4x PDI Builder

Release 6.12.27

Embention

2024-06-14

CONTENTS

1	Quic	k Start	3
	1.1	System Requirements	3
	1.2	Download	3
	1.3	Installation	4
2	Conf	iguration	5
	2.1	Arbiter	12
		2.1.1 GPIO	12
		2.1.2 Initial time	14
		2.1.3 Status	14
	2.2	Input/Output	15
		2.2.1 SCI	15
		2.2.2 I/O Setup	17
		2.2.3 CAN	18
		2.2.4 CAN I/O	19
		2.2.4.1 Configuration	19
		2.2.4.2 Custom message	22
		2.2.5 CAN Setup	23
	2.3	Communications	24
	2.4	Arbitration	26
		2.4.1 Config	27
		2.4.1.1 Absolute Arbitration Variables	27
		2.4.1.2 Relative Arbitration Variables	27
		2.4.1.3 Arbitration Example	27
		2.4.1.4 Config menu	27
		2.4.2 CAN Setup	29
3	Integ	tration examples	31
	3.1	Arbitration configuration	31
	3.2	CAN tunnel	37
	3.3	GPIO command	43
		3.3.1 1x PDI Builder side	43
		3.3.2 4x PDI Builder side	45
4	Trou	hleshooting	40
-	4 1	Maintenance mode	4 0
	<u>4</u> 2	Maintenance mode (loaded with errors)	50
	 43	Migrate configuration	52
	т.Ј		52
5	FAQ		53

5.1	How to calculate a mask											•	•	 •		•			53

4>< Builder

4X PDI Builder is an application for modifying, generating and uploading PDI files for **Arbiters**, the autopilot managers of the redundant **Veronte Autopilot 4x**.

Warning: Select your select a version from an	version before reading any user manual f y Embention user manual.	for software. The following	image shows where to
🧟 Veronte Link 🗙 🕂			~ - ¤ ×
\leftrightarrow \rightarrow C \triangleq manuals.embention.com/ver	onte-link/en/6.12.9/index.html		• •
		la Home Version-6.12.9 →	Languages-EN - Download -
٩	Docs » Veronte Link	1	Quick Start →
Quick Start Operation Integration examples Troubleshooting	Veronte Link		
			usly.

CHAPTER

QUICK START

4x PDI Builder is the main configuration tool to adapt **Arbiters** for a specific system, being its main function to decide which **Autopilot 1x** controls the vehicle at every moment. **4x PDI Builder** includes:

- Communications: through general purpose CAN bus, RS-232, RS-485 and GPIOs.
- Arbitration criterion: **Arbiters** decide which **Autopilot 1x** controls the vehicle according to the criterion decided by the user.

All Veronte devices, including Arbiters, are configured with PDI files.

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 Veronte Autopilot 4x 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

Warning: If users have any problems with the installation, please disable the antivirus and the Windows firewall. Disabling the antivirus depends on the antivirus software.

To disable the firewall:

- Go to "Control Panel" \rightarrow "System and Security" \rightarrow "Windows Defender Firewall"
- Then, click on "Turn windows Defender Firewall on or off".

					_	
→ Y ↑ 🐨 System ar	nd Security > Windows Defender Firewall	~ [©]	Search Control Panel			
Control Panel Home	Help protect your PC with Windows D	efender Firewal				
Control Parlet Home	Windows Defender Firewall can help to prevent h	ackers or malicious s	oftware from gaining access to your PC			
Allow an app or feature through Windows Defender	through the Internet or a network.		5 5 7			
Firewall	Private networks		Connected 🛇			
Change notification settings						
Turn Windows Defender Firewall on or off	Guest or public networks		Not connected \bigotimes			
Restore defaults						
Advanced settings						
Troubleshoot my network						
Curtonia Stilica	115.1. •••					
Customise Settings						
	aner 7 System and Security 7 Windows Defender in	ewait > Custonnise	Jettings	V U	Search Control Parler	
	Customise settings for each typ	e of network				
	You can modify the firewall settings for e	ach type of network t	hat you use.			
	Private network settings		-			
	Turn on Windows Defender	Firewall				
	V ~					
	Block all incoming conn	ections, including the	ose in the list of allowed applications			
	□ Block all incoming conn ☑ Notify me when Window	ections, including the	ose in the list of allowed applications llocks a new app			
	 Block all incoming conn Notify me when Window Outing Windows Defender 	ections, including the /s Defender Firewall b Firewall (not recomm	ose in the list of allowed applications locks a new app mended)			
	Block all incoming conn Notify me when Window Offurn off Windows Defender Public network settings	ections, including the /s Defender Firewall b Firewall (not recomm	ose in the list of allowed applications locks a new app nended)			
	Block all incoming conn Notify me when Window Offurn off Windows Defender Public network settings O Turn on Windows Defender Rock all incoming conn	ections, including the vs Defender Firewall b Firewall (not recomm Firewall	ose in the list of allowed applications locks a new app mended)			
	Block all incoming conn Notify me when Window Offurn off Windows Defender Public network settings O Turn on Windows Defender Block all incoming conn Votify me when Window	ections, including the s Defender Firewall b Firewall (not recomr Firewall ections, including the s Defender Firewall b	ose in the list of allowed applications locks a new app nended) ose in the list of allowed applications locks a new app			
	Block all incoming conn Notify me when Window Orurn off Windows Defender Public network settings O Turn on Windows Defender Block all incoming conn Wolfy me when Window	ections, including the vs Defender Firewall (Firewall (not recomm Firewall ections, including the vs Defender Firewall (Firewall (not recomm	se in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	Block all incoming conn Notify me when Window Offurn off Windows Defender Public network settings O Turn on Windows Defender Block all incoming conn Notify me when Window Offurn off Windows Defender	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	ose in the list of allowed applications locks a new app nended) ose in the list of allowed applications locks a new app nended)			
	 Block all incoming conn Notify me when Window OTurn off Windows Defender Public network settings Turn on Windows Defender Block all incoming conn Notify me when Window OTurn off Windows Defender 	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	ose in the list of allowed applications locks a new app nended) ose in the list of allowed applications locks a new app nended)			
	 Block all incoming conn Notify me when Window OTurn off Windows Defender O Turn on Windows Defender O Turn on Windows Defender Block all incoming conn Notify me when Window OTurn off Windows Defender 	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	ose in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 Block all incoming conn Notify me when Window OTurn off Windows Defender Public network settings Turn on Windows Defender Block all incoming conn Notify me when Window OTurn off Windows Defender 	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	sse in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 Block all incoming conn Notify me when Window Image: State of the stat	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	ose in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 Block all incoming conn Notify me when Window Image: State of the stat	ections, including thi /s Defender Firewall (Firewall (not recomm Firewall ections, including the /s Defender Firewall (Firewall (not recomm	ose in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 Block all incoming conn Notify me when Window Image: State of the stat	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	ose in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 □ Block all incoming conn ☑ Notify me when Window ② ③Turn off Windows Defender Public network settings ○ Turn on Windows Defender □ Block all incoming conn ☑ Notify me when Window ③ ③Turn off Windows Defender 	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm	sse in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 Block all incoming conn Notify me when Window 	ections, including the s Defender Firewall (rirewall (not recomm Firewall ections, including the s Defender Firewall b Firewall (not recomm	sse in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended)			
	 Block all incoming conn Notify me when Window 	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm Firewall (not recomm)	ose in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended) OK Cancel	-		
	 Block all incoming conn Notify me when Window 	ections, including the s Defender Firewall (Firewall (not recomm Firewall ections, including the s Defender Firewall (Firewall (not recomm Firewall (not recomm	ose in the list of allowed applications locks a new app mended) ose in the list of allowed applications locks a new app mended) OK Cancel			

CHAPTER

CONFIGURATION

This section explains each option and parameter available in **4x PDI Builder**.

Once the installation is finished, connect the **Management Board** of the 4x (using the **Connector 4**) with Veronte Link.

Then, open 4x PDI Builder and select the Arbiter to configure.

