CEX PDI Builder

Release 6.12.47

Embention

2024-06-13

CONTENTS

1.1 System Requirements 3 1.2 Download 3 1.3 Installation 4 2 Configuration 5 2.1 CEX 14 2.1.1 CEX base 14 2.1.2 Status 15 2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3.1 GPIO 18 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 29 2.3.3.1 Tunnel 23 2.3.3.2 Custom Messages 27 2.3.3.4 ARINC communication protocol 29 2.3.3.4.1 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 24 2.3.6 Serial 53 2.4 Communications 53	1	Quic	k Start 3
1.2 Download. 3 1.3 Installation 4 1.1 Installation 4 2.1 CEX 14 2.1.1 CEX base 14 2.1.2 Status 15 2.2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 29 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.4.1 ARINC message transmission 34 2.3.4.2 ARINC message transmission 38 2.3.4 Configuration 40 2.3.4.1 ARINC message transmission 33 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input		1.1	System Requirements
1.3 Installation 4 2 Configuration 5 2.1 CEX 14 2.1.1 CEX base 14 2.1.2 Status 15 2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 16 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Turnel 23 2.3.3.1 Turnel 29 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC message reception 34 2.3.4.1 ARINC message reception 34 2.3.4.2 CAN BIO 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Setup 52 2.4 Communications 53 2.5.4 CAN Setup 55 2.5.1 PPM 55 2.5.2		1.2	Download
2 Configuration 5 2.1 CEX 14 2.1.1 CEX base 14 2.1.2 Status 15 2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.3 I/O Setup 23 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.4 ARINC communication protocol 33 2.3.4.1 ARINC message reception 34 2.3.4.2 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4.1 Ports 53 2.5.5 Lick 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox		1.3	Installation
2.11 CEX 14 2.1.2 Status 15 2.2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.3 I/O Setup 23 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.4.1 ARINC message reception 34 2.3.4.2 ARINC message transmission 38 2.3.4 CAN VO 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Setup 52 2.4.1 Ports 53 2.5.5 Stick 53 2.5.1 PPM 55 2.5.2 Exponential 57 <tr< td=""><td>2</td><td>Conf</td><td>jeuration 5</td></tr<>	2	Conf	jeuration 5
2.1.1 CEX base 14 2.1.2 Status 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 29 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 53 2.5 Stick 53 2.5 Stick 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57	-	2.1	CEX 14
2.1.2 Status 15 2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JET box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message transmission 38 2.3.4 Configuration 40 2.3.4.1 Configuration 40 2.3.4.2 CAN I/O 43 2.3.5 Digital Input 43 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 <td></td> <td>2.1</td> <td>2.1.1 CEX base 14</td>		2.1	2.1.1 CEX base 14
2.2 Sensors 15 2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 23 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.4.1 ARINC message transmission 38 2.3.4.2 CAN I/O 40 2.3.4.1 ARINC message transmission 38 2.3.4.2 CAN I/O 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.5 Stick 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58			2.1.2 Status 15
2.2.1 RPM 15 2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.4.1 Communication protocol 34 2.3.4.2 CAN Message reception 34 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6.1 petioses 61 <tr< td=""><td></td><td>2.2</td><td>Sensors</td></tr<>		2.2	Sensors
2.2.2 Lidar 16 2.3 Input/Output 18 2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message reception 34 2.3.4.1 Configuration 40 2.3.4.2 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Ports 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61		2.2	2.2.1 RPM 15
2.3 Input/Output 18 2.3.1 GPIO 19 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.2 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox <t< td=""><td></td><td></td><td>2.2.2.1 Reference 15</td></t<>			2.2.2.1 Reference 15
2.3.1 GPIO 18 2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.4.2 ARINC message reception 40 2.3.4.2 CAN I/O 40 2.3.4.2 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63		2.3	Input/Output 18
2.3.2 PWM 19 2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4 ARINC message reception 34 2.3.3.4.1 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.2 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.2 Scorpion tribunus 62 2.7 <t< td=""><td></td><td>2.00</td><td>2.3.1 GPIO 18</td></t<>		2.00	2.3.1 GPIO 18
2.3.3 I/O Setup 23 2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.1 ARINC message transmission 34 2.3.4.2 ARINC message transmission 34 2.3.4.1 Configuration 40 2.3.4.2 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 53 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Je			2.3.1 ONIO
2.3.3.1 Tunnel 25 2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5.2 Exponential 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			233 I/O Setun
2.3.3.2 Custom Messages 27 2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.1 ARINC message reception 34 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message transmission 38 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5.2 Exponential 55 2.5.3 Trim 55 2.5.4 Output 58 2.5.4 Output 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2331 Tunnel 25
2.3.3.3 JETI box 29 2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message reception 34 2.3.4.1 CAN I/O 40 2.3.4.2 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2332 Custom Messages 27
2.3.3.4 ARINC communication protocol 33 2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message reception 34 2.3.4.2 ARINC message transmission 38 2.3.4 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2 3 3 3 IETI hox 29
2.3.3.4.1 ARINC message reception 34 2.3.3.4.2 ARINC message transmission 38 2.3.4 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5.5 Stick 53 2.5.6 Exponential 57 2.5.7 PPM 55 2.5.8 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2 3 3 4 ARINC communication protocol 33
2.3.3.4.2 ARINC message transmission 38 2.3.4 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2.3.3.4.1 ARINC message recention 34
2.3.4 CAN I/O 40 2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2.3.3.4.2 ARINC message transmission 38
2.3.4.1 Configuration 40 2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			234 CAN I/O 40
2.3.4.2 CAN Telemetry 43 2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 53 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2 3 4 1 Configuration 40
2.3.5 Digital Input 44 2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2342 CAN Telemetry 43
2.3.6 Serial 51 2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			235 Digital Input
2.3.7 CAN Setup 52 2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			236 Serial 51
2.4 Communications 53 2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2 3 7 CAN Setun 52
2.4.1 Ports 53 2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63		24	Communications 53
2.5 Stick 55 2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63		2.1	2 4 1 Ports 53
2.5.1 PPM 55 2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63		2.5	Stick 55
2.5.2 Exponential 57 2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63		2.0	2.5.1 PPM 55
2.5.3 Trim 58 2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2.5.2 Exponential 57
2.5.4 Output 59 2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2.5.2 Experioritation 1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.1.
2.6 Devices 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63			2.5.6 11111
2.6 Jetibox 61 2.6.1 Jetibox 61 2.6.2 Scorpion tribunus 62 2.7 Arbitration 63 2.7.1 Arbitration 64		2.6	Devices 61
2.6.1 Scorpion tribunus 62 2.7 Arbitration 63 2.7.1 Arbitration 64		2.0	2.6.1 Jetihox 61
2.7 Arbitration 63 2.7 1			2.6.2 Scorpion tribunus 62
2.7.1 Arbitration		27	Arbitration 63
A DUI ADUI 100		2.1	2.7.1 Arbitration 64

		2.7.2 Config
		2.7.2.1 Absolute Arbitration Variables
		2.7.2.2 Relative Arbitration Variables
		2.7.2.3 Arbitration Example
		2.7.2.4 Config panel
		2.7.3 CAN Setup
3	Integ	cation examples 69
	3.1	CAN communication
		3.1.1 CAN messages transmission
		3.1.2 CAN messages reception
	3.2	CAN Isolator
		3.2.1 Filtered CAN subnet
		3.2.2 CAN tunnel
	3.3	Commanding/Reading PWMs
		3.3.1 1x PDI Builder side
		3.3.2 CEX PDI Builder side
	3.4	GPIO Command
		3.4.1 1x PDI Builder side
		3.4.2 CEX PDI Builder side
	3.5	Reading/Sending RPMs 95
		3.5.1 CEX PDI Builder side
		3.5.2 1x PDI Builder side
	3.6	Serial communication
		3.6.1 CEX PDI Builder side
		3.6.2 1x PDI Builder side
	3.7	External devices
		3.7.1 Analog devices
		3.7.2 Jetibox
		3.7.3 Scorpion tribunus
		3.7.4 Veronte products
		3.7.4.1 Connection with Autopilot 1x via CAN
		3.7.4.2 Connection with Autopilot 4x via CAN
		3.7.4.2.1 CAN Reception IDs
		3.7.4.2.2 CAN I/O Interconnections
4	Trou	leshooting 123
	4.1	ARINC Communication not working 123
	4.2	Maintenance mode (loaded with errors)
	4.3	Migrate configuration
5	FAQ	131
	5.1	How to calculate a mask

CEX Builder

CEX PDI Builder is an application for modifying, generating and uploading CEX PDI files.

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.								
🖉 Veronte Link 🗙 +		× – 🖬 ×						
\leftrightarrow \rightarrow C $($ manuals.embention.com/ver	onte-link/en/6.12.9/index.html	• •						
		l͡a Home Version-6.12.9 ▼ Languages-EN ▼ Download ▼ ⊕						
٩	Docs » Veronte Link	Quick Start →						
Quick Start	Mananta Link							
Integration examples	veronte Link							
Troubleshooting								
		ΠΤΕ						
	Veronte Link interconnects multiple control stations and autopilot units, s	so they can operate simultaneously.						

CHAPTER

QUICK START

CEX PDI Builder is the main configuration tool to adapt a **CEX** to be suitable for a specific system, being its main goal to expand the available communication protocols. **CEX PDI Builder** includes:

- Communications: Through general purpose CAN bus, inputs and outputs and PWMs.
- Stick control signal management: Compatible with **Stick Expander**, Futaba, Jeti, FrSky and TBS. It includes custom configuration for other sticks.
- Arbitration: CEX is able to send PWM signals using arbitration in the same way Veronte Autopilot 4x does.

Once CEX has been detected on Veronte Link, install CEX 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 the CEX 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 CEX PDI Builder on Windows just execute "CEXPDIBuilder.exe" and follow the Setup Wizard instructions.

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" and then, click on "Turn windows Defender Firewall on or off".								
📽 Windows Defender Firewall				-		×		
← → ∽ ↑ 📽 « System and	d Security > Windows Defender Firewall ~	C Search Control Panel				Q		
Control Panel Home	Help protect your PC with Windows Defender Fi	ewall						
Allow an app or feature through Windows Defender	Windows Defender Firewall can help to prevent hackers or mali through the Internet or a network.	cious software from gaining access to your PC						
Firewall Change notification settings	Private networks	Connected 📀						
Turn Windows Defender Firewall on or off	Guest or public networks	Not connected 📎						
 Restore defaults Advanced settings 								
Troubleshoot my network								
	Fig. 1: Windows I	Defender Firewall						
P Customise Settings				-		×		
← → 👻 ↑ 🔗 > Control Pa	nel \rightarrow System and Security \rightarrow Windows Defender Firewall \rightarrow Cust	omise Settings	~ Ū	Search Control Panel		Q		
	Customise settings for each type of netwo	rk						
	You can modify the firewall settings for each type of ne	twork that you use.						
	Private network settings Turn on Windows Defender Firewall							
	Block all incoming connections, includ	ing those in the list of allowed applications						
	Notify me when Windows Detender Firewall (not r	ewall blocks a new app						
	Public network settings							
	Turn on Windows Defender Firewall	for the state that of all sound and the form						
	☐ Block all incoming connections, include ✓ Notify me when Windows Defender Fire	ewall blocks a new app						
	Turn off Windows Defender Firewall (not i	recommended)						
		OK Cancel						
Fig. 2: Windows Defender Firewall: Settings								

CHAPTER

TWO

CONFIGURATION

Once the installation is finished, open **CEX PDI Builder** and select the unit.



Fig. 1: CEX ID

If it is correctly connected, **CEX PDI Builder** will display the **mode** in which the connected unit is. In addition, a **PDI** error button will appear:



Fig. 2: CEX PDI Builder

• **CEX mode**: CEX 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 in 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 appears:

INFORMATION	×
No errors	0
No PDI Errors found.	ОК

Fig. 3: CEX PDI Builder - PDI Errors message

The user can access now to 3 configuration options:



Fig. 4: CEX PDI Builder Options

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

Note: When an offline configuration is opened, the user can choose between working with CEX with GPIOs or ARINC.

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

CEX PDI Builder		- ×
CEX 2.0 with ARINC -	CEX PDI Builder	* & = 0
CEX 2.0 with ARINC		
CEX 2.0 with GPIOs		
۳		
•		
50		

Fig. 5: Hardware versions

• Upload PDI: A previously exported CEX PDI configuration can be imported to the linked CEX.

Warning: When a configuration is loaded into CEX 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, consult the Migrate configuration - Troubleshooting section of this manual.

• **Open CEX**: By clicking on this option, CEX PDI Builder configuration menu opens with the configuration (the PDI files) loaded in the CEX. Then, the user can modify it **online**.

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



Fig. 6: PDIs files

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

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

Confirmation X	CEX PDI Builder	_		×
Maintenance mode	Build PDI to configure your CEX			
Do you want to enter in maintenance mode? OK Cancel				
	Let the CEX, this option can't be undone Den CEX Open CEX Open CEX Open PDI online and work	with it		
	Maintenance mode	9 (6.12.60	Ŋ	*

Fig. 7: Open CEX

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

CEXPDI Builder - CEX2 45109 - CONNECTED		— ×
CEX 2.0 with ARINC -	CEX PDI Builder	0 3 6 4
0		12345
٣		
€		
9 -6		
8		
0		
		

Fig. 8: Initial menu

 Save and close: After changes are done, press on the save button to apply the changes in the PDI files. While saving, a percentage of saving process is displayed.

CEXPDI B	Builder - CEX2 45109 - CONNECTE	D	0101010		a carca carca car		anananananana	- ×
CEX 2.0 v	with ARINC 👻			• ← Input / Output			±. 0	
0	GPIO	Configur	ation					
	PWM	Priority		Producer		Co	onsumer	
<u>ا</u> ا					\rightarrow	Commgr port 0		Always Ok
• •	CAN I/O				\rightarrow	Commgr port 1		Always Ok
	Digital Input Serial				\rightarrow	Sensi to CAN 1		Always Ok
P	CAN Setup				\rightarrow	None		Always Ok
630					\rightarrow	JETI bax		Always Ok
					7	None		Always Ok
			Q_0^0	Call to Seriel 3	A	None		Always Ok
m			Q_0^0	Cull to Servel 4	Ð	None		Always Ok
			00	CAN to Serial 5	\mathbf{Y}	None		Always Ok
					\rightarrow	RS237-A		Always Ok
					\rightarrow	R\$ 2 32-В		Always Ok
					\rightarrow	None		Always Ok
					\rightarrow	None		Always Ok
				Commgr port 4	\rightarrow			Always Ok
				Commgr port 5	\rightarrow			Always Ok
					\rightarrow			Always Ok
					(Alwave Ok

Fig. 9: Save and close

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



Fig. 10: Confirmation panel - Save

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

Note: This button will only appear if a CEX unit 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 in the CEX.
- 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 CEX.
- 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 CEX 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.

FeedBack	— : nd us your feedback
Repo:*	Your repo
Title:	
🛓 Dow	vnload 🕥 Send

Fig. 11: Feedback

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





2.1.1 CEX base

CEX is able to send information about its version and Jeti telemetry of the devices connected to CEX, this is the "status message". Users can define the CAN Id used for this message:

Enable the **Extended** checkbox to use the extended CAN protocol (with a 29-bit identifier), disable it to use the standard protocol (11-bit identifier).

CEXPDI Builder - CEX2 45109 - CONNECTED)	- ×
CEX 2.0 with ARINC -	() CEX	
Cex base Status Image: Cex base Status Image: Cex base Image: Cex base <	Enable Jeti IX CAN ID Extended CAN 33	

Fig. 12: CEX base panel

2.1.2 Status

This option enables the periodic sending of the status message that Veronte Link uses to recognize CEX.

CEXPDI Builder - CEX2 45109 - CONNECTED)	- ×
CEX 2.0 with ARINC -	O CEX	8 2 8 9
Cex base Status Cex base Cex base Cex	Period 1.0 s	

Fig. 13: Status panel

• **Period:** Enter a desired period to send repeatedly the status message.

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



2.2.1 RPM

CEX can measure RPMs by reading from up to four inputs:

CEXPDI Builder - CEX2 45109 - CONNECTED		- x
CEX 2.0 with ARINC 👻	a Sensors	
RPM	RPM 0 RPM 1 RPM 2 RPM 3	
Lidar Lidar Lidar	Vinits 6.28318530717 Custom Average 5 Minimum 2.0E-4 5 Maximum 0.5 5	

Fig. 14: RPM panel

- Units: Sensor conversion factor. It can be Custom, Radians per pulse and Pulses per cycle.
- Average: It is a filter to avoid voltage spikes. The readout of the pulse can be filtered for the output to be an average. The amount of measurements to do the average needs to be specified.
- **Minimum**: Here the minimum expected pulse period needs to be specified. This will discard spurius pulses (e.g. induced by EMI) which are smaller than this minimum pulse.
- **Maximum**: The maximum period of time allowed without capturing. If no incoming pulse is received for more than this time, the output RPMs will be 0.

A configuration example to send RPM can be found in *Reading/Sending RPMs - Integration examples* section of the present manual.

2.2.2 Lidar

The I2C bus allows the connection of several devices with different addresses to the same line via master-slave communication. At this moment, CEX supports the following Lidar devices:

- Garmin LIDAR-Lite v3: Optical distance measurement sensor with a range from 5 cm to 40 m.
- SF11 Lidar: Long range laser altimeter. Supported SF11/B and SF11/C with a range of maximum 50 m and 0.2 m to 120 m respectively.
- SF20 Lidar: OEM laser altimeter module. Supported SF20/C with a range of 0.2 m to 100 m.

CEX allows up to 5 Lidar devices to be connected to the system at the same time. The configuration menu can be seen below:

CEXPDI Builder - CEX2 45109 - CONNECTE)	- × -
CEX 2.0 with ARINC -	Sensors	8 2 6 6 9
C RPM	Lidar I2C	
	✓ Lidar 0 Garmin Lida ▼ Address 0 dec Digital filter	
*	Lidar 1 SF11 Lidar dress 0 dec Digital filter	
	Lidar 2 SF20 Lidar dress 0 dec Digital filter	
	Lidar 3 Garmin Lida Address 0 dec Digital filter	
	Lidar 4 Garmin Lida Address 0 dec Digital filter	
		

Fig. 15: Lidar devices

After enabling the needed number of Lidar devices, configurable parameters are:

- Type of Lidar.
- Address: With an accepted value between 16 239, this is the origin address from the Lidar being configured.
- **Digital filter**: Enables a low pass filter whose cutoff frequency is manually configured, allowing the user to input any desired value in Hz. It is a **software filter**.

Note: I2C address will be different for different devices, make sure to define it properly by checking the manufacturer documentation.



2.3.1 GPIO

In this panel users can configure each individual GPIO behavior:

CEXPDI	Builder - CEX2 45109 - CONNE	CTED					- ×
CEX 2.0	with ARINC 👻		•🛟 In	put / Output			0 4 4
O	gpio	1 Signal	2 GPIOId	3 ю	4 Pull-up	5 Function	6 Qsel
	PWM	GPIO/ECAP 0	GPIO 24	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
	I/O Setup	GPIO/ECAP 1	GPIO 25	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN I/O	GPIO/ECAP 2	GPIO 26	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
•	Digital Input	GPIO/ECAP 3	GPIO 27	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
0	Serial	PWM 0	GPIO 0	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN Setup	PWM 1	GPIO 1	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
GD		PWM 2	GPIO 2	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 3	GPIO 3	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
•		PWM 4	GPIO 4	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
-		PWM 5	GPIO 5	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 6	GPIO 6	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 7	GPIO 7	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0.9	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Svnc ~

Fig. 16: GPIO panel

- 1. Signal: Pin ID as described in Hardware installation Pinout section of the CEX Hardware Manual.
- 2. GPIOId: GPIO ID of the microcontroller.
- 3. IO: Define GPIO as an input or ouput.
- 4. **Pull-up**: Enable or disable the pull-up resistance.
- 5. **Function**: Mux 0/GPIO: GPIO, Mux 1: PWM, Mux 2 or Mux 3. These are the different functionalities that the GPIO can have, depending on the multiplexer.

Note: When users set **Function** to "Mux 1", it indicates that the corresponding pin is disabled as GPIO and enabled as PWM. Consequently, the **Enable** checkbox in the *PWM menu* for that pin should be activated automatically.

Warning: Check that the corresponding Enable checkbox in the PWM menu is changed.

- 6. **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 whatever is present at 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 before, but checking 6 times instead of 3.
 - ASync: No checks are performed. It is used when it is not used as GPIO.

2.3.2 PWM

In this panel, each PWM can be configured:

CEXPDI	Builder - CEX2 45109 -	CONNECTED	- ×
CEX 2.0	with ARINC 🔫	• 🔂 Input / Output	
	GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	 PWM 0 PWM 1 Sub id 0 Frequency 50.0 Hz Frequency 50.0 Hz Frequency 50.0 Hz Mode Time Min 9.0E-4 Max 0.0021 6 PWM 1 PWM 2 PWM 3 PWM 4 PWM 5 PWM 6 PWM 7	

Fig. 17: PWM panel

Note: PWMs in CEX work in normalized mode, so when the input value is 0 the output value will be the minimum configured, and when the input value is 4095 (12 bits all with ones), the output will be the maximum configured. This

approach allows usage of the maximum resolution for the commanded value.

The configuration parameters are:

1. Enable: Define if the PWM is enabled or not.

Note: This checkbox disables the pin as GPIO and enables it as PWM. Hence, in the *GPIO menu* the "Function" parameter shall change to "Mux 1".

