MICRO LAMBDA

TurboCorse.at

Electronics

Table of contents

Functions of the lambda controller

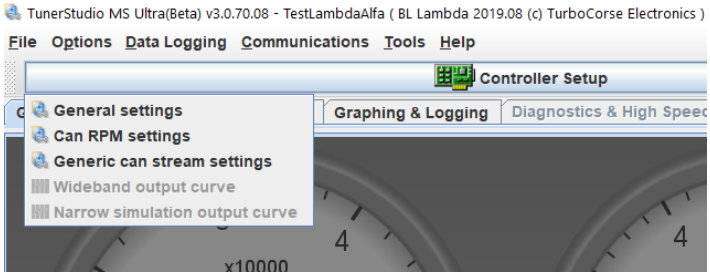
The Micro Lambda controller is based on BL49 Hardware by tooly and
designed to work with a LSU 4.9 lambda sensor.
Sensor interface is internally done by the well-known CJ125 LSU chip from BOSCH.

- Sensor interface CJ125 from BOSCH (No free air calibration)
- 16MHz Automotive 8bit Microcontroller with can bus integrated
- PID Heat strategy according to BOSCH recommendation
- 2x Analog output (0-5V Wideband / Narrowband simulation, curve can be changed)
- 1x Digital input (Enable signal for the controller)
- 1x Can bus with selectable termination resistor
- All in and outputs are short circuit protected
- TunerStudio user interface
- Cycle status and cj125 status display
- All settings can be changed by TunerStudio
- Can bus broadcast with different modes
- Automatic start when receive RPM over can

Credit goes to:
Ruzki (Hardware design, assembly)
Tooly (Hardware design)
Alfagta (Firmware design, assembly)

Functions in detail

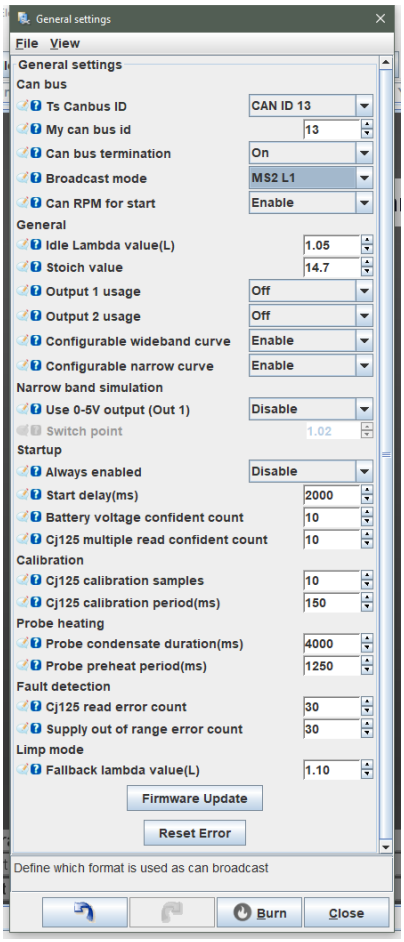
Controller setup



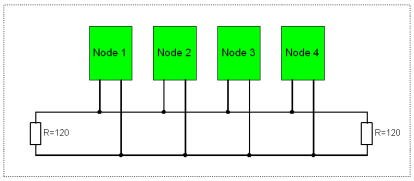
General settings:

Can bus:

Ts Can ID: Id which TS takes to communicate
My Can ID: Id for this controller
If you use more than one lambda controller in one can bus
Network each controller should have it's one ID.
Therefor connect one controller after the other and setup
unique id for each controller with setting "My Canbus ID"



Can termination: Turn on/off the can termination resistor.
A can bus network must be terminated on each end with 120Ohm.
If you use a Megasquirt 2/3 then the Megasquirt terminates one end.
One device in the can network must terminate the other end, usually the
Device which is on the far end.



