Features:

CAN2.0A supported

12V or 24V power supply supported

Electronic overcurrent protection and automotive mini blade fuse

22 switched circuits

Up to 4 low side outputs

Operating temperature range: -20 to +85°C

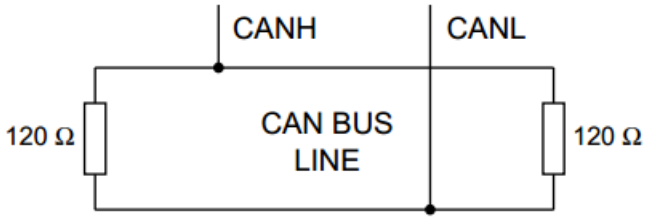
Splash proof (IP65)

Deutsch DRC10-40P-A-004 40pin connector

Contents

1. How to Connect CAN bus:

J5 PIN	FUNCTION
13	CAN L
22	CAN H



Each end of the CAN bus is terminated with 120Ω resistors in compliance with the standard to minimize signal reflections on the bus. You may need to place a 120Ω terminating resistor between CAN-L and CAN-H

NOTE: the PC21B PCB is set up so that it is possible to require the terminating resistor to be built in.

2. PC21B Technical Specifications

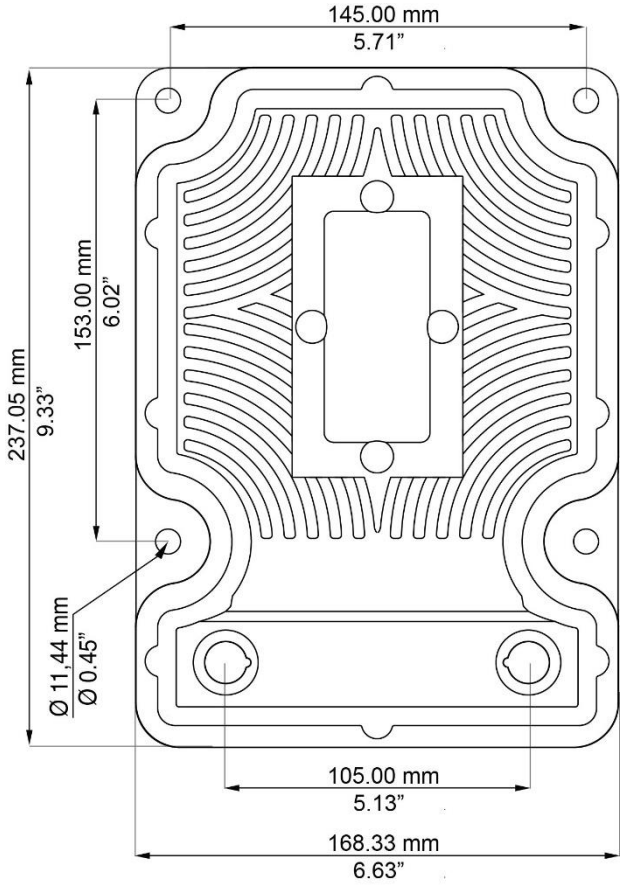
Electrical	**Value**	**Unit**
Supply Voltage (Battery Voltage)	9-16 (12V) or 18-31 (24V)	V
Maximum input current	150	A
Maximum current single pin	15	A
Maximum Digital input voltage	Battery Voltage	V
Digital input low voltage max	5,5	V
Environmental	**Value**	**Unit**
Storage temperature range	-40 to +85	°C
Operating temperature range	-20 to +85	°C
Humidity	0 to 98	%

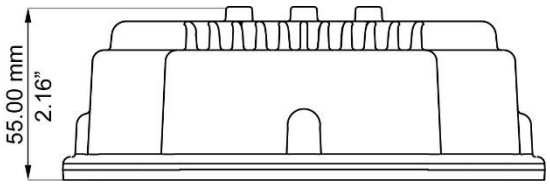
3. Mechanical features

- Housing: aluminium AB46100 anodic oxidation + powder-coating + zinc-coated bronze ZnCu20
- Seal: EPDM GASKET
- Power studs: Blink Marine CuZn20 UNC#14-20

Mechanical dimensions

Dimensions are in millimeters.





Warning: to avoid breakage do not tighten the housing nuts with a torque exceeding 3.5 Nm!

4. Connector 40 pin:

The 40-pin automotive connector DRC10-40P-A004 mates with housing for female terminals DRC16-40SA.

Output	PIN
Ch1	1
Ch2	2
Ch3	3
Ch4	4
Ch5	5
Ch6	6
Ch7	7
Ch8	8
Ch9	9
Ch10	10
Ch11	11
TXRX+	12
CANL	13
TXRX+	14
GND	15
Ch16	16
CH16	17
GND	18
GND	19
Ch20	20
Ch21	21
CANH	22
TXRX-	23
TXRX-	24
GND	25
GND	26
GND	27
GND	28
GND	29
Ch30	30
Ch31	31
Ch32	32
Ch33	33
GND	34
GND	35
GND	36
Ch37	37
Ch38	38
Ch39	39
Ch40	40

5. Electrical Loads Connection

HIGH side switch:

The electrical device is connected between an output pin of the PowerCore and ground.

The output pin state can be HIGH (pin voltage at a battery voltage) or OFF (pin floating).

The ground can be connected either to one of the ground pins of the PowerCore or directly to the battery's negative pole.

The electrical device is ON when the pin state is HIGH and the current consumption of the electrical device is read by the PowerCore.

LOW side switch:

The electrical device is connected between an output pin of the PowerCore with Low Side Drive feature and the positive battery pole.

The output pin state can be LOW (pin voltage at ground) or OFF (pin floating).

The current flowing in the low side switch is not read by the PowerCore. A 25A fuse protects the circuit, but a lower current threshold is needed, the connection to the battery's positive pole should be made through a fuse.

The electrical device is ON when the pin state is LOW.

H-Bridge connection:

The electrical device is connected between two output pins of the PowerCore.

The state of each output pin can be HIGH (pin voltage at battery voltage), LOW (pin voltage at ground) or OFF (pin floating).

