# 1x Software Manual

**Release 7.8/1.3**

Embention Sistemas Inteligentes, S.A.

2026-09-15

# Contents


Scope of Changes ..... 6
Software Applications ..... 7
- Veronte Link ..... 7
- 1x PDI Builder ..... 7
- Veronte Ops ..... 7
- Veronte HIL ..... 8
- Veronte Updater ..... 8
- 1x PDI Calibration ..... 9
- Veronte FDR ..... 9
- Veronte VSA ..... 9
- 1x PDI Tuning ..... 9

Nomenclature ..... 10
- Reference Directions ..... 10
- Axes ..... 12

Core architecture ..... 13
- Task distribution ..... 13
- Monitoring variables ..... 15

Data Transmission ..... 19
Lists of interest ..... 21
- Activation System Error bits ..... 21
- Status Message variables ..... 22
- Extended Status Message variables ..... 23
- List of Addresses ..... 26
- Lists of Variables ..... 31
  - BIT Variables ..... 32
  - Real Variables (RVar) - 32 Bits ..... 78
  - Integer Variables (UVar) - 16 Bits ..... 182
  - Features Variables - 64 Bits ..... 223
- Navigation Variables ..... 235
- List of PDI Errors ..... 240
- List of File System Errors ..... 271

Firmware Changelog ..... 273
- 7.8.18 ..... 273

7.8.32 ..... 274
7.8.38 ..... 276
7.8.50 ..... 276
7.8.66 ..... 278
7.8.80 ..... 279
Step by step ..... 282
VTOL ..... 283
Aircraft specifications ..... 284
1x Air configuration ..... 287
1x PDI Builder configuration ..... 288
Name & Attitude ..... 292
Unit name ..... 292
Attitude ..... 293
Actuators & SU Matrix ..... 294
GPIO to PWM connection ..... 294
Actuators ..... 295
SU Matrix ..... 301
Modes & Phases ..... 306
Modes definition ..... 306
Phases definition ..... 308
Block Programs ..... 309
Navigation ..... 311
Guidance ..... 321
Stick ..... 346
Control Mix ..... 347
Pitching ..... 350
Rolling ..... 355
Yawing ..... 363
Thrust (Vertical) ..... 366
Throttle (Horizontal) ..... 370
Euler to Body Axis ..... 374
Rates to control ..... 376
Control to servo ..... 383
RPM XPlane ..... 387
Compute RTH ..... 387

RTH distance OK ..... 391
Automations ..... 392
Calibration ..... 393
Home ..... 394
Navigation ..... 394
Integral Enable ..... 395
Failsafe ..... 395
Stick ..... 396
Transitions ..... 397
Phase Buttons ..... 399
Transitions table ..... 400
Sensors ..... 401
Accelerometer ..... 402
Gyroscope ..... 405
Magnetometer ..... 409
Dynamic Pressure ..... 409
Static Pressure ..... 410
HIL Configuration ..... 411
Checklist ..... 414
Operation ..... 415
Operation 1 ..... 418
Mission ..... 418
Start Route ..... 422
Runway ..... 423
Home Point ..... 425
Operation 2 ..... 426
Area 1 ..... 426
Area 2 ..... 430
Operation 3 ..... 430
Missions ..... 431
Mark A ..... 431
Mark B ..... 435
Workspace ..... 438
Map ..... 439
Main ..... 439

Displays ..... 441
Charts ..... 443
Flight instruments ..... 445
Simulation ..... 446
Operation 1 ..... 447
Initial and Standby ..... 448
Armed ..... 448
Take-off ..... 449
Hover ..... 449
Cruise ..... 450
Loiter ..... 451
Hold ..... 452
Approach ..... 453
Landing ..... 454
Return to Home ..... 455
Operation 2 ..... 456
Area 2 ..... 457
Area 1 ..... 458
Operation 3 ..... 458
Mark A to Mission 2 ..... 459
Mark B to Approach phase ..... 460
Stick ..... 460
1x GCS configuration ..... 463
Air-GCS connection ..... 467
Communication ..... 467
GND unit configuration ..... 468
Air unit configuration ..... 468
GND-Air communication configuration ..... 469
GND-Air data transmission configuration ..... 470
Radio pairing ..... 472
Stick ..... 473
Ground unit ..... 474
Air unit ..... 476

# Scope of Changes

- Version 1.0
  - Added:
    - First version issued
- Version 1.1
  - Added:
    - Addresses in HEX.
- Version 1.2
  - Added:
    - Integer Variable (Uvar) 167: GNSS2 Fix Type
  - Changed:
    - Integer Variable (Uvar) 117: GNSS1 Fix Type
- Version 1.3
  - Added:
    - New Firmware Changelog sections, from v7.8.18 to v7.8.80
    - Real Variable (RVar) 299: Target Acquisition Period
    - BIT Variable 141: EKF height not teleported
    - BIT Variable 142: EKF lonlat not teleported
    - BIT Variable 144: EKF position fix not fresh
    - BIT Variable 850: GPIO A OFF
    - BIT Variable 851: GPIO B OFF
    - PDI Error 73: Wrong tunnel configuration
  - Changed:
    - BIT Variable 100: Navigation position not fixed, renamed from "Position not fixed / Position Fix"
    - BIT Variable 130: Position fix not ready for EKF, renamed from "EKF Navigation ERROR / EKF Navigation OK"
    - BIT Variable 145: Memory allocation C2 ERROR, moved from ID 139 to ID 145
  - Removed:
    - BIT Variable 329: 3.3V Power Source Bit

# Software Applications

## Veronte Link

**Veronte Link** establishes communication between a computer and any Veronte product by creating a VCP bridge.
It allows multiple control stations and autopilots to be interconnected, operating simultaneously.
**Veronte Link** also includes a post-flight viewer to reproduce all recorded data from previous flights and generate plots and reports.

For more information, visit the Veronte Link user manual.

## 1x PDI Builder

**1x PDI Builder** is the main configuration tool to adapt a **Veronte Autopilot 1x** to a specific vehicle, including user-defined communication protocols.
**1x PDI Builder** includes:

- **Telemetry**: Real-time onboard UAV metrics, such as sensors, actuators, and control states.
- **Configuration**: Edit vehicle settings, such as servo trim, interface/port management, and modes.
- **Automations**: Actions that are automatically executed when a set of configured conditions is met.
- **Block Programs**: **Veronte Autopilot 1x** can be programmed with a user-friendly programming language.

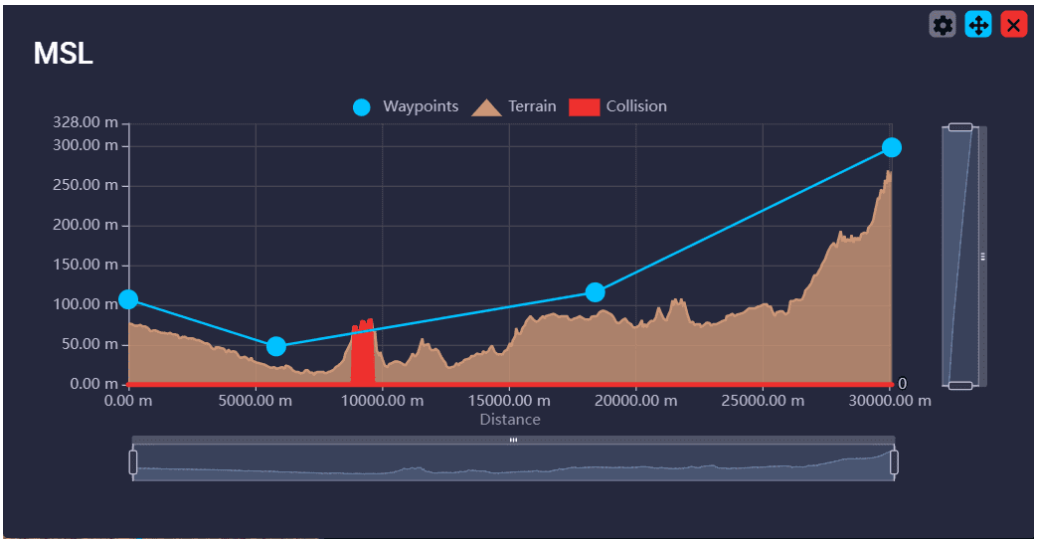
For more information, visit the 1x PDI Builder user manual.

## Veronte Ops

Veronte Ops is the application used to operate and monitor the vehicle during missions.

To know more, read the Veronte Ops user manual.

**Veronte Terrain Provider** estimates and displays the terrain height and the aircraft height.
When working in conjunction with the Veronte Ops application, it helps to avoid collisions.




**Heights example from Veronte Ops widget**

For more information about **Veronte Terrain Provider**, read the Veronte Terrain Provider - Additional apps section of the **Veronte Ops** user manual.

## Veronte HIL

**Veronte HIL** (Hardware In the Loop) is a simulation package for autopilot integration, development, and operator training.
This software allows extensive operation of the flight system in a simulated environment before executing real flight operations.
Its role is to perform HIL simulations with the real autopilot hardware, allowing the use of simulation applications like **X-Plane, Microsoft Flight Simulator,** or **Simulink**.

For more information, visit the Veronte HIL user manual.

## Veronte Updater

**Veronte Updater** updates all Embention products.

For more information, visit the Veronte Updater user manual.

# 1x PDI Calibration

**1x PDI Calibration** sets up calibration parameters for 1x autopilots.
It allows the user to calibrate sensors, servos, and configure the radio module.

For more information, visit the 1x PDI Calibration user manual.

# Veronte FDR

**Veronte FDR** manages autopilot files, allowing users to download registers generated by the autopilot and convert them to CSV files.
Three types of registers can be downloaded: **Onboard log, Fast log,** and **User log**.

For more information, visit the Veronte FDR user manual.

# Veronte VSA

**Veronte VSA** works using a flight simulator to represent worldwide geographical scenarios: lands, seas, mountains, cities, airports, airfields, heliports...
Additionally, an internet connection is not necessary, allowing operation from any location without delays in scenario loading.

**Veronte VSA** displays a 3D view of the piloted aircraft and can be used as a **3D PFD (Primary Flight Display)** when using the first-person camera view. This system allows custom aircraft models to be displayed in the virtual environment. The **Planemaker** tool is available for creating custom models, enabling operators to visualize aircraft models in the interface.

For more information, visit the Veronte VSA user manual.

# 1x PDI Tuning

**1x PDI Tuning** allows managing the **control laws** of the **Autopilot 1x** during operation.
Users can adjust each **P (Proportional), I (Integral),** and **D (Derivative)** gain and select the **PID type** (Standard or Parallel).

For more information, visit the 1x PDI Tuning user manual.

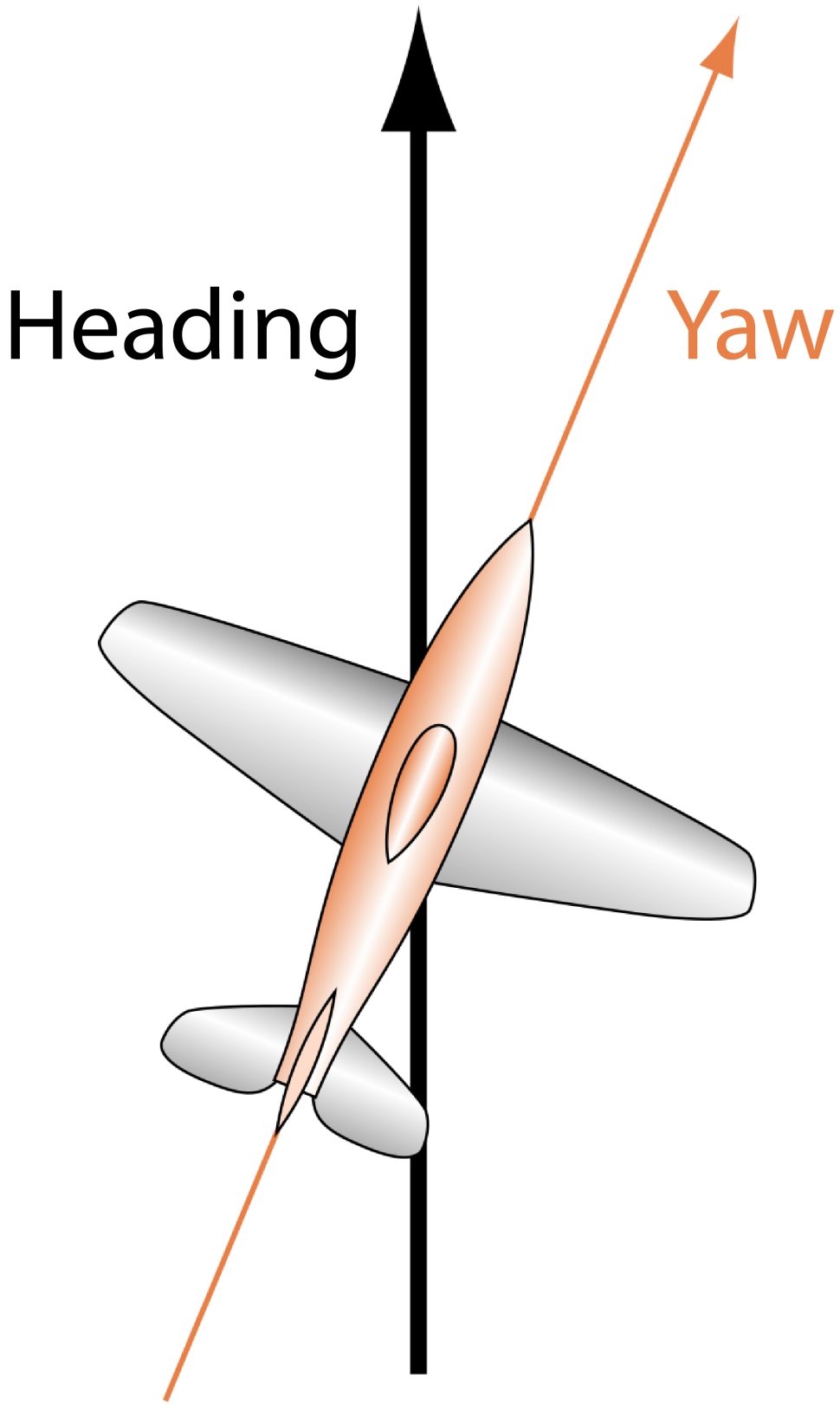
# Nomenclature

This section defines the nomenclature convention employed by the software applications.

## Reference Directions

- **Yaw** is the direction where the aircraft is pointing. It does not depend on the movement, as **Yaw** is aligned with the longitudinal axis of the aircraft.
- **Heading** is the movement direction projected to the ground. **Heading** does not depend on wind or **Yaw** direction; it only depends on the ground and the aircraft's movement.

> **Important**
>
> Both **Yaw** and **Heading** angles are measured concerning the **True North**, not the **Magnetic North**.




**Direction names**

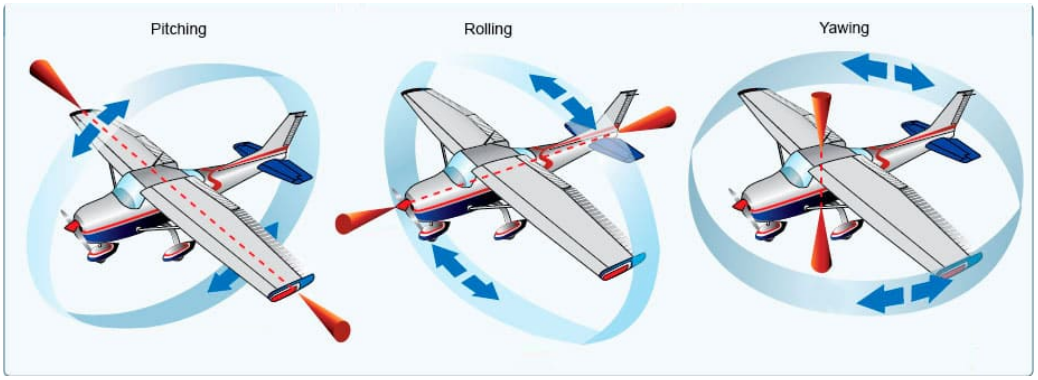
# Axes

All signs are defined according to the **international aeronautical axes convention**:
Any deflection that generates **positive rotational forces** relative to the aircraft's aerodynamic center is considered **positive**, except for the **"y" axis (elevator)**, where downward movement is considered **negative**.

For example, when the elevator moves down, it generates a positive pitch, so the elevator is considered **positive** in the lower position. Main Actuator Rules:

| Actuator | Positive | Negative |
|---|---|---|
| Elevator | Down | Up |
| Rudder | Right | Left |
| Right Aileron | Up | Down |
| Left Aileron | Down | Up |
| Tail Rotor | Right | Left |

Additionally, rotation names are summarized in the figure below:




**Rotation names**

# Core architecture

**Veronte Autopilot 1x** is equipped with a Dual-Core Microcontroller. Both cores, hereinafter called **Core 1 (C1)** and **Core 2 (C2)**, work together to perform information processing operations efficienty and to coordinate the activities of other system components.

It is crucial to understand the performance of C1 and C2 in order to properly handle Autopilot 1x tasks.

- **Core 1**: It presents the following operation threads.
  - **High**: 1kHz guaranteed. Hardware interrupt in charge of certain critical or high-priority tasks which must be executed with a constant and predictable frequency.

    > ⓘ **Note**
    > There is a permitted frequency fluctuation of 1%.

  - **Low**: Not guaranteed rate. In charge of non-priority tasks which tolerate non-guaranteed time consistency.
- **Core 2**: Execution rate of 400 Hz (configurable from 1x PDI Builder application).

## Task distribution

Tasks distribution between processors:

| Core 1 | | Core 2 |
|---|---|---|
| High | Low | |

| Core 1 | | Core 2 |
|---|---|---|
| • Sensors reading<br>• FTS Management<br>• I/O management (data from peripherals to internal pre-processing queues) | • Telemetry and logs management<br>• Communication management<br>• Files management<br>• Additional helper tasks<br>• SD writing | • Guidance, Navigation and Control (GNC)<br>• Mission updating<br>• Automations management<br>• RAM writing |

A proper comprehension of how cores manage different tasks, considering the priorities and execution rates previously explained, may be valuable to avoid delays in data processing and achieving a balanced functioning of the Veronte system.

Aspects to be considered:

- **C1 and C2 information interchange**: Cores share information through the **Cross-Core queue**, hence overfilling this queue may result in late processing of data.
- **Tasks distribution between C1 threads**: If acquisition tasks take too much processing time, C1 low task may not be executed as expected.
- **C1 High interruptions**: C1 high may interrupt C1 low task execution. In this case, C1 low will continue its execution at the same point once high priority tasks are fullfilled.

# Monitoring variables

Distributing resources is decisive for the proper functioning of the system. For that reason, core-related values are monitored.

> ⓘ **Note**
>
> For further information regarding these variables, please consult their IDs in the Lists of Variables section of the present manual.

C1 is monitored by the following variables:

| Core 1 | |
|---|---|
| **High** | **Low** |
| | • C1 Low Frequency Fail (BIT 400)<br>• CIO Max Time (RVar 2054)<br>• CIO Average Time (RVar 2055)<br>• CIO Running Frequency (RVar 2057) |

| Core 1 | |
|---|---|
| • Acquisition step missed (BIT 402)<br>• CIO Hi Overload warning (BIT 403)<br>• Acquisition Task Timestep (RVar 2047)<br>• Acquisition Task Maximum Timestep (RVar 2048)<br>• Acquisition Task Average CPU Ratio (RVar 2050)<br>• Acquisition Task Maximum CPU Ratio (RVar 2051)<br>• Acquisition Task Average Time (RVar 2052)<br>• Acquisition Task Maximum Time (RVar 2053)<br>• Identifier of max duration step in acquisition (UVar 399) | |

C2 is monitored by the following variables:

| Core 2 |
| --- |
| • GNC Fail (BIT 401)<br>• GNC Realtime Error (BIT 404)<br>• GNC Task Average CPU Ratio (RVar 2094)<br>• GNC Task Maximum CPU Ratio (RVar 2095)<br>• GNC Task Average Time (RVar 2096)<br>• GNC Task Maximum Time (RVar 2097)<br>• GNC Task Maximum Timestep (RVar 2098)<br>• Max Duration of Step in GNC (RVar 2099)<br>• GNC Timestep (RVar 2903)<br>• Counter for C2 system (UVar 20) |

Cross-Core queue is monitored by the following variables:

**Cross-Core Queue**

- Cross Core Message Queue CPU Ratio (RVar 2049)
- Cross-Core Message Queue Usage (RVar 2056)

# Data Transmission

**Veronte Autopilot 1x** uses the Veronte Communicatin Protocol (VCP) for its communication with other **Veronte devices** or **Veronte applications**, such as **Veronte Ops**.

Communication originates from any COM Manager port and is sent/received through any interface connected to it (e.g. USB ↔ Commgr port 0).

**Autopilot 1x** will send:

- 1 VCP packet per **vector telemetry**.
- 1 VCP packet of **complementary vector telemetry**; only in case a widget whose variables have not been previoulsy added to the telemetry configuration is added in Veronte Ops.

  > **Note**
  > Complementary vector telemetry always runs at a frequency of 10 Hz (not configurable) and variables are sent uncompressed.

- 1 VCP packet of **health status**.
- 1 VCP packet to update the **route** traced in **Veronte Ops**.

> **Important**
>
> A VCP message contains **11 bytes** of addresses (source and destination), CRC, header, etc., and **up to 255 bytes** of **data for telemetry vectors**.
>
> The maximum packet size supported by **VCP** is **266 bytes**. For more information on the VCP packet, read the Message structure section of the **VCP** user manual.
>
> Thus, it is up to the user to determine, based on its configuration, the amount of data in bytes to be sent from connected devices such as Radios, SatComs, etc., making sure that the bandwith of the device is enough.

**Tip**

Functional tests such as monitoring the value of the variables related to the interface used are recomended.

For example, if a radio connected to the RS232 is used for communication to the Ground Control Station (GCS), it is highly advised to monitor the variables:

- RVar 48 - SCI-D Rx Rate (RS232)
- RVar 49 - SCI-D Tx Rate (RS232)

# Lists of interest

This section contains all the lists with information of interest for the user.

## Activation System Error bits

The **System Error** variable is indicated by bit number 7. This bit checks whether the system is running properly. If one of certain malfunctions occur, the **System Error** will be set as 0 and the FTS will be activated. Otherwise, if everything is OK, it will remain as 1.

The **System Error** is triggered and set as 0 if one of the following unwanted events happens:

- **CIO low** has a frequency lower than 10 Hz. This error is indicated with a 0 in bit 400.
- **CIO high** has a frequency lower than 990 Hz. This error is indicated with a 0 in bit 402.
- **GNC** is 'dead'. This event is indicated with a 0 in bit 401.
- **GNC Realtime Error** because a GNC Step has been missed. This event is indicated with a 0 in bit 404.
- **System Power Up Error bit**. The initial value of this bit depends on the values of the bits listed below and represents their state at power-up. This error is indicated with a 0 in bit 12 if any of the following errors happens:
  - **RAM allocation** is in error state due to trying to use more memory than available. This error is indicated with a 0 in bit 8.
  - **PDI files** have a wrong configuration. This is indicated by a 0 in bit 9.
  - **Main Power supply A** is in error state. This error is indicated with a 0 in bit 117 if any of the following errors happens:
    - **Input supply voltage** is not between 6.5 and 36 V. This voltage is measured by RVar 400.
    - **Voltage received by Veronte through 5V port** is not between 4.75 and 5.25 V. This voltage is measured by RVar 402.
    - **Voltage received by Veronte through 3.6V port** is not between 3.42 and 3.78V. This voltage is measured by RVar 404.

- **File system manager** is in error state. This event is indicated with a 0 in bit 6.
- **Core 1** has a memory overflow allocated for local variables. This error is indicated with a 0 in bit 16.
- **Core 2** has a memory overflow allocated for local variables. This error is indicated with a 0 in bit 17.
- Any user bit configured as **safety bit** is 0. User bits are 1200 to 1499.

## Status Message variables

The variables contained in the Veronte Autopilot 1x status message are the following.

| Type | ID | Name |
|---|---|---|
| UVar | 0 | Actuator Mode |
| UVar | 1 | Phase Identifier |
| RVar | 6 | Yaw |
| RVar | 7 | Pitch |
| RVar | 8 | Roll |
| BIT | 100 | Position Not Fixed |
| UVar | 201 | Current Section |
| UVar | 204 | Current patchset ID (needed for guidance display) |
| RVar | 300 | Relative Timestamp |
| RVar | 505 | North Ground Velocity |
| RVar | 506 | East Ground Velocity |

| Type | ID | Name |
|---|---|---|
| RVar | 507 | Down Ground Velocity |
| UVars | - | Guidance calculation identifier (needed for guidance display).<br><br>It can be one of these variables depending on the guidance used:<br><br>• **ID** 80: Detour calculation identifier<br>• **ID** 81: Approach calculation identifier<br>• **ID** 82: Climb calculation identifier<br>• **ID** 83: Cruise calculation identifier<br>• **ID** 84: Rendezvous calculation identifier<br>• **ID** 85: Taxi calculation identifier<br>• **ID** 86: VTOL calculation identifier |

## Extended Status Message variables

Information about the Veronte Autopilot 1x extended status message is contained in the following set of **bit** variables.

> ⓘ **Note**
>
> The **Extended Status Message** is only sent if one of the bits changes its value.

| ID | Name |
|---|---|
| 5 | Power Error |
| 6 | File System Error |
| 8 | Memory allocation Error |
| 9 | PDI Error |
| 12 | System Power Up Error |
| 14 | FTS-1 Feedback (>=V4.5) |
| 15 | FTS-2 Feedback (>=V4.5) |
| 16 | Stack C1 usage FAIL |
| 17 | Stack C2 usage FAIL |
| 49 | CPU temperature |

| ID | Name |
|---|---|
| | above 398.15K |
| 50 | Sensors Error |
| 51 | IMU 0 Error |
| 52 | IMU 1 Error |
| 53 | Magnetometer 0 Error |
| 73 | CAN-A Error |
| 74 | CAN-B Error |
| 87 | GNSS1 Module Error |
| 88 | GNSS2 Module Error |
| 117 | Main Power Error |
| 118 | SUC Power Error |
| 400 | C1 Low Frequency Fail |
| 401 | GNC Fail |
| 402 | Acquisition step missed |

| ID | Name |
|---|---|
| 403 | CIO Hi Overload warning |

## List of Addresses

Every Embention device communicate with other devices/tools using its address through VCP.

The following list contains all these addresses:

| Address (DEC) | Address (HEX) | Recognized as | Description |
|---|---|---|---|
| 0 | 0x00000000 | Dummy for pdi builders | Dummy for pdi builder |
| 1 | 0x00000001 | Cloud | **Veronte Cloud** address |
| 2 | 0x00000002 | Vlink | Address used by **Veronte Link** app to communicate with Veronte units |
| 2-3 | 0x00000002 - 0x00000003 | App + Address | Veronte applications addresses. **App 2** is the one used by default by Veronte |

| Address (DEC) | Address (HEX) | Recognized as | Description |
|---|---|---|---|
| | | | applications, although App 3 is also available |
| 255-511 | 0x000000FF - 0x000001FF | App dynamic + Address | **Dynamic addresses** for Veronte applications |
| 998 | 0xFFFFFFFF | Broadcast | To **all devices** on a network |
| 999 | 0xFF000000 | Address unknown | This address can be used for a device that **does not** have a **valid address** configured |
| 1000-1777 | 0x000003E8 - 0x000006F1 | 1x v4.0 + Address | Specific address of an **Autopilot 1x** with **hardware version 4.0** |
| 1778-3999 | 0x000006F2 - 0x00000F9F | 1x v4.5 + Address | Specific address of an **Autopilot 1x** with |

| Address (DEC) | Address (HEX) | Recognized as | Description |
|---|---|---|---|
| | | | **hardware version 4.5** |
| 4000-17999 | 0x00000FA0 - 0x0000464F | 1x v4.8 + Address | Specific address of an **Autopilot 1x** with **hardware version 4.8** |
| 18000-19899 | 0x00004650 - 0x00004DBB | 1x BCS + Address | Specific address of a **BCS** unit |
| 19900-19999 | 0x00004DBC - 0x00004E1F | 1x v4.7. For internal use only + Address | Specific address of an **Autopilot 1x** with **hardware version 4.7** |
| 20000-21999 | 0x00004E20 - 0x000055EF | Smart Can Isolator + Address | Specific address of a **Smart Can Isolator** unit |
| 30000-31999 | 0x00007530 - 0x00007CFF | MC01 + Address | Specific address of a **MC01** unit |
| 32000-34999 | 0x00007D00 - 0x000088B7 | MC24 motor controller + Address | Specific address of a **MC24** unit |

| Address (DEC) | Address (HEX) | Recognized as | Description |
|---|---|---|---|
| 35000-35299 | 0x000088B8 - 0x000089E3 | MC110 motor controller + Address | Specific address of a **MC110** unit |
| 35300-39999 | 0x000089E4 - 0x00009C3F | MC110 v2 motor controller + Address | Specific address of a **MC110 v2** unit |
| 40000-41999 | 0x00009C40 - 0x0000A40F | CEX + Address | Specific address of a **CEX** with **hardware version 1.2** |
| 42000-43999 | 0x0000A410 - 0x0000ABDF | MEX + Address | Specific address of a **MEX** unit |
| 44000-49999 | 0x0000ABE0 - 0x0000C34F | CEX2 + Address | Specific address of a **CEX** with **hardware version 2.0** |
| 50000-51089 | 0x0000C350 - 0x0000C791 | Arbiter v1.0 + Address | Specific address of an **Arbiter** with **hardware version 1.0** |
| 51090-51999 | | | |

| Address (DEC) | Address (HEX) | Recognized as | Description |
|---|---|---|---|
| | 0x0000C792 - 0x0000CB1F | Arbiter v1.2 + Address | Specific address of an **Arbiter** with **hardware version 1.2** |
| 52000-59999 | 0x0000CB20 - 0x0000EA5F | Arbiter v1.8 + Address | Specific address of an **Arbiter** with **hardware version 1.8** |
| 60000-65535 | 0x0000EA60 - 0x0000FFFF | Reserved + Address | Reserved addresses |
| 65536-69631 | 0x00010000 - 0x00010FFF | Virtual v4.0 + Address | Specific address of a **Virtual Autopilot 1x** with **hardware version 4.0** |
| 69632-73727 | 0x00011000 - 0x00011FFF | Virtual v4.5 + Address | Specific address of a **Virtual Autopilot 1x** with **hardware version 4.5** |
| 73728-77823 | | | Specific address of a |

| Address (DEC) | Address (HEX) | Recognized as | Description |
|---|---|---|---|
| | 0x00012000 - 0x00012FFF | Virtual v4.8 + Address | **Virtual Autopilot 1x** with **hardware version 4.8** |

## Lists of Variables

This section shows all the variables employed by **Veronte Autopilot 1x**. All of them can be read and sent through telemetry.

These variables are classified into two main groups:

- **System variables**: Non-writable by the user.
- **User variables**: Writable by the user, marked in the tables below as 📝.

> **Important**
>
> Variables labeled as "**Deprecated**" are no longer used by the system.
>
> In order to avoid system incompatibilities, deprecated variables are only written by the autopilot when migrating a configuration from a previous version in which the variable was not obsolete.
> When a configuration is built from scratch, these variables must be defined by the user to be used.
> Thus, users are responsible for their correct assignment.

## BIT Variables

> ⚠ **Warning**
>
> Bit Variables displayed on **Veronte Ops** labels will be shown as Red/Green depending on their state.
> **Red** stands for `0` and **Green** for `1`, changing the name displayed accordingly to the BIT value.

| ID | Name | Description |
|---|---|---|
| 0 | Always Fail | This signal is always fail - 0 for fail, 1 for OK |
| 1 | Always OK | This signal is CIO always OK - 0 for fail, 1 for OK |
| 2 | License Check Pending | License state - 0 for license check pending, 1 for license checked |
| 3 | System Not Ready to Start | System is ready to start operating - 0 for not ready, 1 for ready |
| 4 | No Writing Telemetry | Telemetry is properly sending/ receiving - 0 for no, 1 for yes |
| 5 | Power ERROR | Power supply state - It will be 0 |

| ID | Name | Description |
|---|---|---|
| | | if any of the following conditions happens:<br><br>• **Bit 117** is zero (power for Veronte has a failure)<br>• **Bit 118** is zero (power for SuC has a failure) |
| 6 | File System ERROR | System file manager - Dependent on **File system status (UVar 96)**<br><br>• 0 for Error: if File system status > 0<br>• 1 for running OK: if File system status == 0 |
| 7 | System ERROR | This bit checks whether the system is running properly. 0 for |

| ID | Name | Description |
|---|---|---|
| | | system error, 1 for system OK. |
| 8 | Memory allocation ERROR | RAM allocation - 0 for Error, i.e. trying to use more than available memory, 1 for memory allocation ok |
| 9 | PDI ERROR | PDI files - Dependent on **PDI error source (UVar 50)**<br><br>• 0 for wrong PDI configuration: if PDI Error Source > 0<br>• 1 for running OK: if PDI Error Source == 0 |
| 10 | CIO Low or C2 ERROR (Deprecated) | CIO low or C2 failed. Bits 400 and 401 are recommended instead<br>- 0 for CIO low or C2 failed, 1 for |

| ID | Name | Description |
|---|---|---|
| | | CIO high and C2 OK<br><br>**Warning**<br>**Deprecated variable** |
| 11 | 4x CAN Fail | For more information, check **BIT Variables - 4x Software Manual** |
| 12 | System Power Up ERROR | Power up - 0 for Error, 1 for OK |
| 13 | Reset and Write Disabled | Reset and non-operation PDI writes are allowed - 0 for disabled, 1 for enabled |
| 14 | FTS-1 Feedback (>=V4.5) | Flight Termination System 1, microcontroller state for hardware version 4.7 or higher - 0 for Error, 1 for running OK |
| 15 | | Flight Termination System 2, |

| ID | Name | Description |
|---|---|---|
| | FTS-2 Feedback (>=V4.5) | microcontroller state for hardware version 4.7 or higher - 0 for Error, 1 for running OK |
| 16 | Stack C1 usage FAIL | 0 for Fail, i.e. stack overflow of Core 1, 1 for OK |
| 17 | Stack C2 usage FAIL | 0 for Fail, i.e. stack overflow of core 2, 1 for OK |
| 18 | PDI disabled | PDI Mode - 0 for disabled, 1 for enabled |
| 20-46 | 4X Bit variables | For more information, check **BIT Variables - 4x Software Manual** |
| 47 | 4X Watchdog ERROR | **Note** For version 4.7 or higher<br><br>For more information, check **BIT Variables - 4x** |

| ID | Name | Description |
|---|---|---|
| | | **Software Manual** |
| 49 | CPU temperature above 398.15K | CPU temperature warning - 0 for CPU temperature above 398.15K (125ºC), 1 for CPU temperature below 398.15K (125ºC) |
| 50 | Sensors ERROR | Sensors state - 0 for Error, 1 for running OK Selected sensors are not working or, if external sensors have been selected, they are not connected |
| 51 | IMU 0 ERROR | Sensor IMU 0 - 0 for Error, 1 for OK |
| 52 | IMU 1 ERROR | Sensor IMU 1 - 0 for Error, 1 for OK |
| 53 | Magnetometer 0 ERROR | **Internal 0 Magnetometer** - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 55 | Sensor-External Magnetometer (LIS3MDL) | **External LIS3MDL magnetometer** - 0 for Error, 1 for OK |
| 56 | Sensor-Static pressure 0 | Static Pressure Sensor 0 - 0 for Error, 1 for OK |
| 57 | Sensor-Static pressure 1 | Static Pressure Sensor 1 - 0 for Error, 1 for OK |
| 58 | Sensor-Dynamic pressure | Dynamic Pressure Sensor 0 - 0 for Error, 1 for OK |
| 59 | Sensor-External I2C devices ERROR | 0 for Error, 1 for OK |
| 60-64 | Sensor-External I2C device 0-4 ERROR | External communication I2C device 0-4 - 0 for Error, 1 for OK |
| 65 | SCI-A Transmitting (LTE/EXT. UART) | Serial Communication Interface LTE/EXT. UART transmission - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 66 | SCI-A Receiving (LTE/ EXT. UART) | Serial Communication Interface LTE/EXT. UART reception - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 67 | SCI-B Transmitting (LOS) | Serial Communication Interface LOS transmission - 0 for Error, 1 for OK |
| 68 | SCI-B Receiving (LOS) | Serial Communication Interface LOS reception - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 69 | SCI-C Transmitting (RS485) | Serial Communication Interface RS485 transmission - 0 for Error, 1 for OK |
| 70 | SCI-C Receiving (RS485) | Serial Communication Interface RS485 reception |

| ID | Name | Description |
|---|---|---|
| | | - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 71 | SCI-D Transmitting (RS232) | Serial Communication Interface RS232 transmission - 0 for Error, 1 for OK |
| 72 | SCI-D Receiving (RS232) | Serial Communication Interface RS232 reception - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 73 | CAN-A ERROR | CAN A state - 0 for Error, 1 for OK |
| 74 | CAN-B ERROR | CAN B state - 0 for Error, 1 for OK |
| 75 | CAN-A Warning | CAN A state - 0 for Warning, 1 for OK |
| 76 | CAN-B Warning | CAN B state - 0 for Warning, 1 for OK |
| 77 | | |

| ID | Name | Description |
|---|---|---|
| | Vectornav GPS not fixed | 0 for not fixed, 1 for fixed |
| 78 | Vectornav IMU ERROR | 0 for Error, 1 for OK |
| 79 | Vectornav Mag/Press ERROR | 0 for Error, 1 for OK |
| 80 | Vectornav GPS ERROR | 0 for Error, 1 for OK |
| 81 | Vectornav Navigation ERROR | Navigation state - 0 for Error, 1 for OK |
| 82 | Sensor-External Magnetometer (HSCDTD008A) | **External HSCDTD008A magnetometer** - 0 for Error, 1 for OK |
| 83 | IMU 2 ERROR | Sensor IMU 2 - 0 for Error, 1 for OK |
| 84 | Sensor-Static pressure 2 | Static Pressure Sensor 2 - 0 for Error, 1 for OK |
| 85 | Magnetometer 1 ERROR | **Internal 1 Magnetometer** - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 86 | Sensor-External Magnetometer (MMC5883MA) | **External MMC5883MA magnetometer** - 0 for Error, 1 for OK |
| 87 | GNSS1 Module ERROR | GPS module 1 state - 0 for Error, 1 for OK |
| 88 | GNSS2 Module ERROR | GPS module 2 state - 0 for Error, 1 for OK |
| 89 | Sensor-External Magnetometer (RM3100) | **External RM3100 magnetometer** - 0 for Error, 1 for OK |
| 90 | IMU3 ERROR | Sensor IMU 3 - 0 for Error, 1 for OK |
| 91 | Magnetometer 2 ERROR | **Internal 2 Magnetometer** - 0 for Error, 1 for OK |
| 92 | Magnetometer reserved | 0 for Error, 1 for OK |
| 93 | SCI Expander (v4.8+) | SCI Expander for hardware version |

| ID | Name | Description |
|---|---|---|
| | | 4.8 or higher - 0 for Error, 1 for OK |
| 96 | SCI-A ERROR (LTE/EXT. UART) | SCI A (LTE/EXT. UART) state - 0 for Error in this port (invalid format or configuration), 1 for OK |
| 97 | SCI-B ERROR (LOS) | SCI B (LOS) state - 0 for Error in this port (invalid format or configuration), 1 for OK |
| 98 | SCI-C ERROR (RS485) | SCI C (RS485) state - 0 for Error in this port (invalid format or configuration), 1 for OK |
| 99 | SCI-D ERROR (RS232) | SCI D (RS232) state - 0 for Error in this port (invalid format or configuration), 1 for OK |
| 100 | Navigation position not fixed | 0 if Navigation position not fixed, |

| ID | Name | Description |
|---|---|---|
| | | 1 if Navigation position fix |
| 101 | No valid SRTM at UAV position | 0 for not valid, 1 for valid |
| 102-103 | CAN-A-B Receiving ERROR | CAN A-B communication - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 104-105 | Stick PPM 0-1 not detected | Stick PPM 0-1 - 0 for not detected, 1 for detected |
| 106 | Magnetic field out of bounds (Deprecated) | 0 for magnetic field out of bounds, 1 for OK<br>**Warning**<br>**Deprecated variable** |
| 107 | INS navigation OFF | 0 for INS navigation OFF, 1 for INS navigation ON |
| 108-109 | Stick PPM 2-3 not detected | Stick PPM -3 - 0 for not detected, 1 for detected |

| ID | Name | Description |
|---|---|---|
| 110 | Stick Not Detected | Stick detection - 0 for not detected, 1 for OK |
| 111-112 | CAN-A-B Transmitting ERROR | CAN signals A-B - 0 for Error, i.e. not transmitting, 1 for OK, i.e. transmitting |
| 113 | Iridium not ready | Iridium ready state - 0 for not ready, 1 for ready |
| 114 | No valid Geoid at UAV position | 0 for no valid geoid at UAV position, 1 for valid geoid at UAV position |
| 115 | EKF: Condition Number ERROR | Extended Kalman Filter state - 0 for Error, 1 for OK |
| 116 | Radar Altimeter CAN-RX ERROR | Radar Altimeter Reception State - 0 for Error, 1 for OK |
| 117 | Main Power ERROR | Main power supply A. It will be 0 (indicating error state) if any of the following errors |

| ID | Name | Description |
|---|---|---|
| | | happen:<br><br>• Input supply voltage is not between 6.5 and 36 V. This voltage is measured by **RVar 400**.<br>• Voltage received by Veronte through 5V port is not between 4.75 and 5.25 V. This voltage is measured by **RVar 402**.<br>• Voltage received by Veronte through 3.6V port is not between 3.42 and 3.78V. This voltage is measured by **RVar 404**. |
| 118 | SUC Power ERROR | Power supply for system on microchip. It will |

| ID | Name | Description |
|---|---|---|
| | | be 0 (indicating error state) if any of the following errors happen:<br><br>• Voltage received by Veronte through 3.3V port is out of range. This voltage is measured by **RVar 401**.<br>• Voltage received by Veronte SUC is out of range. This voltage is measured by **RVar 403**. |
| 119 | Not hovering guidance | Hovering guidance state - 0 if cruise guidance is not hovering, 1 if cruise guidance is hovering |
| 120-123 | Pulse 0-3 not detected | Pulse 0-3 detection - 0 for pulse not |

| ID | Name | Description |
|---|---|---|
| | | detected, 1 for detected |
| 124-129 | 4X Bit variables | For more information, check **BIT Variables - 4x Software Manual** |
| 130 | Position fix not ready for EKF | 0 if Position fix not ready for EKF, 1 if Position fix ready for EKF |
| 131 | No magnetic field data | Bit to indicate if there is magnetic field in the SD - 0 for No Magnetic field data, 1 for Magnetic field data OK |
| 132 | Route not finished | 0 for Route not finished, 1 for Route finished |
| 133 | Operation ERROR | 0 if an operation error has occurred, 1 if no operation error has occurred |
| 134 | | Atmospheric parameters have |

| ID | Name | Description |
|---|---|---|
| | Atmosphere not calibrated | not been calibrated or received via command |
| 135 | Position command not received | The system is waiting for an initial position command to begin navigation |
| 136 | Yaw not initialized via command | Initial yaw orientation has not been initialized via operator command |
| 137 | Wind Estimation Command Not Received | 0 if wind estimation was not initialized by command, 1 if wind estimation was initialized by command |
| 139 | Navigation not initialized | General navigation algorithms or states are not yet fully initialized |
| 141 | EKF height not teleported | A vertical position jump (teleport) in |

| ID | Name | Description |
|---|---|---|
| | | the EKF is pending or failed |
| 142 | EKF lonlat not teleported | A horizontal position jump (teleport) in the EKF is pending or failed |
| 144 | EKF position fix not fresh | 0 if EKF position fix not fresh, 1 if the last position update was accepted less than 2.5 s ago |
| 145 | Memory allocation C2 ERROR | 0 if Memory allocation C2 ERROR, 1 if Memory allocation C2 OK |
| 150 | External VCP Navigation ERROR | External VCP Navigation state - 0 for Error, 1 for OK |
| 160 | External Var Navigation ERROR | External Var Navigation state - 0 for Error, 1 for OK |
| 170 | | Selected accelerometer - 0 for Error, 1 OK, |

| ID | Name | Description |
|---|---|---|
| | Selected Accelerometer ERROR | i.e. at least one of the selected accelerometers is OK |
| 171 | Selected Gyroscope ERROR | Selected gyroscope - 0 for Error, 1 O, i.e. if at least one of the selected gyroscopes is OK |
| 172 | Bias Accelerometer Saturated | 0 for bias Saturated, 1 for OK, i.e. not satured |
| 173 | Bias Gyroscope Saturated | 0 for bias Saturated, 1 for OK, i.e. not satured |
| 180 | External attitude | Kind of attitude calculation - 0 for External, 1 for Internal |
| 182 | FTS Activation (>=V4.5) | Flight Termination System activation, for version 4.5 or higher - 0 for not activated, 1 for activated |

| ID | Name | Description |
|---|---|---|
| | | **Important**<br>This bit is **not a FeedbackBit**, i.e., it will not be activated automatically by the system, but acts as a **user bit**. Therefore, for the FTS to be activated, this bit must be included in the Safety bits list. |
| 181 | Expected Magnetic Field At UAV Position Not Valid | The expected magnetic field at the UAV's current position is not valid |
| 183 | 4X Selected | Current 1x Autopilot is the one selected by the arbiter - 0 when this AP is not the selected AP, 1 when this AP is the selected one |

| ID | Name | Description |
|---|---|---|
| 184 | Custom FTS activation | A user-defined custom Flight Termination System activation has occurred |
| 187 | Dynamic Pressure Sensors ERROR | General error detected across all dynamic pressure (airspeed) sensors |
| 188 | Static Pressure Sensors ERROR | Static Pressure Sensors status - 0 for Error, 1 for OK |
| 189 | Magnetometer Sensors ERROR | Magnetometer Sensors status - 0 for Error, 1 for OK |
| 190 | Internest ultrasound position status ERROR | Internest ultrasound position status - 0 for Error, 1 for OK |
| 191 | Internest ultrasound angle status ERROR | Internest ultrasound angle status - 0 for Error, 1 for OK |
| 192 | 4G not connected | LTE/4G cellular module (SARA) is not connected to the network |

| ID | Name | Description |
|---|---|---|
| 200 | GNSS1 Navigation Down | GNSS1 Navigation status - 0 for GNSS Navigation Down, i.e. not used, 1 for GNSS Navigation ON, i.e. used |
| 201 | DGNSS1 Input OFF | DGNSS1 Input status - 0 for DGNSS1 Input Off, i.e. neither GNSS compass nor RTK activated, 1 for DGNSS1 Input ON, i.e. one of them activated |
| 202 | DGNSS1 Navigation OFF | DGNSS1 Navigation status - 0 for DGNSS1 Navigation Off, i.e. neither GNSS compass nor RTK used, 1 for DGNSS1 Navigation ON, i.e. one of them used |
| 203 | GNSS1 Survey In OFF | GNSS1 Survey In status - 0 for GNSS1 Survey In Off, i.e. neither |

| ID | Name | Description |
|---|---|---|
| | | GNSS compass survey nor RTK OFF, 1 for GNSS1 Survey In, i.e. one of them ON |
| 204 | No DGNSS1 Float Solution | 0 for no DGNSS1 Float Solution nor RTK, 1 for DGNSS1 Float Solution or RTK |
| 205 | No DGNSS1 Fixed Solution | 0 for no DGNSS1 Fixed Solution nor RTK, 1 for DGNSS1 Fixed Solution or RTK |
| 206 | DGNSS1 Relative Position Invalid | 0 for invalid navigation position, 1 for valid navigation position |
| 207 | DGNSS1 not Moving baseline mode | 0 for not Moving baseline mode, 1 for Moving baseline mode |
| 210 | DMA peripheral for SPIA A ERROR | DMA peripheral for SPIA A - 0 for Error, 1 for OK |
| 211 | DMA peripheral for | |

| ID | Name | Description |
|---|---|---|
| | MCBSP A ERROR | DMA peripheral for MCBSP A - 0 for Error, 1 for OK |
| 213 | Accelerometer Decimal Var Sensor 0 Error | Accuracy or variance error for accelerometer decimal variable sensor 0 |
| 214 | Accelerometer Decimal Var Sensor 1 Error | Accuracy or variance error for accelerometer decimal variable sensor 1 |
| 215 | Gyroscope Decimal Var Sensor 0 Error | Accuracy or variance error for gyroscope decimal variable sensor 0 |
| 216 | Gyroscope Decimal Var Sensor 1 Error | Accuracy or variance error for gyroscope decimal variable sensor 1 |
| 217 | Magnetometer Decimal Var Sensor 0 Error | Accuracy or variance error for magnetometer decimal variable sensor 0 |

| ID | Name | Description |
|---|---|---|
| 218 | Magnetometer Decimal Var Sensor 1 Error | Accuracy or variance error for magnetometer decimal variable sensor 1 |
| 219 | Static Pressure Decimal Var Sensor 0 Error | Accuracy or variance error for static pressure decimal variable sensor 0 |
| 220 | Static Pressure Decimal Var Sensor 1 Error | Accuracy or variance error for static pressure decimal variable sensor 1 |
| 221 | Dynamic Pressure Decimal Var Sensor 0 Error | Accuracy or variance error for dynamic pressure decimal variable sensor 0 |
| 222 | Dynamic Pressure Decimal Var Sensor 1 Error | Accuracy or variance error for dynamic pressure decimal variable sensor 1 |
| 230-293 | 4X Bit variables | For more information, check **BIT Variables - 4x** |

| ID | Name | Description |
|---|---|---|
| | | **Software Manual** |
| 300 | GNSS2 Navigation Down | GNSS2 Navigation status - 0 for GNSS Navigation Down, i.e. not used, 1 for GNSS Navigation ON, i.e. used |
| 301 | DGNSS2 Input OFF | DGNSS2 Input status - 0 for DGNSS2 Input Off, i.e. neither GNSS compass nor RTK activated, 1 for DGNSS2 Input ON, i.e. one of them activated |
| 302 | DGNSS2 Navigation OFF | DGNSS2 Navigation status - 0 for DGNSS2 Navigation Off, i.e. neither GNSS compass nor RTK used, 1 for DGNSS2 Navigation ON, i.e. one of them used |
| 303 | | |

| ID | Name | Description |
|---|---|---|
| | GNSS2 Survey In OFF | GNSS2 Survey In status - 0 for GNSS2 Survey In Off, i.e. neither GNSS compass survey nor RTK OFF, 1 for GNSS2 Survey In, i.e. one of them ON |
| 304 | No DGNSS2 Float Solution | 0 for no DGNSS2 Float Solution nor RTK, 1 for DGNSS2 Float Solution or RTK |
| 305 | No DGNSS2 Fixed Solution | 0 for no DGNSS2 Fixed Solution nor RTK, 1 for DGNSS2 Fixed Solution or RTK |
| 306 | DGNSS2 Relative Position Invalid | 0 for invalid navigation position, 1 for valid navigation position |
| 307 | DGNSS2 not Moving baseline mode | 0 for not Moving baseline mode, 1 for Moving baseline mode |
| 308 | | |

| ID | Name | Description |
|---|---|---|
| | SCI-E Transmitting (LTE) | SCI-E Transmitting (LTE) - 0 for Error, 1 for OK |
| 309 | SCI-E Receiving (LTE) | SCI-E Receiving (LTE) - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 310 | SCI-F Transmitting (LTE Aux.) | SCI-F Transmitting (LTE Aux.) - 0 for Error, 1 for OK |
| 311 | SCI-F Receiving (LTE Aux.) | SCI-F Receiving (LTE Aux.) - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 312 | SCI E ERROR (LTE) | SCI-E (LTE) - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 313 | SCI F ERROR (LTE Aux.) | SCI-F (LTE Aux.) - 0 for Error, i.e. not receiving, 1 for OK, i.e. receiving |
| 330 | Jetibox COMM ERROR | Jetibox is communicating properly - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 331 | CEX HI-3210 ERROR | HI-3210 is working properly - 0 for Error, 1 for OK |
| 351 | Planet SATCOM connection Fail | Planet SATCOM is connected - 0 for Error, 1 for OK |
| 370-371 | Smart Can Isolator A-B Domain ERROR | 0 for Error, 1 for OK |
| 400 | C1 Low Frequency Fail | C1 Low Frequency - Dependent on **CIO Running Frequency (RVar 2057)** (C1 low frequency)<br><br>• 0 for Fail → CIO Running Frequency < 10 Hz<br>• 1 for OK → CIO Running Frequency > 10 Hz |
| 401 | GNC Fail | 0 for Fail ('dead'), 1 for OK ('alive') - Dependent on |

| ID | Name | Description |
|---|---|---|
| | | **Counter for C2 system (UVar 20)** |
| 402 | Acquisition step missed | • 0 for Acquisition step missed → C1 hi frequency fluctuation is higher than permitted (1%).<br>• 1 for Acquisition Task OK → C1 hi frequency fluctuation is under set limits (1%). |
| 403 | CIO Hi Overload warning | C1 Hi Overload - Dependent on **Acquisition Task Maximum CPU Ratio (RVar 2051)** |

| ID | Name | Description |
|---|---|---|
| | | • 0 for Acquisition Task overload → Acquisition Task Maximum CPU Ratio > 90%<br>• 1 for Acquisition Task usage OK → Acquisition Task Maximum CPU Ratio ≤ 90%<br><br>**Note**<br>Non-recoverable variable |
| 404 | GNC Realtime ERROR | |

| ID | Name | Description |
|---|---|---|
| | | • 0 if C2 frequency < configured frequency (tolerance of 6 microseconds)<br>• 1 if C2 frequency = configured frequency (tolerance of 6 microseconds) |
| 405 | Reserved | 0 for Error, 1 for OK |
| 480 | MC01 Stepper direction output ERROR | 0 for Error, 1 for OK |
| 481 | MC01 Brushless driver fault | 0 for Error, 1 for OK |
| 482 | MC Hall Sensor ERROR | 0 for Error, 1 for OK |
| 483 | MC Sin/Cos Sensor ERROR | 0 for Error, 1 for OK |
| 484 | MC general health ERROR | 0 for health Error, 1 for status OK |

| ID | Name | Description |
|---|---|---|
| 485 | MC Current sensing ERROR | 0 for Error, 1 for OK |
| 486 | MC Phase U Current Calibration ERROR | ADC phase U calibration status - 0 for not calibrated, 1 for calibration OK |
| 487 | MC Phase V Current Calibration ERROR | ADC phase V calibration status - 0 for not calibrated, 1 for calibration OK |
| 488 | MC Phase W Current Calibration ERROR | ADC phase W calibration status - 0 for not calibrated, 1 for calibration OK |
| 489 | MC Maximum Temperature ERROR | Maximum power module temperature exceeded - 0 for Error (exceeded), 1 for OK |
| 490 | MC Phase V ERROR | Power module driver phase error reported - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 491 | MC General Driver ERROR | Power module driver error reported - 0 for Error, 1 for OK |
| 492 | MC Over-current AC | Current AC side limit exceeded - 0 for Error (exceeded), 1 for OK |
| 493 | MC Over-voltage Advertisement | Over-voltage DC side limit advertisement exceeded - 0 for Error (exceeded), 1 for OK |
| 494 | MC Over-voltage Caution | Over-voltage DC side limit caution exceeded - 0 for Error (exceeded), 1 for OK |
| 495 | MC Under-voltage Latching | Critical under-voltage DC side limit violation - 0 for Error, 1 for OK |
| 496 | MC Under-voltage ON Latching | Non critical under-voltage DC side limit violation - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 497 | MC RMS imbalance | Current AC side imbalance - 0 for Error, 1 for OK |
| 498 | MC Open DC fault | Open-circuit DC side fault - 0 for Error, 1 for OK |
| 499 | MC Over-current DC | Current DC side limit exceeded - 0 for Error (exceeded), 1 for OK |
| 500 | Ground effect compensation variance disabled | Ground effect e2 compensation status - 0 for disabled, 1 for enabled |
| 501 | Ground effect compensation measurement disabled | Ground effect correction limit compensation status - 0 for disabled, 1 for enabled |
| 502 | No SRTM data | Bit to indicate if there is SRTM in the SD - 0 for No SRTM data, 1 for SRTM data OK |
| 503 | No geoid data | |

| ID | Name | Description |
|---|---|---|
| | | Bit to indicate if there is Geoid data in the SD - 0 for No geoid data, 1 for Geoid data OK |
| 600 | Wind Estimation OFF | 0 for disabled, 1 for enabled |
| 700-731 | Servo 0-31 Saturated | 0 for Saturated, 1 for OK |
| 732-733 | Phase U-W ERROR | 0 for Error, 1 for OK |
| 734 | HW DC link ERROR | 0 for Error, 1 for OK |
| 735-736 | HW over-current AC-DC ERROR | 0 for Error, 1 for OK |
| 737 | HW Ground Fault Detection ERROR | 0 for HW Ground Fault Detection Error, 1 for HW Ground Fault Detection OK |
| 738 | HW Power Regulator ERROR | 0 for HW Power Regulator error, 1 for HW Power Regulator OK |

| ID | Name | Description |
|---|---|---|
| 739 | HW trip PWM ERROR | 0 for HW general error to trip PWM, 1 for HW general to trip PWM OK |
| 740 | Disconnected HW battery | 0 for Battery disconnected, 1 for Battery connected |
| 741 | DC current calibration ERROR | 0 for DC current calibration Error, 1 for DC current calibration OK |
| 742 | Estimation Position (Deprecated) | Position estimation error (Deprecated sensor/logic status) |
| 743 | Speed reference tracking ERROR | Motor Controller failed to track the commanded speed reference |
| 744 | Position Sensor ERROR | A generic error was detected in the position sensor hardware |
| 745 | SIN/COS 1 Calibration ERROR | Calibration error detected for Sin/Cos sensor 1 |

| ID | Name | Description |
|---|---|---|
| 746 | SIN/COS 2 Calibration ERROR | Calibration error detected for Sin/Cos sensor 2 |
| 747 | Low energy ERROR | Input energy level is too low for reliable motor controller operation |
| 748 | RPM Command Message ERROR | Error detected in the RPM command message format or reception |
| 757 | A temperature sensor error was detected | An internal VMC temperature sensor has reported an out-of-range value |
| 758 | A MOSFET de-saturation was detected | Hardware protection triggered by MOSFET de-saturation (over-current) |
| 759 | MOSFET driver is not ready | The MOSFET power driver stage is not ready for operation |

| ID | Name | Description |
|---|---|---|
| 800-815 | PWM 0-15 GPIO OFF | PWM GPIO 0-15 communication State - 0 for Off, 1 for ON |
| 816-819 | EQEP_A-I (GPIO 17-20) OFF | Input/Output State - 0 for Off, 1 for ON |
| 820-822 | RSSI LED 0-2 OFF | Received Signal Strength Indicator led state - 0 for Off, 1 for ON |
| 823 | GPIO 5 (GPIO28) OFF | GPIO 5 Status (Low/High) - 0 for Off, 1 for ON |
| 824 | GPIO 6 (GPIO61) OFF | GPIO 6 Status (Low/High) - 0 for Off, 1 for ON |
| 825 | GPIO 7 (GPIO60) OFF | GPIO 7 Status (Low/High) - 0 for Off, 1 for ON |
| 826 | GPIO 8 (GPIO59) OFF | GPIO 8 Status (Low/High) - 0 for Off, 1 for ON |
| 827 | GPIO 9 (GPIO17) OFF | GPIO 9 Status (Low/High) - 0 for Off, 1 for ON |

| ID | Name | Description |
|---|---|---|
| 828 | GPIO 10 (GPIO58) OFF | GPIO 10 Status (Low/High) - 0 for Off, 1 for ON |
| 829 | GPIO 11 (GPIO16) OFF | GPIO 11 Status (Low/High) - 0 for Off, 1 for ON |
| 830 | GPIO 12 (GPIO53) OFF | GPIO 12 Status (Low/High) - 0 for Off, 1 for ON |
| 831 | GPIO 13 (GPIO20) OFF | GPIO 13 Status (Low/High) - 0 for Off, 1 for ON |
| 832 | GPIO 14 (GPIO23) OFF | GPIO 14 Status (Low/High) - 0 for Off, 1 for ON |
| 833 | GPIO 15 (GPIO51) OFF | GPIO 15 Status (Low/High) - 0 for Off, 1 for ON |
| 834 | GPIO 16 (GPIO52) OFF | GPIO 16 Status (Low/High) - 0 for Off, 1 for ON |
| 835 | GPIO 17 (GPIO49) OFF | GPIO 17 Status (Low/High) - 0 for Off, 1 for ON |
| 836 | GPIO 18 (GPIO08) OFF | |

| ID | Name | Description |
|---|---|---|
| | | GPIO 18 Status (Low/High) - 0 for Off, 1 for ON |
| 837 | GPIO 19 (GPIO11) OFF | GPIO 19 Status (Low/High) - 0 for Off, 1 for ON |
| 838 | GPIO 20 (GPIO10) OFF | GPIO 20 Status (Low/High) - 0 for Off, 1 for ON |
| 839 | GPIO 21 (GPIO09) OFF | GPIO 21 Status (Low/High) - 0 for Off, 1 for ON |
| 850 | GPIO A OFF | GPIO A Status (Low/High) - 0 for Off, 1 for ON |
| 851 | GPIO B OFF | GPIO B Status (Low/High) - 0 for Off, 1 for ON |
| 900-931 | Virtual GPIO 00-31 OFF | Virtual GPIO 00-31 Status (Low/High) - 0 for Off, 1 for ON |
| 1000-1009 | Simulation BIT 00-09 ERROR | 0 for Error, 1 for OK |
| 1010-1169 | Custom msg 0-159 Rx ERROR | Custom message timeout - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 1170-1171 | Entrance EKF GNSS1-2 OFF | GNSS 1-2 information considered in EKF Navigation - 0 for entrance EKF GNSS OFF, 1 for ON<br>EKF GNSS OFF may be because **Position not fixed** → EKF deactivated → INSS activated<br>**Important**<br>This bit is **not a FeedbackBit**, i.e., it will not be activated automatically by the system, but acts as a **user bit**. |
| 1172 | Entrance EKF GNSS EXT OFF | External GNSS information considered in EKF Navigation - 0 for entrance EKF GNSS EXT OFF, 1 for ON |

| ID | Name | Description |
|---|---|---|
| 1173 | Entrance EKF Internest OFF | Internest information considered in EKF Navigation - 0 for entrance EKF internest OFF, 1 for ON |
| 1174 | Entrance EKF GPSCOMPASS OFF | GNSS Compass information considered in EKF Navigation - 0 for entrance EKF GPSCOMPASS OFF, 1 for ON |
| 1175 | Entrance EKF Magnetometer OFF | Magnetometer information considered in EKF Navigation - 0 for entrance EKF magnetometer OFF, 1 for ON |
| 1176 | Entrance EKF Static press OFF | Static Pressure sensor information considered in EKF Navigation - 0 for entrance EKF static pressure OFF, 1 for ON |

| ID | Name | Description |
|---|---|---|
| 1177 | Entrance EKF Altimeter press OFF | Altimeter information considered in EKF Navigation - 0 for entrance EKF altimeter OFF, 1 for ON |
| 1178 | Entrance EKF Radar-altimeter press OFF | Radar Altimeter information considered in EKF Navigation - 0 for entrance EKF radar-altimeter OFF, 1 for ON |
| 1179 | Entrance EKF DEM OFF | DEM information considered in EKF Navigation - 0 for entrance EKF DEM OFF, 1 for ON |
| 1183 | External navigation sensor ERROR | External navigation sensor - 0 for Error, 1 for OK |
| 1184 | External IMU 0 accelerometer ERROR | External IMU 0 accelerometer - 0 for Error, 1 for OK |
| 1185 | External IMU 0 gyroscope ERROR | External IMU 0 gyroscope - 0 for Error, 1 for OK |

| ID | Name | Description |
|---|---|---|
| 1186 | External IMU 1 accelerometer ERROR | External IMU 1 accelerometer - 0 for Error, 1 for OK |
| 1187 | External IMU 1 gyroscope ERROR | External IMU 1 gyroscope - 0 for Error, 1 for OK |
| 1188 | External magnetometer 0 ERROR | External magnetometer 0 - 0 for Error, 1 for OK |
| 1189 | External magnetometer 1 ERROR | External magnetometer 1 - 0 for Error, 1 for OK |
| 1190-1191 | Sniffer msg 0-1 Rx ERROR | Sniffer receiver message 0-1 - 0 for Error, 1 for OK |
| 1200-2199 | User BIT 00-999 ERROR | User bit 0-999 - 0 for Error, 1 for OK |
| 2200 | BIT Dummy ERROR | Bit for configurable checks - 0 for Error, 1 for OK |

