Hardwire electronics

PDM 15/25

Instruction Manual

www.hardwire-electronics.co.uk

Version 3

PDM 15/25 Instruction Manual

Introduction

A PDM (Power Distribution Module) is a device used to replace conventional relays and fuses in a vehicle electrical system. The Hardwire Electronics PDM takes inputs in the form of physical switches, analogue voltages, or CAN bus messages, and provides power to different electronic devices - such as radiator fans, fuel pumps, ECUs and headlights. The current being drawn from each connected device is continuously monitored. If the measured current is too high due to a fault, the PDM switches off the respective output to prevent further damage to the wiring loom or the connected device. The PDM can then retry the output to see if the fault has cleared, or the user can manually reset the outputs via an input switch.

PDM's offer distinct advantages over conventional systems using fuse boards and relays:

- Greatly simplifies system wiring and reduces system component count, saving you money overall when labour/ time is accounted for.
- Reduces the weight of the electrical system.
- Provides increased reliability over traditional systems using mechanical relays.
- Solid state technology removes the problem of relay contact corrosion and relay bouncing in vibration heavy applications.
- Smart features such as indicator and wiper motor modes remove the need for extra control units.
- Current draw can be monitored and logged to pre-emptively detect faults.
- PDM can be integrated onto the CAN bus with ECUs and keypads for added control and logging functionality.
- CAN bus communication allows your PDM to send/receive data from other CAN bus devices in the vehicle, such as the vehicle ECU. This paves way for advanced control/ safety strategies such as automatic engine cut offs, fuel pump cut offs, radiator fan control and more.
- Easy integration with popular CAN bus keypads, improving system reliability and functionality.

Features

General	
Size	200x160x54mm
Weight	PDM25 660g | PDM15 615g
Operating Temperature	-55 - 90°C [-67 - 194°F]
Construction	Glass Fibre reinforced plastic
IP Rating	IP67 Water and dust ingress protection
Connectors	4x Deutsch DT 12-pin
Update Rate	Up to 800Hz input and output update rate
Inputs	
Type	16 analogue inputs which can also serve as digital inputs.
Measurement Range	Each input can measure voltages between 0-6V
Measurement Resolution	10-bit or 0.01V
Protection	Fully protected for input voltages between -24V to 40V
Added Features	Configure as active high or low. Configurable Hysteresis voltage. Latching or momentary operation.
Outputs	
Type	15/25 High side outputs 80A peak, 20A continuous.
Combined	180A combined output maximum current
Output current control steps	100mA
Protection	Each output is overcurrent, overtemperature, load dump and reverse polarity protected.
5V	1x 5V 500mA output for supplying external sensors.

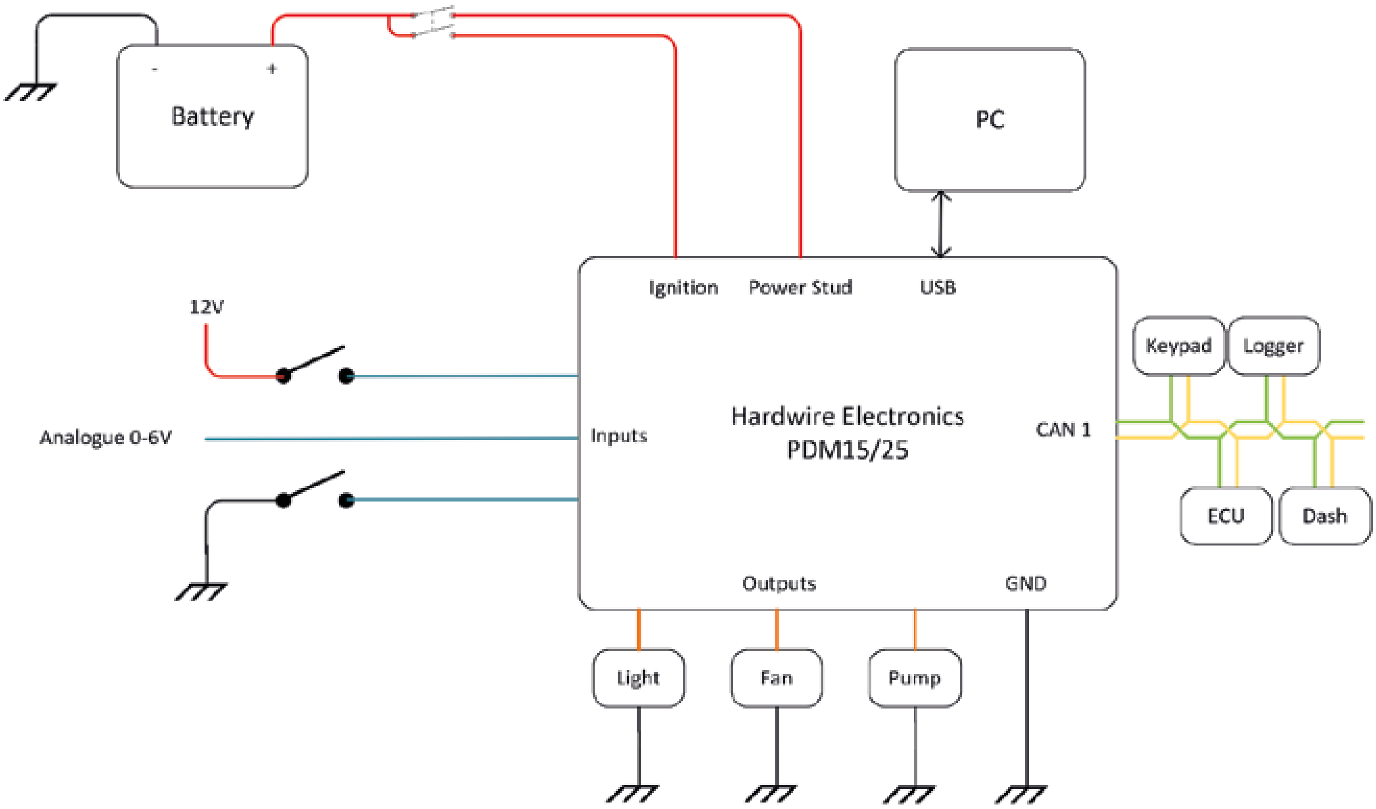
Continued on next page...

Features

Soft Start	PWM soft start can be enabled on outputs to limit inrush current.
PWM	Each output can be Pulse Width Modulated based on a user defined variable. This can limit the speed of radiator fans, fuel pumps, and dim lights.
Added features	Outputs can be configured to stay on for a specified time after the input is removed, useful for thermo-fans. Outputs can be retried multiple times after an over-current event is detected.
CAN bus	
Interface	1x CAN2.0A/B
Bus Speed	50, 100, 125, 250, 500, 1000 kbps
Input Stream	40x inputs, fully customisable variable parsing with bit masking, math functions, and compound CAN message filtering.
Output Stream	40x Outputs, Fully customisable CAN frame construction with math functions.
CAN Keypad	
Outputs can be controlled via a wide range of available CAN bus keypads. Hardwire Electronics PDMs support Blink Marine, Grayhill, and Marlin CAN keypads.	
Logging	
Storage	128Mb of onboard flash storage for logging PDM data
Speed	1Hz to 50Hz
Viewing	Data can be downloaded and graphed via the Hardwire Electronics PDM configurator software, or exported as a .CSV file and graphed in external software.
Functions	
Output Functions	Each Output can be switched with a user defined function.
Logical operators	AND, OR, NOR, XOR, NAND, NOR, >, >=, <, <=, Equal, Not Equal.
Maths Channels	
	Fully customisable Maths Functions
Mathematical Operators	Mathematical Operators - Addition, Subtraction, Multiplication, Division, Modulus, Choose, Min, Max
Binary Operators	Binary Operators - AND, OR, XOR, Left Shift, Right Shift
Counters	
Configurable counters which increment, decrement, and reset based on user defined conditions	
Other	
Gyroscope	2000°/s 3-axis measurement
Accelerometer	-8g to 8g 3-axis measurement
PCB temperature	PCB temperature measured between -40°C to 100°C
Bluetooth	
Version	Bluetooth v4.0 enabled for future app.
USB	
Type	USB 2.0 compliant with Mini-B
Driver	Integrated driver with Windows 10
PC App	
Name	Hardwire Electronics PDM configurator software.
Compatibility	Windows 10 machines
Connectivity	Easy connection to the PDM with USB & windows integrated drivers. No unreliable USB-serial converter needed.

Contents

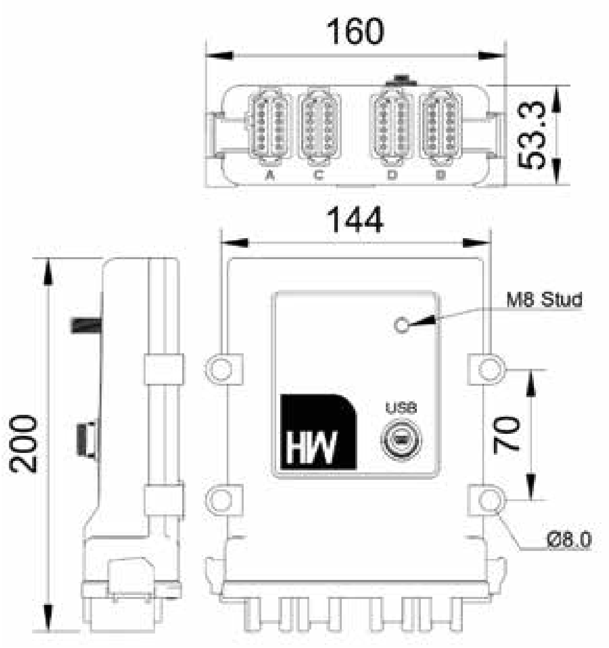
System Overview



Installation

Mounting

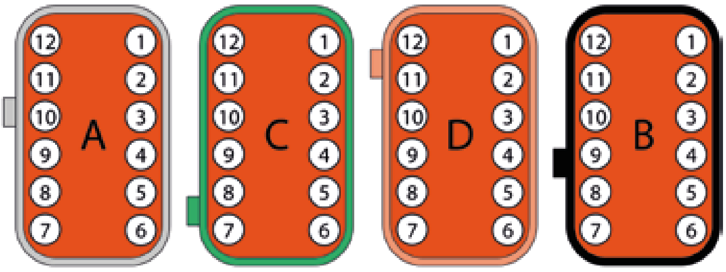
The adjacent diagram details the physical dimensions of the PDM 15 and PDM 25, as well as the hole size and spacing for mounting. The PDM should be mounted in a well-ventilated area to cool the unit when under heavy load. Care should be taken to mount the PDM in a location which minimises the exposure to water and dirt. In most cases, mounting the PDM next to the ECU is the most efficient method, as the wires running between both units are kept to a minimum to save on weight. Around 100mm of clearance should be maintained both on top and in front of the PDM to allow room for the wiring loom and USB to be connected without excess strain.



Wiring

Pinout

The connectors used in the PDM 15/25 are 12-pin Deutsch DT series connectors. There are four connectors used, each with a different colour and mating key. This means that each connector only inserts into one mating socket, eliminating the risk of plugging a connector into the wrong socket and causing damage. Care must be taken when wiring the PDM to ensure that the correct connector is used. The connectors, wedge locks and pins are included with each PDM. **A Pin out table for the PDM is given below.**



PDM Version	Number	Description
15/25	A1	Input 12
15/25	A2	Input 10
15/25	A3	Input 8
15/25	A4	Input 6
15/25	A5	Input 4
15/25	A6	Input 2
15/25	A7	Input 1
15/25	A8	Input 3
15/25	A9	Input 5
15/25	A10	Input 7
15/25	A11	Input 9
15/25	A12	Input 11
15/25	C1	Input 16
15/25	C2	Input 15
15/25	C3	Ignition Input (switch to 12V)
15/25	C4	Output 3
15/25	C5	Output 2
15/25	C6	Output 1
15/25	C7	Ground (Connect to chassis)
15/25	C8	CAN High
15/25	C9	CAN Low
15/25	C10	5V Output
15/25	C11	Input 13
15/25	C12	Input 14
25	D1	Output 17
25	D2	Output 21
25	D3	Output 24
15/25	D4	Output 9
15/25	D5	Output 8
15/25	D6	Output 7
15/25	D7	Output 4
15/25	D8	Output 5
15/25	D9	Output 6
25	D10	Output 24
25	D11	Output 20
25	D12	Output 16
25	B1	Output 19
25	B2	Output 23
25	B3	Output 25
15/25	B4	Output 15
15/25	B5	Output 14
15/25	B6	Output 13
15/25	B7	Output 10
15/25	B8	Output 11
15/25	B9	Output 12
25	B10	Output 25
25	B11	Output 22
25	B12	Output 18

Battery Positive

The Hardwire Electronics PDM Powers the connected devices via the aluminium M8 stud protruding from the top of the enclosure. The PDM current can peak at upwards of 180 Amps, therefore an appropriate gauge copper wire must be used to attach to the M8 stud to avoid excess heat and voltage drop. It is recommended to use at least 2 AWG wire with the supplied ring terminal. Care must be taken when tightening the retaining nut on the stud. The recommended torque to use is 8Nm – exceeding this could damage the internal circuit board.

Battery Negative

Under normal operating conditions, the current flow through the ground pin of the PDM is low. However, when a load dump occurs, the current flow can be much higher. Therefore, it is recommended that at least 20 AWG wire be used.

5V Output

One pin (C10) on the PDM 15/25 is a dedicated 5V output. The 5V output is over current and short circuit protected. No more than 0.5 Amp should be drawn from the 5V output to avoid excessive heating of the internal circuitry.

Ignition Input

To power on the PDM, the Ignition Input must be switched to the battery positive voltage. This input can be wired directly to the accessory input of an ignition key barrel, or alternatively to another dedicated switch. The ignition input may be connected directly to the battery positive voltage - however, this is not advised as the battery will slowly discharge over time when the vehicle is not in use. A battery isolator may be used to combat this problem. When using the PDM, make sure to always power the PDM from the ignition input first before connecting the USB power.

Inputs

The inputs to the Hardwire Electronics PDM operate as analogue inputs, which measure the voltage and compare it against a user configured threshold value. This makes it possible to use both battery-connected and ground-connected switches, as well as raw analogue voltage inputs. The switch inputs should be configured as shown in the system diagram. preferably in the battery connected configuration. Multi-position switches may also be used, whereby each position of the switch outputs a different voltage (see wiper motor section). This technique can be used to expand the inputs of the PDM. The different voltage levels can be used to turn on/off different outputs via the output function configuration.

Outputs

All outputs on the PDM are high sided, meaning that when enabled, they output the positive battery voltage present on the M8 stud. Each output has current sensing, thermal overload protection, static discharge protection, short circuit protection and over/under voltage protection. Each output on the PDM 15/25 is rated to a maximum of 20 Amps steady state, with peak ratings of >80 Amps. The limitation is set by the pin contacts of the connectors used. If a higher current capacity is needed, two or more outputs can be connected in parallel. An appropriate wire gauge must be used for each output, corresponding to the maximum steady state current draw of the connected device. When powering inductive loads, the maximum inductance permissible is 0.1mH at 20 Amps, 1mH at 10 Amps, and 1H at 1Amp. For highly inductive loads, it is recommended to install a diode from the PDM output to the 12V battery terminal, so that large back voltages are discharged effectively.

Installation

CAN
The PDM15/25 come with a CAN2.0A/B compliant controller. For the CAN bus to work correctly, each end of the bus must be terminated with a 120Ω resistor for a combined 60Ω of resistance between the CAN high and CAN low data lines. The PDM does NOT come with the resistor installed, so that it can be quickly added into a pre-existing system without issue. The PDM can operate at a range of bus speeds (see specification).