- 2. **Sub id**: Identifies the sub-id of the PWM and, according to this value, determines the command to be used, using one type of CAN message or another.
- 3. Frequency: PWM output frequency.
- 4. Timeout: If a PWM message is not received in less than this time, the PWM will output the start value.
- 5. Start value: Value used before any PWM message arrives and on timeout.
- 6. Pulse source ID: PWM input ID [0,3], defined in the *Digital Input panel*.
- 7. Active High: Polarity high or low (high if enabled).
- 8. Mode: The available options are Time and Duty cycle.
 - Time: The values indicated in Min and Max parameters are expressed in time units.

CEX 2.0 with ARINC	• Creb	
GPIO PWM	▼ PWM 0	
I/O Setup CAN I/O Digital Input Image: CAN Setup	Sub id 0 Frequency 50.0 Timeout 0.1 Start value 0.001	s
	Pulse source ID 0 PWM 1 PWM 2 PWM 3 PWM 4 PWM 5	min h h m ns m ms ms Time
	 PWM 6 PWM 7 	

Fig. 18: Time units

• **Duty cycle**: This option is a a different way of indicating the pulse width. Now the value indicated in **Min** and **Max** parameters is a percentage which corresponds to the relation between the pulse width over the total period of the sent signal.

So a 100% duty cycle will correspond to a signal with a constant value of 1, while a 0% duty cycle implies a constant signal with value 0. Between this two extremes, the pulse width can vary as in the examples shown in the following figure.



Tig. 19. Duty cycle

Note: Duty cycle percentages can be expressed in percent and per unit.

CEXPDI B	uilder - CEX2 45109 - CONNECTE	D				- x
CEX 2.0 w	ith ARINC 🔫	•🛟 Input / Ou	Itput		4	0
Ö	GPIO PWM	▼ PWM 0	[
≈ 👎 🚱 🚺	I/O Setup CAN I/O Digital Input Serial CAN Setup	Image: Weight of the second	VM 1 Active High Mode Duty cycle Min 0.001 Max 0.002	×1 ×1 %		
		 PWM 1 PWM 2 PWM 3 PWM 4 PWM 5 PWM 6 PWM 7 				

Fig. 20: Duty cycle units

9. Min: This parameter is the pulse width value that will make the servo/actuator go to its lowest position.

It will be the output when the PWM message specifies **0**.

10. Max: This parameter is the pulse width value that will make the servo/actuator go to its **highest position**. It will be the output when the PWM message specifies **4095**.

An example of reading PWM can be found in *Commanding/Reading PWMs - Integration examples* section of the present manual.

2.3.3 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 serial inputs and outputs: external sensors, custom messages, etc.

CEXPDI I	Builder - CEX2 45109 - CONN	ECTED		•🕁 Input / Output	:		± C	- × • • •
0	gpio pwm	Configura Priority	tion	Producer		C	onsumer	
	I/O Setup	🔆 🔛 High	00	RS232-A	\rightarrow	Commgr port 1	00	Always Ok
	CAN I/O	📄 High	00	RS232-B	\rightarrow	Commgr port 2	08	Always Ok
	Digital Input Serial	High	00	RS485-C	\rightarrow	Serial to CAN 1	00	Always Ok
P	CAN Setup	High	00	CAN to Serial 0	\rightarrow	Commgr port 0	¢\$	Always Ok
3		🔲 High	00	CAN to Serial 1	\rightarrow	RS485-C	¢\$	Always Ok
		High	Q0	CAN to Serial 2	\rightarrow	None	¢\$	Always Ok
		High	Q0	CAN to Serial 3	\rightarrow	None	¢\$	Always Ok
÷		High	Q ⁰	CAN to Serial 4	\rightarrow	None	Q ⁰	Always Ok
		High	Q0	CAN to Serial 5	\rightarrow	None	Q ⁰	Always Ok
		High	08	Commgr port 0		Serial to CAN 0	Q ⁰	Always Ok
		High	08	Commgr port 1		RS232-A	Q ⁰	Always Ok
		High	08	Commgr port 2	\rightarrow	RS232-B	Q ⁰	Always Ok
		High	Q ⁰ ₀	Commgr port 3	\rightarrow	None	Q ₀ ^o	Always Ok
		High	00	Commgr port 4	\rightarrow	None	Q0	Always Ok
		High	00	Commgr port 5	\rightarrow	None	Q0	Always Ok
		High	00	Tunnel 0	\rightarrow	None	Q0	Always Ok
		High	08	Tunnel 1		None	86	Alwaye Ok

Fig. 21: I/O Setup panel

- **Priority:** Connections between I/O ports can be marked with **high priority** with this checkbox. If enabled, they will **run at high frequency: 1000 Hz**.
- **Producer:** Functions for creating and sending messages.
- Consumer: Functions for receiving and parsing messages.
- **Bit**: This assigns each connection to a bit. Thus, the connection is activated/deactivated depending on the status of the selected bit. By default, the *Always Ok* bit is set to all connections so that they are always active.

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.
- 4. (*optional*) Configure the **Bit** parameter. By pressing on the bit button, the user can select the bit to assign the connection to.

EXPDI Builder - CEX2 45109 - CON	NECTED					
.0 with ARINC 👻		😋 Input / Outpu	t		± 0	0
GPIO	Configuration					
PWM	Priority	Producer		Co	onsumer	
I/O Setup	High 🕸	RS232-A	\rightarrow	JETI box	Q ₀	Always Ok
CAN I/O	High 🔯	RS232-B	\rightarrow	Commgr port 2	Q0	Always Ok
Serial	High 🔯	RS485-C	\rightarrow	Serial to CAN 1	Q ₀ ⁰	Always Ok
CAN Setup	High 🕸	CAN to Serial 0	\rightarrow	Commgr port 0	Q ₀ ^o	Always Ok
	High 🔯	CAN to Serial 1	\rightarrow	None	Q ₀ ⁰	Always Ok
	High 🔯	CAN to Serial 2	\rightarrow	None	Q ₀ ⁰	Always Ok
	High 🔯	CAN to Serial 3	\rightarrow	None	Q ₀ ⁰	Always Ok
	High 🔯	CAN to Serial 4	\rightarrow	None	Q ₀ ^o	Always Ok
	High 🔯	CAN to Serial 5	\rightarrow	None	Q ₀ ⁰	Always Ok
	High 🔯	Commgr port 0	\rightarrow	Serial to CAN 0	Q ₀ ⁰	Always Ok
	High 🔯	Commgr port 1	\rightarrow	RS232-A	Q ₀	Always Ok
	High 🔯	Commgr port 2	\rightarrow	None	Q ₀	Always Ok
	High 🔯	Commgr port 3	\rightarrow	None	Q ₀	Always Ok
	High 🔯	Commgr port 4	\rightarrow	None	Q0	Always Ok
	High 🔯	Commgr port 5	\rightarrow	None	Q ₀	Always Ok
	High 🔯	Tunnel 0	\rightarrow	None	Q ₀	Always Ok
	High (198)	Tunnel 1		Curtom mercade concumer	1 88	Alwaye Ok

Fig. 22: Bit - Select a variable

The following I/O ports are available:

Field	Description
RS232-A	Serial Port 232 A
RS232-B	Serial Port 232 B
RS485-C	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
Tunnel	Creates a bidirectional brigde between two devices, see <i>Tunnel</i>
Custom message	This allows user to send/receive a serial custom message, see Custom message
producer/consumer	
Y splitter	Used to split a signal into 2
producer/consumer	
CAN to Serial /	Serial to CAN sends serial streams over a CAN Bus / CAN to serial undoes the
Serial to CAN	transformation 'Serial to CAN'
CAN wrapper for	CAN wrapper for serial transmission sends CAN streams over a serial Bus / Serial CAN
serial transmission	unwrapper undoes this transformation. For more information on these ports, please refer to
/ Serial CAN	CAN wrapper/CAN unwrapper - Input/Output section of the 1x PDI Builder user manual
unwrapper	
Tribunus ESC	Reads telemetry data from the Tribunus ESCs by connecting it to one of the serial ports
Lift - MCU	Created for communication with a Lift MCU
JETI box	Simulates a Jetibox to read telemetry from legacy Jeti devices, see JETI box
JETI telemetry	Reads telemetry from Jeti devices
ARINC producer	This allows user to send/receive ARINC 429 messages through HI-3210, see ARINC
channel/consumer	communication protocol

More information about some elements can be found in the following sections.

2.3.3.1 Tunnel

It is possible to configure a Tunnel which is a bidirectional bridge between units that communicate to each other sharing information about an external device connected to the Serial or Digital port.

Let's consider the following image.

CEXPDI E	Builder - CEX2 45109 - CON	NECTED	ananan ar		la a a a a a a a a a a a a a a a a a a			- ×		
CEX 2.0 v	vith ARINC 👻			🚓 Input / Output	:		*			
O	GPIO	Configura	tion							
	PWM	Priority		Producer		C	onsumer	Verente ID	Arr 2	
2	I/O Setup	High	Q^0_0	RS232-A	\rightarrow	Tunnel 1	00	veronte iD	App 2	
	CAN I/O	High	08	RS232-B	\rightarrow	Commgr port 2	08	Parser	No protocol	-
	Digital Input Serial	High	00	RS485-C		Serial to CAN 1	Q ₀	Destination tunnel	Tunnel producer 1	-
P	CAN Setup	High	Q_0^0	CAN to Serial 0	\rightarrow	Commgr port 0	Qo	Time between messages	0.01	s
GD		High	Q0	CAN to Serial 1	\rightarrow	RS485-C	Q0	Bytes to send	22	byte
6		High	Q_0^0	CAN to Serial 2	\rightarrow	None	Q ₀	Always Ok)
		High High	Q^0_0	CAN to Serial 3	\rightarrow	None	Q ₀	Always Ok		
m		High High	Q^0_0	CAN to Serial 4	\rightarrow	None	Q ₀	Always Ok		
		High High	Q^0_0	CAN to Serial 5	\rightarrow	None	Q ₀	Always Ok		
		High	Q_0^0	Commgr port 0	\rightarrow	Serial to CAN 0	Q0	Always Ok		
		High	Q _0	Commgr port 1	\rightarrow	RS232-A	Q ₀	Always Ok		
		High	Q _0	Commgr port 2	\rightarrow	RS232-B	Q ₀	Always Ok		
		High	Q^0_0	Commgr port 3	\rightarrow	None	Q ₀ ⁰	Always Ok		
		High High	Q^0_0	Commgr port 4	\rightarrow	None	Q ₀	Always Ok		
		High High	Q_0^0	Commgr port 5	\rightarrow	None	Q ₀	Always Ok		
		High	Q^0_0	Tunnel 0	\rightarrow	None	Q ₀	Always Ok		
		High	88 I	Tunnal 1		None	88 (Alwave Ok v		

Fig. 23: **Tunnel configuration**

In the image above there is a device connected to the **RS232-A** (**Producer**) and there is a **Tunnel** (**Consumer**) which sends that information to Veronte applications (App 2). On the other hand, a Veronte application, for example Veronte Ops, will receive the signal sent by this CEX unit.

The option available when configuring Tunnel as **Consumer** are:

- Veronte ID: Enter the address that will receive the information. The following options are the most common:
 - App 2: Veronte applications address.
 - Broadcast: All units on the network. Select this option for a generic configuration.
 - Address of a specific Veronte unit, it can be a 1x, a 4x, a CEX, etc.

For more information on the available addresses, see List of addresses section of the CEX Software Manual.

- Parser: The user can choose protocol to parse message data. The options available are:
 - No protocol
 - RTCM3
 - CANserial
- **Destination tunnel**: Number of port is used to avoid mistakes and identify a Tunnel when using more than one, *Tunnel 0, 1 and 2* are available.
- Time between messages.
- Bytes to send: Sets the message size to send.

When configuring **Tunnel as Producer** (i.e. on the unit that receives the information), no configuration is required. It is only necessary to connect it to a Consumer, usually to a serial port.

2.3.3.2 Custom Messages

Warning:

- CEX has a serial limitation shared with all Custom Messages:
 - Maximum number of vectors (fieldset): 100
 - Maximum number of fields: 1000
- Each Custom message **consumer** has a limit of 32 fields.

It is possible to configure the messages sent/received through the serial port and its conversion to system variables by selecting the option **Custom message producer/consumer** and configuring the I/O port.

CEXPDI	Builder - CEX2 45109 - CONN	IECTED	la l							
	with ARINC 👻			🕂 Input / Output			£ 0			
0	GPIO	Configura	ation							
	PWM	Priority		Producer		Consumer				
2	I/O Setup	High	08	CAN to Serial 5	\rightarrow	None	¢8	Always Ok		
•	CAN I/O	High	80	Commar port 0		Serial to CAN 0	00	Always Ok		
	Digital Input			Common port 0				Always Ok		
P	CAN Setup	High	3460	Commgr port 1	_	K5232-A	346g	Always Ok		
	CAN Setup	High	08	Commgr port 2	\rightarrow	None	05	Always Ok		
8		High	Q_0^0	Commgr port 3	\rightarrow	None	Q_0^0	Always Ok		
$\ \ \bigcirc$		High	Q_0^0	Commgr port 4	\rightarrow	None	Q ₀	Always Ok		
		High	Q^0_0	Commgr port 5	\rightarrow	None	Q0	Always Ok		
		High	Q^0_0	Tunnel 0	\rightarrow	None	Q0	Always Ok		
		High	Q_0^0	Tunnel 1	\rightarrow	Custom message consumer 1	Q^{0}_{0}	Always Ok		
		High	Q_0^0	Tunnel 2	\rightarrow	None	Q ₀	Always Ok		
		High	Q_0^0	CAN wrapper 0 for serial tran	\rightarrow	None	Q ₀	Always Ok		
		High	Q^0_0	CAN wrapper 1 for serial tran	\rightarrow	None	Q0	Always Ok		
		High	Q^0_0	Custom message producer 0	\rightarrow	None	Q0	Always Ok		
		High	Q^0_0	Custom message producer 1	\rightarrow	RS485-C	Q	Always Ok		
		📃 High	Q_0^0	Lift - MCU	\rightarrow	None	Q ₀ ⁰	Always Ok		
		High	Q^0_0	Y0 splitter producer 0	\rightarrow	None	Q0	Always Ok		
			-		1 1		-			

Fig. 24: Serial Custom Messages

In the image above can be seen two possible configurations using a Custom Message. The 'red' one is configured to receive a determined message from Tunnel 1 and the 'green' is used to send a Custom Message through a RS-485 serial port. It is also possible to use the same Custom Message for both tasks if the bidirectional relationship is selected (the arrow indicates this, \leftrightarrow).

To configure a Custom message, the user must follow the next steps:

1. Press the **configuration button** (icon) and another window will be displayed. In this window press the "+" icon to add a custom message.

[Consumer] Cons	umer Custom mes	sage consumer	1									->
- 50	Little endian	Time out	Time to Idle	Bit ID: 0	Checksum 0	Matcher 0	Skip O	Variable O	ASCII 0	Position 0	Occupancy	÷
				Memory	age: 0 / 1	5384 byte	c .					
				memory us	age. 07 1	usua byte	3					

Fig. 25: Serial Custom Message configuration

2. When it is already added, the following options are available to configure a custom message:

[Producer] Producer Custom message prod	ucer 1						- ×
Little endian 1.0	od Delay s 0.0 s	Checksum Matcher 0 0	Skip Vari	riable ASCII O O	Position Oc	ccupancy	+

Fig. 26: Custom Message producer configuration

[Consumer] Consumer Custom mess	sage consume	1									- ×
Little endian	Time out 1.0 s	Time to Idle	Bit ID: 0	Checksum 0	Matcher 0	Skip O	Variable 0	ASCII 0	Position 0	Occupancy	+

Fig. 27: Custom Message consumer configuration

- Endianness: Depending on the order in which the device issue the message, it is possible to select:
 - **Big endian**: Set the value from left to right.

- Little endian: Set the value from right to left.
- Mixed endian: Some devices use this format. If users need to configure it, please contact the support team (create a ticket in the customer's Joint Collaboration Framework; for more information, see Tickets section of the JCF manual).
- Period/Time out: This option has a dual role depending on if it is used to transmit or receive data.
 - Period Producer: It is the inverse of the send frequency.
 - **Time out Consumer**: This is the threshold time between receptions to consider that the message is not being received correctly.
- Delay/Time to Idle: This option has a dual role depending on if it is used to transmit or receive data.
 - **Delay Producer**: It is a delay applied before sending the message. This serves to send messages with the same period without overloading the Serial bus.
 - Time to Idle Consumer: This is the time CEX waits before discarding partially parsed bytes.
- **Bit ID**: This option is only available when a message is configured as **Consumer**. The user bit selected in Bit ID box will be true if the message is being received correctly.

Warning: Pay attention that the user bit selected in **Bit ID** is not in use for another task.

To create the structure of the message, click on the edit message button (icon) and then press the "+" icon to add fields to it. The following type of messages are available to configure a structure: Variable, Checksum, Matcher, Skip and Parse ASCII. The configuration of each structure is covered in the Custom Messages types - Input/Output section of the 1x PDI Builder manual.

Warning: Before configuring any message, user has to know the structure it has to have according to the device that is connected to the port. Each device may have a different message structure when it sends or receives information.

2.3.3.3 JETI box

JETIBOX is a universal communication terminal which can be used with any JETI products.

JETIBOX operates as a two-way terminal, showing all data stored in the JETIBOX Compatible product. With the use of **four buttons**, the user can browse the menu and set the selected values to take advantage of the full capability of JETIBOX Compatible products.



Fig. 28: JETI box device

To simulate it, it is necessary to link the specific JETI box IO consumer to a serial port:

CEXPDI E	Builder - CEX2 45109 - CON	NECTED						- ×
	with ARINC 👻			• C Input / Output	t		± 0	• • •
0	GPIO	Configurat	ion					
	PWM	Priority		Producer		C	onsumer	
2	I/O Setup	High	08	RS232-A	\rightarrow	JETI box	Q ₀ ⁰	Always Ok
	CAN I/O	High	08	RS232-B	\rightarrow	None	Q ^o	Always Ok
	Digital Input	High	08	RS485-C	\rightarrow	None	Q0	Always Ok
P	CAN Setup	High	08	CAN to Serial 0	\rightarrow	None	Q0	Always Ok
3		High	08	CAN to Serial 1	\rightarrow	None	00	Always Ok
ā		High	00	CAN to Serial 2	\rightarrow	None	Q ₀	Always Ok
		High	00	CAN to Serial 3	\rightarrow	None	Q ₀	Always Ok
ф.		High	Q0	CAN to Serial 4	\rightarrow	None	Q ₀ ⁰	Always Ok
		High	Q ₀ ⁰	CAN to Serial 5	\rightarrow	None	Q ₀ ⁰	Always Ok
		High	Q0	Commgr port 0	\rightarrow	Serial to CAN 0	Q_0^0	Always Ok
		High	Q0	Commgr port 1	\rightarrow	RS232-A	Q ₀ ⁰	Always Ok
		High	Q0	Commgr port 2	\rightarrow	RS232-B	Q_0^0	Always Ok
		High	Q0	Commgr port 3	\rightarrow	None	Q_0^0	Always Ok
		High	Q0	Commgr port 4	\rightarrow	None	Q_0^0	Always Ok
		High	Q ^o	Commgr port 5	\rightarrow	None	Q ₀ ⁰	Always Ok
		High	00	Tunnel 0	\rightarrow	None	Q ₀ ⁰	Always Ok
		High	100	Tunnel 1	\rightarrow	None	88	Alwave Ok

Fig. 29: JETI box Consumer

ųΩ.	1
with .	icon):

Then, the sequence to retrieve the data shall be configured by clicking on the configuration button (



• To add a Custom Message, just click on the "+" icon:

Fig. 30: JETI box Configuration

The following parameters can be configured in this pop-up window:

- Endianness: Depending on the order in which the device issue the message, it is possible to select:
 - * Big endian: Set the value from left to right.
 - * Little endian: Set the value from right to left.
 - * **Mixed endian**: Some devices use this format. If users need to configure it, please contact the support team (create a ticket in the customer's **Joint Collaboration Framework**; for more information, see Tickets section of the JCF manual).
- Command: Here the user can select between Jeti box Left, Jeti box Down, Jeti box Up, Jeti box Right or Jeti box Nop. These correspond to the four buttons on the physical JETIBOX (see image above), except the Jeti box Nop that is only for simulating a "wait".

For example, to read the **Actual Voltage** of a **Jeti MasterSpin 220** the Consumer must be configured with a series of custom messages (use **Big endian** in all messages).

Comment Charlens Matches Ship Matches 1959 Buckles Operation
Command Checksum Matcher Skip Vanable ASCII Position Occupancy
Command Checksum Matcher Skip Variable ASCII Position Occupancy
Command Checksum Matcher Skip Variable ASCII Position Occupancy
Big endian Jeti Box Down → 0 1 0 0 0 0
Command Checksum Matcher Skip Variable ASCII Position Occupancy
- 📑 🗍 Big endian Jeti Box Down 🔹 0 1 0 0 0 0
Command Checksum Matcher Skip Variable ASCII Position Occupancy
- 📑 🗍 Big endian Jeti Box Down 🔹 0 1 0 0 0 0
Command Checksum Matcher Skip Variable ASCII Position Occupancy
- 📑 🗍 Big endian Jeti Box Down 🔹 0 1 0 0 0 0
Command Checksum Matcher Skip Variable ASCII Position Occupancy
- 📑 🔲 Big endian Jeti Box Down 👻 0 1 0 0 0 🦳
Command Checksum Matcher Skip Variable ASCII Position Occupancy
- 📑 🗍 Big endian Jeti Box Down 🔻 0 1 0 0 0 0
Command Checksum Matcher Skip Variable ASCII Position Occupancy
- 📑 🗍 Big endian Jeti Box Down 🔻 0 1 0 0 0 0
Command Checksum Matcher Skip Variable ASCII Position Occupancy
— 📑 🗍 Big endian Jeti Box Nop 🔹 0 1 8 0 1 0
Memory usage: 0 / 16384 bytes

Fig. 31: JETI box example

Below is a small example with the configuration of one of these messages, the full example can be found in *Jetibox* - *Integration examples* section of this manual.

