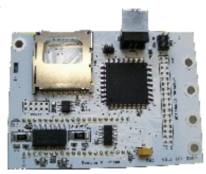
Megasquirt-3 Setting up

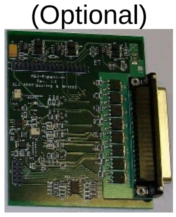
Megasquirt-3 Product Range

MS3 1.5.x

Dated: 2017-01-16



(Optional)



Instructions for setting up your Megasquirt-3 ECU.

This version of the documentation applies to the Megasquirt-3 range of products which includes:

- MS3 on V3.0 or V3.57 mainboard;
- with or without MS3X

and running MS3 firmware 1.4.x (See section 1.4 and 2.10.3 for more detail on version numbers.)

Not covered:

- MegaSquirt-1;
- EMS-Pro
- Megasquirt-2, Microsquirt or Microsquirt-module based ECUs
- MS3-Pro or products using MS3-Pro module - (see the MS3-Pro manuals)

Please report any omissions, errors or improvements on the www.msextra.com forum or to contact@megasquirt.co.uk

0: QuickStart Guide – in brief

This list is provided as the bare essentials that you **must** read. Full details of each step are provided within the body of the manual.

1. Buy or build your Megasquirt!
2. Select and assemble your fuel and ignition systems.
3. Setup your vehicle wiring harness ensuring it is fused.
4. Connect MAP hose to full vacuum source. (Speed Density only.)
5. Install TunerStudio and Megalogviewer software onto you computer.
6. Connect your Megasquirt to a 12V fused supply either in the vehicle or on the bench.
7. Get the serial comms set up.
8. Install the firmware if required.
9. Get your tuning computer to talk to the ECU.
10. Setup sensor calibrations to match hardware. (TPS, O2 sensor, temp sensors.)
11. Check all sensor inputs are reading sensibly in TunerStudio.
12. Set the base settings (engine type, ignition setup, toothed wheel type.)
13. Use the test mode to confirm injectors and coil(s) are functioning.
14. Check for RPM input.
15. Check the cranking timing with fuel disabled.
16. Start the engine and start tuning.

All parts are sold for OFF ROAD RACE-ONLY ground-vehicle use only – see section 1.1 for more details.

0: Contents

Table of Chapters

0: QuickStart Guide – in brief....2
0: Contents....3
1: Introduction....11
2: QuickStart Guide – expanded....15
3: Tuning the engine....56
4: Advanced topics....80
5: Troubleshooting....145
6: Appendix A: Firmware upgrade notes....156
7: Appendix B: Fuel calculations....162
8: Appendix C: Megasquirt Glossary of Terms....164
9: Revision history....171

Contents

0: QuickStart Guide – in brief....2
0: Contents....3
1: Introduction....11
1.1 Emissions and disclaimer....11
1.2 Required tools....11
1.3 How to use the manuals....11
1.4 Products Covered....12
1.4.1 MS3 on V2.2 mainboard (not covered)....13
1.4.2 MS3 on a V3.0 mainboard....13
1.4.3 MS3 on a V3.57 mainboard....13
1.5 Copyrights....14
2: QuickStart Guide – expanded....15
2.1 Buy or build your Megasquirt!....15
2.2 Select and assemble your fuel and ignition systems....15
2.3 Setup your vehicle wiring harness (loom) ensuring it is fused....15
2.4 Connect MAP hose to full vacuum source....15
2.5 Install TunerStudio and Megalogviewer....15
2.6 Power up your ECU....15
2.7 Get the serial comms set up....15
2.7.1 FTDI USB Latency Setting....19
2.7.2 portcheck for diagnostics....20
2.8 Turn off "Hide extension for known file types" (Windows Only)....20
2.9 Install the firmware if required....21
2.9.1 Windows....21
2.9.2 Mac / Linux....24
2.10 Get your tuning computer to talk to the ECU....25
2.10.1 TunerStudio basics....29
2.10.2 Firmware, TunerStudio, tune files (MSQ) - what's all that?....30
2.10.3 Version numbers....30
2.10.3.1 Mainboard version....30
2.10.3.2 Firmware Version....30
2.10.3.3 TunerStudio Version....31
2.10.3.4 MegaLogViewer Version....31
2.10.3.5 Operating system version....31
2.11 Setup sensor calibrations to match hardware....31
2.11.1 Calibrate TPS....31
2.11.2 Calibrate MAP/Baro....32
2.11.3 Unlock calibrations....32
2.11.4 Calibrate CLT and MAT sensor....33
2.11.5 Calibrate AFR table (O2 sensor)....34
2.11.6 Lock calibrations....34
2.12 Check all sensor inputs are reading sensibly in TunerStudio....34

2.12.1 MAP sensor check....34
2.12.2 TPS check....35
2.12.3 CLT sensor check....35
2.12.4 MAT sensor check....35
2.12.5 O2 sensor check....35
2.12.6 MAF sensor check....35
2.12.7 Battery voltage check....35
2.13 Set the base settings....35
2.13.1 Control Algorithm choice....36
2.13.2 Engine Types....37
2.13.2.1 One cylinder engines....37
2.13.2.2 Two cylinder engines....38
2.13.2.3 Three cylinder engines....38
2.13.2.4 Four cylinder engines....38
2.13.2.5 Five cylinder engines....38
2.13.2.6 Six cylinder engines....39
2.13.2.7 Seven cylinder radial engines....39
2.13.2.8 Eight cylinder engines....39
2.13.2.9 Nine cylinder radial engines....39
2.13.2.10 Ten cylinder engines....39
2.13.2.11 Twelve cylinder engines....39
2.13.2.12 Sixteen cylinder engines....39
2.13.2.13 Wankel Rotary Engines....39
2.13.3 Engine and Sequential settings....40
2.13.4 Injector dead-time....40
2.13.5 Ignition setup....41
2.13.6 Idle Valve....42
2.13.7 Other inputs/outputs....42
2.13.8 Gauge/Setting Limits....43
2.13.9 Power Cycle Required....43
2.13.10 Configuration Error....43
2.14 Use the test mode to confirm injectors and coil(s) are functioning....44
2.14.1 Injector testing....45
2.14.2 Coil testing....46
2.14.3 Fuel pump testing....46
2.14.4 Outputs testing....46
2.14.5 Idle Valve Testing....47
2.14.5.1 Using the test mode to determine stepper homing steps....49
2.15 Check for crank/cam tach-in signals....49
2.16 Check for RPM input....52
2.17 Check sensors during cranking....52
2.18 Sample datalog....52
2.19 Check the cranking timing with fuel disabled....52
2.20 Start the engine and start tuning....53
2.21 Run engine up to temperature....54

2.22 Setting running timing....54
3: Tuning the engine....56
3.1 How it works - fuel....56
3.1.1 Cranking mode....56
3.1.2 Run mode....56
3.2 How it works - ignition....58
3.2.1 Cranking mode....58
3.2.2 Run mode....58
3.3 Tables and curves....59
3.4 Tuning fuel....59
3.4.1 Tuning fuel - Speed Density....59
3.4.1.1 AFR table (SD)....60
3.4.1.2 VE table (SD)....60
3.4.2 Tuning fuel - Percent Baro....62
3.4.3 Tuning fuel - Alpha-N....62
3.4.4 Tuning fuel - MAF....62
3.4.4.1 MAF transfer curve....62
3.4.4.2 AFR table (MAF)....63
3.4.4.3 VE trim table (MAF)....63
3.4.4.4 MAFload....64
3.4.5 Tuning fuel - ITB....64
3.4.5.1 ITB Load TPS Switchpoint Curve....65
3.4.5.2 ITB Load at TPS Switchpoint Curve....65
3.4.5.3 AFR Table (ITB)....66
3.4.5.4 VE Table (ITB)....66
3.4.5.5 ITB technical....66
3.4.5.6 Tuning For Idle Air Control....68
3.4.6 Basic acceleration enrichment (AE) tuning....68
3.4.6.1 Main accel enrich settings menu....69
3.4.6.2 Time-based Accel....69
3.4.6.3 Getting it dialed in....70
3.4.7 Startup / warmup fueling....71
3.4.7.1 Priming pulse....72
3.4.7.2 Cranking pulse....72
3.4.7.3 Afterstart (ASE) percentage adder....72
3.4.7.4 Afterstart (ASE) taper....72
3.4.7.5 Warmup enrichment percent multiplier....72
3.4.7.6 Putting it all together....72
3.5 Tuning spark....73
3.6 Getting a good idle....74
3.6.1 Correct Fuel (mixture)....74
3.6.2 Correct Timing....74
3.6.3 Correct Airflow....74
3.7 Datalogging (serial)....75
3.7.1 Starting a datalog....75

3.7.2 Stopping a datalog....75
3.8 SD card datalogging....76
3.8.1 Supported cards....76
3.8.2 Settings....76
3.8.3 Button / Light....77
3.8.4 Button Usage....77
3.8.5 LED Flash Codes....77
3.8.6 Log File Extraction....77
3.8.7 Error codes....78
4: Advanced topics....80
4.1 Sequential Fuel....80
4.1.1 Configuration....80
4.1.2 Injector timing....81
4.1.3 Sequential/Semi-sequential Timing....81
4.1.4 Injector Small Pulsewidths....81
4.2 Rev Limiter....82
4.3 Torque Convertor Lockup....82
4.4 Alternator Control....82
4.5 MAP Sample Settings....83
4.5.1 Event Average....83
4.5.2 Timed Minimum....83
4.6 Flex Fuel....85
4.6.1 Basic System....85
4.6.2 Advanced System....85
4.7 Overrun Fuel Cut....87
4.8 AFR/EGO Control....87
4.8.1 Simple Algorithm with Narrowband Sensor....88
4.8.2 Simple Algorithm with Wideband Sensor....88
4.8.3 PID Algorithm with Wideband Sensor....88
4.9 AFR Safety System....89
4.10 Fuel Pump and Pressure....89
4.11 Knock Detection....90
4.12 Wankel Rotary Engines....91
4.12.1 FC 2 rotor....91
4.12.2 FD 2 rotor....92
4.12.3 RX8 2 rotor....93
4.12.4 Three rotor....93
4.12.5 Four rotor....94
4.12.6 Rotary Split table....95
4.13 Closed-Loop Idle....95
4.14 Idle Advance....97
4.15 Enhanced Accel Enrichment (EAE)....98
4.15.1 Tuning EAE....98
4.16 X-Tau....99
4.16.1 X-Tau tuning tips....99

4.17 Boost Control....100
4.17.1 Single-solenoid Boost Control....100
4.17.2 Open-loop boost control concepts....102
4.17.3 Open-loop Tuning Tips....104
4.17.4 Closed-loop boost control concepts....105
4.17.5 Settings....106
4.17.6 Closed-loop Tuning Tips....106
4.17.7 Boost Dome Control....107
4.17.8 Dome control concepts....108
4.17.9 Dome MAP sensor configuration....110
4.17.10 Wastegate Dome Pressure Control Settings 1....111
4.18 Anti-lag (ALS)....111
4.19 Variable Valve Timing (VVT, VCT, VTEC, VANOS)....112
4.19.1 On/off mode setup....112
4.19.2 Variable mode initial setup....112
4.20 Table-switching / Blending / Staging....113
4.20.1 Example scenarios....114
4.20.2 Fuel blending and switching types....114
4.20.3 Fuel blending and switching settings....117
4.20.3.1 Single table....117
4.20.3.2 Single table with table-switching....117
4.20.3.3 Single table with table-blending....118
4.20.3.4 Combined tables - secondary additive table....118
4.20.3.5 Combined tables - secondary multiplicative table....119
4.20.3.6 Secondary algorithm blended table....119
4.20.3.7 Dual table....120
4.20.3.8 Staged Injection....120
4.20.4 Spark/ignition blending and switching types....121
4.20.5 Spark/ignition blending and switching settings....123
4.20.5.1 Single table....123
4.20.5.2 Single table with table-switching....123
4.20.5.3 Single table with table-blending....123
4.20.5.4 Combined tables - secondary additive table....124
4.20.5.5 Secondary algorithm blended table....124
4.20.6 AFR blending and switching types....125
4.20.7 Boost blending and switching types....126
4.20.8 Table-Switching and Dual Fuel....127
4.21 Wheel Speed, Shaft Speed and Gear Detection....127
4.22 Exhaust Gas Temperature (EGT)....127
4.23 Generic Sensors....128
4.24 Accelerometer input....128
4.25 Traction Control....129
4.26 Launch, 2-step, 3-step, Transbrake....130
4.26.1 Launch and Flat-Shift....130
4.26.2 Variable Launch....130

4.26.3 Trans-brake / Timers / Bump-Box....130
4.26.4 Throttle-stop....131
4.26.5 3-step....131
4.26.6 Timed Retard After Launch....131
4.26.7 Drag race example (manual trans)....131
4.26.8 Speed based launch control....131
4.27 Sequential shift-cut....131
4.28 Nitrous Oxide (N2O)....131
4.28.1 System types....132
4.28.2 Connection....132
4.28.3 Tuning considerations....132
4.29 Water Injection....133
4.30 High Power Time Enrichment....133
4.31 Oil Pressure....133
4.32 Programmable On/Off Outputs....133
4.32.1 Bitwise operations....133
4.33 Generic PWM....134
4.34 Generic Closed-Loop....135
4.35 Check Engine Light (CEL) / Limp Mode....135
4.35.1 Configuration tips....135
4.35.2 Limp Mode....136
4.35.3 Fallback MAP table....136
4.35.4 CEL Flash Codes....137
4.36 CAN Bus Expansion....137
4.36.1 CAN protocol....138
4.36.2 Device-to-device....138
4.36.3 Passthrough....138
4.36.4 CAN wiring....139
4.36.5 CAN settings and data types....140
4.36.5.1 PWM polling....140
4.36.5.2 Digital ports....140
4.36.5.3 PWM outputs....140
4.36.5.4 CANADCs....140
4.36.5.5 VSS....140
4.36.5.6 Gear....140
4.36.5.7 Innovate EGO....140
4.36.5.8 GPS....140
4.36.5.9 RTC....141
4.36.6 CAN Broadcasting....141
4.36.7 Example Settings - CANEGT....141
4.36.8 Example Settings - JBperf IOextender....141
4.36.9 Example Settings - other devices....142
4.37 Test Modes....142
4.38 Long Term Trim (LTT/LTFT)....143
4.39 Race Technology dash integration....143

4.39.1 CAN input....143
4.39.2 CAN output....143
5: Troubleshooting....145
5.1 Resets....145
5.2 RPM-Sync issues....145
5.2.1 Lost sync reasons....146
5.3 Noise Filtering....148
5.3.1 Noise filter....150
5.3.2 Tach interrupt masking....150
5.3.3 Tach period rejection....151
5.3.4 Polarity Check....151
5.4 Communications issues....152
5.4.1 Windows device....152
5.4.2 Mac OSX device....152
5.4.3 Linux device....153
5.4.4 Loopback test for serial or USB-serial cable....153
5.4.5 Loopback test to Megasquirt board....154
5.4.6 Comms test to Megasquirt board....154
6: Appendix A: Firmware upgrade notes....156
6.1 Firmware versions....156
6.2 Alpha, beta, release code? What's the difference?....156
6.3 Upgrading from 1.3.x....156
6.4 Upgrading from 1.2.x....157
6.5 Upgrading from 1.1.x or 1.0.x....158
6.6 Upgrading from MS2/Extra 3.0.x or 3.1.x....159
6.7 Upgrading from MS2/Extra 2.1.0....159
6.8 Upgrading from MS2/BG....159
6.9 Upgrading from MS1/Extra....160
6.10 Upgrading from MS1 (base)....160
7: Appendix B: Fuel calculations....162
7.1 Terms Used....162
7.2 Speed Density....162
7.3 Alpha-N (pure)....162
7.4 Alpha-N (hybrid)....162
7.5 %baro....163
7.6 MAF....163
8: Appendix C: Megasquirt Glossary of Terms....164
9: Revision history....171

1: Introduction

This manual covers general setup and configuration information that applies to the whole Megasquirt-3 product range, it should be used in conjunction with the TunerStudio Reference Guide and the Hardware Guide for your product.

1.1 Emissions and disclaimer

All parts are sold for OFF ROAD RACE-ONLY ground-vehicle use only, or vehicles that pre-date any federal and state emissions control requirements. Aftermarket EFI/EMS systems are not for sale or use on pollution controlled vehicles. Alteration of emission related components constitutes tampering under the US EPA guidelines and can lead to substantial fines and penalties. Your country/state/district may also have specific rules restricting your tampering with your vehicle’s emissions system.

Race parts are inherently dangerous and may cause injury or damage if improperly modified or altered before use. The publishers of this manual will not be held liable for and will not pay you for any injuries or damage caused by misuse, modification, redesign, or alternation of any of our products. The publishers of this manual will not be held in any way responsible for any incidental or consequential damages including direct or indirect labor, towing, lodging, garage, repair, medical, or legal expense in any way attributable to the use of any item in our catalog or to the delay or inconvenience caused by the necessity of replacing or repairing any such item.

1.2 Required tools

Tuning laptop

Stroboscopic timing light

Multi-meter (volts, ohms)

Screwdrivers

Wire cutters

Terminal crimpers

Soldering iron and solder

Heat-shrink tubing

Fire extinguisher

Although not essential, the following are highly recommended:

Oscilloscope or scope-meter or soundcard scope

Test light

Power probe

1.3 How to use the manuals

The documentation for the Megasquirt-3 product family is split across a number of manuals.

Megasquirt-3 Setting Up			
MS3base/V3.0 Hardware	MS3X/V3.0 Hardware	MS3/V3.57 Hardware	MS3X/V3.57 Hardware
Megasquirt-3 TunerStudio Reference (includes MegaLogViewer field names)			

- Setting Up - this covers initial software installation, tuning concepts and some feature guides. Applies to all Megasquirt-3 variants.
- Hardware - wiring, inputs and outputs specific to your ECU model. Be sure to check that you are using the manual that applies to your product!
- TunerStudio Reference - a reference to all of the individual settings. Applies to all Megasquirt-3 variants.

As an absolute minimum, ensure that you have followed all of the steps in the **Quickstart Guide** from the **Setting Up** manual.

Customers new to EFI or less experienced with Megasquirt are strongly advised to read the manuals in greater detail.

This guide includes a number of notes which are indicated as follows:

This symbol indicates an “Information” note.

This type of note is typically used to indicate how to enable an option within the software and other useful tips which are intended to make the software easier to use.

This symbol indicates a “Caution” note.

This type of note is typically used to indicate the order in which you should make changes in order to tune your Megasquirt most efficiently.

This symbol indicates a “Warning” note.

Configuring your Megasquirt incorrectly can potentially cause damage to your engine. In addition, working with some engine components such as ignition coils can be dangerous. Warning notes indicate specific areas where you need to exercise extreme caution.

Please do not rely on these warnings as your only criteria for taking care of yourself or your engine!!

For additional help and support, visit the website **www.msextra.com**

1.4 Products Covered

The Megasquirt-3 range of products all use the same processor at the core but are packaged differently for different markets. The core configuration and setup is almost identical. There are minor differences in ignition settings and wiring.

It is important to understand exactly which product you have.

More details on version numbers are available in section 2.10.3.

1.4.1 MS3 on V2.2 mainboard (not covered)

Although technically possible, the combination of a Megasquirt-3 daughtercard and a V2.2 mainboard is not supported.

Hardware and wiring are **not covered** in these manuals.

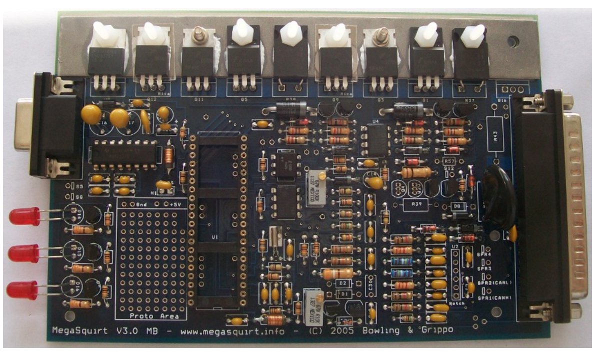
1.4.2 MS3 on a V3.0 mainboard

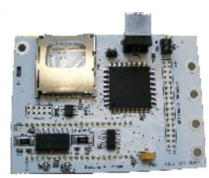
This combination uses a self-assembled V3.0 mainboard and a Megasquirt-3 daughtercard. This is aimed at the DIYer that wants the challenge of self-assembly and gives flexibility for future extension and modification.

By customizing, more inputs and outputs are available. Supports low-impedance injectors. Includes an internal MAP sensor.

Not recommended in kit form for customers afraid of a soldering iron.

The MS3/V3.0 combination is often assembled by sub-suppliers and re-sold to end-users as a completed ECU.


V3.0 mainboard (inside)


MS3 plug-in daughtercard (inside)

Hardware and wiring without MS3X is covered in the MS3base/V3 Hardware Manual.

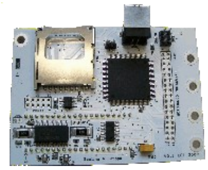

Optional MS3X expansion (inside)

Hardware and wiring with MS3X is covered in the MS3X/V3 Hardware Manual.

1.4.3 MS3 on a V3.57 mainboard

This combination uses a factory-assembled V3.57 mainboard and a Megasquirt-3 daughtercard. This is aimed at the customer that wants a cased Megasquirt-3 but prefers to buy pre-assembled. The V3.57 mainboard has less flexibility for future extension and modification. By customising, more inputs and outputs are available. Supports low-impedance injectors. Includes an internal MAP sensor.


V3.57 mainboard (inside)


MS3 plug-in daughtercard

Hardware and wiring without MS3X is covered in the MS3base/V3.57 Hardware Manual.


Optional MS3X expansion (inside)

Hardware and wiring with MS3X is covered in the MS3X/V3.57 Hardware Manual.

1.5 Copyrights

2: QuickStart Guide – expanded

This section expands on the bare bones quickstart guide with more detail in each step.

2.1 Buy or build your Megasquirt!

Before you can begin your installation, you need to have purchased your Megasquirt.

In the case of a kit you need to assemble it. Kit assembly is covered in the hardware guide for your product.

2.2 Select and assemble your fuel and ignition systems

Refer to the hardware guide for your product for fuel and ignition system details.

Ensure that you have resistor type spark plugs. Non-resistor plugs are not suitable for use with EFI systems.

2.3 Setup your vehicle wiring harness (loom) ensuring it is fused.

Wiring is covered in the hardware guide for your product.

Any unused sensor inputs (MAP, CLT, MAT, TPS, O2) should be connected to sensor ground to prevent the signal 'floating'.

2.4 Connect MAP hose to full vacuum source.

Most installs use the Speed-Density method for fuel calculation and require a MAP sensor. Ensure that the MAP sensor hose is connected to a source of full intake vacuum. i.e. fully downstream of the throttle plate. Engines with individual throttle bodies will want to tee into each throttle runner to gain an averaged signal.

In cases where MAF or TPS (Alpha-N) are used exclusively for load measurement, then the MAP sensor connection can be omitted.

2.5 Install TunerStudio and Megalogviewer

You ECU is tuned using TunerStudio on a with Windows, Mac or Linux computer.

If your ECU came with an install CD or pen drive, run it now to install the software.

The latest versions can be downloaded from www.msextra.com/downloads

For Linux, your current user needs to be a member of the 'dialout', 'lock' and/or 'uucp' groups in order to access your comms port.

See section 2.10.3 for information on version numbers.

2.6 Power up your ECU

Now it is time to power up the ECU. Either in the vehicle or on the bench, apply a fused 12V supply to power and ground inputs. Refer to the hardware guide for your product for wiring details.

2.7 Get the serial comms set up

MS3 uses serial communications for tuning. Some older or "industrial" type laptops will include a true serial port, most current computers only have USB ports.

The MS3 features two serial connections:

a) Internal USB serial. This is FTDI based. Before connecting, be sure to install the FTDI device driver. This should be supplied on the install CD/pen-drive. Or download and run the latest 'setup executable' from http://www.ftdichip.com/Drivers/VCP.htm

See 2.7.1 to configure the Latency Timer setting for the internal FTDI USB port.

b) DB9 serial. This can be connected to a regular DB9 serial port with a straight-through cable. Or to connect to your computer with your own USB-serial adapter.

USB was not designed for automotive environments and some installs may experience problems, especially when using long cable runs. If you experience similar problems it is recommended that you use a short USB-serial cable and a standard DB9-DB9 extension cable. RS232 serial is far better at dealing with electrically noisy environments than USB.

MS3 USB-serial and DB9 serial ports.

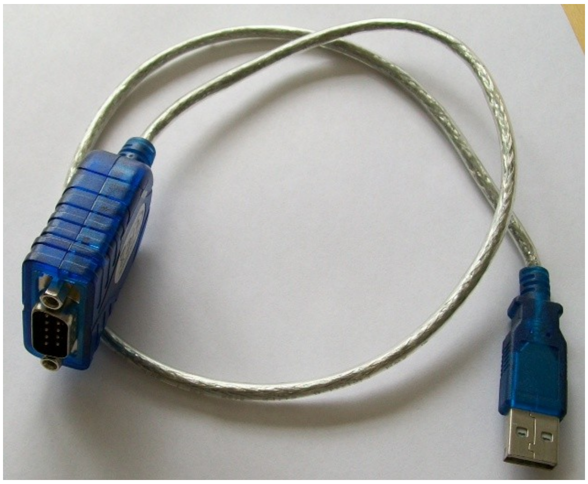
The USB socket implements an FTDI based USB-serial adapter internally.

Use one port or the other – **do not connect to both ports at the same time.**

Standard DB9 male connector on computer.

Standard DB9-DB9 straight through serial cable.

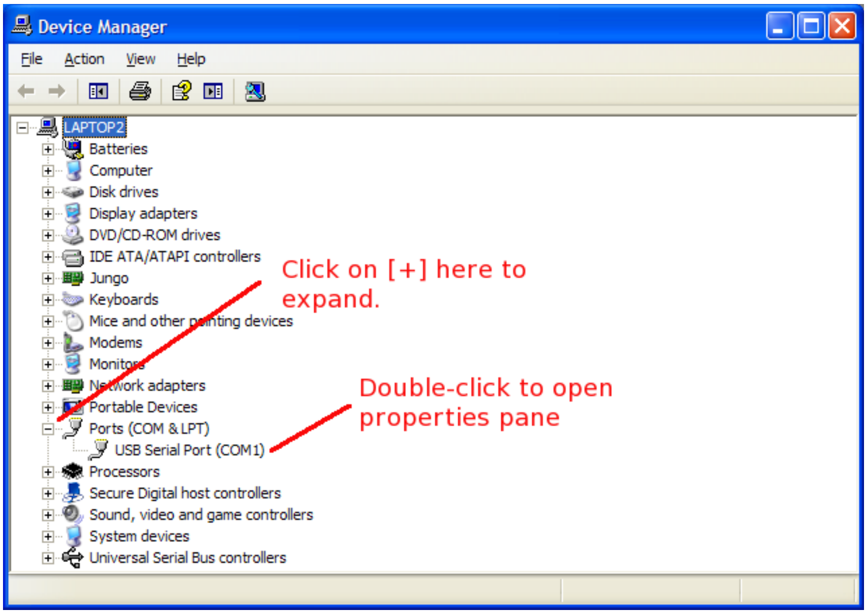
 Computer USB socket

 Typical USB-serial adapter.

Before inserting a USB-serial adapter, follow the manufacturers' instructions for driver configuration.

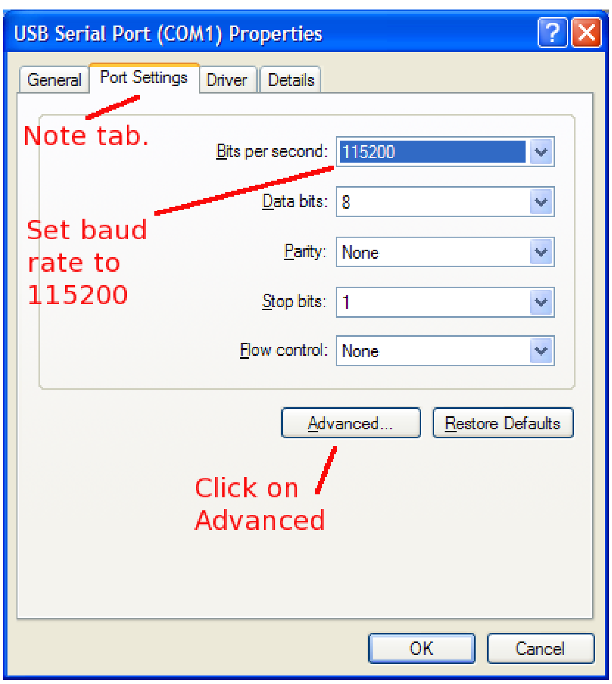
 Typical USB A-B cable.

(Windows only) Having found your true serial port or plugged in your USB-serial adapter, it is desirable to opening up Device Manager and identify the COM port number. The method to open Device Manager changes with each Windows version, so will not be documented here.

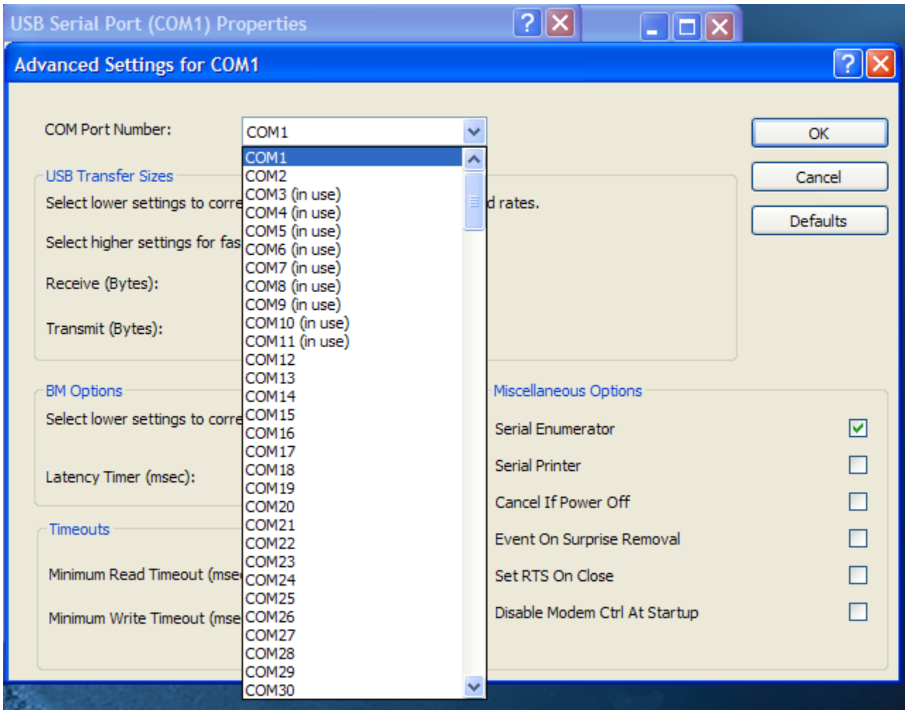


If you DO NOT see "Ports" listed in the devices then you do not have any ports setup. This would usually indicate either you forgot to plug in your USB-serial cable or there is a driver problem. Usually the best solution to the driver problem is to unplug the cable, uninstall the driver and then re-install the driver.

Double click on the port and you will see the Properties pane, select the Port Setting tab.



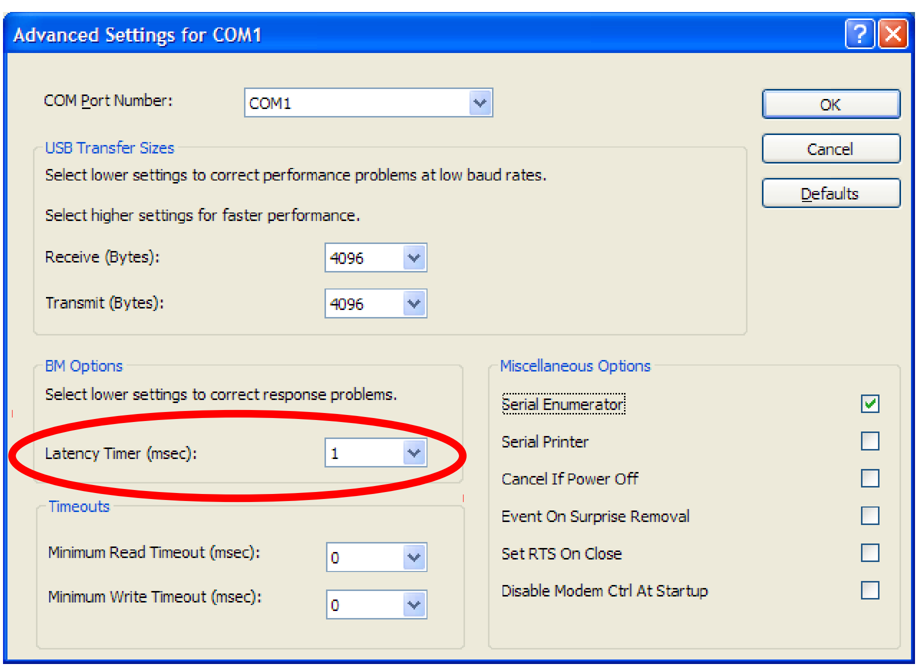
Having set the baud rate, click on the Advanced button. Here you can verify or change the COM port number and set the critical Latency setting for FTDI interfaces.



Here we see that this USB-serial adapter has been assigned COM1 – remember that number!

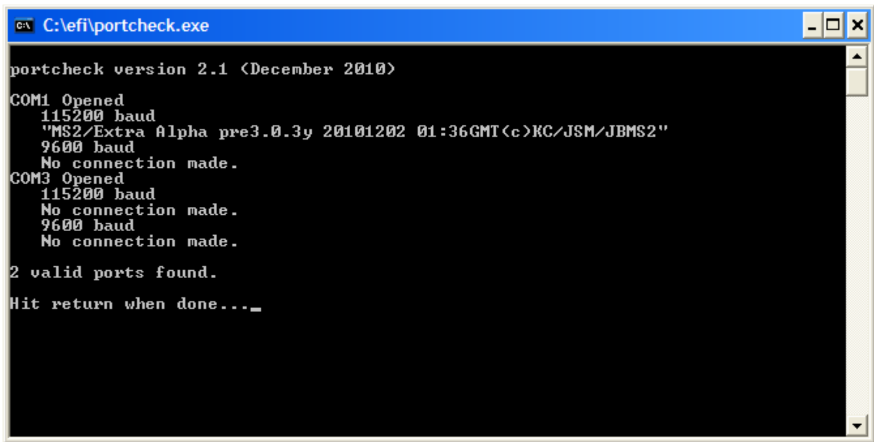
2.7.1 FTDI USB Latency Setting

For FTDI interfaces or when using the direct USB socket on the MS3, you must also configure the "latency" setting to 1. By default the driver is really slow.



2.7.2 portcheck for diagnostics

The portcheck program (Windows only) can be used to investigate communication problems - download and run portcheck.exe from www.msextra.com/downloads make sure your comms cables are connected and the ECU is powered up.



Take a note of the COM port number listed above. In this case it was COM1.

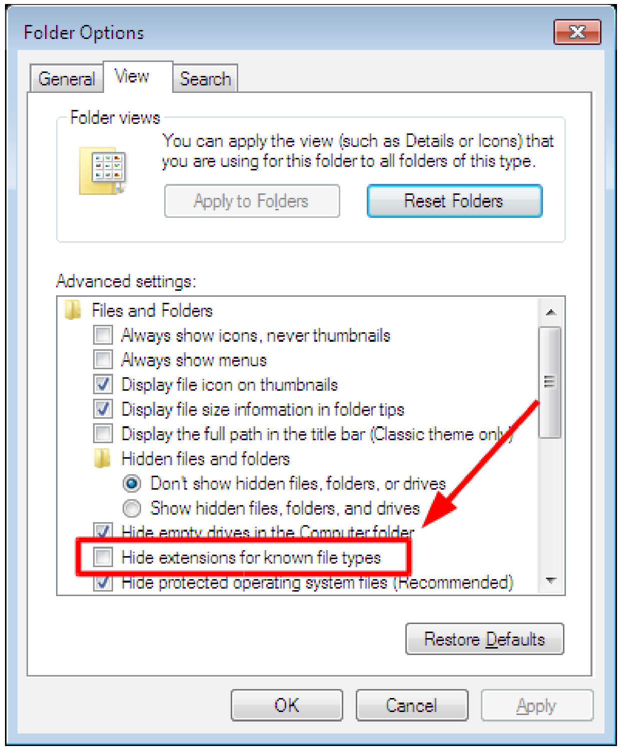
Should you encounter problems getting your serial connection working, refer to the fault finding section.

2.8 Turn off "Hide extension for known file types" (Windows Only)

Windows has a feature to hide part of the filename. It is strongly advised that this is disabled to prevent confusion.

Go to Folder Options (Use Search to find it if required.)

Ensure that "Hide extensions for known files types" is not ticked.



2.9 Install the firmware if required

Plug'n'play ECUs will usually be supplied with the firmware loaded and a tune ready for your engine. In these case there is no reason to load new firmware and you can skip to step 2.10.

For most other situations it is desirable to load in the current firmware version to the ECU before you begin tuning.

The latest versions can be downloaded from www.msextra.com/downloads

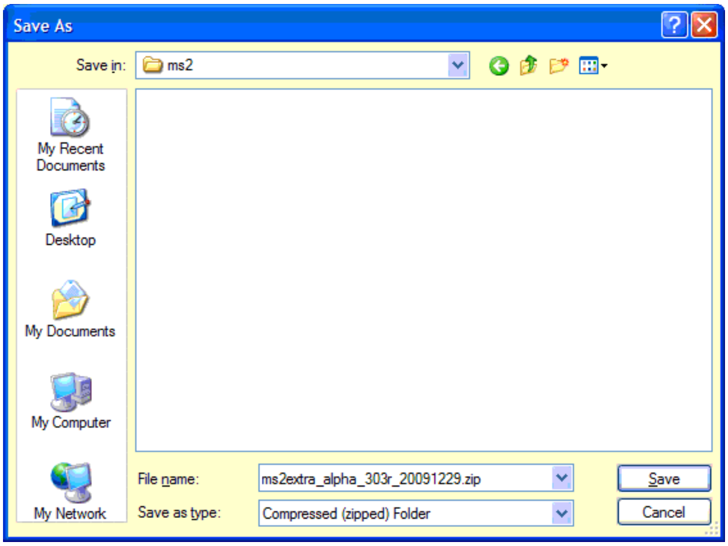
Note! Instead of using the 'DOS' type firmware loader, TunerStudio 3.x includes a wizard style firmware loader - use that.

2.9.1 Windows

i. Create a folder/directory for MS3. e.g. C:\MS3

(Use "My Computer" or "Windows Explorer", open Local Disk C: and Create New Folder, name it MS3).

ii. Download the firmware zip file from the downloads page. Right click and Save Target As into the C:\MS3 folder.

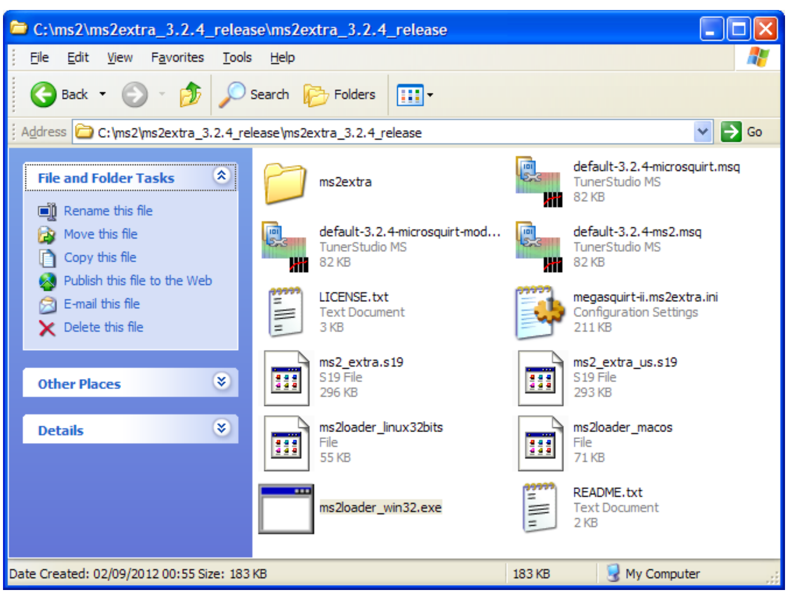


iii. Use My Computer / Windows Explorer and browse to this folder. (e.g. C:\MS3)

iv. Right click on the file you saved and choose Extract All.

v. The defaults are OK, and just click next. (Note that Windows adds a second level of directory.)

vi. You should now find the files have been extracted. Open up the folders until you see this:



vii. Users loading firmware for the first time can skip to viii.

Upgrade users should ensure that you have used Save Tune As to save your existing tune settings.

viii. Exit your tuning software and ensure nothing is using the serial port.

ix. Double click on ms3loader_win32.exe (If you get a security warning, click Run.)

The loader is text based. Answer all of the questions by pressing the required key on your keyboard.

Be careful on the ECU type choice:

MS3 = ECUs with an MS3 daughtercard (also MS3-Gold.)

MS3-Pro = black plastic cased ECU from DIYautotune (also MS3-Pro-module.)

Example:

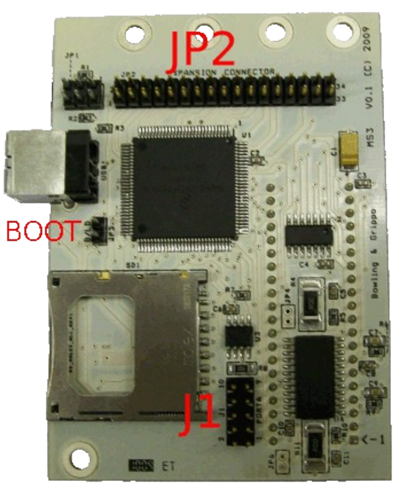
```
Megasquirt-3 Firmware Loader 3.33 2013-12-15
.
Remove the fuse powering your ignition coils.
(Spark outputs may be undefined until you re-load your settings.)
Press enter to continue...
.
Ensure that your Megasquirt/Microsquirt/ECU is
-connected to your computer
-has fused 12V power connected.
Press enter to continue...
.
Ensure that no tuning software is running.
Be sure to Quit the software, not just minimise it..
Press enter to continue...
.
Debug message Level:
1: Normal
2: More detail
3: + serial comms
4: + the s19 file as parsed
Selection (default: 1):
.
Do you want to scan your serial ports automatically? (y/n default y)
.
COM1 MS3-type ECU detected
.
Do you want to use COM1 (y/n default = y)
.
.
.
Found firmware files : ms3pro.s19 : ms3.s19
.
Auto-selected firmware file: ms3.s19 - ok? (y/n default = y)
.
Fetching format
Checking serial format
New serial format >= 001 detected

Preserve sensor calibrations,
and I/O pin states (y/n, default: y)?
.
==========================================================
Settings selected:
Serial port: COM1
S19 File: ms3.s19
Debug level: 1
Jumperless reflash enabled
Preserving sensors calibrations, I/O pin states
Not preserving MSQ / tuning settings
==========================================================
```

```
Press enter to continue...
Sending jumperless flash command
Attempting Wakeup...
Fetching port states
Fetching : sensor calibrations
Checking flash status.
Status ok.
Erasing main flash!
Erased.
Sending firmware to controller...
[==================================================================>] 100.0%
Sending port states...
Sending : sensor calibrations
Wrote 249122 bytes
*** Settings not preserved. Please re-load your MSQ / tuning settings ***
Press enter to continue...
.
```

If this takes longer than a minute then you likely need to adjust the port settings of your USB-serial adapter to reduce the latency setting.

In rare cases, the first time you install firmware on an MS3 it might be necessary to use the boot jumper.



MS3 showing boot jumper location.

2.9.2 Mac / Linux

Mac OS X and Linux users need to run the firmware loader from the command line. The exact procedure varies by release and with the tools you have installed. The loader needs to be run from the directory containing the firmware files.

Depending on distribution you may be able to "open in terminal" or "open here" to launch a terminal in the correct directory, otherwise use 'cd' to change directory to the correct place. Then ./ms3loader_XXXX as appropriate. Note that you MUST use the keyboard, put the mouse to one side.

Then step through the firmware loading process as described in 2.9.1.

2.10 Get your tuning computer to talk to the ECU

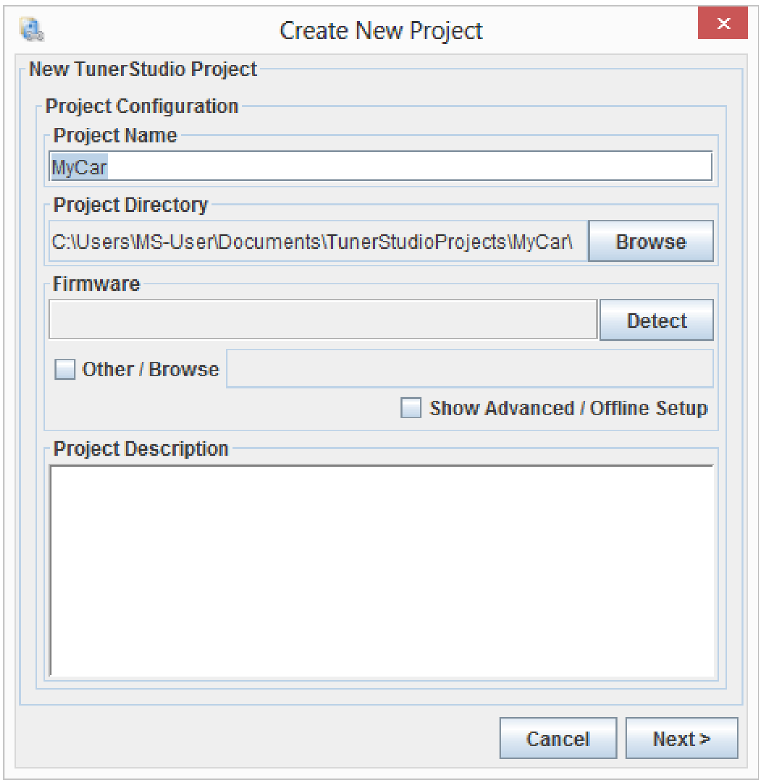
i. **Start TunerStudio**

ii. **Create a new project.**

(More details on this step can be found in the TunerStudio Reference manual in section 2.1)

From the Start Screen select File > New Project

or from the Main Screen select File > Project > New Project.

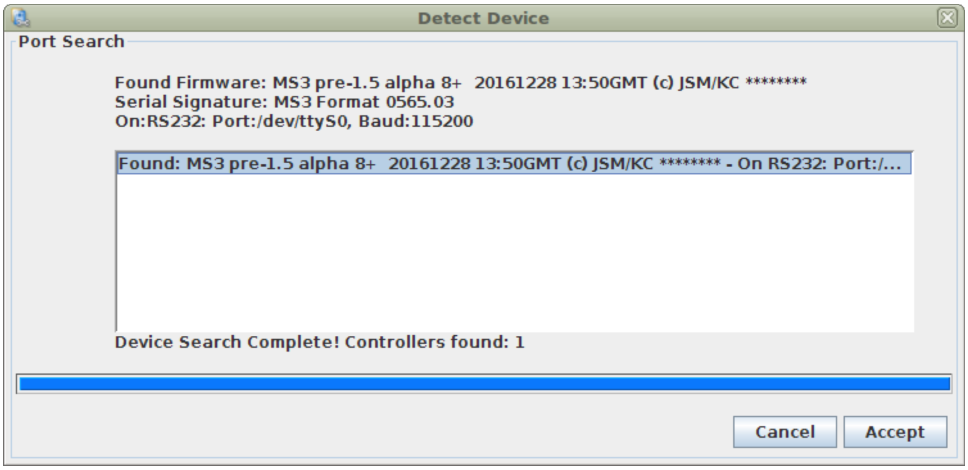


In the Project Name box you can give the project a meaningful name if wish.

Double check that your Megasquirt is powered up and connected to your computer by USB or serial.

iii. Click on Detect

When you click the “Detect” button a “Detect Device” screen will appear and TunerStudio will attempt to find your Megasquirt and its firmware version. If successful it will list your Megasquirt, firmware version and baud rate. Check to see that these appear to match your Megasquirt and then click “Accept” to continue. If the “Detect Device” screen reports “No controller found” check the connection between your computer and your Megasquirt and ensure that your Megasquirt has power and try again. If this still does not resolve the problem move on to the “Other / Browse” method described below to set the firmware up manually.



If your Megasquirt is identified correctly, click “Accept”. Click Next on the 'Create New Project' screen and you can now move on to step v.

If your Megasquirt is identified correctly but clicking “Accept” displays a message that reports “TunerStudio does not have a configuration to support the found hardware” then you will need to use the “Other / Browse” method described below to set up communications with your firmware manually.

iv. Other / Browse

This method is recommended only if TunerStudio is not able to detect your Megasquirt, or if you are using a beta version of the firmware.

The firmware on your Megasquirt may have been provided on a disk or downloaded as a .zip file from the www.msextra.com website. If you have downloaded your firmware and haven't done so already, extract the contents of the .zip file to a directory of your choice.

Clicking in the box next to “Other / Browse” opens a screen which lets you browse to your firmware folder. From here you should be able to select the .ini file that you want to use. It is essential that you use the .ini file that is appropriate to your Megasquirt. There are two .ini files to choose from:

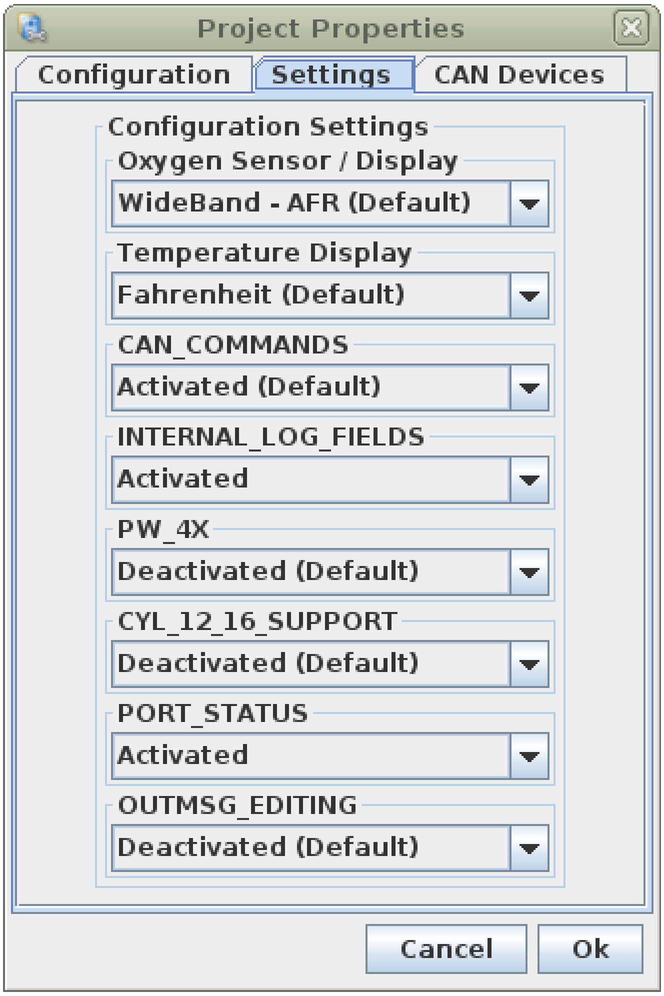
1. ms3.ini – used for traditional MS3.
2. ms3pro.ini – used for the MS3-Pro from DIYautotune.

Select the file that is appropriate to your Megasquirt and then click Next.

v. Show Advanced / Offline Setup

This setting is not generally useful, leave alone.

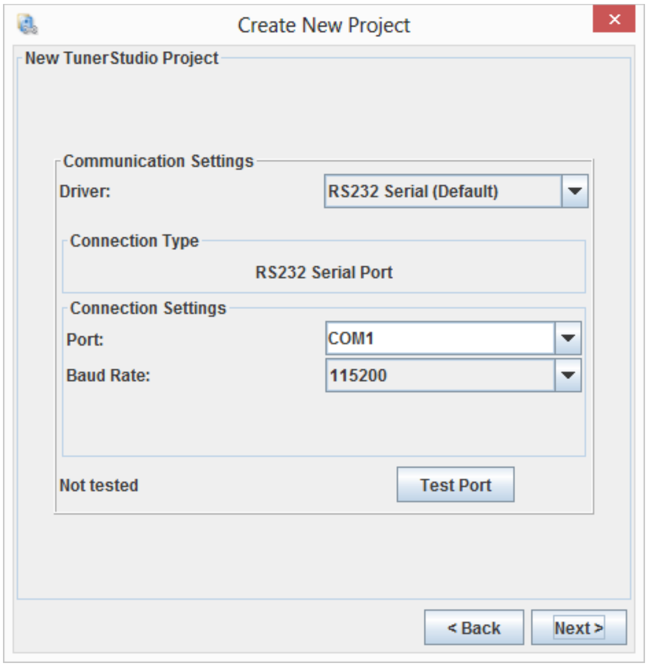
vi. Project Properties - Settings



Oxygen Sensor/Display - selects the default gauges for oxygen or lambda sensor (the AFR/EGO settings must also be used to configure the firmware)
Temperature Display - preferred temperature units
CAN_COMMANDS - normally Activated. Required for TunerStudio to communicate with remote devices over a CAN network instead of directly with USB or serial.
INTERNAL_LOG_FIELDS - normally Deactivated. Allows developers to log special fields into SDcard logs.
PW_4X - normally Deactivated. Used in conjunction with the advanced firmware setting to enable four times the pulsewidth on large super-low revving engines.
CYL_12_16_SUPPORT - normally Deactivated. Enables gauges and settings for cylinders 9-16. Only useful if your ECU has the relevant hardware.
PORT_STATUS - normally Deactivated. Enables individual dash indicators for every input/output pin.
OUTMSG_EDITING - normally Deactivated. Enables a system for sending data to other CAN devices.

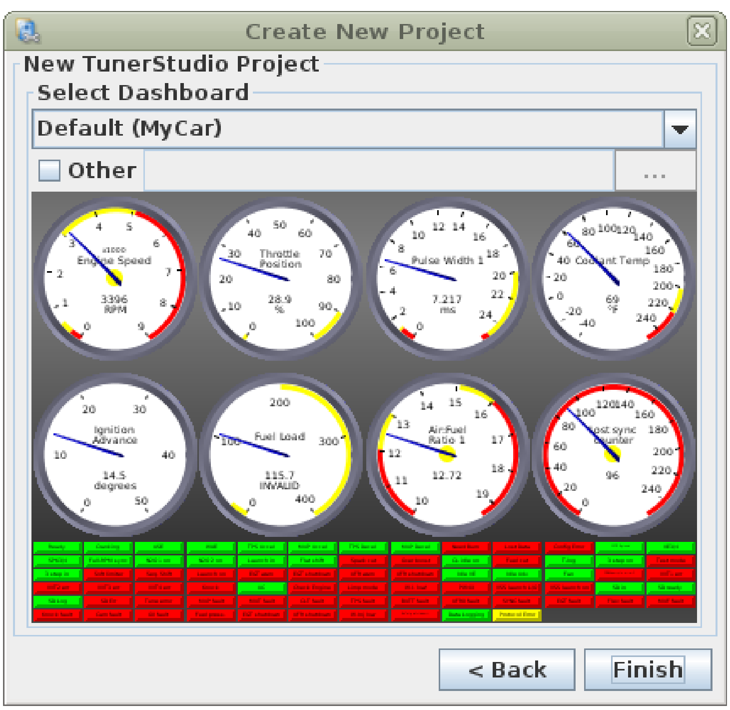
Alter the settings if desired. If you are unsure, the default values are likely to work well. These settings can be changed after your project has been created.

vii. Comms parameters



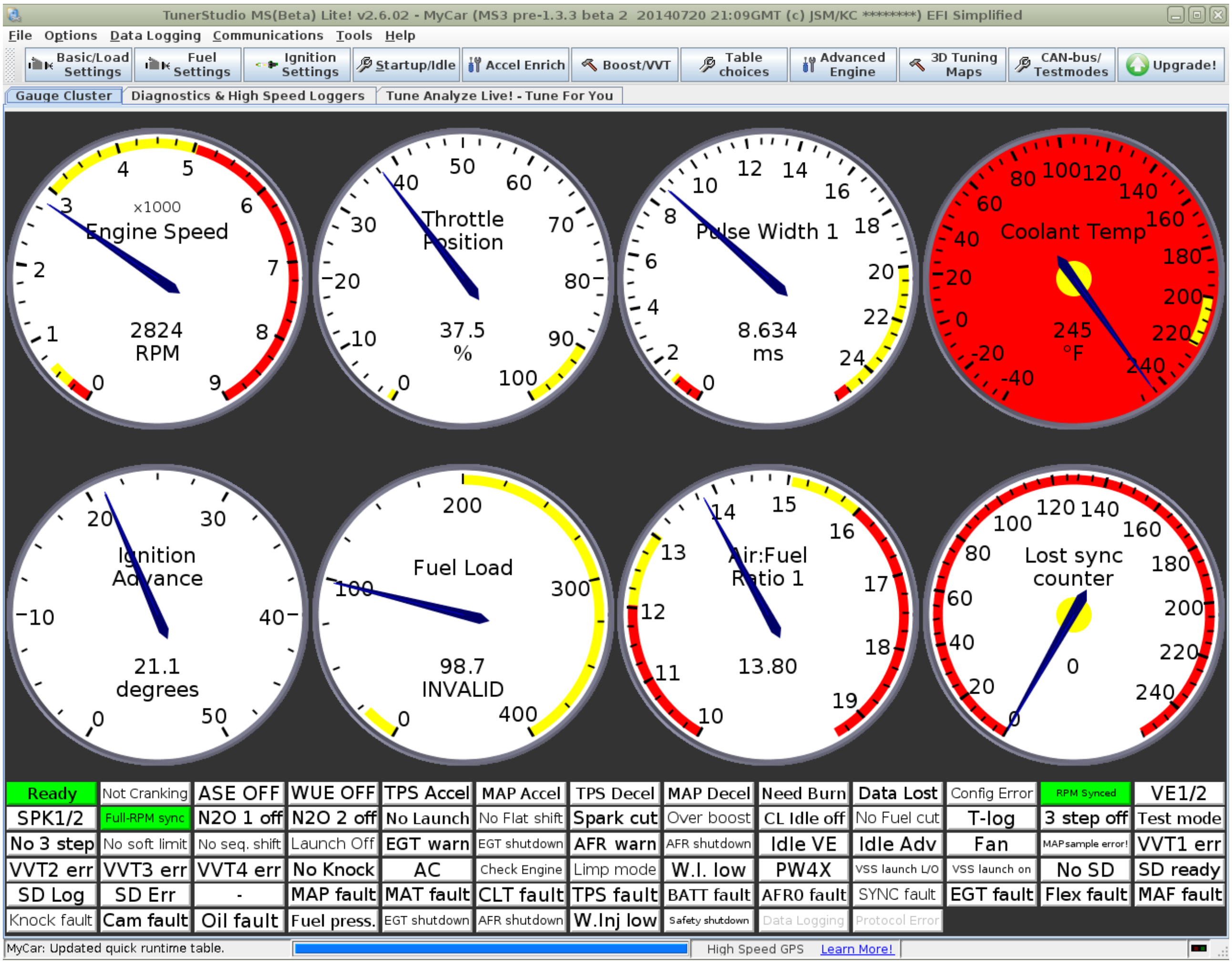
Generally these can be left alone as Detect should have set them. Click Next.

viii. Dash



ix. Click Finish.

