MC01B PDI Builder

Release 6.8.67

Embention

2024-06-13

CONTENTS

1	Quic	sk Start	3
	1.1	System Requirements	3
	1.2	Download	3
	1.3	Installation	4
2	Conf	figuration	9
	2.1	VMC Brushless	13
		2.1.1 Motor	13
		2.1.2 Status	14
	2.2	Input/Output	15
		2.2.1 SCIA Serial CAN	16
		2.2.2 Commgr Serial CAN	17
		2.2.3 CAN id used for CAN motor commands	19
	2.3	Control	19
		2.3.1 Control Brushless	19
		2.3.2 Serial	21
		2.3.3 Encoder	22
	2.4	Communications	23
	2.5	Telemetry	23
	2.0	Telementy	

ncoib Blilder

MC01B PDI Builder is an application employed to configure the MC01B, the brushless variant of MC01.

CHAPTER

QUICK START

MC01B PDI Builder is the main tool for setting all configurable parameters of the MC01B.

Configuration for Veronte MC01B should be created according to the needs of the belonging system.

Once MC01B has been detected on Veronte Link, download and install MC01B PDI Builder.

1.1 System Requirements

Before executing this software, users should check the following sections with the minimum and recommended PC hardware requirements.

Minimum requirements

- CPU: Intel Core i5-8365UE
- RAM: 8 GB DDR4
- STO: 256 GB SSD

Recommended requirements

- CPU: 12th Gen Intel(R) Core(TM) i7-12700H 14 cores up to 4,70 GHz
- RAM: 32 GB
- STO: 1 TB SSD M.2 NVMe PCIe

1.2 Download

Once a MC01B device has been purchased, a GitHub release should be created for the customer with the application.

To access to the release and download the software, read the Releases section of the **Joint Collaboration Framework** manual.

1.3 Installation

To install MC01B PDI Builder on Windows, execute MC01BPDIBuilder.exe and follow the steps:

1. Click on Next:



Fig. 1: Windows Installation - Step 1

2. Select the directory where where the software will be installed (with the Change button), then click on Next:

🖟 MC01BPDIBuilder-6.8.67 Setup	_		\times
Destination Folder Click Next to install to the default folder or click Char	nge to choose another.		Ð
Install MC01BPDIBuilder-6.8.67 to:			
C:\Program Files\Embention\MC01BPDIBuilder\ Change			
Bac	k Next	Can	cel

Fig. 2: Windows Installation - Step 2

3. Finally, click on **Install** (administrator rights are needed):



Fig. 3: Windows Installation - Step 3

4. After a few seconds, the following window will appear indicating the process was succesful. Click on Finish:



Fig. 4: Windows Installation - Step 4

Note: In case of any issue during installation, please disable Windows Defender and Firewall. To disable Firewall, go to "Control Panel" and "Firewall of windows", then click on **Turn off**.



Customise Settings		-	×
← → × ↑ 🔮 > Control Panel > System and Security > Windows Defender Firewall > Customise Settings	~ 0	Search Control Panel	P
Customise settings for each type of network			
You can modify the firewall settings for each type of network that you use.			
Private network settings			
Turn on Windows Defender Firewall			
Block all incoming connections, including those in the list of allowed applications			
✓ Notify me when Windows Defender Firewall blocks a new app			
(Interpretation of Windows Defender Firewall (not recommended)			
Public network settings			
O Turn on Windows Defender Firewall			
Block all incoming connections, including those in the list of allowed applications			
Notify me when Windows Defender Firewall blocks a new app			
Image: A state of the state			
OK Cancel			

CHAPTER

TWO

CONFIGURATION

This section explains each option and parameter available in MC01B PDI Builder.

Once the installation is finished, open MC01B PDI Builder and select the unit:



Fig. 1: MC01B ID

If it is correctly connected, it should appear in Normal mode, as shown in the following figure, or Maintenance mode:



Fig. 2: MC01B PDI Builder

MC01B unit can also appears as: Maintenance mode (loaded with errors) or Normal mode - Disconnected.

Note: Maintenance mode (loaded with errors) appears when something is wrong in the configuration.

The user can access now to 3 configuration options:

- **MC01B**: It allows the user to work with **offline** configurations. A previously exported MC01B PDI configuration can be opened and modified or it is also possible to build a new one from the default configuration.
- Upload PDI: A previously exported MC01B PDI configuration can be imported to the current MC01B flash memory.
- **Open MC01B**: By clicking on this option, **MC01B PDI Builder** configuration menu opens with the configuration (the PDI files) loaded in the MC01B. Then, the user can modify it online.