## Real Variables (RVar) - 32 Bits

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 0 | IAS (Indicated Airspeed) | m/s | Pitot-static measurement speed |
| 1 | TAS (True Airspeed) (Deprecated) | m/s | Speed relative to the airmass in which the vehicle is moving (IAS measurement corrected with Standard Atmosphere data)<br><br>**Warning**<br>**Deprecated variable** |
| 2 | GS (Ground Speed) | m/s | Horizontal speed, relative to the ground |
| 3 | Heading | rad | Direction in which the vehicle velocity vector is pointing |
| 4 | Flight Path Angle | rad | Angle between velocity vector and local horizontal line |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 5 | Bank | rad | Angle around the Longitudinal Euler axis |
| 6 | Yaw | rad | Angle around the Vertical Euler axis |
| 7 | Pitch | rad | Angle around the Transverse Euler axis |
| 8 | Roll | rad | Angle around the Longitudinal Euler axis |
| 9 | Route-Guidance Tangential deviation | m | Tangencial distance to the desired position (guidance) |
| 10 | Route-Guidance Horizontal deviation | m | Horizontal distance to the desired position (guidance) |
| 11 | Route-Guidance Perpendicular deviation | m | Perpendicular distance to the desired position (guidance) |
| 12 | p (Angular Velocity - X Body Axis) | rad/s | Angular velocity around longitudinal axis |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 13 | q (Angular Velocity - Y Body Axis) | rad/s | Angular velocity around lateral axis |
| 14 | r (Angular Velocity - Z Body Axis) | rad/s | Angular velocity around vertical axis |
| 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 Ground Velocity | m/s | GV vector X component |
| 20 | Lateral Ground Velocity | m/s | GV vector Y component |
| 21 | Velocity | m/s | Velocity vector module |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 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 | Estimated air density | $kg/m^3$ | Estimated air density at current altitude |
| 27 | Gravity at UAV's position | $m/s^2$ | Gravity at UAV's position positive down |
| 28 | Co-Yaw | rad | Acrobatic Yaw with Body Z' axis pointing to X |
| 29 | Co-Pitch | rad | Acrobatic Pitch with Body X' axis pointing to -Z |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 30 | Co-Roll | rad | Acrobatic Roll with Y' keeping same as Y |
| 31 | Angular Acceleration - X Body Axis | rad/s² | Acceleration around the longitudinal axis |
| 32 | Angular Acceleration - Y Body Axis | rad/s² | Acceleration around the lateral axis |
| 33 | Angular Acceleration - Z Body Axis | rad/s² | Acceleration around the vertical axis |
| 34 | Body to NED Quaternion qs | customType | First component of body to NED orientation quaternion |
| 35 | Body to NED Quaternion qi | customType | Second component of body to NED orientation quaternion |
| 36 | Body to NED Quaternion qj | customType | Third component of body to NED orientation quaternion |
| 37 | | customType | Fourth component of |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Body to NED Quaternion qk | | body to NED orientation quaternion |
| 40 | RSSI | percentage | Received Signal Strength Indicator<br>**Warning**<br>**Deprecated variable** |
| 42 | SCI-A Rx Rate (LTE/EXT. UART) | bytes/s | Reception rate (in bytes per second) of 4G (hwv < 4.7) or external UART (hwv >=4.7) communications |
| 43 | SCI-A Tx Rate (LTE/EXT. UART) | bytes/s | Transmission rate (in bytes per second) of 4G (hwv < 4.7) or external UART (hwv >=4.7) communications |
| 44 | SCI-B Rx Rate (LOS) | bytes/s | Radio link reception byte rate |
| 45 | | bytes/s | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | SCI-B Tx Rate (LOS) | | Radio link transmission byte rate |
| 46 | SCI-C Rx Rate (RS485) | bytes/s | RS485 communication reception byte rate |
| 47 | SCI-C Tx Rate (RS485) | bytes/s | RS485 communication transmission byte rate |
| 48 | SCI-D Rx Rate (RS232) | bytes/s | RS232 communication reception byte rate |
| 49 | SCI-D Tx Rate (RS232) | bytes/s | RS232 communication transmission byte rate |
| 50 | CAN-A Tx Rate | pkts/s | CAN-A transmission packet rate |
| 51 | CAN-B Tx Rate | pkts/s | CAN-B transmission packet rate |
| 52 | CAN-A Tx skip Rate | pkts/s | CAN-A messages delayed because |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | no mailbox is available for sending |
| 53 | CAN-B Tx skip Rate | pkts/s | CAN-B messages delayed because no mailbox is available for sending |
| 54 | CAN-FD-A Tx rate | pkts/s | Transmission frequency of messages for the CAN Flexible Data-rate A interface |
| 55 | CAN-FD-A Tx skip rate | pkts/s | Frequency of messages not sent on CAN-FD-A due to full buffers or priority |
| 56 | Yaw Rate | rad/s | Rate of change of the yaw angle |
| 57 | Pitch Rate | rad/s | Rate of change of the pitch angle |
| 58 | Roll Rate | rad/s | Rate of change of the roll angle |
| 59-64 | COM 0-5 Parse Error Rate | messages | Each COM discard packages with |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | these frequencies. Messages might be discarded because the calculated and the received CRC are different |
| 65 | GNSS Absolute Time Of Week Milliseconds as Unit32 | customType | Time of the week expressed with milliseconds |
| 66 | GNSS Hours in the Current Day | customType | Elapsed hours in the current day |
| 67 | GNSS Minutes in the Current Hour | customType | Elapsed minutes in the current hour |
| 68 | GNSS Seconds in the Current Minute | customType | Elapsed seconds in the current minute |
| 70 | CAN-A Rx rate | pkts/s | Reception frequency of incoming packets on the CAN-A bus |
| 71 | CAN-A Rx skip rate | pkts/s | Frequency of incoming CAN-A messages ignored |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | or dropped by the system |
| 72 | CAN-B Rx rate | pkts/s | Reception frequency of incoming packets on the CAN-B bus |
| 73 | CAN-B Rx skip rate | pkts/s | Frequency of incoming CAN-B messages ignored or dropped by the system |
| 74 | CAN-FD-A Rx rate | pkts/s | Reception frequency of packets for the CAN Flexible Data-rate A interface |
| 75 | CAN-FD-A Rx skip rate | pkts/s | Frequency of incoming CAN-FD-A messages ignored or dropped |
| 80 | Estimated gyro bias x | rad/s | Gyro bias estimated during IMU calibration |
| 81 | Estimated gyro bias y | rad/s | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | Gyro bias estimated during IMU calibration |
| 82 | Estimated gyro bias z | rad/s | Gyro bias estimated during IMU calibration |
| 83 | Estimated accelerometer bias x | m/s² | Accelerometer bias estimated during IMU calibration |
| 84 | Estimated accelerometer bias y | m/s² | Accelerometer bias estimated during IMU calibration |
| 85 | Estimated accelerometer bias z | m/s² | Accelerometer bias estimated during IMU calibration |
| 90 | SCI-E Rx Rate (LTE) | bytes/s | Reception rate (in bytes per seconds) of the first channel (port0) of the SCI expander (hwv >=4.7) |
| 91 | SCI-E Tx Rate (LTE) | bytes/s | Transmission rate (in bytes per seconds) of the |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | first channel (port0) of the SCI expander (hwv >=4.7) |
| 92 | SCI-F Rx Rate (LTE Aux.) | bytes/s | Reception rate (in bytes per seconds) of the second channel (port1) of the SCI expander (hwv >=4.7) |
| 93 | SCI-F Tx Rate (LTE Aux.) | bytes/s | Transmission rate (in bytes per seconds) of the second channel (port1) of the SCI expander (hwv >=4.7) |
| 100 | Desired IAS (Indicated Airspeed) | m/s | Commanded IAS from guidance |
| 101 | Desired TAS (True Airspeed) | m/s | Commanded TAS from guidance |
| 102 | Desired GS (Ground Speed) | m/s | Commanded GS from guidance |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 103 | Desired Heading | rad | Commanded Heading from guidance |
| 104 | Desired Flight Path Angle | rad | Commanded Flight Path Angle from guidance |
| 105 | Desired Bank | rad | Commanded Bank from guidance |
| 106 | Desired Yaw | rad | Commanded Yaw from guidance |
| 107 | Desired Pitch | rad | Commanded Pitch from guidance |
| 108 | Desired Roll | rad | Commanded Roll from guidance |
| 112 | Desired p (Angular Velocity - X Body Axis) | rad/s | Commanded angular velocity around longitudinal axis |
| 113 | Desired q (Angular Velocity - Y Body Axis) | rad/s | Commanded angular velocity around lateral axis |
| 114 | Desired r (Angular | rad/s | Commanded angular velocity |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Velocity - Z Body Axis) | | around vertical axis |
| 115 | Desired Forward Acceleration - X Body Axis | m/s² | Commanded Forward Acceleration from guidance |
| 116 | Desired Right Acceleration - Y Body Axis | m/s² | Commanded Right Acceleration from guidance |
| 117 | Desired Bottom Acceleration - Z Body Axis | m/s² | Commanded Bottom Acceleration from guidance |
| 118 | Desired RPM | rad/s | Commanded RPM from guidance |
| 119 | Desired Front Ground Velocity | m/s | Commanded Front GV from guidance |
| 120 | Desired Lateral Ground Velocity | m/s | Commanded Lateral GV from guidance |
| 121 | Desired Velocity | m/s | Commanded Velocity from guidance |
| 122 | Desired Forward Load | customType | Commanded Forward Load |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Factor - X Body Axis | | 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 Factor from guidance |
| 125 | Desired Tangential Acceleration | $m/s^2$ | Commanded Tangential Acceleration from guidance |
| 126 | Energy Rate Error | customType | Rate of change of the Total System Energy |
| 127 | Energy Distribution Error | customType | Distribution of system energy between kinetical and geopotential energy |
| 128 | Desired Co-Yaw | rad | Commanded co-yaw from guidance |
| 129 | Desired Co-Pitch | rad | Commanded co-pitch from guidance |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 130 | Desired Co-Roll | rad | Commanded co-roll from guidance |
| 134 | Desired Aerodynamic Heading | rad | Target heading relative to the airflow commanded by the guidance system |
| 135 | Aerodynamic Heading | rad | Current orientation of the vehicle relative to the wind direction |
| 136 | Aerodynamic Heading Error | rad | Difference between the desired and actual aerodynamic heading |
| 137 | Desired Aerodynamic Flight Path Angle | rad | Target vertical angle of the flight path relative to the airmass |
| 138 | Aerodynamic Flight Path Angle | rad | Actual vertical angle of the velocity vector relative to the airmass |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 139 | Aerodynamic Flight Path Angle Error | rad | Difference between the commanded and actual aerodynamic flight path angle |
| 140 | Climbing Initial Heading | rad | Heading in climbing phase (start of the route) |
| 141 | Approach Initial Heading | rad | Heading in approach phase (end of the route) |
| 142 | Headwind Direction | rad | Wind direction estimation |
| 143 | Tailwind Direction | rad | Angle of the vector that would correspond to the opposite of the Headwind vector |
| 144 | Runway Direction | rad | Runway angle |
| 145 | Elevation of current route | rad | Elevation of tangent to current route at its closest point to desired position |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 146 | Azimuth of current route | rad | Azimuth of tangent to current route at its closest point to desired position |
| 147 | Distance to closest obstacle | m | Signed distance to closest obstacle (negative means inside) |
| 148 | Distance of obstacle repulsion | m | Distance at which the obstacles have an effect in the guidance |
| 200 | Desired North Ground Velocity | m/s | Commanded North (NED Coordinates system) GV from guidance |
| 201 | Desired East Ground Velocity | m/s | Commanded East (NED Coordinates system) GV from guidance |
| 202 | Desired Down Ground Velocity | m/s | Commanded Down (NED Coordinates system) GV from guidance |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 203 | Desired 2D MSL (Height Above Mean Sea Level) | m | Commanded MSL from guidance in 2D height mode |
| 204 | Desired 2D AGL (Above Ground Level) - Height | m | Commanded AGL from guidance in 2D height mode |
| 205 | Desired 2D WGS84 Elevation (Height Over The Ellipsoid) | m | Commanded WGS84 Elevation from guidance in 2D height mode |
| 206 | Desired Longitude | rad | Commanded Longitude from guidance |
| 207 | Desired Latitude | rad | 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 | m | Commanded MSL Altitude from guidance |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Level) - Altitude | | |
| 210 | Desired AGL (Above Ground Level) - Height | m | Commanded AGL Altitude from guidance |
| 211 | Time to Waypoint | seg | Estimated time remaining to reach the next active waypoint |
| 212 | Expected North Magnetic Field | T | Predicted North component of the Earth's magnetic field at current UAV position |
| 213 | Expected East Magnetic Field | T | Predicted East component of the Earth's magnetic field at current UAV position |
| 214 | Expected Down Magnetic Field | T | Predicted vertical (Down) component of the Earth's magnetic field |
| 250 | | m | Difference from Desired and |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Guidance North Position Error | | 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 velocity |
| 254 | Guidance PID East Desired Velocity | m/s | Difference from Desired and actual PID east velocity |
| 255 | Guidance PID Down Desired Velocity | m/s | Difference from Desired and actual PID down velocity |
| 256 | Desired Velocity X Body Axis | m/s | Commanded velocity in X-axis from guidance |
| | | m/s | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 257 | Desired Velocity Y Body Axis | | 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 | rad | Yaw from external navigation source |
| 260 | External Pitch | rad | Pitch from external navigation source |
| 261 | External Roll | rad | 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 |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 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 source |
| 269 | External Acceleration y Body Axis | $m/s^2$ | Acceleration y body axis from external navigation source |
| 270 | External Acceleration z Body Axis | $m/s^2$ | Acceleration z body axis from external navigation source |
| 271 | External GNSS Time of Week | s | GNSS Time of week from external navigation source |
| 299 | Target Acquisition Period | s | Expected period of the acquisition (CIO-HI) task. It complements the **RVar 2047** |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 300 | Relative Timestamp | s | Time spent since power-on of the system |
| 301 | Used Memory Space (Deprecated) | byte | SD used memory space<br><br>**Warning**<br>**Deprecated variable** |
| 302 | Free Memory Space (Deprecated) | byte | SD free memory space<br><br>**Warning**<br>**Deprecated variable** |
| 303 | Dynamic Pressure (Deprecated) | Pa | Physical measurement from Pitot (dynamic pressure)<br><br>**Warning**<br>**Deprecated variable** |
| 304 | Static Pressure (Deprecated) | Pa | Physical measurement from Pitot (static |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | preassure)<br><br>**Warning**<br>**Deprecated variable** |
| 305 | Internal Temperature | K | Physical measurement from internal sensors |
| 307 | Accelerometer - X Body Axis | m/s² | Accelerometer measurement for X axis |
| 308 | Accelerometer - Y Body Axis | m/s² | Accelerometer measurement for Y axis |
| 309 | Accelerometer - Z Body Axis | m/s² | Accelerometer measurement for Z axis |
| 310 | Gyroscope - X Body Axis | rad/s | Gyroscope measurement for X axis |
| 311 | Gyroscope - Y Body Axis | rad/s | Gyroscope measurement for Y axis |
| 312 | Gyroscope - Z Body Axis | rad/s | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | Gyroscope measurement for Z axis |
| 313 | Magnetometer - X Body Axis | T | Magnetometer measurement for X axis<br>**Warning**<br>**Deprecated variable** |
| 314 | Magnetometer - Y Body Axis | T | Magnetometer measurement for Y axis<br>**Warning**<br>**Deprecated variable** |
| 315 | Magnetometer - Z Body Axis | T | Magnetometer measurement for Z axis<br>**Warning**<br>**Deprecated variable** |
| 317 | ADC 0 | volt | Voltage measured at the Analog-to- |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | Digital Converter input channel 0 |
| 318 | ADC 1 | volt | Voltage measured at the Analog-to-Digital Converter input channel 1 |
| 319 | ADC 2 | volt | Voltage measured at the Analog-to-Digital Converter input channel 2 |
| 320 | ADC 3 | volt | Voltage measured at the Analog-to-Digital Converter input channel 3 |
| 321 | ADC 4 | volt | Voltage measured at the Analog-to-Digital Converter input channel 4 |
| 322 | Magnetometer 0 Raw X in SI | T | **Internal 0 Magnetometer** raw measurement for X axis |
| 323 | Magnetometer 0 Raw Y in SI | T | **Internal 0 Magnetometer** raw measurement for Y axis |
| 324 | | T | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Magnetometer 0 Raw Z in SI | | **Internal 0 Magnetometer** raw measurement for Z axis |
| 325 | Magnetometer 0 Temperature | K | **Internal 0 Magnetometer** temperature |
| 326 | External LIS3MDL Magnetometer Raw X in SI | T | **External LIS3MDL Magnetometer** raw measurement for X axis |
| 327 | External LIS3MDL Magnetometer Raw Y in SI | T | **External LIS3MDL Magnetometer** raw measurement for Y axis |
| 328 | External LIS3MDL magnetometer raw Z in SI | T | **External LIS3MDL Magnetometer** raw measurement for Z axis |
| 329 | External LIS3MDL magnetometer temperature | K | **External LIS3MDL Magnetometer** temperature |
| 330 | IMU 0 Raw Accelerometer | $m/s^2$ | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | X Measurement | | **IMU 0** raw accelerometer x measurement |
| 331 | IMU 0 Raw Accelerometer Y Measurement | m/s² | **IMU 0** raw accelerometer y measurement |
| 332 | IMU 0 Raw Accelerometer Z Measurement | m/s² | **IMU 0** raw accelerometer z measurement |
| 333 | IMU 0 Raw Gyroscope X Measurement | rad/s | **IMU 0** raw gyroscope x measurement |
| 334 | IMU 0 Raw Gyroscope Y Measurement | rad/s | **IMU 0** raw gyroscope y measurement |
| 335 | IMU 0 Raw Gyroscope Z Measurement | rad/s | **IMU 0** raw gyroscope z measurement |
| 336 | IMU 0 temperature measurement | K | **IMU 0** temperature measurement |
| 337 | IMU 1 Raw Accelerometer X Measurement | m/s² | **IMU 1** raw accelerometer x measurement |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 338 | IMU 1 Raw Accelerometer Y Measurement | m/s² | **IMU 1** raw accelerometer y measurement |
| 339 | IMU 1 Raw Accelerometer Z Measurement | m/s² | **IMU 1** raw accelerometer z measurement |
| 340 | IMU 1 Raw Gyroscope X Measurement | rad/s | **IMU 1** raw gyroscope x measurement |
| 341 | IMU 1 Raw Gyroscope Y Measurement | rad/s | **IMU 1** raw gyroscope y measurement |
| 342 | IMU 1 Raw Gyroscope Z Measurement | rad/s | **IMU 1** raw gyroscope z measurement |
| 343 | IMU 1 temperature Measurement | K | **IMU 1** temperature measurement |
| 344 | Static Pressure Sensor 1 Raw Measurement | Pa | Static pressure sensor 1 raw measurement |
| 345 | Static Pressure Sensor 1 Temperature | K | Static pressure sensor 1 temperature |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 346 | Dynamic Pressure Sensor Raw Measurement | Pa | Dynamic pressure sensor 1 raw measurement |
| 347 | Dynamic Pressure Sensor Temperature | K | Dynamic pressure sensor 1 temperature |
| 348 | Static Pressure Sensor 0 Raw Measurement | Pa | Static pressure sensor 0 raw measurement |
| 349 | Static Pressure Sensor 0 Temperature | K | Static pressure sensor 0 temperature |
| 350 | Vectornav Message Frequency | Hz | External navigation source VectorNav sends messages with this frequency |
| 351 | Vectornav Raw Acc X Measurement | $m/s^2$ | Raw accelerometer X measurement from external navigation source VectorNav |
| 352 | | $m/s^2$ | Raw accelerometer Y |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Vectornav Raw Acc Y Measurement | | measurement from external navigation source VectorNav |
| 353 | Vectornav Raw Acc Z Measurement | m/s² | Raw accelerometer Z measurement from external navigation source VectorNav |
| 354 | Vectornav Raw Gyr X Measurement | rad/s | Raw gyroscope X measurement from external navigation source VectorNav |
| 355 | Vectornav Raw Gyr Y Measurement | rad/s | Raw gyroscope Y measurement from external navigation source VectorNav |
| 356 | Vectornav Raw Gyr Z Measurement | rad/s | Raw gyroscope Z measurement from external navigation source VectorNav |
| 357 | External HSCDTD008A | T | **External HSCDTD008A Magnetometer** |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Magnetometer Raw X in SI | | raw measurement for X axis |
| 358 | External HSCDTD008A Magnetometer Raw Y in SI | T | **External HSCDTD008A Magnetometer** raw measurement for Y axis |
| 359 | External HSCDTD008A Magnetometer Raw Z in SI | T | **External HSCDTD008A Magnetometer** raw measurement for Z axis |
| 360 | External HSCDTD008A Magnetometer Temperature | K | **External HSCDTD008A Magnetometer** temperature |
| 361 | IMU 2 Raw Accelerometer X Measurement | m/s² | **IMU 2** raw accelerometer x measurement |
| 362 | IMU 2 Raw Accelerometer Y Measurement | m/s² | **IMU 2** raw accelerometer y measurement |
| 363 | IMU 2 Raw Accelerometer | m/s² | **IMU 2** raw accelerometer z measurement |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Z Measurement | | |
| 364 | IMU 2 Raw Gyroscope X Measurement | rad/s | **IMU 2** raw gyroscope x measurement |
| 365 | IMU 2 Raw Gyroscope Y Measurement | rad/s | **IMU 2** raw gyroscope y measurement |
| 366 | IMU 2 Raw Gyroscope Z Measurement | rad/s | **IMU 2** raw gyroscope z measurement |
| 367 | IMU 2 Temperature Measurement | K | **IMU 2** temperature measurement |
| 368 | Static Pressure Sensor 2 Raw Measurement | Pa | Static pressure sensor 2 raw measurement |
| 369 | Static Pressure Sensor 2 Temperature | K | Static pressure sensor 2 temperature |
| 370 | Magnetometer 1 Raw Measure X Converted to SI | T | **Internal 1 Magnetometer** raw measurement for X axis converted to SI |
| 371 | | T | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Magnetometer 1 Raw Measure Y Converted to SI | | **Internal 1 Magnetometer** raw measurement for Y axis converted to SI |
| 372 | Magnetometer 1 Raw Measure Z Converted to SI | T | **Internal 1 Magnetometer** raw measurement for Z axis converted to SI |
| 373 | Magnetometer 1 Temperature | K | **Internal 1 Magnetometer** temperature |
| 374 | External Magnetometer MMC5883MA Raw Measure X Converted to SI | T | **External MMC5883MA Magnetometer** raw measurement for X axis converted to SI |
| 375 | External Magnetometer MMC5883MA Raw Measure Y Converted to SI | T | **External MMC5883MA Magnetometer** raw measurement for Y axis converted to SI |
| 376 | External Magnetometer MMC5883MA Raw Measure | T | **External MMC5883MA Magnetometer** raw measurement |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Z Converted to SI | | for Z axis converted to SI |
| 377 | External Magnetometer MMC5883MA Temperature | K | **External MMC5883MA Magnetometer** temperature |
| 378 | External Magnetometer RM3100 Raw Measure X Converted to SI | T | **External RM3100 Magnetometer** raw measurement for X axis converted to SI |
| 379 | External Magnetometer RM3100 Raw Measure Y Converted to SI | T | **External RM3100 Magnetometer** raw measurement for Y axis converted to SI |
| 380 | External Magnetometer RM3100 Raw Measure Z Converted to SI | T | **External RM3100 Magnetometer** raw measurement for Z axis converted to SI |
| 386 | IMU 3 Raw Accelerometer X Measurement | $m/s^2$ | **IMU 3** raw accelerometer x measurement |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 387 | IMU 3 Raw Accelerometer Y Measurement | m/s² | **IMU 3** raw accelerometer y measurement |
| 388 | IMU 3 Raw Accelerometer Z Measurement | m/s² | **IMU 3** raw accelerometer z measurement |
| 389 | IMU 3 Raw Gyroscope X Measurement | rad/s | **IMU 3** raw gyroscope x measurement |
| 390 | IMU 3 Raw Gyroscope Y Measurement | rad/s | **IMU 3** raw gyroscope y measurement |
| 391 | IMU 3 Raw Gyroscope Z Measurement | rad/s | **IMU 3** raw gyroscope z measurement |
| 392 | IMU 3 Temperature Measurement | K | **IMU 3** temperature measurement |
| 393 | Magnetometer 2 Raw Measure X Converted to SI | T | **Internal 2 Magnetometer** raw measurement for X axis converted to SI |
| 394 | Magnetometer 2 Raw | T | **Internal 2 Magnetometer** |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Measure Y Converted to SI | | raw measurement for Y axis converted to SI |
| 395 | Magnetometer 2 Raw Measure Z Converted to SI | T | **Internal 2 Magnetometer** raw measurement for Z axis converted to SI |
| 400 | Power Input | V | Voltage received by Veronte |
| 401 | Power Comicro 3.3V | V | Voltage received by Veronte through 3.3V port |
| 402 | Power 5V | V | Voltage received by Veronte through 5V port |
| 403 | SUC Power Input | V | Voltage received by Veronte SUC |
| 404 | Power 3.6V | V | Voltage received by Veronte through 3.6V port |
| 405 | CPU Temperature | K | Internal computer temperature |
| 406 | External IMU 0 raw accelerometer | $m/s^2$ | Saves the measurements of this external |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | x measurement | | sensor in raw form, as received from the custom messages |
| 407 | External IMU 0 raw accelerometer y measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 408 | External IMU 0 raw accelerometer z measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 409 | External IMU 0 raw gyroscope x measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 410 | External IMU 0 raw gyroscope y measurement | m/s² | Saves the measurements of this external sensor in raw |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | form, as received from the custom messages |
| 411 | External IMU 0 raw gyroscope z measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 412 | External IMU 0 temperature measurement | K | Saves the temperatures of this external sensor in raw form, as received from custom messages |
| 413 | External IMU 1 raw accelerometer x measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 414 | External IMU 1 raw accelerometer y measurement | m/s² | Saves the measurements of this external sensor in raw form, as received |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | from the custom messages |
| 415 | External IMU 1 raw accelerometer z measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 416 | External IMU 1 raw gyroscope x measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 417 | External IMU 1 raw gyroscope y measurement | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 418 | External IMU 1 raw gyroscope z measurement | m/s² | Saves the measurements of this external sensor in raw form, as received |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | from the custom messages |
| 419 | External IMU 1 temperature measurement | K | Saves the temperatures of this external sensor in raw form, as received from custom messages |
| 420 | External magnetometer 0 raw measurement X | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 421 | External magnetometer 0 raw measurement Y | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 422 | External magnetometer 0 raw measurement Z | m/s² | Saves the measurements of this external sensor in raw form, as received |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | from the custom messages |
| 423 | External magnetometer 0 temperature | K | Saves the temperatures of this external sensor in raw form, as received from custom messages |
| 424 | External magnetometer 1 raw measurement X | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 425 | External magnetometer 1 raw measurement Y | m/s² | Saves the measurements of this external sensor in raw form, as received from the custom messages |
| 426 | External magnetometer 1 raw measurement Z | m/s² | Saves the measurements of this external sensor in raw form, as received |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | from the custom messages |
| 427 | External magnetometer 1 temperature | K | Saves the temperatures of this external sensor in raw form, as received from custom messages |
| 428 | External navigation yaw | rad | External navigation yaw |
| 429 | External navigation pitch | rad | External navigation pitch |
| 430 | External navigation roll | rad | External navigation roll |
| 431 | External navigation p | rad/s | External navigation p (angular speed in x body axis) |
| 432 | External navigation q | rad/s | External navigation q (angular speed in y body axis) |
| 433 | External navigation r | rad/s | External navigation r |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | (angular speed in z body axis) |
| 434 | External navigation north velocity | m/s | External navigation north velocity |
| 435 | External navigation east velocity | m/s | External navigation east velocity |
| 436 | External navigation down velocity | m/s | External navigation down velocity |
| 437 | External navigation x-body acceleration | $m/s^2$ | External navigation x-body acceleration |
| 438 | External navigation y-body acceleration | $m/s^2$ | External navigation y-body acceleration |
| 439 | External navigation z-body acceleration | $m/s^2$ | External navigation z-body acceleration |
| 500 | Longitude | rad | East-West geographic coordinate |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 501 | Latitude | rad | 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 sensors or own models with considerable error |
| 505 | North Ground Velocity | m/s | Ground Velocity component in the North direction (NED Coordinates system) |
| 506 | East Ground Velocity | m/s | Ground Velocity component in the East direction (NED Coordinates system) |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 507 | Down Ground Velocity | m/s | Ground Velocity component in the resultant axis from North-East (NED Coordinates system) |
| 508 | Sensor IAS (Indicated Air Speed) | m/s | Pitot-static measurement speed |
| 509 | Angle of Attack - AoA | rad | Angle between reference body line and flow direction vector |
| 510 | Sideslip | rad | Angle between the flow direction vector and the longitudinal axis of the vehicle |
| 511 | GNSS1 MSL | m | Mean sea level (MSL) measurement provided by GPS 1 |
| 512 | GNSS1 AGL | m | Above ground level (AGL level) measurement provided by GPS 1 |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 513 | GNSS2 MSL | m | Mean sea level (MSL) measurement provided by GPS 2 |
| 514 | GNSS2 AGL | m | Above ground level (AGL level) measurement provided by GPS 2 |
| 551 | Sagetech MXS - Longitude decimal part | degree | Sagetech variable, used by block to parse variables for GPS Navigation Data Message<br>**Warning**<br>**Variable for internal use** |
| 552 | Sagetech MXS - Longitude fractional part | degree | Sagetech variable, used by block to parse variables for GPS Navigation Data Message |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | **Warning** **Variable for internal use** |
| 553 | Sagetech MXS - Latitude decimal part | degree | Sagetech variable, used by block to parse variables for GPS Navigation Data Message<br><br>**Warning** **Variable for internal use** |
| 554 | Sagetech MXS - Latitude fractional part | degree | Sagetech variable, used by block to parse variables for GPS Navigation Data Message<br><br>**Warning** **Variable for internal use** |
| 555 | Sagetech MXS - Ground speed | m/s | Sagetech variable, used by block to parse variables for GPS Navigation Data |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | Message<br><br>**Warning**<br>**Variable for internal use** |
| 556 | Sagetech MXS - Ground track | degree | Sagetech variable, used by block to parse variables for GPS Navigation Data Message<br><br>**Warning**<br>**Variable for internal use** |
| 560 | ADS-B Out / Squawk | - | ADS-B Squawk code, 4 digits that allow the operator to inform about its status |
| 600-603 | Temperature 0-3 | K | Variables to be configured with external temperature sensors |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | **Warning** **Deprecated variables** |
| 610 | North Position EKF Variance | $m^2$ | North position Extended Kalman Filter variance |
| 611 | East Position EKF Variance | $m^2$ | East position Extended Kalman Filter variance |
| 612 | Down Position EKF Variance | $m^2$ | Position variance component in the resultant axis from North-East |
| 613 | North Velocity EKF Variance | $m^2/s^2$ | North velocity Extended Kalman Filter variance |
| 614 | East Velocity EKF Variance | $m^2/s^2$ | East velocity Extended Kalman Filter variance |
| 615 | Down Velocity EKF Variance | $m^2/s^2$ | Velocity variance component in the resultant axis from North-East |
| 650 | Gimbal Command Yaw | customType | Yaw sent to the gimbal |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 651 | Gimbal Command Pitch | customType | Pitch sent to the gimbal |
| 652 | Gimbal Stick Yaw | customType | Yaw received from the joystick controlling the gimbal |
| 653 | Gimbal Stick Pitch | customType | Pitch received from the joystick controlling the gimbal |
| 654 | Gimbal Pitch Correction 0 | customType | Correction calculated by the gimbal for the pitch control 0 |
| 655 | Gimbal Pitch Correction 1 | customType | Correction calculated by the gimbal for the pitch control 1 |
| 656 | Gimbal Old Joint 0 | customType | Auxiliar variable 0 for Gimbal control configuration |
| 657 | Gimbal Old Joint 1 | customType | Auxiliar variable 1 for Gimbal control configuration |
| | | customType | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 658 | Cos (Gimbal Yaw) | | Auxiliar variable 0 for Gimbal control configuration |
| 659 | Sin (Gimbal Yaw) | customType | Auxiliar variable 1 for Gimbal control configuration |
| 660 | Gimbal Yaw Radian | customType | Auxiliar variable for Gimbal control configuration |
| 661 | Veronte Gimbal Yaw Output | customType | Yaw value the gimbal is sending as output |
| 662 | Veronte Gimbal Pitch Output | customType | Pitch value the gimbal is sending as output |
| 663 | Gimbal Phi(z) | customType | Auxiliar variable phi for Gimbal control configuration |
| 664 | Gimbal Theta(y) | customType | Auxiliar variable theta for Gimbal control configuration |
| 665 | Gimbal Psi(x) | customType | Auxiliar variable psi for Gimbal control configuration |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 666 | Veronte Gimbal Roll Output (Degrees) | customType | Roll value the gimbal is sending as output |
| 700-705 | RPM 0-5 | rad/s | Angular speed associated to pulse captured 0-5 |
| 750 | Selected Controller Time Step | s | 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 |
| 753 | Selected Controller Derivative Action | customType | PID selected derivative action |
| 754 | Selected Controller Integral Input | customType | PID selected integral input |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 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 |
| 800-823 | PWM 0-23 | customType | Pulse Width Modulation signal 0-23 |
| 900-915 | Stick Input r0 - r15 | customType | Raw stick measurement r0-r15 |
| 950-981 | Stick Input s0 - s31 | customType | **Warning** **Deprecated variables** |
| 1000-1031 | Stick Input y0 - y31 | customType | Servo position commanded stick y0-y31 |
| 1050-1069 | Control Output u0-u19 Before | customType | Commanded control output |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Servo Saturation | | before saturation correction |
| 1100-1104 | Lidar 0-4 Distance | m | Variable configurable for Lidar distances 0-4 |
| 1105-1109 | External Range Sensor 0-4 Measure | m | Variable configurable for external range sensors |
| 1200 | Route-Guidance Distance | m | Shortest distance to desired path (perpendicular distance) |
| 1201 | Radar AGL (Above Ground Level) - Height | m | Radar altimeter measure |
| 1202 | Radar Speed Down | m/s | Radar speed |
| 1203 | External Rotation for Follow Route | rad | Relative vector rotation when using Follow Route |
| 1204 | Time to Impact with Obstacles | s | Time calculated with Distance to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | Obstacle and travel speed |
| 1300-1309 | Timer 0-9 | s | Configurable timers for automations |
| 1320-1321 | CEX/MEX ADC 3.3V Input 0-1 | V | CEX/MEX ADC 3.3 V inputs 0 and 1 |
| 1322-1323 | CEX/MEX ADC 5.0V Input 0-1 | V | CEX/MEX ADC 5.0 V inputs 0 and 1 |
| 1324-1325 | CEX/MEX ADC 12.0V Input 0-1 | V | CEX/MEX ADC 12.0 V inputs 0 and 1 |
| 1326-1327 | CEX/MEX ADC 36.0V Input 0-1 | V | CEX/MEX ADC 36.0 V inputs 0 and 1 |
| 1328-1329 | CEX/MEX ADC vIn 0-1 | V | CEX/MEX External power supplies 0 and 1 |
| 1330 | CEX/MEX PCB Temperature | K | CEX/MEX PCB Temperature (from ADC input) |
| 1331 | CEX/MEX ADC HW Version | V | Hardware version of CEX/MEX ADC |
| 1350-1369 | 4X Real variables | - | For more information, |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | check **Real Variables - 4x Software Manual** |
| 1400 | Velocity - X Body Axis | m/s | Velocity on X-axis |
| 1401 | Velocity - Y Body Axis | m/s | Velocity on Y-axis |
| 1402 | Velocity - Z Body Axis | m/s | Velocity on Z-axis |
| 1403 | Estimated Dynamic Pressure | Pa | Dynamic pressure sensor raw measurement |
| 1404 | Barometric Pressure at Sea Level (QNH) | Pa | Introduced value for QNH |
| 1450-1453 | Captured Pulse 0-3 | customType | Input values from pulses |
| 1483 | External navigation sensor frequency | Hz | External navigation sensor frequency |
| 1484 | External IMU 0 accelerometer | Hz | Reception frequencies of measurements |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | reception frequency | | from External IMU 0 accelerometer |
| 1485 | External IMU 0 gyroscope reception frequency | Hz | Reception frequencies of measurements from External IMU 0 gyroscope |
| 1486 | External IMU 1 accelerometer reception frequency | Hz | Reception frequencies of measurements from External IMU 1 accelerometer |
| 1487 | External IMU 1 gyroscope reception frequency | Hz | Reception frequencies of measurements from External IMU 1 gyroscope |
| 1488-1489 | External magnetometer 0-1 reception frequency | Hz | Reception frequencies of measurements from External magnetometer 0-1 |
| 1490 | Internest Raw X Distance | m | Raw measurements for X-axis internest distance |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1491 | Internest Raw Y Distance | m | Raw measurements for Y-axis internest distance |
| 1492 | Internest Raw Z Distance | m | Raw measurements for Z-axis internest distance |
| 1493 | Internest Raw Angle | rad | Raw measurements for internest angle |
| 1494 | Internest Raw XY standard Deviation | m | Raw measurements for XY axis internest standard deviation |
| 1495 | Internest Raw Z standard Deviation | m | Raw measurements for Z-axis internest standard deviation |
| 1496 | Internest Raw Angle standard Deviation | rad | Raw measurements for internest |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | angle standard deviation |
| 1497 | Internest Position Update Frequency | Hz | Frequency to update internest position |
| 1500 | GNSS1 Absolute Time of Week | s | Data from GNSS1 module: Time of week |
| 1501 | GNSS1 ECEF Position X | m | Data from GNSS1 module: ECEF (Earth-Centered Earth-Fixed coordinate system) X position |
| 1502 | GNSS1 ECEF Position Y | m | Data from GNSS1 module: ECEF (Earth-Centered Earth-Fixed coordinate system) Y position |
| 1503 | GNSS1 ECEF Position Z | m | Data from GNSS1 module: ECEF (Earth-Centered Earth-Fixed coordinate |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | system) Z position |
| 1504 | GNSS1 Longitude | rad | Data from GNSS1 module: Longitude |
| 1505 | GNSS1 Latitude | rad | Data from GNSS1 module: Latitude |
| 1506 | GNSS1 Height Above Ellipsoid (WGS84) | m | Data from GNSS1 module: Height Above Ellipsoid (WGS84) |
| 1509 | GNSS1 PDOP (Dilution of Precision of Position) | customType | Data from GNSS1 module: PDOP - Relation between user position error and satellite position error |
| 1510 | GNSS1 Accuracy | m | Data from GNSS1 module: Accuracy |
| 1511 | GNSS1 Horizontal Accuracy Estimate | m | Data from GNSS1 module: Horizontal accuracy |
| 1512 | GNSS1 Vertical Accuracy Estimate | m | Data from GNSS1 module: Vertical accuracy |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1513 | GNSS1 Velocity North | m/s | Data from GNSS1 module: Velocity in North direction (NED Coordinates system) |
| 1514 | GNSS1 Velocity East | m/s | Data from GNSS1 module: Velocity in East direction (NED Coordinates system) |
| 1515 | GNSS1 Velocity Down | m/s | Data from GNSS1 module: Velocity in Down direction (NED Coordinates system) |
| 1516 | GNSS1 Speed Accuracy Estimate | m/s | Data from GNSS1 module: Speed accuracy |
| 1517 | GNSS1 Related Base Longitude | rad | Data from GNSS1 module: RTK Base longitude |
| 1518 | GNSS1 Related Base Latitude | rad | Data from GNSS1 module: RTK Base latitude |
| 1519 | GNSS1 Related Base | m | Data from GNSS1 module: RTK Base WGS84 altitude |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | WGS84 Altitude | | |
| 1520 | GNSS1 Related Base to Rover Azimuth | rad | Data from GNSS1 module: RTK Base-Rover vector azimuth (Spherical coordinates system) |
| 1521 | GNSS1 Related Base to Rover Elevation | rad | Data from GNSS1 module: RTK Base-Rover vector elevation (Spherical coordinates system) |
| 1522 | GNSS1 Related Base to Rover Distance | m | Data from GNSS1 module: RTK Base-Rover vector distance (Spherical coordinates system) |
| 1523 | GNSS1 Related Base to Rover Accuracy | m | Data from GNSS1 module: RTK Base-Rover vector accuracy |
| 1524 | | m | Data from GNSS1 module: RTK Base |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | GNSS1 Survey in Accuracy | | accuracy when base knows it is fixed in a particular position |
| 1525 | GNSS1 Related Base to Rover North | m | Data from GNSS1 module: RTK Base-Rover vector North (NED Coordinate system) |
| 1526 | GNSS1 Related Base to Rover East | m | Data from GNSS1 module: RTK Base-Rover vector East (NED Coordinate system) |
| 1527 | GNSS1 Related Base to Rover Down | m | Data from GNSS1 module: RTK Base-Rover vector Down (NED Coordinate system) |
| 1528 | GNSS1 Position Frequency | Hz | Data from GNSS1 module: Position frequency |
| 1529 | GNSS1 Jamming Indicator | % | Jamming indicator from U-Blox device 1 for GNSS |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1600 | GNSS2 Absolute Time of Week | s | Data from GNSS2 module: Time of week |
| 1601 | GNSS2 ECEF Position X | m | Data from GNSS2 module: ECEF (Earth-Centered Earth-Fixed coordinate system) X position |
| 1602 | GNSS2 ECEF Position Y | m | Data from GNSS2 module: ECEF (Earth-Centered Earth-Fixed coordinate system) Y position |
| 1603 | GNSS2 ECEF Position Z | m | Data from GNSS2 module: ECEF (Earth-Centered Earth-Fixed coordinate system) Z position |
| 1604 | GNSS2 Longitude | rad | Data from GNSS2 module: Longitude |
| 1605 | | rad | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | GNSS2 Latitude | | Data from GNSS2 module: Latitude |
| 1606 | GNSS2 Height Above Ellipsoid (WGS84) | m | Data from GNSS2 module: Height Above Ellipsoid (WGS84) |
| 1609 | GNSS2 PDOP (Dilution of Precision of Position) | customType | Data from GNSS2 module: PDOP - Relation between user position error and satellite position error |
| 1610 | GNSS2 Accuracy | m | Data from GNSS2 module: Accuracy |
| 1611 | GNSS2 Horizontal Accuracy Estimate | m | Data from GNSS2 module: Horizontal accuracy |
| 1612 | GNSS2 Vertical Accuracy Estimate | m | Data from GNSS2 module: Vertical accuracy |
| 1613 | GNSS2 Velocity North | m/s | Data from GNSS2 module: Velocity in North direction (NED Coordinates system) |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1614 | GNSS2 Velocity East | m/s | Data from GNSS2 module: Velocity in East direction (NED Coordinates system) |
| 1615 | GNSS2 Velocity Down | m/s | Data from GNSS2 module: Velocity in Down direction (NED Coordinates system) |
| 1616 | GNSS2 Speed Accuracy Estimate | m/s | Data from GNSS2 module: Speed accuracy |
| 1617 | GNSS2 Related Base Longitude | rad | Data from GNSS2 module: RTK Base longitude |
| 1618 | GNSS2 Related Base Latitude | rad | Data from GNSS2 module: RTK Base latitude |
| 1619 | GNSS2 Related Base WGS84 Altitude | m | Data from GNSS2 module: RTK Base WGS84 Altitude |
| 1620 | GNSS2 Related Base to Rover Azimuth | rad | Data from GNSS2 module: RTK Base-Rover vector azimuth |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | (Spherical Coordinates system) |
| 1621 | GNSS2 Related Base to Rover Elevation | rad | Data from GNSS2 module: RTK Base-Rover vector elevation (Spherical Coordinates system) |
| 1622 | GNSS2 Related Base to Rover Distance | m | Data from GNSS2 module: RTK Base-Rover vector distance (Spherical Coordinates system) |
| 1623 | GNSS2 Related Base to Rover Accuracy | m | Data from GNSS2 module: RTK Base-Rover vector accuracy |
| 1624 | GNSS2 Survey in Accuracy | m | Data from GNSS2 module: RTK Base accuracy when base knows it is fixed in a particular position |
| 1625 | | m | Data from GNSS2 module: RTK |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | GNSS2 Related Base to Rover North | | Base-Rover vector North (NED Coordinate system) |
| 1626 | GNSS2 Related Base to Rover East | m | Data from GNSS2 module: RTK Base-Rover vector East (NED Coordinate system) |
| 1627 | GNSS2 Related Base to Rover Down | m | Data from GNSS2 module: RTK Base-Rover vector Down (NED Coordinate system) |
| 1628 | GNSS2 Position Frequency | H | Data from GNSS2 module: Position frequency |
| 1629 | GNSS2 Jamming Indicator | % | Jamming indicator from U-Blox device 2 for GNSS |
| 1700-1731 | Actuator Output s0 - s31 | customType | Configurable variable from actuator outputs to be transformed by the system |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1771 | Commanded PDS | customType | PWM command signal for the Power Distribution System actuators |
| 1800 | Distance to Object of Interest 0 | m | Spherical coordinate to object of interest 0: distance |
| 1801 | Azimuth to Object of Interest 0 | rad | Spherical coordinate to object of interest 0: azimuth |
| 1802 | Elevation to Object of Interest 0 | rad | Spherical coordinate to object of interest 0: elevation |
| 1803 | Distance to Object of Interest 1 | m | Spherical coordinate to object of interest 1: distance |
| 1804 | Azimuth to Object of Interest 1 | rad | Spherical coordinate to object of interest 1: azimuth |
| 1805 | Elevation to Object of Interest 1 | rad | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 1: elevation |
| 1806 | Distance to Object of Interest 2 | m | Spherical coordinate to object of interest 2: distance |
| 1807 | Azimuth to Object of Interest 2 | rad | Spherical coordinate to object of interest 2: azimuth |
| 1808 | Elevation to Object of Interest 2 | rad | Spherical coordinate to object of interest 2: elevation |
| 1809 | Distance to Object of Interest 3 | m | Spherical coordinate to object of interest 3: distance |
| 1810 | Azimuth to Object of Interest 3 | rad | Spherical coordinate to object of interest 3: azimuth |
| 1811 | Elevation to Object of Interest 3 | rad | Spherical coordinate to object of interest 3: elevation |
| | | m | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1812 | Distance to Object of Interest 4 | | Spherical coordinate to object of interest 4: distance |
| 1813 | Azimuth to Object of Interest 4 | rad | Spherical coordinate to object of interest 4: azimuth |
| 1814 | Elevation to Object of Interest 4 | rad | Spherical coordinate to object of interest 4: elevation |
| 1815 | Distance to Object of Interest 5 | m | Spherical coordinate to object of interest 5: distance |
| 1816 | Azimuth to Object of Interest 5 | rad | Spherical coordinate to object of interest 5: azimuth |
| 1817 | Elevation to Object of Interest 5 | rad | Spherical coordinate to object of interest 5: elevation |
| 1818 | Distance to Object of Interest 6 | m | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 6: distance |
| 1819 | Azimuth to Object of Interest 6 | rad | Spherical coordinate to object of interest 6: azimuth |
| 1820 | Elevation to Object of Interest 6 | rad | Spherical coordinate to object of interest 6: elevation |
| 1821 | Distance to Object of Interest 7 | m | Spherical coordinate to object of interest 7: distance |
| 1822 | Azimuth to Object of Interest 7 | rad | Spherical coordinate to object of interest 7: azimuth |
| 1823 | Elevation to Object of Interest 7 | rad | Spherical coordinate to object of interest 7: elevation |
| 1824 | Distance to Object of Interest 8 | m | Spherical coordinate to object of interest 8: distance |
| | | rad | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1825 | Azimuth to Object of Interest 8 | | Spherical coordinate to object of interest 8: azimuth |
| 1826 | Elevation to Object of Interest 8 | rad | Spherical coordinate to object of interest 8: elevation |
| 1827 | Distance to Object of Interest 9 | m | Spherical coordinate to object of interest 9: distance |
| 1828 | Azimuth to Object of Interest 9 | rad | Spherical coordinate to object of interest 9: azimuth |
| 1829 | Elevation to Object of Interest 9 | rad | Spherical coordinate to object of interest 9: elevation |
| 1830 | Distance to Object of Interest 10 | m | Spherical coordinate to object of interest 10: distance |
| 1831 | Azimuth to Object of Interest 10 | rad | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 10: azimuth |
| 1832 | Elevation to Object of Interest 10 | rad | Spherical coordinate to object of interest 10: elevation |
| 1833 | Distance to Object of Interest 11 | m | Spherical coordinate to object of interest 11: distance |
| 1834 | Azimuth to Object of Interest 11 | rad | Spherical coordinate to object of interest 11: azimuth |
| 1835 | Elevation to Object of Interest 11 | rad | Spherical coordinate to object of interest 11: elevation |
| 1836 | Distance to Object of Interest 12 | m | Spherical coordinate to object of interest 12: distance |
| 1837 | Azimuth to Object of Interest 12 | rad | Spherical coordinate to object of interest 12: azimuth |
| | | rad | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1838 | Elevation to Object of Interest 12 | | Spherical coordinate to object of interest 12: elevation |
| 1839 | Distance to Object of Interest 13 | m | Spherical coordinate to object of interest 13: distance |
| 1840 | Azimuth to Object of Interest 13 | rad | Spherical coordinate to object of interest 13: azimuth |
| 1841 | Elevation to Object of Interest 13 | rad | Spherical coordinate to object of interest 13: elevation |
| 1842 | Distance to Object of Interest 14 | m | Spherical coordinate to object of interest 14: distance |
| 1843 | Azimuth to Object of Interest 14 | rad | Spherical coordinate to object of interest 14: azimuth |
| 1844 | Elevation to Object of Interest 14 | rad | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 14: elevation |
| 1845 | Distance to Object of Interest 15 | m | Spherical coordinate to object of interest 15: distance |
| 1846 | Azimuth to Object of Interest 15 | rad | Spherical coordinate to object of interest 15: azimuth |
| 1847 | Elevation to Object of Interest 15 | rad | Spherical coordinate to object of interest 15: elevation |
| 1848 | Distance to Object of Interest 16 | m | Spherical coordinate to object of interest 16: distance |
| 1849 | Azimuth to Object of Interest 16 | rad | Spherical coordinate to object of interest 16: azimuth |
| 1850 | Elevation to Object of Interest 16 | rad | Spherical coordinate to object of interest 16: elevation |
| | | m | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1851 | Distance to Object of Interest 17 | | Spherical coordinate to object of interest 17: distance |
| 1852 | Azimuth to Object of Interest 17 | rad | Spherical coordinate to object of interest 17: azimuth |
| 1853 | Elevation to Object of Interest 17 | rad | Spherical coordinate to object of interest 17: elevation |
| 1854 | Distance to Object of Interest 18 | m | Spherical coordinate to object of interest 18: distance |
| 1855 | Azimuth to Object of Interest 18 | rad | Spherical coordinate to object of interest 18: azimuth |
| 1856 | Elevation to Object of Interest 18 | rad | Spherical coordinate to object of interest 18: elevation |
| 1857 | Distance to Object of Interest 19 | m | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 19: distance |
| 1858 | Azimuth to Object of Interest 19 | rad | Spherical coordinate to object of interest 19: azimuth |
| 1859 | Elevation to Object of Interest 19 | rad | Spherical coordinate to object of interest 19: elevation |
| 1860 | Distance to Object of Interest 20 | m | Spherical coordinate to object of interest 20: distance |
| 1861 | Azimuth to Object of Interest 20 | rad | Spherical coordinate to object of interest 20: azimuth |
| 1862 | Elevation to Object of Interest 20 | rad | Spherical coordinate to object of interest 20: elevation |
| 1863 | Distance to Object of Interest 21 | m | Spherical coordinate to object of interest 21: distance |
| | | rad | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1864 | Azimuth to Object of Interest 21 | | Spherical coordinate to object of interest 21: azimuth |
| 1865 | Elevation to Object of Interest 21 | rad | Spherical coordinate to object of interest 21: elevation |
| 1866 | Distance to Object of Interest 22 | m | Spherical coordinate to object of interest 22: distance |
| 1867 | Azimuth to Object of Interest 22 | rad | Spherical coordinate to object of interest 22: azimuth |
| 1868 | Elevation to Object of Interest 22 | rad | Spherical coordinate to object of interest 22: elevation |
| 1869 | Distance to Object of Interest 23 | m | Spherical coordinate to object of interest 23: distance |
| 1870 | Azimuth to Object of Interest 23 | rad | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 23: azimuth |
| 1871 | Elevation to Object of Interest 23 | rad | Spherical coordinate to object of interest 23: elevation |
| 1872 | Distance to Object of Interest 24 | m | Spherical coordinate to object of interest 24: distance |
| 1873 | Azimuth to Object of Interest 24 | rad | Spherical coordinate to object of interest 24: azimuth |
| 1874 | Elevation to Object of Interest 24 | rad | Spherical coordinate to object of interest 24: elevation |
| 1875 | Distance to Object of Interest 25 | m | Spherical coordinate to object of interest 25: distance |
| 1876 | Azimuth to Object of Interest 25 | rad | Spherical coordinate to object of interest 25: azimuth |
| | | rad | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1877 | Elevation to Object of Interest 25 | | Spherical coordinate to object of interest 25: elevation |
| 1878 | Distance to Object of Interest 26 | m | Spherical coordinate to object of interest 26: distance |
| 1879 | Azimuth to Object of Interest 26 | rad | Spherical coordinate to object of interest 26: azimuth |
| 1880 | Elevation to Object of Interest 26 | rad | Spherical coordinate to object of interest 26: elevation |
| 1881 | Distance to Object of Interest 27 | m | Spherical coordinate to object of interest 27: distance |
| 1882 | Azimuth to Object of Interest 27 | rad | Spherical coordinate to object of interest 27: azimuth |
| 1883 | Elevation to Object of Interest 27 | rad | Spherical coordinate to |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | object of interest 27: elevation |
| 1884 | Distance to Object of Interest 28 | m | Spherical coordinate to object of interest 28: distance |
| 1885 | Azimuth to Object of Interest 28 | rad | Spherical coordinate to object of interest 28: azimuth |
| 1886 | Elevation to Object of Interest 28 | rad | Spherical coordinate to object of interest 28: elevation |
| 1887 | Distance to Object of Interest 29 | m | Spherical coordinate to object of interest 29: distance |
| 1888 | Azimuth to Object of Interest 29 | rad | Spherical coordinate to object of interest 29: azimuth |
| 1889 | Elevation to Object of Interest 29 | rad | Spherical coordinate to object of interest 29: elevation |
| | | m | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 1890 | Distance to Object of Interest 30 | | Spherical coordinate to object of interest 30: distance |
| 1891 | Azimuth to Object of Interest 30 | rad | Spherical coordinate to object of interest 30: azimuth |
| 1892 | Elevation to Object of Interest 30 | rad | Spherical coordinate to object of interest 30: elevation |
| 1893 | Distance to Object of Interest 31 | m | Spherical coordinate to object of interest 31: distance |
| 1894 | Azimuth to Object of Interest 31 | rad | Spherical coordinate to object of interest 31: azimuth |
| 1895 | Elevation to Object of Interest 31 | rad | Spherical coordinate to object of interest 31: elevation |
| 2000 | RX Packet Error Rate (on board) | decimal | Value rating RX packets and expected RX |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | packets, given as % error |
| 2001 | TX Packet Error Rate (on board) | decimal | Value rating TX packets and expected TX packets, given as % error |
| 2002 | Computed RX pkt/s Used for RX PER | messages | Packages per second received to the UAV configured in communication statistics |
| 2003 | Remote RX pkt/s Used for TX PER | messages | RX packages per second received and computed through communications |
| 2004 | Computed TX pkt/s Used for TX PER | messages | Packages per second transmitted to the UAV configured in communication statistics |
| 2005 | Remote TX pkt/s Used for RX PER | messages | TX packages per second received and computed |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | through communications |
| 2019 | Stick RX Rate | Hz | Number of stick messages received per second |
| 2020 | Position Fix Time | s | Time spend with GNSS without losing fix |
| 2040-2042 | Tunnel Producer Receive Frequency 0-2 | Hz | Tunnel producer 0-2 receives data at this frequency |
| 2043-2045 | Tunnel Consumer Send Frequency 0-2 | Hz | Tunnel consumer 0-2 receives data at this frequency |
| 2046 | Max Duration of Step in CIO | s | Longest time duration from a step in CIO |
| 2047 | Acquisition Task Timestep | s | Average period to execute the acquisition thread |
| 2048 | Acquisition Task Maximum Timestep | s | Maximum period to execute the acquisition thread |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2049 | Cross Core Message Queue CPU Ratio | percentage | % of time of CPU that CIO waits for inter-core communications queue to be emptied |
| 2050 | Acquisition Task Average CPU Ratio | percentage | Average % of CPU time spent in the acquisition thread |
| 2051 | Acquisition Task Maximum CPU Ratio | percentage | Maximum % of CPU time spent in the acquisition thread |
| 2052 | Acquisition Task Average Time | s | Average time for acquisition thread |
| 2053 | Acquisition Task Maximum Time | s | Maximum time for acquisition thread |
| 2054 | CIO Max Time | s | Maximum time of CIO thread |
| 2055 | CIO Average Time | s | Average and Maximum time of CIO thread |
| 2056 | | % | Percentage of communication employed |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Cross-Core Message Queue Usage | | between both microprocessors |
| 2057 | CIO Running Frequency | Hz | C1 low running frequency |
| 2058 | MC CIO Min Running Frequency | Hz | Minimum assured frequency of low priority task<br><br>**Note**<br>Only for MC |
| 2092 | CPU2 Low priority current frequency | Hz | Current running frequency of the CPU2 low priority tasks |
| 2093 | CPU2 Low priority min frequency | Hz | Minimum recorded frequency of the CPU2 low priority tasks |
| 2094 | GNC Task Average CPU Ratio | percentage | Average % of CPU time of GNC task |
| 2095 | GNC Task Maximum CPU Ratio | percentage | Maximum % of CPU time of GNC task |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2096 | GNC Task Average Time | s | Average time spent on GNC task |
| 2097 | GNC Task Maximum Time | s | Maximum time spent on GNC task |
| 2098 | GNC Task Maximum Timestep | s | Maximum execution period for GNC task |
| 2099 | Max Duration of Step in GNC | s | Maximum duration of one step in GNC |
| 2100 | Gyroscope Based on Accelerometer - X Body Axis | rad/s | Gyroscope measurements obtained from accelerometer X-axis data |
| 2101 | Gyroscope Based on Accelerometer - Y Body Axis | rad/s | Gyroscope measurements obtained from accelerometer Y-axis data |
| 2102 | Gyroscope Based on Accelerometer - Z Body Axis | rad/s | Gyroscope measurements obtained from accelerometer Z-axis data |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2103 | Acceleration North | m/s² | Acceleration in the North direction (NED Coordinates System) |
| 2104 | Acceleration East | m/s² | Acceleration in the East direction (NED Coordinates System) |
| 2105 | Acceleration Down | m/s² | Acceleration in the Down direction (NED Coordinates System) |
| 2112 | Estimated Dem | m | Altitude given by the estimated Digital Elevation Model |
| 2200 | Curve Length Covered | m | Total distance from current mission length covered |
| 2201 | Curve Length | m | Total distance from current mission length |
| 2202 | Curve Length Pending | m | Total distance from current |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | mission length not covered yet |
| 2203 | Curve Parameter Covered | customType | Total length covered from current mission according to parameter selected |
| 2204 | Curve Parameter Range | customType | Total length from current mission according to parameter selected |
| 2205 | Curve Parameter Pending | customType | Total length from current mission to be covered according to parameter selected yet |
| 2206 | Curve Horizontal Curvature | customType | Horizontal curvature of the current curve |
| 2300-2302 | Joint 0-2 of Gimbal 0 | rad | Variables for Gimbal 0 configuration - Angles sent to gimbal as Yaw (0), |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | Pitch (1) and Roll (2) |
| 2303-2305 | Joint 0-2 of Gimbal 1 | rad | Variables for Gimbal 1 configuration - Angles sent to gimbal as Yaw (0), Pitch (1) and Roll (2) |
| 2330 | Control Loop Period | s | MC control loop period |
| 2331 | Control Loop Maximum Period | s | MC maximum control loop period |
| 2332 | Control Loop Duration | s | MC control loop average execution time |
| 2333 | MC Control Loop Maximum Duration | s | MC control loop maximum average execution time |
| 2334 | Control Loop CPU Usage Ratio | % | CPU usage ratio |
| 2335 | MC Control Loop | % | MC maximum CPU usage ratio |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Maximum CPU Usage Ratio | | |
| 2336-2338 | MC U-V-W Phase Current | A | MC U-V-W phase current |
| 2339 | MC Electrical Angle | rad | MC electrical angle |
| 2340 | MC01 Mechanical Angle | rad | MC01 mechanical angle |
| 2341 | MC Mechanical Angular Speed | rad/s | MC mechanical angular speed |
| 2342 | MC01 Desired Mechanical Angle | rad | MC01 desired mechanical angle |
| 2343 | MC01 Position Controller Output | rad/s | MC01 position PDI output |
| 2344 | MC Desired Mechanical Angular Speed | rad/s | MC desired mechanical angular speed |
| 2345 | MC Desired Mechanical Angular Speed After Speed Limiter | rad/s | MC desired mechanical angular speed after speed limiter |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2346 | MC Speed Controller Output | A | MC speed controller output |
| 2347-2348 | MC Alpha-Beta Current | A | MC alpha and beta current after Clarke transformation |
| 2349-2350 | MC Actual Direct Current | A | MC actual direct current |
| 2351 | MC Desired Direct Current | A | MC desired direct current |
| 2352 | MC Desired Quadrature Current | A | MC desired quadrature current |
| 2353 | MC Direct Voltage From Controller Output | V | MC direct voltage from controller output |
| 2354 | MC Quadrature Voltage From Controller Output | V | MC quadrature voltage from controller output |
| 2355-2356 | MC Alpha - Beta Voltage From Current | V | MC Alpha - Beta voltage from |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Controller Output | | current controller output |
| 2357-2358 | MC01 Desired Clarke Alpha-Beta current | customType | MC01 desired Clarke currents |
| 2359-2361 | MC01 U-V-W Phase Space Vector Generator Output | customType | MC01 phase time constants |
| 2362-2364 | MC U-V-W Phase PWM Duty Cycle | percentage | MC U-V-W Phase PWM duty cycle |
| 2365 | MC01 Encoder Raw Angle | rad | MC01 encoder raw measured angle |
| 2366 | MC01 Stepper Output Frequency | Hz | MC01 stepper output frequency |
| 2367 | MC Mechanical Angle Error | rad | MC mechanical angle error |
| 2368-2370 | MC U-V-W Phase BEMF | V | MC U-V-W phase electromechanical force |
| | | A | |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2371 | MC Input Current DC side | | MC input current DC side |
| 2372 | MC Input Normalized Command Speed | customType | MC input normalized command speed |
| 2373-2374 | MC ADC in 0-1 | V | System reserved variables |
| 2375 | MC Logic Board Temperature | K | MC logic board temperature |
| 2376 | MC Power Module Temperature | K | MC IGBT filtered temperature |
| 2377 | MC Motor Temperature | K | MC Motor temperature |
| 2378 | MC Input Voltage DC side | V | MC DC bus voltage |
| 2379-2380 | MC U-V Phase Hall current sensor | customType | System reserved variables |
| 2381 | MC Virtual and estimator | rad | MC Angle offset value from estimated and |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | angle difference | | commanded angle to close control loop |
| 2382 | MC Low speed estimator angle | rad | MC Low speed observer estimated angle |
| 2383 | MC High speed estimator angle | rad | MC High speed observer estimated angle |
| 2384 | MC Low speed estimator speed | rad/s | MC Low speed observer estimated mechanical speed |
| 2385 | MC High speed estimator speed | rad/s | MC High speed observer estimated mechanical speed |
| 2386 | String DC current | A | Direct current measured in the motor string/ power line |
| 2400-2419 | Control Output u0-19 | customType | Control output 0-19 after servo saturation |
| 2450-2469 | | customType | Inner controller output 0-19 |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Inner Controller Output 0-19 | | |
| 2500-2519 | Stick Input u0-u19 | customType | Intermediate values from stick used for arcade mode |
| 2600-2619 | Stick Input d0-d19 | customType | Intermediate values from stick used for arcade mode - delta values |
| 2700-2739 | Operation Variable 00-39 | customType | Configurable values used in different guidances - Position or values or vectors |
| 2800 | Wind Velocity North | m/s | Wind velocity vector pointing North direction (NED Coordinate system) |
| 2801 | Wind Velocity East | m/s | Wind velocity vector pointing East direction (NED Coordinate system) |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2802 | Wind Velocity Down | m/s | Wind velocity vector pointing Down direction (NED Coordinate system) |
| 2803 | Wind Velocity Estimation Uncertainty (Element 0-0) (Deprecated) | m/s | 0-0 element from covariance matrix in wind estimation<br>**Warning**<br>**Deprecated variable** |
| 2804 | Wind Velocity Estimation Uncertainty (Element 1-0) (Deprecated) | m/s | 1-0 element from covariance matrix in wind estimation<br>**Warning**<br>**Deprecated variable** |
| 2805 | Wind Velocity Estimation Uncertainty (Element 2-0) (Deprecated) | m/s | 2-0 element from covariance matrix in wind estimation |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | **Warning** **Deprecated variable** |
| 2806 | Wind Velocity Estimation Uncertainty (Element 0-1) (Deprecated) | m/s | 0-1 element from covariance matrix in wind estimation<br><br>**Warning** **Deprecated variable** |
| 2807 | Wind Velocity Estimation Uncertainty (Element 1-1) (Deprecated) | m/s | 1-1 element from covariance matrix in wind estimation<br><br>**Warning** **Deprecated variable** |
| 2808 | Wind Velocity Estimation Uncertainty (Element 2-1) (Deprecated) | m/s | 2-1 element from covariance matrix in wind estimation<br><br>**Warning** **Deprecated variable** |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| 2809 | Wind Velocity Estimation Uncertainty (Element 0-2) (Deprecated) | m/s | 0-2 element from covariance matrix in wind estimation<br>**Warning**<br>**Deprecated variable** |
| 2810 | Wind Velocity Estimation Uncertainty (Element 1-2) (Deprecated) | m/s | 1-2 element from covariance matrix in wind estimation<br>**Warning**<br>**Deprecated variable** |
| 2811 | Wind Velocity Estimation Uncertainty (Element 2-2) (Deprecated) | m/s | 2-2 element from covariance matrix in wind estimation<br>**Warning**<br>**Deprecated variable** |
| 2812 | Wind Azimuth Angle (Deprecated) | degree | Wind estimated azimuth |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | | | **Warning** **Deprecated variable** |
| 2813 | Wind Velocity in North-East plane (Deprecated) | m/s | Wind velocity norm xy in North-East plane<br><br>**Warning** **Deprecated variable** |
| 2814 | Head wind velocity North | m/s | Head wind velocity North |
| 2815 | Head wind velocity East | m/s | Head wind velocity East |
| 2816 | Head wind velocity Down | m/s | Head wind velocity Down |
| 2830-2893 | Setup real 00-63 | customType | Setup variable 00-63 |
| 2900 | MSL Right from Actual QNH and Pressure Measurement | m | Mean Sea Level obtained from Actual QNH and current Pressure Measurement |
| 2901 | | m | Mean Sea Level calculated for ISO International |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | MSL for ISA and Pressure Measurement | | Standard Atmosphere and Pressure Measurement |
| 2902 | Time Since Entering Current Phase | s | Time-lapse considered since entering the current phase |
| 2903 | GNC Timestep | s | Task execution period from GNC |
| 2904 | Total Flight Time | s | Time-lapse since the vehicle finished Standby<br>**Warning**<br>**Deprecated variable** |
| 2905 | Total Flight Distance | m | Distance covered by the vehicle in all mission length<br>**Warning**<br>**Deprecated variable** |
| 2906 | Reception Frequency of Simulated | Hz | Frequency at which the system receives |

| ID | Name | Units/ Values | Description |
|---|---|---|---|
| | Navigation Data | | Simulation Navigation Data |
| 2907 | Reception Frequency of External Navigation Data | Hz | Frequency at which the system receives External Navigation Data |
| 2908-2928 | Time in Phase 0-20 | s | Time-lapse spent by the vehicle within phase 0-20 |
| 📝 3000-3031 | Simulation Variable 0-31 | customType | Variables used for Simulation data |
| 📝 3100-4099 | User Variable 00-999 (Real - 32 Bits) | customType | Free variables for the user to use |
| 4100 | Zero | customType | Constant value 0 |
| 4101 | Rvar Disabled | customType | Disabled variable |

## Integer Variables (UVar) - 16 Bits

| ID | Name | Description |
|---|---|---|
| 0 | Actuator Mode | Index pointing to the flight mode in use |
| 1 | Phase Identifier | Index pointing to the active phase |

| ID | Name | Description |
|---|---|---|
| 2 | Internal ADC 0 | Internal ADC pin<br><br>**Warning**<br>**Variable for internal use** |
| 3-7 | ADC 0-4 | Direct reading of ADC pins |
| 8-18 | Internal ADC 1-11 | Internal ADC pin<br><br>**Warning**<br>**Variable for internal use** |
| 19 | Current envelope (deprecated) | Index pointing to the used envelope<br><br>**Warning**<br>**Deprecated variable** |
| 20 | Counter for C2 system | C2 counter to monitor if Core 2 is alive |
| 21 | Total memory for blocks allocation (low part) | Total words available for blocks (low part).<br>It is stored in two parts because the size can be a |

| ID | Name | Description |
|---|---|---|
| | | number up to 32 bits, but the uvars are stored in variables of 16, so to get the true size the user would have to put together both the low and the high part.<br><br>**Note**<br>1 word = 2 bytes |
| 22 | Total memory for blocks allocation (high part) | Total words available for blocks (high part).<br>It is stored in two parts because the size can be a number up to 32 bits, but the uvars are stored in variables of 16, so to get the true size the user would have to put together both the low and the high part. |

| ID | Name | Description |
|---|---|---|
| | | **Note**<br>1 word = 2 bytes |
| 23 | Memory used for blocks allocation (low part) | Words used for blocks in allocator (low part).<br>It is stored in two parts because the size can be a number up to 32 bits, but the uvars are stored in variables of 16, so to get the true size the user would have to put together both the low and the high part.<br><br>**Note**<br>1 word = 2 bytes |
| 24 | Memory used for blocks allocation (high part) | Words used for blocks in allocator (high part).<br>It is stored in two parts because the size can be a number up to 32 |

| ID | Name | Description |
|---|---|---|
| | | bits, but the uvars are stored in variables of 16, so to get the true size the user would have to put together both the low and the high part.<br><br>**Note**<br>1 word = 2 bytes |
| 25 | SRTM source at UAV's position | Index for the SRTM source type:<br><br>• 0: Processing<br>• 1: Invalid<br>• 2: 90-meter resolution in equator<br>• 3: 30-meter resolution in equator |
| 26 | Stanag Flight Termination Mode | Specific mode for flight termination according to STANAG standards |
| 50 | | Index for PDI error source |

| ID | Name | Description |
|---|---|---|
| | PDI error source | identification. For further information, consult the **List of PDI errors** |
| 51 | Operation error source | Index for operation error source identification |
| 53 | 4X Veronte ID | ID of this AP in a redundant (0-3) to compare with the selected CAP. For more information, check **Integer Variables - 4x Software Manual** |
| 54 | 4X Veronte CAP | Current Autopilot 1x selected. For more information, check **Integer Variables - 4x Software Manual** |
| 55-75 | 4X Integer variables | For more information, check **Integer Variables - 4x Software Manual** |

| ID | Name | Description |
|---|---|---|
| 80 | Detour calculation identifier | Counter of number of times a route change has been calculated |
| 81 | Approach calculation identifier | Counter of number of times Approach guidance has been calculated |
| 82 | Climb calculation identifier | Counter of number of times Climb guidance has been calculated |
| 83 | Cruise calculation identifier | Counter of number of times Cruise guidance has been calculated |
| 84 | Rendezvous calculation identifier | Counter of number of times Rendezvous guidance has been calculated |
| 85 | Taxi calculation identifier | Counter of number of times Taxi guidance has been calculated |
| 86 | VTOL calculation identifier | Counter of number of times VTOL |

| ID | Name | Description |
|---|---|---|
| | | guidance has been calculated |
| 90 | Version Major | Major software version |
| 91 | Version Minor | Minor software version |
| 92 | Version Revision | Revision software version |
| 93 | UAV address (hight) | Higher part of the UAV system address |
| 94 | UAV address (low) | Lower part of the UAV system address |
| 95 | UAV address | UAV address |
| 96 | File system status | State error for DFS2 FS<br>For further information, consult the **List of File System Errors** |
| 97 | Number of registered partitions on DFS2 File System | Number of registered partitions on DFS2 File System |

| ID | Name | Description |
|---|---|---|
| 100 | GNSS1 Number of Satellites Used in Solution | Number of satellites used in solution |
| 101-102 | GNSS1 rejected-accepted RTCM 1005 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1005 |
| 103-104 | GNSS1 rejected-accepted RTCM 1077 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1077 |
| 105-106 | GNSS1 rejected-accepted RTCM 1087 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1087 |
| 107-108 | GNSS1 rejected-accepted RTCM 1127 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1127 |
| 109-110 | GNSS1 rejected-accepted RTCM 1230 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1230 |

| ID | Name | Description |
|---|---|---|
| 111-112 | GNSS1 rejected-accepted RTCM 4072 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 4072 |
| 113 | GNSS1 rejected RTCM unknown type | Number of RTCM unknown rejected by wrong CRC |
| 114 | GNSS1 week | GNSS1 week |
| 115 | GNSS1 Jamming Status | Output from GPS 1 jamming/ interference monitor:<br><br>• 0 = unknown or feature disabled<br>• 1 = ok ⇒ no significant jamming<br>• 2 = warning ⇒ interference visible but fix Ok<br>• 3 = critical ⇒ interference visible and no fix |
| 116 | | |

| ID | Name | Description |
|---|---|---|
| | GNSS1 Spoofing detection state | GNSS1 Spoofing detection state |
| 117 | GNSS1 Fix Type | Type of fix reported by the GNSS 1 receiver (Ublox):<br><br>• 0: No fix<br>• 1: Dead reckoning only<br>• 2: 2D fix<br>• 3: 3D fix<br>• 4: GNSS + dead reckoning combined<br>• 5: Time only fix |
| 150 | GNSS2 Number of Satellites Used in Solution | Number of Satellites Used in Solution |
| 151-152 | GNSS2 rejected-accepted RTCM 1005 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1005 |
| 153-154 | GNSS2 rejected- | Number of RTCM rejected by wrong CRC - correctly |

| ID | Name | Description |
|---|---|---|
| | accepted RTCM 1077 | received by Ublox 1077 |
| 155-156 | GNSS2 rejected-accepted RTCM 1087 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1087 |
| 157-158 | GNSS2 rejected-accepted RTCM 1127 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1127 |
| 159-160 | GNSS2 rejected-accepted RTCM 1230 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 1230 |
| 161-162 | GNSS2 rejected-accepted RTCM 4072 | Number of RTCM rejected by wrong CRC - correctly received by Ublox 4072 |
| 163 | GNSS2 rejected RTCM unknown type | Number of RTCM unknown rejected by wrong CRC |
| 164 | GNSS2 week | GNSS2 week |
| 165 | | Output from GPS 2 jamming/ |

| ID | Name | Description |
|---|---|---|
| | GNSS2 Jamming Status | interference monitor:<br><br>• 0 = unknown or feature disabled<br>• 1 = ok ⇒ no significant jamming<br>• 2 = warning ⇒ interference visible but fix Ok<br>• 3 = critical ⇒ interference visible and no fix |
| 166 | GNSS2 Spoofing detection state | GNSS2 Spoofing detection state |
| 167 | GNSS2 Fix Type | Type of fix reported by the GNSS 2 receiver (Ublox): |

| ID | Name | Description |
|---|---|---|
| | | • 0: No fix<br>• 1: Dead reckoning only<br>• 2: 2D fix<br>• 3: 3D fix<br>• 4: GNSS + dead reckoning combined<br>• 5: Time only fix |
| 200 | Radar Altimeter State | Index for the radar altimeter state |
| 201 | Current Section | ID of current patch being executed by the autopilot |
| 202 | Last Achieved Section | ID of last patch completed by the autopilot |
| 203 | Track Stage | Index of type of tracked route:<br><br>• 0: No route<br>• 1: Route from mission<br>• 2: Commanded route |
| 204 | Current patchset ID | Index showing the patchset: |

| ID | Name | Description |
|---|---|---|
| | | • 0: Approach<br>• 1: Climb<br>• 2: Route<br>• 3: Taxi<br>• 4: VTOL<br>• 5: Rendezvous<br>• 6: Detour |
| 205 | Amount of laps done | Number of laps completed on the route |
| 310-311 | Iridium sent-received | Number of packets successfully sent/received |
| 312 | Number of UDP packets sent | Total count of UDP telemetry packets transmitted by the system |
| 313 | Number of UDP packets received | Total count of UDP packets successfully received |
| 314 | Number of UDP packets received but discarded | Count of UDP packets dropped due to errors or buffer issues |
| 315 | Number of packets lost | Packets lost in internal |

| ID | Name | Description |
|---|---|---|
| | when sending from C1 to CM | communication between Core 1 and Communications Module |
| 316 | Number of packets lost on send from C1 to CM | Internal port transmission loss count from Core 1 to Communications Module |
| 396 | Error code for IMU 3 | Diagnostic error code generated by the third Inertial Measurement Unit |
| 397 | External navigation sensor status | External navigation sensor status |
| 398 | VectorNav Mode | Index showing external source VectorNav mode |
| 399 | Identifier of max duration step in acquisition | Identifier of maximum duration step in CIO<br>**Warning**<br>**Variable for internal use** |
| 400 | | Internest raw status |

| ID | Name | Description |
|---|---|---|
| | Internest raw status | |
| 401 | Navigation source | Index pointing to the primary navigation source:<br><br>• 1: Internal navigation (Extended Kalman Filter)<br>• 2: Simulated navigation (IRX source)<br>• 3: External navigation using VCP (IRX source)<br>• 4: External navigation using dedicated variables<br>• 5: External navigation from Vectornav VN-300<br>• 6: External sensor navigation |
| 402 | Raw position source identifier | GPS identifier selected as main |

| ID | Name | Description |
|---|---|---|
| 403 | Selected static pressure sensor (Deprecated) | Static pressure sensor selection<br>**Warning**<br>**Deprecated variable** |
| 404 | Selected dynamic pressure sensor (Deprecated) | Dynamic pressure sensor selection |
| 405 | Selected primary accelerometer (deprecated) | Primary accelerometer selection<br>**Warning**<br>**Deprecated variable** |
| 406 | Selected primary gyroscope (deprecated) | Primary gyroscope selection<br>**Warning**<br>**Deprecated variable** |
| 409 | Selected magnetometer (Deprecated) | Magnetometer selection |

| ID | Name | Description |
|---|---|---|
| | | **Warning** **Deprecated variable** |
| 410 | Selected stick priority table | Stick priority table selection |
| 425 | Identifier of max duration step in GNC | Step with maximum duration |
| 426 | Group of user bits selected for CBIT | Index pointing to the selected **list of safety bits**. This is the group of user bits selected to be computed with system CBIT. |
| 441 | Serial to CAN 0 frames dropped | Frames lost during serial data encapsulation into CAN 0 bus |
| 442 | Serial to CAN 1 frames dropped | Frames lost during serial data encapsulation into CAN 1 bus |
| 443 | Serial to CAN 2 frames dropped | Frames lost during serial data encapsulation into CAN 2 bus |

| ID | Name | Description |
|---|---|---|
| 444 | Serial to CAN 3 frames dropped | Frames lost during serial data encapsulation into CAN 3 bus |
| 445 | Serial to CAN 4 frames dropped | Frames lost during serial data encapsulation into CAN 4 bus |
| 446 | Serial to CAN 5 frames dropped | Frames lost during serial data encapsulation into CAN 5 bus |
| 447 | CAN-FD Tx errors | Transmission error count for the CAN Flexible Data-rate bus |
| 448 | CAN-FD Rx errors | Reception error count for the CAN Flexible Data-rate bus |
| 449 | Configured system errors that had triggered | Bitarray containing the errors that can trigger the System Error. |
| 450 | CAN-A Tx errors | CAN A communication errors in transmission |

| ID | Name | Description |
|---|---|---|
| 451 | CAN-A Rx errors | CAN A communication errors in reception |
| 452 | CAN-B Tx errors | CAN B communication errors in transmission |
| 453 | CAN-B Rx errors | CAN B communication errors in reception |
| 454-459 | CAN to Serial 0-5 frames dropped | Lost messages during CAN to Serial transformations |
| 460-461 | First-Last file Periodic log | First-Last file of the periodic log |
| 462-463 | First-Last file Event log | First-Last file of the event log |
| 464-465 | First-Last file Fast log | First-Last file of the fast log |
| 466 | First file RAM log | Identifier of the first log file stored in RAM memory |
| 467 | Last file RAM log | Identifier of the latest log file stored in RAM memory |

| ID | Name | Description |
|---|---|---|
| 480-485 | COM0-5 packet discarded | Discarded packets at COM 0-5 |
| 490 | Number of moving obstacles detected | Number of moving obstacles detected |
| 491-492 | Veronte static cfg CRC(no Operation) of files (Higher-Lower 16 bits) | Veronte static cfg CRC (no Op.) of files |
| 493-494 | Veronte static cfg CRC(no Operation) of memory (Higher-Lower 16 bits) | Veronte static cfg CRC (no Op.) of memory |
| 495-496 | Global configuration state (crc) of files-memory (Higher-Lower 16 bits) | Global configuration state (crc) of files and memory |
| 497 | Config manager status (flash / sd / | Configuration manager status |

| ID | Name | Description |
|---|---|---|
| | maintenance mode) | |
| 498-499 | Global configuration state (crc) of files-memory | Global configuration state (crc) of files and memory |
| 550-557 | Reserved 0-7 | System reserved variables for Gimbal |
| 600-615 | PPM channel 0-15 output | CEX PPM channel outputs |
| 620 | Jetibox max successfully parsed message | **Note**<br>**CEX variable** |
| 700 | Planet Satcom connection Status | Planet Satcom connection Status |
| 701 | Planet Satcom signal strength | Planet Satcom signal strength |
| 710 | ADS-B OUT - Squawk Code | ADS-B Squawk code, 4 digits that allow the operator to inform about its status.<br>This variable is closely related to |

| ID | Name | Description |
|---|---|---|
| | | the management of communications between transponders and Veronte Autopilot 1x.<br><br>**Warning**<br>• **Deprecated variable**<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 711 | ADS-B OUT - ICAO | ADS-B ICAO, 4 ASCII characters assigned by aircraft authority as an identifier. This variable is closely related to the management of communications between transponders and Veronte Autopilot |

| ID | Name | Description |
|---|---|---|
| | | 1x.<br><br>**Warning**<br>• **Deprecated variable**<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 712 | ADS-B OUT - Ident | Index indicating whether the identification is enabled or disabled.<br>This is the identification of the UAV at the request of ATC.<br>This variable is closely related to the management of communications between transponders and Veronte Autopilot |

| ID | Name | Description |
|---|---|---|
| | | 1x.<br><br>**Warning**<br>• **Deprecated variable**<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 713 | ADS-B OUT - Mode | Index of ADS-B mode: IN, OUT or BOTH.<br>This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Deprecated variable**<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 714-721 | ADS-B OUT - Call sign 0-7 | ADS-B Call sign, 9 ASCII characters used by the operator to be identified during communication. These variables are closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Deprecated variables**<br>• If the user modifies these variables, it is not guaranteed that the transponder will continue to function correctly |
| 730 | Ping1090 - Sequence number (Deprecated) | **Warning**<br>**Deprecated variable** |
| 741 | Sagetech MXS - Hemisphere data status (Deprecated) | Sagetech variable, used by block to parse variables for GPS Navigation Data Message. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 742 | Sagetech MXS - Ground track (Deprecated) | Sagetech variable, used by block to parse variables for GPS Navigation Data Message. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 743 | Sagetech MXS - Air speed (Deprecated) | Sagetech variable, used by block to parse variables for GPS Navigation Data Message. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 750 | ADS-B Out / ICAO High | ADS-B ICAO, 4 ASCII characters assigned by aircraft authority as an identifier. This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 751 | ADS-B Out / ICAO Low | ADS-B ICAO, 4 ASCII characters assigned by aircraft authority as an identifier. This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 752 | ADS-B Out / Emitter Type | Type/category of ADS-B emitter. This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 753-760 | ADS-B Out / Call Sign 0-7 | ADS-B Call sign, 9 ASCII characters used by operator to be identified during communication. This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variabled for internal use** (custom message for transponder)<br>• If the user modifies these variables, it is not guaranteed that the transponder will continue to function correctly |
| 761 | ADS-B Out / Type | Model of ADS-B transponder. This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 762 | ADS-B Out / Control | Index of ADS-B control: OFF, ADS-B IN, ADS-B OUT or BOTH (ADS-B IN and OUT).<br>This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 763 | ADS-B Out / Squawk | ADS-B Squawk code, 4 digits that allow the operator to inform about its status.<br>This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 764 | ADS-B Out / Ident | Index indicating whether the identification is enabled or disabled.<br>This is the identification of the UAV at the request of ATC.<br>This variable is closely related to the management of communications between transponders and |

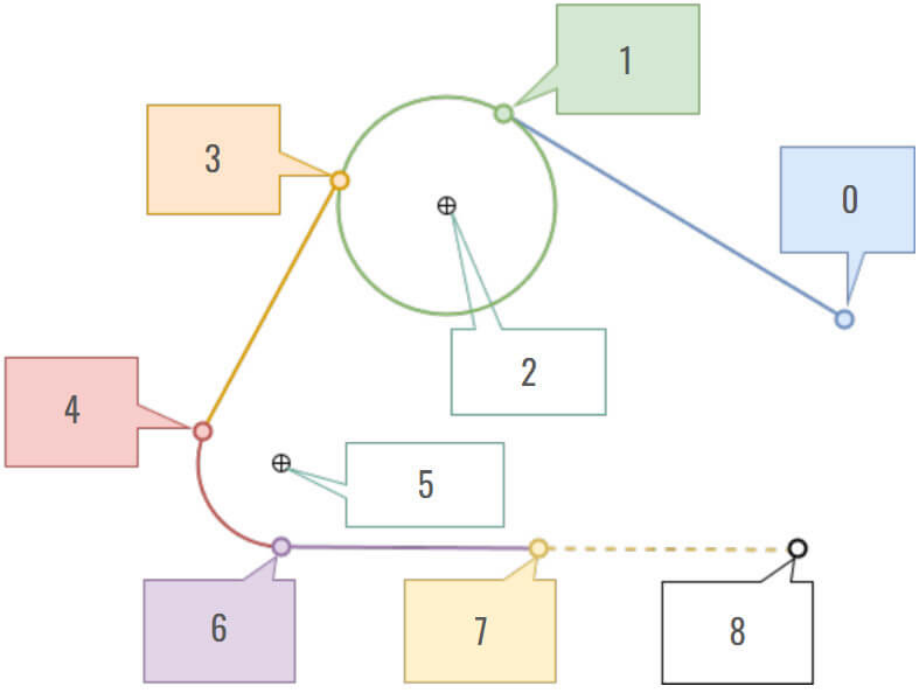
| ID | Name | Description |
|---|---|---|
| | | Veronte Autopilot 1x.<br><br>**Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 765 | ADS-B Out / Custom | Variable for internal use for ADS-B Out. This variable is closely related to the management of communications between transponders and Veronte Autopilot 1x. |

| ID | Name | Description |
|---|---|---|
| | | **Warning**<br>• **Variable for internal use** (custom message for transponder)<br>• If the user modifies this variable, it is not guaranteed that the transponder will continue to function correctly |
| 800 | MC Fault Id | Index of the MC error<br><br>**Warning**<br>**Deprecated variable** |
| 801 | MC Input Control Mode | Index of the MC control input mode: |

| ID | Name | Description |
|---|---|---|
| | | • 1: PPM<br>• 2: CAN<br>• 3: both modes active (CAN priority) |
| 802 | MC Actual Control Machine State | State of motor controller:<br><br>• 0: Motor stop and driver disabled<br>• 1: Calibration of ADC reading<br>• 2: Initial alignment procedure<br>• 3: Open loop procedure<br>• 4: Speed mode |
| 900-909 | Simulation variable 00-09 | Variables used for simulation data |
| 1000-1999 | User Variable 00-999 (Unsigned Integer - 16 bits) | Free variables for user |
| 2000 | Uvar Disabled | Disabled variable |
| 2001 | Zero | |

| ID | Name | Description |
|---|---|---|
| | | Variable with constant 0 value |

## Features Variables - 64 Bits



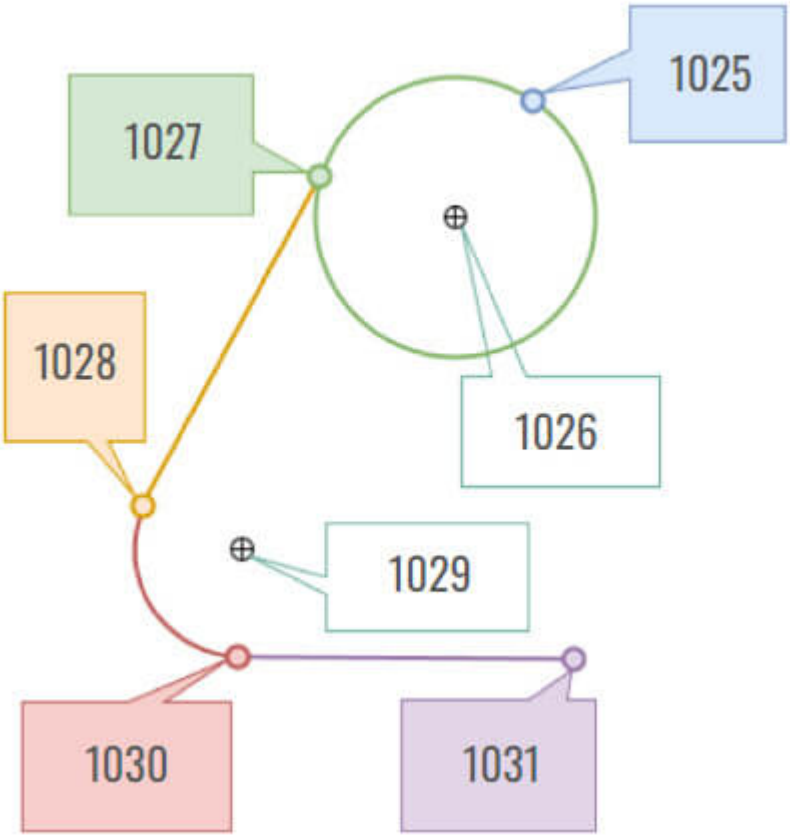

**Landing route - Features variables**

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 0 | Approach Initial Point | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **0** in **Landing capture**. |
| 1 | Approach Loiter Start | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **1** in |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| | | | | **Landing capture**. |
| 2 | Approach Loiter Center | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **2** in **Landing capture**. |
| 3 | Approach Loiter Finish | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **3** in **Landing capture**. |
| 4 | Approach Headturn Start | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **4** in **Landing capture**. |
| 5 | Approach Headturn Center | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **5** in **Landing capture**. |
| 6 | Approach Headturn Finish | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **6** in |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| | | | | **Landing capture**. |
| 7 | Approach Touch Point | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **7** in **Landing capture**. |
| 8 | Approach Runway End | [lon, lat, height] | [rad, rad, m] | Landing Guidance Variable. Point **8** in **Landing capture**. |

> ⓘ **Note**
>
> For further information regarding Landing guidance, please refer to Landing - Guidance blocks of **Block Programs** section of the **1x PDI Builder** user manual.

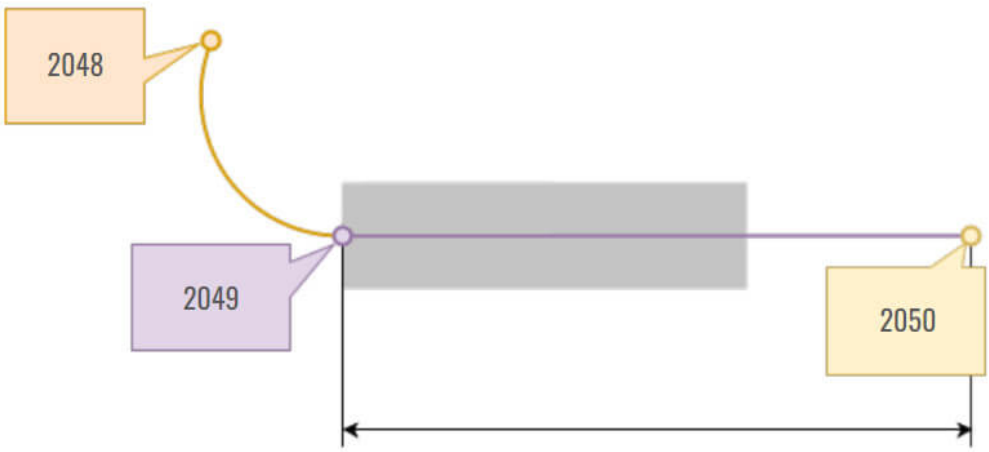



**Climbing route - Features variables**

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 1025 | Climb First Loiter Point | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1025** in **Climbing capture**. |
| 1026 | Climb Loiter Center | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1026** in **Climbing capture**. |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 1027 | Climb Start Loiter | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1027** in **Climbing capture**. |
| 1028 | Climb Finish Headturn | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1028** in **Climbing capture**. |
| 1029 | Climb Headturn Center | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1029** in **Climbing capture**. |
| 1030 | Climb Start Headturn | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1030** in **Climbing capture**. |
| 1031 | Climb Initial Point | [lon, lat, height] | [rad, rad, m] | Climbing Guidance Variable. Point **1031** in **Climbing capture**. |

> **Note**
>
> For further information regarding Climbing guidance, please refer to Climb - Guidance blocks of **Block Programs** section of the **1x PDI Builder** user manual.

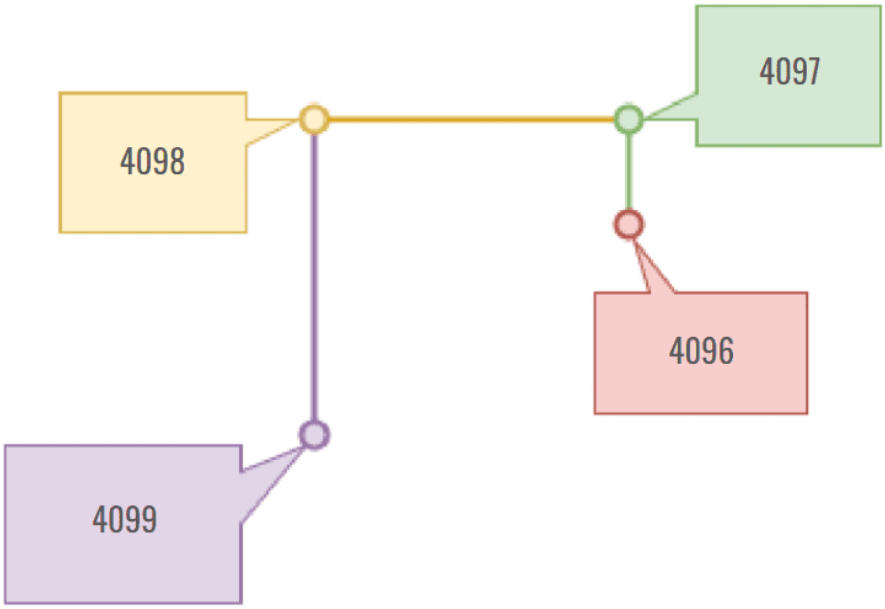



**Taxi route - Features variables**

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 2048 | Taxi Initial Point | [lon, lat, height] | [rad, rad, m] | Taxi Guidance Variable. Point **2048** in **Taxi capture**. |
| 2049 | Taxi Runway First Point | [lon, lat, height] | [rad, rad, m] | Taxi Guidance Variable. Point **2049** |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| | | | | in **Taxi capture**. |
| 2050 | Taxi Runway Final Point | [lon, lat, height] | [rad, rad, m] | Taxi Guidance Variable. Point **2050** in **Taxi capture**. |

> ⓘ **Note**
>
> For further information regarding Taxi guidance, please refer to Taxi - Guidance blocks of **Block Programs** section of the **1x PDI Builder** user manual.




**VTOL route - Features variables**

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 4096 | Vtol 00 | [lon, lat, height] | [rad, rad, m] | VTOL Initial point. Point **4096** in **VTOL route capture**. |
| 4097 | Vtol 01 | [lon, lat, height] | [rad, rad, m] | VTOL Translate Start. Point **4097** in **VTOL route capture**. |
| 4098 | Vtol 02 | [lon, lat, height] | [rad, rad, m] | VTOL Translate End. Point **4098** in **VTOL route capture**. |
| 4099 | Vtol 03 | [lon, lat, height] | [rad, rad, m] | VTOL End point. Point **4099** in **VTOL route capture**. |

> ⓘ **Note**
>
> For further information regarding VTOL guidance, please refer to VTOL - Guidance blocks of **Block Programs** section of the **1x PDI Builder** user manual.

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 3072 | Regvars 00 | [lon, lat, height] | [rad, rad, m] | Regvar Feature Variable. |
| 5120 - 5124 | Detour 00 - 04 | [lon, lat, height] | [rad, rad, m] | Detour Feature Variables. |
| 6144 | Runway Loiter | [lon, lat, height] | [rad, rad, m] | Runway Guidance Variable. |
| 6145 | Runway Touch Point | [lon, lat, height] | [rad, rad, m] | Runway Guidance Variable. |
| 6146 | Runway End Position | [lon, lat, height] | [rad, rad, m] | Runway Guidance Variable. |
| 7168 - 7169 | Gimbal Pointing 00 - 01 | [lon, lat, height] | [rad, rad, m] | Gimbal Pointing. |
| 8192 | UAV position | [lon, lat, height] | [rad, rad, m] | UAV position. |
| 8193 | Current phase | [lon, lat, height] | [rad, rad, m] | Current phase. |
| 8194 | | | | Desired position. |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| | Desired position | [lon, lat, height] | [rad, rad, m] | |
| 8195 | Value used when invalid ID is tried | [lon, lat, height] | [rad, rad, m] | Auxiliar feature - Not valid for users. |
| 8196 | Track position | [lon, lat, height] | [rad, rad, m] | Closer point in route to the current desired position. |
| 8197 | Operator position | [lon, lat, height] | [rad, rad, m] | Operator position. |
| 8198 | Start waypoint in current route | [lon, lat, height] | [rad, rad, m] | Start waypoint in current route. |
| 8199 | End waypoint in current route | [lon, lat, height] | [rad, rad, m] | End waypoint in current route. |
| 9216 - 9235 | Phase 00 - 19 | [lon, lat, height] | [rad, rad, m] | Phase. |
| 10240 - 10247 | Inflight Reference Point 00 - 07 | [lon, lat, height] | [rad, rad, m] | Absolute or relative reference position useful during mission. |
| 11264 - 11327 | Obstacle Sensor 00 - 63 | [lon, lat, height] | [rad, rad, m] | Obstacle Sensor. |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 12288 | Rendezvous 00 | [lon, lat, height] | [rad, rad, m] | Start point. For further information, please refer to **Rendezvous - Guidance blocks** of **Block Programs** section of the **1x PDI Builder** user manual. |
| 12289 | Rendezvous 01 | [lon, lat, height] | [rad, rad, m] | Rendezvous relative point. For further information, please refer to **Rendezvous - Guidance blocks** of **Block Programs** section of the **1x PDI Builder** user manual. |
| 12290 | Rendezvous 02 | [lon, lat, height] | [rad, rad, m] | Docking relative point. For further information, please refer to **Rendezvous - Guidance blocks** of **Block Programs** section of the **1x PDI Builder** user manual. |
| 13312 - 13327 | Moving Obstacles 00 - 15 | [lon, lat, height] | [rad, rad, m] | Moving Obstacle. |
| | | | | 1x PDI Builder allows the configuration of up to **6** |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 14336 - 14353 | Fixed runway features 00 - 17 | [lon, lat, height] | [rad, rad, m] | **runways**.<br><br>Each runway has 3 main points, **Initial point, End point**, and **Loiter Center**, which are repeated every three IDs.<br><br>E.g. Initial point for Runway 0 is 14336, for Runway 1 is 14339, for Runway 2 is 14342, etc. |
| 15360 - 15371 | Fixed spot features 00 - 11 | [lon, lat, height] | [rad, rad, m] | 1x PDI Builder allows the configuration of up to **6 spots**.<br><br>Each spot has 2 main points, **Spot coordinates** and **Loiter Center**, which are repeated every two IDs.<br><br>E.g. Spot coordinates for Spot 0 is 15360, for Spot 1 is 15362, for Spot 2 is **15364**, etc. |
| 28672 - 28703 | Moving Objects 00 - 31 | [lon, lat, height] | [rad, rad, m] | Moving Objects. |

| ID | Name | Form | Units | Description |
|---|---|---|---|---|
| 29696 - 30207 | Waypoint 00 - 511 | [lon, lat, height] | [rad, rad, m] | Route waypoints. |
| 30720 - 30987 | Volume vertex 00 - 267 | [lon, lat, height] | [rad, rad, m] | Volume vertices. |

## Navigation Variables

The following variables of the **Autopilot 1x** take part in the **navigation** performance.

| Type | ID | Name |
|---|---|---|
| Bit Variables | 101 | **No valid SRTM at UAV position** |
| | 114 | **No valid Geoid at UAV position** |
| Real Variables | 2 | **GS (Ground Speed)** |
| | 3 | **Heading** |
| | 4 | **Flight Path Angle** |
| | 5 | **Bank** |
| | 6 | **Yaw** |
| | 7 | **Pitch** |
| | 8 | **Roll** |

| Type | ID | Name |
|---|---|---|
| | 12 | p (Angular Velocity - X Body Axis) |
| | 13 | q (Angular Velocity - Y Body Axis) |
| | 14 | r (Angular Velocity - Z Body Axis) |
| | 15 | Forward Acceleration - X Body Axis |
| | 16 | Right Acceleration - Y Body Axis |
| | 17 | Bottom Acceleration - Z Body Axis |
| | 18 | RPM |
| | 19 | Front Ground Velocity |
| | 20 | Lateral Ground Velocity |
| | 21 | Velocity |
| | 12 | Forward Load Factor - X Body Axis |
| | 23 | Right Load Factor - Y Body Axis |
| | 24 | |

| Type | ID | Name |
|---|---|---|
| | | Bottom Load Factor - Z Body Axis |
| | 25 | Tangential Acceleration |
| | 26 | Estimated air density |
| | 27 | Gravity at UAV's position |
| | 28 | Co-Yaw |
| | 29 | Co-Pitch |
| | 30 | Co-Roll |
| | 31 | Angular Acceleration - X Body Axis |
| | 32 | Angular Acceleration - Y Body Axis |
| | 33 | Angular Acceleration - Z Body Axis |
| | 34 | Body to NED Quaternion qs |
| | 35 | Body to NED Quaternion qi |
| | 36 | Body to NED Quaternion qj |
| | 37 | |

| Type | ID | Name |
|---|---|---|
| | | Body to NED Quaternion qk |
| | 56 | Yaw Rate |
| | 57 | Pitch Rate |
| | 58 | Roll Rate |
| | 80 | Estimated gyro bias x |
| | 81 | Estimated gyro bias y |
| | 82 | Estimated gyro bias z |
| | 83 | Estimated accelerometer bias x |
| | 84 | Estimated accelerometer bias y |
| | 85 | Estimated accelerometer bias z |
| | 500 | Longitude |
| | 501 | Latitude |
| | 502 | WGS84 Elevation (Height Over the Ellipsoid) |
| | 503 | |

| Type | ID | Name |
|---|---|---|
| | | **MSL (Height Above Mean Sea Level) - Altitude** |
| | 504 | **AGL (Above Ground Level) - Height** |
| | 505 | **North Ground Velocity** |
| | 506 | **East Ground Velocity** |
| | 507 | **Down Ground Velocity** |
| | 1400 | **Velocity - X Body Axis** |
| | 1401 | **Velocity - Y Body Axis** |
| | 1402 | **Velocity - Z Body Axis** |
| | 2103 | **Acceleration North** |
| | 2104 | **Acceleration East** |
| | 2105 | **Acceleration Down** |
| Integer Variable | 25 | **SRTM source at UAV's position** |
| Feature Variable | 8192 | **UAV position** |

# List of PDI Errors

The following table explains the list of possible errors from Veronte applications.

> ⓘ **Note**
>
> The decimal value of the PDI Error Source (UVar 50) represents the PDI error ID listed in the following table.

| Code | Nº | Explanation |
|---|---|---|
| err_ok | 0 | No errors detected |
| err_gpio | 1 | GPIOs function configuration |
| err_odt_pool_sz | 2 | Incorrect pool size in on-demand telemetry |
| err_telemetry_alloc | 3 | Could not allocate new telemetry vector |
| err_patch_type | 4 | Incorrect patch type |
| err_patch_needs_next | 5 | Selected type of patch needs a next |
| err_max_poly_evt | 6 | Maximum limit of polygon events reached |
| err_channelmgr | 10 | Channel manager configuration |
| err_blk_cmp_pdi | 11 | Incorrect deserialization of compiled block data |
| err_blk_cmp | 12 | Incorrect output in compiled block |

| Code | Nº | Explanation |
|---|---|---|
| err_blk_cmp_desc | 13 | Invalid pin description in compiled block |
| err_sara | 15 | SARA sim type oor. |
| err_vblk_sensrtm | 16 | Block for SRTM sensor |
| err_arcx | 23 | Arcade axis set of options. |
| err_msg8_consumer_hi | 24 | Custom message consumer cannot be used in HI unless it is an external sensor |
| err_can_consumer_hi | 25 | CAN Custom message consumer cannot be used in HI unless it is an external sensor |
| err_modes | 27 | Stick configuration modes |
| err_blkekfstp | 41 | Static pressure to EKF adapter block |
| err_gnss_blocks | 45 | GNSS constellations configuration (more than allowed) |
| err_cansuite_gpio | 47 | CAN suite gpio |
| err_vrng | 48 | Range sensors |
| err_fmset | 50 | Custom message set |
| err_pwm | 54 | Pwm configuration |
| err_sniffer | 63 | Sniffer wires configuration |

| Code | № | Explanation |
|---|---|---|
| err_sniffer_read_only | 64 | Read-only variable selected in sniffer |
| err_fmsgc_read_only | 65 | Read-only variable selected in serial message consumer |
| err_canmsgc_read_only | 66 | Read-only variable selected in CAN message consumer |
| err_vref_read_only | 67 | Read-only vref variable |
| err_obstacle | 68 | Incorrect type of obstacle. |
| err_obsense | 69 | Obstacle sensing mode or type oor. |
| err_marks | 71 | Incorrect type of mark. |
| err_ext_nav_sen | 72 | Incorrect external navigation sensor configuration. |
| err_tunnel8 | 73 | Wrong tunnel configuration. |
| err_fmsg_p | 74 | Custom message producers msg id oor. |
| err_fmsg_c | 75 | Custom message consumers process parser oor. |
| err_fmsgcan_c | 76 | CAN custom msg consumer msg id oor. |
| err_telem | 77 | Telemetry configuration |
| err_fmsg_p_sz | 78 | Custom message producer occupancy is higher than allowed |

| Code | № | Explanation |
|---|---|---|
| err_fmsg_c_sz | 79 | Custom message consumer occupancy is higher than allowed |
| err_sci | 81 | SCI config error |
| err_events | 82 | Invalid event |
| err_actions | 83 | Actmgr - List of actions. |
| err_evact | 84 | Actmgr - List of related events and actions. |
| err_cmd_not_allowed | 85 | Commands not allowed |
| err_xpc_can_in | 87 | XPC for CAN messages input filters size ok |
| err_xpc_can_out | 88 | XPC for CAN messages output filters size ok |
| err_xpc_can_ser | 89 | XPC for CAN messages serialtocan size ok |
| err_xpc_can_gpio | 90 | XPC for CAN messages virtual gpios size ok |
| err_xpc_can_map | 91 | XPC for CAN messages and check their priority and connections. |
| err_xpc_u8_map | 92 | XPC for u8 messages and check their priority and connections. |
| err_internest | 93 | Internest version in rage |
| err_internest1 | 94 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Internest max_range_vbase in rage |
| err_internest2 | 95 | Internest max_range_vexplore in rage |
| err_gravity_ext | 96 | Incorrect gravity extractor for AHRS configuration |
| err_u8pkrsarray_size | 97 | Incorrect size for U8pkrsarray |
| err_ecap | 101 | Capture. |
| err_cappulse | 116 | ECAP pulse consumers |
| err_i2cdevs | 117 | I2C external devices |
| err_lossy_resize | 120 | Lossy resize error |
| err_rvector_resize | 121 | Rvector resize error |
| err_asciiparser | 122 | ASCII parser invalid configuration |
| err_telemetry_exceeded | 123 | Telemetry size exceeded |
| err_hi_3210_rx_tout | 154 | HI-3210 rx cannot be configured |
| err_hi_3210_tx_tout | 155 | HI-3210 tx cannot be configured |
| err_arbiter_cfg | 156 | Arbiter production configurable invalid |
| err_can_arbiter_cfg | 157 | CAN Arbiter configuration has not consistency with production file or/and it is invalid |

| Code | Nº | Explanation |
|---|---|---|
| err_cmd_rdvzset | 176 | Rendezvous command base_yaw oor |
| err_cmd_taxiget | 183 | Taxi guidance request command. |
| err_cmd_gtrack1 | 188 | Invalid detour command |
| err_cmd_gtrack2 | 189 | Invalid guidance block configuration. |
| err_cmd_speed | 192 | Cruise speed command |
| err_cmd_gtrack | 193 | Invalid detour command |
| err_cmd_gtrkset | 194 | Track request command |
| err_cmd_stksrcr | 208 | Get stick raw channels from selected source |
| err_cmd_vtolset | 212 | VTOL request command |
| err_ini_nok | 213 | Cannot change to a phase different from INI with System BIT not OK and out of PDI mode |
| err_cmd_nav | 215 | Navigation command |
| err_cmd_gpio | 218 | GPIO command |
| err_cmd_gpio1 | 219 | GPIO command |
| err_cmd_gpio2 | 220 | GPIO command |
| err_cmd_gpio3 | 221 | GPIO command |
| err_cmd_phase | 222 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Commanded phase is out of range |
| err_cmd_gimbal1 | 224 | Gimbal commands |
| err_cmd_gimbal | 225 | Gimbal commands |
| err_cmd_var | 235 | Variable set command |
| err_reset | 239 | Reset CPU IRX |
| err_acc2filt | 257 | Internal 2 (IMU) Accelerometer filter |
| err_imu3_filter | 258 | Internal 3 (IMU) filter not in range [0,6] |
| err_imu3_filter_bw | 259 | Internal 3 (IMU) filter not compatible with Bandwidth limit |
| err_imu3_delta | 260 | Internal 3 (IMU) invalid rate limit |
| err_imu2_delta | 261 | Internal 2 (IMU) invalid rate limit |
| err_cansuite_in | 288 | CAN suite producer for veronte |
| err_cansuite_out | 289 | CAN suite consumer for veronte |
| err_cfg_can | 290 | CAN cfg |
| err_resize_can_cex | 291 | CEX CAN cfg |
| err_resize_can_commex | 292 | COMMEX CAN cfg |
| err_jeti_and_lift | 293 | Trying to configure jeti and lift (not enough memory) |

| Code | Nº | Explanation |
|---|---|---|
| err_relf | 500 | Tried to set feature relative to invalid feature, or set relative an "absolute-only" feature |
| err_jid | 501 | Invalid feature |
| err_canid | 502 | Invalid CAN id |
| err_cfgid_mode0 | 503 | Invalid Cfgid PDI (number of PDIs does not match) |
| err_cfgid_mode1 | 504 | Invalid Cfgid PDI mode |
| err_cmd_mgr | 505 | Expected command size does not match |
| err_cmd_mgr1 | 506 | Expected command size does not match |
| err_cancfg | 507 | Invalid CAN configuration |
| err_decimator | 508 | Invalid decimator |
| err_sci_cfg | 509 | Invalid SCI configuration |
| err_field1 | 510 | Maximum ID of real variable exceeded |
| err_field2 | 511 | Maximum ID of user variable exceeded |
| err_field3 | 512 | Maximum ID of bit variable exceeded |
| err_field4 | 513 | Maximum number of decimals for real variable exceeded |

| Code | Nº | Explanation |
|---|---|---|
| err_field5 | 514 | Overflow for real variable detected |
| err_field6 | 515 | Incorrect CRC field |
| err_field7 | 516 | Field matcher number of bits outside range |
| err_field8 | 517 | Field maximum skippable bits exceeded |
| err_field9 | 518 | Maximum ID of real variable saved as string exceeded |
| err_field10 | 519 | Field type out of range |
| err_flogic | 520 | Invalid event composition (Flogic) |
| err_flogic1 | 521 | Invalid event composition (Flogic) |
| err_flogic2 | 522 | Invalid event composition type |
| err_fref | 523 | Invalid type of position reference |
| err_irxtable | 524 | Invalid 3Dtable mode or vector is non-decreasing |
| err_limit | 525 | Invalid limit event type |
| err_lsm6ds3_cfg | 526 | Accelerometer/Gyroscope settings outside range |
| err_pdi_ver | 527 | Incompatible PDI version, there are some PDI files in Veronte |

| Code | Nº | Explanation |
|---|---|---|
| | | from a different version. Try migrating offline and uploading a complete migrated configuration |
| err_rvarsensor | 528 | Id for Rvar out of range for Rvarsensor |
| err_stickrawtrans0 | 529 | K value in stick outside range [-100, 100] or 0 |
| err_stickrawtrans1 | 530 | Maximum value read from stick for Configured range exceeded [4095] |
| err_stickrawtrans2 | 531 | Maximum value read from stick for Raw stick trim exceeded [4095] |
| err_stickrawtrans3 | 532 | Invalid transformation type for stick |
| err_inc_counter | 533 | Invalid ID for incremental counter |
| err_stickcfg3 | 536 | Invalid destination of stick device data |
| err_tllhcompressed | 537 | Longitude/Latitude outside range [-pi, pi]/[-0.5pi, 0.5pi] |
| err_tunpatchset0 | 538 | Patch selected as first has not been enabled |
| err_tunpatchset1 | 539 | Patch selected as next has not been enabled |

| Code | Nº | Explanation |
|---|---|---|
| err_tunpatchset2 | 540 | Patch type does not exist |
| err_tunpatchset3 | 541 | The route cannot have two consecutive patches of type point |
| err_tunpatchset4 | 542 | Patchtype orthodrome has not been enabled |
| err_tunpatchset5 | 543 | Patchtype arc has not been enabled |
| err_tunpatchset6 | 544 | Patchtype ellipse has not been enabled |
| err_tunpatchset8 | 546 | No patchtype has been enabled |
| err_Ubxcfgnav5 | 547 | Dynmodel out of range or incorrect UTC time |
| err_Ubxcfgnavx5 | 548 | Maximum acceptable AssistNow Autonomous orbit error outside range [5, 1000] |
| err_Ubxcfgport | 549 | Port (for Ubx?) is neither SPI nor SCI |
| err_Ubxcfgrate | 550 | Invalid Ublox configuration rate |
| err_Ubxcfgsbas | 551 | Maximum number of SBAS prioritized tracking channels exceeded [3] |
| err_atunarray0 | 552 | Invalid Tunarray index |

| Code | Nº | Explanation |
|---|---|---|
| err_atunarray1 | 553 | Invalid Tunarray size |
| err_Ubxcfgtmode3 | 554 | Error in receiver mode, neither enabled nor disabled |
| err_Uclk | 555 | Invalid chrono event |
| err_Uvarsensor | 556 | Id for Uvar out of range for Uvarsensor, or desired frequency too low (<1Hz) |
| err_Uclkmgr | 557 | Maximum number of event user chronos exceeded |
| err_varinit0 | 558 | Maximum array size exceeded on initial values for user variables |
| err_varinit1 | 559 | Initialized variable is unwritable |
| err_vref0 | 560 | Maximum ID of Rvar variable exceeded in Vref |
| err_vref1 | 561 | Maximum ID of Uvar variable exceeded in Vref |
| err_vref2 | 562 | Maximum ID of Bvar variable exceeded in Vref |
| err_vref3 | 563 | Invalid type of variable in Vref |
| err_xclkcfg0 | 564 | Period time non positive in event |
| err_xclkcfg1 | 565 | Invalid period mode |
| err_xclkcfg2 | 566 | Chrono position direction not correctly normalized |

| Code | Nº | Explanation |
|---|---|---|
| err_xclkcfg3 | 567 | Invalid type of chrono |
| err_blk_batch | 570 | Maximum allowed block nesting depth exceeded [6] or incorrect number of inputs/outputs for block Patch |
| err_blk_ifelse | 571 | Error in the connections for block if/else |
| err_blk_switch | 572 | Error in the connections for block switch |
| err_blk_switch0 | 573 | Invalid switch/ifelse/phase block configuration |
| err_blkmgr | 574 | Invalid block manager configuration |
| err_pinmux | 576 | Invalid switch/ifelse/phase block output configuration |
| err_blk_switchmap | 577 | Invalid mapping to cases in switch/phase block |
| err_accellimit | 579 | Envelope's falling or rising edge is out of accepted limits |
| err_polygrp | 581 | Index of polygon for group outside of range |
| err_polygon | 582 | Less than 3 polygon vertices |
| err_circle | 583 | Circle radius is less than or equal to 0 |

| Code | Nº | Explanation |
|---|---|---|
| err_height | 584 | Height type is neither relative nor absolute |
| err_heightabs | 585 | Invalid absolute height type |
| err_rwy | 586 | Invalid runway preferred type |
| err_driver | 588 | Problem in Driver block configuration |
| err_mwk | 592 | Gyroscope measurement error |
| err_opinctrl | 593 | Invalid PID controller input type |
| err_pid | 594 | Invalid PID integral configuration (tau must be >= 0) |
| err_prediction | 595 | Error in the Model Prediction Control algorithm. Prediction Horizon out of range or zero diagonal matrix R |
| err_sysid | 596 | Error ID for given pdi check |
| err_tsched | 597 | Error ID for given pdi check |
| err_dwma | 598 | Error ID for given pdi check |
| err_iir | 599 | Error ID for given pdi check |
| err_butterworth | 600 | Error ID for given pdi check |
| err_usre2 | 601 | Error incorrect user sensor variance |
| err_ubxcfgtp5 | 603 | Ublox time pulse configuration |

| Code | Nº | Explanation |
|---|---|---|
| err_cfgmgr_load_secure | 604 | Error loading secure mode |
| err_cfgmgr_finit | 605 | Error PDI files |
| err_cfgmgr_timeout | 606 | Error; timeout while loading PDIs |
| err_invalidrotmat | 607 | Invalid rotation matrix (cannot be inverted) |
| err_flyto_setup | 608 | Error in fly to waypoint action |
| err_vblk_apsel | 609 | Invalid block AP selection configuration, channel exceeds maximum number |
| err_vblk_arcade_bounce | 610 | Error in the connections for block Arcade Bounce |
| err_vblk_arcade_extend | 611 | Error in the connections for block Arcade Extend |
| err_vblk_btor | 612 | Error in the connections for bool to real block |
| err_vblk_bound | 613 | Error in the connections for block Bound |
| err_rldcfg0 | 614 | Invalid dynamic pressure EKF entrance configuration |
| err_ubx_tout0 | 616 | Could not receive ACKs from UBlox |
| err_ubx_tout1 | 617 | Could not receive polling from Ublox |

| Code | Nº | Explanation |
|---|---|---|
| err_ubx_nack | 618 | A Ublox configuration message was rejected by a Ublox device (GNSS) |
| err_guid_pid | 619 | Invalid type of guidance controller |
| err_cmd_leg | 620 | Guidance uses an invalid runway or site |
| err_mixarray | 622 | Error in mixarray construction (possibly there is not enough RAM memory to store all the blocks) |
| err_xrtable | 623 | Invalid number of entries for XrTable |
| err_blk_varset | 624 | Block trying to write in an invalid variable, possibly the selected variable is not user writable |
| err_tuntrait | 625 | Error trying to resize an array out of its maximum size |
| err_asuite | 626 | Selected dynamic pressure sensor is not valid in this hardware version |
| err_xpcmap | 627 | Invalid producer/consumer in I/O connections |
| err_blk_arraysplit | 628 | Invalid block: array of less than 2 elements cannot be split |

| Code | Nº | Explanation |
|---|---|---|
| err_blk_array | 629 | Bundle block error, it must have more than one input and the input sizes must be one |
| err_vblk_varget | 630 | Invalid ID for block Read Real |
| err_vblk_vec_ops | 631 | Error in either; Vector: Add, Subtract, Cross product or Matrix rotation |
| err_autotune | 633 | Invalid maximum duration of autotuning process or invalid number of stages for FFT |
| err_vblk_azeld1 | 634 | Error in the connections for block azeld -> xyz |
| err_vblk_azeld | 635 | Error in the connections for block xyz -> azeld |
| err_vblk_dot | 636 | Error in the connections for block Dot Product |
| err_vblk_enctrl | 637 | Error in the connections for block Energy Control or invalid conversion factor from speed difference to desired acceleration |
| err_vblk_bnxb1 | 638 | Error in the connections for block(s) AND/OR |
| err_vblk_r1xr1 | 639 | Error in the connections for block |
| err_vblk_r2xr1 | 640 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Error in the connections for block x+y or invalid subfunction for the block |
| err_vblk_rnxr1 | 641 | Error in the connections for block(s) Multiply/Add Elements/ Norm or invalid subfunction for the block(s) |
| err_vblk_iir | 642 | Error in the connections for block IIR Filter or invalid parameters for the transfer function |
| err_vblk_kmultvec | 643 | Error in the connections for block Scale |
| err_vblk_manual | 644 | Error in the connections for block Manual or invalid stick control channel |
| err_vblk_minmax | 645 | Error in the connections for block(s) Min/Max |
| err_vblk_mix | 646 | Error in the connections for block MIX or invalid mix control channel |
| err_vblk_movern | 647 | Error in the connections for block MIX Move |
| err_vblk_not | 648 | Error in the connections for block NOT |
| err_vblk_phase | 649 | Default case does not exist for block Phase Switch |

| Code | Nº | Explanation |
|---|---|---|
| err_vblk_tsched | 651 | Error in the connections for block T-Sched PID |
| err_vblk_pid | 652 | Invalid configuration or connection of a PID block |
| err_vblk_poly | 653 | Error in the connections for block Polynomial |
| err_vblk_posset | 654 | Error in the connections for block Write Feature or Fid is not user writable |
| err_vblk_predictive | 655 | Error in the connections for block Predictive Control or number of elements for numerator/ denominator unmatched to the expected input size |
| err_vblk_ramp | 656 | Error in the connections for block Ramp or rise time/settling time less than (or equal to) 0 |
| err_vblk_matvec | 657 | Error in the connections for block Linear Transformation or matrix size unmatched to the expected input size |
| err_vblk_rtable3d | 658 | Error in the connections for block 3D Table Interpolation |
| err_vblk_rtob | 659 | Error in the connections for block Real to Bool |
| err_vblk_rtou | 660 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Error in the connections for block Real to Integer |
| err_vblk_runwrap | 661 | Error in the connections for block [-pi,pi] Unwrap |
| err_vblk_utor | 662 | Error in the connections for block Integer to Real |
| err_vblk_relthis | 663 | Error in the connections for block Relative Vector |
| err_cancfg1 | 664 | Number of mailboxes dedicated to rx exceeds maximum [32] or the filter applied to mailbox subset exceeds maximum filter id |
| err_stickvar_cfg | 665 | Decimate time is higher than the minimum period or number of stick virtual inputs exceeds maximum configured for block Virtual stick |
| err_vblk_gimbal | 666 | Error in the connections for block Gimbal |
| err_vblk_hysteresis | 667 | Error in the connections for block Hysteresis |
| err_vblk_arctrim | 668 | Error in the connections for block Arc Trim or control vector unmatched to expected size |
| err_blockprog | 669 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Incomplete set of LSB bits or with bit holes for execution mask or slot is not within the mask |
| err_vblk_n2b | 670 | Error in the connections for block NED to Body/Body to NED |
| err_vblk_pwm | 671 | Error in the connections for block PWM or PWM id exceeds maximum |
| err_vblk_stick | 672 | Error in stick block, connections, dimensions of matrices or stick sources could be wrong |
| err_vblk_u2s | 673 | Error in actuator block, connections or dimensions of matrices could be wrong |
| err_vblk_interp | 674 | Error in vector interpolation block, connections or sizes could be wrong, also the points in the table must be sorted in increasing order of x |
| err_vblk_ratelim | 678 | Error in the connections for block Rate limiter |
| err_vblk_clock | 679 | Unable to reset the clock timer in block Clock |
| err_vblk_mult_varget | 680 | Unable to initialize output vector or invalid variable id in block Read Multiple Reals |

| Code | Nº | Explanation |
|---|---|---|
| err_vblk_mult_varset | 681 | Error in the connections for block Write Multiple Bits/Write Multiple Reals or input vector different from input variables or variable not user writable |
| err_vblk_pid_static | 682 | Unable to subscribe autotune in block PID |
| err_vblk_quatctrl | 683 | Set of configurable variables cannot be 0 or outside their range in block Quaternion Control |
| err_vblk_senstp | 685 | Error in pressure sensor block, could be that the selected pressure sensor in invalid in the current hardware or that the configured variance is negative or zero |
| err_vblk_sengnss | 686 | Error for block GNSS sensor |
| err_vblk_ekfpos | 687 | Error for block EKF position |
| err_vblk_ekfvel | 688 | Error for block EKF Velocity |
| err_vblk_ekfmis | 689 | Error for block EKF Misalignment |
| err_vblk_drnmis | 690 | Error for block EKF GNSS compass |
| err_vblk_senrel | 691 | Error for block Relative position (Sensors) |

| Code | Nº | Explanation |
|---|---|---|
| err_vblk_ekfdem | 692 | Error for block EKF Terrain Height |
| err_vblk_senmag | 693 | Error in magnetometer sensor block, the selected might be invalid in your current hardware or the configured variance is negative or zero |
| err_mdg_gain | 694 | Error for block Madgwick Gain Computer |
| err_vblk_senalt | 696 | Error for block Altimeter |
| err_vblk_ekfalt | 697 | Error for block EKF Altitude |
| err_vblk_ekfvdn | 698 | Error for block EKF Velocity Down |
| err_vblk_nav | 699 | Error for block Navigation |
| err_e2acc | 700 | Error for variance increment due to high acceleration |
| err_vblk_ekfsplit | 701 | Error for block EKF Split |
| err_vblk_fft | 703 | Error ID for block FFT |
| err_vblk_ecu | 705 | Error ID for block ECU control |
| err_vblk_fuzzy | 706 | Error ID for block Fuzzy Logic Controller |
| err_vblk_guidance | 707 | Input of guidance block could not be connected |
| err_vblk_sysid | 709 | Error ID for block System Identification |

| Code | Nº | Explanation |
|---|---|---|
| err_cex_pwm | 710 | Error ID for CEX pwm arbitration, src ID greater than pulses array |
| err_cex_esc_tm | 711 | Error ID for CEX ESC period |
| err_cex_mcu_tm | 712 | Error ID for CEX MCU period |
| err_vblk_climb | 713 | Incorrect climb block operation |
| err_vblk_leg | 714 | Incorrect leg block operation |
| err_flyto | 715 | Incorrect fly to command (non-existing patch) |
| err_vblk_approach | 716 | Incorrect approach block operation |
| err_vblk_yawing | 717 | Incorrect yawing block configuration |
| err_vblk_siggen | 718 | Incorrect signal generation configuration |
| err_vblk_pnav | 719 | Incorrect PNAV guidance configuration |
| err_vblk_genex | 720 | Incorrect GENEX guidance configuration |
| err_vblk_modpnav | 721 | Incorrect ModPNAV guidance configuration |
| err_blk_lib | 722 | Incorrect library |
| err_vblk_ewma | 723 | Incorrect EWMA block configuration |

| Code | Nº | Explanation |
|---|---|---|
| err_uarray_resize | 724 | Incorrect uarray resize |
| err_oprvar | 725 | Incorrect operation/setup rvar configuration |
| err_block_const | 726 | Error in block const |
| err_block_posget | 727 | Error in block posguet |
| err_block_pnavbase | 728 | Error in block pnav base |
| err_block_arcade0 | 729 | Error in block arcade0 |
| err_unescape | 730 | Error in escape itport |
| err_initial_alignment | 731 | The internal AHRS or EKF navigation estimation algorithm could not compute an initial orientation. Try commanding an initial yaw or adding an automation to do so |
| err_fft_block_disable | 732 | The FFT block is temporarily disabled in this version |
| err_vblk_acclim | 733 | Error in block acceleration limiter |
| err_ewma_avgvar | 734 | Error in EWMA average/variance time constants |
| err_sensor_fusion | 735 | Time constants for sensor fusion algorithm are incorrect |
| err_oprng | 736 | Error in operation range configuration |

| Code | Nº | Explanation |
|---|---|---|
| err_oprng_check | 737 | Error in operation range check |
| err_vgeoref | 738 | Error in vgeoref configuration |
| err_notch_filter | 739 | Incorrect notch filter parameters |
| err_notch_frequency | 740 | Incorrect notch filter frequency |
| err_geoid_version | 741 | Incorrect geoid version in SD |
| err_vblk_integrator | 742 | Error in the connections for block Integrator |
| err_vblk_derivative | 743 | Error in the connections for block Derivative |
| err_wrapper_ref | 744 | Incorrect envelope range (minimum must be less or equal than maximum) |
| err_sensor_fusion_sel | 745 | Selected gyroscopes or accelerometers are invalid in this hardware or the default sensor is not active |
| err_volume_id | 746 | Incorrect volume identifier |
| err_fload_missing | 747 | Detected missing file at fload blocking |
| err_hi_3210_rx_cfg | 748 | Incorrect HI-3210 RX configuration |
| err_mixer | 749 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Incorrect mixer block configuration or it does not fit in memory |
| err_scheduler_data | 750 | Incorrect data size for scheduler block |
| err_asc | 751 | Incorrect configuration of ASC block |
| err_scheduler | 752 | Incorrect configuration of scheduler block |
| err_aacg | 753 | Incorrect data size for mixer block |
| err_mat2quat | 754 | Incorrect input to matrix to quaternion block |
| err_quat2mat | 755 | Incorrect input to quaternion to matrix |
| err_wrench_frame | 756 | Incorrect inputs to the wrench frame block |
| err_ffc_3d_block | 757 | Incorrect configuration of ffc3d block |
| err_srv_def_limits | 758 | A default position is not within X limits for servo actuator |
| err_pfield_file | 759 | Error in loading the saved fields |
| err_vblk_wind | 760 | Invalid configuration for wind estimation block |

| Code | Nº | Explanation |
|---|---|---|
| err_vblk_senqinf | 761 | Invalid configuration for dynamic pressure sensor block |
| err_vblk_code_size | 762 | External code block: Size greater than maximum allowed (64KW) |
| err_vblk_code_empty | 763 | External code block empty. No code has been loaded |
| err_vblk_code_ptr | 764 | External code block: Invalid pointer detected |
| err_vblk_code_inputs | 765 | External code block: Inputs read error |
| err_vblk_code_outputs | 766 | External code block: Outputs write error |
| err_vblk_ekfadapter | 767 | Incorrect configuration in custom EKF adapter |
| err_vblk_compiledsil | 768 | Tried to use the compiled code block on a SIL configuration |
| err_event_log | 770 | Invalid Event-Driven Log Fields configuration |
| err_onboard_log | 771 | Invalid On-Board Low Rate Log Fields configuration |
| err_fast_log | 772 | Invalid Fast Log Fields configuration |
| err_updt_magfield | 773 | Magnetic field's partition not updated correctly |

| Code | Nº | Explanation |
|---|---|---|
| err_updt_geoid | 774 | Geoid's partition not updated correctly |
| err_op_var_def_val | 775 | Operation variable default value outside of range |
| err_arbitration | 10000 | Error ID for Arbitration cfg |
| err_arbitration_can | 10001 | Error ID for Arbitration_can cfg |
| err_arbitration_can1 | 10002 | Error ID for Arbitration_can cfg |
| err_arb_cfg0 | 10003 | Error ID for Arb cfg preferred ap oor |
| err_arb_cfg1 | 10004 | Error ID for Arb cfg method oor(out of range) |
| err_arb_cfg2 | 10005 | Error ID for Arb cfg tmin oor |
| err_arb_cfg3 | 10006 | Error ID for Arb cfg hysteresis oor |
| err_ap_nvars | 10007 | Error ID for Autopilot nvars oor |
| err_apcfg_nvars | 10008 | Error ID for Autopilot cfg nvars oor |
| err_jetibox | 10009 | Error ID for sci identifier of Jetibox cfg oor |
| err_jetibox_fmsgcmd | 10010 | Error ID for jetibox fmsg cmd oor |
| err_arb_init_time | 10011 | Error ID for Arbiter Power Init Time less than 0 |
| err_arb_varcfg | 10013 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Incorrect arbiter variable configuration |
| err_hs_base_can_id | 15000 | High speed telemetry invalid Base CAN Id |
| err_hs_tm_nvars | 15001 | High speed telemetry number of variables too big |
| err_vmc_motor | 20000 | Motor cfg is not valid |
| err_vmc_control_mode | 20001 | Control mode is invalid |
| err_vmc_encoder_nbits | 20002 | Number of bits for encoder is invalid |
| err_mc_vmotor | 20003 | Virtual motor cfg invalid |
| err_mc_smo | 20004 | Slide Mode Observer cfg invalid |
| err_mc_control | 20005 | Control cfg invalid |
| err_mc_fault_detection | 20006 | Invalid fault detection limits |
| err_mangle_rate | 20007 | Invalid filter time constant |
| err_low_pll | 20008 | Invalid cut-off frequency |
| err_mc_main | 20009 | Invalid real time frequency |
| err_range_check | 20010 | Invalid range definition |
| err_ext_sens | 20011 | Invalid external sensor configuration |
| err_press_dev | 20012 | |

| Code | Nº | Explanation |
|---|---|---|
| | | Invalid pressure sensor configuration |
| err_press_dps310 | 20014 | Invalid pressure sensor Internal 2 configuration |
| err_press_hsc | 20015 | Invalid pressure sensor Internal 0 configuration |
| err_press_ms56 | 20016 | Invalid pressure sensor Internal 1 configuration |
| err_invalid_cal_table | 20017 | Invalid number of entries in calibration table |
| err_mag_lis3mdl | 20018 | Invalid magnetometer LIS3MDL configuration |
| err_mag_hscdtd | 20019 | Invalid magnetometer HSCDTD008A configuration |
| err_mag_mmc5883ma | 20020 | Invalid magnetometer MMC5883MA configuration |
| err_mag_rm3100 | 20021 | Invalid magnetometer RM3100 configuration |
| err_ex_ussa76_cmd | 20022 | Invalid period of external USSA76 command |
| err_cfgmr_length | 31999 | Unexpected size of PDI or command |
| err_check_test | 65535 | Error ID for given pdi check |

# List of File System Errors

Integer variable File system status (UVar 96) represents several **DFS2 FS-related error states**, as each of its 16 bits indicates a specific error condition.

| BIT | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| STATUS | | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 | 0/1 |

**File system status**

Below is a list of potential error conditions associated with each bit:

| ID | Error description |
|---|---|
| 0 | Index sector is not correct (1 if not correct - 0 if correct) |
| 1 | Error initializing SD (1 if not correct - 0 if correct) |
| 2 | Num partition bigger than allowed (1 if bigger - 0 if not) |
| 3 | No more descriptor available (1 if not available - 0 if available) |
| 4 | The descriptor of a file was not correct (CRC failed) (1 if fail - 0 if not) |
| 5 | The new descriptor couldn't be created (1 if not created - 0 if created) |
| 6 | Error updating the file size (1 if error updating - 0 if no error) |
| 7 | Error formatting because of system description size (1 if error formatting - 0 if not) |

| ID | Error description |
|---|---|
| 8 | Error formatting driver not initialized (1 if error formatting - 0 if not) |
| 9 | Error formatting maximum partition number overpassed (1 if error formatting - 0 if not) |
| 10 | Error formatting not able to write index sector (1 if error formatting - 0 if not) |
| 11 | Error formatting partition not enabled (1 if error formatting - 0 if not) |
| 12 | Error writing the index for the log controller (1 if error formatting - 0 if not) |
| 13 | File was destroyed but was not closed or had pending requests (1 if not closed or pending requests - 0 if closed or had no pending requests) |
| 14 | Unable to read a sector (1 if unable - 0 if able) |

# Firmware Changelog

This section presents the changes between firmware versions of Veronte Autopilot 1x.