This connection is common for DC motor to invert the power supply polarity to change the direction of the motor's rotation.

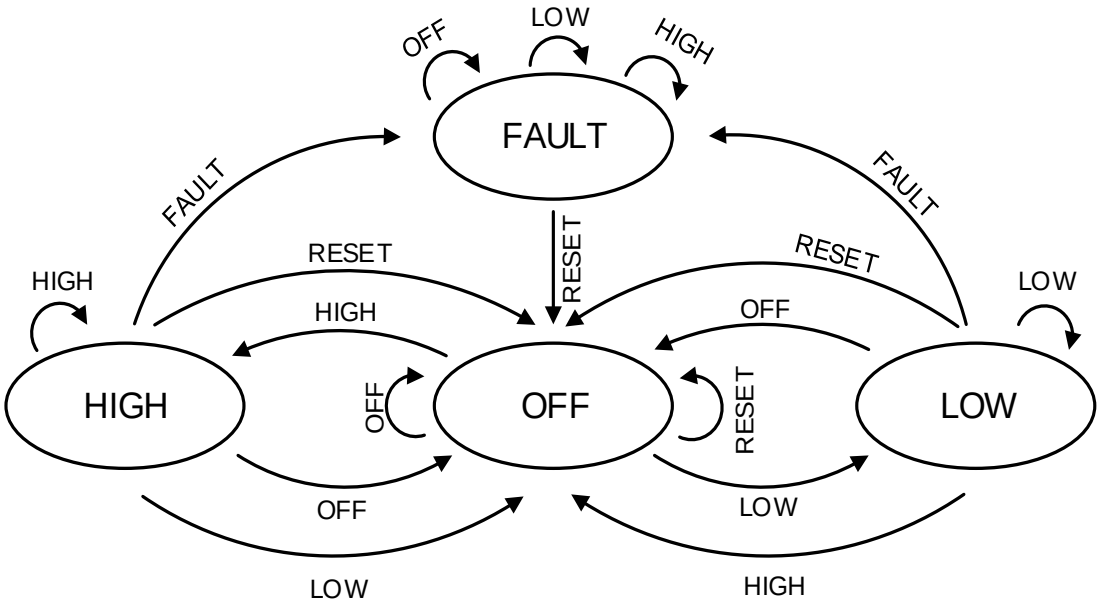
6. Pin Assignment

Connector	Pin	Function	Rating	Low Side Driver	Level Sense	Tach Sense	Optional Diode Protection
J5	1	O	15A				X
J5	2	O	15A				X
J5	3	O	15A				X
J5	4	O	15A				X
J5	5	O	15A				X
J5	6	O	15A				X
J5	7	O	15A	X			X
J5	8	O	15A	X			X
J5	9	O	15A	X			X
J5	10	O	15A	X			X
J5	11	O	15A				X
J5	12	TXRX+	-				
J5	13	CANL	-				
J5	14	TXRX+	-				
J5	15	GND	-				
J5	16	O	15A				X
J5	17	O	15A				X
J5	18	GND	-				
J5	19	GND	-				
J5	20	O	15A				X
J5	21	O	15A				X
J5	22	CANH	-				
J5	23	TXRX-	-				
J5	24	TXRX-	-				
J5	25	GND	-				
J5	26	GND	-				
J5	27	GND	-				
J5	28	GND	-				
J5	29	GND	-				
J5	30	O	15A				X
J5	31	O	15A				X
J5	32	O	15A			X	X
J5	33	O	15A			X	X
J5	34	GND	-				
J5	35	GND	-				
J5	36	GND	-				
J5	37	I/O	15A		X		X
J5	38	I/O	15A		X		X
J5	39	I/O	15A		X		X
J5	40	I/O	15A		X		X

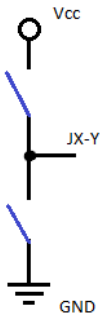
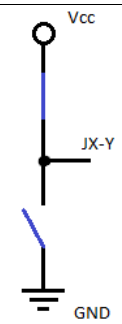
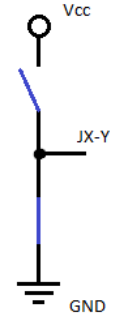
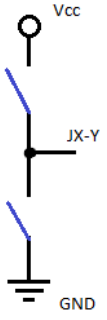
Function Summary:

- **O:** Output high side pin.
- **I/O:** Input / output high side pin.
- **GND:** Internally connected to Negative terminal of the batteries.
- **TXRX+, TXRX-:** RS485 bus signals.
- **CAN-L, CAN-H:** CAN bus signals.

7. Output State Machine Diagram



The output pin can be set to one of the 4 states available:

Output state	Pin voltage	Equivalent Circuit
OFF	Floating	
HIGH	Battery Voltage	
LOW	Ground Voltage	
FAULT	Floating	

To avoid cross conduction, the system does not allow to change the state of an output pin from HIGH to LOW and from LOW to HIGH. The pins must be always set to OFF for at least 300 milliseconds before the polarity can be changed.

As a measure of protection when using inductive loads, longer delays are added so a current peak during the transient period does not active the system's protection against overcurrent.

Each output circuit is protected against short circuits and overcurrent. If a failure event occurs, the pin goes into the FAULT state. To exit the FAULT state, a reset command is needed to return the pin to its initial OFF state.

The LOW state is applicable only on the output pins that a low side driver (refer to the pin assignment table).

8. CANopen Messages Structure

All the data type used are unsigned integer and the syntax is specified in the following table:

octet number	1.	2.	3.	4.	5.	6.	7.	8.
UNSIGNED8	b7..b0							
UNSIGNED16	b7..b0	b15..b8						
UNSIGNED24	b7..b0	b15..b8	b23..b16					
UNSIGNED32	b7..b0	b15..b8	b23..b16	b31..b24				
UNSIGNED40	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32			
UNSIGNED48	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40		
UNSIGNED56	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	
UNSIGNED64	b7..b0	b15..b8	b23..b16	b31..b24	b39..b32	b47..b40	b55..b48	b63..b56

NMT MESSAGES

The Network Management messages follow a master-slave structure. Through NMT services, CANopen devices are initialized, started, reset, or stopped. All CANopen devices are regarded as NMT slaves. NMT messages have CAN-ID always equal to 00h.