Broadcast mode: Define the broadcast mode.
The controller is able to do different broadcast modes.
MS2 L1 → Megasquirt 2 as lambda sensor 1
MS2 L2 → Megasquirt 2 as lambda sensor 2
MS3 L1 → Megasquirt 3 as lambda sensor 1
MS3 L2 → Megasquirt 3 as lambda sensor 2
MS3 L3..6 → Megasquirt 3 as lambda sensor 3..6
AEM 4ch UEGO → as per AEM specification
AEM X UEGO → as per AEM specification
KMS → as per KMS specification
BL49 →
GENERIC → User can define ID, FORMAT, ENDIAN (later more)
MS2 ADC0..7
….more can be added in the future or per request

Can RPM for start: Turn on/off to listen for a rpm signal on the can bus
This will enable an additional menu to setup the parameters

General:

Idle Lambda value: If the controller is in idle mode this lambda value is used.
This is useful for ecu's where the user can't set a start delay for the lambda controller.
During idle, condensate, preheat this value is used then.

Stoich value: This is the stoich value for the fuel type you are using.

Fuel	A/F Ratio	Equivalence Ratio	Lambda
Gasoline Stoichiometric	14.7	1.000	1.000
Gasoline Max Power Rich	12.5	1.760	.8503
Gasoline Max Power Lean	13.23	1.111	.900
E85 Stoichiometric	9.765	1.000	1.000
E85 Max Power Rich	6.975	1.400	.7143
E85 Max Power Lean	8.469	1.153	.8673

Output 1/2 usage: Turn on/off the output
If you send the lambda value over can and you don't need any of the hardware analog outputs you can turn these off with this setting. If analog output is off the voltage is 0.
So turn off the analog outputs which you don't need.

Analog output 1 is normally for wideband signal, this is a real DAC with 10 bit.
Analog output 2 is normally for narrow simulation signal, this is a PWM DAC.
You can use the standard curve or set it up as you want.

Configurable wideband curve: Turn on/off the user configurable curve
Configurable narrow curve: Turn on/off the user configurable curve
With this you can turn off the 2d tables to configure the analog output as you want.
Standard wideband curve is lambda 0.68 = 0V / 1.36 = 5V
Standard narrow curve as per narrow sensor specification.

Narrow band simulation:

Use 0-5V output (Out 1): Turn on/off to use the real DAC for narrow simulation
This can be used when you send lambda value over can and feed narrow simulation to an oem ecu.
Then you can use the real dac to get better resolution.

Switch point: This is only selectable if you don't use the configurable narrow curve
Define the switch point when the narrow simulation goes to rich / lean, normally lambda 1.0

Startup:

Always enabled: With this setting you can force the controller to start directly when power is applied.
When this is setting is enabled the controller starts directly when power is applied and everything else is okay.
(Voltage in range, calibration done, no errors)
This is useful during setup the ecu, lambda controller is ready before you start so you can monitor the lambda value during warmup.
This can also be used if the controller is powered through the fuel pump relay and you don't want to use the enable input or can rpm to start the controller.

Start delay (ms): The controller wait this time before start after the enable signal is applied.
Setup this time high enough to prevent miss start when the controller is powered through the fuel pump relay which is maybe triggered for a short time during priming pulse.

Battery voltage confident count: The battery voltage must be for this read counts in range before controller go to calibration mode. Normally the default value of 10 is good to go.

Cj125 multiple read confident count: The communication to the cj125 must be good for these counts before controller go to calibration mode. Normally the default value of 10 is good to go.

Calibration:

Cj125 calibration samples: The controller read the default lambda and heater adc for this counts.
Normally the default value of 10 is good to go.

Cj125 calibration period (ms): The controller read the adc values from the cj125 with this period during calibration mode. Normally the default value of 150 is good to go.

Probe heating:

Probe condensation duration (ms): The controller stays for this time in the condensation mode with reduced heat power applied as per bosch lsu4.9 specification. Normally the default value of 5000 is good to go.

Probe preheat period (ms): The controller step time during preheat. As per bosch lsu4.9 specification the heat power should be ramped up during preheat, voltage is then raised with 0.4V steps and this period.
Normally the default value of 1500 is good to go.

