



# PDM28

---

## Instruction Manual







# Contents

<b>Introduction</b>	4
Specification	4
<b>System Overview Diagram</b>	6
<b>Installation</b>	7
Mounting	7
Pinout	8
Device Power	9
Status LEDs	9
Inputs	10
Outputs	10
CAN Bus Wiring	10
<b>Configuration Software</b>	11
Connection	11
Configuration	12
Configuration Files	12
Save Configuration	12
Load Configuration	12
Send Configuration	12
Retrieve Configuration	12
General	12
CAN Bus Settings	13
Reset Function	13
Cut Off Function	13
Password	13
<b>Inputs</b>	13
Voltage/ Frequency	13
Mode	14
Momentary Mode	14
Latched Mode	14
Active Level	14
Threshold Voltage	14
Hysteresis Voltage	14
Turn On Delay	14
EMA Filter	14
Input Resistor	15
Pull Up Resistor Example	15
<b>Outputs</b>	16
Output Status	17
Output Current	17
Output Voltage	17
Output Load	17
Output Test	17
Output Enable	17

*Continued on next page...*

Trip Mode	17
Low Fuse	17
High Fuse	17
Peak Fuse & Peak Fuse Time	17
Stay On Time	18
Clear Time	18
Retries	18
Functions	18
PWM	18
Frequency	18
Mapping	19
Soft Start Time	19
<b>NIO Power Modules</b>	<b>20</b>
General Settings	20
Outputs	21
Control Type	21
Inputs	22
<b>Timers</b>	<b>22</b>
<b>Counters</b>	<b>22</b>
<b>Maths Channels</b>	<b>23</b>
<b>Generic Functions</b>	<b>25</b>
<b>Sensor Calibrations</b>	<b>25</b>
<b>2D Tables</b>	<b>26</b>
<b>CAN Keypads</b>	<b>27</b>
Keypad Configuration	27
Keypad Button Configuration	29
Pass Key	30
<b>CAN Bus Basics</b>	<b>30</b>
<b>CAN Inputs</b>	<b>31</b>
CAN Configuration Files	31
DBC File Loader	31
CAN Bus Filter Configuration	32
CAN Input Configuration	32
Test Data	32
Message Filter	33
Payload Size	33
Data Format	33
Bit Position	33
Bit Count	33
Endianness	33
Multiplier	33
Offset	34

# Contents

Timeout	34
Default Value	34
<b>CAN Outputs</b>	<b>34</b>
User CAN Outputs	34
CAN BUS Selection	35
CAN ID	35
Send Mode	35
Payload	35
<b>Can Output Stream</b>	<b>35</b>
<b>LIN Bus</b>	<b>36</b>
General	36
Nodes	37
Signals	37
Encodings	38
Frames	39
Schedule Table	40
<b>Data Logging</b>	<b>41</b>
<b>Monitoring</b>	<b>41</b>
Graph View	41
Variable View	42
CAN Bus Analyser	43
<b>Logging</b>	<b>44</b>
<b>Firmware Updates</b>	<b>45</b>
PDM28 Firmware Update	45
NIO Power Module	45
<b>Examples</b>	<b>45</b>
1. Basic Input Setup	45
2. ECU Triggered Input	46
3. Latching Input Setup	47
4. Basic Output Setup	47
Indicator/Turn Signal Setup	48
PWM Fan Control Setup	49
Video Examples	50

# Introduction

A PDM (Power Distribution Module) is a device used to replace conventional relays and fuses in a vehicle electrical system. The Hardwire Electronics PDM28 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 - using user defined logic functions. 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.

PDMs offer distinct advantages over conventional systems using fuse boards and relays, listed below.

- Greatly simplifies system wiring and reduces system component count, overall saving you money 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.
- 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 the 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.

# Specification

## Controller Module

General	
<b>Size</b>	318x124x58mm
<b>Weight</b>	1640g
<b>Operating Temperature</b>	-55 - 90°C [-67 - 194°F ]
<b>Operating Voltage</b>	4-32V
<b>Quiescent Current</b>	4-5mA
<b>Construction</b>	Machined Aluminium Enclosure, Media Blasted and Anodised Black
<b>IP Rating</b>	IP67 Water and Dust Ingress Protection
<b>Connectors</b>	2x 12-Pin DTM, 2x 12-Pin DT, 1x 4-Pin DTP
<b>Indicator LEDs</b>	50x RGB LEDs
Inputs	
<b>Type</b>	16 analogue inputs which can also serve as digital inputs.
<b>Measurement Range</b>	Each input can measure voltages between 0-28V, independent of the battery voltage.
<b>Measurement Resolution</b>	16-bit or 0.001V
<b>Frequency Measurement</b>	Automatic frequency measurement up to 300Hz
<b>Protection</b>	Fully protected for input voltages between -24V to 40V.
<b>Pull Up/Pull Down Resistors</b>	Each input has a software selectable 10K Pull-up or Pull-down resistor
<b>Added Features</b>	Configure as active high or low. Configurable Hysteresis voltage. Latching or momentary operation.

# Specification

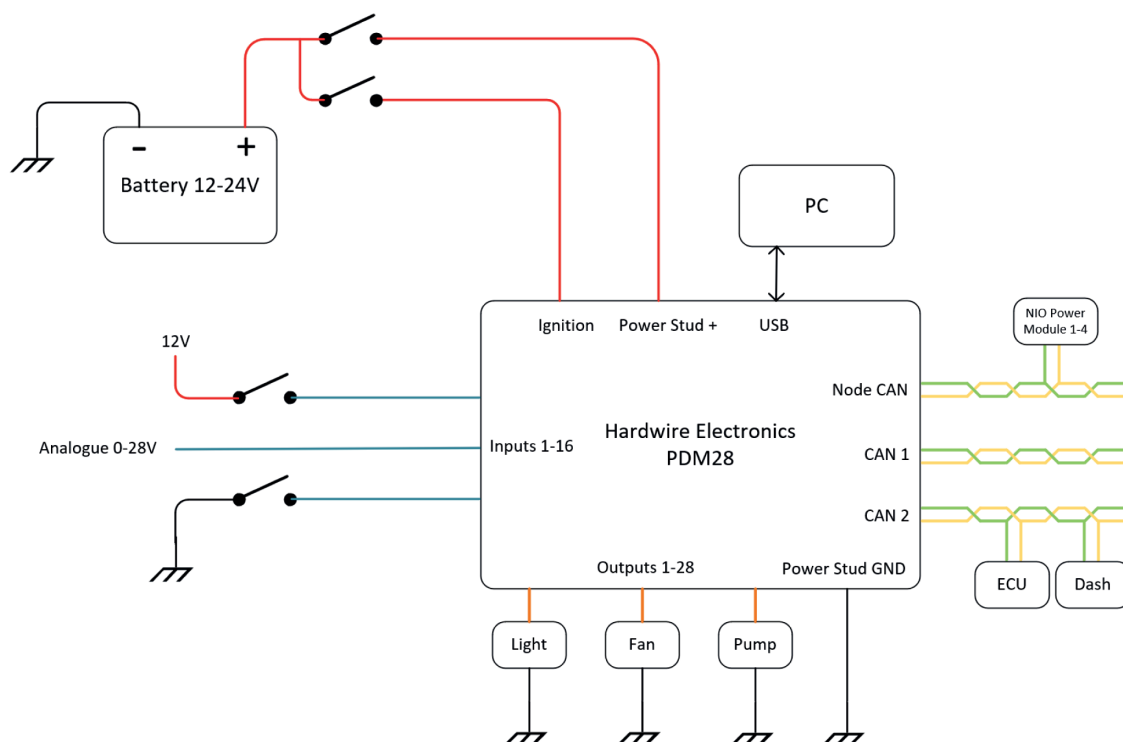
Outputs	
<b>Type</b>	4 high/low side outputs 100A peak, 30A continuous, with reverse current protection. ( <i>Pin-limited to 25A @125C</i> ) 6 high/low side outputs 80A peak, 20A continuous, with reverse current protection. ( <i>Pin-limited to 13A @125C</i> ) 18 high/low side outputs 80A peak, 20A continuous ( <i>Pin-limited to 13A @125C</i> )
<b>Combined</b>	200A combined output maximum current
<b>Output current control steps</b>	100mA
<b>Protection</b>	Each output is overcurrent, overtemperature, load dump and reverse polarity protected.
<b>Soft Start</b>	PWM soft start can be enabled on outputs to limit inrush current.
<b>PWM</b>	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. 1000Hz maximum PWM frequency.
<b>Added Features</b>	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.
LIN Bus	
	1x LIN Bus Port.
CAN Bus	
<b>Interface</b>	3x CAN2.0A/B ports (Bus 3 reserved for NIO Power Module Expansion)
<b>Bus Speed</b>	50, 100, 125, 250, 500, 1000 kbps
<b>Termination Resistor</b>	Both CAN bus ports have software selectable CAN termination resistors.
<b>DBC File Import</b>	Easily import DBC files to configure the CAN inputs of the PDM
<b>Input Stream</b>	100x inputs, fully customisable variable parsing with bit masking, math functions, and compound CAN message filtering.
<b>Output Stream</b>	100x Outputs, Fully customisable CAN frame construction with math function.
CAN Keypad	
	Support for Blink Marine, Grayhill, Motec, Haltech and Marlin CAN keypads. Up to four keypads can be used at once.
Logging	
<b>Storage</b>	128Mb of onboard flash storage for logging PDM data.
<b>Speed</b>	1Hz to 50Hz
<b>Viewing</b>	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	
<b>Output Functions</b>	Each Output can be switched with a user defined function.
<b>Logical Operators</b>	AND, OR, NOR, XOR, NAND, NOR, >, >=, <, <=, Equal, Not Equal.
Maths Channels	
	Fully customisable Maths Functions
<b>Mathematical Operators</b>	Mathematical Operators - Addition, Subtraction, Multiplication, Division,
<b>Binary Operators</b>	Modulus, Choose, Min, Max Binary Operators - AND, OR, XOR, Left Shift, Right Shift

# Specification

Counters	
	Configurable counters which increment, decrement, and reset based on user defined conditions.
Sensor Calibration	
	10x Sensor Calibration Tables
2D Tables	
	10x Individual 20x20 2D map tables
IMU	
<b>Rotation</b>	3-axis rotational position measurement 0.1° resolution
<b>Gyroscope</b>	2000°/s 3-axis measurement
<b>Accelerometer</b>	-8g to 8g 3-axis measurement
PCB Temperature	
	6-Zone PCB temperature measurement.
RTC	
	Real time clock with battery backup for 30 days. Used to store the correct time and date for logs.
USB	
<b>Type</b>	USB 2.0 compliant with Mini-B
<b>Driver</b>	Integrated driver with Windows 10
PC App	
<b>Name</b>	Hardwire Electronics Pro PDM Configurator Software.
<b>Compatibility</b>	Windows 10 & 11 machines
<b>Connectivity</b>	Easy connection to the PDM with USB and windows integrated drivers. No unreliable USB-serial converter needed.

# System Overview Diagram

Figure 1 Typical installation of a PDM28 in a vehicle electrical system.

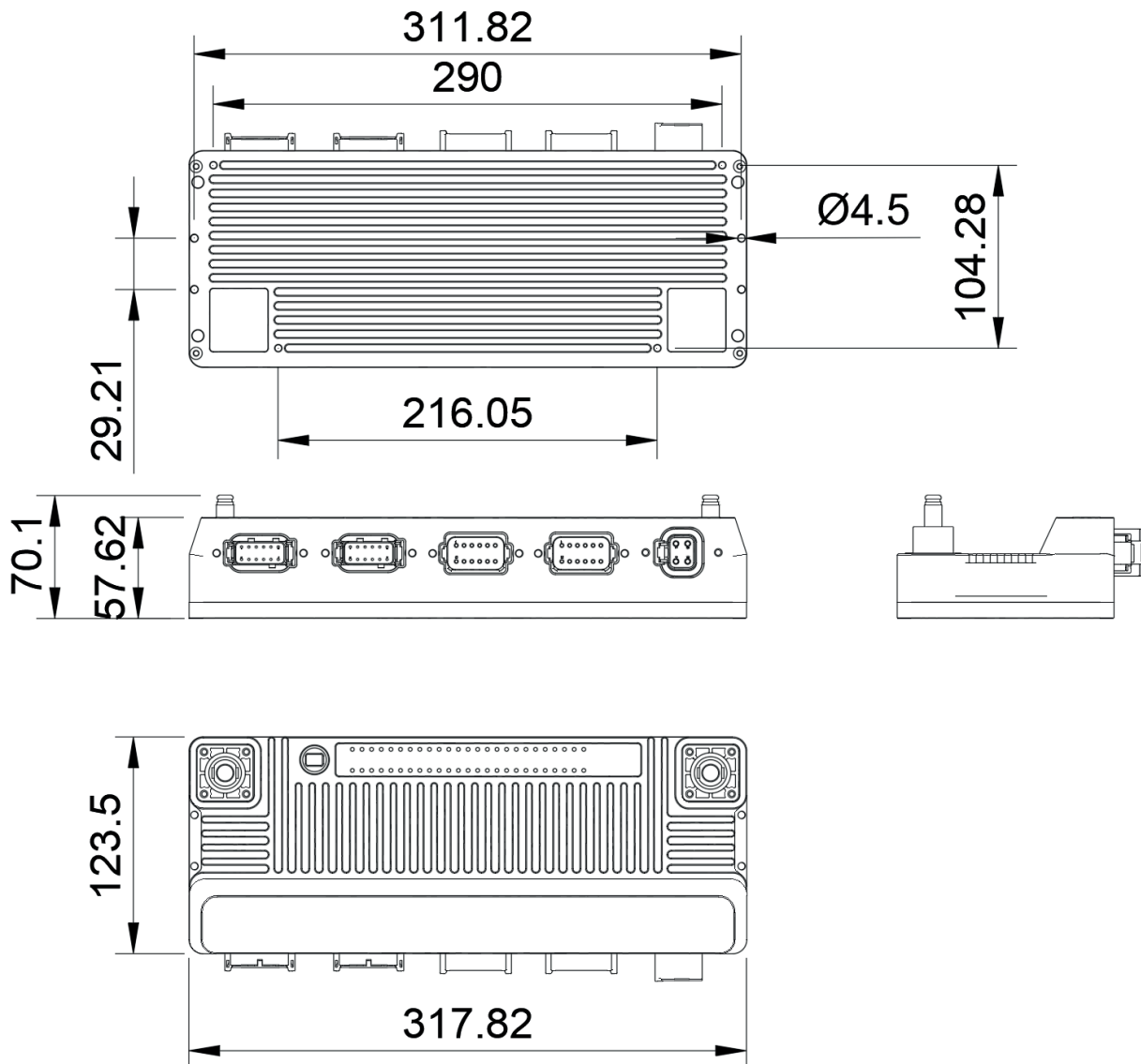


# Installation

## Mounting

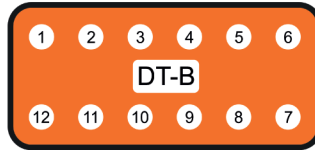
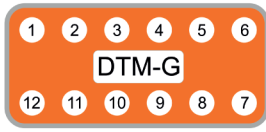
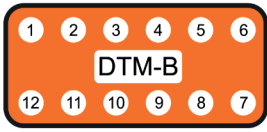
Figure 2 details the physical dimensions of the PDM28, as well as the hole size and spacing for mounting. The PDM should be mounted in a well-ventilated area to cool the unit when under heavy load. Care should be taken to mount the PDM in a location which minimises the exposure to water and dirt. In 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. It is recommended to mount the PDM on rubber mounts to minimise the vibration coupling from the vehicle to the PDM. This will help with IMU data.

Figure 2 PDM28 dimensional diagram - Measurements given in mm.



# Installation

## Pinout



Colour	Pin	Description	Note
	DTM-B-1	PDM Enable	10K Pull Down, 3-24V to activate
	DTM-B-2	LIN	
	DTM-B-3	CAN 2 Low	
	DTM-B-4	CAN 2 High	
	DTM-B-5	CAN 1 Low	
	DTM-B-6	CAN 1 High	
	DTM-B-7	N CAN Low	
	DTM-B-8	N CAN High	
	DTM-B-9	Input 1	0-27V, 10K Pull Up/Down
	DTM-B-10	Input 2	0-27V, 10K Pull Up/Down
	DTM-B-11	Input 3	0-27V, 10K Pull Up/Down
	DTM-B-12	Input 4	0-27V, 10K Pull Up/Down
	DTM-G-1	Input 5	0-27V, 10K Pull Up/Down
	DTM-G-2	Input 6	0-27V, 10K Pull Up/Down
	DTM-G-3	Input 7	0-27V, 10K Pull Up/Down
	DTM-G-4	Input 8	0-27V, 10K Pull Up/Down
	DTM-G-5	Input 9	0-27V, 10K Pull Up/Down
	DTM-G-6	Input 10	0-27V, 10K Pull Up/Down
	DTM-G-7	Input 11	0-27V, 10K Pull Up/Down
	DTM-G-8	Input 12	0-27V, 10K Pull Up/Down
	DTM-G-9	Input 13	0-27V, 10K Pull Up/Down
	DTM-G-10	Input 14	0-27V, 10K Pull Up/Down
	DTM-G-11	Input 15	0-27V, 10K Pull Up/Down
	DTM-G-12	Input 16	0-27V, 10K Pull Up/Down
	DT-B-1	Output 1	20A High/Low side, 1KHz PWM
	DT-B-2	Output 2	20A High/Low side, 1KHz PWM
	DT-B-3	Output 3	20A High/Low side, 1KHz PWM
	DT-B-4	Output 4	20A High/Low side, reverse current blocking, 1KHz PWM
	DT-B-5	Output 5	20A High/Low side, 1KHz PWM
	DT-B-6	Output 6	20A High/Low side, 1KHz PWM
	DT-B-7	Output 7	20A High/Low side, 1KHz PWM
	DT-B-8	Output 8	20A High/Low side, reverse current blocking, 1KHz PWM
	DT-B-9	Output 9	20A High/Low side, 1KHz PWM
	DT-B-10	Output 10	20A High/Low side, 1KHz PWM
	DT-B-11	Output 11	20A High/Low side, 1KHz PWM
	DT-B-12	Output 12	20A High/Low side, 1KHz PWM



# Installation

Colour	Pin	Description	Note
	DT-G-1	Output 13	20A High/Low side, 1KHz PWM
	DT-G-2	Output 14	20A High/Low side, 1KHz PWM
	DT-G-3	Output 15	20A High/Low side, 1KHz PWM
	DT-G-4	Output 16	20A High/Low side, reverse current blocking, 1KHz PWM
	DT-G-5	Output 17	20A High/Low side, 1KHz PWM
	DT-G-6	Output 18	20A High/Low side, 1KHz PWM
	DT-G-7	Output 19	20A High/Low side, 1KHz PWM
	DT-G-8	Output 20	20A High/Low side, reverse current blocking, 1KHz PWM
	DT-G-9	Output 21	20A High/Low side, reverse current blocking, 1KHz PWM
	DT-G-10	Output 22	20A High/Low side, 1KHz PWM
	DT-G-11	Output 23	20A High/Low side, 1KHz PWM
	DT-G-12	Output 24	20A High/Low side, reverse current blocking, 1KHz PWM
	DTP-B-1	Output 25	30A High/Low side, reverse current blocking, 1KHz PWM
	DTP-B-2	Output 26	30A High/Low side, reverse current blocking, 1KHz PWM
	DTP-B-3	Output 27	30A High/Low side, reverse current blocking, 1KHz PWM
	DTP-B-4	Output 28	30A High/Low side, reverse current blocking, 1KHz PWM

\*Limited by maximum current capacity of contact and pin

## Device Power

The PDM28 is powered by two Amphenol RADLOK connectors- Red (Battery Positive 12-24V) and Black (Battery negative)- which make connecting/disconnecting the PDM an easy process with no tools required. The RADLOK connectors are crimped using a standard battery cable crimper. It is recommended to use at least 2AWG/ 35mm<sup>2</sup> wire for the Positive and Negative PDM supplies.

