ECUMASTER PMU-16/PMU-16DL
Preliminary Manual

(2022.09.05, rev. 1.04)

ATTENTION!

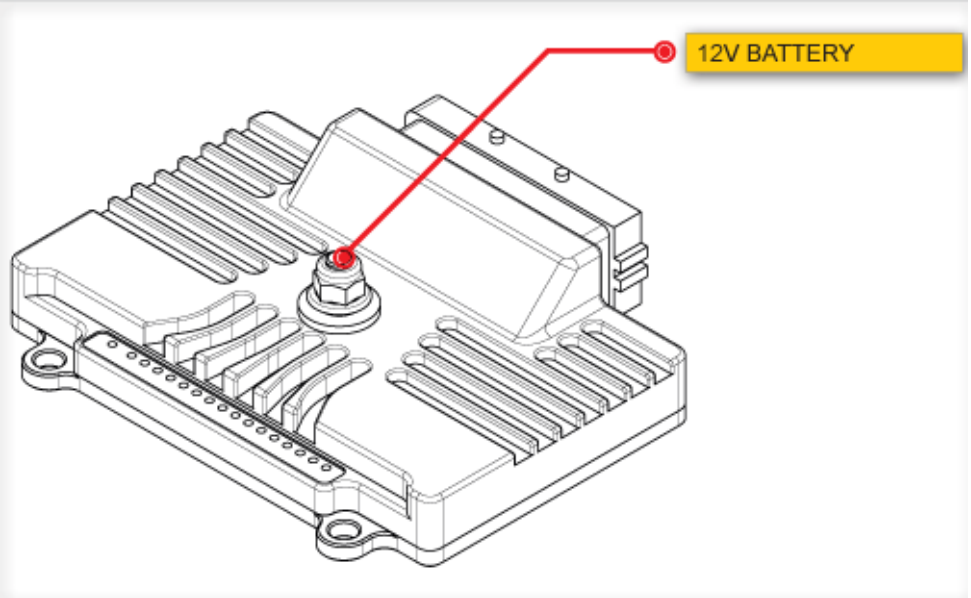
- **The ECUMASTER PMU is designed for motorsport applications only and cannot be used on public roads!**
- **The installation of this device should be performed only by trained specialists. Installation by untrained individuals may cause damage to both the device and the vehicle!**
- **Incorrect configuration of the ECUMASTER PMU can cause serious damage to vehicle components!**
- **Never modify the device's settings while the vehicle is moving as it may cause an accident!**
- **ECUMASTER assumes no responsibility for damage caused by incorrect installation and/or configuration of the device!**
- **To ensure proper use of ECUMASTER PMU and to prevent risk of damage to your vehicle, you must read these instructions and understand them thoroughly before attempting to install this unit.**
- **Never short-circuit the wires of the vehicle's wiring loom or the outputs of the ECUMASTER PMU!**
- **All modifications to the vehicle's wiring loom must be performed with the negative terminal of the battery disconnected.**
- **It is critical that all connections in the wiring loom are properly insulated!**
- **The device must be disconnected before performing any welding on the vehicle!**

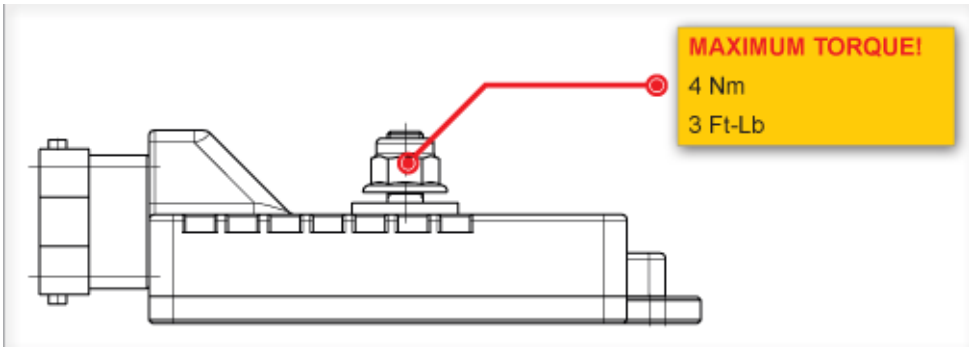
Table Of Contents

ECUMASTER PMU

ECUMASTER PMU is an inteligent power management unit designed to replace the old, traditional and often unreliable fuses and relays. PMU is not only an electronic switch, but a device that can perform all kinds of advanced operations, validate logical conditions, manage abnormal situations, as well as log its parameters for you to have an oversight. PMU allows you to power up external devices such as fan, blinkers, wipers, oil pump etc. and create advanced strategies for those devices using logic with failsafes, condition checking and many more functions. It can communicate and work in tandem with other ECUMASTER CAN devices. It is equipped with over and under current protection, surge protection, 3D gyroscope, accelerometer, LED Status lights, Soft Start, Pulse Width Modulation with Duty Cycle control and more.

Technical





Specification

GENERAL	
Temperature Range	ACEQ100 GRADE1 (-40 to 120C)
CPU	32 bits automotive, 90MIPS
Reverse polarity protection	Yes, internal up to 16V
Operating voltage	6-22V immunity to transients according to ISO 7637
Enclosure	IP65, bespoke CNC machined aluminium
Size and weight	131x112x32.5mm, 345g
Connectors	1 x 39 Automotive connector 1 x M6 stud for battery connection
PC communication	CAN (Peak CAN, ECUMASTER USBCAN, Kvaser)
Multiple PMUs	Up to 5 PMUs can work in tandem
OUTPUTS	
10 x high side up to 25A output	Overcurrent and overheating protection. Outputs may be paired to increase continuous current capability. Current and voltages measured for each output. Inductive load clamp: Vbat - 36V Inductive energy dissipation Ear: 460mJ Peak current: 120A
6 x high side up to 15A output	Overcurrent and overheating protection. Outputs may be paired to increase continuous current capability. Current and voltages measured for each output Indoctive load clamp with diode to: -0.5V Peak current: 120A
Total current output	150A continuous
Output current control step	100mA
PWM	Yes, available for each 25A output Programmable variable Duty Cycle control for each output Separate frequency setting ranging from 4Hz to 400Hz for each output
Soft Start	Yes, available for each 25A output
Low side wipers braking output	Dedicated output with wiper braking feature. Pin shared with output #8 Current limit: 6A
+5V	monitored 5V, 500mA output for powering external sensors

INPUTS	
Analog Inputs	16 inputs, 10 bit resolution, 0-5V (protected), with software selectable 10K Ohm pullup and pulldowns
CAN Keypads	2 x Ecumaster keypads (4, 6, 8, 12 keys), LifeRacing PDU Keypad
OTHER	
Output state indication	16 bicolor LEDs
Accelerometer/Gyroscope	3D accelerometer with 3D gyroscope for logging and crash detection
Real Time Clock	Yes, super capacitor for backup power (up to 3 days)
CAN BUS	
CAN interface	2 x CAN2.0 A/B
CAN standard	2.0A/B 125, 250, 500, 1000 Kbps
Input/Output Stream	User defined with bit masking Up to 48 input messages
LOGGING (PMU16DL only)	
Logging Memory	256 Mbytes
Logging Speed	Variable, defined per channel, up to 500Hz
PC LOGGING	
Logging Speed	Variable, defined per channel, up to 500Hz
FUNCTIONS	
Logical Operations	isTrue, isFalse, =, !=, <, <=, >, >=, AND, OR, XOR, Flash, Pulse, Toggle, Set/Reset Latch
Number of functions	100
Number of operations	250
Update frequency	500Hz
Special functions	Wipers, Blinkers

SOFTWARE INSTALLATION

Compatibility

PMU Software is compatible with Windows XP/Vista/7/8/8.1/10.

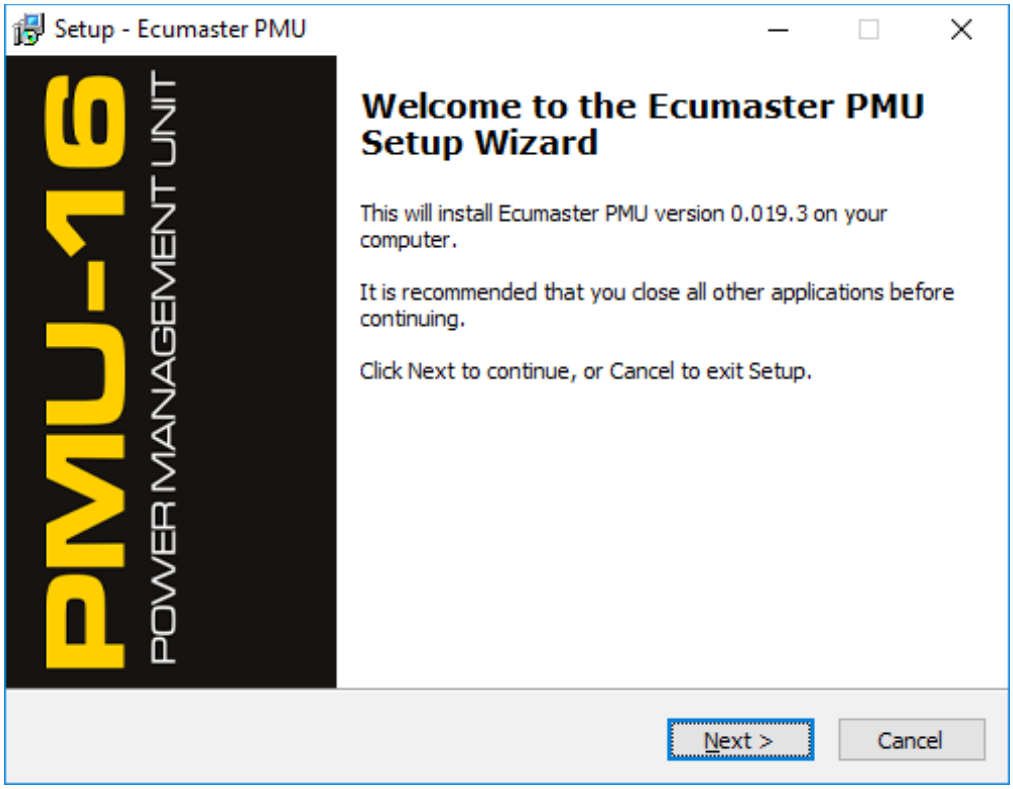
Downloading the PMU software

To configure PMU device a PMU Client must be used (both PMU16 and PMU16DL use the same client). To download PMU software please head to www.ecumaster.com/pmu page.

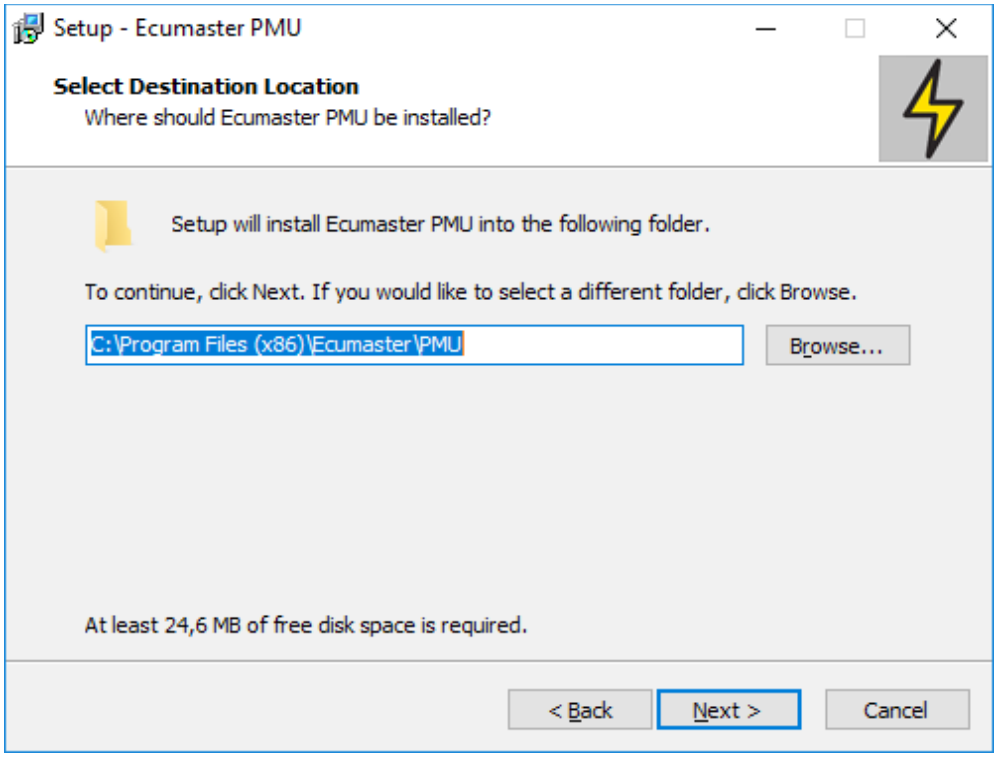
Installing PMU software

To install PMU software, double click the downloaded *"PMUSetup_X_XXX_X.exe".*

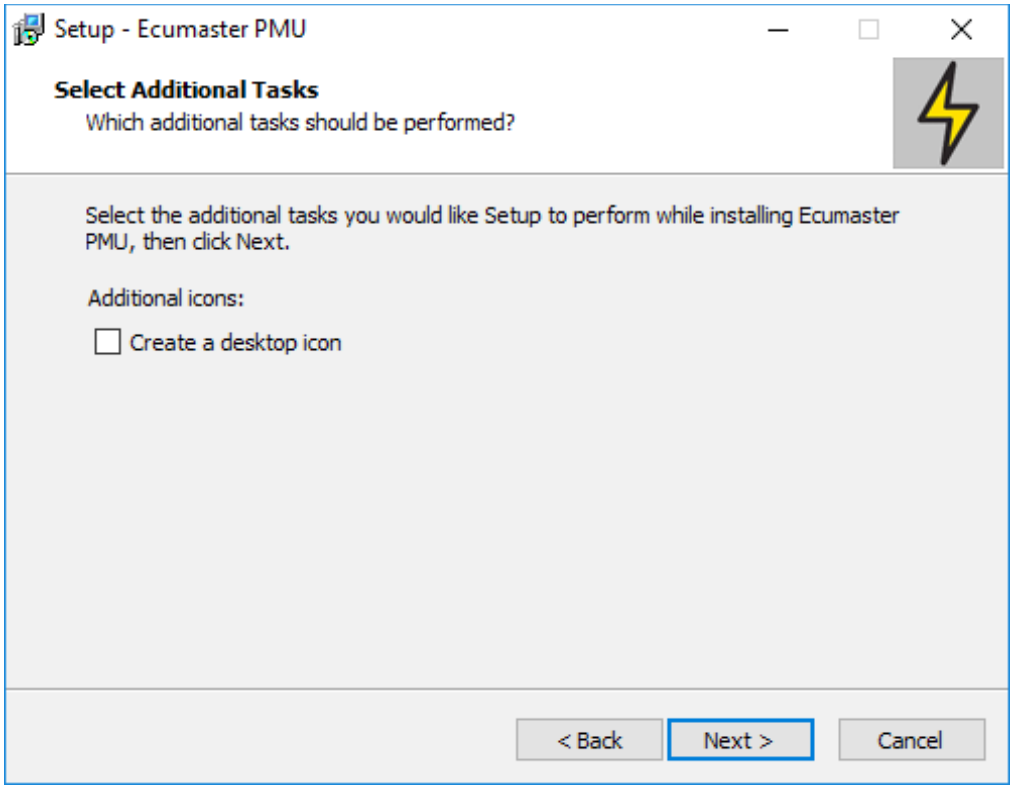
Click *Next* to proceed.



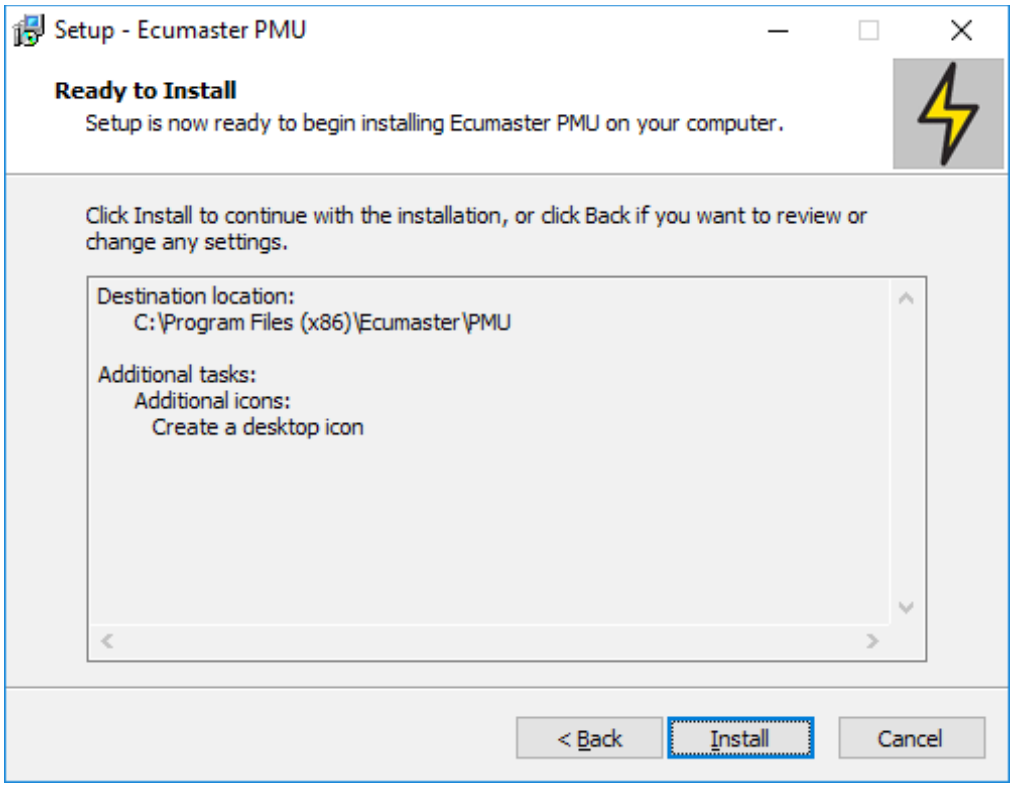
Choose which folder to install the PMU Client to.



Choose wheter you want a desktop icon or not.



This is the summary of your installation, if both the folder and icon choice are correct, press Install to proceed. If not, you can go back to make a quick correction.



After the installation is finished, you can choose to launch PMU Client right away.

CONNECTING PMU TO PC

CAN – USB interface

To properly connect PMU to PC using USB 2.0 a special interface must be used. PMU Client supports three interfaces:

- ECUMASTER USBtoCAN interface (can be bought directly from www.ecumaster.com)
- PEAK Systems PCAN-USB
- Kvaser USBcan

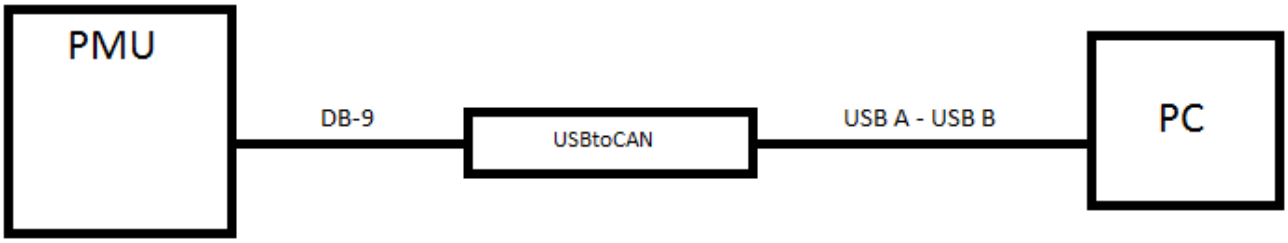
Each interface must have its drivers installed.
Interface must be bought separately.

ECUMASTER USBtoCAN interface

To use ECUMASTER USBtoCAN interface, user must be equipped in following items:

- USB A to USB B adapter to connect the interface to PC
- DB-9 Cable to connect interface to PMU
- ECUMASTER USBtoCAN interface drivers, available at: http://www.ecumaster.com/products/usb-to-can/

To install drivers run *EUSBtoCAN_Driver_v1.0.exe* and follow installation instructions.
Basic cable connection should look like this.

