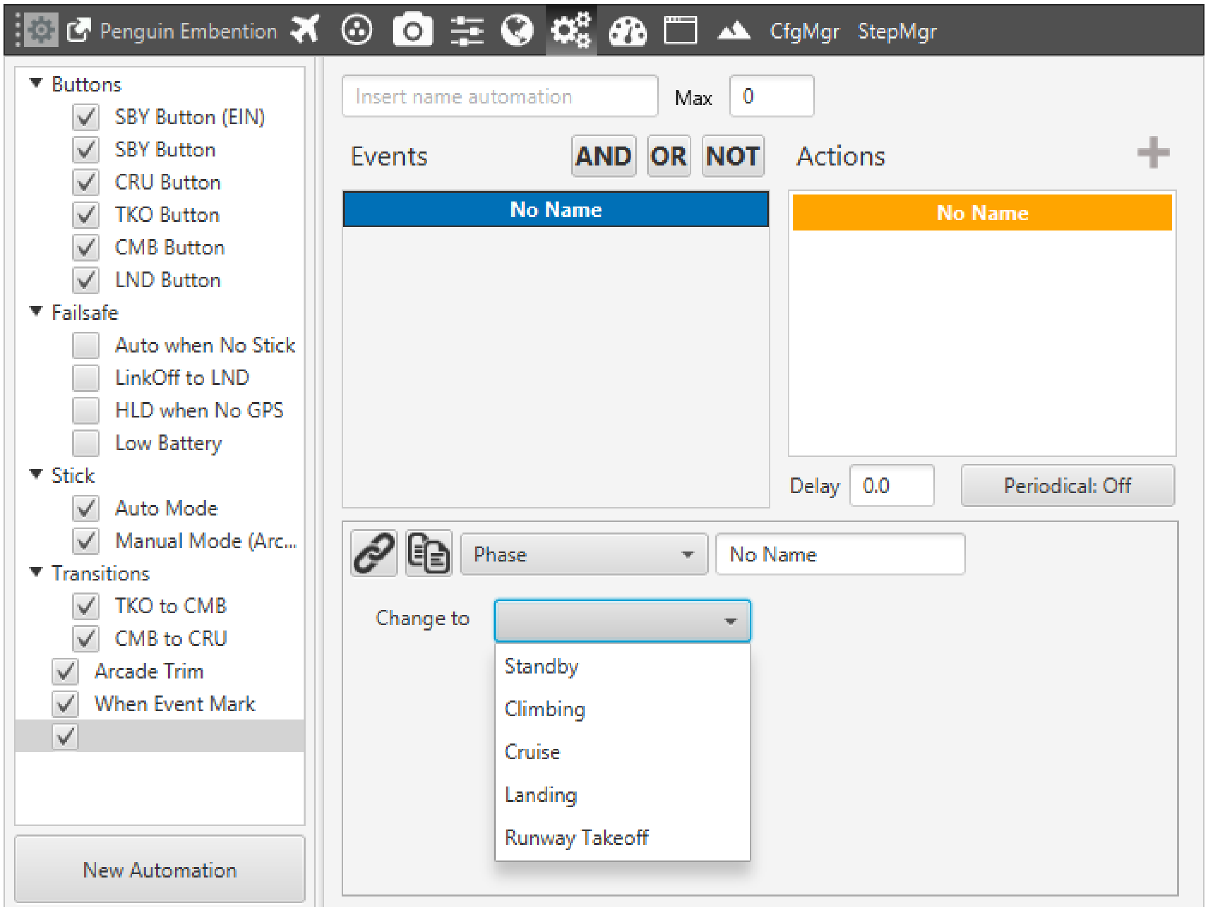
If the output pin has been configured as GPIO (visit section *GPIO*), the possible output signals to be commanded are a **continuous on** or **off pulse**, and a **limited** on and **off pulse**, whose **time** is set by the user.



Action – Mode

6.2.6.2.11 Phases

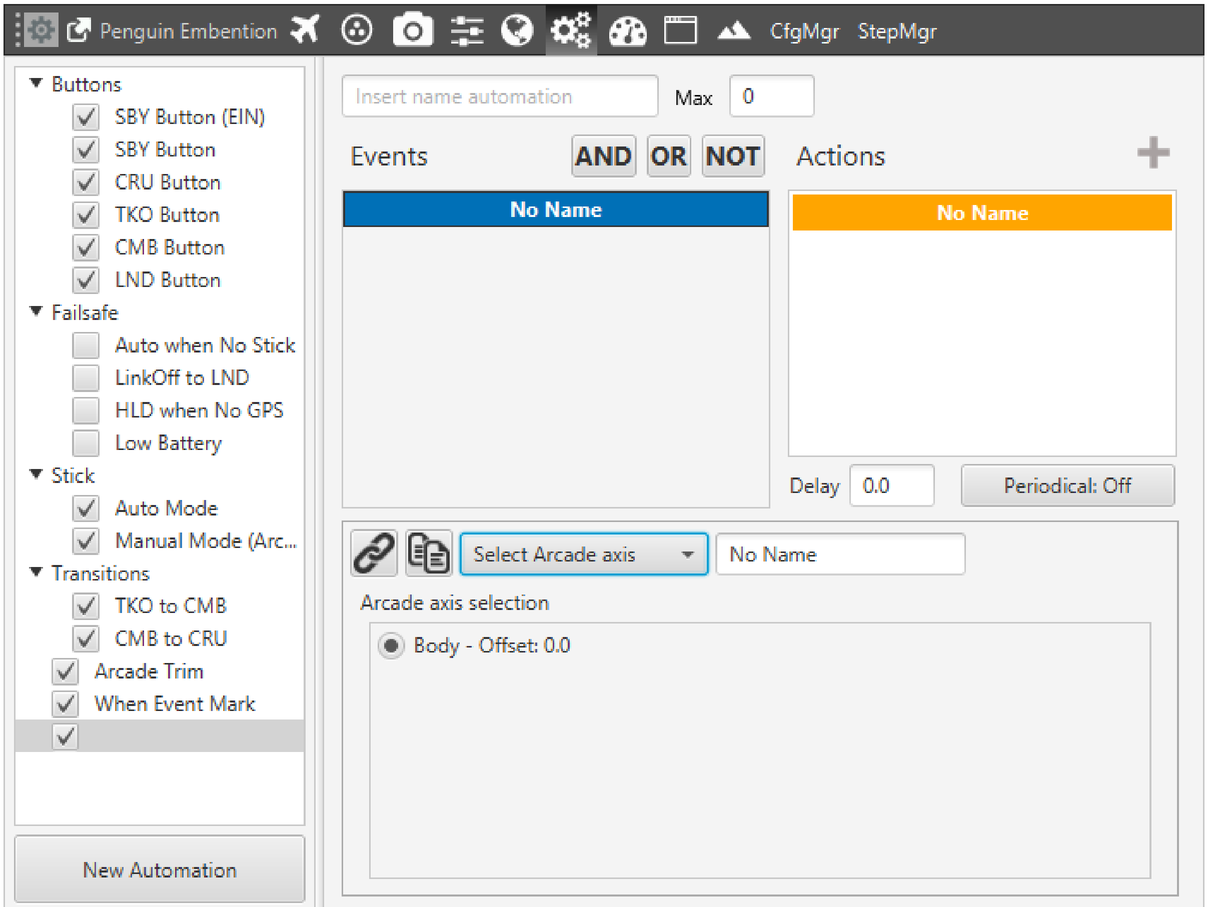
The flight phase is changed to the one selected in this action.



Action – Phases

6.2.6.2.12 Select Arcade Axis

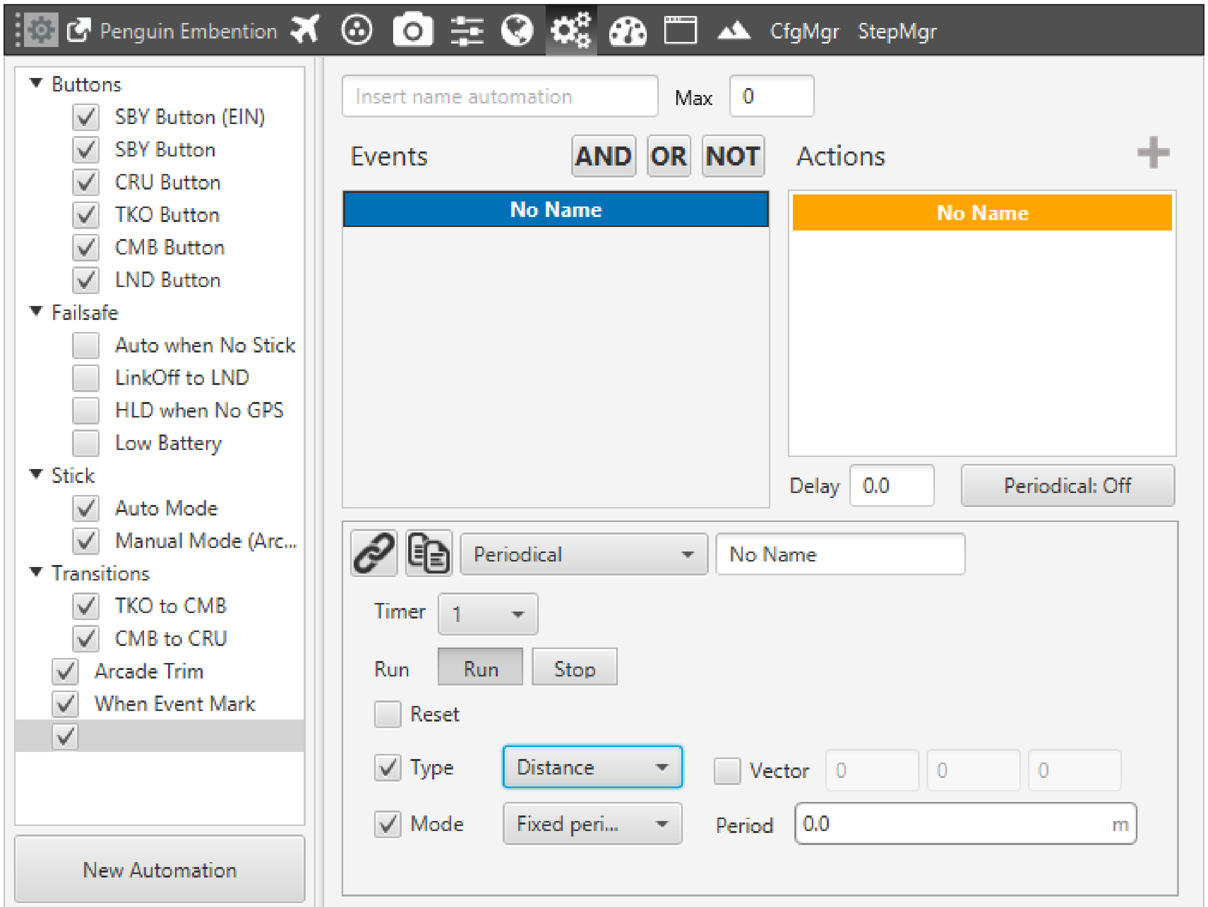
The axes system of the aircraft is changed to one that has been previously created in the Arcade Axis option inside the Control Panel, see section *Arcade Axis*



Action – Select Arcade Axis

6.2.6.2.13 Periodical

This action is used to set a timer during a flight operation.



Action – Periodical

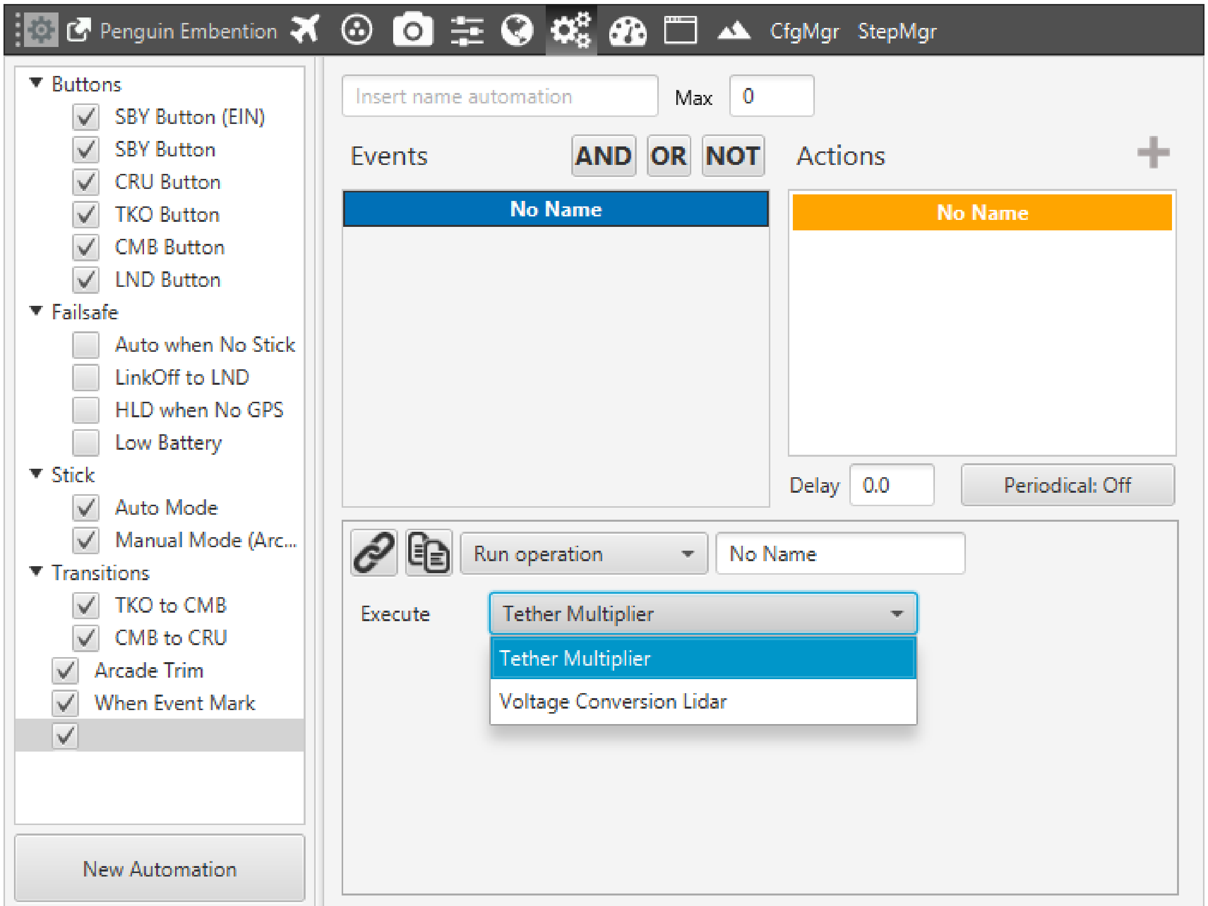
The first parameter is an identifier for the timer, so it can be used in an event for another automation. In order to explain the other parameters of the timer action, a set of examples will be detailed, each of them with different options.

- **Run + Distance/Time + Continuous:** when the action is triggered, the timer will be started and will measure distance/time from that instant until the moment when the autopilot is turned off (or until another automation acts on the same timer).
- **Run + Distance/Time + Fixed Delay/Period:** once the action has been triggered, the timer will start to measure a distance/time. Each time the value indicated in Period is reached, the event linked to this timer will be triggered. For example, if the user wants to take a photo each 25 meters, the timer should have Distance in the Type option and 25 meters in Period, then in another automation, an event of type Timer is created, so each time the timer reaches 25 meters the event will be triggered and the action will be carried out
- **Distance + Vector:** the distance is measured in the direction indicated by the vector.
- **Stop:** the timer will be stopped. Another automation should be created to run it again.
- **Reset:** when this action is active, the timer is reset to zero before starting to measure. If the reset is used with Stop, the timer will be stopped and set back to zero.

The difference between fixed delay and fixed period has been explained in *Automations*.

6.2.6.2.14 Run Operation

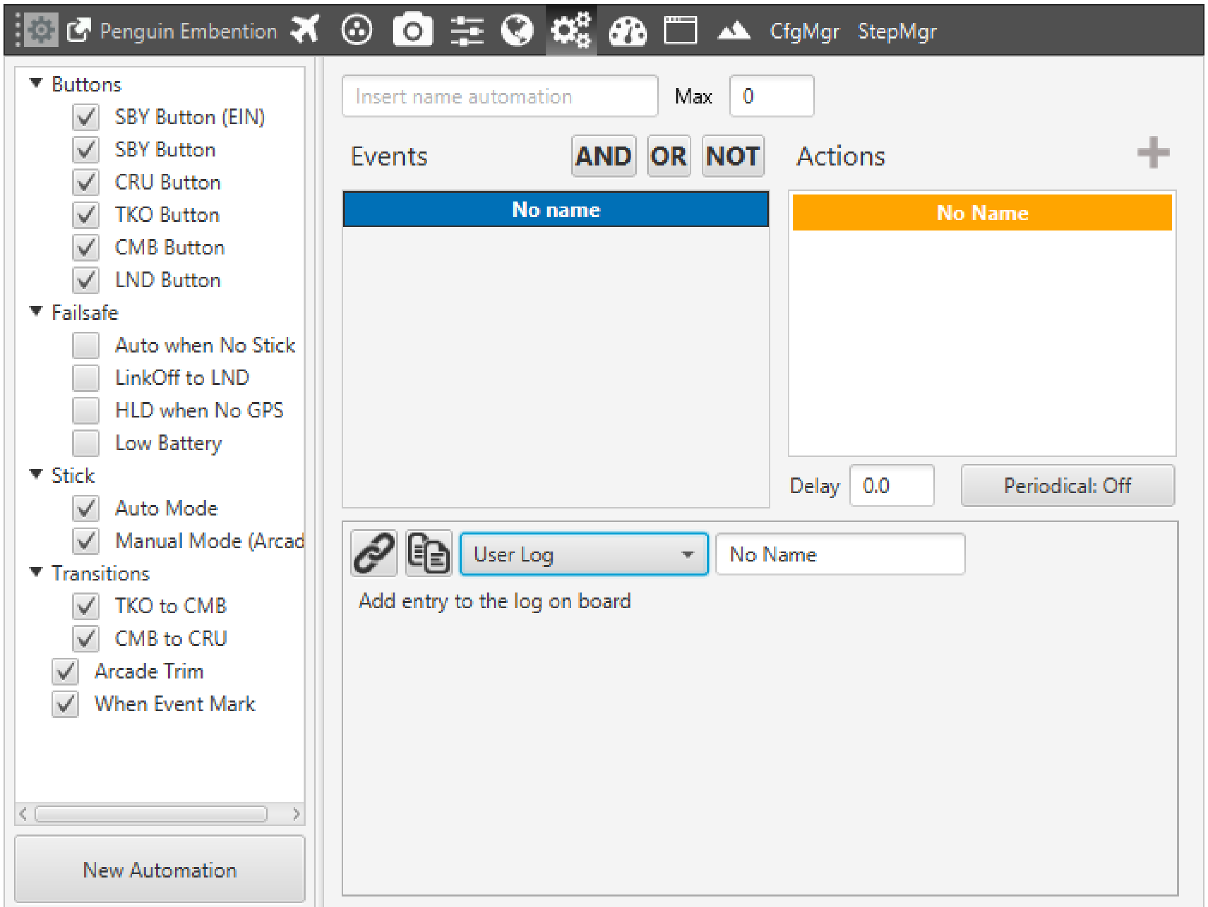
An operation previously defined in **Variables/System Variables** is carried out, see section *System Variables*.



Action – Run Operation

6.2.6.2.15 User Log

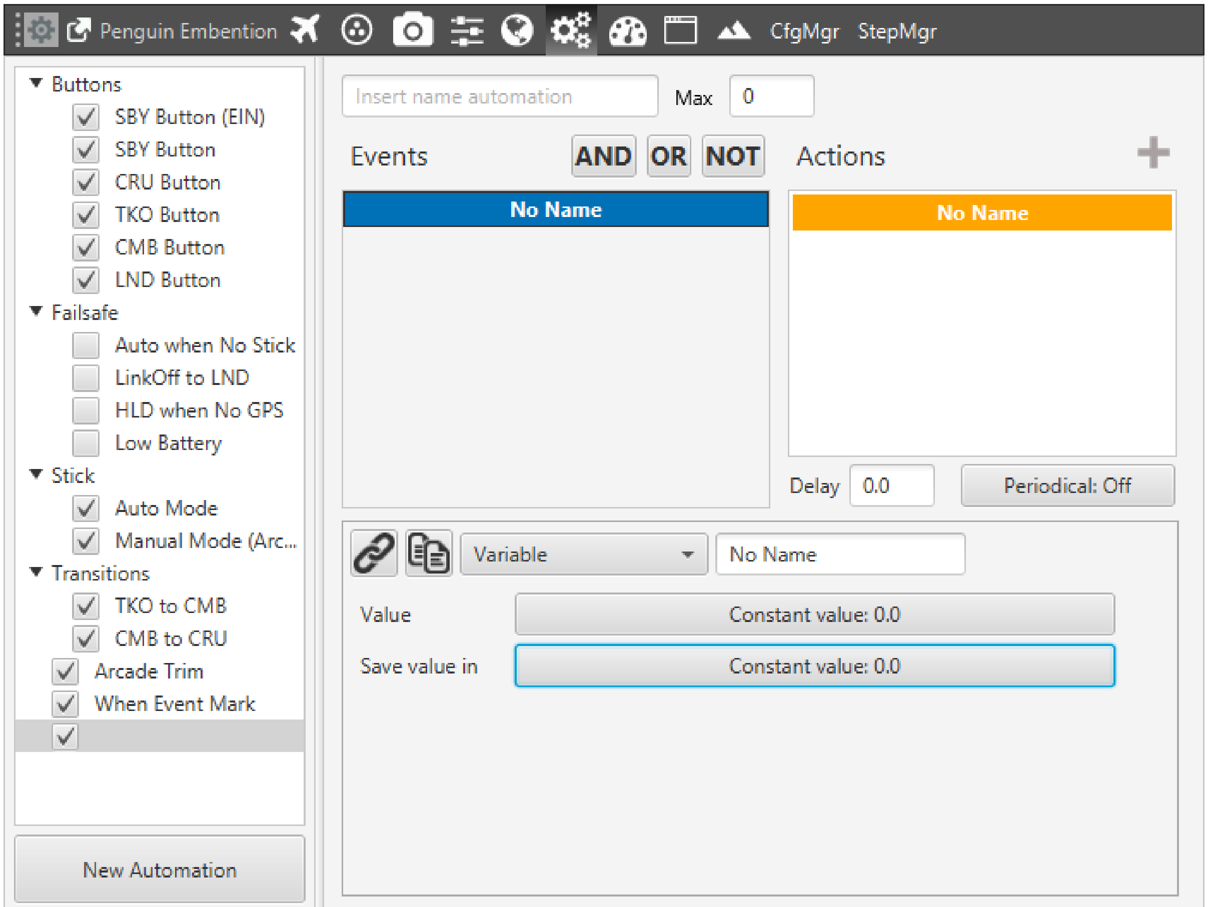
An entry is added to the log on board.



Action – User Log

6.2.6.2.16 Variable

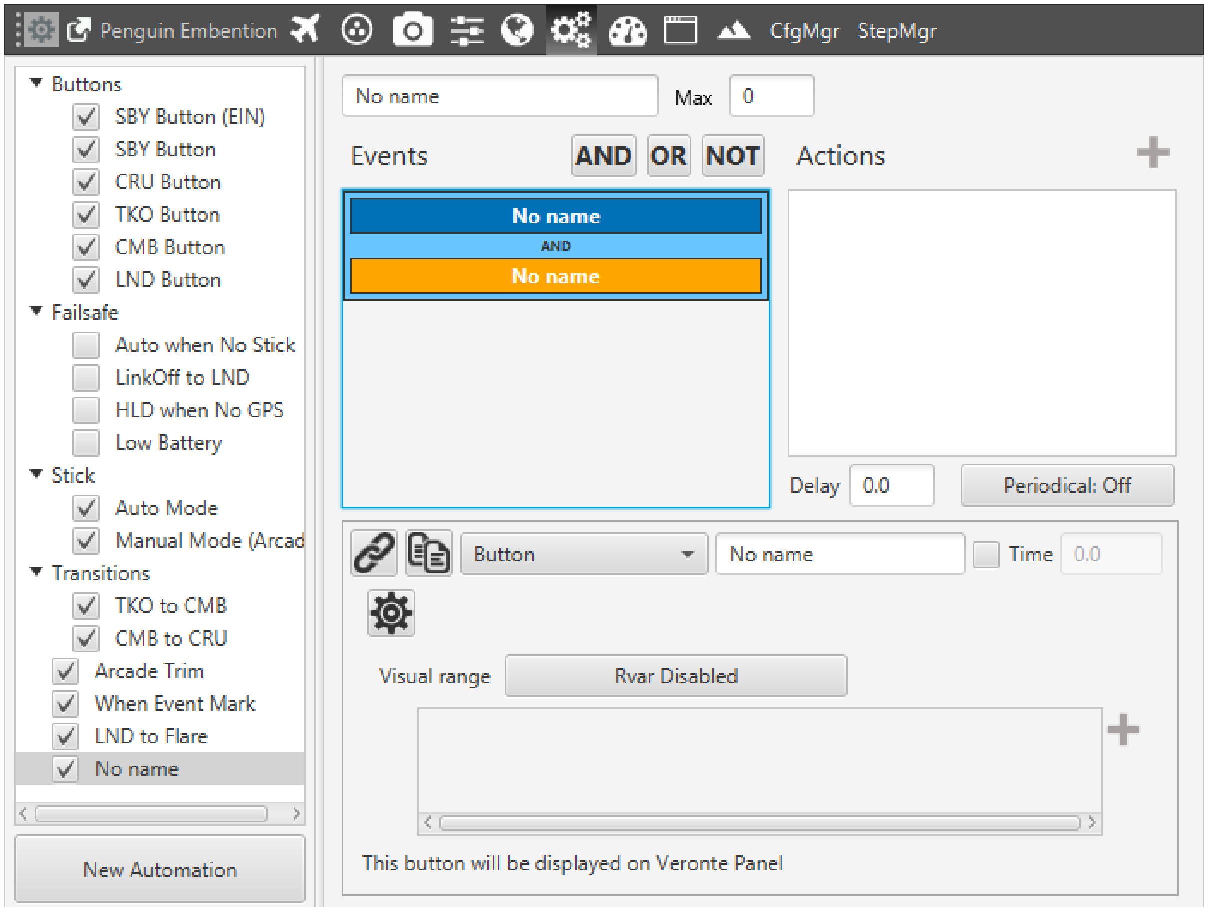
Allows user to select a variable and save it in a user variable.



Action – Variable

The **Actions** box contains all the actions that will be performed when the event (or group of events) has been accomplished.

To create a new action press  . When entering a new Event or Action it is possible to choose from one of the previously created on the system or to create a new one.

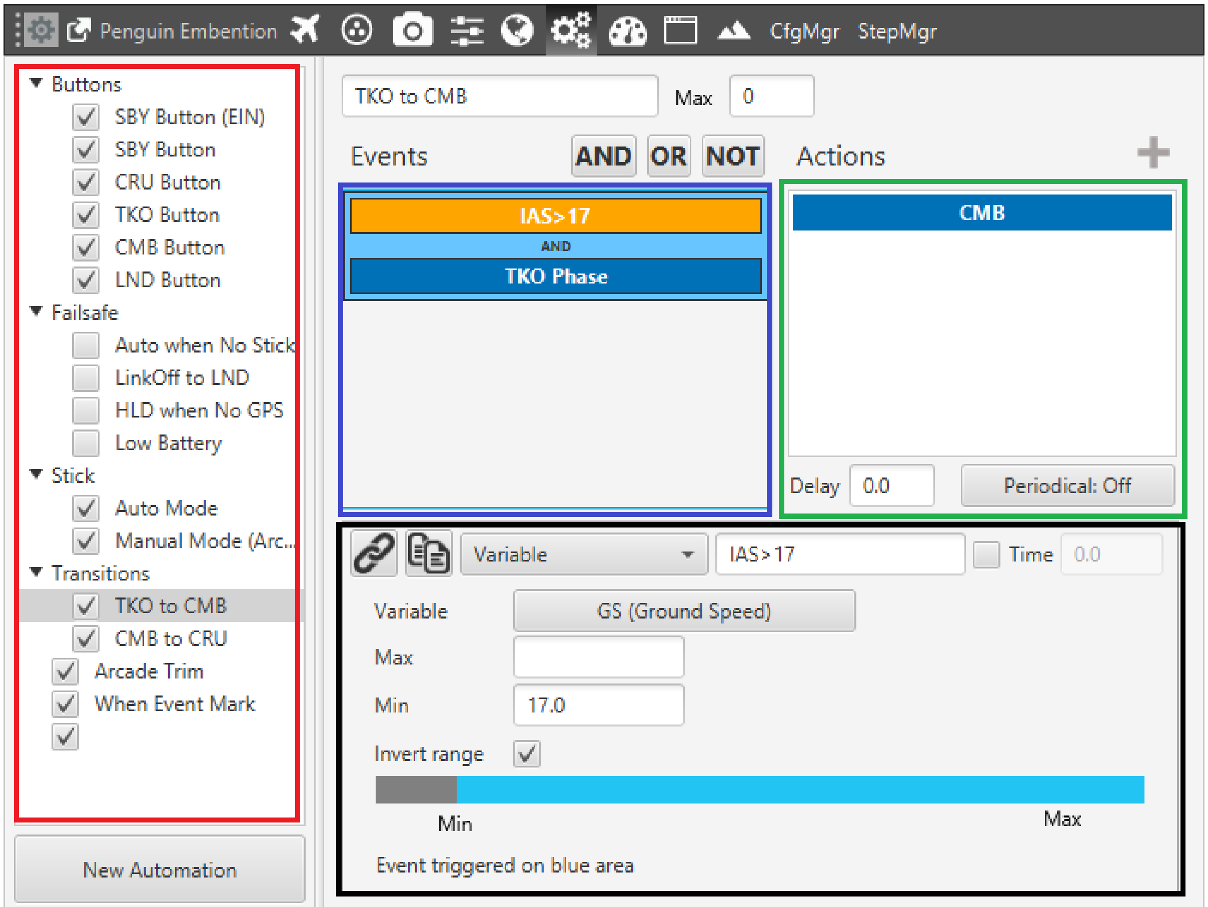


Automations Panel – Actions

When creating a new action, it possible to select different types of actions, these are explained in the next sections.

Automations are actions that are carried out when a combination of events happen, i.e when the events are accomplished the action are done. An example of what an automation could be a change of phase when reaching a certain altitude and speed, moving a servo when a button is clicked and many other possible combinations. In this section all the possible events and action will be explained in detail, so the user can combine them to create the automations that best suit their needs.

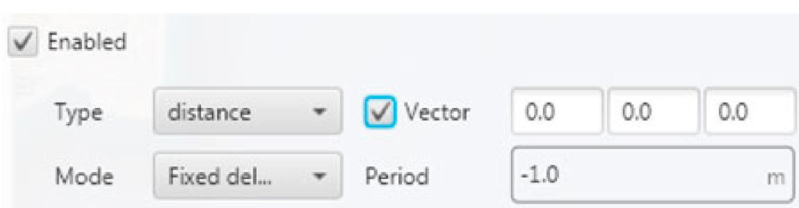
The following figure shows the layout of the automations menu, with a column for the events and another for the actions linked to these events.



Automation Display

Automations (red) are a combination of events (blue) and actions (green). All actions will be performed on event or an event combination triggering. There a are some parameters that can be configured in the events and actions menu and which are applicable independently of the type of event/action configured (black).

- **Delay:** time between the triggering of the event and the beginning of the action.
- **Time:** is a value related to the automations. Indicates how much time the event has to be accomplished in order to trigger the action. For example, if an event is to be above 100 meters and the value of Time is 3 seconds, the platform has to be above 100 meters during at least 3 second to trigger the action.
- **Periodical:** this menu is used to configure actions to take place periodically during the time that the events are active.ç



Automation Selection

The action can be configured to take place each certain distance or time. When using distance, the option Vector makes it possible to measure that distance along a direction specified by that vector. The two modes available for both time and distance are fixed delay and period. In order to explain the difference between them, the following figure is presented as an aid to the user.

Automation Selection

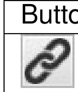
Let's consider that the system evaluates the automations each second (black line), and the automation that contains the periodical option is wanted to execute each 1.5 seconds (red line). In that case, the first action will be triggered at the second 1.5 but will be evaluated at second 2. The second time that the action is evaluated will depend on the mode: if fixed delay is selected, the evaluation of the action will be done 1.5 seconds after it was evaluated the first time, so that will be at second 3.5. On the other hand, if the mode is fixed period, the action will be evaluated 1.5 seconds after the first triggering (not evaluation) so that would be at the second 3. In the real praxis, the evaluation time for the automation is much lower than 1 second so the difference between the modes is much smaller.

6.2.6.3 Other Options

To create a new automation press **New Automation**, a new window will be displayed, users can select a previous one (if exists) or **Create new**.

Right-clicking on an automation makes it possible to **Remove** it or to **Clone** it. When a clone of an automation is created, the changes made in the event panel will be applied to the other one and vice versa, while the actions can be different in each automation. Automations also can be grouped by right-clicking in an automation and selecting the option **Change Group**.

Common configuration options are:

Button	Description
	Select an action or event from the available in the system. When modifying an action or event it will be modified in all automations where it is in use.
	Clone an existing action or event creating a new one with same parameters configured on the start point.

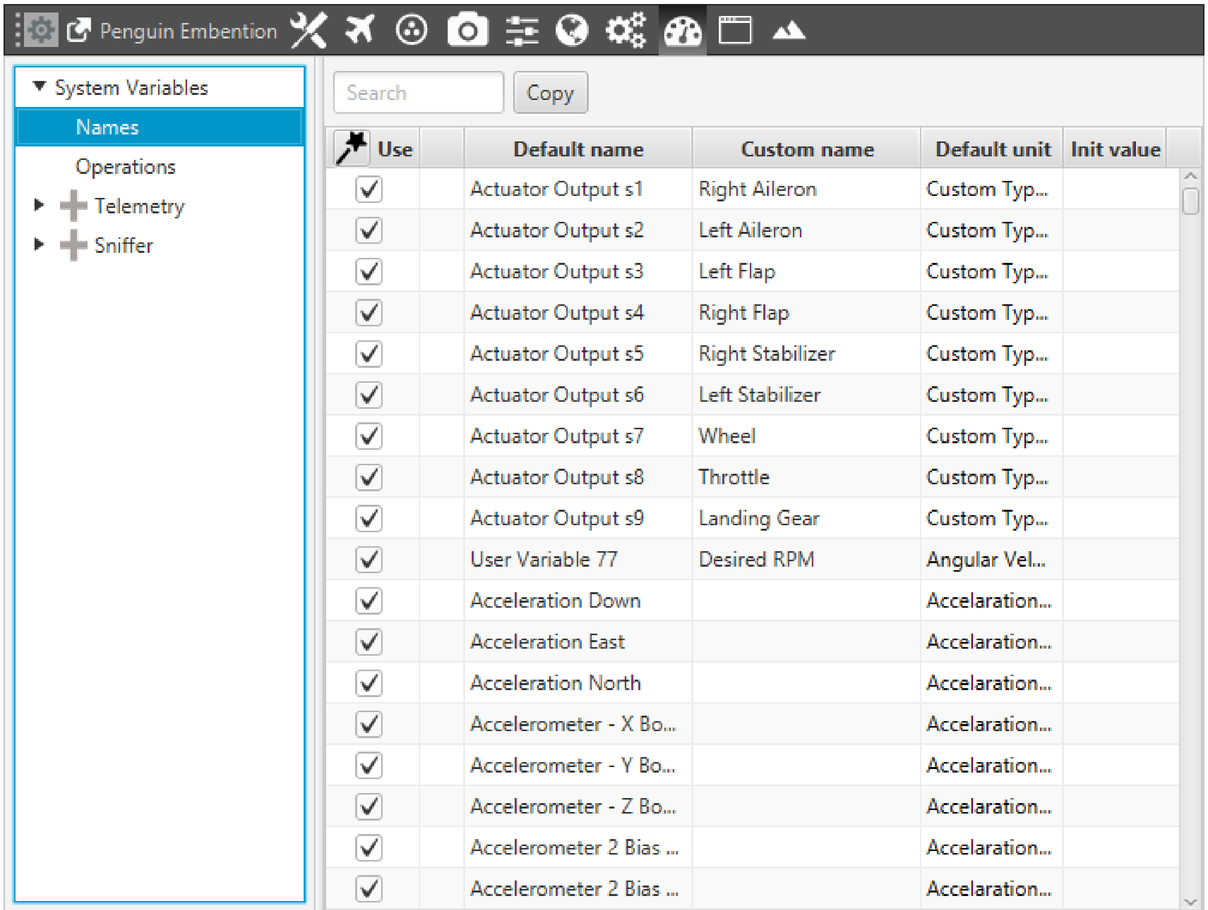
6.2.7 Variables

6.2.7.1 System Variables

6.2.7.1.1 Names

This menu is used to set a custom name for one of the system variables.

1. Click on the **Custom Name** cell of the desired variable and **introduce the new name** for it.
2. When the name is introduced press **Enter** to store the name on the system.
3. Press **Save** to save all changes.



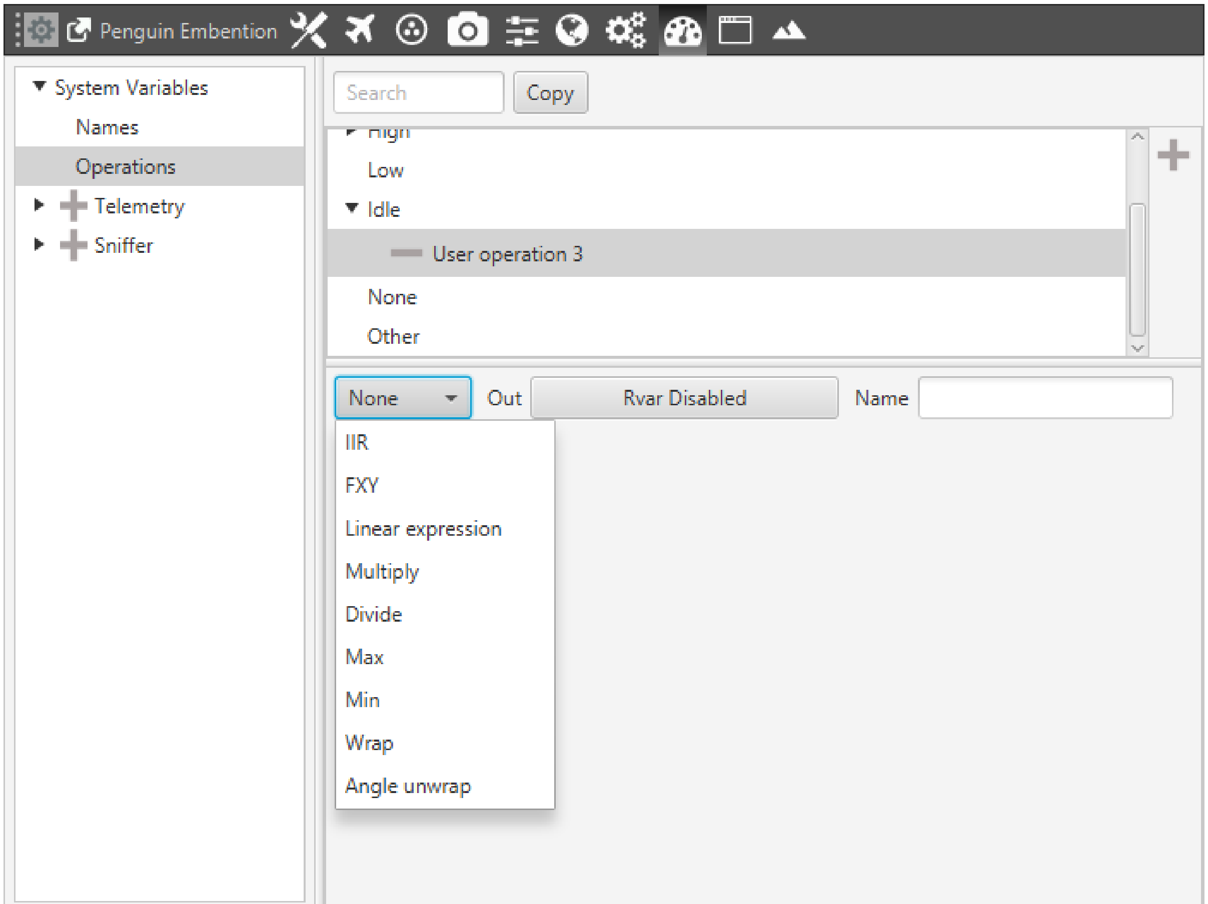
Variable Name Customization

Name	Type	Size
U	Unsigned int	16
R	Float	32
B	Bit	1

Operations

It is possible to configure custom operations to be performed in Veronte by selecting the input and output variables and operation parameters. For instance, in this menu it is possible to configure a Lidar sensor, converting the variable measured (voltage) into a physical variable.

The menu shown in the following figure allows the user to create operations with the system variables.

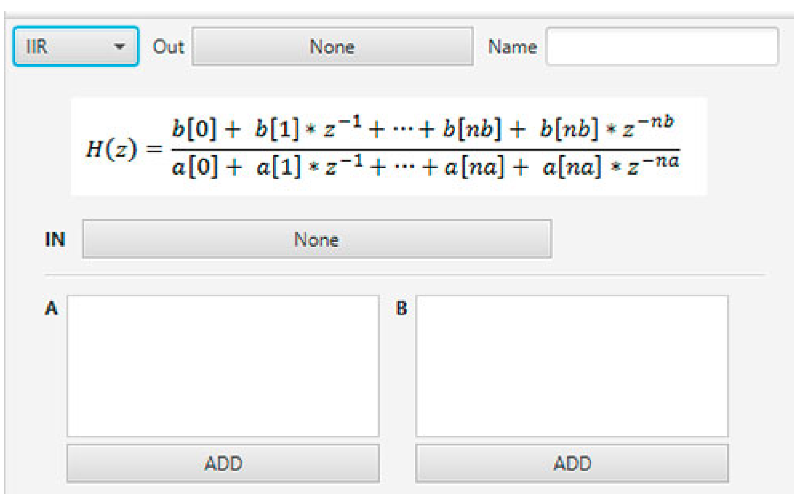


Operations Menu

Super high, High, Low and Idle are the different processing velocities for the tasks performen in Veronte Autopilot (the workin frequencies for each one of them are indicated in **Devices/Veronte/Frequencies**). By default, the new operation appears in **Idle**, but it can be moved to another option just dragging it to the desired place.

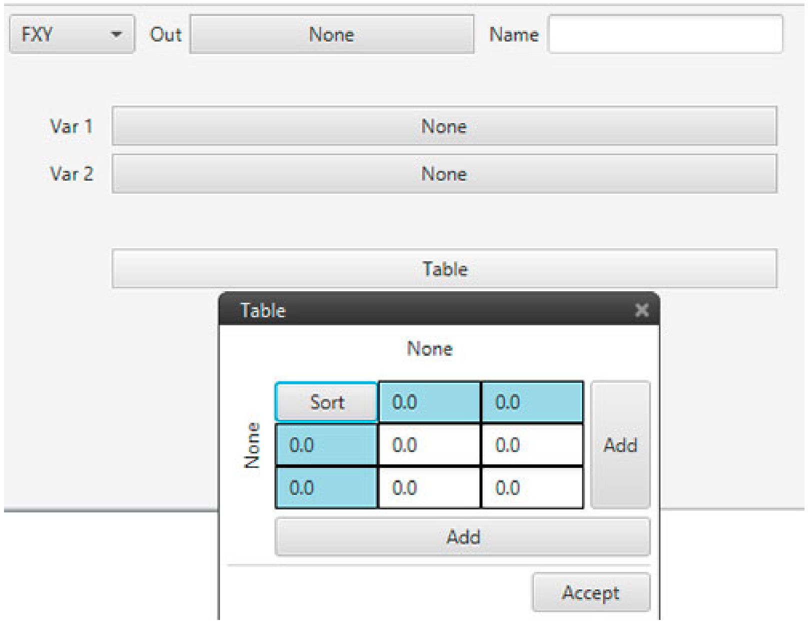
When creating a new operation, the following types are available in Pipe.

- **IIR:** IIR digital filter, enter the parameters for filtering the variable value.



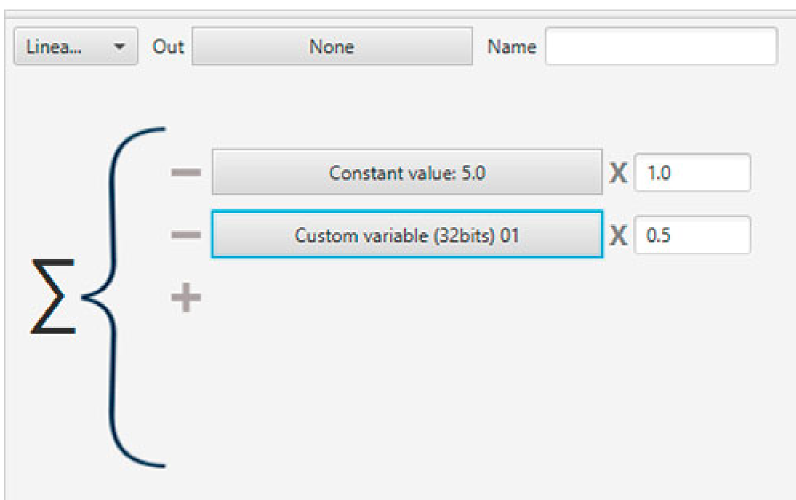
IIR

- **FXY:** FXY matrix, complete the table for setting an output value according to two input ones.



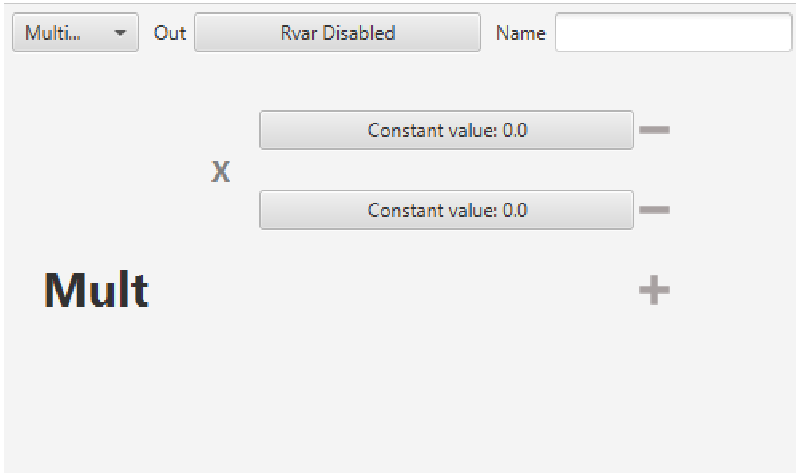
FXY

- **Linear Expression:** Output variable acquires the value of the sum of input variables multiplied for a constant value.



Linear

- **Multiply/Divide:** Output variable acquires the vulue of the multiplication/divition of input variables.

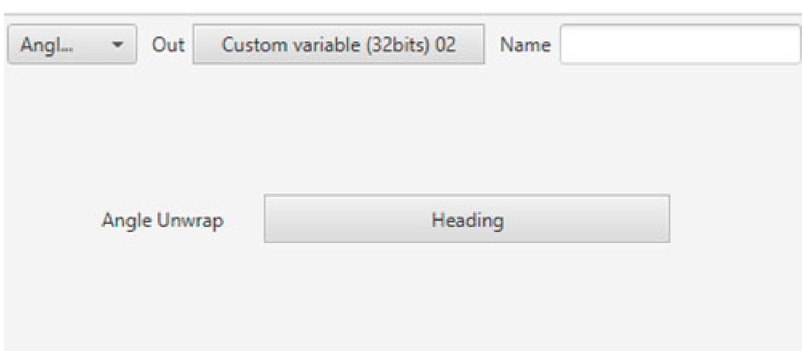


Multiply/Divide

- **Wrap:** Output variable is wrapped to keep value between upper and down limits. For example, if the desired IAS is wrapped between 0 and 12 m/s, the output variable will have a value of 12 when the desired IAS is greater than this value, and a value of 0 when it is negative.

Wrap

- **Angle Unwrap:** When an angle is wrapped in the PIDs, its value is delimited between pi and -pi radians. The operation presented here undoes that calculation by representing the angle in a range starting at 0 radians.



unwrap

6.2.7.2 Telemetry

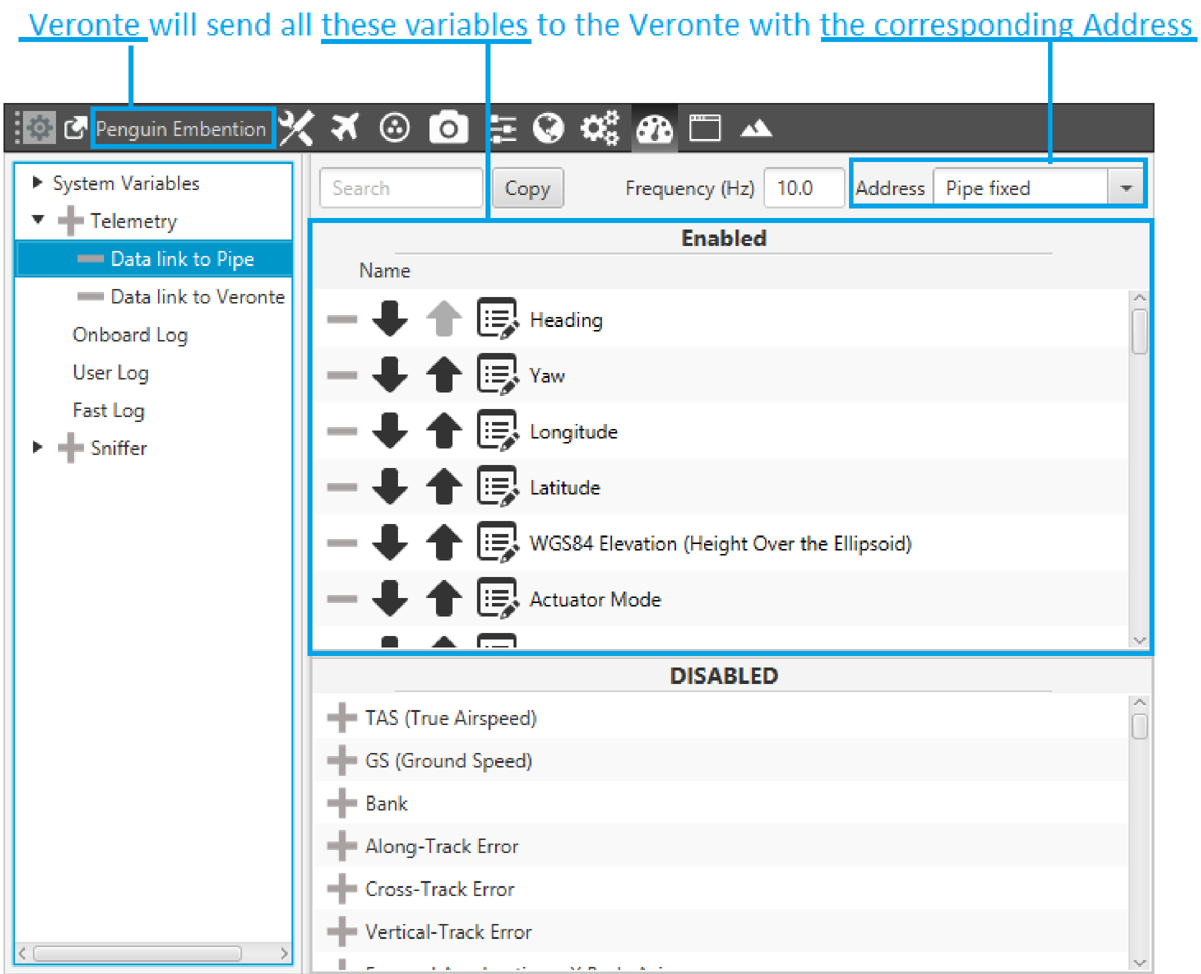
Telemetry controls permit to configure data to be stored or transmitted on the system. There are 4 main items that can be configured within this panel:

Type	Description
Data Link	Configures the variables to send throughout the data link channel.
Onboard Log	Sets the variables to be stored on system Log. (on Veronte SD Card)
User Log	User Log for custom applications.
Fast Log	Saves data at the maximum frequency available on the system. Recording time depends on the selected variables.

Configuration display permits to enable the desired variables for each telemetry file and to set the maximum and minimum values together with precision for each one.

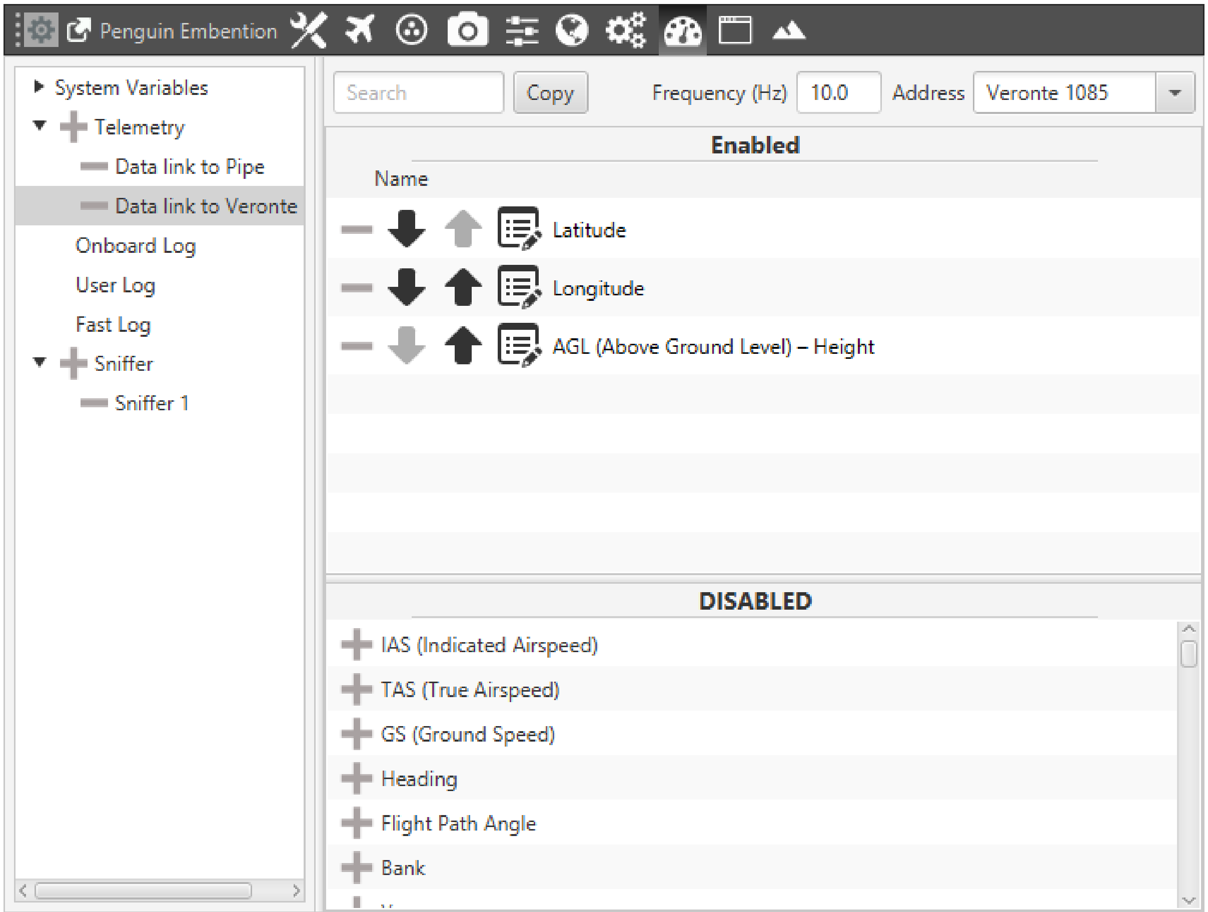
6.2.7.2.1 Data Link

The Data Link contains the variables sent between Veronte Units and Veronte Pipe. By default, the system provides one Data Link that represents the connection between the air autopilot and the software (Pipe). Veronte Air sends the variables to Veronte Ground, being processed when they arrive there by Veronte Pipe.



Telemetry Configuration Menu

Pipe permits the creation of more Data Links too. The user has to configure which unit sends the information and which receives it and the sending rate. As an example, another possible data link could be set between the Air and Ground autopilots directly (without Pipe) and used to send the position of the UAV to the Ground autopilot for the configuration of a tracker. This Data Link example is presented in the following figure.



Data Link (Ground/Air)

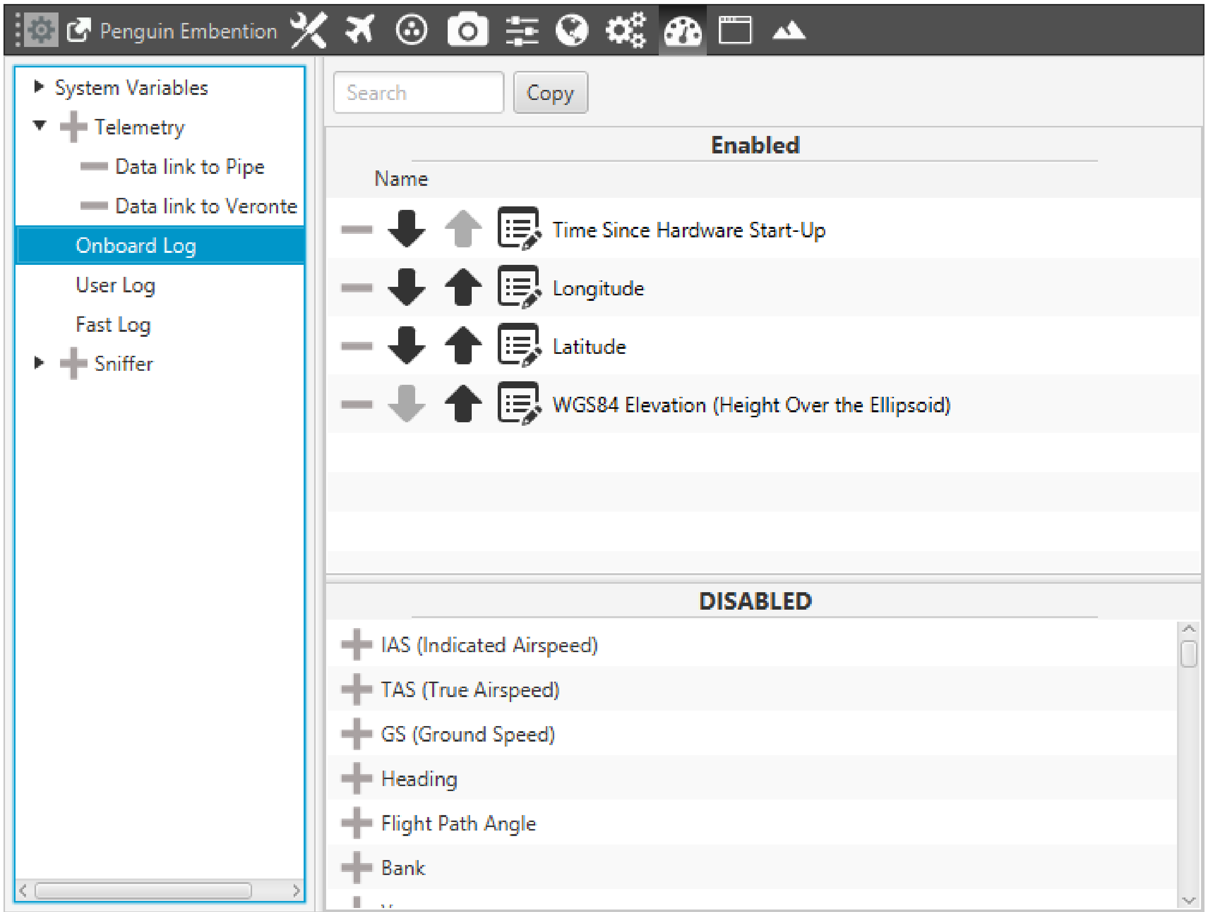
The autopilot **Penguin Embention** will send the Latitude, Longitude and AGL to the autopilot with address **Veronte 1085**. The sending rate is set to 10 Hz.

The unit that receives the telemetry has to configure its sniffer in order to store the data.

> **Warning:** If the number of variables enabled for telemetry communication are higher than the maximum supported by the system, the latest variables will not be sent, so they will display a zero value if shown in the workspace.

6.2.7.2.2 Onboard Log

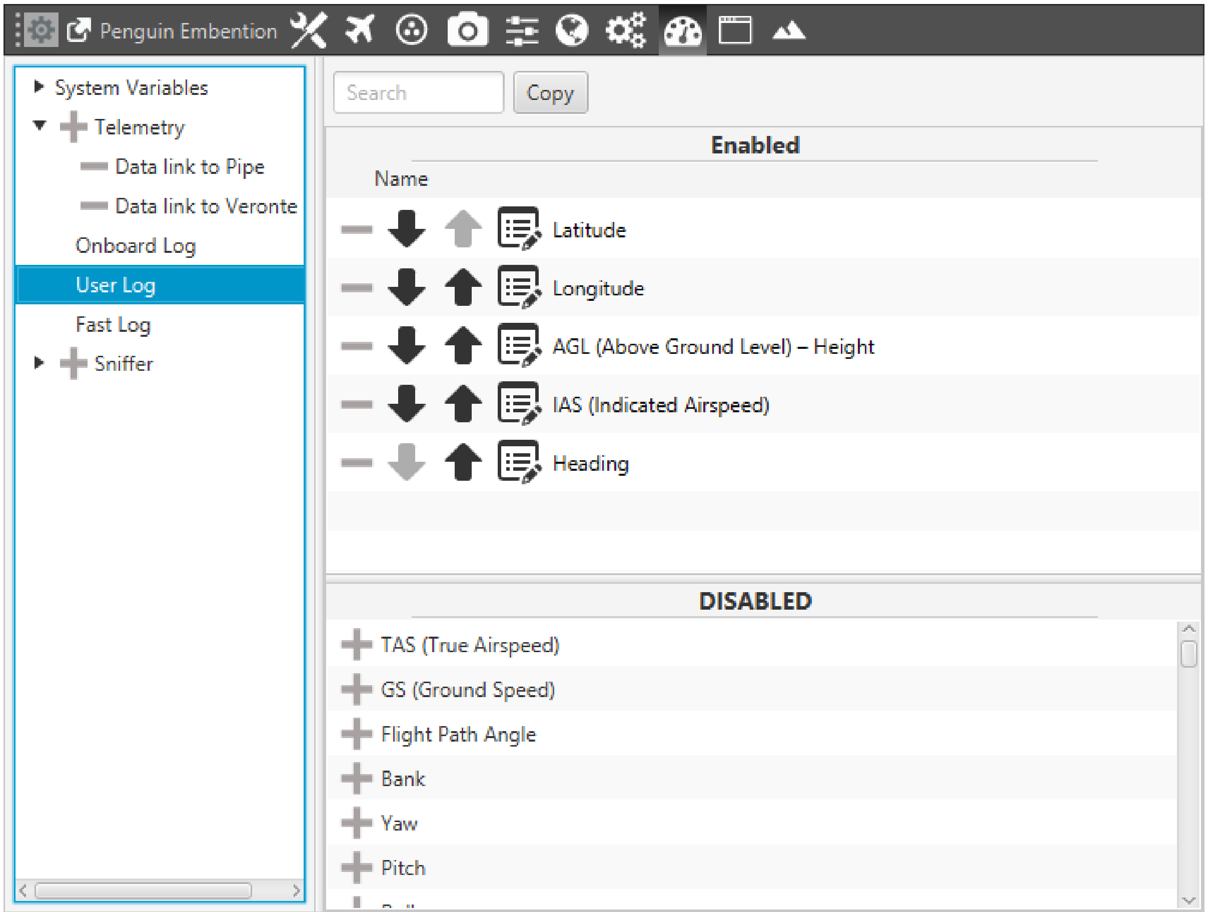
The Onboard Log determines the variables that are being stored on the autopilot SD Card. In this case, there are not sending/receiving units so the only thing to configure here is the list of variables that will be saved on the autopilot internal memory for a further download and processing. The SD management task is **Low**, which is an execution level running at 30 Hz.



Onboard Log

6.2.7.2.3 User Log

The user log contains the variables that are stored according to an automation created by the user. Considering an example, in a photogrammetry mission it is important to record the aircraft location when the photo is taken, so a user log could be used to record a certain set of variables (position, speed, direction…) each time a photo is taken.



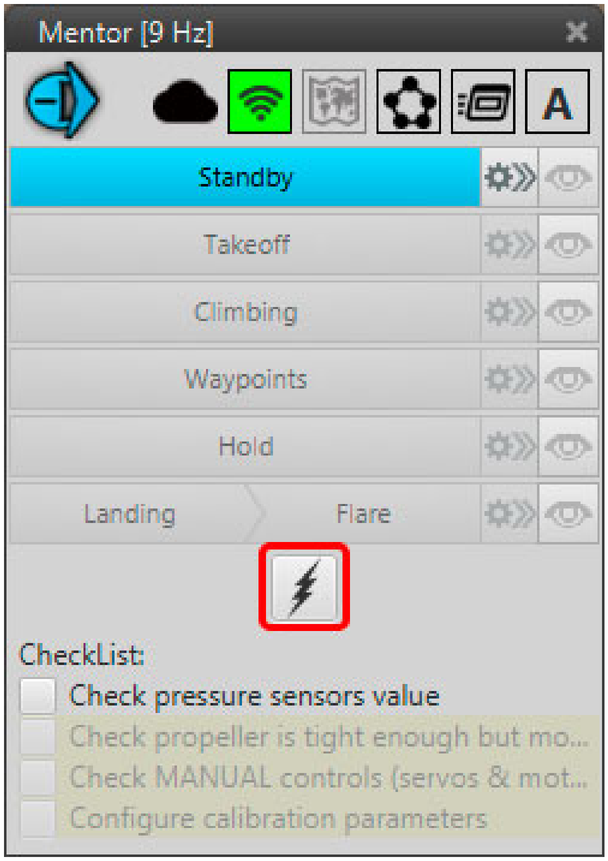
User Log

In order to create a User Log action where an entry is added to the log when a certain set of events are accomplished check ref section 6.9 of the manual (Actions).

6.2.7.2.4 Fast Log

The fast log store the specified variables at the maximum rate available on the system. This tool could be used to save information in an operation that happens extremely fast such as missile launching. The time that this logging process lasts depends on the number of variables being saved.

When a variable is enabled in the Fast Log, a new button will appear in the Veronte Panel. The user must click on it when he wants to activate the fast log tool.

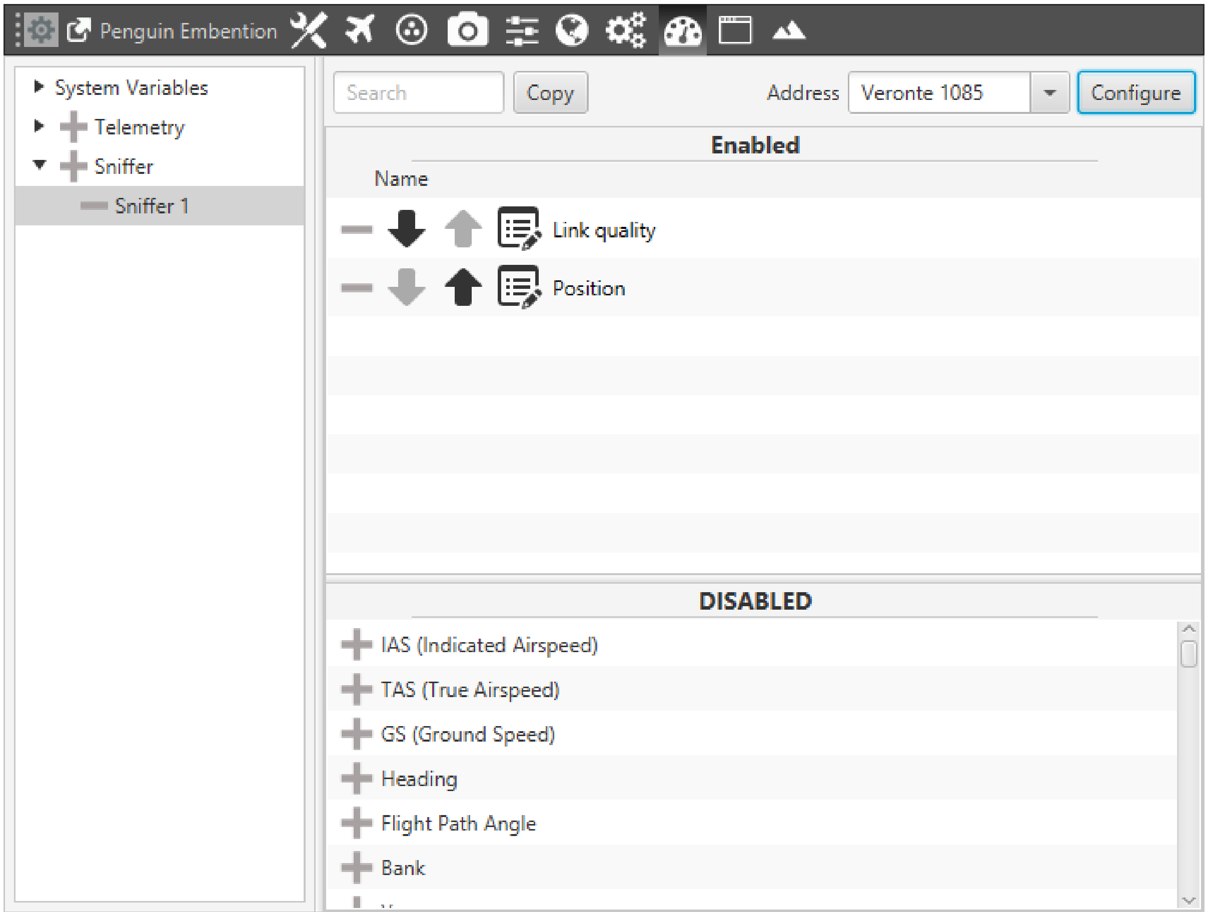


Fast Log button

The downloading of the information of an operation depends on how it has been stored, i.e depends on the type of log (data link, onboard, user or fast). Visit section 10 of the manual to see the information related to the postflight tools contained in Veronte Pipe.

6.2.7.3 Sniffer

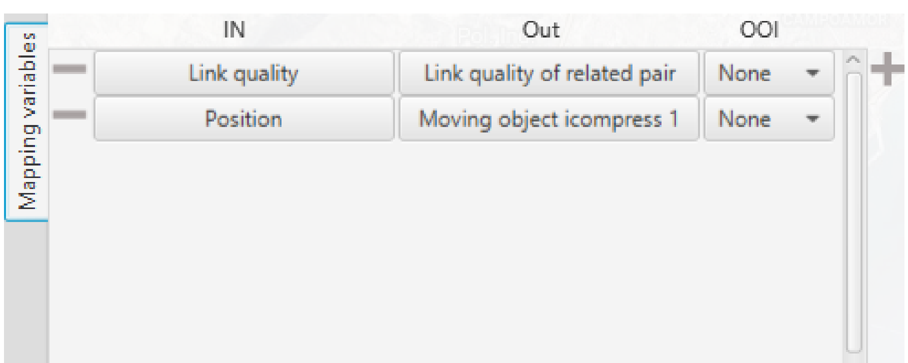
This menu is used to establish a telemetry communication between two autopilots. The autopilot being configured will "listen" the variables indicated in the window **Enabled**, from another autopilot whose address is indicated in **Address**. The sniffer is commonly used to make the aircraft listen the position of the ground station and the link quality.



Sniffer Configuration Menu

The source UAV, in this case, is the ground station (1085), which communicates to the Air UAV its position and the quality of the radio link between them. In **Mapping Variables** (Select **Configure**), the ones send by the ground UAV are indicated in the column **IN**, and they are stored in the variables indicated in **Out** for its later use by the air autopilot.

The sniffer is configured so the air autopilot has information about the state of the communications so it can perform an action when the link is lost, and also to make the aerial platform know about the ground station position so it can perform a mission in relation to that point.



Mapping Variables

The Veronte Unit that sends the data has to be configured as well, in the Telemetry Panel. That unit will send telemetry through a Data Link.

6.2.7.4 List of Variables

> **Warning:** Bit Variables displayed on Labels (see ref Workspace – Gauge Display) will be shown as Red/Green depending on its state. Red stands for 0 and Green for 1, changing the name displayed accordingly to the BIT value.

6.2.7.4.1 32 VAR

ID	Name	Units/Values	Description
0	IAS (Indicated Air Speed)	m/s	Pitot-static measurement speed
1	TAS (True Air Speed)	m/s	Speed relative to the airmass in which the vehicle is moving (I
2	GS (Ground Speed)	m/s	Horizontal speed, relative to the ground
3	Heading	radian	Direction in which the vehicle velocity vector is pointing
4	Flight Path Angle	radian	Angle between velocity vector and local horizontal line
5	Bank	radian	Velocity vector lateral component
6	Yaw	radian	Angle around the Vertical axis of the vehicle
7	Pitch	radian	Angle around the Transverse axis of the vehicle
8	Roll	radian	Angle around the Longitudinal axis of the vehicle
9	Along-Track Error	m	Fix error along the flight track
10	Cross-Track Error	m	Fix error across the flight track
11	Vertical-Track Error	m	Fix error in the vertical flight track
12	Roll Rate	rad/s (RDS)	Rate of change of the Roll angle
13	Pitch Rate	rad/s (RDS)	Rate of change of the Pitch angle
14	Yaw Rate	rad/s (RDS)	Rate of change of the Yaw angle
15	Forward Acceleration – X Body Axis	m/s^2	Acceleration in the X-axis
16	Right Acceleration – Y Body Axis	m/s^2	Acceleration in the Y-axis
17	Bottom Acceleration – Z Body Axis	m/s^2	Acceleration in the Z-axis
18	RPM	rad/s (RDS)	Revolutions per minute configurable for external sensor
19	Front GV (Ground Velocity)	m/s	GV vector X component
20	Lateral GV (Ground Velocity)	m/s	GV vector Y component
21	Velocity	m/s	Velocity vector module
22	Forward Load Factor – X Body Axis	customType	G-force in X body axis
23	Right Load Factor – Y Body Axis	customType	G-force in Y body axis
24	Bottom Load Factor – Z Body Axis	customType	G-force in Z body axis
25	Tangential Acceleration	m/s^2	Absolute acceleration for tangential direction
26	Energy Rate	customType	Variation from system total energy (kinetic & potential)
27	Energy Distribution	customType	Division of energy used between kinetic and potential
28	co-yaw	radian	Acrobatic Yaw with Body Z' axis pointing to X
29	co-pitch	radian	Acrobatic Pitch with Body X' axis pointing to -Z
30	co-roll	radian	Acrobatic Roll with Y' keeping same as Y
31	Roll acceleration	$radian/s^2$	Acceleration around the longitudinal axis
32	Pitch acceleration	$radian/s^2$	Acceleration around the transverse axis
33	Yaw Acceleration	$radian/s^2$	Acceleration around the vertical axis
40	RSSI	decimal	Received Signal Strength Indicator
100	Desired IAS (Indicated Air Speed)	m/s	Commanded IAS from guidance
101	Desired TAS (True Air Speed)	m/s	Commanded TAS from guidance
102	Desired GS (Ground Speed)	m/s	Commanded GS from guidance
103	Desired Heading	radian	Commanded Heading from guidance
104	Desired Flight Path Angle	radian	Commanded Flight Path Angle from guidance

Table 7 – continued from previous page

ID	Name	Units/Values	Description
105	Desired Bank	radian	Commanded Bank from guidance
106	Desired Yaw	radian	Commanded Yaw from guidance
107	Desired Pitch	radian	Commanded Pitch from guidance
108	Desired Roll	radian	Commanded Roll from guidance
109	Desired Along-Track Error	m	Commanded Along-Track error from guidance
110	Desired Cross-Track Error	m	Commanded Cross-Track error from guidance
111	Desired Vertical-Track Error	m	Commanded Vertical-Track error from guidance
112	Desired Roll Rate	rad/s (RDS)	Commanded Roll rate from guidance
113	Desired Pitch Rate	rad/s (RDS)	Commanded Pitch rate from guidance
114	Desired Yaw Rate	rad/s (RDS)	Commanded Yaw rate from guidance
115	Desired Foward Acceleration – X Body Axis	m/s^2	Commanded Forward Acceleration from guidance
116	Desired Right Acceleration – Y Body Axis	m/s^2	Commanded Right Acceleration from guidance
117	Desired Bottom Acceleration – Z Body Axis	m/s^2	Commanded Bottom Acceleration from guidance
118	Desired RPM	rad/s (RDS)	Commanded RPM from guidance
119	Desired Front GV (Ground Velocity)	m/s	Commanded Front GV from guidance
120	Desired Lateral GV (Ground Velocity)	m/s	Commanded Lateral GV from guidance
121	Desired Velocity	m/s	Commanded Velocity from guidance
122	Desired Forward Load Factor – X Body Axis	customType	Commanded Forward Load Factor from guidance
123	Desired Right Load Factor – Y Body Axis	customType	Commanded Right Load Factor from guidance
124	Desired Bottom Load Factor – Z Body Axis	customType	Commanded Bottom Load Facto from guidance
125	Desired Tangential Acceleration	m/s^2	Commanded Tangential Acceleration from guidance
126	Desired Energy Rate	customType	Commanded Energy Rate from guidance
127	Desired Energy Distribution	customType	Commanded Energy Distribution from guidance
128	Desired co-yaw	radian	Commanded co-yaw from guidance
129	Desired co-pitch	radian	Commanded co-pitch from guidance
130	Desired co-roll	radian	Commanded co-roll from guidance
200	Desired North GV (Ground Velocity)	m/s	Commanded North (NED Coordinates system) GV
201	Desired East GV (Ground Velocity)	m/s	Commanded East (NED Coordinates system) GV
202	Desired Down GV (Ground Velocity)	m/s	Commanded Down (NED Coordinates system) GV
203	Desired 2D MSL (Heigh Above Mean Sea Level)	m	Commanded MSL from guidance in 2D height mo
204	Desired 2D AGL (Above Ground Level) – Height	m	Commanded AGL from guidance in 2D height mo
205	Desired 2D WGS84 Elevation (Height Over The Ellipsoid)	m	Commanded WGS84 Elevation from guidance in
206	Desired Longitude	radian	Commanded Longitude from guidance
207	Desired Latitude	radian	Commanded Latitude from guidance
208	Desired WGS84 Elevation (Height Over The Ellipsoid)	m	Commanded WGS84 Elevation from guidance
209	Desired MSL (Height Above Mean Sea Level) – Altitude	m	Commanded MSL Altitude from guidance
210	Desired AGL (Above Ground Level) – Height	m	Commanded AGL Altitude from guidance
250	Guidance north position error	m	Difference from Desired and actual north position
251	Guidance east position error	m	Difference from Desired and actual east position
252	Guidance down position error	m	Difference from Desired and actual down position
253	Guidance PID north desired velocity	m/s	Difference from Desired and actual PID north velo
254	Guidance PID east desired velocity	m/	Difference from Desired and actual PID east veloc
255	Guidance PID down desired velocity	m/s	Difference from Desired and actual PID down velo
256	Desired velocity X body axis	m/s	Commanded velocity in X-axis from guidance
257	Desired velocity Y body axis	m/s	Commanded velocity in Y-axis from guidance
258	Desired velocity Z body axis	m/s	Commanded velocity in Z-axis from guidance
259	External yaw	radian	Yaw from external navigation source

Table 8 – continued from previous page

260	External pitch	radian	Pitch from external navigation source
261	External roll	radian	Roll from external navigation source
262	External Roll Rate	rad/s	Roll rate from external navigation source
263	External Pitch Rate	rad/s	Pitch rate from external navigation source
264	External Yaw Rate	rad/s	Yaw rate from external navigation source
265	External Velocity North	m/s	Velocity North from external navigation source
266	External Velocity East	m/s	Velocity East from external navigation source
267	External Velocity Down	m/s	Velocity Down from external navigation source
268	External acceleration x body axis	m/s^2	Acceleration x body axis from external navigation
269	External acceleration y body axis	m/s^2	Acceleration y body axis from external navigation
270	External acceleration z body axis	m/s^2	Acceleration z body axis from external navigation
271	External GPS Time of Week	s	GNSS Time of week from external navigation sou
300	Time since Hardware Start-Up	sec	Time spent since power-on of the system
301	Used Memory Space	Byte	SD used memory space
302	Free Memory Space	Byte	SD free memory space
303	Dynamic Pressure	pascal	Physical measurement from Pitot
304	Static Pressure	pascal	Physical measurement from Pitot
305	Internal Temperature	kelvin (K)	Physical measurement from internal sensors
306	External Temperature	kelvin (K)	Physical measurement from Veronte sensors
307	Accelerometer – X Body Axis	m/s^2	Accelerometer measurement for X axis
308	Accelerometer – Y Body Axis	m/s^2	Accelerometer measurement for Y axis
309	Accelerometer – Z Body Axis	m/s^2	Accelerometer measurement for Z axis
310	Gyroscope – X Body Axis	rad/s (RDS)	Gyroscope measurement for X axis
311	Gyroscope – Y Body Axis	rad/s (RDS)	Gyroscope measurement for Y axis
312	Gyroscope – Z Body Axis	rad/s (RDS)	Gyroscope measurement for Z axis
313	Magnetometer – X Body Axis	tesla	Magnetometer measurement for X axis
314	Magnetometer – Y Body Axis	tesla	Magnetometer measurement for Y axis
315	Magnetometer – Z Body Axis	tesla	Magnetometer measurement for Z axis
322	Internal magnetometer raw X in SI	tesla	Magnetometer raw measurement for X axis
323	Internal magnetometer raw Y in SI	tesla	Magnetometer raw measurement for Y axis
324	Internal magnetometer raw Z in SI	tesla	Magnetometer raw measurement for Z axis
325	Internal magnetometer temperature	kelvin (K)	Magnetometer temperature
326	External LIS3MDL magnetometer raw X in SI	tesla	LIS3MDL external Magnetometer raw measureme
327	External LIS3MDL magnetometer raw Y in SI	tesla	LIS3MDL external Magnetometer raw measureme
328	External LIS3MDL magnetometer raw Z in SI	tesla	LIS3MDL external Magnetometer raw measureme
329	External LIS3MDL magnetometer temperature	kelvin (K)	LIS3MDL external Magnetometer temperature
330	IMU 0 raw accelerometer x measurement	m/s^2	IMU 0 raw accelerometer x measurement
331	IMU 0 raw accelerometer y measurement	m/s^2	IMU 0 raw accelerometer y measurement
332	IMU 0 raw accelerometer z measurement	m/s^2	IMU 0 raw accelerometer z measurement
333	IMU 0 raw gyroscope x measurement	rad/s (RDS)	IMU 0 raw gyroscope x measurement
334	IMU 0 raw gyroscope y measurement	rad/s (RDS)	IMU 0 raw gyroscope y measurement
335	IMU 0 raw gyroscope z measurement	rad/s (RDS)	IMU 0 raw gyroscope z measurement
336	IMU 0 temperature measurement	kelvin (K)	IMU 0 temperature measurement
337	IMU 1 raw accelerometer x measurement	m/s^2	IMU 1 raw accelerometer x measurement
338	IMU 1 raw accelerometer y measurement	m/s^2	IMU 1 raw accelerometer y measurement
339	IMU 1 raw accelerometer z measurement	m/s^2	IMU 1 raw accelerometer z measurement
340	IMU 1 raw gyroscope x measurement	rad/s (RDS)	IMU 1 raw gyroscope x measurement
341	IMU 1 raw gyroscope y measurement	rad/s (RDS)	IMU 1 raw gyroscope y measurement
342	IMU 1 raw gyroscope z measurement	rad/s (RDS)	IMU 1 raw gyroscope z measurement
343	IMU 1 temperature measurement	kelvin (K)	IMU 1 temperature measurement

Table 8 – continued from previous page

344	Static pressure sensor (MS56) raw measurement	pascal	Static pressure sensor 1 raw measurement
345	Static pressure sensor (MS56) temperature	kelvin (K)	Static pressure sensor 1 temperature
346	Dynamic pressure sensor raw measurement	pascal	Dynamic pressure sensor raw measurement
347	Dynamic pressure sensor temperature	kelvin (K)	Dynamic pressure sensor temperature
348	Static pressure sensor (HSC) raw measurement	pascal	Static pressure sensor 0 raw measurement
349	Static pressure sensor (HSC) temperature	kelvin (K)	Static pressure sensor 0 temperature
350	Vectornav Message Frequency	Hz	Frequency at which external navigation source Ve
351	Vectornav Raw Acc x measurement	m/s^2	Raw accelerometer X measurement from external
352	Vectornav Raw Acc y measurement	m/s^2	Raw accelerometer Y measurement from external
353	Vectornav Raw Acc z measurement	m/s^2	Raw accelerometer Z measurement from external
354	Vectornav Raw Gyr x measurement	m/s^2	Raw gyroscope X measurement from external nav
355	Vectornav Raw Gyr y measurement	m/s^2	Raw gyroscope Y measurement from external nav
356	Vectornav Raw Gyr z measurement	m/s^2	Raw gyroscope Z measurement from external nav
400	Power Input	volts	Power received by Veronte
401	Power Comicro 3.3V	volts	Power received by Veronte through 3.3V port
402	Power 5V	volts	Power received by Veronte through 5V port
403	Power Comicro Regulator	volts	Power received by Veronte Comicro Regulator
404	Power 3.6V	volts	Power received by Veronte through 3.6V port
405	CPU Temperature	kelvin (K)	Internal computer temperature

500	Longitude	radian	East-West geographic coordinate
501	Latitude	radian	North-South geographic coordinate
502	WGS84 Elevation (Height Over the Ellipsoid)	m	Elevation over WGS84 reference frame
503	MSL (Height Above Mean Sea Level) – Altitude	m	Altitude over the Mean Sea Level
504	AGL (Above Ground Level) – Height	m	Height Above Ground Level – Dependent on external
505	Ground Velocity North	m/s	Ground Velocity component in the North direction (N
506	Ground Velocity East	m/s	Ground Velocity component in the East direction (NE
507	Ground Velocity Down	m/s	Ground Velocity component in the resultant axis from
508	Sensor IAS (Indicated Air Speed)	m/s	Pitot-static measurement speed
509	Angle of Attack – AoA	radian	Angle between reference body line and flow direction
510	Sideslip	radian	Angle between the flow direction vector and the long
600-603	Temperature 1-4	kelvin (K)	Variables to be configured with external Temperature
650	Gimbal command yaw	customType	Yaw sent to the gimbal
651	Gimbal command pitch	customType	Pitch sent to the gimbal
652	Gimbal stick yaw	customType	Yaw received from the joystick controlling the gimba
653	Gimbal stick pitch	customType	Pitch received from the joystick controlling the gimba
654	Gimbal pitch correction 1	customType	Correction calculated by the gimbal for the pitch cont
655	Gimbal pitch correction 2	customType	Correction calculated by the gimbal for the pitch cont
656	Gimbal old joint 1	customType	Auxiliar variable for Gimbal control configuration
657	Gimbal old joint 2	customType	Auxiliar variable for Gimbal control configuration
658	Cos (gimbal yaw)	customType	Auxiliar variable for Gimbal control configuration
659	Sin (gimbal yaw)	customType	Auxiliar variable for Gimbal control configuration
660	Gimbal yaw radian/td>	customType	Auxiliar variable for Gimbal control configuration
661	Gimbal yaw output	customType	Yaw value the gimbal is outputting
662	Gimbal pitch output	customType	Pitch value the gimbal is outputting
700-705	RPM 1-6	rad/s (RDS)	RPM associated to pulse captured 1-6
750	Selected controller time step	sec	PID selected time step
751	Selected controller derivative filtered error	customType	PID selected derivative filtered error
752	Selected controller proportional action	customType	PID selected proportional action

Table 9 – continued from previous page

500	Longitude	radian	East-West geographic coordinate
753	Selected controller derivative action	customType	PID selected derivative action
754	Selected controller integral input	customType	PID selected integral input
755	Selected controller integral action	customType	PID selected integral action
756	Selected controller anti-windup input	customType	PID selected anti-windup input
757	Selected controller derivative error	customType	PID selected derivative error
800-815	PWM 1-16	customType	Pulse Width Modulation signal
900-915	Stick Input r1 – r16	customType	Raw stick measurement
1000-1031	Stick Input y1 – y32	customType	Servo position commanded from stick
1100-1104	Lidar 1-5 Distances	m	Variable configurable for Lidar distances
1105-1109	External range sensor 1-5 measurements	m	Variable configurable for external range sensors
1200	Cross-track Distance	customType	Shortest distance to desired path (perpendicular dista
1201	Radar AGL (Above Ground Level) – Height	m	User Variable
1202	Radar Speed Down	m/s	User Variable
1203	External Rotation for Follow Route	radian	Relative vector rotation when using Follow Route
1204	Time to Impact with Obstacles	sec	Time calculated with Distance to Obstacle and travel
1300-1309	Clock 1-10	sec	Configurable timers for automations – Clock 1 corres

1400	Velocity – X Body Axis	m/s	Velocity vector X compo
1401	Velocity – Y Body Axis	m/s	Velocity vector Y compo
1402	Velocity – Z Body Axis	m/s	Velocity vector Z compo
1403	Estimated Dynamic Pressure	pascal	Dynamic pressure sensor
1404	Barometric Pressure at Sea Level (QNH)	pascal	Introduced value for QN
1450-1453	Captured pulse 1-4	customType	Input values from pulses
1490	Internest raw x distance	m	Raw measurements for X
1491	Internest raw y distance	m	Raw measurements for Y
1492	Internest raw z distance	m	Raw measurements for Z
1493	Internest raw angle	radian	Raw measurements for in
1494	Internest raw xy standard deviation	m	Raw measurements for X
1495	Internest raw z standard deviation	m	Raw measurements for Z
1496	Internest raw angle standard deviation	radian	Raw measurements for in
1497	Internest position update frequency	Hz	Frequency at which Inter
1500	GNSS1 Time of Week	s	Data from GNSS1 modul
1501	GNSS1 ECEF Position X	m	Data from GNSS1 modul
1502	GNSS1 ECEF Position Y	m	Data from GNSS1 modul
1503	GNSS1 ECEF Position Z	m	Data from GNSS1 modul
1504	GNSS1 Longitude	radian	Data from GNSS1 modul
1505	GNSS1 Latitude	radian	Data from GNSS1 modul
1506	GNSS1 Height Above Ellipsoid (WGS84)	m	Data from GNSS1 modul
1507	GNSS1 Mean Sea Level (MSL)	m	Data from GNSS1 modul
1508	GNSS1 Above Ground Level (AGL)	m	Data from GNSS1 modul
1509	GNSS1 PDOP (Dilution of Precision of Position)	customType	Data from GNSS1 modul
1510	GNSS1 Accuracy	m	Data from GNSS1 modul
1511	GNSS1 Horizontal Accuracy Estimate	m	Data from GNSS1 modul
1512	GNSS1 Vertical Accuracy Estimate	m	Data from GNSS1 modul
1513	GNSS1 Velocity North	m/s	Data from GNSS1 modul
1514	GNSS1 Velocity East	m/s	Data from GNSS1 modul
1515	GNSS1 Velocity Down	m/s	Data from GNSS1 modul
1516	GNSS1 Speed Accuracy Estimate	m/s	Data from GNSS1 modul

Table 10 – continued from previous page

1517	GNSS1 Related Base Longitude	radian	Data from GNSS1 modul
1518	GNSS1 Related Base Latitude	radian	Data from GNSS1 modul
1519	GNSS1 Related Base WGS84 Altitude	m	Data from GNSS1 modul
1520	GNSS1 Related Base to Rover Azimuth	radian	Data from GNSS1 modul
1521	GNSS1 Related Base to Rover Elevation	radian	Data from GNSS1 modul
1522	GNSS1 Related Base to Rover Distance	m	Data from GNSS1 modul
1523	GNSS1 Related Base to Rover Accuracy	m	Data from GNSS1 modul
1524	GNSS1 Survey in Accuracy	m	Data from GNSS1 modul
1525	GNSS1 Related Base to Rover North	m	Data from GNSS1 modul
1526	GNSS1 Related Base to Rover East	m	Data from GNSS1 modul
1527	GNSS1 Related Base to Rover Down	m	Data from GNSS1 modul
1600	GNSS2 Time of Week	s	Data from GNSS2 modul
1601	GNSS2 ECEF Position X	m	Data from GNSS2 modul
1602	GNSS2 ECEF Position Y	m	Data from GNSS2 modul
1603	GNSS2 ECEF Position Z	m	Data from GNSS2 modul
1604	GNSS2 Longitude	radian	Data from GNSS2 modul
1605	GNSS2 Latitude	radian	Data from GNSS2 modul
1606	GNSS2 Height Above Ellipsoid (WGS84)	m	Data from GNSS2 modul
1607	GNSS2 Mean Sea Level (MSL)	m	Data from GNSS2 modul
1608	GNSS2 Above Ground Level (AGL)	m	Data from GNSS2 modul
1609	GNSS2 PDOP (Dilution of Precision of Position)	customType	Data from GNSS2 modul
1610	GNSS2 Accuracy	m	Data from GNSS2 modul
1611	GNSS2 Horizontal Accuracy Estimate	m	Data from GNSS2 modul
1612	GNSS2 Vertical Accuracy Estimate	m	Data from GNSS2 modul
1613	GNSS2 Velocity North	m/s	Data from GNSS2 modul
1614	GNSS2 Velocity East	m/s	Data from GNSS2 modul
1615	GNSS2 Velocity Down	m/s	Data from GNSS2 modul
1616	GNSS2 Speed Accuracy Estimate	m/s	Data from GNSS2 modul
1617	GNSS2 Related Base Longitude	radian	Data from GNSS2 modul
1618	GNSS2 Related Base Latitude	radian	Data from GNSS2 modul
1619	GNSS2 Related Base WGS84 Altitude	m	Data from GNSS2 modul
1620	GNSS2 Related Base to Rover Azimuth	radian	Data from GNSS2 modul
1621	GNSS2 Related Base to Rover Elevation	radian	Data from GNSS2 modul
1622	GNSS2 Related Base to Rover Distance	m	Data from GNSS2 modul
1623	GNSS2 Related Base to Rover Accuracy	m	Data from GNSS2 modul
1624	GNSS2 Survey in Accuracy	m	Data from GNSS2 modul
1625	GNSS2 Related Base to Rover North	m	Data from GNSS2 modul
1626	GNSS2 Related Base to Rover East	m	Data from GNSS2 modul
1627	GNSS2 Related Base to Rover Down	m	Data from GNSS2 modul
1700-1731	Actuator Output s1 – s32	customType	Configurable variable fro
1800-1895	Distance, azimuth and elevation to Object of Interest 1-32	m, radian and m (respectively)	Spherical coordinates to (

2000	RX Packet Error Rate (on board)	decimal	Value relating RX packets and expected RX
2001	TX Packet Error Rate (on board)	decimal	Value relating TX packets and expected TX
2002	Computed RX pkt/s used for RX PER	messages	Packages per second received to the UAV co
2003	Remote RX pkt/s used for TX PER	messages	Same as Computed RX pkt/s, received throu
2004	Computed TX pkt/s used for TX PER	messages	Packages per second transmitted to the UAV
2005	Remote TX pkt/s used for RX PER	messages	Same as Computed TX pkt/s, received throu
2019	Stick RX rate	Hz	Stick messages received per second

Table 11 – continued from previous page

2020	Position fix Time	sec	Time spend with GNSS not losing fix
2040	Tunnel producer receive frequency 1	Hz	Frequency at which the Tunnel producer rec
2041	Tunnel producer receive frequency 2	Hz	Frequency at which the Tunnel producer rec
2042	Tunnel producer receive frequency 3	Hz	Frequency at which the Tunnel producer rec
2043	Tunnel consumer send frequency 1	Hz	Frequency at which the Tunnel consumer se
2044	Tunnel consumer send frequency 2	Hz	Frequency at which the Tunnel consumer se
2045	Tunnel consumer send frequency 3	Hz	Frequency at which the Tunnel consumer se
2046	Max duration of step in acquisition	sec	Longest time duration from a step in CIO
2047	Acquisition task timestep	sec	Average period for executing the acquisition
2048	Acquisition task maximum timestep	sec	Maximum period for executing the acquisiti
2049	Cross core message queue CPU ratio	percentage	% of time of CPU that CIO waits for inter-c
2050	Acquisition task average CPU ratio	percentage	Average % of time of CPU from the acquisit
2051	Acquisition task maximum CPU ratio	percentage	Maximum % of time of CPU from the acqui
2052	Acquisition task average time	sec	Average time acquisition task has used
2053	Acquisition task maximum time	sec	Maximum time acquisition task has used
2054	CIO Max time	sec	Maximum time in acquisition from Core Inp
2055	CIO average time	sec	Average time in acquisition from Core Input
2094	GNC task average CPU ratio	percentage	Average % of time of CPU from GNC task
2095	GNC task maximum CPU ratio	percentage	Maximum % of time of CPU from GNC tas
2096	GNC task average time	sec	Average time spent on GNC task
2097	GNC task maximum time	sec	Maximum time spent on GNC task
2098	GNC task maximum timestep	sec	Maximum execution period for GNC task
2099	Max duration of step in GNC	sec	Maximum duration of one step in GNC
2100	Gyroscope Based on Accelerometer – X Body Axis	rad/s (RDS)	Gyroscope measurements obtained from acc
2101	Gyroscope Based on Accelerometer – Y Body Axis	rad/s (RDS)	Gyroscope measurements obtained from acc
2102	Gyroscope Based on Accelerometer – Z Body Axis	rad/s (RDS)	Gyroscope measurements obtained from acc
2103	Acceleration North	m/s^2	Acceleration in the North direction (NED C
2104	Acceleration East	m/s^2	Acceleration in the East direction (NED Co
2105	Acceleration Down	m/s^2	Acceleration in the Down direction (NED C
2112	Estimated Dem	m	Altitude given by the estimated Digital Elev
2200	Curve Length Covered	m	Total distance from current mission leg cove
2201	Curve Length	m	Total distance from current mission leg
2202	Curve Length Pending	m	Total distance from current mission leg to be
2203	Curve Parameter Covered	customType	Total amount from current mission leg cover
2204	Curve Parameter Range	customType	Total distance from current mission leg acco
2205	Curve Parameter Pending	customType	Total distance from current mission leg to be
2250-2259	Reserved 1-10	customType	System reserved variables
2300-2302	Joint 1-3 of Gimbal 1	radian	Variables for Gimbal 1 configuration – Angl
2303-2305	Joint 1-3 of Gimbal 2	radian	Variables for Gimbal 2 configuration – Angl
2400-2419	Control Output u1 – u20	customType	PID Output values that are transformed into
2500-2519	Stick Input u1-u20	customType	Intermediate values from stick used for arca
2600-2619	Stick Input d1-d20	customType	Intermediate values from stick used for arca
2700-2712	Operation Guidance 1-13	customType	Configurable values used in different guidan
2800	Wind Velocity North	m/s	Wind velocity vector pointing North directio
2801	Wind Velocity East	m/s	Wind velocity vector pointing East direction
2802	Wind Velocity Down	m/s	Wind velocity vector pointing Down directio
2803	Wind Velocity North Estimation Covariance	m/s	Wind velocity vector pointing North directio
2804	Cross North-East Wind Velocity Estimation Covariance	m/s	Wind velocity vector pointing cross North-E
2805	Wind Velocity Estimation Uncertainty (Element 2-0)	m/s	2-0 element from covariance matrix in wind
2806	Wind Velocity Estimation Uncertainty (Element 0-1)	m/s	0-1 element from covariance matrix in wind

Table 11 – continued from previous page

2807	Wind Velocity Estimation Uncertainty (Element 1-1)	m/s	1-1 element from covariance matrix in wind
2808	Wind Velocity Estimation Uncertainty (Element 2-1)	m/s	2-1 element from covariance matrix in wind
2809	Wind Velocity Estimation Uncertainty (Element 0-2)	m/s	0-2 element from covariance matrix in wind
2810	Wind Velocity Estimation Uncertainty (Element 1-2)	m/s	1-2 element from covariance matrix in wind
2811	Wind Velocity Estimation Uncertainty (Element 2-2)	m/s	2-2 element from covariance matrix in wind
2900	MSL Right from Actual QNH and Pressure Measurement	m	Mean Sea Level obtained from Actual QNH
2901	MSL for ISA and Pressure Measurement	m	Mean Sea Level calculated for ISO Internati
2902	Time Since Entering Current Phase	sec	Time-lapse considered since entering the cu
2903	GNC Timestep	sec	Task execution period from GNC
2904	Total Flight Time	s	Time-lapse since the vehicle exited Standby
2905	Total Flight Distance	m	Distance covered by the vehicle in all missio
2906	Reception Frequency of Simulated Navigation Data	Hz	Frequency at which the system receives Sim
2907	Reception Frequency of External Navigation Data	Hz	Frequency at which the system receives Exte
2908-2927	Time Spent Within Phase 1-20	sec	Time-lapse spent by the vehicle in any phase
3000-3031	Simulation Variable 1-32	customType	Variables used for Simulation data
4100	Zero	customType	Constant value 0
4101	Rvar disabled	customType	Disabled variable

6.2.7.4.2 BIT

ID	Name	Units/Values	Descripti
0-2	Initialisation values	–	Bit for fail
3	System Readiness	–	System is
4	Writing telemetry	–	Telemetry
5	Readiness to flight	–	System is
6	File System state	–	System fil
7	System state	–	System is
8	Memory Allocation	–	Memory a
9	PDI State	–	PDI files a
10	4X Arbiter state	–	Arbiter is
50	Sensors error	–	Error/Run
51	Sensors-Main IMU	–	0 for not e
52	Sensors-Secondary IMU	–	0 for not e
53	Sensors-Magnetometer	–	Internal m
54	Sensors-External magnetometer (HMR2300)	–	HMR2300
55	Sensors-External Magnetometer (LIS3MDL)	–	LIS3MDL
56	Sensors-Static pressure (HSC)	–	HSC Stati
57	Sensors-Static pressure (MS56)	–	MS56 Stat
58	Sensors-Dynamic pressure (HSC)	–	HSC Dyna
59	Sensors-External I2C devices	–	0 for not e
60-64	Sensors-External I2C 0-4	–	External c
65-72	SCI A-D	–	Serial Cor
73-76	CAN A-B	–	CAN Bus
77	Vectornav States for GPS Fix, IMU, Mag/Press, GPS Error, Navigation-Vectornav state	–	0 for error
100	Position not Fixed	–	GNSS data
101	Georeferenced Area	–	0 for being
102-103	CAN A-B	–	CAN A or
104-105	Stick PPM 1-2 detection	–	Stick PPM

Table 12 – continued from previ

ID	Name	Units/Values	Descriptio
106-108	SCI A-C State	–	Defines th
109	MCBSP	–	Defines th
110	Stick state	–	Show if th
111-112	CAN A-B	–	CAN A or
113	Iridium Readiness	–	Iridium re
114-115	EKF	–	Extended
116	Radar Altimeter CAN-RX	–	Radar Alti
117-118	Power A-B State	–	Power A-E
130	EFK Navigation State	–	Extended
150	External VCP State	–	External V
160	External var Navigation Error	–	External N
180	Attitude	–	Kind of at
181	Reserved State	–	System Bi
190	Internest ultrasound position status	–	0 for error
191	Internest ultrasound angle status	–	0 for error
200-206	(D)GNSS1	–	(Differenti
300-306	(D)GNSS2	–	(Differenti
400, 403-405	Reserved	–	System Bi
401	GNC progress	–	GNC Step
402	Acquisition progress	–	Acquisitio
500	Ground effect compensation	–	Ground ef
700-731	Servo 1-32 state	–	Servos sta
800-815	PWM GPIO State	–	PWM GPI
816-819	IO 1-4 State	–	Input/Outp
820-822	RSSI LED 1-3 State	–	Received S
1000-1009	Simulation BIT 1-10 State	–	Variables t
1200-1299	User BIT 1-100	–	Free bits f
2200	BIT Dummy	–	Bit for cor

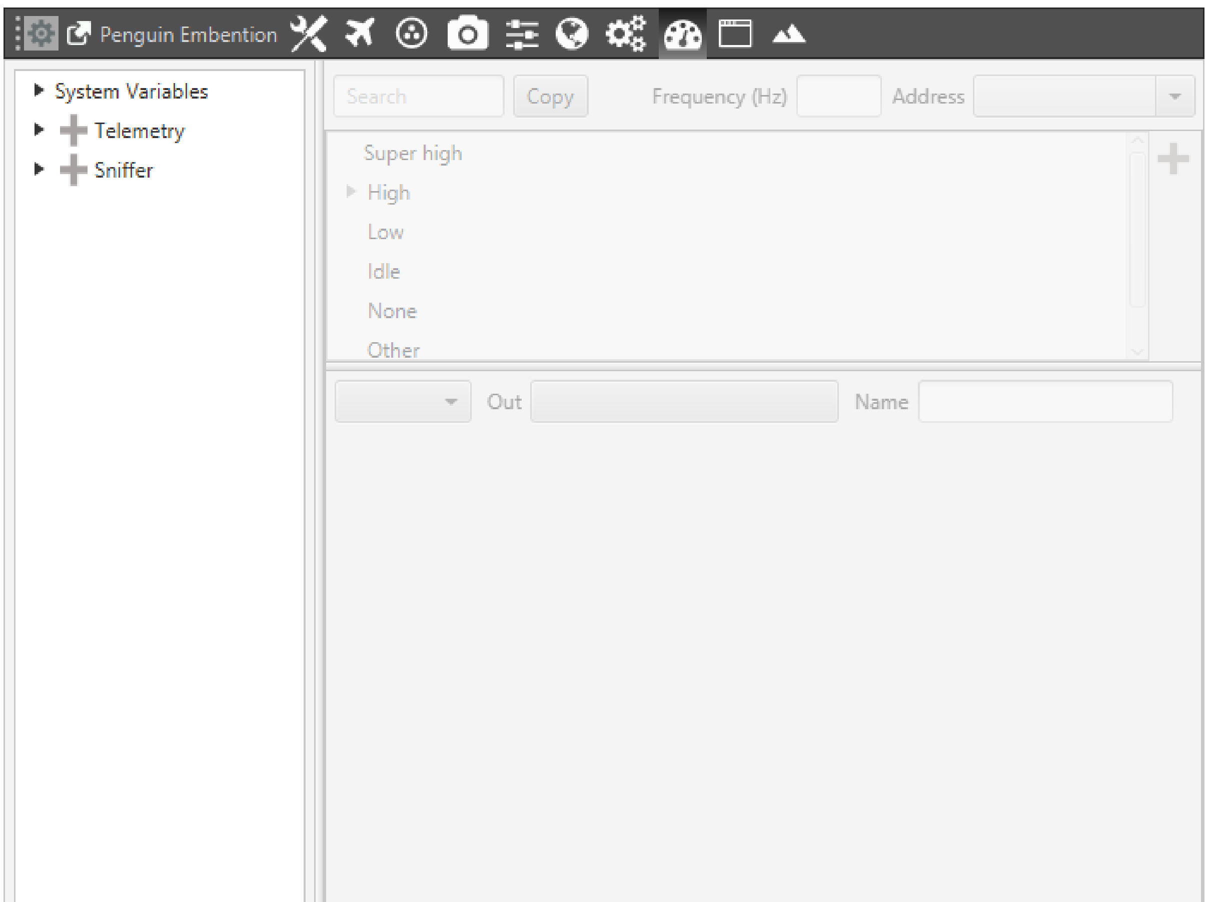
6.2.7.4.3 16 VAR

ID	Name	Units/Values	Description
0	Control Mode	–	Index pointing to the Control mode in use
1	Mission Phase ID	–	Phase Identifier
2 -18	ADC Channel 1-17	–	Internal ADC 1, 7-8 ADC 1-5
19	Current envelope	–	Index pointing to the envelope in use
20	Counter for C2 system BIT	–	Index for number of cycles from Core 2
80	Detour calculation identifier	–	Index for a route change
100-101	GNSS1-2 Satellites Number Information	–	Number of Satellites Used in Solution
200	Radar Altimeter State	–	Index for the radar altimeter state
201	Current Section	–	Index showing section
202	Last Achieved Section	–	Index showing sections achieved
203	Track Stage	–	Index showed when a route change happens
204	Current patchset ID	–	Index showing the patchset
303-305	HMR2300 Magnetometer Raw Measurement	–	X, Y & Z
310-311	Iridium	–	Sent & Received
398	VectorNav Mode	–	Index showing external source VectorNav mode

Table 13 – continued from previous page

ID	Name	Units/Values	Description
399	Identifier of max duration step in acquisition	–	Index
400	Internest raw status	–	Index
401	Navigation source	–	Index pointing to the primary navigation source
402	Raw position source identifier	–	Index
403-410	Sensor selection	–	Static & Dynamic Pressure, Primary accelerometer & g
425	Step in GNC max duration	–	Identifier for the step with maximum duration
450-453	CAN errors	–	Tx & Rx
490	Number of moving objects detected	–	Index
497-499	Configuration	–	Config manager status, global configuration state of fil
500	Transponder sequence number	–	Value of the transponder sequence number
501	System Reserved	–	System variables not configurable
550-553	Command 1-4 gimbal	–	Index
900-909	Simulation variables	–	Variables used for Simulation data
1000-1099	User Variables	–	Free variables for the user to use
2000	Uvar State	–	Index
2001	Zero	–	Uvar with constant 0 value

The following figure shows the menu to manage the variables in the system.



Variables Configuration Menu

The three options available on the **Variables** menu are:

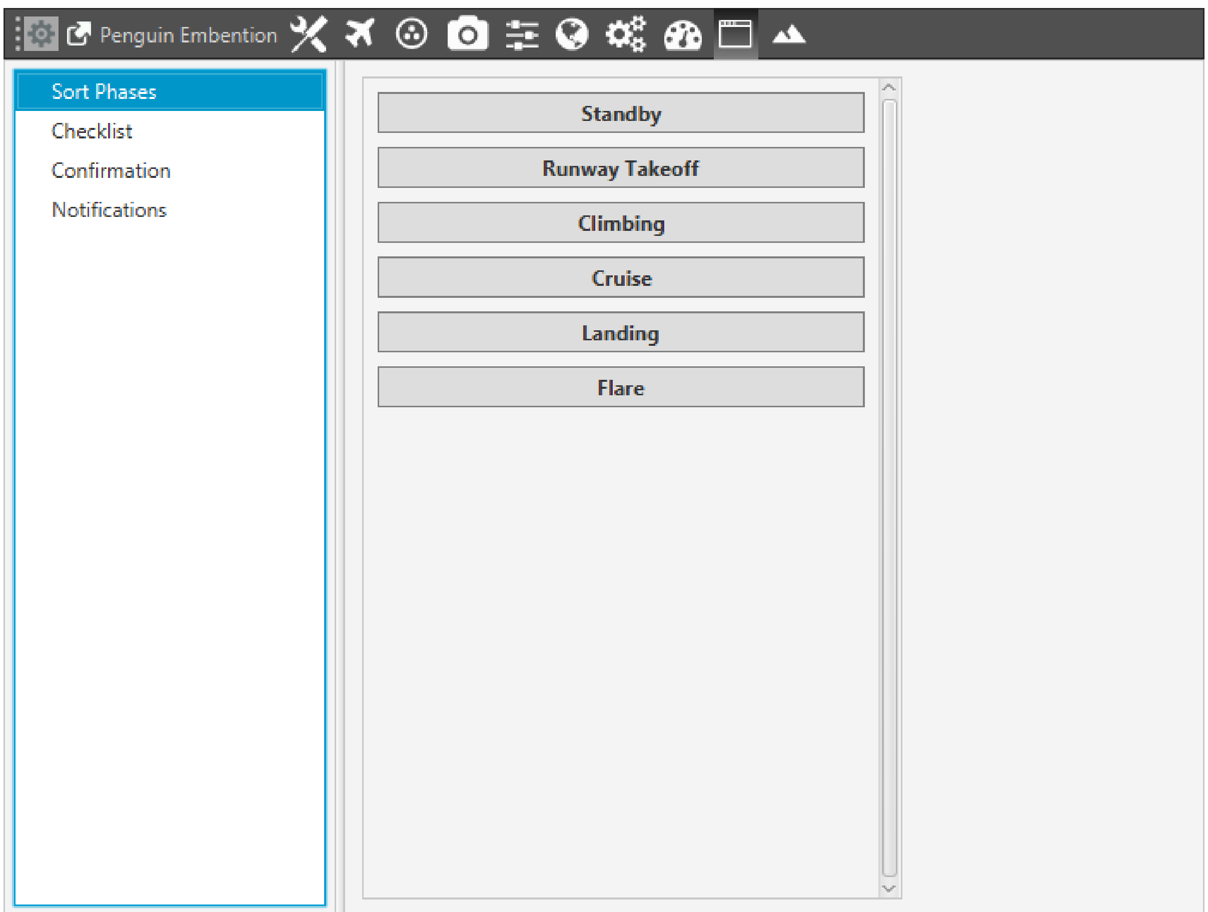
Value	Description	
System Variables	In this menu,	User can find the name of all system variables and carry out operations with them
Telemetry	In this menu user can set the priority of the system variables to be saved. Also, the variables that will be sent between Veronte	
Sniffer	Option used to configure what messages user wants to receive	

6.2.8 Panel

In this menu, the user can decide how the phases will look in Veronte Panel, create checklists for each phase, create manual confirmation for particular phase changes and configure some notification that will appear behind the Veronte Unit (Side Panel) upon activation.

6.2.8.1 Short Phases

The first menu gives the option to arrange the phases that will appear on the panel. Just dragging a phase and moving it wherever is desired will change its location in the panel. More than one phase can be arranged in the same line.