4xVeronte PDI Builder				-		×
4xVeronte						
Build PDI to configure your 4xVeronte						
🛨 Upload PDI		生 Ореі	1 4xVero	onte		
Upload PDI to the 4xVeronte, this option can't be undone		Open PDI o	nline and wo	rk with it		
Not connected	JL		Not sele	rted		•
Hor connected			Not selec	ted		
			Arbiter v	1.8 51090	(6.12.43)	

Fig. 1: Arbiter ID

If the **Veronte Autopilot 4x** is correctly connected, **4x PDI Builder** will display the **mode** in which the connected unit is. In addition, a **PDI error button** will appear:



Fig. 2: 4x PDI Builder

• 4x mode: The 4x unit should appear in Normal mode, as shown in the figure above, or Maintenance mode.

It can also appear as Maintenance mode (loaded with errors) or Normal mode - Disconnected.

Note: Maintenance mode (loaded with errors) appears when something is wrong with the configuration. For more information, see *Maintenance mode* (loaded with errors) - *Troubleshooting* section of this manual.

PDI Errors button: The user can check if the connected unit has PDI Errors by simply clicking on it. If there are no errors, the following message is displayed:

	×
No errors	i
No PDI Errors found.	
	ОК

Fig. 3: 4x PDI Builder - PDI Errors message

The user can access now to three configuration options:

- 🗆 X
生 Open 4xVeronte
Open PDI online and work with it

Fig. 4: 4x PDI Builder options

• **4xVeronte**: It allows the user to work with **offline** configurations. A previously exported 4x PDI configuration can be opened and modified. It is also possible to build a new one from the default configuration.

Note: When an offline configuration is opened, it is possible to select the hardware version to work with on the marked button. Versions **1.8** A and **1.8** B are arbiters A and B of version 1.8 respectively.

It is recommended to select the same hardware version that the user's Arbiter has, so that if the user wishes to load this configuration later to the Arbiter, there will be no compatibility problems.

4xVeronte PDI Builder		— ×
Arbiter hw rev. 1.0 👻	4xVeronte PDI Builder	* & 5 0
Arbiter hw rev. 1.0		
Arbiter hw rev. 1.2		
Arbiter hw rev. 1.8 A		
Arbiter hw rev. 1.8 B		
ST		

Fig. 5: Hardware versions selector

When switching between hardware versions, the following confirmation message appears:



Fig. 6: Hardware versions selector - Confirmation message

• Upload PDI: A previously exported 4x PDI configuration can be imported to the linked Arbiter (PDI files are transferred from computer to Arbiter).

Warning: When a configuration is loaded into Arbiter with a version older than the software version being used, an **automatic migration** from the configuration version to the software version being used will be performed.

For more information on this, see Migrate configuration - Troubleshooting section of this manual.

• **Open 4x Veronte**: It opens the configuration (the PDI files) loaded in the **Arbiter** (PDI files are transferred from **Arbiter** to computer). Then, the user can modify it **online**.

Note: PDI files are Arbiter configuration files. These PDI files are stored in the *setup* folder, holding several .xml files which contain all the control system and parameters.



Fig. 7: PDI files

Finally, click on '**4xVeronte**' to edit a configuration offline or '**Open 4xVeronte**' to open the current configuration and start editing it online.

Note: The 4x unit must enter **Maintenance mode** for the user to start editing, so it is necessary to accept the confirmation panel below.

	1 4xVeronte PDI Builder		_		×
	Build PDI to configure your 4xVeronte				
Confirmation X					
Maintenance mode					
Do you want to enter in maintenance mode? OK Cancel	▲ Upload PDI Upload PDI to the 4xVeronte, this option can't be undone	Open 4xVeront Open PDI online and work with	e ith it		
	Maintenance mode	Arbiter v1.8 5	51091 (6.1	2	*

Fig. 8: Open 4xVeronte

Once in Maintenance mode, the user can access the *Initial menu*. The different 'buttons' that can be seen in this menu of the **4x PDI Builder** are explained below.

4xVerontePDI Builder - Arbiter v1.2 5109	0 - CONNECTED	- ×
Arbiter hw rev. 1.8 A	() Arbiter	8 2 5 9
Arbiter hw rev. 1.8 A so ~	O Arbiter	1234 5

Fig. 9: Initial menu

1. Save and close: After changes are done, press here to apply changes.

While saving, a percentage of saving process is displayed.

4xVeron	tePDI Builder - Arbiter v1.2	2 51091 - CONNECTED			*** X
	v		•∕• Input / Output	I III	
\bigcirc	SCI	Configuration	Custom message		
	I/O Setup	Priority	Producer		Consumer
• ~			B Serial to CAN 1	Output filter 1	
	CAN Setup		B Serial to CAN 2	None	
			B Serial to CAN 3	None	
			Serial to CAN 4	→ None	
			Serial to CAN 5	None	
				None	
			Serial to LAN 7	None.	
			Serial to CAN 8	None	
			Arbiter	Output filter 2	
			B Input filter 1		
		🔨 High 🔇	b Input filter 2		
		igh 🗘	b Input filter 3		
		High 👯	b Input filter 4	None	
		High	nput filter 5	None	
			p Input filter 6	None	
			b Input filter 7	None	
			binnut filter 8	None	

Fig. 10: Save and close

In order to save the configuration in the 4x unit it is necessary to **RESET**, therefore the **4x PDI Builder** software will **close**. For this reason, the user must accept the following panels:

CONFI	RMATION		×
Your dev	ice will be restarted after saving, Do you s	till want to save?	?
CONFIRM	ATION		
	_	ОК	Cancel
		×	
	Configuration saved	Í	
	The configuration is saved in the 4xVeronte		
	_	ОК	
		×	
	Message	Í	
	Your device has been successfully restarted.		
		ОК	

Fig. 11: Save and close - Confirmation panels

Danger: As the Arbiter is reset, it is not advisable to save changes during flight tests.

Note: This button will only appear if an **Arbiter** is connected. When working offline this button will not be available.

- 2. **Export PDI**: After modifying a configuration, press here to save the configuration in the local storage (both folders with the PDI files are downloaded). 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 repository**: The user can import a configuration file from the GitHub repository and modify it. After that, if the **Save and close** button (1) is pressed, this configuration will be uploaded to the connected **arbiter**.
- 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 and close** button (1) is pressed, this configuration will be loaded into the connected **arbiter**.
- 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 **arbiter** 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 or contact sales@embention.com.

dBack	•
Sen	d us your feedback
Repo:*	Your repo
Title:	

Fig. 12: Feedback

6. These are the different functions of the **arbiter**. They are explained in the following menus.





2.1.1 GPIO

In this window, each individual GPIO (General Purpose Input/Output) behavior can be configured:

4xVeron	te PDI Builder						- x
Arbiter I	nw rev. 1.8 B 🔹		C) Arbiter		20	•
\bigcirc	GPIO	Signal	GPIOId	10	Pull-up	Function	Qsel
	Initial time	I/O1	GPIO 0	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
•	Status	I/O2	GPIO 2	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O3	GPIO 4	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
Ŷ		I/O4	GPIO 6	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O5	GPIO 8	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O9	GPIO 5	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O10	GPIO 7	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync

Fig. 13: GPIO panel

- Signal: Pin ID as described in Arbiter connector pinout Hardware Installation section of the 4x Hardware Manual.
- GPIOId: GPIO ID of the microcontroller.
- IO: Define GPIO as an input or ouput.
- Pull-up: Enable or disable the pull-up resistance.
- Function: Mux 0/GPIO, Mux 1, Mux 2 or Mux 3. These are the different functionalities that the GPIO can have, depending on the multiplexer.
- **Qsel**: This is the "input qualification", it is used to control how the value of a GPIO is evaluated. The available options are:
 - **Sync**: The value is taken as the time it is checked (synchronously). This is the default mode of all GPIO pins.
 - 3 Samples: The value is checked 3 times and the value is only changed when the 3 times are the same.
 - 6 Samples: Same as 3 samples, but checking 6 times instead of 3.
 - ASync: No checks are performed. It is used when it is not used as GPIO.

2.1.2 Initial time

If any **Autopilot 1x** has not power supply, the **Management Board** will detect it as a fault. The **Initial time** parameter is the margin of time from the moment when the **Management Board** is powered up until it starts cheking power supplies.