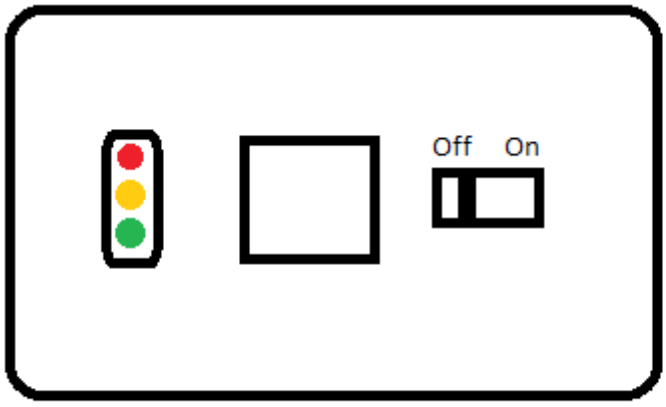


USBtoCAN connection

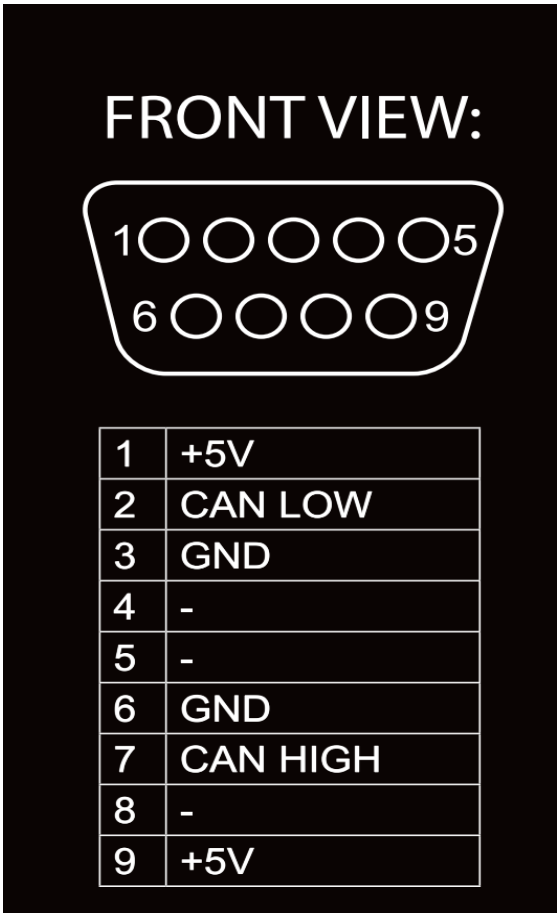
ECUMASTER USBtoCAN also has LED signaling ability:

Color	Description
Green Continuous	Device turned on
Green Flashing	Device turned on and connected to PC
Green and Orange Flashing	Data transfer in progress
Orange Continuous	Device turned on, currently in bootloader
Orange Flashing	Device turned on, firmware update in progress
Red Continuous	Temporary CAN communication error
Red Flashing	Permanent CAN communication error

ECUMASTER USBtoCAN is equipped with 120Ohm CAN terminator which can be switched on or off by user. Picture below shows terminator switch location:



ECUMASTER USBtoCAN pinout



1	+5V
2	CAN LOW
3	GND
4	-
5	-
6	GND
7	CAN HIGH
8	-
9	+5V

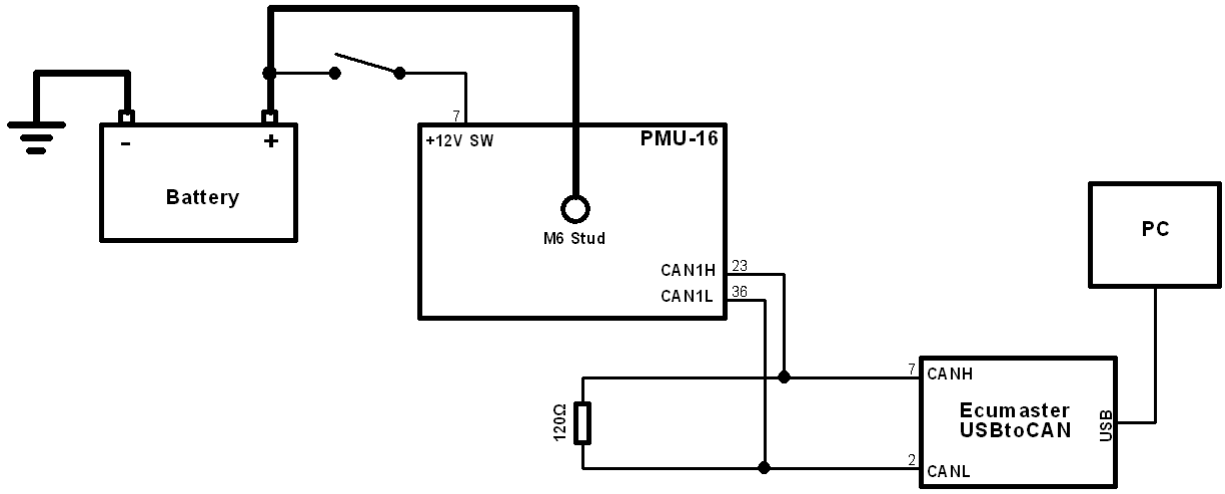
Connecting USB interface, wiring schematics

To connect PMU to PC, CAN1 CANbus must be used. This is the CAN provided for us for PC communication. To wire the USB - CAN interface, CAN1H and CAN1L pins (See *PMU Pinout* section) must be used. Twisted pairing is also recommended. CANbus must also be terminated at both ends of the bus. ECUMASTER USBtoCAN is equipped with one terminator that can be switched on or off (See *ECUMASTER USBtoCAN interface* section).
If the Interface is connected, PMU must be supplied with power, both on the +12SW Pin (See *PMU Pinout* section) and the M6 stud.

ATTENTION !

Ground wire connection between USBtoCAN and PMU-16 can be DANGEROUS!
Ground connection is only allowed if devices before connecting have the same ground potential.
Potential difference can be checked by using voltometer between corresponding grounds.

Here is a simple diagram of USB connection, please note that it assumes that the second terminator is applied by the USBtoCAN interface:



PMU - PC communication

PMU status

PMU device is fitted with LED that signals status of the PMU device.
Picture showing the LED location:

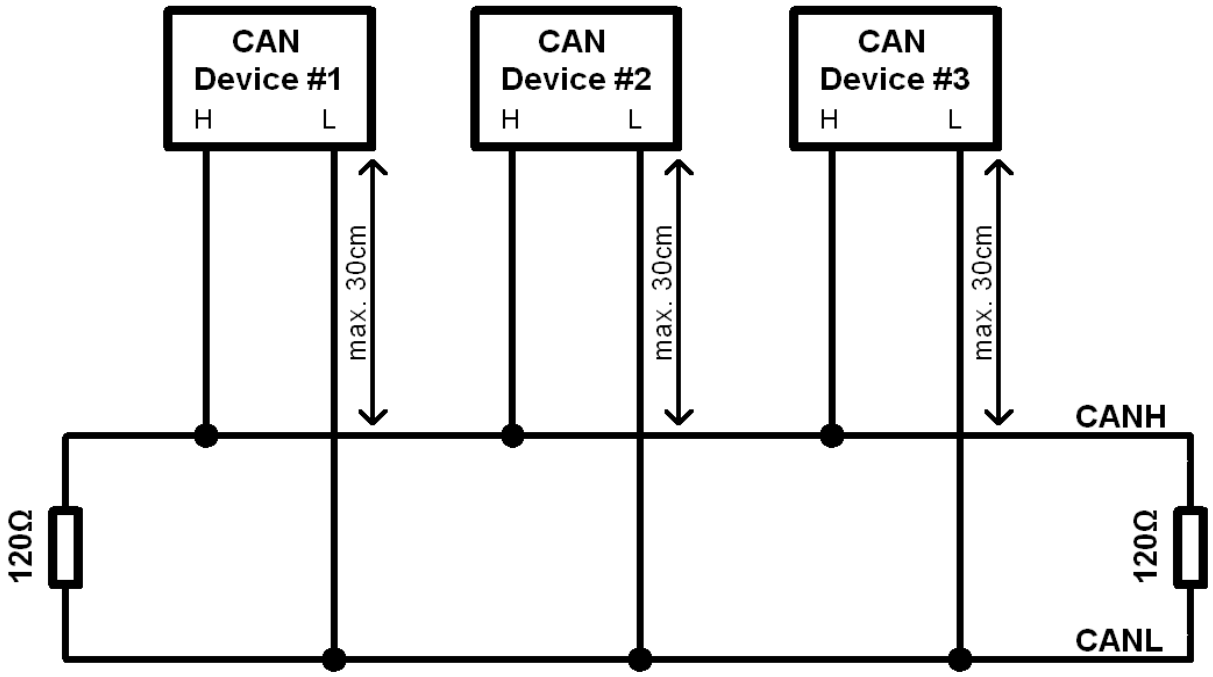
There are 6 possible states of PMU LED:

Color	Status
Green Continuous	Device is active
Orange Continuous	Device is active and connected to PC
Green Flashing Slowly	Device is waiting for Firmware Upgrade
Orange Flashing Slowly	Device is performing Firmware Upgrade and is connected to PMU

	Client
Orange Flashing Fast	Device is performing Make Permanent operation
Red Continuous	Device Error - please contact the distributor or manufacturer directly

CAN protocol, CAN topology

CAN topology looks like this:



CAN Topology

For 1Mbit/s connection (CAN 1), following rules must be abided:

- Maximum unterminated cable length from device to CANbus is 30 cm
- Maximum bus length is 40m.
- Maximum of 30 nodes
- 120 Ohm terminators must be applied at both ends of CAN bus.
- Twisted pairing is required.

For 500kbit/s connection (CAN 2) following rules must be abided:

- Maximum unterminated cable length from device to CANbus is 30 cm
- Maximum bus length is 100m.
- Maximum of 30 nodes
- 120 Ohm terminators must be applied at both ends of CAN bus. PMU is equipped with CAN2 terminator which can be turned on or off via PMU Client (*Tree View → CAN Setup)*
- Twisted pairing is required.

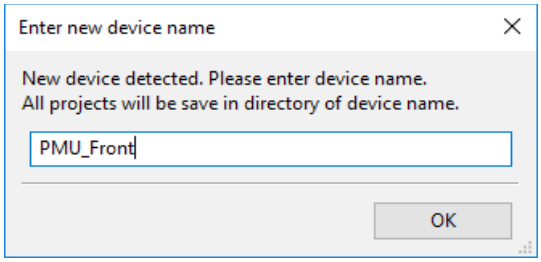
USING PMU SOFTWARE

Launching PMU software

To run PMU software either doubleclick the icon on desktop, or use start menu to find it.

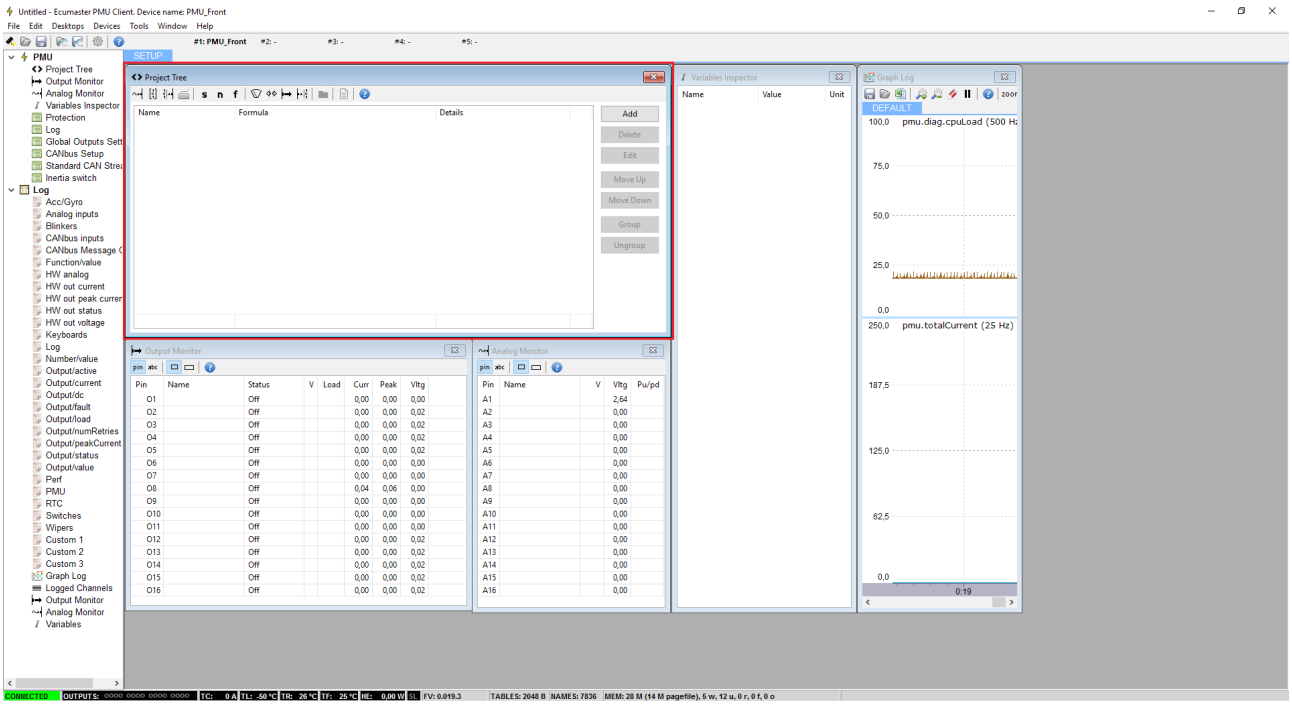
Using PMU software

When PMU Client is launched for the first time, user will be asked to enter the name of his device. All projects will then be saved to the directory corresponding to devices name.



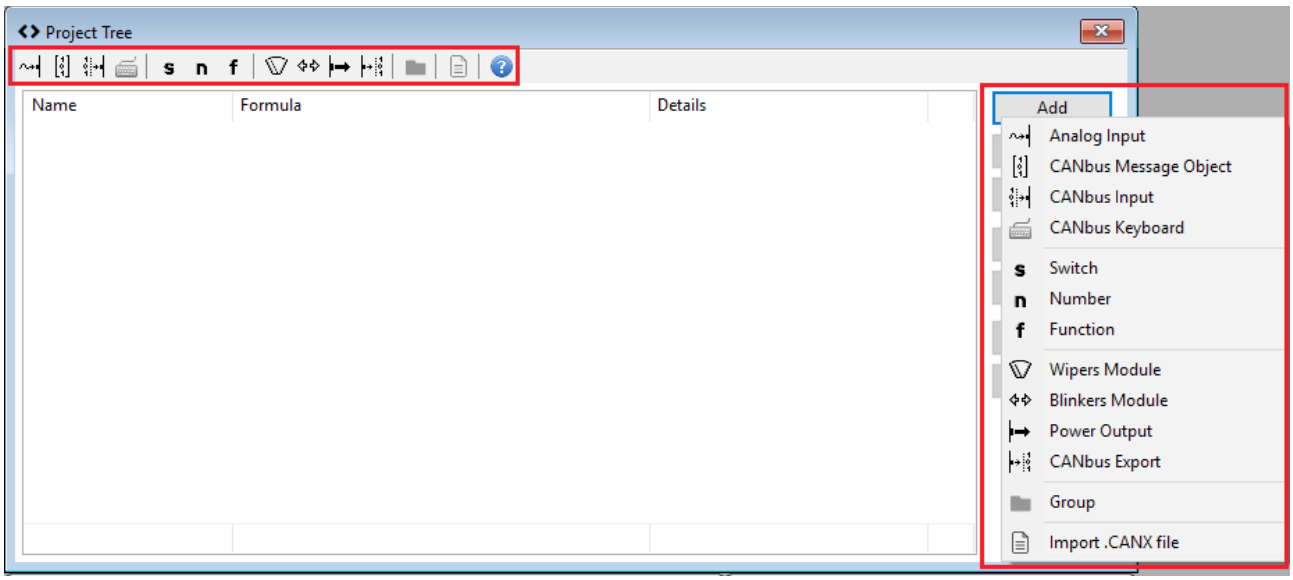
New device name

After entering the Client, main window will appear:



Default Dekstop/Main Window with highlited Project Tree window

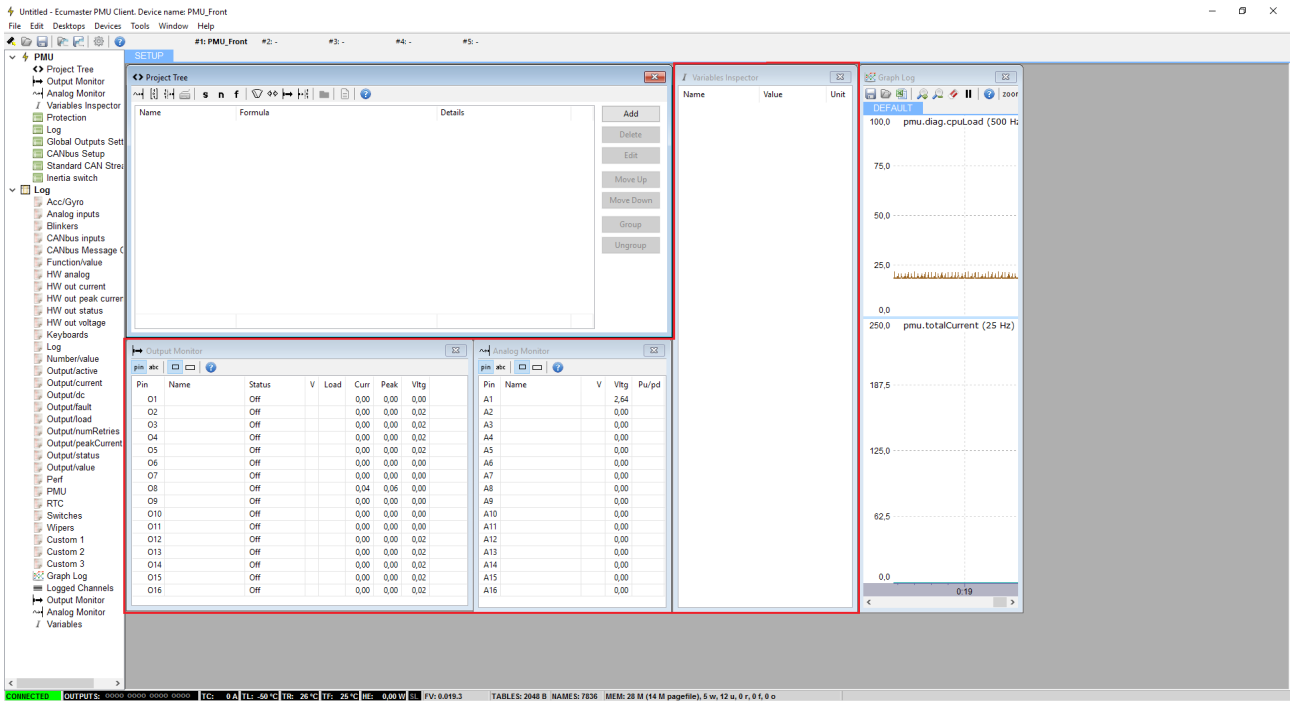
Using the Client is pretty straightforward. The *Project Tree* window is the most significant one. You can use it to set up analog inputs, configure power outputs, create functions etc.
To create elements, either use the toolbar with icons located on *Project Tree* window, use *Alt + A* Keyboard shortcut or click *Add* button, then select the type of element you want to create.