- 1. Expected text: "CONTROLLER TYPE MasterSpin 220~"
 - Command: Jeti box Down
 - Matcher(32) "CONT" 0x434F4E54 (1129270868)
 - Skip(24*8) 192
 - Matcher(32) "220~" 0x3232307E (842150014)
| [Cons | sumer] Consur | ner JETI box | | | | | | | 0000000000 | | | | - × |
|-------|---------------|--------------|--------------|-------|----------|----------|---------|-----------|------------|----------|-----------|-----------|-----|
| _ 6 | | D: I' | Comp | and | Checksum | Matcher | Skip | Variable | ASCII | Position | Occupancy | | î+ |
| - 6 | | Big endian | Jeti Box Do | own 👻 | U | 2 | 8 | U | U | U | | | |
| | - 💷 🗇 | Hatch | er x434F4E54 | | | | | | | | | ^+ | |
| | Value | Bits | | Mask | | | | | | | | | |
| 1 | 129270868 | 32 | 4294967295 | | dec | | | | | | | | |
| Bits | - 💷 🗇 | Skip 2 | 4 | | | | | | | | | | |
| 24 | 4 | | | | | | | | | | | | |
| | - 🗊 🗇 | Skip 24 | 4 | | | | | | | | | | |
| - | - 🗊 🗇 | Skip 24 | 4 | | | | | | | | | | |
| | - 🗊 🗇 | Skip 24 | 4 | | | | | | | | | | |
| - | - 🗊 🗇 | Skip 24 | 4 | | | | | | | | | | |
| - | - 🗊 🗇 | Skip 24 | 4 | | | | | | | | | | |
| - | - 🗊 🗇 | Skip 24 | 4 | | | | | | | | | | |
| | - 🗔 🗇 | Skip 24 | 4 | | | | | | | | | | |
| | - 🗊 🗇 | Hatch | er x3232307E | | | | | | | | | | |
| | Value | Bits | | Mask | | | | | | | | | U |
| 8 | 342150014 | 32 | 4294967295 | | dec | | | | | | | * | ~ |
| | | | | _ | Me | mory usa | ge: 256 | / 16384 b | ytes | | | | |
| | | | | | | | | | | | | | |
| | | | | | | | | | | | | | |

Fig. 32: JETI box custom message example

2.3.3.4 ARINC communication protocol

CEX supports the reception and sending of information following the **ARINC 429 communication standard**. ARINC 429 messages consist of a 32-bit sequence as shown below.





- 1 8: The label/status byte.
- 9 10: Source Destination Identifier (SDI) bits. SDI function is used when specific messages need to be directed to a specific system of a multi-system installation.
- 11 31: Data bits.
- 32: Parity bit. Parity is a method of detecting errors in transmission.

The following configuration must be carried out in **CEX PDI Builder** for enabling reception and transmission of ARINC messages.

2.3.3.4.1 ARINC message reception

1. Go to Input/Output \rightarrow I/O Setup panel \rightarrow Configuration tab.

Connect an ARINC producer to a Custom message consumer.

	INECTED		🚓 Input / Output			± 0				
GPIO	Configurat	Configuration								
PWM	Priority		Producer		Cons	umer				
I/O Setup	High	00	Commgr port 5	\rightarrow	None	$ Q_0^0 $	Always Ok			
CAN I/O	High	08	Tunnel 0	\rightarrow	None	Q ₆ ^o	Always Ok			
Digital Input	High	Q [®]	Tunnel 1	\rightarrow	None		Always Ok			
CAN Setup	High	Q ⁰	Tunnel 2	\rightarrow	None		Always Ok			
Christiap	High	QC I	CAN wrapper 0 for serial tran	\rightarrow	None		Always Ok			
	High	00	CAN wrapper 1 for serial tran	\rightarrow	None	00	Always Ok			
	V High	Q ₀ ^o	Custom message producer 0	\rightarrow	ARINC consumer	Q ₀ ^o	Always Ok			
	High	Q ^o	Custom message producer 1	-	RS232-B		Always Ok			
	High	00	Lift - MCU	\rightarrow	None		Always Ok			
	High	00	Y0 splitter producer 0	\rightarrow	None	Q0	Always Ok			
	High	00	Y0 splitter producer 1	\rightarrow	None		Always Ok			
	High	00	Y1 splitter producer 0	\rightarrow	None	Q0	Always Ok			
	High	00	Y1 splitter producer 1	\rightarrow	None	Q0	Always Ok			
	High	Q ₀ ^o	ARINC producer channel 0	\rightarrow	Custom message consumer 0	Q ₀	Always Ok			
	High	Q 0	ARINC producer channel 1	\rightarrow	None	00	Always Ok			
	High	Q ^o	ARINC producer channel 2	\rightarrow	None	Q_0^0	Always Ok			
	High	Ō.º	ARINC producer channel 3	\rightarrow	None	08	Always Ok			

Fig. 34: I/O Setup - Producer selection

- 2. Configure the **ARINC producer** according to the message to read:
 - Low speed: Selects the transmission rate for the ARINC 429 transmit channel (low if checked, high if not).
 - **Parity check**: When this parameter is checked, the 32nd received ARINC bit is overwritten with **a parity flag**. The flag bit is set to a zero when the received ARINC word, including its parity bit, has an odd number of ones. Contrarily, when this **parity check** is not ticked, all 32-bits are received without parity checking.
 - **SDI filter**: Once this parameter is checked, SDI function is enabled. Users must configure the identifier of the ARINC messages to read (0, 1, 2 or 3) corresponding to the decimal value of SDI bits.

Note: The decimal value is extracted from the binary number formed by bits 10 and 9 in that order.

Bit N	umber	
10	9	Installation Number
0	0	-
0	1	1
1	0	2
1	1	3

Fig.	35:	SDI	bits	- ID	numbers
1 1g.	55.	SDI	DIUS	- 10	numbers

• **Enabled labels**: Users can set the labels of the ARINC messages to read, by selecting the corresponding decimal value of Label bits.

r	0010					1010101010101	-	0101010101010	101010101010		1010101010101							
GPIO DMAA		Configura	tion										6					
		Priority	446		Pi	roduce	er		r		INUI	IC	Co	nsum	er No l	Alw	ays UK	_
	CAN I/O	High	Q^0_0		Tunne	11		-			Nor	ne			X°	Alw	ays Ok	
	Digital Input	High	Q^0_0		Tunne	12		-	•	None			1	X0	Alw	ays Ok		
	Serial	High	Q_0^0	CAN wrap	per 0 fo	or seria	al tran		•		Nor	ne			28	Alw	ays Ok	
	CAN Setup	High	0.0	CAN wrap	per 1 fo	or seria	al tran		•		No	ne			28	Alw	ays Ok	5
		↓ Hiah	00	Custom n	nessage	e prod	ucer 0	-		AR	INC co	nsume	r		xe i	Alw	avs Ok	=
		High	Q.	Custom n	nessage	e prod	ucer 1	1_			RS23	2-B				Alw	avs Ok	5
		High	02		Lift - M	CU		-			No	ne				Alw	avs Ok	5
		High	08	V0 spl	itter nr	oduce	r 0	_			No	ne				Δlw	avs Ok	
		- High	240	VO col	itter pr	oduce	- 1				No			17		Alu	ays Ok	=
			240 240	V1 and	itter pr	oduce	- 0				Nor	ie			~9 49	Alw	ays Ok	
		- High	2495	Y I Spi	itter pr	oduce	ru				INOI	ne			-42 	Alw	ays Ok	
		High	Q6	Enabl	e		- 1	_								A1	01-	
		High	\mathbf{Q}_0^*	Low	/ speed													
		High	00	Pari	ty chec	k												
		High	\mathbf{Q}_{0}^{0}	SDI	filter													
		High High	\mathbf{Q}_0^0	SDI 0				-										
		High (\mathbf{Q}_0^0	Sol o														
					1	2	3	4	5	6	7	8	9	10	11	12	13	14
				16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
				32	33	34	35	36	37	38	39	40	41	42	43	44	45	46
				48	49	50	51	52 69	53 69	54	55	56 72	57	58	59	60	61	62
				80	81	82	83	84	85	86	87	88	89	90	91	92	93	94
				96	97	98	99	100	101	102	103	104	105	106	107	108	109	110
				112	113	114	115	116	117	118	119	120	121	122	123	124	125	126
				128	129	130	131	132	133	134	135	136	137	138	139	140	141	142
				144	145	146	147	148	149	150	151	152	153	154	155	156	157	158
				176	177	178	179	180	181	182	183	184	185	186	187	188	189	190
				192	193	194	195	196	197	198	199	200	201	202	203	204	205	206
				208	200	210	211	212	213	214	215	216	217	218	219	220	221	223

Fig. 36: I/O Setup - ARINC Producer configuration

3. Configure the Custom message consumer as follows:

[Consumer] Consumer Custom message consumer 0	- x
01 - I I I I I I I I I I I I I I I I I I	ne to Idle Checksum Matcher Skip Variable ASCII Position Occupancy D5 s Bit ID: 0 1 1 0 1 0 0
0 (0) - 8 Value Bits Matcher xAC Value Bits Mask 172 8 255 1 (0) - 32 Variable Of (Re Variable Compression User Variable. Uncompress 5 (0) - 8 Fype Bits Endianness Polynomial 8 Little e • (dec eal - 32 Bits) Decimals Encode/Decode Encode Decode Min Max Min Max 0 1.0 0.0 0.0 0 0 Crc - Preset BackFrom BackTo Polynomial StartValue Final XOR R. In R. Out crc8 • 5 0 7 0 0 Binar •
	Memory usage: 9 / 1000 Fields
	memory usage: 9 / 1000 FIElds

Fig. 37: I/O Setup - Consumer configuration

In order to ensure proper communication, it is mandatory to follow the structure above, adding a **Matcher** in the beggining, then a **Variable** (or variables corresponding to the 32 bits of the ARINC message) and a **CRC** at the end of the Custom message with the following values for the parameters:

- Matcher:
 - Value: 172
 - Bits: 8
- Variable or variables: This must correspond to the 32 bits of the ARINC message that users wish to receive. These can be configured in several ways, as long as all bits are represented and there is no loss of data. That is, the ARINC message does not necessarily have to be configured as a 32-bit uncompressed variable such as the *User Variable 00* shown in the example.
- Checksum (CRC):
 - Type: Polynomial
 - Endianness: Little endian
 - CRC-Preset: crc8
 - BackFrom: 5
 - BackTo: 0

- Binary mode

2.3.3.4.2 ARINC message transmission

1. Go to Input/Output \rightarrow I/O Setup panel \rightarrow Configuration tab.

Connect an **Custom message producer** to a **ARINC consumer**, and configure the consumer according the ARINC message to send.

- Low speed: Selects the transmission rate for the ARINC 429 transmit channel (low if checked, high if not).
- **Parity check**: When this parameter is checked, the 32nd transmitted ARINC bit is overwritten with a **parity flag**. Contrarily, when it is not, all 32-bits are transmitted as data.
- Even: When the **Parity check is ticked**, this parameter defines whether the 32nd transmitted bit is set for Even or Odd Parity. Ticking the **Even checkbox** selects **even parity**, if not **odd parity** will be selected.

CEXPDI E	3uilder - CEX2 45109 - CONN	IECTED						- ×
	with ARINC 👻			🕂 Input / Output		8	t C	
0	GPIO	Configura	ation					
	PWM	Priority	2015	Producer		Cons	umer	Always UK
2	I/O Setup	High	08	Tunnel 1	\rightarrow	None	00	Always Ok
•	CAN I/O		2000 2000	Tumeri		None		Always Ok
	Digital Input	High	\$26 	Tunnel 2		None	\$26 	Always Ok
P	CAN Setup	High	Q	CAN wrapper 0 for serial tran	\rightarrow	None	06	Always Ok
	canoctop	High	Q^0_0	CAN wrapper 1 for serial tran	\rightarrow	None	Q0	-
		V High	Q^0_0	Custom message producer 0] → [ARINC consumer	00	Low speed
$\mathbf{\circ}$		High	\mathbf{Q}_0^0	Custom message producer 1]→[RS232-B	Q ₀	Parity check
.		High	Q^0_0	Lift - MCU	\rightarrow	None	Q_0^0	Even party
		High	Q^0_0	Y0 splitter producer 0	 → [None	Q_0^0	Always Ok
		High	Q^0_0	Y0 splitter producer 1	\rightarrow	None	Q0	Always Ok
		High	Q_0^0	Y1 splitter producer 0	 → [None	Q_0^0	Always Ok
		High	Q_0^0	Y1 splitter producer 1	\rightarrow	None	Q_0^0	Always Ok
		High	Q^0_0	ARINC producer channel 0] → [Custom message consumer 0	\mathbf{Q}_{0}^{0}	Always Ok
		High	Q_0^0	ARINC producer channel 1	\rightarrow	None	00	Always Ok
		High	Q_0^0	ARINC producer channel 2	\rightarrow	None	Q0	Always Ok
		High	Q_0^0	ARINC producer channel 3	\rightarrow	None	Q0	Always Ok
		High	Q^0_0	ARINC producer channel 5	\rightarrow	None	Q_0^0	Always Ok

Fig. 38: I/O Setup - ARINC Consumer configuration

2. Configure the Custom message producer as below. In this example, *Relative Timestamp* variable is being sent.



Fig. 39: I/O Setup - Consumer configuration

In order to ensure proper communication, it is mandatory to follow the structure above, adding a **Matcher** in the beggining, then a **Variable** (or variables corresponding to the 32 bits of the ARINC message) and a **CRC** at the end of the Custom message with the following values for the parameters:

- Matcher:
 - Value: 172
 - Bits: 8
- Variable or variables: This must correspond to the 32 bits of the ARINC message that users wish to send. These can be configured in several ways, as long as all bits are represented and there is no loss of data. That is, the ARINC message does not necessarily have to be configured as a 32-bit uncompressed variable such as the *Relative Timestamp* shown in the example.
- Checksum (CRC):
 - Type: Polynomial
 - Endianness: Little endian
 - CRC-Preset: crc8
 - BackFrom: 5
 - BackTo: 0
 - Binary mode

2.3.4 CAN I/O

Warning: Remember that for the reception of CAN messages, Mailboxes need to be configured accordingly.

A CAN (Controller Area Network) Bus is a robust vehicle bus standard widely used in the aviation sector. CEX is fitted with two CAN buses that can be configured independently.

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



Fig. 40: CAN message structure

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

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

The baud rate of both CAN buses can be configured in the CAN Setup panel.

2.3.4.1 Configuration

This menu allows the configuration of the CAN inputs and outputs.

CEX PDI	Builder with ARINC 👻	100 AND 100		⊷ Input / Output			± 0	- × 0 4 4
0	GPIO	Configura	tion C	AN Telemetry 0 CAN Telemet	try 1 CAN	Telemetry 2		
2	I/O Setup	High	00	Application Processor		CAN Output Filter 1	¢\$	Always Ok
	CAN I/O	High	Q ^o	CAN Input Filter 0		CAN to Serial 0	Q2	Always Ok
•	Digital Input	High	Q ⁰	CAN Input Filter 1	→	Application Processor	00	Always Ok
P	CAN Setup	High	00	CAN Input Filter 2	→	None	20	Always Ok
3		High	00	CAN Input Filter 3	\rightarrow	None	02	Always Ok
		High	00	CAN Input Filter 4	\rightarrow	None	00	Always Ok
•		High	Q ^o	CAN Input Filter 5	\rightarrow	None	02	Always Ok
·		High	Q ^o	CAN Input Filter 6	\rightarrow	None	30	Always Ok
		High	0.0	CAN Input Filter 7	\rightarrow	None	30	Always Ok
		High	OS.	CAN Input Filter 8	\rightarrow	None	30	Always Ok
		High	OS.	CAN Input Filter 9	\rightarrow	None	30	Always Ok
		High	OC.	CAN Input Filter 10	\rightarrow	None	30	Always Ok
		High	OC O	CAN Input Filter 11	\rightarrow	None	20	Always Ok
		High	08	CAN unwrapper 0	\rightarrow	None	02	Always Ok
		High	OC I	CAN unwrapper 1	\rightarrow	None	20	Always Ok
		High	de l	Serial to CAN 0		CAN Output Filter 0	00	Always Ok
			00	Serial to CAN 1		None	1 25	Always Ok

Fig. 41: CAN I/O - Configuration panel

In this menu, the user can find the same 'columns' (**Priority**, **Producer**, **Consumer** and **Bit**) 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 in High state it is.

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

On the one hand, CEX has the **producers** shown below:

- **Application Processor**: Sends a specific set of information, the "status message". This message is composed as: version (major, minor and revision), address (serial number), system bit error, system power up bit error, PDI bit error, memory allocation bit, fily system bit error, CAN A bit error, CAN B bit error, arbiter enabled and arbiter status.
- CAN Input Filter: CAN input filters. 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 the icon:

C	M Input Filter 1	1
Port	CAN A	
ld	8	
Mask	2044	de
Filter typ	oe Standa	ard 👻

Fig. 42: CAN 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: Here a CAN Id mask can be set to filter messages. The mask marks which bits of the message id (in binary) are matched.

For example, if standard Ids (11 bits) from 8 to 11 (100 to 111 in binary) want to be admitted, the mask should be set to binary 1111111100, i.e. 2044 in decimal.

Warning: Make sure that the 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 create a mask - FAQ* section of this manual.

- Filter type: The options available are *Standard* (frame format with a 11-bit identifier), *Extended* (frame format with a 29-bit identifier) and *Both*.
- CAN unwrapper: This undoes the 'CAN serial wrapper' action, it has to be connected to *I/O Setup* consumer (*Serial CAN unwrapper*).
- Serial to CAN: Serial messages via CAN output, it has to be connected to I/O Setup consumer. It can be

configured in the **configuration button** (icon), a pop-up window will appear:



Fig. 43: 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 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.
- Time out: This is the threshold time between receptions to consider that it is not being received correctly.

• CAN Custom Producer Telemetry: Telemetry messages sent via CAN (such as CAN custom messages on Veronte Autopilot 1x). They are configured in the next section, see *CAN Telemetry*.

On the other hand, the **consumers** are the following:

- Application Processor: Receives a specific set of information sent by Veronte Autopilot 1x or Arbiter.
- CAN Output Filter: CAN output filters. The user can choose between CAN A, CAN B or BOTH in the

configuration button (icon).



Fig. 44: CAN Output Filter configuration

- CAN serial wrapper: CAN messages via serial output, it has to be connected to *I/O Setup* producer (CAN wrapper for serial transmission).
- CAN GPIO consumer: CAN messages from Autopilot 1x or Autopilot 4x for GPIO inputs. An example of how to implement it can be found in the *GPIO Command Integration examples* section of the present manual.
- CAN to Serial: This undoes the 'Serial to CAN' action, it has to be connected to I/O Setup producer.
- CAN Custom Consumer Telemetry: Telemetry messages received via CAN (such as CAN custom messages on Veronte Autopilot 1x). They are configured in the next section, see *CAN Telemetry*.

2.3.4.2 CAN Telemetry

In the CAN Telemetry tabs (there are 3 available), the user chooses the telemetry to be sent/received via the CAN buses. The following elements can be configured:

- **TX Ini**: Used to configure transmitted messages that are only sent once at the beginning of the operation (sent when CEX boots up). They can be used to initialize some devices.
- TX: Used to configure transmitted messages.
- RX: Used to configure 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.
- CEX has a CAN limitation of 40 TX Ini messages per Custom, 40 TX messages per Custom and 80 RX messages per Custom.

In addition, there is a limit shared with all Customs:

- Maximum number of vectors (fieldset): 160
- Maximum number of fields: 1280

CEXPDI Builder - CEX2 45109 - CONNECTE	D	- x
CEX 2.0 with ARINC -	•← Input / Output	8 2 8 9
GPIO PWM I/O Setup CAN I/O Oigital Input Image: CAN Setup CAN Setup Image: CAN Setup	Configuration CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2 TX Ini TX RX RX Memory usage: 4 / 1280 Fields	

Fig. 45: CAN I/O - CAN Telemetry panel

As this section works in exactly the same way as 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

An example of the sending of CAN telemetry messages can be found in the *CAN communication - Integration examples* section of this manual.

2.3.5 Digital Input

CEX digital signal inputs can be used to measure pulse count, pulse widths and PPM signals from a RC radio.

CEXPDI E	Builder - CEX2 45109 - CONNECT	ED					- x
CEX 2.0 v	with ARINC 👻		• 🛟 Input / Ou	utput		± 0	• • •
0	GPIO	Configurati	ion				
	PWM		Producer		Co	nsumer	
2	I/O Setup	\$\$°	CAP 0	\rightarrow	Pulse 0	00	Always Ok
	CAN I/O	Q0	CAP 1	\rightarrow	RPM 3	06	Always Ok
	Digital Input	OC	CAP 2	→	RPM 0	02	Always Ok
P	Serial CAN Setur	250 C	CAD 2		Dulco 2	100	Always Ok
	CAN Setup		CAP 5		Fuise 5		Always Ok

Fig. 46: Digital Input panel

In this menu, the user can also find the same 'columns' (**Producer**, **Consumer** and **Bit**) 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.

The process to configure a device can be done as follows:

1. Select and configure a **Producer**. There are 4 possible producers: CAP 0 - 3.

Press on the configuration button (icon) and a new pop-up window will show.