Note: PDI files are MC01B configuration files. These files allow for modular control with improved version management. These PDI files are split in 2 folders. Each folder hold several .xml files:

- vmc-brushless: Contains configuration for MC01B. All control system and parameters are stored here.
- **xsd**: This folder holds .xsd files. An XSD file is a definition file specifying the elements and attributes that can be part of an XML document. This ensures that data is properly interpreted, and errors are caught, resulting in appropriate XML validation. Users should never delete, replace or modify it.



Fig. 3: PDI files

Finally, click on 'Open MC01B' to open the configuration and start editing online.

VC01B PDI builder - (v.6.8.67)	_		×
MC01B			
Build PDI to configure your MC01B			
↓ Upload PDI	Open MC01B		
Upload PDI to the MC01B, this option can't be	Open PDI online and work with i	:	
undon.			
Maintenance mode	MC01 30313 (6.8	.93)	*

Fig. 4: Open MC01B

Note: When MC01B PDI is open, the unit changes to Maintenance mode.

The different 'buttons' that can be seen in the initial menu of the MC01B PDI Builder are explained below.

0
(5)

Fig. 5: Initial menu

1. Save PDI: After changes are done, press on the save button to apply the changes.

Note: This button will only appear if a MC01B is connected, i.e. when working offline this button will not be available.

2. **Export PDI**: After modifying a configuration, press the export button to store the configuration in the local storage.

Users can store this configuration in an empty folder or in the folder where the previously imported configuration is stored. With the latter option, the "original" configuration will be overwritten by the one with the new changes.

- 3. **Import PDI from repo**: The user can import a configuration file from the repo and modify it. After that, if the save button is pressed, this configuration will be uploaded to the connected MC01B.
- 4. **Import PDI from local storage**: The user can import a configuration file from the local storage and modify it. After that, if the save button is pressed, this configuration will be loaded into the connected MC01B.
- 5. Feedback: Users can report a problem they have encountered by creating an issue in their own 'Joint Collaboration Framework'. The 'Download' button downloads a zipped folder with the current MC01B configuration and more information needed for Embention to resolve the issue. It is advisable to attach this folder when creating the issue.

Note: The user's 'Joint Collaboration Framework' is simply a own Github repository for each customer.

If the user has any questions about this Joint Collaboration Framework, please see Joint Collaboration Framework user manual.



Fig. 6: Feedback window

6. These are the different functions of MC01B. They are explained in the following sections:





2.1.1 Motor

This section sets the electrical parameters of the motor that will define the dynamics of the controller.

VMC Brushless	MC01BPDI Builder - MC01 30313 - CONNE	ECTED	- x
Motor Maximum Iq 1.0 Status		O VMC Brushless	8 2 2 9
Time to hold Max Iq Image: Constraint of the constraint of t	Motor Status Image: Construction of the second of the sec	Maximum Iq 1.0 Time to hold Max Iq 0.0 s	

Fig. 7: Motor section

- **Maximum Iq**: Maximum permissible quadrature current that the motor can drive. It is a normalized value from *0A* to *3.75A*.
- Time to hold Max Iq: Defines the time while the maximum intensity (*Maximum Iq*) can be overpassed.

Warning: In case of overpassing the Maximum Iq during the stablished period, the MC01 will shut down.

2.1.2 Status

- Enable VCP Status Message enables the periodic sending of the status message that Veronte Link uses to recognise the Veronte MC01B.
- Period: Enter a desired period to send repeatedly the status message.

MC01BP	DI Builder - MC01 30313 - CONNEG	CTED	- x
		O VMC Brushless	8 2 2 9
O	Motor	Enable VCD Status Marsage	
	Status		
÷		Period 1.0 s	
=			
P			

Fig. 8: Status section

Note: VCP is the Veronte Communication Protocol. To know more, read the VCP user manual.

2.2 Contemport

MC01B can send and receive messages through a CAN bus with Serial protocol, this communication system is named as Serial-Over-CAN. This is necessary to establish the communication with a Veronte Autopilot 1x.

Note: TX corresponds to transmitted messages and RX to received messages.

First of all, it is necessary to set the appropriate **Baudrate** (i.e. the communication speed, expressed in bits per second) of the CAN bus.

2.2.1 SCIA Serial CAN

Serial messages received/sent through the SCIA (RS-232) port can be sent via CAN.

