MC110 Software Manual

Release 6.12/1.0

Embention Sistemas Inteligentes, S.A.

2025-02-24

Contents

Software applications

To properly connect and configure a **MC110 unit**, **Veronte Link** and **MC110 PDI Builder** are required.

Veronte Link

Veronte Link establishes communication between a computer and any Veronte product by creating a VCP bridge. It allows to use 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.

MC110 PDI Builder

This tool is used to set all the configurable parameters. Here the user can set, tune and define the motor, control systems and sensors that are going to be used alongside the ESC.

For more information, visit the MC110 PDI Builder user manual.

List of Variables

This section shows the variables employed exclusively by **Veronte MC110**. The rest of variables can be read in the Lists of Variables - Lists of interest section of **1x Software Manual**.

BIT Variables

ID	Name	Description
400	C1 Low Frequency	Low priority task frequency - Dependent on CIO 2057) • 0 for error → CIO Running Freq • 1 for OK → CIO Running Frequ
402	Acquisition Step Missed	• 0 for Acquisition step missed → High priority higher than permitted • 1 for Acquisition Task OK → High priority ta under set limits (1%
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 sta
485	MC Current sensing error	0 for error, 1 for OK
486		ADC phase U not calibrated - 0 for not calibra

ID	Name	Description
	MC Phase U Current Calibration Error	
487	MC Phase V Current Calibration Error	ADC phase V not calibrated - 0 for not calibra
488	MC Phase W Current Calibration Error	ADC phase W not calibrated - 0 for not calibra
489	MC Maximum Temperature Error	Maximum power module temperature exceeded for OK
490	MC Phase Error	Power module driver phase error reported
491	MC General Driver Error	Power module driver error reported - 0
492	MC Over-current AC	Current AC side limit exceeded - 0 for erro
493	MC Over-voltage advertisement	Over-voltage DC side limit advertisement exceed 1 for OK
494	MC Over-voltage caution	Over-voltage DC side limit caution exceeded - 0 OK
495	MC Under-votage latching	Critical under-voltage DC side limit violatio
496	MC Under-votage non latching	Non critical under-voltage DC side limit violat
497	MC RMS imbalance	Current AC side imbalance - 0 for
498	MC Open DC fault	Open-circuite DC side fault - 0 for

ID	Name	Description
499	MC Over-current DC	Current DC side limit exceeded - 0 for erro

Real Variables (RVar) - 32 bits

ID	Name	Units/ Values	De
2057	CIO Running Frequency	Hz	Low priority ta
2058	CIO Min Running Frequency	Hz	Minimum assured fr
2330	Control Loop Period	s	MC con
2331	Control Loop Maximum Period	s	MC maximur
2332	Control Loop Duration	s	MC control loop
2333	MC Control Loop Maximum Duration	s	MC control loop r
2335	MC Control Loop Maximum CPU Usage Ratio	%	MC maximu
2336-2338	MC U-V-W Phase Current	A	MC U-V-
2339	MC Electrical Angle	rad	MC el
2340	MC Mechanical Angle	rad	MC me
2341	MC Mechanical Angular Speed	rad/s	MC mechar

ID	Name	Units/ Values	De
2342	MC Desired Mechanical Angle	rad	MC desired
2343	MC Position Controller Output	rad/s	MC pos
2344	MC Desired Mechanical Angular Speed	rad/s	MC desired me
2345	MC Desired Mechanical Angular Speed After Speed Limiter	rad/s	MC desired mecha sp
2346	MC Speed Controller Output	A	MC sp
2347-2348	MC Clarke Alpha-Beta Current	A	MC alpha and b tra
2349-2350	MC Actual Direct-Quadrature Current	A	MC actual dire
2351-2352	MC Desired Direct-Quadrature Current	A	MC desired dire
2353-2354	MC Direct-Quadrature Voltage From Controller Output	V	MC curr
2355-2356	MC Alpha-Beta Voltage From Current Controller Output	V	MC Clarke
2362-2364	MC U-V-W Phase PWM Duty Cycle	%	MC

ID	Name	Units/ Values	De
2367	MC Mechanical Angle Error	rad	MC mech
2368-2370	MC U-V-W Phase BEMF	V	MC U-V-W phase
2371	MC Input Current DC side	A	DC
2372	MC Input Normalized Command Speed	customType	Speed input rate f
2373-2374	MC ADC in 0-1	V	System r
2375	MC Logic Board Temperature	K	Boarc
2376	MC Power Module Temperature	K	IGBT filte
2377	MC Motor Temperature	K	Moto
2378	MC Input Voltage DC side	V	DC
2379-2380	MC U-V Phase Hall current senso	customType	System r
2381	MC Virtual and estimator angle difference	rad	Angle offset va commanded an

ID	Name	Units/ Values	De
2382	MC Low speed estimator angle	rad	Low speed obs
2383	MC High speed estimator angle	rad	High speed ob
2384	MC Low speed estimator speed	rad/s	Low speed observ
2385	MC High speed estimator speed	rad/s	High speed obser

Integer Variables (UVar) - 16 bits

ID	Name	Description
801	Control Mode	Index of the control input mode: • 1: PPM • 2: CAN • 3: Both modes active (CAN priority)
802	Actual State	State of motor controller: • 1: Motor stop and driver disabled • 2: Calibration of ADC reading • 3: Initial alignment procedure

ID	Name	Description
		• 4: Open loop procedure • 5: Speed mode

CAN Bus protocol

CAN Commands to MC110

MC110 can receive commands from any CAN device. All CAN messages for **MC110** follow the same structure, a string of bits divided in two groups:

Group	Name	Size	Description
1	CAN Id	11-bits: standard 29-bits: extended	If the **CAN Id** matches with the Id of a **MC110** in consumer), the message will be correctly read by th Otherwise, it will be ignore
2	Payload	4 bytes	**Speed** must be represented with a **compressed 3** little endian format. The values of this variable should be in the range [**0** or [**-maximum RPM to maximum RPM**], dependir allow negative commands **Note** Negative values command the opposite move Therefore, the maximum value corresponds with th maximum negative value corresponds with the m

The parameter that is configured in the **MC110** to receive these CAN commads is the **CAN Id** of **Input filter** producer, which has to be linked to the **CAN Cmd** consumer. To know more, read the CAN I/O - Input/Output section of the **MC110 PDI Builder** user manual.

An example for sending commands from **Veronte Autopilot 1x** to a **MC110** unit is explained in the MC110/MC24 - Integration examples section of the **1x PDI Builder** user manual.

Telemetry messages from MC110

Telemetry messages can be transmitted from the **MC110** unit to provide information of interest to the user, such as the board temperature or the input command values.

CAN messages sent by **MC110** have also the structure:

1. **CAN Id**: It can be in standard frame format (11-bits) or in extended frame format (29-bits). The CAN Id frame format will depend on the CAN protocol of the receiving device.
2. **Variable**: Users can send as telemetry the variable they want to know information about. All the variables available to be sent from the **MC110** unit can be consulted in the lists of variables section of this manual.
   The format in which these variables must be sent will depend on the CAN protocol of the device that will read the message.

Detailed information on **how to build CAN messages** can be consulted in the Custom Messages types - Input/Output section of the **1x PDI Builder** user manual.