Phases Disposition

6.2.8.2 Checklists

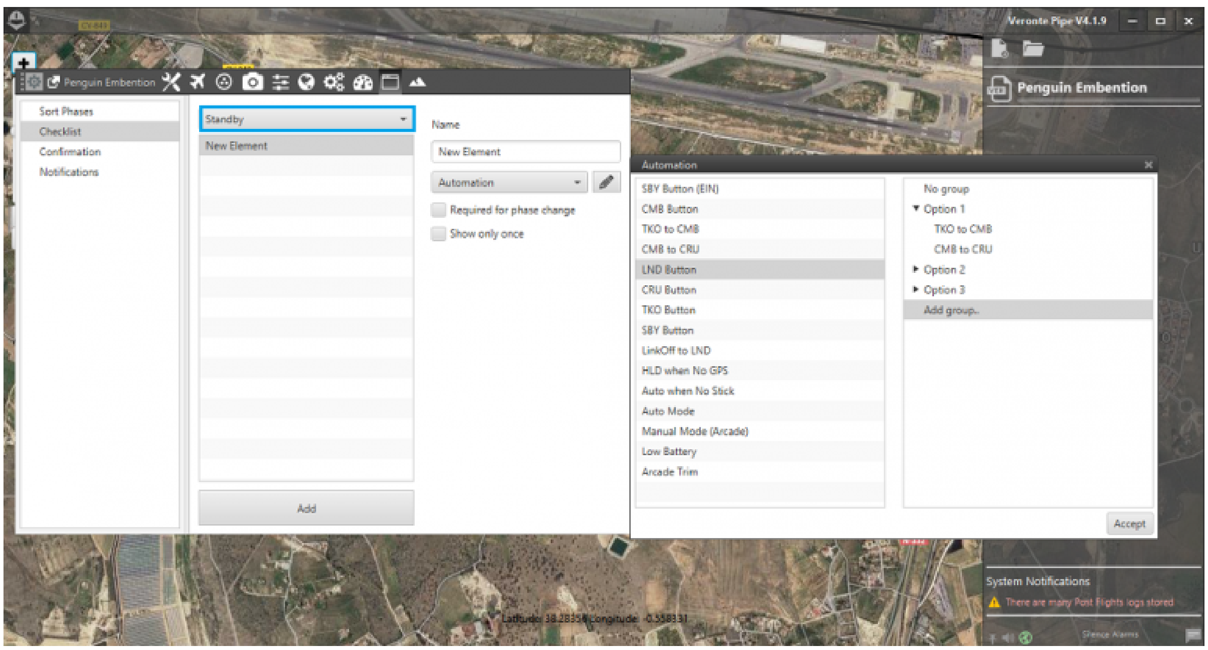
This feature is used to make sure that some requirements have been accomplished, for example, prior to a phase change or to avoid a possible malfunction.

In this menu, the checklist that will appear in the Panel is configured. The blue box displays all the phases configured for the operation. In each one of them, new elements for the checklist can be added with the button **Add**.

The configurable parameters for each element are:

Name: the name that will identify the element. Type: the element chosen from the checklist can be one of the following types.

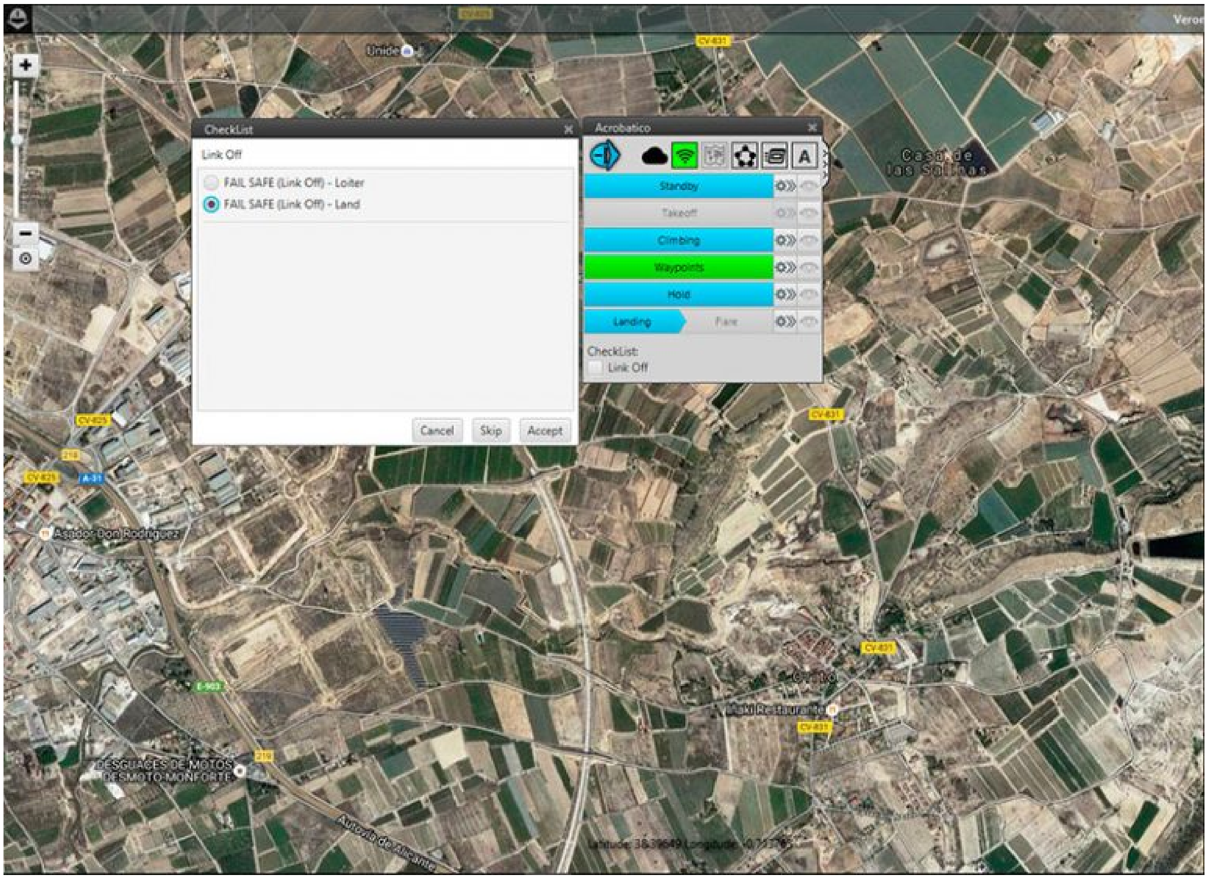
- **None:** any action is performed, been just a check for the user to do something external to Pipe
- **Command:** send to the UAV a position, a yaw angle or the wind velocity in the three axes.
- **In Range Check:** Allows checking if a variable is between the range selected.
- **Atmosphere:** set the atmosphere parameters (temperature…).
- **Automation:** selection between a set of automations, in order to only activate one of them for the phase that the aircraft is entering in.



Checklist Elements

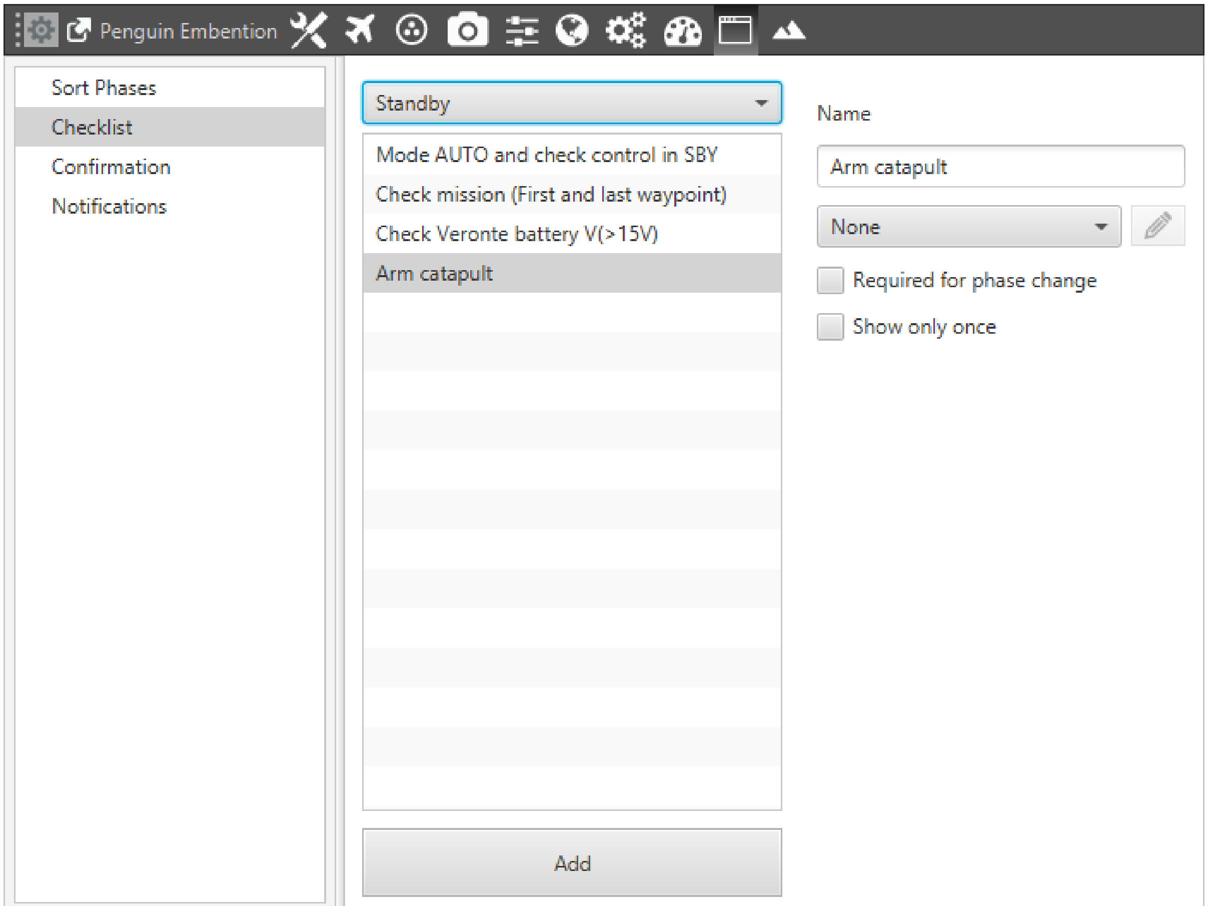
Drag the automations into a group to create the selection, it is possible to have more than one.

As an example, the following figure shows an automation selection on the checklist. The user has to select between two automations previously included in a group. Once one of them is selected, the other one will be automatically deactivated.



Checklist Verification

- **Required for phase change:** the element must be checked to go to another phase.
- **Show only once:** the check will only appear the first time its phase is executed.

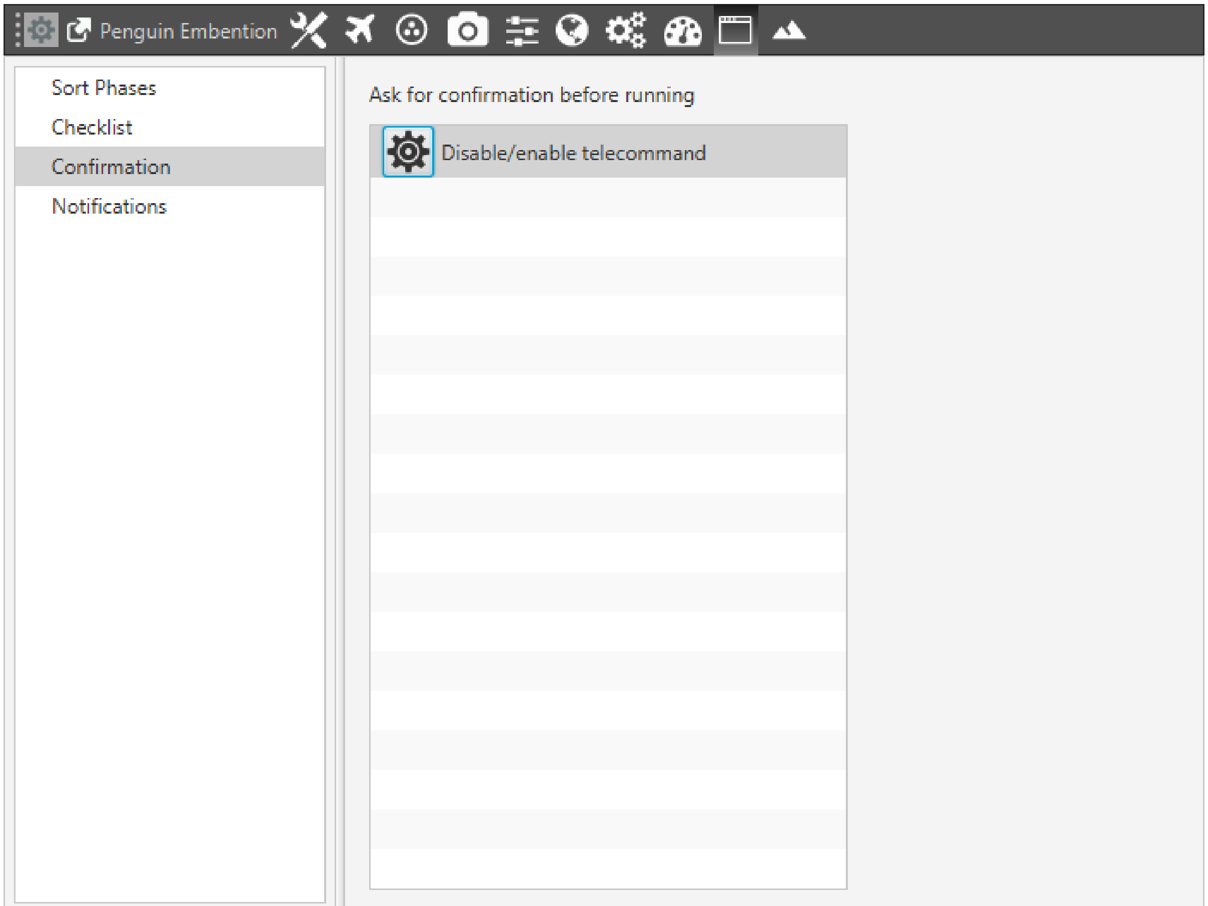


Checklist Configuration

6.2.8.3 Confirmation

This menu is used to create confirmation messages that appear when a button is pressed. It is a safety system for not pressing any keys by mistake.

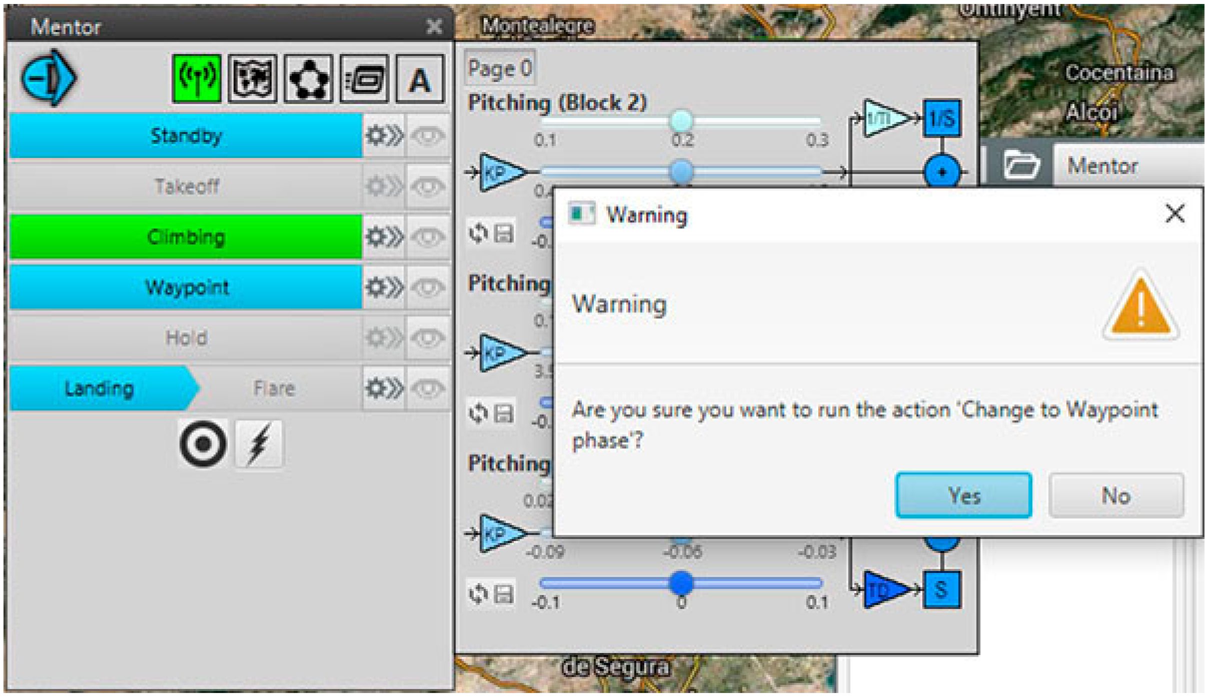
By default, this window displays only one button to ask confirmation from: the radio link.



Panel Confirmation

For the other custom buttons created by the user for Veronte panel, the confirmation option is included when creating the button. When configuring the Button event, the system provides a check to include a confirmation when selecting that option (visit *Automations*).

When the user is trying to do an action (previously selected), Veronte Pipe will show a warning message like the following, asking us confirmation.

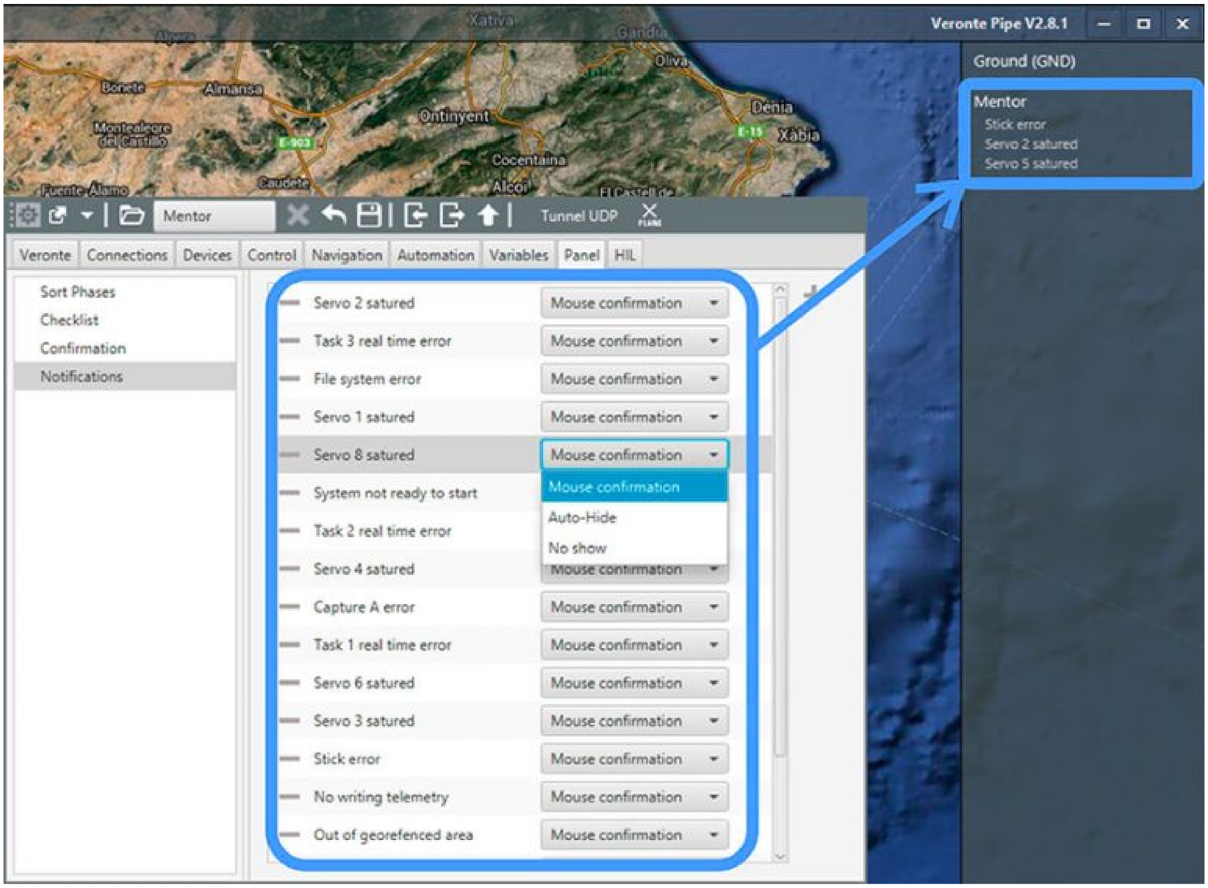


Confirmation – Example

6.2.8.4 Notifications

This menu is used to select which notifications will appear in the Side Panel and the way in which the user wants to hide them:

- **Mouse confirmation** – It is necessary to pass with the mouse on notifications in order to hide them.
- **Auto-hide** – Notification will auto hide in few seconds.
- **No show** – The notifications will not show up.



Panel – Notifications

Besides the notifications that appear by default on this menu, it is possible to include some other system alarms or even custom ones created by the user. Click on “+” to display a popup window that allows the selection of a new variable. It is also possible to delete the notifications shown by default, click on “-” to eliminate it from the error list.

In the following table, it will be described the error types of the notification panel.

Notification	Cause	Possible solution
GPS navigation down	The system is not connected to GPS signal	Check the GPS signal in the area and the GPS antenna connection
No writing telemetry	System is not writing telemetry in the onboard log	Check the Air micro SD. It could be corrupted or without memory available
File system error	System has one or more corrupted files	Try to repeat the operation or check the micro SD files
Out of georeferenced area	The GPS position is out of the areas with terrain altitude information	Go to Terrain Profile in Mission Panel and move manually Coarse and Fine areas or choose Auto to make the system doing it automatically and save.
Power A error	The alimentation voltage is out of the admitted range	Go to "Electrical" section of the manual to check admitted voltages
Servo X satured	The servo number X is actually satured	Check the servos panel to set the moving range
System not ready to start	No GPS connection/Licence expired	See "GPS navigation down"/Update the licence expiration
Task X real time error	System is not able to complete all tasks: 1 means High level,5 means Low level	Check system variables in Variables Panel and move the operations to a lower level.
Stick error	The stick is not connected to Air autopilot	Check radio signal and stick connection (in case of HIL simulation it will be simulated)
Capture A error	The Ground is not receiving the stick signal	Check the Ground-Stick connection
Inside Polygon X	The aircraft is actually inside the selected polygon (X is the polygon number)	Exit the aircraft from the polygon

Warning: In some cases, it will be necessary to restart the autopilot once the solution of one of this problems has been applied ("Out of georeferenced area" for example).

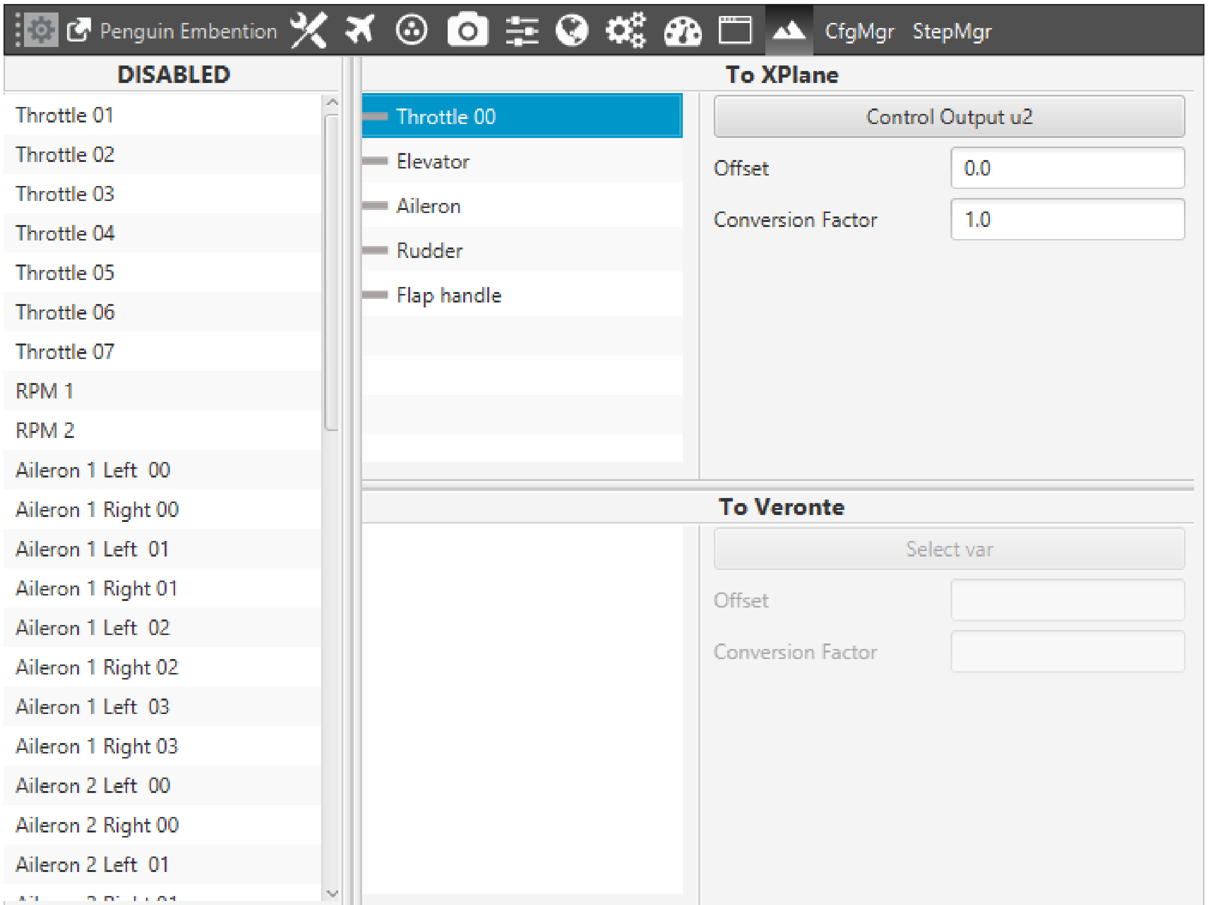
6.2.9 HIL

Professional Hardware In the Loop (HIL) Simulator package is a powerful tool for Veronte Autopilot integration, development and operator training; permitting to extensively operate the system in a safe environment, prior to conducting real flight operations

There are 2 configuration items on Veronte Pipe, one relating communications between the application and the X-Plane simulator and another one refereeing the autopilot configuration. Users can find the complete process including X-Plane settings in *Professional HIL*.

6.2.9.1 Autopilot Configuration

HIL simulation tab is available within Veronte Autopilot setup toolbar. The user can link the variables on Veronte Autopilot with the corresponding ones in X-Plane simulator.



Veronte Pipe – HIL Setup

In this panel, X-Plane variables are available on the left side (**Disables**). In addition, it can be seen two section more **To XPlane** and **To Veronte**.

In order to configure the simulation variables, users have to:

1. Enable the ones that have been configured in the aircraft model (**Plane Marker**). Just drag and drop them into **To Xplane** section.

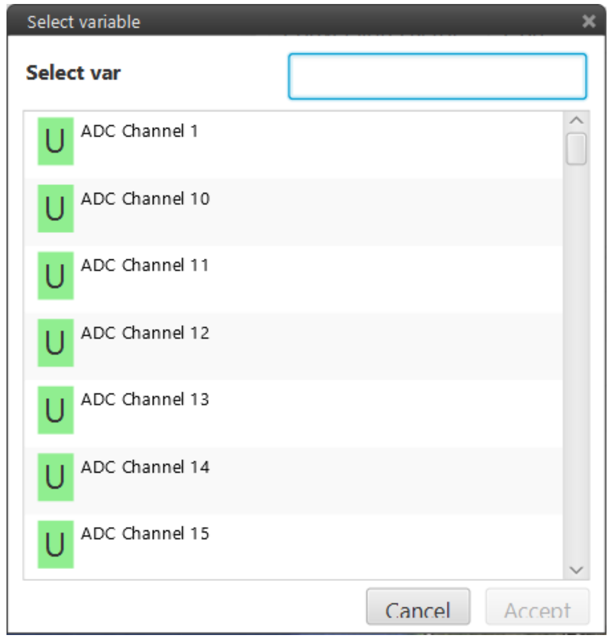
   > **Warning:** Always make sure that surfaces are moving in the right direction and with the correct deflection angle.

   To avoid mistakes is possible to set a positive fixed deflection (Control Tab) in Standby phase for all surfaces and control surface deflections in the X-Plane model.

   Surface control variables are of two types:

   - Radians measure (variables with numbers)
   - Degrees measure (variables with no numbers)

2. Once X-Plane variables have been enabled, select the actuator variable (Control Output) that matches with the ones in Veronte autopilot. A new window will be displayed for each variable.



HIL Panel

3. Set a **Conversion Factor** or **Offset**, if it is necessary. Conversion factor multiplies the Veronte output signal and can be used in case units on Veronte and the X-Plane simulator do not match (Surfaces in X-Plane move normally in a [-1,1] range). The following operation allows to converting an angle [deg] measure to the X-Plane form:

   (Angle + Offset) Conv.Factor = …

   When Angle and Offset are measured in [rad] and the Conversion factor is a constant (normally it can be calculated as 1/(deflection angle in [rad]).

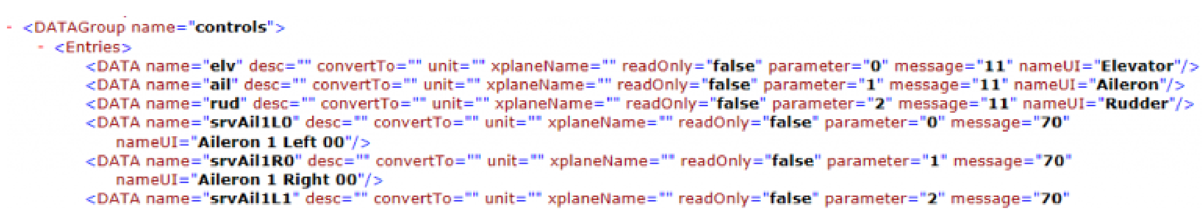
   In the case of [rad] measures, the Conversion factor must be set in 57,29578 ([deg]-[rad] conv. factor).

   Finally, it is necessary to configure the communication between Veronte Pipe and X-Plane, see section below.

6.2.9.2 Adding New variable

When there is any varible that doesn't appear in Veronte Pipe, it is possible to add a new one and linked it with the one correspondent in X-Plane. To do this, user have to edit the file "XplaneDATAGroupConfig.xml".

This file contains the following information:



```
- <DATAGroup name="controls">
  - <Entries>
      <DATA name="elv" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="0" message="11" nameUI="Elevator"/>
      <DATA name="ail" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="1" message="11" nameUI="Aileron"/>
      <DATA name="rud" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="2" message="11" nameUI="Rudder"/>
      <DATA name="srvAil1L0" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="0" message="70"
        nameUI="Aileron 1 Left 00"/>
      <DATA name="srvAil1R0" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="1" message="70"
        nameUI="Aileron 1 Right 00"/>
      <DATA name="srvAil1L1" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="2" message="70"
```

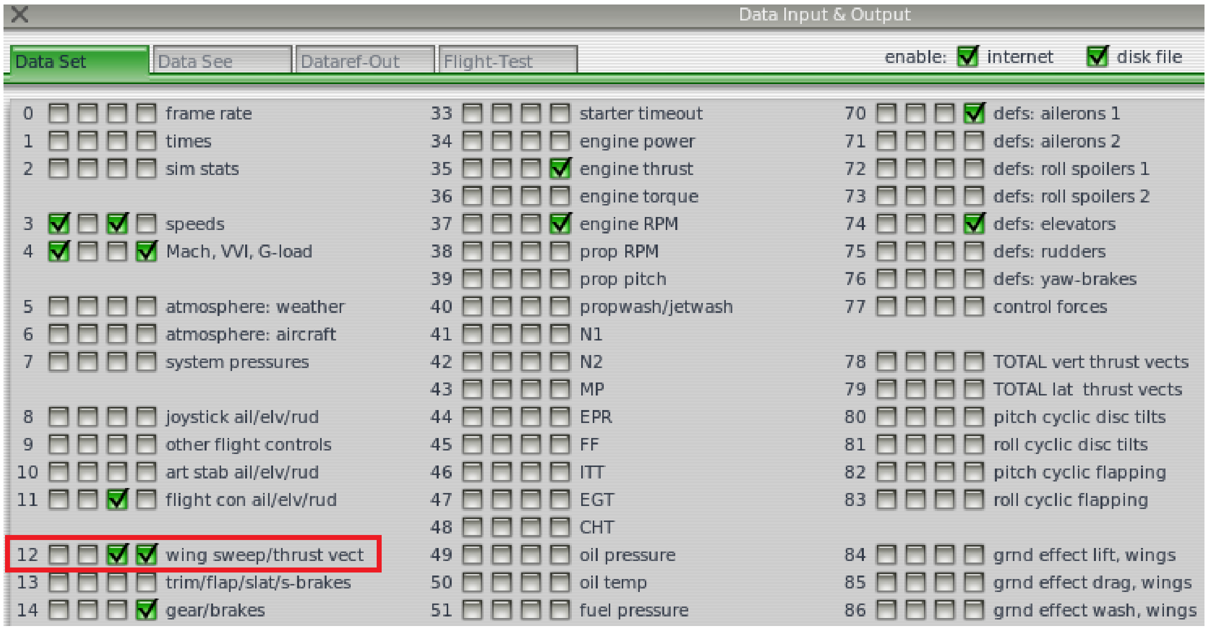
XplaneDATAGroupConfig.xml

- **Data name:** variable's name in X-Plane.
- **Parameter and message:** these values must be set correctly to link the variables.

- **nameUI:** variable's name in Veronte Pipe. User can type the name that will appear in the list.

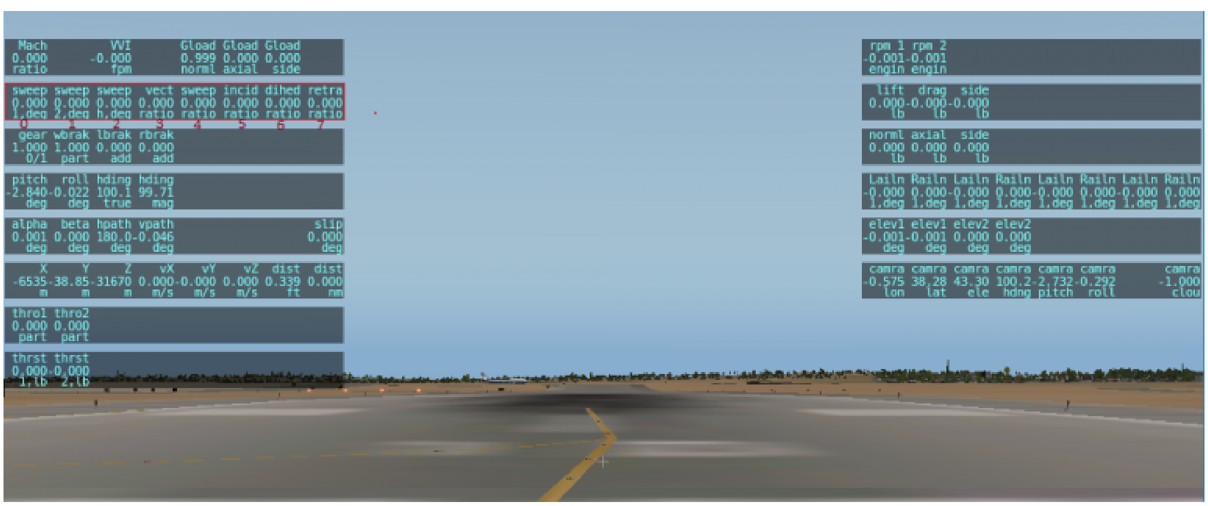
Consider the following example that explains how to add a new variable. Imagine you have a wing with variable incidence (configurable in Plane Maker) and this parameter doesn't appear in Veronte Pipe.

First, user have to check the variable's name, parameter and message in X-Plane. Open X-Plane and go to Setting/Data Input & Output.



Data Input & Output

The message is the number that appears next to each variable. In this case, the data related to a wing with variable incidence is "wing sweep/thrust vect" (see X-Plane manual), so message has to be set with 12. Variable's name and parameter can be found in the following image:



X-Plane

Variables's name is **incid** and parameter is **5**. To find out the parameter's number, user have to count starting from "0" and the left. Once all data is known, it is possible to edit the .xml file and add the new variable with a text editor. The result is shown in the following image:

```
- <DATAGroup name="controls">
  - <Entries>
      <DATA name="elv" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="0" message="11" nameUI="Elevator"/>
      <DATA name="ail" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="1" message="11" nameUI="Aileron"/>
      <DATA name="rud" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="2" message="11" nameUI="Rudder"/>
      <DATA name="incid" desc="" convertTo="" unit="" xplaneName="" readOnly="false" parameter="5" message="12" nameUI="Canard"/>
```

XplaneDATAGroupConfig.xml

The nameUI has been set as **Canard** and this will be displayed in the variables list. Now, user can simulate this variable in X-Plane from Veronte Pipe.

6.2.9.3 Veronte Pipe communications

In order to start the simulation, select the HIL option (**1**) on the Veronte Unit, this is available on the side menu . Popup screen will be displayed (**2**) for selecting the type of simulation and configuring the parameters.

Veronte Pipe – HIL Communications

Changing the default values is only recommended for advanced users. Press start in order to start the data transfer between X-Plane and Veronte Autopilot.

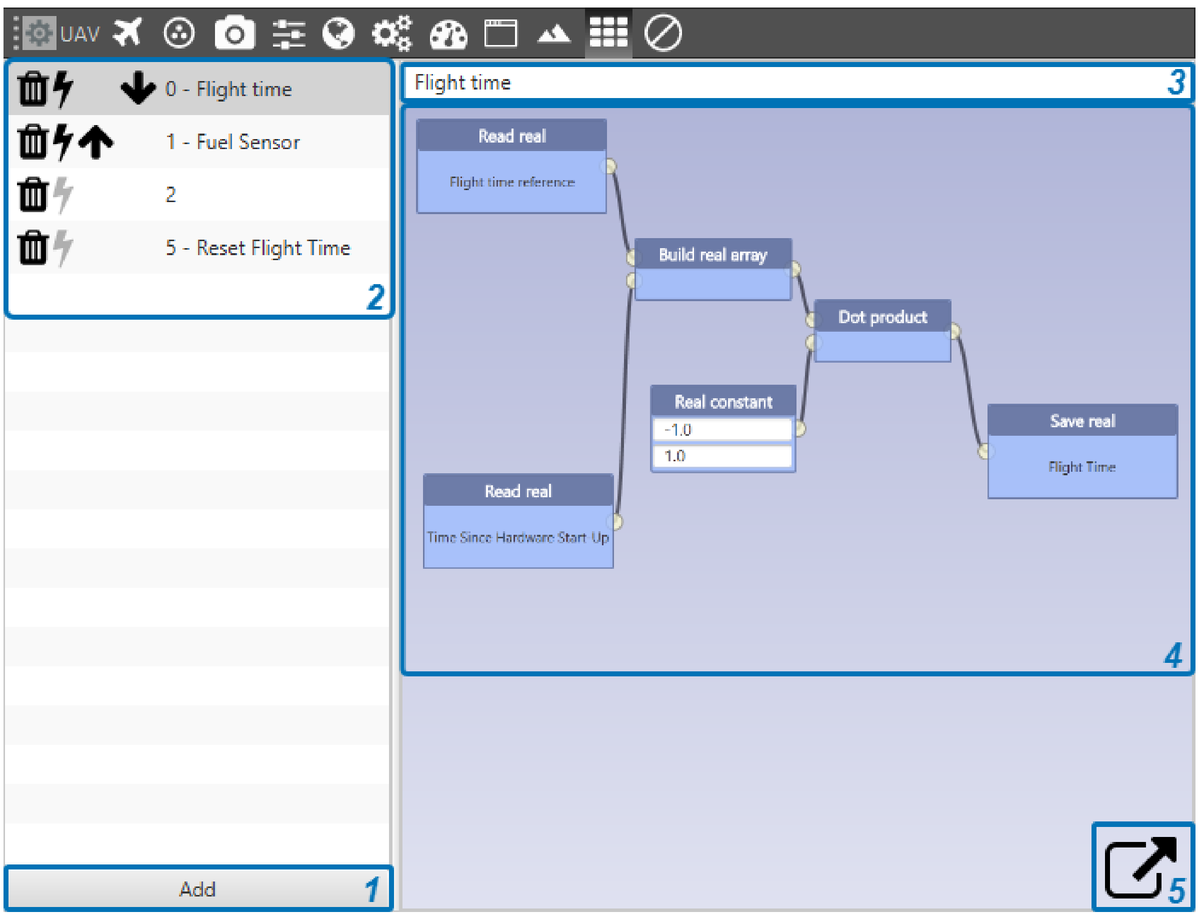
> **Warning:** The simulation must be started when the aircraft is in the Initial phase (the one that gets in once is powered). In this phase the X-Plane will simulate the GPS signal to locate the autopilot in the place indicated by the airport on X-Plane.

6.2.10 Blocks

Veronte autopilot allows the user to perform custom operations, thus adapting to the user's data management requirements. Within this menu the user can select the desired input variables, a wide variety of operations to transform the latter, and the output variables where the final processed value will be stored. For instance, in this menu it is possible to configure a Lidar sensor, converting the measured variable by the sensor (voltage) into a physical variable (meters).

The menu to create custom operations is called **Blocks** – see the Figure below. Within the menu the user has:

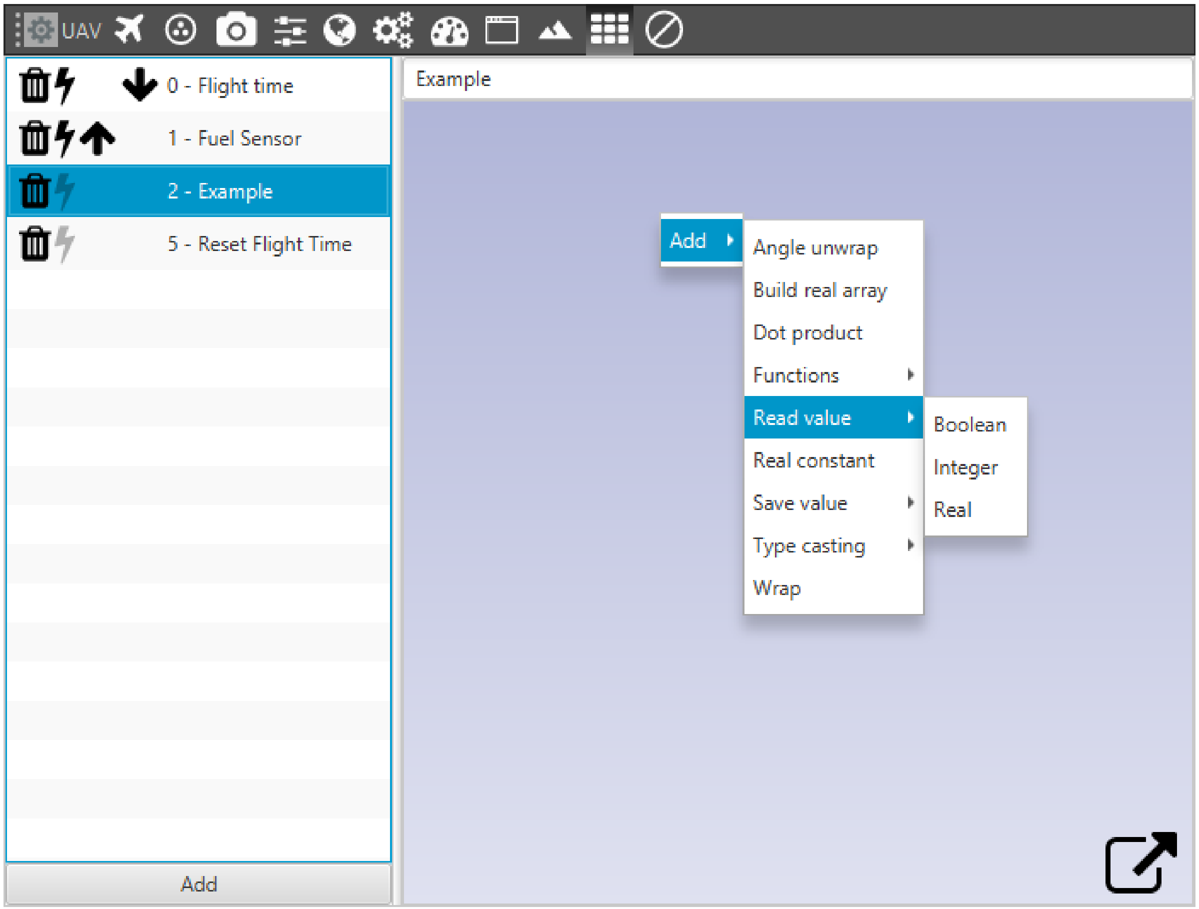
1. *Add button* to add a new custom operation.
2. Custom operations display. Next to every operation entry the user will find the following buttons . The *trash bin* deletes the operation, the *lightning bolt* marks that operation so it is executed continuously, and the *arrows* allow the user to arrange the priority of the operations that have been marked. The non-marked operations are reserved to be executed on demand by creating an Automation.
3. Operation's name.
4. Operation definition: input variable(s), output variable(s) and transformations to be executed.
5. Clicking on the button will enlarge the working space



Blocks Menu

As mentioned above, when the *lightning bolt* is marked the operation is then executed at the same frequency as the GNC algorithm (Guidance, Navigation and Control). Following, the blocks available to define an operation are presented.

To add a block, the user needs to right-click on the background of the menu – see Figure below.



Adding Blocks

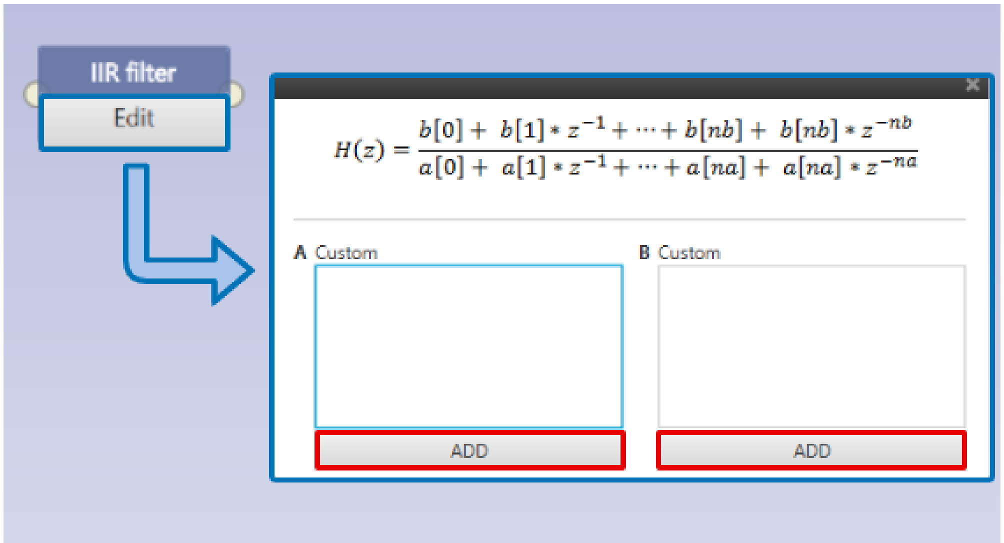
- The user will commonly want to start reading a variable coming from either the autopilot or a sensor/device. The option **Read value** must be selected, choosing among *boolean*, *integer* or *real* as the variable type.
- **Type casting**: this block is used to change from one data type to another. The options available are *Bool to real*, *Integer to real*, *Real to bool* and *Real to integer*.
- **Wrap**: the output variable is wrapped to keep a value between the upper and lower limits. For example, if the desired Inidicated Air Speed (IAS) is wrapped between 0 and 12 m/s, the output variable will have a value of 12 when the desired IAS is greater than this value, and a value of 0 when it is negative.

Wrap Block

- **Angle Unwrap**: when an angle is wrapped in the PIDs, e.g. the heading angle, its value is delimited between π and $-\pi$ radians. The operation presented here undoes that calculation by representing the angle in a range starting at 0 radians.

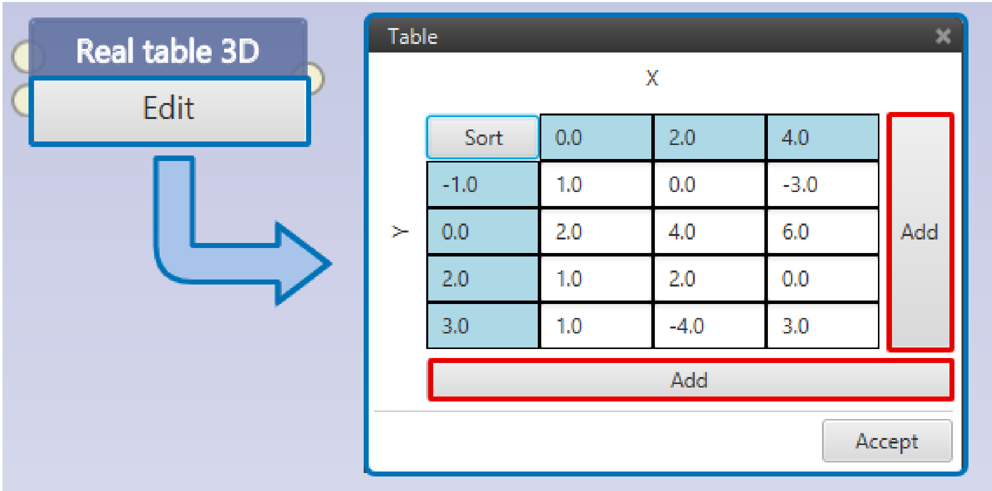
- **Build real array**: when choosing this block, the user needs to define the length of the array. The default value is 3. A block with as many entries as defined will appear. Only one exit dot will be available.
- **Dot product**: this block allows the user to multiply two real arrays of the same length.
- **Real constant**: when choosing this block, a single constant or an array of constants can be defined. The user needs to define the length of the array, which by default is 1 – i.e. a single constant.
- **Save value**: the user must choose a variable or several variables to save the transformed information. The variable type can be *boolean*, *integer* or *real*.
- **Functions: there are 3 kinds of functions depending on the number of input variables required. There are 1x1, 2x1 and N**

  - **1x1 functions:**
    - **Abs**: returns the absolute value, i.e. the non-negative value, of a variable/operation.
    - **Sin**: returns the sine of an angle (in *rad*).
    - **Cos**: returns the cosine of an angle (in *rad*).
    - **Tan**: returns the tangent of an angle (in *rad*).
    - **Arcsin**: inverse function of the sine. It returns an angle (in *rad*).
    - **Arccos**: inverse function of the cosine. It returns an angle (in *rad*).
    - **Arctan**: inverse function of the tangent. It returns an angle (in *rad*).
    - **Sqrt**: returns the square root of a variable/operation.
    - **Exp**: returns the exponential of a variable/operation.
    - **Log**: returns the natural logarithm of a variable/operaton.
    - **Ceil**: returns the least integer greater than or equal to the incoming variable/operation.
    - **Floor**: returns the greatest integer less than or equal to the incoming variable/operation.
    - **Wrap[pi,-pi]**: returns an angle (in *rad*) wrapped between π and $-\pi$.
    - **Wrap[0,2pi]**: returns an angle (in *rad*) wrapped between 0 and 2π.
    - **IIR** digital filter: the output filtered variable depends on the present and past inputs (numerator) as well as the past outputs (denominator). The order n and m of both the numerator and denominator, respectively, is chosen by adding the desired amount of n and m terms – see the Figure below.



IIR Filter Block

- **2x1 functions:**
  - **Add**: returns the sum of two variables.
  - **Substract**: returns the substraction of two variables – the *upper dot* minus the *lower dot*.
  - **Multiply**: returns the multiplication of two variables.
  - **Divide**: returns the division of two variables – the *upper dot* divided by the *lower dot*.
  - **Power**: returns a raised to the b power – where a is the *upper dot* and b the *lower dot*.
  - **Max**: returns the maximum value of two variables.
  - **Min**: returns the minimum value of two variables.
  - **Remainder**: returns the remainder after dividing one variable by another – the remainder of the *upper dot* divided by the *lower dot*.
  - **Atan2**: returns an angle in *rad* following the function $atan2(y, x)$– where y is the *upper dot* and x the *lower dot*.
  - **RTable3D**: returns a variable according to the mapping of two variables – a mapping matrix/table must be defined, where the output variable will be interpolated from.

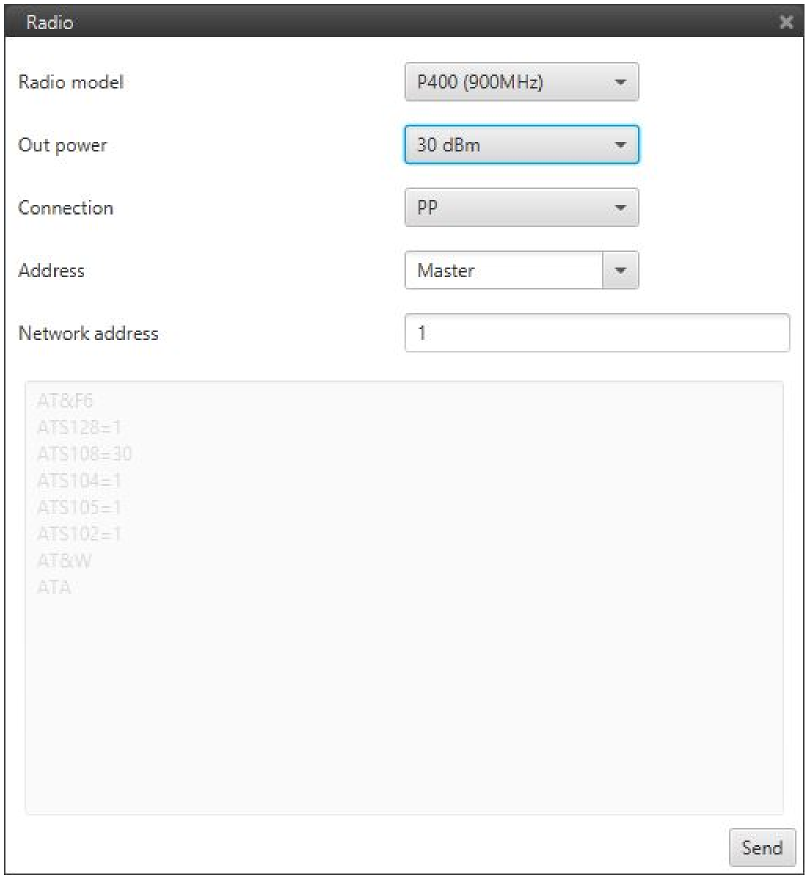


RTable3D Block

- **Nx1 functions:**
  - **Array max**: returns two variables, the maximum value of the array and an integer with the position (starting from 0) of the maximum value on the array. The block has one *input dot* (left side) for the array, and two *output dots* (right side): one for the outcoming maximum value and another for the position (in green – that colour indicates that an integer variable must be used).
  - **Array min**: same principle as the previous block but returning the minimum value.
  - **Array multiply**: multiplies all the components of a real array.

6.2.11 Radio

From this window is possible to modify the configuration of the radio module inside Veronte. By default these are the parameters that are shown in Veronte Pipe.



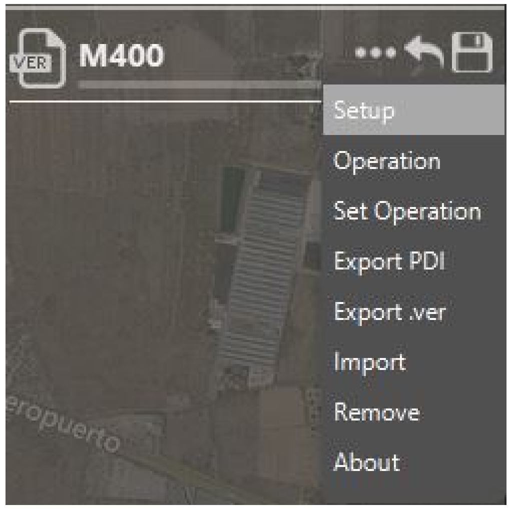
Radio Panel

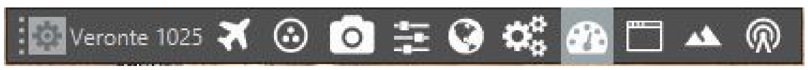
All these parameters have to be set accordingly to the radio module installed in Veronte. Contact Embention if you need the details of your specific radio module.

- **Radio Module:** Each Veronte has only one of the following radio modules P400, P900 and P2400. Select here the radio module installed.
- **Output power:** Sets the available power output.
- **Connection:** Point to Point or Point Multi Point. By default all units are paired with PP connection.
- **Address:** Master or Slave . Select the role for each Veronte by selectig Master or Slave. By default the air unit is defined as Slave and ground units as master.
- **Network address:** the network address is a number that must be equal between Verontes using the same network.

By pressing Send the user sends **all parameters** .

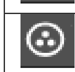
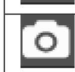
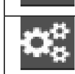
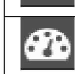
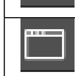
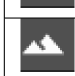
To display Veronte Setup Toolbar, when the autopilot is connected or an offline configuration is opened, in the side panel, click on  and then click Setup. This toolbar allows the user to modify the main features of the Veronte Units.





Setup Toolbar

The different elements of the setup toolbar are detailed in the following table.

Icon	Item	Description
	*Veronte*	Introduce Veronte information.
	*Connections*	Configure I/O connections on Veronte.
	*Devices*	Configure any connected devices: servo, radio, camera…
	*Control*	Introduce control variables and phase configuration.
	*Navigation*	Configure navigation parameters on the system.
	*Automation*	Configure automatic actions on event detection (go home, change phase…).
	*Variables*	Customize variable names and traffic: log, telemetry…
	*Panel*	Configure Veronte Panel layout.
	*HIL*	Configure parameters for Xplane Simulator.
	*Radio*	Configure radio settings.

Each option will be explained in detail in the next sections.

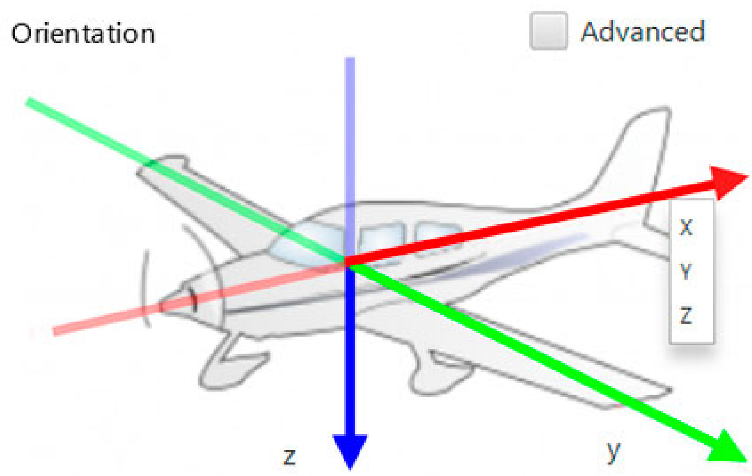
6.3 Installation

This menu allows the user to select the orientation of the autopilot installed.

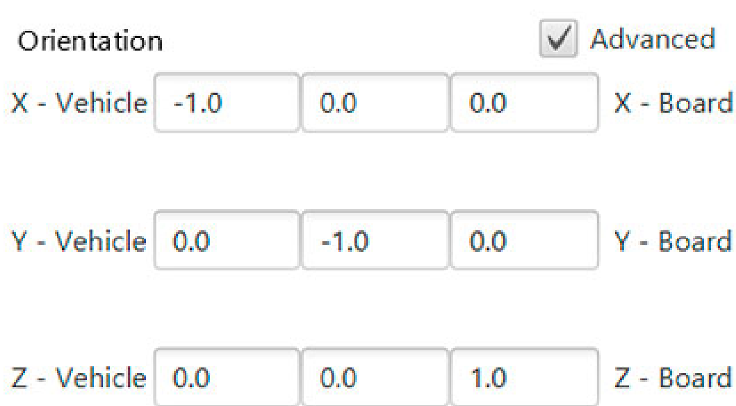
6.3.1 System Position

Aircraft axes are defined according to international aviation convention; Veronte axes are drawn on Veronte box as defined in the *Hardware Installation*.

It is not compulsory to install the autopilot in the platform aligned with the aircraft axes. There could be a rotation, for example, the box can have its X axis directed to one of the wings, like in the case of the figure shown as follows.



Setup Veronte Orientation

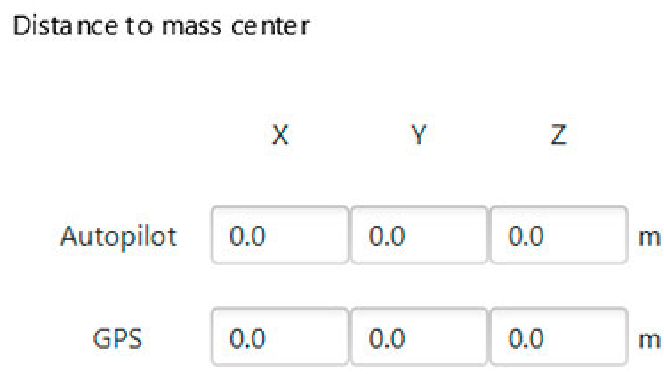


Setup Veronte Orientation Advanced

In order to indicate the box disposition inside the platform, click on the axis in the aircraft diagram and select the corresponding axis according to the autopilot layout.

The case of a non-orthogonal installation is also covered in the software. The right menu of the figure shows the values of the rotation matrix, which are totally modifiable by the user.

Veronte and GPS antenna distance from centre of mass must also be entered. This distance is entered in meters and accordingly to aircraft axes.



Setup Veronte Position

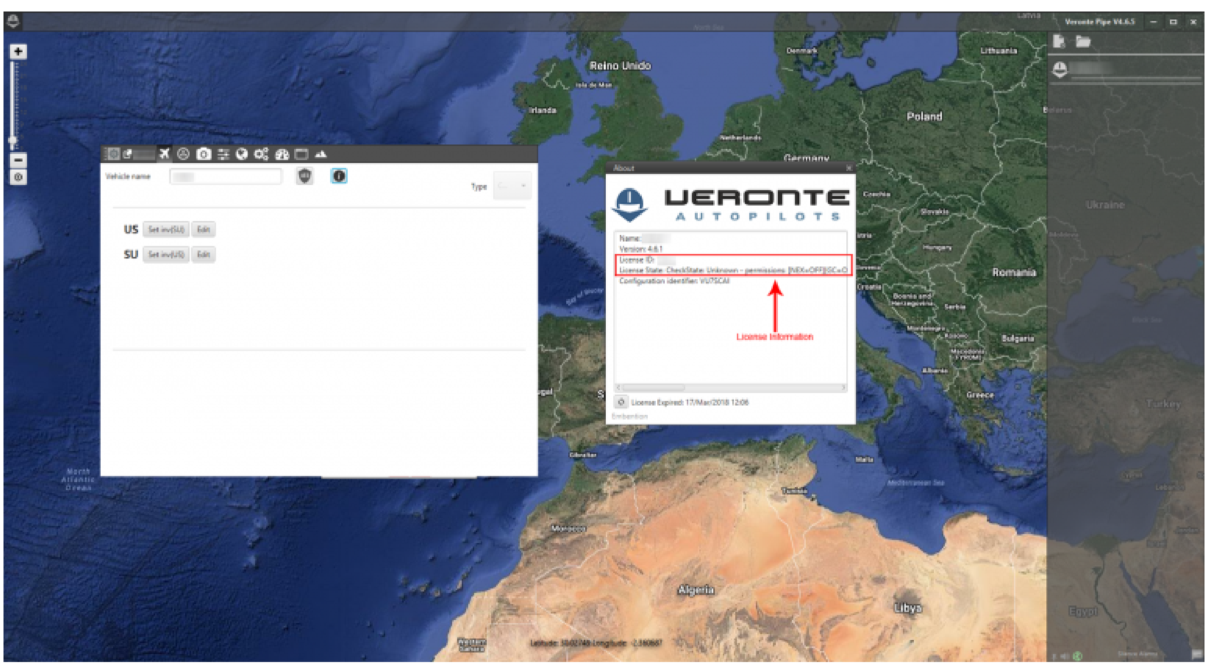
This information is used to take into account the moment produced by the weight of the autopilot and GPS antenna.

Note: Since the latest versions of Pipe, this configuration has to be done in *Devices*-Veronte-Attitud.

6.4 License & Safe Mode

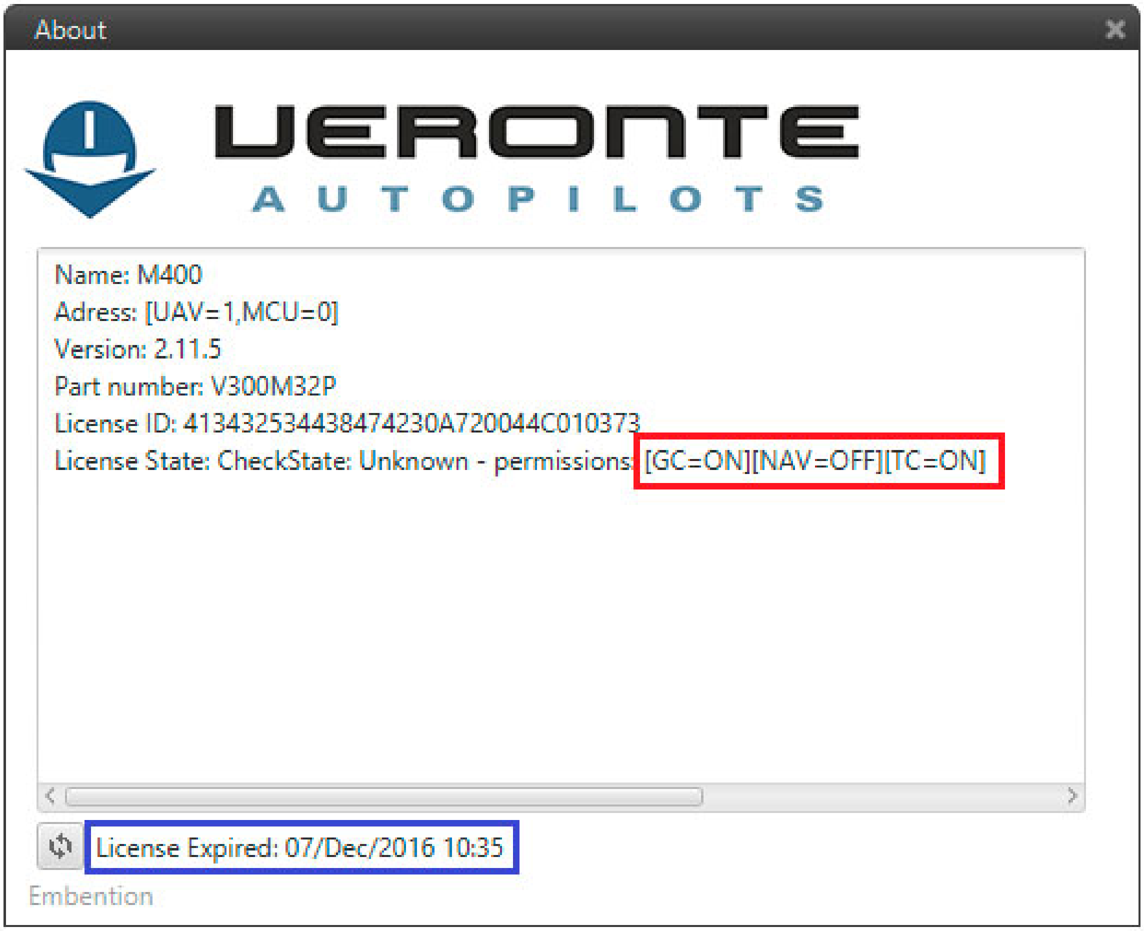
6.4.1 License

To access the License you have to open setup, then select Veronte and click on . In this tab you can check you License of your Veronte. You will have a different License Id for each Veronte (so, License Id Veronte Air is different from License Id Veronte GND). To renovate Veronte License, you must connect Veronte to Internet biweekly.



License

The license panel is useful to check the following information:



License panel

The panel is useful to check the following information:

1. UAV name
2. UAV Address
3. Software Version
4. Autopilot Part Number
5. Software version
6. License ID
7. License State (red)
    - GC: Guidance and Control
    - NAV: Navigation
    - TC: Transmission (RC)
8. License Expiration Date and Time (blue)

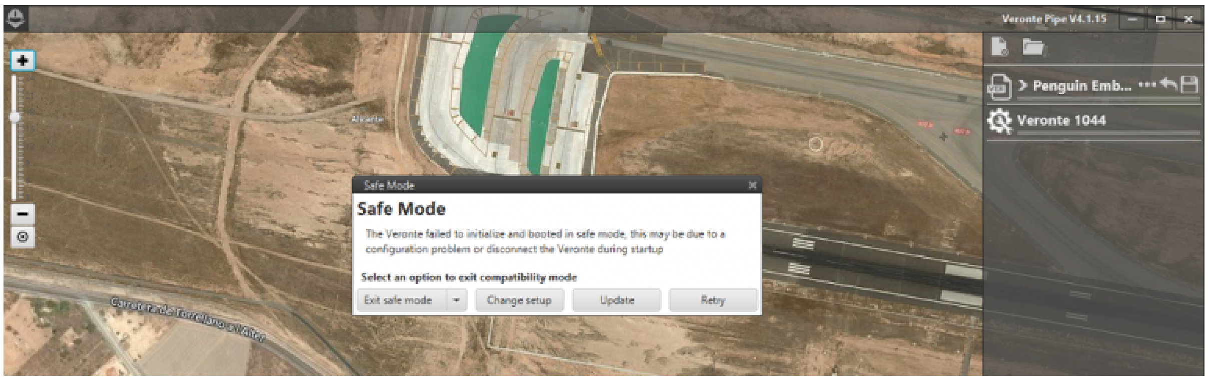
The license state shows the possible action for the selected autopilot (ON=Active, OFF=Not Active). The Ground Autopilot license state for a normal configuration is **ON: OFF: ON**. The Air state is **ON: ON: ON**.

6.4.2 Safe Mode

The safe mode is used to avoid malfunctioning when the autopilot is trying to charge a corrupt file.

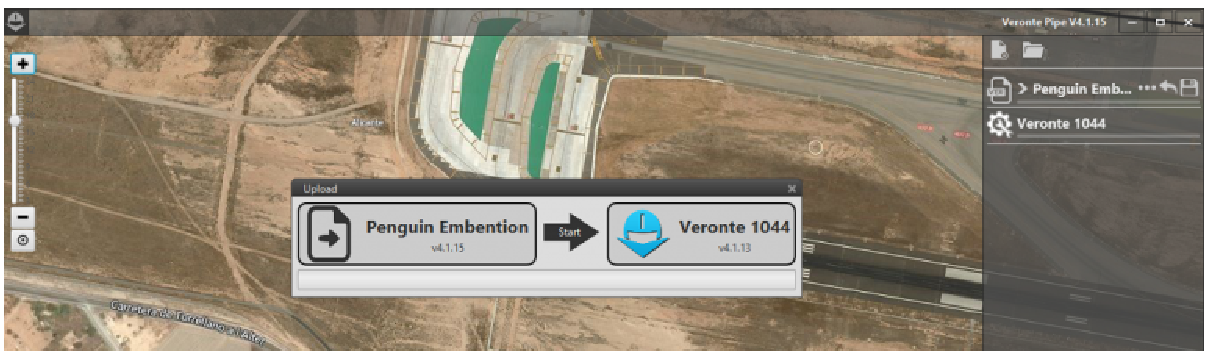
When Veronte is trying to charge a configuration file in its SD card and there is some problem, the safe mode avoids that the autopilot tries to charge the corrupt file again and again entering then in a loop that would not allow communicating with it. Instead of that, the system charges a default configuration stored in its flash memory and enters in what is called the safe mode, allowing the operator to change the configuration and send it again to Veronte.

It is also possible to force the enter in the safe mode by turning the autopilot off and on quickly. Safe mode displays the following window.



Safe Mode

It is possible to **Change setup** and select the configuration file which will be loaded on the autopilot. When a file is selected, a new window will be displayed in Veronte Pipe showing the version and identification of the configuration file and the autopilot where the file is going to be loaded.



Upload Configuration – Safe Mode

With this tool, the configuration file is loaded directly on the autopilot. Now Pipe is only a tool to load the file from the computer to the autopilot and the configuration parameters will not be shown in the software window before being loaded on the autopilot.

The other option available allow the user update the Veronte Unit, this is explained in **Update Onboard Software.**

6.5 Overview

In this section, it is explained how to configure Veronte Units using Veronte Pipe. Each feature available is described in detail in following sub-sections.

Firstly, the user will find how to manage configuration files: import, export, update, etc. Later, this manual will focus on the setup toolbar, this allows the user to change the main settings of the Veronte Units according to the aircraft (Control Loops, Installation, etc.) and mission (Phases, Guidance, etc.) to perform.

Finally, the manual explains how to check Veronte's license, where the user can see the license ID and its status.

6.6 Side Panel Options

The Side Panel allows the user to manage Veronte Units and Configuration files. In this panel each item is displayed, when an autopilot is connected it will appear automatically. The following image shows both items and the options available.

Side Panel Options

The next table gives a brief overview of each option.

Item	Description
Discard	Discard all changes
Save	Save all modified data
Setup	Displays the setup toolbar
Operation	Displays the operation toolbar
Set Operation	Choose one operation to be loaded
Export	Export configuration on Veronte to disk in .ver or PDI files
Import	Import a configuration from disk
About	Displays information on Veronte SN, License and Status (only available for connected Veronte configurations)
Remove	Remove the selected configuration (only available for offline configurations)
HIL	Hardware in the loop simulation
Advanced	Update Veronte Unit and Reboot it

CHAPTER

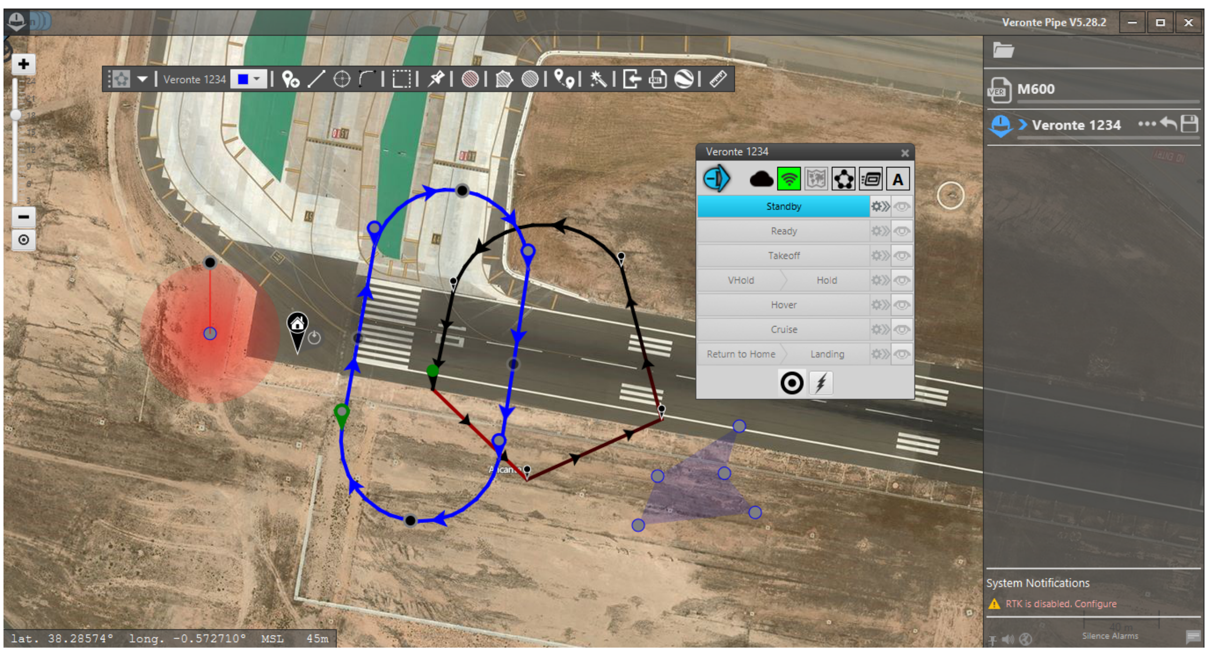
SEVEN

MISSION CONFIGURATION

7.1 Mission Setup

7.1.1 Introducction

A mission can be configured using the Mission Menu. Firstly, Veronte Air has to be selected (it usually appears with the name of the UAV) because **the Mission will be saved in the Air Autopilot**. Then, using the graphical tool of this menu, the trajectory of the aircraft can be drawn, along with other elements to take into account during the mission such as polygons, obstacles or event markers.

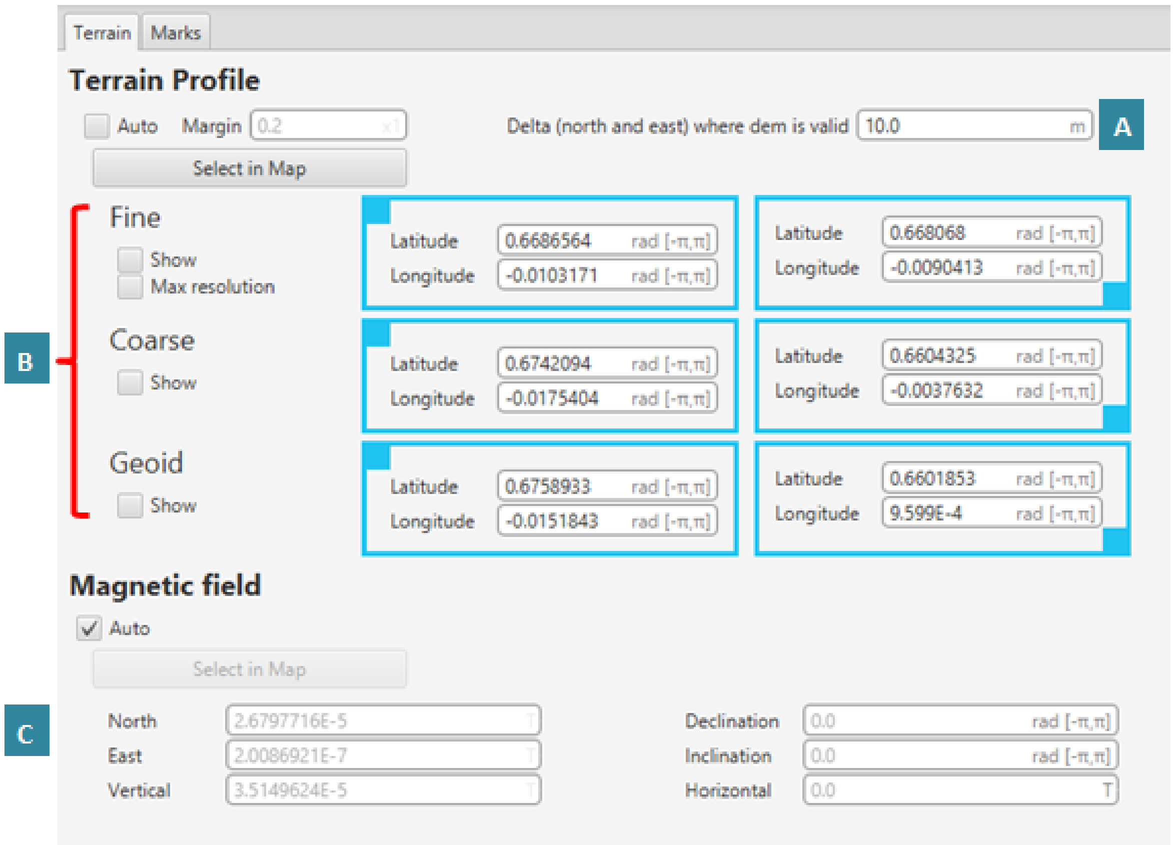


Mission Setup

You can realice two or more missions simultaneously. The path created here will be directly linked to the Cruise guidance mode, so each time the aircraft is in a phase with cruise guidance, it will start to follow the track made in the mission menu.

7.1.2 Terrain profile and Magfield

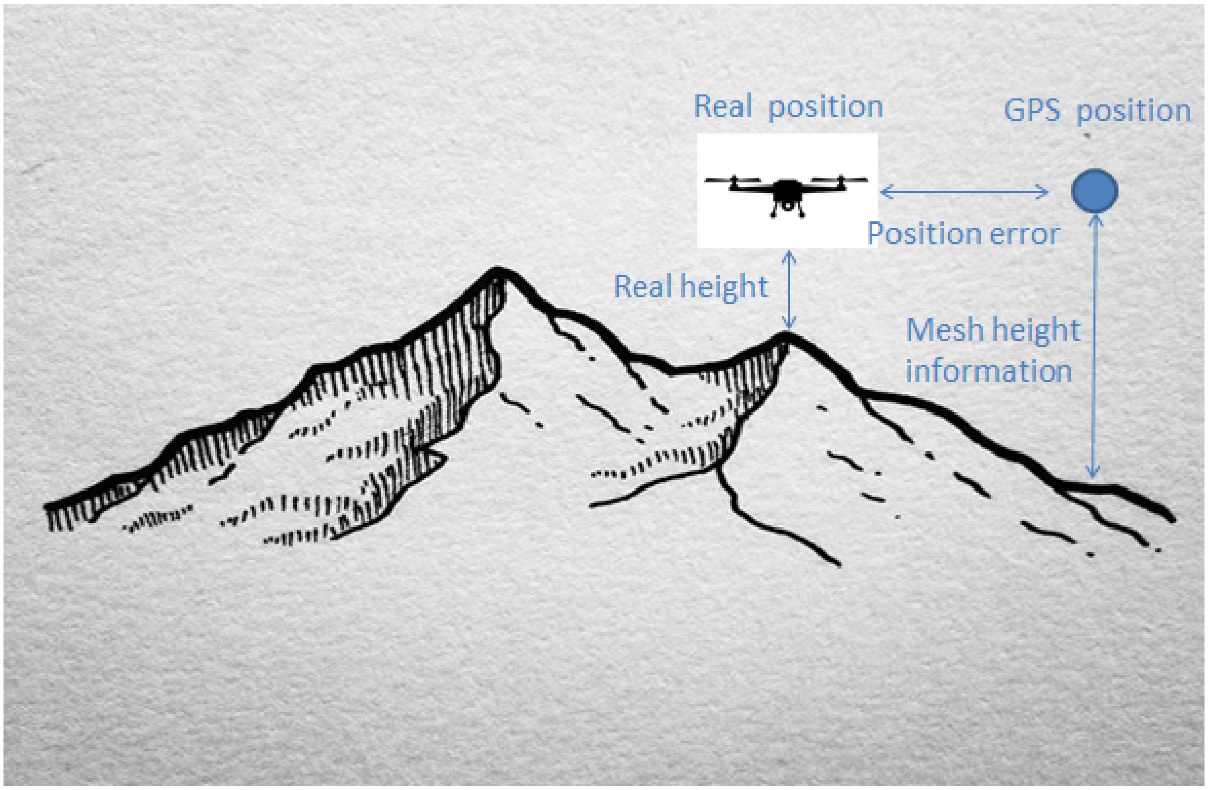
Before to start the configuration of your mission, you have to set the terrain profile and the magnetic field. If you click in  you can access to this settings.



Terrain profile and magnetic field settings

A	Delta parameter
B	Meshes of terrestrial mode
C	Magnetic vector

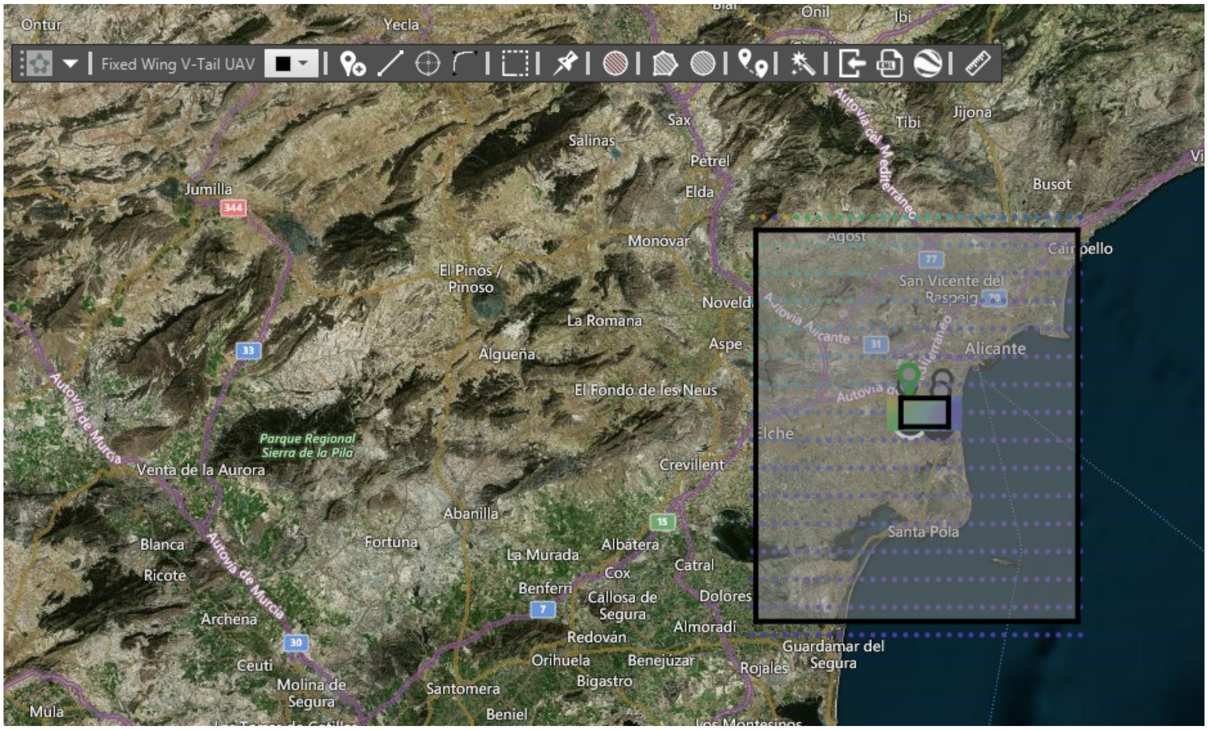
Delta parameter: when you are flying, your aircraft measures the distance to the floor (AGL) through two or more systems: LIDAR, or another similar, and with the data of the meshes. During the flight it is possible that the position error is so large that the measure of the meshes can be wrong. So we define delta parameter to describe the radio of the circunference where we will use both systems, if the position error is bigger than the delta parameter we will use LIDAR alone.



Delta Parameter

Meshes of terrestrial mode: we establish three different meshes (Fine, Coarse and Geoid), for each one we will introduce the coordenates of the upper left corner and the lower right corner or we can selec the zone in the map too.

- *Fine*: Smaller mesh than the coarse one, but with detailed information about the altitude at more points.
- *Coarse*: Big mesh with not much detail.
- *Geoide*: Big mesh which provides the geoid height



Fine and Coarse meshes

The meshes have fixed optimal start dimensions but they can be modified. Dimensions increasing means a lower zone definition so the tool must be changed if really necessary (the greater the resolution the smaller the mesh, because more resolution implies a heavier file to store that data).

If Auto is enabled the system will move automatically the meshes. In order to move them manually and save the updates is important to deselect the Auto tool. The system will display an error if there is not a configured mission on the map.

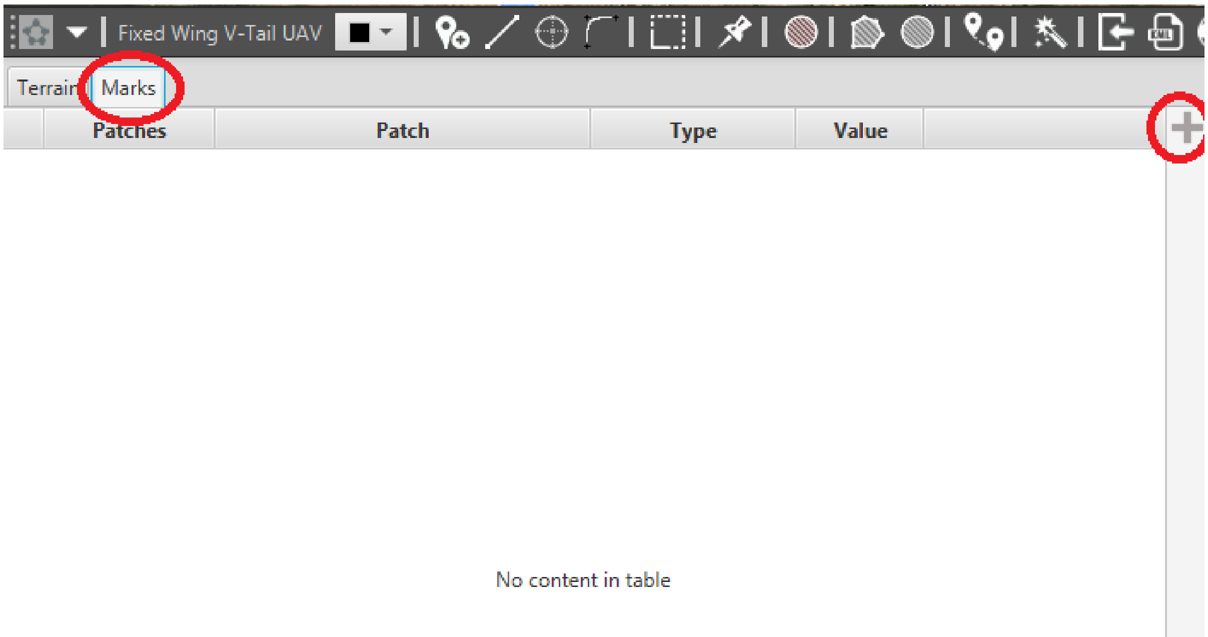
Margin is the percentage at which the system will recalculate the mission if the route is displaced. In other words, if the mission is displaced of the 60% of the area (Coarse or Fine) and the margin is set at 0.8 (the value can be set in percentage by changing the measurement unit) the mission will be not recalculated. If the mission is 81% (or more) away from the previous one the system will recalculate the mission. A low value (or zero) of Margin means more precision of the terrain profile but the system will recalculate more times (or each time) when the mission is modified.

Show option displays the mesh on the screen, allowing the user to move them in order to locate them over the mission area.

Magnetic vector: you can introduce or select in the map the magnetic vector of the zone. As far as the Magfield is concerned, the Auto check has to be marked to take the magnetic declination information of the mission area.

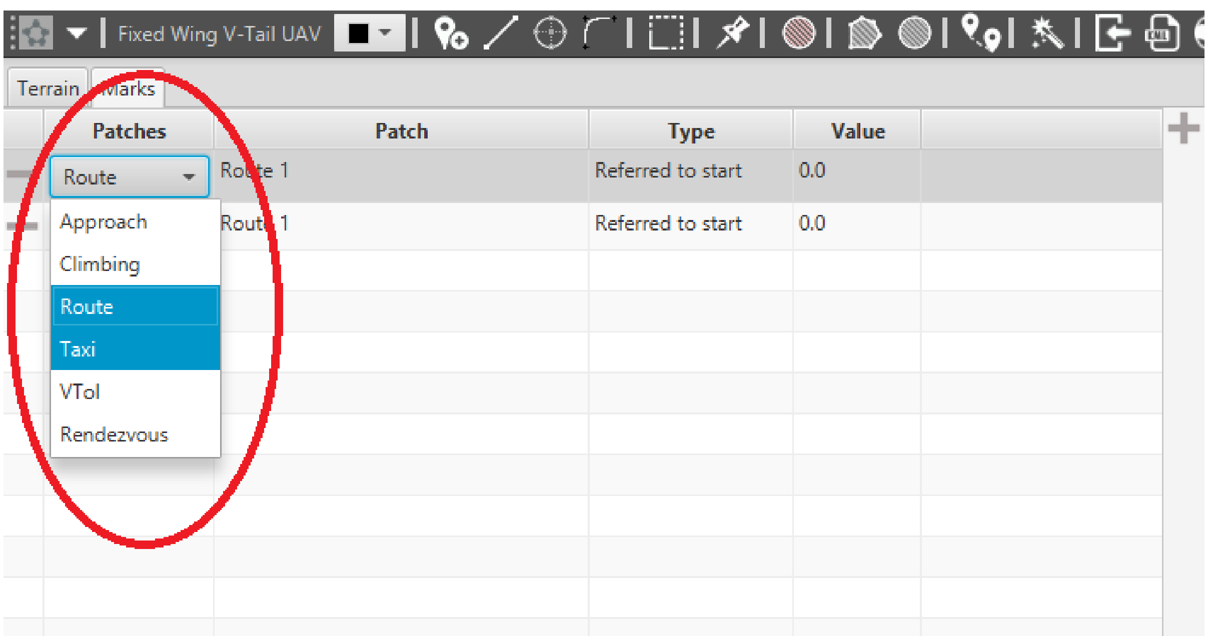
> **Warning:** Before the flight, check that the meshes are over the mission area, especially if carrying out an operation in mountainous terrain.

7.1.3 Marks



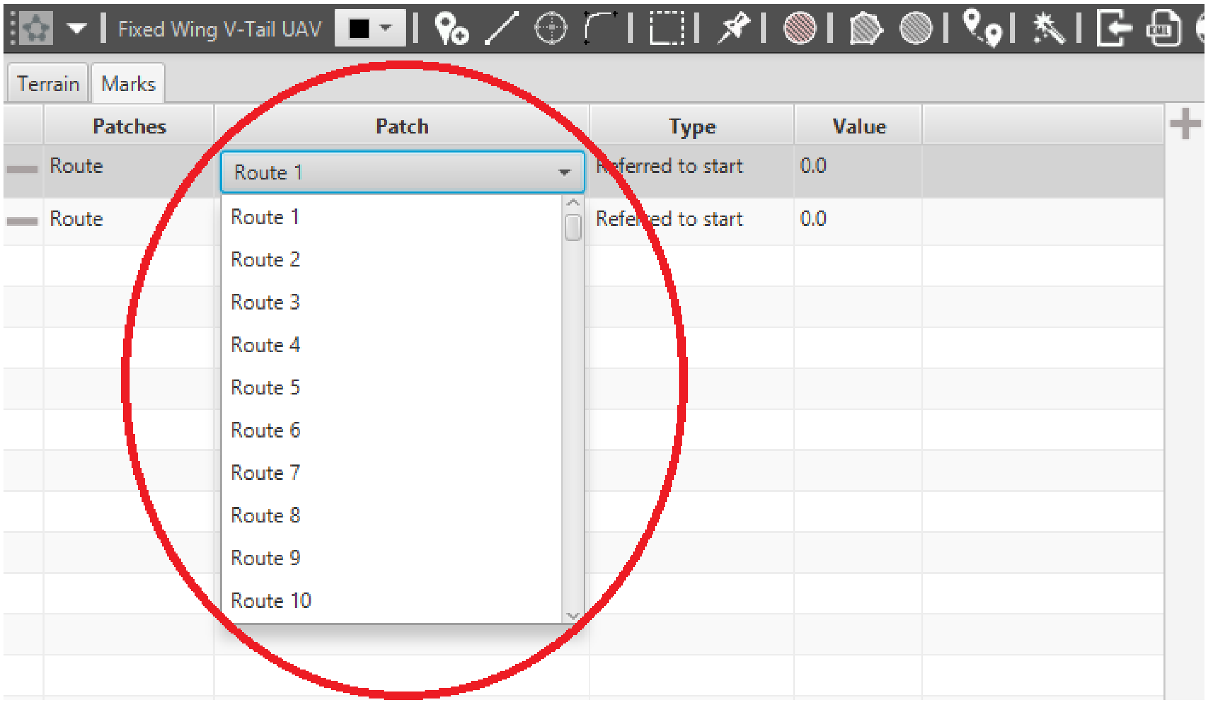
Marks menu

If you click in *Marks* flange you will access to this menu, and with "+" you can add new patches.



Guidance methods

Doing double click in **Route** you can select the guidance method that you want and with double click in **Route n** you will reference the patch to the **Route n** (for example), it is the same procedure for the other methods.



Guidance Method Number

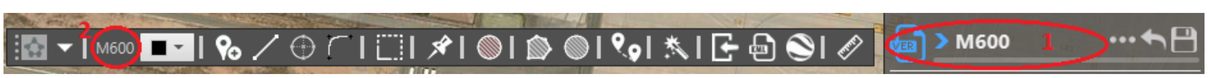
Approach	L0/A1/L2/A3/L4/L5
Climbing	L4/A3/L2/A1
Route	800 waypoints
Taxi	Taxi1/Taxi2
VTol	VTol1/VTol2/VTol3
Rendezvous	Rendezvous1/Rendezvous2/Rendezvous3

This channels are available for each guidance method.

7.2 Mission Tools

This section of the manual contains the information about the different tools available in Veronte Pipe to create the map elements used during the mission. Some of those are: waypoints, lines, circles and so on.

In order to activate the mission tools bar, a Veronte unit has to be select in the right menu.



Veronte Air Selection

Once the autopilot is selected (1), the toolbar will be activated with its corresponding name (2).

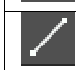
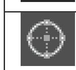
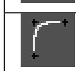
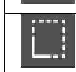
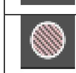
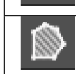
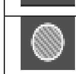
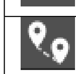
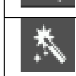
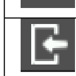
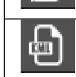
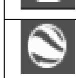
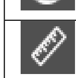
7.3 Mission toolbar

The mission toolbar provides the graphical tools to create the path that the aircraft follows while it is in **cruise phase**. The created mission is stored in the *VER file* of the configuration.

Veronte Setup dialogue can be opened in the main menu clicking on (upper left corner of the screen) and selecting **Mission** in the pull-down menu.

Mission toolbar

The elements that appear on the toolbar are detailed in the following table.

	Open Detail	Displays Terrain and Marks configuration
	Colours	Line colour
	New Waypoint	Add new waypoint on click position
	Segment	Add a straight line
	New Orbit	Add a orbit
	Fly By	Tool to adapt turns in multirotor aircrafts
	Multiple Choice	It is a selection tool
	Event Mark	Add a mark to trigger an event
	Obstacle	Add an obstacle where the plane will not cross
	New Polygon	Draw areas on the map for association with polygon events
	Circular Area	Draw circular areas on the map for association with polygon events
	References	Create points of reference
	Mapping	Draw a polygon for mapping applications.
	Import Route	Import route configuration
	KML	Download a .kml file with the route created.
	Google Hearth	Open Google Earth and show the route that user has drawn
	Ruler	Measure on map

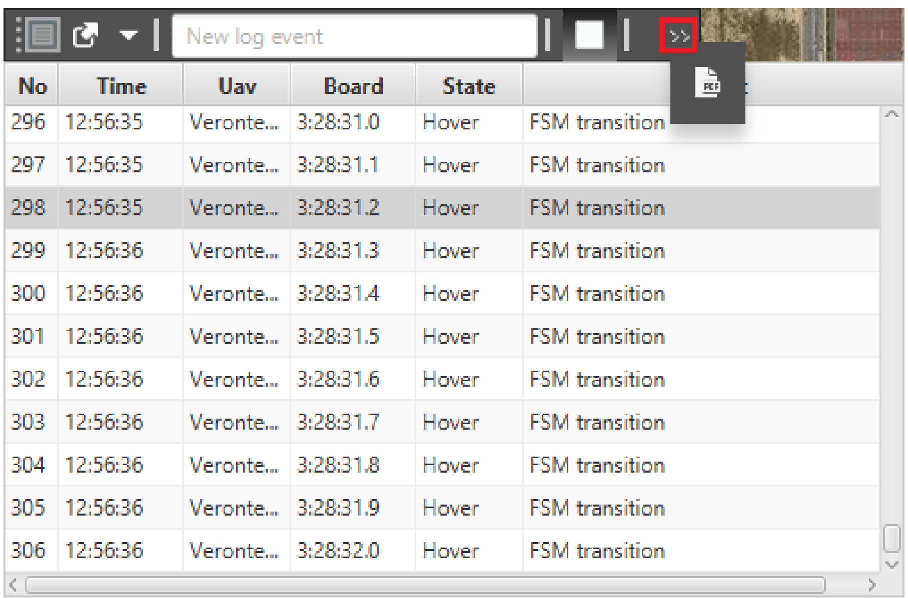
How each one of the tools shown in the previous table works will be detailed in the section *Mission Tools*.

OPERATION

8.1 LOG Record

Log toolbar shows recorded events and permits to introduce custom events to be saved. It can be opened in the main menu clicking on  and selecting Log in the pull-down menu.

In order to create a new log event, introduce event information and press enter to record it on the log.

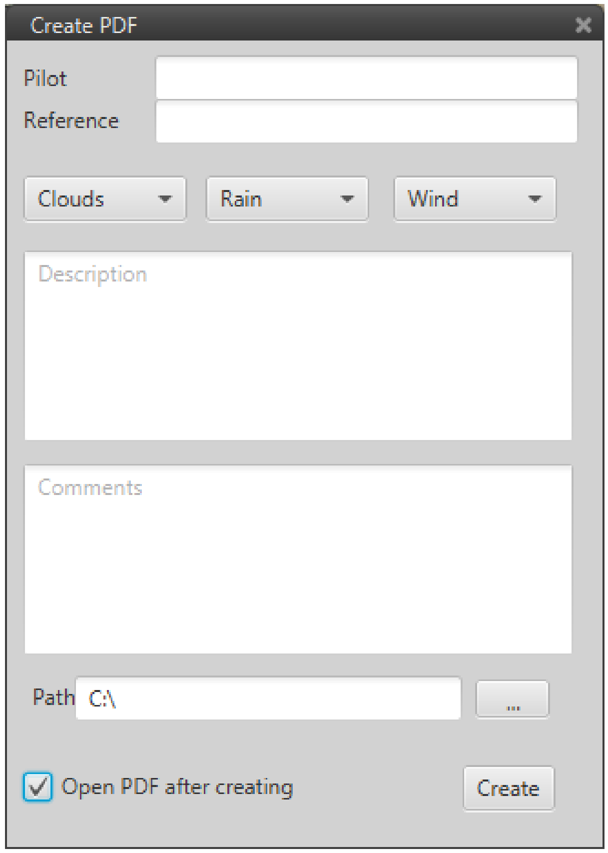


Log Toolbar

- **Record/Stop button:** permits to start/stop capturing log information. By clicking on REC, a new log saving will start.

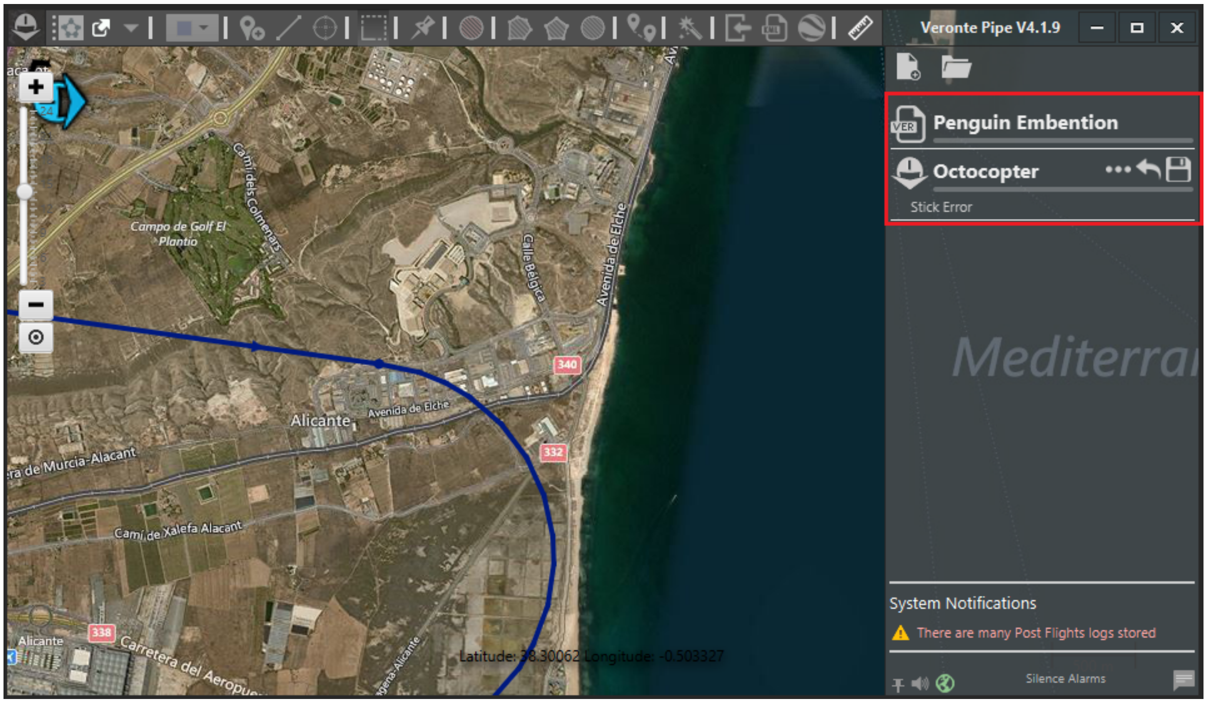
8.1.1 Create PDF

It is possible to generate PDF reports containing saved log information. Click on the Report icon and enter requested information to generate the report.



Report Information

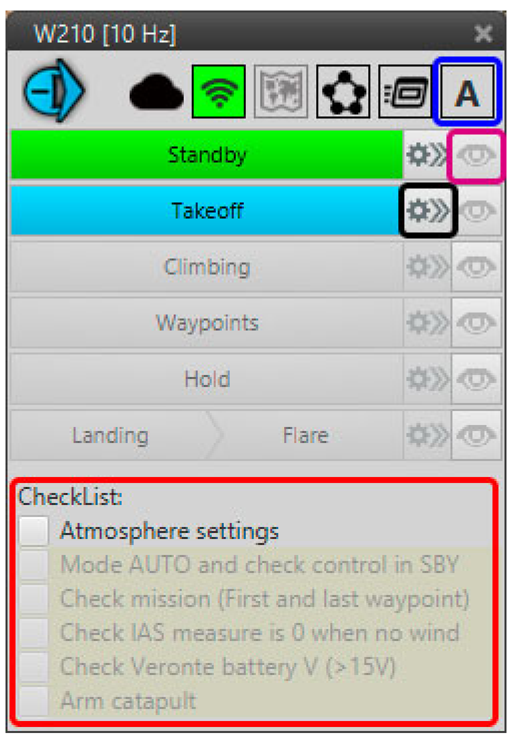
Once both Veronte units, the one on the Ground Station and the one OnBoard, are configured and the mission has been loaded to the aircraft, the system is ready to start the mission. A list with linked Veronte units and configurations are displayed on the side panel. This display shows information and warnings.



Connected Autopilots and Warnings

8.2 Veronte Panel

Double click on any Veronte to display its Veronte Panel. This item is the main interface to control the flight operations.



Veronte Panel

The current phase is marked in green, select one of the available blue phases to change manually. When entering a new phase, all required Checklist elements (red) must be completed.

There are two options when entering a new phase:

- Clicking on the phase name (blue box) will make the aircraft enter the phase with the preconfigured parameters.
- If the **Options button** (black) is selected, the system will enter the phase but a window will appear allowing the user to change the parameters of that phase. Phase parameters can also be configured on the control tab on the setup menu.

The **Mode button** (blue) indicates the actual flight mode (A-Auto, M-Manual…) and allows to change it. Finally, the View button (purple) displays the phase route on the screen.

Dependencies between phases and automatic phase transitions are configured on the automations panel.

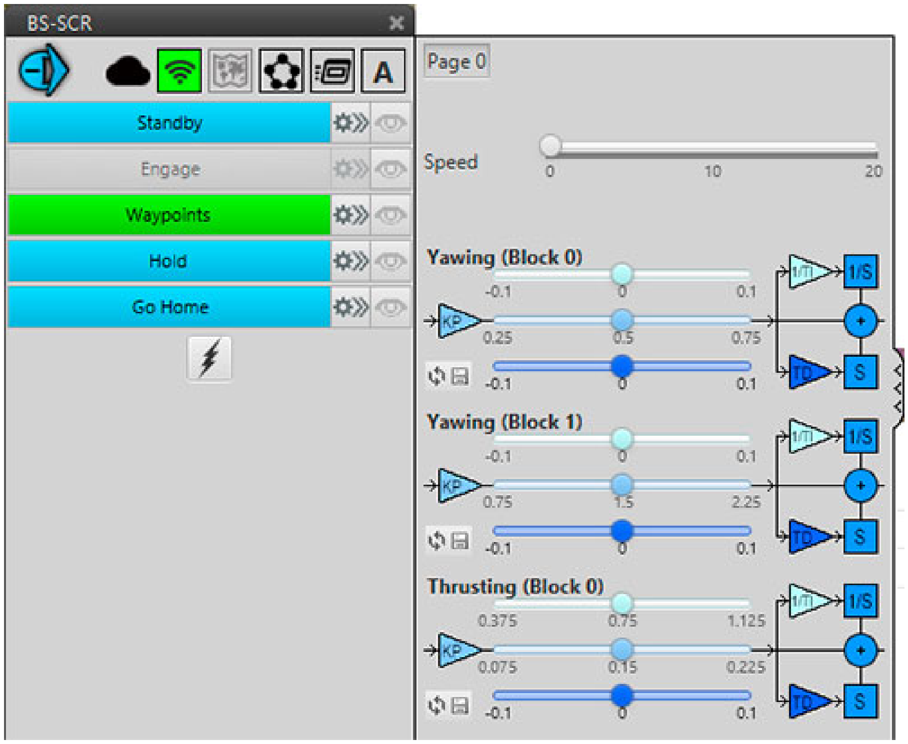
8.3 Flight Options

During a flight operation, the following actions can be performed:

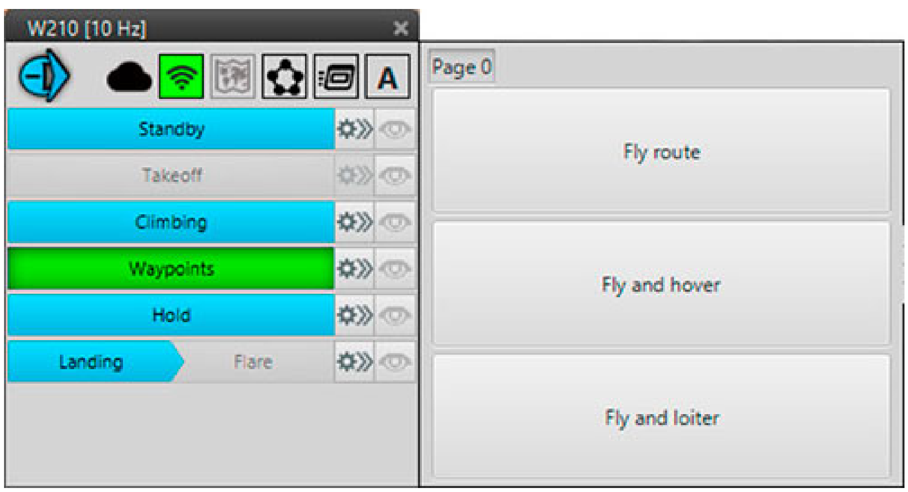
- **Flight monitoring:** Flight data can be monitored in the control station using telemetry displays. Telemetry display configuration can be edited during the flight.
- **Edit mission:** Mission can be edited prior or during the flight. If the mission is modified during the flight, it will not be recalculated until the new mission will be saved. The aircraft behavior will not change and it will follow the last saved mission.
- **Change phase:** Phases permit to set the vehicle configuration to a specific performance. Click on a phase to initiate it.
- **Activate Manual/Assisted/Automatic modes:** As shown in the previous figure of Veronte Panel is used to change between the different flight modes (which were created in Modes, see section *Modes*). This change can also be made with an automation or directly from the joystick (an example of this automation is describen in Automation Examples section).
- **Abort mission:** "Go Home" button can be configured in the automations panel and shown in the lower part of Veronte Panel (see Automation Examples).

8.4 In-flight Settings

During the operation it is possible to adjust the controller parameters and some guidance commands (desired roll, pitch, IAS and so on) directly on Setup – Control, or using the TC Panel, which is a shortcut that shows all this values directly on the Veronte Panel. The configuration of this option is detailed in *TC Panel*.



Controller adjust tools



Guidance tools

8.5 In-flight mission edition

Once the aircraft is flying the mission created a priori, it is still possible to change the route. While the platform is in the air following the waypoints, entering in the mission menu will allow the user to modify those waypoints, routes, loiters and so on "in-flight". Those modifications won't produce any change until the mission is saved. So saving the variations over the mission will make the vehicle to follow now the new route created.

