
MC110 Software Manual

Release 6.12

Embention

2024-06-14

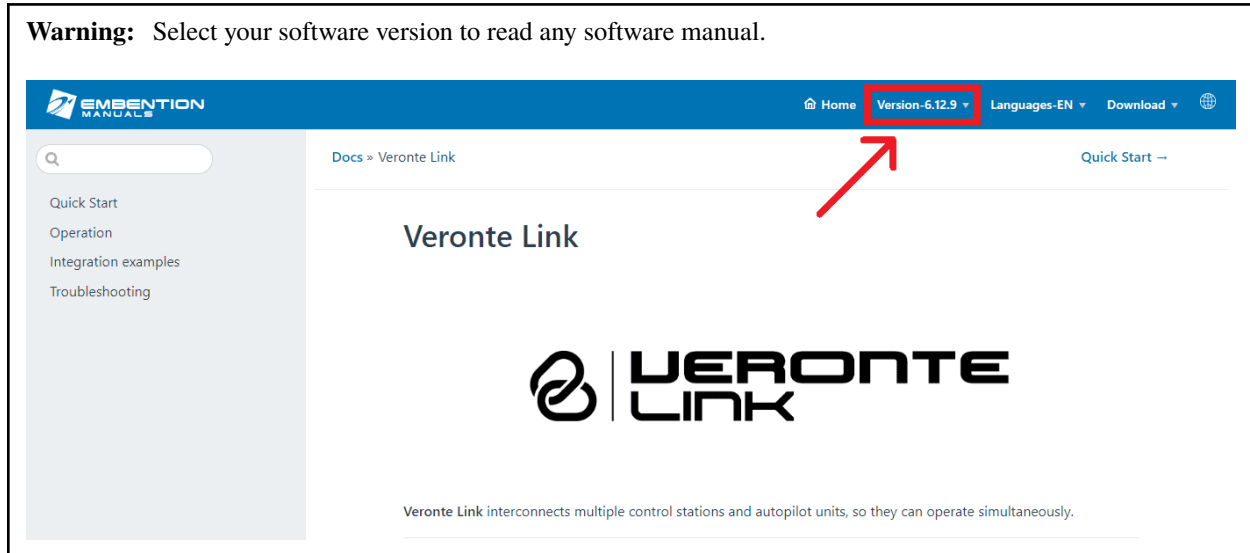
CONTENTS

1	Software applications	3
1.1	Veronte Link	3
1.2	MC110 PDI Builder	3
2	Lists of Variables	5
2.1	BIT Variables	7
2.2	Real Variables (RVar) - 32 Bits	8
2.3	Integer Variables (UVar) - 16 Bits	9
3	CAN Bus protocol	11
3.1	CAN Commands to MC110	11
3.2	Telemetry messages from MC110	11

In this manual the user can consult a brief description of all the applications created and designed to work together with the **Veronte MC110**.

In addition, links are available to access the manuals for each application.

Warning: Select your software version to read any software manual.



SOFTWARE APPLICATIONS

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

1.1 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](#).

1.2 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](#).

LISTS 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**.

2.1 BIT Variables

ID	Name	Description
400	C1 Low Frequency	Low priority task frequency - Dependent on <i>CIO Running Frequency (RVar 2057)</i> <ul style="list-style-type: none"> • 0 for error → <i>CIO Running Frequency</i> < 10 Hz • 1 for OK → <i>CIO Running Frequency</i> > 10 Hz
402	Acquisition Step Missed	<ul style="list-style-type: none"> • 0 for Acquisition step missed → High priority task frequency fluctuation is higher than permitted (1%). • 1 for Acquisition Task OK → High priority task frequency fluctuation is 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 status OK
485	MC Current sensing error	0 for error, 1 for OK
486	MC Phase U Current Calibration Error	ADC phase U not calibrated - 0 for not calibrated, 1 for calibration OK
487	MC Phase V Current Calibration Error	ADC phase V not calibrated - 0 for not calibrated, 1 for calibration OK
488	MC Phase W Current Calibration Error	ADC phase W not calibrated - 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 Error	Power module driver phase error reported - 0 for error, 1 for OK
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 non latching	Non critical under-voltage DC side limit violation - 0 for error, 1 for OK
497	MC RMS imbalance	Current AC side imbalance - 0 for error, 1 for OK
498	MC Open DC fault	Open-circuite 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