The standard dash should display.

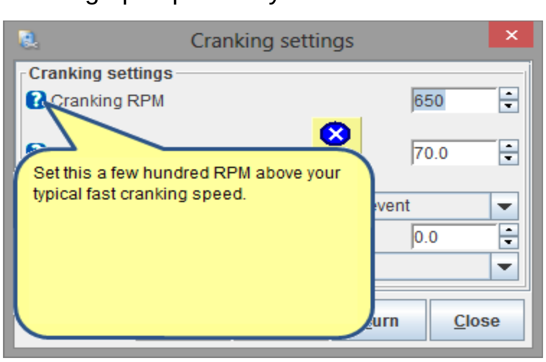


As of TunerStudio v2.6.02 Fuel Load shows as "INVALID" initially. To fix this, right-click on the Fuel Load gauge and re-select "Fuel Load" from Sensor inputs 1.

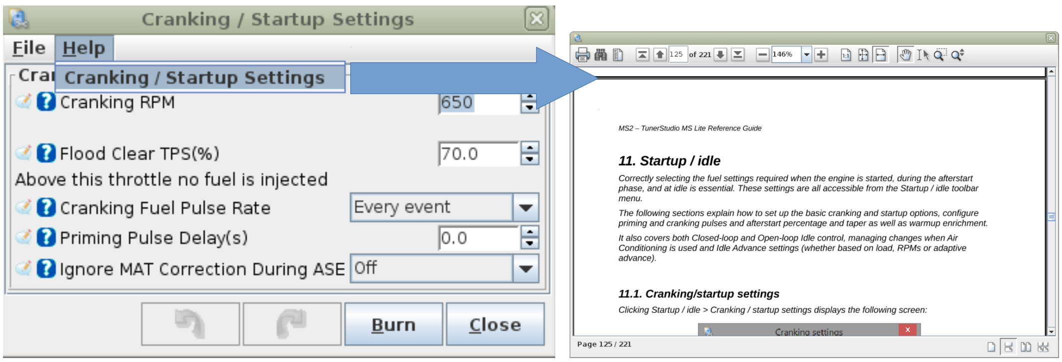
2.10.1 TunerStudio basics

Be sure to read Section 1.5 in the TunerStudio Reference Guide - this covers the user interface in detail.

Two items are worthwhile repeating here. Throughout the software you will find Tooltips - clicking on the [?] will bring up explanatory text.



On most settings screens, you will also find a Help menu - click on this will open up the TunerStudio Reference Guide at the appropriate page so you can read about all of the settings on this particular screen. You may need to be connected to the internet the first time you use this feature, so the PDF can be downloaded.



2.10.2 Firmware, TunerStudio, tune files (MSQ) - what's all that?

The Megasquirt ECU is a self contained computer which runs a program to control your engine. This embedded computer program is called the firmware. Changing firmware versions can give you new engine control features. When you "load firmware" you are erasing the program from the chip and installing a fresh version. This process also wipes out your tune, so be sure to use *Save Tune As* to save a copy of it first!

Your tuning computer runs another program, called TunerStudio, that allows you to tune your ECU. Be sure to understand that TunerStudio is not running the engine! Changing TunerStudio version may give you new tuning or display features, it does not change what can or cannot be controlled on the engine. Typically, just use the latest TunerStudio version, there's no reason not to.

The tune file (MSQ file) is a copy of your engine's tune, saved onto your tuning computer. Typically you can load in tunes that were saved from older versions of Megasquirt firmware. The file format is designed to allow conversion of the data and a relatively smooth transition.

2.10.3 Version numbers

The various hardware and software components within a Megasquirt system have their own version numbers, it is useful to have an understanding of what they are so you can get support for the current product.

2.10.3.1 Mainboard version

The mainboard versions V2.2, V3.0, V3.57 are covered in section 1.4.

2.10.3.2 Firmware Version

This is the most critical version number. The firmware (e.g. ms3.s19) is the computer program that runs inside your Megasquirt and actually controls the engine.

Also supplied with the firmware is the "ini" file (e.g. ms3.ini) which defines all of the tuning parameters and tuning menus for the tuning software.

It is important that you use the "ini" file that matches your firmware. Normally, TunerStudio will automatically detect and figure this out. If it doesn't have the correct file in its local library, it will attempt to download it over the internet. In unusual cases where this doesn't work (e.g. development versions of firmware), you can "browse" and point TunerStudio to the correct matching file.

The title-bar in TunerStudio shows both the TunerStudio version and the firmware version.

2.10.3.3 TunerStudio Version

At time of writing, the current TunerStudio (TS) version is 3.0.16. Most of the tuning user interface (menus etc.) are in fact determined by the firmware version. The TunerStudio version may impact the use of tuning features such as auto-tune or the dashboard displays.

2.10.3.4 MegaLogViewer Version

At time of writing, the current MegaLogViewer (MLV) version is 4.1.10. This will only impact the viewing of datalogs, it does not alter direct tuning or engine operation.

2.10.3.5 Operating system version

e.g. Windows XP, Windows 7, Windows 10, Mac OS X, Ubuntu Linux. Each operating system has its own way to find the version number.

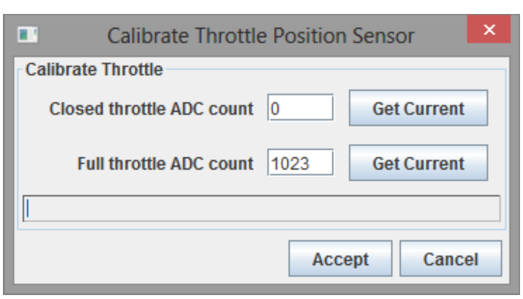
2.11 Setup sensor calibrations to match hardware

Now that the tuning software and ECU are communicating, the next step is to start calibrating the sensor inputs.

These steps are covered in greater detail in the TunerStudio reference manual in section 6.

2.11.1 Calibrate TPS

The Calibrate TPS option allows you to calibrate your throttle position sensor. Clicking Calibrate TPS will display the Calibrate Throttle Position Sensor as shown below:



To calibrate the throttle position sensor do the following:

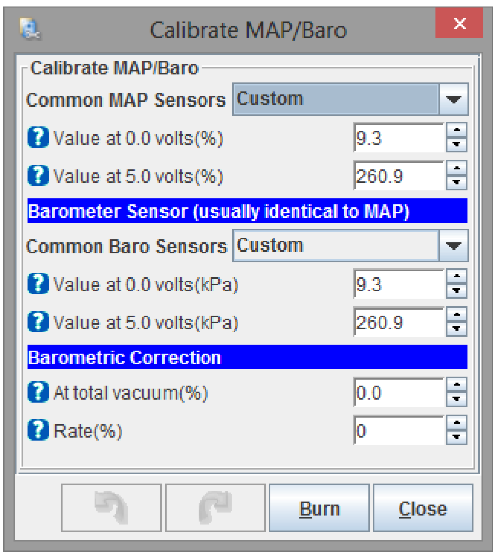
1. Ensure that your Megasquirt is connected and the engine is not running (i.e., KOEO).
2. Ensure that the throttle is closed;

3. Click the “Get Current” button to the right of “Closed throttle ADC count”;
4. Fully open the throttle;
5. Click the “Get Current” button to the right of “Full throttle ADC count”;
6. Click Accept and your throttle sensor will be calibrated.

2.11.2 Calibrate MAP/Baro

The Calibrate MAP / Baro option allows you to calibrate your MAP (Manifold Absolute Pressure) and Barometric Pressure sensors.

Clicking Tools > Calibrate MAP / Baro will display the Calibrate MAP/Baro screen as shown below:



The Common MAP Sensors gives a list of the commonly used MAP Sensors. If you are using one of these sensors select it from the drop down list, otherwise select “Custom” and enter the required numbers.

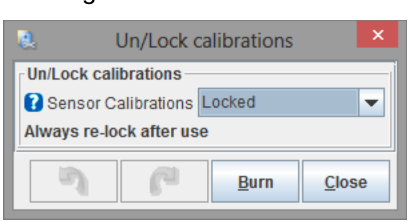
Megasquirt kits are usually supplied with a 2.5 bar MPX4250AP sensor which allows up to 21psi of boost.

Repeat for the Baro sensor settings.

2.11.3 Unlock calibrations

The calibration settings for Thermistor Tables, AFR Tables and MAF Tables can be locked or unlocked to prevent them from being changed accidentally.

Clicking Tools > Un/Lock calibrations will display the Un/Lock calibrations screen as shown below:



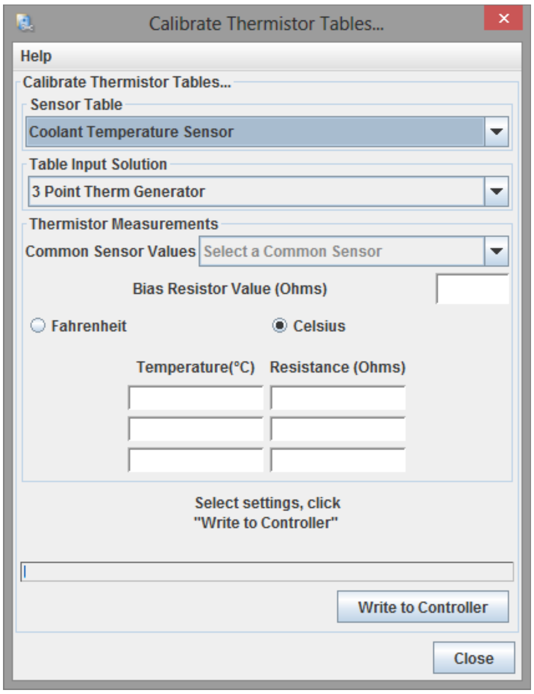
Select Unlocked and then Close.

Ensure that the calibrations are re-locked once you have made your calibrations.

2.11.4 Calibrate CLT and MAT sensor

Megasquirt ECUs are supplied loaded with the correct calibrations for GM temperature sensors. You only need to go through this calibration process if you are using different sensors.

A process for determining the calibration numbers is shown in the Hardware guide section 3.4.



Sensor Table

Select whether you are using a “Coolant Temperature Sensor” or an “Air Temperature Sensor”.

Table Input Solution

This value will normally be set to “3 Point Therm Generator”.

Common Sensor Values

From this option you can select your sensor type. You can select from a predefined list of common sensors from the drop down list, or leave this option unselected if you are using custom settings specific to an unlisted sensor.

Bias Resistor Value (Ohms)

This is 2490 ohms unless you or your vendor have changed the resistors inside the Megasquirt.

Temperature Settings (°C or °F)

Again, if you have selected one of the common sensors these values should be set for you. If you are using a custom sensor enter the three measured temperature and resistance value pairs.

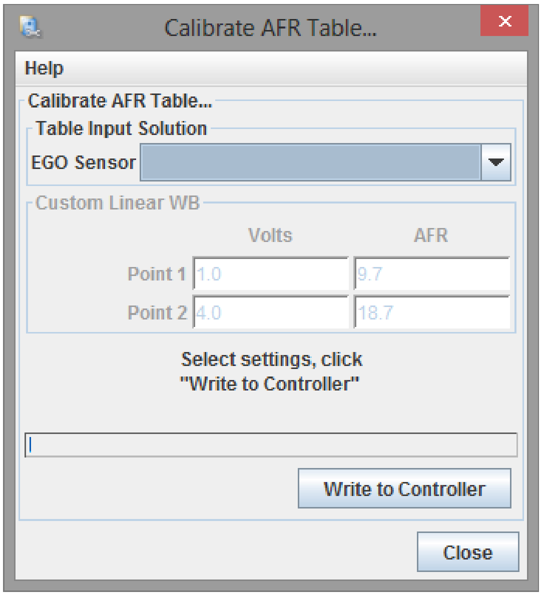
Write to Controller

Once you have completed the settings in this screen click the “Write to controller” button to burn these settings to your Megasquirt.

Repeat for both sensors if required.

2.11.5 Calibrate AFR table (O2 sensor)

Clicking Tools > Calibrate AFR Table will display the Calibrate AFR Table screen as shown below:



From this screen you can set the AFR (air fuel ratio) calibration if needed. The options for these settings are described below:

EGO Sensor

This option allows you to select an exhaust gas oxygen sensor. You can either select from the predefined list, select “Custom Linear WB” (wideband) or select “Custom Inc File”.

Volts / AFR Values

If you have selected an EGO sensor from the predefined list these values will be set for you. If you have selected “Custom Linear WB” then you will need to enter the voltage/AFR pairs from your sensor or controller manual.

Write to Controller

Once you have completed the settings in this screen click the “Write to controller” button to burn these settings to your MS3.

2.11.6 Lock calibrations

Open Tools > Un/Lock calibrations, select Locked and then Close.

2.12 Check all sensor inputs are reading sensibly in TunerStudio.

Now that your have communication and have the sensor inputs calibrated, it is time to check that the sensor inputs make sense.

During these steps you may need to change a gauge on the dashboard within TunerStudio. To change an existing gauge, right-click on it and then select an alternate gauge from the popup menu. You will find the relevant gauges on "Sensor Inputs 1".

2.12.1 MAP sensor check

Ensure that the gauge "Engine MAP" is displayed.

If you are near sea-level a value close to 100kPa should be displayed. At high elevations, expect to see a reading closer to 80kPa. Note: These numbers are with the engine not running.

2.12.2 TPS check

Ensure that the gauge "Throttle Position" is displayed.

The gauge should read 0% when the throttle is closed, smoothly sweeping to 100% when the throttle is fully open.

2.12.3 CLT sensor check

Ensure that the gauge "Coolant Temp" is displayed.

Within a few degrees of the outside temperature is fine. If the sensor is removed from the engine you can heat it up with a typical hair drier and check that the readout changes.

2.12.4 MAT sensor check

Ensure that the gauge "Manifold Air Temp" is displayed.

Within a few degrees of the outside temperature is fine. If the sensor is removed from the engine you can heat it up with a typical hair drier and check that the readout changes.

2.12.5 O2 sensor check

The O2 sensor will not read anything useful without the engine running.

2.12.6 MAF sensor check

Ensure that the gauge "Mass Air Flow" is displayed.

The MAF sensor will read zero with the engine not running, a leaf blower or similar could be used to simulate air-flow to check that it is reading.

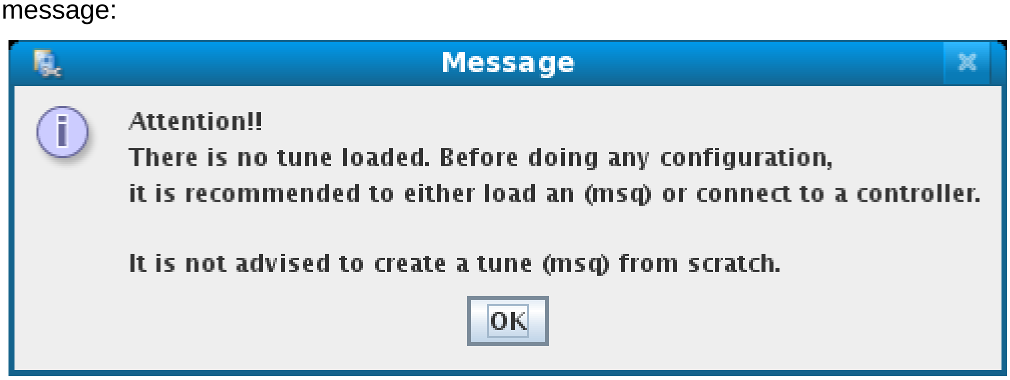
2.12.7 Battery voltage check

Ensure that the gauge "Battery Voltage" is displayed.

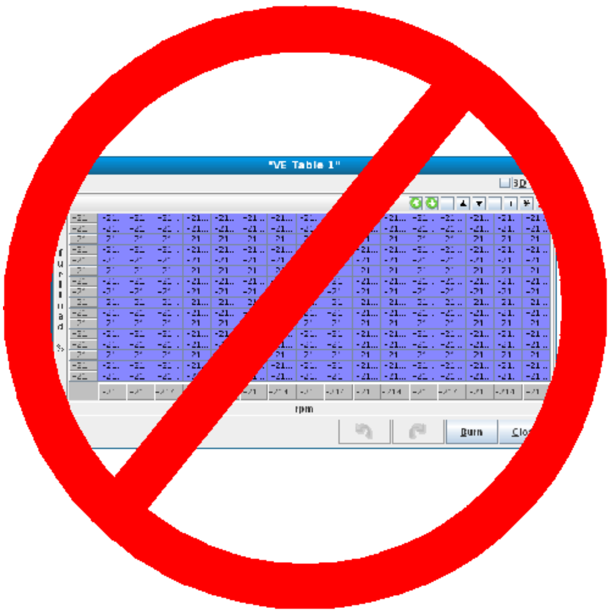
This should display 12-13V.

2.13 Set the base settings

The first rule of tuning is that you do not want to tune from a blank file. You can either connect to the Megasquirt and automatically retrieve the tune on it or open an existing tune file, but stop if you see this message:



This means that you have no tune loaded. This is OK if you are just having a look, but do not try to start a tune like this. TunerStudio will default to a table full of zeroed out values - or in some cases, the numbers will be set to their lowest possible value. It’s not worth chasing down all the nonsense values to turn this “blank file” into a real tune. Start with either the tune in the ECU or one of the sample tunes provided in the software package.



We recommend that you do not have the ignition coils hooked up while doing a basic setup on the Megasquirt. During the setup stage, you will define what type of ignition hardware that you are using, but if the ignition you have doesn’t work with the default settings, you may overheat your ignition module and damage it before you are done with your setup.

The following sections skim through some of the required setting screens - be sure to refer to the TunerStudio Reference Manual which gives an expanded view of each and every setting.

2.13.1 Control Algorithm choice

There are a number of different methods to estimate the airflow into an engine. These depend on the available sensors and their accuracy at predicting flow.

Use Speed-density unless you have a good reason to choose otherwise.

At the most basic level:

- Speed Density uses the MAP (Manifold Absolute Pressure) sensor to determine load. In this case, the vertical axis of any fuel table lookups is in kilopascals (kPa). The maximum value reported by the MAP sensor (in non-turbo applications) will be the same as the barometric pressure.

  Relatively easy to configure, works on boosted engines and familiar to most tuners.

- Alpha-N uses the TPS (throttle position sensor) to estimate engine load.

  The TPS can be a poor estimator of airflow, but it can be useful if a MAP/MAF signal is unavailable or unreliable.

  Be sure to calibrate the throttle range using Tools->Calibrate TPS before using this setting.

- MAF uses an actual mass air-flow sensor and related calculations to directly determine the amount of fuel to inject. In this mode, the VE table is not usually used to determine the amount of fuel to inject.

- ITB uses a combination of MAP and TPS

  This mode was created specifically for naturally aspirated engines running with independent throttle bodies. It combines alpha-n (at high engine loads) with speed density (at low engine loads), using the load calculation that makes the most sense at each RPM. For example, most ITB set-ups do not have good vacuum at idle or low RPM, and slightly touching the throttle makes them lose all vacuum, but at higher RPM start to respond more like a traditional single throttle body engine. This mode allows the use of speed density set-ups at low engine loads and switches to alpha-n at high loads, with an adjustable switchpoint curve over RPM.

- %baro uses a combination of MAP and a barometer reading.

  This setting is similar to the Speed Density setting in that the MAP sensor is used to determine load. However, instead of directly using the manifold pressure, the manifold pressure is divided by barometric pressure to give a percentage of barometric pressure. This setting can be useful for those who regularly drive at high altitudes. It ensures that regardless of barometric pressure, all table lookups operate over 0-100%. For example, if barometric pressure is 80 kPa, and the engine is operating at 50kPa, the actual value used for table lookups is 50kPa/80kPa or 62.5%.

All have pros and cons. Radical engines may need to use a combination of algorithms. Many engines will get good results with Speed Density or MAF.

2.13.2 Engine Types

This sub-section covers specific issues that apply to engines of different cylinder counts, for small engines in particular. Larger car-type engines tend to be simpler to configure.
For all engines you need to configure:

- number of cylinders.
- 2-stroke or 4-stroke.
- even or odd-fire.
- Tach input method.
- Coil type/number (if using spark control.)

  Note that 11, 13, 14 and 15 cylinders engines are not supported.

2.13.2.1 One cylinder engines

Single cylinder engines are by definition even fire.
Two strokes
These have a power stroke every 360 degrees and use a single coil that fires every 360 degrees.
Can use "Basic Trigger" with a single tooth on the crank or "Trigger wheel" with a crank speed wheel.

Four strokes
These have a power stroke every 720 degrees and use a single coil.
Can use "Basic Trigger" with a single tooth on the crank or "Trigger wheel" with a crank speed wheel. Both of these will fire the coil every 360 degrees and require "Wasted spark" to be set.
Unusually a cam-speed trigger could be used to fire the coil every 720 degrees. In this case "Single coil" would be set.

2.13.2.2 Two cylinder engines

Two stroke even fire
If the pistons rise and fall together there will be a power stroke in both cylinders at the same time every 360 degrees.
"Single coil" should be set, but two actual coils are required. Connect both drivers to the same (spark A) logic output.
Can use "Basic Trigger" with a single tooth on the crank or "Trigger wheel" with a crank speed wheel.

If the pistons rise alternately, there will be a power stroke every 180 degrees. Two coils are required.
"COP" should be set. Can use "Twin trigger" with a single tooth on the crank and two pickups spaced 180 degrees apart or "Trigger wheel" with a crank speed wheel.

Two stroke odd fire
Two coils are required.
"COP" should be set. Can use "Twin trigger" with a single tooth on the crank and two pickups spaced the oddfire angle degrees apart or "Trigger wheel" with a crank speed wheel or "Basic Trigger" with two teeth on the crank oddfire angle degrees apart and a single pickup.
It is required to specify the "oddfire small angle" .

Four stroke even fire
Typically a single coil will be used firing in wasted spark.
"Basic Trigger" or "Trigger Wheel" and "Wasted Spark" .
It is also possible to use an additional cam trigger to run true COP.

Four stroke odd fire
Two coils are required, usually firing in wasted spark.
"Basic Trigger" or "Trigger Wheel" and "Wasted Spark" .
It is required to specify the "oddfire small angle" .
It is also possible to use an additional cam trigger to run true COP.

2.13.2.3 Three cylinder engines

Two stroke even fire
A single coil output fires every 120 degrees. Three coils should be connected to it.
"Basic trigger" or "Trigger wheel" can be used.

Four stroke even fire
"Trigger wheel" required. With a crank wheel alone, "Wasted Spark" should be set and three coils will fire every 120 degrees.
With the addition of a cam trigger or a trigger wheel running at cam speed, true COP is possible, firing every 240 degrees.

Odd fire
Not supported at this time. If you want support added please contact the developers.

2.13.2.4 Four cylinder engines

Even fire and odd fire supported.
The even-fire inline four cylinder is the most common configuration for car engines.
Bike engines may be oddfire, the code supports alternating or banked (V-max) odd-firing.
Numerous special wheel decoders exist for OEM specified wheel patterns.
Choose "Single coil", "Wasted Spark" or "COP" to match your engine.

2.13.2.5 Five cylinder engines

Even fire only

Some Audi applications use a triple sensor setup and trigger off the flywheel teeth. This is experimentally supported.
A five window distributor will work with "Basic Trigger" and a single coil.
Or use a crank trigger wheel, a single window distributor and five coils for true COP. (Note that the number of crank teeth needs to be a multiple of five. So 60-2 will work, but 36-1 will NOT.)

2.13.2.6 Six cylinder engines

Even fire and odd fire supported.
Chevy V6 may be even or odd, typically current V6s are even.
Some special wheel decoders exist for OEM specified wheel patterns.
Choose "Single coil", "Wasted Spark" or "COP" to match your engine.

2.13.2.7 Seven cylinder radial engines

Not for airborne use!
7 cylinder radial engines are supported.

2.13.2.8 Eight cylinder engines

Even fire V8s are very common.
Some special wheel decoders exist for OEM specified wheel patterns.
Choose "Single coil", "Wasted Spark", "COP" or "Dual dizzy" to match your engine.

2.13.2.9 Nine cylinder radial engines

Not for airborne use!
9 cylinder radial engines are supported.

2.13.2.10 Ten cylinder engines

Even fire and odd fire supported.
Choose "Single coil" or "Wasted Spark" to match your engine.
Fully sequential fuel and spark supported with custom hardware additions.

2.13.2.11 Twelve cylinder engines

Even fire supported.
Choose "Single coil", "Wasted Spark" or "Dual dizzy" to match your engine.
Fully sequential fuel and spark supported with custom hardware additions.

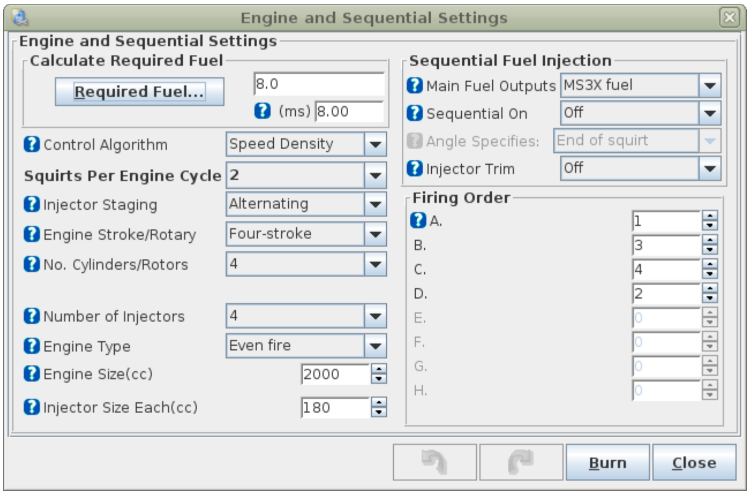
2.13.2.12 Sixteen cylinder engines

Even fire supported, but untested.
Choose "Single coil", "Wasted Spark" or "Dual dizzy" to match your engine.
Fully sequential fuel not allowed.

2.13.2.13 Wankel Rotary Engines

Supported. See section 4.12

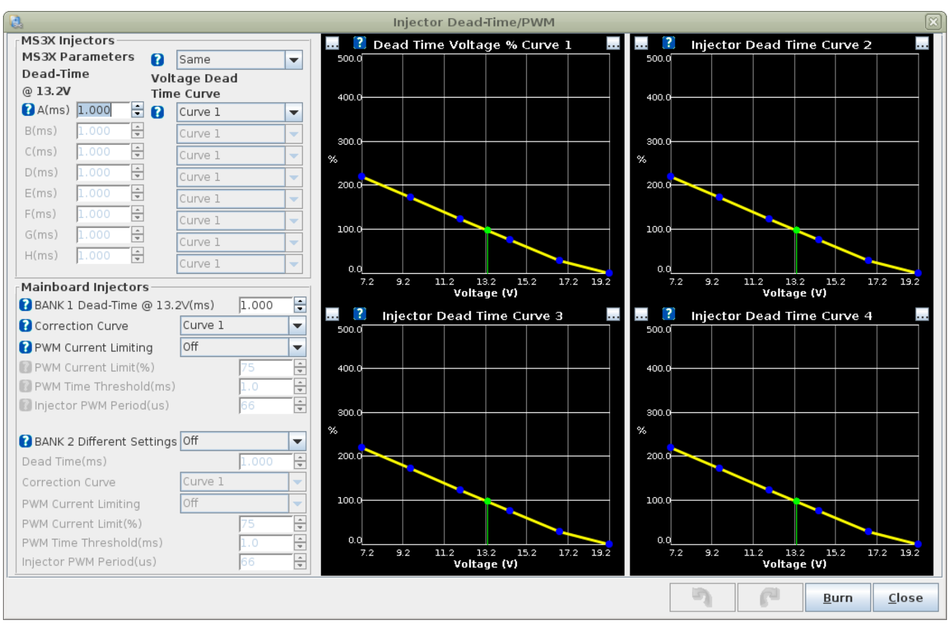
2.13.3 Engine and Sequential settings



The most critical settings are on the left. Complete the Control Algorithm through to Injector size settings. If unsure, leave alone!

Then click on the "Required Fuel" button to calculate the "ReqFuel" number.

2.13.4 Injector dead-time



It is important to select the correct dead-time setting for your injectors and hardware.

Typical dead times at 13.2 volts:

- High impedance injectors - dead time = 0.8-1.0 ms
- Low impedance injectors with resistors - dead time = 0.8-0.9 ms
- Low impedance injectors with PWM (mainboard only) - dead time = 0.6-0.8 ms
- Low impedance injectors with external peak/hold - dead time = 0.6-0.8 ms

Technical explanation of dead-time

Electrical fuel pulse turns on (a).

After a delay, the injector starts to open (b).

Flow ramps up to maximum (c).

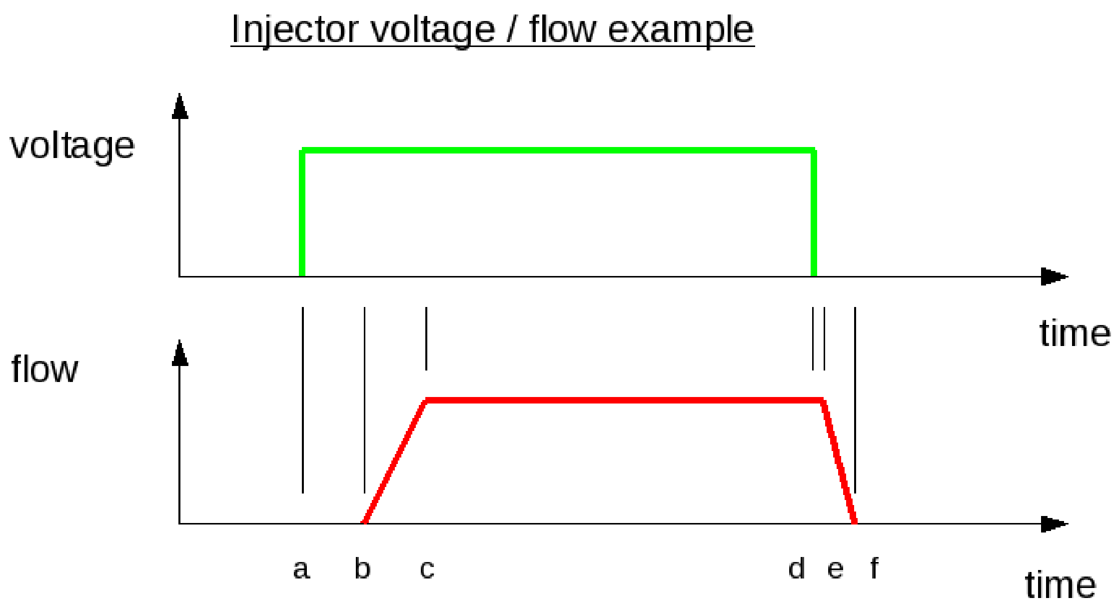
At (d) the electrical pulse ends, at (e) the injector starts to close.

Fuel ceases at (f).

One manufacturer quoted an opening time of 1.2ms and a closing time of 0.4ms. This gave a dead-time of 0.8ms.

It is the dead-time that we care about for correct fuelling.

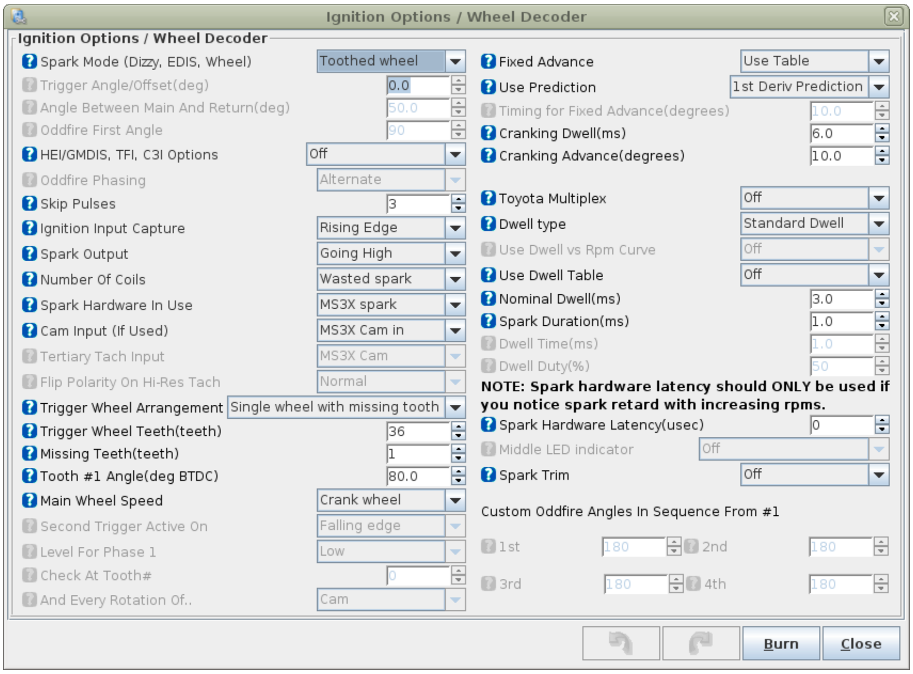
Say the ECU wants to inject 7.000ms of fuel. Allowing for the dead-time, the required electrical pulsewidth in this example would be 7.000 + 0.800 = 7.800ms.



2.13.5 Ignition setup

Wiring and configuring your ignition setup covers the tach input (how Megasquirt gets an RPM signal) and how the ignition coil or coils are controlled.

The exact details depend on which Megasquirt-3 product you have, so this section is covered in the hardware guide for your product.

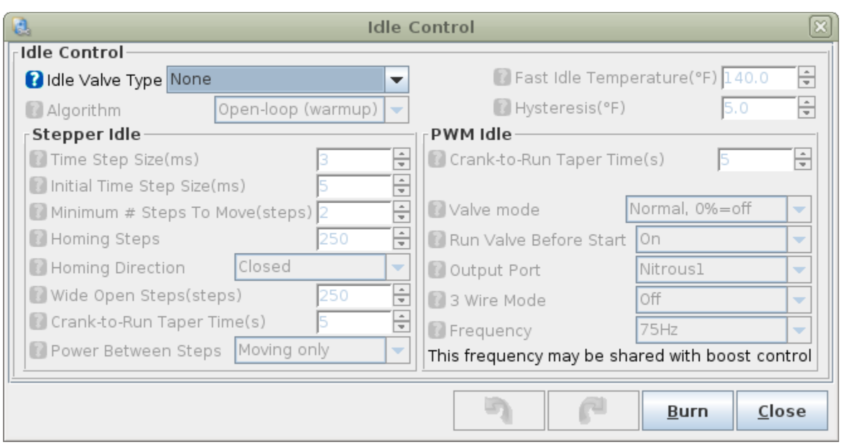


Each setting needs to be reviewed and set appropriately, however, if unsure, leave alone!

2.13.6 Idle Valve

The specific wiring for your idle valve is covered in the hardware guide for your product.

During initial startup it is strongly advised that the open-loop (warmup) algorithm is selected. Do not try to use closed-loop until the engine can maintain a steady idle speed on its own.



Stepper idle users need to configure the number of homing steps - this is covered in the section on idle valve testing - 2.14.5. For now, set the homing steps to 250 and ensure that all values in the warmup curve are less than 180.

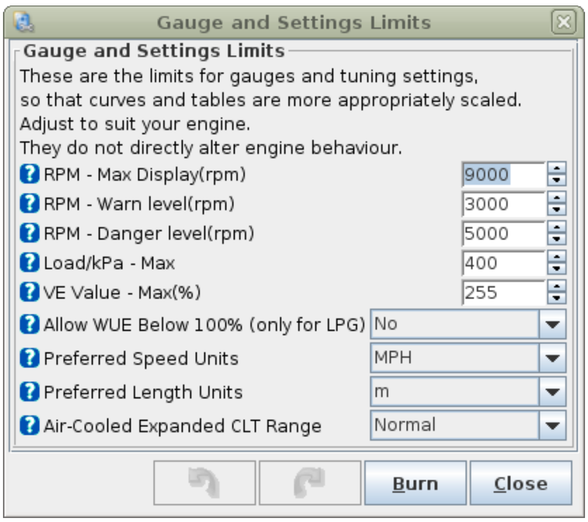
For both PWM idle and stepper idle, a larger duty or steps number means more air i.e. higher idle. The warmup curve will start at larger numbers on a cold engine and smaller numbers on a hot engine.

2.13.7 Other inputs/outputs

During the initial startup it is advised that other, optional inputs and outputs are left disabled (shift lights, fan control, launch control, nitrous control etc.) - so long as they are not required for engine operation.

2.13.8 Gauge/Setting Limits

By default, the RPM settings within TunerStudio are set to display an RPM range suitable for most car engines. If your engine has a higher RPM range, be sure to adjust the Basic/Load Settings -> Gauge and Settings Limits to set the limits that suit your install.



Changing these settings does not alter the behavior of your engine in any way, they ONLY change the limits which can be displayed on the screen and set the threshold limits for your tune.

Once these settings have been changed it is necessary to close and restart TunerStudio for the changes to be reflected in the gauge cluster.

2.13.9 Power Cycle Required

When you change some settings or enable a new feature, this message will pop up.

Typically, you should complete the settings changes on that page and Burn. Then turn the power off (key off), pause, then power on (key on.)

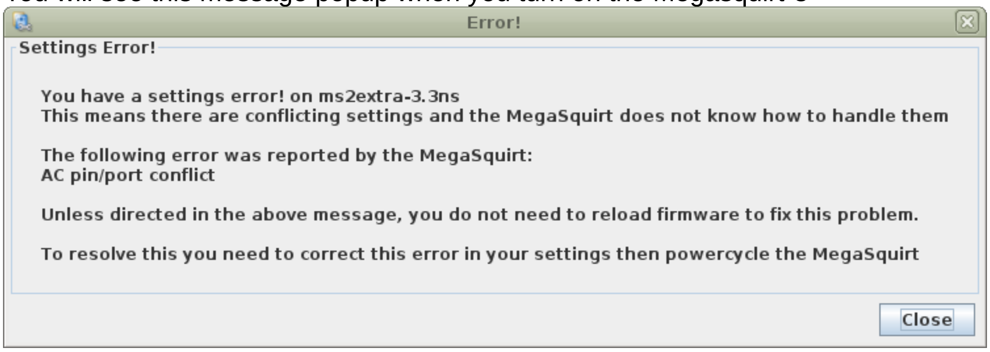
If you are making multiple changes during initial setup, you can wait until you have made all of the changes before cycling the power.

Make sure that you clear this warning message before trying to start the engine.

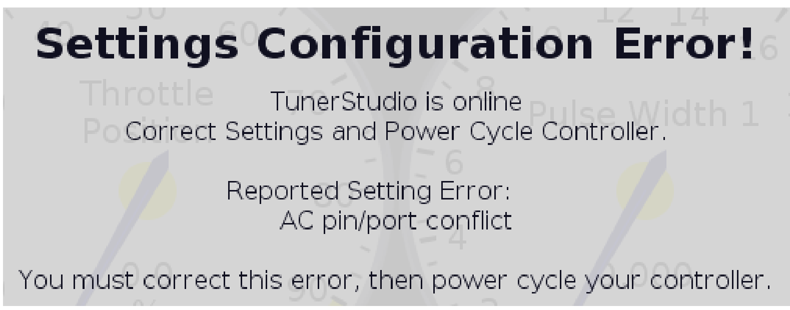
2.13.10 Configuration Error

During the configuration of your Megasquirt-3 you might make a mistake and set an impossible combination of settings. Common mistakes are selecting the same output to two different features or trying to enable an ignition output type that is not possible with your tach input.

You will see this message popup when you turn on the Megasquirt-3



and this message will show across the Gauge Cluster.



The other tell-tale signs are that a bogus high RPM is reported

and a config error indicator lit up.

The final warning message is the fuel pump - at power-on the pump will run three times.

How to fix it?

1. Read the message - it tells you what to do.

2. If you have a pin conflict, take a look at Basic/Load Settings -> Feature List Showing I/O Pins

you should be able to spot where you have set two features to use the same connection.

3. Change the setting that you are being warned about.

4. If you are really stuck, then load in an old tune file (MSQ) that you previously saved using "Save Tune As".

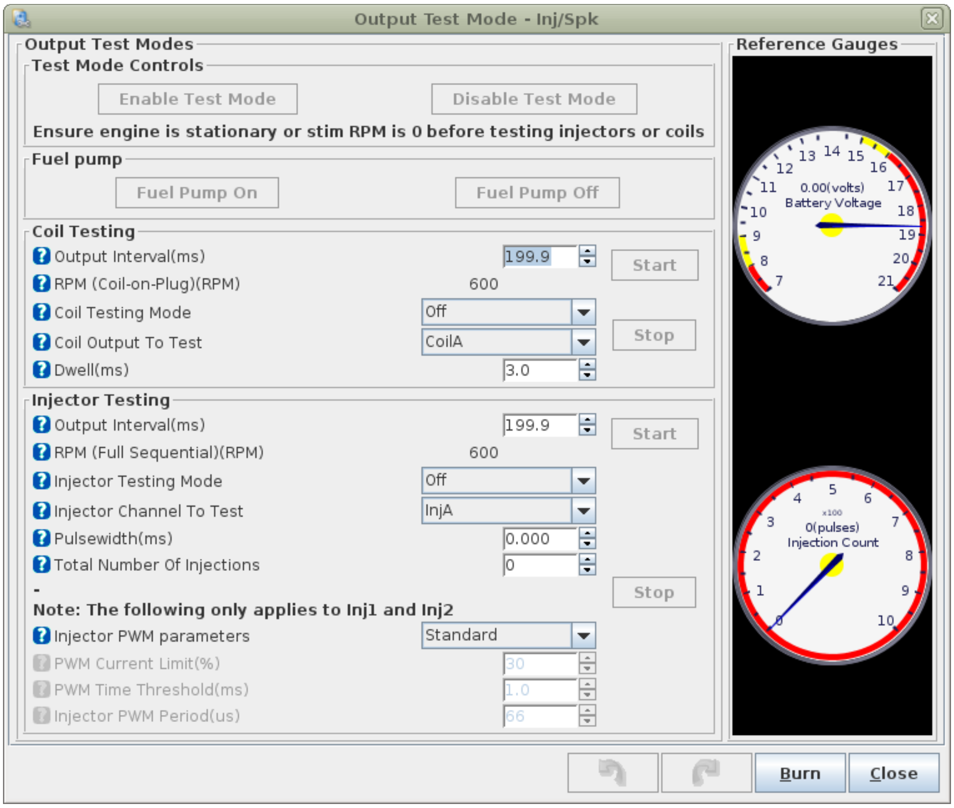
This message has NOTHING to do with wiring or any internal fault.

It is solely a setting mistake that you have made.

2.14 Use the test mode to confirm injectors and coil(s) are functioning

Output Test Mode (select it under the CAN bus / Test Modes menu) is used to check the outputs. The test mode allows direct control of the coils and injectors. Used by accident, the test mode could cause flooding of the engine and potential damage. **Test mode should NOT be used on a running engine.**

Full details are available in the TunerStudio Reference Guide.



The Enable Test Mode button is locked out unless the RPM reading is zero. Click this button to enter test mode, and when you are done, click the Disable Test Mode to go back to normal operation.

2.14.1 Injector testing

The injector test mode can be used both to confirm the injectors are wired correctly and to use the Megasquirt as a controller for an injector test bench. If your injectors are fitted to the engine, you must not run the test mode with fuel pressure or you will fill your engine with fuel! Remove the fuse from your fuel pump and de-pressurize the line first.

To enable injector testing:

- Select Injectors in the Test mode drop down.
- Choose your pulsewidth time in milliseconds. (To just make them click for this test, 3 to 5 ms is fine.)
- Set the interval (e.g. 10ms)
- Choose which injector to test.
- Set Injector testing mode to One.
- If your injectors are powered via the fuel pump relay as is recommended, set the fuel pump to On.
- For functional testing, set the total number of injections to 65535 (the maximum) or for flow pulsewidth testing use your chosen number.
- Click Start.

2.14.2 Coil testing

The coil test mode is useful to confirm you have the coils wired correctly and to check dwell settings. (Do be careful as ignition coils put out a potential lethal voltage. Typically 30,000V or more.)

Often it is best to remove the spark plugs from the engine and lay them onto something conductive and grounded, such as the top of the engine.

To enable coil testing:

- Select coils in the Test mode drop down.
- Choose your dwell time in milliseconds. Start low. e.g. 2.0ms.
- Set the interval (e.g. 50ms).
- Choose which coil to test.
- Set Coil testing mode to One.
- If your coils are powered via the fuel pump relay as is recommended, set the fuel pump to On.
- Click Start.

With wasted spark or coil-on-plug installs, make sure that "Spark A" operates the spark plug for cyl#1.

Note that coil testing will not work with EDIS, TFI, HEI7/8 or GMDIS as the external module controls the coils internally.

2.14.3 Fuel pump testing

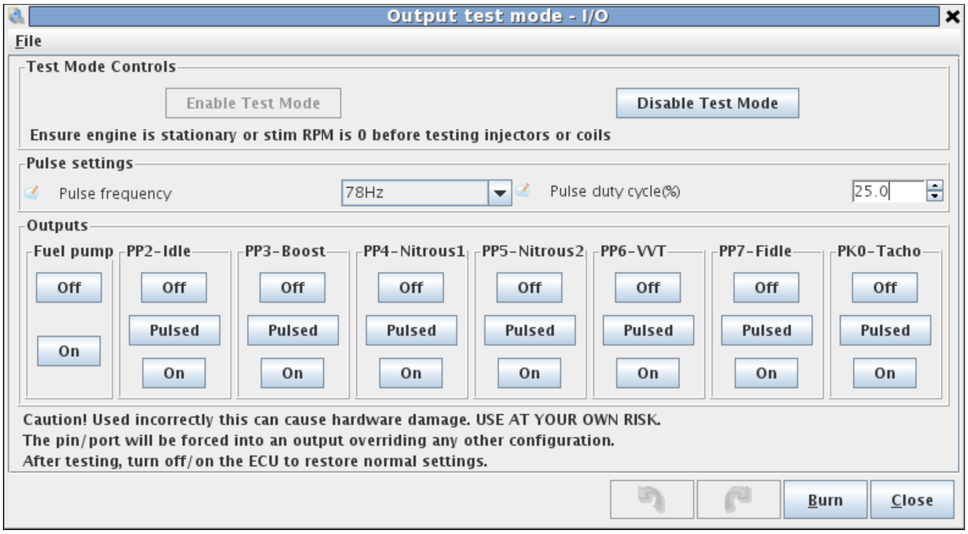
Testing the fuel pump is straightforward; just click the buttons under Fuel Pump to turn it off and on. Reinstall the fuse that you removed before injector testing. You will notice that "Ready" lights up on the TunerStudio dash when the fuel pump is enabled. If you get no activity, try starting again. However, if Ready lights up but the pump doesn't run, you need to check you fuel pump relay and pump wiring. If this is a new fuel system, make sure that the fuel system doesn't leak and that it holds pressure. Note that some aftermarket fuel pressure regulators will hold pressure for only about 30 seconds or so after shutdown even if the fuel system has no leaks. Most OEM fuel pressure regulators will hold pressure for much longer.

If you have a pressure gauge, check the static pressure (engine off.) For most port-injection systems this should be around 3 bar (43.5psi.) Throttle-body injection systems may require a lower pressure.

2.14.4 Outputs testing

This allows you to test some of the on/off and pulsed outputs in a controlled manner.
As with the other tests, be cautious - this mode allows you complete control over the outputs - **it is up to you to ensure that doesn't cause any damage.**

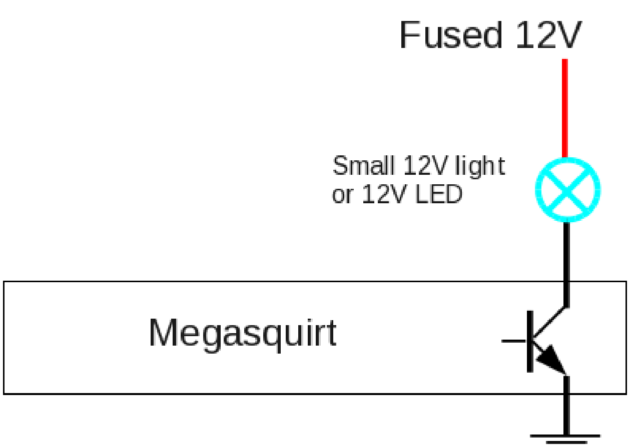
This test mode can be found on CAN Bus/Testmodes > Output test mode - I/O



First ensure that you are key-on, engine-off. (Or that RPM is set to zero on the stim if bench testing.) In this condition the 'Enable Test Mode' button will be available. Click the button to enable the test mode.
For each of the outputs you can select

- **Off** - turns the selected output off.
- **On** - turns the selected output on.
- **Pulsed** - pulses the output at the selected frequency and duty.

Remember that all of the power outputs from the Megasquirt are ground switching, so typically you will not be able to measure any voltage at the output with a meter. Use a small test lamp or test LED connected between 12V and the output.

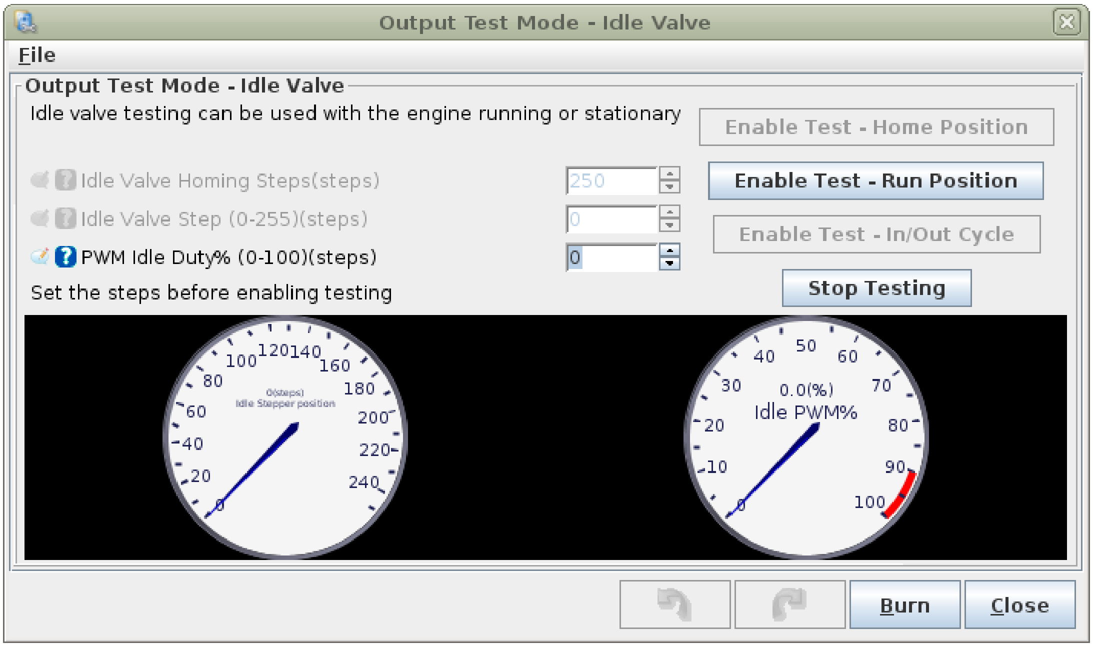


After using this test mode you need to turn the Megasquirt off to reset all output settings.

2.14.5 Idle Valve Testing

The idle valve test mode can be used with the engine running or stationary. Note that with a PWM idle valve you may need to run a temporary fused 12V supply to the valve if you normally take power from the fuel pump relay and are testing with the engine off. You need to have enabled idle control before this is available.

This test mode can be found on CAN Bus/Testmodes > Output test mode - Idle valve



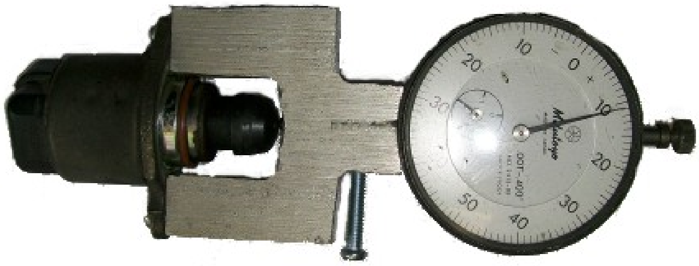
Idle Valve Homing Steps - *This only applies to stepper idle valves.* This setting determines how many steps the motor should move during homing to ensure it has reached the home position. The number needs to be large enough to close the valve from any unknown starting position. (This is because there is no position feedback - in non-automotive control applications there is often a feedback encoder.)

- Idle Valve Step - the desired valve position - larger numbers should give a more open valve and higher RPM. (For stepper motors)
- PWM idle duty% - the desired valve position - larger numbers should give a more open valve and higher rpm. (For PWM valves)

(Note that some Bosch rotary PWM valves valves feature a fail-safe mode so that zero flow is at say 20% duty, above that the valve operates as expected where more duty gives more airflow. Below this specific point is a fail-safe mode that allows airflow even if there is no control signal. This is readily observed during test mode.)

There are four control buttons:

- **Enable Test - Home Position** - for stepper idle valves, this moves the valve through the number of homing steps back to the home position. (If it doesn't get all the way back to the stop, increase the number of homing steps.) The valve will move to the run position shortly afterwards.
- **Enable Test - Run Position** - enable test mode where the steps or duty setting controls the valve position
- **Enable Test - In/Out Cycle** - moves the stepper idle valve in and out continuously.
- **Stop Testing** - disable test mode and allow normal idle control to operate.



Here is an example of a fixture used to measure stepper operation on the bench, although very few users will need one of these.

2.14.5.1 Using the test mode to determine stepper homing steps

In the initial setup, rough values were given for the homing steps and maximum number of steps to use in the curves. Once the engine will start and run and after you have performed some initial tuning, you can return here to validate these settings. The engine may reach 3000RPM or so with a wide open idle valve, so do not attempt this procedure until you are ready to do that. This procedure assumes that you are homing to the closed direction.

- Start the engine and run up to temperature.
- Open CAN Bus/Testmodes > Output test mode - Idle valve.
- Set 'Idle Valve Homing Steps' to 300.
- Click 'Enable Test - Home Position' (You may need to open the throttle to prevent a stall.)
- The idle valve should move to the fully closed position and the revs will drop.
- Set 'Idle Valve Step' to 100.
- Click 'Enable Test - Run Position'
- The idle valve will open somewhat and the revs will increase, note the revs.
- Set 'Idle Valve Step' to say 150.
- The idle valve will open some more and the revs will increase, note the revs.
- Continue to increase the steps number until the revs stop increasing.
- Re-home the valve and check 10 steps either side.

The steps number you have found is the number of steps from fully closed to fully open. Say it was 185. The "homing steps" setting on the idle control page should be set around 10 steps higher than this - to ensure that the idle valve always reaches the closed stop fully at key-on. Ensure that your warmup curve or closed-loop idle settings are a maximum of around 10 steps less than this number.

If you mistakenly try to open the valve more steps than full open, the overrun clutch in the stepper idle valve will prevent damage as the pintle hits the full open stop, but the ECU will lose track of where the valve is. This would most likely result in random idle speeds depending on what temperature you first started the engine.