## 7.8.18

This section presents the changes between the previous firmware version of Veronte Autopilot 1x, **v.7.6.52**, and this firmware version, **v.7.8.18**. For further details, please consult the Service Bulletin nº 00022.

### Added

- Added new block for triggering serial custom message production.
- Added new block to trigger CAN custom message generation.
- Added new block to split a position into its longitude, latitude and WGS84 height as Real64.
- Added new block to trigger CAN producer messages.
- Added new block to trigger serial custom messages.
- Added new Blocks for handling 64-bit mathematical operations.
- Added new variables EKF variances for all states.
- Added new Built in test to indicate initialization of navigation.
- Added new groups of operation custom variables and a block to obtain the position of the closest one.
- Added new PDI for ethernet network configuration.
- Added the capability of relay configured messages over the receiving port.
- Added White list in routing table in order to allow some messages to be routed over the receiving port.
- Added cyrpto support for all STANAG 4586 communications.
- Added configuration to request data link discovers and stablish a connection if possible.

### Changed

- Added group marks. Now mark events can use groups of marks configured in operation.

- Send automatically subsystem alert STANAG 4586 message when the status of a component and subcomponent of a subsystem is higher than nominal.

**Removed**

- SFRBX message support in GNSS receiver as not used.
- Raw GNSS message configuration for GNSS receiver

## 7.8.32

This section presents the changes between the previous firmware version of Veronte Autopilot 1x, **v.7.8.18**, and this firmware version, **v.7.8.32**. For further details, please consult the Service Bulletin nº00023.

**Added**

- Added PDI error number 73 "when tunnel configuration is not valid".
- Added system bits to know when there has been a teleport (reset) in the EKF. BIT 141 (EKF height not teleported) and BIT 142 (EKF lonlat not teleported).
- Support for Soft Iron magnetometer calibration.
- Geographic north calibration in magnetometer.
- Added variables: System Uvar 117 (GNSS1 Fix Type) and Uvar 167 (GNSS2 Fix Type) to publish the Ublox fix type for GNSS1 and GNSS2 modules respectively. The possible values are: 0 (no fix), 1 (dead reckoning only), 2 (2D fix), 3 (3D fix), 4 (GNSS + dead reckoning combined) and 5 (time only fix).
- Added reset in EKF height estimation under certain conditions to prevent false jumps in estimated vertical velocity.
- Added improved EKF reset logic. Measurements must first pass the NIS check before being considered for teleporting. Height teleport occurs when innovation exceeds 20m, state variance exceeds 100m², state variance is greater than 2× sensor variance, and height dead-reckoning time exceeds 5s. The height state is reset to measurement value with 10× measurement variance. Horizontal teleport activates when innovation exceeds 20m, state variance exceeds 100m, state variance is greater than 2× sensor variance,

and horizontal dead-reckoning time exceeds 5s. This operates only when full EKF state is operational and resets north-east position states with 10× measurement variance.
- Added lead time brake of 1 second to compensate the control delay and reach the desired waypoint speed over the waypoint.

**Removed**

- Bit variable 329 (3.3V Power Source Bit) as not being used

**Fixed**

- Fixed ADSB MAVLink input via internal transponder.
- Fixed messages lost when RTCM producer is connected to a Tunnel Consumer.
- Fixed duplicated ACKs.
- Fixed features writing zeros in fast logs.
- Fixed system Ready to Start BIT correctly updated.
- Fixed landing waypoint numbers recovered.
- Fixed where the Earth's rotation correction was incorrectly applied to angular velocity data coming from external navigation sources and simulators, which could cause small attitude and rate errors in those modes.
- Fixed block triggering of CAN messages only sent a message with the change from disabled to enabled the first time
- Fixed when using VectorNav VN300 as a navigation source, the UAV navigation position was being updated even when the received navigation from VectorNav did not have a valid position fix. Now the position is only updated when the VN300 has a valid position fix
- Fixed EKF teleport BITS are now only true in the steps where there is a teleport instead of being true until the next update step
- Fixed body acceleration computation when in navigation simulation mode
- Fixed transition from external to internal navigation when internal navigation had GNSS fix was not working correctly
- Fixed the automatic atmospheric calibration was using an unninitialized height value when the firs height measurement was from a GNSS source with an estimated variance greater than the estimated height variance.

Now if the first height measurement is from a GNSS source that height is used as the initial reference
- Fixed an issue where changes in wind speed or direction could unintentionally alter the commanded airspeed, improving speed hold accuracy in windy and variable wind conditions

## 7.8.38

This section presents the changes between the previous firmware version of Veronte Autopilot 1x, **v.7.8.32**, and this firmware version, **v.7.8.38**. For further details, please consult the Service Bulletin nº00025.

### Changed

- Now the IAS guidance does not rely on the estimated wind speed direction
- The IDs of the landing route sections have been changed

### Fixed

- Fixed migration when using a big amout of custom messages

## 7.8.50

This section presents the changes between the previous firmware version of Veronte Autopilot 1x, **v.7.8.38**, and this firmware version, **v.7.8.50**. For further details, please consult the Service Bulletin nº00027.

### Added

- New simulated sensor option
- Bit variables 850 and 851 to write/read GPIOs A and B respectively
- The possibility of simulate GPS time
- I/O byte producers for U-blox messages coming directly from U-blox modules
- Block for reading runway data, block for setting custom waypoint position and velocity, block for custom waypoing guidance
- Blocks to cast from uint16 to bool and vice versa
- Blocks to convert a bitmask to a uint16 and vice versa

- Blocks to convert from yaw, pitch and roll rates into body angular rates (P, Q, R) and vice versa
- Option to force the track reset when the active guidance block changes to one of the same type (example from cruise to cruise). This reset was already being done when changing between different guidances (example from landing to cruise)
- Added desired state reset in swich between guidance blocks

**Changed**

- ESTG documentation
- Fixed an issue where the UAV could stop before reaching a waypoint when the waypoint reach speed (vpoint) was set to zero. The braking profile now correctly targets zero speed at the waypoint position when vpoint is zero rather than prematurely commanding a stop. This ensures the UAV always crosses the waypoint threshold regardless of the configured reach speed

**Fixed**

- Number of deleted logs retrieved in response message
- Logs in a same session saved in different files management
- Tunnel was generating corrupted messages
- Fix deserialization order of Latitude and Longitude to follow the ADSB_VEHICLE message defined by Mavlink
- Remove ID configuration from updater
- Communications encryption
- Sensor error should be able to be recovered when all sensors are ok after one failed
- Filter configuration in external sensor was reporting a false PDI error
- Guidance flyto mode when the guidance block is overwritting the route flyby configuration
- Computation of co-yaw, co-pitch, and co-roll in HIL navigation mode
- Goto speed now uses the speed of the destination waypoint if the active guidance block is not overwritting the speed
- Fixed an issue where the landing path could display a sharp corner instead of a smooth curve when the loiter position was configured as a waypoint (not as center). The path now correctly computes a tangent transition

between the loiter orbit and the runway approach in all geometric configurations
- STANAG Message 2019: Slave to Location. Fixed offset bearing and altitude calculation. The UA now correctly positions itself at the commanded bearing and distance from the reference location when in Slave to Location flight mode, instead of always offsetting to the north regardless of the configured direction
- In the Custom Guidance Waypoint Set block the set speed shifted (the selected speed was being set in the next waypoint instead of the current one)

## 7.8.66

This section presents the changes between the previous firmware version of Veronte Autopilot 1x, **v.7.8.50**, and this firmware version, **v.7.8.66**. For further details, please consult the Service Bulletin nº00030.

### Added

- OAT (Outside Air Temperature) sensor block to compute the real outside density and use it in TAS (True Airspeed) calculation
- Anti-windup in the control allocation block when using non-square matrices. Control feedback now only reports reduced commands when the servos are genuinely saturated, improving disturbance rejection and tracking
- New system bit with ID 144 and name "EKF position fix not fresh/EKF position fix fresh", which is true when the last position update was accepted (passed the NIS check) less than 2.5 seconds ago

### Changed

- Variable "C2 Memory allocation" moved from ID 139 to 145
- On change of active cruise guidance, when there is no selected "Start route" in the new cruise guidance, the patch the UAV will fly to is the first patch of the route only if the option "Track reset" is selected. Otherwise it will continue flying the patch it was flying before. Before this change, with no enabled "Start route" it always flew to the first patch

**Fixed**

- NMEA position parser
- Log generation when Session uses GPS time
- Ethernet routing feature and SCI-E and SCI-F Tx/Rx Flags
- Ping to Cloud ID
- ADC factor when is a 4x
- Improved cold-start behavior of the navigation filter when GNSS is unavailable: the very first valid non-GNSS altitude measurement (e.g. static pressure) is now used to seed the vertical state instead of being rejected by the NIS consistency check. Previously, accelerometer bias and gravity uncertainty at startup could let the estimated altitude and vertical speed drift far enough that the first altitude fix looked inconsistent and was discarded, leaving the filter unable to recover. With this change, the filter initializes height from that first measurement and resets vertical speed to zero, giving a clean starting point and preventing the unbounded drift on GNSS-denied startup
- Input OAT limited to the range [150, 400]K as a value of 0K produced an invalid air density computation. OAT values outside that range now are discarded and the temperature from the USSA76 atmospheric model is used instead
- The position-fix status now clears when the internal EKF has gone more than 2.5 seconds without accepting a position update
- The waypoint event now triggers only while the active patchset is the Route (cruise) patchset; on any other patchset it stays inactive
- The initial point to fly to when switching to cruise guidance is now always the initial route point when the configured initial point is left blank in the guidance configuration block. Before this change the behaviour was different depending if it was the first time this guidance was flown or successive times

## 7.8.80

This section presents the changes between the previous firmware version of Veronte Autopilot 1x, **v.7.8.66**, and this firmware version, **v.7.8.80**. For further details, please consult the Service Bulletin nº00031.

### Added

- New system variable `Target acquisition period` (ID 299), which contains the expected period of the acquisition (CIO-HI) task. This complements the existing `Acquisition Task Timestep` (ID 2047), which records the actual execution period.

### Changed

- Update DriverLib for CM (ethernet driver)
- AGL measurements now bypass the NIS (Normalized Innovation Squared) check. Because the AGL EKF is an isolated 1-state filter, this allows it to immediately track sudden altitude changes without the risk of cross-state divergence
- Renamed BIT 100 from "Position not fixed/Position Fix" to "Navigation position not fixed/Navigation position fix" to make it more clear that this bit refers to the system level (navigation) position fix instead of the sensor level (GNSS) position fix
- Renamed BIT 130 from "EKF Navigation ERROR/EKF Navigation OK" to "Position fix not ready for EKF/Position fix ready for EKF" to reflect its true meaning, which is that this BIT becomes true when any of the position sensors connected to the internal EKF has position fix

### Fixed

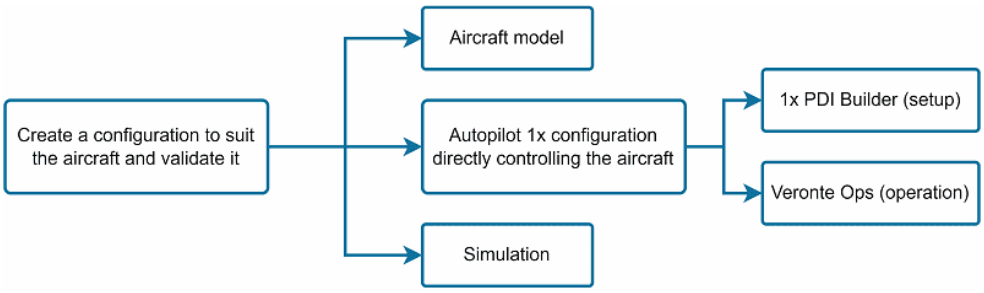
- Desincronization between FDR and firmware when downloading log files
- Improved IMU fault handling for faster failure isolation: To be used in IMU fusion, an IMU must now have both a passing Built-In Test (BIT) and a valid last measurement. Previously, the system relied solely on the comprehensive but slower frequency-based BIT, which could allow stale measurements to be used for several cycles before a failure was flagged. This new real-time check ensures a failing IMU is disabled immediately
- Fixed AGL initialization: The filter now waits for a valid geoid altitude at the UAV's current position before initializing with the first measurement. This ensures accurate conversions between the WGS84, MSL, and AGL reference frames

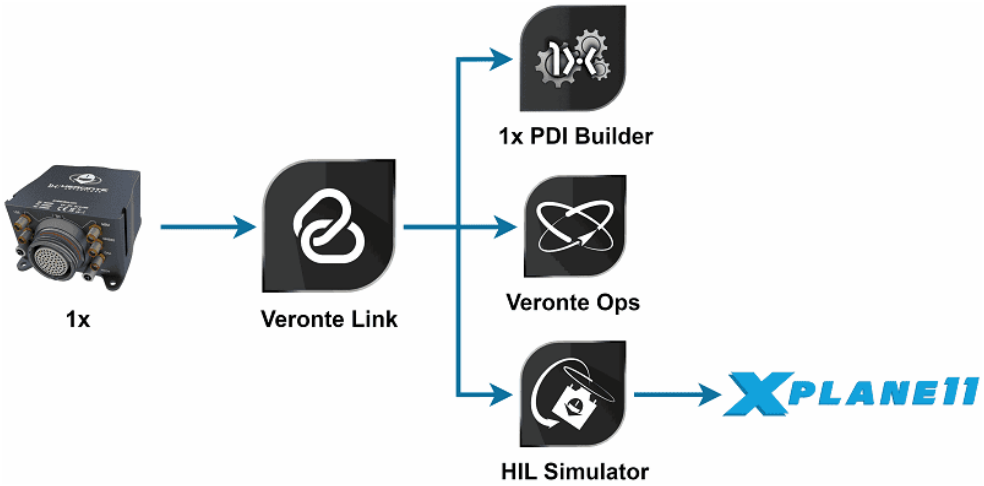
- Fixed problem that caused the waypoint and mark achieved events not to be triggered when they were set in the last waypoint of the route
- Fixed generation of STANAG message 4001 from automations
- Fixed loiter command when it had a limited duration (number of laps or time)
- Transitions between spherical and cartesian acceleration limits in the envelope in the presence of noisy measured IAS could lead to asymmetric speed bounding, which over time could accumulate into high commanded desired velocities. The fix makes envelope switching more robust preventing any cumulative effects on the desired velocity
- Fixed an issue where rejoining a mission via a "Go To" command after a "Fly to Hover" or "Fly to Loiter" action would not respect the target waypoint's speed. The system now forces a guidance velocity reset to ensure the originally configured waypoint speed is recovered

# Step by step

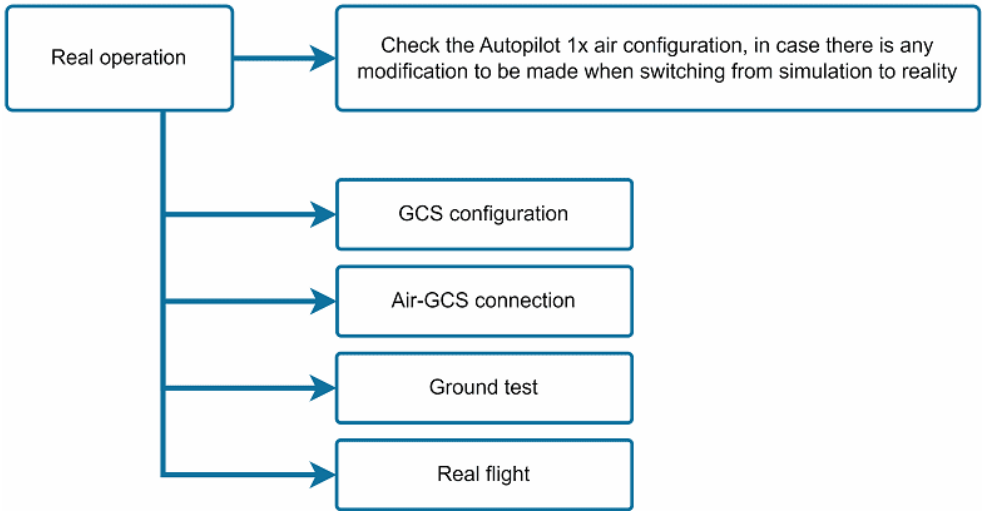
This section explains the main steps to be followed to create a new **PDI configuration** that adapts the Veronte Autopilot 1x performance to the aircraft and mission requirements. The process followed from the creation of a configuration until it is operated in flight with Veronte Autopilot 1x is divided into 2 phases:

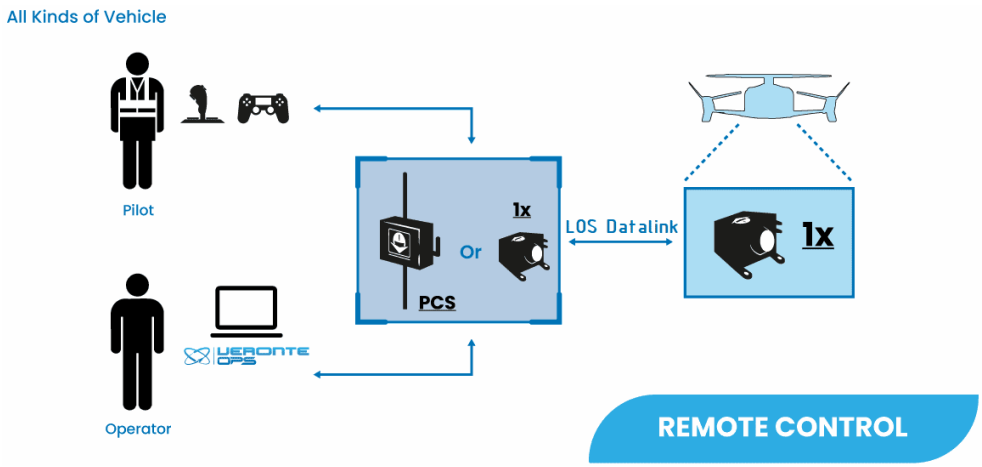
1. **Create a configuration** that suits the expected performance of the aircraft and validate it.



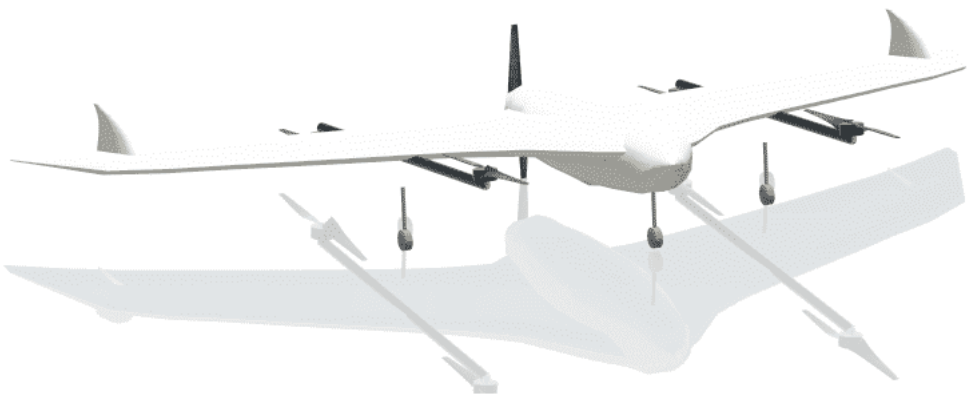




2. **Real operation**




# VTOL



Follow the steps described in each one of the following sections to be able to:

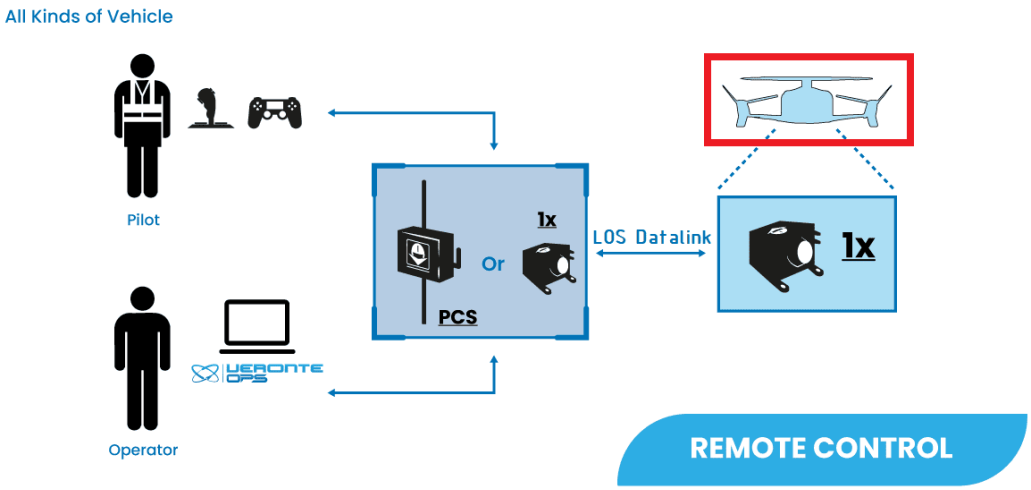
- Adapt a basic Veronte Autopilot 1x configuration to a VTOL aircraft ⇒ 1x Air configuration.

- Perform HIL simulations to validate the aircraft model and configuration ⇒ Simulation.
- Create the Ground Control Station (GCS) configuration that will be part of the standard operation layout ⇒ 1x GCS configuration.
- Establish the communication/connection between both parts of the standard operation layout, the GCS and Autopilot 1x in the VTOL ⇒ Air-GCS connection.
- Ground test to check that all system and electronics work and behave correctly before flight.
- Real operation.

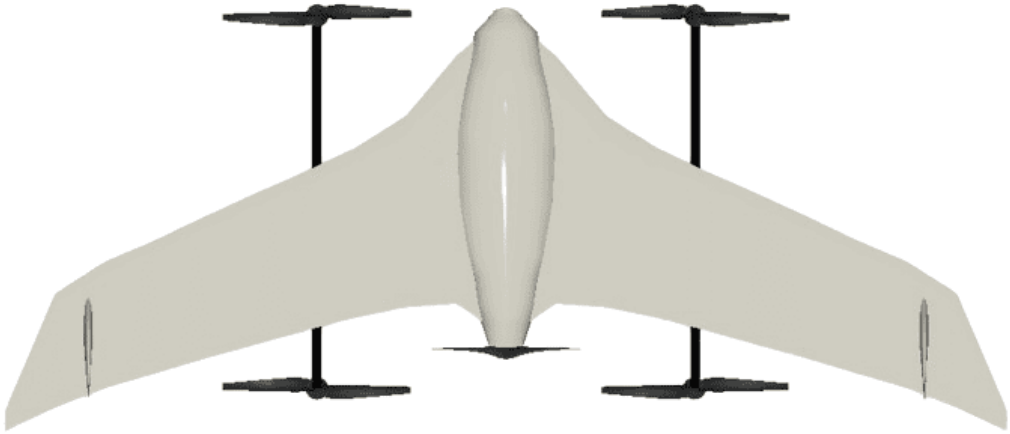
However, first it is necessary to define some aspects of the aircraft model to be taken into account when adjusting the configuration to the expected performance of the aircraft.

## Aircraft specifications

This section defines the main geometrical and operational specifications of the Embention VTOL aircraft.

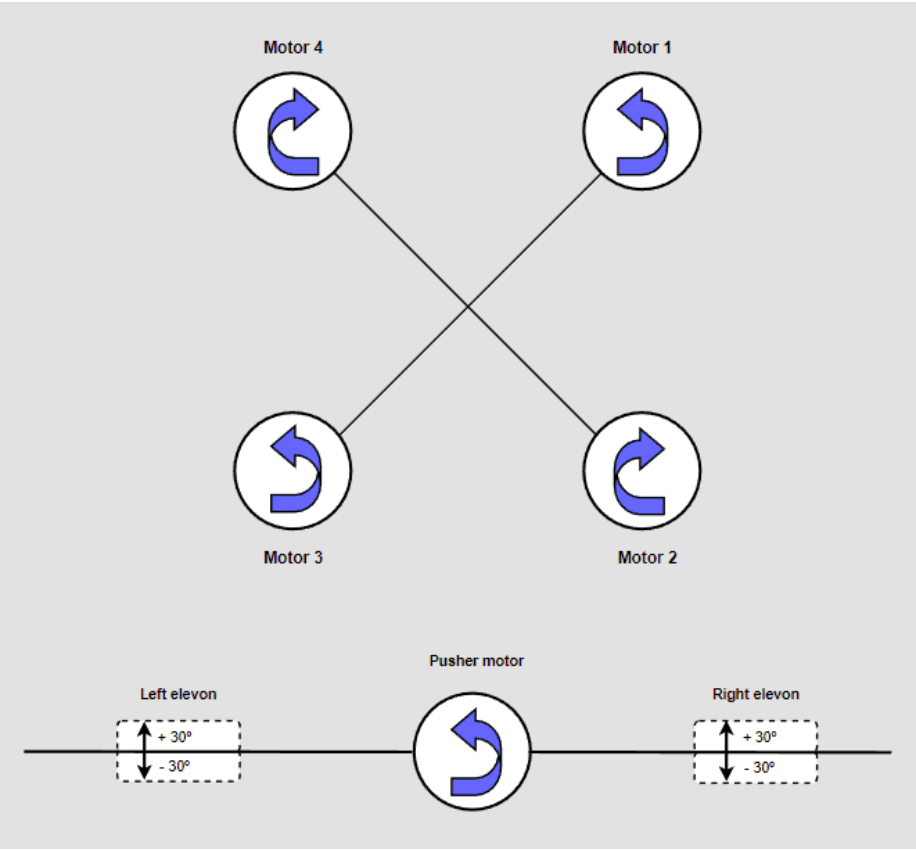



**Embention VTOL Aircraft**



- **Name**: Embention VTOL aircraft
- **Aircraft type**: VTOL
  - The VTOL is composed of **4 vertical propellers** for quadcopter configuration and **1 pusher motor** for Fixed-Wing (FW) configuration.
  - The frame is a **flying wing** composed of **two elevators and two vertical fins without rudder**.

**Explanation**




**Rotation**

### Geometric Specifications

| Parameter | Value |
|---|---|
| Weight | 4.2 Kg |
| Longitudinal CG location | 0.38 m |
| Vertical CG location | 0.0 m |
| Max. Deflection Right Elevator | ±30º |
| Max. Deflection Left Elevator | ±30º |

### Airfoil

| Component | Specification |
|---|---|
| Wing | MH60 |
| Vertical fins | NACA 0009 |

### Engine Specifications

| Parameter | Value |
|---|---|
| Number of propellers | 5 |
| Rotation propeller 1 | CCW |
| Rotation propeller 2 | CW |
| Rotation propeller 3 | CCW |
| Rotation propeller 4 | CW |
| Rotation propeller 5 | CW |
| Propeller 1-4 (vertical) diameter | 15x5.5 |

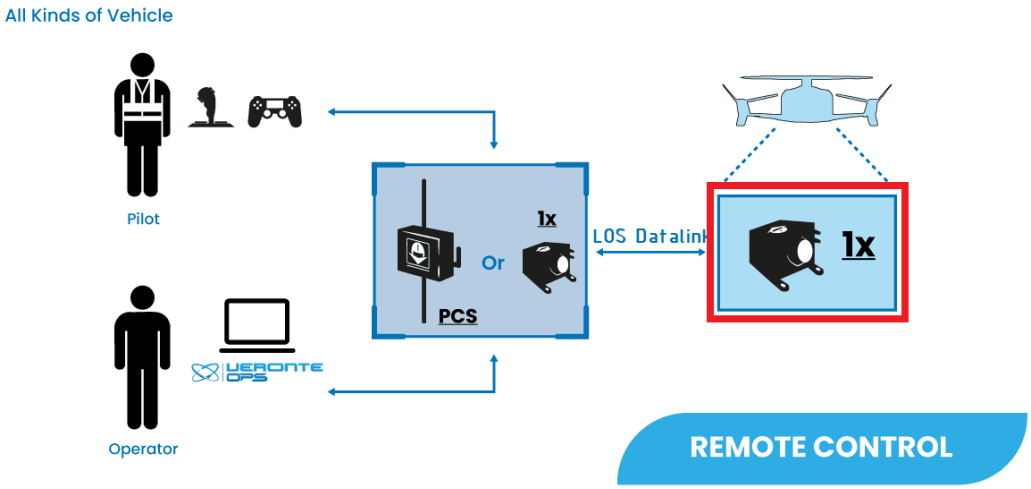
| Parameter | Value |
|---|---|
| Propeller 5 (pusher) diameter | 12x6e |

**Operational Specifications**

| Parameter | Value |
|---|---|
| Stall speed (no flaps) | 15 m/s |
| Cruise speed | 25 m/s |
| Maximum speed | 30 m/s |

## 1x Air configuration

This section defines the Autopilot 1x configuration that will control the aircraft, i.e., that will be inside the VTOL.




This is divided into 2 parts:

- 1x PDI Builder configuration: The creation of the 1x configuration in the **1x PDI Builder** app.
- Operation: The definition of the operation and mission of the aircraft in **Veronte Ops**.

1x PDI Builder configuration

**1x PDI Builder**

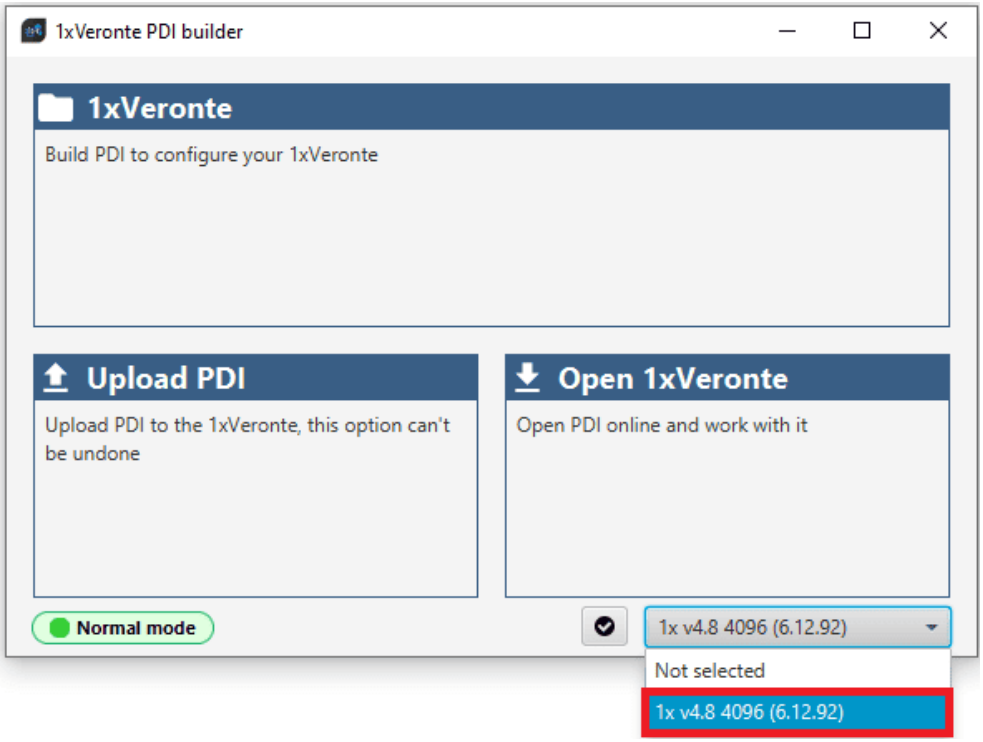
1x PDI Builder is the main configuration tool to adapt a Veronte Autopilot 1x to a specific vehicle, including user-defined communication protocols. 1x PDI Builder allows the user to change the configuration inside the autopilot, create new configurations or work on a configuration offline.

> **Note**
>
> For more information about the installation and use of 1x PDI Builder, visit the 1x PDI Builder user manual.

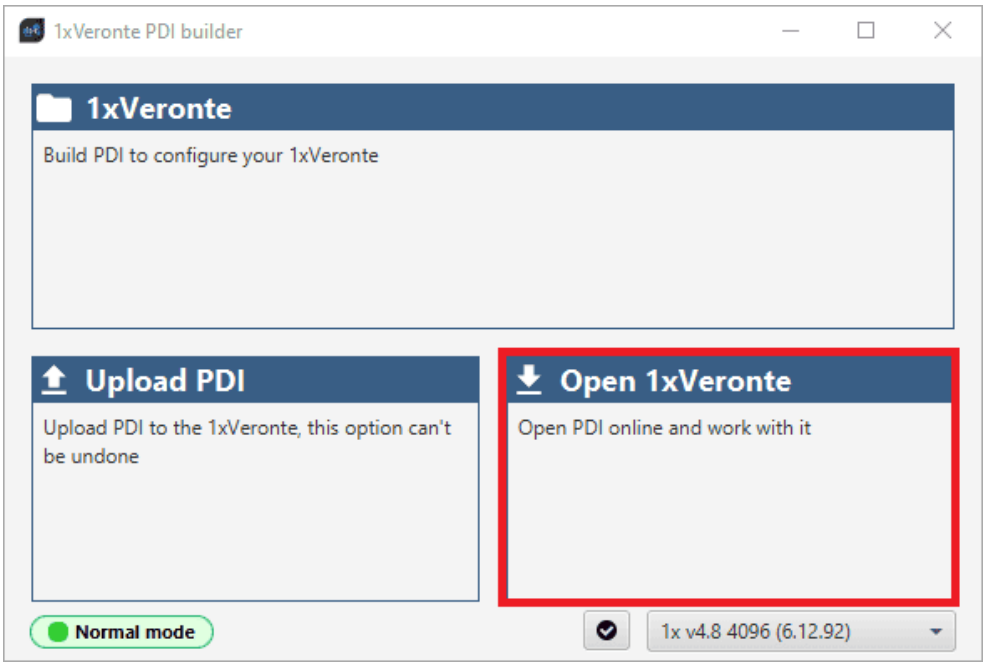
Once the application is installed, the user has two options to configure the autopilot. On the one hand, the user can download one of the templates provided by Embention that best suits his platform, on the other hand, the user can configure the Autopilot 1x from scratch.

> **Tip**
>
> It is highly recommended to follow the first option, which is explained throughout this step-by-step.

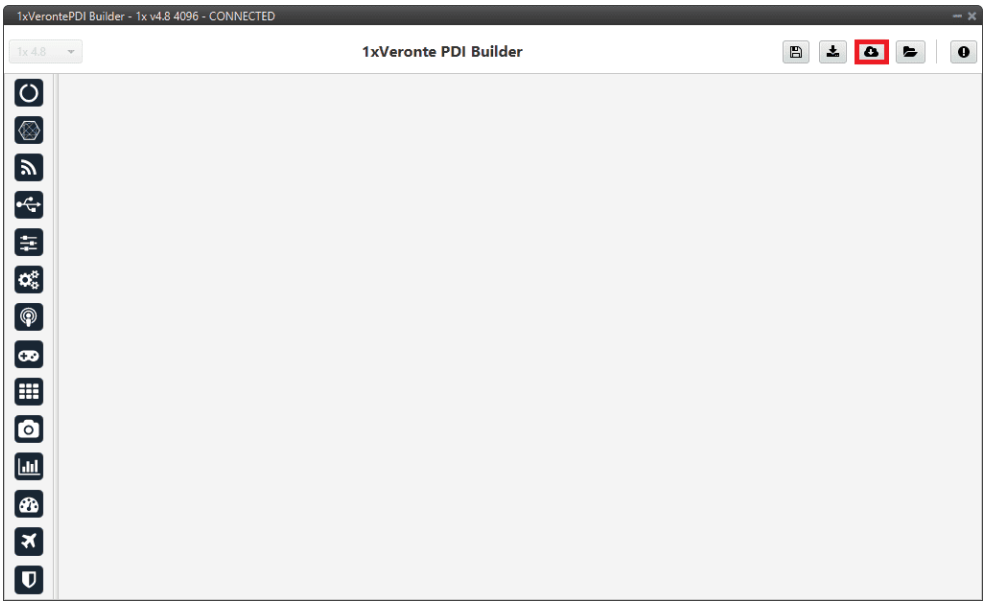
1. Open the 1x PDI Builder application and select the connected Autopilot 1x:



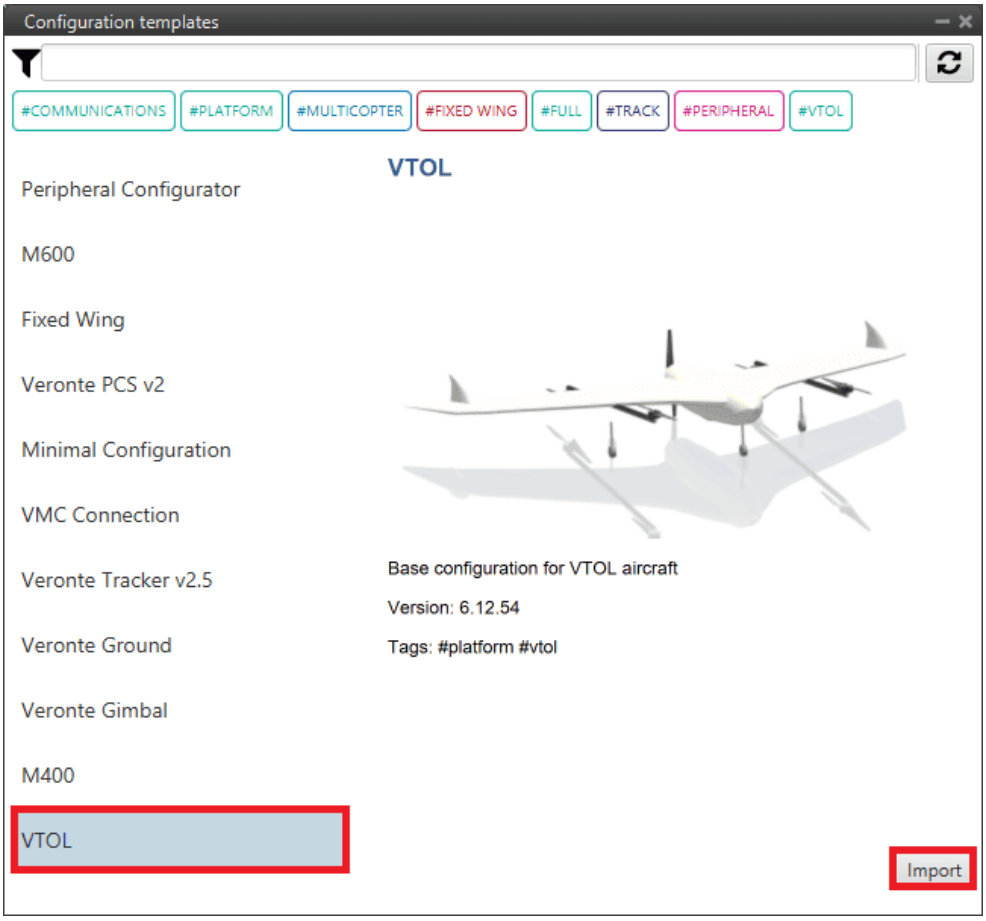

2. Select the **Open 1xVeronte** option to directly edit the configuration loaded in the 1x:



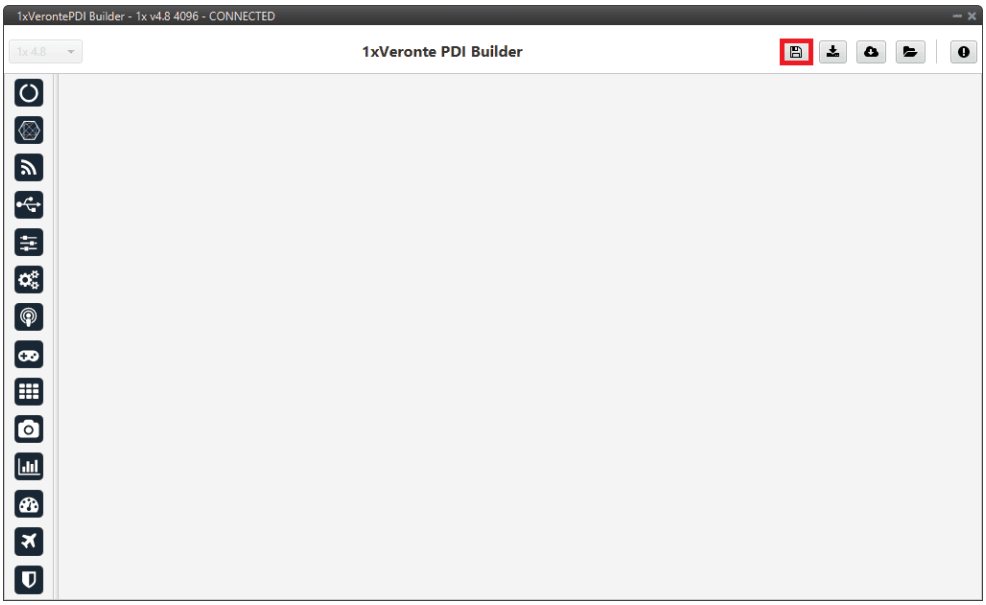

3. Click the **Import PDI from repo** button in the menu bar:




4. Select the **VTOL** template and import it:




5. Finally, click the **Save and close** button to save the downloaded template to the connected Autopilot 1x:




> **Note**
>
> In order to save the configuration, Veronte Autopilot 1x must enter in maintenance mode. Then, after saving any changes, Autopilot 1x will RESET and 1x PDI Builder software will consequently close. For more information, visit the Configuration section of the **1x PDI Builder** user manual.

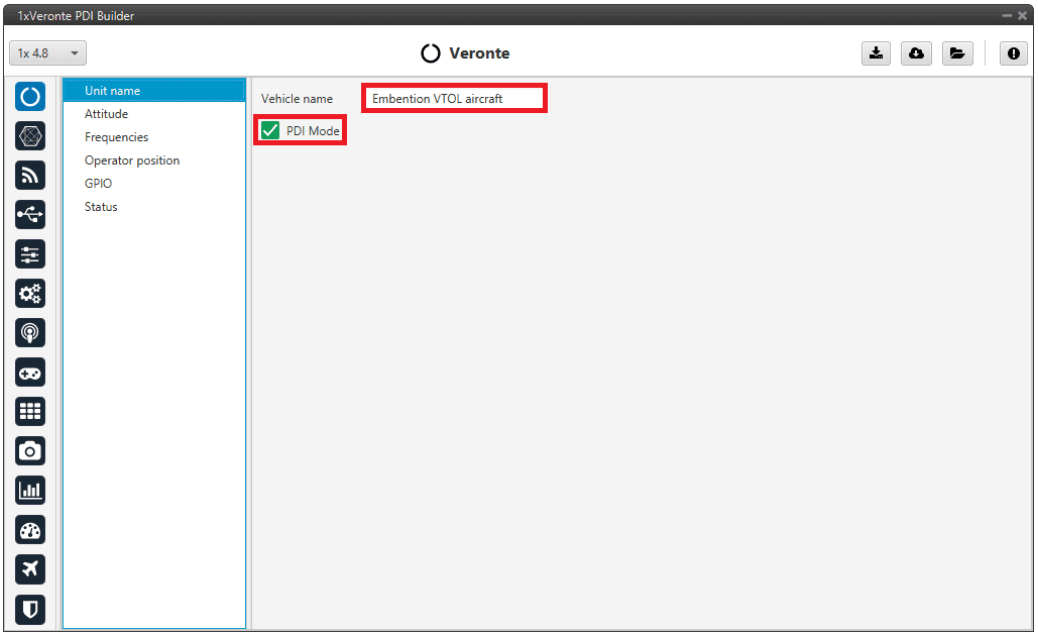
For a basic configuration of the Autopilot 1x, the following parameters must be configured:

- Name & attitude
- Actuators & SU Matrix
- Modes & Phases
- Block Programs
- Automations
- Sensors
- HIL configuration
- Checklist

## Name & Attitude

### Unit name

To start a configuration, it is necessary to define a name for the configuration and it is recommended to be in **PDI mode** while performing the configuration and simulations.



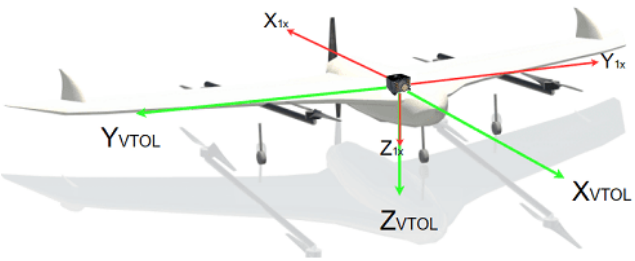

**Unit name & PDI mode**

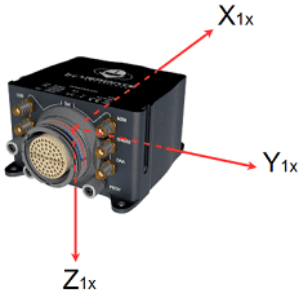
> **Danger**
>
> PDI mode is intended for development purposes since it allows flight phase changes with system, sensor and PDI errors.
>
> It is highly recommended to limit its use to simulation and ground testing of peripherals during the development phase.
>
> Therefore, as it is not advisable to operate in PDI mode, please disable it once the configuration is finished and intended to be used in flight.

For more information, visit the Unit name - Veronte section of the **1x PDI Builder** user manual.

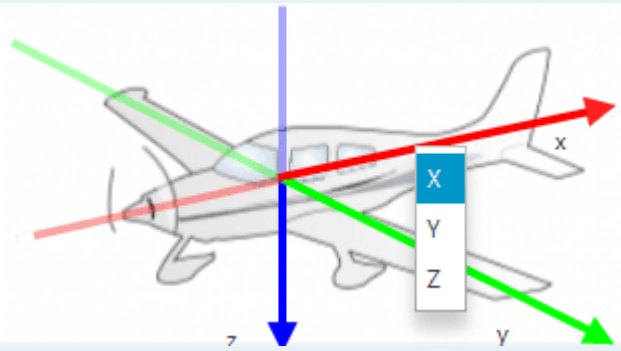
Attitude

The autopilot orientation corresponds to a rotation of 180º around the Z-axis on the aircraft axes.

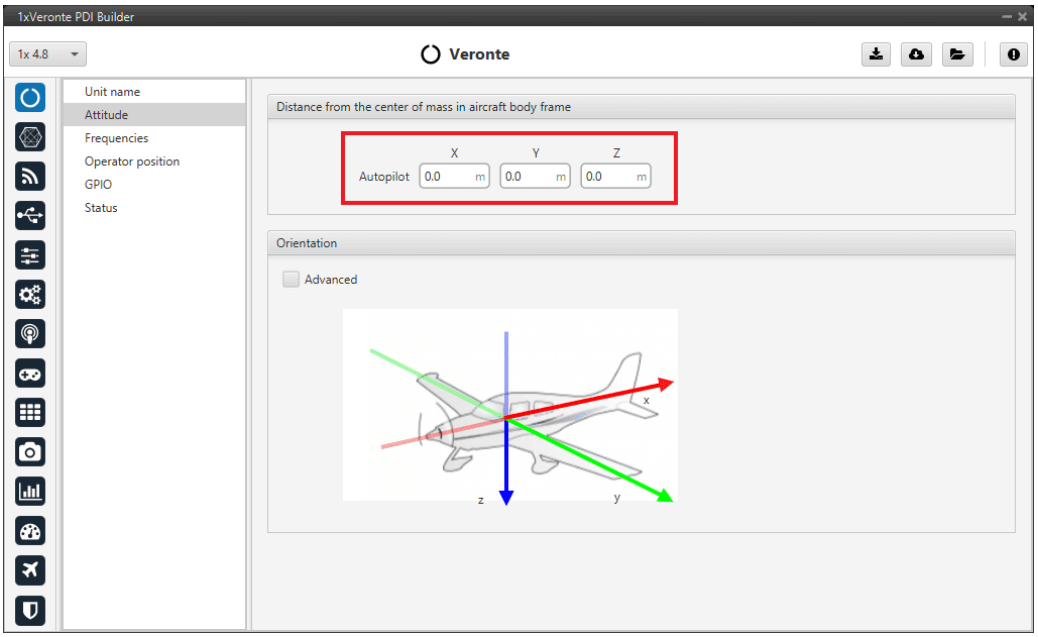



**Orientation**

> **Note**
>
> Since it is a simple orientation to define, we can do it directly from the aircraft schematic.
>
> 
> 
>
> **Change orientation**

Regarding the center of gravity, Autopilot 1x is located at the center of gravity, so the distance to it is zero.




**Embention VTOL aircraft Attitude**

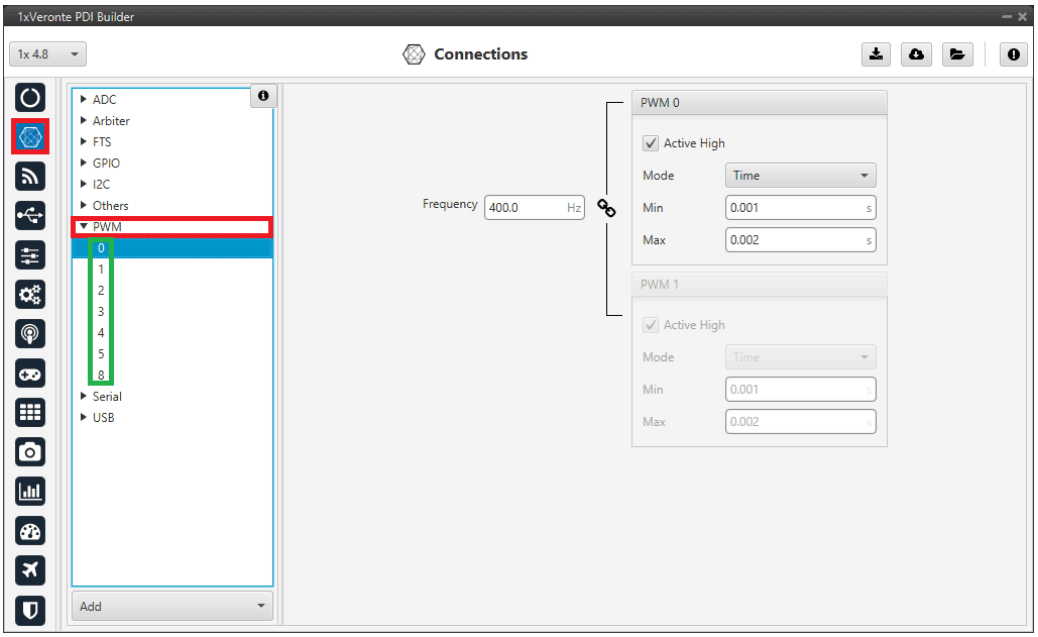
For more information, visit the Attitude - Veronte section of the **1x PDI Builder** user manual.

## Actuators & SU Matrix

### GPIO to PWM connection

Servos and ESC use PWM signals for control. Therefore, since all PWM/GPIO pins are configured as GPIO output by default, it is necessary to change them to PWM output (as many pins as servos/actuators).

> **Note**
>
> For the Embention VTOL aircraft 7 PWM output pins have been configured. Each of these pins will control the movement of each of the 7 servo/actuators available on the platform.




**PWM connections**

For detailed information on how to set up a **GPIO connection to a PWM connection**, visit the PWM - Connections section of the **1x PDI Builder** user manual.

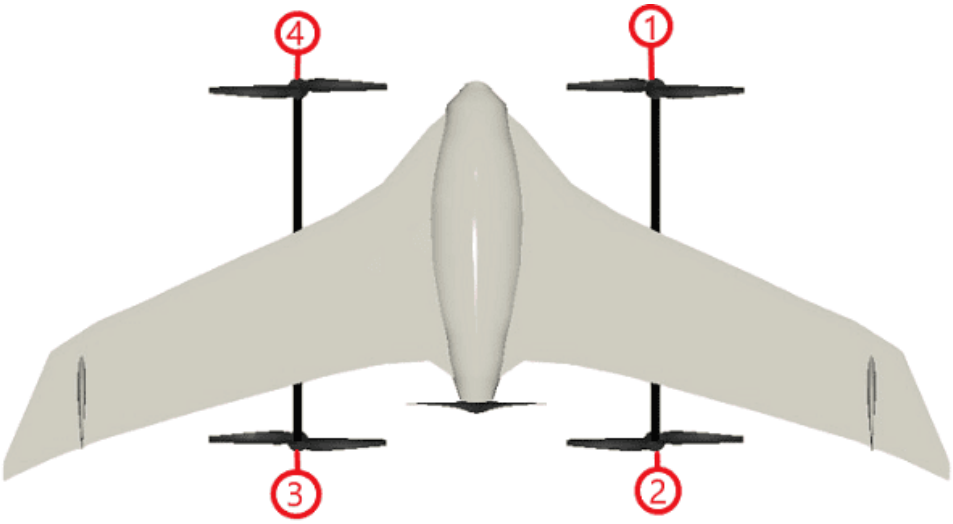
Actuators

The following table shows the correspondence between the actuators of the Embention VTOL aircraft and the servos:

| **Servo** | Servo 0 (S0) | Servo 1 (S1) | Servo 2 (S2) | Servo 3 (S3) | Servo 4 (S4) | Servo 5 (S5) | Servo 6 (S6) |
|---|---|---|---|---|---|---|---|
| **Actuator** | Motor 1 | Motor 2 | Motor 3 | Motor 4 | Pusher motor | Right elevator | Left elevator |

**Motor Numbering**

**In general, the numbering will be defined by the customer**. In this case, the motors are numbered **clockwise**, with number 1 being the motor located to the right of the front of the vehicle.




**Motor numbering**

To define the actuators position, users must first establish a criterion, in the Embention VTOL aircraft it is as follows:

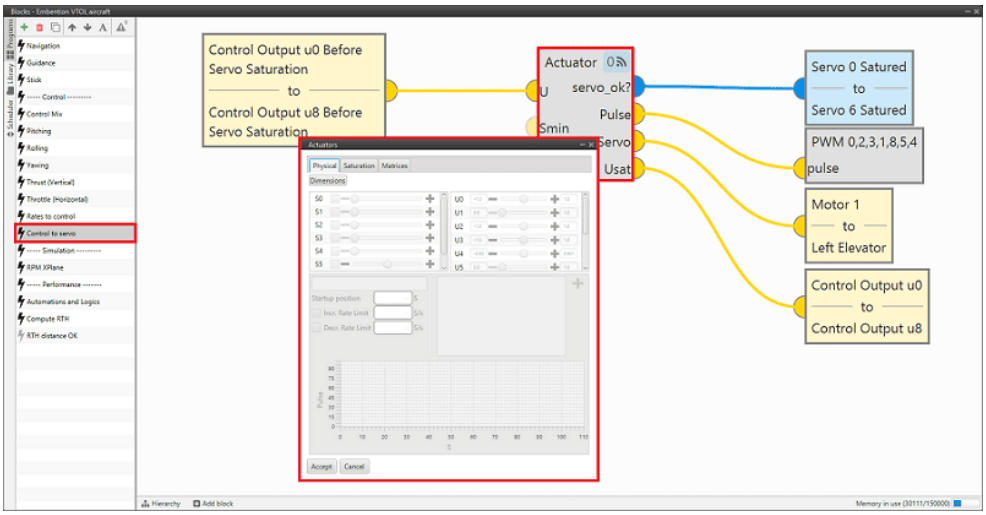
- **Motors**: It is set in the range **0 to 1**.
  - At minimum position (0% RPM) → **0**
  - At maximum position (100% RPM) → **1**
- **Linear actuators**: Defined by angle, from -30º to +30º.

> ⓘ **Note**
> The range of deflection does not have to be symmetrical, although in this case it is.

Next, the position of the actuators must be defined with the Actuator block of the **1x PDI Builder** app. The steps below summarize the configuration to be performed by the user for this block:

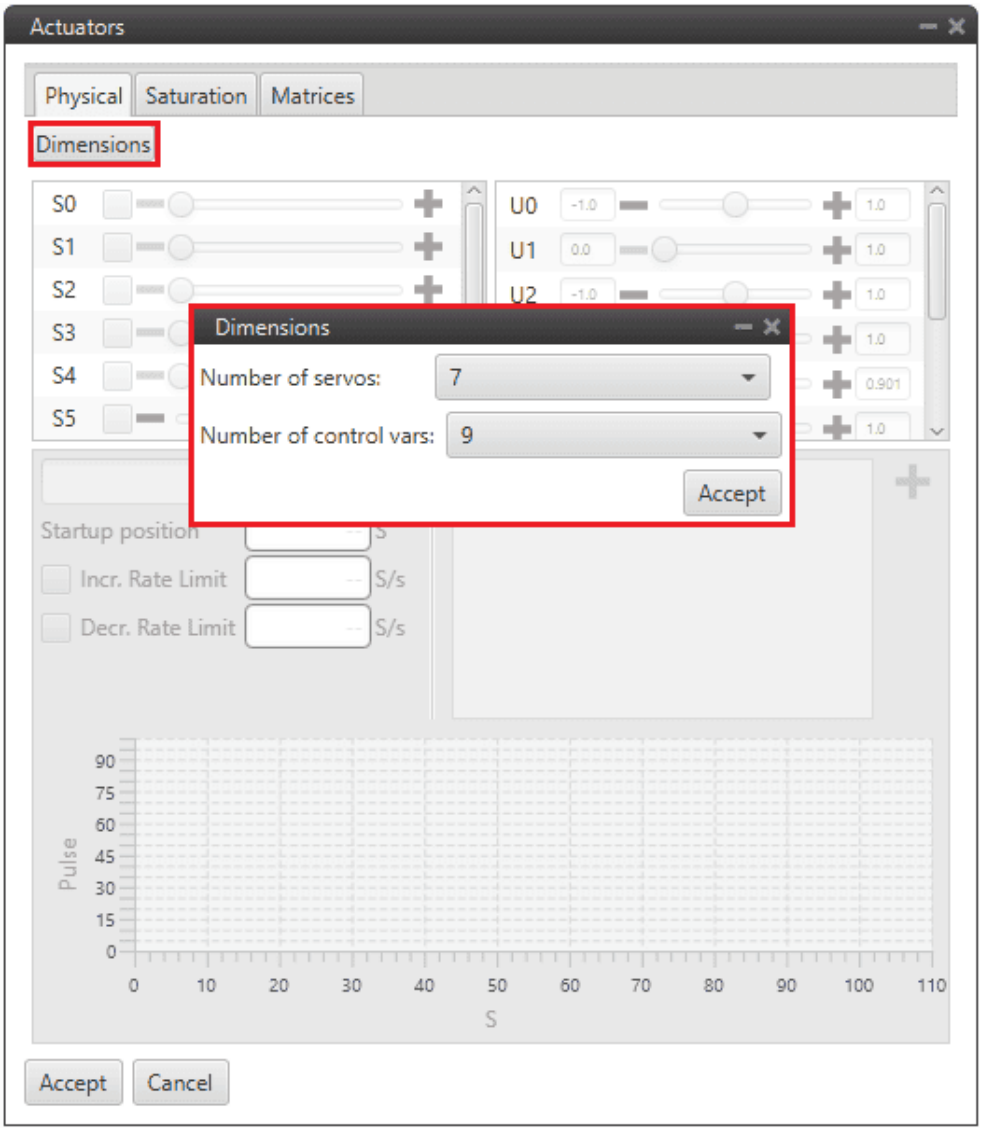
1. Go to **Block Programs menu** → click on "**Launch Editor**".

2. Go to "**Control to servo**" program → double-click on the **Actuator block** to start configuring it.




**Control to servo**

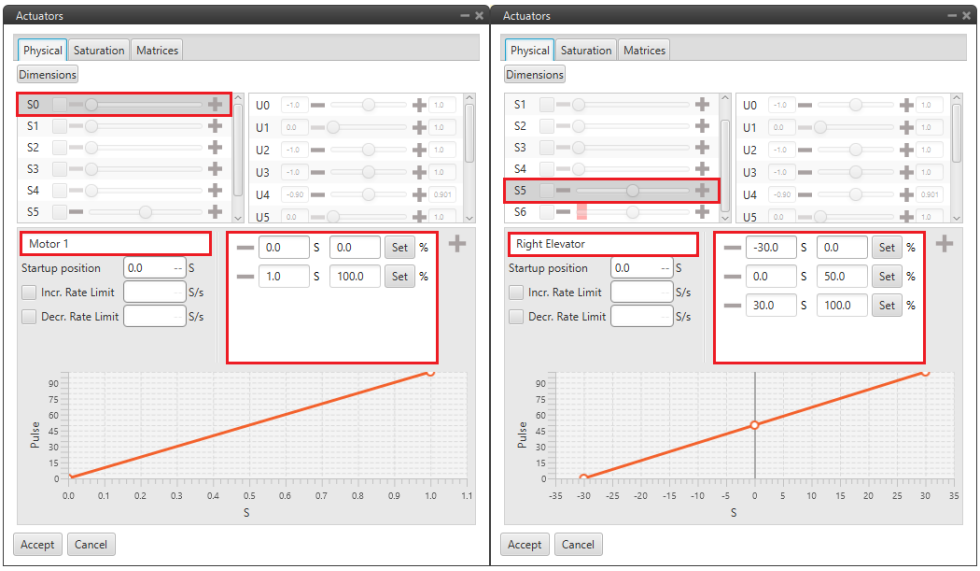
3. Define the dimension of the matrix formed by the servos and the control outputs to be configured.
   For the Embention VTOL aircraft, a 7x9 matrix has to be defined, since there are 7 servos on the platform and 9 control outputs are required.




**Dimensions**

> **Note**
> Once the control outputs have been defined, the configuration of the block programs can begin. These configurations determine the Autopilot's behavior in the automations that allow it to control the aircraft.

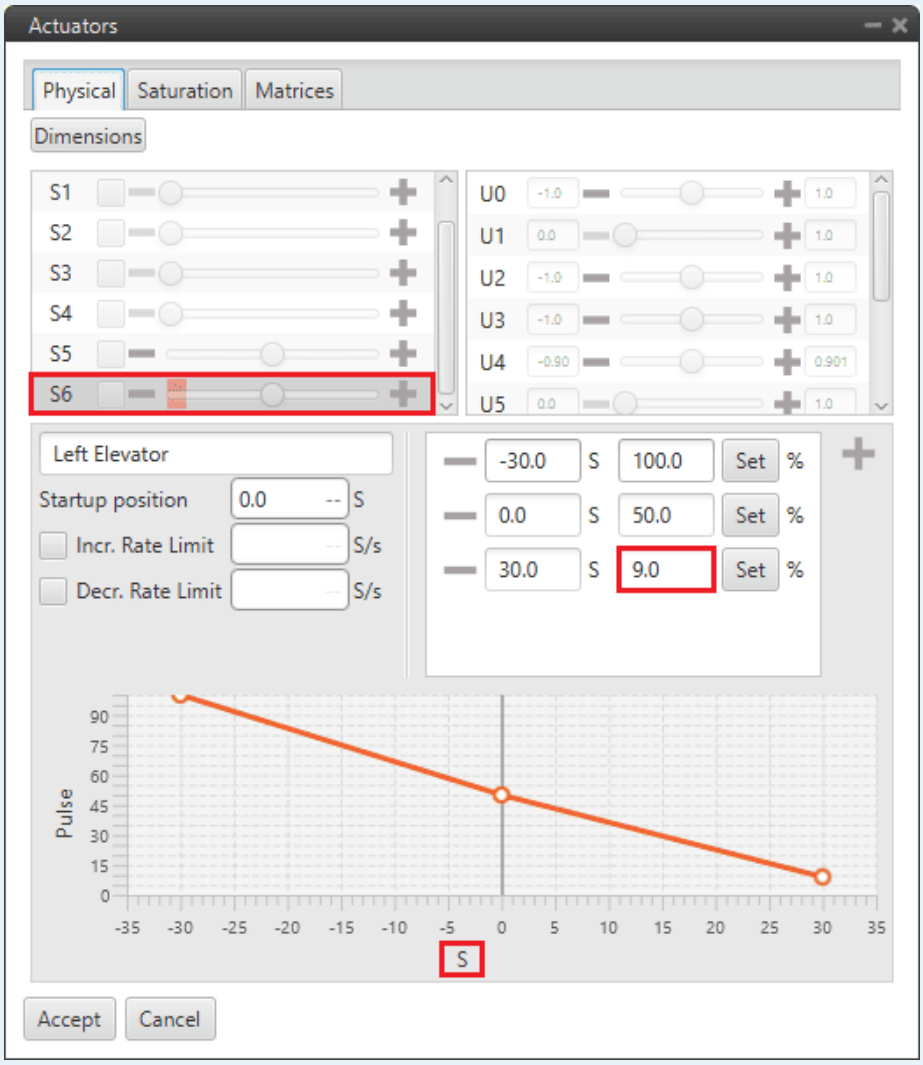
4. For each servo, define the servo position of the servo that responds to the PWM connection pulse, i.e., we try to find the PWM pulse for which the servo position is the desired one.




## Motor 1 / Right Elevator actuator definition

**Note**

The servo position does not have to be symmetrical. In the case of Embention VTOL Aircraft, we have the example of the servo corresponding to the Left Elevator (S6).




**Left Elevator actuator definition**

For this S6, the PWM pulse should allow to achieve a deflection of ± 30º. For the +30º deflection, it is achieved with 9% of the PWM pulse, unlike the Right Elevator (S5) which was obtained with 100%.

**Important**

The variable **S** in the graph is a position that can represent any variable desired by the user (**rad**, **º**, **RPM**).

> **Important**
>
> A complete explanation of the use of the **Actuator block** in the **Control to servo program** can be found at Control to servo - Block Programs section of this manual.

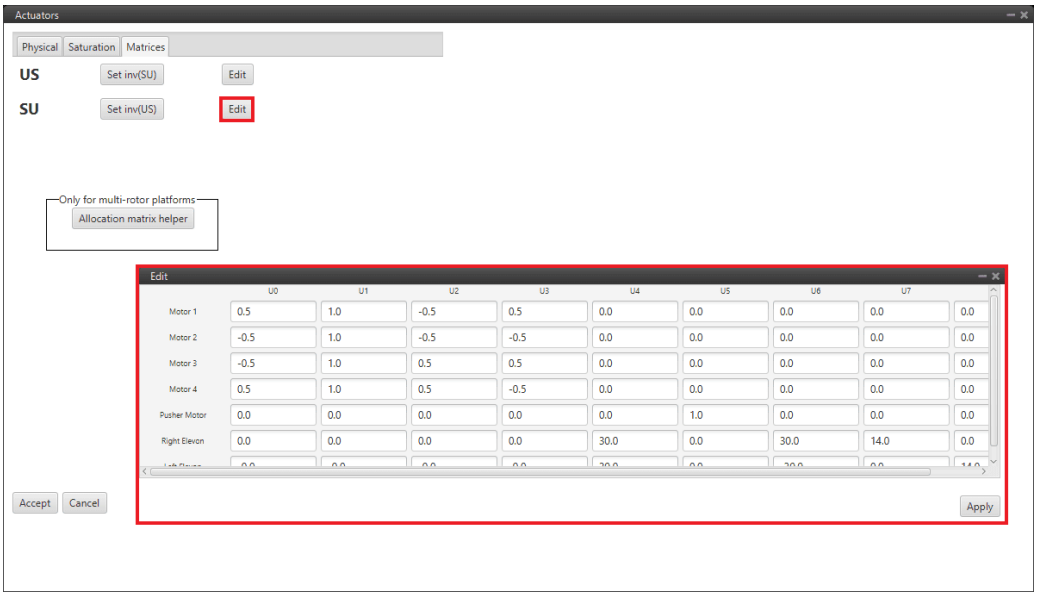
SU Matrix

**SU matrix** contains the relationship between each actuator (S) and its control output (U). Although it is also possible to define the US matrix (inverse of the SU matrix), the SU matrix is usually defined because the visualization of the movement is more intuitive this way.

In the SU matrix of the Embention VTOL aircraft, each of the 9 defined control outputs represents an aircraft control action, as can be seen in the following table.

| **Control output** | u0 | u1 | u2 | u3 | u4 | u5 | u6 | u |
|---|---|---|---|---|---|---|---|---|
| **Action** | Pitch (Quad) | Thrust (Vertical) | Roll (Quad) | Yaw (Quad) | Pitch (Plane) | Throttle (Horizontal) | Roll (Plane) | Adv Ya (Ri |

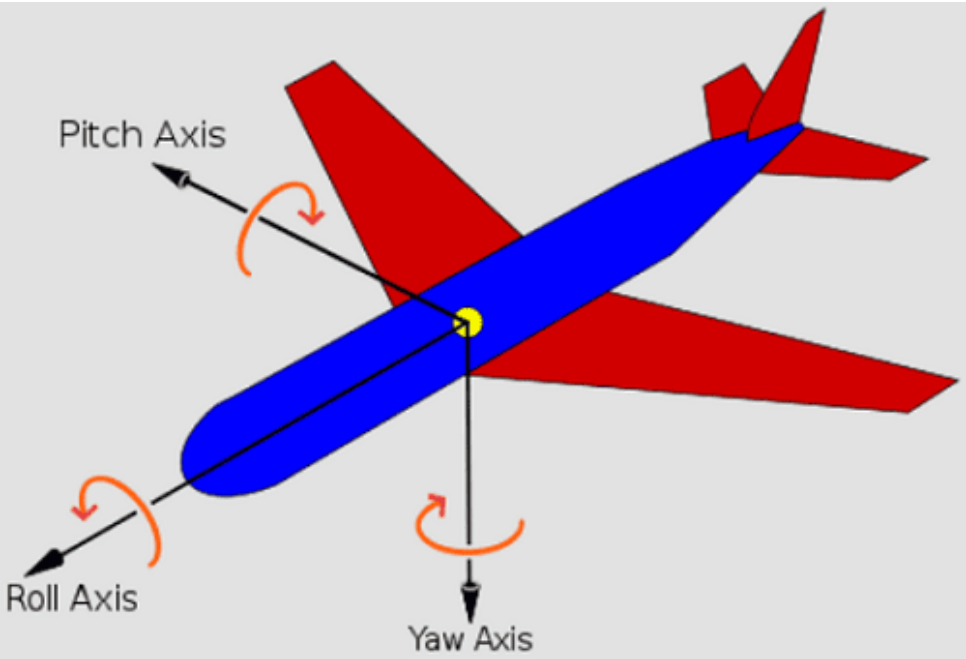
These actions can be performed thanks to the variation of the position of the 7 platform servos already mentioned.




## SU Matrix

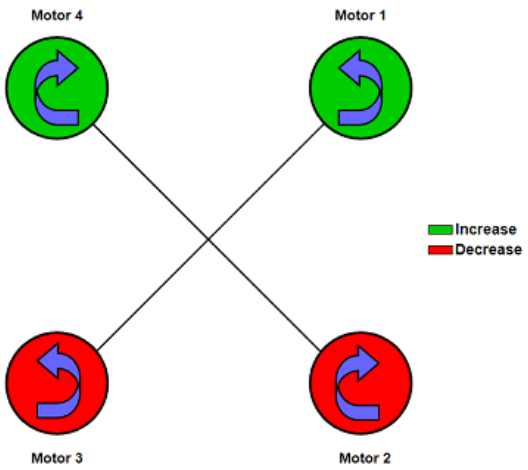
### Sign Convention

- Positive angle means deflection upward or to the right.
- Negative angle means deflection downward or to the left.



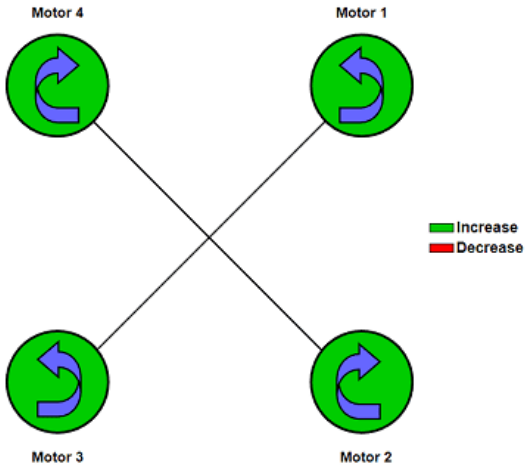

## u0 -- Pitch (Q)

| Servos | u0 |
|---|---|
| Motor 1 | 0.5 |
| Motor 2 | -0.5 |
| Motor 3 | -0.5 |
| Motor 4 | 0.5 |
| Pusher motor | 0.0 |
| Right elevator | 0.0 |
| Left elevator | 0.0 |



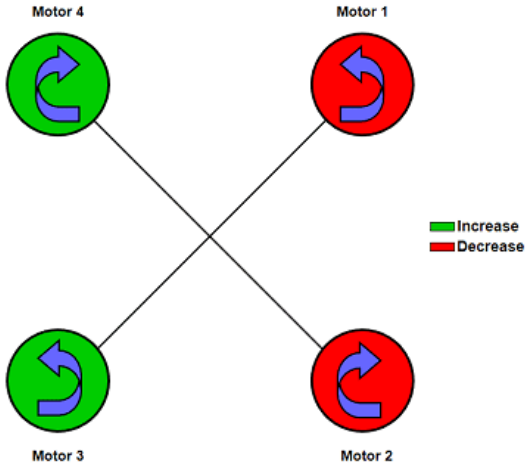

## u1 -- Thrust (vertical)

| Servos | u1 |
|---|---|
| Motor 1 | 0.5 |
| Motor 2 | 0.5 |
| Motor 3 | 0.5 |
| Motor 4 | 0.5 |
| Pusher motor | 0.0 |
| Right elevator | 0.0 |
| Left elevator | 0.0 |



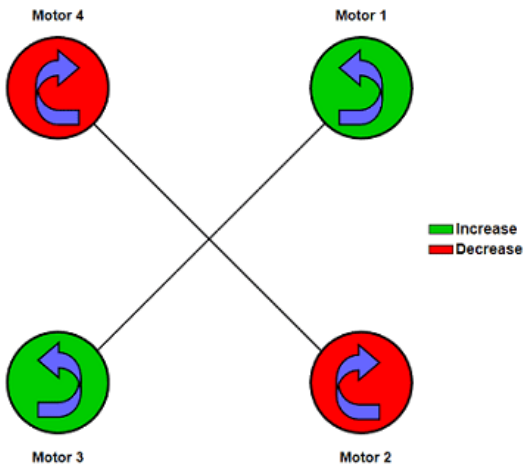

## u2 -- Roll (Q)

| Servos | u2 |
|---|---|
| Motor 1 | -0.5 |
| Motor 2 | -0.5 |
| Motor 3 | 0.5 |
| Motor 4 | 0.5 |
| Pusher motor | 0.0 |
| Right elevator | 0.0 |
| Left elevator | 0.0 |



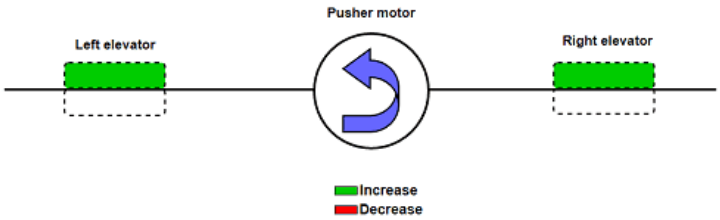

## u3 -- Yaw (Q)

| Servos | u3 |
|---|---|
| Motor 1 | 0.5 |
| Motor 2 | -0.5 |
| Motor 3 | 0.5 |
| Motor 4 | -0.5 |
| Pusher motor | 0.0 |
| Right elevator | 0.0 |
| Left elevator | 0.0 |



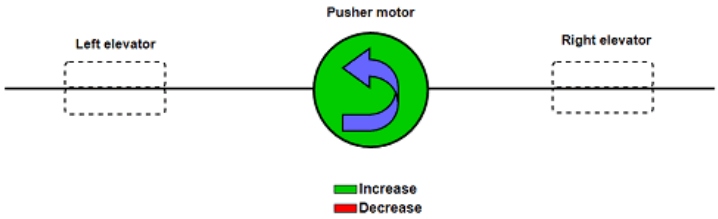

## u4 -- Pitch (P)

| Servos | u4 |
|---|---|
| Motor 1 | 0.0 |
| Motor 2 | 0.0 |
| Motor 3 | 0.0 |
| Motor 4 | 0.0 |
| Pusher motor | 0.0 |
| Right elevator | 30.0 |
| Left elevator | 30.0 |



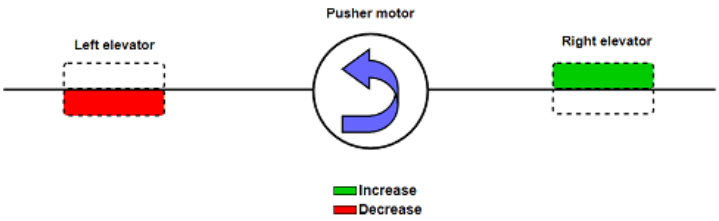

## u5 -- Throttle (horizontal)

| Servos | u5 |
|---|---|
| Motor 1 | 0.0 |
| Motor 2 | 0.0 |
| Motor 3 | 0.0 |
| Motor 4 | 0.0 |
| Pusher motor | 1.0 |
| Right elevator | 0.0 |
| Left elevator | 0.0 |



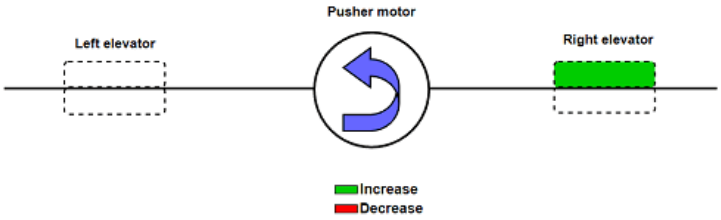

## u6 -- Roll (P)

| Servos | u6 |
|---|---|
| Motor 1 | 0.0 |
| Motor 2 | 0.0 |
| Motor 3 | 0.0 |
| Motor 4 | 0.0 |
| Pusher motor | 0.0 |
| Right elevator | 30.0 |
| Left elevator | -30.0 |



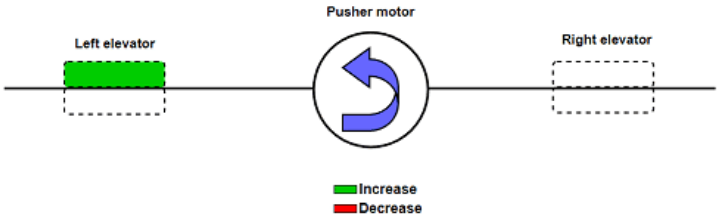

## u7 -- Adverse Yaw (Right)

| Servos | u7 |
|---|---|
| Motor 1 | 0.0 |
| Motor 2 | 0.0 |
| Motor 3 | 0.0 |
| Motor 4 | 0.0 |
| Pusher motor | 0.0 |
| Right elevator | 14.0 |
| Left elevator | 0.0 |




## u8 -- Adverse Yaw (Left)

| Servos | u8 |
|---|---|
| Motor 1 | 0.0 |
| Motor 2 | 0.0 |
| Motor 3 | 0.0 |
| Motor 4 | 0.0 |
| Pusher motor | 0.0 |
| Right elevator | 0.0 |
| Left elevator | 14.0 |




For more information, visit the Actuator - Servos blocks of the **Block Programs** section of the **1x PDI Builder** user manual.

> **Important**
>
> A complete explanation of the use of the **Actuator block** in the **Control to servo program** can be found at Control to servo - Block Programs section of this manual.

## Modes & Phases

### Modes definition

The flight modes determine who is in charge of controlling each one of the aircrafts control channels. Each of the channels represents a control output, defined by the customer in the Block Programs configuration. In the Embention VTOL aircraft these channels are linked in order to each control output of the SU matrix.

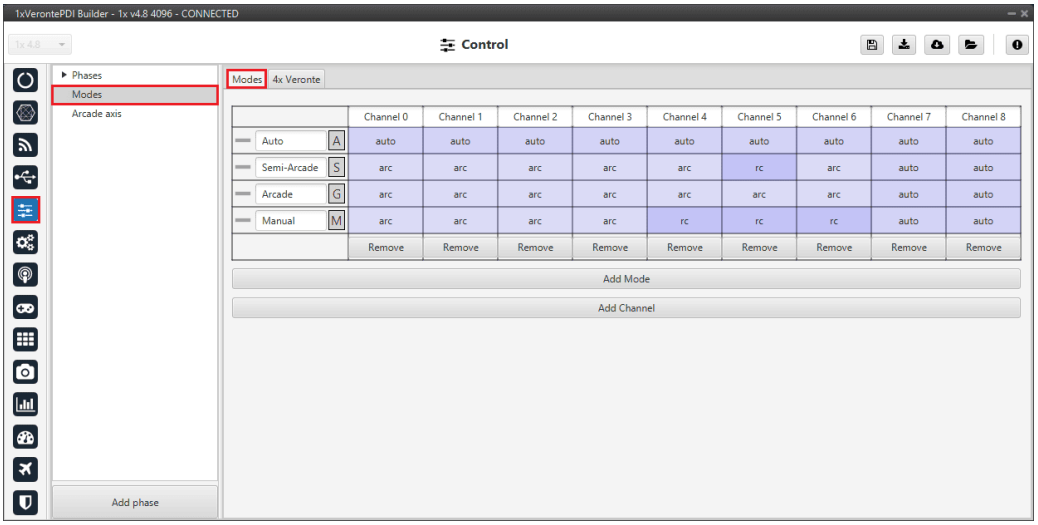
| **Channel** | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|---|
| **Control output** | u0 | u1 | u2 | u3 | u4 | u5 | u6 | u7 | u8 |
| **Action** | Pitch (Q) | Throttle (Q) | Roll (Q) | Yaw (Q) | Pitch (P) | Thrust (P) | Roll (P) | Adverse Yaw Right (P) | Adverse Yaw Left (P) |

There are four different **control modes** and it is possible to combine them to create custom flight modes. The options available are:

- **auto**: Automatic mode.
- **rc**: Radio Control mode, i.e. manual mode.
- **arc**: Arcade mode, this could be considered as a mix between automatic and manual.
- **mix**: In this mode, it is possible to select in which step of the controller the pilot command will enter.

