Intro - Start Here

Where to start learning about FOME.

Power Cycle the ECU

📄 How to pick a fuel method

Use this to explain the pros and cons of each of the methods and which is more suitable for certain builds

📄 Guide to Tuning the ETB and VVT PIDs

Let's discuss the ETB and VVT PID controls and some give techniques for tuning these loops.

📄 Miata MX5 Quick Start Guide

So, you've just got yourself a shiny new BMM MX5 Miata ECU for your NA or NB, nice! If you're wondering how to install and get it running, you've co...

📄 Where to Find Tunes

Tunes and logs uploaded by the community can be found on FOME's tuning portal.

FOME Overview

What is FOME

Fome is an open-source project to create a whole vehicle open ecosystem that is able to work with modern vehicles with a view to safety, reliability, and sustainability. Fome is born out of existing open-source projects and intends to build upon those in a way to meets automotive standards and provides the best possible experience for users and contributors.

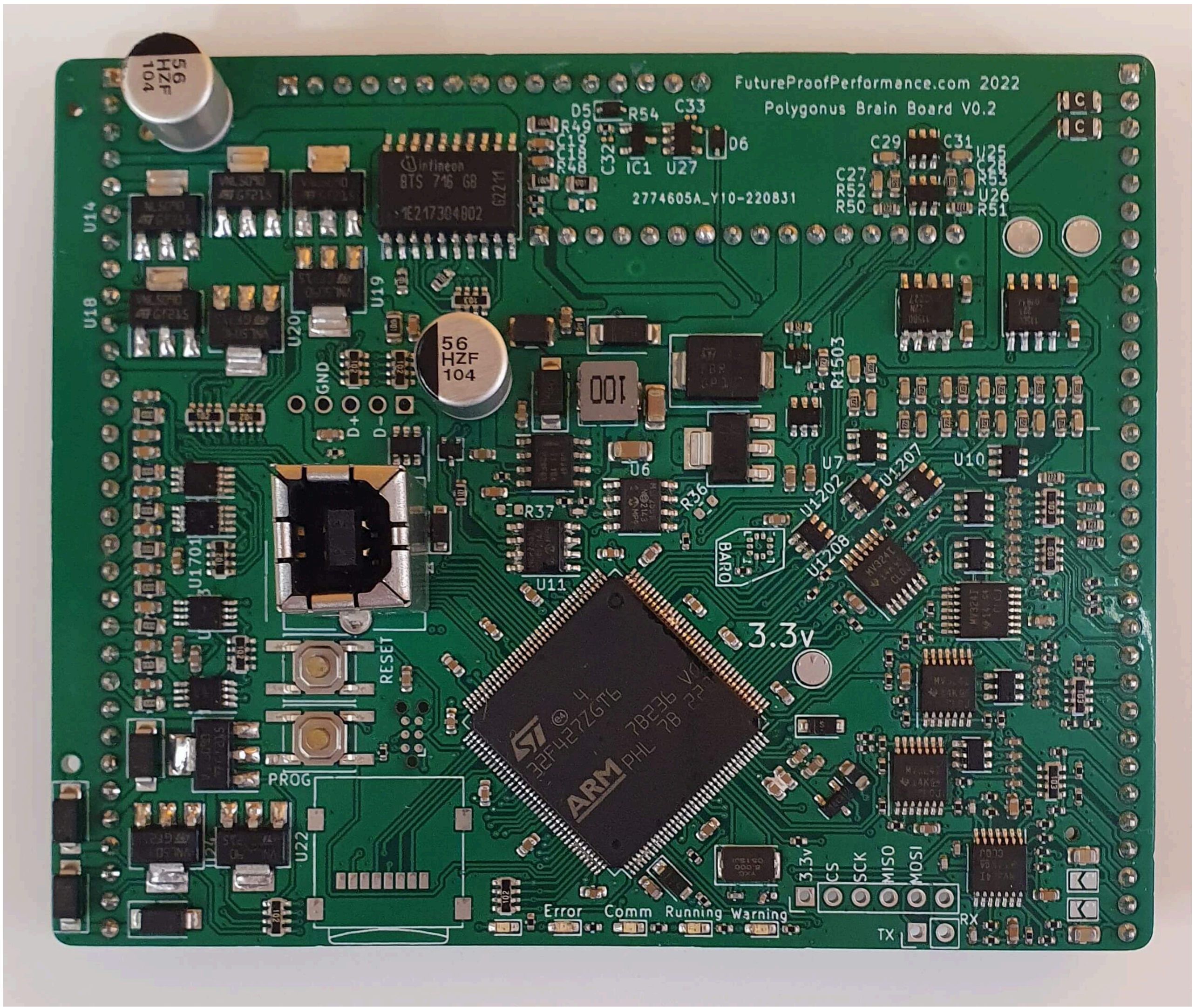
Which FOME hardware to pick

Currently there are a few sweet options for hardware, all have the same brain, but with different baseboards.

BRAIN

Polygonus 🧠

This is where the real action happens. Designed to be consistent across different base board applications to simplify design, ensure repeatability, and reduce cost. The Polygonus (said Puh-LIG-on-us) Brain is on all FOME ECUs.



BASEBOARDS

BMW

For the M50TU and M60 engines (and harnesses), this plug and play FOME ECU will allow you to use your turn signals (cringe jokes). No but really, E36 Chassis is hot right now. Street/Drag/Drift you name it. Everyone who had an S-Chassis now wants an E36. Hell you could use this in other chassis too.


15A Fuse
ETB Output Plug
ETB2+
PROG
RESET
GND
3.3v
CS
SCK
MISO
MOSI
TX
RX
Error
Comm
Running
Warning
FutureProofPerformance.com 2022
Polygonus Brain Board V0.2
INCOMPATIBLE WITH SIEMENS M52,
4 CYLINDER OR NON-VANOS M50
ONLY FOR USE WITH M50TU AND M60 WIRING LOOMS
For use with Polygonus Brain
FutureProofPerformance.com
88 Pin Motronic Base Board
LS B - LS C
AV7 - AV6
AV5 - AV4
AV1 - AV3
Analog Input Plug

Miata

You love big hair, short shorts, and tops off 24/7 (and stereotypes too). Luckily FOME has you in mind with a wide arrangement of options fot the Mazda Miata Platform. BeerMoneyMotorsports in sexy *billet* cases. NA,NB, 1.6, 1.8, VVT, turbo, ITBs, etc. You name, it, and its covered.

Harley

Whether you're cruising the highways, crushing that 1/4 mile, or just want more power, the Harley FOME ECU (designed for 2021+ bikes) is an excellent choice. NMSTEC has them available ready to go, also in a very nice *billet* case.

Coming Soon

Universal

Sometimes you don't know what car your engine is going in, or you want to do a swap. Regardless, a universal option is always good to have. It uses the common and reliable VAG (not that VAG, but Volkswagen Audi Group) plugs with a nice 120 pins. This means 8 cylinders of sequential fueling (up to 12 batch), plenty of inputs and outputs. (20 lowside, 4 highside, CANBUS, SD Card) It should be able to handle 95% of projects out there.

Honda

VTEC baby

Where to get firmware

FOME firmware releases are located here, on Github. You can download a release and install to the ECU with the console, or another way is to use TunerStudio (TS).

Tuner Studio Setup

Make sure that you have downloaded the latest version of TunerStudio (TS) from EFI Analytic's site. Although the base version of the software is free, it is strongly recommended to buy a license for the additional features including auto-tuning and the ability to customize the default dashboard.

Begin the setup by plugging the ECU into the laptop and opening TS. Create a new project and click *detect* under firmware. Select the COM port corresponding to the FOME ECU in the device list. If the COM port cannot be found or the firmware cannot be automatically detected, click *Other/ Browse* and load the .ini file for the ECU which can either be downloaded or found within the ZIP file on the USB device which appears when the ECU is plugged into the computer.

Note

As per the FOME Statement, we want to make clear that FOME is currently a fork of rusEFI, and that present boards are BOTH rusEFI and FOME compatible. As time goes on, and more changes are made, this compatibility may change.

Updating the firmware

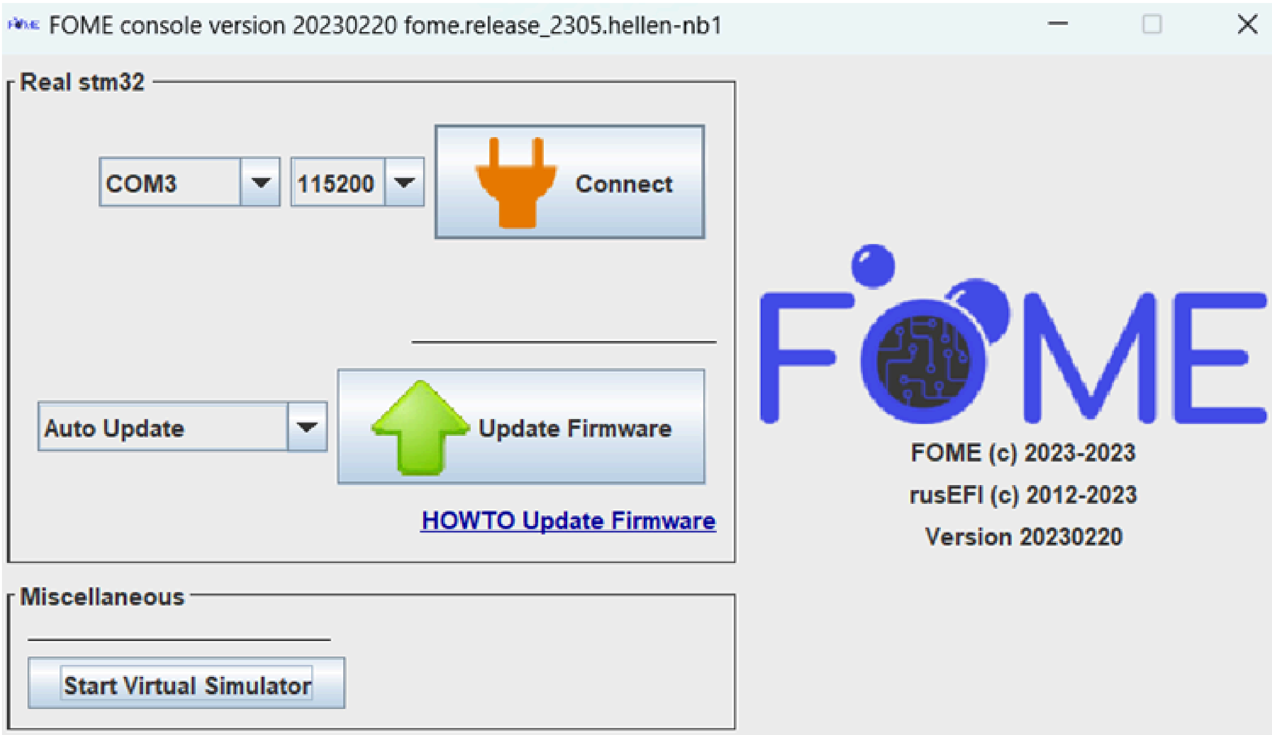
Updating the firmware is easy to do, however it is important to do it correctly to prevent errors or board misconfiguration. A guide to downloading FOME firmware and setting up TunerStudio can be found on this page.

Updating firmware involves a process sometimes called "flashing" or even "programming" the ECU. Generally, these all refer to the same thing: taking the compiled FOME firmware and loading it into the non-volatile flash memory on the ECU.

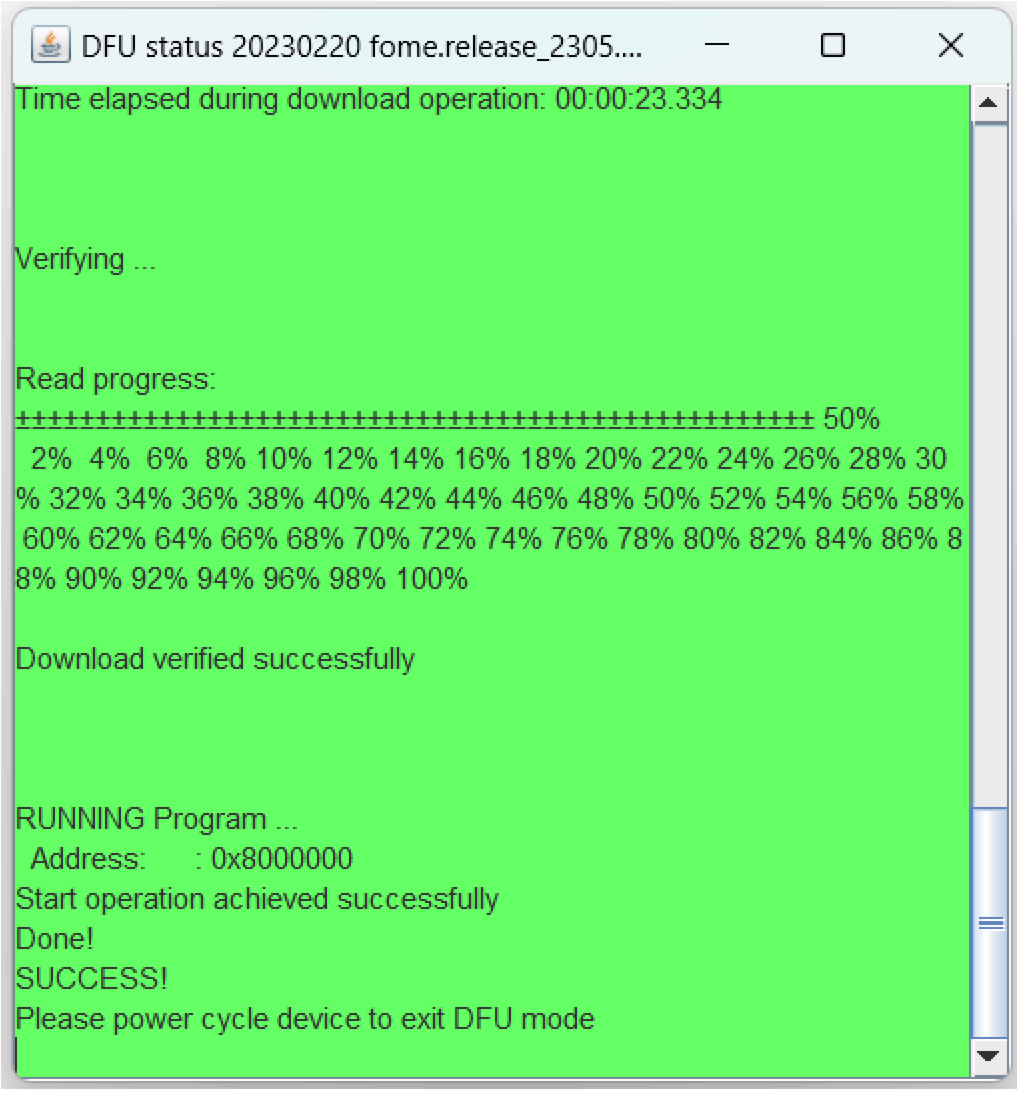
Updating the firmware with FOME console

After confirming the firmware to flash to the ECU, software to do so must be used. FOME provides a utility, the **FOME console**, which is capable of flashing firmware to supported ECUs and is the recommended way to update FOME firmware. The FOME console is included with each firmware release, and bundles can be downloaded from the release page.

After extracting the bundle contents, navigate to the 'drivers/' directory and install the drivers. After, navigate to the `console/` directory and run the FOME console program (e.g. `fome_console.exe` if using Windows). After first confirming **TunerStudio is not running**, connect the ECU to the computer. FOME console will automatically detect the board and present similar to the display below.



Next, choose the *Update Firmware* button and leave the board connected until the console notifies that the update was successful with a green display. Only once the console has presented this success message, disconnected and re-connect the ECU to the computer to reboot it with the new firmware.



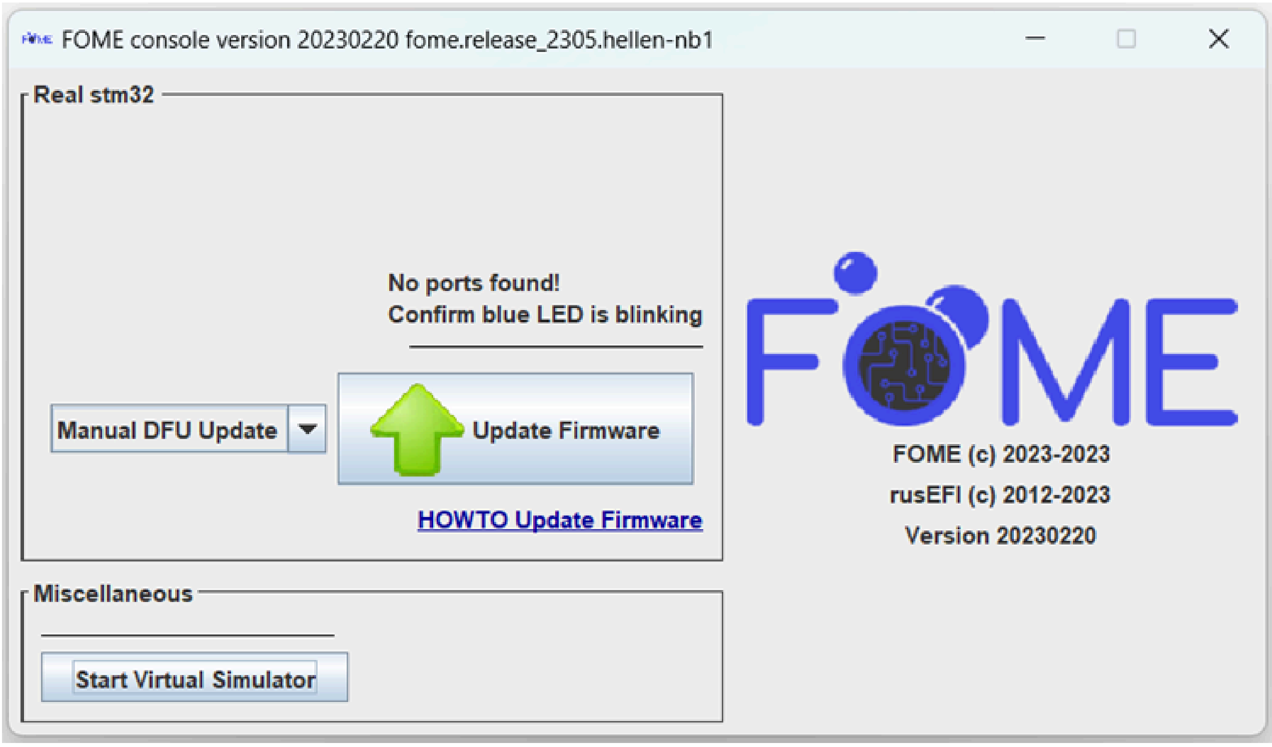
If the update completes successfully, congratulations; proceed to the Setting up TunerStudio section. If the update does not complete successfully, the firmware may need to be loaded by first forcing the board into bootloader or Device Firmware Update (DFU) mode, documented in the next section.

Updating firmware via manual bootloader/DFU mode

In the event of a firmware update/flash failure, or some other firmware corruption issue, such that FOME console nor TunerStudio recognize the ECU when connected, the board needs to be first forced into bootloader mode before updating. In these situations, the FOME console cannot communicate with the firmware and so is unable to configure it into bootloader mode for updating.

To enter bootloader mode, on the ECU you will need to access buttons or pins on the main board to force the processor into this mode. Some boards use a momentary button, likely near to the reset button, to force this mode during power-up. Others might use a set of pins or pads that need shorted to force this mode during power-up. Press the button or short the pins with the ECU disconnected from the computer, then simultaneously connect the ECU into the computer. Once connected, the button can be released or the short removed.

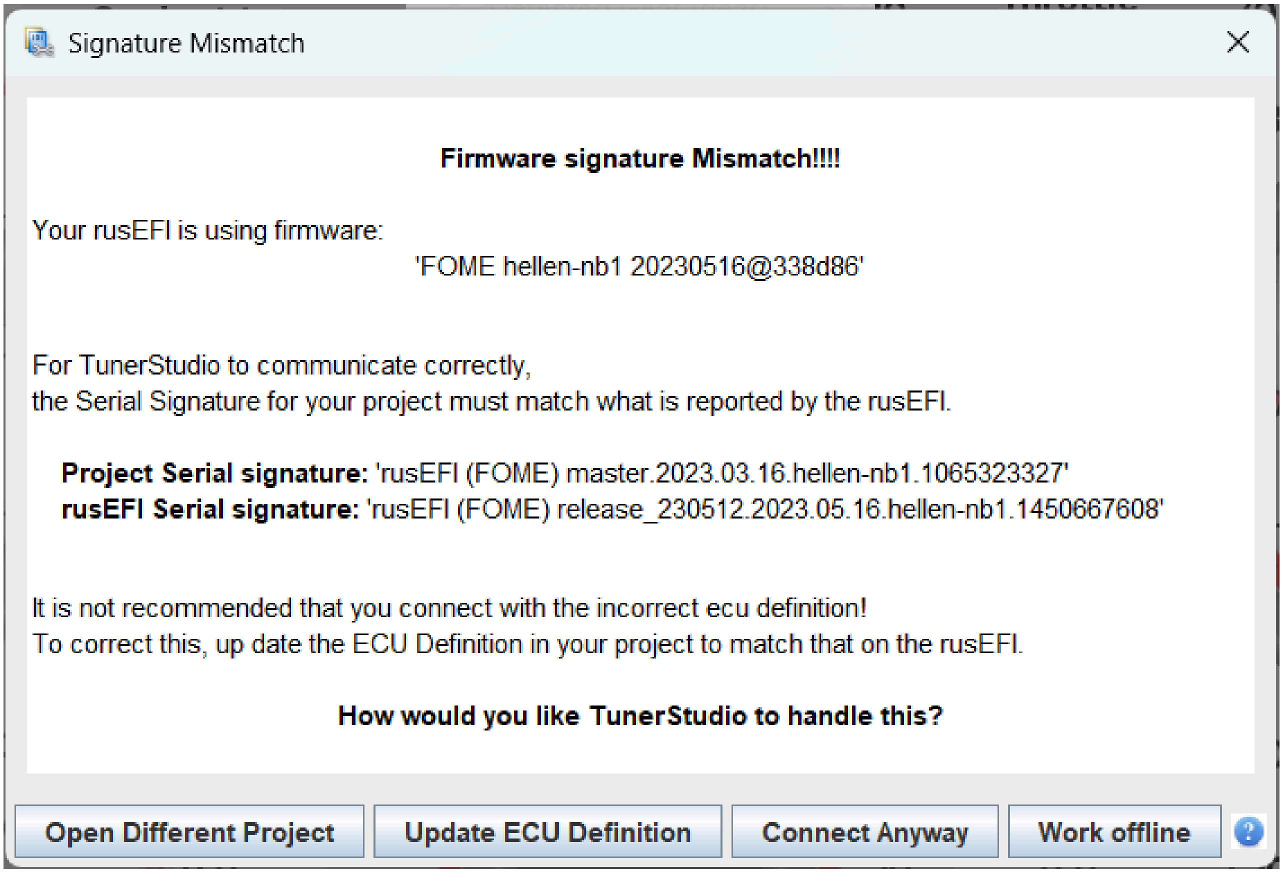
When a board has been successfully booted into bootloader/DFU mode, the FOME console will recognize this and display the manual DFU update selection.



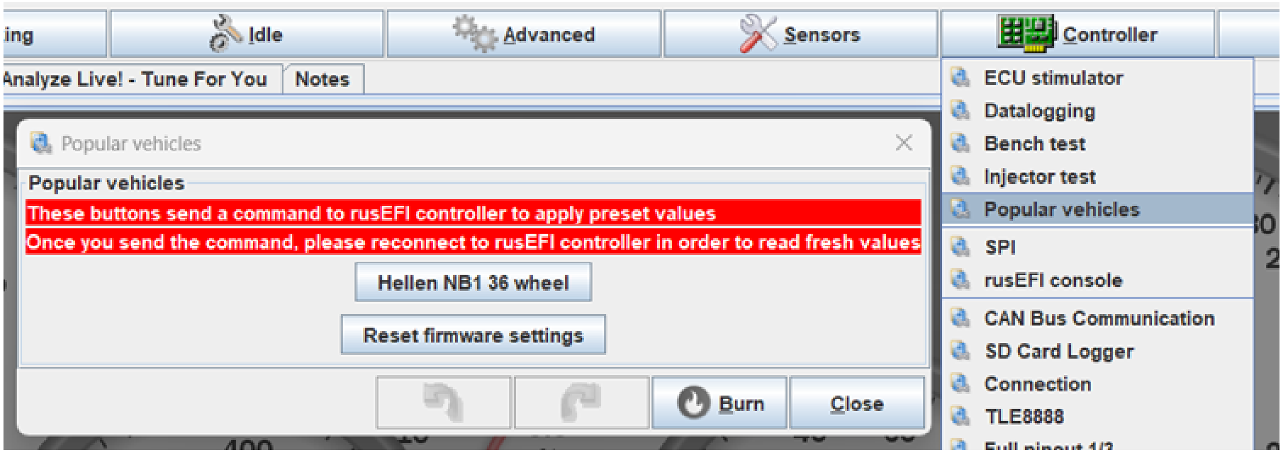
Next, choose the *Update Firmware* button and leave the board connected until the console notifies that the update was successful with a green display. Only once the console has presented this success message, disconnected and re-connect the ECU to the computer to reboot it with the new firmware.

Setting up TunerStudio

With the ECU connected, open TunerStudio with the associated project. TunerStudio will recognize that the project's ECU definition is out of date and display a prompt to update it. Choose the *Update ECU Definition* button, which will attempt to automatically update the definition from official sources. If an official definition cannot be found, manually load the `.ini` definition file associated with the firmware (i.e. from the downloaded bundle).



Next, under the *Controller* tab, open the *Popular vehicles* menu and choose the top button (e.g. *Hellen NB1 36 wheel* as shown here) to load the preset for your vehicle (**not the button to *Reset firmware settings***).



Once the preset has been loaded, you can either start tuning or load your tune from the older firmware versions under *File > Load Tune (msq)*. After that, the update is complete!

Troubleshooting and Other Notes

See the Flashing Software Notes page for troubleshooting and other flashing notes.

Performing a safe first start

page to outline the best practice for doing a first start

How to first setup a board for use

Outputs, etc. Checking correct tune loaded

How to test inputs and outputs

Input tests

Output tests

Tips and Tricks

Power Cycle the ECU

Any change that you make to the tune or ecu should be followed by a power cycle of the ecu. The only exception to this is for tuning tables (VE Timing etc.)

Always Have a Main Relay

The main relay is an essential component of the FOME ECU system. It is responsible for delivering power to the ECU when the key is turned on. Without the main relay, the ECU would not receive any power and would not be able to function properly.

Why Use a Main Relay?

Using a main relay has several benefits. First, it helps to protect the ECU from power surges and voltage spikes that can occur when the engine is starting. Second, it allows for a more secure and stable connection between the ECU and the power source, which can help to prevent electrical interference and noise.

It's important to note that the wiring of the main relay may vary depending on the specific ECU and vehicle configuration. Always refer to the manufacturer's instructions and wiring diagrams when installing the main relay to ensure proper installation and operation.

Tune Sensors While Powered From +12V

Not all boards, perhaps most even, do not provide a fully re-buffed +5 volt supply when powered by e.g. only USB. Therefor, losses from the host supply will leave the +5 volt rail available to sensors drooping in comparison to when powered by a running engine (+12 volt available). Tuning in this manner will likely cause a significant difference in readings compared to a fully powered ECU.

Why Do I Have XYZ Error?

Check error code and compare to this list (grab list out of firmware)

Table Axis

Make sure these are always numerically ascending (i.e. 1, 2, 3; not 1, 1, 3; not 1, 3, 2).

If you have these going from a large number to a smaller, stuff is not going to work.

How to pick a fuel method

Use this to explain the pros and cons of each of the methods and which is more suitable for certain builds

Guide to Tuning the ETB and VVT PIDs

Let's discuss the ETB and VVT PID controls and some give techniques for tuning these loops.

$$u(t) = K_{\mathrm{p}} e(t) + K_{\mathrm{i}} \int_0^t e(\tau)\, \mathrm{d}\tau + K_{\mathrm{d}} \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

More Details

Wikipedia's PID Controller page's operational description.

More details about ETB PID control are found in the Advanced Features section's page ETB PID and Autotune.

More details about VVT PID control are found in the Advanced Features section's page VVT PID Control.

Miata MX5 Quick Start Guide

So, you've just got yourself a shiny new BMM MX5 Miata ECU for your NA or NB, nice! If you're wondering how to install and get it running, you've come to the right place. This guide will cover how to install the ECU to the car with a Bosch LSU 4.9 wideband oxygen sensor and a mass air pressure (MAP) line. Installation of additional sensors or peripherals is covered in the advanced guides.

> ***NOTE:*** Before commencing the ECU installation, it is recommended to jack up the car or drive it onto ramps in the case when the oxygen sensor location is under the vehicle.

Required Tools and Components

- BMM Miata ECU
- BMM wideband adapter harness
- BMM options port pigtail
- Genuine Bosch LSU 4.9 oxygen sensor
- 3 metres of silicone vacuum hose 5/32" or 4mm internal diameter
- 4mm straight barb joiner
- 22mm wrench or 22mm oxygen sensor socket
- Timing light
- USB cable (included with ECU)

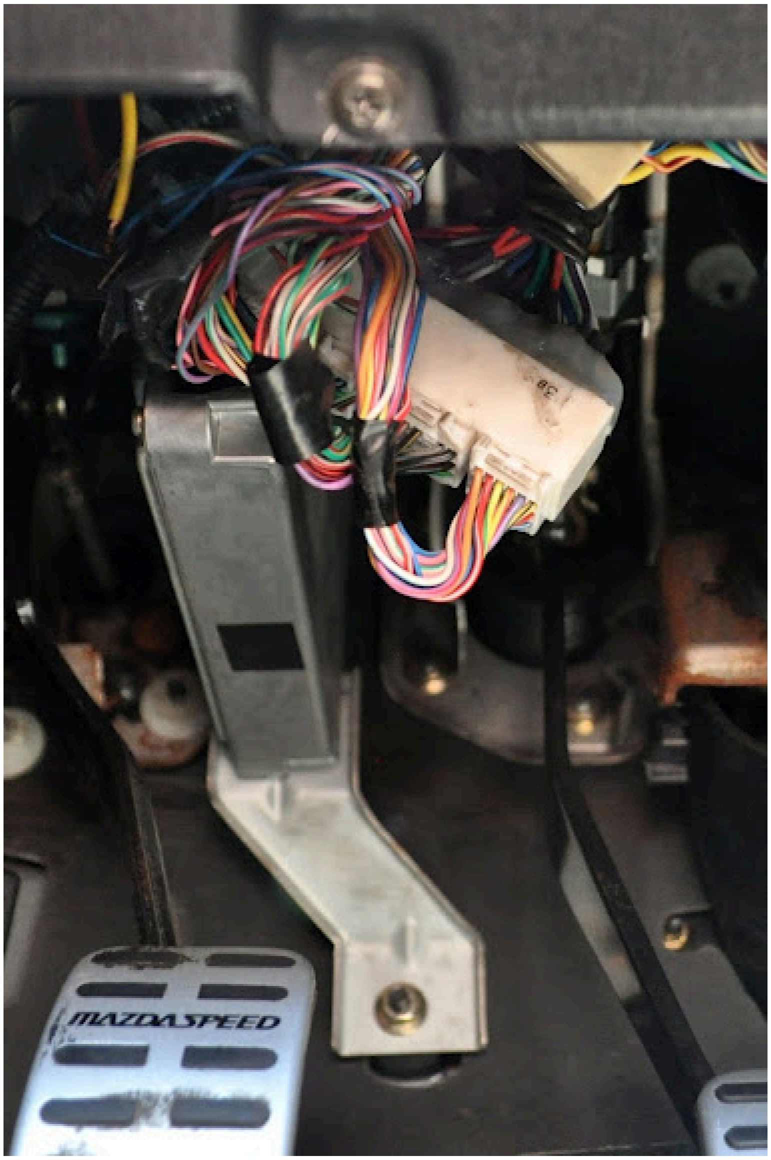
- Windows, Mac or Linux laptop with an installed copy of EFI Analytics TunerStudio
- Spanner and socket set

Removing Original ECU

The stock ECU location for a Miata will be in one of three spots depending on the driving side and year:

Left Hand Drive NB

The ECU can be found above the pedals, next to the steering column.



90-93 Left Hand Drive NA and Right Hand Drive NA/NB

The ECU can be found under the carpet in the passenger side footwell. To access this, the carpet needs to be unhooked from the vertical trim piece on the edge closest to the passenger door. Removing this trim piece can also simplify access. The ECU kick plate will also need to be removed after taking off the five 10mm nuts and bolts holding it in place.

94-97 Right Hand Drive NA

The ECU can be found behind the passenger's seat, under the carpet. Move the seat forwards all the way. Next, the passenger door sill needs to be removed with a philips head screwdriver so that the carpet towards the back of the seat can be pulled back to reveal the ECU.

Once the ECU has been located on your Miata, disconnect the car battery then remove all electrical plugs to the ECU. Un-bolt any remaining ECU mounting brackets from the car with a 10mm socket and the ECU should now be free from the car. The last step is to use a philips head to remove the factory ECU mounting brackets from the stock ECU case for these will be needed to mount the BMM ECU.

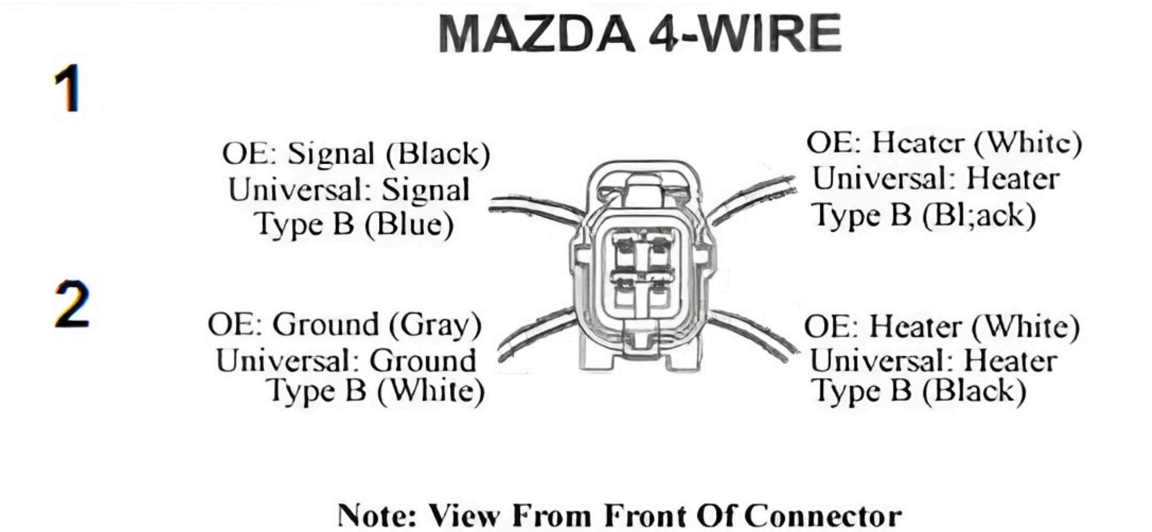
Connecting Wideband Oxygen Sensor

> ***NOTE:*** It is imperative that you use a genuine Bosch LSU 4.9 sensor rather than a cloned product. A fake LSU 4.9 will not provide accurate readings and can cause a lot of headaches down the track. The best way of avoiding a fake sensor is to buy directly from a reputable supplier of vehicle parts rather than generic large online re-sellers. Typical part numbers for this Bosch sensor include: 17025, 17212, 17123 and 17217. The notable difference between these part numbers is the cable length so it is recommended to measure what length you need ahead of time.

Find the factory oxygen sensor on the exhaust and unplug it from the wiring harness. In the case that the car has multiple oxygen sensors, the one to remove is the closest sensor to the engine block before any catalytic converters. Next, unscrew the sensor and replace it with a Bosch LSU 4.9 wide-band sensor. Connect the sensor to the BMM wideband adapter harness. The trailing end of the harness will need to be fed through the firewall into the cabin. The easiest way of doing this, as shown in the image below, is to cut a hole in the nearest firewall bung to the stock ECU location, and feed the cable through that. Cable tie the wiring away from any hot areas of the engine bay. Inside the cabin, connect the wideband adaptor harness plug to the options port pigtail and plug it into the ECU.

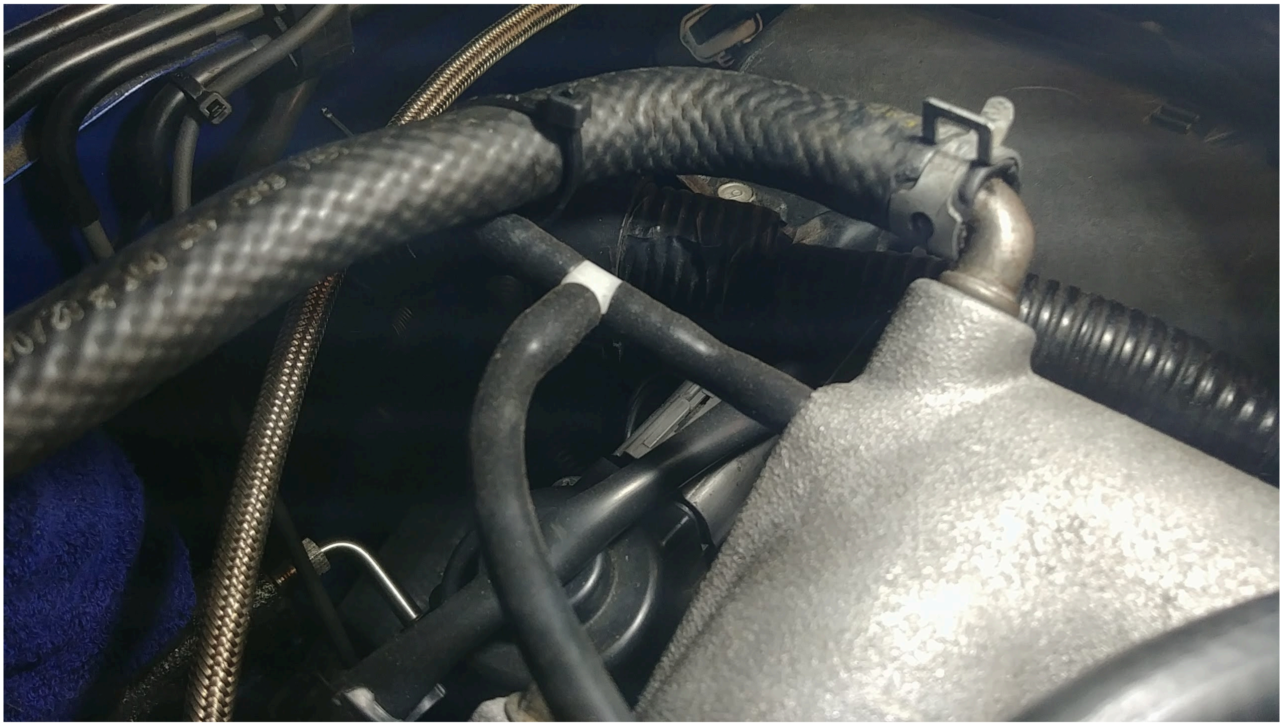
Using an External Wideband Controller

In the case you wish to use an external wideband controller such as an *AEM X-Series Wideband UEGO AFR Sensor Controller Gauge*, the wideband sensor should be plugged into the wideband controller instead of directly into the ECU. The best way to wire in the controller is directly to the old narrow band oxygen sensor plug on the car based off the diagram below. In this diagram, pin 1 goes to the controller analogue output, pin 2 to the signal ground, pin 3 to the controller 12V input and pin 4 to the other controller ground (if applicable). Make sure to double check the voltages on the pins before connecting the controller to them. The external controller also requires additional setup in Tuner Studio which will be covered later.

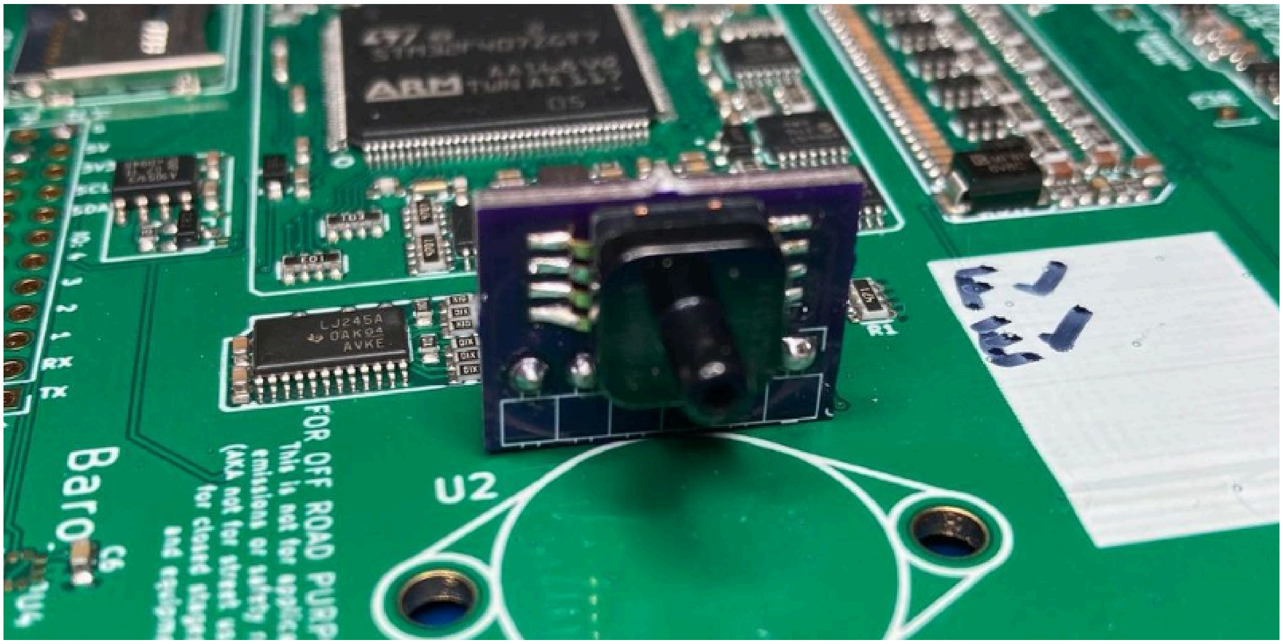


Connecting MAP Line

Look around the intake manifold for any spare vacuum ports that lie after the throttle body and connect the vacuum line. If there are no spare ports, pick one and attach the vacuum line to it using a tee piece. It is recommended but not required to cable tie the vacuum line to the tee. In the image below, there was a free vacuum port on the back of the intake manifold which has been tee'ed off into the MAP line and the blow off valve line (only applicable on turbo charged vehicles).



Like the oxygen sensor, feed the line through the bung in the firewall to the ECU. If you have a 4mm barb joiner, connect the vacuum line to the vacuum line protruding from the BMM ECU case. If you do not have a barb joiner, open up the BMM ECU case with a philips head and feed the vacuum line through the case. Mock up the position of the case in the car before cutting the vacuum line to length. Pull the vacuum line onto the MAP sensor on the ECU (the sensor with the nipple on it pictured below) and optionally fasten it with a small cable tie. The ECU can now be put back into its case.



Using the MAP line combined with an intake air temperature (IAT) sensor, the BMM ECU can run the car using what is known as speed-density air metering. This means that you can unplug the mass air flow (MAF) sensor or the air flow meter (AFM) for the NA 1.6L vehicles. Removing these sensors and replacing them with a pod filter directly to the intake can even result in a fractional power increase from the reduction in intake restriction.

Additional Steps for NA6 Vehicles

The NA6 1.6L vehicles which use an AFM instead of a MAF require a few additional modifications to run with a BMM ECU. A manual NA6 do not have a variable throttle position sensor (TPS) like the automatic NA6, later model NAs and all NBs. An NA6 also needs an external intake air temperature (IAT) sensor wired in as the AFM which has one inside is typically removed. They also require a jumper for the ECU to control the fuel pump which was previously the job of the air flow meter.

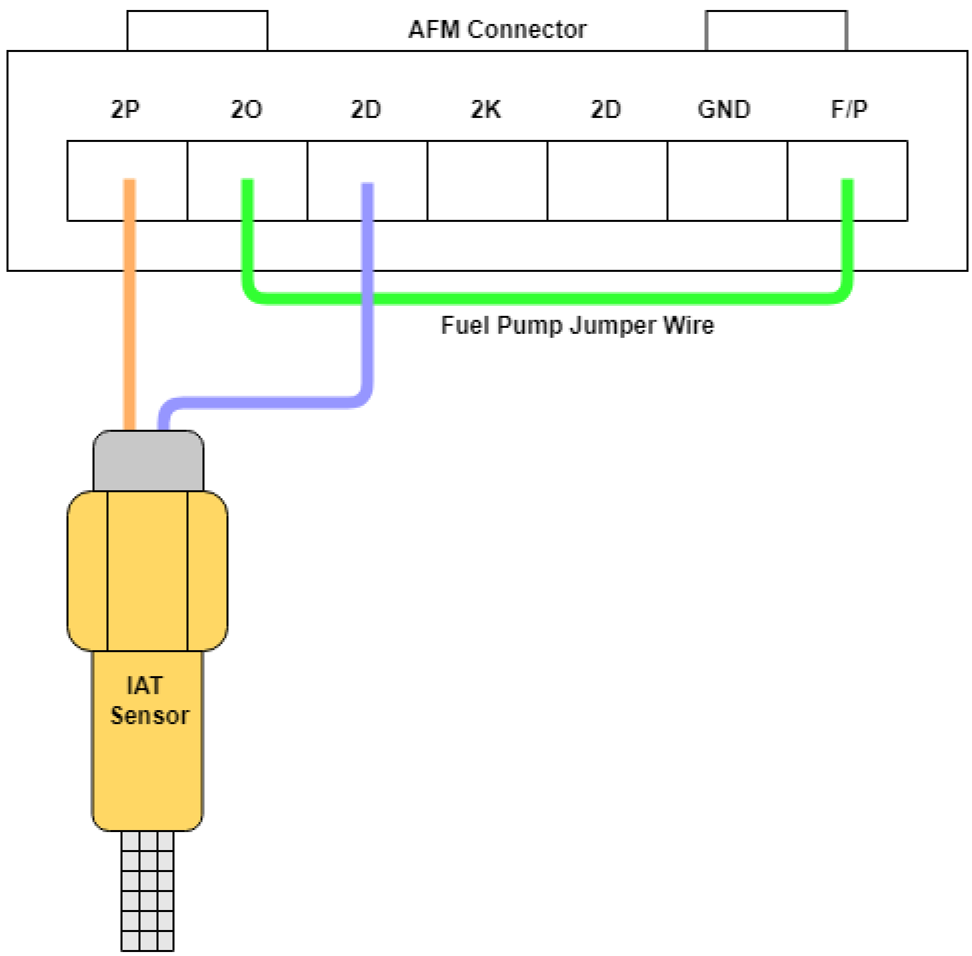
The first step is to disconnect the factory TPS sensor. **This is very important or it will cause a short circuit later.**. The TPS sensor location is shown in the image below.

The BMM ECUs for this vehicle include a KIA TPS and adapter. The KIA TPS will plug straight to the OEM TPS plug without any additional wiring. If using another variable TPS that requires re-wiring, the NA6 TPS connector pinout is as follows:

Function	Cable Colour
Signal	Green/White
Ground	Black/Green

Function	Cable Colour
5V Reference	Red

The next step is to wire up the IAT sensor and to add a jumper wire to the AFM connector as per the wiring diagram below. Any IAT sensor with two wires can be used although a GM IAT sensor is recommended as FOME already has a configuration for it. As the IAT is a resistance-based sensor, the orientation of the wires does not matter.



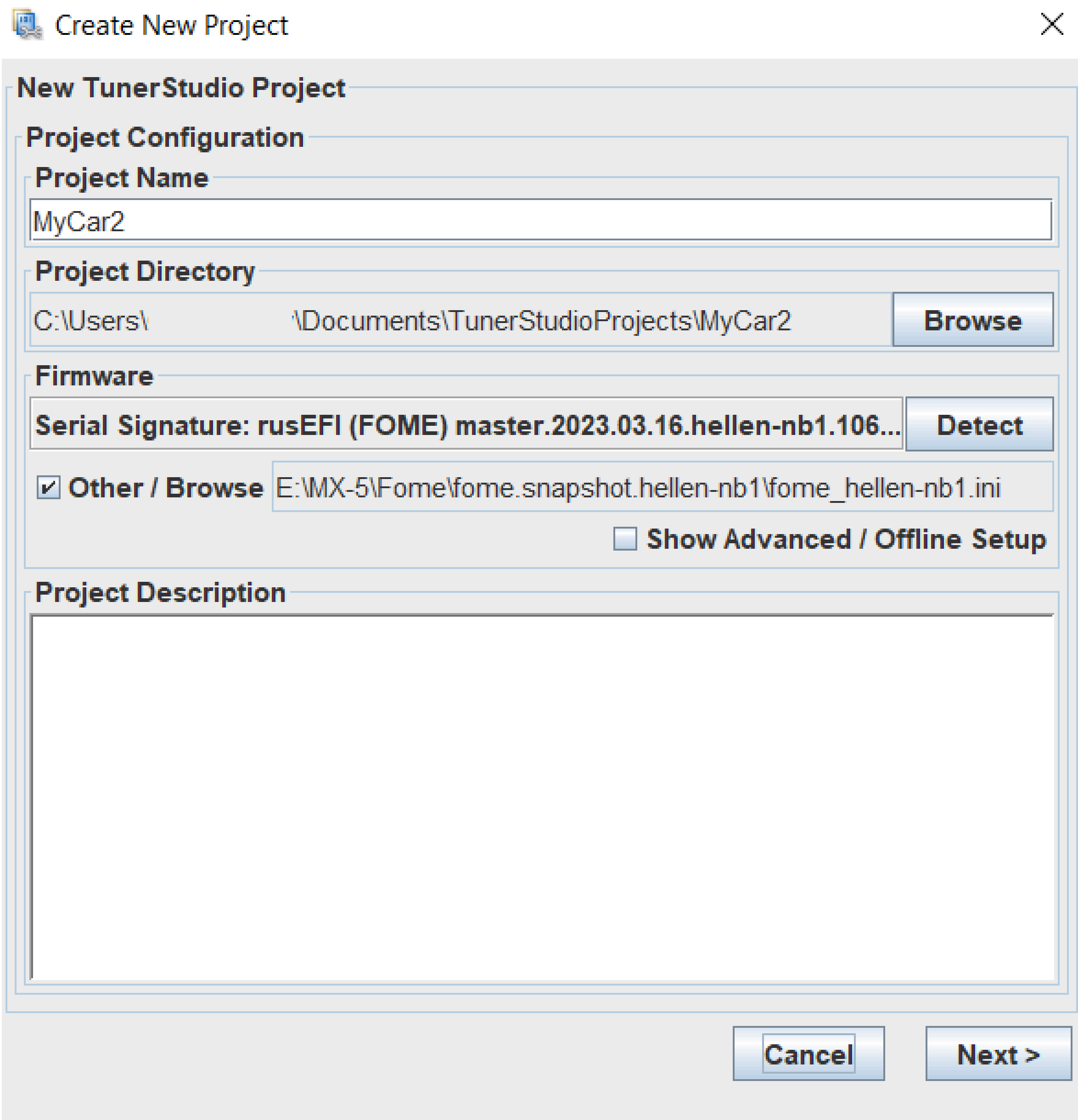
Connecting the ECU

Now that the MAP line and wideband are connected to the ECU, the remaining wiring harness plugs from the OEM wiring loom can be plugged into the ECU. Take the factory ECU mounts and attach them to the BMM ECU case. The ECU can now be re-installed into the factory location. The car battery can now be re-connected.

Tuner Studio Setup

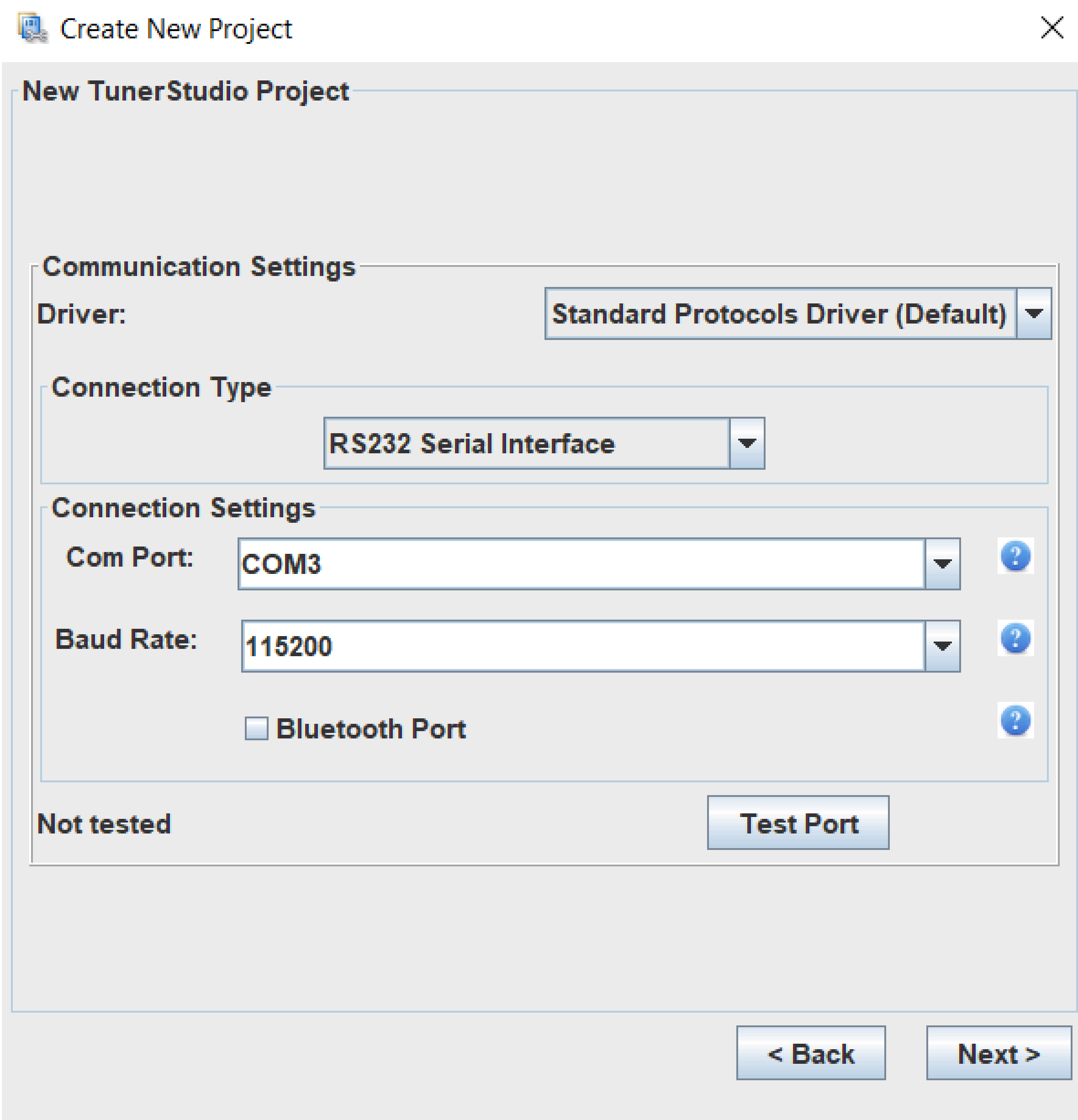
Make sure that you have downloaded the latest version of TunerStudio (TS) from EFI Analytic's site. Although the base version of the software is free, it is strongly recommended to buy a license for the additional features including auto-tuning and the ability to customize the default dashboard.

Begin the setup by plugging the ECU into the laptop and opening TS. Create a new project and click *detect* under firmware. Select the COM port corresponding to the FOME ECU in the device list. If the COM port cannot be found or the firmware cannot be automatically detected, click *Other/ Browse* and load the .ini file for the ECU which can either be downloaded or found within the ZIP file on the USB device which appears when the ECU is plugged into the computer.



In the next dialog choose between lambda or air fuel ratio (AFR) as your display units. lambda is recommended as it is easier to comprehend and tune with. For example, the ideal or stoichiometric AFR for regular petrol is 14.7 (14.7 parts air to 1 part fuel) which corresponds with a lambda of 1. Lambda represents the percentage of air in the combustion chamber compared to the amount needed for ideal or stoichiometric combustion to occur. If a car is running 10% lean, the AFR would be 16.17 and lambda would be 1.1. If the car is 10% rich, AFR would be 13.36 and lambda would be 0.9. Looking at lambda, it is instantly obvious what percentage rich or lean the engine is running but with AFR, it requires more effort. **The only time AFR should be selected here is if you are using an external wideband controller**.

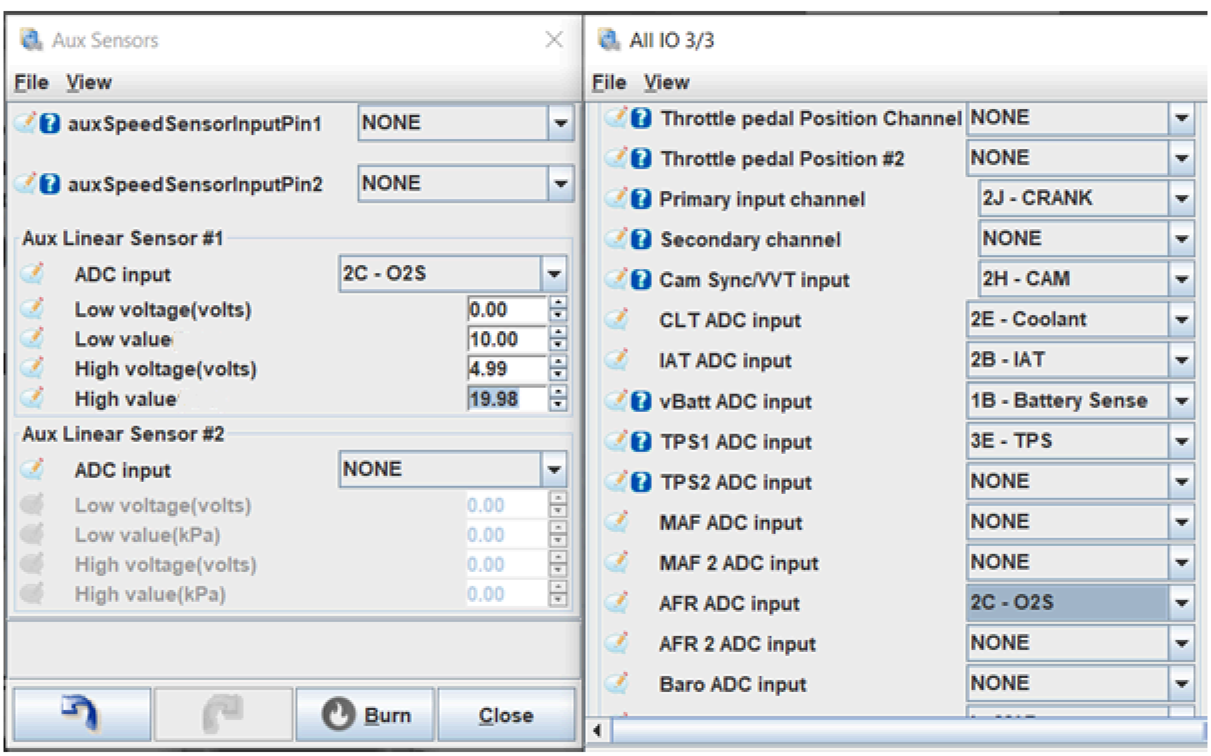
In the third dialog box, configure it as shown in the image below but select the com port which corresponds to your ECU. If unsure, go to the device manager on your computer and it should list the COM port number next to the name of the ECU. Click *Test Port* and if successful, move to the next dialog.



In the final dialog box, select the default gauge layout (you can change this later as you wish) and click *finish*. The last step before cranking the engine is to click the *Ignition* button to open the ignition settings and change the timing mode from *dynamic* to *fixed* and the fixed timing setting to 10 degrees. This will lock the engine to operate at 10 degrees of timing so that you can set the base timing.

Additional Tuner Studio Steps for an External Wideband Controller

To set up the external wideband controller there are several additional steps in Tuner Studio. First, your display units should be set to AFR for this as already stated. If you forgot to do this earlier, press *CTRL + P* to open the vehicle properties. Now, open the *Aux Sensors* dialog under *Sensors* and the *Full Pinout 3/3* dialog under *Controllers*. As per the diagram below, set the *AFR ADC Input* and *ADC Input* to the pin corresponding with *O2S* (pin 2C for the example). for the values in the *Aux Linear Sensor #1* box you need to reference the manual of your wideband controller for what voltages correspond to its AFR outputs. In the example below, 0V corresponds to an AFR of 10.0 and 4.99V corresponds to an AFR of 19.98. Once these are set, click *Burn*.



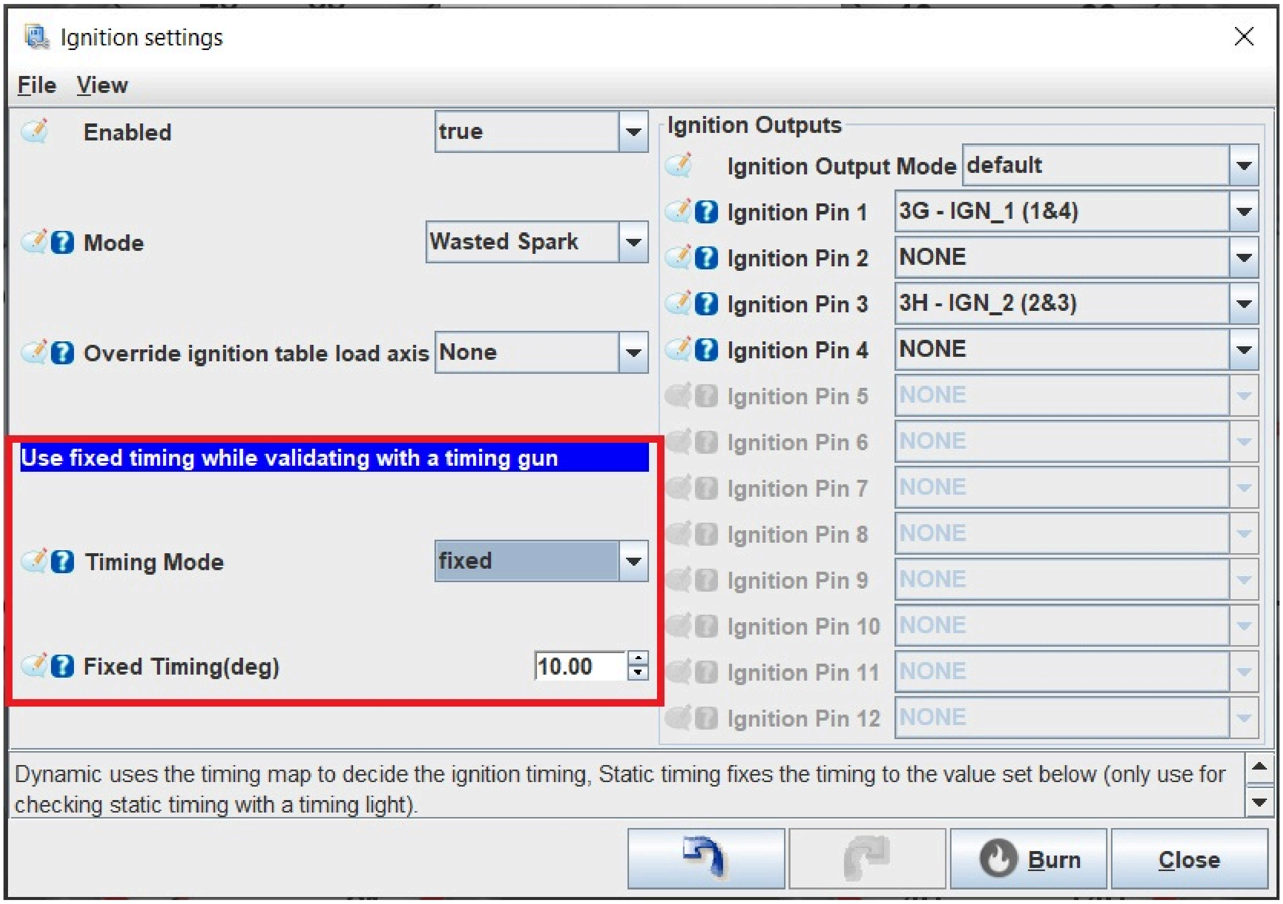
After completing all of the setup steps, you can go ahead and turn the car key two clicks to *ON* and listed for the fuel pump priming. Once the fuel pump has primed, go ahead and start the engine. Let it run for a few seconds and turn it off again. **Do not drive the vehicle yet, there are still several steps to complete before the car is ready for a drive**.

Set Base Timing

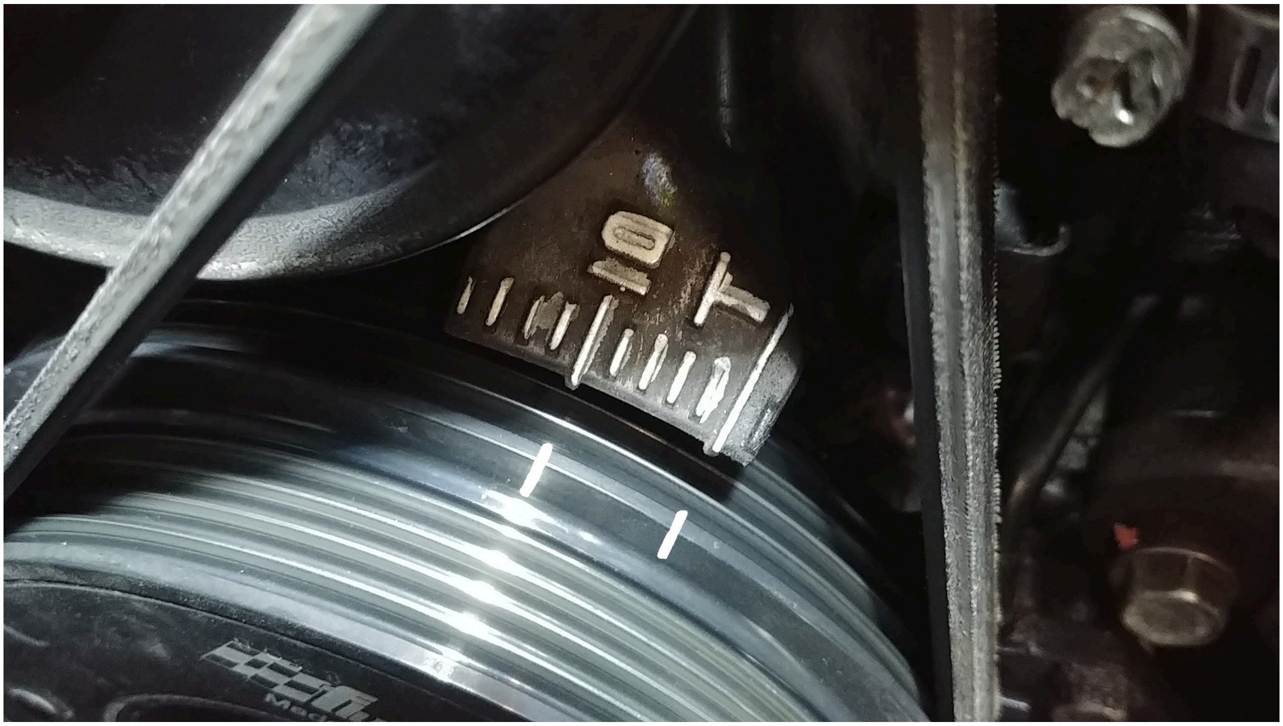
The car should start on the base map although once it is running, the base timing needs to be set up. This syncs the timing between the ECU and the car so that they are both reading the same values. Typically. the base timing will be a few degrees out from the base map as it varies slightly from car to car.

To set the base timing, connect the timing light power to a spare 12V battery and the inductive clamp onto the cylinder 1 spark plug lead (the closest spark plug to the front of the engine bay). Ensure that the arrow on the inductive clamp is pointing along the wire towards the spark plug, not towards the coil pack.

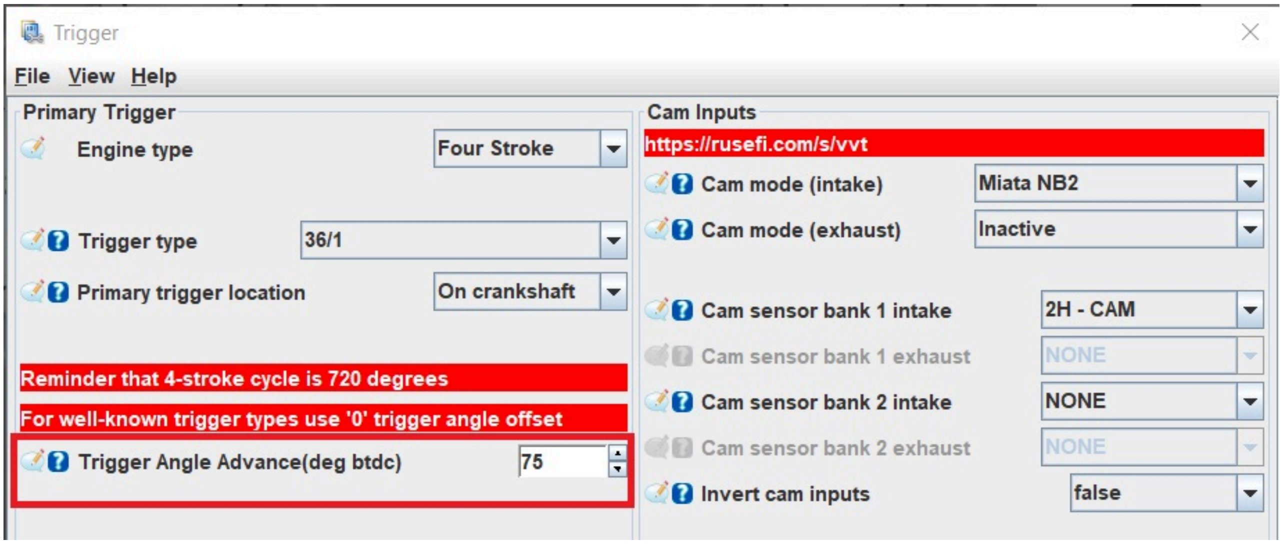
In TS, under *Ignition > Ignition Settings*, set the timing mode to *fixed* and *10* degrees then burn the configuration.



Now start the car and hold the timing gun trigger, shining the light onto the bottom harmonic damper pulley on the front of the engine. This pulley has two timing marks on it and a labelled cover above it. When the timing is spot on, these marks on the rotating pulley will line up with the *10* and *T* marks on the cover as shown below.



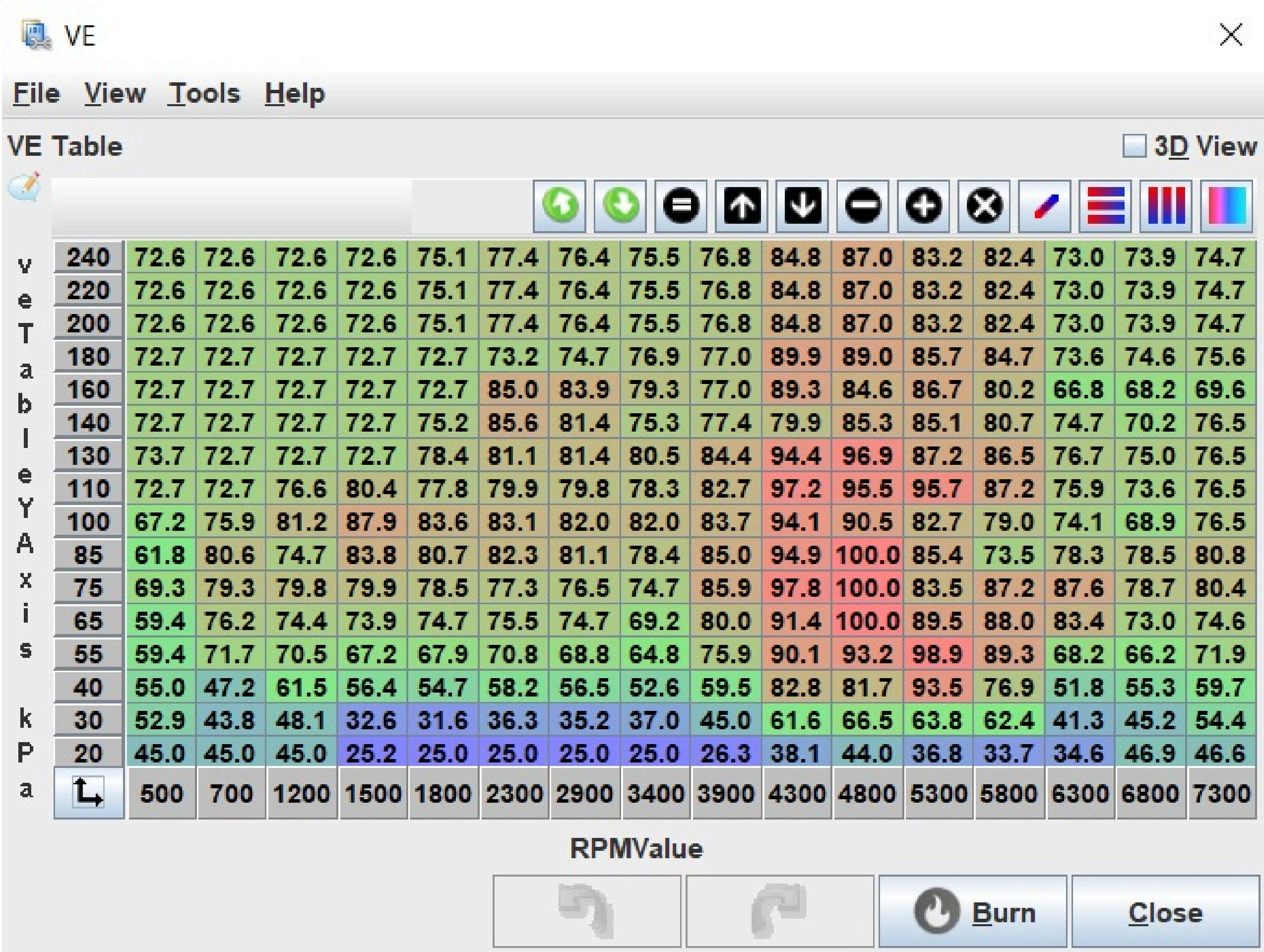
If your timing marks do not line up like in the image above, you will need to change the base timing. Count how many marks the timing is off by and turn the car engine off. In TS, go to *Base Engine > Trigger* and increase/decrease the *Trigger Advance Angle* by the amount of marks the timing was off by then burn the configuration. Repeat this process until the timing marks line up then change the timing mode back from *fixed* to *dynamic*.



