ProyectECU Manual

This manual covers the hardware (sensors, wiring etc), software configuration and tuning elements related to running a The ECU unit. When beginning with the ECU, particularly if it is your first time installing and configuring an engine management system, this manual will assist in understanding The ECU's capabilities and how it should be installed, both in terms of hardware and software/firmware.

Whilst this document will assist in providing information related to The ECU's configuration, it does not cover advanced engine tuning, fuel/ignition strategies, etc. As with any changes to engine management, the possibility of damage to hardware is very real the configuration and testing should be done properly and carefully.

Video tutorials, with recommendations and some procedures Will be uploaded to The ECU yahoo channel.

Getting Started

In terms of starting out with ProyectECU, it can help to understand the various components that make up the system:

A ProyectECU: This is the muscle and contains all the controllers and IO circuits. This is also the brains and contains the processor, memory, and storage. The ECU connects to interact with vehicle wiring.

Firmware: This is the system software that runs on the Processor board and powers its operation. New firmware is released regularly with updates, performance improvements and bug fixes.

As a starting point, it is generally recommended to connect the ECU to a 12v source and connect it to the tuning software (TunerStudio) before moving on to installation on the vehicle. The software configuration in The ECU can be completed without the need for it to be installed in the vehicle and this allows you to explore the software and available options.

It is encouraged to do a test on desk, with 12v and GND, to connect to Laptop/PC install drivers/software create the Tunerstudio with the custom .ini file and to do a proper Hardware test using the Tunerstudio buttons. This all before going and start an actual installation on a vehicle.

About this Manual

This documentation is growing continually, and this means that you may come across gaps in the documentation where little information is currently provided. Please do not hesitate to post on the Facebook official group or chat directly with the webpage if there is something missing that you need critically (or even not so critically).

Beginning with the ECU

ProyectECU hardware and software can be configured in many ways, so it can initially be daunting to understand the requirements and configuration steps.

The links below will help you get an overview of each area, how it should be configured, and how it relates to your overall setup.

Hardware Requirements: What hardware will you need to work with The ECU (sensors, wiring, injectors, coils, etc.)
ECU Specific Information:
The ECU Spartan
The ECU Titan
The ECU Berserker
The ECU Gladiator

The ECU Mercenary
The ECU Blackbox
The ECU Mini

Working with TunerStudio software

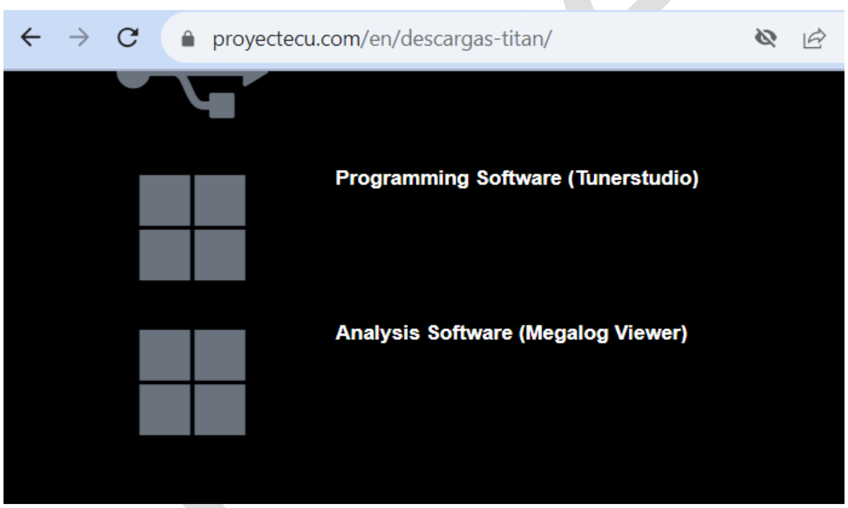
Connecting to TunerStudio

Set up a project in TunerStudio

Downloading Tuner Studio

If you haven't already, grab a copy of Tuner Studio from EFI Analytics Tuner Studio is available for Windows, Mac and linux and will run on most PCs as it's system requirements are fairly low.

For your convenience it can also be downloaded from ProyecECU.com webpage in the download section.