CEXPDI	Builder - CEX2 45109 - CONN	NECTED			- x
	with ARINC 👻	• 🔂 Input / Output		1	• • •
Ö	GPIO	Configuration			
	PWM	Producer	Со	nsumer	
2	I/O Setup		PPM 0	Q6	Always Ok
·~	CAN I/O	GPIO/ECAP 0	None	Q ⁰	Always Ok
	Digital Input Serial	Edge detection First falling edge	None	¢¢	Always Ok
P	CAN Setup		None	Q0	Always Ok
∞ ⊙					
m					

Fig. 47: Digital Input panel - Producer

The pop-up window contains the following configurable elements:

• Enable: By ticking this checkbox, the corresponding producer is enabled.

Note: When enabling it, in the *GPIO menu* the "Function" parameter corresponding to this pin shall automatically change to "Mux 1".

Warning: Please check that this change has been made.

• CAP pin entry: Selects which pin this CAP is associated to and, therefore, to which the device is connected to.

Note: Pins are associated in the following way:

- CAP 0 with GPIO/ECAP 0
- CAP 1 with GPIO/ECAP 1
- CAP 2 with GPIO/ECAP 2
- CAP 3 with GPIO/ECAP 3

• Edge detection: How the pulses are read and transformed into a digital signal (how they are processed). By clicking on the drop-down menu, the following options can be selected:

ingereten					
R	V Enabled	Producer			
~	GPIO/ECAP 0		•		
51	Edge detection	First falling edge	-		
5	¥	First rising edge			
2		First falling edge	<u></u>		

Fig. 48: Digital Input panel - Edge detection option

- First rising edge: With this option, when the rise of the pulse is detected, the data will start to be stored. Recommended when consumer is **PPM** or **Pulse**.
- First falling edge: With this option, when the fall of the pulse is detected, the data will start to be stored.

Note: By clicking on the marked arrows, pulse processing can also be configured, getting a customized arrow scheme.

CEXPDI	Builder - CEX2 45109 - CON	NECTED			- ×
CEX 2.0	with ARINC 👻	• 🗘 Input / Output		± 0	• • •
\bigcirc	GPIO	Configuration			
	PWM	Producer	Cons	umer	
<u>س</u>	I/O Setup	Enabled	Pulse 0	Q ^o	Always Ok
	CAN I/O	GPIO/ECAP 0	None	00	Always Ok
	Digital Input	Edge detection First falling edge	None	00	Always Ok
P	CAN Setup		None	00	Always Ok

Fig. 49: Digital Input panel - Edge detection arrows

2. Click on the **Bind** button to select the type of **Consumer**, it is possible to choose among a PPM 0 (Stick PPM), RPM 0-3 or Pulse 0-3.

CEXPDI	Builder - CEX2 45109 - CONNECTE	D				- ×
CEX 2.0	with ARINC 🔫		• 🕂 Input / Output		8 4 6 5	0
Ο	GPIO	Configuration	1			
	PWM		Producer		Consumer	
2	I/O Setup	\$\$°	CAP 0	→ РРМ О	Bind RPM 0	Ĵî
	CAN I/O	Q0	CAP 1	None	Clear RPM 1 Dk	
	Digital Input	30	CAP 2	None	RPM 2 Dk	
Ø	Serial	979 949	040 D	None	RPM 3	
	CAN Setup	346	CAP 3	None	Pulse 0	
6 29					Pulse 1	
\square					Pulse 2	
					Pulse 3	
-fit-						
						U ~

Fig. 50: Digital Input panel - Consumer

- **PPM 0** selected: The variable where the information is stored is 'PPM channel 0 output'. Stick PPM is configured in the *Stick* section.
- **RPM 0-3** selected: The variables in which the information read here is stored are '**RPM 0-3**'. For more information on the configuration of RPM, see the *RPM* section.
- Pulse 0-3 selected: The variables in which the information read here is stored are 'Captured pulse 0-3'.

It is possible to configure it cliking on the **configuration button** (icon):

CEXPDI	Builder - CEX2 45109 - CONNECTE	- ×
CEX 2.0	with ARINC 👻	+ Input / Output
\bigcirc	GPIO	Configuration
	PWM	Producer Consumer
2	I/O Setup	• CAP 0 • Pulse 0 • Pulse 0 • Pulse 0 • Positive pulse duration • Intervent
	CAN I/O	
	Digital Input	
0	Serial	
	CAN Setup	∞ CAP 3 None ∞ ag x 1.0 y 1.0 —
CD		
		0.8
P		0.7
—		
–		
		0.5
		03
		02
		0.1

Fig. 51: Digital Input panel - Pulse

In the pop-up window, users will find the following options for configuration:

- Mode:
 - * Positive pulse duration: The period of the pulse is obtained. It takes the time in 'High' state.
 - * Negative pulse duration: The period of the pulse is obtained. It takes the time in 'Low' state.



Fig. 52: Positive/Negative pulse duration

- * Positive duty cycle: The duty cycle of the pulse is obtained. It takes the time in 'High' state.
- * Negative duty cycle: The duty cycle of the pulse is obtained. It takes the time in 'Low' state.



Fig. 53: Positive/Negative duty cycle

- **Time out**: This defines the time to consider that no signal is received.
- Function: Here the user can customise a function to handle the values. Normally, a function is set with the points [0,0] and [1,1], so no transformation is applied, input = output. However, the user can configure it as desired.

Example

Let's imagine that **First rising edge** has been selected as the wrap option in Producer and the pulse that CEX has to read is a square signal with a period of 2 seconds and a duty cycle of 25% (see image below).



Fig. 54: Signal generated

On the other hand, if **Positive pulse duration** is selected as Consumer and it is configured as in the previous image (*Digital Input - Pulse*), the value obtained in the variable **Captured pulse** (*Captured pulse 1* in the following example) will be **0.50s**, this is because it is the period of the "Positive pulse" of that pulse.

However, if **Positive duty cycle** is selected as Consumer, the value obtained in the variable **Captured pulse** (*Captured pulse 2* in the following example) will be **0.25**, this is because it is the positive duty cycle of that pulse.





2.3.6 Serial

CEX can use up to three serial peripherals (SCI A, SCI B and SCI C). Serial ports A, B and C parameters can be edited in this menu to fit the serial protocol requirements.

CEXPDI E	CEXPDI Builder - CEX2 45109 - CONNECTED - ×						
	with ARINC 👻	•🔩 Input / Output					
	with ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	SCIA SCIB SCIC Functionality Baudrate 115200 • Length 8 • Stop 1 • Parity Disabled • Use address mode					

Fig. 56: Serial panel

- Baudrate: This specifies how fast data is sent over a serial line.
- Length: This defines the number of data bits in each character: 4 to 8 bits.

- Stop: Number of stop bits sent at the end of every character: 1, 1.5, 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.

Note: SCI A corresponds to port RS232-A, SCI B to port RS232-B and SCI C to port RS485-C.

2.3.7 CAN Setup

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

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

EXPDI Builder - CEX2 45109 - CONNECTE	D						->
CEX 2.0 with ARINC -			•ح	Input /	Output		8 1 8 9
GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	CAN Units Maill # 1 2	IA Ci s ID boxes	ANB DEC V reserved RX: Mailboxes 4 4	Jnits Mask 9 Mailboxo Extended	BIN - es available T ID 1302 400	Baudrate 1000000.0 X: 23 Mask 1111111111 111111111	

Fig. 57: CAN Setup panel

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



2.4.1 Ports

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, CEX can be configured to **route** VCP messages for an external Veronte device with a known address (ID) through a given port.

CEXPDI	Builder	- CEX2 45109 - CONNECT	ED	- ×)
CEX 2.0		INC 👻	Communications	e to to o
Ö	Por	ts		+
2	Ro	uting 0		$\widehat{\square}$
	-	Forward PORT 1 PORT 2 PORT 3 PORT 4 PORT 4 PORT 5	Route	

Fig. 58: Ports menu

Each port can be configured as either one of the following options:

- Forward: Any messages generated by this unit (i.e. Telemetry or response messages to certain commands) will be sent through these ports.
- **Route**: Any messages 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 unsing the Ports action of the Automations menu of **1x PDI Builder**. Routing 0 will always be selected by default when booting CEX.

2.5 Stick

The wired transmitter is configured through the following tabs, which cover the following content:

- Setting of the transmitter's parameters.
- Definition of exponential response-curves for the desired channels.
- Trimming of the channels' neutral position.
- Setting of the data receiving port on the autopilot.

2.5.1 PPM

This tab provides the options to configure CEX to read a PPM radio controller to control the platform.

DI Builder - CEX2 45109	- CONNECTED				
.0 with ARINC 👻		æ	9 Stick	8 ± 0	-
PPM Exponentia	l Trim Output				
Brand Custom	▼ Model	✓ Cha	innels 16		
Pulse polarity	Positive Negative	Sync time	0.004 s		
Min pulse	2.5E-4 s	Max pulse	5.0E-4 s		
Position					
Min accepted	8.0E-4 s	Max accepted	0.0022 s		
Min value encoded	9.0E-4 s	Max value encoded	0.0021 s		
Channel(Disabled	EnabledFilter) 1 2	3 4 5 6 7 8 9 10	11 12 13 14 15 16		
Non linear low pa	ss filter				
Min delta	0.0	Max delta	1000.0		
Min delta alpha	1.0	Max delta alpha	0.02		

Fig. 59: Stick - PPM panel

• **Brand**, **Model** and **Channels**: **CEX PDI Builder** has been configured to provide the user with the expected parameters to configure different transmitters models.

Brand	Models	Channels
Futaba	8J/10J/12K/14SG	8 (for 8J and 10J)
		12 (for 12K and 14SG)
	T18SZ	8
Jeti	DC 16/DC 24	16
FrSky	Taranis X9D	8
	Horus X12S	8
TBS	Crossfire	8
Embention	Stick Expander	16
Custom	-	-

- Custom: If the user's transmitter is not among those mentioned above, choose this option and replace the parameter values with the appropriate ones.
- Pulse polarity: Indicates the pulse polarity:
 - **Positive**: Default signal is low and goes up to high.
 - Negative: Default signal is high and goes down to low.
- Sync time: Minimum time on the PPM output till the next frame. It tells the receiver to reset its channel counter.
- **Minimum/Maximum pulse**: Pulse length, it depends on the system and it is a constant value (usually 0.2-0.5 ms).
- Position
 - Minimum/Maximum accepted: Pulse length accepted for each channel. Standard for R/C servos uses a pulse of 1 ms for the maximum position at one end, 1.5 ms for the midpoint and 2 ms for the maximum position at the opposite end.
 - **Minimum/Maximum encoded**: If there is noise and the signal is varying around the minimum/maximum values accepted, CEX will encode those values to the ones set here. For instance, a pulse length between 0.8-0.9 ms will be considered as one of 0.9 ms.
 - Channels: Sets the number of channels accepted. Besides, it is possible to Disable/Enable/Filter each channel individually.
- Non linear low pass filter
 - Minimum/Maximum delta: Default parameters are recommended.
 - Minimum/Maximum delta alpha: Default parameters are recommended.

The figure below shows the shape of the PPM signal.



Fig. 60: PPM signal

2.5.2 Exponential

The exponential tab allows the user to define an exponential stick response for every channel.

The allowed inputs range from 0 to 1 and there is a graph showing the generated response curve, as can be seen in the figure below.



Fig. 61: Stick - Exponential panel

The **X** axis of the graph corresponds to the stick input and the **Y** axis is the result of applying the exponential function to that stick input.

2.5.3 Trim

By enabling the **Avanced** option, the user can set the expected trim values manually. The user should have a deep knowledge on its transmitter if this option is selected.

Finally, on the right hand side, the **Reset** button puts every parameter back to 0.

CEXPDI	Builder - CEX2 45109 - CONNECTED		- ×
CEX 2.0	with ARINC -	CB Stick	8 2 6 0
Ö	PPM Exponential Trim Output	ıt	
2	Advanced	Reset	
	Trim Range	Neutral Trim Range Neutral	
•	Channel 0 0.(0.(Zero Channel 8 0.0 0.7 Zero	
P	Channel 1 0.0 0.0	Zero Channel 9 0.0 0.7 Zero	
620	Channel 2 0.0 0.0	Zero Channel 10 0.0 0.7 Zero	
	Channel 3 0.(0.1	Zero Channel 11 0.C 0.7 Zero	
⁰	Channel 4 0.0 0.0	Zero Channel 12 0.0 0.7 Zero	
.	Channel 5 0.(0.;	Zero Channel 13 0.0 0.7 Zero	
	Channel 6 0.1 0.7	Zero Channel 14 0.0 0.7 Zero	
	Channel 7 0.1 0.1	Zero Channel 15 0.0 0.7 Zero	

Fig. 62: Stick - Trim panel

2.5.4 Output

In this menu the user sets the receiving port and process the incoming commands. Once the stick has been configured, the commands have to be sent to the air unit.

CEXPDI Builder - CEX2 45109 - CONNECTED		- x
CEX 2.0 with ARINC -	😨 Stick	8 2 6 6 0
PPM Exponential Trim Output		
The second secon		
Remote UAV Broadcast Vin perio	d 0.0: s Delta 0 d 0.2 s	
Enable Initial 1 Port 0		
Remote UAV Min perio Max perio	d 0.0 s Delta 0	

Fig. 63: Stick - Output panel

In this menu, the following parameters can be configured:

- Enable.
- **Initial Channel at destination**: The user indicates to which channel of the air autopilot will be sent the first channel received in the ground unit. The channels arrive at the platform in order and without spaces between them.

For example, if at the GND channels 6, 7, 8, 9 and 10 are enabled, the AIR will receive channels 1, 2, 3, 4 and 5. Therefore channel 6 of the stick will be channel 1 in the AIR configuration.

- **Port**: If more than one transmitter is configured, each transmitter must be configured on a different port. This has to match the port set on the air unit.
- Remote: It has to be enabled if the user wants to allow the delivery of the commands to the platform.
 - UAV: The address of the UAV that receive the commands has to be indicated. The following options are the most common:
 - * App 2: Veronte applications address.
 - * Broadcast: The commands are sent to all units on the network. This option is recommended.
 - * 1x v4.X XXXX: The address of a specific air unit.

For more information on the available addresses, see List of addresses section of the CEX Software Manual.

- **Min period**: As the period is the inverse of the frequency, this is the **maximum frequency**. Therefore, to give the pilot more control, this is the frequency that is set when the **stick is commanding**. A *Min period* of **0.02s** is recommended.
- **Max period**: As the period is the inverse of the frequency, this is the **minimum frequency**. Thus, to free up bandwidth, this is the frequency that is set when the **stick is idle**. A *Max period* of **0.2s** is recommended.
- **Delta**: This parameter determines whether the frequency is set to the minimum or maximum period set above.

If CEX detects a **change above** the **delta value**, the frequency goes to the maximum frequency (minimum period). While if the **changes are less than this value**, it switches to the **minimum frequency** (maximum period). **10 Hz** are recommended.

An example of how to configure a stick can be found in the Stick - Integration examples section of the **1x PDI Builder** user manual.



2.6.1 Jetibox

CEX can simulate a Jetibox to read telemetry from legacy Jeti devices.

CEXPDI Builder - CEX2 45109 - CONNECTE	D	- ×
CEX 2.0 with ARINC -	O Devices	
Jetibox Scorpion tribunus Image: Scorpion tribunus	SCI Port SCI A	

Fig. 64: Jetibox panel

- Enable: It can be enabled or disabled by the user.
- SCI Port: A SCI port must be selected. The available options are: SCI A, SCI B or SCI C.

Besides, a configuration is needed in the Serial and I/O Setup panels.

An example of this can be seen in the Jetibox - Integration examples section of the present manual.

Note: The serial port will be totally reserved for this, so it will not be usable to other things and the I/O Setup affecting it will be ignored.

2.6.2 Scorpion tribunus

CEX can read telemetry from Tribunus ESCs by connecting it to one of its serial ports.

CEXPDI	Builder - CEX2 45109 - CONNEC			
CEXPDI	Builder - CEX2 45109 - CONNEC with ARINC - Jetibox Scorpion tribunus	CTED Concernent Concernent Conce	vices 0.02 0.05 0.05 0	

Fig. 65: Scorpion tribunus panel

The following parameters are configurable:

- Enable: It can be enabled or disabled by the user.
- Telemetry period.
- MCU telemetry 1/2 period: Telemetry period for MCU devices.
- PWM ID: PWM ID associated to the Scorpion Tribunus.

The serial port has to be configured in the I/O Setup section.

An example of this can be seen in the Scorpion tribunus - Integration examples section of this manual.

Note: The serial port will be totally reserved for this, so it will not be usable to other things and the I/O Setup affecting it will be ignored.



The arbitration algorithm in Veronte 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.

2.7.1 Arbitration

CEX is able to output PWMs using arbitration in the same way **Veronte Autopilot 4x** does. This functionality has to be enabled as follows:

CEXPDI Builder - CEX2 45010 - CONNECTED - X						
CEX 2.0 with ARINC	🕂 Arbitration					
Arbitration Config CAN Setup Enable - Master arbitration RX CAN ID C Extended CAN C Arbitration Arbitration CAN						

Fig. 66: Arbitration panel

Master arbitration RX CAN Id is exclusive for CEX when it is used alongside a Veronte Autopilot 4x. If **enabled**, when an arbitration message is received from the Arbiter of the Autopilot 4x (**with the CAN Id configured here**), the selected autopilot will be updated according to the data received.

2.7.2 Config

2.7.2.1 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.

2.7.2.2 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.

2.7.2.3 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		
		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.

2.7.2.4 Config panel

In this panel, the parameters of the arbiter algorithm are set.

CEXPDI	Builder - CEX2 45010 - CONNECTED				- ×
CEX 2.0) with ARINC 🔫	📥 Arbitration		8 2 0	
Ö	Arbitration Config CAN Setup				
	Arbiter				
	Preferred Autopilot 1	1			
	Method Always best	~ (2)			
P	Holding CAP time 0.1	s (3)			
œ	Hysteresis 0.001	x1 (4)			
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $	Use external autopilot 5				
	Variables	_(1)			
-	Variable 1 Max error: 0.0	Weight: 1.0 bsolute: 0	\odot		
	Variable 2 Max error: 0.0	Weight: 1.0 Absolute: 0			
	Variable 3 Max error: 0.0	Weight: 1.0 Absolute: 0 9			
		_			

Fig. 67: 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 0 is selected.
 - Fixed 1: Autopilot 1 is selected.
 - Fixed 2: Autopilot 2 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. Use external autopilot: Enables arbitration of external autopilot.
- 6. Add button: Adds new arbitration variables.

Arbitration variables have the following configurable parameters:

- 7. Max error: Arbitration maximum error for each variable.
- 8. Weight: Arbitration weight for each variable.
- 9. 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.

Note: To delete variables click on "-".

Currently, the maximum amount of arbitration variables supported is 32.

2.7.3 CAN Setup

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

Note: If arbitration is not enabled, only the configuration of the Autopilot 0 will be used.

CEXPDI	Builder - CEX2 45109 - CONNECTED		- ×
CEX 2.0	with ARINC -	👍 Arbitration	8 1 4 5 0
Ö	Arbitration Config CAN Setup		
2	Send status Period 0.2	_1	
•	Send score Period 3.0	2	
	Status message ID 255 Extended 3		
	Autopilot 0 4		
	ID 8		
	Autopilot 1		
÷.	ID 9		
	Autopilot 2		
	Extended		
	ID 10		

Fig. 68: CAN Setup panel

1. Send status and Period: The user can enable status sending and set a status message period.

Note: The *Send Status Period* must be set to **0.2 s**, otherwise, the arbitration algorithm will force the selection of autopilot 0.

- 2. Send score and Period: 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 when there is no External Arbitration and therefore CEX works as arbiter.
 - **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 0-2: 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 CAN communication

CEX can send and receive messages via CAN.

3.1.1 CAN messages transmission



Fig. 1: CAN messages transmission - Communication diagram $CEX \rightarrow 1x$

Users must follow the steps below to send telemetry via CAN and configure an Autopilot 1x to read this telemetry:

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

Connect a CAN Custom Producer Telemetry to a CAN Output Filter as follows:

X PDI Builder		🚓 Input / Output			± 0			
GPIO	Configuration	AN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2						
PWM	Priority High %&	Producer CAN Input Filter b		Co	nsumer	Always Uk		
CAN I/O	High	CAN Input Filter 7	-	None	Q0	Always Ok		
Digital Input	High	CAN Input Filter 8	→ <u> </u>	None	Q ₀	Always Ok		
Serial	High	CAN Input Filter 9	→ <u> </u>	None	\$\$	Always Ok		
CAN Setup	High 🙀	CAN Input Filter 10	→ <u> </u>	None	Q0	Always Ok		
	High 🗘	CAN Input Filter 11	-	None	Q0	Always Ok		
]	High 🔯	CAN unwrapper 0	→ <u> </u>	None	Q ^o	Always Ok		
	High 🛱	CAN unwrapper 1	→ <u> </u>	None	Q ₀ ⁰	Always Ok		
	High 🔯	Serial to CAN 0	→	CAN Output Filter 0	Q0	Always Ok		
	High 🛱	Serial to CAN 1	→	None		Always Ok		
	High 🗘	Serial to CAN 2	→ <u> </u>	None	Q ₀ ^o	Always Ok		
	High 🛱	Serial to CAN 3	→	None	Q0	Always Ok		
	High 🛱	Serial to CAN 4	\rightarrow	None		Always Ok		
	High 🗘	Serial to CAN 5	-	None	Q6	Always Ok		
	High 🕸	CAN Custom Producer 0 Tele	\rightarrow	CAN Output Filter 2	Q ₀	Always Ok		
	High 🛱	CAN Custom Producer 1 Tele	\rightarrow	None	\dot{Q}_0^0	Always Ok		
	📃 High 🔯	CAN Custom Producer 2 Tele	-	None	\dot{Q}^0_0	Always Ok		

Fig. 2: CAN messages transmission - CAN I/O configuration

2. Go to Input/Output menu \rightarrow CAN I/O panel \rightarrow CAN Telemetry tab.

This tab must corresponds to the selected CAN Custom Producer Telemetry. In this case CAN Telemetry 0 as the Producer is CAN Custom Producer 0 Telemetry.