With this parameter it is possible to connect each 1x one by one after the Management Board.

4xVeronte PDI Builder		- ×
Arbiter hw rev. 1.8 B	() Arbiter	7 7 7 6 0
GPIO Initial time	Initial time: 0.0 s	
Status		
		

Fig. 14: Initial time panel

2.1.3 Status

- Enable VCP Status Message: Enables the periodic sending of the status message that Veronte Link uses to recognize an arbiter.
- **Period:** Establishes the desired period to send repeatedly the status message.

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

4xVeronte PDI Builder		- ×
Arbiter hw rev. 1.8 B	() Arbiter	
GPIO Initial time Status	Period 1.0 s	

Fig. 15: Status panel



2.2.1 SCI

Arbiters can use up to three serial SCIs (Serial Communication Interface). Serial ports A, B and C parameters can be edited in this menu to fit the serial protocol requirements.

Note: SCI A corresponds to RS-232 port A (RS232-1), SCI B to RS-232 port B (RS232-2) and SCI C to RS-485.

4xVeronte PDI Builder		- x
Arbiter hw rev. 1.8 B	•<∎ Input / Output	* * * 0
SCI I/O Setup	SCI A SCI B SCI C	
CAN CAN I/O	Functionality	
CAN Setup	Baudrate 115200 -	
	Stop 1	
	Parity Disabled -	
	Use address mode	

Fig. 16: SCI panel

- **Baudrate**: How fast data is sent over a serial line.
- Length: 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**: Is a method of detecting errors in transmission. When parity is used with a serial port, an extra data bit is sent with each data character, arranged so that the number of 1 bits in each character, including the parity bit. **Disabled**, **odd** or **even**.
- 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.

2.2.2 I/O Setup

In this panel the user can establish the relationship between a determined signal with a I/O port. This allows users to configure external sensors, custom messages, etc.

4xVeron	te PDI Builder						505450545054505	- ×
Arbiter I	nw rev. 1.8 B 🔹	•	¢	Input / Output			*	
Ö	SCI	Configurati	on					
	I/O Setup	Priority		Producer		Consumer		
•4	CAN	High I	Q ⁰	RS232 1	\leftrightarrow	Commgr port 1	00	â
6	CAN I/O	High H	Q ⁰ ₀	RS232 2	\leftrightarrow	Commgr port 2	00	
	CAN Setup	High	Q ₀ ⁰	RS485	→	Commgr port 3	00	
-		High	00	Commgr port 0		Serial to CAN 0	Q0	
		High H	Q ₀ ⁰	Commgr port 1	↔	RS232 1	00	
		High H	Q ⁰	Commgr port 2	→	RS232 2	00	
		High H	Q ⁰	Commgr port 3	\leftrightarrow	RS485	00	
		High H	Q ₀ ⁰	Commgr port 4	\rightarrow	None	00	
		High H	Q ⁰ ₀	Commgr port 5	\rightarrow	None	00	
		High H	Q ⁰	CAN to serial 0	\rightarrow	Commgr port 0	00	
		High H	00	CAN to serial 1	\rightarrow	None	00	
		High H		CAN to serial 2	\rightarrow	None	00	
		High H	Q ⁰	CAN to serial 3	\rightarrow	None	00	
		High H	Q ⁰	CAN to serial 4	\rightarrow	None	00	
		High H	00	CAN to serial 5	\rightarrow	None	00	U
		High H	00	CAN to serial 6	\rightarrow	None	00	
		High H	38	CAN to cerial 7		None	88	~

Fig. 17: I/O Setup panel

- **Priority**: Certain connections between I/O ports can be marked with **high priority** with this checkbox. If it is enabled, they will **run at high frequency: 1000 Hz**.
- **Producer**: Functions to create and send messages.
- Consumer: Functions to receive and parse messages.

The following are the steps to setting up reception or transmission between ports:

- 1. Choose the **Producer** to use.
- 2. To configure the desired **Consumer** that will be bind to the chosen **Producer**, it is first required to establish the **relationship** between them:
 - Bind \rightarrow : Unidirectional relationship.
 - **Bind Bidirectional** \leftrightarrow : Bidirectional relationship. This enables a port to receive or send information.

Note: Once the Consumer has been selected, it is possible to undo the selection by pressing the Clear button.

3. Select the desired **Consumer** element.

The following I/O ports are available:

Field	Description
RS232 1	Serial Port 232 A
RS232 2	Serial Port 232 B
RS485	Serial Port 485
Commgr port	COM Manager ports send and receive VCP messages. This is the protocol used by
	Veronte products to communicate. For more information, read the VCP user manual
CAN to serial / Serial to	Serial to CAN sends serial streams over a CAN Bus / CAN to serial undoes the
CAN	transformation 'Serial to CAN'
CAN wrapper for serial	CAN wrapper for serial transmission sends CAN streams over a serial Bus / CAN
transmission / CAN	message unwrapper for serial transmission undoes this transformation. For more
message unwrapper for	information on these ports, please refer to CAN wrapper/CAN unwrapper - Input/Output
serial transmission	section of the 1x PDI Builder user manual

Warning: By default, RS232 1-2 are set up for Arbiter configuration.

If these connections are removed, it will not be possible to communicate with the **Arbiter** in **Normal mode**. In this situation, force the **arbiter** into **Maintenance mode** in order to recover the communication.

2.2.3 CAN

A CAN (Controller Area Network) Bus is a robust standard communication protocol for vehicles widely used in the aviation sector. The **Management Board** has two CAN buses that can be configured independently.

The structure of a CAN message can be seen in the following image:



Fig. 18: CAN message structure

Only the ID is introduced in the system, the rest of the message layout is already coded. The data field is built by the user to send, and parsed when received.

For more information on the CAN Bus protocol, see CAN Bus protocol section of the 4x Software Manual.

The baudrate of both CAN buses can be configured in the Baudrate box.

4xVeron	nte PDI Builder		- ×
Arbiter	hw rev. 1.8 B 🔹	• 😭 Input / Output	* * * •
	SCI I/O Setup CAN CAN I/O CAN Setup	Can A Can B Baudrate 500000.0 Bd	

Fig. 19: CAN panel

2.2.4 CAN I/O

2.2.4.1 Configuration

This panel allows the configuration of CAN communications between different devices.

4xVeron	ite PDI Builder						- x
Arbiter I	hw rev. 1.8 B 🔹	•	 Input / Output 			*	6
0	SCI	Configuration	Custom message				
	I/O Setup	Priority	Producer		Consumer		
•	CAN	High 🛱	Serial to CAN 0	\rightarrow	Output filter 0	Q0	â
	CAN I/O	🗌 High 🔀	Serial to CAN 1	\rightarrow	None	Q0	
	CAN Setup	High	Serial to CAN 2	\rightarrow	None	08	
<u>ش</u>		📄 High 🕼	Serial to CAN 3	\rightarrow	None	Q0	
		High 🛱	Serial to CAN 4	\rightarrow	None	Ø6	
		High 🛱	Serial to CAN 5	\rightarrow	None	Q0	
		🗌 High 🛱	Serial to CAN 6	\rightarrow	None	Q0	
		🗌 High 🛱	Serial to CAN 7	\rightarrow	None	Q0	
		🔲 High 🕰	Arbiter	\rightarrow	Output filter 1	Ø0	
		🗌 High 🛱	Input filter 0	\rightarrow	CAN to serial 0	Ø0	
		🗌 High 🛱	Input filter 1	\rightarrow	Arbiter	Q0	
		High 🛱	Input filter 2	\rightarrow	None	¢6	
		High 🛱	Input filter 3	\rightarrow	None	Q0	U
		🗌 High 🛱	Input filter 4	\rightarrow	None	Ø0	
		High 🛱	Input filter 5	\rightarrow	None	Q0	
		High 🛱	Input filter 6	\rightarrow	None	Q0	
		High 68	Innut filter 7		None	252	~

Fig. 20: Configuration panel

In this panel, the user can find the same 'columns' (**Priority**, **Producer** and **Consumer**) as in the *I/O Setup panel*. In addition, the process for configuring producers and consumers is also the same as described in the *I/O Setup* - *Input/Output* section.