Fault detection:

Cj125 read error count: The controller go to error mode if he reads this count of messages in a row from the cj125 with error flags. Normally the default value of 30 is good to go. (30x16 = 480ms debounce)

Supply out of range count: The controller go to error mode if the supply voltage is out of range for this count in a row. Normally the default value of 30 is good to go. (30x16 = 480ms debounce)

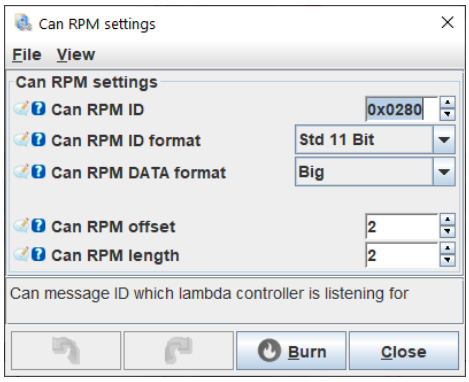
Limp mode:

Fallback lambda value: The controller use this value if an error occurs, this should give the ecu a lean lambda value to prevent engine failure do to a odd lambda value (rich). So for safety the ecu would add fuel instead of take fuel out.

Button:

Firmware update will force the controller to restart, Reset Error force the controller to restart after error without power reset

Can RPM settings:



Can RPM ID: Can bus ID to listen for rpm information.
The controller is listen for this id on the can bus.
If your ecu sends a message with this information you are able to use it to automatically start the controller if the engine is running.
MS2 / 3 can send this information simply turn on the broadcast with 0x280 RPM x1

Can RPM ID format: Can bus format standard 11 bit (2.0A) or Extended 29 bit (2.0B)

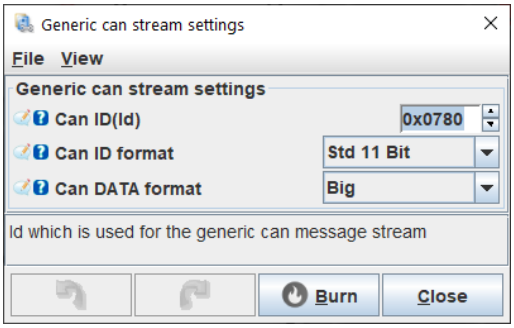
Can RPM DATA format: Endian within the message Big / Little. Big means high byte first.

Can RPM offset: Byte offset within the can message

Can RPM length: Byte length for the rpm info.

Generic can stream settings:

This is only available if broadcast mode is set to „GENERIC“



Can ID: Can ID for broadcast message.
The controller uses this id to send the broadcast message.
The message is defined like this:

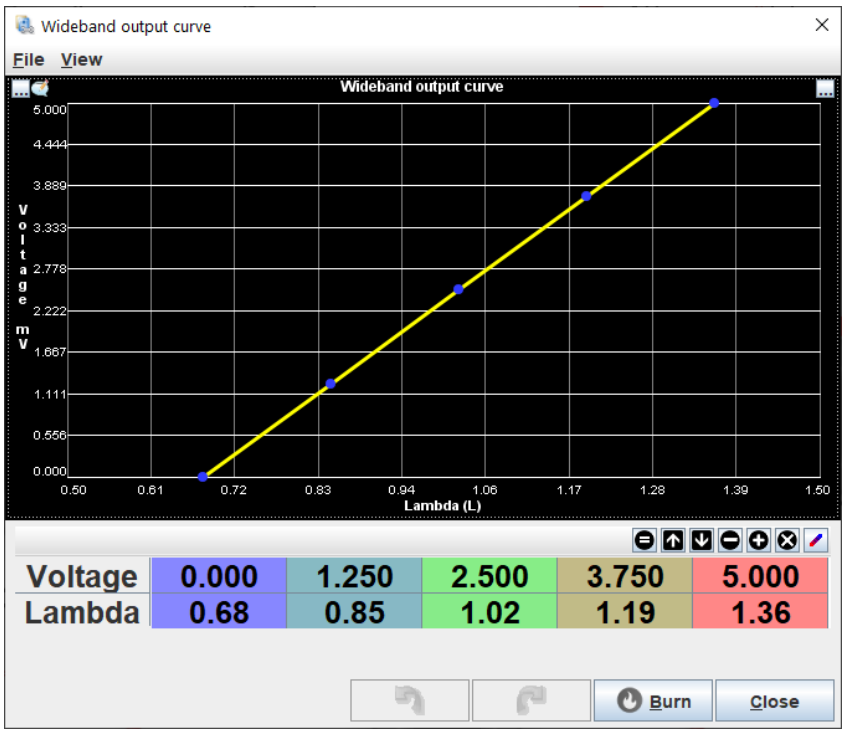
Byte [0] → Stoich value x10 (147 = 14.7)
Byte [1] →Lambda x100 (100 = 1.0)
Byte [2] →Controller mode
1= Preset
2= Start
3 = Calibration
4 = Idle
5 = Condensate
6 = Preheat
7 = Running
8 = Error
Byte[3] → Lambda x1000 (1000 = 1.0) (high/low byte as per format)
Byte[4] → Lambda x1000 (1000 = 1.0) (high/low byte as per format)

Can ID format: Can bus format standard 11 bit (2.0A) or Extended 29 bit (2.0B)

Can DATA format: Endian within the message Big / Little. Big means high byte first.