Select the fields to send in **TX** or **TX** Ini, as it is a Producer. More information on the configuration of telemetry messages can be found in the *CAN Telemetry - Input/Output* section of this manual.

CEX PDI Builder		- ×
CEX 2.0 with ARINC -	• 🕁 Input / Output	0
GPIO PWM N/O Setup CAN I/O Digital Input Serial CAN Setup CAN Setup CAN Setup	Configuration CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2 TX Ini TX D Edit Copy Ext Can id Endianness Period 00 == 0 90 Little endian 1.0 s 90 Little endian 1.0 s NEX KX Memory usage: 0 / 1280 Fields	+

Fig. 3: CAN messages transmission - CAN Telemetry configuration

For example, a telemetry message with a variable set to ID 90:

CAN	message	: ID: 90										- x
Checksur	n Match	er Skip	Variable	ASCII	Position	Occupancy						
0	0	0	1	0	0							
0 (0)	- 32	- 💷	+	• PWM	1							+
		Variable	1 , , ,				Desimals	Encodo /Decodo	Encode	Decode		
		variable			ompression	1	Decimais	Encode/Decode	Min Ma	AX Min Ma	x	
		PWM 1	Unco	ompress		*	0	1.0	0.0 0.0			

Fig. 4: CAN messages transmission - CAN Telemetry example

3. Configure the **Autopilot 1x** to read telemetry from **CEX**, following the CAN messages reception - Integration examples section of the **1x PDI Builder** user manual.

Note: The configured CAN ID must match the one in CEX.

3.1.2 CAN messages reception





Users must follow the steps below to read CAN messages via CAN and configure an Autopilot 1x to send this telemetry:

- 1. Configure the **Autopilot 1x** to send telemetry to **CEX**, following the CAN messages transmission Integration examples section of the **1x PDI Builder** user manual.
- 2. Go to Input/Output menu \rightarrow CAN Setup panel.

Configure a mailbox according to the message to receive. In this example, a message with ID 90 is received through CAN A:

Note: The configured ID must match the one in 1x.

ARINC ¥		•~		output		2 3
GPIO	CANA	CANB				
PWM I/O Setup	Units I	D DEC 👻	Units Mask	BIN 👻	Baudrate 1000000.0	
AN I/O	Mailbo	oxes reserved RX	: 17 Mailbo	xes available	TX: 15	
ital Input	#	Mailboxe	s Extended	ID	Mask	+
rial	1	4		1302	1111111111	
N Setup	2	- 4		8	1111111111	
	3	- 4		9	1111111111	
	4	- 4		10	1111111111	
	5	— 1		90	1111111111	
	-				-	

Fig. 6: CAN messages reception - CAN Setup configuration

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

Connect a CAN Input Filter to a CAN Custom Consumer Telemetry as follows:

CEX PDI	Builder							- ×			
CEX 2.0	with ARINC 👻		🕂 Input	/ Output		i	£ 0	0			
Ö	GPIO	Configuratio	on CAN Telemetry 0	CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2							
	PWM	Priority		Producer		Cons	umer				
2	I/O Setup	🗌 High 🐇	Application	Processor	\rightarrow	CAN Output Filter 1	\mathbf{Q}_{0}^{0}	Always Ok			
•	CAN I/O	🗌 High ⊀	CAN Inpu	ut Filter 0	\rightarrow	CAN to Serial 0	Q0	Always Ok			
	Digital Input	🗌 High 📢	CAN Inpu	ut Filter 1	\rightarrow	Application Processor	Q0	Always Ok			
P	CAN Setup	🗌 High 【	CAN Inpu	ut Filter 2	\rightarrow	CAN Custom Consumer 0 Tel	Q_0^0	Always Ok			
33		🗌 High 【	CAN Inpu	ut Filter 3	\rightarrow	None	Q_0^0	Always Ok			
		🗌 High 【	CAN Inpu	ut Filter 4	\rightarrow	None	Q_0^0	Always Ok			
		High	CAN Inpu	ut Filter 5	\rightarrow	None	Q_0^0	Always Ok			
÷.		🗌 High 【	CAN Inpu	ut Filter 6	\rightarrow	None	Q0	Always Ok			
		High	CAN Inpu	ut Filter 7	\rightarrow	None	Q_0^0	Always Ok			
		🗌 High 【	CAN Inpu	ut Filter 8	\rightarrow	None	Q0	Always Ok			
		High	CAN Inpu	ut Filter 9	\rightarrow	None	Q_0^0	Always Ok			
		🗌 High 【	CAN Inpu	t Filter 10	\rightarrow	None	Q_0^0	Always Ok			
		High	CAN Inpu	t Filter 11	\rightarrow	None	Q_0^0	Always Ok			
		High 4	CAN unw	rapper 0	\rightarrow	None	Q_0^0	Always Ok			
		High 4	CAN unw	rapper 1	\rightarrow	None	Q_0^0	Always Ok			
		🗌 High 【	Serial to	CAN 0	\rightarrow	CAN Output Filter 0	Q _0	Always Ok			
		High 🖁	Sarial to	CAN 1	\rightarrow	None	100	Alwaye Ok			

Fig. 7: CAN messages reception - CAN I/O configuration

4. Click on the **button** of the **CAN Input Filter** and configure the *Port* and *CAN Id* according to the message to receive.

Note: The configured CAN Id must match the one in 1x.

Port	CAN A	-
ld	90	
Mask	2047	dec
Filter type	Standard	•

Fig. 8: CAN messages reception - CAN Input Filter configuration

5. Go to Input/Output menu \rightarrow CAN I/O panel \rightarrow CAN Telemetry 0 tab (as CAN Custom Consumer 0 Telemetry has been selected as consumer).

Add a field to receive the message in RX, since it is a Consumer, and configure it to store the received value in a

user variable. More information on the configuration of Telemetry messages can be found in the *CAN Telemetry* - *Input/Output* section of this manual.

Note: The configured CAN Id must match the one in 1x.

CEX PDI Builder		- ×
CEX 2.0 with ARINC -	• ← Input / Output 🛓 🗅 🖕	0
GPIO PWM I/O Setup CAN I/O Digital Input © CAN Setup CAN Setup	Configuration CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2 TX Ini TX TX TX TX TO Edit Copy Ext O 90 Little endian 1.0 5 0	-
	Memory usage: 0 / 1280 Fields	

Fig. 9: CAN messages reception - CAN Telemetry configuration

3.2 CAN Isolator

CEX can operate as a CAN bus isolator, since it manages both CAN buses as desired. The system has a built-in microcontroller; which manages CAN buses in real-time. Both buses are not electrically connected, in consequence, electrical signals that do not follow the CAN protocol will be isolated.

The functionalities of a CAN isolator are the following:

- CAN Input Filter
- Filtered CAN subnet
- CAN tunnel

The following diagram summarizes the behavior of CEX as a CAN bus isolator:



Fig. 10: Subnets CAN diagram

3.2.1 Filtered CAN subnet

With **CEX** it is possible to isolate CAN nets, by filtering certain messages from one CAN line and only transmitting specific information through CEX to the other one.

In this example, only a certain range of CAN IDs will be allowed to cross from one CAN line to the other using the two CAN ports of **CEX**. Specifically, information will pass **from CAN B** of CEX **to CAN A** of CEX:



Fig. 11: Subnets CAN example - From CAN B to CAN A

The allowed range will be from 0x550 to 0x55F.

1. Create a new mailbox entry for CAN B.

To do this, go to Input/Output menu \rightarrow CAN Setup panel \rightarrow CAN B tab.

Assign some of the mailboxes to it and set the ID to **0x550**. This example uses an ID expressed in **hexadecimal** notation.

CEXPDI Builder - CEX2 45109 - CONNI	ECTED	- ×
CEX 2.0 with ARINC -	• ← Input / Output	8 2 5 0
CEX 20 with ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup CAN Setup CAN Setup	CANA CANB Units ID HEX Units Mask BIN Baudrate 125000.0 Mailboxes reserved RX: 8 Mailboxes available TX: 24 # Mailboxes Extended ID Mask 1 - 4 516 1111111111 2 = 4 550 1111111111	

Fig. 12: Filtered CAN subnet - CAN Setup configuration

2. Set a Mask which will ignore the last 4 bits.

Note: Although the ID has been entered in hexadecimal notation, note that the Mask has been set in **binary** format.

CEXPDI E	Builder - CEX2 45109 - CONNECT	ED			10			- *
	with ARINC 👻			ہ ≺۔	Input /	8 2 6 9		
ا ال	GPIO PWM I/O Setup CAN I/O	CAN/ Units Mailb	ID ID	ANB HEX 👻 I reserved RX:	Jnits Mask 8 Mailboxe	BIN 🔻	Baudrate 125000.0 TX: 24	
•	Digital Input	#		Mailboxes	Extended	ID	Mask	ndan .
P	CAN Setup	1	8162345	4		516	1111111111	_
3		2	IEINEES	4		550	1111110000	
$\begin{tabular}{ c c } \hline \hline$								
#								
		<						

Fig. 13: Filtered CAN subnet - CAN Setup mask configuration

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

Configure CAN Input Filter 2 on CAN B, with the same settings as the Mailbox.

Important: This menu only allows setting the Id with **decimal numbers**. For this example the Id 0x550 is represented as **1360** in decimal format.

CEXPDI	Builder - CEX2 45109 - CONNEC	TED							- ×			
CEX 2.0	with ARINC -			🕂 Inpu	ut / Output			7	0			
O	GPIO	Configura	Configuration CAN Telemetry 0 CAN			emetry 1 CAN Telemetry 2						
	PWM	Priority			Producer	Consumer						
2	I/O Setup	📃 High	Q_0^0	Applicati	on Processor	\rightarrow	CAN Output Filter 1	00	Always Ok			
•	CAN I/O	High	Q^0_0	CAN In	put Filter 0	\rightarrow	None	Q0	Always Ok			
	Digital Input Serial	High	44Q	(None	00	Always Ok			
P	CAN Setup	🔲 High	00	Port	CAN B	-	Application Processor	Q0	Always Ok			
3		📄 High	\mathbf{Q}_0^0	ld	1360		None		Always Ok			
		High	$\dot{\Omega}^0_0$	Mask	11111110000	bin	None		Always Ok			
		High	\dot{Q}^0_0	Filter type	Standard	•	None	00	Always Ok			
-		High	00	CAN In	put Filter 6	\rightarrow	None	00	Always Ok			
		High	\dot{Q}^0_0	CAN In	put Filter 7	\rightarrow	None	00	Always Ok			
		High	00	CAN In	put Filter 8	\rightarrow	None	Q0	Always Ok			
		High	00	CAN In	put Filter 9	\rightarrow	None	Q0	Always Ok			
		High	$\dot{\Omega}^0_0$	CAN Inp	out Filter 10	\rightarrow	None	\$\$	Always Ok			
		High	00	CAN Inp	out Filter 11	\rightarrow	None		Always Ok			
		High	00	CAN ur	nwrapper 0	\rightarrow	None	Q0	Always Ok			
		High	00	CAN ur	nwrapper 1	\rightarrow	None	\$\$	Always Ok			
		✓ High	00	Serial	to CAN 0	\rightarrow	CAN Output Filter 0	00	Always Ok			
		High	450	Sarial	to CAN 1		CAN Output Filter 2					

Fig. 14: Filtered CAN subnet - CAN Input Filter configuration

4. Bind CAN Output Filter 2 to CAN Input Filter 2, and configure CAN Output Filter to CAN A.

XPDI Builder - CEX2 45109 - C	DNNECTED	⊷t⊶ Input / Output			± (- × 0 4 4
GPIO	Configuration	CAN Telemetry 0 CAN Telemet	ry 1 CAN	Telemetry 2		
PWM	Priority	Producer		Co	nsumer	
I/O Setup	🗌 High 🛱	Application Processor	\rightarrow	CAN Output Filter 1	00	Always Ok
CAN I/O	High 🛱	CAN Input Filter 0	\rightarrow	None	00	Always Ok
Digital Input	High	CAN Input Filter 1	\rightarrow	None	Q0	Aluma Ol
CAN Setup	High 🛱	CAN Input Filter 2	\rightarrow	CAN Output Filter 2	00	Port CAN
	High 🕵	CAN Input Filter 3	\rightarrow	None	Q ₀ ⁰	Aiways OK
	High	CAN Input Filter 4	\rightarrow	None	Q0	Always Ok
	High	CAN Input Filter 5	-	None	Q0	Always Ok
	High 🕵	CAN Input Filter 6	\rightarrow	None	Q0	Always Ok
	High 🔂	CAN Input Filter 7	\rightarrow	None	Q ₀	Always Ok
	High 🙀	CAN Input Filter 8	\rightarrow	None	Q0	Always Ok
	High 🛱	CAN Input Filter 9	\rightarrow	None	Q0	Always Ok
	High 🛱	CAN Input Filter 10	\rightarrow	None	Q0	Always Ok
	High 🛱	CAN Input Filter 11	\rightarrow	None	Q0	Always Ok
	High 🗘	CAN unwrapper 0	\rightarrow	None	Q0	Always Ok
	High 🗘	CAN unwrapper 1	\rightarrow	None	Q0	Always Ok
	✓ High 🕰	Serial to CAN 0	\rightarrow	CAN Output Filter 0	Q0	Always Ok
	High 10	Serial to CAN 1		CAN Output Filter 3	80	Alwave Ok V

Fig. 15: Filtered CAN subnet - CAN Output Filter configuration

3.2.2 CAN tunnel

CAN tunnel is a specific type of *a message tunnel*; where messages are tunneled from one CAN port of CEX to the other.

In this example, a transparent tunnel will be created. So, any messages received on **Interface A** will be sent through **Interface B**.



Fig. 16: Subnets CAN example - From CAN A to CAN B

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

1. Create a new mailbox entry for **Interface A**.

To do this, 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.

CEXPDI	Builder - CEX2 45109 - CONNEC	TED						- ×
CEX 2.0	with ARINC 👻			•	Input /	Output		8 2 6 5 0
	with ARINC + GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	CANA Units Mailt 1 2 3	A C/	ANB DEC VI reserved RX: Mailboxes 4 8 8	Jnits Mask 20 Mailbox Extended	BIN BIN BIN BIN BIN BIN BIN BIN BIN BIN BIN	Baudrate 1000000.0 TX: 12 Mask 1111111111 0 0	
		<						

Fig. 17: CAN tunnel - CAN Setup configuration

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

Configure CAN Input Filter 2 on CAN A, with a Mask of 0 and Both as frame format type.

CEXPDI	Builder - CEX2 45109 - CONNEC	CTED		na an a	1931/931/937/937/		11 - 21 - 21 - 21 - 21 - 21 - 21 - 21 -	ndenikan enikan (s. – .)	
	with ARINC 👻		🚓 Input /	/ Output			* 0	0	
0	GPIO	Configuration	Configuration CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2						
	PWM	Priority	P	roducer		Cor	sumer		
2	I/O Setup	High 😳	Application	Processor	\rightarrow	CAN Output Filter 1	00	Always Ok	
•	CAN I/O	High 🔯	CAN Input	Filter 0	\rightarrow	CAN to Serial 0		Always Ok	
	Digital Input Serial	High 🗮				pplication Processor	00	Always Ok	
P	CAN Setup	High 🔯	Port	CAN A	•	CAN to Serial 1	Q ₀ ⁰	Always Ok	
650		High 🛱	ld (0		AN GPIO consumer 0		Always Ok	
		High 🔯	Mask 0)	dec	CAN Output Filter 5	Q0	Always Ok	
		High 🔯	Filter type	Both	•	None	00	Always Ok	
÷.		High 🕵	CAN Input	Filter 6	\rightarrow	None	Q0	Always Ok	
		High 🔯	CAN Input	Filter 7	\rightarrow	None	00	Always Ok	
		High 🔯	CAN Input	Filter 8	\rightarrow	None	Q0	Always Ok	
		High	CAN Input	Filter 9	\rightarrow	None	Q0	Always Ok	
		High 🔯	CAN Input	Filter 10	\rightarrow	None	00	Always Ok	
		High 🕵	CAN Input	Filter 11	\rightarrow	None	Q0	Always Ok	
		High 🕵	CAN unwr	apper 0	\rightarrow	None	Q0	Always Ok	
		High 🕵	CAN unwr	apper 1	\rightarrow	None		Always Ok	
		High 🕵	Serial to (CAN 0	\rightarrow	CAN Output Filter 0	Q ^o	Always Ok	
		High 108	Serial to (CAN 1	-	CAN Output Filter ?	80	Alwave Ok	

Fig. 18: CAN tunnel - CAN Input Filter configuration

3. Bind CAN Output Filter 2 to CAN Input Filter 2, and configure the CAN Output Filter to CAN B.

CEXPDI Builder - CEX2 45109 - CO	NNECTED	🚓 Input / Output			* (-	×
GPIO GPIO	Configuration	CAN Telemetry 0 CAN Telemetry	y 1 CAN	Telemetry 2			
PWM	Priority	Producer		Cor	nsumer		
る I/O Setup	High 0	Application Processor	\rightarrow	CAN Output Filter 1	00	Always Ok	â
CAN I/O	High 🔯	CAN Input Filter 0	\rightarrow	CAN to Serial 0	Q ₀ ^o	Always Ok	
Digital Input	High 🔯	CAN Input Filter 1	\rightarrow	Application Processor	Q0	AL OL	
CAN Setup	High 🕸	CAN Input Filter 2]→[CAN Output Filter 2	00	Port	CAN
3	High 🕸	CAN Input Filter 3	\rightarrow	CAN GPIO consumer 0	Q ₀	Aiways Ok	1-11
3	High 🔯	CAN Input Filter 4]→[CAN Output Filter 5	\mathbf{Q}_{0}^{0}	Always Ok	
	High 🔯	CAN Input Filter 5	\rightarrow	None	Q ₀ ^o	Always Ok	
ii dha an	High 🕵	CAN Input Filter 6	\rightarrow	None	Q0	Always Ok	
	High 🕵	CAN Input Filter 7	\rightarrow	None	Q0	Always Ok	
	High 🛱	CAN Input Filter 8	\rightarrow	None	Q0	Always Ok	
	High 🛱	CAN Input Filter 9	\rightarrow	None	Q0	Always Ok	
	High 🛱	CAN Input Filter 10	\rightarrow	None	Q0	Always Ok	U
	High 🛱	CAN Input Filter 11	\rightarrow	None	Q0	Always Ok	
	High 🕸	CAN unwrapper 0	\rightarrow	None	Q0	Always Ok	
	High 🕸	CAN unwrapper 1	\rightarrow	None	Q0	Always Ok	
	High 🔯	Serial to CAN 0	→	CAN Output Filter 0	$\dot{\mathbf{Q}}^{0}_{0}$	Always Ok	
	High 62	Sarial to CAN 1		None	36	Alwave Ok	~

Fig. 19: CAN tunnel - CAN Output Filter configuration

3.3 Commanding/Reading PWMs

The appropriate PWM message format is described in the Command PWMs - CAN Bus Protocol section of **CEX** Software Manual.



Fig. 20: PWM Command - Communication diagram $1x \rightarrow CEX$

The user can follow the following steps to achieve PWM commanding from 1x to CEX:

3.3.1 1x PDI Builder side

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

Connect a CAN custom message to an Output filter to send the message:

1xVeron	tePDI Builder							- × `
1x 4.8	×		•੯ <mark>•</mark> Inp	ut / Output			8 1 0	9
O	I/O Setup	Configuratio	n Custom message 0	Custom message 1	Custom message 2	Mailboxes		
	CAN Setup	Priority High 1	P Sen	roducer al to CAN 3		Consumer None	¥8	Always Uk
	Digital Input Serial	High 🗘	Seri	al to CAN 4	\rightarrow	None	08	Always Ok
2		High 🗘	Seri	al to CAN 5	\rightarrow	None	08	Always Ok
•		High 🗘	CAN cu	stom message 0	\rightarrow	None	08	Always Ok
		High 🗘	CAN cus	stom message 1	→	Output filter 2	Q ^o	Always Ok
		🗌 High 🔅	CAN cus	stom message 2	\rightarrow	Output filter 3	Q ^o	Always Ok
0,		V High 🗘	6 Inj	put filter 0	\rightarrow	CAN to serial 0	OS	Always Ok
P		V High 🗘	e Inj	put filter 1	\rightarrow	CAN to serial 1	Q ^o	Always Ok
8		🗌 High 🚺	e Inj	put filter 2	\rightarrow	Custom message 0	Q00	Always Ok
		🗌 High 🚺	e In	put filter 3	\rightarrow	Custom message 1	Q ₀ ^o	Always Ok
		🗌 High 🚺	te Inj	put filter 4	\rightarrow	None	Q ₀ ^o	Always Ok
\mathbf{O}		🗌 High 🚺	P Inj	put filter 5	\rightarrow	None	Q ₀ ⁰	Always Ok
.		🗌 High 🗘	CAN	unwrapper 0	\rightarrow	None	Q ₀	Always Ok
<i>A</i> D		🗌 High 🗘	CAN	unwrapper 1	\rightarrow	None	Q ₀	Always Ok
		🗌 High 🛱	CAN G	GPIO remote 0	\rightarrow	None	Q0	Always Ok
X		📄 High 🛱	CAN C	GPIO remote 1	\rightarrow	None	O ₀	Always Ok
		High 🗳		CAN 4x	\rightarrow	None	00	Always Ok

Fig. 21: 1x PDI Builder - CAN Setup configuration

2. Go to Input/Output menu \rightarrow CAN Setup panel \rightarrow Custom Message 2 tab (as CAN custom Message 2 has been selected as producer).