2.15 Check for crank/cam tach-in signals

Before cranking the engine:

- pull the fuses for fuel pump, injectors and coils

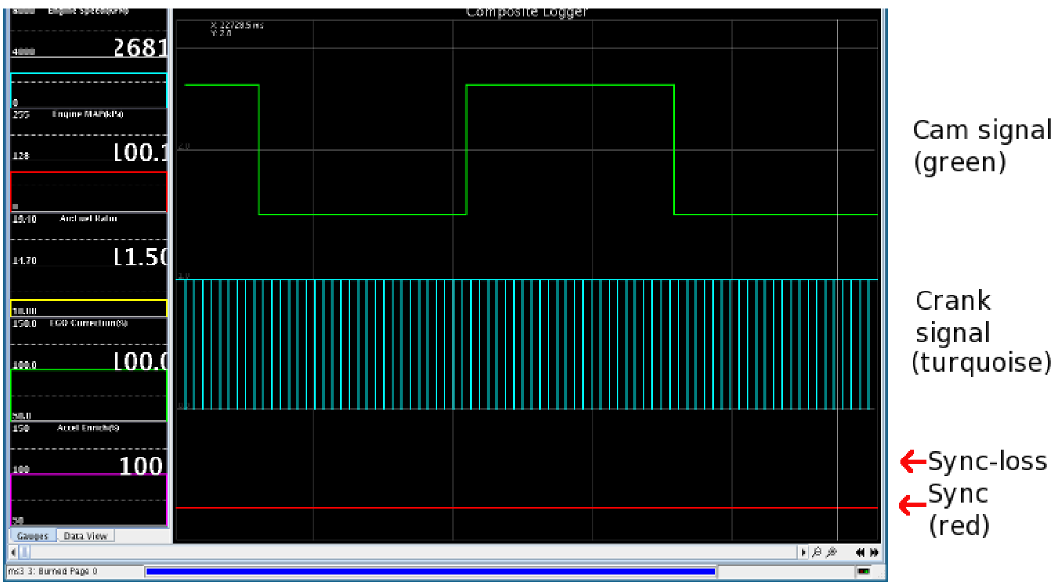
- check that the engine has sufficient oil of the correct grade
- check that any trans cooler lines are connected.
- check that the transmission is in neutral
- check parking brake is firmly latched on

We do not want the engine to start yet.

Key-on the engine, then open up the "Diagnostics & High Speed Loggers" tab, then select the "Composite Logger" from the drop-down and click Start.

Crank the engine for 10-20 seconds.

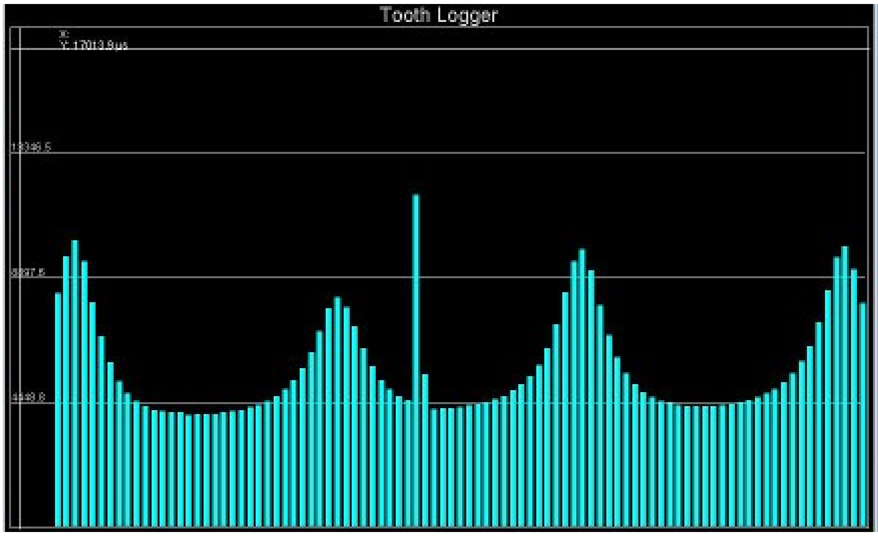
You should see a pattern show up looking something like this:



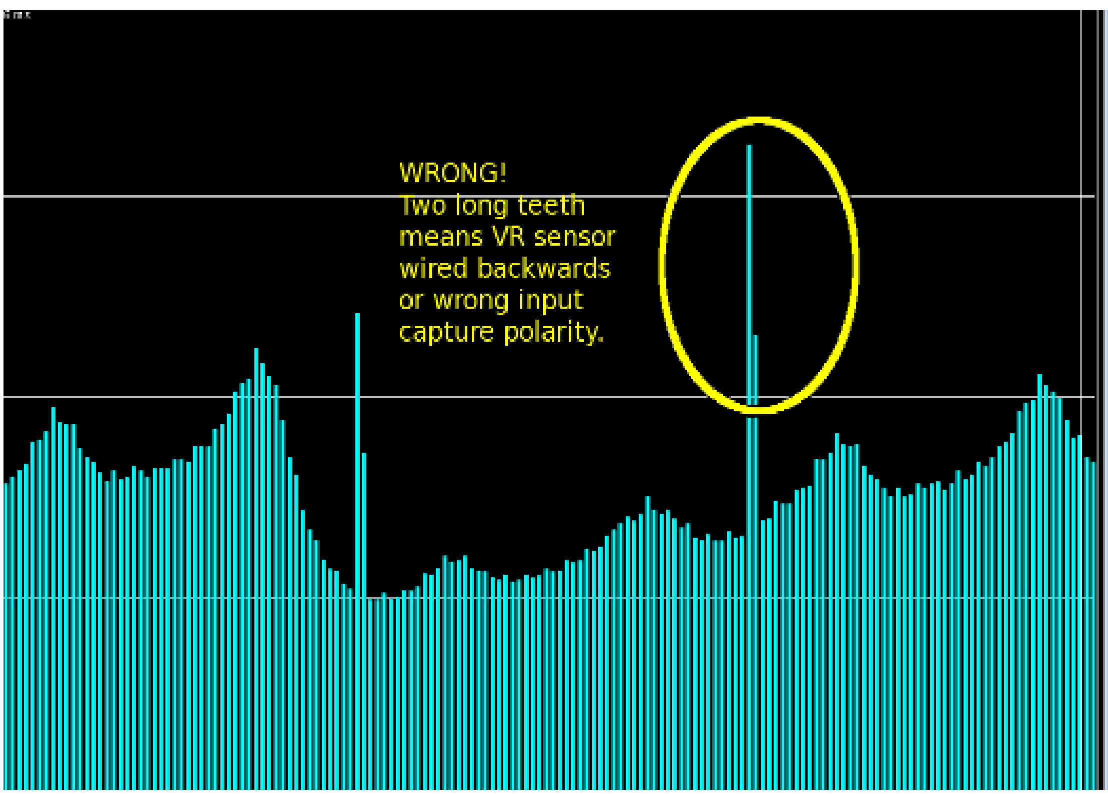
This example includes a cam signal. If your install only uses a crank signal then the green trace will correctly be missing.

If the turquoise "crank" signal is missing, you have a problem and need to fix it.

Next, repeat the process using the "Tooth Logger"



The screenshot here shows a real 60-2 crank pattern at cranking speed, the "missing tooth" is the single long bar in the middle. The cyclic pattern is due to the compression effect as the spark plugs were in.



Here is an example of an incorrect pattern - the missing tooth shows as two long bars instead of one. This won't work. Swap the +/- on the sensor or change the "Ignition Input Capture" from Rising to Falling or vice-versa.

These are just a few examples or the patterns you might see. The Megasquirt product range supports a large number of different trigger input patterns which all look different. Count the number of bars (including the long one) in the repeating pattern on the tooth logger and see if it looks reasonable. 36-1 wheels will have 35 bars total, 60-2 will have 58 total.

2.16 Check for RPM input

Once you have completed the previous step and have the "crank" and "cam" (if used) inputs showing something sensible you can check for RPM.

Switch back to the "Gauge Cluster" tab in TunerStudio.

Once again, before cranking the engine, be sure to pull the fuses for fuel pump, injectors and coils.

Crank the engine and watch the "Engine Speed" gauge and indicators section.

You need to see sensible and stable RPM and the green "RPM Synced" indicator. Expect to see RPMs of 200-600RPM depending on the engine and outside temperature.

If the RPMs are bouncing up and down (e.g. 400, 0, 400) or the RPM Synced indicator is flashing between green and red you have a problem. Refer to section 5.2 for information about common causes for sync-losses.

Do not proceed any further until you have stable RPM and RPM-sync.

2.17 Check sensors during cranking

With the RPM now reading OK, it is worth checking that sensors are still reading correctly during cranking.

Once again, before cranking the engine, be sure to pull the fuses for fuel pump, injectors and coils.

Crank the engine and watch the "Coolant Temp" and "Manifold Air Temp" gauges. The gauges should not change during cranking. If they do change significantly, you almost certainly have a sensor ground wiring fault. Fix that before continuing.

During cranking the "Engine MAP" gauge should drop from around 100kPa to say 80kPa. This will depend on the engine, altitude and throttle position. Some drop is required. If it doesn't move, make sure that the hose is connected from the sensor to the intake manifold. If you aren't using the MAP sensor (Alpha-N or MAF) then you can skip this.

2.18 Sample datalog

With the RPM now reading OK, it is worth checking that sensors are still reading correctly during cranking.

Once again, before cranking the engine, be sure to pull the fuses for fuel pump, injectors and coils.

Start a datalog (Datalogging -> Start Datalogging)

Crank the engine for about 10 seconds.

Stop the datalog (Datalogging -> Stop)

Open the datalog to view (Datalogging -> View with MegaLogViewer)

In Megalogviewer, the field selectors are on the left, ensure that you have MAP, TPS, RPM, CLT, MAT, BattV enabled.

The log should start with BattV at about 12V, MAP at about 100kPa, TPS around 0%. CLT, MAT should reflect outside temperature, RPM will be 0.

At the start of cranking you will see the BattV drop to 8-10V, along with a small drop in MAP. Then the RPMs should register a reading. CLT, MAT, TPS should not change.

Once you have had a look, you can close MegaLogViewer.

2.19 Check the cranking timing with fuel disabled

Having confirmed that the coils work as expected and that we can get stable RPM it is time to check the timing. You may need an assistant.

Yes, this step is really needed - **correct timing is very important.**

(If you feel like skipping this step, please ask yourself whether you would rather check your timing now or rebuild your broken engine soon.)

If you are running wasted spark you need to ensure your strobe is compatible. "Dumb" timing lights are fine; some dial-back lights can be awkward. Use a "2-stroke" setting for wasted-spark or wasted-COP if you have a dial-back light. With a coil-on-plug pencil type coil you will need to remove the coil from the engine and install a short length of regular spark plug wire from the COP to the spark plug, and possibly ground the COP if it was bolted on. Connect your timing light over this temporary plug lead.

If there are no timing marks on your engine, you will need to add some sort of marks and a pointer. You need to mark where the pointer lines up with the damper when the engine is at TDC#1. You can establish TDC with a piston stop and add timing marks with either paint or timing tape.

Once again, before cranking the engine, be sure to pull the fuses for fuel pump and injectors. Re-install the fuse for the coils.

Temporarily set the cranking advance (Ignition Settings -> Ignition Options / Wheel Decoder -> Cranking Advance) to 0.

Crank the engine and confirm that #1 timing is somewhere close TDC.

(In Fuel Only mode, Megasquirt isn't controlling your timing, so you will need to adjust whatever is controlling it as required.)

Adjust the Tooth #1 Angle setting (if running Toothed Wheel) or Trigger Angle / Offset (if running any other spark mode besides Fuel Only). If the timing is too far advanced, increase this number. If the timing is retarded, decrease this number. If adjustments of more than 10 degrees are required, make the change, click Burn, turn the key off and check again. Very large changes don’t take effect until you turn the Megasquirt off and back on again.

Once the timing lines up during cranking you should return the Cranking Advance setting to its previous value (e.g. 10 BTDC) and Burn.

During this step you should also check that the engine is developing oil pressure.

2.20 Start the engine and start tuning

Now that you’ve confirmed sensor inputs, coil and injector outputs, have good RPM and cranking timing is close you are ready to start the engine.

Double check the oil level and that the coolant system is filled.

Replace the fuses for fuel pump, injectors and coils (since we removed them in previous tests.) Ensure that the battery is charged, spark plugs are fitted, plug leads are in place and all hoses are secure. We recommend starting a data log as well. While the engine will usually start if you’ve gone through all the previous steps for setting up, sometimes you’ll need a couple more tweaks, and the data log can be very valuable if it comes to that. Go to the Data Logging menu and select Start Logging. The log starts when you click Save, and stops when you either close TunerStudio or go to the Data Logging menu and select Stop.

Now, hit the starter, and hope the moment for the big pay-off has arrived.

It is common to need to give the engine a little throttle to keep it alive on the first start. If all sounds well, then keep the engine running and warm it up. Check for oil pressure, coolant temperature etc. If anything sounds wrong - stop and investigate.

Take your time! If things aren’t working out, then take a breather and come back another day. Come to the www.msextra.com forum and ask for help. When posting, please include full details of your install: engine type, make, model, ignition setup. The forum is global and we might not be familiar with your engine, so just quoting an engine code is not sufficient. Don’t use the forum to vent your frustration, keep it calm...

If your engine does not start in spite of having a functioning fuel and ignition system and timing that appears to be correct, here are several things we recommend checking that commonly will cause a no start condition.

- If you are running full sequential fuel and / or coil on plug ignition, try changing these settings to semi-sequential fuel and wasted-COP ignition. This will rule out the possibility that the cam sensor hasn’t been set up correctly, and you’re firing on the exhaust stroke instead of the intake stroke. These changes will make it fire on both, which can be very useful for debugging startup problems.
- It may be refusing to start because the fuel isn’t quite right. If the engine does not go above start-up cranking RPM, try adjusting the cranking pulse width (the Cranking Pulse menu under Startup / Idle). If the engine catches but dies immediately, try adjusting the afterstart enrichment (ASE Percentage under Startup / Idle). Both of these are temperature dependent, so be sure you’re adjusting them at your actual coolant temperature. There’s a couple rules of thumb you can use here:
  - Start at what you think may be a lean value, then work your way up in small increments.
  - Pull a spark plug after a failed start and check the electrode. If it’s wet with fuel, decrease the amount of fueling. If it’s dry, increase it. And if the electrode is covered with crud, your problem may just be that you need new spark plugs.
- If the engine responds better if you give it a bit of throttle, you probably have too much fuel (because the extra air helps get your air/fuel ratio closer to what it should be). So you need less fueling. Conversely, if giving it the throttle makes things worse, you probably need more fuel.
- Check for various physical problems. Some examples I’ve seen include coil packs or distributors with one or two spark plug wires run to the wrong cylinders, a broken keyway on the crankshaft pulley causing the timing marks to be 60 degrees off, and even a zip tie that got pinched in the intake manifold gasket next to a coolant passage, which nearly hydrolocked the motor.

2.21 Run engine up to temperature

Once you have got the engine to start, keep it running and bring it up to temperature. It is normal to need to use some throttle to keep it alive. Step 2.18 can be performed during warmup.

While the engine is warming up, check for oil and coolant leaks.

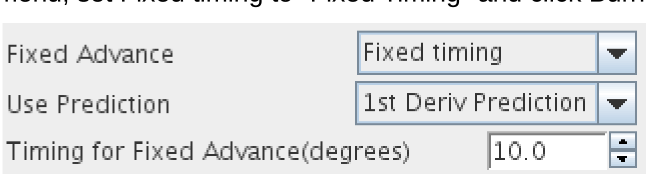
If possible, check the exhaust temperatures on each cylinder. Expect to see a variation between cylinders. Differences of up to 180F (100C) are fairly normal. Beyond that you need to investigate why. There could be an air leak, a faulty injector, a damaged valve seat or other mechanical issues with the engine. If cylinders are very cold, stop the engine and check for a fouled plug.

Pay attention to the engine; if something looks, sounds or smells wrong - STOP!

2.22 Setting running timing

Now that the engine is running, we can come back and make the final timing adjustments. This step can be performed while the engine is warming up.

Hook up your timing light and start your engine. In TunerStudio, on the Ignition Options / Wheel Decoder menu, set Fixed timing to "Fixed Timing" and click Burn.



This locks the timing to the advance you specify in the "Timing for Fixed Advance" box. You do not need to alter your spark table.

- Enter say 15 degrees (or another angle that your engine will idle easily at.)
- Using your timing light, confirm that the timing on the crank matches the fixed advance at low RPMs
- If it does not, then adjust your tooth#1 angle (trigger wheel) or trigger angle/offset (other spark modes.)
- If the actual timing is more advanced than Megasquirt thinks, then increase the trigger angle/offset. If adjustments of more than 10 degrees are required, make the change, click Burn, turn the key off, and restart the engine.
- Once your idle timing is matching up with your fixed angle, increase the engine revs and confirm that the timing does not drift.

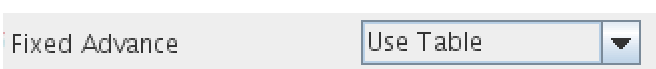
If you get a large timing advance with increased rpms on a VR sensor distributor then it is likely that your ignition input capture is backwards.

For a few ignition configurations, it is also possible that your spark output setting is wrong - however, exercise caution in changing this, as the wrong setting can damage your coils or ignition module. Most configurations should use "Going High" but this does depend on the product and coils.

If the timing retards a little as rpms increase you can make a correction by adjusting the Spark Hardware Latency setting. The larger the number, the more compensation it will apply.

Do not proceed any further with your install if you cannot get stable timing. Note that “stable timing” is a matter of degree - a small block Chevy with an HEI distributor and a worn timing chain, for example, may have 2 or more degrees of timing jitter in normal operation. And many dial-back timing lights will have problems keeping up with rapid changes in RPM. But if you are seeing the timing bouncing around by 10 degrees or more, stop and investigate what’s wrong.

When finished, be sure to turn off the Fixed Timing and return to Use Table and click Burn.



3: Tuning the engine

3.1 How it works - fuel

The amount of fuel injected into the engine is controlled by the fuel pulsewidth "PW". This pulsewidth is calculated by the Megasquirt from a number of factors. It is important to understand what these are to know what settings, curves and tables need to be tuned.

3.1.1 Cranking mode

When the engine is in cranking mode (i.e. before it is has started and is below the "Cranking RPM" setting) there are a limited number of settings that influence the fuelling.

The primary setting is:

- Cranking Pulse curve

Sensor inputs:

- RPM input
- Throttle Position
- Coolant Temperature

The RPM input is used to determine whether the engine is trying to start and should be in cranking mode.

The Throttle Position only takes effect at full throttle - this can be used to disable fuel to clear a flooded engine. If the TPS is broken or malfunctioning, it could enable flood-clear by accident and the engine will not start.

The Coolant Temperature is used as a lookup on the Cranking Pulse curve to determine how much fuel to inject.

(A future revision may also use the ReqFuel number, but that is not used at this time.)

Note that none of the other fuel tables and curves are used during cranking.

3.1.2 Run mode

Once the engine has fired and the RPMs exceed the "Cranking RPM" setting, the run mode calculations are used. Now all of the available settings are used.

The primary settings are:

- ReqFuel
- VE table

The secondary settings are:

- AFR table (only with Incorporate AFR)
- Accel enrichment
- VE Trim
- MAF flow curve
- WarmUp Enrichment curve
- AfterStart Enrichment curve

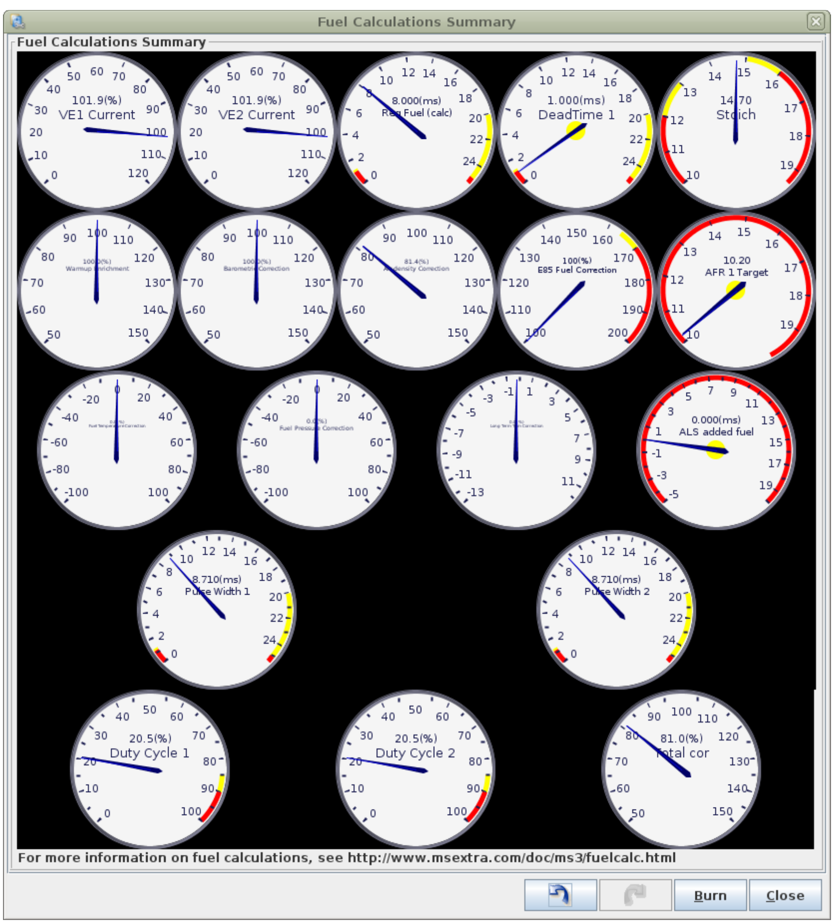
Sensor inputs:

- RPM input
- MAP sensor
- MAF sensor
- Throttle Position
- Air Temperature
- Coolant Temperature
- Flex fuel sensor

The exact tables and sensor inputs used varies depending on the selected fuel Algorithm and are covered in the appropriate tuning section.

Of particular note is the AFR table. When 'incorporate AFR' is off, the AFR table is a reference table only and is not used for fuel calculations. When 'incorporate AFR' is on, this table is included in the fuel calculation.

The Fuel Calculations Summary screen shows many of the factors that are used to arrive at the final pulsewidth.



More detail on the way the fuel calculations work internally may be found in Appendix B.

3.2 How it works - ignition

The final ignition timing (spark advance) is calculated by the Megasquirt from a number of factors. It is important to understand what these are to know what settings, curves and tables need to be tuned.

3.2.1 Cranking mode

When the engine is in cranking mode i.e. before it is has started and is below the "Cranking RPM" setting. There is just one setting that influences the advance.

The primary setting is:

- Cranking Advance setting

Sensor input:

- RPM input

The RPM input is used to determine whether the engine is trying to start and should be in cranking mode. When in cranking mode the "Cranking advance" setting only is used.

Note that none of the other ignition tables and curves are used during cranking.

3.2.2 Run mode

Once the engine has fired and the RPMs exceed the "Cranking RPM" setting, the run mode calculations are used. Now all of the available settings are used.

The primary settings are:

- Ignition table

The secondary settings are:

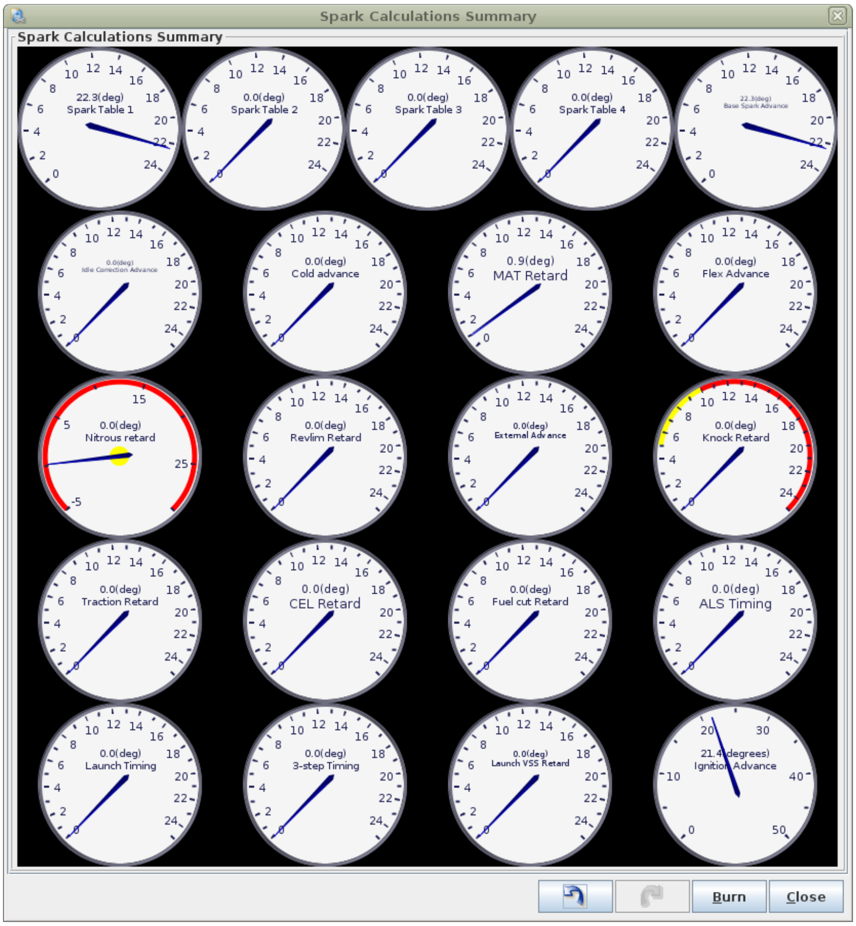
- Cold Advance
- MAT Retard
- Rev/launch limiter retards
- Nitrous retards

Sensor inputs:

- RPM input
- MAP sensor
- MAF sensor
- Throttle Position
- Air Temperature
- Coolant Temperature
- Flex fuel sensor
- Launch/nitrous switch inputs

The exact tables and sensor inputs used varies depending on the selected fuel Algorithm.

The Spark Calculations Summary screen shows many of the factors that are used to arrive at the ignition (spark) advance.



3.3 Tables and curves

The TunerStudio Reference Guide section 1.5 gives an overview of settings screens, curves and tables.

3.4 Tuning fuel

As outlined in section 3.1, there are a number of curves and tables that are used to tune fuel. Now we will cover them in more detail.

Remember that all initial tuning should be on a warmed up engine. (Tune Warmup and Afterstart only once the main tune is good.)

The Primary Fuel Load (Algorithm) determines how fuelling works, these algorithms will be handled in their own sections to keep the details clear. You should have already set this is section 2.12.1

The Engine Load gauge shows the value of the selected load input.

3.4.1 Tuning fuel - Speed Density

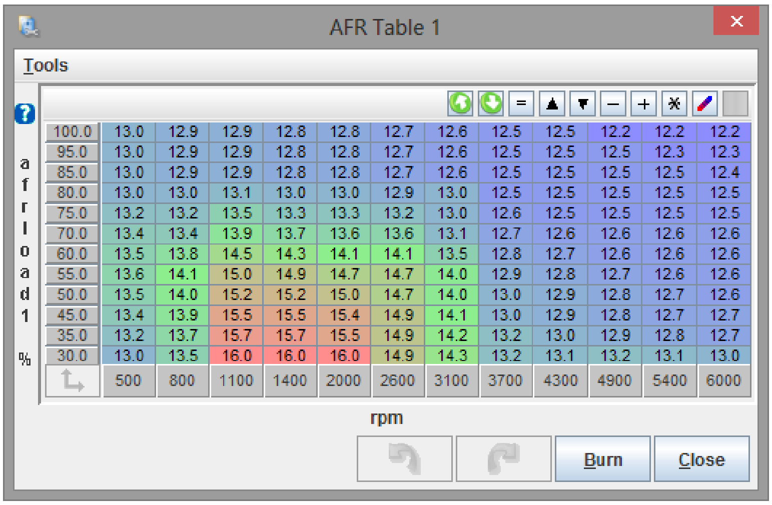
Speed Density uses a MAP (pressure) sensor as the Load input.

3.4.1.1 AFR table (SD)

The AFR table can be used in different ways.

- If you set “Incorporate AFR Target” under the General Settings menu to “include AFRtarget,” it will scale the pulse widths based on the numbers entered.
- It can be used for closed loop O2 correction with a wideband O2 sensor.
- It can be used purely as a reference table for external tuning. In this case it is not used internally at all.

Now, let’s take a look at how you use this particular table.



afrload1 %	500	800	1100	1400	2000	2600	3100	3700	4300	4900	5400	6000
100.0	13.0	12.9	12.9	12.8	12.8	12.7	12.6	12.5	12.5	12.2	12.2	12.2
95.0	13.0	12.9	12.9	12.8	12.8	12.7	12.6	12.5	12.5	12.5	12.3	12.3
85.0	13.0	12.9	12.9	12.8	12.8	12.7	12.6	12.5	12.5	12.5	12.5	12.4
80.0	13.0	13.0	13.1	13.0	13.0	12.9	13.0	12.5	12.5	12.5	12.5	12.5
75.0	13.2	13.2	13.5	13.3	13.3	13.2	13.0	12.6	12.5	12.5	12.5	12.5
70.0	13.4	13.4	13.9	13.7	13.6	13.6	13.1	12.7	12.6	12.6	12.6	12.6
60.0	13.5	13.8	14.5	14.3	14.1	14.1	13.5	12.8	12.7	12.6	12.6	12.6
55.0	13.6	14.1	15.0	14.9	14.7	14.7	14.0	12.9	12.8	12.7	12.6	12.6
50.0	13.5	14.0	15.2	15.2	15.0	14.7	14.0	13.0	12.9	12.8	12.7	12.6
45.0	13.4	13.9	15.5	15.5	15.4	14.9	14.1	13.0	12.9	12.8	12.7	12.7
35.0	13.2	13.7	15.7	15.7	15.5	14.9	14.2	13.2	13.0	12.9	12.8	12.7
30.0	13.0	13.5	16.0	16.0	16.0	14.9	14.3	13.2	13.1	13.2	13.1	13.0

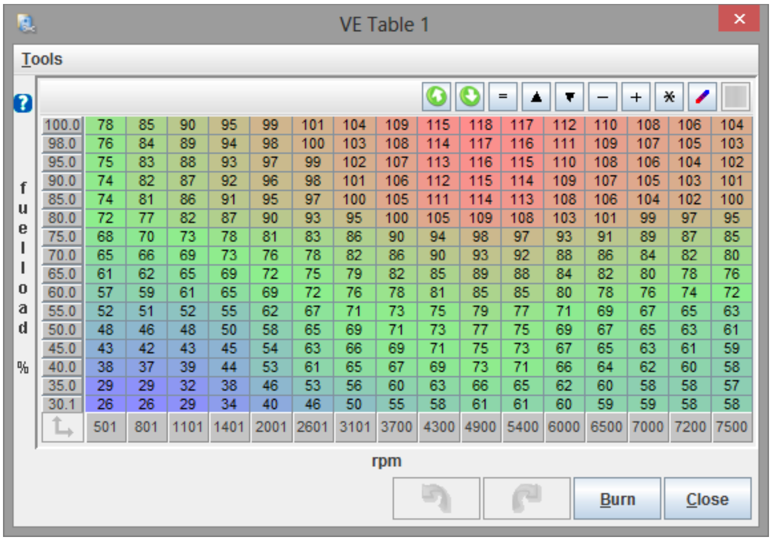
You’ll enter the AFR you want at each RPM and load point. Most sequential fire engines will run best at 14.7:1 at idle. Batch-fire engines will likely need to run richer. At cruising RPM and low load, you can run 14.7:1 or sometimes leaner depending on the engine. Most naturally aspirated engines tend to run best at 13.2 to 13.5 at full throttle, while engines running forced induction may need to run as rich as the mid 11’s under boost.

If you set “Incorporate AFR Target” under the General Settings menu to “include AFRtarget,” this table is used to scale the fueling. A table entry of 14.7:1 will mean no fuel scaling. Numbers less than that will increase the fuel, and numbers higher will decrease the fuel. When using this mode, you’ll tune your VE table to hit your AFR targets, then adjusting the AFR target table will directly adjust the fuel to the desired AFR number.

Before having your engine dyno tuned - make sure the dyno operator is clear about your selection for "Incorporate AFR Target" and whether the AFR table is being used. Otherwise the operator may aim for his own AFR target and ignore the table causing later confusion.

3.4.1.2 VE table (SD)

The VE table is your primary tuning table. At its simplest, bigger numbers equal more fuel.



VE Table 1

Tools

fuel load %	501	801	1101	1401	2001	2601	3101	3700	4300	4900	5400	6000	6500	7000	7200	7500
100.0	78	85	90	95	99	101	104	109	115	118	117	112	110	108	106	104
98.0	76	84	89	94	98	100	103	108	114	117	116	111	109	107	105	103
95.0	75	83	88	93	97	99	102	107	113	116	115	110	108	106	104	102
90.0	74	82	87	92	96	98	101	106	112	115	114	109	107	105	103	101
85.0	74	81	86	91	95	97	100	105	111	114	113	108	106	104	102	100
80.0	72	77	82	87	90	93	95	100	105	109	108	103	101	99	97	95
75.0	68	70	73	78	81	83	86	90	94	98	97	93	91	89	87	85
70.0	65	66	69	73	76	78	82	86	90	93	92	88	86	84	82	80
65.0	61	62	65	69	72	75	79	82	85	89	88	84	82	80	78	76
60.0	57	59	61	65	69	72	76	78	81	85	85	80	78	76	74	72
55.0	52	51	52	55	62	67	71	73	75	79	77	71	69	67	65	63
50.0	48	46	48	50	58	65	69	71	73	77	75	69	67	65	63	61
45.0	43	42	43	45	54	63	66	69	71	75	73	67	65	63	61	59
40.0	38	37	39	44	53	61	65	67	69	73	71	66	64	62	60	58
35.0	29	29	32	38	46	53	56	60	63	66	65	62	60	58	58	57
30.1	26	26	29	34	40	46	50	55	58	61	61	60	59	59	58	58

rpm

Burn Close

In any mode that uses this table, the numbers in the VE table are a percentage. The fueling equation takes the base pulse width from Required Fuel, scales it by the percentage in the VE table, and then applies any other corrections, enrichments, and the like, such as air density correction and warmup enrichment. If you have the tuning set to incorporate the AFR target, theoretically, the VE table will match the engine's actual volumetric efficiency. Don't be alarmed if you need to enter numbers above 100, particularly in boost. The maximum number is 255.

We recommend first setting up an appropriate AFR table, then adjust the numbers in the VE table (upwards to add fuel, downwards to take it away) until your actual air/fuel ratio hits the target table.

Before starting to tune the VE Table, it is recommended that you turn the Acceleration Enrichment (AE) off so you can see what's happening to the mixture as a direct result of the table rather than having fuel added from Accel Enrichment. See the next section on how to switch this off. You may find that you'll need to be light on the throttle, as the AE is needed to fill 'holes' that occur when the throttle opens quickly. Remember to tune the AE when you have a tuned VE Table.

Another item to turn off during tuning is the Overrun settings as the PW may drop to 0.0mS causing lean spots. You'll find this by going to the Fuel Settings menu -> Overrun fuel cut. The default tune has this turned off.

Tuning the VE table involves richening (by increasing the VE) or leaning (be decreasing VE) at each point in the VE table. Most of your driving will occur in a diagonal strip of the VE table, from low rpm, low kPa (i.e. idle) to high rpm, high kPa (i.e. WOT). You can adjust these values using the O2 sensor, data logs, VE Analyze Live in TunerStudio, and/or the seat of your pants. Low rpm and low kPa (say less than half of the max rpm and max kPa) might be able to use stoichiometric or leaner.

Richer mixtures would be used at high rpm and high kPa.

However, the low rpm/high kPa and high rpm/low kPa are not seen as often driving your vehicle.

Basically, if the engine never runs in certain parts of the MAP, then the numbers there should not matter. However, since you may not be able to guess where you will run under every possible set of conditions, you put estimated VE numbers that make sense into the little used areas.

From this frequently used diagonal strip of the VE table, you will be able to see how much the VE rises from one rpm bin to the next, and use these differences to estimate the low rpm, very high kPa numbers and the high rpm, very low kPa numbers. Since you rarely (if ever) run in these parts of the table, the actual numbers will not make much difference, but they will be there "just in case". You are looking to create a smooth VE map wherever possible.

It is a good idea to set either a row of kPa bins in the VE Table to the idle MAP value. The engine may not always idle at the same MAP value when under load, but it's worth trying to set this up for unloaded idle. Most of the time, an engine will idle best if you flatten out the cells above and below where it idles; it's rare for an engine to need large changes in VE at idle.

3.4.2 Tuning fuel - Percent Baro

Percent Baro works very similarly to Speed-Density. The only difference is that barometric pressure is taken into account for the VE table lookup. Instead of MAP being used on the Y axis, MAP/Baro is used instead.

Tuning works similarly to Speed-Density - see section 3.4.1

3.4.3 Tuning fuel - Alpha-N

Alpha-N uses the TPS (throttle) as the Load input. It can be a poor estimator of airflow, but may be useful if a MAP signal is unavailable or unreliable.

Tuning works similarly to Speed-Density - see section 3.4.1

3.4.4 Tuning fuel - MAF

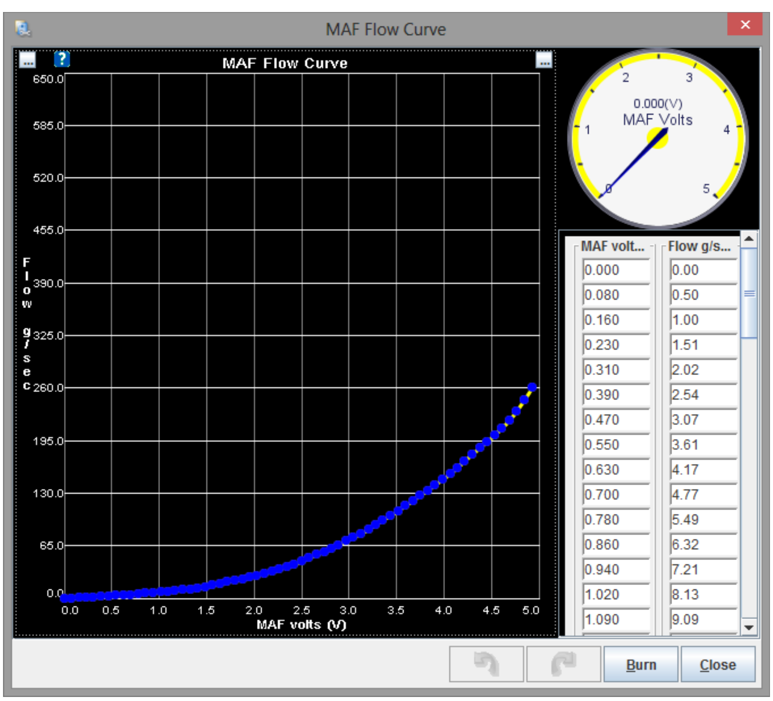
MAF uses an actual mass air-flow sensor as the Load input. It can be a more accurate estimate of airflow, but is sometimes more complex to configure.

Tuning works quite differently from Speed-Density, take note of the following explanations.

Before having your engine dyno tuned - make sure the dyno operator is clear about how MAF mode is tuned.

3.4.4.1 MAF transfer curve

The MAF transfer curve is used to convert the voltage input from the MAF sensor into an actual air-flow number in grammes per second. You need to obtain the actual curve for your sensor or the best estimate.

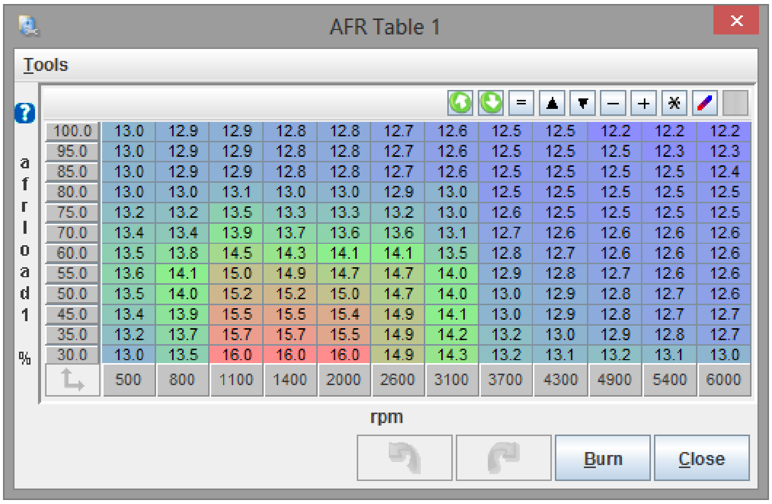


This curve is found on the Basic/Load Settings menu, once you have wired up and configured your MAF sensor. See the TunerStudio Reference Guide.

3.4.4.2 AFR table (MAF)

In MAF mode, the AFR table is your main fuel tuning table. If you do not hit the actual AFR targets, you will need to adjust the MAF flow curve under the Basic / Load Settings menu.

Now, let’s take a look at how you use this particular table.

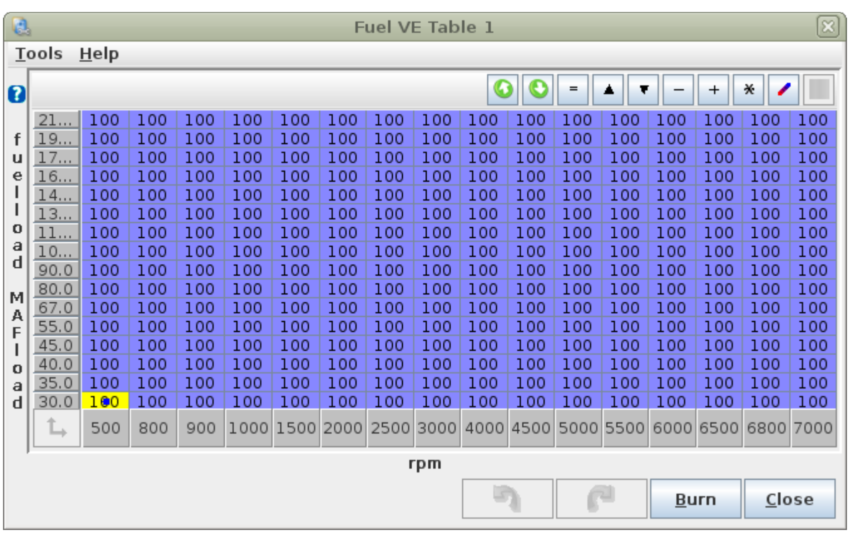


afrload1 % \ rpm	500	800	1100	1400	2000	2600	3100	3700	4300	4900	5400	6000
100.0	13.0	12.9	12.9	12.8	12.8	12.7	12.6	12.5	12.5	12.2	12.2	12.2
95.0	13.0	12.9	12.9	12.8	12.8	12.7	12.6	12.5	12.5	12.5	12.3	12.3
85.0	13.0	12.9	12.9	12.8	12.8	12.7	12.6	12.5	12.5	12.5	12.5	12.4
80.0	13.0	13.0	13.1	13.0	13.0	12.9	13.0	12.5	12.5	12.5	12.5	12.5
75.0	13.2	13.2	13.5	13.3	13.3	13.2	13.0	12.6	12.5	12.5	12.5	12.5
70.0	13.4	13.4	13.9	13.7	13.6	13.6	13.1	12.7	12.6	12.6	12.6	12.6
60.0	13.5	13.8	14.5	14.3	14.1	14.1	13.5	12.8	12.7	12.6	12.6	12.6
55.0	13.6	14.1	15.0	14.9	14.7	14.7	14.0	12.9	12.8	12.7	12.6	12.6
50.0	13.5	14.0	15.2	15.2	15.0	14.7	14.0	13.0	12.9	12.8	12.7	12.6
45.0	13.4	13.9	15.5	15.5	15.4	14.9	14.1	13.0	12.9	12.8	12.7	12.7
35.0	13.2	13.7	15.7	15.7	15.5	14.9	14.2	13.2	13.0	12.9	12.8	12.7
30.0	13.0	13.5	16.0	16.0	16.0	14.9	14.3	13.2	13.1	13.2	13.1	13.0

You’ll enter the AFR you want at each RPM and load point. Most sequential fire engines will run best at 14.7:1 at idle. Batch-fire engines will likely need to run richer. At cruising RPM and low load, you can run 14.7:1 or sometimes leaner depending on the engine. Most naturally aspirated engines tend to run best at 13.2 to 13.5 at full throttle, while engines running forced induction may need to run as rich as the mid 11’s under boost.

3.4.4.3 VE trim table (MAF)

In MAF mode, the VE table can be used to 'trim' the fuelling. Before starting, you MUST set the table to 100% in all cells. (Highlight all cells, press =, type in 100, then OK.)



fuelload MAFload \ rpm	500	800	900	1000	1500	2000	2500	3000	4000	4500	5000	5500	6000	6500	6800	7000
21...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
19...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
17...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
16...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
14...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
13...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
11...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
10...	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
90.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
80.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
67.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
55.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
45.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
40.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
35.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
30.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100

The table will apply a percent trim to the fueling calculated from the MAF sensor curve and AFR table.

Before enabling this table, make your best effort to dial-in the fuel with the MAF transfer curve.

3.4.4.4 MAFload

In MAF mode, the primary fuel load is set to "MAF" because fuel required is directly related to mass air flow. However, ignition and other load based tables are not directly related in the same way. While it would be possible to use MAF volts as the Y-axis on other tables, it would mean that only a very narrow band of the table was used and tuning would suffer. So instead of that, the code calculates a synthesized load value called MAFload. This behaves similarly to MAP in a speed density system and should give a repeatable load value and allow tuning of those tables in the conventional manner. Do not be too concerned about the exact values of MAFload, but do make sure the axes on the table are set to cover all MAFload values and RPMs. When live tuning, the active cell will be highlighted.

A MAF only install will need to use "MAFload" for spark and AFR table lookups. But, if you have a MAP sensor installed you could optionally choose to set "Speed Density" for spark and/or AFR which will then use MAP as the Y-axis in the table.

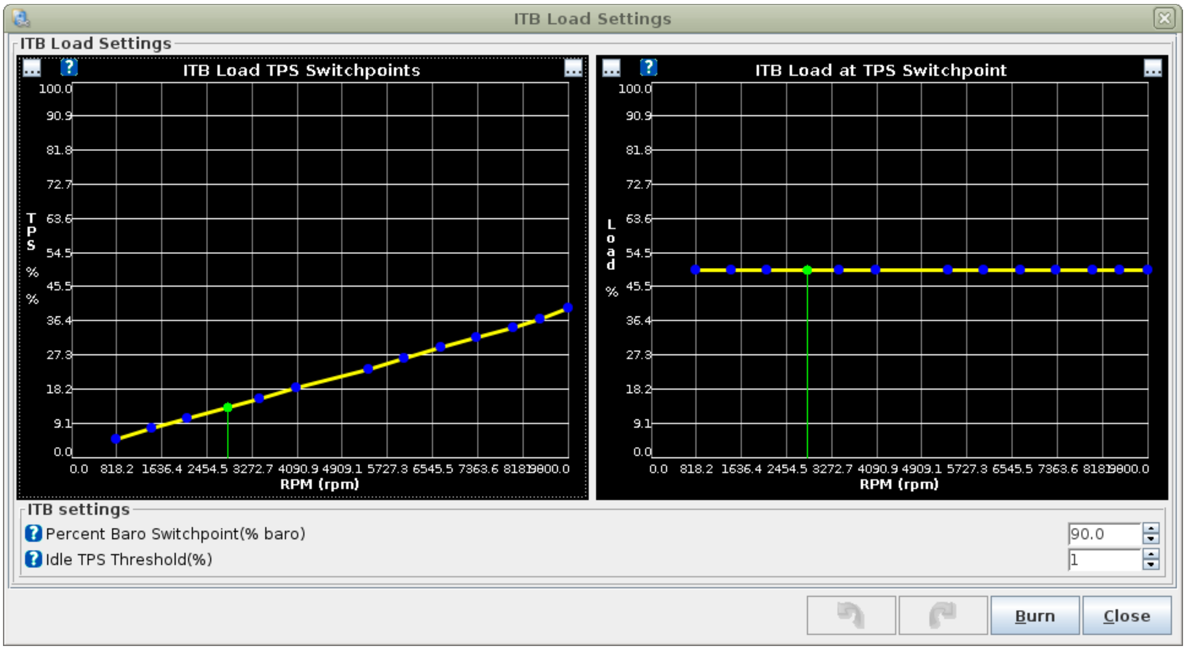
3.4.5 Tuning fuel - ITB

ITB tuning mode builds on the capabilities of the dual table blended tuning approach but solves one of the more significant drawbacks to that tuning mode: it provides the blended Speed-Density/Alpha-N behavior of the blended dual tables in just a single table. This single table approach is a significant improvement as all the automatic tuning tools available through TunerStudio now work correctly with the single table. These tools are not easily used on the blended tuning approach as TunerStudio does not understand the multiplicative coupling between the two tables.

This tuning mode creates a load type called "ITB". This "Primary Fuel Load" (algorithm) is selected in Basic/Load Settings -> General Settings. ITB load is also available for the other tuning settings including AFR, Ignition, and Enhanced Accel Enrichment.

The ITB Load is derived from a combination of MAP and TPS values as well as other ITB-related tuning curves that all work together to create a calculated value that is used as the "ITB Load" and applied to the Y axis of the tuning tables.

ITB mode uses two additional curves as part of the initial setup before the VE table is used for main tuning. **Once these curves have been set, you can leave them alone and focus on tuning the VE table in the conventional manner.**



On the left is the ITB Load TPS Switchpoint curve, on the right is the ITB load at TPS Switchpoint

• % Baro Switchpoint - The MAP value where tuning switches from TPS based to MAP based.

Default is 90% and works for most engines.

• Idle TPS Threshold - A minimum TPS value that must be met to allow the switch to TPS based

tuning to occur. This is sometimes useful in idle tuning.

3.4.5.1 ITB Load TPS Switchpoint Curve

This curve defines the TPS value where the MAP load reaches the %baro switchpoint. This curve will be different for each engine and should be set up using values obtained from log files from your engine.

The curve tends to be fairly linear so you only need a few data points to plot the curve. A data point at low, medium, and high RPM from a log file is usually enough. A spreadsheet or just graph paper can then be used to establish enough data points to fill in the table for this curve.

On a new install, start with the default curve, then carefully run the engine under load to find out what %TPS is required to reach 90kPa at different RPMs. You will likely need to rough in the tune for this step. Then update the TPS switchpoint curve and recommence tuning.

ITB Load tuning requires that the MAP signal be above the %Baro switchpoint and that the TPS value be above the value defined on this curve to switch from Speed-Density tuning to Alpha-N tuning. Therefore, you want this curve to be relatively accurate and you may even want to set the values on the curve a few percent low to ensure that the TPS value has been met when the MAP reaches the %Baro switchpoint.

Besides defining the switch point to Alpha-N tuning, this curve also establishes the lower TPS value that will be used to interpret the range of VE bins allocated to Alpha-N tuning in the VE table.

3.4.5.2 ITB Load at TPS Switchpoint Curve

This curve is used to allocate the bins on the VE table to either Speed-Density or Alpha-N tuning. The area of the VE table below the curve will be used for Speed-Density tuning and the area above the curve will be used for Alpha-N tuning. The shape of this curve defines how much of the VE table will be allocated for use between Speed-Density and Alpha-N tuning for each RPM column. You want to allocate the largest portion of the VE table at each RPM to the tuning mode that has the most non-linear response. The lower RPM region typically requires a little more Speed-Density definition range than the upper RPMs.

The current defaults are 50% at all points, so the top of the VE table is Alpha-N and the bottom is the Speed-Density region.

Unless you have good reason to change it, leave the curve at 50% load at all RPM points.

3.4.5.3 AFR Table (ITB)

In ITB mode, the AFR table works similarly to Speed-Density mode, but using ITB load as the Y axis. See section 3.4.1.1

3.4.5.4 VE Table (ITB)

In ITB mode, the VE table works the similarly to Speed-Density mode, but using ITB load as the Y axis. See section 3.4.1.2

The aim of the ITB mode is to seamlessly switch between Speed-density and Alpha-N with the single tuning table.

3.4.5.5 ITB technical

Here are some of the technical details behind the ITB mode, you may want to skip these.

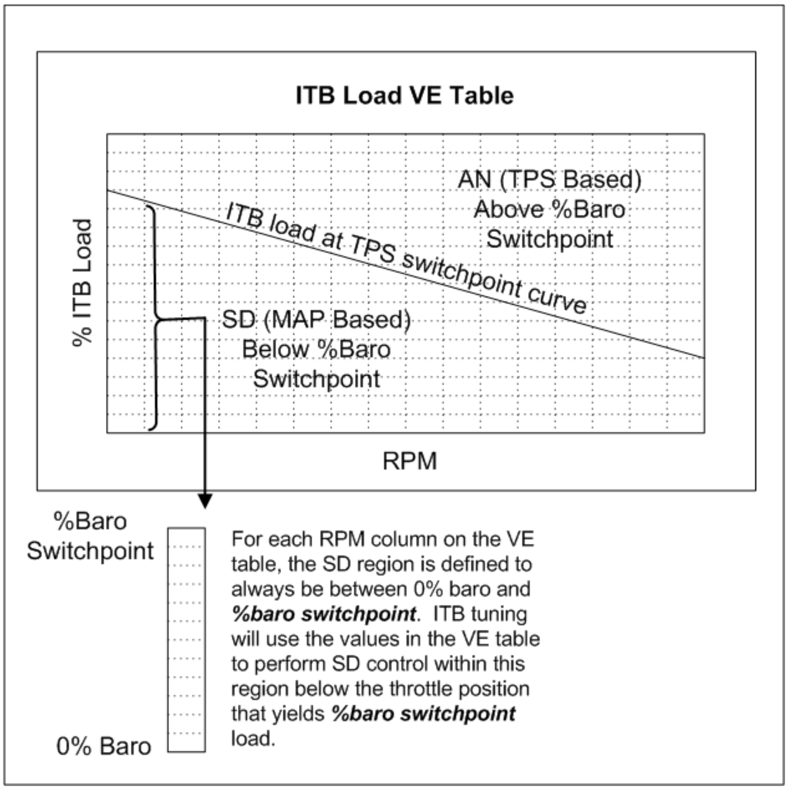
The ITB Load VE table is defined as ITB Load vs. RPM. The ITB Load should not be confused with MAP or TPS, it is neither. What the ITB Load tuning algorithm does is calculate a load value based on MAP, TPS, and the two ITB Load curves. This calculated load value is the Y axis of the ITB Load VE table; it can also be applied to the ignition advance and AFR tables as well.

ITB Load Calculation in Speed-Density Mode

When the throttle position is less than the value defined in the ITB load TPS switchpoint curve or the MAP value is less than the %Baro switchpoint, the tuning algorithm will take the array of cells from the VE table below the ITB load at TPS switchpoint curve and interpret this array within the context of 0% to %Baro switchpoint load.

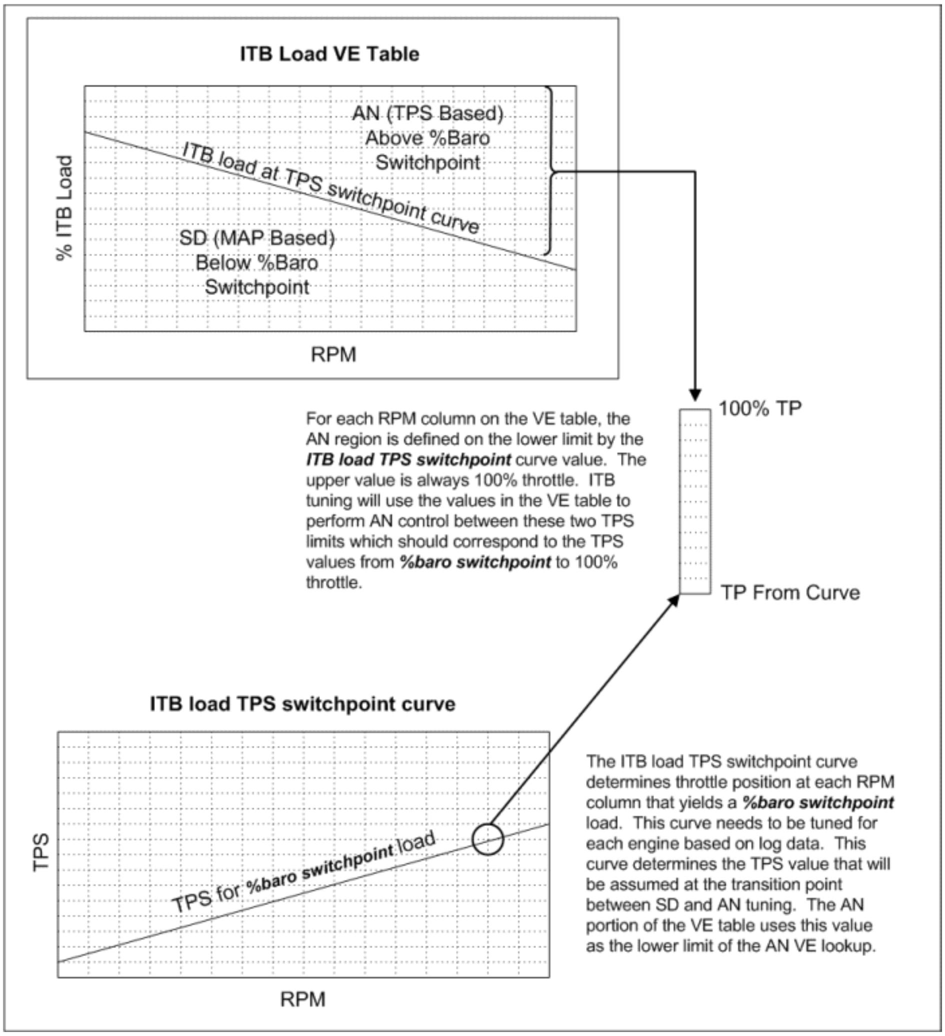
In all examples we take %Baro switchpoint = 90

If you have allocated the region between 0% ITB Load and 60% ITB Load on your VE table for use in Speed-Density tuning and your %baro is 50% then the VE value for 30% ITB load will be used. In this same example, a MAP value of 0kpa would use the 0% ITB load bin and a MAP value of just less than %Baro switchpoint would use the VE value just below the 60% ITB Load value on the VE table.