It is best to Ground the PDM28 to the vehicle chassis close to where the PDM28 is located.

A battery isolator should be installed between the vehicle battery and the PDM28.

The device is enabled (powered up) when >3V is applied to the PDM Enable Pin. This pin should be a switched live, typically coming from a key barrel or ignition switch. A 10K pull down resistor is used on the PDM Enable Pin. When the PDM is not enabled, the PDM28 current consumption will be approximately 10mA.

## Status LEDs

The PDM28 has 50x RGB Status LEDs.

**Power:** Green when the PDM has the correct supply voltage. Blue if the voltage is too low (<7V). Red if the voltage is too high (>28V)

**Temperature:** Green when the PDM temperature is below 80degC, red otherwise.

**CAN:** Green if the CAN TEC and REC Values are below 100, red otherwise.

**Output:** Green if either the HS or LS output is On. Red if either the HS or LS output has tripped due to Over Current. Blue if either the HS or LS output has tripped due to Under Current.

**Input:** Green if the Input is triggered.



# Installation

## Inputs

The PDM28 has 16 inputs which can measure analogue voltages between 0-28V with a sample resolution of 16-bit or 0.001V. The inputs are sampled at a high rate, which allows the frequency of the input signal to be measured up to 300Hz. Each input has a software selectable 10K $\Omega$  pull up and pull down resistor. The great flexibility of the PDM28 Inputs allows analogue, digital, switched 12V and switched ground inputs to be measured with ease.

## Outputs

The PDM28 has 28 Outputs. All 28 outputs are High and Low sided, meaning that they can output the positive battery voltage to source current OR put the output to ground to sink current. Internal circuitry stops both the high and low side being activated at the same time on any given output.

Ten outputs are reverse current protected, meaning that when the voltage present on the output is greater than the battery voltage, the output does not allow current to flow back into the output. This is useful when driving wiper motors with low/high speed windings. Each output has current sensing, thermal overload protection, static discharge protection, short circuit protection and over/under voltage protection.

The outputs are able to switch on and off incredibly quickly, which reduces heat build up when using the output in PWM mode. Four of the outputs have a continuous current rating of 30Amps, (limited by the current capacity of the contact and pin to 25A at 125degC), and should be used for powering larger devices such as radiator fans and large fuel pumps. The remaining outputs have a continuous current rating of 20Amps (limited by the current capacity of the contact and pin to 13A at 125degC). Multiple outputs of the same type can be joined together to increase the current capacity.

## 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 1"/2 cm of length, this is to help mitigate electrical interference on the CAN bus. Each end of the CAN bus **MUST** have a 120 $\Omega$  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 60 $\Omega$  of resistance between the CAN high and CAN low lines. It is important to verify that there is approximately 60  $\Omega$  of resistance between the CAN high and CAN low lines. Some CAN bus devices will have a CAN termination resistor installed internally. The Hardwire PDM28 has software selectable CAN termination resistors on each CAN bus port. If the PDM28 CAN termination resistor is used, then the PDM28 must be placed at the end of the CAN bus.

Care must be taken to ensure that the physical wiring of the CAN bus meets the following requirements -

- The maximum bus length must not exceed 25m
- The CAN bus must be terminated at each end with a 120 $\Omega$  resistor. The PDM28 has a software selectable CAN termination resistor.
- CAN bus 'stubs' must be no longer than 30cm in length
- Twisted pair wire must be used for the CAN bus with a minimum of one twist per 3cm.

# Configuration Software

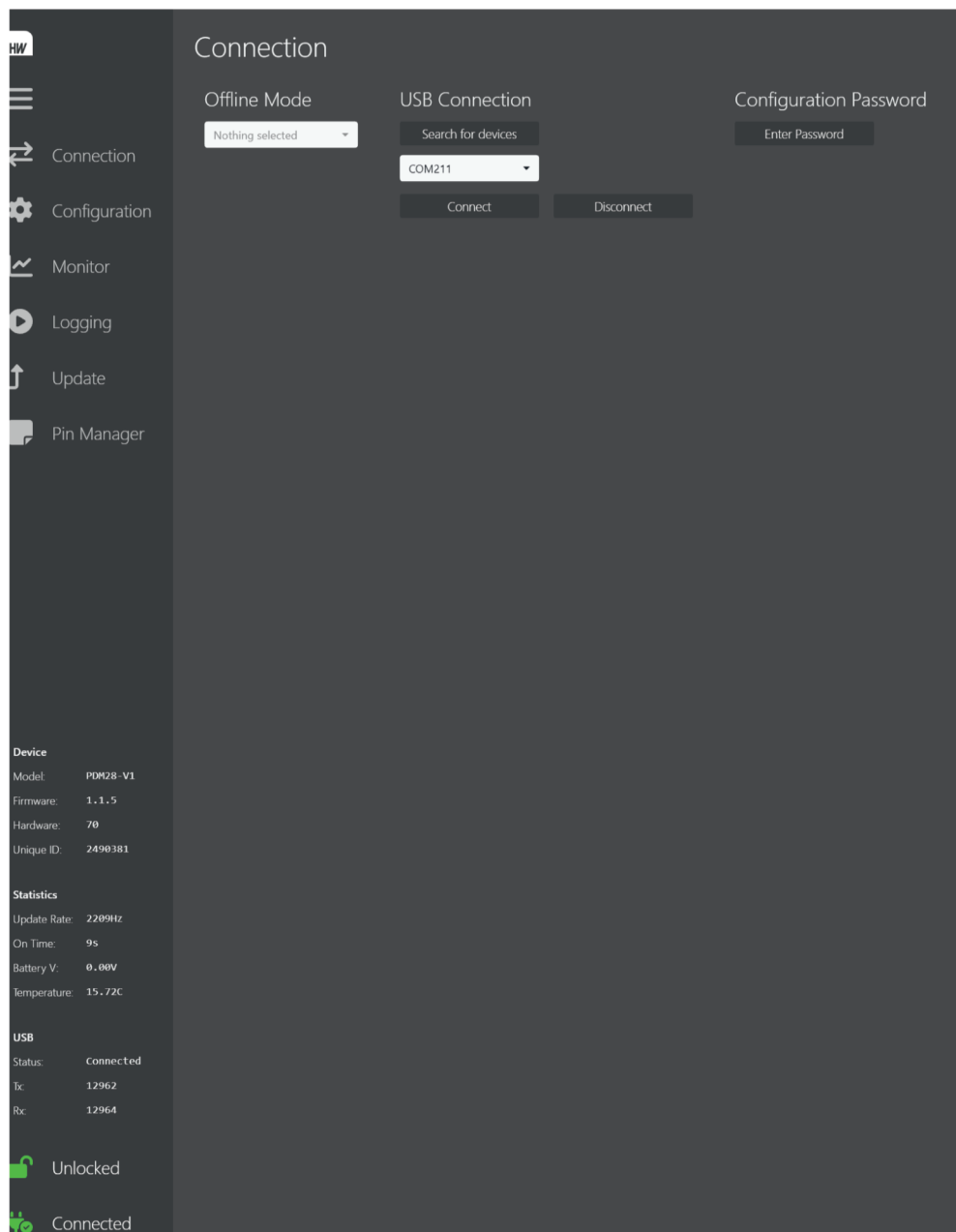
The Hardwire Electronics Configuration Software has been designed with ease of use in mind. There are five main sections to the software - Connection, Configuration, Monitor, Logging and Update. Each section can be accessed via the tabs at the left of the screen in the sidebar. Other important information such as USB connection status can also be found in the sidebar for easy viewing.

The software can be found on the supplied USB drive, or on the Hardwire Electronics Website in the downloads section.

## Connection

To connect to the PDM28, plug in to the PDM28 with the provided USB cable. Navigate to the Connection Tab and click 'Search for Devices.' The PDM28 can then be selected from the list. Now press connect. If a successful connection has been made, then the USB connection status will update in the sidebar, and the Tx and Rx values will begin to count up. If you don't have a PDM, but would still like to access the software, select a model from the Offline Mode select box.

If the PDM configuration has been locked with a password, you will be prompted to enter the password to gain access. If you have forgotten your password, then you will need to reflash the PDM with new firmware.



# Configuration Software

## Configuration - Configuration Files

To configure the PDM28 and connected NIO Power Modules, connect to the PDM28 via USB and navigate to the Configuration Tab.

### Save Configuration

Pressing Save at the top of the screen will prompt the user to locate a directory on the PC in which to save the configuration file. Configuration files should have a .HWPDM file extension.

### Load Configuration

Likewise, pressing Load will prompt the user to navigate to a configuration file which can then be loaded into the configuration software.

### Send Configuration

Press the Send button to send the current configuration to the PDM28, and press Retrieve to retrieve the configuration currently on the PDM28.

### Retrieve Configuration

When retrieving the configuration from the PDM28, if any changes are detected between the current configuration on the PC and the configuration on the PDM28, then the user will be given a choice to either maintain the current configuration on the PC, or alternatively to overwrite the configuration on the PC and load in the configuration from the PDM28 onto the PC.

## General

In the General tab, the user can configure the general settings for the PDM28.

The screenshot displays the Configuration Software interface. The main window is titled "Configuration" and features a top navigation bar with buttons for "load", "Save", "Retrieve", and "Send". Below this is a "Configuration File" section and a horizontal menu with tabs for "General", "Inputs", "Outputs", "NIO Module", "Timers", "Counters", "Maths Channels", "Generic Functions", "Sensor Calibrations", "2D Tables", "CAN Keypad", "CAN Inputs", "CAN Outputs", "CAN Stream", "LIN Bus", and "Data Logging". The "General" tab is active, showing "General Settings" for "NPD CAN". It includes sections for "CAN Bus 1" and "CAN Bus 2", each with "CAN Speed" (set to 1000 kbit/s) and "Termination Resistor" (toggle off). A "Global Functions" section has "Reset" and "Cut Off" options, each with "Expand" and "Enable" (toggle on) settings. A "Configuration Password" section has an "Enter New Password" field and a "Click" button. On the left sidebar, there are icons for "Connection", "Configuration", "Monitor", "Logging", "Update", and "Pin Manager". At the bottom left, a "Device" section shows: Model: PDM28-V1, Firmware: 1.1.5, Hardware: 70, Unique ID: 2490381. A "Statistics" section shows: Update Rate: 2196Hz, On Time: 20 : 37s, Battery V: 0.00V, Temperature: 16.98C. A "USB" section shows: Status: Connected, Tx: 17654, Rx: 17656.

Continued on next page...

# Configuration Software

## CAN Bus Settings

At the top of the tab, the user can select the CAN termination resistors and CAN speed for each CAN bus on the PDM28. CAN bus 1 and 2 are for the user to use, and can interface with external CAN bus devices such as Keypads, Digital Dashboards and ECUs. The N-CAN bus is used to interface with the NIO Power Modules, and has a fixed speed of 500kbit/s.

## Reset Function

The Reset function is used to reset tripped outputs which have either tripped from over current or under current. More information is given on configuring functions later in the instruction manual. When the Reset function evaluates to 'True', each tripped output on the PDM28 and each NIO Power Module will reset, so that normal operation can resume.

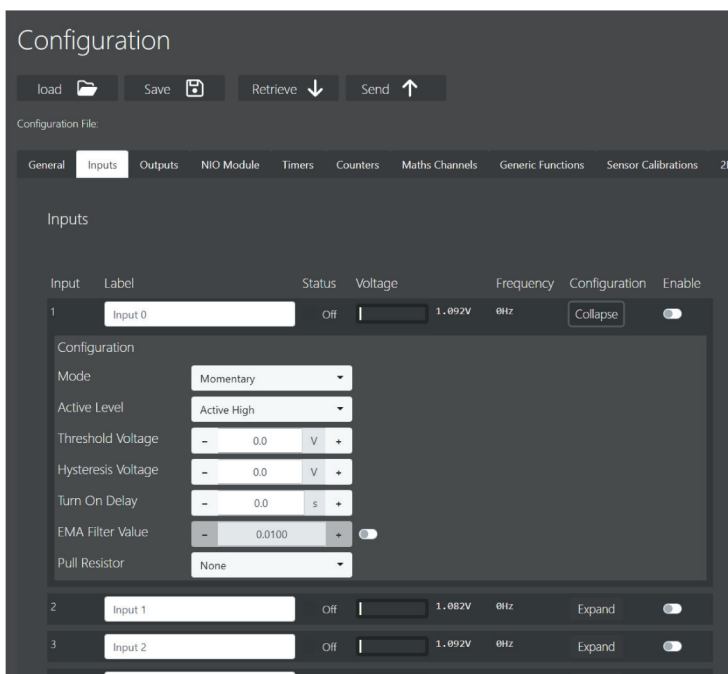
## Cut Off Function

The Cut Off function is used to turn off every output on the PDM28 and each NIO Power Module. This feature is a useful safety feature, to disable the entire vehicle electrical system if a problem occurs. When the Cut Off function evaluates as 'True', each output is immediately turned off. The outputs will remain turned off until the PDM28 power is turned off and back on again, or until the Reset function is triggered.

## Password

Once you have completed your PDM setup, you can lock the configuration with a password. To add a password, click enter new password and check the box to enable it. Each time you connect to the PDM, you will be prompted to enter the password again. If you forget the password, the PDM must be updated with new firmware to wipe the PDM28 clean.

# Inputs



## Voltage/Frequency

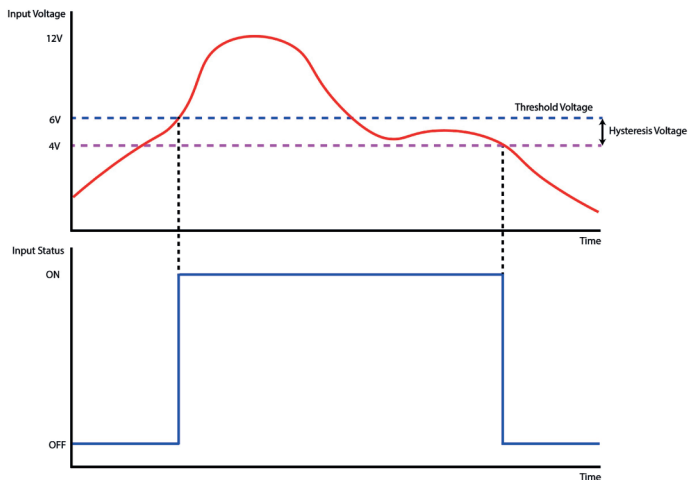
The PDM28 inputs are able to measure both the voltage and frequency of the input signal. The inputs have a voltage measurement range of 0-28V, and a frequency measurement range of 0-300Hz. The frequency measurements are performed automatically, and will work with any periodic waveform between 0 and 300Hz, irrespective of the peak to peak voltage or DC offset value.

# Inputs

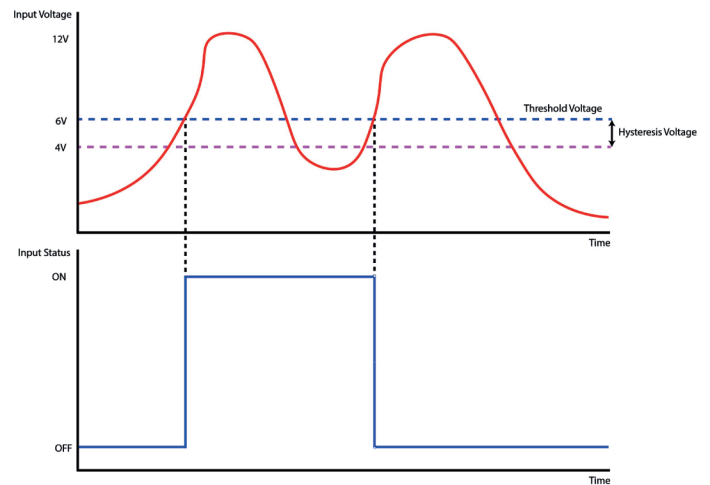
## Mode

The inputs 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. In Momentary Mode, when the input voltage passes the threshold voltage, the input will turn on. In Latched Mode, when the input voltage passes the threshold voltage, the input will toggle its state from Off to On, or from On to Off.

### Momentary Mode



### Latched Mode



## Active Level

Each input can be programmed as Active-High or Active-Low. Active-High causes the input to turn on when the voltage rises above the threshold. Active-Low causes the input to turn on when the voltage falls below the threshold.

## Threshold Voltage

Each input operates as an analogue input. Inputs are active whenever the voltage present on the input passes the Threshold Voltage.

## Hysteresis Voltage

A Hysteresis Voltage can be set, so that once an input is turned on, it does not turn off again until the voltage drops below the Threshold Voltage plus the Hysteresis Voltage. This helps to reduce the effect of voltage noise problems falsely triggering an input and may also be used in conjunction with a thermo-sensor and radiator fan to stop fan flickering.

## Turn On Delay

A turn on delay can be set, so that the input is triggered after the delay period has passed.