Warning: In CAN, in Low state the specified period is not guaranteed, but it is guaranteed with High state.

However, only those **messages that are critical for external devices** should be set as **high priority**, as this may disrupt the proper functioning of the **arbiter**.

Producers:

• Arbiter: It sends a status and a score mesage.

To know more, read Arbitration messages - CAN Bus protocol section of the 4x Software Manual.

• Serial to CAN: Serial messages through CAN output, it has to be connected to *I/O Setup* consumer. It can be configured by clicking on a pop-up window will appear:



Fig. 21: Serial to CAN configuration

- Id: CAN Id must be set and it is used to identify messages. The value set has to be decimal format.
- **Extended**: If it is enabled, the frame format will be 'Extended', i.e. with a 29-bit identifier. Otherwise, the frame format 'Standard' (11-bit identifier) is set by default.
- Time out: This is the threshold time between receptions to consider that it is not being received correctly.
- **Input filter**: Those CAN messages received in one filter can no longer be received in subsequent filters. The following parameters need to be configured by clicking on

A	n	
Port	BOTH	*
Id	1302	
Mask	2047	dec
Filter type	Standard	*

Fig. 22: Input filter configuration

- **Port**: It is required to configure the CAN bus from which it listens, the user can choose between *CAN A*, *CAN B* or *BOTH*.
- Id: CAN Id must be set and it is used to identify messages. The value set has to be decimal format.
- Mask: A CAN Id mask can be set to filter messages. The mask defines which bits of the message id (in binary) should match.

For example, to admit standard Ids (11 bits) from 8 to 11 (100 to 111 in binary) the user should set the mask to binary 1111111100, that is 2044 in decimal.

Warning: Make sure that mask is set properly to be able to receive the desired CAN messages.

The mask should be 11 bits for Standard frame format and 29 bits for Extended frame format.

More information about this can be found in *How to calculate a mask - FAQ* section of this manual.

- Filter type: The available options are *Standard* (frame format with a 11-bit identifier), *Extended* (frame format with a 29-bit identifier) and *Both*.
- **Custom messages**: CAN custom messages transmission. They are configured in the next panel, explained in thr *Custom message* section of the present manual.
- CAN unwrapper: This undoes the 'CAN wrapper' action, it has to be connected to *I/O Setup* consumer (*CAN unwrapper for serial transmission*)..

Consumers:

• Arbiter: It sends ready and arbitration messages.

To know more, read Arbitration messages - CAN Bus protocol section of the 4x Software Manual.

- CAN to serial: This undoes the 'Serial to CAN' action, it has to be connected to I/O Setup producer.
- Output filter: CAN output filters. The user can choose between *CAN A*, *CAN B* or *BOTH* in the configuration button



Fig. 23: Output filter configuration

- **Custom messages**: CAN custom messages reception. They are configured in the next panel, explained in thr *Custom message* section of the present manual.
- CAN GPIO consumer: CAN message from Veronte Autopilot 1x for GPIO inputs. An example of how to implement it can be found in the *GPIO command Integration examples* section of this manual.
- CAN wrapper: CAN message through serial output, it has to be connected to *I/O Setup* producer (*CAN wrapper for serial transmission*).

Warning: The **Input filter** connected to the **Arbiter consumer** must have a mask that allows all arbitration CAN IDs.

2.2.4.2 Custom message

In this tab, the user chooses the variables to be sent/received through CAN buses. The following elements can be configured:

- **TX Ini**: It is used to configure transmitted messages that are only sent once at the beginning of the operation (sent when the **arbiter** boots up). They can be used to initialize some devices.
- TX: Configuration for transmitted messages.
- **RX**: Configuration for the reception messages (where they are stored).

Warning:

- The maximum capacity of a CAN message is 64 bits (8 bytes), so to send more information it must be divided into several messages.
- Each arbiter has a CAN limitation of 40 TX messages, 40 TX Ini messages and 80 RX messages. In addition, the following limits apply:
 - Maximum number of vectors (fieldset): 104
 - Maximum number of fields: 2000

4xVeronte PDI Builder	- ×
Arbiter hw rev. 1.8 B	• Input / Output
SCI I/O Setup CAN CAN I/O CAN Setup	Configuration Custom message TX Ini O1 - EXT Can ID: O Little endian Delay 1.0 s + X KX
	Memory usage: 0 / 2000 Fields

Fig. 24: Custom message panel

Since this section works in a similar way to the CAN Custom Message configuration in the **1x PDI Builder** software, the explanation on how to configure the telemetry messages via CAN is reflected in the Custom Messages - Input/Output section of the **1x PDI Builder** user manual.

2.2.5 CAN Setup

In this screen, users can configure the baudrate and reception mailboxes of each CAN Bus.

Since the **arbiter** is going to receive data on the CAN Bus, it is mandatory to configure a certain number of mailboxes to store that data until the arbiter reads it.

A mailbox can be configured for **multiple CAN message IDs** as long as the mask is configured correctly and these messages are **sent spaced out** with enough time between them to allow the high priority core to read each one individually. More information on masks can be found in *How to calculate a mask - FAQ* section of this manual.

Warning: Since 4x PDI Builder allows up to 32 mailboxes, users should make sure to leave at least one mailbox free for transmission (TX).

If any mailbox is **full** and another message arrives, the **new** message is **discarded**.

4xVeron	te PDI Builder							- ×
Arbiter h	nw rev. 1.8 B 👻			🚓 Input	/ Outpu	ut		4 6 9
0	SCI I/O Setup	CAN		ANB	Units M	lask BIN	Raudrate 1000000.0	
	CAN CAN I/O	Maill	ooxes	reserved RX:	16 Mailbo	xes available	TX: 16	
P	CAN Setup	#		Mailboxes	Extended	ID	Mask	angua .
H.		1	1000000	4		1302	1111111111	
		2	HEREFOR	4		8	1111111111	
		3	11001000	4		9	1111111111	
		4	HEREIGHT	4		10	1111111111	
		_						
		_						
		<						>

Fig. 25: CAN Setup panel

More information about mailboxes can be found in the Mailboxes - Input/Output section of the **1x PDI Builder** user manual.



Ports configuration allows the user to configure which communication ports (Commgr Ports in *I/O Setup*) will be used for communication. When using the Route feature, **arbiters** can be configured to **route** VCP messages for an external Veronte device with a known address (ID) through a given port.

Note: To know more about VCP messages, read the VCP user manual.

4xVeron	te PDI Builder		- ×
Arbiter ł	nw rev. 1.8 B 👻		
0	Ports		+
•	Routing 1		Ô
	Forward PORT 1 PORT 2 PORT 3 PORT 4 PORT 5 Routing 2 Forward PORT 6	Route UAV 998 PORT 1 UAV 998 PORT 1 Route UAV 998 PORT 1 	

Fig. 26: Communications panel

Each port can be configured with the following options:

- Forward: Any message generated by this unit (i.e. telemetry or response messages to certain commands) will be sent through these ports.
- **Route**: Any message received at any Commgr Port with the defined address will be re-sent through the defined port. It is possible to route several addresses through the same port, but is **not possible** to route the same address through several ports. Only the first configured port will be used. Routing also applies to messages generated by the unit for the defined address.

Note: The same port cannot be used as Forward and Route at the same time.

It is possible to define up to 4 routing setups, which can be switched using the Ports action of **1x PDI Builder** software. Routing 1 will always be selected by default when booting an **arbiter**.

Warning: An incorrect port configuration can disable USB communications. If this happens, the **arbiter** will not be detected through Veronte Link.

If this is the case, please visit How to enter in maintenance mode - Troubleshooting section of the **4x Hardware Manual**.



The arbitration algorithm in **Autopilot 4x** is based on a **scoring system**. Each autopilot must send continuously a set of arbitration variables that will be used by the arbiter in order to calculate the score for each unit. Then, based on the scores and the current arbitration mode, the arbiter will choose to keep the current selected autopilot, or switching to one of the other units.