CHAPTER

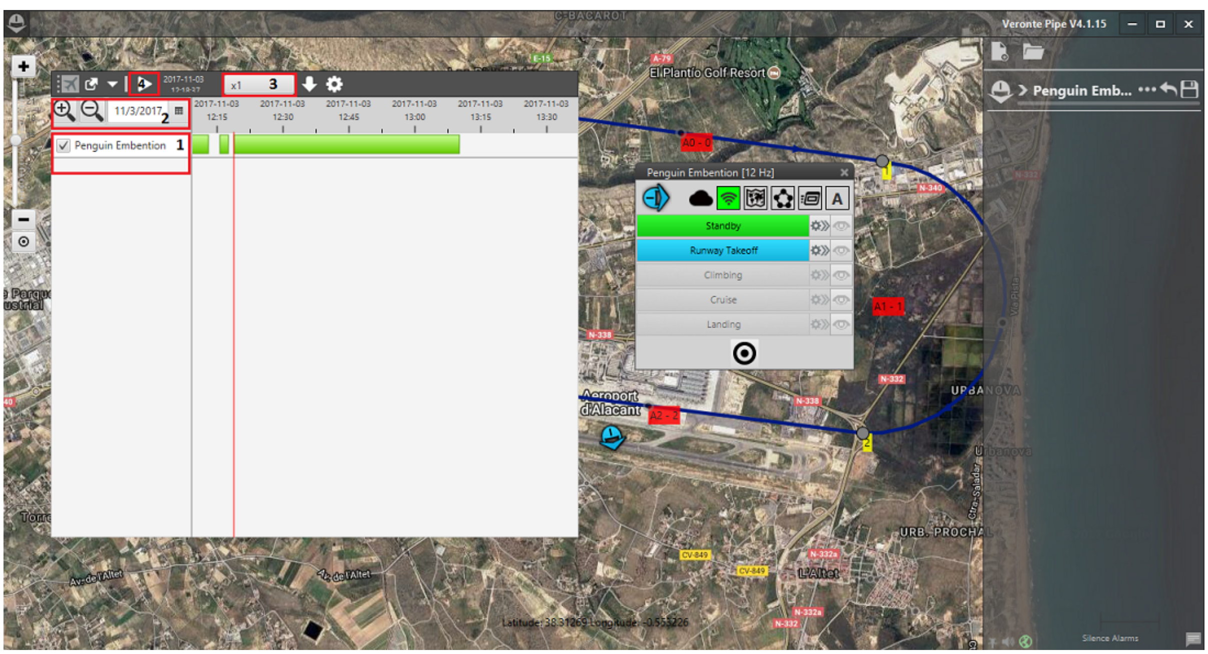
NINE

POST FLIGHT

The post flight menu allows the user to check all the information stored during a flight operation. There exist different options to display the telemetry of a certain operation, and they are presented in this section.

To access the menu, click on  (Main Menu), and then select Post Flight in the pull-down menu.

9.1 Tour Play



Post Flight Panel

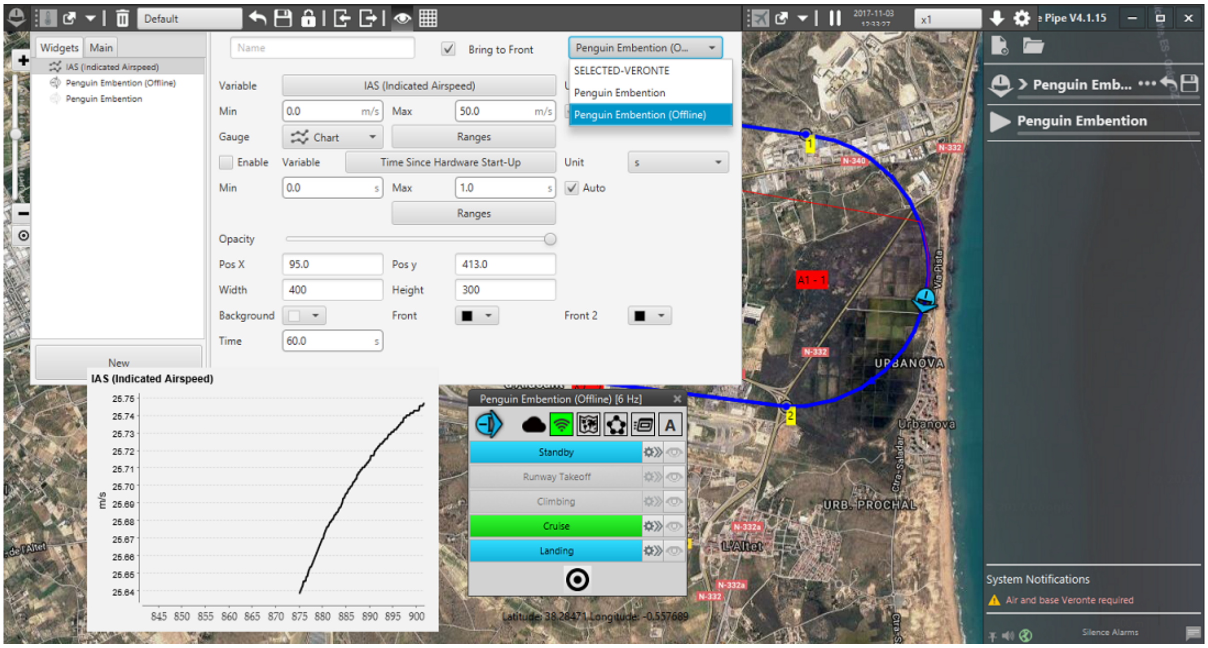
When opening the post flight menu, there are a series of options to configure in order to display the tour:

1. The autopilot, whose information wants to be played on the tour, has to be selected among the ones that appear on the list on the left side.
2. The user here can select the date of the flight that is going to be replayed.
3. The speed of the tour can be configured in this option (x1 means normal speed).

4. The play button starts the tour.

When clicking the play button, a new autopilot will appear in the right sidebar with the name of the unit plus Offline. As it can be seen in the previous figure, a new Veronte Panel appears on the screen showing the phase changes as they occurred in the real operation.

Regarding the telemetry, all the variables that were in the Telemetry Link during the real flight can be displayed in the tour.



Workspace Panel

For example, to display the IAS that the aircraft had during the flight played, just select the offline autopilot in the pull-down menu, and the gauge will show the indicated airspeed of that operation.

Warning: If a variable was not in the telemetry link during a flight, it can not be later shown in the tour of that flight.

9.2 Data Export

The telemetry generated during an operation is stored in different ways (see section *Telemetry*). According to how each one of the telemetry variables has been stored (data link, onboard log, user log and fast log) it has to be downloaded in a different way.

9.2.1 Data link

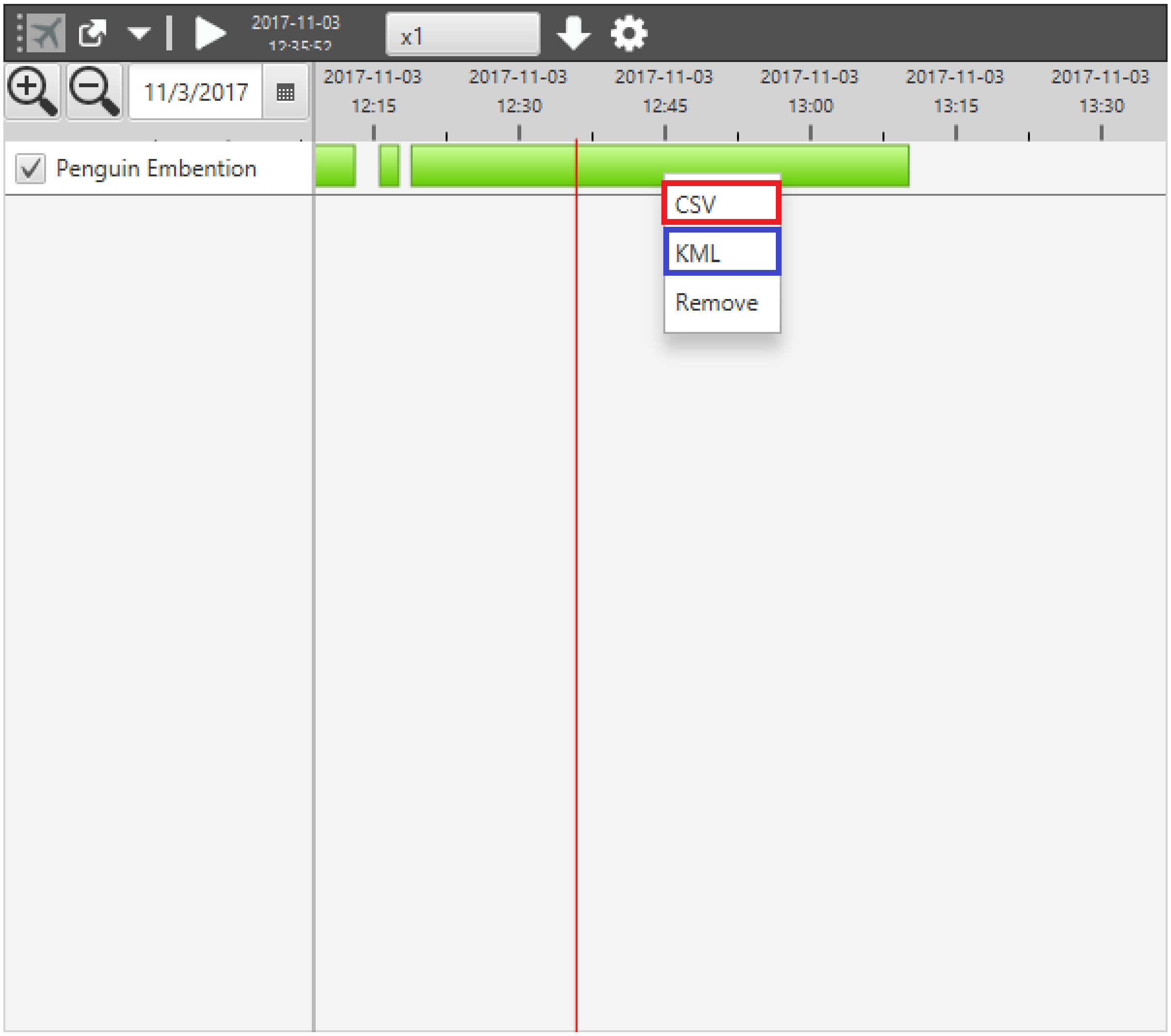
The information obtained directly from the **Data Link** is saved at the "appdata" folder of the operative system. This data is the telemetry sent by the air autopilot to Veronte Pipe, so it is stored in the computer that has the software installed. The exact path (in version 4.X) to obtain it is:

C:UsersName_of_the_userAppDataRoamingVerontePipe4.Xsession

In the folder that appears in this route, there are several folders each one corresponding to one operation. To identify the desired one check the date on the folder name. In order to play a tour of a flight in a different computer, just copy the folder to the same route in the other computer and the information will appear in the post flight menu as was explained in the previous section.

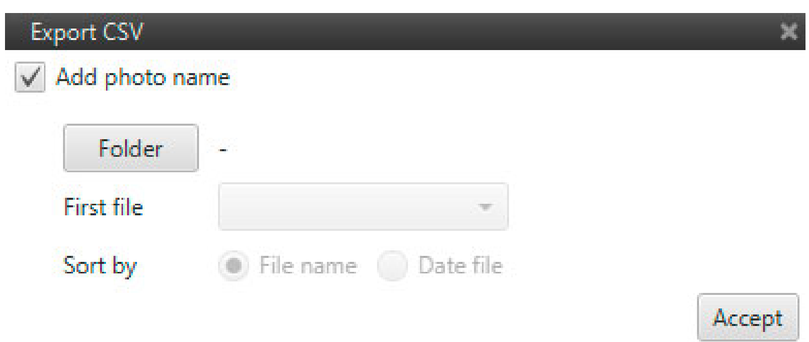
It is also possible to download the information of the Data Link for post-processing purposes. Right-clicking on the green bar of the desired flight will open a menu where it is possible to download that information in:

- **CSV file** – This file uses ";" for data separation and "," for decimal indication.
- **KLM file** – It can be opened using Google Earth and allows to check the aircraft route in 3D.



Post-flight data export

When downloading a CSV file, Pipe provides the user with an option to automatically name the photos taken during a photogrammetry mission.

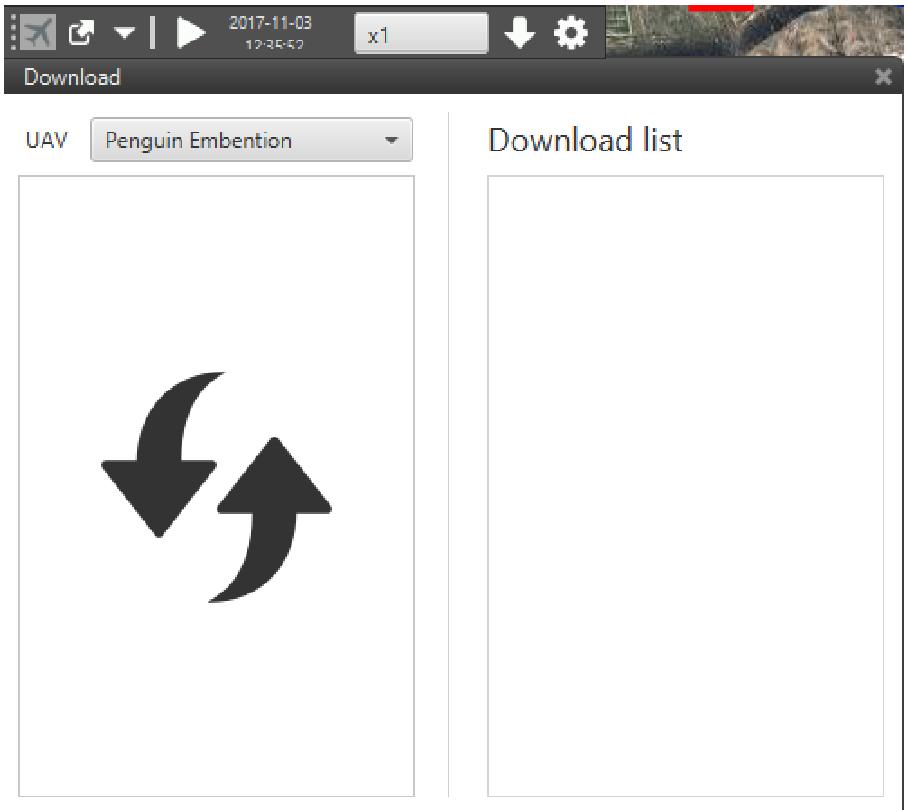


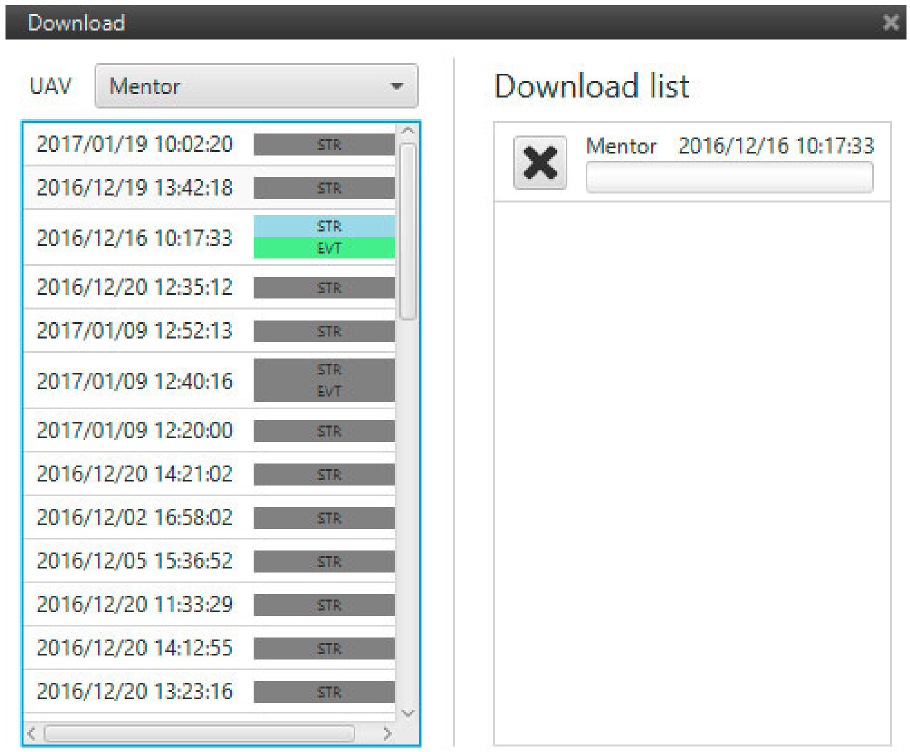
Export CSV

The button Folder allows the selection of the folder that contains the photos taken during the mission. Then, in First file will appear all the files of that folder and the user has to select the file that corresponds to the first photo. Finally, Pipe permits to sort the photo files by name or date.

9.2.2 LOG

The data of the other three type of logs is stored in the SD Card of the air autopilot, so it has to be downloaded to the computer in order to obtain it. To do this, it's sufficient to open Postflight bar and click on the down-directed arrow, then select the Veronte unit for data downloading and refresh the page in order to choose the flight files to be download (checking the flight date and time).

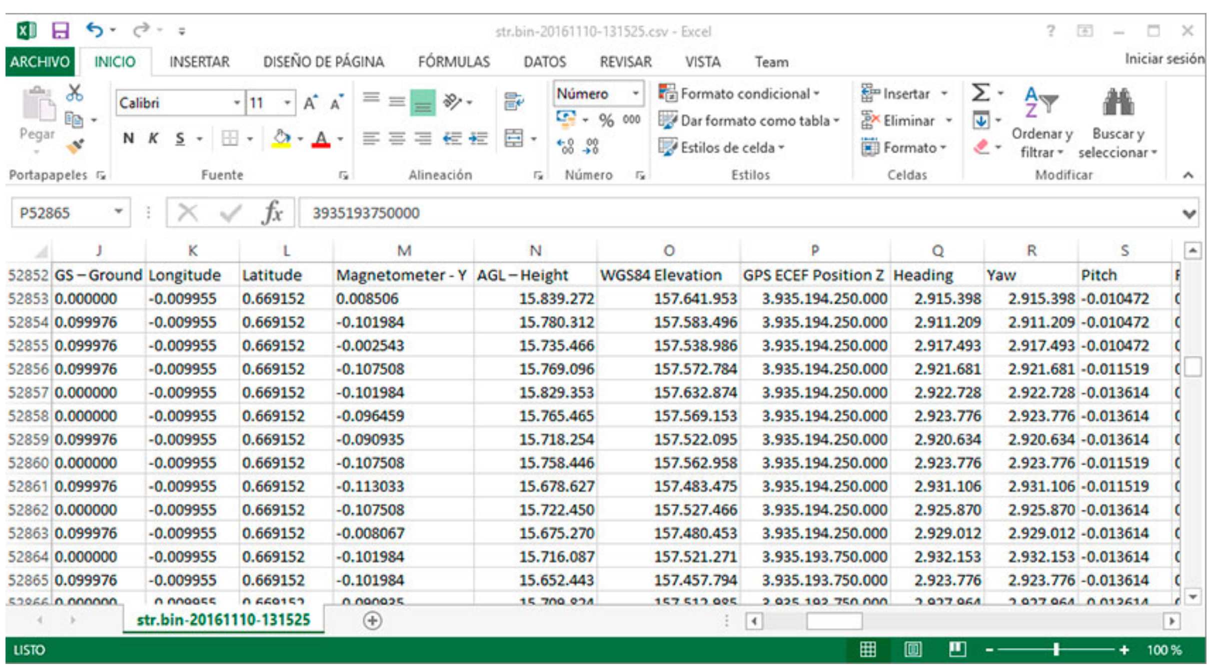




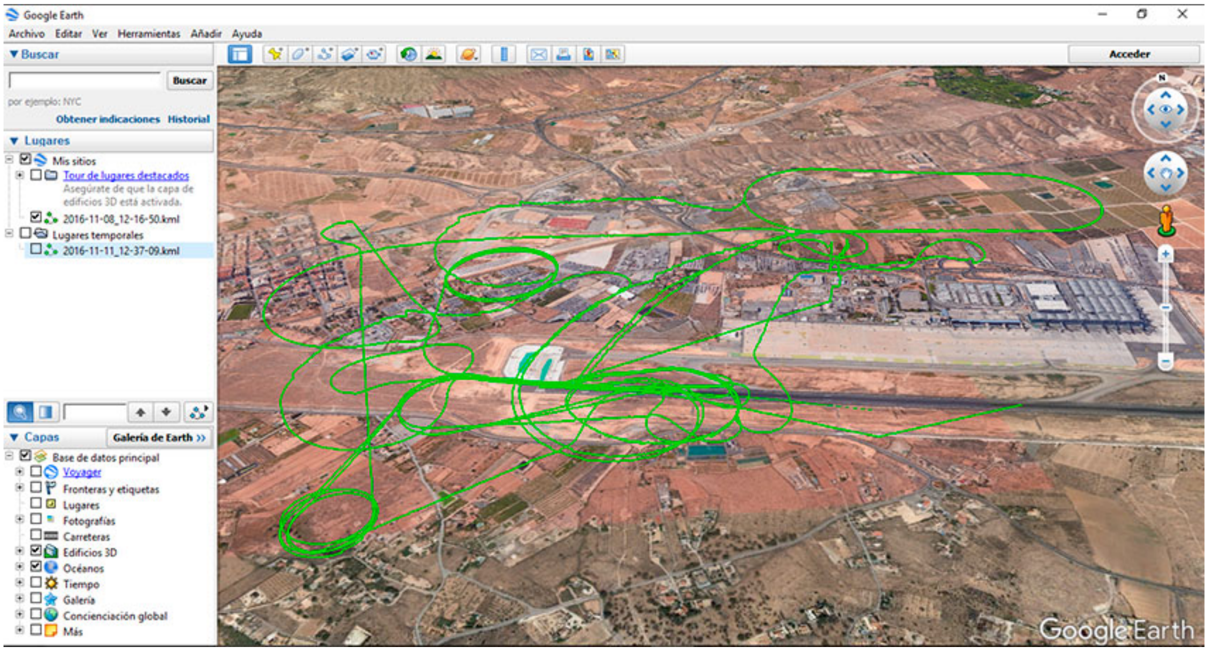
Data Export

The bar STR corresponds with the information of the Onboard Log and EVT is for the User Log. Once the file has been downloaded, the bar will turn green and right-clicking over it (SRT or EVT) will allow the user to download a CSV file with that information.

The following two images show a .CSV and a .KML files opened respectively with Microsoft Excel and Google Earth.



Post flight in Microsoft Excel (.csv)



Post flight in Google Earth (.kml)

CHAPTER

TEN

SIMULATION

10.1 Professional HIL

10.1.1 Mounting

10.1.1.1 Cable Connection

Package content includes the following cable. One connector must be connected to Veronte Autopilot and the other one to the CS Cable or the Aircraft connector in case of HIL Simulation using the complete platform system.

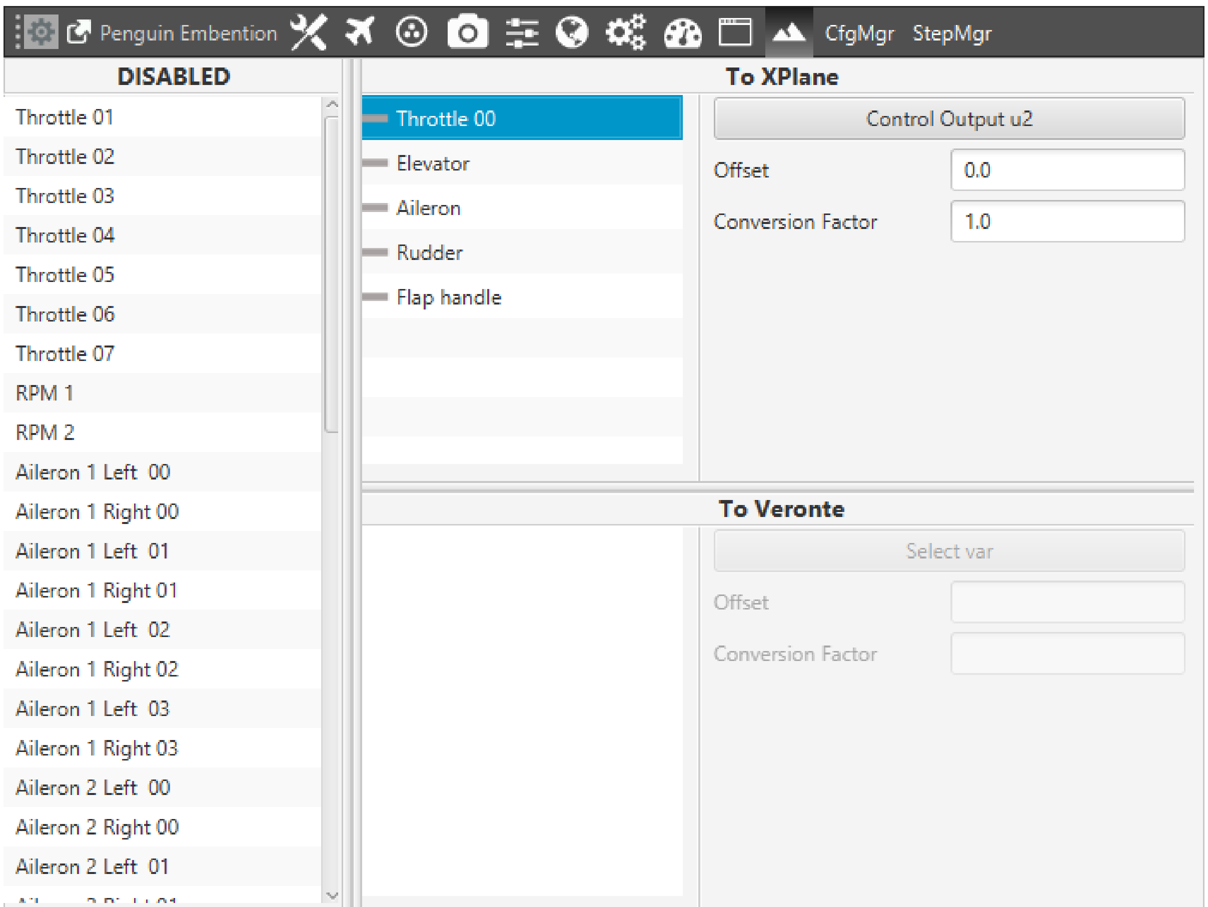
HIL Simulator Cable

10.1.1.2 Veronte Pipe Configuration

There are 2 configuration items on Veronte Pipe, one relating communications between the application and the X-Plane simulator and another one refereeing the autopilot configuration.

10.1.1.2.1 Autopilot Configuration

HIL simulation tab is available within Veronte Autopilot setup toolbar. The user can link the variables on Veronte Autopilot with the corresponding ones in X-Plane simulator.



Veronte Pipe – HIL Setup

In this panel, X-Plane variables are available on the left side (Disables). In addition, it can be seen two section more To XPlane and To Veronte.

In order to configure the simulation variables, users have to:

1. Enable the ones that have been configured in the aircraft model (Plane Marker). Just drag and drop them into To Xplane section.

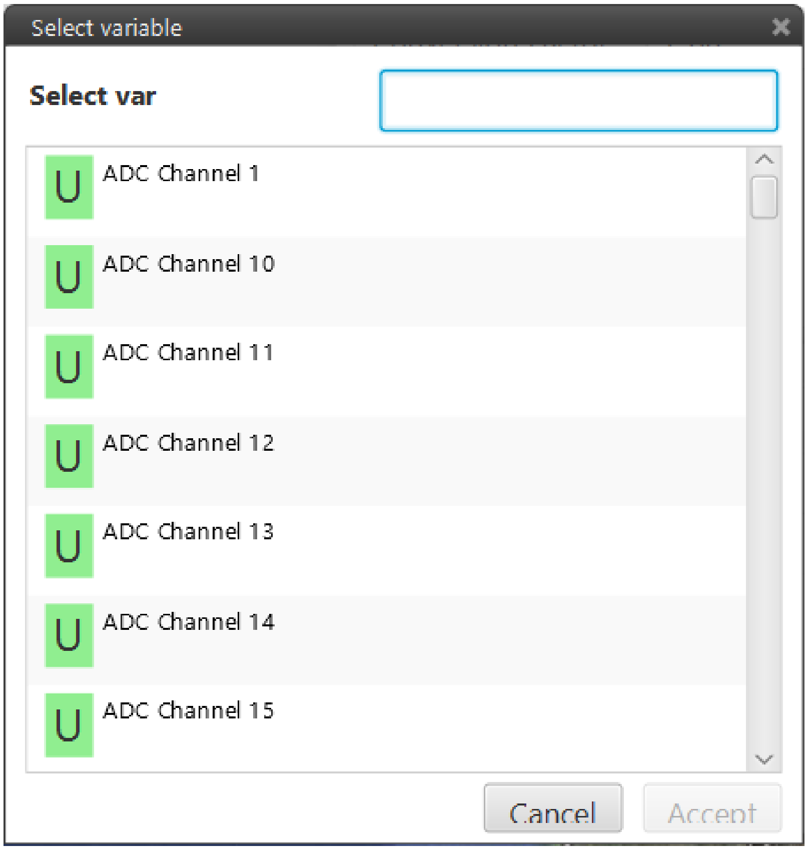
> **Warning:** Always make sure that surfaces are moving in the right direction and with the correct deflection angle.

To avoid mistakes is possible to set a positive fixed deflection (Control Tab) in Standby phase for all surfaces and control surface deflections in the X-Plane model.

Surface control variables are of two types:

- Radians measure (variables with numbers)
- Degrees measure (variables with no numbers)

2. Once X-Plane variables have been enabled, select the actuator variable (Control Output) that matches with the ones in Veronte autopilot. A new window will be displayed for each variable.



3. Set a Conversion Factor or Offset, if it is necessary. Conversion factor multiplies the Veronte output signal and can be used in case units on Veronte and the X-Plane simulator do not match (Surfaces in X-Plane move normally in a [0,1] range). The following operation allows to converting an angle [deg] measure to the X-Plane form:

$$(\text{Angle} + \text{Offset})\ \text{Conv.Factor} = \ldots$$

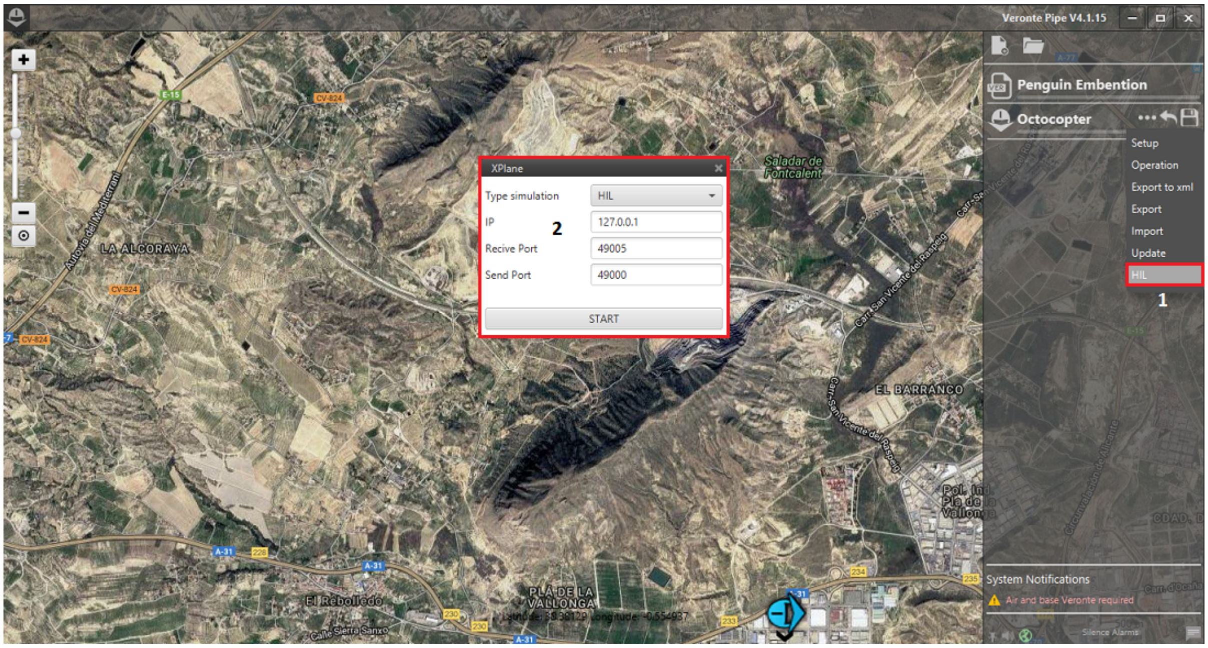
When Angle and Offset are measured in [rad] and the Conversion factor is a constant (normally it can be calculated as 1/(deflection angle in [rad]).

In the case of [rad] measures, the Conversion factor must be set in 57,29578 ([deg]-[rad] conv. factor).

Finally, it is necessary to configure the communication between Veronte Pipe and X-Plane, see section below.

10.1.1.2.2 Veronte Pipe Communications

In order to start the simulation, select the HIL option (**1**) on the Veronte Unit, this is available on the side menu . Popup screen will be displayed (**2**) for selecting the kind of simulation and configuring the parameters.



Veronte Pipe – HIL Communications

Changing the default values is only recommended for advanced users. Press start in order to start the data transfer between X-Plane and Veronte Autopilot.

> **Warning:** The simulation must be started when the aircraft is in the Initial phase (the one that gets in once is powered). In this phase the X-Plane will simulate the GPS signal to locate the autopilot in the place indicated by the airport on X-Plane.

10.1.2 X-Plane 10 Settings

10.1.2.1 X-Plane 10 Configuration

X-Plane 10 demo version can be downloaded from this link.

X-Plane 10 communications settings must be edited in order to have communication with Veronte system. Follow the next steps in order to make a proper configuration.

For low-performance computers, it may be needed to reduce the graphics quality on the simulator, as described below.

10.1.2.1.1 Aircraft Model Installation

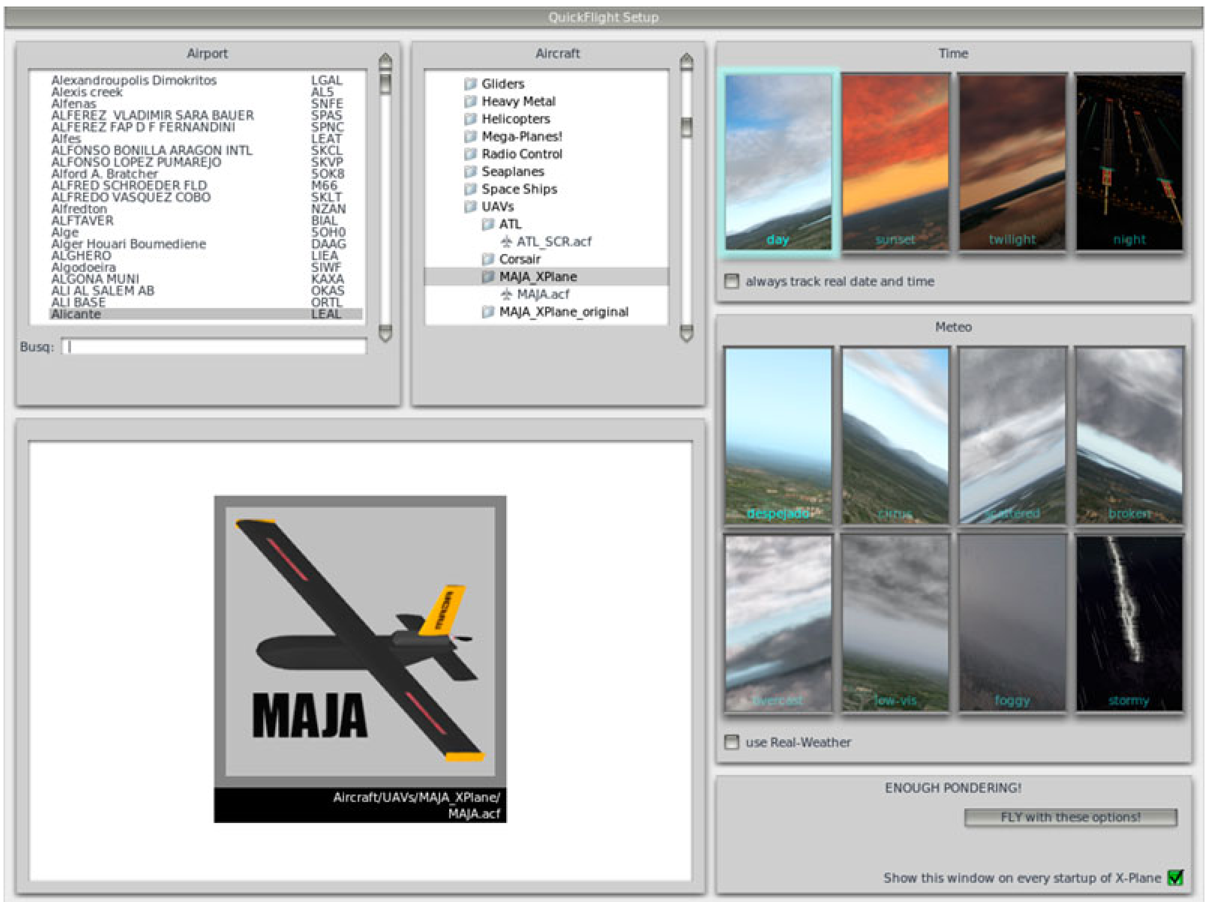
X-Plane 10 simulator is compatible with a wide variety of platforms: airplane, helicopter, multicopter, surface vehicle. . . . In order to create the platform model, Plane Maker tool provided by X-Plane 10 must be used.

Once the aircraft model has been created, it can be integrated on the X-Plane 10 simulator by following next steps:

- Copy the model folder to the "Aircraft" folder within the X-Plane 10 installation directory.
- Copy the content in the "Airfoils" folder, available on the aircraft model folder, to the "Airfoils" directory within the X-Plane 10 installation directory.

10.1.2.1.2 X-Plane 10 Setup

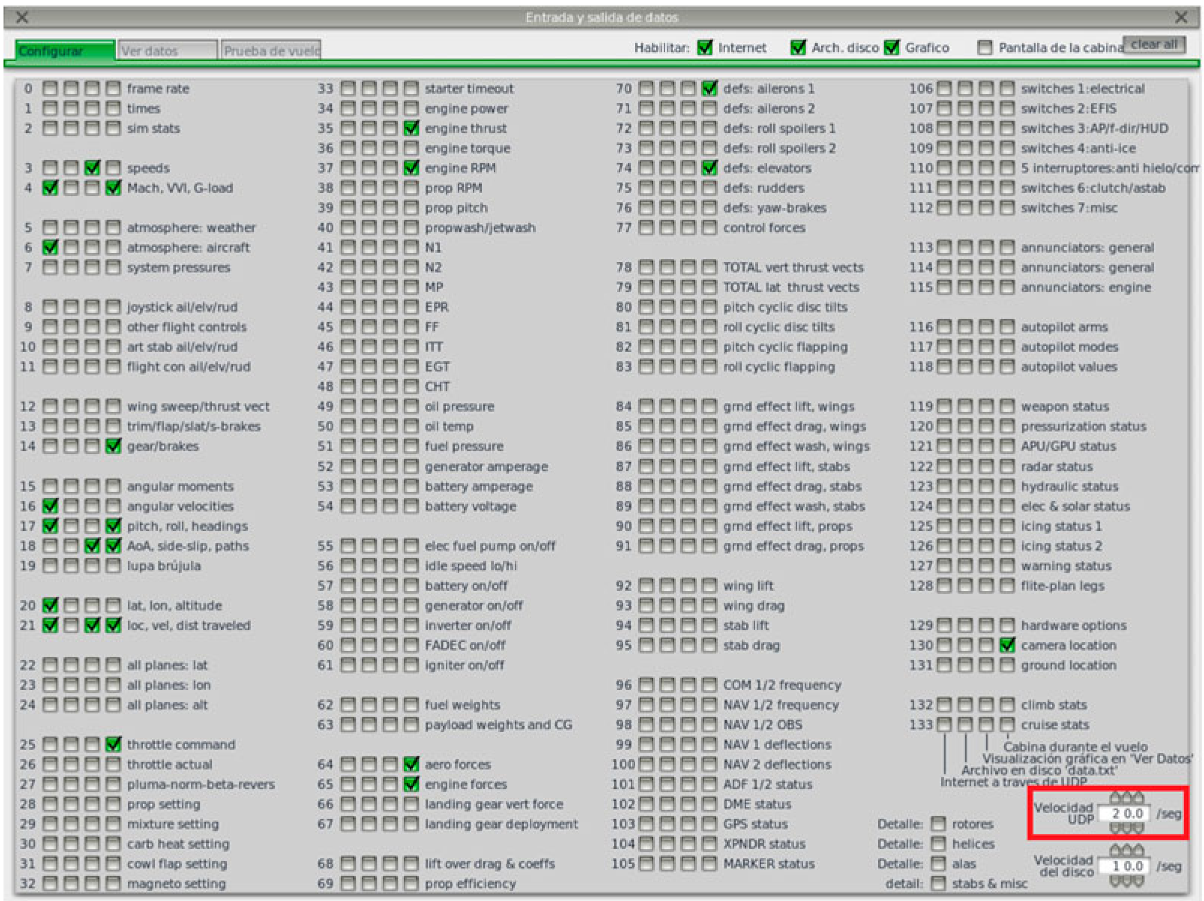
On X-Plane 10 execution, **Quick Flight Setup** window will be displayed; select which aircraft to use, the starting airport and weather conditions to be simulated during the flight.



Quick Flight Setup

Data transmission settings must be edited on the settings tab. Select the input and output data option and edit the UDP speed. This speed must be set to **50/s**.

X-Plane 10 Setup



X-Plane 10 Input and Output Data

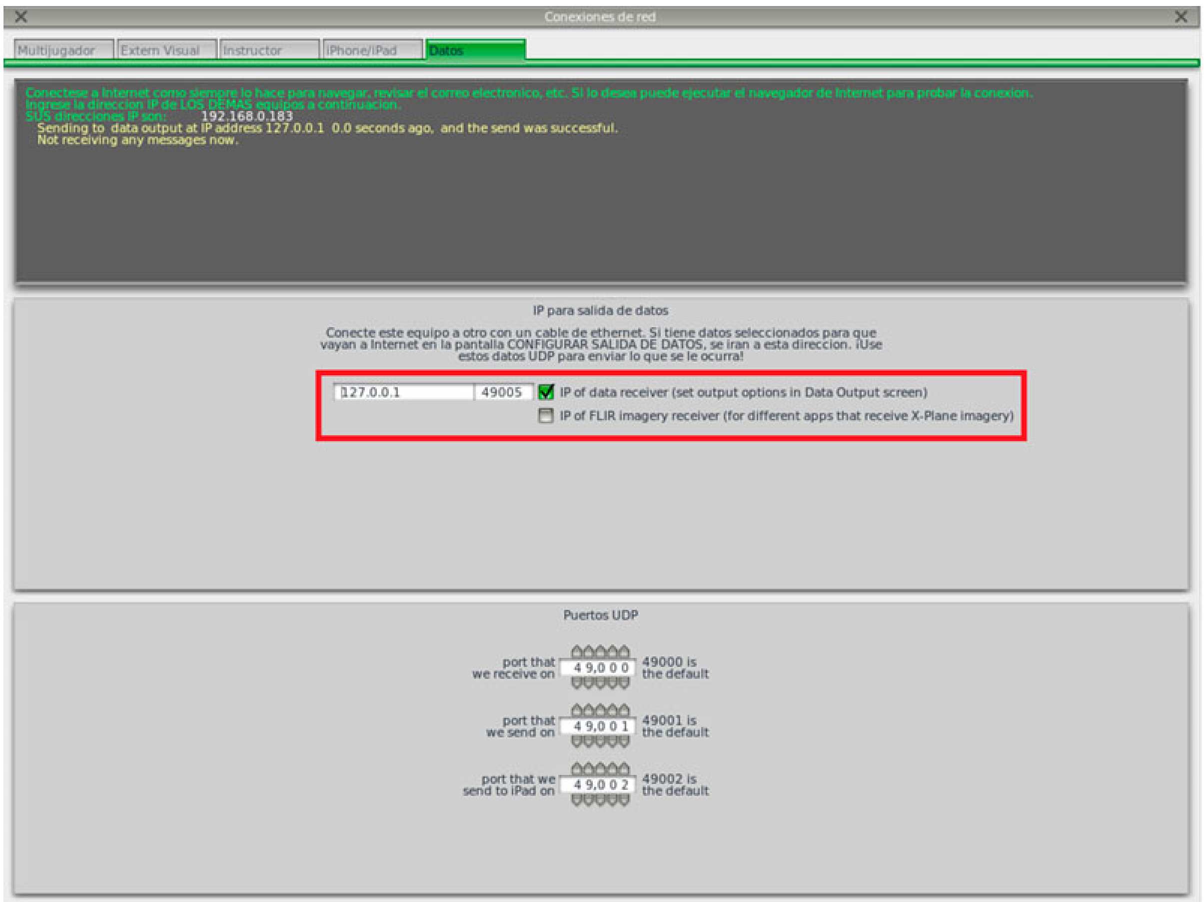
	gSide	Y acceleration – Body axis
16	Angular Velocities	pitchRate	Pitch Rate
		rollRate	Roll Rate
	23	yawRate	Yaw Rate
17	Pitch, Roll, Headings	pitch	Pitch
		roll	Roll
	32	headingTrue	Yaw
18	AoA, side-slip, paths	alpha	used for IAS
		beta	used for IAS
20	lat,lon, altitude	lat	Latitude
		lon	Longitude

48	altMsl		Mean sea level
48	altAgl		Above ground level
21	loc, vec, dist traveled	Vx	X velocity – Body Axis
		Vy	Y velocity – Body Axis
59		Vz	Z velocity – Body Axis

10.1. Professional HIL

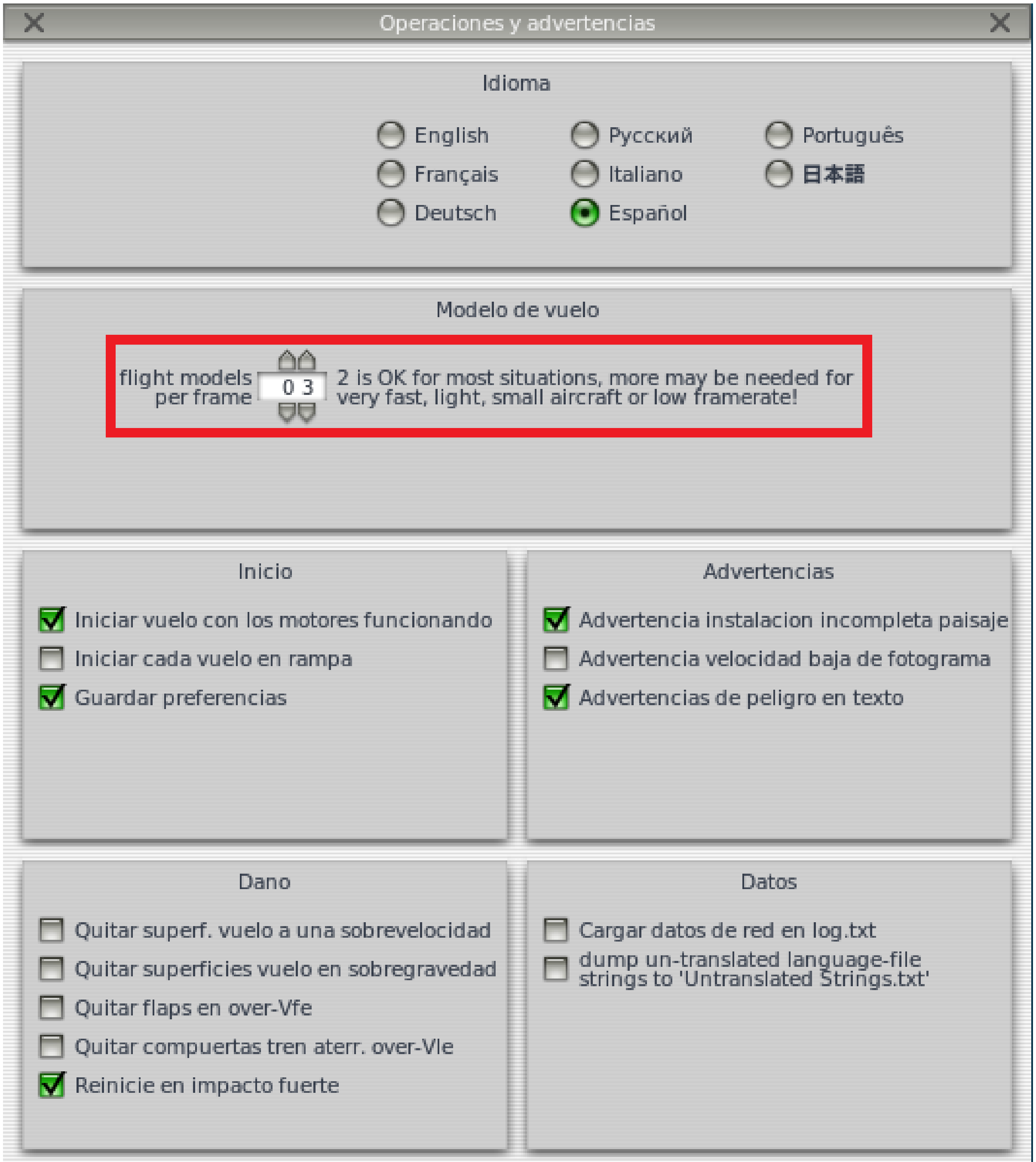
On the **settings** tab, enter into the **network configuration** and select the **data** menu. There user must edit the IP configuration as follows:

- IP: 127.0.0.1 49005
- Check the **IP of data receiver** option
- Uncheck the **IP of Flir Imagery receiver**



X-Plane 10 Network Configuration

On **settings** tab, **General options and warnings**, set the flight models per frame to a minimum of 3. It is needed for small aircraft simulation within X-Plane 10. If your model is vibrating in XPlane 10 you can increase this value but higher PC performances are required.

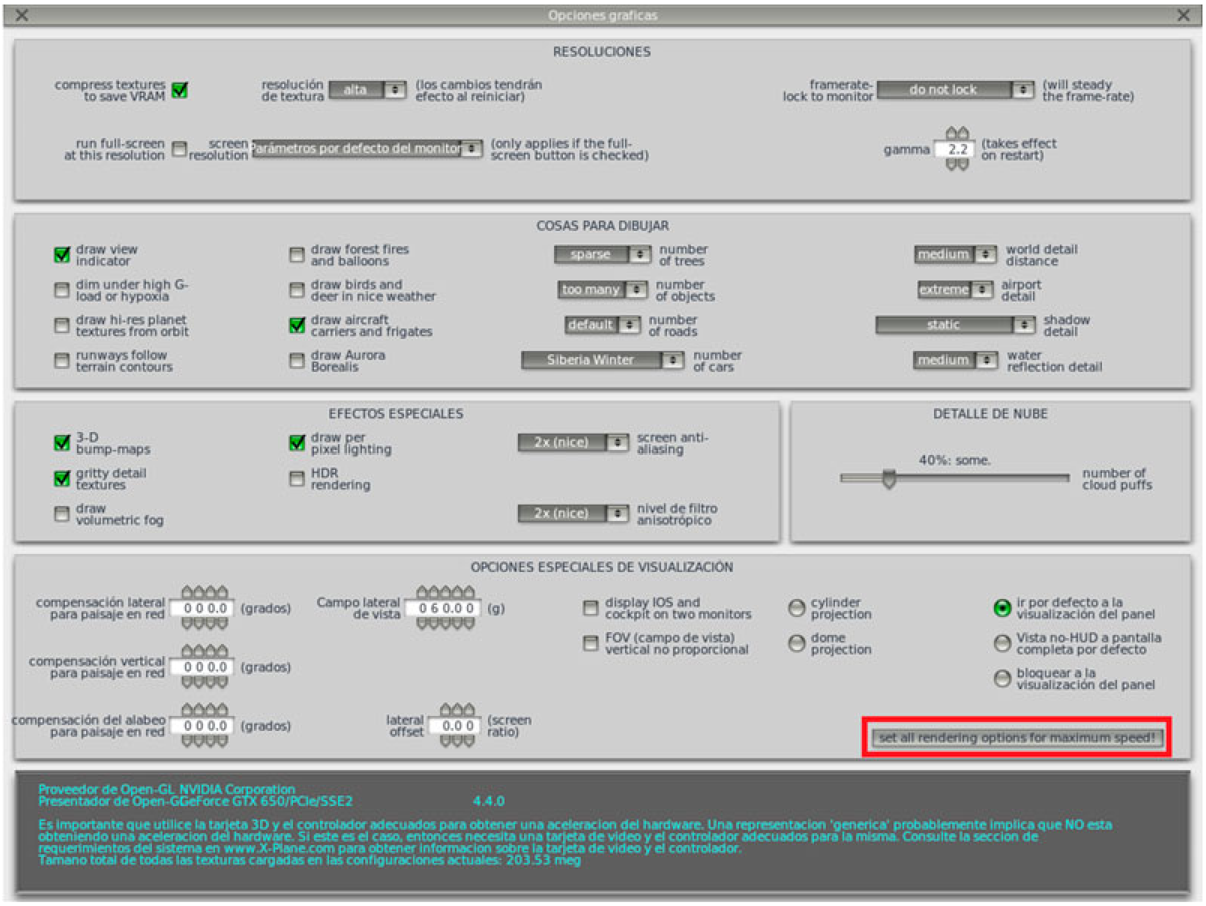


X-Plane 10 General Options and Warnings

10.1.2.2 Low Performance Computer Configuration

In case of using a low performance computer with the HIL Simulator the simulation reliability can decrease. In this case, it is recommended to reduce the graphic quality on the X-Plane 10 simulator.

On the **settings** tab, enter the **graphics option** menu and press on **set all rendering options for maximum speed.**



X-Plane 10 Graphics Option

10.1.3 X-Plane 11 Settings

10.1.3.1 X-Plane 11 Configuration

Since now, Veronte system is compatible with X-Plane 11 for HIL simulation. The demo version of the program can be downloaded from this link.

X-Plane 11 communications settings must be edited in order to have communication with Veronte 4 system. Follow the next steps in order to make a proper configuration.

For low-performance computers, it may be needed to reduce the graphics quality on the simulator, as described below.

10.1.3.1.1 Aircraft Model Installation

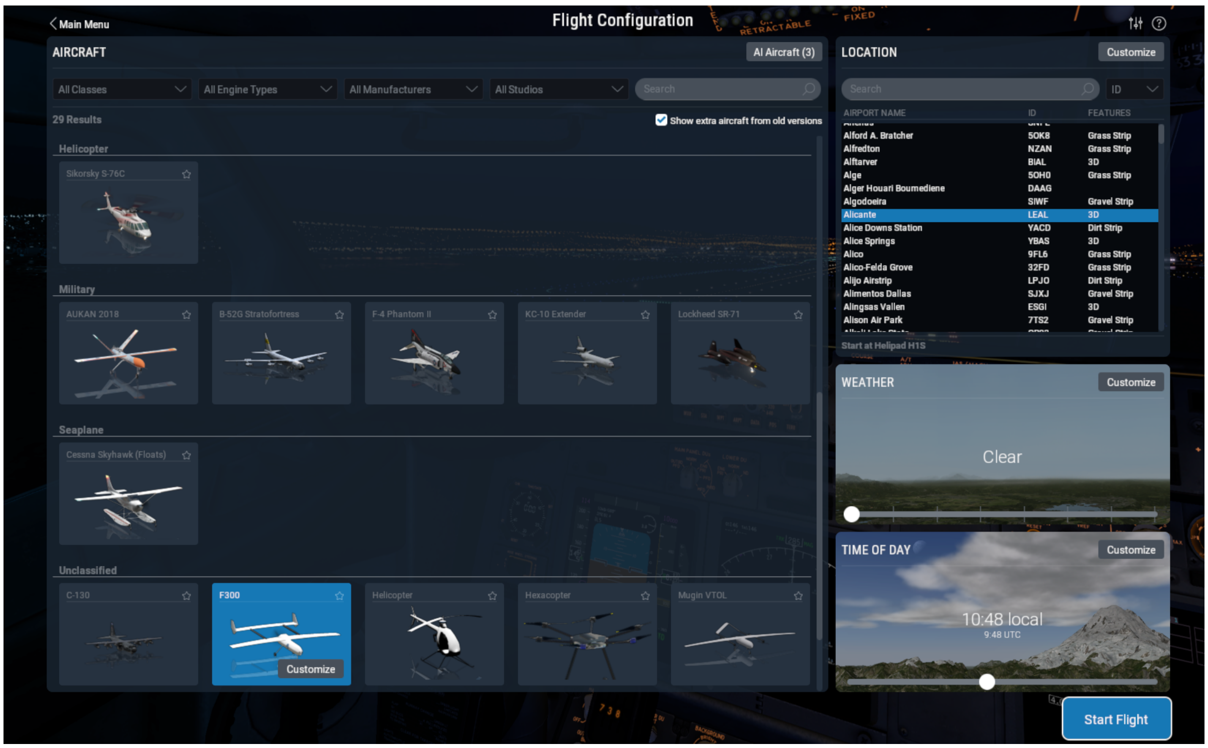
X-Plane 11 simulator is compatible with a wide variety of platforms: airplane, helicopter, multicopter, surface vehicle.… In order to create the platform model, **Plane Maker** tool provided by X-Plane 11 must be used.

Once the aircraft model has been created, it can be integrated on the X-Plane 11 simulator by following next steps:

- Copy the model folder to the "Aircraft" folder within the X-Plane 11 installation directory.
- Copy the content in the "Airfoils" folder, available on the aircraft model folder, to the "Airfoils" directory within the X-Plane 11 installation directory.

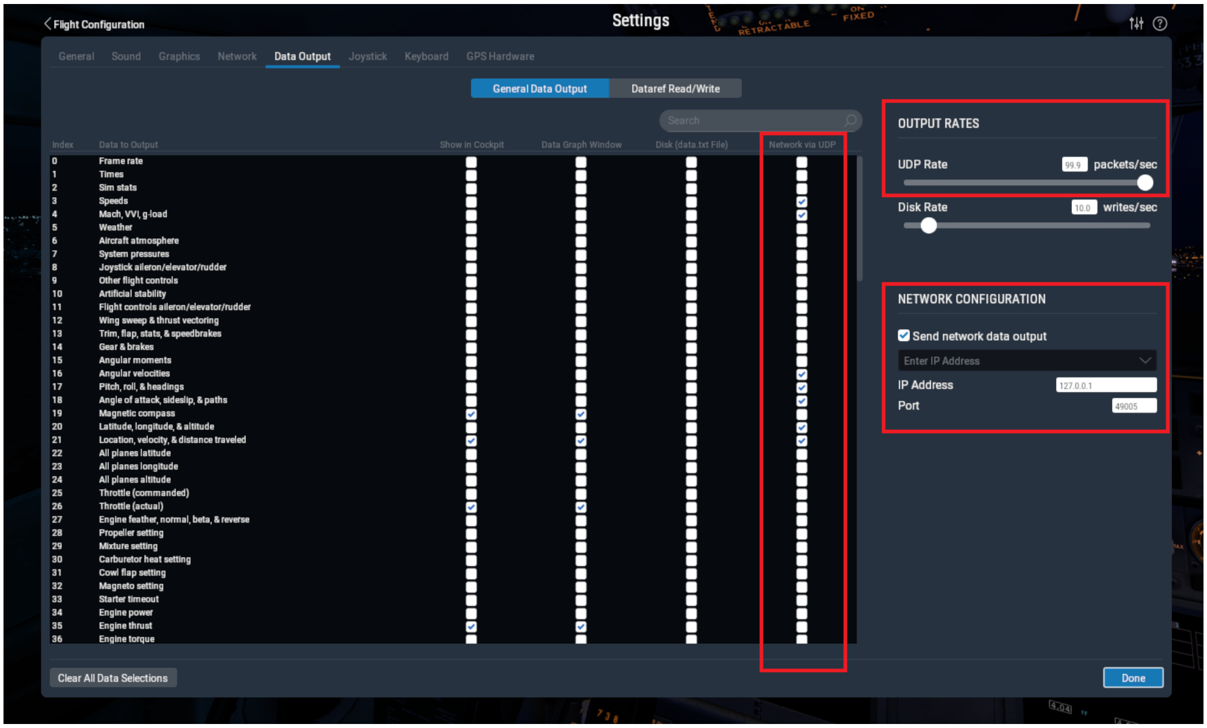
10.1.3.1.2 X-Plane 11 Setup

On X-Plane 11 execution, **Quick Flight Setup** window will be displayed; select which aircraft to use, the starting airport and weather conditions to be simulated during the flight.



Quick Flight Setup

Data transmission settings must be edited on the **Data Output** tab. Select all the variables in order to be sent through the UDP Network and set the UDP rate at maximum speed. The network configuration must be configured as shown in the following picture.



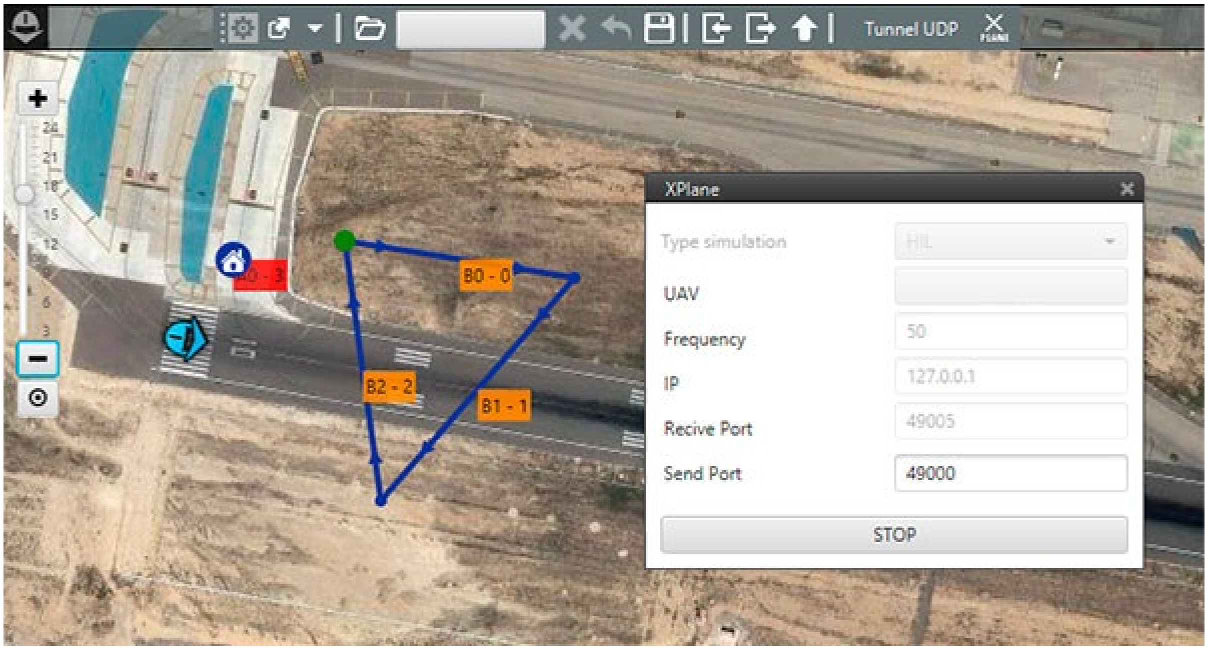
X-Plane 11 I/O configuration

10.1.4 Operation

Once the hardware has been connected and Veronte Pipe and X-Plane have been configured, operation can starts and the system can be operated as if it was on a real flight.

M400 Flight Simulation

When the X-Plane model is uploaded, GPS is simulated and the UAV must be visible on VerontePipe map. With GPS signal it is possible to pass the system to Standby phase and to start flight. X-Plane flight starts from an airport; a custom airport must be defined for simulating out of available airports.



Start GPS simulation

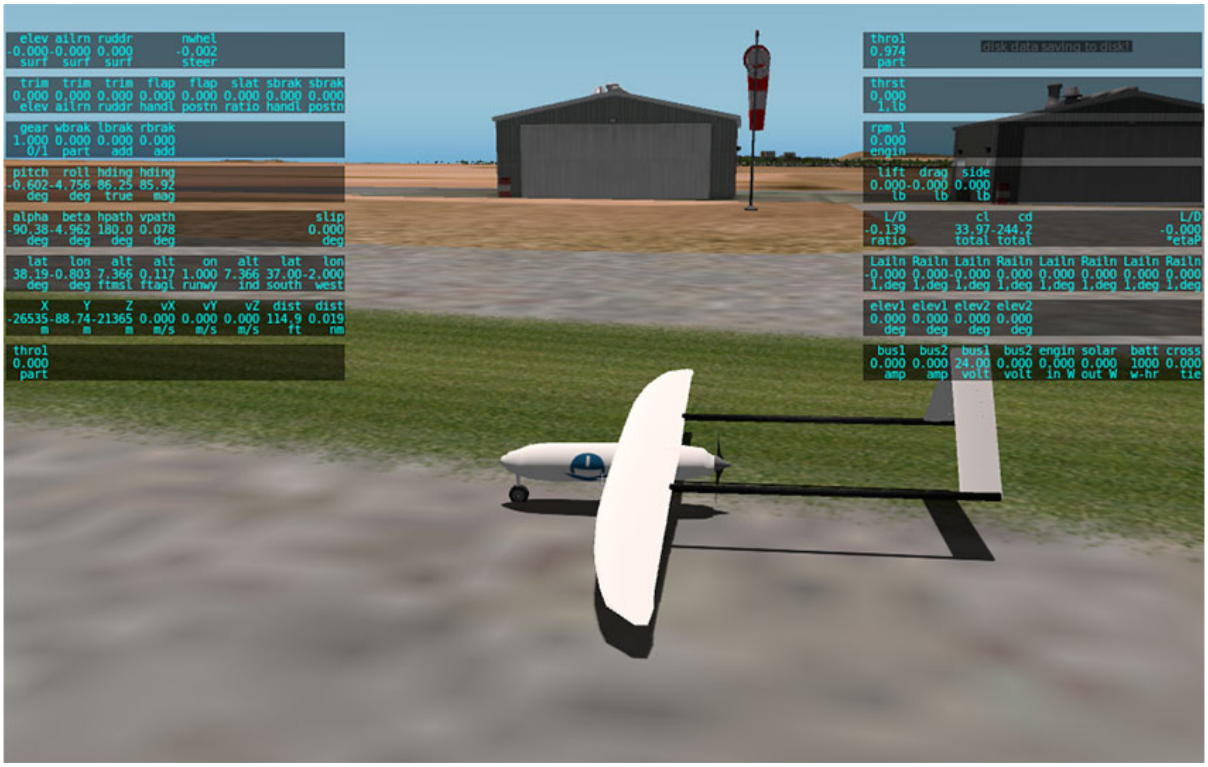
Moreover, it is possible to create a new airport. In order to do that, the user should follow the X-Plane tutorial presented in the next link . Once the aircraft has taken off from the airport on X-Plane the automatic control will start and the aircraft will fly to the defined mission on Veronte Pipe.

> **Warning:** The flight is simulated in all real case aspects. Always make sure that the mission is well configurated and check the terrain profile is in the correct position.

The operator can fly the system as on real flight, being compatible with main Veronte features: real-time mission edit, in-flight automatic to manual control, flight data recording… Sometimes is possible, during an edition saving, that the simulation fails because the simulation link suffers a little interruption (this fact does not exist in real flights).

When using the HIL simulator connected to the platform, control actuators will move as if it were flying. In order to avoid damaging the system or personnel, make sure that the motor is disconnected and there is no shock risk due to the actuators movement.

> **Warning:** Always make sure that motor has been disconnected before starting a simulation. Otherwise the motor will run as if it were flying.



HIL Simulation

Veronte HIL simulator integrates Veronte Autopilot control system within the X-Plane simulator for highly realistic simulation within a safe environment.

Professional Hardware In the Loop (HIL) Simulator package is a powerful tool for Veronte Autopilot integration, development and operator training; permitting to extensively operate the system in a safe environment, prior to conducting real flight operations. Veronte HIL Simulator has been designed for accomplishing with requirements of most demanding unmanned system operators. Some of the main uses of the simulator are:

- Pilot training.
- Veronte configuration for unmanned platform control.
- PID setting.
- Mission configuration.
- Aircraft performance validation.

10.2 SIL Simulink

A software in the loop simulation consists of a Simulink model that simulates the behaviour of the system formed by the autopilot and a vehicle, without having the physical devices connected to the computer, in contrast to the HIL which has both the autopilot and (optionally) vehicle connected to the PC. This option has several advantages when it is compared with a HIL setup:

- Complete simulations without any hardware.
- Possibility of using your own vehicle model: no need to stick to XPlane models. You can add as much physics/complexity as desired.
- Possibility of simulating different kinds of sensors even if they are not fitted in Veronte.

- All results can be exported/visualized to MATLAB workspace simultaneously.
- Veronte Block runs faster than real time, allowing the user to execute a series of simulations in a short time.
- Light computational load.

10.2.1 Autopilot Simulation

The autopilot is implemented in Simulink with an S-Function. This kind of block takes a C, C++, Fortran or even Matlab code, and implements it in a block containing a certain number of inputs and outputs. A typical Veronte s-function is shown below.

imgs/simulation/s-function.png

S-Function containing the autopilot embedded code

Inputs are described in the next table:

PIN	Signal Type	Description	Form	Size	Units
1	Input	Static Pressure 1	[pressure_measurement;sensor temperature]	2x1	Pa / K
2	Input	Static Pressure 2	[pressure_measurement;sensor temperature]	2x1	Pa / K

continues on next page

Table 1 – continued from previous page

PIN	Signal Type	Description	Form	Size	Units
3	Input	Static Pressure 3	[pressure_measurement;sensor temperature]	2x1	Pa / K
4	Input	Dynamic Pressure	[pressure_measurement;sensor temperature]	2x1	Pa / K
5	Input	IMU 1	[acc_x;acc_y;acc_z;gyr_x;gyr_y;gyr_z;sensor temperature]	7x1	m/s^2 / m/s / K
6	Input	IMU 2	[acc_x;acc_y;acc_z;gyr_x;gyr_y;gyr_z;sensor temperature]	7x1	m/s^2 / m/s / K
7	Input	IMU 3	[acc_x;acc_y;acc_z;gyr_x;gyr_y;gyr_z;sensor temperature]	7x1	m/s^2 / m/s / K
8	Input	Magnetometer 1	[mag_x;mag_y;mag_z;sensor temperature]	4x1	T
9	Input	Magnetometer 2	[mag_x;mag_y;mag_z;sensor temperature]	4x1	T
10	Input	Magnetometer 3	[mag_x;mag_y;mag_z;sensor temperature]	4x1	T
11	Input	Magnetometer 4	[mag_x;mag_y;mag_z;sensor temperature]	4x1	T
12	Input	GNSS 1	[1;3;lon;lat;alt;hor_accu;vert_accu;v_north;v_east;v_down;v_accu]	11x1	deg*10^7 / mm / mm/s
13	Input	GNSS 2	[1;3;lon;lat;alt;hor_accu;vert_accu;v_north;v_east;v_down;v_accu]	11x1	deg*10^7 / mm / mm/s
14	Input	Relative Position 1	[1;x_rel;y_rel;z_rel;distance_x;distance_y;distance_z;x_accu;y_accu;z_acc	10x1	cm/mm*10^-1
15	Input	Relative Position 2	[1;x_rel;y_rel;z_rel;distance_x;distance_y;distance_z;x_accu;y_accu;z_acc	10x1	cm/mm*10^-1
16	Input	GPS Time	[week_number;seconds_of_week]	2x1	• / s
17	Input	Lidar 1	[lidar_measurement]	1x1	m
18	Input	Lidar 2	[lidar_measurement]	1x1	m
19	Input	Lidar 3	[lidar_measurement]	1x1	m
20	Input	Lidar 4	[lidar_measurement]	1x1	m
21	Input	Lidar 5	[lidar_measurement]	1x1	m
22	Input	ID Bit Var	[Var_IDs]	50x1	m
23	Input	ID Unsigned Var	[Var_IDs]	50x1	m
24	Input	ID Real Var	[Var_IDs]	50x1	m
25	Input	ADCs	[adc(1-17)]	17x1	•
26	Input	SCIA Data	[serial_data]	1024x1	•
27	Input	SCIA Length	[serial_length]	1x1	•
28	Input	SCIB Data	[serial_data]	1024x1	•

continues on next page

Table 1 – continued from previous page

PIN	Signal Type	Description	Form	Size	Units
29	Input	SCIB Length	[serial_length]	1x1	•
30	Input	SCIC Data	[serial_data]	1024x1	•
31	Input	SCIC Length	[serial_length]	1x1	•
32	Input	SCID Data	[serial_data]	1024x1	•
33	Input	SCID Length	[serial_length]	1x1	•
34	Input	USB Data	[serial_data]	1024x1	•
35	Input	USB Length	[serial_length]	1x1	•

Outputs are the following:

PIN	Signal Type	Description	Form | Size		Units
1	Output	Control Outputs	[control_outputs(1-20)] | 20x1		•
2	Output	Servo Values	[servos(1-32)]	32x1	•
3	Output	Position	[lat;lon;alt]	3x1	rad / m
4	Output	Heights	[msl,agl]	2x1	m
5	Output	Velocities	[longitudinal_v;lateral_v;velocity(module)]	3x1	m/s
6	Output	IAS TAS GS	[ias,tas,gs]	3x1	m/s
7	Output	MSL	[msl_from_qnh;msl_from_ISA]	2x1	m
8	Output	Angle of Attack / Sideslip	[angle_of_attack;sideslip]	2x1	rad
9	Output	Q_Infinty	[dynamic_pressure]	1x1	Pa
10	Output	IAS RAW	[ias_raw]	1x1	m/s
11	Output	Tangential Acceleration	[tangential_acceleration]	1x1	m/s^2
12	Input	Body Velocities	[lon_v;lat_v;vertical_v]	3x1	m/s
13	Output	Angular Velocities	[roll_rate;pitch_rate;yaw_rate]	3x1	rad/s
14	Output	Angular Acceleration	[acc_z_axis;acc_y_axis;acc_x_axis]	3x1	rad/^2
15	Output	Acceleration NED	[acc_north;acc_east;acc_down]	3x1	m/s^2

continues on next page

Table 2 – continued from previous page

PIN	Signal Type	Description	Form | Size		Units
16	Output	Velocity NED	[v_north;v_east;v_down])	3x1	m/s
17	Output	Angles	[Yaw;Pitch;Roll])	3x1	rad
18	Output	Co-Angles	[co-Yaw;co-Pitch;co-Roll]	3x1	rad
19	Output	Aerodynamic Angles	[heading,flight_path;bank_angle]	3x1	rad
20	Output	Acceleration Body	[acc_x,acc_y;acc_z]	3x1	m/s^2
21	Output	Load factor	[nx;ny;nz]	3x1	•
22	Output	SCIA Data	[serial_data]	1024x1	•
23	Output	SCIA Length	[serial_length]	1x1	•
24	Output	SCIB Data	[serial_data]	1024x1	•
25	Output	SCIB Length	[serial_length]	1x1	•
26	Output	SCIC Data	[serial_data]	1024x1	•
27	Output	SCIC Length	[serial_length]	1x1	•
28	Output	SCID Data	[serial_data]	1024x1	•
29	Output	SCID Length	[serial_length]	1x1	•
30	Output	USB Data	[serial_data]	1024x1	•
31	Output	USB Length	[serial_length])	1x1	•
32	Output	Unsigned Variables	[selected variables(1-50)]	50x1	•
33	Output	Bit Variables	[selected variables(1-50)]	50x1	•

continues on next page

Table 2 – continued from previous page

PIN	Signal Type	Description	Form | Size		Units
34	Output	Real Variables	[selected variables(1-50)]	50x1	•

10.2.2 Sensors Simulation & Input Examples

This section aims to ilustrate how to implement the inputs described in the previous section. The structures that are shown here are orientative and, of course, can be adapted by the user.

A basic subsystem that must be built in every flight simulation is an environment model. This model, groups the atmospherical properties and it changes according to different variables as well as the magnetic field at certain coordinates on earth. This block will be the basis of most the sensors that will be shown later on:

imgs/simulation/env.png

Environment Example Block

The static pressure sensor block can be easily derived by taking the pressure from the environment model. The only parameter that must be added is the temperature of the sensor (in this case is the OAT + 60):

imgs/simulation/static.png

Static Pressure Example Block

The dynamic pressure input can be also obtained by using the standard dynamic pressure simulink block. The inputs are the speed (body axis) and the density:

imgs/simulation/dynamic.png

Dynamic Pressure Example Block

There is also a dedicated simulink block in the aerospace blockset that models a set of accelerometers and gyroscopes. The inputs of this block are the acceleration and angular velocities coming out of dynamics of the vehicle.

imgs/simulation/imu.png

IMU Example Block

> **Warning:** Do not connect the gravity port to the IMU block.

The magnetometer block is simply a rotated environment magnetic field where the temperature of sensor has been added (same as before OAT + 60). The reason why there is a selector crossing the signals is because there is a rotation matrix pre-configured in each PDI (the real magnetometer is not aligned with the autopilot axis):

imgs/simulation/magneto.png

Magnetometer Example Block

The GNSS receiver and the relative position input (RTK) are only a multiplexor that creates an array. The position and the velocity are outputs of the vehicle model:

imgs/simulation/gps.png

GNSS Receiver Example Block

imgs/simulation/rned.png

Relative Position (RTK) Example Block

The analogue inputs follow the same reasoning, the user must add the desired values to the mux:

imgs/simulation/adcs.png

ADCs Example Block

Veronte can manage input and output serial ports as explained here. An easy way to create serial frames (data en length wires) is by using the simulink UDP block. Therefore, the data coming in to veronte should be sent though UDP (if this approach is taken):

imgs/simulation/sci.png

SCI Example Block

10.2.3 Monitoring Telemetry with Simulink

As explained before, there are three inputs specially dedicated to select custom telemetry (pins 22,23 and 24). The input structure of those is fixed and must be of size 50, as illustrated here:

imgs/simulation/uvar.png

Telemetry ID Mux

The ID of each variable vailable in Veronte can be easily found in Veronte Pipe by adding a new workspace widget. The ID is labelled right before the variable name:

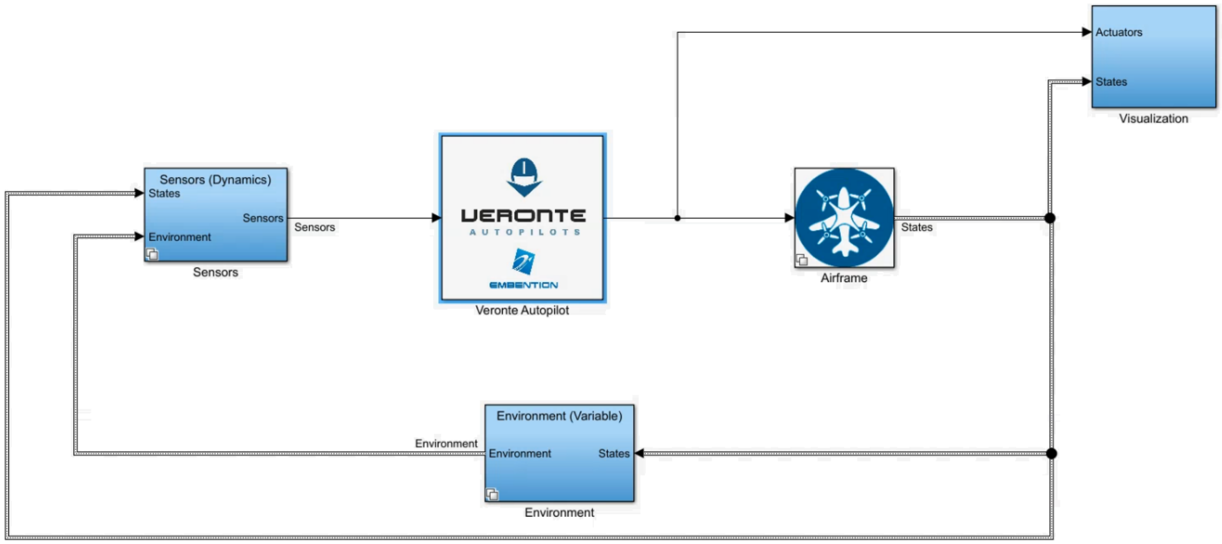
imgs/simulation/ids.png

ID indicator

Lastly, to see their values, just place a *scope* connected to the matching output (pins 32, 33 or 34).

10.2.4 Complete Simulation

After setting the main blocks, the result should look like this:



Complete Setup Example

The main systems are:

- **Veronte Autopilot**: It contains our flight control software.
- **Airframe**: a model of the flight dynamics.
- **Environment**: a model of the atmosphere, magnetic field, WGS84…
- **Sensors**: it contains individual blocks of all the sensors that veronte needs as input.
- **Visualization**: It contains, scopes, flight instruments…

The time step should be set to 0.0002 as shown in the next figure in order to guarantee a good GNC/Adquisition frequency:

imgs/simulation/settings.png

Time step settings

10.2.5 Connecting Simulink & Veronte Pipe

Our Software-in-the-loop simulator can be connected and used alonside Veronte Pipe software. To do it. Follow this steps:

1. Add a UDP serial communication block and connect it to USB data and length.
2. Add a second UDP serial communication block and connect it to the USB output of veronte.

imgs/simulation/udp.png

UDP Blocks

3. Configure your destination port.

imgs/simulation/port1.png

Destination UDP Port

4. Set an ethernet network in Preferences as shown using the destination port selected before.

imgs/simulation/port2.png

Destination UDP Port (Pipe)

10.2.6 Dealing with PDI files

PDI are uploaded exactly the same way that would be uploaded to a physical autopilot. Once connected to Veronte Pipe, a *virtual* autopilot will be detected in safe mode, then these steps can be followed.

Warning: Restart functionalities do no restart the simulation and must be done manually.

Veronte S-function deals with a direct copy of the binary files of a veronte autopilot, if a manual setup is needed, these files will be found in the same folder as the simulink block in a subfolder called *mmc*:

imgs/simulation/mmc.png

Folder structure (matlab)

10.3 3D Simulation

Veronte 3D Simulation

Veronte PFD works using a flight simulator for representing the worldwide geographical scenarios: *lands, seas, mountains, cities, airports, airfields, heliports, etc…* In addition, unlike other 3D PFDs an internet connection is not necessary so it can be operated from any location and environment without any delays in scenario loading.

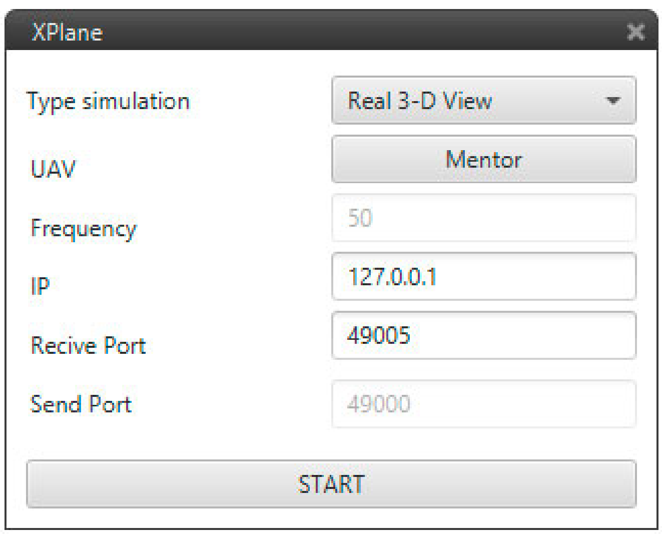
This feature displays a 3D view of the aircraft which is being piloted, while it permits to use it as a 3D PFD (Primary Flight Display) when using the 1st camera view. This system permits to display custom aircraft models in the virtual environment. Planemaker tool is available for creating custom models, thereby the operator can see in the interface aircraft model which is being flown.

Video: Veronte Situational Awareness for UAVs.

10.3.1 Veronte Pipe Configuration

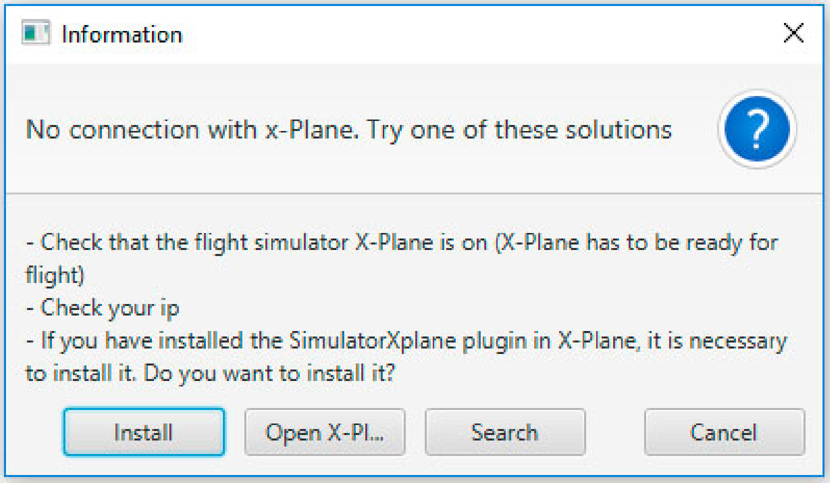
To configure the 3D Simulation in Veronte Pipe it is necessary to complete some steps. First of all, the user must connect Veronte Autopilots to the PC and run Veronte Pipe and XPlane simulator.

When the system is ready, it is possible to open the XPlane connection on Pipe following the path Setup > XPlane. In this panel, the user must select the Real 3-D View from the menu, complete the editable fields as described in section *Mounting* and configure X-Plane following the same section.

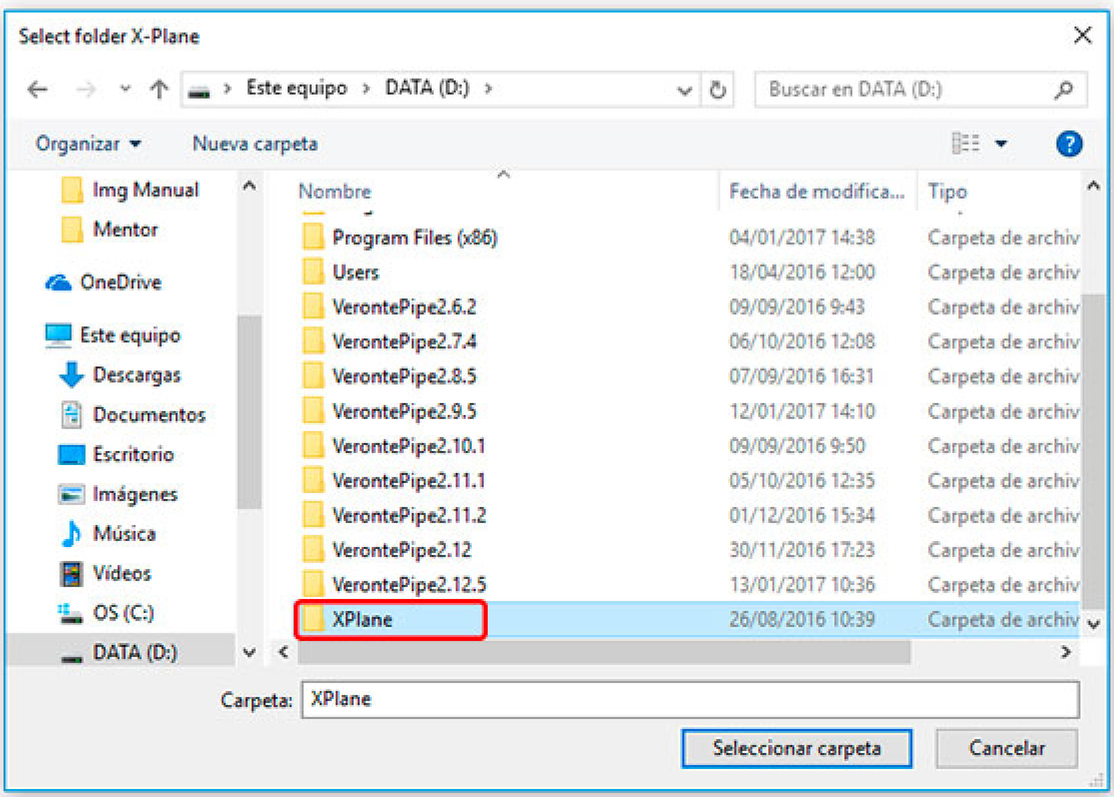


Veronte 3D Simulation

When the connection with XPlane is not possible, an information window shows up. In order to admit the input connection, XPlane requires a plugin installation. By clicking on "Install", it is possible to select the XPlane folder on the PC in order to install it.



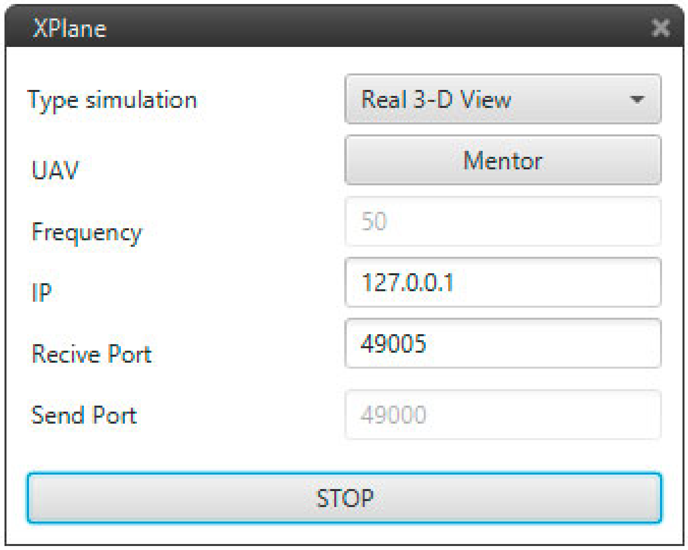
Plugin installation request



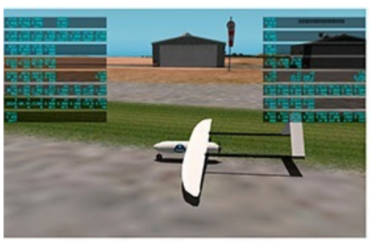
X-Plane folder selection

When the XPlane connection is correctly configured, it is possible to click on "Start" in the XPlane connection panel and the 3D Simulation will start. XPlane will take position on the map and attiude datas from Veronte Pipe and will reproduce the aircraft movement in the simulation environment following the real one.

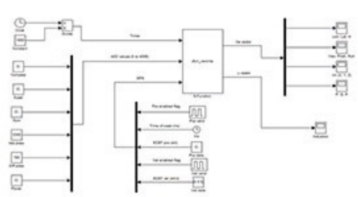
When the flight is over and the simulation is not required, in order to stop the connection with XPlane, it is sufficient to open the XPlane connection panel and click on “Stop”.



Stop XPlane connection



Professional HIL



SIL Simulink

3D Simulation

This section of the manual presents the three possible ways to simulate using Veronte System:

- **Hardware In the Loop simulation (HIL):** This kind of simulation shows Veronte Pipe software performing as it does during a real operation. Meanwhile running the simulation Veronte Autopilot “thinks” it is flying, taking simulator input as real sensor data. The whole flight is controlled by Veronte Autopilot in the virtual environment, making it the perfect tool for vehicle development and for training in the use of Veronte Autopilot.
- **Software In the Loop simulation (SIL):** The software in the loop simulation consists in the creation of a Simulink model that simulates the behaviour of the system formed by the autopilot and platform, without having the physical devices connected to the computer, in contrast to the HIL which has both the autopilot and platform connected to the PC. The SIL allows the user to simulate faster than real time, which allows the user to simulate a system several times.
- **3-D Simulation:** When Veronte Air is flying, it is possible to connect the system with X-Plane simulator through the 3-D Simulator. In this way, the user can see a virtual reproduction of the flight in real time and use it in a FPV-like mode.

CHAPTER

ELEVEN

EXAMPLES

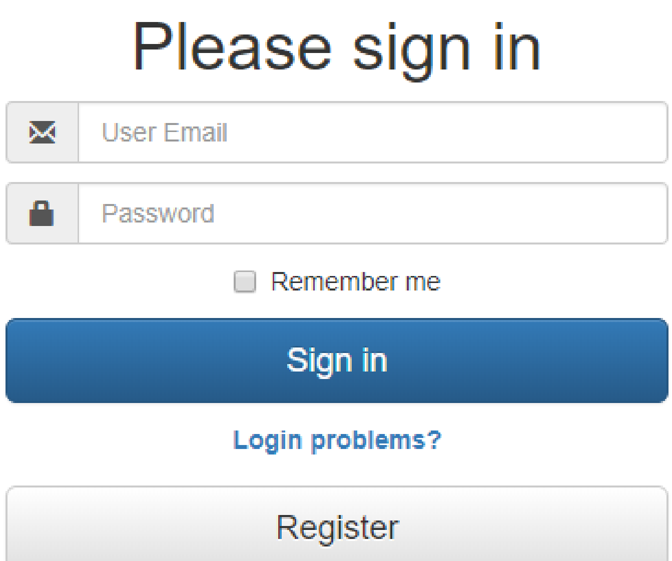
11.1 4G Communication

11.1.1 4G Communication with Veronte Autopilot

Embention integrates its own Veronte Autopilot with a cloud service that allows the user controlling its own platform around the world with real time telemetry.

1. **Add an Autopilot to cloud**

   Enter to cloud.

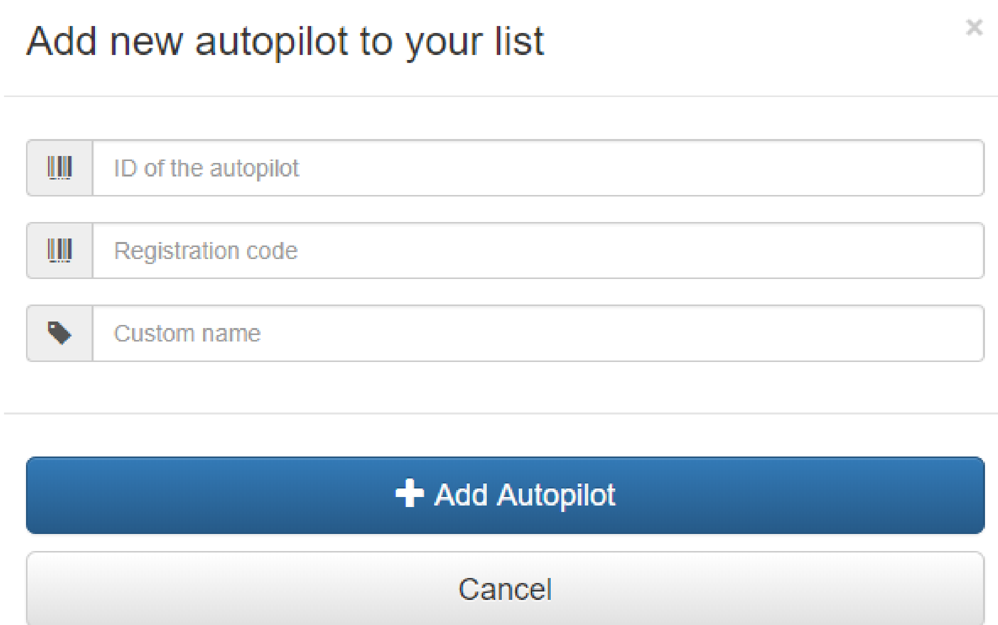


Login

Sign in or create an account to be able to access to the cloud services . From your personal space the user is able to add new Veronte Autopilots by clicking on the top right corner.

Autopilots List

Introduce the information required in the list and a new autopilot will show up on the previous list.



Add new Autopilot

Contact support@embention.com for further details

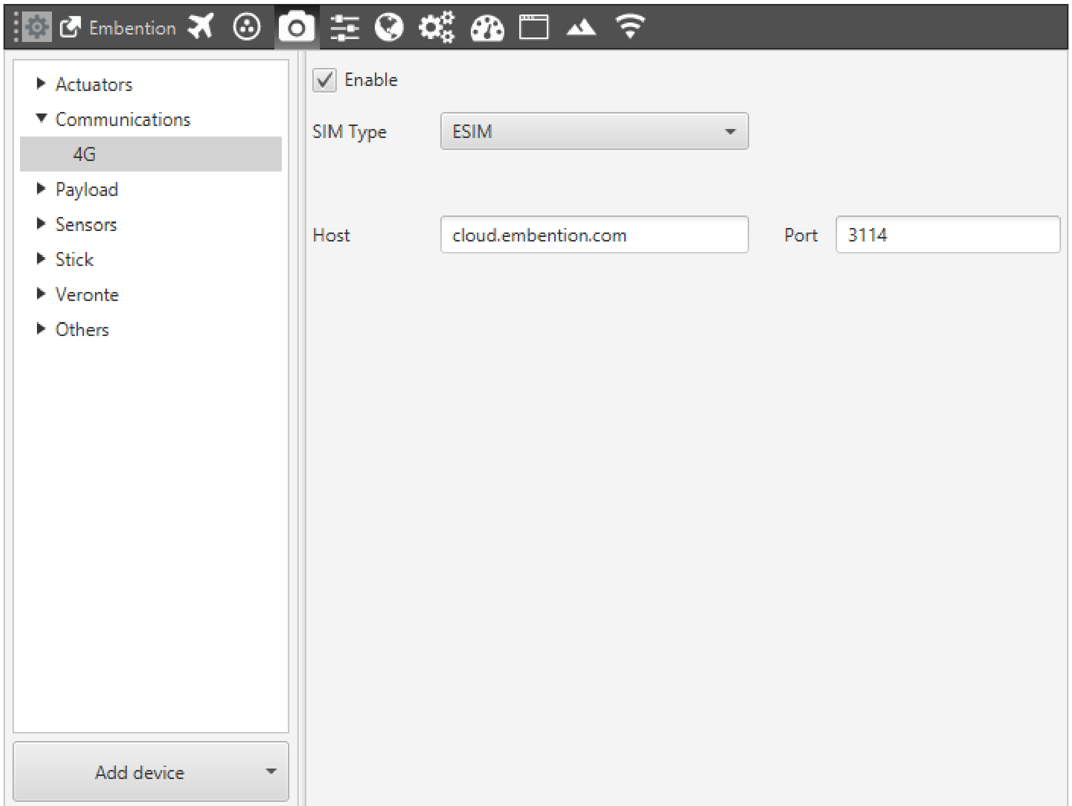
2. **Activate 4G Communication in Veronte Autopilot**

   In order to activate 4G communications open the 'Set up' menu and select 'Devices'. By default the 4G communication is disabled. Select ESIM or SIM accordingly to your set up.

   ESIM

   For all Veronte Autopilots with ESIM activated the following fields should be filled.

   - Host: cloud.embention.com
   - Port: 3114

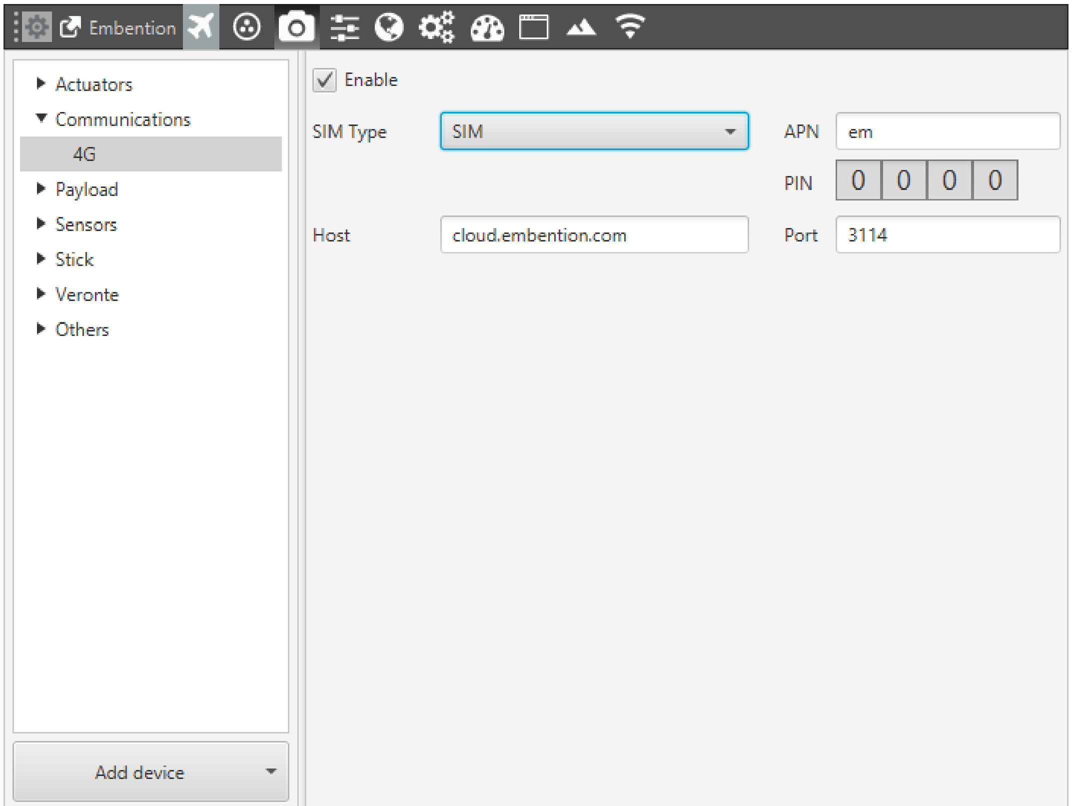


ESIM

SIM

For Veronte Autopilots with a physical SIM card installed in Embention facilities the following parameters have to be set up:

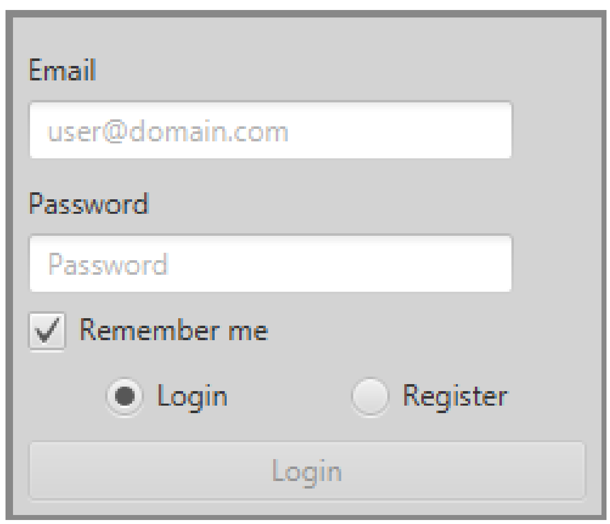
- Host: cloud.embention.com
- Port: 3114
- APN: APN given by the 4G provider.
- PIN: PIN code of the SIM card installed. It should be defined before enabling the communication.



SIM

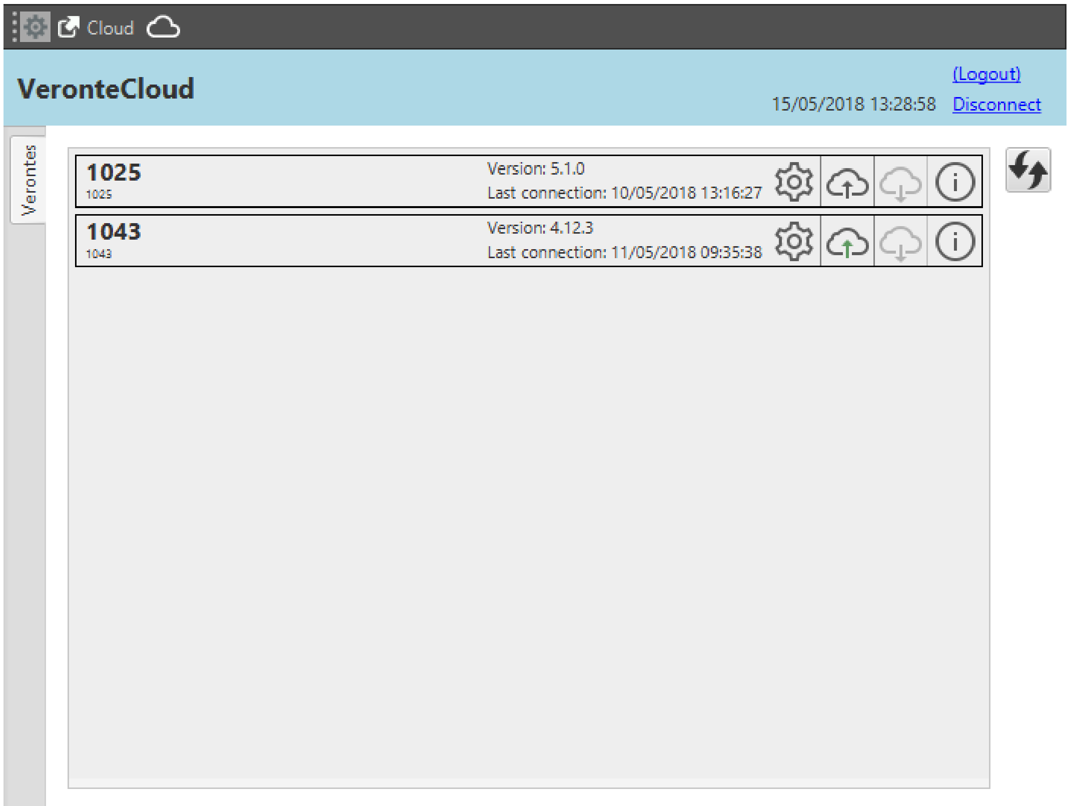
3. **Add a Veronte Autopilot from cloud**

   Open Veronte Pipe and Select the Cloud menu. The user will be requested to enter its own cloud account:



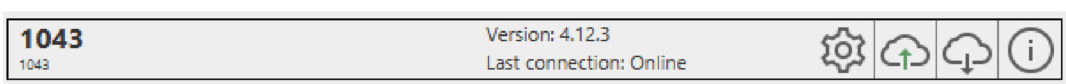
Cloud

All veronte autopilots registered in Veronte Clould will appear showing the **Name, Version** and **Last connection information.**



Cloud

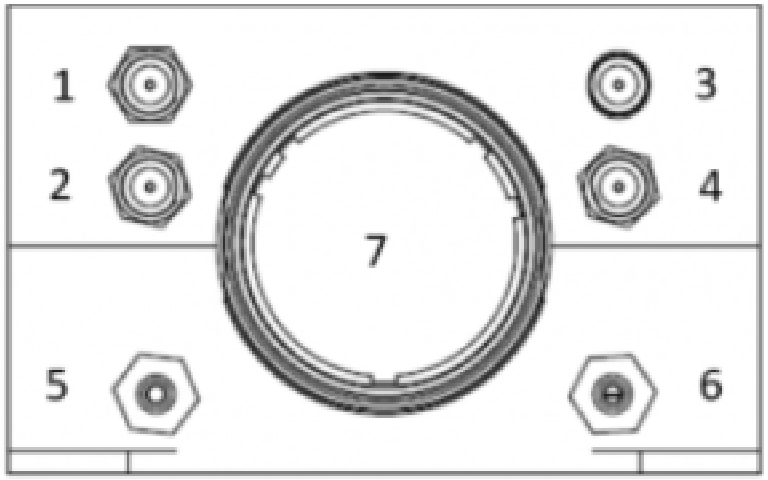
When a Veronte is **ONLINE** the user is able to download the information of the cloud by clicking .



A new Veronte will appear in Pipe with exactly the same capabilities compared with a veronte connected by USB.

4. **Antenna Connection**

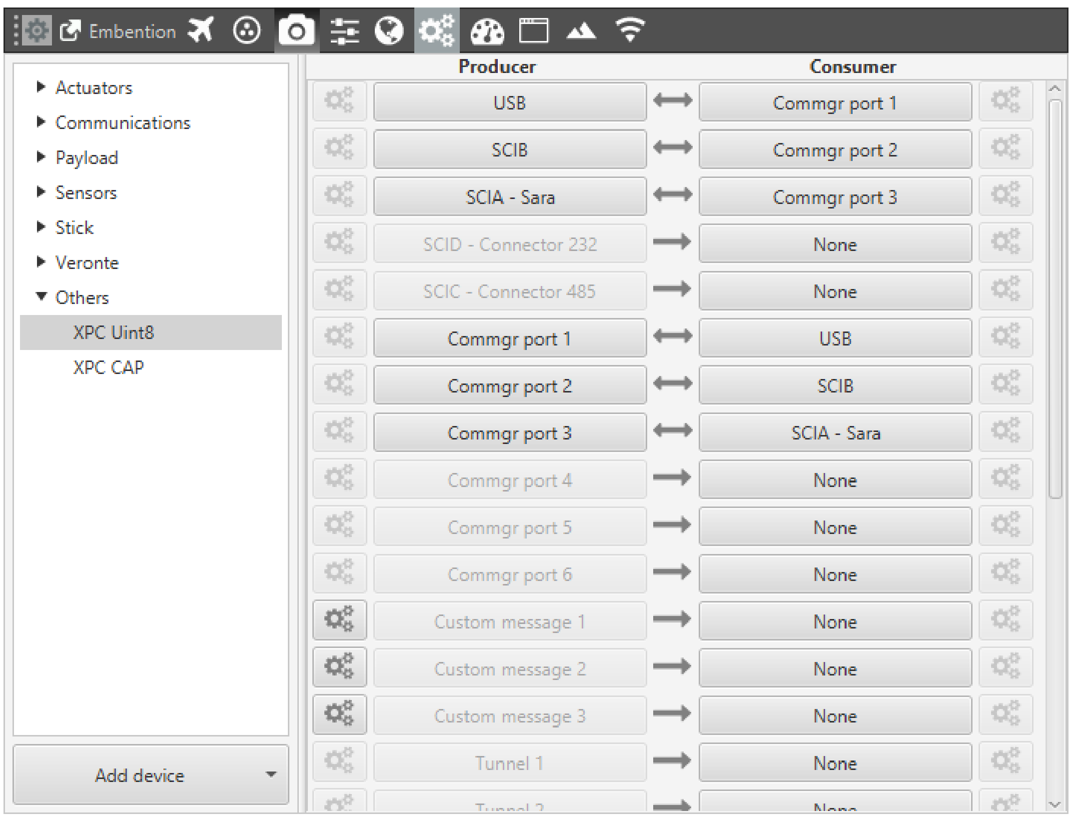
   Connect the 4G antenna in 3 (SSMA connector).