First Drive and Tuning the VE Table

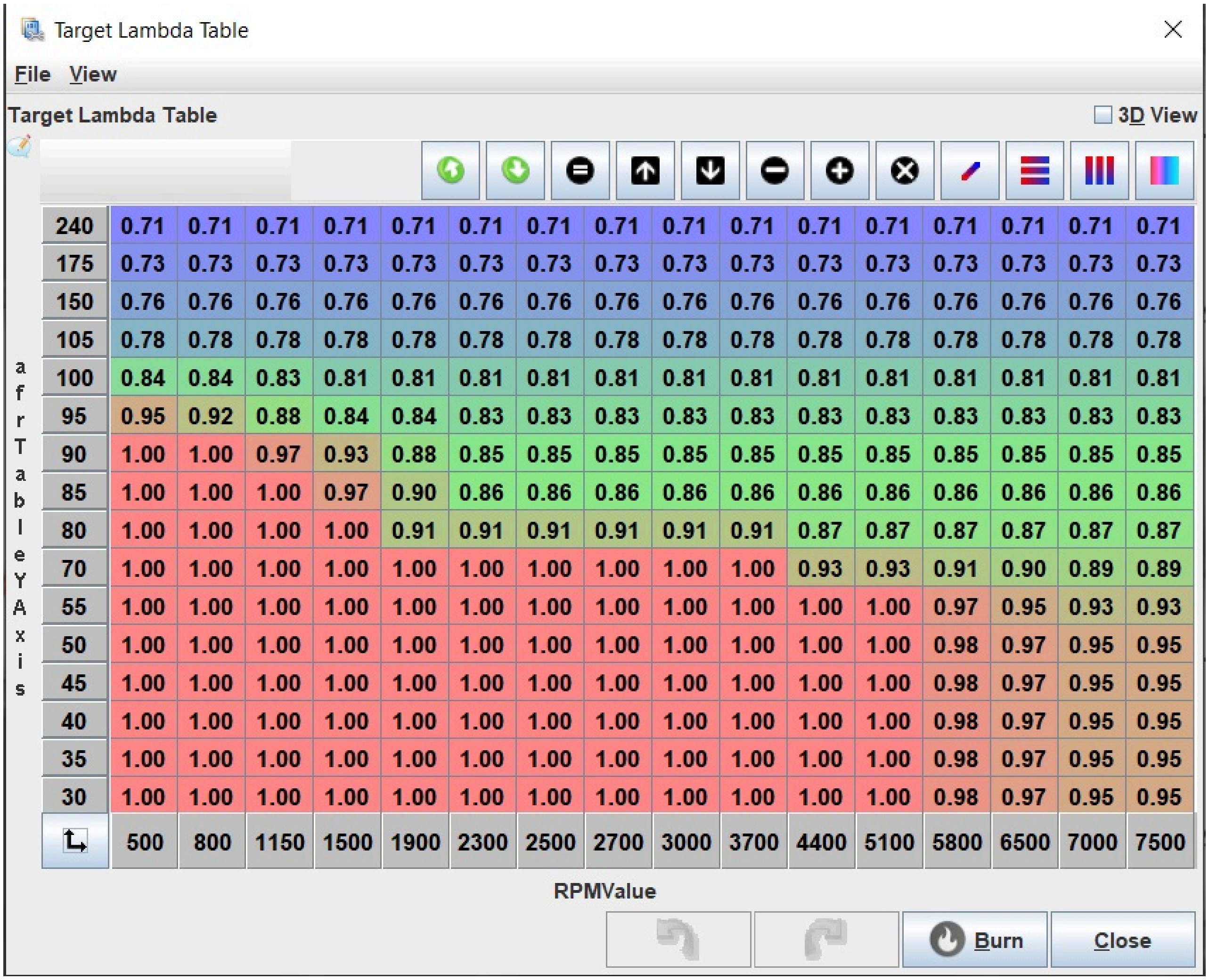
Everything is now ready to take your Miata for its first drive. You can't go and thrash it straight away though as the VE table which determines the fuelling needs to be tuned for your vehicle. Before you take the car for a drive, make sure your laptop is charged.

Start the car and plug the laptop in. Within 30 seconds, the lambda gauge should wake up and start displaying a value. For now, you want that value to be around 1 meaning that the exact amount of fuel is being supplied to the engine for perfect combustion to occur. To change the lambda value, you need to modify the VE Table under *Fuel > VE*. VE stands for volumetric efficiency which is the efficiency that the engine can move the fuel and air mix into and out of the cylinders. An example of a VE table is shown below (do not copy this table as it is off a highly modified vehicle). The table adjusts the VE percentage (represented by the numbers on the grid) based on the engine speed (represented as revolutions per minute - RPM) and engine load (represented as the MAP). With the engine running, blip the throttle and see how the indicator moves around the different table cells as the engine state changes.



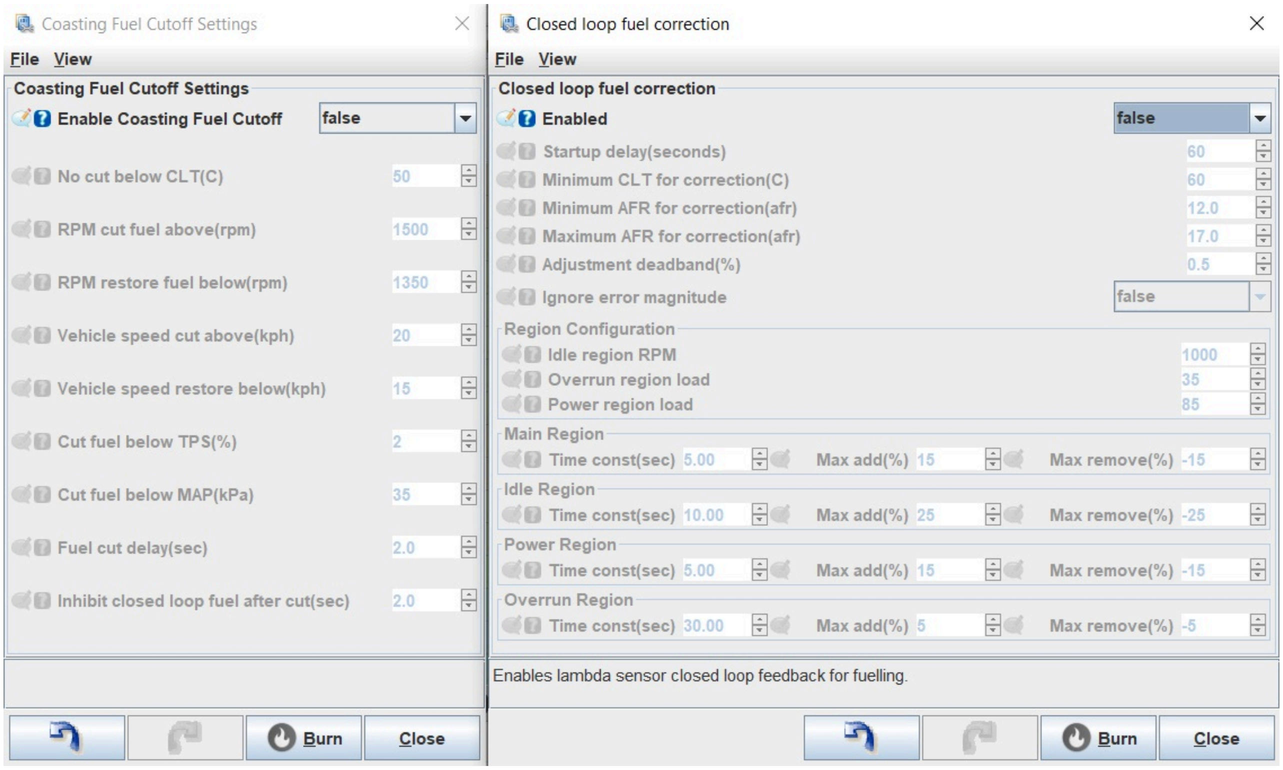
kPa \ RPM	500	700	1200	1500	1800	2300	2900	3400	3900	4300	4800	5300	5800	6300	6800	7300
240	72.6	72.6	72.6	72.6	75.1	77.4	76.4	75.5	76.8	84.8	87.0	83.2	82.4	73.0	73.9	74.7
220	72.6	72.6	72.6	72.6	75.1	77.4	76.4	75.5	76.8	84.8	87.0	83.2	82.4	73.0	73.9	74.7
200	72.6	72.6	72.6	72.6	75.1	77.4	76.4	75.5	76.8	84.8	87.0	83.2	82.4	73.0	73.9	74.7
180	72.7	72.7	72.7	72.7	72.7	73.2	74.7	76.9	77.0	89.9	89.0	85.7	84.7	73.6	74.6	75.6
160	72.7	72.7	72.7	72.7	72.7	85.0	83.9	79.3	77.0	89.3	84.6	86.7	80.2	66.8	68.2	69.6
140	72.7	72.7	72.7	72.7	75.2	85.6	81.4	75.3	77.4	79.9	85.3	85.1	80.7	74.7	70.2	76.5
130	73.7	72.7	72.7	72.7	78.4	81.1	81.4	80.5	84.4	94.4	96.9	87.2	86.5	76.7	75.0	76.5
110	72.7	72.7	76.6	80.4	77.8	79.9	79.8	78.3	82.7	97.2	95.5	95.7	87.2	75.9	73.6	76.5
100	67.2	75.9	81.2	87.9	83.6	83.1	82.0	82.0	83.7	94.1	90.5	82.7	79.0	74.1	68.9	76.5
85	61.8	80.6	74.7	83.8	80.7	82.3	81.1	78.4	85.0	94.9	100.0	85.4	73.5	78.3	78.5	80.8
75	69.3	79.3	79.8	79.9	78.5	77.3	76.5	74.7	85.9	97.8	100.0	83.5	87.2	87.6	78.7	80.4
65	59.4	76.2	74.4	73.9	74.7	75.5	74.7	69.2	80.0	91.4	100.0	89.5	88.0	83.4	73.0	74.6
55	59.4	71.7	70.5	67.2	67.9	70.8	68.8	64.8	75.9	90.1	93.2	98.9	89.3	68.2	66.2	71.9
40	55.0	47.2	61.5	56.4	54.7	58.2	56.5	52.6	59.5	82.8	81.7	93.5	76.9	51.8	55.3	59.7
30	52.9	43.8	48.1	32.6	31.6	36.3	35.2	37.0	45.0	61.6	66.5	63.8	62.4	41.3	45.2	54.4
20	45.0	45.0	45.0	25.2	25.0	25.0	25.0	25.0	26.3	38.1	44.0	36.8	33.7	34.6	46.9	46.6

The general way to tune the VE table is to go through all the cells which the engine will operate within and to adjust the VE percentage until the lambda gauge matches the value in the *Target Lambda Table* shown below and in TS found under *Fuel > Target Lambda*. For example, if the lambda gauge shows 1.1 and the target lambda for that engine state is 1.0, the corresponding VE cell needs to be increased by 10%. The target lambda table supplied with the Miata base map should be sufficient to start with but you can modify it later to make the engine run richer or leaner under certain conditions such as boost or highway cruising respectively.

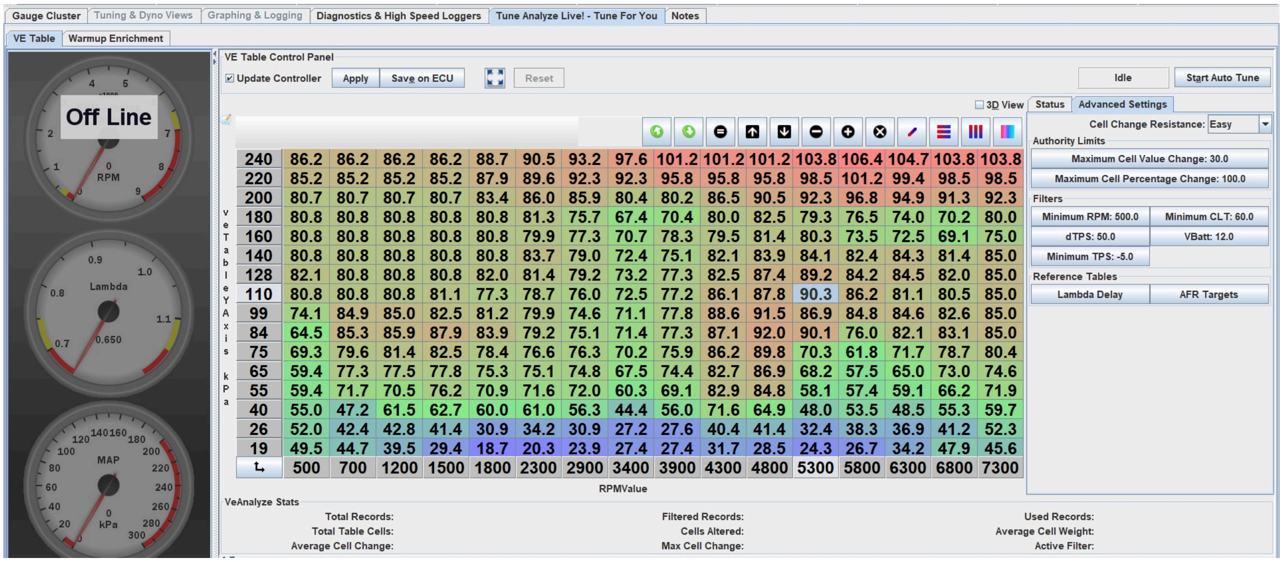


240	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
175	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
150	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
105	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
100	0.84	0.84	0.83	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
95	0.95	0.92	0.88	0.84	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
90	1.00	1.00	0.97	0.93	0.88	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
85	1.00	1.00	1.00	0.97	0.90	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
80	1.00	1.00	1.00	1.00	0.91	0.91	0.91	0.91	0.91	0.91	0.87	0.87	0.87	0.87	0.87	0.87
70	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.93	0.93	0.91	0.90	0.89	0.89
55	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.95	0.93	0.93
50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
45	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
40	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
35	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
	500	800	1150	1500	1900	2300	2500	2700	3000	3700	4400	5100	5800	6500	7000	7500

There are three ways of tuning the VE table. The first way is to drive the car around smoothly as a mate in the passenger seat goes through and changes the VE values until the lambda gauge matches the target lambda. The second and easier way is to use the TS autotuner which is only available in the full version of TS but absolutely worth it. To tune this way, you first need to disable some parameters. Under *Fuel*, open *Closed loop fuel correction* and *Deceleration fuel cut off (DFCO)*, set them both to false and click *burn* with the engine off. The third and easiest (yet most expensive option) is to take the car to a dyno for tuning where they will do either the first or second option themselves. The advantage of a dyno is that they can set it to bring the engine into any state they wish to perfectly configure the VE table.



Next, click the tab labelled *Tune Analyze Live! - Tune For You* to bring up the autotuner. Click to the *Advanced Settings* tab and configure it as shown in the image below. These configuration settings are deliberately quite loose so that TS can quickly tune the general shape of the VE table. On the left side of the *VE Table Control Panel*, you also need to check the box marked *Update Controller* which ensures that the VE table is updated on the ECU as the autotune corrects itself.



Now that the autotuner is set up, start the car and click *Start Auto Tune* on the autotuner. Let the car idle in park whilst it gets up to the minimum temperature. While this happens, you can attempt to change the idle cells in the VE table to get them to a lambda of 1. Once the car has warmed up, **smoothly** drive it around going through the gears and all the way through the rev range. A mix of flat, uphill and downhill driving in different gears is optimal to tune the majority of the engine's operating range. After you are sufficiently happy, click *Stop Auto Tune*, turn the engine off and click *Save on ECU*. You will want to repeat this process several more times, every time dropping the *Cell Change Resistance* and *Authority Limits* to slowly refine your VE table.

When you are satisfied with your VE table, turn closed loop fuel correction and *Deceleration fuel cut off (DFCO)* back to true. You don't actually need DFCO to be enabled although it will save fuel by turning the injectors off when the car is rolling in gear. Your Miata should now be relatively safe to drive but this is only the start of the tuning journey. As you read through the more advanced guides in this wiki, you will learn about all of the different ways the ECU can be configured to improve the drivability and squeeze every drop of performance out of your Miata.

Where to Find Tunes

Tunes and logs uploaded by the community can be found on FOME's tuning portal.

Fundamentals of FOME tuning

The place to learn how to start tuning your FOME Hardware

Quick Start to fuel tuning

These instructions make some pretty serious assumptions:

1. You have a setup that isn't going to grenade if you get things wrong for a few seconds
2. You know to lift off the throttle quickly if things are wrong
3. Your ignition maps are safe. THIS IS A BIG ASSUMPTION.
4. These instructions are oriented towards a naturally aspirated engine. Tuning a boosted engine this way is probably too risky, although it will work. Consider popping wastegate open with shop air, zip tie-ing bypass valve, popping a charge pipe (which has its own risks) or similar method to limit boost to minimal levels

> **DANGER**
>
> If you are not sure about your ignition maps, have a high-strung engine likely to blow up and hurt things, DO NOT DO THIS. These instructions assume a certain degree of forgiveness, which all engines are not necessarily going to give you.

Roughing in a fuel/VE map:

1. Estimate maximum VE value. For something like a 70s pushrod V8, this is probably 85-90%. For a modern overhead cam 4 valve head, this is probably more like 95%. For a very modern engine with VVT and high quality heads, this could be as high as 100%.

2. Set VE table to expected maximum value for loads 90kpa - 100kpa. set target AFR to something safe and THE SAME ACROSS THE BOARD i.e. 0.8 lambda everywhere
3. run engine hard, log AFR. We're talking Wide Open Throttle (WOT) operation
4. adjust injector size so that target AFR at peak torque (VE) is achieved. You will end up with lots of areas too rich and one narrow area in the middle of the powerband where AFR is just right
5. adjust VE at high load in order to achieve uniform AFR at WOT. This will mean decreasing VE as you move away from peak torque, where VE will be highest.
6. set entire table vertically (i.e. same RPM different load) as WOT VE
7. Pick a couple of part throttle load points. Adjust VE at part throttle load points to achieve target AFR. Apply same correction to all RPM points at chosen load.
8. Interpolate from chosen VE points vertically to WOT (or the next highest VE point you tuned)
9. extrapolate trends to lowest throttle loads
10. Set more reasonable target AFRs (i.e. 1.0 lambda at part throttle, 0.85 lambda at power desired)
11. adjust injector latency to achieve reasonable idle AFRs
12. revisit part throttle VE after latency changes with the expectation of small change in the opposite direction as latency changes. i.e. if you increased latency, expect to decrease VE but only a very small amount

Air Fuel Ratio

Air fuel ratio or AFR refers to the mass ratio of air to fuel involved in a combustion cycle. The AFR is important as the amount of fuel injected into the engine is one of the most significant combustion parameters that the ECU can control. The ECU takes the target AFR and determines the correct mass of fuel to inject based on the mass of air approximated using the temperature and pressure.

Why AFR Matters

An engine operates most efficiently and cleanly when the air-fuel ratio is at a specific value called the stoichiometric ratio. This ratio depends on the type of fuel being used, but for gasoline, it is approximately 14.7 parts air to 1 part fuel (14.7:1). When the air-fuel ratio is at the stoichiometric value, all of the fuel is burned, and there is no excess oxygen or unburned fuel left in the exhaust. This results in the least amount of emissions and the highest fuel efficiency. If the air-fuel ratio is too lean (excess air), there is not enough fuel to burn, and the engine may misfire or stall. If the air-fuel ratio is too rich (excess fuel), there is not enough oxygen to burn all the fuel, and the engine may emit more pollutants, have reduced fuel efficiency, and may even cause damage to the engine over time. Therefore, maintaining the proper air-fuel ratio is essential for optimal engine performance, fuel efficiency, exhaust gas temperature, engine knock and emission control in a car engine.

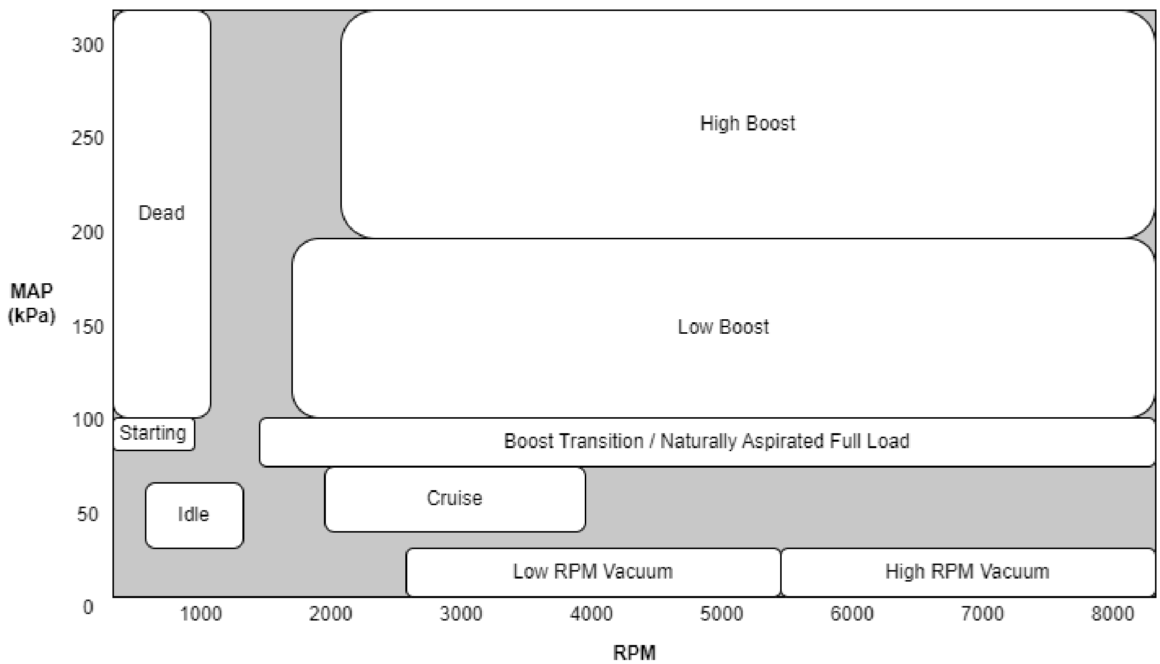
What is Lambda and Why it is a Superior Metric

Lambda, is a dimensionless ratio of the actual air-fuel ratio to the stoichiometric air-fuel ratio. In other words, it is the ratio of the AFR to the stoichiometric AFR (or the measured AFR divided by the stoichiometric AFR). Lambda is a more universal measure of the air-fuel ratio, as it is not affected by the specific fuel being used. The stoichiometric lambda value for any fuel is always 1.0, regardless of the fuel type. For example, if the actual AFR in an engine is 14.7:1 (stoichiometric AFR), then the lambda value is 1.0. If the actual AFR is leaner than 14.7:1, then the lambda value is greater than 1.0, and if it is richer than 14.7:1, then the lambda value is less than 1.0.

Lambda is preferred in engine tuning because it allows for a more precise control of the air-fuel ratio across different fuels and is generally easier to comprehend. For example, if a gasoline car is running 10% lean, the AFR would be 16.17 and lambda would be 1.1. If the car is 10% rich, AFR would be 13.36 and lambda would be 0.9. Looking at lambda, it is instantly obvious what percentage rich or lean the engine is running but with AFR, it requires more effort.

AFR Targets - When to Run Rich, Lean and Stoich

The ideal AFR targets will vary for every engine however there are guidelines for what the targets should be for each operating zone of the engine. These targets will be represented on an AFR target table or map, shown below, which characterizes the various engine operating conditions for their respective engine RPM and MAP. Generally speaking, running richer will decrease engine response at a gain of extra combustion chamber cooling and slightly higher power to a point. Inversely, running leaner will increase engine response at a loss running hotter and reducing power.



Idle and Cruising

For idle, a lambda of 1.0 is generally recommended to achieve a stable idle. When cruising, a lambda of 1.0 is also recommended however this can be raised up to about 1.05 to improve the fuel efficiency of the vehicle on the freeway or traveling a constant speed for long periods of time.

Low and High Load Vacuum

In the low vacuum section of the map, the engine will only operate here when the engine is under minimum load such as rolling in gear with the throttle closed. To save fuel, the engine can be operated up to about 1.05 lambda here or Deceleration Fuel Cutoff (DFCO) can be enabled to disable to injectors entirely and let the vehicle engine brake. DFCO is found under the *Fuel* tab in Tuner Studio.

The high vacuum part of the map is typically only used in the short period between high RPM gear changes or throttle lifts. The engine is usually being driven hard if this part of the map is being used so a target lambda of 0.95 to help cool the cylinders is recommended although a value of 1.0 is also acceptable.

High Load Naturally Aspirated/Boost Transition Zone

For a naturally aspirated (NA) engine, this zone represents the peak operating load which the engine will be placed under. A lambda of about 0.9 is recommended to balance performance with cylinder cooling.

For a forced induction engine, this zone represents the engine's transition into boost. As the engine usually isn't under a lot of load here, a slightly higher lambda of 0.95 is recommended to balance the engine response with some degree cylinder cooling.

Medium and High Boost Zones

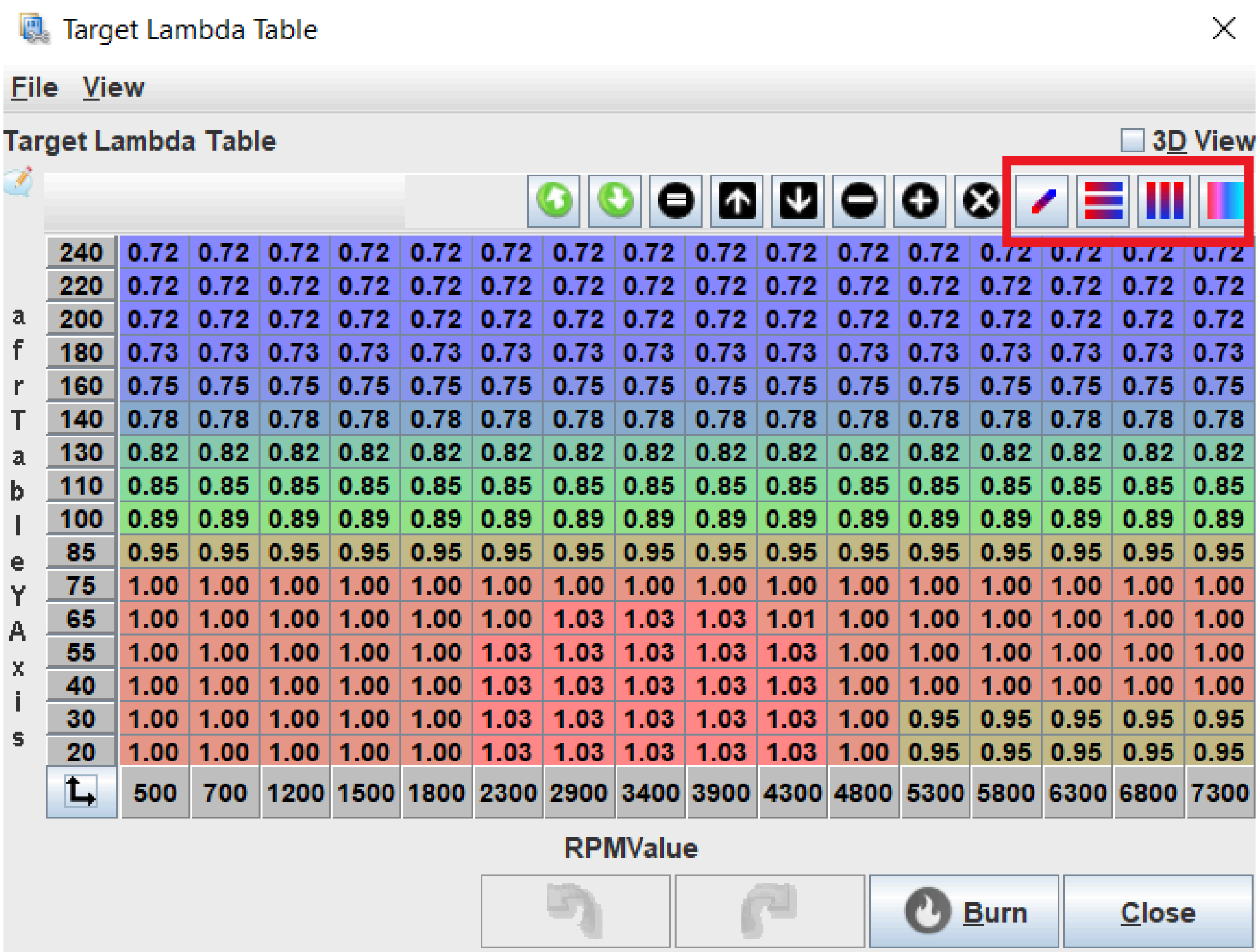
When the engine moves into boost, the engine load increases as does the temperature and pressure of the combustion. Hence, as the boost pressure increases, the AFR needs to get progressively richer. A good starting point for about 200kPa MAP or 14.5PSI of boost is a lambda of 0.78-0.82. For 300kPa MAP or 29PSI of boost, a 0.76-0.8 is generally a good starting range. Of course, every engine will differ so it is important here to research what others have successfully run on similar platforms to you.

Engine Start and Dead Zone

In both of these zones, their target AFRs do not matter a whole lot. The dead zone will never be operated in and the starting zone will never operate with closed loop fuelling as the lambda sensor will only activate after the car is running. The best configuration for these zones it to copy or transition them from the target AFR columns directly next to them for the sake of smoothness in the map.

Merging the zones

On the AFR target diagram, certain operating zones have missing values. To properly select targets for these zones, it's recommended to interpolate and smooth out the values between the defined sections of the map. It's important to create a smooth AFR target map that avoids abrupt changes, as the engine requires gradual variations in AFR to function optimally. To smooth the map in Tuner Studio there are four buttons in the map shown below. From left to right, these buttons interpolate across the entire selected zone, interpolate horizontally only, interpolate vertically only, and smooths out changes between selected cells.



afrTableYAxis	500	700	1200	1500	1800	2300	2900	3400	3900	4300	4800	5300	5800	6300	6800	7300
240	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
220	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
200	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72	0.72
180	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
160	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75	0.75
140	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
130	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82	0.82
110	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
100	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89	0.89
85	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
75	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
65	1.00	1.00	1.00	1.00	1.00	1.00	1.03	1.03	1.03	1.01	1.00	1.00	1.00	1.00	1.00	1.00
55	1.00	1.00	1.00	1.00	1.00	1.03	1.03	1.03	1.03	1.03	1.00	1.00	1.00	1.00	1.00	1.00
40	1.00	1.00	1.00	1.00	1.00	1.03	1.03	1.03	1.03	1.03	1.00	1.00	1.00	1.00	1.00	1.00
30	1.00	1.00	1.00	1.00	1.00	1.03	1.03	1.03	1.03	1.03	1.00	0.95	0.95	0.95	0.95	0.95
20	1.00	1.00	1.00	1.00	1.00	1.03	1.03	1.03	1.03	1.03	1.00	0.95	0.95	0.95	0.95	0.95

AFR For Different Fuels

Fundamentally, an oxygen sensor works in lambda. It measures the oxygen content in the exhaust relative to the open air and outputs a voltage which the ECU or wideband controller can directly convert to lambda. The ECU then converts this to AFR if required by multiplying the lambda by the stoichiometric value of the fuel (typically 14.7 for unleaded). Regardless of the fuel, the oxygen sensor will read the same lambda for any fuel that is burning at its stoichiometric point. A table is shown below comparing the stoichiometric AFR values of common fuels.

Fuel Type	Stoichiometric AFR
Unleaded Gasoline	14.7
E85	9.76
E100	8.98
Diesel	14.5
Methanol	6.46

Acceleration Compensation

What is acceleration compensation and why do I need it?

What types of acceleration compensation are there?

What is charge temperature estimation

Charge temp estimation is a method of predicting the temperature of air moving past the intake valve into the cylinder from less than perfect measurements elsewhere. The temperature of the air entering the cylinder is what matters most for engine management but it is virtually impossible to measure because sensors can't be installed in the cylinder head and they take time to get a stable reading. Charge temp estimation seeks to make a better estimate than reading any one sensor present on the engine. Why not just read the temperature from an air temp sensor in a MAF or charge pipe? The air moving into the engine picks up heat from the engine bay, intake manifold, cylinder head before it actually enters the engine. This additional heat that the air soaks up isn't measured by the sensor. Charge temp estimation tries to mathematically model this extra heat soak to provide a better answer than the sensor alone can provide.

On a practical level, charge temp estimation works by taking an average of two sensors: air temperature (usually measuring the temperature of air entering the engine) and coolant temperature (measuring the temperature of the engine contributing to heat soak). Charge temp estimation is tuned by changing the bias, or how much each sensor contributes. At low load, less air is moving slowly into the engine. It has more opportunity to soak up heat from a hot engine and there are fewer moles of air to spread the heat out between. At higher engine loads, air velocity is higher and there are more moles of air moving by. There is less time for air to soak up heat and there are more moles to divide up heat. Generally, at lower loads coolant temperature has a higher bias to reflect air soaking up more heat from the engine. Generally, at higher airflows, air temperature has a higher bias because it has less time to soak up temperature from a hot engine.

Tuning a charge air temp table properly is a hard problem without heavy instrumentation and an engine dyno. However, understanding the basics of how it operates can help poke it in one direction or another. Generally, when air temp changes produce odd or inconsistent air fuel ratio changes the charge estimation model can be a useful tool to make more consistent operation.

What is knock and why it matters

Knock is rapid, uncontrolled combustion which results in a correspondingly extremely fast increase in cylinder pressure. Knock is often accompanied by extremely high instantaneous combustion temperatures. Knock earned its name from the knocking or rattling noise that is audible from pressure waves smacking pieces of engine.

Knock matters because it breaks things. Peak cylinder pressures during knocking can be tens to hundreds of times higher than normal controlled combustion. The violent changes in pressure almost always exceed design parameters of the engine. Things break - headgaskets, pistons, connecting rods, crankshafts, rodbolts, headstuds, more. The intense heat associated with uncontrolled combustion can torch valves, melt holes in pistons and cylinder heads. Knocking at extremely light loads can sometimes be survived but it is a good idea to not operate engines while knocking.

Knock is typically detected by in-cylinder pressure transducers, ion sensing or acoustic sensors.

It is the job of engine management to take corrective action when knock is detected. This usually takes the form of retarding ignition timing.

Multi-Dimensional Mapping, what is it and how to benefit from it

This page is under construction

Multi-dimensional mapping is a new feature that has been recently introduced to FOME.
It is a method of overlaying multiple compensation maps over the simple Load vs RPM ignition and fuel table.
The term Multi-dimensional comes from tha fact that each compensation table adds an extra dimension to the basic 3d map.
In FOME you are currently able to activate up to 4 additional compensation maps, each with it's own configurable blend curve.

The purpose of these tables is to be able to configure an additional input that changes the fuelling or timing, for instance to compensate for things like Ethanol content, Different fuel octane rating or simple variable valve timing.

How it works

FOME allows the following inputs to be used as the input for each of the compensation maps:
TPS,
MAP,
CLT,
IAT,
Fuel Load,
Ignition Load,
Aux Temp 1,
Aux Temp 2,
Accelerator Pedal Position,
Battery Voltage,
VVT Position for bank 1 or 2 intake and exhaust,
Ethanol % sensed from the Flex Fuel,
Aux Linear Sensors 1 and 2,
GPPWM Outputs 1 to 4,
Lua outputs 1 and 2, RPM,
Detected Gear,

All of the above can be considered the additional dimensions the system is able to be mapped for once selected.
These are selectable in the four "blend config" tables.

Once one of these inputs is chosen in the Compensation table a blend table will become available.
This table is a 0 to 100% table that chooses how much of the adjustment in the compensation will be applied to the main table for each

combination of the dimension and RPM.

Spark timing, MBT and combustion

What to put into this section

Use this folder for covering the overall topics of why things are needed, this is the place to talk about things like why an engine needs certain AFRs, why timing needs to be reduced, what is knock, why do you need to add fuel on acceleration etc

Leave the details of how FOME works for each feature to the specific pages, when relevant we can link the firmware specific page and visavis, keep this and overview as a nuggets intro area

This can likely kill any dumb questions at source and prevent many user issues down the line

Sensors and gauges

Information on all sensor inputs and data outputs

General sensors

9 items

Temperature sensors

3 items

Pressure sensors

3 items

Fuel sensors

3 items

🗃️ Driver controls and feedback

7 items

General sensors

General sensors

📄 Analogue Input Settings

ADC V-Ref

📄 AUX Sensor Inputs

Aux Speed Sensor

📄 Camshaft position

📄 Crankshaft Position

📄 ETB Throttle Position

📄 Mass Air Flow Meter

📄 Throttle position

Throttle position sensors are used to tell the FOME ECU the angle of the throttle blade. Most are 3 wire:

📄 Turbo Speed Sensors

📄 Vehicle Speed Sensor

Notes on setting VSS

Analogue Input Settings

ADC V-Ref

Analogue Divider

Smoothing Factor

AUX Sensor Inputs

Aux Speed Sensor

Aux linear sensor

Aux sensor serial comms settings

Camshaft position

Crankshaft Position

ETB Throttle Position

Mass Air Flow Meter

Throttle position

Throttle position sensors are used to tell the FOME ECU the angle of the throttle blade. Most are 3 wire:

1: Power (Provided from FOME ECU Power Output) 2: Ground (provided from FOME ECU Ground Output) 3: Signal (sent to FOME ECU Lowside input)

*ensuring power and ground come from the FOME ECU will make sure the signal is clean and consistent)

For vehicles with electric throttle bodies, throttle position sensors will need to always have two redundant signals (usually opposite ie 0-5v and 5-0v). This is to ensure safety of the operation of the vehicle.

Turbo Speed Sensors

Vehicle Speed Sensor

Notes on setting VSS

When setting VSS up calculate correct wheel rotations per km, use correct final drive ratio and then use the speed sensor gear ratio as a tuning factor to achieve correct speed reading.

Temperature sensors

Temperature sensors

📄 **Auxillary Temperature Inputs**

📄 **Coolant Temperature**

📄 **Intake Air Temperature**

Auxillary
Temperature Inputs

Coolant Temperature

Intake Air Temperature

Pressure sensors

Pressure sensors

📄 Barometric Pressure Sensing

📄 Manifold Absolute Pressure

MAP sensors

📄 Oil Pressure Sensors

Barometric Pressure Sensing

Manifold Absolute Pressure

MAP sensors

MAP settings

MAP sampling angle - See MAP sampling Angle page

Oil Pressure Sensors

Fuel sensors

Fuel sensors

📄 Flex Fuel sensors

Flex fuel sensors measure the amount of ethanol content in the fuel passing through it. The flex fuel sensor has a pre calibrated signal that is sent fro...

📄 Fuel Level Sensing

📄 Fuel Pressure Sensors

Flex Fuel sensors

Flex fuel sensors measure the amount of ethanol content in the fuel passing through it. The flex fuel sensor has a pre calibrated signal that is sent from the sensor in a 0-5v voltage indicating which amount of ethanol is present. The FOME ECU has the ability to use this data. The signal circuit carries the ethanol percentage via the frequency signal.

The normal range of operating frequency is between 50-150Hz. The microprocessor inside the sensor is capable of a certain amount of self-diagnosis. An output frequency between 180Hz and 190Hz indicates that the fuel is contaminated.

In TunerStudio you select which input pin you have used for the signal from the flex fuel sensor. You can view % of content using a gauge in TunerStudio.

Fuel Level Sensing

Fuel Pressure Sensors

Driver controls and feedback

Driver controls and feedback

📄 Accelerator pedal

📄 Battery Settings

📄 Brake position and pressure sensing

📄 BeerMoneyMotorsports KaN (CAN) Gauge

Product description and purchase link: BMM KaN Multi Fit CAN Gauge.

📄 **Clutch position sensing**

📄 **Rev Counter/Tachometer**

📄 **WBO2 Wideband Lambda Sensor**

Accelerator pedal

Battery Settings

Brake position and pressure sensing

BeerMoneyMotorsports KaN (CAN) Gauge

Product description and purchase link: BMM KaN Multi Fit CAN Gauge.

The KaN gauge is a multi function, multi fit gauge designed exclusively for the Beer Money Motorsports line of ECUs. The gauge is designed to fit into multiple locations on a Miata including the oil pressure gauge hole, an air conditioning vent or into a standard 52mm gauge pod. Data is sent to the gauge via CAN bus communication wires coming from the ECU and the layout of the gauge can be configured wirelessly using its standalone WiFi network. This article will detail the installation and setup of a gauge into your Miata.

Wiring

Open your ECU casing and locate a 5/12V output, ground, CAN High, and CAN Low pins. Match each of these pins to one of the four corresponding wires on the gauge. For best signal over long distances, twist the pair of CAN lines.

WiFi Configuration

When powered, the gauge will start its own standalone wifi network. In the WiFi settings for your device, connect to this network with a phone or laptop and open your browser of choice. Next find the IP address of your gauge. It should be a number like "192.168.X.XXX" which was included with your gauge purchase. In the browser's web address bar, input this IP address of the device into the browser and go to the web page.

The web page should detail the gauge settings to allow for customization of the theme and displayed values.

Physical Installation

Several options are available to mount the gauge on a Miata depending on where you like your gauges, how much you like air conditioning, and how much you like removing your dashboard.

Oil Pressure Gauge Mounting

This is by far the most neat and integrated installation spot. If installing the gauge here, it is recommended to also install an aftermarket oil pressure sensor to the ECU so that the gauge can display your oil pressure among other useful variables

Air Conditioning Vent Mounting

Another clean installation method is to replace an air conditioning eyeball vent on an NA/NB Miata with a 52mm gauge housing.

52mm Gauge Pod Mounting

With the 52mm housing option, the gauge can be mounted into any 52mm gauge pod housing which is the standard size for aftermarket automotive gauges. For Miatas specifically, there are a range of gauge pod mounts fitting to locations like the A pillar, in place of the radio console, on top of the gauge cluster, etc.

Bare PCB Mounting

Purchasing the gauge as a bare PCB allows you to design your own mounting solution. Mount the gauge anywhere from behind your steering wheel to inside a cup holder!

Clutch position sensing

Rev Counter/ Tachometer

WBO2 Wideband Lambda Sensor

Limits and Protections

Limiters, engine protections and safety features

📄 Boost pressure limit

Boost pressure limit can be found under Base Engine > Limits and fallbacks.

📄 Coolant Based RPM limit

📄 FOME Fuel and Ignition Cut Codes ("Clear Reason")

|Code|enum ClearReason|TunerStudio Text|Description|

📄 FOME OBD fault codes and how to read them

The FOME ECU is providing a number of OBD fault codes, they are either standard OBD fault codes or FOME custom fault codes.

📄 Fuel Pressure Compensation

📄 Minimum oil pressure protection

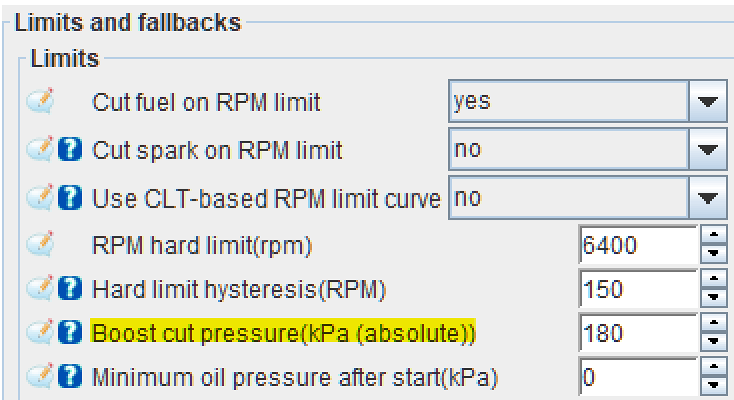
Minimum oil pressure after start

📄 Rev limiters

Rev limiters can be found under Base Engine > Limits and fallbacks.

Boost pressure limit

Boost pressure limit can be found under **Base Engine > Limits and fallbacks**.



Boost cut pressure

This setting defines the upper limit of pressure the engine will operate within. If the engine's MAP reading exceeds the value configured in **Boost cut pressure**, FOME will cut fuel injection until the MAP reading falls 20kPa below the configured value. The 20kPa hysteresis value is hard-coded.

This value is absolute pressure, so if an engine is running 7 PSIG of boost pressure (~148kPa absolute), this value would need to be set above 150kPa. It is a good idea to allow enough of a margin to prevent unnecessary boost cut on small excursions above the target boost pressure.

Set this value to 0 to disable the boost cut limit.

Coolant Based RPM limit

FOME Fuel and Ignition Cut Codes ("Clear Reason")

Code	`enum ClearReason`	**TunerStudio Text**	**Description**
0	`None`	None	no fuel/ignition cut
1	`Fatal`	Fatal Error	disabled due to fatal error
2	`Settings`	Setting Disabled	disabled by setting: `isInjectionEnabled`/`isIgnitionEnabled`
3	`HardLimit`	RPM Limit	disabled by hard (normal) RPM limit: `useCltBasedRpmLimit`/`cltRevLimitRpmBins`/`cltRevLimitRpm`/`rpmHardLimit`/`rpmHardLimitHyst`
4	`FaultRevLimit`	Fault RPM Limit	disabled by (fault) RPM limit -- ETB Problem: 1500, Fatal Error: 0
5	`BoostCut`	Boost Cut	disabled by boost cut pressure threshold: `boostCutPressure`
6	`OilPressure`	Oil Pressure	disabled due to low oil pressure after 5 seconds: `minOilPressureAfterStart`
7	`StopRequested`	Stop Requested	disabled due to engine stop
8	`EtbProblem`	ETB Problem	disabled due to detected ETB problem ("10 percent-seconds of integral error") -- **currently not implemented**
9	`LaunchCut`	Launch Control	disabled by launch control RPM retard condition: `launchRpm`/`launchFuelCutEnable`/`launchSparkCutEnable`/`launchControlEnable`
10	`InjectorDutyCycle`	Max Injector Duty	disabled due to injector duty cycle limits: `maxInjectorDutyInstant`/`maxInjectorDutySustained`/`maxInjectorDutySustainedTimeout` -- resets after falling below 20% duty
11	`FloodClear`	Flood Clear	disabled to allow flood clear mode: `isCylinderCleanupEnabled` -- TPS > 95% while cranking
12	`EnginePhase`	Engine Sync	disabled due to lacking engine-cam sync to avoid non-sync spark: `vvtMode[0]` -- ex: symmetrical crank (NB2, Nissan VQ/MR), uneven firing order (VTwin Harley)
13	`KickStart`	Kick Start	**currently not implemented**
14	`IgnitionOff`	Ignition Off	disabled because the ignition is off, exception: self-stimulation mode
15	`Lua`	Lua	fueling disabled by Lua script ignition cut request
16	`ACR`	ACR	disabled due to Harley Automatic Compression Release: `acrPin`/`acrRevolutions`/`acrDisablePhase` -- avoids spraying fuel everywhere

Code	`enum ClearReason`	TunerStudio Text	Description
17	`LambdaProtection`	Lambda Protection	disabled due to lambda protection: `lambdaProtectionTimeout`

FOME OBD fault codes and how to read them

The FOME ECU is providing a number of OBD fault codes, they are either standard OBD fault codes or FOME custom fault codes.

FOME specific custom codes are listed below

Meaning	Fault Code
CUSTOM_NAN_ENGINE_LOAD	6000
CUSTOM_WRONG_ALGORITHM	6001
CUSTOM_NAN_ENGINE_LOAD_2	6002
CUSTOM_INTEPOLATE_ERROR	6012
CUSTOM_INTEPOLATE_ERROR_2	6013
CUSTOM_INTEPOLATE_ERROR_3	6014

Meaning	Fault Code
CUSTOM_INTEPOLATE_ERROR_4	6015
CUSTOM_PARAM_RANGE	6016
CUSTOM_MAF_NEEDED	6017
CUSTOM_UNKNOWN_ALGORITHM	6018
CUSTOM_OBD_UNKNOWN_FIRING_ORDER	6023
CUSTOM_OBD_WRONG_FIRING_ORDER	6024
CUSTOM_OBD_IGNITION_MODE	6025
CUSTOM_UNEXPECTED_ENGINE_TYPE	6027
CUSTOM_INVALID_TPS_SETTING	6028
CUSTOM_OBD_NAN_INJECTION	6030
CUSTOM_OBD_NEG_INJECTION	6031
CUSTOM_ZERO_DWELL	6032
CUSTOM_DWELL_TOO_LONG	6033
CUSTOM_SKIPPING_STROKE	6034

Meaning	Fault Code
CUSTOM_OBD_ANALOG_INPUT_NOT_CONFIGURED	6038
CUSTOM_OBD_WRONG_ADC_MODE	6039
CUSTOM_OBD_KNOCK_PROCESSOR	6041
CUSTOM_OBD_LOCAL_FREEZE	6042
CUSTOM_LOGGING_BUFFER_OVERFLOW	6044
CUSTOM_OBD_PIN_CONFLICT	6048
CUSTOM_OBD_LOW_FREQUENCY	6049
CUSTOM_OBD_TS_PAGE_MISMATCH	6052
CUSTOM_OBD_TS_OUTPUT_MISMATCH	6053
CUSTOM_TOO_LONG_CRANKING_FUEL_INJECTION	6054
CUSTOM_INTERPOLATE_NAN	6055
ERROR_HISTO_NAME	6056
CUSTOM_OBD_HIGH_FREQUENCY	6058
CUSTOM_OBD_MMC_START1	6060

Meaning	Fault Code
CUSTOM_OBD_MMC_START2	6061
CUSTOM_PID_DTERM	6097
CUSTOM_DWELL	6098
CUSTOM_TS_OVERFLOW	6099
CUSTOM_ERR_OP_MODE	6100
CUSTOM_ERR_TRIGGER_ZERO	6101
CUSTOM_ERR_6102	6102
CUSTOM_ERR_2ND_WATCHDOG	6103
CUSTOM_ERR_INVALID_INJECTION_MODE	6104
CUSTOM_ERR_WAVE_1	6105
CUSTOM_ERR_WAVE_2	6106
CUSTOM_ERR_TEST_ERROR	6107
CUSTOM_ERR_IGNITION_MODE	6108
CUSTOM_ERR_CAN_CONFIGURATION	6109

Meaning	Fault Code
CUSTOM_ERR_INTERPOLATE	6110
CUSTOM_ERR_NOT_INITIALIZED_TRIGGER	6111
CUSTOM_ERR_MAP_TYPE	6112
CUSTOM_ERR_THERM	6113
CUSTOM_ERR_NATURAL_LOGARITHM_ERROR	6114
CUSTOM_ERR_LOOPED_QUEUE	6115
CUSTOM_ERR_PWM_1	6116
CUSTOM_ERR_PWM_2	6117
CUSTOM_ERR_DWELL_DURATION	6118
CUSTOM_ERR_NO_SHAPE	6119
CUSTOM_ERR_SGTP_ARGUMENT	6121
CUSTOM_ERR_INVALID_PIN	6130
CUSTOM_ERR_6131	6131
CUSTOM_ERR_UNKNOWN_PORT	6132

Meaning	Fault Code
CUSTOM_ERR_PIN_ALREADY_USED_1	6133
CUSTOM_ERR_PIN_ALREADY_USED_2	6134
CUSTOM_ERR_6135	6135
CUSTOM_ERR_TCHARGE_NOT_READY	6136
CUSTOM_ERR_TRIGGER_WAVEFORM_TOO_LONG	6137
CUSTOM_ERR_FUEL_TABLE_NOT_READY	6138
CUSTOM_ERR_TCHARGE_NOT_READY2	6139
CUSTOM_ERR_COMMAND_LOWER_CASE_EXPECTED	6140
CUSTOM_ERR_FLASH_CRC_FAILED	6141
CUSTOM_ERR_NOT_INPUT_PIN	6142
CUSTOM_ERR_SKIPPED_TOOTH_SHAPE	6143
CUSTOM_ERR_UNEXPECTED_SHAFT_EVENT	6144
CUSTOM_ERR_SD_MOUNT_FAILED	6145
CUSTOM_ERR_SD_SEEK_FAILED	6146

Meaning	Fault Code
CUSTOM_ERR_OUT_OF_ORDER	6147
CUSTOM_ERR_T2_CHARGE	6148
CUSTOM_ERR_ASSERT	6500
CUSTOM_ERR_ASSERT_VOID	6501
ERROR_FL_STACK_OVERFLOW	6502
CUSTOM_6503	6503
CUSTOM_FLSTACK	6504
CUSTOM_ERR_NAN_TCHARGE	6505
CUSTOM_EGO_TYPE	6506
CUSTOM_LIST_LOOP	6507
CUSTOM_ERR_LOCK_ISSUE	6508
CUSTOM_CONFIG_NOT_READY	6509
CUSTOM_ERR_TRG_ANGLE_ORDER	6510
CUSTOM_ERR_STATE_NULL	6511

Meaning	Fault Code
CUSTOM_ERR_SAME_ANGLE	6512
ERROR_TRIGGER_DRAMA	6513
CUSTOM_MAP_ANGLE_PARAM	6514
CUSTOM_ERR_DISPLAY_MODE	6515
CUSTOM_ERR_ADC_UNKNOWN_CHANNEL	6516
CUSTOM_ERR_ADC_USED	6517
CUSTOM_ERR_ADC_DEPTH_SLOW	6518
CUSTOM_ERR_ADC_DEPTH_FAST	6519
CUSTOM_ERR_6520	6520
CUSTOM_ERR_6521	6521
CUSTOM_ERR_6522	6522
CUSTOM_ERR_6523	6523
CUSTOM_ERR_UNEXPECTED_SPI	6524
CUSTOM_ERR_EXT_MODE	6525

Meaning	Fault Code
CUSTOM_ERR_TIMER_OVERFLOW	6526
CUSTOM_ERR_6527	6527
CUSTOM_ERR_SCHEDULING_ERROR	6528
CUSTOM_ERR_LOGGING_NOT_READY	6529
ERROR_NAN_FIND_INDEX	6530
ERROR_NULL_BUFFER	6531
CUSTOM_ERR_BUFF_INIT_ERROR	6532
CUSTOM_ERR_INTERPOLATE_PARAM	6533
ERROR_LOGGING_SIZE_CALC	6534
CUSTOM_ERR_ADC_CHANNEL	6535
CUSTOM_ERR_ANGLE	6536
CUSTOM_ERR_LOGGING_NULL	6537
CUSTOM_ERR_PARSING_ERROR	6538
CUSTOM_ERR_INJECTOR_LAG	6539

Meaning	Fault Code
CUSTOM_ERR_AXIS_ORDER	6540
CUSTOM_HW_TIMER	6541
CUSTOM_INJ_DURATION	6542
CUSTOM_ADD_BASE	6543
CUSTOM_ERR_6544	6544
CUSTOM_ERR_6545	6545
CUSTOM_UNEXPECTED_TDC_ANGLE	6546
CUSTOM_INVALID_GLOBAL_OFFSET	6547
CUSTOM_UNEXPECTED_MAP_VALUE	6548
CUSTOM_ERR_6549	6549
CUSTOM_ERR_6550	6550
CUSTOM_TRIGGER_SYNC_ANGLE	6551
CUSTOM_TRIGGER_SYNC_ANGLE2	6552
CUSTOM_ERR_6553	6553

Meaning	Fault Code
CUSTOM_ERR_6554	6554
CUSTOM_ERR_6555	6555
CUSTOM_ERR_6556	6556
CUSTOM_ERR_6557	6557
CUSTOM_ERR_6558	6558
CUSTOM_TRIGGER_SYNC_ANGLE_RANGE	6559
CUSTOM_ERR_TRIGGER_ANGLE_RANGE	6560
CUSTOM_ERR_6561	6561
CUSTOM_ERR_6562	6562
CUSTOM_ERR_6563	6563
CUSTOM_ERR_6564	6564
CUSTOM_ERR_6565	6565
CUSTOM_ERR_6566	6566
CUSTOM_ERR_6567	6567

Meaning	Fault Code
CUSTOM_ERR_6568	6568
CUSTOM_ERR_6569	6569
CUSTOM_ERR_6570	6570
CUSTOM_ERR_6571	6571
CUSTOM_ERR_ARRAY_REMOVE	6572
CUSTOM_ERR_6573	6573
CUSTOM_ERR_6574	6574
CUSTOM_ERR_6575	6575
CUSTOM_ERR_6576	6576
CUSTOM_ERR_6577	6577
CUSTOM_NULL_ENGINE_PTR	6578
CUSTOM_DUTY_TOO_LOW	6579
CUSTOM_ERR_6580	6580
CUSTOM_ERR_6581	6581

Meaning	Fault Code
CUSTOM_ERR_6582	6582
CUSTOM_ERR_6583	6583
CUSTOM_ERR_6584	6584
CUSTOM_ERR_6585	6585
CUSTOM_ERR_6586	6586
CUSTOM_ERR_6587	6587
CUSTOM_NULL_SHAPE	6588
CUSTOM_SPARK_ANGLE_1	6589
CUSTOM_ERR_6590	6590
CUSTOM_ERR_6591	6591
CUSTOM_ERR_6592	6592
CUSTOM_ERR_6593	6593
CUSTOM_SHAPE_LEN_ZERO	6594
CUSTOM_TRIGGER_CYCLE	6595

Meaning	Fault Code
CUSTOM_TRIGGER_CYCLE_NAN	6596
CUSTOM_OMODE_UNDEF	6597
CUSTOM_ERR_6598	6598
CUSTOM_ERR_6599	6599
CUSTOM_ERR_6600	6600
CUSTOM_CONSOLE_TOO_MANY	6601
CUSTOM_APPEND_NULL	6602
CUSTOM_ERR_6603	6603
CUSTOM_ERR_6604	6604
CUSTOM_ERR_6605	6605
CUSTOM_ERR_6606	6606
CUSTOM_APPEND_STACK	6607
CUSTOM_ERR_6608	6608
CUSTOM_ERR_6609	6609

Meaning	Fault Code
CUSTOM_ERR_6610	6610
CUSTOM_ERR_6611	6611
CUSTOM_ERR_6612	6612
CUSTOM_ERR_6613	6613
CUSTOM_ERR_6614	6614
CUSTOM_ERR_6615	6615
CUSTOM_ERR_6616	6616
CUSTOM_ERR_TIMER_STATE	6617
CUSTOM_ERR_6618	6618
CUSTOM_ERR_6619	6619
CUSTOM_APPLY_STACK	6620
CUSTOM_ERR_6621	6621
CUSTOM_ERR_6622	6622
CUSTOM_ERR_6623	6623

Meaning	Fault Code
CUSTOM_ERR_6624	6624
CUSTOM_ERR_6625	6625
CUSTOM_EVENT_6626	6626
CUSTOM_STACK_6627	6627
CUSTOM_ERR_6628	6628
CUSTOM_STACK_6629	6629
CUSTOM_ERR_6030	6630
CUSTOM_ERR_6631	6631
CUSTOM_ERR_6632	6632
CUSTOM_ANGLE_NAN	6633
CUSTOM_ERR_6634	6634
CUSTOM_ERR_6635	6635
CUSTOM_ERR_6636	6636
CUSTOM_CONF_NULL	6637

Meaning	Fault Code
CUSTOM_TRIGGER_EVENT_TYPE	6638
CUSTOM_ERR_6639	6639
CUSTOM_TRIGGER_UNEXPECTED	6640
CUSTOM_ERR_6641	6641
CUSTOM_TRIGGER_STACK	6642
CUSTOM_ERR_6643	6643
CUSTOM_IDLE_WAVE_CNT	6644
CUSTOM_ERR_6645	6645
CUSTOM_ERR_6646	6646
CUSTOM_ERR_6647	6647
CUSTOM_ERR_6648	6648
CUSTOM_ERR_6649	6649
CUSTOM_ERR_6650	6650
CUSTOM_ERR_6651	6651

Meaning	Fault Code
CUSTOM_ERR_6652	6652
CUSTOM_ERR_6653	6653
CUSTOM_ERR_6654	6654
CUSTOM_ERR_6655	6655
CUSTOM_ERR_6656	6656
CUSTOM_ERR_6657	6657
CUSTOM_ERR_6658	6658
CUSTOM_ERR_6659	6659
CUSTOM_ERR_6660	6660
CUSTOM_ERR_6661	6661
CUSTOM_ERR_6662	6662
CUSTOM_ERR_6663	6663
CUSTOM_ERR_6664	6664
CUSTOM_ERR_6665	6665

Meaning	Fault Code
CUSTOM_ERR_6666	6666
CUSTOM_ERR_ADCANCE_CALC_ANGLE	6667
CUSTOM_ERR_ETB_TARGET	6668
CUSTOM_ERR_6669	6669
CUSTOM_ERR_6670	6670
CUSTOM_STACK_ADC_6671	6671
CUSTOM_ERR_6672	6672
CUSTOM_ERR_6673	6673
CUSTOM_STACK_SPI	6674
CUSTOM_VVT_SYNC_POSITION	6675
CUSTOM_STACK_ADC	6676
CUSTOM_IH_STACK	6677
CUSTOM_ERR_6678	6678
CUSTOM_ERR6679	6679

Meaning	Fault Code
CUSTOM_ERR_ANGLE_CR	6680
CUSTOM_DELTA_NOT_POSITIVE	6681
CUSTOM_TIMER_WATCHDOG	6682
CUSTOM_SAME_TWICE	6683
CUSTOM_ERR_6684	6684
CUSTOM_ERR_6685	6685
CUSTOM_ERR_6686	6686
CUSTOM_FIRING_LENGTH	6687
CUSTOM_ADVANCE_SPARK	6688
CUSTOM_ERR_6689	6689
CUSTOM_ERR_MAP_START_ASSERT	6690
CUSTOM_ERR_MAP_AVG_OFFSET	6691
CUSTOM_ERR_MAP_CYL_OFFSET	6692
CUSTOM_ERR_PWM_DUTY_ASSERT	6693

Meaning	Fault Code
CUSTOM_ERR_ZERO_CRANKING_FUEL	6694
CUSTOM_NULL_EXECUTOR	6695
CUSTOM_SLOW_NOT_INVOKED	6696
CUSTOM_PWM_CYCLE_START	6697
CUSTOM_ERR_ARRAY_IS_FULL	6698
CUSTOM_ERR_ARRAY_REMOVE_ERROR	6699
CUSTOM_ERR_6700	6700
CUSTOM_CJ125_0	6700
CUSTOM_CJ125_1	6701
CUSTOM_CJ125_2	6702
CUSTOM_ERR_BENCH_PARAM	6703
CUSTOM_ERR_BOTH_FRONTS_REQUIRED	6704
CUSTOM_TLE8888	6705
CUSTOM_KNOCK_WINDOW	6706

Meaning	Fault Code
CUSTOM_ERR_TIMER_TEST_CALLBACK_NOT_HAPPENED	6707
CUSTOM_ERR_TIMER_TEST_CALLBACK_WRONG_TIME	6708
CUSTOM_ERR_6709	6709
CUSTOM_DUTY_INVALID	6710
CUSTOM_PWM_DUTY_TOO_HIGH	6711
CUSTOM_ERR_PWM_STATE_ASSERT	6712
CUSTOM_ERR_PWM_CALLBACK_ASSERT	6713
CUSTOM_ERR_PWM_SWITCH_ASSERT	6714
CUSTOM_ERR_ZERO_E0_MULT	6715
CUSTOM_ERR_ZERO_E85_MULT	6716
CUSTOM_INVALID_ADC	6720
CUSTOM_INVALID_MODE_SETTING	6721
CUSTOM_ERR_TASK_TIMER_OVERFLOW	6722
CUSTOM_NO_ETB_FOR_IDLE	6723

Meaning	Fault Code
CUSTOM_ERR_TLE8888_RESPONSE	6724
CUSTOM_ERR_CJ125_DIAG	6725
CUSTOM_6726	6726
CUSTOM_VVT_MODE_NOT_SELECTED	6727
CUSTOM_ERR_6728	6728
CUSTOM_ARTIFICIAL_MISFIRE	6729
CUSTOM_INSTANT_MAP_DECODING	6899
STACK_USAGE_COMMUNICATION	6900
STACK_USAGE_MIL	6901
CUSTOM_6902	6902
STACK_USAGE_STATUS	6903
STACK_USAGE_4	6904
CUSTOM_OBD_MMC_ERROR	8000
CUSTOM_ERR_CAN_COMMUNICATION	8900

Meaning	Fault Code
CUSTOM_8901	8901
CUSTOM_ERR_CUSTOM_GAPS_BAD	8999
CUSTOM_ERR_TRIGGER_SYNC	9000
CUSTOM_OBD_TRIGGER_WAVEFORM	9001
CUSTOM_PRIMARY_TOO_MANY_TEETH	9002
CUSTOM_PRIMARY_NOT_ENOUGH_TEETH	9003
CUSTOM_CAM_TOO_MANY_TEETH	9004
CUSTOM_CAM_NOT_ENOUGH_TEETH	9005
CUSTOM_PRIMARY_DOUBLED_EDGE	9006
CUSTOM_PRIMARY_BAD_TOOTH_TIMING	9007
CUSTOM_OBD_SKIPPED_SPARK	9009
CUSTOM_OBD_SKIPPED_FUEL	9010
CUSTOM_RE_ADDING_INTO_EXECUTION_QUEUE	9011
CUSTOM_OUT_OF_ORDER_COIL	9012

Meaning	Fault Code
CUSTOM_TOO_LONG_FUEL_INJECTION	9013

Standard OBD fault codes used by FOME

the meanings of standard OBD codes are available on http//:www.obd-codes.com/faq/obd2-codes-explained.php

Meaning	Fault Code
OBD_Fuel_Pressure_Sensor_Missing	90
OBD_Mass_or_Volume_Air_Flow_Circuit_Malfunction	100
OBD_Manifold_Absolute_Pressure_Circuit_Malfunction	105
OBD_Map_Timeout	106
OBD_Map_Low	107
OBD_Map_High	108
OBD_ThermistorConfig	111

Meaning	Fault Code
OBD_Iat_Timeout	110
OBD_Iat_Low	112
OBD_Iat_High	113
OBD_Clt_Timeout	115
OBD_Clt_Low	117
OBD_Clt_High	118
OBD_TPS_Configuration	121
OBD_TPS1_Primary_Timeout	120
OBD_TPS1_Primary_Low	122
OBD_TPS1_Primary_High	123
OBD_FlexSensor_Timeout	176
OBD_FlexSensor_Low	178
OBD_FlexSensor_High	179

Meaning	Fault Code
OBD_Injector_Circuit_1	201
OBD_Injector_Circuit_2	202
OBD_Injector_Circuit_3	203
OBD_Injector_Circuit_4	204
OBD_Injector_Circuit_5	205
OBD_Injector_Circuit_6	206
OBD_Injector_Circuit_7	207
OBD_Injector_Circuit_8	208
OBD_Injector_Circuit_9	209
OBD_Injector_Circuit_10	210
OBD_Injector_Circuit_11	211
OBD_Injector_Circuit_12	212
OBD_TPS1_Secondary_Timeout	220

Meaning	Fault Code
OBD_TPS1_Secondary_Low	222
OBD_TPS1_Secondary_High	223
OBD_TPS2_Primary_Timeout	225
OBD_TPS2_Primary_Low	227
OBD_TPS2_Primary_High	228
OBD_Crankshaft_Position_Sensor_A_Circuit_Malfunction	335
OBD_Camshaft_Position_Sensor_Circuit_Range_Performance	341
OBD_Ignition_Circuit_1	351
OBD_Ignition_Circuit_2	352
OBD_Ignition_Circuit_3	353
OBD_Ignition_Circuit_4	354
OBD_Ignition_Circuit_5	355
OBD_Ignition_Circuit_6	356

Meaning	Fault Code
OBD_Ignition_Circuit_7	357
OBD_Ignition_Circuit_8	358
OBD_Ignition_Circuit_9	359
OBD_Ignition_Circuit_10	360
OBD_Ignition_Circuit_11	361
OBD_Ignition_Circuit_12	362
OBD_Oil_Pressure_Sensor_Malfunction	520
OBD_System_Voltage_Malfunction	560
OBD_PCM_Processor_Fault	606
OBD_Throttle_Actuator_Control_Range_Performance_Bank_1	638
OBD_TPS2_Secondary_Timeout	2120
OBD_TPS2_Secondary_Low	2122
OBD_TPS2_Secondary_High	2123

Meaning	Fault Code
OBD_PPS_Primary_Timeout	2125
OBD_PPS_Primary_Low	2127
OBD_PPS_Primary_High	2128
OBD_PPS_Secondary_Timeout	2130
OBD_PPS_Secondary_Low	2132
OBD_PPS_Secondary_High	2133
OBD_TPS1_Correlation	2135
OBD_TPS2_Correlation	2136
OBD_PPS_Correlation	2137
OBD_Vehicle_Speed_SensorB	2158
OBD_Barometric_Press_Circ	2226
OBD_Barometric_Press_Circ_Range_Perf	2227
OBD_WB_FW_Mismatch	2133, actually:

Meaning	Fault Code
	P2231 O2 Sensor Signal Circ Shorted to Heater Circ Bank1 Sensor 1
Wideband_1_Fault	2900
Wideband_2_Fault	2901

Standard OBD fault codes not used by FOME

Code and Meaning
P0001 Fuel Volume Regulator Control Circuit/Open

Code and Meaning
P0002 Fuel Volume Regulator Control Circuit Range/Performance
P0003 Fuel Volume Regulator Control Circuit Low
P0004 Fuel Volume Regulator Control Circuit High
P0005 Fuel Shutoff Valve "A" Control Circuit/Open
P0006 Fuel Shutoff Valve "A" Control Circuit Low
P0007 Fuel Shutoff Valve "A" Control Circuit High
P0008 Engine Positions System Performance Bank 1
P0009 Engine Position System Performance Bank 2
P0010 "A" Camshaft Position Actuator Circuit (Bank 1)
P0001 Fuel Volume Regulator Control Circuit/Open
P0002 Fuel Volume Regulator Control Circuit Range/Performance
P0003 Fuel Volume Regulator Control Circuit Low
P0004 Fuel Volume Regulator Control Circuit High
P0005 Fuel Shutoff Valve "A" Control Circuit/Open

Code and Meaning
P0006 Fuel Shutoff Valve "A" Control Circuit Low
P0007 Fuel Shutoff Valve "A" Control Circuit High
P0008 Engine Positions System Performance Bank 1
P0009 Engine Position System Performance Bank 2
P0010 "A" Camshaft Position Actuator Circuit (Bank 1)
P0011 "A" Camshaft Position - Timing Over-Advanced or System Performance (Bank 1)
P0012 "A" Camshaft Position - Timing Over-Retarded (Bank 1)
P0013 "B" Camshaft Position - Actuator Circuit (Bank 1)
P0014 "B" Camshaft Position - Timing Over-Advanced or System Performance (Bank 1)
P0015 "B" Camshaft Position -Timing Over-Retarded (Bank 1)
P0016 Crankshaft Position - Camshaft Position Correlation (Bank 1 Sensor A)
P0017 Crankshaft Position - Camshaft Position Correlation (Bank 1 Sensor B)

Code and Meaning
P0018 Crankshaft Position - Camshaft Position Correlation (Bank 2 Sensor A)
P0019 Crankshaft Position - Camshaft Position Correlation (Bank 2 Sensor B)
P0020 "A" Camshaft Position Actuator Circuit (Bank 2)
P0021 "A" Camshaft Position - Timing Over-Advanced or System Performance (Bank 2)
P0022 "A" Camshaft Position - Timing Over-Retarded (Bank 2)
P0023 "B" Camshaft Position - Actuator Circuit (Bank 2)
P0024 "B" Camshaft Position - Timing Over-Advanced or System Performance (Bank 2)
P0025 "B" Camshaft Position - Timing Over-Retarded (Bank 2)
P0026 Intake Valve Control Solenoid Circuit Range/Performance Bank 1
P0027 Exhaust Valve Control solenoid Circuit Range/Performance Bank 1
P0028 Intake valve Control Solenoid Circuit Range/Performance Bank 2
P0029 Exhaust Valve Control Solenoid Circuit Range/Performance Bank

Code and Meaning
2
P0030 HO2S Heater Control Circuit (Bank 1 Sensor 1)
P0031 HO2S Heater Control Circuit Low (Bank 1 Sensor 1)
P0032 HO2S Heater Control Circuit High (Bank 1 Sensor 1)
P0033 Turbo Charger Bypass Valve Control Circuit
P0034 Turbo Charger Bypass Valve Control Circuit Low
P0035 Turbo Charger Bypass Valve Control Circuit High
P0036 HO2S Heater Control Circuit (Bank 1 Sensor 2)
P0037 HO2S Heater Control Circuit Low (Bank 1 Sensor 2)
P0038 HO2S Heater Control Circuit High (Bank 1 Sensor 2)
P0039 Turbo/Super Charger Bypass Valve Control Circuit Range/ Performance
P0040 Upstream Oxygen Sensors Swapped From Bank To Bank
P0041 Downstream Oxygen Sensors Swapped From Bank To Bank
P0042 HO2S Heater Control Circuit (Bank 1 Sensor 3)

Code and Meaning
P0043 HO2S Heater Control Circuit Low (Bank 1 Sensor 3)
P0044 HO2S Heater Control Circuit High (Bank 1 Sensor 3)
P0050 HO2S Heater Control Circuit (Bank 2 Sensor 1)
P0051 HO2S Heater Control Circuit Low (Bank 2 Sensor 1)
P0052 HO2S Heater Control Circuit High (Bank 2 Sensor 1)
P0053 HO2S Heater Resistance (Bank 1, Sensor 1)
P0054 HO2S Heater Resistance (Bank 1, Sensor 2)
P0055 HO2S Heater Resistance (Bank 1, Sensor 3)
P0056 HO2S Heater Control Circuit (Bank 2 Sensor 2)
P0057 HO2S Heater Control Circuit Low (Bank 2 Sensor 2)
P0058 HO2S Heater Control Circuit High (Bank 2 Sensor 2)
P0059 HO2S Heater Resistance (Bank 2, Sensor 1)
P0060 HO2S Heater Resistance (Bank 2, Sensor 2)
P0061 HO2S Heater Resistance (Bank 2, Sensor 3)

Code and Meaning
P0062 HO2S Heater Control Circuit (Bank 2 Sensor 3)
P0063 HO2S Heater Control Circuit Low (Bank 2 Sensor 3)
P0064 HO2S Heater Control Circuit High (Bank 2 Sensor 3)
P0065 Air Assisted Injector Control Range/Performance
P0066 Air Assisted Injector Control Circuit or Circuit Low
P0067 Air Assisted Injector Control Circuit High
P0068 MAP/MAF - Throttle Position Correlation
P0069 Manifold Absolute Pressure - Barometric Pressure Correlation
P0070 Ambient Air Temperature Sensor Circuit
P0071 Ambient Air Temperature Sensor Range/Performance
P0072 Ambient Air Temperature Sensor Circuit Low Input
P0073 Ambient Air Temperature Sensor Circuit High Input
P0074 Ambient Air Temperature Sensor Circuit Intermittent
P0075 Intake Valve Control Solenoid Circuit (Bank 1)

Code and Meaning
P0076 Intake Valve Control Solenoid Circuit Low (Bank 1)
/P0077 Intake Valve Control Solenoid Circuit High (Bank 1)
P0078 Exhaust Valve Control Solenoid Circuit (Bank 1)
P0079 Exhaust Valve Control Solenoid Circuit Low (Bank 1)
/P0080 Exhaust Valve Control Solenoid Circuit High (Bank 1)
P0081 Intake valve Control Solenoid Circuit (Bank 2)
P0082 Intake Valve Control Solenoid Circuit Low (Bank 2)
P0083 Intake Valve Control Solenoid Circuit High (Bank 2)
P0084 Exhaust Valve Control Solenoid Circuit (Bank 2)
P0085 Exhaust Valve Control Solenoid Circuit Low (Bank 2)
P0086 Exhaust Valve Control Solenoid Circuit High (Bank 2)
P0087 Fuel Rail/System Pressure - Too Low
P0088 Fuel Rail/System Pressure - Too High
P0089 Fuel Pressure Regulator 1 Performance