A **Score** is a 32 bit, single precission, floating-point value. This parameter is first computed resulting in a range between **0** and **0xFFFFFFFF**, where **0** is a perfect score (*Score*'). To achieve a better understanding, *Score*' is converted to a value comprehended between 0 an 100 (*Score*), being **100** the best possible score in the end.

Scores are calculated using the **arbitration variables** received from each autopilot at their dedicated addresses. After receiving the value, the following formula is used to compute the score for their respective unit:

$$score'_{i} = \frac{\sum_{j}^{N} min\left[\left(\frac{x_{i,j}-\mu_{j}}{d_{j,max}}\right)^{2}, 1\right] \cdot w_{j}}{\sum_{j}^{N} w_{j}}$$

$$score_i = 100 \cdot (1 - score'_i)$$

Where:

j	Variable index
$x_{i,j}$	<i>j</i> variable from autopilot <i>i</i>
μ_j	<i>j</i> variable reference
$d_{j,max}$	Maximum allowed error
w_j	Weight of <i>j</i> variable
N	Number of variables
$score'_i$	Score' of unit <i>i</i>
$score_i$	Score of unit <i>i</i>

Tip: The Arbitration weights should be used to increase or decrease the relevance that a certain arbitration variable has over the calculation of the score.

Any variable in Veronte can be used as an arbitration variable. Depending on the platform, operation, application etc. the more relevant variables can be selected for its use as arbitration references.

2.4.1 Config

2.4.1.1 Absolute Arbitration Variables

Absolute arbitration variables are indicators that are **inherently good** or **bad**, and so they are added directly to the score.

Examples of absolute arbitration variables are Link Quality, GNSS accuracy or warnings such as Sensors error or Position not fixed.

2.4.1.2 Relative Arbitration Variables

Relative arbitration variables are **not inherently good** or **bad**, and hence need to be compared against the other autopilots in order to calculate its score contribution.

The contribution to the score from a relative arbitration variable will be its **Deviation** from the **Average** of the same variable from each autopilot.

Examples of relative arbitration variables are Attitude, Position, measurements from sensors, etc.

2.4.1.3 Arbitration Example

Autopilot	Var.	Veronte	Туре	$x_{i,j}$	μ_j	w_j	Relative	$Score'_i$	$Score_i$
	N⁰	Variable				_	score		
1	1	Roll	Relative	0.12	0.096	1	0.054	0.043	95.67
	2	Pitch	Relative	0.30	0.283	1	0.027		
	3	GNSS 1	Absolute	1.7		0.01	0.004]	
		Accuracy							
2	1	Roll	Relative	0.10	0.096	1	0.0011	0.0039	99.60
	2	Pitch	Relative	0.28	0.283	1	0.0011		
	3	GNSS 1	Absolute	1.9		0.01	0.005		
		Accuracy							
3	1	Roll	Relative	0.07	0.096	1	0.071	0.046	95.39
	2	Pitch	Relative	0.27	0.283	1	0.017		
	3	GNSS 1	Absolute	1.5		0.01	0.0036	1	
		Accuracy							

In the above example, **AP2** is considered to be **the best**, since it has the **highest total score**. Even though its **GNSS accuracy** is the worst of all 3, its values for pitch and roll are the ones with the lower deviation from the mean.

2.4.1.4 Config menu

In Config menu the parameters of the arbiter algorithm are adjusted.

4xVeronte	PDI Builder	- ×
Arbiter hw	rev. 1.8 B 👻 Arbitration	* 4 5 0
0	Config CAN Setup	
	Arbiter	
	Preferred Autopilot 1 -	
Ŷ	Method Change if worst	
<u>a</u>	Holding CAP time 0.1 s	
	Hysteresis 0.001 ×1	
	Use external autopilot	
	Variables	
	Variable 1 Max error: 0.0 Weight: 1.0 Absolute: 0	
	Variable 2 Max error: 0.0 Weight: 1.0 Value: 0	
	Variable 3 Max error: 0.0 Weight: 1.0 Absolute: 0	
	6 7 8	

Fig. 27: Config panel

- 1. **Preferred**: The preferred autopilot will be chosen in case of a score draw. **Fixed while ok** mode will always select this autopilot first.
- 2. Method: The method of arbitration can be chosen by the user. The available options are:
 - Always best: It chooses always the best autopilot.
 - **Change if worst**: The arbiter will only switch if the currently selected autopilot has the worst score. In that case, it will switch to the one with the best score.
 - **Round robin control**: Using the **Holding CAP** time parameter, the **arbiter** will periodically switch between autopilots. This mode is meant for testing purposes only.
 - Fixed: Arbitration is disabled and one autopilot is selected. In this mode Autopilot 4x will behave as an Autopilot 1x.
 - Fixed 0: Autopilot 1 is selected.
 - Fixed 1: Autopilot 2 is selected.
 - Fixed 2: Autopilot 3 is selected.
 - **Fixed while ok**: This mode does not take into account scores. In this mode, the **Preferred** autopilot will be selected by default. A switch will only happen if the current autopilot is considered **Dead**.
- 3. Holding CAP time: Amount of time needed from last switch in order to allow a new switch.
- 4. **Hysteresis**: When comparing scores, the difference between them needs to be bigger than this proportional value, in order to assess scores. The difference is proportional to the score of the selected autopilot.

i.e: If current selected autopilot is the number 1, arbitration mode is **Always best**, hysteresis is **0.5** and score for AP 1 is **0.3**, AP 2 will need a score lower than **0.15** in order to be selected.

- 5. Enable arbitration of external autopilot.
- 6. Variable Max error: Arbitration maximum error for each variable.
- 7. Variable Weight: Arbitration weight for each variable.
- 8. Variable Absolute: If it is enabled, it will indicate that it is an absolute variable. The value set here will be used to compare the variables in order to choose the best autopilot.

Click on **t** to add variables and **t** to delete them. Currently, the **maximum** amount of **arbitration variables** supported is **32**.

The number for each variable is assigned with the **CAN I/O** configuration in the *arbiter consumer*. With this consumer, arbitration variables are selected with their corresponding numbers.

2.4.2 CAN Setup

Here users can configure the received CAN Ids for each one of the 3 possible **Autopilots 1x** that are sending data to **arbiters**. Therefore, to send data from any of them to **arbiters**, these Ids must be specified.

4xVeror	nte PDI Builder		- x
Arbiter	hw rev. 1.8 B 👻	Arbitration	* • •
Ö	Config CAN Setup		
•	Send status Period 0.25	 ①	
0	Send score Period 3.0		
	Status message ID 255 Extended	3	
m	Autopilot 1		
	L Extended		
	Autopilot 2		
	Extended		
	ID 9		
	Autopilot 3		
	ID 10		
		-	

Fig. 28: CAN Setup panel

- 1. Send status: The user can enable status sending and set a status message period.
- 2. Send score: Score sending can be enabled and a period of scoring messages can be set.

- 3. Status message ID: CAN ID to which Status and Score messages are sent.
 - 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.
- 4. Autopilot 1-3: CAN IDs used for the reception of autopilot 4x arbitration messages for each AP.
 - 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.

CHAPTER

THREE

INTEGRATION EXAMPLES

3.1 Arbitration configuration

The following arbitration configuration has been done **in conjunction** with the integration example detailed in the Autopilots 1x configuration - Integration examples section of the **1x PDI Builder** user manual. Also, this is the **recommended configuration** and a good starting point.

The following schema broadly summarizes the configuration that will be explained:



Fig. 1: Arbitration configuration diagram

Important: All the **Ids** represented in the schema are the **default** ones and those that will be used throughout this example, but users can change them as they wish.

Nonetheless, the Ids designated for arbitration in this configuration must **match** those entered in the **Autopilots 1x configuration** of the **1x PDI Builder** software.

AP 1 arbitration Id	AP 2 arbitration Id	AP 3 arbitration Id	Status message Id
8	9	10	255

Users should follow the steps below:

- 1. Go to Input/Output menu \rightarrow CAN I/O panel \rightarrow Configuration tab.
 - For sending the status and score messages to the Autopilots 1x, connect the Arbiter producer to an Output filter consumer, in this case *Output filter 2* has been selected.