PDM Configuration Software

The Hardwire Electronics PDM configuration software can be downloaded from the Hardwire Electronics website in the downloads section. The software runs on Windows 10 machines only. Simply download the file and run the setup executable. Follow the instructions on screen to install the software to your PC.

The Hardwire Electronics PDM software has been designed with ease of use in mind. There are four sections of the program – Connection, Configuration, Monitor and Update. The user can navigate through the program by clicking on the tabs in the left side banner. Below these tabs is important data about the PDM when it is connected. This information can be used to check if the PDM is functioning correctly.

Connection

Connection - PDM to PC connection

The PDM communicates with the PC via USB. The PDM can be powered up by the USB connection to configure the PDM away from the car, or when the car is turned off. WARNING Extreme care should be taken when the PDM is connected to a PC and the car is running. When the PDM is powered from both the USB and the ignition input, when the igntion input is turned off, the PDM will remain switched on. This means that your outputs will also stay on.

To initiate communication between the PDM and host PC, start the Hardwire Electronics PDM configuration software and navigate to the Connection tab. Press the button 'Search for devices' to find the PDM. Make sure that no other devices are connected to the PC at this time. Now select the PDM from the drop-down menu and click 'Connect.' If a successful connection is established, the device name will be listed below. A connection must be established to configure and monitor the PDM.

Configuration

Configuration files

The configuration settings for the PDM are held in a configuration file which is sent to the PDM via USB. The configuration file holds the user defined values which describe how the inputs, outputs, CAN communication and data logging operates. Navigate to the 'Configuration' tab to change the configuration of the PDM.

Sending

Pressing send will write the current configuration data on the screen into the PDM, overwriting the configuration data currently being held in the PDM. Once the configuration has been sent, the software will display a notification stating that the configuration has sent successfully.

Retrieving

The configuration data currently being held in the PDM can be accessed by pressing the retrieve button. If a difference is detected between the current PDM configuration settings and the on-screen configuration settings, then a warning message will be displayed. The user can decide to keep the current configuration data or overwrite it with the data on the PDM.

Loading

Configuration data can be saved to a file which can then be shared between other PDMs. Pressing the 'Load' button will prompt the user to navigate to the appropriate configuration file. The configuration file will then be loaded into the program and displayed on screen.

Saving

Much like the process of loading external configuration files, the user may also save the current configuration data into a file. Pressing the 'Save' button will prompt the user to locate a directory in which to save the configuration file.

Inputs

Threshold Voltage

Each input on the Hardwire Electronics PDM operates as an analogue input. Inputs are active whenever the voltage present on the input passes a threshold value which is configured in software.

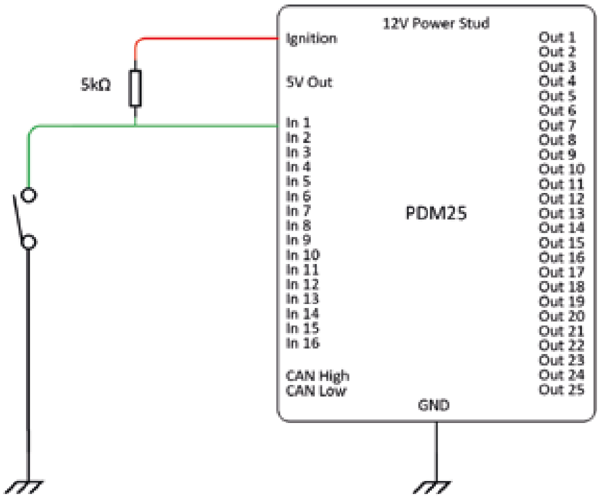
Active Level

Each input can be programmed as active-high or active-low. Active-high causes the input to enable when the voltage rises above the threshold, whereas active-low causes the input to enable when the voltage falls below the threshold. For a simple battery connected switch, the threshold value should be set around 4V for reliable triggering of the input.

Pull-Up and Pull-Down resistors when using switches as inputs

All Hardwire Electronics PDM15/25s manufactured after February of 2021 have weak pull up and pull down resistors installed already on each input. The presence of the resistors keep the input voltage at approximately 3V when the input is left floating. This means that either ground switching or 12V switching can be used without any extra components required.

PDMs manufactured prior to February 2021 only have pull down resistors installed internally. This means that when using ground switching on inputs, you will need to install a pull up resistor. The diagram below details how to install the resistor on an input. If 12V switching is used, then no resistors will be required.



Hysteresis Voltage

A hysteresis voltage can be set, such that after an input is enabled, it does not disable again until the voltage reduces below the set point plus the hysteresis value. This helps to reduce the effect of voltage noise problems falsely triggering an input and may be used in conjunction with a thermo-sensor and radiator fan to stop fan flickering.

Input Mode

The inputs on the PDM may be set to momentary or latched mode. This feature makes it possible to use a simple push button switch to activate components such as fans and switches with one button press to turn the component on and off.

Outputs

The outputs on the Hardwire Electronics PDM are all high-sided. When an output is enabled, it will output the voltage present on the main power input stud (Bat +). This allows the outputs of the PDM to directly drive most electrical components in a vehicle.

Trip Mode

Instant Trip: This mode will instantly cut power on an output once the current threshold is exceeded. This mode is suggested for driving sensitive components such as an ECU, which would need instant protection in case of overcurrent.
Normal Trip: This mode acts more like a conventional fuse and takes a time average of the current flow on the output, allowing for small current spikes above the current threshold without tripping. This mode is useful for driving components such as ignition coils, which draw current in short spikes as the coils charge. In this scenario, the instantaneous current may exceed the threshold value, but the average current still remains below the threshold value.

Low Fuse

The low fuse setting will trip if the current falls below the threshold for over 1 second continuously.
If this functionality is not needed then the value should be set to 0.

High Fuse

The high fuse setting comes into effect after the peak fuse time has elapsed, and sets the current limit for an output working at steady state. If the current exceeds this value, then the output will trip.

Peak Fuse & Peak Fuse Time

The peak fuse setting determines the maximum current that an output can provide before it trips in the initial turn-on phase of the output. The length of the turn-on phase is determined by the peak fuse time setting. The peak fuse setting is used to allow large inrush currents without tripping the output. When turning on loads such as radiator fans or fuel pumps, the initial current draw from the device can often double or triple the steady state current draw.

Stay On Time

After an output is commanded to turn off, the output will remain switched on for the stay on time, before turning off.

Retries

If an output trips due to an over-current condition, then that specific output may me 'retried' several times to attempt to establish normal operating conditions. The output channel will be retried for the number of times specified by this setting. A value of zero will result in continual retries of that output

Clear Time

If an output trips, the output will not be retried until the 'clear time' has passed. This is to allow time for an output device to perhaps cool down or reset itself before being turned back on.

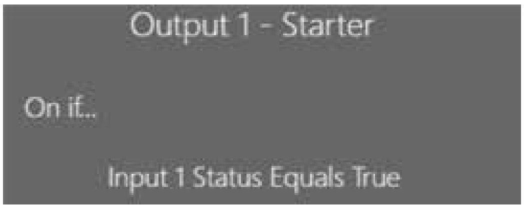
Functions

Functions are tools that can be used to create rules which control the outputs of the PDM. Each output has its own unique function. Each function will evaluate to either a logical TRUE (1) or FALSE (0). These functions can then be used to turn on or off the PDM outputs. If the function evaluates to TRUE, then the output will turn on. If the function evaluates to FALSE, then the output will turn off.
For example - if output 1 was powering a radiator fan, the output function might look something like this - Output 1 'On If CAN Input 1 (Coolant Temp) > 85.' This would turn the radiator fan on if the coolant temperature which is received on a CAN input is greater than 85, More examples are shown later in this document.

Below is a table showing the conditions which can be used in output functions to compare values.

Boolean Values	
True	Logical Value of 1 (ON)
False	Logical Value of 0 (OFF)
Logic Operations	
AND	Returns True (1) if both of the two values being compared are True (1). Returns False (0) otherwise
OR	Returns True (1) if either of the two values being compared are True (1), or they are both True (1). Returns False (0) otherwise
XOR (Exclusive OR)	Returns True (1) if either of the two values being compared are True (1). Returns False (0) otherwise
NOR (Not OR)	Returns True (1) if both of the two values being compared are False (0). Returns False (0) otherwise
NAND (Not AND)	Returns False (0) if both of the two values being compared are True (1). Returns True (1) otherwise
Comparison Operations	
Equals	Returns True (1) if a value is equal to a Constant. Returns False (0) otherwise
Not Equal	Returns True (1) if a value is not equal to a Constant. Returns False (0) otherwise
>	Returns True (1) if a value is greater than a Constant. Returns False (0) otherwise
>=	Returns True (1) if a value is greater than or equal to a Constant. Returns False (0) otherwise
<	Returns True (1) if a value is less than a Constant. Returns False (0) otherwise
<=	Returns True (1) if a value is less than or equal to a Constant. Returns False (0) otherwise

Examples of basic Output Functions:
This example demonstrates sending power to a starter motor when a button is pressed.
In this case, input 1 is the status of a switch input which is active high. This means when the switch is ON (Equals True), the output will be on.

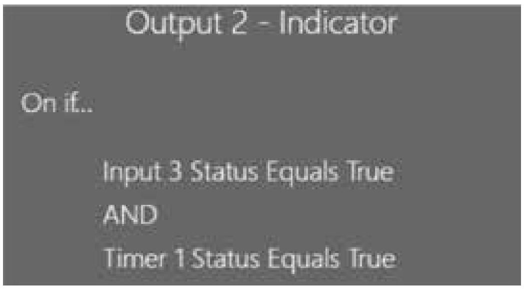


Output 1 - Starter

On if...

Input 1 Status Equals True

This example shows turning on an indicator when a switch is pressed and a timer within the PDM is high, this will cause the output to pulse on and off in time with the timer whenever the switch is enabled.
In this case, input 3 is the status of a switch which is active high. This means when the switch is ON (Equals True) AND the signal from the timer is ON (Equals True), then the output will be on.

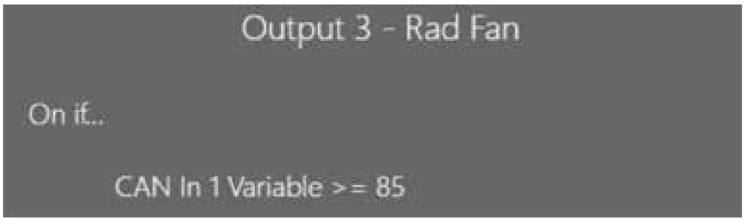


Output 2 - Indicator

On if...

Input 3 Status Equals True
AND
Timer 1 Status Equals True

This example shows sending power to a radiator fan when a variable received on the can bus is above a certain value.
In this case CAN In 1 Variable is the coolant temperature in °C, received over the CAN bus. When this variable is greater than or equal to 85, the function will evaluate to True (1) and the output will turn on.



Output 3 - Rad Fan

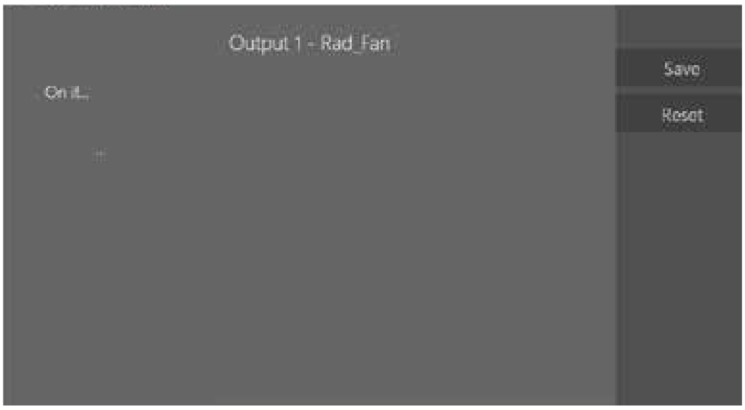
On if...

CAN In 1 Variable >= 85

Output Function
The operation of each output is determined by its own unique function. The output function is a powerful mechanism to determine if an output should be turned on or off. One may create the output function arbitrarily, based on the logical combinations of different PDM variables or constants. The following section will demonstrate how to construct output functions for different applications.

Example 1: Simple Logic Function
In this example, a function will be constructed to turn on output 1 when both input 1 is active AND the PDM temperature is less than 70°C.

Navigate to the configuration tab and click on outputs. Now click on edit configuration for an output. This will open a new window with the configuration settings for that output. Now click on the edit function button. A new window will pop up. At the top of the window, the currently selected output will be displayed. The space below this heading is where the function will be constructed. An output will 'turn on if' the output function evaluates to 'true' or '1.' Pressing the three small dots will bring up a further window.



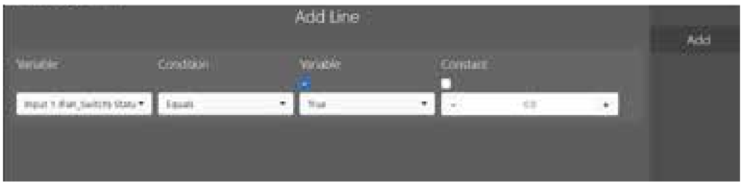
Output 1 - Rad_Fan

On if...

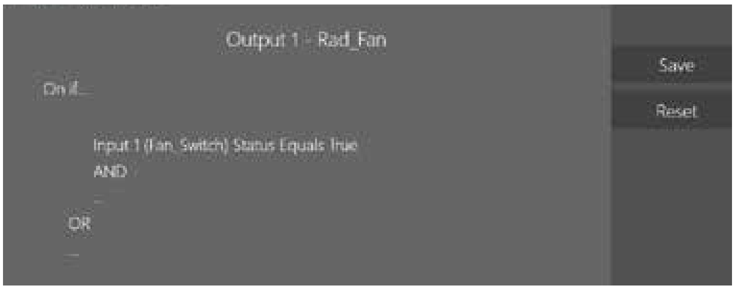
Continued on next page...

Configuration

Here the user can construct a line of the output function. A function line is made up of the logical combination of a PDM variable and another variable or constant. In this demonstration, the output needs to turn on if Input 1 is active. Therefore, the first line of the function can be constructed as shown in the adjacent diagram making sure to tick the 'variable' check box.



Clicking the 'add' button will insert this line into the output function and close the window. The output function will now look like this. At this stage, the output function will evaluate to true (turn on) if input 1 equals true. There is now the option to expand this function to add more conditions. Another function line can be inserted by pressing the three dots below the AND or the three dots below the OR. In this demonstration, the output needs to come on if input 1 is active AND the PDM temperature is below 70°C – therefore the three dots below the AND should be clicked.



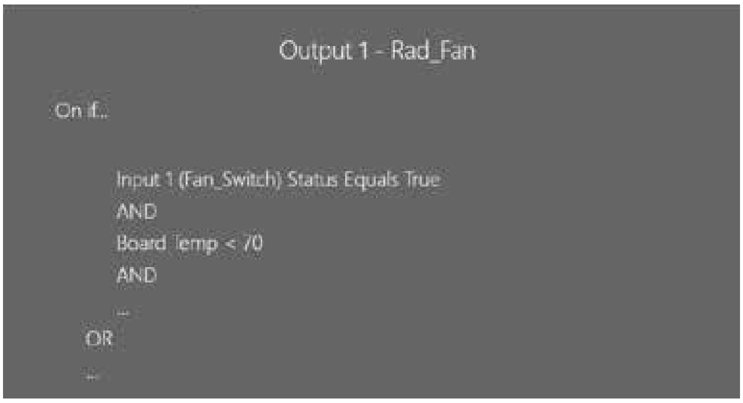
Output 1 - Rad_Fan

On if...