Code and Meaning
P0091 Fuel Pressure Regulator 1 Control Circuit Low
P0092 Fuel Pressure Regulator 1 Control Circuit High
P0093 Fuel System Leak Detected - Large Leak
P0094 Fuel System Leak Detected - Small Leak
P0095 Intake Air Temperature Sensor 2 Circuit
P0096 Intake Air Temperature Sensor 2 Circuit Range/Performance
P0097 Intake Air Temperature Sensor 2 Circuit Low
P0098 Intake Air Temperature Sensor 2 Circuit High
P0099 Intake Air Temperature Sensor 2 Circuit Intermittent/Erratic
P0101 Mass or Volume Air Flow Circuit Range/Performance Problem
P0102 Mass or Volume Air Flow Circuit Low Input
P0103 Mass or Volume Air Flow Circuit High Input
P0104 Mass or Volume Air Flow Circuit Intermittent
P0106 Manifold Absolute Pressure/Barometric Pressure Circuit Range/Performance Problem

Code and Meaning
P0107 Manifold Absolute Pressure/Barometric Pressure Circuit Low Input
P0108 Manifold Absolute Pressure/Barometric Pressure Circuit High Input
P0109 Manifold Absolute Pressure/Barometric Pressure Circuit Intermittent
P0111 Intake Air Temperature Circuit Range/Performance Problem
P0112 Intake Air Temperature Circuit Low Input
P0113 Intake Air Temperature Circuit High Input
P0114 Intake Air Temperature Circuit Intermittent
P0116 Engine Coolant Temperature Circuit Range/Performance Problem
P0117 Engine Coolant Temperature Circuit Low Input
P0118 Engine Coolant Temperature Circuit High Input
P0119 Engine Coolant Temperature Circuit Intermittent
P0122 Throttle Position Sensor/Switch A Circuit Low Input
P0123 Throttle Position Sensor/Switch A Circuit High Input

Code and Meaning
P0124 Throttle Position Sensor/Switch A Circuit Intermittent
P0125 Insufficient Coolant Temperature for Closed Loop Fuel Control
P0126 Insufficient Coolant Temperature for Stable Operation
P0128 Coolant Thermostat (Coolant Temperature Below Thermostat Regulating Temperature)
P0130 02 Sensor Circuit Malfunction (Bank I Sensor 1)
P0131 02 Sensor Circuit Low Voltage (Bank I Sensor I)
P0132 02 Sensor Circuit High Voltage (Bank I Sensor 1)
P0133 02 Sensor Circuit Slow Response (Bank 1 Sensor 1)
P0134 02 Sensor Circuit No Activity Detected (Bank I Sensor 1)
P0135 02 Sensor Heater Circuit Malfunction (Bank 1 Sensor 1)
P0136 02 Sensor Circuit Malfunction (Bank I Sensor 2)
P0137 02 Sensor Circuit Low Voltage (Bank I Sensor 2)
P0138 02 Sensor Circuit High Voltage (Bank I Sensor 2)
P0139 02 Sensor Circuit Slow Response (Bank 1 Sensor 2)

Code and Meaning
P0140 02 Sensor Circuit No Activity Detected (Bank 1 Sensor 2)
P0141 02 Sensor Heater Circuit Malfunction (Bank 1 Sensor 2)
P0142 02 Sensor Circuit Malfunction (Bank I Sensor 3)
P0143 02 Sensor Circuit Low Voltage (Bank I Sensor 3)
P0144 02 Sensor Circuit High Voltage (Bank I Sensor 3)
P0145 02 Sensor Circuit Slow Response (Bank 1 Sensor 3)
P0146 02 Sensor Circuit No Activity Detected (Bank I Sensor 3)
P0147 02 Sensor Heater Circuit Malfunction (Bank I Sensor 3)
P0150 02 Sensor Circuit Malfunction (Bank 2 Sensor I)
P0151 02 Sensor Circuit Low Voltage (Bank 2 Sensor I)
P0152 02 Sensor Circuit High Voltage (Bank 2 Sensor 1)
P0153 02 Sensor Circuit Slow Response (Bank 2 Sensor 1)
P0154 02 Sensor Circuit No Activity Detected (Bank 2 Sensor 1)
P0155 02 Sensor Heater Circuit Malfunction (Bank 2 Sensor 1)

Code and Meaning
P0156 02 Sensor Circuit Malfunction (Bank 2 Sensor 2)
P0157 02 Sensor Circuit Low Voltage (Bank 2 Sensor 2)
P0158 02 Sensor Circuit High Voltage (Bank 2 Sensor 2)
P0159 02 Sensor Circuit Slow Response (Bank 2 Sensor 2)
P0160 02 Sensor Circuit No Activity Detected (Bank 2 Sensor 2)
P0161 02 Sensor Heater Circuit Malfunction (Bank 2 Sensor 2)
P0162 02 Sensor Circuit Malfunction (Bank 2 Sensor 3)
P0163 02 Sensor Circuit Low Voltage (Bank 2 Sensor 3)
P0164 02 Sensor Circuit High Voltage (Bank 2 Sensor 3)
P0165 02 Sensor Circuit Slow Response (Bank 2 Sensor 3)
P0166 02 Sensor Circuit No Activity Detected (Bank 2 Sensor 3)
P0167 02 Sensor Heater Circuit Malfunction (Bank 2 Sensor 3)
P0170 Fuel Trim Malfunction (Bank 1)
P0171 System too Lean (Bank 1)

Code and Meaning
P0172 System too Rich (Bank 1)
P0173 Fuel Trim Malfunction (Bank 2)
P0174 System too Lean (Bank 2)
/P0175 System too Rich (Bank 2)
P0176 Fuel Composition Sensor Circuit Malfunction
P0177 Fuel Composition Sensor Circuit Range/Performance
P0178 Fuel Composition Sensor Circuit Low Input
P0179 Fuel Composition Sensor Circuit High Input
P0180 Fuel Temperature Sensor A Circuit Malfunction
P0181 Fuel Temperature Sensor A Circuit Range/Performance
P0182 Fuel Temperature Sensor A Circuit Low Input
P0183 Fuel Temperature Sensor A Circuit High Input
P0184 Fuel Temperature Sensor A Circuit Intermittent
P0185 Fuel Temperature Sensor B Circuit Malfunction

Code and Meaning
P0186 Fuel Temperature Sensor B Circuit Range/Performance
P0187 Fuel Temperature Sensor B Circuit Low Input
P0188 Fuel Temperature Sensor B Circuit High Input
P0189 Fuel Temperature Sensor B Circuit Intermittent
P0190 Fuel Rail Pressure Sensor Circuit Malfunction
P0191 Fuel Rail Pressure Sensor Circuit Range/Performance
P0192 Fuel Rail Pressure Sensor Circuit Low Input
P0193 Fuel Rail Pressure Sensor Circuit High Input
P0194 Fuel Rail Pressure Sensor Circuit Intermittent
P0195 Engine Oil Temperature Sensor Malfunction
P0196 Engine Oil Temperature Sensor Range/Performance
P0197 Engine Oil Temperature Sensor Low
P0198 Engine Oil Temperature Sensor High
P0199 Engine Oil Temperature Sensor Intermittent

Code and Meaning
P0200 Injector Circuit Malfunction
P0213 Cold Start Injector 1 Malfunction
P0214 Cold Start Injector 2 Malfunction
P0215 Engine Shutoff Solenoid Malfunction
P0216 Injection Timing Control Circuit Malfunction
P0217 Engine Overtemp Condition
P0218 Transmission Over Temperature Condition
P0219 Engine Overspeed Condition
P0220 Throttle/Petal Position Sensor/Switch B Circuit Malfunction
P0221 Throttle/Petal Position Sensor/Switch B Circuit Range/ Performance Problem
P0222 Throttle/Petal Position Sensor/Switch B Circuit Low Input
P0223 Throttle/Petal Position Sensor/Switch B Circuit High Input
P0224 Throttle/Petal Position Sensor/Switch B Circuit Intermittent
P0225 Throttle/Petal Position Sensor/Switch C Circuit Malfunction

Code and Meaning
P0226 Throttle/Petal Position Sensor/Switch C Circuit Range/ Performance Problem
P0227 Throttle/Petal Position Sensor/Switch C Circuit Low Input
P0228 Throttle/Petal Position Sensor/Switch C Circuit High Input
P0229 Throttle/Petal Position Sensor/Switch C Circuit Intermittent
P0230 Fuel Pump Primary Circuit Malfunction
P0231 Fuel Pump Secondary Circuit Low
P0232 Fuel Pump Secondary Circuit High
P0233 Fuel Pump Secondary Circuit Intermittent
P0234 Engine Turbocharger/Supercharger Overboost Condition
P0235 Turbocharger Boost Sensor A Circuit Malfunction
P0236 Turbocharger Boost Sensor A Circuit Range/Performance
P0237 Turbocharger Boost Sensor A Circuit Low
P0238 Turbocharger Boost Sensor A Circuit High
P0239 Turbocharger Boost Sensor B Malfunction

Code and Meaning
P0240 Turbocharger Boost Sensor B Circuit Range/Performance
P0241 Turbocharger Boost Sensor B Circuit Low
P0242 Turbocharger Boost Sensor B Circuit High
P0243 Turbocharger Wastegate Solenoid A Malfunction
P0244 Turbocharger Wastegate Solenoid A Range/Performance
P0245 Turbocharger Wastegate Solenoid A Low
P0246 Turbocharger Wastegate Solenoid A High
P0247 Turbocharger Wastegate Solenoid B Malfunction
P0248 Turbocharger Wastegate Solenoid B Range/Performance
P0249 Turbocharger Wastegate Solenoid B Low
P0250 Turbocharger Wastegate Solenoid B High
P0251 Injection Pump Fuel Metering Control "A" Malfunction (Cam/Rotor/Injector)
P0252 Injection Pump Fuel Metering Control "A" Range/Performance (Cam/Rotor/Injector)

Code and Meaning
P0253 Injection Pump Fuel Metering Control "A" Low (Cam/Rotor/ Injector)
P0254 Injection Pump Fuel Metering Control "A" High (Cam/Rotor/ Injector)
P0255 Injection Pump Fuel Metering Control "A" Intermittent (Cam/ Rotor/Injector)
P0256 Injection Pump Fuel Metering Control "B" Malfunction (Cam/ Rotor/Injector)
P0257 Injection Pump Fuel Metering Control "B" Range/Performance Injector)
P0258 Injection Pump Fuel Metering Control "B" Low (Cam/R
P0259 Injection Pump Fuel Metering Control "B" High (Cam/R
P0260 Injection Pump Fuel Metering Control "B" Intermittent Injector)
P0261 Cylinder I Injector Circuit Low
P0262 Cylinder I Injector Circuit High
P0263 Cylinder I Contribution/Balance Fault
P0264 Cylinder 2 Injector Circuit Low

Code and Meaning
P0265 Cylinder 2 Injector Circuit High
P0266 Cylinder 2 Contribution/Balance Fault
P0267 Cylinder 3 Injector Circuit Low
P0268 Cylinder 3 Injector Circuit High
P0269 Cylinder 3 Contribution/Balance Fault
P0270 Cylinder 4 Injector Circuit Low
P0271 Cylinder 4 Injector Circuit High
P0272 Cylinder 4 Contribution/Balance Fault
P0273 Cylinder 5 Injector Circuit Low
P0274 Cylinder 5 Injector Circuit High
P0275 Cylinder S Contribution/Balance Fault
P0276 Cylinder 6 Injector Circuit Low
P0277 Cylinder 6 Injector Circuit High
P0278 Cylinder 6 Contribution/Balance Fault

Code and Meaning
P0279 Cylinder 7 Injector Circuit Low
P0280 Cylinder 7 Injector Circuit High
P0281 Cylinder 7 Contribution/Balance Fault
P0282 Cylinder 8 Injector Circuit Low
P0283 Cylinder 8 Injector Circuit High
P0284 Cylinder 8 Contribution/Balance Fault
P0285 Cylinder 9 Injector Circuit Low
P0286 Cylinder 9 Injector Circuit High
P0287 Cylinder 9 Contribution/Balance Fault
P0288 Cylinder 10 Injector Circuit Low
P0289 Cylinder 10 Injector Circuit High
P0290 Cylinder 10 Contribution/Balance Fault
P0291 Cylinder 11 Injector Circuit Low
P0292 Cylinder 11 Injector Circuit High

Code and Meaning
P0293 Cylinder 11 Contribution/Balance Fault
P0294 Cylinder 12 Injector Circuit Low
P0295 Cylinder 12 Injector Circuit High
P0296 Cylinder 12 Contribution/Range Fault
P0297 Vehicle Overspeed Condition
P0298 Engine Oil Over Temperature Condition
P0299 Turbocharger/Supercharger "A" Underboost Condition
P0300 Random/Multiple Cylinder Misfire Detected
P0301 Cylinder 1 Misfire Detected
P0302 Cylinder 2 Misfire Detected
P0303 Cylinder 3 Misfire Detected
P0304 Cylinder 4 Misfire Detected
P0305 Cylinder 5 Misfire Detected
P0306 Cylinder 6 Misfire Detected

Code and Meaning
P0307 Cylinder 7 Misfire Detected
P0308 Cylinder 8 Misfire Detected
P0309 Cylinder 9 Misfire Detected
P0310 Cylinder 10 Misfire Detected
P0311 Cylinder 11 Misfire Detected
P0312 Cylinder 12 Misfire Detected
P0313 Misfire Detected with Low Fuel
P0314 Single Cylinder Misfire (Cylinder not Specified)
P0315 Crankshaft Position System Variation Not Learned
P0316 Misfire Detected On Startup (First 1000 Revolutions)
P0317 Rough Road Hardware Not Present
P0318 Rough Road Sensor A Signal Circuit
P0319 Rough Road Sensor B Signal Circuit
P0320 Ignition/Distributor Engine Speed Input Circuit Malfunction

Code and Meaning
P0321 Ignition/Distributor Engine Speed Input Circuit Range/ Performance
P0322 Ignition/Distributor Engine Speed Input Circuit No Signal
P0323 Ignition/Distributor Engine Speed Input Circuit Intermittent
P0324 Knock Control System Error
P0325 Knock Sensor 1 Circuit Malfunction (Bank I or Single Sensor)
P0326 Knock Sensor 1 Circuit Range/Performance (Bank 1 or Single Sensor)
P0327 Knock Sensor 1 Circuit Low Input (Bank I or Single Sensor)
P0328 Knock Sensor 1 Circuit High Input (Bank I or Single Sensor)
P0329 Knock Sensor 1 Circuit Intermittent (Bank 1 or Single Sensor)
P0330 Knock Sensor 2 Circuit Malfunction (Bank 2)
P0331 Knock Sensor 2 Circuit Range/Performance (Bank 2)
P0332 Knock Sensor 2 Circuit Low Input (Bank 2)
P0333 Knock Sensor 2 Circuit High Input (Bank 2)

Code and Meaning
P0334 Knock Sensor 2 Circuit Intermittent (Bank 2)
P0336 Crankshaft Position Sensor A Circuit Range/Performance
P0337 Crankshaft Position Sensor A Circuit Low Input
P0338 Crankshaft Position Sensor A Circuit High Input
P0339 Crankshaft Position Sensor A Circuit Intermittent
P0340 Camshaft Position Sensor Circuit Malfunction
P0342 Camshaft Position Sensor Circuit Low Input
P0343 Camshaft Position Sensor Circuit High Input
P0344 Camshaft Position Sensor Circuit Intermittent
P0345 Camshaft Position Sensor A Circuit Malfunction (Bank 2)
P0346 Camshaft Position Sensor A Circuit Range/Performance (Bank 2)
P0347 Camshaft Position Sensor A Circuit Low Input (Bank 2)
P0348 Camshaft Position Sensor A Circuit High Input (Bank 2)
P0349 Camshaft Position Sensor A Circuit Intermittent (Bank 2)

Code and Meaning
P0350 Ignition Coil Primary/Secondary Circuit Malfunction
P0363 Misfire Detected - Fueling Disabled
P0364 Reserved
P0365 Camshaft Position Sensor "B" Circuit (Bank 1)
P0366 Camshaft Position Sensor "B" Circuit Range/Performance (Bank 1)
P0367 Camshaft Position Sensor "B" Circuit Low (Bank 1)
P0368 Camshaft Position Sensor "B" Circuit High (Bank 1)
P0369 Camshaft Position Sensor "B" Circuit Intermittent (Bank 1)
P0370 Timing Reference High Resolution Signal A Malfunction
P0371 Timing Reference High Resolution Signal A Too Many Pulses
P0372 Timing Reference High Resolution Signal A Too Few Pulses
P0373 Timing Reference High Resolution Signal A Intermittent/Erratic Pulses
P0374 Timing Reference High Resolution Signal A No Pulses

Code and Meaning
P0375 Timing Reference High Resolution Signal B Malfunction
P0376 Timing Reference High Resolution Signal B Too Many Pulses
P0377 Timing Reference High Resolution Signal B Too Few Pulses
P0378 Timing Reference High Resolution Signal B Intermittent/Erratic Pulses
P0379 Timing Reference High Resolution Signal B No Pulses
P0380 Glow Plug/Heater Circuit "A" Malfunction
P0381 Glow Plug/Heater Indicator Circuit Malfunction
P0382 Exhaust Gas Recirculation Flow Malfunction
P0383 Glow Plug Control Module Control Circuit Low
P0384 Glow Plug Control Module Control Circuit High
P0385 Crankshaft Position Sensor B Circuit Malfunction
P0386 Crankshaft Position Sensor B Circuit Range/Performance
P0387 Crankshaft Position Sensor B Circuit Low Input
P0388 Crankshaft Position Sensor B Circuit High Input

Code and Meaning
P0389 Crankshaft Position Sensor B Circuit Intermittent
P0390 Camshaft Position Sensor "B" Circuit (Bank 2)
P0391 Camshaft Position Sensor "B" Circuit Range/Performance (Bank 2)
P0392 Camshaft Position Sensor "B" Circuit Low (Bank 2)
P0393 Camshaft Position Sensor "B" Circuit High (Bank 2)
P0394 Camshaft Position Sensor "B" Circuit Intermittent (Bank 2)
DTC Codes - P0400-P0499 - Auxiliary Emissions Controls
P0400 Exhaust Gas Recirculation Flow Malfunction
P0401 Exhaust Gas Recirculation Flow Insufficient Detected
P0402 Exhaust Gas Recirculation Flow Excessive Detected
P0403 Exhaust Gas Recirculation Circuit Malfunction
P0404 Exhaust Gas Recirculation Circuit Range/Performance
P0405 Exhaust Gas Recirculation Sensor A Circuit Low
P0406 Exhaust Gas Recirculation Sensor A Circuit High

Code and Meaning
P0407 Exhaust Gas Recirculation Sensor B Circuit Low
P0408 Exhaust Gas Recirculation Sensor B Circuit High
P0409 Exhaust Gas Recirculation Sensor "A" Circuit
P0410 Secondary Air Injection System Malfunction
P0411 Secondary Air Injection System Incorrect Flow Detected
P0412 Secondary Air Injection System Switching Valve A Circuit Malfunction
P0413 Secondary Air Injection System Switching Valve A Circuit Open
P0414 Secondary Air Injection System Switching Valve A Circuit Shorted
P0415 Secondary Air Injection System Switching Valve B Circuit Malfunction
P0416 Secondary Air Injection System Switching Valve B Circuit Open
P0417 Secondary Air Injection System Switching Valve B Circuit Shorted
P0418 Secondary Air Injection System Relay "A" Circuit Malfunction

Code and Meaning
P0419 Secondary Air Injection System Relay "B" Circuit Malfunction
P0420 Catalyst System Efficiency Below Threshold (Bank 1)
P0421 Warm Up Catalyst Efficiency Below Threshold (Bank 1)
P0422 Main Catalyst Efficiency Below Threshold (Bank 1)
P0423 Heated Catalyst Efficiency Below Threshold (Bank 1)
P0424 Heated Catalyst Temperature Below Threshold (Bank 1)
P0424 Heated Catalyst Temperature Below Threshold (Bank 1)
P0425 Catalyst Temperature Sensor (Bank 1)
P0426 Catalyst Temperature Sensor Range/Performance (Bank 1)
P0427 Catalyst Temperature Sensor Low (Bank 1)
P0428 Catalyst Temperature Sensor High (Bank 1)
P0429 Catalyst Heater Control Circuit
P0430 Catalyst System Efficiency Below Threshold (Bank 2)
P0431 Warm Up Catalyst Efficiency Below Threshold (Bank 2)

Code and Meaning
P0432 Main Catalyst Efficiency Below Threshold (Bank 2)
P0433 Heated Catalyst Efficiency Below Threshold (Bank 2)
P0434 Heated Catalyst Temperature Below Threshold (Bank 2)
P0435 Catalyst Temperature Sensor (Bank 2)
P0436 Catalyst Temperature Sensor Range/Performance (Bank 2)
P0437 Catalyst Temperature Sensor Low (Bank 2)
P0438 Catalyst Temperature Sensor High (Bank 2)
P0439 Catalyst Heater Control Circuit
P0440 Evaporative Emission Control System Malfunction
P0441 Evaporative Emission Control System Incorrect Purge Flow
P0442 Evaporative Emission Control System Leak Detected (small leak)
P0443 Evaporative Emission Control System Purge Control Valve Circuit
P0444 Evaporative Emission Control System Purge Control Valve Circuit Open

Code and Meaning
P0445 Evaporative Emission Control System Purge Control Valve Circuit Shorted
P0446 Evaporative Emission Control System Vent Control Circuit Malfunction
P0447 Evaporative Emission Control System Vent Control Circuit Open
P0448 Evaporative Emission Control System Vent Control Circuit Shorted
P0449 Evaporative Emission Control System Vent Valve/Solenoid Circuit Malfunction
P0450 Evaporative Emission Control System Pressure Sensor Malfunction
P0451 Evaporative Emission Control System Pressure Sensor Range/Performance
P0452 Evaporative Emission Control System Pressure Sensor Low Input
P0453 Evaporative Emission Control System Pressure Sensor High Input
P0454 Evaporative Emission Control System Pressure Sensor Intermittent

Code and Meaning
P0455 Evaporative Emission Control System Leak Detected (gross leak)
P0456 Evaporative Emissions System Small Leak Detected
P0457 Evaporative Emission Control System Leak Detected
P0458 Evaporative Emission System Purge Control Valve Circuit Low
P0459 Evaporative Emission System Purge Control Valve Circuit High
P0460 Fuel Level Sensor Circuit Malfunction
P0461 Fuel Level Sensor Circuit Range/Performance
P0462 Fuel Level Sensor Circuit Low Input
P0463 Fuel Level Sensor Circuit High Input
P0464 Fuel Level Sensor Circuit Intermittent
P0465 Purge Flow Sensor Circuit Malfunction
P0466 Purge Flow Sensor Circuit Range/Performance
P0467 Purge Flow Sensor Circuit Low Input
P0468 Purge Flow Sensor Circuit High Input

Code and Meaning
P0469 Purge Flow Sensor Circuit Intermittent
P0470 Exhaust Pressure Sensor Malfunction
P0471 Exhaust Pressure Sensor Range/Performance
P0472 Exhaust Pressure Sensor Low
P0473 Exhaust Pressure Sensor High
P0474 Exhaust Pressure Sensor Intermittent
P0475 Exhaust Pressure Control Valve Malfunction
P0476 Exhaust Pressure Control Valve Range/Performance
P0477 Exhaust Pressure Control Valve Low
P0478 Exhaust Pressure Control Valve High
P0479 Exhaust Pressure Control Valve Intermittent
P0480 Cooling Fan I Control Circuit Malfunction
P0481 Cooling Fan 2 Control Circuit Malfunction
P0482 Cooling Fan 3 Control Circuit Malfunction

Code and Meaning
P0483 Cooling Fan Rationality Check Malfunction
P0484 Cooling Fan Circuit Over Current
P0485 Cooling Fan Power/Ground Circuit Malfunction
P0486 Exhaust Gas Recirculation Sensor "B" Circuit
P0487 Exhaust Gas Recirculation Throttle Control Circuit "A" Open
P0488 Exhaust Gas Recirculation Throttle Control Circuit "A" Range/ Performance
P0489 Exhaust Gas Recirculation "A" Control Circuit Low
P0490 Exhaust Gas Recirculation "A" Control Circuit High
P0491 Secondary Air Injection System Insufficient Flow Bank 1
P0492 Secondary Air Injection System Insufficient Flow Bank 2
P0493 Fan Overspeed
P0494 Fan Speed Low
P0495 Fan Speed High
P0496 EVAP Flow During A Non-Purge Condition

Code and Meaning
P0497 Evaporative Emission System Low Purge Flow
P0498 Evaporative Emission System Vent Valve Control Circuit Low
P0499 Evaporative Emission System Vent Valve Control Circuit High
DTC Codes - P0500-P0599 - Vehicle Speed Controls and Idle Control System
P0500 Vehicle Speed Sensor Malfunction
P0501 Vehicle Speed Sensor Range/Performance
P0502 Vehicle Speed Sensor Low Input
P0503 Vehicle Speed Sensor Intermittent/Erratic/High
P0504 Brake Switch "A"/"B" Correlation
P0505 Idle Control System Malfunction
P0506 Idle Control System RPM Lower Than Expected
P0507 Idle Control System RPM Higher Than Expected
P0508 Idle Air Control System Circuit Low
P0509 Idle Air Control System Circuit High

Code and Meaning
P0510 Closed Throttle Position Switch Malfunction
P0511 Idle Air Control Circuit
P0512 Starter Request Circuit
P0513 Incorrect Immobilizer Key
P0514 Battery Temperature Sensor Circuit Range/Performance
P0515 Battery Temperature Sensor Circuit
P0516 Battery Temperature Sensor Circuit Low
P0517 Battery Temperature Sensor Circuit High
P0518 Idle Air Control Circuit Intermittent
P0519 Idle Air Control System Performance
P0520 Engine Oil Pressure Sensor/Switch Circuit Malfunction
P0521 Engine Oil Pressure Sensor/Switch Circuit Range/Performance
P0522 Engine Oil Pressure Sensor/Switch Circuit Low Voltage
P0523 Engine Oil Pressure Sensor/Switch Circuit High Voltage

Code and Meaning
P0524 Engine Oil Pressure Too Low
P0525 Cruise Control Servo Control Circuit Range/Performance
P0526 Fan Speed Sensor Circuit
P0527 Fan Speed Sensor Circuit Range/Performance
P0528 Fan Speed Sensor Circuit No Signal
P0529 Fan Speed Sensor Circuit Intermittent
P0530 A/C Refrigerant Pressure Sensor Circuit Malfunction
P0531 A/C Refrigerant Pressure Sensor Circuit Range/Performance
P0532 A/C Refrigerant Pressure Sensor Circuit Low Input
P0533 A/C Refrigerant Pressure Sensor Circuit High Input
P0534 Air Conditioner Refrigerant Charge Loss
P0535 A/C Evaporator Temperature Sensor Circuit
P0536 A/C Evaporator Temperature Sensor Circuit Range/Performance
P0537 A/C Evaporator Temperature Sensor Circuit Low

Code and Meaning
P0538 A/C Evaporator Temperature Sensor Circuit High
P0539 A/C Evaporator Temperature Sensor Circuit Intermittent
P0540 Intake Air Heater "A" Circuit
P0541 Intake Air Heater "A" Circuit Low
P0542 Intake Air Heater "A" Circuit High
P0543 Intake Air Heater "A" Circuit Open
P0544 Exhaust Gas Temperature Sensor Circuit
P0545 Exhaust Gas Temperature Sensor Circuit Low
P0546 Exhaust Gas Temperature Sensor Circuit High
P0547 Exhaust Gas Temperature Sensor Circuit
P0548 Exhaust Gas Temperature Sensor Circuit Low
P0549 Exhaust Gas Temperature Sensor Circuit High
P0550 Power Steering Pressure Sensor Circuit Malfunction
P0551 Power Steering Pressure Sensor Circuit Range/Performance

Code and Meaning
P0552 Power Steering Pressure Sensor Circuit Low Input
P0553 Power Steering Pressure Sensor Circuit High Input
P0554 Power Steering Pressure Sensor Circuit Intermittent
P0555 Brake Booster Pressure Sensor Circuit
P0556 Brake Booster Pressure Sensor Circuit Range/Performance
P0557 Brake Booster Pressure Sensor Circuit Low
P0558 Brake Booster Pressure Sensor Circuit High
P0559 Brake Booster Pressure Sensor Circuit Intermittent
P0561 System Voltage Unstable
P0562 System Voltage Low
P0563 System Voltage High
P0564 Cruise Control Multi-Function Input "A" Circuit
P0565 Cruise Control On Signal Malfunction
P0566 Cruise Control Off Signal Malfunction

Code and Meaning
P0567 Cruise Control Resume Signal Malfunction
P0568 Cruise Control Set Signal Malfunction
P0569 Cruise Control Coast Signal Malfunction
P0570 Cruise Control Accel Signal Malfunction
P0571 Cruise Control/Brake Switch A Circuit Malfunction
P0572 Cruise Control/Brake Switch A Circuit Low
P0573 Cruise Control/Brake Switch A Circuit High
P0574 Cruise Control System - Vehicle Speed Too High
P0575 Cruise Control Input Circuit
P0576 Cruise Control Input Circuit Low
P0577 Cruise Control Input Circuit High
P0578 Cruise Control Multi-Function Input "A" Circuit Stuck
P0579 Cruise Control Multi-Function Input "A" Circuit Range/ Performance
P0580 Cruise Control Multi-Function Input "A" Circuit Low

Code and Meaning
P0581 Cruise Control Multi-Function Input "A" Circuit High
P0582 Cruise Control Vacuum Control Circuit/Open
P0583 Cruise Control Vacuum Control Circuit Low
P0584 Cruise Control Vacuum Control Circuit High
P0585 Cruise Control Multi-Function Input "A"/"B" Correlation
P0586 Cruise Control Vent Control Circuit/Open
P0587 Cruise Control Vent Control Circuit Low
P0588 Cruise Control Vent Control Circuit High
P0589 Cruise Control Multi-Function Input "B" Circuit
P0590 Cruise Control Multi-Function Input "B" Circuit Stuck
P0591 Cruise Control Multi-Function Input "B" Circuit Range/ Performance
P0592 Cruise Control Multi-Function Input "B" Circuit Low
P0593 Cruise Control Multi-Function Input "B" Circuit High
P0594 Cruise Control Servo Control Circuit/Open

Code and Meaning
P0595 Cruise Control Servo Control Circuit Low
P0596 Cruise Control Servo Control Circuit High
P0597 Thermostat Heater Control Circuit/Open
P0598 Thermostat Heater Control Circuit Low
P0599 Thermostat Heater Control Circuit High
P0600 Serial Communication Link Malfunction
P0601 Internal Control Module Memory Check Sum Error
P0602 Control Module Programming Error
P0603 Internal Control Module Keep Alive Memory (KAM) Error
P0604 Internal Control Module Random Access Memory (RAM) Error
P0605 Internal Control Module Read Only Memory (ROM) Error
P0607 Control Module Performance
P0608 Control Module VSS Output "A" Malfunction
P0609 Control Module VSS Output "B" Malfunction

Code and Meaning
P0610 Control Module Vehicle Options Error
P0611 Fuel Injector Control Module Performance
P0612 Fuel Injector Control Module Relay Control
P0613 TCM Processor
P0614 ECM / TCM Incompatible
P0615 Starter Relay Circuit
P0616 Starter Relay Circuit Low
P0617 Starter Relay Circuit High
P0618 Alternative Fuel Control Module KAM Error
P0619 Alternative Fuel Control Module RAM/ROM Error
P0620 Generator Control Circuit Malfunction
P0621 Generator Lamp "L" Control Circuit Malfunction
P0622 Generator Field "F" Control Circuit Malfunction
P0623 Generator Lamp Control Circuit

Code and Meaning
P0624 Fuel Cap Lamp Control Circuit
P0625 Generator Field/F Terminal Circuit Low
P0626 Generator Field/F Terminal Circuit High
P0627 Fuel Pump “A” Control Circuit /Open
P0628 Fuel Pump “A” Control Circuit Low
P0629 Fuel Pump “A” Control Circuit High
P0630 VIN Not Programmed or Incompatible - ECM/PCM
P0631 VIN Not Programmed or Incompatible - TCM
P0632 Odometer Not Programmed - ECM/PCM
P0633 Immobilizer Key Not Programmed - ECM/PCM
P0634 PCM/ECM/TCM Internal Temperature Too High
P0635 Power Steering Control Circuit
P0636 Power Steering Control Circuit Low
P0637 Power Steering Control Circuit High

Code and Meaning
P0639 Throttle Actuator Control Range/Performance (Bank 2)
P0640 Intake Air Heater Control Circuit
P0641 Sensor Reference Voltage “A” Circuit/Open
P0642 Sensor Reference Voltage “A” Circuit Low
P0643 Sensor Reference Voltage “A” Circuit High
P0644 Driver Display Serial Communication Circuit
P0645 A/C Clutch Relay Control Circuit
P0646 A/C Clutch Relay Control Circuit Low
P0647 A/C Clutch Relay Control Circuit High
P0648 Immobilizer Lamp Control Circuit
P0649 Speed Control Lamp Control Circuit
P0650 Malfunction Indicator Lamp (MIL) Control Circuit Malfunction
P0651 Sensor Reference Voltage “B” Circuit/Open
P0652 Sensor Reference Voltage “B” Circuit Low

Code and Meaning
P0653 Sensor Reference Voltage “B” Circuit High
P0654 Engine RPM Output Circuit Malfunction
P0655 Engine Hot Lamp Output Control Circuit Malfunction
P0656 Fuel Level Output Circuit Malfunction
P0657 Actuator Supply Voltage "A" Circuit/Open
P0658 Actuator Supply Voltage "A" Circuit Low
P0659 Actuator Supply Voltage "A" Circuit High
P0660 Intake Manifold Tuning Valve Control Circuit/Open Bank 1
P0661 Intake Manifold Tuning Valve Control Circuit Low Bank 1
P0662 Intake Manifold Tuning Valve Control Circuit High Bank 1
P0663 Intake Manifold Tuning Valve Control Circuit/Open Bank 2
P0664 Intake Manifold Tuning Valve Control Circuit Low Bank 2
P0665 Intake Manifold Tuning Valve Control Circuit High Bank 2
P0666 PCM/ECM/TCM Internal Temperature Sensor Circuit

Code and Meaning
P0667 PCM/ECM/TCM Internal Temperature Sensor Range/Performance
P0668 PCM/ECM/TCM Internal Temperature Sensor Circuit Low
P0669 PCM/ECM/TCM Internal Temperature Sensor Circuit High
P0670 Glow Plug Module Control Circuit
P0671 Cylinder 1 Glow Plug Circuit
P0672 Cylinder 2 Glow Plug Circuit
P0673 Cylinder 3 Glow Plug Circuit
P0674 Cylinder 4 Glow Plug Circuit
P0675 Cylinder 5 Glow Plug Circuit
P0676 Cylinder 6 Glow Plug Circuit
P0677 Cylinder 7 Glow Plug Circuit
P0678 Cylinder 8 Glow Plug Circuit
P0679 Cylinder 9 Glow Plug Circuit
P0680 Cylinder 10 Glow Plug Circuit

Code and Meaning
P0681 Cylinder 11 Glow Plug Circuit
P0682 Cylinder 12 Glow Plug Circuit
P0683 Glow Plug Control Module to PCM Communication Circuit
P0684 Glow Plug Control Module to PCM Communication Circuit Range/ Performance
P0685 ECM/PCM Power Relay Control Circuit /Open
P0686 ECM/PCM Power Relay Control Circuit Low
P0687 ECM/PCM Power Relay Control Circuit High
P0688 ECM/PCM Power Relay Sense Circuit /Open
P0689 ECM/PCM Power Relay Sense Circuit Low
P0690 ECM/PCM Power Relay Sense Circuit High
P0691 Fan 1 Control Circuit Low P0692 Fan 1 Control Circuit High
P0693 Fan 2 Control Circuit Low
P0694 Fan 2 Control Circuit High
P0695 Fan 3 Control Circuit Low

Code and Meaning
P0696 Fan 3 Control Circuit High
P0697 Sensor Reference Voltage “C” Circuit/Open
P0698 Sensor Reference Voltage “C” Circuit Low
P0699 Sensor Reference Voltage “C” Circuit High
P0700 Transmission Control System Malfunction
P0701 Transmission Control System Range/Performance
P0702 Transmission Control System Electrical
P0703 Torque Converter/Brake Switch B Circuit Malfunction
P0704 Clutch Switch Input Circuit Malfunction
P0705 Transmission Range Sensor Circuit malfunction (PRNDL Input)
P0706 Transmission Range Sensor Circuit Range/Performance
P0707 Transmission Range Sensor Circuit Low Input
P0708 Transmission Range Sensor Circuit High Input
P0709 Transmission Range Sensor Circuit Intermittent

Code and Meaning
P0710 Transmission Fluid Temperature Sensor Circuit Malfunction
P0711 Transmission Fluid Temperature Sensor Circuit Range/ Performance
P0712 Transmission Fluid Temperature Sensor Circuit Low Input
P0713 Transmission Fluid Temperature Sensor Circuit High Input
P0714 Transmission Fluid Temperature Sensor Circuit Intermittent
P0715 Input/Turbine Speed Sensor Circuit Malfunction
P0716 Input/Turbine Speed Sensor Circuit Range/Performance
P0717 Input/Turbine Speed Sensor Circuit No Signal
P0718 Input/Turbine Speed Sensor Circuit Intermittent
P0719 Torque Converter/Brake Switch B Circuit Low
P0720 Output Speed Sensor Circuit Malfunction
P0721 Output Speed Sensor Range/Performance
P0722 Output Speed Sensor No Signal
P0723 Output Speed Sensor Intermittent

Code and Meaning
P0724 Torque Converter/Brake Switch B Circuit High
P0725 Engine Speed input Circuit Malfunction
P0726 Engine Speed Input Circuit Range/Performance
P0727 Engine Speed Input Circuit No Signal
P0728 Engine Speed Input Circuit Intermittent
P0729 Gear 6 Incorrect Ratio
P0730 Incorrect Gear Ratio
P0731 Gear I Incorrect ratio
P0732 Gear 2 Incorrect ratio
P0733 Gear 3 Incorrect ratio
P0734 Gear 4 Incorrect ratio
P0735 Gear 5 Incorrect ratio
P0736 Reverse incorrect gear ratio
P0737 TCM Engine Speed Output Circuit

Code and Meaning
P0738 TCM Engine Speed Output Circuit Low
P0739 TCM Engine Speed Output Circuit High
P0740 Torque Converter Clutch Circuit Malfunction
P0741 Torque Converter Clutch Circuit Performance or Stuck Off
P0742 Torque Converter Clutch Circuit Stock On
P0743 Torque Converter Clutch Circuit Electrical
P0744 Torque Converter Clutch Circuit Intermittent
P0745 Pressure Control Solenoid Malfunction
P0746 Pressure Control Solenoid Performance or Stuck Off
P0747 Pressure Control Solenoid Stuck On
P0748 Pressure Control Solenoid Electrical
P0749 Pressure Control Solenoid Intermittent
P0750 Shift Solenoid A Malfunction
P0751 Shift Solenoid A Performance or Stuck Off

Code and Meaning
P0752 Shift Solenoid A Stuck On
P0753 Shift Solenoid A Electrical
P0754 Shift Solenoid A Intermittent
P0755 Shift Solenoid B Malfunction
P0756 Shift Solenoid B Performance or Stock Off
P0757 Shift Solenoid B Stuck On
P0758 Shift Solenoid B Electrical
P0759 Shift Solenoid B Intermittent
P0760 Shift Solenoid C Malfunction
P0761 Shift Solenoid C Performance or Stuck Off
P0762 Shift Solenoid C Stuck On
P0763 Shift Solenoid C Electrical
P0764 Shift Solenoid C Intermittent
P0765 Shift Solenoid D Malfunction

Code and Meaning
P0766 Shift Solenoid D Performance or Stuck Off
P0767 Shift Solenoid D Stuck On
P0768 Shift Solenoid D Electrical
P0769 Shift Solenoid D Intermittent
P0770 Shift Solenoid E Malfunction
P0771 Shift Solenoid E Performance or Stuck Off
P0772 Shift Solenoid E Stuck On
P0773 Shift Solenoid E Electrical
P0774 Shift Solenoid E Intermittent
P0775 Pressure Control Solenoid "B"
P0776 Pressure Control Solenoid "B" Performance or Stuck off
P0777 Pressure Control Solenoid "B" Stuck On
P0778 Pressure Control Solenoid "B" Electrical
P0779 Pressure Control Solenoid "B" Intermittent

Code and Meaning
P0780 Shift Malfunction
P0781 1-2 Shift Malfunction
P0782 2-3 Shift Malfunction
P0783 3-4 Shift Malfunction
P0784 4-5 Shift Malfunction
P0785 Shift/Timing Solenoid Malfunction
P0786 Shift/Timing Solenoid Range/Performance
P0787 Shift/Timing Solenoid Low
P0788 Shift/Timing Solenoid High
P0789 Shift/Timing Solenoid Intermittent
P0790 Normal/Performance Switch Circuit Malfunction
P0791 Intermediate Shaft Speed Sensor “A” Circuit
P0792 Intermediate Shaft Speed Sensor “A” Circuit Range/Performance
P0793 Intermediate Shaft Speed Sensor “A” Circuit No Signal

Code and Meaning
P0794 Intermediate Shaft Speed Sensor “A” Circuit Intermittent
P0795 Pressure Control Solenoid "C"
P0796 Pressure Control Solenoid "C" Performance or Stuck off
P0797 Pressure Control Solenoid "C" Stuck On
P0798 Pressure Control Solenoid "C" Electrical
P0799 Pressure Control Solenoid "C" Intermittent
P0800 Transfer Case Control System (MIL Request)
P0801 Reverse Inhibit Control Circuit Malfunction
P0802 Transmission Control System MIL Request Circuit/Open
P0803 1-4 Upshift (Skip Shift) Solenoid Control Circuit Malfunction
P0804 1-4 Upshift (Skip Shift) Lamp Control Circuit Malfunction
P0805 Clutch Position Sensor Circuit
P0806 Clutch Position Sensor Circuit Range/Performance
P0807 Clutch Position Sensor Circuit Low

Code and Meaning
P0808 Clutch Position Sensor Circuit High
P0809 Clutch Position Sensor Circuit Intermittent
P0810 Clutch Position Control Error
P0811 Excessive Clutch Slippage
P0812 Reverse Input Circuit
P0813 Reverse Output Circuit
P0814 Transmission Range Display Circuit
P0815 Upshift Switch Circuit
P0816 Downshift Switch Circuit
P0817 Starter Disable Circuit
P0818 Driveline Disconnect Switch Input Circuit
P0819 Up and Down Shift Switch to Transmission Range Correlation
P0820 Gear Lever X-Y Position Sensor Circuit
P0821 Gear Lever X Position Circuit

Code and Meaning
P0822 Gear Lever Y Position Circuit
P0823 Gear Lever X Position Circuit Intermittent
P0824 Gear Lever Y Position Circuit Intermittent
P0825 Gear Lever Push-Pull Switch (Shift Anticipate
P0826 Up and Down Shift Switch Circuit
P0827 Up and Down Shift Switch Circuit Low
P0828 Up and Down Shift Switch Circuit High
P0829 5-6 Shift
P0830 Clutch Pedal Switch "A" Circuit
P0831 Clutch Pedal Switch "A" Circuit Low
P0832 Clutch Pedal Switch "A" Circuit High
P0833 Clutch Pedal Switch "B" Circuit
P0834 Clutch Pedal Switch "B" Circuit Low
P0835 Clutch Pedal Switch "B" Circuit High

Code and Meaning
P0836 Four Wheel Drive (4WD) Switch Circuit
P0837 Four Wheel Drive (4WD) Switch Circuit Range/Performance
P0838 Four Wheel Drive (4WD) Switch Circuit Low
P0839 Four Wheel Drive (4WD) Switch Circuit High P0840 Transmission Fluid Pressure Sensor/Switch "A" Circuit
P0840 Transmission Fluid Pressure Sensor/Switch "A" Circuit
P0841 Transmission Fluid Pressure Sensor/Switch "A" Circuit Range/ Performance
P0842 Transmission Fluid Pressure Sensor/Switch "A" Circuit Low
P0843 Transmission Fluid Pressure Sensor/Switch "A" Circuit High
P0844 Transmission Fluid Pressure Sensor/Switch "A" Circuit Intermittent
P0845 Transmission Fluid Pressure Sensor/Switch "B" Circuit
P0846 Transmission Fluid Pressure Sensor/Switch "B" Circuit Range/ Performance
P0847 Transmission Fluid Pressure Sensor/Switch "B" Circuit Low

Code and Meaning
P0848 Transmission Fluid Pressure Sensor/Switch "B" Circuit High
P0849 Transmission Fluid Pressure Sensor/Switch "B" Circuit Intermittent
P0850 Park/Neutral Switch Input Circuit
P0851 Park/Neutral Switch Input Circuit Low
P0852 Park/Neutral Switch Input Circuit High
P0853 Drive Switch Input Circuit
P0854 Drive Switch Input Circuit Low
P0855 Drive Switch Input Circuit High
P0856 Traction Control Input Signal
P0857 Traction Control Input Signal Range/Performance
P0858 Traction Control Input Signal Low
P0859 Traction Control Input Signal High
P0860 Gear Shift Module Communication Circuit
P0861 Gear Shift Module Communication Circuit Low

Code and Meaning
P0862 Gear Shift Module Communication Circuit High
P0863 TCM Communication Circuit
P0864 TCM Communication Circuit Range/Performance
P0865 TCM Communication Circuit Low
P0866 TCM Communication Circuit High
P0867 Transmission Fluid Pressure
P0868 Transmission Fluid Pressure Low P0869 Transmission Fluid Pressure High
P0869 Transmission Fluid Pressure High
P0870 Transmission Fluid Pressure Sensor/Switch “C” Circuit
P0871 Transmission Fluid Pressure Sensor/Switch “C” Circuit Range/ Performance
P0872 Transmission Fluid Pressure Sensor/Switch “C” Circuit Low
P0873 Transmission Fluid Pressure Sensor/Switch “C” Circuit High
P0874 Transmission Fluid Pressure Sensor/Switch “C” Circuit Intermittent

Code and Meaning
P0875 Transmission Fluid Pressure Sensor/Switch “D” Circuit
P0876 Transmission Fluid Pressure Sensor/Switch “D” Circuit Range/ Performance
P0877 Transmission Fluid Pressure Sensor/Switch “D” Circuit Low
P0878 Transmission Fluid Pressure Sensor/Switch “D” Circuit High
P0879 Transmission Fluid Pressure Sensor/Switch “D” Circuit Intermittent
P0880 TCM Power Input Signal
P0881 TCM Power Input Signal Range/Performance
P0882 TCM Power Input Signal Low
P0883 TCM Power Input Signal High
P0884 TCM Power Input Signal Intermittent
P0885 TCM Power Relay Control Circuit/Open
P0886 TCM Power Relay Control Circuit Low
P0887 TCM Power Relay Control Circuit High

Code and Meaning
P0888 TCM Power Relay Sense Circuit
P0889 TCM Power Relay Sense Circuit Range/Performance
P0890 TCM Power Relay Sense Circuit Low
P0891 TCM Power Relay Sense Circuit High
P0892 TCM Power Relay Sense Circuit Intermittent
P0893 Multiple Gears Engaged
P0894 Transmission Component Slipping
P0895 Shift Time Too Short
P0896 Shift Time Too Long
P0897 Transmission Fluid Deteriorated
P0898 Transmission Control System MIL Request Circuit Low
P0899 Transmission Control System MIL Request Circuit High
P2000 NOx Trap Efficiency Below Threshold Bank1
P2001 NOx Trap Efficiency Below Threshold Bank2

Code and Meaning
P2002 Particulate Trap Efficiency Below Threshold Bank1
P2003 Particulate Trap Efficiency Below Threshold Bank2
P2004 Intake Manifold Runner Ctrl Stuck Open Bank1
P2005 Intake Manifold Runner Ctrl Stuck Open Bank2
P2006 Intake Manifold Runner Ctrl Stuck Closed Bank1
P2007 Intake Manifold Runner Ctrl Stuck Closed Bank2
P2008 Intake Manifold Runner Ctrl Circ/Open Bank1
P2009 Intake Manifold Runner Ctrl Circ Low Bank1
P2010 Intake Manifold Runner Ctrl Circ High Bank1
P2011 Intake Manifold Runner Ctrl Circ/Open Bank2
P2012 Intake Manifold Runner Ctrl Circ Low Bank2
P2013 Intake Manifold Runner Ctrl Circ High Bank2
P2014 Intake Manifold Runner Pos Sensor/Switch Circ Bank1
P2015 Intake Manifold Runner Pos Sensor/Switch Circ Range/Perf Bank1

Code and Meaning
P2016 Intake Manifold Runner Pos Sensor/Switch Circ Low Bank1
P2017 Intake Manifold Runner Pos Sensor/Switch Circ High Bank1
P2018 Intake Manifold Runner Pos Sensor/Switch Circ Interm Bank1
P2019 Intake Manifold Runner Pos Sensor/Switch Circ Bank2
P2020 Intake Manifold Runner Pos Sensor/Switch Circ Range/Perf Bank2
P2021 Intake Manifold Runner Pos Sensor/Switch Circ Low Bank2
P2022 Intake Manifold Runner Pos Sensor/Switch Circ High Bank2
P2023 Intake Manifold Runner Pos Sensor/Switch Circ Interm Bank2
P2024 EVAP Fuel Vapor Temp Sensor Circ
P2025 EVAP Fuel Vapor Temp Sensor Perf
P2026 EVAP Fuel Vapor Temp Sensor Circ Low Voltage
P2027 EVAP Fuel Vapor Temp Sensor Circ High Voltage
P2028 EVAP Fuel Vapor Temp Sensor Circ Interm
P2029 Fuel Fired Heater Disabled

Code and Meaning
P2030 Fuel Fired Heater Perf
P2031 Exhaust Gas Temp Sensor Circ Bank1 Sensor 2
P2032 Exhaust Gas Temp Sensor Circ Low Bank1 Sensor 2
P2033 Exhaust Gas Temp Sensor Circ High Bank1 Sensor 2
P2034 Exhaust Gas Temp Sensor Circ Bank2 Sensor 2
P2035 Exhaust Gas Temp Sensor Circ Low Bank2 Sensor 2
P2036 Exhaust Gas Temp Sensor Circ High Bank2 Sensor 2
P2037 Reductant Inj Air Press Sensor Circ
P2038 Reductant Inj Air Press Sensor Circ Range/Perf
P2039 Reductant Inj Air Press Sensor Circ Low Input
P2040 Reductant Inj Air Press Sensor Circ High Input
P2041 Reductant Inj Air Press Sensor Circ Interm
P2042 Reductant Temp Sensor Circ
P2043 Reductant Temp Sensor Circ Range/Perf

Code and Meaning
P2044 Reductant Temp Sensor Circ Low Input
P2045 Reductant Temp Sensor Circ High Input
P2046 Reductant Temp Sensor Circ Interm
P2047 Reductant Injector Circ/Open Bank1 Unit 1
P2048 Reductant Injector Circ Low Bank1 Unit 1
P2049 Reductant Injector Circ High Bank1 Unit 1
P2050 Reductant Injector Circ/Open Bank2 Unit 1
P2051 Reductant Injector Circ Low Bank2 Unit 1
P2052 Reductant Injector Circ High Bank2 Unit 1
P2053 Reductant Injector Circ/Open Bank1 Unit 2
P2054 Reductant Injector Circ Low Bank1 Unit 2
P2055 Reductant Injector Circ High Bank1 Unit 2
P2056 Reductant Injector Circ/Open Bank2 Unit 2
P2057 Reductant Injector Circ Low Bank2 Unit 2

Code and Meaning
P2058 Reductant Injector Circ High Bank2 Unit 2
P2059 Reductant Inj Air Pump Ctrl Circ/Open
P2060 Reductant Inj Air Pump Ctrl Circ Low
P2061 Reductant Inj Air Pump Ctrl Circ High
P2062 Reductant Supply Ctrl Circ/Open
P2063 Reductant Supply Ctrl Circ Low
P2064 Reductant Supply Ctrl Circ High
P2065 Fuel Level SensorB Circ
P2066 Fuel Level SensorB Perf
P2067 Fuel Level SensorB Circ Low
P2068 Fuel Level SensorB Circ High
P2069 Fuel Level SensorB Circ Interm
P2070 Intake Manifold Tuning (IMT) Valve Stuck Open
P2071 IMT Valve Stuck Closed

Code and Meaning
P2072 Throttle Actuator Control System - Ice Blockage
P2073 Manifold Absolute Pressure/Mass Air Flow - Throttle Position Correlation at Idle
P2074 Manifold Absolute Pressure/Mass Air Flow - Throttle Position Correlation at Higher Load
P2075 IMT Valve Pos Sensor/Switch Circ
P2076 IMT Valve Pos Sensor/Switch Circ Range/Perf
P2077 IMT Valve Pos Sensor/Switch Circ Low
P2078 IMT Valve Pos Sensor/Switch Circ High
P2079 IMT Valve Pos Sensor/Switch Circ Interm
P2080 Exhaust Gas Temp Sensor Circ Range/Perf Bank1 Sensor 1
P2081 Exhaust Gas Temp Sensor Circ Interm Bank1 Sensor 1
P2082 Exhaust Gas Temp Sensor Circ Range/Perf Bank2 Sensor 1
P2083 Exhaust Gas Temp Sensor Circ Interm Bank2 Sensor 1
P2084 Exhaust Gas Temp Sensor Circ Range/Perf Bank1 Sensor 2

Code and Meaning
P2085 Exhaust Gas Temp Sensor Circ Interm Bank1 Sensor 2
P2086 Exhaust Gas Temp Sensor Circ Range/Perf Bank2 Sensor 2
P2087 Exhaust Gas Temp Sensor Circ Interm Bank2 Sensor 2
P2088 A Camshaft Pos Actuator Ctrl Circ Low Bank1
P2089 A Camshaft Pos Actuator Ctrl Circ High Bank1
P2090 B Camshaft Pos Actuator Ctrl Circ Low Bank1
P2091 B Camshaft Pos Actuator Ctrl Circ High Bank1
P2092 A Camshaft Pos Actuator Ctrl Circ Low Bank2
P2093 A Camshaft Pos Actuator Ctrl Circ High Bank2
P2094 B Camshaft Pos Actuator Ctrl Circ Low Bank2
P2095 B Camshaft Pos Actuator Ctrl Circ High Bank2
P2096 Post Catalyst Fuel Trim Sys Too Lean Bank1
P2097 Post Catalyst Fuel Trim Sys Too Rich Bank1
P2098 Post Catalyst Fuel Trim Sys Too Lean Bank2

Code and Meaning
P2099 Post Catalyst Fuel Trim Sys Too Rich Bank2
P2100 Throttle Actuator Ctrl Motor Circ/Open
P2101 Throttle Actuator Ctrl Motor Circ Range/Perf
P2102 Throttle Actuator Ctrl Motor Circ Low
P2103 Throttle Actuator Ctrl Motor Circ High
P2104 Throttle Actuator Ctrl Sys-Forced Idle
P2105 Throttle Actuator Ctrl Sys-Forced Engine Shutdown
P2106 Throttle Actuator Ctrl Sys-Forced Limited Power
P2107 Throttle Actuator Ctrl Mod Processor
P2108 Throttle Actuator Ctrl Mod Perf
P2109 Throttle/Pedal Pos SensorA Minimum Stop Perf
P2110 Throttle Actuator Ctrl Sys-Forced Limited RPM
P2111 Throttle Actuator Ctrl Sys-Stuck Open
P2112 Throttle Actuator Ctrl Sys-Stuck Closed

Code and Meaning
P2113 Throttle/Pedal Pos SensorB Minimum Stop Perf
P2114 Throttle/Pedal Pos Sensor C Minimum Stop Perf
P2115 Throttle/Pedal Pos Sensor D Minimum Stop Perf
P2116 Throttle/Pedal Pos Sensor E Minimum Stop Perf
P2117 Throttle/Pedal Pos Sensor F Minimum Stop Perf
P2118 Throttle Actuator Ctrl Motor Current Range/Perf
P2119 Throttle Actuator Ctrl Throttle Body Range/Perf
P2120 Throttle/Pedal Pos Sensor/Switch D Circ
P2121 Throttle/Pedal Pos Sensor/Switch D Circ Range/Perf
P2122 Throttle/Pedal Pos Sensor/Switch D Circ Low Input
P2123 Throttle/Pedal Pos Sensor/Switch D Circ High Input
P2124 Throttle/Pedal Pos Sensor/Switch D Circ Interm
P2125 Throttle/Pedal Pos Sensor/Switch E Circ
P2126 Throttle/Pedal Pos Sensor/Switch E Circ Range/Perf

Code and Meaning
P2127 Throttle/Pedal Pos Sensor/Switch E Circ Low Input
P2128 Throttle/Pedal Pos Sensor/Switch E Circ High Input
P2129 Throttle/Pedal Pos Sensor/Switch E Circ Interm
P2130 Throttle/Pedal Pos Sensor/Switch F Circ
P2131 Throttle/Pedal Pos Sensor/Switch F Circ Range Perf
P2132 Throttle/Pedal Pos Sensor/Switch F Circ Low Input
P2133 Throttle/Pedal Pos Sensor/Switch F Circ High Input
P2134 Throttle/Pedal Pos Sensor/Switch F Circ Interm
P2135 Throttle/Pedal Pos Sensor/Switch A / B Voltage Correlation
P2136 Throttle/Pedal Pos Sensor/Switch A / C Voltage Correlation
P2137 Throttle/Pedal Pos Sensor/Switch B / C Voltage Correlation
P2138 Throttle/Pedal Pos Sensor/Switch D / E Voltage Correlation
P2139 Throttle/Pedal Pos Sensor/Switch D / F Voltage Correlation
P2140 Throttle/Pedal Pos Sensor/Switch E / F Voltage Correlation

Code and Meaning
P2141 Exhaust Gas Recirculation Throttle Ctrl Circ Low
P2142 Exhaust Gas Recirculation Throttle Ctrl Circ High
P2143 Exhaust Gas Recirculation Vent Ctrl Circ/Open
P2144 Exhaust Gas Recirculation Vent Ctrl Circ Low
P2145 Exhaust Gas Recirculation Vent Ctrl Circ High
P2146 Fuel Injector Group A Supply Voltage Circ/Open
P2147 Fuel Injector Group A Supply Voltage Circ Low
P2148 Fuel Injector Group A Supply Voltage Circ High
P2149 Fuel Injector Group B Supply Voltage Circ/Open
P2150 Fuel Injector Group B Supply Voltage Circ Low
P2151 Fuel Injector Group B Supply Voltage Circ High
P2152 Fuel Injector Group C Supply Voltage Circ/Open
P2153 Fuel Injector Group C Supply Voltage Circ Low
P2154 Fuel Injector Group C Supply Voltage Circ High

Code and Meaning
P2155 Fuel Injector Group D Supply Voltage Circ/Open
P2156 Fuel Injector Group D Supply Voltage Circ Low
P2157 Fuel Injector Group D Supply Voltage Circ High
P2158 Vehicle Speed SensorB
P2159 Vehicle Speed SensorB Range/Perf
P2160 Vehicle Speed SensorB Circ Low
P2161 Vehicle Speed SensorB Interm/Erratic
P2162 Vehicle Speed SensorA / B Correlation
P2163 Throttle/Pedal Pos SensorA Maximum Stop Perf
P2164 Throttle/Pedal Pos SensorB Maximum Stop Perf
P2165 Throttle/Pedal Pos Sensor C Maximum Stop Perf
P2166 Throttle/Pedal Pos Sensor D Maximum Stop Perf
P2167 Throttle/Pedal Pos Sensor E Maximum Stop Perf
P2168 Throttle/Pedal Pos Sensor F Maximum Stop Perf

Code and Meaning
P2169 Exhaust Press Reg Vent Solenoid Ctrl Circ/Open
P2170 Exhaust Press Reg Vent Solenoid Ctrl Circ Low
P2171 Exhaust Press Reg Vent Solenoid Ctrl Circ High
P2172 Throttle Actuator Ctrl Sys-Sudden High Airflow Detected
P2173 Throttle Actuator Ctrl Sys-High Airflow Detected
P2174 Throttle Actuator Ctrl Sys-Sudden Low Airflow Detected
P2175 Throttle Actuator Ctrl Sys-Low Airflow Detected
P2176 Throttle Actuator Ctrl Sys-Idle Pos Not Learned
P2177 Sys Too Lean Off Idle Bank1
P2178 Sys Too Rich Off Idle Bank1
P2179 Sys Too Lean Off Idle Bank2
P2180 Sys Too Rich Off Idle Bank2
P2181 Cooling System Performance
P2182 Engine Coolant Temp Sensor 2 Circ

Code and Meaning
P2183 Engine Coolant Temp Sensor 2 Circ Range/Perf
P2184 Engine Coolant Temp Sensor 2 Circ Low
P2185 Engine Coolant Temp Sensor 2 Circ High
P2186 Engine Coolant Temp Sensor 2 Circ Interm/Erratic
P2187 Sys Too Lean at Idle Bank1
P2188 Sys Too Rich at Idle Bank1
P2189 Sys Too Lean at Idle Bank2
P2190 Sys Too Rich at Idle Bank2
P2191 Sys Too Lean at Higher Load Bank1
P2192 Sys Too Rich at Higher Load Bank1
P2193 Sys Too Lean at Higher Load Bank2
P2194 Sys Too Rich at Higher Load Bank2
P2195 O2 Sensor Signal Stuck Lean Bank1 Sensor 1
P2196 O2 Sensor Signal Stuck Rich Bank1 Sensor 1

Code and Meaning
P2197 O2 Sensor Signal Stuck Lean Bank2 Sensor 1
P2198 O2 Sensor Signal Stuck Rich Bank2 Sensor 1
P2199 Intake Air Temp Sensor 1 / 2 Correlation
P2200 NOx Sensor Circ Bank1
P2201 NOx Sensor Circ Range/Perf Bank1
P2202 NOx Sensor Circ Low Input Bank1
P2203 NOx Sensor Circ High Input Bank1
P2204 NOx Sensor Circ Interm Input Bank1
P2205 NOx Sensor Heater Ctrl Circ/Open Bank1
P2206 NOx Sensor Heater Ctrl Circ Low Bank1
P2207 NOx Sensor Heater Ctrl Circ High Bank1
P2208 NOx Sensor Heater Sense Circ Bank1
P2209 NOx Sensor Heater Sense Circ Range/Perf Bank1
P2210 NOx Sensor Heater Sense Circ Low Input Bank1

Code and Meaning
P2211 NOx Sensor Heater Sense Circ High Input Bank1
P2212 NOx Sensor Heater Sense Circ Interm Bank1
P2213 NOx Sensor Circ Bank2
P2214 NOx Sensor Circ Range/Perf Bank2
P2215 NOx Sensor Circ Low Input Bank2
P2216 NOx Sensor Circ High Input Bank2
P2217 NOx Sensor Circ Interm Input Bank2
P2218 NOx Sensor Heater Ctrl Circ/Open Bank2
P2219 NOx Sensor Heater Ctrl Circ Low Bank2
P2220 NOx Sensor Heater Ctrl Circ High Bank2
P2221 NOx Sensor Heater Sense Circ Bank2
P2222 NOx Sensor Heater Sense Circ Range/Perf Bank2
P2223 NOx Sensor Heater Sense Circ Low Bank2
P2224 NOx Sensor Heater Sense Circ High Bank2

Code and Meaning
P2225 NOx Sensor Heater Sense Circ Interm Bank2
P2228 Barometric Press Circ Low
P2229 Barometric Press Circ High
P2230 Barometric Press Circ Interm
P2232 O2 Sensor Signal Circ Shorted to Heater Circ Bank1 Sensor 2
P2233 O2 Sensor Signal Circ Shorted to Heater Circ Bank1 Sensor 3
P2234 O2 Sensor Signal Circ Shorted to Heater Circ Bank2 Sensor 1
P2235 O2 Sensor Signal Circ Shorted to Heater Circ Bank2 Sensor 2
P2236 O2 Sensor Signal Circ Shorted to Heater Circ Bank2 Sensor 3
P2237 O2 Sensor Positive Current Ctrl Circ/Open Bank1 Sensor 1
P2238 O2 Sensor Positive Current Ctrl Circ Low Bank1 Sensor 1
P2239 O2 Sensor Positive Current Ctrl Circ High Bank1 Sensor 1
P2240 O2 Sensor Positive Current Ctrl Circ/Open Bank2 Sensor 1
P2241 O2 Sensor Positive Current Ctrl Circ Low Bank2 Sensor 1

Code and Meaning
P2242 O2 Sensor Positive Current Ctrl Circ High Bank2 Sensor 1
P2243 O2 Sensor Ref Voltage Circ/Open Bank1 Sensor 1
P2244 O2 Sensor Ref Voltage Perf Bank1 Sensor 1
P2245 O2 Sensor Ref Voltage Circ Low Bank1 Sensor 1
P2246 O2 Sensor Ref Voltage Circ High Bank1 Sensor 1
P2247 O2 Sensor Ref Voltage Circ/Open Bank2 Sensor 1
P2248 O2 Sensor Ref Voltage Perf Bank2 Sensor 1
P2249 O2 Sensor Ref Voltage Circ Low Bank2 Sensor 1
P2250 O2 Sensor Ref Voltage Circ High Bank2 Sensor 1
P2251 O2 Sensor Negative Current Ctrl Circ/Open Bank1 Sensor 1
P2252 O2 Sensor Negative Current Ctrl Circ Low Bank1 Sensor 1
P2253 O2 Sensor Negative Current Ctrl Circ High Bank1 Sensor 1
P2254 O2 Sensor Negative Current Ctrl Circ/Open Bank2 Sensor 1
P2255 O2 Sensor Negative Current Ctrl Circ Low Bank2 Sensor 1

Code and Meaning
P2256 O2 Sensor Negative Current Ctrl Circ High Bank2 Sensor 1
P2257 Sec Air Inj Sys Ctrl A Circ Low
P2258 Sec Air Inj Sys Ctrl A Circ High
P2259 Sec Air Inj Sys Ctrl B Circ Low
P2260 Sec Air Inj Sys Ctrl B Circ High
P2261 T/S Charger Bypass Valve-Mechanical
P2262 Turbo Boost Press Not Detected-Mechanical
P2263 T/S Charger Boost Sys Perf
P2264 Water in Fuel Sensor Circ
P2265 Water in Fuel Sensor Circ Range/Perf
P2266 Water in Fuel Sensor Circ Low
P2267 Water in Fuel Sensor Circ High
P2268 Water in Fuel Sensor Circ Interm
P2269 Water in Fuel Condition

Code and Meaning
P2270 O2 Sensor Signal Stuck Lean Bank1 Sensor 2
P2271 O2 Sensor Signal Stuck Rich Bank1 Sensor 2
P2272 O2 Sensor Signal Stuck Lean Bank2 Sensor 2
P2273 O2 Sensor Signal Stuck Rich Bank2 Sensor 2
P2274 O2 Sensor Signal Stuck Lean Bank1 Sensor 3
P2275 O2 Sensor Signal Stuck Rich Bank1 Sensor 3
P2276 O2 Sensor Signal Stuck Lean Bank2 Sensor 3
P2277 O2 Sensor Signal Stuck Rich Bank2 Sensor 3
P2278 O2 Sensor Signals Swapped Bank1 Sensor 3 / Bank2 Sensor 3
P2279 Intake Air Sys Leak
P2280 Air Flow Restriction / Air Leak Between Air Filter and MAF
P2281 Air Leak Between MAF and Throttle Body
P2282 Air Leak Between Throttle Body and Intake Valves
P2283 Injector Ctrl Press Sensor Circ

Code and Meaning
P2284 Injector Ctrl Press Sensor Circ Range/Perf
P2285 Injector Ctrl Press Sensor Circ Low
P2286 Injector Ctrl Press Sensor Circ High
P2287 Injector Ctrl Press Sensor Circ Interm
P2288 Injector Ctrl Press Too High
P2289 Injector Ctrl Press Too High-Engine Off
P2290 Injector Ctrl Press Too Low
P2291 Injector Ctrl Press Too Low-Engine Cranking
P2292 Injector Ctrl Press Erratic
P2293 Fuel Press Reg 2 Perf
P2294 Fuel Press Reg 2 Ctrl Circ
P2295 Fuel Press Reg 2 Ctrl Circ Low
P2296 Fuel Press Reg 2 Ctrl Circ High
P2297 O2 Sensor Out of Range During Deceleration Bank1 Sensor 1

Code and Meaning
P2298 O2 Sensor Out of Range During Deceleration Bank2 Sensor 1
P2299 Brake Pedal Pos / Accelerator Pedal Pos Incompatible
P2300 Ignition Coil A Pri Ctrl Circ Low
P2301 Ignition Coil A Pri Ctrl Circ High
P2302 Ignition Coil A Sec Circ
P2303 Ignition Coil B Pri Ctrl Circ Low
P2304 Ignition Coil B Pri Ctrl Circ High
P2305 Ignition Coil B Sec Circ
P2306 Ignition Coil C Pri Ctrl Circ Low
P2307 Ignition Coil C Pri Ctrl Circ High
P2308 Ignition Coil C Sec Circ
P2309 Ignition Coil D Pri Ctrl Circ Low
P2310 Ignition Coil D Pri Ctrl Circ High
P2311 Ignition Coil D Sec Circ

Code and Meaning
P2312 Ignition Coil E Pri Ctrl Circ Low
P2313 Ignition Coil E Pri Ctrl Circ High
P2314 Ignition Coil E Sec Circ
P2315 Ignition Coil F Pri Ctrl Circ Low
P2316 Ignition Coil F Pri Ctrl Circ High
P2317 Ignition Coil F Sec Circ
P2318 Ignition Coil G Pri Ctrl Circ Low
P2319 Ignition Coil G Pri Ctrl Circ High
P2320 Ignition Coil G Sec Circ
P2321 Ignition Coil H Pri Ctrl Circ Low
P2322 Ignition Coil H Pri Ctrl Circ High
P2323 Ignition Coil H Sec Circ
P2324 Ignition Coil I Pri Ctrl Circ Low
P2325 Ignition Coil I Pri Ctrl Circ High

Code and Meaning
P2326 Ignition Coil I Sec Circ
P2327 Ignition Coil J Pri Ctrl Circ Low
P2328 Ignition Coil J Pri Ctrl Circ High
P2329 Ignition Coil J Sec Circ
P2330 Ignition Coil K Pri Ctrl Circ Low
P2331 Ignition Coil K Pri Ctrl Circ High
P2332 Ignition Coil K Sec Circ
P2333 Ignition Coil L Pri Ctrl Circ Low
P2334 Ignition Coil L Pri Ctrl Circ High
P2335 Ignition Coil L Sec Circ
P2336 Cylinder #1 Above Knock Threshold
P2337 Cylinder #2 Above Knock Threshold
P2338 Cylinder #3 Above Knock Threshold
P2339 Cylinder #4 Above Knock Threshold

Code and Meaning
P2340 Cylinder #5 Above Knock Threshold
P2341 Cylinder #6 Above Knock Threshold
P2342 Cylinder #7 Above Knock Threshold
P2343 Cylinder #8 Above Knock Threshold
P2344 Cylinder #9 Above Knock Threshold
P2345 Cylinder #10 Above Knock Threshold
P2346 Cylinder #11 Above Knock Threshold
P2347 Cylinder #12 Above Knock Threshold
P2400 EVAP Leak Detection Pump Ctrl Circ/Open
P2401 EVAP Leak Detection Pump Ctrl Circ Low
P2402 EVAP Leak Detection Pump Ctrl Circ High
P2403 EVAP Leak Detection Pump Sense Circ/Open
P2404 EVAP Leak Detection Pump Sense Circ Range/Perf
P2405 EVAP Leak Detection Pump Sense Circ Low

Code and Meaning
P2406 EVAP Leak Detection Pump Sense Circ High
P2407 EVAP Leak Detection Pump Sense Circ Interm/Erratic
P2408 Fuel Cap Sensor/Switch Circ
P2409 Fuel Cap Sensor/Switch Circ Range/Perf
P2410 Fuel Cap Sensor/Switch Circ Low
P2411 Fuel Cap Sensor/Switch Circ High
P2412 Fuel Cap Sensor/Switch Circ Interm/Erratic
P2413 Exhaust Gas Recirculation Sys Perf
P2414 O2 Sensor Exhaust Sample Error Bank1 Sensor 1
P2415 O2 Sensor Exhaust Sample Error Bank2 Sensor 1
P2416 O2 Sensor Signals Swapped Bank1 Sensor 2 / Bank1 Sensor 3
P2417 O2 Sensor Signals Swapped Bank2 Sensor 2 / Bank2 Sensor 3
P2418 EVAP Switching Valve Ctrl Circ /Open
P2419 EVAP Switching Valve Ctrl Circ Low

Code and Meaning
P2420 EVAP Switching Valve Ctrl Circ High
P2421 EVAP Vent Valve Stuck Open
P2422 EVAP Vent Valve Stuck Closed
P2423 HC Adsorption Catalyst Efficiency Below Threshold Bank1
P2424 HC Adsorption Catalyst Efficiency Below Threshold Bank2
P2425 Exhaust Gas Recirculation Cooling Valve Ctrl Circ/Open
P2426 Exhaust Gas Recirculation Cooling Valve Ctrl Circ Low
P2427 Exhaust Gas Recirculation Cooling Valve Ctrl Circ High
P2428 Exhaust Gas Temp Too High Bank1
P2429 Exhaust Gas Temp Too High Bank2
P2430 Sec Air Inj Sys Air Flow/Press Sensor Circ Bank1
P2431 Sec Air Inj Sys Air Flow/Press Sensor Circ Range/Perf Bank1
P2432 Sec Air Inj Sys Air Flow/Press Sensor Circ Low Bank1
P2433 Sec Air Inj Sys Air Flow/Press Sensor Circ High Bank1

Code and Meaning
P2434 Sec Air Inj Sys Air Flow/Press Sensor Circ Interm/Erratic Bank1
P2435 Sec Air Inj Sys Air Flow/Press Sensor Circ Bank2
P2436 Sec Air Inj Sys Air Flow/Press Sensor Circ Range/Perf Bank2
P2437 Sec Air Inj Sys Air Flow/Press Sensor Circ Low Bank2
P2438 Sec Air Inj Sys Air Flow/Press Sensor Circ High Bank2
P2439 Sec Air Inj Sys Air Flow/Press Sensor Circ Interm/Eratic Bank2
P2440 Sec Air Inj Sys Switching Valve Stuck Open Bank1
P2441 Sec Air Inj Sys Switching Valve Stuck Closed Bank1
P2442 Sec Air Inj Sys Switching Valve Stuck Open Bank2
P2443 Sec Air Inj Sys Switching Valve Stuck Closed Bank2
P2444 Sec Air Inj Sys Pump Stuck On Bank1
P2445 Sec Air Inj Sys Pump Stuck Off Bank1
P2446 Sec Air Inj Sys Pump Stuck On Bank2
P2447 Sec Air Inj Sys Pump Stuck Off Bank2