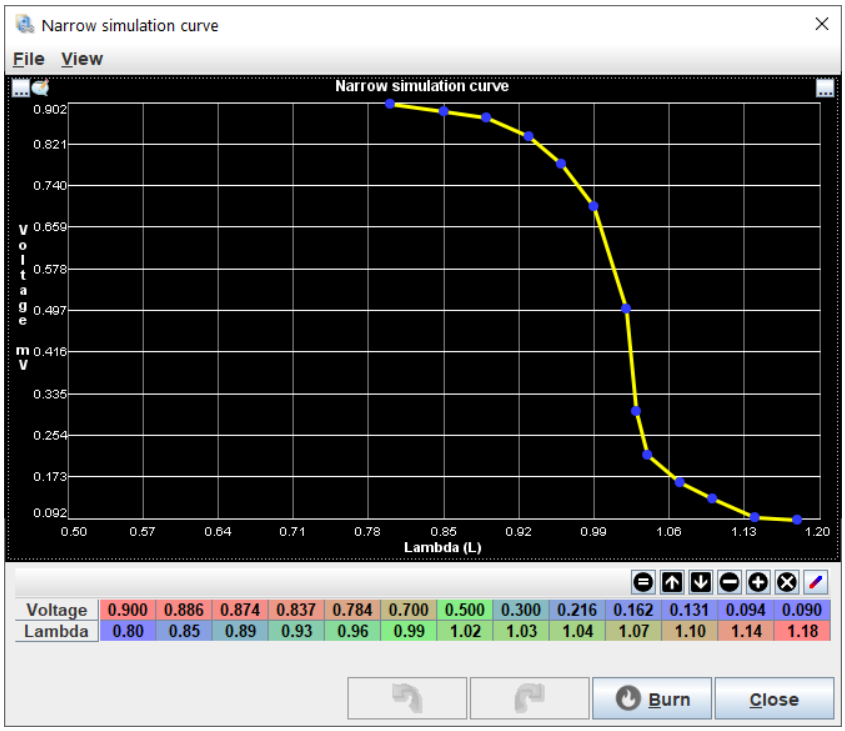
Wideband output curve:

This is only available if configurable wideband curve is „Enabled“
5 Point 2d table for wideband output

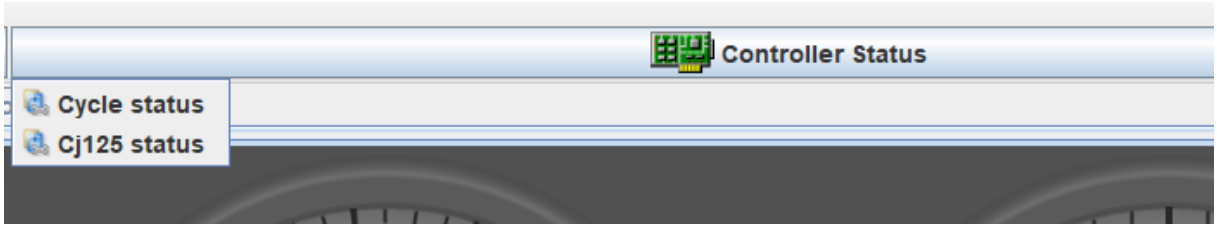


Narrow simulation output curve:

This is only available if configurable narrow curve is „Enabled“
13 Point 2d table for narrow simulation output