Project Tree, adding new Elements

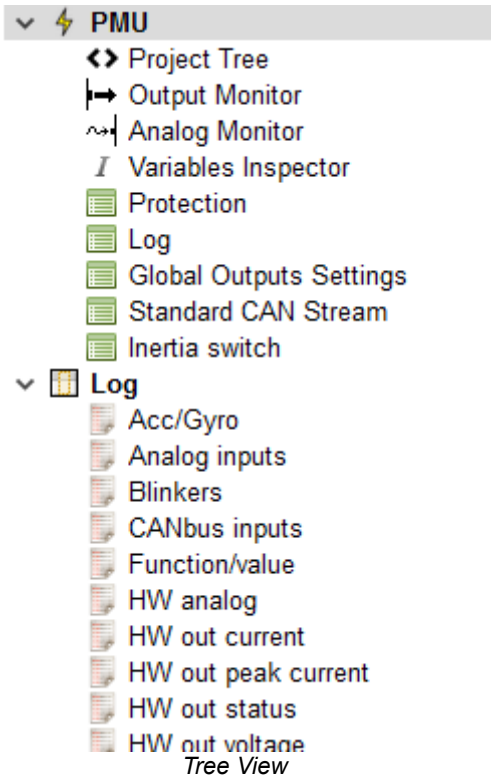
To upload current project to on-device flash memory, either use the *Make Permanent* button, or use *F2* keyboard button. The PMU status LED will flash with orange color (See *PMU Status* section).
To save a copy of your current project on hard drive either use *Ctrl+S* keyboard shortcut, or use the Clients toolbar.

Right on the main screen there are also windows with monitoring functionality. They are updated in real time and show various parameters reported from particular pins as well as values of elements created by user.



Monitoring Windows

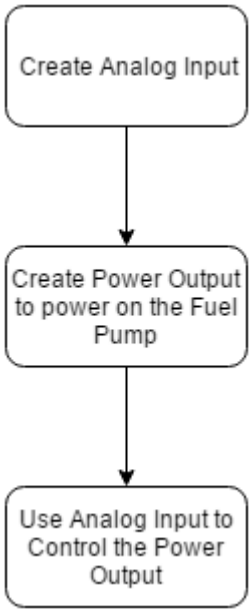
On the left, there is a *Tree View* double clicking any item on it, will bring up it's window to current desktop.

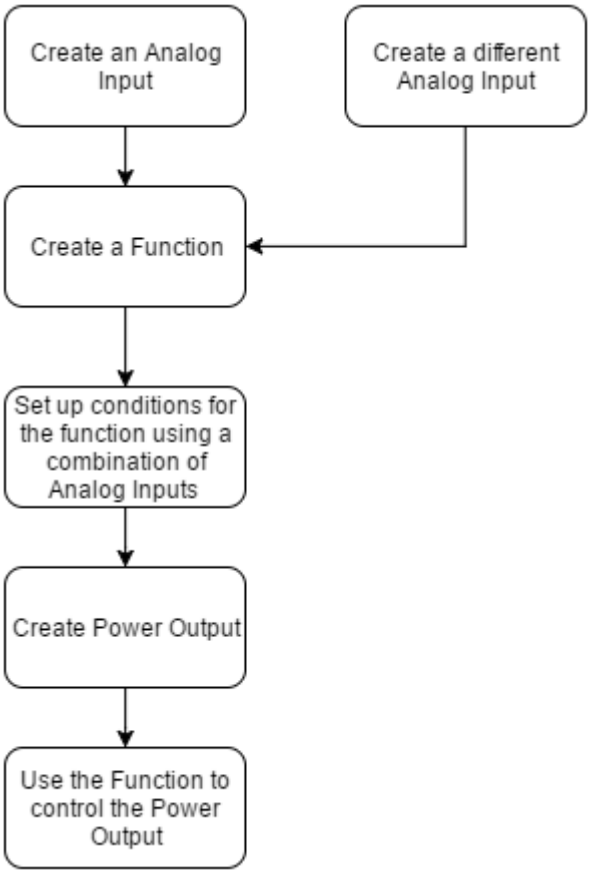


Tree View

PMU Client workflow

PMU Client workflow is really simple. You can monitor parameters of your channels, make adjustments, follow the graph log to understand channels behavior, you can create elements, then use those to control other elements or channels.
Elements and channels in this case can be anything, Power Output, Analog Input, Function, Switch etc.
Let's take a look at two more in-depth use cases.





These are just two examples of how PMU Client can be used, but the possibilities are endless.

ELEMENT TYPES

Analog Input

Analog inputs are input devices connected directly to the PMU.
Below is the list of analog inputs supported by PMU.

- Switch - Active High
- Switch - Active Low
- Rotary Switch
- Analog Sensor

Head to *Wiring* section to see wiring diagrams for each type of analog input.

Power Output

Power Outputs are devices powered by PMU which are connected to particular output pins.
Power Output examples:

- Fuel Pump
- Fan
- Oil Pump

Every Power Output has over current, under current (both user configurable) and overheat protection. They can be either turned on by default, controlled by function or triggered by another element such as Analog Input

CANbus Message Object (Mob)

CANbus Message Objects are interfaces needed for data gathering from CAN Stream. They read a particular number of frames (user defined, up to 8) starting from particular ID (also user defined) from chosen CANbus. Maximum of 8 CANbus Message Objects can be created for each CANbus. They also have special timeout flag which can be used to set up a different strategy if a connection from another CAN device times out.

CANbus Input

CANbus Inputs use CANbus Message Objects to extract data from chosen frame. They are able to read a user defined number of bits starting from user defined position, then apply offsets, division or multiplication to the extracted data.

CANbus Keyboard

CANbus Keyboard is an element that provides communication with Keyboard. It is used to define and handle Keyboard buttons, set their color, type etc.

Switch

You can use two types of switches:

- Latching Switch – switches between user defined number of states.
- Press Hold Switch – short press switches between high and low state, holding it triggers third state.

n Number

Number is simply an integer. It can be either a typed in constant or value of chosen channel. Number is mostly used for comparison or to control *Duty Cycle* in Pulse Width Modulation (See *PWM* section).

f Function

Function is one of the most important elements. It can be used to create a set of rules, conditions to Power on an output device. Function always evaluates to either **0** or **1**. If the function is used as control channel for a Power Output, the Power Output will be turned on if function returns **1** and turned off if function returns **0**. Functions will be explained in depth later on in *Functions* section.

Wipers Module

Wipers module is a separate module to control the Wipers. It consists of two Power Outputs for slow and fast wiper speed and an Analog Input for *Park Switch.* To use park functionality, Output Pin O8 is provided to use for slow wiper speed. To see the wiring instructions, head to *Wiring* part of the manual.

Blinkers Module

Blinkers module is a bit like *Wipers Module,* it is a separate element which controls the blinkers. It consists of two Power Outputs for left and right blinkers and needs three inputs (ie. CANbus Keyboard) to control the Left Blinker, Right Blinker and Hazard Lights.

CANbus Export

CANbus Export allows you to broadcast (send) data to the CANbus, which then can be used by other CAN device. For example, you can transmit the status of a Fan to other PMU or EMU Black

MANAGING ELEMENTS

Saving elements

Most elements can be saved to hard drive. To save your element, click icon on elements toolbar and choose where you want to save it.

Loading saved elements

Elements that can be saved, can also be loaded. To load your element, click icon on elements toolbar and choose file to load.

FUNCTIONS

Main Principle

The idea behind functions is to create a set of rules by combining different *operations for* various channels or elements. This set of rules is evaluated to a logical **true (1)** or **false (0)** result. Function then can be used to turn, for example, Power Output on or off. Quick example could be turning the secondary fuel pump on if a fault is detected on the primary one.

Operations

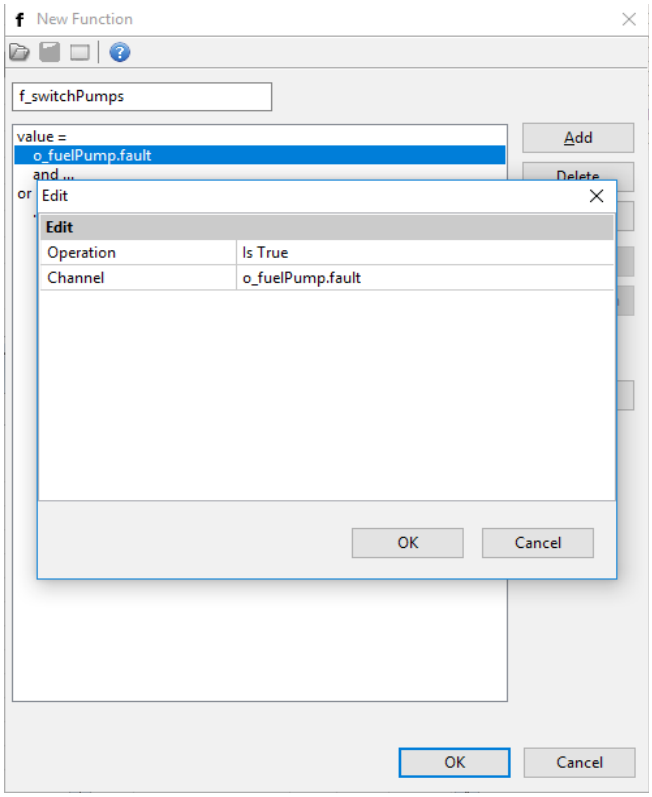
Following operations are available to use in functions:

Test Operations	
Is True	Returns **true (1)** if channel is true (its value is other than 0)
Is False	Returns **false (0)** if channel is false (its value is equal to 0)
Comparison Operations	
Equal	Returns **1** if **Channel** is equal to **Constant**
Not Equal	Returns **1** if **Channel** is not equal to **Constant**
Less	Returns **1** if **Channel** is less than **Constant**
Less or Equal	Returns **1** if **Channel** is less or equal to **Constant**
Greater	Returns **1** if **Channel** is greater than **Constant**
Greater or Equal	Returns **1** if **Channel** is greater than or equal to **Constant**
Logical Operations	
And	Returns **1** if both **Channel #1** and **Channel #2** are true
Or	Returns **1** if either **Channel #1** or **Channel #2** or both are true
Xor (Exclusive Or)	Returns **1** if either **Channel #1** or **Channel #2** are true, but not both at the same time
Pulse Generation Operations	
Flash	Flash is triggered by **Channel** and returns **1** for the time specified in **Time On** and **0** for the time specified in **Time Off** as long as **Channel** is in High State. If Low State is detected, Flash will imediately turn off.
Pulse	Pulse switches between **1** for the time specified in **Time On** and **0** for the time specified in **Time Off** for the amount of times specified in **Count**. It can be set to trigger by Rising Edge or Falling Edge. Pulse will still continue for Count number of times even if Channels signal is lost. Pulse will also ignore any Channel input until Count is reached. If Time On is set to 0, Pulse will generate a short impulse.

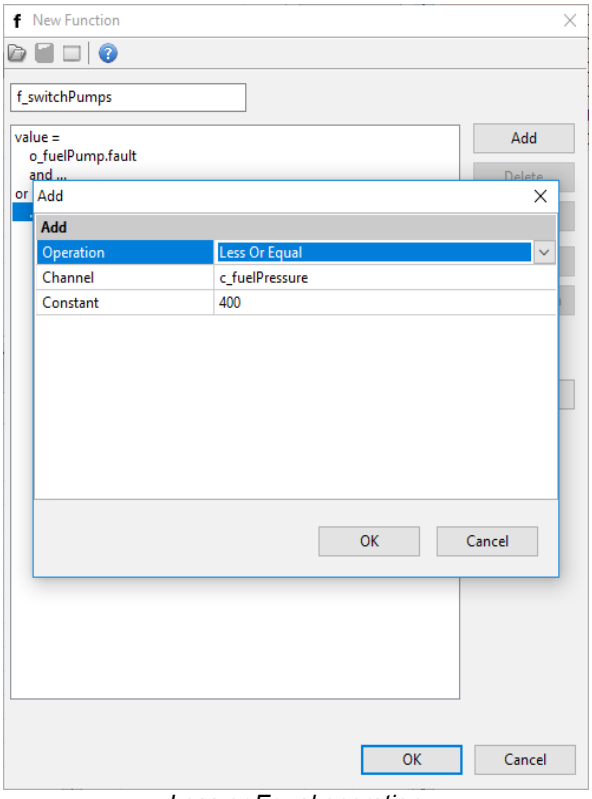
Operations with state	
Toggle	Toggles between **0** and **1** whenever **Channel** triggers a signal edge. It can be set to trigger by Rising Edge or Falling Edge If Default State is checked, default state of the Function after device is turned on will be 1, otherwise 0. Toggle remembers its last state, therefore if **Channel** signal is lost, Toggle will remain in the last steady state.
Set-Reset Latch	**Set Channel** sets the Latch to **1** when a high state is detected on the selected **Channel**, **Reset Channel** resets the Latch to **0** when a high state is detected on the **Channel** selected here. Reset has a priority, therefore if Channels both for Set and Reset are at high level, the Latch will be reset. If Default State is checked, default state of the Function after device is turned on will be 1, otherwise 0. Set-Reset Latch remembers its last state, therefore if both Channels are at low state, Set-Reset Latch will remain in the last steady state.

Function examples

We want the secondary fuel pump to turn on if a fault is detected on the first one or when fuel pressure drops below or is equal to 400kPa. To do this, we need to create a new function, let's call it *f_switchPumps* . Now to test for the first condition we will use ***Is True*** operation and to test for the second, we will use ***Less Or Equal*** operation. Their configuration will look like this:

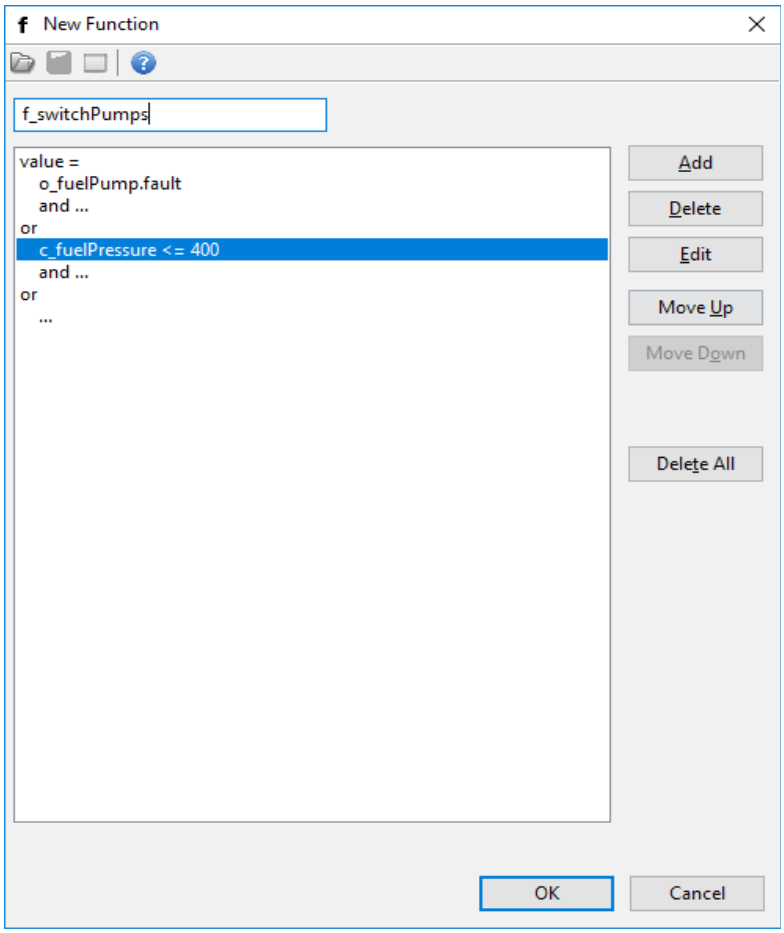


Is True operation



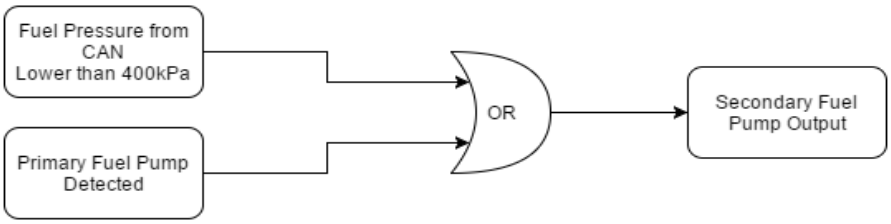
Less or Equal operation

And the final function will look like this:



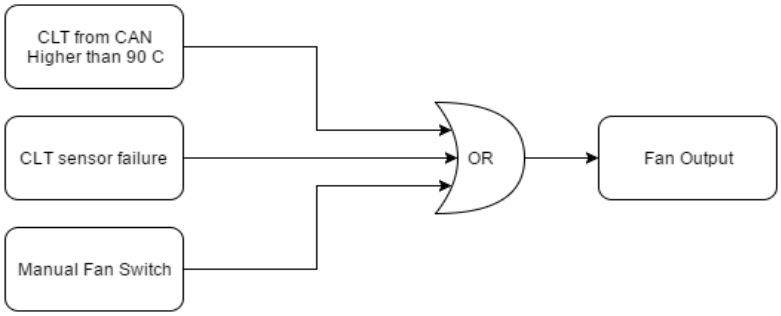
Switch Pumps function

he flow diagram for that function:



Switch Pumps function diagram

Another diagram example of Fan Output:



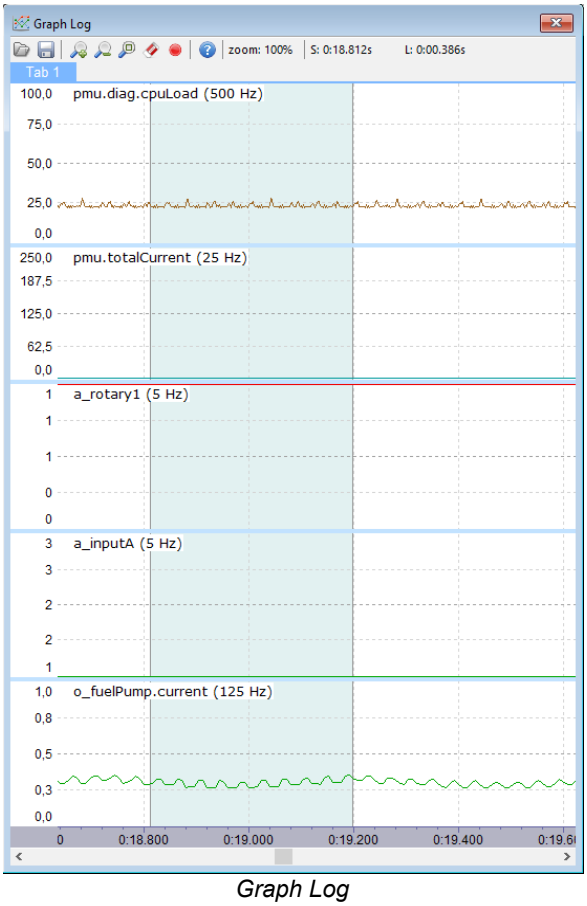
Fan Output function diagram example

CHANNEL LOGGING

Graph Log

PMU Client is equipped with a tool to show channels graph in real time. There are two ways to add a channel to the graph. First way is to right click on the graph log window, then select *Add* and type the channels name. The second way is to select a channel in Output Monitor, Analog Monitor Variable Selector, or any window from *Tree View → Log* group, then press *Insert* keyboard key. By default two channels are displayed on graph log – the *pmu.diag.cpuLoad* which displays the CPU load and the *pmu.totalCurrent* which displays total current drawn from PMU.

Here is how the Graph Log looks like with few more channels added:



Graph Log

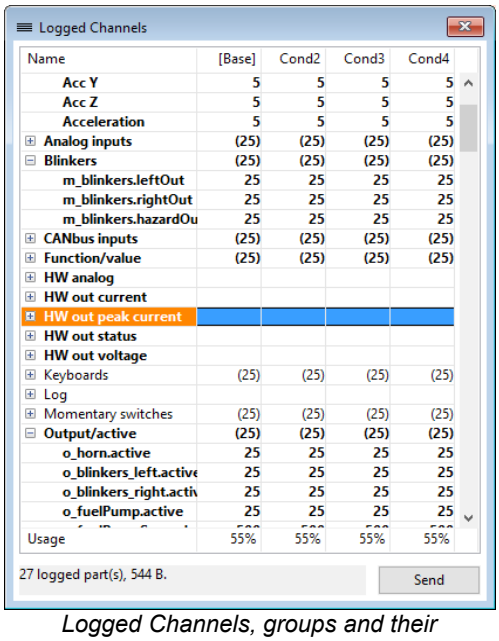
Logging Frequency

You can change the logging frequency of a channel by right clicking the channel on graph log, then selecting *Set Log Frequency* and choosing the desired frequency. Another way is to use Logged Channels window *(Tree View → Logged Channels)* . In that window you can see how parameters are groupped and their logging frequency. It is possible to change logging frequency of a whole group in the same way as a single channel.

