

TECH NOTE

Configuring CAN BUS communication between EMU BLACK or EMU CAN and PMU16



Revision 1.0



Introduction

EMU output control (eg. coolant fan control fuel pump, start/stop, etc) may be done through the PMU using CAN BUS, simplifying the vehicle electrical system.

Requirements

For EMU BLACK firmware 2.045 or higher is required. For EMU firmware 1.188 or higher is required For PMU firmware 0.0265.0 or higher is required

Connection diagram

First, connect the EMU and PMU via CAN BUS. Here is an example of that connection.



Next, set the correct CAN BUS speed. In this example we use 500kbps and termination on both devices. Remember to use twisted pair cables to connect any CAN BUS devices!

For EMU BLACK the configuration of CAN should look like the following:

🔲 CAN, Serial - CAN	_ 🗆 🗙
De 🔒 🗆 🛛	
CAN	
CAN-Bus speed	500 Kbps
Enable terminator 1200hm	
Send EMU stream over CAN-Bus	
EMU strem base ID(HEX)	600
Send data to BTCAN module	
CAN-Bus dashboard	None



It is critical that you enable "Send EMU stream over CAN-Bus". This activates the EMU standard CAN stream with critical ECU parameters like RPM, CLT, state of coolant fan, fuel pump, etc. More information about the EMU standard stream can be found in help menus of the EMU software. Another important parameter is EMU stream base ID. In this example the value is 0x600 and is the default CAN stream ID.

On the PMU side, CAN#2 input must be configured

×
500 Kbps

The next step is to import the *canx* file into the PMU by pressing the indicated icon.



Next select EMU_BLACK.canx file (for both classic EMU and EMU BLACK)

A selection window will appear:

Import CANX File
Name: m_emublack
CANbus: CAN1
Base ID (hex): 0x0600
Select Channels:
☐ 🔽 frame +0
▼ c_ecu_rpm
▼ c_ecu_tps
▼ c_ecu_iat
▼ c_ecu_map
✓ c_ecu_injPW
Frame +1
✓ c_ecu_aIn1
▼ c_ecu_aIn2
▼ c_ecu_aIn3
✓ c_ecu_aIn4
Frame +2
▼ c_ecu_vSpd
☑ c_ecu_baro
√ c_eαu_oiT
√ c_eαu_oiP
▼ c_ecu_fuelP
₹ c_eαu_dt
☐ 🗹 frame +3
Select All Select None
OK Cancel

First select the proper CAN BUS input (CAN2) Base ID must be the same as Base ID in EMU Black (in our example it is 0x600). Next mark the channels that you want to use in the PMU. *In our example we select c_ecu_rpm, c_ecu_fuelPumpSt, and*

in our example we select c_ecu_rpm, c_ecu_tuelPumpSt, al c_ecu_coolantFanSt.



The project tree should look like this:

<	Project Tree				
^	서 🕅 🚧 🚎 snf 🖾 🗇 🚧 🖿 📄 📄 😨				
	Name	Formula	Details		Add
1	[‡] m_emublack	CAN2 0x600 - 8 frames			Delete
	tite c_ecu_rpm	at 0, u16 le, default set 0, timeout hold	m_emublack +0 (0x600 @ CAN2)		Delete
	<pre>↔ c_ecu_fuelPumpSt</pre>	at 7, u8, extract(c1, p0), default set 0, timeout hold	m_emublack +6 (0x606 @ CAN2)		Edit
	<pre>↔ c_ecu_coolantFanSt</pre>	at 7, u8, extract(c1, p1), default set 0, timeout hold	m_emublack +6 (0x606 @ CAN2)		
H					Move Up
l					Move Down
l					Group
l					Ungroup

You can use Group to make the project more clear. Select all elements and press Group button. Then set the group name. In our example it is EMU CAN INPUT

\diamond	Project Tree			×
~**	🚺 🚧 🚎 🖌 s n 🕆	f │ \[\] 🗇 🚧 ┝╬ │ 🖿 │ 📄 │ 😨		
Na	me	Formula	Details	Add
Ξ	EMU CAN INPUT			Delete
	🕴 m_emublack	CAN2 0x600 - 8 frames		Delete
	+ c_ecu_rpm	at 0, u16 le, default set 0, timeout hold	m_emublack +0 (0x600 @ CAN2)	Edit
	+ c_ecu_fuelPumpSt	at 7, u8, extract(c1, p0), default set 0, timeout hold	m_emublack +6 (0x606 @ CAN2)	
	+ c_ecu_coolantFanSt	at 7, u8, extract(c1, p1), default set 0, timeout hold	m_emublack +6 (0x606 @ CAN2)	Move Up
				Move Down
				Group
				Ungroup

To use CAN output in EMU BLACK you need to select CAN BUS output for the strategy. For example fuel pump control:

🔲 Outputs - Fuel pump	<u>- 🗆 ×</u>
🗁 🗖 🗖 🗿	
Fuel pump	
After start activity	2 s
Output	CAN-Bus
Invert output	



Now you can use data from EMU ECU.

Activation of fuel pump based on EMU request:

First add power output by clicking icon on the toolbar

Project Tree			×
~+ [] [+ s n f ♡ +> ++ + ■] ②			
Name Formula	Power Output	Details	Add
🖃 🖿 EMU CAN INPUT			Delete
] m_emublack CAN2 0x600	rames		0.01000
c_ecu_rpm at 0, u16 le, c	ult set 0, timeout hold	m_emublack +0 (0x600 @ CAN2)	Edit
c_ecu_fuelPumpSt at 7, u8, extr	c1, p0), default set 0, timeout hold	m_emublack +6 (0x606 @ CAN2)	
c_ecu_coolantFanSt at 7, u8, extr	c1, p1), default set 0, timeout hold	m_emublack +6 (0x606 @ CAN2)	Move Up
			Move Down
			Group
			Ungroup

The configuration window:

⊨ New Power Output
Name: o_fuelPump
Pin: single V 01 (25A) V
Inrush Current [A]: 80,0 Max Current [A]: 15,0 Min Current [A]: 0,0
Retry Count: 3 Retry Every [s]: 1,00
<u>P</u> WM Configuration
O Default: ☑ On/Off
Channel: c_ecu_fuelPumpSt
C Eormula: <more></more>
OK Cancel

Set output name and configure the currents. As you can see in the picture above, Channel *c_ecu_fuelPumpSt* is used for output activation. Press F4 or click the button to select the channel from a list.



Here is an example of a shift light output that will activate when engine RPM are above 9000.

Hew Power Out	put	×
D 🗐 🗆 🛛		
<u>N</u> ame:	o_revLimiterLight	
Pin:	single 💌 O6 (15A) 💌	
Inrush Current [A]: Max Current [A]: Min Current [A]:	10,0 Thrush Time [s]: 1,00 5,0 0,0 T	
 Retry Count: Retry Forever 	3 * Retry Every [s]: 1,00 *	
PWM Configura	ation	
C Default:	☑ _On/Off	
C Channel:		
€ Eormula:	value = c_ecu_rpm > 9000 and or 	Add Delete Edit Move Up Move Dgwn Delete All
	ОК	Cancel

In this case the following formula is used: *c_ecu_rpm* > 9000