Code and Meaning
P2453 Particulate Matter Trap Differential Pressure Sensor Signal Performance
P2455 DPF Differential Pressure Sensor Short to Voltage
P2500 Generator Lamp/L-Terminal Circ Low
P2501 Generator Lamp/L-Terminal Circ High
P2502 Charging Sys Voltage
P2503 Charging Sys Voltage Low
P2504 Charging Sys Voltage High
P2505 ECM/PCM Power Input Signal
P2506 ECM/PCM Power Input Signal Range/Perf
P2507 ECM/PCM Power Input Signal Low
P2508 ECM/PCM Power Input Signal High
P2509 ECM/PCM Power Input Signal Interm
P2510 ECM/PCM Power Relay Sense Circ Range/Perf
P2511 ECM/PCM Power Relay Sense Circ Interm

Code and Meaning
P2512 Event Data Recorder Request Circ/ Open
P2513 Event Data Recorder Request Circ Low
P2514 Event Data Recorder Request Circ High
P2515 A/C Refrigerant Press SensorB Circ
P2516 A/C Refrigerant Press SensorB Circ Range/Perf
P2517 A/C Refrigerant Press SensorB Circ Low
P2518 A/C Refrigerant Press SensorB Circ High
P2519 A/C Request A Circ
P2520 A/C Request A Circ Low
P2521 A/C Request A Circ High
P2522 A/C Request B Circ
P2523 A/C Request B Circ Low
P2524 A/C Request B Circ High
P2525 Vacuum Reservoir Press Sensor Circ

Code and Meaning
P2526 Vacuum Reservoir Press Sensor Circ Range/Perf
P2527 Vacuum Reservoir Press Sensor Circ Low
P2528 Vacuum Reservoir Press Sensor Circ High
P2529 Vacuum Reservoir Press Sensor Circ Interm
P2530 Ignition Switch Run Pos Circ
P2531 Ignition Switch Run Pos Circ Low
P2532 Ignition Switch Run Pos Circ High
P2533 Ignition Switch Run/Start Pos Circ
P2534 Ignition Switch Run/Start Pos Circ Low
P2535 Ignition Switch Run/Start Pos Circ High
P2536 Ignition Switch Accessory Pos Circ
P2537 Ignition Switch Accessory Pos Circ Low
P2538 Ignition Switch Accessory Pos Circ High
P2539 Low Press Fuel Sys Sensor Circ
P2540 Low Press Fuel Sys Sensor Circ Range/Perf

Code and Meaning
P2541 Low Press Fuel Sys Sensor Circ Low
P2542 Low Press Fuel Sys Sensor Circ High
P2543 Low Press Fuel Sys Sensor Circ Interm
P2544 Torque Mgmt Request Input Signal A
P2545 Torque Mgmt Request Input Signal A Range/Perf
P2546 Torque Mgmt Request Input Signal A Low
P2547 Torque Mgmt Request Input Signal A High
P2548 Torque Mgmt Request Input Signal B
P2549 Torque Mgmt Request Input Signal B Range/Perf
P2550 Torque Mgmt Request Input Signal B Low
P2551 Torque Mgmt Request Input Signal B High
P2552 Throttle/Fuel Inhibit Circ
P2553 Throttle/Fuel Inhibit Circ Range/Perf
P2554 Throttle/Fuel Inhibit Circ Low
P2555 Throttle/Fuel Inhibit Circ High

Code and Meaning
P2556 Engine Coolant Level Sensor/Switch Circ
P2557 Engine Coolant Level Sensor/Switch Circ Range/Perf
P2558 Engine Coolant Level Sensor/Switch Circ Low
P2559 Engine Coolant Level Sensor/Switch Circ High
P2560 Engine Coolant Level Low
P2561 A/C Ctrl Mod Requested MIL Illumination
P2562 Turbocharger Boost Ctrl Pos Sensor Circ
P2563 Turbocharger Boost Ctrl Pos Sensor Circ Range/Perf
P2564 Turbocharger Boost Ctrl Pos Sensor Circ Low
P2565 Turbocharger Boost Ctrl Pos Sensor Circ High
P2566 Turbocharger Boost Ctrl Pos Sensor Circ Interm
P2567 Direct Ozone Reduction Catalyst Temp Sensor Circ
P2568 Direct Ozone Reduction Catalyst Temp Sensor Circ Range/Perf
P2569 Direct Ozone Reduction Catalyst Temp Sensor Circ Low
P2570 Direct Ozone Reduction Catalyst Temp Sensor Circ High

Code and Meaning
P2571 Direct Ozone Reduction Catalyst Temp Sensor Circ Interm/ Erratic4
P2572 Direct Ozone Reduction Catalyst Deterioration Sensor Circ4
P2573 Direct Ozone Reduction Catalyst Deterioration Sensor Circ Range/Perf4
P2574 Direct Ozone Reduction Catalyst Deterioration Sensor Circ Low4
P2575 Direct Ozone Reduction Catalyst Deterioration Sensor Circ High4
P2576 Direct Ozone Reduction Catalyst Deterioration Sensor Circ Interm/Erratic
P2577 Direct Ozone Reduction Catalyst Efficiency Below Threshold
P2600 Coolant Pump Ctrl Circ/Open
P2601 Coolant Pump Ctrl Circ Range/Perf
P2602 Coolant Pump Ctrl Circ Low
P2603 Coolant Pump Ctrl Circ High
P2604 Intake Air Heater A Circ Range/Perf
P2605 Intake Air Heater A Circ/Open
P2606 Intake Air Heater B Circ Range/Perf

Code and Meaning
P2607 Intake Air Heater B Circ Low
P2608 Intake Air Heater B Circ High
P2609 Intake Air Heater Sys Perf
P2610 ECM/PCM Internal Engine Off Timer Perf
P2611 A/C Refrigerant Distribution Valve Ctrl Circ/Open
P2612 A/C Refrigerant Distribution Valve Ctrl Circ Low
P2613 A/C Refrigerant Distribution Valve Ctrl Circ High
P2614 Camshaft Pos Signal Output Circ/Open
P2615 Camshaft Pos Signal Output Circ Low
P2616 Camshaft Pos Signal Output Circ High
P2617 Crank Pos Signal Output Circ/Open
P2618 Crank Pos Signal Output Circ Low
P2619 Crank Pos Signal Output Circ High
P2620 Throttle Pos Output Circ/Open
P2621 Throttle Pos Output Circ Low

Code and Meaning
P2622 Throttle Pos Output Circ High
P2623 Injector Ctrl Press Reg Circ/Open
P2624 Injector Ctrl Press Reg Circ Low
P2625 Injector Ctrl Press Reg Circ High
P2626 O2 Sensor Pumping Current Trim Circ/Open Bank1 Sensor 1
P2627 O2 Sensor Pumping Current Trim Circ Low Bank1 Sensor 1
P2628 O2 Sensor Pumping Current Trim Circ High Bank1 Sensor 1
P2629 O2 Sensor Pumping Current Trim Circ/Open Bank2 Sensor 1
P2630 O2 Sensor Pumping Current Trim Circ Low Bank2 Sensor 1
P2631 O2 Sensor Pumping Current Trim Circ High Bank2 Sensor 1
P2632 Fuel Pump B Ctrl Circ /Open
P2633 Fuel Pump B Ctrl Circ Low
P2634 Fuel Pump B Ctrl Circ High
P2635 Fuel Pump A Low Flow / Perf
P2636 Fuel Pump B Low Flow / Perf

Code and Meaning
P2637 Torque Mgmt Feedback Signal A
P2638 Torque Mgmt Feedback Signal A Range/Perf
P2639 Torque Mgmt Feedback Signal A Low
P2640 Torque Mgmt Feedback Signal A High
P2641 Torque Mgmt Feedback Signal B
P2642 Torque Mgmt Feedback Signal B Range/Perf
P2643 Torque Mgmt Feedback Signal B Low
P2644 Torque Mgmt Feedback Signal B High
P2645 A Rocker Arm Actuator Ctrl Circ/Open Bank1
P2646 A Rocker Arm Actuator Sys Perf or Stuck Off Bank1
P2647 A Rocker Arm Actuator Sys Stuck On Bank1
P2648 A Rocker Arm Actuator Ctrl Circ Low Bank1
P2649 A Rocker Arm Actuator Ctrl Circ High Bank1
P2650 B Rocker Arm Actuator Ctrl Circ/Open Bank1
P2651 B Rocker Arm Actuator Sys Perf or Stuck Off Bank1

Code and Meaning
P2652 B Rocker Arm Actuator Sys Stuck On Bank1
P2653 B Rocker Arm Actuator Ctrl Circ Low Bank1
P2654 B Rocker Arm Actuator Ctrl Circ High Bank1
P2655 A Rocker Arm Actuator Ctrl Circ/Open Bank2
P2656 A Rocker Arm Actuator Sys Perf or Stuck Off Bank2
P2657 A Rocker Arm Actuator Sys Stuck On Bank2
P2658 A Rocker Arm Actuator Ctrl Circ Low Bank2
P2659 A Rocker Arm Actuator Ctrl Circ High Bank2
P2660 B Rocker Arm Actuator Ctrl Circ/Open Bank2
P2661 B Rocker Arm Actuator Sys Perf or Stuck Off Bank2
P2662 B Rocker Arm Actuator Sys Stuck On Bank2
P2663 B Rocker Arm Actuator Ctrl Circ Low Bank2
P2664 B Rocker Arm Actuator Ctrl Circ High Bank2
P2665 Fuel Shutoff Valve B Ctrl Circ/Open
P2666 Fuel Shutoff Valve B Ctrl Circ Low

Code and Meaning
P2667 Fuel Shutoff Valve B Ctrl Circ High
P2668 Fuel Mode Indicator Lamp Ctrl Circ
P2669 Actuator Supply Voltage B Circ /Open
P2670 Actuator Supply Voltage B Circ Low
P2671 Actuator Supply Voltage B Circ High
P2700 Trans Friction Element A Apply Time Range/Perf
P2701 Trans Friction Element B Apply Time Range/Perf
P2702 Trans Friction Element C Apply Time Range/Perf
P2703 Trans Friction Element D Apply Time Range/Perf
P2704 Trans Friction Element E Apply Time Range/Perf
P2705 Trans Friction Element F Apply Time Range/Perf
P2706 Shift Solenoid F
P2707 Shift Solenoid F Perf or Stuck Off
P2708 Shift Solenoid F Stuck On
P2709 Shift Solenoid F Electrical

Code and Meaning
P2710 Shift Solenoid F Interm
P2711 Unexpected Mechanical Gear Disengagement
P2712 Hydraulic Power Unit Leakage
P2713 Press Ctrl Solenoid D
P2714 Press Ctrl Solenoid D Perf or Stuck Off
P2715 Press Ctrl Solenoid D Stuck On
P2716 Press Ctrl Solenoid D Electrical
P2717 Press Ctrl Solenoid D Interm
P2718 Press Ctrl Solenoid D Ctrl Circ / Open
P2719 Press Ctrl Solenoid D Ctrl Circ Range/Perf
P2720 Press Ctrl Solenoid D Ctrl Circ Low
P2721 Press Ctrl Solenoid D Ctrl Circ High
P2722 Press Ctrl Solenoid E
P2723 Press Ctrl Solenoid E Perf or Stuck Off
P2724 Press Ctrl Solenoid E Stuck On

Code and Meaning
P2725 Press Ctrl Solenoid E Electrical
P2726 Press Ctrl Solenoid E Interm
P2727 Press Ctrl Solenoid E Ctrl Circ / Open
P2728 Press Ctrl Solenoid E Ctrl Circ Range/Perf
P2729 Press Ctrl Solenoid E Ctrl Circ Low
P2730 Press Ctrl Solenoid E Ctrl Circ High
P2731 Press Ctrl Solenoid F
P2732 Press Ctrl Solenoid F Perf or Stuck Off
P2733 Press Ctrl Solenoid F Stuck On
P2734 Press Ctrl Solenoid F Electrical
P2735 Press Ctrl Solenoid F Interm
P2736 Press Ctrl Solenoid F Ctrl Circ/Open
P2737 Press Ctrl Solenoid F Ctrl Circ Range/Perf
P2738 Press Ctrl Solenoid F Ctrl Circ Low
P2739 Press Ctrl Solenoid F Ctrl Circ High

Code and Meaning
P2740 Trans Fluid Temp SensorB Circ
P2741 Trans Fluid Temp SensorB Circ Range Perf
P2742 Trans Fluid Temp SensorB Circ Low
P2743 Trans Fluid Temp SensorB Circ High
P2744 Trans Fluid Temp SensorB Circ Interm
P2745 Intermediate Shaft Speed SensorB Circ
P2746 Intermediate Shaft Speed SensorB Circ Range/Perf
P2747 Intermediate Shaft Speed SensorB Circ No Signal
P2748 Intermediate Shaft Speed SensorB Circ Interm
P2749 Intermediate Shaft Speed Sensor C Circ
P2750 Intermediate Shaft Speed Sensor C Circ Range/Perf
P2751 Intermediate Shaft Speed Sensor C Circ No Signal
P2752 Intermediate Shaft Speed Sensor C Circ Interm
P2753 Trans Fluid Cooler Ctrl Circ/Open
P2754 Trans Fluid Cooler Ctrl Circ Low

Code and Meaning
P2755 Trans Fluid Cooler Ctrl Circ High
P2756 Torq Conv Clutch Press Ctrl Solenoid
P2757 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ Perf or Stuck Off
P2758 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ Stuck On
P2759 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ Electrical
P2760 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ Interm
P2761 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ/Open
P2762 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ Range/Perf
P2763 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ High
P2764 Torq Conv Clutch Press Ctrl Solenoid Ctrl Circ Low
P2765 Input/Turbine Speed SensorB Circ
P2766 Input/Turbine Speed SensorB Circ Range/Perf
P2767 Input/Turbine Speed SensorB Circ No Signal
P2768 Input/Turbine Speed SensorB Circ Interm
P2769 Torq Conv Clutch Circ Low

Code and Meaning
P2770 Torq Conv Clutch Circ High
P2771 4WD Low Switch Circ
P2772 4WD Low Switch Circ Range/Perf
P2773 4WD Low Switch Circ Low
P2774 4WD Low Switch Circ High
P2775 Upshift Switch Circ Range/Perf
P2776 Upshift Switch Circ Low
P2777 Upshift Switch Circ High
P2778 Upshift Switch Circ Interm/Erratic
P2779 Downshift Switch Circ Range/Perf
P2780 Downshift Switch Circ Low
P2781 Downshift Switch Circ High
P2782 Downshift Switch Circ Interm/Erratic
P2783 Torq Conv Temp Too High
P2784 Input/Turbine Speed SensorA/B Correlation

Code and Meaning
P2785 Clutch Actuator Temp Too High
P2786 Gear Shift Actuator Temp Too High
P2787 Clutch Temp Too High
P2788 Auto Shift Manual Adaptive Learning at Limit
P2789 Clutch Adaptive Learning at Limit
P2790 Gate Select Direction Circ
P2791 Gate Select Direction Circ Low
P2792 Gate Select Direction Circ High
P2793 Gear Shift Direction Circ
P2794 Gear Shift Direction Circ Low
P2795 Gear Shift Direction Circ High
P2A00 O2 Sensor Circ Range/Perf Bank1 Sensor 1
P2A01 O2 Sensor Circ Range/Perf Bank1 Sensor 2
P2A02 O2 Sensor Circ Range/Perf Bank1 Sensor 3
P2A03 O2 Sensor Circ Range/Perf Bank2 Sensor 1

Code and Meaning
P2A04 O2 Sensor Circ Range/Perf Bank2 Sensor 2
P2A05 O2 Sensor Circ Range/Perf Bank2 Sensor 3
P3xxx Generic Powertrain Diagnostic Codes DTC
P3400 Cylinder Deactivation Sys Bank1
P3401 Cyl1 Deactivation/Intake Valve Ctrl Circ/Open
P3402 Cyl1 Deactivation/Intake Valve Ctrl Perf
P3403 Cyl1 Deactivation/Intake Valve Ctrl Circ Low
P3404 Cyl1 Deactivation/Intake Valve Ctrl Circ High
P3405 Cyl1 Exhaust Valve Ctrl Circ/Open
P3406 Cyl1 Exhaust Valve Ctrl Perf
P3407 Cyl1 Exhaust Valve Ctrl Circ Low
P3408 Cyl1 Exhaust Valve Ctrl Circ High
P3409 Cyl2 Deactivation/Intake Valve Ctrl Circ/Open
P3410 Cyl2 Deactivation/Intake Valve Ctrl Perf
P3411 Cyl2 Deactivation/Intake Valve Ctrl Circ Low

Code and Meaning
P3412 Cyl2 Deactivation/Intake Valve Ctrl Circ High
P3413 Cyl2 Exhaust Valve Ctrl Circ/Open
P3414 Cyl2 Exhaust Valve Ctrl Perf
P3415 Cyl2 Exhaust Valve Ctrl Circ Low
P3416 Cyl2 Exhaust Valve Ctrl Circ High
P3417 Cyl3 Deactivation/Intake Valve Ctrl Circ/Open
P3418 Cyl3 Deactivation/Intake Valve Ctrl Perf
P3419 Cyl3 Deactivation/Intake Valve Ctrl Circ Low
P3420 Cyl3 Deactivation/Intake Valve Ctrl Circ High
P3421 Cyl3 Exhaust Valve Ctrl Circ/Open
P3422 Cyl3 Exhaust Valve Ctrl Perf
P3423 Cyl3 Exhaust Valve Ctrl Circ Low
P3424 Cyl3 Exhaust Valve Ctrl Circ High
P3425 Cyl4 Deactivation/Intake Valve Ctrl Circ/Open
P3426 Cyl4 Deactivation/Intake Valve Ctrl Perf

Code and Meaning
P3427 Cyl4 Deactivation/Intake Valve Ctrl Circ Low
P3428 Cyl4 Deactivation/Intake Valve Ctrl Circ High
P3429 Cyl4 Exhaust Valve Ctrl Circ/Open
P3430 Cyl4 Exhaust Valve Ctrl Perf
P3431 Cyl4 Exhaust Valve Ctrl Circ Low
P3432 Cyl4 Exhaust Valve Ctrl Circ High
P3433 Cyl5 Deactivation/Intake Valve Ctrl Circ/Open
P3434 Cyl5 Deactivation/Intake Valve Ctrl Perf
P3435 Cyl5 Deactivation/Intake Valve Ctrl Circ Low
P3436 Cyl5 Deactivation/Intake Valve Ctrl Circ High
P3437 Cyl5 Exhaust Valve Ctrl Circ/Open
P3438 Cyl5 Exhaust Valve Ctrl Perf
P3439 Cyl5 Exhaust Valve Ctrl Circ Low
P3440 Cyl5 Exhaust Valve Ctrl Circ High
P3441 Cyl6 Deactivation/Intake Valve Ctrl Circ/Open

Code and Meaning
P3442 Cyl6 Deactivation/Intake Valve Ctrl Perf
P3443 Cyl6 Deactivation/Intake Valve Ctrl Circ Low
P3444 Cyl6 Deactivation/Intake Valve Ctrl Circ High
P3445 Cyl6 Exhaust Valve Ctrl Circ/Open
P3446 Cyl6 Exhaust Valve Ctrl Perf
P3447 Cyl6 Exhaust Valve Ctrl Circ Low
P3448 Cyl6 Exhaust Valve Ctrl Circ High
P3449 Cyl7 Deactivation/Intake Valve Ctrl Circ/Open
P3450 Cyl7 Deactivation/Intake Valve Ctrl Perf
P3451 Cyl7 Deactivation/Intake Valve Ctrl Circ Low
P3452 Cyl7 Deactivation/Intake Valve Ctrl Circ High
P3453 Cyl7 Exhaust Valve Ctrl Circ/Open
P3454 Cyl7 Exhaust Valve Ctrl Perf
P3455 Cyl7 Exhaust Valve Ctrl Circ Low
P3456 Cyl7 Exhaust Valve Ctrl Circ High

Code and Meaning
P3457 Cyl8 Deactivation/Intake Valve Ctrl Circ/Open
P3458 Cyl8 Deactivation/Intake Valve Ctrl Perf
P3459 Cyl8 Deactivation/Intake Valve Ctrl Circ Low
P3460 Cyl8 Deactivation/Intake Valve Ctrl Circ High
P3461 Cyl8 Exhaust Valve Ctrl Circ/Open
P3462 Cyl8 Exhaust Valve Ctrl Perf
P3463 Cyl8 Exhaust Valve Ctrl Circ Low
P3464 Cyl8 Exhaust Valve Ctrl Circ High
P3465 Cyl9 Deactivation/Intake Valve Ctrl Circ/Open
P3466 Cyl9 Deactivation/Intake Valve Ctrl Perf
P3467 Cyl9 Deactivation/Intake Valve Ctrl Circ Low
P3468 Cyl9 Deactivation/Intake Valve Ctrl Circ High
P3469 Cyl9 Exhaust Valve Ctrl Circ/Open
P3470 Cyl9 Exhaust Valve Ctrl Perf
P3471 Cyl9 Exhaust Valve Ctrl Circ Low

Code and Meaning
P3472 Cyl9 Exhaust Valve Ctrl Circ High
P3473 Cyl10 Deactivation/Intake Valve Ctrl Circ/Open
P3474 Cyl10 Deactivation/Intake Valve Ctrl Perf
P3475 Cyl10 Deactivation/Intake Valve Ctrl Circ Low
P3476 Cyl10 Deactivation/Intake Valve Ctrl Circ High
P3477 Cyl10 Exhaust Valve Ctrl Circ/Open
P3478 Cyl10 Exhaust Valve Ctrl Perf
P3479 Cyl10 Exhaust Valve Ctrl Circ Low
P3480 Cyl10 Exhaust Valve Ctrl Circ High
P3481 Cyl11 Deactivation/Intake Valve Ctrl Circ/Open
P3482 Cyl11 Deactivation/Intake Valve Ctrl Perf
P3483 Cyl11 Deactivation/Intake Valve Ctrl Circ Low
P3484 Cyl11 Deactivation/Intake Valve Ctrl Circ High
P3485 Cyl11 Exhaust Valve Ctrl Circ/Open
P3486 Cyl11 Exhaust Valve Ctrl Perf

Code and Meaning
P3487 Cyl11 Exhaust Valve Ctrl Circ Low
P3488 Cyl11 Exhaust Valve Ctrl Circ High
P3489 Cyl12 Deactivation/Intake Valve Ctrl Circ/Open
P3490 Cyl12 Deactivation/Intake Valve Ctrl Perf
P3491 Cyl12 Deactivation/Intake Valve Ctrl Circ Low
P3492 Cyl12 Deactivation/Intake Valve Ctrl Circ High
P3493 Cyl12 Exhaust Valve Ctrl Circ/Open

Fuel Pressure Compensation

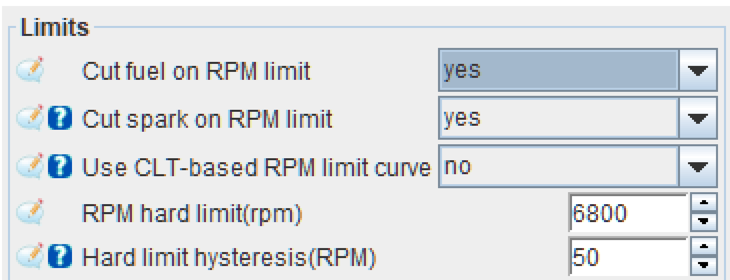
Minimum oil pressure protection

Minimum oil pressure after start

Rev limiters

Rev limiters can be found under *Base Engine > Limits and fallbacks*.

Hard cuts



Hard cuts apply when an engine reaches the *RPM hard limit*. In most cases, it is recommended to use fuel cut instead of spark cut.

Fuel cut

If *Cut fuel on RPM limit* is set to **yes**, FOME will stop injecting fuel once the engine reaches the *RPM hard limit*. The engine speed must drop below the *RPM hard limit* by the number of RPM specified in *Hard limit hysteresis* before fuel injection is resumed.

Spark cut

If *Cut spark on RPM limit* is set to **yes**, FOME will stop firing the ignition coil(s) once the engine reaches the *RPM hard limit*. The engine speed must drop below the *RPM hard limit* by the number of RPM specified in

Hard limit hysteresis before ignition is resumed.

> **DANGER**
>
> If *Cut fuel on RPM limit* is set to **no** then fuel will continue to be injected during the spark cut. This can cause a build-up of unburned fuel in the exhaust system which may ignite when ignition resumes and can cause damage to the engine and exhaust system.

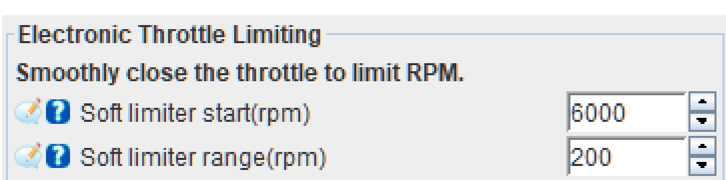
> **WARNING**
>
> Cutting spark can lead to engine valvetrain damage due to the immediate reduction in torque. It is recommended to use fuel cut instead of spark cut in most cases.

Both cut

Both fuel cut and spark cut can be used together. This reduces the risk of unburned fuel in the exhaust when using spark cut, but the risk of valvetrain damage still applies.

Soft limiting

Electronic Throttle Limiting



Engines equipped with an Electronic Throttle Body (ETB) can utilize Electronic Throttle Limiting. This will smoothly close the throttle starting at the *Soft limiter start* engine speed and fully close it once the engine speed has increased by the number of RPM defined in *Soft limiter range*.

For example, if *Soft limiter start* is 6000 RPM and *Soft limiter range* is 200 RPM, the maximum ETB position possible at 6100 RPM is 50%, and at 6200 RPM the electronic throttle will be fully closed even if the accelerator pedal is commanding a larger opening.

Basic Features

Basic Features

🗃️ Cranking control

4 items

🗃️ Idle control

6 items

📄 Bench Test Commands

📄 Boost Control

Before you start tuning the boost control, it is essential the make sure that you have a safe boost cut pressure set under Base Engine > Limits and Fa…

📄 Cylinder Bank Selection

📄 Data logging settings

Basic Data logging settings

📄 Fan Controllers

📄 Fuel Pump Control

📄 General Purpose PWM (GPPMW)

📄 **Main Relay Control**

Cranking control

Cranking control

🗃️ **Cranking fuel**

3 items

📄 **Advanced Cranking Features**

📄 **Cranking Idle Air Control**

📄 **Cranking RPM Limit**

Cranking fuel

Cranking fuel

📄 **Cranking Duration Fuel Multiplier**

📄 **Cranking Fuel Coolant Multiplier**

📄 **Fuel priming pulse**

Cranking Duration Fuel Multiplier

Cranking Fuel Coolant Multiplier

Fuel priming pulse

Advanced Cranking Features

Cranking Idle Air Control

Cranking RPM Limit

Idle control

Idle control

📄 Advanced Idle Control Features

Idle tables for cranking taper

📄 Idle Control Hardware

Solenoids

📄 Idle Specific Ignition Table

The idle ignition table acts much like the open loop idle control in the regard that it is another open loop system contributing to the idle of the vehicle…

📄 Idle Settings

Idle settings

📄 Idle Tuning

One of the most challenging aspects of achieving optimal engine performance is idle tuning. Unlike fueling, which can be quantified with instrumenta…

📄 Idle Specific VE Table

image

Advanced Idle Control Features

Idle tables for cranking taper

Setting this value to **true** enables the use of the Idle Ignition Table and Idle VE Table during the cranking to idle taper period. See *Cranking Settings > After cranking IAC taper duration (cycles)*. If set to **false** these tables are only active when idle is detected; see Idle Detection Thresholds section of the *Idle settings* menu.

Coasting Idle tables

Setting this value to **true** will enable the *Coasting IAC Position* table. This will manually over-ride the Idle Air Control valve position during the coasting phase. This can be used to help reduce engine braking and may be useful if the engine has difficulty returning to idle.

The coasting phase is determined by first checking if the TPS (or throttle pedal position if using ETB) is less than the *TPS Threshold*. If so, FOME next checks if the current RPM is greater than the maximum idle RPM (`Idle Target RPM + RPM Upper limit`) and the engine is no longer in the *cranking IAC taper duration*. If all of these conditions are true, the engine phase is coasting and the *Coasting Idle Table* is used.

Idle Control Hardware

Solenoids

Single wire

Dual wire

PWM frequency

Stepper Motors

Dedicated stepper hardware

Step pin, direction, mode, enable etc

Dual H bridge

Direction 1/2, disable etc

PushPull outputs

Stepper a+,b+,a-,b- etc

Idle Specific Ignition Table

The idle ignition table acts much like the open loop idle control in the regard that it is another open loop system contributing to the idle of the vehicle. Adjusting the engine ignition timing at idle changes the engine torque output where an increase in timing produces more torque and a reduction reduces the torque. This change in torque can be used to adjust the engine speed and achieve a stable idle by reducing the timing above the idle RPM target and increasing it below the target.

Before tuning the ignition table, make sure to have your base open loop idle position set.

To tune the ignition table, start by setting the maximum and minimum values about 500RPM above and below your target idle speed. At your target idle speed, put in your desired timing angle. 10 degrees is a good starting point but a higher timing angle such as 14 degrees will give the engine a bit more torque at idle which helps the engine to quickly rev up from idle. Start by making a linear curve between the maximum and minimum RPM values in the table with values in the range of 20 to 5 degrees. As all engines respond differently, you may want to use different starting values but these are good generalizations. Run the car with your values and see how well it maintains idle. Start to adjust the timing values so that the timing pushes and pulls the engine RPM to the target. You may need to change the shape of the curve so that only small timing adjustments are made near target and large corrections are made if the RPM significantly deviates.

Idle Settings

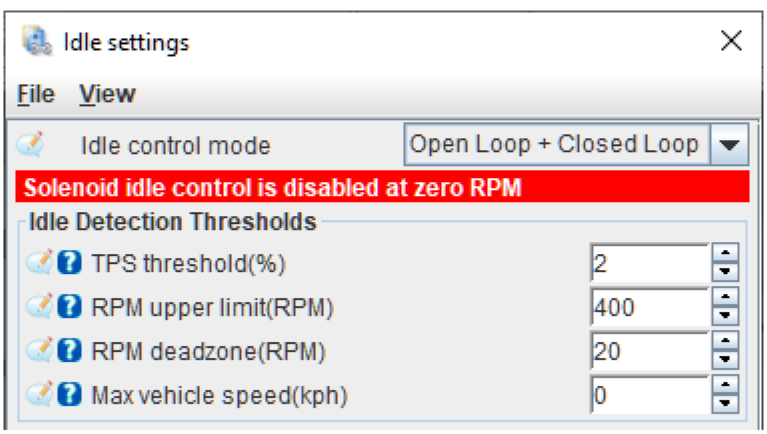
Idle settings

Idle Control mode

Open Loop disables the Closed Loop Idle strategy for the IACV/ETB. *Open Loop + Closed Loop* enables the Closed Loop Idle Strategy for the IACV/ETB. *Closed Loop Idle Ignition Timing* can be enabled separately.

Idle Detection Thresholds

The below settings are utilized to prevent the closed loop idle air strategy from engaging when it is not desired.



TPS Threshold

The TPS value must be below this % before the idle state can be entered. When using an Electronic Throttle Body (ETB) this references the

Accelerator Pedal Position sensor instead.

RPM upper limit

The engine speed must be at or below the *idle target RPM* plus this value before the idle state can be entered. For example, if the *idle target RPM* is 750 RPM and this value is set to 200, the engine speed must be at or below 950 RPM before closed loop idle can be entered.

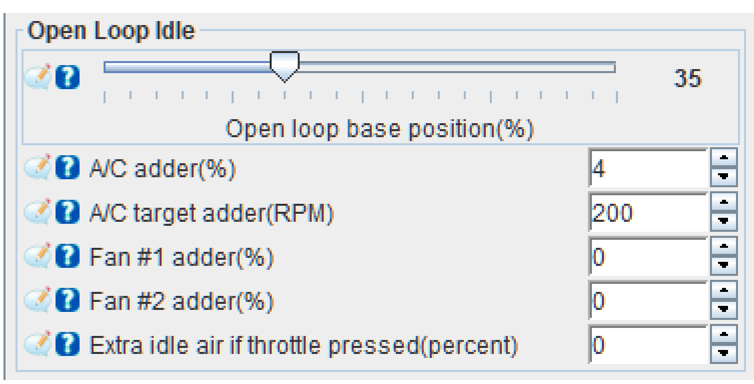
RPM deadzone

When the engine speed is within this many RPM of the target idle RPM, the closed loop idle algorithm is disabled. This is to prevent unwanted oscillation around the target.

Max vehicle speed

The VSS must be at or below this speed before the idle state can be entered. Setting this to 0 disables this check and enables closed loop idle air control at any speed.

Open Loop Idle



Open loop base position

Used to set the base value for idle control. Typically set to provide an idle slightly above the target RPM when the engine is fully warmed up. Log variable is *Idle: Open loop*.

A/C adder

This percentage is added to the *open loop base position* when the A/C is active, used to compensate for the additional load the A/C compressor puts on the engine.

A/C target adder

Added to the closed loop *Idle Target RPM* when the A/C is active, can be used to ensure the compressor is spinning quickly enough for desired cooling.

Fan #1 adder, Fan #2 adder

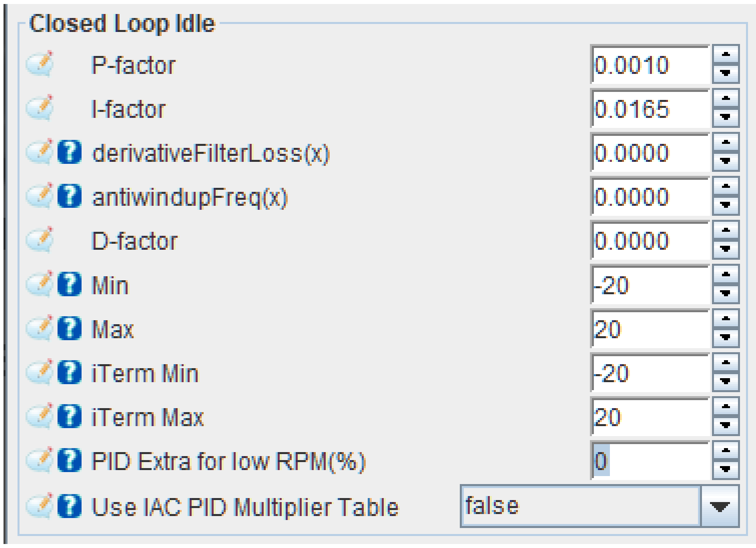
Added to the *open loop base position* when the fans are activated. May compensate for the additional electrical load on the alternator.

Extra idle air if throttle pressed

Closed Loop Idle

The closed loop idle air strategy can utilize all three terms (Proportional, Integral, Derivative) to provide accurate control of the IACV or ETB position and achieve a stable target idle. The 3 terms are calculated, added up and then limited by the *Min* or *Max* values to arrive at the final

Idle: Closed Loop output. The error (difference between the *Idle Target RPM* and actual RPM) is written on the variable *idleStatus_Error*.



P-factor

Sets the Proportional gain of the closed loop idle air strategy, used to generate the P Term. Unity gain results in a P term output equal to the error, `(Idle Target RPM - actual RPM)`. Log variable is *idleStatus_pTerm*.

For example, an instantaneous error of 100 RPM with a P-factor gain of 0.5 would result in a P term output of 50%.

I-factor

Sets the Integral gain of the closed loop idle air strategy, used to generate the I Term. Unity gain results in an I term output equal to the error/second, `(Idle Target RPM - actual RPM)/sec`. Log variable is *idleStatus_iTerm*.

For example, a consistent error of 100 RPM with an I-factor gain of 0.01

would result in an I term output that increases by 1% every second.

derivativeFilterLoss

antiwindupFreq

Used to limit the Integral term (iTerm) windup when the closed loop idle air strategy output is being limited by the min/max duty cycle limit. Once the Integral term has been calculated and limited if appropriate (*iTerm Min*, *iTerm Max*), if the output of the closed loop idle air PID controller exceeds the overall PID *Min* or *Max* settings, the Integral term is further limited to prevent integral windup. As long as the output of the closed loop idle air strategy exceeds the limits set by *Min* or *Max*, the I term is continuously modified.

```
iTerm += dTime(sec) * antiwindupFreq * (ClosedLoopLimitedOutput - ClosedLoopOutput)
```

> **INFO**
>
> *dTime* is the delta time since the closed loop idle air PID controller last ran.

Keep in mind that the Integral term is updated every time the closed loop idle air PID control strategy runs and will continue to be modified based on the error and *I-factor* gain. Additionally, if *ClosedLoopLimitedOutput* equals *ClosedLoopOutput*, antiwindupFreq has no effect.

D-factor

Sets the Derivative gain of the closed loop idle air strategy, used to generate the D Term. Log variable is *idleStatus_dTerm*.

Min, Max

Sets the minimum and maximum duty cycle modifier that can be commanded by the closed loop idle air strategy. The P, I and D terms are added up then limited based on the *Min* and *Max* values to give the *Idle: Closed loop* output. This is then added to the *open loop base position* to result in the final output, *Idle: Position*.

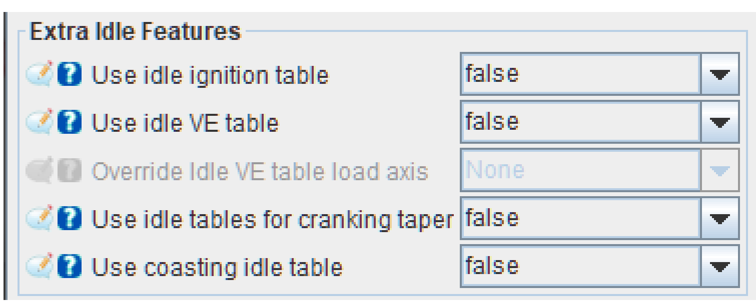
iTerm Min, iTerm Max

Sets the minimum and maximum duty cycle modifier for the *I-factor* specifically.

PID Extra for low RPM

Use IAC PID Multiplier Table

Extra Idle Features



Use idle ignition table

Setting this to **true** activates a separate ignition timing table (2D) for idle conditions; this can help idle stability by using ignition retard and advance either side of the desired idle speed. **false** disables the timing table.

Use idle VE table

Setting this to **true** activates a separate fuel table (3D) for idle, which allows fine tuning of the idle fuelling. **false** disables the VE table.

See the Idle VE Table page for more detail about idle-specific VE settings.

Override idle VE table load axis

Override the Y axis (load) value used for only the Idle VE table. Setting this to *none* disables the override.

Use idle tables for cranking taper

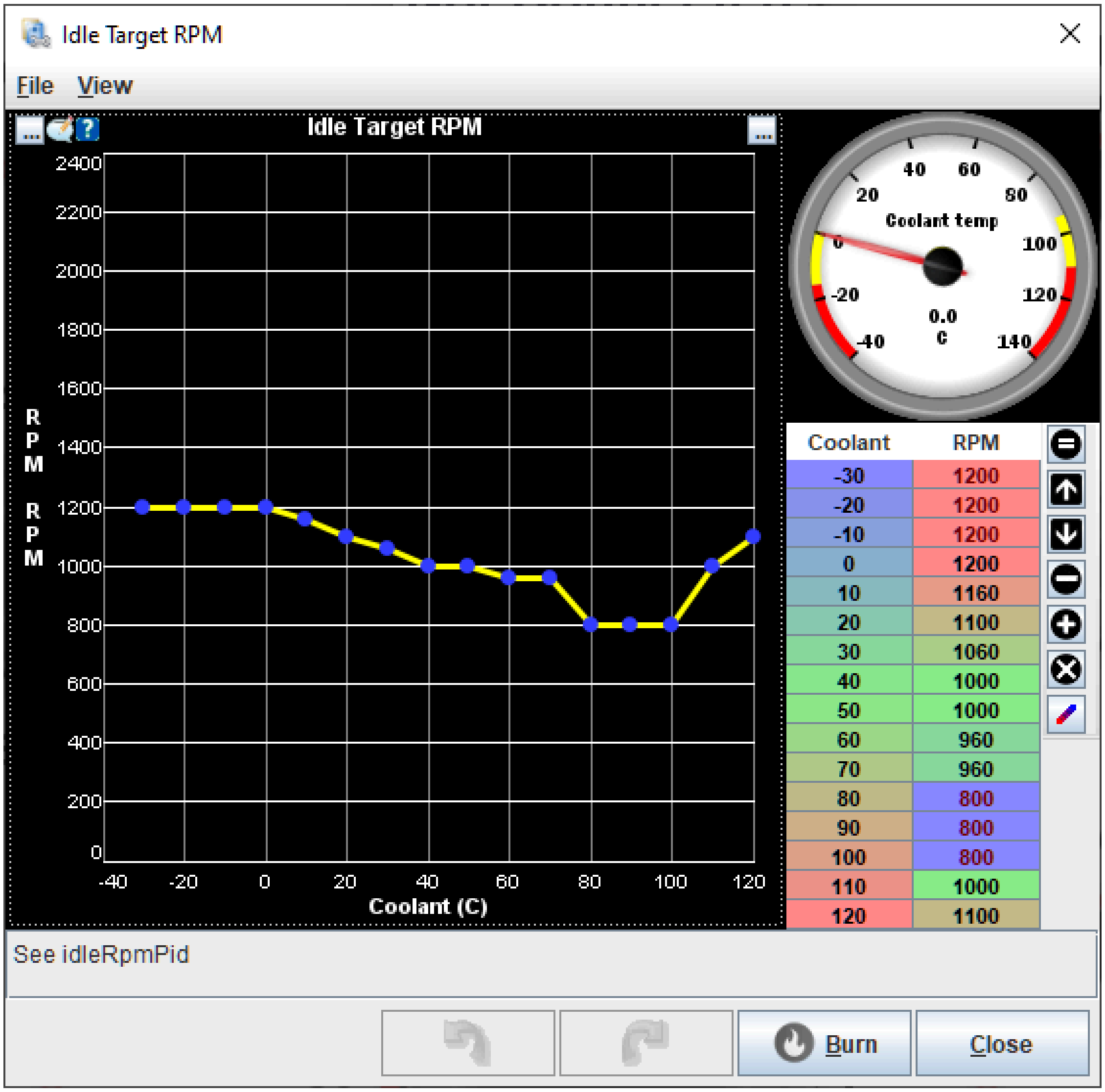
This enables the separate ignition timing and VE tables not only for idle conditions, also during the postcranking-to-idle taper transition. See *Cranking > Cranking settings > Idle air valve > After cranking IAC taper duration*.

Use coasting idle table

Override the IAC position during overrun conditions. This can be used to help reduce engine braking, for large engines in light weight cars or for engines that have trouble returning to idle.

Idle Target RPM

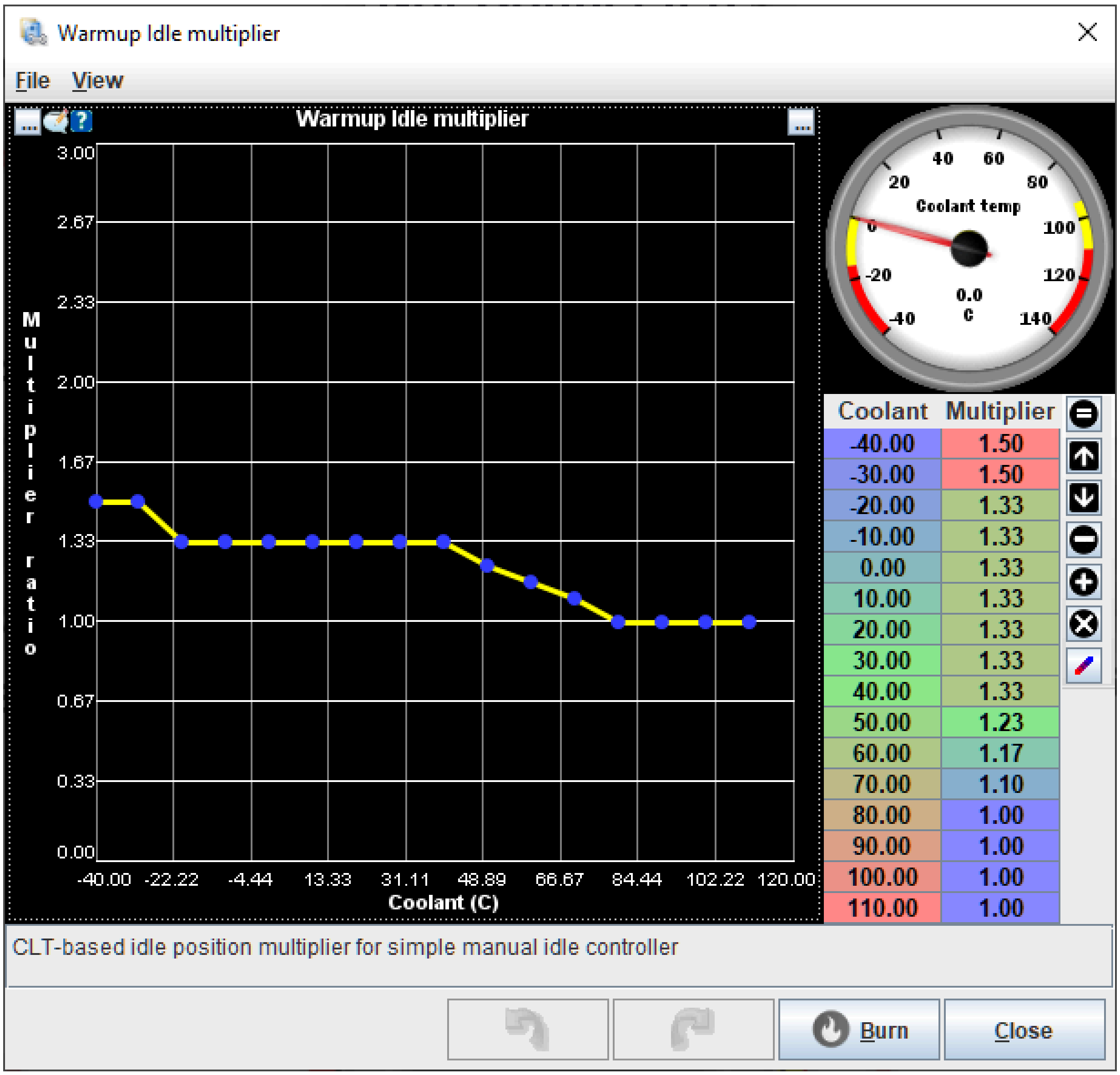
Found under *Idle > Target RPM*.



Defines the target idle RPM used by the main Closed Loop Idle air strategy as well as *Closed Loop Idle Ignition Timing*. Log variable is *Idle: Target RPM*.

Warmup idle multiplier/CLT multiplier

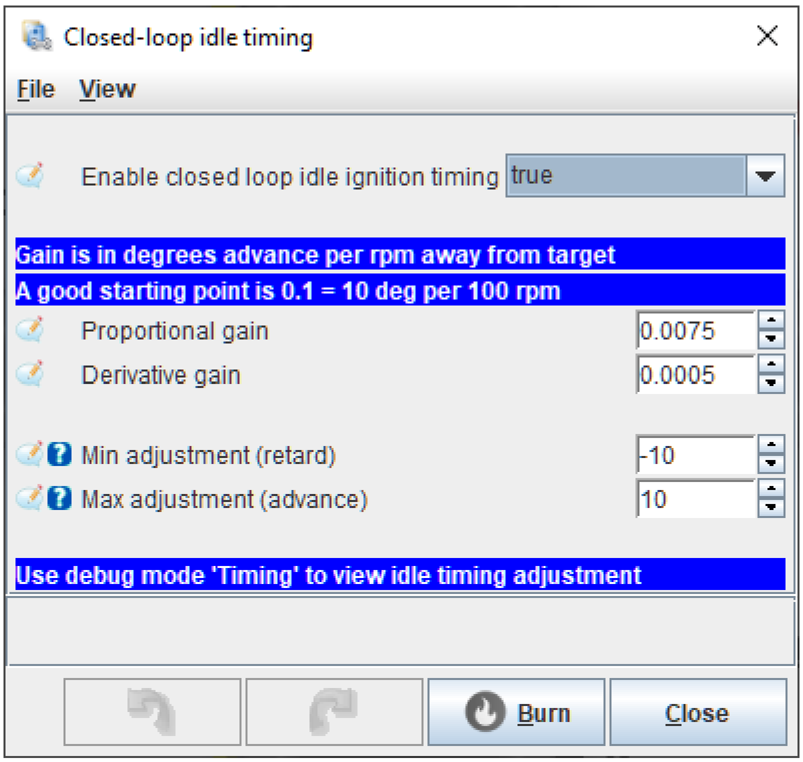
Found under *Idle > CLT Multiplier*.



The *open loop base position* value is multiplied by the value in this table. For example, if the *open loop base position* value was 30% and the multiplier was 1.50 at 0 degC, the commanded base position at 0 degC would be 45%. A multiplier of 1 would simply output the open loop base position value.

Closed-loop idle timing

Found under *Idle > Closed-loop idle timing*.



Enable closed loop idle ignition timing

True enables the closed loop idle ignition timing loop strategy, **false** disables it.

Proportional gain

Unity gain (1.0000) results in 1 deg CKA for every 1 RPM of error between actual engine speed and target engine speed. For example, a gain of 0.1000 results in 1 deg CKA for every 10 RPM of error. If the engine speed is 100 RPM below the *Idle target RPM* the resulting output would be +10 deg CKA (advance).

Derivative gain

Min Adjustment, Max Adjustment

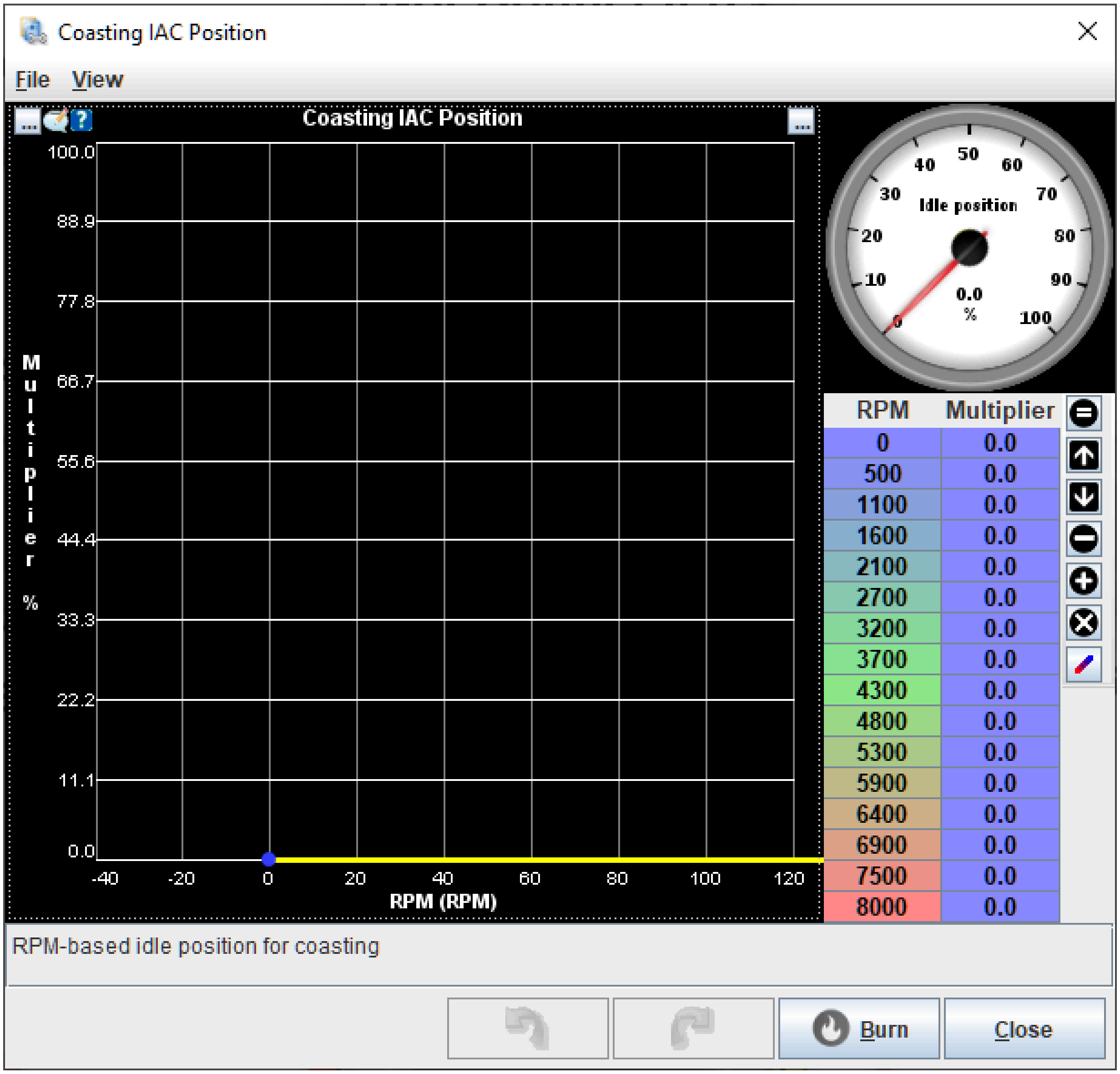
Limits the minimum and maximum amount of timing adjustment output by the *closed loop idle ignition timing* strategy. Keep these values to the minimum required to avoid excessive spark advance or retard at idle speeds.

IAC PID Multiplier

Only enabled when *Use IAC PID Multiplier Table* is set to **true**.

Coasting IAC Position

Found under *Idle > Coasting IAC Position*.



Only enabled when *Use coasting Idle Table* is set to **true**. 2D Table. When the engine is in the *coasting* phase, the IAC base position is set to the value defined by this table. This table is not used if the engine is not in the *coasting* phase and the *open loop base position* will be used instead.

> **NOTE**
>
> Despite the *multiplier* label in the table, this table sets the IAC

> position in % duty cycle just like *open loop base position* - this is not a multiplier applied to the base position.

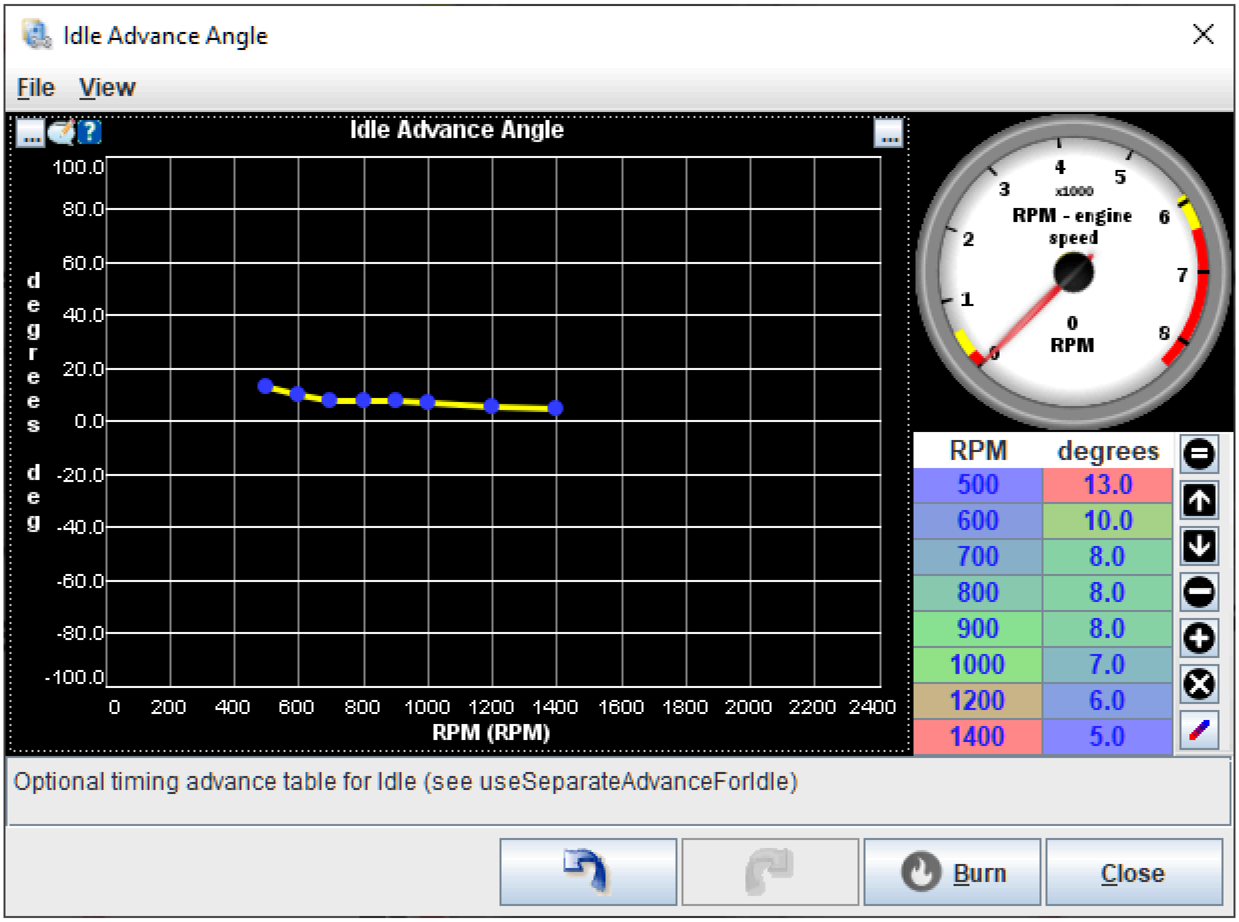
Idle VE

Only enabled when *Use idle VE table* is set to **true**. 3D table. Used in place of the regular VE table when idle is active, or during the cranking taper if *Use idle tables for cranking taper* is set to **true**.

See the Idle VE Table page for more detail about idle-specific VE settings.

Idle Ignition Advance

Found under *Idle > Ignition Advance*.



Only enabled when *Use idle ignition table* is set to **true**. 2D table (curve). The values in this table are used in place of the regular ignition curve when idle is active, or during the cranking taper if *Use idle tables for cranking taper* is set to **true**. Unlike *Closed-loop Idle Timing*, this table does not react to *Idle target RPM* and is open loop only.

Idle Tuning

One of the most challenging aspects of achieving optimal engine performance is idle tuning. Unlike fueling, which can be quantified with instrumentation like an AFR gauge, idle tuning involves tweaking a range of options to achieve a specific RPM. However, the choices made in the tuning process have a significant impact on the stability and robustness of the engine's idle.

Idle can be compared to a see-saw with three main factors affecting it: fuel, ignition, and air. These elements determine the amount of torque produced by the engine, which in turn affects idle stability. To maintain a steady idle, the torque output must also remain relatively constant. Essentially, idle represents a state of balance, where the motor's load (caused by internal friction, accessories like the alternator, AC, torque converter/trans pump, or supercharger) is counterbalanced by the amount of torque generated by the engine. When load and torque are equal, there is no net acceleration, and the RPM remains constant. In essence, this is the overall concept to grasp.

Prerequisites to a Stable Idle

The foundation for a stable idle is reliant on a motor that already runs well. Before tuning the idle, there are several features which must be tuned relatively well first.

Battery Voltage

Battery voltage tends to be lowest at idle because the alternator is spinning at a slower speed. As a result, voltage can fluctuate frequently, especially when electrical loads such as blower motors and headlights are turned on or off. It is essential to fine-tune the battery and alternator control settings to prevent sudden AFR swings due to voltage changes. This can be challenging to tune, but intentionally creating electrical loads by turning on the headlights, AC or blower fans is a good way of changing the alternator load so it can be tuned to output a near constant voltage under all conditions.

Idle Detection Thresholds

Within the idle settings, the first set of parameters to configure are the idle detection thresholds. These are a set of conditions which must be met for the ECU to establish that the car is idling. The first is the *TPS Threshold (%)* and is the maximum throttle position for idle to be enabled. After calibrating your TPS through *Tools > Calibrate TPS*, 1-2% is a good range to set this to. The *RPM upper limit* is the maximum RPM above the idle target that will still be considered idle. Typically 300-500 RPM is a good range so that the car won't move out of idle if the AC or another load requiring an RPM increase is enabled. *RPM deadzone* is the range around target in which the closed loop controllers will be disabled to prevent oscillations. 50-100 RPM is a good range here but this is discussed later. The *Max vehicle speed* dictates the maximum vehicle speed which can be considered idling. Setting this to 20-40 Kph prevents a scenario where the car is slowing down, still in gear, and the car goes into idle which tries to drag the RPM down and can lead to an idle droop when the clutch is pressed in.

Idle AFR

The quality of the fuel and ignition tuning around and inside the idle region can significantly impact idle stability. It is crucial to tune the fueling in and around the idle region to maintain a consistent AFR, as any changes in AFR due to load or RPM can also affect the torque output. It is recommended to enable the *Idle VE Table* so that the fuelling in the idle region can be more finely tuned. Like tuning the battery voltage, you want to intentionally create more loads on the engine to tune the ide VE values for different loading conditions. You can get this relatively dialled in however, it is likely that this table will need tweaking during the idle tuning process to maintain stable AFRs in idle. It is also important to merge the edges of the idle VE table with your normal VE table so that the fuelling transitions coming off idle are constant.

Idle Air Control Valve Offsets

Load can vary during idle and AC is the most significant factor that can significantly impact the engine load at idle. For engines equipped with an automatic transmission, the load in park/neutral is significantly different from that in drive. It is usually necessary to add IAC duty cycle/air to compensate for the torque required to spin the AC compressor. Again, it is crucial to ensure that AFRs don't swing excessively when moving between different idle operational areas. In the idle settings, start with about 10-15% for the *AC adder (%)* parameter. As the AC increases the alternator load, the engine torque also needs to increase when the AC is enabled so it is recommended to do this by increasing the engine RPM by 100-200 with the *AC target adder (RPM)*. The cooling fans also have associated adders (*Fan #x adder(%)*) although these don't usually need to be increased unless you notice a significant drop in RPM when the fans come on.

Open Loop Idle

Tuning the open loop idle is the first step in achieving a stable idle. In some cases, an open loop idle may be good enough for your purposes although it is highly recommended to set up a closed loop idle after perfecting the open loop. To start with, you must decide on what RPM the engine should idle at. To determine this, you must consider various factors such as flywheel and rotating inertia, driven accessories via belt or gear, noise level, personal preference, among others. It may involve an iterative process of selecting a target RPM, attempting to achieve it, and revising expectations. Generally, higher RPM idling is more manageable because the engine produces more torque. Although it is not used in the open loop idle configuration, open the *Target RPM* table in the idle settings and put in the target RPM you decided on. In the table, it is recommended to taper the idle RPM from about 200 RPM higher than target when the coolant temp is at about 20C to the actual RPM target at 60C approximately. This is to compensate for the extra drag on the engine as it heats up and the tolerances become looser.

Idle Airflow

The first step of tuning the idle RPM is to set up the open loop idle airflow. Suppose you aim to set the engine to idle at 800 RPM (minimum), the airflow must be tuned to make the engine idle 50-150RPM higher than the 800 RPM target. This can be done by adjusting the *Open loop base position (%)* parameter, the idle adjustment screw, or the throttle end stop. It is recommended to tune these adjustments so the *Open loop base position (%)* parameter (which is the base duty cycle of the IACV) is at about 30% when the engine is idling 50-150RPM above target. This gives some room for the closed loop controller to adjust the duty cycle. It's essential to ensure that AFRs remain stable as you adjust the airflow. Before enabling idle control, a stable foundation is necessary. In this example, we will say that the engine idles at 950 RPM (150 RPM above target) with 10 degrees of timing, 1.0 lambda and 30% IACV duty cycle.

> **NOTE**
>
> If you are using an electronic throttle body (ETB), the *Open loop base position* parameter is applied to the *Electronic throttle idle range* value to calculate the idle TPS value. If you need to set the *Open loop base position* significantly higher than 30%, try increasing the *Electronic throttle idle range* value until you are able to get the desired idle engine speed with an *Open loop base position* value of approximately 30%. If the inverse is true and the *Open loop base position* is too low, try decreasing the *Electronic throttle idle range* value.
>
> Find the *Electronic throttle idle range (%)* value under the *Idle > Idle hardware* menu.

Idle Ignition Table

The idle ignition table acts much like the *open loop base position* in the regard that it is another open loop system contributing to the idle of the vehicle. Adjusting the engine ignition timing at idle changes the engine torque output where an increase in timing produces more torque and a reduction reduces the torque. This change in torque can be used to adjust the engine speed and achieve a stable idle by reducing the timing above the idle RPM target and increasing it below the target.

To tune the ignition table, start by setting the maximum and minimum values about 500RPM above and below your target idle speed and interpolate the values between these. At your target idle speed in the table, put in your desired timing angle. 10 degrees is a good starting point but a higher timing angle such as 15 degrees will give the engine a bit more torque at idle which helps the engine to quickly rev up from idle. The idle timing angle will vary from engine to engine so it is always worth researching what others have used on your specific engine.

The next step is to fill in the remaining cells of the idle timing table. At the minimum RPM, a value of 20 degrees is recommended as this is roughly where maximum brake torque (MBT) is. To fill in the gaps between the minimum and target RPM, use the interpolation button. For the maximum engine RPM value, a value in the range of 0-10 degrees less than the target is recommended (do not go below 0 degrees of timing). Once again, interpolate the values between the maximum RPM and target. For the RPM values above idle target, if you have AC, it can actually be beneficial to leave the spark angles as the same as the idle target. This allows the engine to idle up and increase the torque to compensate when the AC is enabled. If the timing reduces above target, the idle air valve (when tuned for closed loop) may need to compensate more for the extra load on the engine.

Run the car with your values and see how well it maintains idle. Give it a few throttle blips to see how the idle settles. Start to adjust the timing values so that the timing pushes and pulls the engine RPM to the target and stabilizes within 50 RPM. You may need to change the shape of the idle timing curve so that only small timing adjustments are made near target and large corrections are made if the RPM significantly deviates. The trick is to essentially make a table of accurate guesses for what the timing will need to be in order to kick the engine RPM back to the target. The better your guesses are, the less work the closed loop timing controller will need to do when you implement it.

Closed Loop Idle

Closed loop idle control uses a combination of closed loop timing adjustments and idle air control valve adjustments to achieve a stable idle. Achieving a stable idle can be difficult as it requires tuning two separate controllers which operate in parallel to control the engine RPM. The best way to achieve a good idle is to use the closed loop idle timing for the larger and quicker corrections with the idle air controller acting to compensate for slower changes.

Before tuning the closed loop idle, you need to have properly tuned the idle VE, idle ignition table and open loop idle control. To tune the closed loop controllers, you also need some understanding of a closed loop controller called a PID controller. This video by RC Model Reviews perfectly explains PID controllers in basic terms.

Closed Loop Timing

It is now time to turn on closed loop idle timing. It's important to note that closed loop idle should not make significant changes all the time for a stable idle. However, closed loop idle timing adjustment is crucial for long-term stability, and it's necessary to keep the engine idling when changes occur such as AC, headlights, fans, etc. Closed loop idle timing relies heavily on the proportional gain with little to no derivative gain to act as a damper. A good place to start is with a proportional gain of 0.1 and a derivative gain of 0.05. For every engine, these values will need to be varied until the idle remains stable within 50 RPM of the target. To ensure the closed loop idle doesn't cause issues such as knocking, it's recommended to set boundaries on timing with an overall minimum and maximum adjustment from the open loop table values. A good starting point is a -5 degree minimum and +5 degree maximum but these can be increased up to 10 degrees if you need more aggressive control.

Closed Loop Idle Air

Closed loop idle air control is a powerful tool, but one that requires extreme caution as it can dramatically impact the engine RPM. Even small changes in airflow can have a significant impact, and this is compounded because the speed which an electronic throttle body or idle valves operate is much slower than spark control. To effectively use closed loop air control, a deadband of about 50-100RPM must be established around the target to allow for minor corrections to work.

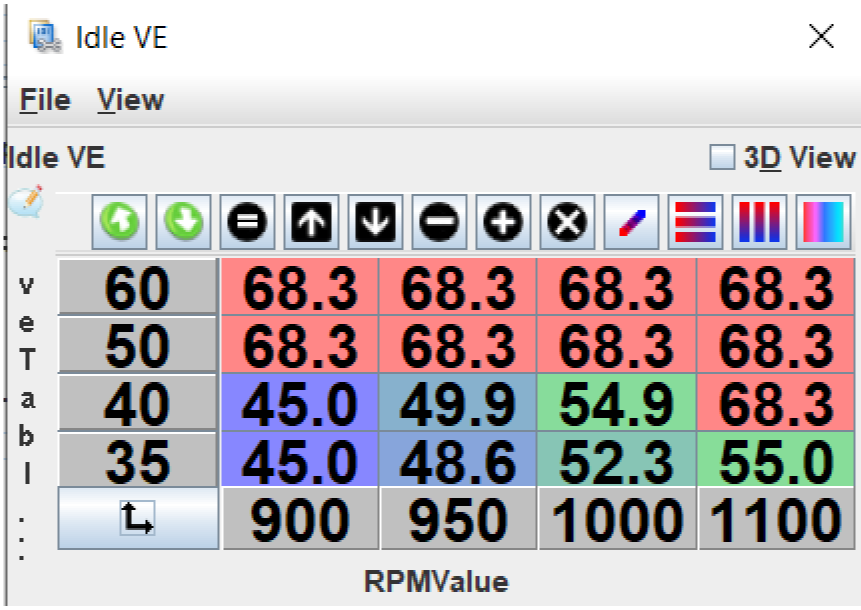
In the *Closed Loop Idle* box in the idle settings, the controller should be set up with a very small proportional gain and a sufficiently larger integral gain. This ensures that the idle air controller reacts slowly and only to relatively large disturbances which the closed loop timing cannot compensate for itself. You may need a very small derivative gain however the controller should react slowly enough that it might not be necessary.

One of the things to be wary of here is that an integral dominated controller can suffer from a phenomenon known as windup where the integral gain becomes very large if the RPM stays high or low for a long period of time. This causes the RPM to massively overshoots the target. To mitigate this, you can clamp the amount which the integral term can increase or decrease the IACV duty cycle by. A safe recommendation is to set the *iTerm Min* & *iTerm Max* to -/+10 respectively but anywhere up to -/+20 should be reasonably safe.

To assist in the overall effectiveness of the controller, the minimum and maximum duty cycle variation of the entire idle air controller can be clamped. This effectively limits how fast and with how much force the idle controller is allowed to increase or decrease the RPM. A safe range for the *Min* and *Max* is -10 and 20 respectively however these values can be changes around if the controller is responding more or less than you desire. For example, sometimes it is worth putting the maximum to a value such as 50 so that the controller has the authority to bring up the RPM if it gets sufficiently low.

Another setting of interest that should be used sparingly is the *PID Extra for low RPM(%)*. This effectively gives the controller a kick if the RPM gets too low and needs to take dramatic action. This parameter can be used to compensate for the recommended small proportional gain which is usually responsible for kicking the RPM up if it deviates too far from target. Realistically a well tuned idle controller shouldn't need this but the world isn't perfect so this serves as a band-aid for scenarios where the controller is struggling to increase RPM and is at risk of stalling. Normally, set this to 0% but if you absolutely can't fix an idle drooping problem, experiment with values up to 100% gradually increasing the percentage until the idle is kicked up to around target. Be sparing with this as a high percentage can kick the RPM way above target causing an unstable idle.

Idle Specific VE Table



The idle specific VE table is a smaller 4x4 table referenced only when the engine is operating in the idle condition. As the engine is under very little load during idle, changes in engine load can have a significant effect on the AFR. These loads can be from the air conditioning compressor engaging or from the alternator duty cycle increasing from extra electrical loads such as cooling fans, headlights or even electric windows. By adjusting the idle VE table to encapsulate the finite idle RPM and load ranges, you can tune the idle to operate close to the AFR target regardless of the engine load.

To tune the idle VE, make sure *closed loop fuel correction* under *Fuel* is disabled. Proceed to warm the car up and switch off all cooling fans, headlights and air conditioning. Tune the corresponding idle VE table cells until the AFR target is reached. Now turn on the accessories one by one, tuning the corresponding cells the idle VE table. Next repeat this process with multiple accessories on at once such as the air conditioning and headlights. Once the cells have been tuned, interpolate to fill any unused cells and smooth the cells together. In the circumstance that engaging the accessories has a large effect on the AFR, you may not be able to smooth the table much.

Bench Test Commands

Boost Control

Before you start tuning the boost control, it is essential the make sure that you have a safe boost cut pressure set under *Base Engine > Limits and Fallbacks*. Within this menu, it is also recommended to set only fuel to be cut on the RPM limit as cutting fuel and spark can lead to excess fuel igniting inside the turbine housing causing excess wear to the turbo.

Electronic boost control (EBC) is a method of controlling a vehicle's boost from an ECU or boost controller. EBC allows for more precise control of boost resulting in the ability reduce spool up time, set the boost to an exact figure or tune the boost to flatten a torque curve.

Boost Control Options

Three Port Boost Control Solenoid

A three port boost control solenoid works by quickly alternating between diverting boost pressure to atmosphere and to the wastegate. By adjusting the duty cycle or time spent diverting air to atmosphere, the solenoid effectively changes the boost pressure that the wastegate sees allowing it to stay closed for up to three times the wastegate spring pressure.

For most vehicles, this is the most common method of electronic boost control as it it cheap, can be easily plumbed into most turbos and for the most part does a good job of controlling the boost.

A three-port boost solenoid has three ports: an inlet port, an outlet port, and a control port. The inlet port is connected to the turbocharger outlet, the outlet port is connected to the wastegate actuator, and the control port is connected to the ECU. The ECU sends a signal to the control port to regulate the amount of boost pressure produced by the turbocharger.

