Product Manual

IQSBlock 20 Output I/O Module

Part Number

PN 100001162

STW Technic, LP
3000 Northwoods Parkway
Suite 240
Peachtree Corners, GA 30071
United States of America

support@stw-technic.com
https://stw-technic.com

20200513_v2.1

Contents

Controller Area Network Message Structure:

SAE J1939 Compliance:

This module shall make use of the SAE J1939 Proprietary Message structure (both proprietary A and B messages) which is defined within document SAE J1939-21 and is as follows:

5.4.5 Group Function

This Message Type is used for groups of special functions (e.g. proprietary functions, network management functions, multipacket transport functions, etc.). Each group function is recognized by its assigned PGN. See Figure 10 and Figure 11. Table D1 in 0 has the Suspect Parameter Number assignments to parameters in these messages. The function itself is defined within the data structure (typically the first byte of the data field). More detailed explanation of the group functions proprietary and transport protocol will be given in the sections to follow. The proprietary group function provides a means to transmit proprietary messages in a way that eliminates CAN Identifier usage conflicts between different manufacturers. It also provides a means for receiving and distinguishing proprietary messages for use when desired. Group Functions may need to provide their own request, ACK, and/or NACK mechanisms if the messages defined in J1939-21 are not sufficient.

A request using PGN 59904 (reference Section 5.4.2) can be used to find out if a specific Parameter Group of the message type, Group Function, is supported. If it is supported, then the responding device sends the Acknowledgment PGN with the control byte equal to zero, for Positive Acknowledgment, or equal to two, Access Denied or equal to three, Cannot Respond. If it is not supported, the responding device sends the Acknowledgment PGN with the control byte set to one, for Negative Acknowledgment. The remaining portions of the SAE J1939 PDU format and Parameter Group must be filled in appropriately (reference Section 5.4.4). Note that per the definitions in this paragraph the phrase "not supported" means that the PG is not transmitted. It is not possible to determine whether a device will act upon (when received) the PG by using this method.

Parameter Group Name:	**Proprietary A**
Definition:	This proprietary PG uses the destination specific PDU format allowing manufacturers to direct their proprietary communications to a specific destination node. How the data field of this message is used is up to each manufacturer. Use of proprietary messages is at the manufacturer's discretion with the constraint that significant percentages (2 percent or more) of vehicle network utilization must be avoided.
Transmission repetition rate:	Per user requirements
Data length:	0 to 1785 bytes (multipacket supported)
Data Page:	0
PDU Format:	239
PDU Specific:	Destination Address
Default priority:	6
Parameter Group Number:	61184 ($00EF00_{16}$)
Byte: 1-8	Manufacturer specific use (see 5.1.2)

Data ranges for parameters used by this Group Function:
None defined by SAE

FIGURE 10 PROPRIETARY A PGN DEFINITION

Parameter Group Name:	**Proprietary B**
Definition:	This proprietary PG uses the PDU2 Format message allowing manufacturers to define the PS (GE) field content as they desire. However, significant percentages (2 percent or more) of vehicle network utilization must be avoided. How the PS (GE) and data fields of this message are used is up to each manufacturer. The data length of these messages has been left up to each manufacturer. Therefore, two manufacturers, say of transmissions, may use the same GE value and it may very well have a different Data Length Code. Receivers of this information would need to differentiate between the two manufacturers.
Transmission repetition rate:	Per user requirements
Data length:	0 to 1785 bytes (multipacket supported)
Data Page:	0
PDU Format:	255
PDU Specific:	Group Extension (manufacturer assigned)
Default priority:	6
Parameter Group Number:	65280 to 65535 ($00FF00_{16}$ to $00FFFF_{16}$)
Byte: 1-8	Manufacturer defined usage (see Section 5.1.2)

Data ranges for parameters used by this Group Function:
Manufacturer defined usage allows the Data Length Code being different per component supplier and source address. Caution should be used when using the Proprietary B Parameter Group (PGN = 65280) because multiple source addresses can use the same Proprietary B PGN value for different purposes.

FIGURE 11 PROPRIETARY B PGN DEFINITION

Proprietary A message Summary:

Proprietary A message overview:

This message is intended to be used as a destination specific message (point to point) and shall be sent directly to an intended device on the CAN bus by using defined PGN's and NODE ID's in association with message priority on the CAN bus.

Proprietary A message CAN-ID:

The CAN-ID consists of 4 Bytes of data containing 29-bits (variable) to identify the message parameters, the 3 additional bits are setup to give the message's priority on the CAN bus.

- CAN-ID: 0x00EF_ _ _ _
    - 00 Contains the message priority and the beginning of the assigned Proprietary PGN
    - EF Contains the remaining bits of the assigned Proprietary PGN
    - _ _ Contains the NODE-ID for the intended recipient (destination) of this message
    - _ _ Contains the NODE-ID (source) for this message
- Example CAN-ID: 0x00EF0201
    - 00 Contains the message priority and the beginning of the assigned Proprietary PGN
        - This byte's values are assigned specifically as Proprietary A message
    - EF Contains the remaining bits of the assigned Proprietary PGN
        - This byte's values are assigned specifically as Proprietary A message
    - 02 Contains the NODE-ID for the intended recipient (destination) of this message
        - This message is being sent to NODE-ID 0x02 on the CAN bus
    - 01 Contains the NODE-ID (source) for this message
        - The device sending this message has a NODE-ID 0x01 on the CAN bus

Proprietary B message Summary:

Proprietary B message definition overview:

This PGN message is intended to be used as a global broadcast message and is not intended to be directed at any specific device on the CAN bus.