This **Output filter** must be set to **CAN A** port, as this is the CAN Bus employed by the arbiter to communicate with **Autopilots 1x**. For more information on the CAN architecture of an **Autopilot 4x**, see the General description - Technical section of the **4x Hardware Manual**.

• For receiving the ready and arbitration variables messages from the Autopilots 1x, connect an Input filter producer to the Arbiter consumer, in this case *Input filter 2* has been selected.

To read these message correctly, it is necessary to configure this Input filter to CAN A port, with Id 8 and mask 2044 (decimal format), this will read messages with Ids from 8 to 10.

	Decimal format	Binary format
AP 1 arbitration Id	8	000 0000 1000
AP 2 arbitration Id	9	000 0000 1001
AP 3 arbitration Id	10	000 0000 1010
Mask	2044	111 1111 1100

Note: The following table shows the Ids and mask in binary and decimal format used in this example.

4xVeron Arbiter h	te PDI Builder w rev. 1.8 A 🛛 👻		•ক্রি Input / Output		20	- × 0
0	SCI	Configuration C	ustom message			
	I/O Setup	Priority	Producer	Consumer		
•4	CAN	V High 🕸	Serial to CAN 1	\rightarrow	Output filter 1	\$
0	CAN I/O	High 🗘	Serial to CAN 2	\rightarrow	None	Q ⁰
	CAN Setup	High 🗘	Serial to CAN 3	\rightarrow	None	Ø6
÷		High 🗘	Serial to CAN 4	\rightarrow	None	Q0
		High 🗘	Serial to CAN 5	\rightarrow	None	Q0
		High 🗘	Serial to CAN 6	\rightarrow	None	Q0
		High 🗘	Serial to CAN 7	\rightarrow	None	Q0
		High 🗘	Serial to CAN 8	\rightarrow	None	OS .
		High 🗘	Arbiter	\rightarrow	Output filter 2	¢¢
		✓ High 🛱	Input filter 1	\rightarrow	CAN to serial 1	08
		High 🗘	Input filter 2	\rightarrow	Arbiter	Q ₀
		✓ High 🕸	Input filter 3	\rightarrow	Output filter 3	Ф°
		✓ High 🛱	Input filter 4	\rightarrow	Output filter 4	¢°
		High 🔯	Input filter 5	\rightarrow	None	Q0
		High 🔅	Input filter 6	\rightarrow	None	Q ₀ ⁰
		High 🔯	Input filter 7	\rightarrow	None	Q0
		High 👸	Input filter 8	\rightarrow	None	- 36 V

Fig. 2: Arbitration configuration - CAN I/O configuration



Fig. 3: Arbitration configuration - Output filter configuration

Port	CAN A	
ld	8	
Mask	2044	de
Filter type	Standard	

Fig. 4: Arbitration configuration - Input filter configuration

2. Go to Input/Output menu \rightarrow CAN Setup panel \rightarrow CAN A tab.

SCI I/O Setup CAN CANB CAN Units ID CAN CAN I/O CAN Setup Mailboxes reserved RX: 26 Mail # Mailboxes Extended 1 4 2 4	sk BIN boxes available T2 ed ID 1302 8	Baudrate 1000000.0) X: 6 Mask 1111111111	+
CAN Setup # Mailboxes Extend 1 -4 - 2 -4	ed ID 1302 8	Mask	+
	1302 8	1111111111	
2 — 4	8		
		1111111111	
3 — 4	9	1111111111	
4 — 4	10	1111111111	
5 - 10	0	0	
<) >	

Configure at least 4 mailboxes in CAN A for each *arbitration Id*:

Fig. 5: Arbitration configuration - CAN Setup configuration

3. Go to Arbitration menu \rightarrow **Config panel**.

Here users have to configure the necessary parameters for the arbitration algorithm. For more information on this algorithm, see *Arbitration* section of the present manual.

/eronte PDI Builder		
iter hw rev. 1.8 A 🔹 👻	🚠 Arbitration	205
Config CAN Setup		
Arbiter		
Preferred	Autopilot 1 👻	
Method	Change if worst 👻	
Holding CAP time	2.0 s	
Hysteresis	0.1 x1	
Use external auto	oilot	
Variables		
- Variable 1 N	fax error: 0.35 Weight: 2.0 Absolute: 0	
	fax error: 0.35 Weight: 2.0 Absolute: 0	
	fax error: 1.0 Weight: 3.0 📝 Absolute: 1	
	fax error: 1.0 Weight: 3.0 📝 Absolute: 0	

Fig. 6: Arbitration configuration - Config (Arbitration) configuration

- Arbiter: These are the recommended values for each parameter:
 - Preferred: Autopilot 1
 - Method: Change if worst
 - Holding CAP time: 2.0 s
 - Hysteresis: 0.1
- Variables: Add as many variables as have been configured in the 1x PDI Builder software as arbitration variables.

The variables chosen in this example, as well as their configuration parameters, are presented below:

Variable	Max error	Weight	Absolute
Roll	0.35	2.0	-
Pitch	0.35	2.0	-
Position not fixed	1.0	3.0	Enabled 1
Custom Arbitration Variable	1.0	3.0	Enabled 0

Note: Remember that the number for each variable is assigned with the CAN I/O configuration in the *arbiter consumer*. With this consumer, arbitration variables are selected with their corresponding numbers.

4. Go to Arbitration menu \rightarrow CAN Setup panel.

Finally, users must enable the sending of **status** and **scores messages** from the arbiter and configure the CAN Ids described above.

4xVeror	nte PDI Builder		- ×
Arbiter	hw rev. 1.8 A 🔹	🕂 Arbitration	
Ö	Config CAN Setup		
•	Send status Period 0.15		
	Send score Period 0.25		
	Status message ID 255 E	ctended	
<u>ش</u>	Autopilot 1		
	Extended		
	ID 8		
	Autopilot 2		
	Extended		
	ID 9		
	Autopilot 3		
	Extended		
	ID 10		

Fig. 7: Arbitration configuration - CAN Setup (Arbitration) configuration

• Send status: Enabled

- Period: 0.15

• Send score: Enabled

- Period: 0.25

- Status message ID: 255
- Autopilot 1: ID 8
- Autopilot 2: ID 9
- Autopilot 3: ID 10

3.2 CAN tunnel

A CAN tunnel is the continuous transmission of messages between both CAN ports. Veronte Autopilot 4x can use any arbiter as a CAN tunnel, or both at the same time. The following diagram shows the physical connections of CAN buses inside the Autopilot 4x.



Fig. 8: Internal CAN connections diagram

Note: The following explanation is an example of how to configure arbiter A as a CAN tunnel. To configure arbiter B, the procedure is the same, but changing the arbiter selected.

In this example, a transparent tunnel will be created. So, any messages received on **Interface INT** (named as CAN A in **4x PDI Builder**) will be sent to **Interface EXT** (named as CAN B) and **vice versa**.



Fig. 9: CAN tunnel - Arbiter A diagram

Optionally, the mailboxes can be equally distributed to support both: standard and extended CAN IDs.

- 1. Create a new mailbox entry for **Interface INT**.
 - Go to Input/Output menu \rightarrow CAN Setup panel \rightarrow CAN A tab.
 - Assign half of the mailboxes to it and set a **Mask** of 0.

4xVeronte PDI Builder		– x
Arbiter hw rev. 1.8 A	• 🕞 Input / Output	2050
SCI I/O Setup CAN CAN I/O CAN Setup	CANA CANB Units ID DEC Units Mask BIN Baudrate 1000000.0 Mailboxes reserved RX: 20 Mailboxes available TX: 12 # Mailboxes Extended ID Mask 1 4 1302 1111111111 2 8 0 0	Enable Terminator
	3 8 0 0	

Fig. 10: CAN tunnel - CAN Setup configuration: CAN A

2. Create a new mailbox entry for **Interface EXT**. Repeat the same configuration as explained in step 1, but this time for **CAN B**.