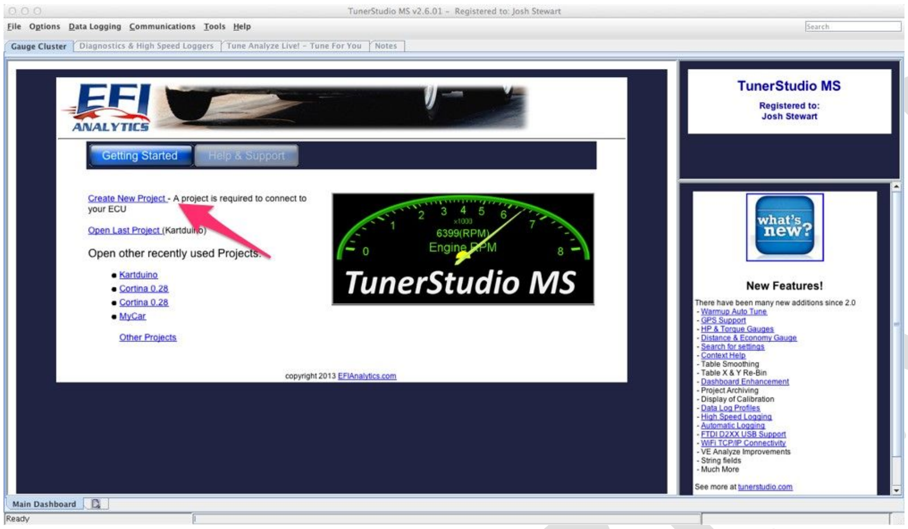
The current minimum version of TunerStudio required is 3.0.7, but the latest version is usually recommended.

If you find Tuner Studio to be useful, please consider paying for a license. This is a fantastic program from a single developer that rivals the best tuning software in the world, it's worth the money.