Input 1 (Fan_Switch) Status Equals True
AND
...
OR
...

Another line can be constructed as shown before and added to the output function. The resulting output function will look like this.

Clicking the save button to the right side of the window will save the function and close the window.



Output 1 - Rad_Fan

On if...

Input 1 (Fan_Switch) Status Equals True
AND
Board Temp < 70
AND
...
OR
...

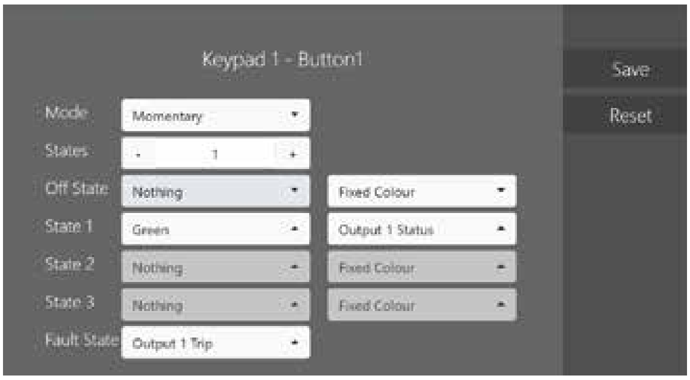
Continued on next page...

Example 2: CAN-Keypad

In this example, a function will be constructed to turn on Output 1 when a button is pressed on a compatible CAN-Keypad.

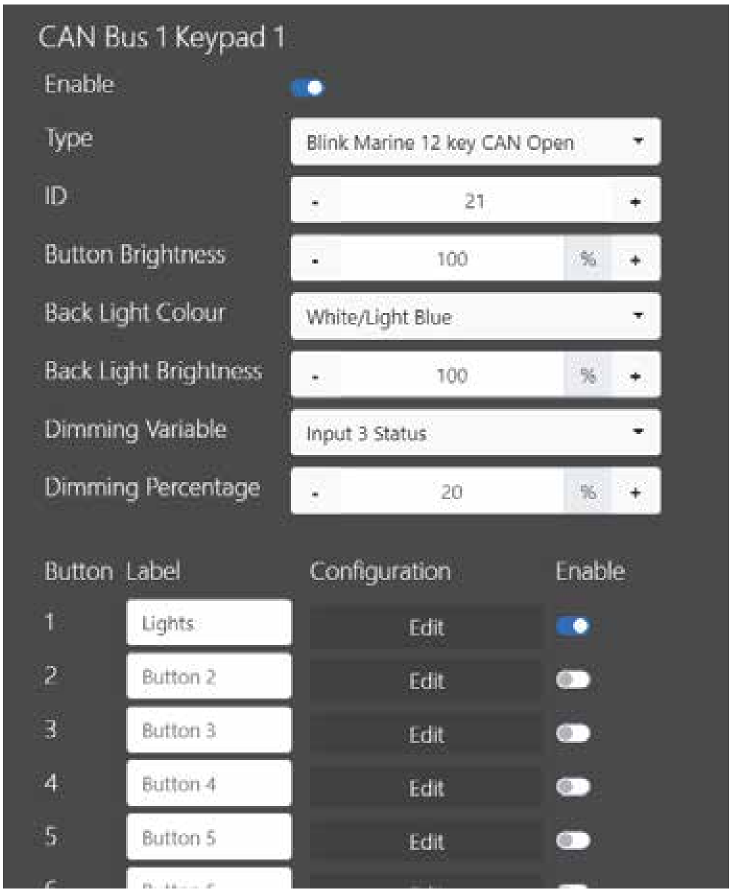
First, ensure that a CAN-Keypad is connected and correctly configured (see CAN-Keypad section). Once connected and configured, the keypad switch inputs can be used in an output function in exactly the same way as a physical input.

The keypad has been configured to work in a momentary fashion, when the button is pressed the button state is On/True, and when the button is released the button state is Off/False. The keypad button is also configured to turn green when the button is pressed to indicate to the user that the keypad button is in the On/True state.



Keypad 1 - Button1

Mode	Momentary	
States	1	
Off State	Nothing	Fixed Colour
State 1	Green	Output 1 Status
State 2	Nothing	Fixed Colour
State 3	Nothing	Fixed Colour
Fault State	Output 1 Trip	

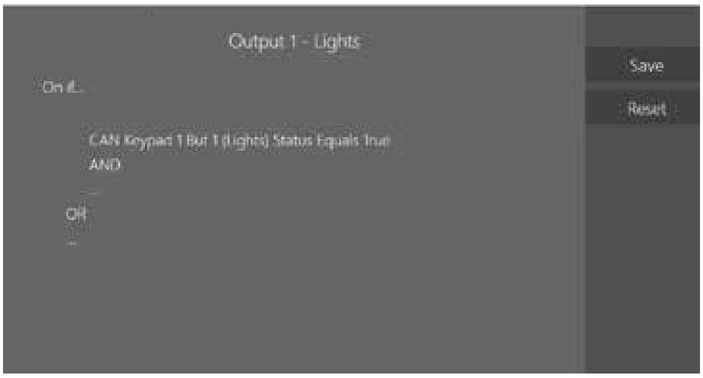


CAN Bus 1 Keypad 1

Enable	●
Type	Blink Marine 12 key CAN Open
ID	21
Button Brightness	100 %
Back Light Colour	White/Light Blue
Back Light Brightness	100 %
Dimming Variable	Input 3 Status
Dimming Percentage	20 %

Button	Label	Configuration	Enable
1	Lights	Edit	●
2	Button 2	Edit	○
3	Button 3	Edit	○
4	Button 4	Edit	○
5	Button 5	Edit	○

The state of the button can then be used in an output function as demonstrated in the adjacent image. Output 1 will only be on if CAN keypad button 1 is pressed and in the 'True (1)' state, otherwise it will be off.

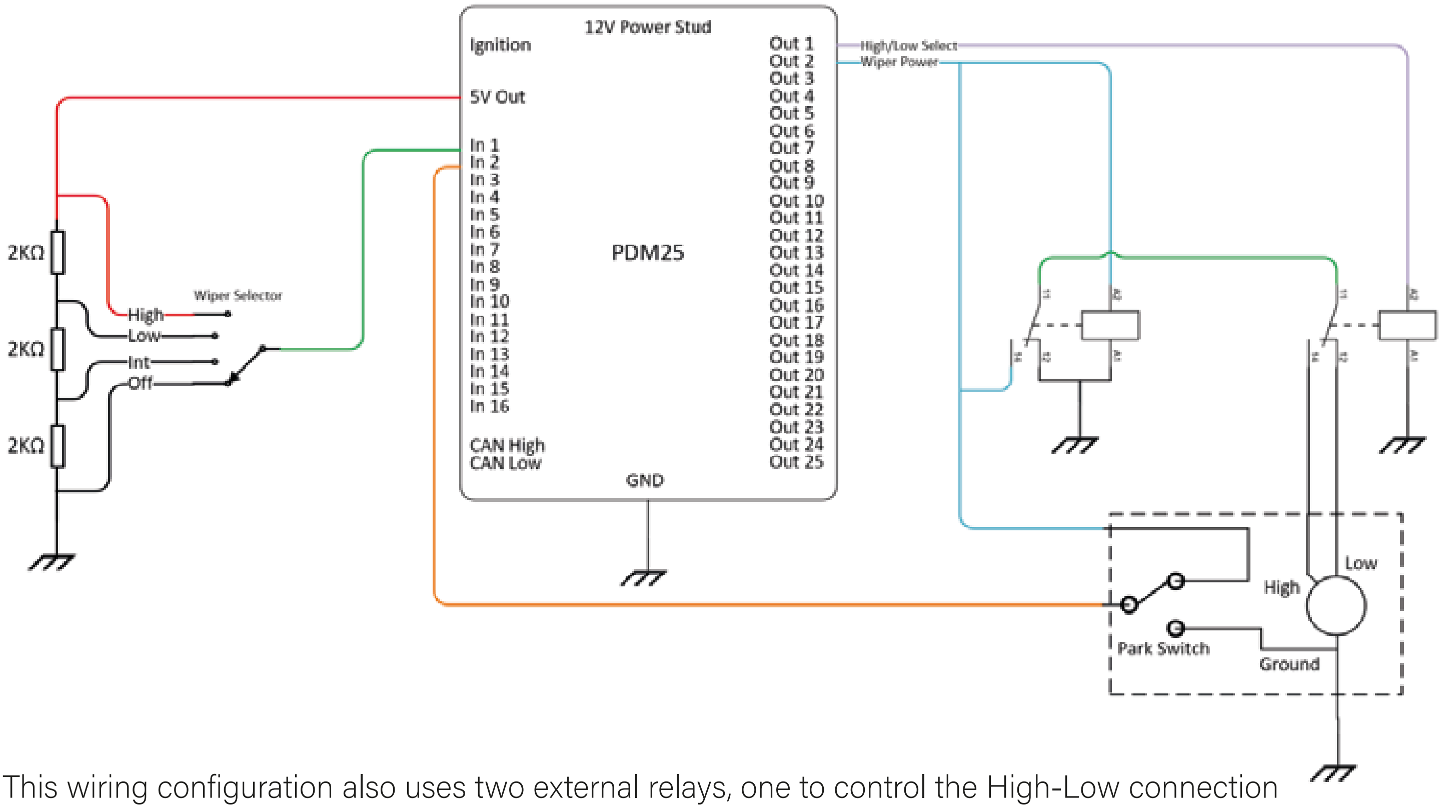


Output 1 - Lights

```
On if...
    CAN Keypad 1 But 1 (Lights) Status Equals True
    AND
    ...
  OR
  ...
```

Example 3: Wiper Motor Control

In this example, output functions will be used to control wiper motors. The functions and wiring configuration will allow for High, Low and Intermittent wipers, as well as wiper motor park.
First, the motor should be wired up as demonstrated in the diagram below. This configuration will use two inputs and two outputs for control of the wipers.



This wiring configuration also uses two external relays, one to control the High-Low connection to the motor and a second to control power to the motor.

Understanding how this configuration works is helpful when following the section on setting up the output functions for wiper control. The basic overview of this configuration is explained below:

Input 1: This input will read different voltages depending on which position the switch is in (the switch is usually on a stalk on the steering column). This input is used to select low, high and intermittent settings for the motor.

Input 2: This input is connected to the park switch, which is high throughout the majority of the motor's rotation and low when it is in the park location. This input is used to detect when the PDM should brake the wiper motor and park it.

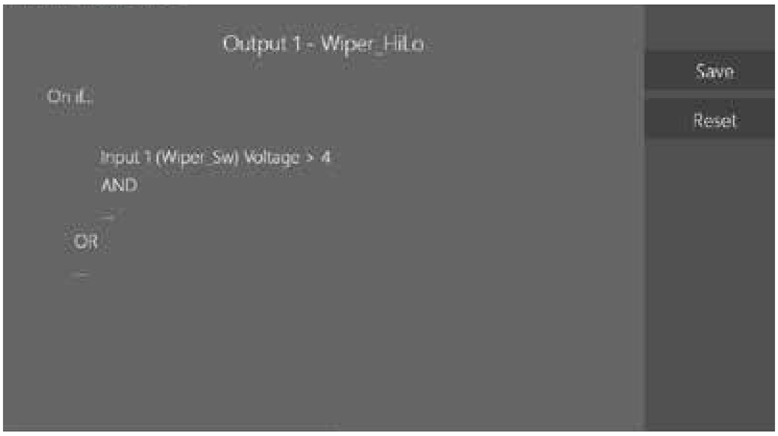
Output 1: This output is connected to a relay which is in turn connected to the high and low speed connections on the motor. The normally closed connection of the relay is connected to the motor low and the normally open is connected to the motor high speed connection. Therefore when Output 1 is Low, power will be sent to the Low speed motor connection and when it is High, power will be sent to the motor High speed connection.

Output 2: This output is connected to both the park switch and the relay used to send power to the motor. The normally closed connection of this relay is connected to ground and the common is connected to the other relay, meaning when the output is low, both sides of the wiper motor will be grounded and it will not turn. The normally open connection is connected to the PDM output, meaning when the output is High, power will be sent to the motor and it will turn.

Continued on next page...

The output functions to control the wipers are shown below.

This is the output function for output 1 which will send output power to the relay when the switch on input 1 is in the HIGH position meaning the relay is energised and power will go to the High speed motor connection. If this switch is in any other position, the output is low and power will be connected to the Low speed motor connection.



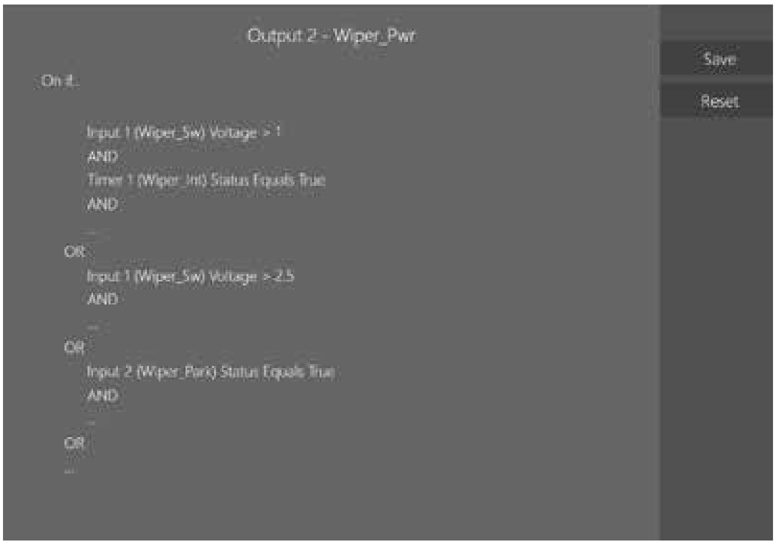
Output 1 - Wiper_HiLo

```
On if...
      Input 1 (Wiper_Sw) Voltage > 4
      AND
      ...
   OR
   ...
```

This is the output function for output 2.
The first section is for the condition when the switch on input 1 is in the Intermittent position, this will power the wiper motor whenever a timer set by the user is high. (see Timer Configuration)

The second section will provide power on output 2 when the switch on input 1 is in the Low and High positions, allowing continuous power to be sent to the wiper motor.

The third section will provide power on output 2 when the park switch on input 1 is in the High position (wiper not home), allowing power to be sent to the wiper motor until it has reached the home position. When the wiper reaches the home position, the park switch goes low and power to the wiper is removed. When the park switch is connected to ground (i.e the motor is in the park position) and the switch on input 1 is in the Off position, the wiper power relay will turn off, meaning both sides of the motor will be grounded, which will brake the motor and bring it to a rest.