For more information on these control modes, visit the Modes - Control section of the **1x PDI Builder** user manual.

For each of the channels of the 4 **flight modes** defined, a control mode must be assigned to it.




**Modes definition**

> **Note**
>
> The controls to perform the control action Adverse Yaw, correspond to channels 7 and 8 and are performed in **Auto** mode.
>
> This is because these actions are complementary to the Roll movement, so being a correction, they must be performed automatically by the autopilot **for all flight modes**.

- **Auto flight mode** `A`
  In this mode, all control outputs (channels) defining the aircraft performances are automatically controlled by the autopilot (auto mode).
- **Semi-Arcade flight mode** `S`
  In this mode, the Pitch, Thrust, Roll and Yaw controls are controlled in **arc mode**. That is, the input information is sent from the pilot's stick to the autopilot, and it is the autopilot that takes over the performance. The Throttle action is controlled directly by the pilot stick in **rc mode**.
- **Arcade flight mode** `G`

In this flight mode, the aircraft performances are controlled in **arc mode**. So, when the pilot commands a desired pitch, roll, IAS, heading and so on, and it will be the control system that will be in charge of making the platform follow those commands.

- **Manual flight mode** `M`

  The manual flight mode mixes **rc** control mode with **arc**:

  - For the quadcopter control, **arc mode** must be defined since the pilot's stick cannot act on the motors.
  - For the fixed wing control, **rc mode** can be set so that the pilot's stick acts directly on the elevators and throttle.

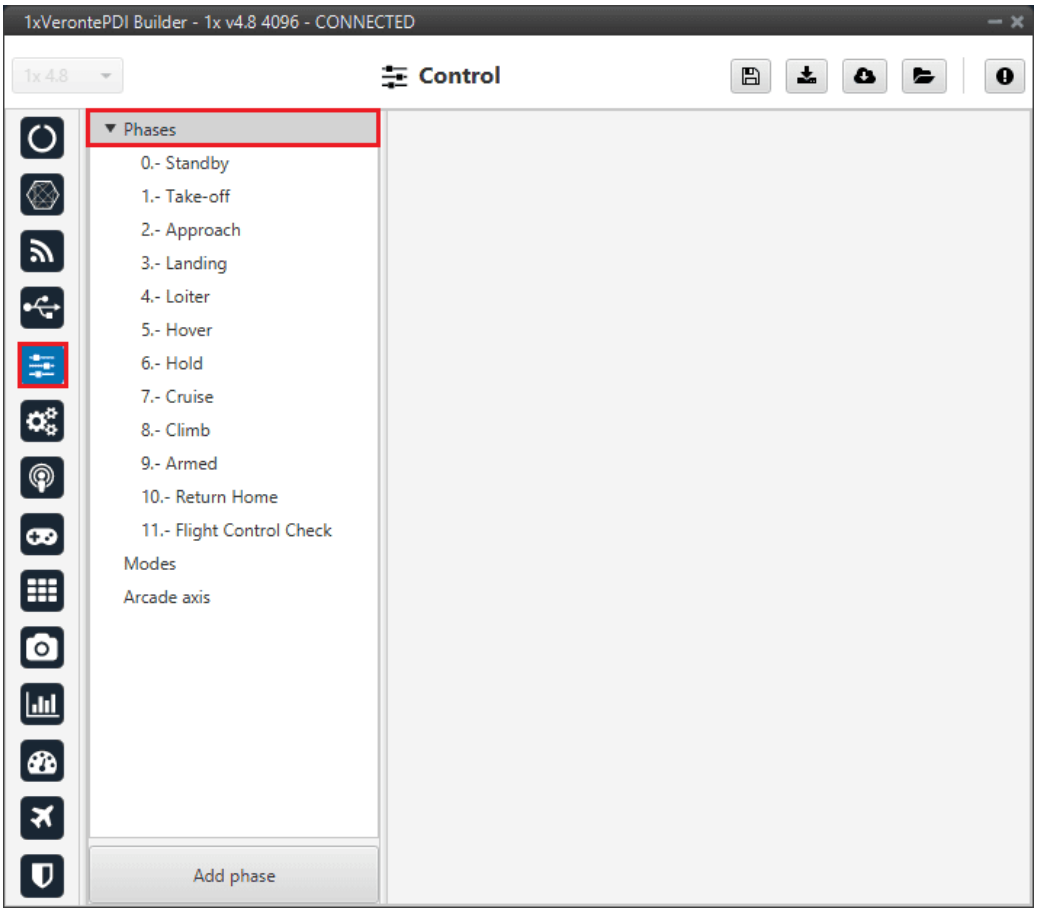
Phases definition

The flight phases that will control the aircraft in the different stages of the operation must be created (defined not configured).

For the operation of the Embention VTOL aircraft 12 flight phases have been defined:

1. **Standby**: All actuators are deactivated.
2. **Take-off**: Aircraft vertical take-off.
3. **Approach**: Execute approach maneuver to the runway defined for VTOL.
4. **Landing**: Aircraft vertical landing.
5. **Loiter**: Loiter at current position and altitude.
6. **Hover**: Maintain current 3D position.
7. **Hold**: Stabilize attitude and vertical speed.
8. **Cruise**: Follow defined mission.
9. **Climb**: Ascend to the loiter from the runway.
10. **Armed**: Vertical propellers run-up.
11. **Return Home**: Go to one of the pre-defined landing points and hover.
12. **Flight Control Check**: Manual check of elevators and horizontal motor.

> ⓘ **Note**
>
> In addition to these phases, when Veronte Autopilot 1x boots up it is by default in the "**Initial**" phase.



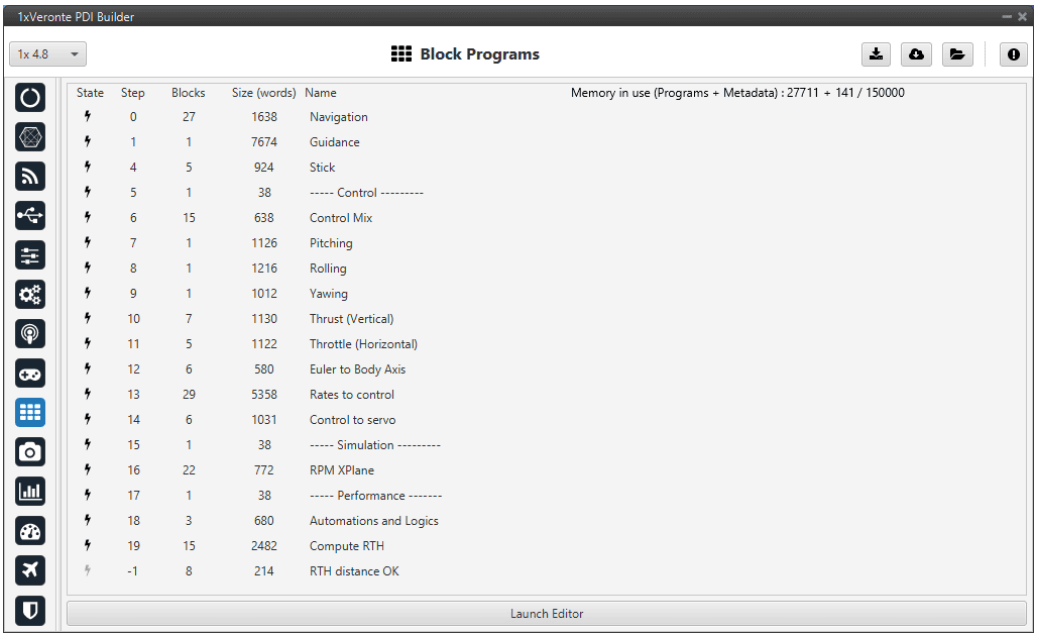

**Phases definition**

> **Note**
>
> The configuration of the flight phases (guidance and control commands) is done in the Block Programs menu.

For more information, refer to the Phases - Control section of the **1x PDI Builder** user manual.

## Block Programs

In this section the user can configure and develop the different programs for aircraft control, guidance, navigation, etc.




| State | Step | Blocks | Size (words) | Name |
|---|---|---|---|---|
| | 0 | 27 | 1638 | Navigation |
| | 1 | 1 | 7674 | Guidance |
| | 4 | 5 | 924 | Stick |
| | 5 | 1 | 38 | ----- Control --------- |
| | 6 | 15 | 638 | Control Mix |
| | 7 | 1 | 1126 | Pitching |
| | 8 | 1 | 1216 | Rolling |
| | 9 | 1 | 1012 | Yawing |
| | 10 | 7 | 1130 | Thrust (Vertical) |
| | 11 | 5 | 1122 | Throttle (Horizontal) |
| | 12 | 6 | 580 | Euler to Body Axis |
| | 13 | 29 | 5358 | Rates to control |
| | 14 | 6 | 1031 | Control to servo |
| | 15 | 1 | 38 | ----- Simulation --------- |
| | 16 | 22 | 772 | RPM XPlane |
| | 17 | 1 | 38 | ----- Performance ------- |
| | 18 | 3 | 680 | Automations and Logics |
| | 19 | 15 | 2482 | Compute RTH |
| | -1 | 8 | 214 | RTH distance OK |

**Block Programs menu**

The following programs have been developed for the configuration of the Embention VTOl aircraft:

> ⓘ **Note**
>
> The program name is defined by the user, so they do not have to match these.

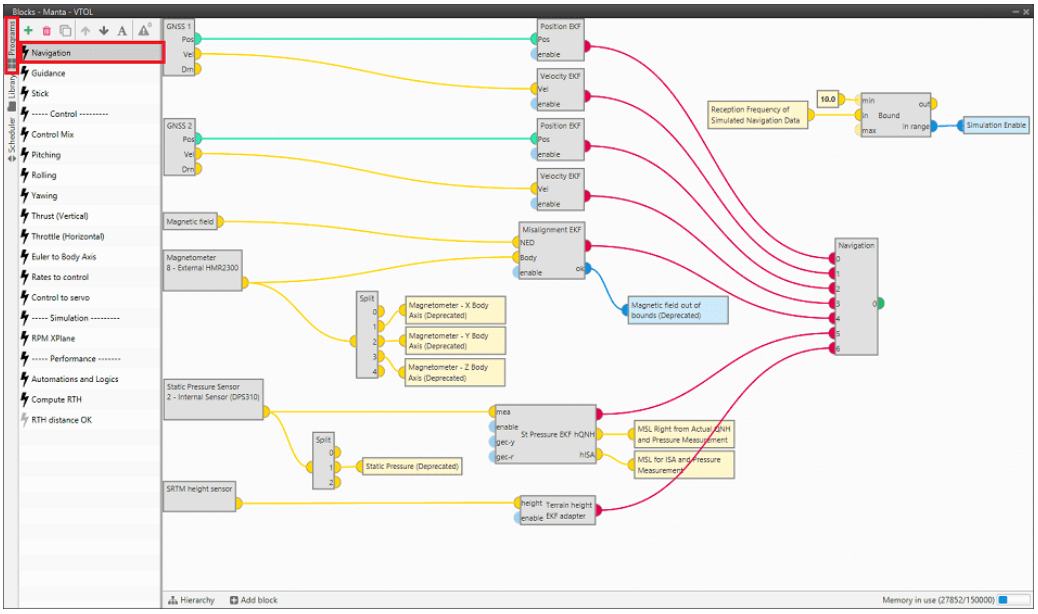
- Navigation: Control laws that allow navigation through the use of both **internal and external sensors**.
- Guidance: Control laws that allow obtaining a precise **guidance** to perform the **desired mission**.
- Stick: Block program to transform pilot stick actions into **input values** to Autopilot 1x.
- Control Mix: Block program that establishes the optimal **flight configuration** as a function of the IAS value.
- Pitching: Control laws for performing perform **Pitch** in both multicopter and fixed-wing (FW) flight configurations.
- Rolling: Control laws for performing **Roll** in both multicopter and fixed-wing (FW) flight configurations.
- Yawing: Control laws for performing **Yaw** in both multicopter and fixed-wing (FW) flight configurations.

- Thrust (Vertical): Control laws for performing **Thrust** in multicopter flight configurations.
- Throttle (Horizontal): Control laws for performing **Thrust** in fixed-wing (FW) flight configurations.
- Euler to Body Axis: Program that transforms a reference system from **Euler Axis** to **Body Axis**.
- Rates to control: Control laws that transform the desired guidance variables into **output controls**.
- Control to servo: Control laws that transform output controls into **servo actions**.
- RPM XPlane: Program that allows to transform the aircraft real performance in a **simulated environment**.
- Compute RTH: Program to allow the aircraft to choose from several points which is the best one to be used as **RTH point** for the RTH phase.
- RTH distance OK: Program to check if a particular point is suitable to start the **RTH phase**.

For more information, visit the Block Programs section of the **1x PDI Builder** user manual.

Navigation

The following program is defined to configure all sensors that will be used to determine the navigation of the Autopilot 1x.




**Navigation - Program**

Follow the steps below for a basic configuration of the sensors that enable 1x navigation.

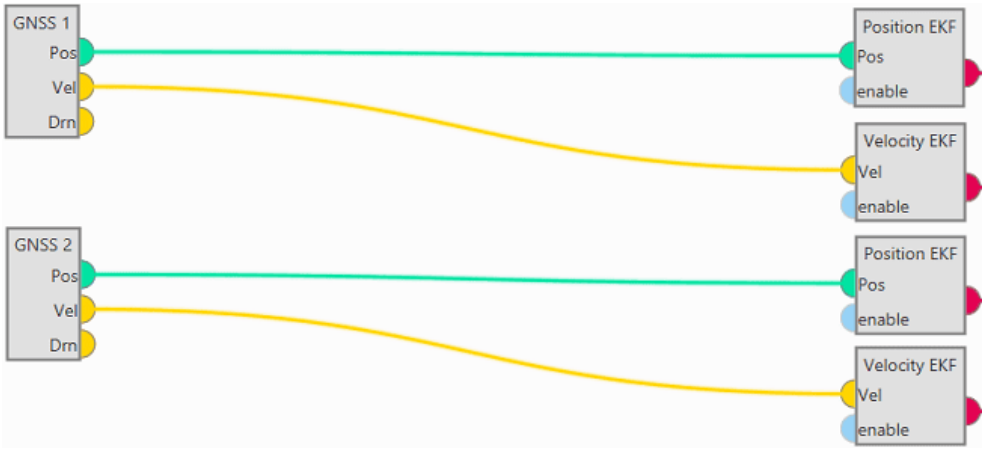
> **Important**
>
> Each sensor used in the autopilot navigation must be connected to its corresponding **EKF adapter** in order to be read correctly by the Navigation block algorithm.

### GNSS Sensors

> **Important**
>
> The GNSS sensor block must be connected to two different EKF adapters for its two outputs, Position EKF block and Velocity EKF block.

For more information on this block, visit the GNSS sensor - Sensors blocks section of the **1x PDI Builder** user manual.

- GNSS blocks



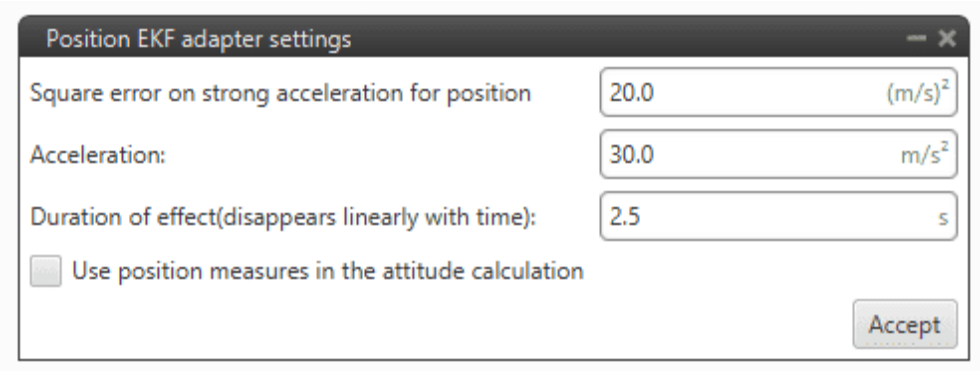
- GNSS
  - Meas Rate: 0.25 s
  - PPP: Disabled
- Survey In: Disabled
- SPI Port - Mask in
  - ublox: Enabled
  - nmea: Disabled

        - rtcm: Disabled
        - rtcm3: Enabled
    - SPI Port - Mask out
        - ublox: Enabled
        - nmea: Disabled
        - rtcm3: Disabled

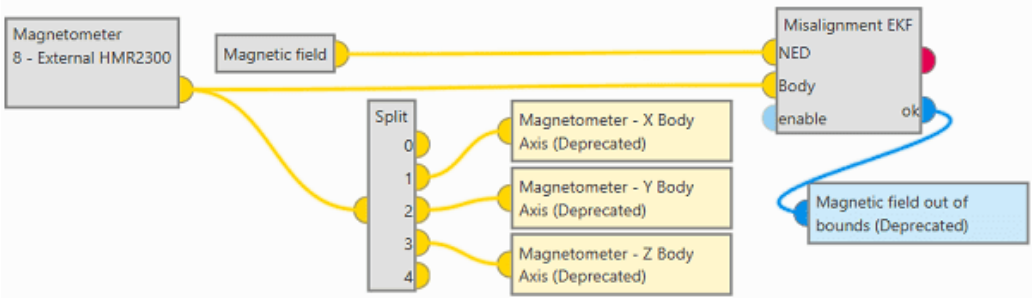
- Position/Velocity EKF Adapters

> ⓘ **Note**
> The EKF adapters used for the GNSS sensor block, Position EKF block and Velocity EKF block, have the same configuration parameters, so only one of them is shown, the other being similar.

    - Square error on strong acceleration for position = 20.0; $(m/s)^2$
    - Acceleration = 30.0; $m/s^2$
    - Duration of effect (disappears lineary with time) = 2.5; $s$
    - Use position measures in the attitude calculation: *Disabled*




### Magnetometer Sensor



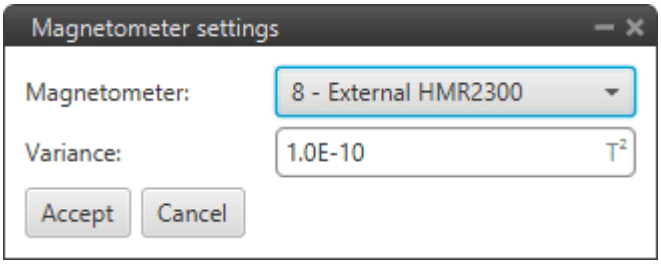

For more information, visit the Magnetometer - Sensors blocks section of the **1x PDI Builder** user manual.

The basic configuration of these blocks is shown below:

- Magnetic field block

  This block has no configuration. For a detailed explanation about it, refer to the Magnetic Field - Sensors blocks section of the **1x PDI Builder** user manual,

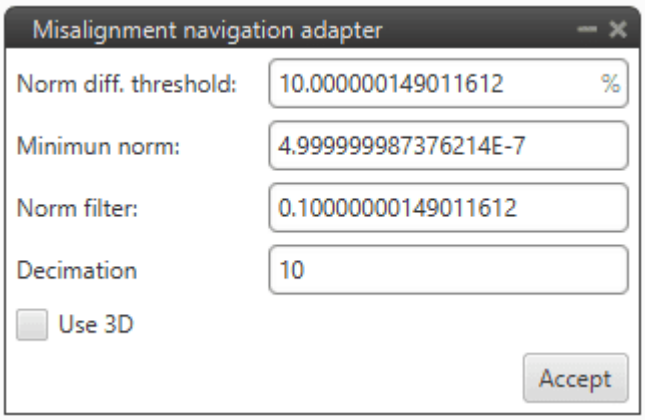
- Magnetometer block

  - 


    Magnetometer: External HMR2300

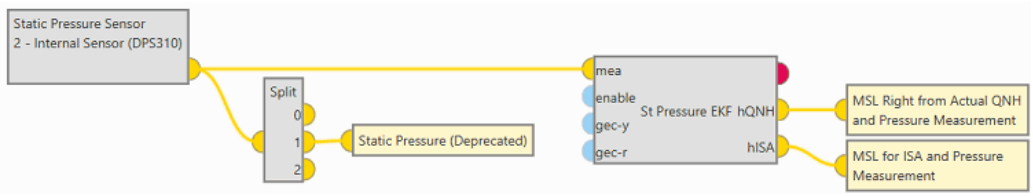
  - Variance: 1.0E-10 T²

- Misalignment EKF Adapter

- 

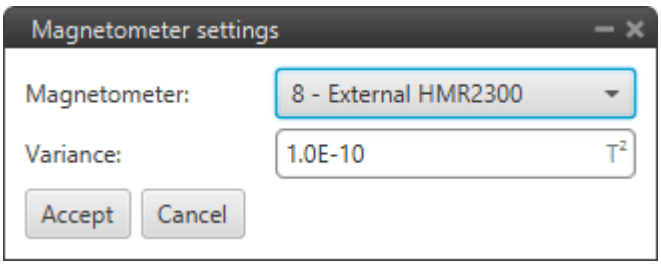

  Norm diff. threshold: 10 %
  - Minimun norm: 5E-7
  - Norm filter: 0.1
  - Decimation: 10
  - Use 3D: Disabled

### Static Pressure Sensor




For more information, visit the Static Pressure - Sensors blocks section of the **1x PDI Builder** user manual.

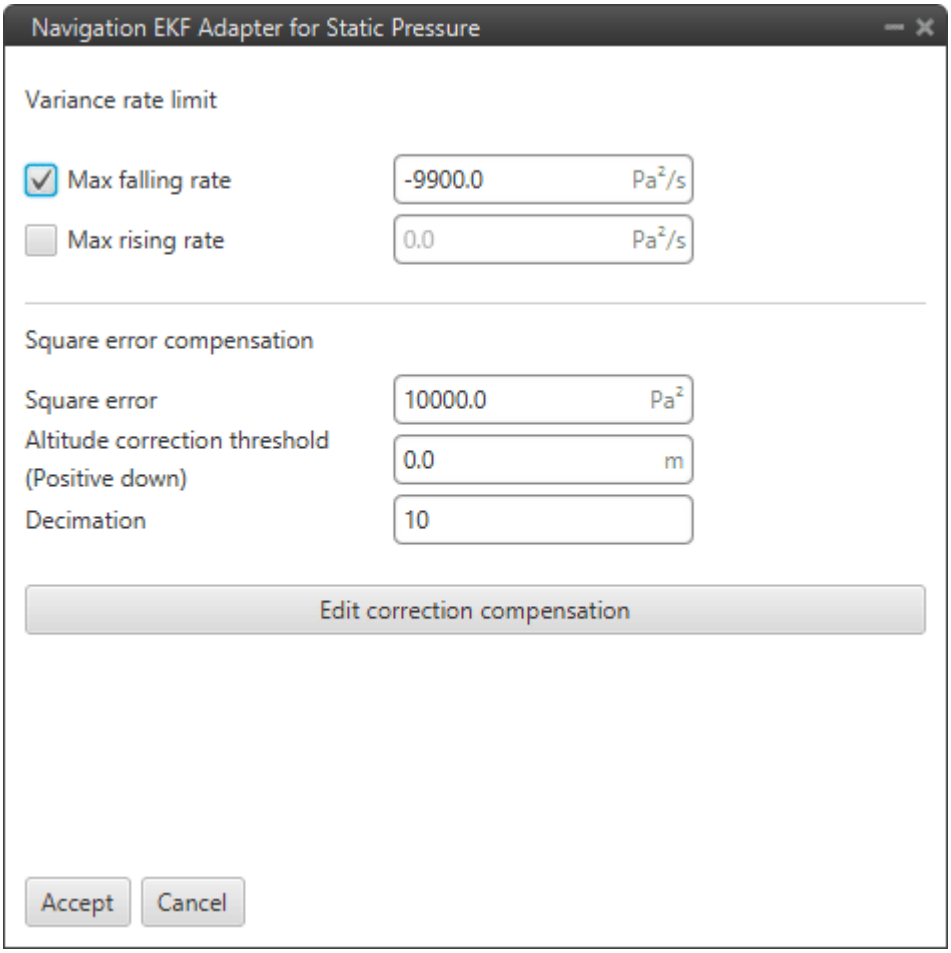
The basic configuration of these blocks is shown below:

- Static Pressure block
  - 


    Static pressure sensor: Internal Sensor (DPS310)

- Variance: 1000.0 Pa²

- Static Pressure EKF Adapter




- Max falling rate: Enabled, ⇒ -9900.0 Pa²/s
- Max rising rate: Disabled
- Square error: 10000.0 Pa²
- Altitude correction threshold: 0.0 m
- Decimation: 10
- Edit correction compensation:

  Points entered:

  -5.0 ; -1.5

-4.0 ; -0.5

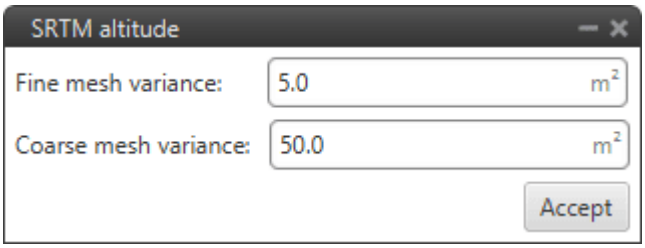
-0.5 ; -0.5

1.0 ; 1.0

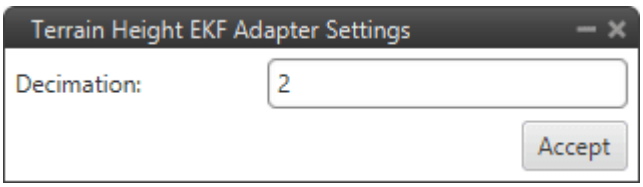
### SRTM height Sensor

For more information, visit the SRTM height - Sensors block section of the **1x PDI Builder** user manual.

The basic configuration of these blocks is shown below:

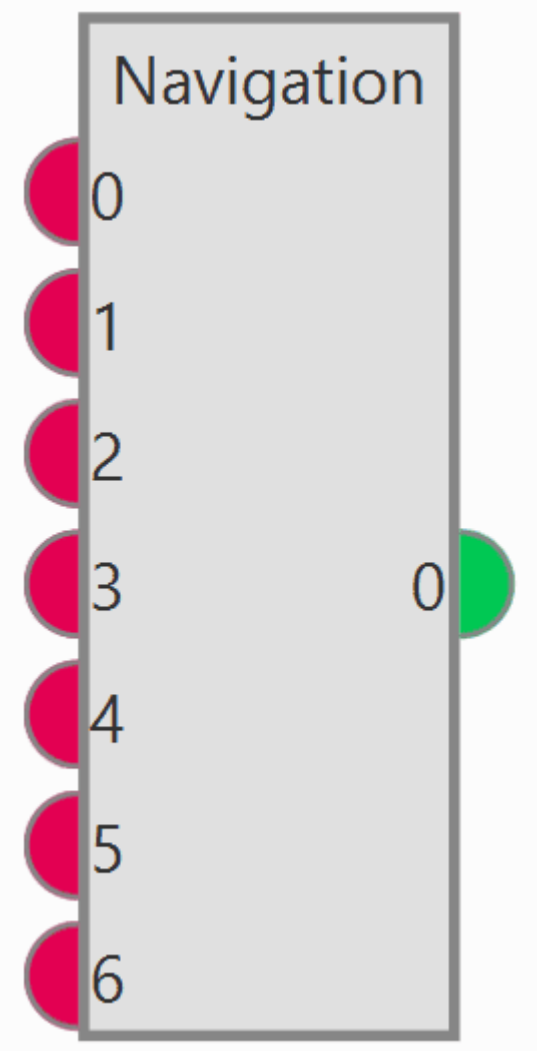
- SRTM height block
  - 


    Fine mesh variance: 5.0 m²
  - Coarse mesh variance: 50.0 m²

- Terrain height EKF Adapter
  - 

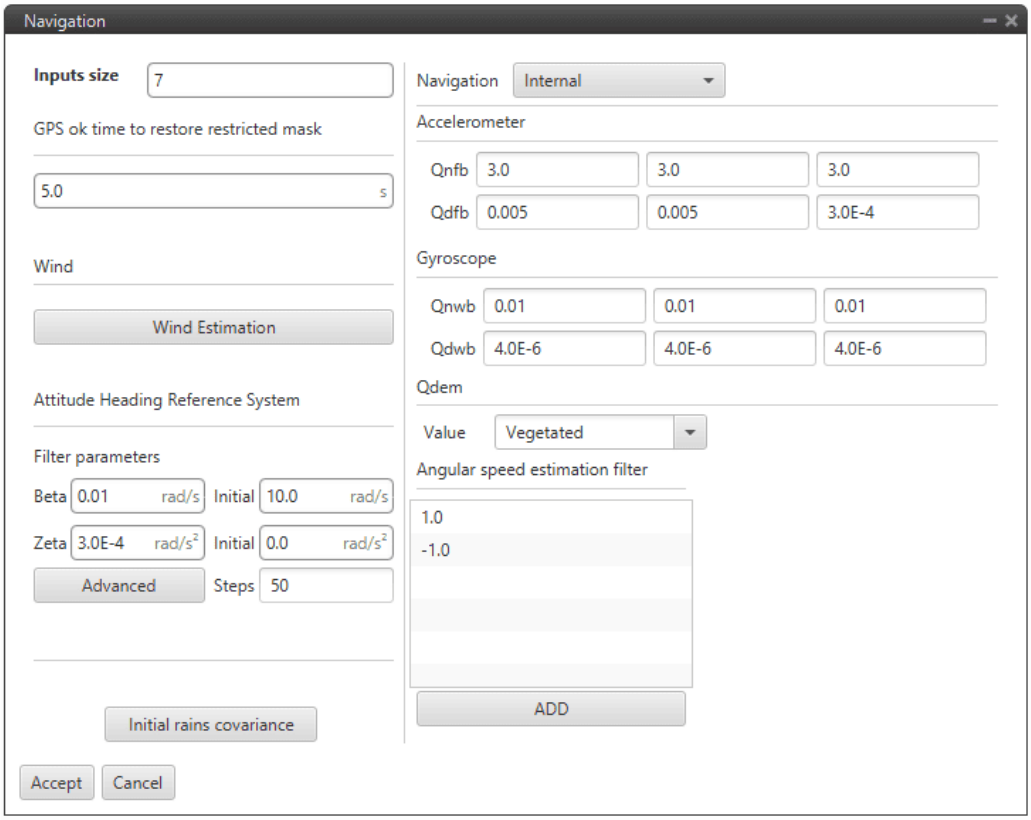

    Decimation: 2

### Navigation Block




For more information, visit the Navigation - Navigation blocks section of the **1x PDI Builder** user manual.

The basic configuration of this block is shown below:




- Inputs size: **7**

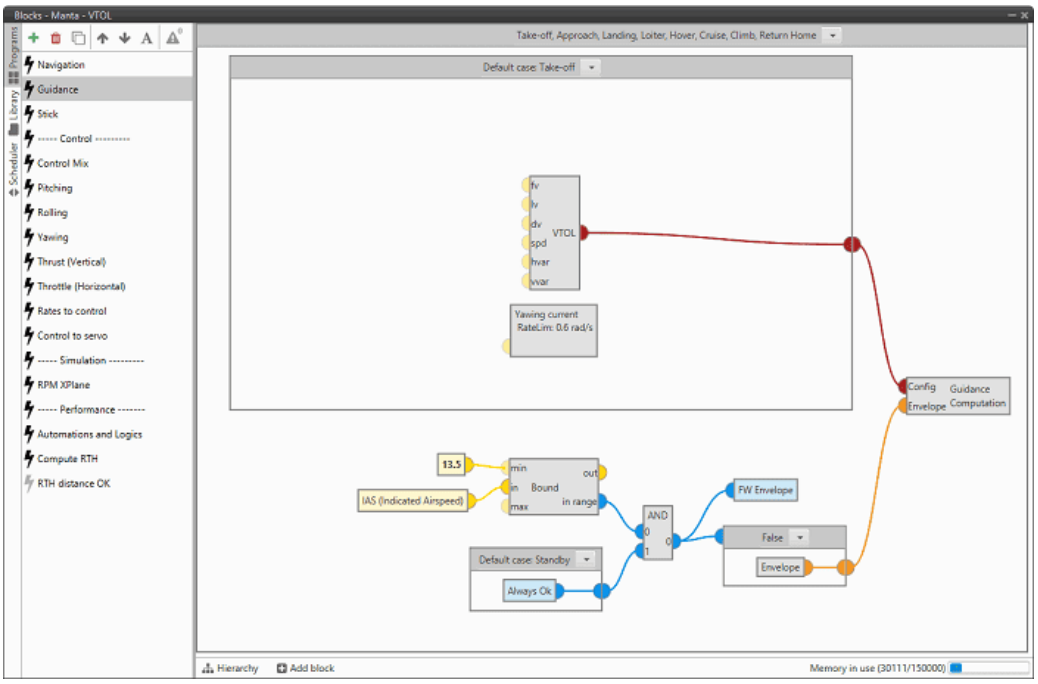
> **Important**
>
> Navigation block is configured to have as many inputs as EKF adapters are used in the sensors. These adapters are the ones that transmit the measurements to the Navigation block.

- GPS ok time to restore restricted mask: **5.0**
- Filter parameters
  - Beta: $0.01\ rad/s$ ; Initial: $10.0\ rad/s$
  - Zeta: $3.0E-4\ rad/s^2$ ; Initial: $0.0\ rad/s^2$
- Navigation: *Internal*
- Accelerometer
  - Qnfb: $3.0$ ; $3.0$ ; $3.0$
  - Qdfb: $0.005$ ; $0.005$ ; $3.0E-4$
- Gyroscope
  - Qnwb: $0.01$ ; $0.01$ ; $0.01$
  - Qdwb: $4.0E-6$ ; $4.0E-6$ ; $4.0E-6$

- Qdem
  - Value: *Vegetated*

## Guidance

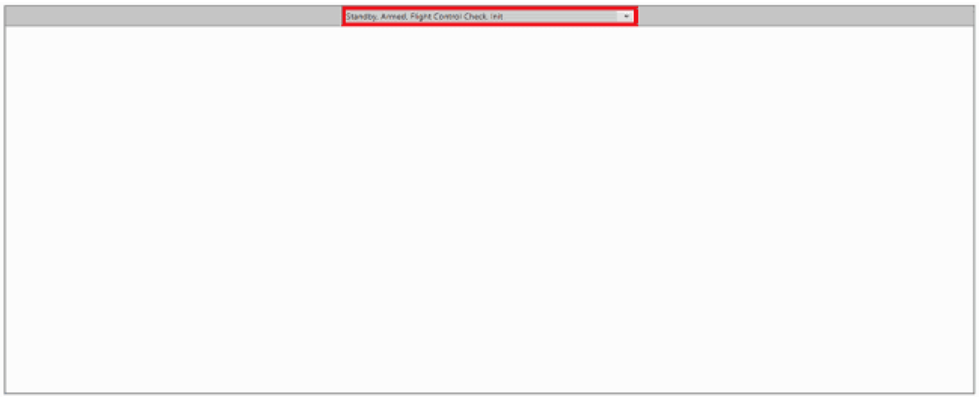
The guidance program defines the **control laws** in each of the flight phases.




**Guidance - Program**

- **Standby, Armed, Flight Control Check, Init**

  In these flight phases, guidance control is not necessary since the aircraft is not flying.

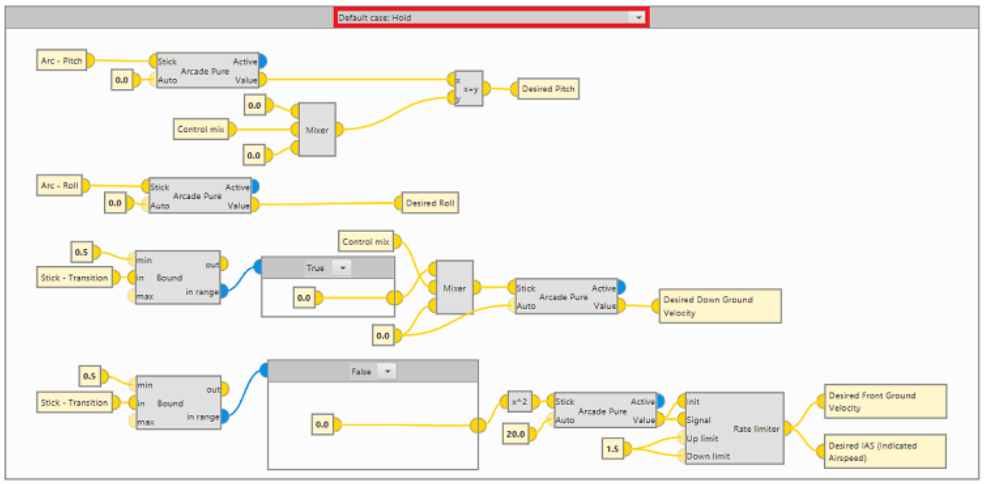



**Guidance - Standby, Armed, Flight Control Check and Init phases**

- **Hold**

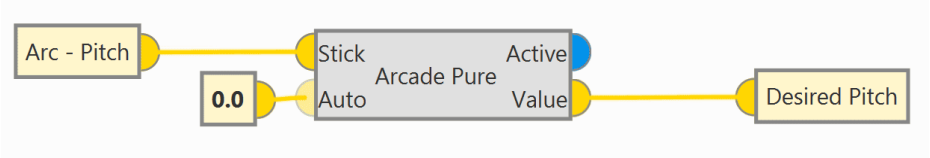
The Hold flight phase stabilizes the attitude and vertical speed of the aircraft.

> **Important**
>
> The values in Auto mode of the control variables are null, being the Stick commands the ones that modify these variables.

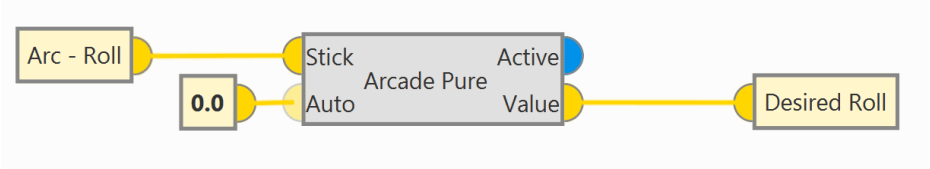



**Guidance - Hold phase**

- The **Desired Pitch** value is obtained from the Arcade Pure block. The Stick input reads the value of the Arc-Pitch variable, which will directly define the **Desired Pitch** when the aircraft is in Arcade mode.

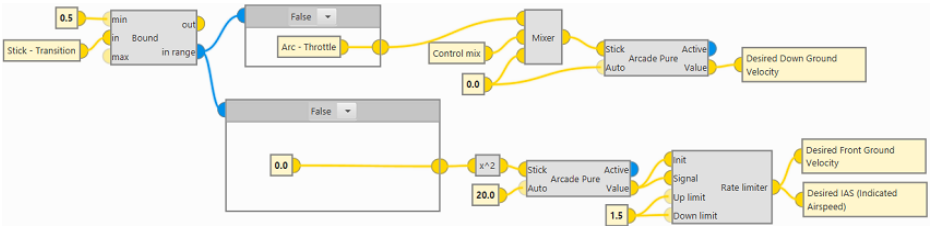



- The **Desired Roll** value is obtained from the Arcade Pure block. The Stick input reads the value of the Arc-Roll variable, which will directly define the **Desired Roll** when the aircraft is in Arcade mode.

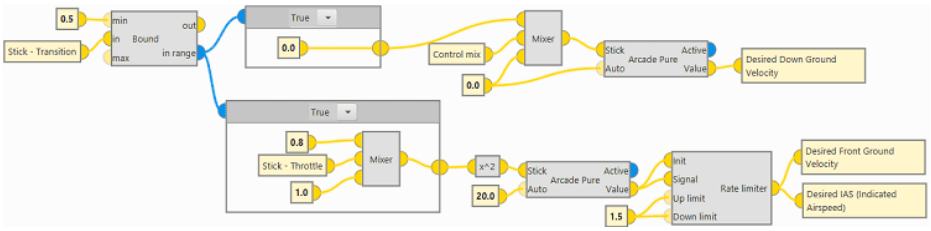



- If the value of the **Stick-Transition** variable is **less than 0.5** (aircraft flying in **quadcopter configuration**):
  - The **Desired Down Ground Velocity** value is obtained from the Arcade Pure block, where the Stick input directly reads the value of the Arc-Throttle variable.

- The **Desired Front Ground Velocity** and **Desired IAS** values are obtained from the Arcade Pure block in Auto mode with a value of **20.0**.

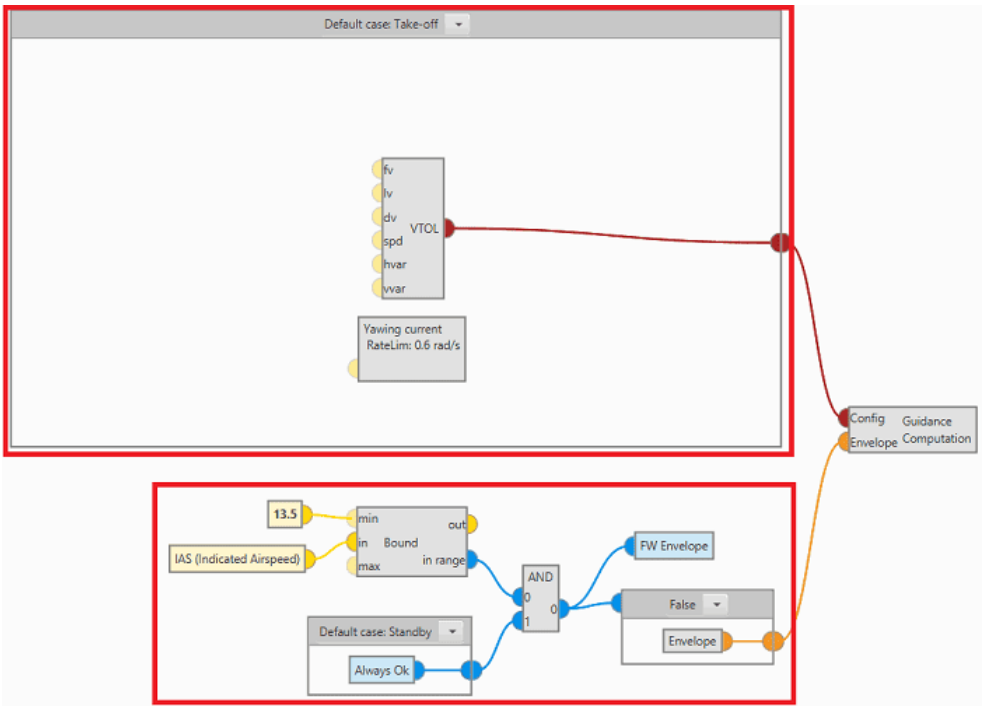



- If the value of the **Stick-Transition** variable is **greater than 0.5** the aircraft transitions to **FW configuration** and the control variable becomes **Desired Front Ground Velocity**:
  - The **Desired Down Ground Velocity** value is null.
  - The **Desired Front Ground Velocity** and **Desired IAS** values are obtained from the Arcade Pure block, in which the Stick input has a value dependent on the Stick-Throttle variable.




- **Take-off, Approach, Landing, Loiter, Hover, Cruise, Climb, Return Home**

  In the following phases the guidance control is mainly configured with the Guidance Computation block, which is set with two inputs: **Config** and **Envelope**. The latter being common for this group of phases.



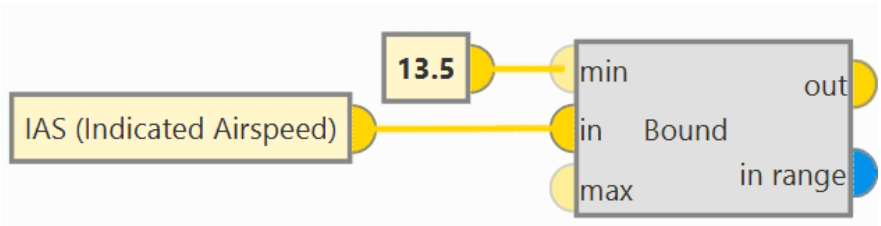

**Guidance - Take-off, Approach, Landing, Loiter, Hover, Cruise, Climb and Return Home phases**

- **Envelope input**

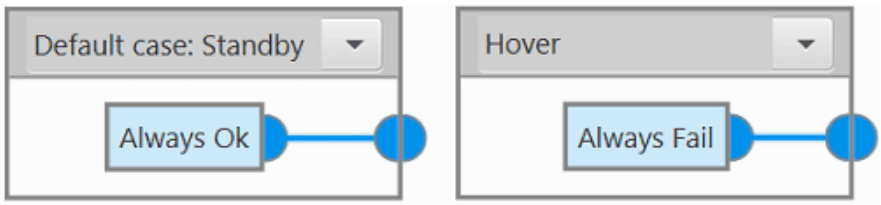
  The Envelope input must always be connected to the Envelope block. This block defines the limits that must not be exceeded during operation, so these limits will depend on the flight configuration (Multi or FW).

  To establish which flight configuration the aircraft is in, the following logic is defined:

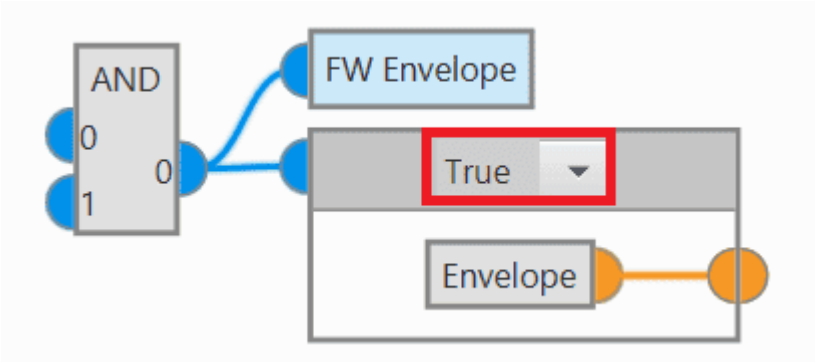
  1. The Bound block returns the bit in range **true** if IAS is greater than **13.5 m/s**.

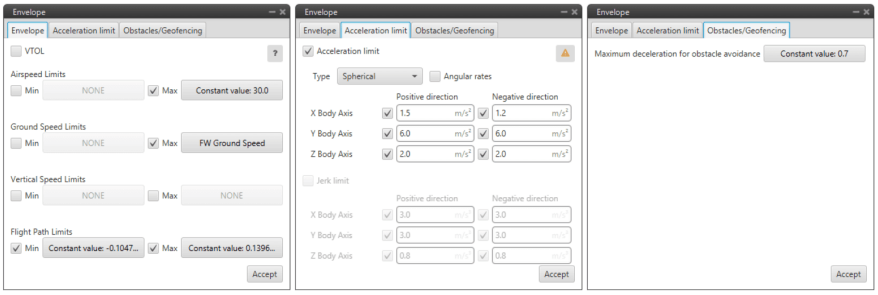
     
     

  2. Always Ok bit in flight phase Standby or Always Fail bit in flight phase Hover.



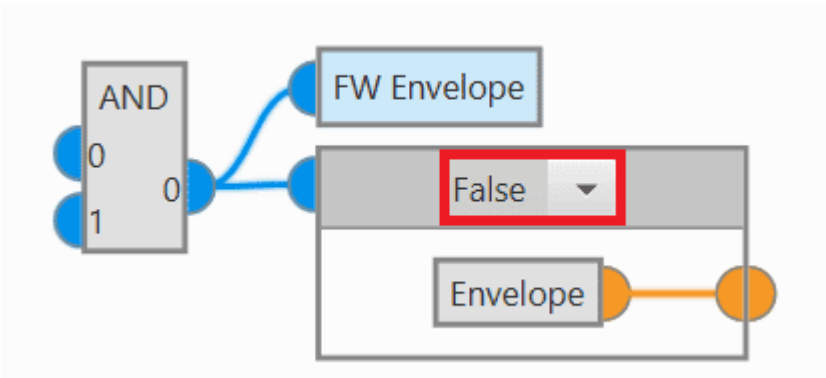

3. If both **bits** (1 and 2) are **true**, the FW Envelope bit is **true** and the Envelope block is defined for the **FW configuration**.

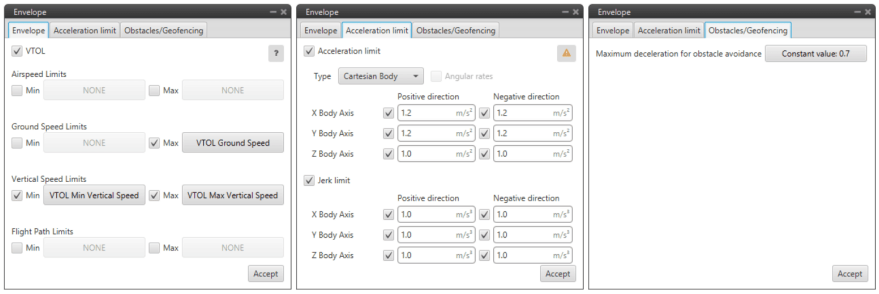



**Envelope block - FW configuration**

4. If it is **not** satisfied that both **bits** (1 and 2) are **true**, the FW Envelope bit is **false** and the Envelope block is set for the **quadcopter configuration**.






**Envelope block - Quadcopter configuration**

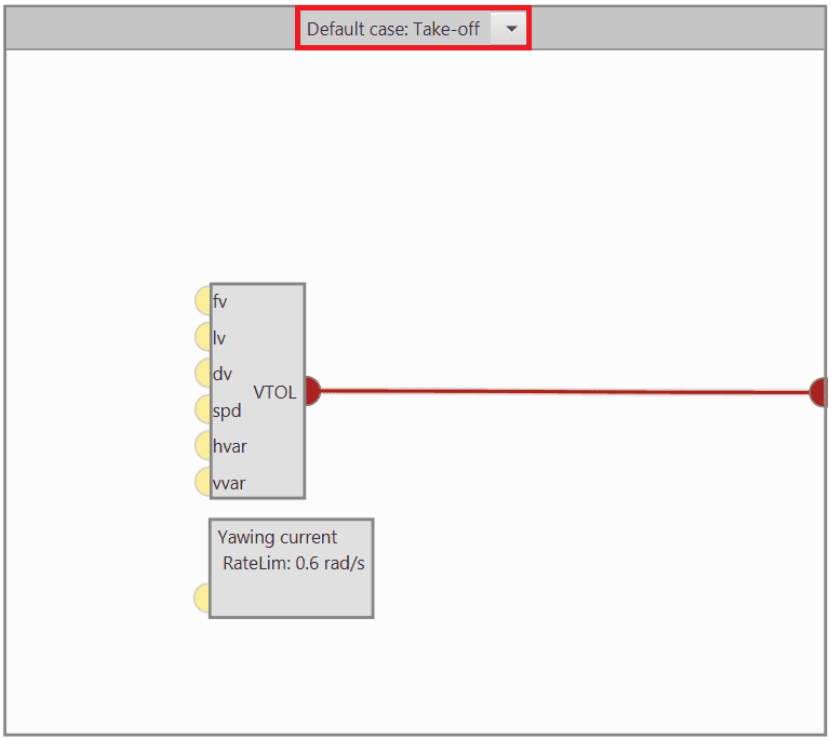
- **Config input**

  The Config input must always be connected to a Guidance block, which will vary depending on the flight phase.

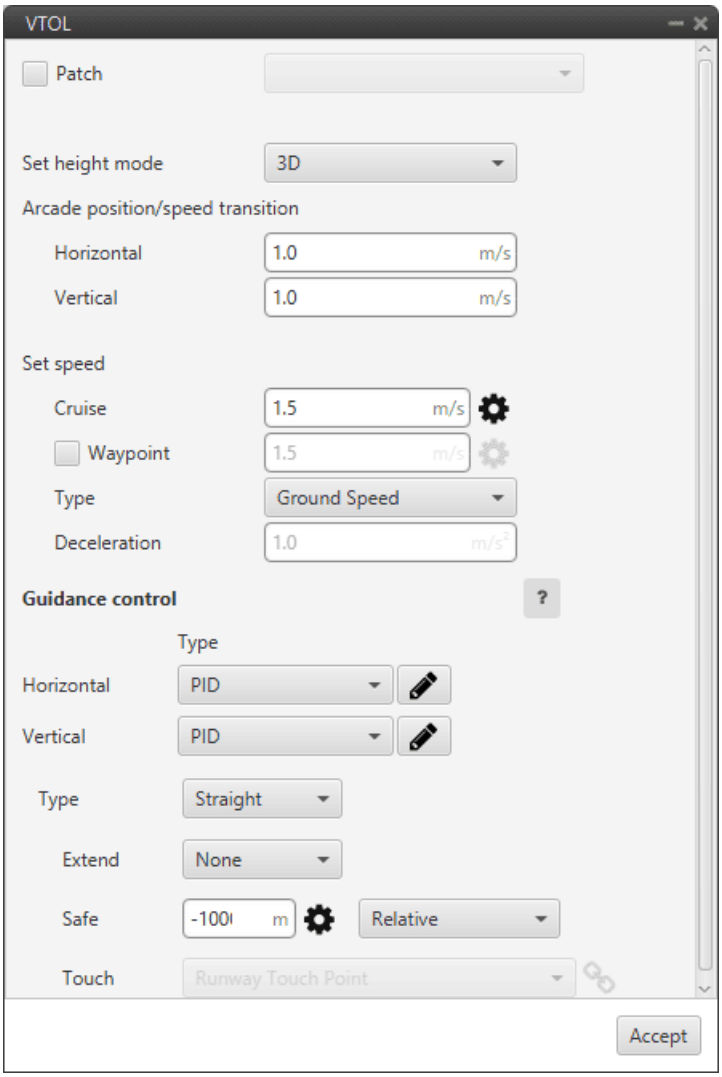
  - **Take-off**

    The guidance configuration in this phase consists of the VTOL block, which is linked to the Config input, and the Yawing current block.

    - **VTOL** guidance is used in multicopters for the **take-off** and landing operations.




A basic configuration of this block is shown below:




- Patch: Disabled
- Set height mode: 3D
- Arcade position/speed transition

  Horizontal: 1.0 m/s

  Vertical: 1.0 m/s
- Set speed

  Cruise: 1.5 m/s

  Waypoint: Disabled

  Type: Ground Speed

Deceleration: 1.0 m/s²

- Guidance control

  Horizontal: PID

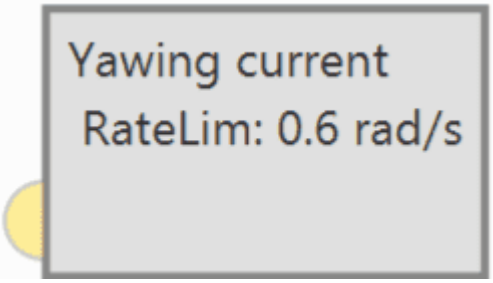
  Vertical: PID

  Type: Straight

  Extend: None

  Safe: -1000.0 m ⇒ Relative

- With the Yawing current block it is intended that the aircraft **maintains the yaw angle it has when entering the phase**, i.e. Desired Yaw = Current Yaw.



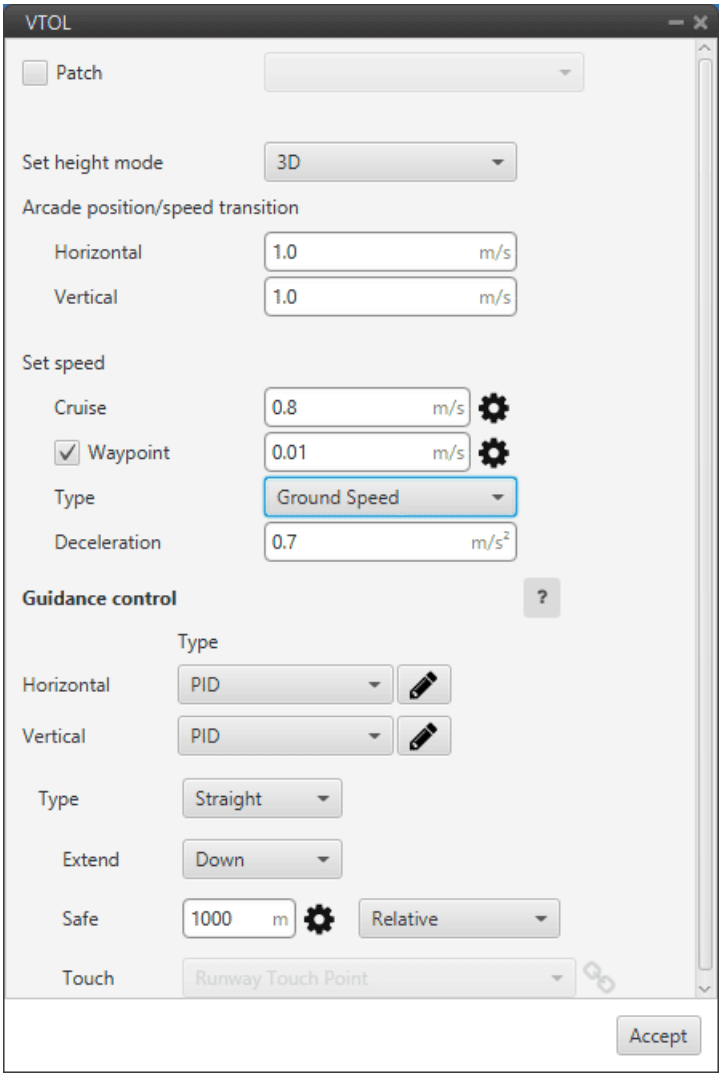
  In addition, a **limit rate value** of $0.6\ rad/s$ has been defined to control the speed at which the aircraft yaws in the guidance of this phase.

- **Landing**

  The guidance configuration in this phase consists of the VTOL block, which is linked to the Config input, and the Yawing current block.

  - **VTOL** guidance is used in multicopters for the take-off and **landing** operations.

A basic configuration of this block is shown below:




- Patch: Disabled

- Set height mode: 3D
- Arcade position/speed transition

  Horizontal: 1.0 m/s

  Vertical: 1.0 m/s
- Set speed

  Cruise: 0.8 m/s

  Waypoint: 0.01 m/s

  Type: Ground Speed

  Deceleration: 0.7 m/s²
- Guidance control

  Horizontal: PID

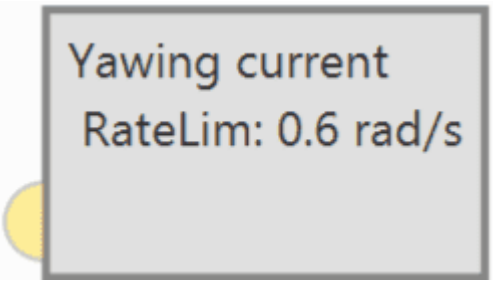
  Vertical: PID

  Type: Straight

  Extend: Down

  Safe: -1000.0 m ⇒ Relative

- With the Yawing current block it is intended that the aircraft **maintains the yaw angle it has when entering the phase**, i.e. Desired Yaw = Current Yaw.

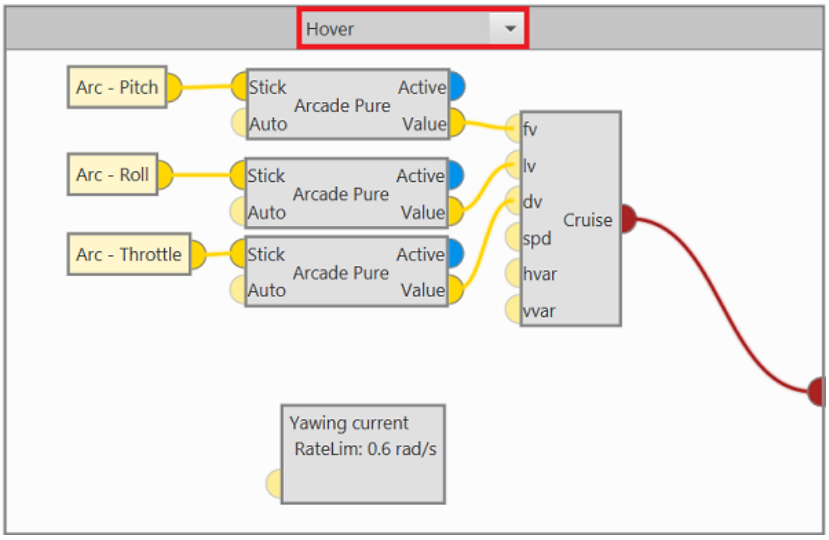


In addition, a **limit rate value** of $0.6\ rad/s$ has been defined to control the speed at which the aircraft yaws in the guidance of this phase.

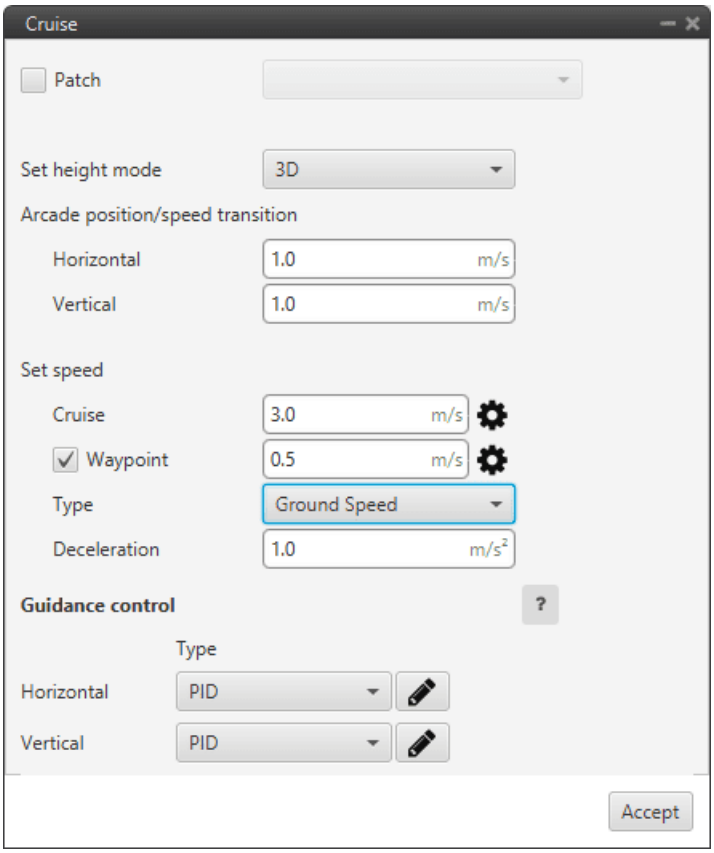
- **Hover**

  The guidance configuration in this phase consists of the Cruise block, which is linked to the Config input, and the Yawing current block.

  - **Cruise** guidance is used to make the aircraft follow a position-based route created by the user.




A basic configuration of this block is shown below:




- Patch: Disabled
- Set height mode: 3D
- Arcade position/speed transition

  Horizontal: 1.0 m/s

  Vertical: 1.0 m/s
- Set speed

  Cruise: 3 m/s

  Waypoint: 0.5 m/s

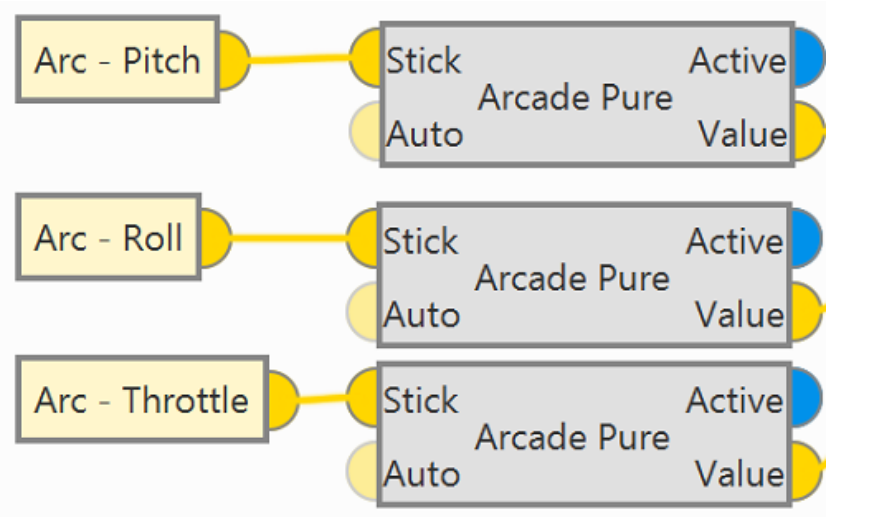
  Type: Ground Speed

  Deceleration: 1.0 m/s²
- Guidance control

  Horizontal: PID

  Vertical: PID

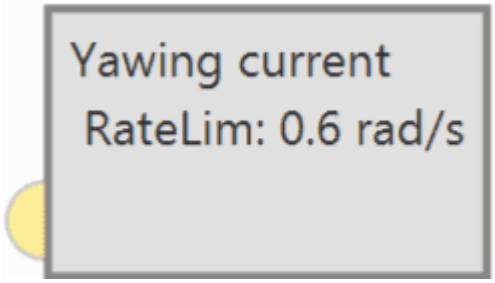
Moreover, the output of each of the **Arcade Pure** blocks are used as inputs for this guidance block. These output values match the value of the Stick input when the aircraft is controlled in **Arcade mode**.




- Arcade Pure (Arc-Pitch) ⇒ **fv**: The **Pitch** value commanded by the **stick** will be the **first** component of the desired 'hover here' arcade velocity in the horizontal plane.
- Arcade Pure (Arc-Roll) ⇒ **lv**: The **Roll** value commanded by the **stick** will be the **second** component of the desired 'hover here' arcade velocity in the horizontal plane.
- Arcade Pure (Arc-Throttle) ⇒ **dv**: The **Throttle** value commanded by the **stick** will be the **down** (vertical) component of the desired 'hover here' arcade velocity.

For more information, visit the Guidance blocks common configuration - Guidance blocks section of the **1x PDI Builder** user manual.

- With the Yawing current block it is intended that the aircraft **maintains the yaw angle it has when entering the phase**, i.e. Desired Yaw = Current Yaw.

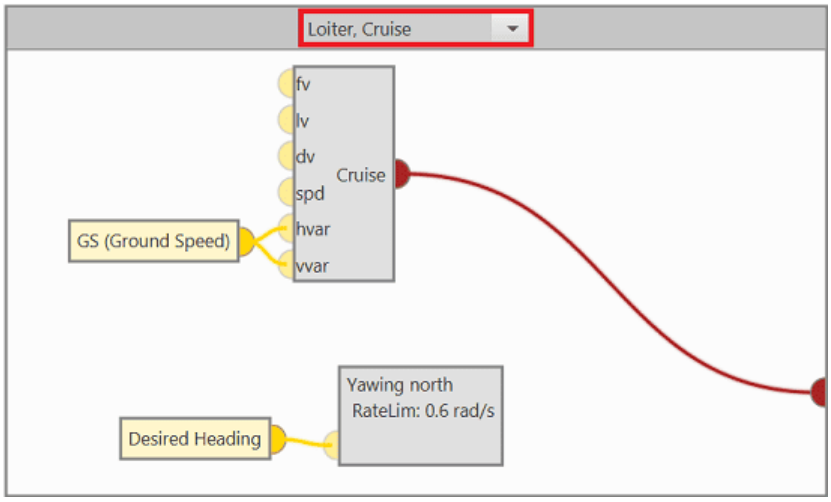


In addition, a **limit rate value** of $0.6\ rad/s$ has been defined to control the speed at which the aircraft yaws in the guidance of this phase.

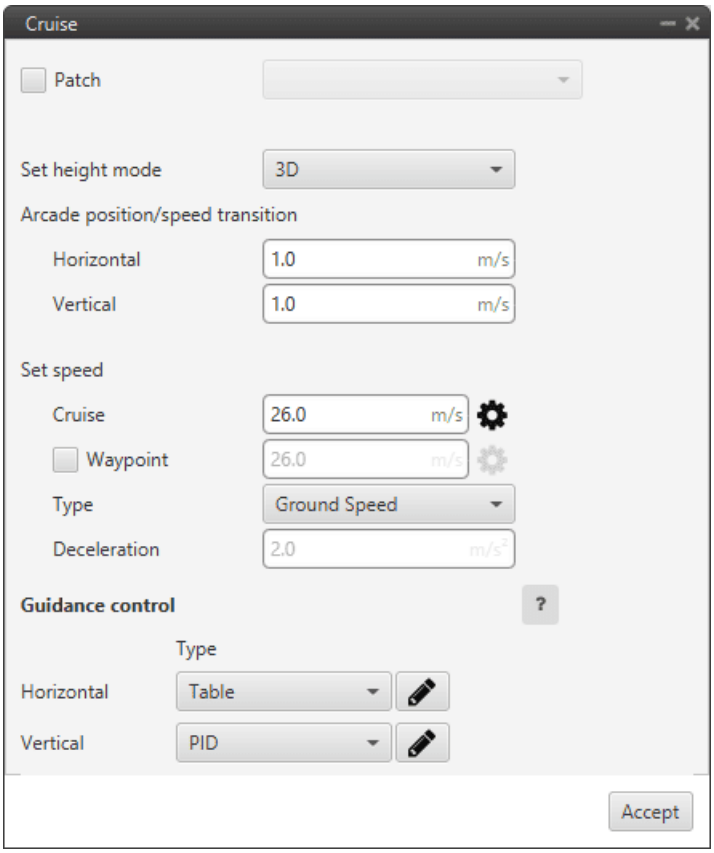
- **Loiter** and **Cruise**

  The guidance configuration for theses phases is composed of the Cruise block, which is linked to the Config input, and the Yawing north block.

  - **Cruise** guidance is used to make the aircraft follow a position-based route created by the user.




A basic configuration of this block is shown below:

- 


  Patch: Disabled

- Set height mode: 3D

- Arcade position/speed transition

  Horizontal: 1.0 m/s

  Vertical: 1.0 m/s

- Set speed

  Cruise: 26 m/s

  Waypoint: Disabled

  Type: Ground Speed

  Deceleration: 2.0 m/s²

- Guidance control

  Horizontal: PID

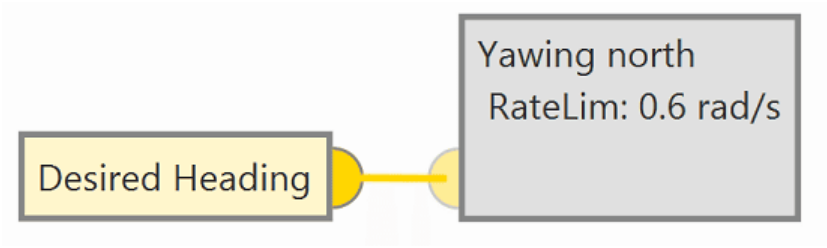
  Vertical: PID

In addition, the GS (Ground Speed) variable is linked to 2 inputs of this guidance block:

- GS (Ground Speed) ⇒ **hvar**: **Horizontal** scale variable used for the T-Sched PID.
- GS (Ground Speed) ⇒ **vvar**: **Vertical** scale variable used for the T-Sched PID.

For more information, visit the Guidance blocks common configuration - Guidance blocks section of the **1x PDI Builder** user manual.

- With the Yawing north block it is desired that the aircraft maintains the yaw to the north (angle 0º). However, if there is any variable connected to the input of the block, this is taken as an offset with respect to the reference, that being 0º, the aircraft will point directly to the angle indicated as offset.

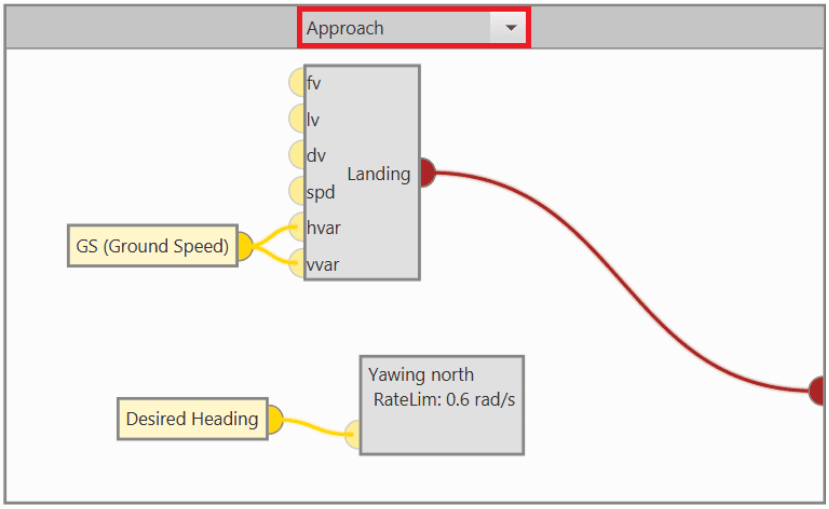


In this case, the variable Desired Heading is connected to the input, so Desired Yaw = Desired Heading.
In addition, a **limit rate value** of $0.6\ rad/s$ has been defined for controlling the speed at which the aircraft yaws in the guidance of this phase.

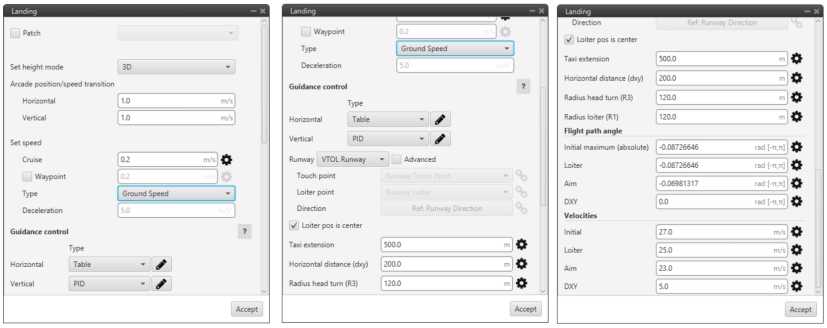
- **Approach**

  The guidance configuration in this phase consists of the Landing block, which is linked to the Config input, and the Yawing north block.

  - **Landing** guidance is used to generate the flying path the aircraft will follow when landing on a certain runway.




A basic configuration of this block is shown below:

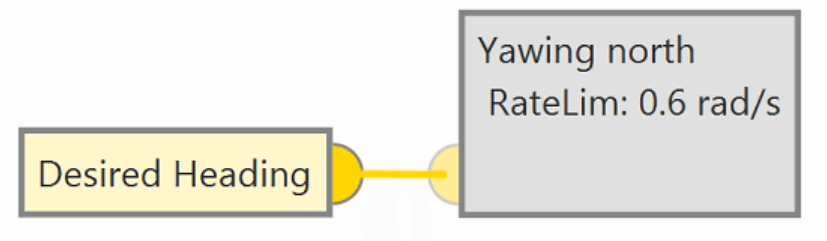


In addition, the GS (Ground Speed) variable is linked to 2 inputs of this guidance block:

- GS (Ground Speed) ⇒ **hvar**: **Horizontal** scale variable used for the T-Sched PID.
- GS (Ground Speed) ⇒ **vvar**: **Vertical** scale variable used for the T-Sched PID.

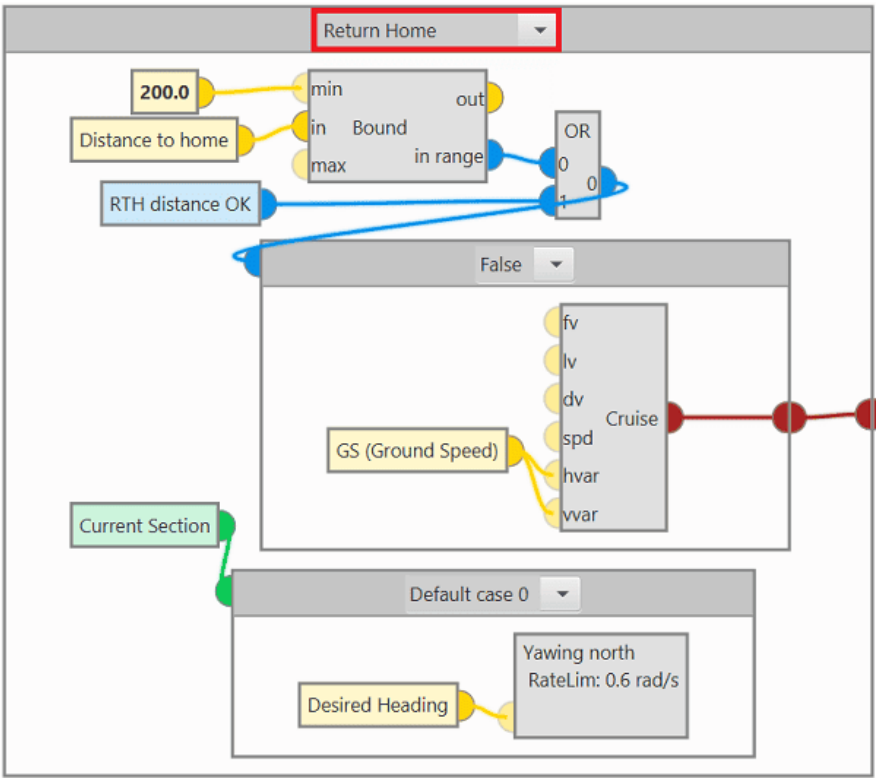
For more information, visit the Guidance blocks common configuration - Guidance blocks section of the **1x PDI Builder** user manual.

- With the Yawing north block it is desired that the aircraft maintains the yaw to the north (angle 0º). However, if there is any variable connected to the input of the block, this is taken as an offset with respect to the reference, that being 0º, the aircraft will point directly to the angle indicated as offset.



In this case, the variable Desired Heading is connected to the input, so Desired Yaw = Desired Heading.
In addition, a **limit rate value** of $0.6\ rad/s$ has been defined for controlling the speed at which the aircraft yaws in the guidance of this phase.

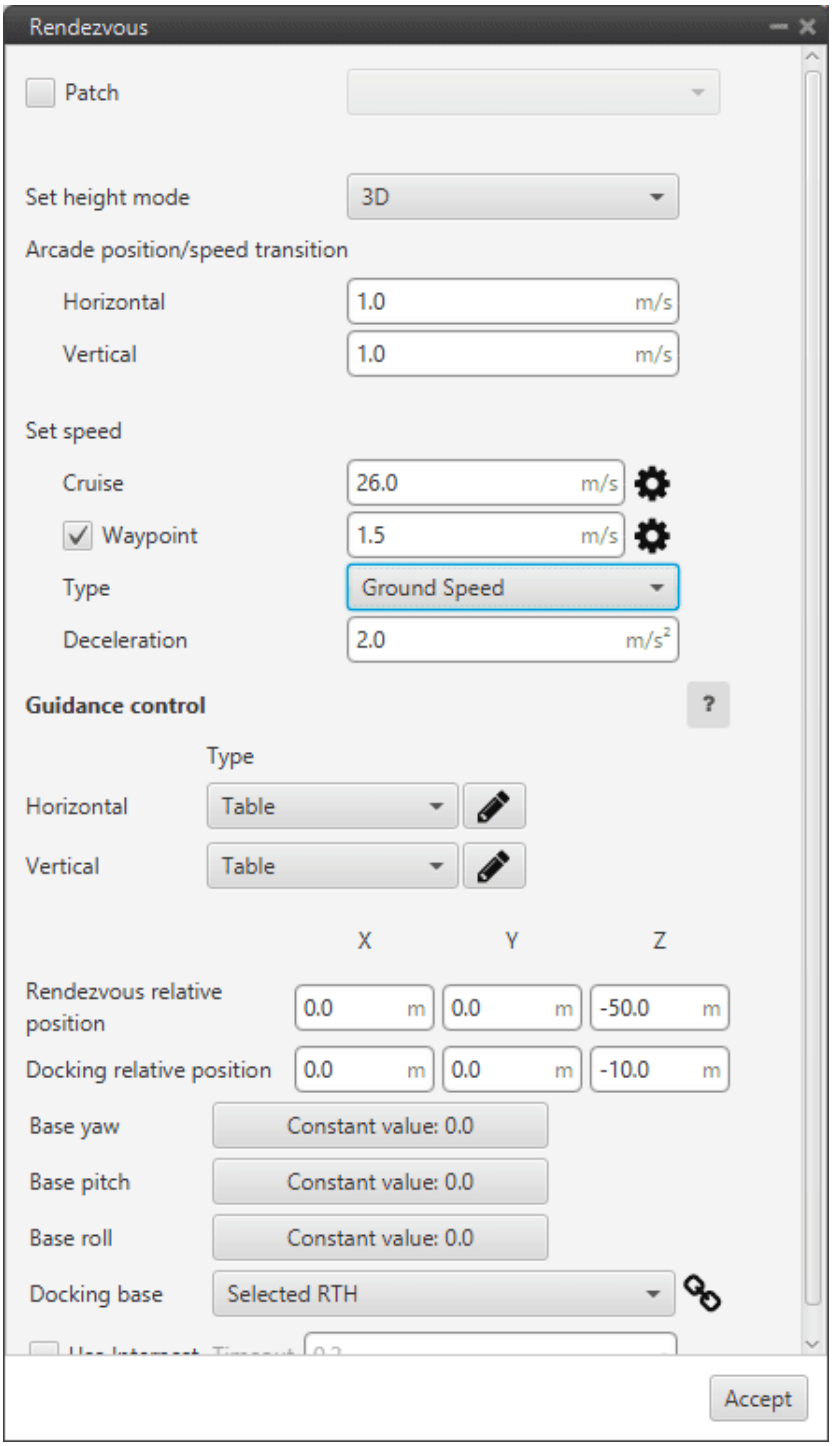
- **Return Home**
  On the one hand, the configuration of the guidance in this phase which will be linked to the Config input, depends on the definition of the **Home** point.




- If the Home point is **defined**, the Rendezvous block defines the guidance configuration for this phase.

> **Explanation**
>
> For the Home point to be defined, one of the two input bits of the **OR** block must be **true**.

**Rendezvous** guidance is used to follow a route to the runway. A basic configuration of this block is shown below:



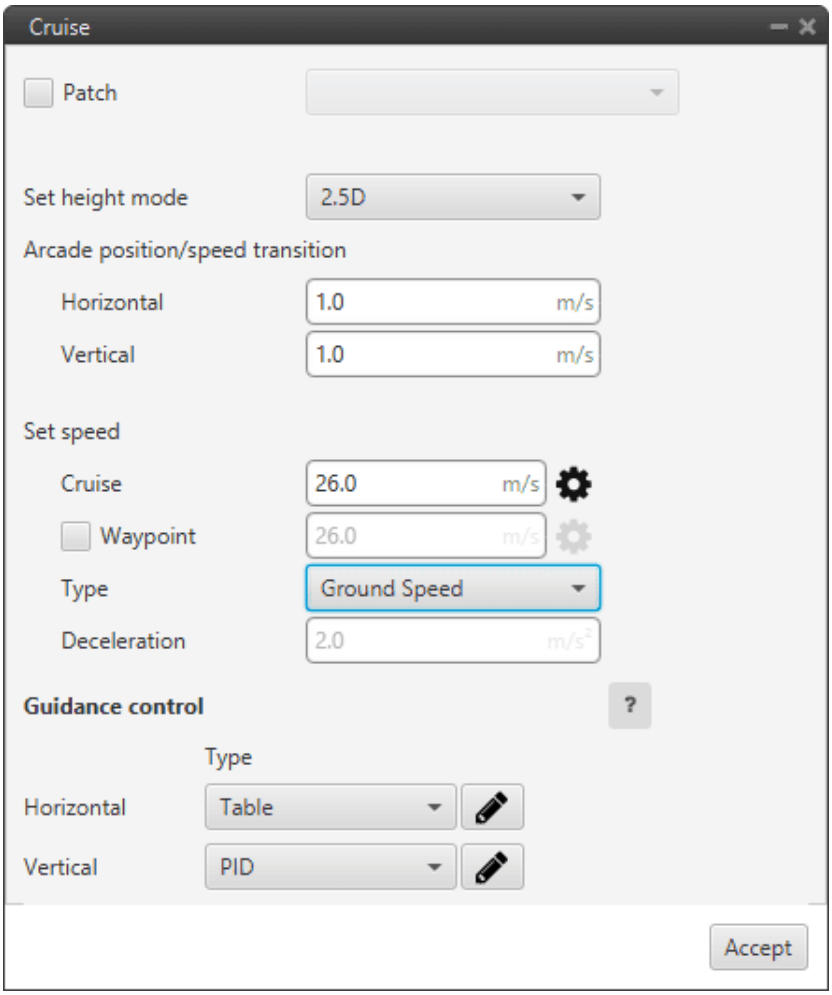

In addition, the GS (Ground Speed) variable is linked to 2 inputs of this guidance block:

- GS (Ground Speed) ⇒ **hvar**: **Horizontal** scale variable used for the T-Sched PID.
- GS (Ground Speed) ⇒ **vvar**: **Vertical** scale variable used for the T-Sched PID.

For more information, visit the Guidance blocks common configuration - Guidance blocks section of the **1x PDI Builder** user manual.

- If the Home point is **not defined**, the Cruise block defines the guidance configuration for this phase.

  **Cruise** guidance is used to follow a route until the Home point is defined. A basic configuration of this block is shown below:

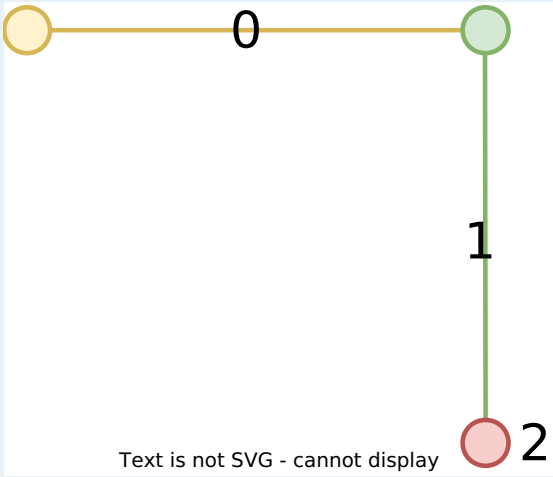



  In addition, the GS (Ground Speed) variable is linked to 2 inputs of this guidance block:

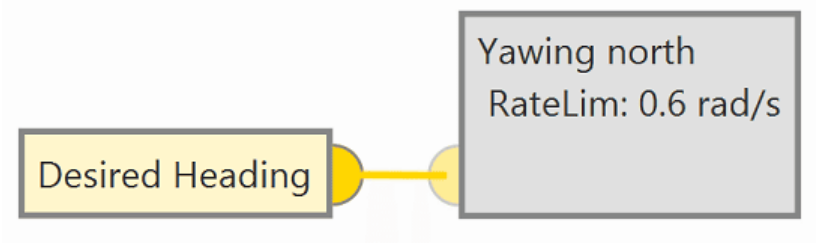
  - GS (Ground Speed) ⇒ **hvar**: **Horizontal** scale variable used for the T-Sched PID.
  - GS (Ground Speed) ⇒ **vvar**: **Vertical** scale variable used for the T-Sched PID.

For more information, visit the Guidance blocks common configuration - Guidance blocks section of the **1x PDI Builder** user manual.

On the other hand, the value of the **Current Section** variable defines the aircraft reference for **Yaw**, i.e. it defines which yawing guidance block will be employed.

> ⓘ **Note**
>
> Rendezvous guidance is divided into 3 sections (0, 1 and 2) where the aircraft behavior changes.
>
> 
> 
>
> **Sections of the rendezvous guidance**

- In **case 0**, the Yawing north block is used.

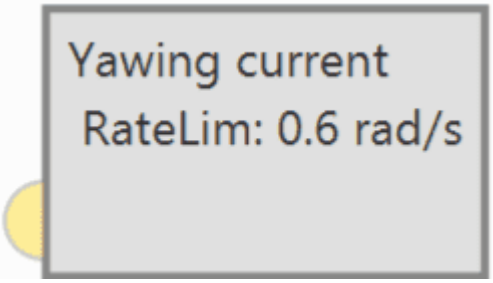


With this block it is desired that the aircraft maintains the yaw to the north (angle 0º).

However, if there is any variable connected to the input of the block, this is taken as an offset with respect to the reference, that being 0º, the aircraft will point directly to the angle indicated as offset.

In this case, the variable Desired Heading is connected to the input, so Desired Yaw = Desired Heading.
In addition, a **limit rate value** of $0.6\ rad/s$ has been defined for controlling the speed at which the aircraft yaws in the guidance of this phase.

- In **cases 1 and 2**, the Yawing current block is used:



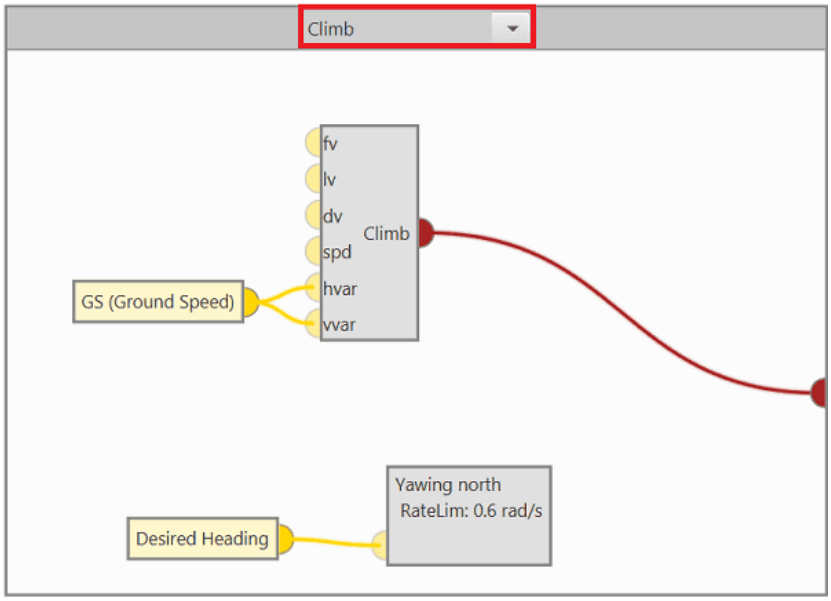
With the Yawing current block it is intended that the aircraft **maintains the yaw angle it has when entering the phase**, i.e. Desired Yaw = Current Yaw.

In addition, a **limit rate value** of $0.6\ rad/s$ has been defined to control the speed at which the aircraft yaws in the guidance of this phase.

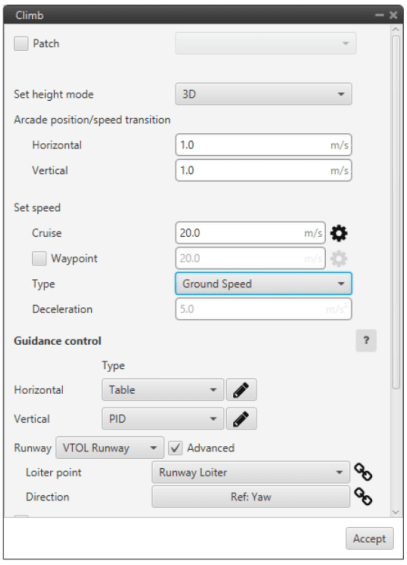
- **Climb**

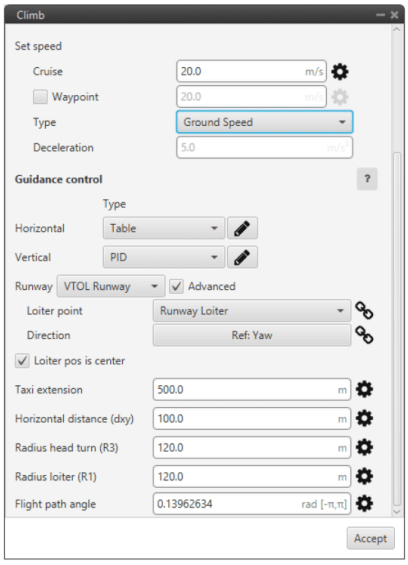
The guidance configuration in this phase consists of the Climb block, which is linked to the Config input, and the Yawing north block.

- **Climb** guidance is used to make the aircraft climb from the start of the phase to another altitude.




A basic configuration of this block is shown below:



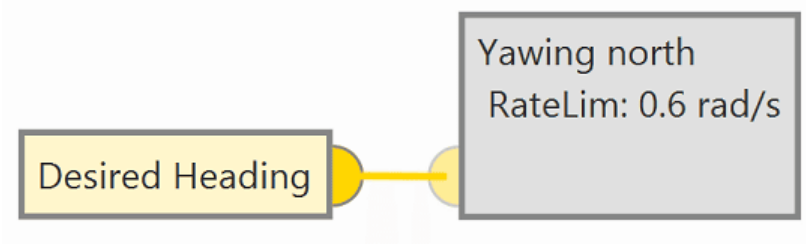

In addition, the GS (Ground Speed) variable is linked to 2 inputs of this guidance block:

- GS (Ground Speed) ⇒ **hvar**: **Horizontal** scale variable used for the T-Sched PID.

- GS (Ground Speed) ⇒ **vvar**: **Vertical** scale variable used for the T-Sched PID.