00h	1-byte command specifier	1-byte NODE-ID	6 bytes not used

PDO MESSAGES

PDO are fast telegram messages that can simply manage the most important functions. All PDOs have an equivalent SDO message. There are no answers for this type of messages. PDO messages have identifiers from 180h to 57Fh.

Identifier	8-byte data

SDO MESSAGES

SDO are more complex messages that completely manage all the functions of the PowerCore.

SDO messages have identifiers from 580h to 67Fh and always expect an answer or an acknowledge reply.

Identifier	Command byte	2-byte index	1-byte sub index	4-byte data

Identifier: The messages to the PowerCore shall have 600h+current CAN ID identifier.
The messages from the PowerCore have 580h+ current CAN ID identifier.

Command byte:

40h:	request to read a register	60h: write acknowledge
43h:	response with 4-byte data	23h: request to write 4-byte data
4Fh:	response with 1-byte data	2Fh: request to write 1-byte data
4Bh:	response with 2-byte data	2Bh: request to write 2-byte data
80h:	error response	

Every answer has index and sub index echo.

The error responses have the byte data containing the abort codes.

Abort codes implemented:

0602 0000h:	Object does not exist in the object dictionary
0609 0011h:	Sub-index does not exist
0609 0030h:	Invalid value for parameter
0601 0002h:	Attempt to write a read only object

9. CAN Messages for Managing Output Pins

Output state	Direction	Can message
HIGH	Write	RPDO 200h + node-ID
	Read/Write	SDO Object 2001h sub-index 01h
	Read/Write	SDO Object 2008h sub-index 02h
LOW	Write	RPDO 200h + node-ID
	Read/Write	SDO Object 2001h sub-index 02h
	Read/Write	SDO Object 2008h sub-index 03h
OFF	Write	RPDO 200h + node-ID
	Read/Write	SDO Object 2001h sub-index 00h
	Read/Write	SDO Object 2008h sub-index 01h
FAULT	Read when occur	TPDO 180h + node-ID
	Read	SDO Object 2001h sub-index 03h
	Read	SDO Object 2008h sub-index 04h
RESET	Write	SDO Object 2001h sub-index 04h
PWM	Write	SDO Object 2009h sub-index 01h

10. CAN bus Default Settings

Setting	Default Status or Level	How to Change
Baud rate	125 kbit/s	Object 2010h
CANopen Node ID	0Ch	Object 2011h
CANopen Node State	Pre-operational	NMT Message Start CANopen node Object 2013h Device active on startup
Periodic Message Transmission	Disabled	Object 2012h
PX output current threshold	10A	Object 2002h
Producer heartbeat time	Disabled	Object 1017h
Consumer heartbeat time	Disabled	Object 1016h
Boot-up service	Active	Object 2015h
Restore default parameters	-	Object 1011h

NMT MESSAGES

The Network Management Messages follow a master-slave structure. Through NMT services, CANopen devices are initialized, started, reset, or stopped. All CANopen devices are regarded as NMT slaves. NMT messages have CAN-ID always equal to 00h.

11. Start CANopen Node

Identifier	00h	
Byte 0	01h	Start CANopen node
Byte 1	XXh	PowerCore CAN ID 00h: start all the devices 0Ch: start the PowerCore with CAN ID = 0Ch.
Byte 2, 7	00h	Not used

Example:

Direction	Identifier	Format	Message
To PowerCore	0	Std	01 0C

12. Enter Pre-operational

Identifier	00h	
Byte 0	80h	Enter pre-operational
Byte 1	XXh	PowerCore CAN ID 00h: start all the devices 0Ch: start the PowerCore with CAN ID = 0Ch.
Byte 2, 7	00h	Not used

Example:

Direction	Identifier	Format	Message
To PowerCore	0	Std	80 0C

13. Reset CANopen Node

Identifier	00h	
Byte 0	81h	Reset CANopen node
Byte 1	XXh	PowerCore CAN ID 00h: start all the devices 0Ch: start the PowerCore with CAN ID = 0Ch.
Byte 2, 7	00h	Not used

Example:

Direction	Identifier	Format	Message
To PowerCore	0	Std	81 0C

14. Stop CANopen Node

Identifier	00h	
Byte 0	XXh	02h: Stop CANopen node
Byte 1	YYh	PowerCore CAN ID 00h: Stop all the devices 0Ch: Stop the PowerCore with CAN ID = 0Ch.
Byte 2, 7	00h	Not used

Example:

Direction	Identifier	Format	Message
To PowerCore	0	Std	02 0C

15. Boot-up Service

This service is used to signal that a NMT slave has entered the NMT Pre-operational state.

Identifier	700h + current CAN ID	Default 70Ch
Byte 0	00h	One data byte is transmitted with value 0.

Example:

Direction	Identifier	Format	Message
From PowerCore	70Ch	Std	00h

The PowerCore with CAN ID 0Ch has entered the NMT state Pre-operational.

16. Heartbeat Message

The heartbeat mechanism for a CANopen device is established by configuring the heartbeat producer to cyclically transmitting the heartbeat message. One or more CANopen devices in the network should be aware of this message. If by any chance the heartbeat producer fails to send the heartbeat message, the local application on the heartbeat consumer will be notified.

In the case the CANopen device starts with a heartbeat producer time different from 0, then the boot-up message is regarded as the first heartbeat message.