## EMA Filter

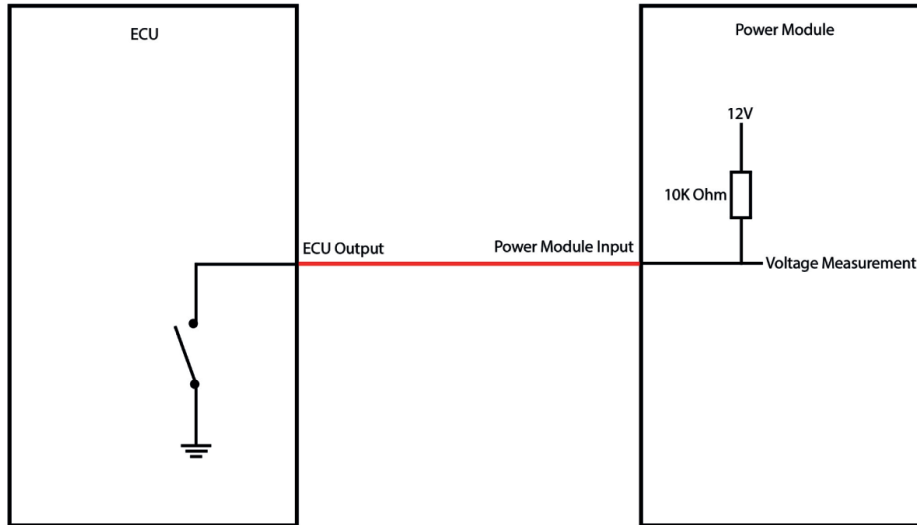
An EMA (Exponential Moving Average) filter can be applied to the input voltage. This can help to smooth out noisy input voltages. A higher EMA value will smooth the input voltage more.

# Inputs

## Input Resistor

Each input has a software selectable 10K $\Omega$  Pull Up and Pull Down resistor. The purpose of a Pull Up resistor is to 'pull' the voltage on the input up to the voltage on the RADLOK power connector. This is needed for ground connected switches, so that when the ground connected switch is pressed, the input voltage goes from high to low, through a threshold voltage. Conversely, a Pull Down resistor is used to pull the input voltage down to ground. This is needed when using a switch connected to power, so that when the switch is pressed, the voltage goes from low to high, through a threshold voltage.

### *Pull Up Resistor Example*





# Outputs

This section will detail how to configure an output for the PDM28 and for NIO Power Modules. Navigate to the Output Tab to configure the outputs. Each PDM28 output is High/Low sided. This means that both the High and Low side need to be configured independently. HS denotes High side, LS denotes Low side. If both the high and low side on a single output are instructed to turn on, only the high side will turn on. The low side will remain off until the high side is turned off.

The screenshot displays the configuration interface for Output 1 (HS). The top status bar shows: Output 1 HS, Status Off, Current 0.00A, Voltage 0.00V, Load 0.0%, and Enable buttons for Test, Configuration, and Enable.

**Configuration:**

- Trip Mode: Normal
- Low Fuse: 5.0 A
- High Fuse: 10.0 A
- Peak Fuse: 20.0 A
- Peak Fuse Time: 5.0 s
- Stay On Time: 1.0 s
- Turn On Delay: 0.0 s
- Clear Time: 0.0 s
- Retries: 2

**Graph:** A graph showing Current/A vs Time/s. The y-axis ranges from 0 to 30, and the x-axis from 0 to 10. The graph includes: Example Current Draw (yellow line), Over Current (red area), Operating Current (green area), and Under Current (blue area). The current draw peaks at approximately 18A at 2s and then settles around 8A.

**Function:** Output 1 HS On if -

**Logic:** Variable 1: True, Condition: AND, Variable 2: True, Constant: 0.0

**PWM:**

- Frequency: 100
- Soft Start Time: 100 ms
- Mapping: True
- Duty Cycle: Variable Value

**Duty Cycle Table:**

Duty Cycle	Variable Value
0%	0.00
10%	0.00
20%	0.00
30%	0.00
40%	0.00
50%	0.00
60%	0.00
70%	0.00
80%	0.00
90%	0.00
100%	0.00

**Graph:** A graph showing Variable Value vs Duty Cycle %. The y-axis ranges from -1 to 1, and the x-axis from 0 to 100. The variable value is constant at 0.00 across all duty cycle percentages.



# Outputs

## Output Status

Off=0, On=1, Tripped Over Current=2, Tripped Under Current=3.

## Output Current

How much current is flowing through the output.

## Output Voltage

The voltage on the output pin.

## Output Load

Ratio of the High Fuse current threshold to the output current, expressed as a percentage. If the load reaches 100%, it will trip from over current.

## Output Test

The user can manually test each output. The output must be enabled and configured in order to test it.

## Output Enable

Enables or Disables the output.

Pressing 'Expand' will open up the full configuration settings for that output.

## Trip Mode

**Instant Trip:** This mode will instantly turn the output off if the current flowing through the output goes over the current threshold. This setting is good for devices that need instant overcurrent protection.

**Normal Trip:** This mode acts like a conventional fuse and allows small current spikes over the current threshold without tripping. This setting is useful for devices that draw current in short bursts, such as ignition coils.

## Low Fuse

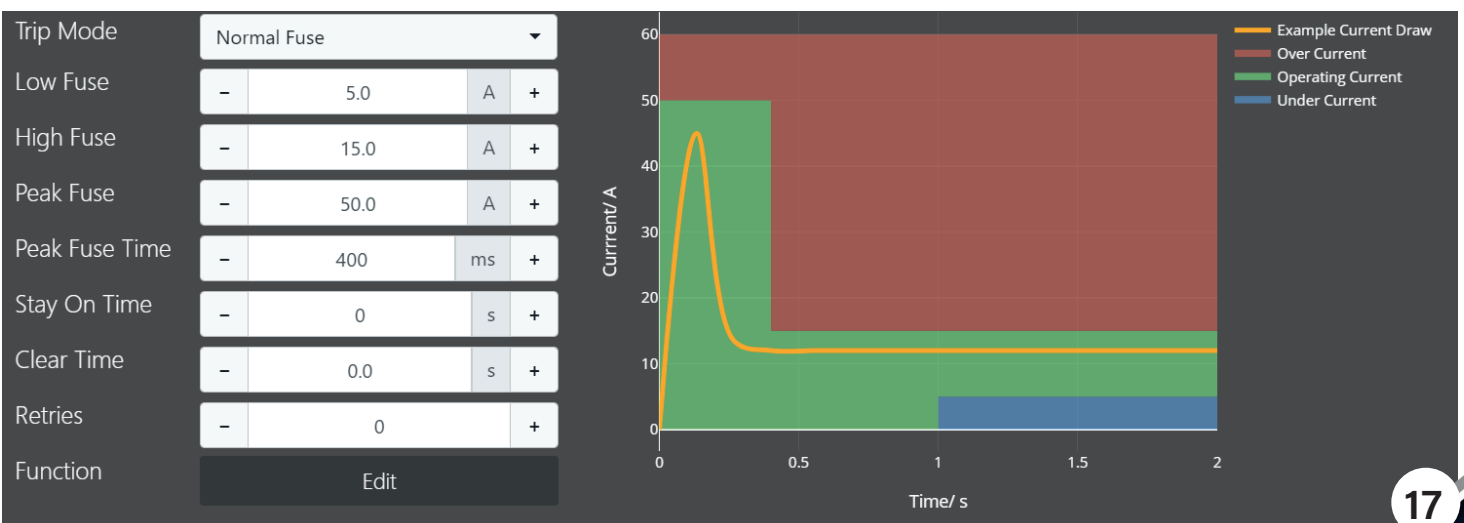
The low fuse setting will trip if the current falls below the threshold for over 1 second continuously. If this functionality is not needed then the value should be set to 0.

## 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.



# Configuration Software

## 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.

## 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.

## Retries

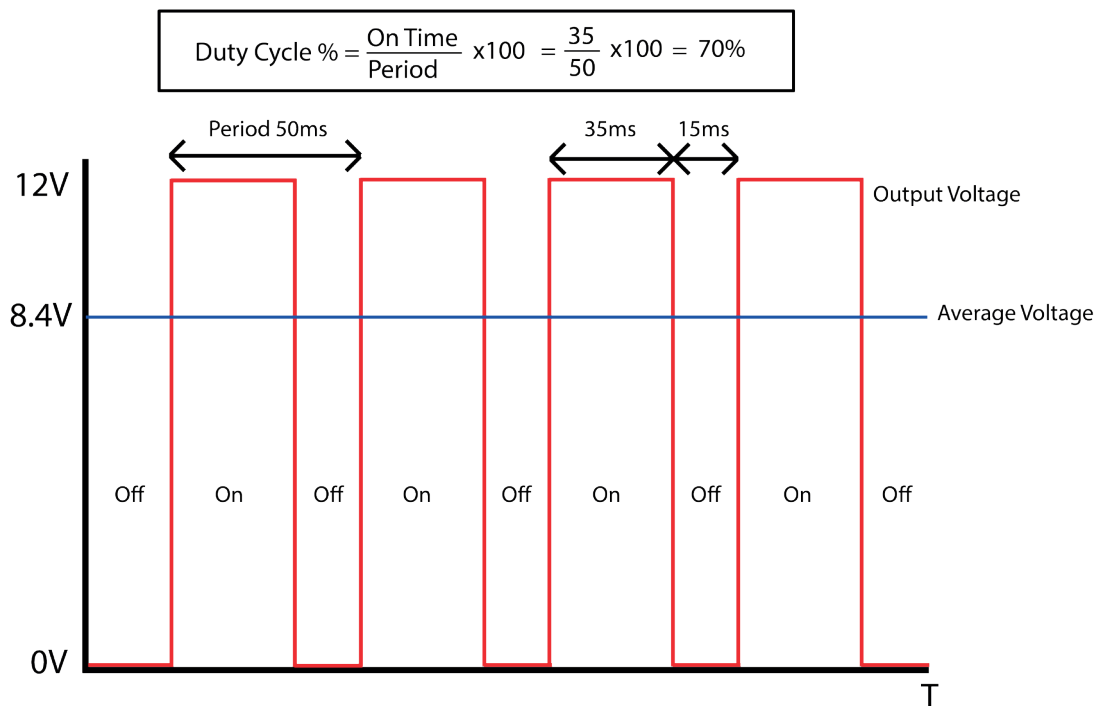
If an output trips due to an overcurrent condition, then that output may be 'retried' several times to try and reestablish normal operation of the output. The output channel will be retried for the number of times specified by this setting. A value of zero will result in continual retries of that output.

## Functions

Output Functions are used to control if an output should be On or Off. The PDM28 evaluates the logic function; if the function evaluates to True then the output will turn On, if the output evaluates to False then the output will turn Off. Examples of how to use logic functions will be given later in the instruction manual.

## PWM

All outputs on the PDM28 are PWM (Pulse Width Modulation) capable. Advanced, high speed switching circuitry allows for PWM operation at up to 1KHz with minimal heat dissipation. PWM is used to vary the average voltage present on the output. This is useful for varying the speed or intensity of motors or lights. PWM works by switching the output on and off at high speed, with the ratio of ON to OFF time (duty cycle) dictating the average voltage on the output. A lower duty cycle means that the output will be OFF for longer than it is ON, resulting in a low average voltage on the output. Conversely, a higher duty cycle means that the output will be ON for longer than it is OFF, resulting in a higher average output voltage. The duty cycle can vary between 0% (output always off) to 100% (output always on).



## Frequency

The PWM frequency dictates the speed at which the outputs are turned on and off. A higher frequency can result in smoother operation of the device which is being driven. Frequency is equal to 1/Period.

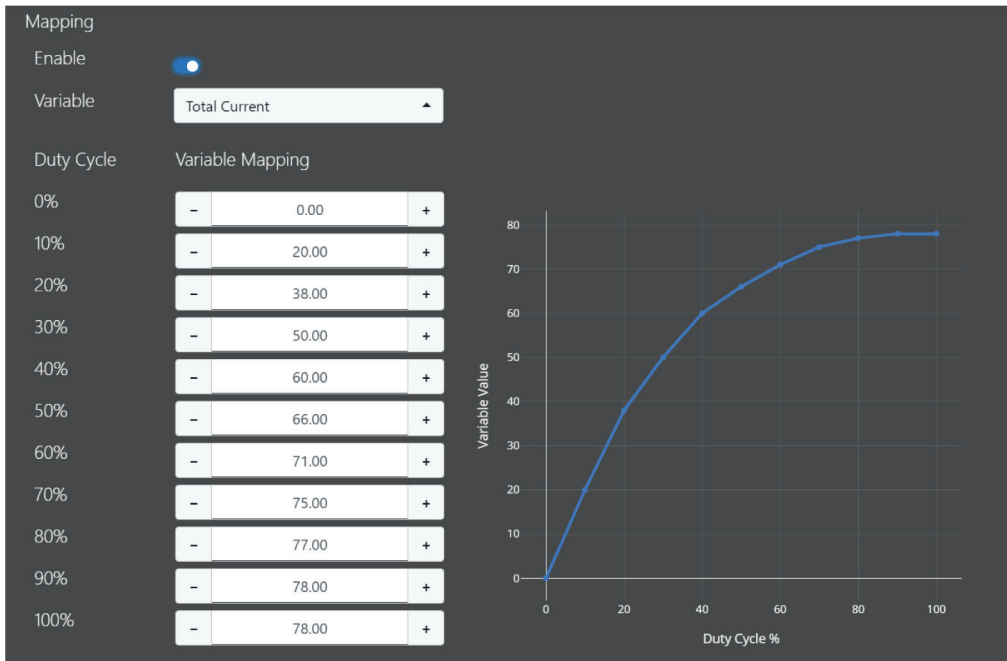
# Configuration Software

## Mapping

To vary the PWM duty cycle, it is necessary to 'map' the required value of duty cycle to a PDM28 variable. As the PDM28 variable changes, the PWM duty cycle changes. This allows the user to vary the PWM duty cycle on the output as desired.

To configure the PWM Mapping, first select a PDM28 variable from the drop down list.

For each value of duty cycle, a value of the PDM28 variable can be chosen. Eleven variable values must be chosen for the different values of PWM duty cycle. The duty cycle value is linearly interpolated between the values given. The variable value can either increase or decrease with duty cycle.

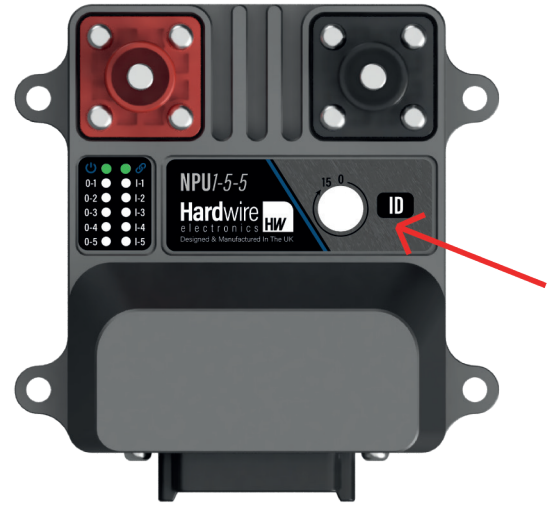


## Soft Start Time

Soft Start is a feature used to slowly turn on a device, by ramping the duty cycle on the output from 0% to 100% over a specified period. This results in the average voltage on the output increasing from 0% to 100%, slowly turning the device on. This feature is useful for starting devices such as radiator fans, which inherently have a large mechanical inertia. This large mechanical inertia means that the fan draws a large current initially to accelerate the fan blade, before the current drops back down to a base level when the fan reaches full speed.

# NIO Power Modules

Each Power Module in the system must be wired into the N-CAN bus according to the recommendations given in the Installation section. It is important that the termination resistors are installed in the appropriate position at each end of the bus, and that stub lengths are kept to less than 30cm in length. Failure to follow these recommendations may result in an unreliable system, or an inoperable one. Each NIO Power Module on the bus **MUST** have a different ID. The ID can be selected by removing the rubber grommet on top of the unit, and turning the selection dial with a small flat head screwdriver included in the kit. The NIO ID can be set between 0-15.



## General Settings

A NIO Power Module can be added to the system by navigating to the Power Module Tab, and clicking 'Add'. This will insert a Power Module template into the configurator. From here, an appropriate label and ID can be given to match that of the Power Module ID selected with the dial. Click the enable check box to enable the Power Module on the system.

Before configuring the Power Module any further, click send at the top of the screen. Apply power to the Power Module. If the system is configured correctly, the status of the Power Module should turn green and say 'Connected'. The connection LED on the Power Module should also be green.

If you do not see that the Power Module is connected, check that the CAN bus is wired appropriately and that the Power Module has power and ground.

If a successful connection has been made, the PDM28 will start to communicate with the Power Module and retrieve data. Click the Expand button, the following information will now be visible -

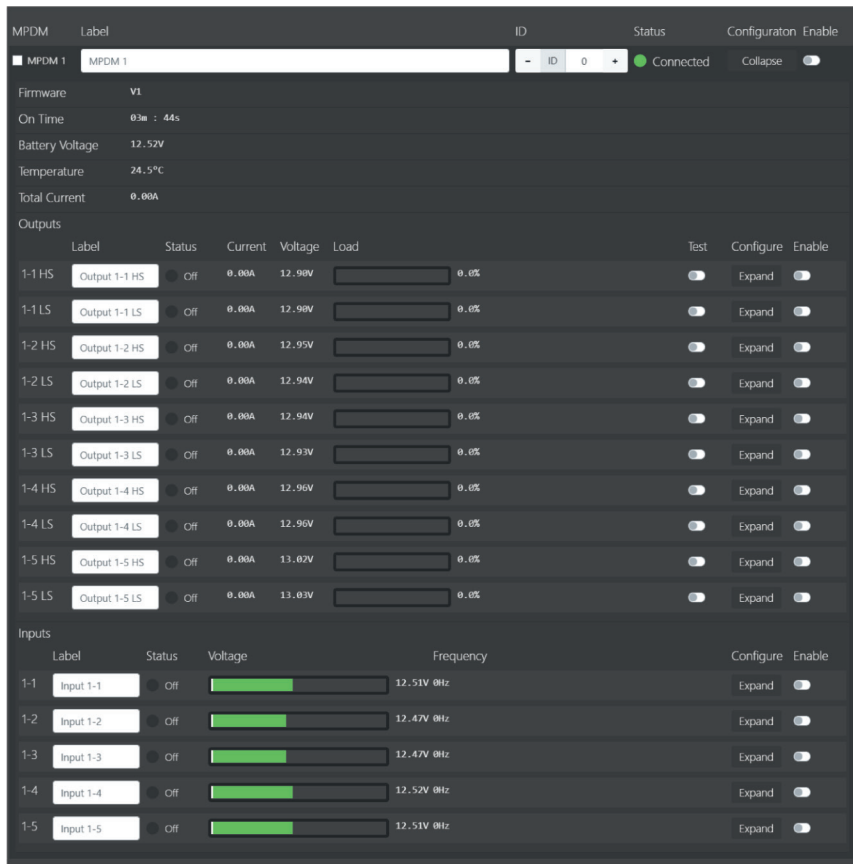
- Firmware Version - The current Power Module Firmware Version.
- On Time - How long the Power Module has been powered on.
- Battery Voltage - The voltage present on the power terminals of the Power Module.
- Temperature - The internal temperature of the Power Module.
- Total Current - The sum of the current flow on each output of the Power Module.