Proprietary B message CAN-ID:

The CAN-ID consists of 4 Bytes of data containing 29-bits (variable) to identify the message parameters, the 3 additional bits are setup to give the message's priority on the CAN bus.

- CAN-ID: 0x00FF_ _ _ _
    - 00 Contains the message priority and the beginning of the assigned Proprietary PGN
    - FF Contains the remaining bits of the assigned Proprietary PGN
    - _ _ Contains the global message number
    - _ _ Contains the NODE-ID (source) for this message

- Example CAN-ID: 0x00FF01B0
    - 00 Contains the message priority and the beginning of the assigned Proprietary PGN
        - This byte's values are assigned specifically as Proprietary B message
    - FF Contains the remaining bits of the assigned Proprietary PGN
        - Any global broadcast message will have the value 0xFF in this location
    - 01 Contains the value that identifies the global message number
        - Message number can be anything from (0x00-0xFF)
    - B0 Contains the NODE-ID (source) for this message
        - The device sending this message has a NODE-ID 0xB0 on the CAN bus

Message Data Breakdown:

Message Data Length:

Messages will all have a consistent data length of 8 bytes total.

Message Structure:

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
Multiplex	Subset	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5

Multiplex, Byte 0:

Messages will have a specific and consistent byte containing a predetermined value which allows multiple functions to be controlled.

Subset, Byte 1:

Messages will have a specific and consistent byte containing a subset value which allows for ample control bits available and can therefore be assigned within required function messages.

Data 0 – 5, Byte 2 - 7:

Messages will have 6 bytes which contain all necessary data.

Broadcast Sent from Output Module (Wake Up):

This message follows the Proprietary Message B structure and will be broadcast on the CAN bus as a global message.

Wake Up Broadcast (0x01):

This message shall be sent on a 2 second cycle and provides specific module information.

Multiplex Byte Value: 0x01

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x01	Month	Day	Year MSB	Year LSB	Code Major Rev	Code Minor Rev	Module Status

Module Status:	Hex Value:
Needs Config.	0x00
Config. Ok	0x01
Config. Unknown	0x02
No Comm.	0x05
Active Run	0x06

Messages Sent to Output Module (Configurations):

These messages follow the Proprietary Message A structure and will be directed to the address (node) ID of this module.

Configure Serial Number, Echo, and Open Load Fault Enable (0xA0):

This basic startup message will be sent to an output module as a “wake up” indicator. Saved Output configurations can be loaded by sending this message.

Multiplex Byte Value: 0xA0

Subset Byte Value: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA0	0x00	Serial Byte [0]	Serial Byte [1]	Serial Byte [2]	Serial Byte [3]	Echo Byte	Open Load Fault Enable

Serial Byte(s):

- Must have within range of 0x01 to 0xFE
- If any number sent is 0x00 or 0xFF the entire message will be ignored

Echo Byte:

- By turning on the echo feature, any message sent to the output module will be returned to the sending node exactly as it was sent, this function is intended for diagnostics only as it will double the number of messages sent to the module, it is not for normal use.
- [0x00] Echo Off, [0x01] Echo On

Open Load Fault Enable:

- By enabling open load faults, the output module will flash a fault on the output LED if that output is on and reading less than 280mA (digital) or 375mA (PWM).
- [0x00] open load fault disabled, [0x01] open load fault enabled (open load fault is enabled by default)

Configuring Status Outputs (0xA1):

Outputs must be configured in either PWM or Digital status before they will function and activate. The following section outlines the necessary message required to setup outputs.

Multiplex Byte Value: 0xA1

Subset Byte Value: 0x00 (outputs 1 to 12) or 0x01 (outputs 13 to 20)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA1	0x00	Out2/Out1	Out4/Out3	Out6/Out5	Out8/Out7	Out10/Out9	Out12/Out11

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA1	0x01	Out14/Out13	Out16/Out15	Out18/Out17	Out20/Out19	0xFF	0xFF

Bytes will be divided into 4 high and 4 low bits which will be assigned a designated output.

Example: Byte 2 contains a hex value of 0xFF

- Yellow = High Bits (Even numbered outputs)
- Blue = Low Bits (Odd numbered outputs)

Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits

Outputs can be configured in 2 different states which are Digital or PWM.

Output Status:	Hex Value:	Binary Value:
DIGITAL	0x01	0 0 0 1
PWM	0x02	0 0 1 0

Example: Message to setup output 1-10 PWM, outputs 11 and 12 DIGITAL.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA1	0x00	0x22	0x22	0x22	0x22	0x22	0x11

Control Digital Outputs (0xA2):

Digital outputs have two possible active states which can be switched between, ON and OFF.

Multiplex Byte Value: 0xA2

Subset Byte Value: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA2	0x00	0x00	0x00	0x00	0x00	0x00	0xFF

Each Data Byte will be divided into 4 sets of bits (2 bits each), these sets will be assigned a specific output to control and will be setup to active or deactivate that output.