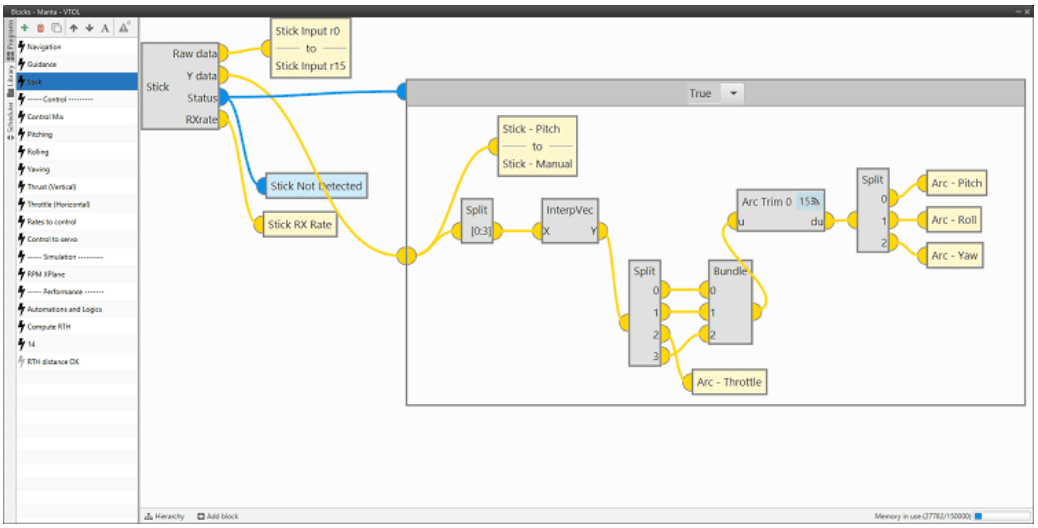
For more information, visit the Guidance blocks common configuration - Guidance blocks section of the **1x PDI Builder** user manual.

- With the Yawing north block it is desired that the aircraft maintains the yaw to the north (angle 0º). However, if there is any variable connected to the input of the block, this is taken as an offset with respect to the reference, that being 0º, the aircraft will point directly to the angle indicated as offset.



In this case, the variable Desired Heading is connected to the input, so Desired Yaw = Desired Heading.
In addition, a **limit rate value** of $0.6\ rad/s$ has been defined for controlling the speed at which the aircraft yaws in the guidance of this phase.

Stick




**Stick - Program**

Users can observe 2 parts in this program:

- **Stick block**

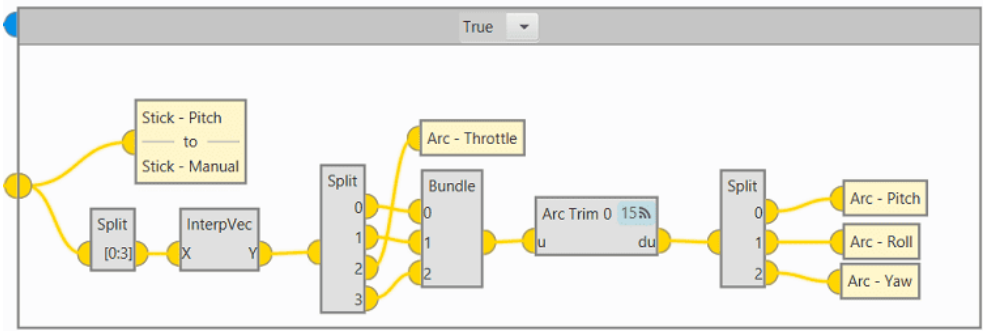
  Stick block is a stick reader, with it the user can configure the stick parameters for manual and arcade modes.

  - The configuration of this block while performing **simulation** is detailed in the Stick - Simulation section of this manual.
  - In the section Stick - Air-GCS Connection of this manual, the user can find more information on configuring the **Stick** block for **real operation**.

- **If-Else Switch block**

  This block is defined to write the control variables that are commanded by the stick:

  - **Arc - Pitch**
  - **Arc - Roll**
  - **Arc - Yaw**
  - **Arc - Throttle**

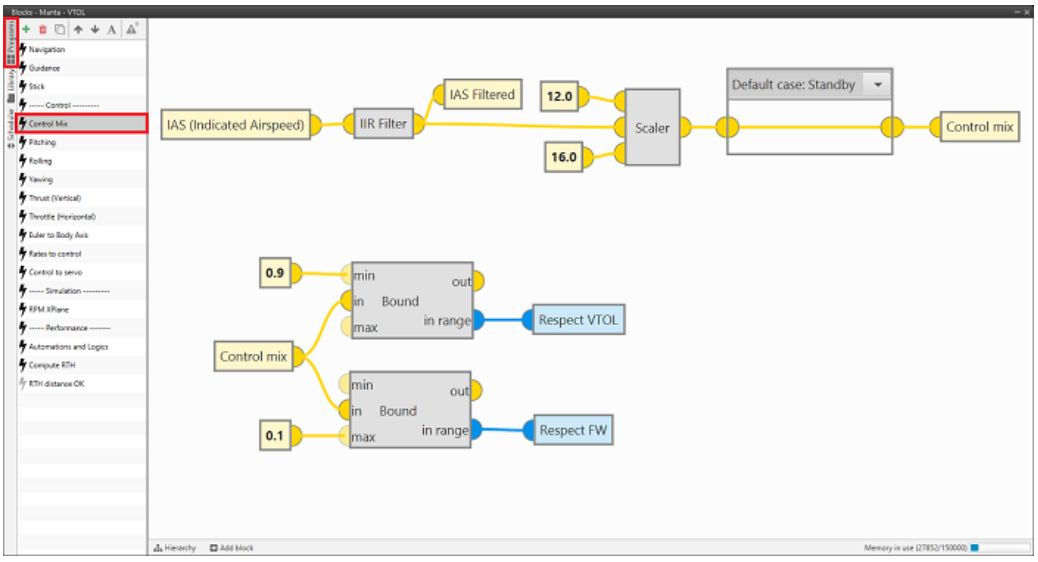



The behavior of this block depends on the value of the output Status of the Stick block. When this output is **True**, the output Y data of the Stick block is linked to this block as an entry and the transformation of the values of this output follows the following logic:

1. The Split block is used to transform the input vector to form one with the first four variables of the input vector.
2. This vector is interpolated with the InterpVec block.
3. With the Split and Bundle blocks, this vector is transformed into a three-component vector.
4. Finally, the vector is passed through the Arc Trim block and the variables are named by separating them with the Split block.

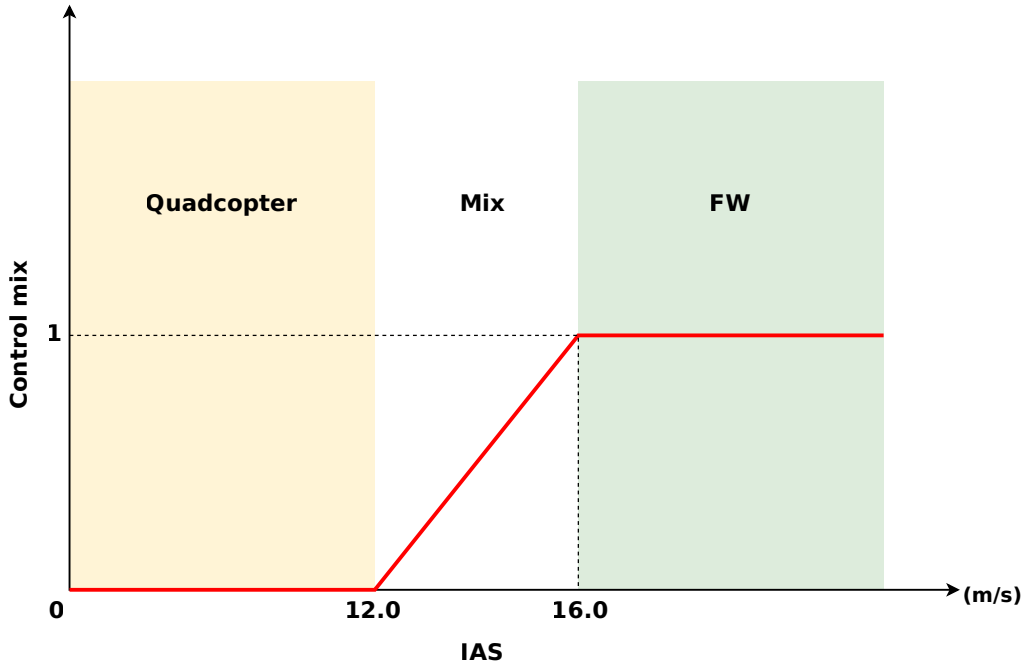
Control Mix

Control Mix - Program is based on the value of the IAS (Indicated Airspeep) to determine the **flight configuration** of the platform (**Multi or FW**). The program executes several calculations to determine the Control mix variable, which has a value from 0 to 1, and indicates the speed at which the platform should switch its flight configuration to FW.



### Control Mix - Program

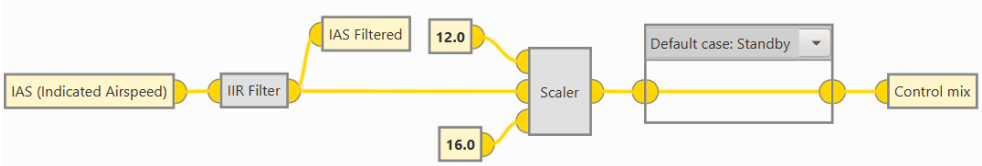
Analytically, the following diagram shows the operation of the program, which provides a value to the Control mix variable depending on the value of the IAS.



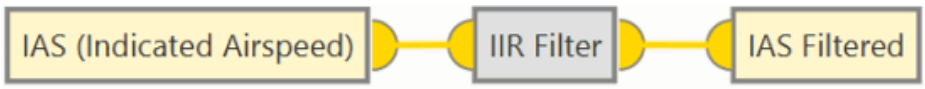

- **Control mix = 0** ⇒ indicates that the platform should adopt a quadcopter configuration.
- **0 < Control mix < 1** ⇒ indicates that the platform will operate in multicopter mode and in FW mode equally.
- **Control mix = 1** ⇒ indicates that the platform should adopt a FW configuration.

The operation of the program can be divided into two stages. A first stage in which, depending on the IAS, a value is assigned to the Control mix variable and a second stage in which, depending on the value of the Control mix variable, the flight configuration is determined.

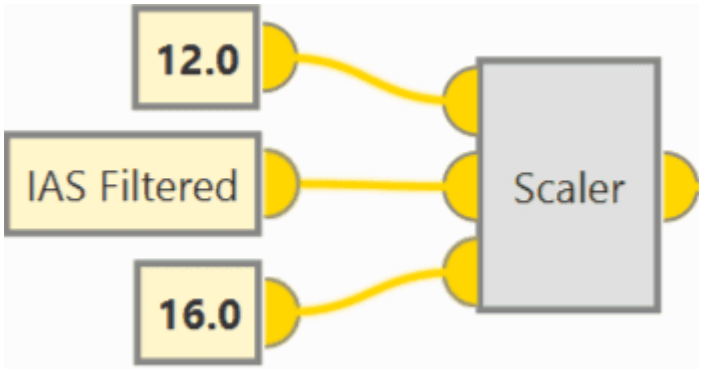
1. **Value of the variable**



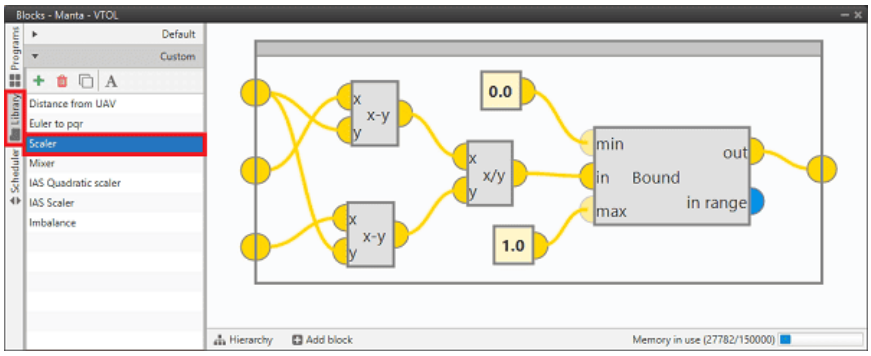

   - To evaluate the IAS variable in a controlled way, the IIR Filter block is applied to it, which returns the IAS Filtered variable.




- The IAS Filtered variable is evaluated in the **Scaler** block. This block returns a value between 0 and 1, with respect to two limit values.

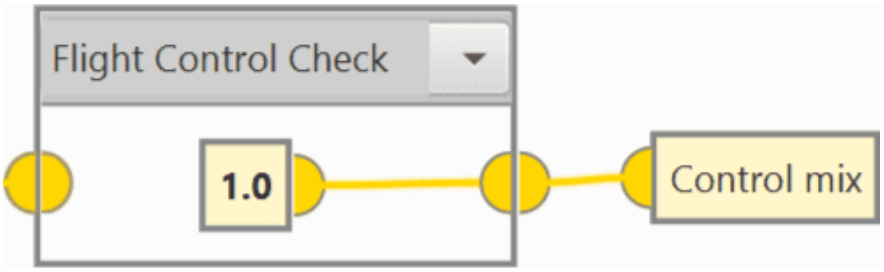



The **Scaler** block is a **custom** block located in the **Library**.

> **Scaler Block - Explanation**
>
> **Scaler** block rates the input value with respect to the range defined by the limit values (12 to 16).
>
> 
> 
>
> **Scaler - Program**
>
> Bound block returns a maximum value **1** and a minimum value **0**.

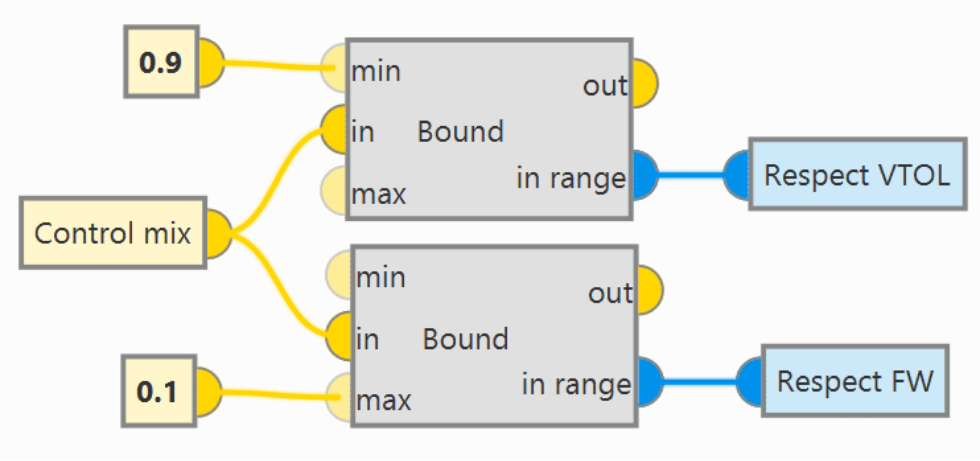
- Depending on the flight phase, **Flight Control Check** or **Standby**, the Control mix variable is assigned.
  - **Standby**: In this flight phase the Control mix variable is the output value of the **Scaler** block.

- **Flight Control Check**: In this flight phase, the Control mix variable has value **1.0** since it is necessary for the aircraft to act in FW configuration.




1. **Flight configuration**

   This part of the program is used to keep stable the PID controllers that allow the control of the aircraft. This control is performed with the value of the Control mix variable.



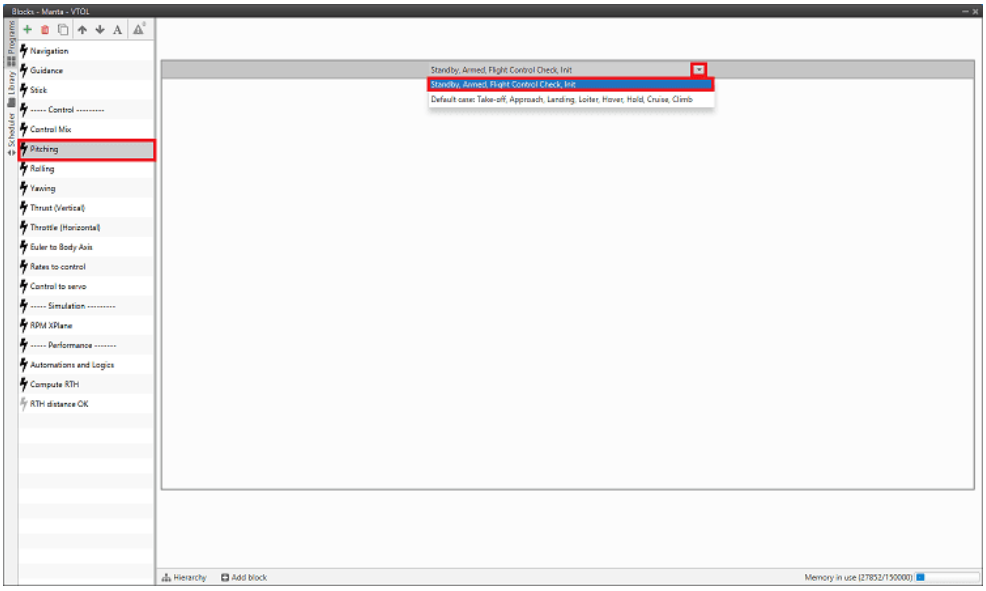

   - If **Control mix > 0.9**, Respect VTOL bit is true and the PID quadcopter controller is stabilized.
   - If **Control mix < 0.1**, Respect FW bit is true and the PID FW controller is stabilized.

Pitching

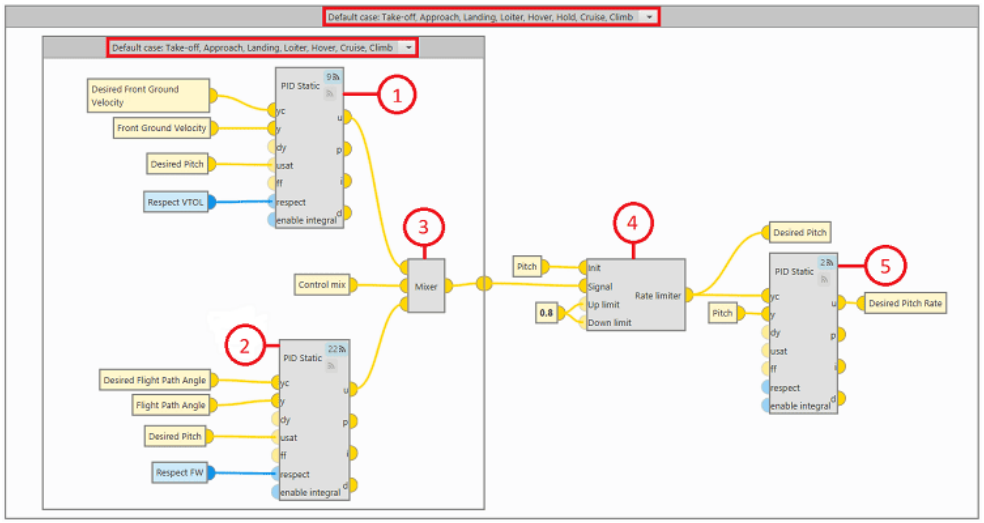
The following program is used to define the behavior and control of the aircraft to perform **Pitch**. The design of the program is made according to the flight phase, dividing the program into three groups of phases where the design of the control laws are similar.

- **Standby, Armed, Flight Control Check, Init**

In these flight phases, the aircraft does not require any design laws for pitch control since it is not flying.

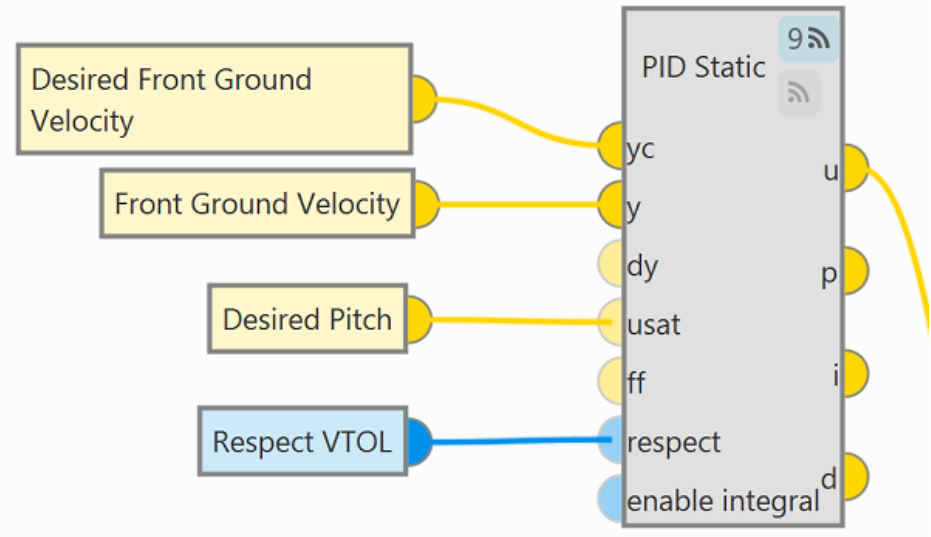



- **Take-off, Approach, Landing, Loiter, Hover, Cruise, Climb**
  Pitch control will depend on the Control mix variable, since the aircraft flight configuration is essential in controlling the aircraft.




The program is basically controlled by 3 PID controllers, which provide the speed ratio as a function of the input variable and the desired variable.

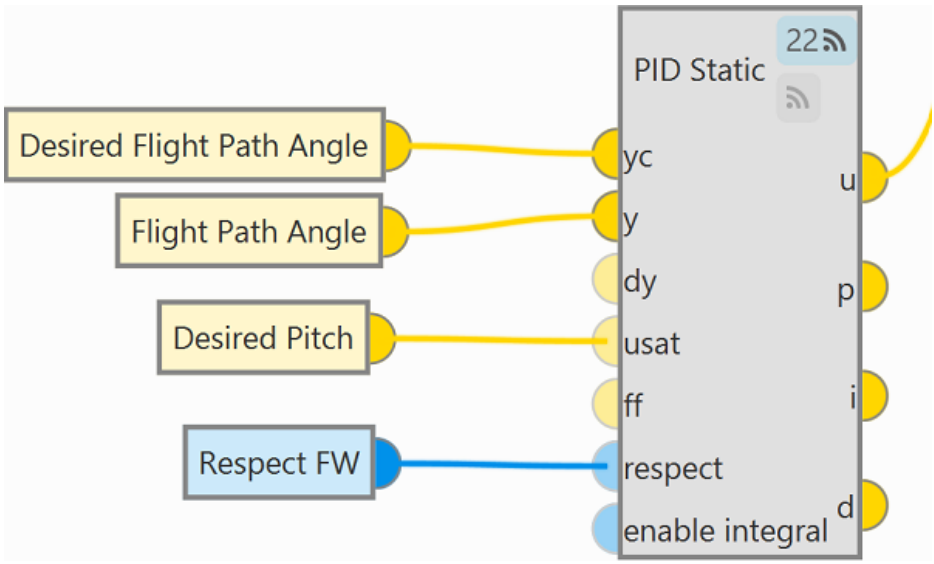
1. The PID controller is defined for **quadcopter** flight configuration.
   In the **quadcopter** configuration, Pitch is performed to be able to control the **Front Ground Velocity**, so this velocity will be the input variable in the PID controller.




> **Important**
>
> When the **Respect VTOL** bit (set in the Control Mix program) is true, the **u** output of the PID block is equal to the **usat** input. That is, when the bit is true, the output of the PID block will be the output variable of the Rate limiter block, Desired Pitch.

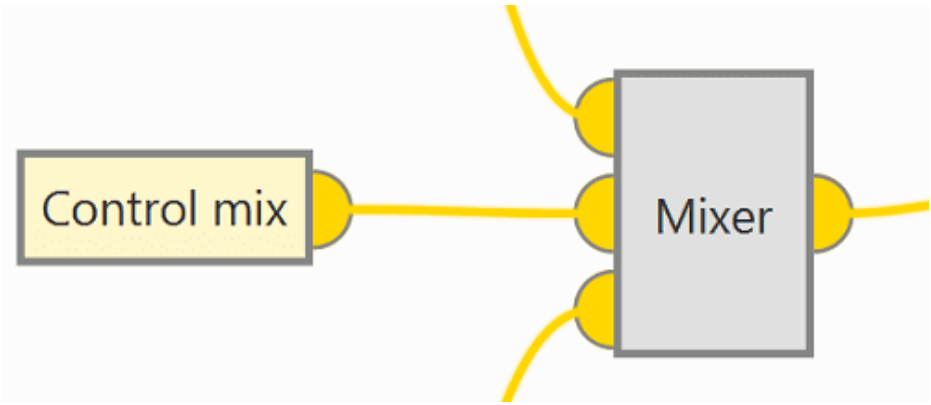
2. The PID controller is defined for the **FW** flight configuration.

   In **FW** configuration, Pitch is performed to control the **Flight Path Angle**, so this variable will be the input variable in the PID controller.

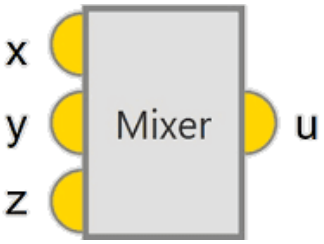


> **Important**
> When the **Respect FW** bit (set in the Control Mix program) is true, the **u** output of the PID block is equal to the **usat** input. That is, when the bit is true, the output of the PID block will be the output variable of the Rate limiter block, Desired Pitch.

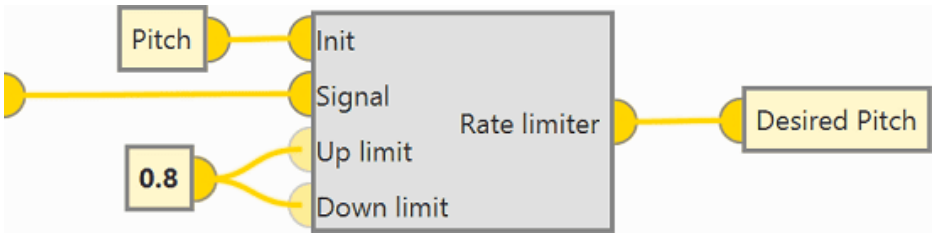
3. **Mixer** block provides the output signals of the above PID controllers depending on the Control mix variable.



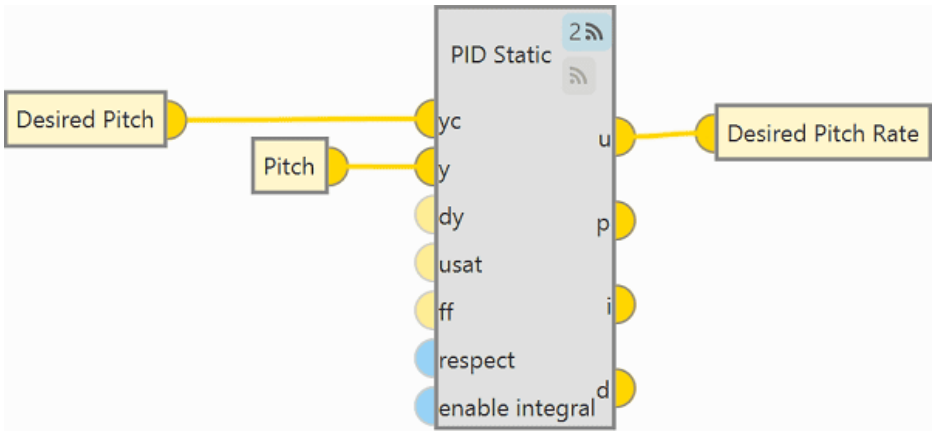

**Mixer** block is a **custom** program located in the **Library**.

> **Mixer Block - Explanation**
> **Mixer** block implements the following relationship between input variables and output variables.
>
> 
> 
>
> $$u = zy + x(1-y)$$
>
> For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

4. The Rate limiter block is defined to adjust the input signal to a **controlled gain**.
   This block limits the rate of change of the variable by limiting the rate of rise and fall of the variable. The purpose is to achieve a controlled rate of change that allows safe operation of the aircraft.

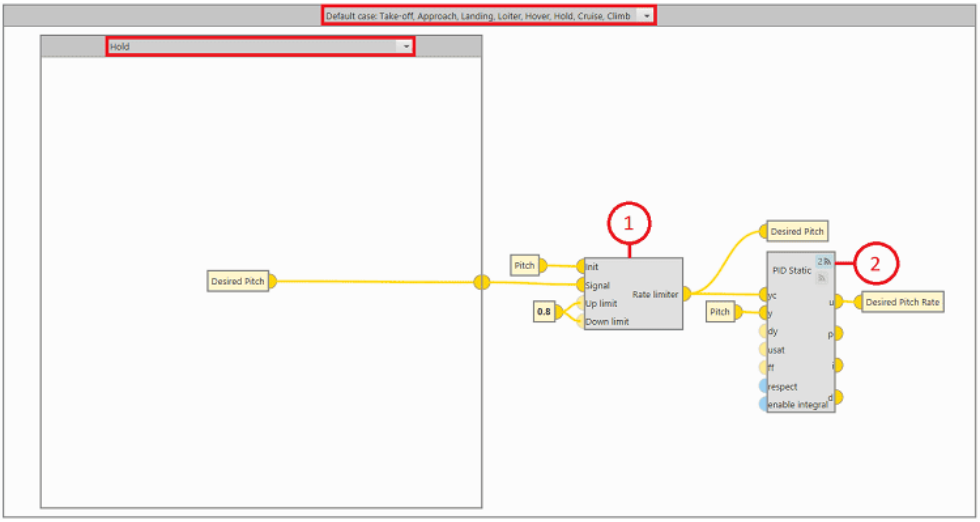



5. A PID controller is defined giving the **Desired Pitch Rate**.



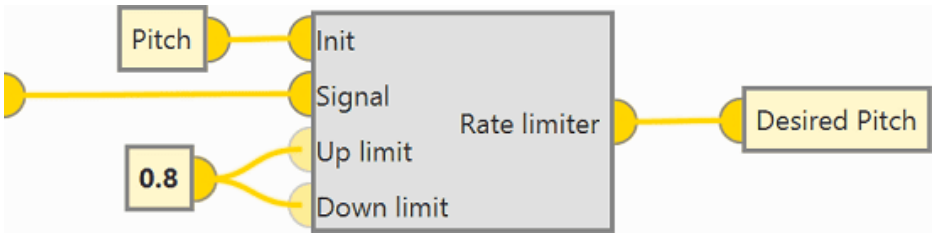

- **Hold**

  In this flight phase, the output of the PID controller is the **Desired Pitch Rate**. Since this flight phase is intended to keep the aircraft **attitude constant**, no PID controller is defined to provide a rate of change.



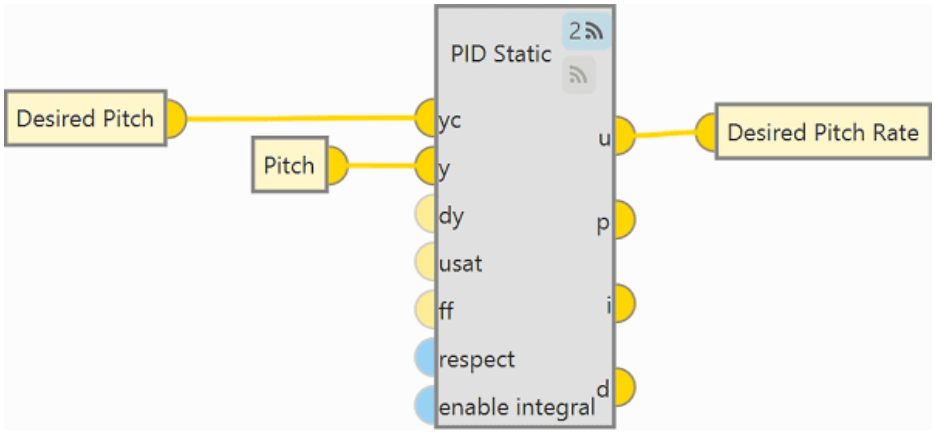

1. The Rate limiter block is defined to adjust the input signal to a controlled gain.

   This block limits the rate of change of the variable by limiting the rate of rise and fall of the variable. The objective is to achieve a controlled rate of change that allows safe operation of the aircraft.




The Signal input variable of the Rate limiter block, Desired Pitch, is generated by the Guidance program. In the Hold flight phase it is intended to stabilize the aircraft attitude, that is why the Desired Pitch control is independent of the flight configuration.

2. A PID controller is defined giving the **Desired Pitch Rate** variable.



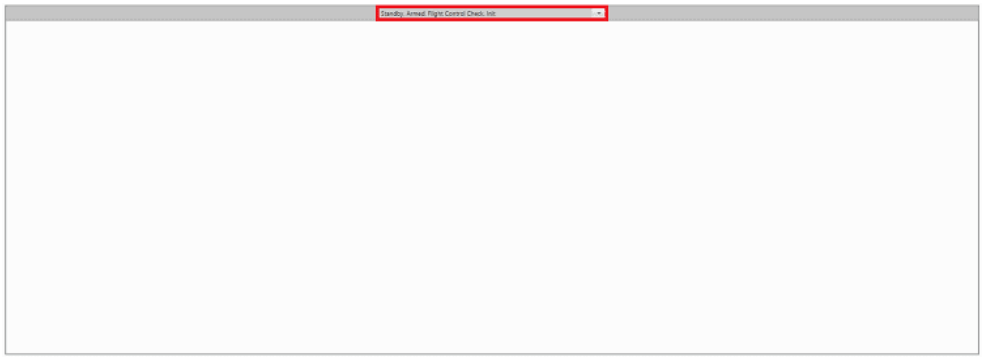

Rolling

The following program is used to define the behavior and control of the aircraft to perform **Roll**. The logic of the program design is similar to the one used in the program to control the Pitch, it is done according to the flight phase in which the aircraft is, dividing the program into four groups of phases where the design of the control laws are similar.

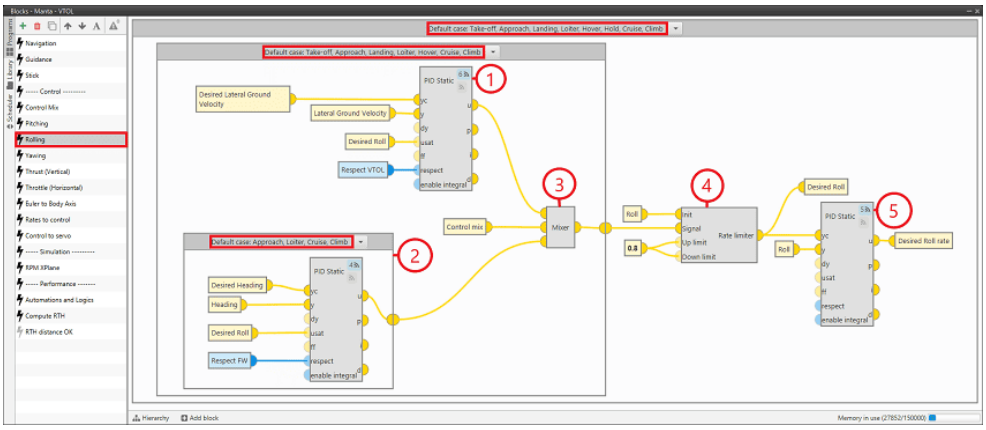
> **Important**
>
> The effect on the aircraft when rolling depends on the flight configuration in which it is flying.
>
> If rolling is performed flying in **quadcopter configuration**, the effect on the aircraft is the variation of the **Lateral Ground Velocity** and if rolling is performed flying in **FW configuration** the effect on the aircraft is the variation of the **Heading**.

- **Standby, Armed, Flight Control Check, Init**

  In these flight phases, the aircraft does not require any design laws for roll control since it is not flying and is therefore at rest with respect to the ground.



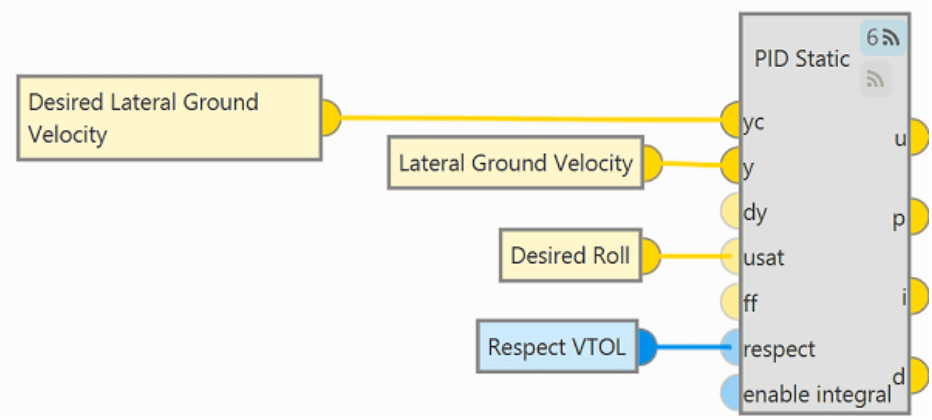
- **Approach, Loiter, Cruise, Climb**




The program is basically controlled by 3 PID controllers, which provide the speed ratio as a function of the input variable and the desired variable.

1. PID controller for **quadcopter** flight configuration.

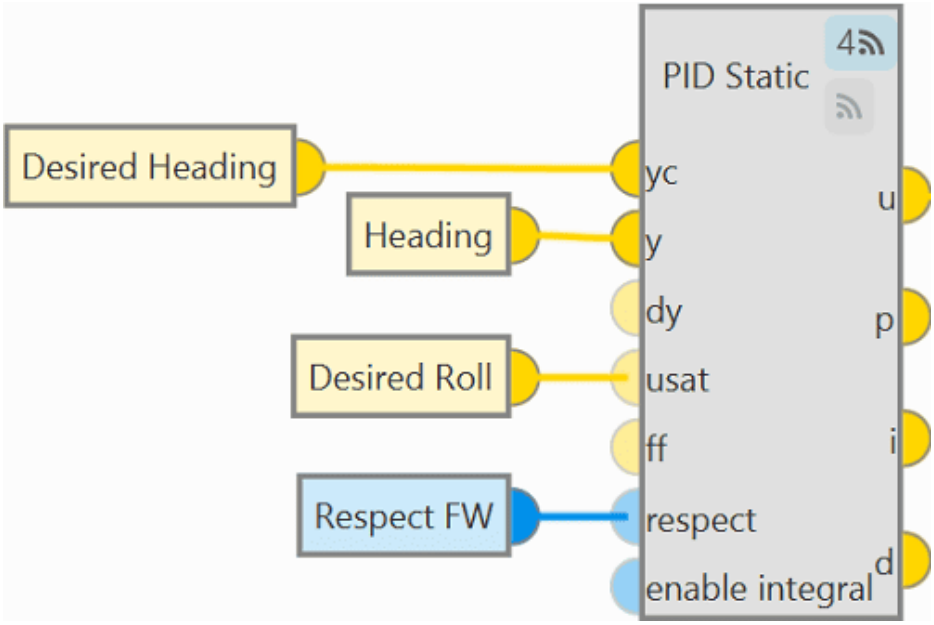
In **quadcopter** configuration, Roll is performed to be able to control the **Lateral Ground Velocity**, so this velocity will be the input variable in the PID controller.




> **Important**
>
> When the **Respect VTOL** bit (set in the Control Mix program) is true, the **u** output of the PID block is equal to the **usat** input. That is, when the bit is true, the output of the PID block will be the output variable of the Rate limiter block, Desired Roll.

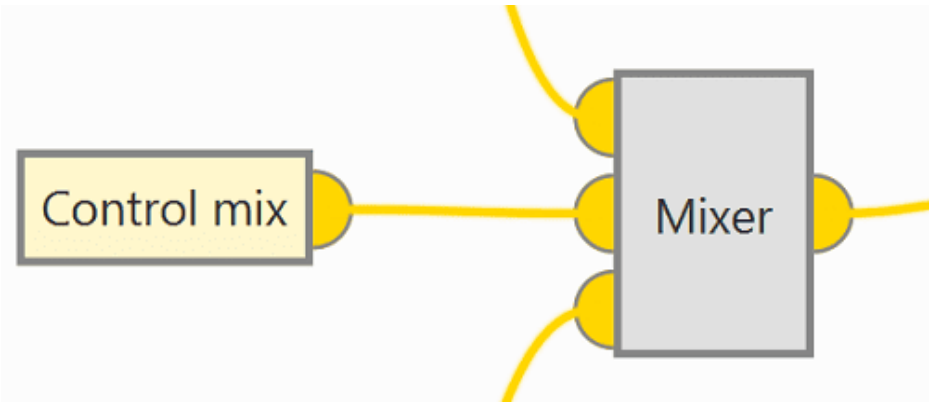
2. PID controller for **FW** flight configuration.

   In **FW** configuration, Roll is performed to control the **Heading**, so this variable will be the input variable in the PID controller.

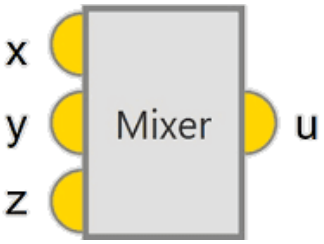


> **Important**
> When the **Respect FW** bit (set in the Control Mix program) is true, the **u** output of the PID block is equal to the **usat** input. That is, when the bit is true, the output of the PID block will be the output variable of the Rate limiter block, Desired Roll.

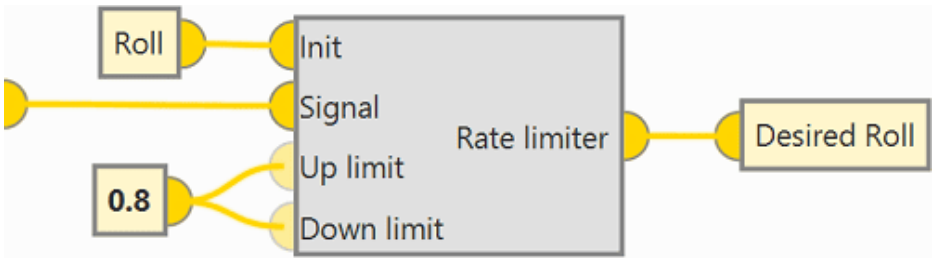
3. **Mixer** block provides the output signals of the above PID controllers depending on the Control mix variable.




**Mixer** block is a **custom** program located in the **Library**.

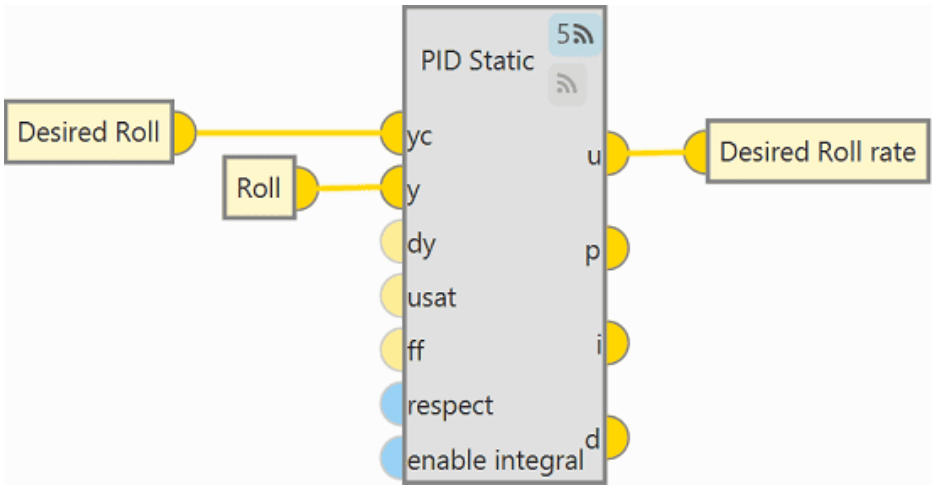
> **Mixer Block - Explanation**
> **Mixer** block implements the following relationship between input variables and output variables.
>
> 
> 
>
> $$u = zy + x(1-y)$$
>
> For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

4. Rate limiter block is defined to adjust the input signal to a **controlled gain**.
   It limits the rate of change of the variable controlled by the PID by limiting the rate of rise and fall of the variable. The purpose is to achieve a controlled rate of change that allows safe operation of the aircraft.

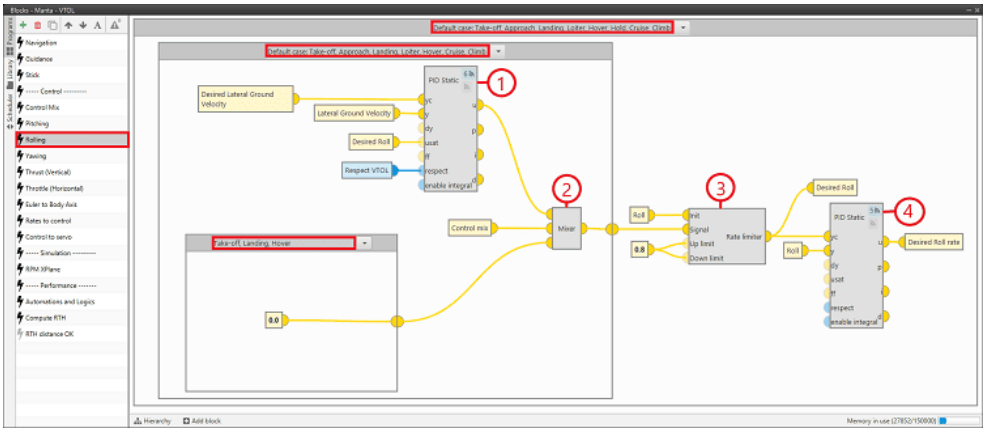



5. PID controller giving the **Desired Roll rate**.

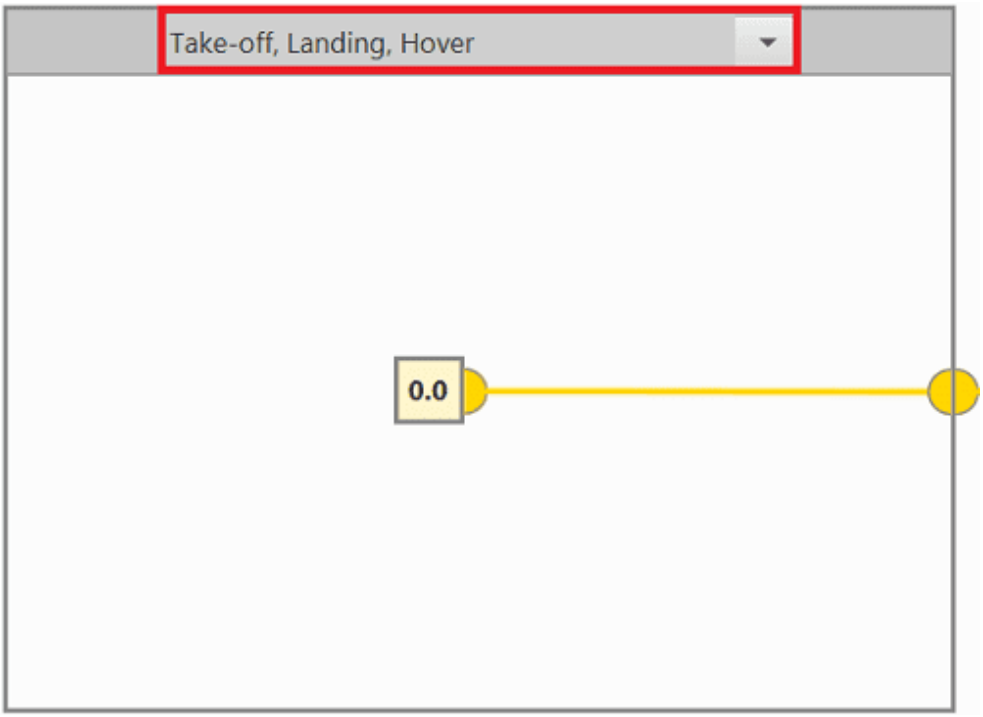
   This controller will provide the rate of change of the **Roll** variable.




- **Take-off, Landing, Hover**

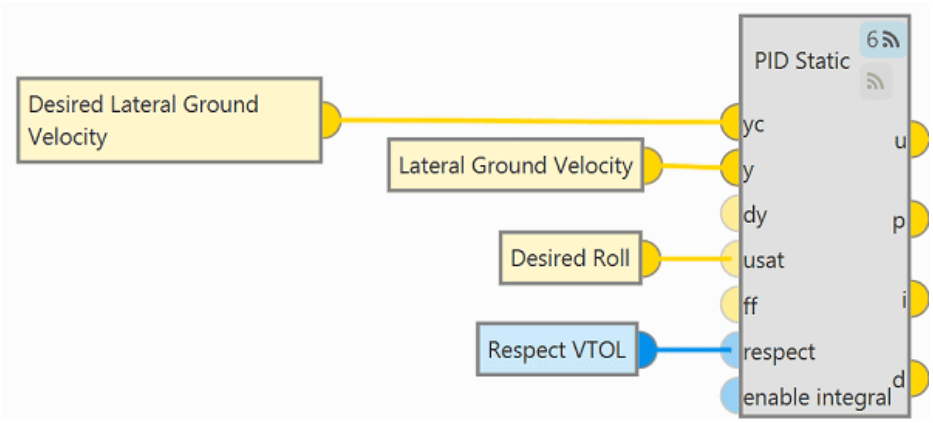


In these flight phases the **FW** flight configuration is not taken into account, since **these flight phases must be performed in quadcopter configuration**. For this reason, **no** PID controller is defined to control the **Heading**.



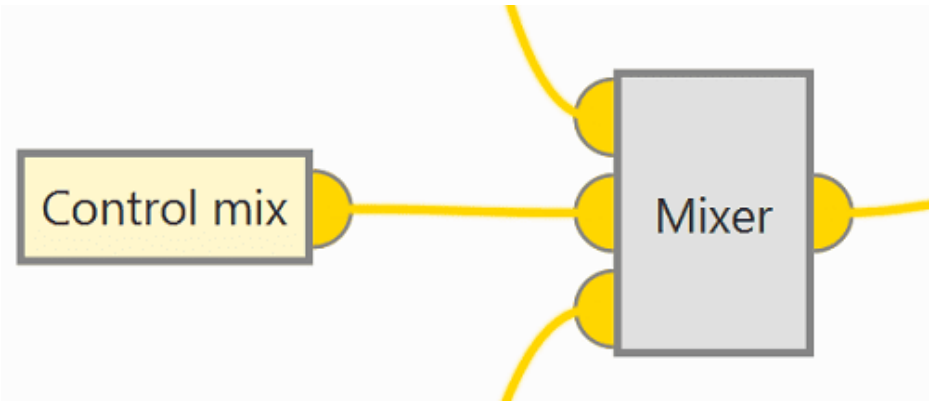

1. PID controller for **quadcopter** flight configuration.

   This controller will provide the rate of change of the Lateral Ground Velocity variable.



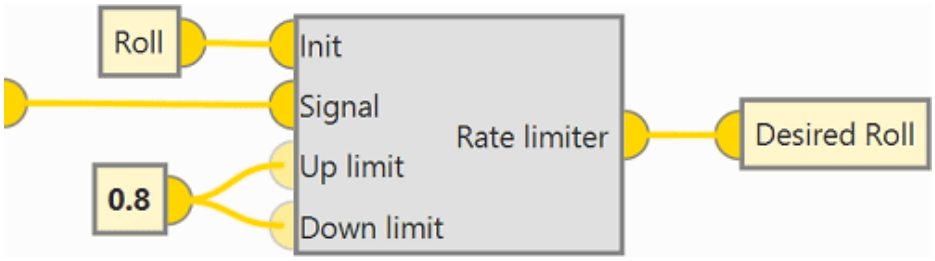

> **Important**
>
> When the **Respect VTOL** bit (set in the Control Mix program) is true, the **u** output of the PID block is equal to the **usat** input. That is, when the bit is true, the output of the PID block will be the output variable of the Rate limiter block, Desired Roll.

2. **Mixer** block provides the output signal of the above PID controller depending on the Control mix variable.



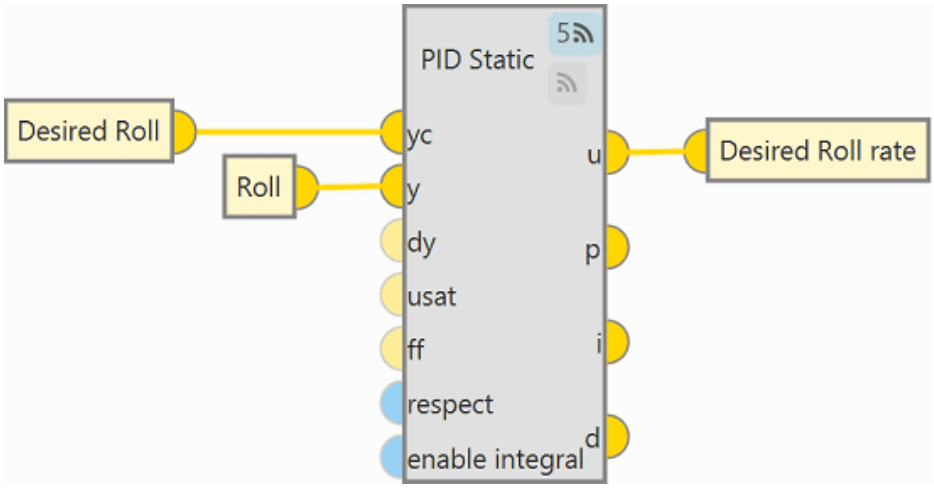

3. Rate limiter block is defined to adjust the input signal to a **controlled gain**.

   It limits the rate of change of the variable controlled by the PID by limiting the rate of rise and fall of the variable. The purpose is to achieve a controlled rate of change that allows safe operation of the aircraft.

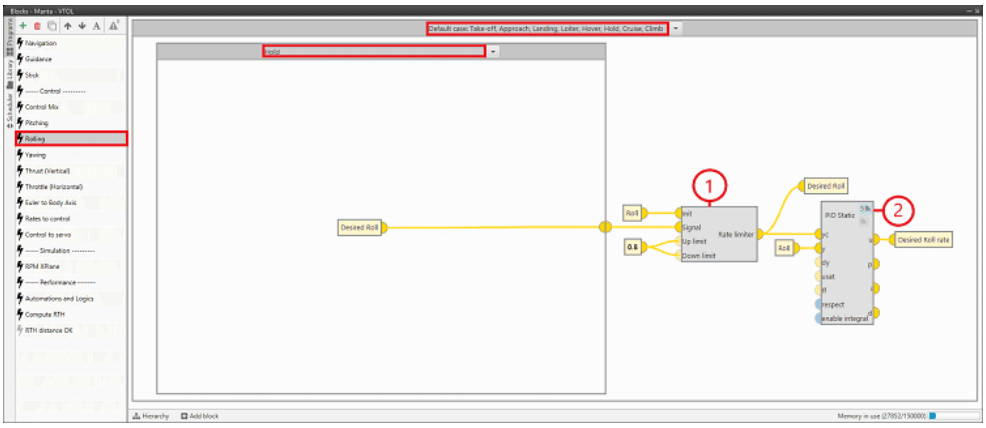



4. PID controller giving the control variable **Desired Roll rate**.

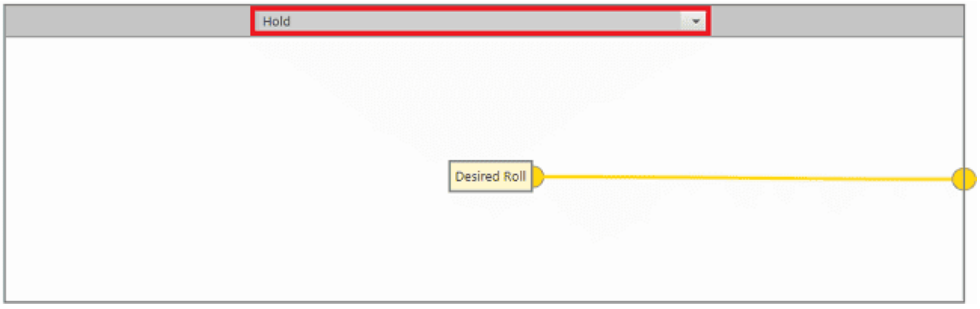
   This controller will provide the rate of change of the **Roll** variable.




- **Hold**

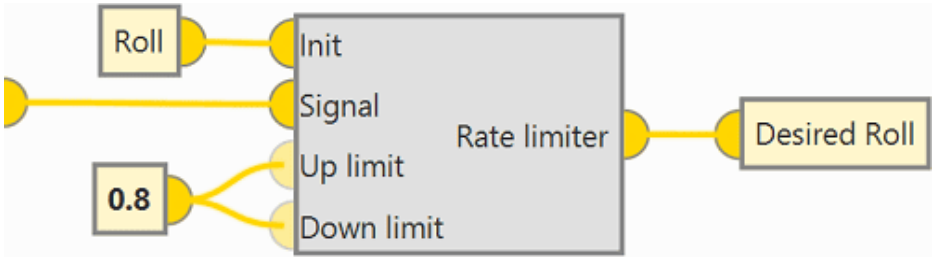



In this flight phase it is **not necessary** to define the **Roll control**, since it is intended to keep a **constant attitude** of the aircraft, so the value of the **Desired Roll** variable will come directly from the Guidance program.



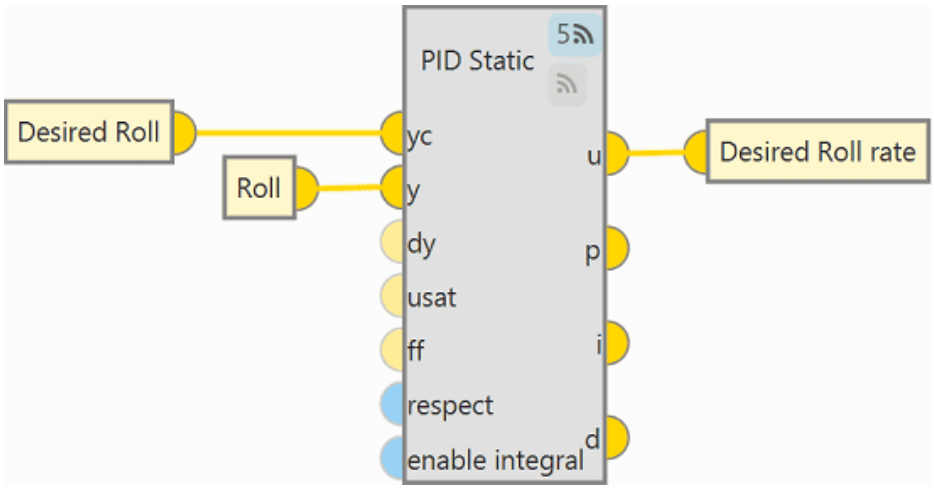

1. Rate limiter block is defined to adjust the input signal to a **controlled gain**.

   It limits the rate of change of the variable controlled by the PID by limiting the rate of rise and fall of the variable. The objective is to achieve a controlled rate of change that allows safe operation of the aircraft.

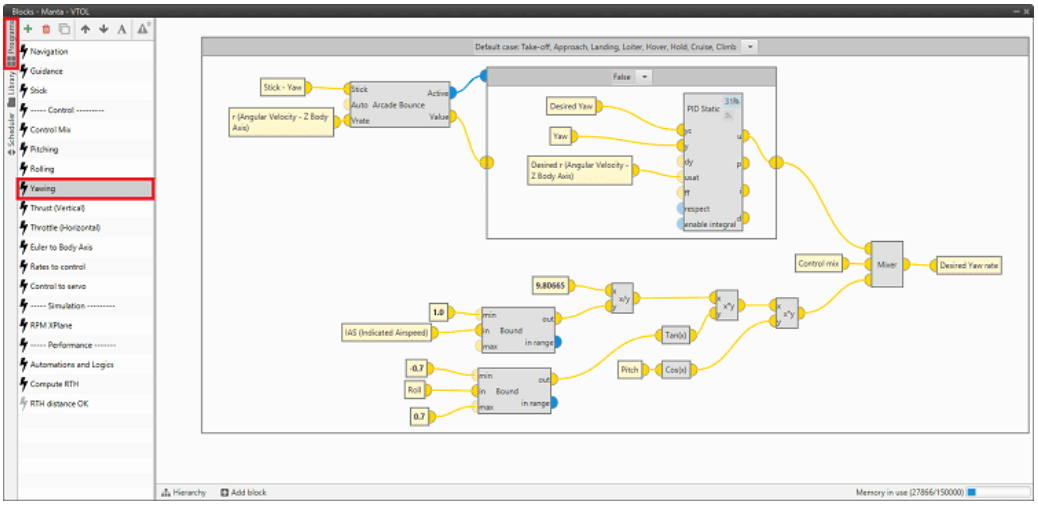
2. PID controller giving the control variable **Desired Roll rate**.

   This controller will provide the rate of change of the **Roll** variable.



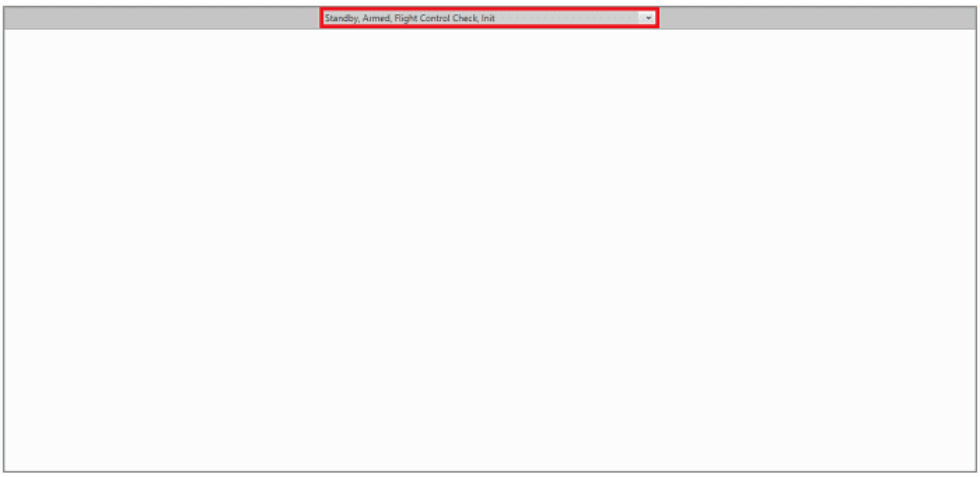

## Yawing

As with the Pitching and Rolling programs, the program to control the **Yaw** is performed depending on the phase of flight in which it is flying.

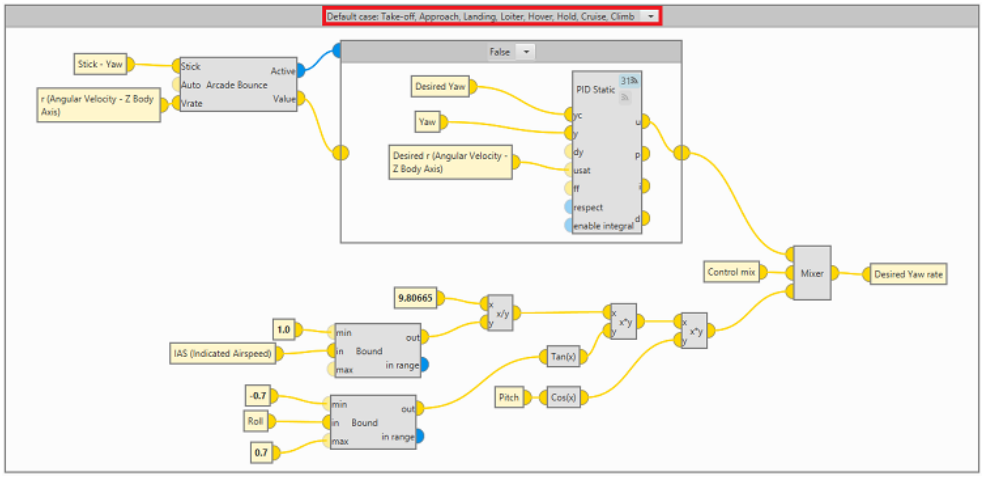


- **Standby, Armed, Flight Control Check, Init**

  For these flight phases no definition of the yaw control program is required as they are phases where the aircraft is not flying.

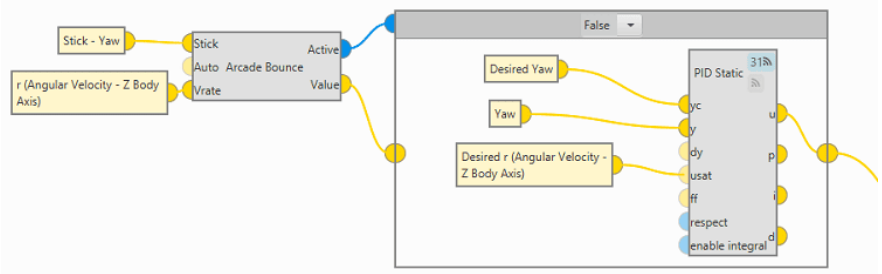


- **Take-off, Approach, Landing, Loiter, Hover, Hold, Cruise, Climb**
  In these flight phases, **Yaw** control is performed depending on the aircraft flight configuration, quadcopter or FW. For this reason, it is necessary to define two control paths, one to control the Yaw when the aircraft flies at **low speeds (Multi)** and another one when the aircraft flies at **high speeds** and, therefore transitions to the **Fixed Wing** (FW) flight configuration.



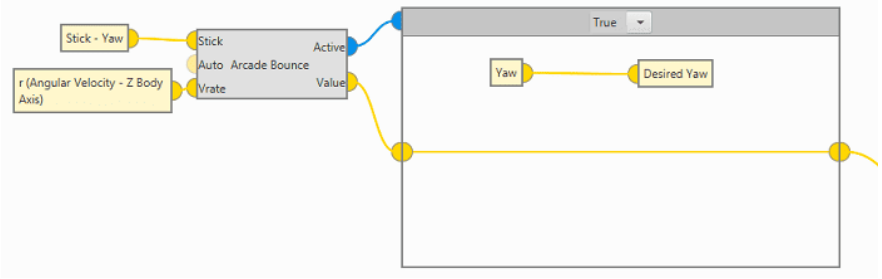

The program logic has been divided into three sections/parts. Two of them to define the aircraft control according to the flight configuration and, once these controls are defined, in the third section of this program, the output signals of both are provided to the Control mix variable by means of the **Mixer** block.

1. **Quadcopter control**
   - **False**
     In this case, Autopilot 1x is controlled in **Auto** mode, so the Arcade Bounce block will return a **false** bit. Therefore, **Yaw** control is performed with a PID controller.



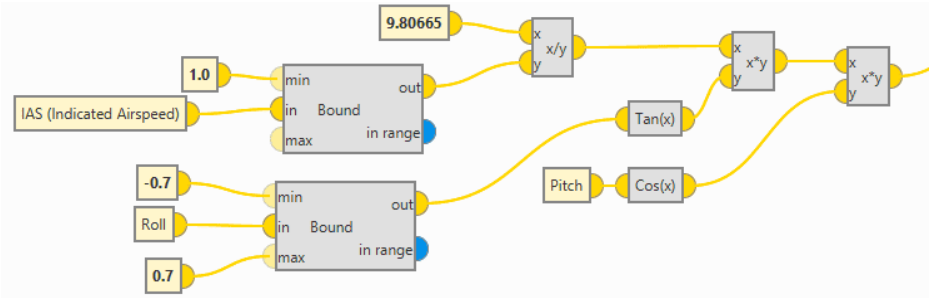

   - **True**

In this case, Autopilot 1x takes into account the Stick input in the Arcade Bounde block and returns a **true** bit. Therefore, the value of the **Desired Yaw rate** is the one commanded by the stick entry.




2. **FW control**

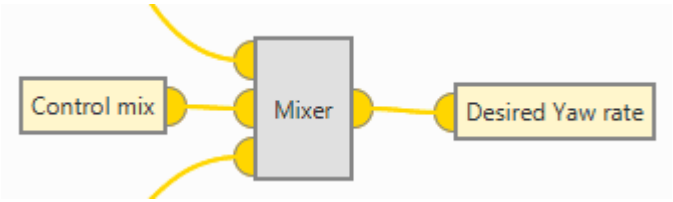
   In FW configuration, the turns do not depend only on the yaw as in quadcopter configuration, but also on the roll and pitch to perform a coordinated turn, hereafter called "Coordinate Turn". The following algorithm represents this **Coordinate Turn** performed by the aircraft:




   Bound block limits the **IAS** and **Roll** values. The minimum value for the **IAS** is $1.0\ m/s$, and the value for the **Roll** angle should be between $[-0.7,\ 0.7]\ rad$.

3. **Mix control**

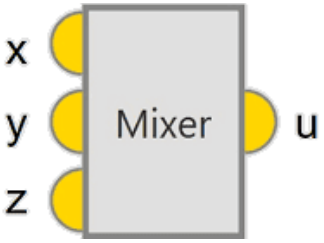
   **Mixer** block has as inputs the output signals of the previous controls and the Control mix variable. In this way, the block selects which configuration controls the aircraft, providing the **Desired Yaw rate** value.




   **Mixer** block is a **custom** program located in the **Library**.

**Mixer Block - Explanation**

**Mixer** block implements the following relationship between input variables and output variables.



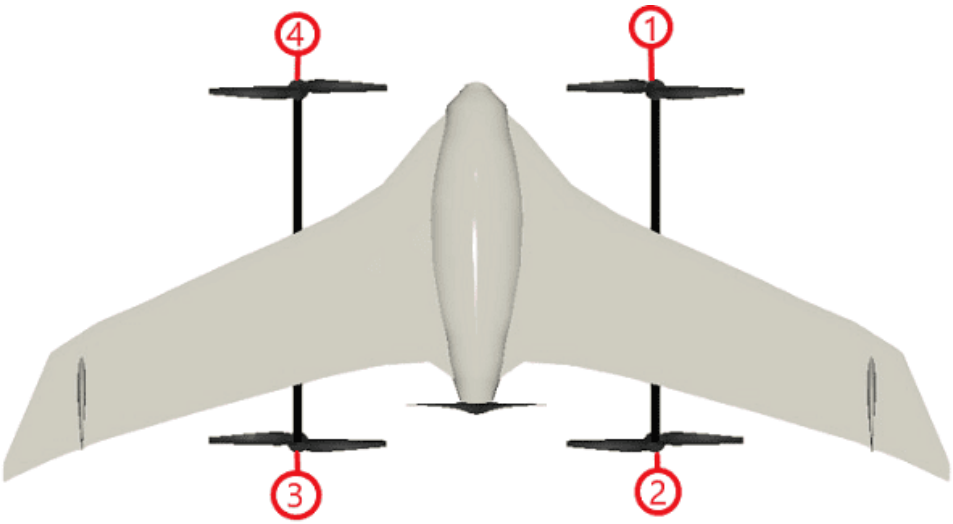

$$u = zy + x(1-y)$$

For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

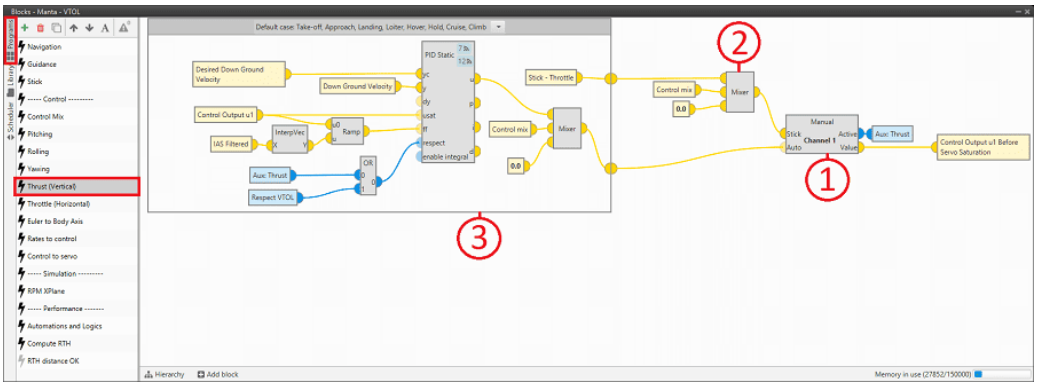
Thrust (Vertical)

**Explanation**

Thrust program is defined to control the **vertical thrust** of the aircraft. For this purpose, the motors involved are the vertical ones, which allow the aircraft to fly in **quadcopter** configuration.

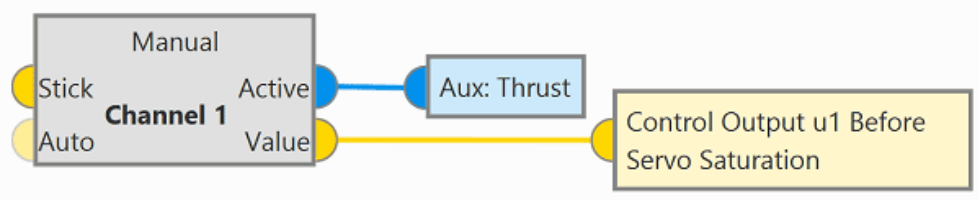



Thrust actuation is performed with the use of **Channel 1**. This channel is controlled with the Manual block, which is configured to process thrust control depending on the control mode. This block switches between two input signals according to the current mode of the configured channel.



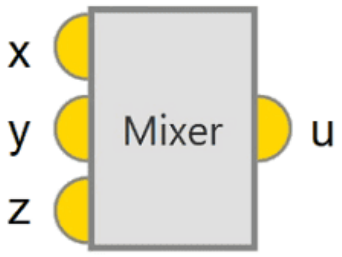

**Thrust (Vertical) - Program**

1. The **Manual** block receives two input signals, Stick and Auto and provides two output signals, Value, which depends on the flight mode, and Active.

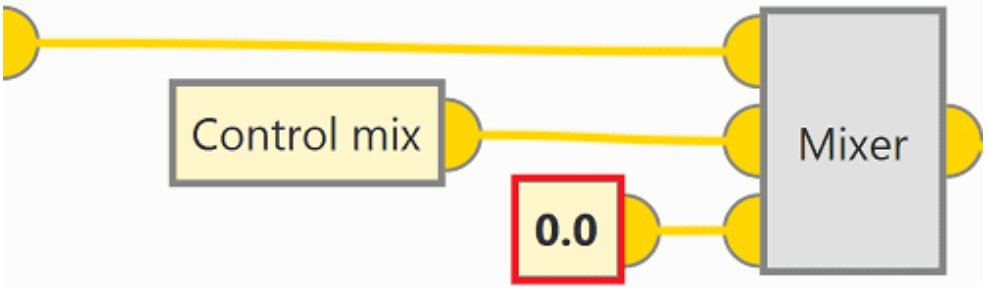



   The Value output of this block must be linked to the Control Output u1 Before Servo Saturation variable, since this variable controls the actuation of the Thrust defined in the Actuator block.
   Moreover, the Aux: Thrust bit is **true** when the Autopilot 1x is **not controlled by the stick**.

2. The **Mixer** block provides the value for the Stick input to the **Manual** block as a function of the Control mix variable. This is a **custom** block located in the **Library**.

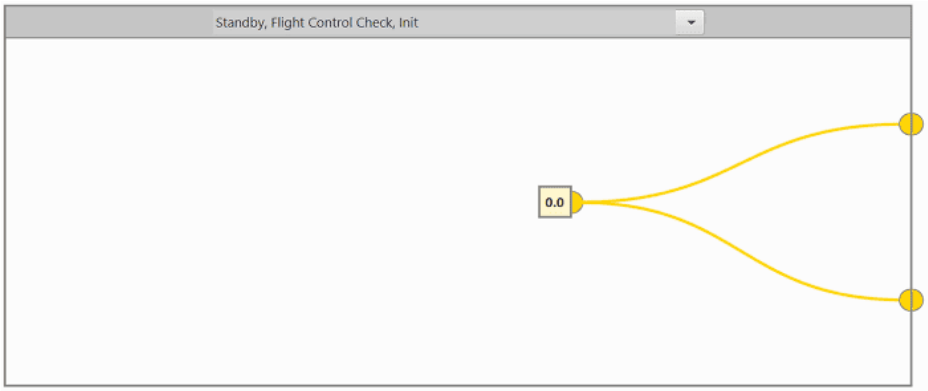
   > **Mixer Block - Explanation**
   > **Mixer** block implements the following relationship between input variables and output variables:
   >
   > 
   > 
   >
   > $$u = zy + x(1-y)$$
   >
   > For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

As the Control mix variable indicates the control rate of the aircraft in FW flight configuration, only the part proportional to the rate controlled in quadcopter configuration is maintained.




3. The input variable Auto to the **Manual** block and the input variable to the **Mixer** block, which corresponds to the **quadcopter** configuration, depends on the flight phase:
   - **Standby, Flight Control Check, Init**

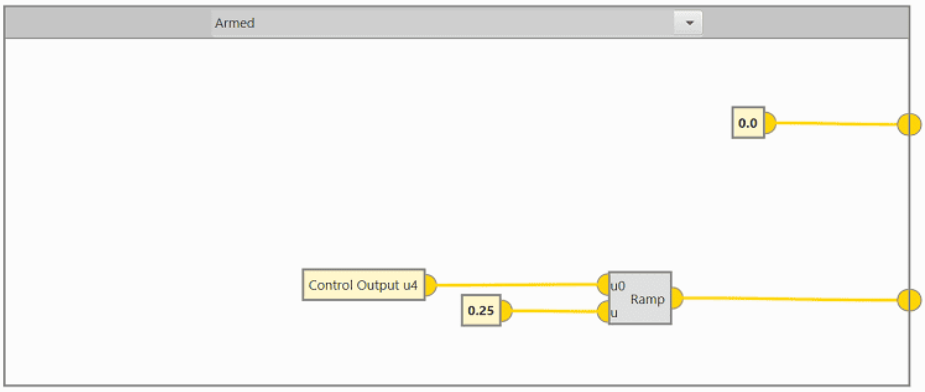
     Since no Thrust action is required in these flight phases, the values for the Stick and Auto inputs are set to **0.0**. Therefore, the output value of the Manual block is null. This is because the Control Output u1 Before Servo Saturation variable must be null regardless of the flight control mode.

   - **Armed**

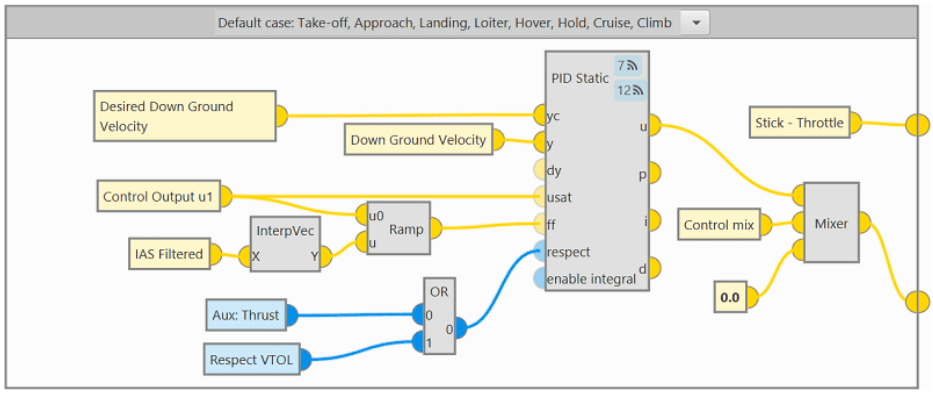
     In the Armed flight phase, the Autopilot 1x automatically activates the aircraft's four motors up to **25%** of their **maximum RPM**.

     > ⓘ **Note**
     > This RPM value is not sufficient to lift the aircraft, it only activates the motors.




- **Stick input**: This value remains zero since in this flight phase the **aircraft** is **only controlled in auto mode**.
- **Auto input**: Corresponds to the output of the Ramp block.
  This block is used to activate the motors at 25% of their maximum RPM, as it allows to control the rate of change of the Control Output u4 variable.

◦ **Take-off, Approach, Landing, Loiter, Hover, Hold, Cruise, Climb**

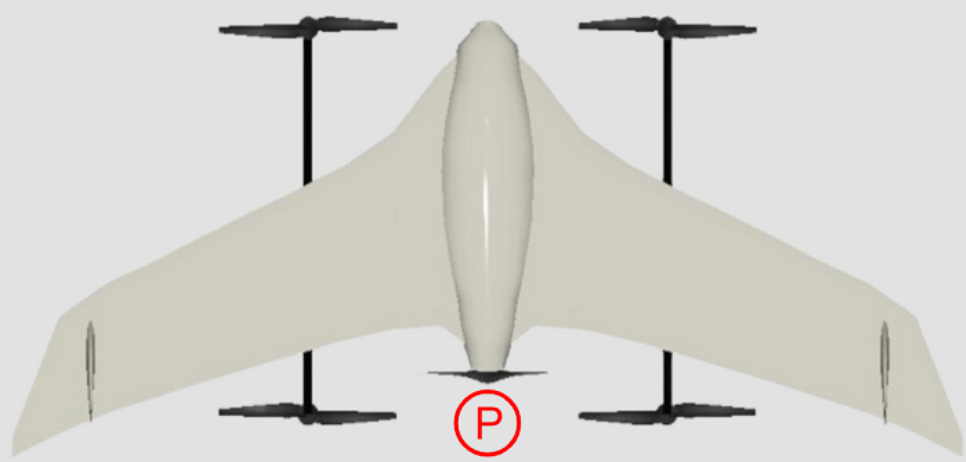



- **Stick input**: This value is defined as the part of the **Stick-Throttle** variable that is controlled by the quadcopter.
- **Auto input**: The Auto input is written by the **Mixer** block. This block receives 3 input signals, 2 of them correspond to the Thrust control in **quadcopter (1)** and **FW (2)** flight configurations, and the value of the third input is the variable Control mix written by the Control Mix program.
  1. The control signal in the **quadcopter flight configuration** is defined with a PID controller.
  2. The control signal in the **FW flight configuration** is **zero** since no Thrust control is performed in this flight configuration.

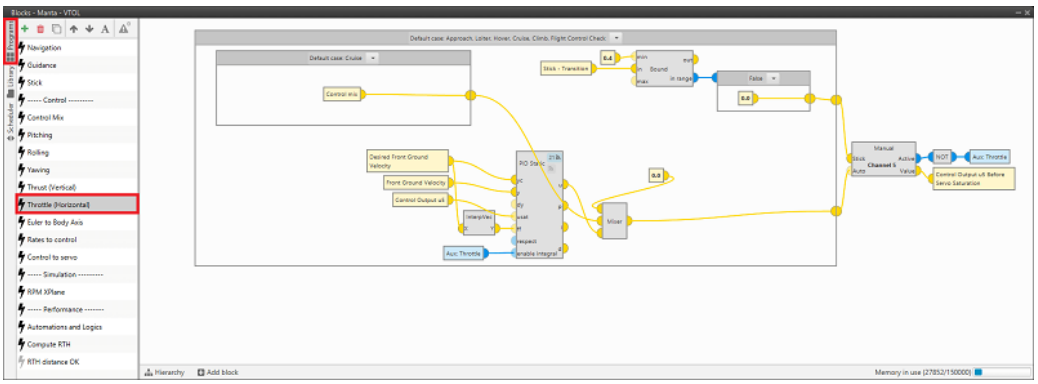
Throttle (Horizontal)

### Explanation

Throttle program is defined to control the **horizontal thrust** of the aircraft. For this purpose, pusher motor is involved, which allow the aircraft to fly in **FW** configuration.

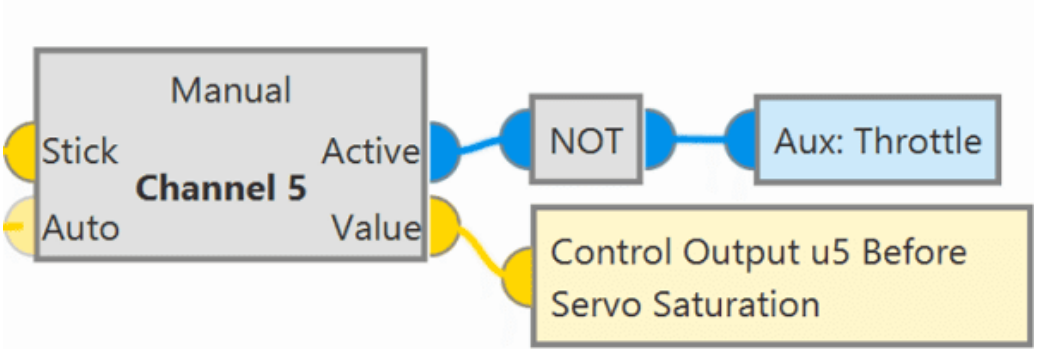



Throttle actuation is performed with the use of **Channel 5**. This channel is controlled with the Manual block, which is configured to process throttle control depending on the control mode. This block switches between two input signals according to the current mode of the configured channel.




**Throttle (Horizontal) - Program**

The **Manual** block receives two input signals, Stick and Auto and provides two output signals, Value, which depends on the flight mode, and Active.

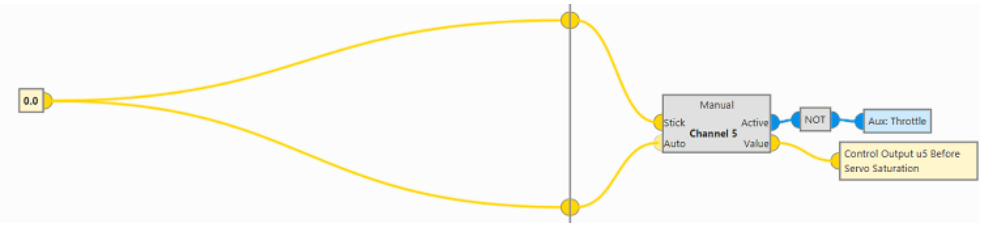



The Value output of this block must be linked to the Control Output u5 Before Servo Saturation variable, since this variable controls the actuation of the Throttle defined in the Actuator block.

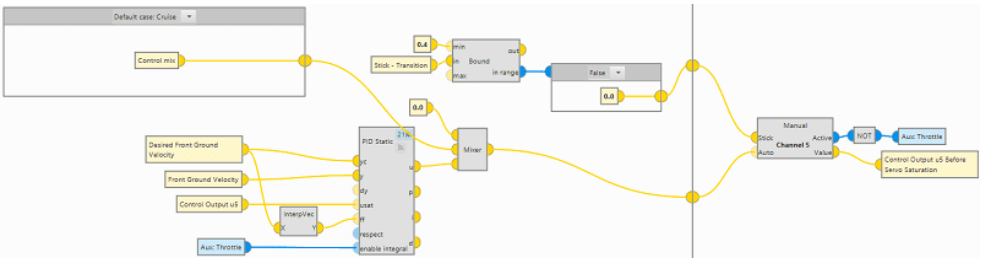
Moreover, the Aux: Throttle bit is **true** when the Autopilot 1x is **not controlled by the stick**.

The input variables, Stick and Auto, are different depending on the flight phase:

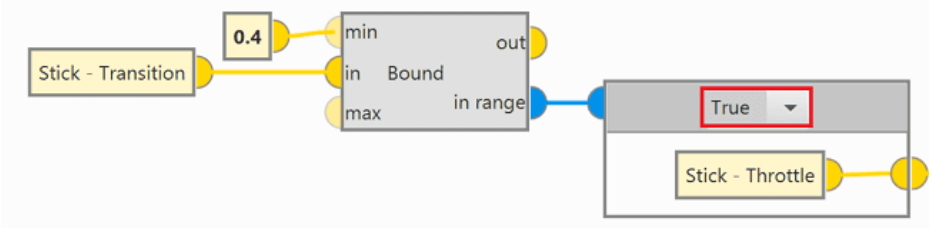
- **Standby, Take-off, Landing, Init**
  Since no Throttle action is required in these flight phases, the values for the Stick and Auto inputs are set to **0.0**. Therefore, the output value of the Manual block is null.



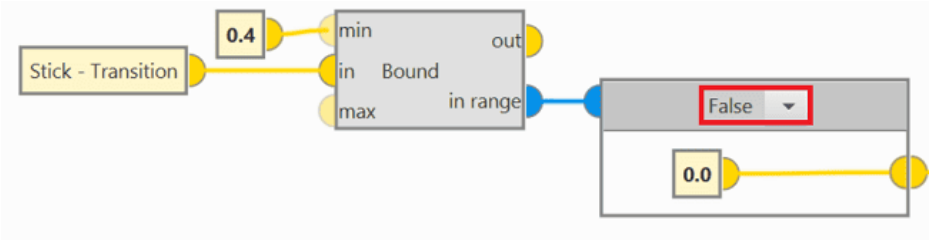

- **Approach, Loiter, Hover, Cruise, Climb, Flight Control Check**



  - **Stick input**
    If the Stick-Transition variable has a value **greater than 0.4** ⇒ in range bit will be **true** and the Stick input of the **Manual** block will be written by the Stick-Throttle variable.




Otherwise, if this variable has a value **less than 0.4** ⇒ in range bit will be **false** and the Stick input will have a value of **0.0**.

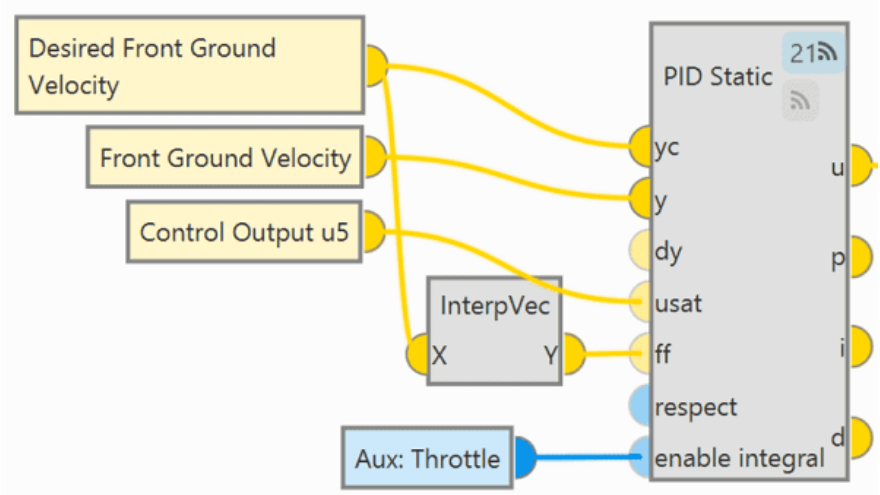



> ⓘ **Note**
>
> The Stick-Transition variable represents the position of the Stick command that defines the aircraft control mode. This command has two values: **1** for control of the aircraft using the **stick pilot**, and **0** for control of the aircraft in **Auto** mode, i.e., with Autopilot 1x.

- **Auto input**

  The Auto input is written by the **Mixer** block. This blocks receives 3 input signals, 2 of them correspond to the Throttle control signals in **quadcopter (1)** and **FW (2)** flight configurations, and the value of the third signal **(3)** will be different depending on the flight phase.

  1. The control signal in the **quadcopter flight configuration** is **zero** since no Throttle control is performed in this flight configuration.
  2. The control signal in the **FW flight configuration** is defined with a PID controller.

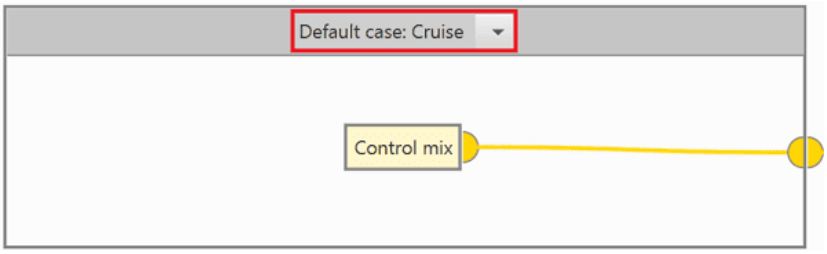



3. The input variable to the **Mixer** block is different in the Cruise and Hold flight phases:

   - **Cruise**

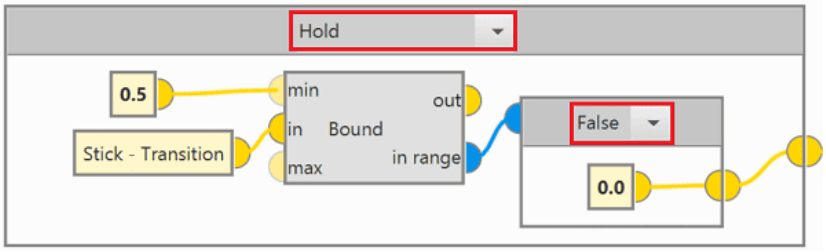
     In this flight phase, the input to the **Mixer** block is simply the variable Control mix written by the Control Mix program.

     > **Important**
     > This also applies to all other flight phases that are not explicitly defined, as this is the "Default case".

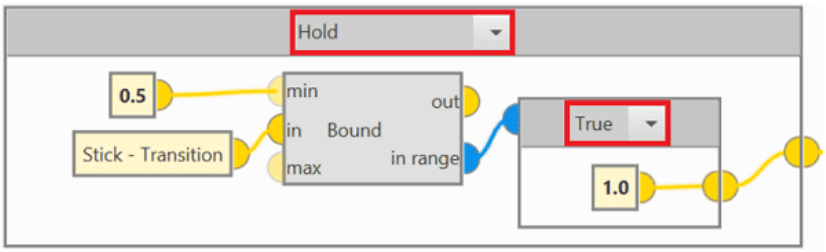
     
     

   - **Hold**

     In this flight phase, if the Stick-Transition variable has a value **less than 0.5** ⇒ Input for Mixer block = **0.0**.

     Otherwise, if the Stick-Transition variable has a value **greater than 0.5** ⇒ Input for Mixer block = **1.0**.

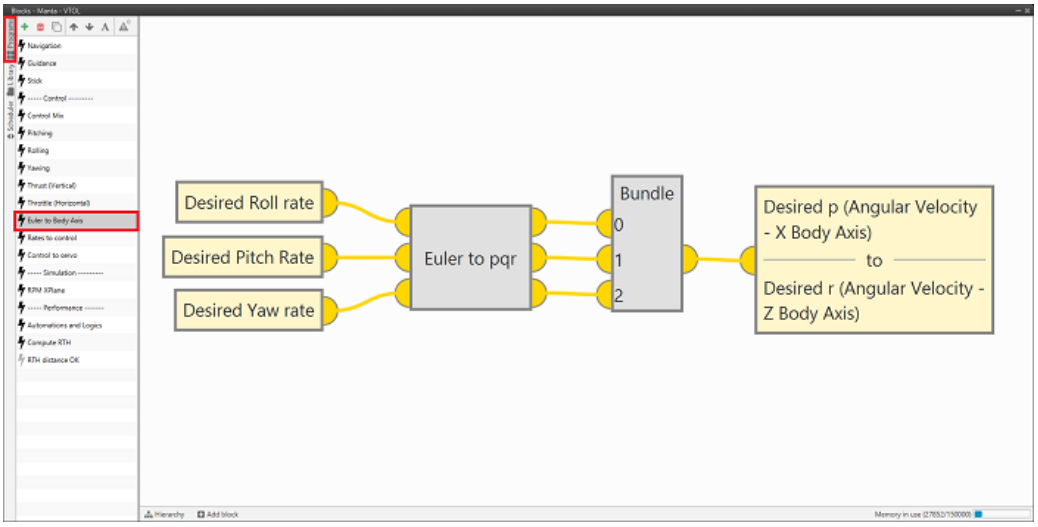



> **Note**
>
> This control logic represents the limits of the Control mix variable, being **0** quadcopter configuration and **1** FW configuration.