MPDM	Label	ID	Status	Configuraton	Enable
MPDM 1	MPDM 1	- ID 0 +	Connected	Collapse	<input checked="" type="checkbox"/>
Firmware	V1				
On Time	50s				
Battery Voltage	12.52V				
Temperature	21.3°C				
Total Current	0.00A				

On the following page is a diagram showing the settings for each output and input of the Power Module.

*Continued on next page...*

# NIO Power Modules



## Outputs

Each Power Module has 5 outputs. Each output is both high and low sided. This means that the output can either output a positive voltage or a ground. Internally, this can be thought of as a switch that can either switch the positive voltage to the output, or switch the ground to the output.

Both the high and low sides can be controlled individually. High side outputs are referred to as HS, and low side outputs are referred to as LS. The label format for each output is as follows, Power Module number - Output Number - HS/LS. For example, Output 2-5-LS would refer to Power Module 2, Output 5, Low side. Each Output is labeled in the left most column.

Each Output has two configuration rows for the High and Low side. Internal circuitry ensures that both the high and low side can not be active at the same time. If both the high and low side are commanded to turn on at the same time, then the high side will turn on and the low side will turn off. This prevents a straight short from power to ground.

NIO Power Module Outputs are configured in the same way as PDM28 Outputs in the previous section.

## Control Type

In addition to the PDM28 Output configuration settings, the NIO Power Module Outputs also have the Control Type Setting.

In conventional Power Distribution Modules, the outputs are controlled by the PDM directly, resulting in a low latency between the 'On command' and the output turning on. In the Node Power Distribution System, the PDM28 is used to evaluate the output function for each output on the NIO Power Modules. If the PDM28 evaluates that an output should be on, it must send a CAN message to the Power Module to turn its output on. The time taken to send the CAN message can be a couple of milliseconds. In most cases, this small time delay is negligible and does not affect the operation of the system. However, If the output needs to be turned on and off at high speed, then the latency can be too high. In this case, it may be beneficial to trigger the output based on an input on that Power Module, bypassing the PDM28 control altogether. The Control Type specifies the control scheme for the output. It can either be controlled by the PDM28 output function, or from any one of the inputs on that Power Module directly.

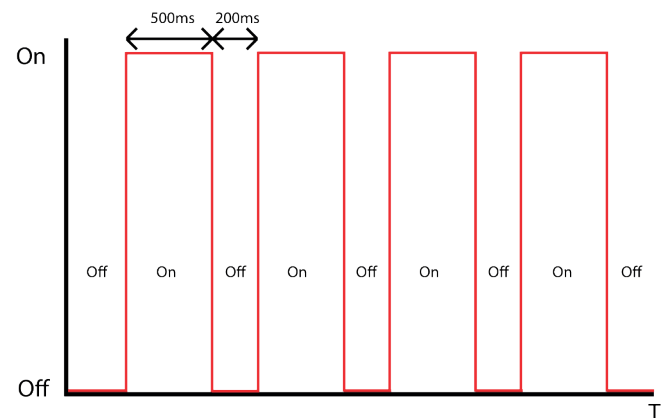
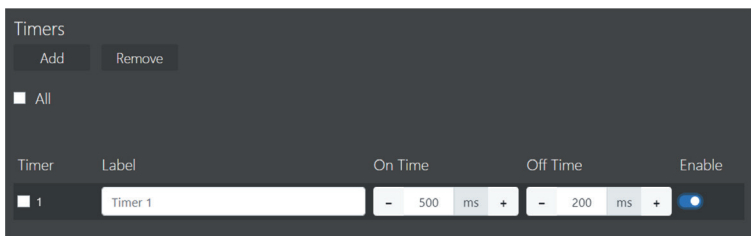
# NIO Power Modules

## Inputs

Each Power Module has 5 analogue inputs. The Inputs are configured in a similar manner to the PDM28 Inputs.

## Timers

The PDM28 has up to 30 individual On/Off Timers, which can be used in output functions to achieve advanced functionality such as Flashing indicator lights, Intermittent wiper motors, and more. To configure a Timer, go to the Timer Tab and click 'Add'. This will add a Timer into the current configuration. The Timer can then be given a label and be enabled. Each Timer has an ON and an 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.



## Counters

The PDM28 has up to 30 individual Counters. Counters are variables which can be incremented or decremented, and are triggered by logic expressions, such as when an input status equals true. 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 PDM output behaviors. To configure a counter, go to the Counter Tab and click 'Add'. This will add a counter into the current configuration. The counter can then be given a label and enabled. Click 'Expand' to expand the configuration for the counter. Counters are always given a Start Value. This is the value which the Counter starts from when the device is powered up, or when the counter is reset. The Loop Around Value is the maximum or minimum value that the counter will count to before looping back to the Start value.

# Counters

The Counter can be Incremented, Decremented, or Reset, based on three logic expressions. When the logic expression evaluates to 'True', the counter will Increment, Decrement, or Reset.

The Counters can count between a value of -2147483647 and +2147483647. Below is a typical example of a counter configuration.

The screenshot shows a configuration window for 'Counter 1'. At the top, there are tabs for 'Counter' and 'Label', and buttons for 'Configuration' and 'Enable'. The 'Counter' tab is active, showing a list of counters with '1' selected. The 'Label' field contains 'Counter 1'. To the right of the label are 'Collapse' and 'Enable' buttons. The main configuration area is divided into three sections: 'Increment', 'Decrement', and 'Reset'. Each section has a toggle switch and a logic expression configuration. The 'Increment' section has a toggle that is turned on. The 'Decrement' section has a toggle that is turned on. The 'Reset' section has a toggle that is turned on. Each logic expression configuration includes a 'Variable 1' dropdown, a 'Condition' dropdown, a 'Variable 2' dropdown, and a 'Constant' checkbox with a value field. For 'Increment', the condition is 'Equals' and the variable is 'True'. For 'Decrement', the condition is 'Equals' and the variable is 'True'. For 'Reset', the condition is 'Equals' and the variable is 'True'. Each logic expression also has a 'Constant' checkbox and a value field set to '0.0'.

# Maths Channels

The PDM28 has up to 30 Maths Channels. Maths Channels are an advanced feature which allow the user to create their own mathematical expressions which are evaluated by the PDM28. The value of the Maths Channels can then be used as needed.

To configure a Maths Channel, go to the Maths Channel Tab and click 'Add'. This will add a Maths Channel into the current configuration. The Maths Channel can then be given a label and be enabled. Click 'Expand' to configure the functionality of the Maths Channel.

The structure of the mathematical equation is similar to that of 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.

The Mathematical Equation is built up by adding lines to the equation. To add a line, click the three dots in the editor window. This will select that line to edit. The line can be configured by choosing the required mathematical operation from the drop down menu below. Click 'Add/Change' to add the line into the equation.

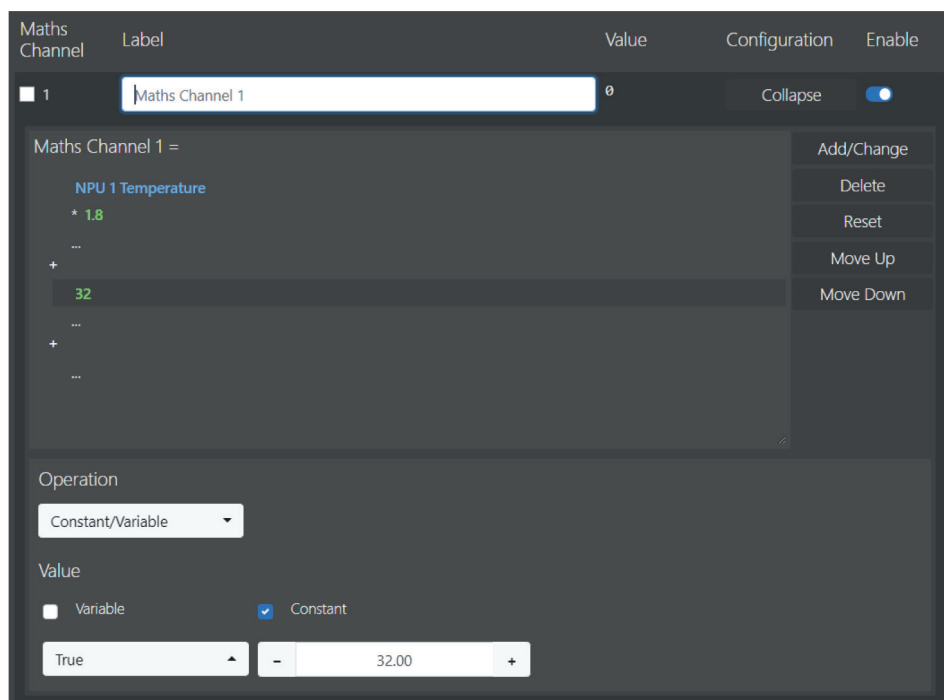
Equation lines can also be deleted, or the equation can be reset to default by clicking the 'Delete' and 'Reset' button respectively.

There are a number of mathematical operations that can be selected from the Operation drop down list, shown on the next page.



Mathematical Operators	
<b>Constant/Variable</b>	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
<b>Addition</b>	The Addition operator adds the two operands together. Each operand can be either a constant number or a PDM variable
<b>Subtraction</b>	The Subtraction operator subtracts the second operand from the first. Each operand can be either a constant number or a PDM variable
<b>Division</b>	The Division operator divides the equation section by the divisor. The divisor can be either a constant number or a PDM variable.
<b>Modulus</b>	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	
<b>AND</b>	Returns the bitwise AND operation of the two operands.
<b>OR</b>	Returns the bitwise OR operation of the two operands.
<b>XOR (Exclusive OR)</b>	Returns the bitwise XOR operation of the two operands.
<b>Left Shift &lt;&lt;</b>	Shifts the equation section to the left by between 0-32
<b>Right Shift &gt;&gt;</b>	Shifts the equation section to the right by between 0-32
Function Operations	
<b>Choose</b>	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.
<b>Min</b>	Returns the smallest of the two values
<b>Max</b>	Returns the largest of the two values.

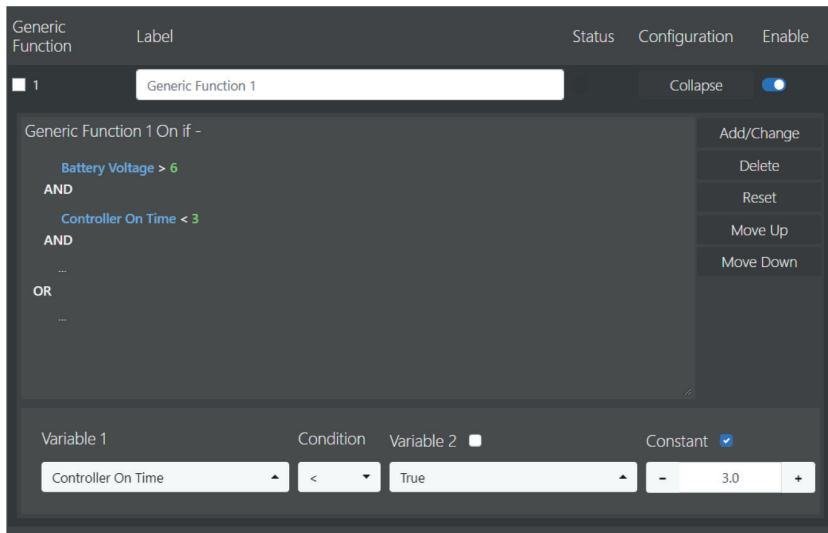
In the below example, a Maths channel is configured to convert the temperature of the NIO Power Module from degrees C to Fahrenheit.





# Generic Functions

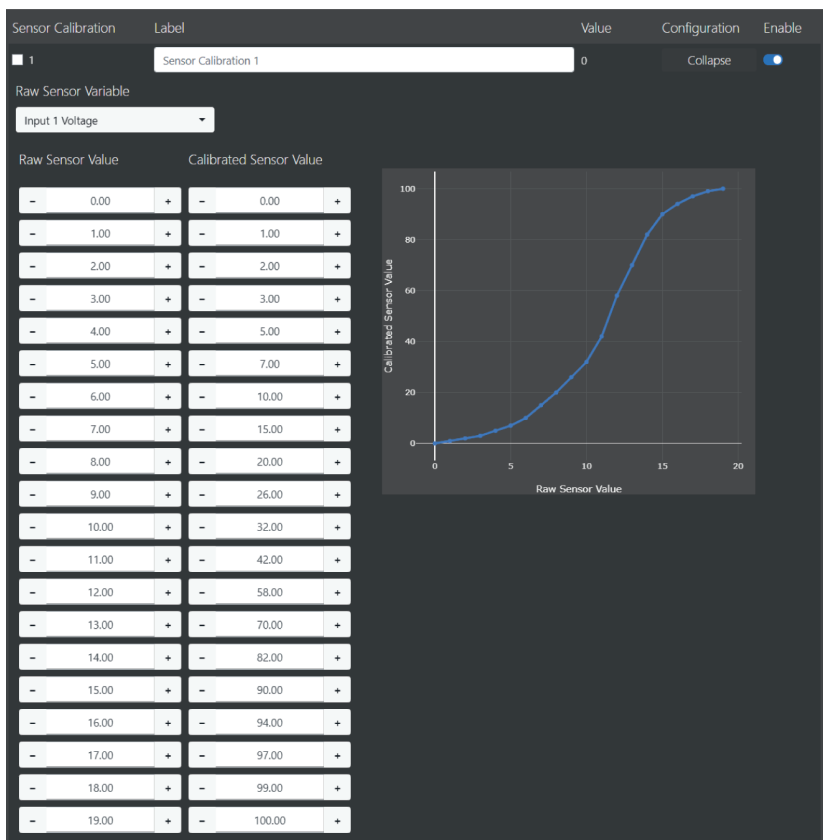
Generic functions operate in a similar fashion to the functions found in the global reset, global cutoff, and output sections. The function is evaluated constantly by the PDM, and the result (either 1 or 0) can be used elsewhere in the PDM for more advanced behaviour.



# Sensor Calibrations

Sensor calibrations allow one to take a raw value/sensor reading and scale/offset its value at different points. This can be used for example, to linearise a temperature sensor, or to scale a throttle position sensor between 0 and 100%.

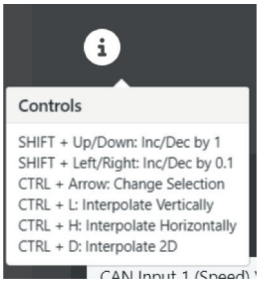
Select the raw sensor value (this can be any variable in the PDM), and then use the arrows to adjust the calibrated value to your desired place. Each raw sensor value will map to the calibrated value according to the graph, with linear interpolation between the points.



# 2D Tables

The PDM28 has 5 individual 20x20 2D Tables. 2D tables allow you to take two input variables and map their values to an output.

First start by selecting your two input variables. In this made-up example, vehicle speed and coolant temperature has been selected, and the output value can be used for radiator fan PWM duty cycle %. You can then select the range that these variables will fall within. In this case, we would only be interested in a vehicle speed greater than 0 and less than 200, and a temperature greater than -40degC and less than 150degC. The values at each point in the table can then be input as desired.



Hover over the 'i' Icon to see the keyboard shortcuts which can be used for interpolation and mass data change.

In the example below, when variable X (speed) is 0 and variable y (temperature) is 0, the fan can be turned off (0% duty cycle). When the speed is 0 and the temperature is 140, the fan needs to be turned on fully (100% duty). When the speed is 150 and the temperature is 90, the fan can be turned down (10% duty).

2D Table

Label: Fan\_Duty\_ Value: 0.00 Configuration: Collapse Enable:

Variable X: CAN Input 1 (Speed) Value Variable Y: CAN Input 2 (CLT) Value

X Minimum: 0.00 Y Minimum: -40.00

X Maximum: 200.00 Y Maximum: 130.00

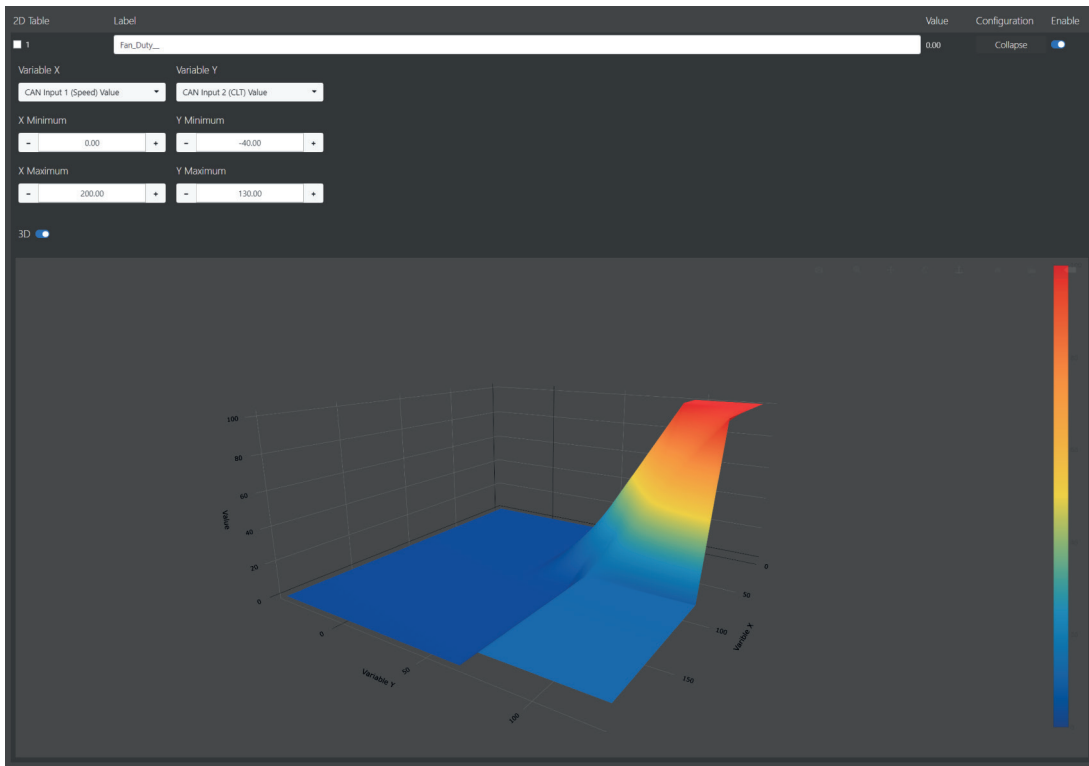
3D:

		Variable X																				
		0,00	10,53	21,05	31,58	42,11	52,63	63,16	73,68	84,21	94,74	105,26	115,79	126,32	136,84	147,37	157,89	168,42	178,95	189,47	200,00	
Variable Y	-40,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
	-31,05	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
	-22,11	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
	-13,16	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
	-4,21	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	
	4,74	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	13,68	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	22,63	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	31,58	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	40,53	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	49,47	16,67	16,67	16,67	14,34	12,00	9,67	7,33	5,00	3,33	1,67	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	58,42	33,33	33,33	33,33	28,66	24,00	19,33	14,67	10,00	6,67	3,33	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	67,37	50,00	50,00	50,00	43,00	36,00	29,00	22,00	15,00	10,00	5,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
	76,32	66,67	66,67	66,67	57,34	48,00	38,67	29,33	20,00	16,67	13,33	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	85,26	83,33	83,33	83,33	73,33	63,33	53,33	43,33	33,33	25,55	17,78	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	94,21	100,00	100,00	100,00	89,33	78,67	68,00	57,34	46,67	34,45	22,22	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	103,16	100,00	100,00	100,00	92,00	84,00	76,00	68,00	60,00	43,33	26,67	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	112,11	100,00	100,00	100,00	94,67	89,33	84,00	78,66	73,33	52,22	31,11	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	121,05	100,00	100,00	100,00	97,33	94,67	92,00	89,34	86,67	61,11	35,56	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00
	130,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	100,00	70,00	40,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00	10,00

Continued on next page...