Example: Byte 2 (Bits) __ __ __ __ __ __ __ __

Yellow controls Output 4

Blue controls Output 3

Green controls Output 2

Pink controls Output 1

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0xA2	0x00	Out4/Out3/Out2/Out1	Out8/Out7/Out6/Out5	Out12/Out11/Out10/Out9

Byte 5	Byte 6	Byte 7
Out16/Out15/Out14/Out13	Out20/Out19/Out18/Out17	0xFF

Digital State:	Hex Value:	Binary Value:
OFF	0x00	0 0
ON	0x01	0 1

Example: Message to turn ON outputs 2, 7, 9, 12, 15, and 16. All other outputs are OFF.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA1	0x00	0x04	0x10	0x41	0x50	0x00	0xFF

Setup Frequency for PWM Outputs (0xA3):

Frequency must be setup before an output desired to use with PWM functionality will work.

Multiplex Byte Value: 0xA3

Subset Byte Values: 0x00 (outputs 1 to 6), 0x01 (outputs 7 to 10)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA3	0x00	Ouput1	Output2	Output3	Output4	Output5	Output6

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA3	0x01	Output7	Output8	Output9	Output10	0xFF	0xFF

Outputs [1-10] have a frequency range of 5Hz to 250Hz, this allows a single byte of data to contain a single output frequency.

Example: Set Frequencies of Output 8 to 150hz & Output 10 to 50hz. All others set as 0 (non-PWM).

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA3	0x01	0x00	0x96	0x00	0x32	0x00	0x00

Control Duty Cycle for PWM Outputs (0xA4):

Duty Cycle must be setup before an output desired to use with PWM functionality will work. A single output duty cycle will be contained within two separate data bytes, the two data bytes will be designated as a high data byte and a low data byte.

Multiplex Byte Value: 0xA4

Subset Byte Values: 0x00 (outputs 1 to 3), 0x01 (outputs 4 to 6), 0x02 (outputs 7 to 9), 0x03 (output 10)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA4	0x00	Ouput1High	Output1Low	Output2High	Output2Low	Output3High	Output3Low

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA4	0x01	Ouput4High	Output4Low	Output5High	Output5Low	Output6High	Output6Low

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA4	0x02	Ouput7High	Output7Low	Output8High	Output8Low	Output9High	Output9Low

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA4	0x03	Ouput10High	Output10Low	0xFF	0xFF	0xFF	0xFF

Duty cycle for an output is determined by combining the high and low bytes, the high byte acts as a value which is shifted 8 bits to allow for larger values (bits marked X are unused and will be ignored):

(High byte) & (Low byte)

X X X X _ _ _ _ & _ _ _ _ _ _ _ _

By utilizing a two-byte structure messages allow for implementation of a higher resolution 0.1% duty cycle.

Example: Duty cycle = 64.1%, Decimal Value = 641
Hex Value = 0x281, High byte = 0x02 & Low byte = 0x81
Binary value = 0 0 1 0 1 0 0 0 0 0 0 1

Example: Set Output 7 to 75.5%, Output 8 to 0% (off), and Output 9 to 35.8%.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA4	0x02	0x02	0xF3	0x00	0x00	0x01	0x66

Set CAN Timeout (0xA5):

This message will allow an end user to change the value of the CAN timeout which determines when the module status will change to no communication. When a new timeout value is set, the module will retain that value (through multiple power cycles) until a new timeout value is set.

Multiplex Byte Value: 0xA5
Subset Value: 0x00
Timeout Value: Range of [0x01 - 0x0A] (1-10 seconds; decimal value)
Default Timeout = 2 seconds

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA5	0x00	0x _ _	0xFF	0xFF	0xFF	0xFF	0xFF

Example: Set Timeout to 5 seconds (hex value 0x05)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA5	0x00	0x05	0xFF	0xFF	0xFF	0xFF	0xFF

Reset Output Faults (0xA6):

If a fault has been detected on an output, sending this message will reset that fault condition for outputs specified by this message.

Multiplex Byte Value: 0xA6

Subset Byte Value: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA6	0x00	0x00	0x00	0x00	0x00	0x00	0xFF

Each Data Byte will be divided into 4 sets of bits (2 bits each), these sets will be assigned a specific output to control and will be setup to reset fault condition on that output.

Example: Byte 2 (Bits) __ __ __ __ __ __ __ __

Yellow controls Output4

Blue controls Output3

Green controls Output2

Pink controls Output1

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0xA6	0x00	Out4/Out3/Out2/Out1	Out8/Out7/Out6/Out5	Out12/Out11/Out10/Out9

Byte 5	Byte 6	Byte 7
Out16/Out15/Out14/Out13	Out20/Out19/Out18/Out17	0xFF

Reset Fault:	Hex Value:	Binary Value:
Reset	0x01	0 1

Example: Message to reset faults on Outputs 2, 7, 9, 12, 15, and 16.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA6	0x00	0x04	0x10	0x41	0x50	0x00	0xFF

Configure MSB Desired CAN ID (0xA7):

This message will allow an end user to change the 4 Most significant bits of the CAN ID. This change will automatically save if message is received. Note, the Least significant bits of the data sent will be used.