Identifier	700h + current CAN ID	Default 70Ch
Byte 0	XXh	XXh: State of heartbeat producer 00h: Boot-up 04h: Stopped 05h: Operational 7Fh: Pre-operational

Example:

Direction	Identifier	Format	Message	Data
From PowerCore	70Ch	Std	00h	Boot up
From PowerCore	70Ch	Std	7Fh	Pre-operational
To PowerCore	00h	Std	01h 0Ch	Start PowerCore with CAN id = 0Ch
From PowerCore	70Ch	Std	05h	Operational

PDO Messages

17. Set Output (OFF, HIGH and LOW)

Identifier	200h + current CAN ID	Default 20Ch
Byte 0	P8O P7O P6O P5O P4O P3O P2O P1O	'1'= set OFF '0'= do nothing
Byte 1	P31O P30O P21O P20O P16O P11O P10O P9O	'1'= set OFF '0'= do nothing
Byte 2	0 0 P40O P39O P38O P37O P33O P32O	'1'= set OFF '0'= do nothing
Byte 3	P8H P7H P6H P5H P4H P3H P2H P1H	'1'= set HIGH '0'= do nothing
Byte 4	P31H P30H P21H P20H P16H P11H P10H P9H	'1'= set HIGH '0'= do nothing
Byte 5	0 0 P40H P39H P38H P37H P33H P32H	'1'= set HIGH '0'= do nothing
Byte 6	P8L P7L 0 0 0 0 0 0	'1'= set LOW '0'= do nothing
Byte 7	0 0 0 0 0 0 P10L P9L	'1'= set LOW '0'= do nothing

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	20C	Std	81 00 00 00 00 00 00 00	Set PINs 8 and 1 OFF
To PowerCore	20C	Std	00 00 00 03 00 00 00 00	Set PINs 1 and 2 HIGH
To PowerCore	20C	Std	00 00 00 00 00 00 C0 03	Set PINs 7, 8, 9 and 10 LOW

If the same pin is set to more than one state, the priority is OFF, HIGH and LOW.
It is not possible to change the state of an output pin from HIGH to LOW and from LOW to HIGH.
The pins must be always set to OFF for at least 300 milliseconds before the polarity can be changed.

18. Fault Message

This message is sent by the PowerCore when an overcurrent is detected.

Identifier	**180h + current CAN ID**	**Default 18Ch**
Byte 0	P8 P7 P6 P5 P4 P3 P2 P1	'1' = fault '0'= not fault
Byte 1	P31 P30 P21 P20 P16 P11 P10 P9	'1' = fault '0'= not fault
Byte 2	0 0 P40 P39 P38 P37 P33 P32	'1' = fault '0'= not fault
Byte 3,7	00h	Not used

Examples

Direction	**Identifier**	**Format**	**Message**	**Data**
From PowerCore	18C	Std	18 00 00 00 00 00 00 00	PINs 4 and 5 fault state
From PowerCore	18C	Std	00 00 60 00 00 00 00 00	PINs 39 and 40 fault state

SDO messages:

19. Object 2001h: Single Pin State

This object sets and reads the state of each output pins.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Fh	Write 1-byte data
Byte 1	01h	CAN Object 2001h
Byte 2	20h	
Byte 3	00h	Highest sub-index supported (read only)
	01h	P1
	02h	P2
	03h	P3
	04h	P4
	05h	P5
	06h	P6
	07h	P7
	08h	P8
	09h	P9
	0Ah	P10
	0Bh	P11
	0Ch	P16
	0Dh	P20
	0Eh	P21
	0Fh	P30
	10h	P31
	11h	P32
	12h	P33
	13h	P37
	14h	P38
	15h	P39
	16h	P40
Byte 4	00h	OFF
	01h	HIGH
	02h	LOW (on supported pins)
	03h	FAULT (read only)
	04h	RESET
Byte 5,7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 01 20 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 01 20 00 16 00 00 00	16h is the highest sub-index
To PowerCore	60C	Std	40 01 20 04 00 00 00 00	Read state PIN 4
From PowerCore	58C	Std	4F 01 20 04 01 00 00 00	PIN 4 HIGH
To PowerCore	60C	Std	40 01 20 07 00 00 00 00	Read state PIN 7
From PowerCore	58C	Std	40 01 20 07 02 00 00 00	PIN 7 LOW
To PowerCore	60C	Std	2F 01 20 16 04 00 00 00	RESET PIN 40
From PowerCore	58C	Std	60 01 20 16 00 00 00 00	ACK
To PowerCore	60C	Std	2F 01 20 10 01 00 00 00	Set PIN 31 to HIGH
From PowerCore	58C	Std	60 01 20 10 00 00 00 00	ACK

20. Object 2002h: Output Threshold Current

This object sets and reads for each output pins the current thresholds. The maximum current of each pin is 15A. If a too high value is selected, a response error will be transmitted, and the value will be automatically set to the maximum. The default threshold is 10A.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Fh	Write 1-byte data
Byte 1	02h	CAN Object 2002h
Byte 2	20h	
Byte 3	00h	Highest sub-index supported (read only)
	01h	P1
	02h	P2
	03h	P3
	04h	P4
	05h	P5
	06h	P6
	07h	P7
	08h	P8
	09h	P9
	0Ah	P10
	0Bh	P11
	0Ch	P16
	0Dh	P20
	0Eh	P21
	0Fh	P30
	10h	P31
	11h	P32
	12h	P33
	13h	P37
	14h	P38
	15h	P39
	16h	P40
Byte 4	XXh	XXh: Threshold value in ampere (A)
Byte 5,7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 02 20 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 02 20 00 16 00 00 00	16h is the highest sub-index supported
To PowerCore	60C	Std	40 02 20 01 00 00 00 00	Read PIN 1 threshold current
From PowerCore	58C	Std	4F 02 20 01 0A 00 00 00	10A
To PowerCore	60C	Std	2F 02 20 0E 05 00 00 00	Set PIN 21 threshold current to 5A
From PowerCore	58C	Std	60 02 20 06 00 00 00 00	ACK
To PowerCore	60C	Std	2F 02 20 0C 06 00 00 00	Set PIN 16 threshold current to 6A
From PowerCore	58C	Std	60 02 20 08 00 00 00 00	ACK

21. Object 2003h: Read Digital Input

This object reads digital input values. The digital input is applicable only on some pins (refer to the pin assignment table). The unsupported pins have always the value '0'.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	03h	CAN Object 2003h
Byte 2	20h	
Byte 3	00h	Highest sub-index supported
	01h	Read input from PIN37 to PIN40
Byte 4,7	00h	Not used

From PowerCore:

Identifier	58Ch (580h + current CAN ID)	
Byte 0	4Fh	Response length 1-byte
Byte 1	03h	CAN Object 2003h
Byte 2	20h	

Byte 3	00h	sub-index
Byte 4	01h	Highest sub-index supported
Byte 5,7	00h	Not used

Byte 3	01h	sub-index
Byte 4	0 0 0 0 P40 P39 P38 P37	Digital input
Byte 5,7	00h	Not used

22. Object 2006h: Protection Trip Delay

This object sets and reads for each output pin the protection trip delay. It is possible to select a value between 100ms and 900ms.
The default value is 100ms.