## 2D Tables

Clicking the 3D button allows you to view the table in 3-Dimensions.



## CAN Keypads

Using CAN bus 1 and 2, the PDM28 can communicate with up to 4 individual CAN bus Keypads. The PDM28 is able to communicate with the following keypad devices -

- Blink Marine CANOpen 4-15 button keypads.
- Grayhill CANOpen 6-20 button keypads.
- Motec 6-20 button keypads.
- Marlin Technology J1939 8 button keypad.

Keypads are a great way to control the functionality of the Power Distribution System. The Keypad buttons can be used to control the state of multiple outputs, and the LED indicators on each button provide visual indicators to the user.

To install a Keypad into the system, make sure that the CAN bus lines are connected to either CAN bus 1 or 2, and that the wiring is done in accordance with the instructions given in the CAN bus wiring section. Up to 4 Keypads can be used on the same CAN bus.

It is important that the Keypads do not have the same ID. If the Keypads have the same ID, then the system will not work. It is also important that all of the Keypads are configured to run at the same CAN bus speed.

### Keypad Configuration

To configure a Keypad, go to the CAN Keypad tab and select one of the four available Keypads to configure. Click the enable button to enable the keypad, and select the appropriate CAN bus which the Keypad is installed on.

Choose the keypad which is being used from the drop down menu.

Choose the appropriate ID which the keypad is configured to. Make sure to observe that the Keypad ID is entered in the decimal format in the configuration software.

*Continued on next page...*

# CAN Keypads

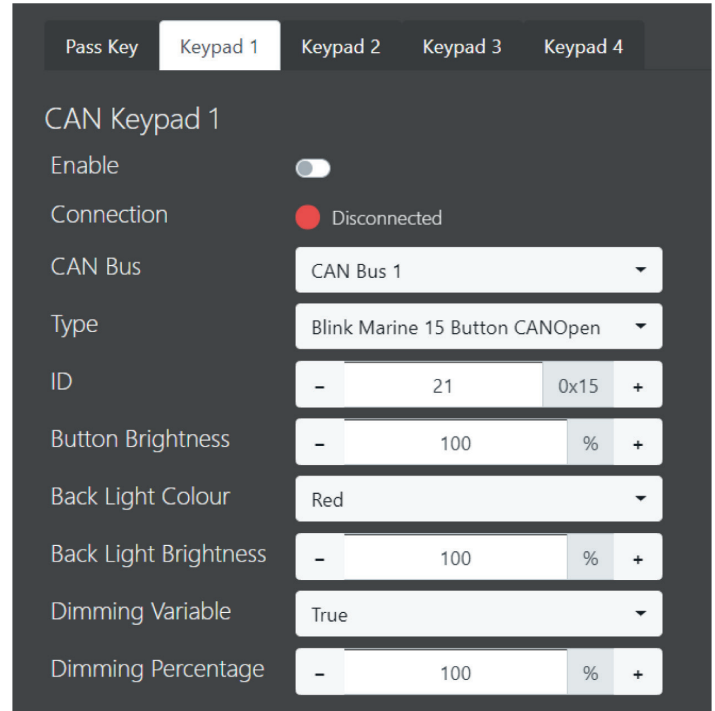
If the keypad is to be powered from a PDM output, on a Power Module, set up that output and enable it. Press 'Send' to send the configuration to the Controller Module. The Keypad should now light up with the backlight colour given in the Keypad configuration tab, and the Connection status should turn Green and say 'Connected.'

## If the Keypad does not light up, check the following is correct -

- The CAN bus speed of the Keypad and PDM28 is identical
- The CAN bus is wired correctly in accordance with the recommendations given in the CAN bus wiring section.
- The correct CAN bus is selected in the Keypad configuration
- The ID of the Keypad is the same as the ID in the Keypad configuration.

Once the Keypad has successfully connected with the PDM28, the Keypad functionality can be configured.

The Button Brightness, Backlight Colour, and Backlight Brightness can be selected from the drop down menus. The Dimming Variable is used to dim the keypad lights when the Dimming Variable is equal to True. This is useful for night time operation. The Dimming Percentage is the brightness of the Buttons and Backlight as a percentage of the Button Brightness and Backlight Brightness settings.



## Keypad Button Configuration

Keypad 1 - Button1

Mode: States

States: 3

Hold To Reset:

	Off State	State 1	State 2	State 3	Fault State
LED Colour	Nothing	Green	Blue	Violet	Red
LED Condition	True	True	True	True	Output 1 Trip

Keypad Dial

Minimum: 0

Maximum: 16

LED Offset: 0

Save

Reset

Each Keypad Button can be given a label and be enabled. Click the 'Expand' Button to configure the operation of the Keypad Button.

The Mode of the button can be either Momentary, Latched, or States.

In **Momentary Mode**, the Keypad Button State will be True when the button is pressed, and False when it is not pressed.

In **Latched Mode**, the Keypad Button State will toggle between True and False each time the Button is pressed.

In **States Mode**, each time the Button is pressed, the Button State will increment. The number of states can be set between 1 and 3. When the Button State reaches the limit, it will return back to 0.

Hold To Reset is used to reset the Button state back to 0 when the button is held for one second.

For each Button State, the Button Colour and functionality can be set. The first drop down menu selects which colour the Button will be in that state. The second drop down menu selects a variable which is used to say when the button is that colour.

For instance, if the colour is selected to be Green, and the variable is selected to be 'True' - when the button is in that state, the button will be Green constantly.

If the colour selected is Green, and the variable selected is 'NIO1 Output 1 Status', then the button will be Green when in that state AND when the state of 'NIO1 Output 1 Status' is True.

This is a useful feature, as the Keypad Button Colour can be changed based on the state of a variable. This, for example, allows the button to flash in time with a turn signal output, indicating to the user that the output is flashing.

The Keypad Dial setting is used with the Blink Marine PKP-3500-SI-MT Keypads which have two rotary encoder dials on them. Each dial can take a value of between 0-16. The LED Ring around the outside of the dial indicates what the dial value is. The Dial can have a minimum and maximum value. The LED offset offsets the start position of the LED ring around the dial. When the 'Hold To Turn' checkbox is enabled, the rotary dial must be pressed in and turned for the dial value to change.

For a full video explanation of this feature,  
check out the video in this QR code.



## CAN Keypads

### Pass Key

As a layer of security for the Power Distribution System, a Pass Key can be added to the Keypads. If the Pass Key is enabled, then the Keypads will not be operational until the correct sequence of Keypad Buttons has been pressed. This can stop unwanted users from potentially starting up a vehicle and driving it away.



## CAN Bus Basics

**For a full detailed video explanation of the CAN Bus, check out our beginners guide which can be found in the following QR code**



# CAN Inputs

The PDM28 has two user accessible CAN bus ports, which can be used to interface with other devices in the vehicle - such as ECUs, Digital Dashboards, Sensors, and more. Up to 100 individual CAN Inputs can be configured, spread across either CAN bus port.

The CAN bus on a vehicle can become heavily crowded with CAN messages from different devices on the bus. Configurable hardware filters are used in the PDM28 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. Once the CAN message has passed through the acceptance filters, the CAN message is then processed by a CAN Input with matching ID.

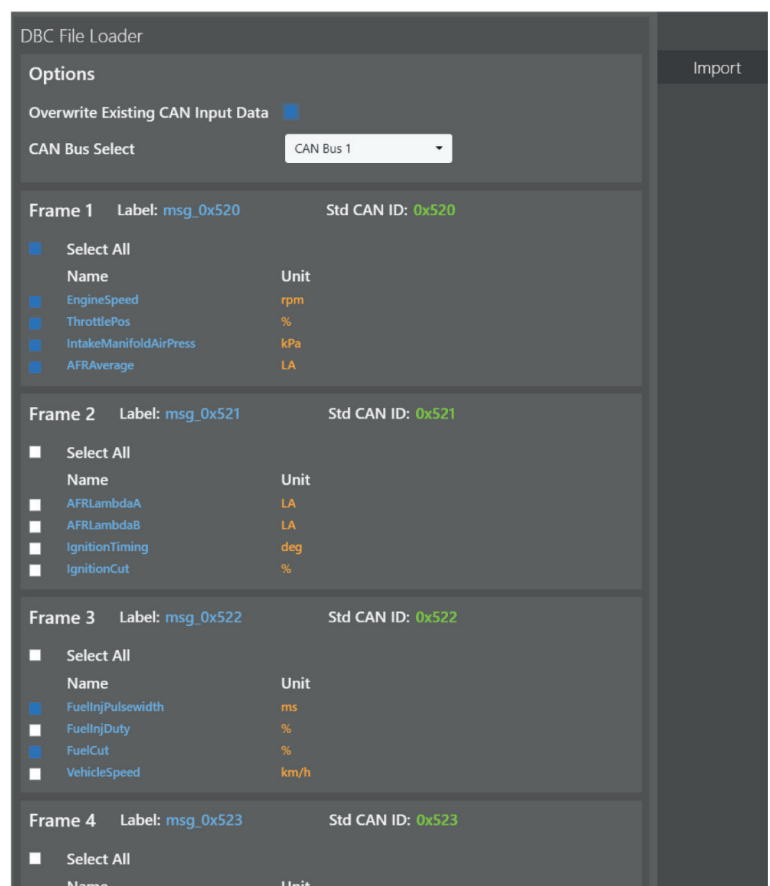
## CAN Configuration Files

To store the configuration of the CAN inputs on the PDM28, the configuration can be saved as a file. A CAN Configuration file can also be loaded into software if needed. The CAN Configuration files are specific to the Hardwire Electronics PDM28s.

## DBC File Loader

.DBC Files are a standard used to specify how CAN messages are sent on a CAN bus. For example, an ECU manufacturer may have a DBC file which describes how the CAN data it sends out is formatted. The DBC file can then be used by another device to set up its own CAN inputs automatically.

The configuration software allows one to load up a DBC file, and choose which signals are required. Press 'Import .DBC File', and select the DBC file which needs to be loaded. A new window will appear which shows all of the available signals in the DBC file. The correct CAN bus can be selected at the top, and each DBC file signal can be selected with the checkboxes below. Once all the required signals have been selected, press Import. This will automatically configure the PDM28 CAN Filters and CAN Inputs. By default, the CAN Filters and CAN Inputs from the DBC file will be added onto the existing CAN Filters and CAN Inputs in the configuration. To overwrite the current CAN Filters and CAN Inputs, click the 'Overwrite Existing CAN Input Data' checkbox before pressing Import.

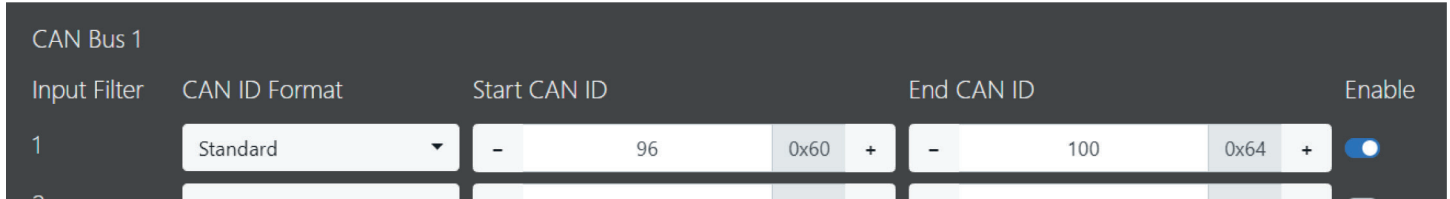




# CAN Inputs

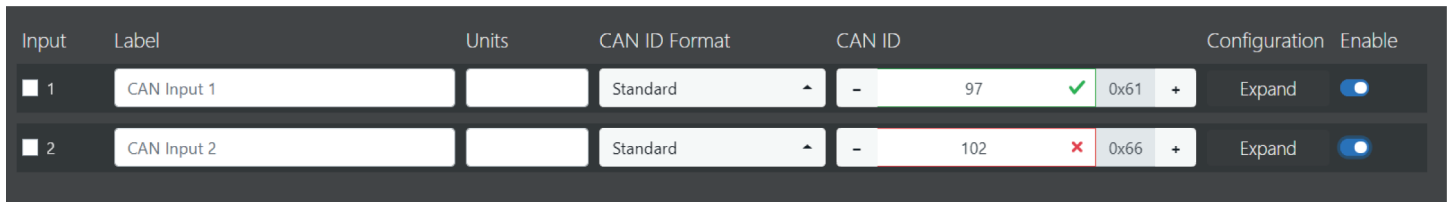
## CAN Bus Filter Configuration

Each CAN Bus Port on the PDM28 has 10 dedicated hardware based CAN Filters. Each Filter can filter either Standard or Extended ID CAN messages. The Filters have a Start and End CAN ID. CAN Messages with an ID within the Filter range pass through, and those that do not are discarded. Filters can be enabled and disabled by clicking the checkbox.



## CAN Input Configuration

To add a CAN Input, click the 'Add' Button. Up to 100 CAN Inputs can be added to the configuration. The CAN Input can be given a Label and a Unit, useful for viewing data in the Monitoring Tab. The CAN ID Format can be selected as either a Standard or Extended ID. If the CAN ID is valid, it will have a green tick next to it. If the CAN ID is invalid, it will have a red tick next to it. The CAN ID is valid if it falls within one of the CAN Filter ranges.



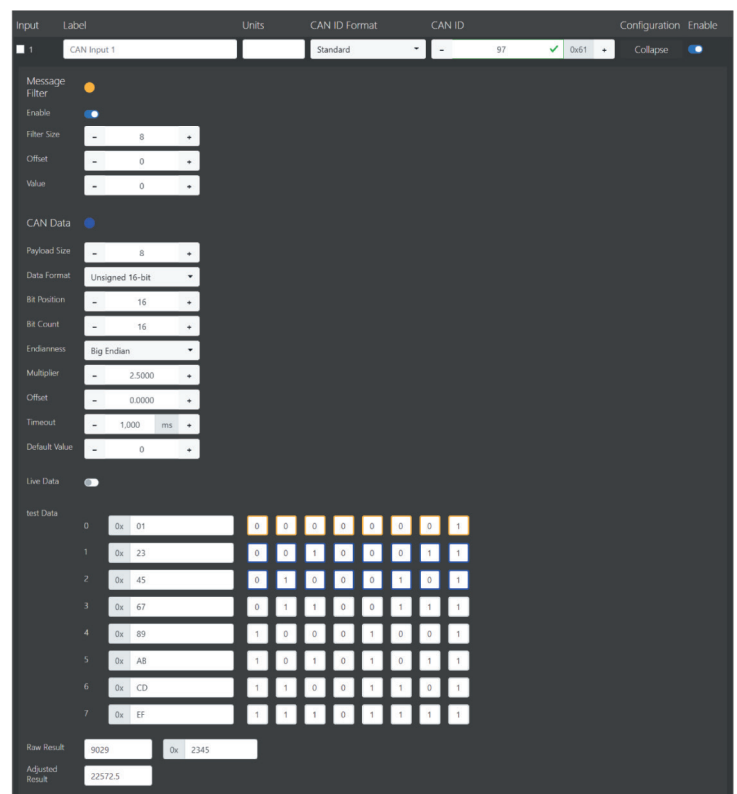
Click Expand to configure the remainder of the CAN Input settings. In the dropdown, all of the parameters for the CAN Input can be configured.

The aim of the CAN Input is to parse off the appropriate data from the payload bytes of the CAN message. A CAN Message can have up to 8 payload bytes. Within the payload bytes may be a number of different variables. For example, the first byte may contain engine RPM, the second byte may contain water temperature etc. The CAN Input is configured to parse off the specific variable value which is required.

## Test Data

To make configuring the CAN Inputs easier, test data has been added to the bottom of the window. The test data is simply 8 data bytes which mimic the data that would reach the CAN Input. The Blue squares represent the data which is being parsed from the payload. The Orange Squares represent which bits the Message Filter uses. The resulting CAN Input variable value is shown below.

Once an appropriate CAN Input Filter has been enabled, the 'Live Data' check box can be clicked. This will show the live data on the CAN bus in the test data bytes. The CAN Input can then be configured to correctly parse the data from the payload in real time.





# CAN Inputs

## Message Filter

In addition to filtering the CAN messages by ID, the PDM28 can also further filter the CAN messages by data in the payload. This is useful when a device on the CAN bus uses compound or segmented CAN messages to send data. Compound CAN messages are used to send different types of data with the same CAN ID. Usually, the first byte of the payload acts as a 'sub ID', and increments in value for each data type being sent.