Bolded parameters are the ones being logged (not all parameters are logged by default !). Double clicking an item here will toggle between logging and not logging it.
If a new element is created, every subchannel related to it is logged as well.

If you are familiar with EMU logging these are the key differences:

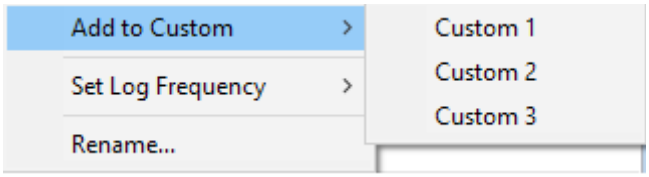
- Not all parameters are logged by default
- Parameters are groupped and their logging frequencies can be changed from 1 Hz to 500 Hz
- Every time an element is created, its subchannels are logged



Logged Channels, groups and their frequencies

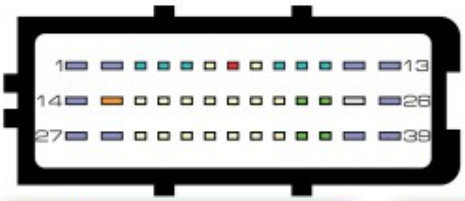
Custom Log

PMU Client allows user to have 3 separate custom logging groups. Any channel can be added to custom log *(Tree View → Custom1, Custom2, Custom3)*. To add channel to custom group, press right on the channel you want to add to custom log group and select *Add to Custom* then choose the group.



Adding Channel to custom log

PMU PINOUT



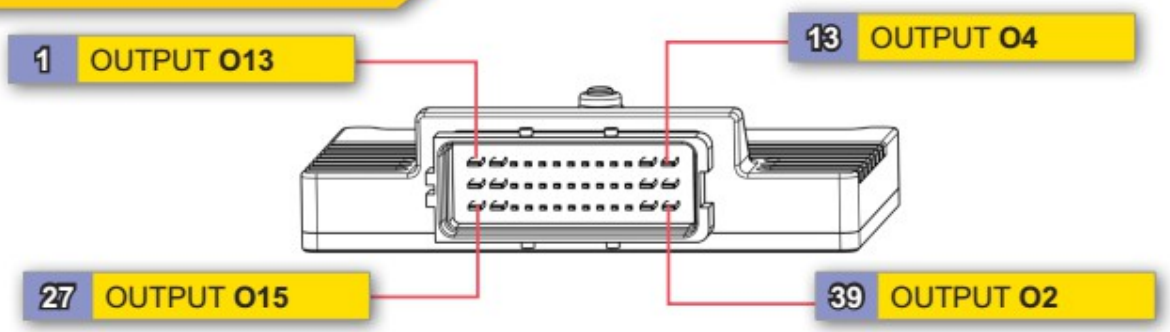
PMU-16

POWER MANAGEMENT UNIT

1	OUTPUT **O13**	14	OUTPUT **O14**	27	OUTPUT **O15**
2	OUTPUT **O12**	15	+5V OUTPUT	28	OUTPUT **O16**
3	OUTPUT **O11**	16	INPUT **A2**	29	INPUT **A1**
4	OUTPUT **O10**	17	INPUT **A4**	30	INPUT **A3**
5	OUTPUT **O9**	18	INPUT **A6**	31	INPUT **A5**
6	INPUT **A9**	19	INPUT **A8**	32	INPUT **A7**
7	+12V SW	20	INPUT **A11**	33	INPUT **A10**
8	INPUT **A14**	21	INPUT **A13**	34	INPUT **A12**
9	OUTPUT **O8**	22	INPUT **A16**	35	INPUT **A15**
10	OUTPUT **O7**	23	CAN1H	36	CAN1L
11	OUTPUT **O6**	24	CAN2H	37	CAN2L
12	OUTPUT **O5**	25	GROUND	38	OUTPUT **O1**
13	OUTPUT **O4**	26	OUTPUT **O3**	39	OUTPUT **O2**

25A	15A	**O8** - Wipers output

HOW TO READ:



Output Pins

There are 16 output pins for you to use, 10 pins up to 25A and 6 pins up to 15A. They can be also used in parallel to increase current capacity (See *Using output pins in parallel* section)
All Pins are equipped with over and under current protection, short circuit protection, as well as thermal protection. In case of any of this scenario happening, the output pin will be shut down and appropriate message will be displayed in the PMU Client and the PMU itself.
For 25A Output Pins, Soft Start is available and PWM with Duty Cycle control (See *PWM* section)

Input Pins

There are 16 input pins for you to use as well as a separate +5V Pin to provide power for Analog Inputs such as rotary switches, or analog sensors.

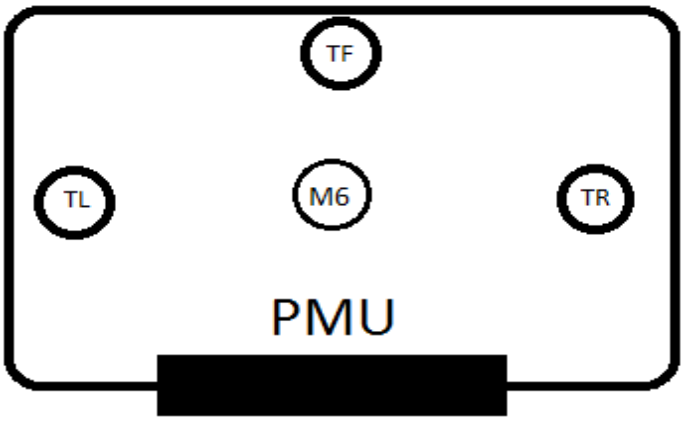
PMU TEMPERATURE, PLACEMENT AND LOAD BALANCING

PMU Placement

PMU should be located in a place that protects it from weather conditions, road debris and road hazard. Even though PMU case works as a radiator, it is also recommended to place PMU somewhere where heat can be dissipated easily, preferably with good airflow.

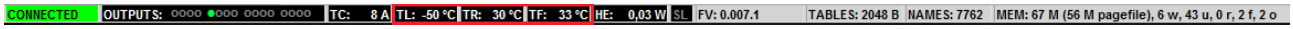
PMU Temperature

PMU has three thermometers located on its board to independently measure temperature in three separate places.



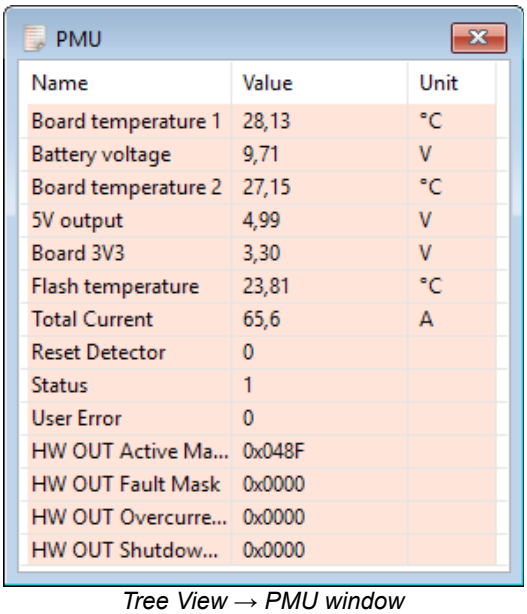
Thermometers location

There are two places where temperature can be checked in PMU Client, first is the status bar and the values *TL, TR, TF (Temperature Left, Temperature Right, Temperature Flash):*



Status Bar

Second is the PMU window which can be accessed by double clicking *PMU* on *Tree View:*

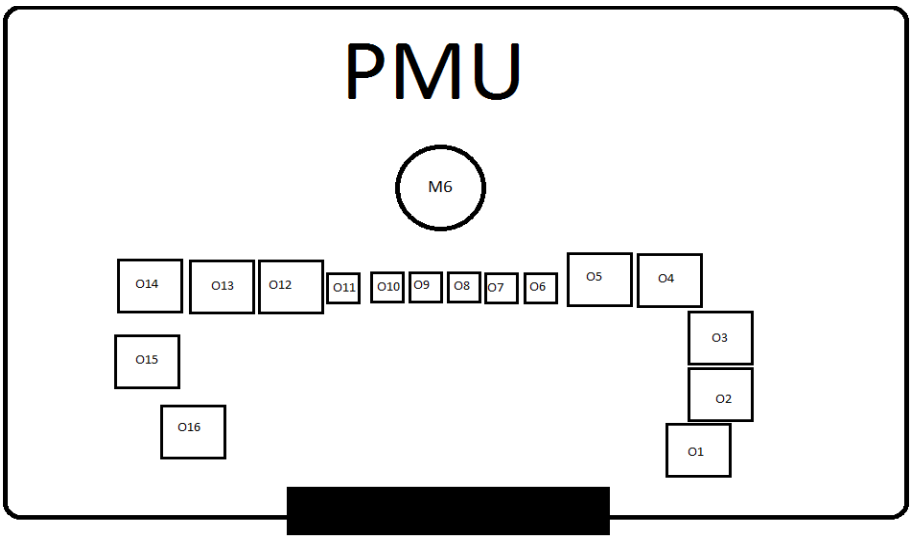


Tree View → PMU window

If you are experiencing high temperatures, it is advised to move the PMU to a colder place or provide better airflow to current location.

Output Load Balancing

Another way to handle high temperatures is to connect the devices that draw the most current in a way that the transistors for their Output Pins are not grouped up together. In other words, it is a good rule of thumb not to connect devices that draw the most current to output pins which transistors are located next to each other. This way they can dissipate heat better, resulting in lower temperatures. Picture below shows the transistor placement for every output pin:



PMU output transistors location

Output Pin signaling and status

PMU has an ability to signal condition of each Power Output. It is displayed on the device itself and in PMU software. When a new Power Output is created, a special variable which represents the status of Power Output, called *output_name.status* is created as well. This variable can be used to create failsafes or conditions, to set up redundancy etc.

All possible values of each o*.status:

Status value in Client	LED color	Status
0	None	OFF
1	Green	ACTIVE (ON) (everything okay)
2	Orange	UNDERCURRENT
3	Red	OVERCURRENT (software limit detected and output is tripped-off by software)
7	Red	THERMAL SHUTDOWN (hardware shutdown occurs, overtemperature)

[1]) Consider everything equal or greater than 3 as OVERCURRENT (software or hardware).

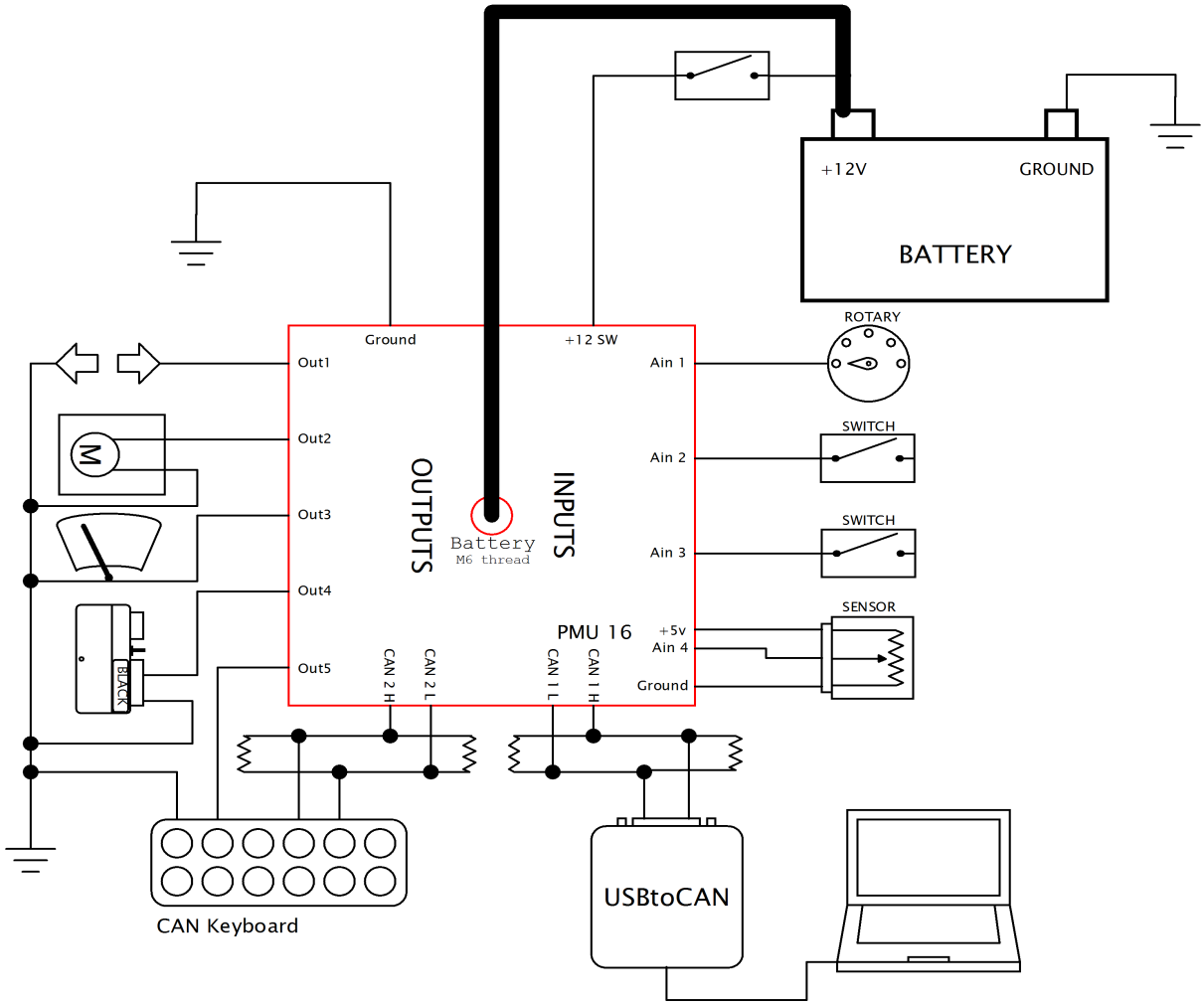
[2]) In UNDERCURRENT condition, the output may work (.active=1) or not (.active=0)

[3]) OVERCURRENT means that the current over time is higher than specified parameters (maybe you need to tune parameters)

[4]) THERMAL SHUTDOWN means that no matter what parameters you specify, your setup WILL NOT WORK (you will need to split the output into two pins, install an additional flyback diode, disable the SoftStart option, lower the PWM frequency, or disable PWM at all)

WIRING

Basic diagram



PMU Basic Diagram

This is the basic PMU communication and connection diagram which contains few key elements:

- PC Communication takes place on CAN 1 as this is the CAN specified to use for PC connection.
- CAN 1 has two 120 Ohm terminators on CAN bus. They are necessary, as PMU does not provide termination on CAN 1.
- CANbus Keyboard is connected to CAN 2.
- Power to PMU is supplied in two ways: First using the ignition which connects to +12SW Pin (See *PMU Pinout* section), second using positive battery terminal which connects to M6 Bolt located on PMU case.

Wire Size

Suggested wire size for continuous current (chassis wiring, Tefzel)

5A	10A	15A	25A
1mm^2	1.5mm^2	2.5mm^2	4mm^2
AWG 18	AWG16	AWG14	AWG12

For the power connecter at least 25mm^2 (AWG3) is required !

Load examples for popular devices

Device	Inrush Current [A]	Continuous Current [A]
Fuel Pump	15	7
Fan	50	20
Front Lights	16	10
EMU Black	30	25

Connector current capacity

Outputs current capacity is limited by connector terminal current capacity.

The Sicma 2.8 terminals current capacity is 25A @ 23°C when only single terminal is loaded and AWG12 (4mm^2) wire is crimped correctly.

The Sicma 1.5 terminals current capacity is 19A @ 23°C when only single terminal is loaded and AWG14 (2.5mm^2) wire is crimped correctly.

Poorly crimped terminal, not completely seated terminal or smaller than recommended wire size will reduce terminal current capacity.

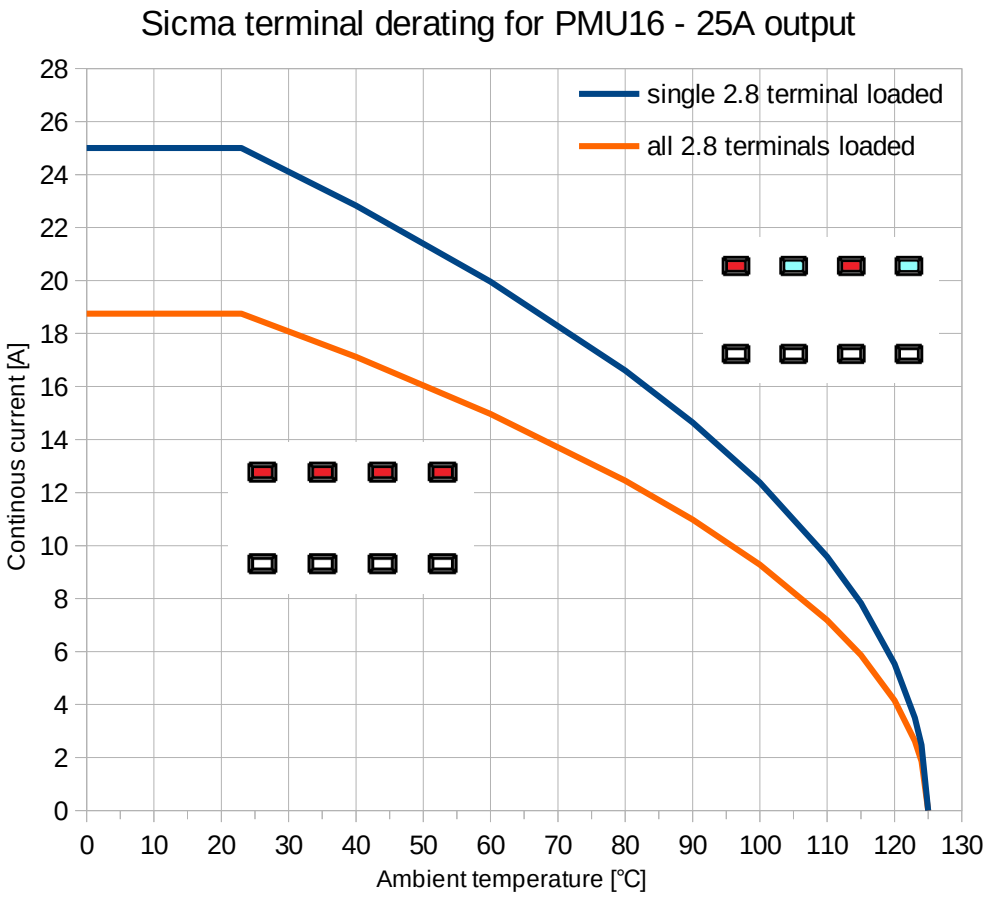
Terminals should be derated according to anticipated temperature of connector environment and used load balancing pattern.