Four Port Boost Control Solenoid

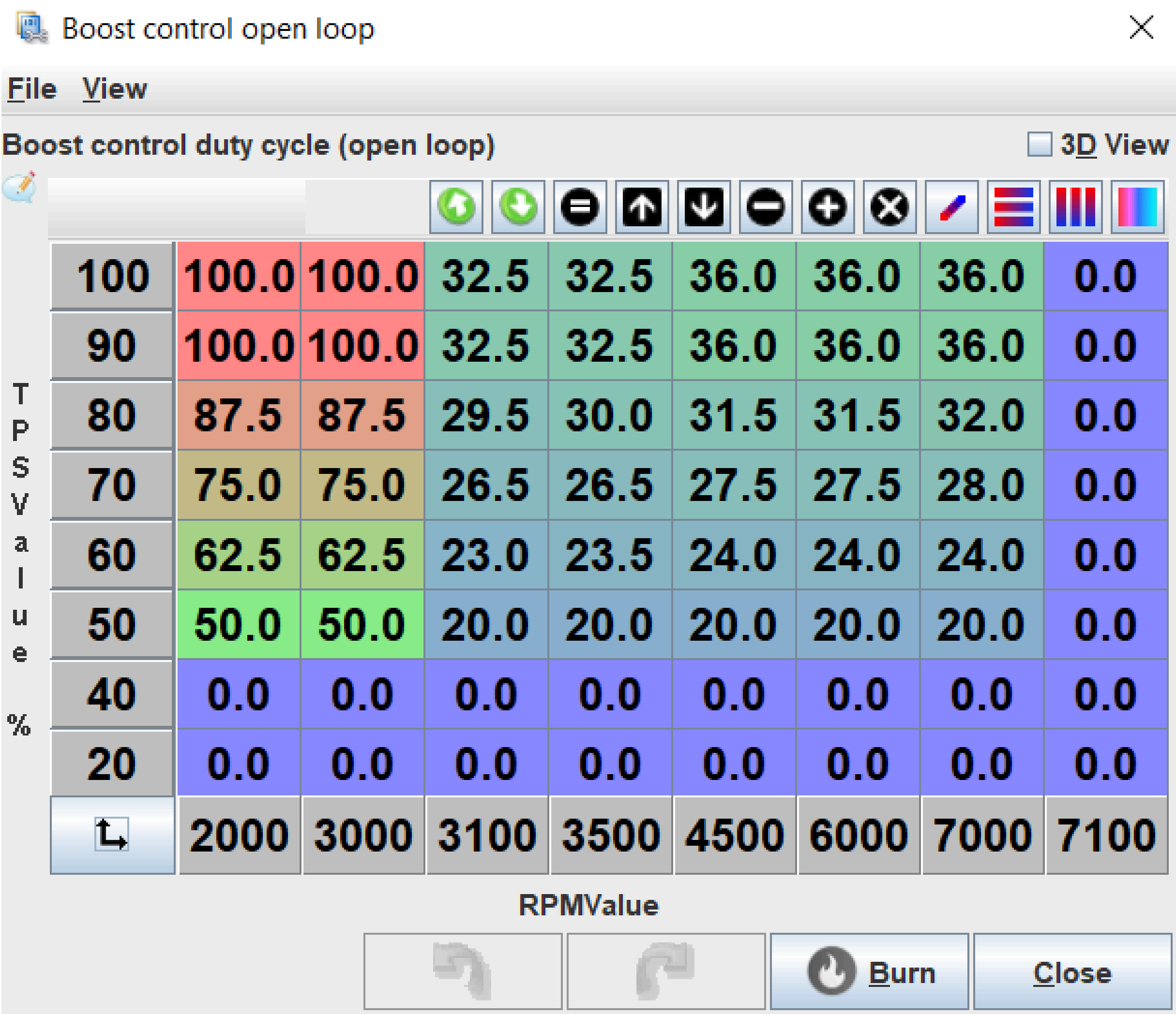
A four-port boost solenoid has an additional port, known as a reference port. The reference port is connected to the intake manifold or the atmosphere, and it provides a reference pressure to the boost solenoid. The boost solenoid uses this reference pressure to regulate the boost pressure produced by the turbocharger. The four-port design allows for more precise control over the boost pressure to a three port solenoid because it takes into account the reference pressure in addition to the control signal from the ECU. By bleeding off excess pressure when needed, the four-port boost solenoid can maintain a consistent and accurate boost pressure, resulting in better performance and reliability for the turbocharged engine.

Electric Wastegate Actuator

An electric wastegate actuator replaces a conventional spring based wastegate with an electric actuator to regulate the boost pressure.

Open Loop Boost Control

Open loop boost control is the simplest form of electronic boost control where the boost control solenoid is assigned a specific duty cycle based on the engine RPM and throttle pedal position. With open loop control, you need to guess what duty cycle will correspond with your desired boost pressure and work from there. An example of an open loop boost control table is shown below.



TPS % \ RPM	2000	3000	3100	3500	4500	6000	7000	7100
100	100.0	100.0	32.5	32.5	36.0	36.0	36.0	0.0
90	100.0	100.0	32.5	32.5	36.0	36.0	36.0	0.0
80	87.5	87.5	29.5	30.0	31.5	31.5	32.0	0.0
70	75.0	75.0	26.5	26.5	27.5	27.5	28.0	0.0
60	62.5	62.5	23.0	23.5	24.0	24.0	24.0	0.0
50	50.0	50.0	20.0	20.0	20.0	20.0	20.0	0.0
40	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Tuning Open Loop Boost Control

Before you start tuning the boost, consider what you are looking for the boost controller to do. Do you want the fastest possible spool up? Do you want the boost to come on all at once or progressively build up? Do you want to accelerate as fast as possible right on the limit of grip or to spin the tyres?

1. Decide what the minimum boost pressure you want to run will be and install the corresponding wastegate spring.

2. Disconnect any boost control solenoids and connect the wastegate directly to a source of boost pressure.

3. Verify that the boost cut works by setting it to a low value such as 100kPA. Once verified that it works, move it 10-20kPa above your desired boost pressure.

4. Take the car for a drive doing a range of pulls and take notice of how the boost responds. In first and second gear, when does the car move into boost and when does it stabilize? In higher gears, when does the car move into boost and when does it stabilize? Once the boost stabilizes, does the boost curve increase, decrease or stay flat? As an example, say that you have a turbo Miata running at sea level (100kPa atmospheric pressure) and the turbo moves into positive boost at 2000RPM and stabilizes at 3500RPM where it stays at 150kPA (50kPA or ~7PSI of boost) until redline.

5. The safest way of determining the ideal duty cycle for your boost goals is to start at the minimum and work up from there. Start by setting every cell on the open loop table to 20% duty cycle and do several full throttle pulls in 2nd-4th gear noting the boost it reaches. 20% is generally the minimum duty cycle the solenoid needs to run and this can be tested by turning the key to *On* and with the engine off, push the throttle listening for the solenoid clicking. Gradually

increase the duty cycle in steps of up to 5% until the car car reach your desired boost pressure at full throttle.

6. Once the desired boost pressure is reached, look at how the boost curve moves once it stabilizes. If it continues to rise past target pressure, progressively raise the duty cycle in the table as the RPM increases. If the boost drops, do the opposite and raise the duty cycle as the RPM increases. Continue to adjust the duty cycles until the stable boost stays within 5-10kPA of target.

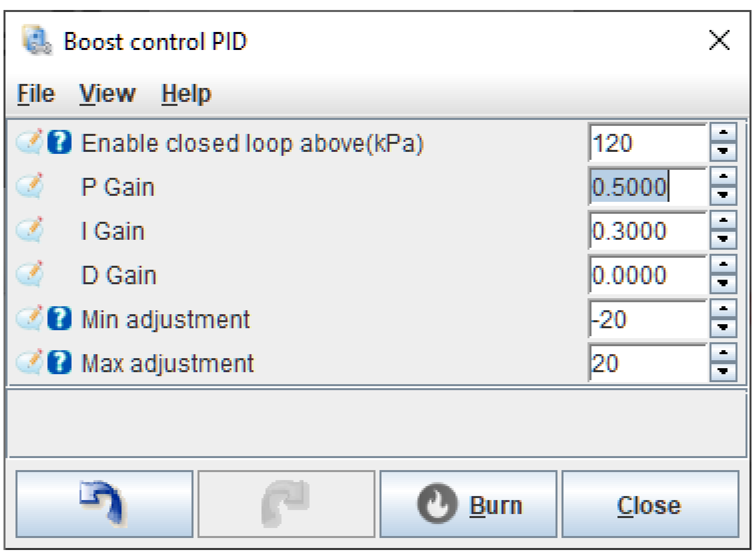
7. In your boost control table, decide on the minimum throttle position where you would like the boost control to activate. 40-50% is generally a good starting point. Below the starting point, set every cell to zero. At the starting throttle position row, set the duty cycle to the minimum duty cycle required to operate the boost solenoid which is usually 20%. Now interpolate vertically between the minimum boost throttle position row to the 100% throttle position row so that at in a given RPM column, the duty cycle will increase as the throttle is pushed more.

8. Optionally for a fast spool up time, recall what RPM the boost reached target in 3rd or 4th gear. Now rescale the table by so that the new lowest or second lowest RPM column is 1000RPM below the boost target and the second or third RPM column corresponds to the RPM when the boost target is reached. Set the columns below target boost to 100% and test it out with some pulls in different gears. Progressively raise the RPM of the 100% duty cycle columns as much as you can before the boost goes above the target pressure and leave them there. It is recommended to back off the 100% duty cycle RPM slightly from the absolute highest it can be as a factor of safety. To

make transition to boost smoother, the 100% duty cycle columns can also be vertically interpolated between 100% and a lower duty cycle such as 50% at the minimum boost RPM.

Closed Loop Boost Control

Closed loop boost control builds on the open loop boost controller and actively adjusts the solenoid duty cycle to reach the specified target pressure. The controller starts at the specified duty cycle in a given cell in the open loop table and adds or subtracts to that value until the target pressure is reached. The closed loop boost controller uses a PID controller to adjust the duty cycle. A PID controller works by measuring the error between the measured boost and the desired boost and calculating values for the P, I and D terms based off of the error. These terms are then added together to form the calculated duty cycle which the controller will measure the boost pressure response to and will continually adjust its calculated duty cycle until the target pressure is achieved.



P-Term

The P-term is calculated by multiplying the error in kPa times the Proportional gain (*P Gain*) value to generate an instantaneous output.

For example, if *P Gain* is set to 0.2 and there is a 20kPa error, it will add a value of `0.2 * 20 = 4%` to the duty cycle.

I-Term

The I-term is calculated by multiplying the error in kPa times the seconds that there is error times the Integral gain (*I Gain*) value, then adding the result to the existing I-term value for the final output.

For example, if *I Gain* is set to 0.1 and there is 10kPa of error for 2 seconds, it will add a value of `0.1 * 10 * 2 = 2%` to the duty cycle over that time period. As long as the error is present the I-term will continue to increase. Using the last example, after 4 seconds the I-term will be adding 4% to the duty cycle provided the error is still 10 kPa.

D-Term

The D-term is calculated by multiplying the change in error in kPa per second (or rate of change in kPa per second) times the Derivative gain (*D Gain*) value to generate an instantaneous output.

For example, if *D Gain* is set to 0.2 and the boost is rising at 10kPa per second, a value of `0.2 * 10 = 5%` would be subtracted from the duty cycle to slow the rate of approach to the target.

Min adjustment, Max adjustment

The *min adjustment* and *max adjustment* settings define the maximum duty cycle that can be subtracted from or added to the boost control open loop position.

Once the P, I and D terms are added together, the PID controller output is limited by the *min adjustment* and *max adjustment* values and then added to the boost control open loop value for the final output. For example, if *max adjustment* is set to 20, even with a very high gain or a very large error the maximum final duty cycle will be whatever is in the open loop table + 20%.

Cylinder Bank Selection

Data logging settings

Basic Data logging settings

SD card logging

Diagnostic and High speed logger

Fan Controllers

Fuel Pump Control

General Purpose PWM (GPPMW)

Main Relay Control

Advanced Features

Information on all sensor inputs and data outputs

🗃️ ETB

6 items

🗃️ VVT

3 items

📄 CAN

What is CAN and CANBUS?

📄 Cylinder Angle Offset Configuration

Manual configuration of the cylinder bank angles for oddfire and difficult engines.

📄 Debug Mode

Debug mode is planned to be obsoleted.

🗃️ Fuel-Related

5 items

🗃️ GDI

1 items

📄 Knock Detection Setup Guide

Choosing a Knock Sensor and Basic Setup

📄 Launch Control

Or how to break your gearbox and engine

📄 Lua Scripting

Use Lua to extend and customize firmware behavior

📄 MAP Sampling Angle

📄 Multi Dimension Mapping

Basic outline

📄 Override VE. Ignition and AFR table axis

🗃️ Rotary

1 items

📄 Smart Alternator Control

🗃️ Spark-Related

1 items

ETB

Electronic Throttle Body

Electronic Throttle Bias Table

The Electronic TB Bias Table/ETB Bias Curve (Feed Forward) is calculated using the *ETB Target* value to determine the duty bias. This duty bias is added to the output of the ETB PID loop to arrive at the final ETB duty. Values in this table range from -100% to 100%.

Idle control using ETB

All features needed to do ETB idle control discussed here, needs to cover the way ETB changes some of the basic idle settings to function differently when ETB idle is operational.

ETB PID and Autotune

PID Settings

pFactor, iFactor, dFactor

These are the Proportional, Integral and Derivative gains which apply to the Electronic Throttle Body (ETB) PID loop. Tuning these settings will affect how the ETB reacts. They output on log variables *etbStatus_pTerm*, *etbStatus_iTerm* and *etbStatus_dTerm* respectively.

pid min, pid max

When the ETB PID loop updates, the pFactor, iFactor, and dFactor outputs are added together. They are clamped by *pid min* and *pid max* and will never be less than *pid min* or greater than *pid max*.

iTermMin, iTermMax

Similar to *pid min* and *pid max*, these values apply specifically to the *iFactor* output. The result of *iFactor* will never be less than *iTermMin* or greater than *iTermMax*.

PID AutoTune

Log Variables

etbStatus_output

This value is the sum of all of the etbStatus_pTerm, etbStatus_iTerm and etbStatus_dTerm values. This value is clamped to upper and lower limits by *pid min* and *pid max*. It is added to the *ETB Bias Curve* value to give the final *ETB: Duty* output.

ETB: Duty

The sum of *etbStatus_Output* added to the *ETB Bias Curve* value.

ETB pedal target mapping

Electronic Throttle Settings

ETB enable/disable

H-bridge function

PWM frequency

ETB idle - See Idle control using ETB page

ETB PID - See ETB PID page

Redundant Sensors for ETB Position

Reason for redundancy in ETB position sensing

Redundancy provides safety in the critical sensors controlling the engine's power.

FOME tolerates up to 5% deviation in the redundant sensors, to account for slight non-linearity and inaccuracy in the sensors, and to allow for fluctuations due to environmental changes and extremes.

Throttle position and accelerator pedal position sensors support redundancy.

Fully Redundant sensors

Sensors will typically be of this type, providing redundant sensing across the entire sensing range.

Things like 0.5-4.5 volt primary sensing with delta, inverted, or half-value characteristics for the secondary sensor.

Partial Redundancy sensors

Some Ford and Toyota applications are of this type, either TPS, APPS, or both.

These sensors are linear like the fully redundant sensors, but the secondary sensor doesn't cover the full sensing range. Instead, it reaches full-scale well before the primary sensor, providing only "partial" redundancy and a overall non-linear output over the full range.

FOME supports these sensors with the `tpsSecondaryMaximum` and `ppsSecondaryMaximum` tune configurations. When necessary, these values should be tuned to indicate the percent of the primary sensor's full-scale reading at which the secondary sensor has just reached it's maximum. There are a number of techniques to determine this value, both directly/precisely and indirectly.

A direct method would be like:

1. calibrate the sensor's minimum and maximum, e.g. `tps1PrimaryMin`/`tps1PrimaryMax` and `tps1SecondaryMin`/`tps1SecondaryMax`
2. manipulate the sensor until the secondary value just reaches its maximum value, and record the primary's percentage at this point as the sensor's secondary maximum, e.g. `tpsSecondaryMaximum`

Another method is to iterate upon a close value, looking for TPS error while manipulating the sensor, and adjusting the value until the error goes away. This can lead to imprecision in the calibration however, and may lead to redundancy faults when environmental conditions change too much from that of its calibration.

Valid values are in the range of 20-100%, with `100` effectively disabling the feature. The value `0` also disables the feature, and values less than 20% are in error.

VVT

Variable Cam Timing

On/Off VVT Control

VVT PID Control

VVT Target Tables

CAN

What is CAN and CANBUS?

CAN (Controller Area Network) is a robust vehicle bus standard designed to facilitate communication among various components without a host computer. It is used in many modern cars, and is a great way to get data from the car to the **ECU**. It is also a great way to get data from the **ECU** to other devices, such as a dash, or a data logger.

CANBUS / CAN bus, or Controller Area Network Bus, refers to the physical medium (wires and connections) through which the CAN protocol operates. It's the network of interconnected CAN devices within a vehicle or industrial setup.

DBC

A **DBC** (Database Container) file is a standard file format used in the context of Controller Area Network (CAN) networks. It contains information about messages and signals that can be transmitted over a CAN network.

In a **DBC** file:

- Messages represent specific packets of data sent over the CAN network, each identified by a unique ID.
- Signals are individual pieces of data within messages, specifying the data's name, start bit, length, endianess, and other attributes. Signals

can represent information such as vehicle speed, engine RPM, or sensor readings.

FOME's DBC file

SOURCE

FOME-Tech/fome-fw/blob/master/firmware/controllers/can/FOME_CAN_verbose.dbc

Loading...

Cylinder Angle Offset Configuration

Manual configuration of the cylinder bank angles for oddfire and difficult engines.

Debug Mode

Debug mode is planned to be obsoleted.

Replacement are log channels.

Debug mode is an advanced troubleshooting feature allowing one to monitor the internal state of some FOME subsystems.

In TunerStudio, select "Base Engine Settings -> Debug Mode" and use the gauges from "Debug" category.

Barometric-Pressure-Compensation

Charge Temperature Estimation

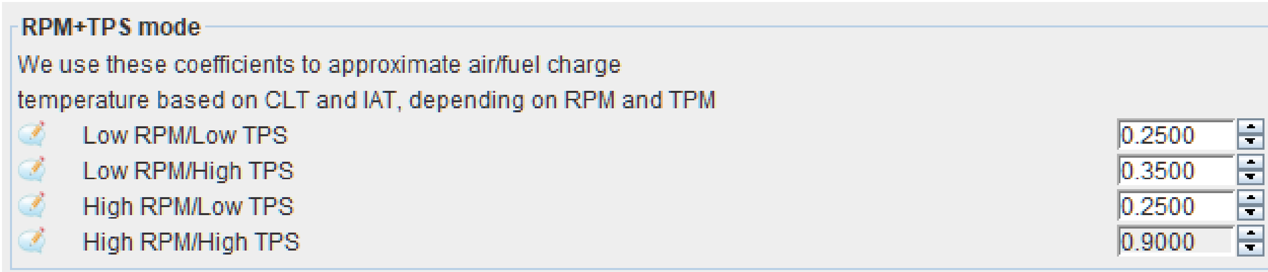
Charge temperature estimation is used to model the increase in intake air temperature after the IAT sensor by the heat of the intake piping, intake manifold, cylinder head, etc. Without accurate charge temperature estimation it is possible that an engine may exhibit leaner air fuel ratios at higher IATs, especially under low load like idle and light cruise. This is common when the VE table is tuned at lower IATs without charge temperature estimation.

Mode

Three modes of charge temperature estimation are available; *RPM+TPS*, *Air Mass Interpolation* and *Table*.

RPM+TPS (Default)

Interpolates between four coefficients based on RPM and TPS to generate a coefficient for estimating the intake air temperature. The coefficient determines the percentage of the estimated air temperature that comes from the IAT with the remainder coming from the CLT.

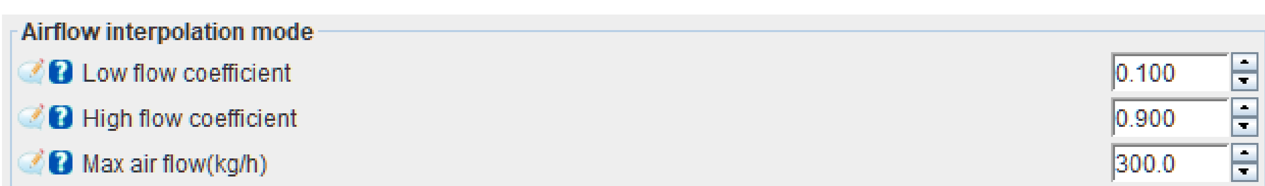


For example, a coefficient of 0.2500 means that 25% of the estimated air temperature is from the IAT, and 75% from the CLT. If the IAT was 40C and the CLT was 80C, the calculation for estimated air temperature would be `(40C * 0.25) + (80C * 0.75) = 70C`.

A coefficient of 0.9000 (which may be used at high loads/RPM) would result in the calculation `(40C * 0.90) + (80C * 0.10) = 44C`. A coefficient of 1 means that the charge temperature estimate would be the same as IAT.

Air Mass Interpolation

Also labeled as Airflow interpolation mode. This uses linear interpolation between the *low flow coefficient* at zero flow and the *high flow coefficient* at the *max air flow* to calculate the balance between IAT and CLT to arrive at the final estimated air temperature. Similar to RPM+TPS, the calculated coefficient determines what percentage of the IAT makes up the charge temperature estimate.

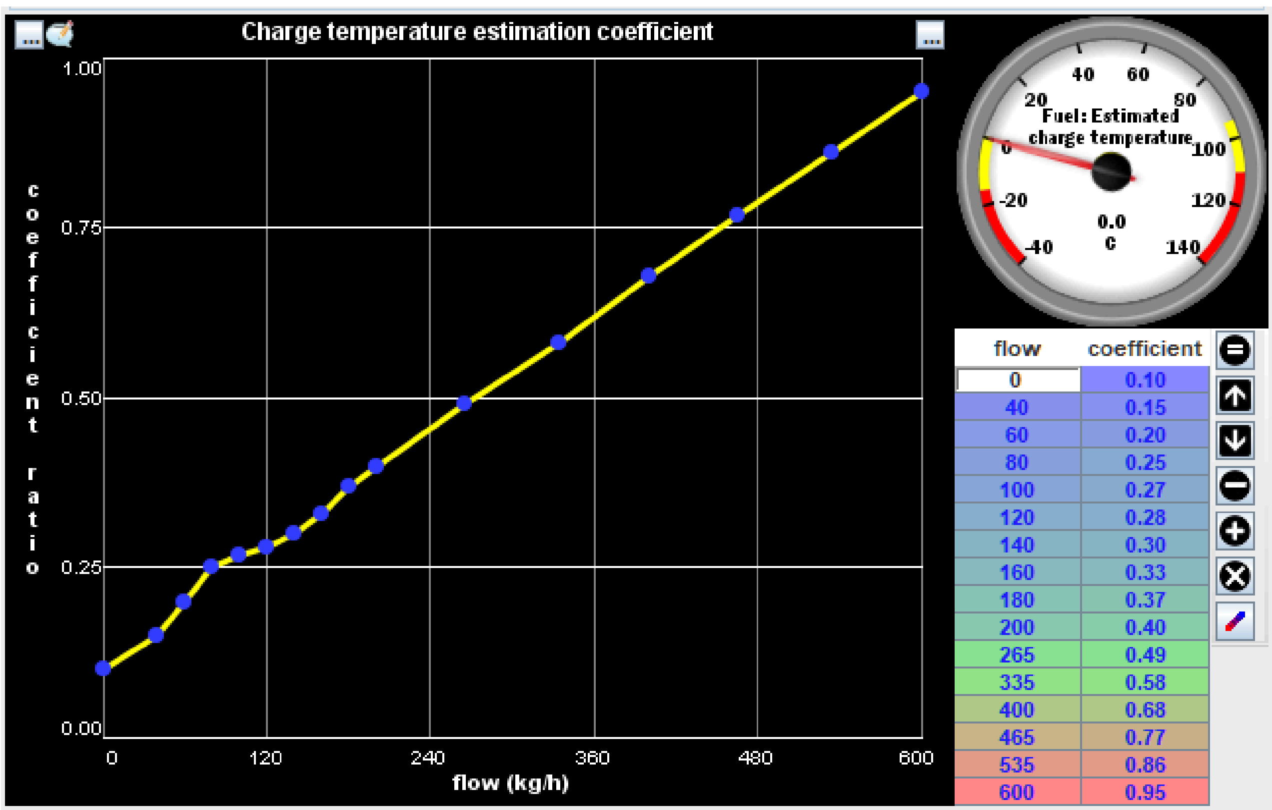


For example, a *low flow coefficient* of 0.100, a *high flow coefficent* of

0.900 and a *max air flow* of 300 kg/h would result in a calculated coefficient of 0.5 at an air flow of 150 kg/h. This means that the charge temperature estimate would be a 50/50 split between IAT and CLT. At or above *max air flow*, the calculated coefficient would be 0.9. Similar to the above examples in RPM+TPS mode (40C IAT and 80C CLT), the resultant charge temperature estimate would be `(40C * 0.90) + (80C * 0.10) = 44C` at or above *max air flow*.

Table

Uses a 2D table to return the charge temperature estimation coefficient based on the calculated flow rate. Similar to the above examples, the resultant charge temperature estimate is calculated by `(IAT * coefficient) + (CLT * [1 - coefficient])`.



Increase rate limit, Decrease rate limit

These settings control how quickly the charge temperature estimate value can increase or decrease to the new calculated value. The unit is degrees Celsius/second (degC/sec). Refer to the output *Air: Charge temperature estimate*. Note that there are two outputs with this name, one in degC and the other in degK.

For example, if the current charge temperature estimate value is `40C` and the modeled charge temperature should be `50C`, an *increase rate limit* of 1 deg/sec means that it would take 10 seconds for the charge temperature estimate value to increase from `40C` to `50C`.

Conversely, if the current charge temperature estimate is `50C` and the modeled charge temperature changed back to `40C`, a *decrease rate limit* of 5 deg/sec means that it would take 2 seconds for the charge temperature estimate value to decrease from `50C` to `40C`.

Closed Loop Fuel Correction

Injection Phase Settings

Injector Testing Mode

Not sure if we keep this?

High Pressure Fuel Pump Settings

Knock Detection Setup Guide

Choosing a Knock Sensor and Basic Setup

Knock sensors are, essentially, microphones. Some have a built in attenuation to mechanically/acoustically filter around a certain frequency (most older sensors, i.e. Gen3 GM V8) and most newer ones are "flat" response (i.e. most "donut" Bosch). As the processing power available on the OEM ECUs improved, there was a trend towards doing filtering in software and using flat-response sensors. Using a knock sensor from a newer (2010+) vehicle is probably a good idea because newer sensors are more likely to be flat-response. The algorithms in FOME are designed for flat-response sensors.

Knock sensors are going to register noise from the engine. FOME tries to separate this noise from knock from cylinder pressure by looking for specific frequencies determined by physics. It is impossible for knock sensors to not also pick up other noise. Solid engine mounts can drive knock sensing systems batshit. Tires and poor alignments can show up in knock sensor data, especially with solid mounts. Bearings in idler pulleys can show up as engine RPM or harmonic-of-RPM noise. Anything that transmits vibration into the engine will compete with the noise from combustion knock that you are trying to sniff out. Also, electrical interference can contaminate the signal from knock sensors. Most often, this comes from poor grounding or proximity of knock sensor wiring to sources of electrical noise like alternators and ignition coils or wires. Shielded wires to knock sensors can help alleviate this.

Knock sensor placement can also have a huge role. Many newer engines have multiple knock sensors for this reason. You may need to try different mounting locations. If you know that the engine you are dealing with is known for having problems on a particular cylinder, you may want to place the knock sensor closer to that cylinder.

Bottom line: to get the most out of FOME's knock control, setup and configuration for the individual setup is needed.

Basic Setup Process

1. Enable the knock sensor and calculate the knock frequency using an approximation formula.
2. Remove ignition timing from the ignition map, adjust other parameters to prevent pre-detonation events.
3. Record a log of engine performance and knock sensor levels across full RPM range.
4. Restore the tune and maps to its original configuration.
5. Review log data to create a baseline curve for knock detection.
6. Adjust knock retard aggression to determine the level of response.
7. Set up the Max Knock Retard table to define maximum allowable ignition timing retardation.

> **WARNING**
>
> The following steps assumes your ECU is running a recent release of FOME (i.e. 2312 or newer)

1. **Enable the knock sensor**, and calculate the estimated knock filter frequency (kHz). For now, configure the first setting “cylinder bore” as 0.00mm.

    - An adequate approximation formula for Knock Frequency is “Knock Frequency = 900,000/(π * 0.5 * cylinder bore diameter)”

    - Once the approximated knock frequency is calculated, use the second-order harmonics of the estimated frequency. The second-order frequency are multiples of the original calculated frequency. So twice the Knock Frequency. We do this to increase sensitivity, improve signal to noise ratio, and general “robustness” in frequency analysis.

    - For now, Set knock detection window start to 0.00, This feature is for advanced users only.

    - This formula is derived from the relationship between the speed of sound, the bore diameter, and the frequency of knock waves. It assumes that the speed of sound is approximately constant and that the knock waves travel at a specific angle through the combustion chamber. As an example, the NA6 Miata’s estimated knock frequency is 7300 Hz or 7.3kHz & its 2nd order harmonic would be 14.60Khz.

DANGER

The next few steps assume your car is running well enough to take a low-load full rpm log to define the engine knock threshold curve.

INFO

An engine knock threshold curve shows how the sensitivity of knock sensors changes with engine speed. It's a graph where the horizontal axis represents engine speed (in RPM) and the vertical axis shows the knock sensor's sensitivity level (in dBV). Tuning this curve ensures the ECU reacts appropriately to protect the engine while maximizing performance. Each setup is going to have a unique curve. Factors such as engine mounts, valvetrain components, bearings in accessories, electrical noise getting into knock wiring can affect this. Experimentally measuring the curve is the best way to get good results.

2. **Start by removing ignition timing** from the ignition map, an approach is to remove at least 3 degrees of timing, increase octane rating, decrease boost and any other parameters that can contribute to pre-detonation events. When determining a baseline knock curve, you want to be CERTAIN that the engine is not knocking.

3. **Record a log at a minimum of 100 Samples per second from idle to redline** (Under Communications/Data Rate). This can be performed in a few ways. The better the Data, the better the Threshold curve:

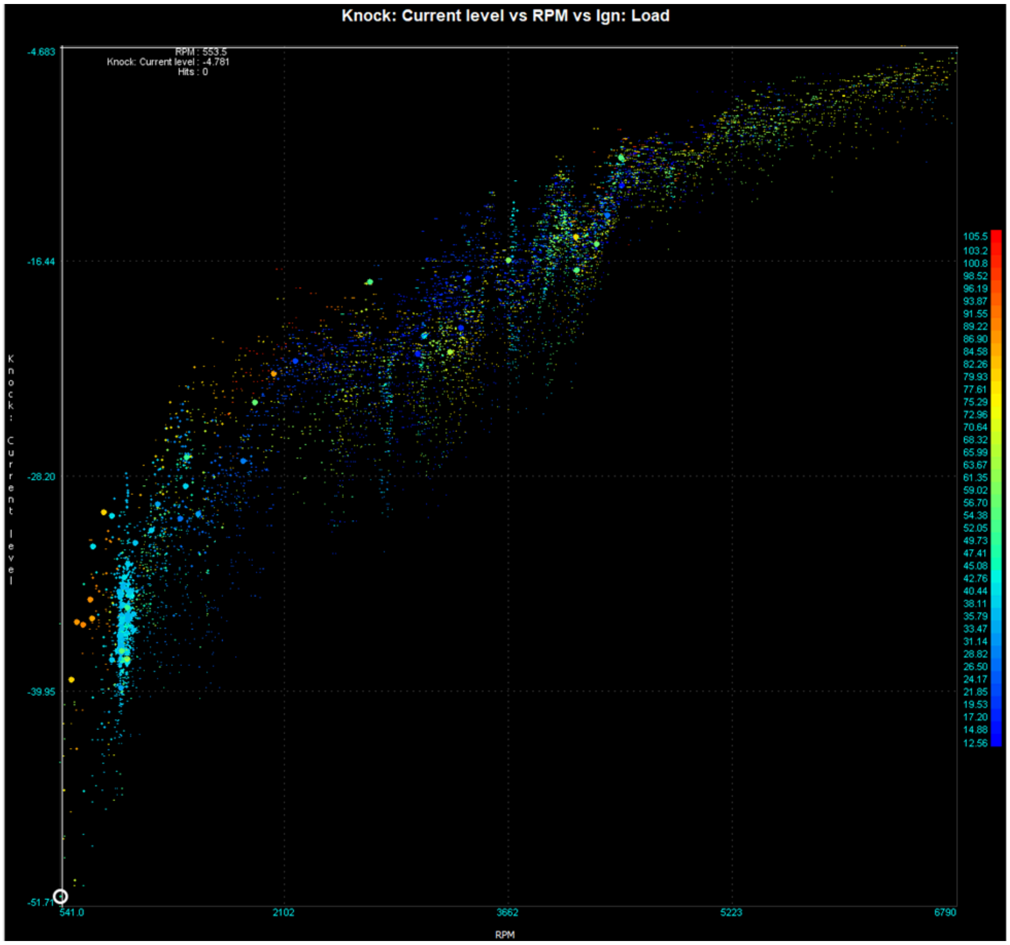
    - On Jack stands with slight load applied such as brakes

    - Driving on the road in a controlled fashion
    - Monitoring engine noise in deceleration while the injectors are cut is a valid method. This sometimes neglects noise from belt-driven accessories and should not be relied on for supercharged vehicles or vehicles with lots of belt-driven accessories.
    - It is CRITICAL that the engine is not knocking when you record this data.

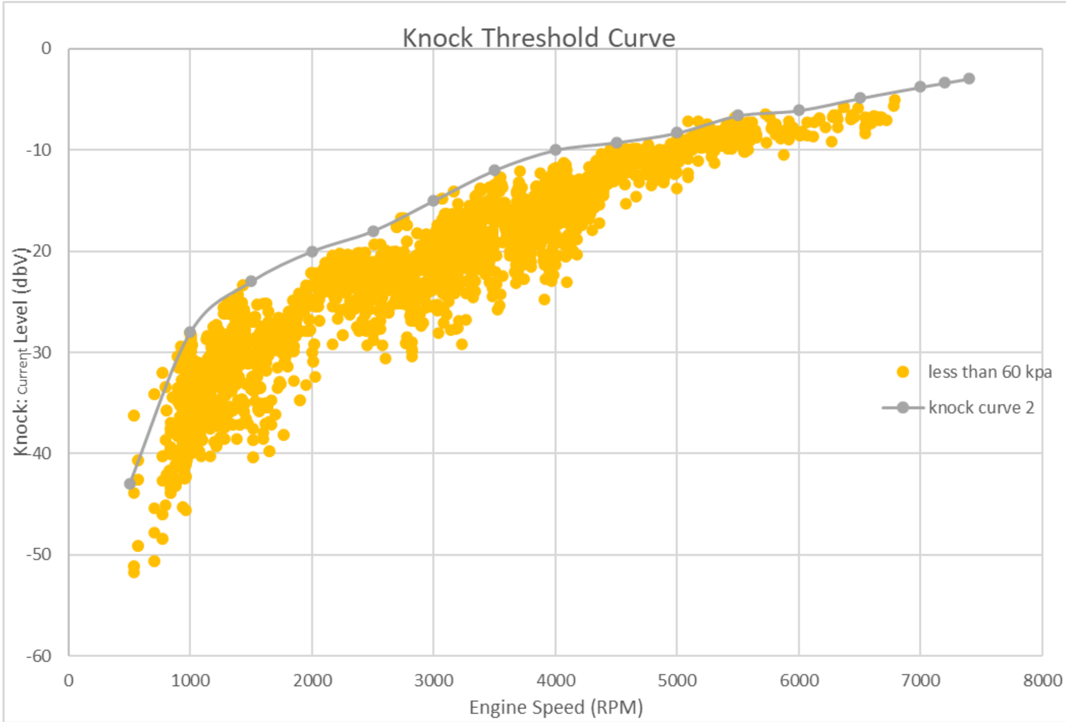
4. **Restore the tune back to the previous configuration** (restore any timing removed, adjust AFR targets, etc.).

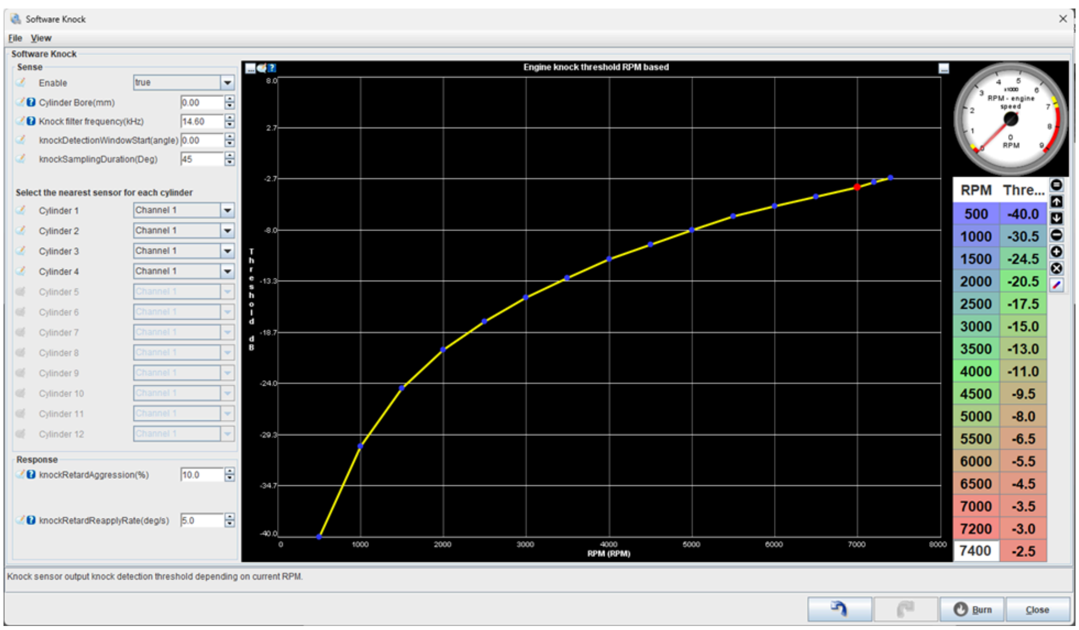
5. **Review the log in MegaLog Viewer and generate a scatter plot** of the “Knock: Current Level” vs RPM:

    - Ideally the plot is the low-load noise of the engine throughout the whole rpm range. It should look something like this below.
        - Further filtering can be applied in megalog viewer to remove high manifold pressure and deceleration noise. Use these expressions to help analysis the measured data.
            - Deceleration : “ [RPM-4]<=[Field.RPM]&&[TPS]<50 “
            - High Load : “ [Ign: Load]>60 “



- Once the above plot for your engine has been generated, we can use this data to create a plot that will be used in Tunerstudio's table on the bottom right. This is a baseline curve, further logs can be taken and more data analysis can used to refine the curve.
- The curve used in Tunerstudio should fit over the top of the low-load noise and also be "tight" to the measured data. As shown below, the orange scater is all the data less than 60kpa and the grey is a curve can be used as a baseline threshold in Tunerstudio.
  - An active system is better than one missing low level knocks





6. **Set up the Response of the Knock controller's parameter "Knock retard aggression"** A generalized rule of thumb would be 5% is considered adequate where as 15% being very aggressive:

    - The knock retardation amount is determined by calculating the distortion from the optimal ignition timing, multiplying it by the configured knock retard aggression percentage to determine the desired retardation, and then applying this retardation to the current knock retardation value.

7. **Set up the Max Knock Retard table**:

    - The max knock table defines the maximum allowable knock values that the knock controller can use to retard ignition timing, with the Y-axis representing ignition load, the X-axis engine speed in RPM, and the Z-axis indicating the degree of timing retardation permitted for each combination of load and speed.

90	0.00	8.00	8.00	8.00	12.00	8.00
70	5.00	10.00	12.00	12.00	12.00	10.00
50	10.00	10.00	15.00	15.00	15.00	15.00
40	10.00	15.00	15.00	15.00	15.00	15.00
30	10.00	15.00	15.00	15.00	15.00	15.00
20	5.00	10.00	15.00	15.00	15.00	15.00
↳	1000	2000	3000	4000	5200	6000

Recommended Threshold Settings

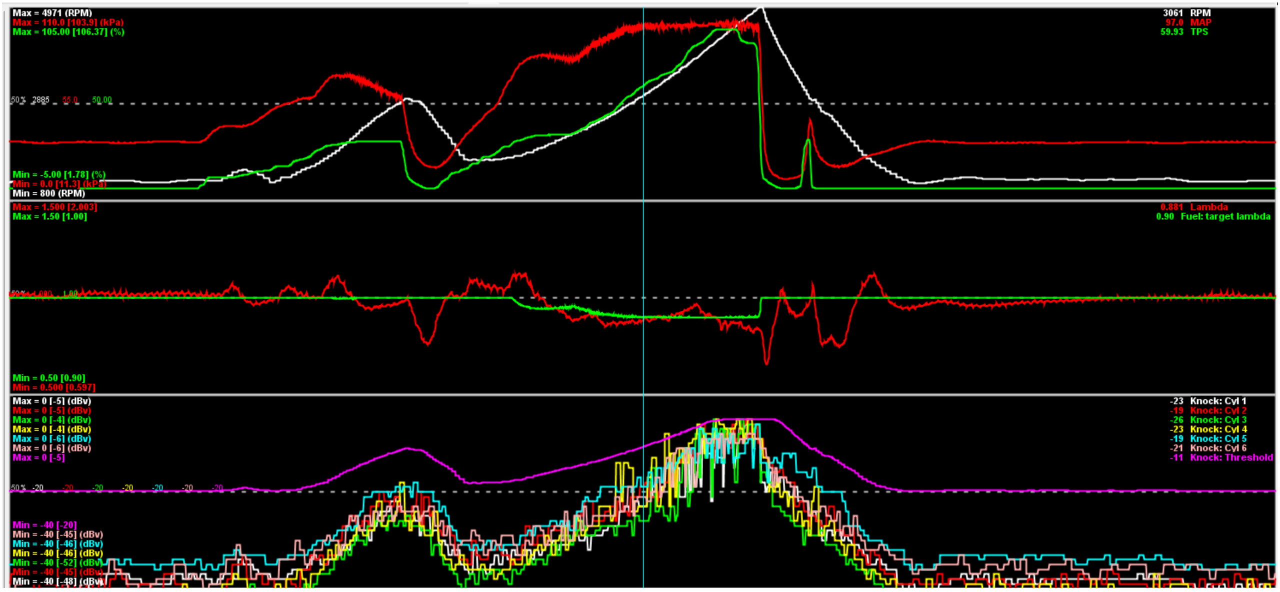
A good starting point is setting your threshold approximately 4dB above normal engine sound levels. This provides enough margin to avoid false positives while still detecting actual knock events.

Interpreting Knock Data

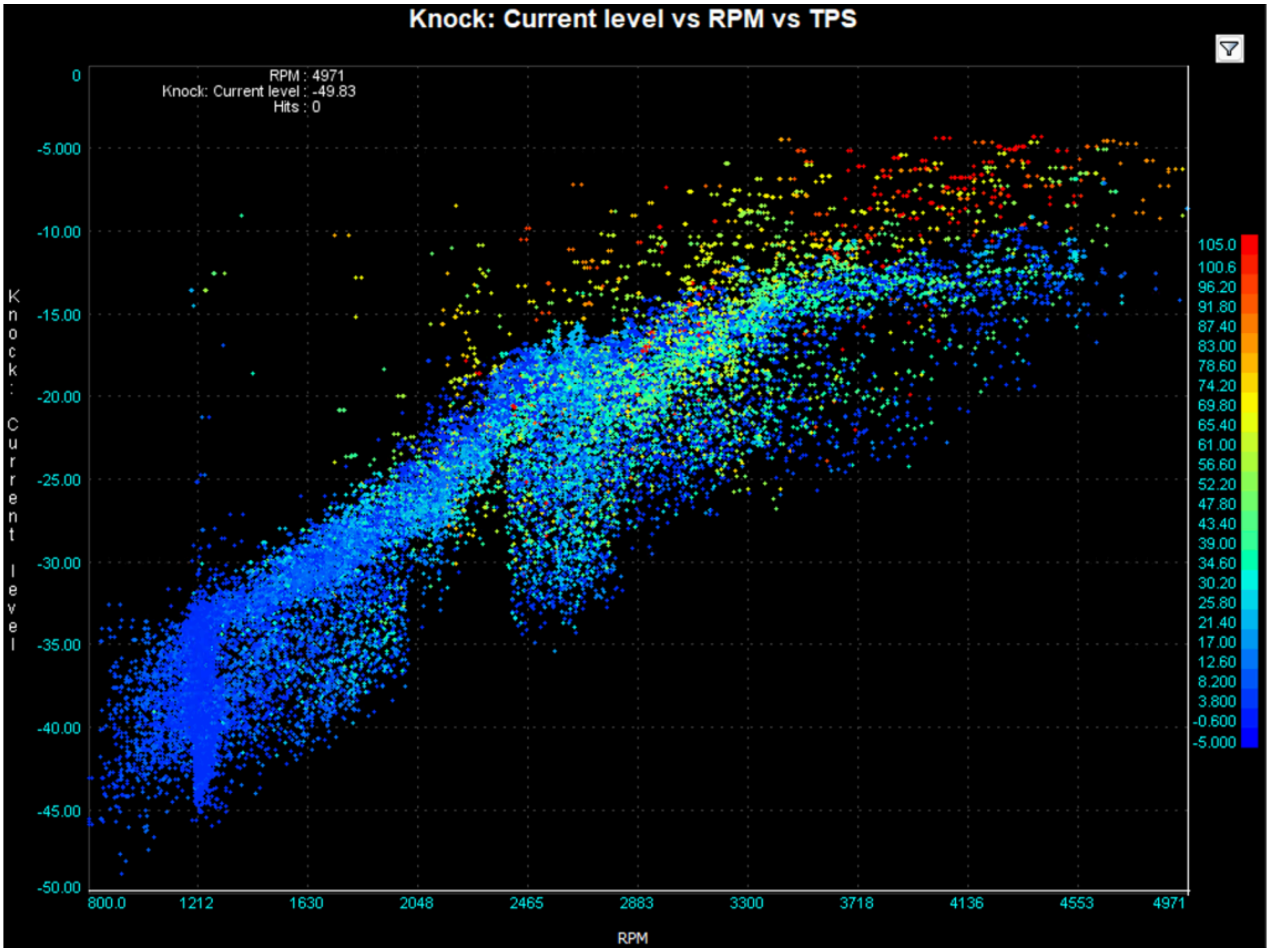
Signal-noise ratio comes into play here. Your knock sensor is going to capture noise because engines are loud. Some of that noise is going to be "normal" things like timing gears, pumps, belt-driven accessories, bearings, the gears of a supercharger, vibrations/harmonics between the engine and the frame of the car with solid mounts, injectors tapping, etc. This is the "noise" in signal-noise ratio for knock purposes. The "signal" is the auditory component of knock from each cylinder. One of the reason why most OEMs have independent gains for each cylinder is that the SIGNAL is not the same amplitude for each cylinder, so gains are used to compensate. Looking at individual cylinder scatter graphs are going to help determine appropriate gains for each cylinder.

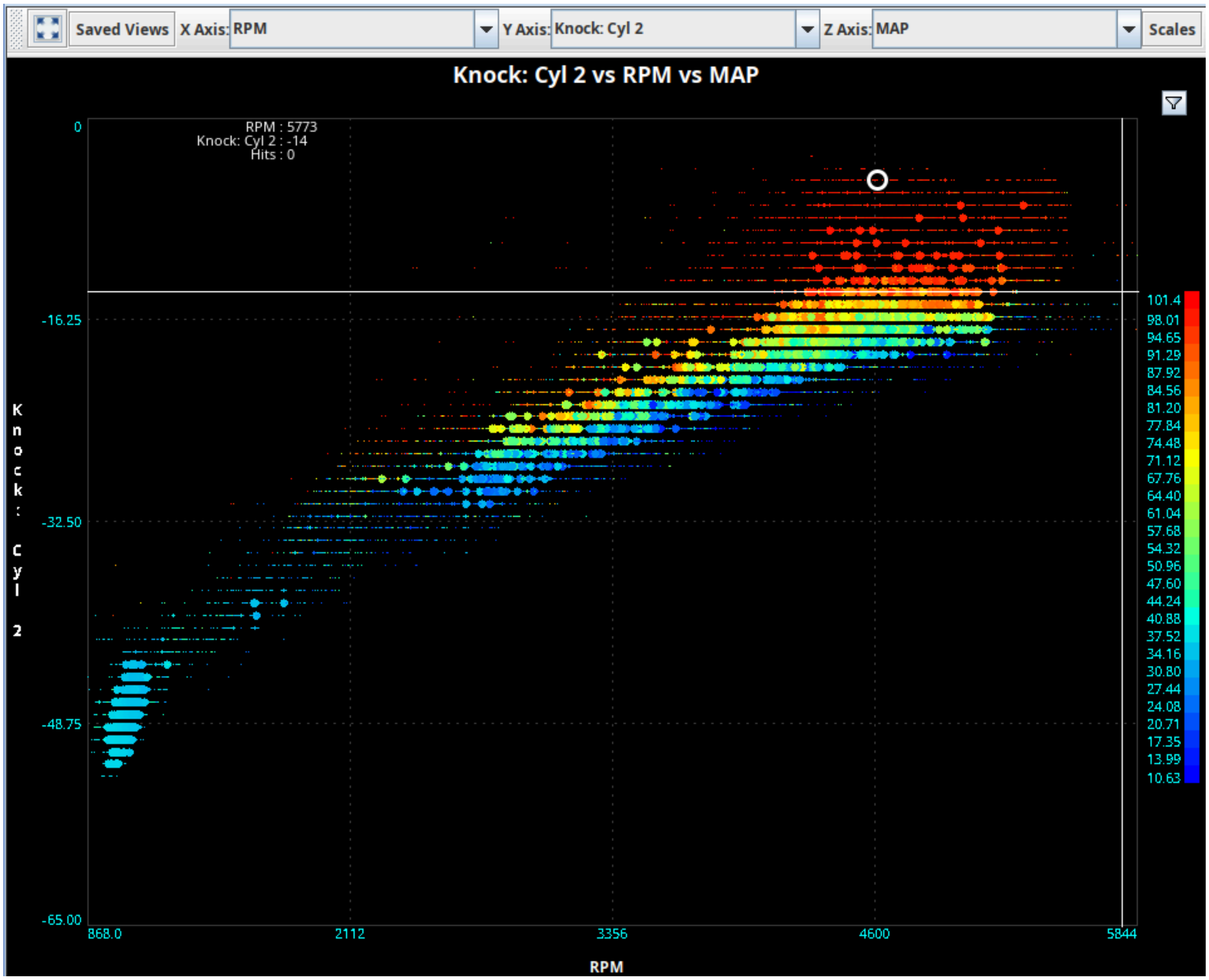
You can get better knock sensing performance by either increasing the "volume" of knock events (generally: move the knock sensor closer to the cylinder, etc.) or decrease the ambient "noise" (fix loud bearings, use softter engine mounts, etc.). If you quiet the engine down by decresing "noise" then knock energy will stand out more. Simple idea, sometimes difficult implementation.

Here's an example of sharp spikes which are knock

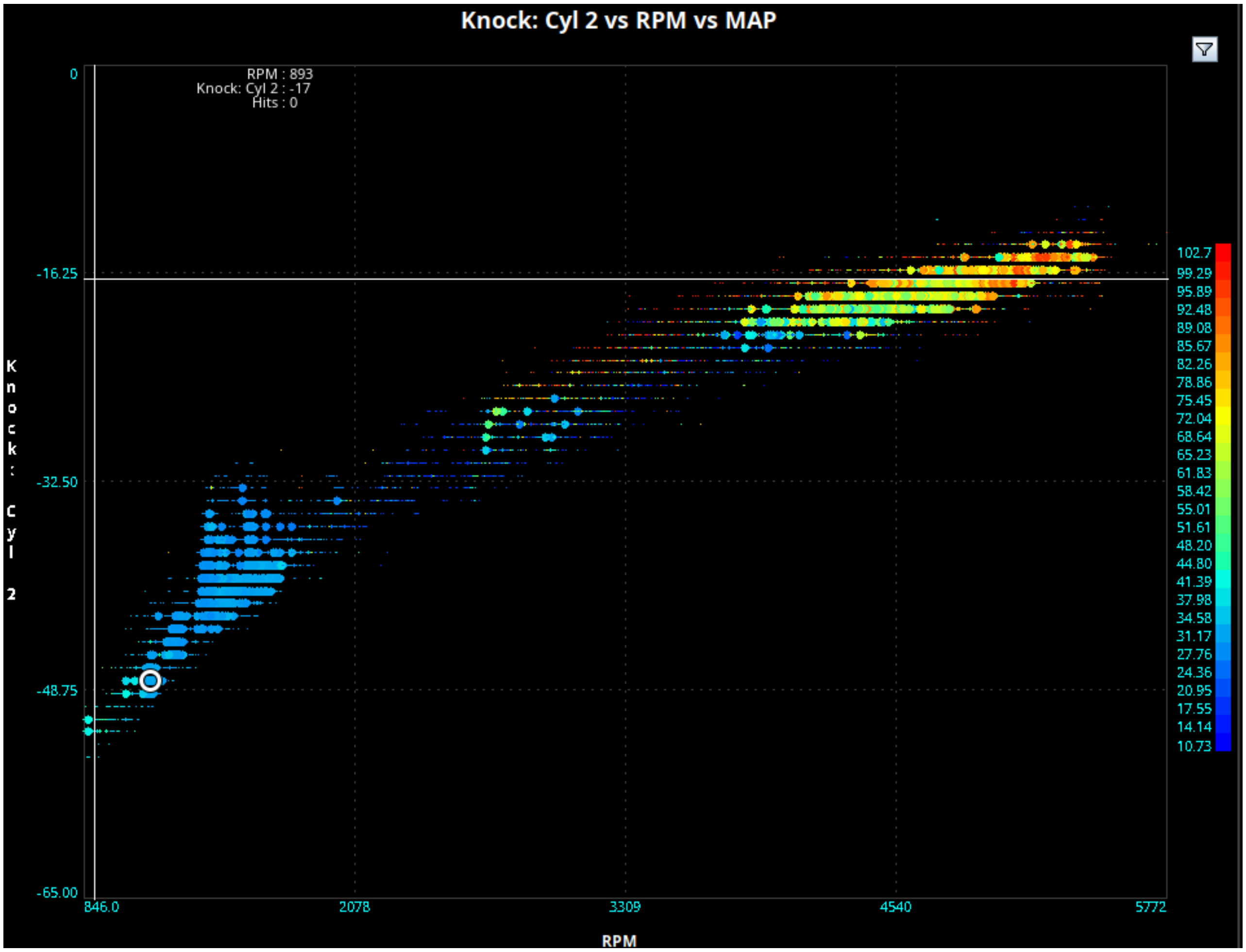


The scatter plot is also useful. Knock generally shows up as outliers outside of the normal noise envelope, especially at high load


Knock: Current level vs RPM vs TPS
RPM : 4971
Knock: Current level : -49.83
Hits : 0
Knock: Current level
0
-5.000
-10.00
-15.00
-20.00
-25.00
-30.00
-35.00
-40.00
-45.00
-50.00
800.0
1212
1630
2048
2465
2883
3300
3718
4136
4553
4971
RPM
105.0
100.6
96.20
91.80
87.40
83.00
78.60
74.20
69.80
65.40
61.00
56.60
52.20
47.80
43.40
39.00
34.60
30.20
25.80
21.40
17.00
12.60
8.200
3.800
-0.600
-5.000



Contrast the image above to a normal noise scatter plot (no knock), which might look like this



The normal noise scatter plot also has a significantly better (higher) signal-noise ratio than the knocking plot above. Independent of the knock events, noise levels occur in a smaller range and the vertical size of the "band" across all RPMs is much smaller. The smaller vertical distance can be interpreted as noise levels being more consistent.

Knock appears as sharp spikes in the time domain. Since knock is a stochastic process, you won't necessarily get the same knock on every cycle. Look for patterns such as:

- Correlation between knock level and throttle position at constant RPM
- Spikes in knock level that correspond to specific load conditions
- Cylinder-specific knock patterns

Reapply Rate Configuration

Setting the reapply rate can be tricky:

- Low load conditions need faster reapply rates
- High load conditions need slower reapply rates

Some users implement a workaround by using zeros in the bottom line of the max knock retard table.

Hardware Connections

The knock sensor is essentially a piezo microphone element:

- For testing/analysis, you can attach a 3.5mm connector to your knock sensor wires and plug into a laptop
- The knock input on ECUs (like Proteus) is very different from a normal analog input
- The sensor connection is balanced, similar to audio equipment principles

Common Issues & Debugging

1. **No Timing Retard**: Ensure your max knock retard table isn't all zeros
2. **False Positives**: If you're getting knock detection without audible knock, recalibrate your threshold
3. **Noise Filtering**: In TunerStudio, set filter level to 2 to clean up MAP signals and reduce noise

4. **Rich Mixtures**: Beyond 0.8 lambda, additional fuel enrichment won't help with knock mitigation

Advanced Analysis

For deeper knock analysis, consider using a spectrogram to visualize the frequency characteristics of engine noise and knock events.

Remember: When setting up knock detection, it's prudent to start with conservative timing and gradually optimize rather than risking engine damage.

Launch Control

Or how to break your gearbox and engine

Lua Scripting

Introduction

FOME allows to extend and customize firmware functionality and behavior by providing a Lua script interpreter. Various sensors, signals, and state are provided for reading and manipulating , allowing to tailor a control strategy to fit the applications needs.

This page documents the most up-to-date version of FOME's Lua scripting support: not all interfaces are supported in earlier versions.

Overview

FOME provides Lua interface with a number of functions and types to interface with the firmware and to discern and manipulate its state and configuration. At a high level, the interface comprises these categories:

- A small utility library, including timers, and user-defined lookups; see the Utilities reference.
- General input and output; see the Input and Output reference.
- Firmware sensors and control; see the Sensors reference.
- CAN bus communication; see the CAN bus reference.
- SENT protocol communication; see the SENT protocol reference.
- Firmware state and configuration; see the Firmware ... TODO reference.

For examples see the files in FOME's `lua/examples/` directory.

For a basic introduction see this wiki section.

Conventions

- The Lua interpreter will trigger an error if there is a mistake in the program: check the FOME Console to see errors and script output.
- Unless otherwise mentioned, all `index` parameters start with the first element at index 0.

Writing Your Script

The entire Lua script is read and validated at startup, then a global script function named `onTick` is invoked periodically by the firmware.

Here is a simple script to illustrate this behavior:

```lua
print('Hello FOME via Lua!')

function onTick()
    print('FOME called onTick()')
end
```

User-Defined Lookup Tables and Curves

FOME provides for user-defined lookup tables and curves for use with Lua scripting. These tables and curves are set in the FOME configuration (via TunerStudio) and lookups are interpolated along their definition.

The tables and curves have user-defineable names up to sixteen characters long. Their names and definitions are configurable in the *Advanced > Lua Calibrations* menu in TunerStudio.

3D Tables

FOME provides four user-definable three-dimensional tables for use with Lua scripting. The first table affords the most precision, defined by single-precision floating-point values, while the remaining three tables are defined by 8-bit integers; all tables are eight by eight in dimension, defined by 16-bit integer coordinates.

TODO: insert TunerStudio Script Table dialog screenshots

Two functions are provided to interact with the user-defined tables:

- `findTableIndex(name)`
- `table3d(index, x, y)`

2D Curves

FOME provides six user-definable two-dimensional curves for use with Lua scripting. The first two curves affords the most accuracy, defined by sixteen single-precision floating-point coordinates, while the remaining four curves are defined by eight single-precision floating-point coordinates.

TODO: insert TunerStudio Script Curve dialog screenshots

Two functions are provided to interact with the user-defined curves:

- `findCurveIndex(name)`
- `curve(index, x)`

User-Defined Settings

FOME provides eight user-definable single-precision floating-point settings for use with Lua scripting.

- `findSetting(name, defaultValue)`

Persistent Values

FOME provides 64 numeric persistent values for use with Lua scripting. Persistent values store data in RAM that is backed up in such a way as to survive ignition/power cycles.

TODO: provide more detail of this new feature

- `getPersistentValue(index)`
- `storePersistentValue(index, value)`

Lua Interface Reference

Utilities

> **INFO**
>
> These functions are included in all builds of FOME unless otherwise noted.

`mcu_standby()`

> **WARNING**
>
> `mcu_standby` is only availble in FOME builds targetting STM32 F4 and STM32 F7 MCUs.

Causes the firmware to place the MCU into a low current consumption standby mode.

no parameters

`print(message)`

Print a line of text to the ECU's log.

parameter	type	description
`message`	string	The message to print. Pass a string (or number) and it will be printed to the log.

`setDebug(index, value)`

Sets the debug channel of the specified index to the given value.

> **NOTE**
>
> `setDebug` only works when FOME's debug mode is set to `Lua`.

parameter	type	description
`index`	integer	The index of the debug channel to set; 1 through 6.
`value`	float	The value to set the specified debug channel to.

`setTickRate(frequency)`

Set the frequency at which the firmware passes context to the Lua script. Primarily, this controls how often FOME calls the script's `onTick` function. Additionally, this affects how often other functions and callbacks of the script, like `onCanRx`, are invoked.

The default rate set at startup is 10 times per second (10 Hz).

parameter	type	description
`frequency`	float	The tick rate to set, in hertz. Values are clamped to be not less than 1 hertz and not more than 200 hertz.

`crc8_j1850(data, length)`

TODO: computes the OBD-II (SAE J1850) CRC-8 cyclic redundancy check on up to eight bytes of data

parameter	type	description
`data`	integer table	
`length`	integer	

`interpolate(x1, y1, x2, y2, x)`

Linearly interpolate a value `x` along the line defined by two points `(x1, y1)` and `(x2, y2)`.

parameter	type	description
`x1`	float	The x-axis value of the first point of the line.
`y1`	float	The y-axis value of the first point of the line.
`x2`	float	The x-axis value of the second point of the line.
`y2`	float	The y-axis value of the second point of the line.
`x`	float	The x-axis value of the point to interpolate along the line defined by `(x1, y1)` and `(x2, y2)`.

`findTableIndex(name)`

Determine the user-defined Lua script table index identified by its name.

parameter	type	description
`name`	string	The name of the user-defined table to determine the index of.

`table3d(index, x, y)`

Lookup a linearly interpolated value from the specified user-defined Lua script table. Tables are identified by their 1-based index: 1, 2, 3, 4.

parameter	type	description
`index`	integer	The index of the user-defined table to lookup into; 1 through 4.
`x`	float	The x-axis value to lookup in the specified table.
`y`	float	The y-axis value to lookup in the specified table.

`findCurveIndex(name)`

Determine the user-defined Lua script curve index identified by its name.

parameter	type	description
`name`	string	The name of the user-defined curve to determine the index of.

`curve(index, x)`

Lookup a linearly interpolated value from the specified user-defined Lua script curve. Curves are identified by their 1-based index: 1, 2, 3, 4, 5, 6.

parameter	type	description
`index`	integer	The index of the user-defined curve to lookup into; 1 through 6.
`x`	float	The x-axis value to lookup in the specified curve.

`findSetting(name, defaultValue)`

Retrieve the value of the user-defined Lua setting identified by its name, or the supplied value if the setting does not exist.

This is most useful when the script developer and consumer are different people, and also when editing the Lua script in TunerStudio.

parameter	type	description
`name`	string	The name of the user-defined setting to retrieve the value of.
`defaultValue`	float	The value to use if specified user-defined

parameter	type	description
		setting does not exist.

`getPersistentValue(index)`

Returns the persisted value currently stored for the given index. Persistent values are indentified by their 1-based index: 1, 2, 3, ..., 64.

parameter	type	description
`index`	integer	The index of the persistent value to retrieve; 1 through 64.

`storePersistentValue(index, value)`

Stores the given value to the persistent index specified. Persistent values are indentified by their 1-based index: 1, 2, 3, ..., 64.

parameter	type	description
`index`	integer	The index of the persistent value to store; 1 through 64.
`value`	number	The persistent value to set the specified index to.

`Timer`

`Timer` is Lua type that keeps track of elapsed seconds. The timer does not initialize to a reset state; instead it initializes to a "foreverish" value.

```
timer = Timer.new()
```

`Timer.new()`

no parameters

`Timer.getElapsedSeconds()`

Returns the number of seconds since the timer was last reset.

no parameters

`Timer.reset()`

Resets the timer to begin counting from zero.

no parameters

`Pid`

`Pid` is a Lua type that implements a PID (proportional-integral-derivative) controller.

$$u(t) = K_\mathrm{p} e(t) + K_\mathrm{i} \int_0^t e(\tau)\,\mathrm{d}\tau + K_\mathrm{d} \frac{\mathrm{d}e(t)}{\mathrm{d}t}$$

The implementation automatically tracks the time delta between `Pid.get` invocations.

```
pid = Pid.new(1.5, 0.2, 0.1, -80, 80)
```

`Pid.new(kp, ki, kd, min, max)`

parameter	type	description
`kp`	float	K_p, the proportional factor of the PID control.
`ki`	float	K_i, the integral factor of the PID control.
`kd`	float	K_d, the derivative factor of the PID control.
`min`	float	The minimum output value of the PID control; output is limited above this value.
`max`	float	The maximum output value of the PID control; output is limited below this value.

`Pid.get(target, input)`

Retrieves the PID controller's output given the target setpoint and the output of the process or system the PID is controlling.

parameter	type	description
`target`	float	The target setpoint of the PID controller.
`input`	float	The output of the process or system that is feedback to the PID controller.

`Pid.setOffset(offset)`

Sets the amount to statically bias the PID controller output by.

parameter	type	description
`offset`	float	The amount to add to the computed PID controller output.

`Pid.reset()`

Resets the PID controller.

no parameters

Input and Output

`getAuxAnalog(index)`

parameter	type	description
`index`	integer	

`getAuxDigital(index)`

parameter	type	description
`index`	integer	

`getDigital(index)`

parameter	type	description
`index`	integer	

`readPin(name)`

Returns the physical value of an MCU pin by its name.

parameter	type	description
`name`	string	The name of the MCU pin to return the value of; e.g. "PD15".

`startPwm(index, frequency, duty)`

parameter	type	description
`index`	integer	The index of the PWM output to control; 0 through 7.
`frequency`	float	The frequency to set the PWM output to. Values are clamped to be not less than 1 hertz and not more than 1000 hertz.
`duty`	float	The duty cycle (on time) to set the PWM output to. Values are clamped to be not less than 0.0 and not more than 1.0.

`setPwmDuty(index, duty)`

parameter	type	description
`index`	integer	The index of the PWM output to control; 0 through 7.

parameter	type	description
`duty`	float	The duty cycle (on time) to set the PWM output to. Values are clamped to be not less than 0.0 and not more than 1.0.

`setPwmFreq(index, frequency)`

parameter	type	description
`index`	integer	The index of the PWM output to control; 0 through 7.
`frequency`	float	The frequency to set the PWM output to. Values are clamped to be not less than 1 hertz and not more than 1000 hertz.

`setLuaGauge(index, value)`

Sets the given Lua gauge to the provided value. Currently two Lua guages are supported: indices 1 and 2. This can also be accomplished by using the Lua `Sensor` interface, but `setLuaGauge` is more convenient to use.

parameter	type	description
`index`	integer	The index of the Lua gauge to set.

parameter	type	description
`value`	number	The value to set the Lua gauge to.

Sensors

`hasSensor(index)`

Checks whether a particular sensor is configured (whether it is currently valid or not).

parameter	type	description
`index`	integer	The index of the sensor to check; see `sensor_type.h`.

`getSensor(name)`

Returns the value of a sensor by its name.

parameter	type	description
`name`	string	The name of the sensor to get the value of; see `sensor_type.h`.

`getSensorRaw(index)`

Returns the raw value of a sensor by its name. For most sensors this means the analog voltage on the relevant input pin.

> **NOTE**
>
> Returns 0 if the sensor doesn't support raw readings, isn't configured/valid, or has failed.

parameter	type	description
`name`	string	The name of the sensor to get the value of; see `sensor_type.h`.

`getSensorByIndex(index)`

Returns the value of a sensor by its index.

parameter	type	description
`index`	integer	The index of the sensor to get the value of; see `sensor_type.h`.

`Sensor`

`Sensor` is a Lua type that allows to control the value of sensors. The type is implemented as a "stored-value" sensor, that operates asynchronously and whose value is invalidated periodically upon a given timeout (initially: 100 milliseconds).

```
sensor = Sensor.new("OilPressure")
```

`Sensor.new(name)`

parameter	type	description
`name`	string	The name of the sensor to control; see `sensor_type.h`.

`Sensor.set(value)`

Sets the controlled sensor's value as provided.

parameter	type	description
`value`	float	The value to set the controlled sensor to.

`Sensor.setRedundant(isRedundant)`

Sets the redundancy-aspect of the controlled sensor.

parameter	type	description
`isRedundant`	boolean	Whether or not the controlled sensor is redundant.

`Sensor.setTimeout(timeoutMs)`

Sets the timeout for the controlled sensor's stored-value.

parameter	type	description
`timeoutMs`	integer	How long the controlled sensors value is valid for, in milliseconds.

`Sensor.invalidate()`

Invalidates the controlled sensor's stored-value.

no parameters

Firmware ... TODO

TODO

`getOutput(name)`

Returns the value of an "output" from FOME: allows to inspect "internal" firmware state.

TODO: reference list of valid outputs

parameter	type	description
`name`	string	The name of a FOME output/state to return the value of.

`setClutchUpState(isUp)`

Use `setClutchUpState` to tell FOME about CAN-based brake pedal.

parameter	type	description
`isUp`	boolean	Whether the clutch is up or not.

`setBrakePedalState(isUp)`

Use `setBrakePedalState` to tell FOME about CAN-based brake pedal.

parameter	type	description
`isUp`	boolean	Whether the brake pedal is up or not.

`setAcRequestState(isRequested)`

Use `setAcRequestState` to tell FOME about CAN-based A/C request.

parameter	type	description
`isRequested`	boolean	Whether the A/C is requested on or not.

`restartEtb()`

TODO

no parameters

`setEtbDisabled(isDisabled)`

TODO

parameter	type	description
`isDisabled`	boolean	Whether the ETB is disabled or not.

`setIgnDisabled(isDisabled)`

TODO: `setIgnDisabled` function for all kinds of cranking safety systems

parameter	type	description
`isDisabled`	boolean	Whether the ignition is disabled or not.

`setAcDisabled(isDisabled)`

TODO: Disable/suppress A/C functionality regardless of what and how enables it, an override kind of deal.

parameter	type	description
`isDisabled`	boolean	Whether the A/C is disabled or not.

`getTimeSinceAcToggleMs()`

TODO

no parameters

`getCalibration(name)`

TODO: Gets current calibration value for specified scalar setting `name`. For example `getCalibration("cranking.rpm")`

For complete list of possible calibration names (valid parameter values) and descriptions see `value_lookup_generated.md`.

parameter	type	description
`name`	string	TODO

`setCalibration(name, value, needEvent)`

TODO: Sets specified calibration setting to specified value. Fires calibration change event depending on needEvent parameter.

For example `setCalibration("cranking.rpm", 900, false)`

parameter	type	description
`name`	string	TODO
`value`	number	TODO
`needEvent`	boolean	TODO

`setTimingAdd(angle)`

TODO: Use negative values to retard timing.

parameter	type	description
`angle`	float	TODO

`setTimingMult(coefficient)`

TODO

parameter	type	description
`coefficient`	float	TODO

`setFuelAdd(amount)`

TODO: Amount of fuel mass to add to injection, scaled by fuel multiplier (`setFuelMult(coefficient)`); initially 0.

parameter	type	description
`amount`	float	TODO

`setFuelMult(coefficient)`

TODO: Amount to scale added fuel mass by; initially 1.0;

parameter	type	description
`coefficient`	float	TODO

`setEtbAdd(percent)`

TODO: Amount of ETB to add, as a percent of the wide-open value: e.g. `10` for +10%. The value is a static amount to add to the determined value, e.g. TPS of 5% w/ `10` results in 15% ETB. #torque

parameter	type	description
`percent`	float	TODO

`getGlobalConfigurationVersion()`

TODO

no parameters

`getFan()`

TODO

no parameters

`getAirmass()`

TODO

no parameters

`setAirmass(airmass, load)`

TODO

parameters	type	description
`airmass`	float	TODO
`load`	float	TODO: percent

`resetOdometer()`

TODO

no parameters

`stopEngine()`

TODO

no parameters

`vin(index)`

Lookup a character of the set vehicle identification number (VIN) at the given index.

parameter	type	description
`index`	number	The character index of the set VIN to return.

`setIdleAdd`

TODO

parameters	type	description
`amount`	float	TODO

`setIdleAddRpm`

TODO

parameters	type	description
`amount`	float	TODO

CAN Bus

> **INFO**
>
> These functions are included in builds of FOME that incorporate CAN support.

All FOME boards support at least one CAN bus, which has index 1. Some recent boards support multiple CAN buses, with index 2 and higher.

The `canRxAdd` and `canRxAddMask` functions are available in various forms with differing number and types of parameters. They are conceptually the same, with `canRxAddMask` providing an extra argument to filter specific frames by a bit-mask. Currently, FOME allows for up to 48 different CAN frame reception filters, and will issue an error when attempting to add more filters than the limit (*`OBD_PCM_Processor_Fault`: Too many Lua CAN RX filters*).

When not using the forms of `canRxAdd` and `canRxAddMask` with a `callback` argument, FOME will invoke the global `onCanRx` function defined in the Lua script. Otherwise, the function referenced in the `callback` argument will be invoked.

The `callback` argument and `onCanRx` function is expected to be a Lua function with the following parameters:

parameter	type	description
`bus`	integer	The CAN bus index the frame was received on.
`id`	integer	The CAN ID of the received frame.
`dlc`	integer	The received CAN frame's data length.
`data`	integer table	The received CAN frame's data; an integer table from index 0 through (`dlc` - 1).

A script using the `canRxAdd`/`canRxAddMask` functions might look as follows:

```
function onCanRx(bus, id, dlc, data)
    -- Do things with received CAN frame data!
end

function handleSpecialCanRx(bus, id, dlc, data)
    -- Do things with received CAN frame data!
end

-- data handled in global onCanRx function
canRxAdd(1, 0x55)

-- data handled in special callback function
canRxAddMask(2, 0x40, 0x94, handleSpecialCanRx)
```

`canRxAdd(bus, id, callback)`

Adds a CAN frame receiption filter, filtering by CAN bus and CAN ID, which invokes the supplied function when a CAN frame passes the filter.

parameter	type	description
`bus`	integer	The CAN bus index to add the reception frame filter for.
`id`	integer	The CAN ID to add the reception frame filter for.
`callback`	function	The function to invoke when a received CAN frame passes the added filter.

`canRxAdd(bus, id)`

Adds a CAN frame reception filter, filtering by CAN bus and CAN ID.

parameter	type	description
`bus`	integer	The CAN bus index to add the reception frame filter for.
`id`	integer	The CAN ID to add the reception frame filter for.

`canRxAdd(id, callback)`

Adds a CAN frame reception filter, filtering by CAN ID, on all available CAN buses, which invokes the supplied function when a CAN frame passes the filter.

parameter	type	description
`id`	integer	The CAN ID to add the reception frame filter for.
`callback`	function	The function to invoke when a received CAN frame passes the added filter.

`canRxAdd(id)`

Adds a CAN frame reception filter, filtering by CAN ID on all CAN buses.

parameter	type	description
`id`	integer	The CAN ID to add the reception frame filter for.

`canRxAddMask(bus, id, mask, callback)`

Adds a CAN frame reception filter, filtering by CAN bus and CAN ID, which invokes the supplied function when a CAN frame passes the filter.

parameter	type	description
`bus`	integer	The CAN bus to add the reception frame filter for.
`id`	integer	The CAN ID to add the reception frame filter for.
`callback`	function	The function to invoke when a received CAN frame passes the added filter.

`canRxAddMask(bus, id, mask)`

Adds a CAN frame reception filter, filtering by CAN bus and CAN ID.

parameter	type	description
`bus`	integer	The CAN bus index to add the reception frame filter for.
`id`	integer	The CAN ID to add the reception frame filter for.