2.2 Real Variables (RVar) - 32 Bits

ID	Name	Units/Values	Description
2057	CIO Running Frequency	Hz	Low priority task running frequency
2058	CIO Min Running Frequency	Hz	Minimum assured frequency of low pri
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 tim
2333	MC Control Loop Maximum Duration	s	MC control loop maximum execution t
2335	MC Control Loop Maximum CPU Usage Ratio	%	MC 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	MC Mechanical Angle	rad	MC mechanical angle
2341	MC Mechanical Angular Speed	rad/s	MC mechanical angular speed
2342	MC Desired Mechanical Angle	rad	MC desired mechanical angle
2343	MC Position Controller Output	rad/s	MC 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
2346	MC Speed Controller Output	A	MC speed PDI output
2347-2348	MC Clarke Alpha-Beta Current	A	MC alpha and beta current after Clarke
2349-2350	MC Actual Direct-Quadrature Current	A	MC actual direct-quadrature currents
2351-2352	MC Desired Direct-Quadrature Current	A	MC desired direct-quadrature currents
2353-2354	MC Direct-Quadrature Voltage From Controller Output	V	MC current PIDs outputs
2355-2356	MC Alpha-Beta Voltage From Current Controller Output	V	MC Clarke alpha-beta currents
2362-2364	MC U-V-W Phase PWM Duty Cycle	%	MC PWM outputs
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 f
2371	MC Input Current DC side	A	DC bus current
2372	MC Input Normalized Command Speed	customType	Speed input rate from source (CAN or
2373-2374	MC ADC in 0-1	V	_____
2375	MC Logic Board Temperature	K	Board temperature
2376	MC Power Module Temperature	K	IGBT filtered temperature
2377	MC Motor Temperature	K	Motor temperature
2378	MC Input Voltage DC side	V	DC bus voltage
2379-2380	MC U-V Phase Hall current sensor	customType	_____
2381	MC Virtual and estimator angle difference	rad	Angle offset value from estimated and
2382	MC Low speed estimator angle	rad	Low speed observer estimated angle
2383	MC High speed estimator angle	rad	High speed observer estimated angle
2384	MC Low speed estimator speed	rad/s	Low speed observer estimated mechan
2385	MC High speed estimator speed	rad/s	High speed observer estimated mechan

2.3 Integer Variables (UVar) - 16 Bits

ID	Name	Description
801	Control Mode	Index of the control input mode: <ul style="list-style-type: none">• 1: PPM• 2: CAN• 3: Both modes active (CAN priority)
802	Actual State	State of motor controller: <ul style="list-style-type: none">• 0: Motor stop and driver disabled• 1: Calibration of ADC reading• 2: Initial alignment procedure• 3: Open loop procedure• 4: Speed mode

CAN BUS PROTOCOL

3.1 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	4 bytes	CAN Id could be in standard (11-bits) or extended frame format (29-bits). If the CAN Id matches with the Id of a MC110 input filter (a type of CAN consumer), the message will be correctly read by the MC speed filter consumer. Otherwise it will be ignored.
2	Speed	4 bytes	Speed must be represented with a compressed 32-bit signed variable with little endian format. The values of this variable should be in the range [0 to maximum RPM (speed)] or [-maximum RPM to maximum RPM], this depends if users want to allow negative commands. Note: Negative values command the opposite movement to positive ones. Therefore, the maximum value corresponds with the maximum speed and the maximum negative value corresponds with the maximum reversed speed.

The parameter that is configured in the **MC110** to receive these CAN commands 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.

3.2 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.