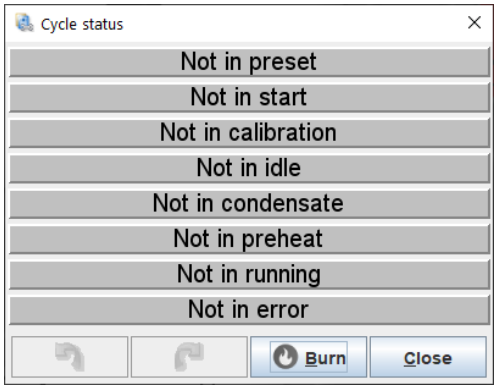


Controller status



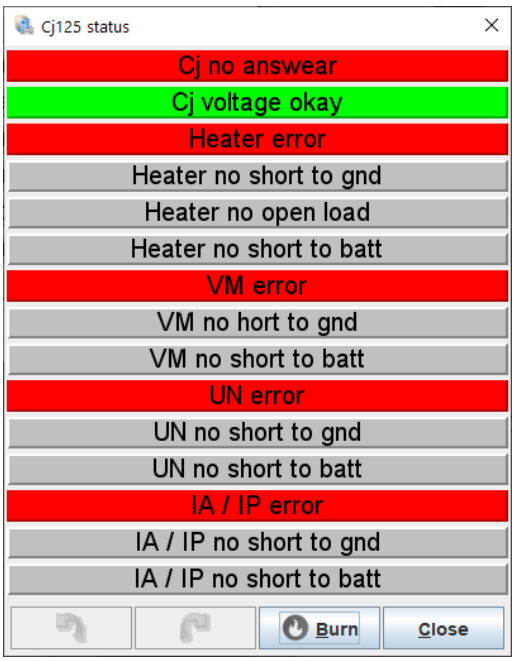
Cycle status:

Here you can see in which mode the controller is.
The actual mode is green highlighted



Cj125 status:

Here you can see the status from the cj125.



Wiring

The controller can be ordered in 2 different ways, open end or with connector.
When ordered with connector a Deutsch DT series 6way connected is attached.

Connector at controller harness: DT04-6P-C015 / Pins: 1060-16-0622
Connector at car harness: DT06-6S-C015 / Pins: 1062-16-0644

Power the controller from a switched power source (KL15). 10,8..16,5 Volt.
A maximum of 3A is needed per controller (3A Preheat cold sensor, ~0,7A Running)

When there is a connector attached to the controller only 6 wires are used.
The enable input wire is not available, so only 2 start modes are possible

- Always enabled , power the controller from the fuel pump relay to start when engine starts
  Always enabled parameter must be set to “Enable”
- Can RPM start, controller starts when rpm > 500 is received over can bus.
  Always enabled parameter must be set to “Disabled”

Take in mind for the other end of the connector the can bus wires must be twisted with a minimum of 1 twist per 25,4mm.

Connection	Function
1	+12V In (5A fused)
2	Can High
3	Out 1 (0-5V wideband signal) 0V = 0,68L / 10AFR | 5V = 1,36 L / 20AFR
4	Out 2 (0-1V narrow simulation signal)
5	Can Low
6	Gnd In

Table 1 controller with connector

Connection	Function
Red	+12V In (5A fused)
Black	Gnd In
Purple	Enable input, active when connected to gnd
Yellow	Out 1 (0-5V wideband signal) 0V = 0,68L / 10AFR | 5V = 1,36 L / 20AFR
Green	Out 2 (0-1V narrow simulation signal
Orange/Green	Can High
Orange/Brown	Can Low

Table 2 controller with open end

Can bus broadcast

Every 16ms a broadcast message stream is send to the can bus if enabled.
The message format is according CAN 2.0A 11 bit identifier.

Here you can see the layout of the broadcast messages.

B2U = Big Endian (Motorola Format), 2 Bytes, Unsigned
B2S = Big Endian (Motorola Format), 2 Bytes, Signed
1U = 1 Byte, Unsigned

ID (Hex)	Offset (Bytes)	Format	Comment
5C3 (BL49)	0	1U	Stoich value x10 (147 = 14.7)
	1	1U	Lambda value x100 (100 = 1.0)
	2	1U	Mode 1= Preset 2= Start 3 = Calibration 4 = Idle 5 = Condensate 6 = Preheat 7 = Running 8 = Error