`canRxAddMask(id, mask, callback)`

Adds a CAN frame reception filter, filtering by CAN ID, on all available CAN buses, which invokes the supplied function when a CAN frame passes the filter.

parameter	type	description
`id`	integer	The CAN ID to add the reception frame filter for.
`callback`	function	The function to invoke when a received CAN frame passes the added filter.

`canRxAddMask(id, mask)`

Adds a CAN frame reception filter, filtering by CAN ID on all CAN buses.

parameter	type	description
`id`	integer	The CAN ID to add the reception frame filter for.

`enableCanTx(isEnabled)`

> **INFO**
>
> `enableCanTx` is available in all builds of FOME, regardless of incorporated CAN support.

Used to enable or disable CAN transmission. CAN transmission is enabled by default at startup.

parameter	type	description
`isEnabled`	boolean	Whether or not CAN transmission is enabled.

`txCan(bus, id, isExtended, data)`

Transmits a CAN frame on the specified CAN bus, with the supplied CAN ID and data.

parameter	type	description
`bus`	integer	The CAN bus index to transmit the frame on.
`id`	integer	The CAN ID to transmit with the frame.
`isExtended`	integer	Whether to transmit a standard (11-bit ID) or extended (29-bit ID) CAN frame.
`data`	integer table	The data to transmit with the CAN frame.

SENT Protocol (SAE J2716)

> **INFO**
>
> These functions are included in builds of FOME that incorporate SAE J2716 SENT support.

> ⚠ **WARNING**
>
> These functions are still in development and not fully documented or supported. Use is discouraged.

`getSentValue(index)`

Retrieves the value of the last valid message of the specified SENT channel.

parameter	type	description
`index`	integer	The SENT channel to retrieve the value of; 0 through 3.

`getSentValues(index)`

Retrieves the values of the last valid message of the specified SENT channel.

parameter	type	description
`index`	integer	The SENT channel to retrieve the values of; 0 through 3.

Lua Functions/Hooks

Launch Control

`setSparkSkipRatio`

Crankshaft Position Input

`selfStimulateRPM`

`getEngineState`

`getTimeSinceTriggerEventMs`

Boost Control

`setBoostTargetAdd`

`setBoostTargetMult`

`setBoostDutyAdd`

Vehicle Speed

`getCurrentGear`

`getRpmInGear`

MAP Sampling Angle

Multi Dimension Mapping

Basic outline

Multiplier and Bias tables

VE multiplier

Ignition Adder

What can be done with them

Flex-Fuel or Dual Fuel

ITB TPS blending

Something Wacky

Override VE. Ignition and AFR table axis

Rotary engine specific settings

Smart Alternator Control

Multispark

Fuel

Fuel

🗃️ Fuel algorithms

4 items

🗃️ Fuel settings

10 items

🗃️ Acceleration

2 items

Fuel algorithms

Fuel algorithms

📄 AlphaN Strategy

📄 Lua Strategy

Use Lua to control the engine fueling strategy/control algorithm.

📄 MAF Strategy

Current Status

📄 Speed Density Strategy

Engine Load is a function of MAP, VE and target AFR

AlphaN Strategy

Lua Strategy

Use Lua to control the engine fueling strategy/control algorithm.

- Lua: `getAirmass`
- Lua: `setAirmass(load)`

MAF Strategy

Current Status

The MAF based fueling of FOME is still undergoing development, the current status has the fueling functioning correctly but presents a few tuning challenges due to TunerStudio integration and the Spark Table still being reliant on the old engine load math.

This is an evolving situation at present and thus MAF fueling is still considered experimental.
Please only use for development work at your own risk.

MAF fuel theory - The detail

The implementation of the MAF in FOME is intended to replicate the functionality of OEM systems and as such is more complex than some other systems.

The foundation of the MAF system is the Mass Air Flow sensor itself, this is a device using a hot wire, hot film or vane to directly measure the flow of air into the engine. Obviously this sensor does not give out an airflow value, it gives us a voltage, current or PWM signal that represents the flow. FOME can interpret a voltage or current MAF at this time via a transfer function table.

The Transfer function table is used to convert the raw MAF sensor reading into a Kg/h (Kilogram Per Hour) airflow.

This Kg/h value is then processed into a required fuel quantity by the following calculations:

g/s = Kg/h * 1000 / 3600

n/s = rpm / 60

airPerRevolution = g/s / n/s

cylinder Airmass = airPerRevolution / half Cylinder number

In FOME we use a correction factor table to modify this measured air mass to allow correction of any errors in the measurement due to dynamic air flow effects. To do this we need to have a "load" value that allows us to have a Load Vs Speed fuel table.

StandardAirCharge = engine displacement / number of cylinders * 1.2929

This produces the air mass for cylinder filling at 100% VE under standard SAE conditions. Using this we can relate our cylinder air mass back to a standardised 100% cylinder filling and thus we have a "load" value to use when tuning.

airChargeLoad = 100 * cylinder Airmass / Standard AirCharge

The required fuel is now simply corrected by adjusting the measured air by the value in the VE table, this value is simply a %.

corrected Cylinder Airmass = cylinder Airmass * (VE map value / 100) fuelMassGram = corrected Cylinder Airmass / desired AFR

pulse Width = fuelMass / injector flow (in g/s)

Using this method FOME is able to directly measure the air flow into an engine and calculate the required fuel with minimal tuning.

MAF fuel tuning - The quick version

To tune FOME using the MAF is probably the quickest and easiest method provided you have a working MAF sensor and the correct information to input in the Transfer Function Table.

The first thing to do is input the voltage (or current) to Kg/h information into the Transfer Function Table in TunerStudio.

Secondly you will need to decide what Air/Fuel ratio you would like your engine to run at and input this into the AFR table in TunerStudio. For a first start a value of 14 is perfectly acceptable for gasoline. This table is the primary source of the desired fuel mixture, it will be this table that is tuned to decide the engines target AFR. A future update will make this fueling table dynamic so that an input % of ethanol in the main fueling dialog will change the fuel density and thus the required fuel mass injected. The result of this will be that users can leave this table tuned as though it were for pure gasoline (14.7:1 stoichiometric) and the % ethanol input will make sure the fuelling stays at the same Lambda value. This has the advantage of working well with aftermarket wideband controllers that generally work in AFR using pure gasoline as the standard.

Before starting the engine for the first time it is wise to ensure the Fuel Table is filled with values of "100", a value of 100 means that the fuel calculation uses 100% of its measured air mass to decide on the fuel injection pulse.
Tuning this table will adjust for dynamic airflow effects that happen in the inlet of an engine and will allow small (or large but hopefully not) corrections to the fuel injection which may be required to have the engine meet it's desired air/fuel target.
This table should only be tuned if the engine is not meeting the desired air/fuel target under relatively steady state conditions (i.e. without any acceleration enrichment or overrun fuel cut). If a different air/fuel ratio is desired at a specific load or RPM then the AFR Table is the correct table to adjust instead.

Some useful MAF sensor maths in this link

Speed Density Strategy

> Engine Load is a function of MAP, VE and target AFR
> http://articles.sae.org/8539/

Fuel settings

Fuel settings

📄 **Air Fuel Ratio Setting**

📄 **Coolant Temperature Multiplier**

📄 **Individual Cylinder Trim**

📄 **Deceleration fuel cutoff**

📄 **How flex fuel works in FOME**

Place to cover the flex fuel, how it works and how it is related to the multidimensional mapping.

📄 Fuel injection Mode

Place to cover the simultaneous, batch and sequential fuel injection

📄 TPS fuel multiplier

📄 Intake Air Temperature Fuel Multiplier

📄 Fuel injector dead time settings

📄 Small Pulse Width correction

Polynomial

Air Fuel Ratio Setting

Coolant Temperature Multiplier

Individual Cylinder Trim

Deceleration fuel cutoff

How flex fuel works in FOME

Place to cover the flex fuel, how it works and how it is related to the multidimensional mapping.

Fuel injection Mode

Place to cover the simultaneous, batch and sequential fuel injection

TPS fuel multiplier

Intake Air Temperature Fuel Multiplier

Fuel injector dead time settings

Small Pulse Width correction

Polynomial

2 slope/Ford method

Acceleration

Acceleration

📄 TPS Based Acceleration Correction

There are three menus for TPS-Based Acceleration Correction, including Accel/Decel Enrichment, TPS/TPS Acceleration Extra Fuel and TPS/TPS AE RP...

📄 Wall Wetting Based Acceleration Compensation

Wall Wetting (alpha version)

TPS Based Acceleration Correction

There are three menus for TPS-Based Acceleration Correction, including *Accel/Decel Enrichment*, *TPS/TPS Acceleration Extra Fuel* and *TPS/TPS AE RPM Correction*.

Accel/Decel Enrichment

TPS

- **Length**: How long to look back for events that will trigger TPS-based acceleration enrichment. Increasing this time will trigger enrichment for longer when a throttle position change occurs as the strategy can "look back" over a longer period of time. The delta is determined by comparing the minimum TPS to the maximum TPS value across all events in this time period. See variable *Fuel: TPS AE change* for the delta value.
- **Accel Threshold**: Minimum delta of TPS over the time period specified by *Length* to activate acceleration enrichment. Actual TPS change has to be above this value in order for TPS/TPS acceleration to kick in. The actual added injection pulsewidth for accel enrichment is handled by the *TPS/TPS Acceleration Extra Fuel* table.

- **Decel Threshold**: Maximum change delta of TPS percentage over the time period specified by *Length*. Currently not used as the Decel Fuel Enleanment Coefficient (*tpsDecelEnleanmentMultiplier*) is not exposed in TunerStudio.

Accelerator Pump Model

- **Fraction Period**:
- **Fraction Divisor**:

Wall Wetting Based Acceleration Compensation

Wall Wetting (alpha version)

- **Wall fueling model type**: *Basic (constants)* use the *Evaporation Time Constant* and *Added to Wall Coefficient* values for Beta and Tau respectively, *Advanced (tables)* uses the *Wall Wetting AE Evaporation Time* and *Wall Weeting AE Impact Fraction* tables to calculate the Beta and Tau values respectively.

Basic

- **Evaporation Time Constant / Tau**: Length of time in seconds the deposited wall fuel takes to dissipate after the start of acceleration. *wwaeTau*
- **Added to Wall Coeff / beta**: Fractional representation of fuel settling on the intake/port walls. 0 = No fuel settling on port walls 1 = All the fuel settling on port walls. Setting this to 0 disables the wall wetting enrichment. *wwaeBeta*

Advanced

Wall wetting AE evaporation time

Used to calculate the Tau value (*wwaeTau*) of wall-wetting function.

- **Evap time vs. CLT**: Sets the base evaporation time based on CLT. Warmer engines will have a lower evaporation time.
- **Evap time vs. MAP**: Sets the multiplier of the base evaporation time based on MAP. Lower MAP values will have a lower multiplier (shorter evaporation time), higher MAP values will have a higher multiplier (longer evaporation time).

Wall wetting AE impact fraction

Used to calculate the Beta value (*wwaeBeta*) of wall-wetting function.

- **Impact fraction vs. CLT**: Sets the base impact fraction based on CLT. Colder engines will have a higher impact fraction (more fuel sticks to the walls), warmer engines will have a lower fraction (more fuel sucked into the intake). Values range from 0 to 1.
- **Impact fraction vs. MAP**: Sets the multiplier of the base impact fraction based on MAP. Lower MAP values will have a lower multiplier (more fuel sucked into the intake, higher MAP values will have a higher multiplier (more fuel sticks to the walls).

Hardware

Hardware

📄 Flashing Software Notes

Some boards, notably those with processors utilizing certain dual bank memory layout, like the STM32F7, suffer from an

🗃️ Hardware-Circuits

8 items

🗃️ Hardware-Details

6 items

📄 Simulator

In addition to the numerous specific ECU hardware FOME targets, also made available is a build called the FOME

Flashing Software Notes

> ⓘ **NOTE**
>
> Some boards, notably those with processors utilizing certain dual bank memory layout, like the STM32F7, suffer from an issue such that flashing does not succeed correctly, due to an error with the erasure mechanism. For these situations, manually erase the flash memory before flashing; i.e. perform a full/mass erase procedure first.

While the FOME console is the recommended and officially supported way to update FOME firmware on supported boards, other mechanisms exist to flash firmware to board supporting FOME. These tools are also useful in odd situations, like the event of a firmware update/flash failure, or some other firmware corruption issue, such that FOME console nor TunerStudio recognize the ECU when connected.

For the ECU, you will need to access buttons or pins on the main board to force the processor into bootloader/DFU mode. Some boards use a momentary button, likely near to the reset button, to force this mode during power-up. Others might use a set of pins or pads that need shorted to force this mode during power-up. Press the button or short the pins with the ECU disconnected from the computer, then simultaneously connect the ECU into the computer. Once connected, the button can be released or the short removed.

STM32CubeProgrammer

STM32CubeProgrammer is an officially STMicroelectronics supported tool to flash STM32 processors. Information about and how to download and install the tool can be found on the STM wiki.

In STM32CubeProgrammer, select the *USB* tab on the right and refresh the port in the right side menu until a DFU is detected. Click *Connect* to connect the ECU to the programmer. Click the *Read* button to read the device memory and to validate that the ECU is connected.

Once validated, choose the *Open File* button and load the file marked *fome.bin* within the firmware bundle downloaded earlier. This is a binary file containing the ECU firmware. Now choose the *Download* button to write this firmware to the ECU. Give it up to several minutes to download and once the status console confirms successful download, disconnect from the programmer then unplug the ECU and plug it back in. The ECU should now connect to the computer and TS normally again.

dfu-util

dfu-util is a free software tool to flash Device Firmware Upgrade (DFU) compatible devices, including STM32 processors. It supports most popular operating systems, including MacOS and Linux. Information about dfu-util can be found on the dfu-util website.

Updating with dfu-util is as straight-forward as invoking it like so:

```
dfu-util -v -d 0483:df11 -a 0 -D fome.dfu -s :leave
```

Note that the `:leave` option does not always work, which would otherwise automatically instruct the processor to leave DFU mode and load the firmware; the board must be manually reset to load the flashed firmware.

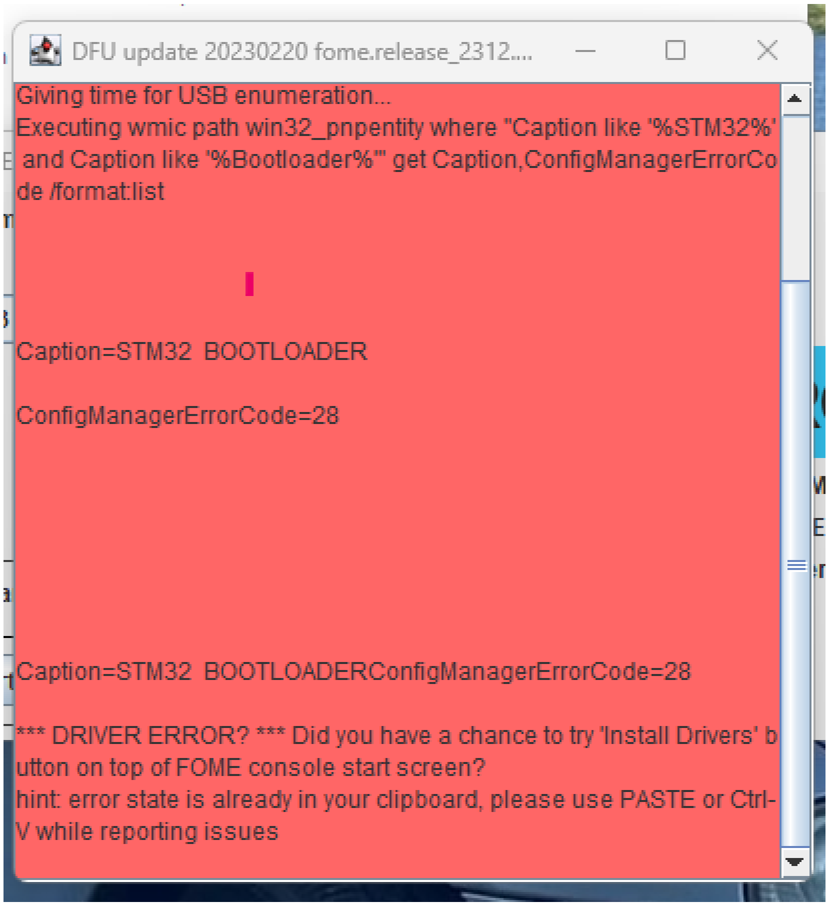
To erase a device, use the following invocation:

```
dfu-util -v -d 0483:df11 -a 0 -s :mass-erase:force
```

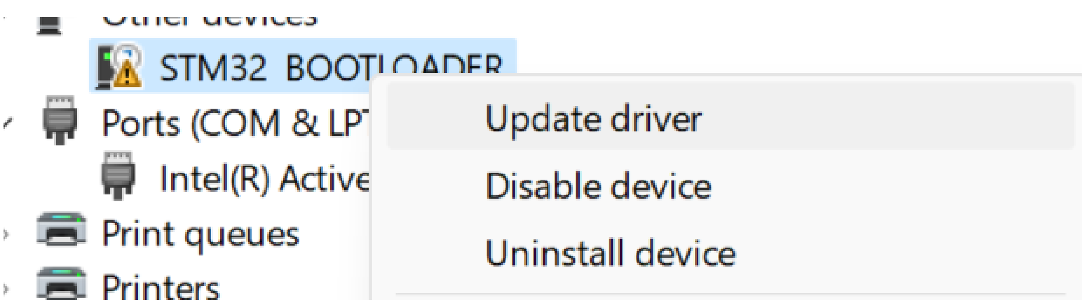
Troubleshooting updates prior to release FOME 2312

If an error occurs during the DFU update for release 2312 a manual driver update may be required. This may be the case if the ECU is not yet using the OpenBLT bootloader implementation.

When FOME console prompts with a "ConfigManagerErrorCode=28", proceed by opening Device Manager. Look for the device named "STM32 BOOTLOADER"; it may be under "Other Devices".



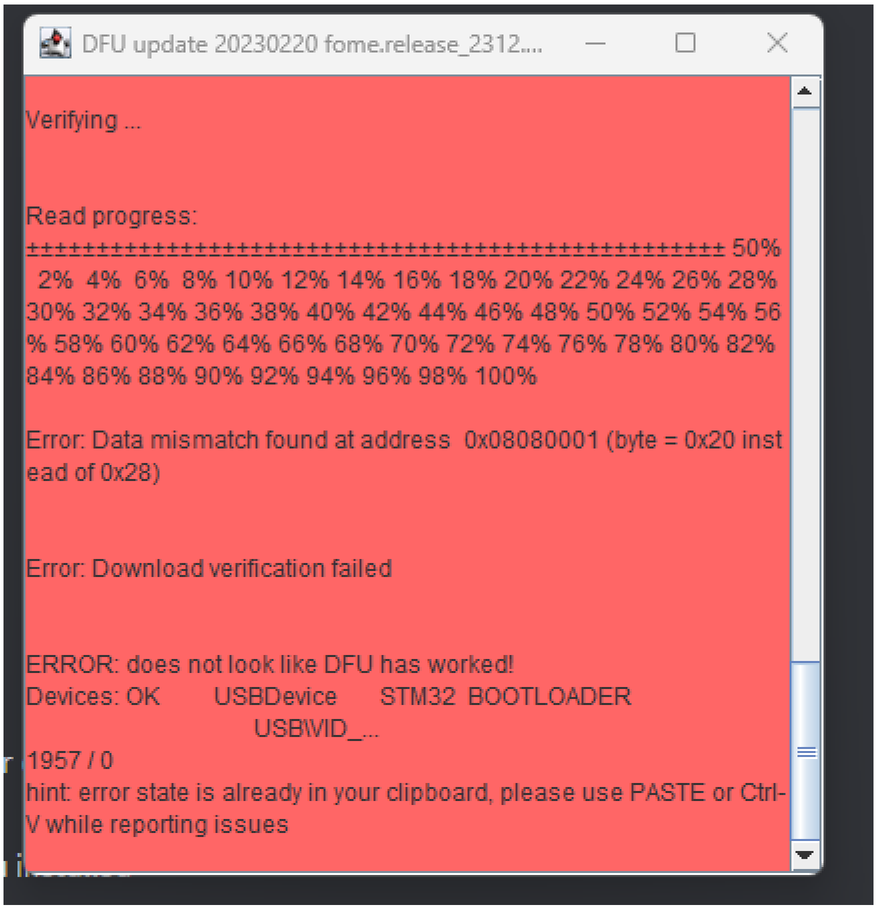
Right click the STM32 BOOTLOADER device and select "Update driver" from the context menu.



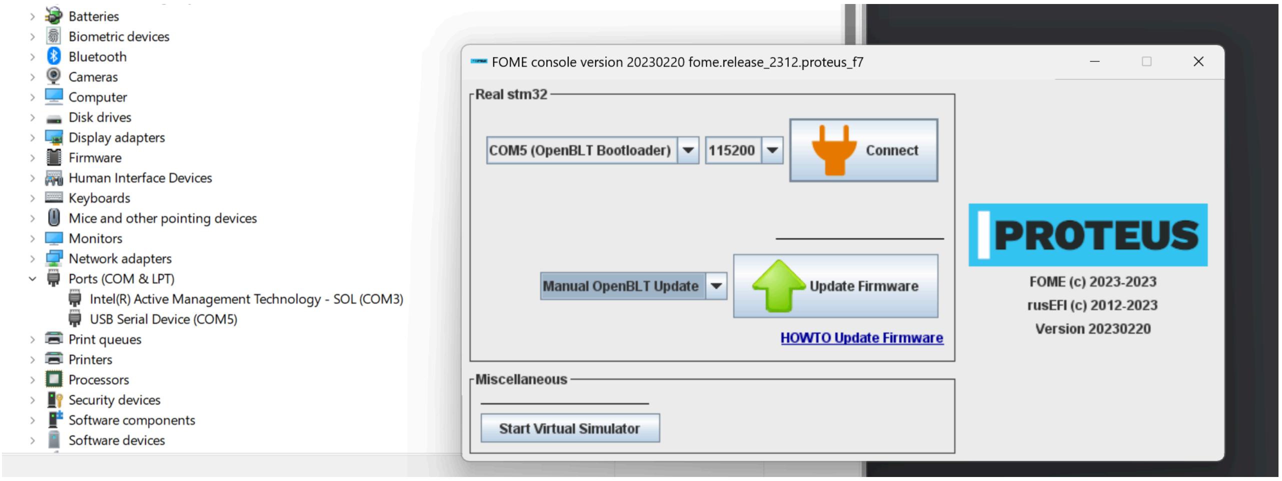
Select "Browse my computer for drivers" and proceed to browse and have Windows update the drivers for STM32 BOOTLOADER by directing it to "../drivers/silent_st_drivers/" directory, which is extracted from the ".../drivers/silent_st_drivers2.exe" archive.

Then, unplug the ECU and retry the firmware udpate from FOME console.

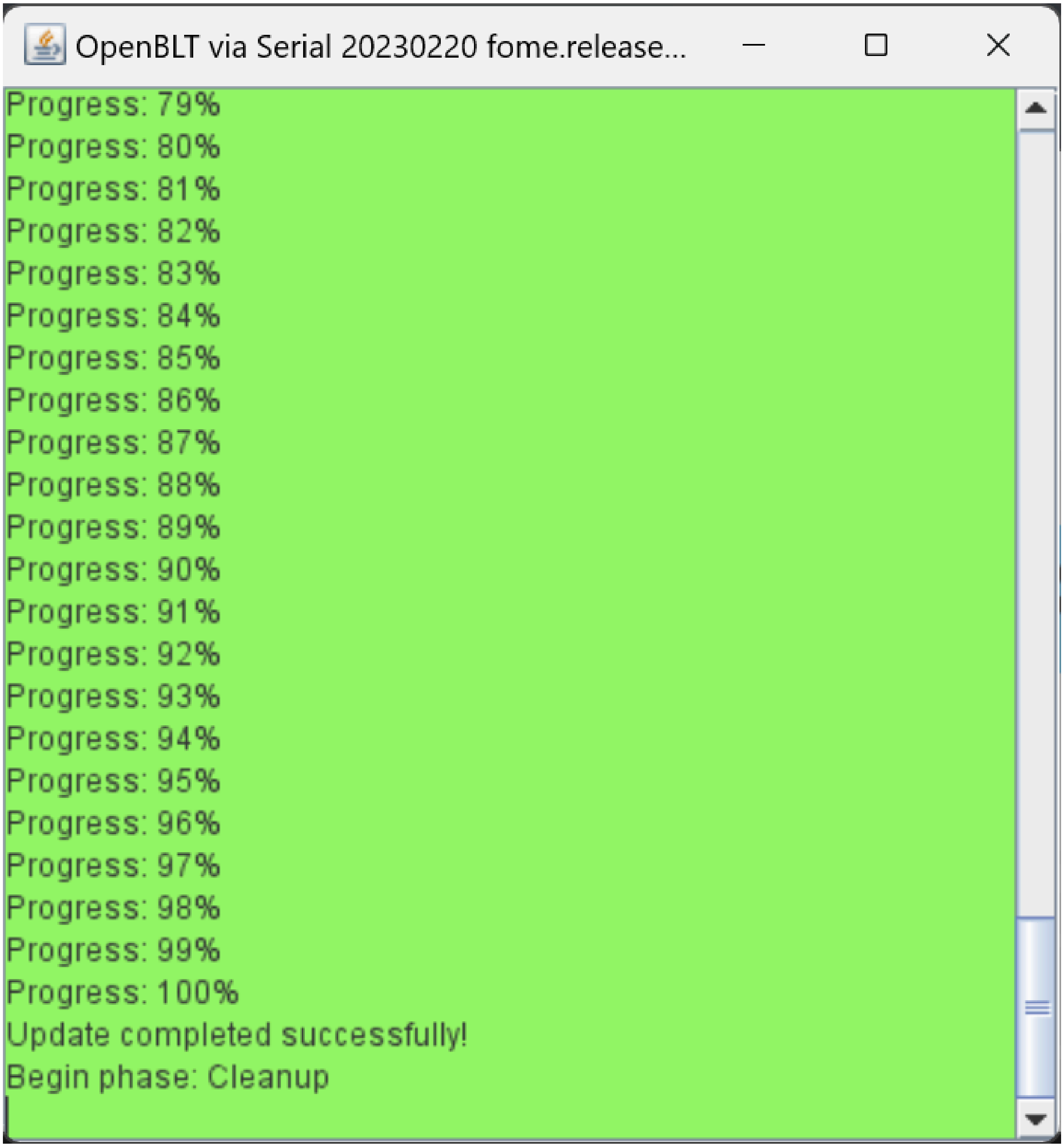
Other DFU update errors may occur, such as "Data mismatch found at address" or "looks like ECU didn't reboot to OpenBLT."



Again, unplug the ECU and retry the firmware update. At this point, FOME console should have detected the ECU as "OpenBLT Bootloader". Proceed by updating the firmware via "Manual OpenBLT Update."



If prompted with "Update completed successfully!", proceed by power cycling the ECU. From this release onwards the ECU can be updated using OpenBLT and should not require any manual driver installs.


OpenBLT via Serial 20230220 fome.release...
Progress: 79%
Progress: 80%
Progress: 81%
Progress: 82%
Progress: 83%
Progress: 84%
Progress: 85%
Progress: 86%
Progress: 87%
Progress: 88%
Progress: 89%
Progress: 90%
Progress: 91%
Progress: 92%
Progress: 93%
Progress: 94%
Progress: 95%
Progress: 96%
Progress: 97%
Progress: 98%
Progress: 99%
Progress: 100%
Update completed successfully!
Begin phase: Cleanup

Discrete-VR

High-Low Driver Circuits

Highside Driver Circuits

Lowside Driver Circuits

PT2001 Low-Z driver

12v, 5v and 3.3v regulation circuits

STM32 Compatibility with FOME

What to write in this section

Details on our various generic driver layouts, what they are and the basics of how they work.

Base Boards

Brain Boards

BeerMoneyMotorsports CAN gauge

A page to outline the hardware and wiring of the BMM KaN Multi Fit CAN gauge.

BMM: KaN Pinout and Wiring

FOME low-z injector driver

Details on the low impedance injector driver board, its wiring and function

Power steering controller

Wideband Oxygen Sensor Configuration Guide

This guide explains how to set up and configure a wideband oxygen sensor with a FOME wideband controller.

Overview

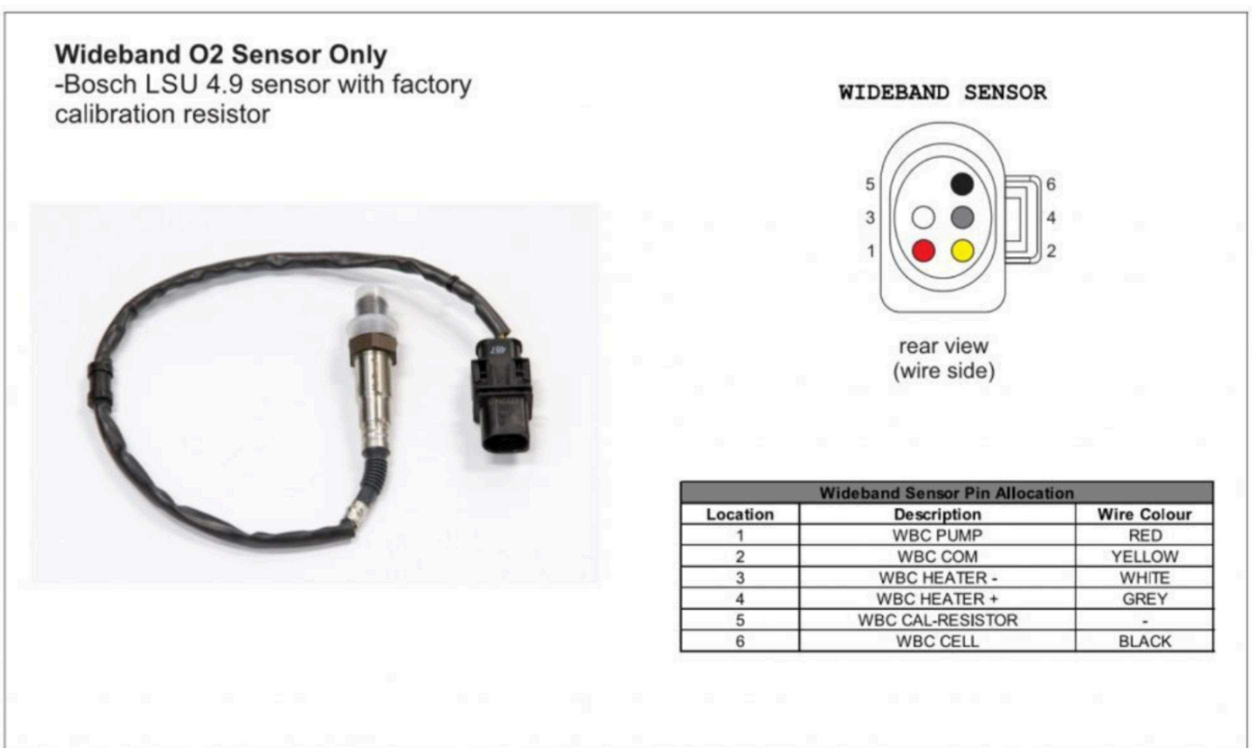
Wideband oxygen sensors are required for accurate AFR measurement. Unlike narrowband sensors that only indicate if the mixture is rich or lean, wideband sensors provide precise AFR readings across a wide range.

Sensor Pinout Reference

LSU 4.9 Sensor Pinout

Pin Number	Function	Description
1	IP	Pump Current
2	VM	Virtual Ground

Pin Number	Function	Description
3	HEAT-	Heater Negative
4	HEAT+	Heater Positive
5	RT	Calibration Resistor
6	UN	Nernst Voltage



Wideband Sensor Pin Allocation		
Location	Description	Wire Colour
1	WBC PUMP	RED
2	WBC COM	YELLOW
3	WBC HEATER -	WHITE
4	WBC HEATER +	GREY
5	WBC CAL-RESISTOR	-
6	WBC CELL	BLACK

Note: The calibration resistor is internal to the sensor. On the sensor connector, there are only 5 wires but 6 pins, as pins 1 and 5 have a laser-trimmed resistor between them.

F042 Controller Module Pinout

Pin Number	Function	Notes
1	Ground	Connect to chassis ground
2	LSU Ip	Connect to sensor pin 1
3	LSU Heater -	Connect to sensor pin 3
4	LSU Heater +	Connect to sensor pin 4
5	No connection	
6	12v Supply	Connect to fuel pump supply or independent relay controlled by ECU
7	No connection	
8	LSU Vm	Connect to sensor pin 2
9	LSU Rtrim	Connect to sensor pin 5

Pin Number	Function	Notes
10	LSU Un	Connect to sensor pin 6
11	CAN H	CAN bus high
12	CAN L	CAN bus low

LED Status Indicators

Blue LED	Green LED	Meaning
off	fast flash	Sensor warming up
off	slow flash	Sensor hot, operating normally
flashing	off	See error code table below
alternating	alternating	Bootloader, see below

Error Codes

When the green LED is off and the blue LED is flashing, this indicates an error. The blue LED will blink a certain number of times, with a 2-second pause, then repeat.

Blinks	Meaning
3	Sensor failed to heat up within 30 seconds
4	Sensor overheated after entering closed loop mode
5	Sensor unexpectedly cold after entering closed loop mode (bad wiring?)

Bootloader LED Codes

Blink pattern	Meaning
alternating slowly	Firmware integrity check failed, please retry firmware upgrade
alternating quickly	Waiting for bootloader entry command, only occurs for ~1 second before launching firmware

Initial Setup

Firmware Installation

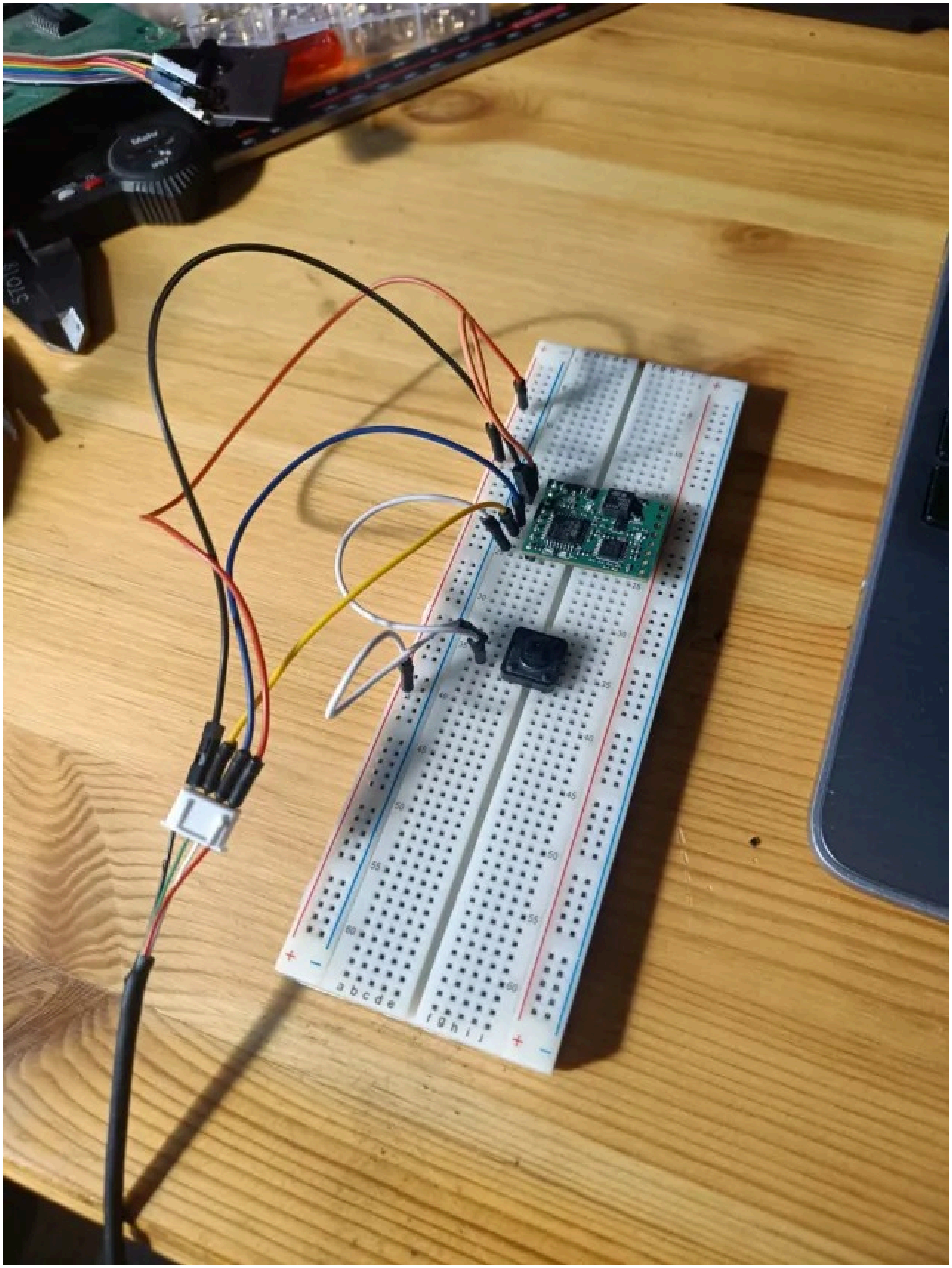
If ordering from somewhere like JLC, wideband modules require one-time initial programming

DFU Method (Standalone WBO2)

For standalone units without a programmer, use the DFU (Device Firmware Upgrade) method:

1. Connect USB directly to the board (5V, GND, DATA+, DATA-)
2. Pull BOOT0 to 5V during power-up to enter DFU mode
3. Make sure the transceiver isn't connected (connector PCB), otherwise USB won't work
4. Use STM32Cube software to flash the firmware (ST Link utility may not work)

Download firmware binary from: mck1117/wideband Releases on GitHub.


a b c d e
f g h i j

Common Issues and Solutions

Extension Cable Issues

When using extension cables, ensure all pins are connected. Some aftermarket extension cables (like certain Innovate models) may omit the calibration resistor pin (Pin 5/RT).

Simulator

In addition to the numerous specific ECU hardware FOME targets, also made available is a build called the FOME Simulator. The FOME Simulator is a special build of FOME designed to run on a modern computer. Naturally, many features are missing as a general computer lacks hardware support, but the core remains. Additionally some functions are adapted to commonly available features and sensors and outputs are mocked.

FOME Simulator not only aids development of FOME by aiding debugging, but also allows users to become familiar with the system and tuning without spending any money and dealing with physical hardware.

FOME Simulator is provided with every release and snapshot bundle, and can be launched directly from the FOME Console or run manually (the `console/fome_simulator.exe` binary).

Currently the FOME Simulator is only built and distributed for Windows (x86_64), but it successfully builds and runs on other platforms like Linux, and even works correctly under Wine.

CAN

Sensors

Outputs

Self Stimulation

Ignition

Ignition

🗃️ Ignition-Hardware

3 items

🗃️ Ignition-Settings

7 items

🗃️ Ignition-Types

4 items

Ignition Modules

place for listing known good ignition modules

Ignition coils

Known good ignition coils

Name	Description
GM LS	Found on most GM LS engines

Coils with known issues

Name	Description
VW Wasted spark	Found on the VR6, etc

Smart and Dumb Coils

Dwell settings

How to correctly configure the firing order and ignition pins

Place to describe the correct settings of firing order with regard to the pin config.

Intake Air Temperature Ignition Adder

Ignition advance table

Ignition coolant correction

Ignition Cylinder Trim

Ignition Mode

See Ignition Types Section?

Individual coils

Something to explain individual coils can be set up in either wasted or sequential

Sequential ignition mode

Single coil or distributor ignition

Wasted Spark Ignition Mode

FOME always fires in wasted spark mode while cranking, as a feature.

Vehicle Specific

Vehicle Specific

🗃️ Mazda-MX5-Miata

6 items

Miata MX5 Quick Start Guide

So, you've just got yourself a shiny new BMM MX5 Miata ECU for your NA or NB, nice! If you're wondering how to install and get it running, you've come to the right place. This guide will cover how to install the ECU to the car with a Bosch LSU 4.9 wideband oxygen sensor and a mass air pressure (MAP) line. Installation of additional sensors or peripherals is covered in the advanced guides.

> ***NOTE:*** Before commencing the ECU installation, it is recommended to jack up the car or drive it onto ramps in the case when the oxygen sensor location is under the vehicle.

Required Tools and Components

- BMM Miata ECU
- BMM wideband adapter harness
- BMM options port pigtail
- Genuine Bosch LSU 4.9 oxygen sensor
- 3 metres of silicone vacuum hose 5/32" or 4mm internal diameter
- 4mm straight barb joiner
- 22mm wrench or 22mm oxygen sensor socket
- Timing light
- USB cable (included with ECU)

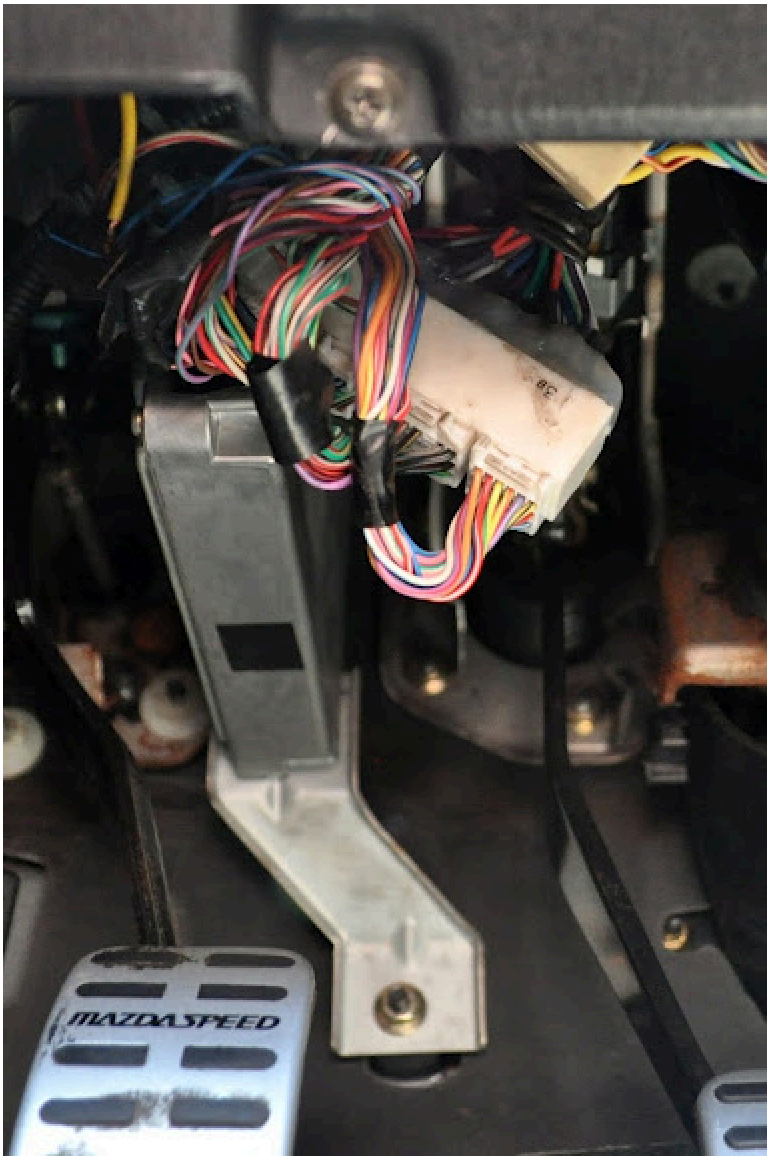
- Windows, Mac or Linux laptop with an installed copy of EFI Analytics TunerStudio
- Spanner and socket set

Removing Original ECU

The stock ECU location for a Miata will be in one of three spots depending on the driving side and year:

Left Hand Drive NB

The ECU can be found above the pedals, next to the steering column.



90-93 Left Hand Drive NA and Right Hand Drive NA/NB

The ECU can be found under the carpet in the passenger side footwell. To access this, the carpet needs to be unhooked from the vertical trim piece on the edge closest to the passenger door. Removing this trim piece can also simplify access. The ECU kick plate will also need to be removed after taking off the five 10mm nuts and bolts holding it in place.

94-97 Right Hand Drive NA

The ECU can be found behind the passenger's seat, under the carpet. Move the seat forwards all the way. Next, the passenger door sill needs to be removed with a philips head screwdriver so that the carpet towards the back of the seat can be pulled back to reveal the ECU.

Once the ECU has been located on your Miata, disconnect the car battery then remove all electrical plugs to the ECU. Un-bolt any remaining ECU mounting brackets from the car with a 10mm socket and the ECU should now be free from the car. The last step is to use a philips head to remove the factory ECU mounting brackets from the stock ECU case for these will be needed to mount the BMM ECU.

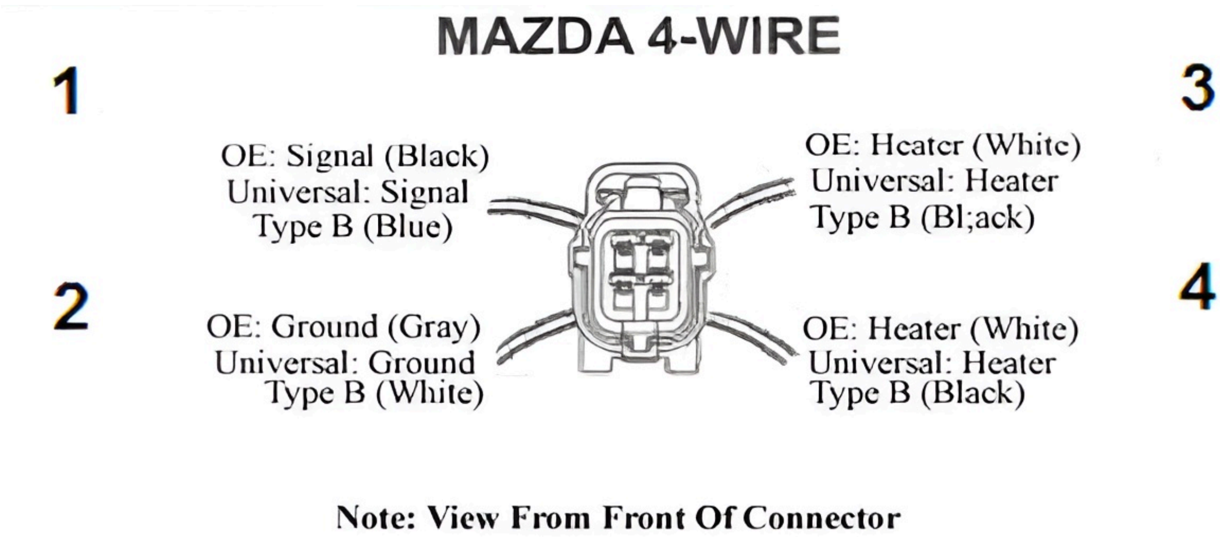
Connecting Wideband Oxygen Sensor

> ***NOTE:*** It is imperative that you use a genuine Bosch LSU 4.9 sensor rather than a cloned product. A fake LSU 4.9 will not provide accurate readings and can cause a lot of headaches down the track. The best way of avoiding a fake sensor is to buy directly from a reputable supplier of vehicle parts rather than generic large online re-sellers. Typical part numbers for this Bosch sensor include: 17025, 17212, 17123 and 17217. The notable difference between these part numbers is the cable length so it is recommended to measure what length you need ahead of time.

Find the factory oxygen sensor on the exhaust and unplug it from the wiring harness. In the case that the car has multiple oxygen sensors, the one to remove is the closest sensor to the engine block before any catalytic converters. Next, unscrew the sensor and replace it with a Bosch LSU 4.9 wide-band sensor. Connect the sensor to the BMM wideband adapter harness. The trailing end of the harness will need to be fed through the firewall into the cabin. The easiest way of doing this, as shown in the image below, is to cut a hole in the nearest firewall bung to the stock ECU location, and feed the cable through that. Cable tie the wiring away from any hot areas of the engine bay. Inside the cabin, connect the wideband adaptor harness plug to the options port pigtail and plug it into the ECU.

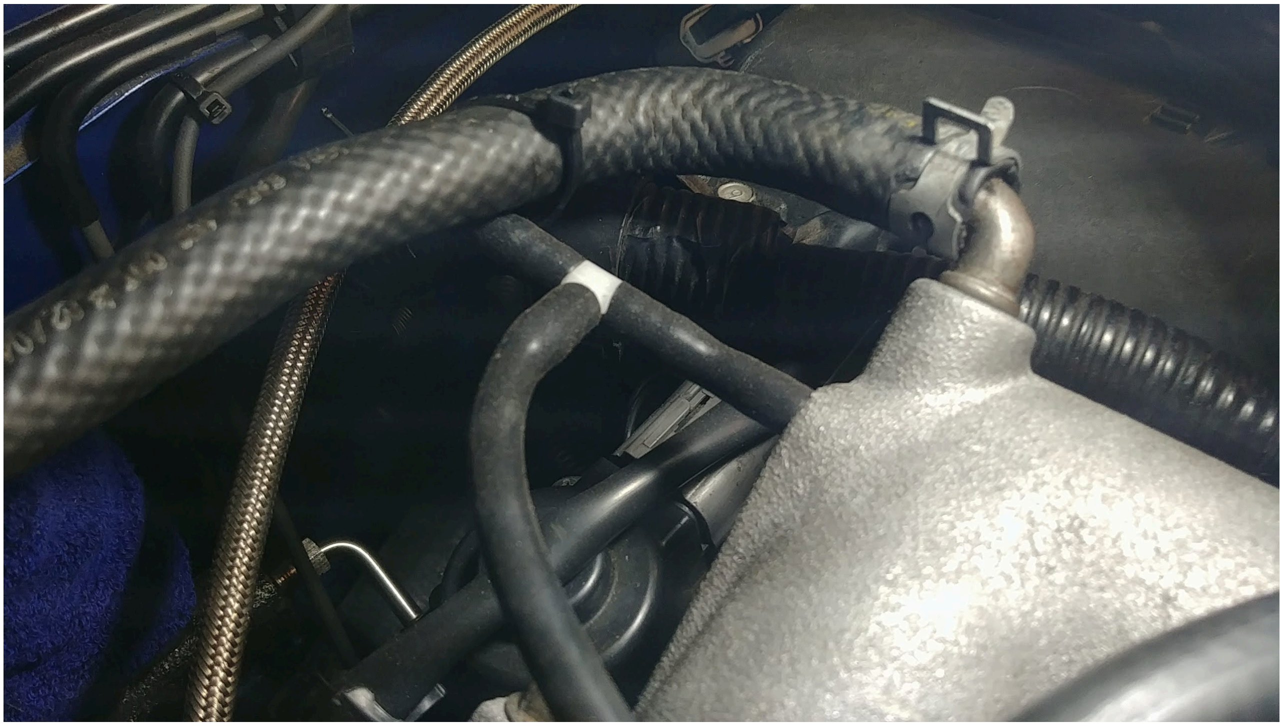
Using an External Wideband Controller

In the case you wish to use an external wideband controller such as an *AEM X-Series Wideband UEGO AFR Sensor Controller Gauge*, the wideband sensor should be plugged into the wideband controller instead of directly into the ECU. The best way to wire in the controller is directly to the old narrow band oxygen sensor plug on the car based off the diagram below. In this diagram, pin 1 goes to the controller analogue output, pin 2 to the signal ground, pin 3 to the controller 12V input and pin 4 to the other controller ground (if applicable). Make sure to double check the voltages on the pins before connecting the controller to them. The external controller also requires additional setup in Tuner Studio which will be covered later.

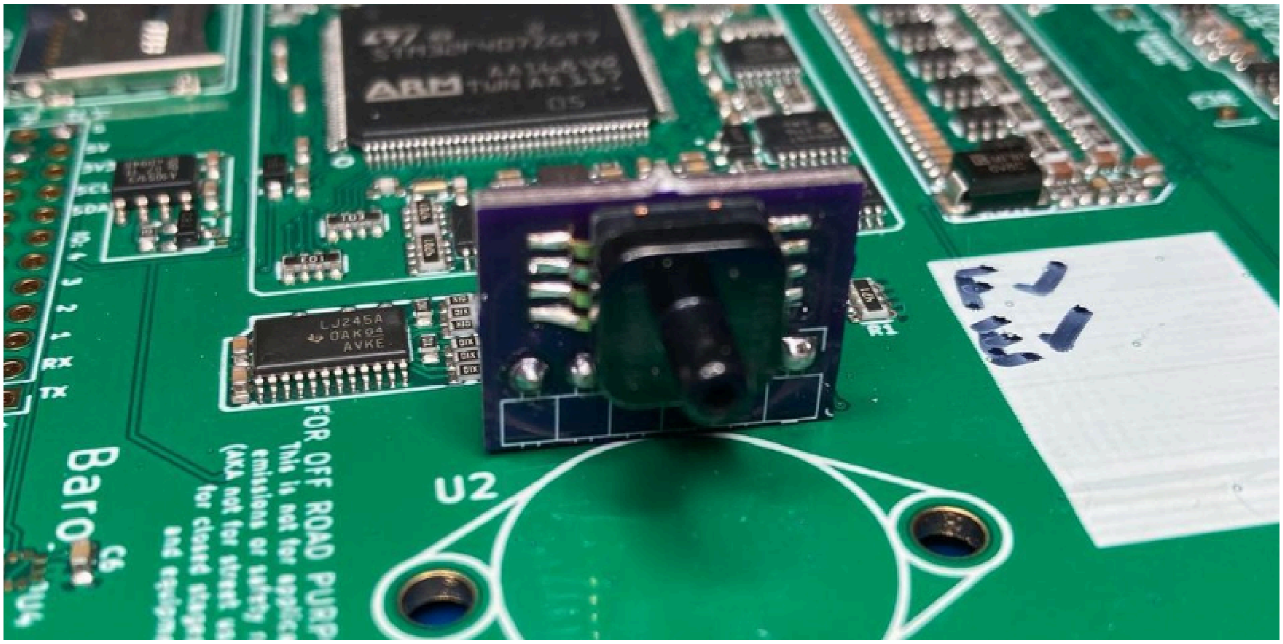


Connecting MAP Line

Look around the intake manifold for any spare vacuum ports that lie after the throttle body and connect the vacuum line. If there are no spare ports, pick one and attach the vacuum line to it using a tee piece. It is recommended but not required to cable tie the vacuum line to the tee. In the image below, there was a free vacuum port on the back of the intake manifold which has been tee'ed off into the MAP line and the blow off valve line (only applicable on turbo charged vehicles).



Like the oxygen sensor, feed the line through the bung in the firewall to the ECU. If you have a 4mm barb joiner, connect the vacuum line to the vacuum line protruding from the BMM ECU case. If you do not have a barb joiner, open up the BMM ECU case with a philips head and feed the vacuum line through the case. Mock up the position of the case in the car before cutting the vacuum line to length. Pull the vacuum line onto the MAP sensor on the ECU (the sensor with the nipple on it pictured below) and optionally fasten it with a small cable tie. The ECU can now be put back into its case.

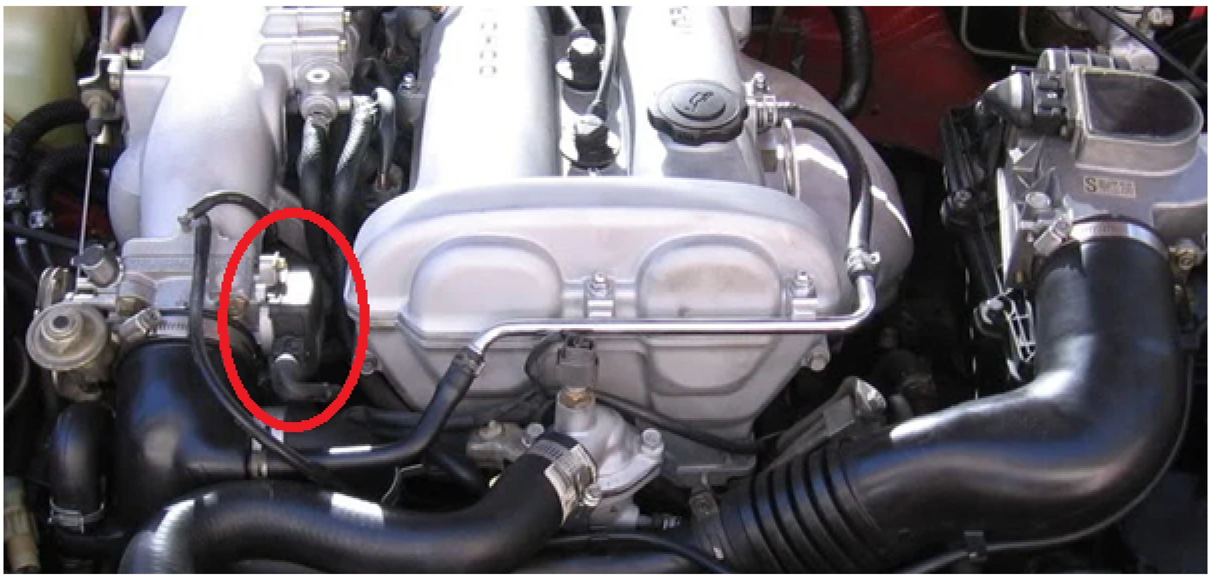


Using the MAP line combined with an intake air temperature (IAT) sensor, the BMM ECU can run the car using what is known as speed-density air metering. This means that you can unplug the mass air flow (MAF) sensor or the air flow meter (AFM) for the NA 1.6L vehicles. Removing these sensors and replacing them with a pod filter directly to the intake can even result in a fractional power increase from the reduction in intake restriction.

Additional Steps for NA6 Vehicles

The NA6 1.6L vehicles which use an AFM instead of a MAF require a few additional modifications to run with a BMM ECU. A manual NA6 do not have a variable throttle position sensor (TPS) like the automatic NA6, later model NAs and all NBs. An NA6 also needs an external intake air temperature (IAT) sensor wired in as the AFM which has one inside is typically removed. They also require a jumper for the ECU to control the fuel pump which was previously the job of the air flow meter. Additionally, it is recommended to ground the sensors to the ECU rather than the chassis as detailed in the NA6 ECU Grounding Guide.

The first step is to disconnect the factory TPS sensor. **This is very important or it will cause a short circuit later.**. The TPS sensor location is shown in the image below.

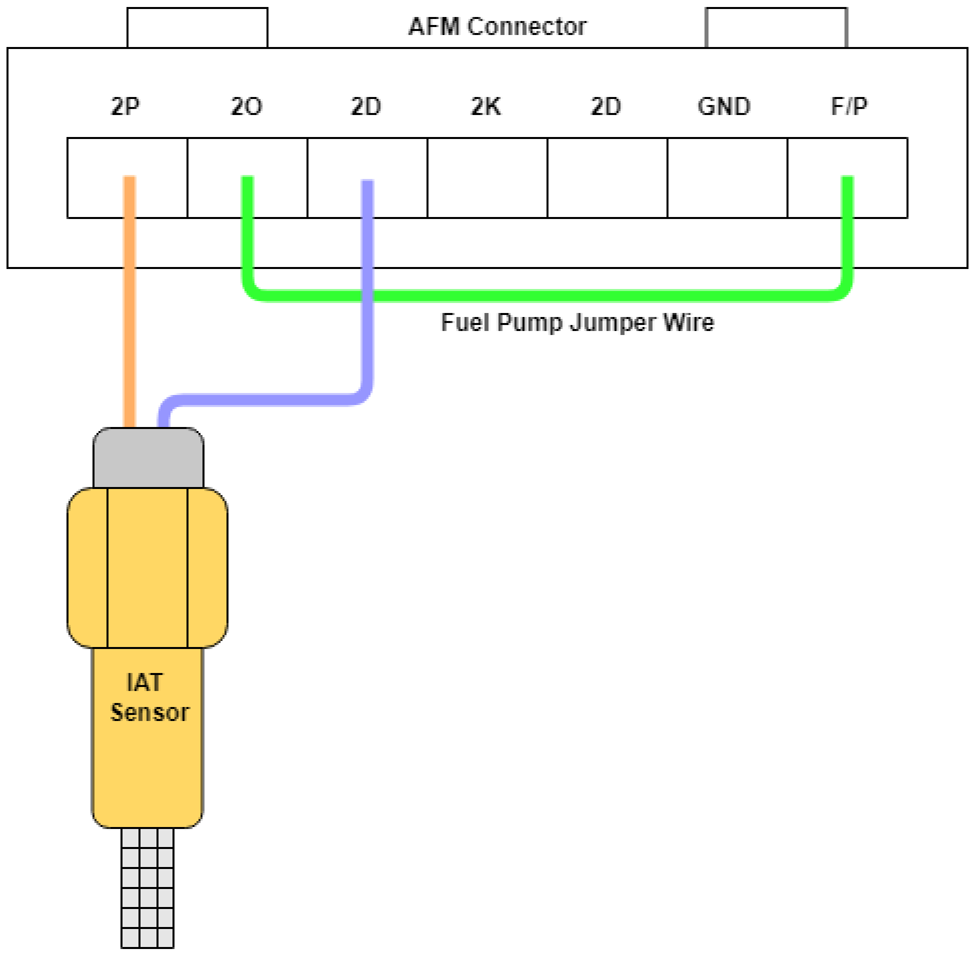


The BMM ECUs for this vehicle include a KIA TPS and adapter. The KIA TPS will plug straight to the OEM TPS plug without any additional wiring. If using another variable TPS that requires re-wiring, the NA6 TPS connector pinout is as follows:

Function	Cable Colour
Signal	Green/White

Function	Cable Colour
Ground	Black/Green
5V Reference	Red

The next step is to wire up the IAT sensor and to add a jumper wire to the AFM connector as per the wiring diagram below. Any IAT sensor with two wires can be used although a GM IAT sensor is recommended as FOME already has a configuration for it. As the IAT is a resistance-based sensor, the orientation of the wires does not matter.



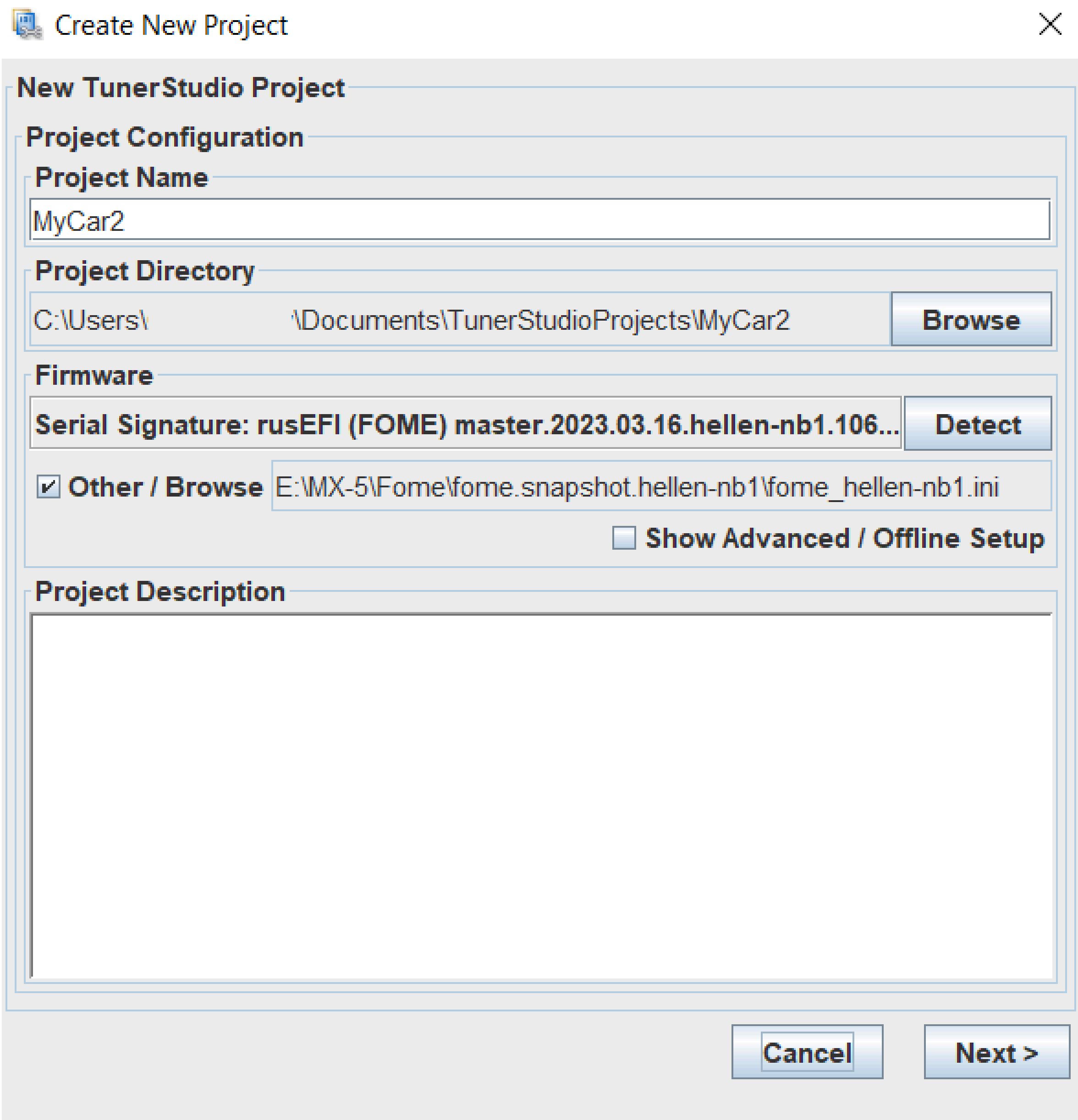
Connecting the ECU

Now that the MAP line and wideband are connected to the ECU, the remaining wiring harness plugs from the OEM wiring loom can be plugged into the ECU. Take the factory ECU mounts and attach them to the BMM ECU case. The ECU can now be re-installed into the factory location. The car battery can now be re-connected.

Tuner Studio Setup

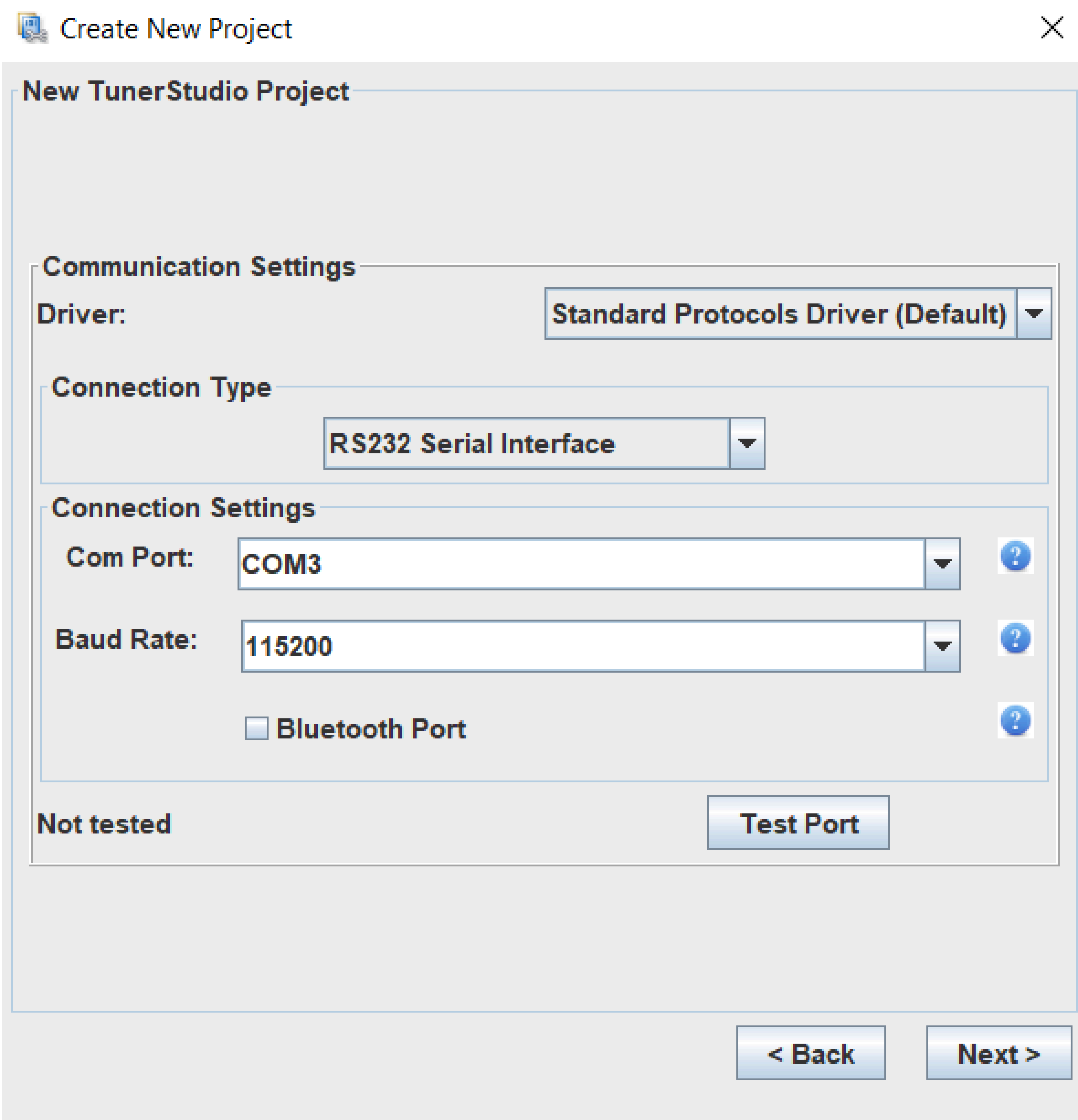
Make sure that you have downloaded the latest version of TunerStudio (TS) from EFI Analytic's site. Although the base version of the software is free, it is strongly recommended to buy a license for the additional features including auto-tuning and the ability to customize the default dashboard.

Begin the setup by plugging the ECU into the laptop and opening TS. Create a new project and click *detect* under firmware. Select the COM port corresponding to the FOME ECU in the device list. If the COM port cannot be found or the firmware cannot be automatically detected, click *Other/ Browse* and load the .ini file for the ECU which can either be downloaded or found within the ZIP file on the USB device which appears when the ECU is plugged into the computer.



In the next dialog choose between lambda or air fuel ratio (AFR) as your display units. lambda is recommended as it is easier to comprehend and tune with. For example, the ideal or stoichiometric AFR for regular petrol is 14.7 (14.7 parts air to 1 part fuel) which corresponds with a lambda of 1. Lambda represents the percentage of air in the combustion chamber compared to the amount needed for ideal or stoichiometric combustion to occur. If a car is running 10% lean, the AFR would be 16.17 and lambda would be 1.1. If the car is 10% rich, AFR would be 13.36 and lambda would be 0.9. Looking at lambda, it is instantly obvious what percentage rich or lean the engine is running but with AFR, it requires more effort. **The only time AFR should be selected here is if you are using an external wideband controller**.

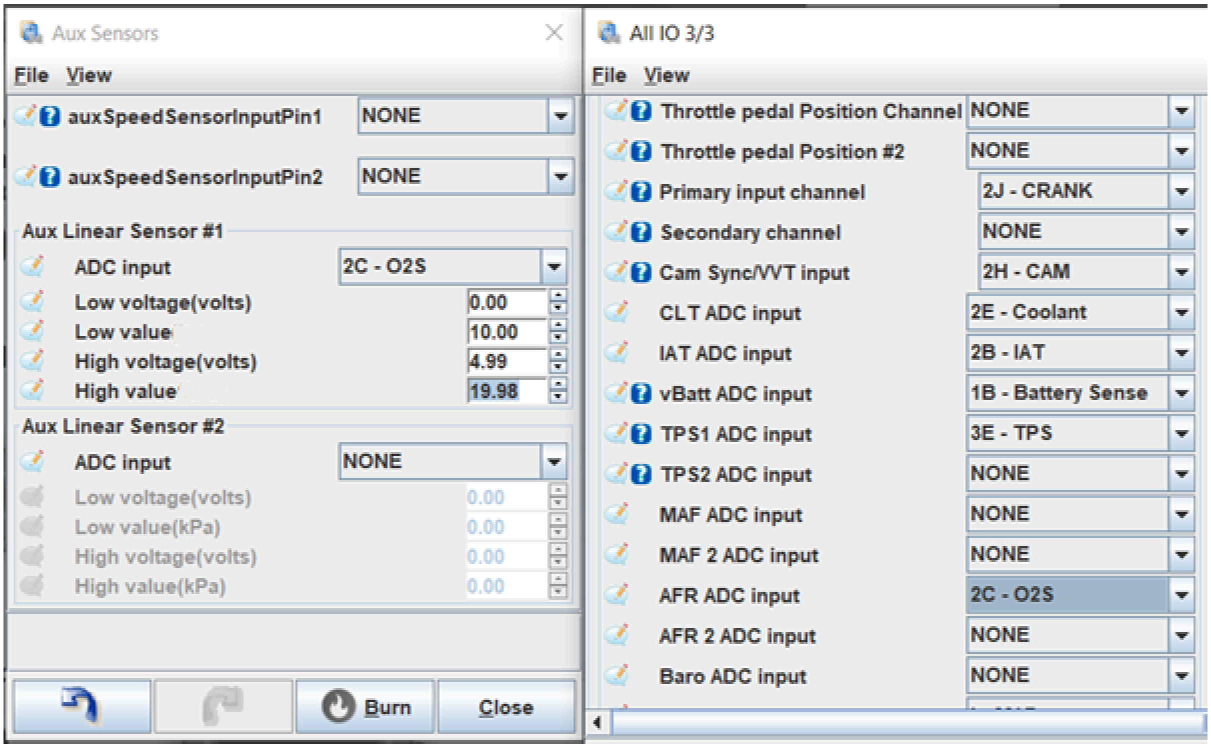
In the third dialog box, configure it as shown in the image below but select the com port which corresponds to your ECU. If unsure, go to the device manager on your computer and it should list the COM port number next to the name of the ECU. Click *Test Port* and if successful, move to the next dialog.



In the final dialog box, select the default gauge layout (you can change this later as you wish) and click *finish*. The last step before cranking the engine is to click the *Ignition* button to open the ignition settings and change the timing mode from *dynamic* to *fixed* and the fixed timing setting to 10 degrees. This will lock the engine to operate at 10 degrees of timing so that you can set the base timing.

Additional Tuner Studio Steps for an External Wideband Controller

To set up the external wideband controller there are several additional steps in Tuner Studio. First, your display units should be set to AFR for this as already stated. If you forgot to do this earlier, press *CTRL + P* to open the vehicle properties. Now, open the *Aux Sensors* dialog under *Sensors* and the *Full Pinout 3/3* dialog under *Controllers*. As per the diagram below, set the *AFR ADC Input* and *ADC Input* to the pin corresponding with *O2S* (pin 2C for the example). for the values in the *Aux Linear Sensor #1* box you need to reference the manual of your wideband controller for what voltages correspond to its AFR outputs. In the example below, 0V corresponds to an AFR of 10.0 and 4.99V corresponds to an AFR of 19.98. Once these are set, click *Burn*. If TS does not exactly correspond to the readings on your wideband, you can adjust with the correction value.