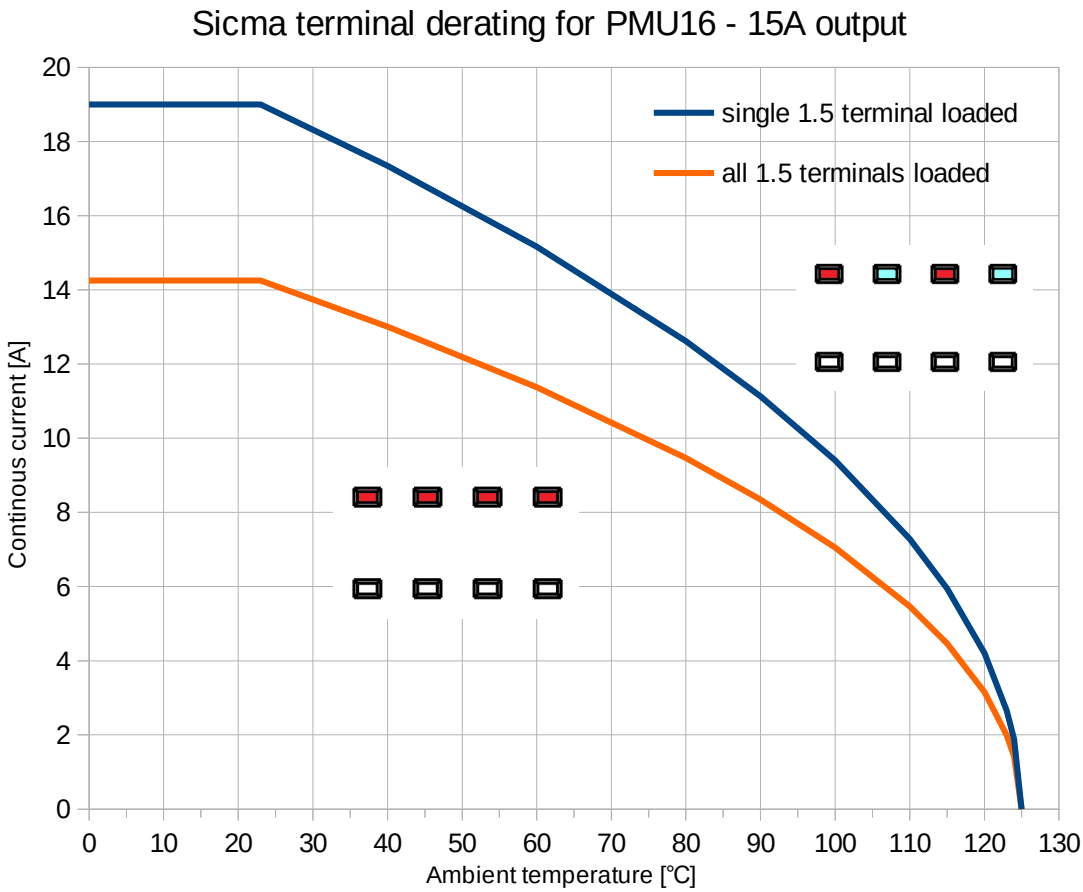
To safely utilize current capacity, don't place heavily loaded terminals next to each other.

Two 2.8 terminals placed next to each other can only carry 38A total @ 23°C not 50A!

Please also note the temperature derating.

Single 2.8 terminal working in 40°C ambient temperature can safely conduct only 23A current.



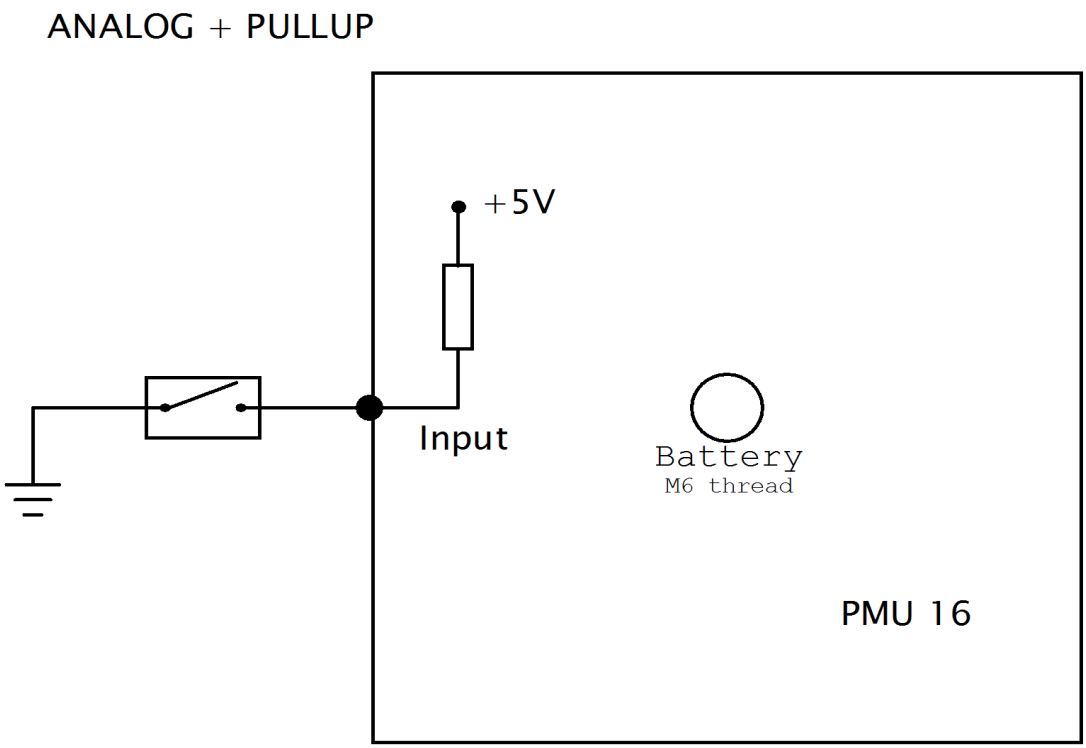

Sicma terminal derating for PMU16 - 15A output
Continous current [A]
Ambient temperature [°C]
single 1.5 terminal loaded
all 1.5 terminals loaded
0
2
4
6
8
10
12
14
16
18
20
0
10
20
30
40
50
60
70
80
90
100
110
120
130

ANALOG INPUT WIRING

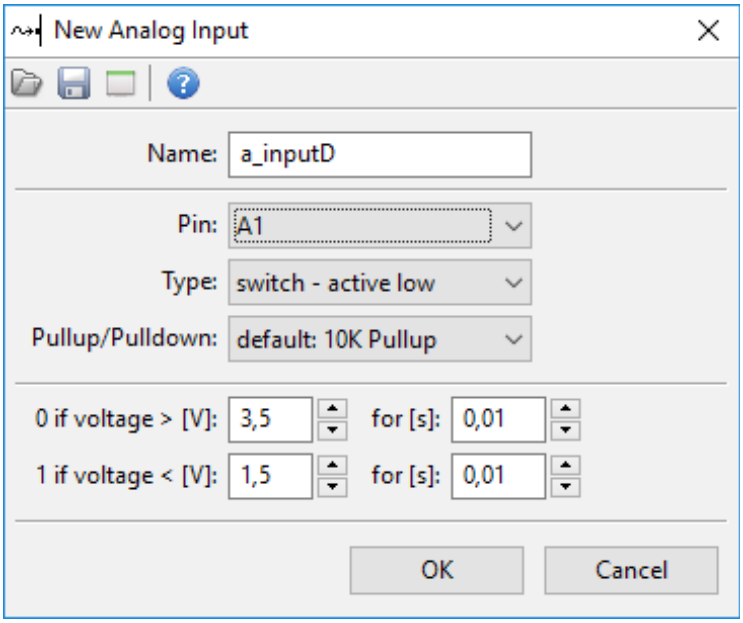
Wiring schematics

Most important thing to remember is that PMU has dedicated +5V Pin (See *PMU Pinout* section) can be used for things like rotary switch, active low switches and analog sensors.
Below are basic wiring diagrams.

- Switch connected to GND with Pull Up

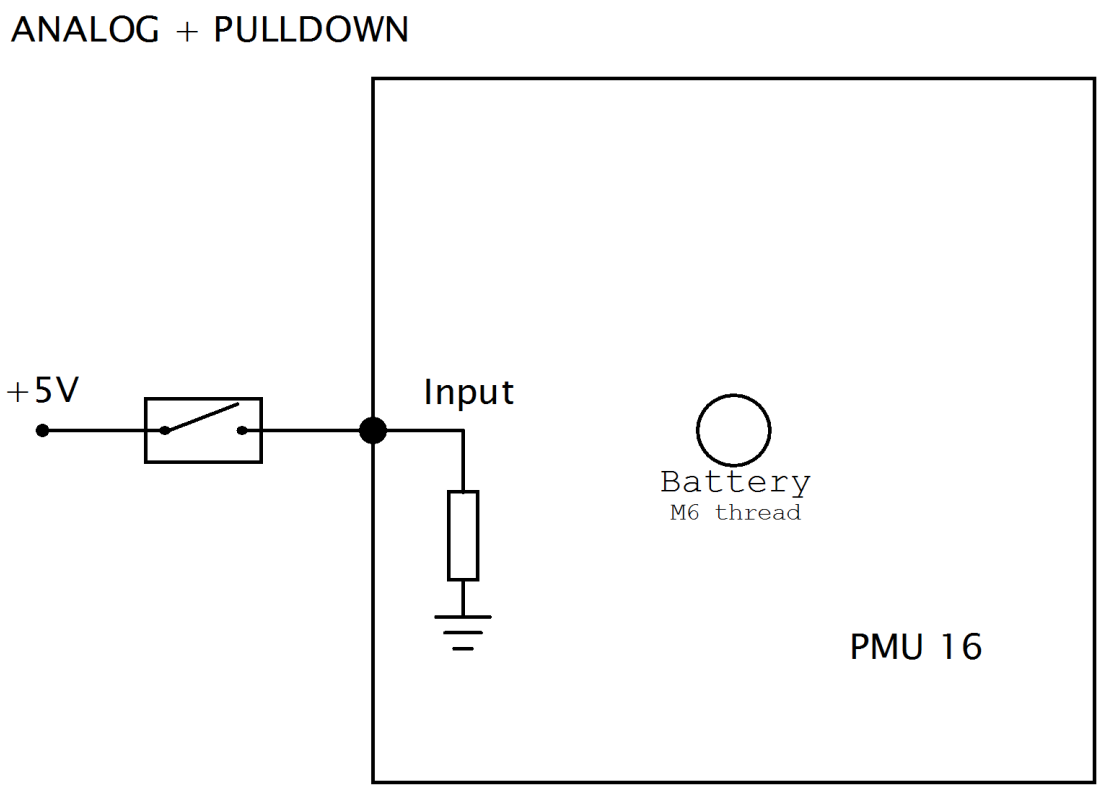


- PMU Client configuration:

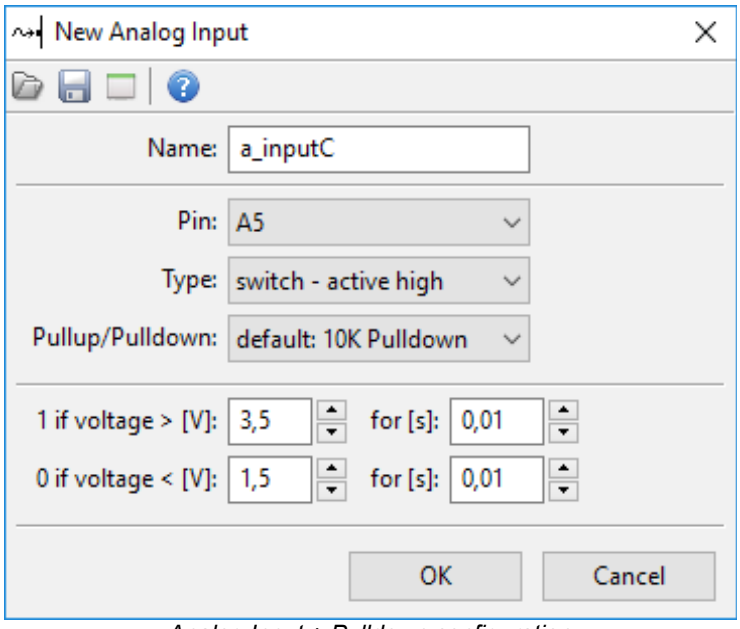


Analog Input + Pullup configuration

- Switch connected to +5V Pin with pull down:

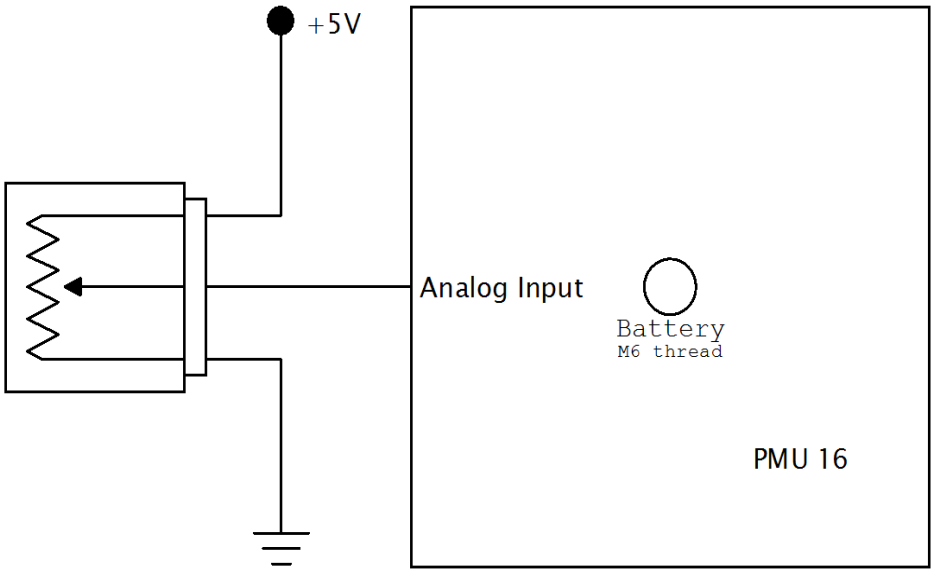


PMU Client configuration:

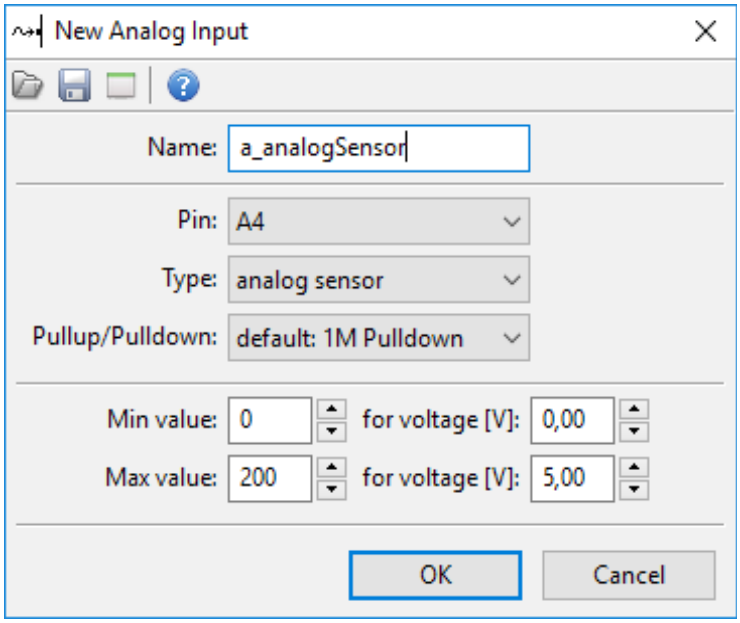


Analog Input + Pulldown configuration

- Rotary switch



- PMU Client configuration:

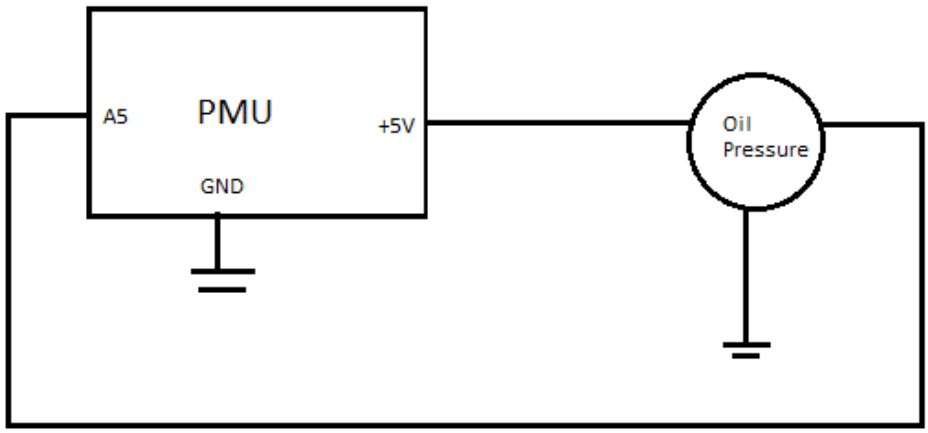


Rotary Switch configuration

Analog Sensor

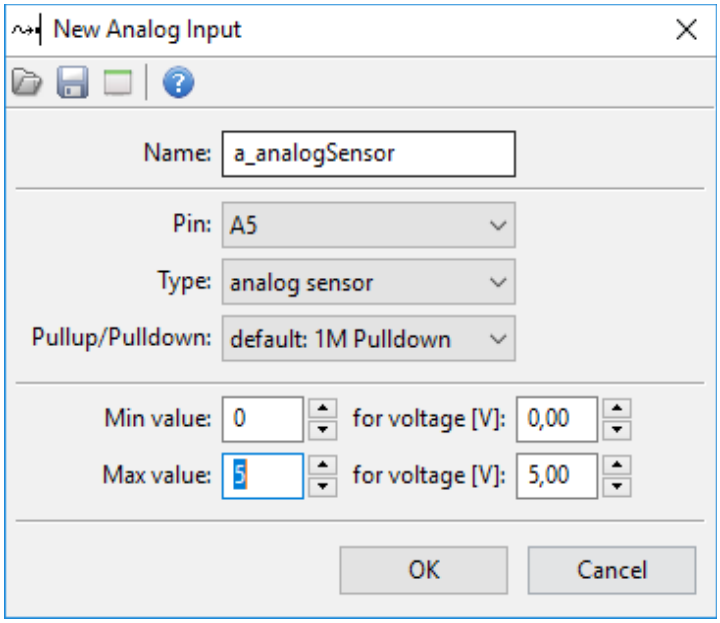
To connect analog Sensor, +5V and GND Pins must be used (See *PMU Pinout* section). Output from Analog Sensor should be connected as a regular Analog Input to PMU.

Analog Sensor diagram:



Analog Sensor diagram

PMU Client Configuration:



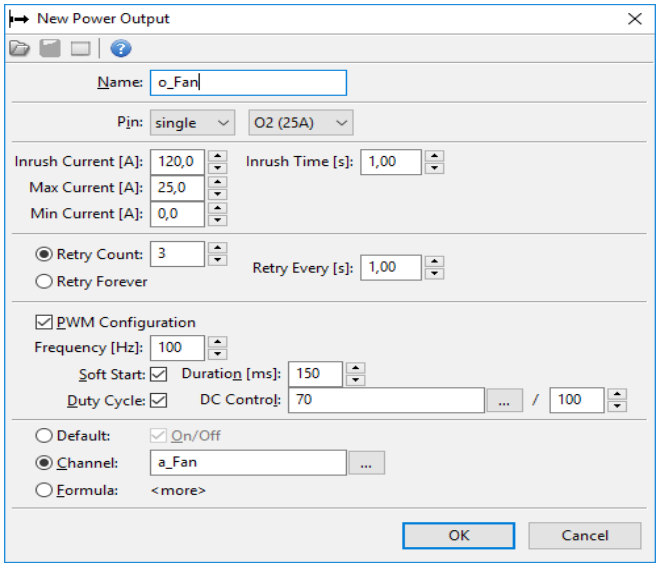
PMU Client analog sensor configuration

OUTPUT DEVICES

Power Output

Power Outputs are elements that control external devices. You can set up the Minimum Current, Maximum Current, Inrush Current, Inrush Time, PWM, and mechanisms to switch the Power Output on or off.