Antenna Connector

5. **XPC UINT8 Configuration**

   Check SCIA-Sara configuration match the following scheme:



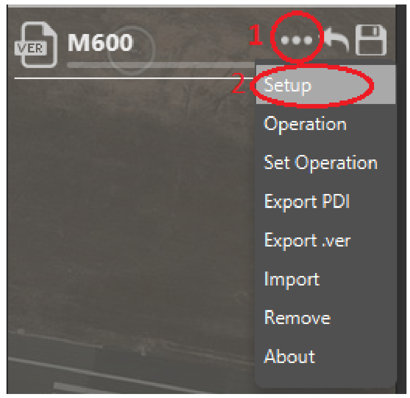
XPC Uint8

11.2 Configurations

11.2.1 Device Configurations

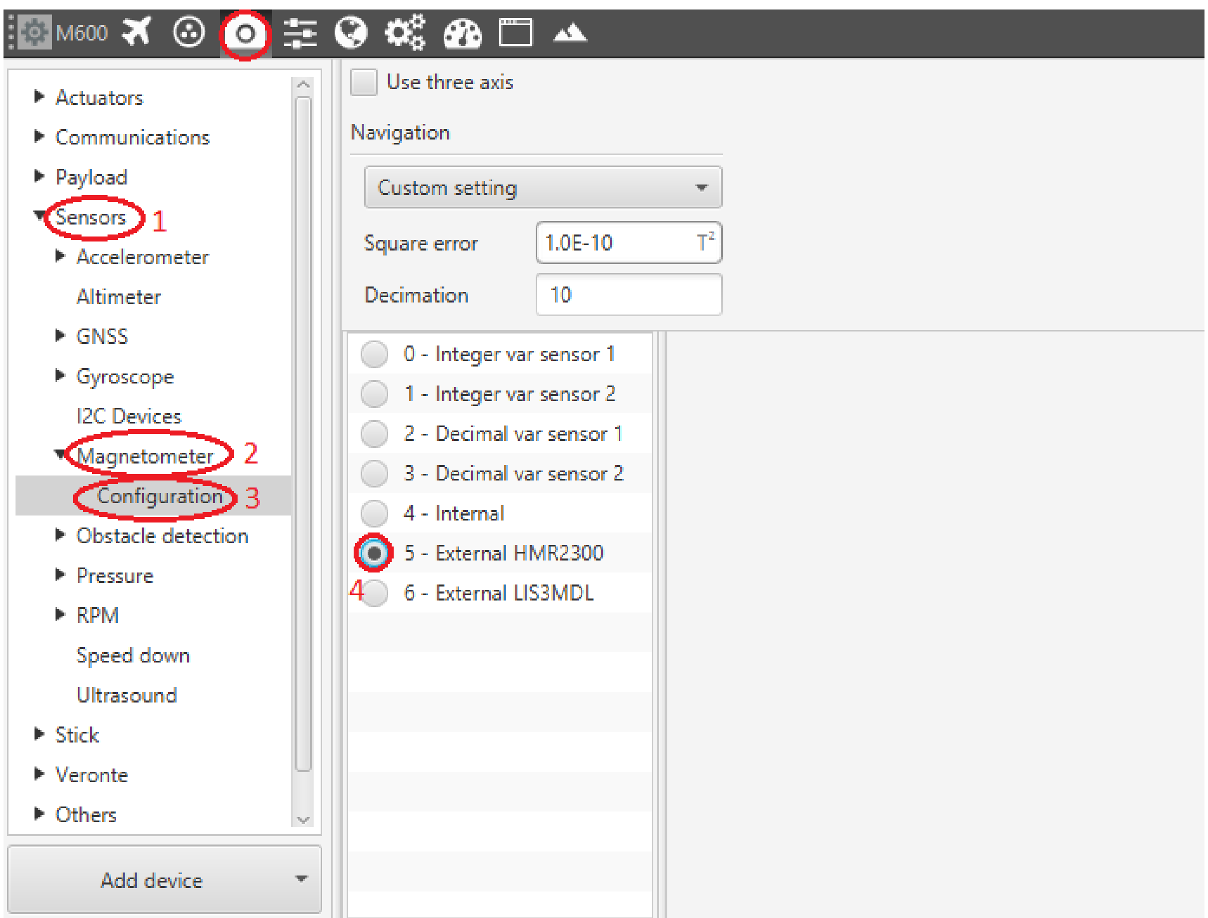
11.2.1.1 Magnetometer Honeywell HMR2300-232

We will configure this device in Pipe direcly. If we click in:



Step 1

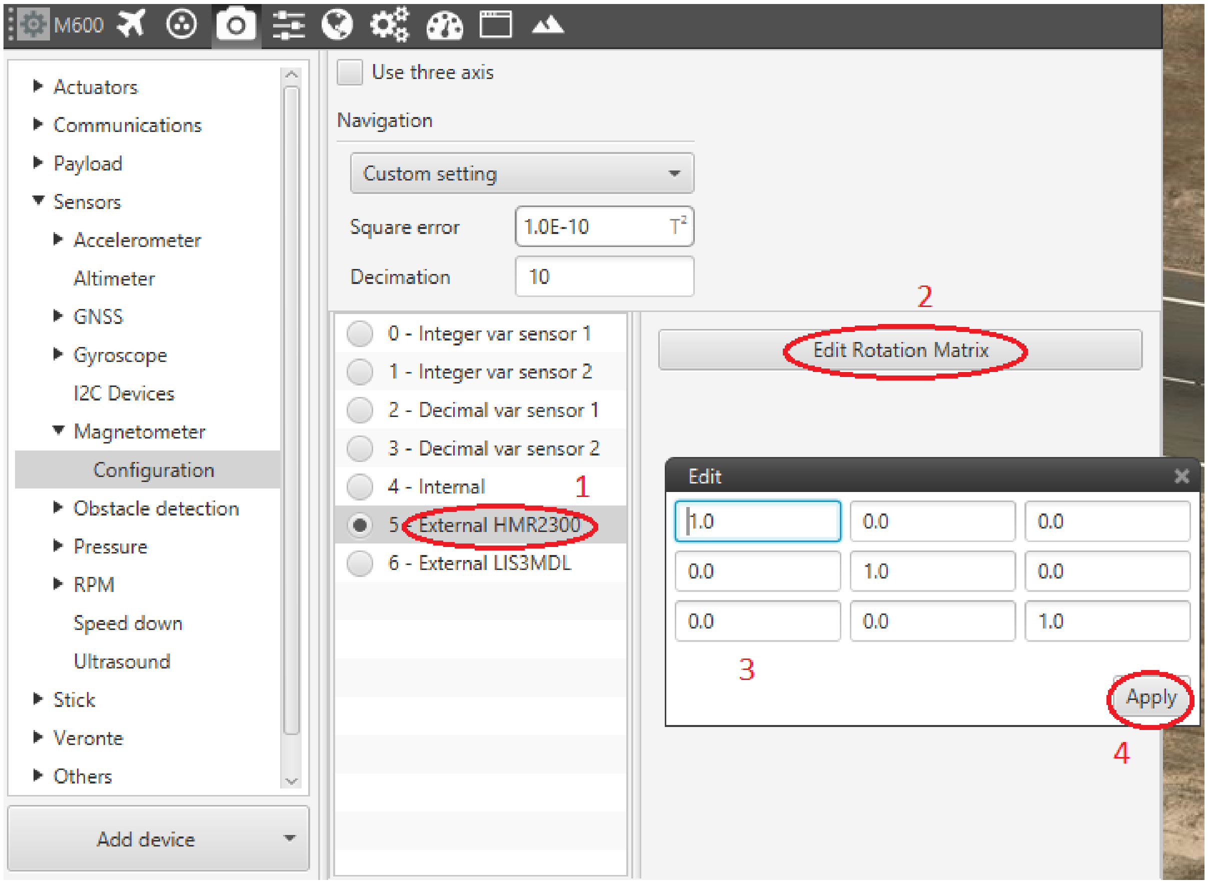
Pipe will display you the Setup Toolbar (for more information see *Setup Toolbar*):



Step 2

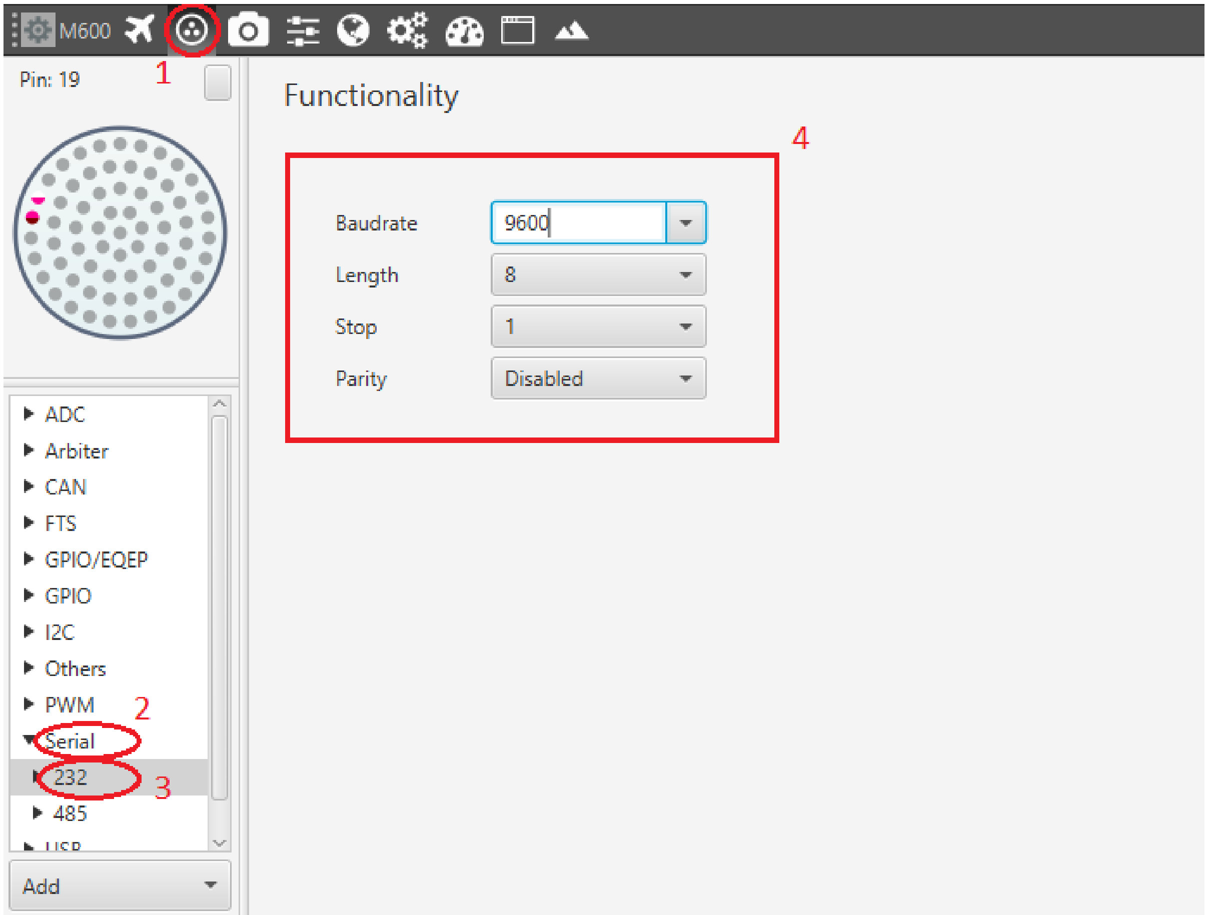
Following the steps you will arrive to the configuration panel and you have to choose **5 - External HMR2300**.

Furthermore, you can configure your own Rotation Matrix clicking in:



Step 3

To finish the configuration of our magnetometer, you have to go to the **Connections** section in the **Setup Toolbar**.



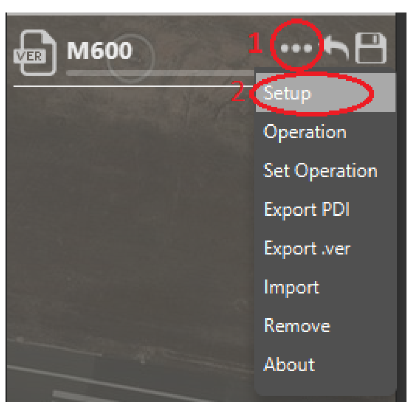
Step 4

If you follow the indications and introuce the values shown in the **Figure: Step 4** your magnetometer will be configured.

To get more information of the Magnetometer Honeywell HMR2300-232 datasheet you can go to: Smart Digital Magnetometer HMR2300

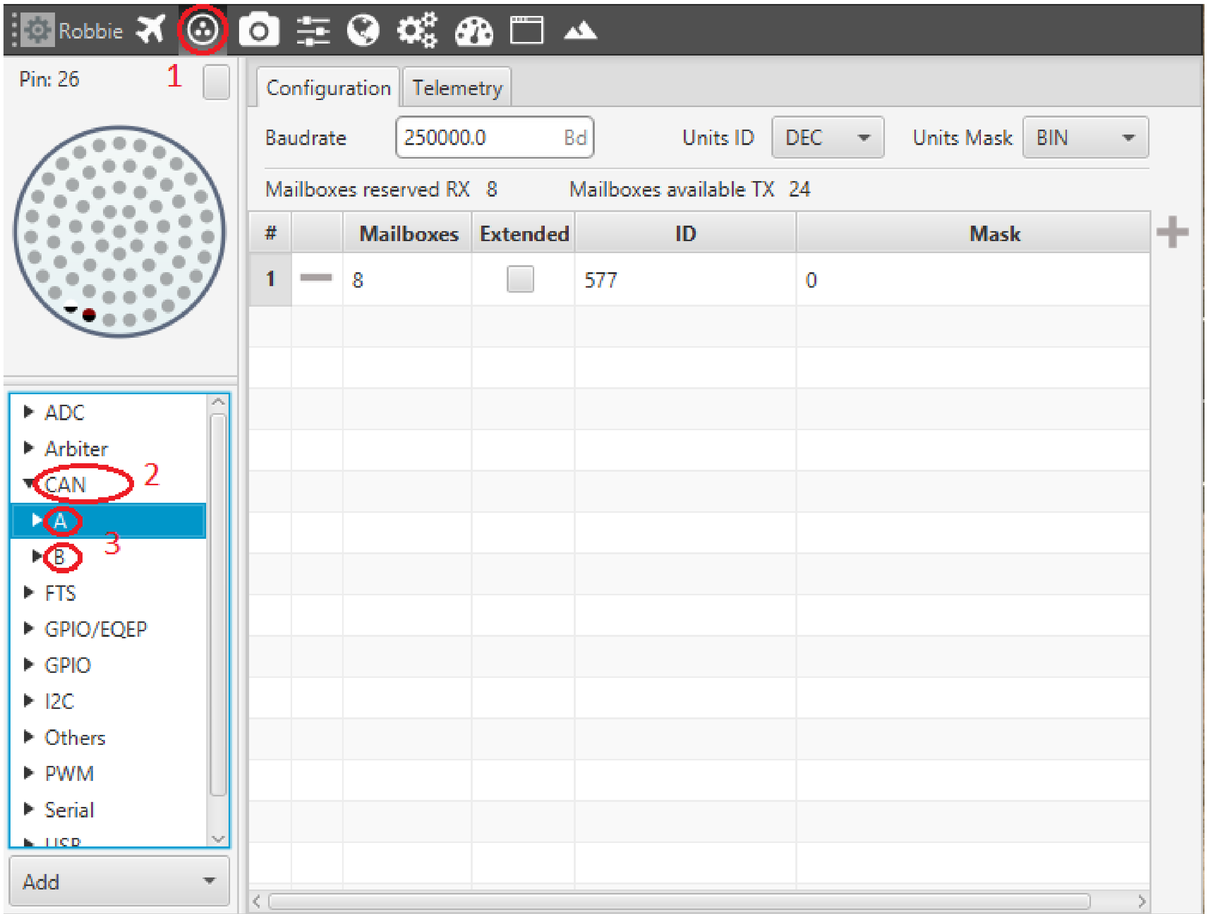
11.2.1.2 Radar Altimeter Smartmicro

We will configure this device in Pipe direcly. If we click in:



Step 1

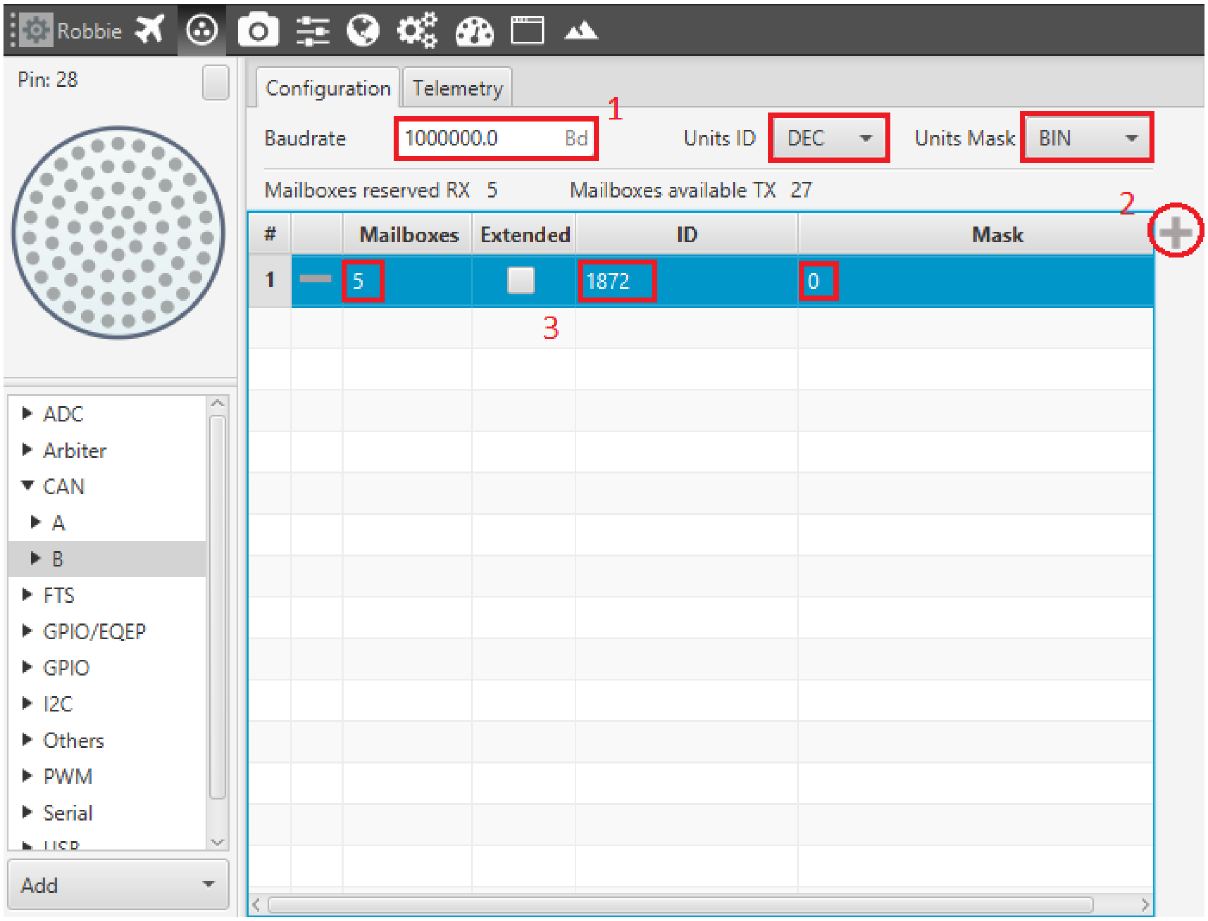
Pipe will display you the Setup Toolbar (for more information see *Setup Toolbar*):



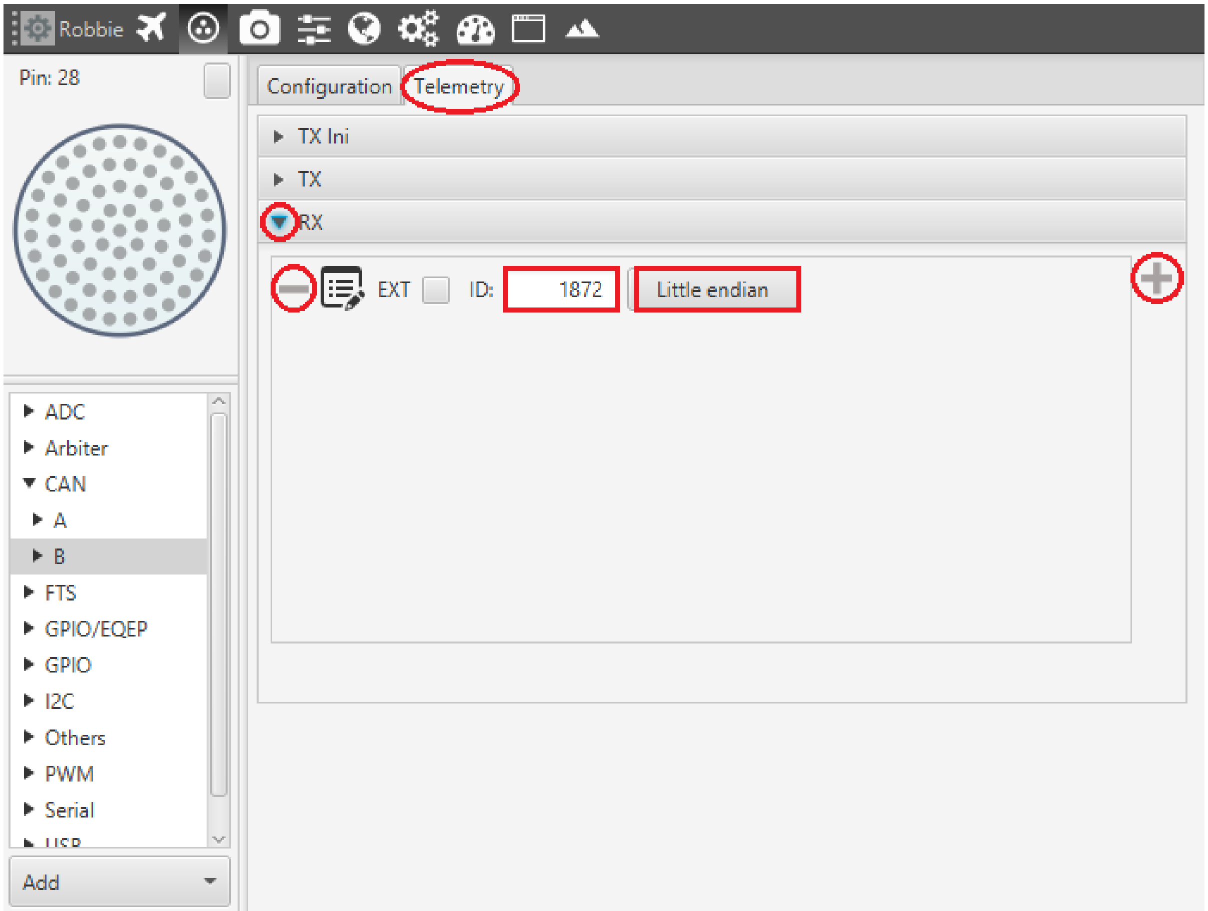
Step 2

Following the steps you can access successfully in the CAN connections data sheet, here we will configure our altimeter. As you can see, you can work with two independent CAN data buses (A and B), each one have two cables (High and Low) one with positive polarity and another one negative, the figure of the upper left corner shows you what pins they are.

In this example we have configured the Smartmicro Radar-Altimeter in channel B, the recommended values to this configurtion are:

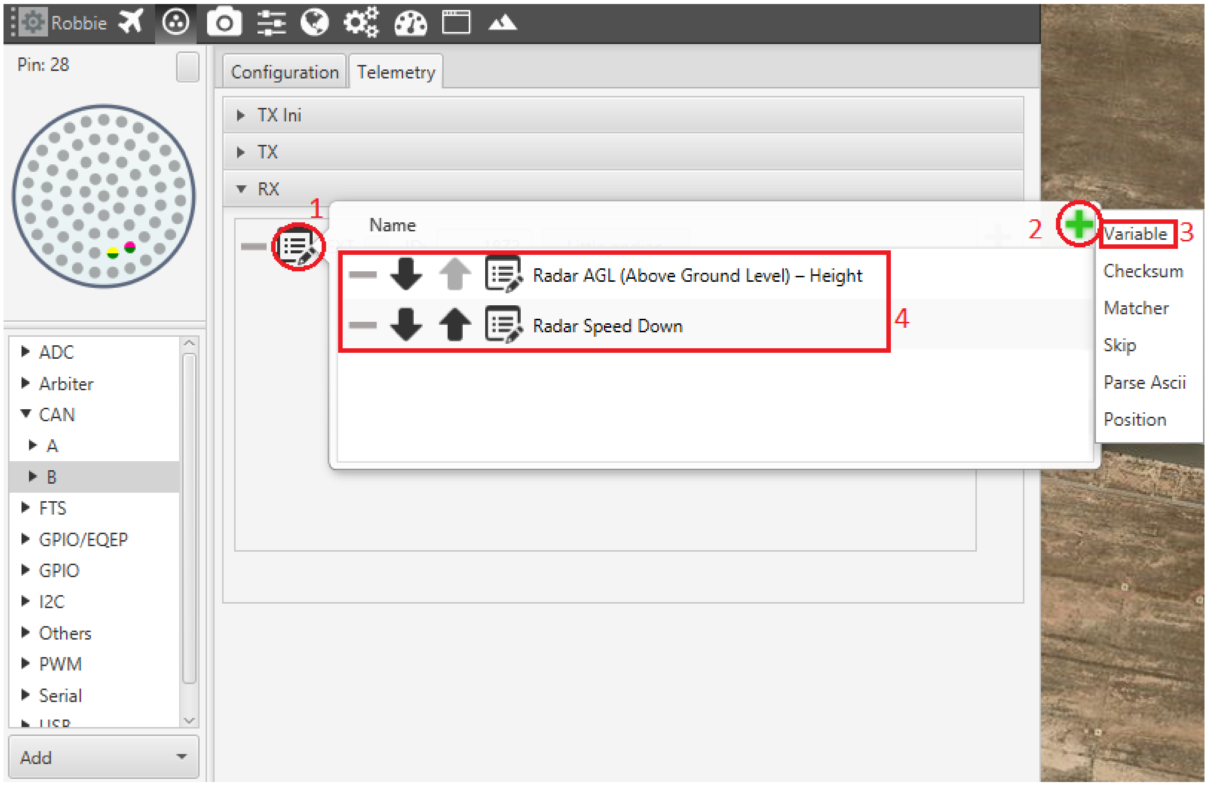

Robbie
Pin: 28
Configuration
Telemetry
Baudrate
1000000.0
Bd
1
Units ID
DEC
Units Mask
BIN
Mailboxes reserved RX 5
Mailboxes available TX 27
2
#
Mailboxes
Extended
ID
Mask
1
5
1872
0
3
ADC
Arbiter
CAN
A
B
FTS
GPIO/EQEP
GPIO
I2C
Others
PWM
Serial
Add

Step 3



Step 4

In **Telemetry** section you can add or delete messages clicking in "+" or "-"



Step 5

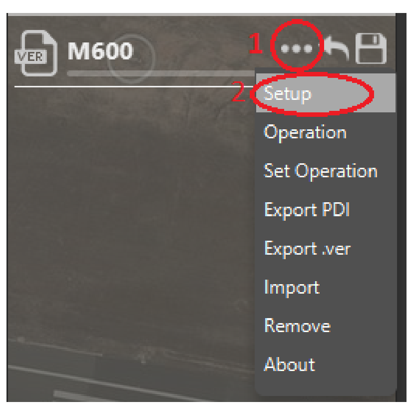
If you have followed the indications and introuced the values shown in previous figures, your radar-altimeter will be configured.

To get more information of the Smartmicro Radar - Altimeter datasheet you can go to:

- Micro Radar Altimeter User Manual
- Micro Radar Altimeter Data Sheet

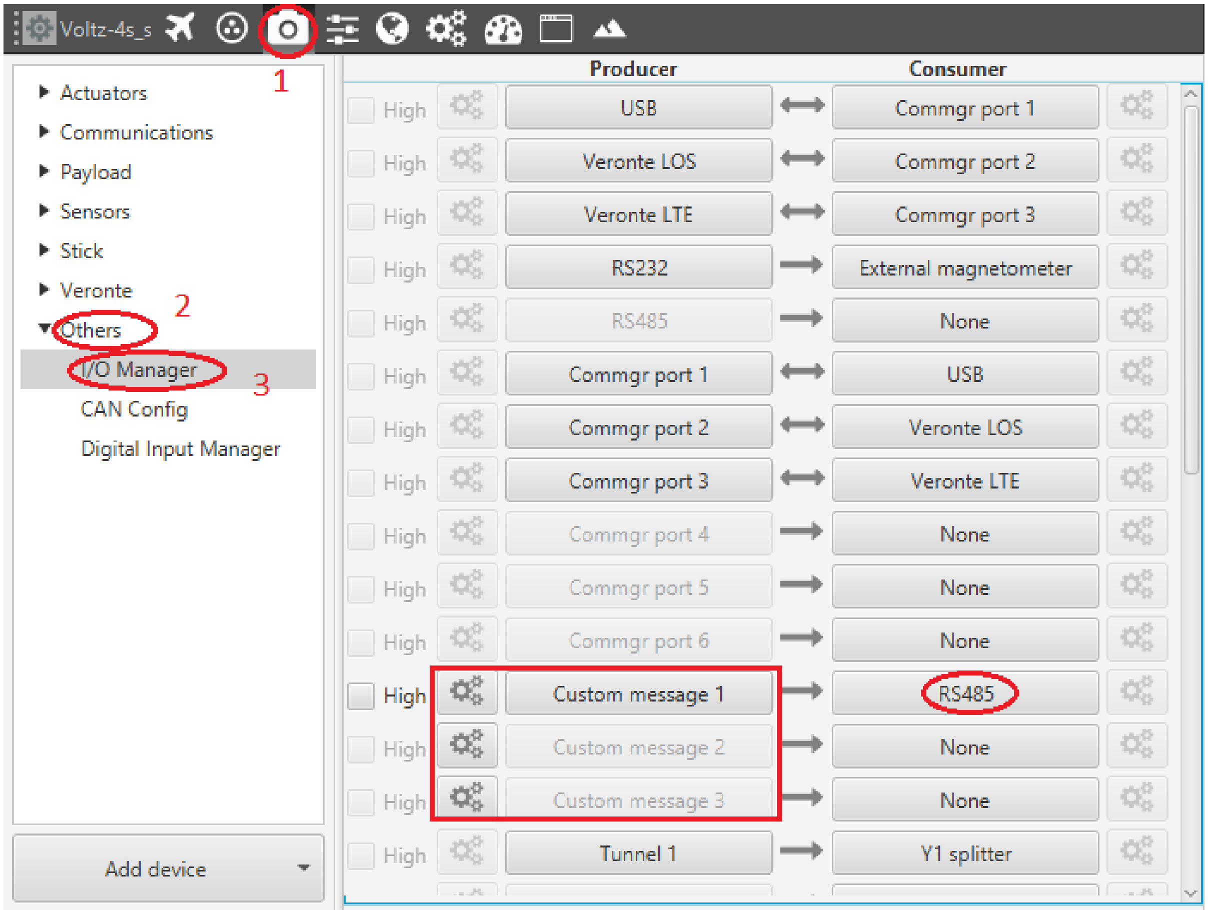
11.2.1.3 Volz Servo

We will configure this device in Pipe direcly. If we click in:



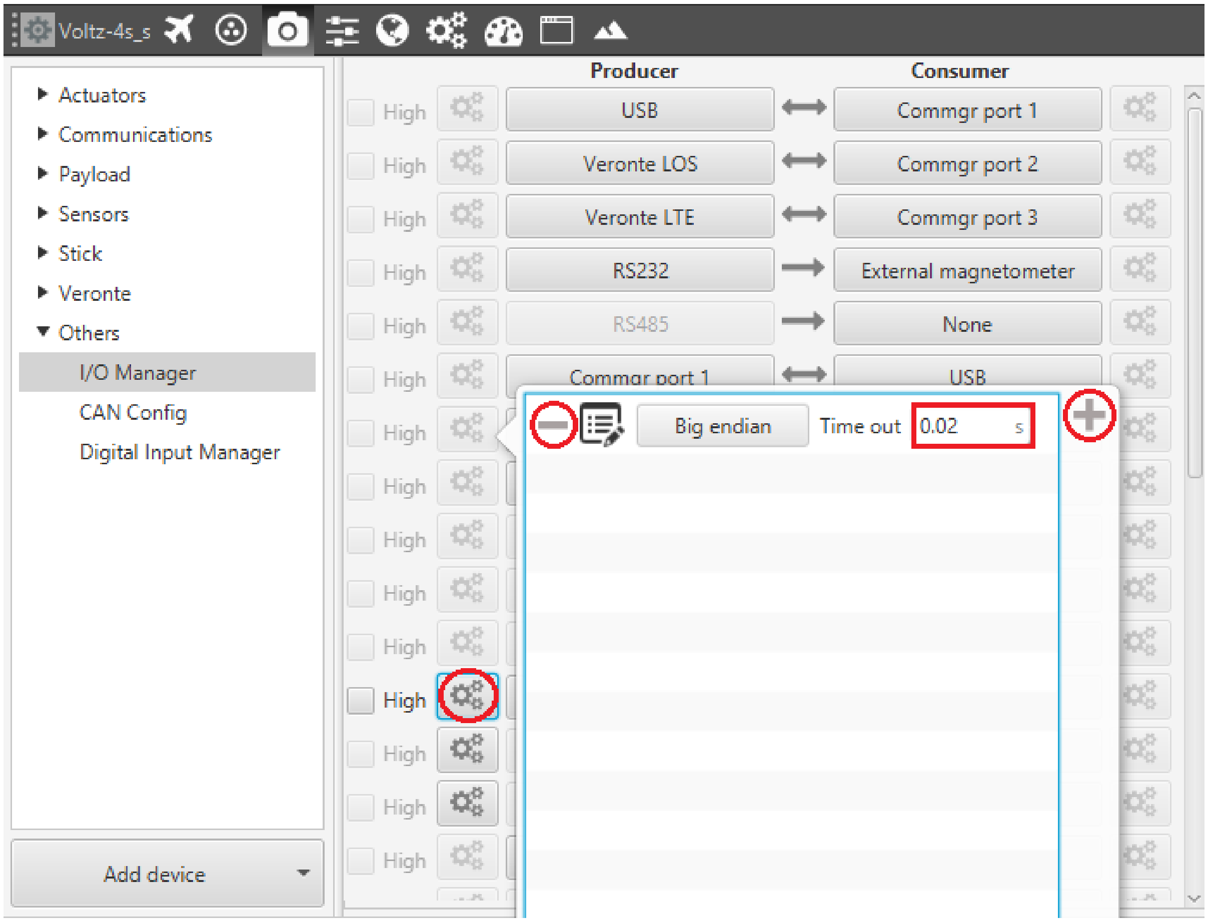
Step 1

Pipe will display you the Setup Toolbar (for more information see *Setup Toolbar*):



Step 2

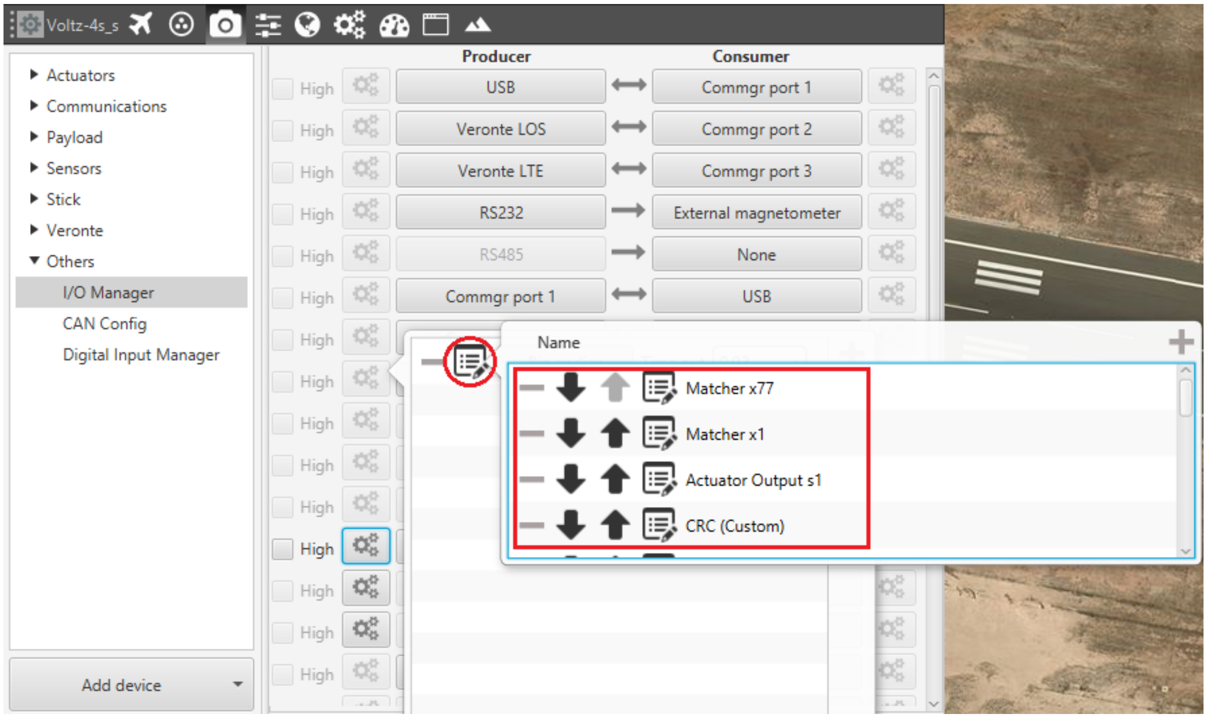
Following the steps you can access successfully in the I/O Manager menu, here you will find three diferents channels to use in your Volz Servo, you have to be sure that the consumer cell (right column) of your Custom message is configured with **RS485**.



Step 3

You can add or delete messages clicking in "+" or "-".

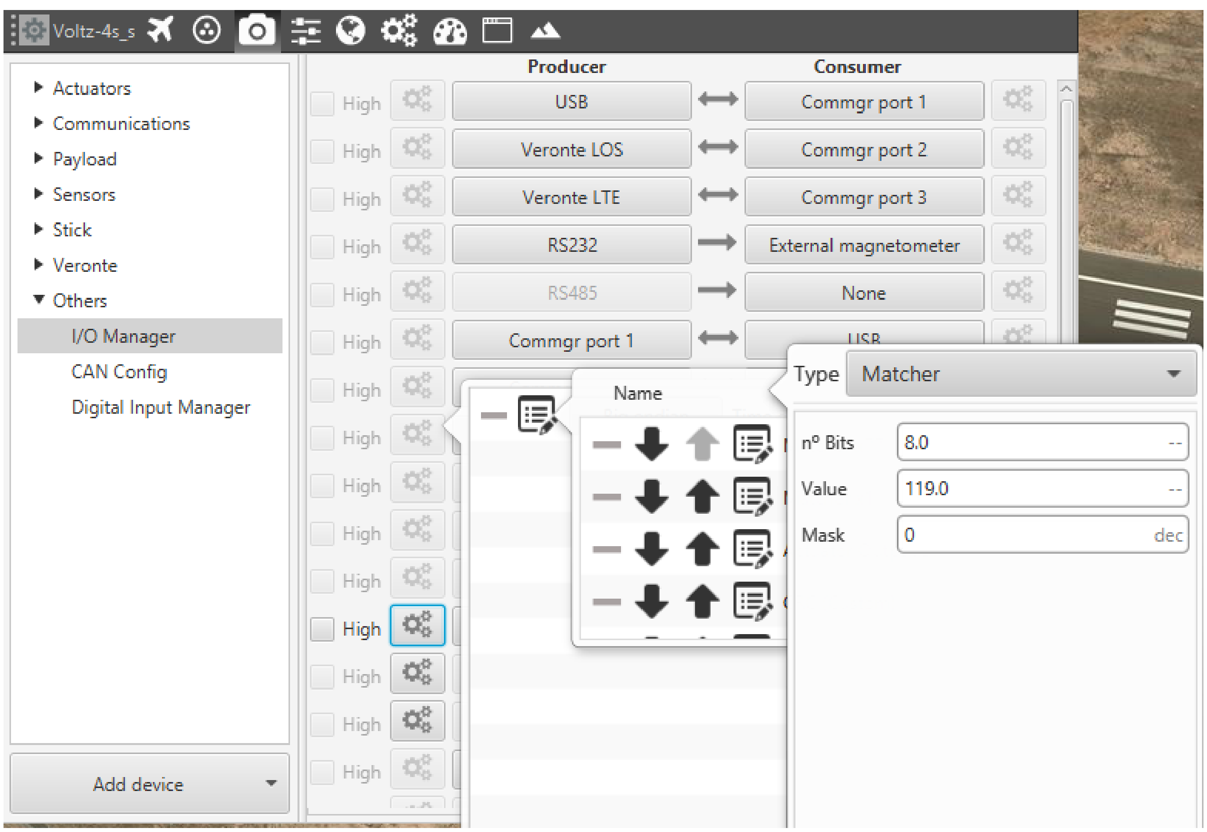
Time Out: frequency of the messages.



Step 4

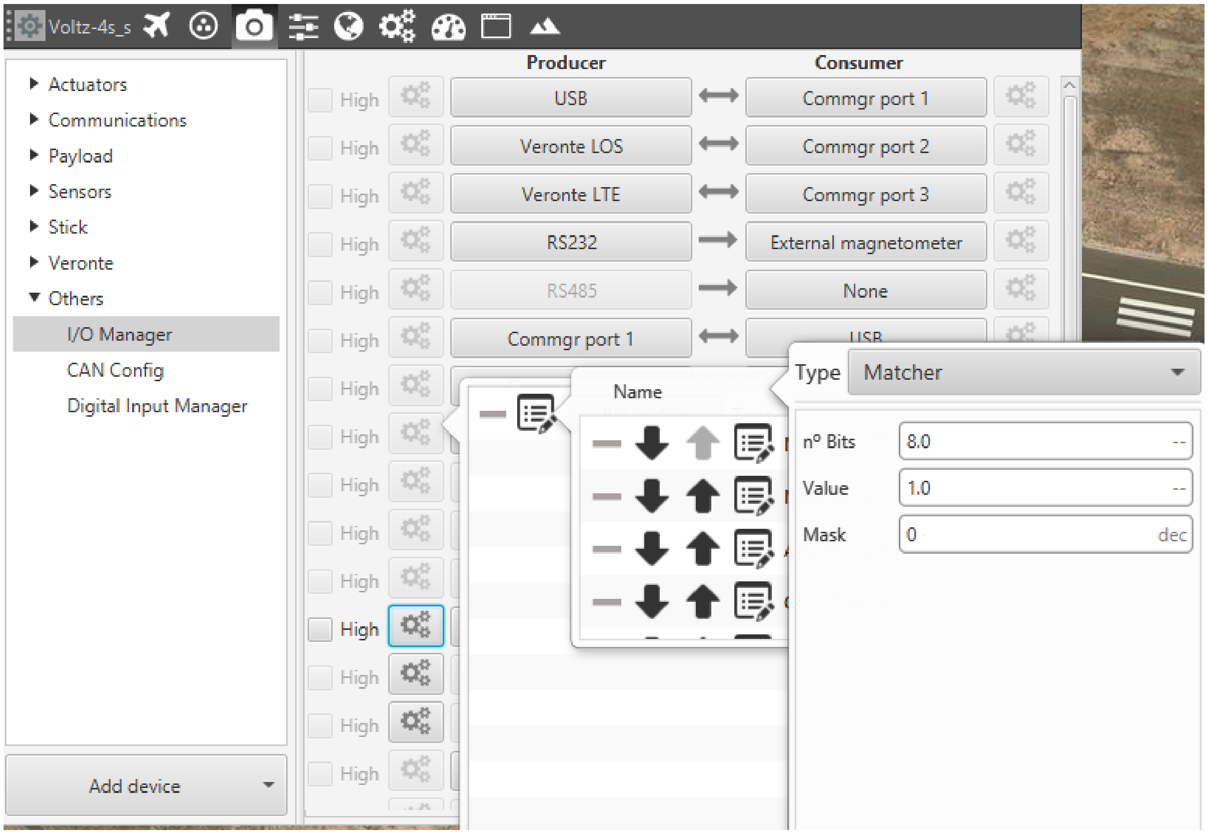
We differentiate four items:

- **Matcher x77**: activate the silence mode, servo do not reply when it receive the messsage.



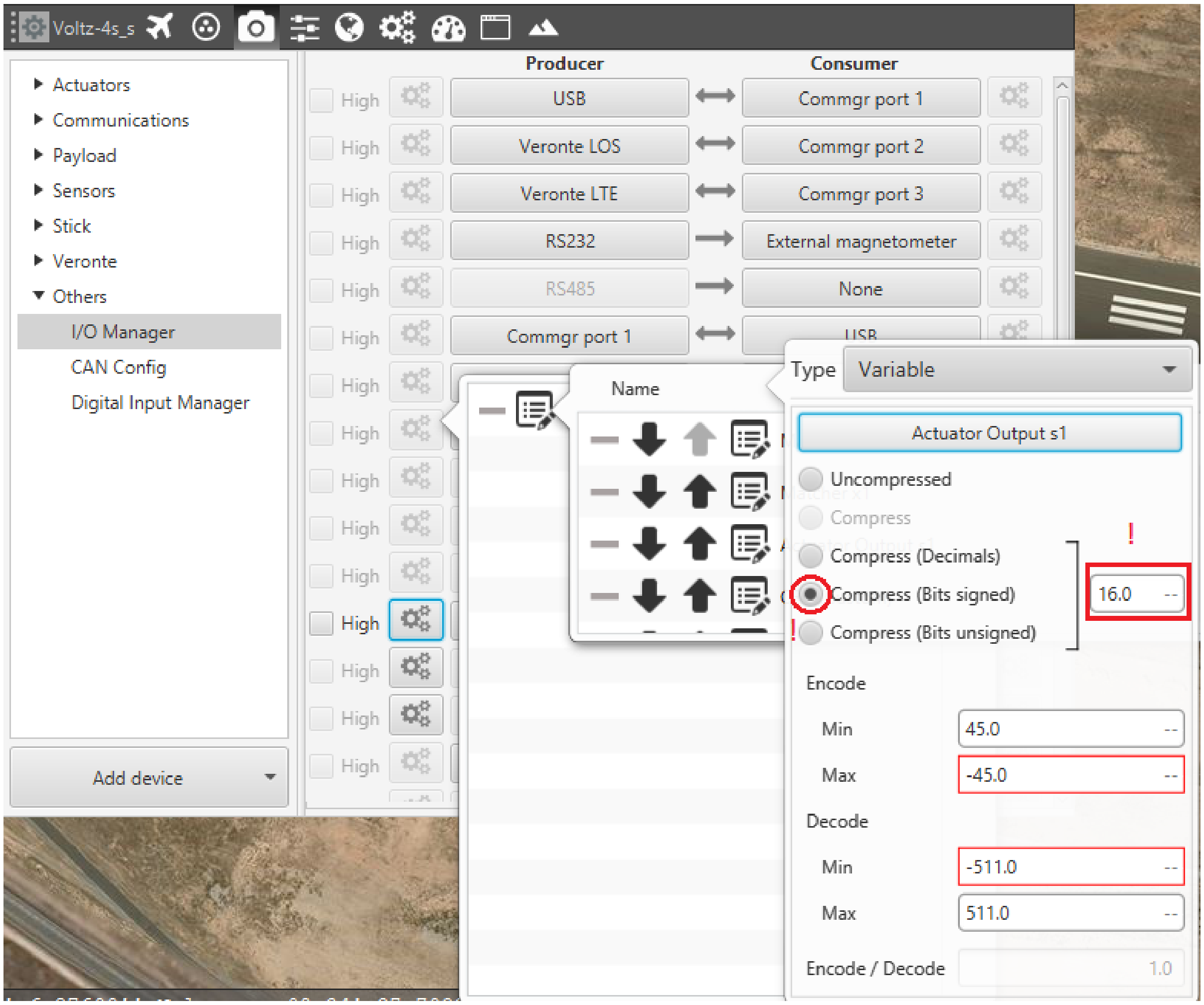
Step 5

- **Martcher x1**: it represents the identification of the servo.



Step 6

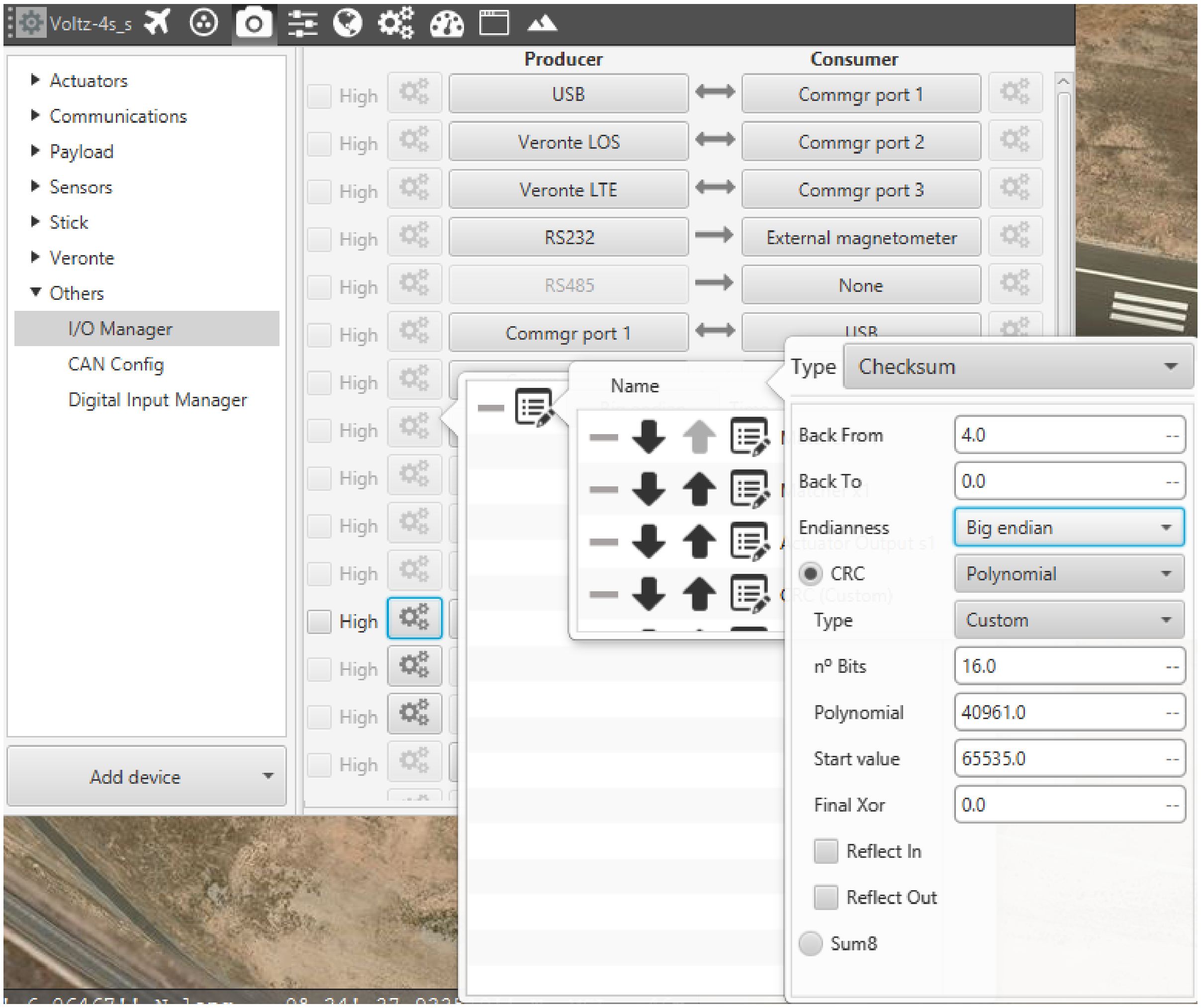
- **Actuator Output s1:** it references the output variable of the servo.



Step 7

Value **"16"** is very important to realice a correct configurtion.

- **CRC:** it is a parameter used by *Veronte* to check the consistency of the messages.



Step 8

This structure is repeated depending of the servos number.

If you have followed the indications and introuced the values shown in previous figures, your Volz Servo will be configured.

This section contains the whole configuration process of four different devices in Veronte Pipe. All the different menus that appear in the software will be covered so the final user of the system gains a fully understanding of how to totally configure its device trough some real examples.

The devices available are:

- Magnetometer P/N HMR2300-232 (Local P/N SESN007)
- Volz Servo Actuators
- Radar Altimeter P/N SESN001

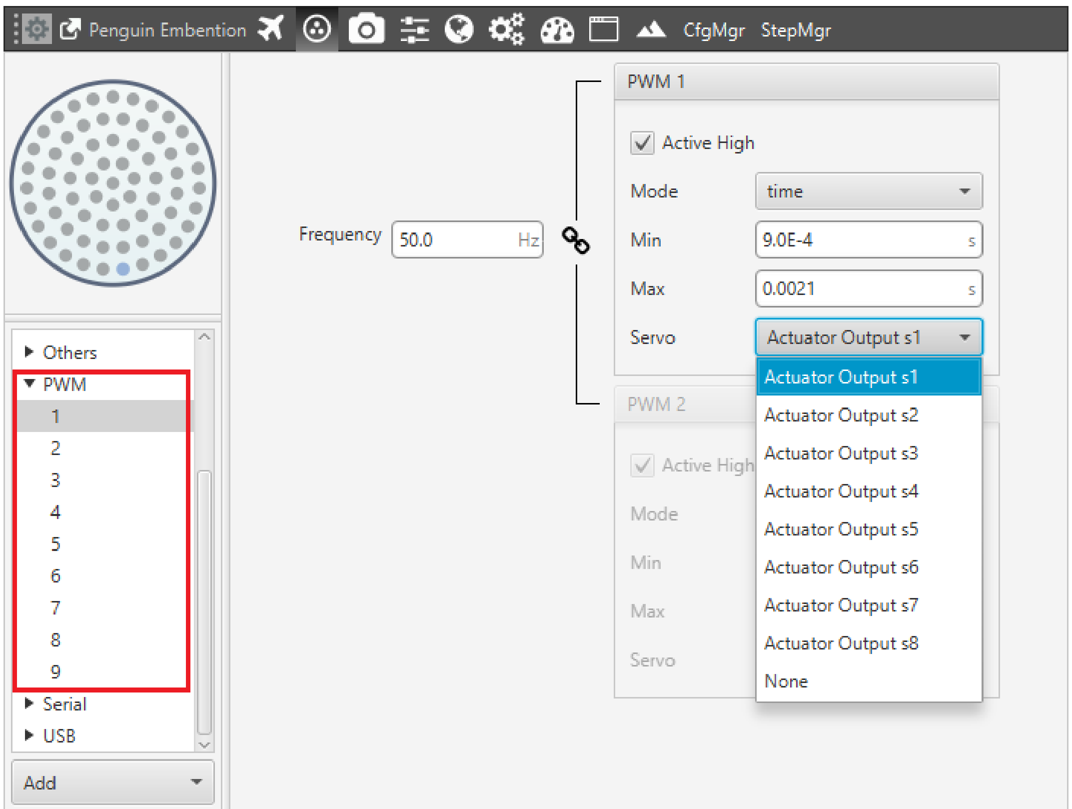
11.2.2 Plataform Configurations

11.2.2.1 Fixed Wing-Mentor

11.2.2.1.1 Servo Configuration

Once the Autopilot has been installed and connected according to the guidelines that appear in the section *Hardware Installation* , the first step of the configuration process is the adjustment and trimming of the servos. The next list contains the instructions to follow in order to trim the platform servos.

1. Set which servo number will have each one of the controls. **Mentor** has a conventional aircraft configuration, so there are five controls: elevator, two ailerons, rudder and throttle. Each one of these controls is assigned to a value of the "Actuator Output" vector (s) according to the pin where it is connected. It is possible to assign any value to any control, i.e the control connected to Pin1 can be s1, s2, s3, s4 or s5.



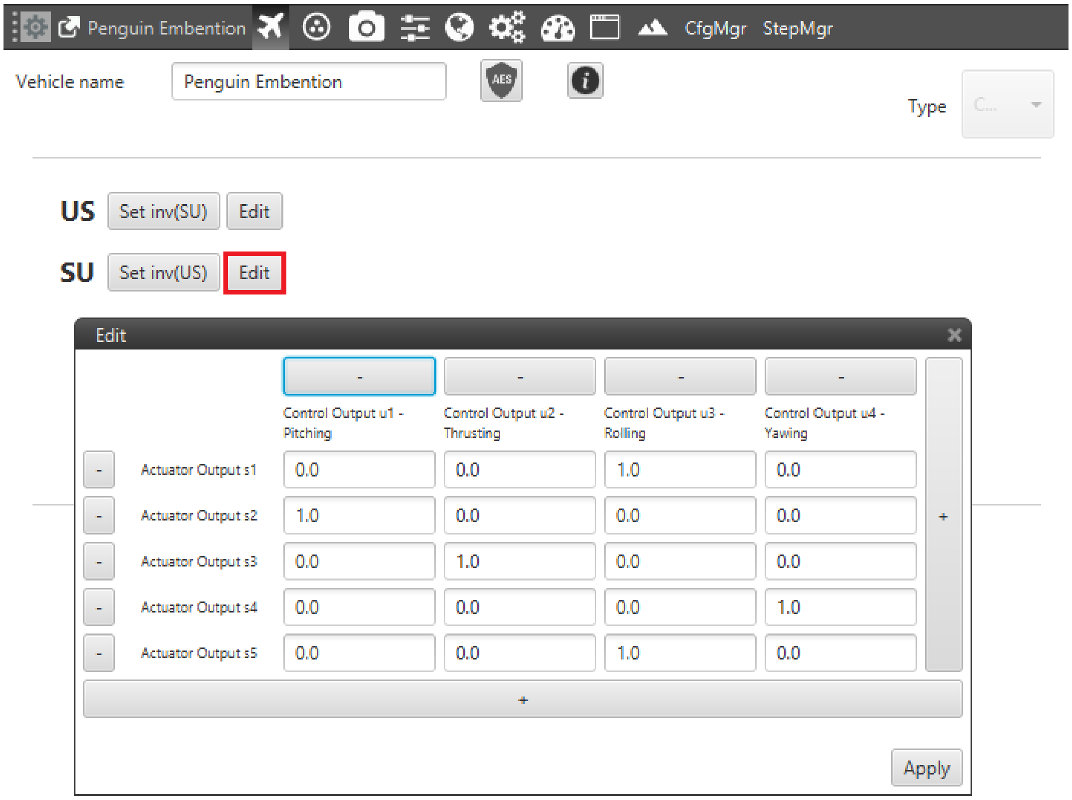
Actuator Output

- **Right Aileron:** Actuator Output s1
- **Elevator:** Actuator Output s2
- **Throttle:** Actuator Output s3
- **Rudder:** Actuator Output s4
- **Left Aileron:** Actuator Output s5

Note: Name **Actuator Output s** can be changed to other more identifiable according to the real actuator such as right or left aileron, see section *System Variables*.

2. With the five controls now having a value of the S vector, the **SU** matrix can now be edited. Here the user configures the relation between the controller outputs ("u") and the servo movements ("s"). In the case of

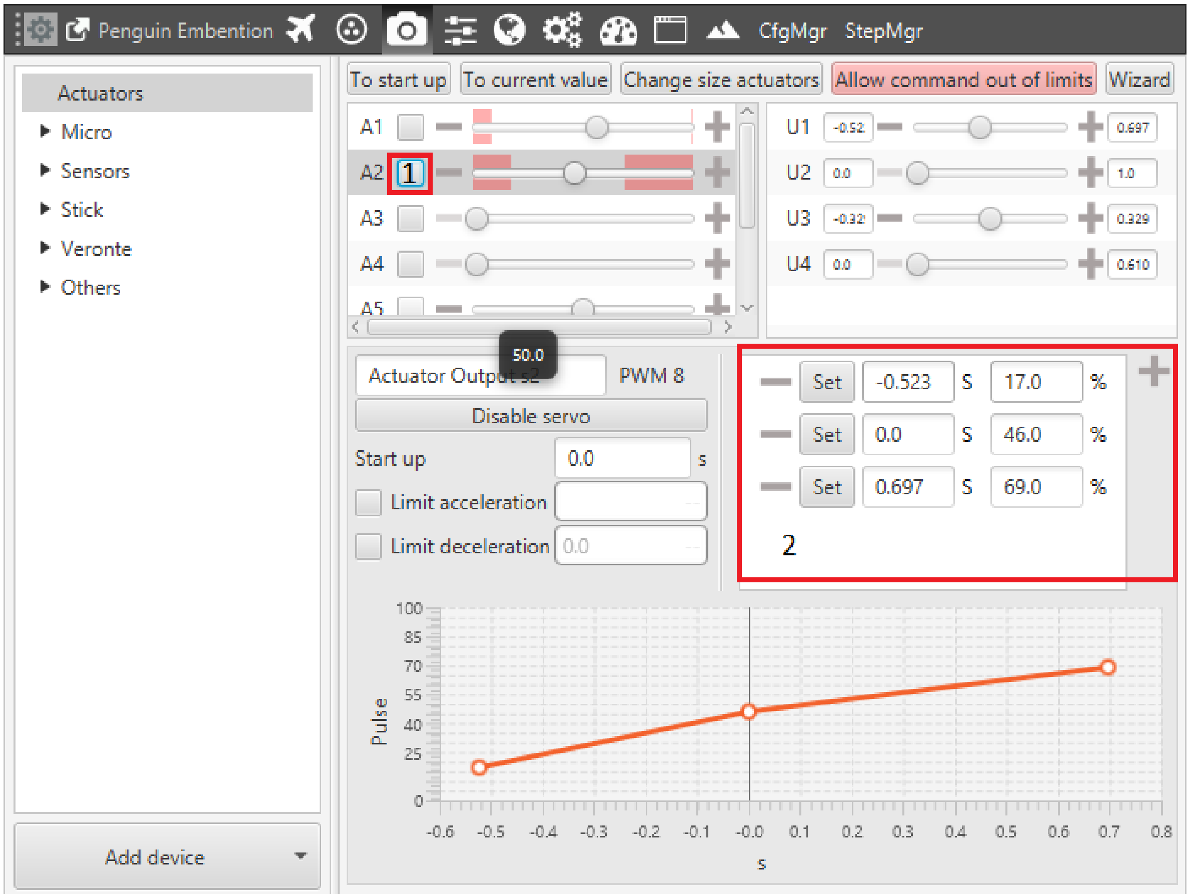
Mentor, each control channel is related with only one servo: the pitching with the elevator, the rolling with the ailerons, the yawing with the rudder and the thrusting with the throttle, so the **SU** matrix will be as follows, taking into account which s corresponds which control according to the connections.



SU Matrix Configuration

- **Pitching (u1):** Actuator Output s2 (Elevator)
- **Thrusting (u2):** Actuator Output s3 (Throttle)
- **Rolling (u3):** Actuator Output s1 and s5 (Ailerons)
- **Yawing (u4):** Actuator Output s4 (Rudder)

3. Finally, the servo position has to be linked with a real variable, for example, the angle of the control surface or the throttle level. Besides, here are set the physical limits of the servo in the case when it is not possible for it to move in the full range (0-100%).



Servos Configuration

Servos can be moved from the actuators menu when being in INI phase (no phase in green in Veronte Panel).

- When **1** is marked, the slide bar allows the movement of that servo.
- The curve points are added in **2**.

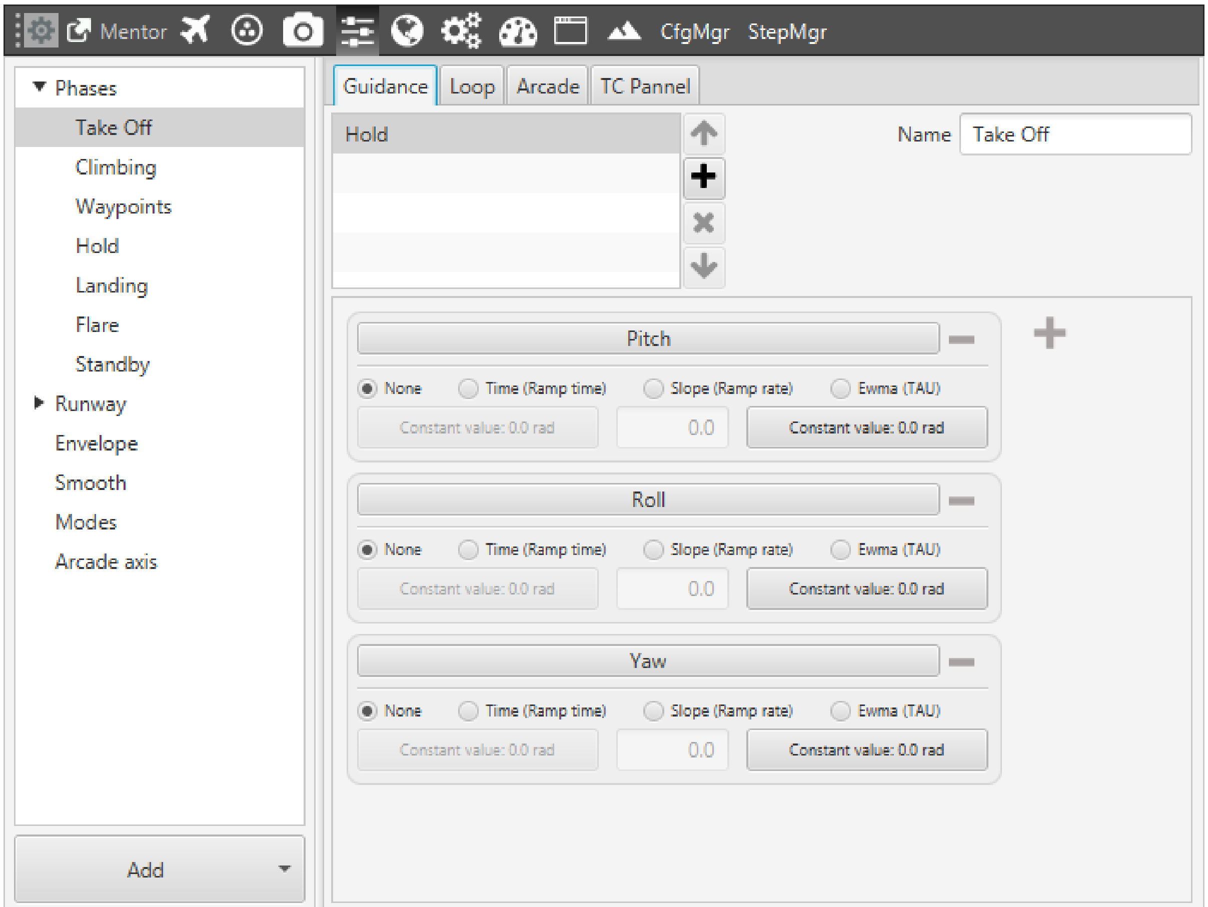
Let's consider the elevator (s2,A2) as an example. In Mentor, the elevator can not be moved totally down because it hits the landing gear, and neither totally up because it hits the rudder. Taking note of the pulse value at this points, the limits of the servo are obtained: 17% and 69% in the example of the previous figure. Measuring also the surface deflection at those points (-0.523 and 0.697 radians) the servo curve is totally defined (including also the point where the deflection is zero).

11.2.2.1.2 Mission Phases

In this section, it will be explained how to configure a whole mission for an airplane, in this case, **Mentor**. The typical phases will be detailed and the guidance for each one of them will be presented.

11.2.2.1.2.1 Takeoff

The takeoff phase will be the one where the aircraft goes from the initial to the lift off point.

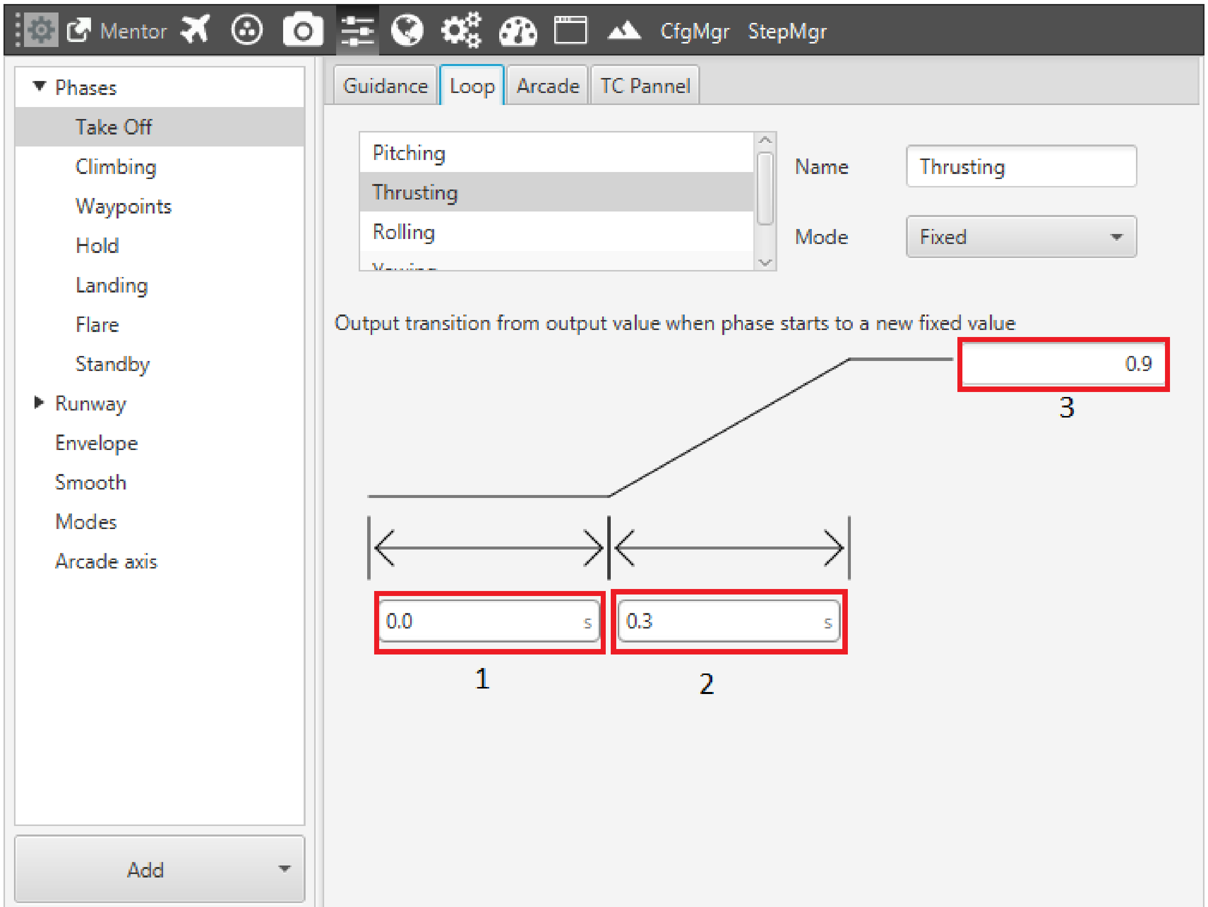


Take Off Phase – Guidance

The guidance for this phase will be a **Hold** of three variables:

- **Pitch and Roll Angle:** kept at 0 [rad].
- **Yaw Angle:** kept at the value that the aircraft has when is set on the runway i.e, the variable selected will be *Yaw*.

Regarding the control loop, there are PIDs in the pitch, roll and yaw control channels, while the throttle has a ramp as defined in the following figure.

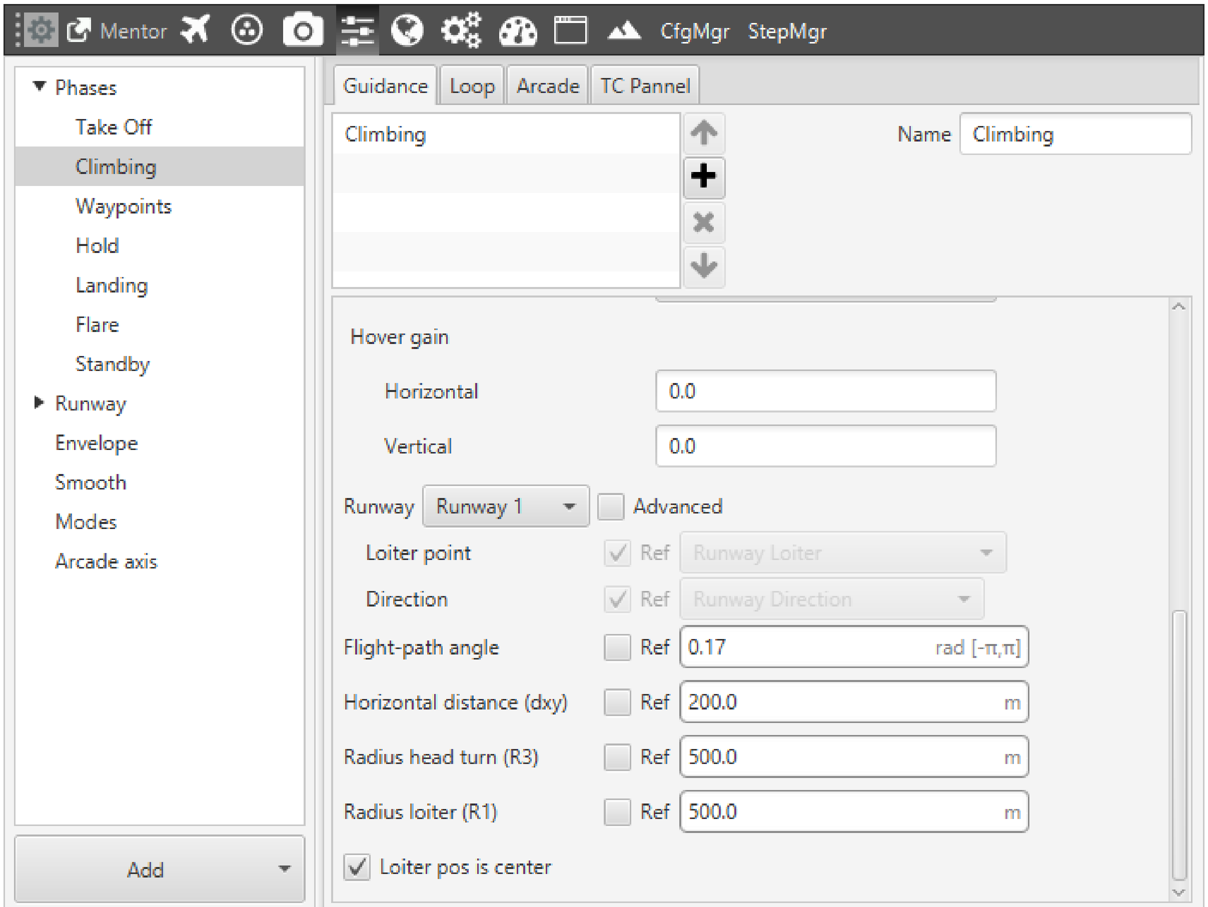


Take Off Phase – Loop

1. Here the initial value is set, in this case 0 because the aircraft starts from a standing point.
2. Time of transtion, 0.3 seconds, the throttle is increased from the initial to the final value.
3. Final value, 0.9 .

11.2.2.1.2.2 Climbing

The climbing phase is used to make the airplane reach the cruise altitude after the takeoff.



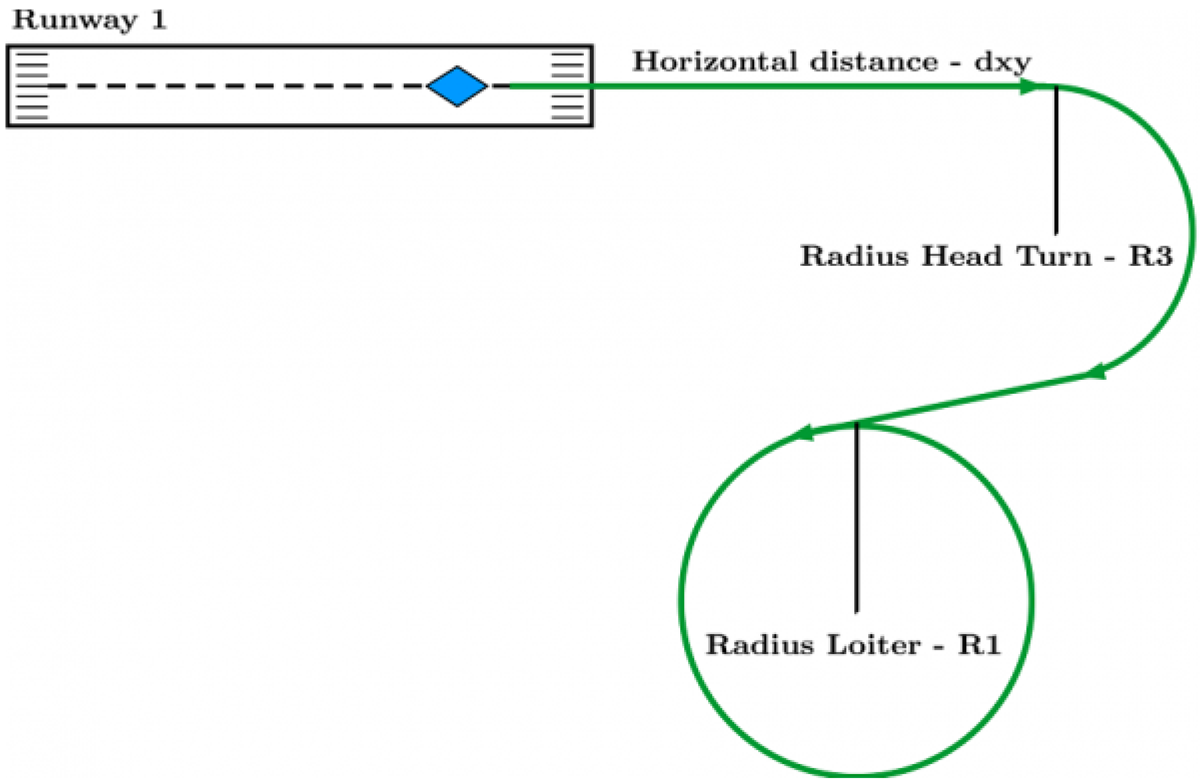
Climbing Phase – Guidance

This fase consists of Climbing Guidance:

- **Line attraction:** value related to how strongly the aircraft tries to reach a path (climb path in this case). This value is commonly between 20 and 40 for airplanes.
- **Set speed:** speed that will have the airplane during the climb, 15 [m/s] .

In this phase, the aircraft reaches the mission altitude by performing a spiral route. In order to determine the correct path, these parameters must be set:

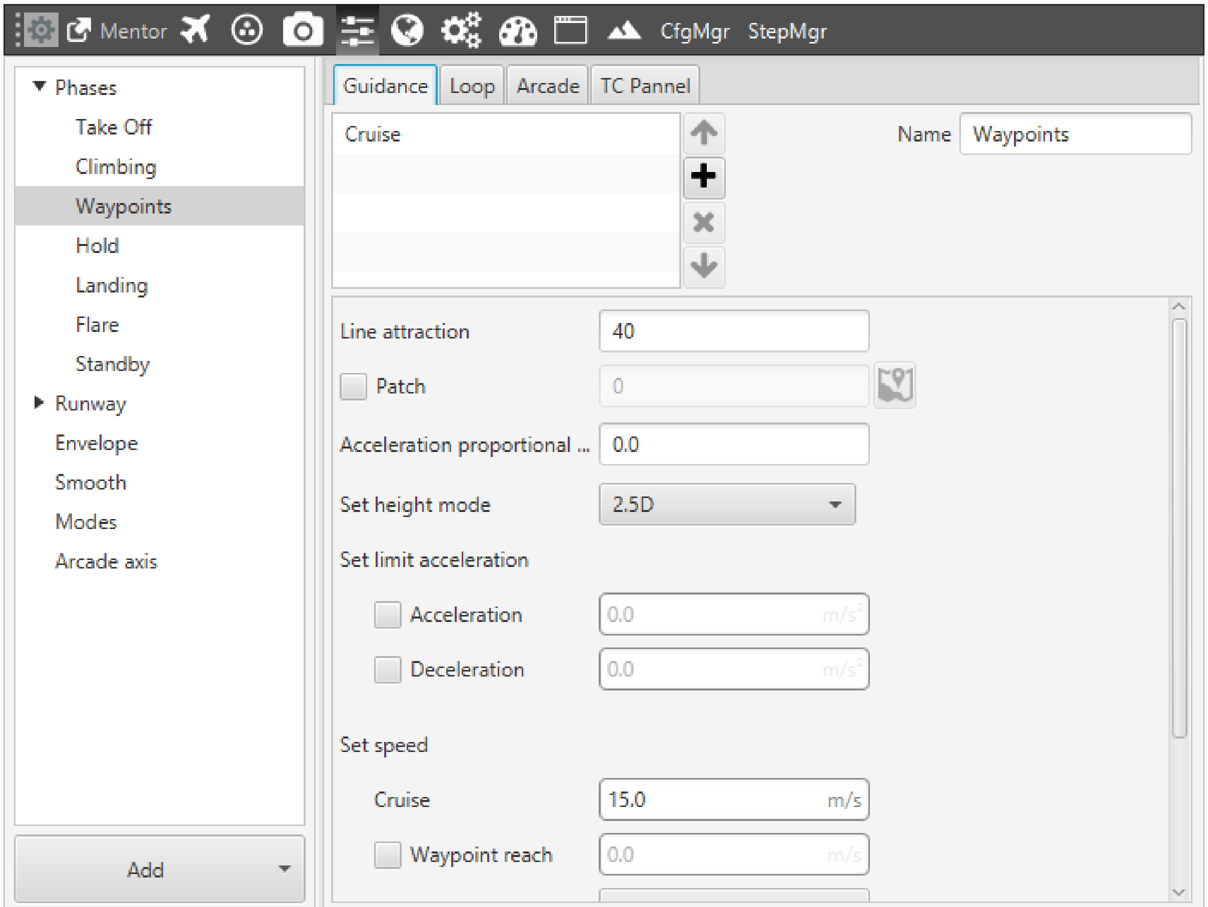
- **Runway:** here is selected the runway which previously has been edited in its configuration menu.
- **Flight Path Angle:** angle at which the aircraft will climb, 0.17.
- **Horizontal Distance:** is the distance from the point where the aircraft enters in the phase which contains the climbing guidance, to the start of the circular climbing path, 200 [m].
- **Radius Head Turn R3:** radius of the turn made to head the airplane towards the loiter direction, 500 [m].
- **Radius Loiter R1:** radius of the loiter ascending made by the aircraft to reach an altitude suitable, 500 [m].



Climbing Path

11.2.2.1.2.3 Cruise

The cruise phase is where the aircraft follows a route marked by a set of waypoints, which are defined by the user in the **Mission** menu.



Cruise Phase – Guidance

This fase consists of a Cruise Guidance. Parameters set here are the same as the ones from the climbing phase.

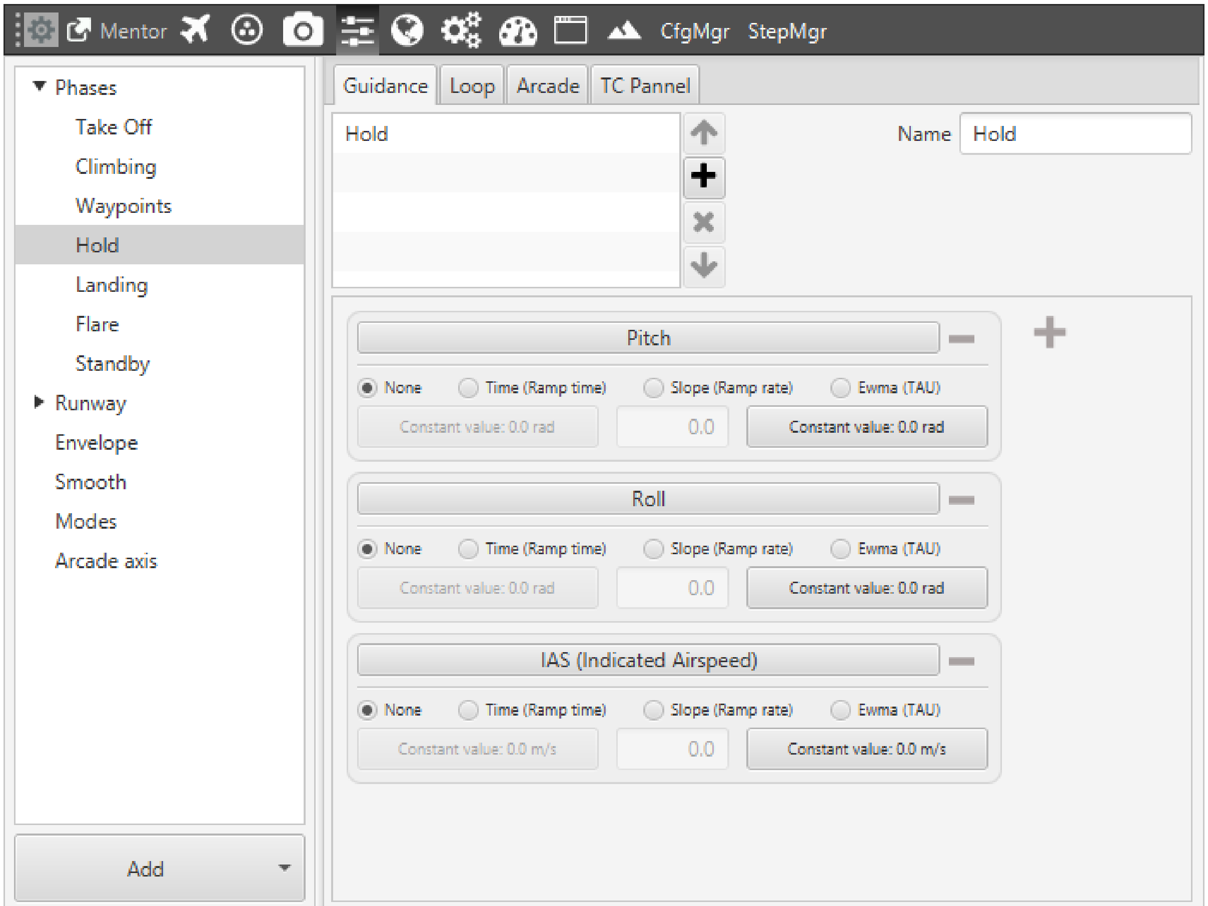
- **Line attraction:** 40.
- **Set Speed:** 15 [m/s].

In this guidance, there is an option related to the gains used to recover the hover point in an multicopter, **Hover Gain**. In this case, the platform configured is an airplane, so this option will not be used.

Warning: When using the Cruise phase, the aircraft will automatically follow the waypoint route. An automation has to be created to make the platform perform in a different way.

11.2.2.1.2.4 Hold

This phase is used to keep the aircraft at a constant heading, for example, when the radio connection is lost.



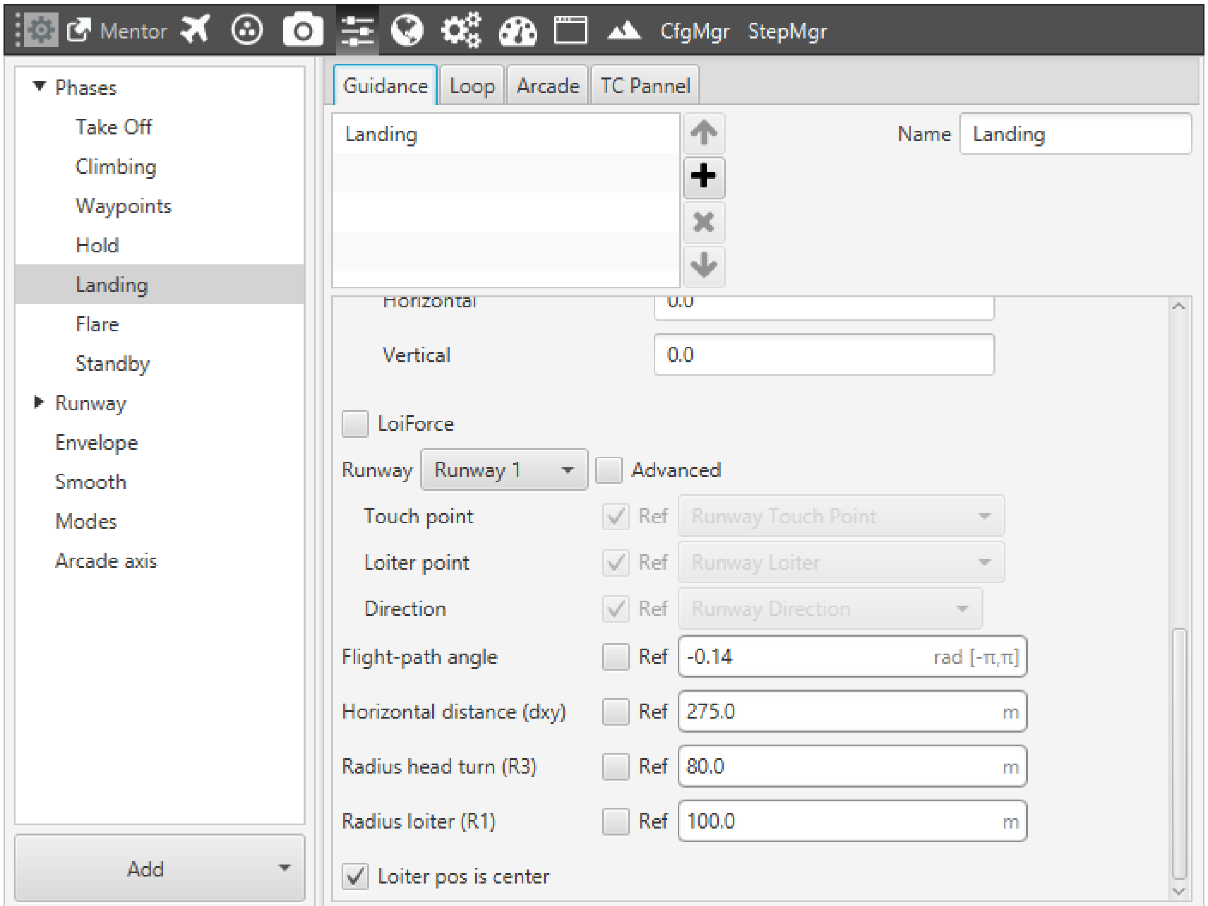
Hold Phase – Guidance

When this phase is active, the aircraft flies in a straight line, until another phase is commanded.

- **Pitch, Roll and IAS:** kept at a constant value 0.

11.2.2.1.2.5 Landing

This phase is used to make the aircraft land at a certain airport. When the flight altitude is too big, this phase contains the parameters which define the route performed by the platform to descent until an altitude where it can line up with the runway.



Landing Phase – Guidance

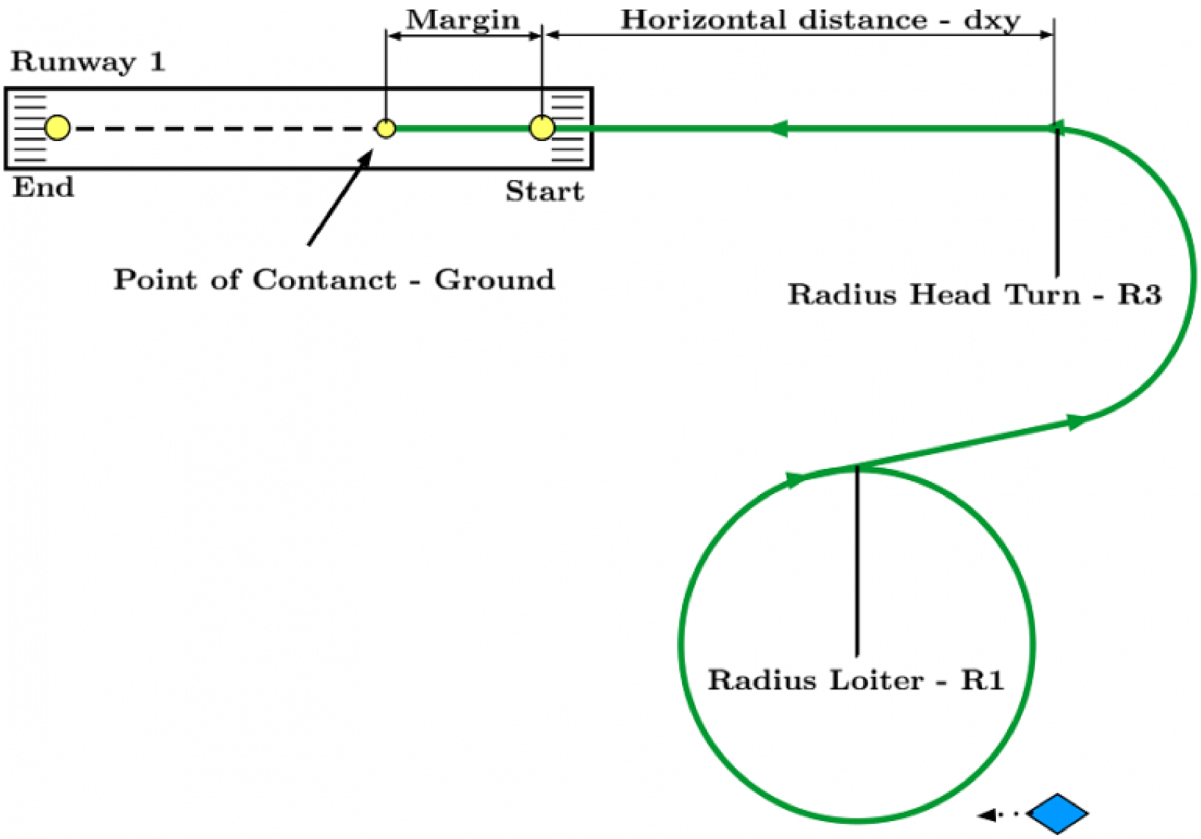
The guidance to be configured is Landing:

- **Line Attraction:** 20.
- **Set Speed–Cruise:** 14 [m/s].

Finally, the following parameters define the path during this phase.

- **Runway:** here is selected the runway which previously has been edited in its configuration menu.
- **Flight Path Angle:** angle at which the aircraft will descend, -0.14 [rad].
- **Horizontal Distance:** is the distance from the point where the aircraft enters in the phase which contains the climbing guidance, to the start of the circular climbing path, 275 [m].
- **Radius Head Turn R3:** radius of the turn made to head the airplane towards the runway direction, 80 [m].
- **Radius loiter R1:** radius of the loiter descending made by the aircraft to reach an altitude suitable to perform the landing manoeuvre, 100 [m].

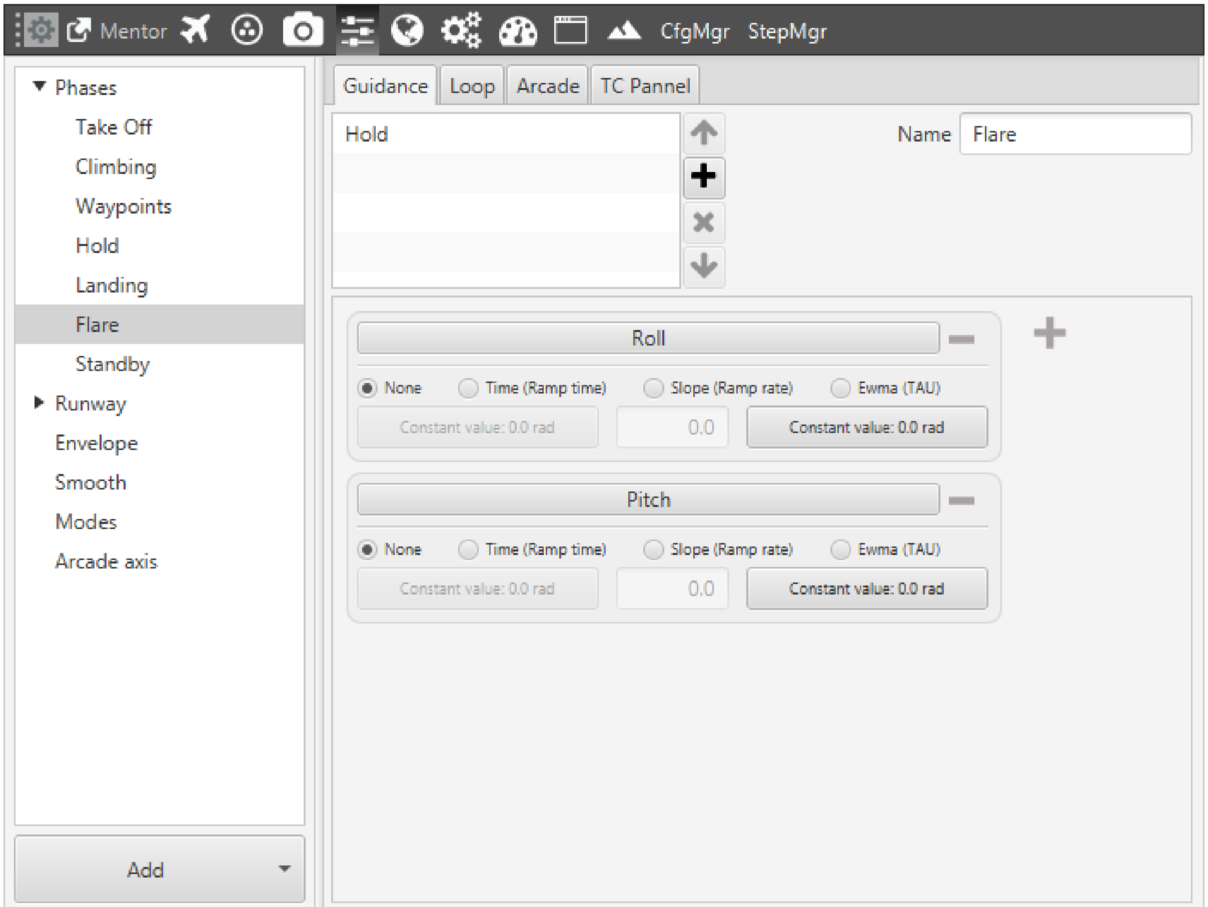
The route of this phase is shown in the following figure, where each one of the parameters that define it are defined.



Landing Route

11.2.2.1.2.6 Flare

The flare is a maneuver made by an aircraft just before the touchdown. Consists on a rise of the nose to decelerate the descent rate and set a proper attitude before touching the ground.



Flare Phase – Guidance

Considering what is wanted in this phase, the guidance to command is a **Hold**.

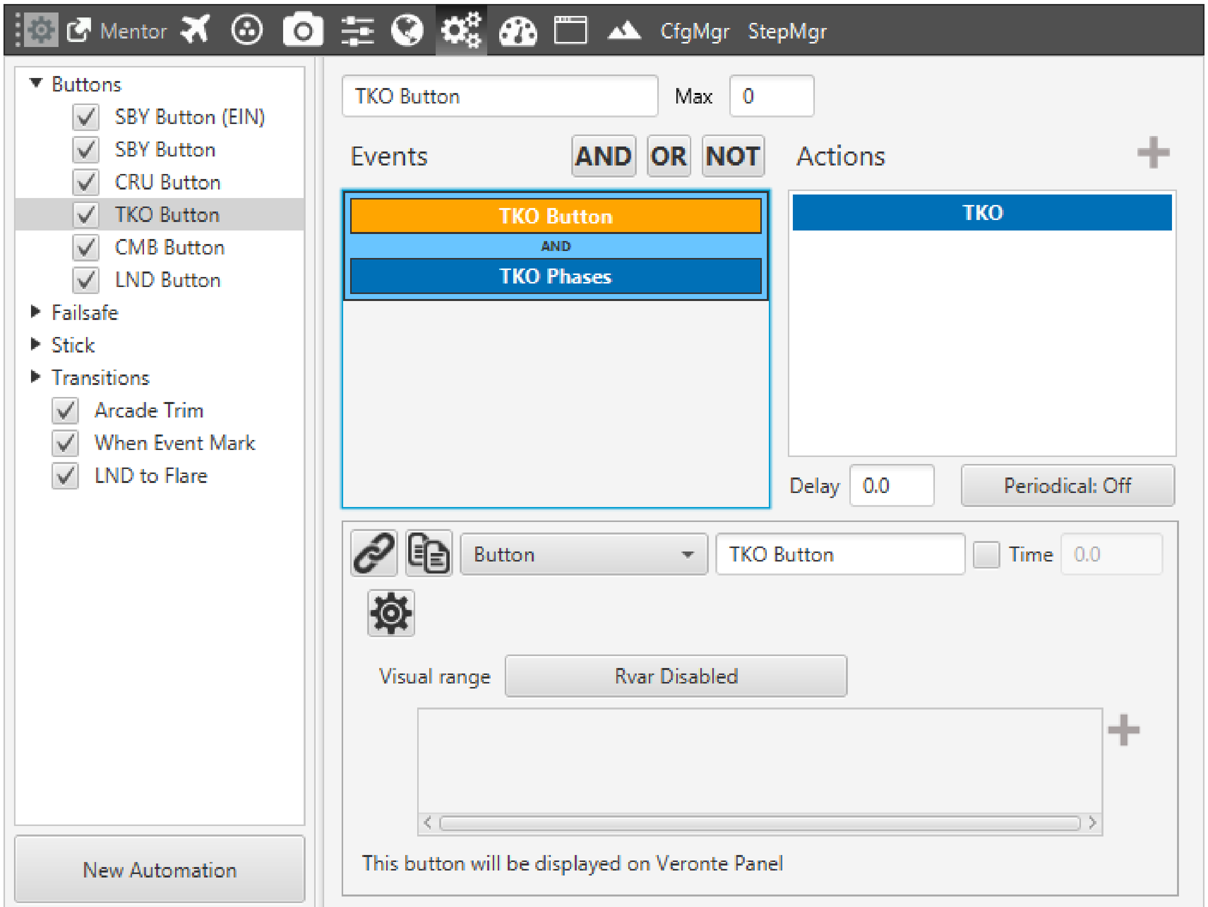
- **Roll – Pitch Angle:** kept at a constant value 0 [rad].

Regarding the thrust in this phase, the engines are shut off, so the mode of the controller is **off**.

11.2.2.1.3 Automations

Automations are the mechanisms used to perform an action when some event is triggered. These actions could be a change from one phase to another, taking a photo, dropping a payload and so on. In this section, the conventional automations used in a flight of the Mentor airplane will be detailed.

The first automations to be created are the ones linked to the buttons of Veronte Panel. When clicking one of these buttons the phase is changed to the one shown on the button label.



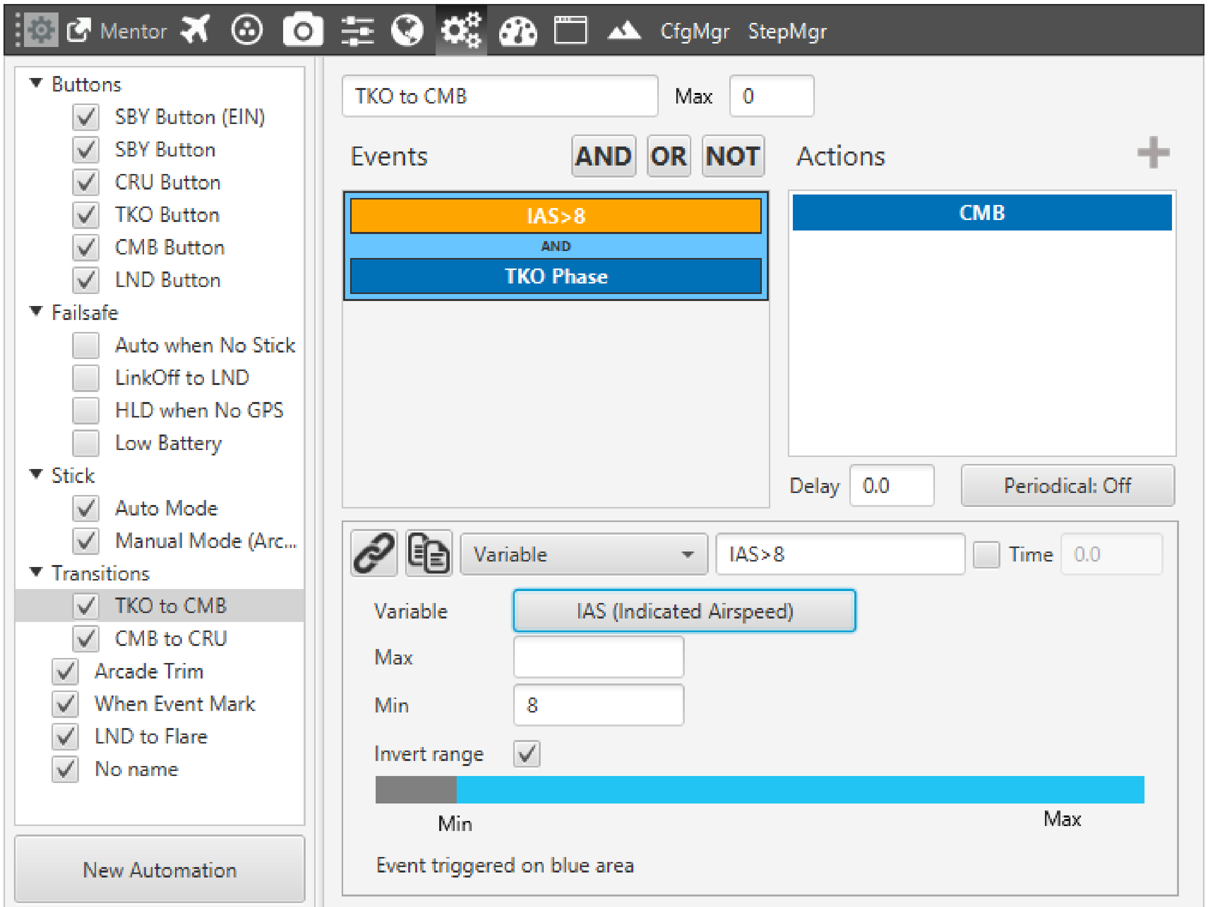
Automation Panel

Normally, besides the button event, a safety condition is added to the automation, which consists on letting only the system to change to a phase when being in a certain set of phases. For example, only change to cruise when being in climbing or change to landing when being in cruise or climbing. It is only necessary to create an AND condition with the phases to change from. In the previous figure, the automation is the change to takeoff when clicking on Veronte panel, being on Standby. This process is repeated for the rest of phases.

11.2.2.1.3.1 Takeoff to Climb

The change from the take off to the climbing phase occurs when the IAS of Mentor on the runway is greater than 8 m/s. There are two events to be configured:

- **Variable:** IAS > 8 [m/s].
- **Phases:** Standby.



Take off to Climb – Automation

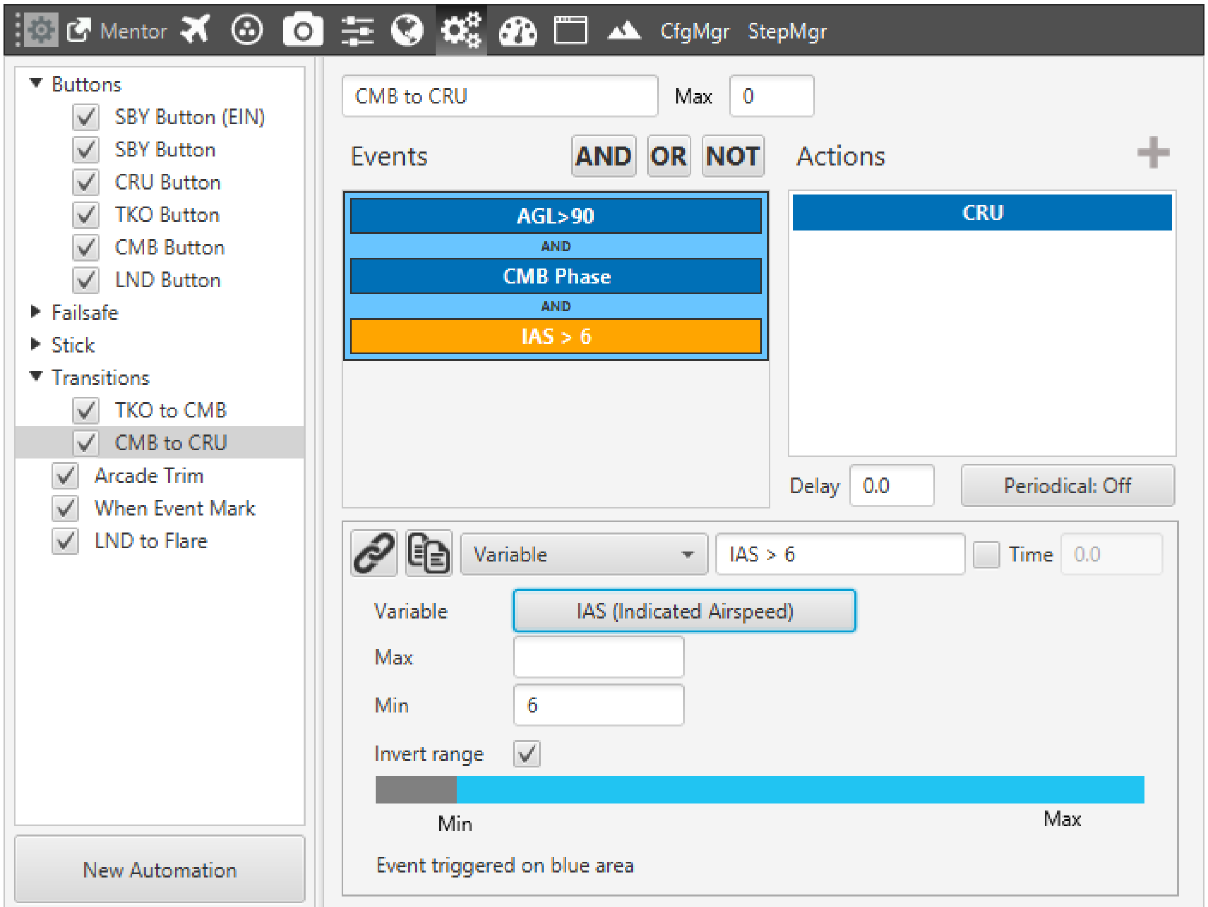
The action will be a change of Phase:

- **Phase:** Climbing.

11.2.2.1.3.2 Climbing to Cruise

When a certain altitude is reached, Mentor changes to the Cruise phase where it starts to follow the path determined by the user. As a safety condition, the change will only happen when having a speed greater than 6 meters per second. There are three events to be configured:

- **Variable:** AGL > 90 [m].
- **Variable:** IAS > 6 [m/s].
- **Phases:** Climbing.



to Cruise - Automation

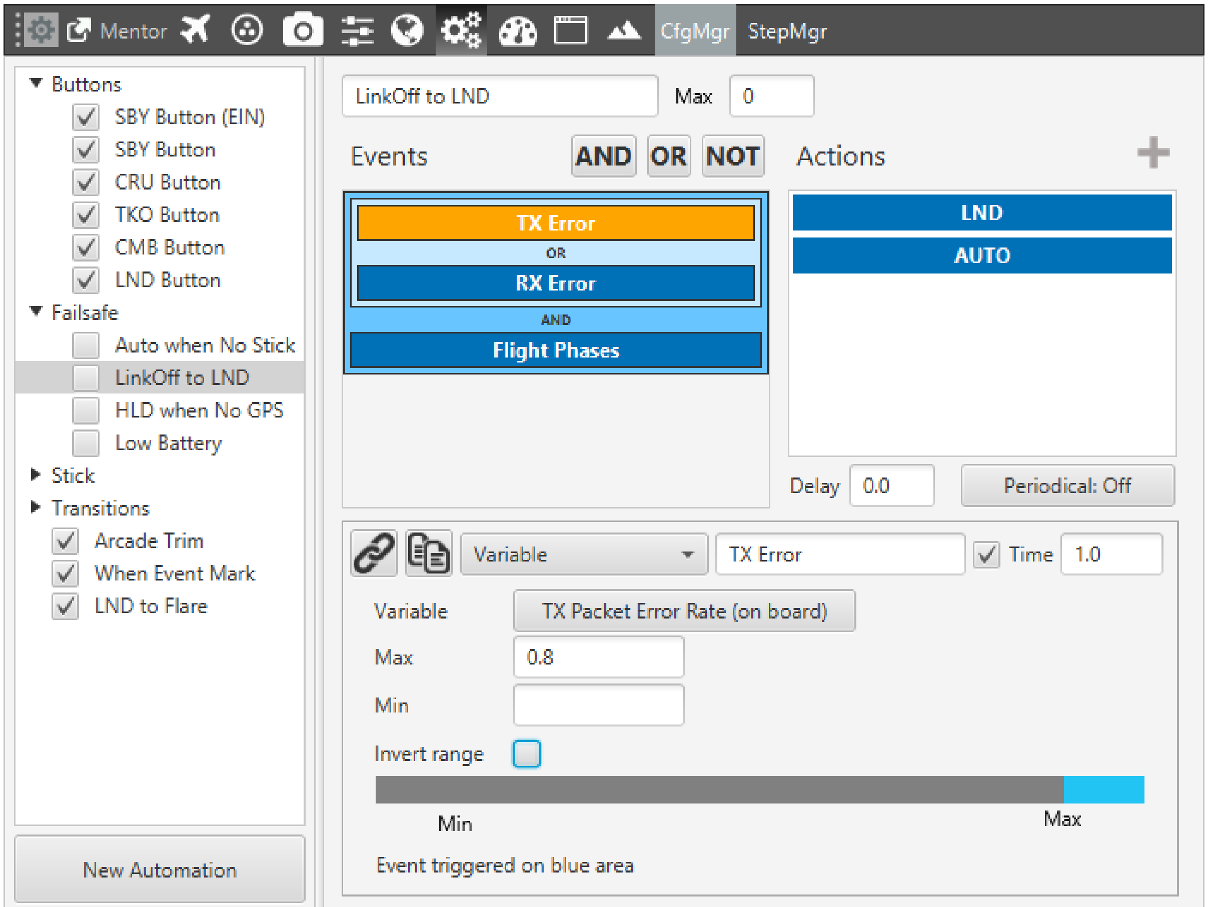
Finally, the action will be a change of Phase:

- **Phase:** Cruise.

11.2.2.1.3.3 Radio Error

When the radio connection from the ground station to Mentor is lost, the aircraft is forced to change to auto mode and land. It is necessary to configure two events:

- **Variable:** TX or RX error (TX /RX Packet Error Rate (on board)), greater than 0.
- **Phases:** Climbing or Cruise.



Radio Error – Automation

Two actions are configured:

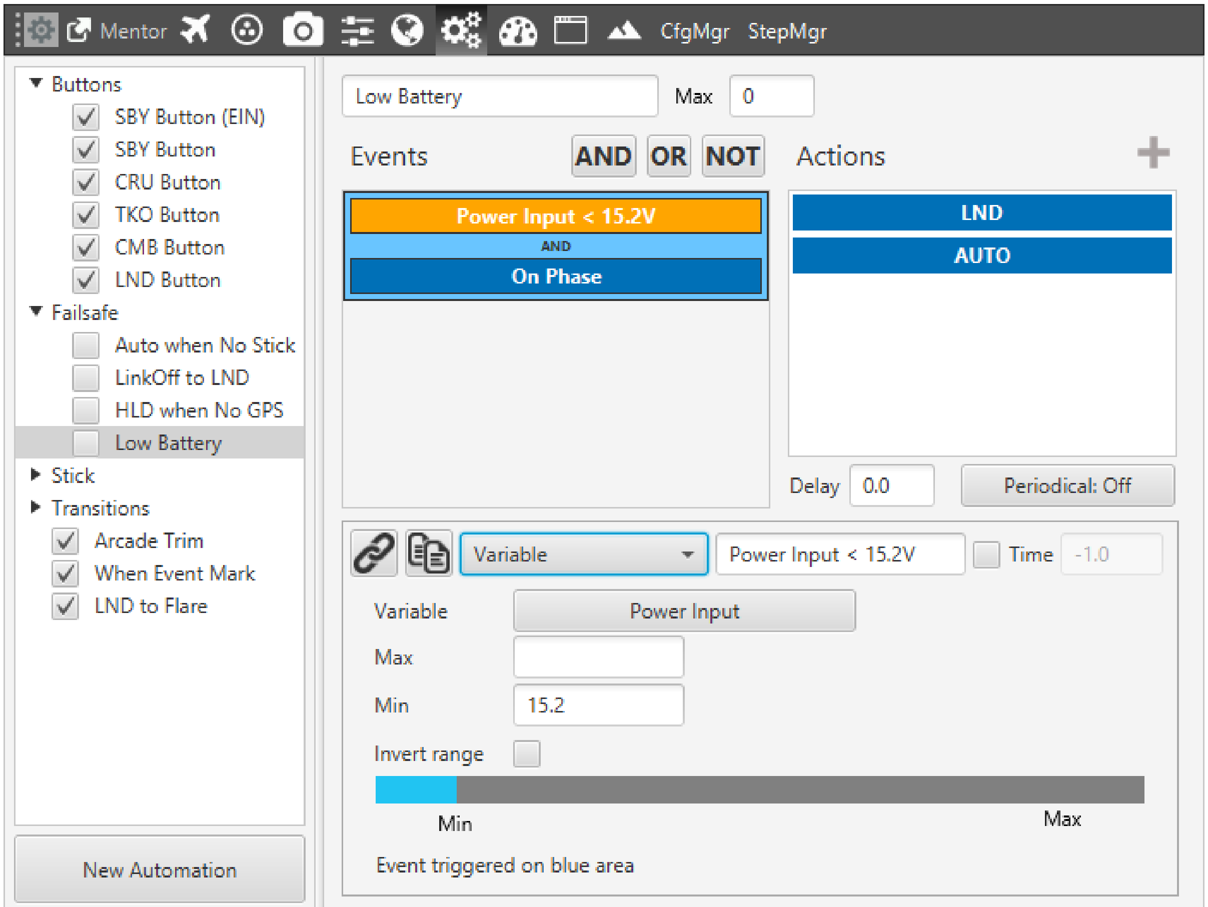
- **Phase:** Landing.
- **Mode:** Auto.

When the value of the flag that evaluates the TX or RX error ("TX Packet Error Rate (on board)") has a value greater than 0.8, being the aircraft in climb or cruise phases, it is forced to land. The value 0.8 means that 80 percent of the packets send trough the radio link have been lost.

11.2.2.1.3.4 Low Battery

When the battery is below a certain level (15.2 Volts in this case), and the aircraft is in climb or cruise, it is automatically commanded to land. The events to configure are:

- **Variable:** Power Input < 15.2 [V].
- **Phases:** Climbing or Cruise.



