

ECUMASTER ADU

Application Note



MAXXECU

Revision 1.0

1. Copyright and trademarks

All trademarks, service marks, trade names, product names and logos appearing in this document are the property of their respective owners.

2. Introduction

This application note explains how to connect and configure the MAXXECU series with the ECUMASTER ADU.

3. Electrical connection

The MAXXECU is able to send the standard log stream over the CAN BUS. There are two options for connection. If the MAXXECU CAN1 speed is set to 1Mbps then you may use ADU CAN1 or CAN2. If the MAXXECU CAN1 speed is set to 500kbps then you may only use ADU CAN2.

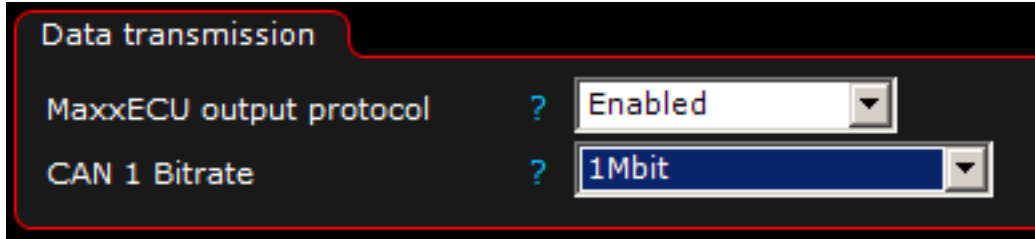
MAXXECU terminal	ADU CAN1	ADU CAN2	Comment
E2	4	6	CAN L
E3	5	7	CAN H

Twisted pair cable is required for any CAN BUS connection.

Ensure that the CAN BUS is properly terminated.

4. ADU and MAXECU configuration

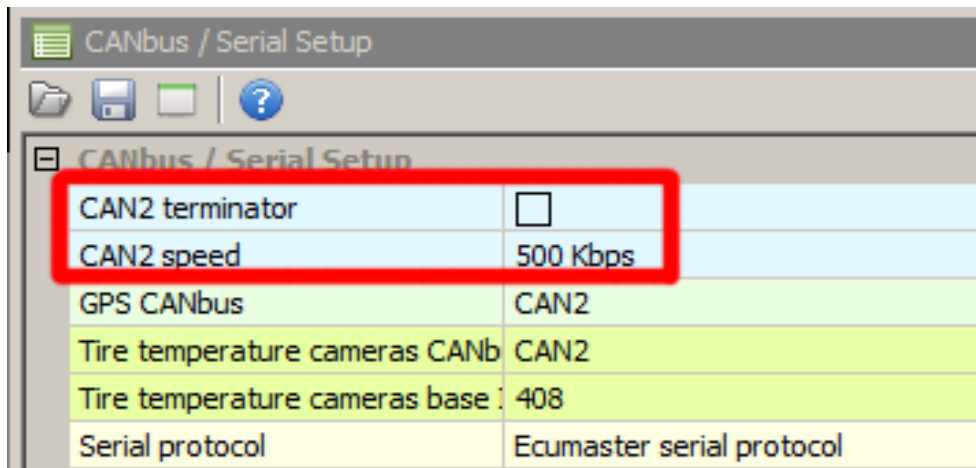
The first step is to enable the MAXXECU CAN output protocol. The option is available in “*Configuration / CAN settings / Data transmission*”.



It is important to select the proper CAN BUS speed. If you choose to connect ECU to CAN1 or CAN2 at 1Mbps you must select 1Mbit speed. If you choose to connect to CAN2 with 500kbps speed, you must to select 500kbit.

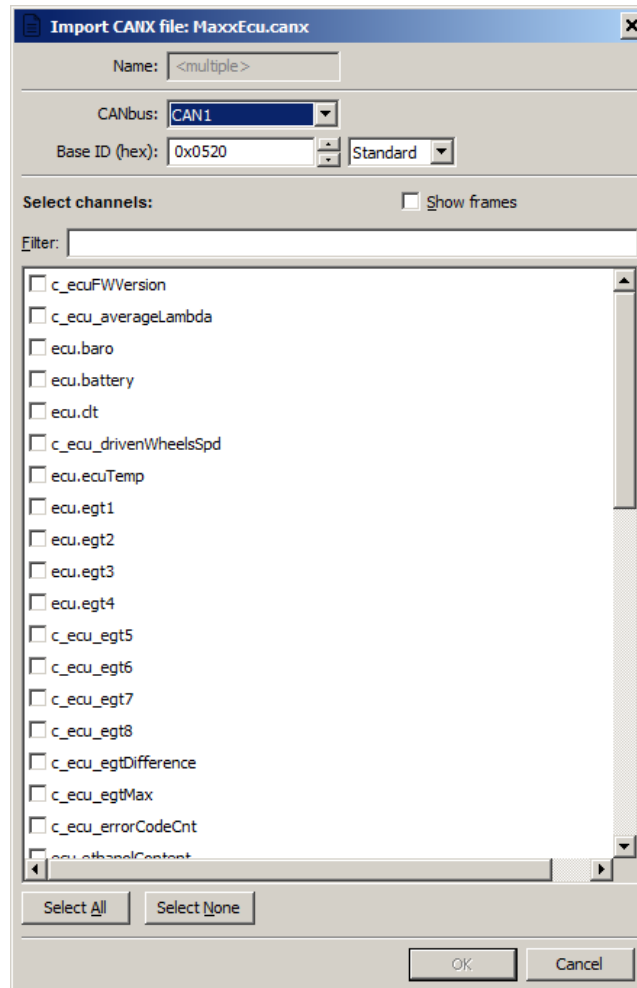
If you use ADU CAN1, the speed is fixed at 1Mbps and no CAN configuration is required. If you choose to connect MAXXECU to CAN2, you must set proper CAN BUS speed and termination.