PMU Client Configuration:

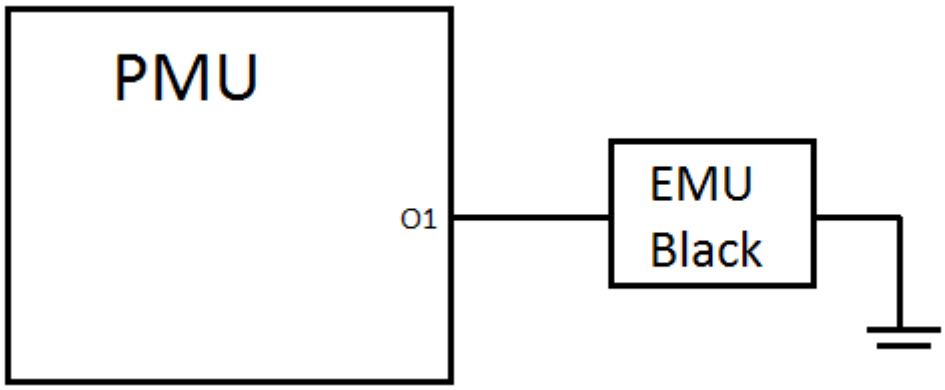


Power Output Configuration

Simple Power Output Setup

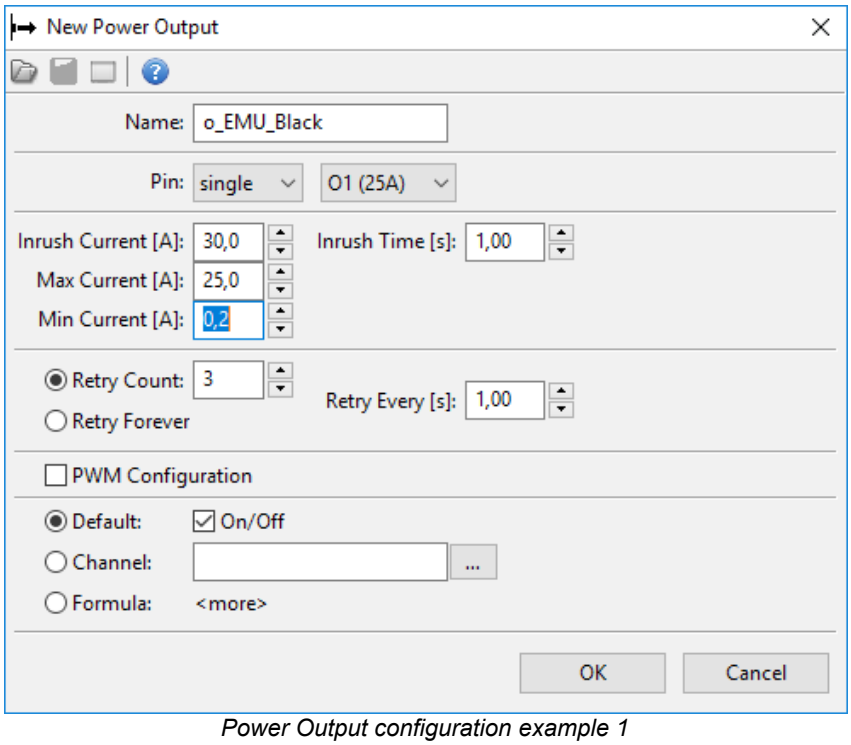
This is a quick example of two popular setups, their wiring diagrams and PMU Client configuration.
Please note that output pins shown here are just an example.

- Power Output that is turned on as soon as PMU is turned on:



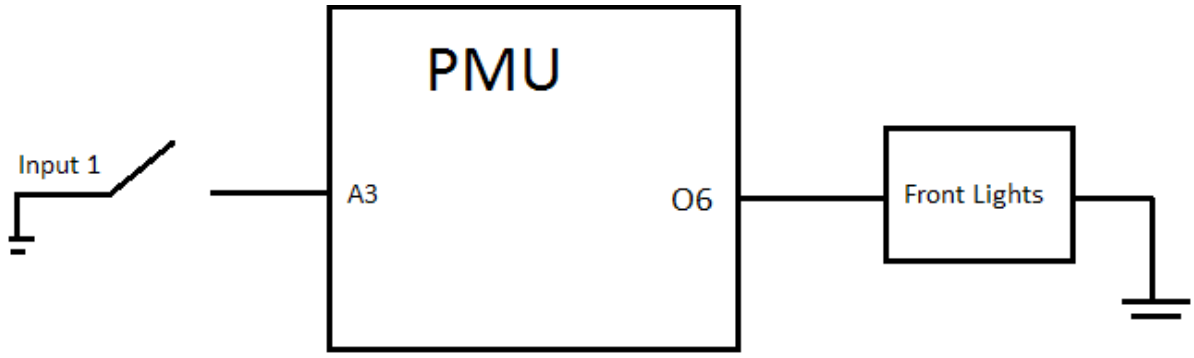
Power Output Diagram example 1

PMU Client Configuration:



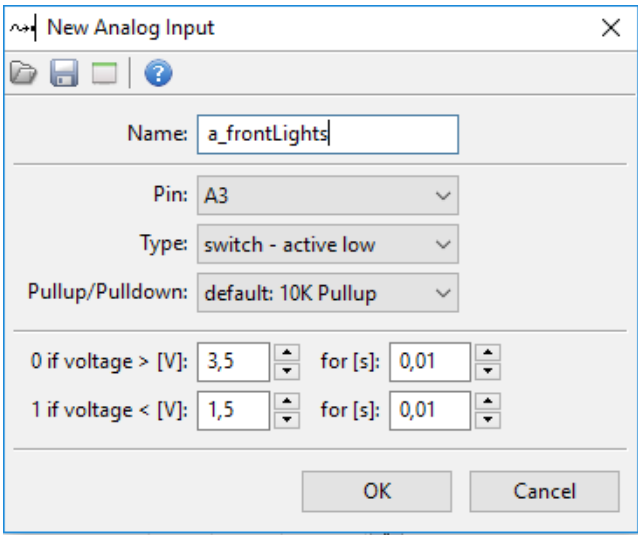
Power Output configuration example 1

- Power Output that gets switched on by an Analog Input:

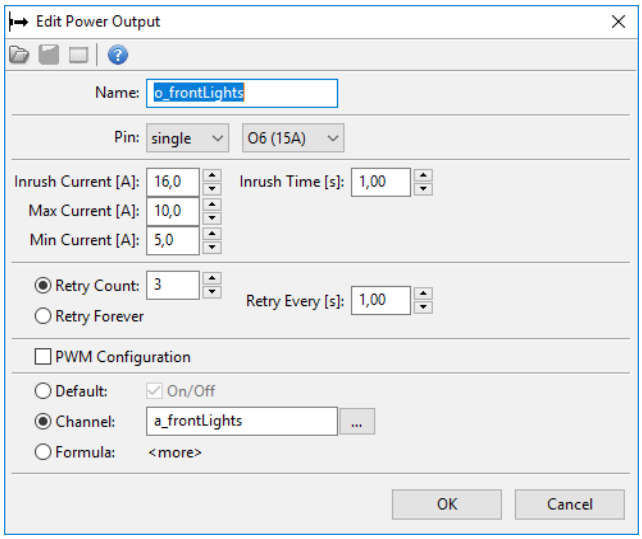


Power Output diagram example 2

PMU Client Configuration:



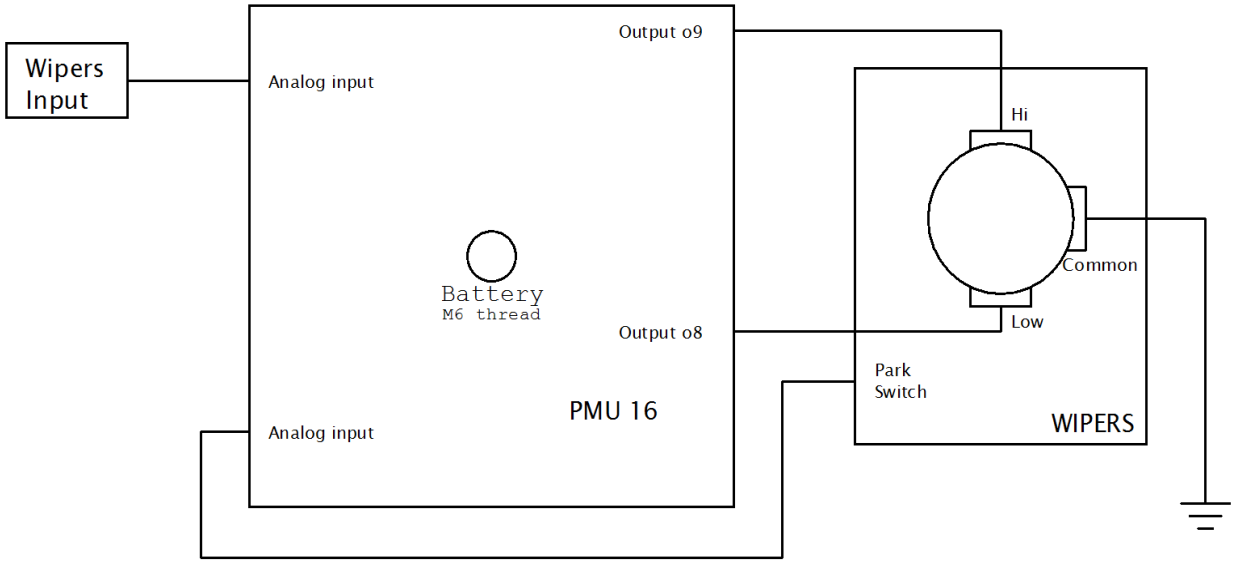
Analog Input configuration example 2



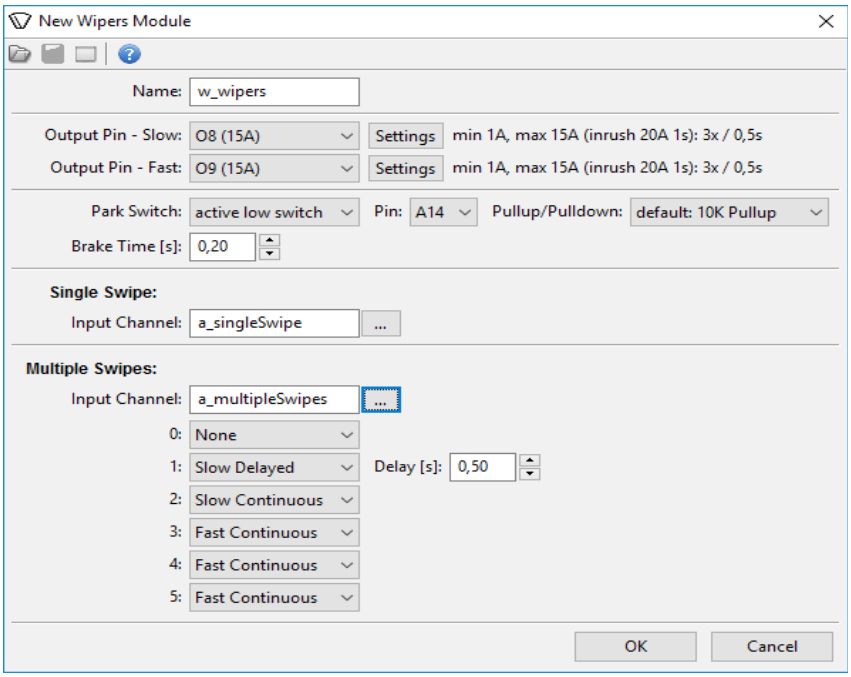
Power Output configuration example 2

Wipers

To connect wipers O8 Output Pin must be used for Slow Wiper output. This is the Pin provided to use and it's the only way to use the park switch ability. For fast wiper output any output pin can be used except for O8. Park switch must be connected as an Analog Input.

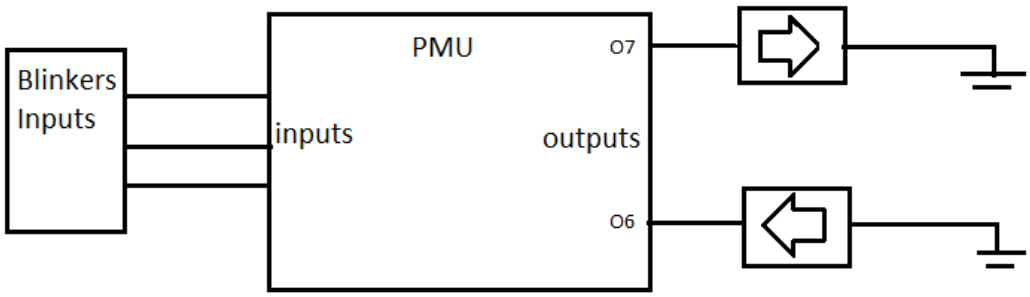


Configuring wipers in PMU Client is pretty easy. There is a module made specifically for wiper configuration and should be used to set up wipers trouble-free *(Project Tree → Wipers Module)*. PMU Client Configuration:



PMU Client wipers configuration

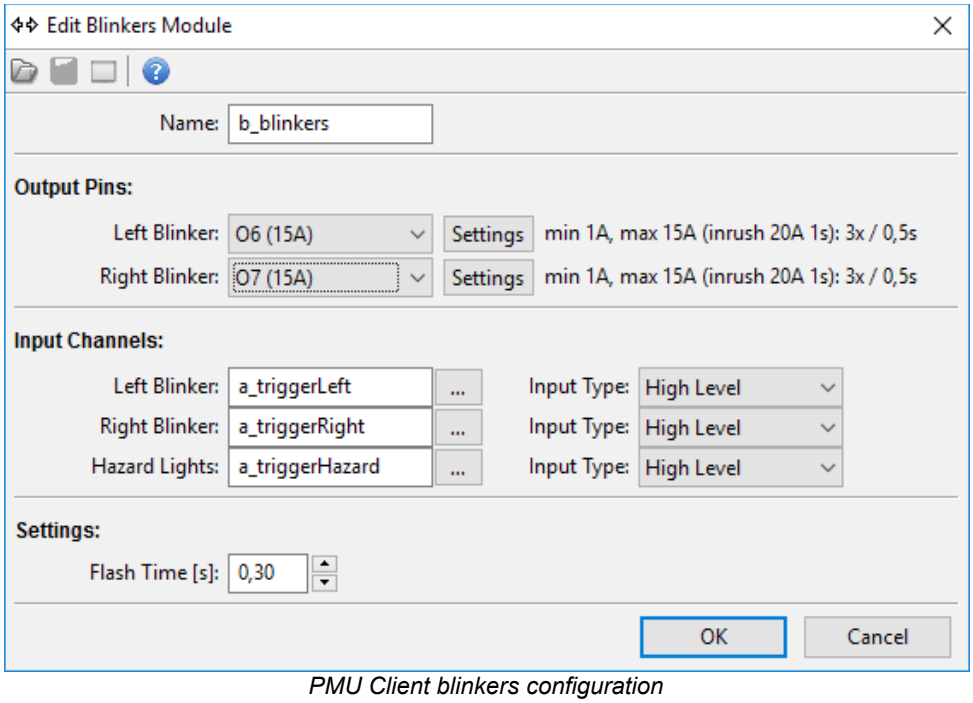
Blinkers



Blinkers diagram

Blinkers should be simply connected to PMU Output Pins. Three input channels should be used in this configuration - two channels to control blinkers, one channel to control hazard lights.

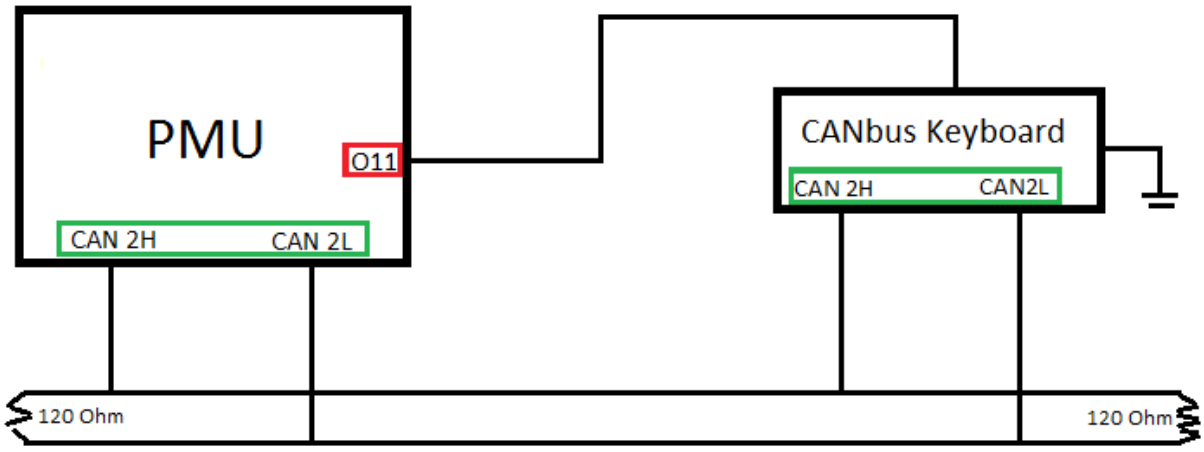
PMU Client also provides special module designed to configure blinkers in easy way (*Project Tree → Blinkers Module).*



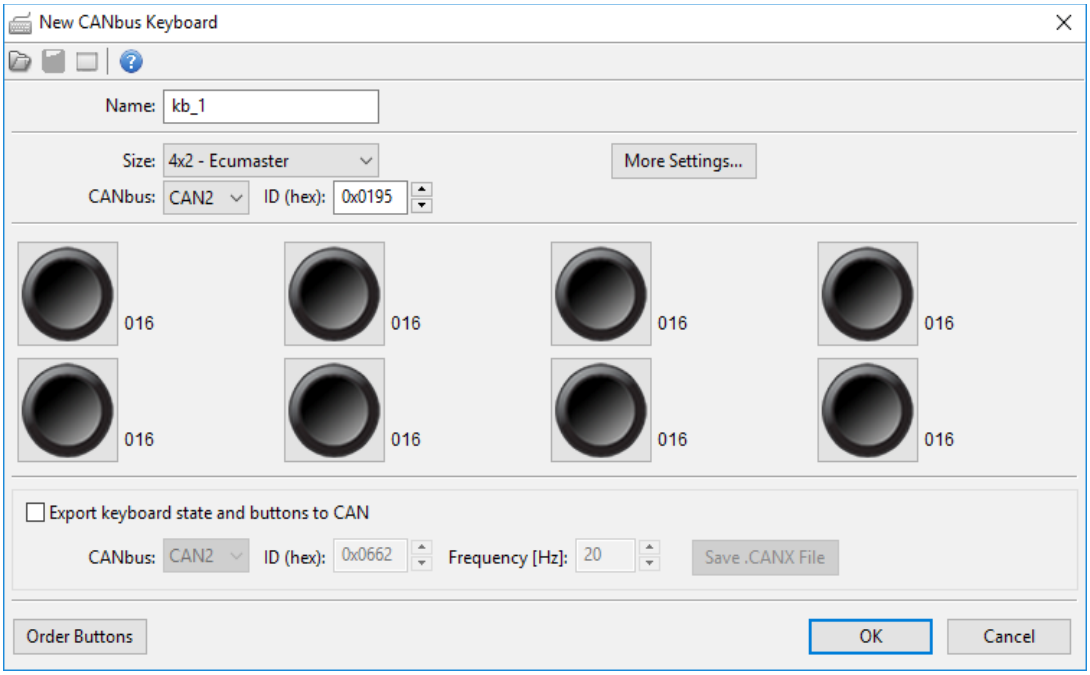
PMU Client blinkers configuration

CANbus Keyboard

CANbus Keyboard can be connected to any CANbus. If CAN2 was chosen, user has the ability to change CAN2 speed for CANbus Keyboards such as Life Racing (*Tree View → CANbus Setup).* CAN connection must be terminated at both ends. It is possible to use CAN2 terminator from PMU (*Tree View → CANbus Setup)*. To provide power to CANbus Keyboard one of Output Pins can be used. Up to two CANbus Keyboards can be connected.



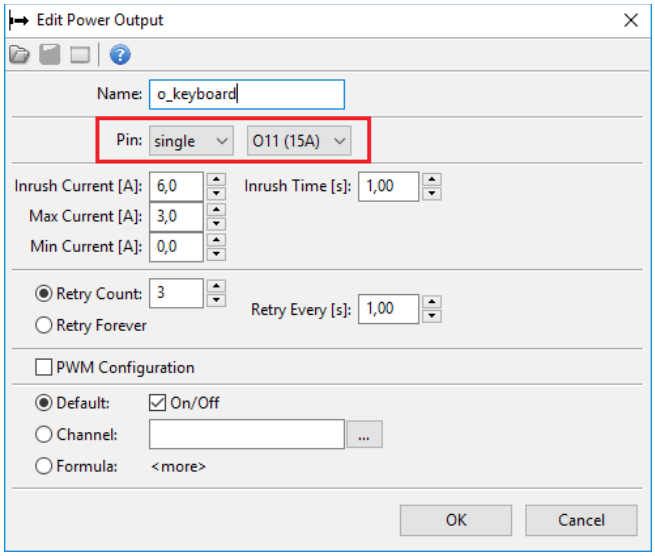
CANbus Keyborad diagram



PMU Client CANbus configuration

PMU Client provides special module for CANbus Keyboard configuration and setup *(Project Tree → CANbus Keyboard)*. Refer to your CANbus Keyboard manufacturers documentation to find out the ID under which CANbus Keyboard works at.