Multiplex Byte Value: 0xA7
Subset Value: 0x00
Node ID Value: Range of [0x00 – 0x0F] (0 to 15; decimal value) (Node ID range)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA7	0x00	0x _ _	0xFF	0xFF	0xFF	0xFF	0xFF

Example: Set Node ID to 0x5_ (hex value 0x05)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA7	0x00	0x05	0xFF	0xFF	0xFF	0xFF	0xFF

Configure Desired Baud Rate (0xA8):

This message will allow an end user to determine the first baud-rate the module begins listening for when powered on. Baud rate is saved automatically and will take effect the next power cycle.

Multiplex Byte Value: 0xA8
Subset Value: 0x00
Baud Rate: Value of [0x00 – 0x03]

Baud-Rate:	Hex Value:	Binary Value:
125K	0x00	00
250K	0x01	01
500K	0x02	10
1M	0x03	11

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA8	0x00	0x _ _	0xFF	0xFF	0xFF	0xFF	0xFF

Example: 0x02, 500k on next power cycle

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xA8	0x00	0x02	0xFF	0xFF	0xFF	0xFF	0xFF

Messages Sent to Output Module (Requests):

Request Serial, Echo, and Open Load Fault Enable Values (0xB0):

This message will request a response from the module (reply message 0xC0), which will return values of serial number, echo, and open load diagnostics enable.

Multiplex Byte Value: 0xB0

Subset Byte Values: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB0	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Output Status (0xB1):

This message will request a response from the module (reply message 0xC1), either a specific group or all statuses can be requested. An output status can be either DIGITAL or PWM.

Multiplex Byte Value: 0xB1

Subset Byte Values:
0x00 = Request outputs 1 to 12
0x01 = Request outputs 13 to 20
0x02 = Request all outputs (Both 0xC1 subsets will be sent at the same time)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB1	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Output Frequency (0xB2):

This message will request a response from the module (reply message 0xC2), either a specific group or all frequencies can be requested. Only outputs 1-10 are PWM and are applicable for this message.

Multiplex Byte Value: 0xB2

Subset Byte Values:
0x00 = For request outputs 1 to 6
0x01 = For request outputs 7 to 10
0x02 = For request all outputs (Both 0xC2 subsets will be sent at the same time)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB2	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Output PWM Duty Cycle (0xB3):

This message will request a response from the module (reply message 0xC3), either a specific group or all PWM duty cycles can be requested.

Multiplex Byte Value: 0xB3

Subset Byte Values: 0x00 = Request outputs 1 to 3
0x01 = Request outputs 4 to 6
0x02 = Request outputs 7 to 9
0x03 = Request outputs 10
0x04 = Request all outputs (All 0xC3 subsets will be sent at the same time)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB3	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request CAN Timeout Value (0xB4):

This message will request a response from the module (reply message 0xC4). This allows an end user to request the value of the CAN timeout (which determines when module status changes to no communication).

Multiplex Byte Value: 0xB4
Subset Value: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB4	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Digital Output States (0xB5):

This message will request a response from the module (reply message 0xC5). This allows the end user to request the digital state (ON/OFF) of all outputs.

Multiplex Byte Value: 0xB5
Subset Value: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB5	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Output Currents (0xB6):

This message will request a response from the module (reply message 0xC6). This is a request for the current readings of either specific or all outputs.

Multiplex Byte Value: 0xB6

Subset Byte Values:
0x00 = Request outputs 1 to 6
0x01 = Request outputs 7 to 12
0x02 = Request outputs 13 to 18
0x03 = Request outputs 19 and 20
0x04 = Request all outputs (All 0xC6 subsets will be sent at the same time)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB6	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Output Faults (0xB7):

This message will request a response from the module (reply message 0xC7). This is a request for the faults of either specific or all outputs.

Multiplex Byte Value: 0xB7

Subset Byte Values:
0x00 = Request outputs 1 to 13
0x01 = Request outputs 14 to 20
0x02 = Request all outputs (All 0xC7 subsets will be sent at the same time)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB7	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Module Temperature (0xB8):

This message will request a response from the module (reply message 0xC8) for its temperature reading.

Multiplex Byte Value: 0xB8
Subset Byte Values: 0x00 = Request temperature

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB8	0x00	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF

Request Configuration save (0xB9):

This message will request the current module configuration of output statuses and related frequencies of PWM outputs to be saved. In order to save, the serial numbers sent must match the serial numbers previously loaded for configuration of the module. On the next power cycle when the saved serial number is entered for configuration, it will load the saved configuration.

Multiplex Byte Value: 0xB9

Subset Byte Values: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xB9	0x00	Serial Number [0]	Serial Number [1]	Serial Number [2]	Serial Number [3]	0xFF	0xFF

Messages Sent from Output Module (Reply):

These messages follow the Proprietary Message A structure and will be sent from the address (node) ID of this module to its intended destination.

Reply Serial Number, Echo, and Open Load Fault Enable Values (0xC0):

This message is a direct reply to a node requesting serial number, echo values, and open load diagnostics enable (request message 0xB0). Echo function is intended for diagnostic use only, not normal use.