Low Battery – Automation

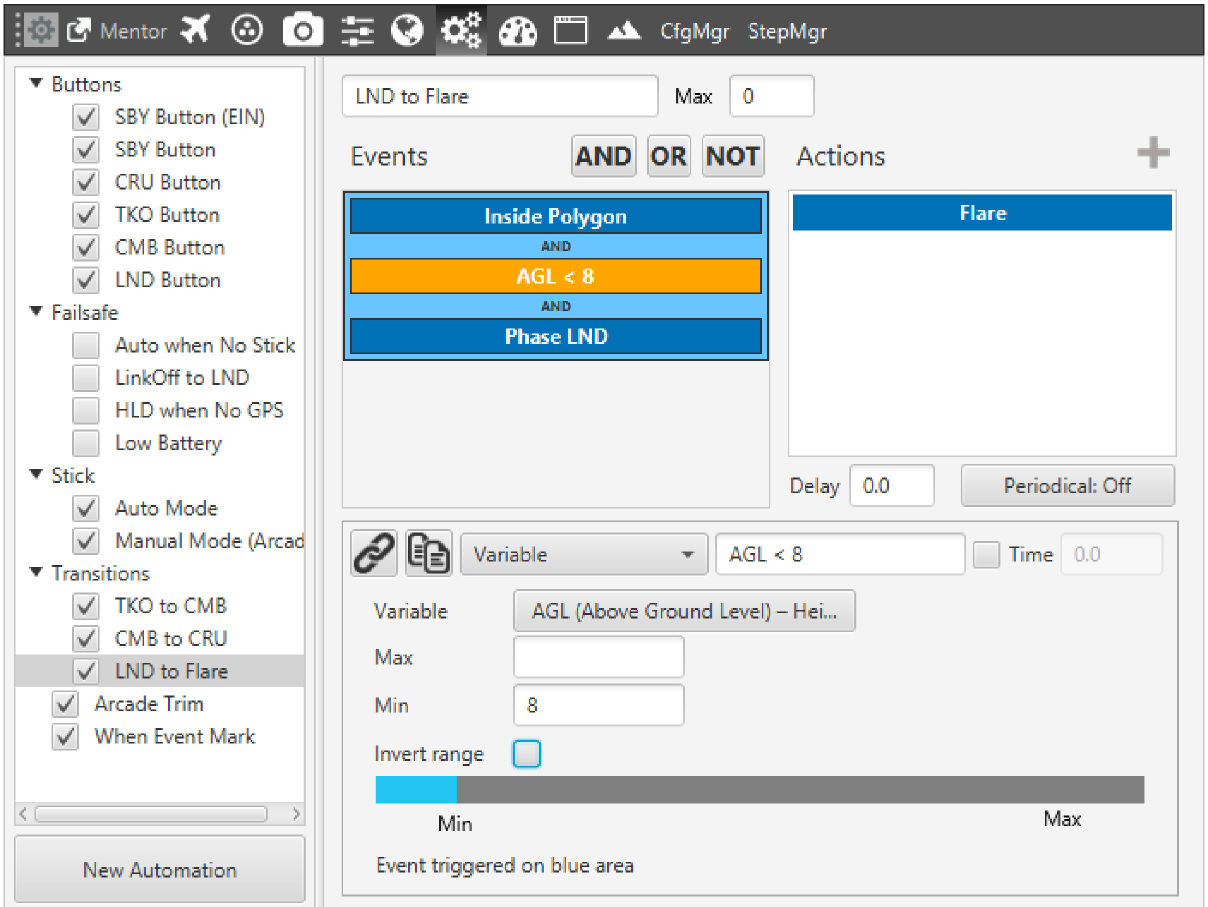
Two actions are configured:

- **Phase:** Landing.
- **Mode:** Auto.

11.2.2.1.3.5 Landing to Flare

The change from the landing to flare phase is triggered when the aircraft is above the runway and about to touch the ground. This idea is implemented with a set of three events:

- **AGL:** the aircraft is below a certain altitude, in this case, 8 meters.
- **Inside Polygon:** Mentor enters in a polygon defined in the map on the runway head.
- **Landing Phase:** the aircraft is landing phase.



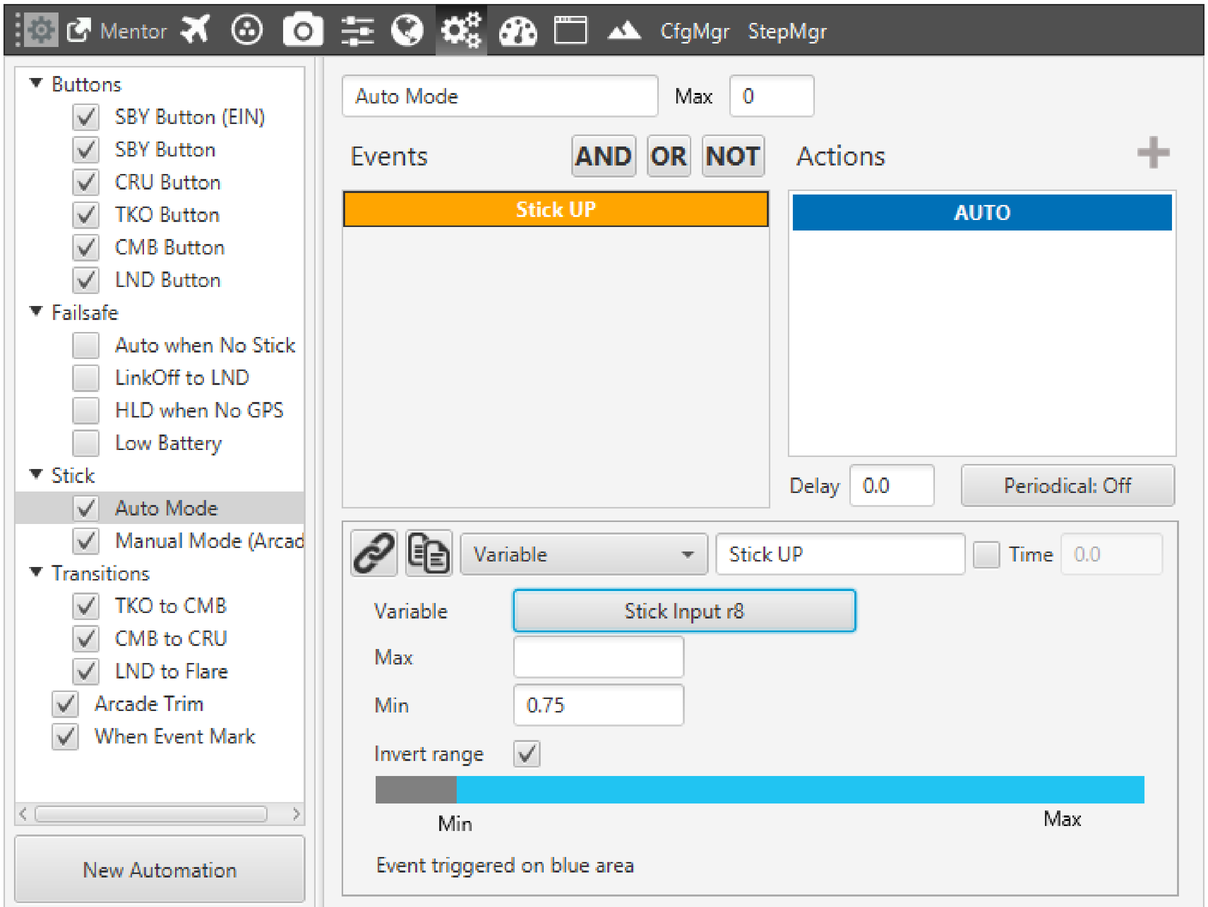
Landing to Flare

When these events are fulfilled, Mentor enters in the Flare phase and lands.

11.2.2.1.3.6 Stick Auto

This automation changes the control mode according to the command sent by the radio controller. Only an event is necessary:

- **Variable:** Stick Input rX, greater than 0.75.



Stick Auto

The action is:

- **Mode**: Auto

As it can be seen, the channel that controls the mode is the 8, so according to its value the mode is changed.

The process is the same when creating manual mode, but now the value of Stick Input r8 has to be lower than 0.25.

The Mentor UAV is a radio controlled, electric powered trainer aircraft. Made of special foam, is one of the largest RC aircrafts made of this material.

Mentor

It is a platform with the conventional controls of an airplane: elevator, ailerons, rudder and throttle, so it can be a good example of how to configure a typical airplane in Veronte Pipe.

11.2.2.2 Flying Wing-W210

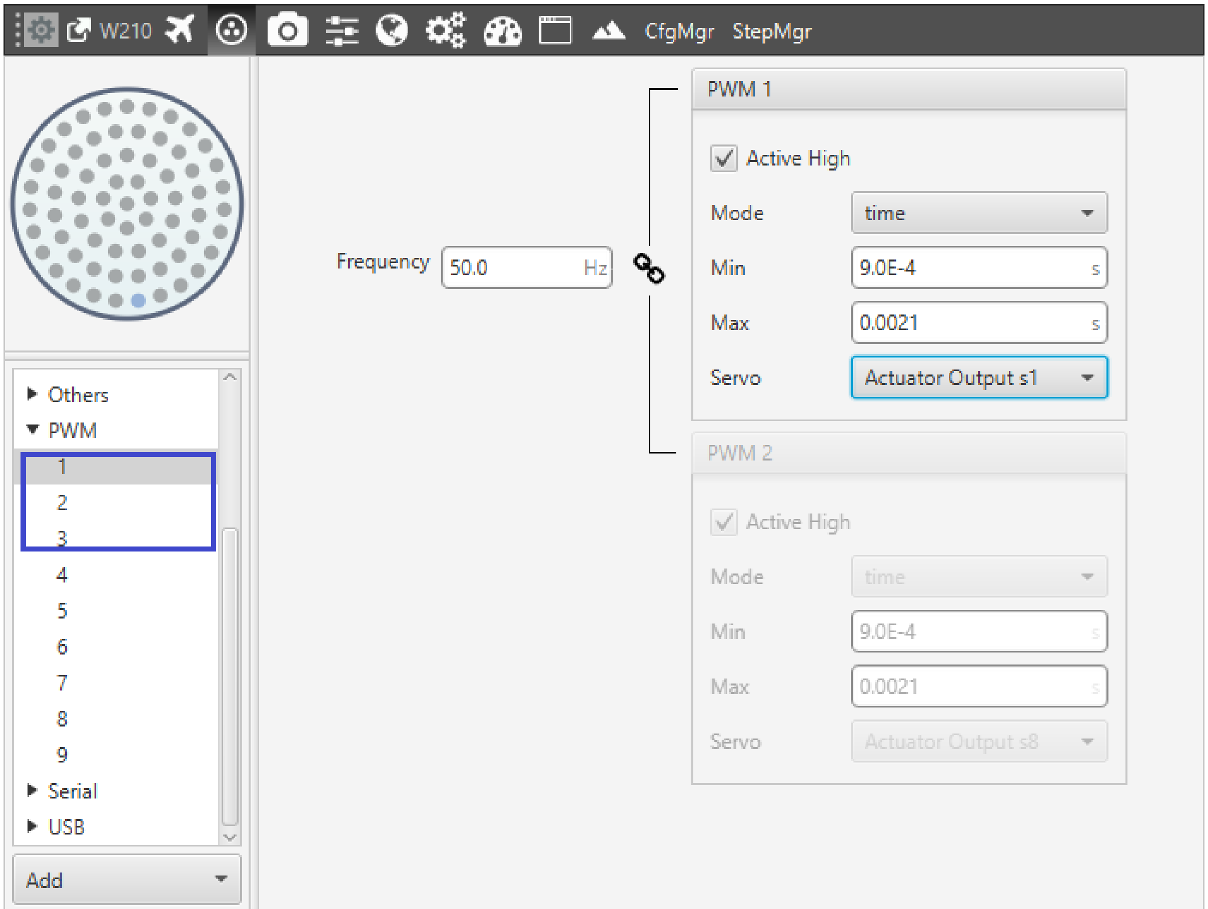
11.2.2.2.1 Servo Configuration

W210 configuration process can be performed by using a VerontePipe software version connected with the hardware system and the Autopilot as explained in section *Aircraft Mounting* of the manual.

11.2.2.2.1.1 Servos Output

The first step of the process is the servos configuration.

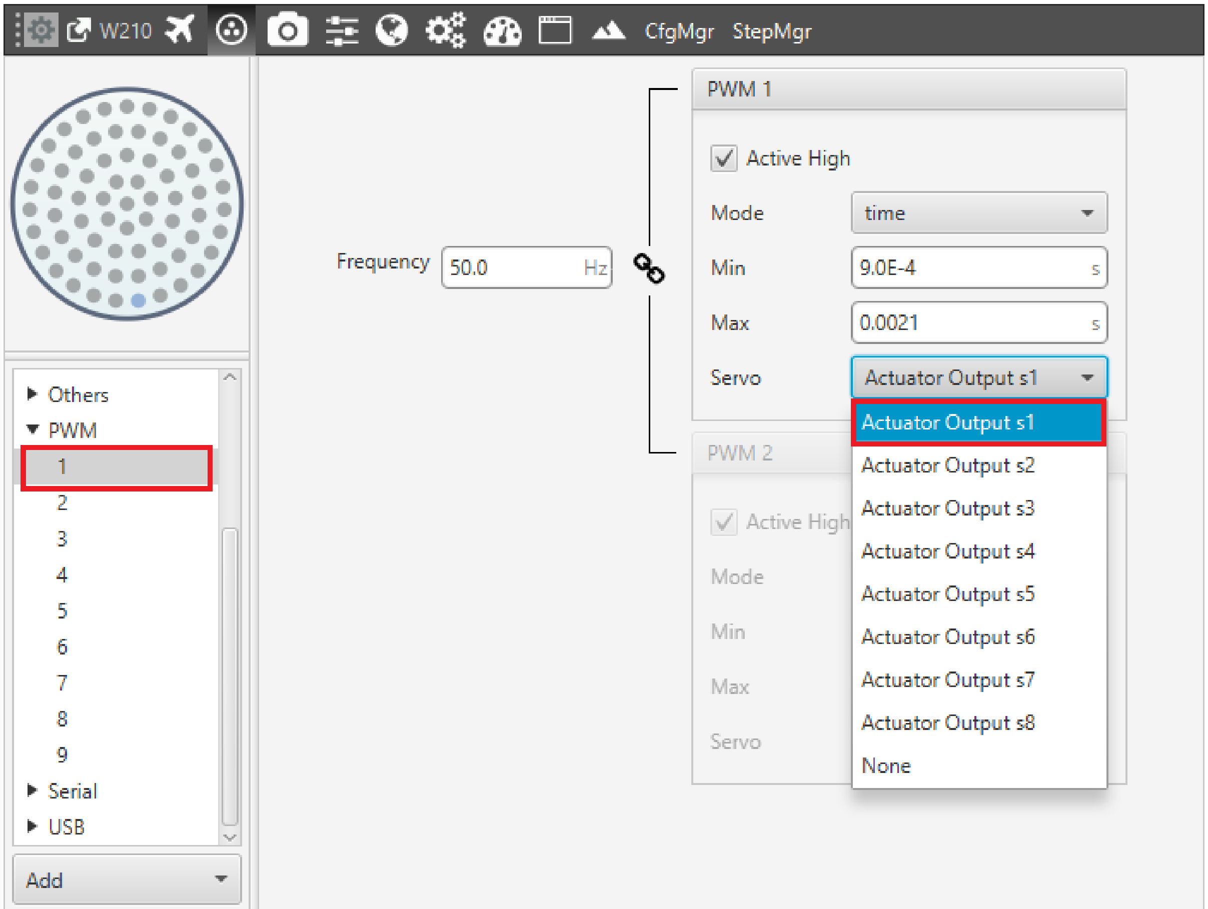
The controls of the airplane are the two control surfaces and throttle. Each one of these control corresponds to a pin of the connector and they must be positioned in the same order in an **S** vector who represents the **Actuator Output**. It is possible to connect any pin to any control surface or command but the easiest way to perform it and avoid confusion is by following the pins number.



Output-pin links

In this case, it is possible to use only 3 pins:

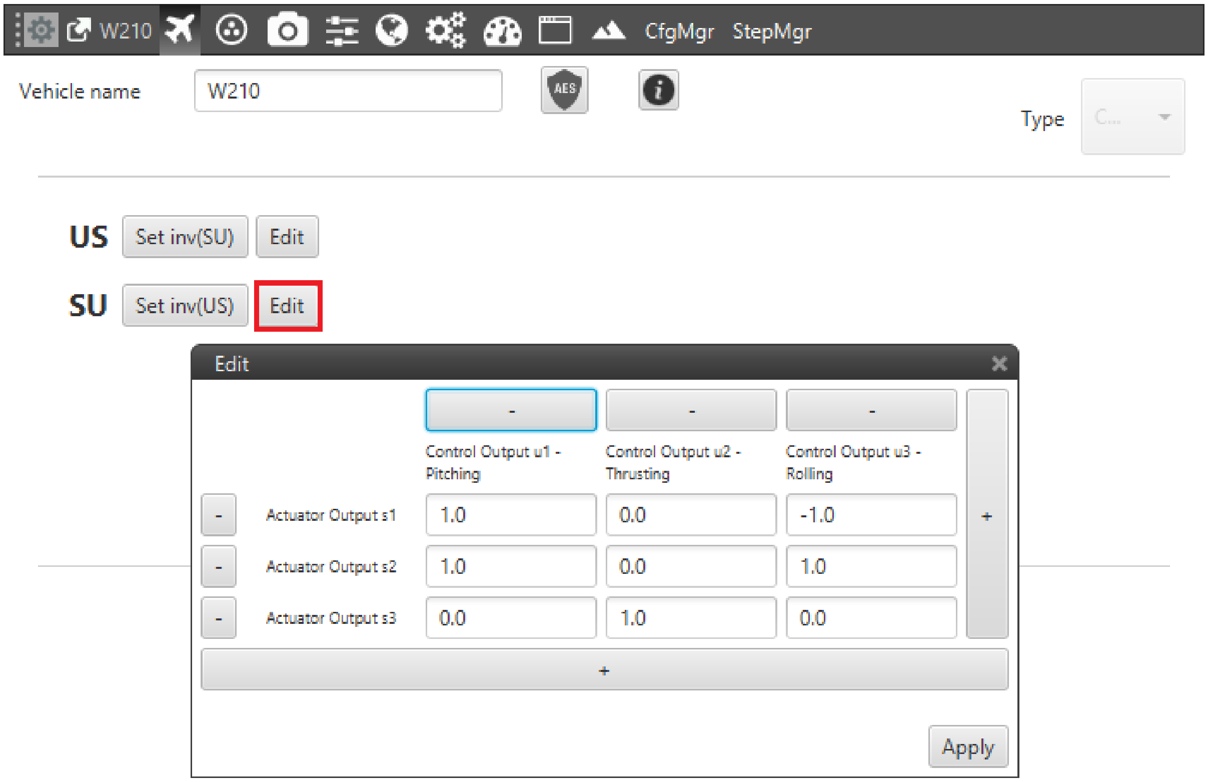
- Output 1 – Actuator Output s1 (**Aileron 1**)
- Output 2 – Actuator Output s2 (**Aileron 2**)
- Output 3 – Actuator Output s3 (**Throttle**)



Output 1 – pin 1 configuration

11.2.2.2.1.2 SU Matrix

At this point, the **S** vector is defined and the **SU** matrix can be edited. By clicking on **Edit** it is possible to configure the relation between the controller outputs (**U** vector) and the servo movements (**S** vector).



SU matrix editing

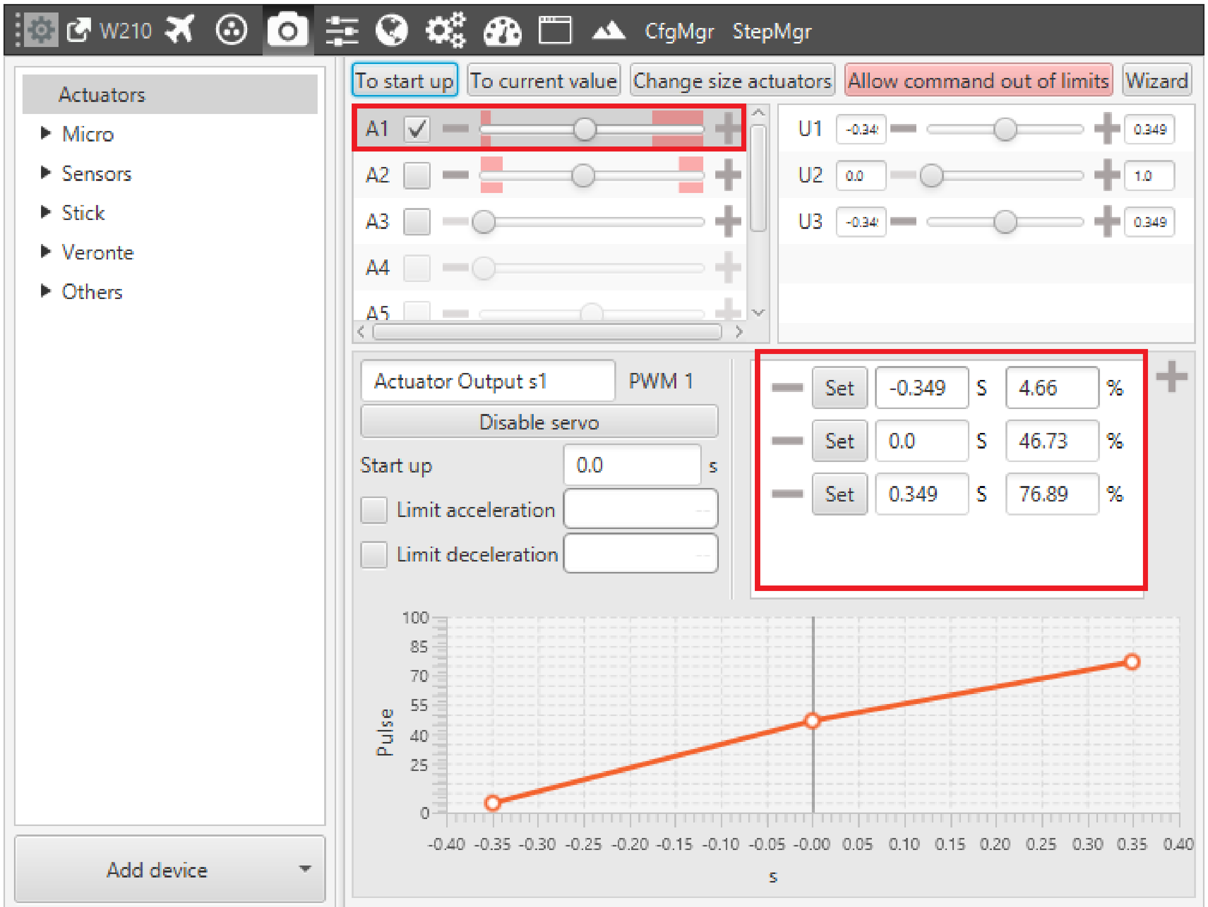
The W210 is configured as follow:

1. **Pitch Control:** control surfaces must be moved in the same direction to modify the pitching angle. The contribution of the actuators has same magnitude and direction.
2. **Thrust Control:** the Actuator Output 1 is the only one that allows a thrusting change.
3. **Roll Control:** in this case, the contribution of the actuators must be set with the same magnitude and reverse direction in order to perform a rotation around the X axis.

Warning: This panel shows the reference system of the aircraft. It must be positioned in the same way of the Autopilot's one. If it results different, it can be edited by clicking on the corresponding axis in order to reverse its direction.

11.2.2.2.1.3 System Trim

As a final step, the system has to be trimmed. This can be performed by moving servos in three different positions: zero position, minimum and maximum deflection angle (angles are usually limited physically). These positions must be inserted and saved in the software by clicking on **Set** when the actuator is in the desired position. Otherwise, position can be introduced manually.



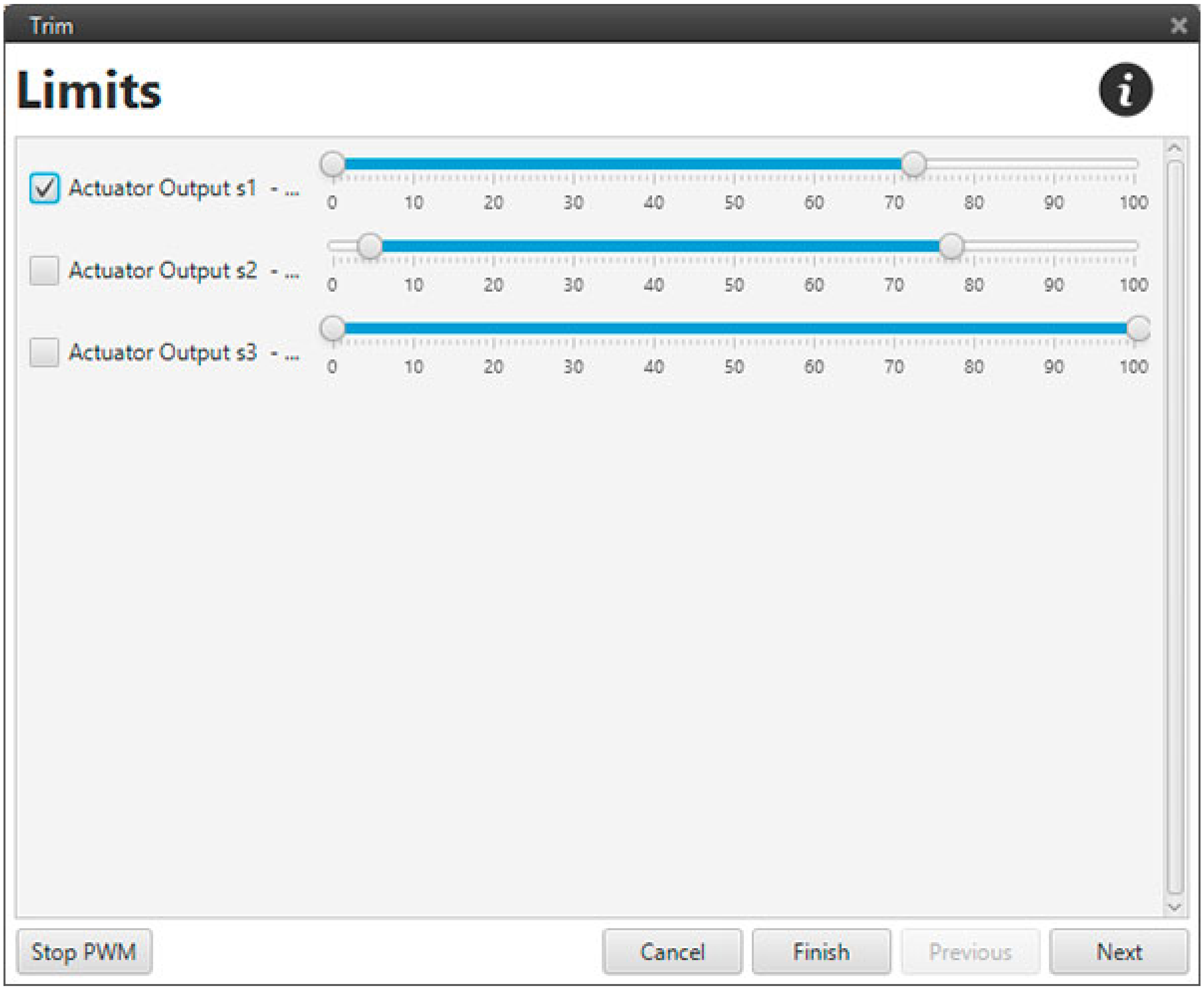
Actuator 1 Trimming

The picture shows the setting of the elevator number 1:

- **Minimum:** -0.34906 [rad] deflection; 4.6682%
- **Zero position:** 0.0[rad] deflection; 46.7317%
- **Maximum:** 0.34906 [rad] deflection; 76.8904%

> **Warning:** The actuators can be moved directly from VerontePipe only when the system is in an Initial phase. During the actuator run, if the desired position is in the **Out of range** zone (red zone), it is possible to click on **Allow command out of limits** in order to move completely the actuator and find the correct position.

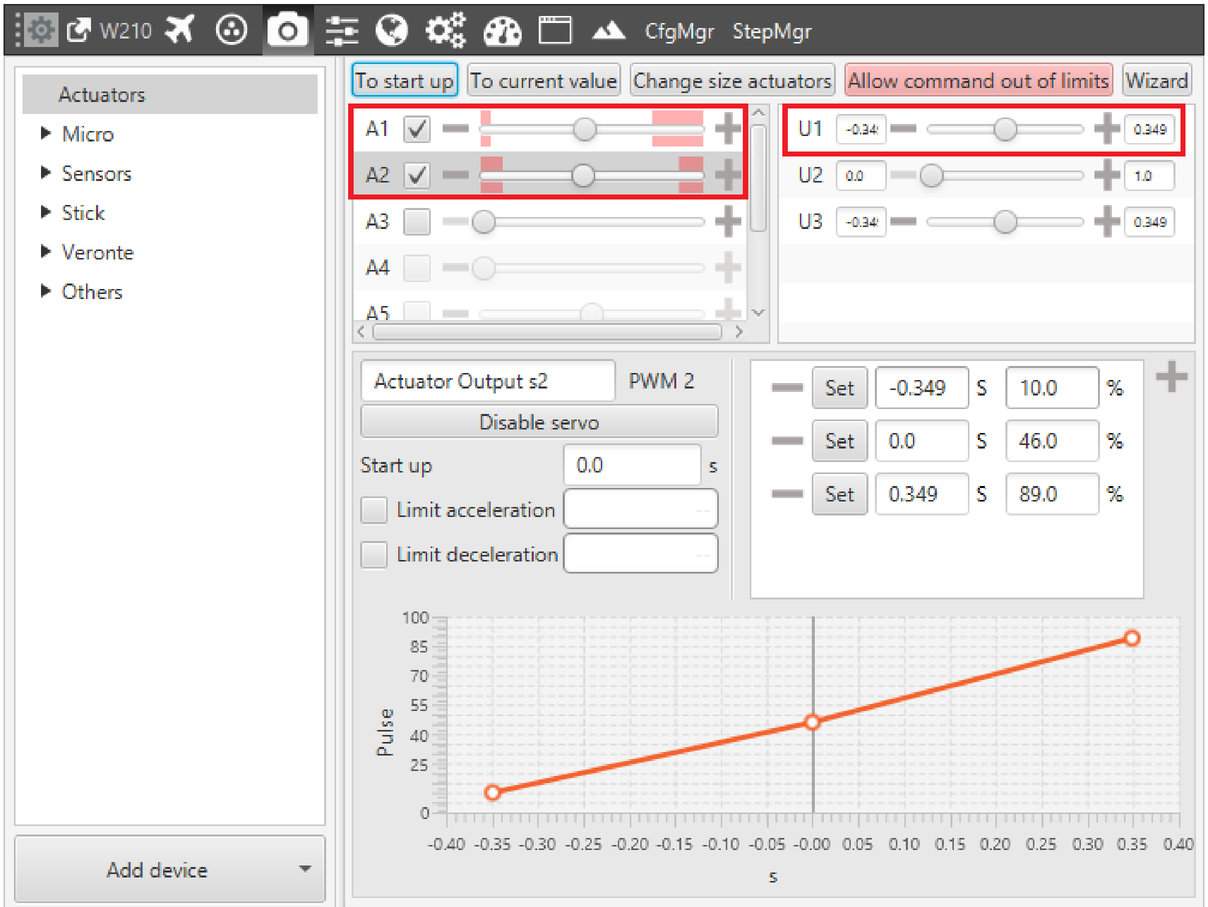
This procedure can be performed in the same way by using a **Wizard**. This tool allows moving actuators limits easily and finding the correct range.



Trim wizard tool

In order to perform a final checking, it is possible to select the desired channel and testing pitch, roll and thrust controls.

The image below shows a pitching output testing. By moving the U1 control, surfaces must change the position according to the reference system: positive corresponds to nose down and negative to nose up.



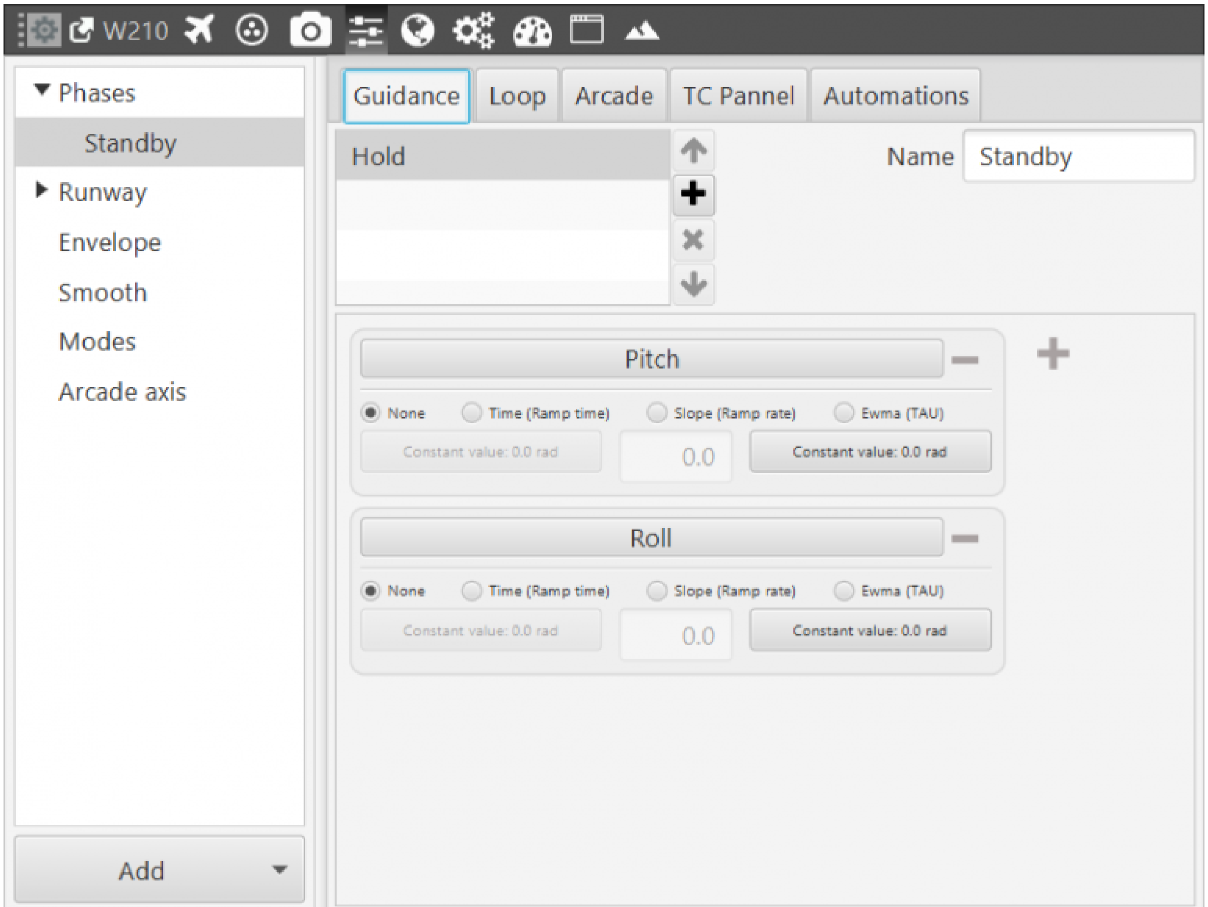
Pitching test

11.2.2.2.2 Mission Phases

In this section, it will be explained the W210 typical mission profile and the mission phases will be detailed.

11.2.2.2.2.1 Standby

Standby phase is a preliminary phase of the operation. During this phase is possible to check, for example, if the aircraft is correctly controlling the attitude by change it and seeing the control surfaces moving.



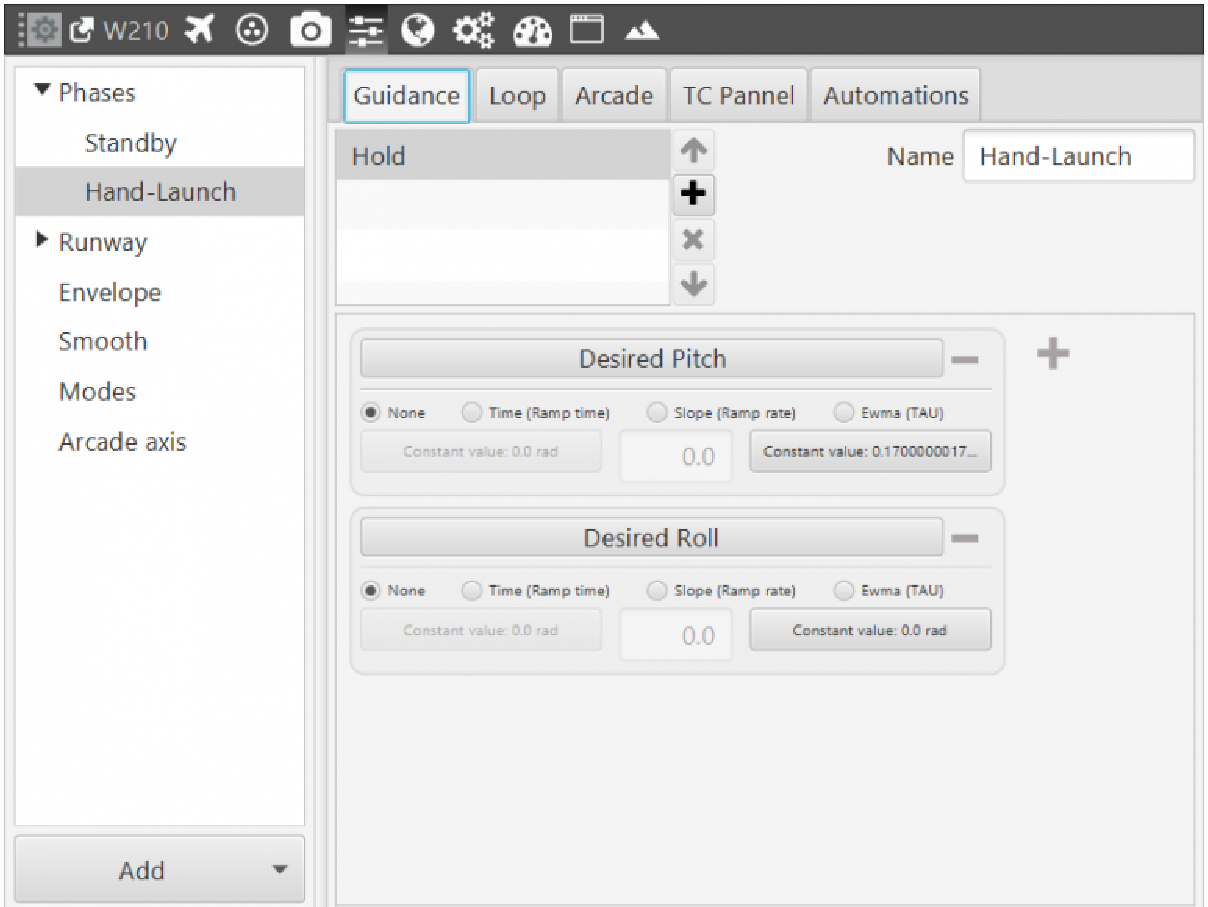
Standby phase panel

The guidance is a Hold of two angles:

- **Pitch and Roll:** kept at 0 [rad].

11.2.2.2.2.2 Hand-Launch

Hand-Launch phase is the first launch modality. In this case, the aircraft is launched "manually" by a person who has to increase is velocity trying to maintain the platform in a correct attitude. The image below shows the Hand-Launch phase configuration panel.



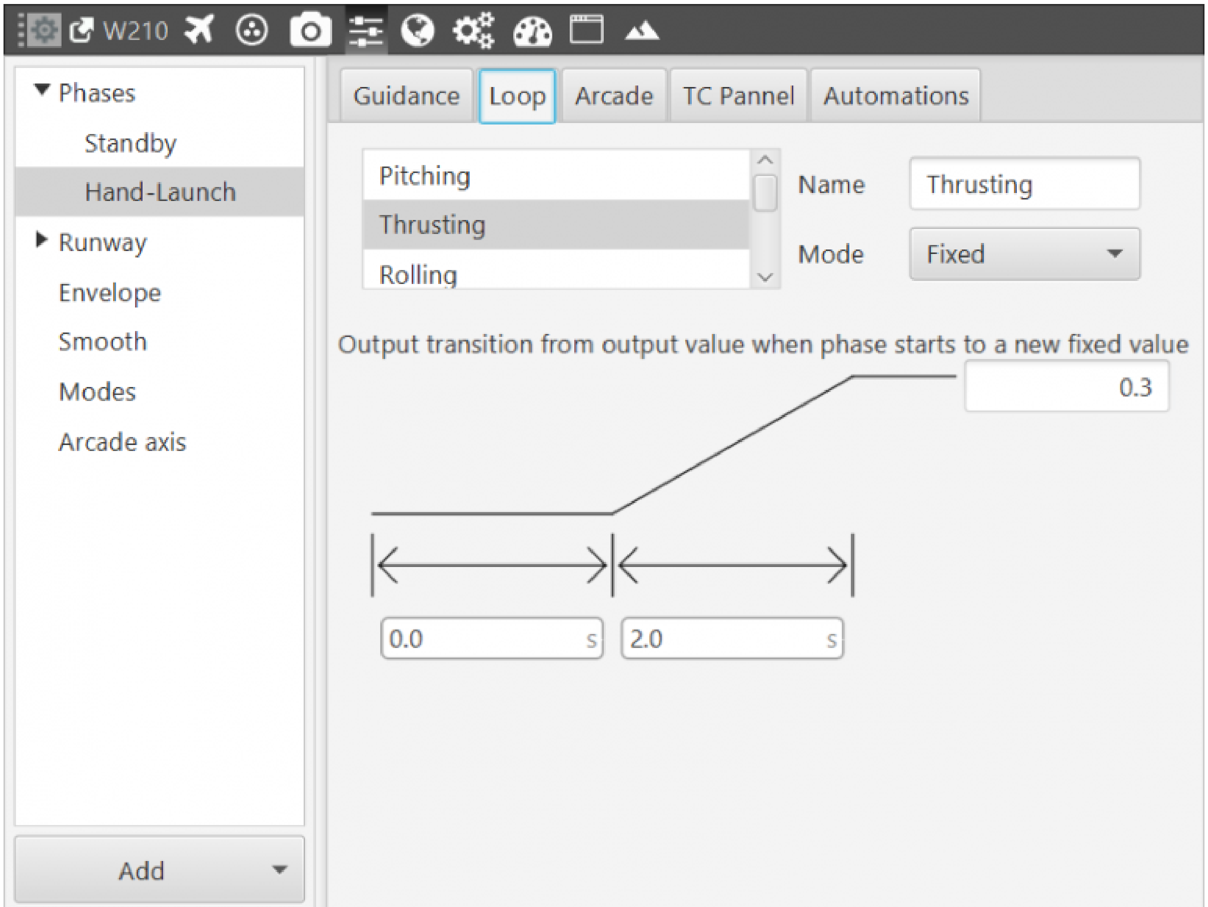
Hand-Launch phase panel

In this phase, the Guidance is a Hold of two variables:

- **Desired pitch:** at 0.17 [rad].
- **Desired roll:** at 0.0 [rad].

It means the platform control system will keep to zero the roll angle and the pitch angle to 10º degrees.

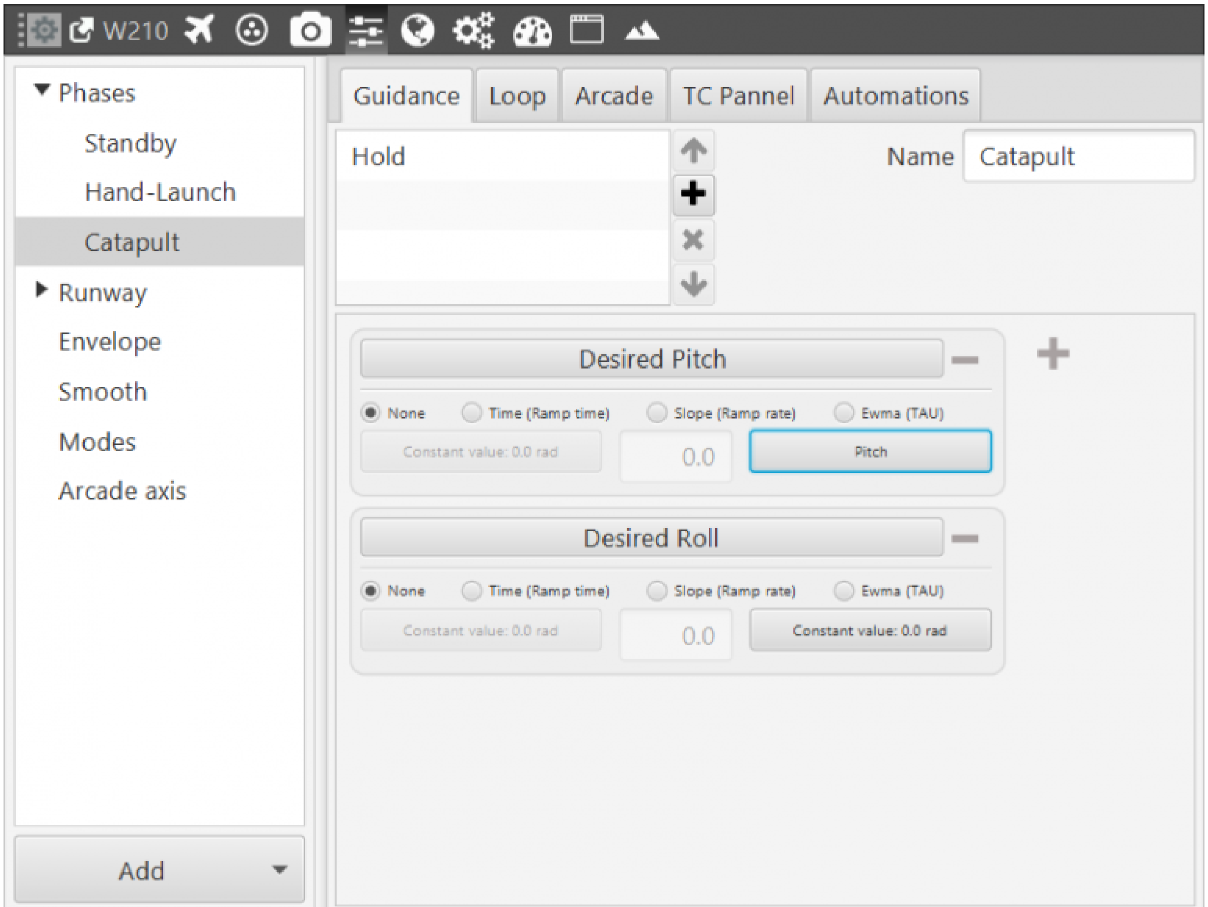
Furthermore, during this phase the motor starts. The image below shows the thrust behavior: during the Hand-Launch phase, thrust starts from zero and reaches the fixed value of 0.3 in 2 seconds.



Hand-Launch phase thrust

11.2.2.2.2.3 Catapult

Catapult phase is the second launch modality. The takeoff is performed using a catapult which allows reaching the desired speed and maintaining the correct attitude during the launch. The image below shows the Catapult phase configuration panel.



Catapult phase panel

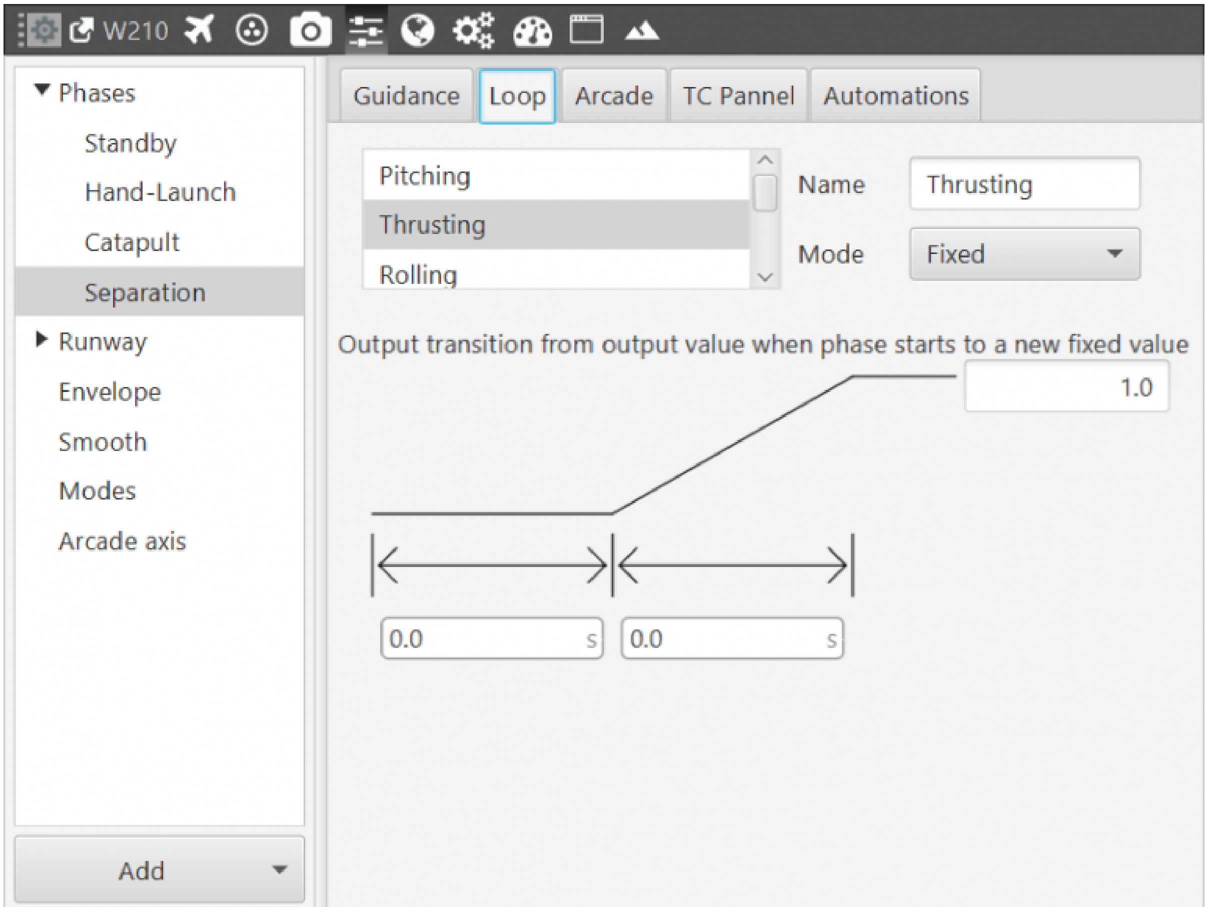
As happens in the Hand-Launch phase, the Guidance is a Hold of twop variables:

- **Desired Pitch angle:** at the current pitch angle.
- **Desired Roll angle:** at 0.0 [rad].

Thrusting, in this phase, is kept at zero.

11.2.2.2.2.4 Separation

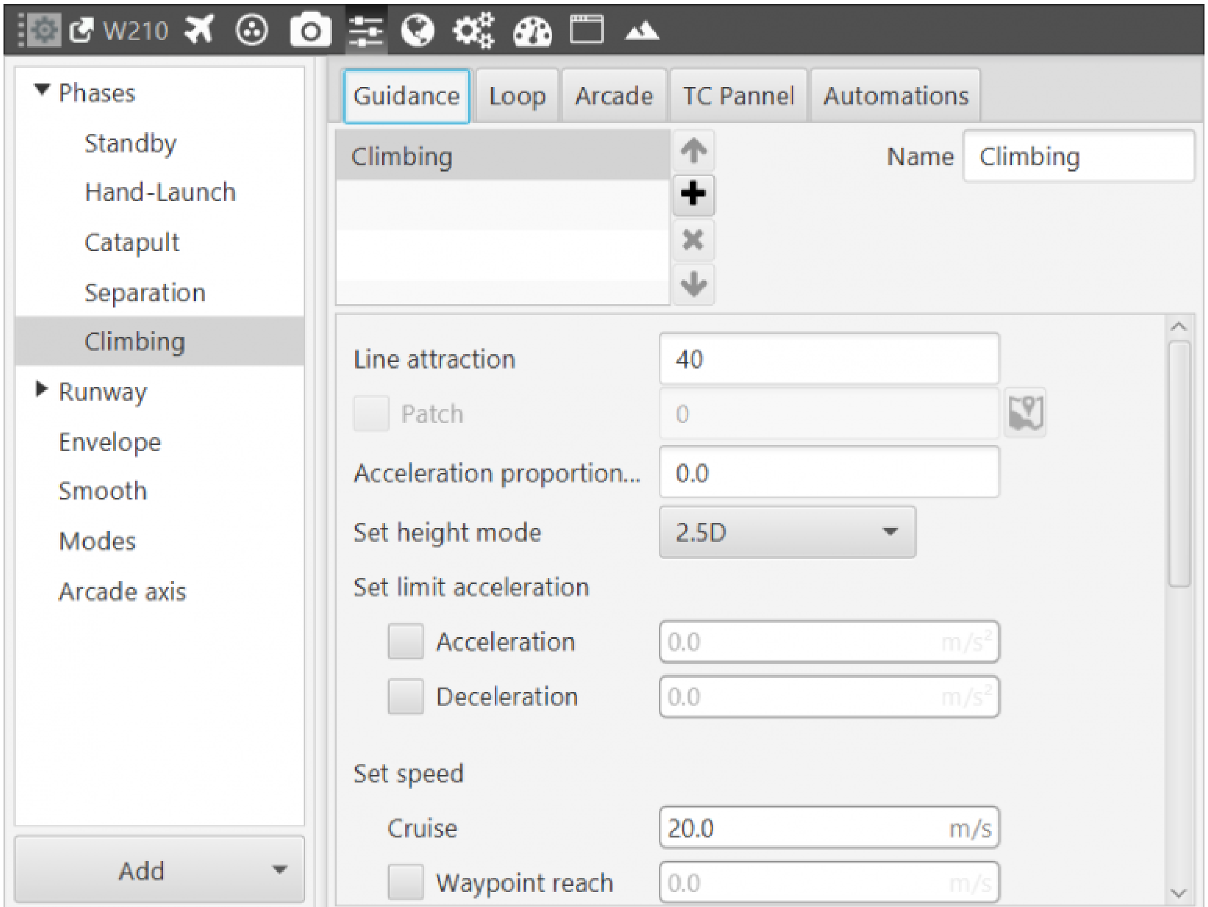
Separation is the second phase of this takeoff modality. Guidance is the same **Hold** configuration of the Catapult phase but Thrusting changes. In fact, Thrusting is an instantaneous step from 0 to 1 in order to switch on the motor only when the fan can not be able to hit the catapult.



Separation thrusting

11.2.2.2.2.5 Climbing

Climbing phase is configured to make the airplane reach the mission altitude after the takeoff.



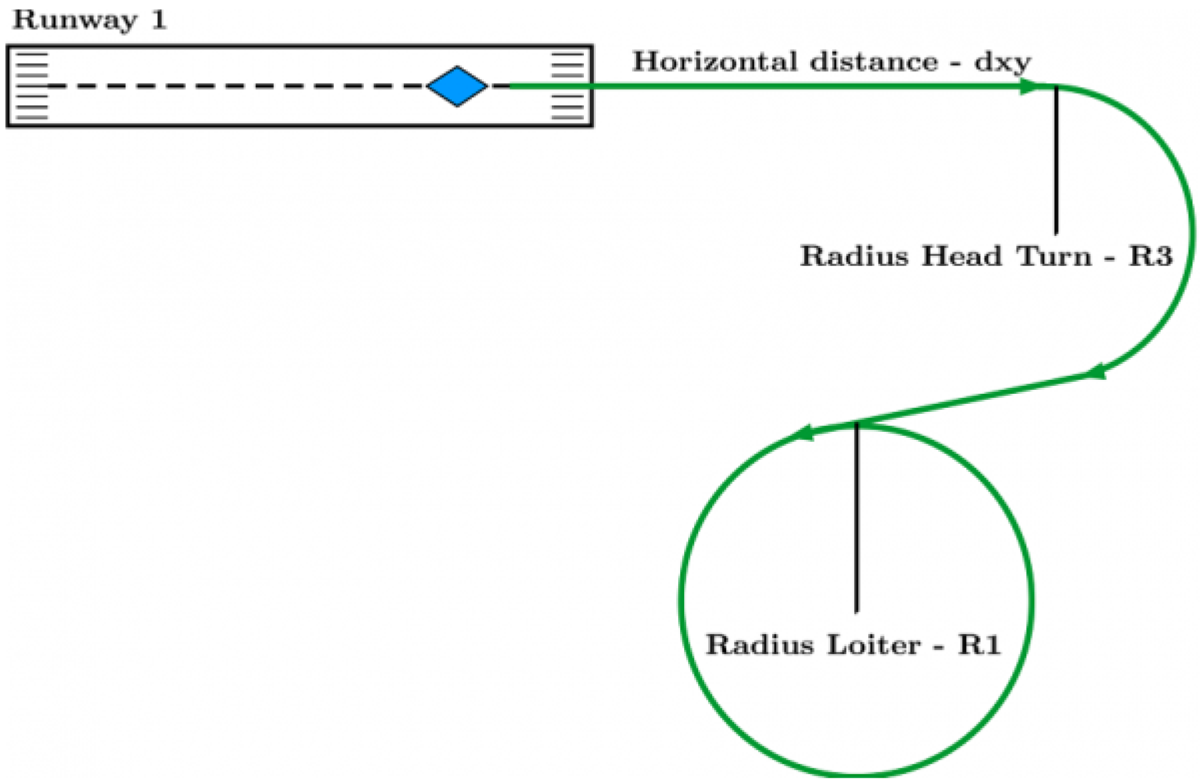
Climbing phase panel

This fase consists of a **Climbing** Guidance:

- **Line attraction:** value related to how strongly the aircraft tries to reach a path (climb path in this case). This value is commonly between 20 and 40 for airplanes. In this case, the value es set at 40.
- **Set speed:** speed that will have the airplane during the climb, in this case 20 [m/s] is a good value .

In this phase, the aircraft reaches the mission altitude by performing a spiral route. In order to determine the correct path, these parameters must be set:

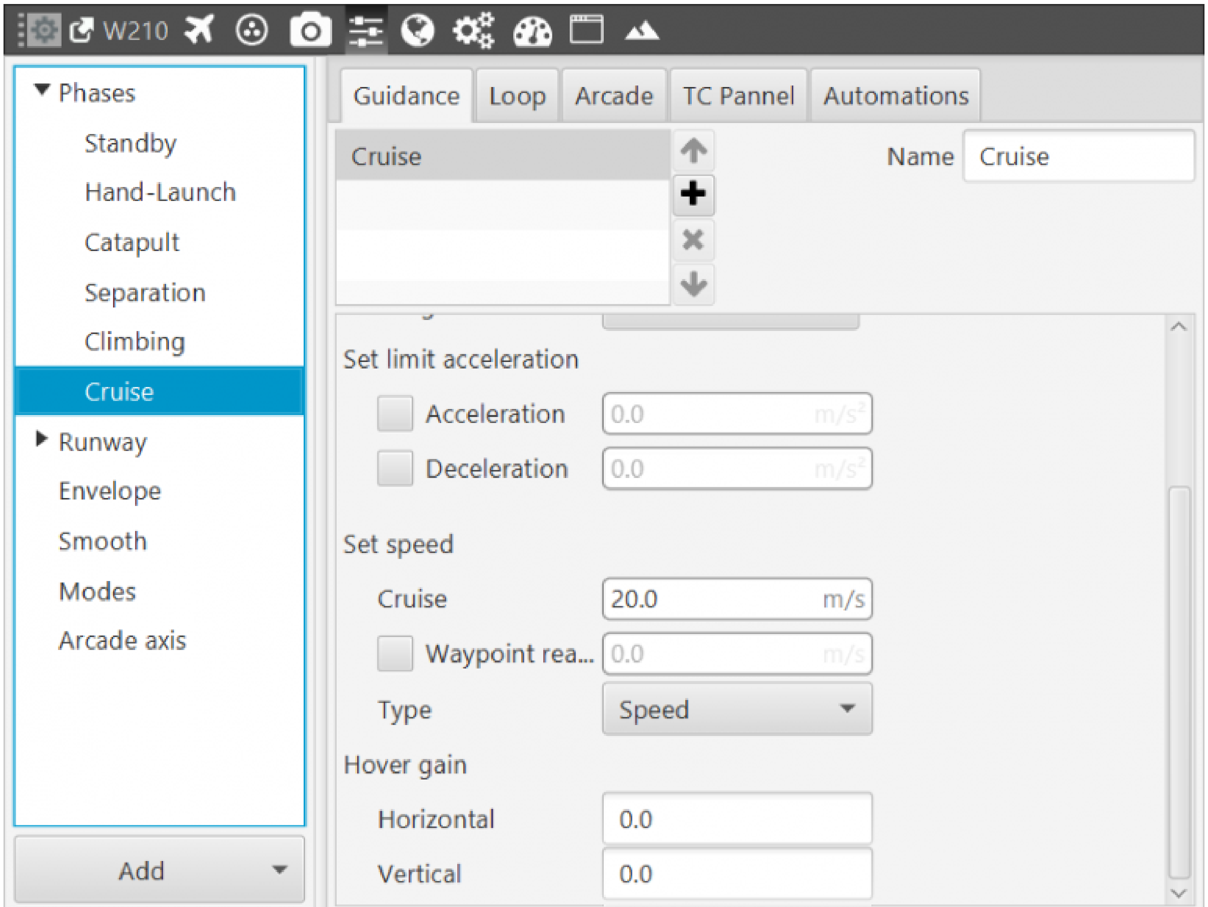
- **Runway:** here is selected the runway which previously has been edited in its configuration menu.
- **Flight Path Angle:** angle at which the aircraft will climb, 0.1396 [rad].
- **Horizontal Distance:** is the distance from the point where the aircraft enters in the phase which contains the climbing guidance, to the start of the circular climbing path, 100 [m].
- **Radius Head Turn R3:** radius of the turn made to head the airplane towards the loiter direction, 100 [m].
- **Radius loiter R1:** radius of the loiter ascending made by the aircraft to reach an altitude suitable, 100 [m].



Climbing Path

11.2.2.2.2.6 Cruise

In this phase, the Guidance is **Cruise**. The aircraft follows a route marked by a set of waypoints, which are defined by the user in the **Mission** menu.



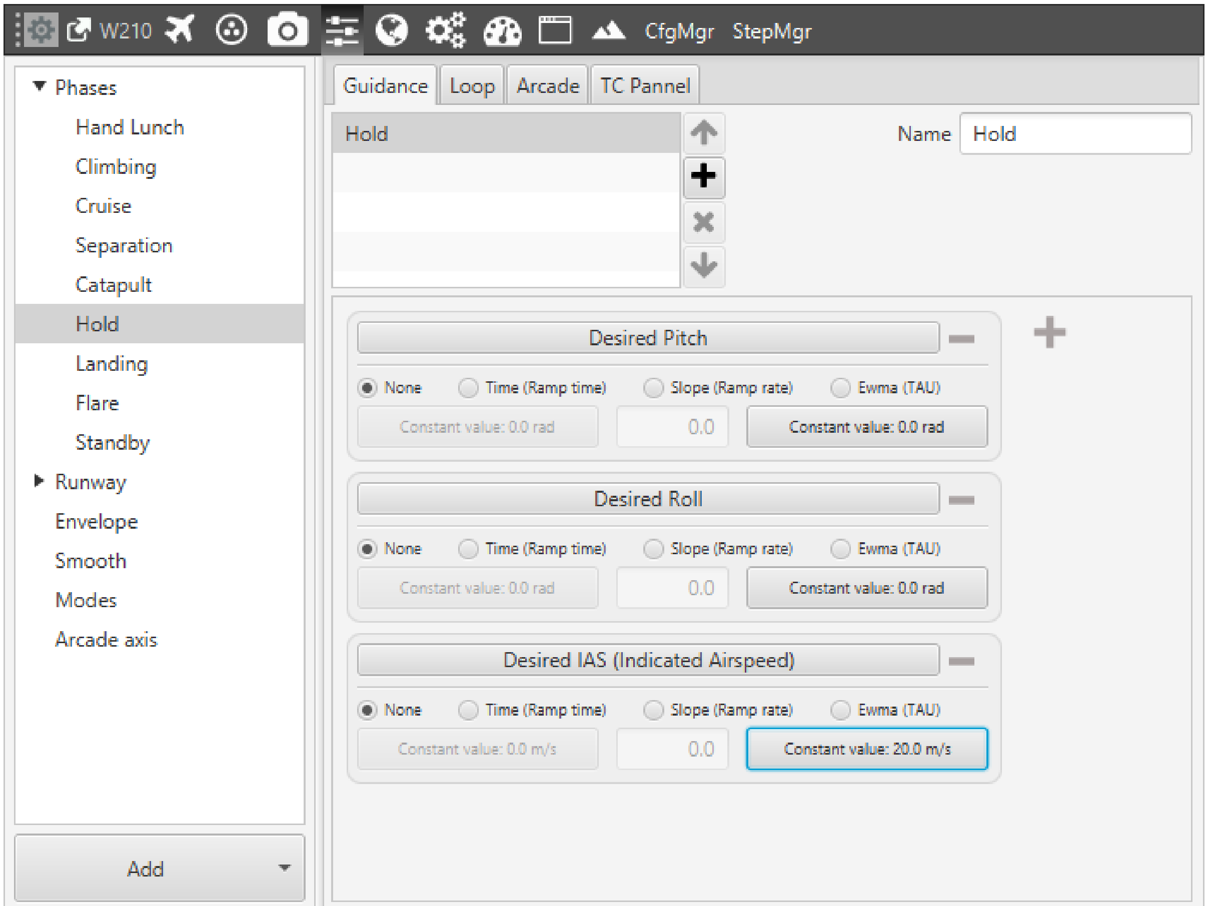
Cruise phase panel

Parameters set here are the same as the ones from the climbing phase.

- **Line attraction:** 40.
- **Set Speed:** 20 [m/s].

In this guidance, there is an option related to the gains used to recover the hover point in an multicopter, **Hover Gain**. In this case, the platform configured is an airplane, so this option will not be used.

11.2.2.2.2.7 Hold



Hold phase panel

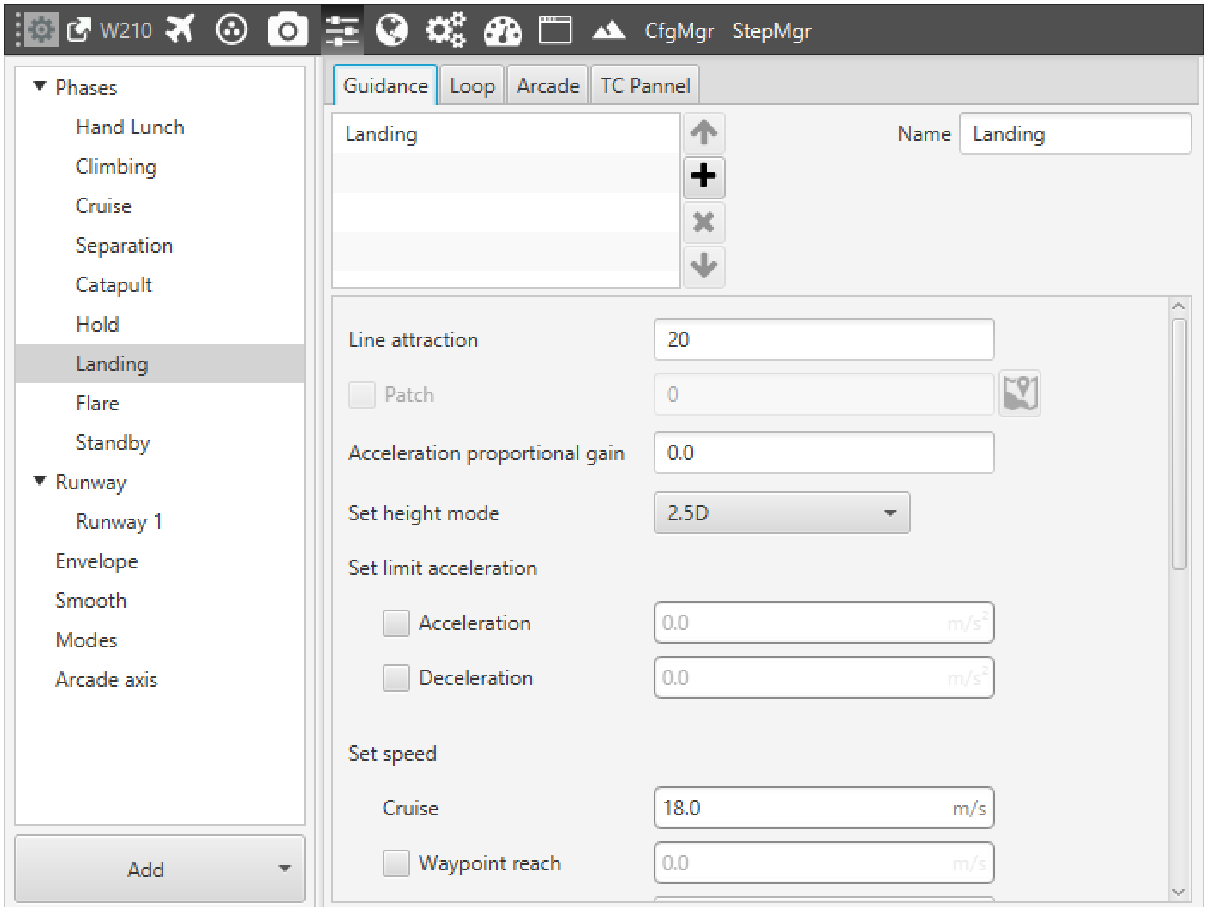
In this phase, the Guidance is a **Hold** of three variables:

- **Desired Pitch:** at 0 [rad].
- **Desired Roll:** at 0 [rad].
- **Desired I.A.S:** (Indicated Air Speed) kept at a constant value of 20 [m/s]

It means the aircraft will maintain this attitude until the next phase change.

11.2.2.2.2.8 Landing

This phase is used to make the aircraft land at a certain airport. Also, when the flight altitude is too big, this phase contains the parameters which define the route performed by the platform to descent until an altitude where it can line up with the runway.



Landing phase panel

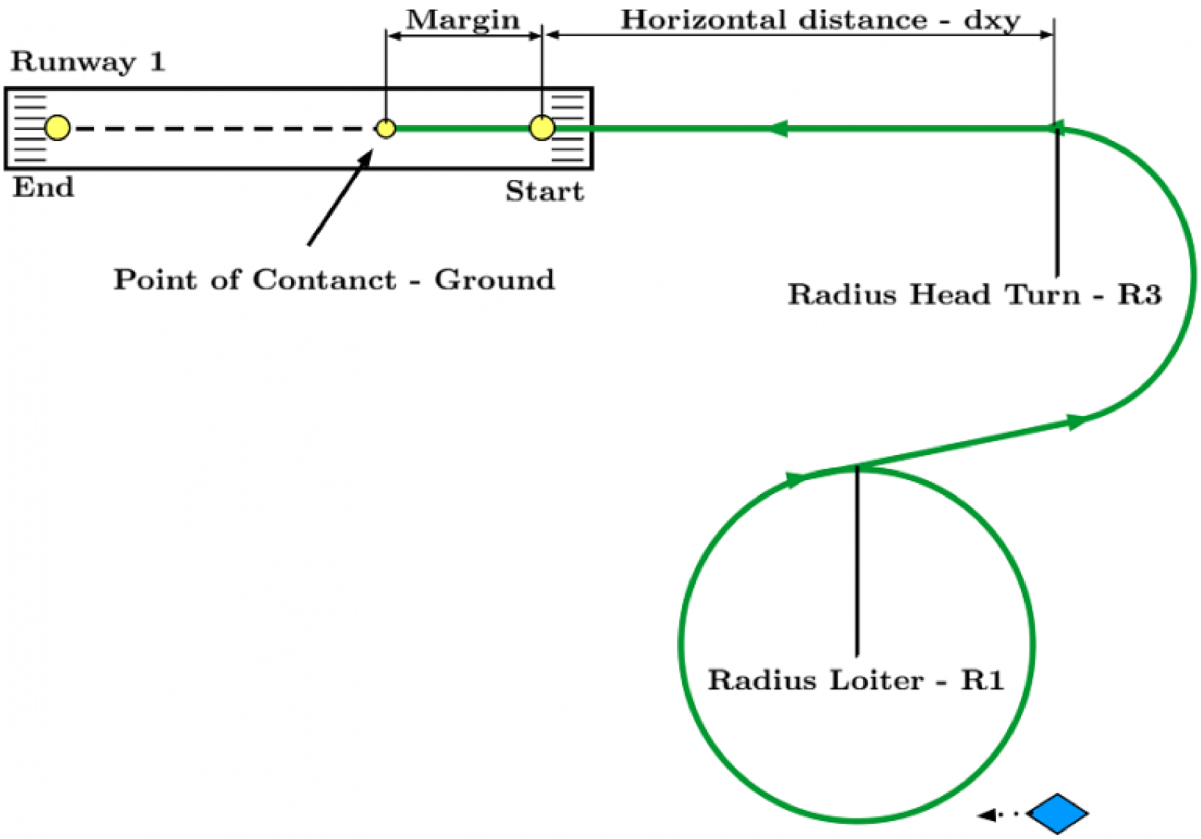
The guidance to be configured is Landing:

- **Line Attraction:** value related to how strongly the aircraft tries to reach a path, kept at 20.
- **Set Speed–Cruise:** 18 [m/s]

Finally, the following parameters define the path during this phase.

- **Runway:** here is selected the runway which previously has been edited in its configuration menu.
- **Flight Path Angle:** angle at which the aircraft will descend, -0.14 [rad].
- **Horizontal Distance:** is the distance from the point where the aircraft enters in the phase which contains the climbing guidance, to the start of the circular climbing path, 300 [m].
- **Radius Head Turn R3:** radius of the turn made to head the airplane towards the runway direction, 150 [m].
- **Radius loiter R1:** radius of the loiter descending made by the aircraft to reach an altitude suitable to perform the landing manoeuvre, 120 [m].

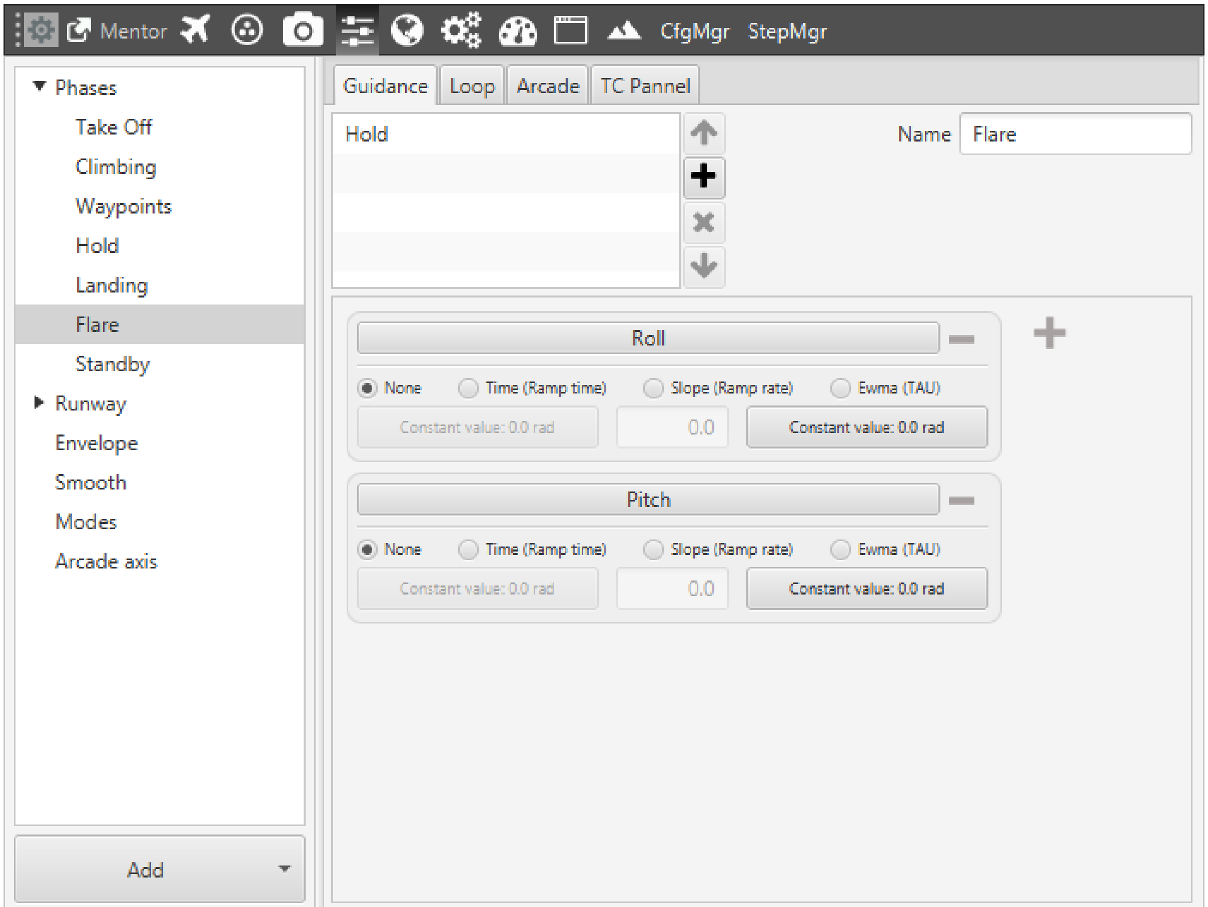
The route of this phase is shown in the following figure, where each one of the parameters that define it are defined.



Landing Route

11.2.2.2.2.9 Flare

Flare phase allows defining a point or a zone near the runway where the aircraft will perform a pitch angle change (nose-up) in order to modify its attitude before the touchdown and avoid a possible crash due to a nose-ground direct contact.



Flare phase panel

- **Roll – Pitch Angle:** kept at a constant value 0 [rad].

Regarding the thrust in this phase, the engines are shut off, so the mode of the controller is **off**.

11.2.2.2.3 L200 Catapult

L200 Catapult

From the Embention website it is possible to download User Manual and the Datasheet of the L200 Catapult.

11.2.2.2.3.1 Configuration

In order to configure correctly the launch system, it is necessary to perform two different actions:

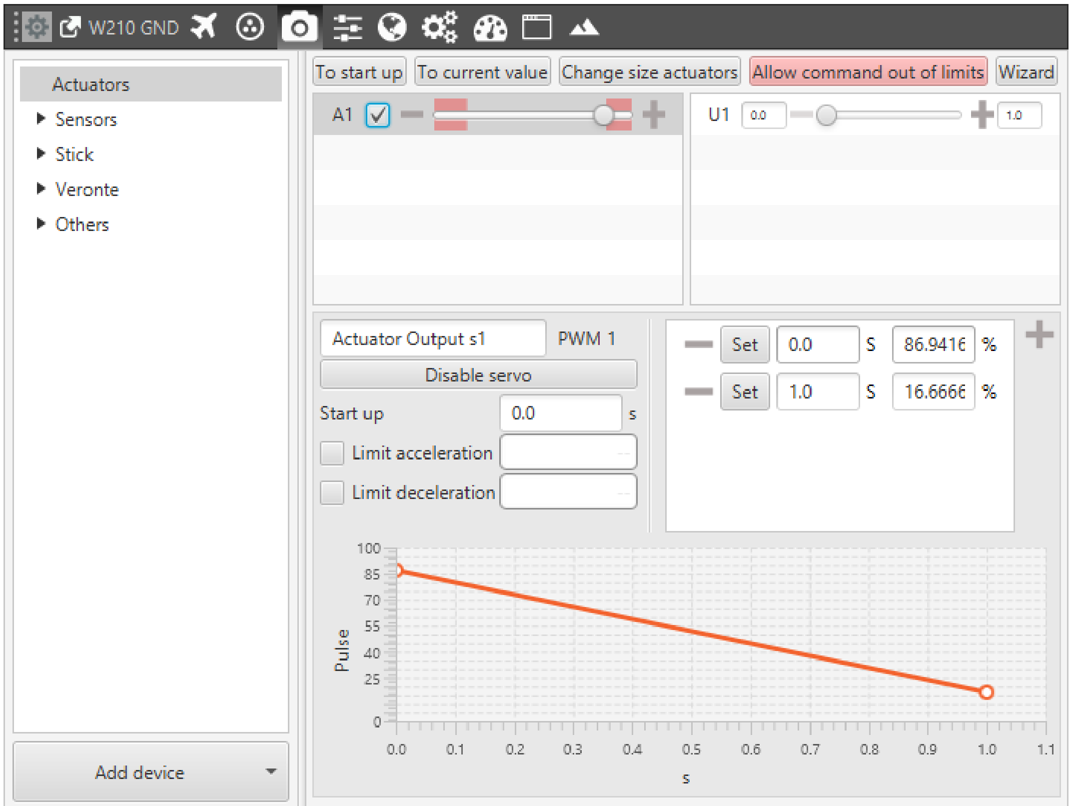
- Trimming of the launch actuator.
- Create the automation for the launch button.

To set them, it is necessary to connect the catapult to Veronte Ground using the catapult wire and completing the system with a PC connection with Veronte Pipe.

11.2.2.2.3.2 Actuator Trimming

To trim the actuator it is necessary to create the catapult actuator by following the path **Devices>Control>Actuators**.

1. Click on **Change size actuators** and select the actuator number (1) from the window.

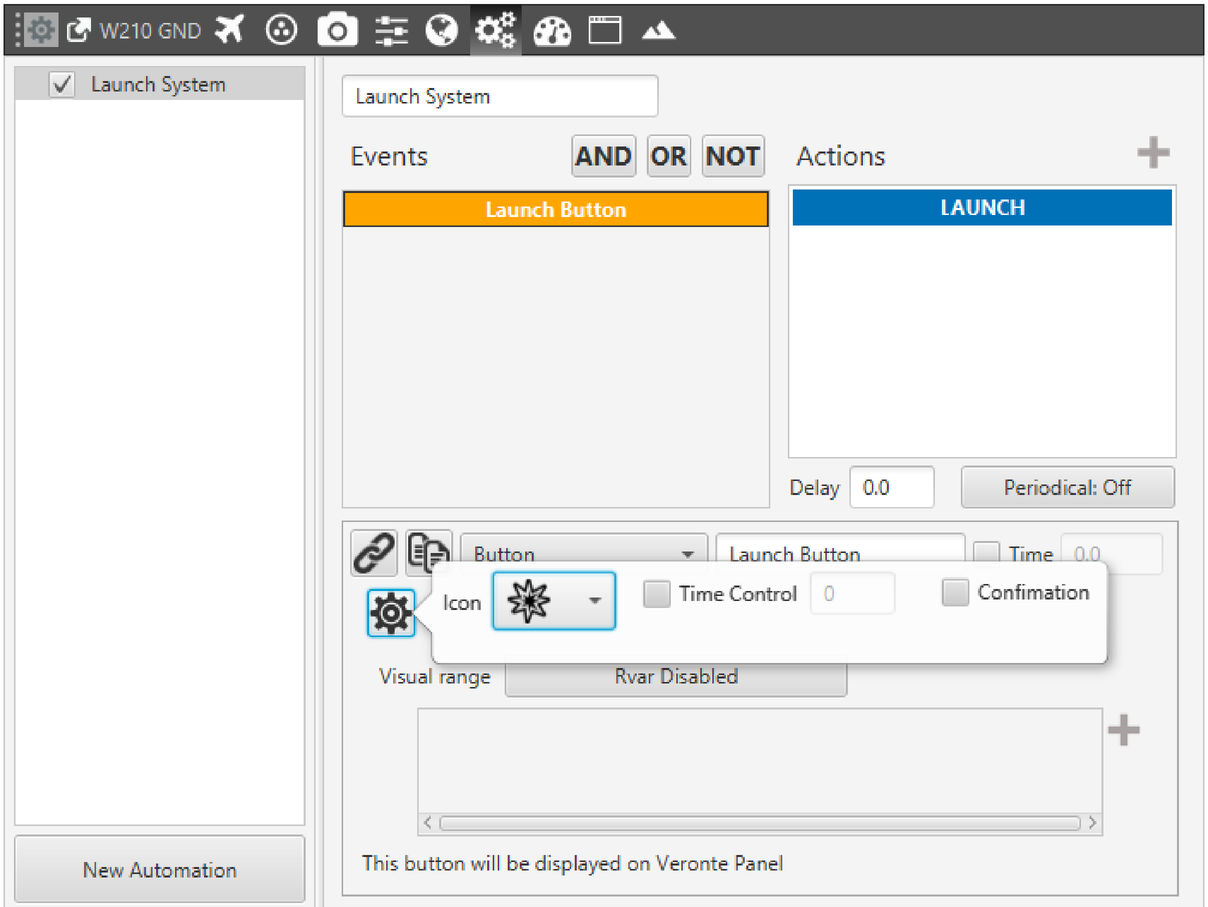


Catapult actuator trimming

2. Once created an actuator, it is possible to trim it. Default values are shown in the previous figure, but user must be sure the servo can move covering all useful range, from bottom to top (0-1).

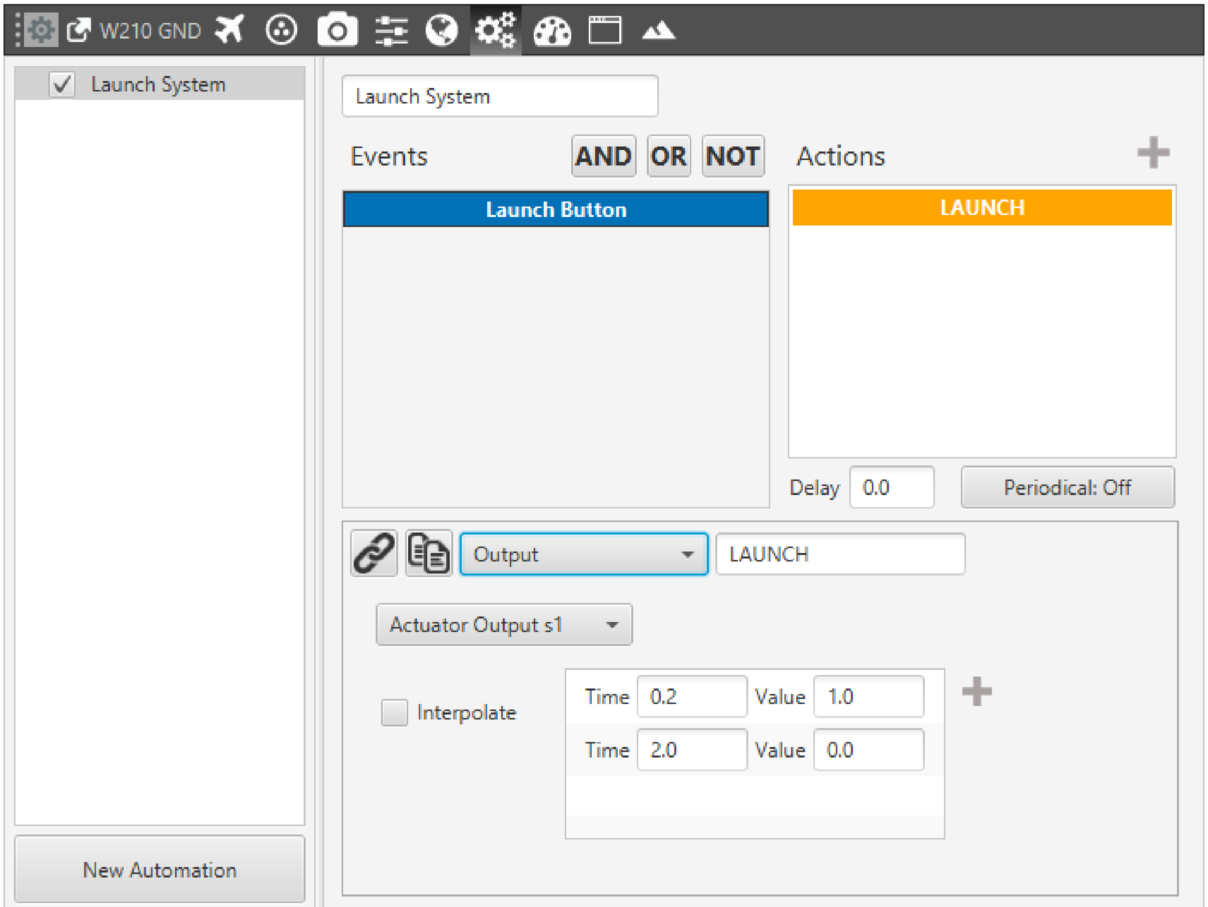
11.2.2.2.3.3 Launch Automation

The automation must be created in the Ground autopilot. The typical Event of the automation is a Button which will appear in the Ground Veronte Panel.



Automation Event

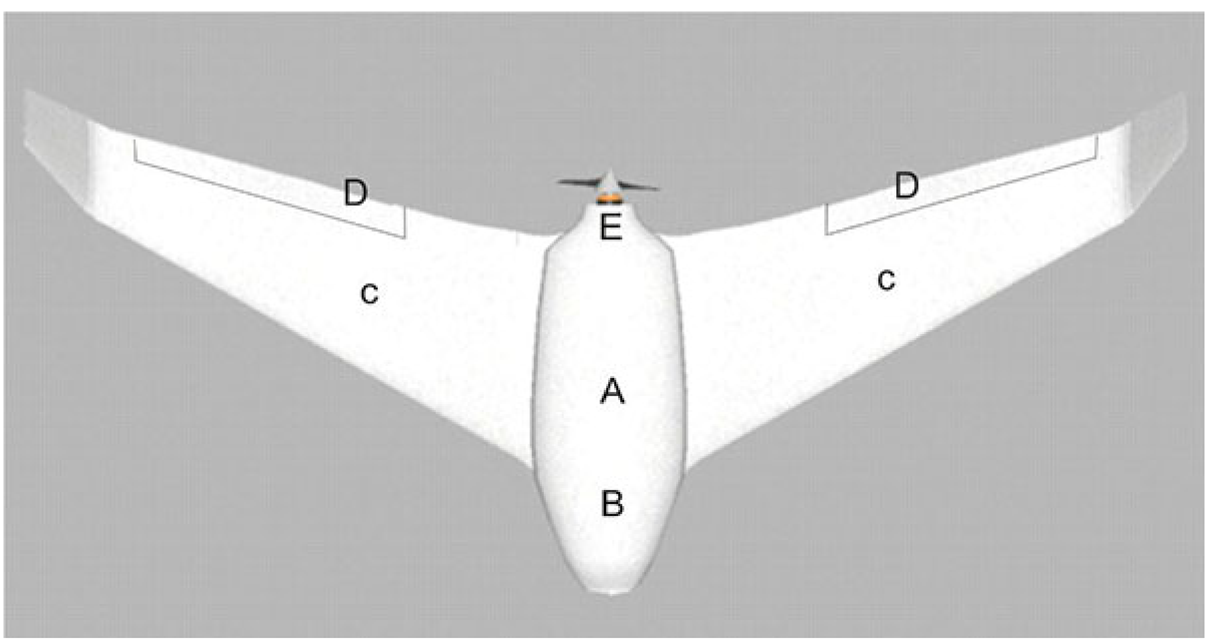
The Action is the change from 0 to 1 of the actuator value. A second value of the actuator should be set in order to restore the initial null value (in this case, after 2 seconds).



Automation Action

W210

This airplane has a "flying wing" structure and can be piloted by using only two symmetrical control surfaces. They can be moved to control the aircraft attitude in a symmetrical way to perform a pitch variation and in an unsymmetrical way to control the roll.



W210 parts in XPlane model

- Power battery

- Control electronics
- Detachable wings
- Control surfaces (ailerons)
- Power motor and propeller

11.2.2.3 Quadcopter-M400

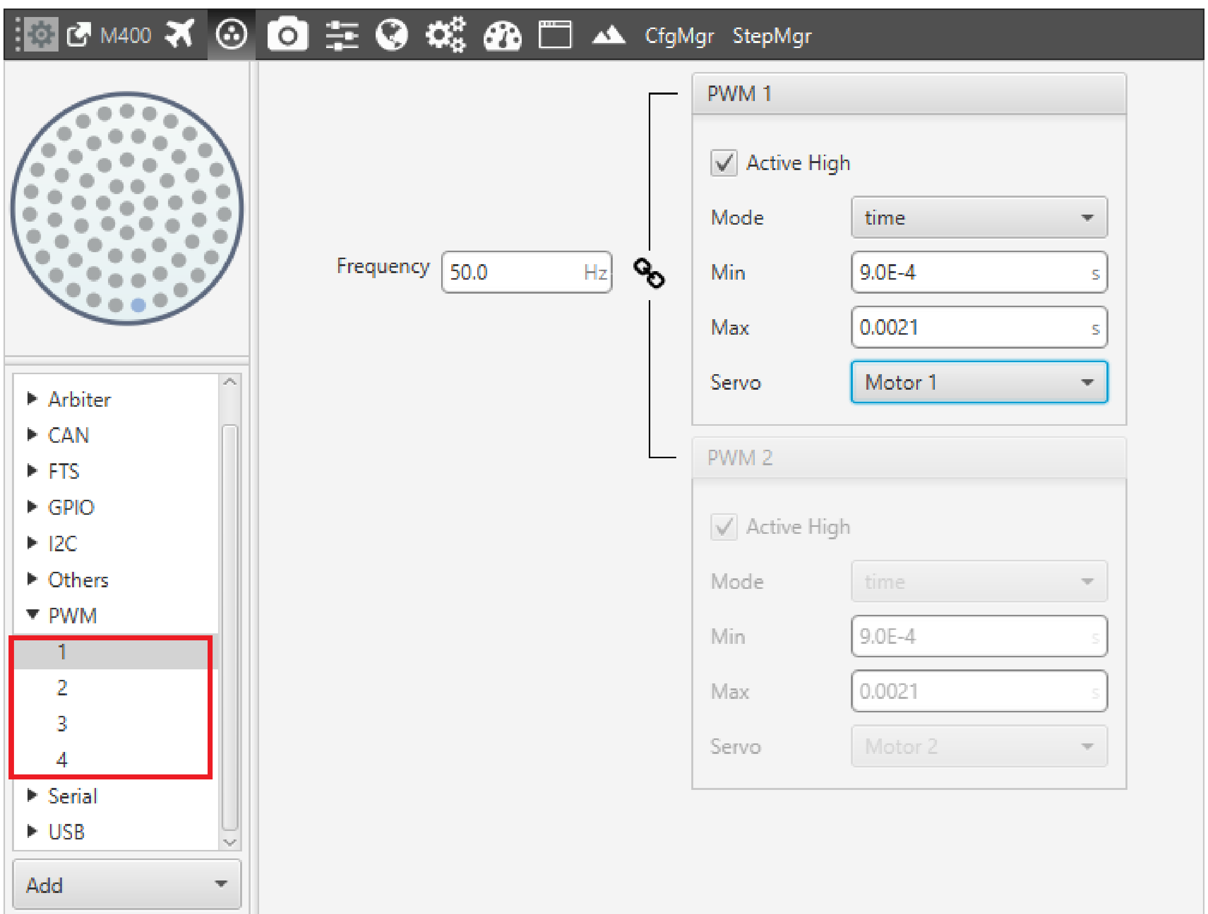
11.2.2.3.1 Servo Configuration

The M400 configuration process can be performed by using a VerontePipe software version connected with the hardware system and the Autopilot as explained in the section *Hardware Installation* of the manual.

11.2.2.3.1.1 Servos Output

The first step of the process is the servos configuration.

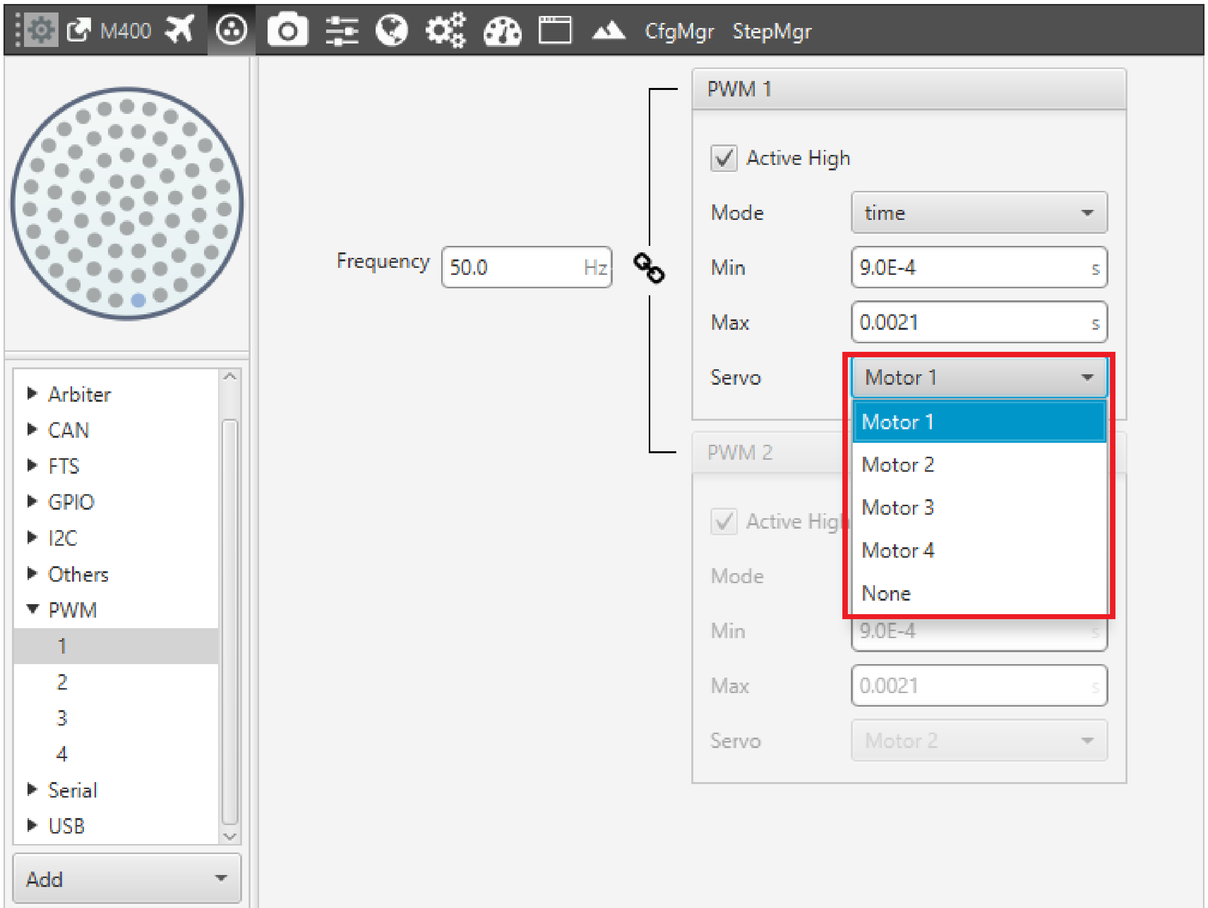
In this case, the controls of the airplane are the four electric motors. Motors can be controlled by changing their thrust, so each one of them corresponds to a pin of the connector and they must be positioned in the same order in an S vector who represents the **Actuator Output**. It is possible to connect any pin to any command but the easiest way to perform it and avoid confusion is by following the pins number.



Output-pin connections

In this case, it is possible to use 4 pins:

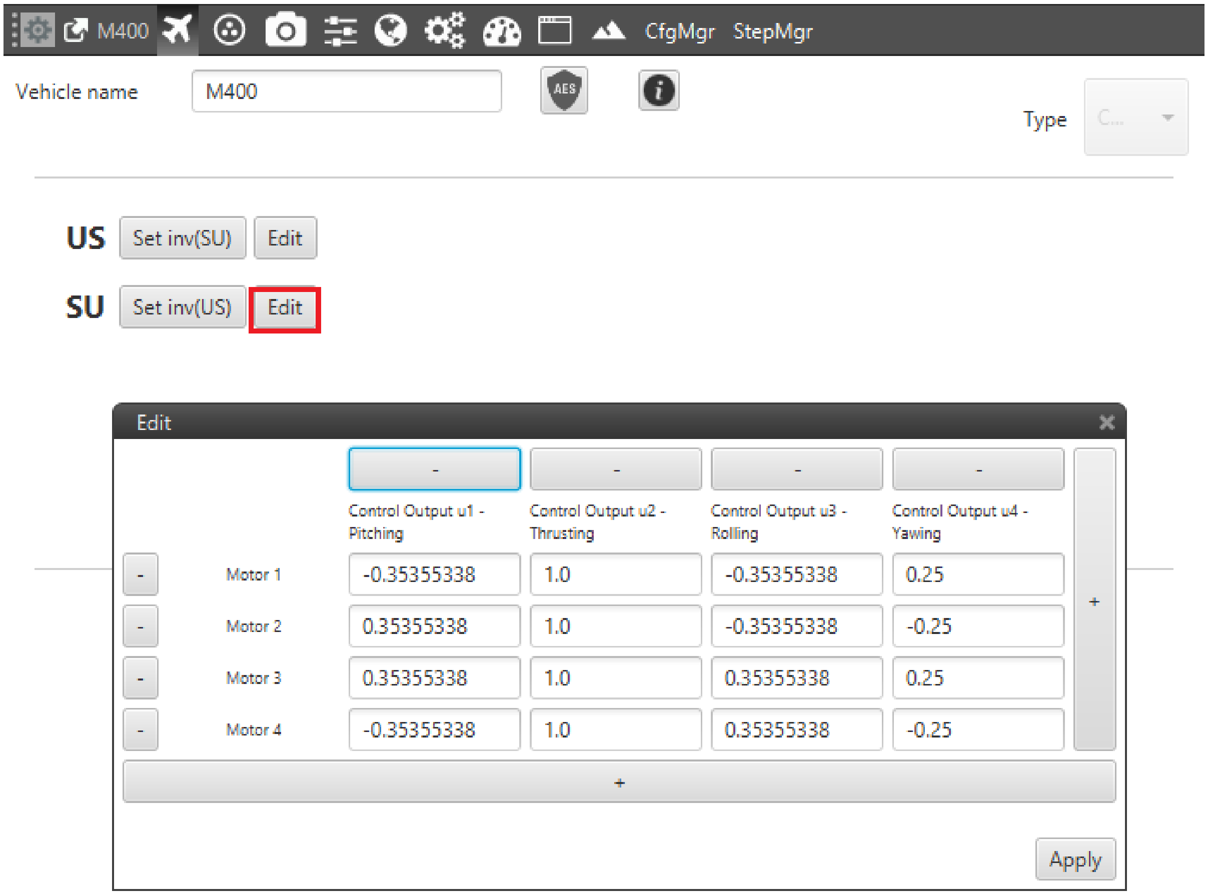
- Output 1 – Actuator Output s1 (Motor 1)
- Output 2 – Actuator Output s2 (Motor 2)
- Output 3 – Actuator Output s3 (Motor 3)
- Output 4 – Actuator Output s4 (Motor 4)



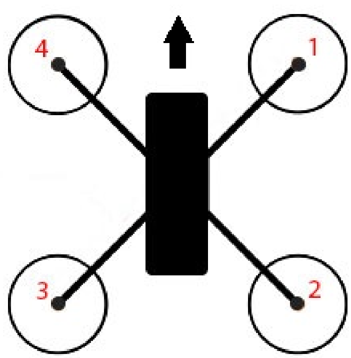
Output 1 – pin 1 configuration

11.2.2.3.1.2 SU Matrix

At this point, the **S** vector is defined and the **SU** matrix can be edited. By clicking on **Edit** it is possible to configure the relation between the controller outputs (**U** vector) and the servo movements (**S** vector).



Output 1 – SU matrix edit



SU matrix and motors numbers

The M400 is configured as follow:

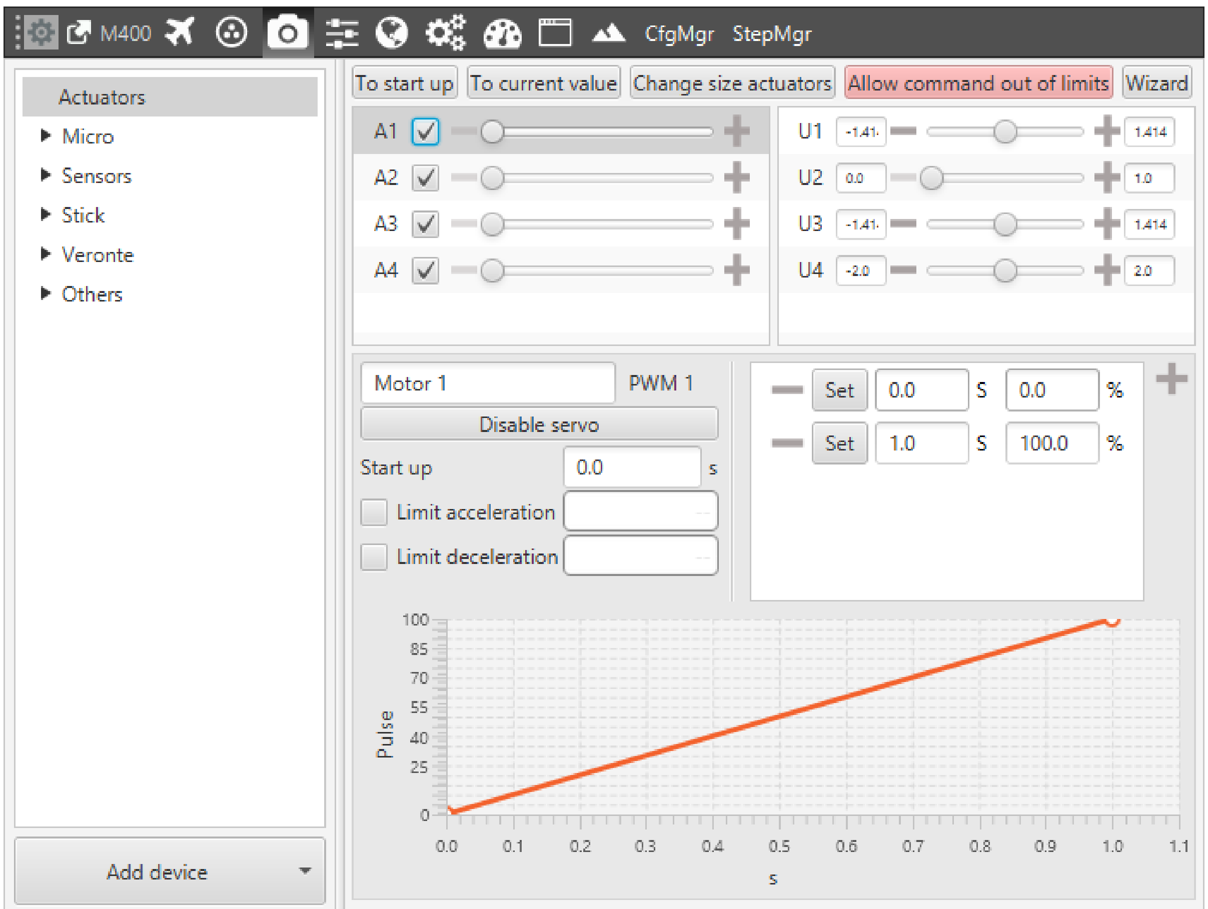
1. **Pitch Angle Control:** Control Output 1 is configured to perform a positive pitch angle change when motors 1-4 decrease their RPMs and motors 2-3 increase RPM value.

2. **Thrust Control:** the Control Output 2 is the one that allows a thrusting change (0-1).
3. **Roll Angle Control:** Control Output 3 is configured to perform a positive roll angle change when motors 1-2 decrease their RPMs and motors 3-4 increase RPM value.
4. **Yaw Angle Control:** Control Output 4 is configured to perform a positive yaw angle change when motors 2-4 decrease their RPMs and motors 1-3 increase RPM value, in this case with different proportions.

Warning: This panel shows the reference system of the aircraft too. It must be positioned in the same way of the Autopilot's one. If it results different, it can be edited by clicking on the corresponding axis in order to reverse its direction.

11.2.2.3.1.3 System Trim

As a final step, the system has to be trimmed. In this case, each motor will have a minimum and maximum value as shown in the picture.



M400 trim

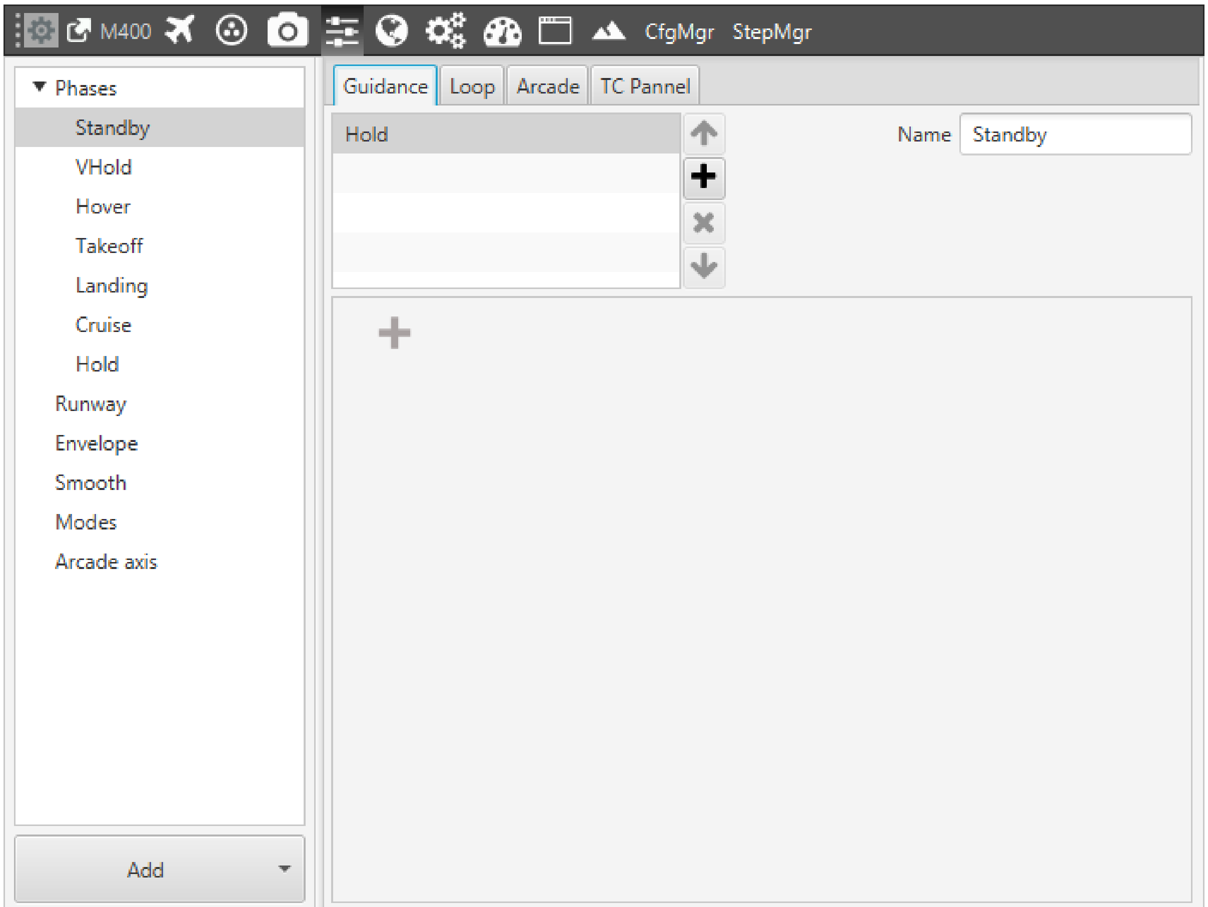
- **Minimum:** 0.
- **Maximum:** 1.

11.2.2.3.2 Mission Phases

In this section, it will be explained the M400 typical mission profile and the mission phases will be detailed.

11.2.2.3.2.1 Standby

Standby phase is a preliminary phase of the operation. The guidance is simply a **Hold** with no change. Normally, the automation which allows the system to pass to Standby phase requires the GPS signal.