Build a CAN custom message using the PWM variable and the appropriate message format.

Select the fields to send in the **TX** as it is a Producer. CAN ID is arbitrary, for this example ID 100 will be used:

1xVerontePDI Builder		- ×
1x 4.8 👻	•🛟 Input / Output	0
I/O Setup	Configuration Custom message 0 Custom message 1 Custom message 2 Mailboxes	
CAN Setup	► TX Ini	
Serial	▼ TX	
	ID Edit Copy Ext Can id Endianness Period	
E	CAN message ID: 100 Checksum Matcher Skip Variable ASCII Position Occupancy	X
¢\$	0 1 0 1 0 🔳	
	0 (0) - 8 — 🗊 🗇 💠 Matcher x2	militan
D	Value Bits Mask	
\bigcirc	Variable Compression Bits Encode/Decode Min Max Min Max	
	PWM 1 Compress - Bits Unsigned T2 1.0 0.0 1.0 0 4095	
æ		
X		
	< <u>(</u>)	>

Fig. 22: 1x PDI Builder - CAN Custom message configuration

Message format:

- Matcher (2) [8 bits]
 - PWM 0 [12 bits]
 - PWM 1 [12 bits]
 - PWM 2 [12 bits]
 - PWM 3 [12 bits]
- Matcher (3) [8 bits]
 - PWM 4 [12 bits]
 - PWM 5 [12 bits]
 - PWM 6 [12 bits]

- PWM 7 [12 bits]
- Each PWM variable has to be set using the following format:
 - Variable: PWM X
 - Compression: Compress Bits Unsigned
 - **Bits**: 12
 - Encode: Min=0.0 / Max=1.0
 - Decode: Min=0 / Max=4095

3.3.2 CEX PDI Builder side

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

Create the mailbox to receive the new message (ID 100):

EX 2.0 with ARINC -		•	· Input /	Output		* 0 5 0
CEX 2.0 with ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup CAN Setup The serial	CAN Units Maill # 1 2	ANB DEC V U s reserved RX: 4 1	Input /	Output	Baudrate 1000000.0 X: 27 Mask 1111111111 1111111111	
	<					

Fig. 23: CEX PDI Builder - CAN Setup configuration

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

Connect a CAN Input Filter with the right CAN ID to the Application Processor consumer:

		- 1 - 1							
GPIO	Configuration (Configuration CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2							
PWM	Priority	Producer		Со	nsumer				
I/O Setup	High 📿	Application Processor	\rightarrow	None	05	Always C			
CAN I/O	High 🕵	CAN Input Filter 0	→	Application Processor	Qo	Always C			
Serial	High 🕵	CAN Input Filter 1	\rightarrow	None	Q0	Always C			
CAN Setup	High 🙀	CAN Input Filter 2	\rightarrow	None	Q0	Always C			
	High 🕵	CAN Input Filter 3	\rightarrow	None	Q0	Always C			
	High 🔯	CAN Input Filter 4	\rightarrow	None	Qo	Always C			
	High 🕵	CAN Input Filter 5	\rightarrow	None	Q0	Always C			
	High 🕵	CAN Input Filter 6	\rightarrow	None	Q0	Always C			
	High 🙀	CAN Input Filter 7	\rightarrow	None	Q0	Always C			
	High 🕵	CAN Input Filter 8	\rightarrow	None	Q0	Always C			
	High 🔯	CAN Input Filter 9	\rightarrow	None	Q0	Always C			
	High 🕵	CAN Input Filter 10	\rightarrow	None	Q0	Always C			
	High 🕵	CAN Input Filter 11	\rightarrow	None	Q	Always C			
	High 😡	CAN unwrapper 0	\rightarrow	None	Qo	Always C			
	High 🕵	CAN unwrapper 1	\rightarrow	None	Qo	Always C			
	Sub		1.						

Fig. 24: CEX PDI Builder - CAN I/O configuration



Fig. 25: CEX PDI Builder - CAN Input Filter configuration

5. Go to Input/Output menu \rightarrow PWM panel \rightarrow PWM 1 tab (as PWM 1 has been selected in 1x PDI Builder). Configure parameters for PWM 1:

CEX PDI Builder		- >
CEX 2.0 with ARINC -	• C→ Input / Output	0
GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup CO CT CO CO	 PWM 0 PWM 1 Image: PWM 2 Sub id Image: PWM 2 Image: PWM 3 Image: PWM 4 Image: PWM 4 Image: PWM 6 Image: PWM 7 	

Fig. 26: CEX PDI Builder - PWM configuration

6. Go to Input/Output menu \rightarrow GPIO panel.

Check that this PWM pin (**PWM 1** in this case) has correctly switched from GPIO to PWM. It should look like this:

CEX PD	l Builder						- x
CEX 2.0	with ARINC 🔹		•€ In	put / Output		*	0 4 4
\bigcirc	GPIO	Signal	GPIOId	ю	Pull-up	Function	Qsel
	PWM	GPIO/ECAP 0	GPIO 24	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
ッ	I/O Setup	GPIO/ECAP 1	GPIO 25	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN I/O	GPIO/ECAP 2	GPIO 26	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
•	Digital Input	GPIO/ECAP 3	GPIO 27	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
0	Serial	PWM 0	GPIO 0	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN Setup	PWM 1	GPIO 1	GPIO as output	Pull-up disabled	Mux 1	Sync
670		PWM 2	GPIO 2	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 3	GPIO 3	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
•		PWM 4	GPIO 4	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
.		PWM 5	GPIO 5	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 6	GPIO 6	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 7	GPIO 7	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 9	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Svnc v

Fig. 27: CEX PDI Builder - GPIO configuration

- IO: GPIO as output.
- Function: Mux 1.
- 7. Finally, save the changes.

3.4 GPIO Command

The following are the steps to send a GPIO command from the **Veronte Autopilot 1x**, receive it at **CEX** and process it, so that **CEX** carries out the command.

Warning: Remember that for the reception of CAN messages, Mailboxes need to be configured accordingly.

GPIO Command is very similar to PWM command with a few exceptions.

3.4.1 1x PDI Builder side

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

Connect, as Producer, a 'CAN GPIO remote' to an Output filter:

1xVeront	tePDI Builder		in (naga sga er su an se su		n (n) (n) (n) (n) (n) (n) N (n) (n	10 (20 (0) (20			- ×
1x 4.8	*			🚓 Input / Outpu	t			8 1 0	9
0	I/O Setup	Configurati	ation Custom message 0 Custom message 1 Custom message 2 Mailboxes		Mailboxes				
	CAN Setup	Priority High	ψt _O	Producer Senal to CAN 3			Consumer None	Wo .	Always Uk
	Serial	High H	¢°	Serial to CAN 4		\rightarrow	None	Q0	Always Ok
۳		High H	ϕ_{α}^{α}	Serial to CAN 5		\rightarrow	None	Q0	Always Ok
•		High H		CAN custom message (D	\rightarrow	None	Q0	Always Ok
		High		CAN custom message	1]→	Output filter 2	Q0	Always Ok
		High H		CAN custom message 2	2	\rightarrow	None	QC C	Always Ok
Ω.;		V High	¢₿	Input filter 0		→	CAN to serial 0	Q0	Always Ok
P		V High	$\dot{\Omega}^{0}_{0}$	Input filter 1		\rightarrow	CAN to serial 1	Q0	Always Ok
c 20		High H	$\dot{\mathbf{D}}_{0}^{0}$	Input filter 2		\rightarrow	Custom message 0	Q0	Always Ok
		High H	$\dot{\mathbf{D}}^{0}_{0}$	Input filter 3		\rightarrow	Custom message 1	Q0	Always Ok
		High H	$\dot{\mathbf{Q}}^{0}_{0}$	Input filter 4		\rightarrow	None	0°	Always Ok
		High H	$\dot{\mathbf{Q}}^{0}_{0}$	Input filter 5		\rightarrow	None	Q ₀	Always Ok
		High H		CAN unwrapper 0		\rightarrow	None	Q ₀	Always Ok
68		High H		CAN unwrapper 1		\rightarrow	None	Q0	Always Ok
		High H	\mathbf{Q}_{0}^{0}	CAN GPIO remote 0		\rightarrow	None	Q ₀ ^o	Always Ok
		High H	Q ₀ ^o	CAN GPIO remote 1		\rightarrow	Output filter 3	Q ₀	Always Ok
U		High H		CAN 4x		\rightarrow	None	¢\$	Always Ok

Fig. 28: 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.

	CANL		NI
R	Period	0.1	s
	ld of the genera CAN messages	ted 40	dec Extended
	Destination	Value	Ĵ
	Virtual 01	Virtual 01	-
	Virtual 02	Virtual 01	-
	Virtual 03		-
	Virtual 04		

Fig. 29: 1x PDI Builder - CAN GPIO remote configuration

2. Go to Automations menu.

GPIO must be activated using an 'Output action':

1xVerontePDI Builder	Automations	
Image: Second system 30 - GPIO Command Image: Second system Envelope Image: Second system Fail Safe Image: Second system Transitions Image: Second system Stick Image: Second system Buttons Image: Second system Stick Image: Secon	GPIO Command Events AND OR NOT Actions + 51 - GPIO activated 20 - Activate GPIO	
Image: Second state Image: Second state	Delay 0.0 s Periodical: Off	

Fig. 30: 1x PDI Builder - Automation configuration

3.4.2 CEX PDI Builder side

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

Finally, connect a CAN Input Filter to the CAN GPIO consumer:

CEX PD	Builder				909039090399		igenen generalig	- X	
CEX 2.0	with ARINC 👻			🕂 Input / Output		(* 0		
0	GPIO Configuration			CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2					
	PWM	Priority		Producer		Con	sumer		
2	I/O Setup	High	Q_0^0	Application Processor	\rightarrow	None	Q ₀	Always Ok	
•	CAN I/O	High	Q_0^0	CAN Input Filter 0	→	Application Processor		Always Ok	
	Digital Input Serial	📃 High	Q_0°	CAN Input Filter 1	\rightarrow	CAN GPIO consumer 0	Q ^o	Always Ok	
P	CAN Setup	High	Q_{0}^{0}	CAN Input Filter 2	\rightarrow	None	Q0	Always Ok	
639		High	00	CAN Input Filter 3	\rightarrow	None	Q0	Always Ok	
		High	Q_0°	CAN Input Filter 4	\rightarrow	None	Q0	Always Ok	
		High	Q_0^0	CAN Input Filter 5	\rightarrow	None	Q0	Always Ok	
		High	Q_0^0	CAN Input Filter 6	\rightarrow	None		Always Ok	
		High	Q_0^0	CAN Input Filter 7	\rightarrow	None	Q0	Always Ok	
		High	Q_0°	CAN Input Filter 8	\rightarrow	None	Q ^o	Always Ok	
		High	Q_0^0	CAN Input Filter 9	\rightarrow	None		Always Ok	
		High	Q_0^0	CAN Input Filter 10	\rightarrow	None	Q0	Always Ok	
		High	Q_0^0	CAN Input Filter 11	\rightarrow	None	Q ₀	Always Ok	
		High	Ω_0^0	CAN unwrapper 0	\rightarrow	None	O ₀	Always Ok	
		High	Q_0^0	CAN unwrapper 1	\rightarrow	None	Q0	Always Ok	
		High	Q_0°	Serial to CAN 0	\rightarrow	None	Q0	Always Ok	
		High 1	45	Serial to CAN 1		None	80	Alwave Ok	

Fig. 31: CEX PDI Builder - CAN I/O configuration

The Id must match the one configured in the 1x PDI Builder as Output filter:



Fig. 32: CEX PDI Builder - CAN Input Filter configuration

3.5 Reading/Sending RPMs

This section presents the steps to follow to read RPMs.

3.5.1 CEX PDI Builder side

1. Go to Input/Output menu \rightarrow **GPIO panel**.

RPM can be read on the available digital inputs GPIO/ECAP 0-3. The chosen pin needs to be configured as "GPIO as input". In the example shown here, GPIO/ECAP 1 is chosen.

CEX PDI	Builder						- x
CEX 2.0	with ARINC -		•⇐ Inj	4 6 9			
\bigcirc	GPIO	Signal	GPIOId	10	Pull-up	Function	Qsel
	PWM	GPIO/ECAP 0	GPIO 24	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
2	I/O Setup	GPIO/ECAP 1	GPIO 25	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN I/O	GPIO/ECAP 2	GPIO 26	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
	Digital Input	GPIO/ECAP 3	GPIO 27	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
Ø	Serial	PWM 0	GPIO 0	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN Setup	PWM 1	GPIO 1	GPIO as output	Pull-up disabled	Mux 1	Sync
CD		PWM 2	GPIO 2	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 3	GPIO 3	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 4	GPIO 4	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
-		PWM 5	GPIO 5	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 6	GPIO 6	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		PWM 7	GPIO 7	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
		I/O 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 9	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Svnc 🗸

Fig. 33: CEX PDI Builder - GPIO configuration

2. Go to Input/Output menu \rightarrow **Digital Input panel**.

There are 4 possible producers: CAP 0-3. One needs to be chosen and linked to one of the RPMs consumers (RPMs 0-3). For more information on Digital input configuration, see *Digital Input - Input/Output* section of this manual.

Then, select an edge dectection option: "First rising edge" or "First falling edge".

GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup CAN Setup CAN Setup CA	
PWM V/O Setup CAP 0 None Image: Consumer in the set of the set	
None None CAN I/O Digital Input Serial CAN Setup C	
CAN I/O Digital Input Digital Input	Always Ok
Digital Input Serial CAN Setup CAN Setup CAP 3 None CAP 3 None	Always Ok
Image: Serial can be can b	Always Ok
	Always Ok

Fig. 34: CEX PDI Builder - Digital Input configuration



Fig. 35: CEX PDI Builder - Edge detection configuration

3. Go to Sensors menu \rightarrow RPM panel \rightarrow RPM 1 tab (as RPM 1 has been selected in the Digital Input panel).

Here the expected pulse needs to be defined. For more information on RPM configuration, see *RPM* - *Sensors* section of this manual.

CEX PDI Builder		- X
CEX 2.0 with ARINC -	<i>ت</i> و ا	ensors 🛓 🛆 🗲 🛛 0
RPM Lidar	RPM 0 RPM 1 RPM 2 RPM	3
	Units	32 Pulses per cycle 🔹
	Average	5
	Minimum	1.0E-4 s
	Maximum	0.5 s

Fig. 36: CEX PDI Builder - RPM configuration

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

Connect a **CAN Custom Producer Telemetry** to a **CAN Output Filter** as follows (in this example, the **RMP1** variable is sent via **CAN B** bus of the **CEX**):

PDI Builder	n de lieu	ייזי בארי היא מייז בייזי איז איז איז איז איז איז איז איז איז	יונה איז וראי או האיז וראי לעריי או האייין נ	כונה ישונה ישורה ישונה ישונה ישונה יישונה יי	- ×
0 with ARINC 👻		•੯ Input / Output		* (0
GPIO	Configuration	CAN Telemetry 0 CAN Telemetry	1 CAN Telemetry 2		
PWM	Priority High 🐙	Producer CAN Input Filter b	Nor	e Consumer	Always Uk 🔨
CAN I/O	High	CAN Input Filter 7	Nor	ie Ø	Always Ok
Digital Input	High	CAN Input Filter 8	Nor	e Ø	Always Ok
Serial	High 🙀	CAN Input Filter 9	Nor	e Ø	Always Ok
CAN Setup	High 🕵	CAN Input Filter 10	Non	e Ø	Always Ok
	High 🕵	CAN Input Filter 11	Non	e Ø	Always Ok
	High 🔯	CAN unwrapper 0	Non	e Ø	Always Ok
	High 😡	CAN unwrapper 1	Non	e Q ₀ ⁰	Always Ok
	High 🔯	Serial to CAN 0	Non	e Ø	Always Ok
	High 🔯	Serial to CAN 1	Non	e Q ₀ ⁰	Always Ok
	High 🔯	Serial to CAN 2	Non	e Q ₀ ⁰	Always Ok
	High 🛱	Serial to CAN 3	Non	e Q ₀ ⁰	Always Ok
	High 🛱	Serial to CAN 4	Non	e Q ₀ ⁰	Always Ok
	High 🛱	Serial to CAN 5	Non	e Q ₀ ⁰	Always Ok
	High 😡	CAN Custom Producer 0 Tele	Non	e Q ₀ ⁰	Al
	High 🕸	CAN Custom Producer 1 Tele	CAN Outpu	it Filter 1	Port See CAN
	High 🔯	CAN Custom Producer 2 Tele	Non	e Q ₀ ⁰	Aiways OK

Fig. 37: CEX PDI Builder - CAN I/O configuration

5. Go to the **CAN Telemetry tab** that corresponds to the selected CAN Custom Producer Telemetry. In this case CAN Telemetry **1** as the Producer is CAN Custom Producer **1** Telemetry.

A new telemetry message needs to be created with its correspondent ID, endianness and period.

- Can ID: 1200.
- Endianness: Little endian.
- Period: 0.01 s.

In the telemetry message one of CEX's variables needs to be selected. As **RPM 1** has been chosen as consumer in the **Digital Input**, the variable to be sent is **RPM1**.

CEX PD	l Builder		- ×
CEX 2.0	with ARINC -	⊷ Input / Output	0 2 0
) ® 👫 🖉 C	GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration CAN Telemetry 0 CAN Telemetry 1 CAN Telemetry 2 TX Ini TX ID Edit Copy Ext Can id Endianness Period 00 I200 Little endian 0.01 s S	+
	CAN message ID: 1200 Checksum Matcher Skip Vari 0 0 0 1 0 (0) - 32 - I C C C C C C C C C C C C C C C C C C	able ASCII Position Occupancy 1 0 0	-×

Fig. 38: Reading RPMs - CAN Telemetry configuration

More information on the configuration of Telemetry messages can be found in the *CAN Telemetry - Input/Output* section of the present manual.

3.5.2 1x PDI Builder side

6. Go to UI menu \rightarrow Variables panel \rightarrow **Real Vars tab**.

Rename a Real User Variable (32 bits) that will be used to store the value received from **CEX**:

1xVeron	tePDI Builder					- × -
1x 4.8	×		🚯 UI		8 1	0 2 0
0	Operation elements Variables	Bits	Unsigned Real Vars Features			
\otimes		user				
۳		1d 3228	User Variable 128 (Real - 32 Bits)	Custom name	Custom Type ()	Init value (SI)
•4		3229	User Variable 129 (Real - 32 Bits)	CEX RPM 1	Custom Type ()	
		3231	User Variable 131 (Real - 32 Bits)		Custom Type ()	
		3232	User Variable 132 (Real - 32 Bits)		Custom Type ()	
Q [*]		3233	User Variable 133 (Real - 32 Bits)		Custom Type ()	
P		3234	User Variable 134 (Real - 32 Bits)		Custom Type ()	
50		3235	User Variable 135 (Real - 32 Bits)		Custom Type ()	0
		3236	User Variable 136 (Real - 32 Bits)		Custom Type ()	
		3237	User Variable 137 (Real - 32 Bits)		Custom Type ()	
		3238	User Variable 138 (Real - 32 Bits)		Custom Type ()	
		3239	User Variable 139 (Real - 32 Bits)		Custom Type ()	
2		3240	User Variable 140 (Real - 32 Bits)		Custom Type ()	
X		3241	User Variable 141 (Real - 32 Bits)		Custom Type ()	
		3242	User Variable 142 (Real - 32 Bits)		Custom Type ()	
		3243	User Variable 143 (Real - 32 Bits)		Custom Type ()	~

Fig. 39: 1x PDI Builder - User Variable renamed

7. Go to Input/Output menu \rightarrow CAN Setup panel \rightarrow Mailboxes tab.

Configure, in CAN B, some mailboxes to receive the message with ID 1200:

1xVeronte	ePDI Builder									X	
1x 4.8	*				•🕁 Inp	ut / Outp	ut		8 1 8	6	
0	I/O Setup	Con	Configuration Custom message 0 Custom message 1 Custom message 2 Mailboxes								
\otimes	CAN Setup Digital Input	CAN	A	CAN B							
2	Serial	Unit	s ID	DEC 🔹	Jnits <mark>M</mark> ask	BIN 👻	Baudrate 1000000.0	Enable Terminate	or		
•		Mail	boxes	reserved RX:	8 Mailboxe	es available T	X: 24				
		#	8000000	Mailboxes	Extended	ID 1301	Mask	mim			
		2	8000005	4		1200	111111111				
5											
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $											
æ											
X											
		< _)	>			

Fig. 40: 1x PDI Builder - Mailbox configuration

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

Connect an **Input filter** to a **Custom message consumer**. Both CAN buses of the Autopilot 1x can be used, as well as normal IDs and extended IDs:

1xVeron	tePDI Builder							- >
	v		🚓 Inpu	ut / Output			8 1 0	9
0	I/O Setup	Configuration	Custom message 0	Custom message 1	Custom message 2	Mailboxes	1990 TH NGC THE BOOM OF THE CASE AND AND THE CASE AND AND THE T	ch transfordio fin do fin fisici ni adaptato fin do fin fisici
	CAN Setup	Priority High 🔩	Pr Sena	oducer al to CAN 3		Consumer	520	Always Uk
	Serial	High 🗘	Seria	al to CAN 4	\rightarrow	None	Q ₀	Always Ok
۳		High 🗘	Seria	al to CAN 5	\rightarrow	None	Q ₀	Always Ok
•		High 🖏	CAN cus	tom message 0	\rightarrow	None	Q ₀	Always Ok
		High 🕸	CAN cus	tom message 1	→	Output filter 2	Q0	Always Ok
		High 😳	CAN cus	tom message 2	\rightarrow	Output filter 3	Q0	Always Ok
Q 2		V High	Inp	out filter 0	\rightarrow	CAN to serial 0		Always Ok
P		V High	Inp	out filter 1	\rightarrow	CAN to serial 1		Always Ok
679		High 🔯	Inp	out filter 2	→	Custom message 0		Always Ok
		High	Inp	out filter 3	\rightarrow	Custom message 1	Q ₀	Always Ok
		High 🗘	Inp	out filter 4	\rightarrow	None	Q0	Always Ok
$\ \ \circ \ \ $		High 🗘	Ing	out filter 5	\rightarrow	None	\$\$°	Always Ok
		High 40	CAN	unwrapper 0	\rightarrow	None	\$\$	Always Ok
æ		High 🛱	CAN	unwrapper 1	\rightarrow	None	\$\$	Always Ok
		High 🗘	CAN G	PIO remote 0	\rightarrow	None	\$\$	Always Ok
-		High 🗘	CAN G	PIO remote 1	\rightarrow	None	\$\$	Always Ok
		High 40		CAN 4x	\rightarrow	None	Q ₀	Always Ok

Fig. 41: 1x PDI Builder - CAN Setup configuration



Fig. 42: 1x PDI Builder - Input filter configuration

9. Go to Input/Output menu \rightarrow CAN Setup panel \rightarrow Custom message 1 tab (as Custom Message 1 has been selected as consumer).