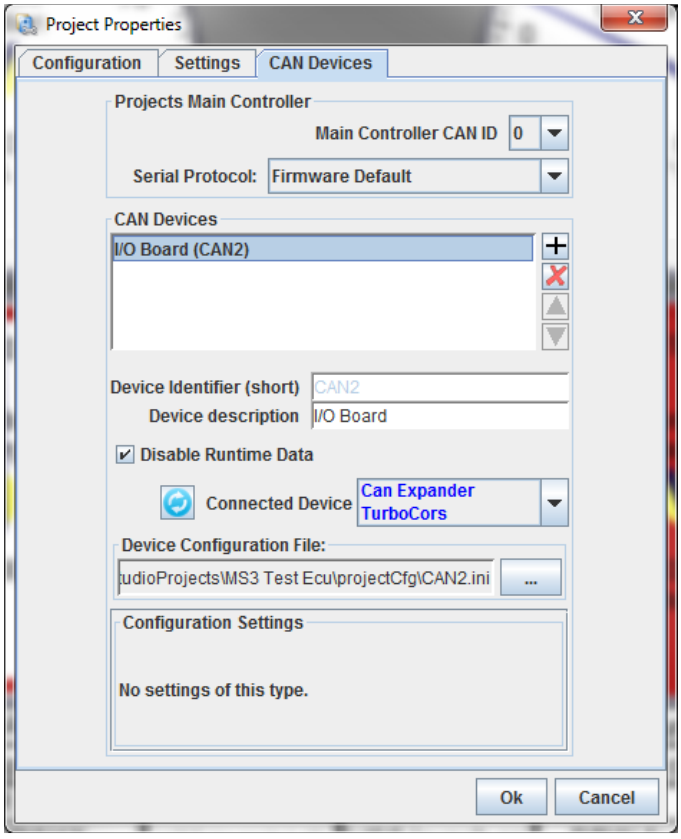
Table 3

Configuration to act as a can device within tuner studio

Connect the lambda controller as can device in TS to your existing Megasquirt 2/3 project.

Following steps are required:

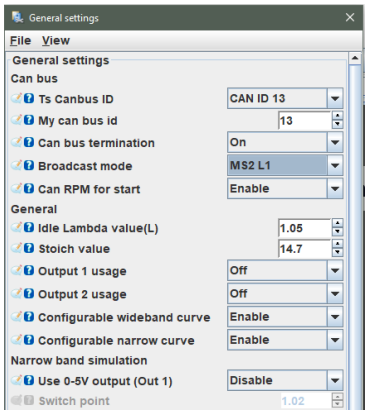
- Open your TS project.
- Project Settings. File→Vehicle Project→Project Properties, or by press CTRL + P.
- Menu „CAN Devices“.
- With button „+“you can add a new device.
- Fill in the device description.
- Under „Connected Device“ choose the can id. (Normally = 13)
- Under „Device Configuration File“ choose the ini file.
- Click „Ok“ to finish the settings.
- TS restarts the project then you have a drop down list on the left menu bar



Setup Megasquirt 2

The following steps are required to work with Megasquirt 2 Lambda 1 or 2

Lambda controller:

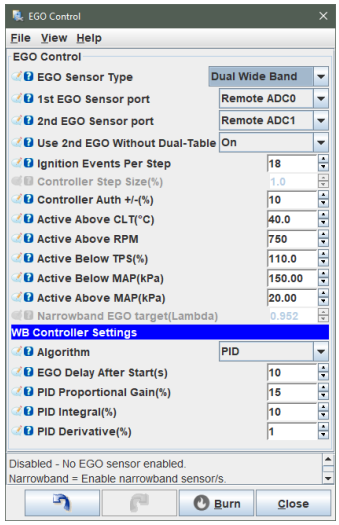


Broadcast mode:
Select „MS2 L1“ for controller as 1st Ego Sensor port
Select „MS2 L2“ for controller as 2nd Ego Sensor port
(One micro lambda controller can act as one lambda channel on the can bus)

After that press “Burn” setup finished for micro lambda controller

Megasquirt 2:

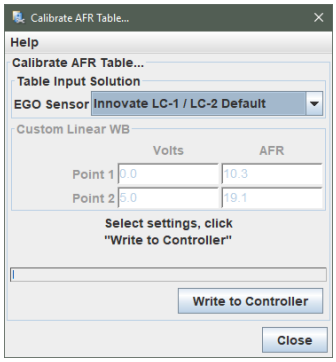
Go to „Fuel Settings“ → “EGO Control” to get the following menu:



EGO Sensor Type:
“Single Wide Band” if you have on micro lambda controller
or “Dual Wide Band” if you have 2 controllers.