ITB Load Calculation in Alpha-N Mode

When the throttle position is greater than or equal to the ITB load TPS switchpoint curve and the MAP value is greater than or equal to the %Baro switchpoint, the tuning algorithm will take the array of cells from the VE table above the ITB load at TPS switchpoint curve and interpret this array within the context of TPS position. The lower TPS value used for this interpretation is taken from the ITB load TPS switchpoint curve and the upper TPS value is always 100%.



For example, if you have allocated the region between 60% ITB Load and 100% ITB Load on your VE table for use in Alpha-N tuning and also assigned a value of 10% TPS on your ITB load TPS switchpoint curve then a TPS value of 55% would yield an ITB Load value of 80% and the VE bin for 80% ITB Load would be used. In this same example, a TPS value of 10% would use the 60% ITB Load bin on the VE table and 100% TPS would use the 100% ITB Load bin.

3.4.5.6 Tuning For Idle Air Control

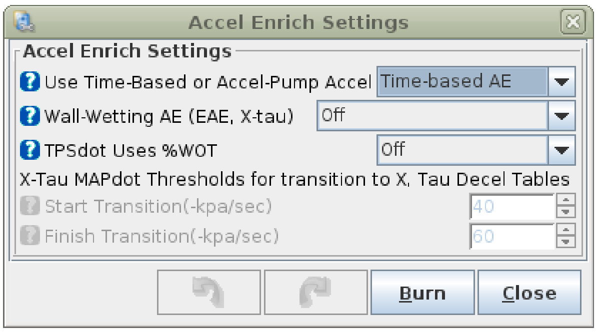
The Idle TPS Threshold % setting defines the minimum TPS value required to allow the transition to Alpha-N tuning. This feature allows the fast idle MAP to exceed the %Baro switchpoint without entering Alpha-N mode. On engines with aggressive cams with higher overlap, it is possible for the fast idle MAP to exceed the %Baro switchpoint. When this happens, you do not want to enter Alpha-N mode tuning since the throttle is still reading fully closed. The Idle TPS Threshold % prevents you entering Alpha-N mode and you can use the warm up enrichment curve to compensate for any tuning errors caused by being stuck at the maximum Speed-Density bin while the MAP is greater than the %Baro switchpoint.

3.4.6 Basic acceleration enrichment (AE) tuning

Megasquirt-3 has basic acceleration enrichment, plus model based acceleration enrichment that we'll discuss later.

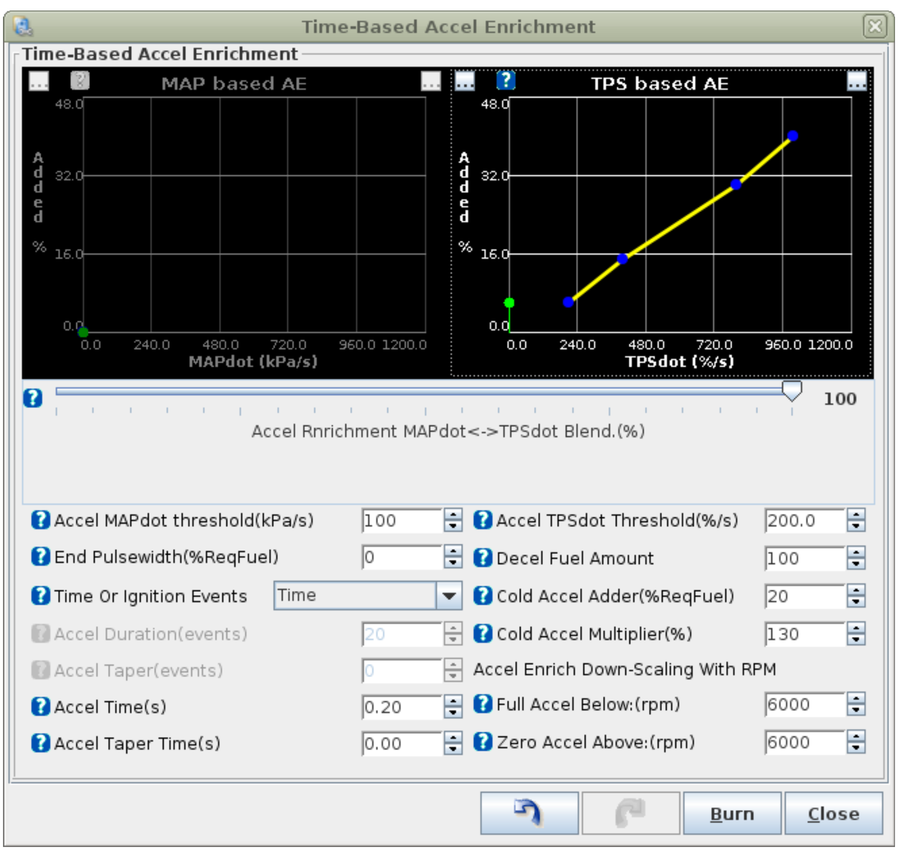
The basic time based accel enrichment uses variables called TPSdot and MAPdot. The “dot” is a calculus notation that indicates rate of change with respect to time. Don’t worry, you don’t need to know calculus to tune this. The amount is expressed as a percent of change over time. With TPSdot, it is percent opening versus time; a TPSdot rate of 100 %/sec would be opening from closed to wide open throttle in one second, or the equivalent speed, such as going from 25% to 75% throttle in 1/2 second. The MAPdot is measured in kPa per second. Megasquirt-3 can use either one, or a combination of the two, to trigger acceleration enrichment.

3.4.6.1 Main accel enrich settings menu



This menu allows wall-wetting accel enrichment to be optionally enabled and determines whether the %WOT table is used with TPSdot.

3.4.6.2 Time-based Accel



The two curves are the primary tuning tools. They set the percentage of pulse width to add as a function of the rate of change of MAP and TPS readings. The adder % is a percentage of the main Required Fuel number. Under these charts, you can set a percentage to blend MAP and TPS driven acceleration enrichment. 100% is entirely TPS driven, and 0% is entirely MAP driven. In between, the percentage will scale the MAP driven pulse width adder down and the TPS driven pulse width adder up.

Once triggered, the acceleration enrichment will be active for a defined time period.

3.4.6.3 Getting it dialed in

Regardless of which of the two settings you use, many of the aspects of acceleration enrichment tuning will be the same. Before you start tuning the acceleration enrichment, make sure your VE tables are dialed in. Adding a correctly tuned amount of acceleration enrichment to an incorrectly tuned fuel table is still going to result in an incorrect amount of fuel. Since the acceleration enrichment is stacked on top of the fuel the ECU is commanding based on the VE table, you'll want to tune AE last, after you've got the solid foundation of a well tuned VE table under you.

The standard AE can be thought of like the accelerator pump on a carb, which shoots in some extra fuel when you suddenly hit the throttle to compensate for the massive influx of air when you open the throttle blades. The goal is to compensate for this influx of air for an instant during that transient until the transient is over and the VE tables are again providing the proper amount of fueling. If your background is tuning carburetors, you can think of this as a very powerful fully adjustable/tunable accelerator pump that lets you compensate for more than one rate of throttle position change. Slow 'tip-in' can be adjusted with a certain PW adder, medium with another larger PW shot, mid-fast with another size PW squirt, and very fast throttle movements with another.

MAP based or TPS based AE? If you have a variable Throttle Position Sensor, we'd recommend 100% TPS based AE as a starting point. MAP based AE does the trick if you're running an engine without a usable TPS, but TPS based AE is generally easy to tune to the proper sensitivity level such that it activates when you want it to, and not when you don't.

Setting the TPSdot/MAPdot Threshold Levels First up, you need to determine what your TPSdot or MAPdot threshold will need to be. Below the line graph charts, you'll see a TPSdot threshold and a MAPdot threshold. These are the minimum amount of change of TPS position or MAP pressure reading, respectively, that are needed to trigger acceleration enrichment. Your goal is to set these high enough that the acceleration enrichment won't accidentally trigger at steady throttle, but low enough that it's sensitive enough to catch small or slow throttle movements as well. Setting it too high can mean the AE won't kick in when you need it. You want it as low as you can get it without it causing AE to trigger when it's not supposed to, such as during steady throttle.

Getting this dialed in can be done methodically, or by trial and error. Here's the methodical method.

You'll need to have a safe way to accelerate from the bottom to the top of a gear, preferably your 1:1 ratio gear, and you'll want to datalog your engine idling, as well as these WOT pulls with your Megasquirt-3 and TunerStudio. Ideally this is done on a dyno. Two to three pulls is better than one, but there's no need to get carried away. Then you can stop the datalogging and load this log up into MegaLogViewer.

In MegaLogViewer, on the left using the drop-down boxes that let you select what data you are viewing in your logs, set one of these to TPSdot, and/or MAPdot. You can set another to TPS to show you what throttle position you were at as well. Additionally, monitoring injector pulsewidth (select PW in the log to view that data) is helpful too, as you'll be able to see the commanded injector pulsewidth including the extra fuel added during an AE event. Lastly, near the bottom right corner of MegaLogViewer you'll see four indicators. TP AE, TP DE, MAP AE and MAP DE. If either AE is active, that's a Accel event and AE is adding fuel. If either DE is active, that's a Decel event and AE is pulling fuel if configured to do so.

Next, look at the datalog section of the car idling. The TPS signal should be rock solid, no movement. If it's jumping around, you have a noise issues on your TPS signal. This can be caused by ECU sensor signal wiring that's been routed too close to high voltage sources such as ignition coils, plug wires, the distributor, etc. Re-routing those wires away from those high-voltage sources is always a good idea. Additional possibilities for noise to get in the TPS signal could be a bad sensor ground (be sure you've grounded it the Megasquirt-3 sensor return wire ONLY), or a faulty throttle position sensor. Similarly noise on the MAP signal could be electrical, but is more likely to actually be real movement in the manifold pressure, at least at the source you're grabbing the vacuum signal from, meaning the port on your intake manifold that's connected to the MAP sensor.

If you're seeing AE events at idle, your TPSdot or MAPdot threshold is set too sensitive (too low of a number). You'll see the pulse width bounce around when this happens as well; AE is just doing what you told it to do.

Now look at the datalog sections for the WOT pulls, step through these in the logs and look for any time AE activated (Accel or Decel). It shouldn't be active at all on WOT pulls, because you're at Wide Open Throttle, you're not moving the throttle around, so AE is not needed. Again, if you are seeing AE activate during WOT pulls, your threshold is too sensitive.

In either case if you're triggering AE when you shouldn't be, you'll want to increase the threshold value a bit above the highest TPSdot (or MAPdot) value you see it reach in the logs. This is the lowest you can set your threshold in order to be a sensitive as possible while not triggering at steady state throttle (or idle, which is also steady state throttle, it's the steady state of NO throttle).

Note this setting is also the same setting you should use for your lowest (slowest rate of change) TPSdot or MAPdot field in the next section.

Tuning the tables Now that you have these, it's time to tune the PW adder line graphs. To tune these, start with the lowest speed throttle change position, and practice 'tipping in' at that rate of speed. Use the real-time line graph at the bottom of the page to see how fast you're tipping in.

When you've become used to applying the throttle at the right rate of speed to be close to the TPSdot number you're using in the 'slow' setting, watch your wideband O2 sensor readout. You want to be able to determine what's happening when you stab the throttle at that rate of speed. Is it going lean or rich? Note that it's hard to see this on a gauge, as often it will go one then the other really fast. You want the first response; if it goes lean then rich you'll usually see it hit 20:1 or maybe 22:1 for just an instant, and then as it goes back to running purely the table it may go to 13:1 or whatever... you want to watch that initial response, and if the first response is to go lean, then you want to add fuel; if the first response is to go rich, you want to reduce fuel (PW).

A wideband O2 sensor is immensely helpful here, but don't get too caught up in seeing perfect AFR numbers through the whole transient event. Adjust it until it feels right, snappy and responsive with no hesitation. The feel is more important than looking for perfect numbers. If you want perfect numbers, you may want to try the model based acceleration enrichment. Setup basic AE first as described here. Next you'll adjust the other set points you've setup for faster throttle rate changes.

Repeat the process for the medium speed tip in, then the mid-fast, then the fast.

3.4.7 Startup / warmup fueling

On the first couple starts, you'll want to use whatever combination of fuel and throttle can nurse the engine along until it is warmed up, and dial in the fuel at idle while hot. The reason is that the main fuel tune will affect startup behavior, and you will want to have good, solid values in the VE table (at least at idle) before you can get final values for the startup sequence. Here is the sequence that Megasquirt-3 fueling goes through while starting.

1. When the key is on, the Megasquirt-3 will turn the fuel pump on for 2 seconds and fire one priming pulse to clear air from the lines. (You can disable this feature by setting the priming pulse to zero, this will also stop the fuel pump priming.)

2. When the engine starts turning, the Megasquirt-3 will run off the cranking pulse width table until the RPM rises above the cranking RPM threshold.

3. Once the RPM rises above the cranking threshold, the Megasquirt-3 transitions to the main fuel table. It will initially multiply the pulse width from this table by the warmup enrichment (WUE) and afterstart enrichment (ASE) combined.

4. The ASE tapers off based on a user defined curve. After the ASE tapers away, the MS3 will run on the main fuel table multiplied by WUE.

5. When the engine reaches normal operating temperature, as defined by the WUE curve, the engine will run on the main fuel table, and the warmup cycle is over.

3.4.7.1 Priming pulse

This curve allows you to specify an initial pulse, in milliseconds, that is fired on the first turn of the key. You can set the pulse width as a function of coolant temperature.

3.4.7.2 Cranking pulse

This curve allows you to specify an initial pulse, in milliseconds, that is fired during cranking. There is one pulse per cylinder "event." You can set the pulse width as a function of coolant temperature.

As this curve is in raw pulsewidth, if you swap injectors you will need to retune it.

3.4.7.3 Afterstart (ASE) percentage adder

This is a curve of percentage adder versus time. The Megasquirt-3 will apply the full amount (looked up against temperature) immediately after RPM climbs past cranking RPM and then taper it off linearly through the ASE taper time. This number is added to the WUE number and then the combined number is multiplied by the number the main fuel loop calculates. 0% is no afterstart enrichment.

3.4.7.4 Afterstart (ASE) taper

Specifies how long in engine events (cycles) to apply afterstart enrichment.

3.4.7.5 Warmup enrichment percent multiplier

Specifies the percentage multiplier for fuel during warmup. At cold temperatures 200-300% may be required. At fully warm (the final row in the table or point on the curve) 100% MUST be specified.

3.4.7.6 Putting it all together

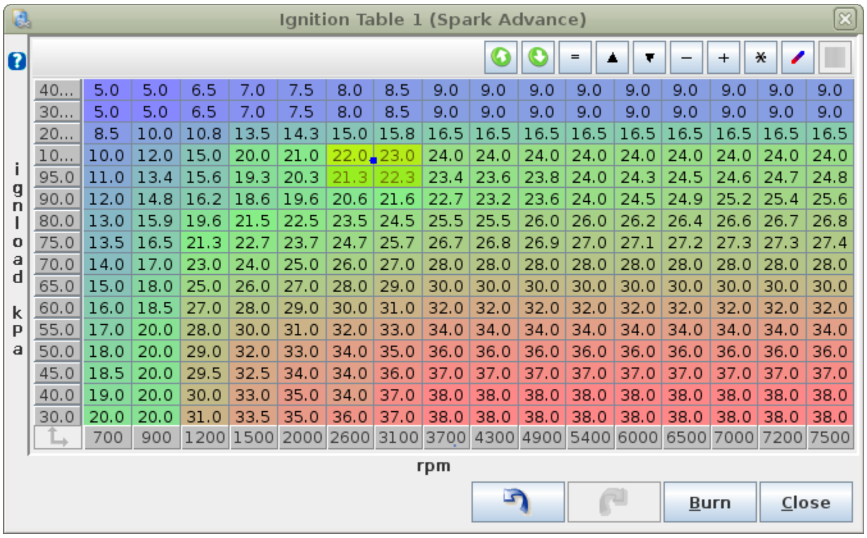
If the engine isn’t starting, pay attention to the RPM it runs through.

If the engine does not catch, but cranks at a steady RPM, you will need to adjust the cranking pulse width. If the engine fires up, but then stalls in a few seconds, you will need to adjust the afterstart enrichment. You cannot rely on an O2 sensor feedback for cranking, but here are some rules of thumb you can use.

- It’s better to start off at a point you expect to be lean and add fuel in small increments, to avoid flooding the engine.
- If giving the engine a small bit of throttle helps, you probably have too much fuel. Conversely, if this makes it worse, you probably have too little.
- You can pull a spark plug and check if it’s wet with fuel or dry
    - Wet - you need to reduce the fueling at the point where it stalled
    - Dry - you need to add more fuel.

3.5 Tuning spark

Ignition is tuned on a 3D table.



ignload kpa \ rpm	700	900	1200	1500	2000	2600	3100	3700	4300	4900	5400	6000	6500	7000	7200	7500
40...	5.0	5.0	6.5	7.0	7.5	8.0	8.5	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0
30...	5.0	5.0	6.5	7.0	7.5	8.0	8.5	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0	9.0
20...	8.5	10.0	10.8	13.5	14.3	15.0	15.8	16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5	16.5
10...	10.0	12.0	15.0	20.0	21.0	22.0	23.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0	24.0
95.0	11.0	13.4	15.6	19.3	20.3	21.3	22.3	23.4	23.6	23.8	24.0	24.3	24.5	24.6	24.7	24.8
90.0	12.0	14.8	16.2	18.6	19.6	20.6	21.6	22.7	23.2	23.6	24.0	24.5	24.9	25.2	25.4	25.6
80.0	13.0	15.9	19.6	21.5	22.5	23.5	24.5	25.5	25.5	26.0	26.0	26.2	26.4	26.6	26.7	26.8
75.0	13.5	16.5	21.3	22.7	23.7	24.7	25.7	26.7	26.8	26.9	27.0	27.1	27.2	27.3	27.3	27.4
70.0	14.0	17.0	23.0	24.0	25.0	26.0	27.0	28.0	28.0	28.0	28.0	28.0	28.0	28.0	28.0	28.0
65.0	15.0	18.0	25.0	26.0	27.0	28.0	29.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0	30.0
60.0	16.0	18.5	27.0	28.0	29.0	30.0	31.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0	32.0
55.0	17.0	20.0	28.0	30.0	31.0	32.0	33.0	34.0	34.0	34.0	34.0	34.0	34.0	34.0	34.0	34.0
50.0	18.0	20.0	29.0	32.0	33.0	34.0	35.0	36.0	36.0	36.0	36.0	36.0	36.0	36.0	36.0	36.0
45.0	18.5	20.0	29.5	32.5	34.0	34.0	36.0	37.0	37.0	37.0	37.0	37.0	37.0	37.0	37.0	37.0
40.0	19.0	20.0	30.0	33.0	35.0	34.0	37.0	38.0	38.0	38.0	38.0	38.0	38.0	38.0	38.0	38.0
30.0	20.0	20.0	31.0	33.5	35.0	36.0	37.0	38.0	38.0	38.0	38.0	38.0	38.0	38.0	38.0	38.0

This table specifies the spark advance in absolute numbers in degrees BTDC (before top dead center) as a function of RPM and load. The timing you see with a timing light should match the timing table, unless there are spark trim tables in effect that make changes to it. We ordinarily recommend tuning the spark map on a dyno, but here are a few general pointers for an initial setup.

The idea is that the idle and low speed areas are set to around 8-20deg. Usually idle will be around 8-15°, but this depends on your engine's design. If you set the first row in the RPM range as a little under your usual idle (e.g. 600 if your engine idles at 800ish) and add some advance here this can help stabilize the idle, so if the engine stumbles into this area the slight increase in advance will help it to speed up a little so it doesn't stall. The cruising area of the map should have a reasonably high advance, (low to high 30's) as the mixture will be reasonably lean and therefore will give a slower burn.

The overrun area can have an even greater advance, this will allow you to run lean in that section. At Wide Open Throttle (WOT) the spark map needs to be RPM based (analogous to centrifugal advance on an old style distributor) coming in at the right rate relative to engine RPM. Typically, you want it "all in" by about 2800-3200 RPMs for a street performance motor, although many engines will benefit from even more advance after the torque starts decreasing, something you can't do with a mechanical advance.

Note the optimum amount of total advance is not necessarily the most that doesn't cause detonation. For example, with a modern cylinder head design, you might get maximum power at 32°BTDC on a large V8, but might not experience any detonation until 38°- 40°. However, you will still want the advance to come in as quickly as possible (without knocking) up to 32°. This number won't be 32 degrees for all engines; for example, we've found that the new Chrysler Hemi can have problems at more than 22 degrees of timing at full throttle, and many small bore engines with pent-roof combustion chambers run numbers in the mid 20 degree range.

When going into boost (above 100kPa), values will need to be lower than when out of boost. A common rule of thumb is that 1 degree of advance should be removed for every 2 psi (13kPa) of boost; this is simply a rough guide, and lots of things can depend on how much to remove. Advance below 100% load is often simply the flip side of retarding the timing under boost; as cylinder filling decreases, you will need more timing.

Tuning timing under boost is safest on the dyno where the operator can listen out for knock and dial in the advance for best safe power.

3.6 Getting a good idle

Idle tuning has several components. One is how far the throttle is open - if you have an idle valve, this can be adjusted with the Idle PWM Duty Table or Idle Steps which lets you specify the valve opening amount in terms of temperature. If you don’t have an idle valve, you can use a set screw to adjust the throttle blade angle. There is also closed loop idle tuning, which we will cover later in the manual. Closed loop idle tuning targets a specific idle speed. But first, you need to get a stable idle without closed loop tuning. If your idle is already hunting because of a bad tune, putting closed loop on an idle that already has trouble will make it worse. **Start with "Open-loop (warmup)".**

We’ll take some time here to explain how to tune your Megasquirt-3 for a smooth idle. The first key may surprise many novice tuners: **The idle control settings are not the most important thing to adjust to get a smooth idle.**

This may seem counter-intuitive, but consider this: If the engine is idling at constant load with the throttle held at a constant opening, shouldn’t you have a constant idle speed? If your idle surges or hunts under these conditions, and your idle control valve is staying at one opening value, the idle control settings are not the problem. To get a steady idle, you need three things in place: correct fuel, correct timing, and correct airflow. These problems should be addressed individually.

3.6.1 Correct Fuel (mixture)

Engines don’t like to idle lean; a lean air/fuel ratio is one of the biggest causes of a surging idle. On an engine with sequential injection, this is simple enough - get the engine idling at 14.7:1 for gasoline, and you’re good to go. Batch fire engines, however, have issues at idle with intake pulse reversion sending some of the fuel into an adjacent cylinder. So, if your setup only allows batch fire, you’ll need to add a little more fuel so all the cylinders can stay at 14.7:1 or richer. Batch fire engines typically idle best in the mid 13’s, so if you’re leaner than that, put in some more fuel and see if that stabilizes the idle.

You’ll need to make sure the air/fuel ratio is maintained during cold start. If your idle hunts when the engine is cold, but stabilizes when it warms up, try more warm up enrichment.

3.6.2 Correct Timing

Timing is a very useful tool for controlling your idle speed. Ideally, you’ll want the timing to be set to slightly below the timing that would give it the most power - on most piston engines, this typically works out to somewhere in the 10 to 18 degree range. Then you can add a row around 300 RPM below your target idle speed. The timing in this row can be bumped up by around 2 to 4 degrees over the idle timing. The 3d view below shows a spark table set up this way.

Setting up your timing like this has a stabilizing effect on the idle. When the idle speed drops, the timing advances, increasing torque and moving the idle speed back up. The result is an automatic feedback loop that helps maintain a constant idle speed. Not every engine really needs this. However, you need to look out for timing curves that do the opposite. If you’ve specified less timing below the target idle speed than you have at the idle speed, this often makes for an unstable idle as the ECU pulls timing while the idle speed falls. This can make a slight disturbance in idle RPM amplify itself into an idle speed that cycles up and down rapidly.

Too much timing can be trouble. If the timing at your regular idle speed already puts you at the most power output, you have nowhere to go but down. Over-advanced timing will hurt power as well - beyond a certain point, adding more timing will not bump up the idle, and taken to extremes, it can do the exact opposite. So you want to "hold back" a little timing at your idle RPM and run less than the amount of timing that would give it the most power.

3.6.3 Correct Airflow

You’ll need to get the right amount of air to maintain a good idle speed.

How to do this depends on what sort of idle valve you have.

No idle valve at all? Yes, you can run without an idle valve. Just use your idle speed set screw to adjust it open enough that the engine doesn't stall on cold starts. Sure, your idle speed with the engine warmed up will be higher than it could be, and you'll have no way to adjust for stuff like the A/C or electrical loads, but it'll work.

The wiring of idle valves is covered in the Hardware Guide for your product.

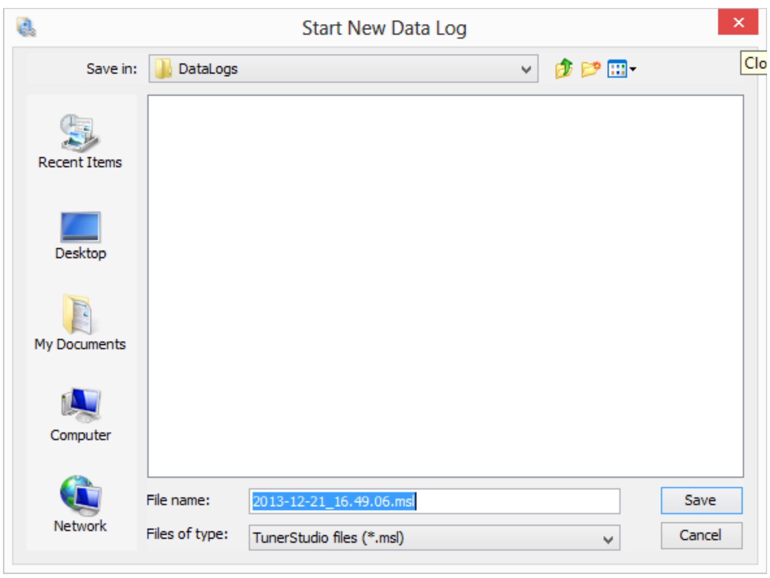
3.7 Datalogging (serial)

Recording datalogs and reviewing them is a key part of ongoing tuning for most vehicles.

See section 4 in the TunerStudio Reference Guide and the (planned) MegaLogViewer Reference Guide.

3.7.1 Starting a datalog

Clicking Data Logging > Start will display the Start New Data Log dialog as shown below:



This will prompt you to save a TunerStudio (.msl) file which will contain the data that is logged from your Megasquirt. By default the file will be saved in the DataLogs directory within the current project directory, although you can choose to browse and save it elsewhere if you prefer. As soon as you save the file the software will start to log data from your Megasquirt into that file.

When data logging is running the "Data Logging" indicator label at the bottom of the Gauge Cluster will turn green.

3.7.2 Stopping a datalog

Data Logging > Stop will cease data logging and the collated data file selected in the above section will be closed. Your data is now ready to be reviewed using MegaLogViewer.

At this stage the "Data Logging" indicator label at the bottom of the Gauge Cluster will return to its usual color.

3.8 SD card datalogging

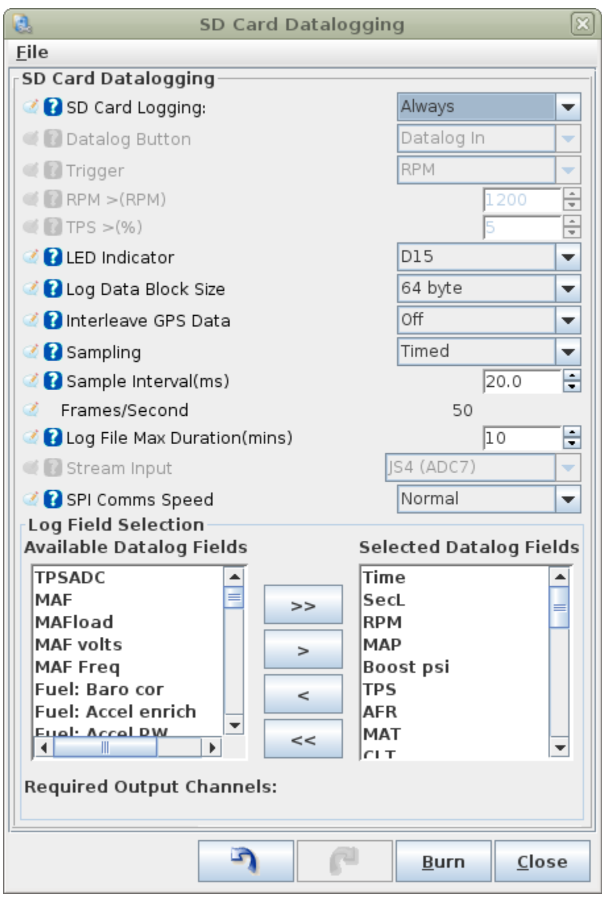
Megasquirt-3 has a built-in SD card slot which allows fast datalogging to a supported SD card without the need for a laptop. This operation can be continual, triggered by an event or button controlled. Once logging is complete, TunerStudio is used to convert the compact SD log data files into regular datalogs that can be viewed in MegaLogViewer software as normal.

3.8.1 Supported cards

Most SD cards and SDHC cards ought to work.
The firmware supports cards formatted with FAT16 or FAT32 format and with or without a partition table.
MMC cards are not supported due to their slimmer size.
Various SD cards have been tested and there is a wide range of speeds observed. MS3 communicates with the card using the "SPI" serial protocol which is more suited to an engine controller, but cannot match the data rates used within video cameras or PCs.

3.8.2 Settings

The configuration for the SD card is on the Datalogging menu in TunerStudio.



See the TunerStudio reference for detail on the settings.

The SDcard system records smaller blocks of data at a higher speed than regular 'serial' datalogs on a laptop. As a result of this, you are required to select which Datalog Fields you want recorded.

An optional extra that can be very useful with the SD logging feature is a realtime clock. Without the realtime clock, the MS3 has no idea what time or date it is and all log files on the card are stored with a generic time-stamp.
There are presently two supported clock options.
a. On the MS3X card there is space to install a clock module (coming soon)
b. The JBperf IO-Extender has a realtime clock which is used via CAN.
Configuration for both of these options is on the "CAN, VSS, gear, RTC" page.

For reference, the MS3-Pro has a clock built-in.

3.8.3 Button / Light

Within "always" mode you do not need the button. It is required for "button" mode to start and stop datalogs. The LED/light output is optional in either mode but can be helpful to more quickly diagnose any error conditions and confirm that logging is happening. It is recommended when using button mode. When using the MS3X spark outputs, it is likely convenient to set the "middle LED" as the LED output as no extra wiring is required.

See your Hardware guide for switch input and light output wiring.

3.8.4 Button Usage

When Megasquirt-3 powers up it will create an empty datalog file ready for logging.
Press and hold the button for over half a second to start a log.
A quick press of the button will end the log.
When extracting the log files you may find a blank log file on the card after your real logs, this is normal.

3.8.5 LED Flash Codes

Off - SD logging not active
On (solid) - Ready but not logging
Fast flashing - Initializing card
Slow flashing - Logging
Pause, sequence of flashes - error code

3.8.6 Log File Extraction

The log files on the card use a compact binary format. Before you can view these files it is necessary to convert them to the conventional log format using TunerStudio.

There are two main methods to retrieve the logs from the SDcard

- Remove the card from the MS3 and insert into your computer

  When inserting the card into a Windows computer, cancel any dialogues that pop up or messages about camera files etc.

  Open TunerStudio and ensure your current Project and tune file are open.
  Go to DataLogging > Import/Conversion > Convert Binary Log
  Browse to find the file you want to import from the SD card, select it and then click ok.
  A progress window should pop up as the log file is converted.
  Repeat for all files needed. Multiple files can be selected.

- Download the data directly over the USB/serial link.

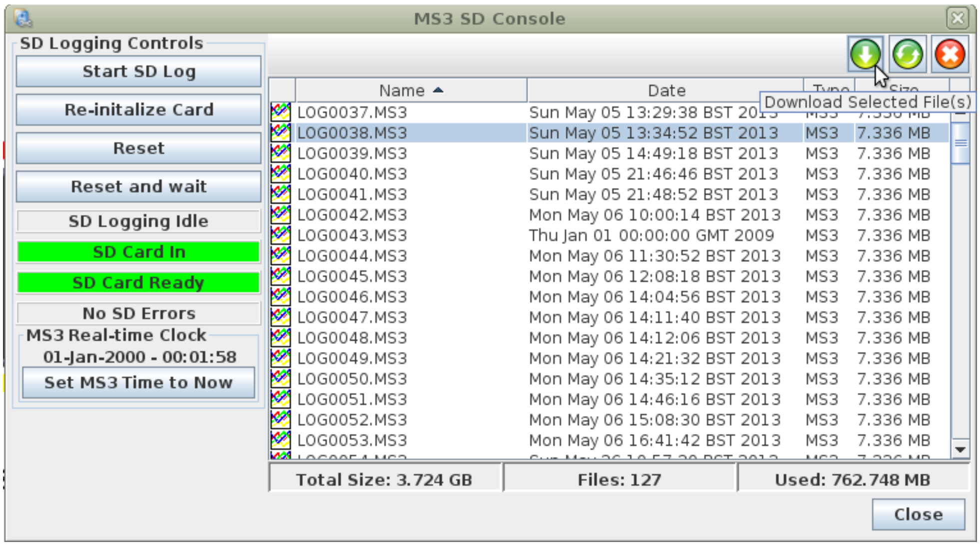
  Turn on Megasquirt, but do not start the engine.

  Open TunerStudio and ensure your current Project and tune file are open.
  Go to DataLogging > Browse/Import SD card.

Find the file you want to import from the SD card, select it and then click the green down arrow to Download Selected File.

TunerStudio will download the raw log files from the Megasquirt and then convert them to a regular log file.

See also the "SDcard readback baud rate" setting on the Communications->Megasquirt Baud Rate screen. Most configurations will support 230400 baud for readback which can halve the download time.



Processed datalogs from either method are stored along with regular datalogs in the Datalogs directory under your project. View the logs with MegaLogViewer.

3.8.7 Error codes

If the card fails to initialize or has another problem you will usually get an error code. This is visible through the "SD error" gauge in TunerStudio or through the LED flash sequences.

Error no.	Name	Meaning / remedy
0	no error	OK
1	sending clocks	Card fault?
2	idle	Card fault?
3	init	Card fault?
4	set blocksize	Card fault?
5	request CSD	Unsupported card?
6	reading MBR	Remove files and reformat card
7	MBR end marker not found	Remove files and reformat card
8	no partition 1 defined	Remove files and reformat card
9	reading partition boot sector	Card fault?
10	reading directory	Card fault?
11	(not used)	
12	reading FAT	Card fault?

13	reading FAT continuation sector	Card fault?
14	writing FAT	Card fault?
15	reading directory	Card fault?
16	writing directory	Card fault?
17	writing log sector	Card fault?
18	unsupported non-FAT16	Remove files and reformat card
19	-	
20	SDHC detection	
21	VCA check pattern failed	
22	VCA voltage rejected	
23	OCR1	
24	OCR2	
25	OCR3	
26	OCR4	
28	processing CSD	Try a lower SPI speed
29	write failed	
30	directory full	Root directory is full, delete some files
31	can't find space in FAT	Card is full, delete some files
32	looks like VBR but not valid	Remove files and reformat card

4: Advanced topics

The settings for all topics here are covered in the TunerStudio Reference Guide. The aim here is to add some background and tuning information beyond the basic settings description for some of the features.

4.1 Sequential Fuel

The Megasquirt-3 can operate fuel in a batch-fire or sequential modes. The different products in the range have different capabilities for sequential fuel.

- MS3base - supports up to 2 injector outputs.
- MS3X - supports up to 8 injector outputs as standard. Up to 12 with modifications.

Consult the Hardware Guide for your product for specific details on wiring or any modifications required.

Sequential and Semi-sequential using the mainboard outputs (MS3base)

The two mainboard fuel channels may be used for full sequential on four stroke one and two cylinder engines if a cam sensor is used to give the full 720 degrees of position information.
When using the two mainboard fuel channels for semi-sequential the injectors will be grouped. On a 4cyl typical 1&4 on channel 1 and 2&3 on channel 2.

Only certain combinations of settings are permitted with semi-sequential on the mainboard outputs. Be sure to follow this table or the code will give you a configuration error and show rpm = 65102.
3 or more cylinders are not supported as sequential on the mainboard outputs.
3 and 5 or more cylinders are not supported as semi-sequential on the mainboard outputs.

Cylinders	Mode	Valid Option
1	Sequential	1 squirt simultaneous
1	Semi-Seq	not possible
2	Sequential	1 squirt simultaneous
2	Semi-seq	not possible
4	Semi-seq	4 squirts alternating

Sequential and Semi-sequential using the MS3X outputs

Semi-sequential in MS3 operates for fuel like 'wasted-COP' does for spark. Each output operates once every 360 degrees with half the fuel required for a full cycle. A cam sensor is not typically required to run semi-sequential, just a crank sensor.
Due to the way the code operates, there are a number of restrictions in this mode and cylinder trim is not possible.
Semi-sequential may, however, be a useful 'stepping-stone' during an install as the injector wiring is identical to full sequential.
For engines above 8 cylinders, semi-sequential operates a single injector output per injector pair. The two injectors must be wired up in pairs. i.e. V12 semi-sequential uses 6 injector channels.

4.1.1 Configuration

The menu items to configure the sequential injection are located on Basic/Load Settings -> Engine and Sequential Settings. The settings on the right are used to configure all the general aspects of sequential injection. The VE, trim and injection timing tables are available on the Fuel menu.

4.1.2 Injector timing

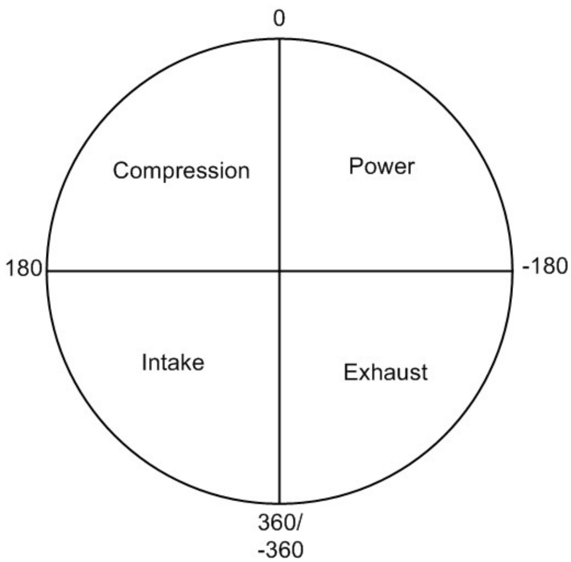
For sequential mode, the timing value is with respect to TDC on the power stroke when both valves are closed (same as ignition).

For sequential siamese mode, the timing is with respect to TDC on the intake stroke which is the start of the injection window for the outer cylinders. And the timing 2 values are only used in sequential siamese mode with the first timing value being for the outer cylinders (1 and 4) while the second is for the inner cylinders (2 and 3).

4.1.3 Sequential/Semi-sequential Timing

The image below shows how the timing values relate to the engine cycle. The timing value of 0 degrees corresponds to TDC between the compression and power stokes. (The same as ignition timing.)

Positive values are for timing in advance of this point up to 360 degrees which corresponds to TDC between the exhaust and intake strokes with 180 degrees being BDC between the intake and compression strokes.



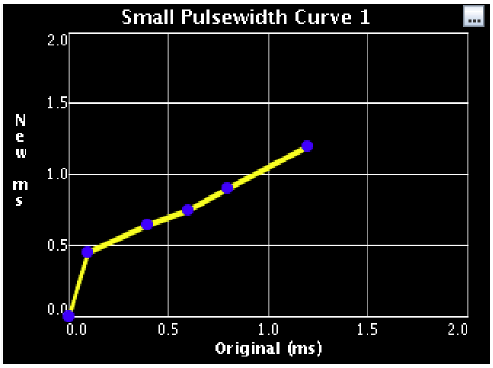
Negative values are for timing retarded from the 0 degree point with -360 degrees being TDC between the exhaust and intake strokes and -180 degrees being BDC between the power and exhaust strokes.

4.1.4 Injector Small Pulsewidths

At most pulsewidths (say above 2ms) injectors behave in a linear manner, where 10% increase in effective pulsewidth gives a 10% increase in fuel flow. This linear behavior is relied upon in the Megasquirt fuel calculations.
However, at small pulsewidths, injectors may behave non-linearly.

The Injector Small Pulsewidths feature allows the user to make a correction and make the non-linear region more linear in nature. Unless you have calibration data for your injectors measured on your Megasquirt, this feature should be left turned off.
The X-axis of the curve is the Megasquirt pulsewidth (before dead-time is applied) and the Y-axis is the required injector pulsewidth (before dead-time) that the injector needs to approximate linearity.
Most injectors observed exhibit an 'S' shaped non-linear region. Nothing happens for the first few 0.1ms after dead time, so a curve that effectively adds on PW to the very low values will help linearise this somewhat.
The non-linear region also frequently shows a 'lump' which will make linearisation tricky. See the dead time measurement section for real measured data.

Here are some example curves based on data from real injectors. Note that no allowance is made for the 'lump' which will distort the results. The injectors were linear by 1.2ms (plus dead time) so the curve is arranged from 0-1.2ms.



Original (m	New ms
0.000	0.000
0.100	0.450
0.400	0.650
0.600	0.750
0.800	0.900
1.200	1.200

Ensure that the start and end points match. i.e. here 0.000, 0.000 and 1.200, 1.200

However, the best approach is to avoid operating injectors in this non-linear region altogether. Size injectors appropriately and use a second set of staged injectors if a large dynamic range is needed.

4.2 Rev Limiter

The rev limiter offers engine protection from over-revving. It is strongly advised that this is enabled on all engines. There are three main types of limiter. After setting the hard limit, you need to pick a method.

- Spark Retard - as the limit is approached the timing is retarded reducing torque
- Spark Cut - as the limit is approached sparks are omitted (cut) to give an aggressive limiter. Not for use with catalysts.
- Fuel Cut - the method that OEMs. Gives an abrupt cut at the hard limit.

4.3 Torque Convertor Lockup

Megasquirt gives the ability to control electric lockup on automatic transmissions. This is for part-electric transmissions (e.g. 700R4, A4LD) but not fully electric (e.g. 4L60E, 4L80E). As a minimum TPS and MAP are used to set lockup criteria. Optionally you can use an enable switch input, a brake light switch to unlock, speed criteria and gear settings.

4.4 Alternator Control

Many current alternator designs use computer control instead of a standalone or internal regulator. This gives the opportunity for additional control strategies for energy saving and optimised battery charging.

The Megasquirt alternator control system allows control of the following types:

- On/Off - basic control, allows alternator "on" to be delayed after start. Typically on the 'L' lamp terminal.
- Open-loop Frequency - Frequency applied sets target voltage (Ford)
- Open-loop Duty - Duty cycle applied sets target voltage (GM)
- Closed-loop Field Control - MS3 uses a closed-loop algorithm to control the alternator field to target a voltage (Chrysler)
- High-Speed feedback field control - MS3 monitors the voltage at 20kHz and switches the field on or

off to match the monitored voltage to the target voltage. (99-05 Miata)

The other settings allow you to define your control strategy. Optionally you can reduce the target voltage during full load (this might be useful during racing to reduce engine load) or increase the target voltage during over-run (allowing a little extra battery charging.)

The current monitor input is for reference only and is not used for any calculations.

Specific wiring is covered in the Hardware Manual for your product.

4.5 MAP Sample Settings

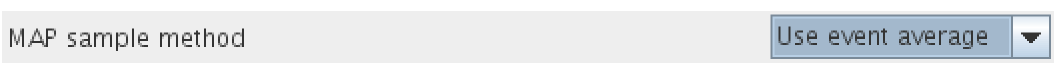
On an engine that is using the MAP sensor (most installs except pure MAF or pure Alpha-N) it is important to get a repeatable and stable MAP sensor signal. Note that **repeatable** is more important than 'accurate'. There are two schemes available

- Event Average - the MAP signal is averaged between each ignition event. This is useful for most engines.
- Timed Minimum - the MAP signal is sampled in a defined sample window and the minimum reading is used. This is useful mainly on 1,2cyl engines.

4.5.1 Event Average

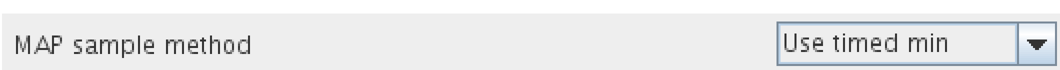
For the majority of engines, this scheme is proven to give the most consistent results. The multiple runners and overlapping intake events on a multi-cylinder engine lead to resonance and pulsations in MAP that vary across the RPM range and load. Taking an average over the cycle side-steps these problems and gives a useable MAP reading for the fuel calculations and other table lookups.

Once this mode is enabled, the other settings on the MAP sampling page are not needed or used.

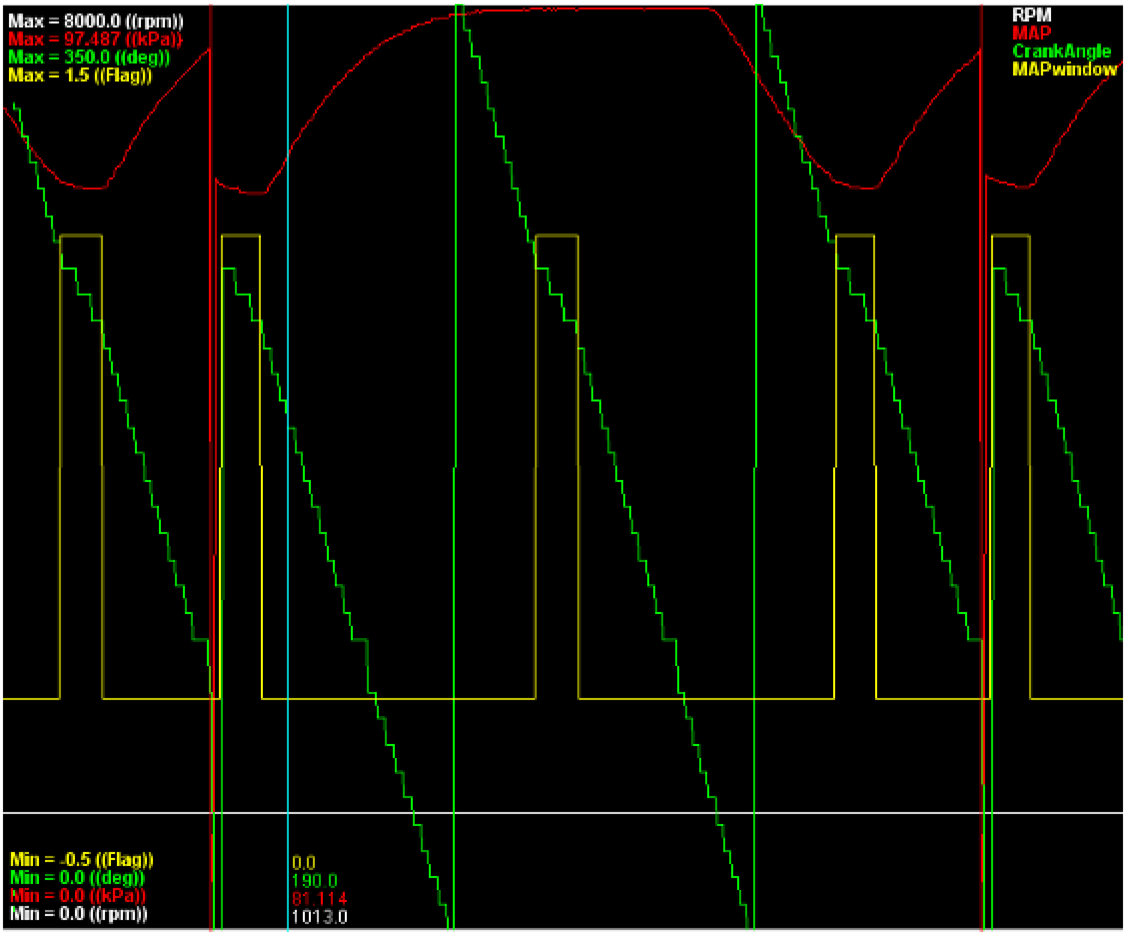


4.5.2 Timed Minimum

For some engines, particularly 1 and 2 cylinder four stroke engines, where there are fewer and more intermittent intake events it is necessary to specify where MAP sampling will occur in order to get a repeatable reading.



Here is a MAP log from a 1cyl 4-stroke engine.



Observe the large variation in MAP signal (red) during the 720 degree cycle. It can be seen that one MAP window (yellow) falls on the intake stroke (where the MAP reaches a minimum) and the next MAP window is on the power stroke (MAP is near atmospheric.) Crank Angle (green) is measured in deg BTDC, so counts down from 720.

On this engine it was important to:

- Set the appropriate MAP sample angle and window width to capture the dip in MAP
- To set No. Events to 2.
- Locate the MAP sensor close to the engine (or use short hose.)

The ideal MAP sample angle may either be determined by:

- using the MAP logger (registered TunerStudio only) and opening the datalogs created in MegaLogViewer to ensure the MAP window aligns with the dip in MAP
- by watching the MAP gauge on the TunerStudio dash and adjusting the MAP sample angle in (say) 10 degree steps until the minimum is found.

Note that on most engines the best sample angle will vary with RPM.

The Window Width should be set so that it covers the dip repeatably.

The way the number of events setting can be demonstrated with the following two examples:

Example 1 - with No. Events set to 1.

Event	Mininum MAP reading in window	MAP value used for fuel calcs.

1	70	70
1	98	98
1	70.5	70.5
1	98	98

This results in the MAP bouncing around between 70 and 98 resulting in unpredictable tuning. Using **Event Average** would be equally poor.

Example 2 - with No. Events set to 2.

Event	Mininum MAP reading in window	MAP value used for fuel calcs.
1	70	70
2	98	**70**
1	70.5	70.5
2	98	**70.5**

This gives a more realistic and more repeatable MAP signal and is a good start for tuning - the atmospheric readings of 98kPa on the 'dead' sample windows are ignored.

The Phase detect threshold is in use on this particular engine. It allows 1cyl and certain oddfire 2cyl engines to run sequential fuel and spark without a cam sensor. The Megasquirt 'looks' at the MAP signal during the reading and compares it to the threshold to determine whether this is an intake stroke.

4.6 Flex Fuel

The flex fuel system allows the use of E85 or other ethanol and petroleum blended fuel mixes to be used when a special fuel composition sensor is fitted. The support is specifically based around the use of a GM or Ford flex fuel sensor which outputs a varying frequency signal depending on the fuel composition. This information can then be used to add more fuel and advance timing. Ethanol blends need more fuel volume than unblended fuel and different advance.

Sensor wiring is covered in the Hardware manual.

4.6.1 Basic System

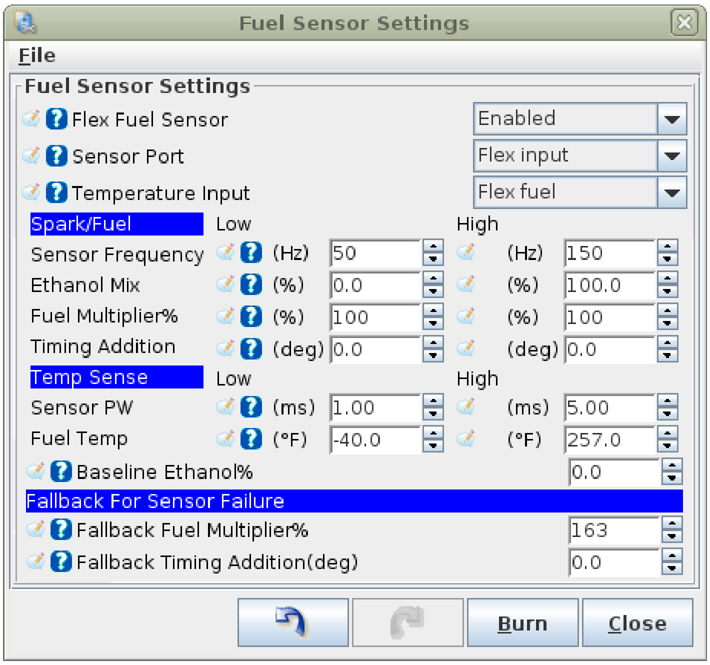
The basic system uses a two-point system (low/high) and interpolates between the values. For many installs this may be sufficient.

It is expected that for 0% ethanol, the settings will use 100% fuel and 0deg timing addition, this means that the basic fuel and timing are in operation. For 100% ethanol the fuel is scaled up and the timing advanced. This is the 'simple' control method. See later for a more complex blended table method.
The fallback setting should be chosen so that in the case of a failed sensor and an unknown fuel mix, you protect your engine from damage. The default settings richen the mixture as if fully ethanol and do not advance the timing. A negative 'addition' will retard the timing.

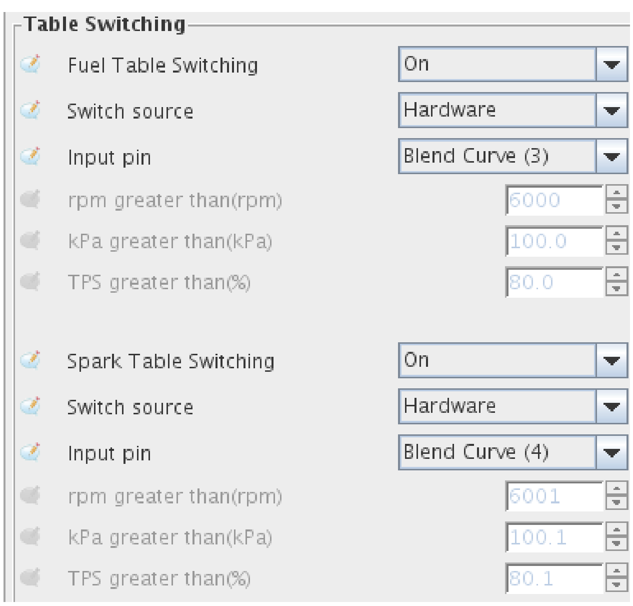
4.6.2 Advanced System

For additional control you may choose to enable table blending based upon the ethanol percentage reported by the sensor. This allows you to develop a pure gasoline tune (VE1) and a pure E85 tune (VE3) and then blend between them. The same applies to ignition timing and boost - with closed loop boost you are able to target more boost with higher ethanol percentages. Be aware of the additional complexity that comes with this additional control, it is advised that the basic method is used initially.

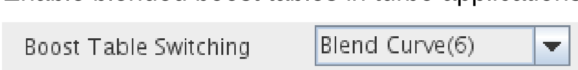
When using the blended tables method, it is suggested that the fuel and timing changes in the main flex screen are cancelled out. i.e. fuel multiplier low/high are both set to 100% and timing addition low/high are both set to zero. With these settings the blended fuel and spark tables will be in full control.
While the examples show all three tables being blended, you could choose to blend one or two depending on your needs.



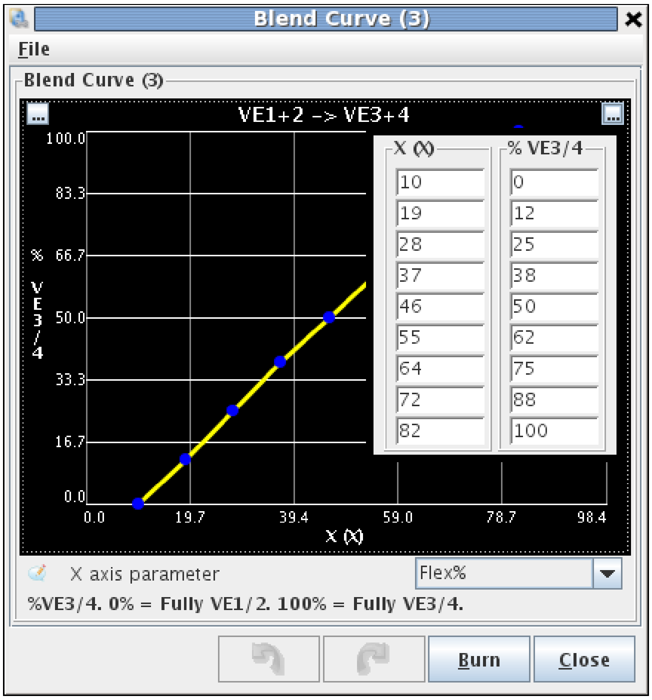
Enable blended fuel and spark tables



Enable blended boost tables in turbo applications.



For the three table sets, define blend curves. This example gives a linear blend between E10 and E82. Fuel is controlled fully by VE1 at 10% ethanol and below. Fuel is controlled fully by VE3 at 82% ethanol and above. The curve can be adjusted to suit your application.



Having enabled the blended tables and defined the curves, you need to define the second table sets.

- For the fuel table VE3 you can start by copying across your VE1 table and then applying a scaling factor all over.
- For spark table Spk3 you can start by copying across your Spk1 table and then adding timing all over.
- For boost table Boost2 you can start by copying across your Boost1 table and then adding boost all over.