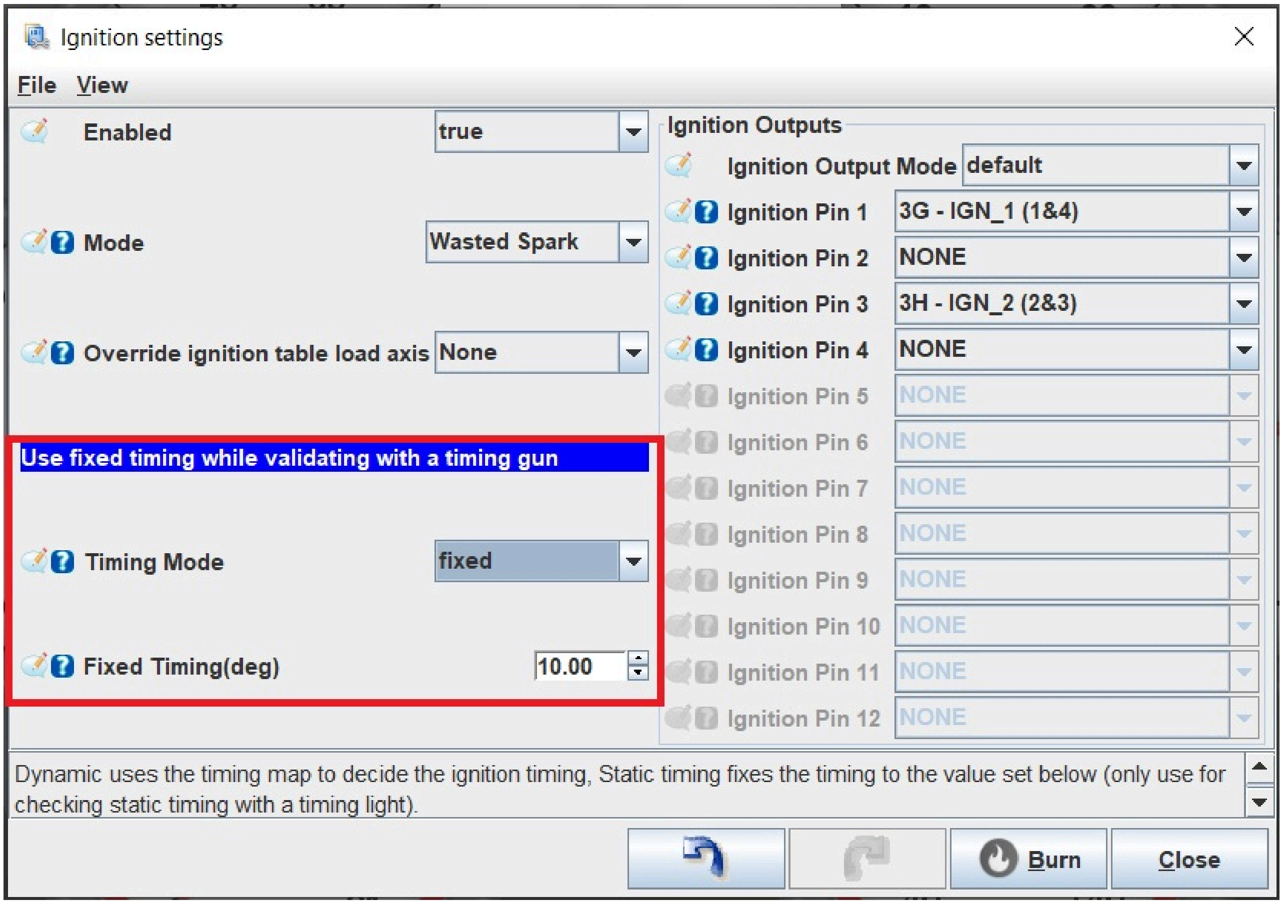
After completing all of the setup steps, you can go ahead and turn the car key two clicks to *ON* and listed for the fuel pump priming. Once the fuel pump has primed, go ahead and start the engine. Let it run for a few seconds and turn it off again. **Do not drive the vehicle yet, there are still several steps to complete before the car is ready for a drive**.

Set Base Timing

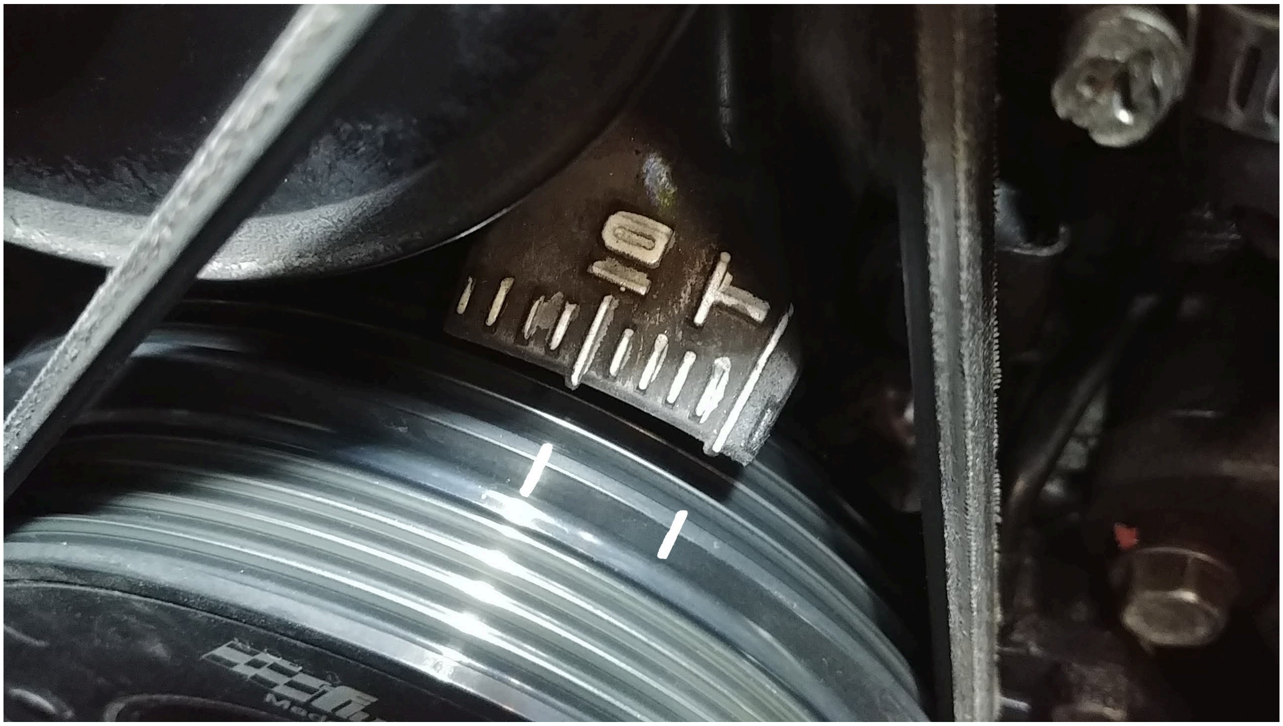
The car should start on the base map although once it is running, the base timing needs to be set up. This syncs the timing between the ECU and the car so that they are both reading the same values. Typically. the base timing will be a few degrees out from the base map as it varies slightly from car to car.

To set the base timing, connect the timing light power to a spare 12V battery and the inductive clamp onto the cylinder 1 spark plug lead (the closest spark plug to the front of the engine bay). Ensure that the arrow on the inductive clamp is pointing along the wire towards the spark plug, not towards the coil pack.

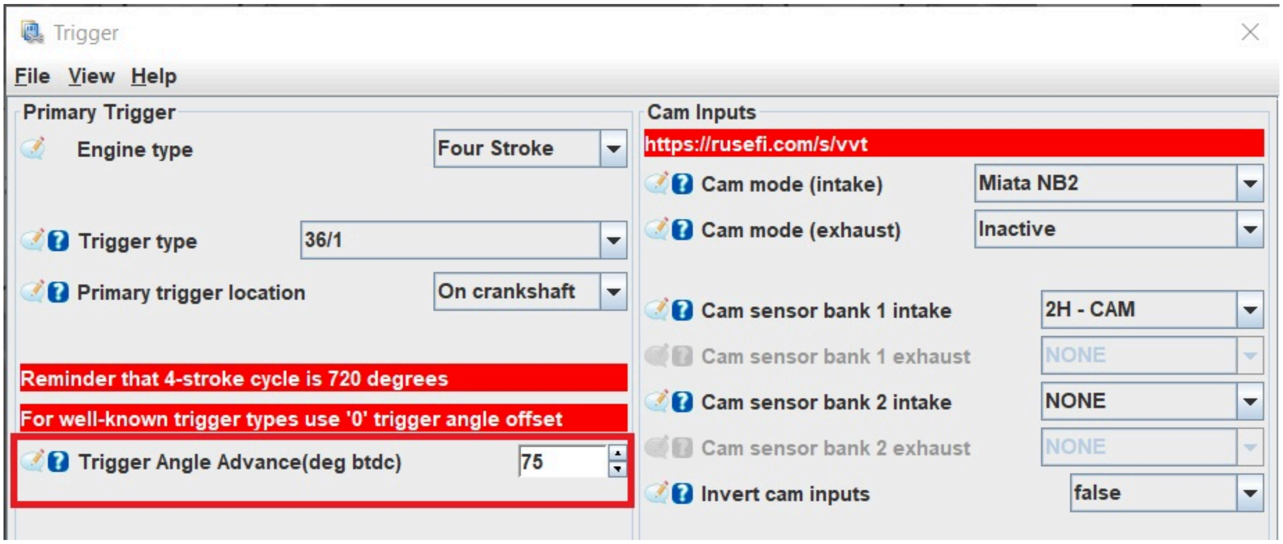
In TS, under *Ignition > Ignition Settings*, set the timing mode to *fixed* and *10* degrees then burn the configuration.



Now start the car and hold the timing gun trigger, shining the light onto the bottom harmonic damper pulley on the front of the engine. This pulley has two timing marks on it and a labelled cover above it. When the timing is spot on, these marks on the rotating pulley will line up with the *10* and *T* marks on the cover as shown below.



If your timing marks do not line up like in the image above, you will need to change the base timing. Count how many marks the timing is off by and turn the car engine off. In TS, go to *Base Engine > Trigger* and increase/decrease the *Trigger Advance Angle* by the amount of marks the timing was off by then burn the configuration. Repeat this process until the timing marks line up then change the timing mode back from *fixed* to *dynamic*.

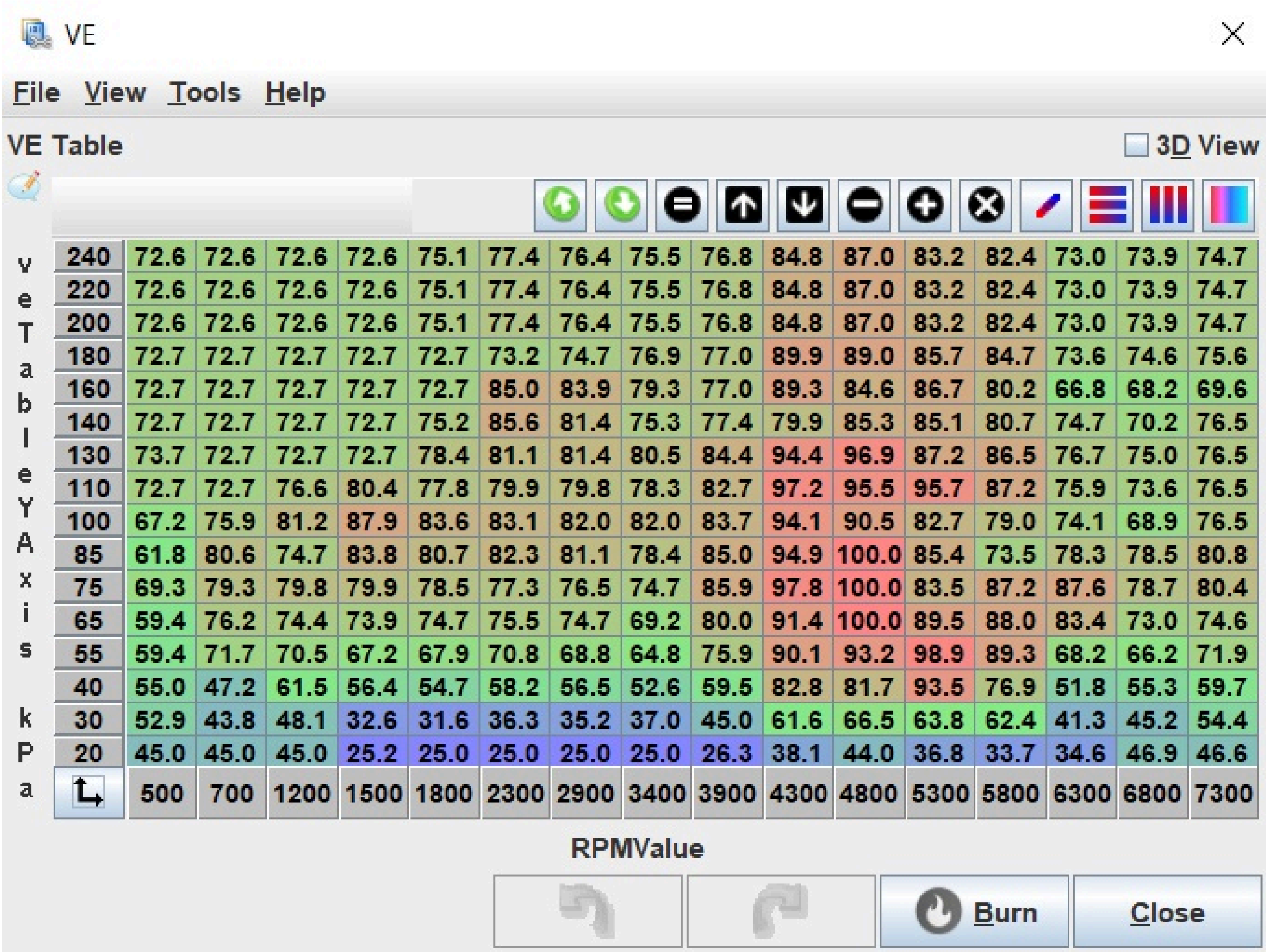


First Drive and Tuning the VE Table

Everything is now ready to take your Miata for its first drive. You can't go and thrash it straight away though as the VE table which determines the fuelling needs to be tuned for your vehicle. Before you take the car for a drive, make sure your laptop is charged.

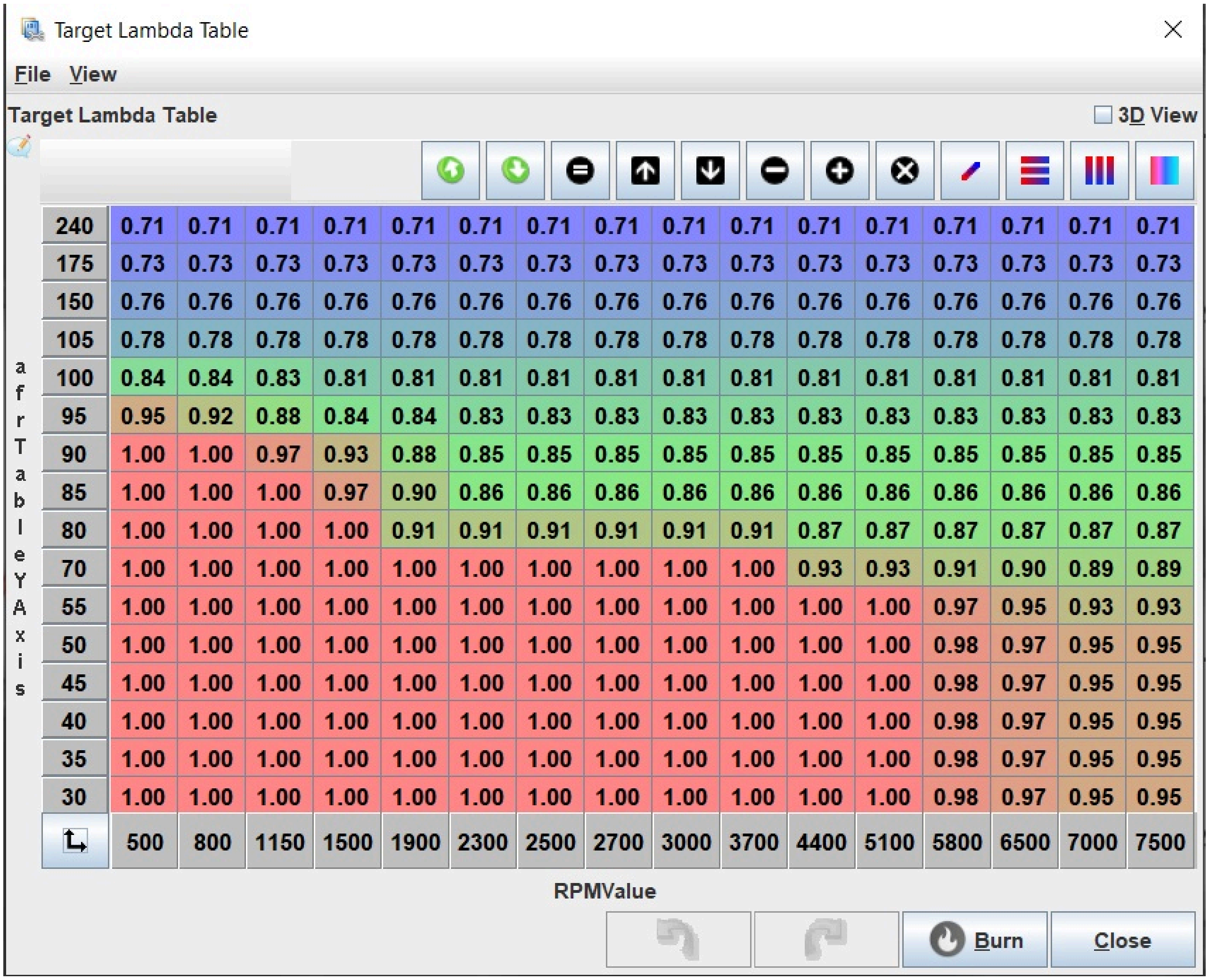
Start the car and plug the laptop in. Within 30 seconds, the lambda gauge should wake up and start displaying a value. For now, you want that value to be around 1 meaning that the exact amount of fuel is being supplied to the engine for perfect combustion to occur. To change the lambda value, you need to modify the VE Table under *Fuel > VE*. VE stands for volumetric efficiency which is the efficiency that the engine can move the fuel and air mix into and out of the cylinders. An example of a VE table is shown below (do not copy this table as it is off a highly modified vehicle). The table adjusts the VE percentage (represented by the numbers on the grid) based on the engine speed (represented as revolutions per minute -

RPM) and engine load (represented as the MAP). With the engine running, blip the throttle and see how the indicator moves around the different table cells as the engine state changes.



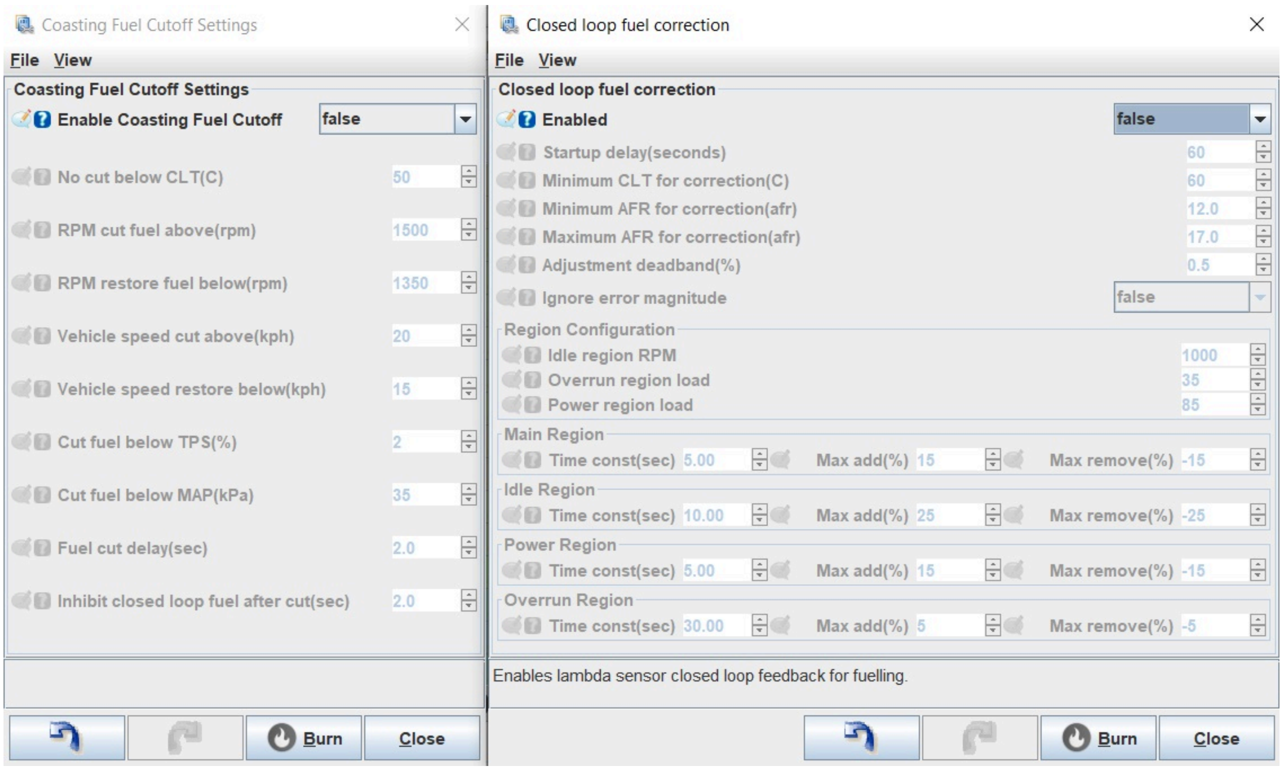
veTableYAxis (kPa) \ RPMValue	500	700	1200	1500	1800	2300	2900	3400	3900	4300	4800	5300	5800	6300	6800	7300
240	72.6	72.6	72.6	72.6	75.1	77.4	76.4	75.5	76.8	84.8	87.0	83.2	82.4	73.0	73.9	74.7
220	72.6	72.6	72.6	72.6	75.1	77.4	76.4	75.5	76.8	84.8	87.0	83.2	82.4	73.0	73.9	74.7
200	72.6	72.6	72.6	72.6	75.1	77.4	76.4	75.5	76.8	84.8	87.0	83.2	82.4	73.0	73.9	74.7
180	72.7	72.7	72.7	72.7	72.7	73.2	74.7	76.9	77.0	89.9	89.0	85.7	84.7	73.6	74.6	75.6
160	72.7	72.7	72.7	72.7	72.7	85.0	83.9	79.3	77.0	89.3	84.6	86.7	80.2	66.8	68.2	69.6
140	72.7	72.7	72.7	72.7	75.2	85.6	81.4	75.3	77.4	79.9	85.3	85.1	80.7	74.7	70.2	76.5
130	73.7	72.7	72.7	72.7	78.4	81.1	81.4	80.5	84.4	94.4	96.9	87.2	86.5	76.7	75.0	76.5
110	72.7	72.7	76.6	80.4	77.8	79.9	79.8	78.3	82.7	97.2	95.5	95.7	87.2	75.9	73.6	76.5
100	67.2	75.9	81.2	87.9	83.6	83.1	82.0	82.0	83.7	94.1	90.5	82.7	79.0	74.1	68.9	76.5
85	61.8	80.6	74.7	83.8	80.7	82.3	81.1	78.4	85.0	94.9	100.0	85.4	73.5	78.3	78.5	80.8
75	69.3	79.3	79.8	79.9	78.5	77.3	76.5	74.7	85.9	97.8	100.0	83.5	87.2	87.6	78.7	80.4
65	59.4	76.2	74.4	73.9	74.7	75.5	74.7	69.2	80.0	91.4	100.0	89.5	88.0	83.4	73.0	74.6
55	59.4	71.7	70.5	67.2	67.9	70.8	68.8	64.8	75.9	90.1	93.2	98.9	89.3	68.2	66.2	71.9
40	55.0	47.2	61.5	56.4	54.7	58.2	56.5	52.6	59.5	82.8	81.7	93.5	76.9	51.8	55.3	59.7
30	52.9	43.8	48.1	32.6	31.6	36.3	35.2	37.0	45.0	61.6	66.5	63.8	62.4	41.3	45.2	54.4
20	45.0	45.0	45.0	25.2	25.0	25.0	25.0	25.0	26.3	38.1	44.0	36.8	33.7	34.6	46.9	46.6

The general way to tune the VE table is to go through all the cells which the engine will operate within and to adjust the VE percentage until the lambda gauge matches the value in the *Target Lambda Table* shown below and in TS found under *Fuel > Target Lambda*. For example, if the lambda gauge shows 1.1 and the target lambda for that engine state is 1.0, the corresponding VE cell needs to be increased by 10%. The target lambda table supplied with the Miata base map should be sufficient to start with but you can modify it later to make the engine run richer or leaner under certain conditions such as boost or highway cruising respectively.

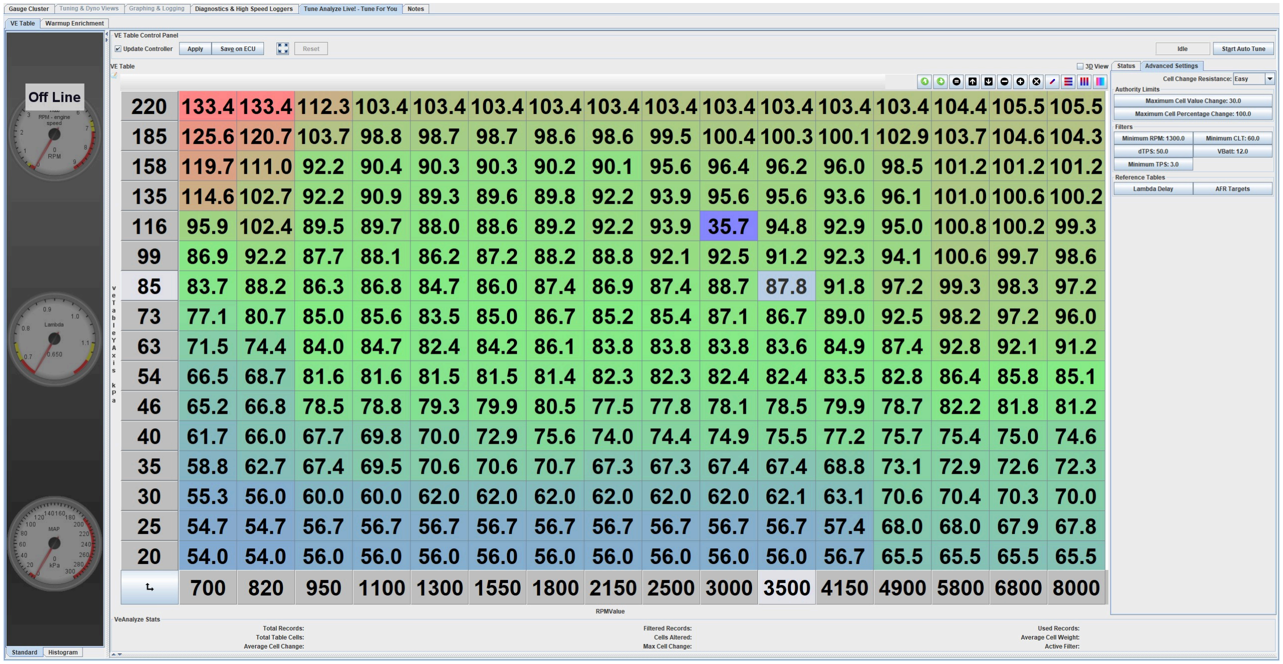


afrTableYAxis \ RPMValue	500	800	1150	1500	1900	2300	2500	2700	3000	3700	4400	5100	5800	6500	7000	7500
240	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71	0.71
175	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73	0.73
150	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76	0.76
105	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78	0.78
100	0.84	0.84	0.83	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81	0.81
95	0.95	0.92	0.88	0.84	0.84	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83	0.83
90	1.00	1.00	0.97	0.93	0.88	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
85	1.00	1.00	1.00	0.97	0.90	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86	0.86
80	1.00	1.00	1.00	1.00	0.91	0.91	0.91	0.91	0.91	0.91	0.87	0.87	0.87	0.87	0.87	0.87
70	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.93	0.93	0.91	0.90	0.89	0.89
55	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.97	0.95	0.93	0.93
50	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
45	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
40	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
35	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95
30	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.98	0.97	0.95	0.95

There are three ways of tuning the VE table. The first way is to drive the car around smoothly as a mate in the passenger seat goes through and changes the VE values until the lambda gauge matches the target lambda. The second and easier way is to use the TS autotuner which is only available in the full version of TS but absolutely worth it. To tune this way, you first need to disable some parameters. Under *Fuel*, open *Closed* *loop fuel correction* and *Deceleration fuel cut off (DFCO)*, set them both to false and click *burn* with the engine off. The third and easiest (yet most expensive option) is to take the car to a dyno for tuning where they will do either the first or second option themselves. The advantage of a dyno is that they can set it to bring the engine into any state they wish to perfectly configure the VE table.



Next, click the tab labelled *Tune Analyze Live! - Tune For You* to bring up the autotuner. Click to the *Advanced Settings* tab and configure it as shown in the image below. These configuration settings are deliberately quite loose so that TS can quickly tune the general shape of the VE table. On the left side of the *VE Table Control Panel*, you also need to check the box marked *Update Controller* which ensures that the VE table is updated on the ECU as the autotune corrects itself.



220	133.4	133.4	112.3	103.4	103.4	103.4	103.4	103.4	103.4	103.4	103.4	103.4	103.4	104.4	105.5	105.5
185	125.6	120.7	103.7	98.8	98.7	98.7	98.6	98.6	99.5	100.4	100.3	100.1	102.9	103.7	104.6	104.3
158	119.7	111.0	92.2	90.4	90.3	90.3	90.2	90.1	95.6	96.4	96.2	96.0	98.5	101.2	101.2	101.2
135	114.6	102.7	92.2	90.9	89.3	89.6	89.8	92.2	93.9	95.6	95.6	93.6	96.1	101.0	100.6	100.2
116	95.9	102.4	89.5	89.7	88.0	88.6	89.2	92.2	93.9	35.7	94.8	92.9	95.0	100.8	100.2	99.3
99	86.9	92.2	87.7	88.1	86.2	87.2	88.2	88.8	92.1	92.5	91.2	92.3	94.1	100.6	99.7	98.6
85	83.7	88.2	86.3	86.8	84.7	86.0	87.4	86.9	87.4	88.7	87.8	91.8	97.2	99.3	98.3	97.2
73	77.1	80.7	85.0	85.6	83.5	85.0	86.7	85.2	85.4	87.1	86.7	89.0	92.5	98.2	97.2	96.0
63	71.5	74.4	84.0	84.7	82.4	84.2	86.1	83.8	83.8	83.8	83.6	84.9	87.4	92.8	92.1	91.2
54	66.5	68.7	81.6	81.6	81.5	81.5	81.4	82.3	82.3	82.4	82.4	83.5	82.8	86.4	85.8	85.1
46	65.2	66.8	78.5	78.8	79.3	79.9	80.5	77.5	77.8	78.1	78.5	79.9	78.7	82.2	81.8	81.2
40	61.7	66.0	67.7	69.8	70.0	72.9	75.6	74.0	74.4	74.9	75.5	77.2	75.7	75.4	75.0	74.6
35	58.8	62.7	67.4	69.5	70.6	70.6	70.7	67.3	67.3	67.4	67.4	68.8	73.1	72.9	72.6	72.3
30	55.3	56.0	60.0	60.0	62.0	62.0	62.0	62.0	62.0	62.0	62.1	63.1	70.6	70.4	70.3	70.0
25	54.7	54.7	56.7	56.7	56.7	56.7	56.7	56.7	56.7	56.7	56.7	57.4	68.0	68.0	67.9	67.8
20	54.0	54.0	56.0	56.0	56.0	56.0	56.0	56.0	56.0	56.0	56.0	56.7	65.5	65.5	65.5	65.5
↳	700	820	950	1100	1300	1550	1800	2150	2500	3000	3500	4150	4900	5800	6800	8000

Setting	Value
Cell Change Resistance	Easy
Max Cell Value Change	30
Max Cell Percentage Change	100
Minimum Rpm	1300
Minimum CLT	60
dTPS	50
vBatt	12
Minimum TPS	3

Now that the autotuner is set up, start the car and click *Start Auto Tune* on the autotuner. Let the car idle in park whilst it gets up to the minimum temperature. While this happens, you can change the idle cells in the VE table to get them to a lambda of 1. Follow the Cursor on the VE table while idling and adjust the values your car is at while monitoring the lambda value. Once the car has warmed up its time to autotune. After you are sufficiently happy, click Stop Auto Tune, turn the engine off and click Save on ECU. Below is an autotune guideline.

Safety First

- Conduct tuning in a safe environment, such as a closed course or dyno, if possible.
- Have a second person to monitor the software and make adjustments, allowing the driver to focus on driving.

Driving Guidelines

1. Warm Up the Engine

- Start with the engine at normal operating temperature to ensure accurate readings.

2. Cover a Range of Operating Conditions

- **Idle:** Let the engine idle for a few minutes to manually tune and collect data at low load and RPM.
- **Cruising:** Drive at a steady speed on flat roads to cover light to moderate load and mid-RPM ranges.

- **Acceleration:** Perform smooth accelerations from low to high RPM to gather data across a range of engine loads.
- **Deceleration:** Allow the car to decelerate without braking to gather data on how the engine behaves under low or negative load.
- **Full Throttle:** Perform wide-open throttle (WOT) pulls if safe to do so, to collect data at high load and RPM.

3. Vary the Driving Conditions

- Include different gears to change the load on the engine.
- Use hills or inclines to increase the load and vary the driving conditions.
- Drive at different speeds and RPM ranges to ensure a comprehensive dataset.

4. Smooth and Controlled Inputs

- Avoid sudden throttle changes, as they can cause transient conditions that may not provide useful steady-state data.
- Aim for smooth transitions in throttle, brake, and gear changes.

Logging and Reviewing Data

1. Continuous Logging

- Ensure the TunerStudio software logs continuously during the entire drive.
- Monitor real-time Lambda readings and make sure they are within safe limits to avoid engine damage.

2. Data Review

- After collecting sufficient data, stop and review the logs to identify any areas where the lambda deviates from the target.
- The log can be used to review areas where adjustments to the VE table are needed.

3. Initial Adjustments

- Make initial adjustments to the VE table based on the collected data. Such as increasing a few cells by 3-4% and smoothing the area.

Iterative Process

1. Repeat and Refine

- Drive again, following the same driving guidelines to gather more data and further refine the settings in autotune to achieve a better VE table.
- Multiple iterations with multiple settings may be necessary to achieve an optimal VE table.
- You can Lock cells by highlighting them and locking them. Such as Idle or Coasting sections.

For Example:

Setting	Value
Cell Change Resistance	Hard / Very Hard
Max Cell Value Change	10 / 5
Max Cell Percentage Change	30 / 10
Minimum Rpm	1300
Minimum CLT	60
dTPS	50
vBatt	12
Minimum TPS	3

2. Monitor Consistency

- Continuously monitor the consistency of lambda readings against the target table.
- Ensure that adjustments lead to stable and reliable engine performance.

3. Fine-Tuning

- After a few iterations, fine-tune specific areas where the lambda is still not meeting the target.
- Focus on problematic zones, such as transitions between idle and

cruise, or low RPM and high load conditions.

Summary

The process of auto-tuning the VE table using a wideband lambda sensor and a target AFR table involves systematic and controlled driving to gather data across a range of operating conditions. By following the outlined steps, the software can make accurate adjustments to the VE table, resulting in improved engine performance and efficiency. Remember, safety is paramount, and tuning should always be conducted in a controlled environment.

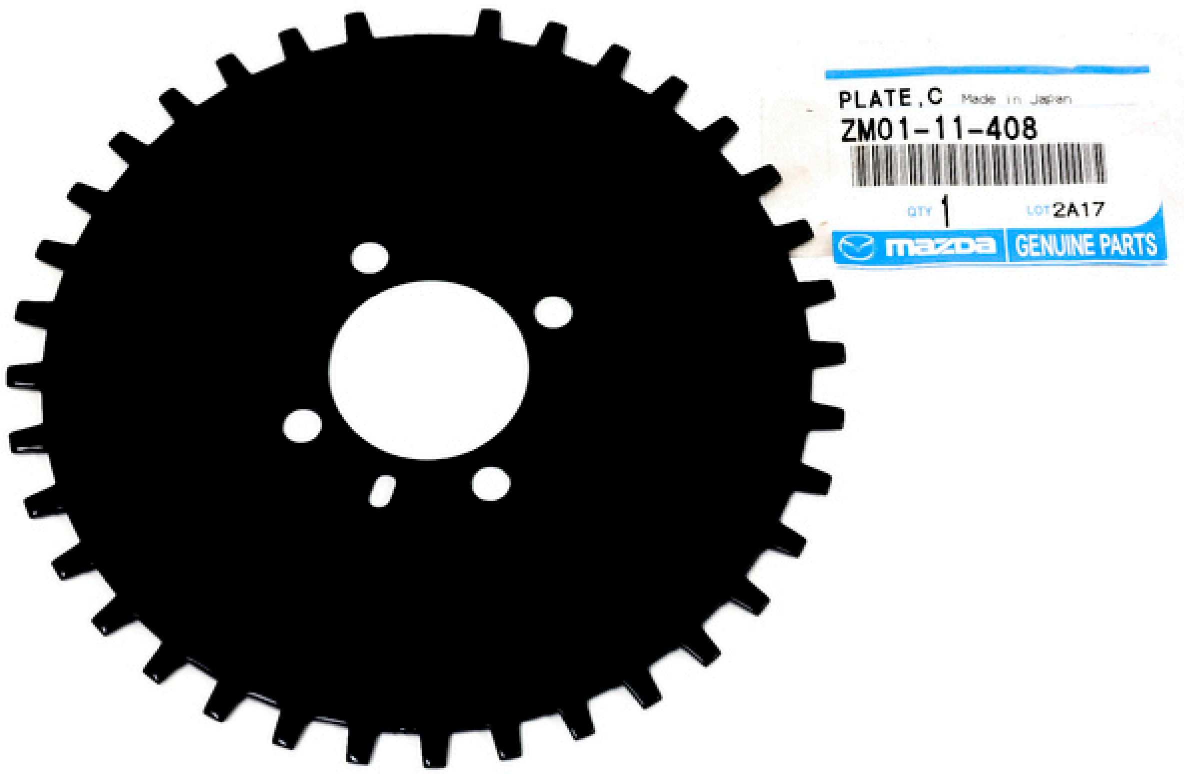
When you are satisfied with your VE table, turn closed loop fuel correction and *Deceleration fuel cut off (DFCO)* back to true. You don't actually need DFCO to be enabled although it will save fuel by turning the injectors off when the car is rolling in gear. Your Miata should now be relatively safe to drive but this is only the start of the tuning journey. As you read through the more advanced guides in this wiki, you will learn about all of the different ways the ECU can be configured to improve the drivability and squeeze every drop of performance out of your Miata.

Miata 36T Trigger Wheel Installation

A trigger wheel with more teeth on the crankshaft provides a finer resolution of the crankshaft position signal. This finer resolution enhances timing control, reduces signal noise and vibration, and improves performance, especially at high RPM. Overall, it contributes to better engine stability and drivability, making it particularly beneficial in high-performance or racing scenarios.

For Miatas, there are a range of upgraded trigger wheels available, most with 35 or 34 teeth referred to as a 36-1T or 36-2T trigger wheel. The triggers nominally have 36 teeth and either one or two teeth are removed so the crank position sensor will receive a signal when the crank has completed a full rotation.

The trigger wheel recommended for BMM ECUs is the 1999-2001 Mazda Protege 36-1 Trigger Wheel (part # ZM0111408). This trigger wheel is compatible with the stock crankshaft pulley and a Fluiddampr crankshaft pulley. If using an ATI damper, you must purchase a compatible trigger wheel.



Installation

Before completing this installation it is recommended to have the car running on a BMM ECU with the stock trigger wheel. This way, only a few parameters in the tune need to be changed to get the car working with this part.

Disconnect the Battery

Ensure the engine is off and disconnect the battery for safety.

Remove Engine Belts

Remove the engine belts obstructing access to the crankshaft pulley. Typically, there are two belts: the accessory belt which drives the power steering and A/C and the alternator belt which also drives the water pump. The belt tensioners are respectively located on the power steering pump (top right on the engine from the front) and the alternator (bottom left). With the belts removed, you should have clear access to the crankshaft pulley.

Rotate Engine to Top Dead Center (TDC)

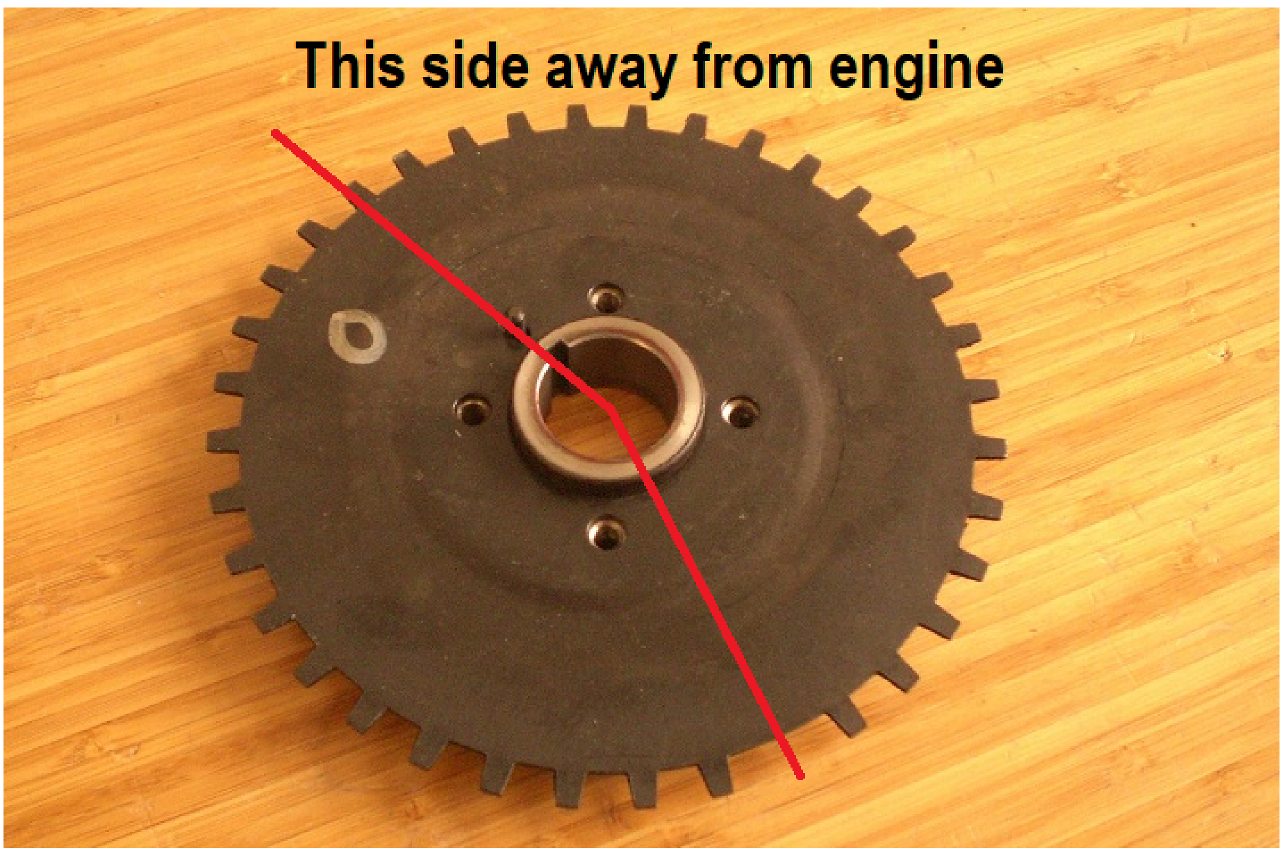
Using a 21mm socket on the bolt in the centre of the crank pulley, rotate the engine clockwise until the marks on the crank pulley line up with the timing marks on the timing belt cover.

Remove the Crankshaft Pulley

Remove the crankshaft pulley by undoing the four 10mm bolts on the front. The large centre bolt doesn't need to be removed if using the stock or a Fluiddampr pulley. An ATI damper will require removal of the crankshaft bolt to install so consult their installation manual if required. Behind the pulley should be the OEM trigger wheel. Go ahead and remove it, put it in the bin or use it as a beer coaster.

Install the New Trigger Wheel

The new trigger wheel can go on to the engine in two orientations, correctly and backwards. To radially align the trigger wheel, there is a dowel pin on the crank which slots into a hole in the trigger wheel. At TDC, the dowel pin should be vertical or at the 12 o'clock position. The correct orientation is to put the trigger wheel on, as per the diagrams below, with the centre recessed part of the trigger pointing towards the crank, the white dot facing outwards and the missing tooth at approximately the 7 o'clock position when the engine is at TDC and the dowel pin is vertical. The trigger wheel is the wrong way around if the teeth are further forwards of the engine than the centre recces, the white dot is facing towards the engine, or the missing tooth is at the 5 o'clock position at engine TDC.



If installing an ATI damper and trigger wheel or a Fluiddampr, consult the manual as the installation has several additional steps which include bolting the trigger wheel to the aftermarket crankshaft pulley damper. In the case of a Fluiddampr, the orientation of the Mazda 323 trigger wheel will be identical to installation onto a stock pulley.

Re-Install Crank Pulley and Adjust Crank Sensor

Re-install the crank pulley and the four 10mm bolts (109-151 inch/lbs 13-17 Nm). Check the clearance between the crank sensor and the tip of a tooth on the timing wheel, there should be a 0.5-1.5 mm or 0.020-0.059 inch gap between the tooth and the sensor. If you later come up with trigger errors, this gap may need to be reduced. To reduce the gap, loosen the 10mm bolt holding the crank sensor and wiggle it to the desired position.

Re-Install Belts and Connect Battery

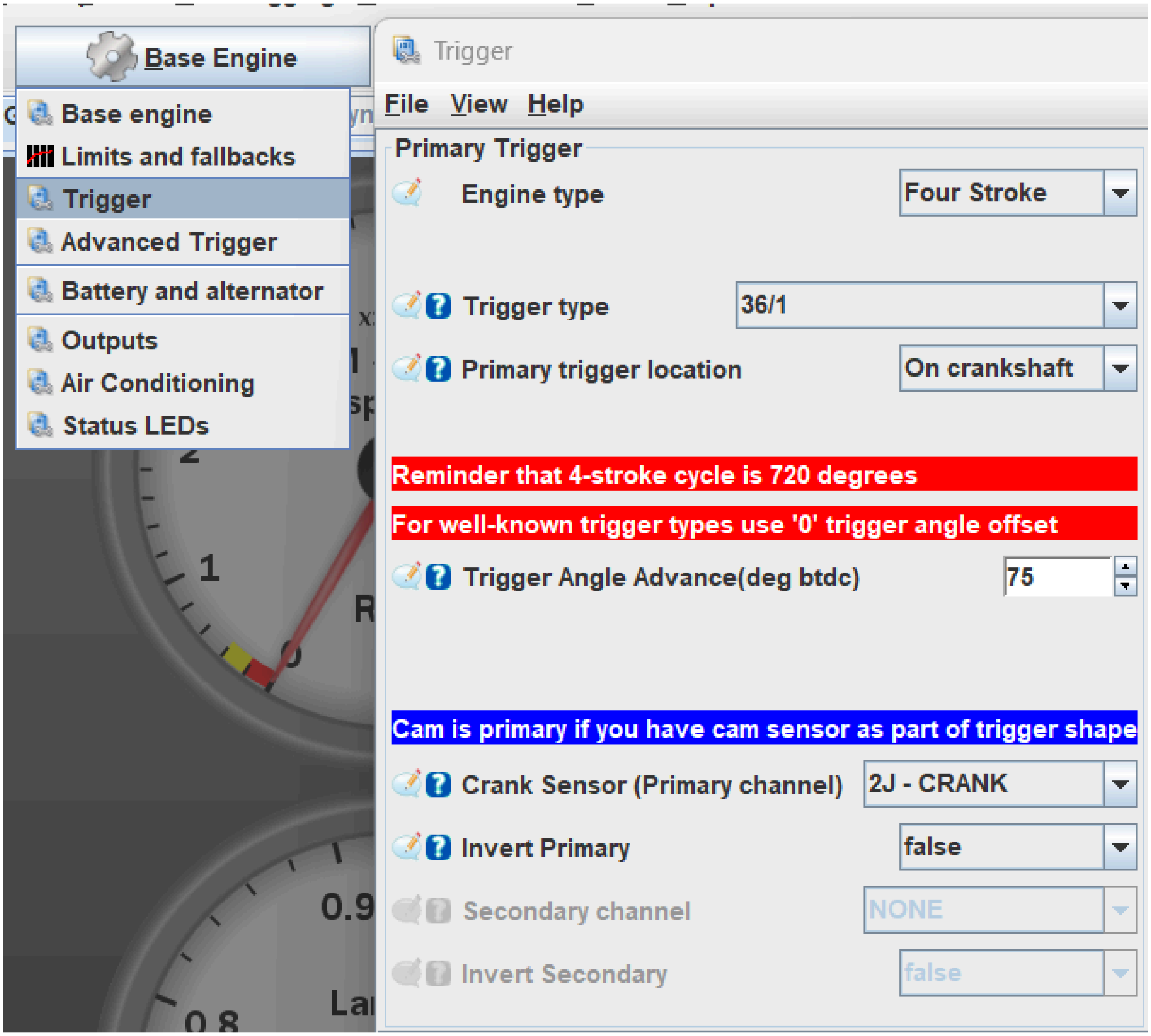
Re-install the accessory and alternator belts checking they are suitably tight. Now connect the battery back to the car. The mechanical installation is now complete and it's time to boot up the computer.

TunerStudio Settings

In the "Trigger" menu under the Base Engine tab, change the trigger type to 36/1 and the trigger angle advance to 75 degrees.

If using a 36-2 trigger wheel instead, the first setting would be 36/2 and the advance will likely be different so check with the manufacturer. In the case of a 36-2T trigger wheel for an ATI damper, the advance angle should be around 148 degrees.

These settings will likely get the car started with the Mazda 323 timing wheel however the trigger angle advance may need to be iterated to perfectly match the timing on your car. To check or set the base timing, set the timing to fixed in TunerStudio and use a timing light on the crank to measure the timing. If the timing is not 10 degrees BTDC, iterate the trigger angle advance in TS until it is spot on. This is covered in "Set Base Timing" within the Miata Quick Start Guide.



Now after ensuring your timing is reverted back to dynamic in TS, the installation should now be complete!

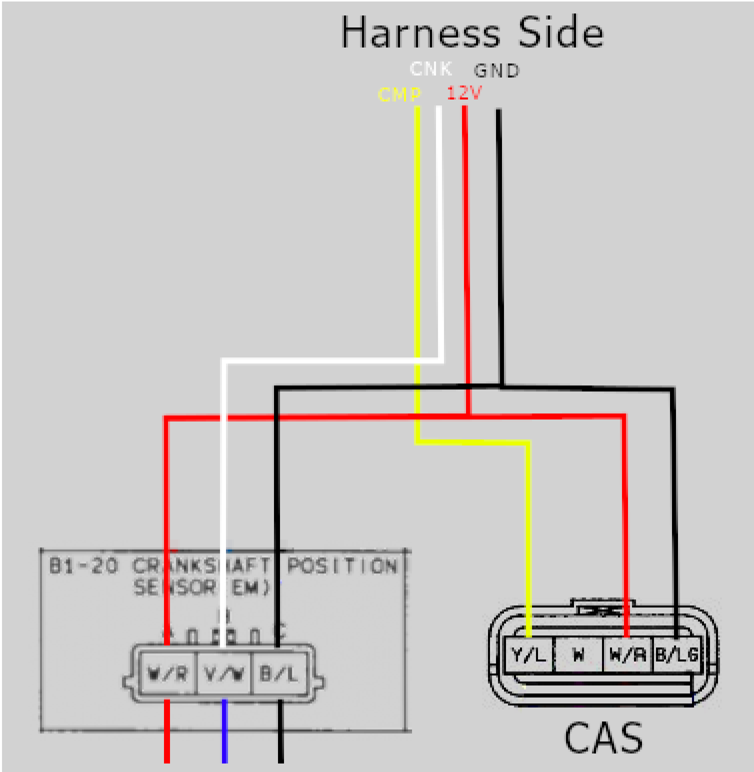
Additional Steps For NA Miatas

Crank Position Sensor

NAs either do not come with any crank position sensor, or have a VR sensor. In either case, you need to use a Hall effect CPS from a 1999-2005 Miata. The CPS will be bolted in place with an M5 or M6 bolt, and possibly some washers, in order to align properly with your trigger wheel. Ensure that the pigtail from the sensor is secured such that it does not get damaged by the nearby belts.

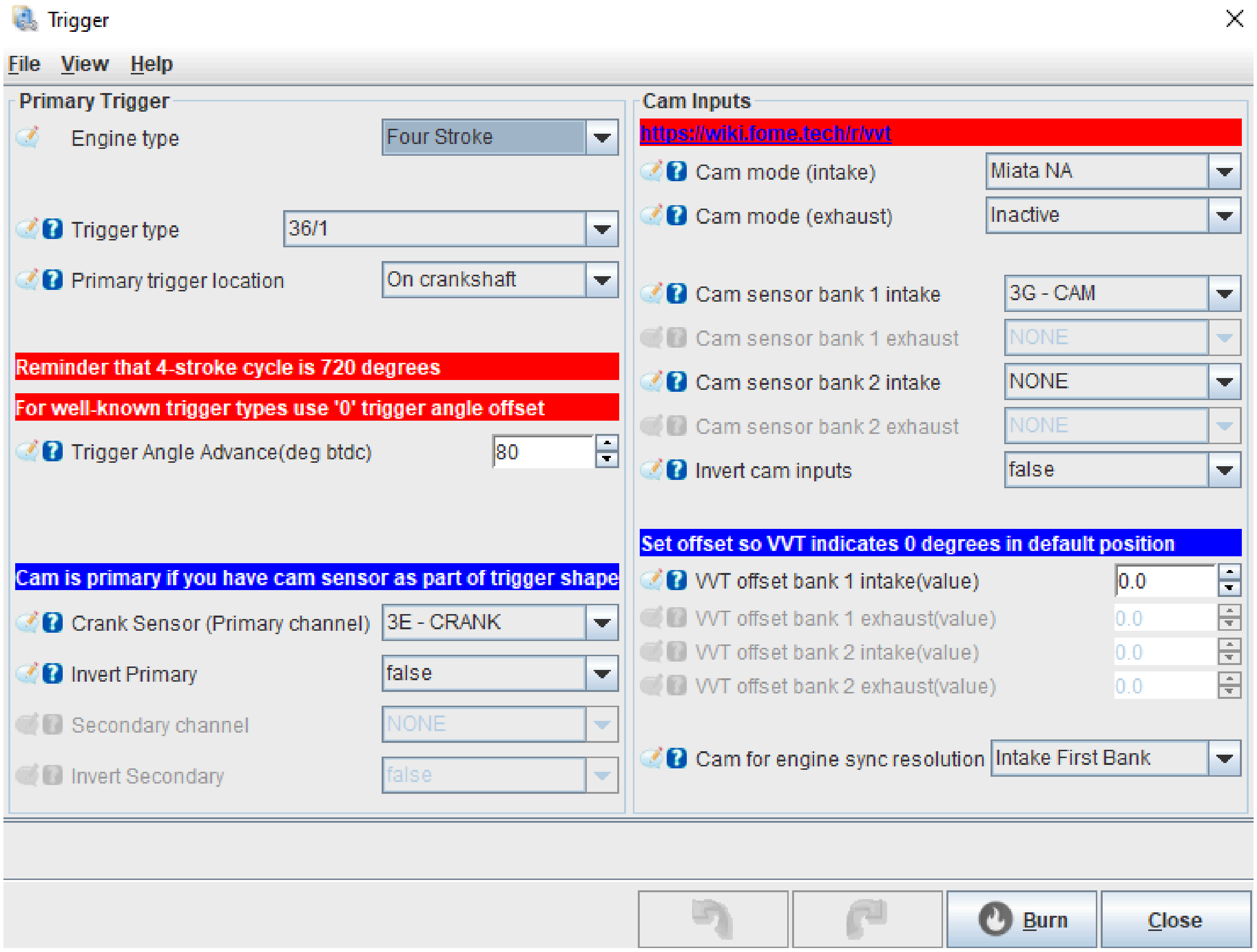
Wiring

No additional wires need to be run to the ECU, but the wiring will require slight modification. This can either be done by cutting the factory harness, or by making a making a sub-harness that plugs into the CAS receptacle. Either way, you'll need to match the connectivity shown in the diagram below. Both sensors should receive power and ground, CMP should only connect to the CAS, CNK should only connect to the new crank position sensor.



Tunserstudio Settings

In order to enable sequential fuel or ignition, the cam input settings pictured here should be used. Regardless if you have a 1.6 or 1.8, Cam Mode (intake) should be set to Miata NA, and Cam Mode (exhaust) should be inactive. This requires a firmware release from after October 1, 2024


Trigger
File View Help
Primary Trigger
Engine type
Four Stroke
Trigger type
36/1
Primary trigger location
On crankshaft
Reminder that 4-stroke cycle is 720 degrees
For well-known trigger types use '0' trigger angle offset
Trigger Angle Advance(deg btdc)
80
Cam is primary if you have cam sensor as part of trigger shape
Crank Sensor (Primary channel)
3E - CRANK
Invert Primary
false
Secondary channel
NONE
Invert Secondary
false
Cam Inputs
https://wiki.fome.tech/r/vvt
Cam mode (intake)
Miata NA
Cam mode (exhaust)
Inactive
Cam sensor bank 1 intake
3G - CAM
Cam sensor bank 1 exhaust
NONE
Cam sensor bank 2 intake
NONE
Cam sensor bank 2 exhaust
NONE
Invert cam inputs
false
Set offset so VVT indicates 0 degrees in default position
VVT offset bank 1 intake(value)
0.0
VVT offset bank 1 exhaust(value)
0.0
VVT offset bank 2 intake(value)
0.0
VVT offset bank 2 exhaust(value)
0.0
Cam for engine sync resolution
Intake First Bank
Burn
Close

Miata Coil On Plug Conversion

Converting your Miata to run a coil on plug (COP) ignition setup is a great engine upgrade for those with forced induction or high-performance engines. Offering a stronger spark and improved reliability, coil-on-plug technology outperforms the coil packs of the 1990s and early 2000s Miatas. It also eliminates the need for ignition wire replacements and reduces electrical interference, making it a more robust and appealing option in automotive applications.

This guide will discuss the steps to convert your stock NA or NB Miata ignition system to COP.

Parts Required

Ignition Coils and Connectors

Toyota and Audi/VW ignition coils are the most common. For Toyota coils, ones off a 1ZZ engine will work as well as many other variants. Some part numbers for Toyota coils are: 90080-19015, 90919-02239, 90080-19023, 90919-02234. The part number for the coil connector is: 90980-11885.

For Audi coils, the ones used are referred to as R8 coils as they are used on the Audi R8 (as well as most other Audi's and some VWs). The most common part number is: 06E905115G however there are also many variants that will work depending which brand of "R8 coil" you want and how much you want to spend. The part number for the R8 coil connector is: 1J0973724.

Both of these ignition coils work and both have their distinct advantages. The R8 coils securely clip onto the spark plugs however they protrude further from the spark plug holes than stock so usually require a form of spacer to keep them from wiggling around. The Toyota ones sit flusher however they also require a bracket for their mounting bolt to secure them in place. For the NA6 and earlier NA8 cars, the tachometer takes a signal from the ignition coils, only the Toyota coils have this pin. You can run R8 coils on these cars however your tachometer will not work unless you wire it directly to the ECU.

Car Wiring Harness Connectors

If you wish to solder directly to the harness, you can ignore this however those wishing for a more elegant solution prefer to clip their COP harness onto the existing vehicle harness.

For a 1.6L, this is more difficult and it is either recommended to buy a spare igniter to take apart or simply wire directly into the harness. For the other NA and NB Miatas utilizing a coil pack on the back of the valve cover, a compatible connector part number is: 6098-0144.

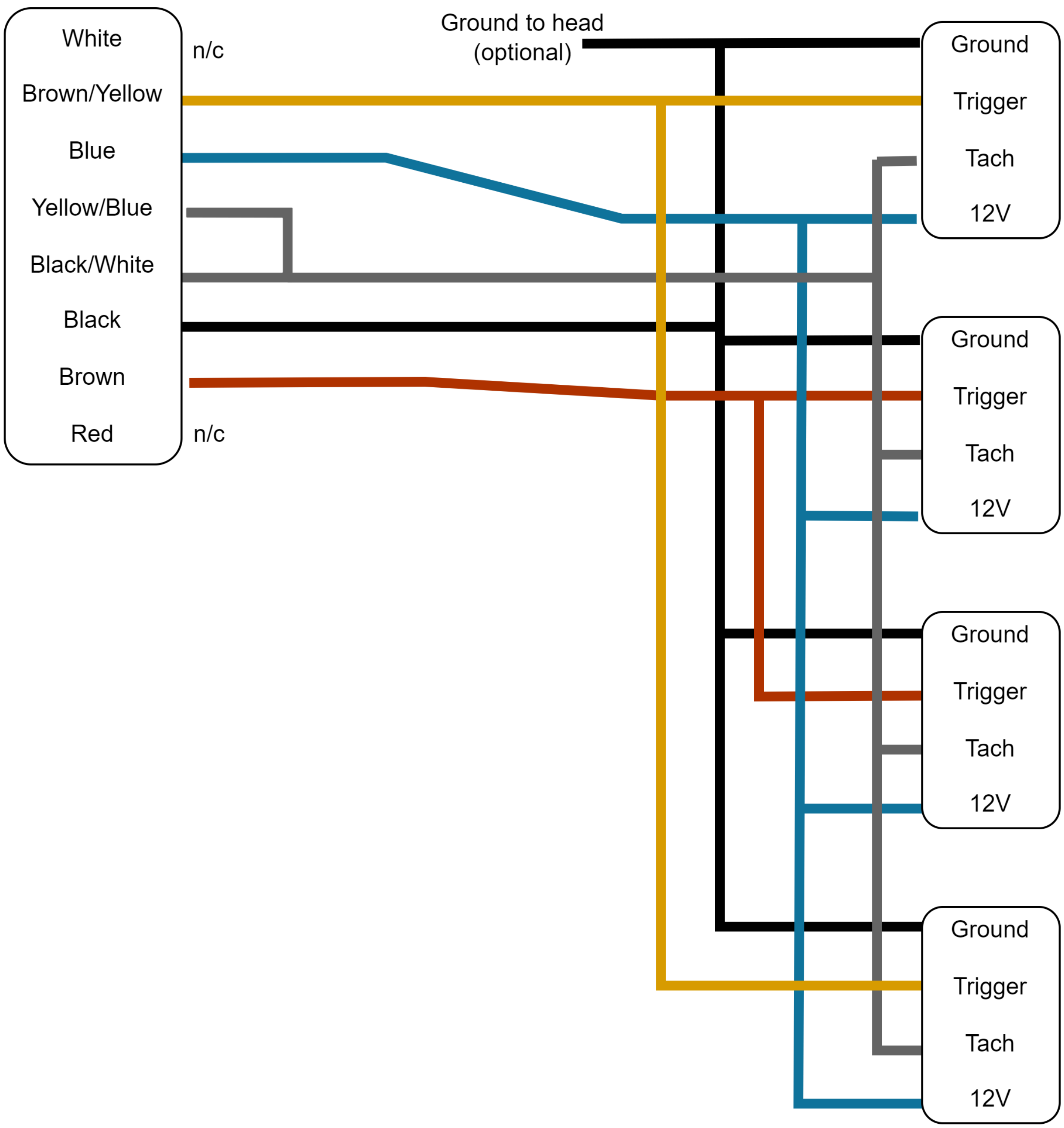
Other Components

- 10,000uF capacitor to be installed between the 12V and GND which helps to filter electrical noise however this is not strictly necessary.
- 18-22 AWG wire to help reduce the internal resistance and carry the

power to drive the coils.

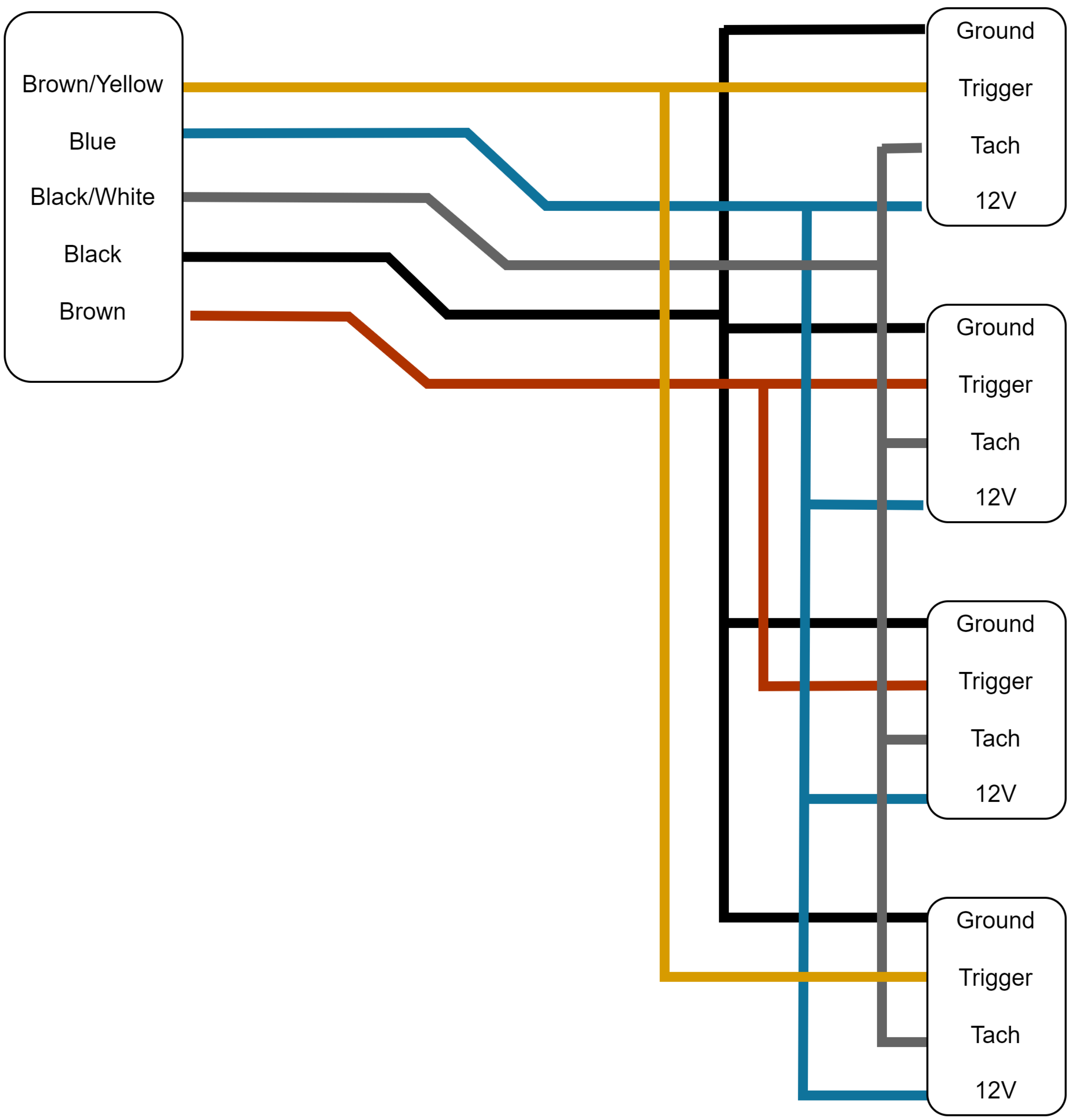
- Wiring harness or electrical tape to wrap over the harness
- Crimping tool
- Soldering equipment
- Heat shrink
- Wire strippers
- Wiring
- The wiring diagrams for each model NA/NB is shown below.

NA6 (1.6L)



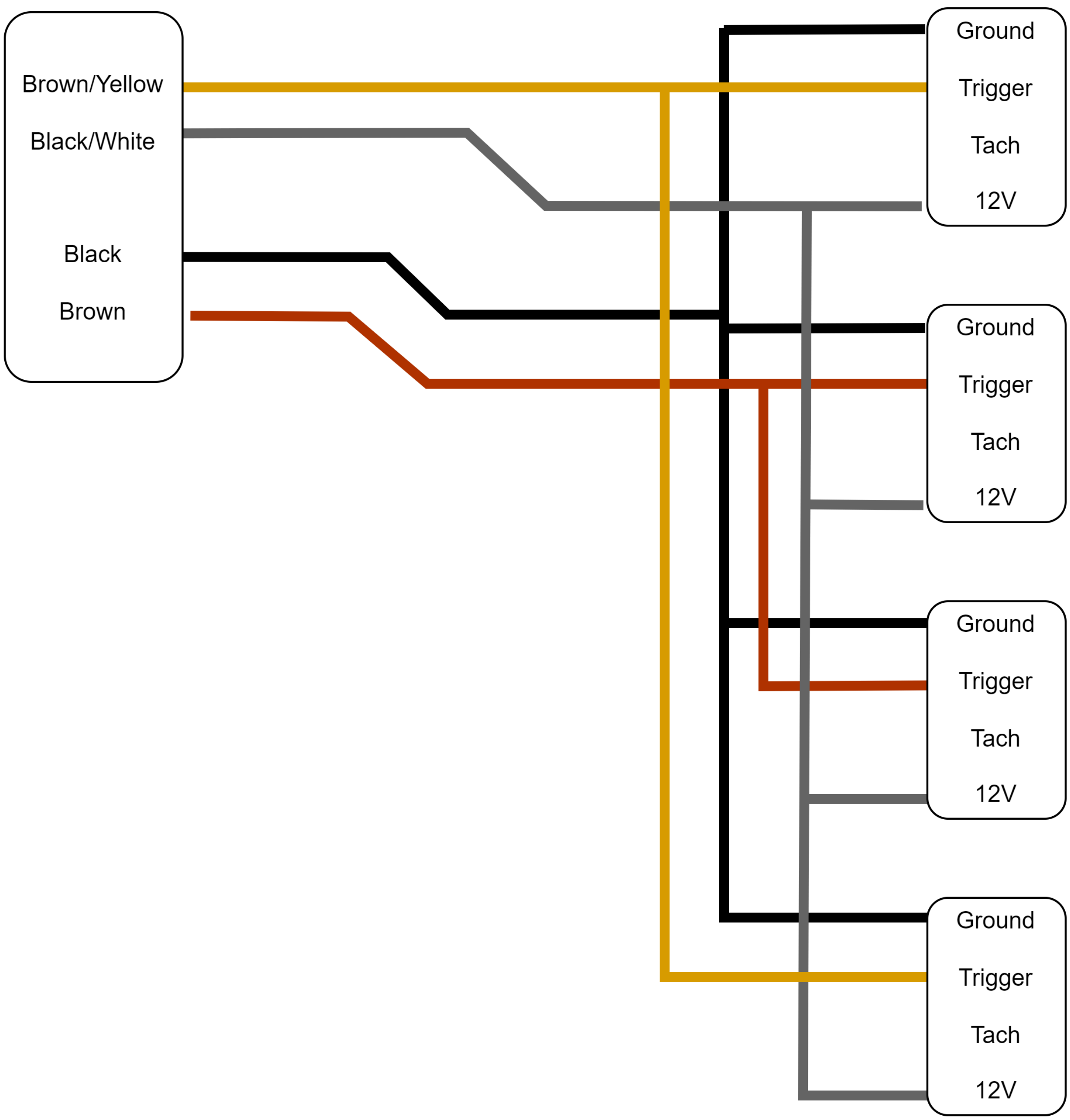
White
Brown/Yellow
Blue
Yellow/Blue
Black/White
Black
Brown
Red
n/c
n/c
Ground to head
(optional)
Ground
Trigger
Tach
12V
Ground
Trigger
Tach
12V
Ground
Trigger
Tach
12V
Ground
Trigger
Tach
12V

NA8 94-95 (1.8L)



Brown/Yellow
Blue
Black/White
Black
Brown
Ground
Trigger
Tach
12V
Ground
Trigger
Tach
12V
Ground
Trigger
Tach
12V
Ground
Trigger
Tach
12V

NA8 95+ /NB1 (1.8L)



Toyota Coil Pinout

From left to right on the coil is the ground, ECU signal, tachometer signal, and 12V. For later NA8 and NB1 Miatas, the tachometer signal does not need to be wired up.

R8 Coil Pinout

From left to right on the coil is the ground, ECU signal, ground, and 12V. The two grounds can be connected together. Note that the R8 coils have no tachometer signal meaning that on NA6 and early NA8 cars, the tachometer will need to get its signal from the ECU. Later NA8s and NB1s do not have this issue.

COP Harness

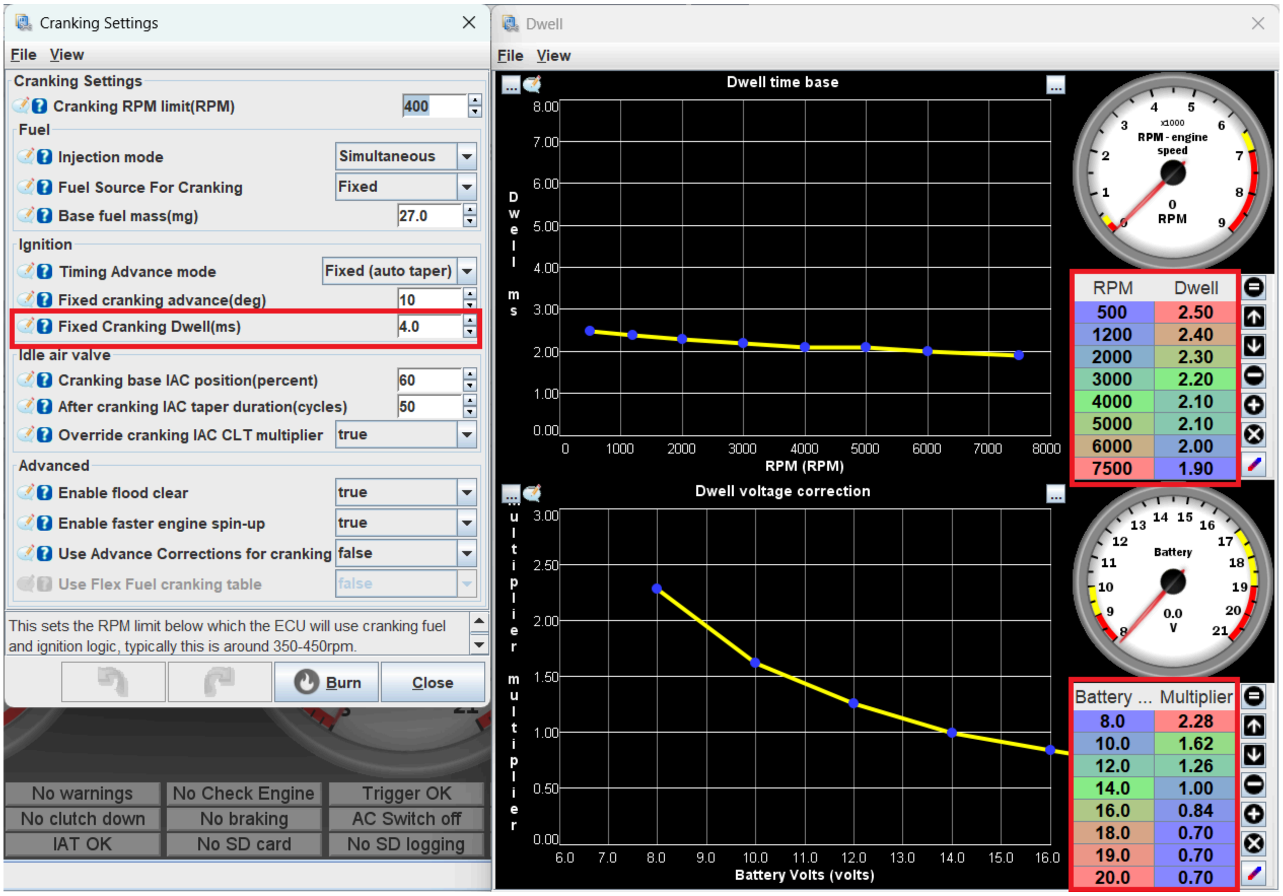
To make the neatest harness, cut the wires to length with the coils installed in the car in their desired orientations. Leave a small amount of excess so there is no tension on the wiring harness and room in case a wire needs to be re-stripped. Make sure to cover each exposed solder joint with electrical tape or heat shrink then go over the entire harness with tape to protect it. It is recommended to check your harness wiring several times for shorts or mistakes before powering on the car.