4xVeron	te PDI Builder						- x
Arbiter H	nw rev. 1.8 A 🛛 👻		•	• Input /	/ Output		* * * 0
Arbiter I	sCI I/O Setup CAN CAN I/O CAN Setup	CAN Units Maill # 1 2 3	ANB DEC V U reserved RX: Mailboxes 4 8 8	► Input / Jnits Mask 20 Mailbox Extended	VOutput	Baudrate 1000000.0 TX: 12 Mask 11111111111 0 0	Enable Terminator
		<)	8

Fig. 11: CAN tunnel - CAN Setup configuration: CAN B

3. Go to Input/Output menu \rightarrow CAN I/O panel \rightarrow Configuration tab.

Configure an **Input filter** on **CAN A**, with CAN id 0, a **Mask** of 0 and **Both** as frame format type. In this case, *Input filter 3* has been chosen.

4xVeront	e PDI Builder					- ×
Arbiter h	w rev. 1.8 A 🔹		🚓 Input / Out	put	± 0	9
0	SCI	Configuration	Custom message			
	I/O Setup	Priority	Producer	Consumer	•	
•	CAN	V High 🖏	Serial to C	AN 1	Output filter 1	
	CAN I/O	High 🙀	Serial to C	AN 2	None	Ø0
	CAN Setup	High 🕵	Serial to C	AN 3	None	00
		High 🕵	Serial to C	AN 4	None	Q0
		High 🕵	Serial to C	AN 5	None	Q
		High 🙀	Serial to C	AN 6	None	Q _0
		High 🙀	Serial to C	AN 7	None	Q0
		High 🙀	Serial to C	AN 8	None	Q0
		High 🔯	Arbite	r	Output filter 2	00
		High 🛱	Input filte	er 1	CAN to serial 1	Ø6
		High 🕵			Arbiter	Ø6
		📄 High 🕵	Port CAN A	· · · · · · · · · · · · · · · · · · ·	None	Ø0
		V High	ld 0		Output filter 4	Ø.
		High 🕵	Mask 0	dec]	None	Q0
		High 🕵	Filter type Both		None	Q0
		High 🕵	Input filte	er 7	None	Q0
		THE MAR	Els.		NI	- 400 v

Fig. 12: CAN tunnel - Input filter configuration for CAN A

4. Bind an **Output filter** to the **Input filter** previously configured and, configure the Output filter to **CAN B**. In this case, *Output filter 3* has been chosen.

4xVeronte PDI Builder					- ×
Arbiter hw rev. 1.8 A		🚓 Input / Output		20	•
SCI	Configuration Cu	stom message			
I/O Setup	Priority	Producer	Consumer		
CAN	✓ High	Serial to CAN 1	\rightarrow	Output filter 1	\$
CAN I/O	High 👯	Serial to CAN 2	\rightarrow	None	00
CAN Setup	High 👯	Serial to CAN 3	\rightarrow	None	00
5	High 🕵	Serial to CAN 4	\rightarrow	None	08
	High 🕵	Serial to CAN 5	\rightarrow	None	08
	High 👯	Serial to CAN 6	\rightarrow	None	00
	High 🙀	Serial to CAN 7	\rightarrow	None	00
	High 🙀	Serial to CAN 8	\rightarrow	None	08
	High 🔯	Arbiter	\rightarrow	Output filter 2	00
	✓ High	Input filter 1	\rightarrow	CAN to serial 1	00
	High 🔯	Input filter 2	\rightarrow	Arbiter	05
	High 🗱	Input filter 3	\rightarrow	Output filter 3	\$
	V High	Input filter 4	\rightarrow	Output filter 4	¢¢
	High 🙀	Input filter 5	\rightarrow	None	Q ⁰
	High 🙀	Input filter 6	\rightarrow	None	00
	High 🙀	Input filter 7	\rightarrow	None	00
	Contract of the contract of th	Laura Elaura A		NI	atelli 🗸

Fig. 13: CAN tunnel - Output filter configuration for CAN B

5. Configure another **Input filter** on **CAN B**, with CAN id 0, a **Mask** of 0 and **Both** as frame format type. In this case, *Input filter 4* has been chosen.

4xVeror	nte PDI Builder							– ×
Arbiter	hw rev. 1.8 A 🔹		•	🕁 Inp	ut / Output		* •	• 0
Ö	SCI	Configurat	tion Custo	om messag	e			
	I/O Setup	Priority		Produ	cer	Consume	r	
•	CAN	🗸 High	00	9	Serial to CAN 1	\rightarrow	Output filter 1	\$
0	CAN I/O	High	00	9	Serial to CAN 2	\rightarrow	None	08
	CAN Setup	High	00	9	Serial to CAN 3	\rightarrow	None	00
÷.		High	00	9	Serial to CAN 4	\rightarrow	None	06
		High	00	9	Serial to CAN 5	\rightarrow	None	06
		High	00	9	Serial to CAN 6	\rightarrow	None	06
		High	00	9	Serial to CAN 7	\rightarrow	None	06
		High	00	9	Serial to CAN 8	\rightarrow	None	06
		High	06		Arbiter	\rightarrow	Output filter 2	00
		✓ High	00		Input filter 1	\rightarrow	CAN to serial 1	06
		High	00		Input filter 2	\rightarrow	Arbiter	06
		High	00		Input filter J		Output filter 3	00
		📃 High	CC Po	rt	CAN B		Output filter 4	00
		High	Co Id		0		None	06
		High	Ø₿ ^M	ask	0	dec	None	00
		High	¢¢ ^{Fil}	ter type	Both		None	00
		High	88 C		Innut filter 8		None	- 86 v

Fig. 14: CAN tunnel - Input filter configuration for CAN B

6. Bind an **Output filter** to the **Input filter** previously configured and, configure the Output filter to **CAN A**. In this case, *Output filter 4* has been chosen.

4xVeronte PDI Builder					— ×
Arbiter hw rev. 1.8 A 🛛 👻		🕂 Input / Output		1	9
O SCI	Configuration Cust	tom message			
I/O Setup	Priority	Producer	Consumer		
CAN	V High 🕸	Serial to CAN 1	\rightarrow	Output filter 1	Q ⁰
CAN I/O	High 🗱	Serial to CAN 2	\rightarrow	None	06
CAN Setup	High 🗱	Serial to CAN 3	\rightarrow	None	06
	High 🔯	Serial to CAN 4	\rightarrow	None	06
	High 🗱	Serial to CAN 5	\rightarrow	None	00
	High 🔯	Serial to CAN 6	\rightarrow	None	06
	High 🔯	Serial to CAN 7	\rightarrow	None	06
	High 🔯	Serial to CAN 8	\rightarrow	None	06
	High 🔯	Arbiter	\rightarrow	Output filter 2	Ø6
	V High	Input filter 1	\rightarrow	CAN to serial 1	06
	High 👯	Input filter 2	\rightarrow	Arbiter	06
	High 🕵	Input filter 3	\rightarrow	Output filter 3	Q ₀ ^o
	High	Input filter 4	\rightarrow	Output filter 4	08
	High 🛱	Input filter 5	\rightarrow	None	00
	High 🔅	Input filter 6	\rightarrow	None	00
	High 🔯	Input filter 7	\rightarrow	None	00
	High #	Input filter 8	\rightarrow	None	

Fig. 15: CAN tunnel - Output filter configuration for CAN A

3.3 GPIO command

The following are the steps to send a GPIO command from the **Veronte Autopilot 1x**, receive it at **Arbiter** and process it, so that **Arbiter** carries out the command. GPIO Command is very similar to PWM command with a few exceptions.

Warning: For the reception of CAN messages, Mailboxes need to be configured accordingly.