Euler to Body Axis

The following program defines the transformation between **Euler** axis and **Body** axis.

The control variables Desired Roll rate, Desired Pitch rate and Desired Yaw rate are expressed in Euler axis, so to transform these variables into control outputs, it is necessary to transform them to the aircraft axis (Body axis).

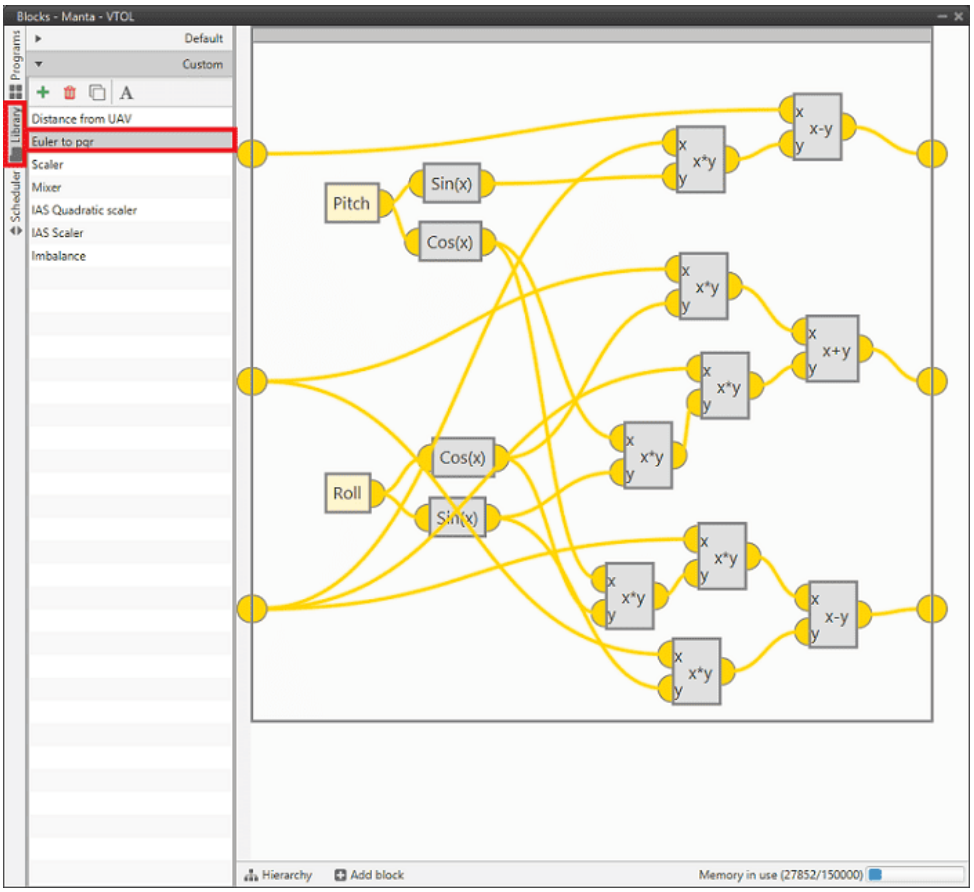



**Euler to Body Axis - Program**

For this purpose, a **custom block**, **Euler to pqr** block, has been created in the **Library blocks**. In this block, the mathematical relationships necessary to achieve this transformation have been defined.

### Euler To Pqr Block - Explanation

The algorithm of this block and the mathematical relationships it implements are shown below:




**Euler to pqr block**

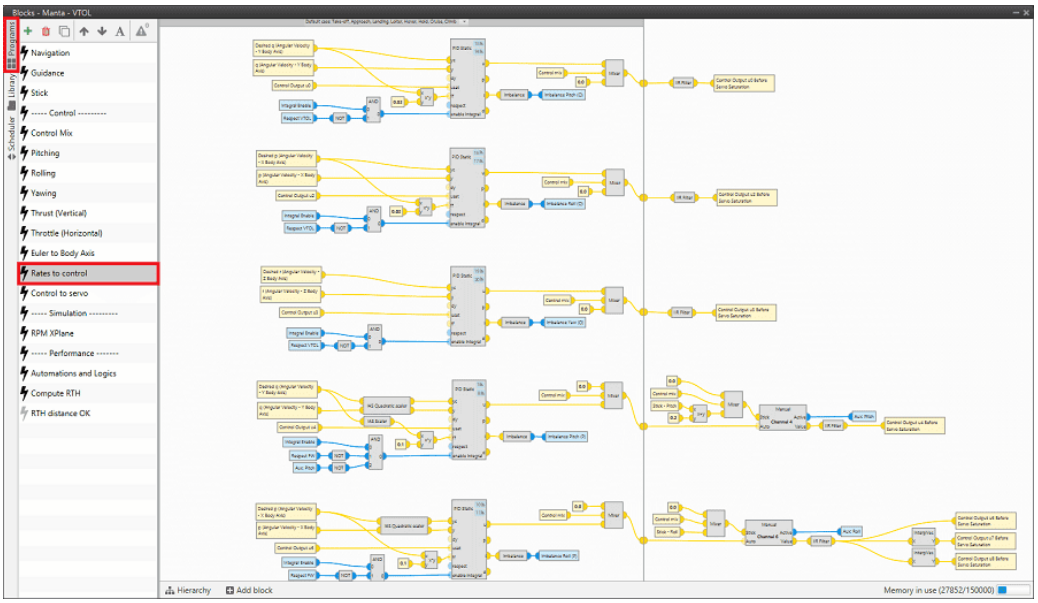
The transformation between axes that is performed is as follows:

$$\begin{bmatrix} P \\ Q \\ R \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin(\theta) \\ 0 & \cos(\phi) & \sin(\phi)\cos(\theta) \\ 0 & -\sin(\phi) & \cos(\phi)\cos(\theta) \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix}$$

For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

## Rates to control

The following program is defined to control the value of the **control outputs** to perform the desired action defined by the Guidance and Navigation program variables.




**Rates to control - Program**

For the definition of this program, it is first necessary to create the following **custom blocks** in the **Library blocks**. For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

### Ias Scaler Block - Explanation

IAS Scaler allows to **scale linearly** an input variable with the indicated airspeed.

This block implements the following formula:

$$u = \frac{v_i}{\max(v,\ v_i)} \cdot x$$

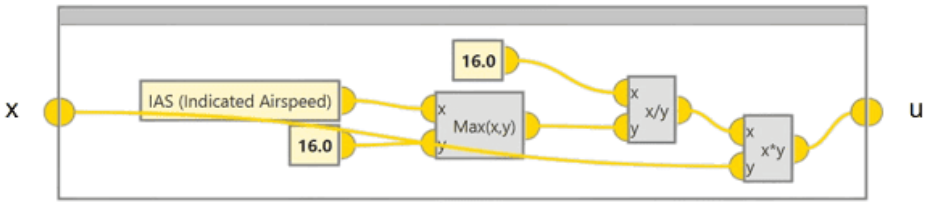
Where,

- $u$ : output
- $v_i$ : nominal speed, in this case 16 m/s
- $v$ : actual measured speed, in this case Indicated Airspeed (IAS)
- $x$ : input

Therefore, it would be:

$$u = \frac{16}{\max(IAS,\ 16)} \cdot x$$

Note that this formula has a lower saturation value of 16 m/s, so the **scaling** in this case is done for **speeds higher than the nominal**.



**Ias Quadratic Scaler Block - Explanation**

IAS Quadratic Scaler allows **quadratic scaling** two input variables with the indicated airspeed.

This block implements the following formulas:

$$u_1 = \left( \frac{v_i}{\max(v,\ v_i)} \right)^2 \cdot x_1$$

$$u_2 = \left( \frac{v_i}{\max(v,\ v_i)} \right)^2 \cdot x_2$$

Where,

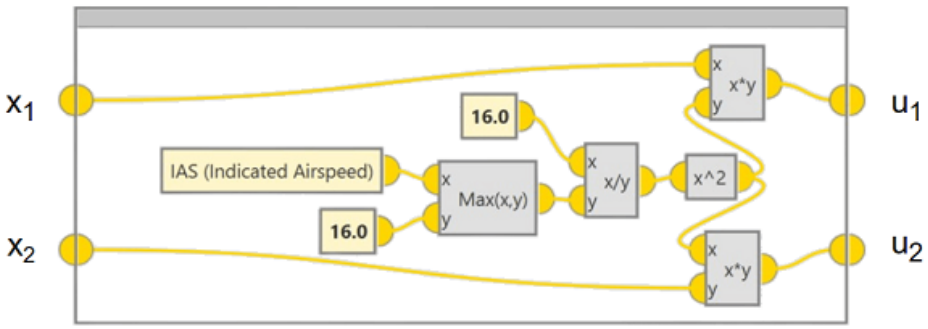
- $u_1$ : output variable 1
- $u_2$ : output variable 2
- $v_i$ : nominal speed, in this case 16 m/s
- $v$ : actual measured speed, in this case Indicated Airspeed (IAS)
- $x_1$ : input variable 1
- $x_2$ : input variable 2

Therefore, it would be:

$$u_1 = \left( \frac{16}{\max(IAS,\ 16)} \right)^2 \cdot x_1$$

$$u_2 = \left( \frac{16}{\max(IAS,\ 16)} \right)^2 \cdot x_2$$

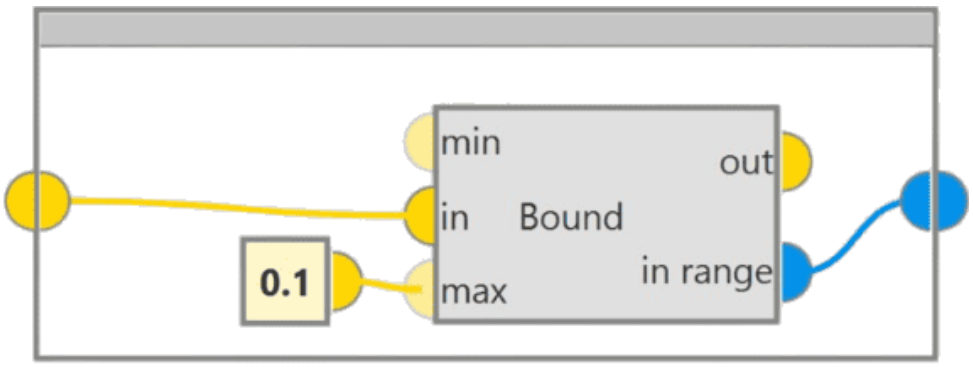
Note that this formula has a lower saturation value of 16 m/s, so the **scaling** in this case is done for **speeds higher than the nominal**.




### Imbalance Block - Explanation

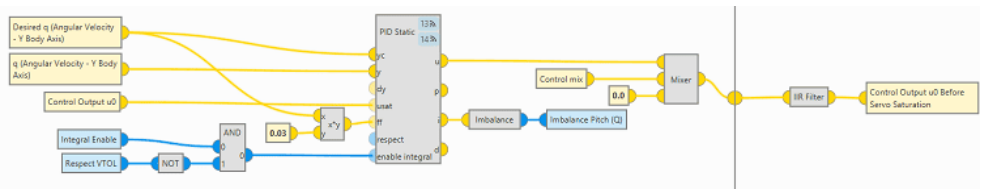
This **Imbalance** block is used to determine whether a variable remains constant over time. This is achieve by checking that the rate of change of the evaluated variable is as close as possible to **0.0**.

To do this, the input variable is evaluated with the Bound block where the output in range is set to **true** if the value of the variable is **less than 0.1**.




Once the custom blocks have been created, in this program the different control outputs are defined for each of the control actions in quadcopter or FW configuration:

- **Pitch quadcopter - u0**

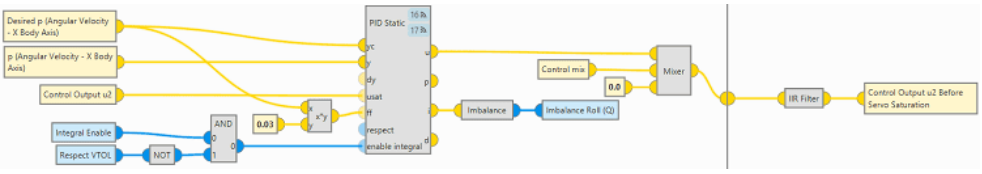



  - PID controller allows to calculate the output response to reach the Desired q (Angular Velocity - Y Body Axis) variable.

- **Mixer** block provides the output signal from the PID controller to the **quadcopter** flight configuration rate.
- IIR Filter block adapts the signal with a low pass filter. In this way, the aircraft is able to perform a controlled actuation.

> ⓘ **Note**
> This control logic allows Autopilot 1x to compare the variable **q (Angular Velocity - Y Body Axis)**, obtained in the **Navigation** program, and the **Desired q (Angular Velocity - Y Body Axis)** variable, obtained in the **Guidance** program, and determine the response that the aircraft must have to reach the desired variable. To achieve this desired value, the **Control Output u0** variable is determined, which defines the position of the actuators/servos for **pitching** in **quadcopter** flight configuration.

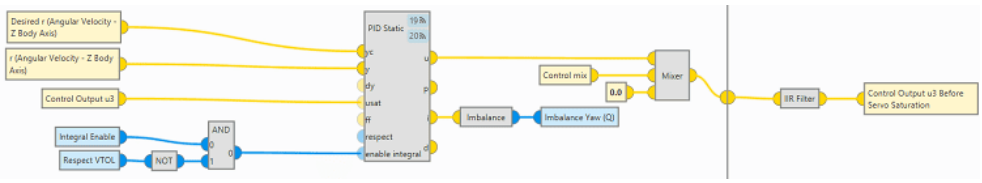
- **Roll quadcopter - u2**




- PID controller allows to calculate the output response to reach the Desired p (Angular Velocity - X Body Axis) variable.
- **Mixer** block provides the output signal from the PID controller to the **quadcopter** flight configuration rate.
- IIR Filter block adapts the signal with a low pass filter. In this way, the aircraft is able to perform a controlled actuation.

> **Note**
>
> This control logic allows Autopilot 1x to compare the variable **p (Angular Velocity - X Body Axis)**, obtained in the **Navigation** program, and the **Desired p (Angular Velocity - X Body Axis)** variable, obtained in the **Guidance** program, and determine the response that the aircraft must have to reach the desired variable. To achieve this desired value, the **Control Output u2** variable is determined, which defines the position of the actuators/servos for **rolling** in **quadcopter** flight configuration.

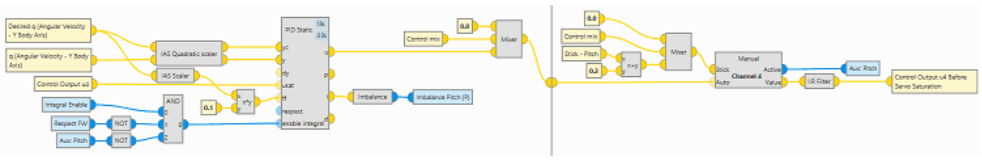
- **Yaw quadcopter - u3**




  - PID controller allows to calculate the output response to reach the Desired r (Angular Velocity - Z Body Axis) variable.
  - **Mixer** block provides the output signal from the PID controller to the **quadcopter** flight configuration rate.
  - IIR Filter block adapts the signal with a low pass filter. In this way, the aircraft is able to perform a controlled actuation.

> **Note**
>
> This control logic allows Autopilot 1x to compare the variable **r (Angular Velocity - Z Body Axis)**, obtained in the **Navigation** program, and the **Desired r (Angular Velocity - Z Body Axis)** variable, obtained in the **Guidance** program, and determine the response that the aircraft must have to reach the desired variable. To achieve this desired value, the **Control Output u3** variable is determined, which defines the position of the actuators/servos for **yawing** in **quadcopter** flight configuration.

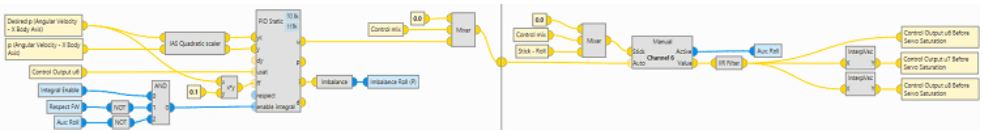
- **Pitch FW - u4**



  - **IAS Quadratic scaler** block
  - **IAS Scaler** block
  - PID controller providing the desired change variable to **Pitch** in **FW** flight configuration.
  - **Mixer** block provides the output signal from the PID controller to the **FW** flight configuration rate.
  - Manual block changes the output value depending on the control mode.
  - IIR Filter block adapts the signal with a low pass filter. In this way, the aircraft is able to perform a controlled actuation.
  - InterpVec block applies the configured table interpolation on each of the components of the input vector.

> ⓘ **Note**
>
> This control logic allows Autopilot 1x to compare the variable **q (Angular Velocity - Y Body Axis)**, obtained in the **Navigation** program, and the **Desired q (Angular Velocity - Y Body Axis)** variable, obtained in the **Guidance** program, and determine the response that the aircraft must have to reach the desired variable.
>
> To achieve this desired value, the **Control Output u4** variable is determined, which defines the position of the actuators/servos for **pitching** in **FW** flight configuration.

- **Roll FW - u6**



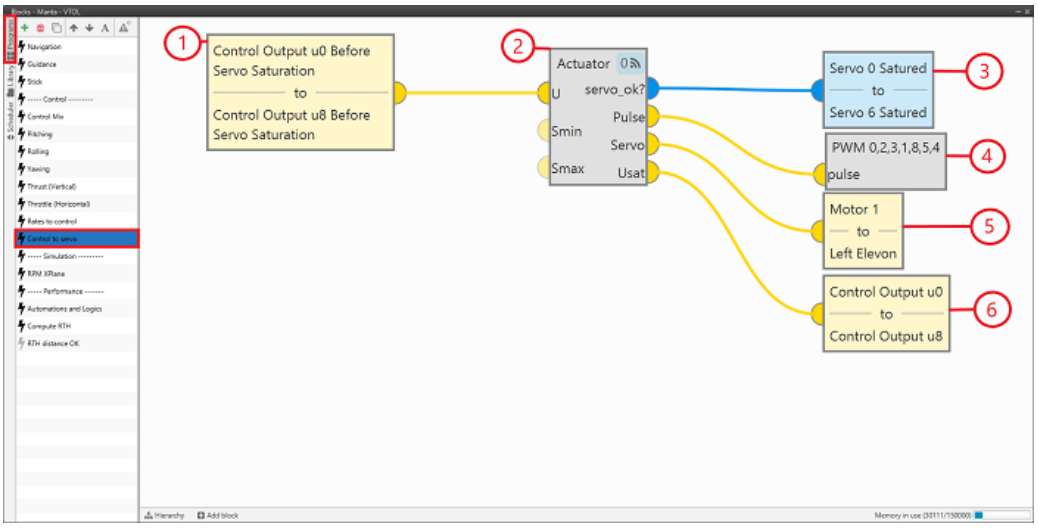
  - **IAS Quadratic scaler block**
  - PID controller providing the desired change variable to **Roll** in **FW** flight configuration.

- **Mixer** block provides the output signal from the PID controller to the **FW** flight configuration rate.
- Manual block changes the output value depending on the control mode.
- IIR Filter block adapts the signal with a low pass filter. In this way, the aircraft is able to perform a controlled actuation.
- InterpVec block applies the configured table interpolation on each of the components of the input vector.

> ⓘ **Note**
>
> This control logic allows Autopilot 1x to compare the variable **p (Angular Velocity - X Body Axis)**, obtained in the **Navigation** program, and the **Desired p (Angular Velocity - X Body Axis)** variable, obtained in the **Guidance** program, and determine the response that the aircraft must have to reach the desired variable. To achieve this desired value, the **Control Output u6** variable is determined, which defines the position of the actuators/servos for **rolling** in **FW** flight configuration.

Control to servo

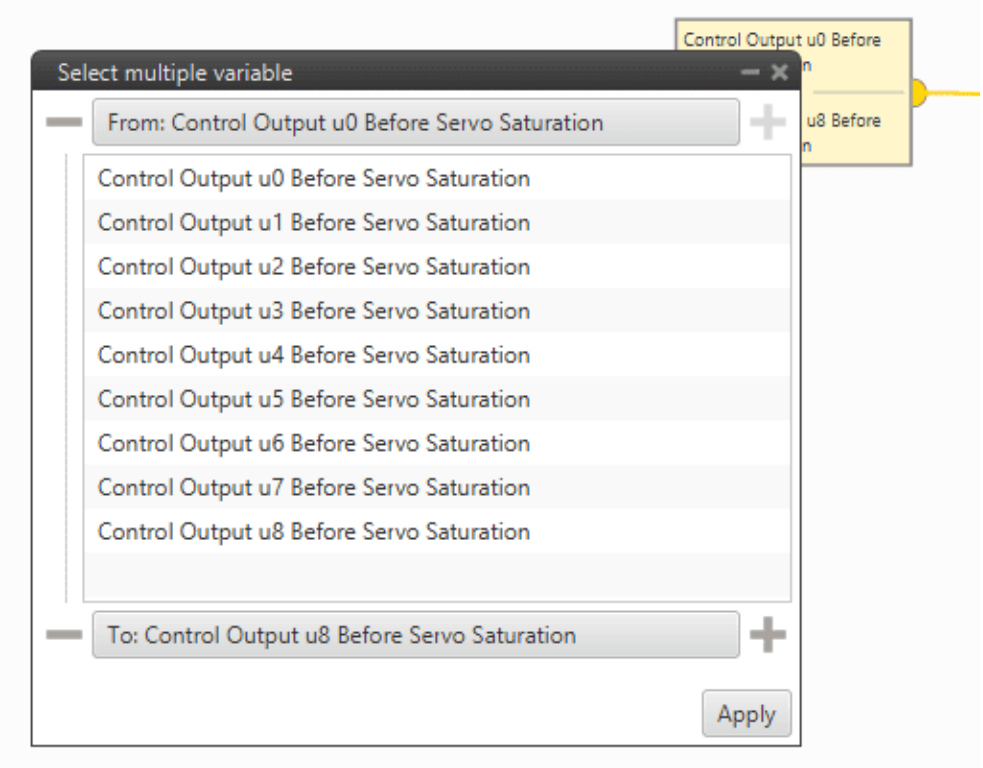
The following program allows the user to configure the transformation between the action and the servo value. To do this, the **Actuator** block and the different inputs and outputs must be configured.




**Control to servo - Program**

In this program it is necessary to define and configure the following blocks:

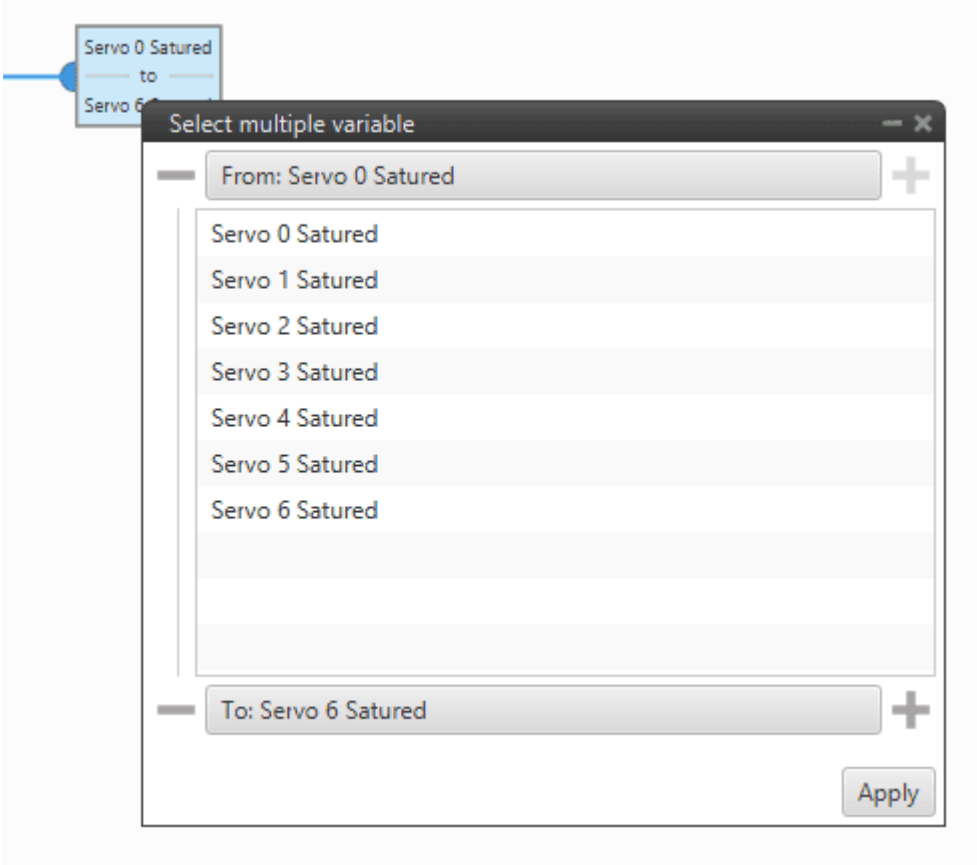
1. A Read multiple Reals block with the variables that define the actions commanded by the control programs.




2. Actuator block that transforms input variables into servo values.

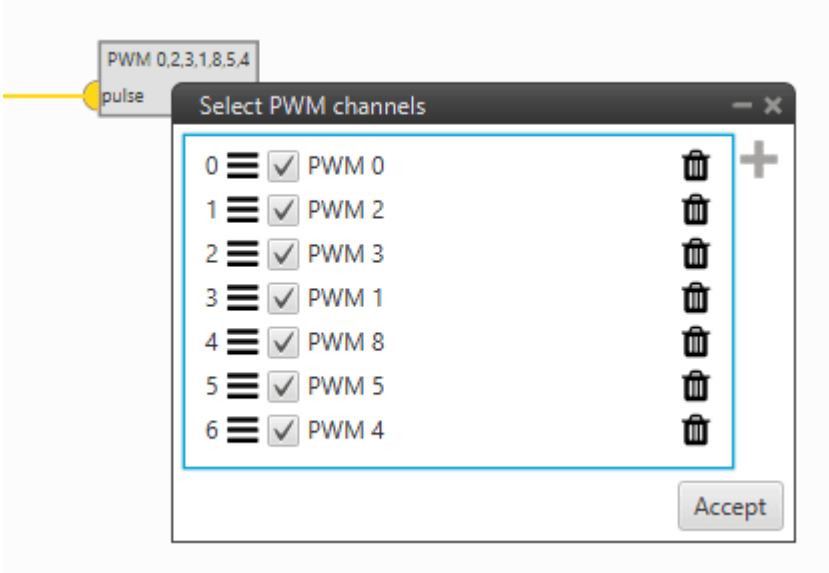
> **Important**
>
> The configuration of the **Actuator** block is defined in the Actuators - Actuators & SU Matrix section of this manual.

3. A Write multiple Bits block with the servo bits that had to be trimmed to prevent saturation.

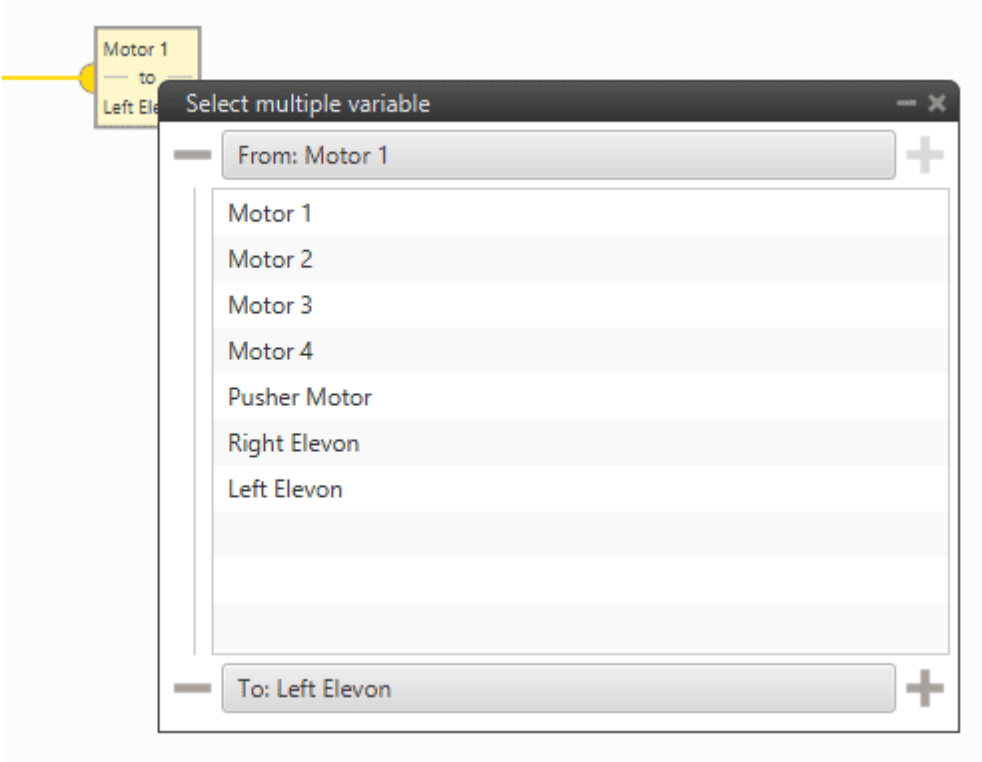



4. A PWM block to indicate the PWM variables to which the PWM pulse should be transmitted for each servo.

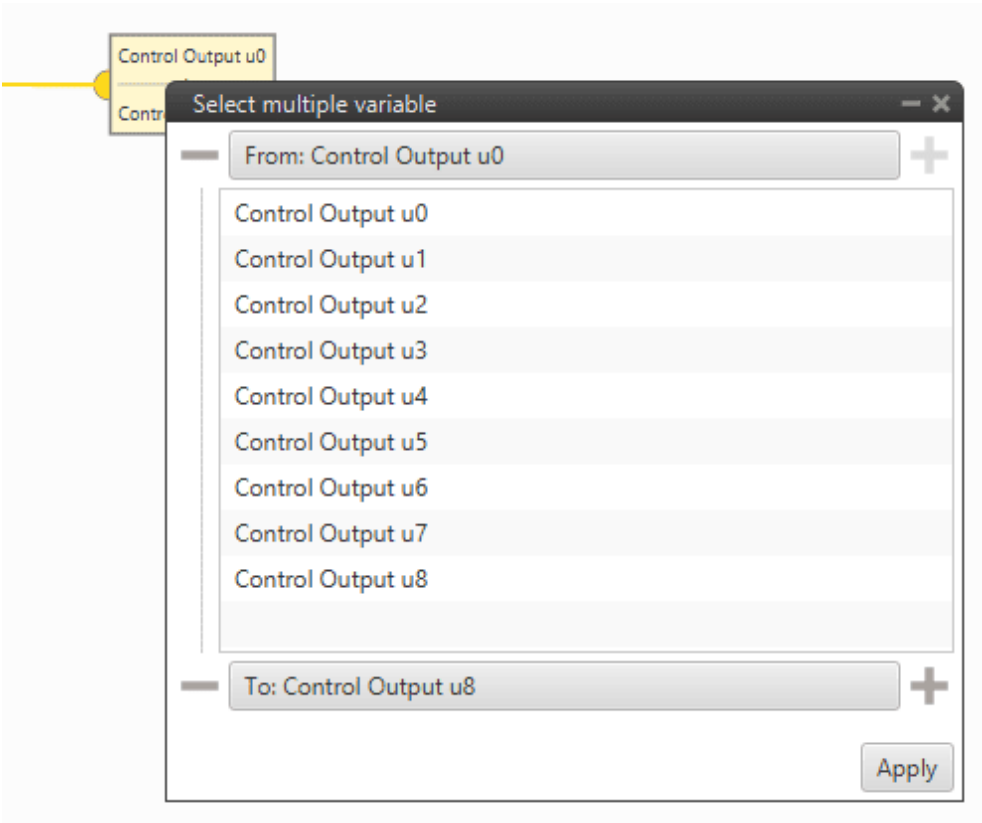
> **Note**
> The pins to which the servos are connected must match those previously changed from GPIO to PWM.



5. A Write multiple Reals block that writes the servo value that generates the actuator motion to the servo variables.



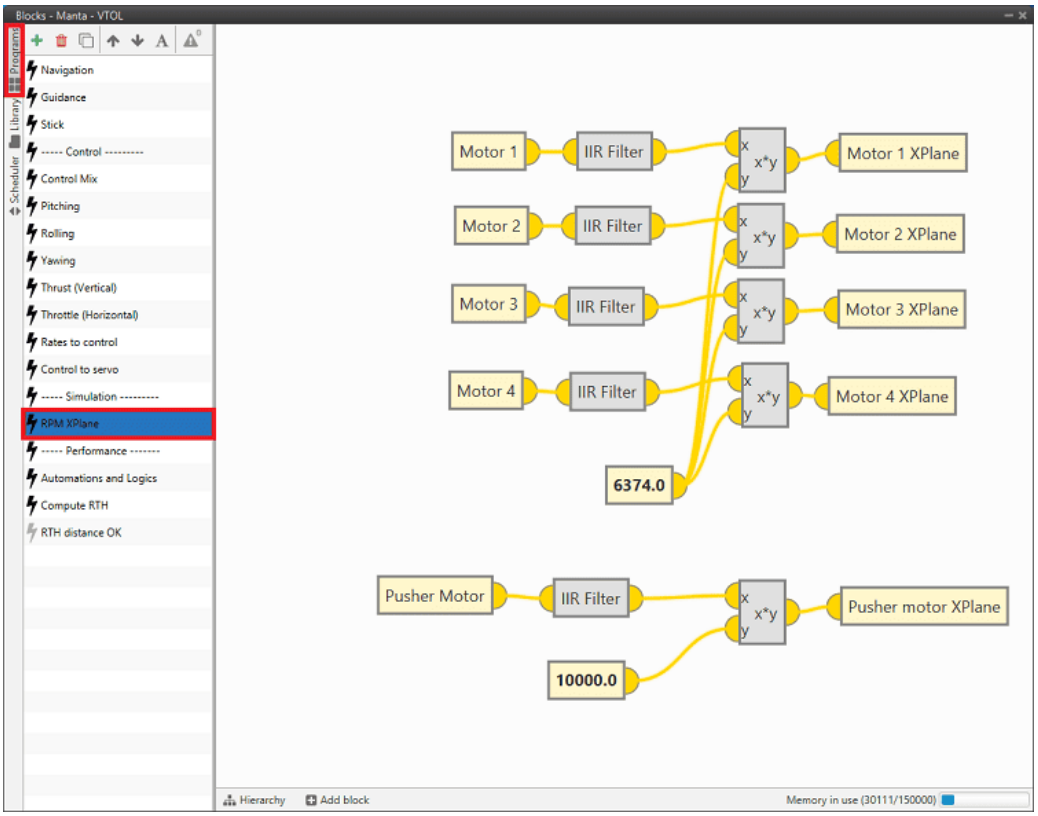

6. A Write multiple Reals block with the variables for Control actions after servo saturation.



RPM XPlane

The following program is defined to adapt the performance of the aircraft in real flight to a simulation environment with X-Plane11 software.

This allows to use the same 1x air configuration without having to make 2 different ones for the "phase" of creating and validating the configuration and the real operation.



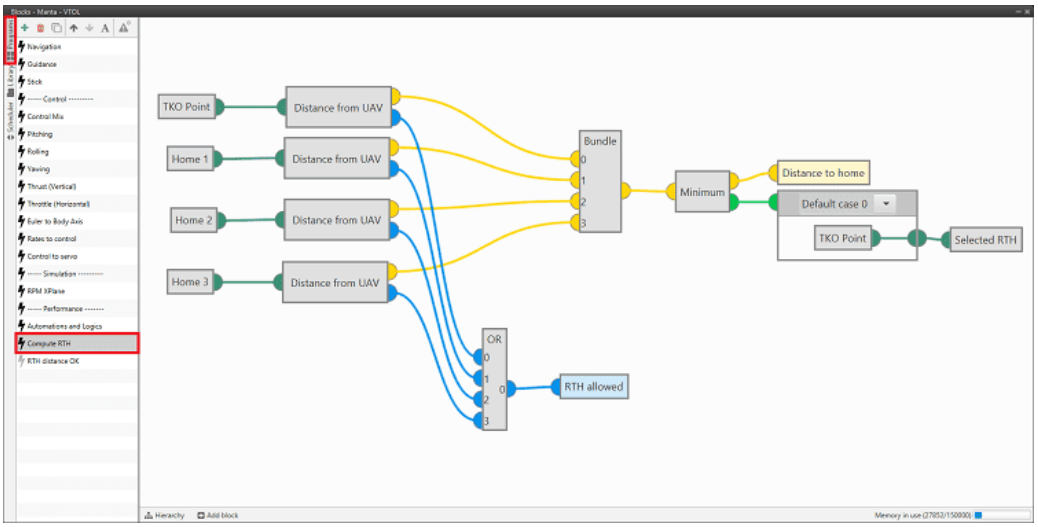

**RPM XPlane - Program**

For this purpose, the IIR Filter block is used. This block is a low pass filter that allows the motor RPM of the motor to be raised in a controlled way.

Motor X variables have a value from **0 to 1** and are multiplied by the **maximum RPM** of the motor. The result of this multiplication will be the value used in the simulation.

Compute RTH

The following program is defined to set which **Home point** the Autopilot 1x will use when the Return to Home (RTH) phase is executed.



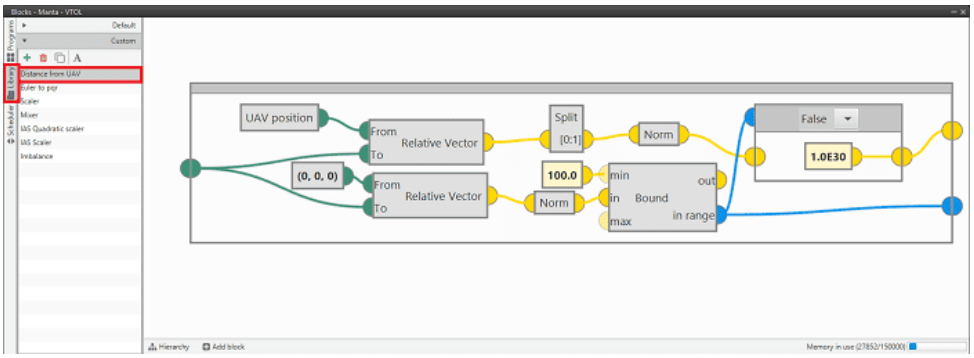

## Compute RTH - Program

The program is defined as follows:

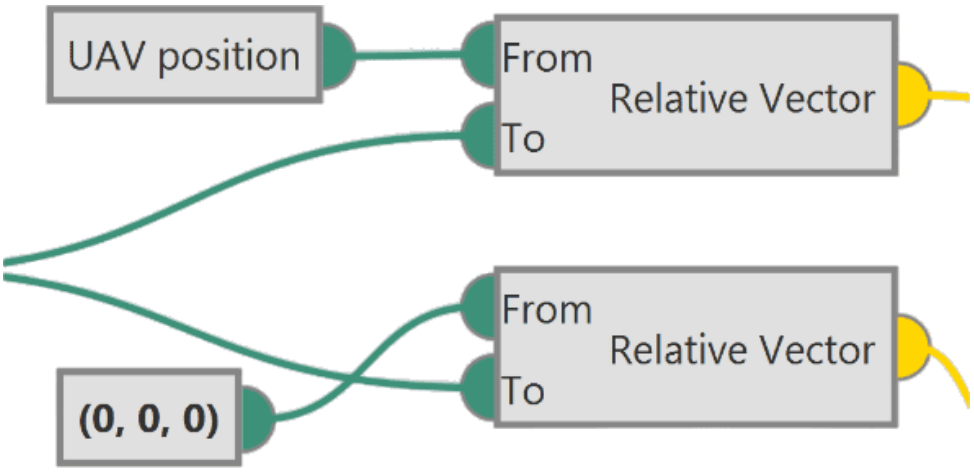
The distance between the UAV and each of the predefined points (TKO Point, Home 1, Home 2, Home 3) is calculated using the **Distance from UAV** block. This is a **custom** block created in the **Library blocks**.

### Distance From Uav Block - Explanation

This block follows the following structure:

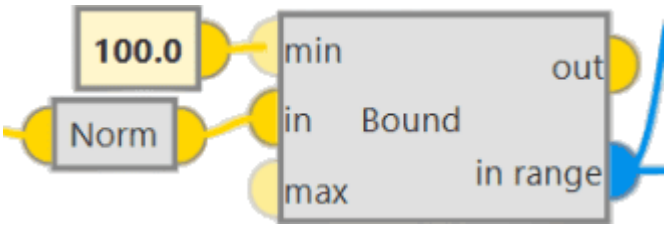



The distance **from the UAV position to the predefined point** and the distance **from the NED origin to the predefined point** are calculated. These distances are calculated using the **Relative Vector** block, which returns a relative vector in NED frame from the two input positions. For more information on this block, visit the Positions blocks - Block Programs section of the **1x PDI Builder** user manual.



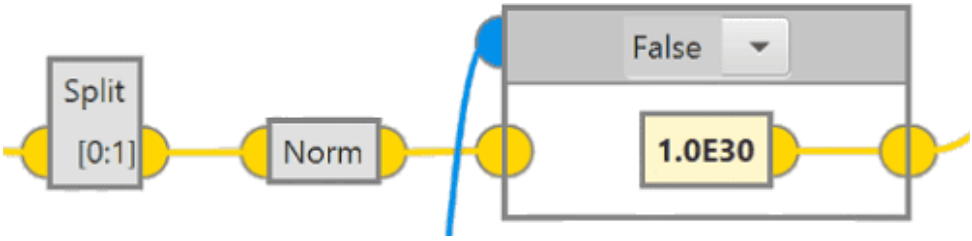

#### Distance from (0, 0, 0) to Home

- Once this distance has been calculated, the Relative Vector block returns a vector of three components (North, East, Down).
- Then, the Norm block returns the norm of this vector and the Bound block evaluates if this value is within the defined limits (**100 m minimum**). The reason for defining this minimum is because if no Home point has been predefined, Autopilot 1x will take as point the origin **(0, 0, 0)**. Thus, to avoid guidance errors when flying around the origin, this minimum distance is defined.



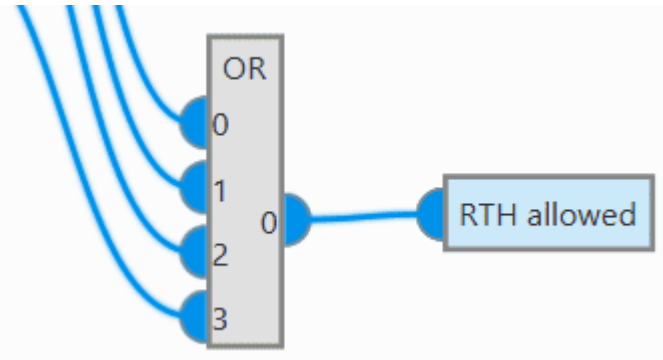

**Distance from UAV position to Home**

- Once this distance has been calculated, the Relative Vector block returns a vector of three components (North, East, Down).
- The first 2 components of this vector should be used in the program to determine the distance between the two points. To do this, the Split block is used to create a vector with the first two variables and then, the Norm block returns the norm of this vector.
- This value will only be taken as valid if the **distance to the origin** is **greater than 100 m**, so the Bound block will return the bit as **true**. Otherwise, if the bit is **false**, i.e., the distance is **less than 100 m**, a distance large enough for this point to be discarded as a valid point is defined.

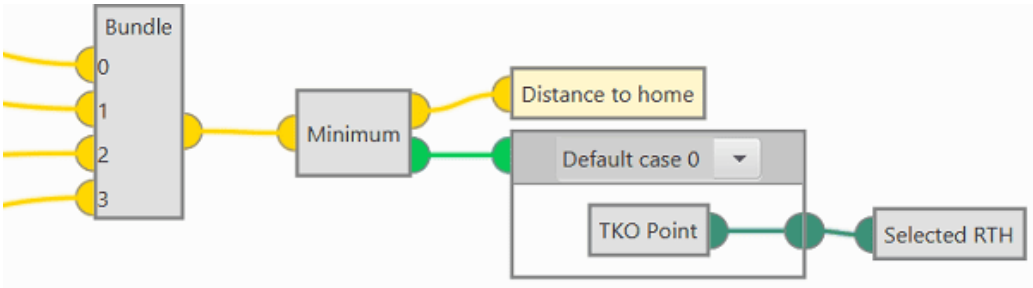



For more information on custom blocks, visit the Library blocks - Block Programs section of the **1x PDI Builder** user manual.

After evaluating the distance between the defined points and the UAV, the bit RTH allowed will be **true** if one or more of them can be taken as a **valid point**. In case **no valid point** is found, this bit will be **false** and it will **not** be possible to transition to the **RTH phase**.

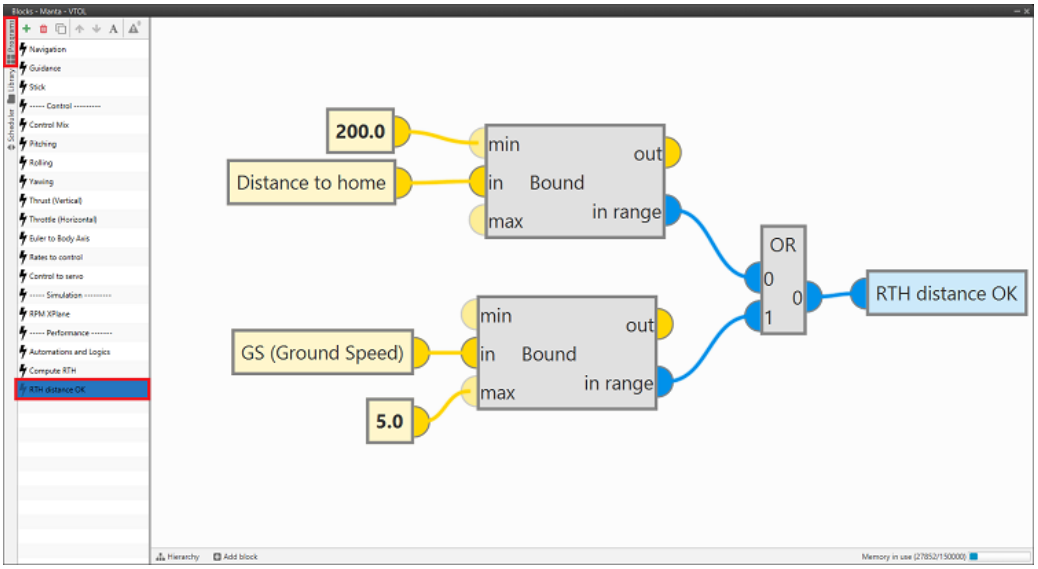



Finally, among all the valid Home points, the one to which the UAV has less distance is chosen and selected as the point to use in the RTH phase.




RTH distance OK

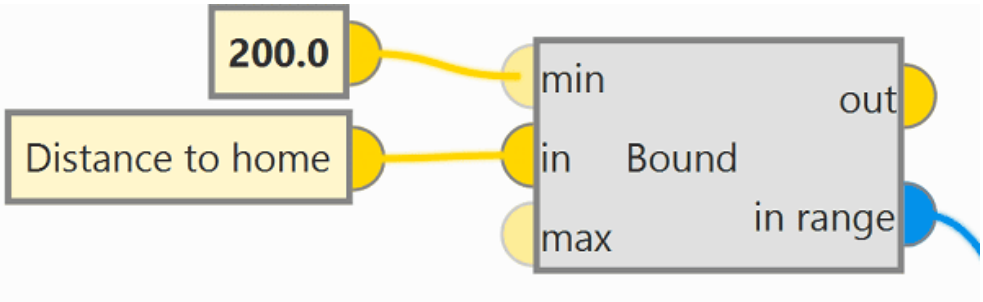
The following program is defined to determine whether the **Home point** to be used in the **RTH phase** selected in the previous program, Compute RTH, is valid or not.



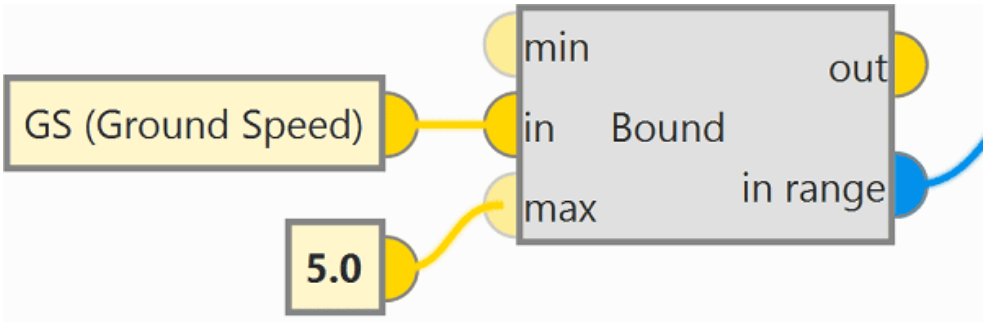

**RTH distance OK - Program**

For this purpose, the program defines two conditions that determine the validity of the **Home point** selected:

- **Distance to home**
  A **minimum distance** of **200 meters** is set at which the **Home point** has to be from the **UAV**. If the value of the Distance to home variable is within the defined range, the Bound block returns a **true** bit.




- **GS (Ground Speed)**
  A **maximum GS** of **5.0 m/s** is defined. If this speed is within the defined range, the Bound block returns a **true** bit.

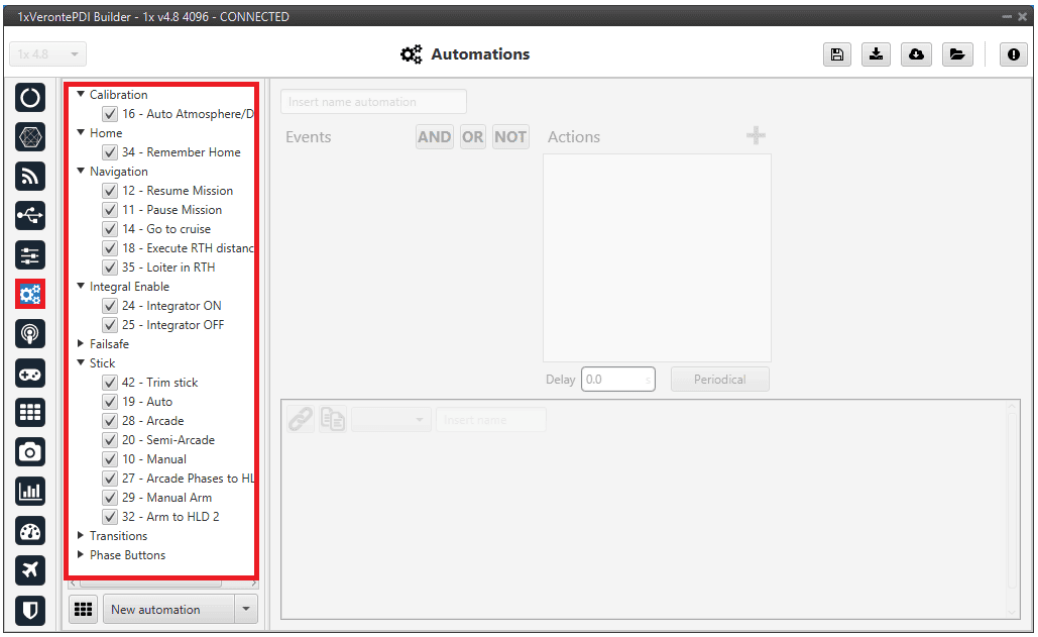



If **one** of these two conditions is met, the **Home point** is taken as **valid** and Autopilot 1x will guide to this point. Otherwise, if neither of the two conditions is met, the Loiter in RTH automation will cause the aircraft to perfom a **Loiter** until one of those two conditions is met.

These conditions allow the aircraft to perform a **controlled transition**, having enough distance and time to perform it safely.

## Automations

Automations are actions that are carried out when a combination of events happens, i.e., when the events are accomplished the action is done. For more information on automations, visit the Automations section of the **1x PDI Builder** user manual.

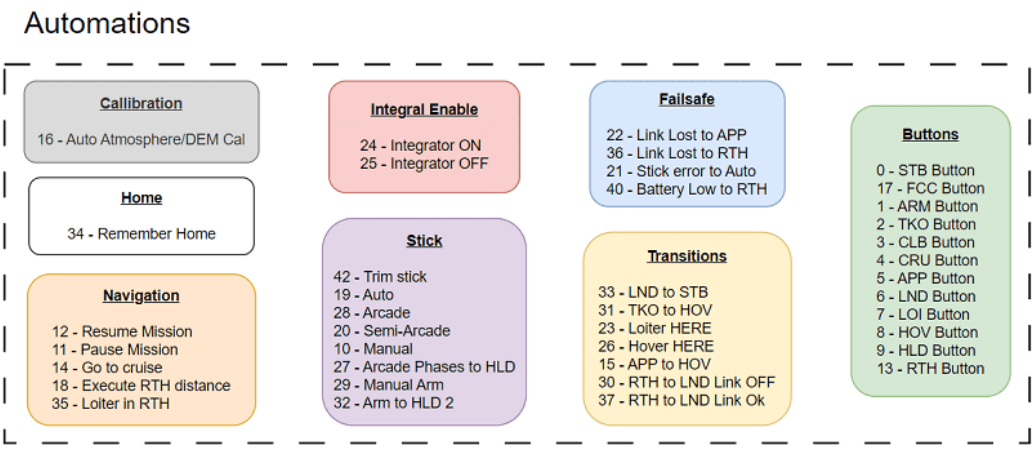



## VTOL Automations

The different events, actions and automations that have been created for the operation of the Embention VTOL aircraft are shown below.

> ⚠ **Warning**
>
> The user can create as many automations as desired up to a maximum of **500 events**, **120 actions** and **100 automations**.

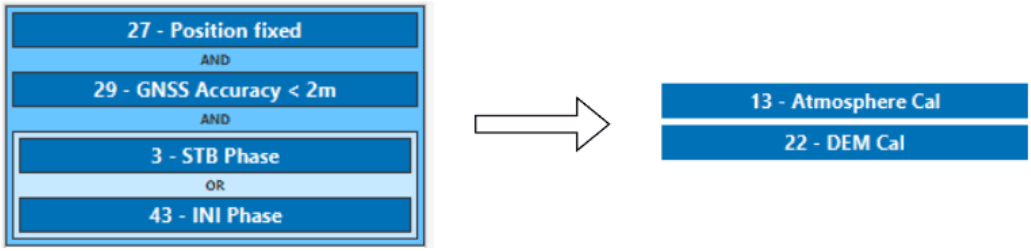
The automations have been divided into groups depending on the functionality for which they have been created:




Calibration

### 16 - Auto Atmosphere/DEM Cal

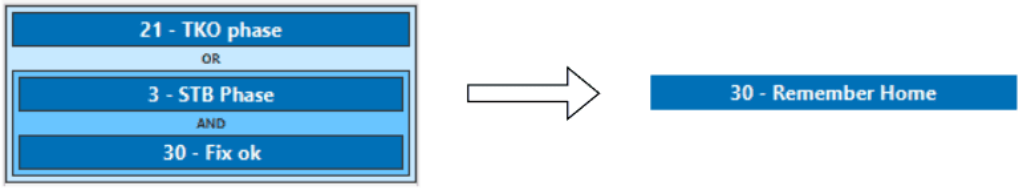
This automation establishes the events that must be triggered for the autopilot to perform an atmospheric calibration. For this, the autopilot position must be fixed, with a GNSS accuracy of less than 2 meters, and be in the STB or INI phase.




Home

### 34 - Remember Home

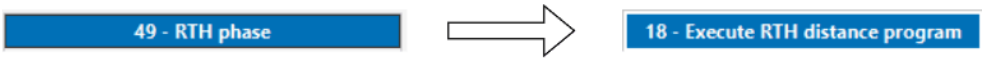
This automation remembers the take-off point. In the RTH phase, the autopilot guides the aircraft to the nearest Home point, so if no other point has been established in the mission, the autopilot will automatically establish a path to guide the aircraft back to the take-off point.




Navigation

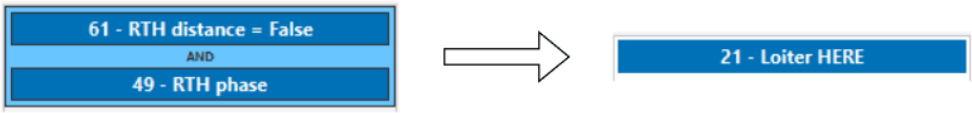
- **18 - Execute RTH distance**

  When the autopilot is in RTH phase, it executes the program RTH distance ok which calculates the distance to the nearest Home point.




- **35 - Loiter in RTH**

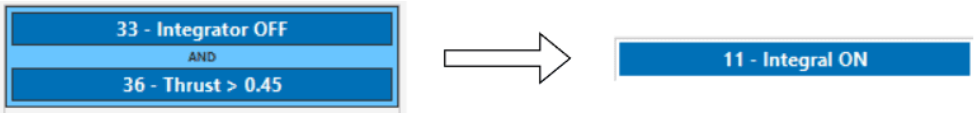
  This automation allows the aircraft to perform a Loiter when it is in RTH phase and does not meet the conditions to take the RTH point as correct.



Integral Enable

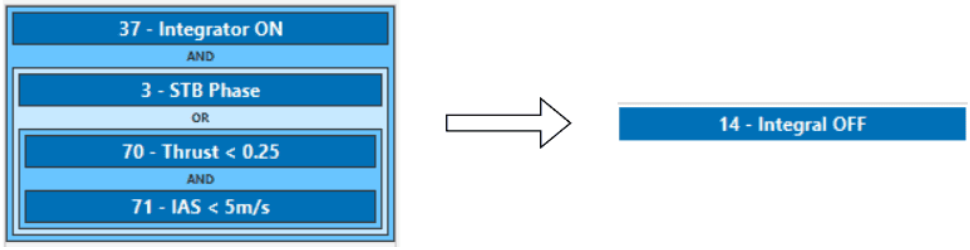
The following automations set the value of the **Integral Enable** bit, which is used as the value for the enable integral input of the **PID controllers** used in Rates to control program.

- **24 - Integrator ON**




  When **Integral ON** is triggered, a value of **1** is stored in the Integral Enable bit, turning **true**. Therefore, the enable integral input of the PDI controller is true, which means that the PDI works as usual.

- **25 - Integrator OFF**



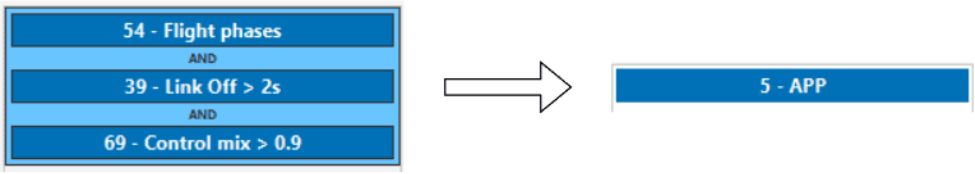

  When **Integral OFF** is triggered, a value of **0** is stored in the Integral Enable bit, turning **false**. Therefore, the enable integral input of the PDI controller is false, which implies that the integral term is exponentially discharged.

Failsafe

The following automatioms are defined to ensure the performance of the aircraft by activating predetermined actions in case of **failure**. For example, in case of stick error, stick connection error or in case of low battery.

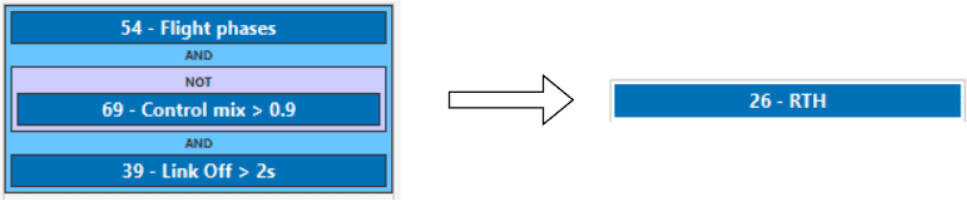
- **22 - Link Lost to APP**

  When the aircraft is in one of the flight phases (Loiter, Hover, Hold, Cruise, Climb, Approach, Return Home or Take-off), with a control mix value **greater than 0.9** and loses radio connection for 2 seconds, it automatically switches to the Approach (APP) phase.



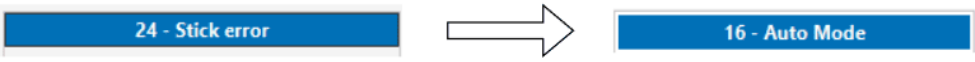
- **36 - Link Lost to RTH**

  When the aircraft is in one of the flight phases (Loiter, Hover, Hold, Cruise, Climb, Approach, Return Home or Take-off), with a control mix value **less than 0.9** and loses radio connection for 2 seconds, it automatically switches to the RTH phase.



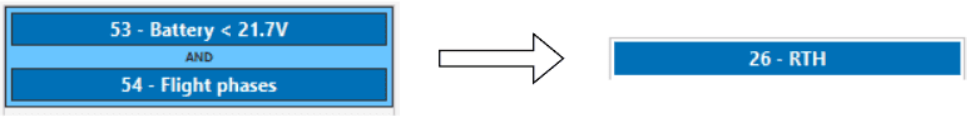

- **21 - Stick error to Auto**

  The autopilot switches to auto mode when Stick error bit is false.




- **40 - Battery Low to RTH**

  When the battery drops below a certain value, in this case 21.7 V, and the aircraft is in one of the flight phases (Loiter, Hover, Hold, Cruise, Climb, Approach, Return Home or Take-off), the autopilot will automatically guide the aircraft to RTH phase.

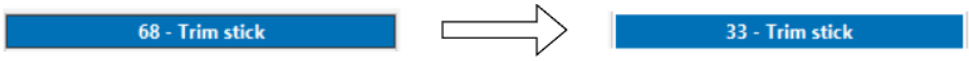



Stick

The following automations change the flight mode depending on the stick position, for this the Stick-mode and Stick-Manual variables are used.

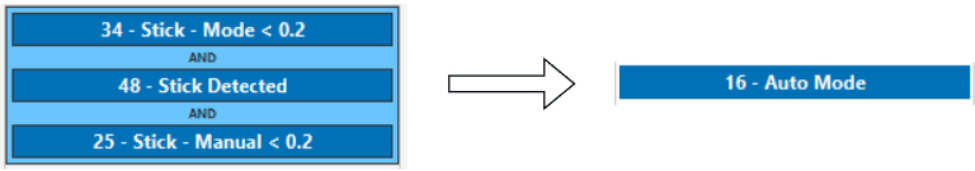
> ⓘ **Note**
>
> - Stick-mode: 3 positions (0; 0.5; 1)
> - Stick-manual: 2 positions (0; 1)

- **42 - Trim stick**



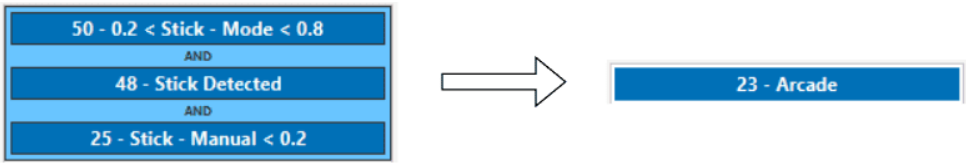

- **19 - Auto**

The autopilot switches to Auto mode when the Stick-mode variable has a value of **0** and the Stick-manual variable has a value of **0**.



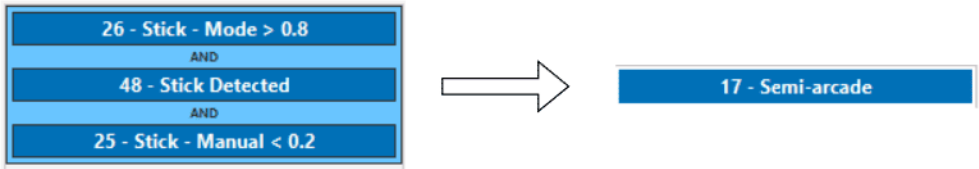

- **28 - Arcade**

  The autopilot switches to Arcade mode when the Stick-mode variable has a value of **0.5** and the Stick-manual variable has a value of **0**.



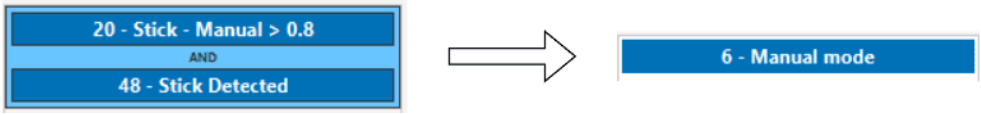

- **20 - Semi-Arcade**

  The autopilot switches to Semi-Arcade mode when the Stick-mode variable has a value of **1** and the Stick-manual variable has a value of **0**.




- **10 - Manual**

  The autopilot switches to Manual mode when the Stick-manual variable has a value of **1**.



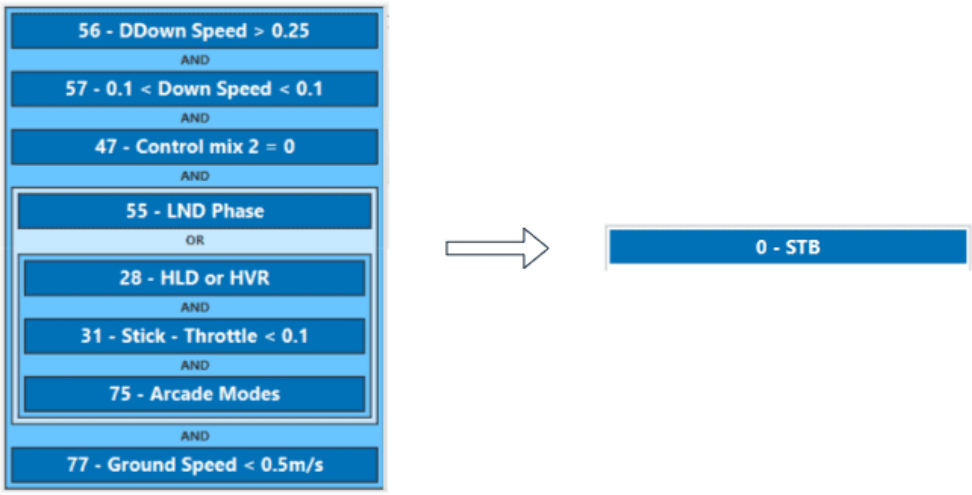

Transitions

- **33 - LND to STB**

  This automation is complex due to the number of events that must be triggered for the aircraft to automatically transition from the Landing phase to the Standby phase:

  - The aircraft must be on the ground, which means that the **Down Speed is zero** (event 57) as well as have a speed with respect to the ground that allows it to stop safely (event 77).
  - While in Landing phase, the **Desired Down Speed** (DDown Speed) must have a value **greater than 0** (event 56).

- The aircraft must be in **quadcopter flight configuration** (event 47).
- The aircraft must be in the Landing phase or in the Hold or Hover phases, in arcade mode and with a value of 0 in the Stick-throttle command.




- **31 - TKO to HOV**

  The following automation indicates the height above ground level at which the aircraft passes from Take-off to the Hover phase.

- **23 - Loiter HERE**

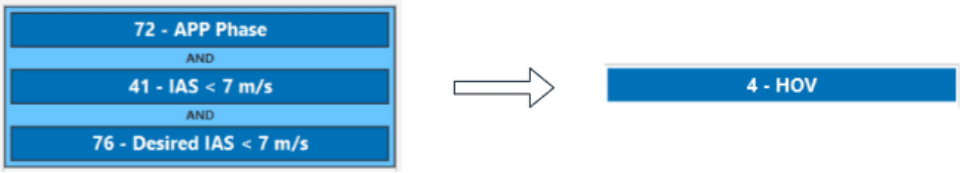
  With this automation, the aircraft performs a Loiter around the point where it transitions to the Loiter phase.

- **26 - Hover HERE**

  The aircraft performs the Hover actuation at the point where it transitions to the Hover phase.

- **15 - APP to HOV**

  In the Approach phase, if the airspeed is less than 7 m/s, the aircraft will move to the Hover phase at that point. This automation is used to perform a Hover over the defined Home point.

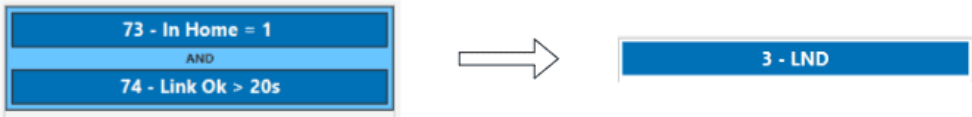



- **30 - RTH to LND Link Off**

  If Hover is performed at the defined Home point and the connection is lost for 1 minute, Autopilot 1x transitions to the Landing phase.

- **37 - RTH to LND Link Ok**

  When the aircraft performs a Hover over the Home point for more than 20 seconds without losing connection, it automatically transitions to the Landing phase.




Phase Buttons

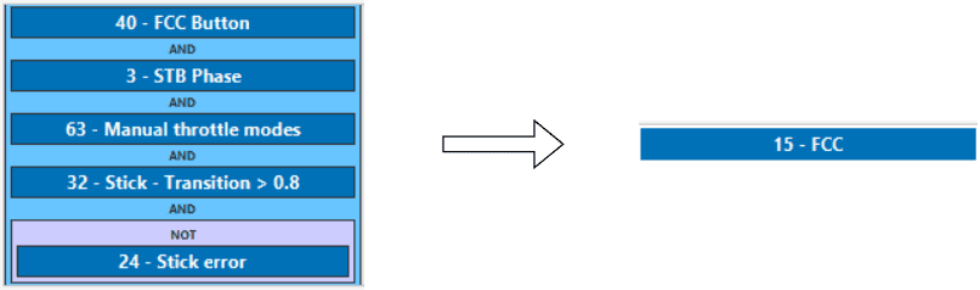
The following automations are defined for entering and transitioning between flight phases in Veronte Ops. These actions are triggered by buttons that will appear in the **Veronte Panel**.

For some of these automations it will only be necessary to "slide" the button to trigger the flight phase change action, however, for others it will be necessary that some additional event is fulfilled to trigger the phase change.

> ⓘ **Note**
>
> For more information on the possible flight phases, visit Phases definition - Modes & Phases section of this manual.

- **17 - FCC Button**.

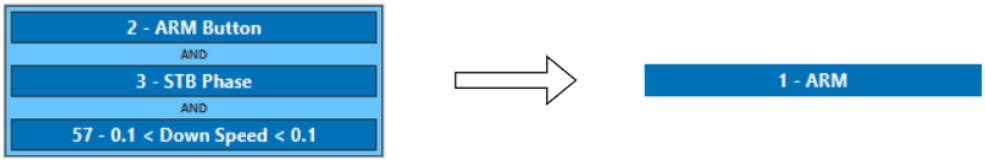
  To start the Flight Control Check (FCC) phase, in addition to pressing the FCC Button, Autopilot 1x must be in STB phase and the stick pilot must command the 1x.




This flight phase is only performed in FW flight configuration, so the Stick - Transition variable that defines the configuration in which the aircraft flies must be greater than 0.8. This is because this variable is 0 when the configuration is quadcopter and 1 when the configuration is FW.

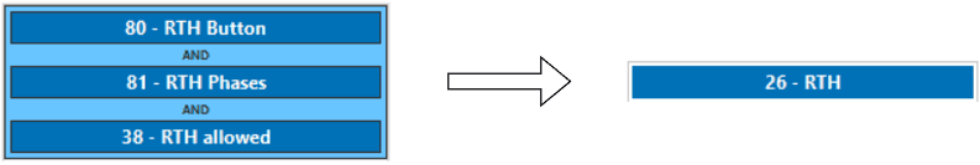
- **1 - ARM Button**.

  To switch to Armed (ARM) phase, the ARM Button must be pressed, the autopilot must be in STB phase and the Down speed must be around 0.



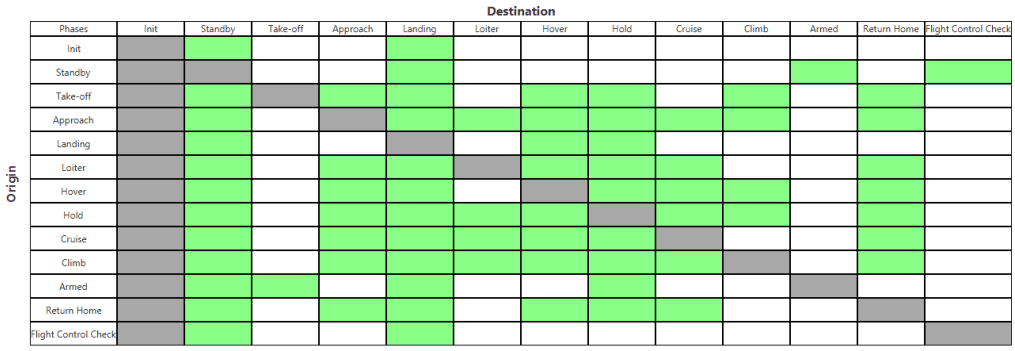

- **13 - RTH Button**.

  To switch to RTH phase, the RTH Button must be pressed, RTH allowed bit must be true (when the autopilot can guide the platform to a defined Home point) and the autopilot must be in one of the flight phases that allows it to switch to RTH (HOV, APP, LOI, CRUISE, HLD).




Transitions table

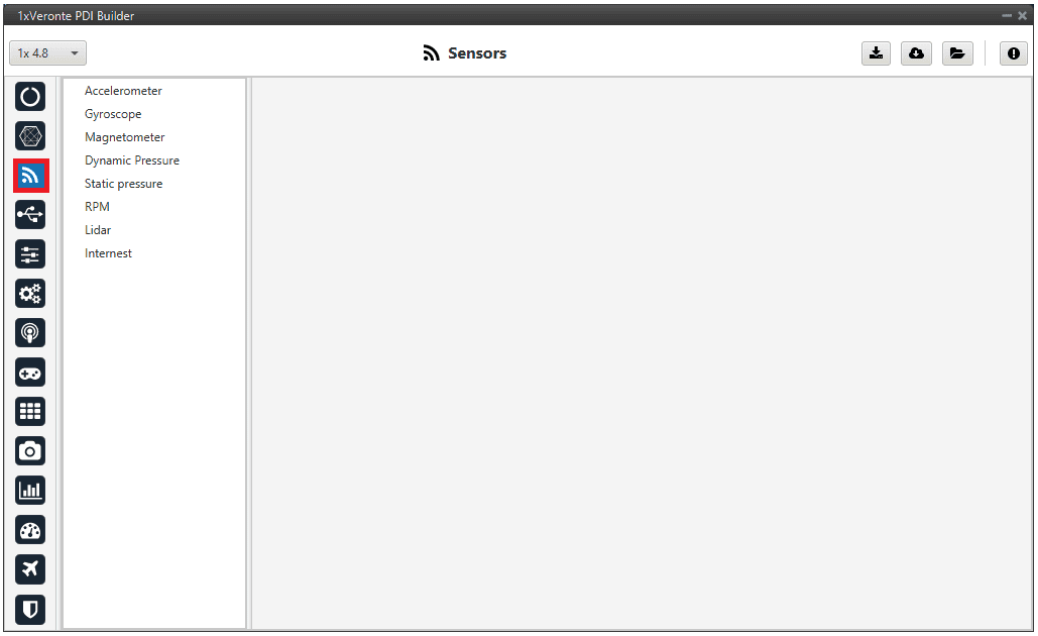
The following table shows the different possibilities for a transition between two flight phases.




**Transitions Table**

## Sensors

The configuration of the different sensors used for autopilot navigation is explained below.



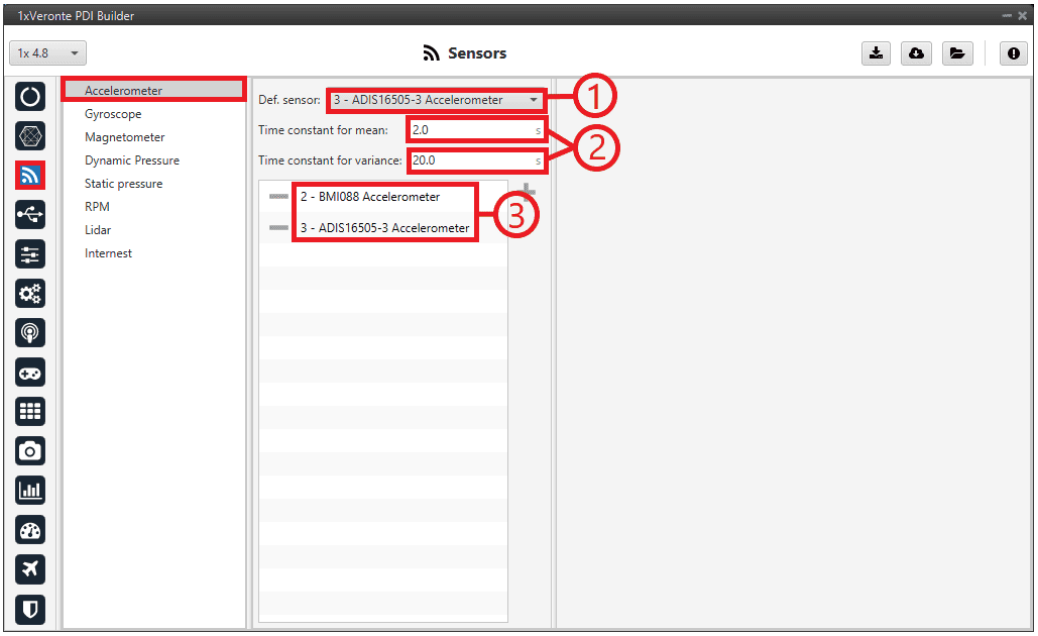

**Sensors Panel**

Veronte Autopilot 1x incorporates 3 Inertial Measurement Units (IMUs) that allow the 1x system to measure different variables and that are the main source of navigation data.

## Accelerometer

> **Note**
>
> For a detailed explanation of the available accelerometer sensors, please visit the Accelerometer - Sensors section of the **1x PDI Builder** user manual.

The accelerometer from the IMU can be configured as explained in the panels shown below.

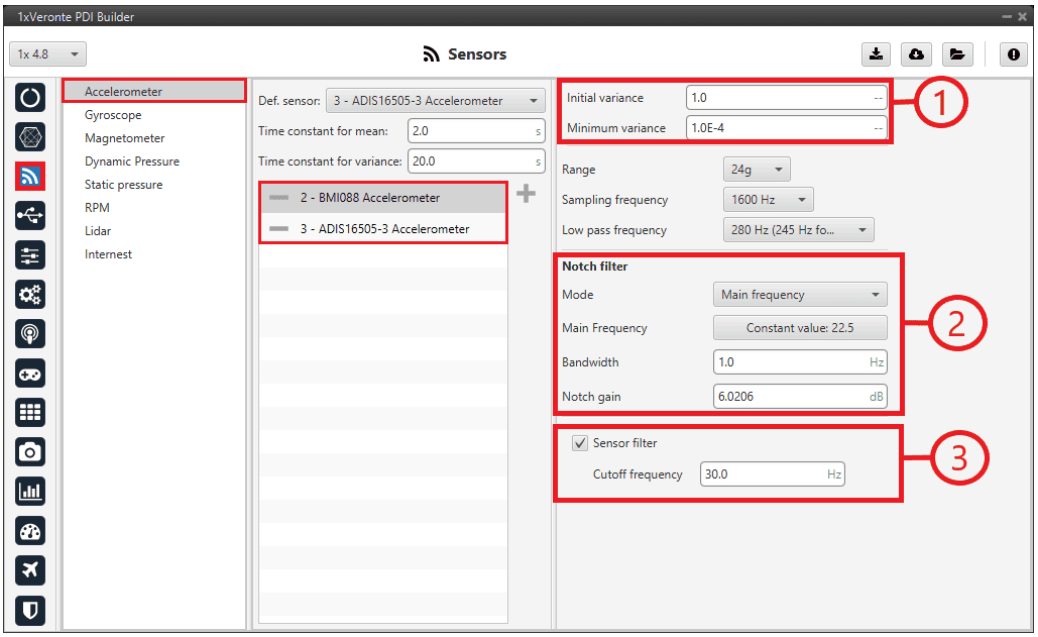



**Accelerometer Panel**

> **Important**
>
> It is recommended to select multiple of these sensors for the navigation algorithm, so that Veronte Autopilot 1x performs a combination of all the measurements of the selected accelerometers.
>
> This combination consists of calculating the means and variances of each of these accelerometers in a given time (Time constant for mean and Time constant for variance) to obtain a weighted mean with the inverse of the variance. The lower the variance, the greater the weight of that sensor in the mean.

1. The ADIS16505-3 Accelerometer sensor has been chosen as the **default sensor**. If all other sensors fail, the measurement value will be that of the default sensor.
2. A time of 2 seconds has been defined for the **Time constant for mean**, and 20 seconds for the **Time constant for variance**.
3. The **BMI088** and **ADIS16505-3** accelerometers sensors have been selected and their configuration is described below.

### Common configuration of BMI088 Accelerometer and ADIS16505-3 Accelerometer



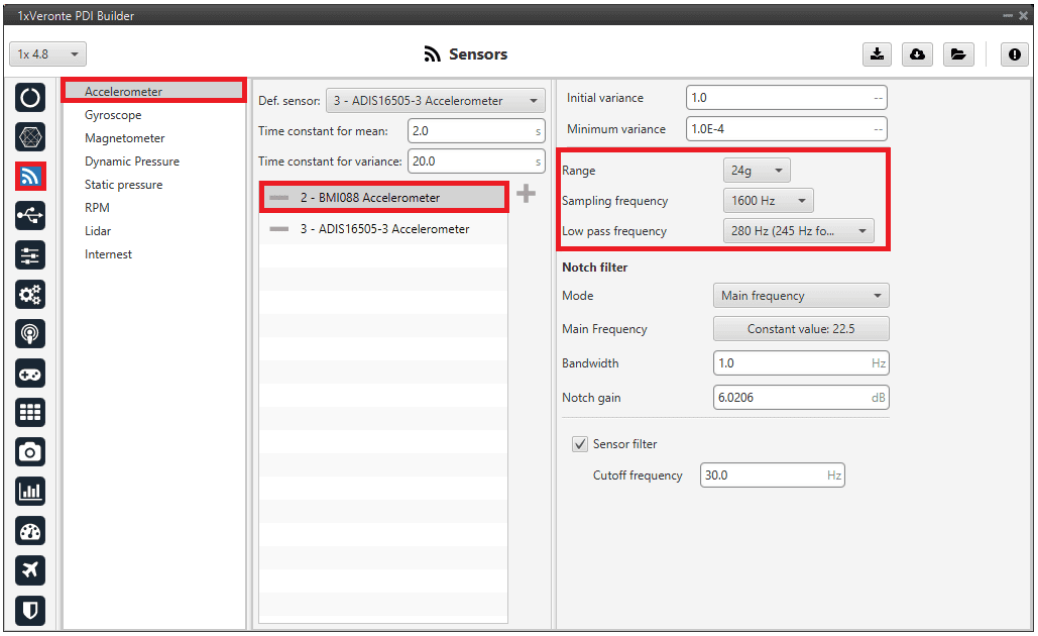
### Common configuration

1. **Initial variance = 1.0**

   **Minimum variance = 1.0E-4**
2. **Notch filter**: It is a filter that dampens signals only at a specific frequency.
   - **Mode**: Main frequency
   - **Main Frequency**: 22.5
   - **Bandwidth**: 1.0 Hz
   - **Notch gain**: 6.0206 dB
3. **Sensor filter**: Enabled.
   - **Cutoff frequency**: 30.0 Hz

### BMI088 Accelerometer

This panel displays the possible parameters that can be configured for the internal BMI088 Accelerometer.



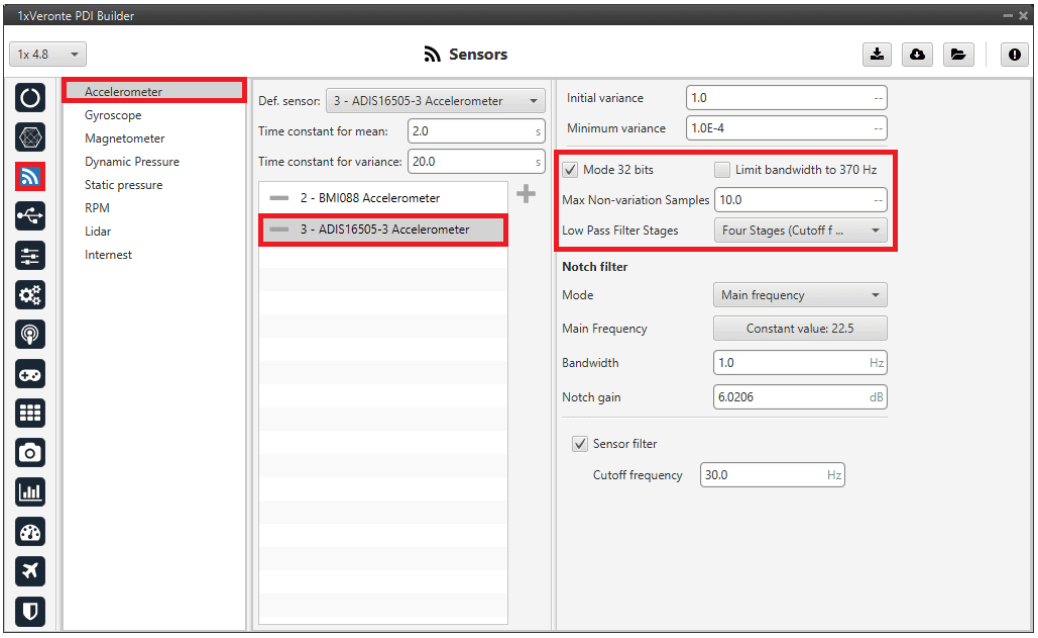

**BMI088 Accelerometer Panel**

- **Range**: We select a high range of forces that the accelerometer can measure. The lowest accuracy is accepted so that the system does not saturate.
- **Sampling frequency**: That is the frequency at which the measurements are read out. We recommend the highest (**1600Hz**).

- **Low pass frequency**: The cutoff frequency is set to **280Hz (245 Hz for the Z axis)**.

### ADIS16505-3 Accelerometer

This panel displays the possible parameters that can be configured for the internal ADIS16505-3 Accelerometer.



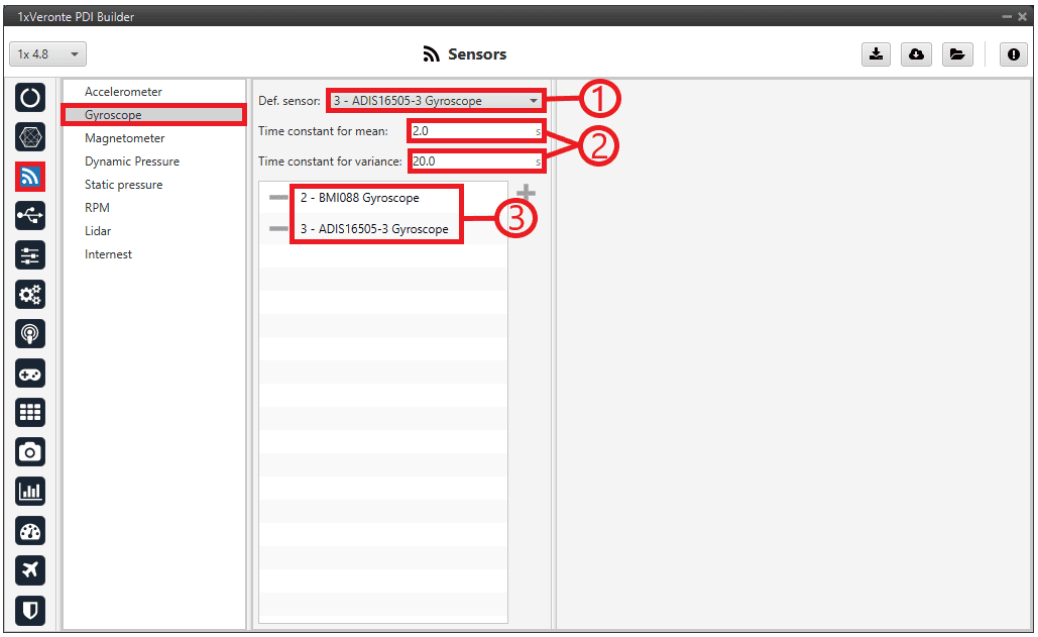

**ADIS16505-3 Accelerometer Panel**

- **Mode 32 bits**: Following the recommendation, we **enable** it.
- **Limit bandwith to 370Hz**: Following the recommendation, we **disable** it.
- **Max Non-variation Samples**: This is configured manually.
- **Low Pass Filter Stages**: Following the recommendation, **4 stages (Cutoff f=40Hz)** are configured.

Gyroscope

> ⓘ **Note**
>
> For a detailed explanation of the available gyroscope sensors, please visit the Gyroscope - Sensors section of the **1x PDI Builder** user manual.

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

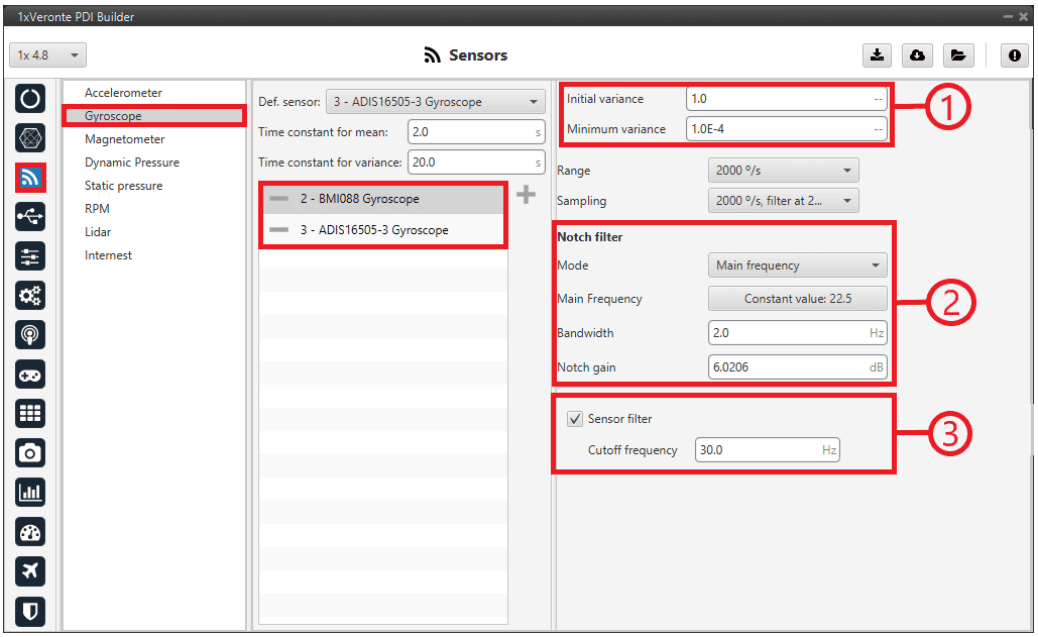



**Gyroscope Panel**

> **Important**
>
> It is recommended to select multiple of these sensors for the navigation algorithm, so that Veronte Autopilot 1x performs a combination of all the measurements of the selected gyroscopes.
>
> This combination consists of calculating the means and variances of each of these gyroscopes in a given time (Time constant for mean and Time constant for variance) to obtain a weighted mean with the inverse of the variance. The lower the variance, the greater the weight of that sensor in the mean.

1. The ADIS16505-3 Gyroscope sensor has been chosen as the **default sensor**. If all other sensors fail, the measurement value will be that of the default sensor.
2. A time of 2 seconds has been defined for the **Time constant for mean**, and 20 seconds for the **Time constant for variance**.
3. The **BMI088** and **ADIS16505-3** gyroscopes sensors have been selected and their configuration is described below.

### Common configuration of BMI088 Gyroscope and ADIS16505-3 Gyroscope



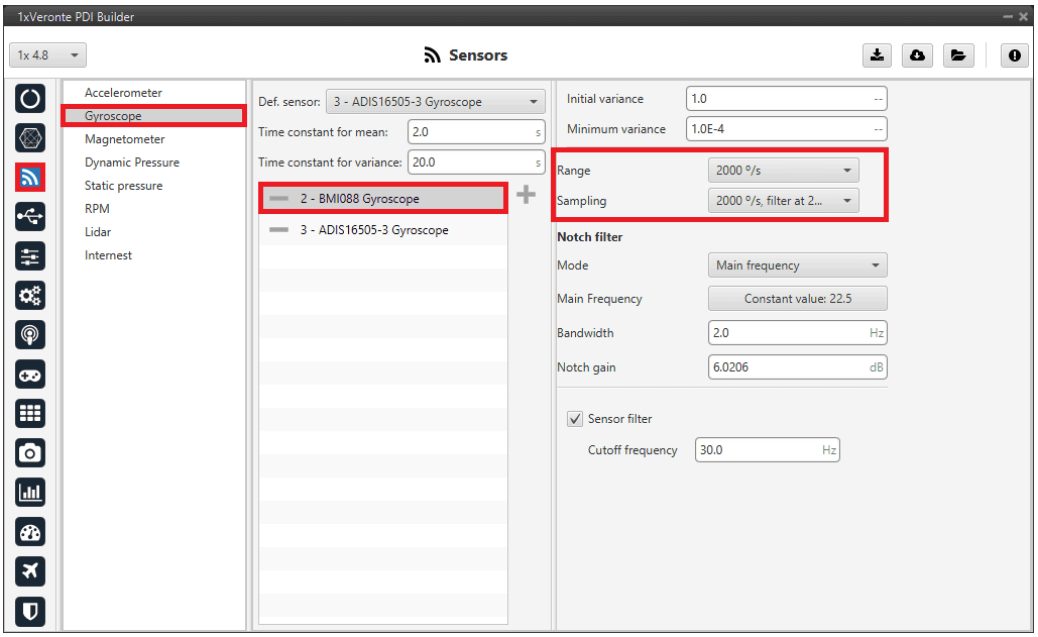

### Common configuration

1. **Initial variance = 1.0**

   **Minimum variance = 1.0E-4**
2. **Notch filter**: It is a filter that dampens signals only at a specific frequency.
   - **Mode**: Main frequency
   - **Main Frequency**: 22.5
   - **Bandwidth**: 1.0 Hz
   - **Notch gain**: 6.0206 dB
3. **Sensor filter**: Enabled.
   - **Cutoff frequency**: 30.0 Hz

### BMI088 Gyroscope

This panel displays the possible parameters that can be configured for the internal BMI088 Gyroscope.