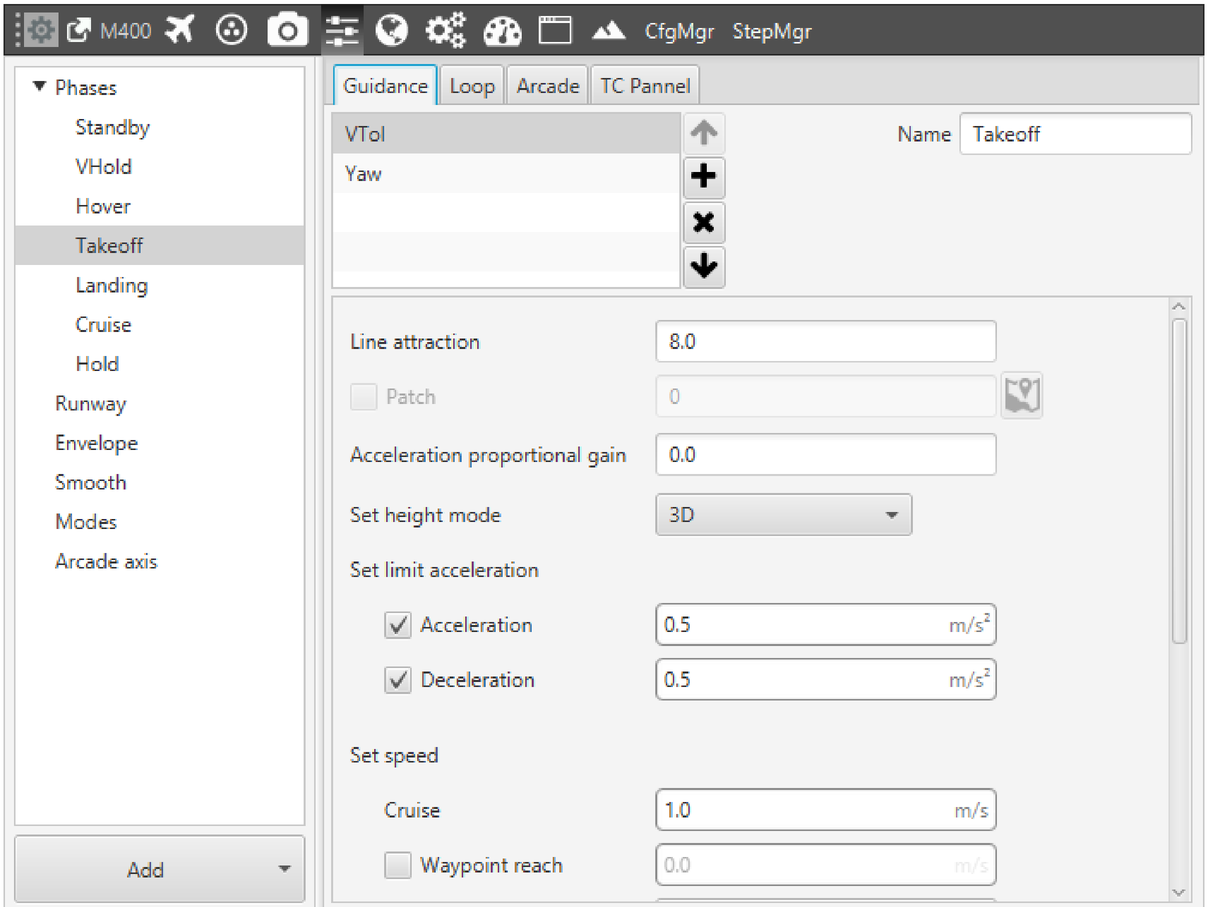


Standby phase panel

11.2.2.3.2.2 Takeoff

Take-off phase is composed of two Guidances: VTol and Yaw. The first one is a vertical guidance and where following parameters are set:

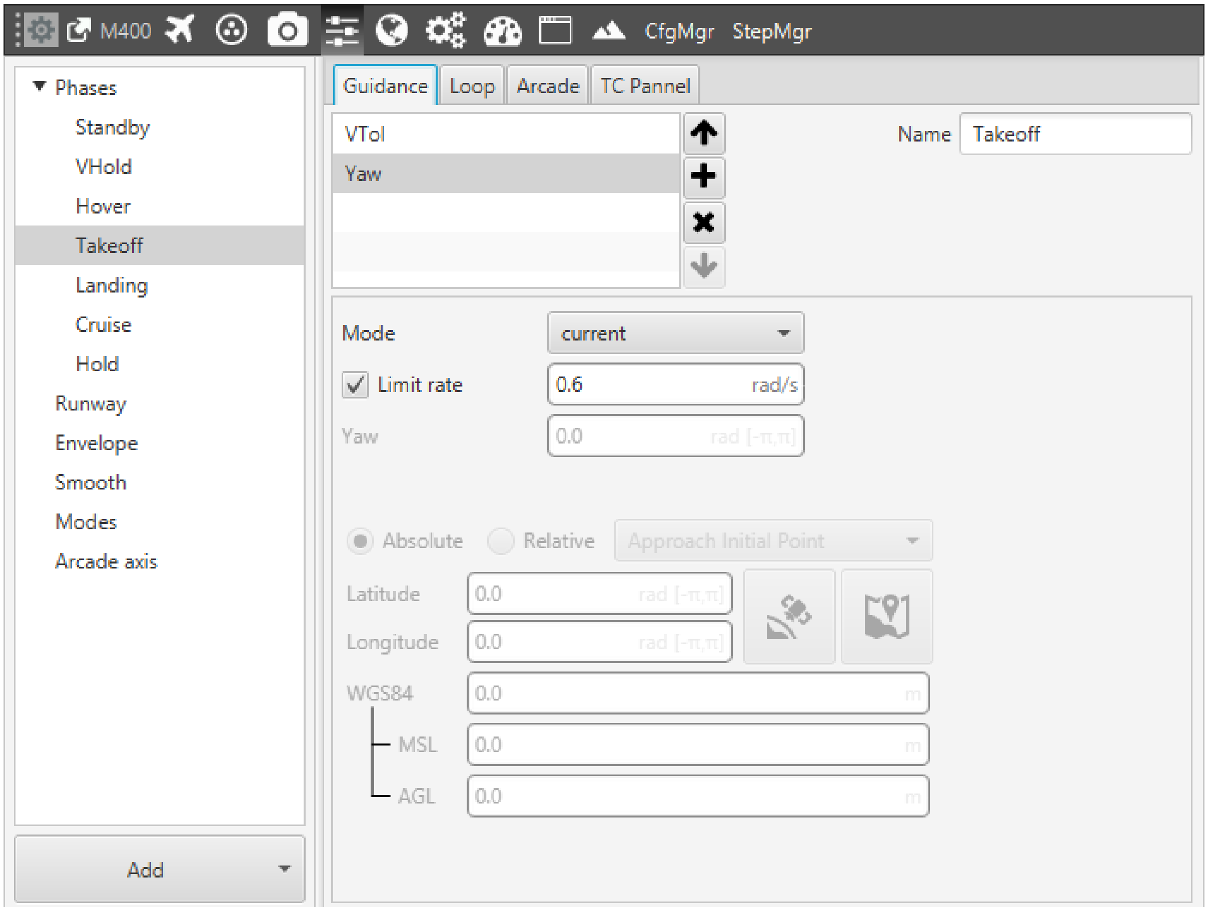
- **Line attraction:** 8.0 [m].
- **Set Speed:** 1.0 [m/s].
- **Horizontal Hover Gain:** 0.34.
- **Vertical Hover Gain:** 0.1.
- **Safe Distance:** -10 [m] and Relative (Positive down). With this value the platform will ascend.



VTol panel

Yaw guidance allows choosing the M400 orientation during takeoff:

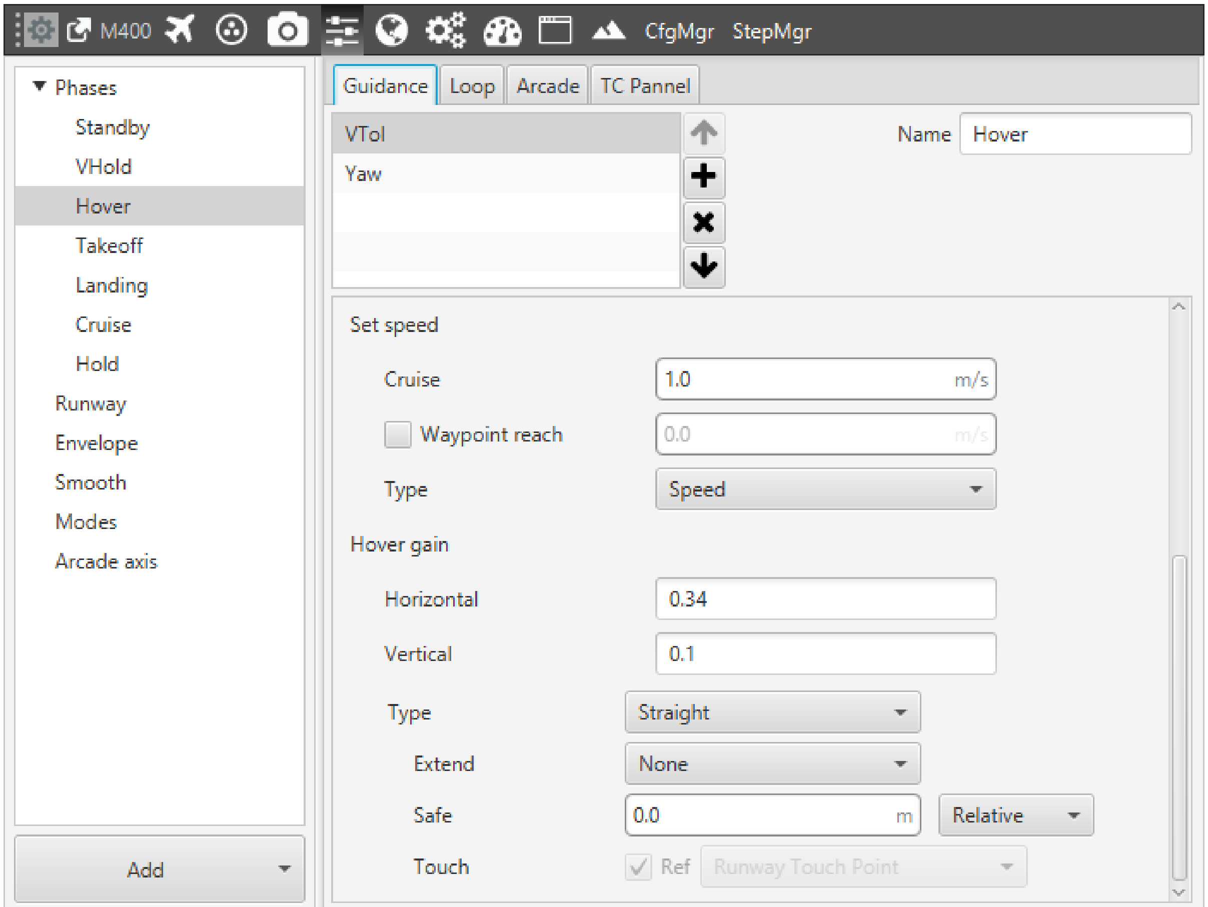
- **Mode:** set to **Current**.
- **Limit rate:** limited rotation speed of 0.6 [rad/s].



Yaw panel

11.2.2.3.2.3 Hover

Hover phase is configured to allow the multicopter maintain the position in the air.



Hover phase panel

The Guidance is a **VTol** as the one in Take Off phase but, in this case:

- **Safe distance:** set at 0 [m] and Relative.
- **Horizontal Hover Gain:** 0.34.
- **Vertical Hover Gain:** 0.1.

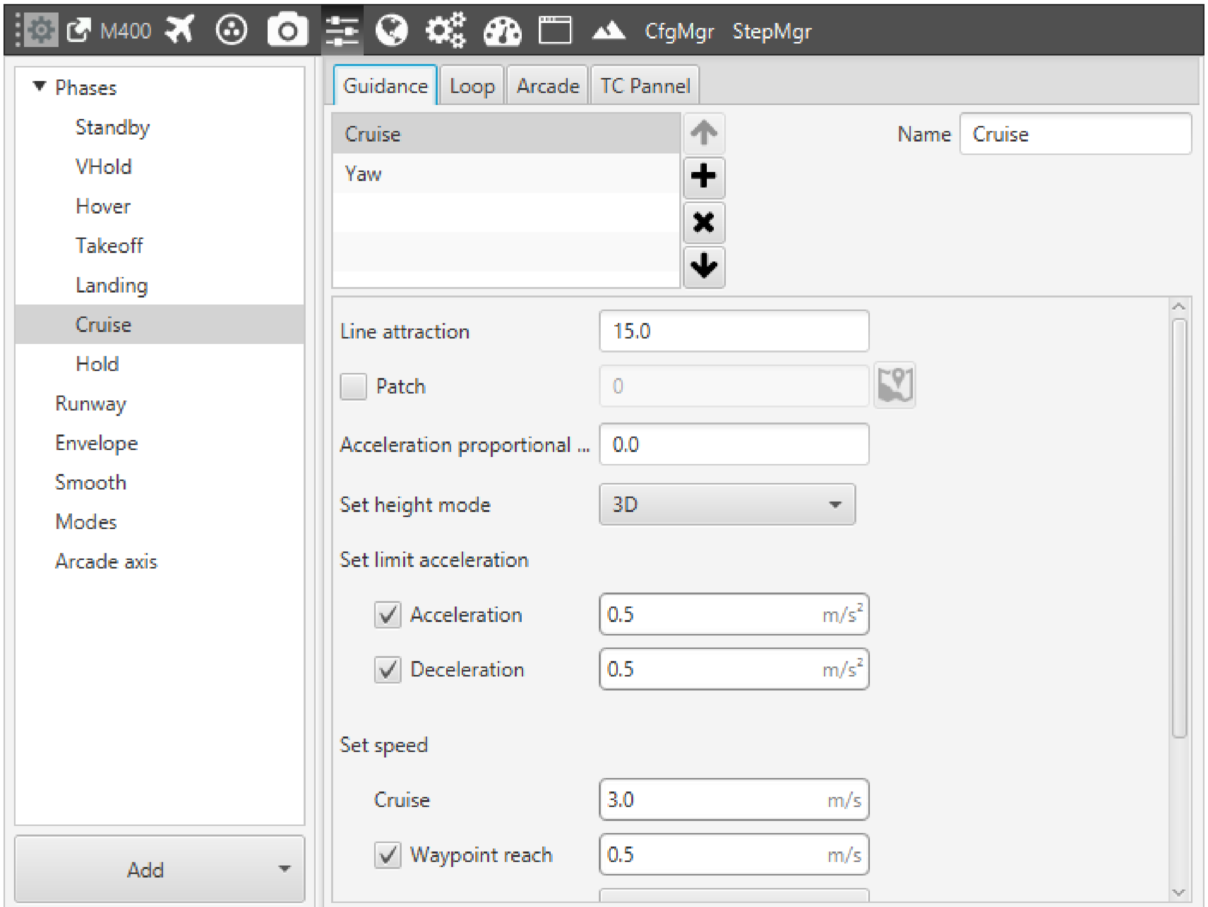
The **Yaw** guidance is the same of the Take Off phase.

11.2.2.3.2.4 Cruise

In this phase, the Guidances are Cruise and Yaw.

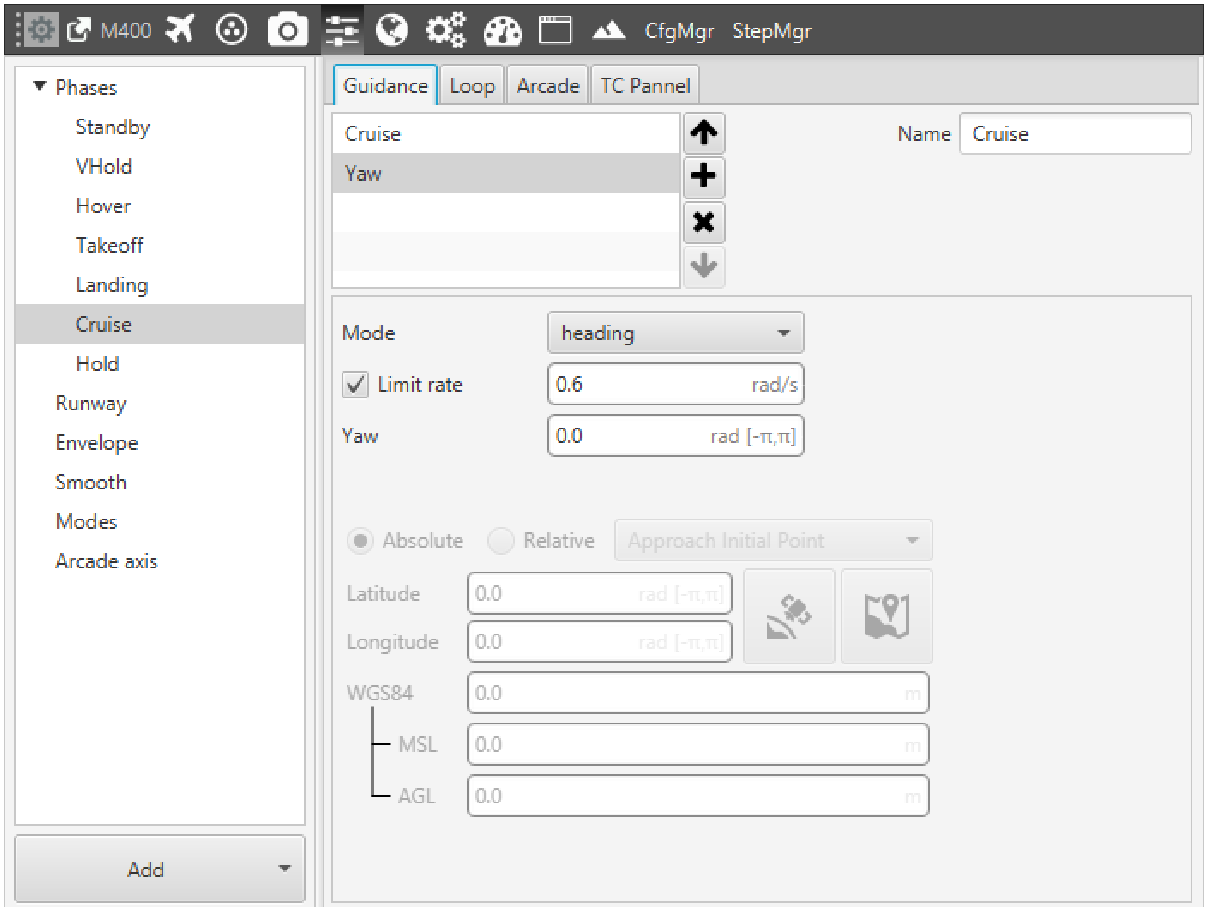
The parameters to set are the Cruise Speeds (Cruise and Waypoint reach) and Acceleration limits.

- **Acceleration limit:** 0.5.
- **Deceleration limit:** 0.5.
- **Cruise Speed:** 3.0 [m/s].
- **Waypoint reach:** 0.5 [m/s].



Cruise panel

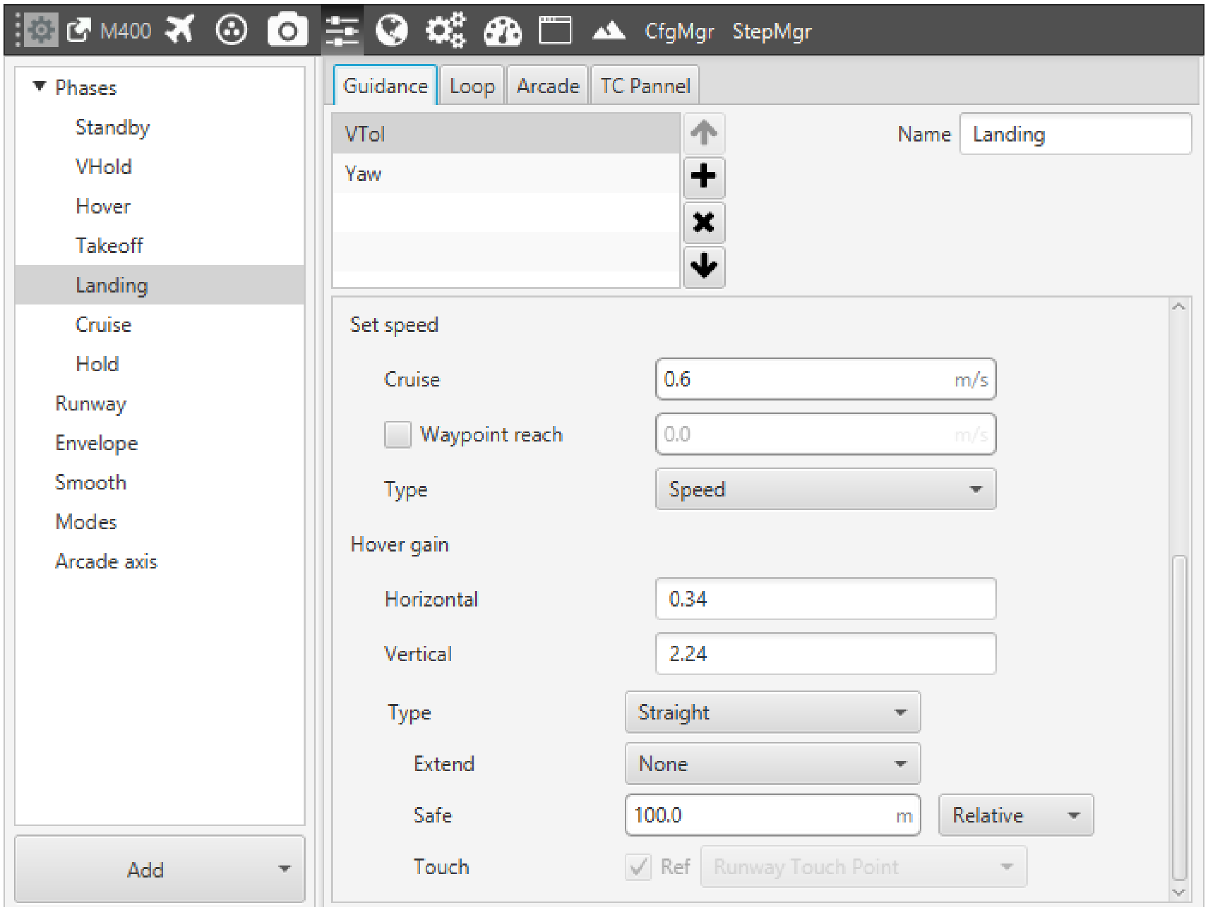
Yaw Guidance is set in order to make the multicopter controlling yaw angle following the heading angle.



Yaw Guidance panel

11.2.2.3.2.5 Landing

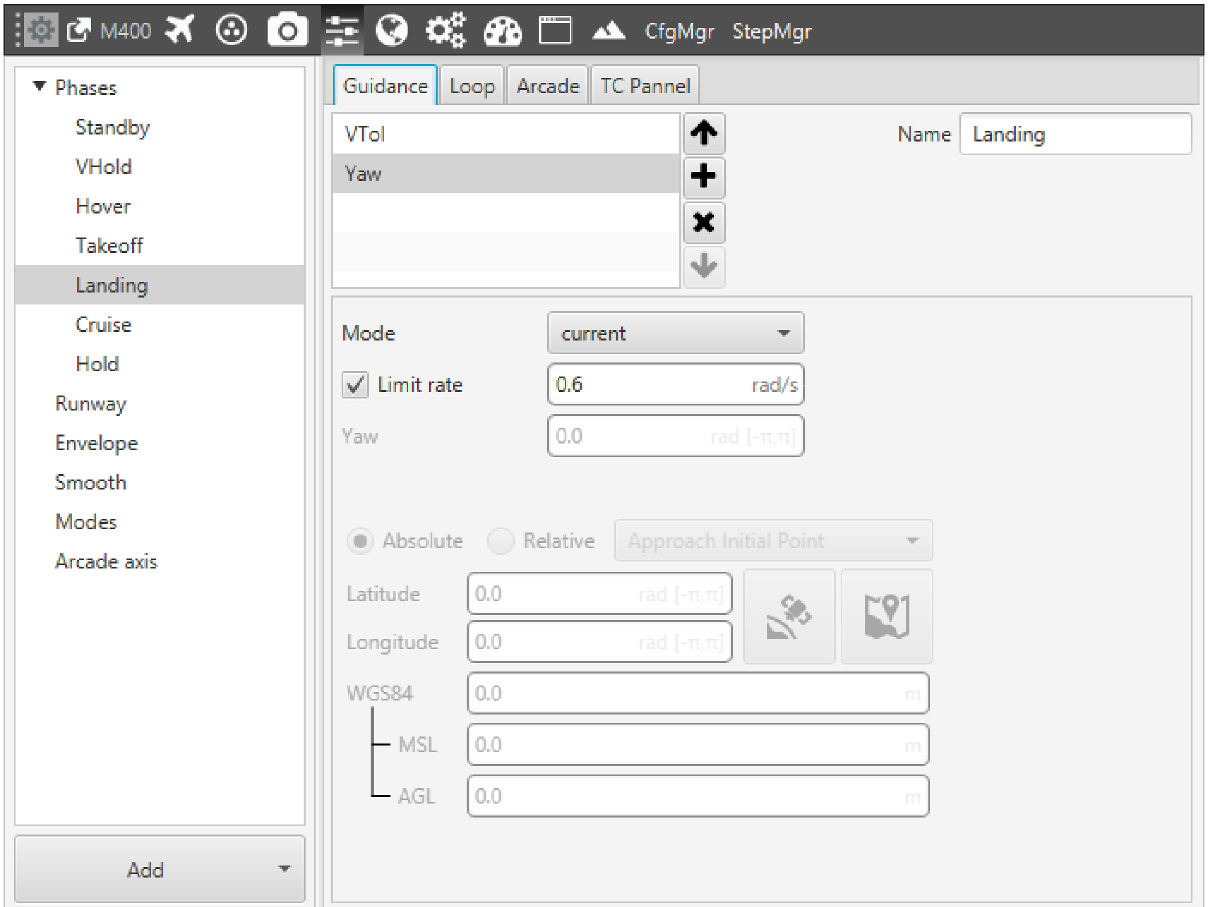
This phase has the same settings of the Take-off with some changes.



Landing phase panel

- **Line Attraction:** kept at 8.
- **Set Speed-Cruise:** 0.6 [m/s].
- **Horizontal Hover Gain:** 0.34.
- **Vertical Hover Gain:** 2.24.
- **Safe Distance:** 100.0 [m] and Relative (Positive Down). In this case, we are supposing that at the instant when the Landing phase starts, the multicopter is flying at AGL<100m. It is possible select Extend – Down and the aircraft will descend till user desire.

Yaw Guidance is configured in order to let the quadcopter with the Current yaw angle and limiting the yaw rate at 0.6 [rad/s].



Yaw Guidance panel

11.2.2.3.2.6 VHOLD

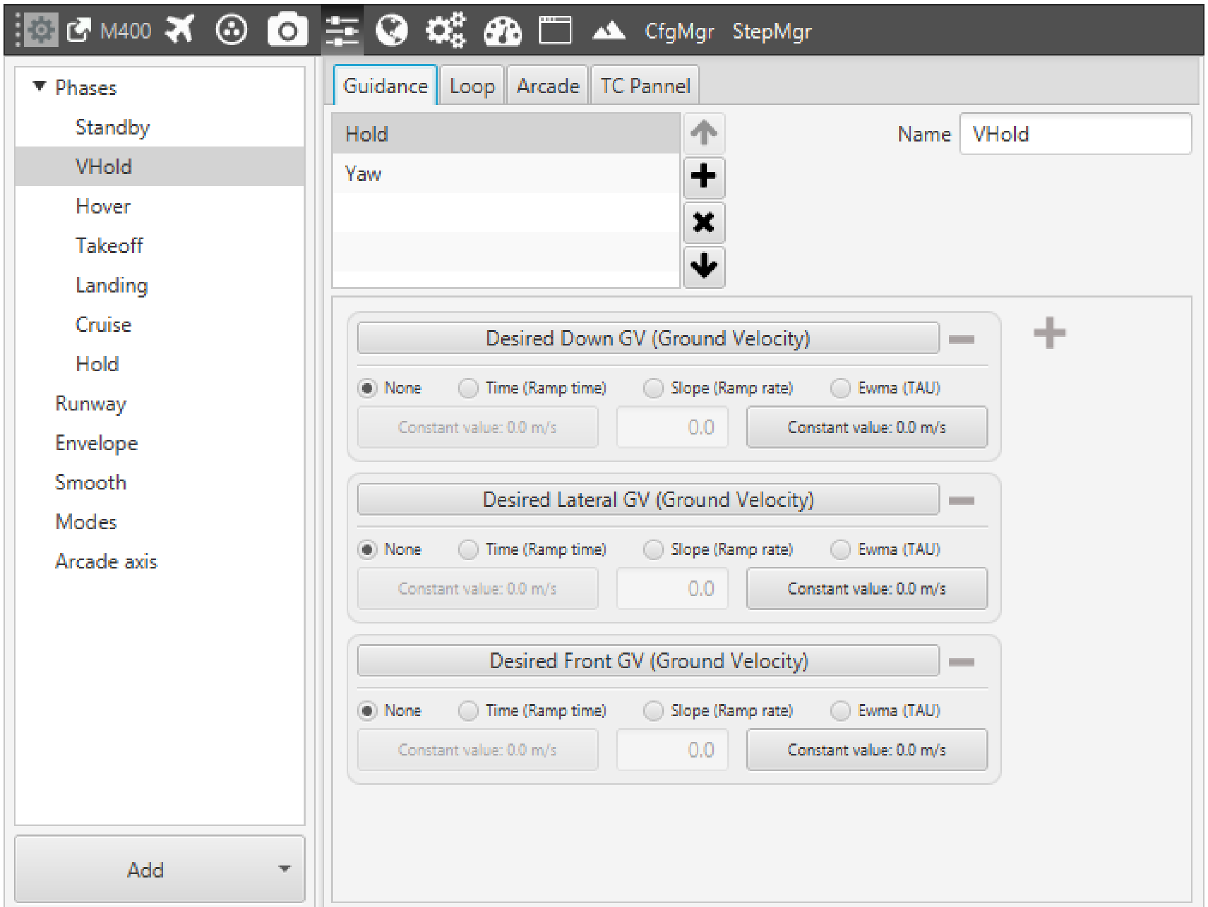
VHold phase is used to perform a Stick controlled flight. The phase guidance is composed of **Hold** and **Yaw** guidances.

Hold guidance is used to keep the following values at a specific value:

- **Desired D0wn GV:** at 0.0 [m/s].
- **Desired Lateral GV:** at 0.0 [m/s].
- **Desired Front GV:** at 0.0 [m/s].

In this way, if the stick control is not active, the multicopter will keep its position in the air without the need of phase changing.

Yaw Guidance is set on Current. In this phase, the Arcade allows controlling the platform depending on the set gains.

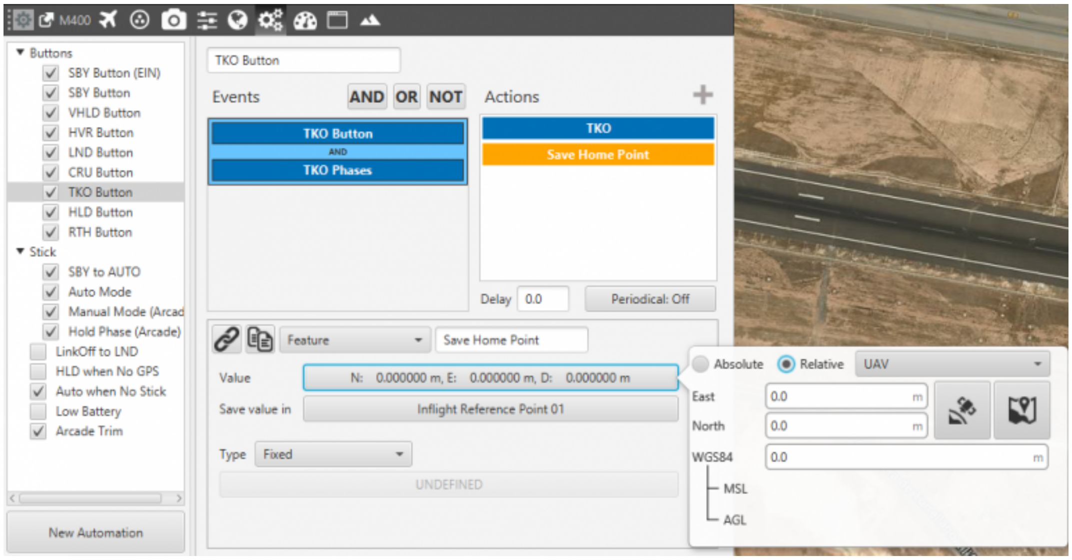


Manual phase panel

11.2.2.3.2.7 Return to Home

This operation is used to make a multicopter return to a point, considering it as Home. In order to perform this, is necessary to create two automations and a new phase, all process is explained below.

1. First create an Automation to save the current position, this has to be the one where the aircraft take-off (Home). This action can be added when creating the Take Off Button. See section *Veronte Panel Buttons* for more information about creating a button.

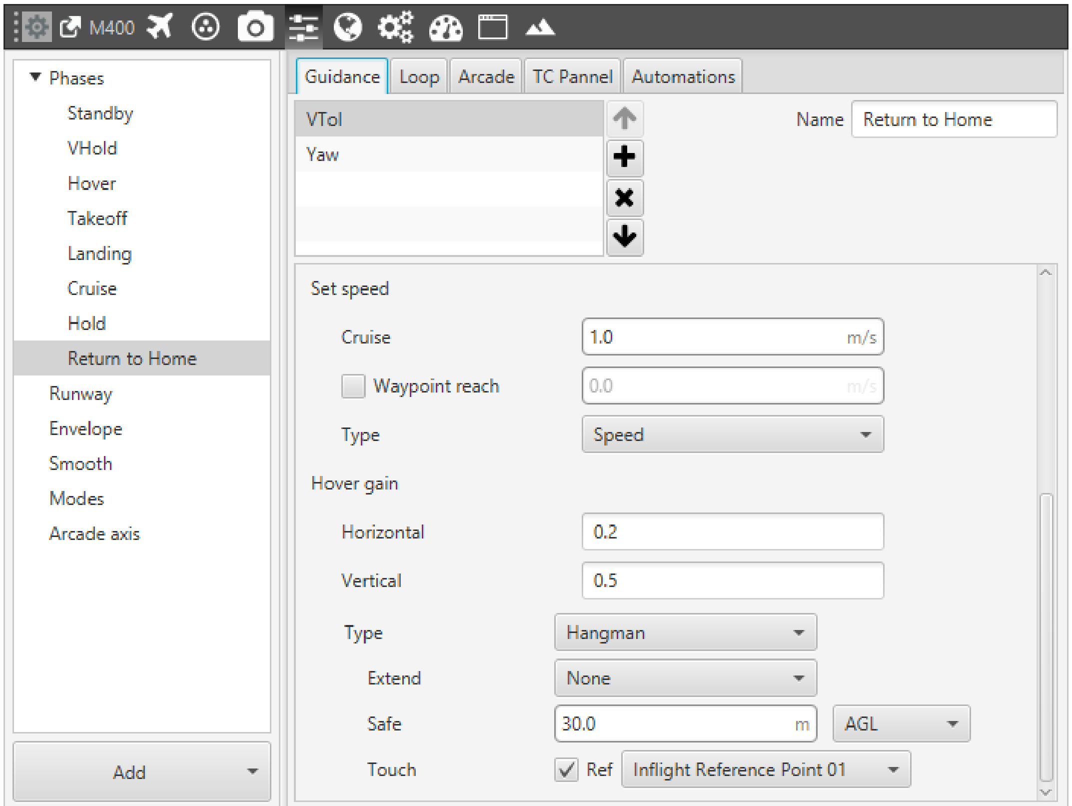


Automation – TKO Button

It is only necessary to add the action Feature and configure it as shown below:

- **Value:** select Relative and UAV. The parameters of position have to be set to 0, it means that the value to save is the UAV position.
- **Save Value in:** select Inflight Reference Point.
- **Type:** fixed.

2. The following step consists in creating a new phase. This phase can be named as **Return to Home** and it is necessary to include two guidances **VTol** and **Yaw**. VTol guidance has to be configured as follow:



Phase – Return to Home

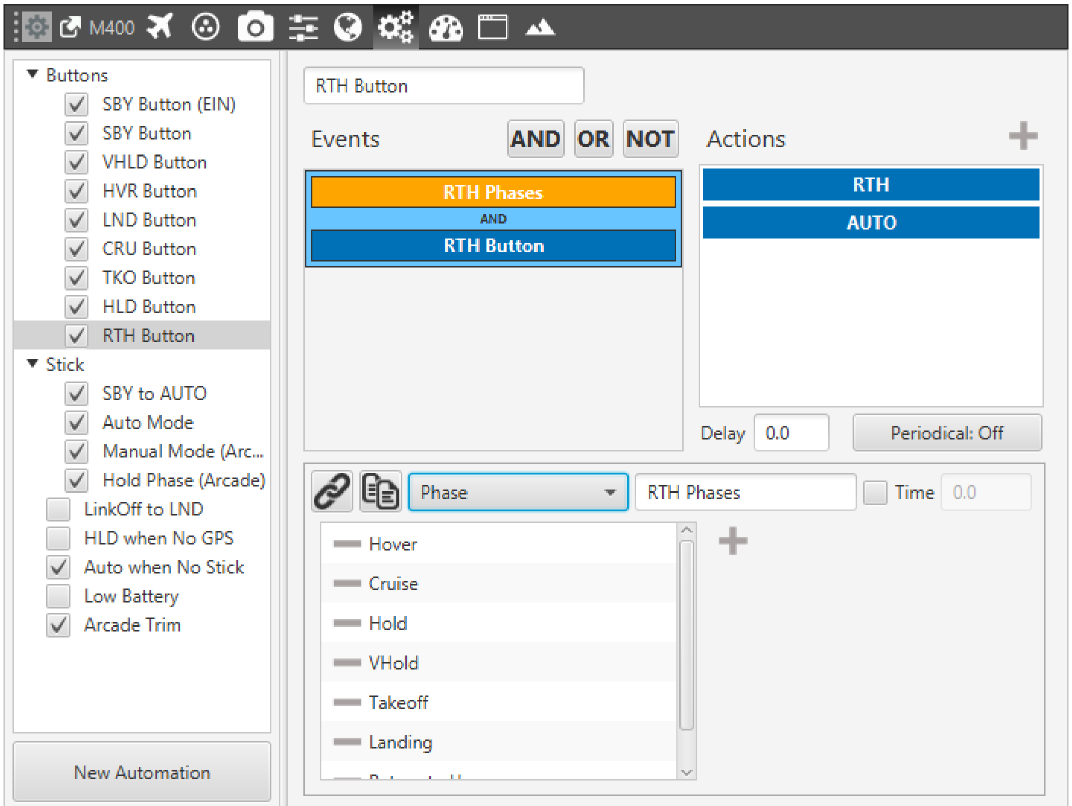
See section *VTOL Guidance* for more information about configuring this guidance. It is worth noting the following parameters:

- **Type:** Hangman is selected.
- **Extend:** None
- **Safe:** 30 [m] and AGL, this means that the aircrat will perfom the Hangman path reaching an altitude of 30 [m] AGL.
- **Touch:** here is defined the point where the platform will touch ground. This has been defined in the first step (Inflight Reference Point) and now it is possible to select it.

Yaw guidance can be set to **current** and user may want to limit the rate.

3. The last step consits in the creation of a Button on the Veronte panel. Go to Automation Panel and create the following Events and Actions.

   Two events have to created: **Phase** and **Button**.



Automation

The actions configured are:

- **Phase:** here is selected the phase that the platform will enter in i.e, Return To Home.
- **Mode:** AUTO is selected.

Finally, user only have to click on the Button which appear on Veronte Panel to perform a Return Home.

M400

This M400 multirotor has a quadcopter structure. Its control is performed by using a differential thrust for each motor depending on the desired attitude change:

- ALTITUDE
- PITCH ANGLE
- ROLL ANGLE
- YAW ANGLE

11.2.2.4 Helicopter T-Rex600

11.2.2.4.1 Servo Configuration

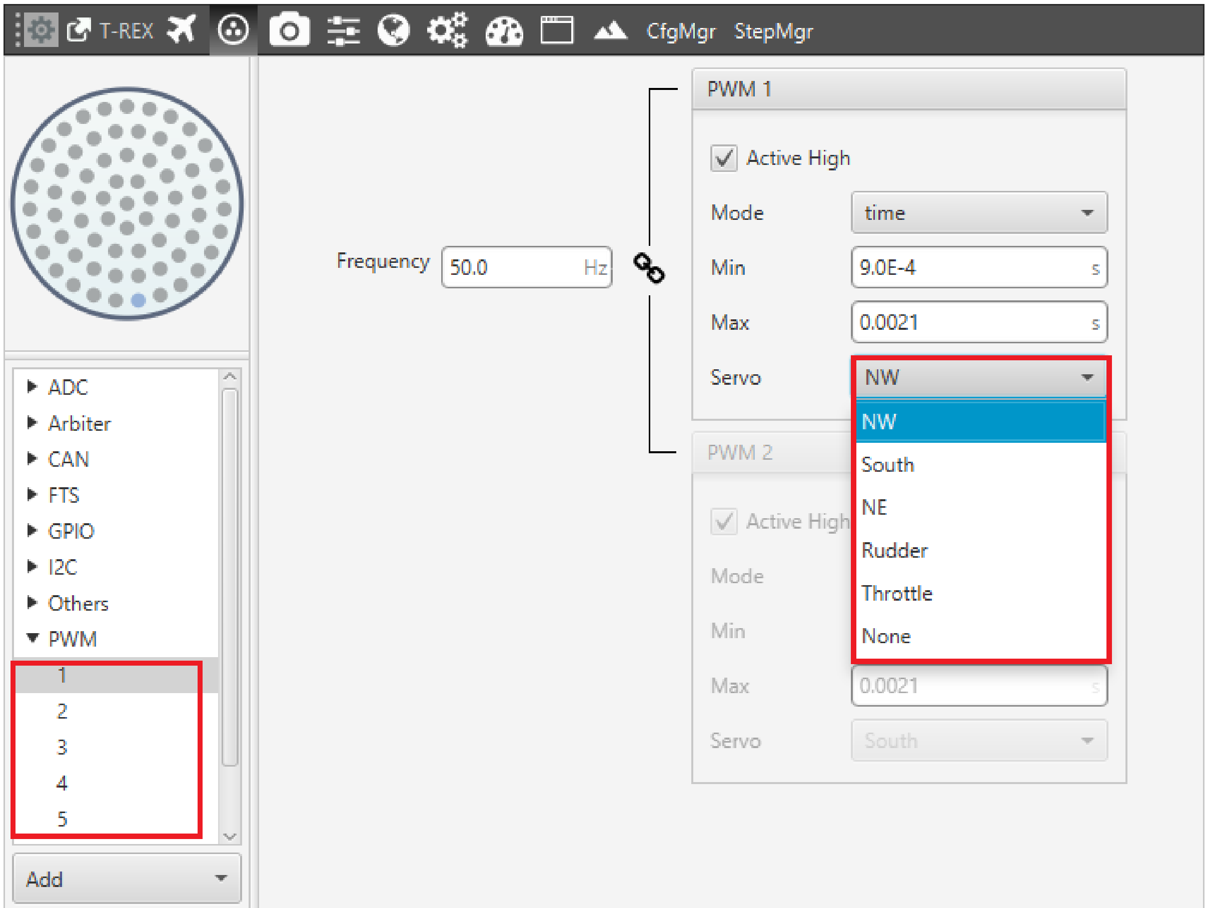
The T-Rex integration process can be performed by using Veronte Pipe connected with the hardware system and the Autopilot as explained in section *Aircraft Mounting* of the manual.

11.2.2.4.1.1 Servos Output

The first step of the process is the servos configuration. In this section will be explained how to set the servo-output matrix.

In this case, the controls of the helicopter are the cyclic, collective, RPM (main rotor) and blades pitch angle changing (tail rotor). To control the main rotor moving it is necessary to configure correctly the three servos that allow controlling the blades pitch angle and the one which changes the main rotor RPMs. In order to have the tail rotor control and to be able to change the blades pitch, another servo must be configured.

All the servos correspond to a pin of the connector and they must be positioned in the **S** vector who represents the **Actuator Outputs**. It is possible to connect any pin to any command but the easiest way to perform it and avoid confusion is by following the pins number.



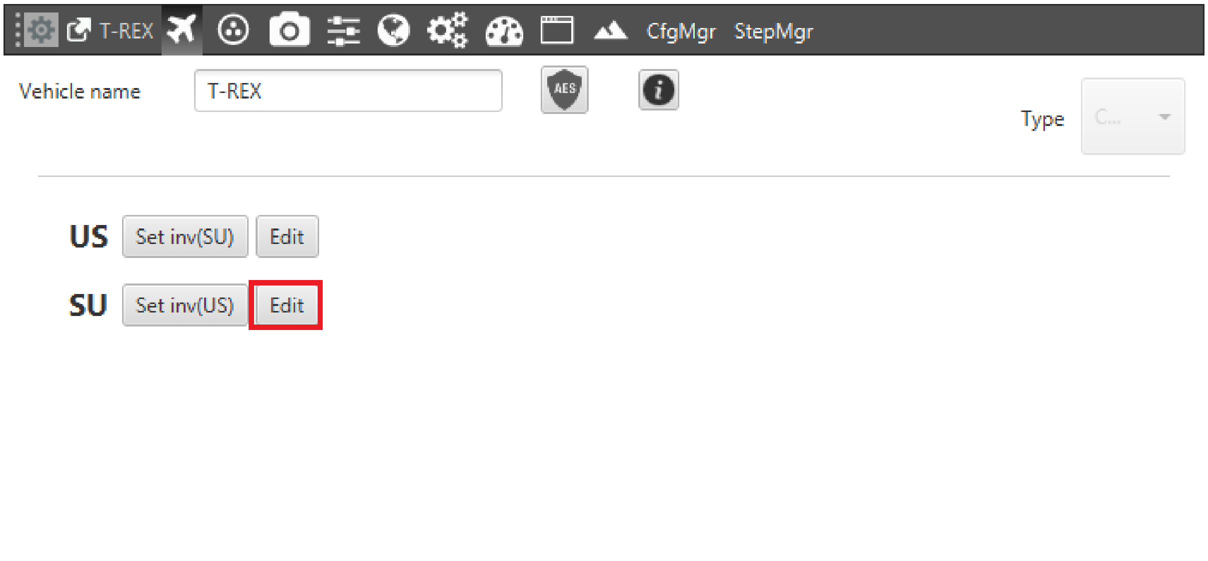
T-Rex Outputs

In this case, it is necessary to use 5 pins:

- **Output 1** – NW main rotor cyclic/collective
- **Output 2** – South main rotor cyclic/collective
- **Output 3** – NE main rotor cyclic/collective
- **Output 4** – Main rotor RPMs
- **Output 5** – Tail rotor blades angle changing

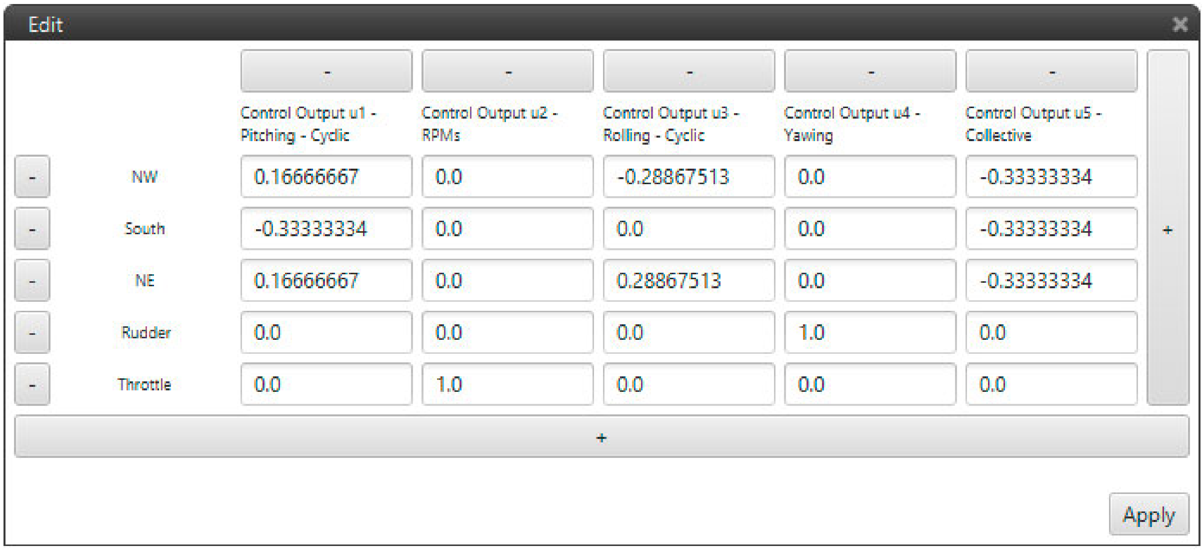
11.2.2.4.1.2 SU Matrix

At this point, the **S** vector is defined and the **SU** matrix can be edited. By clicking on **Edit** it is possible to configure the relation between the controller outputs (**U** vector) and the servo movements (**S** vector).



SU matrix edit

Warning: This panel shows the reference system of the aircraft too. It must be positioned in the same way of the Autopilot's one. If it results different, it can be edited by clicking on the corresponding axis in order to reverse its direction.

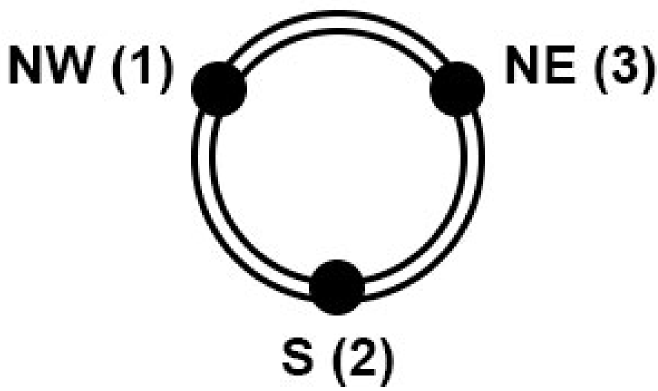


Edit

	Control Output u1 - Pitching - Cyclic	Control Output u2 - RPMs	Control Output u3 - Rolling - Cyclic	Control Output u4 - Yawing	Control Output u5 - Collective
NW	0.16666667	0.0	-0.28867513	0.0	-0.33333334
South	-0.33333334	0.0	0.0	0.0	-0.33333334
NE	0.16666667	0.0	0.28867513	0.0	-0.33333334
Rudder	0.0	0.0	0.0	1.0	0.0
Throttle	0.0	1.0	0.0	0.0	0.0

Apply

SU matrix



T-Rex swashplate actuators

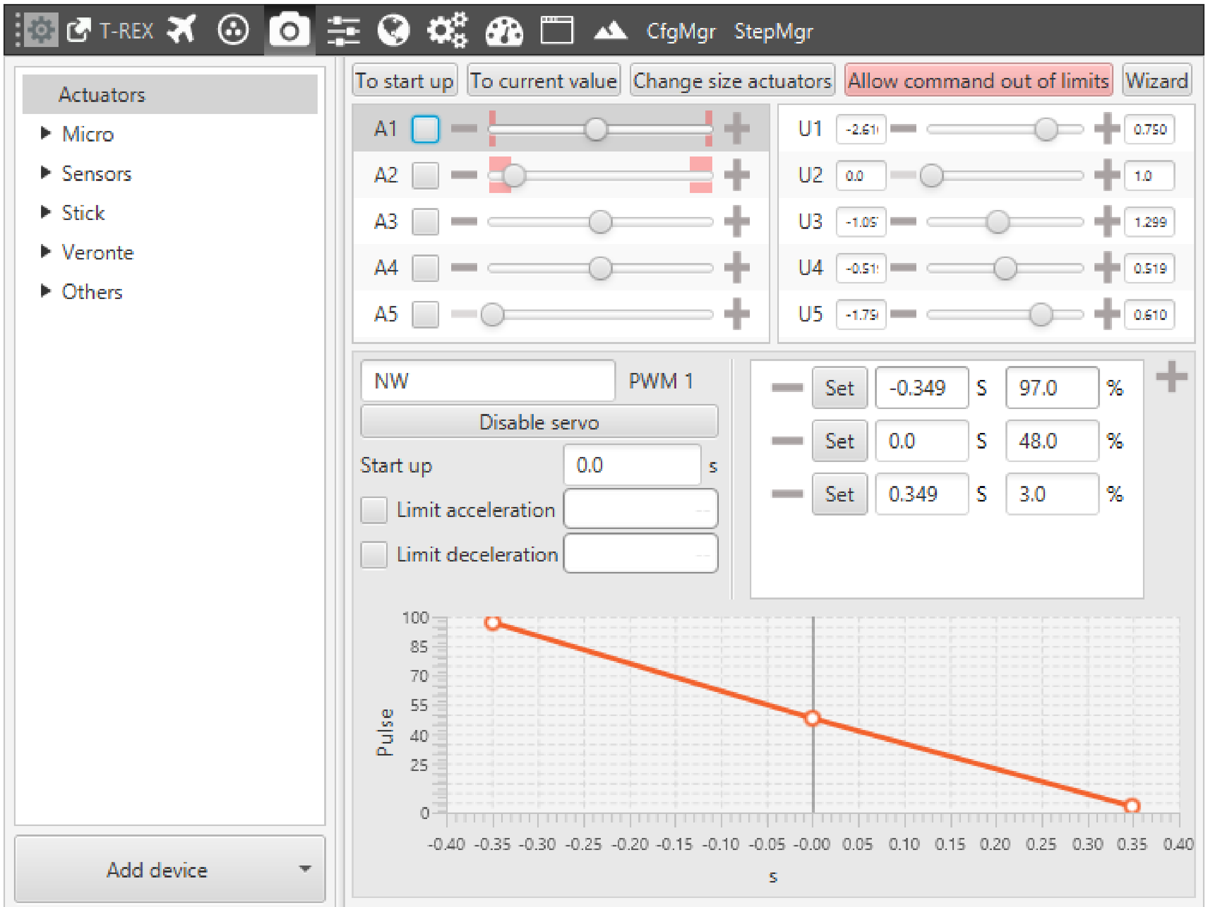
The previous image shows the swashplate of the T-Rex with the 3 actuators able to modify its rotation plane (X axis is positive up). As showed in the SU matrix image, controls are performed as follow:

1. **Pitch Angle Control:** Control Output 1 is configured to perform a positive pitch angle change when servos 1-3 move following Z axis positive and servo 2 moves in the opposite direction with double magnitude.
2. **RPM Control (Main rotor):** the Control Output 2 (servo number 5) is the one that allows an RPM change with the throttle moving (0-1).
3. **Roll Angle Control:** Control Output 3 is configured to perform a positive roll angle change when servos 1-3 move with an opposite direction (servo 2 is in the longitudinal plane of the aircraft, so it has not any influence).
4. **Yaw Angle Control:** Control Output 4 is configured to perform a positive yaw angle change when tail rotor decreases the blades pitch angle.

5. **Collective Control:** The altitude control is performed using the Control Output 5. A positive change of the pitch angle blade is performed when servos 1-2-3 move contemporaneously in the same direction.

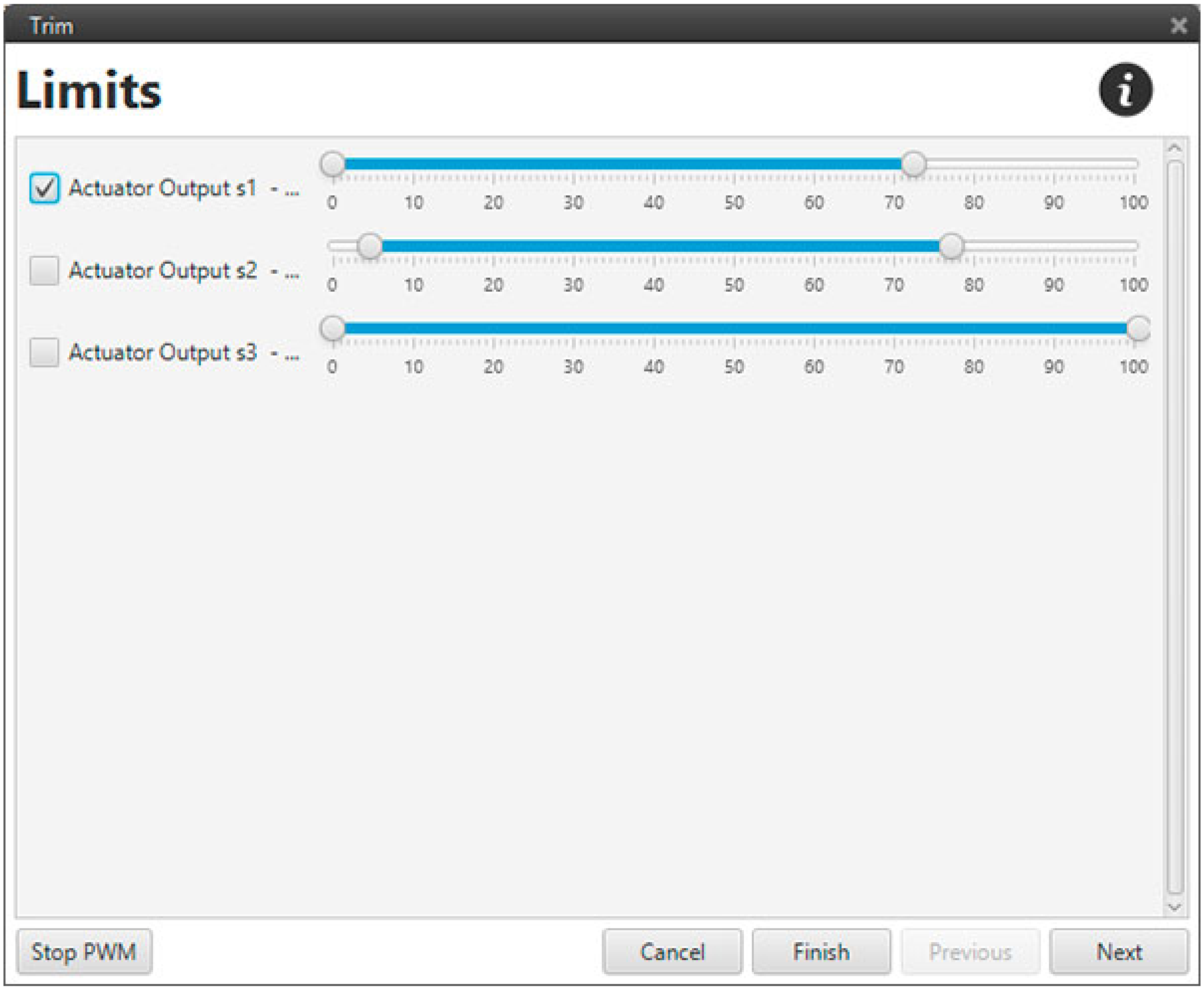
11.2.2.4.1.3 System Trim

As a final step, the system has to be trimmed. This can be performed by moving servos in three different positions: zero position, minimum and maximum (blade angles are usually limited physically). These positions must be inserted and saved in the software by clicking on Set when the actuator is in the desired position or introducing them manually.



Actuator 1 trimming

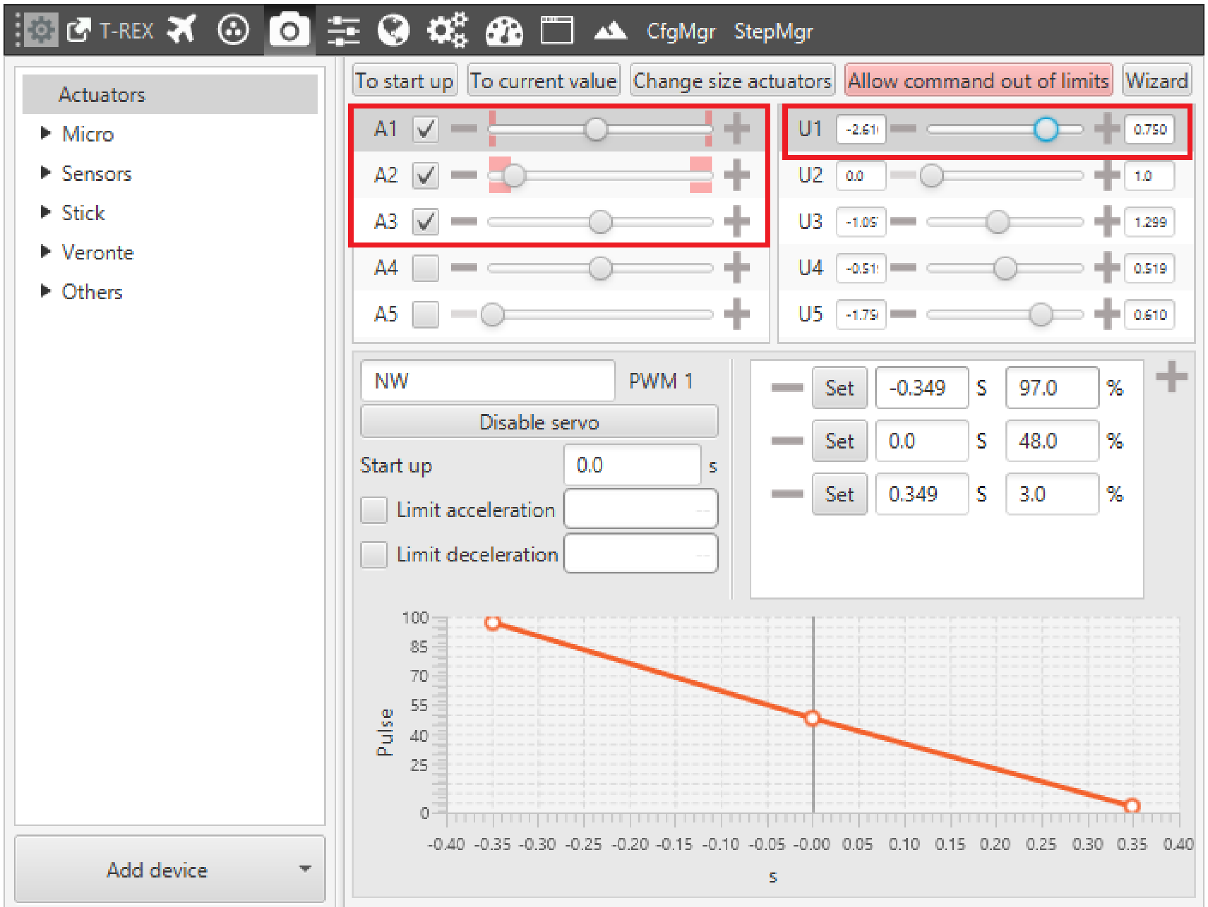
This procedure can be performed in the same way by using a **Wizard**. This tool allows moving actuators limits easily and finding the correct range.



Trim wizard tool

In order to perform a final checking, it is possible to select the desired channel and testing pitching, rolling, yawing and thrusting controls.

The image below shows a pitching output testing. By moving the U1 control (Pitching – Cyclic), main rotor servos change the position according to the reference system: positive corresponds to nose down and negative to nose up.



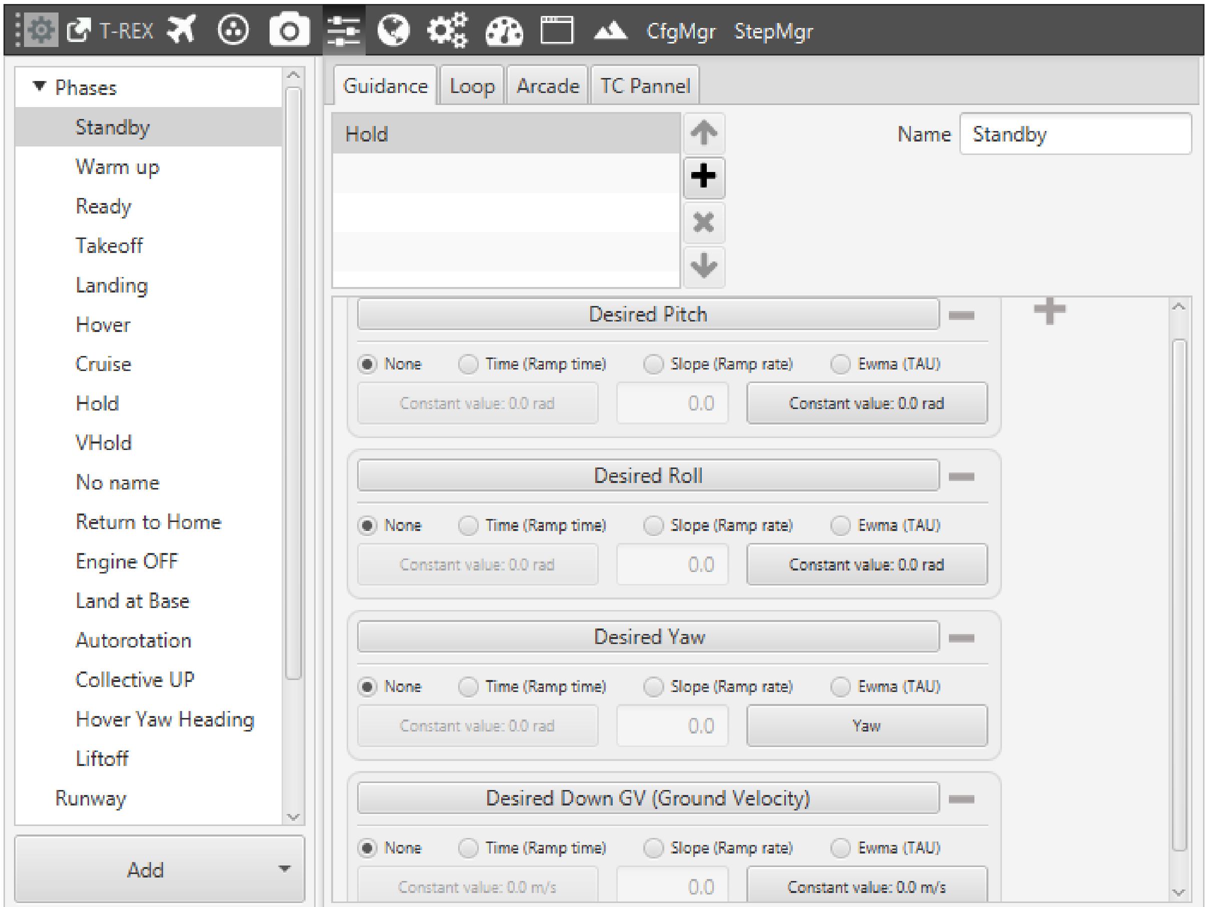
Pitching test

11.2.2.4.2 Mission Phases

In this section, it will be explained the T-Rex typical mission profile and the mission phases will be detailed.

11.2.2.4.2.1 Stanby

Standby phase is a preliminary phase of the operation. During this phase is possible to check, for example, if the aircraft is correctly controlling the attitude by moving it and watching the servos positions changing.



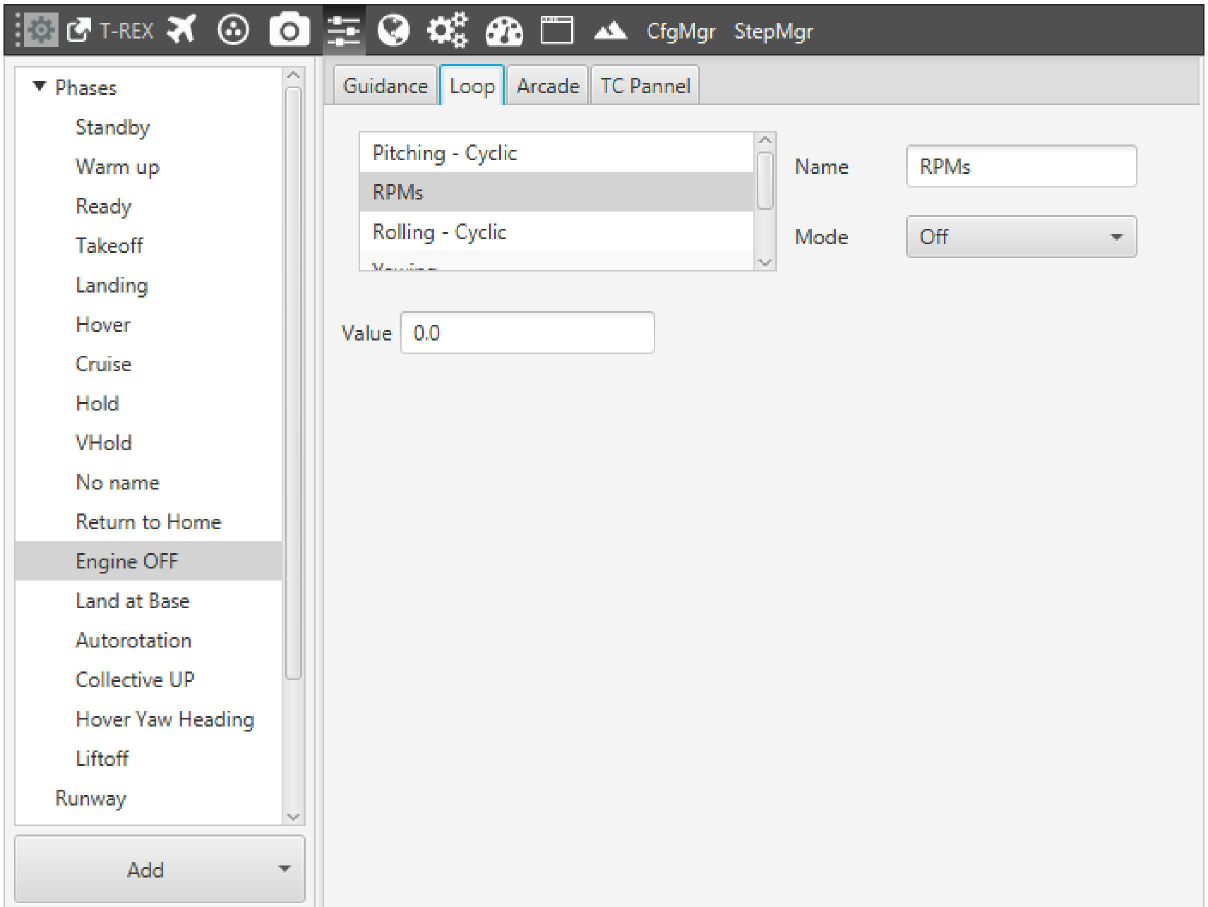
Standby phase panel

The guidance is a **Hold** of four variables:

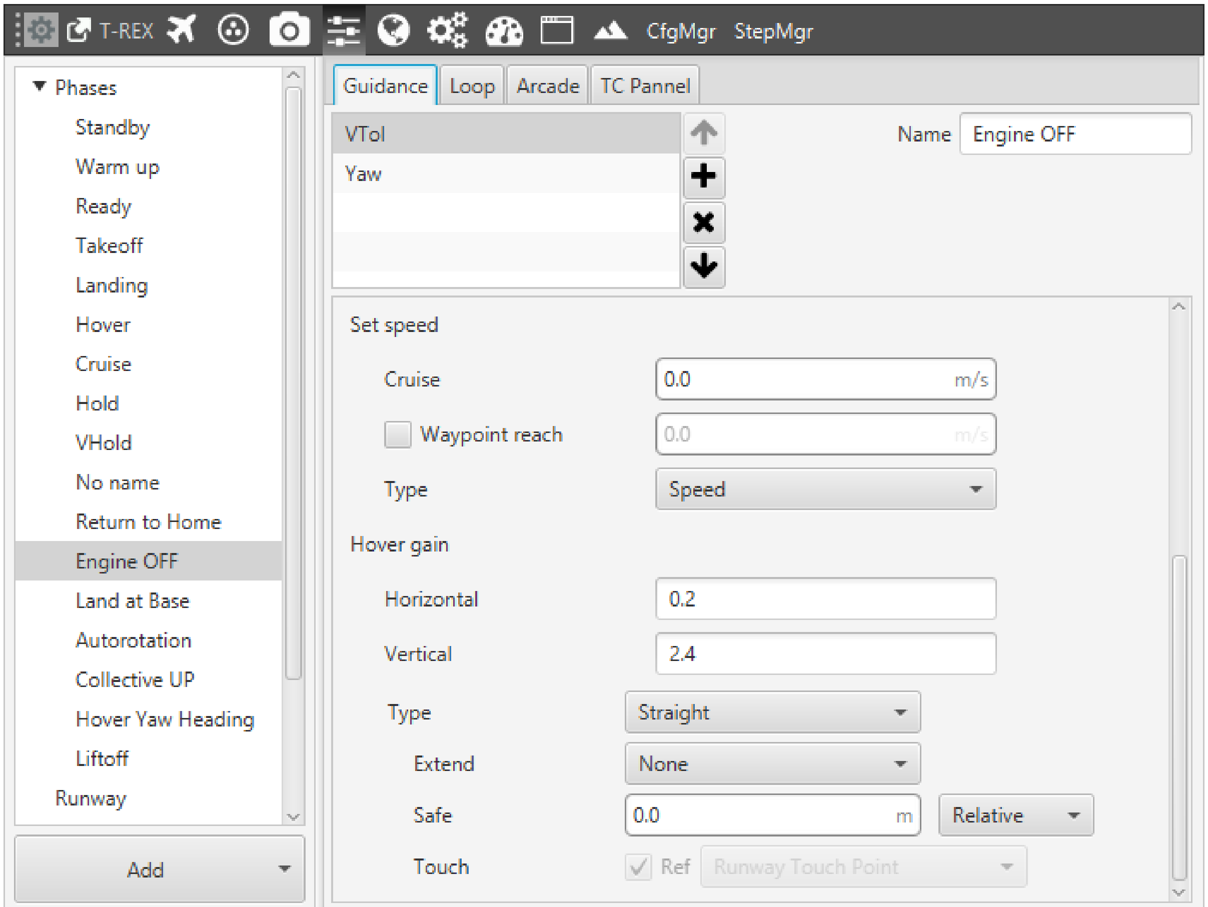
- **Pitch and Roll angles:** at 0 [rad].
- **Ground Speed:** at 0 [m/s].
- **Yaw angle:** set at the current.

11.2.2.4.2.2 Engine Off

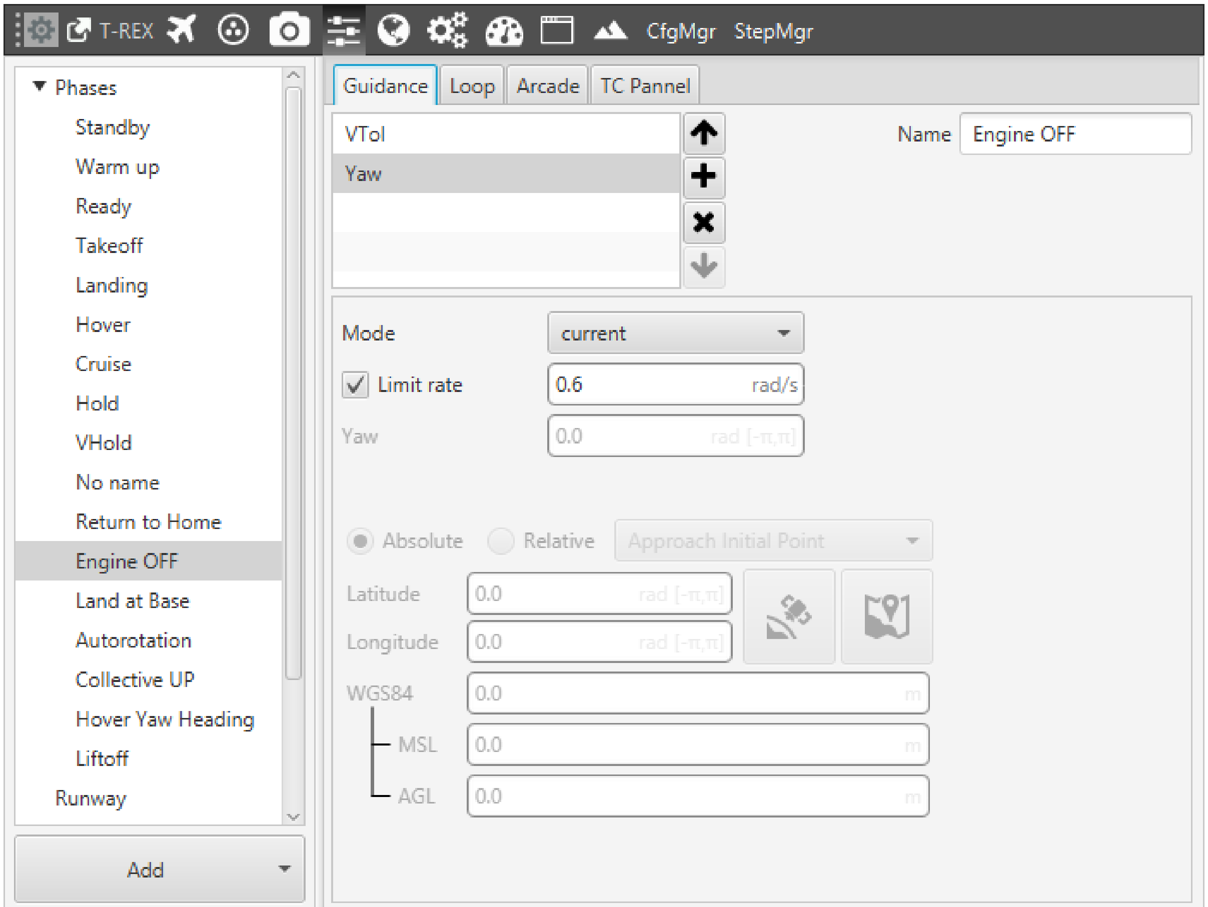
Engine OFF is the phase which allows the platform decreasing main rotor RPMs without attitude changes. In fact, the decreasing phase is not instantaneous and the autopilot needs to keep the control until the RPMs would not be able to make attitude changes. Following this fact, this phase has a **thrusting control** set at zero and VTol and **Yaw** Guidances.



Thrusting fixed control



VTol Guidance settings



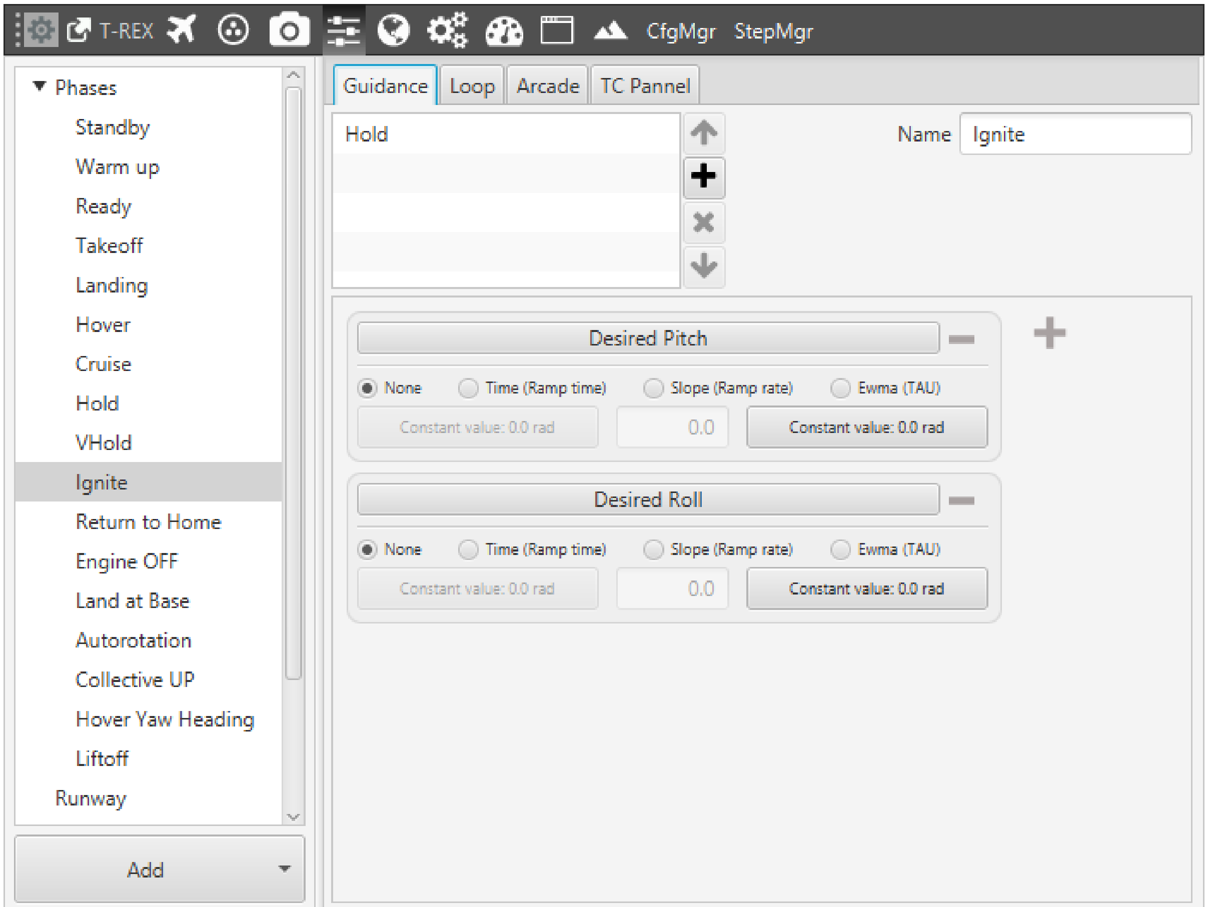
Yaw Guidance settings

11.2.2.4.2.3 Ignite

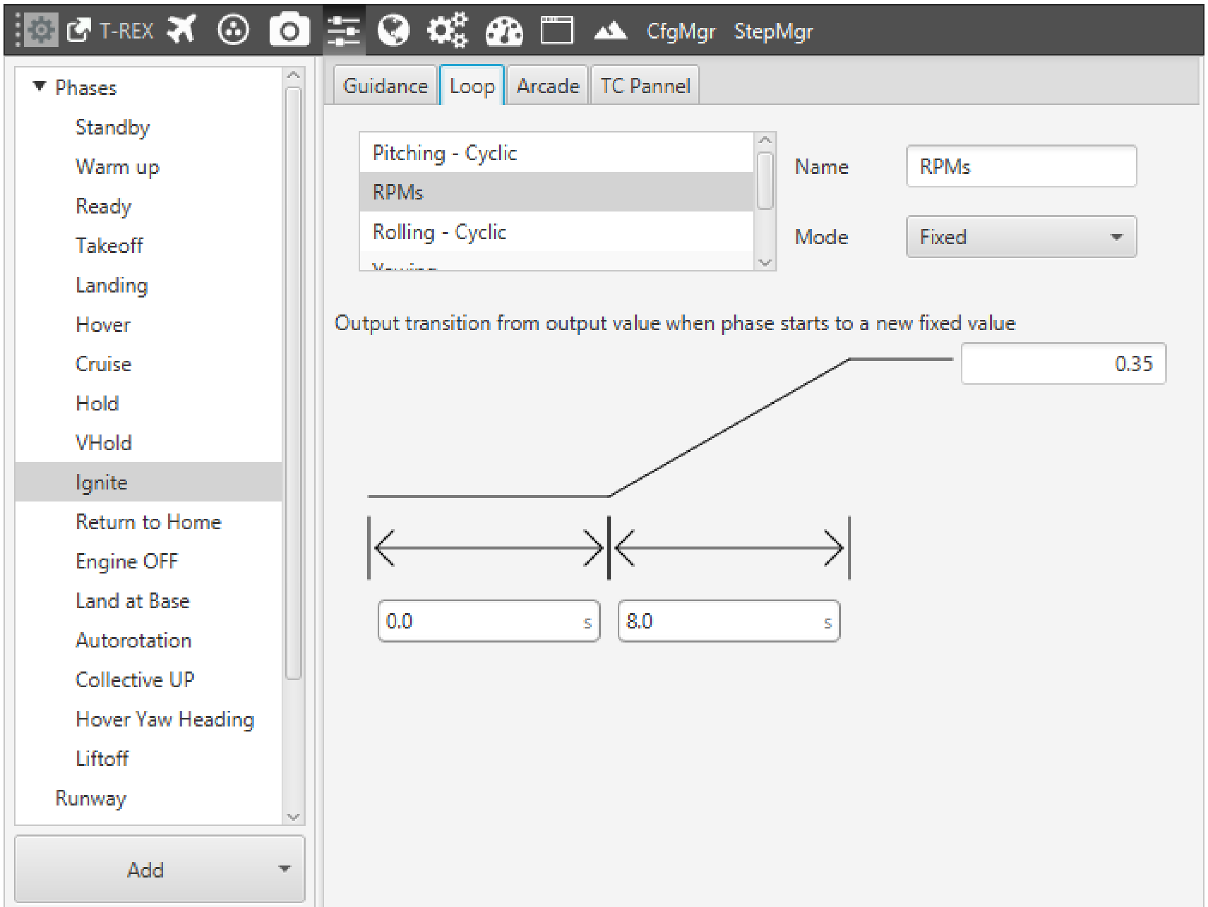
The Guidance of this phase is a **Hold**, which allows to mantain:

- **Desired Roll and Pitch angles:** at 0 [rad].

During this phase RPM of the engine will increase until 35 % of thrust.



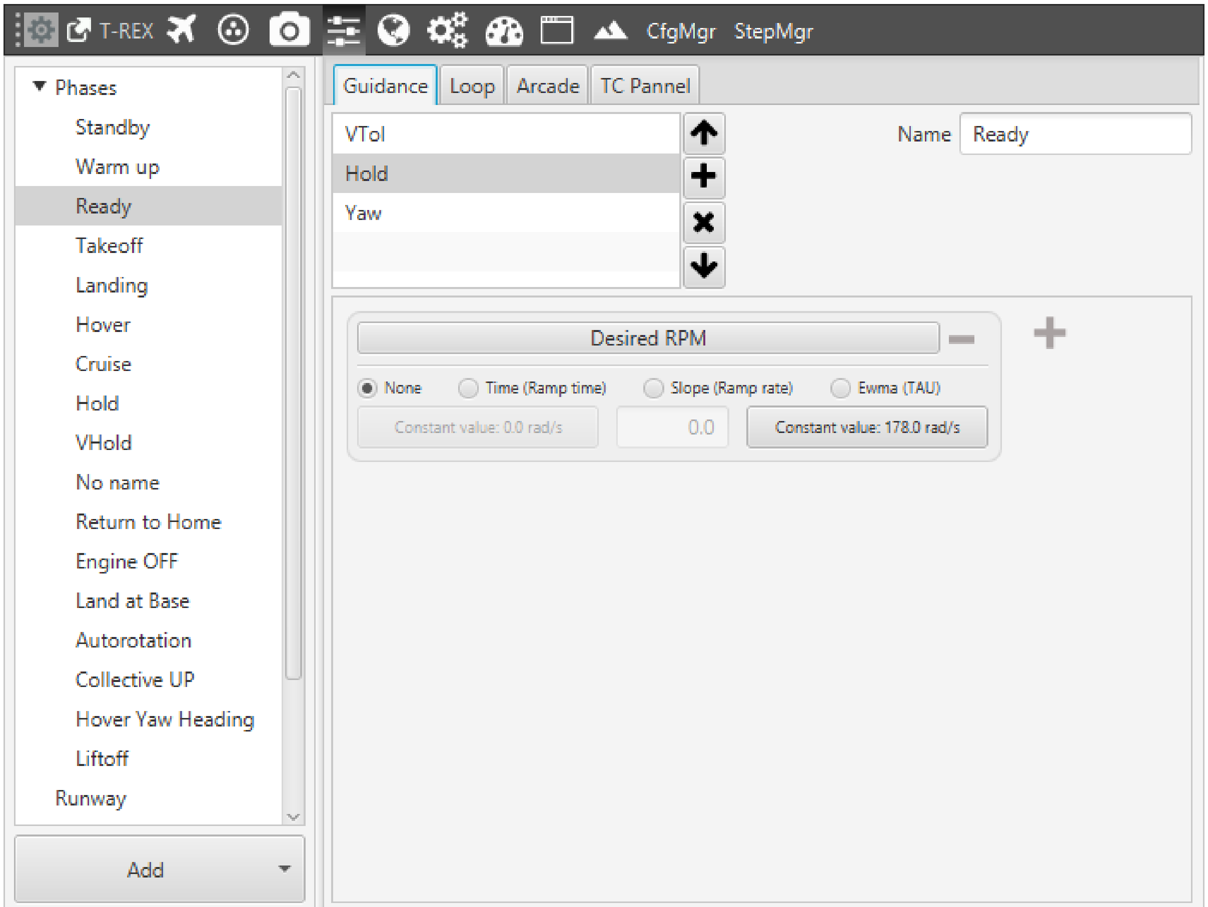
Hold Guidance configuration



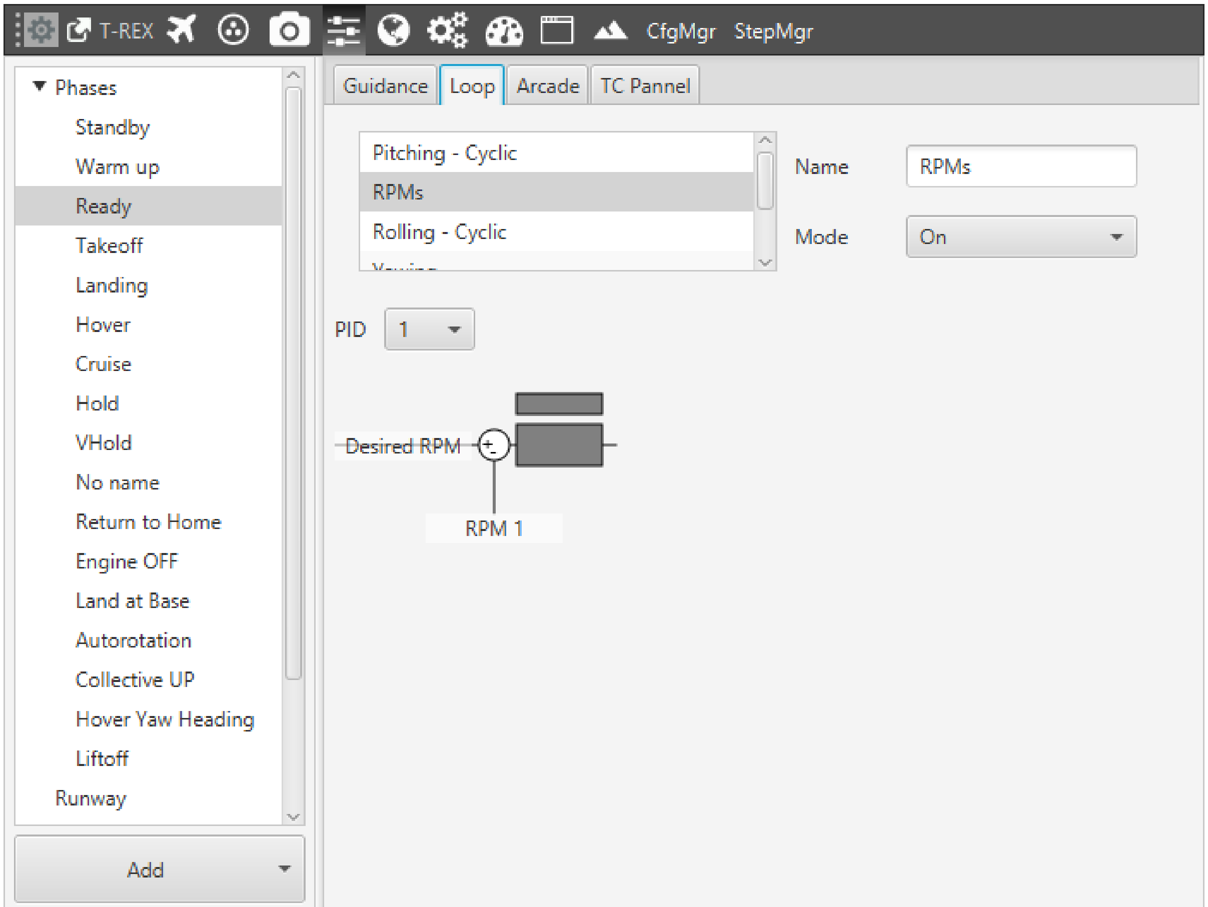
Engine RPM increasing settings

11.2.2.4.2.4 Ready

This phase is the one that starts to control RPM value with PID controller to maintain it at the desired value (Desired RPM). The Guidance are a **Hold** of RPMs (this guidance will be present in all flight phases), and the same **VTol** and **Yaw** of the Standby phase.



Desired RPM value

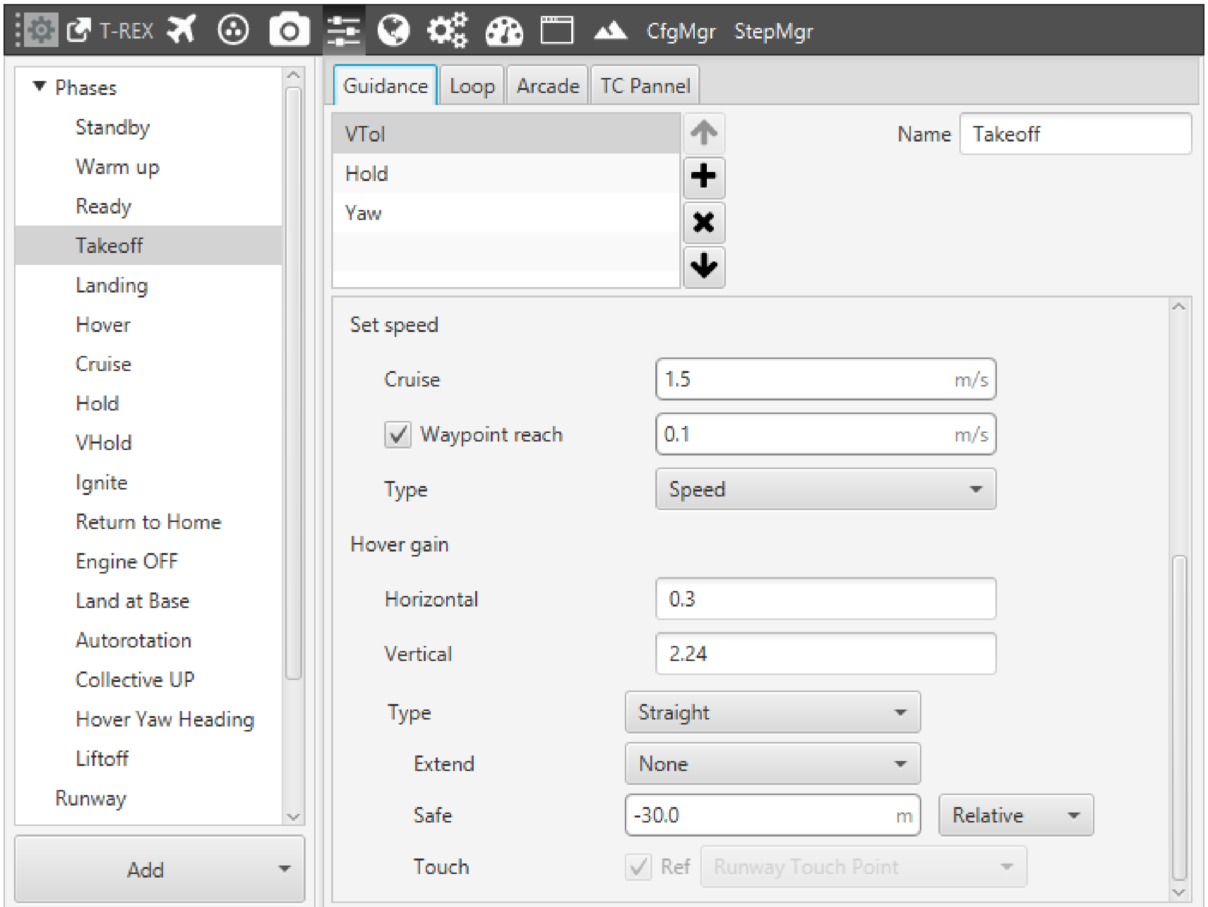


RPM control loop

11.2.2.4.2.5 Takeoff

Take-off phase is composed by 3 Guidances: **VTol**, **Hold** (RPM) and **Yaw** (Current). The first one is a vertical guidance and following parameters are set:

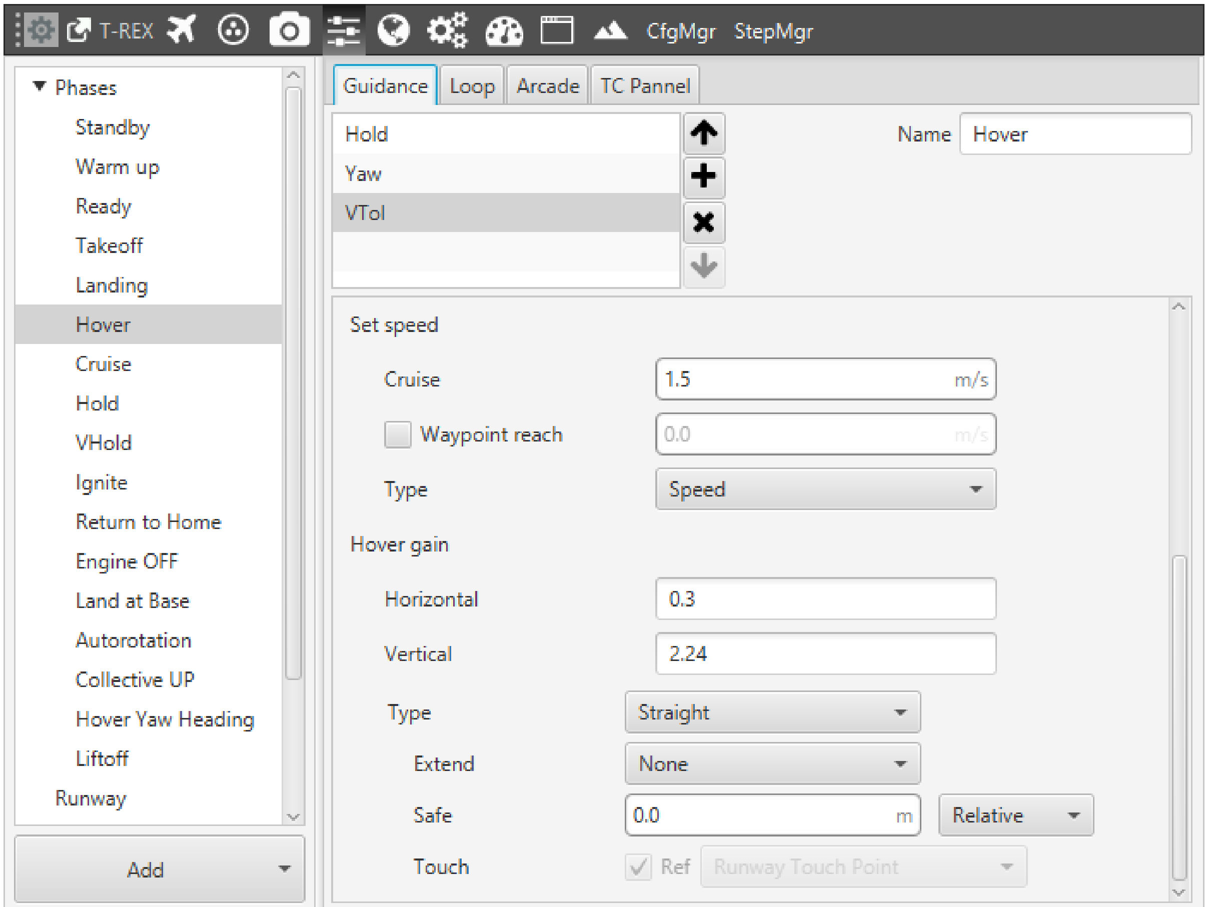
- **Line attraction:** 5.0 [m].
- **Set Speed – Cruise:** 1.5 [m/s].
- **Horizontal Hover Gain:** 0.3.
- **Vertical Hover Gain:** 2.24.
- **Safe Distance:** -30 [m] and Relative (Positive down).



VTol panel

11.2.2.4.2.6 Hover

Hover phase is configured to allow the helicopter maintain the position in the air. The only difference between this phase and the Take Off phase is the **Safe distance** value which is set at 0 [m] in this case.

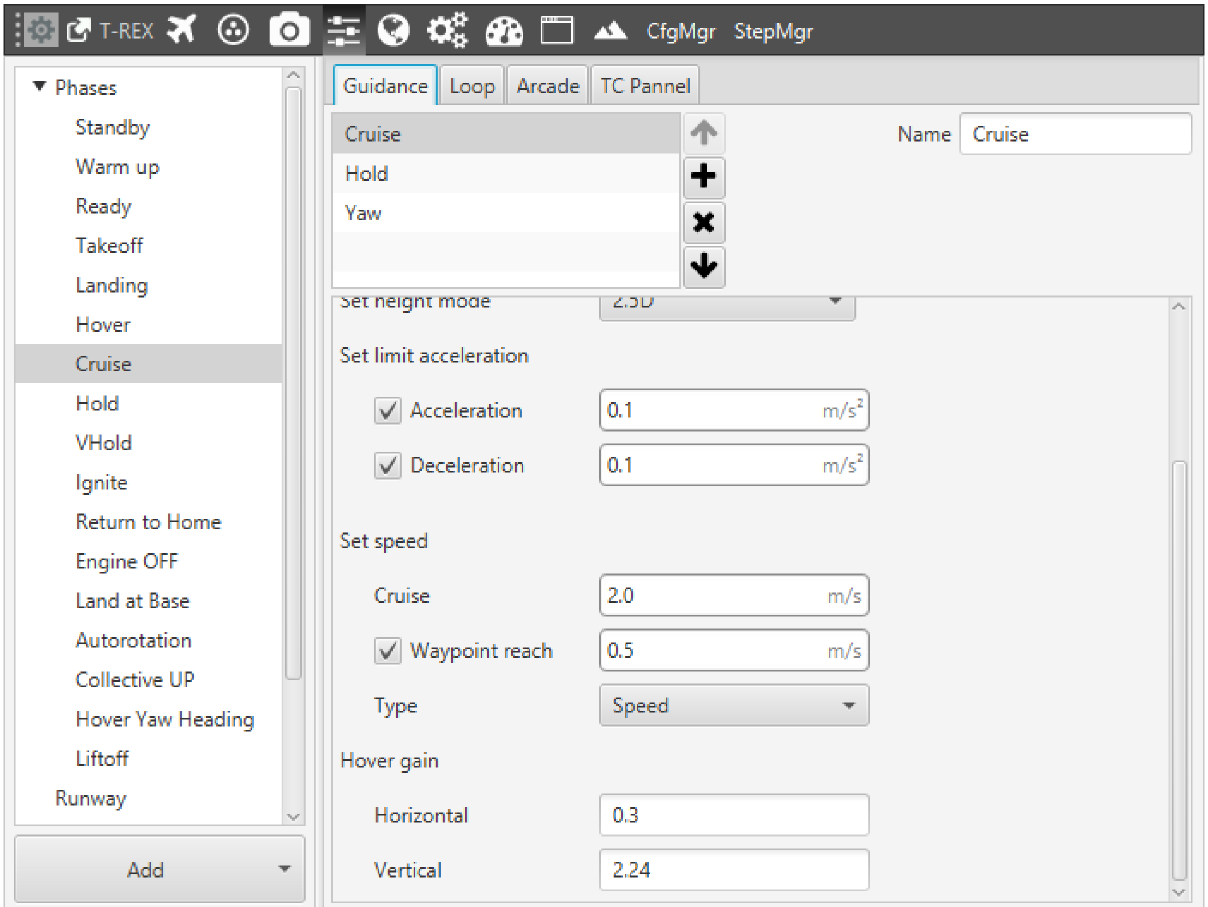


Hover phase panel

11.2.2.4.2.7 Cruise

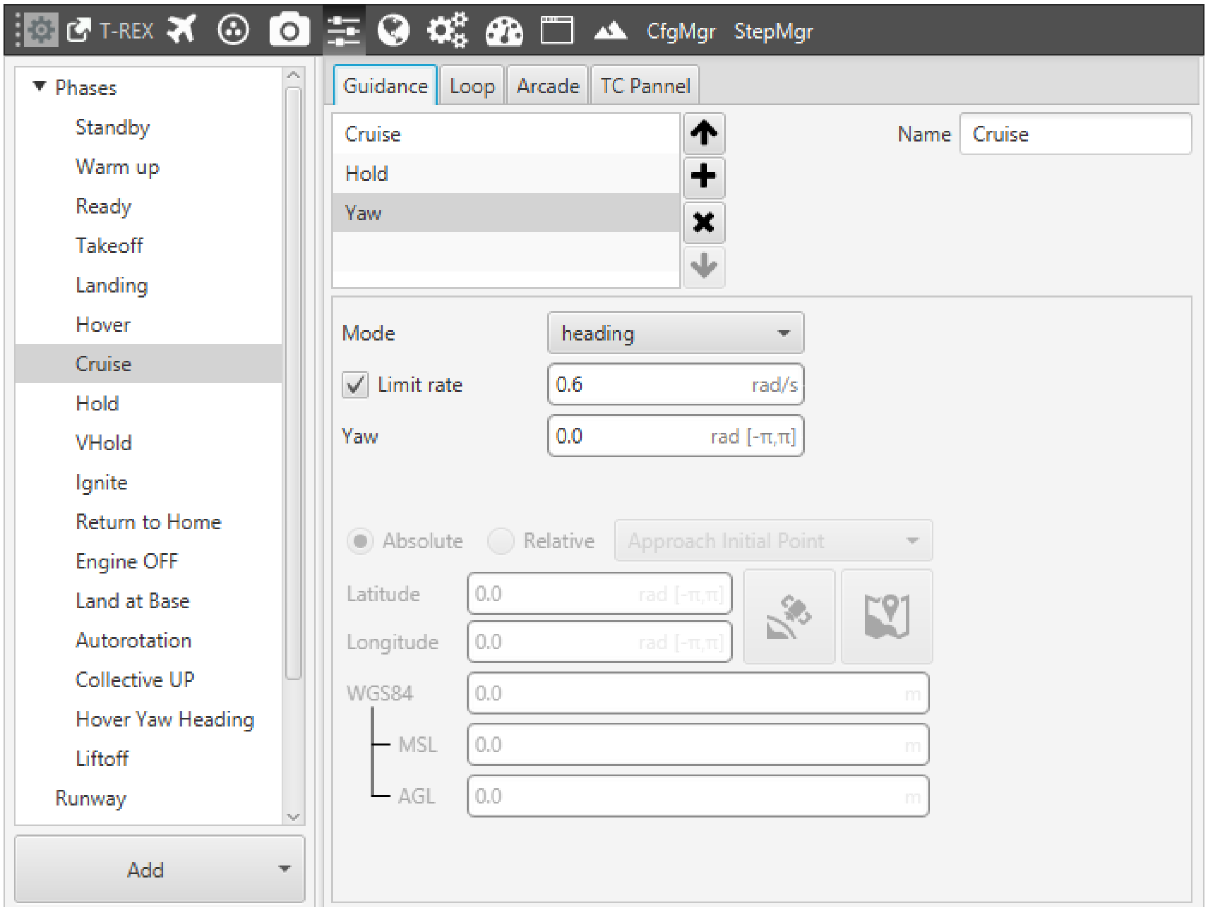
In this phase, the Guidance is Cruise. The parameters to set are the Cruise Speeds (Cruise and Waypoint reach) and Acceleration limits.

- **Acceleration limit:** 0.1.
- **Deceleration limit:** 0.1.
- **Set Speed – Cruise:** 2.0 [m/s].
- **Waypoint reach:** 0.5 [m/s].



Cruise panel

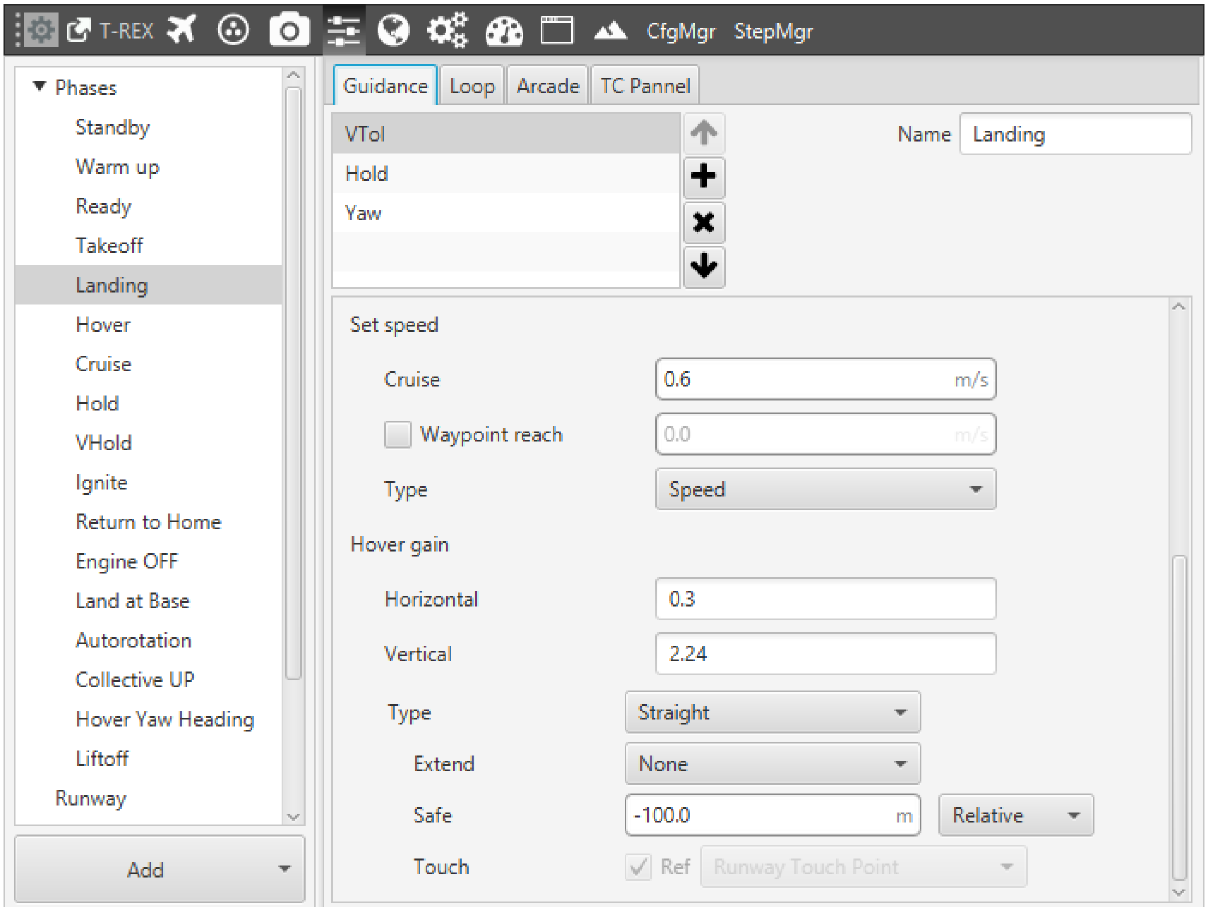
Yaw Guidance is set in order to make the helicopter controlling yaw angle following the heading angle.



Yaw Guidance panel

11.2.2.4.2.8 Landing

This phase has the same settings of the Take-off with a positive value of Safe distance (positive down).



Landing phase panel

In this phase are set the same parameters of the Take Off phase. The only 2 changes are:

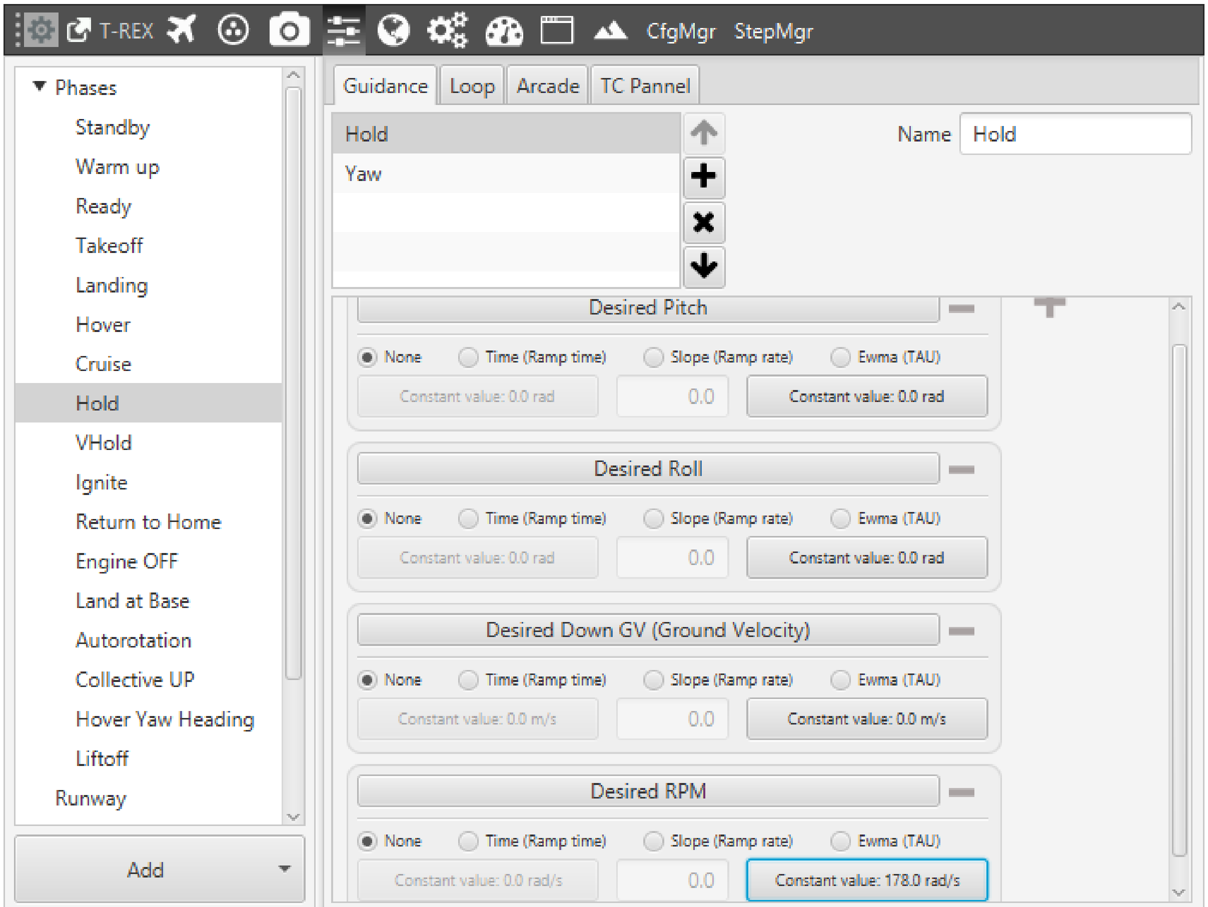
- **Set Speed – Cruise:** 0.6 [m/s].
- **Safe Distance:** -100.0 [m] and Relative (Positive down). In this case, we are supposing that at the instant when the Landing phase starts, the helicopter is flying at AGL < 100 [m].

Yaw Guidance is configured in order to let the helicopter with the Current yaw angle and limiting the **yaw rate** at 0.6 [rad/s].

