MC01 Software Manual

Release 6.12

Embention

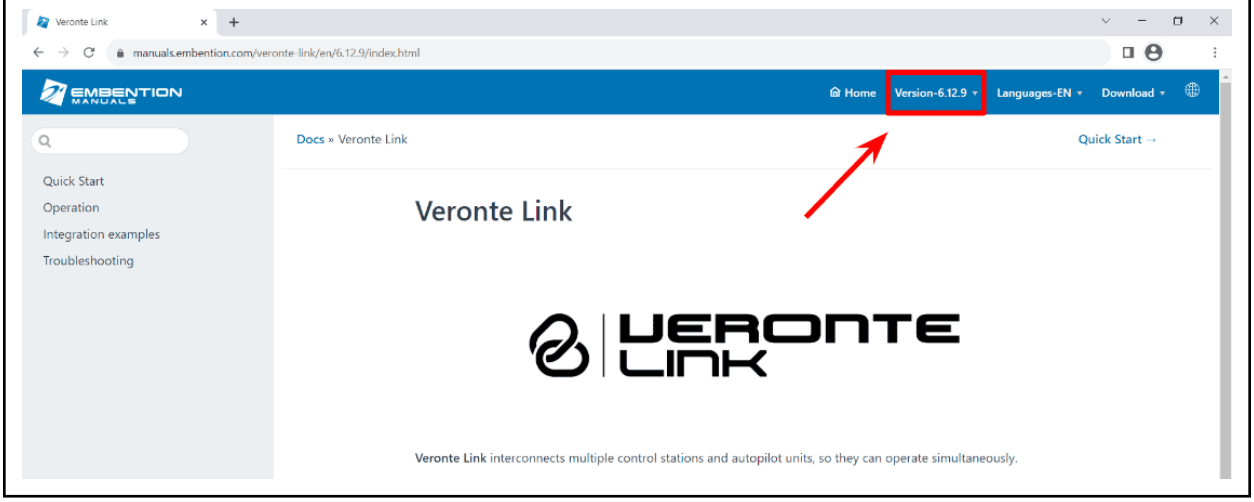
2024-02-12

CONTENTS

In this manual the user can consult a brief description of all the applications created and designed to work together with the **Veronte MC01**.

In addition, links are available to access the manuals for each of these applications.

Warning: Select your version before reading any user manual for software. The following image shows where to select a version from any Embention user manual.



CHAPTER

ONE

SOFTWARE APPLICATIONS

For configuring **MC01**, Veronte Link is required to connect an **Autopilot 1x** to the computer. This autopilot must be configured to operate as a CAN-USB (or CAN-RS) converter in 1x PDI Builder for then, configuring **MC01** through its corresponding **PDI Builder**.

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 1x PDI Builder

1x PDI Builder is the main configuration tool to adapt a **Veronte Autopilot 1x** to a specific application, including user-defined commnication 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 are accomplished.
- Block Programs: **Veronte Autopilot 1x** can be programmed with a friendly-user programming language.

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

1.3 MC01S PDI Builder

MC01S PDI Builder configures the stepper variant of **MC01** (MC01S). This application allows to adapt control, communications and telemetry to each motor implementation.

To know more, read the MC01S PDI Builder user manual.

1.4 MC01B PDI Builder

MC01B PDI Builder configures the brushless variant of **MC01** (MC01B). This application allows to adapt control, communications and telemetry to each motor implementation.

To use it, consult the MC01B PDI Builder user manual.

CHAPTER

TWO

LISTS OF VARIABLES

This section shows the variables employed exclusively by **Veronte MC01**. The rest of variables can be read in the Lists of variables section of **1x Software Manual**.

2.1 BIT Variables

ID	Name	Description
400	C1 Low Frequency	Low priority task frequency - Dependent on *CIO Running Frequency (RVar 2057)* • 0 for error → *CIO Running Frequency* < 10 Hz • 1 for OK → *CIO Running Frequency* > 10 Hz
402	Acquisition Step Missed	• 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%).
480	MC Stepper direction output	0 for error, 1 for OK
481	MC Brushless driver fault	0 for error, 1 for OK
484	MC General health error	0 for health error, 1 for status 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 priority task
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 execution time
2334	Control Loop CPU Usage Ratio	%	MC CPU usage ratio
2335	MC Control Loop Maximum CPU Usage Ratio	%	MC maximum CPU usage ratio
2336-2338	MC U-V-W Phase Current	-	MC U-V-W phase current **Note:** These are normalized dimensionless variables
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 after speed limiter
2346	MC Speed Controller	-	MC speed PDI output

CHAPTER

THREE

CAN BUS PROTOCOL

All CAN messages for **MC01** follow the same structure: a chain of bits divided in three groups:

<table>
<tr><th>Position</th><th>Name</th><th>Size</th><th>Description</th></tr>
<tr><td>1</td><td>CAN Id</td><td>2 bytes</td><td colspan="2">If the Id matches with the Id of the MC01, the message will be read. Otherwhise, it will be ignored</td></tr>
<tr><td rowspan="4">2</td><td rowspan="4">Mode</td><td rowspan="4">1 byte</td><td colspan="2">It indicates what kind of order is receiving the MC01</td></tr>
<tr><td>Value</td><td>Order</td></tr>
<tr><td>0</td><td>Turn off</td></tr>
<tr><td>3</td><td>Move to the angular position written in Data</td></tr>
<tr><td>4</td><td>Move to the angular speed written in Data</td></tr>
<tr><td>3</td><td>Data</td><td>3 bytes</td><td colspan="2">• With Mode 3, it indicates the decoded angular position in radians
• With Mode 4, it indicates the decoded angular speed in radians per second</td></tr>
</table>

The parameter that is configured in the **MC01** is the **CAN Id**. To do it, use its respective PDI Builder and manual:

- For **MC01B** read the Input/Output section of the **MC01B PDI Builder** user manual.
- For **MC01S** read the Input/Output section of the **MC01S PDI Builder** user manual.