One method to copy a complete table is to use **File > Save Dialog Settings** on say VE1, then open VE3 and **File > Load Dialog Settings** and load in the file you just saved.

Having configured your settings, as ever, you should proceed with caution - operate the engine gently and review datalogs to confirm that fuel, timing and boost are behaving as expected. For tuning it is expected that you will need to tune on fully gasoline (E10) to develop good VE1, Spk1, Boost1 tables. Then tune fully on E85 to develop good VE3, Spk3, Boost2 tables. If attempting to use auto-tune, ensure that the correct tables are being adjusted. Time on the dyno is likely a good investment.

For background information see also the Table Blending section.

4.7 Overrun Fuel Cut

This is a method that can achieve minor fuel savings when coasting and off the throttle. When configuring, be sure that it will only activate when you are truly in overrun (e.g. throttle shut and coasting at mid- to high-speed.) Make sure that it does not activate at part throttle or close to idle.

You are advised to start tuning with this disabled to avoid confusion.

4.8 AFR/EGO Control

First, configure the basic settings with reference to the TunerStudio Reference Guide.

A single or multiple sensors can be connected, either locally or via expansion devices. The settings allow you to configure which sensor is reading from which cylinders.

4.8.1 Simple Algorithm with Narrowband Sensor

A narrowband sensor is only accurate at exactly stoichiometric mixtures for the fuel being used (14.7:1 for gasoline). At around 0.5 volts, the mixture is stoichiometric. For leaner mixtures (above 14.7:1 for gasoline, above 1.0 lambda) the voltage dips slightly below 0.5 volts. For richer mixtures, the voltage goes above 0.5 volts. This behavior means that it is not possible to hold an exact mixture when running closed-loop with a narrowband sensor.

Because of this, the best algorithm to use with a narrowband sensor is the "simple" algorithm.

The simple algorithm adjusts the mixture richer if the sensor reads lean, and leaner if the sensor reads rich. It adjusts Controller Step Size percent every Ignition Events per Step. This can lead to a small oscillation in O2-based correction once the AFR reaches close to stoichiometric. This oscillation can be desirable to keep a 3-way catalyst operating correctly.

The following steps are recommended when tuning the simple algorithm with a narrowband sensor:

Ignition Events per Step - When first tuning the engine, this should be set to a fairly low number (4-8) so that if the AFR is very far off, it is corrected quickly. Once the engine is better tuned, this number can be switched to a higher number to gain more stable correction behavior (8-16 or more).

Controller Step Size - When first tuning the engine, this should be set to 2% so that when correcting, the engine reaches stoichiometric quickly. Once the engine is well tuned, this should be reduced to 1% to gain more stable correction.

Controller Auth - When first tuning the engine, this should be set to 20% or higher. Care must be taken to watch how the algorithm is correcting. In some situations, it is possible for the sensor to read very lean when really the engine is running very rich. Once the engine is tuned, this should be set between 5% and 10%.

Engagement Settings - Most of the remaining settings control how and when the closed loop algorithm is engaged. Engagement with a narrowband sensor should happen when the engine is nearly fully warm, 500-1000 rpm above idle, below 80% throttle, below about 80% load, just above the lowest load seen when barely pressing the throttle, and at least 30 seconds after the engine starts. These settings are because the sensor must be hot to operate, must not be used at high load due to the fact that the engine should be operated rich of stoichiometric, and must not be used at very low load because the oscillations will cause the engine speed to oscillate.

4.8.2 Simple Algorithm with Wideband Sensor

Tuning the simple algorithm with a wideband sensor is essentially the same as tuning it with a narrowband sensor with the caveat that the AFR target table is used to set the AFR target. It is still recommended that the EGO algorithm not be used at high throttle position/load due to the fact that the accuracy of the wideband sensor decreases dramatically with pressure and temperature changes caused by high load.

4.8.3 PID Algorithm with Wideband Sensor

It is recommended to start by tuning the 'I' term until the target is reached with minimal oscillation.

Additionally, since the response of most wideband controllers and sensors is linear with AFR, a larger 'P' term can be used to help correct for fast changes in AFR. Caution must still be used however since there is a significant delay between the amount of fuel being injected changing and Megasquirt registering an AFR change as a result.

Finally, a small amount of 'D' term can be used to help slow response during very fast changes. This helps reduce overshoot of the target.

4.9 AFR Safety System

The "AFR Safety" System is designed to help you protect your engine from unwanted dangerous lean conditions. Typically this will be used for wide open throttle and/or boosted conditions. It only makes sense to use this feature with a wideband lambda sensor on an already well tuned engine.
The system compares the wideband EGO input against your existing AFR target table and a safety limit difference table.
If this situation persists beyond a time you specify, the engine will be shutdown as a protective measure until you get out of boost, off the throttle and down in revs.
NOTE This system will only work if configured correctly and will not protect your engine against bad tuning or knock! However, it is likely it would detect a failing fuel pump or regulator. A single faulty injector could easily go un-noticed with a single wideband reading the average AFR.
This shutdown mechanism is shared with the EGT over-temperature settings.

Each install is different and it is up to the installer and tuner to decide on safe limits !
The example data in the screenshot above has the system only active above 95kPa and 2500rpm. Once in this region the safety limit difference table is the controlling factor.
For example if at a certain point you have set your target AFR (AFR table) to be 11.9 and you then decide that a worst case of 12.3:1 AFR is allowable, the safety limit table would have 0.4 in that position. i.e. allowed AFR = target AFR + safety limit = 11.9 + 0.4 = 12.3

If then your wideband EGO sensor reports 12.4 or leaner, the warning output will be activated immediately. (Typically this would connect to a dash mounted warning light.)
If this situation persists beyond the time limit (shown as 0.5 seconds here) then the shutdown mode begins. If the AFR returns to a "safe" level before the time limit elapses no shutdown takes place. Review datalogs before setting this time. The lower it is, then the sooner the shutdown kicks in. This could help protect your engine but setting it too low may cause annoying false alarms.

The first step in the shutdown procedure is to cut spark for a specified time (shown as 0.5 seconds here.) During this period fuel continues with the intention of cooling the engine internals as the lean condition is likely to have generated excess heat. On vehicles fitted with a catalyst, 0.0 should normally be specified as catalysts are not designed to cope with raw fuel.

After the "kill spark for" time, fuel is also cut.

Normal engine operation will resume only when your Throttle, MAP and RPM are less than the limits you set.

In operation, if you trip the shutdown, the engine will suddenly die. While this will be irritating, hopefully it is less annoying than the engine rebuild you might have needed.

4.10 Fuel Pump and Pressure

The fuel pump and pressure control system allows for:

- On/Off - relay operated pump with traditional pressure regulator
- Open-loop PWM - typically used for variable speed pumps
- Closed-loop PWM - typically used for returnless systems

The recommended install uses an On/Off pump and a vacuum referenced regulator.

Some fuel systems (GM) use a fixed line pressure, the ECU can automatically adjust the PW to compensate for this. (Otherwise the VE table would have a larger range of values.)

4.11 Knock Detection

Note: While this is a fairly good setup as far as knock sensing goes in a race ECU, it isn't perfect. This can provide an extra layer of protection, but never use knock sensing as your only spark tuning tool!

Spark knock is the sound of abnormal combustion in an engine. Once combustion in a spark-ignition internal combustion engine is initiated by a spark, the flame front is designed to spread from the spark plug and travel across the combustion chamber rapidly and smoothly. As the flame front propagates across the chamber, the remaining unburnt air-fuel mixture can ignite spontaneously (auto-ignites) before the flame front arrives, due to the increasing pressure and temperature in the combustion chamber. When this occurs, there is a sudden jump in the pressure in the cylinder. This causes in the characteristic knocking or pinging sound. It is most common at low-mid rpms and high load, such as ascending a hill in too high a gear.

Prolonged heavy knock is likely to cause severe and permanent engine damage and must be avoided.

It is a common misconception that engines make most torque just before knock. In reality, there isn't much of a connection between the knock threshold and the timing that makes best torque. A knock resistant engine may start losing power well before the onset of knock, while a knock limited engine may have the point where it makes best power past the knock threshold (and not safe to reach without higher octane fuel.) Ideally, set timing on a dyno to achieve maximum brake torque (MBT) timing. Even with a well tuned engine, factors such as fuel octane, intake air temperature, coolant temperature, engine age and condition, air/fuel ratios, air density, altitude and humidity and others can push the engine from a safe condition to borderline knock or worse. The knock control system is a safety measure designed to retard timing under these conditions and safeguard the engine.

The MS3 can be used with an internal module that can process one or two knock sensors. It also allows input from an external module that gives an on/off, "knock" or "no knock" signal. The knock sensor inputs would use dedicated wires intended for connecting directly to a knock sensor. External modules would use the digital input wires instead.

There are two main types of knock sensor: resonant and wideband sensors. The resonant sensor is tuned to a particular frequency and is only likely to work on an engine with a very similar bore size. (Bore size determines 'ping' frequency.) The wideband sensors give a lower voltage output but operate over a wider frequency range, and can usually be applied to different engines.

The GM resonant knock sensor pictured above is tuned to a specify frequency, like a tuning fork. When this frequency is applied to the sensor (through its connection to the engine), a piezoelectric crystal inside the sensor generates a small voltage (~1 volt), much like a microphone. As an example, some Corvette knock sensors (GM PN 1997562, 1997699, or Standard Motor Products KS45, KS46, KS49, or KS117) have a design frequency of 5200 Hz, and they produce a signal between 4800 Hz and 5600 Hz.

Most Bosch sensors with a hole through the center are wideband knock sensors, adaptable to many engine types.

The sensor should be mounted near the top of the engine block, as close to the center as practical. Do not mount it close to noisy components such as the fuel pump or cam shaft lifters. Mounting the knock sensor in the cylinder head is not a good idea because of valve train noise. Finding a suitable location for the sensor is crucial. Wherever possible, use the location specified by the manufacturer for that engine family.

Ideally, you will be able to find a suitable threaded hole in your block to which you can mount the sensor. If not, an alternative is to drill and tap the block, or thread a steel adapter to accommodate the sensor on one end and a stub with the thread to match those in an existing pretapped boss in your block. Note that it may be necessary to change the sensor location if you cannot isolate engine noise while allowing MegaSquirt to identify knock.

If you choose to drill and tap your block, choose a thick area of the block with a boss that is at least 3⁄4" (19 mm) thick. Drill a 1⁄2" (13mm) hole. The hole should be 0.500" to 0.625" (13 mm to 16 mm) deep. Make absolutely sure that it is safe to drill a hole this size - a poorly placed hole can trash the engine block.

The GM knock sensors have a 3/8" NPT thread. Tap the hole with a 9/16" UNF starter tap. Go in 4 turns of the tap to begin with, clean out the chips and try the sensor for fit. Keep tapping one turn at a time until the sensor threads in 4 to 5 turns with hand pressure. Stop tapping when the sensor will screw into the hole 6 to 7 threads with a wrench. Note that the thread on the knock sensor is a tapered thread.

The Bosch sensor can be more simply attached by bolting it to the block. It uses a standard EV1 style injector connector.

The settings for knock control are covered in the TunerStudio reference.

4.12 Wankel Rotary Engines

Stock FC, FD, RX8 ignition types are supported as well as 3 and 4 rotor variants. Tach input will typically come from a 24/1 Nippondenso CAS.
Only MS3X injector and logic spark outputs are documented here.

The tach input and spark output will need configuring also, see the Hardware Guide for your product for Rotary specific details.

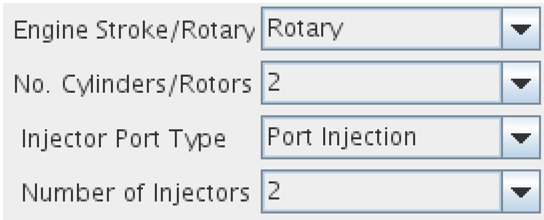
The oil metering pump is not presently supported, use pre-mix.

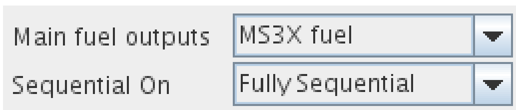
4.12.1 FC 2 rotor

The FC engine uses a single logic wasted spark coil for leading spark on both rotors and combined switched logic coil for trailing ignition.

Settings:

- Engine stroke = rotary
- Rotors = 2
- Injectors = 2
- Fuel hardware = MS3X
- Number of coils = wasted spark
- Rotary trailing mode = FC
- Spark hardware = MS3X
- Staged injection = on





Wiring:

Spk A Leading coil (IGt-L)
Spk B Trailing coil (IGt-T)
Spk C Trailing coil select

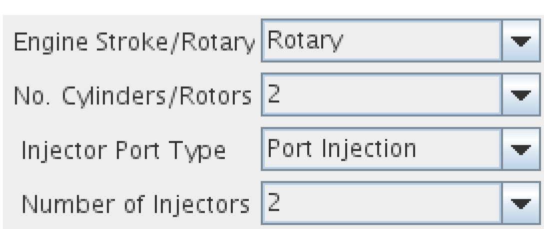
Inj A Front rotor primary injector
Inj B Rear rotor primary injector
Inj C Front rotor secondary injector
Inj D Rear rotor secondary injector

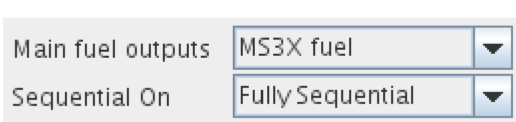
4.12.2 FD 2 rotor

The FD engine uses a single logic wasted spark coil for leading spark on both rotors and individual logic coils for trailing ignition.

Settings:

- Engine stroke = rotary
- Rotors = 2
- Injectors = 2
- Fuel hardware = MS3X
- Number of coils = wasted spark
- Rotary trailing mode = FD
- Spark hardware = MS3X
- Staged injection = on





Wiring:

Spk A Leading coil (IGt-L)

Spk C Front trailing coil

Spk D Rear trailing coil

Note that Spk B is not used

Inj A Front rotor primary injector

Inj B Rear rotor primary injector

Inj C Front rotor secondary injector

Inj D Rear rotor secondary injector

4.12.3 RX8 2 rotor

The RX8 engine uses four individual logic coils for each spark plug.

Settings:

- Engine stroke = rotary
- Rotors = 2
- Injectors = 2
- Fuel hardware = MS3X
- Number of coils = coil on plug
- Rotary trailing mode = n/a (grayed out)
- Spark hardware = MS3X
- Staged injection = on

Wiring:

Spk A Front Leading coil

Spk B Rear Leading coil

Spk C Front trailing coil

Spk D Rear trailing coil

Inj A Front rotor primary injector

Inj B Rear rotor primary injector

Inj C Front rotor secondary injector

Inj D Rear rotor secondary injector

4.12.4 Three rotor

Designed to run COP ignition for leading and trailing on each rotor. (Six coils.)

Settings:

- Engine stroke = rotary
- Rotors = 3
- Injectors = 3
- Fuel hardware = MS3X
- Number of coils = coil on plug
- Rotary trailing mode = n/a (grayed out)

- Spark hardware = MS3X
- Staged injection = on

Wiring:

Spk A	Leading coil A
Spk B	Leading coil B
Spk C	Leading coil C
Spk D	Trailing coil A
Spk E	Trailing coil B
Spk F	Trailing coil C

Inj A	Primary injector A
Inj B	Primary injector B
Inj C	Primary injector C
Inj D	Secondary injector A
Inj E	Secondary injector B
Inj F	Secondary injector C

The coils and injectors need to be wired in firing-order sequence. The outputs fire A, B, C, D, A....

4.12.5 Four rotor

Designed to run COP ignition for leading and trailing on each rotor. (Eight coils.)

Settings:

- Engine stroke = rotary
- Rotors = 4
- Injectors = 4
- Fuel hardware = MS3X
- Number of coils = coil on plug
- Rotary trailing mode = n/a (grayed out)
- Spark hardware = MS3X
- Staged injection = on

Wiring:

Spk A	Leading coil A
Spk B	Leading coil B
Spk C	Leading coil C
Spk D	Leading coil D
Spk E	Trailing coil A
Spk F	Trailing coil B
Spk G	Trailing coil C
Spk H	Trailing coil D

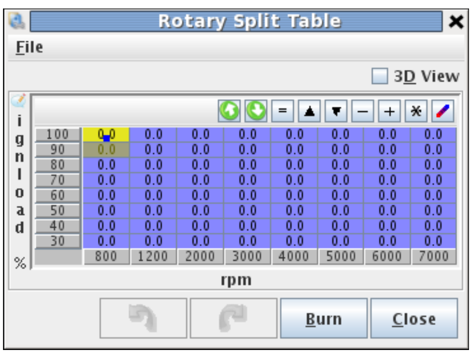
Inj A	Primary injector A

Inj B	Primary injector B
Inj C	Primary injector C
Inj D	Primary injector D
Inj E	Secondary injector A
Inj F	Secondary injector B
Inj G	Secondary injector C
Inj H	Secondary injector D

The coils and injectors need to be wired in firing-order sequence. The outputs fire A, B, C, D, A....

4.12.6 Rotary Split table

This allows you to specify the split in degrees between leading and trailing sparks. Positive numbers mean the trailing is later (normal). Negative means that trailing is fired before leading.



4.13 Closed-Loop Idle

Before trying to tune closed loop idle speed control, you must configure open-loop (warmup) idle speed control. With open-loop control, a higher step-count or duty should yield higher RPM. Make sure that this is the case, and that smooth idle speed can be attained with open-loop before moving on to closed loop control.

There are two main things to tune when tuning closed-loop idle speed control:

- Conditions for entering PID control
- PID gains

It is recommended that tuning is done in stages. For example, PID cannot be tuned if the code is never entering the PID loop. Because of this it is a good idea to start by tuning the conditions for entering PID control.

These settings include:

- Idle Activation TPS threshold - Set as low as possible. If the TPS has a bit of noise, set it to around 1%, otherwise set it to 0.3%-0.5%.
- RPMdot threshold - Use this setting so that the code can tell the difference between decelerating with closed throttle (engine braking) and sitting at one RPM. Set this as low as possible without being below what is normal RPMdot jitter with the engine RPM not changing. Typical values will be between 50 and 75 RPM/sec.
- Max decel load - The code assumes that if MAP is lower than this setting, the driver must be

decelerating, and not "locked out" of the PID loop. Set this to a value just under the load seen with an idle slightly higher than the current target RPM + the Idle Activation RPM Adder. This can be done with the idle test mode, and setting the valve position manually.

- PID delay - This should be set so that the RPM dropping on throttle lift can come to a rest slightly higher than the target RPM, and become stable there. Between three and five seconds normally works the best.
- Crank to Run Taper - This setting controls how long after starting the code will delay before entering PID. Between three and five seconds works well for this setting.
- PID disable RPMdot - A good value for this setting will typically be in the 200-400 RPM/sec range. If the engine speed suddenly accelerates with no throttle input (like if the clutch is engaged while the car is rolling and in gear), it must accelerate at a rate greater than this setting before the PID code will be disengaged. Setting this value too high can lead to stalls after engaging the clutch in this manner.

To tell whether the code is entering PID idle control, the "CL Idle" indicator in TunerStudio must be used. If the current gauge cluster in TunerStudio does not include this indicator, temporarily switch to a cluster that does or right-click on an indicator and change it to "CL Idle".

The idle speed control algorithm was designed to emulate OEM car idle speed regulation. The sequence of events that the code was designed to follow are listed below:

1. Throttle Lift - On throttle lift, the code opens the valve to the value learned in the last iteration of the PID loop + the dashpot adder. The logic here is that the last learned value should result in an RPM close to the target RPM. The dashpot adder is added so that when RPM settles, it settles to an RPM slightly higher than the target. This is in case the air conditioning was turned on or IAT increased or anything else that might make RPM lower than the last time the PID code ran.
2. RPM settles - After throttle lift, eventually the clutch is pushed in and RPM drops to wherever it will settle given the learned value + the dashpot adder. Hopefully the idle has settled to an RPM that is less than the commanded target + the Idle Activation RPM adder. If so, then the code will wait for the amount of time specified by the PID delay, and then enter PID control. If RPM settles above the commanded target + Idle Activation RPM adder, the code then starts checking the PID lockout detection conditions. Assuming those conditions are met, the code will still enter the PID loop after the amount of time specified by the PID delay.
3. PID control activates, RPM starts dropping to target - After the PID delay expires, the PID code will be activated. RPM will slowly drop to the target over the number of seconds specified by the PID ramp to target time.
4. Normal idle speed reached - RPM reaches the commanded target. PID continues regulating RPM until the throttle is pressed.

Once the code is reliably entering PID on every throttle lift, it is time to actually tune the PID code to reach and hold the RPM target.

The settings that are associated with or affect the operation of the PID algorithm are listed below:

- Idle Valve Minimum and Maximum Duty/steps - These should be set to the minimum and maximum values that should be used during PID loop and driving operation. In addition, having these set further apart results in the PID loop being more sensitive (making changes to the output given much smaller changes in input).
- PID Control Interval - This controls how often the PID code runs. Setting this too high can result in sluggish response to sudden changes in load, such as the Air Conditioning being turned on. Setting it too low can result in the loop being overly sensitive to RPM changes. Typically 100ms works well.
- Tuning Mode - Basic internally sets P=100, I=100, D=0. Advanced allows full control

- Sensitivity Slider - This sets how sensitive the PID control is, start low and work up if response is too slow.
- PID controller gains - These control the actual response of the code to changes in RPM, as well as how well the code will reach the target. Tips for tuning these are listed below.

The following basic steps should be used for tuning the PID controller gains in advanced mode.

1. Zero all the gains - Set all the gains to 0%. This is so that the effects of tuning the I-term in the next step are not confused with the effects of any other setting.
2. Tune the Integral (I) gain - The Integral gain is the only term that controls whether the code actually reaches its target. Higher values for Integral gain will result in the code being able to get closer to the commanded target; however, a value that is too high will result in oscillation. The easiest way to determine a good value for the I term is to keep increasing it until oscillation occurs, then slightly lower it. If this value is increased to 200% without reaching a point where oscillation occurs, then the RPM with valve opened setting can be decreased as far as necessary, and the open duty/steps setting and closed duty/steps setting can be made further apart to make the PID loop more sensitive.
3. Tune the Proportional (P) gain - After tuning the I gain so that the RPM reaches the commanded target without oscillation, the P gain can be tuned. The best way to tune this is to set it as high as possible without getting any oscillation. After setting this, try turning on the air conditioning or other accessories that normally lower RPM or increase load. When these accessories are turned on, the RPM should dip a bit then recover (the valve position should increase significantly). Using longer PID ramp to target times can also make it so that when the PID algorithm engages, a higher P gain can be set without causing oscillation.
4. Tune the Derivative (D) gain - For most users, use of the D gain should not be necessary. It substantially dampens the response of the loop.

Some final tips:

- Idle Fuel Tuning - Before even attempting to tune Closed-loop Idle speed control, tune the area around idle so that if RPM goes up or down or load goes up or down, the AFR stays close to the same value. Changing AFR can affect idle speed, which can then cause the PID code to try to correct, getting into an unrecoverable oscillation.
- Spark Advance - Ensure that the ignition timing around idle is fairly smooth. Large changes could lead to oscillation.
- The 'Initial Value' table can be used to 'prime' the closed-loop algorithm with a set of known good idle valve positions.

4.14 Idle Advance

The Idle Advance feature is useful to fine-tune ignition timing at idle. It is particularly useful to help catch sudden load increases on the engine at idle by increasing timing when load increases to help the engine generate more power, keeping RPM from dropping severely. The distinct idle-advance feature can be useful for engines where the idle conditions (RPM/MAP) are similar to driving conditions.

Idle Advance Tuning

There are two main types of settings to tune for the Idle advance feature:

- Idle Advance engagement settings - These settings control the conditions under which Idle Advance will engage.
- Idle Advance Timing curve - This curve controls the actual ignition timing once all the Idle Advance engagement conditions have been met.

Tuning Idle Advance Engagement Settings

The Idle Advance engagement settings should be set so that idle advance will engage in roughly the same conditions that occur during normal, warmed-up idle.

Settings recommendations:

Go to idle advance when:

- TPS is below - This setting should be set as low as possible. Typically settings between 0.5% and 1% should be used. If numbers that are too low are used, then idle advance may not engage if there is some play in the throttle body or there are minor electrical fluctuations that cause the closed TPS % to vary. If numbers that are too high are used, then idle advance may engage at undesirable times.
- and RPM is below - This setting should typically be set just above the desired idle RPM, and below the lowest RPM at which the driver normally drives in gear. For example, if the desired idle RPM is 800, then a good value for this setting is 1000.
- and load is above - This setting should be set just below the load value seen during a normal idle with no load on the engine.
- and CLT is above - This setting should be set to the temperature at which the engine idle characteristics no longer change. Generally this is when the engine is fully warm.
- and after delay - This setting should be set to a value that is long enough for the engine RPM and load to become stable before idle advance engages.

Tuning Idle Advance Timing

In general, the most stable idle is reached by decreasing the idle timing, and increasing the amount of air entering the engine (using an idle air valve or similar). As such the idle advance timing should be as low as possible while retaining a smooth idle. Since less timing is used during normal idle conditions, as load increases, the timing should also increase to counteract RPM decrease when the load increases.

4.15 Enhanced Accel Enrichment (EAE)

The Enhanced Acceleration Enrichment feature is based on the concept that the fuel injected does not all enter the engine on every injector squirt. Instead, a portion of the fuel collects on the port and/or intake runner walls. The fuel collected there forms a puddle, from which some fuel enters the engine on every intake event.

During steady-state conditions (such as cruise or idle), the amount of fuel entering the puddle, the amount of fuel leaving the puddle, and the amount of each injector squirt going directly into the engine reach a state of equilibrium. However, during throttle opening or throttle closing transient conditions, the amounts of fuel entering the puddle, leaving the puddle, and going directly into the engine change. Until equilibrium is reached again, the amount of injected fuel must change to ensure that the intended amount of fuel (from the normal fuel equations and VE table lookup) still enters the engine.

For reference, the accelerator-pump on a carburettor is a rudimentary method to cope with this same wall-wetting issue.

The EAE algorithm tracks the various quantities and adjusts the amount of fuel injected accordingly, but does so based on several adjustment curves and settings. These are detailed in the next section.

The EAE curves are described in the TunerStudio Reference Guide.

4.15.1 Tuning EAE

Since EAE's main purpose is to ensure that the proper amount of fuel specified by the VE table (and other enrichments) actually gets into the engine, it is essential that the VE table as close as possible to correct before tuning EAE.

Since EAE must track the amount of fuel collected on the port walls in order to function, it is also necessary for EAE to be enabled at all times.

The following procedure should be followed to tune EAE:

1. Tune VE, Warmup enrichments, and all other fuel-related features.
2. Turn off normal Acceleration enrichment by setting the TPSdot and MAPdot thresholds to extremely high numbers.
3. Make sure that the VE table covers all the way down to Cranking RPM and to extremely low kPa values.
4. Enable EAE in the Acceleration Enrichment dialog box.
5. With the engine at a steady RPM in a high gear, step on the throttle SLOWLY, and note the response feel and the AFR.
6. At the same RPM, lift off the throttle, and note the response feel and the AFR.
7. Tune the EAE Adhere-to-walls curve and EAE Sucked-from-walls curve until AFR and response are smooth and stable.
8. Make sure that the throttle movements used are small and slow, allowing the AFR to reach steady-state before moving the throttle again. Make sure that the whole load range is covered, and that every load seen during engine operation is covered by each of the curves.
9. Choose a few other RPM ranges, and slowly step on and release the throttle. Tune the EAE Adhere-to-walls RPM correction and EAE Sucked-from-walls RPM correction curves until the response and AFR are correct at the RPMs chosen. Typically Idle and high cruise RPMs should be chosen. High cruise RPMs for example are when speed is maintained but the gear selection is reduced by one or two gears.
10. Shut off the engine, and allow it to cool completely. Start the engine; as the coolant temperature increases, adjust the EAE Adhere-to-walls CLT correction and EAE Sucked-from-walls CLT correction curves so that response and AFR are stable.
11. Once small, slow throttle movements are tuned, larger ones can be verified, as well as normal driving with gear shifts
12. Finally, try to quickly blip the throttle while free-revving. If response is slower than desired, a very small amount of TPSdot or MAPdot acceleration enrichment may be re-enabled. Take care to only use it for high TPSdot values and use very little. Just enough to get EAE to respond is all that is required.

4.16 X-Tau

X-Tau is an alternate model based acceleration enrichment, with slightly different equations. This uses a number called X to represent the percentage of fuel sticking to intake walls, and tau to represent the amount of time it takes for the fuel that sticks to the walls to reach the cylinder. While EAE uses a load scalar curve, X-Tau uses two separate, RPM based curves for acceleration and deceleration.

The X-Tau algorithm uses the rate of change of the MAP reading, not the absolute value of the MAP reading, to determine which of these two curves to use. At steady MAP, it blends between the two.

4.16.1 X-Tau tuning tips

You should start with low X and low Tau, such as the default values. You should then see if the X-Tau helps. Start by adjusting the X factor. If that doesn't help, try increasing the Tau time table entries in the areas where you are having troubles with lean spots (engine coughs on accel).

The Tau time table is deliberately conservative, so in most cases it will require increasing, by perhaps 50% to 100%.

You may need to try adjusting the lag filter values for the MAP sensor for less filtering. The X-Tau mode needs to see the rapid rate-of-change of MAP and the lag filters can reduce this if set too low. Generally you do not want to set it to less than 50%, and may need to set it to 70% to 80%.

As you dial-in the X-Tau parameters, the engine will likely become overly rich due to both normal and X-Tau enrichments being applied. You can try reducing the existing TPSdot or MAPdot based accel/decel enrichments, which do NOT go away when you specify the X-tau option and increasing the X-tau variables. (Note that you may have to increase the cold accel multiplier as you do this.)

4.17 Boost Control

The most basic boost setup uses spring pressure only and connects the manifold (or turbo outlet) directly to the wastegate actuator. For systems that require more control, the Megasquirt software offers options for electronic boost control.

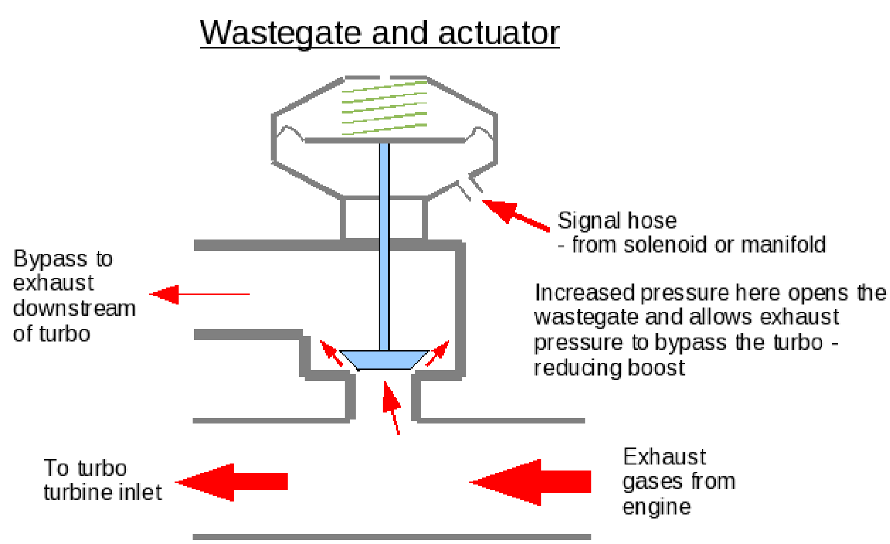
- Single-solenoid - the common OEM style of control
- Dome control - drag racing style with CO2 pressure and twin solenoids

4.17.1 Single-solenoid Boost Control

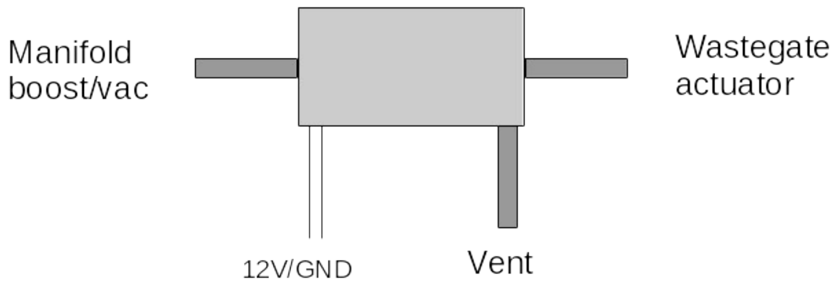
For simpler OEM-style boost control a single solenoid can be used to control the wastegate. (This was the only boost control available before 1.5 firmware.)

The wiring of your solenoid is covered in the Hardware Guide for your product.

Solenoid and wastegate plumbing are common across the range. There are many different plumbing schemes available, just one is presented here. If the wiring to the solenoid breaks, it should fail safe and run the engine at minimum boost (spring.)

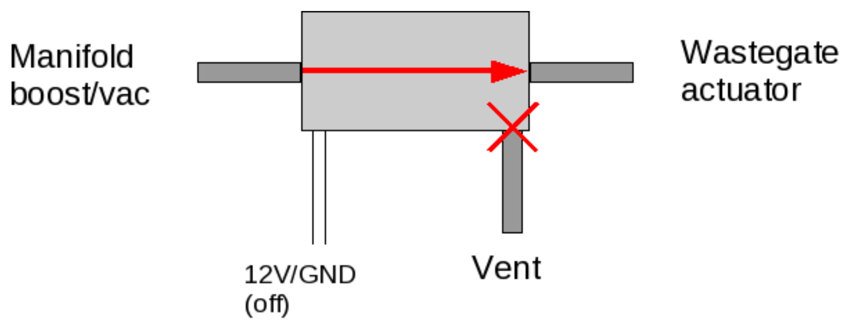


Boost control solenoid example



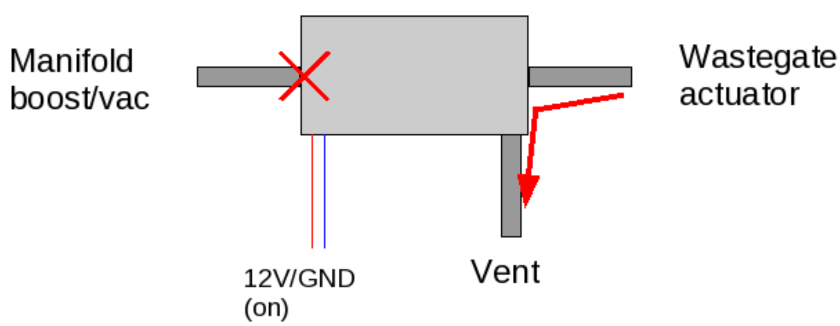
Note! Solenoids vary in outlet arrangement, check yours!

What the solenoid does in different conditions:



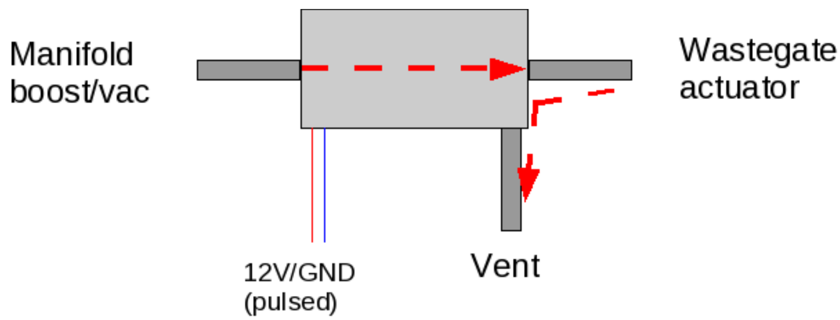
No voltage applied

Failsafe situation.
Full pressure goes to open wastegate actuator.
Spring pressure boost only



12V voltage applied

Solenoid blocks pressure.
Wastegate actuator pressure released through vent.
Wastegate closed.
Maximum boost.



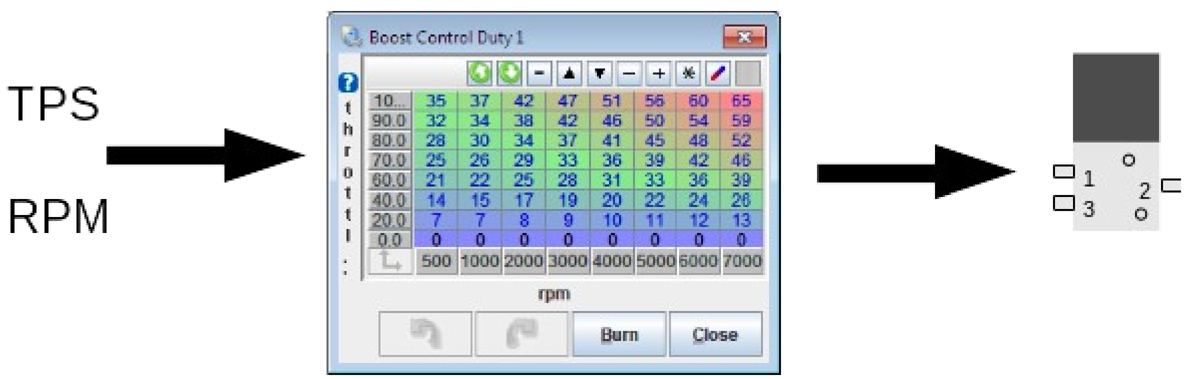
Pulsed 12V voltage applied (PWM)

Solenoid allows some pressure to pass.
Wastegate actuator partly pressurised.
Wastegate partly closed.
Partial boost.

4.17.2 Open-loop boost control concepts

With open-loop boost control, the lookup tables directly specify the solenoid duty. During tuning the desired duty will be determined. No attempt is made to match a specific boost pressure and no allowance is made for changes due to external conditions such as the air temperature and pressure.

Single table open loop boost duty



throttle \ rpm	500	1000	2000	3000	4000	5000	6000	7000
10...	35	37	42	47	51	56	60	65
90.0	32	34	38	42	46	50	54	59
80.0	28	30	34	37	41	45	48	52
70.0	25	26	29	33	36	39	42	46
60.0	21	22	25	28	31	33	36	39
40.0	14	15	17	19	20	22	24	26
20.0	7	7	8	9	10	11	12	13
0.0	0	0	0	0	0	0	0	0

Table switching allows alternative duty table.

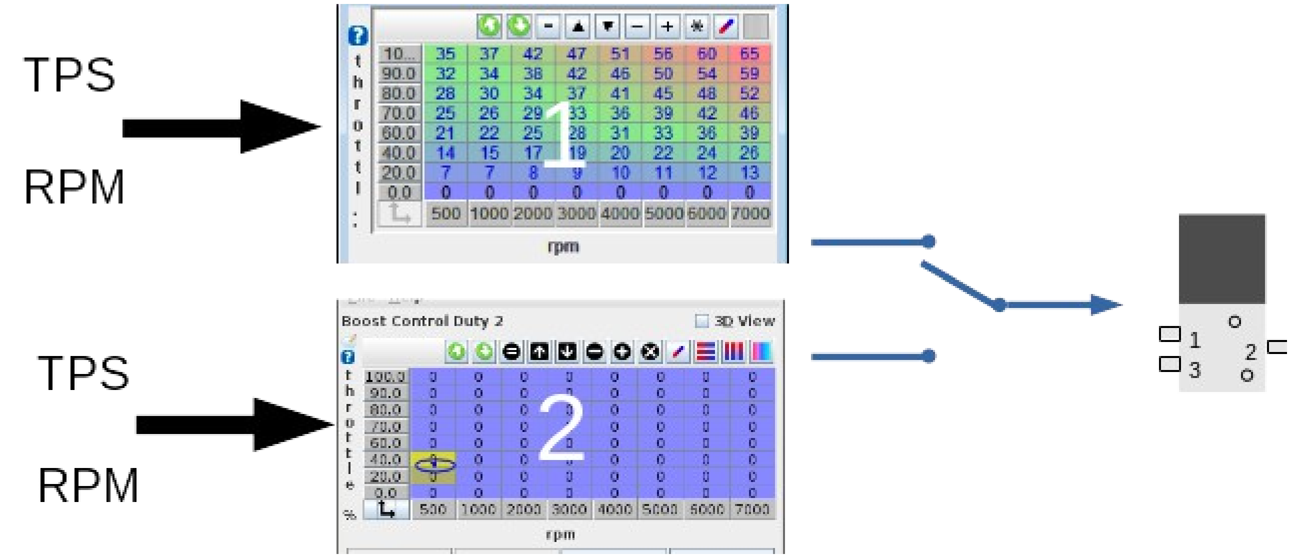
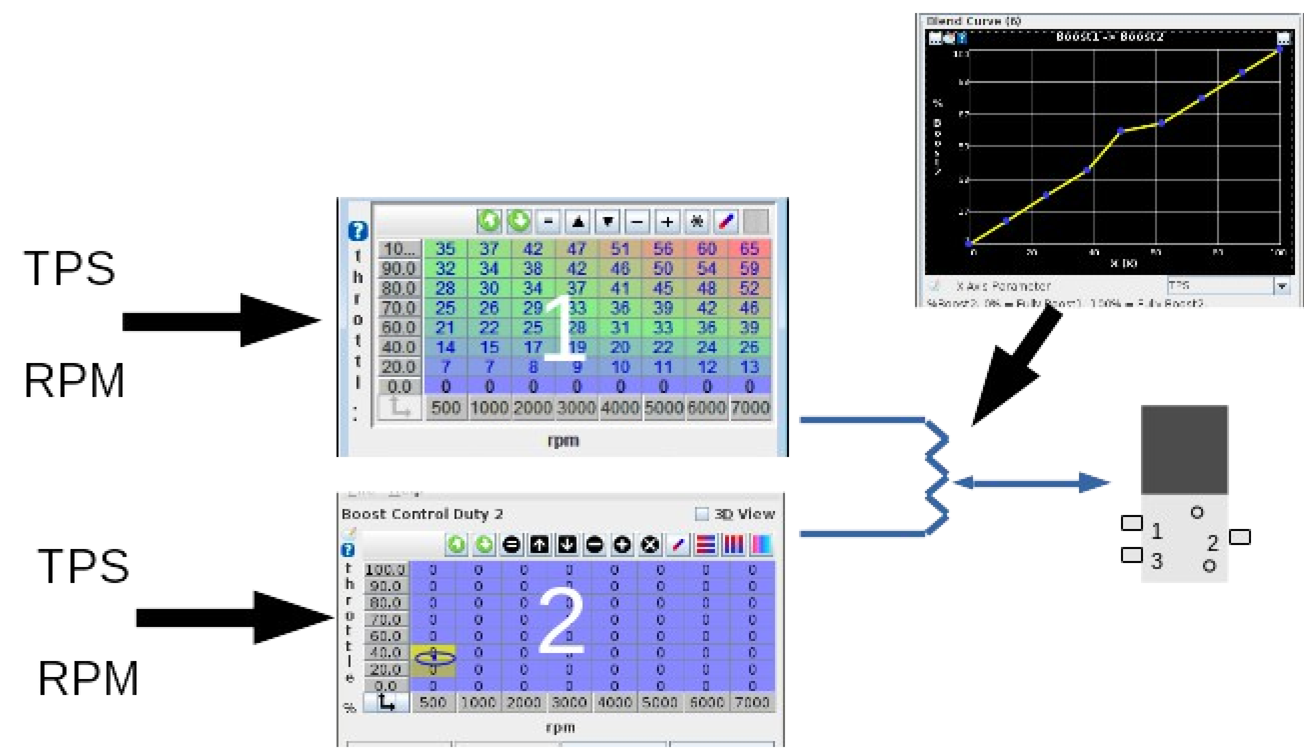
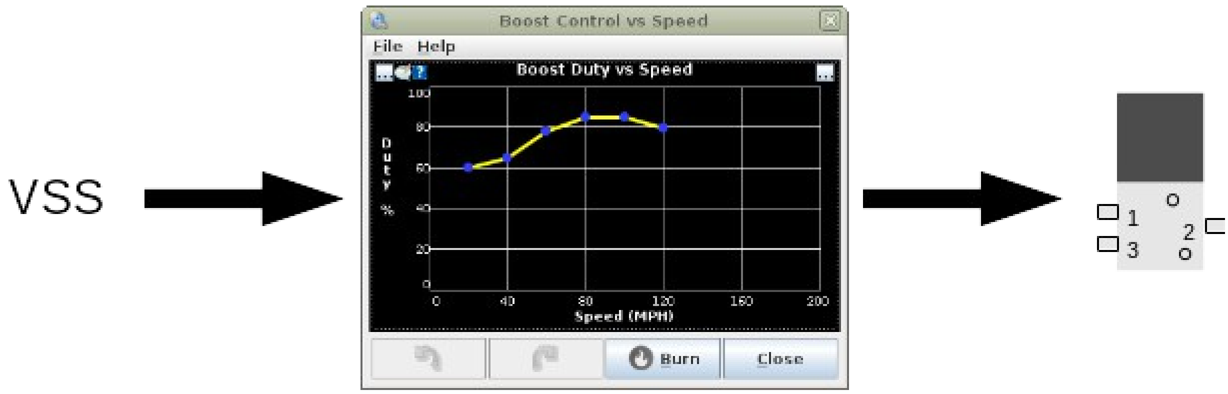


Table blending allows a blend between duty table 1 and duty table 2.

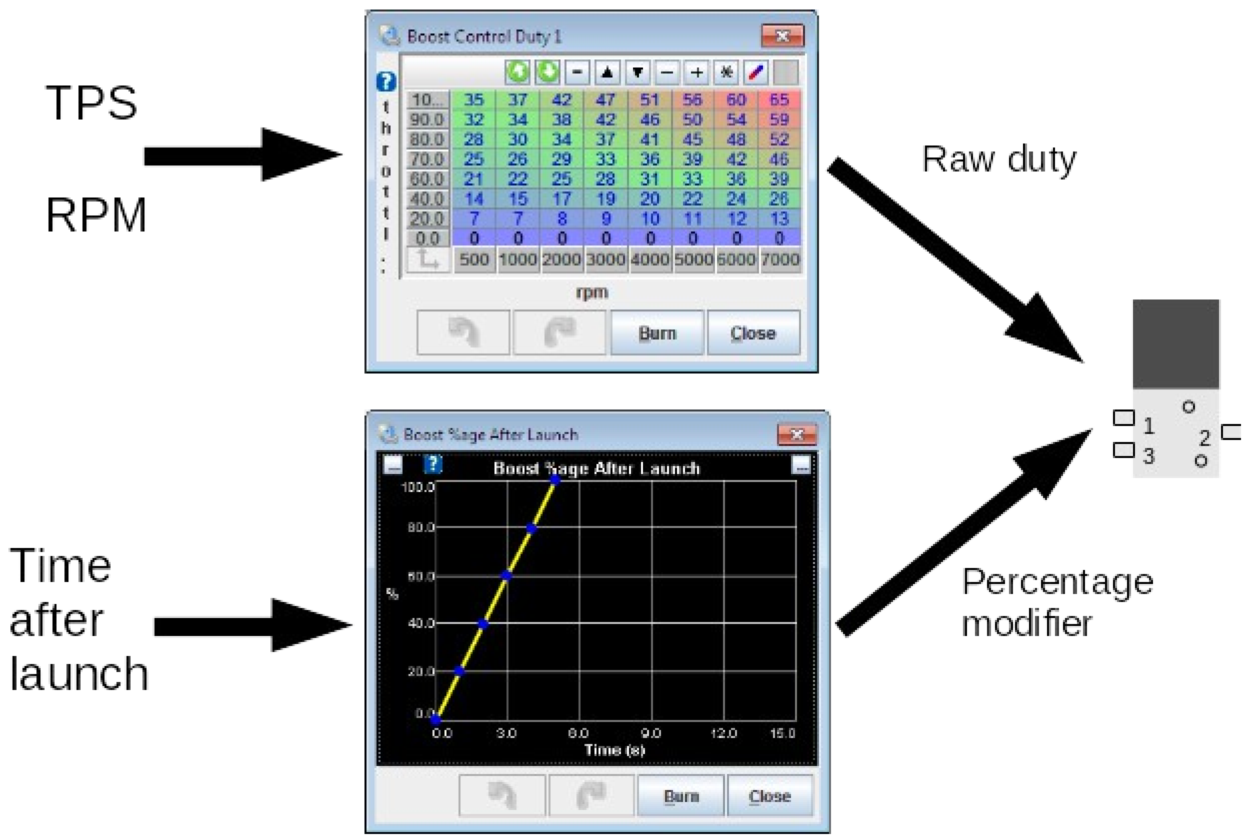


Specific launch duty allows a specified duty output when in launch mode.

Duty vs. speed replaces the duty table with a wheel speed (VSS) lookup curve when above a set TPS% threshold.



Timed boost acts as a modifier to the boost duty based on the time since launch.

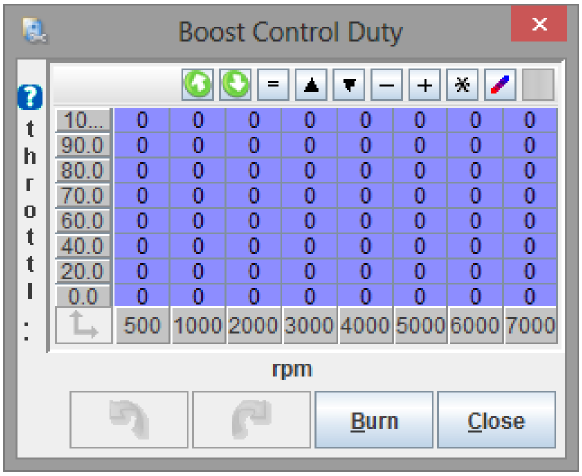


4.17.3 Open-loop Tuning Tips

With open-loop boost control, the wastegate solenoid output duty is directly set in a table looked up against TPS and RPM.

Ensure that more duty = more boost. 0% should give least boost (spring) and 100% should give maximum boost. If it does not, check your plumbing or flip the Output Polarity setting.

Try out different frequency settings and find the settings that work the best for the particular solenoid being used.

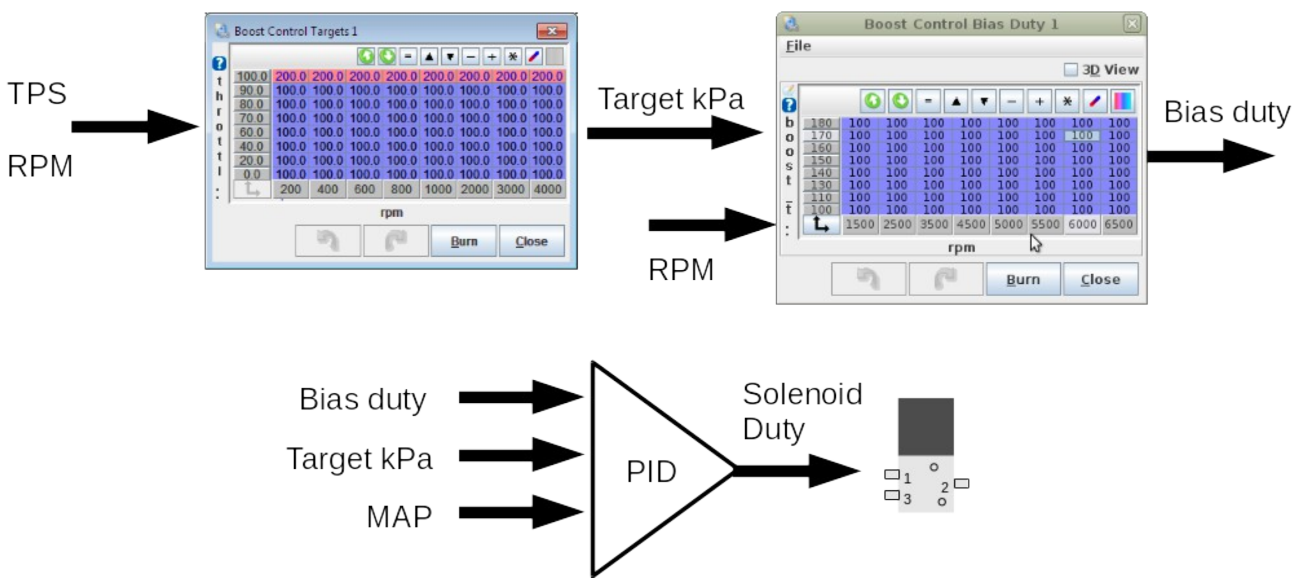


To tune the actual boost levels, just adjust the duty table so that boost reaches the desired level at each point in the table.

4.17.4 Closed-loop boost control concepts

With closed-loop boost control, the main lookup table is a target manifold pressure. A 'bias' table is used to specify a best estimate solenoid duty for each target kPa and RPM. Closed loop P.I.D. control is applied on top.

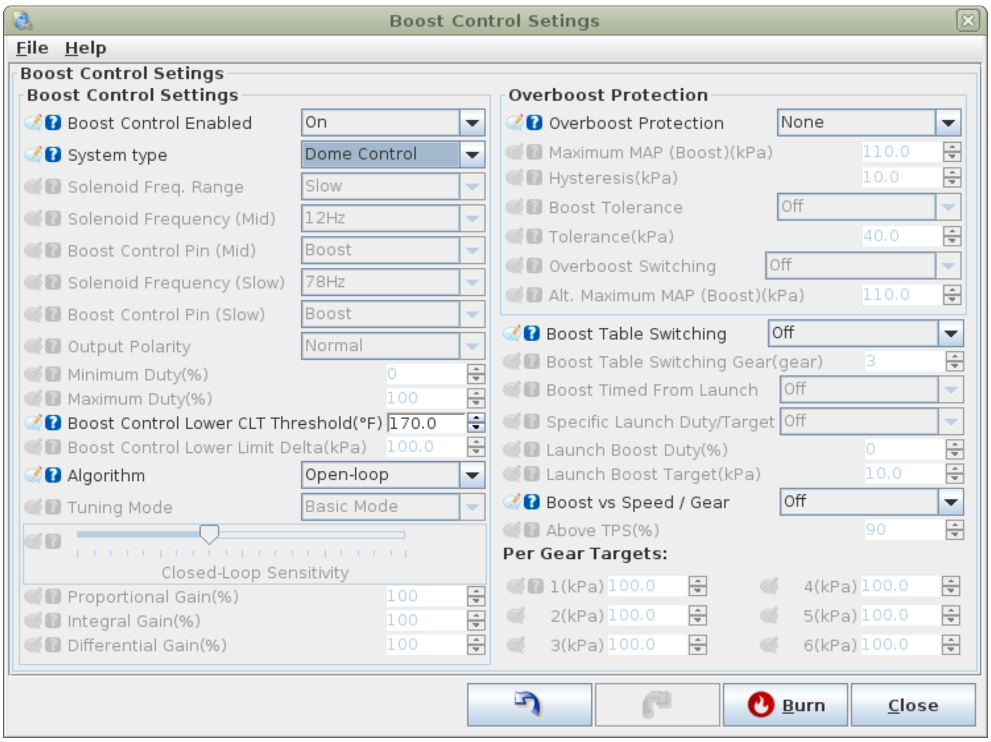
The bias table and P.I.D. parameters are adjusted during tuning.Closed loop concept



After initial tuning, the boost target (kPa) table can be adjusted without re-tuning the bias table. Speed and other modifers can change the boost target (kPa) - the bias table is used to lookup the best estimate duty for this boost level.

As with Open-loop boost, many other features allow the boost target kPa to be modified.

4.17.5 Settings



Specifics of the settings are covered in the TunerStudio Reference Manual.

4.17.6 Closed-loop Tuning Tips

This section reviews some tips for tuning closed-loop boost control. Tuning closed-loop boost control should be done in three steps:

- Tune open-loop boost control - In order to make sure that you have the proper polarity set for your valve, and to get a feel for the boost that certain duty cycles yield, it is recommended that open loop boost control is tuned first.
- Set Up Overboost Protection - The closed-loop boost control code uses the maximum boost set in Overboost Protection for some of its internal calculations. This was done purposely so that during the tuning process for boost control, if the settings are wildly incorrect, boost will still not climb any higher than the Overboost Protection allows it.
- Set the Boost Control Targets. These are the target (desired) boost level at various RPM and TPS points. Typically lower throttle positions will have lower boost targets.
- Tune Proportional-Integral-Derivative (PID) gains - The PID gains are the main controls for how quickly the boost will reach the target, and how close it will remain to the target through the RPM

range. Steps for tuning this can be found below.

Closed-loop settings

Typically, the defaults for the following settings can be used:

- *Control Interval* - The default for this setting is 20 ms. This is typically a good place to start. Lower settings can be used if overshoot cannot be tuned out when tuning the PID parameters.
- *Minimum Duty* - A closed duty of 0% is the default. This should be tuned to the value that starts to close the wastegate, but typically 0% works well.
- *Maximum Duty* - An open duty value of 100% is the default. This should be tuned to the value that fully closes the wastegate, but typically 100% works well.
- *Boost Control Lower Limit* Delta - Boost pressure must be within this many kPa of the target boost before closed loop control will activate. Outside of this range the valve is held wide open (keeping the wastegate shut) for fastest spooling.
- Tuning Mode - Basic internally sets P=100, I=100, D=100. Advanced allows full control
- Sensitivity Slider - This sets how sensitive the PID control is, start low and work up if response is too slow.
- PID controller gains - These control the actual response of the code to changes in RPM, as well as how well the code will reach the target. Tips for tuning these are listed below.

NOTE: The output polarity setting only changes the actual duty on the output. Regardless of this setting, the boost control algorithm is designed to operate on the assumption that more duty = more boost.