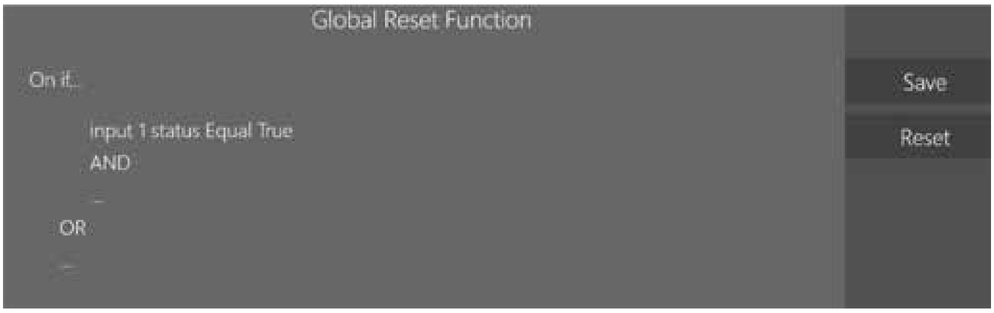


Output 2 - Wiper_Pwr

```
On if...
      Input 1 (Wiper_Sw) Voltage > 1
      AND
      Timer 1 (Wiper_Int) Status Equals True
      AND
      ...
   OR
      Input 1 (Wiper_Sw) Voltage > 2.5
      AND
      ...
   OR
      Input 2 (Wiper_Park) Status Equals True
      AND
      ...
   OR
   ...
```

Global Reset

Global reset functionality allows all outputs that have been tripped to be reset at once, the example below shows the use of a physical input being used to reset all tripped outputs. In this example, when the status of input 1 equals true (e.g if a switch is pressed), then all of the tripped PDM outputs will be reset. The global reset function works in an identical fashion to the output functions. (see outputs configuration)



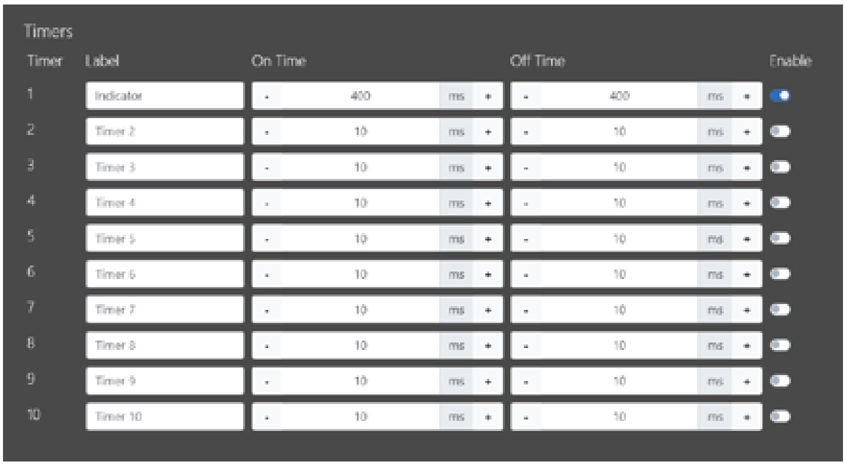
Global Reset Function

```
On if...
      input 1 status Equal True
      AND
      ...
   OR
   ...
```

Timer

The PDM 15/25 have 10 customisable timers which can be used in conjunction with the output functions to achieve advanced PDM functionality – such as flashing indicators, intermittent wiper motors and more.

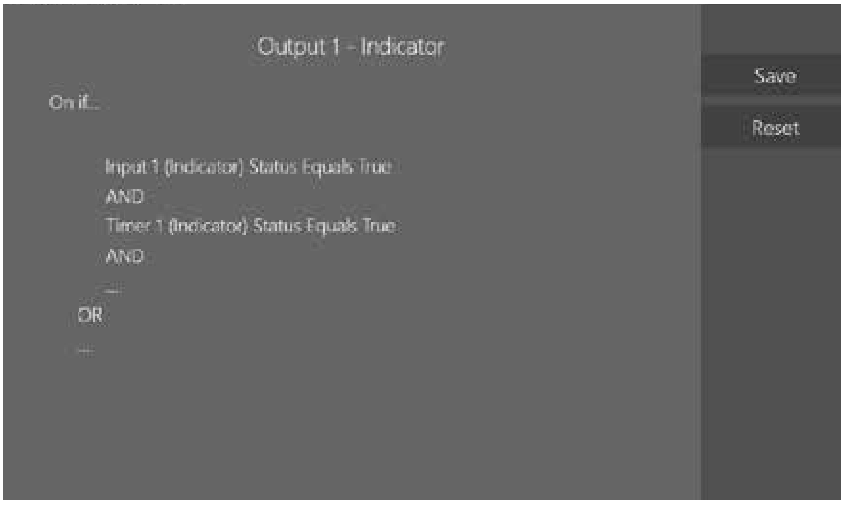
Each timer has an ON and OFF time, given in milliseconds. When a timer is enabled, it will run automatically in the background, switching state from ON to OFF at the given intervals. It is up to the user to use the state of the timer in the output function. Here is a demonstration of a typical set up for an indicator in the PDM. Timer 1 has both the ON and OFF time set to 400ms, resulting in a timer switching frequency of 1.25Hz – or 75 flashes per minute.



Timers

Timer	Label	On Time	Off Time	Enable
1	Indicator	- 400 ms +	- 400 ms +	●
2	Timer 2	- 10 ms +	- 10 ms +	○
3	Timer 3	- 10 ms +	- 10 ms +	○
4	Timer 4	- 10 ms +	- 10 ms +	○
5	Timer 5	- 10 ms +	- 10 ms +	○
6	Timer 6	- 10 ms +	- 10 ms +	○
7	Timer 7	- 10 ms +	- 10 ms +	○
8	Timer 8	- 10 ms +	- 10 ms +	○
9	Timer 9	- 10 ms +	- 10 ms +	○
10	Timer 10	- 10 ms +	- 10 ms +	○

The timer is now used in the Output 1 function. In this example, Input 1 is used as an indicator switch. Output 1 is on if – input 1 (indicator switch) is True, AND, Timer 1 status is True. Because the timer status changes every 400ms, the result is that the indicator output also changes every 400ms (The output flashes at 1.25Hz). This method of using a timer in the logic function can be used for other applications in a similar manner.



Output 1 - Indicator

On if...

Input 1 (Indicator) Status Equals True
AND
Timer 1 (Indicator) Status Equals True
AND
...
OR
...

PWM and Soft Start Configuration

WARNING

Each output on the PDM15/25 has PWM and Soft Start capability. Using PWM or Soft Start on high power outputs such as large radiator fans or fuel pumps puts added pressure on the internal circuitry, creating substantial heat. The higher the PWM frequency, the more heat will be created. For outputs which draw more than 10Amps under normal operating conditions, using PWM is not recommended on single outputs. Multiple outputs can be joined together to share the load and ensure long term reliability of the PDM.

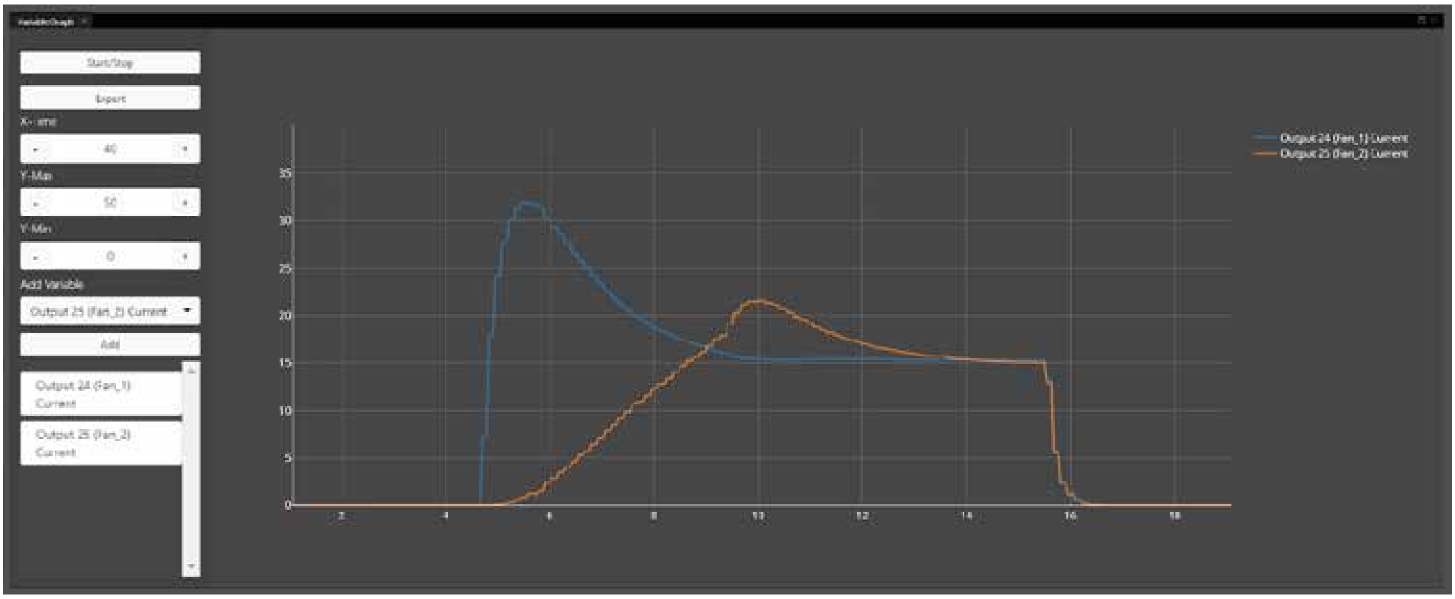
PDMs with a date code prior to February 2021, will need to have extra componentry to drive loads with PWM enabled. Back voltage will cause current measurement errors in the output which will cause it to trip. A series diode is needed to block the reverse voltage on the output. For higher current applications, an ideal diode will be needed for reduced heat dissipation.

PDMs with a date code post February 2021 do not need any extra components for driving outputs with PWM enabled.

Soft Start

Soft start is used to gradually turn on an output to reduce the high initial current draw. This is achieved by rapidly pulsing the output on and off while increasing the duty cycle from 0% up to 100% over a period of time set in the output configuration.

Below is a log of two radiator fan outputs. Output 24 (Blue) does not have soft start enabled, causing a large initial current draw when the output is turned on. The initial high current draw is caused by the inertia of the radiator fan as it comes up to speed. Output 25 (Orange) does have soft start enabled with a soft start time of 5 seconds. As can be seen from the graph, the current draw slowly ramps up over time, eliminating the high current spike present on Output 24.



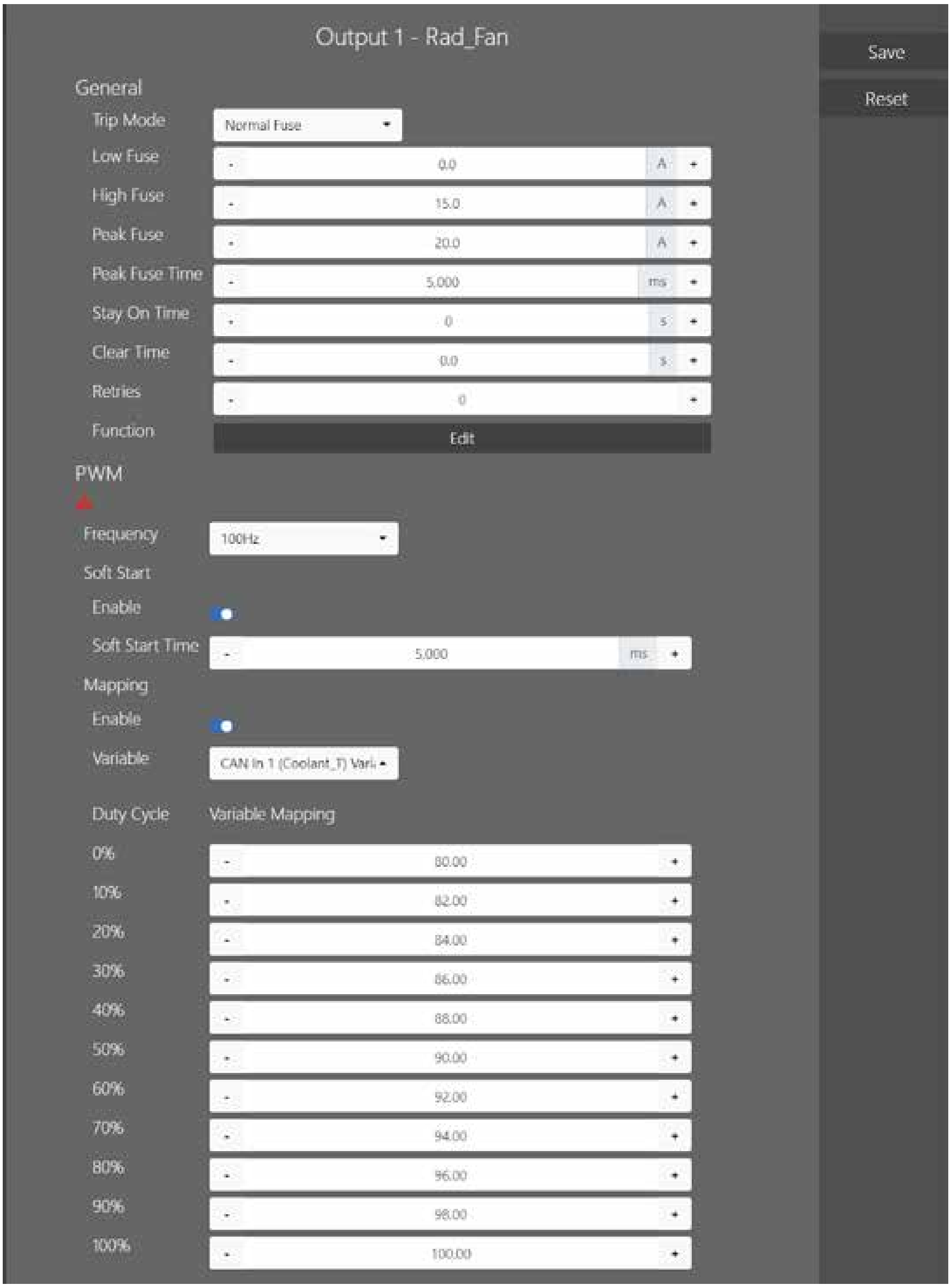
VariableGraph
Start/Stop
Export
X-Limit
40
Y-Max
50
Y-Min
0
Add Variable
Output 25 (Fan_2) Current
Add
Output 24 (Fan_1) Current
Output 25 (Fan_2) Current
35
30
25
20
15
10
5
0
2
4
6
8
10
12
14
16
18
Output 24 (Fan_1) Current
Output 25 (Fan_2) Current

PWM

PWM is used to vary the average voltage present on an output, by rapidly turning the output on and off. The ratio of the 'on' to 'off' time (duty cycle) is what dictates the average voltage on the output , and hence the current draw of the output. For example, this feature is useful for varying the speed of a radiator fan, or fuel pump, or even the brightness of headlights.

Below is a configuration for a typical radiator fan output. PWM is used to increase the speed of the fan as the engine coolant temperature increases. The engine coolant temperature data is received via the CAN bus. By selecting CAN input 1 (Coolant Temperature) input as the PWM variable, the output duty cycle can be mapped to different values for the Coolant temperature. In this example, a coolant temperature below 80 results in a 0% duty cycle, so the output is turned off. As the Coolant temperature increases, the duty cycle also increases until reaching 100% duty cycle at 100 degrees.

This method can be used with CAN inputs, Input voltages or Maths channels as the PWM variable.