Multiplex Byte Value = 0xC0

Subset Byte Value = 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC0	0x00	Serial Byte [0]	Serial Byte [1]	Serial Byte [2]	Serial Byte [3]	Echo Byte	Open Load Fault Enable

Echo/Open Load Fault:	Hex Value:	Binary Value:
INACTIVE	0x00	0 0 0 0
ACTIVE	0x01	0 0 0 1

Reply Output Status (0xC1):

This message is a direct reply to a node requesting output status (request message 0xB1).

Multiplex Byte Value: 0xC1

Subset Byte Value: 0x00 (outputs 1 to 12) or 0x01 (outputs 13 to 20)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC1	0x00	Out2/Out1	Out4/Out3	Out6/Out5	Out8/Out7	Out10/Out9	Out12/Out11

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC1	0x01	Out14/Out13	Out16/Out15	Out18/Out17	Out20/Out19	0xFF	0xFF

Bytes will be divided into 4 high and 4 low bits which will be assigned a designated output.

Example: Byte 2 contains a hex value of 0xFF

- Yellow = High Bits (Even numbered outputs)
- Blue = Low Bits (Odd numbered outputs)

Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits

Output Status:	Hex Value:	Binary Value:
DIGITAL	0x01	0 0 0 1
PWM	0x02	0 0 1 0

Example: Reply message with outputs [1-6] configured as PWM and outputs [7-12] configured as digital

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC1	0x00	0x22	0x22	0x22	0x11	0x11	0x11

Reply Frequency for PWM Outputs (0xC2):

This message is a direct reply to a node requesting frequency for PWM outputs (request message 0xB2). Each individual output is contained in a single data byte.

Multiplex Byte Value: 0xC2

Subset Byte Values: 0x00 (outputs 1 to 6) & 0x01 (outputs 7 to 10)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC2	0x00	Ouput1	Output2	Output3	Output4	Output5	Output6

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC2	0x01	Output7	Output8	Output9	Output10	0xFF	0xFF

Example: Frequency of output 8 to 150hz and output 10 to 50hz. All others set as 0 (non-PWM).

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC2	0x01	0x00	0x96	0x00	0x32	0xFF	0xFF

Reply Duty Cycle for PWM Outputs (0xC3):

This message is a direct reply to a node requesting duty cycle for PWM outputs (request message 0xB3). A single output duty cycle will be contained within two separate data bytes, the two data bytes will be designated as a high data byte and a low data byte.

Multiplex Byte Value: 0xC3

Subset Byte Values: 0x00 (outputs 1 to 3), 0x01 (outputs 4 to 6), 0x02 (outputs 7 to 9), 0x03 (output 10)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC3	0x00	Ouput1High	Output1Low	Output2High	Output2Low	Output3High	Output3Low

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC3	0x01	Ouput4High	Output4Low	Output5High	Output5Low	Output6High	Output6Low

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC3	0x02	Ouput7High	Output7Low	Output8High	Output8Low	Output9High	Output9Low

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC3	0x03	Ouput10High	Output10Low	0xFF	0xFF	0xFF	0xFF

Duty cycle for an output is determined by combining the high and low bytes, the high byte acts as a value which is shifted 8 bits to allow for larger values (bits marked X are unused and will be ignored):

(High byte) & (Low byte)

X X X X __ __ __ __ & __ __ __ __ __ __ __ __

By utilizing a two-byte structure messages allow for implementation of a higher resolution 0.1% duty cycle.

Example: This message shows Output 7 set to 75.5%, Output 8 set to 0% (off), and Output 9 set to 35.8%.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC3	0x02	0x02	0xF3	0x00	0x00	0x01	0x66

Reply CAN Timeout (0xC4):

This message is a direct reply to a node requesting the CAN timeout value (request message 0xB4).

Multiplex Byte Value: 0xC4
Subset Value: 0x00
Timeout value: Range of [0x01 - 0x0A] (1-10 seconds; decimal value)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC4	0x00	0x _ _	0xFF	0xFF	0xFF	0xFF	0xFF

Example: Timeout has been set to 10 seconds (hex value 0x0A)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC3	0x00	0x0A	0xFF	0xFF	0xFF	0xFF	0xFF

Reply Digital Output States(0xC5):

This message is a direct reply to a node requesting digital output states (request message 0xB5). An output state can either be ON or OFF.

Multiplex Byte Value: 0xC5

Subset Byte Value: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC5	0x00	0x00	0x00	0x00	0x00	0x00	0xFF

Each Data Byte will be divided into 4 sets of bits (2 bits each), these sets will be assigned a specific output to control and will be setup to active or deactivate that output.

Example: Byte 2 (Bits) __ __ __ __ __ __ __ __

Yellow controls Output 4
Blue controls Output 3
Green controls Output 2
Pink controls Output 1

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4
0xC5	0x00	Out4/Out3/Out2/Out1	Out8/Out7/Out6/Out5	Out12/Out11/Out10/Out9

Byte 5	Byte 6	Byte 7
Out16/Out15/Out14/Out13	Out20/Out19/Out18/Out17	0xFF

Digital State:	Hex Value:	Binary Value:
OFF	0x00	0 0
ON	0x01	0 1

Example: Outputs 2, 7, 9, 12, 15, and 16 are on and all other digital outputs are OFF.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC5	0x00	0x04	0x10	0x41	0x50	0x00	0xFF

Reply Outputs Current Readings (0xC6):

This message is a direct reply to a node requesting readings of output currents (request message 0xB6). Please note that the current reading is shifted one decimal place to the right, this must be considered when reading this response message.