Tune the PID gains (advanced mode):

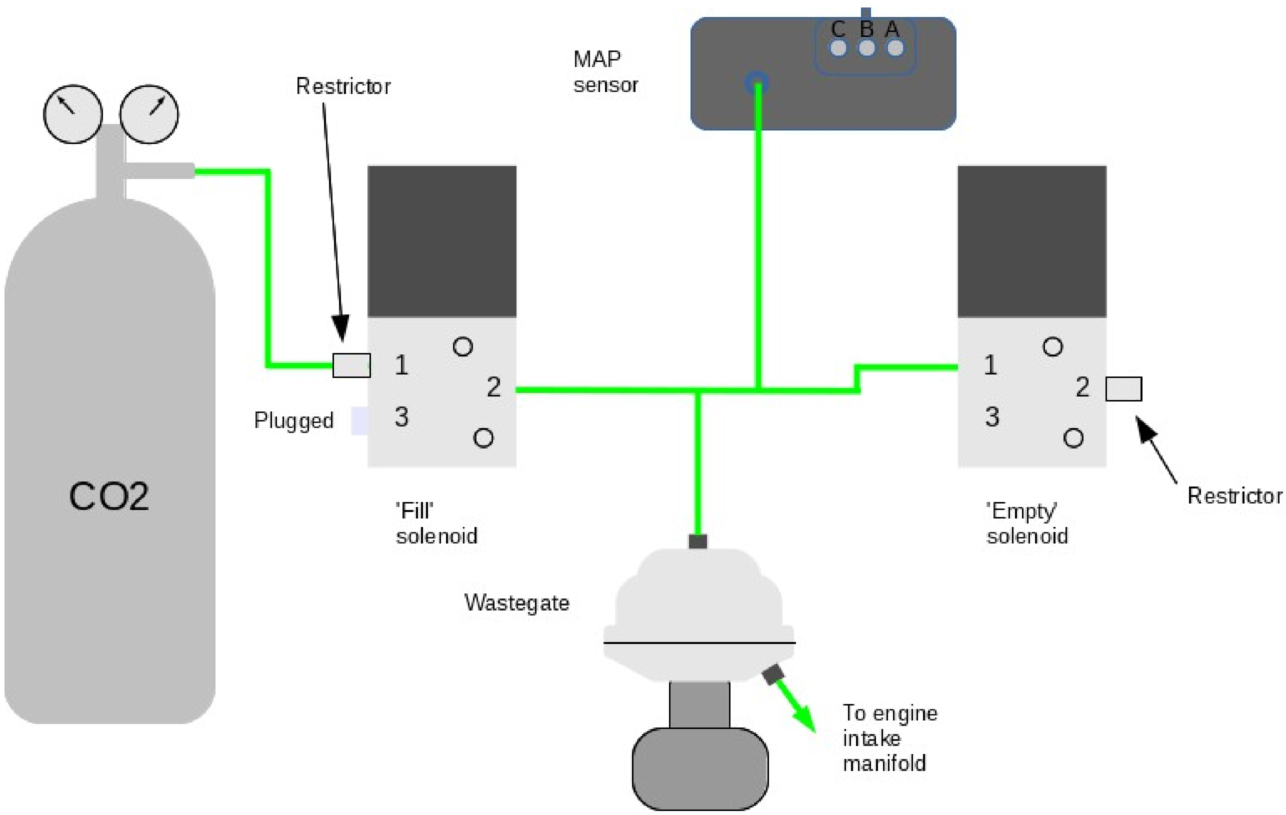
1. Set Integral and Differential Gains to 0% - To make tuning the Proportional gain easier, set the Integral and Differential gains to 0%.
2. Set Proportional gain to 100% and slowly lower - While tuning Proportional gain, higher numbers mean slower boost climb and lower final boost. For safety, start with a very high gain (100% should be sufficient). Find the RPM that typically spools quickly, and fully and quickly depress the accelerator. Note how much boost is reached. If boost overshoots the target dramatically, increase the Proportional gain. Otherwise, reduce the Proportional gain and try again. Do this until boost reaches the target with a small amount of overshoot.
3. Tune the Integral Gain - The next step after the target is reached consistently is to tune the Integral gain. Starting from the RPM used to tune the P-gain, fully depress the accelerator and watch the boost as the engine climbs through the RPM range. As the engine accelerates through the rev range, the boost will probably creep away from the target. Keep increasing the I gain until the controller adequately maintains the target with minimal oscillation. It may be necessary to increase the P gain a bit after tuning the I gain since the two gains tend to counteract each other.
4. Tune the Derivative Gain - Increase the D gain until the overshoot is minimized. Care must be taken when increasing the D gain as too much D gain can over-dampen the effects of the P and I gains.

4.17.7 Boost Dome Control

Particularly in drag-racing, it is common to use a compressed CO2 supply to forcibly pressurise the top portion of the wastegate actuator to achieve boost levels above spring pressure.

Two three-ported solenoids are required as well as an additional pressure sensor to monitor the dome pressure. The 'vent' port on the 'fill' solenoid must be plugged. Restrictors will likely be required on fill and empty solenoids to avoid overshooting.

The compressed gas supply is typically CO2 and needs to be regulated to a pressure a little above the maximum dome pressure required. For testing, compressed air works fine too.



Keep the hose between the MAP sensor and the wastegate as short as possible for better control.

4.17.8 Dome control concepts

Dome control is linked to the main boost control system in one of two fundamentally different ways:

a. Open-loop

> The target table is dome pressure targets. The manifold pressure (MAP) is ignored. The dome control system adjusts its solenoids to achieve the dome target.

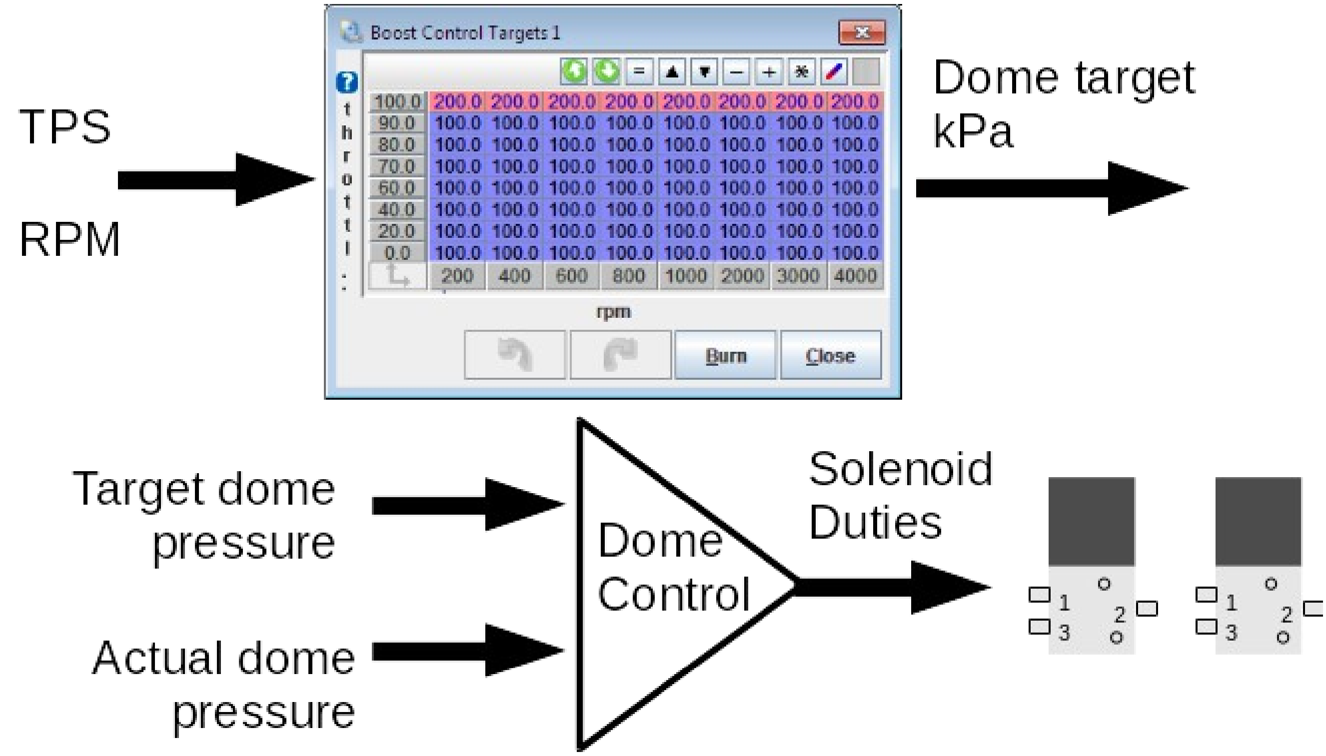
b. Closed-loop

> The target table is manifold pressure (MAP) targets. The closed-loop system adjusts the dome target to achieve the target MAP. The dome control system adjusts its solenoids to achieve the dome target.

a. Open loop

The "boost target" table is actually "dome target".

Boost target feeds directly into dome system. All existing modifiers (boost vs. gear etc.) work as before. This mechanism is used by other ECUs on the market. There is no reference made to engine MAP, so it is up to the tuner to determine what dome pressure they need.

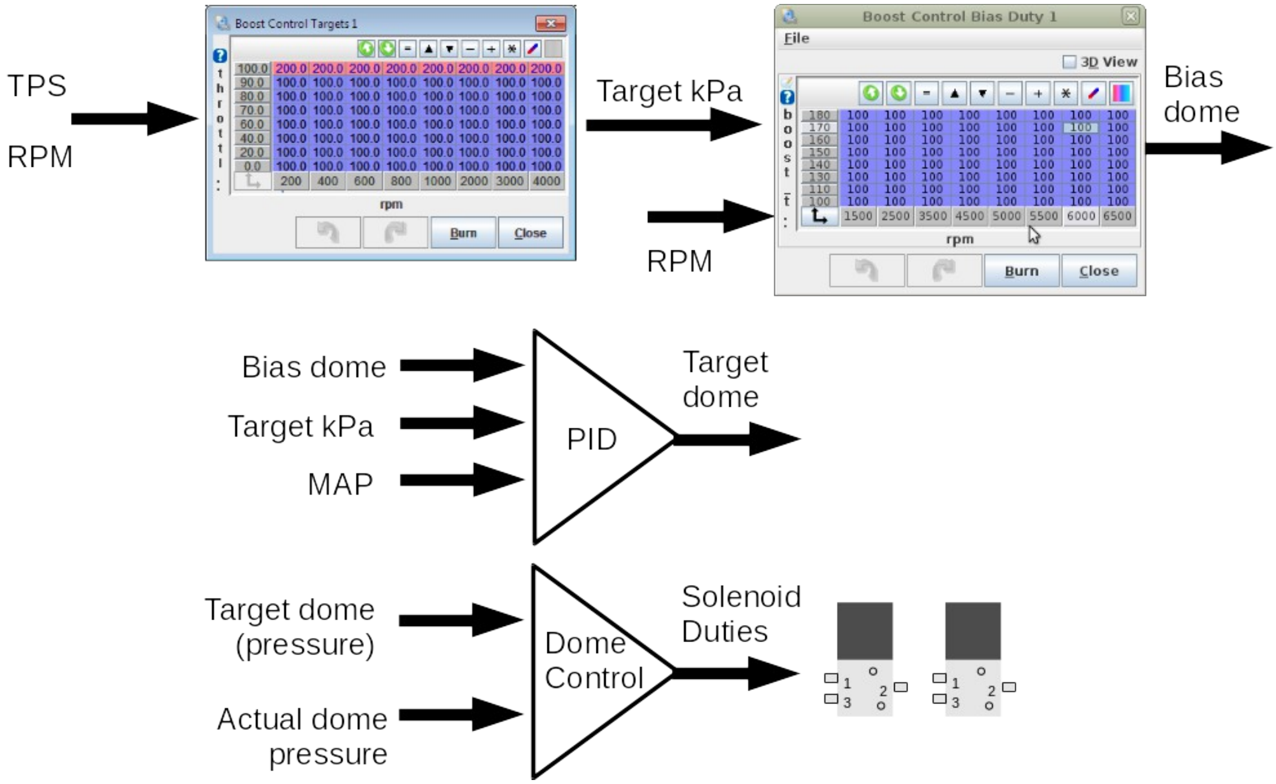


b. Closed loop.

The "boost target" table is the target manifold pressure (MAP).

The boost control system attempt to control the dome pressure to achieve the desired target boost.

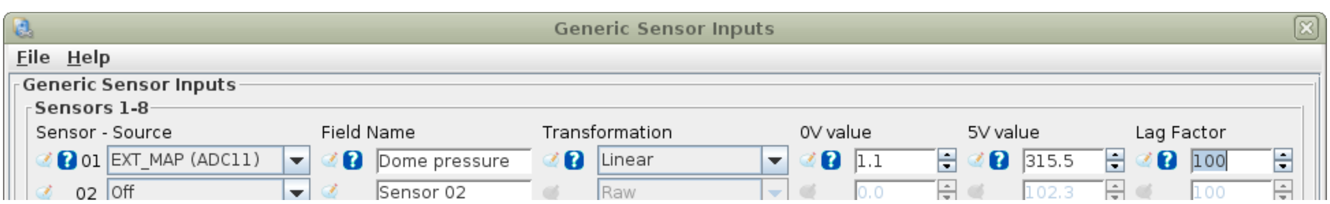
A bias table is used that gives a 'hint' to the code of the needed dome pressure for a given target boost.



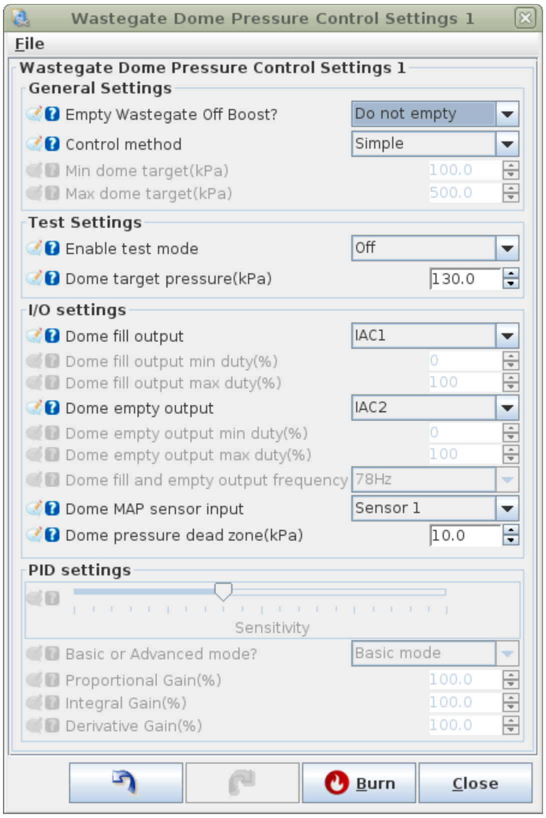
4.17.9 Dome MAP sensor configuration

The pressure sensor must be configured through the "Generic Sensors" system and works in absolute pressure. i.e. approximately 100kPa is atmospheric pressure.

The following screenshot is for a GM 3bar MAP sensor.



4.17.10 Wastegate Dome Pressure Control Settings 1



Specifics of the settings are covered in the TunerStudio Reference Manual.

4.18 Anti-lag (ALS)

Anti-lag is a combination of features you can use to keep the turbo spooled while shifting, beyond the basic flat shift rev limiter. The MS3 has several strategies it can use to dump extra fuel into the exhaust and increase exhaust gas temperatures under these conditions. This increase in exhaust temperature will shorten turbo life and possibly result in increased wear and tear on other parts of the exhaust system as well.

All anti-lag variations require an input switch to arm the system. When active, the anti-lag system will add fuel and retard timing to a number specified by the ALS tables. You can also allow the MS3 to use the following additional strategies when the anti-lag is active:

- Use a cyclic fuel cut
- Use a cyclic spark cut to dump unburned fuel into the exhaust
- Activate a solenoid or additional injector output to send fuel or air into the exhaust manifold directly
- Open the idle valve a set amount
- Open a solenoid to let additional air into the intake manifold

Note - use extreme caution if using any system that dumps fuel directly into the exhaust! This type of anti-lag can turn “danger to manifold” from an ironic “The Fast and the Furious” reference into a very real hazard.

As many anti-lag systems also have the throttle or idle control valve open by a large amount, the MS3 can also use a rotational idle fuel cut to reduce idle speed on engines using anti-lag. The rotational idle engages when anti-lag is off.

Anti-lag is only used when you release the throttle to shift; see the Launch Control section for how to set up the MS3 for no lift shifting.

Since anti-lag creates so much heat, the MS3 has several settings to limit the anti-lag system’s use to limit heat buildup.

4.19 Variable Valve Timing (VVT, VCT, VTEC, VANOS)

Many engines now feature variable valve timing. This allows a balance between engine performance and economy, with the ECU automatically adjusting the valve timing to suit operating conditions. The particular implementation varies with engine and manufacturer. The first systems available are of the "on/off" variety where the cam is moved by a fixed angle only, and there is no feedback system.

Examples of this include:

- Honda first generation VTEC
- Toyota first generation VVT
- BMW single VANOS

More advanced systems use continuously variable cam controls - typically an oil solenoid valve controlled by a PWM signal from the ECU. The ECU uses one camshaft position sensor per cam to compare the cam advance to a crankshaft position sensor reading. Examples of continuously variable systems include:

- Toyota VVTi
- BMW double VANOS
- Ford Zetec and Ti-VCT
- Ford Coyote

Most continuously variable VVT systems use a single solenoid where a duty near 50% holds the cam in position. More than 50% causes it to advance or retard. Less than 50% moves it in the opposite direction. The most complex system currently supported is the BMW V10 Vanos system with four cams being controlled individually. The first cam sensor is used for engine phase information and cam position. The other three cam sensors are used to monitor cam position only.

Note that outside of the V10 applications, BMW’s Double VANOS is a bit of an oddity - it uses separate solenoids to advance and retard the cam, much like a 3 wire PWM idle control valve. Support for this system is under development in the 1.4.x firmware.

If you are controlling more than one cam, you will need to use additional input circuits for the cam input. See your hardware manual for wiring or customization.

See the TunerStudio Reference for details of each setting.

4.19.1 On/off mode setup

For on/off mode, no feedback of cam position is used and most settings on the VVT settings screen are unused and grayed out. The output value, frequency and injector timing adjustment are set through the main settings screen.
In place of the angle target table, an on/off table is used. In this table, set 100 for cells where VVT should be active and 0 for cell where it should be inactive. Do not use other values.

4.19.2 Variable mode initial setup

The most basic setup is to make the wiring connections for inputs and outputs. Most engines will run safely with VVT inactive; ensure this is the case on your engine. Having completed the wiring, set the basic configuration reflecting the wiring choices you made, the control type and the number of teeth on the cam wheels. Leave the min/max cam angles as zero for now.

On your TunerStudio dashboard, enable the gauges VVT_angle (1,2,3,4 as required). Start the engine and observe the VVT angle gauges. Use the test mode for each VVT with 0 and 100% duty and observe the swing of the cam angles. You will hear the engine tone change as the cam angle changes. It is important that the vvt angles you observe are stable and vary evenly. With continuously variable cam timing, the vvt angle will normally swing from fully retarded to fully advanced (and vice versa) quite quickly with any duty beyond the hold/neutral duty, so it will be unlikely that you will be able to catch the cam at anything other than the limits of its travel. Double check that the cam min/max angles are presently set to zero.

For each cam, record the minimum and maximum angles you observe while varying the test duty.

Enter these into the min/max fields. Note that the maximum MUST be larger than the minimum. (In the instance where the angles cross 720 degrees, add 720 to the maximum angle. e.g. if minimum was 700 and maximum was 40 degrees, enter 700 and 760.) At this stage you should also be able to determine if more duty is retarding or advancing each cam.

Internally the Megasquirt code reads from the multiple cam teeth and reports a single angle. On a few engines, the angle may 'jump' during test mode (e.g. from 178,179, to 0, 1.) Find the minimum angle in the upper range of angles (e.g. 160) and enter that as the minimum angle for that channel, then you should find that the angles sweep up as e.g. 178,179,180,181 etc.

Having configured the inputs and outputs, proceed with PID tuning. Take datalogs and observe the vvt actual angle compared to the target angle. We recommend tuning in much the same way as boost control. Start with the I and D terms zeroed out, dial in the P term first, followed by the I. If you have problems with the cam overshooting its targets, gradually increase the D term until it damps out the overshoots.

Once the cam control is working, you may dial in the VVT activation tables. These are usually best to dial in on a dyno.

Most VVT set-ups have the intake cam fully retarded when "at rest" and the exhaust cam fully advanced "at rest". In this case it is recommended to set the exhaust cam to use retard values in the table. i.e. 0 means to leave the cam in the rest position and 5 would mean to retard the cam by 5 degrees.

4.20 Table-switching / Blending / Staging

For many installs a single fuel table is all that is required! New users are strongly advised to start with a single table!

MS3 offers a number of methods of switching and blending between tables. These methods can be used to effectively extend the basic table size, for better tunability with different fuel algorithms (e.g. low vs high rpms), for dual fuels, or for different fuel mixes. Presently, fuel, ignition, boost and AFR targets may be blended or switched. Due to the complexity introduced with multiple tables, only experienced users should consider enabling these features. Additionally, the algorithm blending should only be enabled once each table is well tuned. Autotuning (VE Analyze Live) will not function on some combinations of blended tables.

Terms used:

• Table switching - the ability to swap from one table to another, based on RPM, TPS, MAP or a physical switch input

• Combined tables - two tables are combined to give a single VE (or spark) result

• Algorithm blending - two tables with different calculation algorithms are used to calculate fuel (or spark) and then the result is blended together

• Dual table - two tables operate independently to control injectors or injector banks

• Staged injection - one fuel calculation control two banks of injectors. e.g. small injectors for idling and larger injector for full power. (See Staged Injection page.)

Algorithms are different strategies for determining the 'load' on an engine and then calculating fuel and spark etc. As a brief recap:

• Speed-Density - Uses MAP as load.

• Alpha-N - Uses TPS as load.

• MAF - Measures actual airflow. Spark load uses a calculated percentage of cylinder filling.

• %baro - Uses MAP divided by barometric pressure as load.

• ITB - Uses a mix of MAP and TPS as load.

4.20.1 Example scenarios

Why use table switching or blending at all? Actually, most users won't need to, and can get by just fine with a single table. Here are some examples of engines that could genuinely need table switching or blending.

• A big block Chevy has two different types of port designs, one half of which flow considerably better than the other half. Such an engine may need true dual table mode (although you can also use the individual cylinder trim to achieve the same effect.)

• A large cammed motor with a poor MAP signal below 2,000 RPM that stabilizes at higher RPM could use blended alpha N and SD, with the motor running on alpha-N at low RPM and speed density at high RPM.

• A turbo motor with a very large MAF for good fuel metering that runs out of MAF resolution at low flow could use blended speed density and MAF fueling, with speed density at low RPM and the MAF at high RPM.

• A motor with a very wide powerband could use two separate speed density tables, switching them based on RPM. This can effectively be used to create a 30 x 16 fuel table.

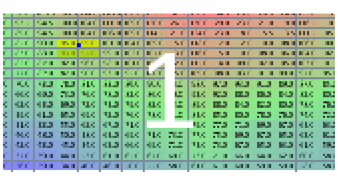
• A motor with switchable intake runners could use separate tables for when the long runners are open vs when the short runners are open.

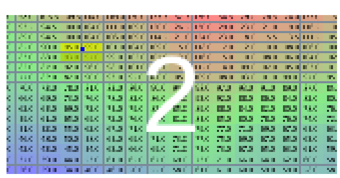
• A race car that runs in classes with different fuel requirements could have separate maps for pump gas and race gas.

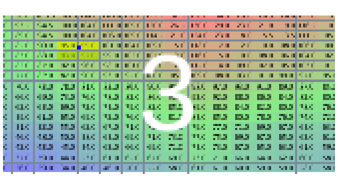
• A vehicle with a flex fuel sensor can blend in between separate gasoline and ethanol maps, interpolating between the two for the percent of fuel in the tank.

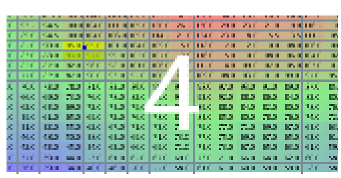
4.20.2 Fuel blending and switching types

There are four fuel VE tables, VE1, VE2, VE3, VE4



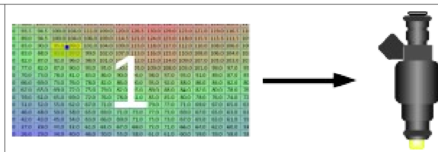
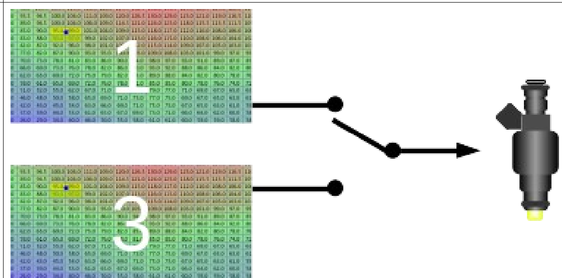
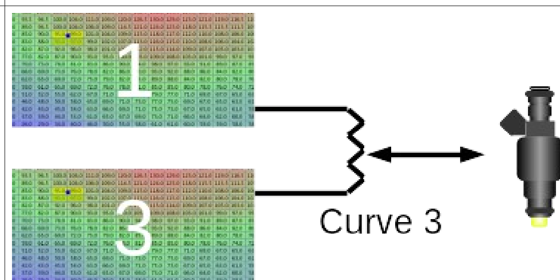
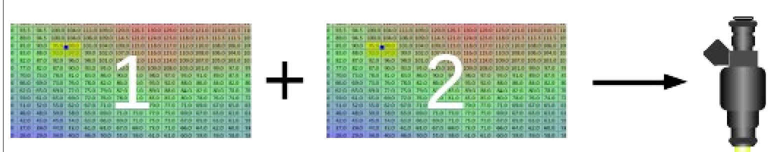
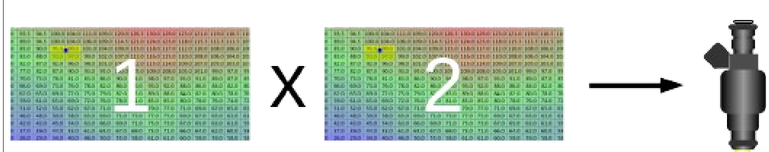




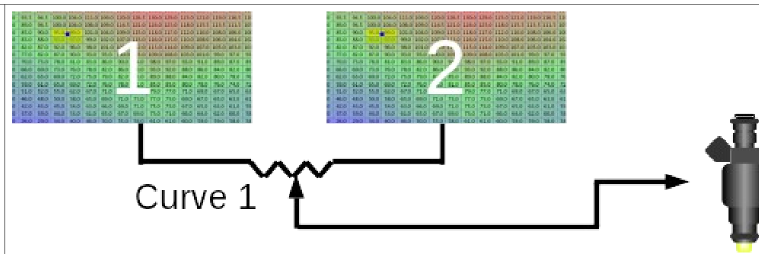


Here are some of the possible combinations of fuel table blending and switching. Electrical symbols for switches and variable resistors illustrate the switch or blend functions.

Method	Diagram

<table>
<tr><td>Single table. One fuel algorithm uses VE1.
This is the most common setup.</td><td>
</td></tr>
<tr><td>Single table with table-switching. One fuel algorithm uses VE1 or VE3 depending on tableswitch setting/input. Could be used with dual-fuel or also as a method to extend effective table size.</td><td>
</td></tr>
<tr><td>Single table with table-blending. One fuel algorithm uses a proportion of VE1 or VE3 depending on the blend input and curve (3).
The tables could be blended based on ethanol percentage in the fuel from a flex-fuel sensor.</td><td>
</td></tr>
<tr><td>Combined tables - secondary additive table. Fuel is calculated by adding the lookup from VE1 and VE2.
VE2 can be used a tweak to VE1 to add a 4th dimension to the fuel table e.g. including a throttle component in a speed-density tune. 0% in the secondary table means no change, 2% means add 2%</td><td>
</td></tr>
<tr><td>Combined tables - secondary multiplicative table. Fuel is calculated by multiplying the lookup from VE1 and VE2.
VE2 can be used a tweak to VE1 to add a 4th dimension to the fuel table e.g. including a throttle component in a speed-density tune. 100% in the secondary table means no change, 102% means add 2% etc.</td><td>
</td></tr>
</table>

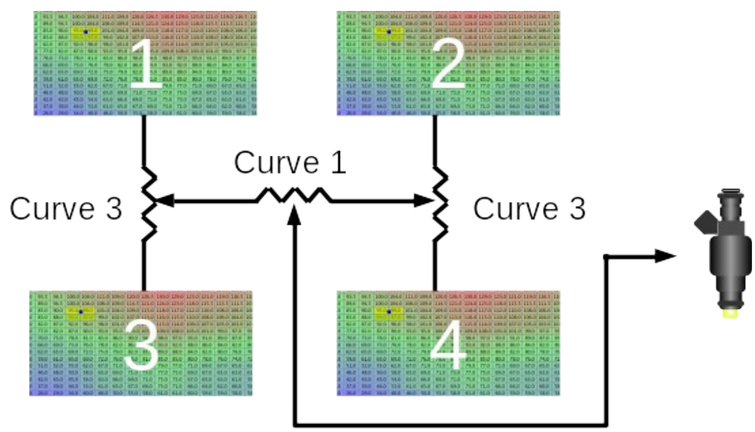
Secondary algorithm blended table.
Fuel is calculated from VE1 using the primary fuel algorithm and also from VE2 using the secondary fuel algorithm and then the result is blended.
This could be used to have a blend between a Speed-Density in VE1 and MAF tune in VE2. The blend curve (1) allows you to alter where each algorithm is in use. Each table should be tuned individually i.e. set the curve to 0% and tune VE1, then set to 100% and tune VE2, afterwards enable blending.



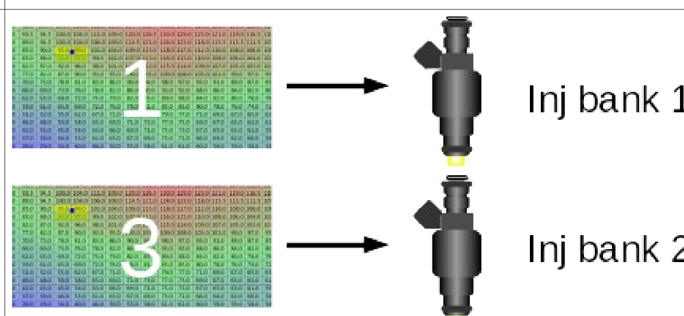
Two way blend. Fuel is calculated from VE1 and VE3 using the primary fuel algorithm. These are blended. Then VE2 and VE4 are calculated using the secondary fuel algorithm and blended. The VE1/3 result is then blended with the VE2/4 result.
This could be considered for E85 (VE3/4) vs regular gasoline (VE1/2) where algorithm blending is also used between left and right, such as a blend of Speed Density and MAF.

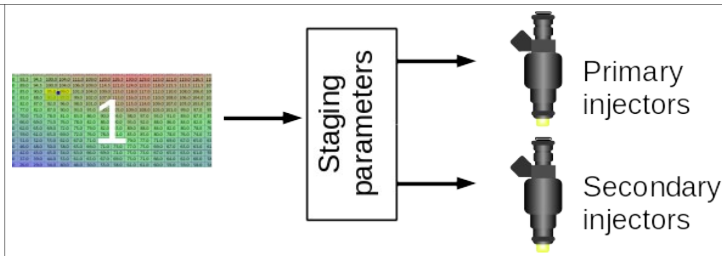
This is incompatible with 'Dual Table' as that uses VE3/4 for the second bank.

Two way blending is complex ! Do not enable it without fully tuning each individual table.



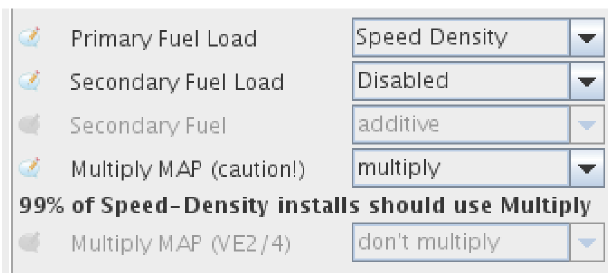
Dual table. Fuel for injector bank 1 is calculated from VE1 and fuel for injector bank 2 is calculated from VE3.
This could be used where an engine has radically different fuel requirements between banks (or cylinders in a V twin.) or as part of a dual-fuel install where the second bank is used for LPG or methanol etc.



Injector staging. Fuel is calculated from VE1 and the staging parameters determine the split between primary and secondary injectors. This is typically used on engines with a large dynamic range so that small injectors may be used at low loads for good tunability with additional (staged) large injectors coming online at full power. This concept is used from the factory on Mazda rotary engines.	
More variations. There are many more combinations possible to include double blending, switching between blended tables etc.	

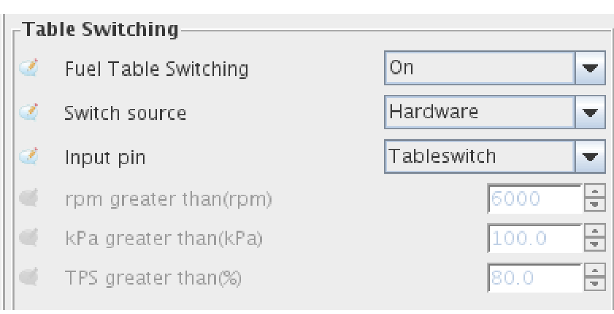
4.20.3 Fuel blending and switching settings

4.20.3.1 Single table



By far the most common install is to use a single Speed-Density fuel table. This really should be your starting point before considering other load methods and certainly before multiple-table configurations are enabled.
Multiply MAP controls whether MAP is included in the calculation for VE1 (and VE3). For regular Speed-Density this should almost always be enabled. For 'Pure' Alpha-N, leave it disabled. For 'Hybrid' Alpha-N, enable it. For MAF it is ignored.

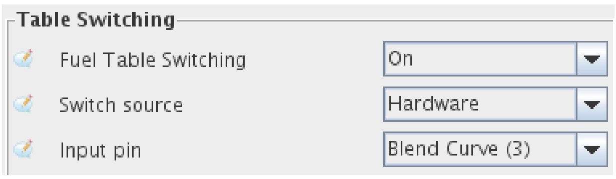
4.20.3.2 Single table with table-switching

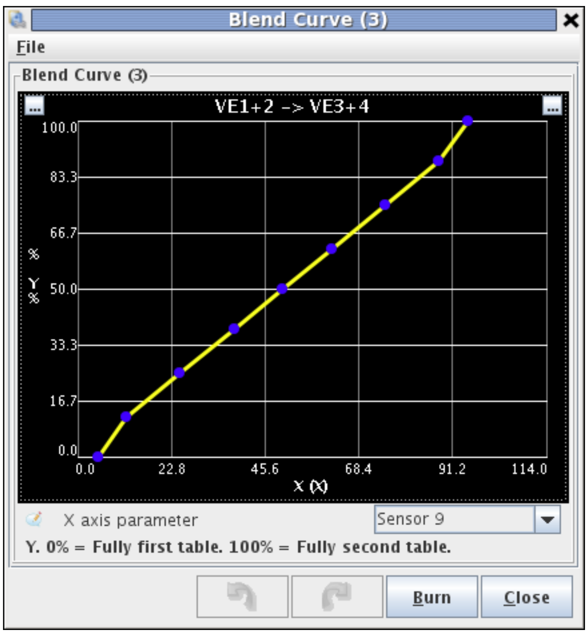


A physical switch input (shown here connected to 'Tableswitch') is used to swap between VE1 and VE3 for fuelling. The same algorithm is used for both tables.

The RPM, TPS, kPa switch points can be used as a form of table extension where say VE1 is 0-3500rpm and VE3 is 3500-7000rpm.

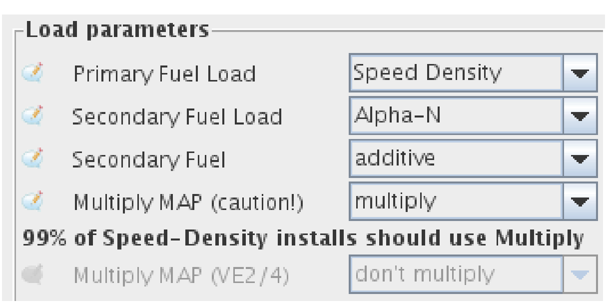
4.20.3.3 Single table with table-blending





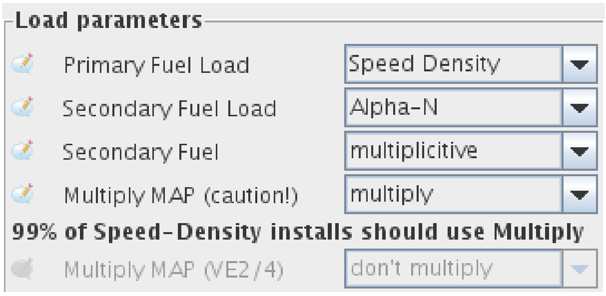
Blend curve (3) is used to define the blend between VE1 and VE3 which use the same algorithm. At 0% of 'sensor9' (could be a trim pot) the fuel is calculated from VE1. At 100% of 'sensor9' the fuel is calculated from VE3. In-between, proportions of table will be used. It is very important to tune each table individually - trying to retune with a partial blend is likely to result in confusion or failure.

4.20.3.4 Combined tables - secondary additive table



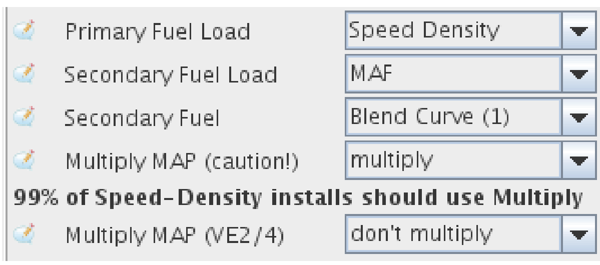
VE1 is combined with VE2 using addition (the VE numbers are added together.) This can allow a secondary algorithm to act as a modifier or with the same primary and secondary algorithm selected, it may be used as a means to extend table size. e.g. VE1 = 0-100kPa and VE2 = 100-400kPa

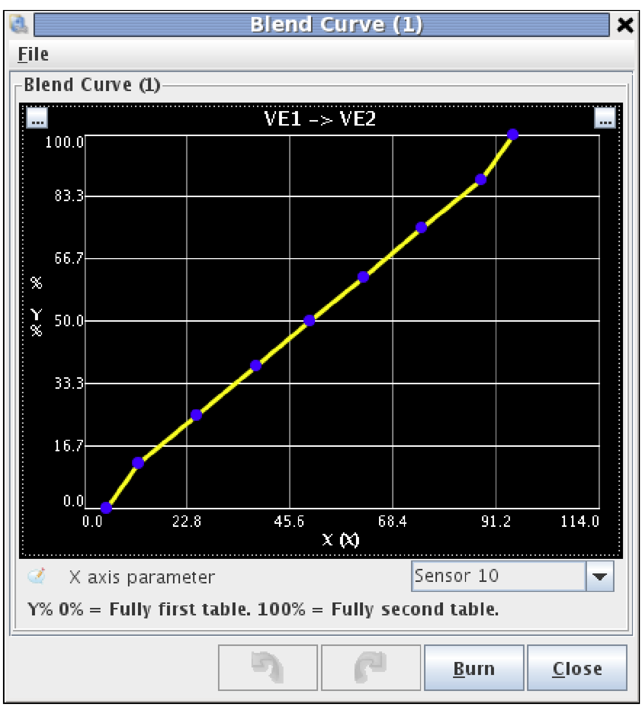
4.20.3.5 Combined tables - secondary multiplicative table



Very similar to the previous option but the VE numbers are multiplied.

4.20.3.6 Secondary algorithm blended table



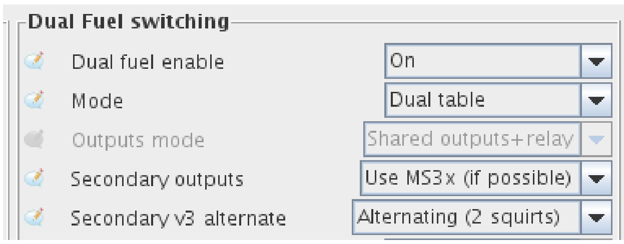


Here VE1 is setup as a Speed-Density tune with multiply MAP turned ON. VE2 is the trim table for MAF.

Blend curve (1) is used to define the blend between the two algorithms. At 0% of 'sensor10' (could be a trim pot) the fuel is calculated 100% using the Speed-Density Tune. At 100% of 'sensor10' the fuel is 100% MAF based. In-between, proportions of the fuel pulsewidth calculated from each algorithm will be used. It is very important to tune each table individually - trying to retune with a partial blend is likely to result in confusion or failure.

Multiply MAP (VE2/4) controls whether MAP is included in the calculation for VE2 and VE4. For regular Speed-Density this should almost always be enabled. For 'Pure' Alpha-N, leave it disabled. For 'Hybrid' Alpha-N, enable it. For MAF it is ignored.

4.20.3.7 Dual table



Fuel for injector bank 1 is calculated from VE1 and fuel for injector bank 2 is calculated from VE3. This can also be used in conjunction with VE1/2 (and VE3/4) combined tables and algorithm blending. It is not compatible with table switching or blending from VE1 to VE3.

4.20.3.8 Staged Injection

Staged injection allows the use of one set of small injectors for low-load and/or low RPM operation of the engine with the ability to engage a second set of injectors at higher load and RPM when the primary set of injectors would otherwise reach their maximum operational duty cycle. The staged injection function supports staging equally between the primary injectors and secondary injectors during staged operation as well as staging completely to the secondary set of injectors. The channel availability varies depending on the hardware you have. Staging can operate in sequential or batch modes.

If you have enough outputs to use timed injection on the secondary injection, the Secondary Injection Timing Table is enabled (in addition to the standard injection timing table), so the user must remember to tune secondary injection timing separately from primary injection timing.

Tuning Staged Injection

It is recommended that on any setup with secondaries placed further up the intake tract than the primaries, table-based staging is used. It is possible to achieve a much smoother transition to staged injection in all situations when tuning with this method.

Tuning Table-based Staged Injection

The following tips should be followed when tuning table-based staged injection:

- RPM and Load transition bins - Make the two RPM bins and two Load bins where staged injection first engages close together. Also make the staging percent jump to 10-20% almost immediately (as shown in the dialog at the beginning of the staged injection section). This is so that a very small amount of time is spent with the secondary injectors at or near the injector opening time for those injectors. Spending a lot of time near the injector opening time can lead to inconsistent fueling, especially if the secondary injector opening time has not been determined and the default value is being used.
- Transition to 100% engaged - The transition to 100% engaged should be determined using experimentation. In general, the transition should be set so that the primaries stay close to their maximum duty cycle (80% is recommended) for as long as possible. This ensures that reduction to the primary pulse-width does not result in a lean situation. The table displayed at the beginning of the staged injection settings section is a good example of how to tune table-based staging for a smooth transition on a naturally aspirated engine.

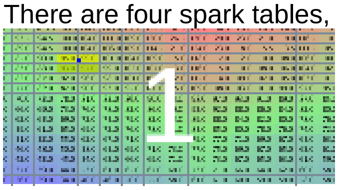
Tuning All other Staged Injection Modes

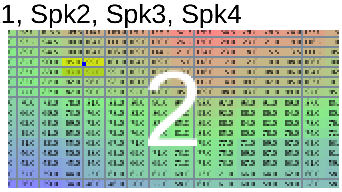
All staged injection modes that do not use the table to determine the staging amount can be tuned similarly. The following tips should be used:

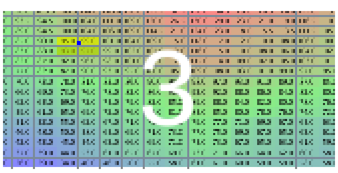
- Primary staging parameter - It is usually recommended that the primary staging parameter used is Duty. This ensures that staged injection engages when the duty cycle of the primary injectors warrants it instead of trying to guess what RPM or load will cause use of the secondaries to be necessary.
- Secondary staging parameter - It is recommended that this parameter only be used with forced induction engines. It should be used to ensure that staging is fully complete before going into boost so that any lean spots caused by staging are gone.
- Gradual transition - The gradual transition code was introduced to try to solve the same problems that table-based staging solves. Notably the small lean spot in AFR briefly after staged injection engages. It should be set to transition over as many ignition events as possible for the smoothest transition. If doing this still causes a lean spot, the primary reduction delay can be used along with the secondary enrichment setting to make sure that slightly more fuel than calculated using the normal fuel calculations is injected. If enabling the gradual transition feature still does not get rid of the brief lean spot after staging is engaged, it is recommended that table-based staging is used.

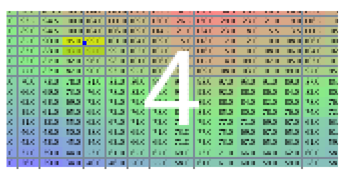
4.20.4 Spark/ignition blending and switching types

There are four spark tables, Spk1, Spk2, Spk3, Spk4

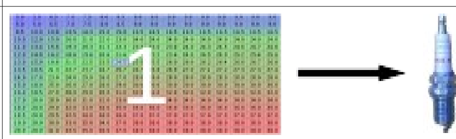
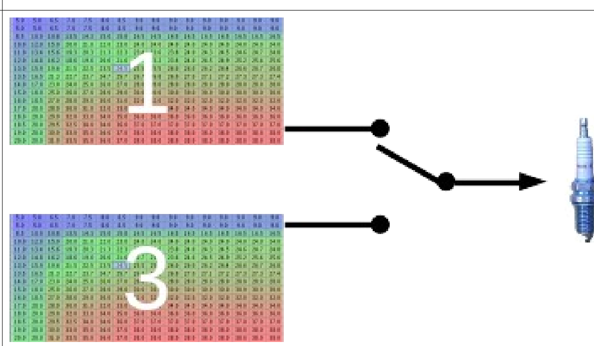


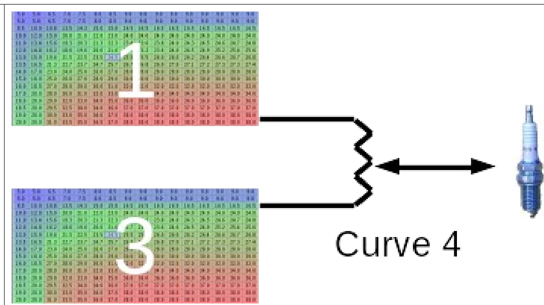
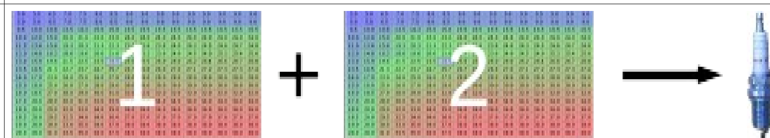
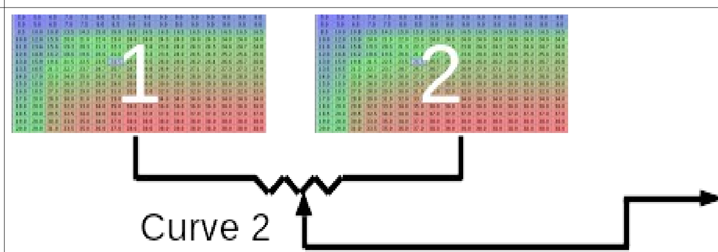
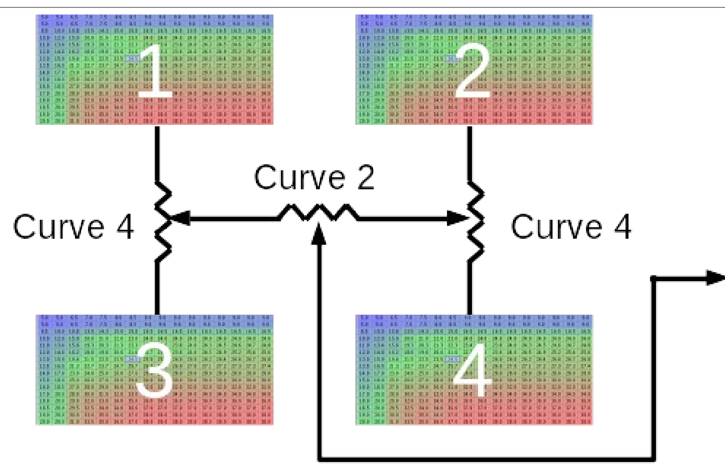






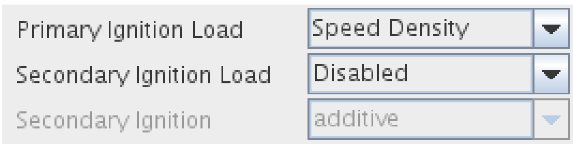
The switching and blending options work very similarly to fuel.

Method	**Diagram**
Single table. One ignition algorithm uses Spk1. This is the most common setup.	
Single table with table-switching. One ignition algorithm uses Spk1 or Spk3 depending on tableswitch setting/input. Could be used with dual-fuel or also as a method to extend effective table size.	

Single table with table-blending. One ignition algorithm uses a proportion of Spk1 or Spk3 depending on the blend input and curve (2). The tables could be blended based on ethanol percentage in the fuel from a flex-fuel sensor.	
Combined tables - secondary additive table. Ignition is calculated by adding the lookup from Spk1 and Spk2. Spk2 can be used a tweak to Spk1 to add a 4th dimension to the fuel table e.g. including a throttle component in a speed-density tune.	
Secondary algorithm blended table. Ignition is calculated from Spk1 using the primary ignition algorithm and also from Spk2 using the secondary ignition algorithm and then the result is blended. This could be used to have a blend between a Speed-Density in Spk1 and MAF tune in Spk3. The blend curve (2) allows you to alter where each algorithm is in use. Each table should be tuned individually i.e. set the curve to 0% and tune Spk1, then set to 100% and tune Spk2, afterwards enable blending.	
Two way blend. Ignition is calculated from Spk1 and Spk3 using the primary ignition algorithm. These are blended. Then Spk2 and Spk4 are calculated using the secondary ignition algorithm and blended. The Spk1/3 result is then blended with the Spk2/4 result. This could be considered for high-octane (Spk3/4) vs low octane (Spk1/2) where algorithm blending is also used between left and right.	

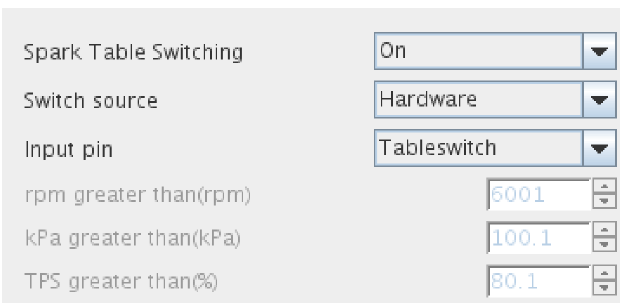
4.20.5 Spark/ignition blending and switching settings

4.20.5.1 Single table



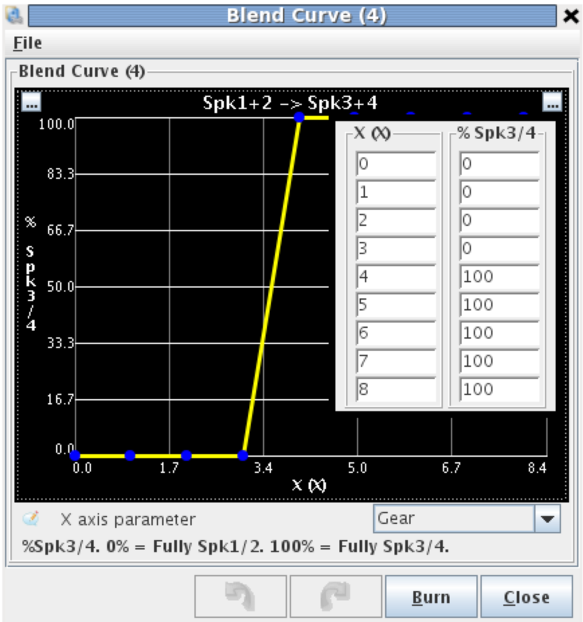
By far the most common install is to use a single Speed-Density ignition table. This really should be your starting point before considering other load methods and certainly before multiple-table configurations are enabled.

4.20.5.2 Single table with table-switching



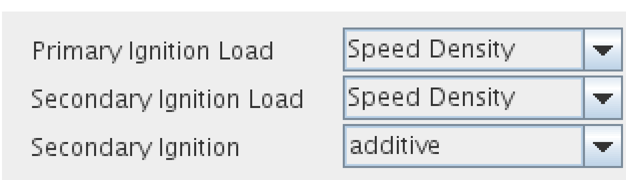
A physical switch input (shown here connected to 'Tableswitch') is used to swap between Spk1 and Spk3 for ignition. The same algorithm is used for both tables.
The RPM, TPS, kPa switch points can be used as a form of table extension where say Spk1 is 0-3500rpm and Spk3 is 3500-7000rpm.

4.20.5.3 Single table with table-blending



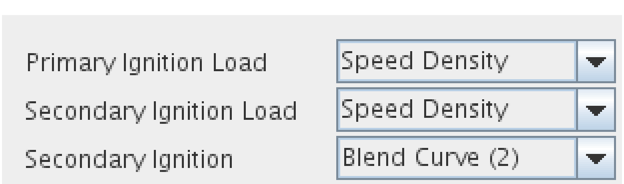
Blend curve (4) is used to define the blend between Spk1 and Spk3 which use the same algorithm. Here it is used for gear based spark table switching. Spk1 is used for gears 1,2,3. Spk3 is used for gears 4,5.
It is very important to tune each table individually - trying to retune with a partial blend is likely to result in confusion or failure.

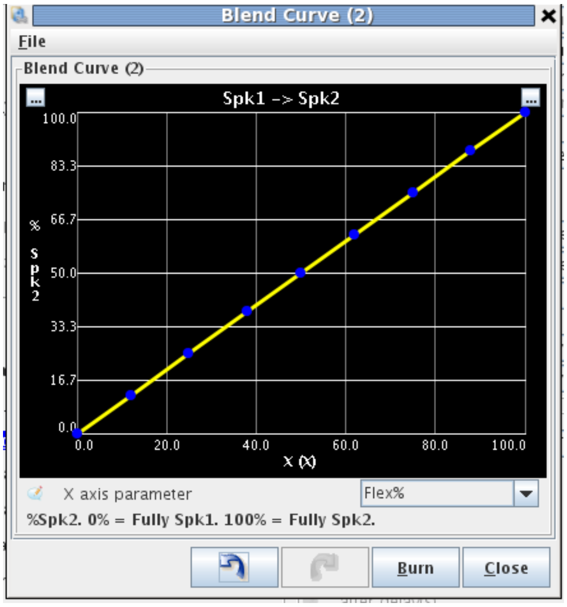
4.20.5.4 Combined tables - secondary additive table



Spk1 is combined with Spk2 using addition (the advance numbers are added together.) This can allow a secondary algorithm to act as a modifier or as a means to extend table size e.g. Spk1 = 0-100kPa and Spk2 = 100-400kPa with the same primary and secondary algorithm selected.

4.20.5.5 Secondary algorithm blended table

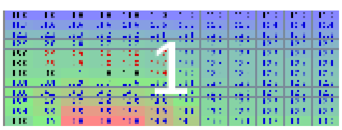


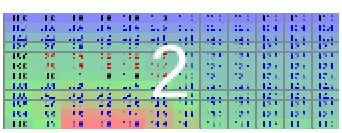


Blend curve (2) is used to define the blend between Spk1 and Spk2 which may use different algorithms. Here it is used to blend between the spark tables based on flex fuel %age.
It is very important to tune each table individually - trying to retune with a partial blend is likely to result in confusion or failure.

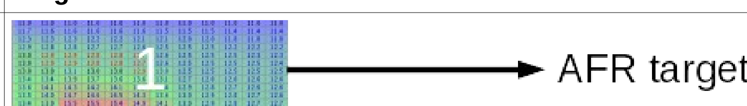
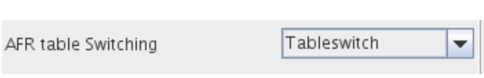
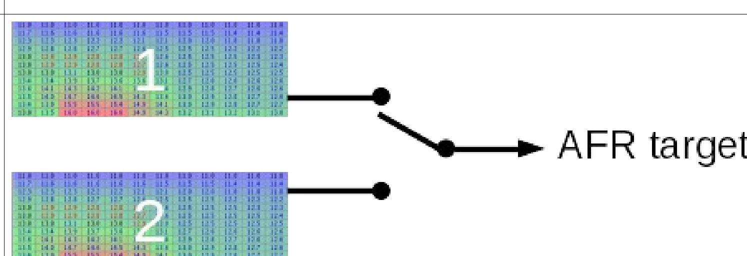
4.20.6 AFR blending and switching types

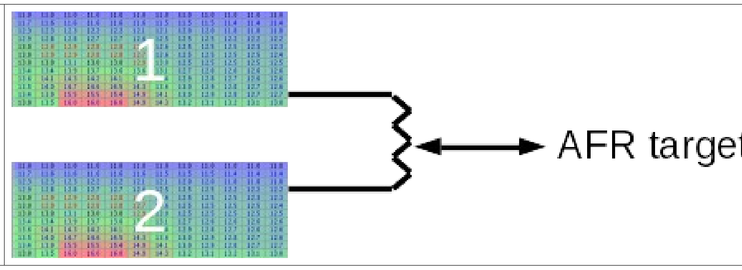
There are two AFR tables, AFR1, AFR2





There are a number of options to choose between the tables.