Output 1 - Rad_Fan

General

Trip Mode: Normal Fuse

Low Fuse: 0.0 A

High Fuse: 15.0 A

Peak Fuse: 20.0 A

Peak Fuse Time: 5.000 ms

Stay On Time: 0 s

Clear Time: 0.0 s

Retries: 0

Function

PWM

Frequency: 100Hz

Soft Start

Enable

Soft Start Time: 5.000 ms

Mapping

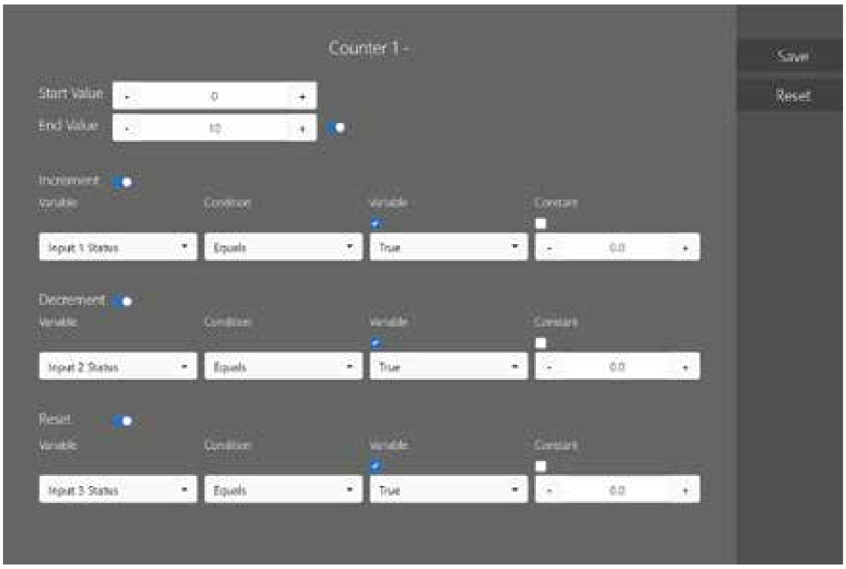
Enable

Variable: CAN In 1 (Coolant_T) Vari...

Duty Cycle	Variable Mapping
0%	80.00
10%	82.00
20%	84.00
30%	86.00
40%	88.00
50%	90.00
60%	92.00
70%	94.00
80%	96.00
90%	98.00
100%	100.00

Counter Configuration

Counters are variables which can be incremented or decremented based on certain events such as button presses. Counters can be used for more advanced functionality, such as incrementing a number every time a button is pressed, and then using the value of the counter to produce different behaviour on the PDM outputs. To configure a counter, go to the Configuration tab and click on counters. Here you can click on the Edit button to configure a counter.


Counter 1 -

Start Value: 0
End Value: 10

Increment
Variable	Condition	Variable	Constant
Input 1 Status	Equals	True	0.0

Decrement
Variable	Condition	Variable	Constant
Input 2 Status	Equals	True	0.0

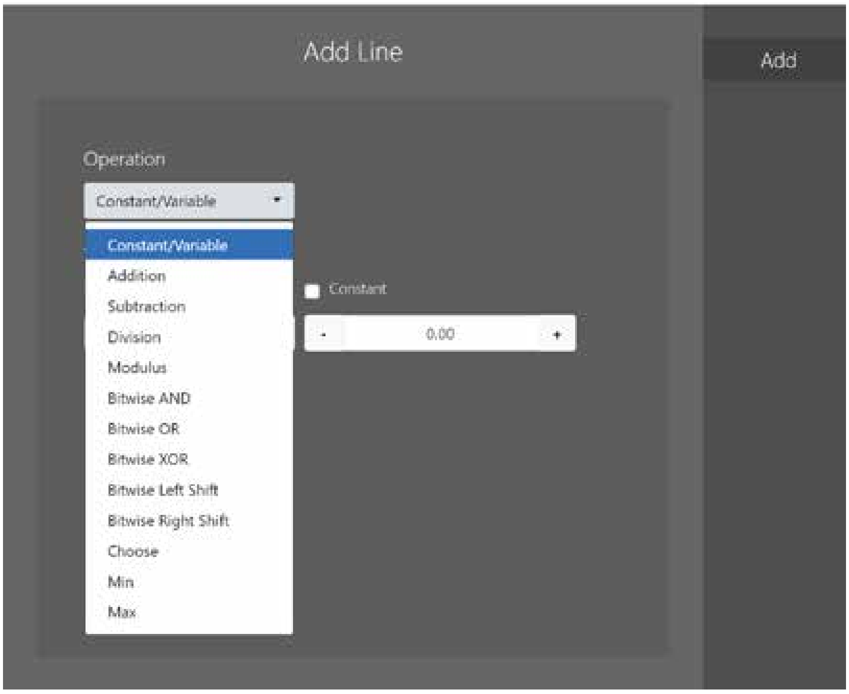
Reset
Variable	Condition	Variable	Constant
Input 3 Status	Equals	True	0.0

Here is an example of a very simple counter used to increment a number if Input 1 is True, or decrement the number if Input 2 is true. The start value of the counter is the value the counter takes when the PDM first starts up, or when the counter is reset. The end value of the counter is used as a limit for the counter value. When the counter increments past the end value, the counter will reset to the start value. If the end value is disabled, then the counter will increment forever. The counter can be incremented, decremented, or reset by three independent conditions. The conditions are the same as found in the output function lines. Depending on which condition evaluates to True, the counter will increment, decrement or reset itself. The PDM counters can count between a value of -2147483647 and +2147483647.

Maths Channel Configuration

Maths channels are an advanced PDM feature which allow the user to create their own mathematical equations which are evaluated by the PDM. The value of the equations can then be used by the PDM for controlling the behaviour of PDM outputs.

To configure a Maths channel, go to the Configuration tab and then to the Maths channel tab. Click on the Edit button to configure a Maths Channel.


Add Line

Operation: Constant/Variable

□ Constant 0.00

The structure of the mathematical equation is similar to that of the logic functions. It is important to be aware of 'order of operations' when constructing an equation. The structure of the equation helps to avoid order of operation problems. Values are converted to floating point when used in the mathematical equations.

For Bitwise operations, the operands will be converted to unsigned 32-bit numbers. New equation lines can be added to the equation by double clicking on the three dots. Clicking the three dots will bring up a new window where a line of the equation can be made. **There are a number of mathematical operations that can be selected from the Operation drop down list, shown in the diagram to the right.**

Continued on next page...

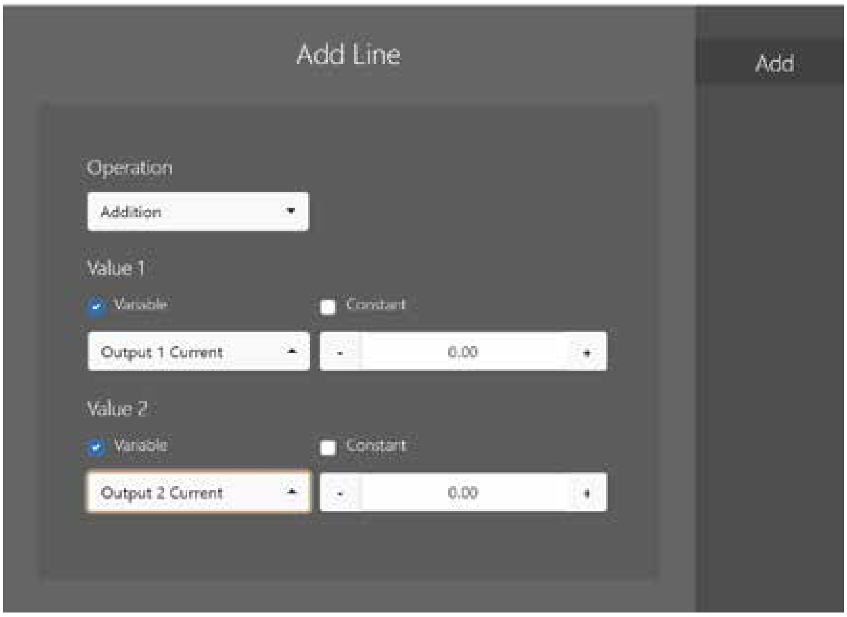
Maths Channel Configuration - *continued*

Below is a table which details the function of each operation which can be selected.

Mathmatical Operators	
Constant/ Variable	**A constant** is just a number which is fixed, and can be used in the mathematical equation. **A variable** is a number which changes, and can be selected from the list of variables inside the PDM i.e Output 1 Current, Input 1 Voltage.
Addition	The Addition operator adds the two operands together. Each operand can be either a constant number or a PDM variable.
Subtraction	The Subtraction operator subtracts the second operand from the first. Each operand can be either a constant number or a PDM variable.
Division	The Division operator divides the equation section by the divisor. The divisor can be either a constant number or a PDM variable.
Modulus	The Modulus operator performs the modulo operation on the equation with the divisor. This gives the remainder when the equation section is divided by the divisor. The divisor can be either a constant number or a PDM variable.
Bitwise Operators	
AND	Returns the bitwise AND operation of the two operands.
OR	Returns the bitwise OR operation of the two operands.
XOR (Exclusive OR)	Returns the bitwise XOR operation of the two operands.
Left Shift <<	Shifts the equation section to the left by between 0-32.
Right Shift >>	Shifts the equation section to the right by between 0-32.
Function Operations	
Choose	If the condition is true, the Choose function will return the first value. If the condition is false, the Choose function will return the second value.
Min	Returns the smallest of the two values.
Max	Returns the largest of the two values.

Continued on next page...

As an example, here is how one would configure a maths channel to add together two output currents. This would be useful if two PDM outputs are sharing a load, and the total current draw of the load is needed. Using the Addition operation, the two output currents can be added together.


Add Line

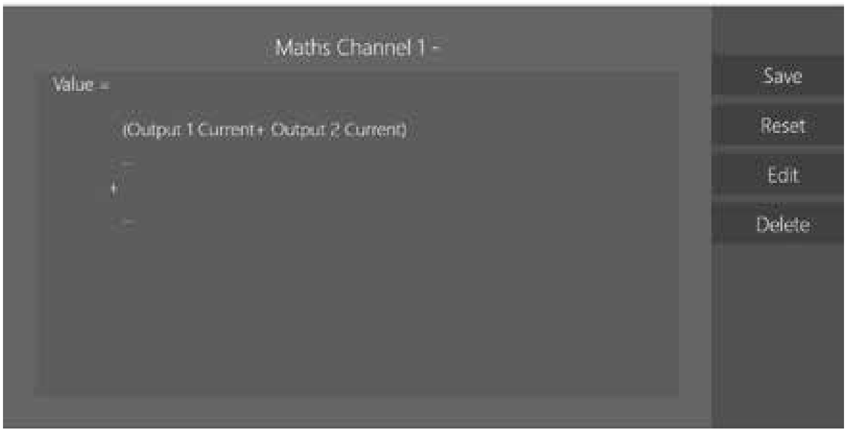
Operation: Addition

Value 1: ☑ Variable ☐ Constant
Output 1 Current | - 0.00 +

Value 2: ☑ Variable ☐ Constant
Output 2 Current | - 0.00 +

The resulting mathematical equation would then look like this -

More equation lines can be added by double clicking on the three dots.


Maths Channel 1 -

Value =

(Output 1 Current+ Output 2 Current)

...
+
...

CAN-bus Communication

The CAN bus is a popular method for sending data between different devices in a vehicle. Data is sent in a digital format on the CAN bus. When a device sends out a CAN message, also known as CAN frame, every device on the CAN bus receives the message. It is up to the receiving devices to figure out if the received message is useful to them or not. To aid this process, each message on the CAN bus starts with an ID. The ID is used to determine what type of CAN message is being sent. After the ID comes the data bytes. A CAN frame can hold between 0 and 8 data bytes.

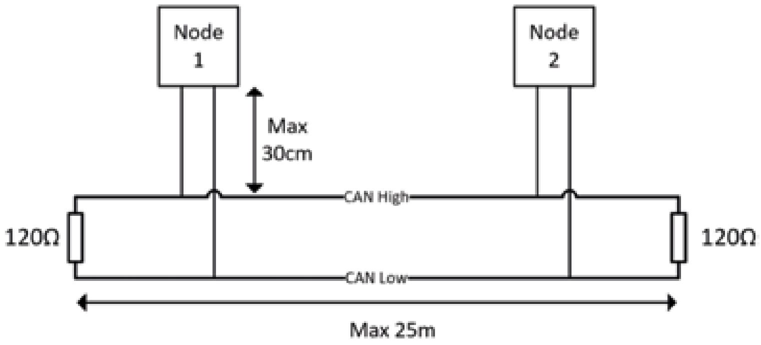
To send a CAN message, you must specify the ID of the message, how many data bytes you want to send and the contents of the data bytes.

To receive a CAN message, you must filter out the message based on its ID, and then extract the data bytes from the CAN messages that make it through the filter.

This system can be used to allow the PDM to communicate with other CAN bus devices, such ECUs, A to D converters, Keypads and more. It is important to note that every device on the CAN bus must be configured to the same speed. If even one device has the incorrect CAN bus speed set, it can bring down the CAN bus and stop all communication.

CAN-bus Wiring

The CAN bus is designed to be extremely robust and fault tolerant, however to ensure the best reliability, it is necessary to follow best practices when wiring up a CAN bus network. The physical CAN bus is constructed with two wires, CAN high and CAN low. The wires should be twisted around each other at approximately 1 twist per inch of length, this is to help mitigate electrical interference on the CAN bus. Each end of the CAN bus MUST have a 120Ohm termination resistor installed between the CAN high and CAN low lines. The two termination resistors are effectively wired in parallel, resulting in a total of 60Ohms of resistance between the CAN high and CAN low lines. It is important to verify that there is approximately 60Ohms of resistance between the CAN high and CAN low lines. Some CAN bus devices will have a CAN termination resistor installed internally. The Hardwire PDM15/25 **DO NOT** have a termination resistor installed internally.



For a bus operating up to 1Mbit/s the recommended practises are:
Maximum bus length = 25m
120Ω termination resistors must be used at both ends of the CAN-bus, to result in 60Ohms total resistance between the CAN high and CAN low lines.
Each unterminated 'stub' must be shorter than 30cm (best to aim for as short as possible)
Maximum of 30 nodes (devices) on the CAN bus.
Twisted pair wires must be used with approximately 1 full twist per inch of length.
For bus speeds lower than 1Mbit/s, following these practises will ensure reliability in the bus.

CAN Keypad

The PDM 15/25 can communicate with a range of CAN bus keypads, including the Grayhill CANOpen, Blink Marine CANOpen, and Marlin Technology J1939 keypads. Using a CAN keypad is a great way to expand the inputs to your PDM, without having to manually wire each switch to a physical input. Instead, the CAN bus keypads connect with two wires - CAN high and CAN low, and can provide upto 40 additional inputs to the PDM. As well as providing extra inputs to the PDM, the CAN keypads provide visual feedback as to the state of the PDM outputs by turning the switches different colours depending on the output states.

To configure the keypad to work with a PDM, make sure that the keypad is wired to the PDM CAN high and CAN low pins, and that the Keypad has the appropriate 12V and Ground power applied. Please also make sure that the CAN bus wiring is in accordance with the previous section on CAN bus wiring. Navigate to the CAN Keypad tab in the configurator software. Up to two keypads can be used simultaneously on the same CAN bus. It is important that both keypads are operating on the same CAN bus speed as one another, and that they have different CAN bus IDs. For setting up a single keypad, select the Keypad 1 tab.

General Keypad Configuration

Enable - Press enable to activate the keypad

Type - Specifies the type of keypad which is being used.