Identifier	600h + current CAN ID	Default 60Ch	
Byte 0	40h	Read Device Register	
	2Fh	Write 1-byte data	
Byte 1	06h	CAN Object 2006h	
Byte 2	20h		
Byte 3	00h	Highest sub-index supported (read only)	
	01h	P1	
	02h	P2	
	03h	P3	
	04h	P4	
	05h	P5	
	06h	P6	
	07h	P7	
	08h	P8	
	09h	P9	
	0Ah	P10	
	0Bh	P11	
	0Ch	P16	
	0Dh	P20	
	0Eh	P21	
	0Fh	P30	
	10h	P31	
	11h	P32	
	12h	P33	
	13h	P37	
	14h	P38	
	15h	P39	
	16h	P40	
Byte 4	XXh	XXh: Protection Trip Delay (ms)	
		01h=100 (default) 02h=200 03h=300 04h=400 05h=500	06h=600 07h=700 08h=800 09h=900
Byte 5,7	00h	Not used	

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 06 20 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 06 20 00 16 00 00 00	16h is the highest sub-index supported
To PowerCore	60C	Std	2F 06 20 12 09 00 00 00	Set protection trip delay for PIN 33 to 900ms
From PowerCore	58C	Std	60 06 20 01 00 00 00 00	ACK

23. Object 2007h: Read PowerCore Electronic Values

This object reads the analog values of the output currents and the battery voltage.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	07h	CAN Object 2007h
Byte 2	20h	
Byte 3	00h	Highest sub-index supported (read only)
	01h	P1 current Amp
	02h	P2 current Amp
	03h	P3 current Amp
	04h	P4 current Amp
	05h	P5 current Amp
	06h	P6 current Amp
	07h	P7 current Amp
	08h	P8 current Amp
	09h	P9 current Amp
	0Ah	P10 current Amp
	0Bh	P11 current Amp
	0Ch	P16 current Amp
	0Dh	P20 current Amp
	0Eh	P21 current Amp
	0Fh	P30 current Amp
	10h	P31 current Amp
	11h	P32 current Amp
	12h	P33 current Amp
	13h	P37 current Amp
	14h	P38 current Amp
	15h	P39 current Amp
	16h	P40 current Amp
	17h	Battery voltage: 35,5V = FFh
Byte 4,7	00h	Not used

Answer:

Identifier	58Ch (580h + current CAN ID)	
Byte 0	4Fh	1-byte response
Byte 1	07h	CAN Object 2007h
Byte 2	20h	
Byte 3	XXh	XXh: Sub index echo
Byte 4	YYh	YYh: Current [A] Voltage= $YY_d \cdot 35,5/255$
Byte 5,7	00h	Not used

24. Object 2008h: Output State

This object sets and reads the state of each output pin.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	23h	Write 4-bytes
Byte 1	08h	CAN Object 2008h
Byte 2	20h	
Byte 3	00h	Highest sub-index supported (read only)
	01h	Output OFF
	02h	Output HIGH
	03h	Output LOW (on supported pins)[1]
	04h	Output FAULT (read only)
Byte 4	P8 P7 P6 P5 P4 P3 P2 P1	Each bit sets a pin
Byte 5	P31 P30 P21 P20 P16 P11 P10 P9	
Byte 6	0 0 P40 P39 P38 P37 P33 P32	
Byte 7	00h	Not used

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	23 08 20 02 08 00 00 00	Set PIN 4 HIGH
From PowerCore	58C	Std	60 08 20 02 00 00 00 00	ACK
To PowerCore	60C	Std	40 08 20 01 00 00 00 00	Who is OFF?
From PowerCore	58C	Std	43 08 20 01 00 00 00 00	No output OFF
To PowerCore	60C	Std	40 08 20 02 00 00 00 00	Who is HIGH?
From PowerCore	58C	Std	43 08 20 02 FF FF 00 00	PINs from 1 to 11 and 16, 20, 21, 30, 31 are HIGH
To PowerCore	60C	Std	23 08 20 03 C0 03 00 00	Set PINs 7, 8, 9, 10 LOW
From PowerCore	58C	Std	60 08 20 03 00 00 00 00	ACK
To PowerCore	60C	Std	40 08 20 04 00 00 00 00	Who is FAULT?
From PowerCore	58C	Std	43 08 20 04 01 00 00 00	PIN 1 is fault state

[1] Please refer to the Pin Assignment chapter to know which outputs support this feature.

25. Object 2009h: PWM State

This object enables the PWM on the outputs (PIN 6, PIN 10 and PIN 31) supporting this feature.
NOTE: if this feature is enabled when the outputs are already HIGH, the previous command is overvritten.
NOTE 2: the PWM state is applicable to a single pin or all the pins supporting this feature at the same time.
NOTE 3: the use of PIN 6 and PIN 31 as PWM state is not recommended for purely inductive loads (e.g.: motor, extractor fan)!
NOTE 4: the PWM state is applicable to a signal whose frequency does not exceed 980Hz (for PIN 6 and PIN 10) or 245Hz (for PIN31 only.)