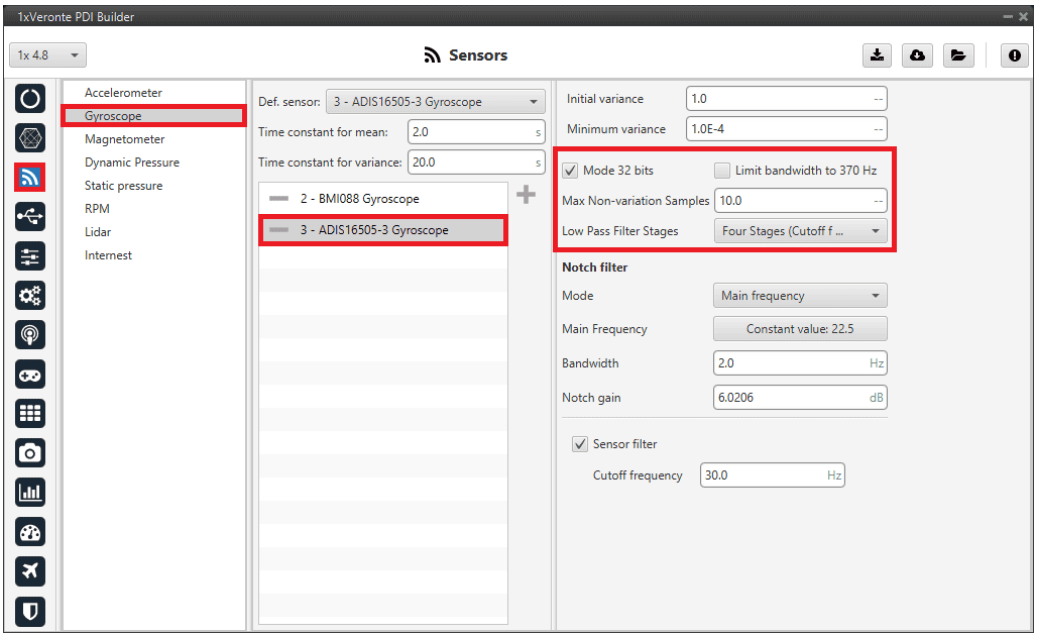



**BMI088 Gyroscope Panel**

- **Range**: We select a high performance range, **2000°s.**. The lower accuracy is accepted so that the system does not saturate.
- **Sampling**: The angular velocity is set to **2000°s filter at 230 Hz**.

### ADIS16505-3 Gyroscope

This panel displays the possible parameters that can be configured for the internal ADIS16505-3 Gyroscope.



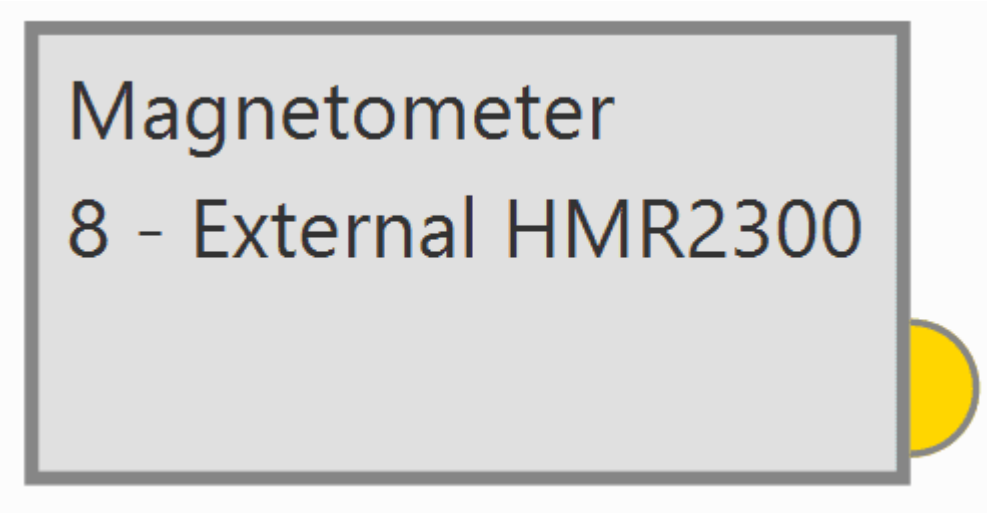

**ADIS16505-3 Gyroscope Panel**

- **Mode 32 bits**: Following the recommendation, we **enable** it.

- **Limit bandwith to 370Hz**: Following the recommendation, we **disable** it.
- **Max Non-variation Samples**: This is configured manually.
- **Low Pass Filter Stages**: Following the recommendation, **4 stages (Cutoff f=40Hz)** are configured.

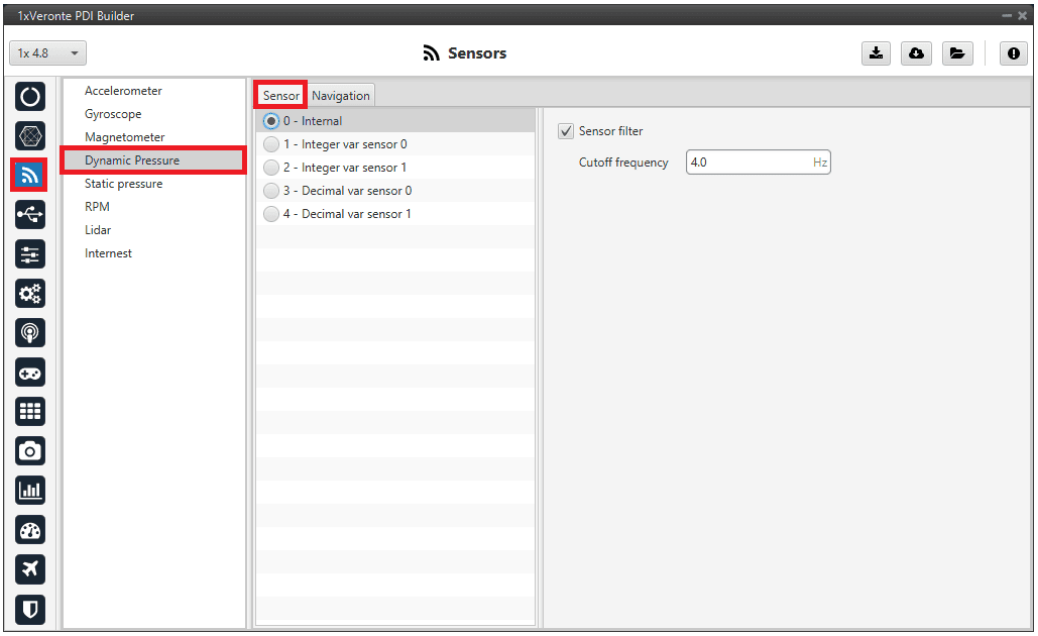
### Magnetometer

It is not necessary to configure the magnetometer in this section. It is selected directly in the Navigation program - Block programs.




### Dynamic Pressure

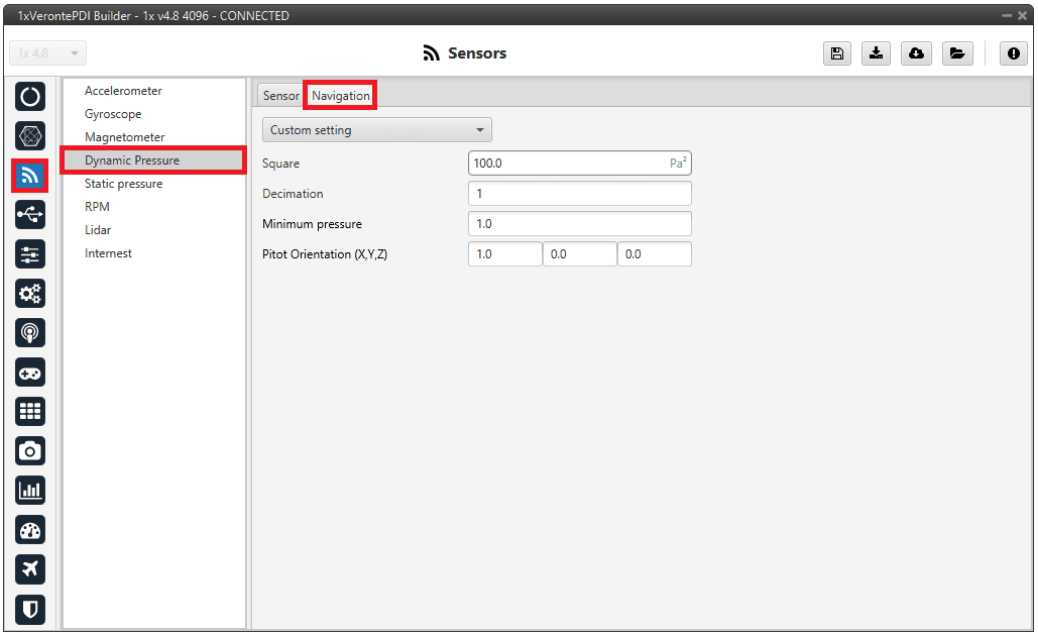
The Internal channel has been selected. In this case the Autopilot 1x uses a value provided by the internal sensor.




**Sensor panel - Dynamic Pressure**

A **cutoff frequency** of **1.0 Hz** has been defined in the **Sensor** panel for the low pass filter. It is necessary to **enable** the **sensor filter** to be able to define the cutoff frequency.

The image below shows a basic configuration of the **Navigation** panel.

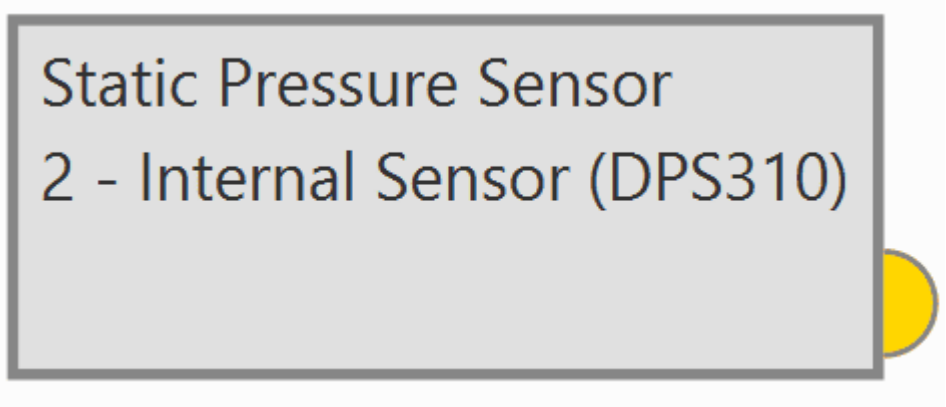



**Navigation panel - Dynamic Pressure**

> ⓘ **Note**
>
> For a detailed explanation of the available dynamic pressure sensors, please visit the Dynamic Pressure - Sensors section of the **1x PDI Builder** user manual.

Static Pressure

It is not necessary to configure the Static Pressure sensor in this section. It is selected directly in the Navigation program - Block programs.

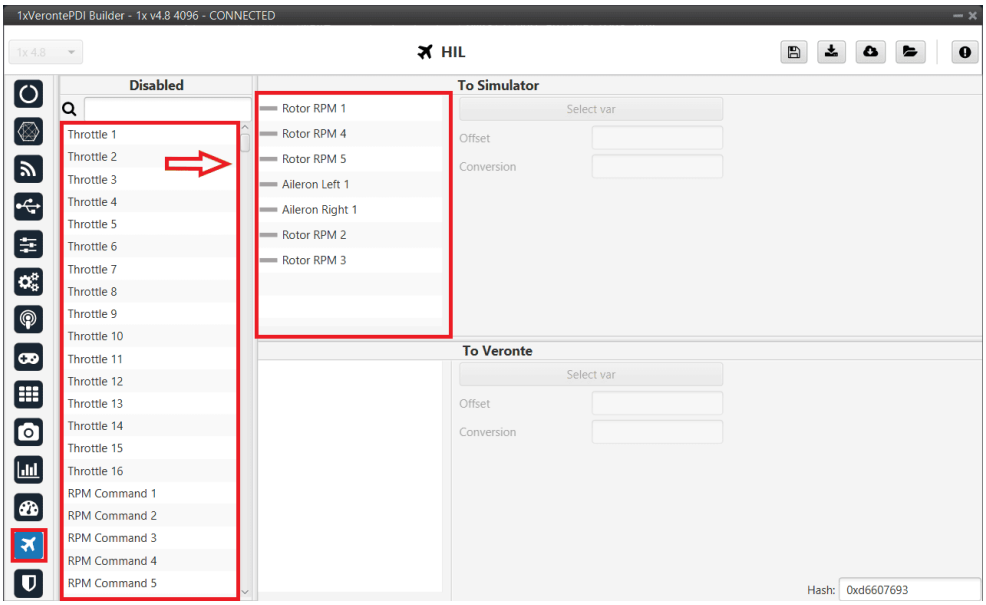



## HIL Configuration

HIL Simulator is the Veronte application that allows Veronte Autopilot 1x to communicate with a simulator acting as a "bridge" between both. Therefore, users must configure this menu to simulate the performance of the Embention VTOL aircraft 1x in the selected simulator, in this case X-Plane11 has been chosen. For more information, please visit HIL Simulator user manual.

To configure the simulation variables follow the steps below:

1. Find the **simulator variables** that have been configured in the aircraft model.
   Once found, drag and drop each of them into the **To Simulator** panel. In this panel users can see all the variables selected to be sent to the simulator.
   For Embention VTOL Aircraft select the following variables:
   - Rotor RPM 1
   - Rotor RPM 2
   - Rotor RPM 3
   - Rotor RPM 4
   - Rotor RPM 5
   - Aileron Right 1
   - Aileron Left 1

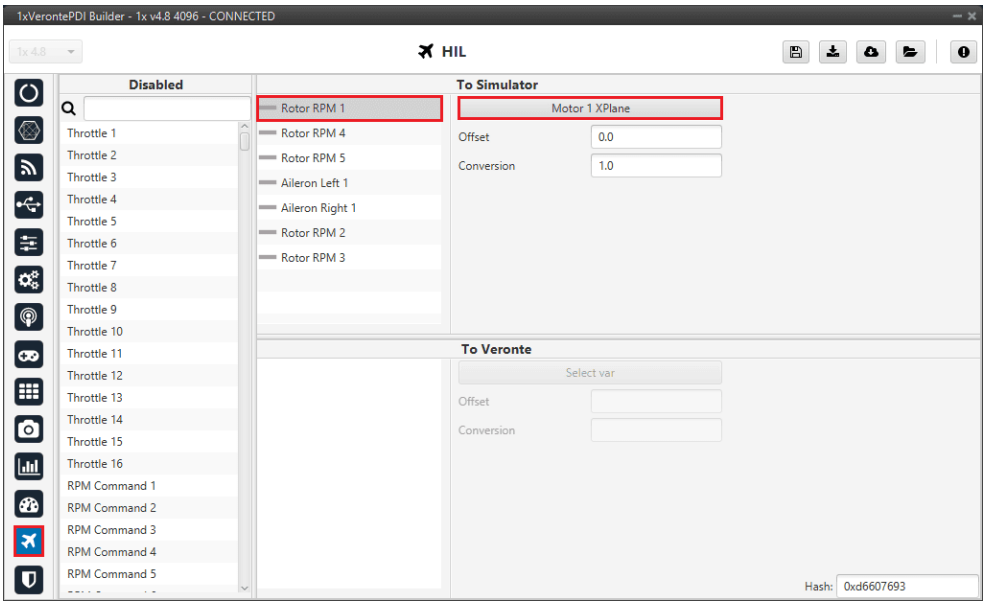



**Simulator variables**

2. Associate each actuator variable (Control Output) of Autopilot 1x that matches that of the simulator.

> ⓘ **Note**
>
> The variables that have been assigned in the Control to servo program to each actuator are the ones to be linked to the simulator internal variables.

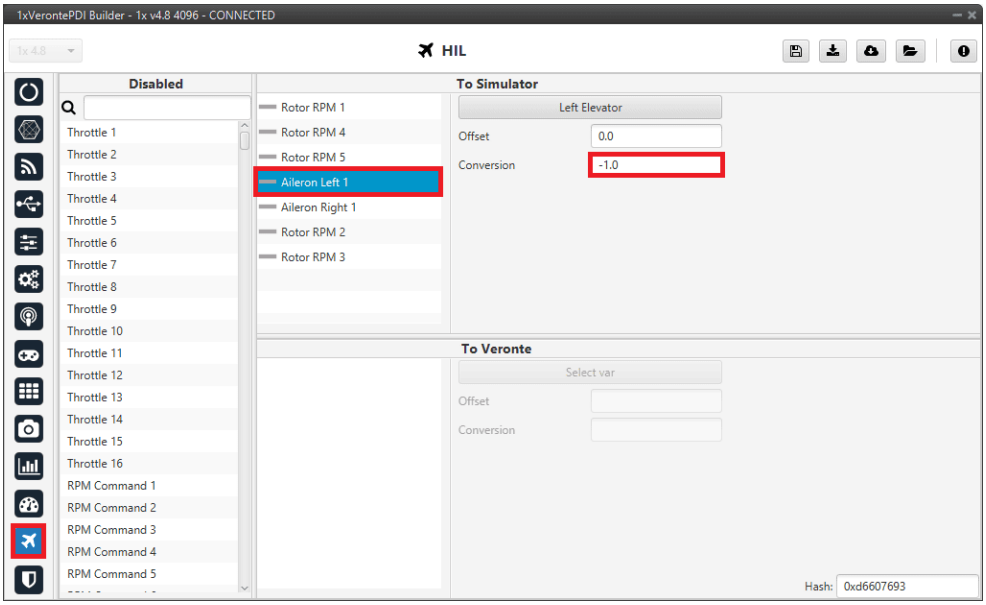
- Rotor RPM 1 ⇒ Motor 1 XPlane
- Rotor RPM 2 ⇒ Motor 2 XPlane
- Rotor RPM 3 ⇒ Motor 3 XPlane
- Rotor RPM 4 ⇒ Motor 4 XPlane
- Rotor RPM 5 ⇒ Pusher motor Xplane
- Aileron Right 1 ⇒ Right Elevator
- Aileron Left 1 ⇒ Left Elevator




**Autopilot 1x variables**

3. Set the **Offset** and **Conversion** variables if necessary.

   To adjust the aircraft performance to the simulation, a **conversion** factor of **- 1.0** has been defined for both **elevators**. This factor allows the simulation to match the actual aircraft conditions.



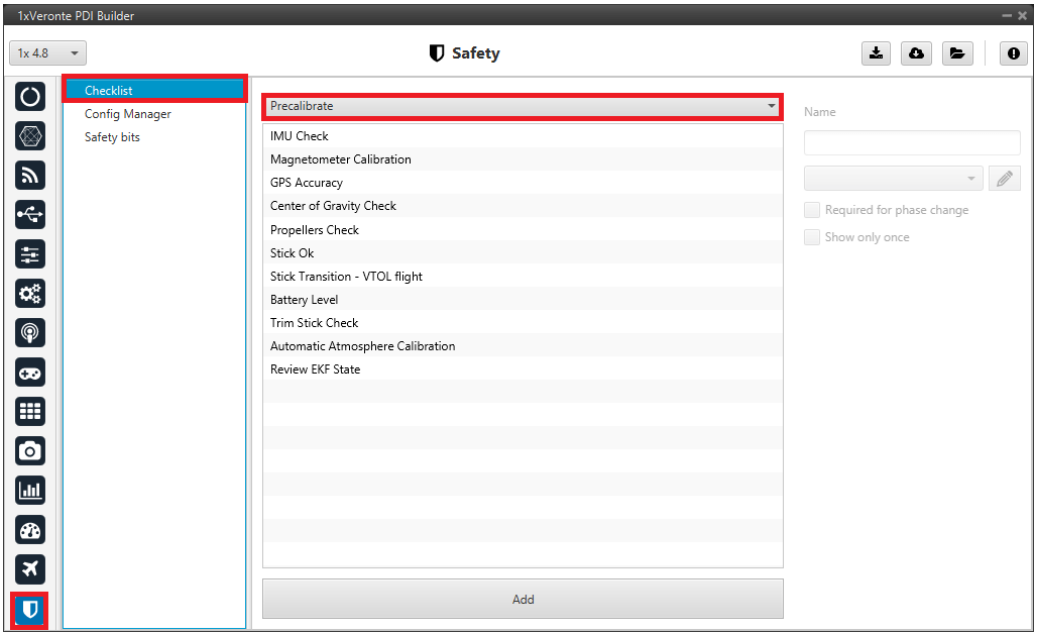

**Conversion - Right elevator**

For more information on this menu, visit the HIL section of the **1x PDI Builder** user manual.

## Checklist

This feature is used to ensure that certain requirements have been accomplished, for example, prior to a phase change or to avoid a possible malfunction.

The checklist configured here will appear in **Veronte Ops** in the **Checklist** widget. For more information on this, visit Veronte Ops manual.




**Checklist panel**

In this configuration, a basic Checklist has been defined. In it, 2 different types of checks are configured, checks for operator information only (None) and checks automatically performed by Veronte Autopilot 1x (In Range check).

In the **Precalibrate** phase, the following checks are established:

- IMU Check ⇒ None
- Magnetometer Calibration ⇒ None
- GPS Accuracy ⇒ None
- Center of Gravity Check ⇒ None
- Propellers Check ⇒ None
- Stick Ok ⇒ **In Range Check**
    - Check performed automatically if the Stick Not Detected variable is in the range [0.9 - 1.1].

- Stick Transition - VTOL flight ⇒ **In Range Check**
  - Check performed automatically if the Stick - Transition variable is in the range [0.0 - 0.4].
- Battery Level ⇒ **In Range Check**
  - Check performed automatically if the Power Input variable is in the range [23.8 - 26.0].
- Trim Stick Check ⇒ None
- Automatic Atmosphere Calibration ⇒ None
- Reviw EKF State ⇒ None

For more information on the Checklist options, visit the Safety - Configuration section of the **1x PDI Builder** user manual.

Operation

This section will explain how to design a basic operation in **Veronte Ops**, as well as the different automations and events that can be combined to allow Autopilot 1x to guide the aircraft according to the desired laws.

Veronte Ops is an application for vehicle **operation and monitoring** during missions. For more information, visit the Veronte Ops user manual.

To familiarize the user with the basic functioning of this tool, 3 operations will be explained as examples. Previously, the following steps must be followed to start designing an operation:

1. Download, install and open **Veronte Ops**.

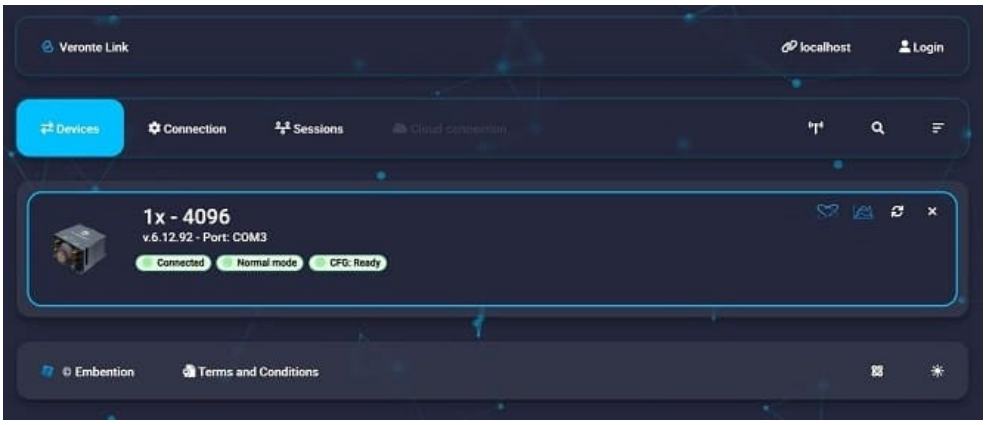
   There are 2 ways to work with this application: as a website app or by installing it throught an executable.

   > **Hint**
   > It is recommend to use the **executable** version if working **offline**.

   For more information, visit the Download and Installation - Quick Start section of the **Veronte Ops** user manual.
2. Connect Autopilot 1x to the PC (usually via USB) for design operation.

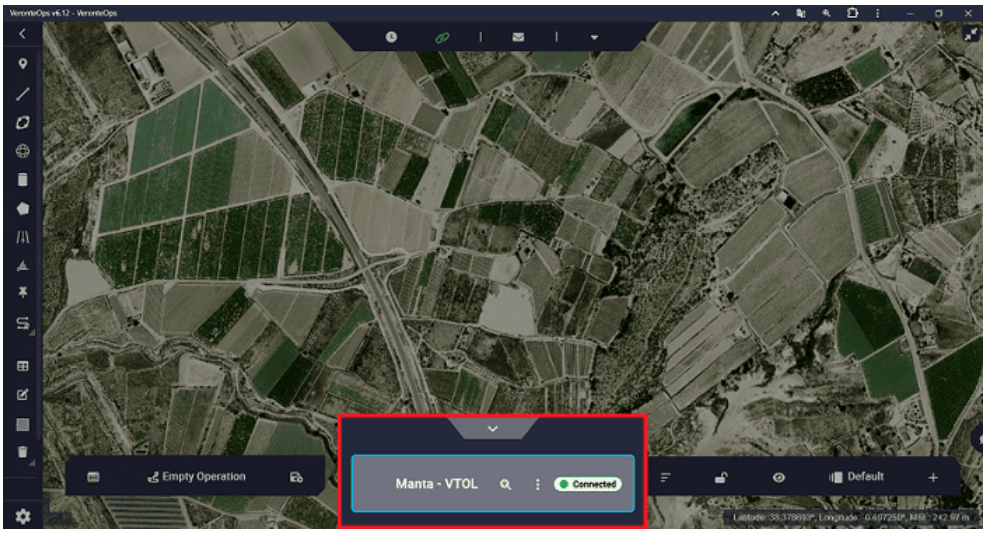
   For more information on this connection, visit the Basic connection for operation - Quick Start of the **1x Hardware Manual**.

3. In order to work with the autopilot with Veronte applications, it is first necessary to configure **Veronte Link** to recognize it. If users need more details on how do this, visit the Serial connection - Integration examples section of the **Veronte Link** user manual.




**Operation - Veronte Link**

4. Once the autopilot is recognized in **Veronte Link**, it will appear in Veronte Ops to be selected:




**Operation - Platform selected**

For more information, visit the Platform - Panels section of the **Veronte Ops** user manual.

5. Finally, navigate on the map to the location where the mission is to be created.

Followed these steps, the following three operations are defined:

- Operation 1. This is a simple operation, in which the aircraft performs a mission consisting of a closed loop passing through all flight phases.

- Operation 2. In this operation, the different options offered by the **areas tools** (Sphere, Cylinder, Prism) will be used. These can be linked to an automation or defined as **obstacles**.
- Operation 3. This operation combines the use of the **marks** option and automations to create an operation composed of two missions.

Once the operation is configured, it is necessary to **save and upload** it to Autopilot 1x. To do that, users must use the following options available in the **Operation Panel**:

**Operation panel actions**

1. **Save Operation**: Click on it to apply any change if necessary.
2. **Upload to**: Click on it to update the loaded operation in Autopilot 1x with the new **saved changes**.

> **Note**
> This option will only be available if the changes made have been previously saved, i.e. the **Save Operation** option must have been used.

The following message indicates that the operation configuration has been uploaded successfully.

**Configuration Saved**

For more information, visit the Operation actions - Operation section of the **Veronte Ops** user manual.

Finally, to start flying, it is useful to create a **Workspace** to control the mission. The Workspace section of this manual presents the main elements advised for setting up a workspace.

For more information, visit the Workspace - Panels section of the **Veronte Ops** user manual.

## Operation 1

This operation will be defined as a closed route in figure of `8`. It is a simple and basic mission that will serve as an example to learn how to use the Veronte Ops application.

### Mission

Once the desired location for the mission has been selected, there are two ways to create an `8` figure route:

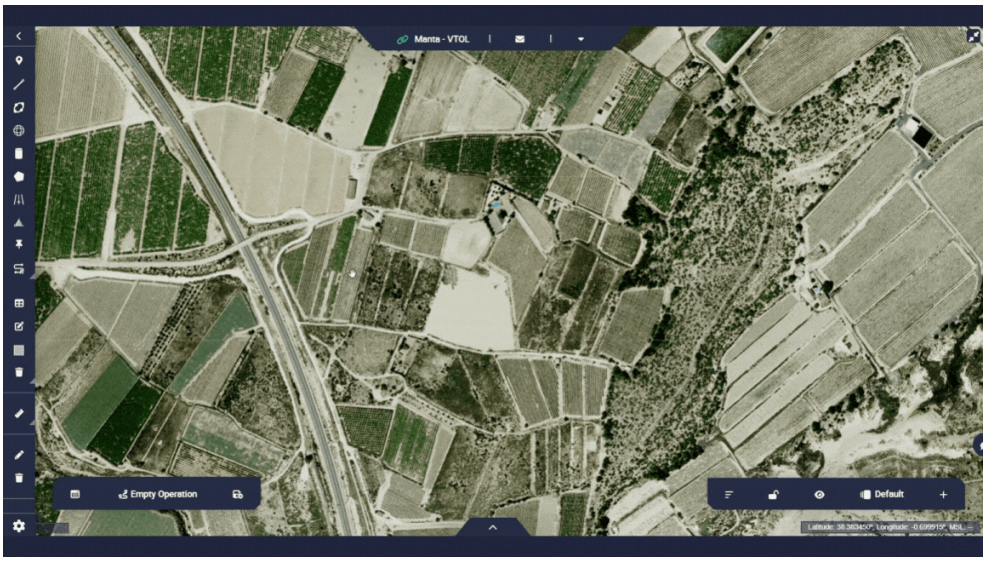
- Generate route option.
- Segments option.

#### Generate route

This option allows to create missions with predefined routes: **Spiral**, **Photogrammetry** and **Figure**.

Follow the steps below to define the mission using the Generate route option:

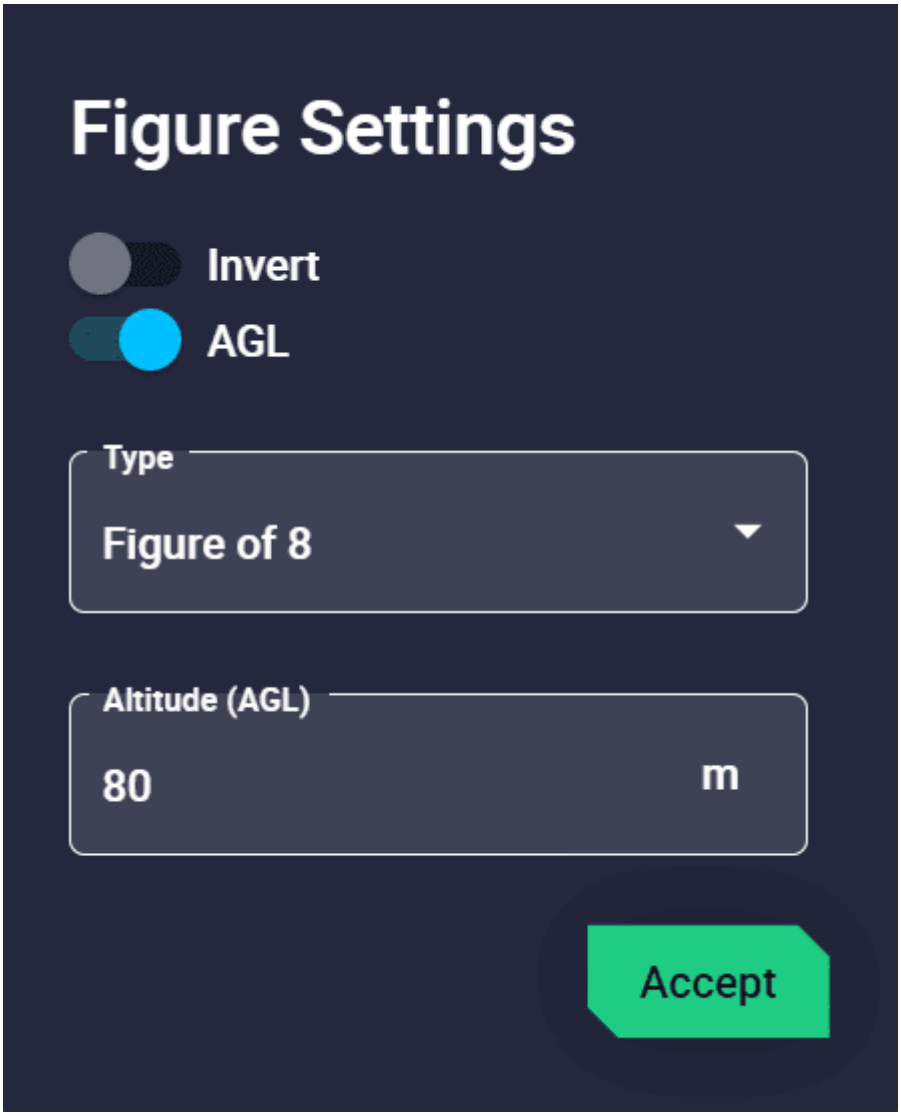
1. Go to Mission Toolbar → Generate route → **Create figure**.
   **Click and drag to create the figure** in the desired location.



**Operation 1 - Create Figure**

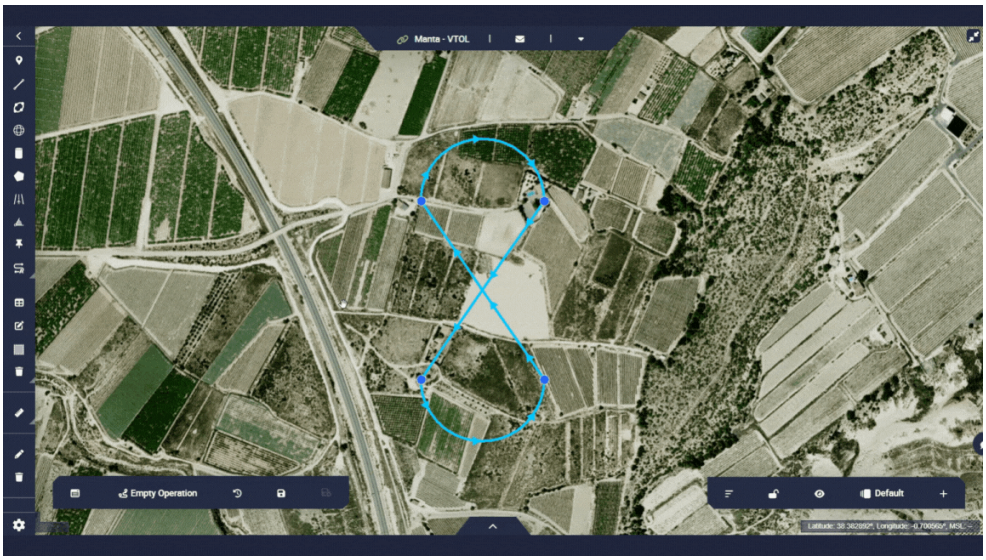
2. Configure the **Figure Settings**.

   For more information on the parameters to be configured, visit the Generate route - Operation section of the **Veronte Ops** user manual.




**Operation 1 - Figure Settings**

3. Use the Edit Mission option in the Mission Toolbar if the user wants to edit the location of a waypoint or the curvature of the route.




**Operation 1 - Edit Mission**

### Segments

This option allows the user to define the location and altitude of each waypoint, thus defining the mission in more detail.

Follow the steps below to define the mission using Segments.

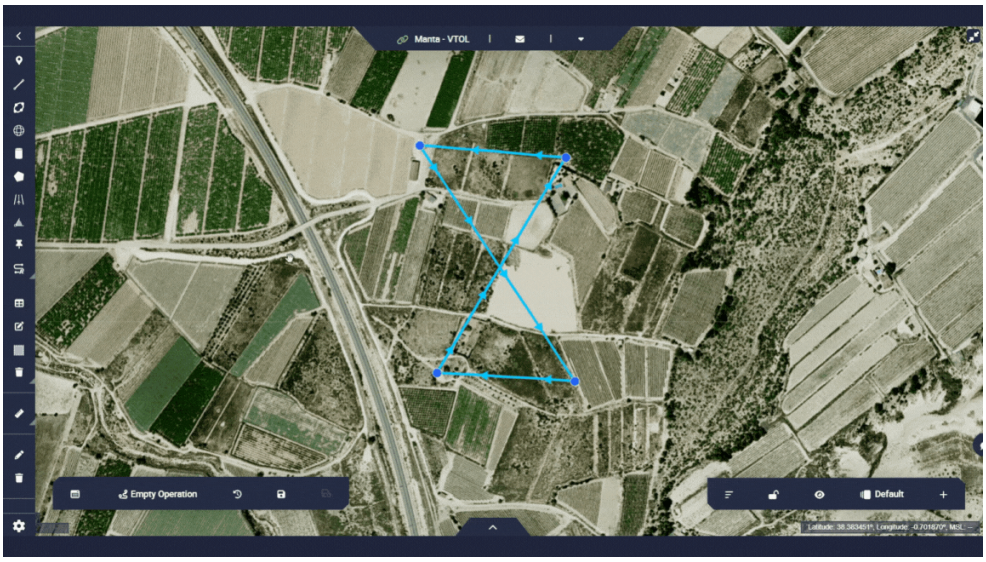
1. Go to Mission Toolbar → **Segment**.

   Set as many segments as wished. These segments, separated by waypoints, will draw the mission.

   > **Hint**
   > Set the segments following the desired direction of the route.

**Operation 1 - Segment**

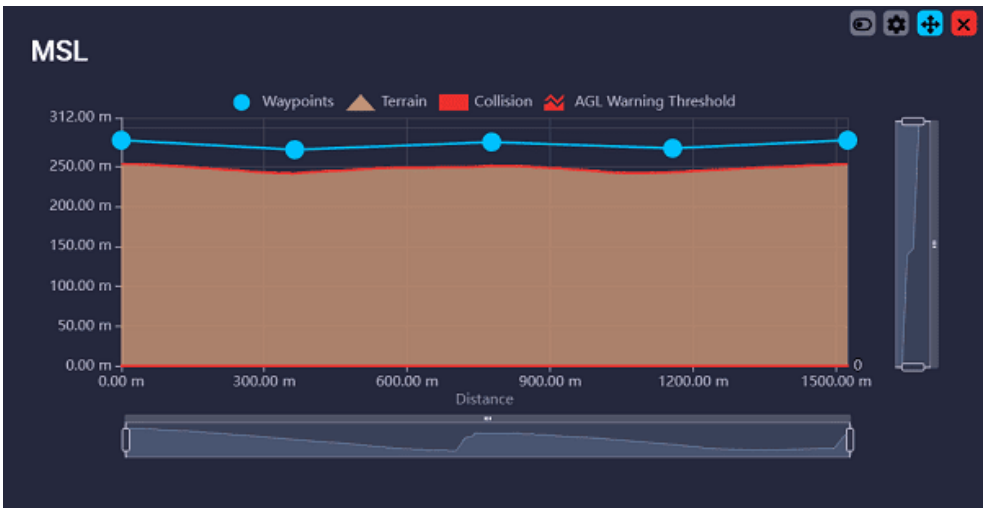
2. Use the Edit Mission option in the Mission Toolbar if the user wants to edit the location of a waypoint and the curvature of the route.




**Operation 1 - Edit Mission**

3. Check the altitude of the route waypoints with respect to the ground, **to make sure that the route is feasible**, i.e. does not interfere with the terrain.

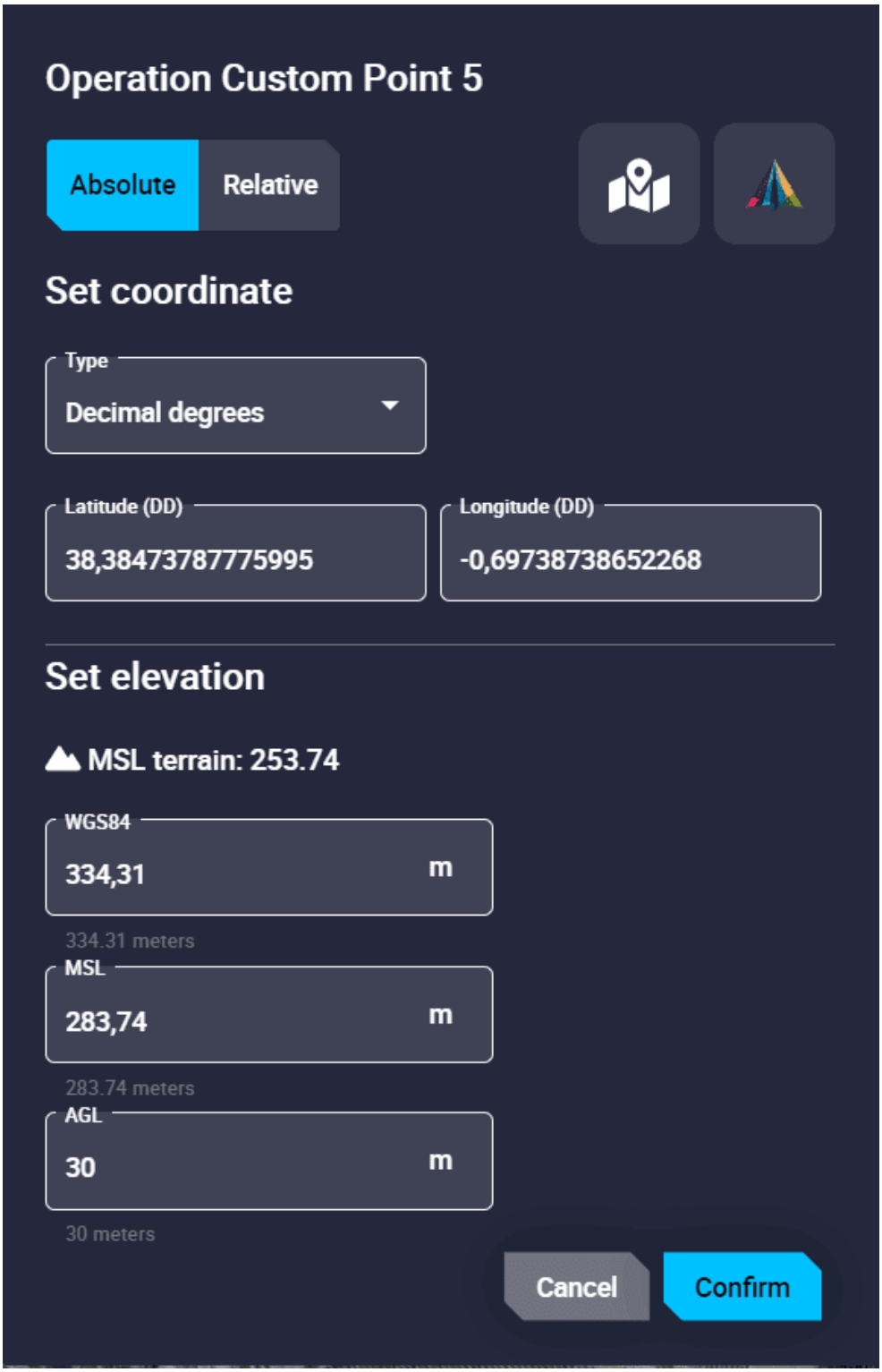
   To do this, right-click on any waypoint of the route → **Open elevations**.




**Operation 1 - Elevations panel**

4. Configure the settings for each waypoint to define its altitude and location.

   To do this, right-click on a waypoint → **Waypoints Settings**.




**Operation 1 - Waypoints Settings**

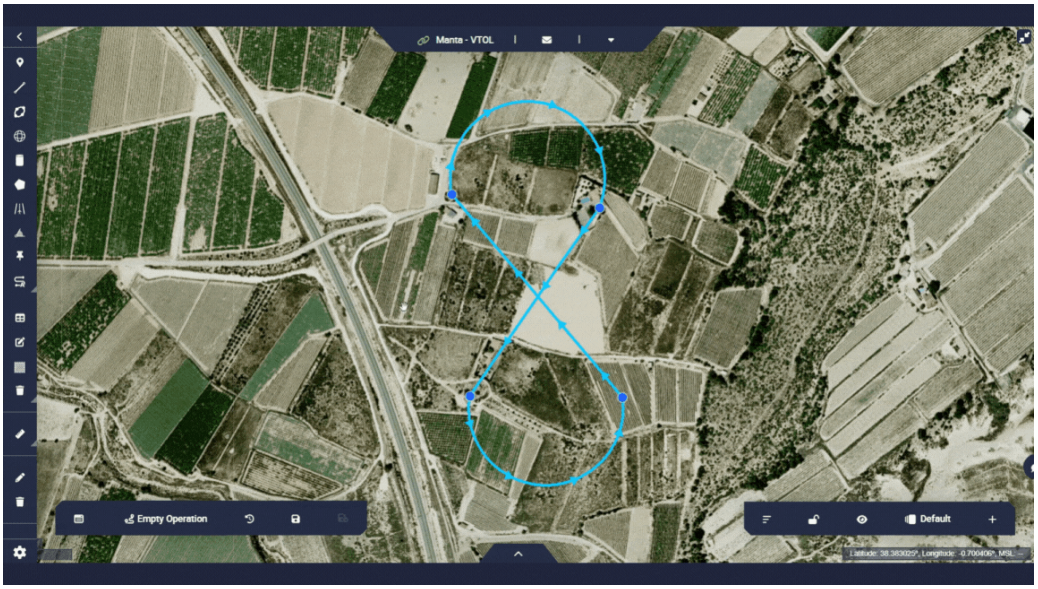
For more information on the parameters to be configured, visit the Waypoint - Operation section of the **Veronte Ops** user manual.

Start Route

The user must define a waypoint as the start of the route. This will be the point where the aircraft will start the Cruise phase.

> **Important**
>
> If this point is not defined, **Veronte Ops** will not allow the operation to be saved.

To define a waypoint as the start of the route, right click on the desired waypoint to be the start of the route → Select the **Set Start Route** option:



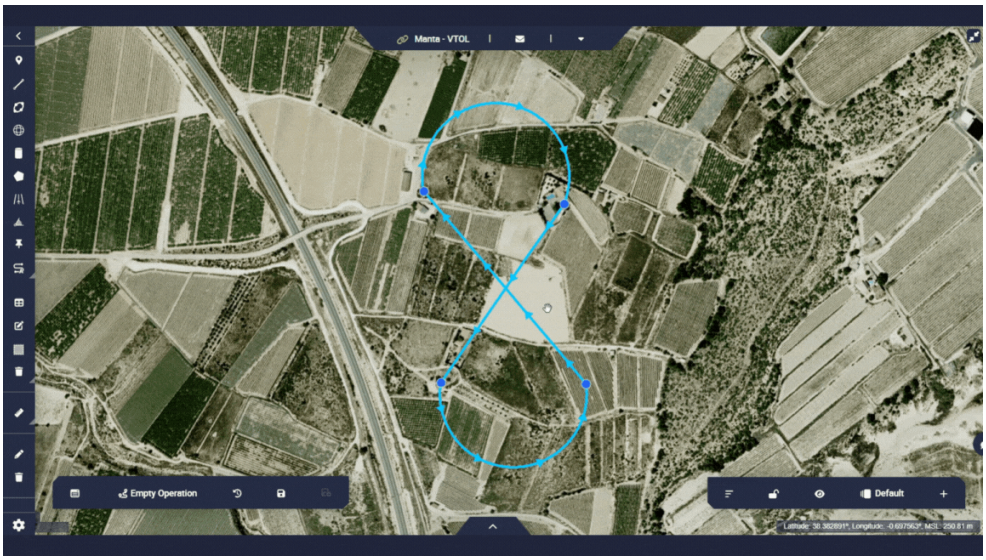

**Operation 1 - Start Route**

Runway

For the operation of the Embention VTOL aircraft it is necessary to define a runway since in the Approach phase a route will be generated to guide the aircraft to fly to it.

Follow the steps below to define a runway:

1. Go to Mission Toolbar → **Set a Runway**.
2. Click on the map to define the start point of the runway → Click on a second point to define the end of the runway.
3. Use the Edit Mission option to locate the Loiter point where the user wants it to be.
   This will be the point around which the Loiter will take place.




**Operation 1 - Runway**

4. Go to Operation Panel → Customize → **Runway**.

   Link the runway defined in the **1x PDI Builder** configuration with the just created runway.

**Operation 1 - Set runway**

> ⓘ **Note**
>
> Adjust the runway parameters so that the aircraft performs the desired operation.
>
> For more information on the parameters to be configured, visit the Runway - Operation section of the **Veronte Ops** user manual.

Home Point

The Home point will be the point used to execute the **Compute RTH** program of the **1x PDI Builder** configuration. This will be the point to which the Autopilot 1x will guide the aircraft in the Return to Home phase.

Follow the steps below to define a waypoint as Home point:

1. Go to Mission Toolbar → **Waypoint**.
2. Click on the map to set a waypoint.

> ⓘ **Note**
>
> The user also has the option to use an existing waypoint.

3. Go to Operation Panel → Customize → **Custom Points**.

   Link the Home point defined in the **1x PDI Builder** configuration with the desired waypoint.

**Operation 1 - Home Point**

> ⓘ **Note**
>
> If the user does not define this point, Autopilot 1x will guide the aircraft to the **take-off point** in the Return Home phase.

## Operation 2

This operation implements some of the main functionalities of the **areas** option. Generally, these areas are used as **events** that trigger an action, or as **obstacles** that the guidance program will avoid flying over.

As an example, two areas will be added to the Operation 1 mission:

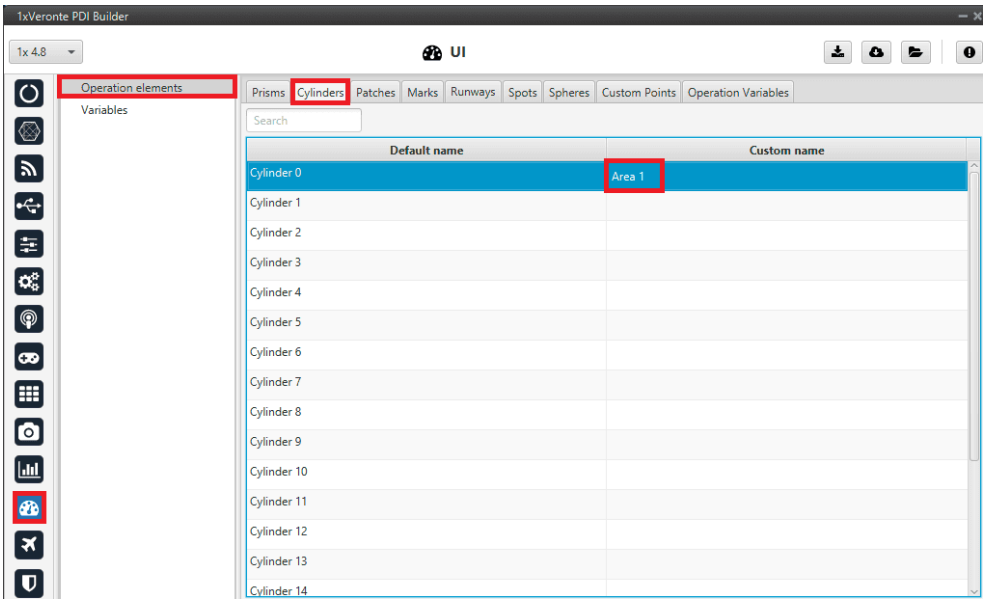
- Area 1 → Within this area the aircraft will switch directly to the **Return to Home** phase.
- Area 2 → Area defined as **obstacle**, the autopilot system will prevent the aircraft from entering this area.

> **Important**
>
> To configure the **areas with events and actions** in **Veronte Ops**, it is necessary to first define them in the Operation elements panel of the **UI menu** of the **1x PDI Builder** app.

> **Note**
>
> In this case, the areas used will be cylinders, although they can also be prisms or spheres following the same steps.

### Area 1

1. In the 1x PDI Builder app → go to UI menu → Operation elements panel → **Cylinders tab**.
   Define the area by setting a **custom name**:

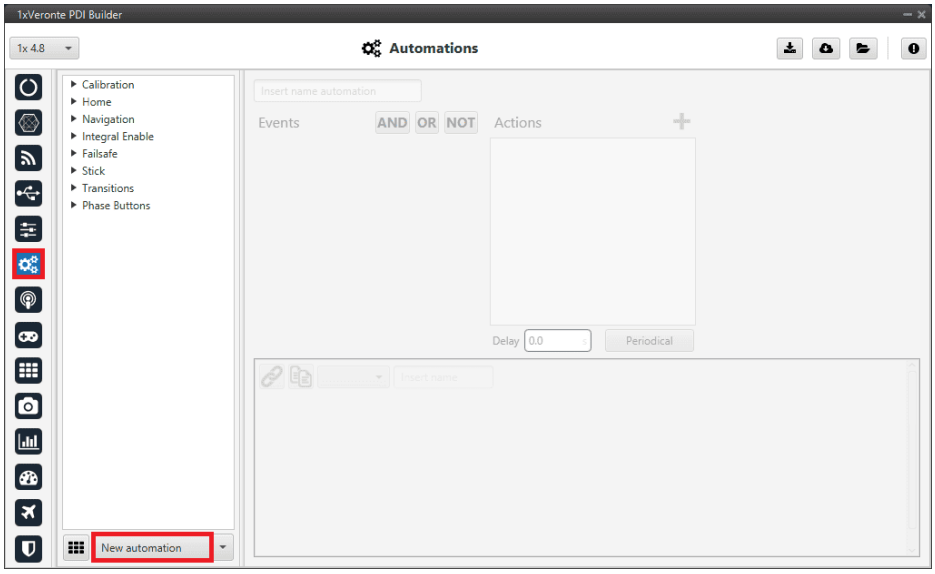



**Operation 2 - UI menu**

2. In the 1x PDI Builder app → go to **Automations menu**.

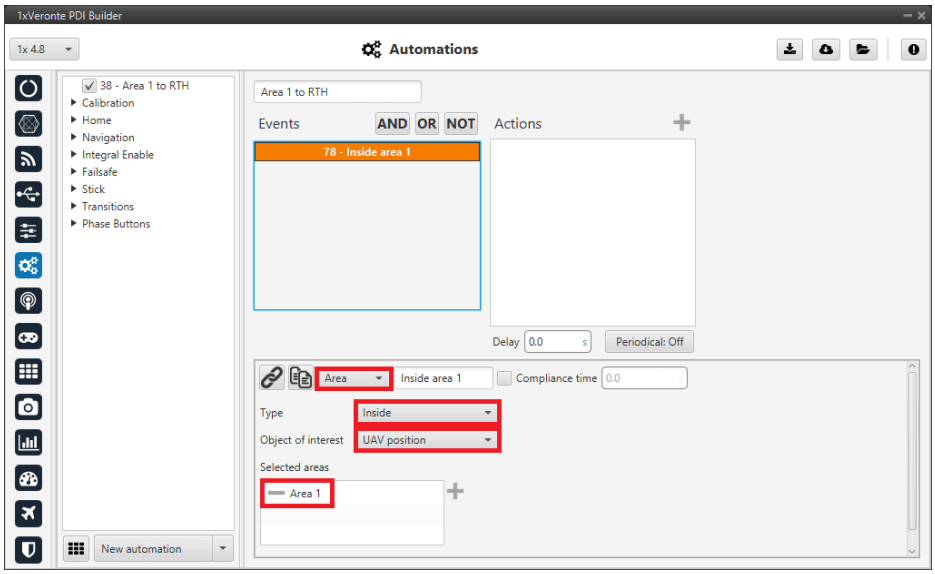
   An automation will be created for the autopilot to pass to the Return to Home phase when it is inside Area 1:

   - Create a **new automation**.



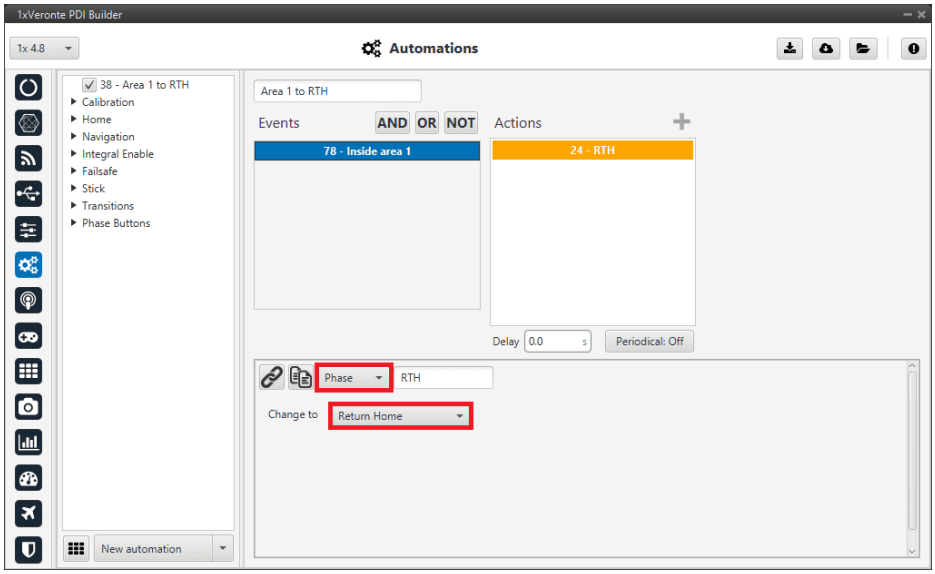

**Operation 2 - New Automation**

   - To configure the desired event, select the **Area** as the event type and choose the **Inside** type, since the event is that the **aircraft is inside area 1**.




**Operation 2 - Event**

- Define an **action** of type **Phase**. In this case, the action is to change to Return Home phase.




**Operation 2 - Action**

For more information, visit the Automations - Configuration section of the **1x PDI Builder** user manual.

3. Save the configuration so that the changes made will appear in **Veronte Ops**.
4. Create the area in **Veronte Ops** (in the mission created for Operation 1). For this:

   In Veronte Ops → go to Mission Toolbar → **Set a Cylinder**.

   **Click and drag to set a cylinder** to the desired position.

5. Use the Edit Mission option of the Mission Toolbar if the user wishes to edit the exact location and radius of the cylinder.

**Operation 2 - Cylinder**

6. Configure the cylinder by setting its height and radius.

   For more information on the parameters to be configured, visit the Cylinder - Operation section of the **Veronte Ops** user manual.

7. Go to Operation Panel → Customize → Areas → **Cylinders**.

   Link the area defined in the **1x PDI Builder** configuration to the just created cylinder.

**Operation 2 - Set cylinder**

Area 2

1. Create the area in **Veronte Ops** (in the mission created for Operation 1).
   For this:
   In Veronte Ops → go to Mission Toolbar → **Set a Cylinder**.
   **Click and drag to set a cylinder** to the desired position.
2. Use the Edit Mission option of the Mission Toolbar if the user wishes to edit the exact location and radius of the cylinder.
3. Right click on the configured cylinder → Select **Set Obstacle** option.

**Operation 2 - Obstacle**

4. Configure the cylinder by setting its height and radius.
   For more information on the parameters to be configured, visit the Cylinder - Operation section of the **Veronte Ops** user manual.

> ⓘ **Note**
>
> It is also possible to create an automation associated to an obstacle. To do so, once the area has been defined as an obstacle, follow the steps explained in Area 1 to create the automation.

## Operation 3

This operation implements some of the functionalities offered by the combination of the **Automations** with the **Mark** option of the Mission Toolbar.

To explain this tool, an operation composed of two missions with closed circuits is created.

In the first mission, which is the same as the one created for Operation 1, the **Mark A** is defined. Then, when the aircraft achieves this mark an automation will send the aircraft to move to the second mission. In this second mission, the **Mark B** will be defined from which the aircraft will start the **Approach** phase.

Missions

As mentioned above, two closed circuits are defined:

- The first circuit with the figure of `8`.
- The second circuit, at the same altitude as the first one, with the figure of `Racetrack`.

**Operation 3 - Missions**

Mark A

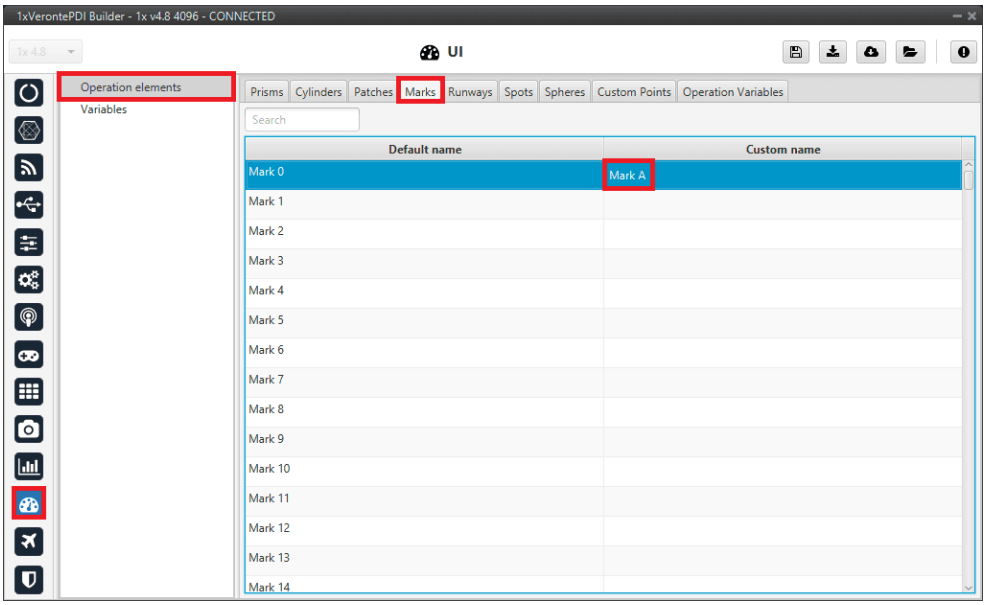
The operation starts with the route of the first mission, therefore, a point of this circuit is defined as the Start Route. At one point on this route the **Mark A** is defined.

Follow the steps below to define the mark and the automation associated with it:

1. First it is required to define the mark in the **1x PDI Builder** configuration.

In the 1x PDI Builder app → go to UI menu → Operation elements panel → **Marks tab**.

Define the mark by setting a **custom name**.

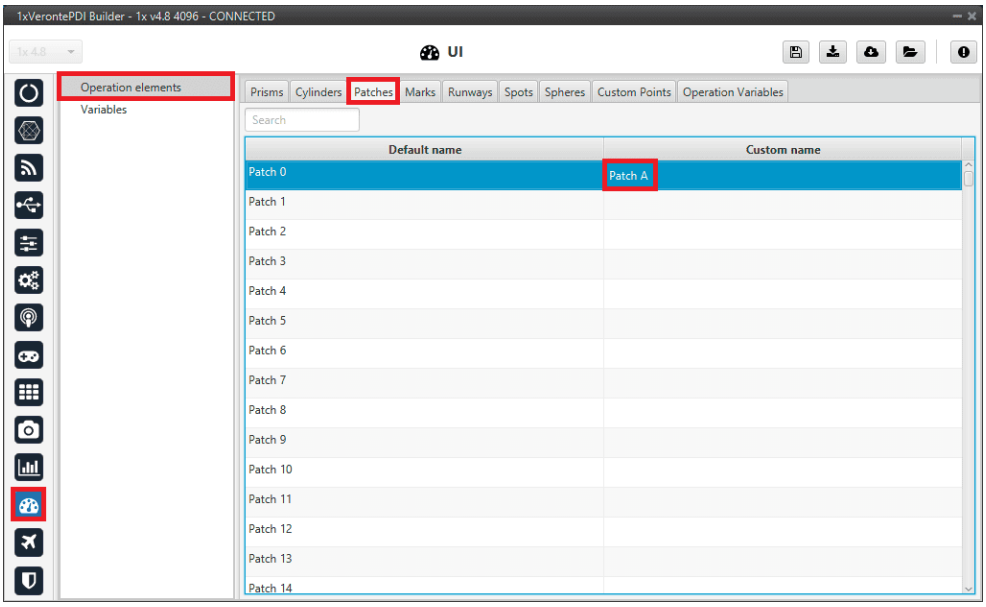



**Operation 3 - Mark A definition**

2. It is also necessary to define the **patch** where the autopilot will guide the aircraft when it reaches **Mark A**.

   In the 1x PDI Builder app → go to UI menu → Operation elements panel → **Patches tab**.

   Define the patch by setting a **custom name**.



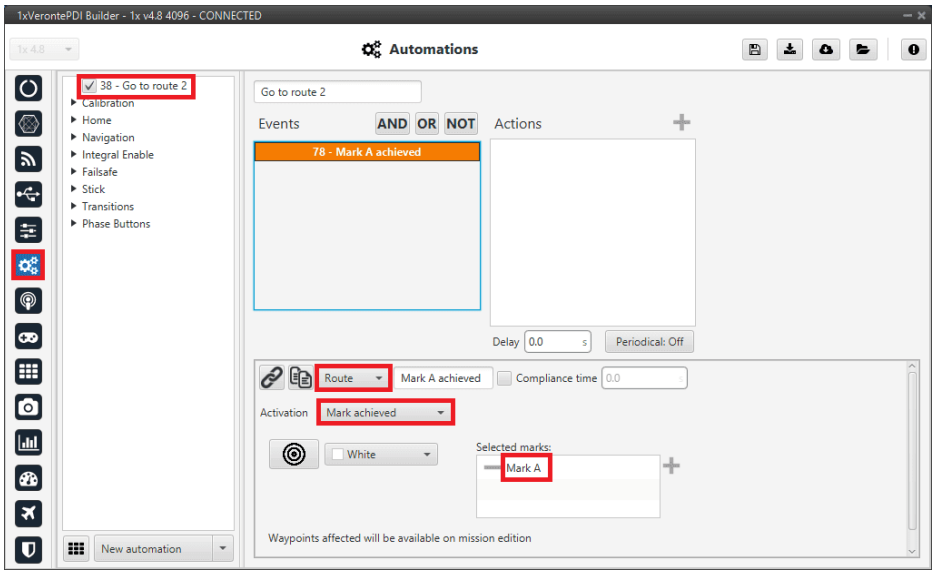

**Operation 3 - Patch A definition**

3. In the 1x PDI Builder app → go to Automations menu → **New automation**.

An automation will be created so that, once this mark is achieved, the aircraft **will go to** the desired point:

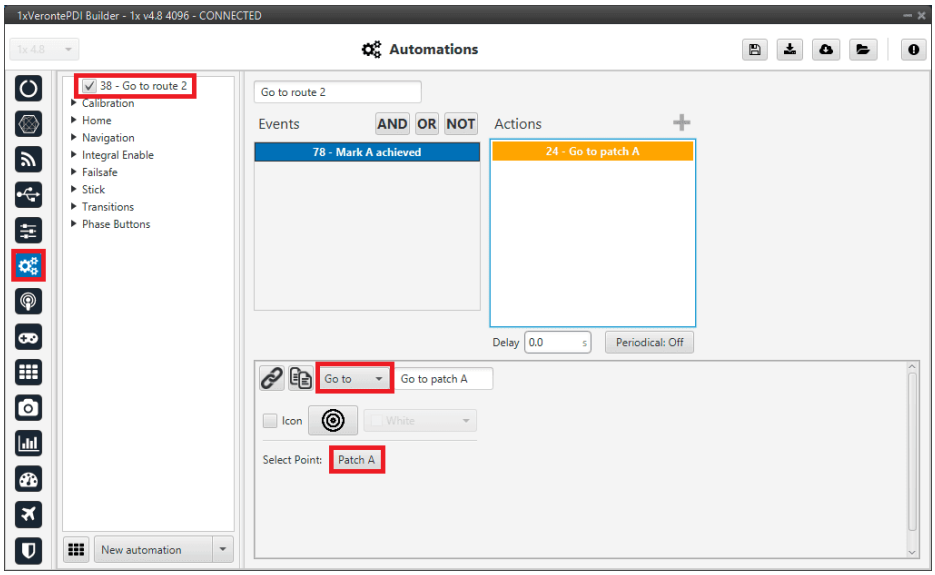
- To configure this event, select the **Route** type and choose the **mark achieved** option.

  For more information, visit the Route - Events section of the **1x PDI Builder** user manual.




**Operation 3 - Mark A event**

- To define the action triggered by the event, set up a **Go to** action. This action guides the aircraft to the selected **patch**. For more information, visit the Go to - Actions section of the **1x PDI Builder** user manual.




**Operation 3 - Mark A action**

4. Save the configuration so that the changes made will appear in **Veronte Ops**.

5. Create the mark in **Veronte Ops**. For this:

   In Veronte Ops → go to Mission Toolbar → **Mark**.

   Click on the desired patch to place a mark on the route.

   Then, with the Edit Mission option of the Mission Toolbar, users can more precisely place the mark.

**Operation 3 - Mark A creation**

6. Go to Operation Panel → Customize → **Marks**.

   Link the mark defined in the **1x PDI Builder** configuration with the just created mark.

**Operation 3 - Set Mark A**

7. Go to Operation Panel → Customize → **Patches**.

Link the patch defined in the **1x PDI Builder** configuration to the desired patch where the autopilot will guide the aircraft. In this case, a patch from the second mission.

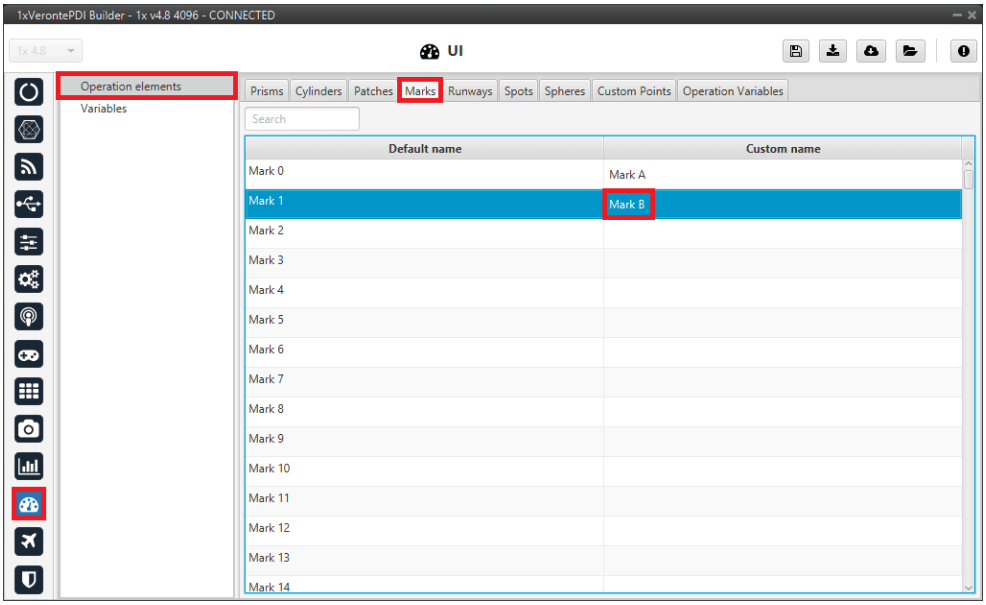
**Operation 3 - Set Patch A**

Mark B

Follow the steps below to define the **Mark B**:

1. In the 1x PDI Builder app → go to UI menu → Operation elements panel → **Marks tab**.

   Define the mark by setting a **custom name**.



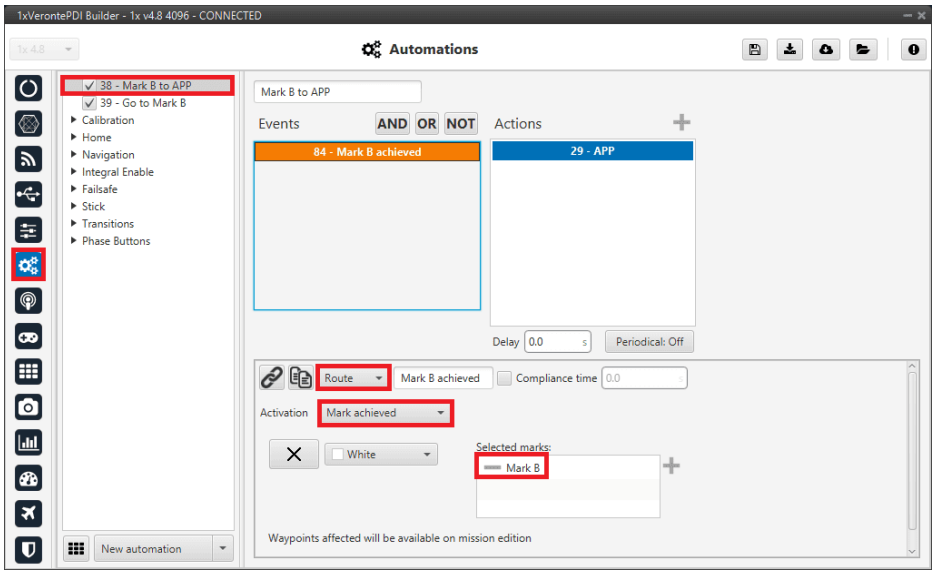

**Operation 3 - Mark B definition**

2. In the 1x PDI Builder app → go to Automations menu → **New automation**.

An automation will be created so that, once this mark is achieved, the aircraft begins its **approach** phase.

- To configure this event, select the **Route** type and choose the **mark achieved** option.

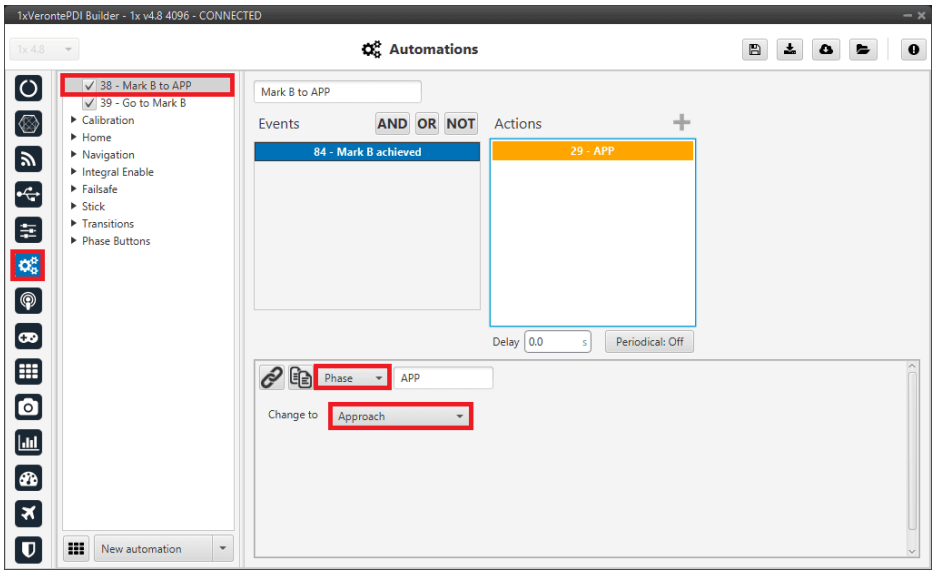
  For more information, visit the Route - Events section of the **1x PDI Builder** user manual.




**Operation 3 - Mark B event**

- To define the action triggered by the event, configure a **Phase** action and select the phase to change, in this case to the Approach phase.

  For more information, visit the Phase - Actions section of the **1x PDI Builder** user manual.



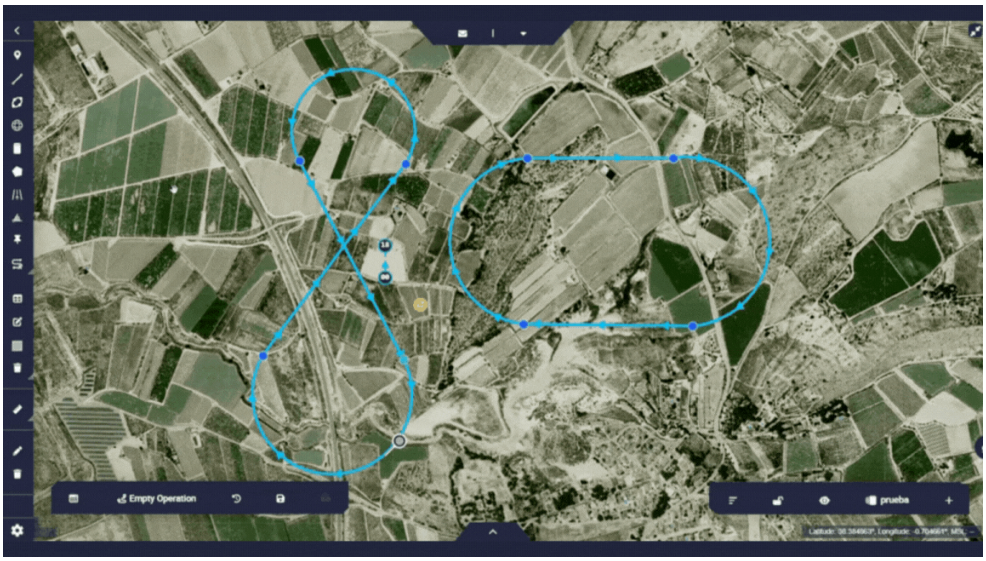

**Operation 3 - Mark B action**

3. Save the configuration so that the changes made will appear in **Veronte Ops**.
4. Create the mark in **Veronte Ops**. For this:

   In Veronte Ops → go to Mission Toolbar → **Mark**.

   Click on the desired patch to place a mark on the route.

   Then, with the Edit Mission option of the Mission Toolbar, users can more precisely place the mark.




**Operation 3 - Mark B creation**

5. Go to Operation Panel → Customize → **Marks**.

   Link the mark defined in the **1x PDI Builder** configuration with the just created mark.

**Operation 3 - Set Mark B**

## Workspace

The creation of a customized workspace allows the user to display the desired information to monitor the operation.

**Workspace example**

For more information, visit the Workspace - Panels section of the **Veronte Ops** user manual.

This section explains the **main** settings and **widgets** that are recommended be incorporated in the workspace to perform a **basic** flight.

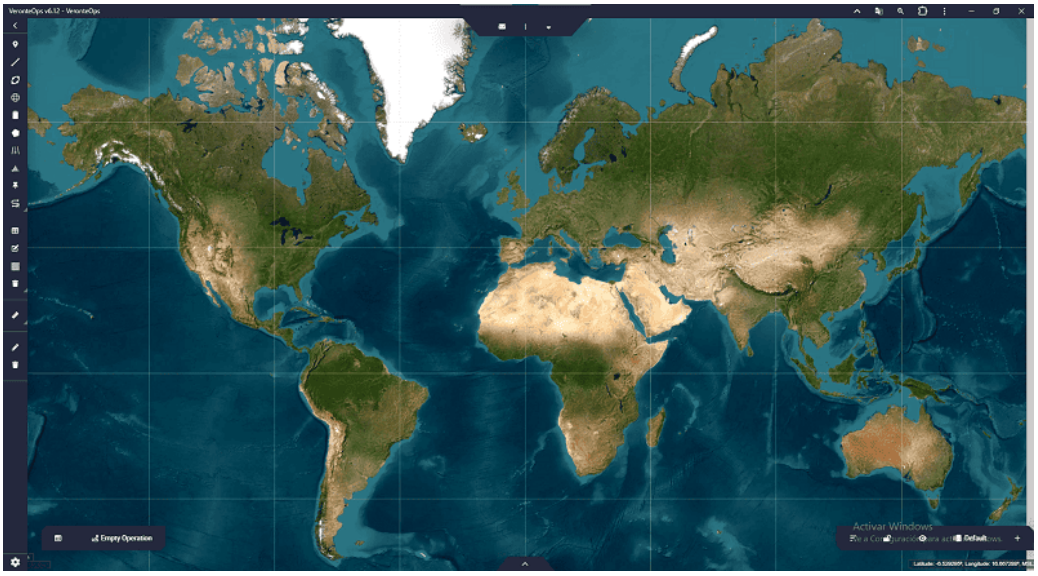
> **Note**
>
> This workspace is designed specifically for the operation being performed, so not all the widgets explained below are necessarily useful for all operations.

> **Important**
>
> The widgets explained are not mandatory, each user should configure the workspace by adding the widgets that provide the best information about the operation.

Map

Veronte Ops offers the possibility to choose the map on which users want to operate. Users can choose from the maps provided by Veronte Ops or upload their own map.

In this operation, the **ESRI Satellite** map has been employed.



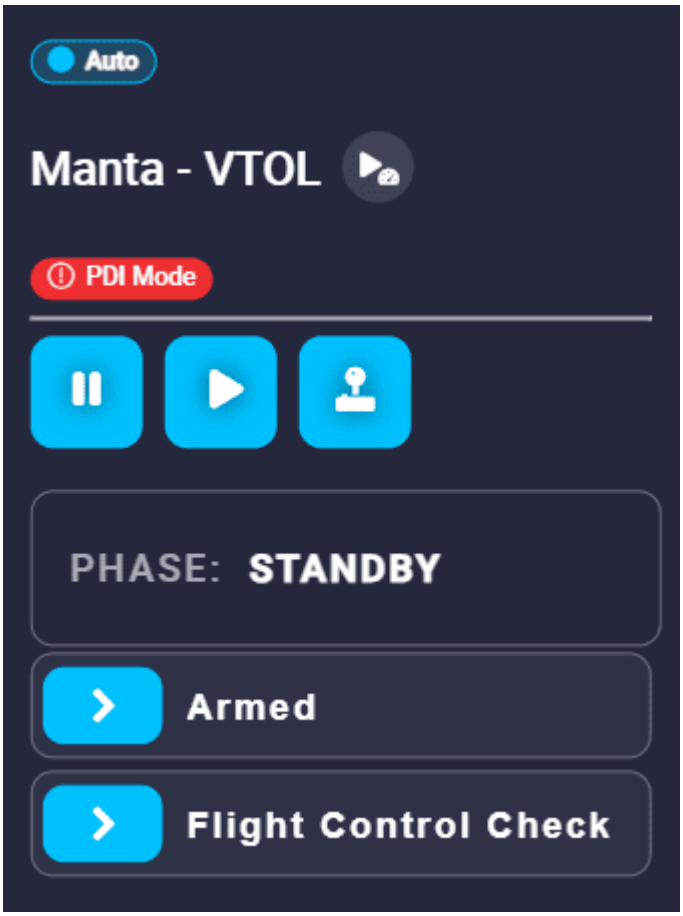

**Workspace - Map**

For more information, visit the Map option - Workspace section of the **Veronte Ops** user manual.

Main

Widgets from the "Main" category that have been used are presented below. For more information on these widgets, visit the Main - Widgets section of the **Veronte Ops** user manual.

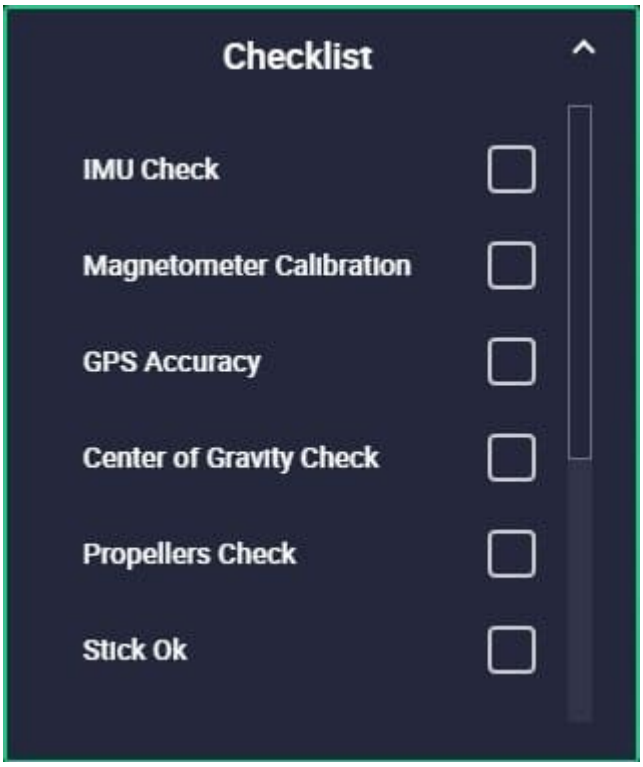
- Veronte Panel. This pannel is the basic operator tool. It includes all basic commands (phase and action buttons) and information needed during a standard mission. These commands can be triggered with a single click, by sliding and/or automatically.




**Workspace - Veronte Panel**

> **Important**
>
> **PDI Mode**: This only appers if PDI Mode is configured in the **1x PDI Builder** app.

- Checklist. This panel is used to make sure that some requirements have been met. This represents the checklist explained in Checklist - 1x PDI Builder configuration section.




**Workspace - Checklist**

> **Important**
> It must be previously defined in the Checklist panel of the **Safety menu** of the **1x PDI Builder** app.

Displays

These widgets allow the user to display in real time the value of a variable.

For more information, visit the Displays - Widgets section of the **Veronte Ops** user manual.

- Label. With labels the user can display the value of an **Integer variable**, **Real variable** or **Bit variable**. For a basic flight, the following labels are recommended:

| Variable | Label | Type |
|---|---|---|
| GNSS1 Accuracy | GNSS1 Accuracy: 5840307.000 m | Real variable |

| Variable | Label | Type |
|---|---|---|
| GNSS2 Accuracy | GNSS2 Accuracy: 5842059.500 m | Real variable |
| AGL | AGL:21.83 m | Real variable |
| MSL | MSL:272.01 m | Real variable |
| Position Fix | Position Fix | Bit variable |
| System Error | System Error | Bit variable |
| Sensors Error | Sensors Error | Bit variable |
| Stick not detected | Stick Not Detected | Bit variable |

It may also be interesting to display other variables in the labels:

- Stick-Pitch, Stick-Roll, Stick-Throttle, etc.
- Arc-Pitch, Arc-Roll, Arc-Throttle, Arc-Yaw
- Pitching (Q), Pitching (P), Rolling (Q), Rolling (P), etc
- Down Ground Velocity
- Static Pressure
- PDI error
- Power input

Charts

For more information, visit the Charts - Widgets section of the **Veronte Ops** user manual.

- Chart. Charts widgets allow the user to represent any variable of the system with respect to time. Thus, the user can visualize the evolution of that variable over time.

**Workspace - Charts**

For a **basic** flight, the following variables are recommended for charts:

| Variable 1 | Variable 2 |
|---|---|
| Desired Roll | Roll |
| Desired Pitch | Pitch |
| Desired Yaw | Yaw |
| Desired Ground Speed (GS) | Ground Speed (GS) |
| Desired Front GV | Front GV |
| Desired Down GV | Down GV |

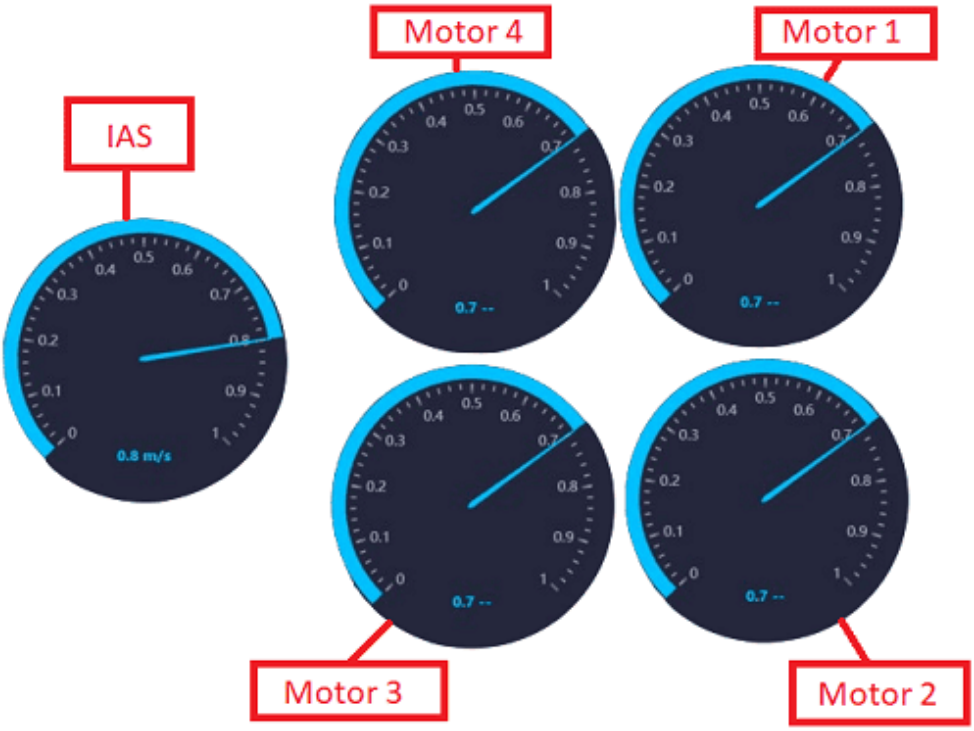
| Variable 1 | Variable 2 |
|---|---|
| Desired Lateral GV | Lateral GV |
| Desired Heading | Heading |
| Desired IAS | IAS |
| Desired FPA | Flight Path Angle (FPA) |

It may also be interesting to plot the following variables:

- Control mix
- Roll rate, Pitch rate, Yaw rate
- P, Q, R (Angular Velocity)

- Gauge. Gauge widget is a data visualisation tool that can be used to show the progress of data or display data in ranges in a precise and compact area.

  This widget can also be useful to represent variables such as **IAS** or **Motor RPM**:



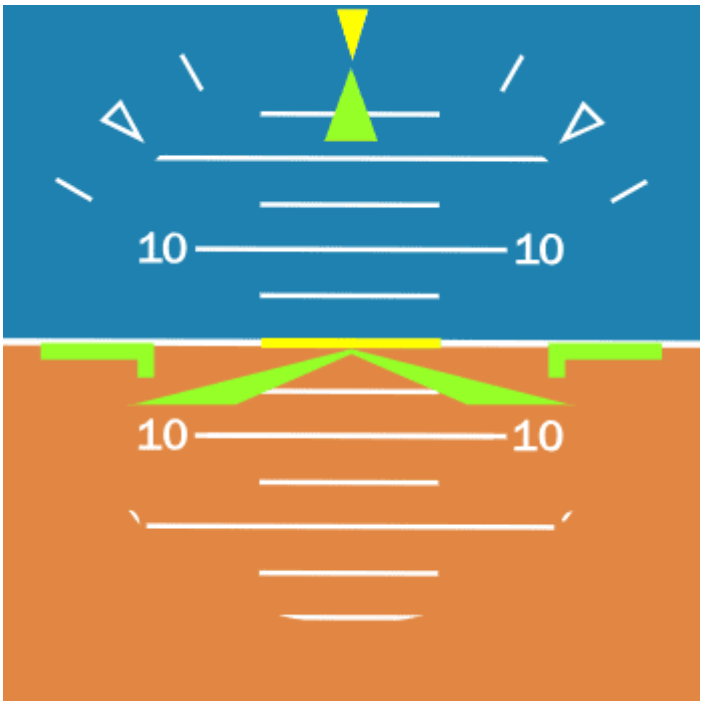
### Workspace - Gauge

Flight instruments

**Workspace - Flight instruments**

For more information, visit the Flight instruments - Widgets section of the **Veronte Ops** user manual.

- Attitude. The attitude widget, commonly known as Primary Flight Display (PFD) or 'artificial horizon', represents graphically the attitude of the aircraft (**roll** and **pitch**).



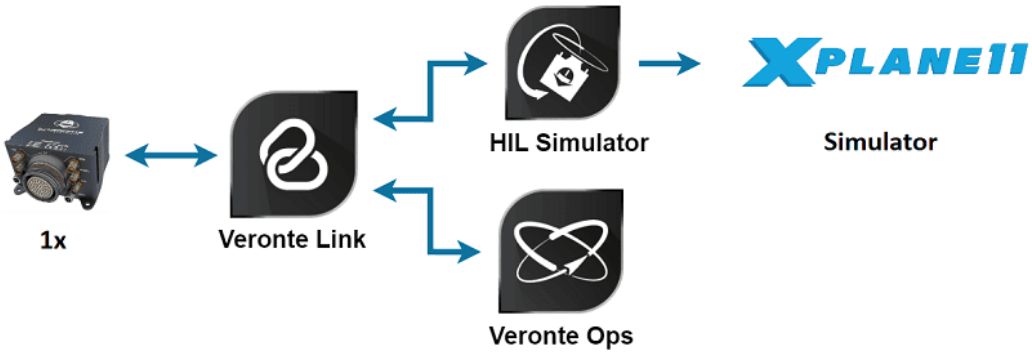

**Workspace - Attitude**

- Heading. The heading widget, commonly known as **compass**, usually shows the platform's yaw relative to the magnetic north.

**Workspace - Heading**

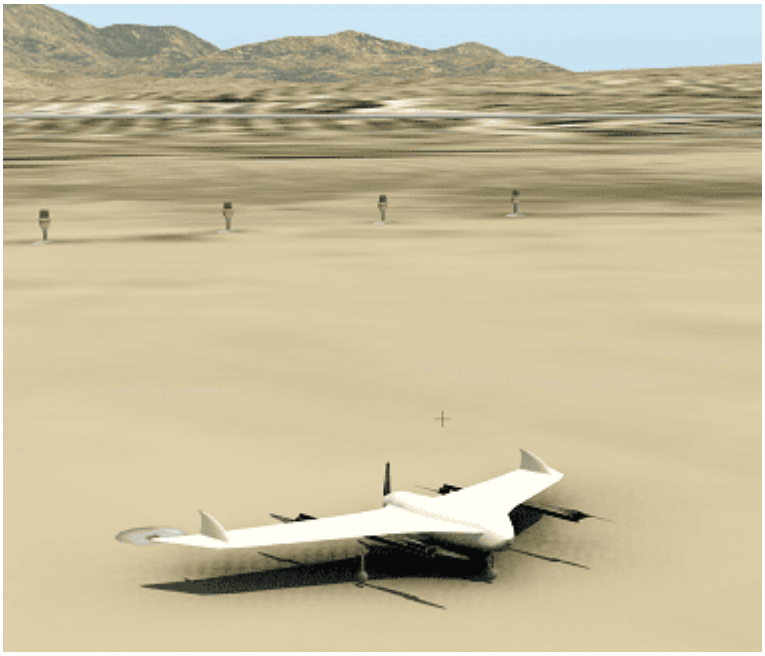
## Simulation

This section explains the steps to follow to simulate in **X-Plane 11** the operations defined in the Operation section.