ID - Enter the correct CAN ID for the keypad you are using. Generally, for Blink marine keypads the default ID is 21 decimal, 0x15 hex. For Grayhill keypads the default ID is 10 decimal , 0x0A hex. **CAN bus IDs are entered in the decimal format.**

Button Brightness - Sets the brightness of the keypad button LEDs between 0-100%

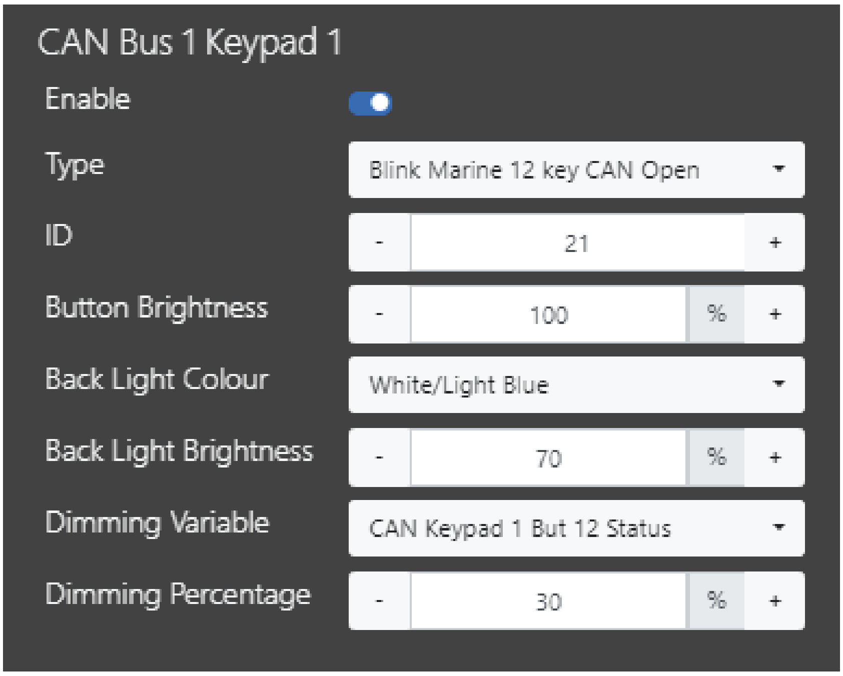
Backlight Colour - Only applicable to Blink Marine Keypads, this setting sets the colour of the illuminating back light on the keypad.

Backlight Brightness - Sets the brightness of the keypad backlight LEDs between 0-100%.

Dimming Variable - The dimming variable can be used to dim the keypad backlight and buttons whenever the dimming variable is equal to 'True' or '1.' This feature is useful for vehicles which are used during both the day and night.

Dimming Percentage - When the dimming variable is equal to 'True' or '1,' both the keypad backlight and buttons will dim to this percentage of their normal brightness. For example, if the Button brightness is set to 100%, the backlight brightness is set to 50%, and the dimming percentage is set to 10%. When the dimming variable is equal to 'True,' the resulting brightness of the keypad buttons will be 10%, and the backlight 5%.

A typical setup for a Blink Marine 12 Key CANOpen keypad is shown in the adjacent image.



CAN Bus 1 Keypad 1

Enable: ●

Type: Blink Marine 12 key CAN Open

ID: 21

Button Brightness: 100 %

Back Light Colour: White/Light Blue

Back Light Brightness: 70 %

Dimming Variable: CAN Keypad 1 But 12 Status

Dimming Percentage: 30 %

Button Specific Configuration

Mode - Each CAN keypad button can operate in one of three modes - Momentary, Latching or States. Each Mode of operation is described below -

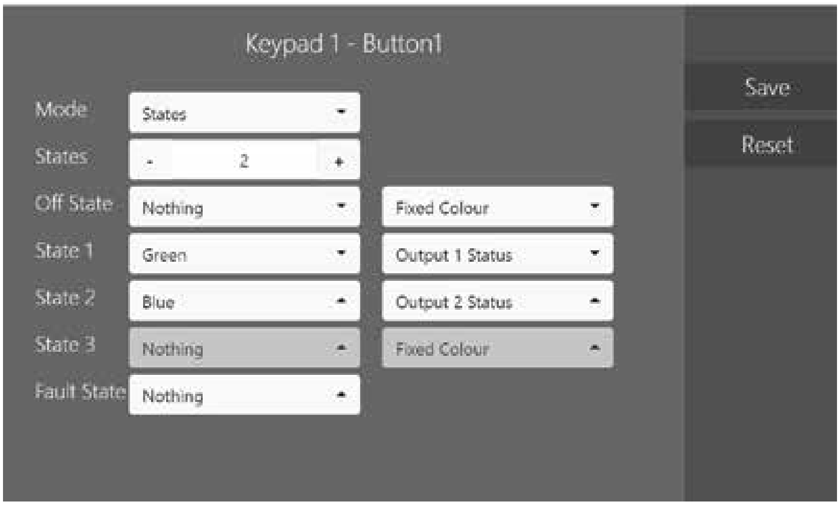
Momentary - When the button is pressed, the state of the button will go from 0/Off to 1/On. When the button is released, the button state will return from 1/On to 0/Off. This mode of operation is useful for powering PDM outputs only while the button is pressed, such as when powering a starter motor on a vehicle engine.

Latching - When the button is pressed, the state of the button will go from 0/Off to 1/On. When the button is released, the state of the button will remain the same (the button has latched on). When the button is pressed once more, the button state will return from 1/On to 0/Off. This mode of operation is useful for powering PDM outputs which need to stay on when the button is pressed once, such as headlights or radiator fans.

States - Up to three 'on' states can be configured in this mode of operation. Every time the button is pressed, the state of the button will increase by one. If the button has reached the maximum state, then when the button is pressed once more, the state of the button will return to the 0/Off state. This mode of operation is useful for powering different outputs with each press of the button. For example - one may want to power the low speed winding of a wiper motor when the button is pressed once, and then the high speed winding of a wiper motor when the button is pressed again. This mode allows for more advanced functionality to be extracted from each button on the keypad.

State Colours

Each state of the Keypad button can be assigned a specific colour. When the button is in each state, the colour of the button will change to the colour which has been specified for that state. In the example below, the keypad button will turn green when in state 1, and blue when in state 2. This provides a useful visual indication to the user about what state the keypad button is in. In addition, each keypad button state can be configured to be a fixed colour, or to match the status of an output. For example, when setting up an indicator, it would be useful for the keypad button to flash on and off in relation to the indicator output.



Keypad 1 - Button1

Mode	States	
States	- 2 +	
Off State	Nothing	Fixed Colour
State 1	Green	Output 1 Status
State 2	Blue	Output 2 Status
State 3	Nothing	Fixed Colour
Fault State	Nothing	

Fault State

Each button on the keypad can be mapped to a PDM output, such that when that output trips due to an over current, the button on the keypad turns red. This provides a visual indicator to the user that the output has tripped, so that the problem can be fixed and the output reset.

CAN Input

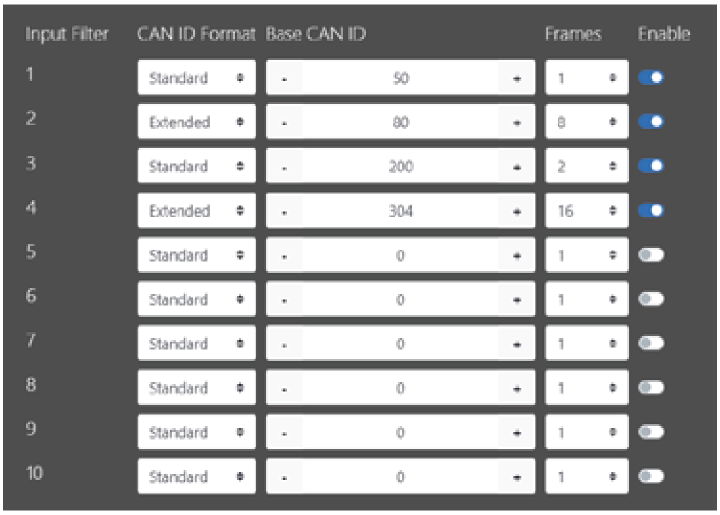
The CAN bus is a two-wire system whereby different devices connected to the bus can communicate with each other. A CAN frame (message) is simply made up of an ID and a payload (data). When a device sends a CAN frame, every other device on the bus receives the message - it is up to each device on the bus to decide if the message is relevant to them.

Configuration

CAN Input Filters

The CAN bus on a vehicle can become heavily crowded with CAN frames from different devices on the bus. Configurable hardware filters are used in the PDM to decipher frames of interest so that the processor is not overloaded with processing unwanted CAN messages. These filters only allow CAN frames with specific IDs to pass through for the processor to deal with, others are discarded automatically.

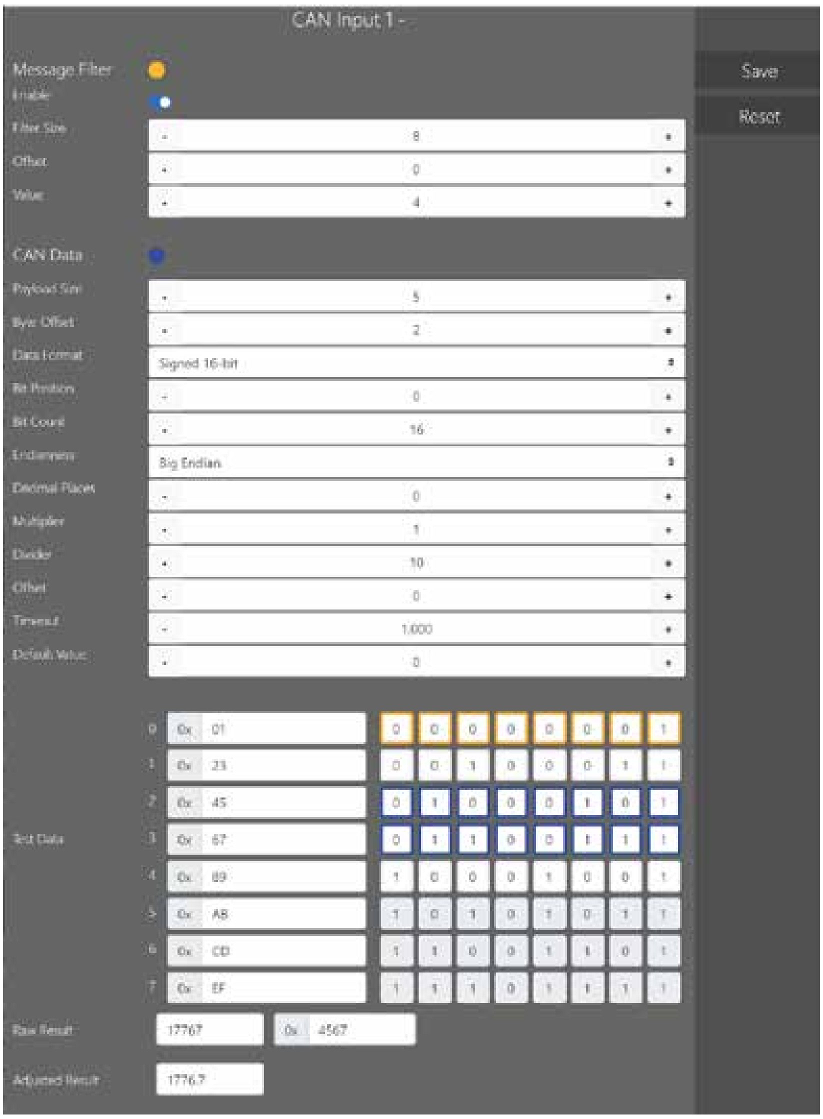
To configure the CAN input filters, navigate to the configuration page and select the CAN inputs tab. Up to 10 filters can be configured. To configure a specific filter, select the ID format, the ID value, and how many frames you want to filter from the base ID onwards, and press enable. It is important to select the correct ID format, consult the instruction manual of the device which is sending the CAN frame to determine which ID is used. Up to 16 consecutive ID CAN frames can be extracted with each filter. Here is a typical example of how the CAN filters may be set up to filter out a range of CAN frames. It is important to note that the IDs are set in decimal format, not hexadecimal.



Input Filter	CAN ID Format	Base CAN ID	Frames	Enable
1	Standard	50	1	●
2	Extended	80	8	●
3	Standard	200	2	●
4	Extended	304	16	●
5	Standard	0	1	○
6	Standard	0	1	○
7	Standard	0	1	○
8	Standard	0	1	○
9	Standard	0	1	○
10	Standard	0	1	○

CAN Inputs

A CAN input can be configured to extract information from CAN frames that have passed through the CAN input filters. Up to 40 CAN inputs can be configured. To configure an input, select the CAN input filter and frame number to use. The CAN input will receive CAN frames with an ID equal to the filter base ID + the frame number -1. You can also add a name and units for the CAN input, which is displayed in the CAN input monitoring tab to make viewing the CAN inputs easier. Click the configuration button – a new window will pop up where the input settings can be changed.



CAN Input 1 -

Message Filter	●
Enable	●
Filter Size	8
Offset	0
Value	4
CAN Data	●
Payload Size	5
Byte Offset	2
Data Format	Signed 16-bit
Bit Position	0
Bit Count	16
Endianness	Big Endian
Decimal Places	0
Multiplier	1
Divider	10
Offset	0
Timeout	1,000
Default Value	0

Test Data

0	0x 01	0	0	0	0	0	0	0	1
1	0x 23	0	0	1	0	0	0	1	1
2	0x 45	0	1	0	0	0	1	0	1
3	0x 67	0	1	1	0	0	1	1	1
4	0x 89	1	0	0	0	1	0	0	1
5	0x AB	1	0	1	0	1	0	1	1
6	0x CD	1	1	0	0	1	1	0	1
7	0x EF	1	1	1	0	1	1	1	1

Raw Result 17767 0x 4567

Adjusted Result 1776.7

Configuration

CAN Input Example

To best explain the setup procedure for a CAN input, an example will be used whereby a sensor value output by a typical CAN device is extracted from its CAN output message stream. **The data sheet for the CAN output stream from the device is shown below.**

Bit Rate	500Kb/sec
Format	29 Bit ID (extended)
Endianness	Big
Transmit Rate	100Hz
Message ID	0x00000180
DLC	5

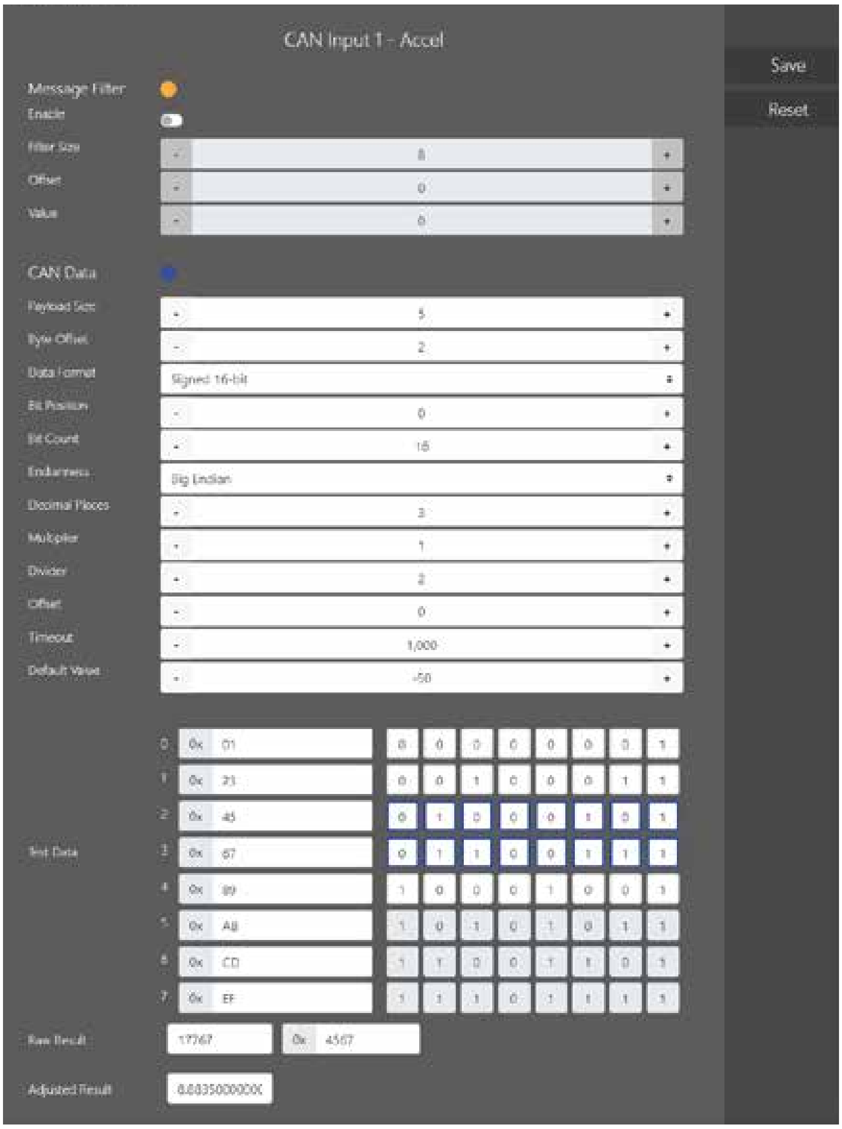
Byte	Label	Data Type	Scaling	Offset	Range
0-1	Temperature	16-bit unsigned	0.01 degree/bit	0	0-655.35
2-3	Acceleration	16-bit signed	0.0005g/bit	0	-16.384g to 16.384g
4	System Volts	8-bit unsigned	0.1V/bit	0	0 to 25.5 V

CAN Input Filter

In this example, the acceleration data will be extracted from the CAN message. The CAN bus bit rate is stated as 500Kb/sec. This must match the CAN bus bit rate of the PDM, which can be configured in the global configuration tab. The format of the ID is 29-bit (extended), with a message ID of 0x00000180. A CAN input filter can be configured to let through messages of this specific ID. It is important to note that the ID 0x00000180 is in hexadecimal format (signified by the 0x). This value must be converted to a decimal number, in this case 384, when inputting the CAN ID into the CAN filter configuration.