This menu internally configures the following connection:



Fig. 9: Input/Output menu - SCIA Serial CAN schema

MC01BP	DI Builder - MC01 30313 - CONNECTED					- x
		• <a>← Input / Output		8	4	0
0	Baudrate 1000	000.0 Bd				
•	SCIA Serial CAN					
E	тх	RX				
	Extended	Extended				
P	CAN 1303	CAN	1304			
	Timeout 6.7E-4 s	Number of mailboxes used for rx	12			
	Commgr Serial CAN					
	ТХ	RX				
	Extended	Extended				
	CAN 1301	CAN	1302			
	Timeout 6.7E-4 s	Number of mailboxes used for rx	12			
	CAN id used for CAN motor comm	ands				
	Extended					
	CAN 1					
	Number of mailboxes used for rx 1					

Fig. 10: Input/Output menu - SCIA Serial CAN section

• TX:

- **Extended**: If enabled, the frame format will be this, 'Extended', i.e. with a 29-bit identifier. Otherwise, the frame format 'Standard' (11-bit identifier) is set by default.
- CAN: CAN Id of the message to be sent must be set. The value set has to be decimal format.
- Timeout: This is the threshold time between receptions to consider that it is not being received correctly.
- RX: Configuration for receiving messages (as Mailboxes in the Veronte Autopilot 1x configuration)
 - Extended: Select this option for 29-bit IDs.
 - CAN: 11-bits (Standard) or 29-bits (Extended) ID used to identify the CAN message to be received. The value set has to be decimal format.
 - Number of mailboxes used for rx.

2.2.2 Commgr Serial CAN

Serial messages received/sent through the Commgr port can be sent via CAN.

Warning: This configuration is already done by default in order to allow communication via CAN between 1x Autopilot and MC01B. It must not be modified.

This menu internally configures the following connection:



Fig. 11: Input/Output menu - Commgr Serial CAN schema

MC01BF	MC01BPDI Builder - MC01 30313 - CONNECTED - ×							
	•🛟 Input / Output							
() •~•	Baudrate 1000 SCIA Serial CAN	000.0 Bd						
	TX Extended CAN 1303 Timeout 6.7E-4 s	RX Extended CAN I304 Number of mailboxes used for rx 12						
	Commgr Serial CAN TX Extended CAN 1301 Timeout 6.7E-4 S	RX Extended CAN I302 Number of mailboxes used for rx 12						
CAN id used for CAN motor comm Extended CAN 1 Number of mailboxes used for rx 1		<u>iands</u>						

Fig. 12: Input/Output menu - Commgr Serial CAN section

• TX:

- **Extended**: If enabled, the frame format will be this, 'Extended', i.e. with a 29-bit identifier. Otherwise, the frame format 'Standard' (11-bit identifier) is set by default.
- CAN: CAN Id of the message to be sent must be set. The value set has to be decimal format.
- Timeout: This is the threshold time between receptions to consider that it is not being received correctly.
- RX: Configuration for receiving messages (as Mailboxes in the Veronte Autopilot 1x configuration)
 - Extended: Select this option for 29-bit IDs.
 - CAN: 11-bits (Standard) or 29-bits (Extended) ID used to identify the CAN message to be received. The value set has to be decimal format.
 - Number of mailboxes used for rx.

2.2.3 CAN id used for CAN motor commands

MC01B requires a CAN id to send the desired position to the motor.

MC01BP	DI Builder - MC01 30313 - CONNECTED			- ×
		🚓 Input / Output		
Ö	Baudrate 1000	000.0 Bd		
•	SCIA Serial CAN			
	тх	RX		
	Extended	Extended		
P	CAN 1303	CAN	1304	
	Timeout 6.7E-4 s	Number of mailboxes used for rx	12	
	Commgr Serial CAN			
	ТХ	RX		
	Extended	Extended		
	CAN 1301	CAN	1302	
	Timeout 6.7E-4 s	Number of mailboxes used for rx	12	
	CAN id used for CAN motor comm	<u>iands</u>		
	Extended			
	CAN 1			
	Number of mailboxes used for rx 1			

Fig. 13: Input/Output menu - CAN id used for CAN motor commands section

- **Extended**: If enabled, the frame format will be this, 'Extended', i.e. with a 29-bit identifier. Otherwise, the frame format 'Standard' (11-bit identifier) is set by default.
- CAN: CAN Id through which the MC01B will send the position of the motor.
- Number of mailboxes for rx.



2.3.1 Control Brushless

First, the PI (Proportional and Integral) controller is presented.

The form of the PI is the classical parallel form:

$$u = K_p \cdot \left(1 + \frac{1}{T_i} \cdot s\right)$$

Where:

- K_p is the **Proportional loop gain**.
- Integral gain refers to the quotient $\frac{1}{T_i}$.
- Lower and Upper saturation gain are the limits to which the PI limit its output.