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	23h	Set Device Register
Byte 1	09h	CAN Object 2009h
Byte 2	20h	
Byte 3	00h	Highest sub-index supported (read only)
	01h	Output PWM
Byte 4	XXh	Duty cycle level (D) of PIN 6 00h-FEh → 0%-100%
Byte 5	YYh	Duty cycle level (D) of PIN 10 00h-FEh → 0%-100%
Byte 6	ZZh	Duty cycle level (D) of PIN 31 00h-FEh → 0%-100%
Byte 7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 09 20 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 09 20 00 01 00 00 00	01h is the highest sub-index
To PowerCore	60C	Std	23 09 20 01 1A 00 00 00	Set PWM on PIN 6 with D=10%
From PowerCore	58C	Std	60 09 20 01 00 00 00 00	ACK
To PowerCore	60C	Std	23 09 20 01 00 80 00 00	Set PWM on PIN 10 with D=50%
From PowerCore	58C	Std	60 09 20 01 00 00 00 00	ACK
To PowerCore	60C	Std	23 09 20 01 00 CC CC 00	Set PWM on PINs 10 and 31 with D=80%
From PowerCore	58C	Std	60 09 20 01 00 00 00 00	ACK
To PowerCore	60C	Std	40 09 20 01 00 00 00 00	Who is PWM?
From PowerCore	58C	Std	43 09 20 01 1A 00 00 00	PIN 6 in PWM state with D=10%

26. Object 200Ah: Tachometer counter

This object allows to read the current tachometer counter for PIN 32 and to set the start value. The default counter value is 0000h.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Bh	Write 2-byte data
Byte 1	0Ah	CAN Object 200Ah
Byte 2	20h	
Byte 3	00h	Highest sub-index supported
	01h	Set start counter value
Byte 4	XXh	00-FF
Byte 5	YYh	00-FF
Byte 6,7	00h	Not used

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 0A 20 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 0A 20 00 01 00 00 00	01h is the highest sub-index supported
To PowerCore	60C	Std	40 0A 20 01 00 00 00 00	Read current tachometer value
From PowerCore	58C	Std	4B 0A 20 01 AA AA 00 00	Current tachometer value = AAAAh
To PowerCore	60C	Std	2B 0A 20 0 01 00 00 00	Set start tachometer value to 01h
From PowerCore	58C	Std	60 0A 20 00 00 00 00 00	ACK

27. Object 2010h: Baud rate settings

This object sets and reads the baud rate.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Fh	Write 1-byte data
Byte 1	10h	CAN Object 2010h
Byte 2	20h	
Byte 3	00h	Sub index
Byte 4	XXh	XXh: 02h: 500 kbit/s 03h: 250 kbit/s 04h: 125kbit/s
Byte 5,7	00h	Not used

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 10 20 00 00 00 00 00	Read baud rate
From PowerCore	58C	Std	4F 10 20 00 03 00 00 00	Baud rate = 250k
To PowerCore	60C	Std	2F 10 20 00 02 00 00 00	Set baud rate at 500k
From PowerCore	58C	Std	60 10 20 00 00 00 00 00	ACK
To PowerCore	60C	Std	2F 10 20 00 04 00 00 00	Set baud rate at 125k
From PowerCore	58C	Std	60 10 20 00 00 00 00 00	ACK

28. Object 2011h: Set node ID

This object sets and reads the CANopen node-id.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Fh	Write 1-byte data
Byte 1	11h	CAN Object 2011h
Byte 2	20h	
Byte 3	00h	Sub index
Byte 4	XXh	New node id: (01h – 7Fh)
Byte 5,7	00h	Not used

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 11 20 00 00 00 00 00	Read Node ID
From PowerCore	58C	Std	4F 11 20 00 15 00 00 00	ID = 15h
To PowerCore	60C	Std	2F 11 20 00 16 00 00 00	Set ID = 16h
From PowerCore	58C	Std	60 11 20 00 00 00 00 00	ACK

29. Object 2012h: Set Periodic Messages

This object sets the periodic transmission of TPDO 180h (Fault Message).

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Bh	Write 2-byte data
Byte 1	12h	CAN Object 2012h
Byte 2	20h	
Byte 3	00h	Sub index
Byte 4	YYh	YYh: Periodic transmission timer in milliseconds LSByte
Byte 5	XXh	XXh: Periodic transmission timer in milliseconds MSByte
Byte 6,7	00h	Not used

Periodic transmission timer: XXYYh (from 0005h to FFFEh: from 5 to 65534 milliseconds).

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 12 20 00 00 00 00 00	Read period set
From PowerCore	58C	Std	4B 12 20 00 0A 00 00 00	Period = 100ms
To PowerCore	60C	Std	2B 12 20 00 32 00 00 00	Set period = 500ms
From PowerCore	58C	Std	60 12 20 00 00 00 00 00	ACK

30. Object 2013h: Set Device Active at Startup

If the device is active on startup don’ t need start command from master.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Fh	Write 1-byte data
Byte 1	13h	CAN Object 2013h
Byte 2	20h	
Byte 3	00h	Sub index
Byte 4	XXh	00h: NOT ACTIVE 01h: ACTIVE
Byte 5,7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 13 20 00 00 00 00 00	Read register
From PowerCore	58C	Std	4F 13 20 00 00 00 00 00	Not active at start
To PowerCore	60C	Std	2F 13 20 00 01 00 00 00	Set active on start
From PowerCore	58C	Std	60 13 20 00 00 00 00 00	ACK

31. Object 2014h: Set Boot-up Service

This object enables and disables the boot-up message.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Fh	Write 1-byte data
Byte 1	14h	CAN Object 2014h
Byte 2	20h	
Byte 3	00h	Sub index
Byte 4	XXh	00h: NOT ACTIVE 01h: ACTIVE
Byte 5,7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 14 20 00 00 00 00 00	Read register
From PowerCore	58C	Std	4F 14 20 00 00 00 00 00	Not active at start
To PowerCore	60C	Std	2F 14 20 00 01 00 00 00	Set active on start
From PowerCore	58C	Std	60 14 20 00 00 00 00 00	ACK

32. Object 1016h: Consumer Heartbeat Time

The consumer heartbeat time object shall indicate the expected heartbeat cycle times. Monitoring of the heartbeat producer shall start after the reception of the first heartbeat.

NOTE 1: the heartbeat consumer time should be greater (typically twice) than the related heartbeat time to be monitored coming from the producer.