11.2.2.4.2.9 Hold

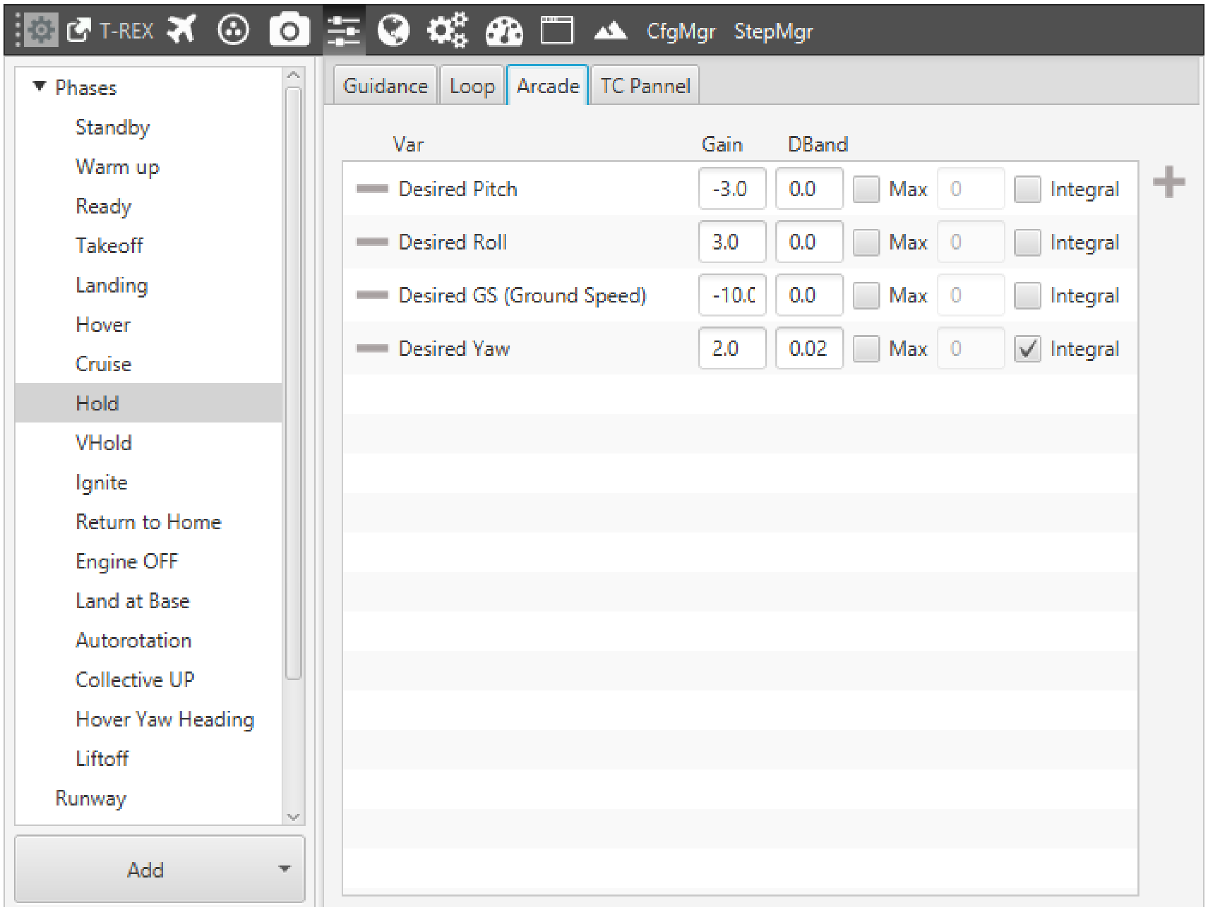
Hold phase is used to perform a Stick controlled flight. The phase guidance is composed of Hold and Yaw guidances. It is a Hold which allows maintaining:

- **Desired pitch:** at 0.0 [rad].
- **Desired roll:** at 0.0 [rad].
- **Desired Down Ground Velocity:** at 0.0 [m/s].
- **Desired RPMs:** at 178.0 [rad/s].



Hold phase panel

In this way, if the stick control is not active, the helicopter will keep its position in the air without the need of phase changing. The **Yaw** Guidance is set on **Current**. In this phase, the Arcade allows controlling the platform depending on the set gains.



Arcade panel

The T-Rex has a helicopter structure. Its control is performed by using the pitch changing of the main rotor blades (cyclic and collective control) and the tail rotor blades depending on the desired attitude change:

- ALTITUDE – Collective control
- PITCH ANGLE – Cyclic control
- ROLL ANGLE – Cyclic control
- YAW ANGLE – Tail blades pitch angle control

T-Rex 600

11.2.2.5 Hybrid

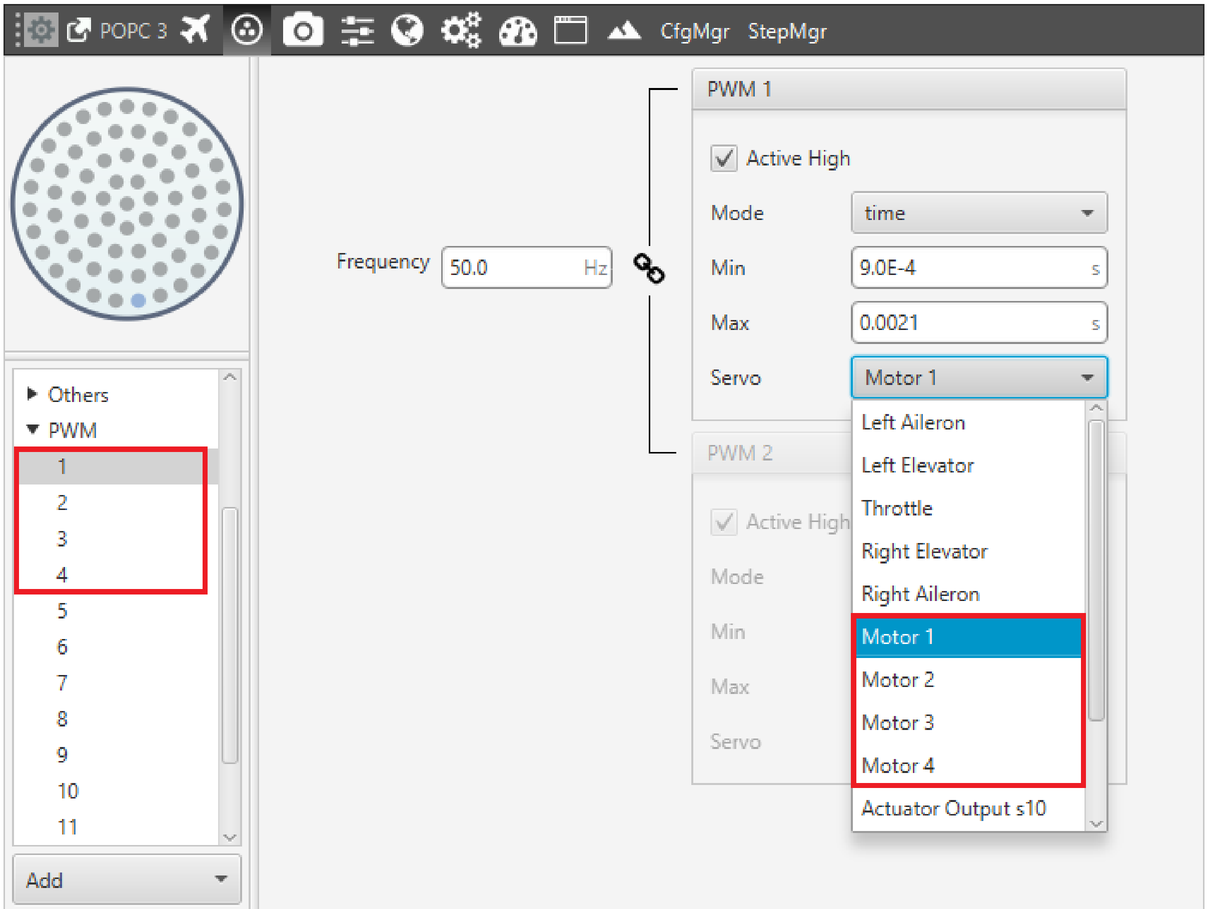
11.2.2.5.1 Servo Configuration

Once the Autopilot has been installed and connected according to the guidelines that appear in section 2.1 of this manual, the first step of the configuration process is the adjustment and trimming of the servos. This section contains the instructions to follow in order to trim the platform servos using Veronte Pipe.

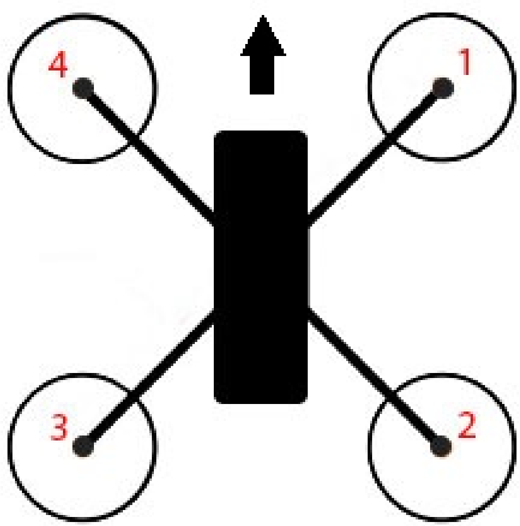
11.2.2.5.1.1 Quadcopter servos

11.2.2.5.1.2 Servos Output

In this case, the controls of the airplane are the four electric motors. Motors can be controlled by changing their thrust (from 0 to 1), so each one of them corresponds to a pin of the connector and they must be positioned in the same order in an S vector who represents the **Actuator Output**. It is possible to connect any pin to any command but the easiest way to perform it and avoid confusion is by following the pins number.



Quadcopter motors connections



Motors numbers

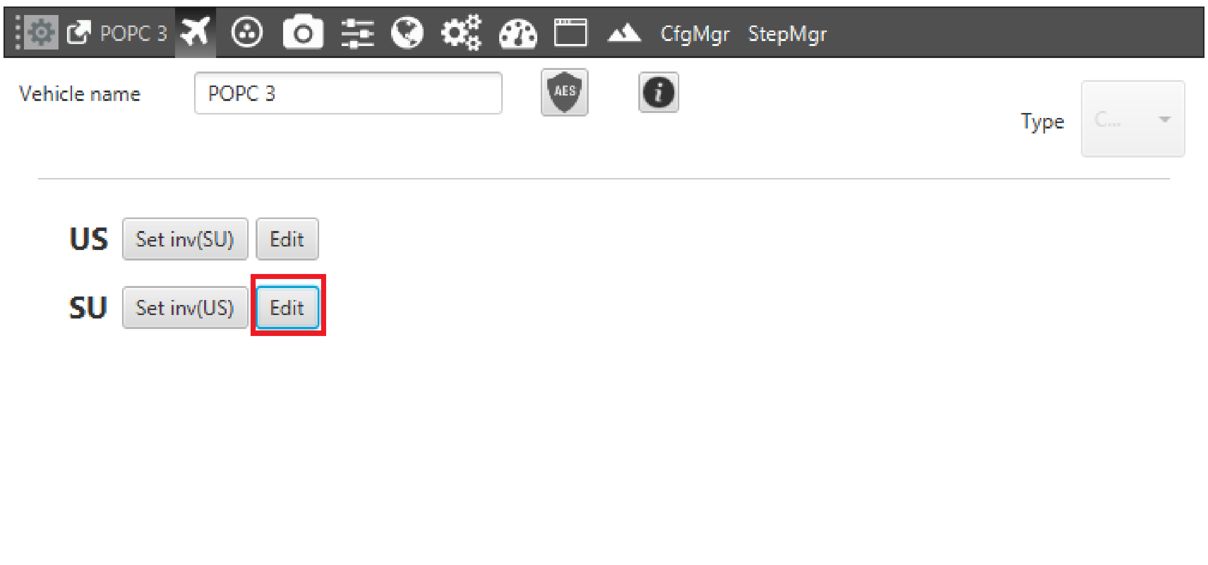
In this case, it is necessary to use 4 pins:

- **Motor 1:** Actuator Output s7.

- **Motor 2:** Actuator Output s8.
- **Motor 3:** Actuator Output s9.
- **Motor 4:** Actuator Output s10.

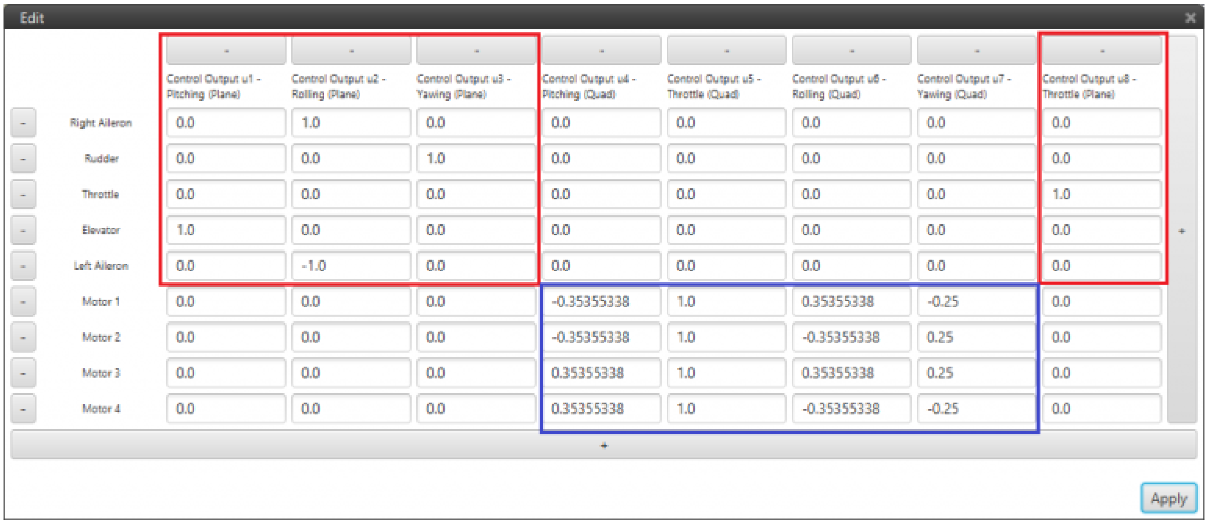
11.2.2.5.1.3 SU Matrix

At this point, the S vector is defined and the SU matrix can be edited. By clicking on **Edit** it is possible to configure the relation between the controller outputs (**U** vector) and the servo movements (**S** vector).



SU matrix edit

> **Warning:** This panel shows the reference system of the aircraft too. It must be positioned in the same way of the Autopilot's one. If it results different, it can be edited by clicking on the corresponding axis in order to reverse its direction.



	Control Output u1 - Pitching (Plane)	Control Output u2 - Rolling (Plane)	Control Output u3 - Yawing (Plane)	Control Output u4 - Pitching (Quad)	Control Output u5 - Throttle (Quad)	Control Output u6 - Rolling (Quad)	Control Output u7 - Yawing (Quad)	Control Output u8 - Throttle (Plane)
Right Aileron	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
Rudder	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
Throttle	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
Elevator	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Left Aileron	0.0	-1.0	0.0	0.0	0.0	0.0	0.0	0.0
Motor 1	0.0	0.0	0.0	-0.35355338	1.0	0.35355338	-0.25	0.0
Motor 2	0.0	0.0	0.0	-0.35355338	1.0	-0.35355338	0.25	0.0
Motor 3	0.0	0.0	0.0	0.35355338	1.0	0.35355338	0.25	0.0
Motor 4	0.0	0.0	0.0	0.35355338	1.0	-0.35355338	-0.25	0.0

SU Matrix (blue – airplane, red – quadcopter)

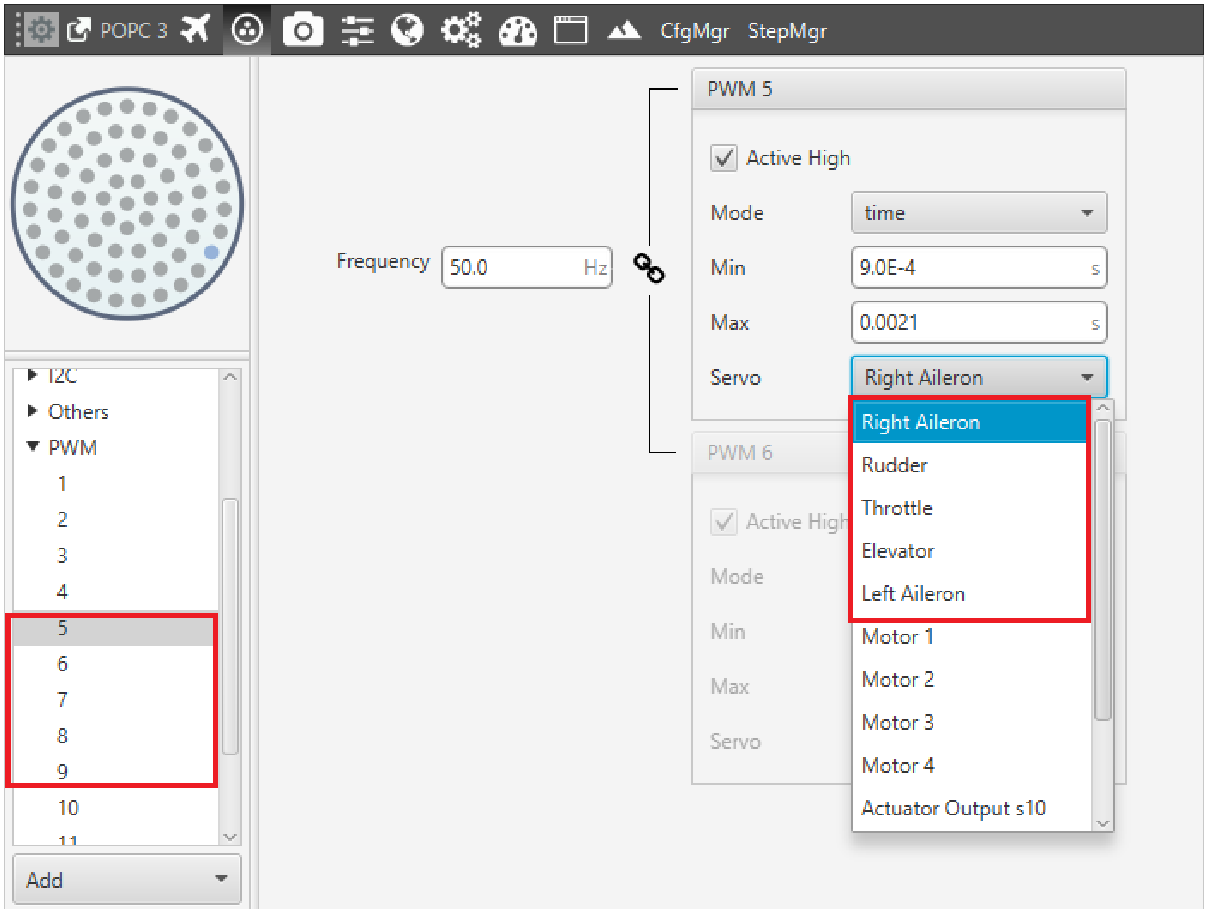
The Hybrid during quadcopter phases is configured as follow:

1. **Pitch Angle Control:** Control Output 5 is configured to perform a positive pitch angle change when motors 1-4 decrease their RPMs and motors 2-3 increase RPM value.
2. **Thrust Control:** the Control Output 6 is the one that allows a thrusting change (0-1).
3. **Roll Angle Control:** Control Output 7 is configured to perform a positive roll angle change when motors 1-2 decrease their RPMs and motors 3-4 increase RPM value.
4. **Yaw Angle Control:** Control Output 8 is configured to perform a positive yaw angle change when motors 2-4 decrease their RPMs and motors 1-3 increase RPM value, in this case with different proportions.

11.2.2.5.1.4 Plane Servos

11.2.2.5.1.5 Servos Output

In this case, it is necessary to configure a conventional aircraft configuration, so there are five controls: elevator, two ailerons, rudder and throttle. Each one of these controls is assigned to a value of the **Actuator Output** vector (s) according to the pin where it is connected and avoiding to use the quadcopter connection pins.



Plane servos connections

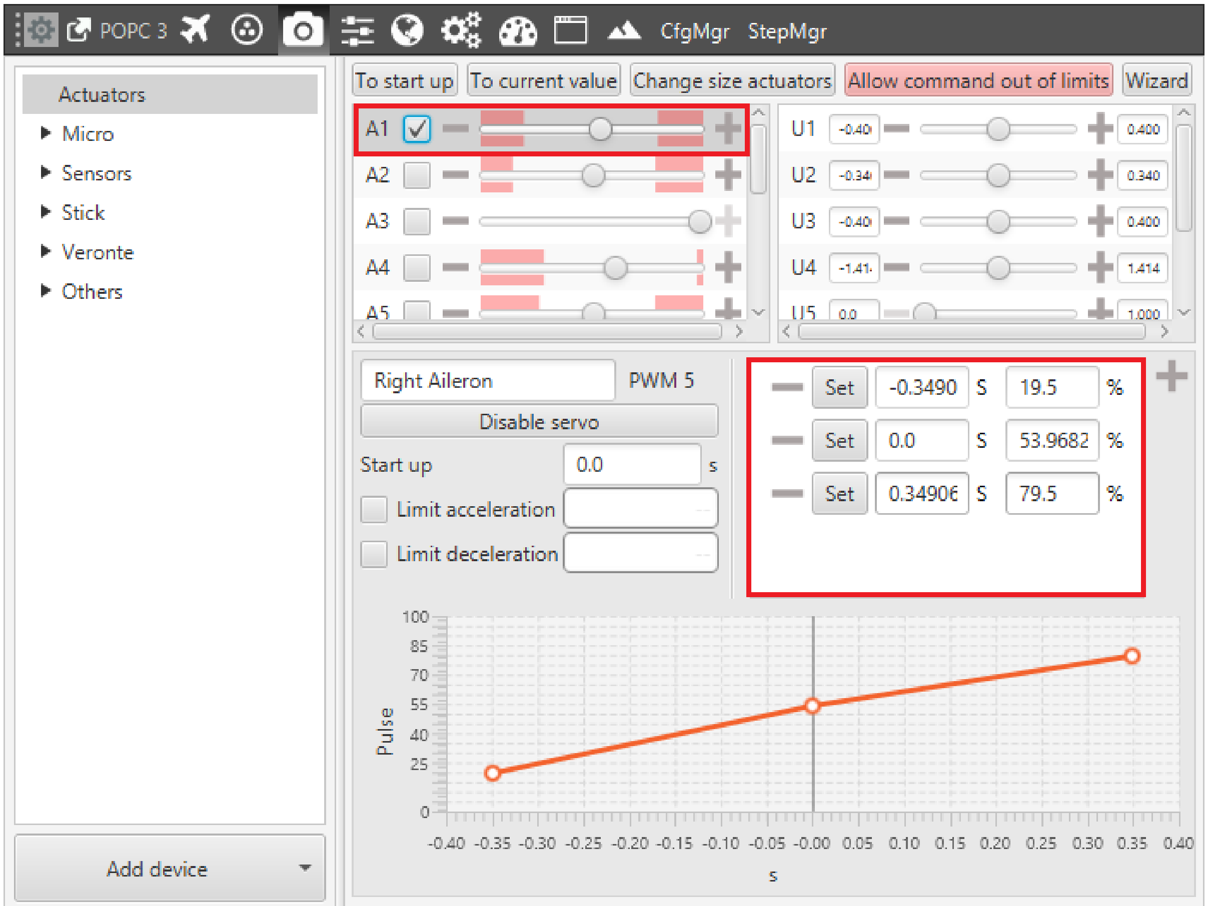
1. **Right Aileron:** Actuator Output s1
2. **Elevator:** Actuator Output s4
3. **Throttle:** Actuator Output s3
4. **Rudder:** Actuator Output s2
5. **Left Aileron:** Actuator Output s5

When the five controls are defined, the SU matrix plane part (blue part) can now be completed. In the case of this Hybrid, each control channel is related with only one servo: the pitching with the elevator, the rolling with the ailerons, the yawing with the rudder and the thrusting with the throttle:

- **Pitching (u1):** Actuator Output s4 (Elevator)
- **Thrusting (u2):** Actuator Output s3 (Throttle)
- **Rolling (u3):** Actuator Output s1 and s5 (Ailerons)
- **Yawing (u4):** Actuator Output s2 (Rudder)

11.2.2.5.1.6 System Trim

As a final step, the system has to be trimmed. This action can be performed by moving servos in three different positions: zero position, minimum and maximum deflection angle (angles are usually limited physically). These positions must be inserted and saved in the software by clicking on “Set” when the actuator is in the desired position.



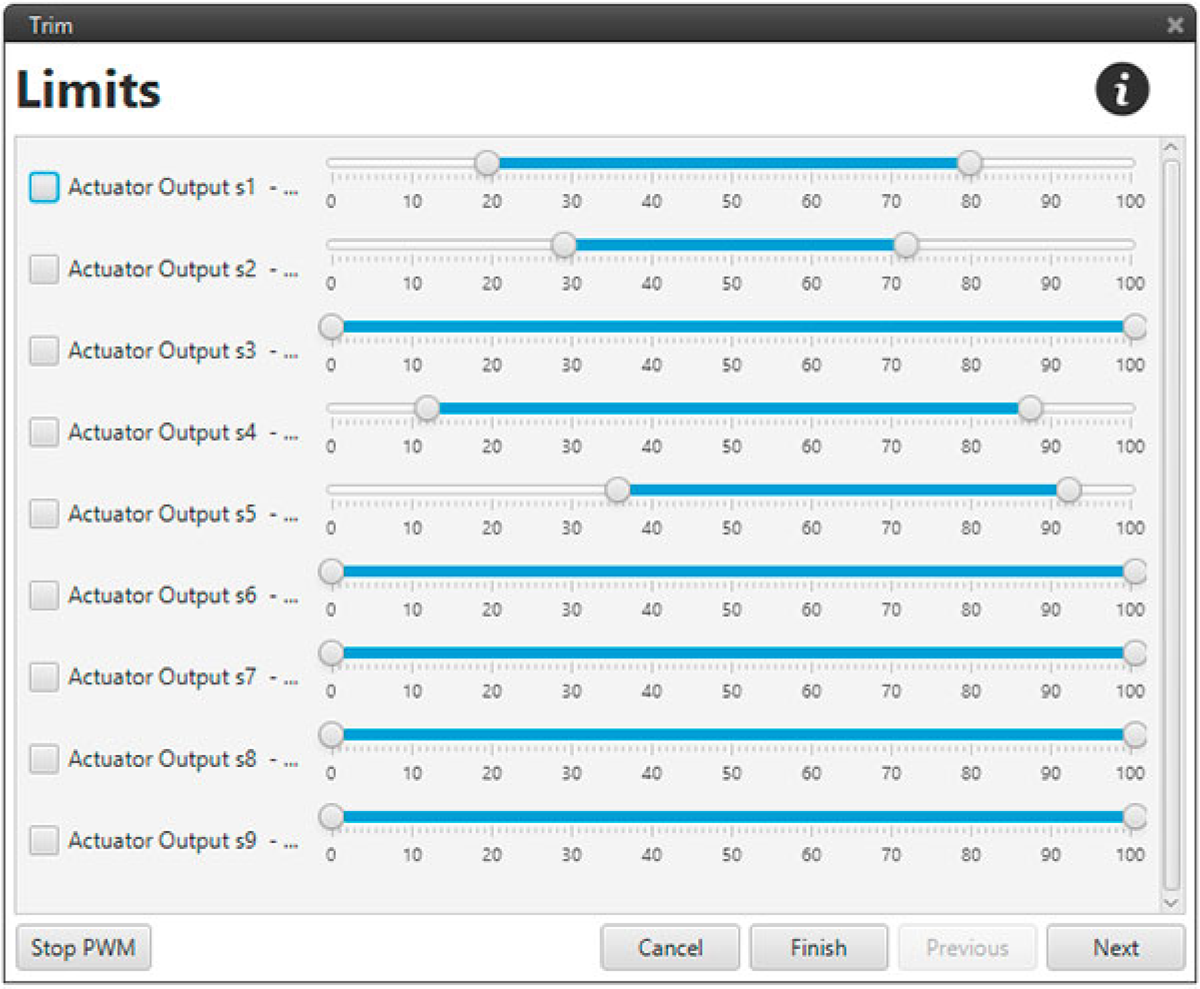
Actuator 1 trimming

The picture shows the setting of the elevator number 1:

- **Minimum:** -0.34906 [rad] deflection ; 19.5%.
- **Zero position:** 0.0[rad] deflection ; 53.9682%.
- **Maximum:** 0.34906 [rad] deflection; 79.5%.

> **Warning:** The actuators can be moved directly from Veronte Pipe only when the system is in an **Initial** phase. During the actuator run, if the desired position is in the “out of range” zone (red zone), it is possible to click on “Allow command out of limits” in order to move completely the actuator and find the correct position.

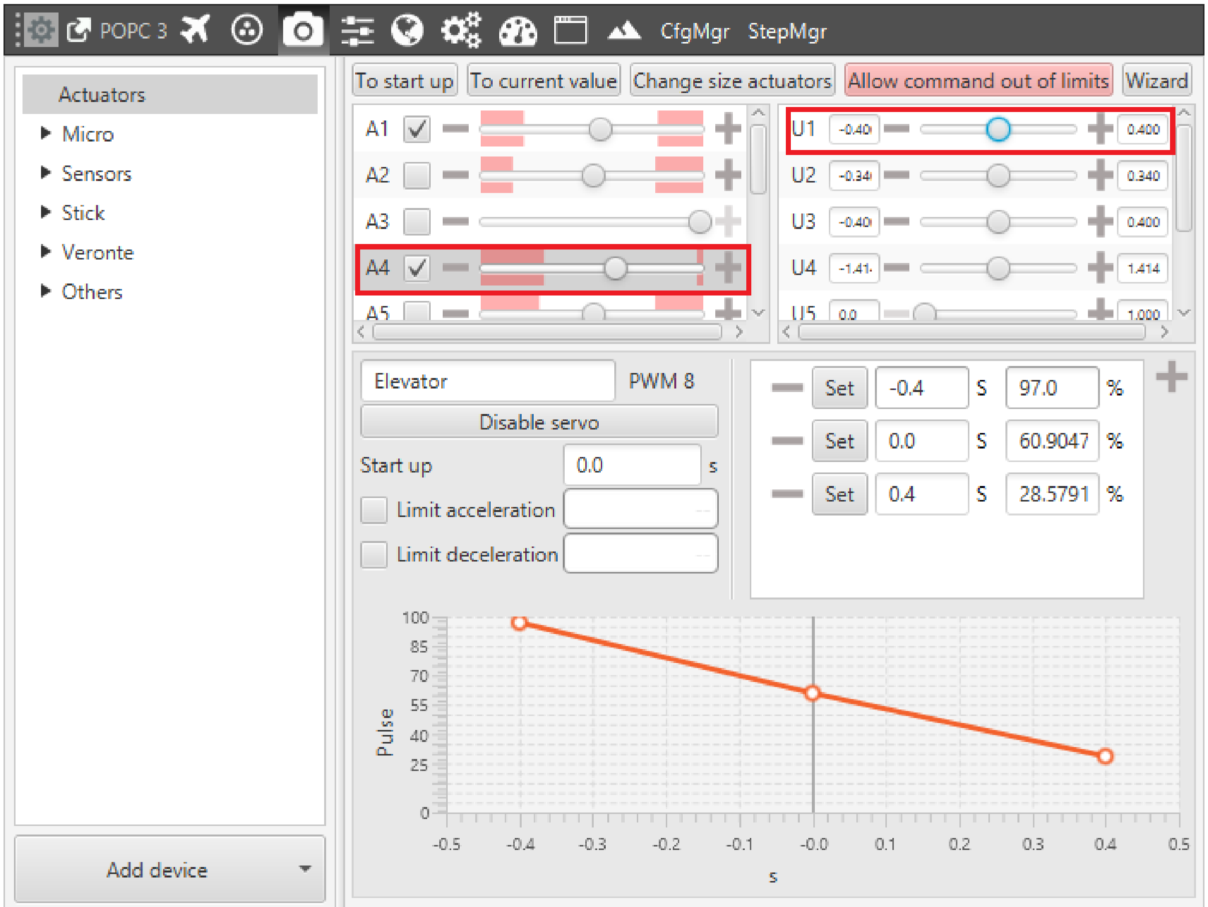
This procedure can be performed in the same way by using a “wizard”. This tool allows moving actuators limits easily and finding the correct range.



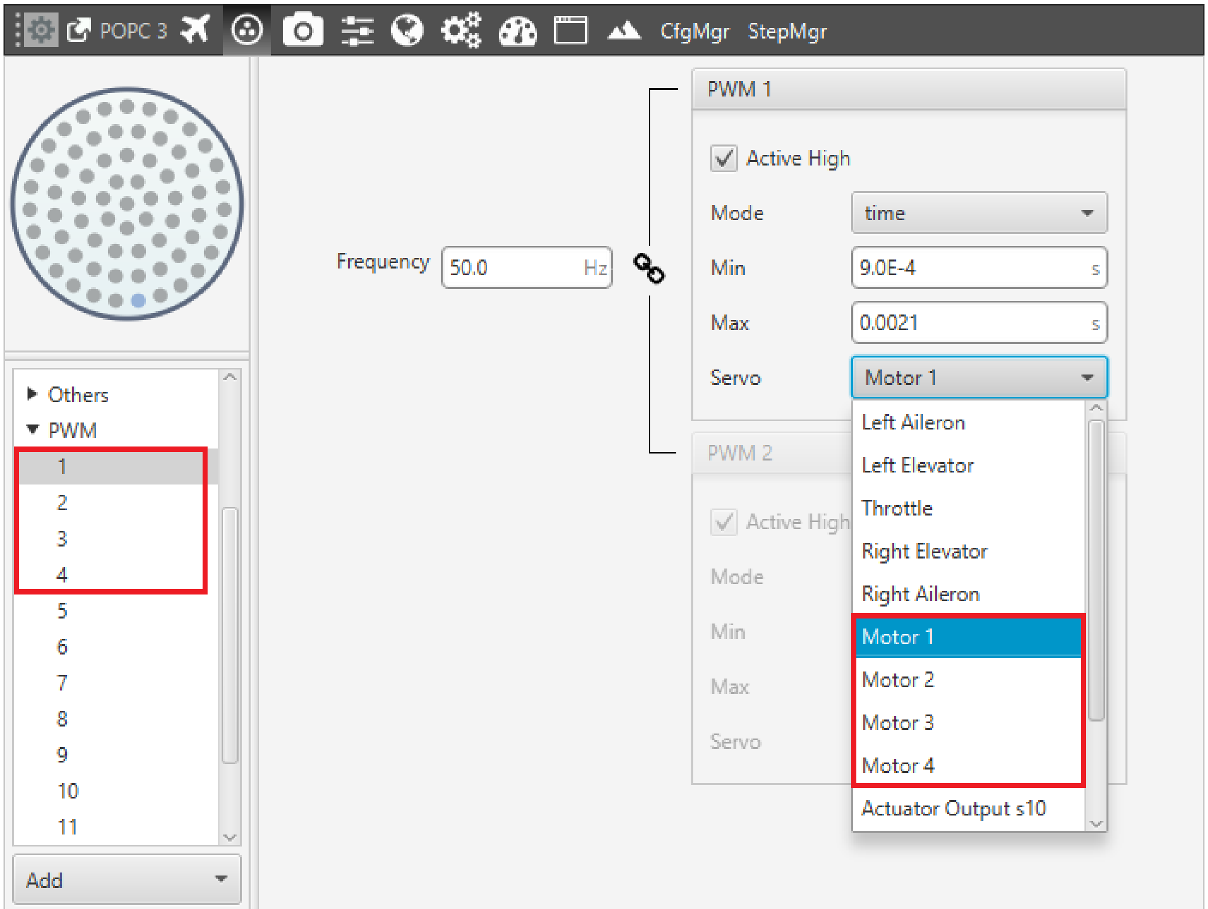
Trim wizard tool

In order to perform a final checking, it is possible to select the desired channel and testing pitch, roll, yaw and thrust controls.

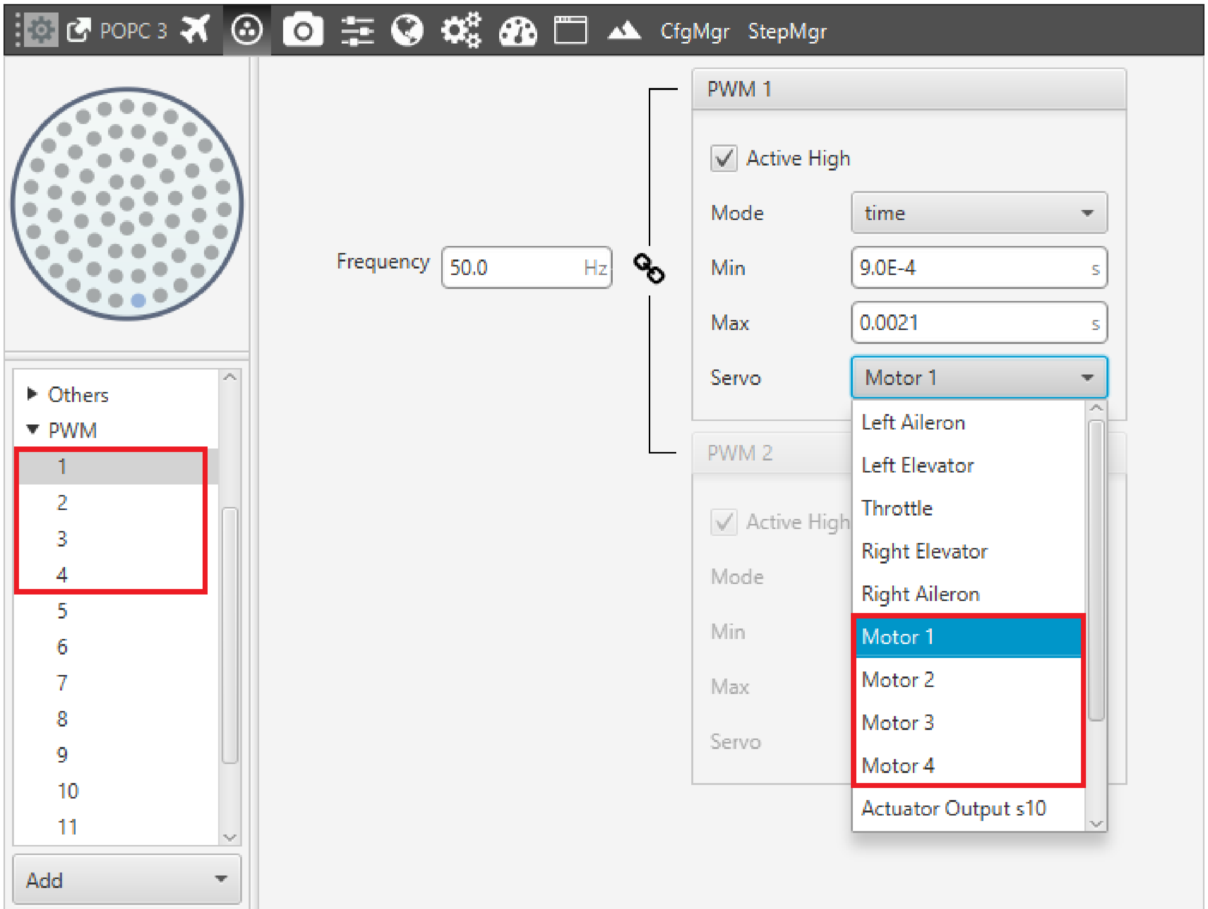
The image below shows a pitching output testing. By moving the U1 control, surfaces must change the position according to the reference system: positive corresponds to nose down and negative to nose up.



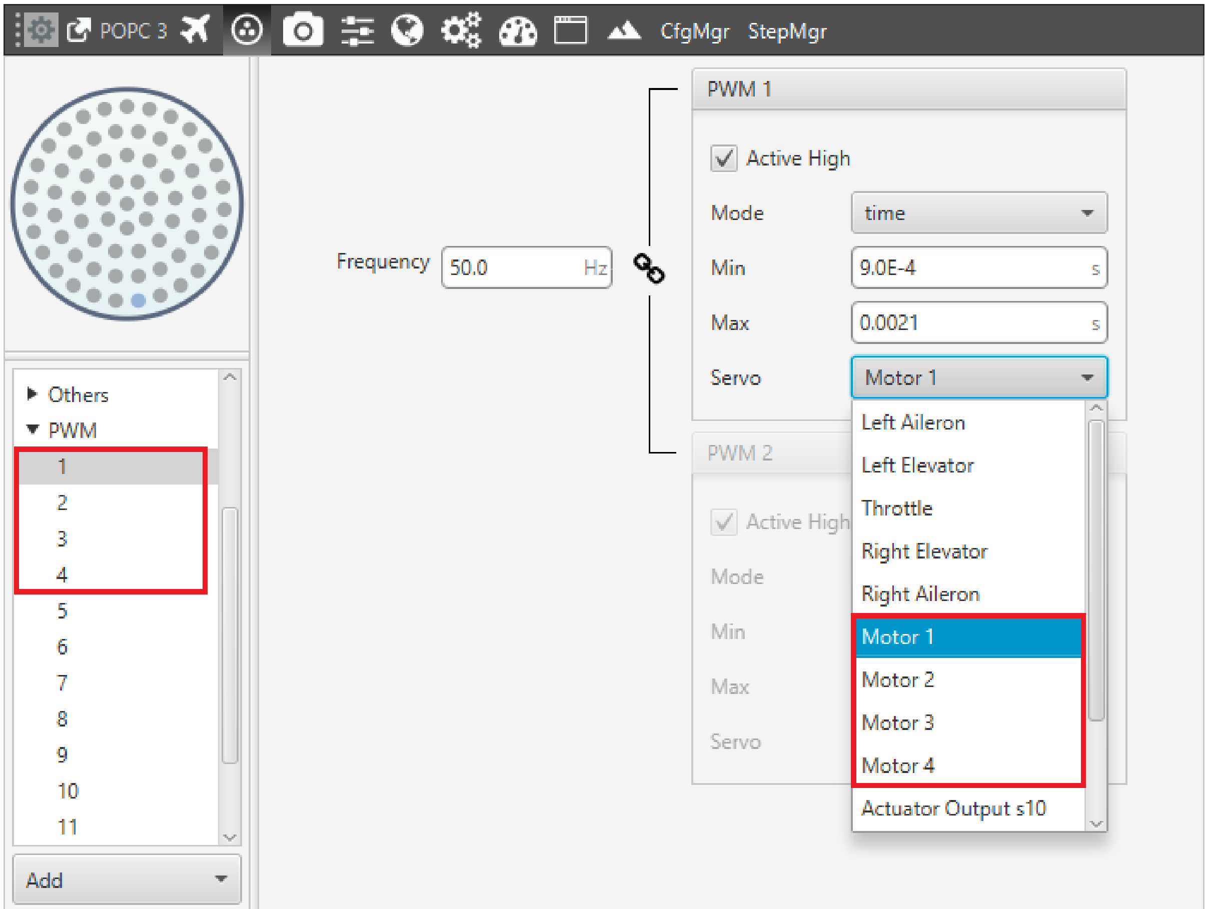
Pitching test



Quadcopter motors connections



Quadcopter motors connections



Quadcopter motors connections

11.2.2.5.2 Mission Phases

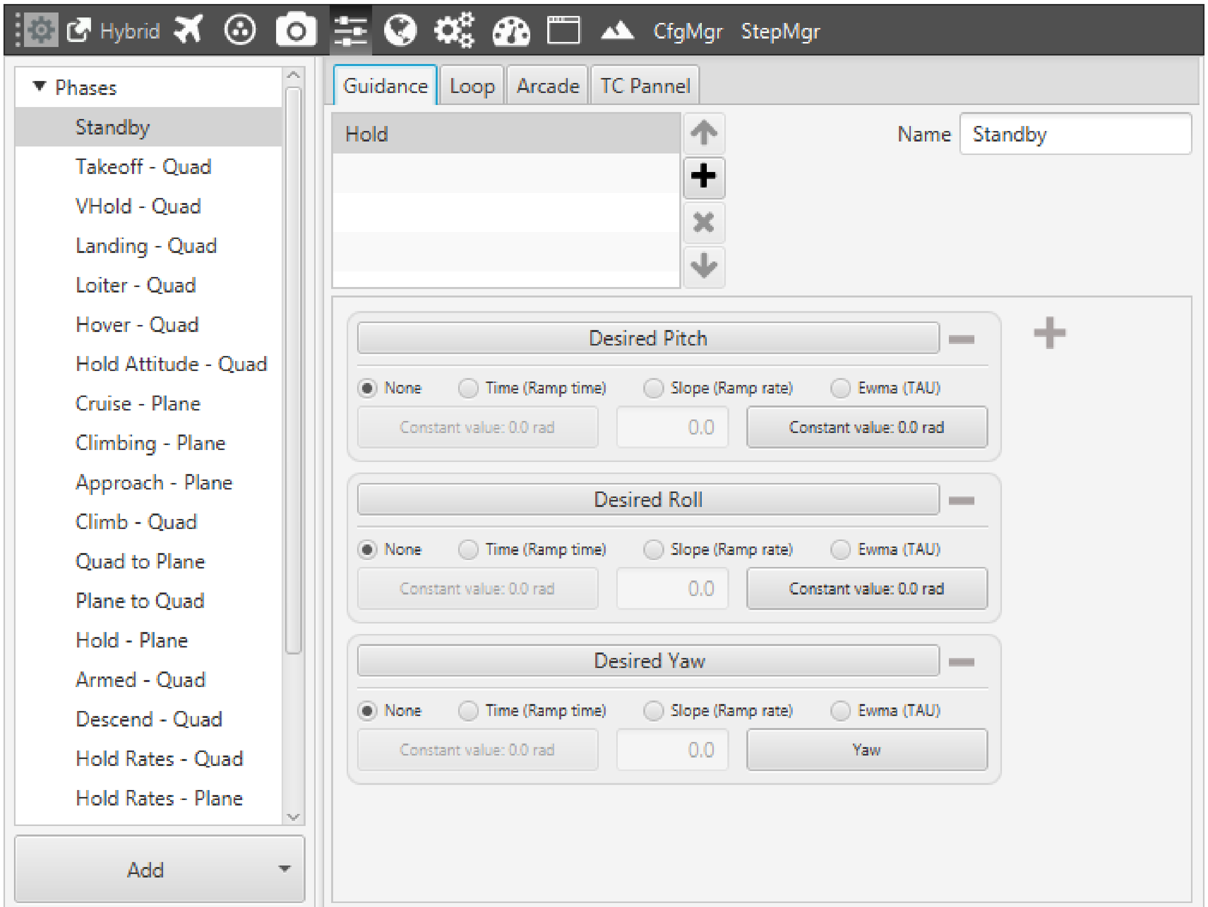
In this section, it will be explained the Hybrid typical mission profile and the mission phases will be detailed. The common phases of Quadcopter and Plane are:

- Standby
- Transition Quad to Plane
- Transition Plane to Quad

For all the other phases, the flight type will be specified.

11.2.2.5.2.1 Standby

Standby phase is a preliminary phase of the operation. Normally, the automation which allows the system to pass to Standby phase requires the GPS signal.



Standby phase panel

The guidance is a **Hold** of three variables:

- **Desired Pitch and Roll angles:** at 0 [rad].
- **Desired Yaw angle:** set at the current.

Hybrid

This hybrid aircraft has an airplane – quadcopter integrated structure. Its control is performed by using typicals surfaces control for an airplane configuration during plane phases:

- PITCH
- ROLL
- YAW
- THRUST

And a differential thrust for each quadcopter motor depending on the desired attitude change during quadcopter phases.

The two transition phases are the most critical, in fact, it is important to keep the aircraft control during plane-quadcopter changing avoiding command overlaps.

This section contains the whole configuration process of four different platforms in Veronte Pipe. All the different menus that appear in the software will be covered so the final user of the system gains a fully understanding of how to totally configure its platform trough some real examples.

The platforms available are:

- Aircraft: **Mentor.**
- Flying Wing: **W210.**
- Multicopter: **M400.**
- Helicopter: **T-Rex.**

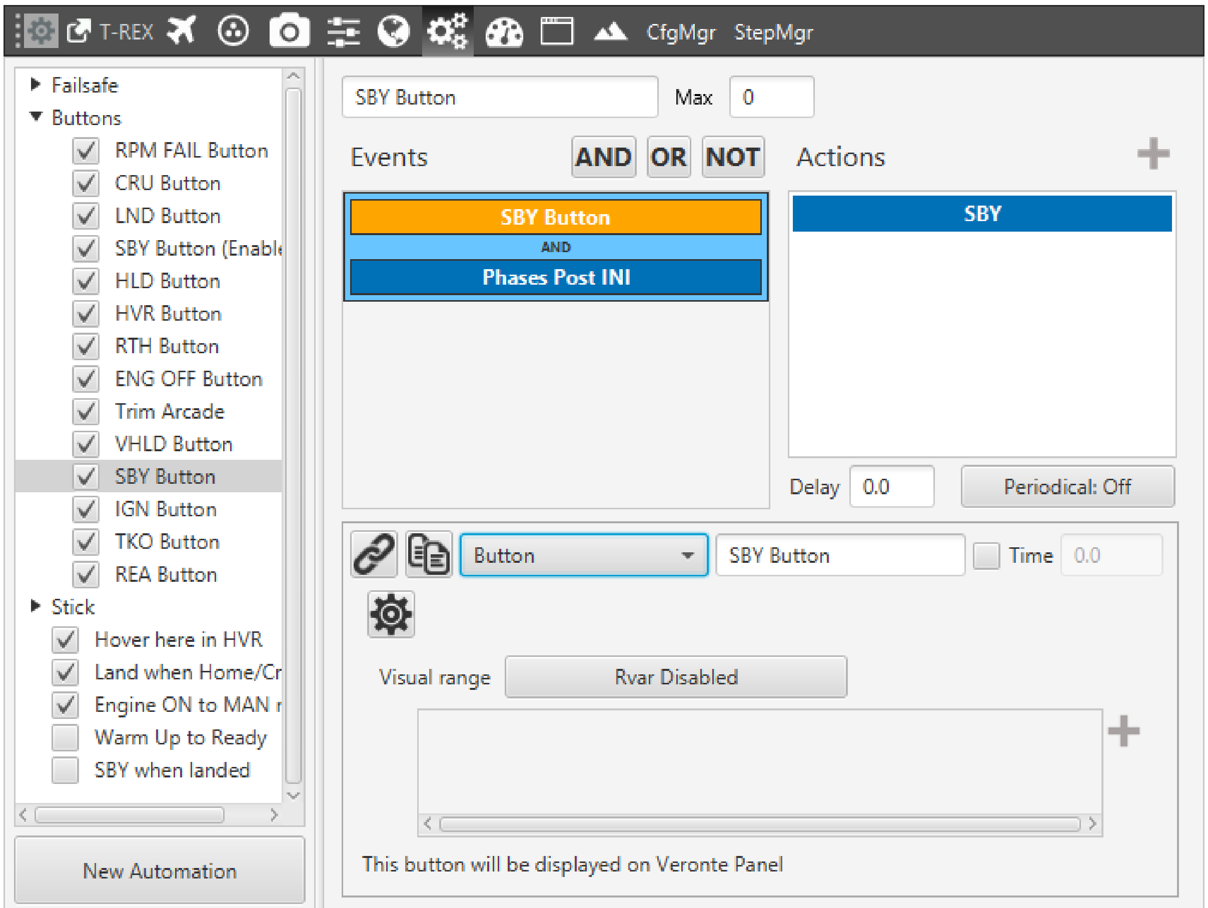
In this section we will explain the configuration of some devices and plataforms to improve the understandig of the user.

11.3 Automations

11.3.1 Veronte Panel Buttons

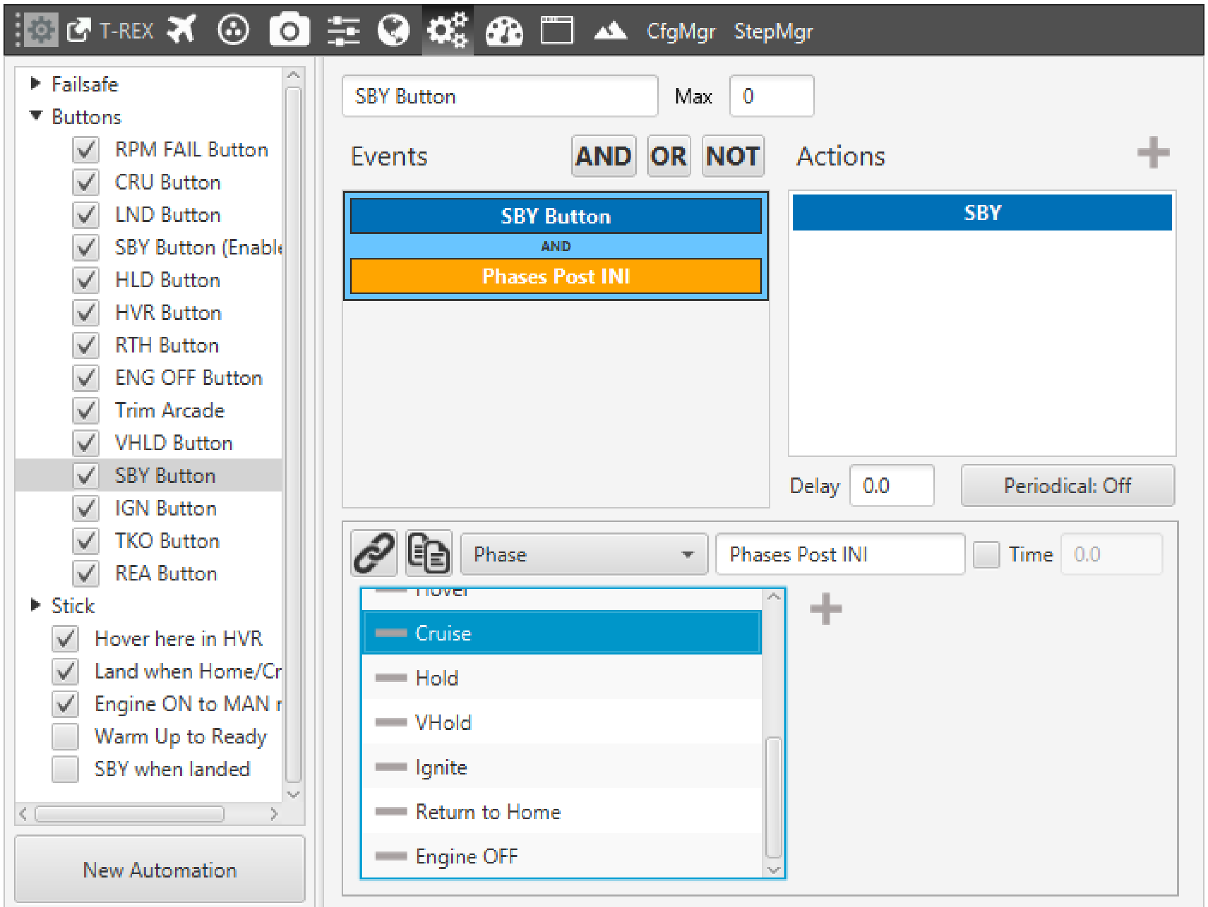
11.3.1.1 Phase Change Buttons

The first automations to be created are the ones linked to the buttons of Veronte Panel. When clicking one of these buttons the phase is changed to the one shown on the button label. The image below shows the Standby Button automation.

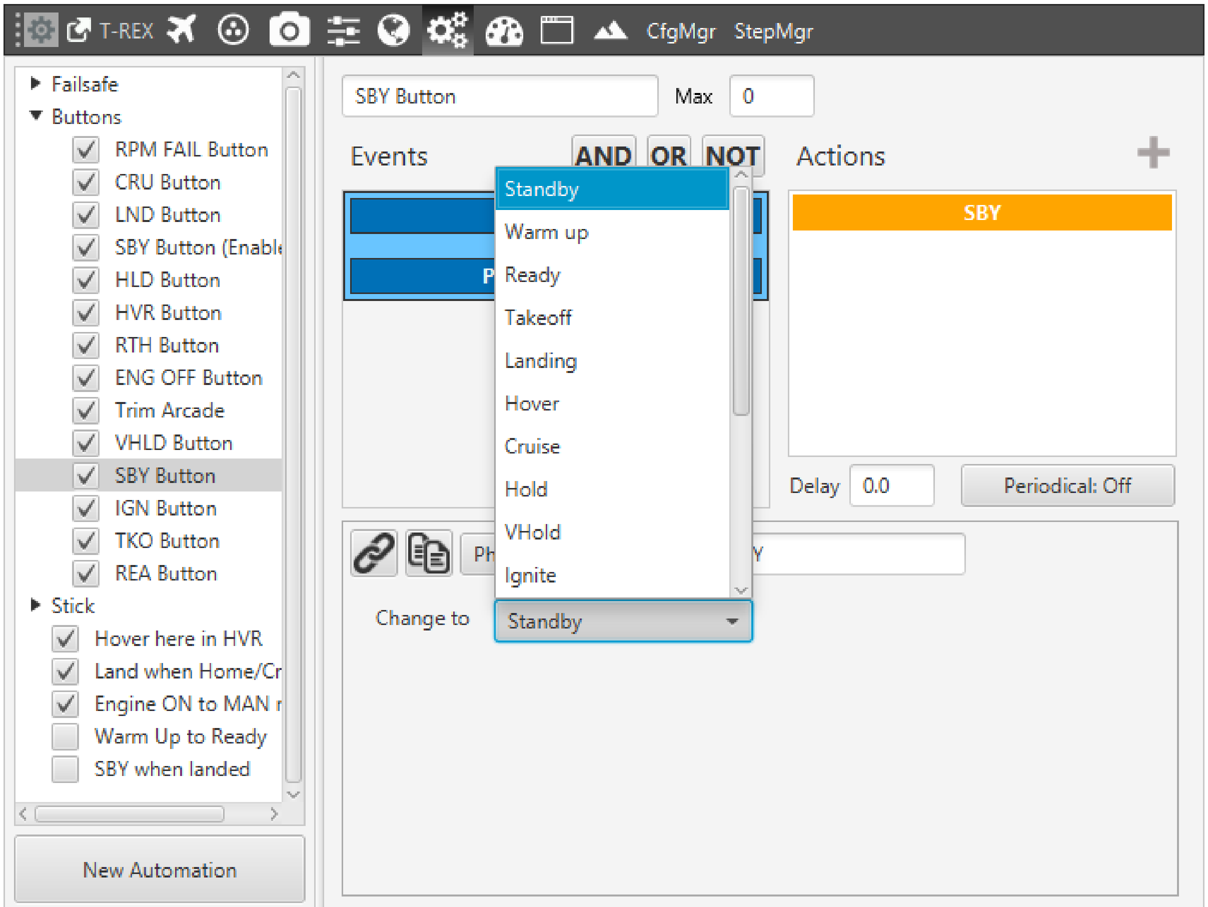


Standby button automation

Normally, besides the button event, a safety condition is added to the automation, which consists on letting only the system to change to a phase when being in a certain set of phases. It is only necessary to create an **AND** condition with the phases to change from. For example, in this case, the set of phases is shown in the following image.


T-REX
CfgMgr
StepMgr
Failsafe
Buttons
RPM FAIL Button
CRU Button
LND Button
SBY Button (Enable
HLD Button
HVR Button
RTH Button
ENG OFF Button
Trim Arcade
VHLD Button
SBY Button
IGN Button
TKO Button
REA Button
Stick
Hover here in HVR
Land when Home/Cr
Engine ON to MAN r
Warm Up to Ready
SBY when landed
New Automation
SBY Button
Max
0
Events
AND
OR
NOT
Actions
SBY Button
AND
Phases Post INI
SBY
Delay
0.0
Periodical: Off
Phase
Phases Post INI
Time
0.0
Cruise
Hold
VHold
Ignite
Return to Home
Engine OFF

Phases set

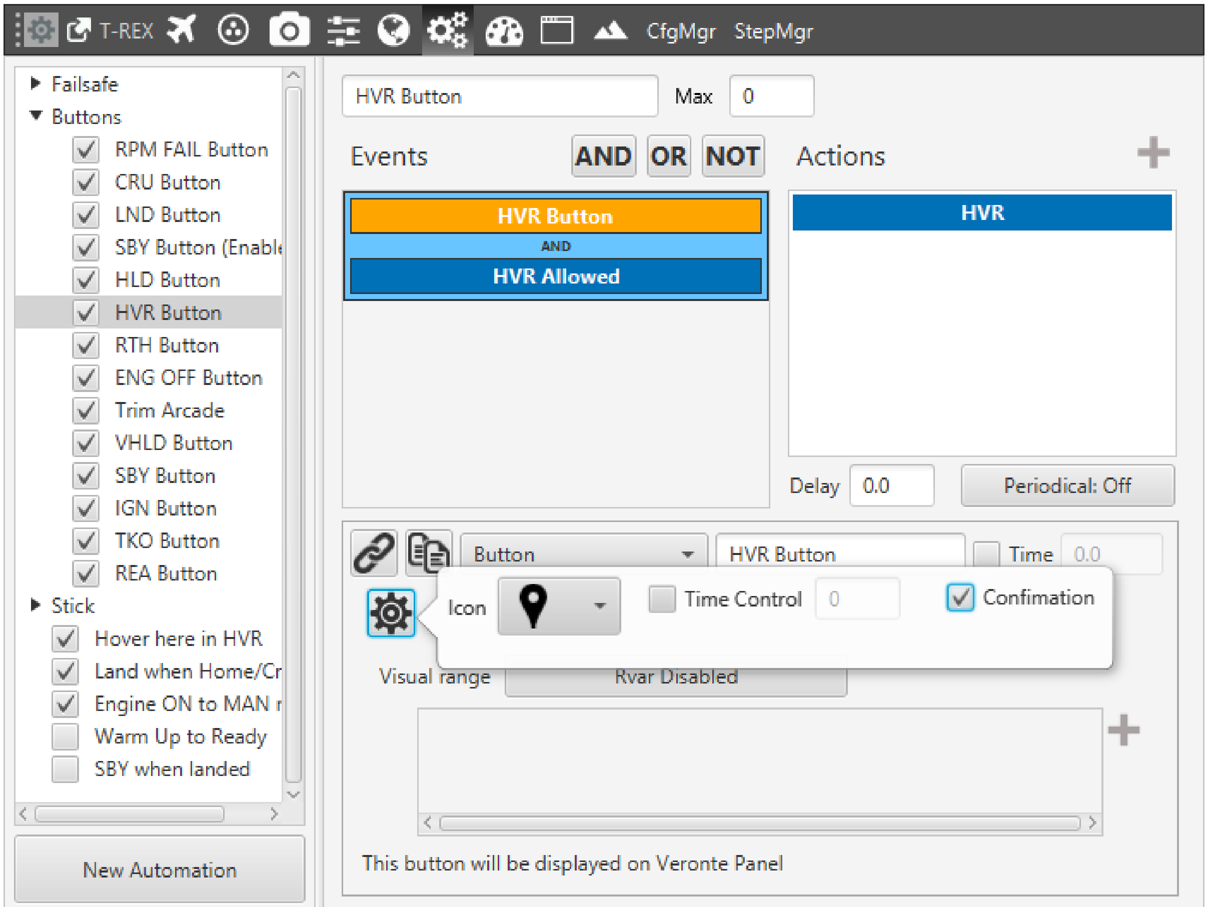


Standby action

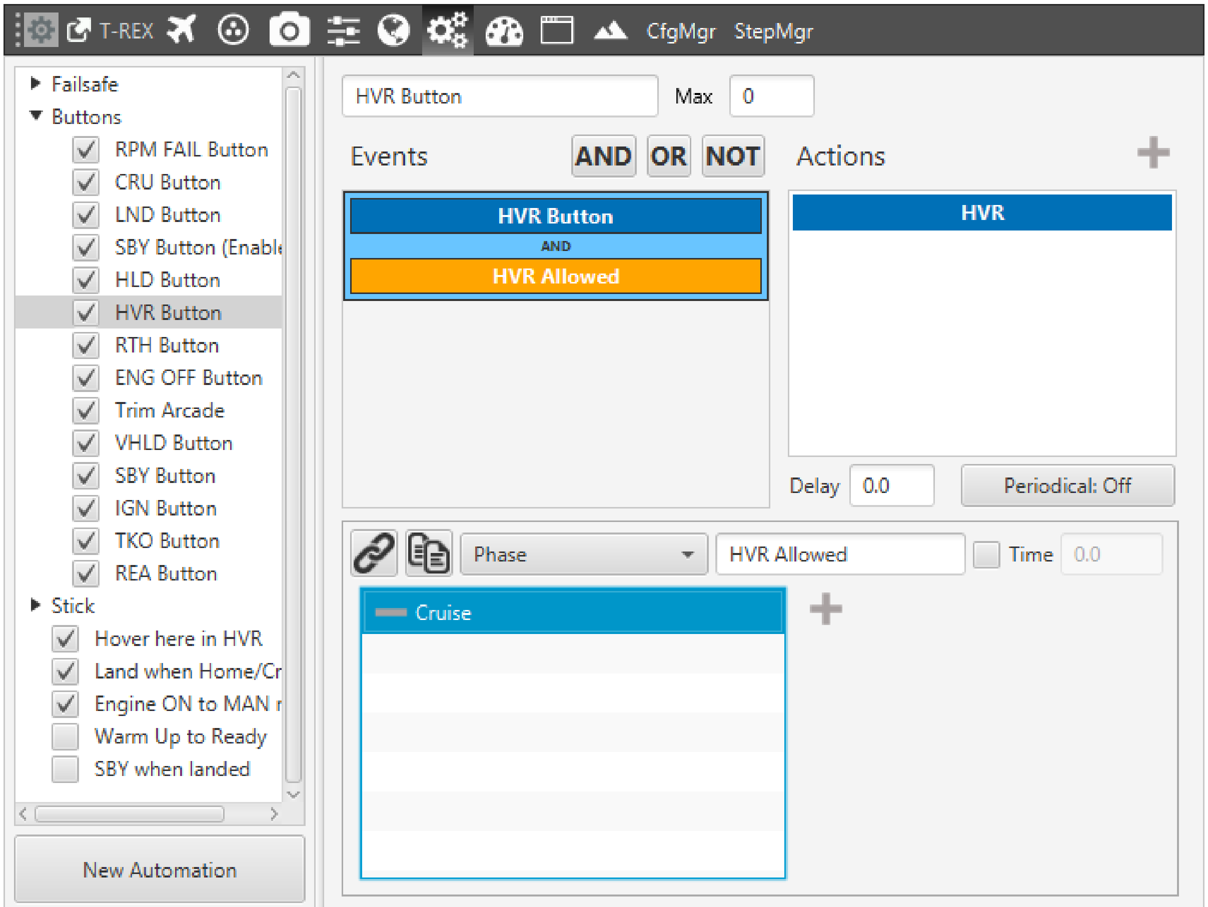
In the previous figure, the automation is the change to StandBy when clicking on Veronte panel, being on any phase selected in the event. This process is normally repeated for the rest of phases.

11.3.1.2 Generic Button

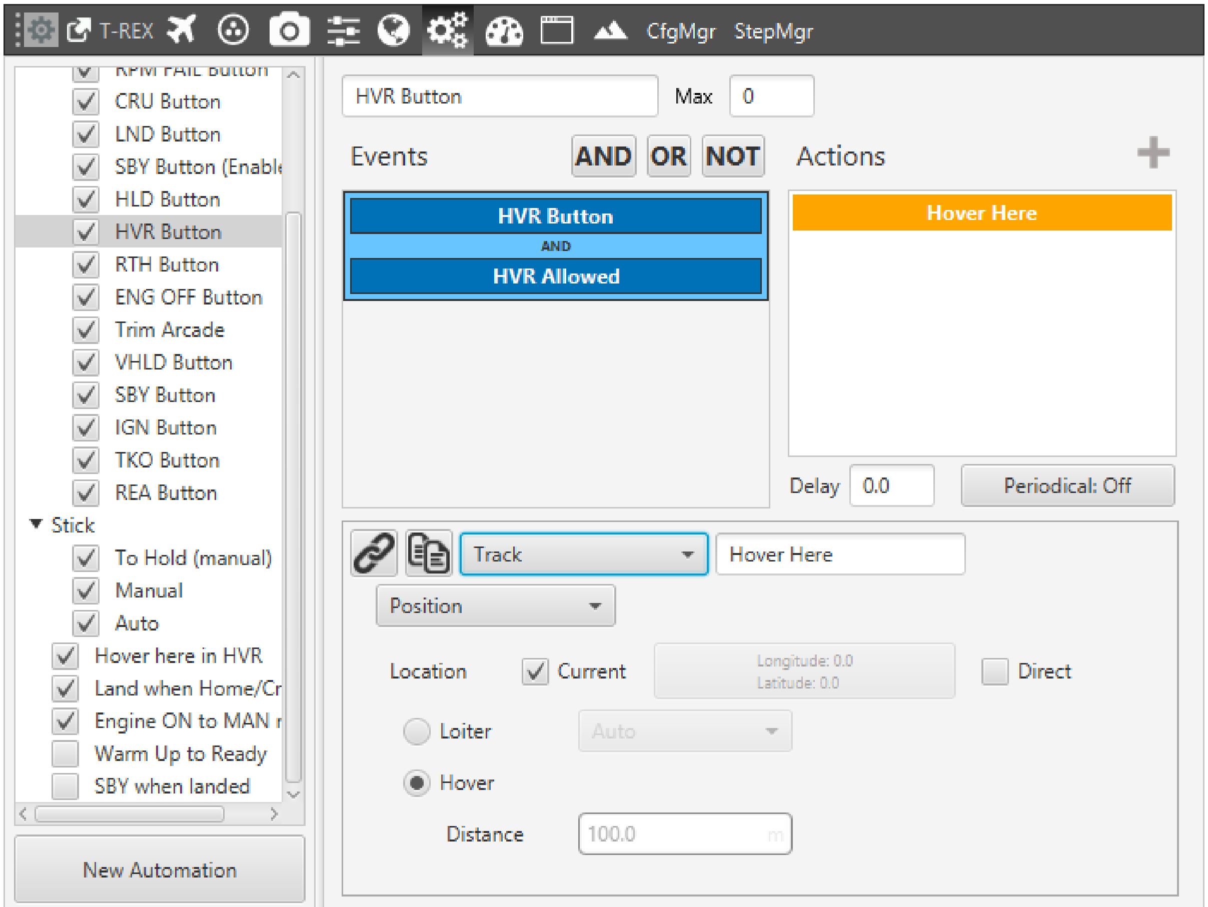
There are other types of button which can be created on the Veronte Panel. In this case will be detailed the Hover Button which allows performing a hover (only for multicopters, for airplanes could be set a loiter point) on the point of the map exactly when the linked button is pushed. The images below show the automation set.



Button Icon selection


T-REX
CfgMgr
StepMgr
Failsafe
Buttons
RPM FAIL Button
CRU Button
LND Button
SBY Button (Enable
HLD Button
HVR Button
RTH Button
ENG OFF Button
Trim Arcade
VHLD Button
SBY Button
IGN Button
TKO Button
REA Button
Stick
Hover here in HVR
Land when Home/Cr
Engine ON to MAN r
Warm Up to Ready
SBY when landed
New Automation
HVR Button
Max
0
Events
AND
OR
NOT
Actions
HVR Button
AND
HVR Allowed
HVR
Delay
0.0
Periodical: Off
Phase
HVR Allowed
Time
0.0
Cruise

Phases set

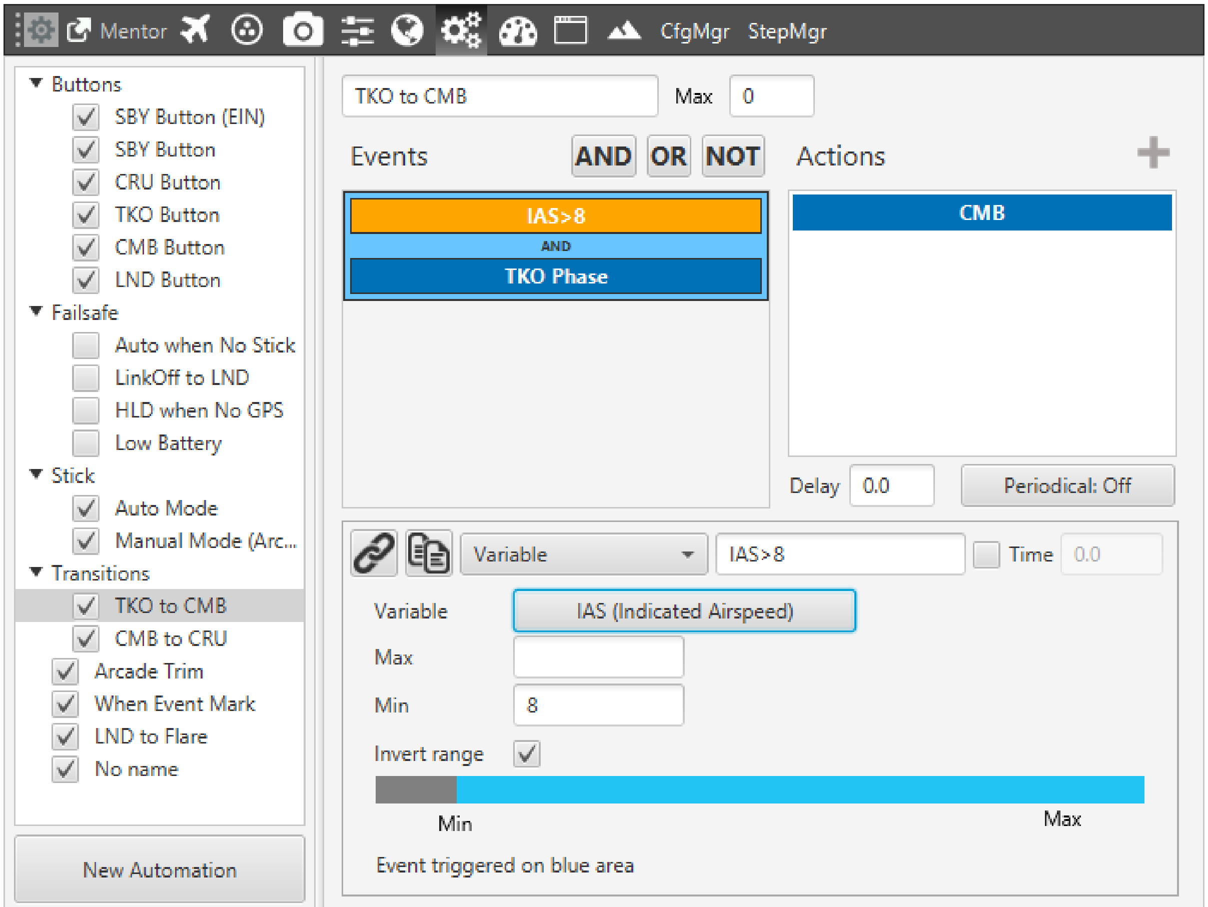


Hover button defining

11.3.2 Phase Changing

11.3.2.1 Takeoff to Climb Change

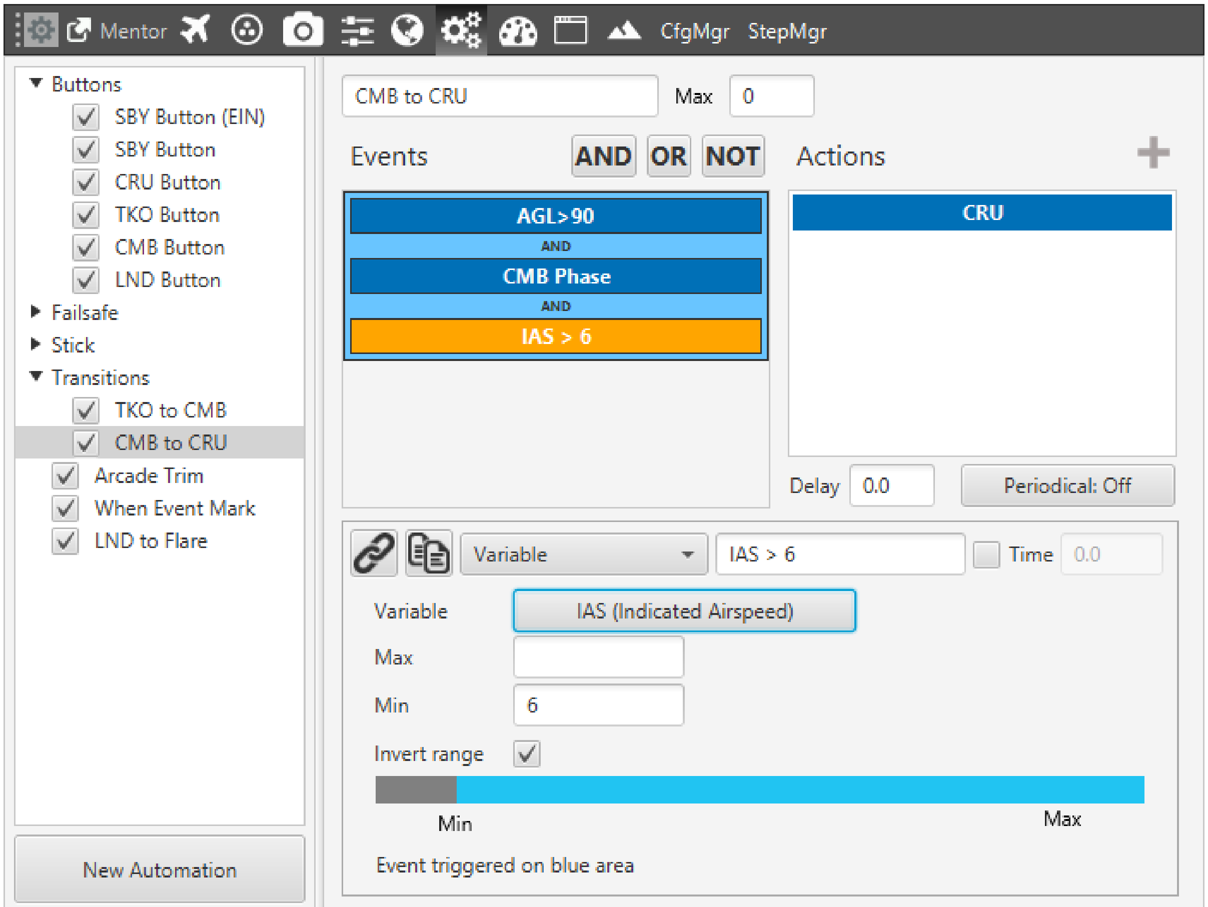
The change from the take off to the climbing phase, in this case, occurs when the IAS of the airplane on the runway is greater than 8 m/s. The phase condition is set on Takeoff phase and the action is defined to pass in Climbing phase.



Takeoff to climb automation

11.3.2.2 Climbing to Cruise Change

This kind of phase change is normally performed when a certain altitude is reached. In this case, the airplane changes to the "Cruise" phase where it starts to follow the path determined by the user. As a safety condition, the change will only happen when having a speed greater than 6 meters per second.



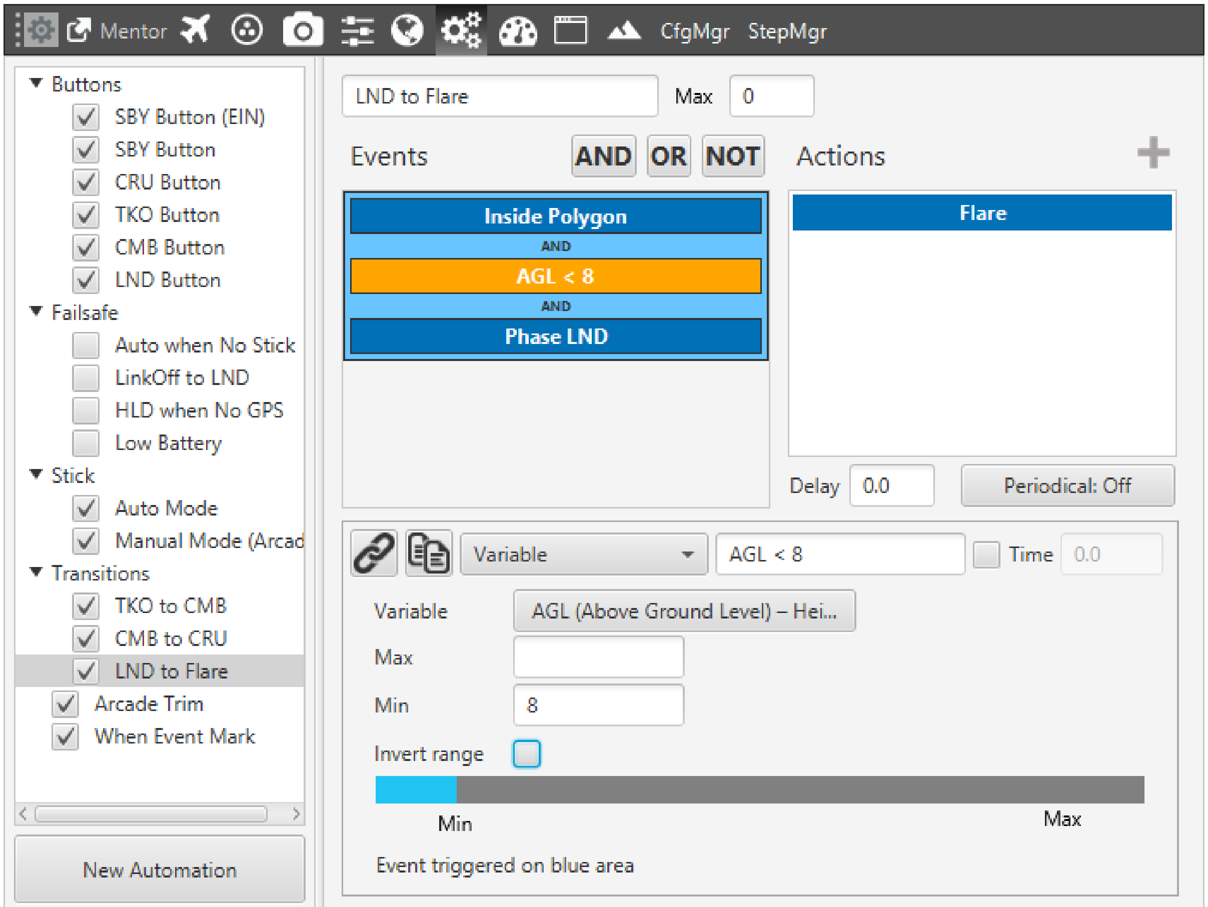
Climbing to cruise change definition

The AGL to the change of phase is 90 meters. Both the velocity and altitude are specified through an event of type "Variable".

11.3.2.3 Landing to Flare Change

The change from the landing to flare phase is performed when the aircraft is above the runway and about to touch the ground. This idea is implemented with a set of three events:

- **AGL:** the aircraft is below a certain altitude, in this case, 8 meters.
- **Inside Polygon:** Mentor enters in a polygon defined in the map on the runway head.
- **Landing Phase:** the aircraft is landing phase.



Landing to Flare

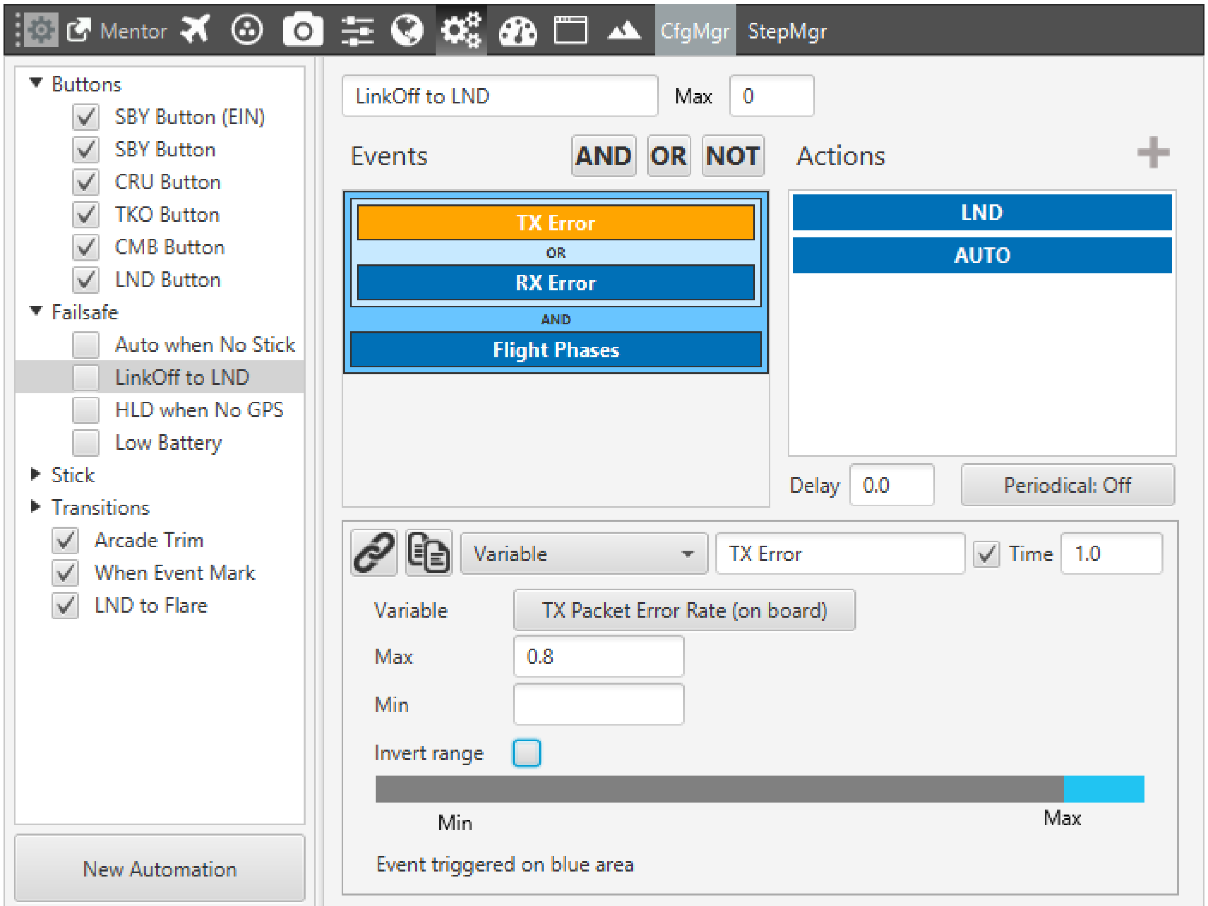
When these events are fulfilled, airplane enters in the Flare phase and lands.

11.3.3 Failsafe

Veronte allows users to create different types of Failsafe automations according to they need combining different Actions and Events. The more common automations are shown below.

11.3.3.1 Radio Error

When the radio connection from the ground station to Veronte Air is lost, the aircraft is forced to change to auto mode and landing phase.

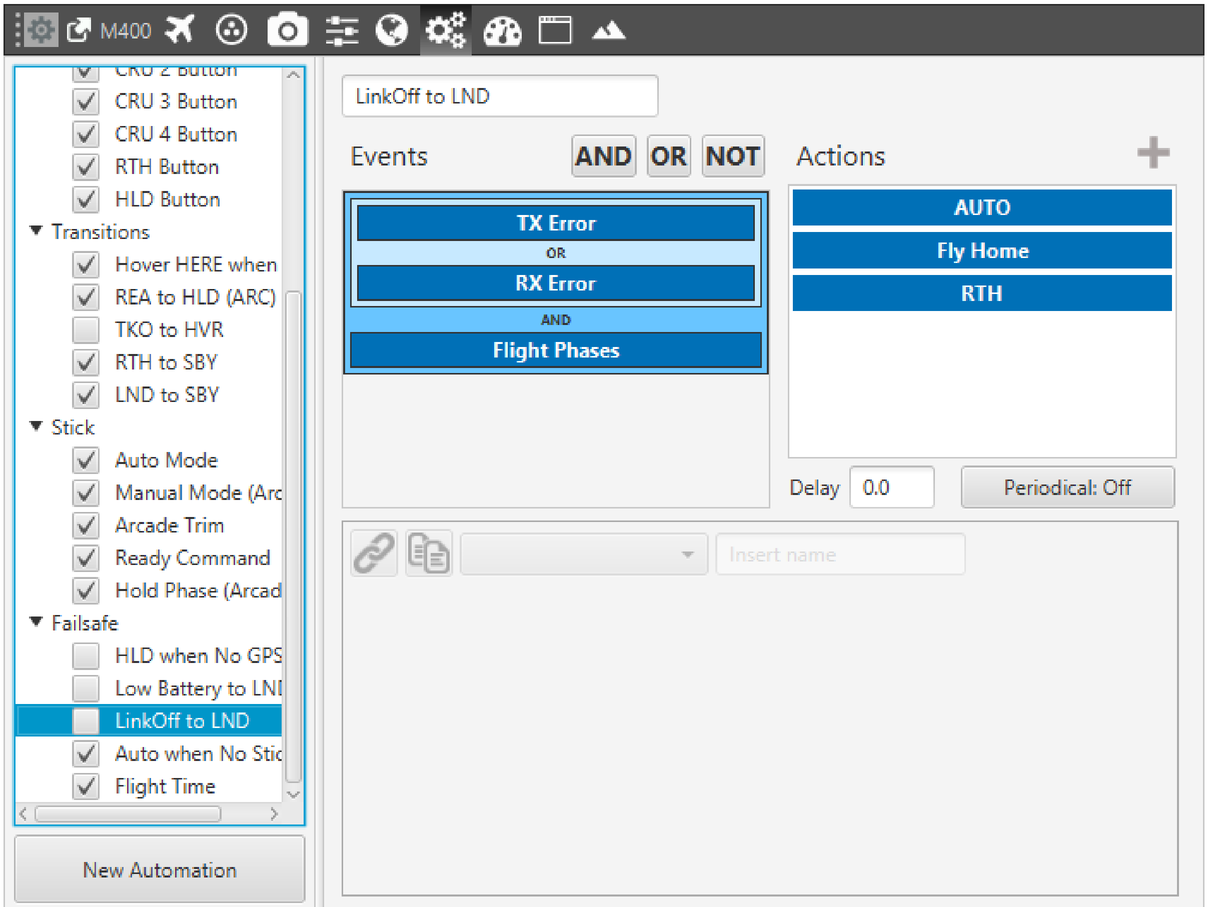


Failsafe for radio link-off – Plane

When the value of the flag that evaluates the TX or RX error ("TX/RX Packet Error Rate (on board)") has a value of 1 (0.8 is established for safety), and the aircraft is in a certain phase (climb and cruise), it is forced to land.

Video Tutorial: https://www.youtube.com/watch?v=mmzRw9V9OCs

When using a multicopter the same proccess is done with some changes.



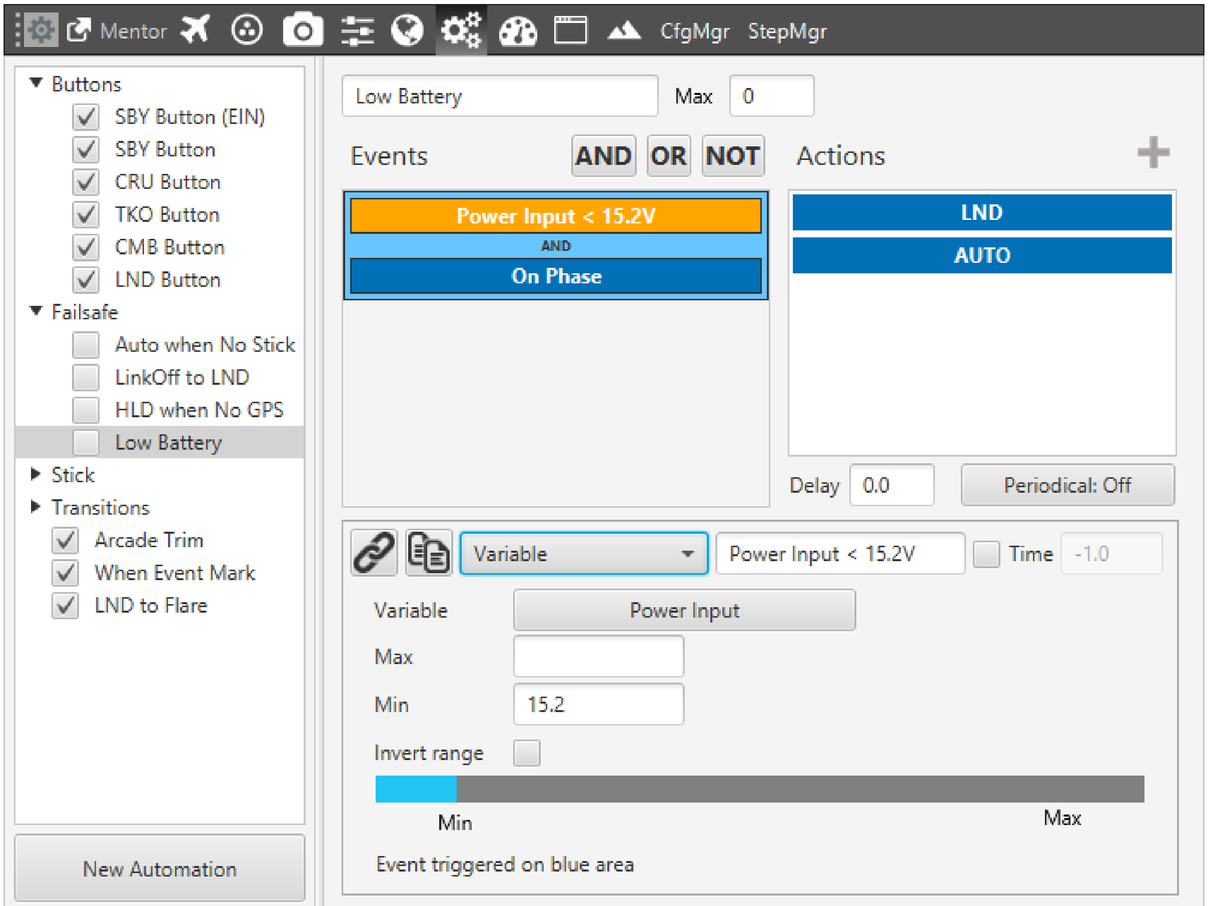
Failsafe for radio link-off – Multicopter

The actions carried out now are:

- **Mode:** select a change to AUTO mode.
- **Phase:** change to the phase Return to Home, in this phase is definied the Home point. See the example of this phase for the quadcopter M400 in section *Mission Phases*.
- **Go to:** as safety condition, this action is added and the Home point is selected.

11.3.3.2 Low Bettery

When the battery is below a certain level (15.2 Volts in this case), and the aircraft is in climb or cruise, it is automatically commanded to land.

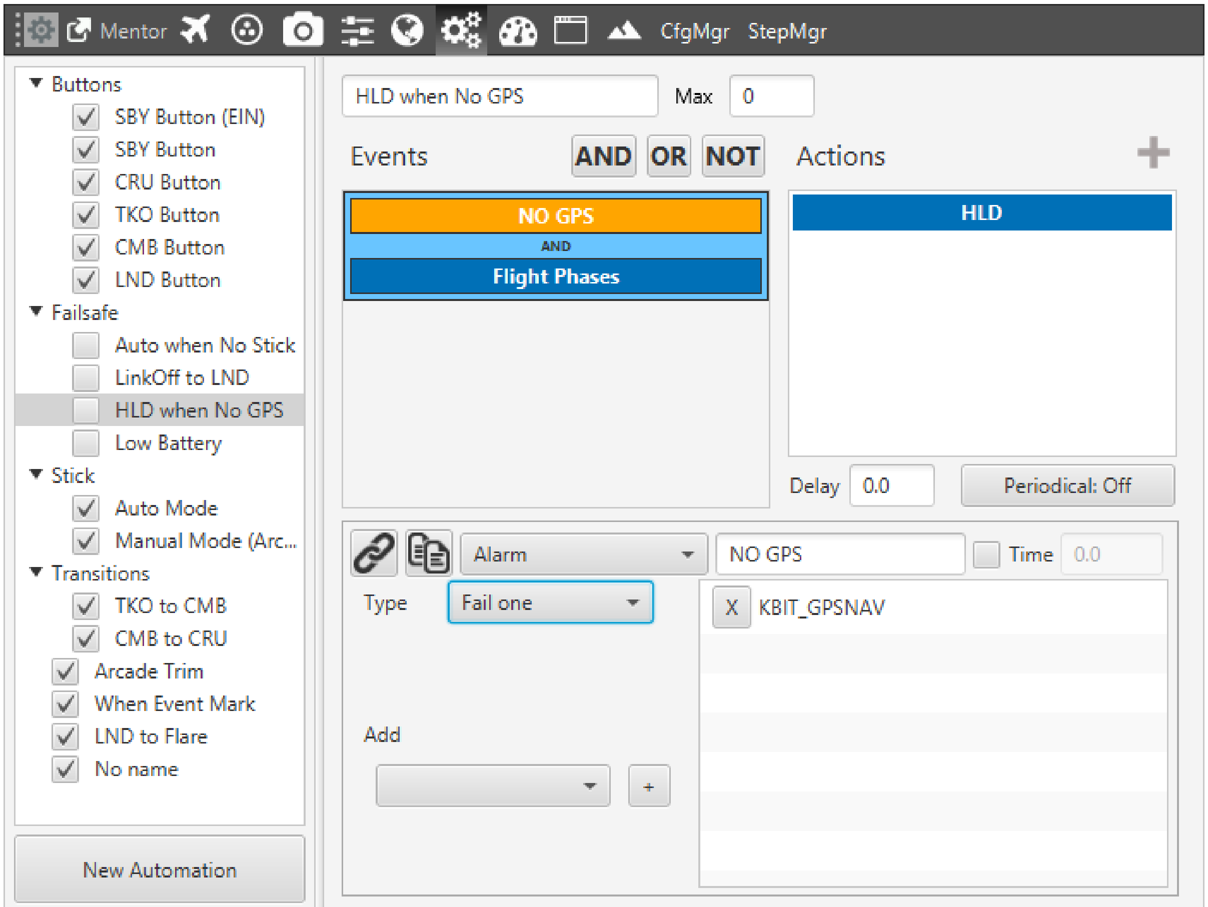


Failsafe for low battery

11.3.3.3 GPS Signal lost

When the UAV is not receiving the GPS signal, the system automatically adapts the navigation algorithm to avoid a possible accident by obtaining the aircraft attitude trough the AHRS reference system. Besides that, it is also needed that the aircraft changes to a flight phase where the control is made over the attitude angles instead of position variables (heading, flight path). This phase is commonly known as HOLD, which has a guidance to keep pitch, roll and IAS at a certain value, and with PIDs in pitch and roll having two blocks for that angles and its rates.

The event of this fail safe automation will be the loss of GPS signal (type alarm), while the action is type phase (Hold).



Failsafe for GPS signal lost

11.3.4 Photogrammetry

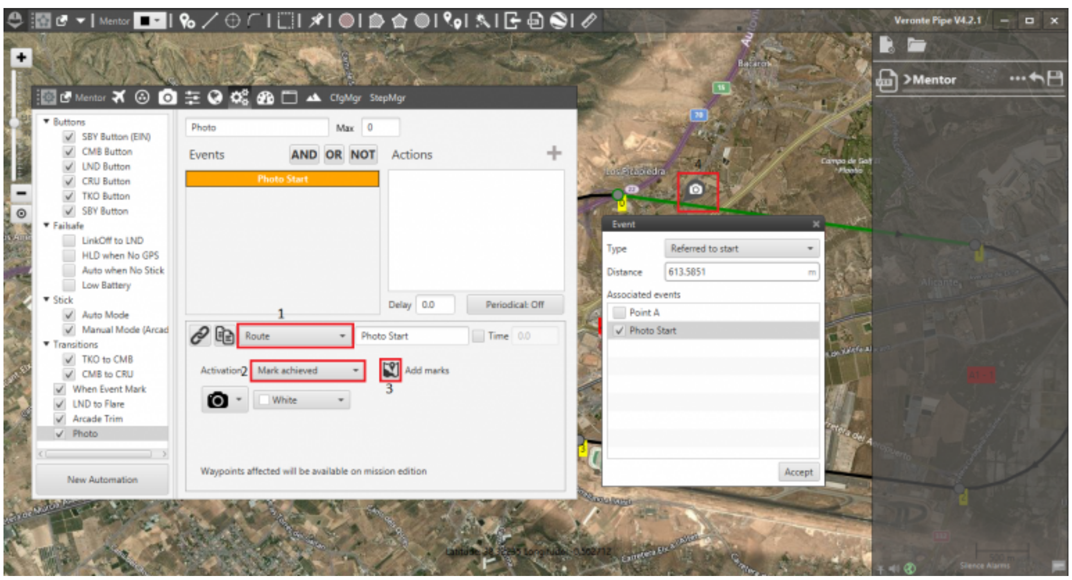
Veronte Pipe incorporates the Photogrammetry mission, see section Photogrammetry. It is possible to create it without using the tool incorporated within the mission toolbar.

In the photogrammetry missions, the platform is wanted to take photos during the time that it is following a path previously defined. Commonly, the photo taking process begins at a certain location over the route and is repeated each certain distance. To create this automation the process detailed now has to be followed.

1. Create an **Event Marker** at the point where the photogrammetry mission has to start.

Photogrammetry Automation 1

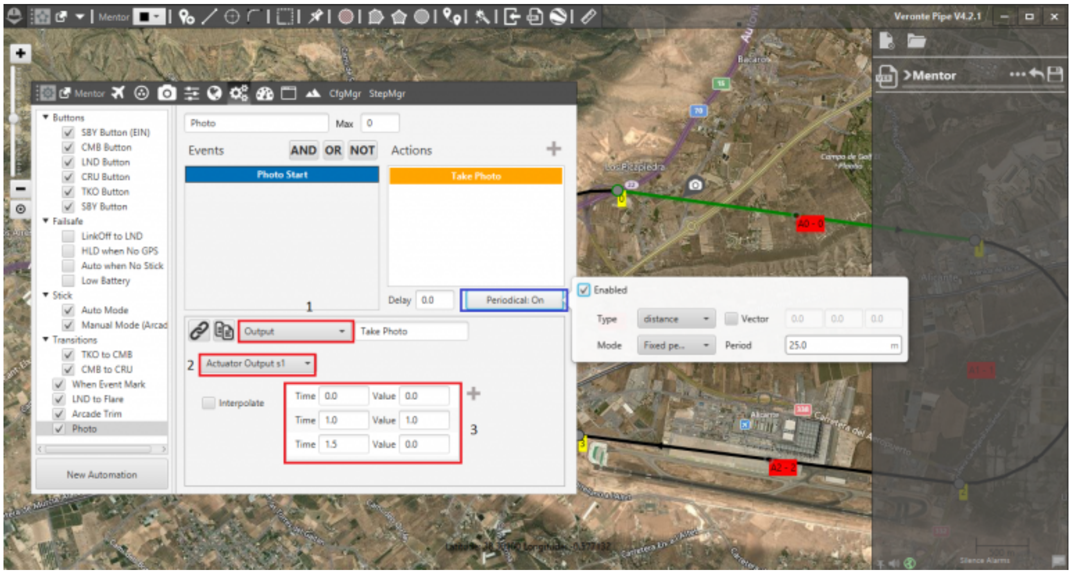
2. That **Event Marker** has then to be linked with an event that will trigger the process. Go to Automation in order to create it.



Photogrammetry Automation 2

The event is type **Route (1)** and will be triggered when the point is reached (**2**) (the other option is when flying towards the point). If the event is linked with a waypoint instead of an Event Marker, the point is selected in the map with button (**3**). Finally, with the event created in the automation menu, double-clicking on the event marker allows to link this automation to it.

3. The last step is to create the actions to be performed when the event is triggered.



Photogrammetry Automation 3

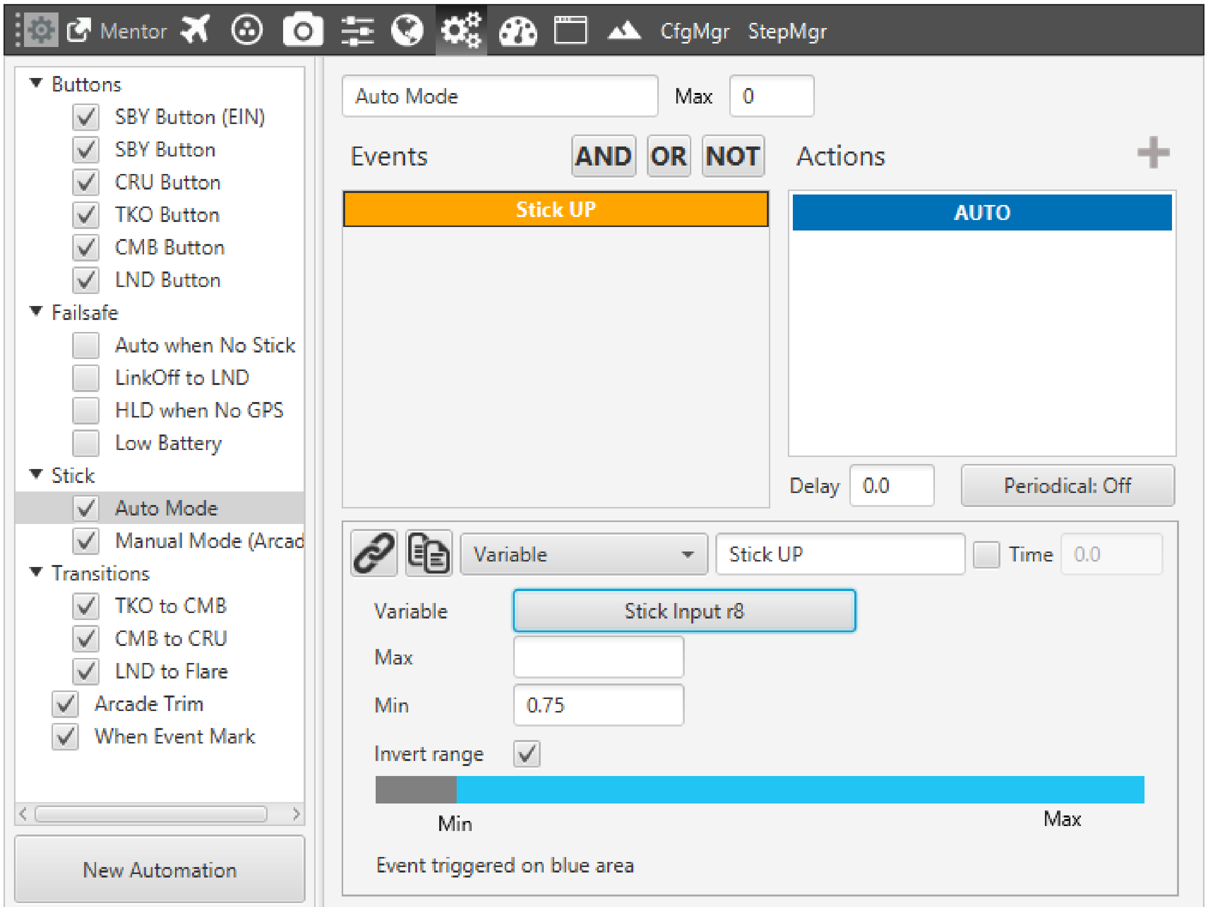
When the point is reached, a timer starts to measure the distance between the locations at which a photo will be taken. **Periodical** (blue box) has to be enabled, users have to indicate that the timer measures distance (the other option is time), and finally indicate that the distance measured is a fixed value (25 meters in this case).

The action type (**1**) is **Output**, which is used to send a signal trough one of the output pins of Veronte. In (**2**) is selected the actuator that is connected to the camera (or with the device that is controlled with this automation). Finally, in (**3**) are indicated the values of the PWM signal at each time instant.

11.3.5 Others

11.3.5.1 Stick Auto

This automation changes the control mode according to the command sent by the radio controller.



Stick Auto automation

In this case, the channel that controls the mode is the 8, so according to its value the mode is changed. With an event of type Variable the automation is defined. The process is the same for the manual mode, but when the r8 variable has a value lower than 0.25.

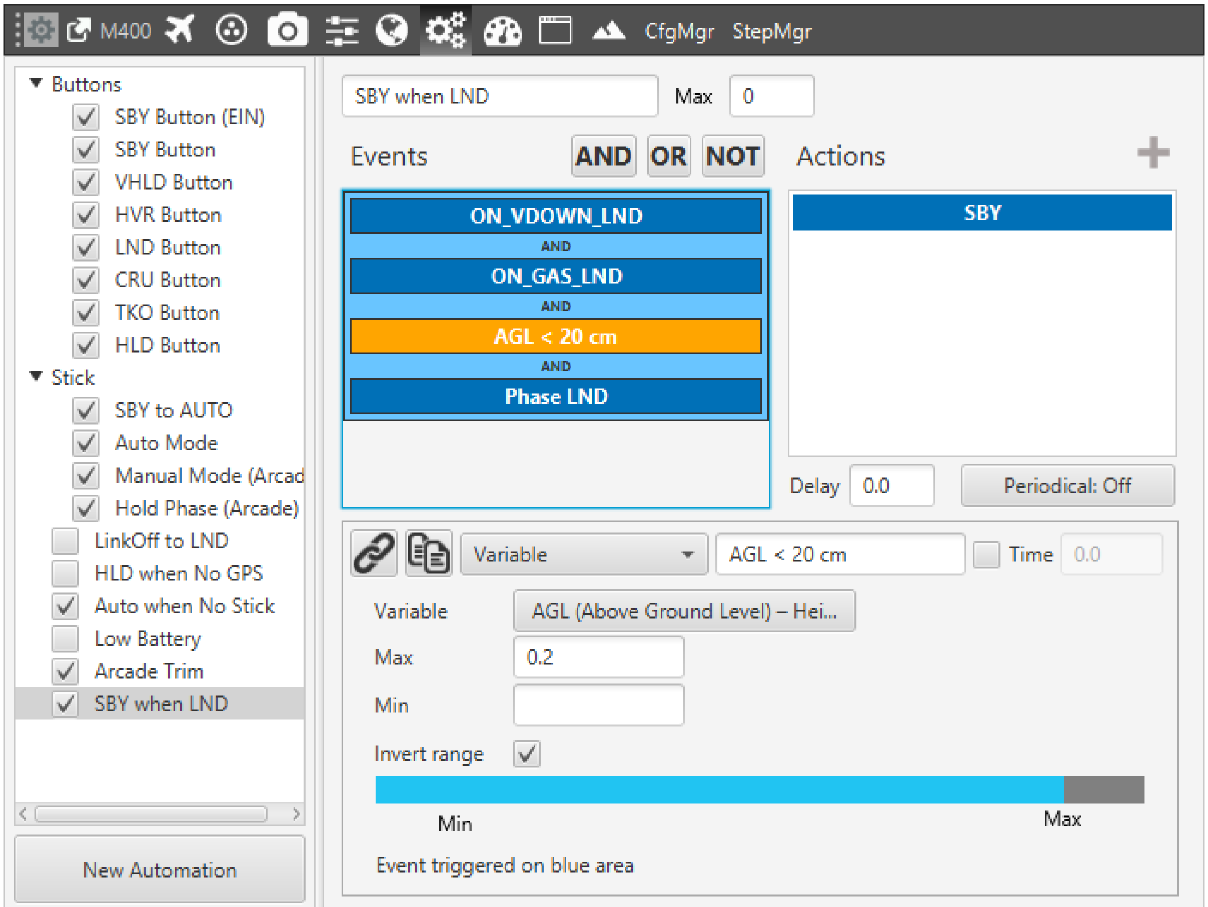
Video Tutorial: https://www.youtube.com/watch?v=C3A6P1jgFV4&feature=youtu.be

11.3.5.2 Automatic Landing (Multicopterss)

If the user wants the multicopter stops the motors when landed (pass to Standby automatically), it is possible to configure an automation which performs this action.

Normally, to control the automatic landing, the following variables are used:

- Ground Speed Down
- AGL (Above ground level altitude)
- Thrust control
- Landing current phase



Landing auto configuration

Variable values must be chosen depending on multicopter (rotor number, dimensions, altitude sensors, etc…) and an integral control increase can be taken into account in order to perform a faster RPM descending and a better landing.

The automations are the mechanisms used to perform an action when some event is verified (using the Boolean Logic AND, OR and NOT tools). These actions could be a change from one phase to another, taking a photo, dropping a payload and so on. In this section, the following automation types will be detailed:

- Veronte Panel buttons
- Phase changing
- Failsafe (link-off, power low, GoHome point)
- Photogrammetry
- Others

11.4 Lidar Integration

11.4.1 Lidar Configuration

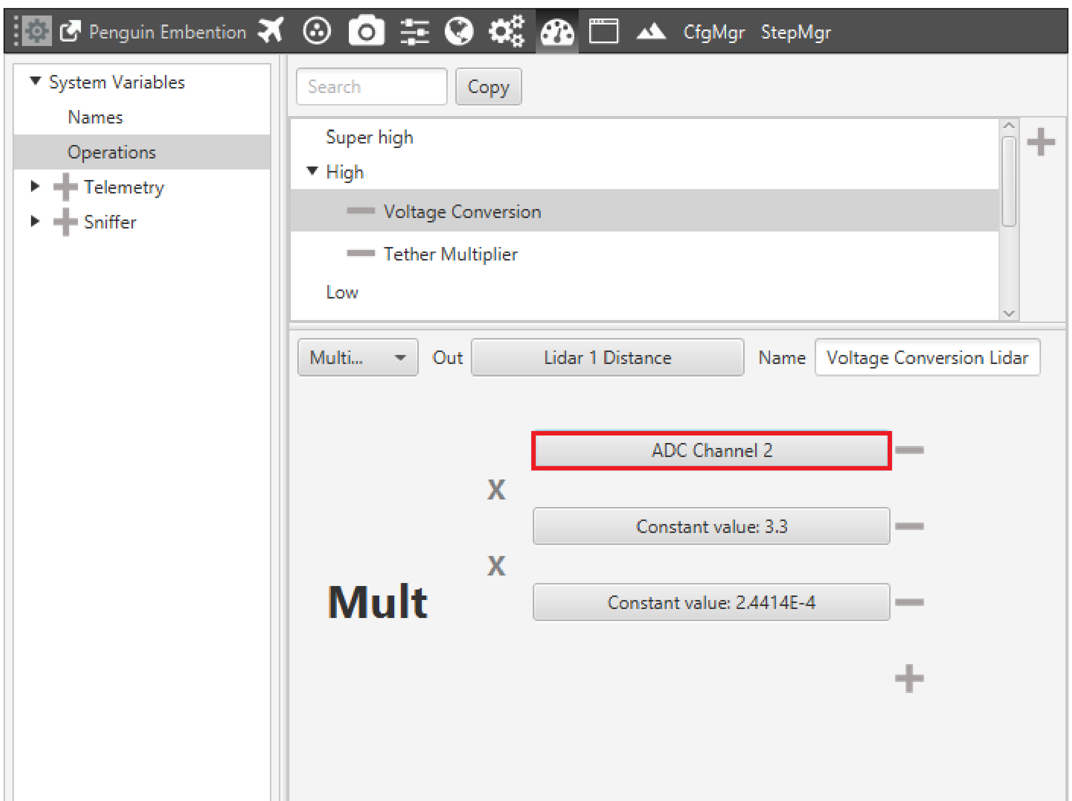
The integration between Veronte and Lidar is performed using the analog interface on the main connector which produces a voltage proportional to the measured altitude. In this example, a lightweight laser altimeter for above-ground level measurement (SF11) is configured.