3.3.1 1x PDI Builder side

1. Go to Input/Output menu \rightarrow CAN Setup panel \rightarrow Configuration tab.

Connect a CAN GPIO remote producer to an Output filter consumer:

1xVeron	tePDI Builder - 1x v4.8 4041 ·	- CONNECTED						- x
1x 4.8	-		🕂 Inp	ut / Output			8 1 0	0
0	I/O Setup	Configuration	Custom message 0	Custom message 1	Custom message 2	Mailboxes		
	CAN Setup	Priority High 🙀	P Sen	roducer al to CAN 3		Consumer None	Mo	Always Uk
	Serial	High 🗘	Seri	al to CAN 4	\rightarrow	None	¢\$	Always Ok
۳		🗌 High 🗘	Seri	al to CAN 5	\rightarrow	None	Ø6	Always Ok
•		V High	CAN cu	stom message 0	\rightarrow	None	Ø0	Always Ok
		High	CAN cu	stom message 1	\rightarrow	None	\$\$	Always Ok
		🗌 High 🔯	CAN cu	stom message 2	\rightarrow	None	Q ⁰	Always Ok
¥8		V High 🚺	₽ ₽ In	put filter 0	\rightarrow	CAN to serial 0	Ф¢	Always Ok
P		V High 🚺	₿ ₿ In	put filter 1	\rightarrow	CAN to serial 1	Ф¢	Always Ok
600		🗌 High 🔯	₿ ₿ In	put filter 2	\rightarrow	CAN 4x	Q ⁰	Always Ok
		🗌 High 🚺	₿ ₽	put filter 3	\rightarrow	None	Q ⁰	Always Ok
		🗌 High 🚺	₿ ₽	put filter 4	\rightarrow	None	Q ⁰	Always Ok
◙		🗌 High 🚺	₿ ₽	put filter 5	\rightarrow	None	Ф¢	Always Ok
		🗌 High 🚺	CAN	unwrapper 0	\rightarrow	None	Q ⁰	Always Ok
676		High	CAN	unwrapper 1	\rightarrow	None	Q2	Always Ok
		🗌 High 🗘	CAN C	SPIO remote 0	\rightarrow	Output filter 3	Q ₀	Always Ok
		🗌 High 🗘	CAN C	SPIO remote 1	\rightarrow	None	Ø\$	Always Ok
U		High 🗘		CAN 4x	→	Output filter 1	Q ^o	Always Ok

Fig. 16: 1x PDI Builder - CAN Setup configuration

CAN GPIO remote must be configured. For more information on its configuration, see CAN Setup - Input/Output section of the **1x PDI Builder** user manual.

Period	0.1	S
ld of the genera CAN messages	ted 40	dec Extended
Destination	Value	Ô
Virtual 01	Virtual 00	-
Virtual 02	Virtual 00	~
Virtual 03		~

Fig. 17: 1x PDI Builder - CAN GPIO remote configuration

2. Go to Automations menu.

Create a new automation to activate the GPIO using an **Output action**:

1xVerontePDI Builder - 1x v4.8 4041 - CONNE	CTED	- ×
1x 4.8 👻	Q ₆ Automations	8 2 8 5 9
 39 - GPIO Command Calibration Home Navigation Integral Enable Failsafe Stick Transitions Phase Buttons 	GPIO Command Events AND OR NOT Actions + 64 - GPIO activated 25 - Activate GPIO	
© 0	Delay 0.0 s Periodical: Off	
	Output Activate GPIO Virtual 1 Off On Pulse Pulse off Time 0.0	
Image: New automation		U

Fig. 18: 1x PDI Builder - Automation configuration

3.3.2 4x PDI Builder side

3. Go to Input/Output menu \rightarrow CAN Setup panel.

Create the mailbox to receive the message configured in the **1x PDI Builder** (ID 40):

4xVerontePDI Builder - Arbiter v1.2 51091 -	CONNECT	ED				– ×
Arbiter hw rev. 1.8 A		•	• Input /	/ Output		
SCI I/O Setup CAN CAN I/O CAN Setup CAN Setup	CANA Units ID Mailboxe 7 3 4 5 6	CANB DEC BEC Mailboxes 4 4 4 4 4 10 1 1	Jnits Mask 27 Mailbox Extended	BIN	Baudrate 1000000.0 TX: 5 Mask 1111111111 1111111111 1111111111 111111	

Fig. 19: 4x PDI Builder - CAN Setup configuration

4. Go to Input/Output menu \rightarrow CAN I/O panel \rightarrow Configuration tab.

Finally, connect an **Input filter** producer with the **right CAN ID** to a **CAN GPIO consumer**.

4xVeron	tePDI Builder - Arbiter v1.2 510	091 - CONNECTED				- ×
	w rev. 1.8 A 👘 💌		🕂 Input / Output		B 1 0 !	9
Ö	SCI	Configuration	Custom message			
	I/O Setup	Priority	Producer	Consumer		
•~~	CAN CAN I/O	High 🔯	Serial to CAN 3	\rightarrow	CAN wrapper 2	
P	CAN Setup	High 🛱	Serial to CAN 4	\rightarrow	None	\$\$
.		High	Serial to CAN 5	\rightarrow	None	Q0
-		High	Serial to CAN 6	\rightarrow	None	00°
		High 🛱	Serial to CAN 7	\rightarrow	None	Q0
		High 🛱	Serial to CAN 8	\rightarrow	None	00
		High 🗘	Arbiter	\rightarrow	Output filter 2	Q ₀
		High 🛱	Input filter 1	\rightarrow	None	00
		High 🛱	Input filter 2	\rightarrow	Arbiter	00
		🗌 High 🔯	Input filter 3	\rightarrow	CAN GPIO consumer 1	00
		High 👯	Input filter 4	\rightarrow	None	00
		High 🛱	Input filter 5	\rightarrow	None	Q0
		High 🔯	Input filter 6	\rightarrow	None	08
		High 🔯	Input filter 7	\rightarrow	None	Q00
		High 🙀	Input filter 8	\rightarrow	None	00
		High 🗘	Input filter 9	\rightarrow	None	Q0

Fig. 20: 4x PDI Builder - CAN I/O configuration



Fig. 21: 4x PDI Builder - Input filter configuration

CHAPTER

FOUR

TROUBLESHOOTING

4.1 Maintenance mode

The user can simply enter maintenance mode with **4x PDI Builder** by clicking on the "Normal mode" button in the initial menu. Press the same button to return to "normal mode".

🔟 4xVeronte PDI Builder		-		×
4xVeronte				
Build PDI to configure your 4xVeronte				
▲ Upload PDI	👲 Open 4xVer	onte		
Upload PDI to the 4xVeronte, this option can't be undone	Open PDI online and w	ork with it		
Normal mode	Arbiter	v1.8 51090	(6.12	*

Fig. 1: Enter in maintenance mode



Fig. 2: Exit maintenance mode

4.2 Maintenance mode (loaded with errors)

The following error message may appear when trying to save a change or import a configuration.

ERROR	×
Error	×
Error when switching to normal mode,	check the configuration.

Fig. 3: Error message

Therefore, the **arbiter** will be in 'Maintenance mode (loaded with errors)':



To check what the source of the problem is, the user can simply click on the **PDI Error button**, which will show what the PDI Error is in the **PDI Errors panel**.

In this panel, users can found:

- PDI ID: ID of the PDI Error.
- **PDI Error Description**: Description of this PDI Error. A list of all PDI Errors can also be accessed in the List of PDI errors section of the **4x Software Manual**.
- Config ID: ID of the configurable (.xml file) containing the data in which the PDI Error has been caused.
- **Config Description**: Description of the configurable (.xml file) containing the data in which the PDI Error has been caused.

Clicking the **Export** button will export a .csv file with the same information shown in this PDI Errors panel.

This is useful while the configuration is in progress, however, if the user encounters this situation during the operation, it is also possible to look up the cause of the PDI Error directly on the **Platform panel** of **Veronte Ops**. For more information about this panel, see Platform panel section of the **Veronte Ops** user manual.



Fig. 4: Error message

Then, it will be possible to modify the Arbiter configuration to fix the error.

In addition, the list of PDI Errors can be read in the List of PDI errors section of the 4x Software Manual.

4.3 Migrate configuration

Warning: When performing automatic migration from a previous version to the current version of the software, errors may occur.

It is then the responsibility of the user to check the subsequent result.

CHAPTER

FIVE

FAQ

5.1 How to calculate a mask

This section attaches a python program that allows users to easily calculate their mask in standard or extended frame format by simply entering the CAN Ids as a **vector**. In addition, this program also converts each Id entered into binary.



An example of the execution of this program is shown below:



Fig. 1: Example of maskCalculator program