To open CAN2 configuration, press F9 to show the pane selector. Then open “*General / CAN BUS Serial setup*”.

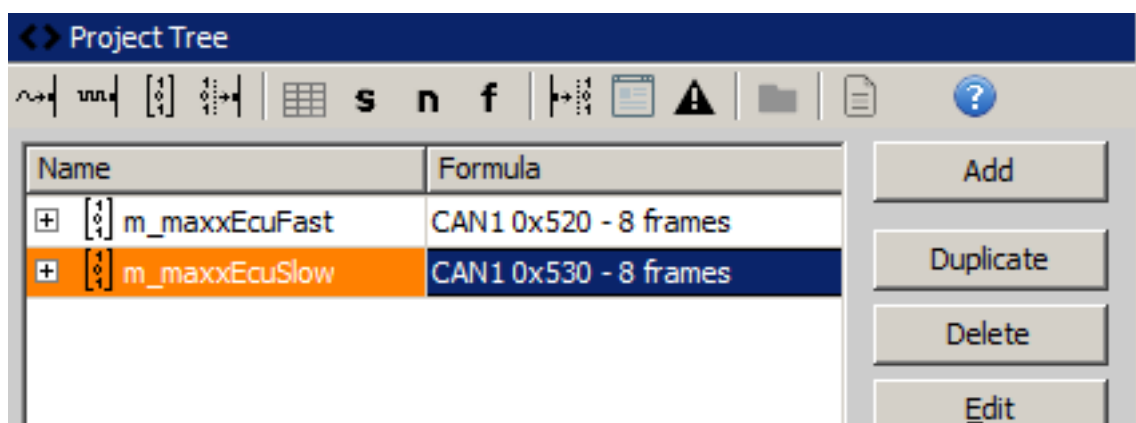


The next step is to load the CANX file with MAXXECU channel definitions.

On the Project tree, click the “Add” button and select “Import .CANX file”. When the file dialog opens, select the “MaxxEcu.canx” file. The following dialog appears:



At this point, select the CAN BUS that will be used for communication (CAN1 or CAN2) and the channels you want to read. In most situations all channels should be loaded (Select All). The project tree should look like the following:



If you open “m_maxxEcuFast” or “m_maxxEcuSlow” mobs, all available CAN inputs should be visible.

5. Supported channels

ADU channel	Description
ecu.baro	Barometric pressure
ecu.battery	Battery voltage
ecu.clt	Engine coolant temperature
ecu.ecuTemp	ECU internal temperature
ecu.egt1	Exhaust gases temperature 1
ecu.egt2	Exhaust gases temperature 2
ecu.egt3	Exhaust gases temperature 3
ecu.egt4	Exhaust gases temperature 4
ecu.ethanolContent	Fuel ethanol content
ecu.gear	Current gear
ecu.iat	Intake manifold temperature
ecu.ignAngle	Ignition advance
ecu.injDC	Injectors DC
ecu.injPW	Injectors pulse width
ecu.lambda1	Lambda from oxygen sensor #1
ecu.lambda1Trim	Current lambda #1 fuel trim
ecu.lambda2	Lambda from oxygen sensor #2
ecu.lambda2Trim	Current lambda #2 fuel trim
ecu.map	Manifold absolute pressure
ecu.rpm	Engine RPM
ecu.speed	Vehicle speed
ecu.tps	Throttle position sensor
c_ecuFWVersion	ECU firmware version
c_ecu_averageLambda	Average lambda value from sensor #1 and #2
c_ecu_drivenWheelsSpd	Speed of driven wheels
c_ecu_egt5	Exhaust gases temperature 5
c_ecu_egt6	Exhaust gases temperature 6
c_ecu_egt7	Exhaust gases temperature 7
c_ecu_egt8	Exhaust gases temperature 8
c_ecu_egtDifference	Exhaust gases difference between the lowest and the highest temperature
c_ecu_egtMax	Maximum exhaust gases temperature
c_ecu_errorCodeCnt	The number of active engine error codes
c_ecu_fuelCut	Fuel cut percent

c_ecu_fuelTrimTotal	Value of total fuel trim
c_ecu_ignCompTotal	Value of ignition advance correction
c_ecu_ignCut	Ignition cut percent
c_ecu_syncLostCnt	The number of trigger sync lost
c_ecu_tcPowerLimit	Traction control power limit
c_ecu_undrivenWheelsSpd	Speed of un-driven wheels
c_ecu_userAin1	Value of user analog in #1
c_ecu_userAin2	Value of user analog in #2
c_ecu_userAin3	Value of user analog in #3
c_ecu_userAin4	Value of user analog in #4
c_ecu_wheelSlip	Wheel sleep percent
c_ecu_wheelSlipTrgt	Wheel sleep target percent