Example: Current reading of 4.1 amps, shift the decimal to the right 1 place to have a value of 41, which is now a hex value of 0x29.

Multiplex Byte Value: 0xC6

Subset Byte Value: 0x00 (outputs 1 to 6), 0x01 (outputs 7 to 12), 0x02 (outputs 13 to 18), and 0x03 (Outputs 19 and 20)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC6	0x00	Out1	Out2	Out3	Out4	Out5	Out6

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC6	0x01	Out7	Out8	Out9	Out10	Out11	Out12

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC6	0x02	Out13	Out14	Out15	Out16	Out17	Out18

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC6	0x03	Out19	Out20	0xFF	0xFF	0xFF	0xFF

Example: Reply message that Output 1 reading 4.3 amps, Output 2 is reading 10.5 amps, Output 5 is reading 11 amps, and Output 6 is reading 1.3 amps.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC6	0x00	0x2B	0x69	0x00	0x00	0x6E	0x0D

Reply Output Faults (0xC7):

This message is a direct reply to a node requesting readings of output faults (request message 0xB7).

Multiplex Byte Value: 0xC7

Subset Byte Value: 0x00 (outputs 1 to 12) or 0x01 (outputs 13 to 20)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC7	0x00	Out2/Out1	Out4/Out3	Out6/Out5	Out8/Out7	Out10/Out9	Out12/Out11

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC7	0x01	Out14/Out13	Out16/Out15	Out18/Out17	Out20/Out19	0xFF	0xFF

Bytes will be divided into 4 high and 4 low bits which will be assigned a designated output.

Example: Byte 2 contains a hex value of 0xFF

- Yellow = High Bits (Even numbered outputs)
- Blue = Low Bits (Odd numbered outputs)

Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits	High Bits / Low Bits

Output Fault Status:	Hex Value:	Binary Value:
No Fault Output Off	0x00	0 0 0 0
No Fault Output On	0x01	0 0 0 1
Open Load	0x02	0 0 1 0
Over Current	0x04	0 1 0 0
Short to Ground	0x08	1 0 0 0

Example: Message return of output 1 no fault on, output 4 open load, output 6 over current, and output 10 short to ground, all other outputs no fault output off.

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC7	0x00	0x01	0x20	0x40	0x00	0x80	0x00

Reply Module Temperature (0xC8):

This message is a response from the module (request message 0xB8). Convert the temperature data received to the decimal for the temperature reading in Celsius.

Multiplex Byte Value: 0xC8
Subset Byte Values: 0x00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0xC8	0x00	Temperature	0xFF	0xFF	0xFF	0xFF	0xFF

Communication, Configuration, and Activation Example:

- The following is an example provided to assist in ease of operation and specifically, configuration of an output of this module. This example will setup Output[1] as a digital output then set it to an on state
- This module has a default baud rate of 250Kbps.
- The example module will use the default CAN ID
    - Default CAN ID = 0xB0
    - No address pins are active (grounded) for this example please reference SPC_99-0691_001, Ground Pin Matrix section for more information
- A wake-up broadcast message shall be sent by the module at a frequency of once every two seconds, reference message 0x01 for more details.
    - Received CAN ID: 00FF01B0
    - Please reference “Broadcast sent from Output Module, 0x01” for more information
    - The status of the module will be “Configuration Unknown”
- Send Request message at a set frequency (recommended 1 second) to prevent the module from entering “No Comm” status.
    - Recommended sending message 0xB6 or 0xB7, this example will use 0xB6
    - Send CAN ID: 00EFB001 Send Data: B6 00 FF FF FF FF FF FF
        - Note that the message must be directed specifically to the module (the highlighted and underlined section), reference Proprietary A message for more information, all other values will be accepted for non-highlighted values
    - The module will respond with current data for Output[1] – Output[6], message 0xC6

- Configure Serial Number, reference message 0xA0 for more details

    - Serial number must be sent before setup of outputs, serial numbers cannot be 0x00 or 0xFF, if any bytes contain these values the message will be ignored
    - Send CAN ID: 00EFB001 Send Data: A0 00 01 02 03 04 00 01
    - Please note that this example is considering the module in question not to have a saved configuration and therefore will enter “Needs Configuration” status
- Configure Output[1] as a digital output, reference message 0xA1 for more details
    - Set the configuration bits for Output[1] to 0x01, all other outputs are also digital and equal to 0x01
    - Send CAN ID: 00EFB001 Send Data: A1 00 11 11 11 11 11 11
        - Output[1] bits have been highlighted and underlined
    - Upon receiving this message, the module will enter “Configuration Ok” status
- Control Output[1] as a digital output, reference message 0xA2 for more details
    - Set the assigned digital control bit to 1
    - Send CAN ID: 00EFB001 Send Data: A1 00 01 00 00 00 00 00
- Output[1] will activate upon receiving Control message
    - If a load has been attached to the output and is within the specified current draw limit:
        - LED[1] will turn on and remain solid on until deactivated
        - Message 0xC6, the response message from module, will provide current reading for Output[1] according to how much current is being drawn
        - If Request 0xB7 message is sent to the module:
            - Module will respond with 0xC7 message and indicate Output[1] is reading on and not faulted.
    - If a load has not been attached to the output
        - LED[1] will blink, indicating an open load if open load diagnostics are enabled
        - If Request 0xB7 message is sent to the module:
            - Module will respond with 0xC7 message and indicate Output[1] is reading an open load fault.