Configure the reading of the message and store the received value in the user variable renamed above:

1xVerontePDI Builder			- ×
1x 4.8 -	• 🛶 Input / Output	8 1 0	• • •
Ix 4.8 VO Setup CAN Setup Digital Input Serial CAN me Image: Serial Can me	Image: Configuration <		-×
Image: Constraint of the second s	ID Edit Copy Ext Can id Endianness Time out Bit ID 00 III IIII IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	+	-

Fig. 43: 1x PDI Builder - Custom Message configuration

3.6 Serial communication

The user can send information through the serial ports of the **CEX**. For this purpose, please note the following explanation:

3.6.1 CEX PDI Builder side

- 1. I/O Setup configuration
 - Reception of serial information on RS232-A (producer) is stored in Serial to CAN 2 (consumer).
 - Transmission of serial information is sent using the CAN to Serial 2 (producer) via RS232-A (consumer).

('Serial to CAN' and 'CAN to Serial' are explained in the CAN I/O - Input/Output section of this manual.)

CEX PD)l Builder							>
CEX 2.0) with ARINC 👻			•🛟 Input / Output	:		* 0	0
0	GPIO	Configura	tion					
	PWM	Priority	_	Producer		(onsumer	
2	I/O Setup	High	Q ₀ ^o	RS232-A	\rightarrow	Serial to CAN 2	00	Always Ok
·~	CAN I/O	High	Q0	RS232-B	\rightarrow	Commgr port 2	¢¢	Always Ok
	Digital Input	High	Q0	RS485-C	\rightarrow	Commgr port 3	Q\$	Always Ok
P	CAN Setup	High	Q ₀ ^o	CAN to Serial 0	\rightarrow	None	OS I	Always Ok
630		High	Q0	CAN to Serial 1	\rightarrow	None	08	Always Ok
		High	00	CAN to Serial 2	\rightarrow	RS232-A	Q0	Always Ok
		High	00	CAN to Serial 3	\rightarrow	None	Q\$	Always Ok
÷.		High	Q0	CAN to Serial 4	\rightarrow	None	Q\$	Always Ok
		High	Q ₀ ⁰	CAN to Serial 5	\rightarrow	None	OS C	Always Ok
		High	Q ⁰ ₀	Commgr port 0	\rightarrow	None	08	Always Ok
		High	Q ₀ ⁰	Commgr port 1	\rightarrow	None	Q ^o	Always Ok
		High	00	Commgr port 2	\rightarrow	None	Q0	Always Ok
		High	Q0	Commgr port 3	\rightarrow	None	OS I	Always Ok
		High	00	Commgr port 4	\rightarrow	None	Q ⁰	Always Ok
		High	Q ₀ ⁰	Commgr port 5	\rightarrow	None	¢,	Always Ok
		High	Q0	Tunnel 0	\rightarrow	None	Q0	Always Ok
		Hiab	88	Tunnel 1		None	80	Alwaye Ok

Fig. 44: CEX PDI Builder - I/O Setup configuration

2. CAN I/O configuration

- The information that will be sent via serial port RS232-A is going to be received on the CEX through its CAN B port. A CAN Id of 51 is added to the CAN Input Filter. The incoming information (from Veronte Autopilot 1x) is processed in 'CAN to Serial 2'.
- The information **coming from port RS232-A** and processed as '**Serial to CAN 2**' is going to be linked to a **CAN Output Filter**. The information of '**Serial to CAN 2**' is going to be sent via **CAN B** with a **CAN ID of 50** (to be read by **Veronte Autopilot 1x**).
| CEX PDI | Builder | | | | | | | - | | | | | |
|--------------|---------------------------------------|-----------|----------------------|----------------------------|---------------|-----------------------|-------------------------------|-----------|--|--|--|--|--|
| CEX 2.0 | CEX 2.0 with ARINC - + Input / Output | | | | | | | | | | | | |
| Ö | GPIO | Configura | ation | AN Telemetry 0 CAN Telemet | ry 1 CAN | Telemetry 2 | | | | | | | |
| | PWM | Priority | -) (| Producer | | Consumer | | | | | | | |
| 9 | I/O Setup | High | \mathbf{Q}_{0}^{0} | CAN Input Filter 1 | → | Application Processor | Q | Always Ok | | | | | |
| •-{* | Digital Input | 🗌 High | \mathbf{Q}^{0}_{0} | CAN Input Filter 2 | → | CAN to Serial 1 | Q0 | Always Ok | | | | | |
| 0 | Serial | 🗌 High | 00 | CAN Input Filter 3 | \rightarrow | CAN to Serial 2 | QC . | Always Ok | | | | | |
| | CAN Setup | 📄 High | \mathbf{Q}^{0}_{0} | CAN Input Filter 4 | \rightarrow | None | Q ₀ | Always Ok | | | | | |
| 600 | | High | \mathbf{Q}_0^0 | CAN Input Filter 5 | \rightarrow | None | Q0 | Always Ok | | | | | |
| \mathbf{O} | | High | Q^0_0 | CAN Input Filter 6 | \rightarrow | None | 00 | Always Ok | | | | | |
| . | | High | \mathbf{Q}_0^0 | CAN Input Filter 7 | \rightarrow | None | 00 | Always Ok | | | | | |
| - | | High | Q^0_0 | CAN Input Filter 8 | \rightarrow | None | 00 | Always Ok | | | | | |
| | | High | Q^0_0 | CAN Input Filter 9 | \rightarrow | None | 00 | Always Ok | | | | | |
| | | High | Q^0_0 | CAN Input Filter 10 | \rightarrow | None | 00 | Always Ok | | | | | |
| | | High | Q^0_0 | CAN Input Filter 11 | \rightarrow | None | 00 | Always Ok | | | | | |
| | | High | 00 | CAN unwrapper 0 | \rightarrow | None | Q0 | Always Ok | | | | | |
| | | High | Q_0^0 | CAN unwrapper 1 | \rightarrow | None | Q0 | Always Ok | | | | | |
| | | High | \mathbf{Q}_0^0 | Serial to CAN 0 | \rightarrow | None | Q ₀ | Always Ok | | | | | |
| | | 📄 High | Q^0_0 | Serial to CAN 1 | \rightarrow | None | Q0 | AL | | | | | |
| | | 🗌 High | \mathbf{Q}_{0}^{0} | Serial to CAN 2 | \rightarrow | CAN Output Filter 3 | Q ₀ ^o (| Port | | | | | |

Fig. 45: CEX PDI Builder - CAN I/O configuration



Fig. 46: CEX PDI Builder - CAN Input Filter configuration



Fig. 47: CEX PDI Builder - Serial to CAN configuration

3. CAN Setup (mailboxes) configuration

Mailboxes need to be defined for the reception of CAN messages. In the example above, mailboxes for ID 51 need to be added on CAN B port.

CEX PDI Builder		- ×
CEX 2.0 with ARINC -	• <a>finput / Output	* • • •
GPIO DMAA	CANA CANB	
I/O Setup	Units ID DEC Units Mask BIN Baudrate 500000.0	
CAN I/O	Mailboxes reserved RX: 8 Mailboxes available TX: 24	
Digital Input	# Mailboxes Extended ID Mask	+
© Serial	1 — 4 — 1302 111111111	
CAN Setup	2 — 4 51 111111111	
—		
-		

Fig. 48: CEX PDI Builder - CAN Setup configuration

3.6.2 1x PDI Builder side

4. I/O Setup configuration

On the I/O Setup panel, link an 'RS custom message' to a 'Serial to CAN' with the serial data that the Autopilot 1x is going to send to CEX.

Then, link a 'CAN to Serial' to another 'RS custom message' with the expected serial messages that the CEX will receive in the selected serial port.

T		🕰 Input / Output			8 2 0	
I/O Setup	Configuration	Producer		Consumer		
Digital Input	High High	RS custom message 2	\rightarrow	Serial to CAN 0	og]	Always (
Serial	High 🕸	Tunnel 0		None	08	Always (
	High 🕵	Tunnel 1	\rightarrow	None	08	Always (
	High 🔯	Tunnel 2		None	OC C	Always (
	High 🔯	GPS 0 RTCM	\rightarrow	None	Q ^o	Always C
	High 🔯	GPS 1 RTCM	\rightarrow	None	Q0	Always (
	High 🔯	External HMR2300 magnetometer		None	Q0	Always (
	High 🔯	Y0 splitter A	\rightarrow	None	Q ₀	Always C
	High 🔯	Y0 splitter B	\rightarrow	None	Q0	Always (
	High 🔯	Y1 splitter A		None	Q0	Always (
	High 🔯	Y1 splitter B		None	Q ₀ ⁰	Always C
	High 🔯	Y2 splitter A	\rightarrow	None	Q ₀ ^o	Always C
	High 🔯	Y2 splitter B	\rightarrow	None	Q0	Always C
	High 🔯	Iridium	\rightarrow	None	Q0	Always C
	High 00	Unescape port	\rightarrow	None	30	Always (

Fig. 49: 1x PDI Builder - I/O Setup configuration

5. CAN Setup configuration

As for the CAN I/O, the same IDs employed in CEX for the CAN Input and CAN Output Filters are going to be employed on Veronte Autopilot 1x side, but they need to be inverted.

Therefore, the **Input filter** linked to the chosen 'CAN to Serial' needs to have **ID 50**. And the **Output filter** linked to the chosen 'Serial to CAN' will have **ID 51**.

erontePDI Builder						
.8 -		•😋 Input / Output		(8 1 0	
I/O Setup	Configuration Cust	om message 0 Custom message 1	Custom message 2	Mailboxes		
CAN Setup	Priority	Producer		Consumer	544	
Digital Input	High 🕵	Serial to CAN 0		Output filter 2	Q ₂	Always Ok
Serial	High 👯	Serial to CAN 1	\rightarrow	None	\$\$	Always Ok
1	High 🕰	Serial to CAN 2	\rightarrow	None	Q0	Always Ok
	High 🛱	Serial to CAN 3	\rightarrow	None	Q0	Always Ok
	High 🛱	Serial to CAN 4	\rightarrow	None		Always Ok
	High 🛱	Serial to CAN 5	\rightarrow	None		Always Ok
	High 🕸	CAN custom message 0	\rightarrow	None	\$\$\$	Always Ok
	High 🕸	CAN custom message 1	\rightarrow	None	Q0	Always Ok
	High 😳	CAN custom message 2	\rightarrow	Output filter 3	¢₀°	Always Ok
	V High	Input filter 0	→	CAN to serial 0	\$\$	Always Ok
	High 🕵	Input filter 1	\rightarrow	None	Q ₀ ⁰	Always Ok
	High 🛱	Input filter 2	\rightarrow	None	Q0	Always Ok
	High 🛱	Input filter 3	\rightarrow	None	Q0	Always Ok
	High 🛱	Input filter 4	\rightarrow	None	Q0	Always Ok
	High 🛱	Input filter 5	\rightarrow	None	Q0	Always Ok
	High 🔯	CAN unwrapper 0	\rightarrow	None	Q0	Always Ok
	High (198)	CAN unurannar 1	\rightarrow	None		Alwaye Ok

Fig. 50: 1x PDI Builder - CAN Setup configuration



Fig. 51: 1x PDI Builder - Serial to CAN configuration



Fig. 52: 1x PDI Builder - Input filter configuration

6. Mailboxes configuration

Some mailboxes with ID 50 will have to be created on whichever chosen reception CAN bus.

1xVeron	tePDI Builder		- ×
1x 4.8	¥	• 🛟 Input / Output	0
Ö	I/O Setup	Configuration Custom message 0 Custom message 1 Custom message 2 Mailboxes	
	CAN Setup	CAN A CAN B	
<u>س</u>	Serial	Units ID DEC Units Mask BIN Baudrate 1000000.0 - Enable Terminator	
•		Mailboxes reserved RX: 8 Mailboxes available TX: 24	
		# Mailboxes Extended ID Mask 1 -4 1301 1111111111	
¢ °		2 — 4 50 111111111	
P			
50			
$\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ $			
æ			
X			

Fig. 53: 1x PDI Builder - Mailboxes configuration

3.7 External devices

The step-by-step instructions for the following external devices are explained in detail in the following sections:

- Analog devices
- Jetibox
- Scorpion tribunus
- Veronte products

3.7.1 Analog devices

CEX is configured to read voltages from *ANALOG 0-7* inputs (**pins 43-50**), automatically storing the received values in **Real variables 1320-1327**.

Note: The correspondence of analog pins and these variables can be consulted in the Real Variables list of **CEX Software Manual**.

For reading information from an analog device, make the following connections:

- 1. Connect the analog output of the device to one of the designated pins of CEX (pins 43-50).
- 2. Connect the GND of the device to the ground for analog signals of CEX (pin 42).

Note: For further details on the pins of CEX, consult the Pinout - Hardware Installation section of the CEX Hardware Manual.

3.7.2 Jetibox

To enable CEX to simulate a Jetibox to read telemetry from legacy Jeti devices, the following steps need to be taken:

1. Go to Devices menu \rightarrow **Jetibox panel**.

Enable it and select a **SCI Port**, in this example SCI A is selected:

CEX 2.0 with ARINC -	0
Itable Scorpion tribunus SCI Port SCI Port <td< th=""><th></th></td<>	

Fig. 54: Jetibox - Devices configuration

2. Go to Input/Output menu \rightarrow Serial panel.

Configure the serial port (the one selected in step 1) as follows:

- Baudrate: 9800
- Length: 8
- Stop: 2 stop bits
- Parity: Odd
- Use address mode: Enabled

CEXPDI B	uilder - CEX2 45109 - CONNECTE	Đ	— ×
CEX 2.0 w	vith ARINC 👻	•😋 Input / Output	8 1 2 5 0
\Box	GPIO	SCLA SCLB SCLC	
	PWM		
2	I/O Setup	Functionality	
	CAN I/O		
<u> </u>	Digital Input	Baudrate 9800 -	
0	Serial	Length 8 👻	
Ľ	CAN Setup	Stop 2 👻	
62 0		Parity Odd -	
		✓ Use address mode	
-			

Fig. 55: Jetibox - Serial configuration

Note: The serial port will be completely reserved for this, so it cannot be used for any other purpose and the I/O Setup configuration affecting it will be ignored.

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

Link the specific **JETI box** consumer to the **serial port** configured in **step 2**:

0 with ARINC 🚽			🕂 Input / Output	:		± 0			
GPIO	Configurat	tion							
PWM	Priority		Producer		Consumer				
I/O Setup	High	Q6	RS232-A	\rightarrow	JETI box	$\mathbf{Q}_{\mathbf{Q}}^{\mathbf{Q}}$	Always Ok		
CAN I/O	High	Q6	RS232-B	\rightarrow	None	Q ⁰	Always Ok		
Digital Input	High	Q ⁰	RS485-C	\rightarrow	None	Q0	Always Ok		
CAN Setup	High	Q ⁰	CAN to Serial 0	\rightarrow	None	Q0	Always Ok		
	High	Q0	CAN to Serial 1	\rightarrow	None	Q\$	Always Ok		
	High	00	CAN to Serial 2	\rightarrow	None	Q0	Always Ok		
	High	Q00	CAN to Serial 3	\rightarrow	None	Q0	Always Ok		
	High	Q0	CAN to Serial 4	\rightarrow	None	\dot{Q}^{0}_{0}	Always Ok		
	High	Q0	CAN to Serial 5	\rightarrow	None	Q0	Always Ok		
	High	Q0	Commgr port 0	\rightarrow	Serial to CAN 0	Q0	Always Ok		
	High	\$¢	Commgr port 1	\rightarrow	RS232-A	Q ₀ ^o	Always Ok		
	High	Q0	Commgr port 2	\rightarrow	RS232-B	Q0	Always Ok		
	High	Ф <mark>0</mark>	Commgr port 3	\rightarrow	None	Q0	Always Ok		
	High	00	Commgr port 4	\rightarrow	None	Ø0	Always Ok		
	High	Ф <mark>0</mark>	Commgr port 5	\rightarrow	None	Ø0	Always Ok		
	High	Q0	Tunnel 0	\rightarrow	None	Q ₀ ^o	Always Ok		
	High	00	Tunnel 1		None	36	Always Ok		

Fig. 56: Jetibox - I/O Setup configuration

4. Configure the Jetibox IO consumer to retrieve the data. Click on the **configuration button**

CEX PDI Builder	- × • 🕁 Input / Output	
GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Producer Consumer Priority Producer Consumer High 0° RS232-A JETI box 0° High 0° RS232-B Commar port 2 0° Always Ok High 0° RS485-C → Commar port 3 0° Always Ok [Consumer] Consumer JETI box Command Checksum Matcher Skip Variable ASCII Position Occupancy	- ×
	Memory usage: 0 / 1000 Fields	

Fig. 57: Jetibox - JETI box Consumer configuration

Below is an example of the custom messages needed to read the Actual Voltage from a Jeti MasterSpin 220 (use **Big** endian in all messages):

[Consumer] Consumer JE	TI box									
		Command		Checksum	Matcher	Skin	Variable	ASCIL	Position	Occupancy
	ndian	Jeti Box Down		0	2	8	0	0	0	
		Command		Checksum	Matcher	Skin	Variable	ASCIL	Position	Occupancy
	ndian	Jeti Box Down	•	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 🗐 🗍 🛛 Big e	ndian	Jeti Box Down	-	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 📑 🗍 🛛 Big e	ndian	Jeti Box Down	*	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 📑 🗍 🛛 Big e	ndian	Jeti Box Down	*	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
🗕 📑 🗍 🛛 Big e	ndian	Jeti Box Down	*	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 📑 🔲 🛛 Big e	ndian	Jeti Box Down	٠	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 📑 🔲 🛛 Big e	ndian	Jeti Box Down	۳	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 📑 🗍 🛛 Big e	ndian	Jeti Box Down	۰	0	1	0	0	0	0	
		Command		Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy
— 📑 🗍 🛛 Big e	ndian	Jeti Box Nop	۳	0	1	8	0	1	0	
				Me	emory usa	ige: 0 /	16384 by	tes		

Fig. 58: JETI box example

- 1. Expected text: "CONTROLLER TYPE MasterSpin 220~"
 - Command: Jeti Box Down
 - Matcher(32) "CONT" 0x434F4E54 (1129270868)
 - Skip(24*8) 192
 - Matcher(32) "220~" 0x3232307E (842150014)

[(Consumer] Consu	mer JETI box							000000000				- ×
			Com	mand	Checksum	Matcher	Skip	Variable	ASCII	Position	Occupancy		î +
		Big endian	Jeti Box D	own 🔻	0	2	8	0	0	0			
	- 🗊 🗍	Hatch	ner x434F4E54	•								^ +	
	Value	Bits		Mask									
	1129270868	32	4294967295		dec								
	- 📑 🗍 Bits	• 🕂 Skip 2	14										
	24												
	- 🗊 🗇	🕂 Skip 2	14										
	- 🗊 🗇) 💠 Skip 2	14										
	- 🗊 🗇) 💠 Skip 2	14										
	- 🗊 🗇) 💠 Skip 2	.4										
	- 🗊 🗇) 💠 Skip 2	14										
	- 🗊 🗇) 💠 Skip 2	14										
	- 💷 🗍) 🕂 Skip 2	14										
	- 🗊 🗇	Hatch	ner x3232307E										
	Value	Bits		Mask									
	842150014	32	4294967295		dec							~	~
					Me	mory usa	ge: 256	/ 16384 b	ytes				

Fig. 59: JETI box custom message example

- 2. Expected text: "MeasureOrSetting MEASURE ~"
 - Command: Jeti Box Down
 - Matcher(32) "Meas" 0x4D656173 (1298489715)
- 3. Expected text: "Max Temperature"...
 - Command: Jeti Box Down
 - Matcher(32) "Max" 0x4D617820 (1298233376)
- 4. Expected text: "Min Temperature"...
 - Command: Jeti Box Down
 - Matcher(32) "Min" 0x4D696E20 (1298755104)
- 5. Expected text: "Actual Temperatu"...
 - Command: Jeti Box Down
 - Matcher(32) "Actu" 0x41637475 (1097036917)
- 6. Expected text: "MaxCurrent"...
 - Command: Jeti Box Down