You can get more information about our *Lidar* clicking here

11.4.2 Veronte configuration - ACD In

When Lidar connection to Veronte is performed, the autopilot configuration has to be completed (see section Connections). To do this, it is necessary to configure the operations to set the correct conversion from the voltage measurement to the physical variable.

1. Go to **Variables/System Variables/Operations**.
2. Configure the following operation, see section *System Variables* for more information on how to create a custom operation.



Lidar Configuration

3. Save the result of this linear operation in a variable (selected by the user), in this case Lidar 1 Distance [m].

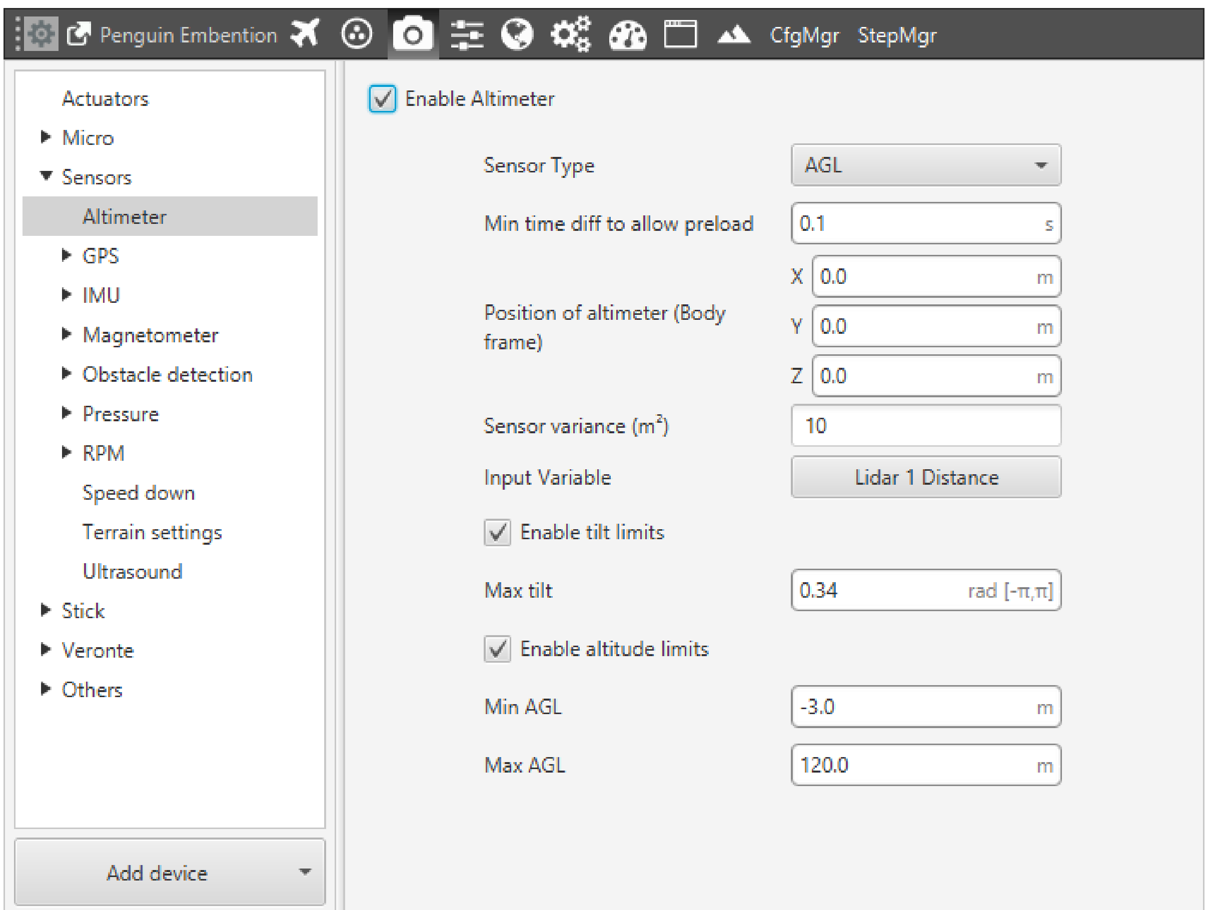
Users have to select the ADC channel where the device is connected (ADC Channel 2-6), and set the previous operation that converts the voltage measured.

It is possible to create an alarm variable that will be triggered when the measured value is lower or greater than a fixed values, see section *Gauge Display* . For example, in this case, an alarm could be triggered when the measure is lower than 0 [m] or greater than 120 [m] meters (Lidar limits). The alarm is stored in a variable selected by the user (in this case, the alarm Power B error is used).

Lidar alarm example

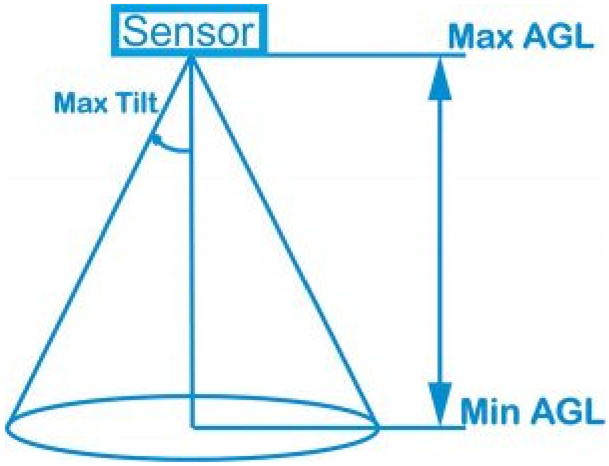
11.4.3 Veronte Configuration - altimeter

The Altimeter configuration has to be defined in the Devices panel of Veronte (Devices > Sensors > Altimeter). Configuration tools are explained in section *Altimeter* but a good configuration for the SF11 Lidar is the one showed in the following figure.



Altimeter configuration example

The following figure shows a diagram with the values of maximum and minimum Above Ground Level altimeter, and the maximum tilt angle.



Payload limits

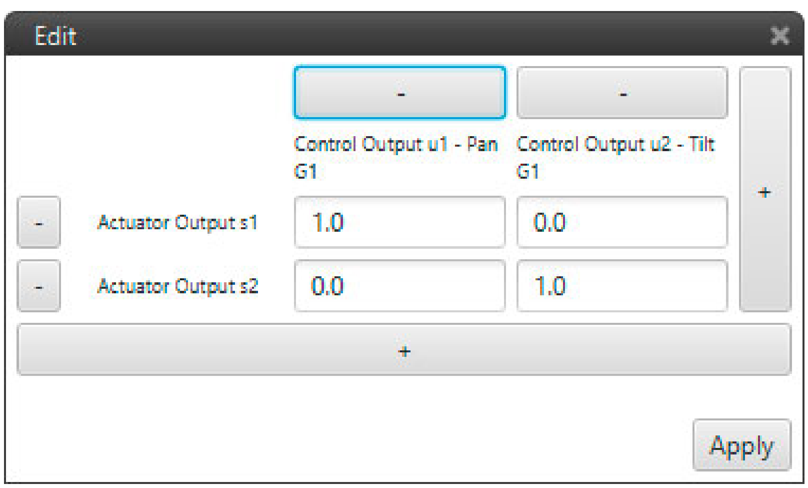
It is important to correctly define the distance of the altimeter from the gravity center and the AGL and Tilt angle limits in order to avoid a possible measurement errors.

It is possible to test the correct measurement of the Lidar by positioning it at different AGL altitudes and checking the associated variable in the workspace.

11.5 Veronte Tracker

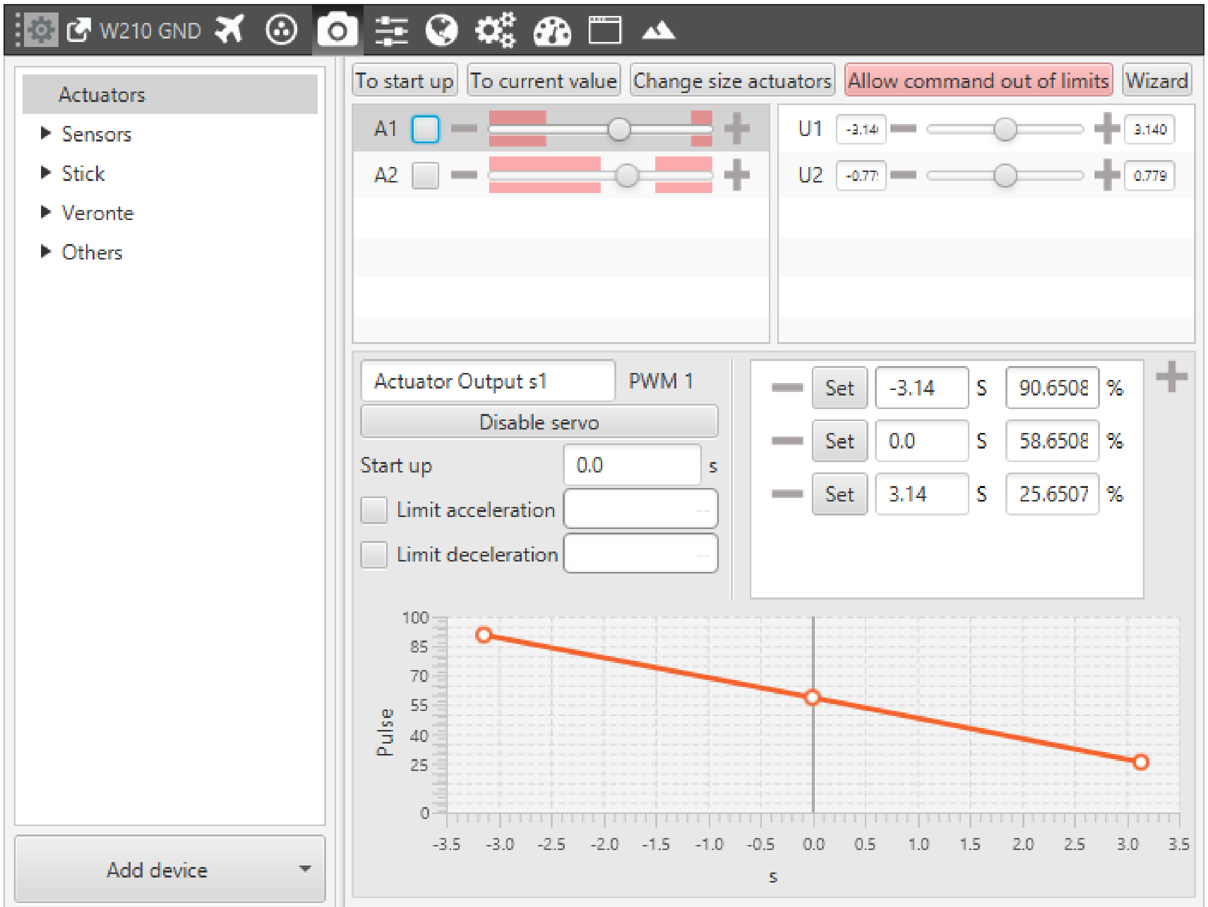
11.5.1 Veronte Ground Configuration

The first step of the configuration is the creation of the SU matrix including 2 controls (PAN-TILT) and 2 servos, the ones installed on Veronte Tracker in order to move Pan and Tilt axes.



SU Matrix

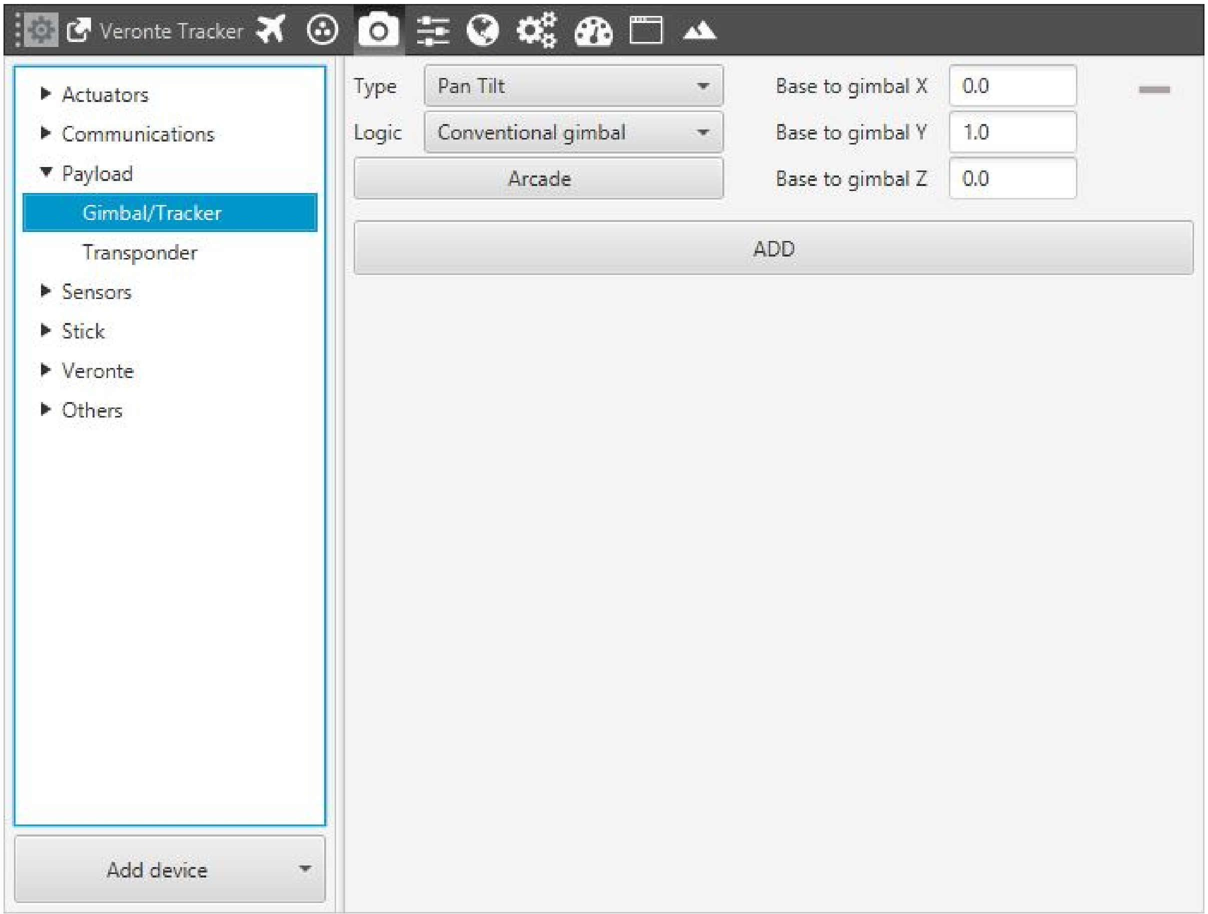
Once the SU matrix is complete, it is possible trimming both servos with the same procedure of aircraft servos in the **Devices > Actuators panel**. The user must set the center of the antenna and then maximum and minimum angles. The procedure has to been repeated for both servos.



Servos Trimming

When servos are trimmed, the tracker has to be configured in the Devices >Payload panel as:

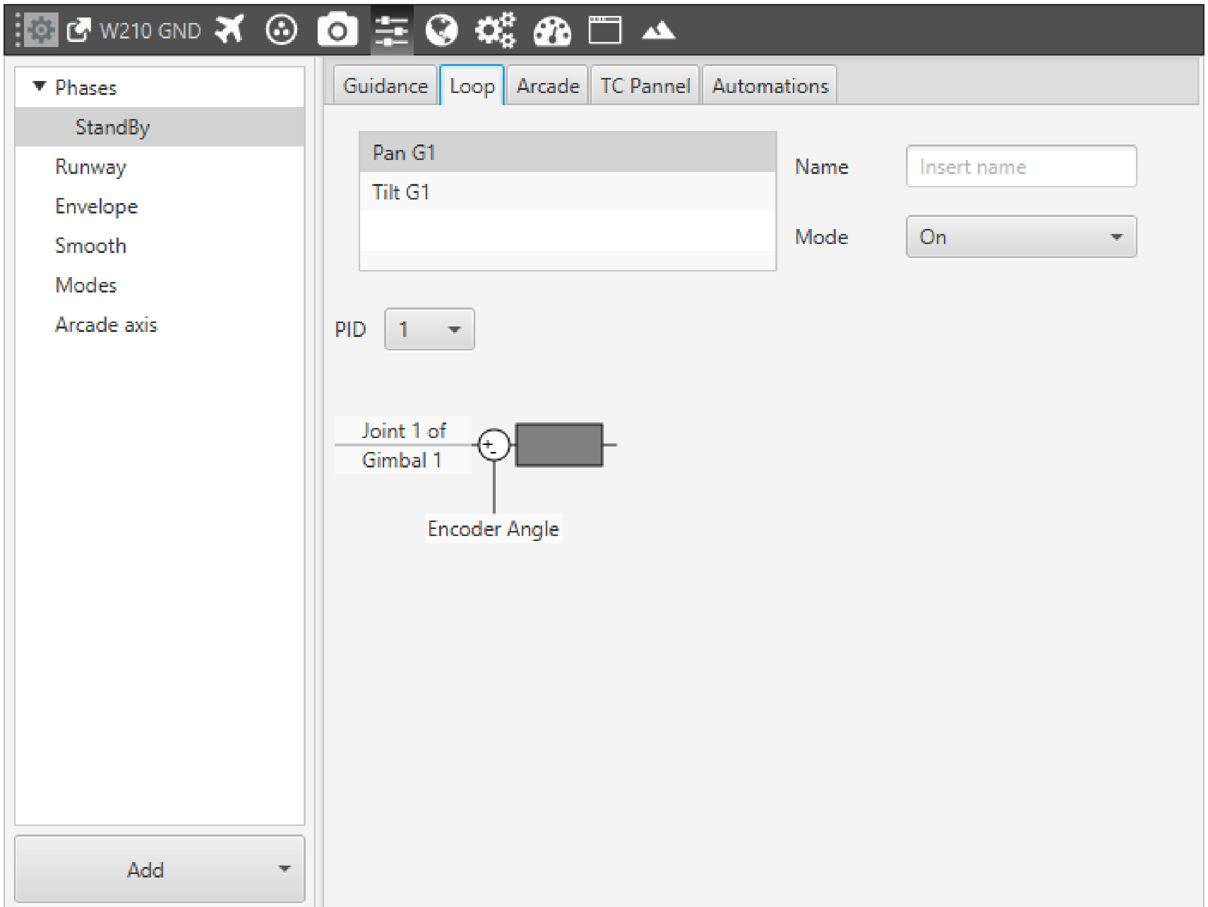
- **Type:** Pan Tilt.
- **Logic:** Conventional Gimbal, with Horizontal (Pan) and vertical (Tilt) axes.



Tracker Payload configuration

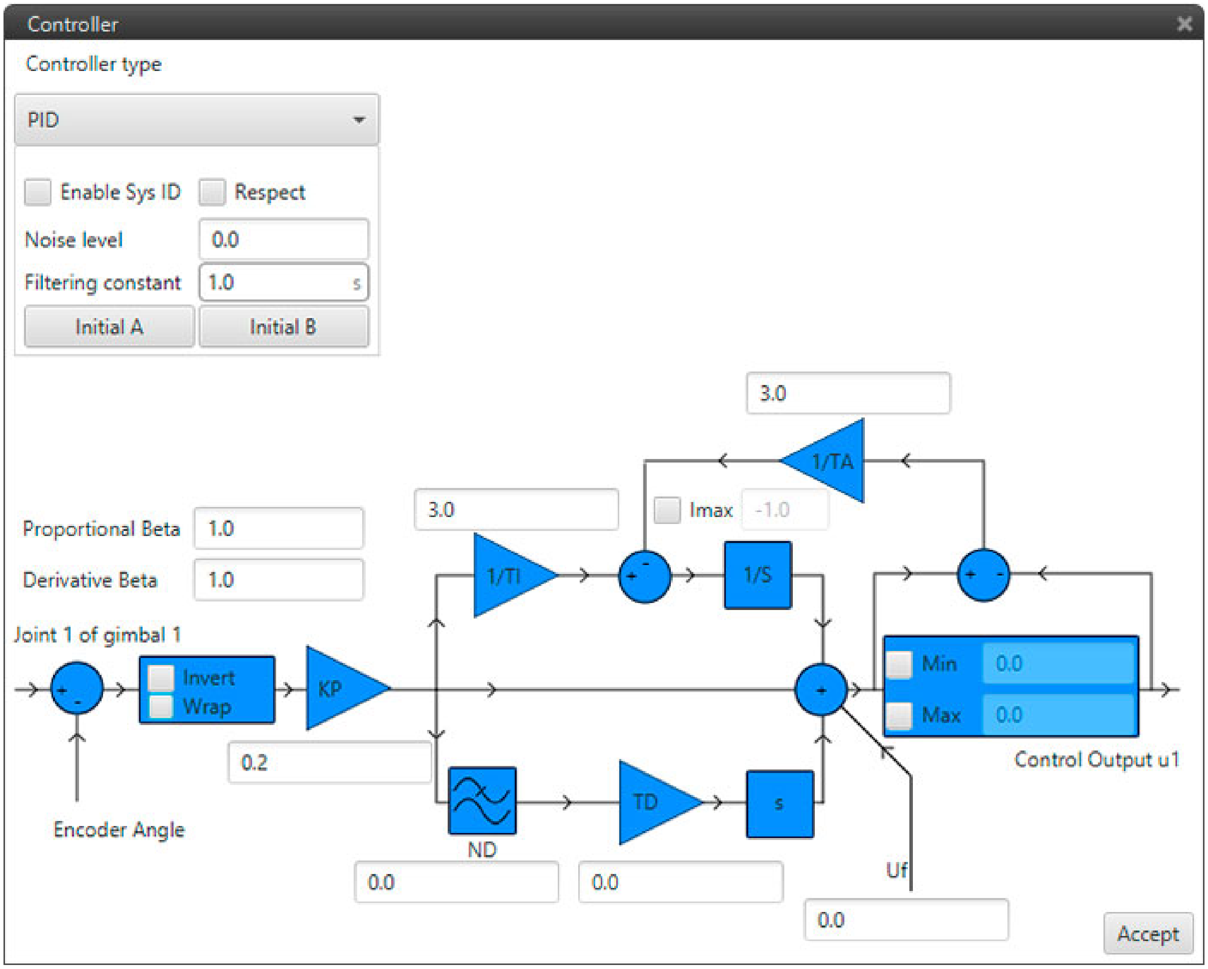
Next step is the configuration of the control loops for each axe: Pan and Tilt.

- Pan control loop is a closed loop with Joint 1 of gimbal 1 entry variable (the desired Pan angle) and the Encoder Angle variable in the feedback loop. The output is configured as the SU matrix: Control Output u1.



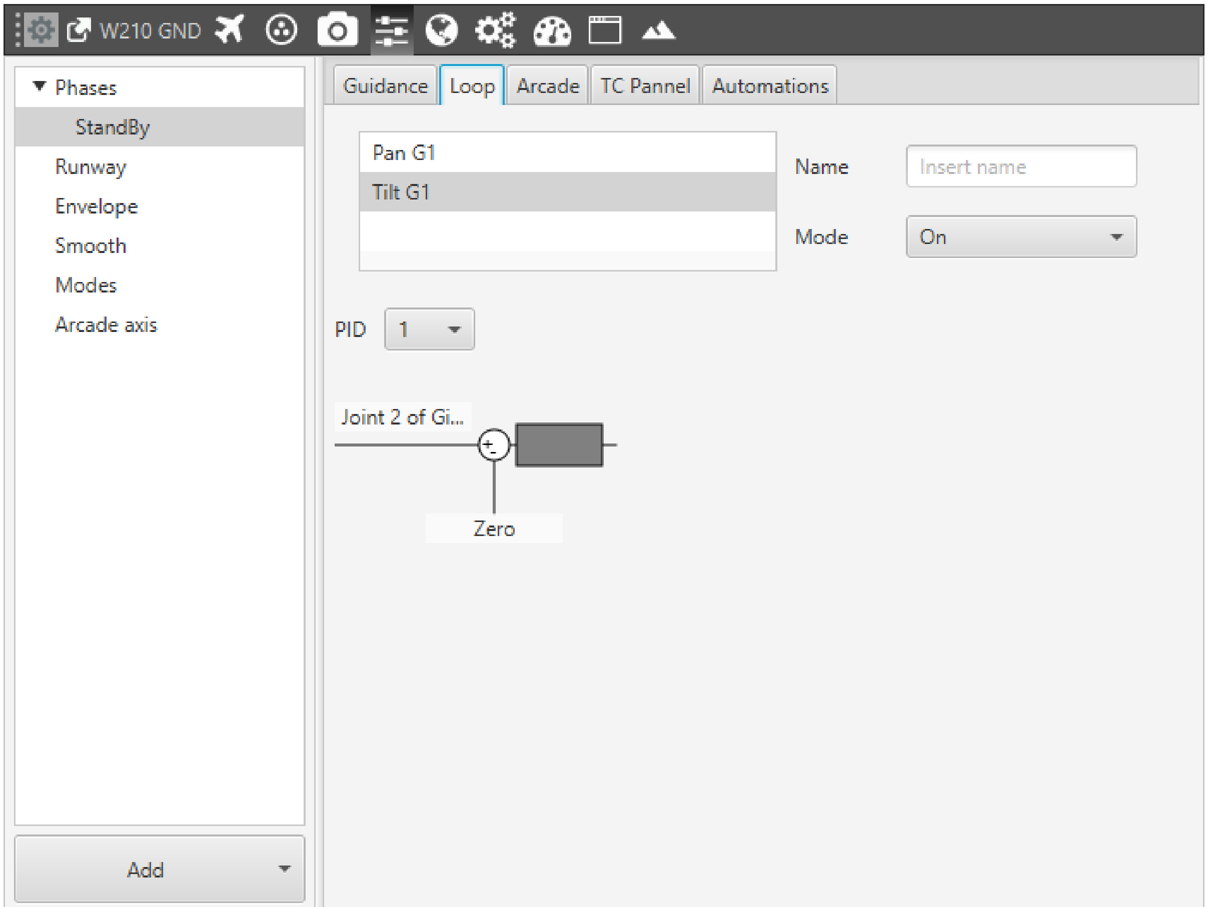
Pan Loop Configuration

A typical setting for the loop is presented in the following figure. The proportional gain is set at 0.2 and the integral gain at 3.0.

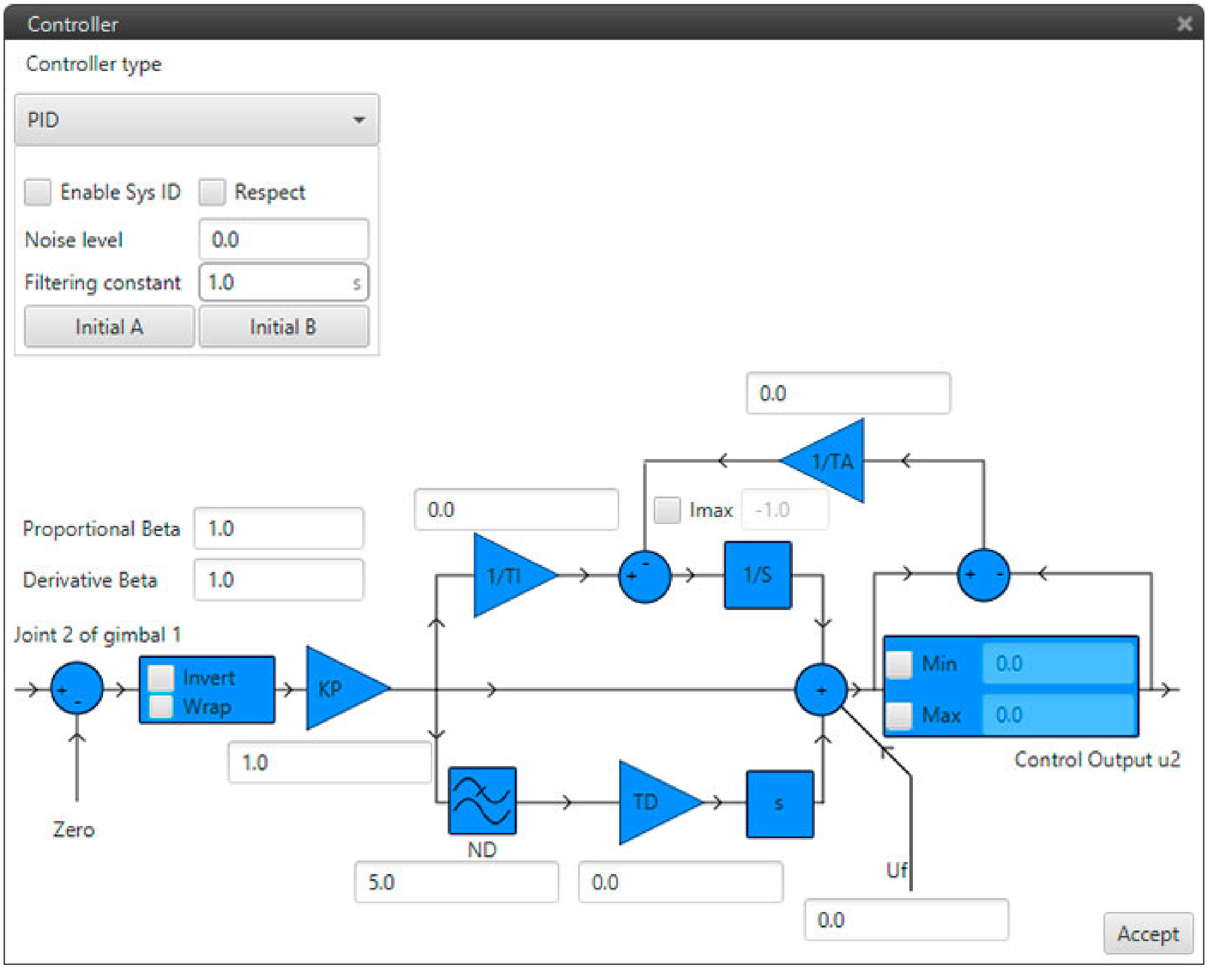


Pan Control Loop

- For the Tilt control, an open control loop is sufficient. **Joint 2 of gimbal 1** is the entry variable (the desired Tilt angle) and the **Zero** variable is configured for the feedback loop. For those cases where an **encoder** is present it should replace **Zero** as feedback for Tilt angle. The output is configured as the SU matrix: **Control Output u2**.

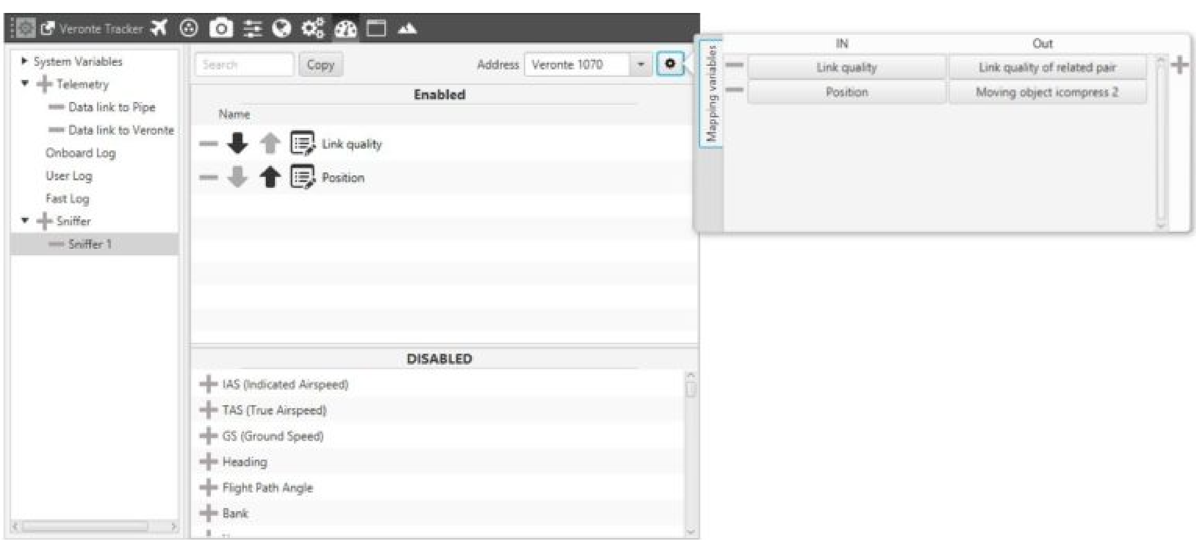


Tilt Loop Configuration



Tilt Control Loop

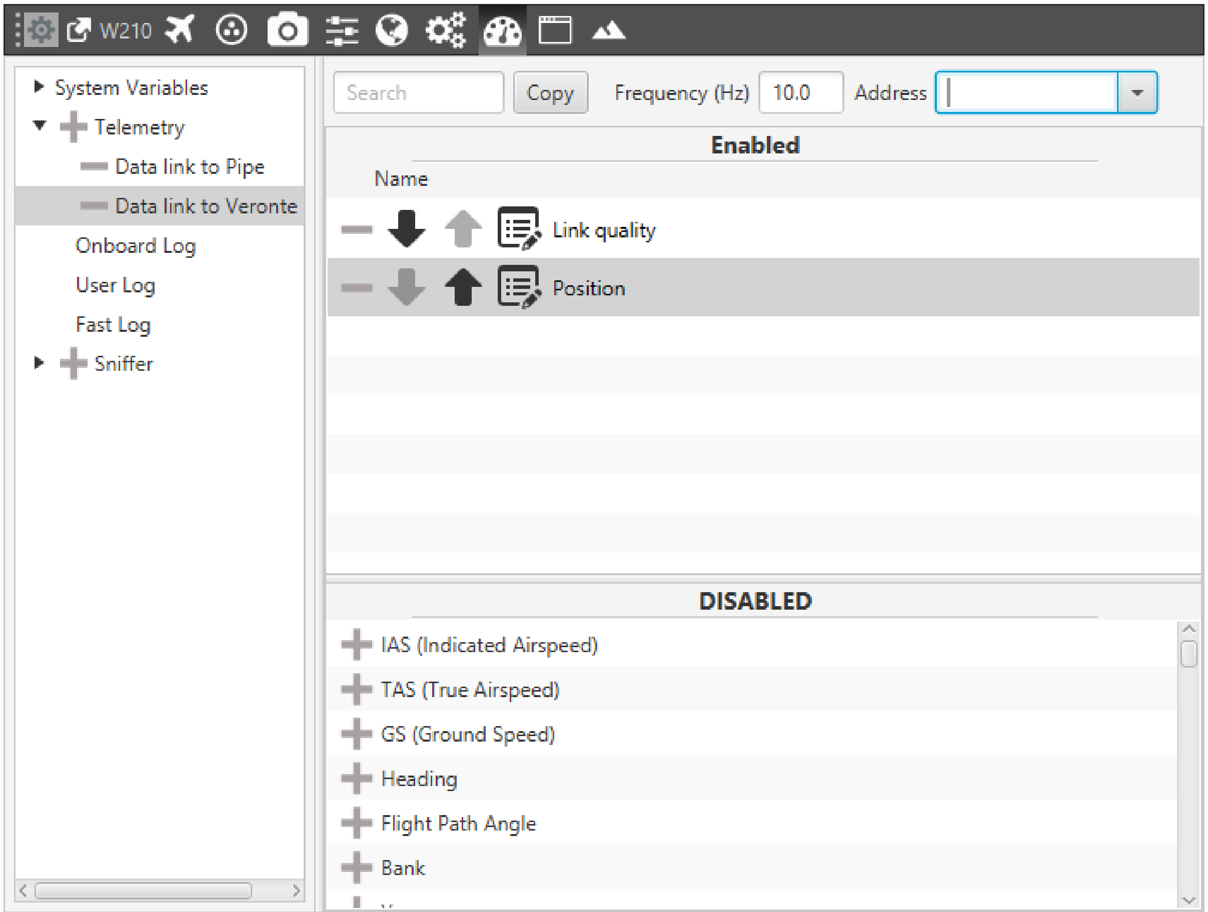
The last step of the Veronte Ground configuration is the Sniffer setting. Configuring a new Sniffer means allowing the Ground Station "knowing" the Veronte Air (UAV Address) position during flight and associating it with a Moving object. Veronte Air has to be configured to send its position to Veronte Ground. On **Address** it has to be selected the Veronte Air to be tracked.



Sniffer Ground Configuration

11.5.2 Veronte Air Configuration

Configuration of Veronte Air includes only one step which is the Data Link configuration. The new Data Link has to be defined as showed in the following image: the variable to send is the Position and **Address** has to be set according Veronte Ground (Unit that will receive this data) with a frequency of 10.0 Hz.



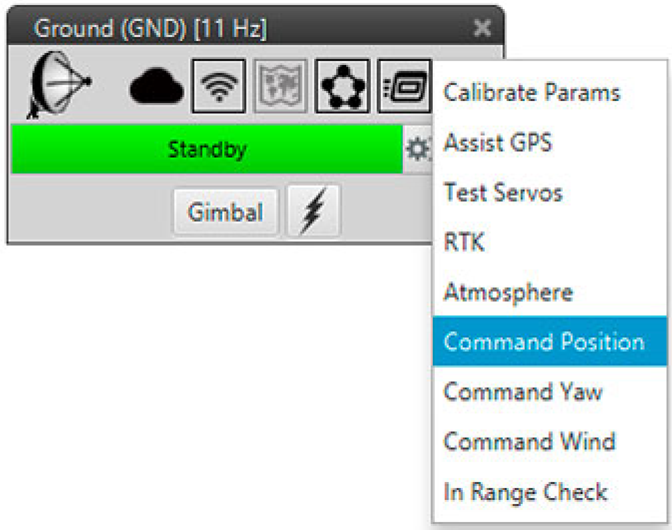
Data Link Veronte Air Configuration

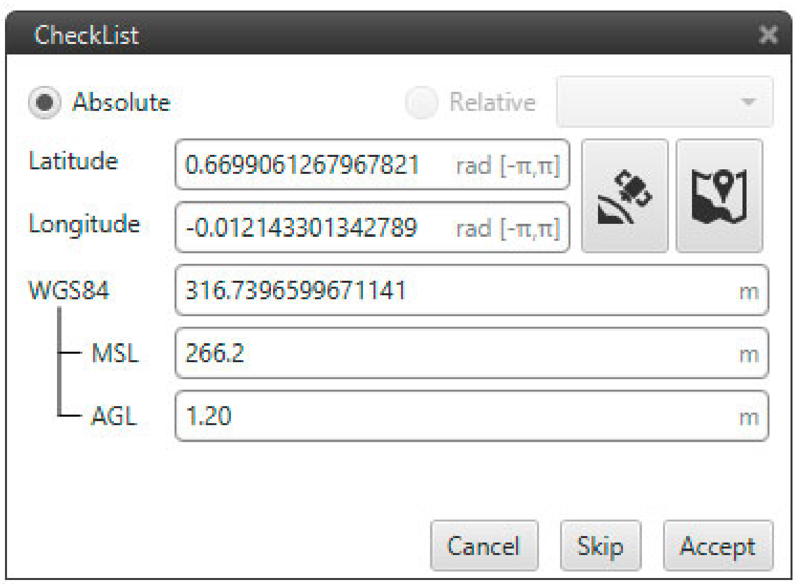
11.5.3 Tracking Operation

Tracking Operation

In order to activate the tracking, the user must follow some steps.

1. First of all, it is necessary to configure the Tracker (Veronte Ground) position on the map. To do this it is sufficient to go in the **Veronte Ground Panel**, open the **Run Task** and then the **Command Position** window.

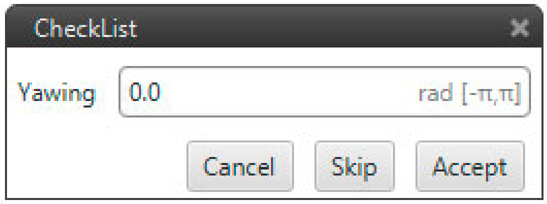




Tracker Position configuration

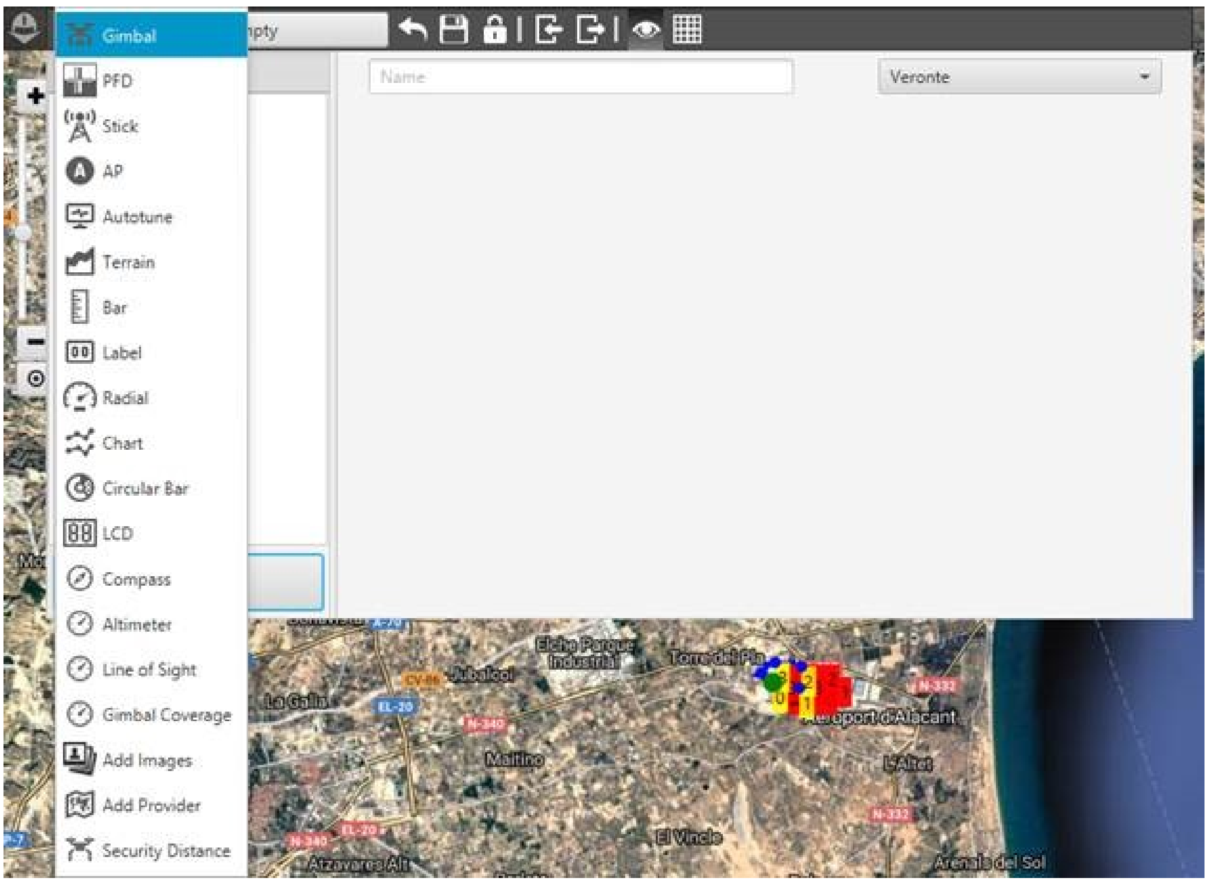
In this panel, it is necessary to insert the tracker position. The user can perform this action inserting Latitude and Longitude values manually or directly clicking on the Map location. The tracker altitude has to be defined too: normally, the tracker tripod is set at 1.20/1.40 [m] so this value has to be inserted in the AGL altitude.

2. Introduce **Yaw initial orientation** of the tracker. Go into **Veronte Ground Panel**, open the **Run Task** and then select **Command Yaw**.



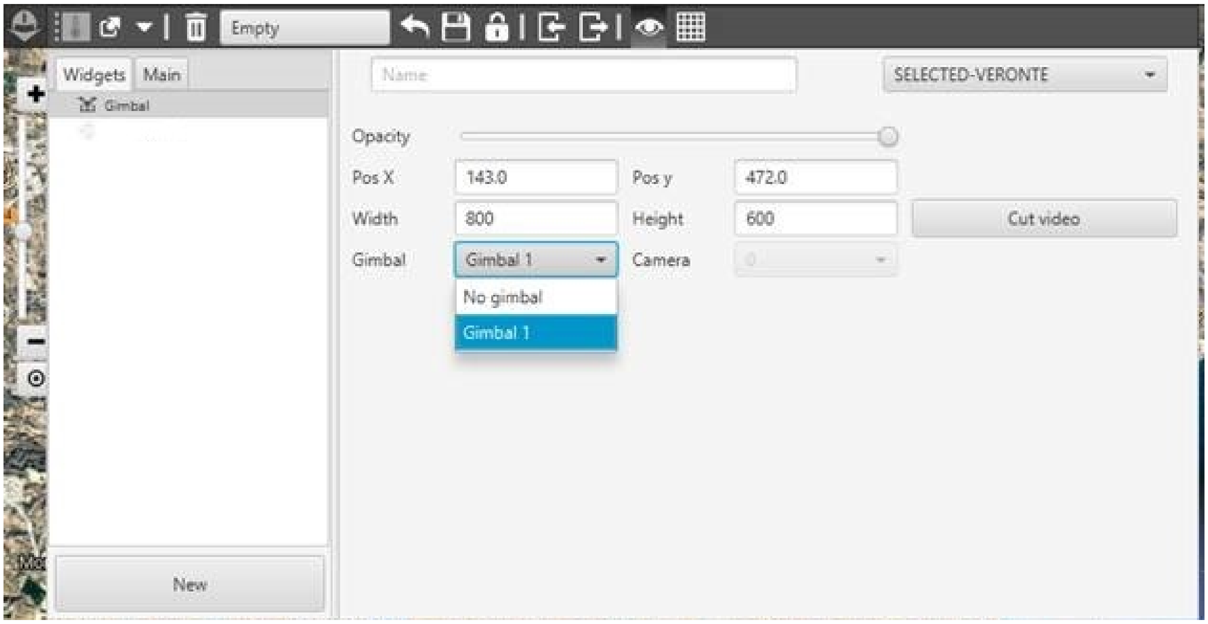
Tracker Orientation configuration

When Veronte Tracker is completely positioned, it is possible to activate the tracking. To do this it is necessary to add a Gimbal widget on the Workspace.



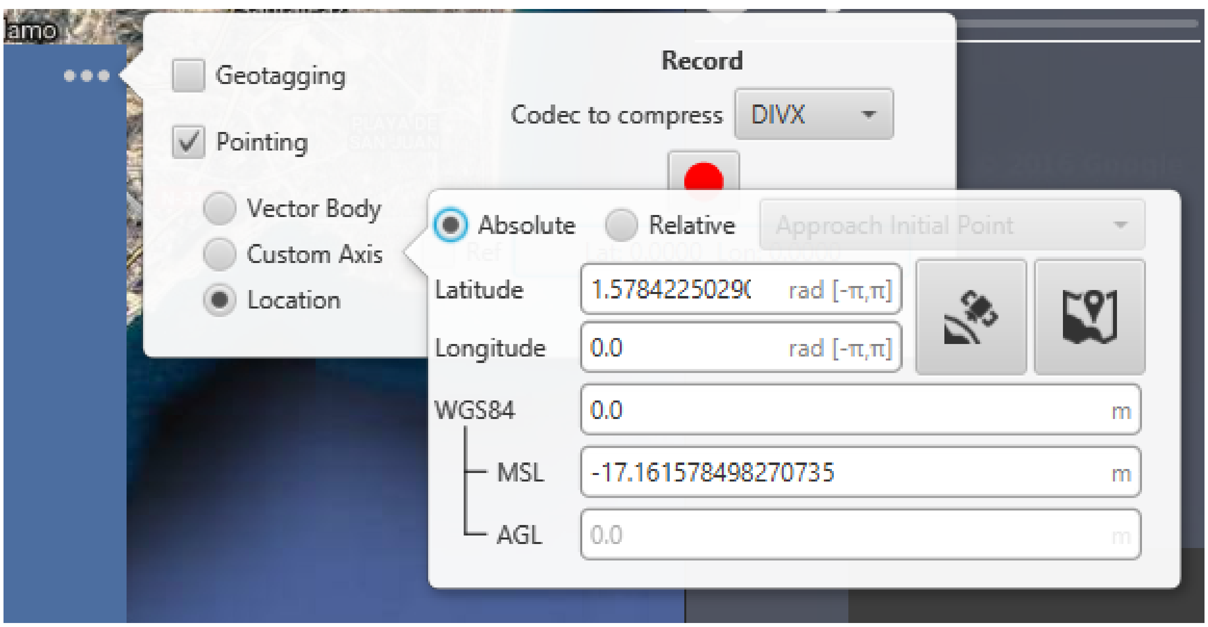
Workspace: Gimbal

Select the Payload as the one in the Veronte Ground Devices configuration.



Workspace: Gimbal widget

After doing it click on ' . . . ' placed on the top right corner of the widget Gimbal.



Gimbal widget: Pointing

In this window it is possible to set where the tracker will point. It can be a absolute position, like the one from the image, or a **Relative position** . For example the **Moving object incompress 2** (Veronte Air in this case).

It is possible to test the correct tracking by moving the Veronte Air (or the platform if it is possible) and checking the tracker correspondent rotation in Pan and Tilt. To perform this test it is useful to maintain the UAV at > 20 [m] from the Ground Station because of the bad transmission link in case of small distances.

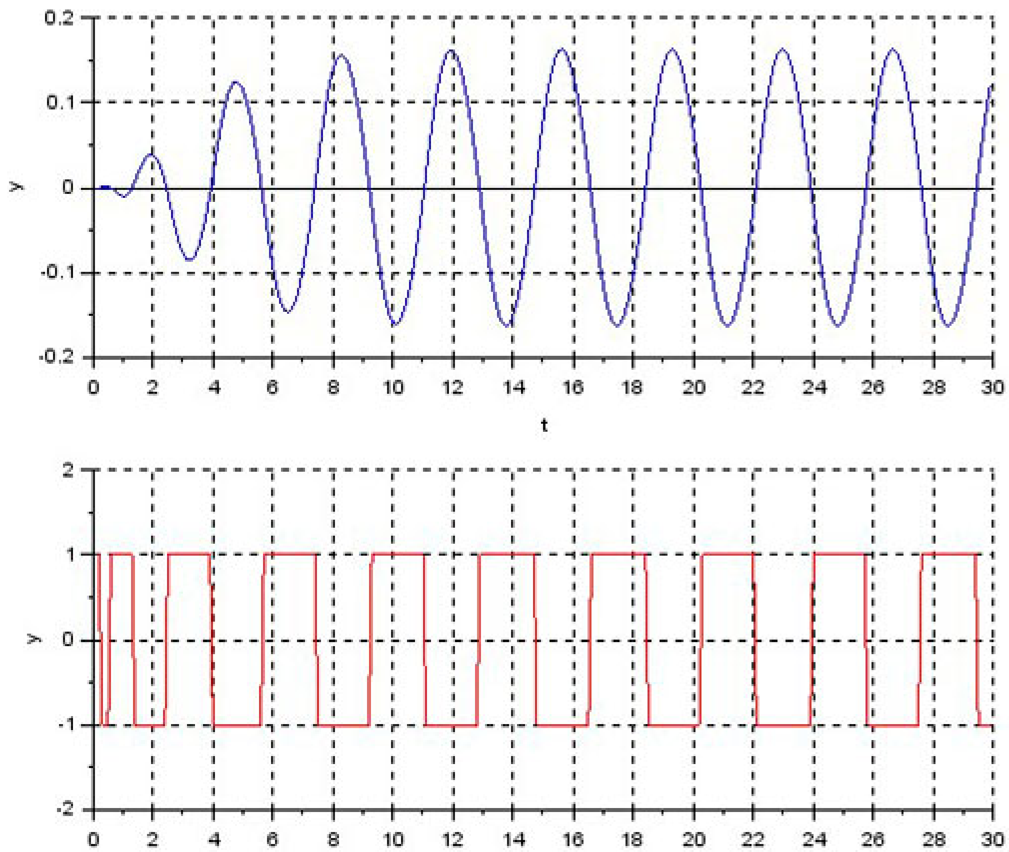
Veronte Tracker

Veronte Tracker is a high performance tracking antenna specifically designed for most demanding applications. The system can install any directional antenna for maximizing system operation capabilities. Embedded control actuators and installed encoders permit to automatically point the antenna with unique precision. Height (Tilt control) and orientation (Pan control) given to the antenna makes the device perfect for long range operations.

Video: Tracking Antenna – Veronte Tracker

11.6 Autotune

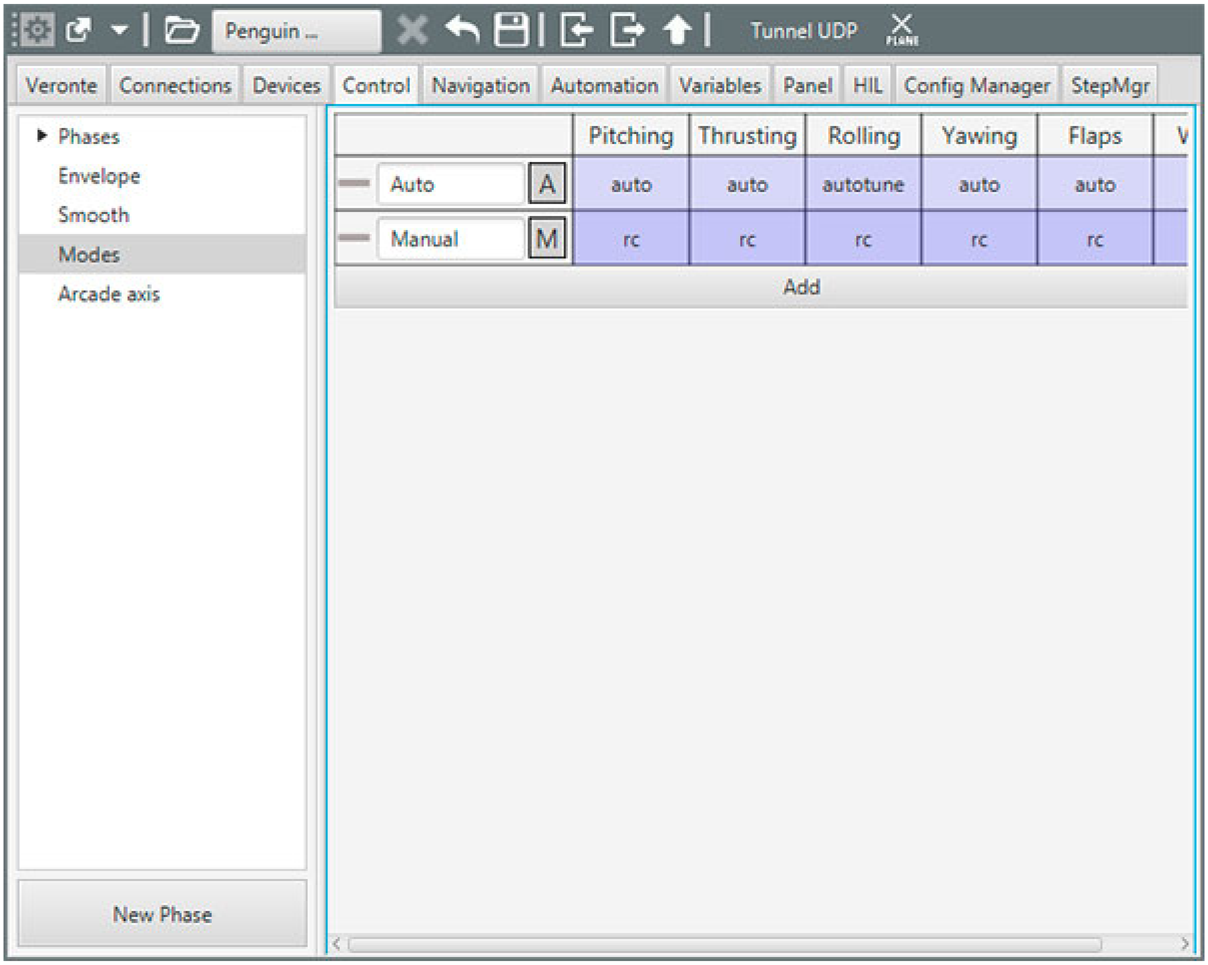
The Autotune allows finding the PID controller gains automatically. During the process, the PID controller is replaced with a relay function and the controller parameters are then determined from the period and the amplitude of the oscillation of the system by using FFTs.



Autotune output (blue) and input (red)

11.6.1 Modes Configuration

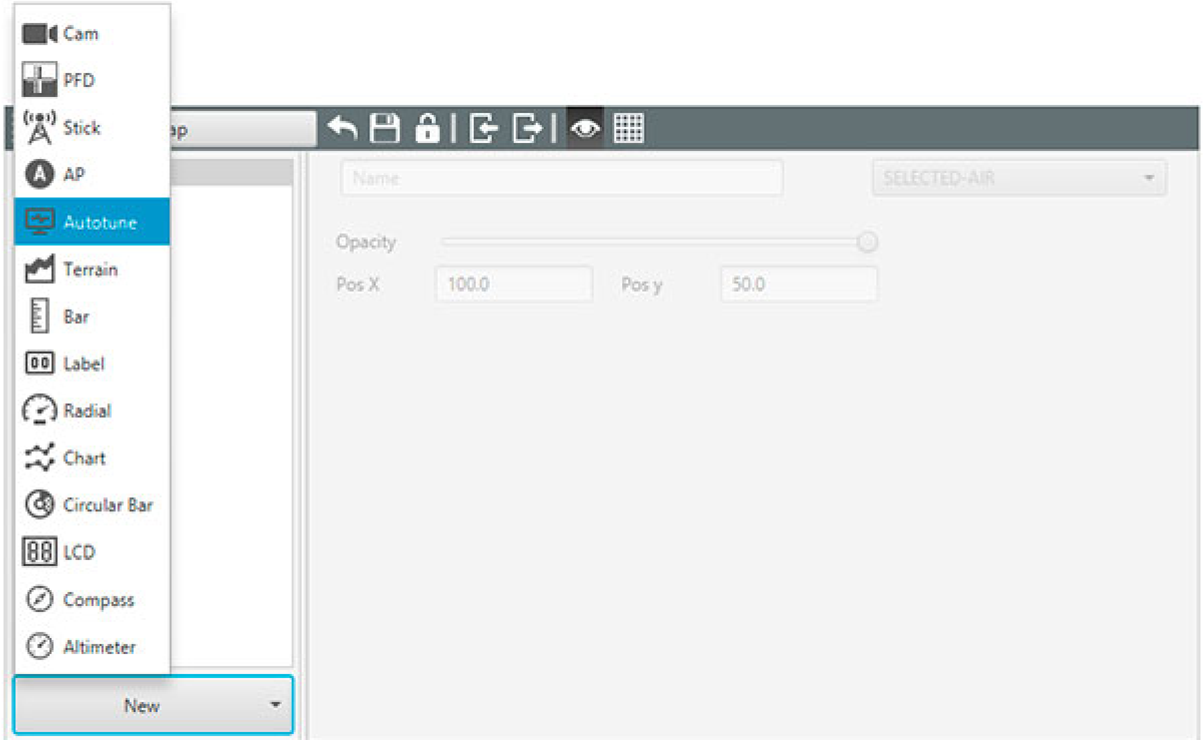
To be able to use the autotuning tool, the Autotune Mode has to be selected in the Modes menu for the control output which must be autotuned. In the following image, for example, for the rolling control is set the Autotune mode.



Autotune mode

11.6.2 Workspace Configuration

In the Workspace panel, it is possible to select the Autotune Tool.



Autotune tool in the Workspace

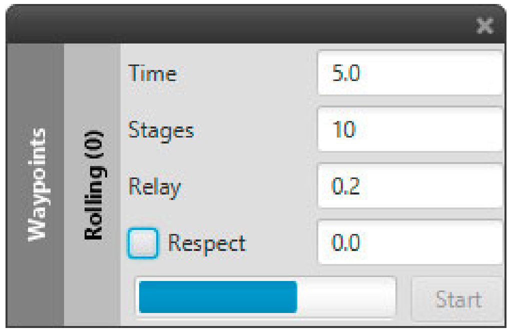
When the window is opened, it is possible to select the loop for the autotuning. If some control loop does not have a PID controller configured, it appears with transparency and it can not be autotuned (red).

Autotune loop selecting

When the loop is selected, the window changes and some parameters are showed and they can be edited:

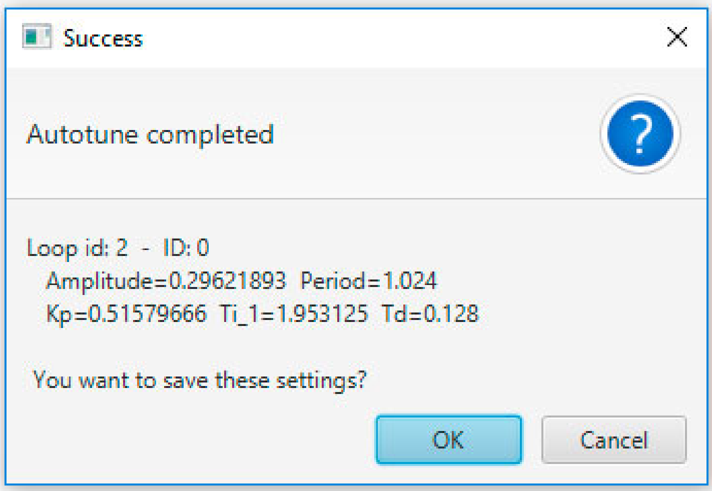
- **Time:** It is the period of time in which the Autotune is performed [s].
- **Stages:** The number of stages of the Fast Fourier Transform (a value between 5 and 10 is allowed).
- **Relay:** This is the amplitude of the Relay function (R). The value has to be chosen in according with the proportional gain in the PID of the autotuned variable.
- **Respect:** A mean value of the variable has to be selected. If the respect is select, the autotune will start from the last value of the variable. If not, the value can be edited by the user and the Autotune Relay function will start from this value and it will go from -R to R.

When all parameters are set, it is possible to click on Start and the autotuning process will begin and the blue bar will move until the end of the process. The left bar of the window allows to check the flight Phase and the selected loop.



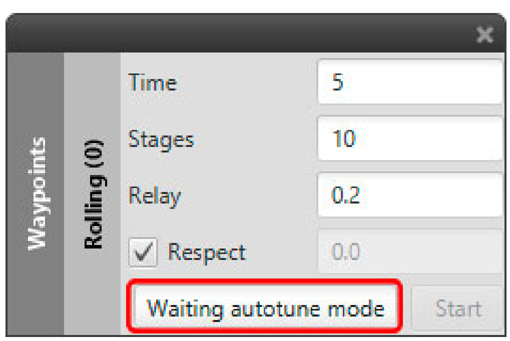
Autotune process

If the process is performed successfully, a new window will show up showing the PID gains found in the Autotune process and the output wave amplitude and period. From this window, it is possible to save the new values in the configuration or cancel them.



Autotune completed

Warning: Always be sure the gain values are in the correct range. A bad value, if saved in the configuration, could cause a control loss of the platform.



Process error

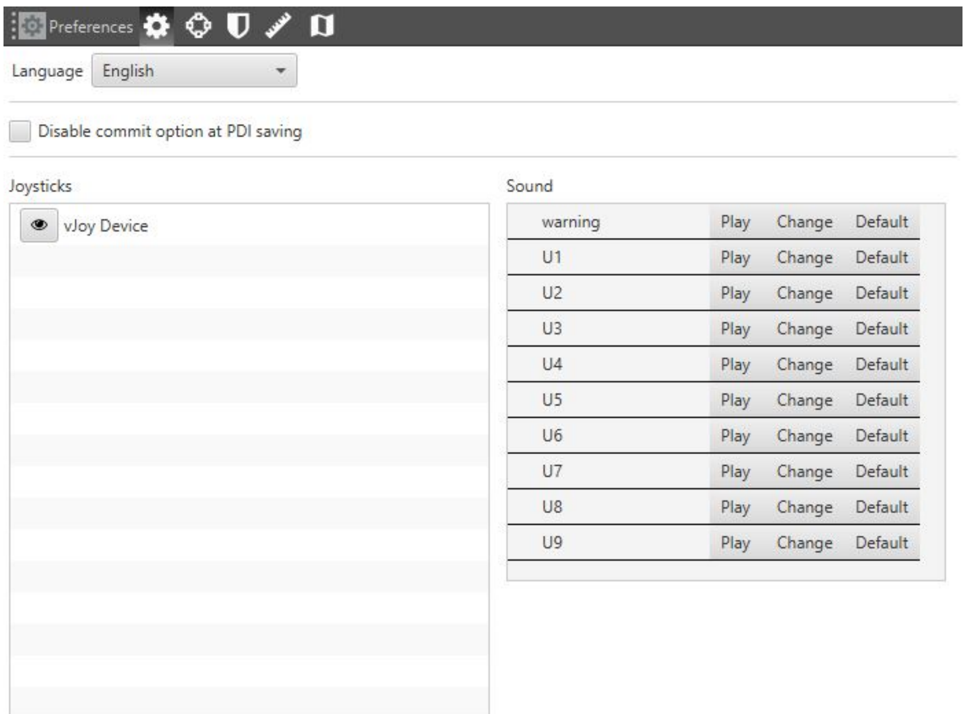
When the Autotune is started, it is possible to see a message in the blue bar: **"Waiting for autotune mode"**. In this case, the autotune mode for the selected control is not correctly set. The user must change it in the control panel.

11.7 USB joystick integration

11.7.1 Adding a joystick

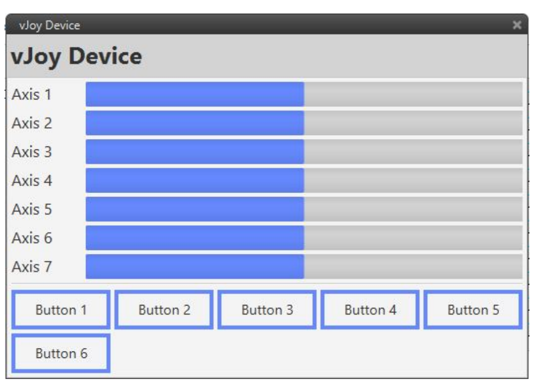
Veronte Pipe is able to detect USB devices such as joysticks. The buttons and axis of these devices can be read and configured to send stick information to Veronte Autopilot.

1. Connect your USB device to the computer and open Veronte Pipe.
2. Go to **Preferences/General** in Veronte Pipe. The USB device will be automatically detected and displayed.



Joystick device

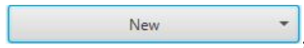
Clicking  the readings of the USB device channels will be visualized.



Usb device channels

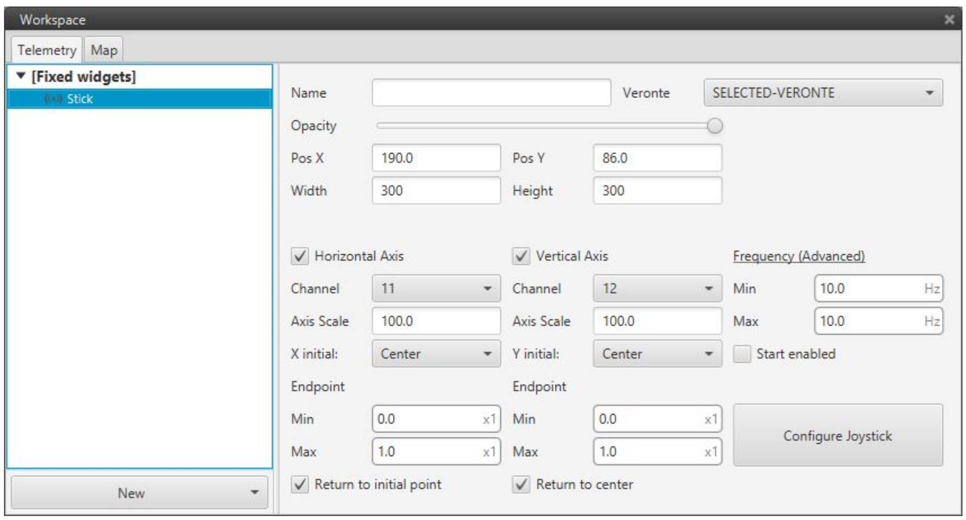
11.7.2 Defining a virtual stick

Once the usb device is recognized:

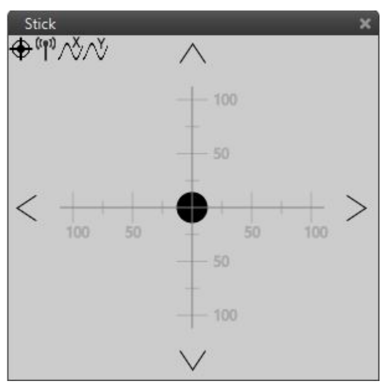
1. Add a virtual Stick: On the workspace window, select 

, 

and then stick.



Josytick workspace menu



Josytick widget

2. Assign a USB device channel to a stick input variable.

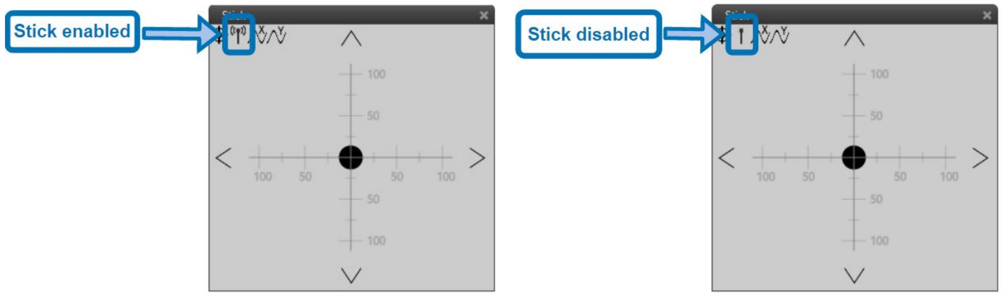
   1. For each axis specify which channel will be considered by choosing one of the channels listed.

   2. Select "Configure joystick" and follow the instructions.

3. Verify that the movement of the virtual stick is associated with the movement of the USB device.

4. Check that the stick input variables are changing with the movement of the virtual stick. i.e for channel 1, stick input r1.

Note: The virtual stick will be active by clicking on the antenna button.



Virtual stick state

Note: It is recommended disabling the axis that are not used.

Warning: A virtual stick can command on the same channels as a physical stick box. Make sure that the channels are not interfering to each other.

In this section, a series of different examples will be presented in order to improve the understanding that the final user will have of Veronte Pipe.

The different topics cover in this section are:

- **Configurations:** it will be explained how to configure some different platforms in Veronte Pipe, starting with the servo trimming process and the creation of the mission phases of a simple flight operation.
- **Automations:** this section will be focused on the creation of the common automations that are used for operating a UAV. Presenting this ones will make it easier for the user to develop its own automations for more specific operations.
- **Stick configuration:** this section shows how to configure completely a new joystick.
- **Lidar integration:** in the lidar section is explained how to set the settings of the Altimeter using the analogic port.
- **Autotune:** the complete example of the autotune configuration and operation.
- **4G Communication:** Steps to configure 4G communication with Veronte Autopilots.

CHAPTER

TWELVE

TROUBLESHOOTING

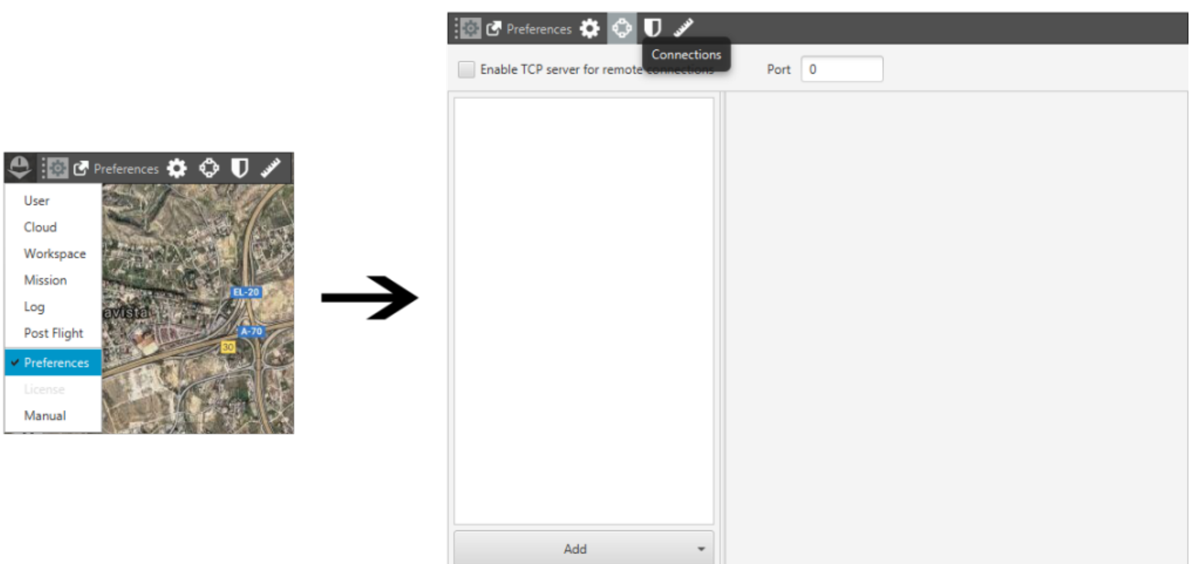
12.1 Veronte not detected

If a Veronte Unit is not detected by Veronte Pipe:

- Check if the unit it is correctly connected to the PC.
- Check if the connection is configured in **PreferencesConnections**.

If the connection it is not configured:

1. Go to **Preferences > Connections**
2. Select **Add** and then, if the device is conected using the USB port select **Serial COM**.
3. When the connection is completely configured, it is necessary to click on **Save Connection** in order to make it effective.



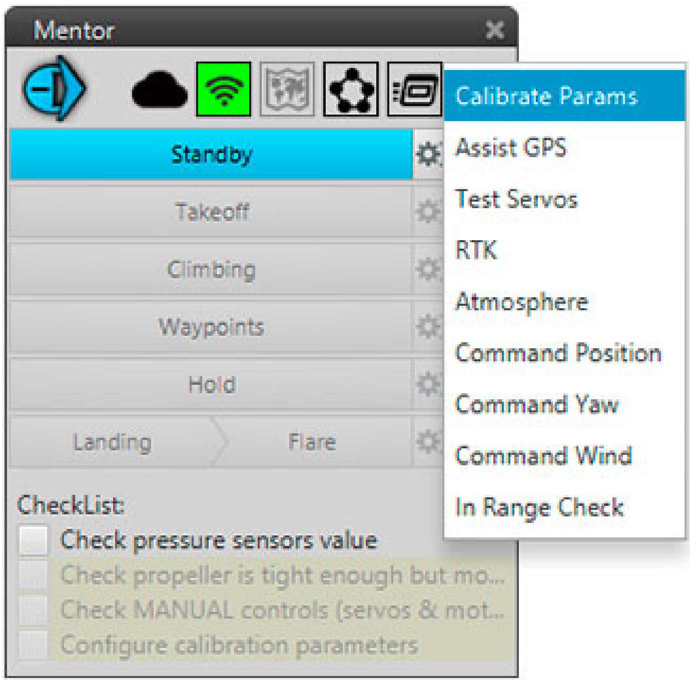
Preferences Panel

If the problem persists, maybe there is a problem with the configuration file. Try accessing the safe mode and upload another configuration file into the Veronte Unit. For more information see section *License & Safe Mode*.

12.2 PFD Attitude

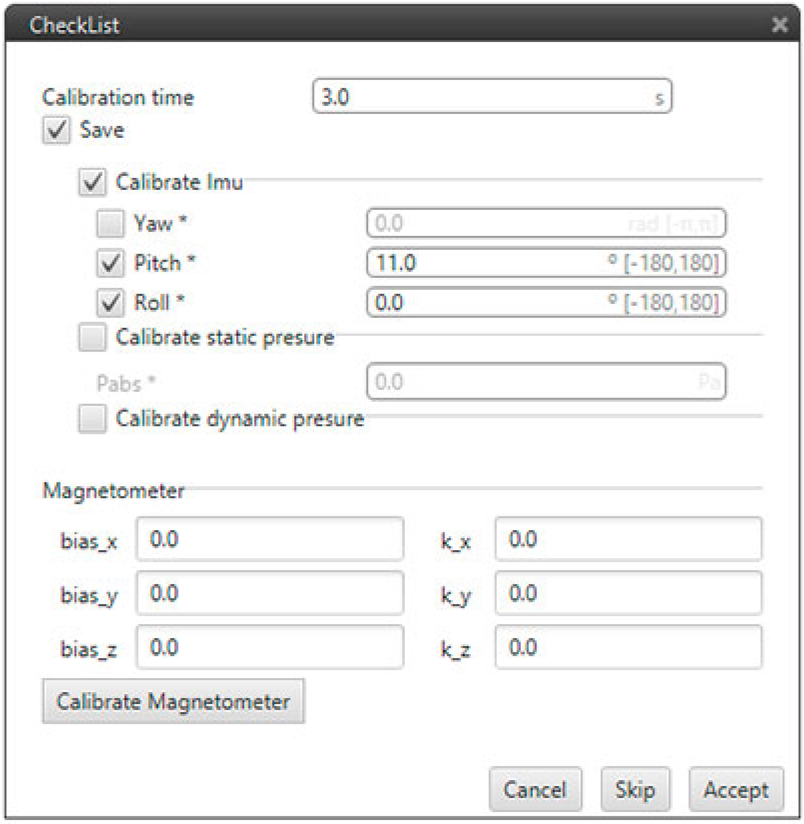
When the PFD not displays the correct attitude, there may be an error with the calibration of the inertial measurement unit (IMU). To solve this issue follow the next steps:

1. Go to Verote Panel.
2. Click on Calibrate Params.



Calibrate Params

3. A new window will be displayed, here user has to procede to calibrate the IMU, see section *Sensors Calibration* for more information about this process.



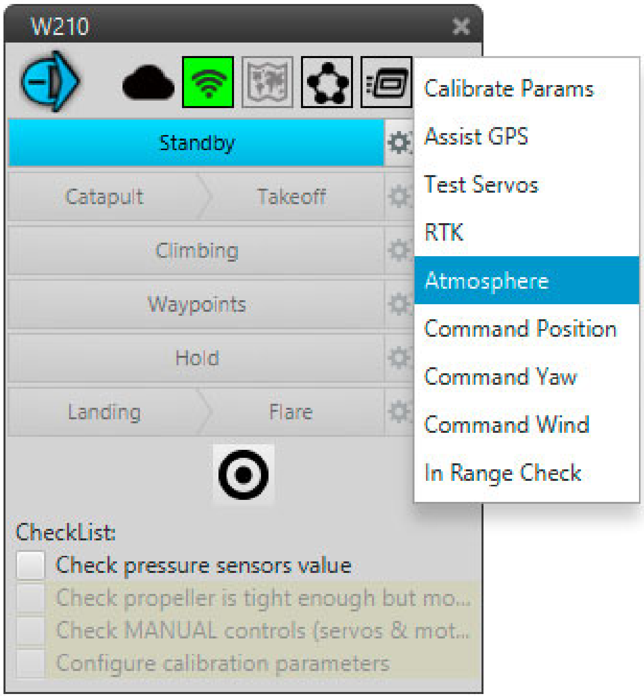
IMU Calibration

12.3 Altitude Error

When the AGL lectures obtained from telemetry are showing strange results, like a very great value when the UAV is in the ground or even negative ones, the problem can come from different areas and the possible solution will be presented in this section.

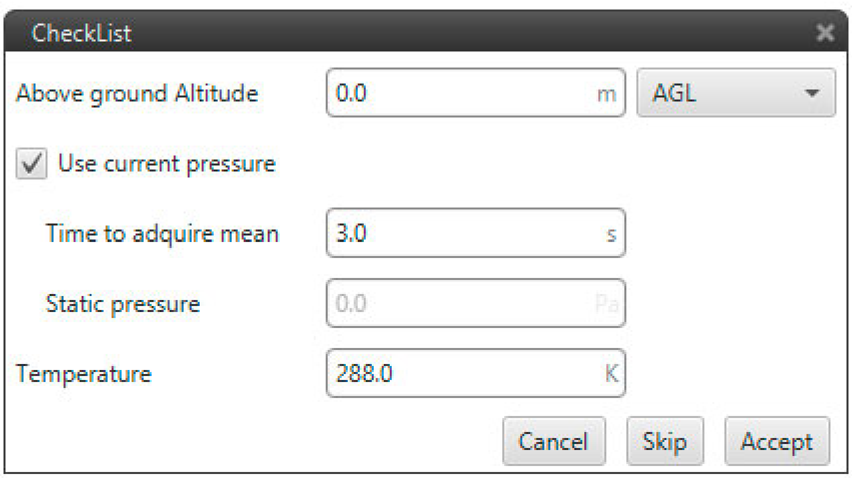
12.3.1 Atmosphere Calibration

The atmosphere has to be calibrated according to the current temperature. The calibration options are open directly on Veronte Panel.



Calibration Menu – Atmosphere

The pop-up window contains the atmosphere parameters, which should be varied to match the actual ones.

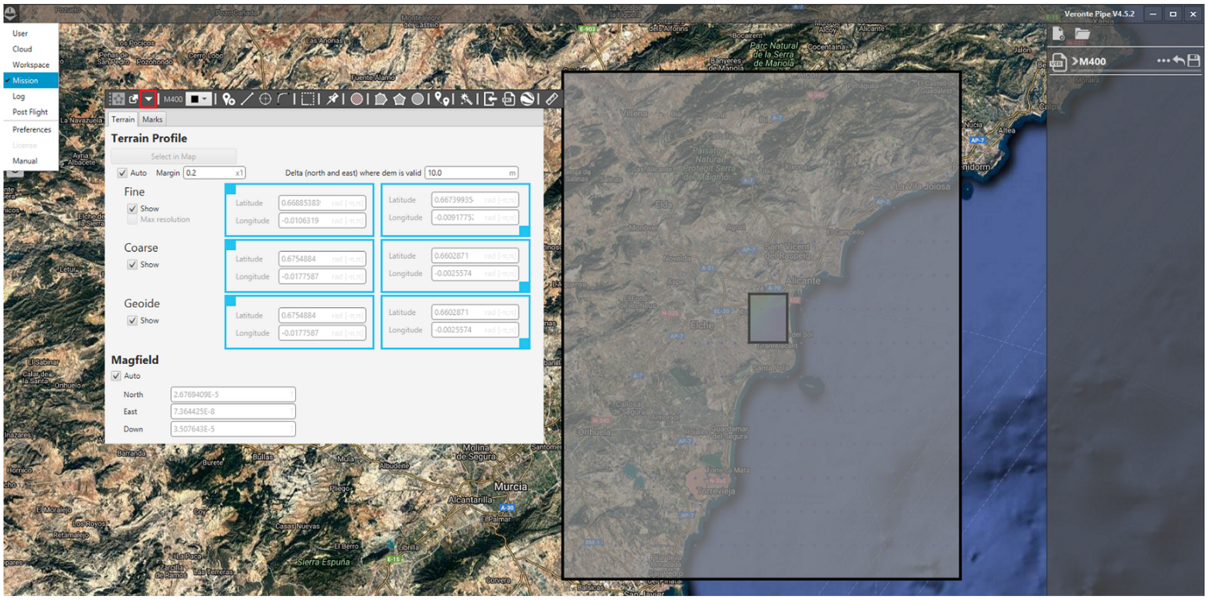


Atmosphere Calibration Window

12.3.2 Terrain Altitude Meshes

Another source of error could be the terrain model, which contains the information about the heights in the mission area. It could happen that the mesh with this data is not located over the mission trajectory, so when having an error in the height measures this option should be checked.

Meshes are displayed in **Mission/Terrain Profile**, clicking the option show.



Terrain Height Meshes

- **Show:** this option makes the mesh visible. They can be moved directly by dragging them, so they have to be located over the mission waypoints.
- **Auto:** to avoid a future error, select this option. When this is enabled, meshes are moved automatically to the mission area.

12.4 HIL Simulation Fails

When HIL Simulation does not work properly, is possible that:

- Simulation frequency is not well configured. In this case, check the XPlane and VerontePipe frequency as described in section *Professional HIL* of the manual.
- The aircraft in XPlane is load in a different airport, so be sure of loading it in the correct one.
- Terrain profile is not correctly positioned: Coarse and Fine zones must be positioned on the mission to have the correct terrain information (see section Mission Setup).
- The radio connection is down. Check the radio link (Antenna, Cable).
- Reboot the whole system: Veronte Pipe, Veronte Autopilot and PC.

12.5 Telemetry Overflow Error

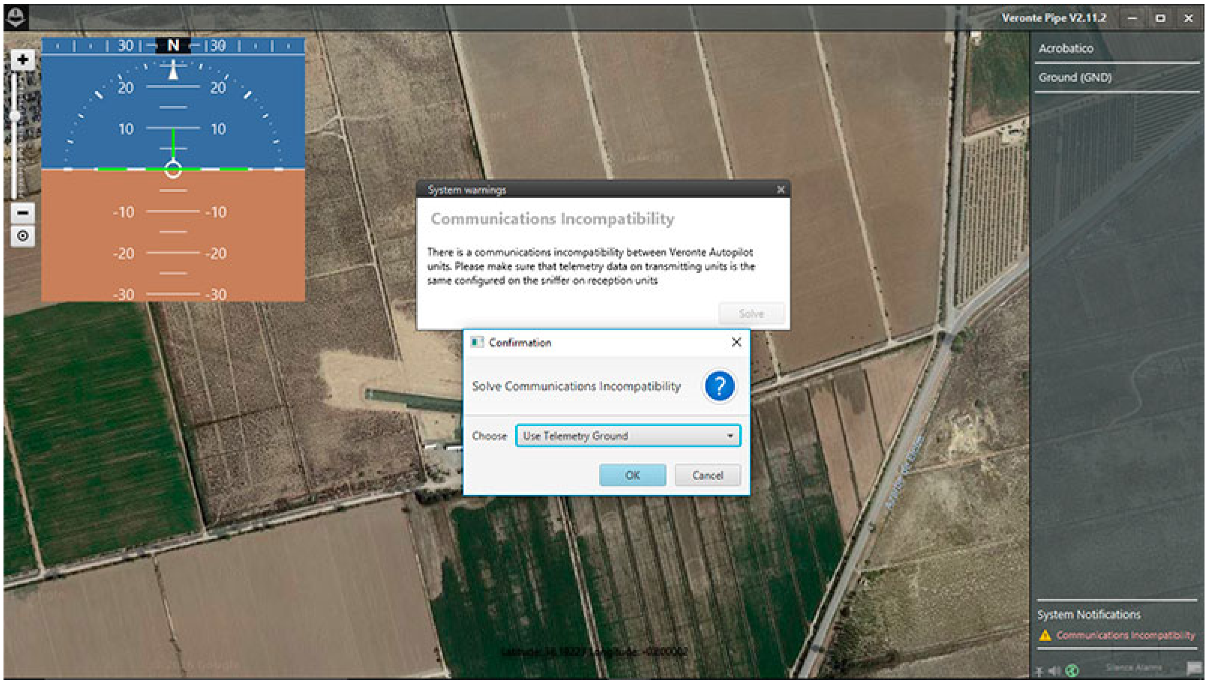
The telemetry Data Link between Veronte Units (GND-AIR) has a limit on the number of variables that it can transmit. If this number is exceeded, the system will not transmit the last ones added and an error will appear on the sidebar (below the autopilot that has this error).

Telemetry Overflow error

To solve this problem, delete from the Data Link some variables until the error disappears (visit section *Telemetry* to check how to deal with the telemetry on Veronte Pipe). This problem is not critical to the flight. The operation can be carried out, with the only difference that the last variables added to the Link can not be displayed on the workspace.

12.6 Communications Incompatibility

This error appears on the lower part of the sidebar. It means that the variables sent by an autopilot to the other one are not the same as the ones sniffed ("captures") by the one who is receiving the telemetry.

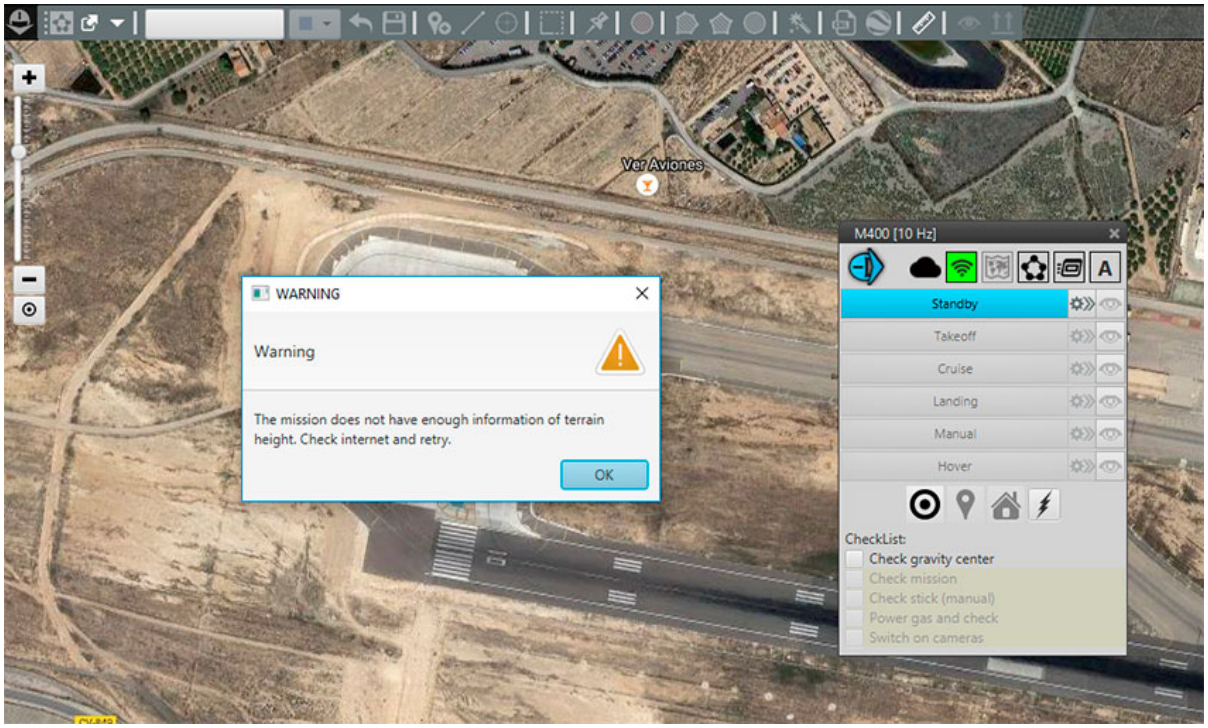


Communications Incompatibility

The software provides a dialogue which solves this problem by selecting which one of the options (telemetry or sniffer) will "take control" over the other one making it having the same variables. Check the sniffer and telemetry link before the operation to make sure that the variables sent and listen are the correct ones (section *Variables*).

12.7 No enough information of terrain height

When the user tries to save a new mission or an edited one, it is possible that a Warning message appears: **"The mission does not have enough information of terrain height. Check the internet and retry".**



Warning terrain height

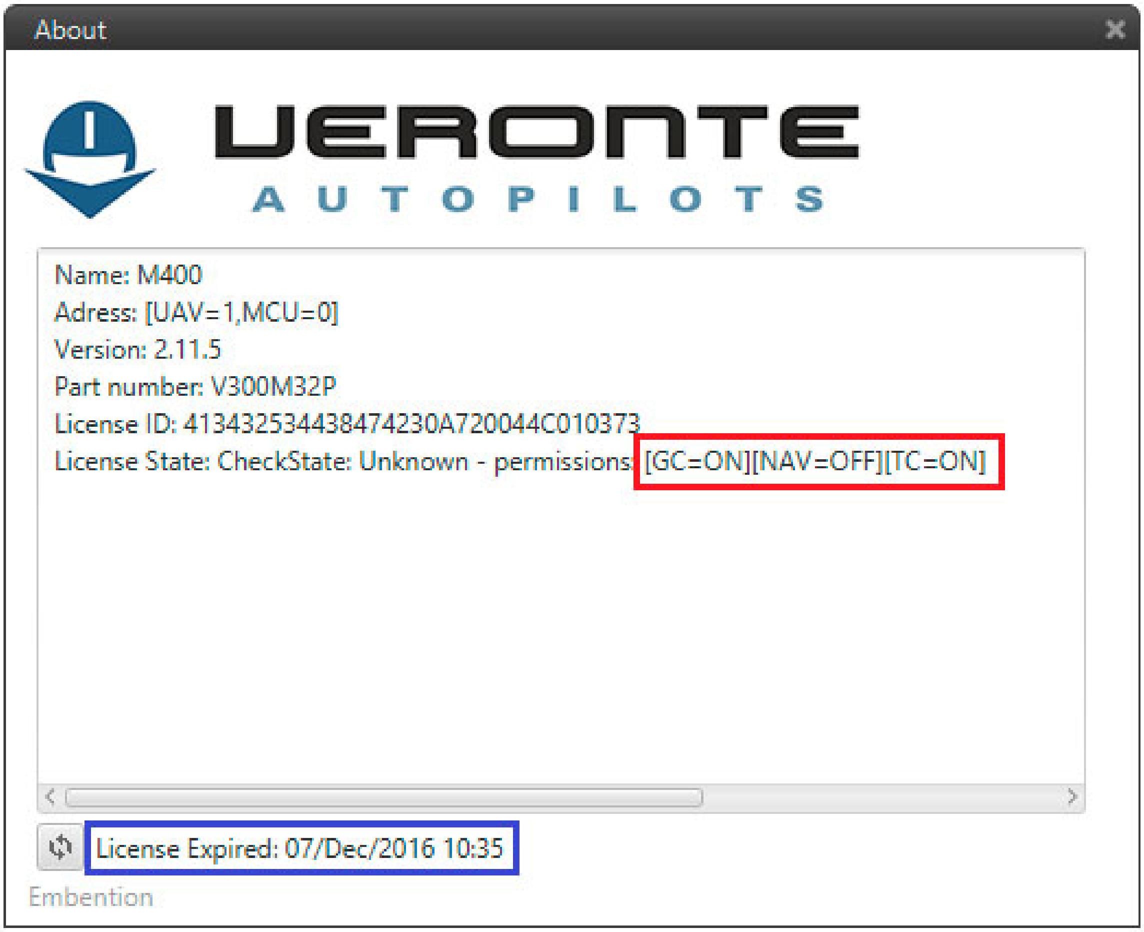
It can be solved in two different ways:

- The problem is the internet connection, check it and re-try.
- The mission has been saved with no waypoints or routes. In this case, the system does not have information about mission location and it is not able to update terrain information.

12.8 License Expired

When Veronte Autopilots start, the license state is always checked. As described in section *License & Safe Mode*, the Veronte license has to be renovated biweekly.

If the license warning is displayed in the right bar (below the autopilot name) as showed in the following figure, it is sufficient a left-click on it and the license panel will show up. The panel is useful to check the following information:



License panel

1. License State (red)
2. License Expiration Date and Time (blue)

In order to solve the problem, user must check:

- **Internet connection:** to renovate the license it is necessary an internet connection.
- **GPS connection:** if during the license checking the GPS connection is down, the second check (NAV) will be kept to OFF and the license warning will not disappear.
- **PC date and time:** check the date and time settings of the PC. If the date is not correct the system could not be able to check or renovate the license.

CHAPTER

THIRTEEN

ACRONYMS & DEFINITIONS

13.1 Acronyms

16 VAR	16 Bits variables (Integers)
32 VAR	32 Bits variables (Reals)
ADC	Analog to Digital Converter
AGL	Above Ground Level
AoA	Angle of Attack
ARC	Arcade Mode
AUTO	Automatic Mode
BIT	Bit Variables
CAN	Controller Area Network
CAP	Capture Module
CMB	Climb Phase
CRU	Cruise Phase
DC	Direct Current
DGPS	Differential GPS
ECAP	Enhanced CAP
ECEF	Earth Centered – Earth Fixed
EGNOS	European Geostationary Navigation Overlay Service
EKF	Extended Kalman Filter
FCS	Flight Control System
FHSS	Frequency Hopping Spread Spectrum
FLR	Flare Phase
FTS	Flight Termination System
GIS	Geographical Information System
GND	Ground
GNSS	Global Navigation Satellite Systems
GPIO	General Purpose Input Output
GPS	Global Positioning System
GS	Ground Speed
GS	Ground Segment
HLD	Hold Phase
HUM	Hardware User Manual
I2C	Inter-Integrated Circuit
IAS	Indicated Air Speed
ID	Identification
ISM	Industrial Scientific and Medical

continues on next page

Table 1 – continued from previous page

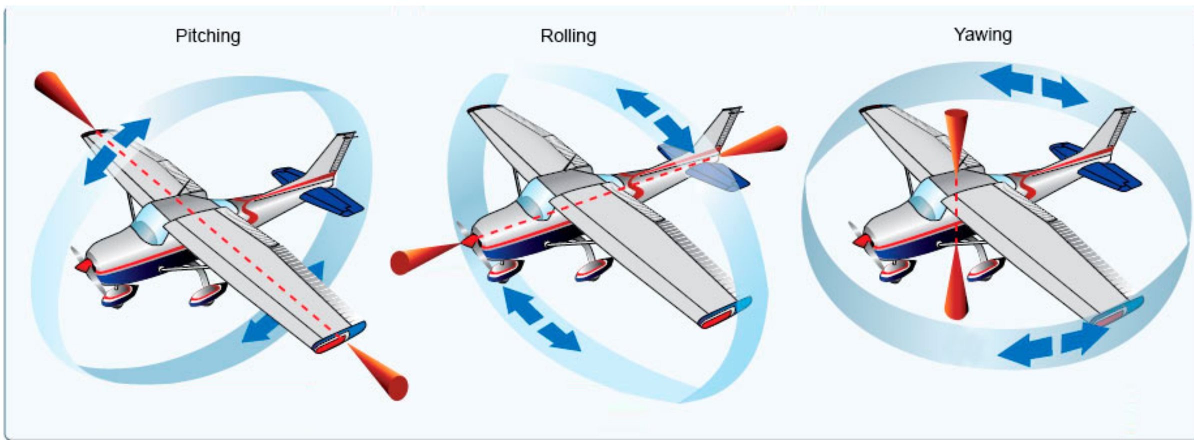
LED	Light-Emitting Diode
LND	Landing Phase
MSL	Mean Sea Level
PFD	Primary Flight Display
PID	Proportional Integral Derivative
PWM	Pulse Width Modulation
QNH	Barometric atmospheric pressure adjusted to sea level
RC	Radio Control Mode
RF	Radio Frequency
RPAS	Remotely Piloted Aircraft System
RPM	Revolutions Per Minute
RS 232	Recommended Standard 232
RS 485	Recommended Standard 485
RX	Reception
SMA	SubMiniature Version A Connector
SSMA	Miniature-Sized Connector
STB	Standby Phase
SU	Servo-Output matrix
SUM	Software User Manual
TAS	True Air Speed
TKO	TakeOff Phase
TX	Transmission
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
US	Output-Servo matrix
VTOL	Vertical TakeOff and Landing
WGS 84	World Geodetic System 84
WP	Waypoint

13.2 Definitions

- **Control Phase:** The operation is divided into phases in which the UAV has a specific performance. Each of this phases is called a control phase.
- **Control Channel:** It is each of the signals used to control a behaviour or action.
- **Control Mode:** It is possible to make a manual control of the UAV by stick, assisted control and fully automatic control.
- **Actuator:** It is a mechanic device to provide force to move or “act” another mechanical device.

AXES CONVENTION

All signs in the system are managed according to the international aeronautical axes convention: It is considered positive any deflection that generates positive rotational forces or moments about the aerodynamic centre of the aircraft, except for “y” axis (elevator) where it is considered negative a positive moment.



Sign Convention

Example, an elevator down position will generate a positive pitch so the elevator is considered positive on down position. Main actuators rules:

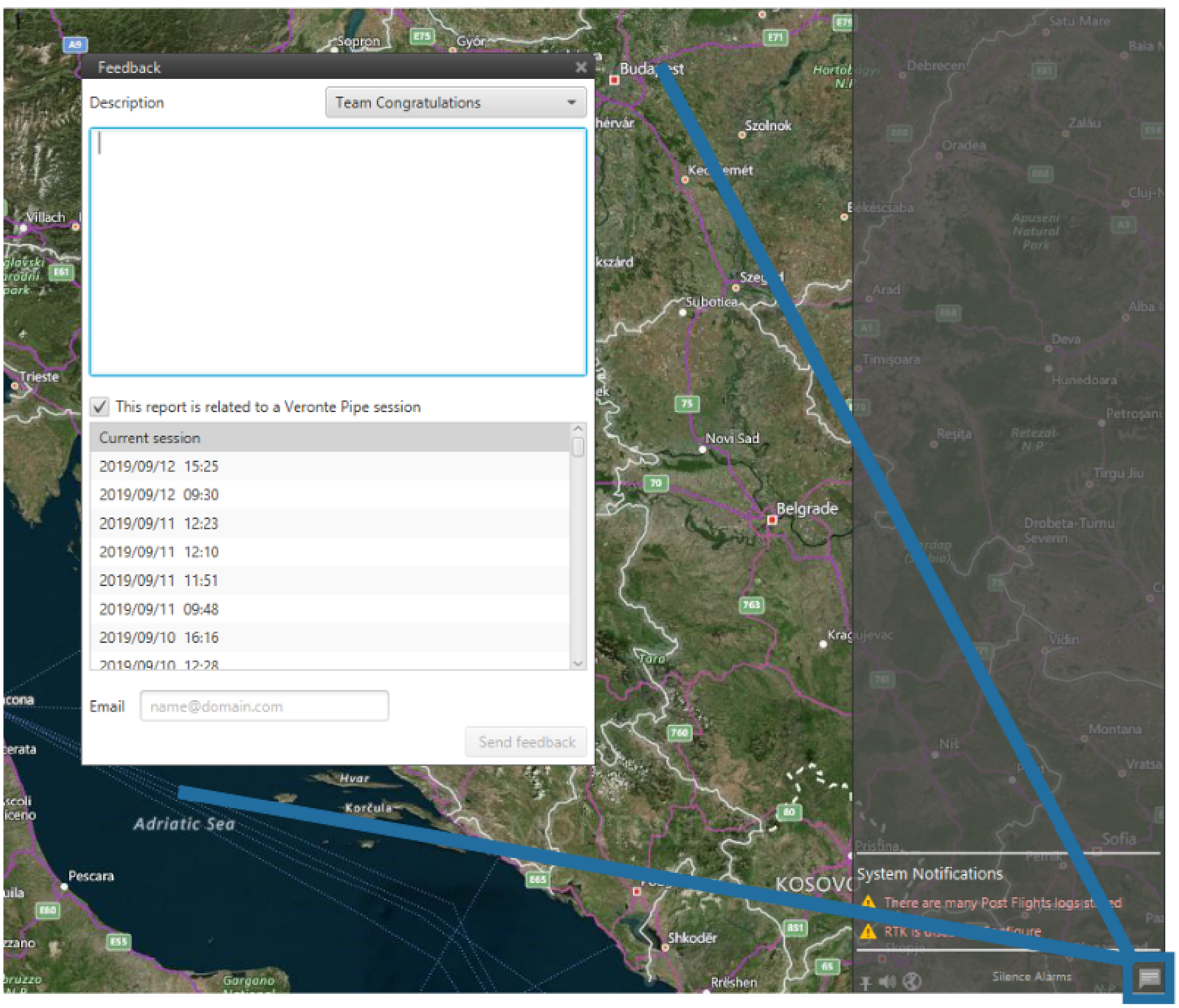
Actuator	Positive	Negative
Elevator	Down	Up
Rudder	Right	Left
Right Aileron	Up	Down
Left Aileron	Down	Up
Tail Rotor	Right	Left

CHAPTER FIFTEEN

FEEDBACK

15.1 Feedback Panel

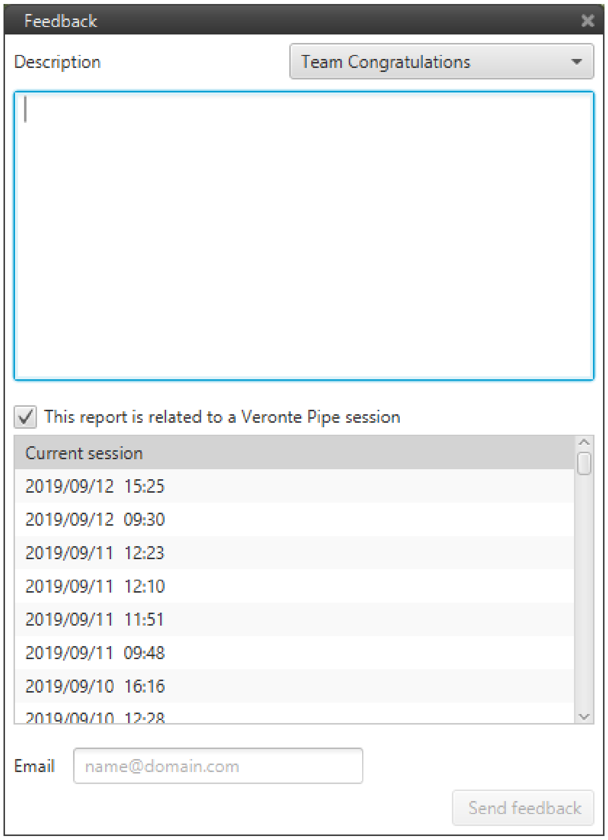
If you'd like to provide us with feedback on future releases, issue report or even team congratulations, please provide us with all possible details about your Veronte experience and we will be happy to listen and to help you.



Feedback Panel

We have created the tool shown in the previous figure in order to allow the user to directly give us feedback about:

- **Team Congratulations:** Give us your positive feedback about your Veronte experience.
- **Suggestion for Next Releases:** All your comments and suggestions would be greatly appreciated.
- **Configuration Support:** Your feedback will be sent directly to our configuration team. They will help you with all your configuration troubles.
- **Failure Report:** It would be extremely useful to help our developing team reaching all possible information about Veronte System bugs or failures.



Feedback Panel Options

Once the feedback type is selected, the user can add a **Description** to detail the content. Furthermore, the report can be sent adding the reference to the **Veronte Pipe session**, which could be the actual one or the one selected from the list. If the user is interested in having an answer directly to his e-mail address, an **Email** can be inserted in the corresponding field.

When the window is entirely completed and an internet connection is available, it is necessary to click on Send feedback and the report will be sent.

CHAPTER SIXTEEN

4X REDUNDANT AUTOPILOT

4x Redundant Autopilot

16.1 Introducction

4xVeronte Autopilot is a triple redundant version of the Veronte Autopilot. It includes three complete Veronte Autopilot modules together with a dissimilar arbiter for detecting system failures and selecting the module in charge of the control. The autopilot selected as the master will be the one controlling the actuators and communicating with the payloads.

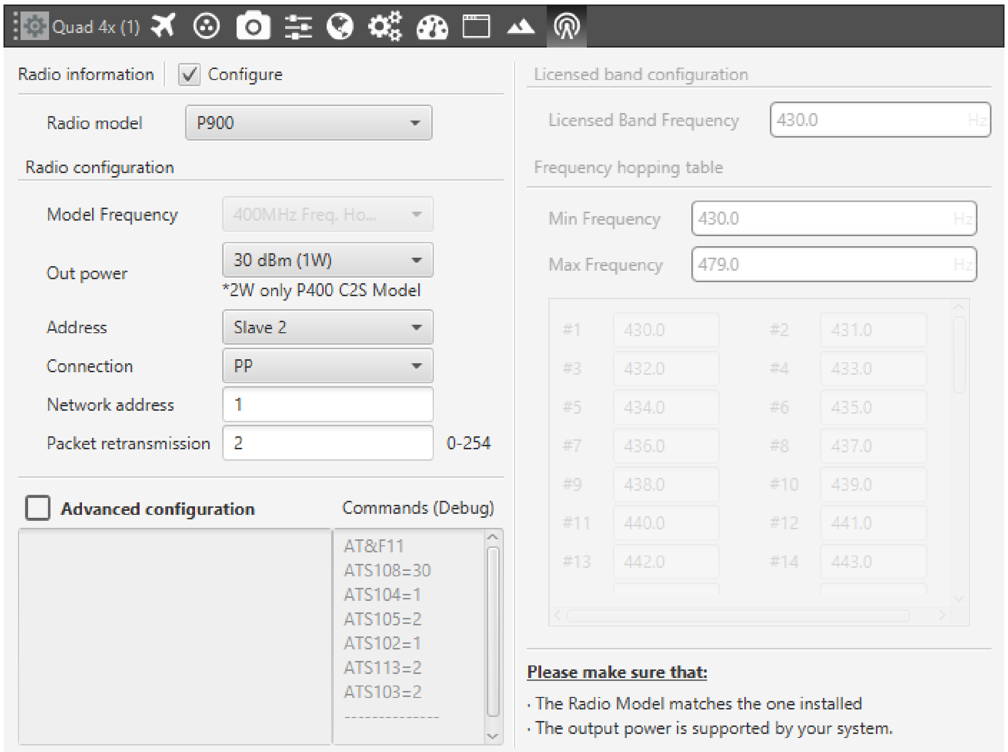
Each Veronte autopilot contains all the electronics and sensors in order to properly execute all the functions needed to control the UAV. Veronte executes in real time all the guidance, navigation and control algorithms for the carrying airframe, acting on the control surfaces and propulsion system and processing the signals from different sensors: accelerometers, gyroscopes, magnetometer, static pressure, dynamic pressure, GNSS and external sensors.

All three modules are managed by a dissimilar microprocessor. This arbiter includes voting algorithms for managing the module in charge of vehicle control. This microprocessor compares data from all modules in real time and processes it for discarding any autopilot module showing an undesired performance.

16.2 Setup

16.2.1 Radio

Each one of the Vereonte Autopilots modules inside 4xVeronte Autopilot has it own radio module featuring frequency hopping the best possible communication. For each one of the module the user can set the configuration from veronte Pipe.



Radio menu in Veronte Pipe

Note: For each particular case the configuration of the radio may vary. Common radio configurations are PP for complet redundant communicatoion or PMP for one 4xVeronte Autopilot and one Ground Autopilot

16.2.2 CAN bus

4xVeronte Autopilot includes three complete Veronte Autopilot modules that have to be configured independtly in order to work porperly. 4xVeronte Autopilot is compatible with all configuration files from **Veronte Autopilot**. The communication channel between Veronte Autopilots modules and arbitrer is done by using CAN bus so in order to start arbitring, Veronte Autopilots modules have to send the following information.

Configuration	Value
Baudrate	1 Mbps
Endianess	Little endian

Each Veronte shall send using its CAN-TX ID and in little endian format.

16.2.2.1 Variable value message:

Byte0	0	Telemetry message
Byte1	variable id	Variable id sent (0 to 127)
Byte2-5	variable value	Variable value as Float (32 bits)

16.2.2.2 Status message:

Byte0	0	Telemetry message
Byte1	0xFF	Status flag
Byte3	bit0	1:Ready, 0:Not ready
	bit7-1	Reserved

By default the arbiter will send the following CAN messages.

16.2.2.3 Variable value message:

Byte0	0	Telemetry message
Byte1	N, Autopilot id	Autopilot [0, 3]
Byte2-5	value	Autopilot score as Float(32 bits)

16.2.2.4 Status message:

Byte0	0	Telemetry message
Byte1	0xFF	Status flag
Byte3	bit6-0	Selected autopilot
	bit7	Arbitrating 0:false, 1:true
Byte4	bit0	AP0 Alive
	bit1	AP1 Alive
	bit2	AP2 Alive
	bit3	AP3 Alive (external)
	bit4	AP0 Ready
	bit5	AP1 Ready
	bit6	AP2 Ready
	bit7	AP3 Ready (external)
Byte5	bit0	system ok
	bit1	start
	bit2	pdi
	bit3	memory alloc
	bit4	cana bus on
	bit5	canb bus on
	bit6	Reserved
	bit7	Reserved
Byte6	bit0	system ok
	bit1	Vcc A
	bit2	Vcc B
	bit3	Vcc arb
	bit4	Vcc1
	bit5	Vcc2
	bit6	Vcc3
	bit7	Reserved

16.2.3 Control

One modification has to be implemented in order to have smooth transition between Veronte Autopilots modules. This modification involves sending the control outputs of Veronte Autopilots modules to the rest of autopilots so it is considered in the transition. The user can refer to the configuration examples for this implementation.

Veronte autopilot

Veronte Autopilot is a miniaturized high reliability avionics system for advanced control of **unmanned systems.** This control system embeds a state-of-the-art suite of sensors and processors together with LOS and BLOS M2M datalink radio, all with reduced size and weight.

4X Veronte Autopilot is a **triple redundant** version of **Veronte Autopilot**. It includes three complete Veronte Autopilot modules fully integrated with a dissimilar arbiter for detecting system failures and selecting the module in charge of the control.

Veronte Pipe is the software designed for operating any Veronte powered platform. Users achieve a combination of easy-to-use application, real-time response and, firstly, safe operations.

The sections that follow this introduction explain in detail all the features of Veronte Pipe, the software used to control, configure and operate the platform fitted with the autopilot. Besides, the reader will find in *Hardware Installation* information about how to install both Veronte Autopilots in the aircraft (or the selected type of vehicle).