The Filter Size in bits, the Offset in bits, and the numerical value to filter out, can all be configured with the plus and minus arrows. The test data at the bottom of the configuration window shows how the bits of the payload which will be filtered, highlighted in orange.

## Payload Size

A CAN message can have between 0 and 8 data bytes. The CAN Input payload size must exactly match the payload size of the CAN Input for the message to be received.

## Data Format

The format of the data in the payload can change depending on the type of data being sent. Data which can have negative values is called 'signed' data, and data which is positive only is called 'unsigned' data. The data can also have a different size, between 8 and 32 bits.

## Bit Position

The Bit Position chooses the bit which the data is parsed from. Bit 0 is the least significant bit, and bit 7 is the most significant bit.

## Bit Count

The Bit Count sets how many bits are parsed, starting from the Bit Position Value.

## Endianness

Endianness describes the order that data bytes are sent for multiple byte variables. Big Endian order means that the most significant byte is sent first with the least significant byte after. Little Endian order means that the least significant byte is sent first with the most significant byte after.

For example, consider the following CAN Message payload -

Payload Bite	0	1	2	3	4	5	6	7
Value	0x01	0x23	0x45	0x67	0x89	0xAB	0xCD	0xEF

The variable to parse is a 16-bit variable on Byte 1 and 2. If the data is in Big Endian format, the 16-bit variable would be 0x2345. If the data was Little Endian format, the 16-bit variable would be 0x4523.

## Multiplier

The previous CAN settings are responsible for parsing the appropriate data bits from the CAN message payload. The Following settings are responsible for scaling and offsetting the raw data so that it is in the correct format.

The Multiplier takes the raw variable value and multiplies it with a constant. The Multiplier value can range from 0.0001 to 1000.

A multiplier is often used to preserve greater accuracy in the data being sent. For example, if an ECU needs to send a value of 8.36 over the CAN bus, then it would first need to multiply the value by 100 and send the resulting 836 instead.

Then the receiving device can apply a multiplication of 0.01 to the raw data to return the 836 value back to the original 8.36, reserving the two decimal places of precision.

# CAN Inputs

## Offset

After the raw value has passed the multiplier step, an offset can be added. This can be used to adjust the value for calibration purposes. For example, if the value of 8.36 is actually supposed to be 9.00, then an offset of 0.64 can be applied.

## Timeout

The Timeout value is how long the current CAN value is valid for in milliseconds. For example, If it is known that data should be received every 0.5s, but no data is received for 1s, then it can be assumed that an error has occurred. The CAN variable value will then return to a default value.

## Default Value

The Default Value is the initial value of the CAN Input on device start up, or when the timeout has occurred.

# CAN Outputs

The PDM28 has up to 100 individual user defined CAN Outputs which can be configured on either CAN bus 1 or 2. In addition to the user defined CAN Outputs, the user can enable the CAN Output stream, which sends preformatted data at a regular interval.

## User CAN Outputs

A CAN Output can be added by clicking the 'Add' Button. A Label can then be given to that CAN Output. Click 'Expand' to further configure the CAN Output.

Output	Label	Configuration	Enable
1	CAN Output 1	Collapse	<input checked="" type="checkbox"/>

CAN Bus	CAN Bus 1
ID Format	Standard
ID	120 0x78
Send Mode	Periodic
Send Frequency	10 Hz
Trigger Variable	True
Payload Size	4 Bytes

Channel	Variable	Data Format	Multiplier
Channel 1	Total Current	16-bit	1
Channel 2	CAN Input 1 Value	16-bit	10
Channel 3	True	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

# CAN Outputs

## CAN Bus Selection

The CAN Bus that the CAN Output message will be sent on can be selected from the drop down menu.

## CAN ID

The CAN Message ID format can then be selected as either Standard or Extended, and the value for the ID can be chosen.

## Send Mode

The CAN Outputs can either be sent periodically, or be sent on a trigger.

When the Send Mode is set to Periodic, the CAN Output will send at the interval defined by Send Frequency. When the Send Mode is set to Triggered, the CAN Output will send every time the Trigger variable goes from False to True. This is useful for sending a CAN Message when a certain event takes place.

## Payload

A CAN Message can have between 0 and 8 bytes. This is selected with the Payload Size setting.

The Payload is filled with PDM28 Variables. An 8-bit value will take up 1 byte, a 16-bit value will take up 2 bytes etc. To configure a Channel, select the variable which needs to be sent. The Data Length can then be changed. The appropriate Data length must be selected so that the variable does not 'overflow.' For example, an 8-bit variable can hold a number between 0 and 255. If the PDM28 Variable exceeds a value of 255, then the value will overflow, and loop back to 0. If an 8-bit variable was used to send a value of 260, then the value sent would be 4 due to an overflow. Here, a 16-bit value should be used instead.

A Multiplier can be added to the data to be sent. This is used to preserve precision in the sent data. For example, if the PDM28 Variable to send is 15.4, then without a Multiplier, the value sent would be 15 which loses the decimal place. Here, using a Multiplier of 10 would mean a value of 154 would be sent. The receiving CAN device can then divide the value by 10 to return to the correct value of 15.4.

# CAN Output Stream

In addition to the customisable CAN outputs, there is an option to enable the PDM CAN stream.

The PDM CAN stream is a collection of PDM variables which have been placed in a user friendly format which can be exported as a DBC file and used by other devices on the bus. There is lots of data which can be sent in the CAN output stream, so you can select the data you want to send out to minimise bus load. You can also vary the send rate of the data for the same effect. An offset can be applied to the CAN stream IDs if needed, to help avoid any CAN ID clashes.

Frame	ID	DLC	Send Rate	Description	View	Enable
0	4096 (0x1000)	8	10 Hz	Date	Collapse	On
1	4097 (0x1001)	8	10 Hz	PDM Stats 1	Expand	On
2	4098 (0x1002)	8	10 Hz	PDM Stats 2	Expand	On
3	4099 (0x1003)	8	10 Hz	IMU Acceleration	Expand	On
4	4100 (0x1004)	8	10 Hz	IMU Rotation	Expand	On
5	4101 (0x1005)	8	10 Hz	IMU Gyroscope	Expand	On
6	4102 (0x1006)	8	10 Hz	Output 1 HS	Expand	On

# LIN Bus

LIN Bus is a low cost communication protocol primarily used for interfacing with sensors and actuators. The speed of the bus is low when compared to CAN bus, and the data rate is hence limited. The low speed however means that only a single wire is needed, reducing wiring complexity significantly.

Unlike CAN bus, LIN bus is very structured in the way that data is sent and received. There is no bus arbitration, and so each LIN 'node' sends its data in a predetermined window of time to avoid bus conflicts. The structured nature of LIN bus necessitates the use of an 'LDF' file. An LDF file is analogous to a DBC file.

It describes the following -

Which nodes are on the bus

Which signals are used

How the signals are encoded/decoded

How the signals are combined into frames

Which nodes send/receive each frame

What order and time period are each frame sent.

In most cases, the manufacturer of the LIN bus node will have an LDF file made up, which can be imported into the Hardwire Electronics configurator software. To help explain each part, an LDF file for a water pump has been loaded into the configurator software.

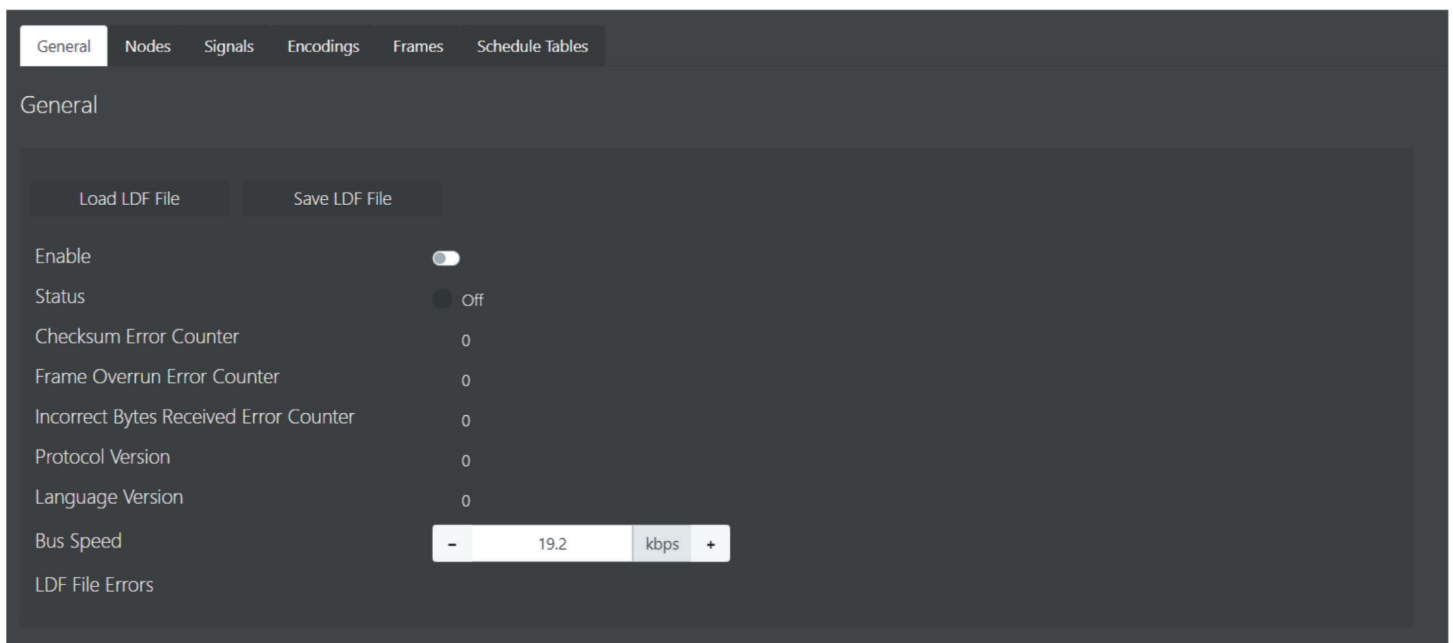
## General

In the general tab, you can load in an LDF file. It is not completely necessary to use an LDF file, but it is difficult to not use one, as you will need to know the LIN node specific parameters.

Click enable to enable LIN bus on the PDM.

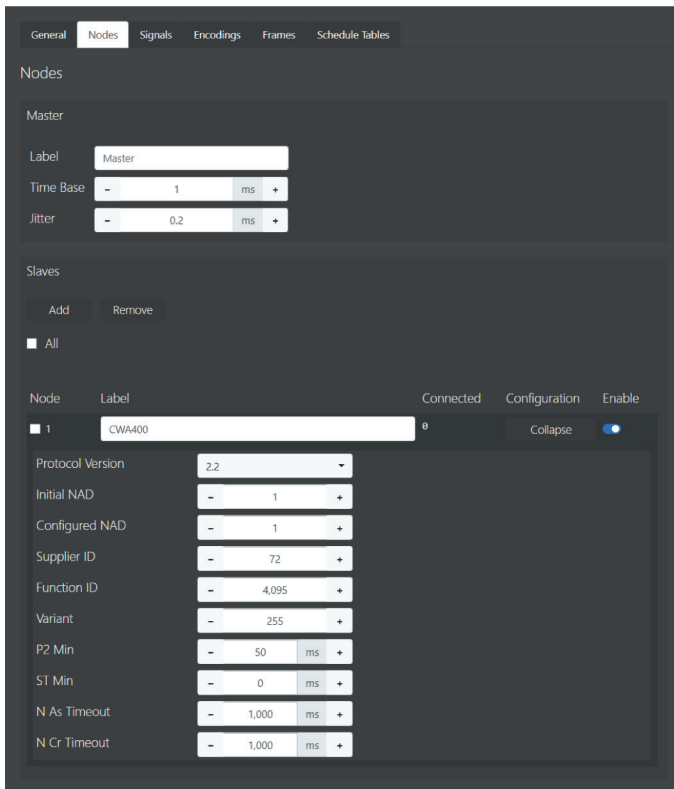
The status of the LIN bus, and various error counters are listed below.

If the checksum, frame overrun or incorrect bytes error counters continue to increase, the LIN bus status flag will turn red - indicating a problem. The corresponding LED on the PDM will also turn red.



## Nodes

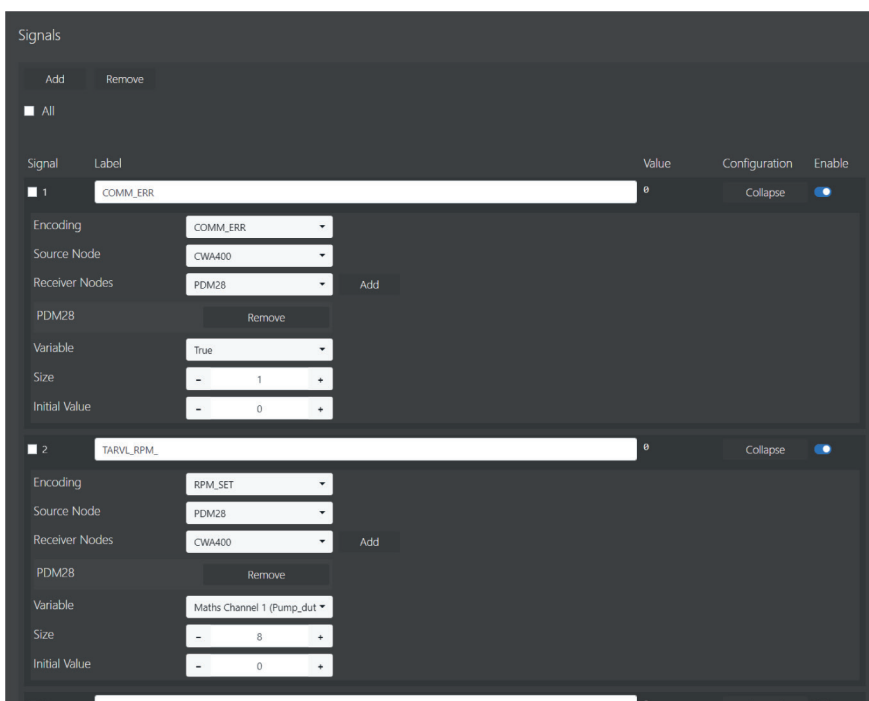
In the Nodes tab, we first have the Master node which is the PDM. Below this we have each slave which is on the Bus. Multiple slaves can be on the bus. The parameters for the nodes is taken from the LDF file for the water pump.



## Signals

The signals tab is where the LIN bus signals are defined. The signals can go from the PDM to the pump (to control the pump), or from the pump to the PDM (to get information from the pump).

In this example the first signal goes from the pump to the PDM, and contains a communication error status message. The second signal goes from the PDM to the pump, and contains the desired RPM the pump motor should run at. With signals that travel from the PDM to another LIN node, you must select the data which you wish to be sent. In this case, a maths channel which has the desired pump RPM is sent.



## Encodings

Each signal sent on the LIN bus has an associated encoding. This tells us how the raw data should be interpreted. Continuing on from the previous section on signals, the encodings for the communication status error, and pump RPM are shown below.

The communication status error is a logical value of 1 or 0, representing error or no error respectively. The pump RPM is a physical value with a varying value. It is scaled and offset to get the actual RPM value, and windowed between a minimum and maximum of 0-5000. The raw value which is sent, is the maths channel variable shown in the signals tab.

The screenshot shows the 'Encodings' configuration window for the signal 'COMM\_ERR'. At the top, there are 'Add' and 'Remove' buttons. Below them is a list of encodings with a checkbox for '1'. The main configuration area includes a 'Type' dropdown set to 'Physical Value', a 'Text Info' field, and four numeric input fields: 'Range Minimum' (0), 'Range Maximum' (0), 'scalar' (0.0), and 'Offset' (0.0). Below these are 'Add' and 'Change' buttons. The 'Encoding Parts' section contains a table with two rows: 'Logical Value' 'NoError [0]' and 'Logical Value' 'Error [1]', each with a 'Remove' button.

Encoding	Label	Configuration
1	COMM_ERR	Collapse

Type: Physical Value

Text Info:

Range Minimum: - 0 +

Range Maximum: - 0 +

scalar: - 0.0 +

Offset: - 0.0 +

Encoding Parts

Type	Details	
Logical Value	NoError [0]	Remove
Logical Value	Error [1]	Remove

The screenshot shows the 'Encodings' configuration window for the signal 'RPM'. It features the same layout as the 'COMM\_ERR' window. The 'Type' dropdown is set to 'Physical Value'. The 'Encoding Parts' section contains a table with one row: 'Physical Value' 'RPM = 23.6 \* (Raw Value) + 0 [0, 5000]', with a 'Remove' button.

Encoding	Label	Configuration
7	RPM	Collapse

Type: Physical Value

Text Info:

Range Minimum: - 0 +

Range Maximum: - 0 +

scalar: - 0.0 +

Offset: - 0.0 +

Encoding Parts

Type	Details	
Physical Value	RPM = 23.6 * (Raw Value) + 0 [0, 5000]	Remove

## Frames

In this simple example with a signal water pump and a PDM28 on the CAN bus, only two frames are sent. One frame is sent from the PDM to the pump, and the other frame is sent from the pump to the PDM. Each frame is constructed of individual signals with offsets - piled together into the 8 byte LIN bus packet payload.

The frames are unconditional frames, as they are sent unconditionally in a repeated fashion according to the schedule table.

The screenshot displays a software interface for configuring LIN bus frames. It shows two frames, Frame 1 and Frame 2, with their respective parameters and signal mappings.