The simulation will be performed using the HIL Simulator tool.

> **Important**
>
> To configure X-Plane correctly, follow the steps of X-Plane 11 section of the **HIL Simulator** user manual.



Once **X-Plane 11** has been correctly configured and the simulation has been started, in the following sections the user will be able to find the simulation of the operations:

- Operation 1
- Operation 2
- Operation 3

It also explains the Stick configuration to be performed in order to integrate and simulate the missions using a **stick**.

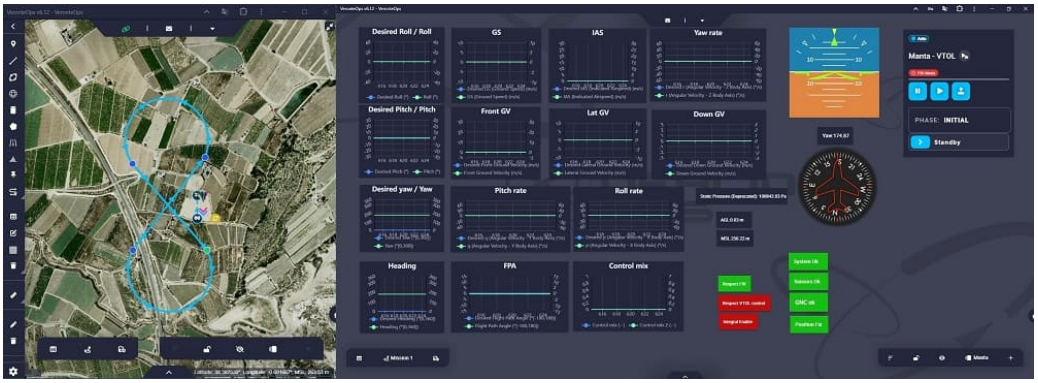
Operation 1

In this operation, the aircraft performs a mission consisting of a runway for the approach phase in Fixed-Wing (FW) flight configuration and a closed loop passing through all flight phases.

The performance of the aircraft in each of these phases and the transitions between phases that are triggered, are explained below.

## Initial and Standby

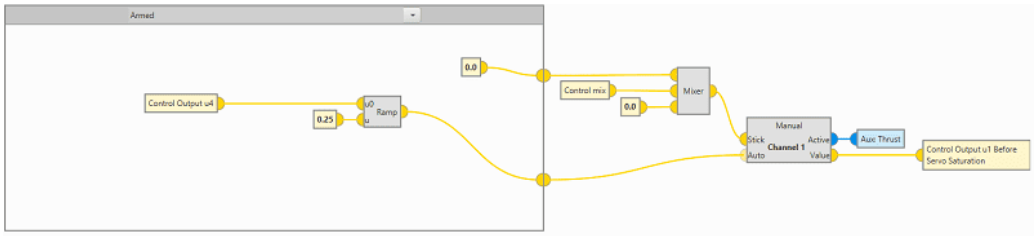
In these flight phases all actuators are disabled. The guidance program does not send any commands to the actuators, so the charts do not show any data.




**Operation 1 - Initial flight phase**

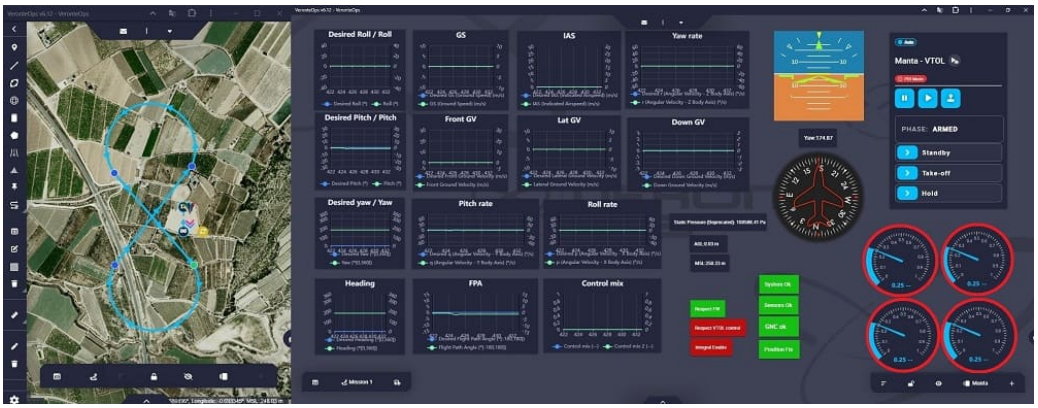
## Armed

In this flight phase, the four motors providing the thrust increase the RPM to **25%**. This increase, as explained, has been defined in the block programs.




**Operation 1 - Armed in Block Programs**

This activation of the motors can be easily visualized with the gauges added to the workspace.

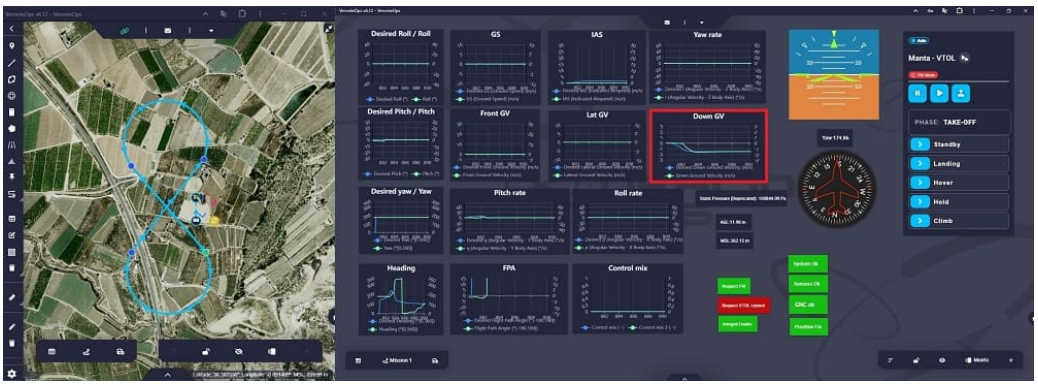



**Operation 1 - Armed flight phase**

## Take-off

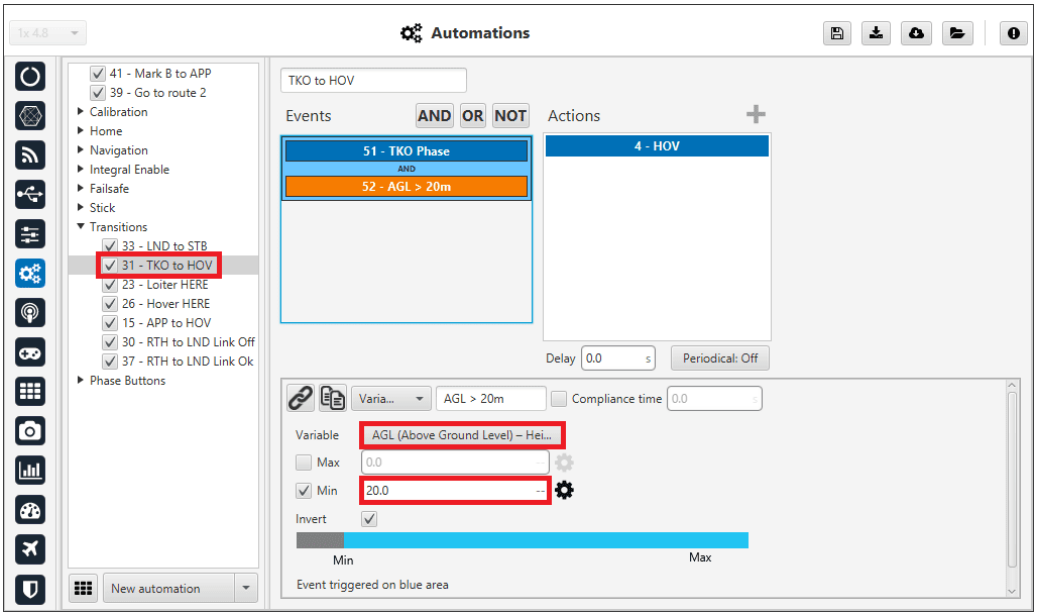
Since Take-off phase is performed with the aircraft in **quadcopter** flight configuration, the control variable is the Down Ground Velocity (DGV).

The aircraft performs a take-off at a DGV of **1.5 m/s**.




**Operation 1 - Take-off flight phase**

Once the aircraft reaches **20 meters** (AGL), Autopilot 1x transitions to the **Hover** flight phase. This automation has been explained in the Automations - 1x PDI Builder configuration section.




**Operation 1 - TKO to HOV automation**

## Hover

In this flight phase, Veronte Autopilot 1x stabilizes the 3D position, so the aircraft attitude control variables (Desired Pitch, Desired Roll, Desired Yaw), Desired Down Ground Speed and Desired IAS variables are **null and constant**.

**Operation 1 - Hover flight phase**

## Cruise

When the operator commands Autopilot 1x to swtich to the **Cruise** flight phase, a path to the mission point defined as **Start Route** is automatically generated.

To switch from Hover to Cruise, the aircraft changes the flight configuration from quadcopter to **FW**. This change can be represented by plotting the Control mix variable on a chart.

In this workspace, two labels have been added to show the behavior of the PID controllers for quadcopter and FW control:

In this flight phase, the control variables for **attitude** are: Desired Pitch, Desired Roll and Desired Yaw. And for **aircraft speed**: Front Ground Velocity and Lateral Ground Speed.

**Operation 1 - Cruise flight phase**

**Veronte Ops** allows users to visualize the aircraft trajectory during the mission. For more information, visit the Platform icon - Veronte Ops configuration of the **Veronte Ops** user manual.

**Operation 1 - Trajectory in cruise flight phase**

> ⓘ **Note**
>
> In this operation, the aircraft will remain in the Cruise phase until a phase change is commanded.

Loiter

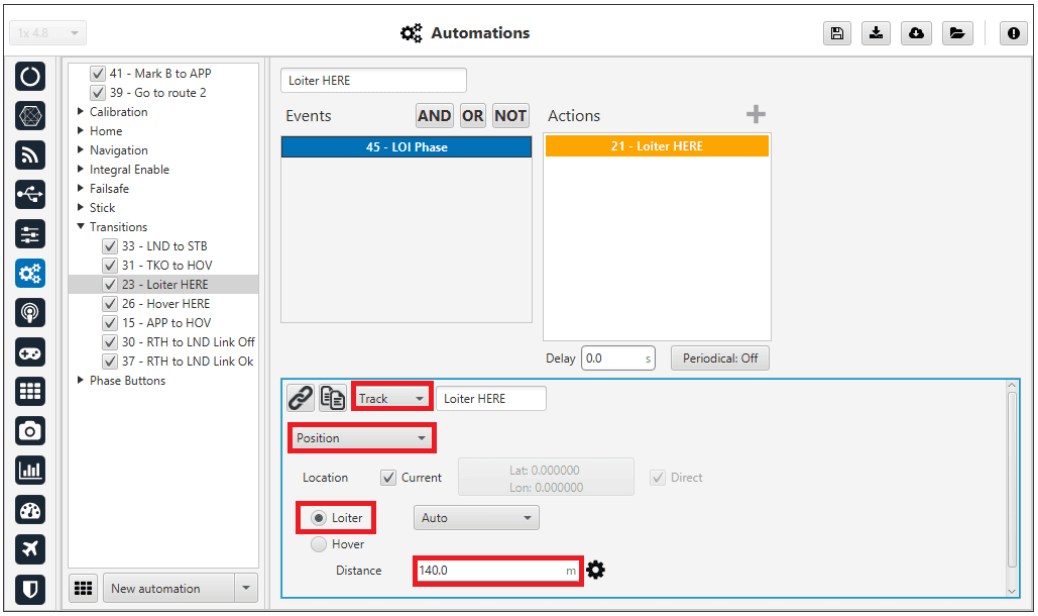
> ⓘ **Note**
>
> Loiter phase can only be entered manually by the user from its phase button on the Veronte Panel.

When the operator commands Autopilot 1x to switch to Loiter flight phase, a circular path is generated around a point, which is the current position where the aircraft is at that moment.

> ⓘ **Note**
>
> The aircraft follows this route until another phase change command is given.

**Operation 1 - Loiter flight phase**

The path configuration for Loiter is defined in the following automation:




**Operation 1 - Loiter here automation**

For more information, visit the Automations - 1x Air configuration section of this manual.

## Hold

In this flight phase, the aircraft attitude and vertical velocity are stabilized. The control variables Desired Pitch, Desired Roll, Desired Yaw and Desired Down Ground Velocity are **null and constant**.

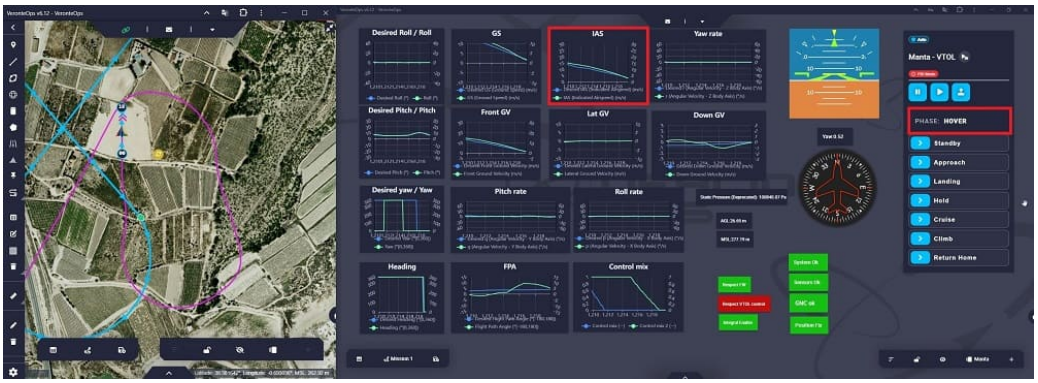
**Operation 1 - Hold flight phase**

## Approach

When the aircraft enters the Approach flight phase, the guidance program automatically generates a path to the runway. This path will have enough distance for the aircraft to lose altitude and perform the approach correctly.

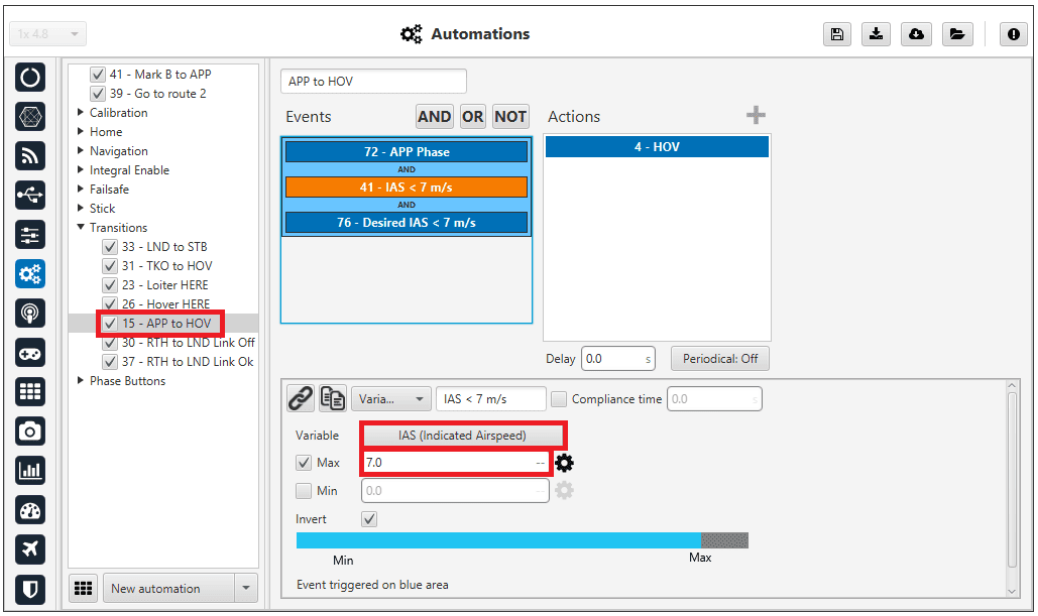
**Operation 1 - Approach flight phase**

Approach phase ends on the runway, where the aircraft transitions to the **Hover** flight phase.




**Operation 1 - Approach to Hover transition**

The automation that enables the transition from the **Approach** phase to the **Hover** phase has been defined in the 1x PDI Builder configuration:

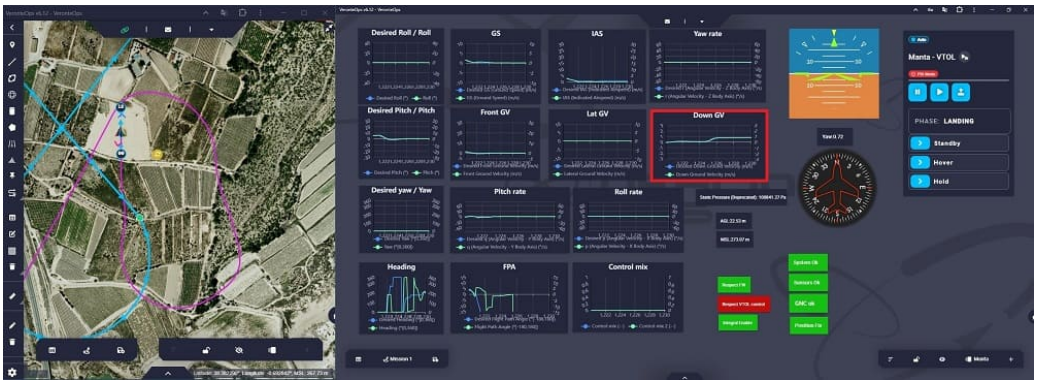



**Operation 1 - APP to HOV automation**

For more information, visit the Automations - 1x Air configuration section of this manual.

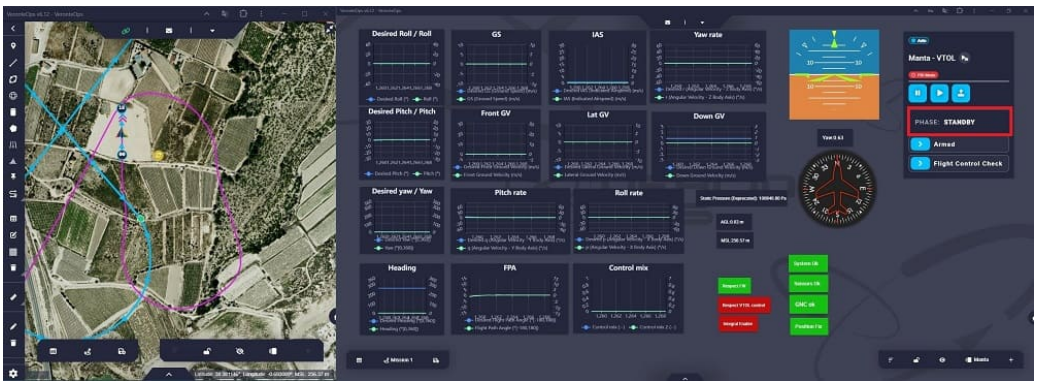
## Landing

The control variable for the Landing phase is the Desired Down Ground Velocity:



**Operation 1 - Landing flight phase**

At the end of the Landing phase, the aircraft enters the **Standby** phase.



### Operation 1 - Landing to Standby transition

The automation that allows the transition from the **Landing** phase to the **Standby** phase has been defined in the 1x PDI Builder configuration:

**Operation 1 - LND to STB automation**

For more information, visit the Automations - 1x Air configuration section of this manual.

## Return to Home

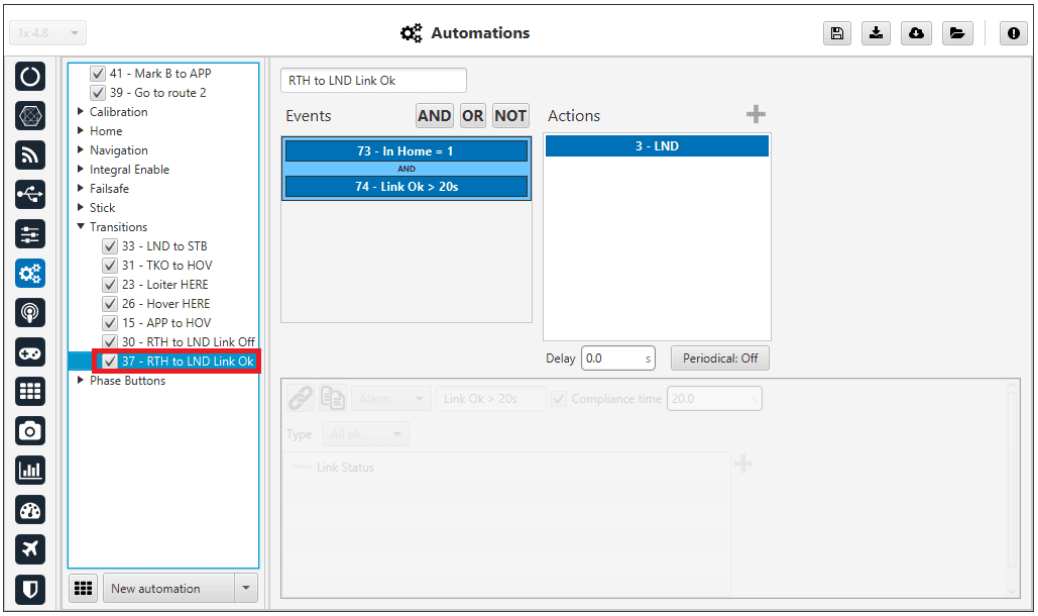
When the aircraft enters the Return to Home phase, the guidance program automatically generates a path to the point defined as **Home**.

**Operation 1 - Return to Home flight phase**

This flight phase ends at the **Home** point, where the aircraft passes to the **Landing** flight phase.

**Operation 1 - Return to Home to Landing transition**

The automation that enables the transition from the **Return to Home** phase to the **Landing** phase has been defined in the 1x PDI Builder configuration:




**Operation 1 - RTH to LND automation**

For more information, visit the Automations - 1x Air configuration section of this manual.

Operation 2

Operation 2 is the same as Operation 1 but with the addition of two areas that trigger an action:

- **Area 1**: Change to Return to Home flight phase.
- **Area 2**: Obstacle.

The actions triggered in this operation will be performed while the aircraft is flying in the Cruise phase. This phase begins at the point defined as **Start**

**Route**, therefore, following the route orientation, the aircraft will pass firstly through area 2 and secondly through area 1.

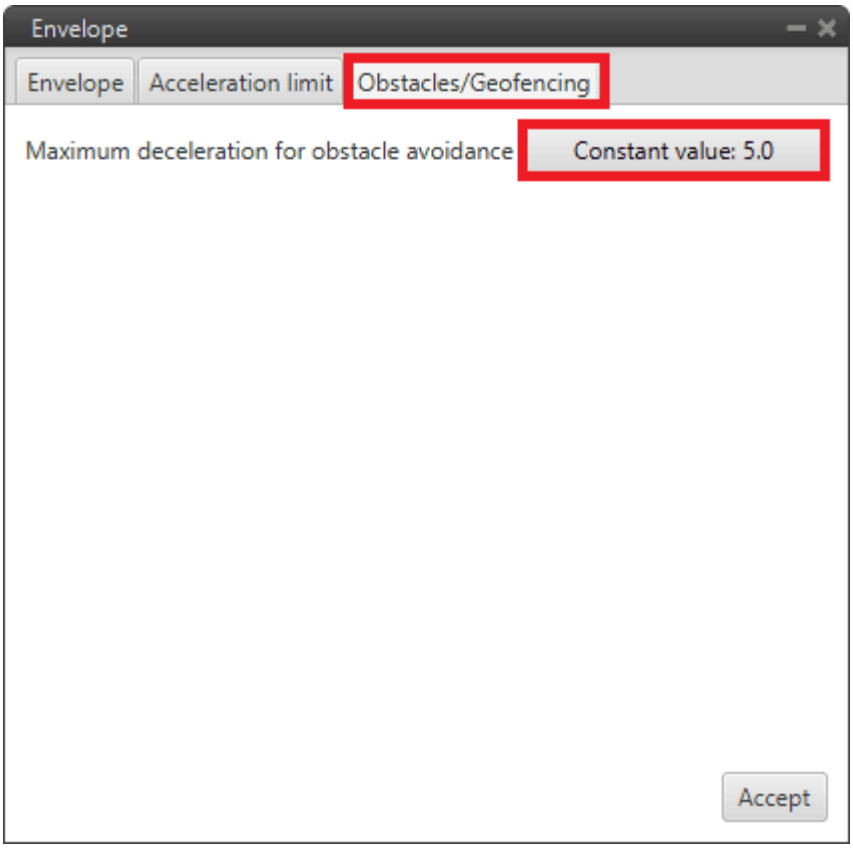
> ⓘ **Note**
>
> The transitions between phases and basic actions to perform the operation have already been explained in Operation 1, so only the parts specific to Operation 2 are detailed in this section.

### Area 2

As area 2 has been defined as an **obstacle**, the guidance program will prevent the aircraft from flying **within** this area by creating an alternative trajectory:

**Operation 2 - Obstacle avoidance trajectory**

The Maximum deceleration for obstacle avoidance is defined in the Envelope block - Guidance program of the 1x PDI Builder configuration.




**Operation 2 - Envelope block**

### Area 1

Area 1 is linked to a phase change automation that is activated when the aircraft flies **inside** it. **Autopilot 1x** will switch to the **Return Home** phase of flight.

**Operation 2 - Area 1 to RTH**

### Operation 3

In this operation, two missions have been defined, each consisting of a closed loop and a mark:

- **Mark A**: Triggers the **Go To** action to Mission 2.

- **Mark B**: Triggers the action to change to the **Approach** flight phase.

> ⓘ **Note**
>
> The transitions between phases and basic actions to perform the operation have already been explained in Operation 1, so only the aspects specific to Operation 3 are detailed in this section.

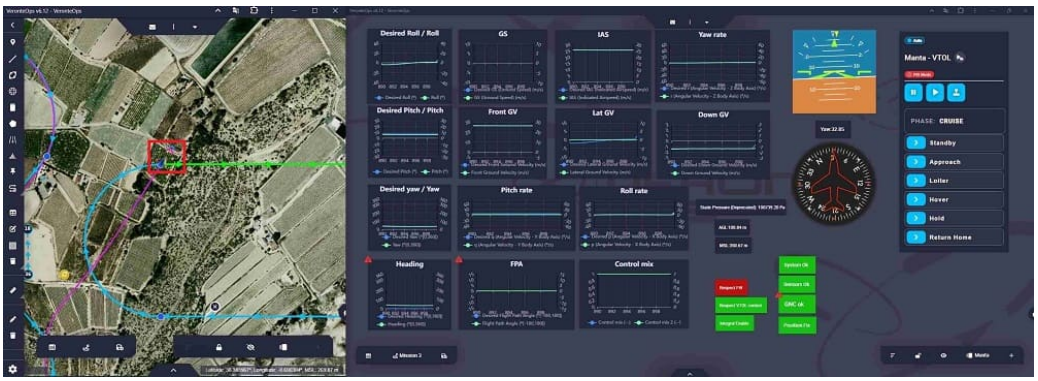
### Mark A to Mission 2

The cruise phase begins at the point defined as the **Start Route**.

When the aircraft flies through the first mission and reaches **Mark A**, the action to **go to** the second mission is triggered. For this purpose, the guidance program generates a path between the current location of Autopilot 1x (i.e. the point where the mark is placed) and the selected **patch**:

**Operation 3 - Mark A to Patch A**

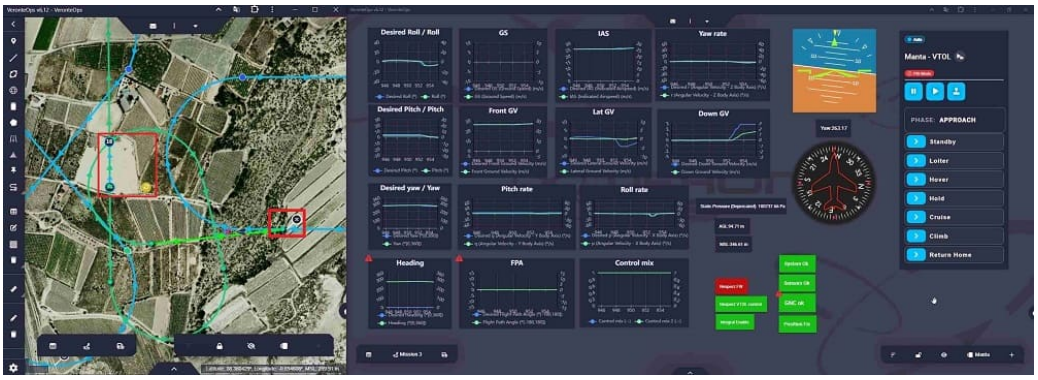
When the aircraft arrives at the selected patch, it will continue flying Mission 2.




**Operation 3 - Mission 2**

## Mark B to Approach phase

When the aircraft reaches the **Mark B** in the second mission, a phase change to the **approach** phase is triggered. Consequently, the guidance program will automatically generate a route to the **runway**.

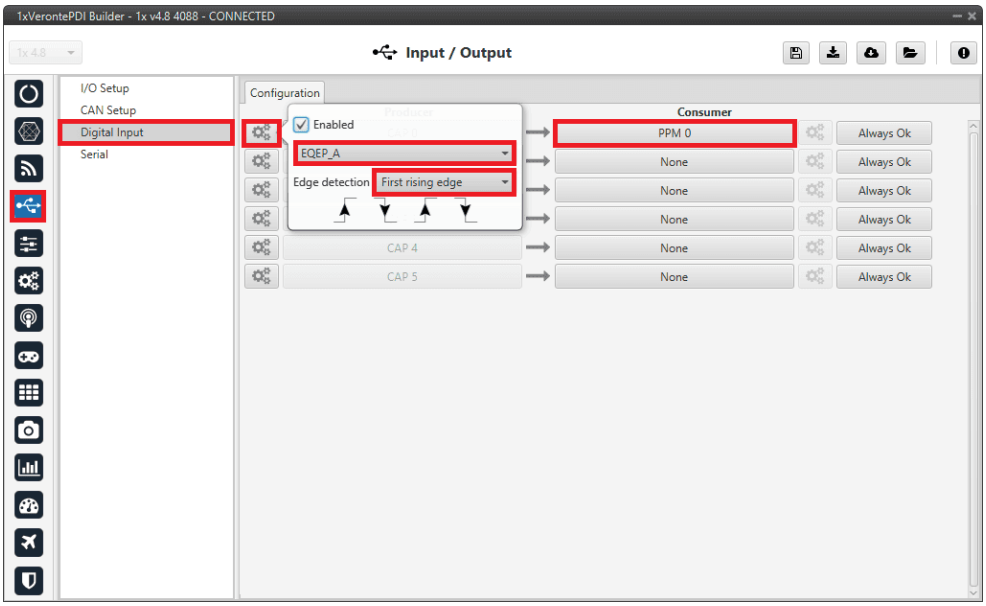


**Operation 3 - Mark B to Approach phase**

### Stick

When performing HIL simulations, the user only uses one Autopilot 1x. Follow the steps below to configure the stick for this case:

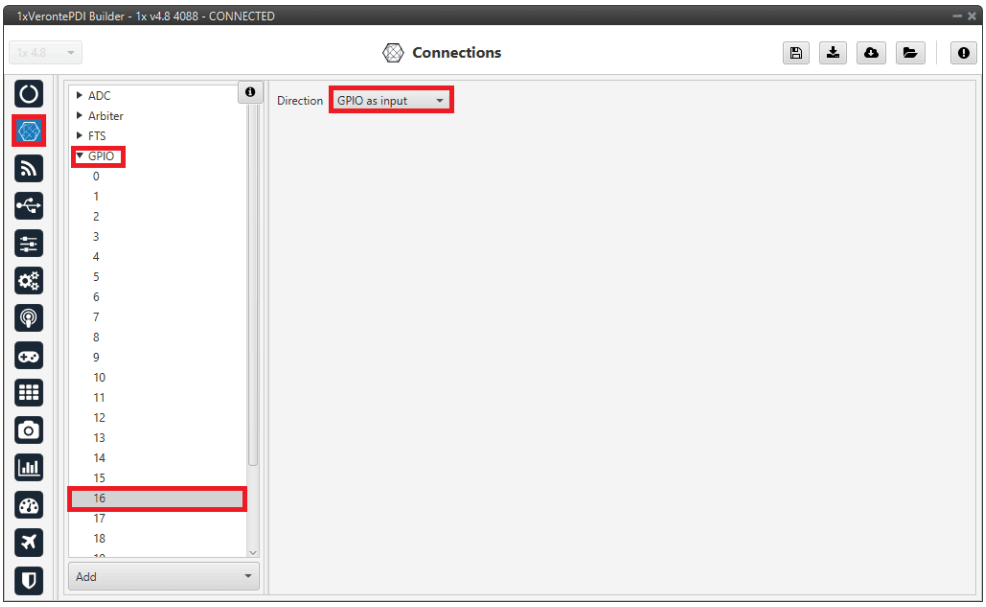
1. Go to Input/Output menu → **Digital Input panel**.
   - Producer: **CAP 0**
     - Enabled
     - Select the pin to which the transmitter is connected, in this case EQEP_A
     - Edge detection: First rising edge
   - Consumer: **PPM 0**



**Simulation - Digital Input configuration**

2. Go to Connections menu → **GPIO panel**.

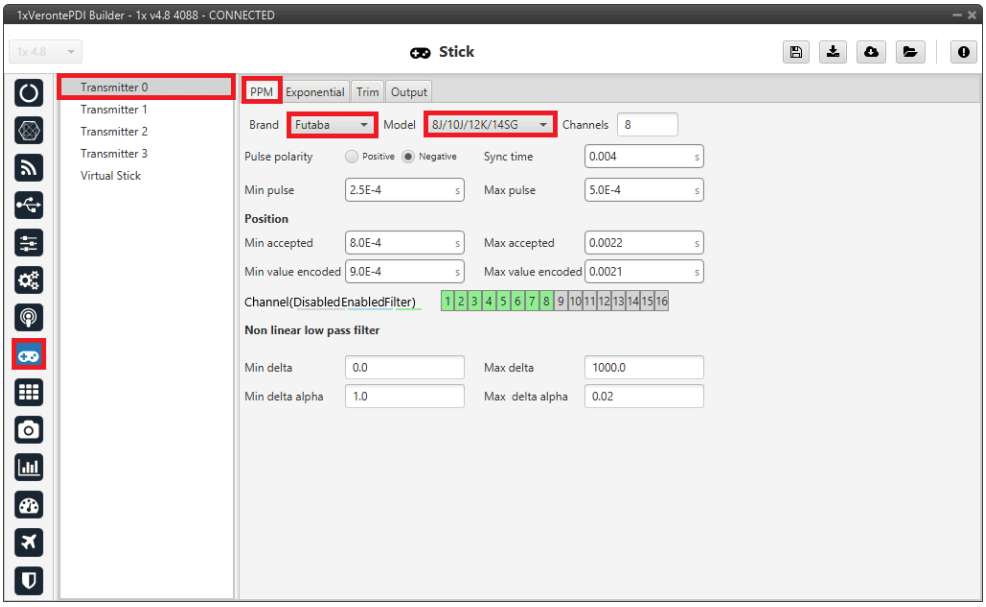
   Verify that the pin to which the transmitter is connected, in this case GPIO 16 (i.e., EQEP A), is set as **input**.




**Simulation - GPIO configuration**

3. Go to Stick menu → Transmitter 0 panel → **PPM tab**.

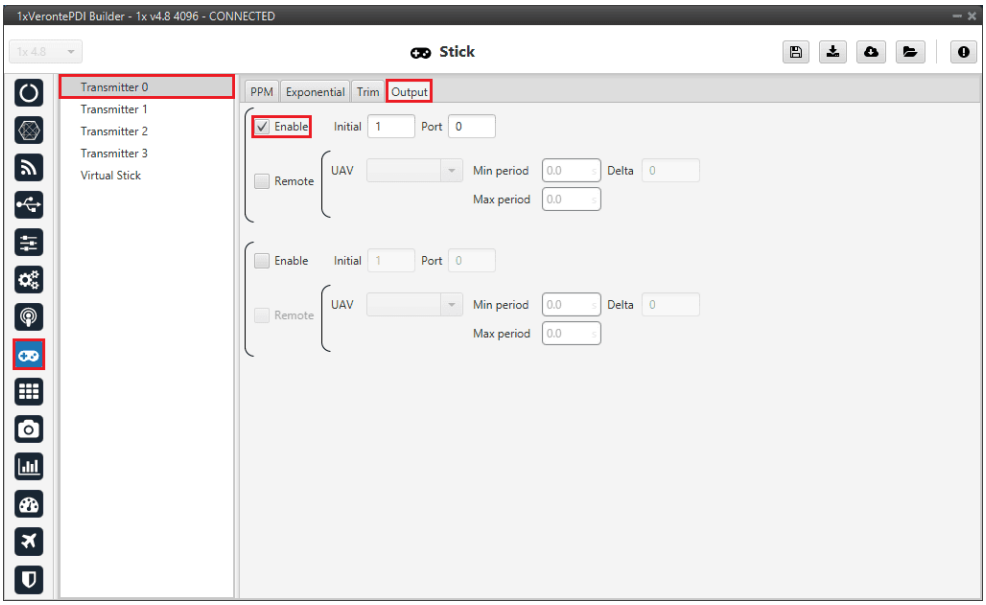
   Select the brand of transmitter that applies.




**Simulation - PPM configuration**

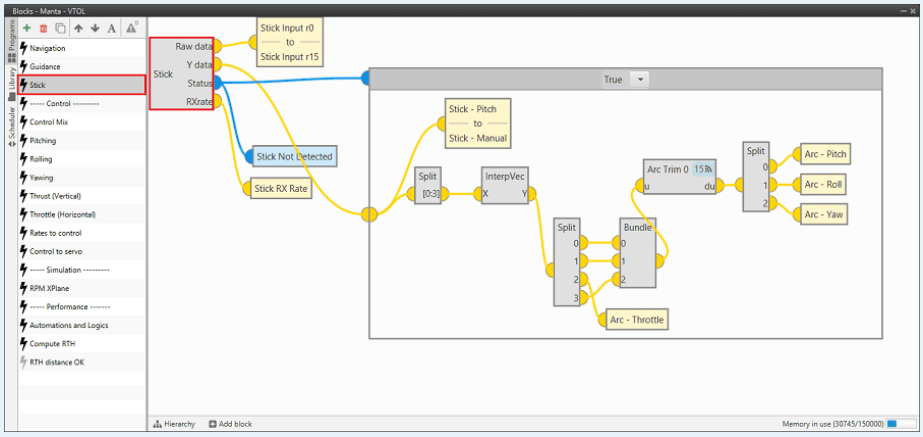
4. Go to Stick menu → Transmitter 0 panel → **Output tab**.

   Just click on **Enable**.

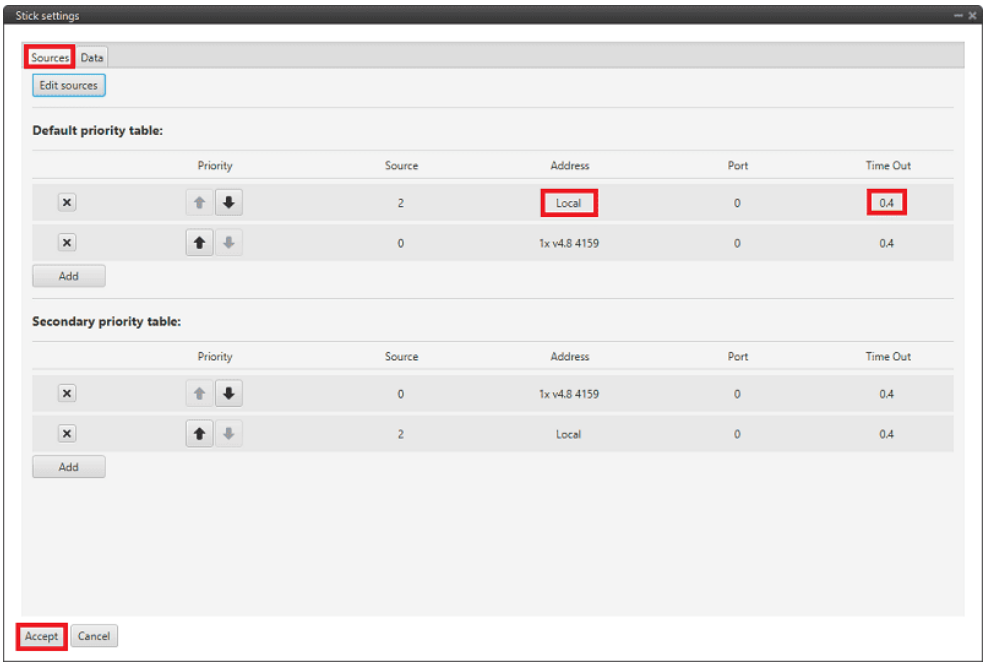



**Simulation - Output configuration**

5. Go to Block Programs menu → **Stick program** → Double click on the **Stick block** → **Edit sources**.

> **ⓘ Note**
>
> This is the Stick progam explained in the **1x Air configuration** section of this manual.
>
> 
> 
>
> **Stick program**

Input the **Local address** to receive the stick information from that source and put it as the **highest priority** in the priority table. We recommend a Time Out of **0.4 s**.

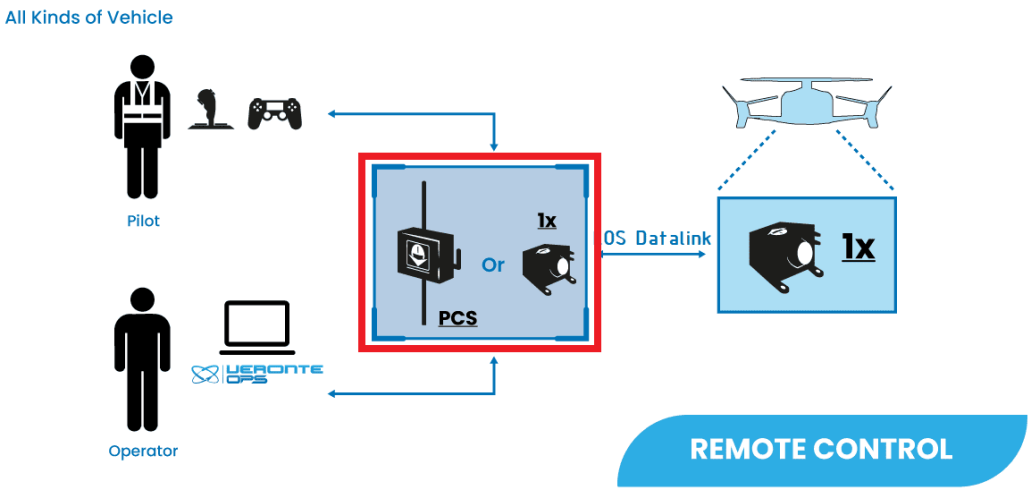



**Simulation - Stick block configuration**

## 1x GCS configuration

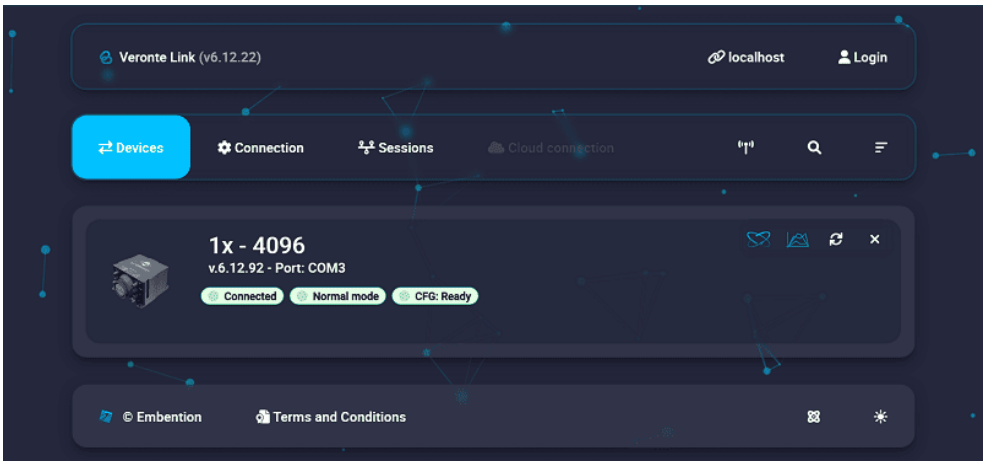
This section defines the GCS (Ground Control Station) configuration to be used to communicate with the Veronte Autopilot 1x on board the aircraft.

> ⓘ **Note**
>
> In this case, an Autopilot 1x is used as GCS.



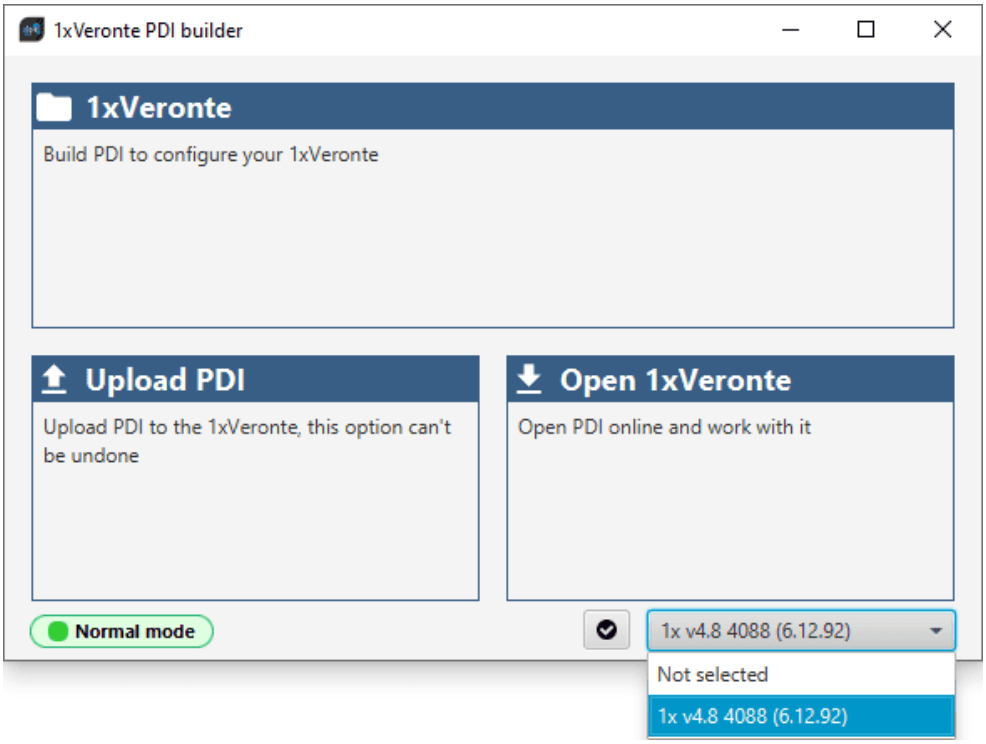
To configure the 1x Ground unit, the user must download the **Veronte Ground** template available in the **1x PDI Builder** software. To do so, follow the steps below:

1. Connect the Autopilot 1x to be configured as Veronte Ground to the PC, in this case, the connection is made via **USB**. Users can refer to the Basic connection for operation - Quick Start section of the **1x Hardware Manual** for instructions on how to make this connection.
2. Check in **Veronte Link** that the Autopilot 1x is in **ready** status.

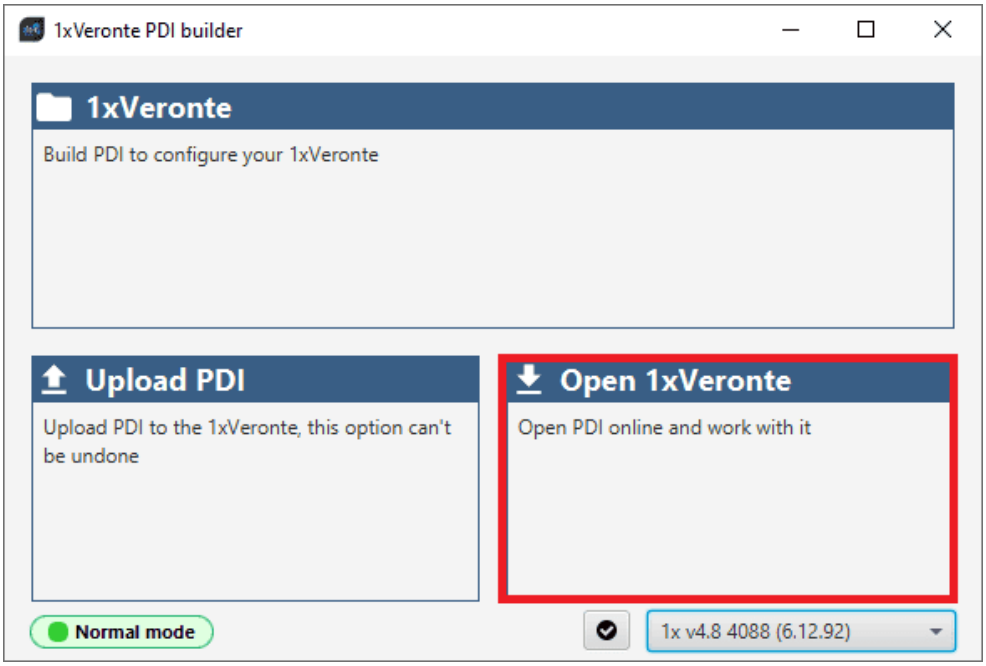



   For a detailed explanation of how to set up this connection, please refer to the Serial connection - Integration examples section of the **Veronte Link** user manual.

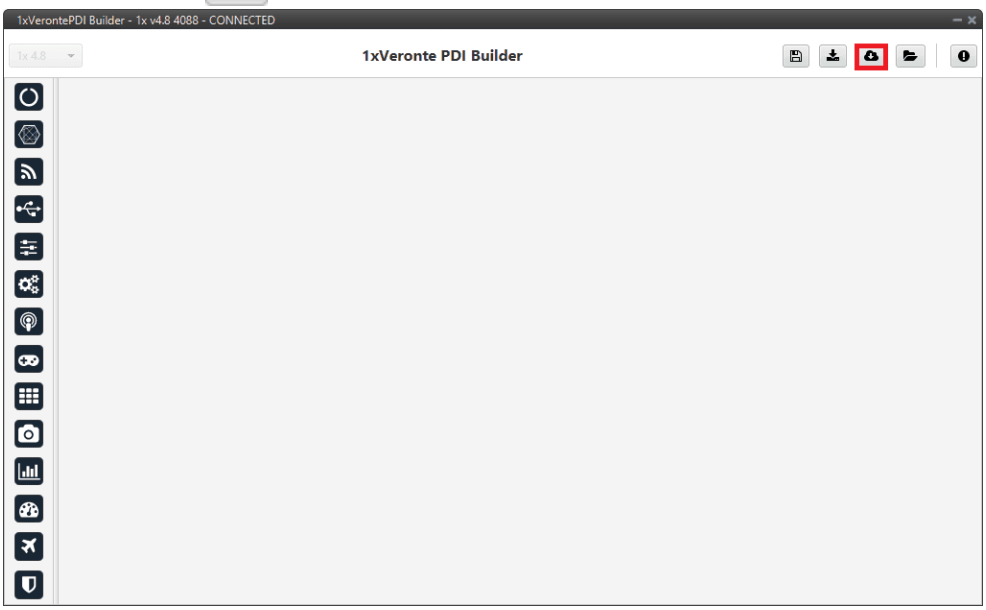
3. Open the **1x PDI Builder** app and select the Autopilot 1x:



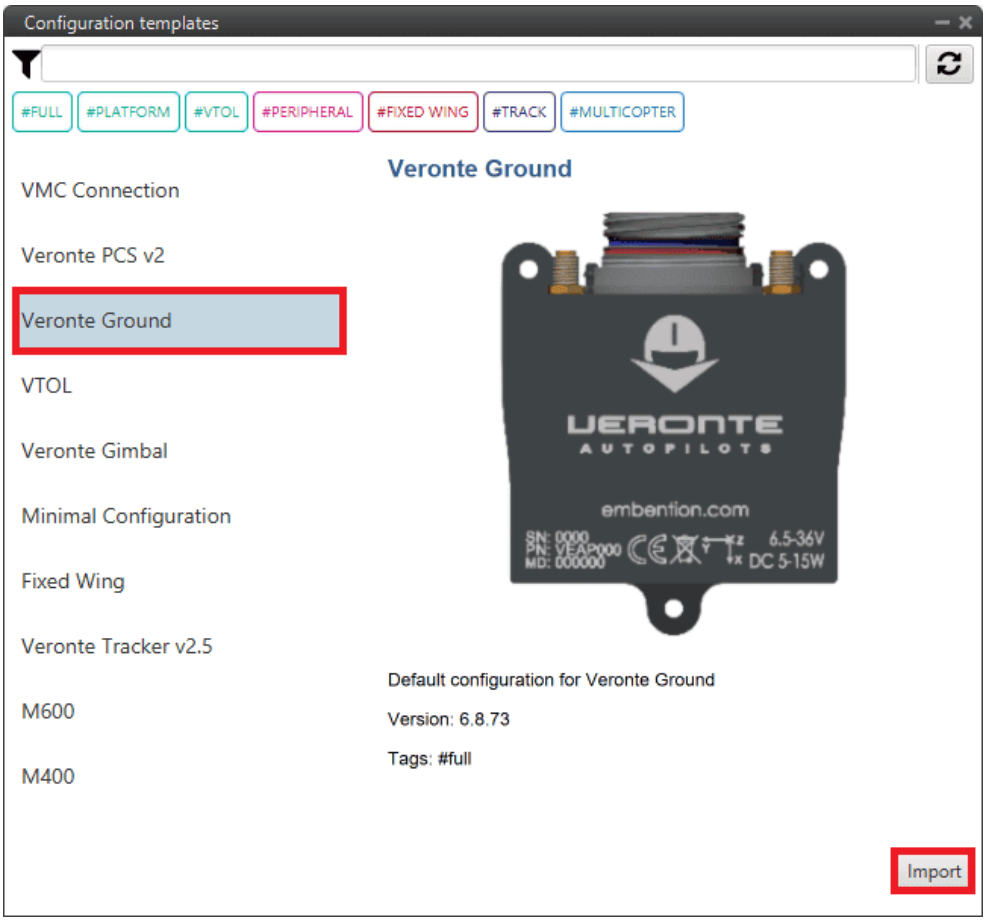
4. Select the **Open 1xVeronte** option to access the configuration of the connected unit:



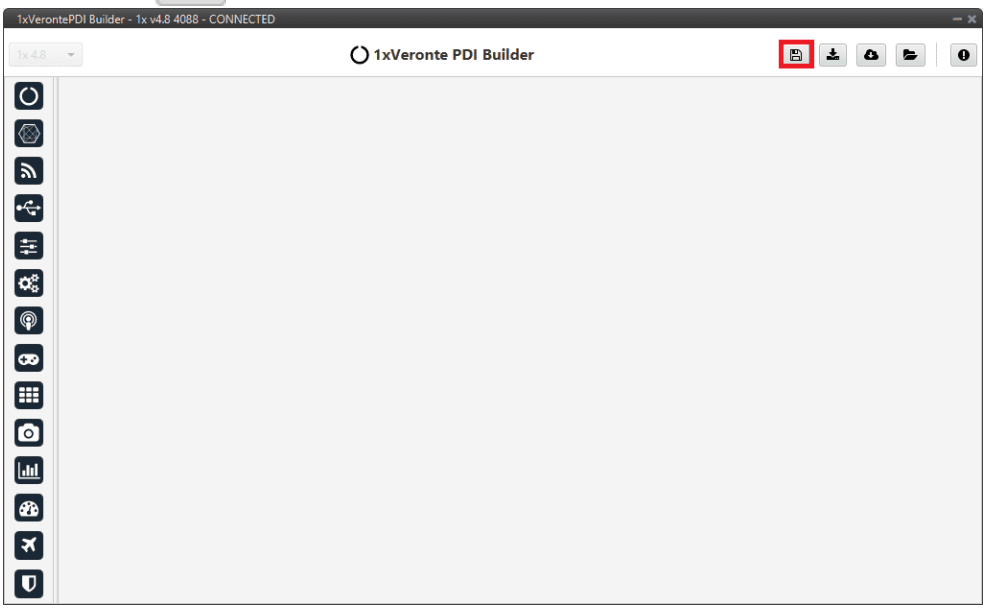

5. Click on the button in the menu bar to access the Veronte **templates**:




6. Select the **Veronte Ground** template and click **import**:




7. Click the button to load the downloaded template into Autopilot 1x.

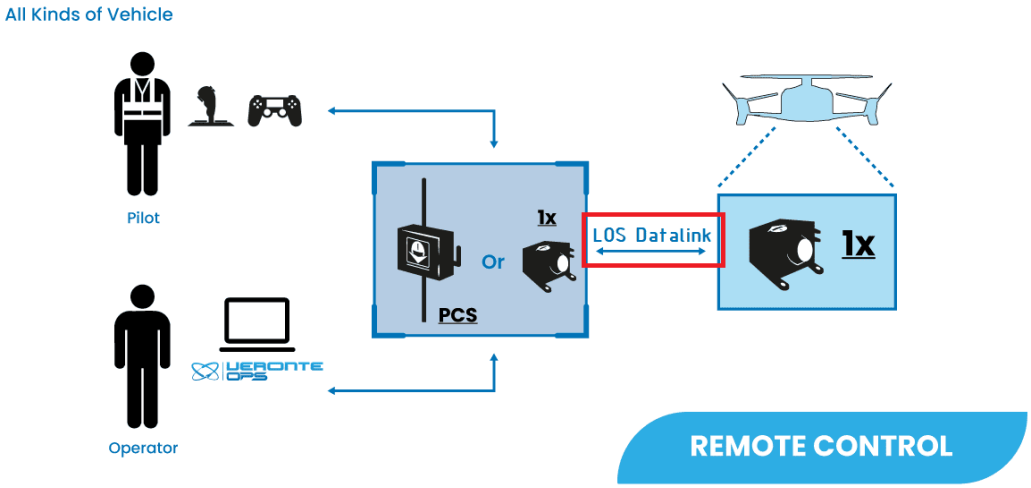


> **Important**
> In order to save the configuration, Veronte Autopilot 1x must enter in maintenance mode. Then, after saving any changes, Autopilot 1x will RESET and 1x PDI Builder software will consequently close. For more information on this, visit the Configuration section of the **1x PDI Builder** user manual.

Once the **Veronte Ground** template has been downloaded and uploaded into the Autopilot 1x, follow the steps in the next section, Air-GCS connection, to set up the connection between GCS and the on-board Autopilot 1x.

## Air-GCS connection

This section defines how to configure the air-ground Autopilots 1x to communicate with each other and send the desired information.




- Communication. This section defines the different configurations that must be performed to ensure communication between Autopilots 1x.
- Stick. This section defines the configuration of the Autopilots 1x to receive the commands sent by the Stick.

### Communication

This section explains the steps to follow to set up **radio communication** between Autopilots 1x. Follow the steps below:

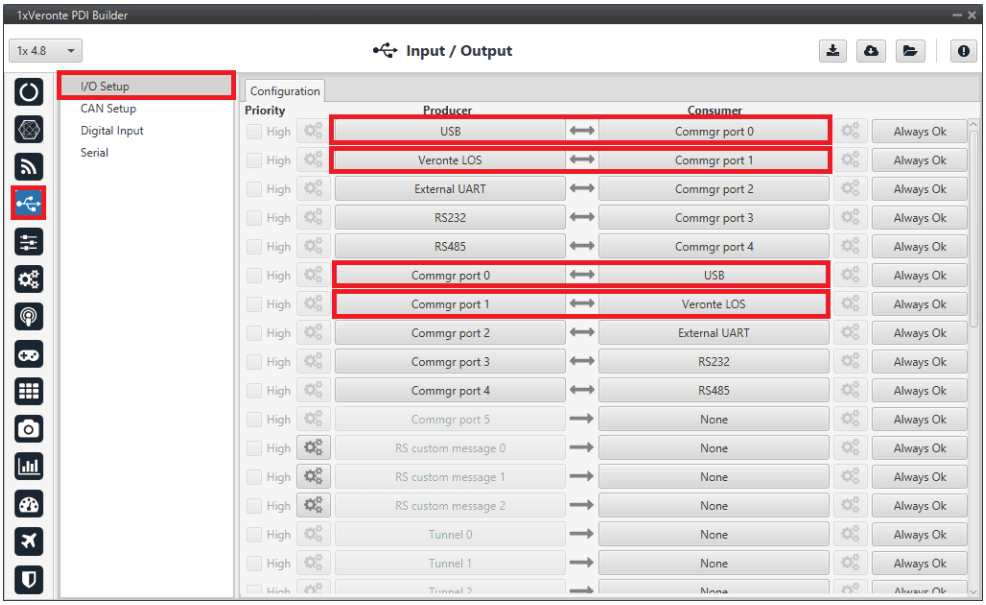
## GND unit configuration

1. Go to Input/Output menu → **I/O Setup panel**.

   The following connections betweens **producers** and **consumers** must be configured here:

   - **USB** ↔ **Commgr port 0**.
   - **Veronte LOS** ↔ **Commgr port 1**.
   - **Commgr port 0** ↔ **USB**.
   - **Commgr port 1** ↔ **Veronte LOS**.

   > **Note**
   >
   > It is not necessary to use these particular COM Manager ports, however, the connections between these producer and consumer must be **bidirectional**.




**Ground unit - Input/Output configuration**

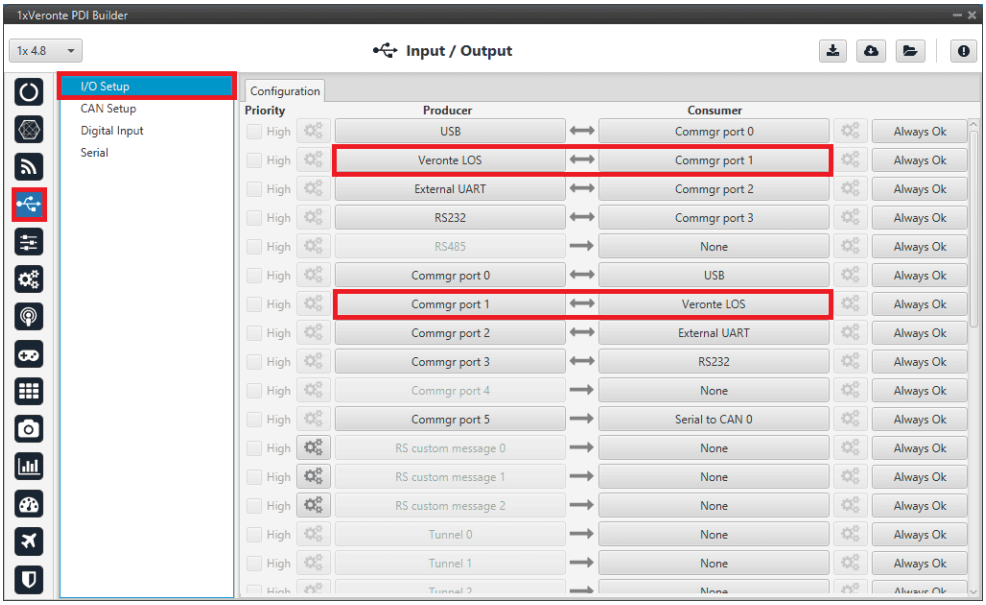
## Air unit configuration

1. Go to Input/Output menu → **I/O Setup panel**.

   Configure the following connections between **producers** and **consumers**:

   - **Veronte LOS** ↔ **Commgr port 1**.
   - **Commgr port 1** ↔ **Veronte LOS**.

> **Note**
>
> It is not necessary to use these particular COM Manager ports, however, the connections between these producer and consumer must be **bidirectional**.




**Air unit - Input/Output configuration**

For more information, visit the Input/Output - Configuration section of the **1x PDI Builder** user manual.

## GND-Air communication configuration

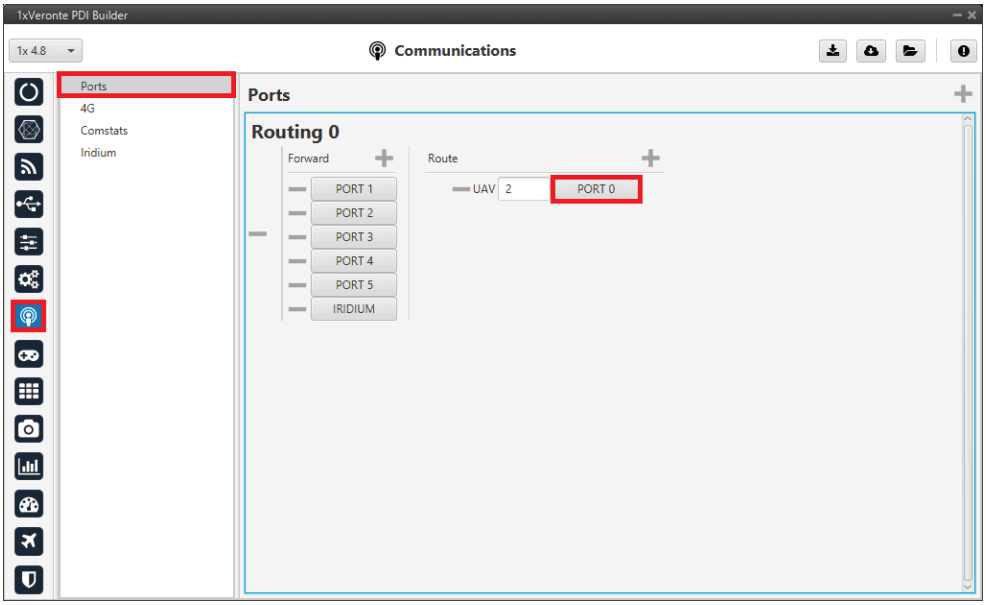
To establish a proper communication between the ground and air units, it is necessary to configure the **ports** menu of the **ground unit**.

1. Go to Communications menu → **Ports panel**.
2. In order to establish the connection between the PC and the 1x air unit thanks to the radio connection with the 1x ground, a routing of the communications must be made to the port to which the USB consumer is connected.

   Therefore, set a routing of **Address 2** (address by which Autopilots 1x ground and air units **communicates with all Veronte applications**) through **Commgr port 0**.

   This way, any messages that are received through a Commgr Port (i.e. through Veronte LOS) with **address 2**, will be re-routed through **Port 0**

(USB) and received by **Veronte Ops** software, including any messages generated by 1x ground unit itself.




**Ground unit - Routing configuration**

For more information, please refer to the Communication between Veronte Autopilots 1x - Integration examples section of the **1x PDI Builder** user manual.

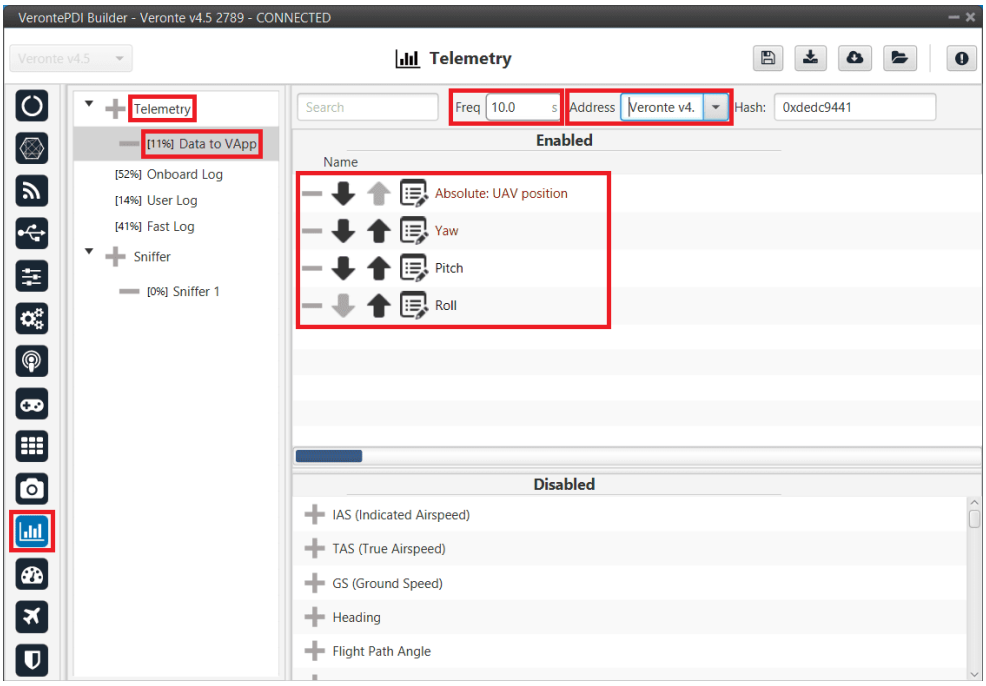
## GND-Air data transmission configuration

To enable the data transmission between the ground and air units, the **telemetry** and **sniffer** menus must be configured, respectively.

A simple example of use between a ground unit and an air unit is shown below:

In the **1x ground unit**:

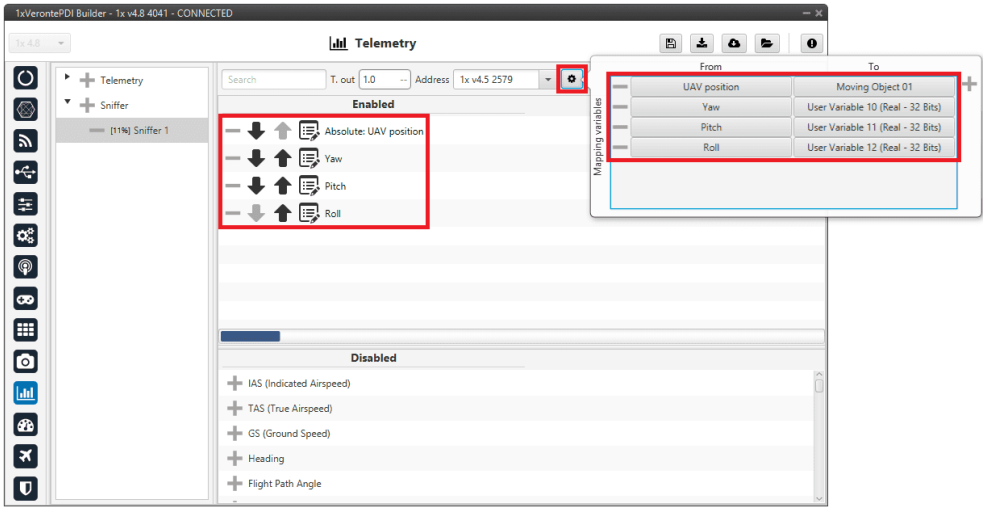
1. Go to Telemetry menu → Telemetry panel → **Data link to VApp tab** (for more information about this, see Data vectors - Telemetry section of **1x PDI Builder** user manual).
2. Add the variables: Absolute: UAV position, Yaw, Pitch and Roll.
3. Set a **Frequency**, it is recommended to set it to **10 Hz**.
4. On **Address**, point to the **1x air unit** (it is needed to have both units connected through the radio in order to be able to see them on the menu).




**1x ground unit - Telemetry**

For the **1x air** unit:

1. Go to Telemetry menu → **Sniffer panel** (for more information about this, see Sniffer - Telemetry section of **1x PDI Builder** user manual).
2. Add a new Sniffer.
3. Configure the same **variables** (keeping the **same order**) than in the ground unit.
4. On **Address**, point to the **1x ground unit**.
5. 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).




**1x air unit - Sniffer**

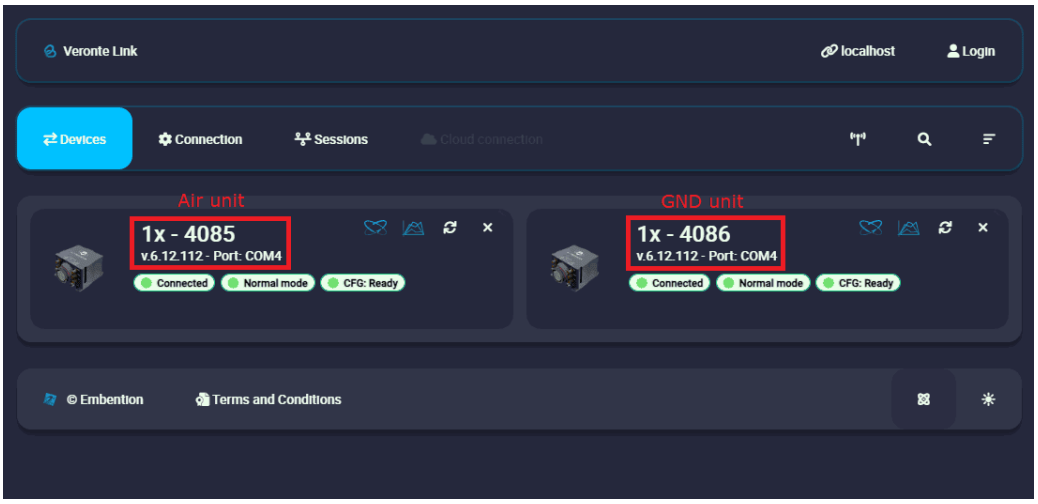
## Radio pairing

In order to establish communication between the two units, it is necessary that the **radios** of both devices are **paired**. For this operation, there is an internal **Digi** radio in both Veronte Autopilots 1x.

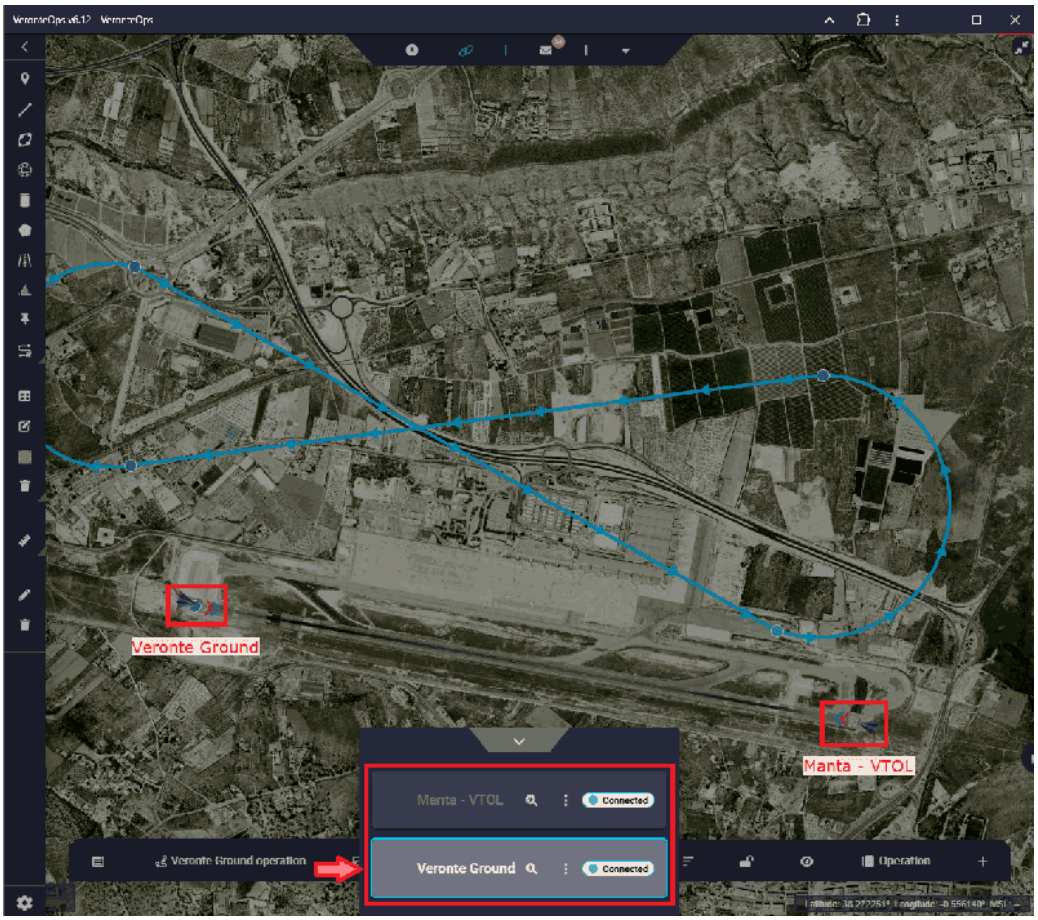
Follow the steps described in the Digi internal radio - Integration examples section of the **1x PDI Builder** user manual, for both units, GND and air.

If everything went well, the user will be able to see the air unit in both **Veronte Link** and **Veronte Ops** by **physically connecting only the ground unit**.

> ⓘ **Note**
>
> Autopilot 1x configured as the GND unit has address **4086** and the air unit has address **4085**.



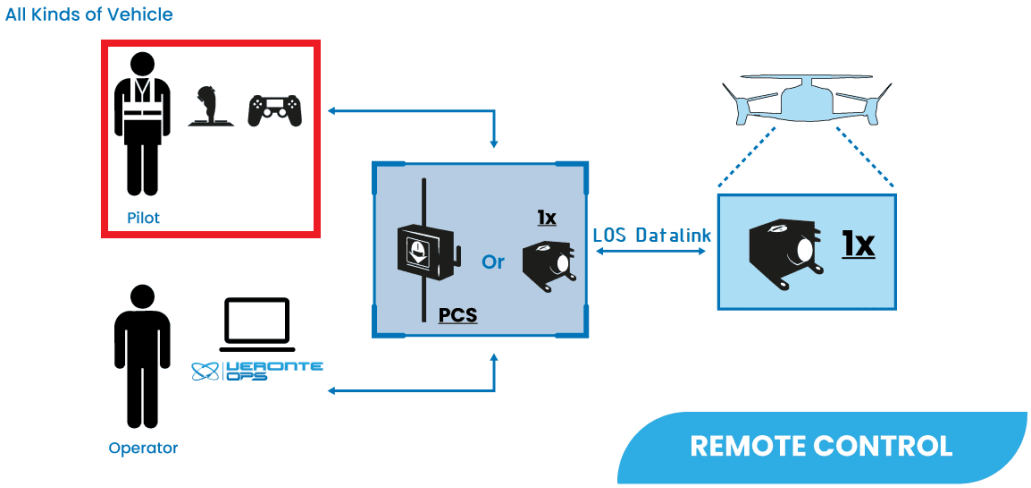

**Radio pairing - Veronte Link**




**Radio pairing - Veronte Ops**

Stick

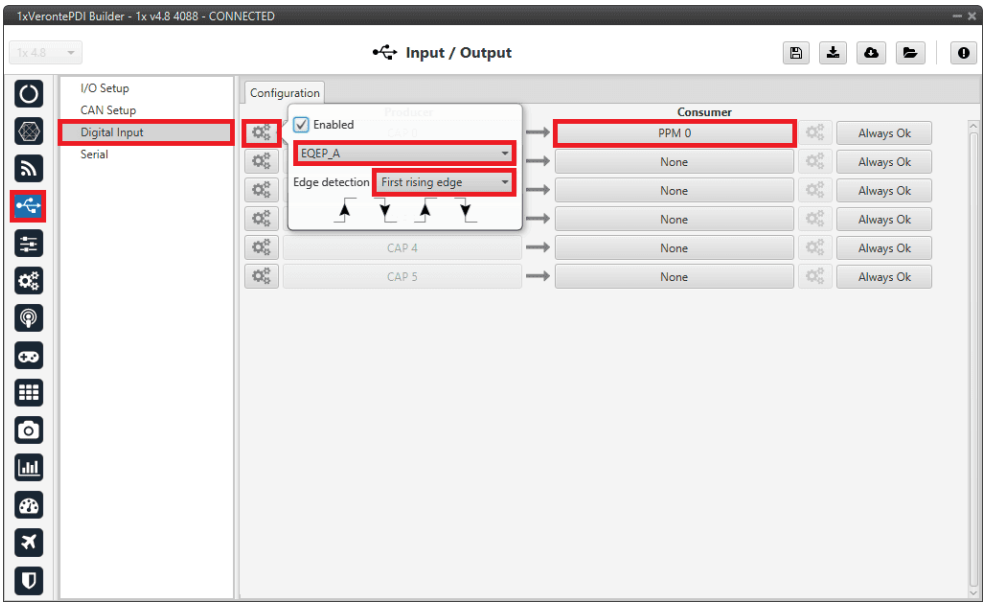
**Veronte Standard Layout** is the case where the 1x ground unit (or BCS/PCS) sends commands directly to the 1x air unit. So the stick is connected directly to the **1x ground unit**.




Follow the steps below to perform a correct stick configuration on both units.

## Ground unit

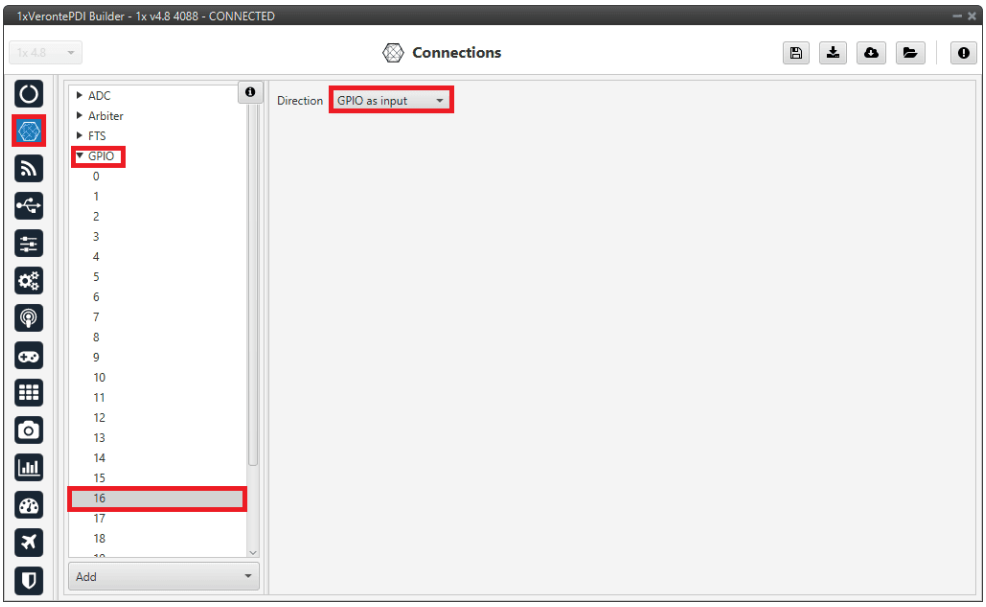
1. Go to Input/Output menu → **Digital Input panel**.
   - Producer: **CAP 0**
     - Enabled
     - Select the pin to which the transmitter is connected, in this case EQEP_A
     - Edge detection: First rising edge
   - Consumer: **PPM 0**




**Ground unit - Digital Input configuration**

2. Go to Connections menu → **GPIO panel**.

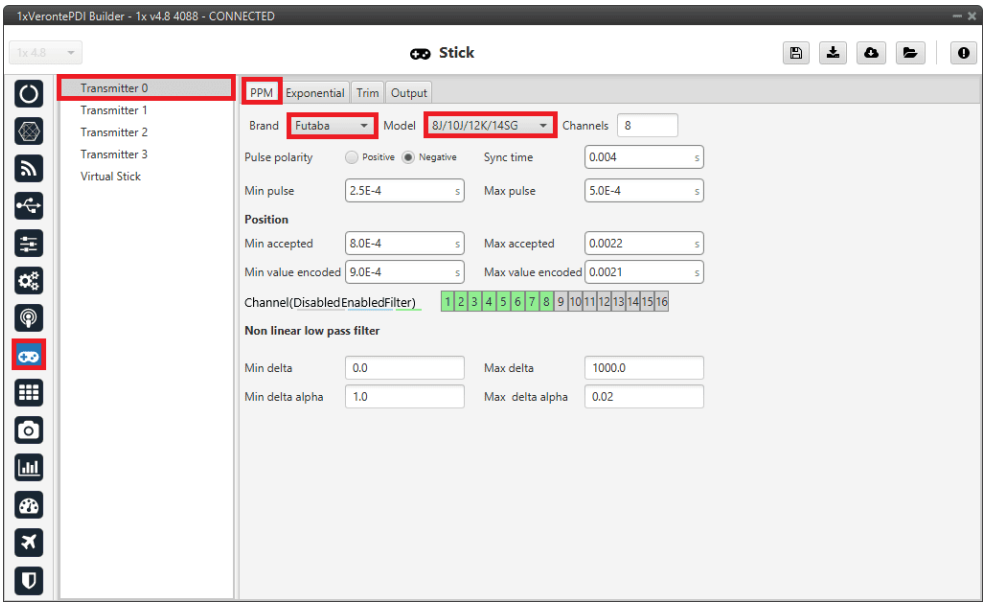
   Verify that the pin to which the transmitter is connected, in this case GPIO 16 (i.e., EQEP A), is set as **input**.



**Ground unit - GPIO configuration**

3. Go to Stick menu → Transmitter 0 panel → **PPM tab**.

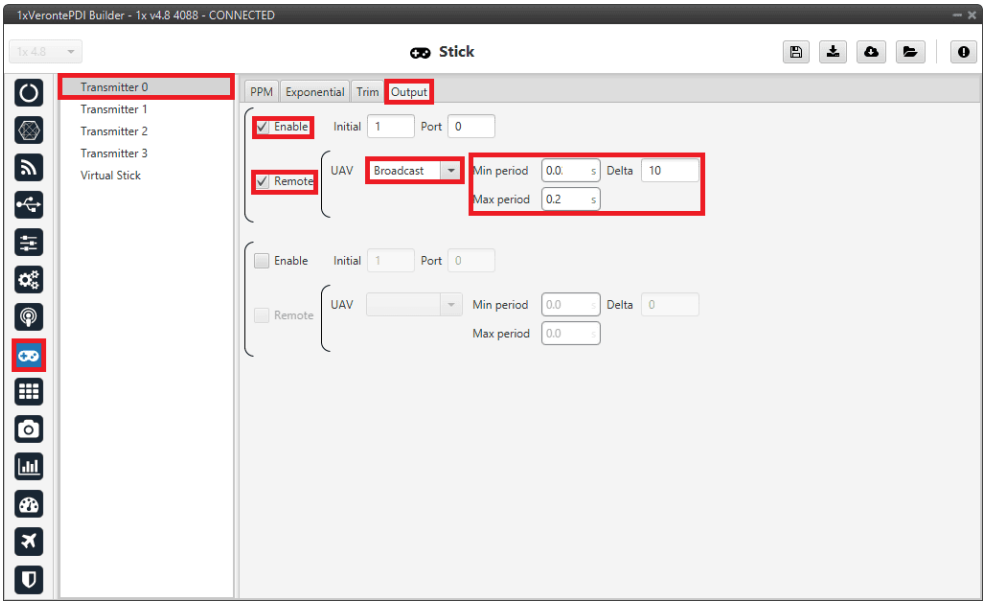
   Select the brand of transmitter that applies.




**Ground unit - PPM configuration**

4. Go to Stick menu → Transmitter 0 panel → **Output tab**.

   Click on **Enable** and on **Remote** to send the stick information to the air unit. The recommended values for the configurable parameters have been entered here, as detailed in the Output - Stick section of the **1x PDI Builder** user manual.




**Ground unit - Output configuration**

If all these settings are correct, users can check that **'Stick PPM 0 not detected'** variable of the **GND unit** is true.

**Stick PPM 0 not detected variable - True**

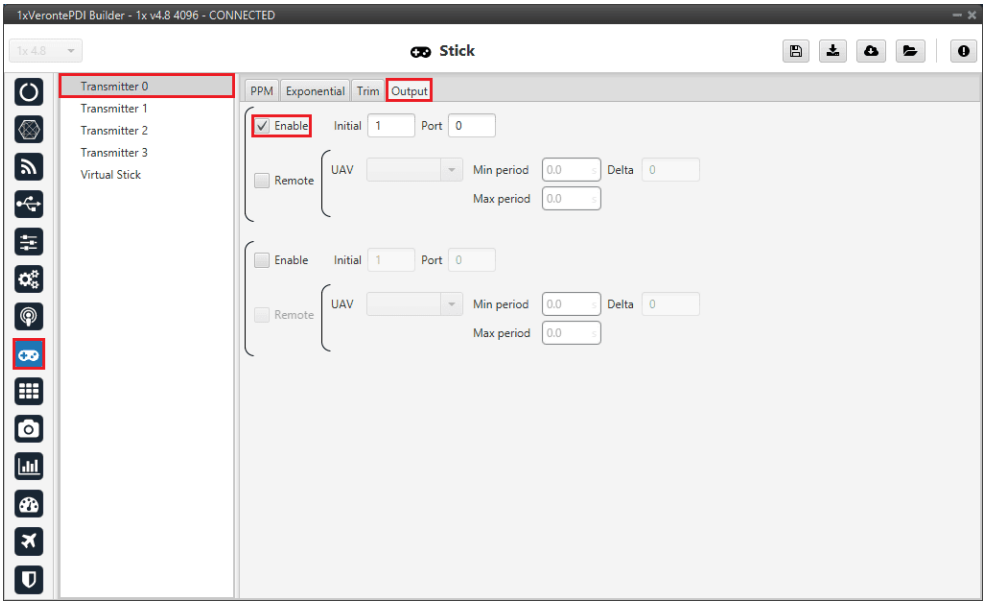
## Air unit

1. Go to Stick menu → Transmitter 0 panel → **PPM tab**.

   Select the brand of transmitter that applies (make the same configuration as for the ground unit).

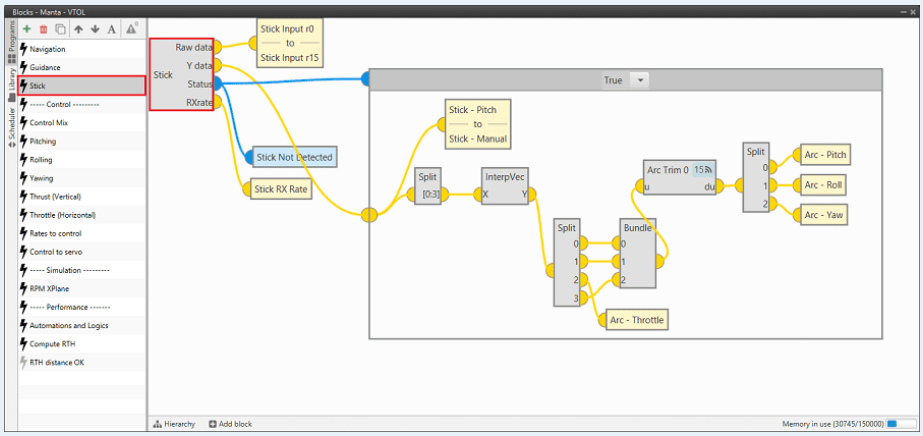
2. Go to Stick menu → Transmitter 0 panel → **Output tab**.

   Just click on **Enable**.

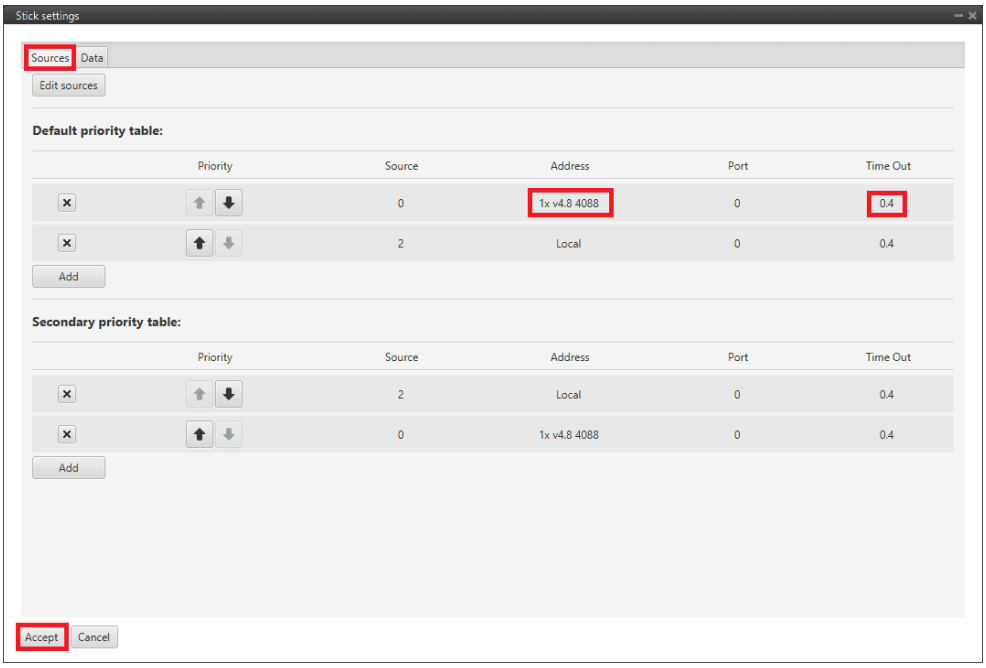



**Air unit - Output configuration**

3. Go to Block Programs menu → **Stick program** → Double click on the **Stick block** → **Edit sources**.

> **ⓘ Note**
>
> This is the Stick progam explained in the **1x Air configuration** section of this manual.
>
> 
> 
>
> **Stick program**

Input the **ground unit address** to receive the stick information from that source and put it as the **highest priority** in the priority table. We recommend a Time Out of **0.4 s**.




**Air unit - Stick block configuration**

Then, if all is correct, users can check that **'Stick not detected'** variable of the **AIR unit** is true.

### Stick not detected variable - True

And that means that the communication between the GND and the AIR unit is correctly configured.