CAN Input Configuration

Once the filter is set up and enabled, a CAN input can be configured. The correct settings for extracting the acceleration value from the CAN frame are shown right. Each part will be explained in detail. Example data is shown at the bottom of the window. The Blue highlighted area represents the data which will be extracted from the CAN frame. This value is taken as the 'Raw Result' which can then be adjusted with multipliers and offsets to reach the 'Adjusted Result.' The 'Adjusted Result' is the value which is stored in the PDM as the PDM CAN variable.



CAN Input 1 - Accel

Message Filter	●
Enable	
Filter Size	8
Offset	0
Value	0
CAN Data	●
Payload Size	5
Byte Offset	2
Data Format	Signed 16-bit
Bit Position	0
Bit Count	16
Endianness	Big Endian
Decimal Places	3
Multiplier	1
Divider	2
Offset	0
Timeout	1,000
Default Value	-50

Test Data

0	0x 01	0	0	0	0	0	0	0	1
1	0x 23	0	0	1	0	0	0	1	1
2	0x 45	0	1	0	0	0	1	0	1
3	0x 67	0	1	1	0	0	1	1	1
4	0x 89	1	0	0	0	1	0	0	1
5	0x AB	1	0	1	0	1	0	1	1
6	0x CD	1	1	0	0	1	1	0	1
7	0x EF	1	1	1	0	1	1	1	1

Raw Result: 17767 0x 4567

Adjusted Result: 8.88350000000

Message Filter

The Hardwire Electronics PDM allows the user to receive compound CAN frames with the use of a secondary message filter. Compound CAN frames are used to send a range of CAN frames with different data payloads, while keeping the CAN frame ID constant. In the majority of cases, the number of different frames sent ranges from 0-255. Each frame is represented with a frame number to signify which frame it is. The frame number is often stored in the first byte. In a typical system, the CAN frames are sent in a loop, starting with frame number 0, then 1, 2 etc. Each time the frame number increments, the first byte in the data payload increases signifying which frame number it is.
The message filter allows one to filter a CAN frame by not only the ID, but also by a secondary ID in data payload. This worked example does not use compound CAN messaging, so in this case the filter is disabled.

Payload Size

The payload size (DLC) represents the number of data bytes in the CAN frame which can range from 0-8. This value must exactly match the number of data bytes in the CAN frame.

Byte Offset

This is the byte where the bit masking of the CAN frame starts. In the example the acceleration variable starts at byte 2, therefore an offset of 2 is added. This is shown in the test data as the highlighted blue section starts on byte 2 (3rd byte).

Data Format

Negative numbers are represented in binary as 'signed' numbers, which use a method called 2's complement to signify a negative value. The example uses a signed 16-bit value, meaning that the data can be negative or positive.

Bit Position

The bit position signifies at which bit the bit masking starts. In this example, the acceleration variable data starts at bit 0 of byte 2, therefore no bit offset is required.

Bit Count

The bit count signifies how many bits of data to mask in the CAN frame. In this example, the acceleration variable takes up 16 bits, therefore the bit count must also be 16-bits.

Endianness

Endianness represents which byte comes first in the CAN frame. In this example, byte 2 and 3 are concatenated to form the 16-bit acceleration value. The data is also sent in big-endian form, meaning that byte 2 holds the top half (most significant part) of the data and byte 3 holds the lower half (least significant part) of the data. Little endian would be the opposite. In most cases, big endian format is used to transmit CAN data.

Decimal places

The data sent in a CAN frame often needs to be converted into a form which is more useful. In this case, the acceleration data is sent as 0.0005g/bit, effectively making the raw binary value 2000x larger than the true acceleration g value. Increasing the decimal places effectively divides by 10 and shifts the decimal place.

Multiplier

This value multiplies the raw binary value by any amount.

Divider

This value divides the raw binary value by any amount. In this example, this problem is solved by moving the decimal place over by three and using a divisor of 2.

Offset

This value adds an offset to the adjusted value. The offset allows sensor data to be adjusted to show the correct value. For example, if it is known that the acceleration g variable is out by +0.5g across the full scale, then an offset can be used of -0.5 to correct for the error.

Timeout

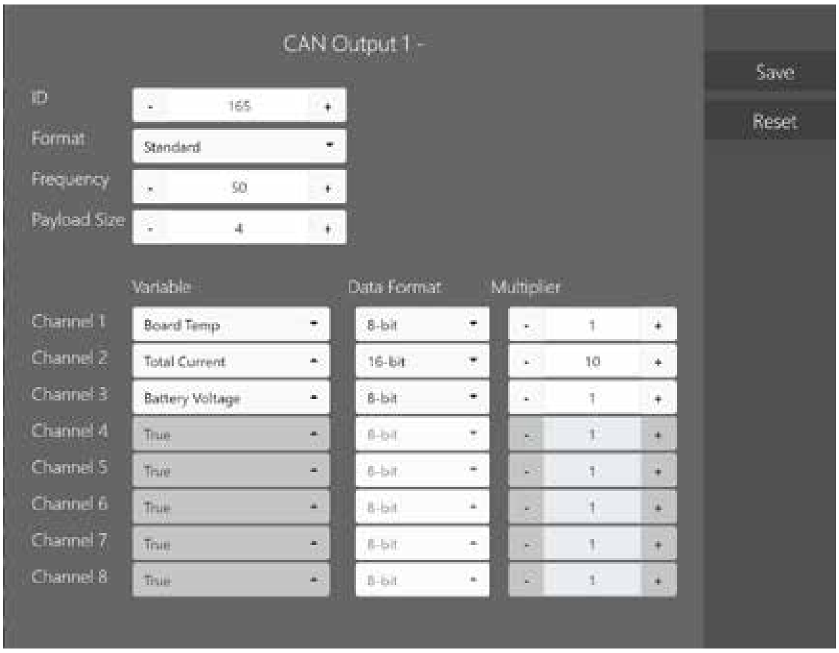
The timeout value is given in milliseconds and represents the maximum allowed time since the last CAN message was received before the timeout event triggers. This feature is necessary in order to validate the PDM CAN variable data. In this example we can see that the device outputs a CAN frame at a rate of 100Hz – or every 10 milliseconds. If the PDM does not receive a message within 10 milliseconds of the previous message, then a problem has occurred. To account for slight timing discrepancies, a timeout value greater than 10 milliseconds should be used. In this case, 1000 milliseconds has been used.

Default Value

When the PDM is first turned on, the value of the PDM CAN variable will be set to the default value. After the CAN input has timed out, the PDM CAN variable will also be set to the default value. In this example, it is known that the maximum valid values of the acceleration are -16.384g to 16.384g. The default PDM CAN variable has been set to -50g. This means that if we see a PDM CAN variable of -50g, then it can be assumed that a fault has occurred and the acceleration data is invalid.

CAN Output

CAN Outputs can be configured to transmit a CAN frame at up to 50Hz. The CAN frame may contain between 0-8 data bytes, which can be filled with a range of PDM variables. To configure a CAN output, navigate to the configuration page and click on the CAN output tab. Click on the configuration button for a chosen output number; a new window will pop up where the output can be configured. An example CAN output configuration is shown to the right.



CAN Output 1 -

ID: 165
Format: Standard
Frequency: 50
Payload Size: 4

	Variable	Data Format	Multiplier
Channel 1	Board Temp	8-bit	1
Channel 2	Total Current	16-bit	10
Channel 3	Battery Voltage	8-bit	1
Channel 4	True	8-bit	1
Channel 5	True	8-bit	1
Channel 6	True	8-bit	1
Channel 7	True	8-bit	1
Channel 8	True	8-bit	1

ID

The CAN ID value is represented in decimal form. The maximum value of the ID is determined by the ID format. A lower ID represents a CAN frame with higher priority. If two devices on the CAN bus attempt to send a frame at the same time, the frame with the lower ID will always send first. This is important as when the CAN bus is heavily loaded, some frames may take longer before they send if they have a higher ID. This also means that important, perhaps safety critical, CAN frames should always have a low ID.

Format

The CAN ID can be either standard (11-bits) or extended (29-bits). Extended ID can be used if all the standard ID's are taken.

Frequency

This is the frequency that the CAN message will be sent in Hz.

Payload Size

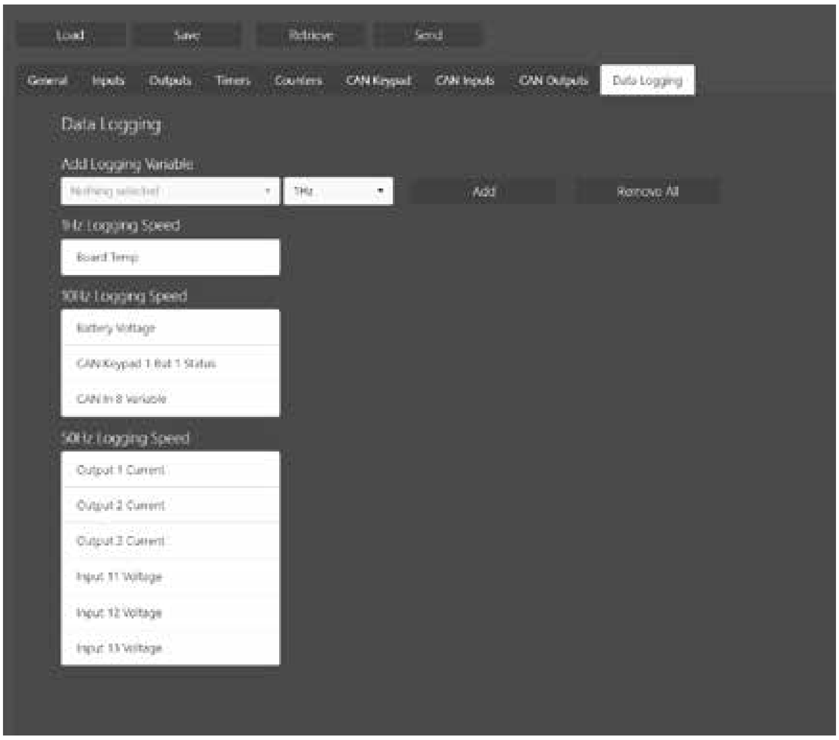
The payload size describes how many data bytes will be sent. In the example CAN output set up, the payload size is set to 4, meaning that there are 4 bytes of data sent in the frame.

The CAN output in the demonstration is constructed from three PDM variables – circuit board temperature, total combined output current and battery voltage. The three variables take up three channels on the output. Each variable must have its data format selected as either 8-bit or 16-bit. It can be noted that the circuit board temperature is being sent as an 8-bit value, but the circuit board temperature is a floating-point number. In this case, the variable will be truncated to fit into the 8-bits of space. To get around this problem, 16-bits of space should be used along with a multiplier. If the circuit board temperature was 32.14°C, placing the variable into a 16-bit space with a multiplier of 100 would result in a transmitted value of 3214 – which can then be divided by the recipient CAN enabled device, preserving the two decimal places of precision.
When the desired output CAN frame has been constructed, press save and then enable the CAN output.

Data Logging

The PDM 15 and 25 both come with 128Mb of on-board flash storage for logging data. This logged data can be downloaded and viewed in the Hardwire Electronics PDM configurator software to diagnose issues and ensure that each electrical component in the vehicle is functioning correctly.

To enable data logging on the PDM, navigate to the configuration tab and then to the Data logging tab. Here, PDM variables can be selected from a drop down list and added to the data logging list. There are three speeds that data can be logged at - 1,10 and 50Hz. Logging more variables at higher speed will consume the data logging memory more quickly. Fast changing variables such as output currents and input voltages should be logged at higher speeds, whereas slow changing variables such as board temperature should be logged at low speeds. To change the logging speed of a variable, click on the variable to highlight it and drag it to the appropriate section. Multiple variables can be selected and moved at once.
Each time the PDM is restarted, or a new configuration is sent, a new data log will start. Data logs are numbered in ascending order. When the data logging memory is full, the data log will start to overwrite the oldest data stored in memory.