1st EGO Sensor port: “Set to Remote ADC0”
2nd EGO Sensor port: “Set to Remote ADC1”

Go to “Tools” → “Calibrate AFR Table” to get the following menu:



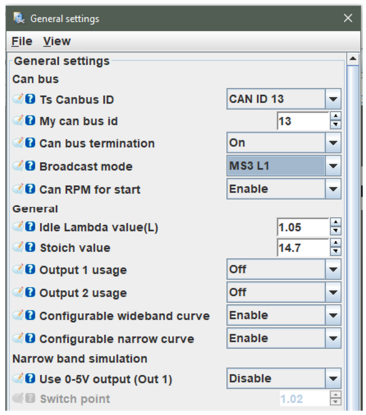
Select Innovate LC-1 / LC-2 Default
Press „Write to Controller“

Settings for MS2 finished.
If Micro Lambda controller is on and can bus is correctly connected you should see the lambda values at the gauges.

Setup Megasquirt 3

The following steps are required to work with Megasquirt 3 Lambda 1..6

Lambda controller:

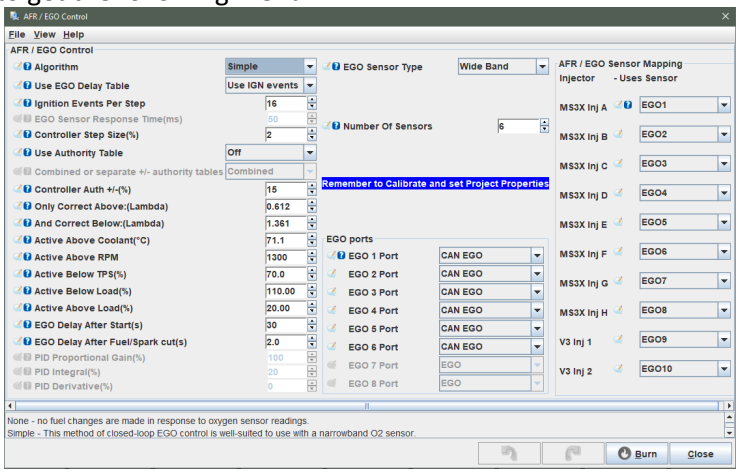


Broadcast mode:
Select „MS3 L1“ for controller as 1st Ego Sensor port
Select „MS3 L2“ for controller as 2nd Ego Sensor port
Select „MS3 L3“ for controller as 3rd Ego Sensor port
Select „MS3 L4“ for controller as 4th Ego Sensor port
Select „MS3 L5“ for controller as 5th Ego Sensor port
Select „MS3 L6“ for controller as 6th Ego Sensor port
(One micro lambda controller can act as one lambda channel on the can bus)

After that press “Burn” setup finished for micro lambda controller

Megasquirt 3:

Go to „Fuel Settings“ → “EGO Control” to get the following menu:



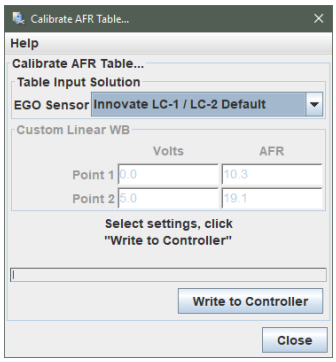
EGO Sensor Type: “Wideband”

Number of Sensors: “Number of controller you have installed”

EGO 1..6 Port: „CAN EGO“

AFR / EGO Sensor Mapping: „Map the correct EGO sensor to the injector channel“
(This is also covered in the MS3 manual)

Go to “Tools” → “Calibrate AFR Table” to get the following menu:



Select Innovate LC-1 / LC-2 Default
Press „Write to Controller“

Settings for MS3 finished.
If Micro Lambda controller is on and can bus is correctly connected you should see the lambda values at the gauges.

Specification

Supply voltage	10,8..16,5 Volt
Current	3A during cold sensor heat up, ~0,7A Running heated
Temp range	-10..85°C

Table 4

Revisions

V1.0	30.12.2019	First edition
V1.1	05.01.2020	Updated Table 1 con layout
V1.2	21.02.2020	Updated function description
V1.3	09.04.2020	Added ms2/3 setup help