- Matcher(32) "MaxC" 0x4D617843 (1298233411)
- 7. Expected text: "MinCurrent"...
 - Command: Jeti Box Down
 - Matcher(32) "MinC" 0x4D696E43 (1298755139)
- 8. Expected text: "Max Voltage"...
 - Command: Jeti Box Down
 - Matcher(32) "Max" 0x4D617820 (1298233376)
- 9. Expected text: "Min Voltage"...
 - Command: Jeti Box Down
 - Matcher(32) "Min" 0x4D696E20 (1298755104)
- 10. Expected text: "Actual Voltage 11,86 V "
 - Command: Jeti Box Nop
 - Matcher(32) "Actu" 0x41637475 (1097036917)
 - Skip(12*8) 96
 - Parse ascii: int(2), decimal(2), separartor(`,')

[Consumer] Consumer JETI box			- ×
💳 📑 🔲 Big endian Jeti Box Down	• 0 1 0	0 0 0	^ 🕂
	Checksum Matcher Skip	Variable ASCII Position Occupancy	
Big endian Jeti Box Nop	• 0 1 8	0 1 0	
— 📑 🗍 💠 Matcher x41637475			î +
Value Bits Mask			
1097036917 32 4294967295	dec		
- 💷 🗍 🕂 Skip 12			
Bits			
12			
— 📑 🗍 🕂 Skip 12			
— □ ↓ skip 12			
□ □ ➡ Skip 12			
— 📑 🗍 🕂 Skip 12			
— 🗐 🕂 Skip 12			
- 📑 🗇 🕂 Parse ASCII			
Variable Char in inter part	Char in decimal part Division cha	r	
User Variabl 2	2 ,		U U
Received and the second s	Memory usage: 0	16384 hytes	→ ×
	memory usage. 07	10504 Dytes	

Fig. 60: JETI box last custom message example

3.7.3 Scorpion tribunus

CEX is able to read telemetry from Tribunus ESCs by connecting it to one of its serial ports. The following steps are necessary to configure it:

1. Go to Devices menu \rightarrow Scorpion tribunus panel.

Enable it and configure it as shown:

CEXPDI Builder - CEX2 45109 - CONNECTE	D		- ×
CEX 2.0 with ARINC 🔫	O De	vices	8 4 6 6
Jetibox	Enable		
Scorpion tribunus	Telemetry period	0.02	
	MCU Telemetry 1 period	0.05	
	MCU Telemetry 2 period	0.05	
	PWM ID	0	
5 0			
			

Fig. 61: Scorpion tribunus - Devices configuration

2. Go to Input/Output menu \rightarrow I/O Setup panel.

Link the specific Tribunus ESC consumer to the desired port:

2.0 with ARINC -		• ← Input / Outpu	t	8	± 0	
GPIO	Configuration					
PWM	Priority	Producer		C	onsumer	
I/O Setup	High 🕸	RS232-A	\rightarrow	Commgr port 1	Q0	Always Ok
CAN I/O	High 🔯	RS232-B	\rightarrow	Commgr port 2	00	Always Ok
Digital Input Serial	High 😂	RS485-C	\rightarrow	Tribunus ESC	00	Always Ok
CAN Setup	High 🕸	CAN to Serial 0	\rightarrow	Commgr port 0	Q ₀ ⁰	Always Ok
	High 🔯	CAN to Serial 1	\rightarrow	None	Q ₀	Always Ok
	High 🔯	CAN to Serial 2	\rightarrow	None	Q ₀	Always Ok
	High 🔯	CAN to Serial 3	\rightarrow	None	Q0	Always Ok
	High 🔯	CAN to Serial 4	\rightarrow	None	Q ₀ ⁰	Always Ok
	High 🔯	CAN to Serial 5	\rightarrow	None	Q ⁰	Always Ok
	High 🔯	Commgr port 0	\rightarrow	Serial to CAN 0	Q ₀ ^o	Always Ok
	High 🔯	Commgr port 1	\rightarrow	RS232-A	Q0	Always Ok
	High 🔯	Commgr port 2	\rightarrow	RS232-B	Q ₀	Always Ok
	High 🔯	Commgr port 3	\rightarrow	RS485-C	Q0	Always Ok
	High 🔯	Commgr port 4	\rightarrow	None	Q ₀	Always Ok
	High 🔯	Commgr port 5	\rightarrow	None	¢¢,	Always Ok
	High 🔯	Tunnel 0	\rightarrow	None	Q ₀ ^o	Always Ok
	High M	Tunnel 1	\rightarrow	None	100	Always Ok

Fig. 62: Scorpion tribunus - I/O Setup configuration

3.7.4 Veronte products

3.7.4.1 Connection with Autopilot 1x via CAN

No configuration is necessary in CEX, the configuration for communication with Autoilot 1x is set by default.

The configuration required in **1x PDI Builder** to communicate with **CEX** via CAN is explained in the CEX/MEX - Integration examples section of the **1x PDI Builder** user manual.

3.7.4.2 Connection with Autopilot 4x via CAN

3.7.4.2.1 CAN Reception IDs

First, users can setup the receiving CAN Ids for each one of the 3 possible **Veronte Autopilots 1x** sending data to CEX in the **Arbitration menu**.

Go to Arbitration menu \rightarrow CAN Setup panel:

CEX PD	l Builder		- ×
CEX 2.0	with ARINC -	Arbitration	7 7 2 9
Ö	Arbitration Config CAN Setup		
2	Send status Period 0.2		
•	Send score Period 3.0		
	Status message ID 255 Extended		
Ŷ	Autopilot 0		
53	Extended		
\Box			
ĕ			
m	ID 9		
	Autopilot 2		
	Extended		
	ID 10		

Fig. 63: Arbitration - CAN Reception IDs

Note: If arbitration is not enabled, and therefore only one Autopilot 1x is being used, no CAN Ids need to be configured here.

3.7.4.2.2 CAN I/O Interconnections

Once the CAN IDs are set, users shall configure:

- CAN Input Filters to be used (as communication with CEX has to be always through a Filter).
- The connection between input filters and data Consumers.

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

CEX PDI	Builder		•€• Input / Output			± 0	- × - 0 = 0
Ö	GPIO	Configuration C	AN Telemetry 0 CAN Telemet	try 1 CAN	Telemetry 2		
2	I/O Setup	High 🔍	Application Processor		CAN Output Filter 1	¢₿	Always Ok
	CAN I/O	High 🕵	CAN Input Filter 0		CAN to Serial 0	CCC	Always Ok
	Digital Input	High 📽	CAN Input Filter 1		Application Processor	QC I	Always Ok
P	CAN Setup	High	CAN Input Filter 2		None	00	Always Ok
3		High	CAN Input Filter 3		None	Q ₀ ^o	Always Ok
		High 🗘	CAN Input Filter 4	\rightarrow	None	Q ₀ ^o	Always Ok
		High 🕵	CAN Input Filter 5	\rightarrow	None	Q0	Always Ok
÷.		High 🗘	CAN Input Filter 6	\rightarrow	None	Q_0^0	Always Ok
		High 🗘	CAN Input Filter 7	\rightarrow	None		Always Ok
		High 🗘	CAN Input Filter 8	\rightarrow	None	Q ₀	Always Ok
		High 🔯	CAN Input Filter 9	\rightarrow	None		Always Ok
		High 🗘	CAN Input Filter 10	\rightarrow	None	Q ₀	Always Ok
		High 🗘	CAN Input Filter 11	\rightarrow	None	Q0	Always Ok
		High 🕸	CAN unwrapper 0	\rightarrow	None		Always Ok
		High 🕸	CAN unwrapper 1	\rightarrow	None	Q0	Always Ok
		High 🔯	Serial to CAN 0		CAN Output Filter 0	Ø,	Always Ok
		High 62	Serial to CAN 1		None	100	Alwave Ok

Fig. 64: Arbitration - CAN Input Filters

For more information on CAN I/O configuration, see the CAN I/O - Input/Output section of this manual.

Next step is to connect each of the desired data Producers to a **CAN Output Filter**, and configure both the Producer and the CAN Output Filter:

		• - •				
GPIO	Configuration CA	AN Telemetry 0 CAN Telemetry 1	CAN Telemetry 2			
I/O Setup	High 🔅	Application Processor	CAN Output Filter 2		Port	CAN B
CAN I/O	High 🕵	CAN Input Filter 0	→ None	402	AIWays UK	
Digital Input	High 🕵	CAN Input Filter 1	CAN to Serial 1	102	Always Ok	
Serial CAN Setup	High 🕰	CAN Input Filter 2	Application Processor	02	Always Ok	
o moctop	High 🕰	CAN Input Filter 3	→ None	100	Always Ok	
	High OC	CAN Input Filter 4	→ None		Always Ok	
	High OC	CAN Input Filter 5	→ None		Always Ok	
	High 🕵	CAN Input Filter 6	→ None	-02	Always Ok	
	High 🕵	CAN Input Filter 7	→ None	-02	Always Ok	
	High 🕵	CAN Input Filter 8	→ None	102	Always Ok	
	High OS	CAN Input Filter 9	→ None	102	Always Ok	
	High OS	CAN Input Filter 10	→ None	102	Always Ok	
	High OS	CAN Input Filter 11	→ None	02	Always Ok	
	High Ø	CAN unwrapper 0	→ None	402	Always Ok	
	High C	CAN unwrapper 1	→ None	-02	Always Ok	
	High Ø	Serial to CAN 0	→ None	408	Always Ok	
				1000		
Builder vith ARINC 👻	Hinh 🐯	Serial to CAN 1	CANI Output Filter 1	1 68 2	Alusur Ok —	×
Builder with ARINC	Configuration	Serial to CAN 1	CAN Output Eilter 1 CAN Telemetry 2	<u>48</u>	Alwave Ok	×
uilder ith ARINC GPIO PWM	Configuration Priority Priority	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor	CAN Output Filter 1 CAN Telemetry 2 CAN Output Filter 2 CAN OUtput	tate الم	Always UK	×
Auilder GPIO PWM I/O Setup CAN I/O	Configuration Priority High Q	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0	CAN Output Filter 1 CAN Telemetry 2 CAN Telemetry 2 CAN Output Filter 2 CAN Output Filter 2 None	tate tate	Always Ok Always Ok	×
uilder ith ARINC GPIO PWM I/O Setup CAN I/O Digital Input	Configuration C Priority Fign & High & High & High &	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1	CAN Output Filter 1 CAN Telemetry 2 CAN Telemetry 2 CAN Output Filter 2 O None CAN to Serial 1	sumer ₩s C	Always OK Always OK Always Ok Always Ok	×
titilder ith ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial Schulo Can	Configuration Priority Prign © High © High © High ©	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2	CAN Output Filter 1 CAN Telemetry 2 CAN Output Filter 2 CAN Output Filter 2 None CAN to Serial 1 Application Processor		Alwaye Ol-	×
ith ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority Figh & High & High & High & High & K High & K Configuration	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2 CAN Input Filter 3	CAN Cutnut Eilter 1 CAN Telemetry 2 CAN Output Filter 2 CAN OUtput Filter 2 CAN to Serial 1 Application Processor None None	88 ■ 48 ■	Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok	×
uilder ith ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High © High © High © High © High © High ©	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 AN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2 CAN Input Filter 3 CAN Input Filter 4	CAN Cutnut Filter 1 CAN Telemetry 2 CAN Telemetry 2 CAN Output Filter 2 CO None CAN to Serial 1 Application Processor None None None		Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok	×
GPIO PWM VO Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High © High © High © High © High © High © High © High © Configuration	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5	CAN Cutnut Filter 1 CAN Telemetry 2 CAN Telemetry 2 CAN Utput Filter 2 CAN to Serial 1 Application Processor None None None None None None None None	xe xe nsumer xes 0°; 0°; 0°; 0°; 0°; 0°; 0°; 0°;	Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok Always Ok	×
Builder vith ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High ©S High ©S High ©S High ©S High ©S High ©S High ©S	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 5 CAN Input Filter 6	CAN Cutruit Eilter 1 CAN Telemetry 2 CAN Telemetry 2 CAN to Serial 1 CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	×
uilder th ARINC - GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority rign %6 High ¢¢ High ¢¢ High ¢¢ High ¢¢ High ¢¢ High ¢¢ High ¢¢	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 6 CAN Input Filter 7	CAN Curtourt Eilter 1 CAN Telemetry 2 CAN Telemetry 2 CAN to Serial 1 CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	X
uilder ith ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration C Priority Fign & High & C	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 6 CAN Input Filter 7 CAN Input Filter 8	CAN Output Filter 1 CAN Telemetry 2 CAN Telemetry 2 CAN to Serial 1 CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	×
uilder th ARINC GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High & High &	Serial to CAN 1 CAN Input / Output AN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 2 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 5 CAN Input Filter 7 CAN Input Filter 8 CAN Input Filter 9	CAN Cutnut Eilter 1 CAN Cutnut Eilter 1 CAN Telemetry 2 CAN Output Filter 2 CO None CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	X
uilder th ARINC GPIO PWM VO Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High ©S High ©S	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 3 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 5 CAN Input Filter 7 CAN Input Filter 7 CAN Input Filter 8 CAN Input Filter 9 CAN Input Filter 10	CAN Cutnut Filter 1 CAN Cutnut Filter 1 CAN Telemetry 2 Co Can Output Filter 2 Co Can Output Filter 2 Co Can to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	×
uilder th ARINC - GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High ©S High ©S	Serial to CAN 1 CAN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 3 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 5 CAN Input Filter 7 CAN Input Filter 7 CAN Input Filter 7 CAN Input Filter 9 CAN Input Filter 10 CAN Input Filter 11	CAN Cutruit Eilter 1 CAN Cutruit Eilter 1 CAN Telemetry 2 CAN Output Filter 2 CAN to Serial 1 CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	×
silder th ARINC - GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority High & High &	Serial to CAN 1 Can Input / Output AN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 5 CAN Input Filter 7 CAN Input Filter 10 CAN Input Filter 11 CAN unwrapper 0	CAN Cutruit Eilter 1 CAN Cutruit Eilter 1 CAN Telemetry 2 Can Output Filter 2 None CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	X
uilder th ARINC - GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority rign %6 High %8 High %8	Serial to CAN 1 Can Input / Output AN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 7 CAN Input Filter 7 CAN Input Filter 9 CAN Input Filter 10 CAN Input Filter 11 CAN Input Filter 11 CAN unwrapper 0 CAN unwrapper 1	CAN Curtourt Eilter 1 CAN Curtourt Eilter 1 CAN Telemetry 2 CAN Output Filter 2 None CAN to Serial 1 Application Processor None None None None None None None None		Always Ok Always Ok	×
ilder th ARINC - GPIO PWM I/O Setup CAN I/O Digital Input Serial CAN Setup	Configuration Priority Fign & G High & C High &	Serial to CAN 1 Can Input / Output AN Telemetry 0 CAN Telemetry 1 Producer Application Processor CAN Input Filter 0 CAN Input Filter 1 CAN Input Filter 3 CAN Input Filter 4 CAN Input Filter 5 CAN Input Filter 5 CAN Input Filter 7 CAN Input Filter 7 CAN Input Filter 9 CAN Input Filter 9 CAN Input Filter 10 CAN Input Filter 11 CAN Input Filter 11 CAN unwrapper 0 CAN unwrapper 1 Serial to CAN 0	CAN Output Filter 1 CAN Telemetry 2 CAN Telemetry 2 CAN to Serial 1 CAN to Ser		Always Ok Always Ok	

Fig. 65: Arbitration - CAN Output Filters

CHAPTER

TROUBLESHOOTING

4.1 ARINC Communication not working

If ARINC communication does not work on the user's CEX unit, it may be due to an incorrect configuration of the **GPIO** *panel*. GPIOs configuration should be as shown below:

XPDI Builder - CEX2 45109 - CO	NNECTED					
X 2.0 with ARINC -		•🕁 In	put / Output		8 1	
	Signal	GPIOId	ю	Pull-up	Function	Qsel
PWM	PWM 3	GPIO 3	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
I/O Setup	PWM 4	GPIO 4	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
CAN I/O	PWM 5	GPIO 5	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
Digital Input	PWM 6	GPIO 6	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
Serial	PWM 7	GPIO 7	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
CAN Setup	I/O 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
	I/O 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
	I/O 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
]	I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	1/0 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	1/0 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 9	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 10	GPIO 51	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 11	GPIO 52	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 12	GPIO 49	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 13	GPIO 8	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 14	GPIO 11	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 15	GPIO 10	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
	I/O 16	GPIO 9	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync

Fig. 1: ARINC Communication not working - GPIO panel correct

If users don't have this panel configured this way and they don't want to lose the CEX configuration they have already done, follow the next steps:

1. Open a configuration in **CEX PDI Builder** in **offline** mode by clicking on **CEX** option in the first window of the application:



Fig. 2: ARINC Communication not working - Open configuration offline

2. Change the **hardware version** to CEX 2.0 with GPIO to be able to change the GPIOs configuration of the GPIO panel.



Fig. 3: ARINC Communication not working - Change hardware version

3. Next, **import** the CEX configuration to be fixed.

To do this, click on the button and select the configuration from the local storage:



Fig. 4: ARINC Communication not working - Import configuration from local storage

4. Go to Input/Output menu \rightarrow GPIO panel.

The configuration of this panel should be as shown below:

CEX PDI	Builder	ananya any anany anany anany anany a					- ×
CEX 2.0	with GPIOs 🔹		•∕ <mark>t</mark> In	put / Output		*	9
Ö	GPIO	Signal	GPIOId	ю	Pull-up	Function	Qsel
	PWM	PWM 3	GPIO 3	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync ^
	I/O Setup	PWM 4	GPIO 4	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN I/O	PWM 5	GPIO 5	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
•	Digital Input	PWM 6	GPIO 6	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
	Serial	PWM 7	GPIO 7	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync
(CP)	CAN Setup	I/O 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
620		I/O 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
\circ		I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
-ff-		I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 9	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 10	GPIO 51	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 11	GPIO 52	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 12	GPIO 49	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 13	GPIO 8	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 14	GPIO 11	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 15	GPIO 10	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 16	GPIO 9	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync

Fig. 5: ARINC Communication not working - GPIO panel

5. Once the changes have been made, **export** this configuration by clicking on the **b**utton and store it in the same folder from which it was previously imported to overwrite it with the correct settings.

			sur, output							
GPIO	Signal	GPIOId	ю	Pull-up	Function	Qsel		Select folder		
PWM	PWM 3	GPIO 3	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync	^	← → × ↑ 🔤 « Deskto	p > configurations	Search configurations
I/O Setup	PWM 4	GPIO 4	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync				
CAN I/O	PWM 5	GPIO 5	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync		Organize 👻 New folder		(F
Digital Input	PWM	GPIO 6	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync			No. of Concession, Name	
Serial	PWM 7	GPIO 7	GPIO as output	Pull-up disabled	Mux 0 / GPIO	Sync		🖈 Quick access		
CAN Setup	/0 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync		💻 This PC		
	1/0 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync		3D Objects		
	1/0 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync		Desktop	cex	
	I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Documents	configuration	
	I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Downloads		
	I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Music		
	I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Pictures		
	1/0 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Videos		
	1/0 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Local Disk (C:)		
	109	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync				
	1/0 10	GPIO 51	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync				
	1/0 11	GPIO 52	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync				
	I/O 12	GPIO 49	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync				
	I/O 13	GPIO 8	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Folder	ex configuration	
	I/O 14	GPIO 11	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync		Tolder.	ci comgututon	
	110.10									Select Folder

Fig. 6: ARINC Communication not working - Export configuration

6. Finally, **upload** the newly exported configuration to the **CEX** unit and check that the **GPIO panel** is correctly configured.

CEXPDI	Builder - CEX2 45109 - COI	NNECTED	-				
	with ARINC 👻		•∕ᢏ• In	put / Output			9 2 6
O	GPIO	Signal	GPIOId	ю	Pull-up	Function	Qsel
	PWM	PWM 3	GPIO 3	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
ッ	I/O Setup	PWM 4	GPIO 4	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
	CAN I/O	PWM 5	GPIO 5	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
•47	Digital Input	PWM 6	GPIO 6	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
6	Serial	PWM 7	GPIO 7	GPIO as input	Pull-up disabled	Mux 0 / GPIO	Sync
¥2	CAN Setup	I/O 0	GPIO 28	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
60		I/O 1	GPIO 61	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
		I/O 2	GPIO 60	GPIO as output	Pull-up disabled	Mux 0 / GPIO	ASync
0		I/O 3	GPIO 59	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 4	GPIO 17	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
-		I/O 5	GPIO 58	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 6	GPIO 16	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 7	GPIO 53	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		1/0 8	GPIO 20	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 9	GPIO 23	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 10	GPIO 51	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 11	GPIO 52	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 12	GPIO 49	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 13	GPIO 8	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 14	GPIO 11	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 15	GPIO 10	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync
		I/O 16	GPIO 9	GPIO as input	Pull-up enabled	Mux 0 / GPIO	Sync

Fig. 7: ARINC Communication not working - GPIO panel correctly configured

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	$\mathbf{\times}$
Error when switching to normal mode	, check the configuration.

Fig. 8: Error message

Therefore, CEX will be in 'Maintenance mode (loaded with errors)':



Fig. 9: 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:

PDI Errors - Device: 45109 — 🛪							
PDI ID	PDI Error Description	Config ID	Config Desc.				
502	Invalid CAN id	315	CEX CAN IN Filters				
			▲ Export Close				

Fig. 10: Maintenance mode (loaded with errors) - PDI Errors panel

- **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 **1x 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.

Finally, it is possible to access the CEX configuration to fix this error.

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