Data Logging

Add Logging Variable: Nothing selected | 1Hz

1Hz Logging Speed
- Board Temp

10Hz Logging Speed
- Battery Voltage
- CAN Keypad 1 But 1 Status
- CAN In 8 Variable

50Hz Logging Speed
- Output 1 Current
- Output 2 Current
- Output 3 Current
- Input 11 Voltage
- Input 12 Voltage
- Input 13 Voltage

Monitoring

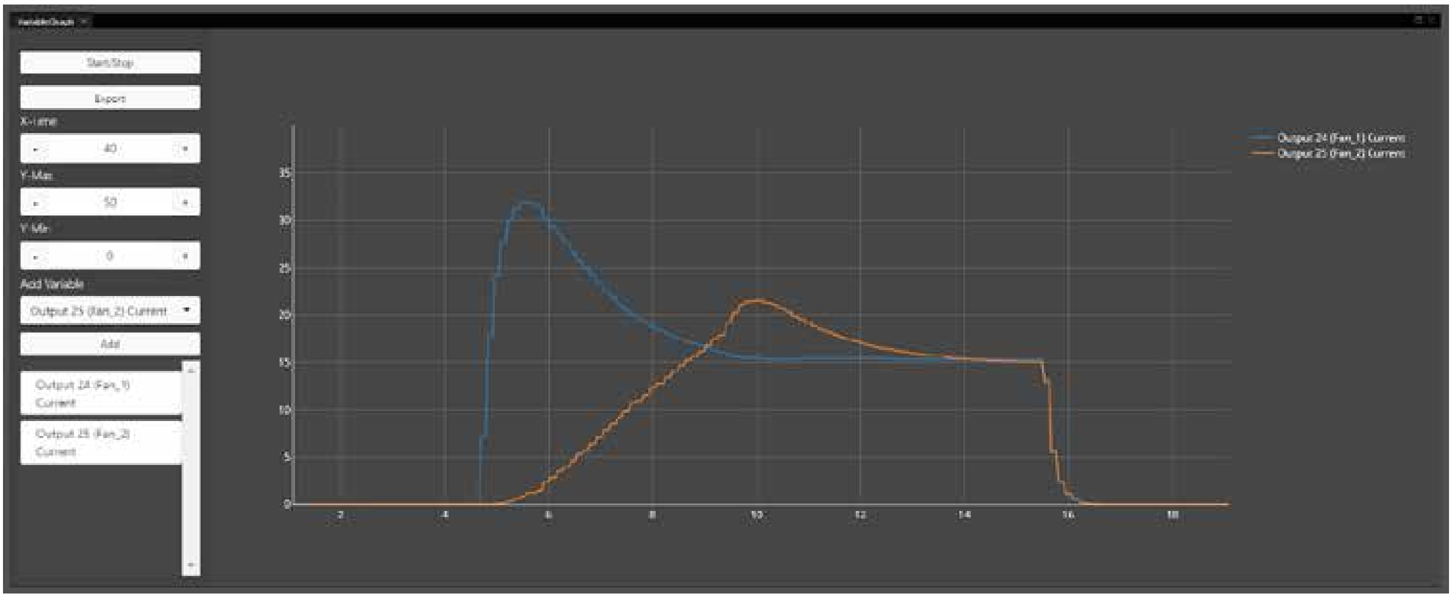
One of the distinct advantages of using a PDM over a conventional relay and fuse system, is that the exact state of the electrical system can be monitored and logged in real time– a necessity in high reliability, high performance applications. The Hardwire Electronics PDM Configurator software allows one to monitor the value of each PDM variable, such as the input voltages, output currents and more.

Live Graph View

The live graph view provides a real-time graphical representation of the PDM variables. This is useful for observing peak inrush current on outputs, the inrush time and the steady state current draw. To access the live graph view, connect the PDM to the PC. Navigate to the monitor tab and then to the live graph view tab. By default, no PDM variables will be shown. To add a variable to the live view graph, click on the drop-down menu and select an appropriate variable. Then press add; the variable will be shown on the graph, as well as in a list below. To remove a variable, double click on the variable in the list.

To adjust the amount of data shown on the graph, change the value in the X-Time box. The X-Time is the number of seconds of data shown on the graph. The Y-Min and Y-Max values can be changed to adjust the vertical scale of the graph to better fit the data onto the screen. Alternatively pressing the auto-scale button to the top right of the graph will automatically adjust the scaling of the graph.

Live data gathering from the PDM can be halted at any moment by pressing the stop button at the top of the graph window. With the live graphing of variables stopped, the buttons at the top of the graph may be used to zoom in/out, save images of the graph, or ro export the graph as a .csv file for viewing on external software. Like with the data logging graphs, the graph windows can be resized, docked, and tabbed by clicking and dragging on the top of the graphing window



X-Time: 40

Y-Max: 50

Y-Min: 0

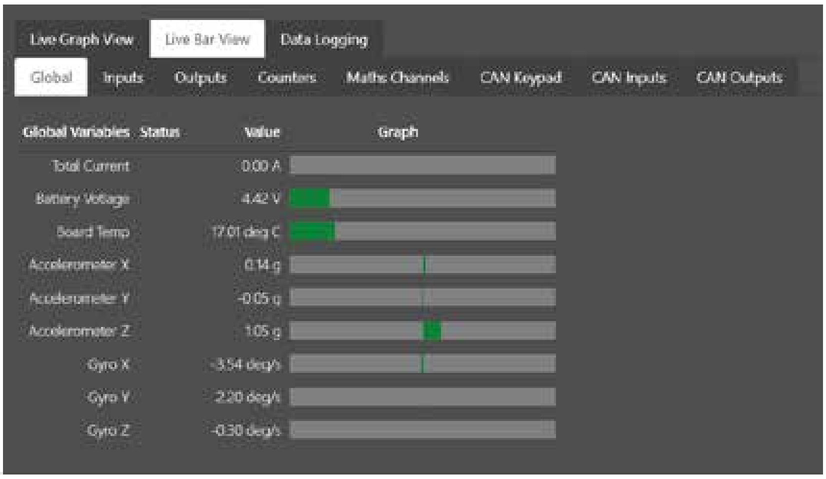
Add Variable: Output 25 (Fan_2) Current

Output 24 (Fan_1) Current

Output 25 (Fan_2) Current

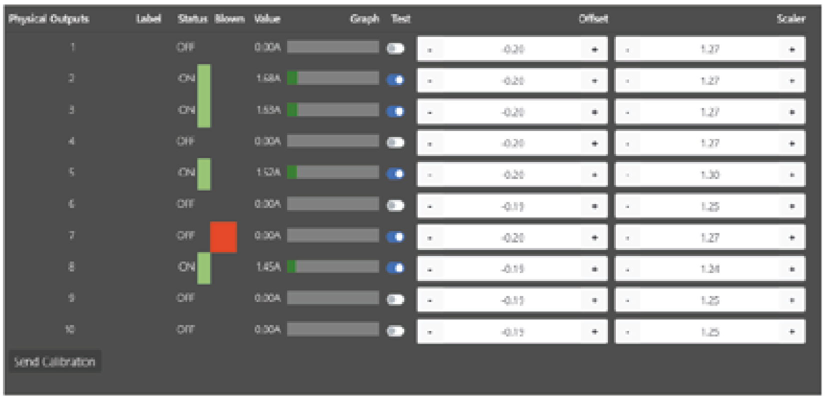
Live Bar View

The live Bar view is like the live graph view - it displays information to the user in real time. When in the monitoring page, click on the live bar view tab. Five more tabs will be available below for the user to click on – Global, Inputs, Outputs, CAN Inputs and CAN Outputs. In each tab, relevant data will be shown in numerical form as well as in a bar graph format.



Global Variables	Status	Value	Graph
Total Current		0.00 A	
Battery Voltage		4.42 V	
Board Temp		17.01 deg C	
Accelerometer X		0.14 g	
Accelerometer Y		-0.05 g	
Accelerometer Z		1.05 g	
Gyro X		-3.54 deg/s	
Gyro Y		2.20 deg/s	
Gyro Z		-0.30 deg/s	

In the outputs monitoring tab, each output can be manually turned on and off to test the output functionality. To test an output, it must first be enabled in the configuration tab, and have appropriate current limits set. The offset and scalar values can also be adjusted for each output, which allows one to fine tune the current sensing on each output if needed.



Physical Outputs	Label	Status	Blown	Value	Graph	Test	Offset	Scaler
1		OFF		0.00A		○	- -0.20 +	- 1.27 +
2		ON		1.68A		●	- -0.20 +	- 1.27 +
3		ON		1.53A		●	- -0.20 +	- 1.27 +
4		OFF		0.00A		○	- -0.20 +	- 1.27 +
5		ON		1.52A		●	- -0.20 +	- 1.30 +
6		OFF		0.00A		○	- -0.15 +	- 1.25 +
7		OFF	🟥	0.00A		●	- -0.20 +	- 1.27 +
8		ON		1.45A		●	- -0.15 +	- 1.24 +
9		OFF		0.00A		○	- -0.15 +	- 1.25 +
10		OFF		0.00A		○	- -0.15 +	- 1.25 +

If an input is on and operating correctly, then a green bar will appear. If an output has tripped from over current, then a red bar will appear.

If an output has tripped from under current, then a blue bar will appear.

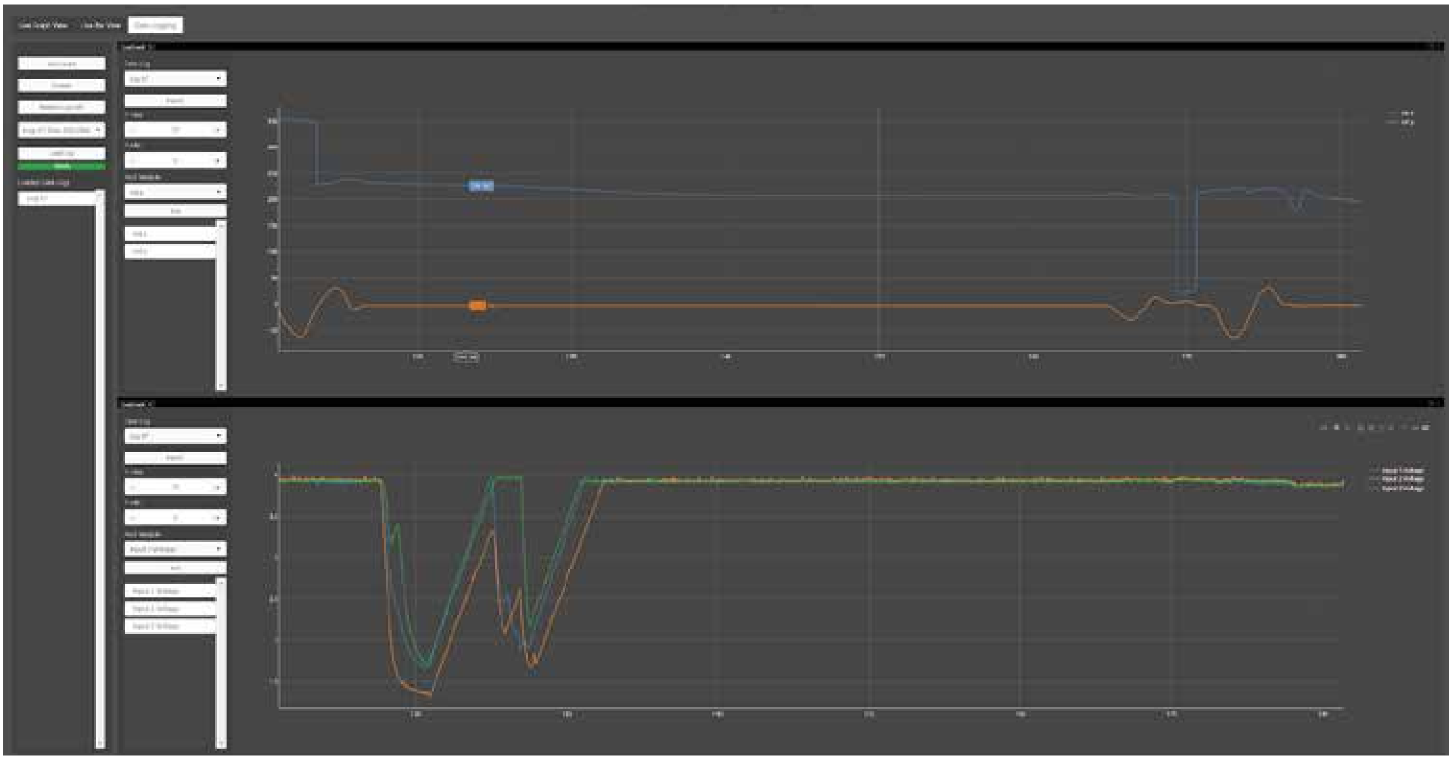
If the output has tripped, then it will be necessary to reset the tripped outputs by restarting the PDM or by triggering the global reset function.

Data Logging

The Hardwire Electronics PDM configurator software can download data logging files from a PDM and display the data in a graphical format. To retrieve data logs, connect to the PDM and navigate to the monitor tab and to the data logging tab. Click on the button 'Retrieve Log Info' to retrieve a list of the data logs stored on the PDM. To view a data log, select the data log and then click the button 'Load Log.' This will start the process of downloading the data log from the PDM.

To view the data log, click on the button 'Add Graph' to add a new graph into the window. In the graph window, there is a list of the currently loaded data logs. Select a data log from the list. Now select the variables from the data log which you wish to view. Double clicking on the variables in the list will remove them from the graph. The graph can be scaled by adjusting the Y-Min and Y-Max values. Alternatively, scrolling the mouse while hovering over the graph will allow one to zoom into different sections of the graph for closer detail. The X-Axis is represented in seconds and the Y-Axis is represented in the base units for that variable - ie current is displayed in Amps, Voltages in Volts etc.

To export the data log, click on the export button. This will allow one to save the data log as a .csv file, which can then be loaded into alternative datalogging or spreadsheet software.
Multiple data logs can be loaded into the configurator software and be viewed on multiple graphs. Graph windows can be resized, docked, and tabbed by clicking and dragging on the graph windows.



Firmware updates

As new software features are added to the Hardwire Electronics PDMs, it is often needed to update the firmware on the PDMs. This can be performed from the Hardwire Electronics PDM Configurator software.

Navigate to the Hardwire Electronics website and to the downloads section. Download the appropriate software for your PDM and save it to a directory on your PC. You will need to download the .exe file for the PDM configurator software, and the appropriate .bin file for the PDM firmware.

To update the firmware on the PDM, navigate to the Update tab and click on the 'Enter Update Mode' button. This will notify the PDM that the user wants to update the firmware. Now unplug the PDM to turn it off. Plug the PDM back in and connect to the PDM in the usual fashion. The device will now be in firmware update mode, which can be confirmed by looking at the status in the sidebar of the PC application which should read 'Firmware Update.' Navigate to the Update tab and click Select Your File Here. Click on the PDM firmware which you just downloaded from the Hardwire Electronics website. Click Open and then Program. After a few seconds, the status indicator bar should begin to move, signifying that the download procedure has successfully started. If the status bar does not move after 10 seconds, unplug the PDM and start again following the same instructions. When the firmware has successfully been written to the PDM, a notification will pop up alerting you of the successful download. Click ok to accept the new firmware.
It is important that both the PDM and PDM configurator software are on the same version.

It is also important that you do not remove power from the device while the firmware is updating.

PDM Version	Number	Description	Notes
15/25	A1	Input 12	
15/25	A2	Input 10	
15/25	A3	Input 8	
15/25	A4	Input 6	
15/25	A5	Input 4	
15/25	A6	Input 2	
15/25	A7	Input 1	
15/25	A8	Input 3	
15/25	A9	Input 5	
15/25	A10	Input 7	
15/25	A11	Input 9	
15/25	A12	Input 11	
15/25	C1	Input 16	
15/25	C2	Input 15	
15/25	C3	Ignition Input (switch to 12V)	
15/25	C4	Output 3	
15/25	C5	Output 2	
15/25	C6	Output 1	
15/25	C7	Ground (Connect to chassis)	
15/25	C8	CAN High	
15/25	C9	CAN Low	
15/25	C10	5V Output	
15/25	C11	Input 13	
15/25	C12	Input 14	
25	D1	Output 17	
25	D2	Output 21	
25	D3	Output 24	
15/25	D4	Output 9	
15/25	D5	Output 8	
15/25	D6	Output 7	
15/25	D7	Output 4	
15/25	D8	Output 5	
15/25	D9	Output 6	
25	D10	Output 24	
25	D11	Output 20	
25	D12	Output 16	
25	B1	Output 19	
25	B2	Output 23	
25	B3	Output 25	
15/25	B4	Output 15	
15/25	B5	Output 14	
15/25	B6	Output 13	
15/25	B7	Output 10	
15/25	B8	Output 11	
15/25	B9	Output 12	
25	B10	Output 25	
25	B11	Output 22	
25	B12	Output 18	

Hardwire
electronics

Version 3

www.hardwire-electronics.co.uk