Coil Mounts

For both coils, there are a range of third-party suppliers selling mounting brackets. If you wish to DIY a mount, they can be made relatively inexpensively using aluminum or 3D printed spacers (ABS or other high temperature plastics are recommended).

ECU Dwell Time Setup

The dwell times need to be modified in TunerStudio before starting the car with the new COPs. Recommended safe dwell settings are shown below for both Toyota and R8 COPs. Note that dwell times can vary significantly between different ignition coils and it is recommended to do some additional research on your specific coil. The longer the dwell time, the longer the coil charges for and the stronger the spark. Too little dwell will result in a weak spark and too much dwell can draw an excessive amount of current, possibly melting the coil.


Cranking Settings
File View
Cranking Settings
Cranking RPM limit(RPM) 400
Fuel
Injection mode Simultaneous
Fuel Source For Cranking Fixed
Base fuel mass(mg) 27.0
Ignition
Timing Advance mode Fixed (auto taper)
Fixed cranking advance(deg) 10
Fixed Cranking Dwell(ms) 4.0
Idle air valve
Cranking base IAC position(percent) 60
After cranking IAC taper duration(cycles) 50
Override cranking IAC CLT multiplier true
Advanced
Enable flood clear true
Enable faster engine spin-up true
Use Advance Corrections for cranking false
Use Flex Fuel cranking table false
This sets the RPM limit below which the ECU will use cranking fuel and ignition logic, typically this is around 350-450rpm.
Burn
Close
No warnings
No Check Engine
Trigger OK
No clutch down
No braking
AC Switch off
IAT OK
No SD card
No SD logging
Dwell
File View
Dwell time base
Dwell ms
RPM (RPM)
RPM - engine speed
RPM Dwell
500 2.50
1200 2.40
2000 2.30
3000 2.20
4000 2.10
5000 2.10
6000 2.00
7500 1.90
Dwell voltage correction
multiplier multiplier
Battery Volts (volts)
Battery
Battery ... Multiplier
8.0 2.28
10.0 1.62
12.0 1.26
14.0 1.00
16.0 0.84
18.0 0.70
19.0 0.70
20.0 0.70

Miata Knock Sensor Upgrade

Installation of Knock sensor

For **NA6 & NA8** there is a M10x1.25 threaded blind hole that should be utilized for a knock sensor located above the Factory Oil Pressure sending unit below the intake manifold. You can access this with the intake manifold installed.

There are two steps required to install a **Bosch KS4-P** sensor onto a 90-97 Miata block.

1. **Create clearance for the Knock Sensor**
    - Utilizing the factory Mazda oil pressure sender unit
        - Install a 1/8 BPST spacer (male to female)
        - Relocate using Oil sender remote mount kit
    - Replace OEM sender unit with a Pressure Transducer
        - Adapt from 1/8 BPST to Required fitting type (make sure to account for the spacing of the knock sensor)
2. **Mounting the knock sensor to the block and torquing it to 20nm.**
    - Given the engine's M10x1.25 specs, options include employing a M10x1.25 threadsert with a M8 X 1.25 Internal Thread, crafting a

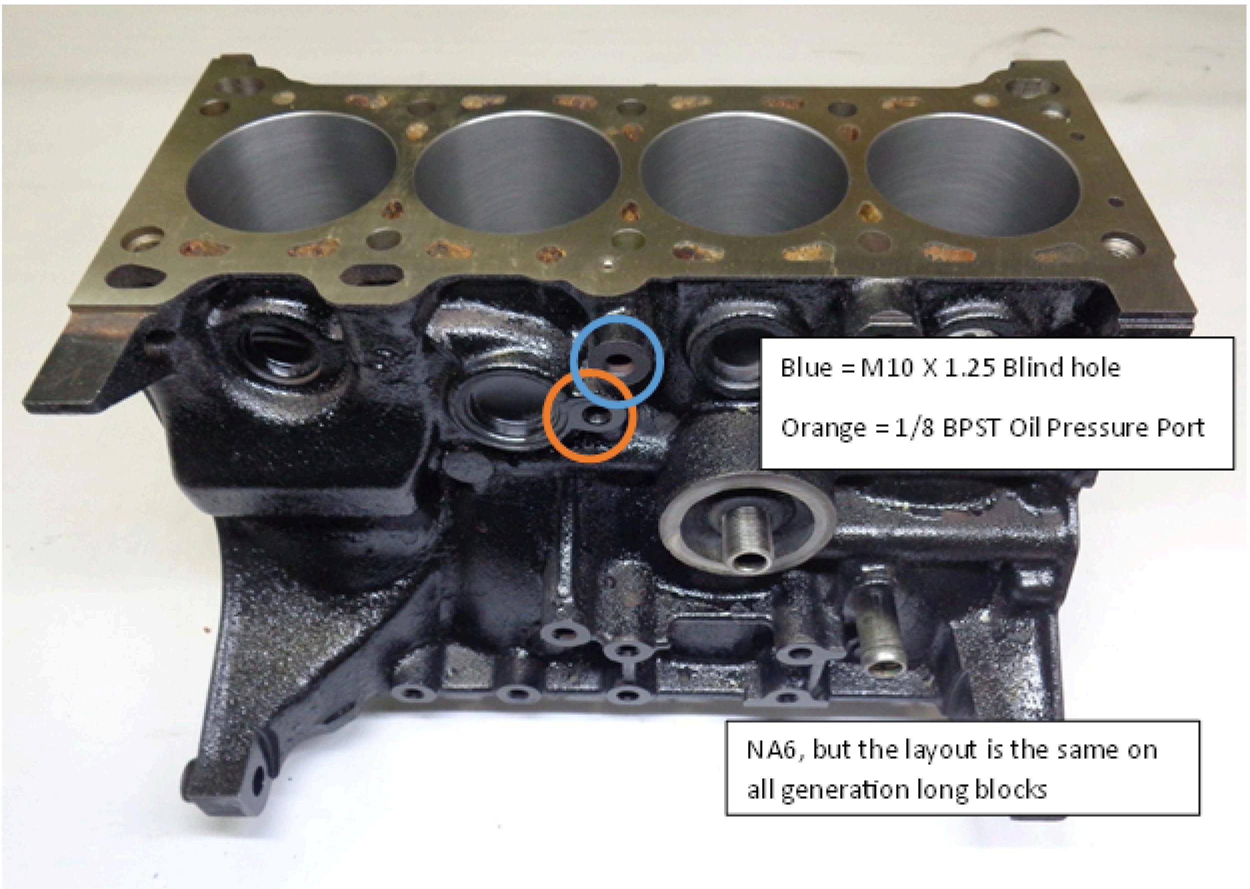
custom M10x1.25 to M8x1.25 threaded stud, or using a m8x1.25 Allen Bolt and using a Die to cut m10x1.25 threads onto the external section of the bolt head.

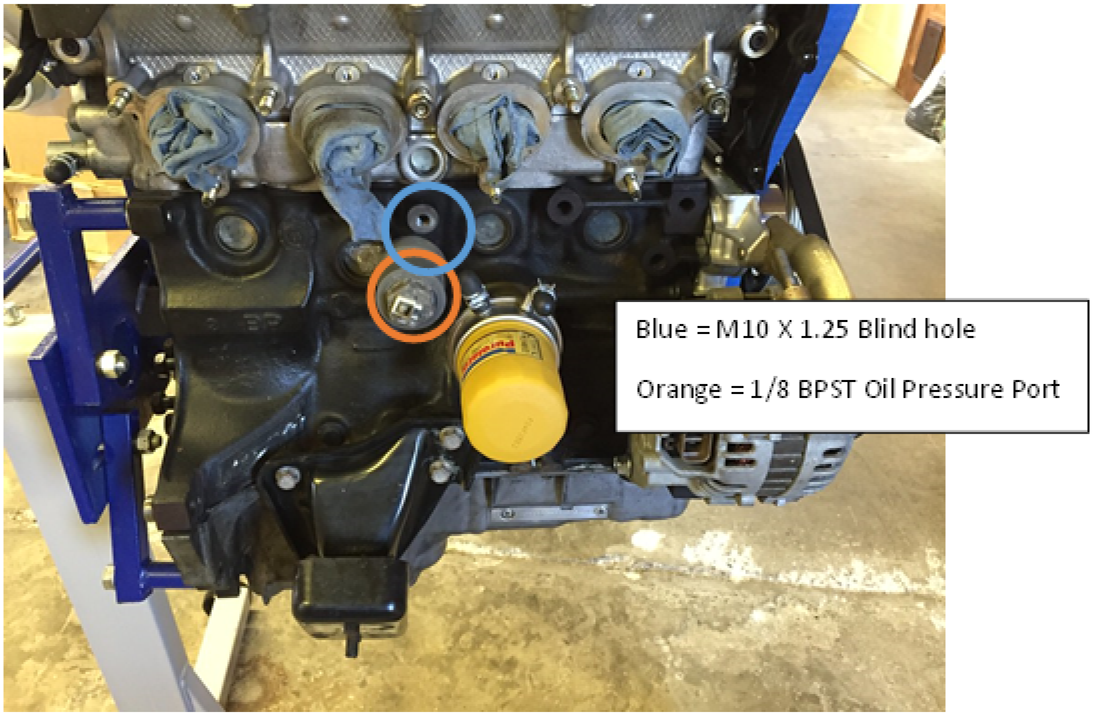
For **NB models**, a factory knock sensor is pre-installed on the block. However, replacing it with a Bosch KS4-P knock sensor can offer a broader frequency bandwidth for knock detection. To make the switch, similar steps apply, necessitating the use of a M10x1.25 threadsert with a M8 X 1.25 Internal Thread, a custom M10x1.25 to M8x1.25 threaded stud, or a M8x1.25 allen nolt with threads cut using a die onto the bolt head's external section. Once the Knock Sensor is torqued to the block, wiring it into the correct pinout on the ECU is the subsequent step.

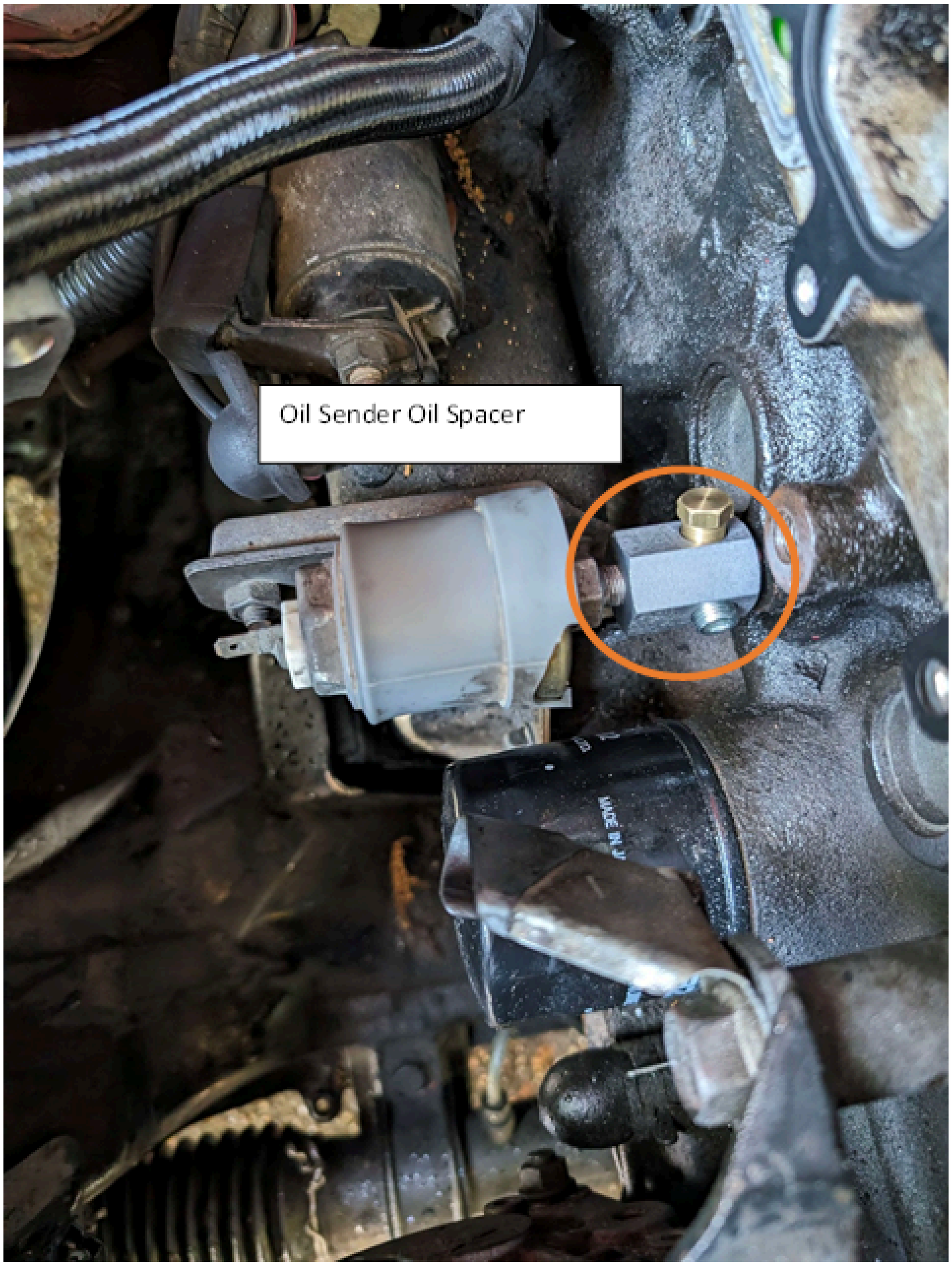
Wiring might entail additional pins and connectors depending on the Miata's model year. Below are the pinouts for the current rev of ecu’s:

- 90-95 – 2P
- 96-97 – 1S
- NB1 – 1S
- NB2 – 4M

Overview of the Block




Blue = M10 X 1.25 Blind hole
Orange = 1/8 BPST Oil Pressure Port


Oil Sender Oil Spacer


Example of M10 to M8
Stud


Oil Sender Oil Spacer &
Knock Sensor

Tuner studio Setup

When knock occurs due to factors like advanced ignition timing or high cylinder pressures, the sensor detects these vibrations. The ECU then adjusts ignition timing to prevent further knock. This constant monitoring and adjustment help protect against damage. To set this up requires the following steps:

Summary of knock detection and protection:

1. Enable the knock sensor and calculate the knock frequency using an approximation formula.
2. Remove ignition timing from the ignition map, adjust other parameters to prevent pre-detonation events.
3. Record a log of engine performance and knock sensor levels across full RPM range.
4. Restore the tune and maps to its original configuration.
5. Review log data to create a baseline curve for knock detection.
6. Adjust knock retard aggression to determine the level of response.
7. Set up the Max Knock Retard table to define maximum allowable ignition timing retardation.

> **WARNING**
>
> The following steps assumes your ECU is running the latest release of FOME (2312)

1. **Enable the knock sensor**, and calculate the estimated knock filter

frequency (kHz). For now, configure the first setting “cylinder bore” as 0.00mm.

- An adequate approximation formula for Knock Frequency is “Knock Frequency = 900,000/(π * 0.5 * cylinder bore diameter)”
- Once the approximated knock frequency is calculated, use the second-order harmonics of the estimated frequency. The second-order frequency are multiples of the original calculated frequency. So twice the Knock Frequency. We do this to increase sensitivity, improve signal to noise ratio, and general “robustness” in frequency analysis.
- For now, Set knock detection window start to 0.00, This feature is for advanced users only.
- This formula is derived from the relationship between the speed of sound, the bore diameter, and the frequency of knock waves. It assumes that the speed of sound is approximately constant and that the knock waves travel at a specific angle through the combustion chamber. As an example, the NA6’s estimated knock frequency is 7300 Hz or 7.3kHz & its 2nd order harmonic would be 14.60Khz.

> **DANGER**
>
> The next few steps assume your car is running well enough to take a low-load full rpm log to define the engine knock threshold curve.

> **INFO**
>
> An engine knock threshold curve shows how the sensitivity of knock sensors changes with engine speed. It's a graph where the horizontal axis represents engine speed (in RPM) and the vertical axis shows the knock sensor's sensitivity level (in dBV). Tuning this curve ensures the ECU reacts appropriately to protect the engine while maximizing performance.

2. **Start by removing ignition timing** from the ignition map, an approach is to remove 3 degrees of timing, increase octane rating, decrease boost and any other parameters that can contribute to pre-detonation events.

3. **Record a log at a minimum of 100 Samples per second from idle to redline** (Under Communications/Data Rate). This can be performed in a few ways. The better the Data, the better the Threshold curve:

    - On Jack stands with slight load applied such as brakes
    - Driving on the road in a controlled fashion

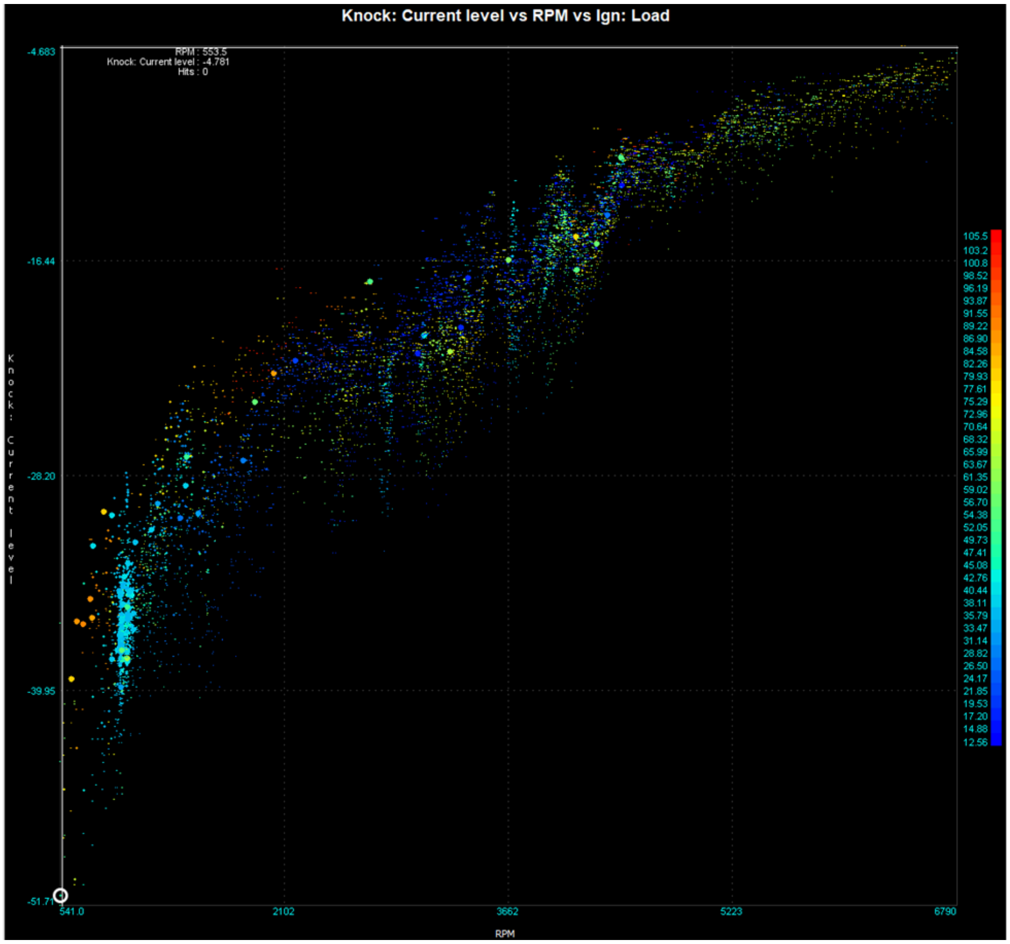
4. **Restore the tune back to the previous configuration** (for example, add 3 degrees of ignition timing).

5. **Review the log in MegaLog Viewer and generate a scatter plot** of the “Knock: Current Level” vs RPM:

    - Ideally the plot is the low-load noise of the engine throughout the whole rpm range. It should look something like this below.
        - Further filtering can be applied in megalog viewer to remove

high manifold pressure and deceleration noise. Use these expressions to help analysis the measured data.

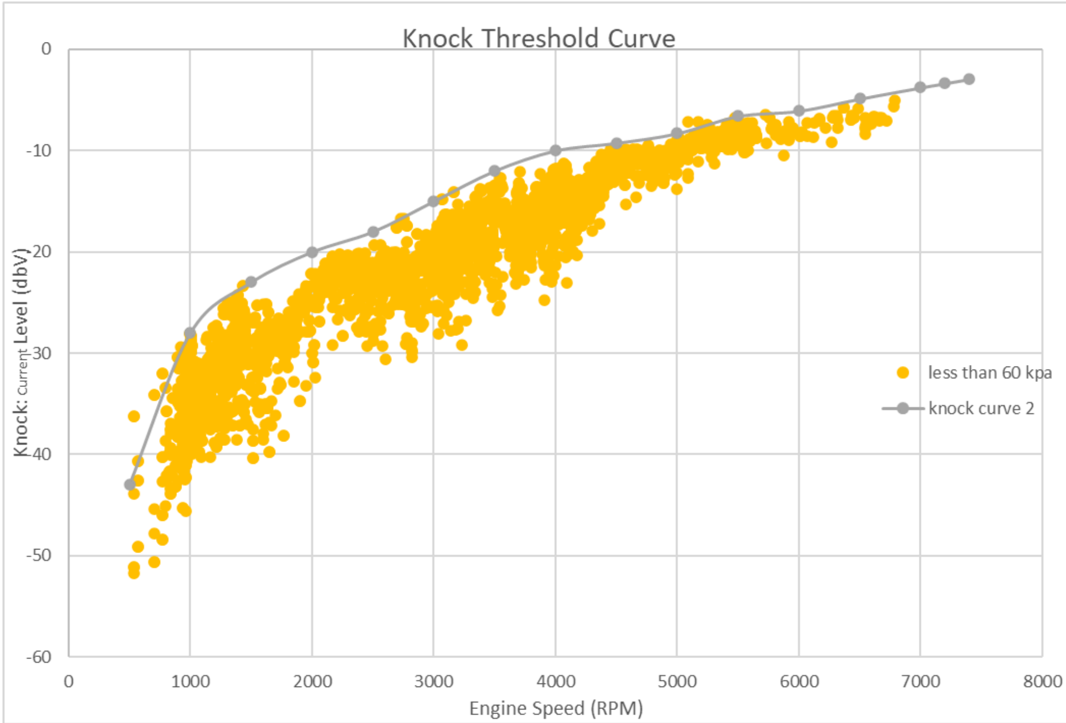
- Deceleration : “ [RPM-4]<=[Field.RPM]&&[TPS]<50 “
- High Load : “ [Ign: Load]>60 “

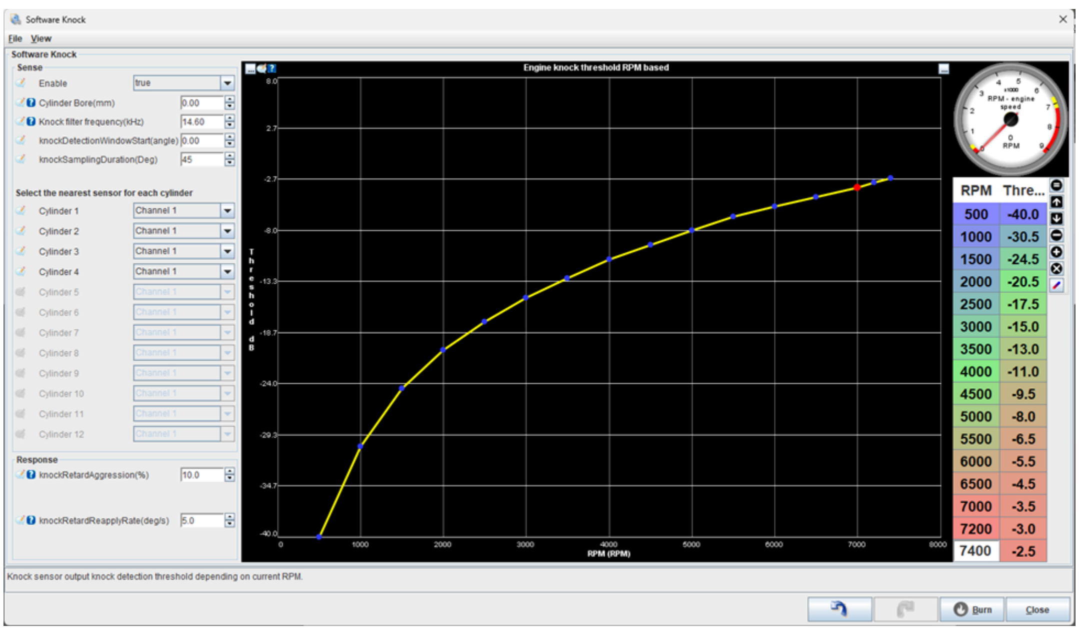


- Once the above plot for your engine has been generated, we can use this data to create a plot that will be used in Tunerstudio’s table on the bottom right. This is a baseline curve, further logs can be taken and more data reduction can used to refine the curve.
- The curve used in Tunerstudio should fit over the top of the low-load noise and also be “tight” to the measured data. As shown below, the orange scater is all the data less than 60kpa and the

grey is a curve can be used as a baseline threshold in Tunerstudio.

- An active system is better than one missing low level knocks





6. **Set up the Response of the Knock controller's parameter "Knock retard aggression"** A generalized rule of thumb would be

5% is considered adequate where as 15% being very aggressive:

- The knock retardation amount is determined by calculating the distortion from the optimal ignition timing, multiplying it by the configured knock retard aggression percentage to determine the desired retardation, and then applying this retardation to the current knock retardation value.

7. **Set up the Max Knock Retard table**:

    - The max knock table defines the maximum allowable knock values that the knock controller can use to retard ignition timing, with the Y-axis representing ignition load, the X-axis engine speed in RPM, and the Z-axis indicating the degree of timing retardation permitted for each combination of load and speed.

90	0.00	8.00	8.00	8.00	12.00	8.00
70	5.00	10.00	12.00	12.00	12.00	10.00
50	10.00	10.00	15.00	15.00	15.00	15.00
40	10.00	15.00	15.00	15.00	15.00	15.00
30	10.00	15.00	15.00	15.00	15.00	15.00
20	5.00	10.00	15.00	15.00	15.00	15.00
⮡	1000	2000	3000	4000	5200	6000

Conclusion

Installing a knock sensor in a Mazda Miata involves accessing a threaded hole above the oil pressure sending unit under the intake manifold. For models NB and onwards, a factory-installed knock sensor is available, but upgrading to a Bosch KS4-P sensor broadens the frequency bandwidth for better knock detection. Installation requires creating clearance for the sensor and mounting it to the block, followed by wiring it into the correct pinout on the ECU. Once installed, the knock sensor continuously monitors knock events, allowing the ECU to adjust ignition timing and prevent damage from subsequent knock events. Setting up knock detection involves calculating the knock frequency, recording engine performance logs, and adjusting parameters such as knock retard aggression and the Max Knock Retard table to optimize engine protection and performance.

Knock Sensor Part list

Knock	**Bosch - Off the Shelf**	
Bosch KS4-P	0 261 231 173	FCP Euro 2-Pin RB-Kp.1 (F02U.B00.966-01) or 0 261 231 188 Ballenger 2-Pin Jetronic (D261.205.288-01) [LK-2 Connector]
Bosch KS4-R	0 261 231 218	eBay 2-Pin RB-Kp.1 or 0 261 231 223 eBay 2-Pin RB-Kp.3 (F02U.B00.967-01)
Bosch KS-1-S	0 261 231 120	FCP Euro Amazon 2-Pin RB-Kp.1 (F02U.B00.966-01) or 2-Pin Jetronic (D261.205.288-01) [LK-2 Connector]

Miata VVT Setup and Tuning

Tuning the NB2 BP-Z3/BP-VE VVT motor can be a daunting task. This guide will detail the process for configuring and tuning the VVT for an NB2 VVT motor using a BMM ECU.

Required Equipment

- Vehicle running an NB2 VVT motor.
- Access to a dyno or a safe location to tune the vehicle using Virtual Dyno. An actual dyno is preferable to virtual dyno due to the accuracy of readings and ability to safely vary the operating point of the engine.
- Laptop with TunerStudio, MegaLog Viewer and optionally Virtual Dyno installed.

Initial TunerStudio Configuration

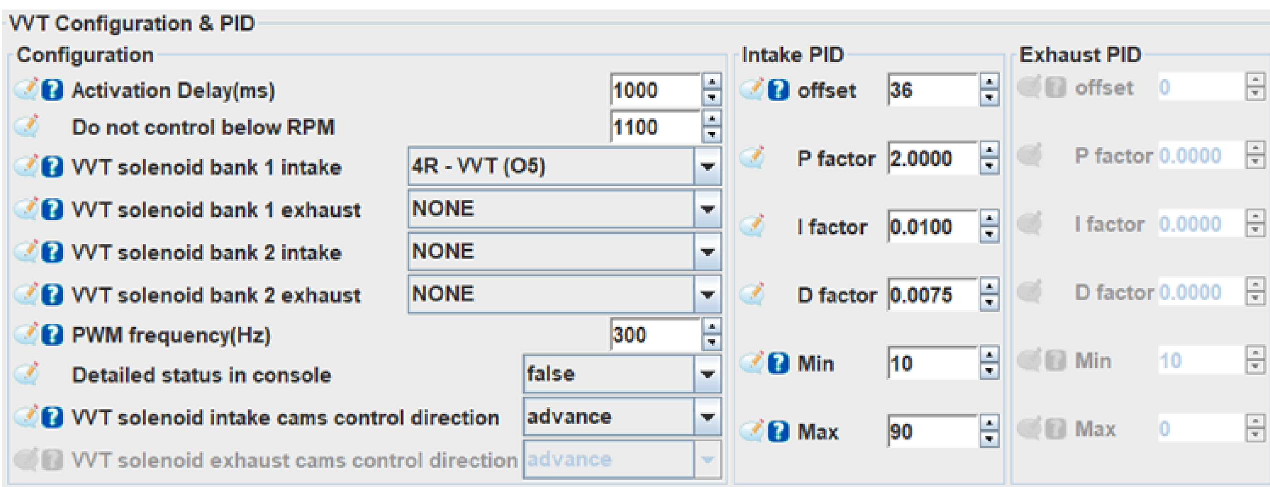
Open TunerStudio and open the "VVT Configuration and PID" menu under the advanced tab. Going down the settings in this menu first is the activation delay. This is the initial delay when the car is started before the VVT is enabled. A setting of 1000-5000ms (1-5s) should be suitable here to give the engine enough time to build oil pressure before enabling the VVT.

The "Do not control below RPM" setting or activation RPM disables VVT until the engine RPM exceeds this value. Ideally set it about at 100-200 RPM or so above your idle RPM to keep it disabled during idle. This will simplify the idle tuning as the engine torque from the VVT changing can lead to a fluctuating idle.

For the VVT solenoid banks, an NB2 motor only has VVT on the intake so select the pin labelled as "VVT" for the "VVT solenoid bank 1 intake" setting.

On the right side of the menu are the PID control options. If you are new to PID tuning, plenty of guides exist detailing how it works. This video by RCModelReviews details how PIDs work on a basic level.

Pictured below are some reasonable default settings however it is still recommended to configure the settings yourself as settings can vary from car to car.

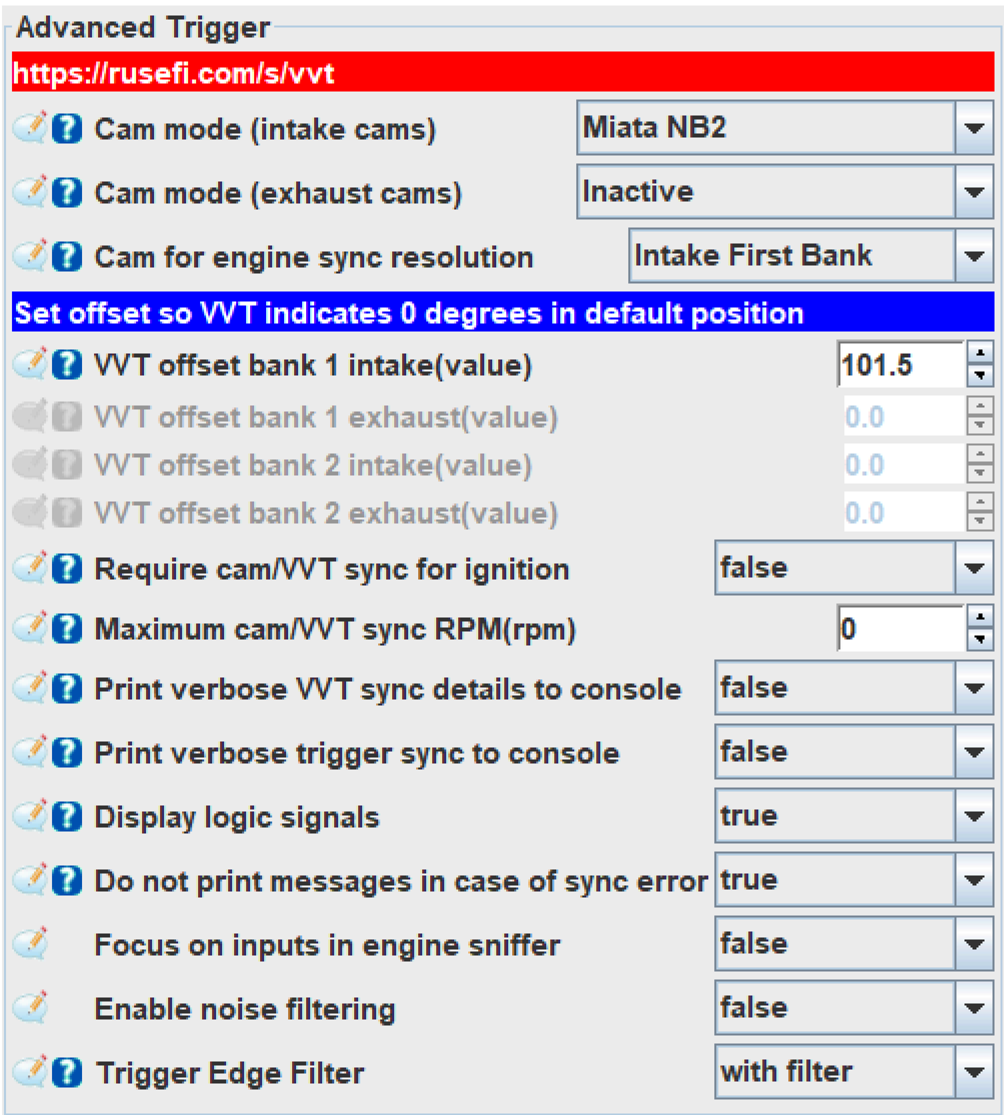


VVT Offset Setting

Before configuring the VVT PID, the basic VVT angle offset needs to be configured. This is to calibrate what the ECU thinks is 0 degrees VVT angle to the actual VVT angle on the car, similar to setting the base timing. **This is different to the offset in the PID control menu**. Open the "Trigger" menu under the "Base Engine" tab and locate the "VVT offset bank 1 intake(value)" setting. In the engine bay, unplug the VVT solenoid highlighted in the image below:

Right click one of the gauges on the background of TunerStudio and under VVT, change it to "VVT bank 1 intake vvtPositionB1IGauge". This will show a live reading of the detected VVT position on the car.

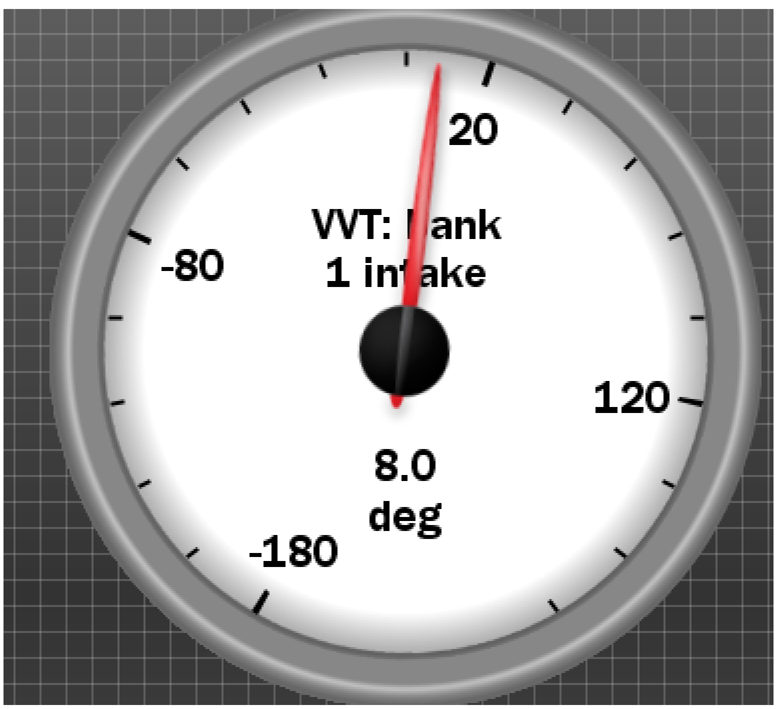
Start the car with the solenoid unplugged and read the VVT position on the gauge. Change the "VVT offset bank 1 intake(value)" setting until the VVT position reads 0. The VVT angle is now calibrated and you can shut off the car and plug the solenoid back in. Your offset should be similar to that shown below:



PID Tuning VVT Settings

Under the PID settings the offset is the VVT solenoid duty cycle offset. Basically, the solenoid will only activate if it is pulsed above a threshold duty cycle and the PID controller needs to know what duty cycle this occurs. PID tuning can be a tedious process and it is recommended to outsource it to a professional if you are not confident.

To tune the offset, open the "VVT Closed Loop Target" table and set every cell to a constant value such as 10 degrees. Set the P gain to 1. Right click one of the gauges on the background of TS and under VVT, change it to "VVT bank 1 intake - vvtPositionB1IGauge". This will show a live reading of the detected VVT position on the car.

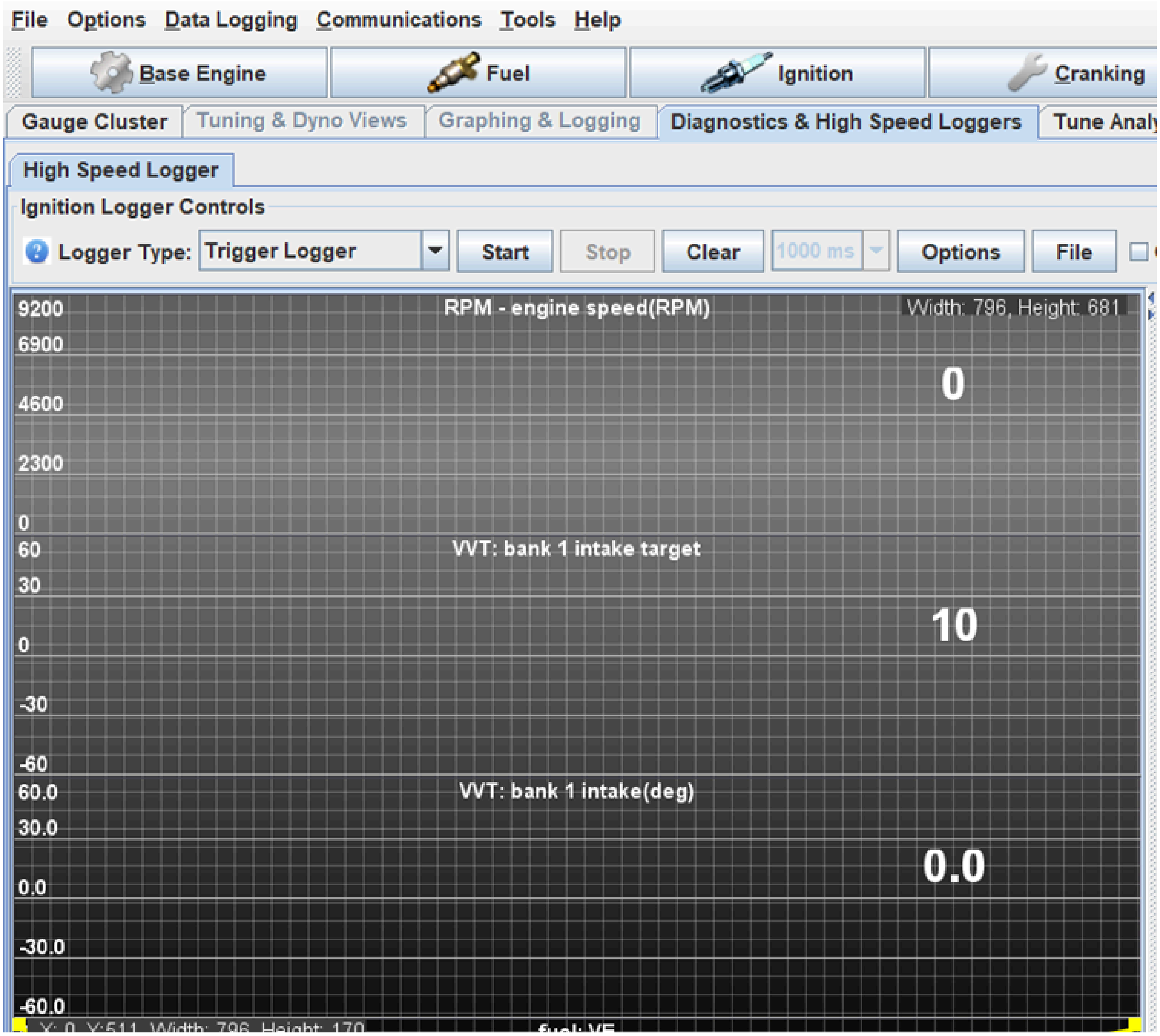


Start with an offset value of e.g. 10. Start the car and hold the engine RPM at a constant value above the threshold RPM and observe the VVT position gauge. If the gauge does not change, increase the offset by 5-10 and repeat until you find the minimum offset required for the gauge to change VVT position. Note that the actual VVT position won't be correct yet as the PID controller still needs to be set up. A value in the range of 30-40 is expected.

Next is the PID tuning. This step can take a while and is important to do thoroughly to ensure that the VVT can reach its target angles quickly and without overshooting significantly. The PID tuning is best done on a dyno or with a mate to drive the car whilst you tune it. Alternatively you can take a driving datalog, analyze it in MegaLogViewer, revise the PID settings and take another driving log. This method would be very time consuming and the first two are recommended.

To tune the P, I, and D settings you can either start from scratch or work from other users values to refine them. Below are some PID values you can work from if you wish. The process of tuning the PIDs is to put various angle targets in the VVT intake target table at different engine loads and RPM and to move the engine around between them. As the VVT target changes, the PID loop will activate to attempt to reach the new target. By watching the response of how quickly and with how much overshoot the controller reaches the target, the PID controller can be tuned.

An example on the dyno this might look like having all VVT target cells below 1500 RPM at 0 and above that at 20 degrees. In the "diagnostics and high speed loggers" tab you can set one graph to be the VVT position and the other as the VVT target as shown below:



Next, rev and hold the engine up to e.g. 2000 RPM and watch how quickly the VVT position rises to match the VVT target. If the position overshoots, decrease P or increase D, vice versa if it undershoots. If the position slowly drifts from 20 degrees, you would increase the I gain. The PID settings would be varied until the VVT position quickly rises or falls to and VVT target changes without much over or undershoot. To properly optimize the PID tune, it is recommended to repeat a similar tuning process with lots of VVT angles.

Once the VVT is PID tuned, the VVT target table can be tuned.

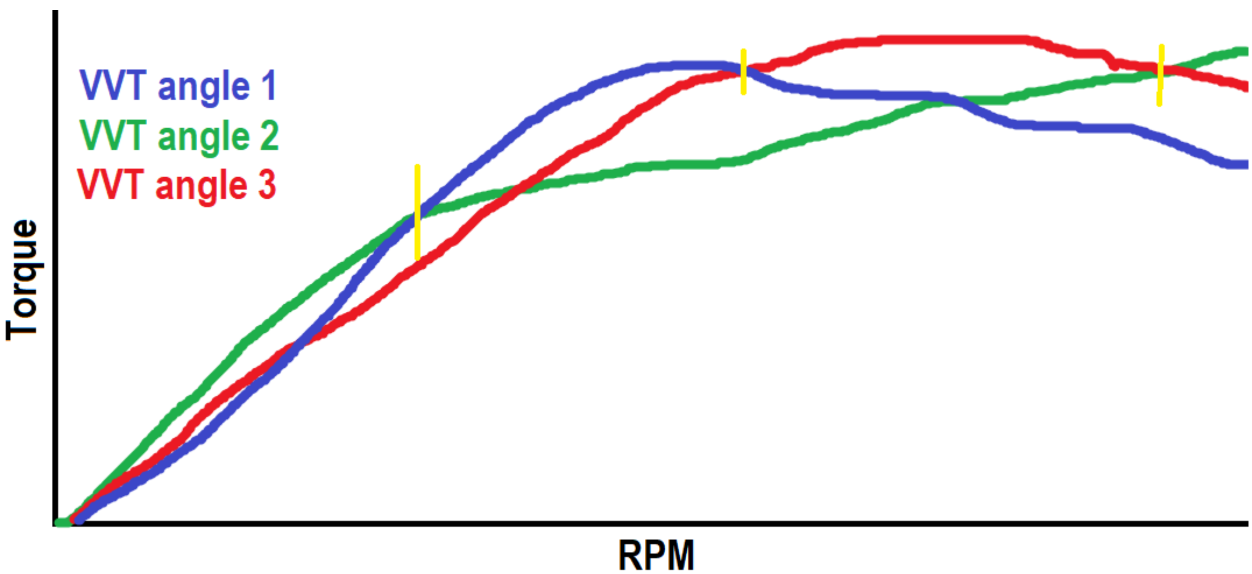
VVT Target Angle Tuning

Tuning the VVT angle is where the power gains are made. At this point the VVT should be fully set up and PID tuned to quickly reach the target position. If you cannot be bothered tuning the VVT angle, copying another users settings may yield decent results provided they have done the proper tuning however there will always be slight variances from car to car. A reasonable target table example is shown below:

fueling \ RPMValue	800	1500	2500	3500	4250	5000	6000	7200
100	8	15	25	35	35	15	0	0
75	8	15	25	30	30	15	0	0
60	8	15	20	25	25	15	0	0
40	8	12	18	20	20	10	0	0
25	8	12	18	20	20	10	0	0
10	0	0	15	20	20	10	0	0
5	0	0	10	10	10	10	0	0
0	0	0	0	0	0	0	0	0

The target angle is tuned in two sections, ramp run tuning and steady state tuning. Ramp runs are used to tune the VVT at maximum throttle and steady state tuning is used to tune the angles for partial throttle applications when the engine isn't fully loaded. This tuning is best done on a dyno for the best accuracy.

To tune the ramp run, set the entire VVT target table to 0 and perform a run. Save this run and do another, this time incrementing the whole VVT target table angle by 5-10 degrees depending how many dyno pulls you want to do. Repeat this process until the maximum VVT angle of 44 degrees is reached. Now load up all of the dyno runs and overlay them on top of each other. There will be points across these graphs where the torque from one VVT angle rises above the others. Power can be used as well but it is easier to do the tuning using the torque plots. For a given RPM range, take the VVT angle with the highest torque and put that into the maximum load section of the VVT target table. Repeat this for every RPM range where a different VVT angle yields a higher engine torque until the maximum load row of the table is populated. An example ramp run dyno graph is shown below where there are several "dyno runs" at different VVT angles. The yellow lines indicate the crossover points where a certain VVT angle has the most torque. Simply set the highest torque producing VVT angle in the chart for a given RPM range to that in the target table.



If you have some way of maintaining a constant throttle position below 100% such as a chock under the pedal (**on the dyno only!!!**) or limiting the maximum throttle body travel, ramp runs can be used to efficiently populate the whole VVT table for conditions where the engine isn't fully loaded.

To steady state tune on a dyno, the dyno will hold the engine at a specific RPM and load so you can vary the VVT angle in real time to see which angle has the highest torque. You then move to another cell and repeat this to tune the lower engine load sections of the VVT target table. To save time, it is possible go along a row tuning every second cell then interpolating between them. The downsides to steady state tuning is the higher load and RPM will quickly heat up the engine so it is crucial to monitor the temperature. This method also takes quite a lot of time but will yield precise results as the angle can be varied in small increments as the torque is measured instantaneously.

Road tuning with steady state is not possible as there is no way to measure the torque of the engine as the car sits at a specific load and RPM.

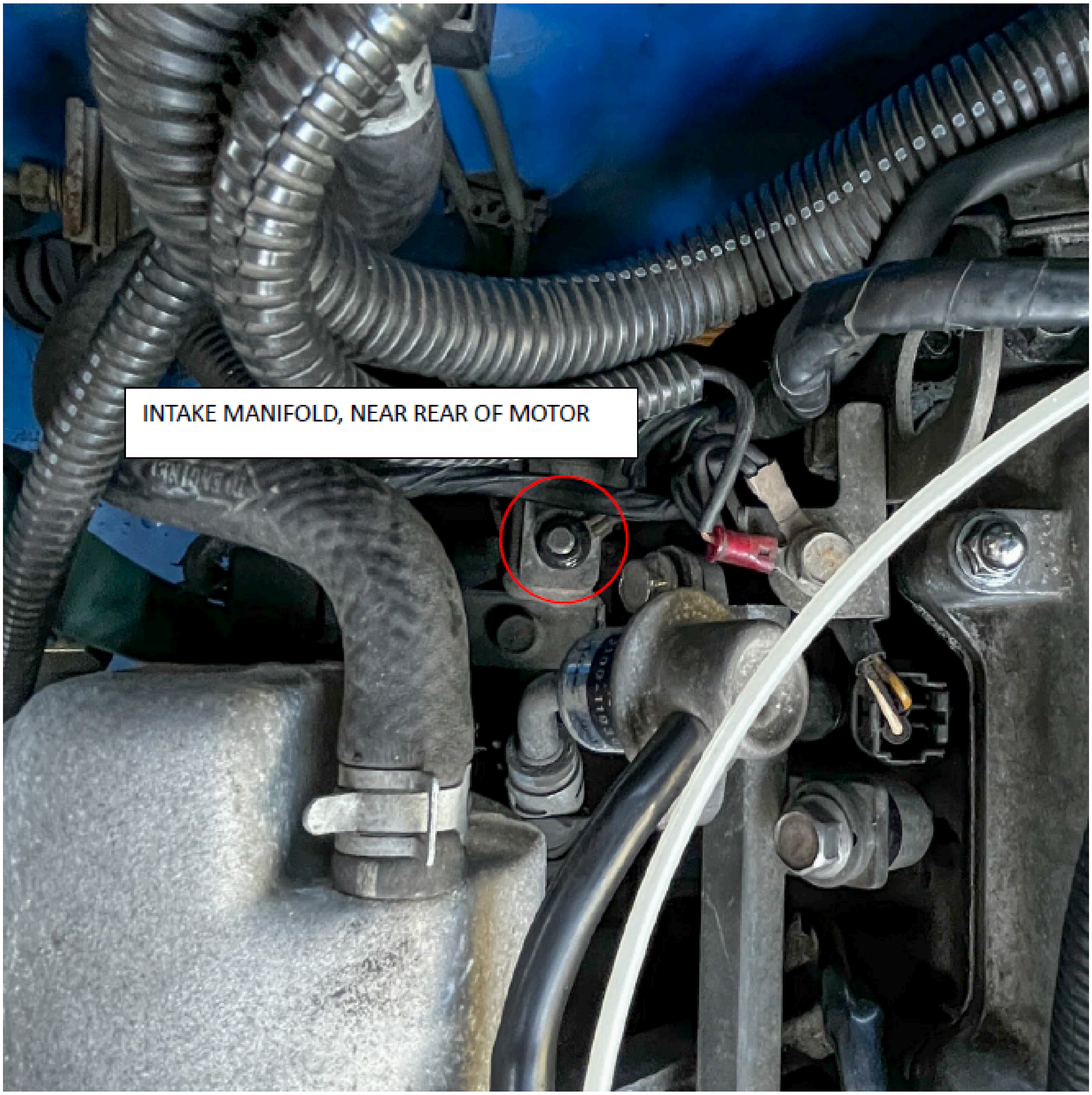
NA6 Miata ECU Grounding

When installing a standalone ECU on an NA6, there is one step that is often overlooked and not well-documented. This step is to ensure that the car sensors are grounded to the ECU instead of the chassis.

Grounding sensors directly to the ECU rather than the chassis provides numerous benefits including reduced electrical noise, improved signal integrity, and better sensor performance. Eliminating factory ECU ground connections to the chassis becomes particularly important when swapping to a standalone ECU with additional sensors, as it ensures a centralized grounding system and avoids potential conflicts in sensor readings.

Instructions

There is a stud coming off the intake manifold below the fuel rail, holding a P clamp for the wire harness with a connector to the Cam Angle Sensor. This is where a pair of grounds meet on a round ring terminal: one being a chassis ground (black), the other being an ECU ground (black and green). The black and green wire must be cut from this ring terminal and isolated. The location of this wire is shown in the images below:


INTAKE MANIFOLD, NEAR REAR OF MOTOR


BLACK AND GREEN GROUND WIRE
JAPAN

Note: If your car's ground is different due to modifications, follow the harness from the passenger side firewall to the cam angle sensor, and the pair of grounds may have been relocated to another location. See the NA6 wiring diagram at the end of this guide for further information.

Benefits

Sensor grounds play a critical role in ensuring accurate readings and proper functionality of various vehicle sensors. It is preferable to have sensor grounds connected to the ECU rather than the chassis. Eliminating a factory ECU ground connected to the chassis is important for several reasons:

- Noise and Interference Reduction: Grounding sensors directly to the ECU helps minimize electrical noise and interference that can affect sensor signals. The ECU acts as a central point for grounding, providing a cleaner and more stable electrical reference for sensor operation.
- Signal Integrity: Grounding sensors to the ECU ensures consistent and reliable signal integrity. By eliminating ground loops and potential differences between sensor grounds and the ECU ground, accuracy and precision in sensor readings are maintained, leading to more reliable engine management and diagnostics.
- Improved Sensor Performance: Sensors rely on stable ground connections to function optimally. By grounding sensors to the ECU, they benefit from a more controlled electrical environment, resulting in improved sensor performance, responsiveness, and accuracy.
- Preventing Ground Offset Issues: When sensors are grounded to the chassis, there can be voltage potential differences between the sensor ground and the ECU ground. This can lead to ground offset issues, where sensor readings are skewed or erratic due to voltage differentials. By grounding sensors directly to the ECU, these issues

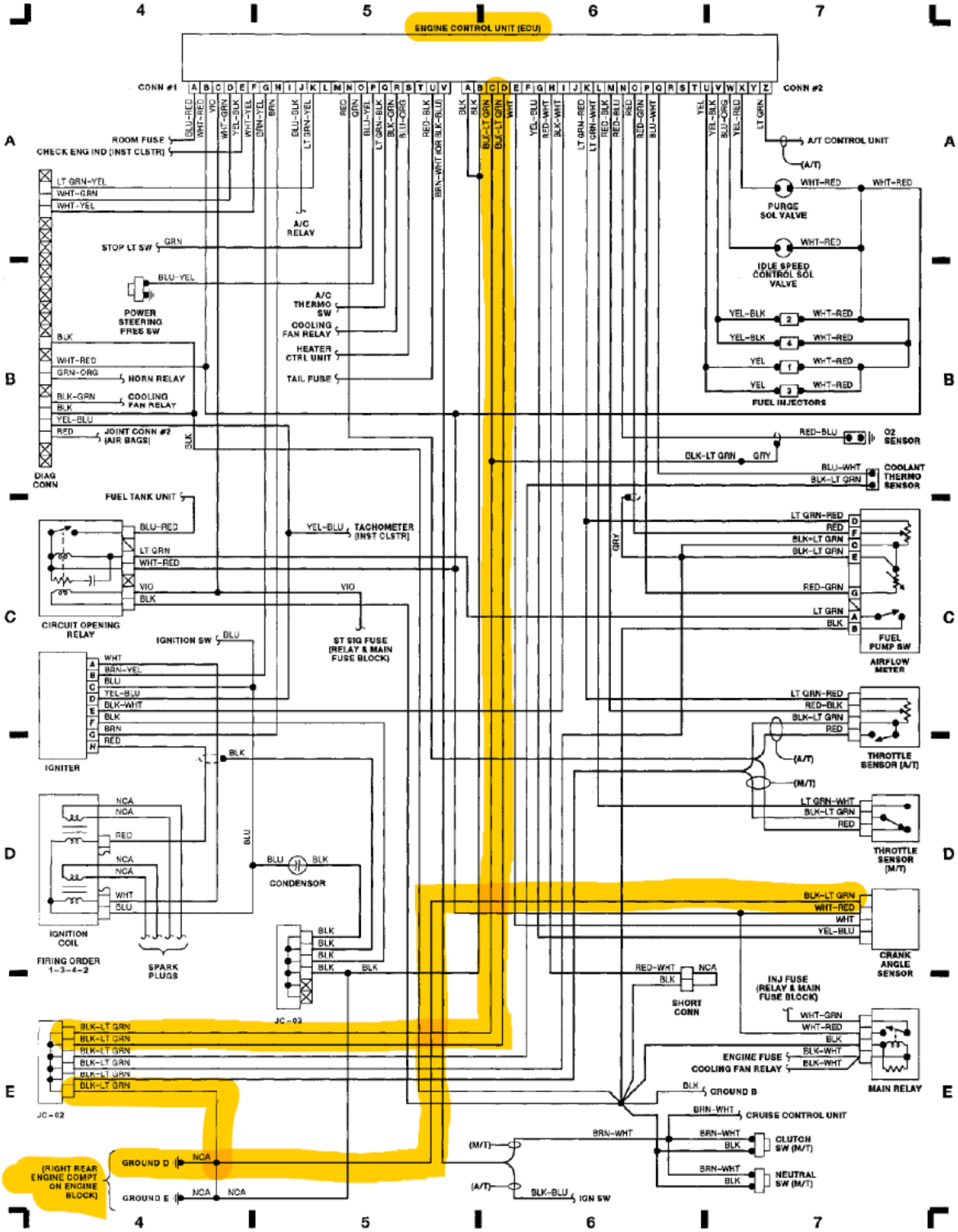
are minimized or eliminated. Maintaining Factory ECU Grounding: Factory ECU ground connections to the chassis should be eliminated if aftermarket modifications are made to the vehicle's electrical system. This ensures that sensor grounds are centralized and routed through the ECU, maintaining a consistent electrical reference and preventing potential conflicts or inconsistencies in sensor readings.

NA6 Wiring

If you are having difficulties tracing the correct wires to isolate, the wiring diagram below highlights the black and green ground wire and where it connects to on the car. By cutting the black and green wire from the engine block, you can see that the crank angle sensor is only wired to the ground on the ECU.

```
GROUND LOCATIONS TABLE
ÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄÄ
Component                                            Figure No. (Location)

GROUND A: (NEAR BATTERY)  .............................  1 (A 2)
GROUND B: (LEFT SIDE OF ENGINE
 COMPARTMENT, JC-01)  .................................  1 (E 3)
GROUND C: (RIGHT SIDE OF UNDERDASH NEAR BLOWER)  .......  1 (E 3)
GROUND D: (RIGHT REAR COMPARTMENT
 ON ENGINE BLOCK)  .................................  3 (E 9-10)
GROUND E: (RIGHT REAR ENGINE COMPARTMENT
 ON ENGINE BLOCK)  ....................................  2 (E 4)
GROUND F: (RIGHT SIDE OF UNDERDASH,
 ABOVE BLOWER)  ......................................  5 (B 17)
GROUND G: (LEFT SIDE OF UNDERDASH,
 ABOVE FUSE BLOCK, JC-04)  ...........................  5 (B 17)
GROUND H: (FAR RIGHT SIDE OF UNDERDASH)  ..............  5 (E 17)
GROUND I: (FAR LEFT SIDE OF UNDERDASH)  ...............  5 (E 17)
GROUND J: (REAR OF TRUNK, BETWEEN LICENSE LIGHTS)  ....  6 (E 22)
GROUND K: (RIGHR REAR TRUNK)  ........................  6 (C 22)
```


ENGINE CONTROL UNIT (ECU)
CONN #1
CONN #2
ROOM FUSE
CHECK ENG IND (INST CLSTR)
A/C RELAY
A/T CONTROL UNIT
(A/T)
PURGE SOL VALVE
IDLE SPEED CONTROL SOL VALVE
STOP LT SW
POWER STEERING PRES SW
A/C THERMO SW
COOLING FAN RELAY
HEATER CTRL UNIT
TAIL FUSE
HORN RELAY
COOLING FAN RELAY
JOINT CONN #2 (AIR BAGS)
FUEL INJECTORS
O2 SENSOR
COOLANT THERMO SENSOR
DIAG CONN
FUEL TANK UNIT
TACHOMETER (INST CLSTR)
CIRCUIT OPENING RELAY
ST SIG FUSE (RELAY & MAIN FUSE BLOCK)
IGNITION SW
FUEL PUMP SW
AIRFLOW METER
IGNITER
THROTTLE SENSOR (A/T)
(M/T)
THROTTLE SENSOR (M/T)
CONDENSOR
IGNITION COIL
FIRING ORDER 1-3-4-2
SPARK PLUGS
JC-03
CRANK ANGLE SENSOR
SHORT CONN
INJ FUSE (RELAY & MAIN FUSE BLOCK)
ENGINE FUSE
COOLING FAN RELAY
MAIN RELAY
JC-02
GROUND B
CRUISE CONTROL UNIT
CLUTCH SW (M/T)
NEUTRAL SW (M/T)
IGN SW
(RIGHT REAR ENGINE COMPT ON ENGINE BLOCK)
GROUND D
GROUND E

93G03450

How to add, change and test a page in the wiki

Prerequisites

- A free login in GitHub

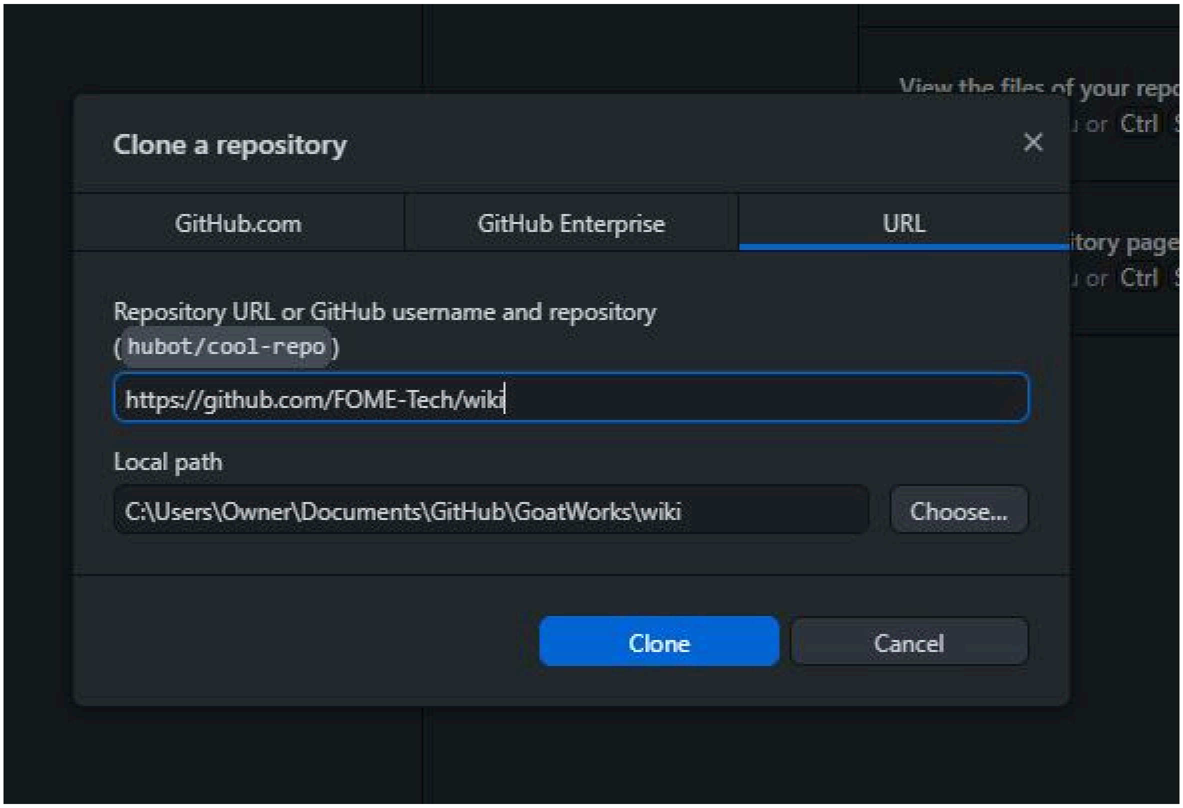
Simplest approach to make a small change

- go to end of page and click on "Edit this page" link
- follow GitHub's instructions for "Editing files in another user's repository".

Best way to edit or add a larger number of pages

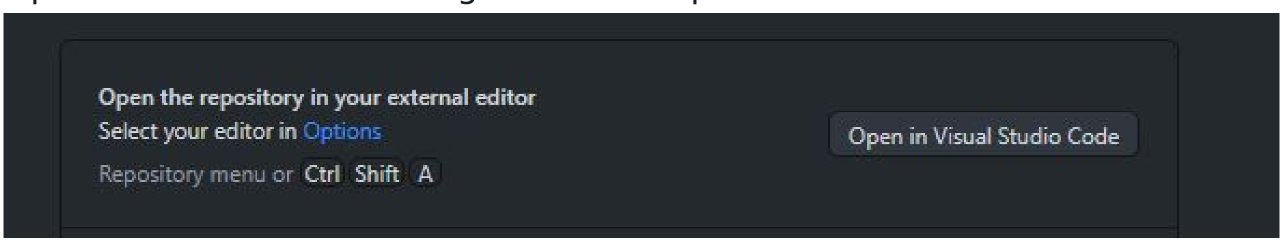
Get this: Github Desktop

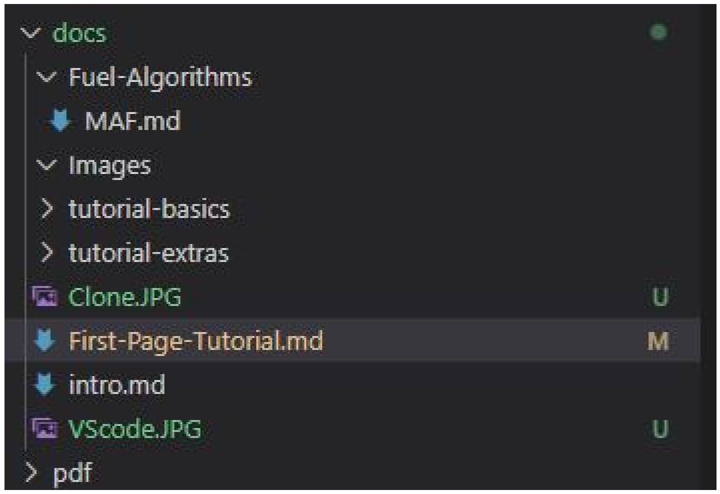
Clone the wiki repo Wiki Repo



Get VScode: VSCode

Open Wiki in VScode from github desktop



Start to edit the docs 

Pages are creates with a right click and named ##-xxxx.md, they will show up as pages in the wiki.

Folders are creatable with a right click, they will show up like this in the wiki 

To control ordering of topics in sidebar navigation the so called "numbered prefix" concept is used. This concept is further explained in [linked section of Docusaurus user manual]

To make images work the images need to be in the same folder as the .md, image names are also case sensitive so check capitals.

This markdown cheat sheet will tell you everything you need to format pages

https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Save your work and proceed to testing of change.

How to test your changes

Prerequisites for local test

- Node.js version 18 or above:
    - When installing Node.js, you are recommended to check all checkboxes related to dependencies.

run Docusaurus locally

Change directory:

```
cd wiki
```

Install NodeJS dependencies:

```
npm install
```

Run the development server:

```
npm start
```

The `cd` command changes the directory you're working with. In order to work with your newly created Docusaurus site, you'll need to navigate the terminal there.

The `npm install` installs all the dependencies for your site. You'll need to run this command once when you first download the project, and again only if you `package-lock.json` file changes.

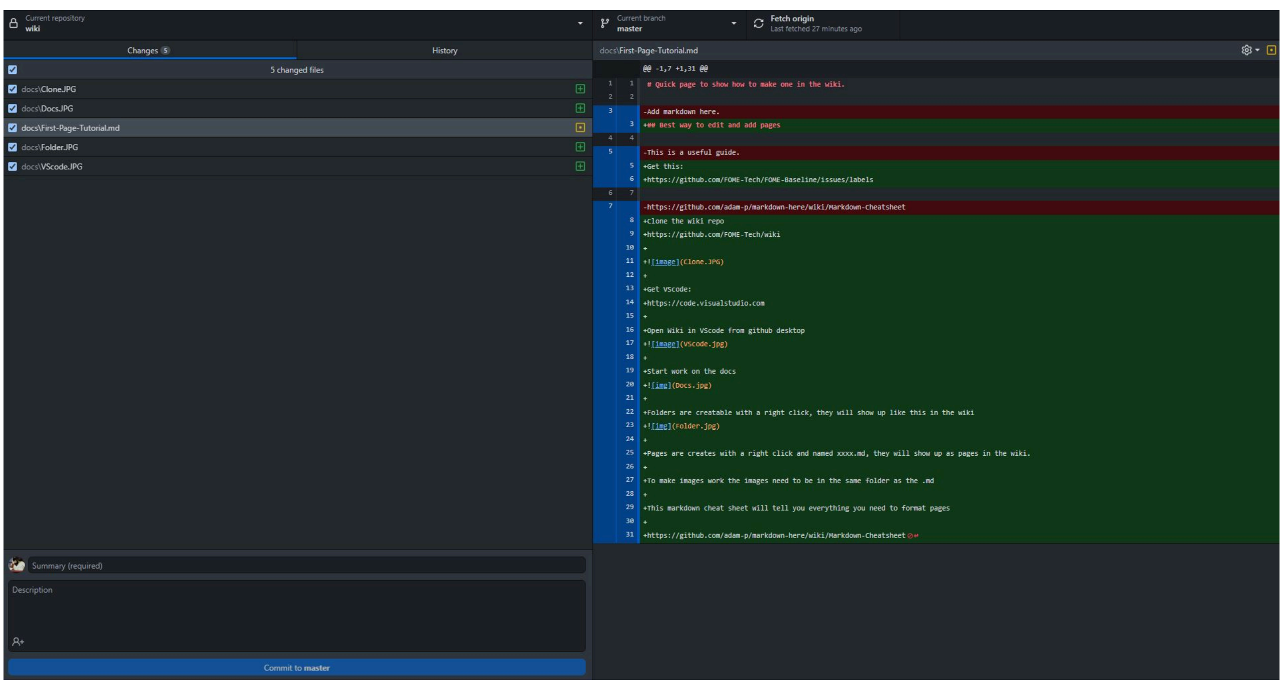
The `npm start` command builds your website locally and serves it through a development server, ready for you to view at http://localhost:3000.

Open `docs/13-How-to-edit-wiki.md` (this page) and edit some lines: the site **reloads automatically** and displays your changes.

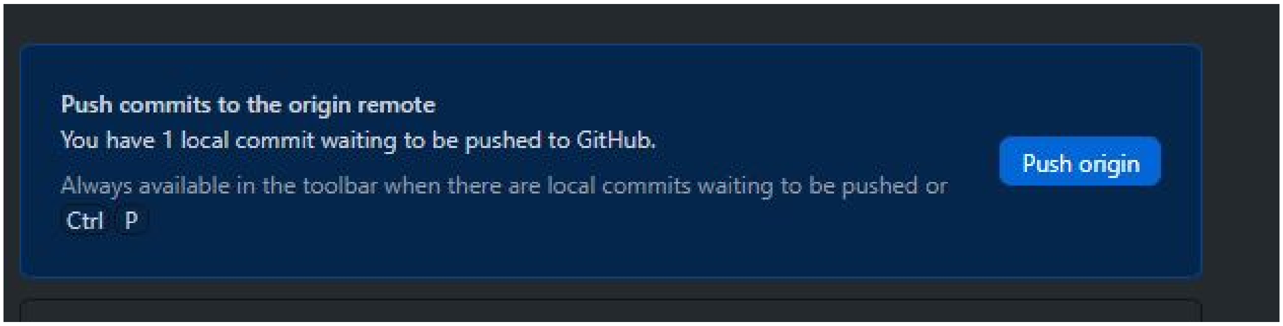
How to publish your change to wiki

in VScode and go to github desktop

It will ask you for a title, description and to commit the changes (see bottom left of picture)



Once that is done you can push your changes and they will get automatically updated to the wiki



How to structure

- Stuff!
- More stuff!
- Stuff that looks like stuff but isn't!

Subtitle 1

Subtitle 2

Subtitle 3

This kind of subtitle

You can also use like **this** **link to FOME's site**.

Additional Information

Some lists and links to useful information

📄 Trusted Suppliers of parts

This page contains information on suppliers that users of FOME have purchased items from and had good results.

Trusted Suppliers of parts

This page contains information on suppliers that users of FOME have purchased items from and had good results.

Deutsch connectors - aliexpress Jorch

Wiring (TXL, etc.) - Prowireusa

Wiring - 4RCustomsWire

Brand & Marketing Resources

Here, you can find assets related to our project's branding, trademarks, and marketing materials. While our project is open source and inclusive, we encourage users to respect our brand identity.

Assets

Logo

Download LogoFiles.zip

Font

Download Comfortaa.zip