Method	Diagram
Single table. Target AFR is determined from AFR1. This is the most common setup.	
Switched tables. Target AFR is determined from AFR1 or AFR2. A physical switch input determines which table is used. 	

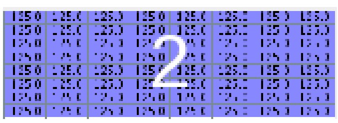
Blended tables. Target AFR is determined as a blend between AFR1 and AFR2 using Blend Curve (5)  	

The target AFR is then be used for closed loop EGO or for Incorporate AFR.

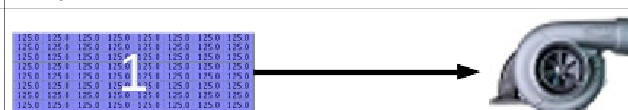
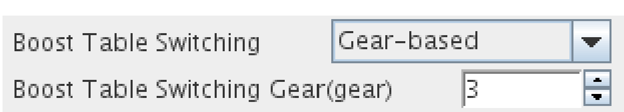
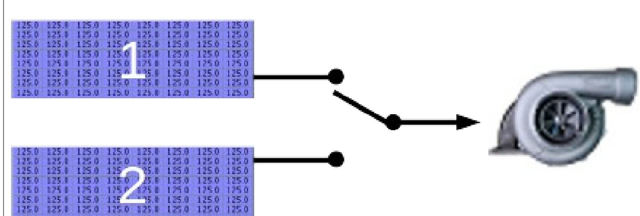
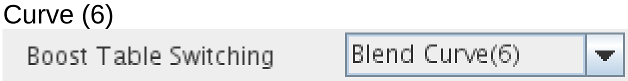
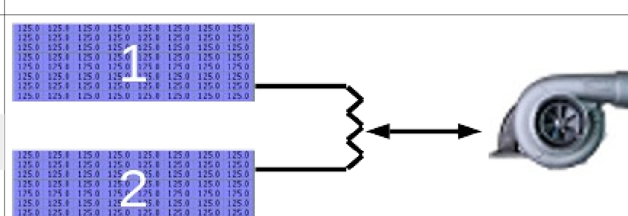
Note that even in "Dual Table" mode only a single AFR target number is in use at one time i.e. both channels will target the same number if closed loop is enabled.

4.20.7 Boost blending and switching types

There are two boost tables, boost1, boost2



In Closed Loop mode these two tables are target boost. In Open Loop they are raw boost solenoid duty cycle.
There are a number of options to choose between the tables.

Method	**Diagram**
Single table. Boost is determined from boost table 1. This is the most common setup.	
Switched tables. Boost is determined from boost table 1 or boost table 2. Either a physical switch or a gear option determines which table is used.   	
Blended tables. Boost is determined as a blend between boost table 1 and boost table 2 using Blend Curve (6)  	

4.20.8 Table-Switching and Dual Fuel

The MS3 allows for various dual fuel settings to allow for liquid vs. gaseous fuels etc. Temperature and pressure compensation tables are provided.

The settings are covered in the TunerStudio reference.

4.21 Wheel Speed, Shaft Speed and Gear Detection

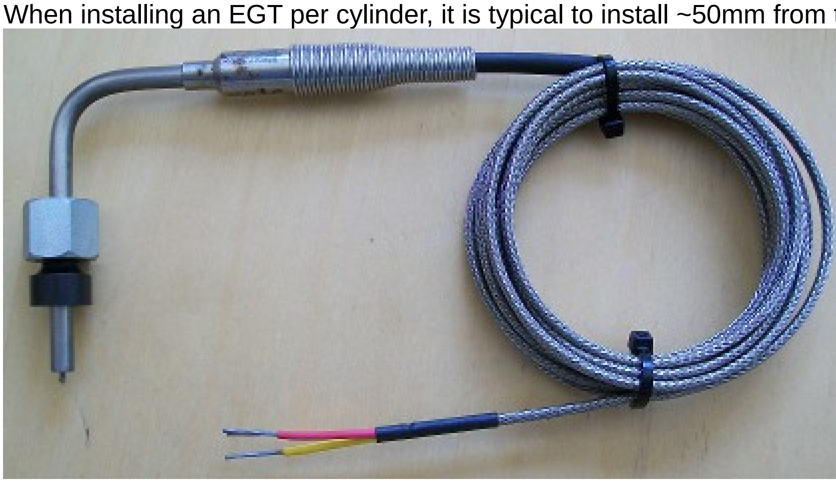
The vehicle speed sensors (VSS) allows up to four wheel speeds to be monitored. The inputs read from either a wheel/axle mounted pickup or a pickup that is part of the transmission. The input can be magnetic (with suitable interface) or more directly a hall type input. This input is converted to a linear speed (ms^{-1}, mph or kph.) For optimum detection of speed from rest (e.g. a racing launch) use a geartooth or hall sensor and a multi-toothed trigger wheel.
The shaft speed inputs work in a similar way to the vehicle speed but calculates to RPMs. It could be used as gearbox input shaft speed to monitor convertor or clutch slip or for turbo impeller speed.
The VSS output feature provides a pulsed output in proportion to the VSS for driving a digital speedometer.
The gear detection calculates the current gear selected in the gearbox by a number of methods, this can be used for features such as boost by gear.

Refer to the hardware guide for wiring examples.

4.22 Exhaust Gas Temperature (EGT)

The EGT system in Megasquirt-3 allows inputs from EGT sensors to be datalogged by serial or SDcard. It optionally allows the engine to be shutdown if EGT exceeds a set temperature. If an EGT probe is fitted to each exhaust port it can be used in conjunction with per-cylinder fuel trim to balance cylinders to account for air flow and injector flow differences. EGT is used extensively in piston driven aircraft and much useful information can be found from aircraft literature and then applied to your non-aero engine.
(EGT differences between cylinders can be significant at idle, evidence shows that V8s often run cold on the four corner cylinders and hot on the inner four. One engine required over 10% fuel trim to balance these air-flow differences.)

To use the EGT system an EGT-probe (usually a K-type thermocouple) is installed in the exhaust manifold. When installing an EGT per cylinder, it is typical to install ~50mm from the head.



The output from a thermocouple is a very small voltage and a thermocouple amplifier is required. Options are covered in the hardware manual.

- an external thermocouple amplifier that provides a 0-5V analogue output that is then connected to one of the Megasquirt-3 analogue inputs.
- an external thermocouple amplifier box that connects to Megasquirt-3 via CAN.

- a DIY option that builds an amplifier into the Megasquirt-3.

4.23 Generic Sensors

The Generic Sensors allows optional analogue inputs to be connected and easily recorded in datalogs or used as parameters in the Output Ports. The analogue inputs are in the 0-5V range and are internally converted with a 10bit ADC (Analogue to Digital Convertor) giving 1024 steps. Typical uses include:
-pre- and post- intercooler pressure and temperature
-fuel pressure
-nitrous pressure
-exhaust back-pressure
-pedal position
-suspension travel

The MS3X card provides three ready to use 0-5V analogue inputs. With some DIY there are two more analogue inputs available on the mainboard. For even more, a CAN connected extension board may be used up to a maximum of 16 supported inputs.

The converted signals are then available for datalogging or for other internal features.

4.24 Accelerometer input

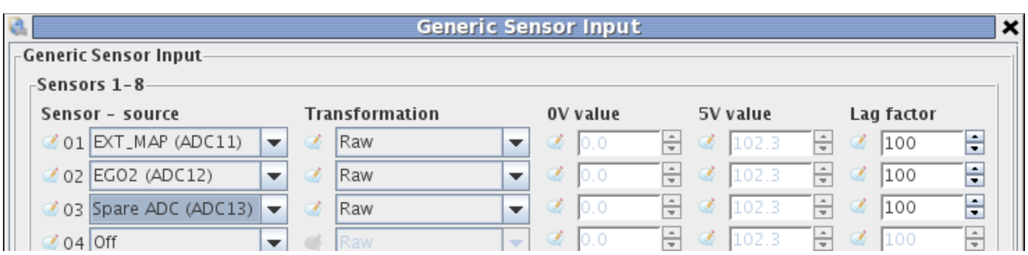
The accelerometer input allows for a user-supplied one, two or three axis accelerometer to be connected to the Megasquirt and datalogged. It may be connected to any of the five spare analogue inputs or remotely via a CAN extension board (e.g. The JB-perf IO-Extender can be supplied with a 3 axis accelerometer on-board.)
The accelerometer is calibrated to +/-1g for convenience, but internally the code uses SI measurement units (ms-2) so accelerometer acceleration can be compared to vss measured acceleration.

During the wiring step you should have identified which ADC ports are connected to your accelerometer. The accelerometer dialogue allows you to define these ports and the calibration.

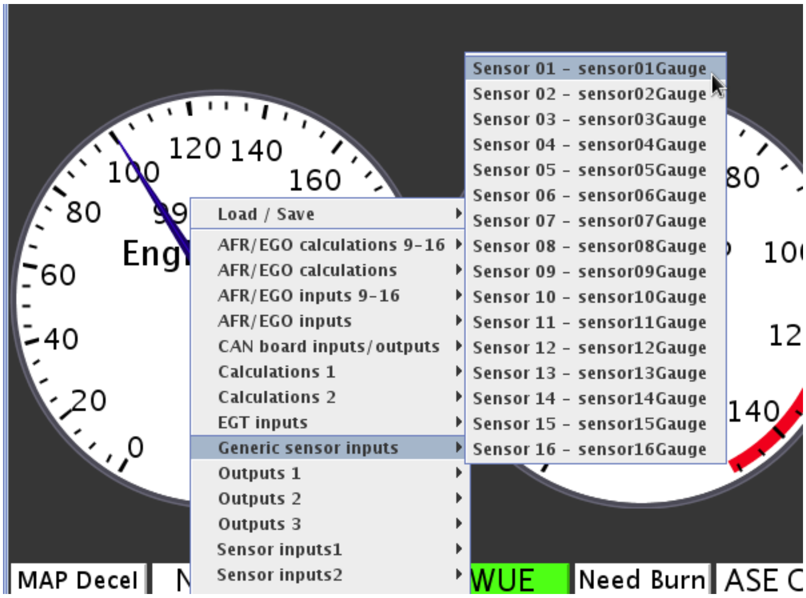
However, **BEFORE** enabling the accelerometer, you need to collect the calibration data.

Collecting calibration data
In this example we will assume that the three axes are connected to ADC11, ADC12, ADC13.

1. Ensure that the accelerometer is wired in and powered on.
2. Open up Generic Sensors and define three spare 'sensors' with the three accelerometer inputs.



3. Next, define a gauge on your TunerStudio dash to match one of these inputs. Right click on a gauge and then re-define it to match one of the sensor inputs you just defined.



4. Physically move and rotate the accelerometer until it gives the minimum reading on that gauge. Write it down.

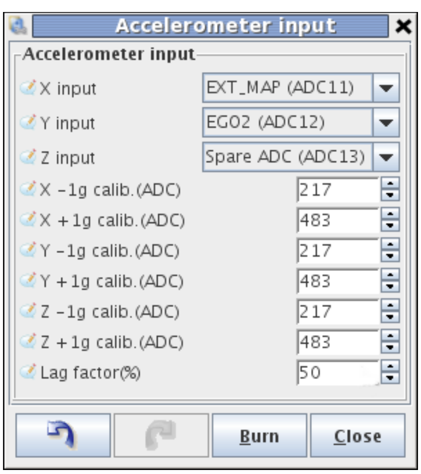
5. Repeat for the maximum reading on that gauge. Write it down.

The minimum value will be for -9.81ms-2 (-1g) and the maximum will be for +9.81ms-2 (+1g).

5. Repeat for all three axes.

6. Return to the Generic Sensors screen and turn off the three inputs you defined.

7. Go to the Accelerometer dialogue and enable the three inputs and enter the pairs of calibration data you captured



8. The process is now complete and in future the acceleration will be recorded in datalogs as accelX, accelY, accelZ.

4.25 Traction Control

The MS3 offers several strategies for traction control:

- VSS Slip %: The MS3 compares a speed sensor on the driven wheels (VSS1) to one on the non-driven wheels (VSS2), and reduces power if the driving wheel speed exceeds the non-driven wheel speed by a specified amount.

- Perfect Run: The MS3 monitors a single speed sensor. You specify the maximum speed the vehicle can reach in a given amount of time, and the MS3 considers any acceleration faster than this to be wheel spin and engages traction control. As this strategy is intended for drag racing, Perfect Run traction control requires starting with launch control active to engage. Check with your sanctioning body whether traction control is permitted in your class!
- Perfect Run RPM: A variant of traction control for racing events which do not allow the ECU to monitor vehicle speed. In this mode, you specify the engine RPM as a function of time. The MS3 will engage the traction control if the RPM increases faster than the specified curve. Like regular Perfect Run, this mode requires starting with launch control active before it engages.
- Switch Input: The MS3 engages traction control in response to a signal from an external traction control module. Used on cars with OEM traction control modules which send an on/off signal, such as some Corvettes.

Traction control allows several different responses to reduce power, depending on what features you are using.

- Retarding the spark timing.
- Using a rolling spark cut.
- Adding fuel.
- Reducing or shutting off a nitrous oxide injection system.
- Reducing boost on a turbocharged motor.

4.26 Launch, 2-step, 3-step, Transbrake

There are a number of features designed for race vehicles. Most require some inputs or outputs connecting. See your Hardware guide for "Switch inputs" and output wiring.

4.26.1 Launch and Flat-Shift

The Megasquirt-3 Launch control system is effectively a 2-step limiter and is typically used in drag racing applications for consistent hard launches and/or to build boost with a turbo-charged engine.
The flat shift settings allow full throttle shifts on a clutched vehicle, a switch on the clutch triggers the flat-shift rev limiting (for bikes, see also the bike shifting system as that may be more suitable.) It is important to install the clutch switch with adjustability as the exact height has an effect on how well flat-shifting works. Flat shift is unlikely to be useful when running an automatic transmission.

4.26.2 Variable Launch

With the additional of an external potentiometer (0-5V signal) the exact launch RPM can easily be adjusted at the track without using a laptop.

4.26.3 Trans-brake / Timers / Bump-Box

The transbrake control offers basic delay from button release (top amber) to trans-brake release or for turbo-cars the turbo-car staging feature allows you to pre-stage, build boost then "bump" the trans-brake to creep forwards into full-stage.

4.26.4 Throttle-stop

Linked to the trans-brake control is throttle-stop control, this allows a throttle-stop to be activated at a set time after the trans-brake button is released for bracket cars.

4.26.5 3-step

The 3-step is typically used as a burnout limiter and will often be linked to a line-lock.

4.26.6 Timed Retard After Launch

Many high power cars can easily overpower the start-line, this feature allows ignition timing to be retarded at the start of run and fed back in as the cars gets moving.

4.26.7 Drag race example (manual trans)

You pull forward into the burnout box, stand on the brakes and flick on the line-lock. This also enables the 3-step. You now do your burnout and the 3-step limits are applied. After your burnout you pull into stage, depress the clutch (activating the launch switch) and then push the throttle to the floor. The launch settings will hold the engine close to your chosen rpm. When you change gear keep the throttle planted and only use the clutch. Now the flat shift settings come into play and the revs will be limited to those settings. After changing gear and lifting your foot off the clutch, the limiter is disabled. Be sure to also set the normal rev limiter should you miss a shift.

4.26.8 Speed based launch control

This is another variation of launch control that can be useful for autocross. The RPM limit and timing retard can be adjusted by wheel speed. In this case VSS1 should be connected to a non-driven wheel.

4.27 Sequential shift-cut

Another drag racing feature, this one is intended for use with air shifters and sequentially shifted transmissions. The MS3 can cut the ignition during shifting and also control the air shifter solenoid valve. Note that automatic shift control and gear based delays require setting up the gear input under Speed and Gear Sensors. The shifting will happen in this sequence:

1. As soon as the button is pressed or automatic shift is required, the air shifter solenoid is turned on.

2. The MS3 pauses for a time defined by the shift cut delay before cutting spark.

3. The MS3 waits for the specified spark cut time plus any gear based delay, if applied. During this point, the solenoid turns off before the ignition resumes.

4. The ignition resumes firing.

5. The MS3 starts the re-shift hold off timer. The MS3 will not shift again until this timer runs out.

4.28 Nitrous Oxide (N2O)

Nitrous oxide is a power-adder frequently used in drag racing to add 50-100% power to an engine for short durations. **Used incorrectly it can melt pistons and destroy engines in seconds.**

The nitrous system within Megasquirt allows control over the nitrous and optional fuel solenoid to enable the nitrous injection when desired and also retard the ignition timing. The system supports two stages of on/off control.

Fully tuning nitrous is a topic beyond the scope of this manual, we shall just cover some basics.

4.28.1 System types

There are two main categories of nitrous systems - wet and dry.

Wet

A wet system injects fuel into the manifold ahead of the injectors (e.g. under the throttle body,) this can result in fuel distribution problems and mixture varying between cylinders.

The simplest example of a wet system is a "plate" system that installs under the throttle body (or carb) above the intake manifold and utilizes spray bars for nitrous and fuel. Other variations exist where single or multiple nozzles are utilized.

A standalone wet nitrous system has the advantage that when used with the suppliers' suggested jetting it will typically install and work. (More hardware, less tuning.)

Dry

With a dry system the additional fuel required for the nitrous is injected by increasing the pulsewidth on the existing fuel injectors. In a port-injection system this minimizes the fuel distribution problems. A dry system can also have fewer system components as the pre-existing fuel system is utilized. However, because the fuel and nitrous can be controlled independently, the software settings are critical to getting the mixture correct. (Less hardware, more tuning.)

On/Off

With On/Off control the nitrous/fuel solenoids are either On or Off. This is the standard operating mode.

Progressive / pulsed

Megasquirt-3 does not implement progressive control

Multi-stage

Note! - you are advised to always start with a simple system using small jets. Big hits with multiple stages are significantly more difficult to install and tune correctly.

With a single on/off stage using fixed jetting the flow of nitrous (and fuel) is approximately constant under all conditions. This means that if you are getting a claimed 100hp at 3000rpm when you activate the system, by 6000rpm the engine is turning twice as fast and each intake event will only receive half the nitrous/fuel so you will get half the torque benefit.

One answer might be to use two stages based on rpm. The first stage is still jetted at 100hp, but at 4500rpm you bring in a second stage of say 25hp. This will make up for some of the "lost" torque.

4.28.2 Connection

Wiring of the solenoids / relays is covered in the Hardware Guide for your product.

4.28.3 Tuning considerations

In general, applying up to 50% extra hp to your engine with a kit manufacturers' jetting and suggested retard works well and will not cause problems to your engine. However, it is very important to take plug readings and check for any signs of lean-ness or detonation. As mentioned already, it is quite easy to destroy your engine with nitrous given the wrong combination.

Easy mistakes to make with bad consequences:

- Mixed up fuel/nitrous jets.
- Fuel pressure too low (mismatched to fuel jet size)
- Incorrect dry fuel pulsewidth

- Insufficient timing retard
- Too hot a spark plug
- Faulty fuel solenoid or wiring
- Blocked fuel lines or jets
- System activated at too low an rpm

4.29 Water Injection

Water injection allows for control of a water pump and mapped control of injector pulses. This is typically used on boosted engines to keep intake temperatures in check.

4.30 High Power Time Enrichment

Excessive internal temperature will easily lead to engine destruction. While your engine may be able to withstand a certain mixture for short period of time, it may not cope for too long as piston temperatures rise. This feature allows the mixture to automatically be enriched when full load is used for extended periods with the aim of protecting the engine.

4.31 Oil Pressure

The oil pressure system allows the monitoring of oil pressure and warning if pressure is too low at a given RPM.

4.32 Programmable On/Off Outputs

This screen is for setting up generic on/off outputs for features like shift lights, variable intake control solenoids, and other items that don't need complicated control settings. These settings are not required when the outputs are used for other built-in features.
(e.g. do not try to configure a spare port FIDLE if you want to use FIDLE with the idle-control system. If you have configured an output to be used for two features the code will report a 'configuration - fix the mistake in your settings.)

Typically the output will be wired to a relay, a small lamp or a solenoid.

All of the MS3 realtime data channels can be used as parameters. Bit-field variables can be used also with the '&' operator. e.g. you could configure a lamp to light up if you hit the rev limiter.

4.32.1 Bitwise operations

The bitwise AND operation requires some further explanation and an example. Take the "engine" data channel. From the TunerStudio Reference manual, you can see the definition:

128	64	32	16	8	4	2	1
MAP decel enrichment	MAP accel enrichment	TPS decel enleanment	TPS accel enrichment	WUE active	ASE active	Cranking	Ready (active tach signal and pump running)

If you are unfamiliar with bits, bytes and binary, then some background reading may be required.

For bitwise operations,

result = ((value & THRESHOLD) == HYST)

Say you wanted an output to activate when you are in WUE (warmup) mode, but not TPS accel.

TPS accel has a value of 16. WUE has a value of 8.

We want to check both of these bits, so need to set the "threshold" (bitmask) to 16 + 8 = 24.

We then want to check that TPS accel is clear and WUE is set, so set the "hyst" (match) = 8.

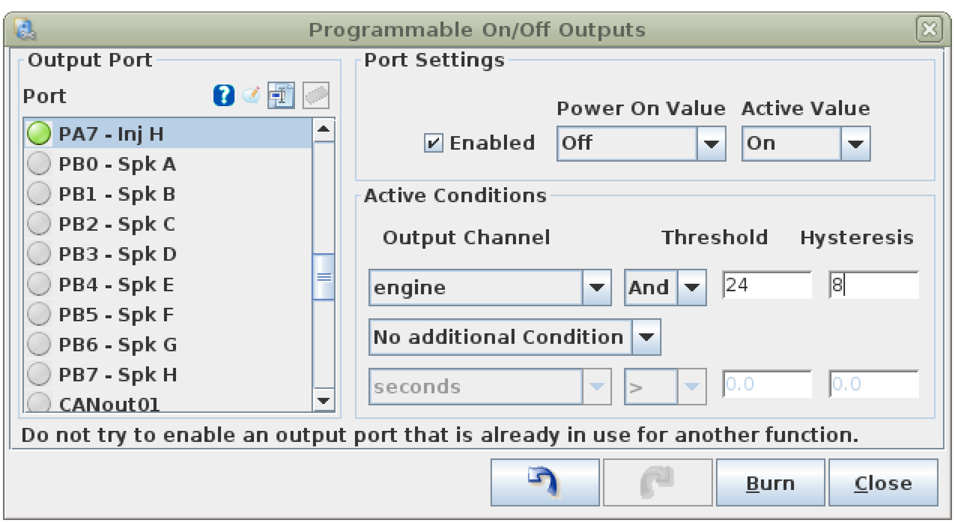
	128	64	32	16	8	4	2	1
Field name	MAP decel enrichment	MAP accel enrichment	TPS decel enleanment	TPS accel enrichment	WUE active	ASE active	Cranking	Ready
Example value	0	0	0	0	1	1	0	1
Bitmask (threshold)	0	0	0	1	1	0	0	0
Answer	0	0	0	0	1	0	0	0
Match (hysteresis)	0	0	0	0	1	0	0	0

In a given column (bit) a bitwise AND operation is performed - if the value and the mask are both '1', then the answer is '1'. In this example,

TPS accel has 0 & 1 -> 0

WUE has 1 & 1 -> 1

All 8 bits of the answer row are then compared to the match value row. In the example it is equal, so the on/off output is enabled.



4.33 Generic PWM

The Generic PWM system allows up to six independent open-loop PWM (pulse width modulated) or on/off outputs. The output is typically used to drive a solenoid valve (PWM mode) or relay (on/off.) PWM is a method that pulses an output on and off, the on-time percentage (duty) is variable, this is often used for idle valves and allows a valve to "float" in-between closed and open for variable flow rates.

The duties are programmed through either:

- a 6x6 table of output duty from RPM vs. load Y-axis.

- a 12 point curve of output duty from load.

For the load you can choose from any of the realtime data values, including any additional sensors you added.

4.34 Generic Closed-Loop

The Generic Closed-Loop system allows up to two independent closed-loop PWM (pulse width modulated) channels. The output is typically used to drive a solenoid valve (PWM mode.)

The target values are programmed through an 8x8 table of target from RPM vs. load Y-axis.

The feedback and load values are chosen from any of the realtime data values, including any additional sensors you added.

4.35 Check Engine Light (CEL) / Limp Mode

The MS3 has a sensor validation system that can sanity check the sensor inputs. If the sensor appears to be malfunctioning, then it can switch on a Check Engine Light. Additionally, limited operating strategies (limp modes) can be enabled. The aim of these settings is firstly to warn you of a failed sensor that could cause poor or damaging engine operation and secondly, through the limp modes, to allow some engine operation even with the failed sensor. The checking scheme varies slightly for each sensor. Most sensors are checked against minimum and maximum possible values that can occur if the wire breaks or is shorted to ground or 5V. The other criteria is minimum and maximum fluctuation - this is to detect a stuck or wildly flailing sensor. All of these parameters are engine specific, and you will need to dial them in for your installation.

The sensor validation often works with ADC counts. ADC stands for analog to digital converter, and refers to a circuit in the MS3 processor that converts an analog voltage to a digital readout.

The MS3 converts sensor readings into a 0 to 5 volt signal that goes to the processor, and the processor uses a 10 bit converter to turn this into a value that ranges from 0 at 0 voltage to 1023 at 5 volts. Usually, these ADC counts are hidden from the user - you're more interested in what the sensor reading actually means than a raw number. However, with sensor diagnostics, you'll be looking at the ADC count directly, since we're looking at establishing when a signal doesn't mean anything other than that your sensor isn't working. For starting settings, try a minimum ADC value of 5, and a maximum ADC value of 1018.

The fluctuation units are a bit arbitrary; we recommend data logging the values to see what is normal, and using that to set the minimum and maximum values.

During configuration and testing, the "status5" gauge (found under X-code dev) can be set to display the raw ADC value for a sensor or the current fluctuation value. Datalogging these values under a range of operating conditions will help you learn appropriate values for you particular engine.

4.35.1 Configuration tips

MAP sensor - the minimum fluctuation setting is likely the most useful setting here as it can be used to detect a failed or 'stuck' sensor or if a hose fell off.

CLT sensor - this should be smooth in operation, the max fluctuations can be used to detect a fault.

TPS - The min/max limits will often detect a fault.

Battery - Maximum fluctuation can be used to warn of charging system faults or ground issues causing excessive voltage 'noise'.

EGO sensor - The fluctuation settings are key to detecting a failed sensor. Many wideband controllers will get 'stuck' if the sensor fails although appearing to show a valid AFR reading. Combined with closed-loop EGO or autotuning this can be troublesome without this detection scheme. Other controllers allow an 'out of range' voltage output - match the min/max ADC readings to capture this. Under normal operation, the AFR is unlikely to read one value continuously, use the minimum fluctuations setting to detect a fault.

EGT - The temperatures are not checked immediately after startup, so the minimum temperature can be set to 200°C to detect a dead cylinder or faulty sensor. The maximum will depend on your engine. N/A could be around 900°C, boosted perhaps 1000°C.

4.35.2 Limp Mode

After detecting a sensor fault, the MS3 can take action based on what it has detected - either substitute a replacement sensor reading, ignore the sensor, or use an alternate sensor. The limp mode can also impose a low rev limit and limit boost. The purpose of the limp mode is to get you home or off the race-track etc. without damaging the engine. Certain faults will cause poor operation; the aim is to allow some operation in preference to no operation.

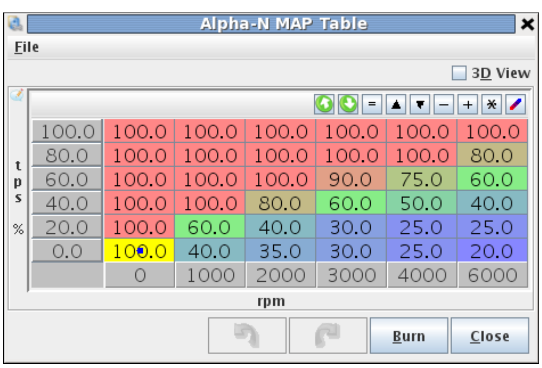
Here is what action the MS3 can take in response to sensor failures.

- MAP sensor failure: The MS3 can either continue running on the MAP sensor but apply the rev and boost limits, or it can substitute a fallback MAP table that guesses the MAP reading based on throttle position and RPM.
- MAT sensor failure: Substitutes a fixed sensor reading.
- CLT sensor failure: Approximates a coolant sensor reading based on time after start. Note - It's a good idea to set the warmed up value to a point high enough to turn your cooling fan on, if the MS3 is controlling the fans.
- TPS failure: Disables TPS based acceleration enrichment and assumes a zero TPS value.
- EGO sensor failure: Turns off all closed loop O2 correction.
- Flex fuel sensor failure: Uses the fallback timing addition and fuel multiplier defined on the flex

  fuel screen.

4.35.3 Fallback MAP table

In the event of a MAP sensor failure, the MS3 can use a 'synthesized' MAP value based on TPS and RPM. This will be less accurate than the real MAP reading, but the aim is to allow some driveability in the absence of a real sensor reading before the root cause failure is fixed.



tps %						
100.0	100.0	100.0	100.0	100.0	100.0	100.0
80.0	100.0	100.0	100.0	100.0	100.0	80.0
60.0	100.0	100.0	100.0	90.0	75.0	60.0
40.0	100.0	100.0	80.0	60.0	50.0	40.0
20.0	100.0	60.0	40.0	30.0	25.0	25.0
0.0	100.0	40.0	35.0	30.0	25.0	20.0
rpm	0	1000	2000	3000	4000	6000

The initial vales for this table can be derived from datalogs comparing MAP, RPM, TPS while driving. You can use MegaLogViewer's scatter plot feature (registered version only) to see what MAP reading is common for the throttle position and RPM range in a data log; the shape of the scatter plot will be close to the table required.

To test and tune this table, configure MAP sensor checking and limp mode, then unhook the MAP hose and wait until 'Limp Mode' engages. Then this table will be operative and you can tune.

4.35.4 CEL Flash Codes

Optionally, flash-codes can be enabled on the Check Engine Light.

The 1.3 firmware supports flash codes on the check engine light.

# flashes	Reason	CEL Status	CEL Status2
2	MAP fault	1	-
3	MAT fault	2	-
4	CLT fault	4	-
5	TPS fault	8	-
6	BATT fault	16	-
7	AFR0 fault	32	-
8	SYNC fault	64	-
9	EGT fault	128	-
10	Flex fault	256	-
11	MAF fault	512	-
12	Knock fault	1024	-
13	Cam fault	2048	-
14	Oil fault	4096	-
15	Fuel press	8192	-
16	EGT shutdown	16384	-
17	AFR shutdown	32768	-
18	W.Inj low	-	1
19	Safety shutdown	-	2

There will be a long flash to indicate the start of a sequence and then a series of short flashes. Count the short flashes. e.g.

- . . . .

is code 11.

4.36 CAN Bus Expansion

CAN is a two wire in-vehicle communications system that allows different vehicle computers to "talk" to each other. In essence it is an expansion wiring system.

Megasquirt 3 comes with CAN and allows it to communicate with other (Megasquirt) CAN enabled devices such as 3rd party dashes, CANEGT, GPIO, JBperf's IO-x and likely other add-on boards.

Benefits:

- Simple connection. Just connect the two wires.
- Connect additional sensor inputs or control outputs.
- Multiple Megasquirt devices can share sensor and other data over the CAN network.
- You can tune multiple Megasquirt devices concurrently with your computer plugged in to one.

There are two main modes used over CAN

- Device-to-device - where the ECUs exchange data with each other autonomously
- Passthrough - where TunerStudio is used to tune a device remotely over the CAN network

4.36.1 CAN protocol

The Megasquirt system as standard uses a proprietary messaging system using 29bit headers. This makes it incompatible with many other third party CAN devices.

A small number of standard 11bit header broadcast messages are supported in firmware 1.3.x

In firmware 1.4.x greater support is added to broadcast and receive standard 11bit header messages for greater interoperability. Release is planned for Q1 2015 as a free upgrade for Megasquirt customers.

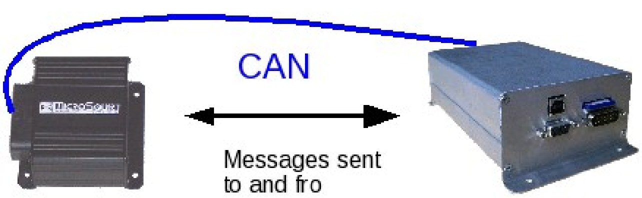
Further details of the internals of the Megasquirt-CAN system are available in a technical note.

4.36.2 Device-to-device

In this mode two devices on the CAN network exchange data independent of the tuning computer.

e.g. A transmission controller fetching engine data from the Megasquirt3, or the Megasquirt 3 polling devices on the network and datalogging the results.

Once configured the data transmission will happen as and when required with no user intervention.



The two devices request and receive data on their own, independent of any tuning computer

4.36.3 Passthrough

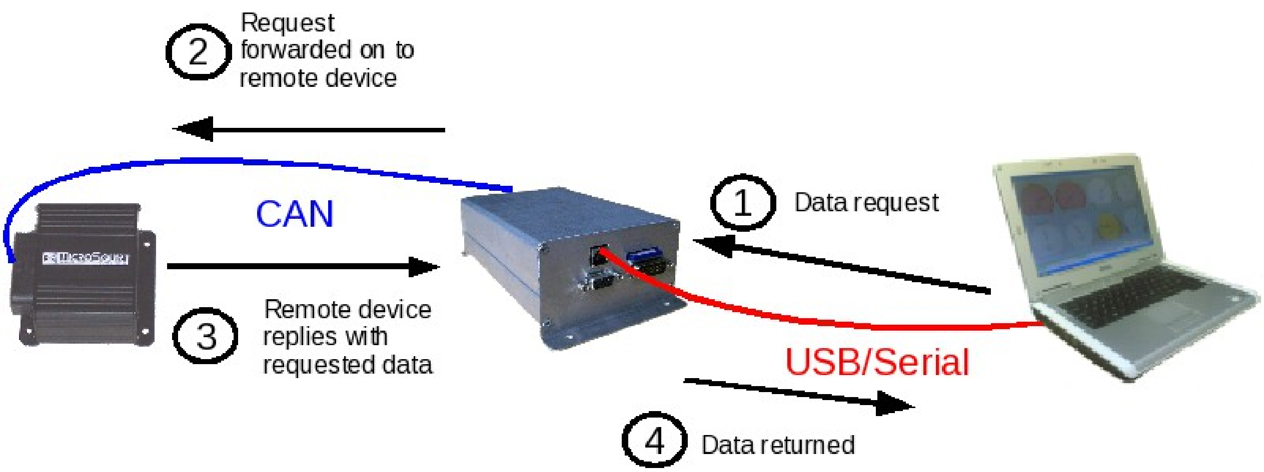
In this mode messages from your tuning computer are relayed to remote devices over the CAN network.

e.g. tuning the remote device from your laptop.

This can be really useful when you have more than one Megasquirt device on your vehicle, with a single connection point for your tuning computer you can "connect" to any of the devices on the network and view data, tune etc. You could also use the computer to datalog from multiple devices concurrently. The logging speed will be reduced the more devices you try to log from.

TunerStudio allows CAN devices to be added to the main project so you can view and tune multiple CAN devices concurrently. However, you should consider carefully if this is right for your application. Usually the best approach is to create a project for the remote device alone, use that project for configuration and then let the MS3 fetch data using Device-to-Device mode - this will result in faster logging and is less complicated.

The tuning computer is connected to the Megasquirt-3 either by serial or USB-serial.
The Megasquirt-3 and remote device are connected by the two wires of CAN.



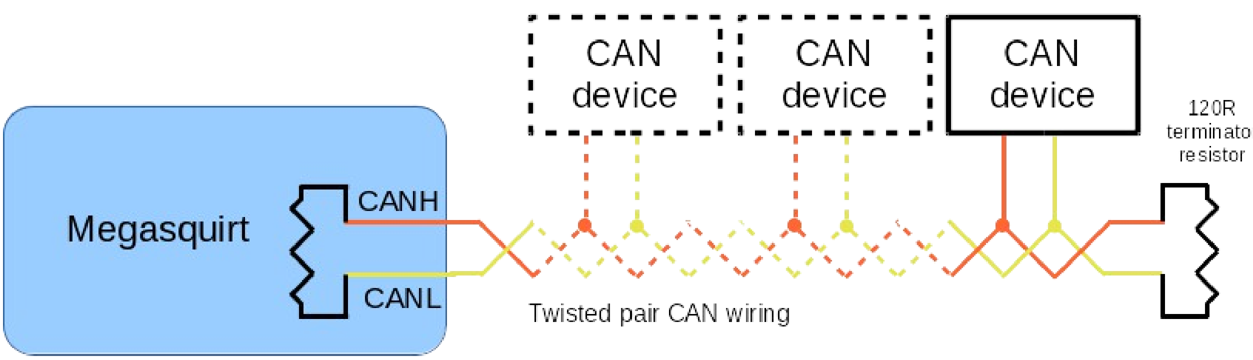
Step 1.
Tuning computer sends USB/serial request to Megasquirt (addressed to remote device.)

Step 2.
Megasquirt 3 recognizes this as a passthrough request and relays it on to the remote device over CAN

Step 3.
Remote device answers request and sends back data over CAN.

Step 4.
Megasquirt 3 receives data and forwards it on to tuning computer by USB/serial.

4.36.4 CAN wiring



<u>Key points to note.</u>

- CANH to CANH,
- CANL to CANL
- One 120 ohm termination resistor at each end of the "bus" - this will almost always be inside the box. The Megasquirt 3 comes with the resistor already in place, so no action is required.
- Note that neither wire is connected to ground.
- If your Megasquirt-3 is pre-assembled, ensure that the CAN connections are actually present inside the box.

Otherwise - complain to your vendor!
If you assembled the board yourself, then make sure that YOU installed the jumper wires.
On V3.0 and V3.57 boards you need the following jumpers

- JS6 - SPR1/CANH
- JS8 - SPR2/CANL.

- If you only have two devices connected then the "bus" will simply be two wires running from one box to the other, as shown in the MS3/Microsquirt example diagram above.
- If you are connecting three or more devices you will need to connect them as above and ensure only the two end devices have terminators and the middle ones do not.
- For short runs of cable a simple pair of wires is likely ok, for large runs it may be advisable to use twisted pair or twin core screened cable.

4.36.5 CAN settings and data types

The Megasquirt-CAN network typically works by passing along sections of memory tables. As a result, many of the settings will deal with telling the MS3 where the table is in memory and how to retrieve it.

4.36.5.1 PWM polling

Allows frequency data to be collected from an expansion device. The "PWM" channel can then be used by the speed system to calculate VSS.

4.36.5.2 Digital ports

This allows the MS3 to use up to eight remote on/off inputs (CANIN1-8) and up to sixteen remote on/off outputs (CANOUT1-16).
CANIN1 is bit 0 from the remote input port and CANIN8 is bit7.
CANOUT1 is bit 0 from the first remote output port and CANOUT8 is bit7.
CANOUT9 is bit 0 from the second remote output port and CANIN16 is bit7.

4.36.5.3 PWM outputs

Allows PWM/frequency outputs to be used on an expansion device. e.g. PWM Idle.

4.36.5.4 CANADCs

Allows analogue data to be collected from an expansion device. The captured data can then be used by features such as Generic Sensors, EGT or accelerometer.

4.36.5.5 VSS

Allows pre-calculated speed data (rather than pulse frequency) to be collected from an expansion device.

4.36.5.6 Gear

Allows a gear number to be collected from an expansion device.

4.36.5.7 Innovate EGO

Allows Innovate wideband data to be collected from a JBperf IO-extender or CANEGT.

4.36.5.8 GPS

Allows GPS data to be collected from a JBperf IO-extender.

4.36.5.9 RTC

Allows date and time to be collected from a JBperf IO-extender.

4.36.6 CAN Broadcasting

This allows a handful of pre-defined standard 11bit CAN messages to be broadcast for specific customer applications. (More could be added - please ask.)

4.36.7 Example Settings - CANEGT

The **CANEGT** from DIYAutoTune.com makes use of the Megasquirt CAN system and allows the MS3 to read 8 thermocouples for EGT and collect data from Innovate widebands over serial (using the serial connection gives a more accurate reading than the traditional analogue connection.)

Enable ADC polling = Enabled

MS3 CANADC	Can id	Table	Offset	Function
CANADC1-4	1	7	2	EGT1-4
CANADC5-8	1	7	10	EGT5-8

On the EGT / Thermocouple Inputs screen. EGT channels 1 through 8 correspond to CAN ADC01 through CAN ADC08. For Calibration, set Temp at 0V to 32°F (0°C) and Temp at 5V to 2282°F (1250°C)

On the CAN VSS, Gear, EGO, GPS screen set

Fetch Innovate EGO data = Enabled

Remote CANid = 1

Remote Table = 7

Remote Offset = 153

Under Fuel Settings, select AFR / EGO Control, and enter the number of sensors. Set all EGO ports in use to CAN EGO. You may then assign the EGO sensors to their respective cylinders under AFR / EGO Sensor Mapping.

4.36.8 Example Settings - JBperf IOextender

The **Extender** from JBPERF makes use of the Megasquirt CAN system and allows the MS3 to collect the data from the Extender's inputs. The Extender has up to:
8 general or thermistor inputs
8 thermocouple EGT inputs
3 axis accelerometer
Realtime clock
Global Positioning System

Refer to the Extender documentation on the JBPERF site for initial configuration. Once you have wired and set up the device, you need to configure the MS3 to use the data. This is working in Device-to-Device mode - do not add the IO-extender to your main MS3 project.

ADC inputs

Up to 24 analog channels are available from the IO-Extender. Once these are captured they may then be

used with native MS3 features.
Enable ADC polling = Enabled

MS3 CANADC	Can id	Table	Offset	Extender ADC	Function
CANADC1-4	5	7	2	ADC0-3	general ADC inputs
CANADC5-8	5	7	10	ADC4-7	general ADC inputs
CANADC9-12	5	7	18	ADC8-11	EGT inputs
CANADC13-16	5	7	26	ADC12-15	EGT inputs
CANADC17-20	5	7	34	ADC16-19	Battv, accelerometer inputs
CANADC21-24	5	7	42	ADC20-23	3v3 ref, general ADC inputs

NOTE: Be sure to set the Extender to 10bit ADC for compatibility with MS3.

Digital ports
You can configure ports 1 and 2 as digital output ports (16 channels) and port 3 as a digital input with the following settings.

CAN device = 5

Table = 7

Enable input port = Enable

Offset = 77

Enable output port = Two

Offset = 75

4.36.9 Example Settings - other devices

Information request - if you have settings that work with other standard Megasquirt devices, please post on the www.msextra.com forum, so that information can be included here.

4.37 Test Modes

The Megasquirt includes a number of test modes that are helpful during inititial installation and for fault diagnosis.

- Output test mode inj/spk - Injectors and coils testing.
- Output test mode I/O - Allows outputs to be turned on, off or pulsed.
- Output test mode idle valve - Idle valve testing.
- Injector sequential testing - Allows temporary change to injection mode.
- Inj/Spk disabling testing - Allows individual coils or injectors to be turned off. This can be used on sequential installs to check for any "dead" cylinders or to correlate per-cylinder EGT or widebands. Beware of flooding cylinders when disabling their coils.

4.38 Long Term Trim (LTT/LTFT)

Long-term fuel trim can be used to automatically make minor corrections to fuelling to allow for operating changes. It should only be used **after** the engine is fully tuned. It requires EGO control to be enabled and operational.

In operation, the system monitors the current EGO correction percentage and adjusts Long Term Trim Table. It is desirable for this to be a small correction so transient effects are ignored. The Sampling Interval and Softness settings control this.

Typically the sensor should be sampled every 30 seconds or so with a write to flash every 10-15 minutes. Internally, the Megasquirt maintains two copies of the trim table, so that in the event of power being shut-down during a table write there should be a workable copy preserved.

The Long Term Trim Table is used the same as other lookup tables and the correction factor is applied to the fuel pulsewidth. No changes are ever made to the VE tables - the LTT table is intentionally independent. By having its own table, there is no chance for a defective O2 sensor to corrupt a working VE table.

If the fuel table is retuned, then the LTT table should be reset to zero with the **Zero Tables** button.

The type of memory used within the Megasquirt means that it is not desirable to continually save the changes that are learned, or the flash memory would literally wear out. So instead, there are two methods to save the learned LTT table periodically.

- Button - the user can press a button to force the table to be saved
- Timed - when the Megasquirt decides that the table has changed significantly, it can automatically save the changes.

Due to an issue in TunerStudio 2.6, when Long Term Trim is used you need to disable "High Speed Runtime" on Communications->Megasquirt Baud Rate.

4.39 Race Technology dash integration

Race Technology's "DASH2 PRO" supports CAN input and CAN output with suitable licenses.

4.39.1 CAN input

Enabling the CAN input allows data from the Megasquirt CAN Dash Broadcast to be read onto the dash - RPM, CLT, MAT, injector PW etc.

Race Technology typically supply their units pre-configured to match the target ECU. If you need to reconfigure, you will want to request the file "MegaSquirt ECU CAN DL1 SD - DASH2 PRO.CAN". This can be loaded into the unit using the "DASH2 PRO Configuration" software on the CAN inputs page. The author found that the analogue water temp input needed to be disabled first.

Megasquirt configuration is simply to enable Dash Broadcasting.

4.39.2 CAN output

The CAN output allows the dash to broadcast its internal accelerometer and GPS data to the Megasquirt.

The default data was set for 29bit CAN ID length which is not compatible with Megasquirt. The following settings on the CAN outputs page are required:

CAN baud rate: 500kbit

CAN ID length: 11bit

Tick "Use default RT CAN addresses"

OK

Untick "Use default RT CAN addresses"

Confirm/update the following data:

CAN messages	**Address**	**RTR**	**Max sample rate**	**Note**
Accelerations XYZ	0x300	Disabled	25Hz	Required for Accelerometer input.
GPS time	0x301	Disabled	5Hz	Not currently used by MS3
GPS position LLH 1	0x302	Disabled	5Hz	Required for GPS input.
GPS position LLH 2	0x303	Disabled	5Hz	Required for GPS input.
GPS speed 2D and 3D	0x310	Disabled	10Hz	Required for GPS input.

All other channels should be set to a Max sample rate of "Disabled" unless another device on the CAN is expecting them.

Megasquirt configuration requires setting:

GPS data via CAN = "Race Technology 11bit"

CAN base address = 769 (This is the decimal equivalent of the 0x301 above.)

Fetch Accelerometer Data = "Race Technology 11bit"

CAN base address = 768 (This is the decimal equivalent of the 0x300 above.)

It is possible to receive the accelerometer data on firmware 1.4.x without this specific receiving mode using CAN Receiving. Note - this is not required on firmware 1.5.x !

Enable the CAN receiving master enable and configure the following.

Channel	**ID**	**Offset**	**Size**	**Mult**	**Divide**	**Add**
CANADC22	768	2	L2S	1	1	1000
CANADC23	768	4	L2S	1	1	1000
CANADC24	768	6	L2S	1	1	1000

On the accelerometer page set:

X = CANADC22

Y = CANADC23

Z = CANADC24

X -1g calib = 0

X +1g calib = 2000

Y -1g calib = 0

Y +1g calib = 2000

Z -1g calib = 0

Z +1g calib = 2000

5: Troubleshooting

5.1 Resets

A reset is when a disruption in power or a spike in voltage causes the processor to briefly shut down. This will cause the engine to hiccup while running, and will display a couple other symptoms. The Megasquirt-3 will reset when it is powered up, and this is the right time for it to reset. Having it reset on a running engine, however, is something you'll want to fix right away.

- The Megasquirt-3 will briefly disconnect from the laptop or other devices on USB or RS232.
- Data logs will indicate a "Mark (number) - RESET".
- The SecL count will drop to zero.

The last item is a key part for spotting true resets in a data log. If the engine hiccups, but the SecL count does not drop back to zero, you've got something else, such as a fuel cut, loss of RPM signal, or other issue. If it's a real reset, here are the items we recommend checking first.

First, make sure the Megasquirt-3 has all ground wires used, and grounded to the engine block or cylinder

head. Chassis grounds can have issues caused by rust or points where current runs through nothing but a thin spot weld.

Check where the Megasquirt-3 is getting power from a clean source - ideally, you'll want it to be powered off the battery, with no devices that draw significant current getting power from the same feed. Many resets come not from EMI / RFI interference, but from something like an ignition coil, fuel pump or wideband controller injecting voltage spikes into the power feed.

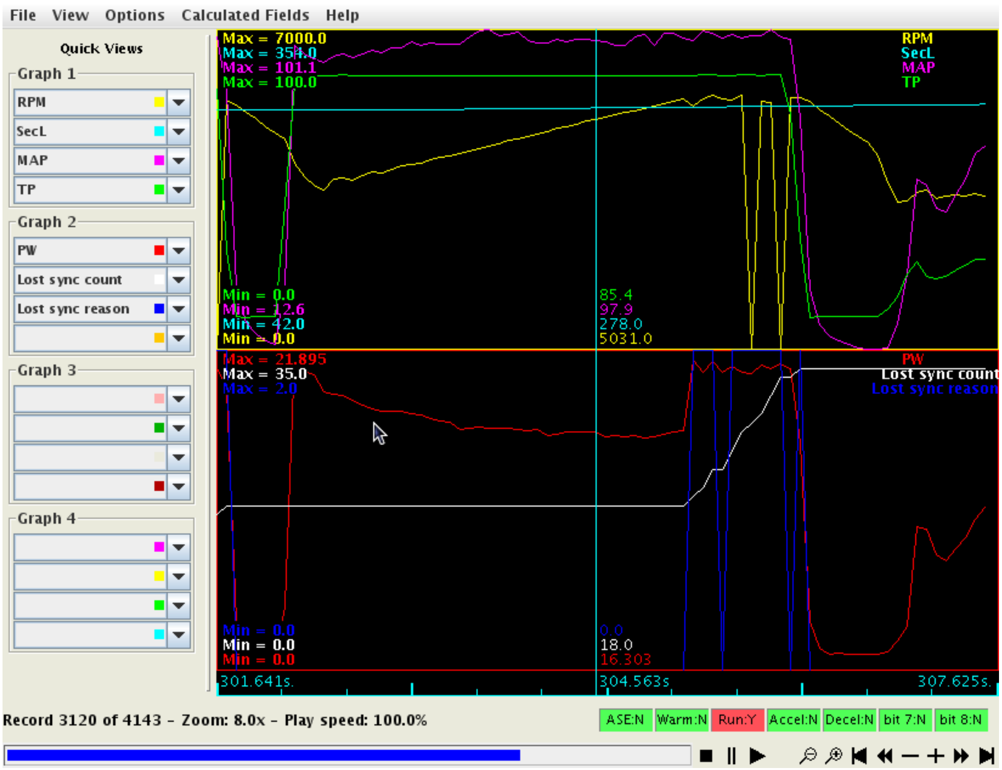
Make sure that you are using resistor type spark plugs, good spark plug wires (if you're not running coil on plug), and that the spark plug wires are not within one inch of any wires going to or from the Megasquirt-3.

5.2 RPM-Sync issues

Certain conditions such as electrical noise can cause the Megasquirt-3 to lose sync with the engine position. This is called a sync loss event or a sync error. The "RPM-sync" indicator on the dash will briefly turn red. Common symptoms of this include:

- Feels like a misfire or a false rev limiter
- RPM in datalog suddenly drops to zero
- RPM in datalog suddenly spikes up
- "Lost sync count" increments while "SecL" counts up normally. (NB. If SecL suddenly drops to zero this is a "reset"; see section 8.1.)

Taking and reviewing a datalog is a required step for troubleshooting this. This datalog extract below (viewed in Megalogviewer) shows the rpm (yellow trace) dropping to zero and at the same time the "Lost sync count" increases. The "Lost sync reason" shows reason code 2.



5.2.1 Lost sync reasons

The Megasquirt-3 can report several different lost sync reasons in the data logs. Most of these simply show that there’s a problem and don’t particularly pinpoint where it is, but some are more specific than others. In particular, if you get Lost Sync Reason 11 or 17 in a specific RPM band, try changing the ignition input capture and /or second trigger active edge.

0 no problem

2 missing tooth at wrong time

10 too many teeth before end of sequence

11 too few teeth before second trigger

12 too many sync errors

13 dizzy wrong edge

14 trigger return vane size

15 and 16 - EDIS SAW signal error

17 second trigger not found when expected

20 subaru 6/7 tooth 6 error

21 subaru 6/7 tooth 3 error

22 Rover #2 missing tooth error

23 420A long tooth not found

24 420A cam phase wrong

25, 26, and 27 pattern does not otherwise match 420A

28 Pattern does not match 36-1+1

29 36-2-2-2 semi sync failed

30 36-2-2-2 tooth 14 error

31 Miata 99-00 - 2 cams not seen

32 Miata 99-00 - 0 cams seen

33 6G72 - tooth 2 error

34 6G72 - tooth 4 error

35 Pattern does not match Weber-Marelli

36 CAS 4/1 error

37, 38, and 39 Pattern does not match 4G63

40 and 41 Twin trigger error

42 Pattern does not match Chrysler 2.2/2.5

43 Pattern does not match Renix

44 Pattern does not match Suzuki Swift

45 and 46 Pattern does not match Vitara

47 Pattern does not match Daihatsu 3

48 Pattern does not match Daihatsu 4

49 Pattern does not match VTR1000

50 Pattern does not match Rover #3

51 Pattern does not match GM 7X

52 36-2-2-2 tooth 30 error

53 rc51 semi error

54 rc51 re-sync error tooth 6

55 rc51 re-sync error tooth 16

56 rc51 re-sync error tooth 18

61 NGC crank pattern

62 NGC crank pattern

63 QR25DE

68 LS1 resync failed

69 YZF1000 resync failed

70 36-1+1 no cam

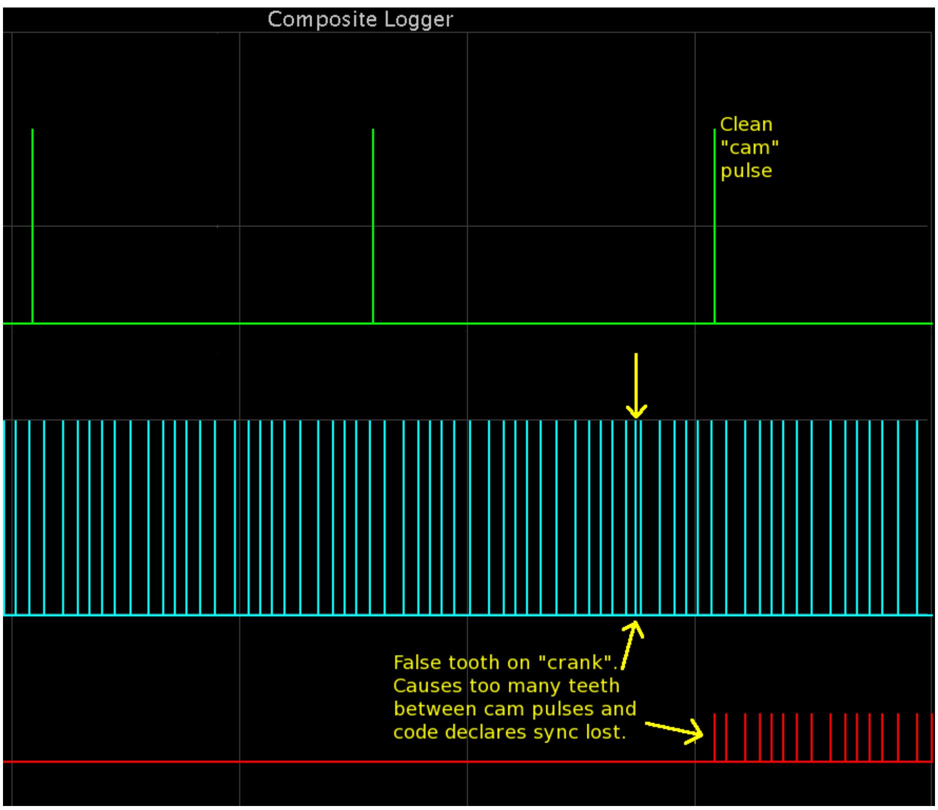
Having identified that you have a sync loss problem, you need to attempt to eliminate it. Check for any basic problems:

- Loose sensor
- Loose cabling

- Sensor wires running close to spark plug wires or coils
- Slop in timing belt
- VR sensor wired backwards
- Fouled plugs causing misfire
- Non-resistor plugs
- Excessively rich mixture causing misfire
- Excessively lean mixture causing misfire

Having ruled out all those possible causes, you should use the tooth and composite loggers to record the incoming crank/cam pattern and look for the problem. You may well want to enable some noise filtering. See the TunerStudio Reference Guide for additional information on the high speed loggers and noise filtering settings.