**Frame 1: STATUS\_AUWP**

- ID: 37
- Size: 8 Bytes
- Publisher: CWA400
- Signal: COMM\_ERR
- Send Count: 0
- Configuration: Collapse
- Enable:

Signal	Bits	Offset	Action
ST_TARVL_RPM_AUWP_LIN	8	0	Remove
AVL_VL_RPM_AUWP_LIN	8	8	Remove
U_AUWP_LIN	8	16	Remove
TEMP_AUWP_LIN	8	24	Remove
L_AUWP_LIN	8	32	Remove
ST_PRM_END_AUWP_LIN	2	40	Remove
ST_RPM_MONI_AUWP_LIN	2	42	Remove
ST_EMMOD_IN_AUWP_LIN	2	44	Remove
ERR_ST_OTMP_AUWP_LIN	2	46	Remove
ERR_ST_OCU_AUWP_LIN	2	48	Remove
ERR_ST_DRYR_AUWP_LIN	2	50	Remove

**Frame 2: CONTROL\_AUWP**

- ID: 26
- Size: 2 Bytes
- Publisher: PDM28
- Signal: TARVL\_RPM
- Send Count: 0
- Configuration: Collapse
- Enable:

Signal	Bits	Offset	Action
TARVL_RPM_AUWP_LIN	8	0	Remove
FS_PRM_AUWP_LIN	3	12	Remove

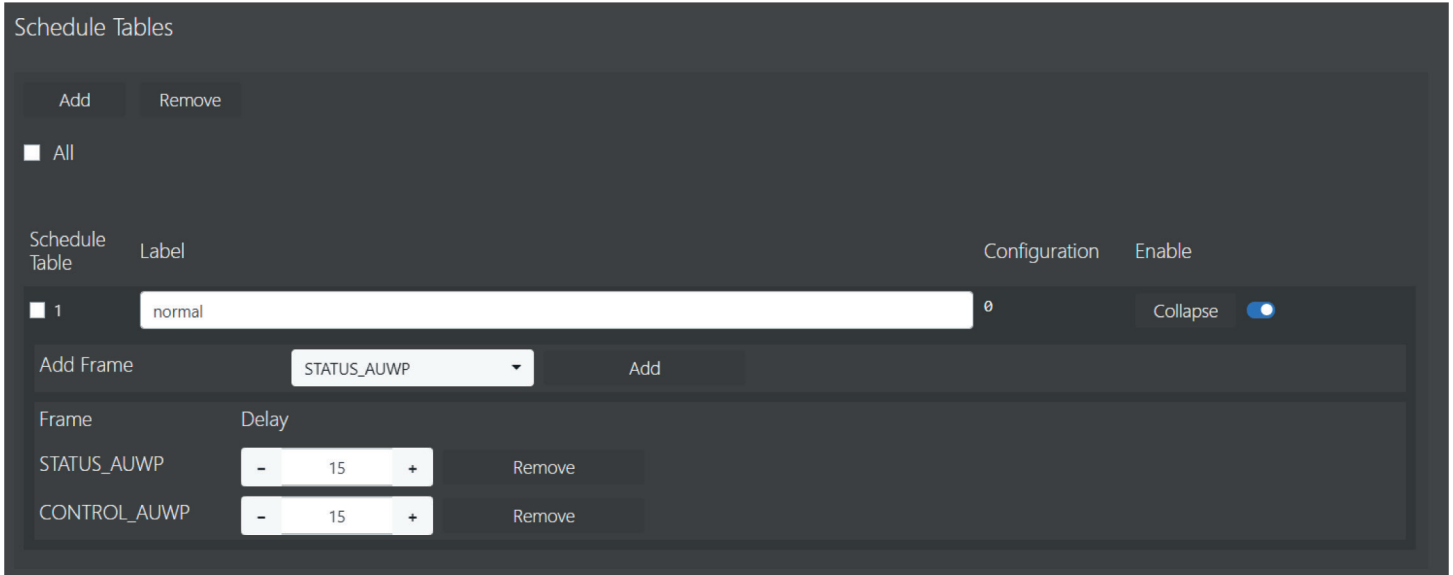
The interface also includes a byte-level view of the payload for each frame, showing the bit distribution across 8 bytes. For Frame 1, the payload is 8 bytes long, with each byte containing 8 bits. For Frame 2, the payload is 2 bytes long, with the first byte containing 8 bits and the second byte containing 3 bits.



# LIN Bus

## Schedule Table

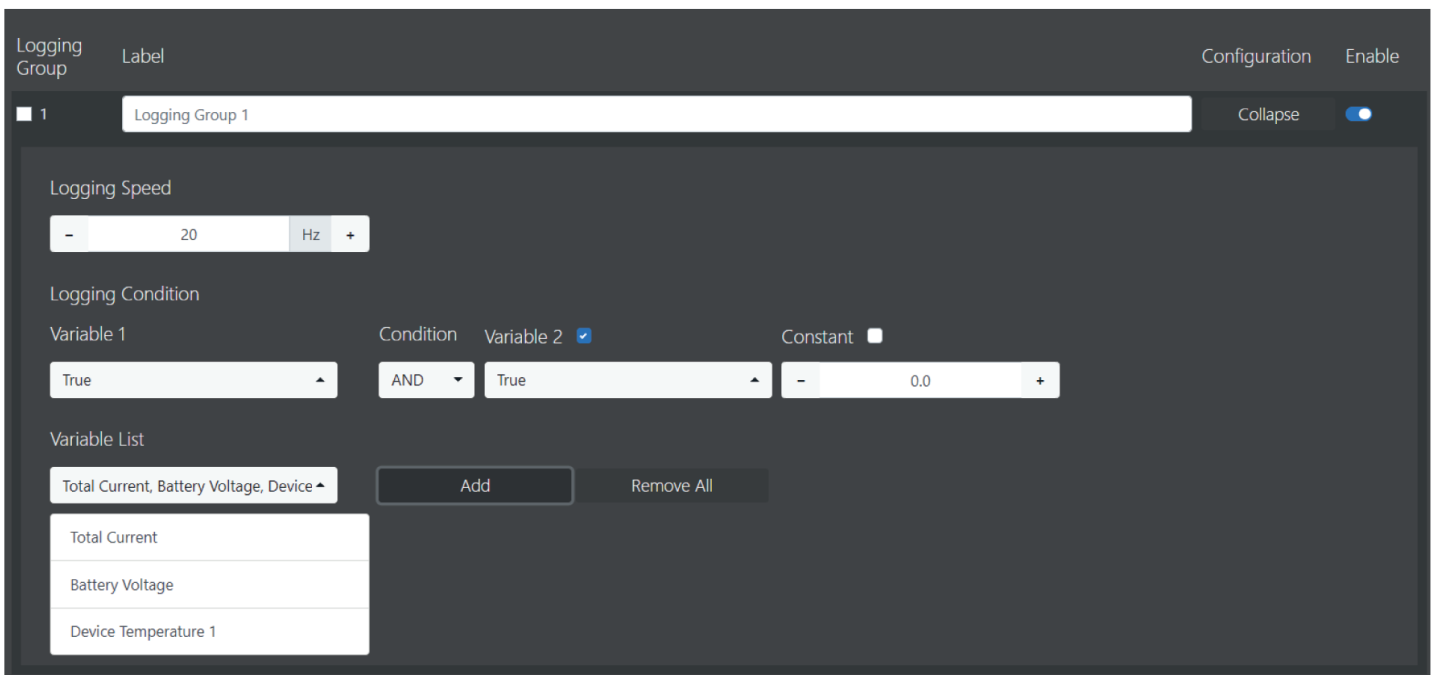
In this example with only two frames sent between the PDM and pump, the schedule table is simple. The schedule table dictates the timing in which each frame is sent. The schedule table runs constantly, in this case with a 15ms delay between each message.



# Data Logging

The PDM28 has 128Mb of on-board flash storage for logging data. This logged data can be downloaded and viewed on the PC to diagnose issues, and ensure that each electrical component in the vehicle is functioning correctly.

To enable data logging on the PDM28, navigate to the Configuration tab and then to the Data logging tab. A Logging Group can be added by clicking the 'Add' Button. A Label can then be given to that Logging Group. Click 'Expand' to further configure the Logging Group.



# Data Logging

Each Logging Group is responsible for data logging a collection of variables. These variables are logged at the Logging Speed, which can be changed between 1-50Hz. The variables are logged whenever the Logging Condition evaluates to True. In this example, the Logging Condition is set to 'True AND True', which will mean that the data is continually logged. Variables to log can be selected below.

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 PCB temperature should be logged at low speeds. Each time the PDM28 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.

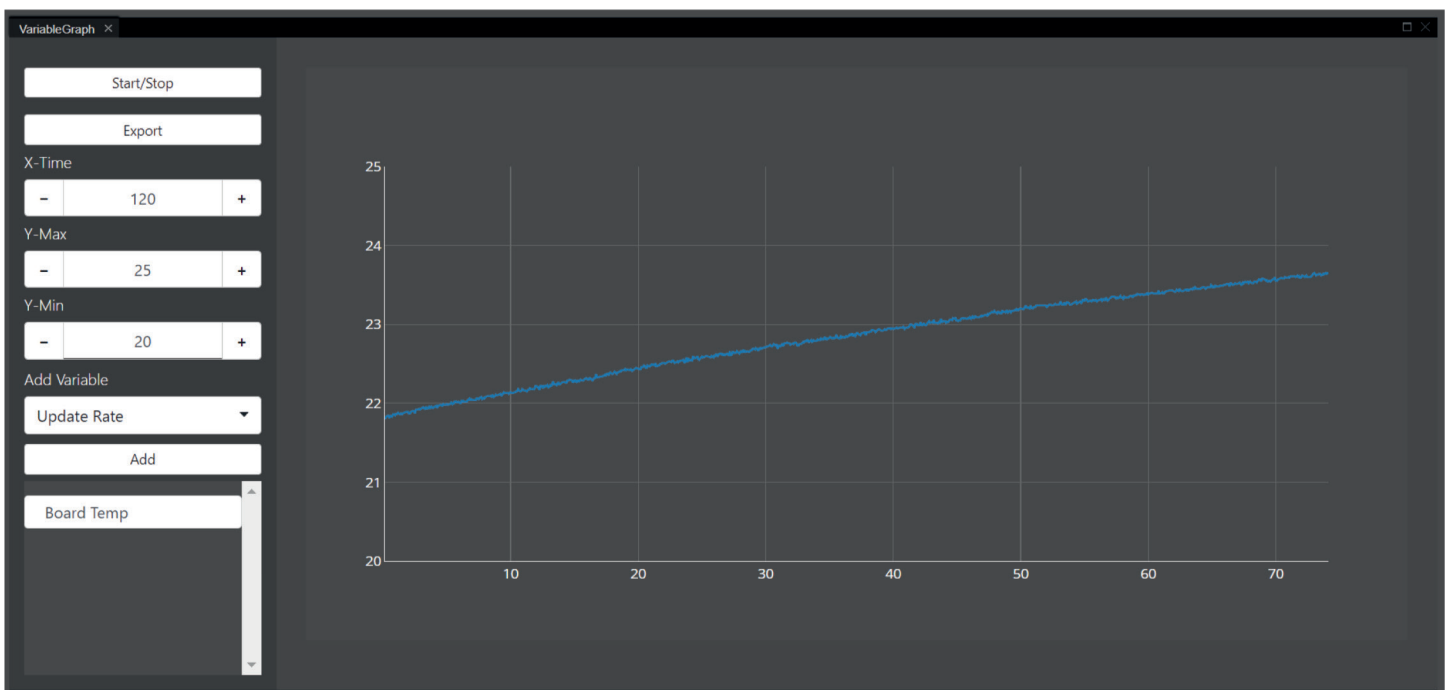
# Monitoring

One of the distinct advantages of using a Solid State Power Distribution System 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 Configurator software allows one to monitor the value of each variable in the system, such as the Power Module Input Voltages, Output Currents and more.

## Graph View

One of the easiest ways to visualise the different variables is with a Live Graph. To add a graph, navigate to the Monitor Tab and to Graph View and click 'Add'. This will insert a graph window. To add a variable to the 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.



# Monitoring

Live data gathering from the PDM28 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 to export the graph as a .csv file for viewing on external software. The graph windows can be resized, docked, and tabbed by clicking and dragging on the top of the graphing window.

## Variable View

To view the numerical value of different PDM28 and NIO Power Module variables, navigate to the Variable View Tab.

To make it easier to look at multiple collections of variables, one can add a variable 'Group'. Each group can be given a name.

Click expand, and then add variables to the group in the drop down menu.

Variable Groups can be saved to a file to make it easier to get the collection of variables you want to look at as quickly as possible. Each individual variable can be given a different colour as the user wishes.

The screenshot displays the 'Variable View' interface. At the top, there is a 'Group' section with a label 'Variable Group 1' and an 'Expand' button. Below this is a 'Variable List' section with a dropdown menu showing 'Acceleration Total, Acceleration X Axis' and buttons for 'Add' and 'Remove All'. The main area is a table of variables with columns for 'Variables', 'Value', and 'Unit'. Each row includes a color-coded square next to the variable name and a corresponding progress bar or gauge.

Variables	Value	Unit
Acceleration Total	0.328	
Acceleration X Axis	-0.266	ms <sup>-2</sup>
Acceleration Y Axis	0.358	ms <sup>-2</sup>
Acceleration Z Axis	0.237	ms <sup>-2</sup>
Rotation Angle X Axis	-7.936	deg
Rotation Angle Y Axis	14.25	deg
Rotation Angle Z Axis	336.937	deg
Rotation Speed X Axis	97.686	deg/s
Rotation Speed Y Axis	184.936	deg/s
Rotation Speed Z Axis	-159.436	deg/s

# Monitoring

## CAN Bus Analyser

Being able to easily see what CAN bus messages are on the CAN bus is important when configuring a Power Distribution System. This can be difficult to do if CAN filters need to be set up before any CAN messages can be received. Hardwire Electronics have solved this problem by introducing a CAN Bus analyser feature into the software. To start the CAN Bus Analyser, navigate to the monitor tab and to CAN Bus analyser, then click start.

The CAN Bus Analyser removes all CAN Bus Filters on the PDM28 CAN Bus ports, allowing all CAN messages to be received. In the image below, the CAN Bus Analyser is running in Continuous View mode. In Continuous View Mode, the CAN messages are received and shown in order. As more CAN bus messages are received, the display will scroll upwards to show the latest message.

**CAN Bus Analyser**

Stop Fixed View Start Log Save Log

**CAN Monitor**

Bus	Time	Msg	ID Type	ID	D0	D1	D2	D3	D4	D5	D6	D7	Count	Period
1	98.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0	0.000
1	99.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	1	1.000
1	100.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	2	1.000
1	101.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	3	1.000
1	102.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	4	1.000
1	103.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	5	1.000
1	104.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	6	1.000
1	105.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	7	1.000
1	106.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	8	1.000
1	107.341	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	9	1.000

**CAN Transmit**

Bus ID Type ID DLC D0 D1 D2 D3 D4 D5 D6 D7

1 Std - 0 0x00 + - 0 + 00 00 00 00 00 00 00 00 Send

The image below shows the CAN Bus Analyser in Fixed View Mode. In Fixed View Mode, only the latest CAN Bus message with each unique ID is displayed.

CAN Bus messages can be sent onto the CAN Bus using the CAN Transmit options at the bottom. The user can select the appropriate CAN Bus to send on, the ID Type and Value, the data length and finally the value of each data byte.

**CAN Bus Analyser**

Stop Continuous View Start Log Save Log

**CAN Monitor**

Bus	Time	Msg	ID Type	ID	D0	D1	D2	D3	D4	D5	D6	D7	Count	Period
1	328.231	Rx	Std	0x123	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	0xFF	197	1.000
1	328.781	Rx	Std	0x10	0x12	0x34	0x56	0x78	0x9A				2710	0.050
1	325.445	Tx	Std	0xE	0x23	0x45							5	0.409

**CAN Transmit**

Bus ID Type ID DLC D0 D1 D2 D3 D4 D5 D6 D7

1 Std - 14 0x0E + - 2 + 23 45 00 00 00 00 00 00 Send

# Monitoring

When the CAN Bus Analyser is running, the data can be logged in real time to a CSV file. This data can then be viewed in an external program at a later date. Press Start Log to start a data log. Press Stop Log to stop logging CAN data. Once a data log has been stopped, the user can save the Log to the PC.

# Logging

The Hardwire Electronics Configurator software can download data log files from a PDM28 and display the data in a graphical format. To retrieve data logs, connect to the PDM28 and navigate to the data logging tab. Click on the button 'Retrieve Log Info' to retrieve a list of the data logs stored on the PDM28. 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 PDM28.



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 - i.e. 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

Software updates allow the user to take advantage of the newest features available for both the PDM28 and NIO Power Module. To update the firmware on either device, make sure that the PDM28 is connected to the PC, and that the Power Module is communicating with the PDM28.

## PDM28 Firmware Update

To update the PDM28 firmware, click the 'Select File' button. Click on the appropriate .bin file PDM28 Firmware.

Click the 'Update' Button to commence the update process. The green bar will start to move to indicate that the update is taking place. Once the update is complete, the PDM28 will restart and attempt to reconnect to the PC to resume normal operation.

## NIO Power Module

To update the Power Module firmware, make sure that the Power Module is turned on and connected to the PDM28. Select the appropriate Power Module from the drop down list. Click the 'Select File' button. Click on the appropriate .bin file Power Module Firmware.

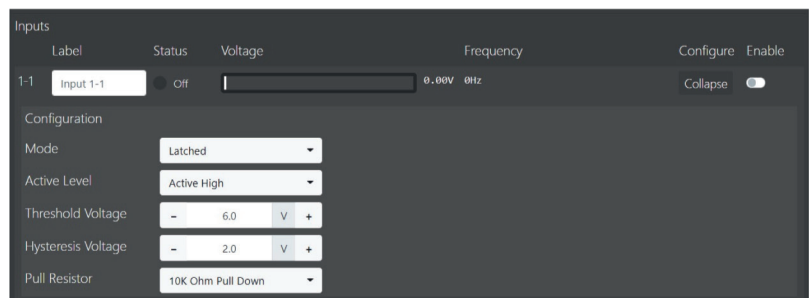
Click the 'Update' Button to commence the update process. The green bar will start to move to indicate that the update is taking place. Once the Update has taken place, the Power Module will restart and begin to operate normally.

# Examples

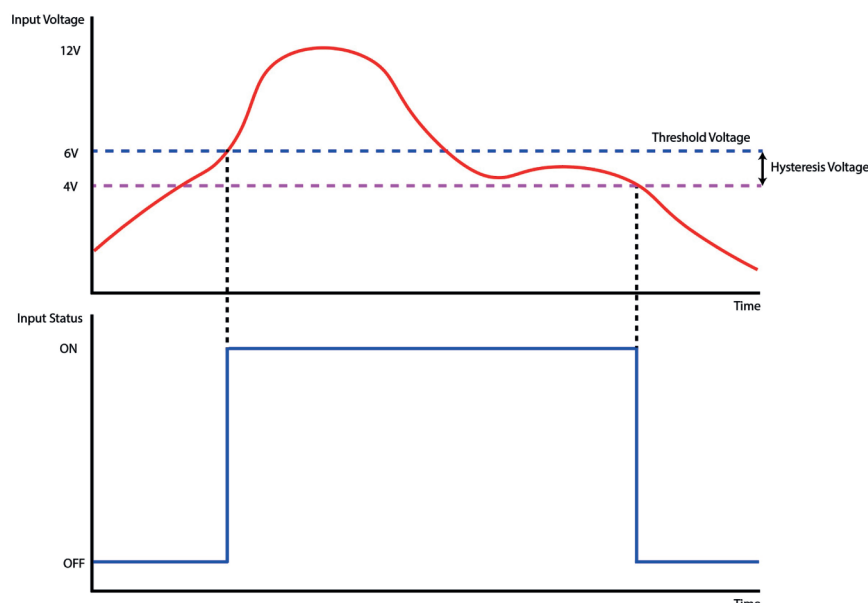
This section aims to provide a number of examples of how the Node Power Distribution System can be set up. The examples assume that you have a PDM28 and Power Module connected and working properly.