NOTE 2: if the PowerCore does not receive the heartbeat message producer anymore, it turns off all the outputs and goes into pre-operational state until a new NMT start message is received, even if the producer restarts to transmit the heartbeat.

NOTE 3: if the consumer heartbeat time is set with a value lower than the producer one, the PowerCore will not be able to change its state from pre-operational to operational.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	23h	Set device register
Byte 1	16h	CAN Object 1016h
Byte 2	10h	
Byte 3	ZZh	00h: Highest sub-index supported (read-only) 01h: Sub-index (read/write)
Byte 4	YYh	YYh: Heartbeat time in milliseconds LSByte
Byte 5	XXh	XXh: Heartbeat time in milliseconds MSByte
Byte 6	NNh	Node to be monitored 01h-7Fh (01h default)
Byte 7	00h	Reserved

Heartbeat time: XXYYh (from 000Ah to FFFEh: from 10 to 65534 milliseconds)
When the period is set to 0000h, the consumer heartbeat function is disabled.

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 16 10 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 16 10 00 01 00 00 00	01h is the highest sub-index supported
To PowerCore	60C	Std	23 16 10 01 64 00 7E 00	Set heartbeat time consumer = 100ms expected from the node 7Eh
From PowerCore	58C	Std	60 16 10 01 00 00 00 00	ACK
To PowerCore	60C	Std	23 16 10 01 F4 01 01 00	Set heartbeat time consumer= 500ms expected from the node 01h
From PowerCore	58C	Std	60 16 10 01 00 00 00 00	ACK
To PowerCore	60C	Std	40 16 10 01 00 00 00 00	Read heartbeat consumer time expected from the node 01h
From PowerCore	58C	Std	43 16 10 01 F4 01 01 00	Heartbeat consumer time set to 500ms

33. Object 1017h: Producer Heartbeat Time

The producer heartbeat time shall indicate the configured cycle time of the heartbeat.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
	2Bh	Set device register
Byte 1	17h	CAN Object 1017h
Byte 2	10h	
Byte 3	00h	Sub index
Byte 4	YYh	YYh: Heartbeat time in milliseconds
Byte 5	XXh	XXh: Heartbeat time in milliseconds
Byte 6, 7	00h	Not used

Heartbeat time: XXYYh (from 0005h to FFFEh: 5ms to 65534 milliseconds)

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 17 10 00 00 00 00 00	Read heartbeat time
From PowerCore	58C	Std	4B 17 10 00 64 00 00 00	Heartbeat time = 100ms
To PowerCore	60C	Std	2B 17 10 00 00 00 00 00	Switch off the heartbeat
From PowerCore	58C	Std	60 17 10 00 00 00 00 00	ACK
To PowerCore	60C	Std	2B 17 10 00 32 00 00 00	Heartbeat time = 50ms
From PowerCore	58C	Std	60 17 10 00 00 00 00 00	ACK
To PowerCore	60C	Std	2B 17 10 00 F4 01 00 00	Heartbeat time = 500ms
From PowerCore	58C	Std	60 17 10 00 00 00 00 00	ACK

34. Object 1000h: Device Type

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	00h	CAN Object 1000h
Byte 2	10h	
Byte 3	00h	Sub Index
Byte 4, 7	00h	Not used

Example:

Direction	**Identifier**	**Format**	**Message**
To PowerCore	60C	Std	40 00 10 00 00 00 00 00
From PowerCore	58C	Std	43 00 10 00 91 01 03 00

Device profile number: 30191h generic I/O module.

I/O Functionality: digital I/O.

Mapping PDOs: Device specific PDO supported.

35. Object 1001h: Error Register

This object is not yet implemented in the device.

36. Object 1008h: Manufacturer Device Name

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	08h	CAN Object 1008h
Byte 2	10h	
Byte 3, 7	00h	Not used

1° additional byte

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	60h	Read Device Register Next Byte
Byte 1, 7	00h	Not used

2° additional byte

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	70h	Read Device Register Next Byte
Byte 1, 7	00h	Not used

Example:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 08 10 00 00 00 00 00	
From PowerCore	58C	Std	41 08 10 00 0B 00 00 00	
To PowerCore	60C	Std	60 00 00 00 00 00 00 00	
From PowerCore	58C	Std	00 42 6C 69 6E 6B 4D 61	BlinkMa
To PowerCore	60C	Std	70 00 00 00 00 00 00 00	
From PowerCore	58C	Std	17 72 69 6E 65 00 00 00	rine

Manufacturer Device Name: BlinkMarine
The first byte of the last data message replied is 17h.

37. Object 1009h: Manufacturer Hardware Revision

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	09h	CAN Object 1009h
Byte 2	10h	
Byte 3, 7	00h	Not used

Example:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 09 10 00 00 00 00 00	
From PowerCore	58C	Std	43 09 10 00 56 5F 30 31	V_01

Manufacturer Hardware Revision: V_01

38. Object 100Ah: Manufacturer Firmware Revision

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	0Ah	CAN Object 100Ah
Byte 2	10h	
Byte 3, 7	00h	Not used

Example:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 0A 10 00 00 00 00 00	
From PowerCore	58C	Std	43 0A 10 00 31 2E 30 30	1.00

Manufacturer Firmware Revision: 1.00

39. Object 100Bh: Model ID

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	0Bh	CAN Object 100Bh
Byte 2	10h	
Byte 3, 7	00h	Not used

1° additional byte

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	60h	Read Device Register second byte
Byte 1, 7	00h	Not used

Example:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 0B 10 00 00 00 00 00	
From PowerCore	58C	Std	41 0B 10 00 07 00 00 00	
To PowerCore	60C	Std	60 00 00 00 00 00 00 00	
From PowerCore	58C	Std	01 50 43 32 31 42 00 00	PC21B

Model ID: PowerCore

The first byte of the last data message replied is 01h.

40. Object 1011h: Restore default parameters

With this object the default values of parameters according to the communication profile, device profile, and application profile are restored. This procedure shall only be executed when the specific signature ''load'' is written to the sub-index 01h. When the message shown in the following table is transmitted, the default values shall be restored after the keypad is reset.