Control Brushless is the main control menu, where users must enter the control parameters according to their motor.

The basic blocks that define the control are four *PI control loops* that should be tuned with the motor characterization.

MC01BPDI Builder - MC01 30313 - CONNE	CTED			- ×
	ŧco	ontrol		8 2 2 9
Control Brushless	Quadrature current cor	ntroller	Direct current controlle	r
Encoder	Proportional loop gain	9.0	Proportional loop gain	5.5
	Integral gain	0.33	Integral gain	0.2
	Lower saturation gain	-0.8	Lower saturation gain	-0.4
	Upper saturation gain	0.8	Upper saturation gain	0.4
	Speed controller		Position controller	
	Proportional loop gain	0.18	Proportional loop gain	22.0
	Integral gain	0.04	Integral gain	0.0
	Lower saturation gain	-0.4	Lower saturation gain	-8.0
	Upper saturation gain	0.4	Upper saturation gain	8.0
	Max falling rate -10	000.0		
	Max rising rate 100	00.0		
	Speed filter time constant	0.01		



Both quadrature and direct current PI control must be defined. Also, the quadrature and direct PI gain settings are usually the same, but it will depend on whether the motor parameters are the same on both axes.

- Quadrature current controller: Torque regulator. It is controlled by a *PI control*.
- Direct current controller: Flux regulator. This is controlled by a PI control.
- Speed controller: It is controlled by a *PI control*.
- Position controller: It is controlled by a *PI control*.

Maximum desired motor acceleration:

- Max falling rate: Deceleration rate. Expressed in rad/s^2 .
- Max rising rate: Acceleration rate. Expressed in rad/s^2 .

• Speed filter time constant: This is the time constant of the first order filter of the speed calculation. The lower value, the lower filtering.

The time constant is approximately defined as:

$$\tau = \frac{1}{2 \pi f_c}$$

Where:

– τ : time constant

– f_c : cutoff frequency

2.3.2 Serial

MC01B can only use one serial peripheral, SCIA port, the serial parameters that fit the serial protocol requirements can be edited in this menu:

MC01BP	DI Builder - MC01 30313 - CONNE	CTED			- ×
		\$	E Control		8 2 2 9
\bigcirc	Control Brushless	Functionality			
	Serial	Tunctionality			
•	Encoder	Baudrate	57600	•	
Ŧ		Length	8	•	
		Stop	1	•	
¥2		Parity	Disabled	•	
<u> </u>		Use address	mode		

Fig. 15: Serial section

- Baudrate: This field specifies how fast data is sent over a serial line.
- Length: Defines the number of data bits for each character: 4 to 8 bits.
- Stop: Number of stop bits sent at the end of each character: 1, 1.5 or 2.

• **Parity**: Method to detect errors during transmission. When parity is used with a serial port, an extra data bit will be sent with each data character.

The bits of each character (including parity bit) will be even or odd according to **parity** mode (**odd**, **even** or **disabled**).

• Use address mode: 9-bit data framing uses the bit typically associated with parity error detection to identify address messages. Sent serial data that does not have the address bit set will be ignored (unless the device had previously identified an address message associated with it).

This option can be disabled or enabled.

2.3.3 Encoder

In this section, the user must define the number of bits employed by the encoder to communicate the position of the motor.

MC01BPDI Builder - MC01 30313 - CONNECTED - ×							
		幸 Control	8 2 6 6				
Ο	Control Brushless	Number of Lite 10					
	Serial	Number of bits:					
•~~	Encoder						
=							
P							
1.1.1							

Fig. 16: Encoder section

2.4 Communications

This menu configures the SCIA port parameters in the same way as in the Serial section of the Control menu.

Note: Modifications carried out here will be applied in the parameters of the serial menu and vice versa.

0

Fig. 17: Communications menu

2.5 🛄 Telemetry

In the Telemetry menu, the user chooses the variable or group of variables to be sent via the CAN bus. The following items can be configured:

MC01BPDI Builder	- MC01 30313 - CONNECTED		- ×
		lılı Telemetry	8 1 2 5 0
	EXT Can ID:	1434 Little endian Period 1.0 s	
		Memory usage: 64 / 512000 bytes	

Fig. 18: Telemetry menu

By clicking on the + icon more messages can be added and by clicking on the - icon, messages can be deleted. Messages are configured clicking on the icon:



Fig. 19: Telemetry menu - Custom messages example

Since this section works in a similar way to the CAN Custom Message configuration in the **1x PDI Builder software**, the explanation to configure the telemetry messages via CAN can be found in the CAN Setup -> Input/Output section of the **1x PDI Builder user manual**.