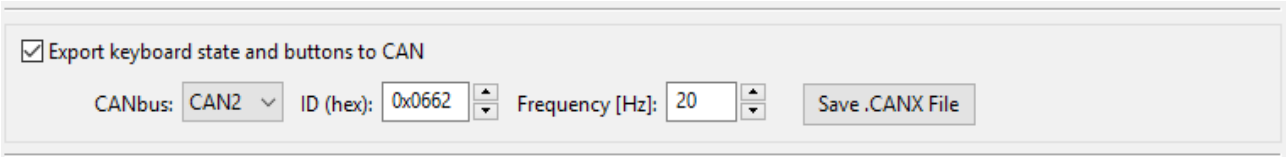
Power Output configuration for CANbus Keyboard:



CANbus Keyboard power output configuration

Exporting Keyboard state

While configuring CANbus Keyboard, you can choose to broadcast the Keyboard state and buttons on chosen CANbus Interface, ID and frequency so that other device such as secondary PMU or EMU Black can acquire and use them.



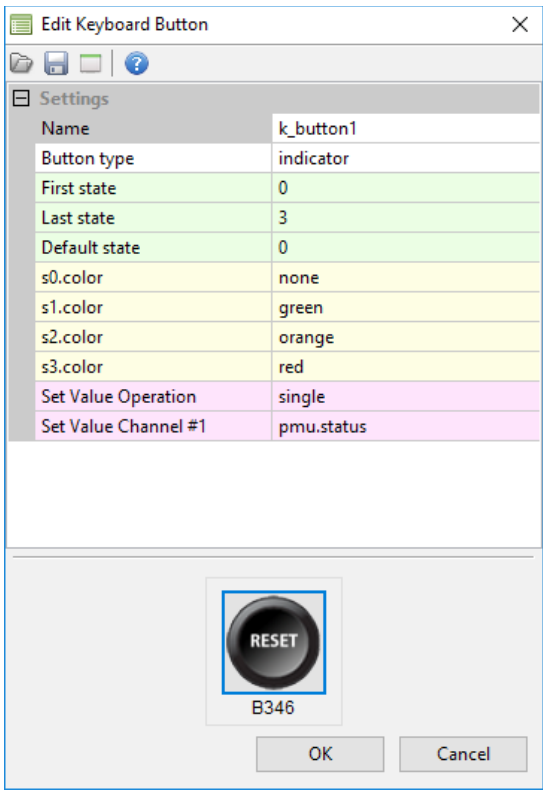
Keyboard Export

If CANbus Keyboard is supposed to be used by EMU Black as well, it can be solved in the following way:

- PMU controls the CANbus Keyboard, handles key switches, backlighting
- PMU exports keyboard state and buttons to CAN under ID 0x0662
- EMU Black reads data exported by PMU from CANbus on ID 0x0662

Using CANbus Keyboard to signal status

CANbus Keyboard can also be used as a signaling device. You can change the color of any button to represent either the PMU status, or any Power Output status (see *Output Pin signaling and status* section). To achieve that, a CANbus Keyboard must be configured and connected (See *CANbus Keyboard* section) and chosen button must be set up as an *indicator* button in PMU Client.
Here is an example configuration:



Keyboard status indicator configuration

This example configures button *k_button1* to display the value of *pmu.status* channel in the following way:

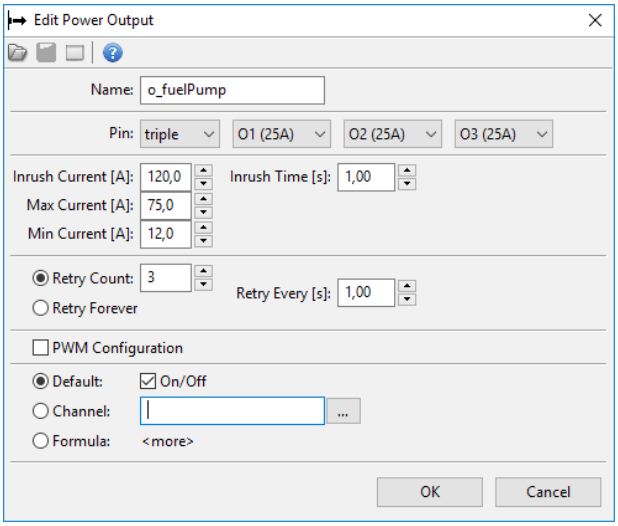
Status	Color	Description
0	None	All Power Outputs are Off (*pmu.status* equal to 0)
1	Green	At least one Power Output is active (on) and there is no under or over current condition (*pmu.status* equal to 1)
2	Orange	At least one Power Output is under current and there is no over current condition (*pmu.status* equal to 2)
3	Red	At least one Power Output is over current (*pmu.status* equal or above 3)

Keep in mind that the button still works as a normal non-latching switch and can be set up to, for instance, trigger Master Retry (See *Master Retry Channel* section) to reset all outputs.

Using output pins in parallel

PMU allows user to connect output pins in parallel to increase current capacity. Up to three output pins of the same amperage can be used, granting up to 75A continuous current.

PMU Client configuration example:



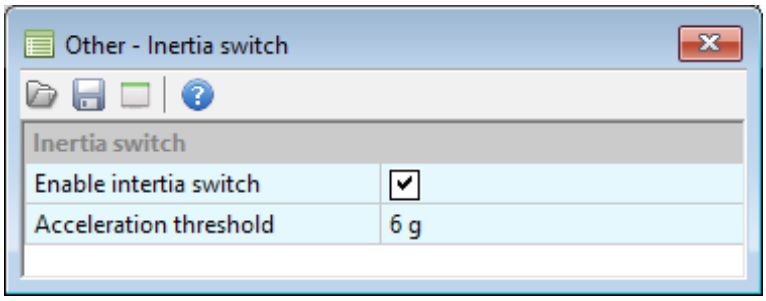
Power Output configuration with parallel pins

INERTIA SWITCH

PMU is equipped with a failsafe switch which gets triggered in case of a crash. PMU constantly monitors its gyroscopic values and acceleration in each axis to react quickly in the worst case scenario.

Inertia Switch

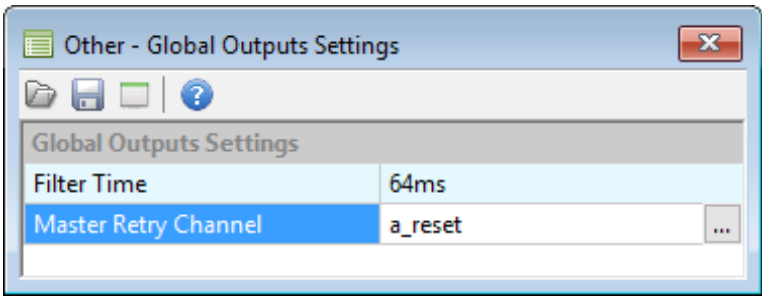
Inertia Switch immediately shuts down all Power Outputs to prevent any mishaps happening in the event of a crash. Inertia Switch gets triggered on after exceeding acceleration threshold specified by user. Inertia Switch can be accessed by *Tree View* → *Inertia Switch* :



Inertia Switch configuration

MASTER RETRY CHANNEL

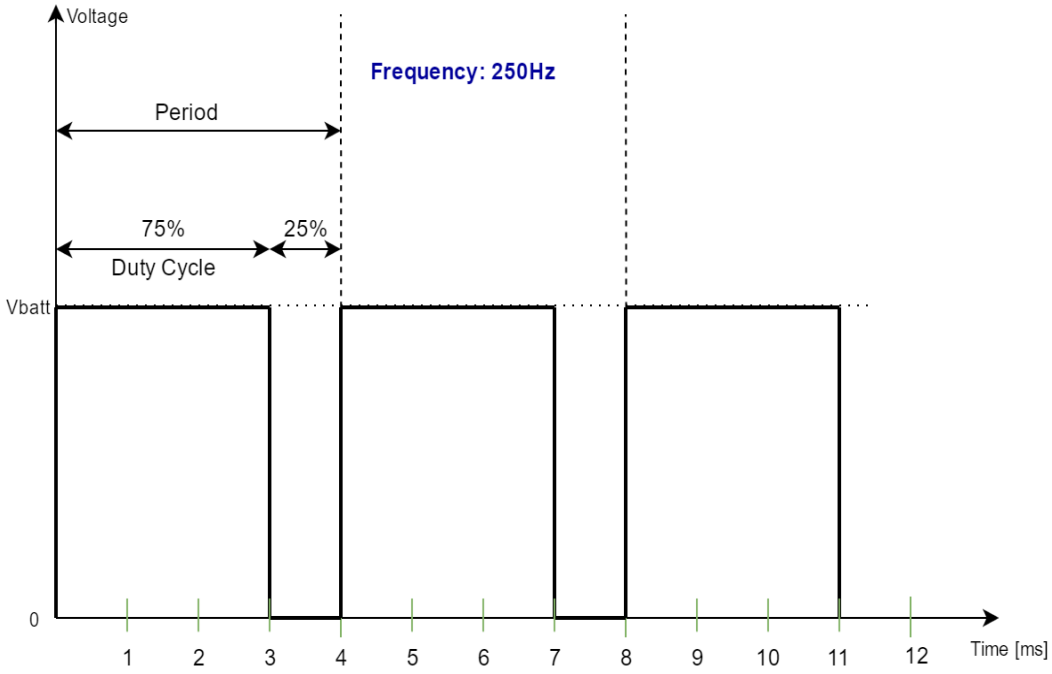
Master Retry Channel allows user to perform reset of Power Outputs. If a Power Output has signaled a fault, due to either over or under current condition, triggering a Master Retry will reset it back to default state. Master Retry can be triggered by any channel or element. Master Retry configuration can be found in *Tree View → Global Output Settings → Master Retry Channel .*



Master Retry configuration

PWM (PULSE WIDTH MODULATION)

PMU has the ability to apply PWM to 25A Output Pins.
The main principle of Pulse Width Modulation is to limit the amount of Power supplied to Power Output by alternating power Output on and off.

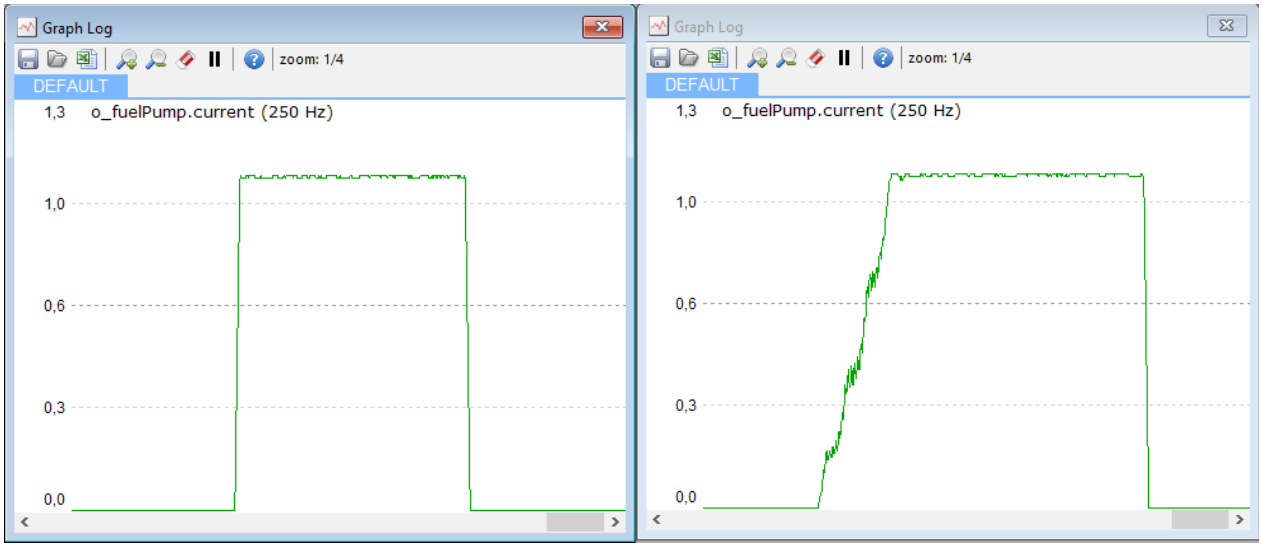


Pulse Width Modulation

Keep in mind that PWM introduces energy loss to heat due to transistor switching on and off. Higher frequencies generate more heat, therefore If you are experiencing overheated status on Power Outputs or high heat in general, either lower the *Frequency* of PWM, or use a flyback diode to eliminate flyback and lower the thermal load (See *Using flyback diode* section).

Soft Start

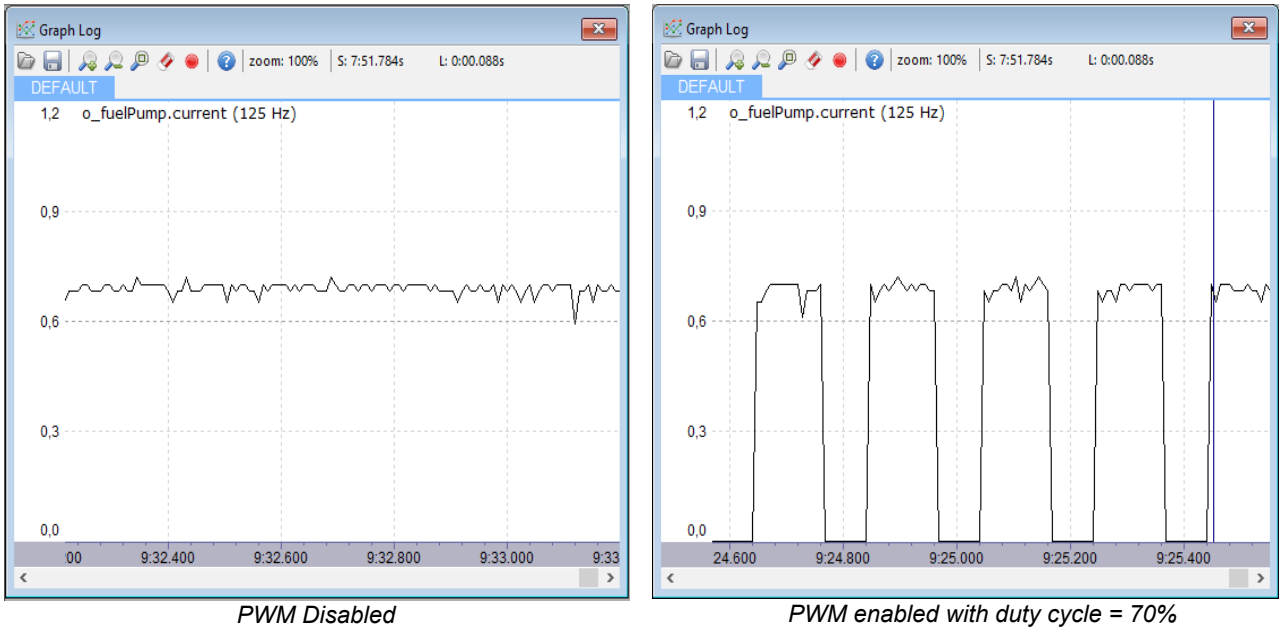
PMU has the ability to enable a Soft Start on 25A output Pins. Soft start should be used to prevent switch-on surges of greater altitude. By varying the time on and time off with chosen frequency and duration, a Power Output reaches its state gradually.



Soft Start Disabled *Soft Start Enabled*

Duty Cycle

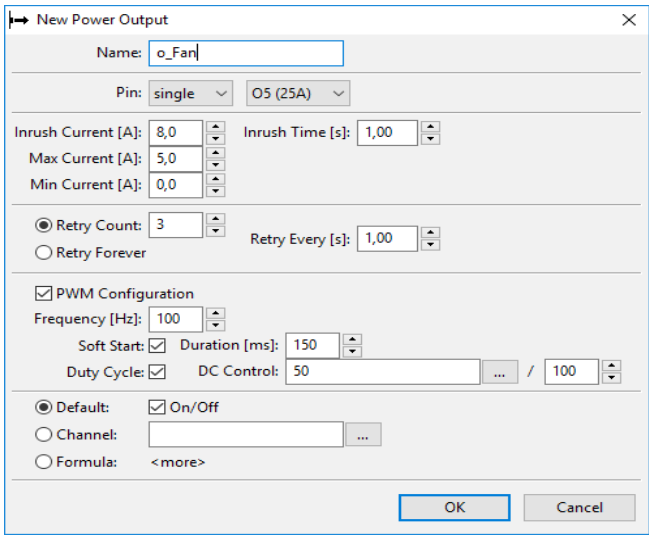
Duty Cycle represents the percentage of time the Power Output is turned On. If the Duty cycle is set to 50%, the Power Output will be turned on for ½ of time period.



PWM Disabled *PWM enabled with duty cycle = 70%*

Duty Cycle can be either set to constant value, or controlled by a Channel.

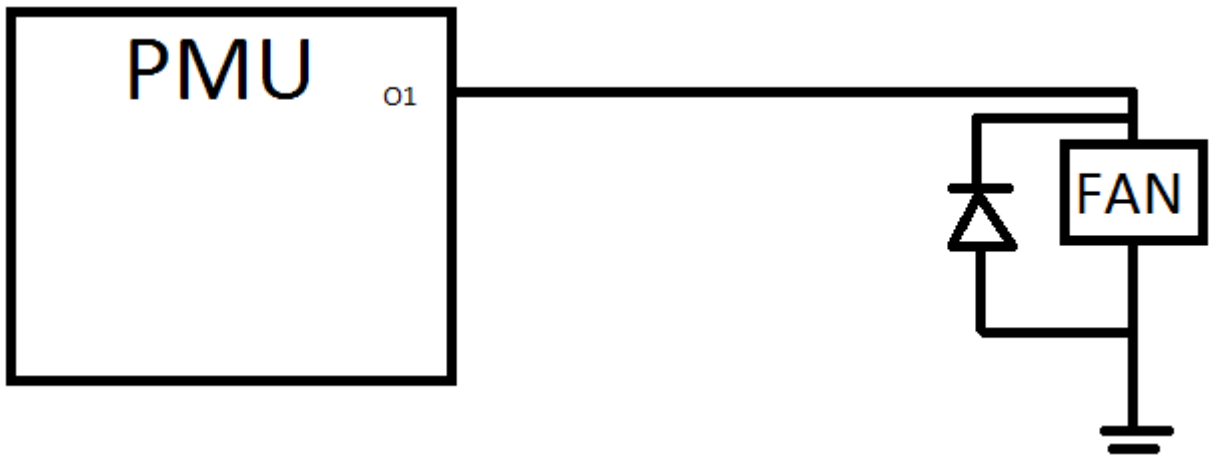
PMU Client Configuration Example:



PWM Configuration example

Using flyback diode

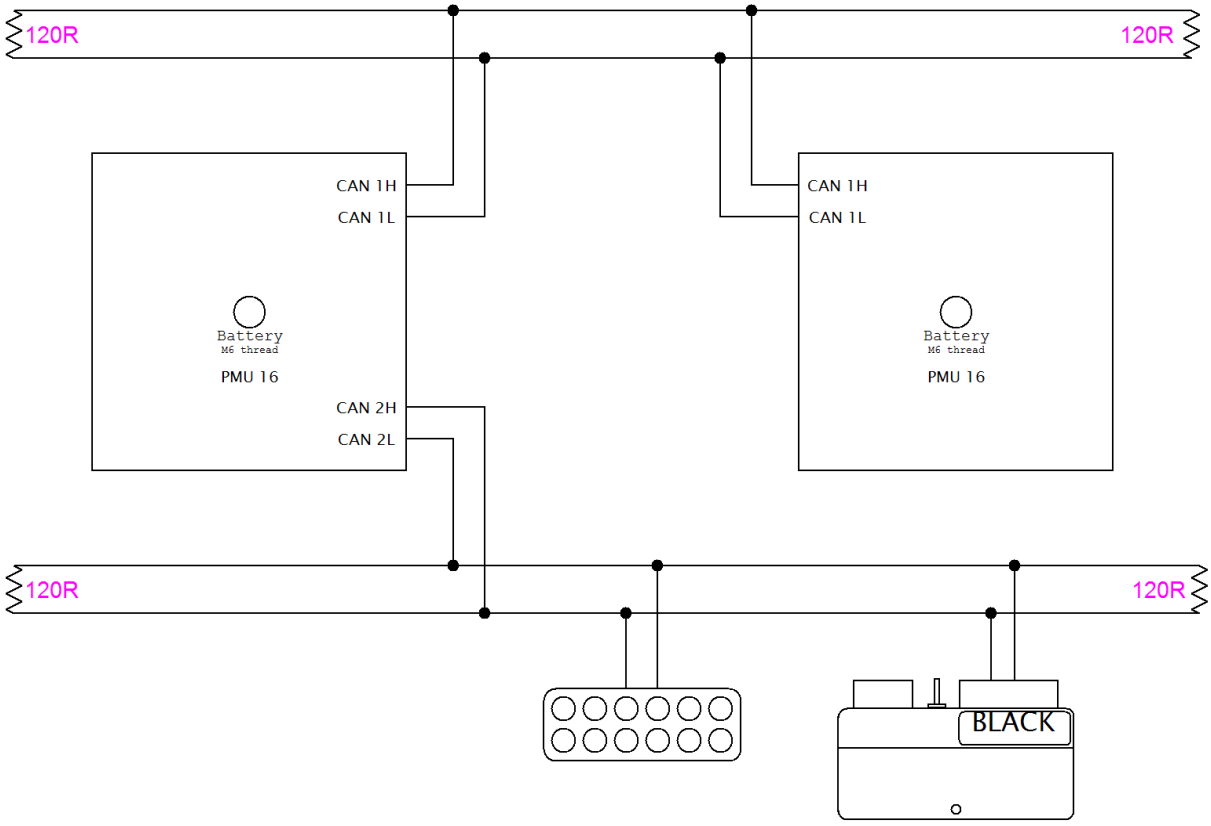
PWM does introduce a bit of heat depending on the Device powered, and PWM settings. If you are experiencing *Overheated* status on any Pulse Width Modulated Power Output, a flyback diode can be used to eliminate flyback therefore lowering the output transistors thermal load.