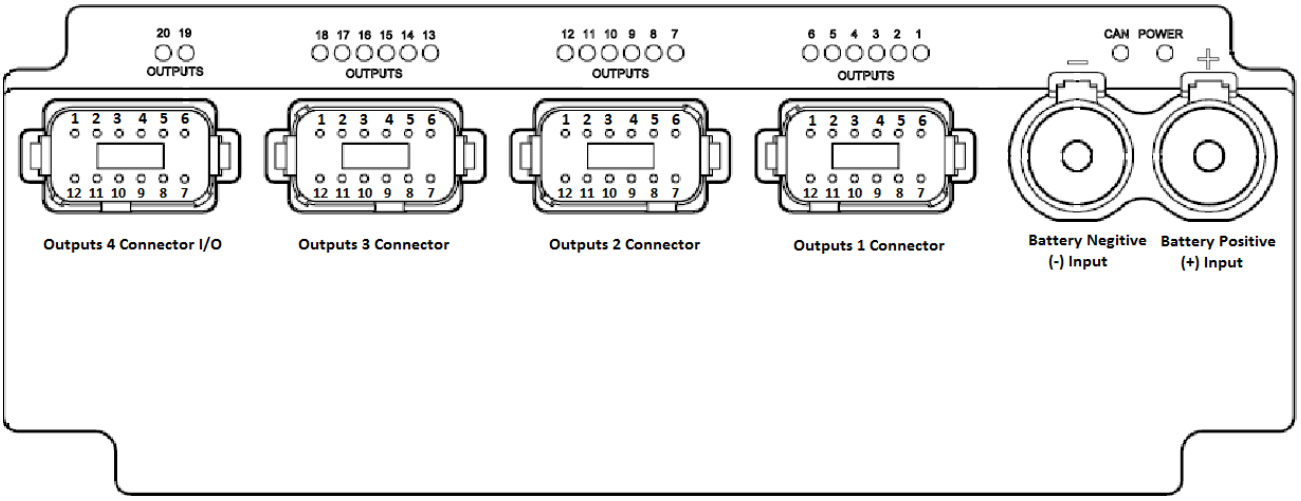
Connections & Wiring:

LEDs/Indication:

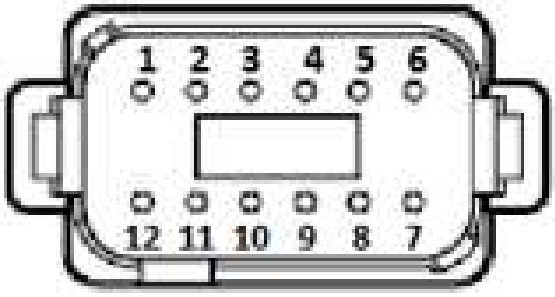
LED (Green)	Solid Off	Solid On	Flashing
Power	No Power to Unit	Power to Unit	N/A
CAN	No Microcontroller Operation	N/A	0.5secs on/4.5secs off = No CAN Comm. 0.5Hz = Valid CAN Comm. (Normal) 2Hz = Unit Overtemperature
Output	Output Off (no-fault)	Output On (no-fault)	2Hz = Output Faulted (overcurrent/short circuit/open load detected)

Module Reference Guide:

Enclosure Connector/Pin reference:



Connector Reference Pinout:



Battery Positive (+) Input:

Deutsch DTHD04-1-4P equivalent integrated into enclosure

Mating Connector Deutsch DTHD06-1-4S, Mating Deutsch Contact 0462-203-04141

Battery Negative (-) Input:

Deutsch DTHD04-1-4P equivalent integrated into enclosure

Mating Connector Deutsch DTHD06-1-4S, Mating Deutsch Contact 0462-203-04141

Outputs 1 Connector:

Connector: Deutsch DTF15-12PD equivalent integrated into enclosure

Mating Connector: DT06-12SD (Brown), Contacts: 0462-201-16141, Wedgelock: W12S

Pinout:

Pin:	Function:	Pair:
1	Ground	
2	Ground	
3	Ground	
4	Ground	
5	Ground	
6	Ground	
7	Output 1 Protected Sourcing (+Batt)	1
8	Output 2 Protected Sourcing (+Batt)	1
9	Output 3 Protected Sourcing (+Batt)	2
10	Output 4 Protected Sourcing (+Batt)	2
11	Output 5 Protected Sourcing (+Batt)	3
12	Output 6 Protected Sourcing (+Batt)	3

Outputs 2 Connector:

Connector: Deutsch DTF15-12PC equivalent integrated into enclosure

Mating Connector: DT06-12SC (Green), Contacts: 0462-201-16141, Wedgelock: W12S

Pinout:

Pin:	Function:	Pair:
1	Ground	
2	Ground	
3	Ground	
4	Ground	
5	Ground	
6	Ground	
7	Output 7 Protected Sourcing (+Batt)	4
8	Output 8 Protected Sourcing (+Batt)	4
9	Output 9 Protected Sourcing (+Batt)	5
10	Output 10 Protected Sourcing (+Batt)	5
11	Output 11 Protected Sourcing (+Batt)	6
12	Output 12 Protected Sourcing (+Batt)	6

Outputs 3 Connector:

Connector: Deutsch DTF15-12PB equivalent integrated into enclosure

Mating Connector: DT06-12SB (Black), Contacts: 0462-201-16141, Wedgelock: W12S

Pinout:

Pin:	Function:	Pair:
1	Ground	
2	Ground	
3	Ground	
4	Ground	
5	Ground	
6	Ground	
7	Output 13 Protected Sourcing (+Batt)	7
8	Output 14 Protected Sourcing (+Batt)	7
9	Output 15 Protected Sourcing (+Batt)	8
10	Output 16 Protected Sourcing (+Batt)	8
11	Output 17 Protected Sourcing (+Batt)	9
12	Output 18 Protected Sourcing (+Batt)	9

Outputs 4 & I/O Connector:

Connector: Deutsch DTF15-12PA equivalent integrated into enclosure

Mating Connector: DT06-12SA (Grey), Contacts: 0462-201-16141, Wedgelock: W12S

Pinout:

Pin:	Function:	Pair
1	Input 3, Ground pin Matrix	
2	Input 4, Ground pin Matrix	
3	N/C	
4	N/C	
5	Ground	
6	Ground	
7	Output 19 Protected Sourcing (+Batt)	10
8	Output 20 Protected Sourcing (+Batt)	10
9	CAN Low	
10	CAN High	
11	Input 1, Ground pin Matrix	
12	Input 2, Ground pin Matrix	

Mounting:

The unit can be mounted using 4x M5x0.8 fasteners or 4x #10-32 fasteners. Recommended torque is 30 in-lb. Max torque is 50 in-lb.

Ground pin Matrix (LSB CAN ID):

The Ground pin Matrix provides the end user with the ability to change the modules 4 Least Significant Bits of its CAN ID, by sinking inputs to ground via external harnessing. Please note that these inputs are only read in during powerup. Each input can be thought of as a specific bit in a 4-place binary number. If the input is not grounded the value of the bit is 0. When ground is applied the value of the bit is 1.

Input 4:	Input 3:	Input 2:	Input 1:	LSB CAN ID:
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	A
1	0	1	1	B
1	1	0	0	C
1	1	0	1	D
1	1	1	0	E
1	1	1	1	F

Key:
0 = Pin not connected or high
1 = Pin connected to ground

Environmental Requirements:

Parameter	**Min**	**Nominal**	**Max**	**Notes**
Storage/Operating Temperature	-40C	-	85C	-
Storage/Operating Humidity	0%	-	100%	Including Condensing.
Ingress Protection	-	-	IP67	Protected Against Immersion up to 1m for 1hour. With all connectors properly mated and strain relieved.
Operating Shock	-	-	25G	3 axes, 6 pulses per axes.
Drop Test	-	-	1m	Drop from height of 1m onto concrete, all dropped surfaces, without damage to unit that affects function. Connectors are not mated.

Operation of Units:

The unit runs on J1939 protocol. Once initialized and CAN communication has been established, the unit will allow control of the outputs via J1939 messaging structure. If CAN communication is lost during operations, any outputs that are on will be turned off automatically. The units will provide analog current feedback for each output, as well as a diagnostic state for each output.

Current feedback is shifted by 1 to the right to incorporate decimals. Divide by 10 to get current in Amps.

Diagnostic states should be as follows:

Diagnostics	Output State	Fault	Latch Off
Off	Output Off & No Fault	No	No
On	Output On & No Fault	No	No
Open/No Load	Output On & Current < 280mA (digital) or 375mA (PWMing)	Yes	No
Overcurrent	Current > 3A < 10A (for greater than 2 seconds)	Yes	Yes
Short Circuit/Self-Protect	Current > 10A or output self-protect (overtemperature, short circuit, etc.)	Yes	Yes

If a fault is detected it will be latched in until it is reset by a power cycle or a CAN message. Any overcurrent or short circuit/self-protect fault will cause the output to latch off until it is reset by a power cycle or CAN message. A detection of an open/no load will only occur when the output is on and will not be latched in the diagnostic until reset. An open/no load fault will not cause the output to turn off if on, nor will it prevent an output that is off from turning on. Open load detection can be disabled using wake up message (see messaging specification). LED indicator for each output should be as described in the LED indication section.

When in operational mode, the unit will constantly monitor the temperature of the output FETs via an analog thermistor. When the temperature reaches 130 degrees Celsius, all outputs will turn off and the CAN LED will flash at a rate of 2Hz. The sensor temperature can be read any time via CAN. The FETs also monitor their own temperature. If the temperature of any FET reaches 170 degrees Celsius, it will turn off its output and report an overcurrent fault.

The unit may have its CAN ID (upper nibble changed via CAN message—lower nibble changed by addressing inputs) and CAN Baud rate (250K default, changed via CAN message 0xA8) modified. Changes will not take effect until after a power cycle to limit CAN bus errors. Both are stored to non-volatile memory.

END OF DOCUMENT