## 1. Basic Input Setup

This example shows a basic setup for a NIO Power Module Input, so that the input triggers when 12V is applied. This configuration does not use a Pull-down resistor, but if you are using a push button switch to trigger the input, then it is recommended to select the pull down resistor from the menu.



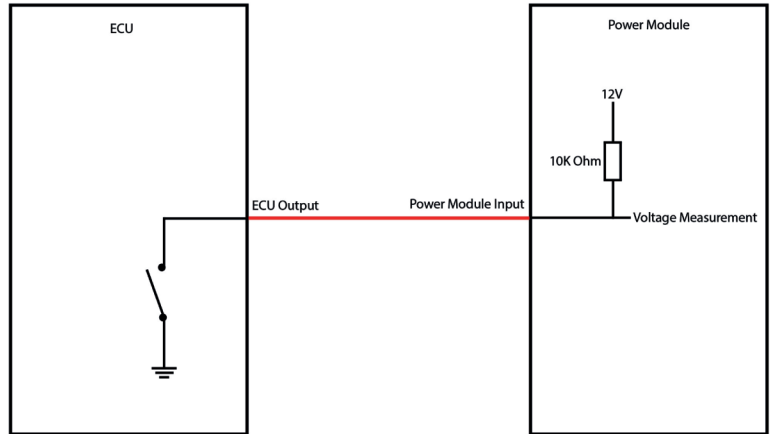
From the graph below, one can see that the input triggers when the input voltage rises above 6V. With a Hysteresis voltage of 2V, the input does not turn off until the voltage drops below 4V (6V-2V). The Hysteresis voltage can help to eliminate false triggering of the input if the signal is noisy.



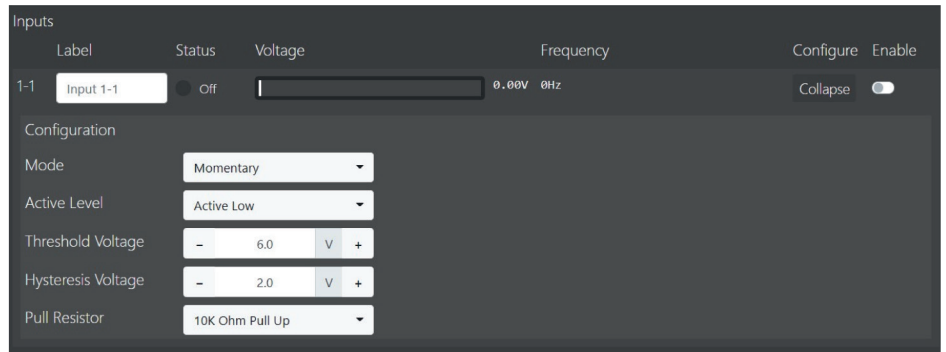
# Examples

## 2. ECU Triggered Input

This example demonstrates the typical procedure to have an ECU output trigger a Power Module input. Most ECU outputs are low-sided, meaning that they switch ground onto the output. When the output is not triggered, this leaves the output 'floating,' meaning that the voltage on the output is undefined. If this is tied to a Power Module input, the voltage measurement in this case can be sporadic. To combat this problem, it is important to enable the Pull-up resistor on the Power Module input. This means that when the ECU output is not triggered, the voltage on the output is 'pulled' up to 12V, or the voltage powering the Power Module. Some ECUs have a Pull-up resistor internally on the output, but many do not. It is therefore recommended to always use the Power Module Pull-up resistor when interfacing with Low Sided ECU Outputs.

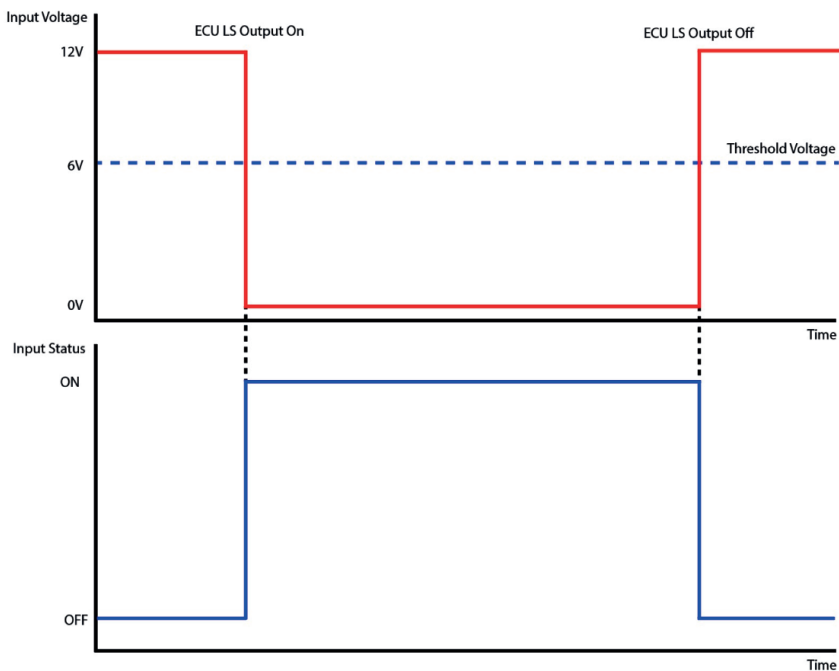


The Power Module Input is configured to be active low, so that when the ECU output is driven to ground, the input on the Power Module triggers. The Threshold Voltage is set to 6V, this is half of the vehicle battery voltage. The Hysteresis voltage is set to 2V, but in most cases this will not be necessary with ECU outputs as they are hard on/off. Finally, the Pull-up resistor is selected as discussed before.



If a High-sided ECU output is used, then everything can be kept the same except from the Active Level and Pull resistor. They should be set to Active High and Pull-Down in this case.

**Below is the voltage on the input as the ECU Low Side output triggers.**



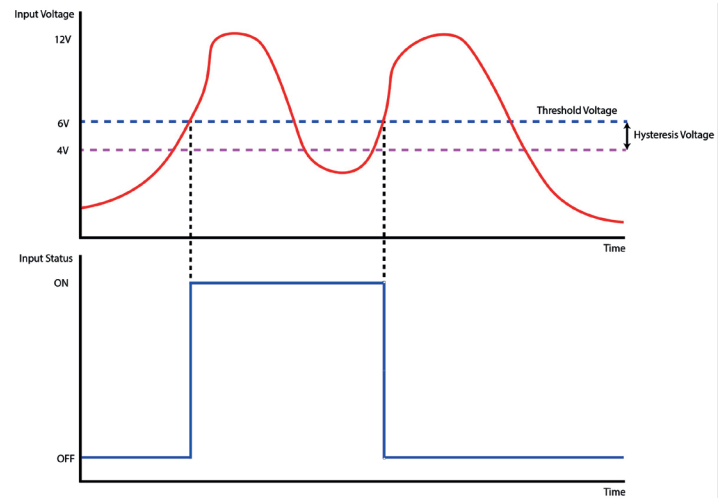
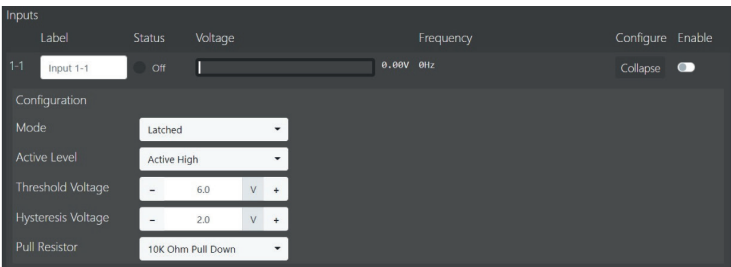


# Examples

## 3. Latching Input Setup

This example shows a set up for an input configured in Latching Mode. In Latching Mode, the input status toggles everytime the input is triggered. In this case, the threshold voltage is set to 6V and the input is set to active high, so each time the voltage rises above 6V, the input toggles state.

This setup is useful for latching an input on when zpressing a momentary button switch.



## 4. Basic Output Setup

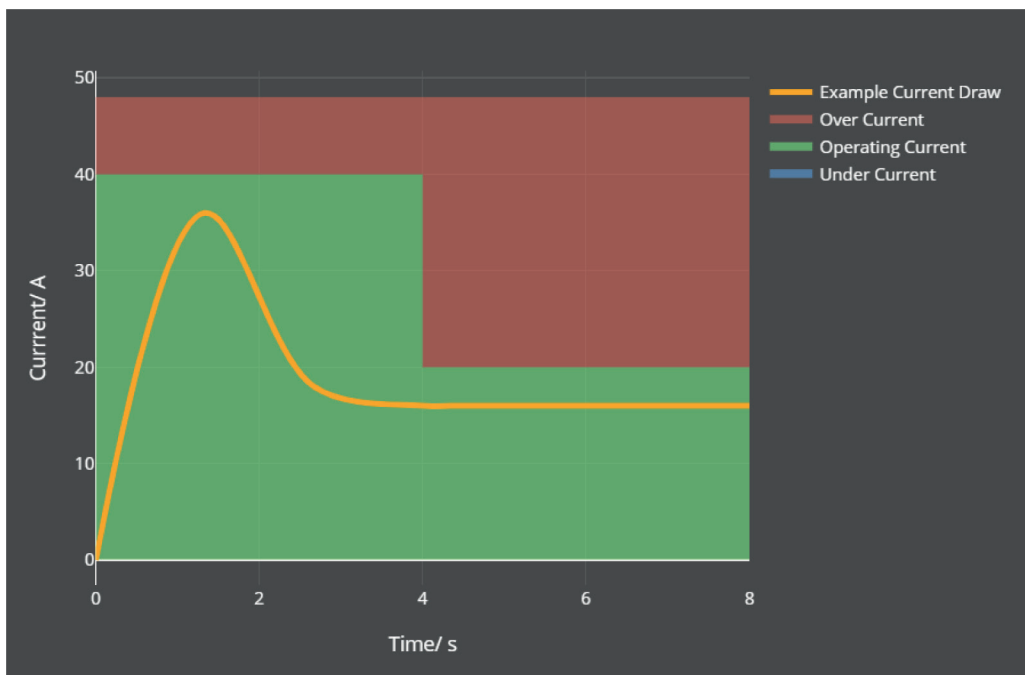
This example will show how to set up a simple output that turns on when a button on a CAN bus Keypad is pressed. This method is also applicable for switching an output when an input is triggered.

This example is based on the configuration of a typical Radiator Fan with the following parameters.

<b>Nominal Current Draw</b>	<b>17A</b>
<b>Start Up Current Peak</b>	<b>36A</b>

A Radiator Fan is a mechanical object which has inertia to overcome. On start-up, the fan motor has to accelerate and overcome this inertia to reach full speed. This acceleration of the fan motor results in a high start-up current, which then settles down to normal operating current once the motor is at full speed. An example of the current draw on start-up is shown below.

The current draw of the motor quickly rises to a peak of 36A, before gradually falling to 17A.



# Examples

The configuration settings of the PDM Output are shown in the adjacent image. The High Fuse, Peak Fuse, and Peak Fuse Time have been carefully selected based on the current draw characteristics of the Fan.

The Peak Fuse is set to 40A which is 4A above the peak current draw of the Fan to allow some room for error. The High Fuse has been set to 20A which is 3A above the nominal current draw of the Fan. The Peak Fuse Time has been set to 4 seconds.

This is to allow time for the current draw to come down to below the High Fuse threshold. When setting up an output, it may take some time to tune these values in. Effects such as temperature and component age/condition can all affect the current draw. It is therefore necessary to add approximately 10-20% margin on the current trip limits, as seen in this example.

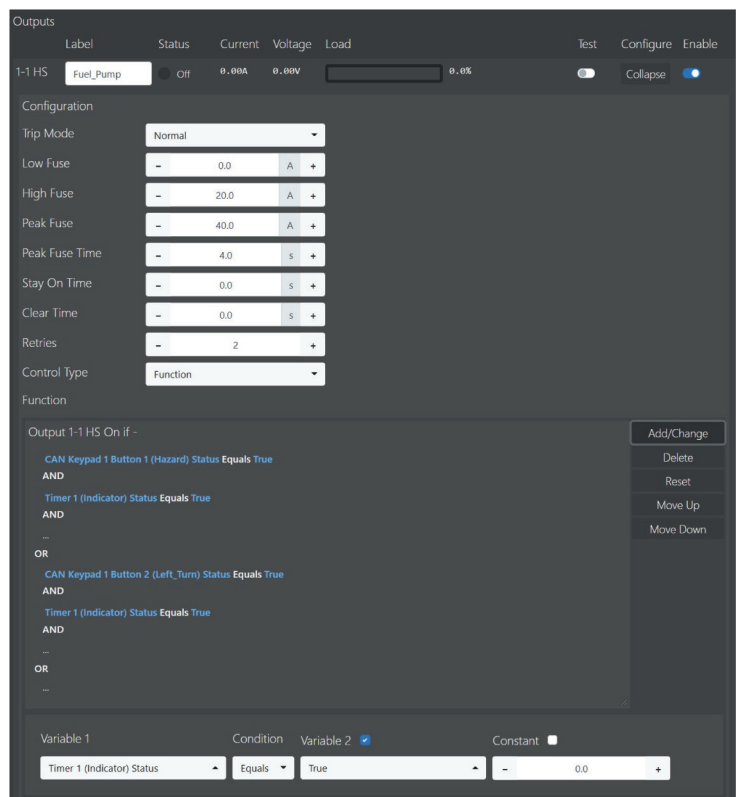
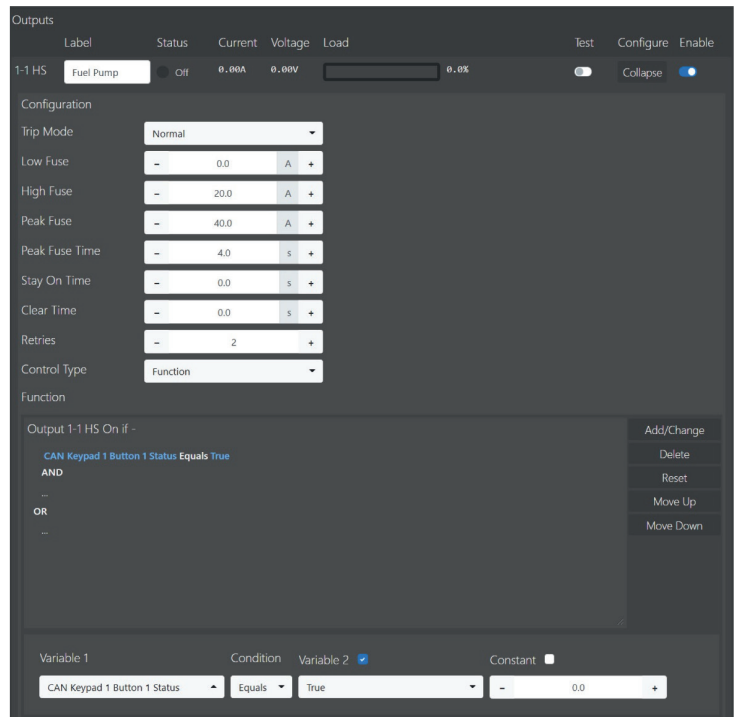
## Indicator/Turn Signal Setup

To set up left and right Indicators with Hazard light functionality, first add a timer to the configuration and set the On and Off time to 400ms.

Configure the CAN Bus Keypad or Inputs so that the Indicator Button and Hazard Button both latch on when the button is pressed.

The Output for the Indicator can then be configured as shown here -

Timer 1 simply alternates between the On and Off state every 400ms. The status of the Timer can be AND'd with the status of the Indicator button. When both the Timer and Button status is true, the output will be on, and when either or both are false, the output will be off. The same can be repeated for the other indicator output.



# Examples

## PWM Fan Control Setup

In this example, an output has been configured to control a radiator fan using PWM. For most applications, PWM is not required for radiator fans, but for users that need finer control, this is a good method to achieve that. The first thing to note is that the output function is set so that the output will turn on if either CAN Keypad 1 Button 1 Status is equal to true, or if the coolant temperature (received via CAN) is above 80 degrees C. If the output function does not evaluate to True, then the output will not turn on, no matter the PWM mapping.

The next part of the setup is for the PWM of the output. The PWM adjusts the duty cycle of the fan output based on the coolant temperature. The higher the coolant temperature, the higher the duty cycle, and hence speed, of the fan. To increase the duty cycle of the output with coolant temperature, the PWM mapping variable is set to be the CAN Input Coolant Temperature Value.

The screenshot displays the configuration for a PWM output. At the top, the output is labeled '1-1 HS' and 'Output 1-1 HS', with a status of 'Off', current of '0.00A', voltage of '12.93V', and a duty cycle of '0.0%'. The configuration section includes Trip Mode (Normal), Low Fuse (0.0 A), High Fuse (20.0 A), Peak Fuse (20.0 A), Peak Fuse Time (5.0 s), Stay On Time (0.0 s), Clear Time (0.0 s), Retries (0), and Control Type (Function). The function is 'Output 1-1 HS On if -' with conditions: 'CAN Keypad 1 Button 1 Status Equals True' AND 'CAN Input 1 (CLT) Value > 80'. The PWM settings are: Frequency (500), Soft Start Time (100 ms), Mapping (CAN Input 1 (CLT) Value), and a Duty Cycle table. The graph shows the relationship between Variable Value and Duty Cycle %.

Duty Cycle	Variable Value
0%	70.00
10%	74.00
20%	78.00
30%	81.00
40%	83.50
50%	85.50
60%	87.00
70%	88.00
80%	89.00
90%	89.50
100%	90.00













HW  
POWER  
DISTRIBUTION  
MODULE  
DIAGNOSTIC  
SOLID STATE

Hardwire HW

control  
MotorSport  
TECH  
PROGRAMMING  
SWITCH Panel  
FUSE  
MODULE  
BACK CAR  
PDM  
MotorSport  
MANANCE TECHNOLOGY  
current  
FUSE control  
component  
WEB  
MODULE  
PROTECTION  
VOLTAGE  
HW  
f HW  
Instagram  
in  
YouTube  
PDM  
MotorSport  
HW