Flyback diode diagram

USING MULTIPLE PMUs

Basic Diagram



CAN communication diagram

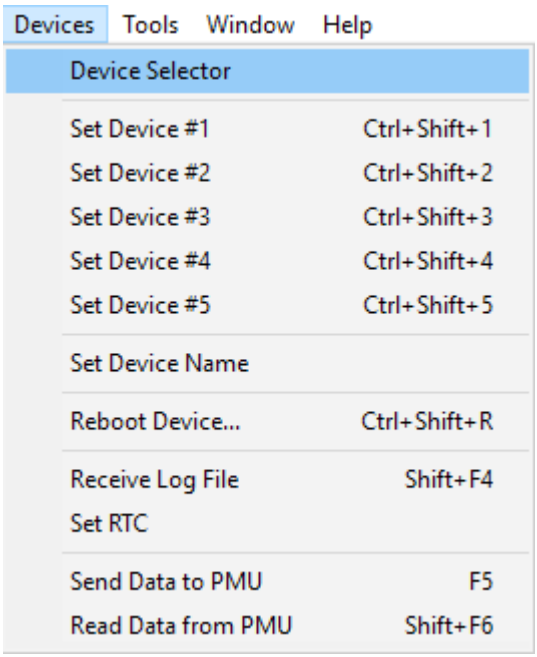
To use PMUs in tandem, it is recommended to use CANbus 1. Both devices must be connected to the same bus, terminated at both ends with 120 Ohm resistors. Up to five PMUs can be used at once. For more informations regarding CAN bus see *CAN Topology* section.

Communication

There is no need for a special configuration per se, but for this setup to work optimally it is advised to make use of CANbus Export and CANbus Input functionality to transmit important data between PMUs. It is vital for those two PMUs to communicate with each other, so that proper strategies can be set up. See *CAN Stream* section to find out more about CANbus communication.

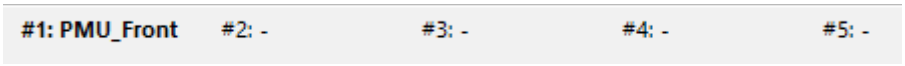
Using PMU Client with multiple PMUs

PMU Client allows up to 5 PMUs to be connected simultaneously. You can easily switch between them either using *Menu Bar → Devices → Set Device #X* or using *Ctrl + Shift + X* keyboard shortcut, where X is the device Number.



Switching between connected PMUs

All currently connected PMUs are also displayed as a list.



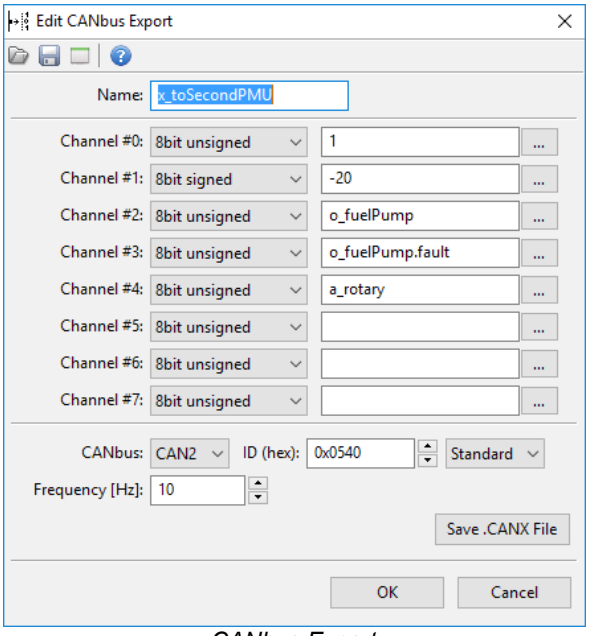
List of connected PMUs

CAN STREAM

CANbus Export

CANbus Export allows user to broadcast various informations on chosen CANbus. It is essential tool to communicate with other PMUs or ECUMASTER EMU Black device.
CANbus Export can broadcast channel states, values, voltage readings, which then can be read using CANbus Input on second PMU or EMU Black. PMU Client also allows you to save your CANbus Export as a *.CANX* file. This file can be imported by any other PMU.

CANbus Export Example:



CANbus Export

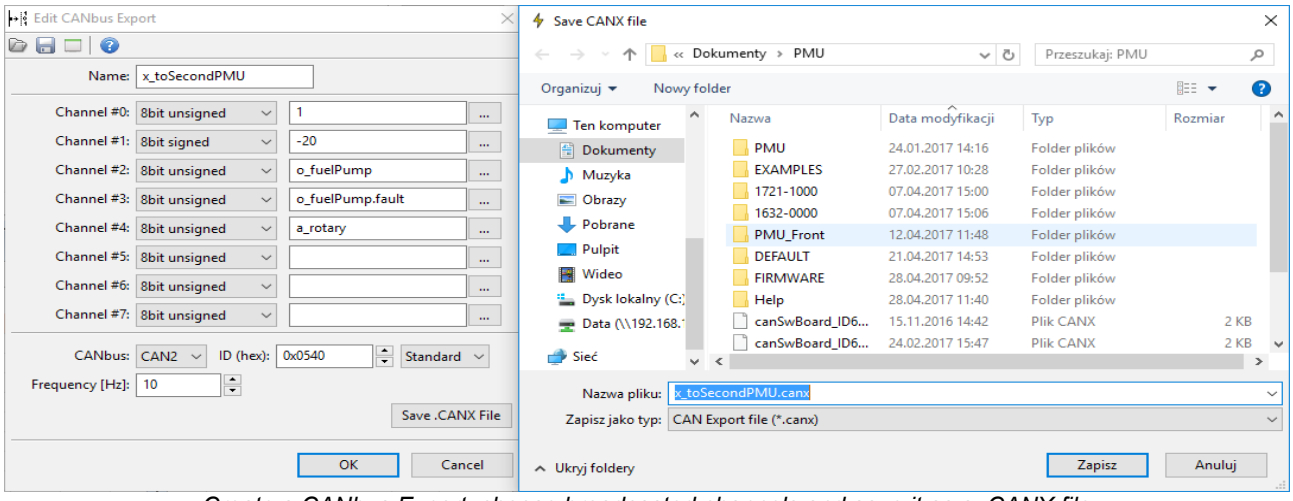
CANbus Input

CANbus Input allows user to read data from CANbus stream using CANbus Mob (See *CANbus Message Object* section). You can choose the starting frame, how many bytes to read, how many bits to read and the endianess (byte order) of the data read. If a CANbus Export was created on secondary PMU, CANbus Input (with correct *CANbus Message Object*) can be used to read exported parameters.

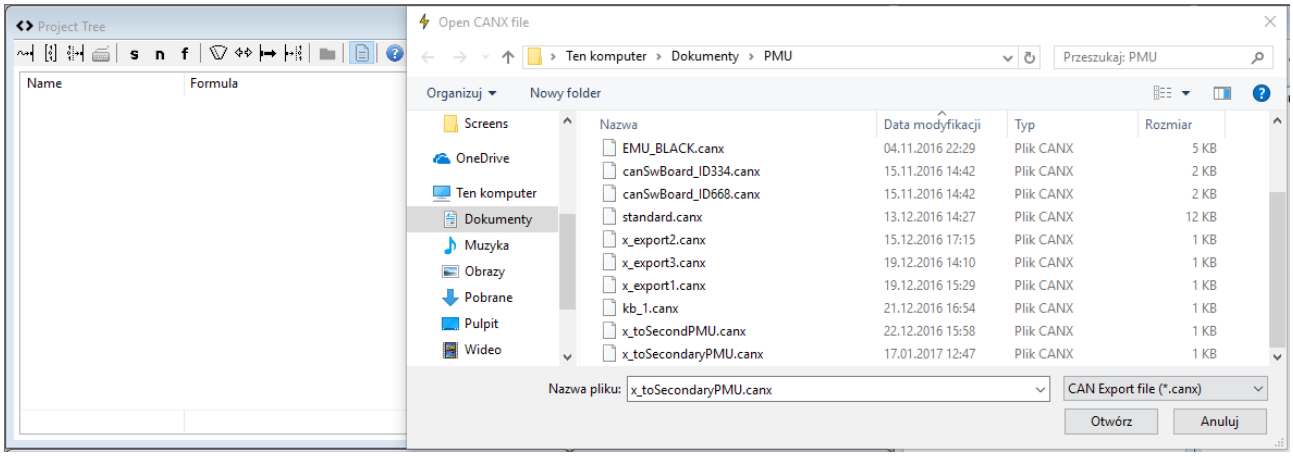
Importing .CANX File

When creating a CANbus Export, you have the option to save this Export as a .CANX file.
This file can be imported by other PMU to automatically create CANbus Mob with correct ID, Mob name and CANbus Inputs that correspond to exported Channel.

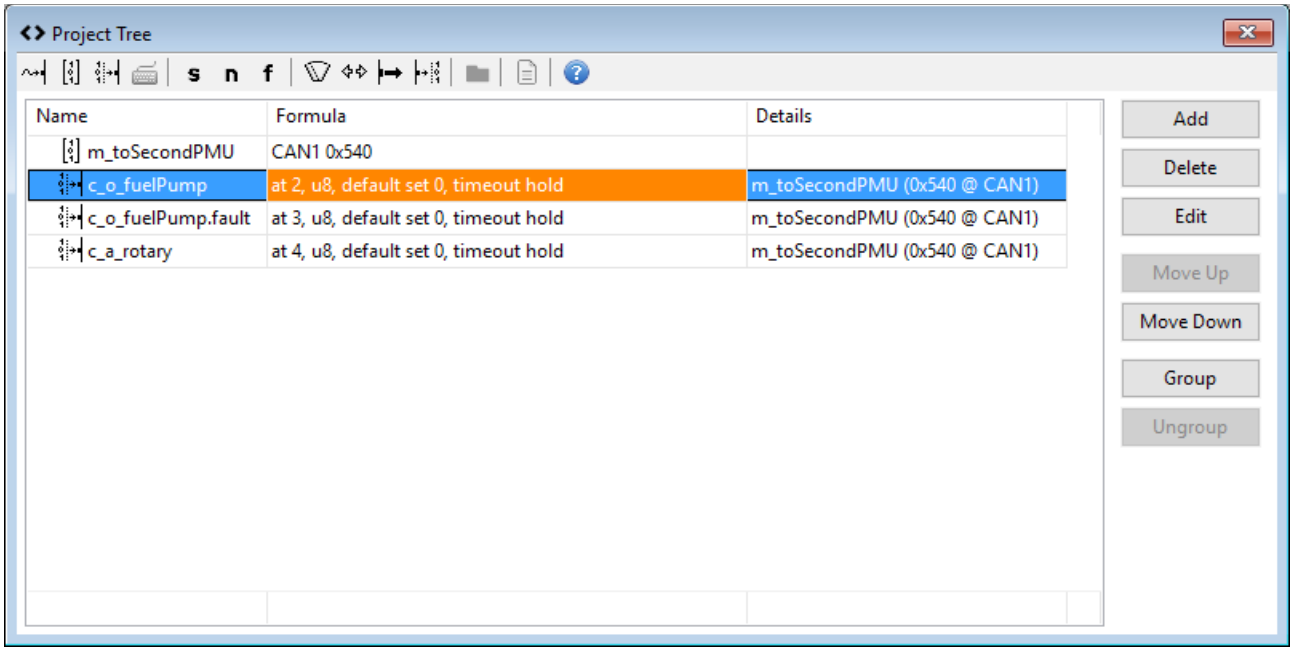
Example:



Create a CANbus Export, choose broadcasted channels and save it as a .CANX file



Use Import .CANX option from Project Tree to open the .CANX file



As we can see, a CANbus Mob and CANbus Inputs were automatically created with correct ID

Standard CAN Stream

Standard CAN Stream *(Tree View → Standard CAN Stream)* gives user the ability to broadcast key PMU parameters over CAN bus to have an oversight.
Parameters themselves are predefined, but user has the ability to broadcast only some part of them, on chosen CAN bus with chosen ID.

Here is how CAN Stream frames are constructed:

ID	BaseID + 0			Frequency: 20Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	(reserved)	4 bits (0xF0)	Unsigned	0-15	-	0	-
	User error	1 bit (0x08)	Unsigned	0-1	-	0	-
	PMU Status [1])	3 bits (0x07)	Unsigned	0-7	-	0	-
1	Total Current	8 bits	Unsigned	0-255	1A/bit	0	A
2	Battery Voltage	8 bits	Unsigned	0-27.75	0.1088V/bit	0	V
3	Board Temperature Left	8 bits	Unsigned	0-255	1C/bit	0	C
4	Board Temperature Right	8 bits	Unsigned	0-255	1C/bit	0	C
5	Flash Temperature	8 bits	Unsigned	0-255	1C/bit	0	C
6	(reserved)	2 bits (0xC0)	Unsigned	0-3	-	0	-
	I6.active [2]) I5.active I4.active I3.active I2.active I1.active	1 bit (0x20) 1 bit (0x10) 1 bit (0x08) 1 bit (0x04) 1 bit (0x02) 1 bit (0x01)	Unsigned	0-1 0-1 0-1 0-1 0-1 0-1	-	0	-
7	(reserved)	2 bits (0xC0)	Unsigned	0-3	-	0	-
	I6.error [2]) I5.error I4.error I3.error I2.error I1.error	1 bit (0x20) 1 bit (0x10) 1 bit (0x08) 1 bit (0x04) 1 bit (0x02) 1 bit (0x01)	Unsigned	0-1 0-1 0-1 0-1 0-1 0-1	-	0	-

[1]) The PMU Status is just the maximum value of all of the 16 outputs. This pmu.status show the status of output in the “worst”condition.

[2]) The state of the low side outputs.

ID	BaseID + 1			Frequency: 20Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	o1.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o1.active	1 bit (0x10)		0-1			
	o2.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o2.active	1 bit (0x01)		0-1			
1	o3.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o3.active	1 bit (0x10)		0-1			
	o4.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o4.active	1 bit (0x01)		0-1			
2	o5.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o5.active	1 bit (0x10)		0-1			
	o6.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o6.active	1 bit (0x01)		0-1			
3	o7.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o7.active	1 bit (0x10)		0-1			
	o8.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o8.active	1 bit (0x01)		0-1			
4	o9.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o9.active	1 bit (0x10)		0-1			
	o10.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o10.active	1 bit (0x01)		0-1			
5	o11.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o11.active	1 bit (0x10)		0-1			
	o12.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o12.active	1 bit (0x01)		0-1			
6	o13.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o13.active	1 bit (0x10)		0-1			
	o14.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o14.active	1 bit (0x01)		0-1			
7	o15.status [3])	3 bits (0xE0)	Unsigned	0-7	-	0	-
	o15.active	1 bit (0x10)		0-1			
	o16.status [3])	3 bits (0x0E)	Unsigned	0-7	-	0	-
	o16.active	1 bit (0x01)		0-1			

[3]) All possible values of each o*.status: 0 -OFF, 1 – ACTIVE (ON), 2 – UNDERCURRENT, 3 – OVERCURRENT, 7 – THERMAL SHUTDOWN. More information is in the section 'Output Pin signaling and status.'

ID	BaseID + 2			Frequency: 62.5Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	a1.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
1	a2.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
2	a3.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
3	a4.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
4	a5.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
5	a6.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
6	a7.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
7	a8.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V

ID	BaseID + 3			Frequency: 62.5Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	a9.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
1	a10.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
2	a11.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
3	a12.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
4	a13.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
5	a14.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
6	a15.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V
7	a16.voltage	8 bits	Unsigned	0-5	0.0196V/bit	0	V

ID	BaseID + 4			Frequency: 20Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	o1.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
1	o2.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
2	o3.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
3	o4.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
4	o5.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
5	o6.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
6	o7.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
7	o8.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A

ID	BaseID + 5			Frequency: 20Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	o9.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
1	o10.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
2	o11.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
3	o12.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
4	o13.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
5	o14.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
6	o15.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A
7	o16.current	8 bits	Unsigned	0-63.75	0.25A/bit	0	A

ID	BaseID + 6			Frequency: 20Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	o1.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
1	o2.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
2	o3.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
3	o4.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
4	o5.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
5	o6.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
6	o7.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
7	o8.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V

ID	BaseID + 7			Frequency: 20Hz			
ByteID	Channel	Data Width	Data Type	Range	Resolution	Offset	Unit
0	o9.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
1	o10.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
2	o11.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
3	o12.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
4	o13.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
5	o14.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
6	o15.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V
7	o16.voltage	8 bits	Unsigned	0-16.19	0.063V/bit	0	V

DOCUMENT REVISION HISTORY

Revision	Date	Changes
1.00	2017.05.01	- initial release
1.01	2017.07.10	- added information about ground connection with USBtoCAN
1.02	2018.06.04	- clarified outputs specification
1.03	2020.09.02	- terminal current rating and derating
1.04	2022.09.05	- Standard CAN Stream: added states of low side outputs in PMU AS - Standard CAN Stream: frequencies have been corrected to match the actual device values - Standard CAN Stream: PMU Status and status of each outputs explained

Output Number	Terminal	Max Current	Wire Size	Current	Name	Comment
O1	38	25A				
O2	39	25A				
O3	26	25A				
O4	13	25A				
O5	12	25A				
O6	11	15A				
O7	10	15A				
O8	9	15A				(wipers)
O9	5	15A				
O10	4	15A				
O11	3	15A				
O12	2	25A				
O13	1	25A				
O14	14	25A				
O15	27	25A				
O16	28	25A				

Analog Number	Terminal	Type	Pullup/ Pulldown	Name	Comment
A1	29				
A2	16				
A3	30				
A4	17				
A5	31				
A6	18				
A7	32				
A8	19				
A9	6				
A10	33				
A11	20				
A12	34				
A13	21				
A14	8				
A15	35				
A16	22				