Identifier	600h + current CAN ID	Default 60C
Byte 0	40h	Read Device Register
	23h	Set Device Register
Byte 1	11h	CAN Object 1011h
Byte 2	10h	
Byte 3	00h	Highest sub-index supported
	01h	Restore all parameters
Byte 4	6Ch	Character 1 ''l''
Byte 5	6Fh	Character 2 ''o''
Byte 6	61h	Character 3 ''a''
Byte 7	64h	Character 4 ''d''

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 11 10 00 00 00 00 00	Read highest sub-index
From PowerCore	58C	Std	4F 11 10 00 01 00 00 00	1
To PowerCore	60C	Std	23 11 10 01 6C 6F 61 64	'load'
From PowerCore	58C	Std	60 11 10 01 00 00 00 00	

41. Object 1018h: Identity object

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	18h	CAN Object 1018h
Byte 2	10h	
Byte 3	XXh	00h: Highest sub-index supported 01h: Vendor ID
Byte 4,7	00h	Not used

Example:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 18 10 00 00 00 00 00	
From PowerCore	58C	Std	4F 18 10 00 01 00 00 00	01h
To PowerCore	60C	Std	40 18 10 01 00 00 00 00	
From PowerCore	58C	Std	43 18 10 01 E2 03 00 00	000003E2h

Highest sub-index supported: 01h;

BlinkMarine Vendor ID: 000003E2h.

42. Object 1400h: Receive PDO 0 Communication Parameter

Describes the RPDO 0 parameters for setting the output PDO message.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	00h	CAN Object 1400h
Byte 2	14h	
Byte 3	00h	Highest sub-index supported
	01h	COB-ID used by RPDO
	02h	Transmission type
Byte 4,7	00h	Not used

Examples:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 00 14 00 00 00 0 0 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 00 14 00 02 00 00 00	02h
To PowerCore	60C	Std	40 00 14 01 00 00 00 00	
From PowerCore	58C	Std	43 00 14 01 0C 02 00 00	0000 020Ch
To PowerCore	60C	Std	40 00 14 02 00 00 00 00	
From PowerCore	58C	Std	4F 00 14 02 FE 00 00 00	FEh

Highest sub-index supported: 02h;

COB-ID used by RPDO: 0000 020Ch = 0000 0200h + Node-ID;

Transmission type: event-driven (manufacturer-specific).

43. Object 1600h: Receive PDO 0 Mapping Parameter

Describes the RPDO 0 mapping parameters for setting the output PDO message.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	00h	CAN Object 1600h
Byte 2	16h	
Byte 3	XXh	00h: Number of mapped objects
		01h: PDO Mapping Entry 1
		02h: PDO Mapping Entry 2
		03h: PDO Mapping Entry 3
Byte 4,7	00h	Not used

Example:

Direction	Identifier	Format	Message	Data
To PowerCore	60C	Std	40 00 16 00 00 00 00 00	Read number of mapped objects
From PowerCore	58C	Std	4F 00 16 00 03 00 00 00	03
To PowerCore	60C	Std	40 00 16 01 00 00 00 00	
From PowerCore	58C	Std	43 00 16 01 18 01 08 20	2008 01 18
To PowerCore	60C	Std	40 00 16 02 00 00 00 00	
From PowerCore	58C	Std	43 00 16 02 18 02 08 20	2008 02 18
To PowerCore	60C	Std	40 00 16 03 00 00 00 00	
From PowerCore	58C	Std	43 00 16 03 10 03 08 20	2008 03 10

Number of mapped objects: 03h;

PDO Mapping ENTRY 1: set output pin OFF: index 2008h, sub-index 01h, length 18h;
PDO Mapping ENTRY 2: set output pin HIGH: index 2008h, sub-index 02h, length 18h;
PDO Mapping ENTRY 3: set output pin LOW: index 2008h, sub-index 03h, length 10h.

44. Object 1800h: Transmit PDO 0 Communication Parameter

Describes the TPDO communication parameters for output FAULT state.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	00h	CAN Object 1800h
Byte 2	18h	
Byte 3	00h	Highest sub-index supported
	01h	COB-ID used by TPDO
	02h	Transmission type
Byte 4,7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 00 18 00 00 00 00 00	Read highest sub-index supported
From PowerCore	58C	Std	4F 00 18 00 02 00 00 00	02h
To PowerCore	60C	Std	40 00 18 01 00 00 00 00	
From PowerCore	58C	Std	43 00 18 01 8C 01 00 00	0000 018Ch
To PowerCore	60C	Std	40 00 18 02 00 00 00 00	
From PowerCore	58C	Std	4F 00 18 02 FE 00 00 00	FEh

Highest sub-index supported: 02h;

COB-ID used by TPDO: 0000 018Ch; 0000 0180h + Node-ID;

Transmission type: event-driven (manufacturer-specific).

45. Object 1A00h: Transmit PDO 0 Mapping Parameter

Describes the TPDO 0 communication parameters for reading output FAULT states.

Identifier	600h + current CAN ID	Default 60Ch
Byte 0	40h	Read Device Register
Byte 1	00h	CAN Object 1A00h
Byte 2	1Ah	
Byte 3	00h	Number of mapped objects
	01h	1st application object
Byte 4,7	00h	Not used

Examples:

Direction	**Identifier**	**Format**	**Message**	**Data**
To PowerCore	60C	Std	40 00 1A 00 00 00 00 00	Read number of mapped objects
From PowerCore	58C	Std	4F 00 1A 00 01 00 00 00	01
To PowerCore	60C	Std	40 00 1A 01 00 00 00 00	
From PowerCore	58C	Std	43 00 1A 01 18 04 08 20	2008 04 18

Number of mapped objects: 1;

Application object: Read output FAULT state;

Index 2008h; sub-index 04h; length 18h.

46. Revision history

Date	Manual Revision	Comment	Related SW version
12/04/2021	1.0	First release	x.x
04/05/2021	1.1	Second release: - Added NOTE 3 in the object 2009h - Added warning note in the chapter 3	x.x