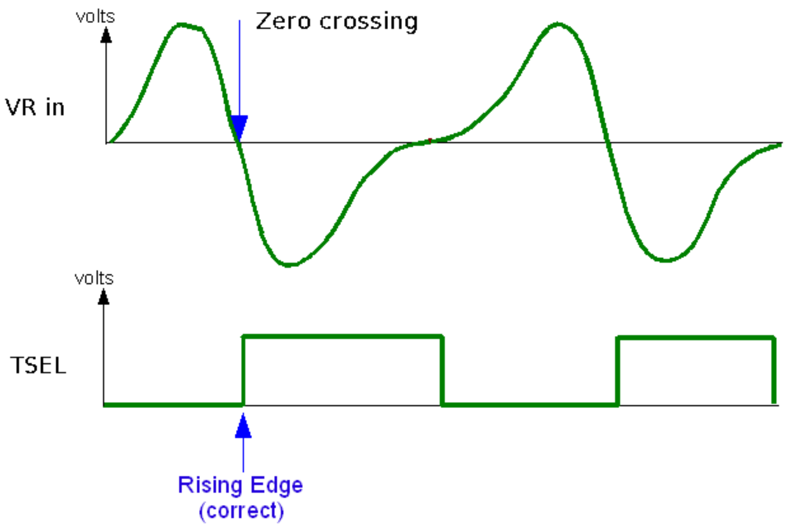
Example composite log showing noise causing loss of RPM-sync.



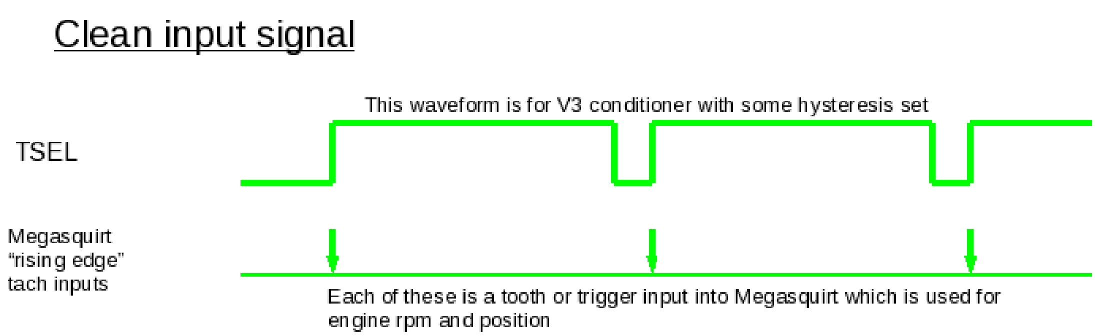
5.3 Noise Filtering

The noise filter settings can be used to "clean up" the incoming tach signal to remove interference and defend against RPM-sync issues.

The exact settings are very install specific, we'll cover some of the background theory of what the settings do. The naming of the inputs may refer to different hardware versions, but the concepts remain the same.

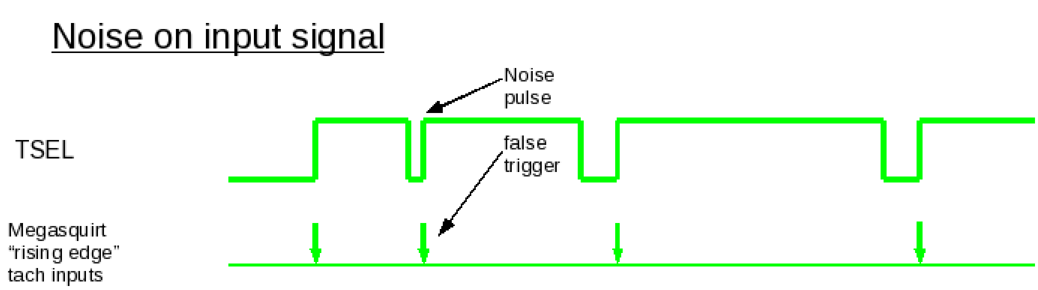


The Megasquirt needs to be configured for rising edge in this instance as that matches the 'sharp' well defined zero crossing of the input signal. We can now look at what the Megasquirt code will see.



This shows how the rising edge configuration results in what we'll call a 'single edged' signal. The Megasquirt is only aware of that rising edge and pays no attention to the length or the high or low periods or the falling edge.

Now, insert some noise.



The noise pulse creates a false trigger to the Megasquirt, this could cause an rpm blip on the fuel only install, a misfire or a total loss of sync on the more advanced installs.

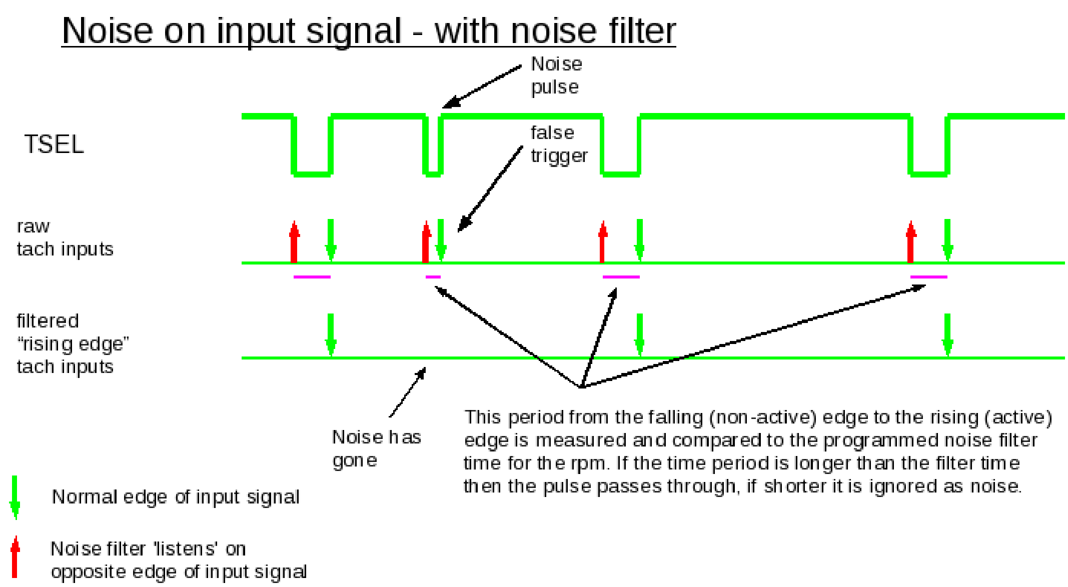
Having established what the noise signal might look like, how to get rid of it?

These are the main methods available to improve situations

- noise filter
- tach masking
- tach period rejection
- polarity check

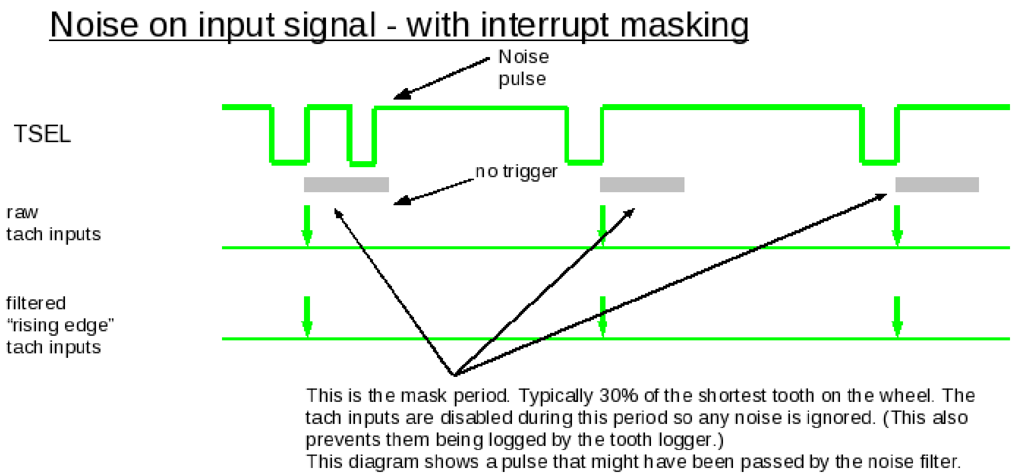
5.3.1 Noise filter

The noise filter works on the principle that the noise is likely to be of short duration, far shorter than a genuine tooth. If all input conditioners are setup correctly, we should be able to detect this condition by measuring the width of the input pulse. A 'long' pulse is real, a 'short' pulse is noise.



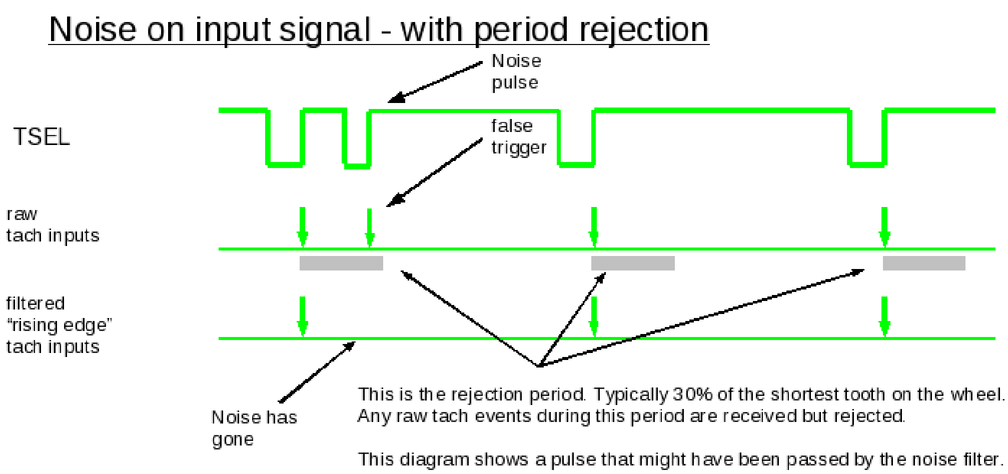
5.3.2 Tach interrupt masking

This approach is slightly different. It does not calculate the pulse width, but looks at where it is within the period. If we have suddenly received an early pulse (within 30% of the expected time) then it must be noise and is rejected. The code disables the hardware interrupt for this calculated mask period and any inputs that happen while the interrupts are off are " invisible." There is a drawback though, on the uneven wheels the code calculates this time based on the smallest tooth. A pattern such as 420A has widely varying tooth sizes, so the benefit is reduced. (Additionally, but less importantly, the tooth logger cannot record the noise inputs as they are not captured.)



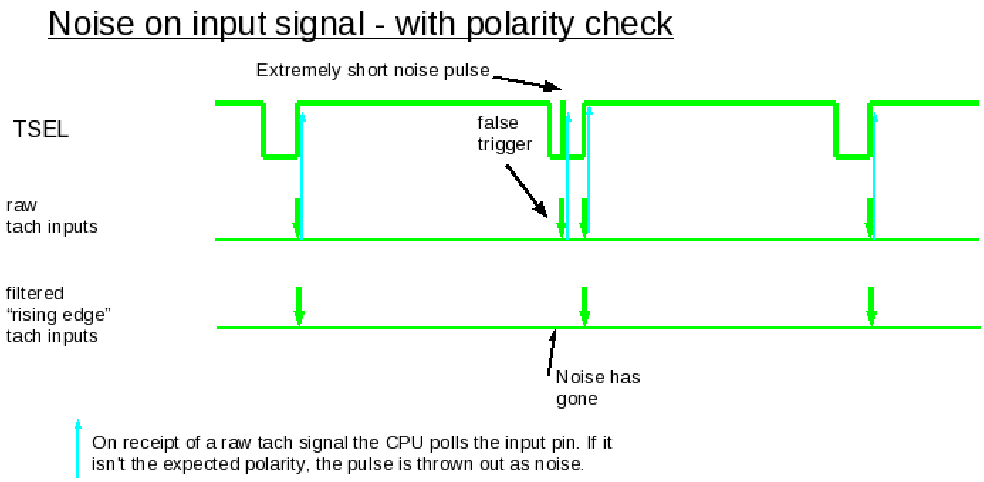
5.3.3 Tach period rejection

This is similar to the concept behind interrupt masking, but the interrupt is still enabled. The code compares the times and rejects it if it falls within the rejection period.



5.3.4 Polarity Check

This method should weed out extremely short noise pulses. If the input signal changes state very quickly, the pulse will already have completed before the Megasquirt has had time to react. By polling the input pin to the CPU the code checks to see if the polarity is the expected one, if not it must be noise and the input is ignored. This method is automatically enabled.

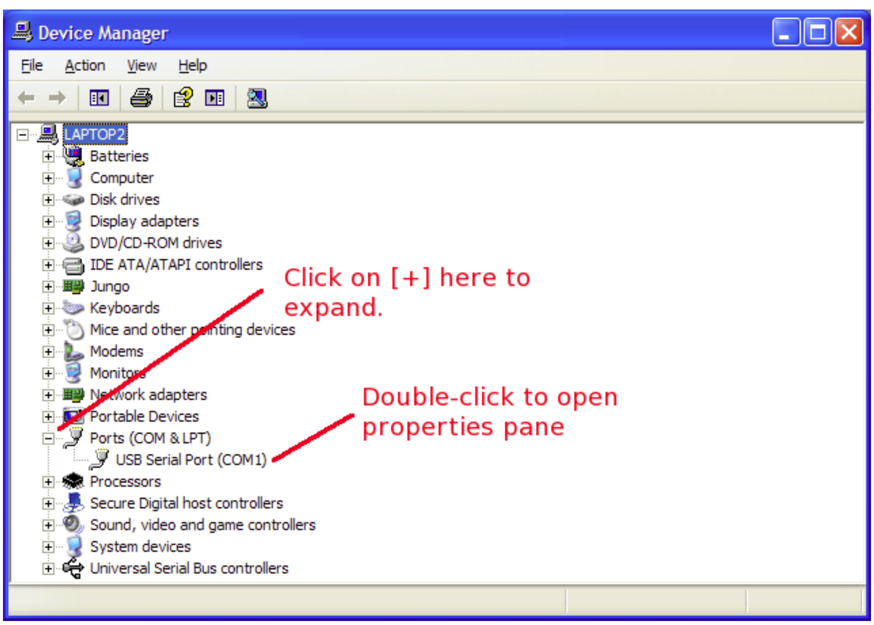


5.4 Communications issues

Sometimes on a new install, you may have difficulties establishing communication with the Megasquirt. The first step is to ensure that your computer is "seeing" the serial interface or adapter.

5.4.1 Windows device

On Windows systems, Device Manager will show comms ports under "Ports".



If nothing is showing under "Ports" then you likely have a configuration or driver problem on your computer. You need fix that first. Make sure the USB cable is securely inserted or try a different socket. If that doesn't resolve it, try de-installing and then re-installing the device driver.

5.4.2 Mac OSX device

USB-serial adapters should show up as files such as /dev/tty.usbserialXX

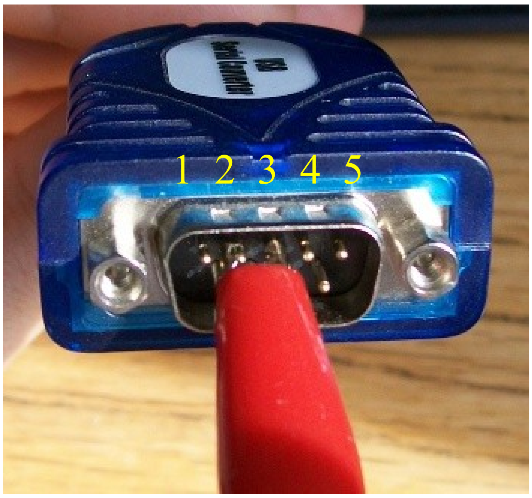
5.4.3 Linux device

USB-serial adapters should show up as files such as /dev/ttyUSB0

5.4.4 Loopback test for serial or USB-serial cable

With a direct serial or a USB-serial cable (not direct USB), you can perform a loopback test to see if serial data is getting out from your computer and can be "looped-back" in again.

Carefully short together pins 2 & 3 of the plug in your cable.



Start TunerStudio and open Communications -> Miniterminal.

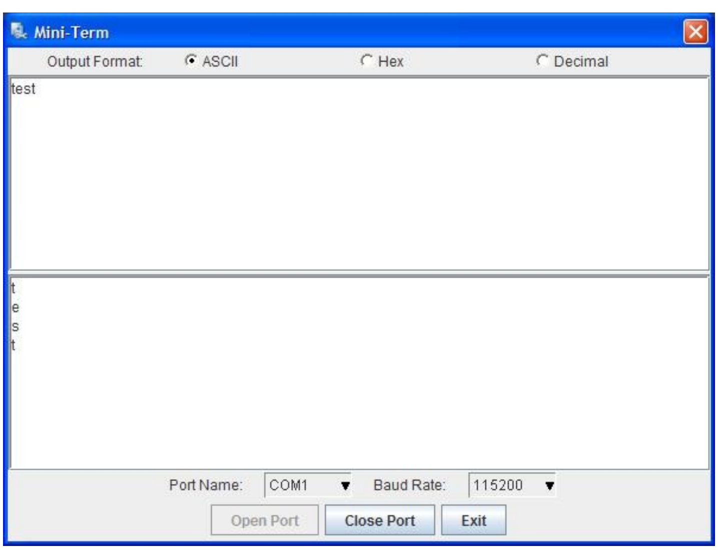
Select the port as above (i.e. if your connection was COM5 in device manager, then choose COM5 here.)

Select the baud rate to 115200

Click to Open the port.

In the upper pane, type in 'test'

If the serial port is working correctly, you should see 'test' come back in the lower pane.



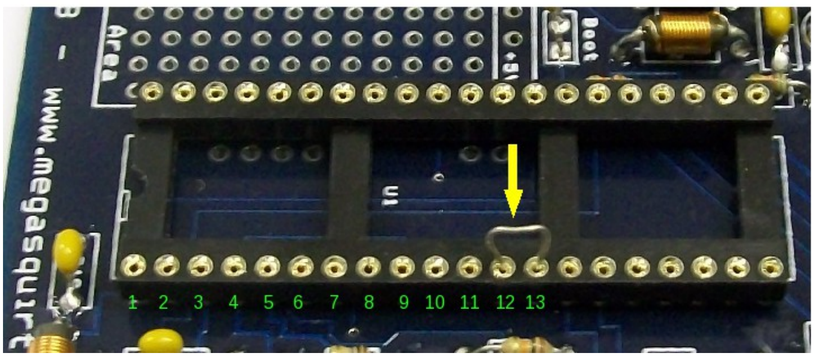
If nothing comes back in the lower pane, double check that pins 2&3 are shorted together. If they are, then your USB-serial adapter or serial port may be faulty. Try on a different computer. Or try a different USB-serial adapter (FTDI based are recommended.)

5.4.5 Loopback test to Megasquirt board

With a direct serial or a USB-serial cable (not direct USB), you can perform a loopback test to see if serial data is getting out from your computer and through the Megasquirt board and can be "looped-back" in again.

Very carefully, prise out the MS3 CPU card U1. Undo the retaining screws and then lift it out progressively by prying up each end a fraction at a time. Be certain not to bend the pins.

Install a jumper betweens pins 12 & 13 of U1



Make sure there are no short circuits or loose wires around, then turn on your Megasquirt (connect a fused 12V supply.)

As in section 5.4.3, open up TunerStudio and perform the loopback test in MiniTerminal.

If test 5.4.3 worked ok, but 5.4.4 does not work, it suggests there may be a problem on the Megasquirt board. Either there is no power, a solder connection is faulty or the serial chip (U6) is damaged.

Carefully re-install the MS3 CPU card U1 after testing is complete.

5.4.6 Comms test to Megasquirt board

This test requires that firmware is already installed (section 2.8). If you are unable to install firmware, then review the previous sections.

Start TunerStudio and open Communications -> Miniterminal. Ensure your Megasquirt is powered and connected to your computer.

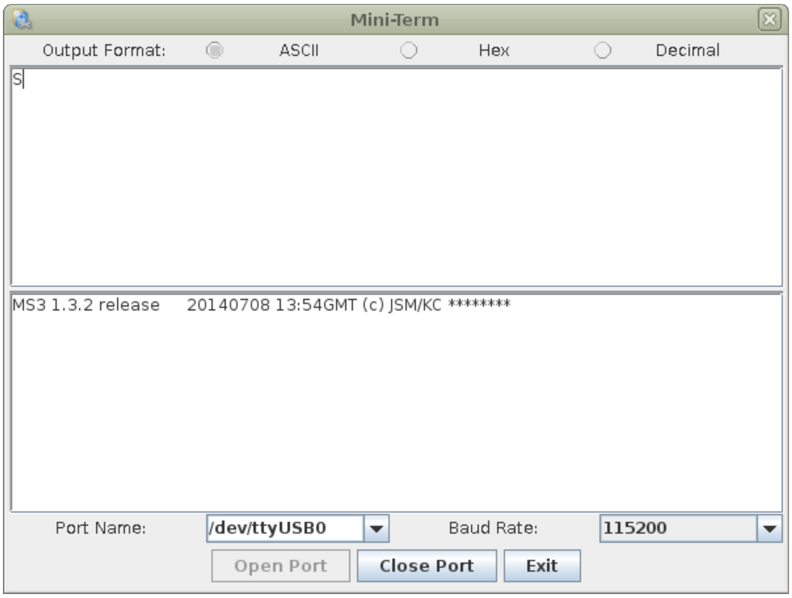
Select the port as above (i.e. if your connection was COM5 in device manager, then choose COM5 here.)

Select the baud rate 115200

Click to Open the port.

In the upper pane, type a capital 'S' and you should see a signature string returned in the lower window as shown here. If you do not then there is a serial problem somewhere.

(The text returned here is the firmware version string, the same as will normally be shown in the TunerStudio title bar.)


Mini-Term
Output Format:
ASCII
Hex
Decimal
S
MS3 1.3.2 release 20140708 13:54GMT (c) JSM/KC ********
Port Name:
/dev/ttyUSB0
Baud Rate:
115200
Open Port
Close Port
Exit

6: Appendix A: Firmware upgrade notes

This section of the manual covers any specific issues you may encounter upgrading to the current firmware version from older releases.

New users do not need to read this section.

Between each major firmware version there will likely be significant changes, some may require an element of re-tuning. Releases within each firmware version (e.g. 1.4.0, 1.4.1, 1.4.2) are released to resolve any issues that become known and should be useable without any settings changes or re-tuning.

6.1 Firmware versions

MS3 firmwares	Status	Comment
1.0.x	Obsolete	Old release code
1.1.x	Obsolete	Old release code
1.2.x	Obsolete	Old release code
1.3.x	Superceded	Old release code
1.4.x	Current	The current release code family
pre-1.5	Development	Development codes leading to 1.5.x

6.2 Alpha, beta, release code? What's the difference?

During code development there are a number of key phases as the firmware evolves. The naming is intended to identify which stage the firmware is at.

- Release - the firmware is considered suitable for general release, features are complete and believed to work as described. **New users should be using release code.**
- Beta - the firmware is considered suitable for moderate release, features have been completed, but extended testing is required to ensure everything works as described.
- Alpha - anything goes. New features, incomplete code, test code. The latest features will be made available as an alpha release, but it is up to the tester to determine if it works correctly. New users are advised to avoid alpha releases.

6.3 Upgrading from 1.3.x

You can mostly load your 1.3.x tune into 1.4.x

Key changes and Gotchas - from 1.3.x

1. Baro

Previously the code used to divide by the barometer when calculating the fuel pulsewidth. If using an old tune you need to enable the "old style" baro calculation to enable the old behaviour.

2. CLT rev limiter

The TPS bypass rev limit setting has been removed, the standard hard limit is now used in bypass mode to reduce confusion.

3. Closed-loop boost

The initial values table is now a bias table. This needs to be tuned as an open-loop table using "setup mode" before enabling full closed-loop. See the tooltips and manual.

4. On/Off outputs using bitwise AND

The meaning of threshold and hysteresis has changed, see the TunerStudio Reference manual for the settings. An example is given in this manual.

5. A list of other settings that have changed is provided in the TunerStudio Reference manual.

6.4 Upgrading from 1.2.x

You can mostly load your 1.2.x tune into 1.3.x

Key changes and Gotchas - from 1.2.x

1. Idle control settings re-arranged

   Need to reset:

      Stepper vs. PWM idle valve

      Open-loop vs. Closed-loop

      Output pin for PWM.

2. Closed-loop idle control

The settings are re-arranged to simplify setup, but existing users will need to retune.

3. GM/TFI ignition settings changed.

4. MAF configuration changed.

5. VSS input configuration changed.

6. Generic PWM output type and frequency selection changed.

7. On/Off outputs enabling changed.

8. Various tables now use "kPa" instead of "%", the numbers should be ok.

9. VVT adds min/max duty settings, you may want to use them.

10. Two TunerStudio dash indicators have been renamed:

   Not Synced -> Not RPM Synced

   Half-sync -> Half-RPM sync

You will need to re-load your dash (right-click on a blank area of the dash select Load/Save, then Load dashboard, then Accept) or update manually by right-clicking on the indicator and picking the new name.

11. Rev limiting is changed. Check your settings.

12. Some "trigger wheel" users may need to change their tooth#1 angle if it

was close to 360deg or 720deg. (A previous bug was fixed.)

6.5 Upgrading from 1.1.x or 1.0.x

You can mostly load your 1.0.x or 1.1.x tune into 1.3.x

Key changes and Gotchas - from 1.1.x and earlier

1. MAF

There is a new MAF calibration implementation. The old method is still valid.

2. Baro

The default settings for baro were on the MAP sensor calibration page as 147, -47 with a tweak curve on top. The new method sets these two numbers to zero and exclusively uses the adjustment curve. 100% means un-altered fuelling.

3. Air-density

In previous versions, there was an internal calculation for air density with a tweak curve on top. The new method exclusively uses the adjustment curve. 100% means un-altered fuelling.

Note: MAT does change air density and the speed-density equation relies on this to estimate intake charge.

4. Spark output polarity

The name has been simplified

"Going High" - was called "Going High (Inverted)"

"Going Low" - was called "Going Low (Normal)"

DOUBLE CHECK YOUR SETTING BEFORE CONNECTING COILS.

5. Boost Control

To make things more intuitive we have now changed the firmware so that

- "Normal" is the most typical output polarity setting

- larger boost duty% numbers mean more boost.

If you are a new user starting out from scratch, then nothing needs to be changed, just use the default settings.

If you are upgrading from a previous firmware version, then the Boost Output polarity setting will be the opposite. For most users it will now be "Normal". This is true whether using open- or closed-loop boost control.

If you're using closed-loop, nothing else should need to change.

Open loop boost users most likely need to set their table so that:

new_cell_value = 100 - old_cell_value

Remembering that more duty = more boost.

6. Serial protocol

The firmware now uses the "newserial" protocol. This requires compatible tuning software and firmware loaders. (Note, it you want to revert to an older firmware version you will need to use the new firmware loader, or use the boot jumper.)

6.6 Upgrading from MS2/Extra 3.0.x or 3.1.x

Largely the same as for 2.1.0 code.
BUT - if you are using 4 injector outputs with a modified MS2 card **you will need to re-wire** to use the new dedicated MS3 output pins.

6.7 Upgrading from MS2/Extra 2.1.0

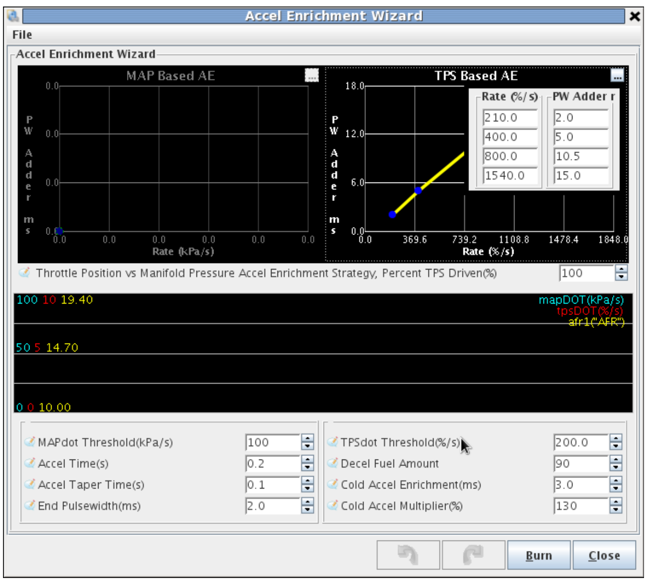
You can mostly load your MS2/Extra 2.1.0 tune into MS3 1.3.x

Key differences

Sequential fuel supported. Accel enrichment TPSdot values 10x larger.

Be sure to alter these values in the Acceleration Wizard. Open up the values table for the TPS based AE curve and multiply all values in the rate% column by 10. Find TPSdot threshold and multiply that by 10.

These are the default settings:



The PW adder values will need to be tuned to suit your injector size and engine behaviour.

See also the notes in section A.3

6.8 Upgrading from MS2/BG

Do not try to load your old MS2/BG (V2.890 etc.) settings.

If your fuel and spark table are well tuned, then retain them. Otherwise start from scratch as if a new user.

Follow the steps in A.7 to export and import your tables.

Key differences

Ignition mode selection is direct, chose your tach input source.

Check your spark output! JS10 is standard on MS2/BG, this will be different on MS3.

The regular wheel decoder requires a single "Tooth #1 angle" instead of skip teeth etc. It automatically calculates the triggers itself and enables the required number of spark outputs depending on how many cylinders you have and your selection of single coil, wasted spark, coil-on-plug or dual-dizzy.

PWM idle valve duties work so that 0% means fully closed.

Stepper idle valves work so that 0 steps is fully closed.

An alternate wall-wetting transient method "EAE" is available.

Rev limiting is changed

Larger fuel and spark tables.

Many new features.

Note that if your Megasquirt is already wired up and working with MS2/BG code then you should not need to change any of that wiring when moving to MS3 - carefully review your settings for input capture and spark output polarity and bring them across.

6.9 Upgrading from MS1/Extra

Do not try to load your old MS1/Extra (also referred to as Megasquirt'n'spark or MSnEDIS) settings into Megasquirt-3.

If your fuel and spark table are well tuned, then retain them. Otherwise start from scratch as if a new user.

Follow the steps above in A.7 to export and import your tables.

Key differences

Almost all other settings are somewhat different between MS1/Extra and Megasquirt-3.

The wheel decoder now requires a single "Tooth #1 angle" and automatically calculates the triggers itself and enables the required number of spark outputs depending on how many cylinders you have and your selection of single coil, wasted spark, coil-on-plug or dual-dizzy.

Spark A,B,C are available on the LEDs (not FIDLE)

Sparks D,E,F have moved

Launch pins are different

Nitrous pins are different and the wiring method has changed

3 stage shift lights not supported

Spare ADCs the same.

12V power feed to the chip to enable the stepper driver (Caution! Do not refit the Megasquirt-1 CPU with the 12V feed in place.)

Whether you imported a VE table or not, you now must go through all settings to configure for your engine. e.g. number of cylinders, ReqFuel, ignition triggering, spark and fuel hardware in use.

6.10 Upgrading from MS1 (base)

Do not try to load your old Megasquirt-1 settings into Megasquirt-3.

If your fuel VE table is well tuned, then retain it. Otherwise start from scratch (online) as if a new user.

To transfer your VE table

Open up your tuning software and load your Megasquirt-1 tune.

Open the VE table. Export as VEX.

Close tuning software.

If you are swapping the Megasquirt-3 card in place of the Megasquirt-1 processor you may well want to add the 12V feed to the chip. (Notes to be added.)

Setup Tunerstudio and create a new project for the Megasquirt 2.

Power up your Megasquirt 2 and open the tuning software.

Ensure that you see the default settings i.e. it isn't all blanks or zeros.

Open the Fuel VE table.

Import the VEX you previously saved.

Whether you imported a VE table or not, you now must go through all settings to configure for your engine. e.g. number of cylinders, ReqFuel, ignition triggering, spark and fuel hardware in use.

7: Appendix B: Fuel calculations

There are a number of different methods to estimate the airflow into an engine. This page will explain how the fuel is mathematically calculated using each method.

7.1 Terms Used

See also the Glossary for general EFI related abbreviations.

PW - base fuel pulsewidth before per-cylinder trims etc.

DT - injector dead-time (opening time less closing time)

ReqFuel - the global fuel constant

VE - the 'VE' table. Often expressed as VE[RPM,MAP] meaning that this lookup table is based on RPM and MAP in the axes.

AirDen - air density correction (centred on 100%) calculated from the MAT sensor reading.

BaroCor - barometric correction (centred on 100%) calculated from the initial MAP reading or dedicated sensor corrections - other adjustments such as warmup enrichment, closed loop EGO, acceleration

7.2 Speed Density

The primary load input is the MAP sensor.

With 'Multiply MAP', without 'Include AFR target'

PW = DT + (ReqFuel * MAP * VE[RPM,MAP] * AirDen * BaroCor * corrections)

With 'Multiply MAP', with 'Include AFR target'

PW = DT + (ReqFuel * MAP * Stoich/AFRtarget * VE[RPM,MAP] * AirDen * BaroCor * corrections)

7.3 Alpha-N (pure)

The primary load input is the throttle, this is often a poor indicator.

Without 'Multiply MAP', without 'Include AFR target'

PW = DT + (ReqFuel * VE[RPM,TPS] * AirDen * BaroCor * corrections)

Without 'Multiply MAP', with 'Include AFR target'

PW = DT + (ReqFuel * Stoich/AFRtarget * VE[RPM,TPS] * AirDen * BaroCor * corrections)

7.4 Alpha-N (hybrid)

The primary load input is the throttle, but MAP is also included in the calculation.

With 'Multiply MAP', without 'Include AFR target'

PW = DT + (ReqFuel * MAP * VE[RPM,TPS] * AirDen * BaroCor * corrections)

With 'Multiply MAP', with 'Include AFR target'

PW = DT + (ReqFuel * MAP * Stoich/AFRtarget * VE[RPM,TPS] * AirDen * BaroCor * corrections)

7.5 %baro

The primary load input is the MAP sensor. The VE table lookup uses %baro = MAP/Baro.

With 'Multiply MAP', without 'Include AFR target'

PW = DT + (ReqFuel * MAP * VE[RPM,%baro] * AirDen * BaroCor * corrections)

With 'Multiply MAP', with 'Include AFR target'

PW = DT + (ReqFuel * MAP * Stoich/AFRtarget * VE[RPM,%baro] * AirDen * BaroCor * corrections)

7.6 MAF

The primary load input is the MAF sensor.

With 'Include AFR target'

PW = DT + (ReqFuel * Stoich/AFRtarget * (MAFload * AirDen) * corrections)

This is somewhat different to the other algorithms. As the MAF sensor is actually measuring airflow, the value (ideally) includes the effect of VE, air density and baro.

There are a number of calculations to arrive at this point, the casual reader can skip these. For the full gory details, see the source code. As the processor uses integer maths, many values are scaled *10, *100 etc to preserve precision.

First the raw MAF reading is established from the flow calibration curve.

MAF = flowcurve[MAF volts] in 0.01g/s units

Then..

MAFCoef = 1010048999 / engine_size_cc

Then the flow-load value MAFload

MAFload = (MAFCoef / AirDen) * MAF) / RPM MAFload in 0.1 units, AirDen 1000 = 100%, RPM in 1 units.)

For code efficiency and to improve precision, instead of dividing by AirDen and then multiplying it back out again, the code actually calculates a version of MAFload without the air density correction.

MAFload_no_air = (MAFCoef / 1000) * MAF) / RPM

So the equation becomes:

PW = DT + (ReqFuel * Stoich/AFRtarget * MAFload_no_air * corrections)

Comparing this to Speed Density, MAFload_no_air takes the place of (MAP * VE[RPM,MAP] * AirDen * BaroCor)

8: Appendix C: Megasquirt Glossary of Terms

ADC – stands for "analog-to-digital converter". In this case, it is part of the conversion circuitry in the CPU that translates the varying voltage signal to a digital value that the CPU can understand and operate on. All of the sensors (TPS, MAP, CLT, MAT) send their signal to a particular ADC pin on the processor. The ADC result is used by the processor as a 'count'. MS-I has an eight bit ADC, so the counts can be from 0 to 255. MS2 and MS3 have a more precise 10-bit ADC, so the count can be from 0-1023. Both of these are mapped over a 0-5 Volt range (so the voltage into the ADC = ADC count * 5.0/1023 for MS2, for example); however, higher external voltage may be brought down to a 0-5 volt range with a voltage divider or boosted with an amplifier (such as for EGTs).

AE – Acceleration Enrichment, the enriched mixture provided when the throttle position sensor signal (TPSdot) or map sensor signal (MAPdot) changes at various rates.

AFM – Air Flow meter. Often a vane or flapper type device used to measure air-flow into an engine.

AFR – Air Fuel Ratio, the mass ratio of air to fuel in the combustion chamber. See lambda, NB- and WB-EGO sensors, below.

Alpha-N – Fuelling algorithm that uses throttle position as the primary load. Should not be used on turbocharged engines.

ASE – After Start Enrichment, the enriched mixture provided for a number of engine cycles when MegaSquirt detects that the engine has transitioned from cranking to running.

AMC – Automatic Mixture Control, a control system available in MS2/BG code to automatically tune fuel within the ECU using feedback from the oxygen sensor without use of a laptop. Should be used with great caution as a defective O2 sensor could cause a good VE table to be wiped out.

ATDC – After Top Dead Center, the crankshaft position with respect to the piston being at the top of its travel, meaning it has passed it's highest position and is descending.

Baro – Barometer - the ambient air pressure. At sea level this is around 100kPa. At high elevations it may be 80kPa. The fuelling needs to take account of this.

%baro – a fuelling algorithm where engine load = MAP/Baro and the speed-density system is used. This alters what values are looked up in the fuel VE table.

Barometric correction – a calculation that alters fuelling based on the ambient air pressure. Important at varying elevations. Of little importance in low-lying countries.

BIP373 – a robust ignition driver transistor from Bosch. Features over-currrent and over-temperature protection. Used in all good Megasquirt kits. Beware of lesser alternatives.

BTDC – Before Top Dead Center, the crankshaft position with respect to the piston being at the top of its travel, meaning it has NOT passed it's highest position and is rising. Most normal spark event occur BTDC.

CAN – (Controller Area Network) - a dedicated automotive networking system to allow different automotive processors to communicate and share inputs and calculated results. Used as a convenient way to extend the inputs and outputs of the Megasquirt ECU. (Not available on Megasquirt-1)

Carbon Monoxide (CO) – Poisonous gas produced during combustion process. In an automotive context, generally refers to regulated carbon monoxide (CO) tail pipe emissions.

CAS – Crank Angle Sensor. Often used to refer to combined crank and cam position sensors mounted where historically a distributor would have been.

Catalytic converter – a chemical device in the vehicle's exhaust system that can reduce the amount of regulated emissions emitted by converting Nox to N2 + 02, CO to CO2 and HC to CO2 + H20. Catalytic means the converter active substrate facilitates the reactions, but is not consumed (and thus has a long life with no replenishment requirements).

CID – Cubic Inch Displacement. The imperial measure of the swept volume of the pistons. 61.02 cubic inches = 1 litre.

Closed loop – refers to those times when an EFI computer is using feedback from a sensor to alter outputs

Closed loop boost – Megasquirt uses a PID to keep boost in line with a boost target table

Closed loop EGO – Megasquirt controls the EGO to the target table by varying the injected fuel amounts.

Closed loop idle – Megasquirt controls the idle rpm by opening or closing the idle valve as required.

CHT – Cylinder Head Temperature, used instead of coolant temperature (CLT) on air-cooled engines.

CKP – Crankshaft position sensor

CLT – Coolant Temperature sensor (aka. CTS). Usually the CLT sensor is an NTC (Negative Temperature Coefficient) thermistor, or a resistor whose resistance varies with temperature (NTC means the resistance goes down as the temperature goes up).

CMP – Camshaft position sensor

CNP – Coil Near Plug – See Coil on Plug

Coil On Plug – One ignition coil is installed per spark plug. Either directly on top of the plug e.g., a 'pencil' coil. Or in close proximity with a short plug-lead (actually coil near plug.) This setup can give maximum spark energy and can be the neatest install with least lost spark energy.

COP – Coil on Plug – See Coil on Plug

CPS – Possibly ambiguous Crankshaft or camshaft position sensor

CPU – Central Processing Unit, aka. "processor" or "microprocessor" the computational engine that performs the calculations to operate the injection and ignition functions in MegaSquirt®. It has a number of support circuits, like the power circuit, the clock circuit, the serial and CAN communications circuits, and various input and output conditioning circuits.

CTS – Coolant Temperature Sensor (aka. CLT). Usually the CTS is an NTC (Negative Temperature Coefficient) thermistor, or a resistor whose resistance varies with temperature (NTC means the resistance goes down as the temperature goes up.

Diode – A two lead electrical device that allows current to flow in only one direction (see also Zener diodes).

DIY – Do-It-Yourself.

DMM – (digital multi meter) electronic current/resistance/potential measuring tool. (Required tool for any Megasquirt install.) May also be referred to as DVM (digital volt meter.)

Dual table – A firmware feature allowing different injector outputs to have independent fuel tables. Can be used as an awkward method to achieve per-bank trim. More commonly used (with external relays) to map for regular fuel and LPG.

Duty Cycle – (DC)– A number indicating the amount of time that some signal is at full power. In the context of MegaSquirt EFI Controller, duty cycle is used to describe the amount of time that the injectors are on, and to describe the “hold” part of the peak and hold injector drivers (see Low Impedance Injectors, below).

ECU – (Electronic Control Unit) is the general term for a fuel injection controller, of which MegaSquirt is an example.

EDIS – Electronic Distributorless Ignition System is Ford's wasted-spark computer-controlled ignition module, which has a simple two wire hookup to MegaSquirt for full mapped ignition control.

EGO Sensor – Exhaust Gas Oxygen sensor, used to describe the sensor in the exhaust that measures the lean/rich state of the intake mixture. Used to control the fuel via a feedback algorithm called “closed loop”.

EGR – Exhaust Gas Recirculation. Used by OEMs for emissions purposes. Not supported by Megasquirt.

EGT – (Exhaust Gas Temperature) is the temperature of the exhaust gases, typically measured with a K-type thermocouple. Megasquirt does not have on-board hardware for a direct connection. An add-on board is required.

FET – (field effect transistor) - In MegaSquirt EFI Controller, the transistors used to control the activation of the injectors.

FIdle – Fast Idle. A device used to control idle speed with additional air supplied by a vacuum solenoid. More commonly a fully variable PWM idle valve is used instead. This term is most frequently applied to a connection from the Megasquirt ECU that can be used to drive an on/off valve or a PWM valve.

Gamma – Used to indicate the change in a fuel amount from the calculated amount.

GammaE – A collection of all enrichments and minor corrections to fuel (see speed density.) Now renamed as 'totalcor'.

Gear tooth sensor – an "active", tooth presence sensor. It is based on the Hall effect. The Hall effect sensor consists of semiconductor material which will conduct current when the material is subject to a magnetic field. These types of sensors include a magnet within the sensor and will detect a steel tooth in close proximity. No external magnets are required. Very simple to use.

Gego – Gego is short for 'Gamma – Exhaust Gas Oxygen'. It is the change applied to the fuelling equation based on the EGO O2 sensor feedback. This feedback can (and is) done in a number of different ways, depending on the type of sensor, etc. The important thing is that this is a result of an external measurement, not a pure calculation, so it appears a bit mysteriously in the equation. Now renamed as 'egocor'.

Hall sensor – an "active", magnetic field presence sensor. It is based on the Hall effect. The Hall effect is the change of resistance in a semiconductor in a magnetic field. The Hall effect sensor consists of semiconductor material which will conduct current when the material is subject to a magnetic field. These types of sensors require a "flying magnet", wheel. Instead of teeth on the wheel, as in a variable reluctor sensor, you must have small magnet and a shutter wheel.

Hydrocarbon (HC): in an automotive context, generally refers to regulated unburned hydrocarbon tail pipe emissions.

HEI – , the distributor based electronically controlled ignition system from General Motors. There are a number of variants, identified by the module they use:

- 4-Pin module: non-computer controlled electronic ignition,
- 7-Pin module: computer controlled electronic ignition used in 'large cap' distributors,
- 8-Pin module: computer controlled electronic ignition used in 'small cap' distributors.

High Impedance Injectors – (a.k.a. hi-Z or high ohm) Fuel injectors designed to work with a simple switch in a 12 volt circuit, no special signal conditioning is required to drive them. The resistance of a high impedance injector is about 10-15 ohms.

Hz – (Hertz) the measurement of the frequency of a cyclical event, it represent the number of times per second the cycle is completed.

IAC – Idle Air Controller, though this term is sometimes used more generally, it usually refers to GM's 4 wire stepper motor controller for additional idle air (and hence engine speed) during warm-up and for idle regulation.

IAT sensor – Intake Air Temperature sensor, same as MAT, see below.

IGBT – Insulated Gate Bipolar Transistor a particular kind of transistor especially suitable for driving ignition coils. e.g., VB921, BIP373

Incorporate AFR – An MS2 and MS3 feature where the AFR target table is included in the fuelling equation. The Fuel VE table then becomes far closer to a true VE table. Enrichment or enleanment from stoichiometric comes from the AFR table. Ideally this setting should be turned on before any tuning and the AFR table set to your target AFRs. Then the VE table is tuned to achieve that AFR. Turning the setting on/off will require a complete re-tune.

Interpolate – The dictionary meaning is to insert an intermediate value into a series by estimating or calculating it from surrounding known values. On all tables and curves, when you are between points, the Megasquirt interpolates the intermediate values. Some older other brands were unable to do this and required monstrously large tables to overcome this limitation.

kPa (kiloPascals) – the measurement of air pressure used in MegaSquirt® computations. It ranges from 0 (vacuum) to 101.3 kPa (standard atmospheric pressure at sea level) to 250 kPa (21psi of boost) or higher.

Lambda – an alternative term to AFR where 1.0 is stoichiometric. Lambda numbering is fuel independent. 1.0 is always stoich.

Low Impedance Injectors – (a.k.a low-Z, low ohm) Fuel injectors that are designed to run at a much lower current than would be supplied by a direct 12 volt connection. They require a special signal that is initially at full current (4-6 amps, a.k.a. “peak current”) for about 1.0-1.5 ms, but then drops down to about 1 amp (“hold current”) for the rest of the opening pulse. The resistance of a low-impedance injector is typically 1-3 ohms. These may need a "peak and hold" board or a resistor pack.

LSU-4 – Bosch wide-band oxygen sensor.

MAF sensor – Mass Air Flow sensor. A sensor used to measure the mass of air flow through the intake tract (which must be sealed at the senor and downstream).

MAP sensor – Manifold Absolute Pressure sensor. Measure of the absolute pressure in the intake manifold (related to the engine vacuum), to determine the load on the engine and the consequent fueling requirements. The standard MAP sensor in MegaSquirt® is the MPX4250 (2.50 BAR, or 15 psi (vacuum) + 21 psig (boost)). A basis of the "speed density" fuelling algorithm.

MAPdot – rate of change of MAP value. (dMAP/dt) Used mainly to trigger Accel enrichment.

MAT Sensor – Manifold Air Temperature sensor, the same as IAT. The MAT circuit is identical to the CLT circuit, see CLT, above.

.MSL – file extension used by the MegaSquirt for storing datalogs (.XLS might also be used for datalogs with older versions of tuning software).

MPX4250AP – the standard internal MAP sensor used in MegaSquirt.

MS – MegaSquirt, used in this document to refer to the MegaSquirt® fuel injection controller or its embedded software. e.g., MS1 = Megasquirt-1, MS2 = Megasquirt-2, MS3 = Megasquirt-3

MS1/Extra – The most commonly used firmware on the Megasquirt-1 chip. Gives fuel and spark control. Used to be called Megasquirt'n'spark-extra or MSnS-extra. The original basic MS1 code was fuel only.

MS2/Extra – Firmware for the MS2 chip and Microsquirt with multiple spark outputs, many wheel decoders, closed loop idle, boost, nitrous, table switching and more

MS2/BG – More basic firmware for the MS2 chip and Microsquirt from Bowling and Grippo.

.MSQ – file extension used for saving MegaSquirt fuel injection controller user settings (aka. "parameters") on a tuning computer. Since approximately 2005, these have been readable XML format, so you can open these in Notepad or similar text editors. This format allows some exchange of tune data between different code versions.

NB-EGO Sensor – Narrow Band EGO sensor, gives a switch at the stoichiometric ratio (the chemically correct mixture of air and fuel), but unreliable for AFR other than stoichiometric.

Ohm's Law – A fundamental law of electricity, that states that the current flow (I) is equal to the voltage (V) divided by the resistance (R), or:

$I = V/R$

Oxides of Nitrogen (NOx) – in an automotive context, generally refers to regulated 'mono-nitrogen oxides' (NO and NO2) tail pipe emissions.

OEM – (original equipment manufacturer) - refers to parts produced for initial assembly of a new vehicle. (As opposed to after market parts.)

Open Loop – refers to those times when MegaSquirt works entirely from its calculations with no feedback from a sensor (compare Closed Loop.)

Open Loop Boost – there is no feedback from the map sensor – boost duty is controlled by the duty table only.

Open Loop EGO – there is no feedback from the oxygen sensor – fuelling is from the VE table only.

Open Loop Idle – there is no feedback loop with rpm – idle valve position is set only from the table.

PCB – (printed circuit board) – the fiberglass board that has the MegaSquirt component layout and circuits imprinted on it.

PID – (Proportional, Integral, Derivative) - a method of closed-loop feedback control

Pull-up Circuit – A very simple circuit consisting of a voltage supply and a current limiting resistor designed to prevent a signal from floating, it forces the signal to either be high (equal to the pull up voltage) or low (grounded).

P&H Injectors – Peak and hold injectors; see Low Impedance injectors.

PIP – Profile Ignition Pick-up is the term used for the signal sent from Ford's Electronic Distributorless Ignition System (EDIS) to the electronic control unit. The EDIS module decodes the 36-1 wheel and sends 2, 3 or 4 PIP pulses per engine revolution depending on EDIS4,6,8. The PIP signal into the ECU is a square wave switched at 12 volts. It provides information about both the engine speed and position.

PW – (Pulse Width) is the amount of time a signal is applied during each period. For example, is the amount of time (in milliseconds) an injector is pulled low (grounded) to inject fuel.

Pulse Width Modulation (PWM) – A signal with a fixed pulse width (frequency), which is turned on for part of the pulse. The percent of time that the signal is on is called its duty cycle. PWM is used to limit current to low impedance fuel injectors, as well as for solenoids such as idle valves, boost solenoids and VVT

Realtime baro – a dedicated second pressure sensor is used to monitor ambient pressure. Required if your vehicle will go on journeys at varying elevations.

Required Fuel – (Req_Fuel) The injector pulse width, in milliseconds, required to supply the fuel for a single injection event at stoichiometric combustion, 100% volumetric efficiency and standard temperature.

RPM – Revs Per Minute. The term used to describe engine speed.

SAW – Spark Advance Word is the 'returning' signal to a Ford EDIS ignition unit from the ECU that sets the amount of ignition advance requested. It is in the form of a 5 volt square wave.

Semi-sequential – Fuelling system where injection events are timed to each cylinder (like spark is) but occur twice per cycle. Offers some of the benefits of full sequential without needing a cam wheel input.

(Fully) Sequential – Fuelling system where injection events are timed to each cylinder (like spark is) and occur once per cycle. Do not confuse with direct injection. Sequential uses normal injectors of a normal size, but their timing can give slight power and/or economy increases

Stim (MegaStimulator) – the original stimulator is a small board which plugs into the connector of the MegaSquirt® controller. It simulates all the sensor the inputs the MegaSquirt® controller would normally see (but the rpm signal is only suitable for use with distributor or EDIS configurations) and provides power to the MegaSquirt® controller. The Stimulator also allows you to monitor the MegaSquirt® controller's injection pulses [actual], fuel pump relay operation, and fast idle solenoid output with four LEDs. The JimStim is an enhanced version that simulates many wheel patterns and has more LEDs for multiple spark outputs.

Speed Density – Fuel control algorithm that calculates fuelling based primarily on pressure, air temperature and volumetric efficiency.
PW = dead time + (ReqFuel * MAP * VE(rpm,map) * GammaE)
All the corrections are included within GammaE.

Stoichiometric – the chemically ideal air:fuel mixture. Expressed as 1.0 Lambda for all fuels or 14.7 for petroleum/gasoline.

SPOUT – Spark Out is the spark advance signal sent from MS-II™ to the Ford TFI module to set the timing advance.

Table blending – A firmware feature that allows the user to combine two tables, usually using different algorithms. e.g., one table could be a regular speed-density table and the other alpha-n. This could be useful on an individual throttle body install or as a way of extending the table size.

Table switching – A firmware feature that allows the user to swap to a different fuel, spark, etc table. Can be used in conjunction with dual fuels or nitrous or as a way of extending the table size.

TBI – Throttle Body Injection is a form of injection is which the fuel is injected above the throttle(s). It was typically used on older engines since it can be a simpler system, but is also found on some very high output racing engines because the vaporization time is longer than with port injection.

TPI – Tuned Port Injection is General Motors bank-fire port fuel injection system. It was widely used on 305 and 350 cid V8 in the mid to late 1980's.

TPS – Throttle Position Sensor, a voltage divider that gives information to a MegaSquirt® controller about throttle opening, from which it computes rate of throttle opening for acceleration enrichment.

TPSdot – rate of change of TPS value. (dTPS/dt) Used mainly to trigger Accel enrichment.

Transistor – A solid-state (no moving parts) electronic device that uses a small voltage (on the base pin) to control a larger current (across the emitter and collector pins). There are two basic types of transistors, PNP and NPN. A PNP transistor allows current to flow if there is no signal on the base, and restricts the current as the base current increases. An NPN transistor allows current to flow in proportion to the signal on the base, and restricts the current as the base current decreases.

TS (TunerStudio MS) – The standard software for tuning MegaSquirt EFI controllers. (The previous, now obsolete, software was called Megatune.)

Volatile Organic Compounds (VOC) – in an automotive context, generally refers to organic components or precursors to smog formation. Organic compounds are all chemical compounds containing carbon-hydrogen (C-H) bonds of covalent character. The U.S. Environmental Protection Agency defines a VOC as any organic compound that participates in a photoreaction. An example of a VOC is PAN (peroxyacetyl nitrate), CH3COOONO2.

Voltage divider – A simple circuit that uses two resistors in series to reduce the output voltage Vo proportionally to the input voltage Vi. Two resistors, R1 and R2, are arranged in series, with one end connected to the external signal, and the other to ground. The signal is taken from between the two resistors. From Ohm's Law we can calculate that the total current will be I = Vi/(R1+R2). The same current flows in each resistor because they are in series. The voltage drop across each resistor is: V1 = I*R1 and V2 = I*R2 = Vo. V2 is output voltage Vo, which we can write as:

Vo = Vi* R2/(R1+R2)

V2.2 – The first mainstream circuit board used within Megasquirt. While still available for sale, it is largely obsolete.

V3.0 – The current circuit board used within Megasquirt (accurate Nov 2010). This superceded the V2.2 adding better support for low-z injectors and a circuit for VR conditioning.

V3.57 – A surface mount circuit board, largely the same as the V3.0 board.

VB921 – a particular IGBT designed for use with automotive ignition coils. Now superceded by BIP373.

VE – Volumetric Efficiency. The actual amount of air being pumped by the engine as compared to its theoretical maximum. A 2000 cc engine will theoretically move 2000 cc of air in one cycle at 100% efficiency. If the engine is actually running at 75% VE, then it will move 1500 cc of air on each cycle.

Vref – a 5 Volt supply used to power the TPS (and sometimes other external components needing a 5 Volt supply, like ignition modules or pull-ups).

VR sensor - Variable reluctor sensor – is an induction type sensor, it is "passive", i.e., it does not require a power source, and has a small magnet built in. Needs to use a conditioning circuit – as provided on the V3.0 and V3.57 mainboard. Cannot be used directly on the V2.2 mainboard.

Wasted-COP – a method where individual coils are fitted per spark plug, but the coils are fired in a 'wasted' manner twice per cycle. This is used during startup in some wheel modes and could also be used temporarily by a user before full coil-on-plug is installed.

Wasted Spark – A method of firing spark plugs in which one 'double-ended' coil simultaneously fires two coils on different cylinders. One of the cylinders is the intended 'target', and is near TDC on its compression stroke, the other cylinder is offset by 360° in the firing order (of the 720° 4-stroke cycle), so it is near TDC on its exhaust stroke. The second spark is said to be 'wasted' because it does not ignite a mixture. On the other hand, the hot ionized exhaust gases require little energy to create a spark, so nearly all the energy goes to the 'target' cylinder. An example of a wasted spark system is Ford's EDIS. The advantage is that while a missing tooth crank wheel is required, no cam sync signal (and the corresponding wheel and sensor) are needed.

WB-EGO Sensor – Wide Band EGO sensor, can be used to derive real AFR data with mixtures from 10:1 to 20:1, i.e., anything you are likely to be interested in. Allows for lean or rich tuning.

WOT – Wide open throttle.

WUE – Warm Up Enrichment, the enriched mixture applied when the coolant temperature is low.

Zener diode – A diode that allows current to flow in one direction, unless the reverse voltage is higher than its rated avalanche voltage, in which case it will flow in the reverse direction. Often used to limit excessive voltage to certain circuits by shunting it either to ground or the power supply.

9: Revision history

2014-07-22	Revision history started
2014-07-23	Added VVT section 4.9
2014-08-13	Upgrade notes. Added lots more content.
2014-08-14	Re-arrange section 4. Add some more content.
2014-08-29	Fix some typos is SDcard section.
2014-08-30	Add a para to boost section.
2014-09-07	Added note about resistor plugs. Replace missing MS3 pix. Fix RX8 coil info typo.
2014-11-17	Remove spurious references to MS3-Pro.
2015-01-30	Apply edits from 'Laminar'. Idle homing steps setup procedure. Linux group note. Edit glossary.
2015-01-31	Remove rogue CAN section.
2015-02-02	Minor edits.
2015-07-09	Add bitwise AND explanation to on/off outputs.
2016-12-29	Add missing FTDI config and tweak serial setup section.
2017-01-16	Merge in boost dome control information. TODO - TS firmware loader