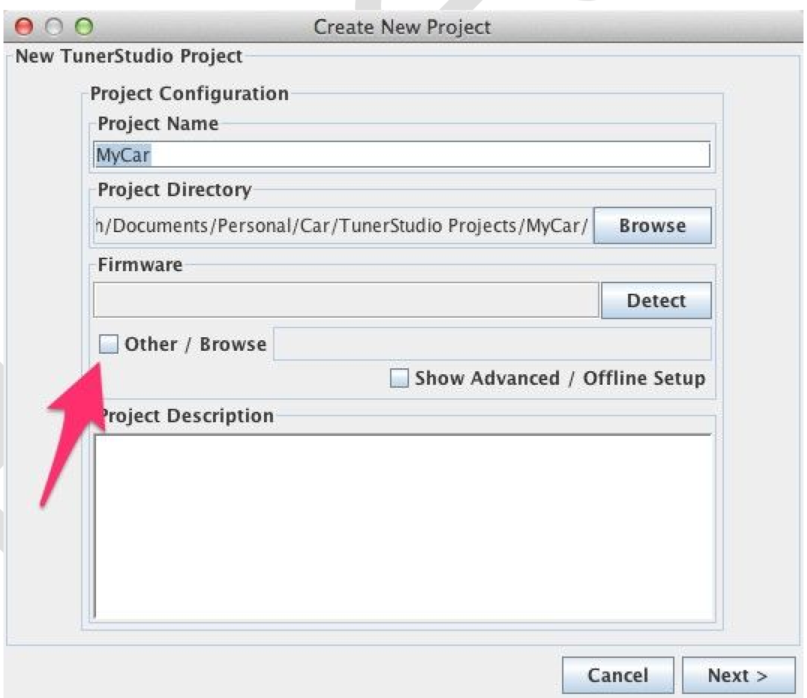
Setting up your project

Create new project

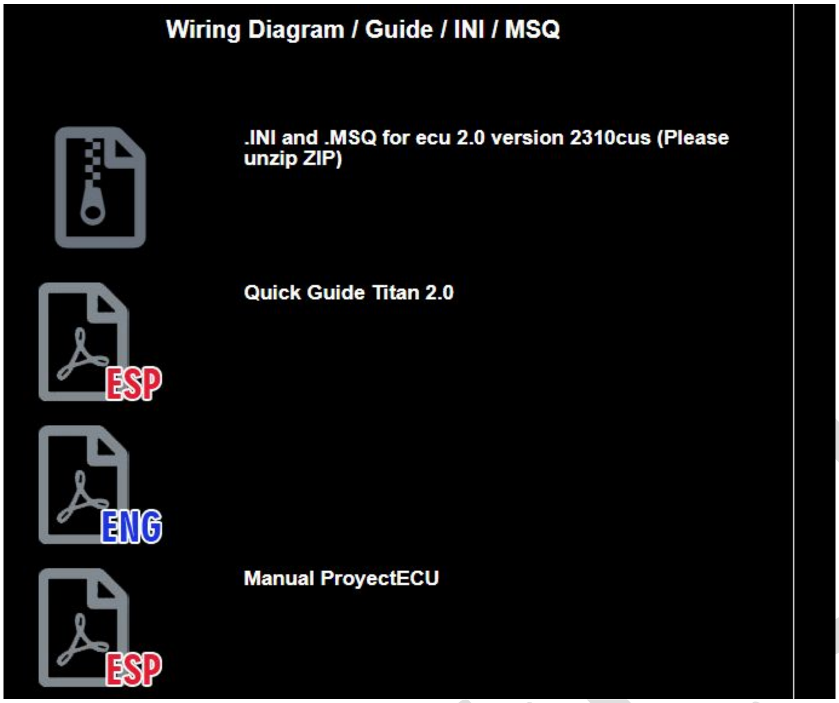
When you first start TunerStudio, you'll need to setup a new project which contains the settings, tune, logs etc. On the start up screen, select 'Create new project'

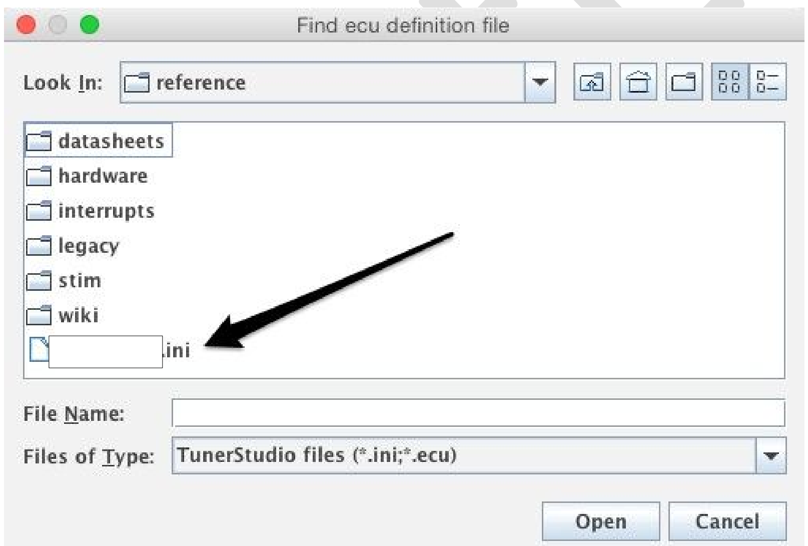


Give you project a name and select the directory you want the project to be stored in. Tuner Studio then requires a firmware definition file in order to communicate with the processor. Tick the 'Other / Browse' button.



Then download the .ini file from ProyecECU.com webpage and unzip the zip file and browse to the directory where the ini file is located.

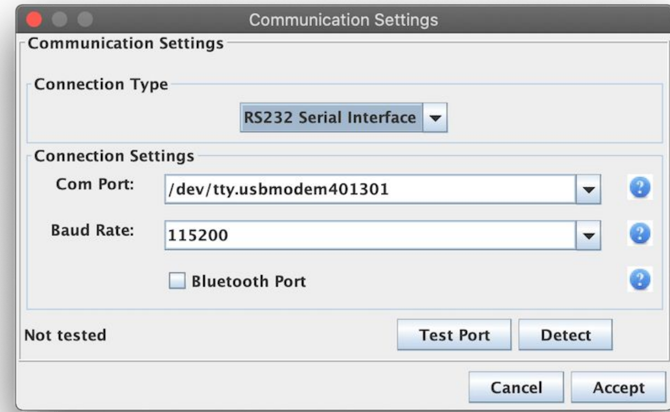




Configuration options

Comms settings

Select your comms options. The exact port name will depend on which operating system you are running, and this will be the same as in the Processor IDE. Baud rate should be 115200.

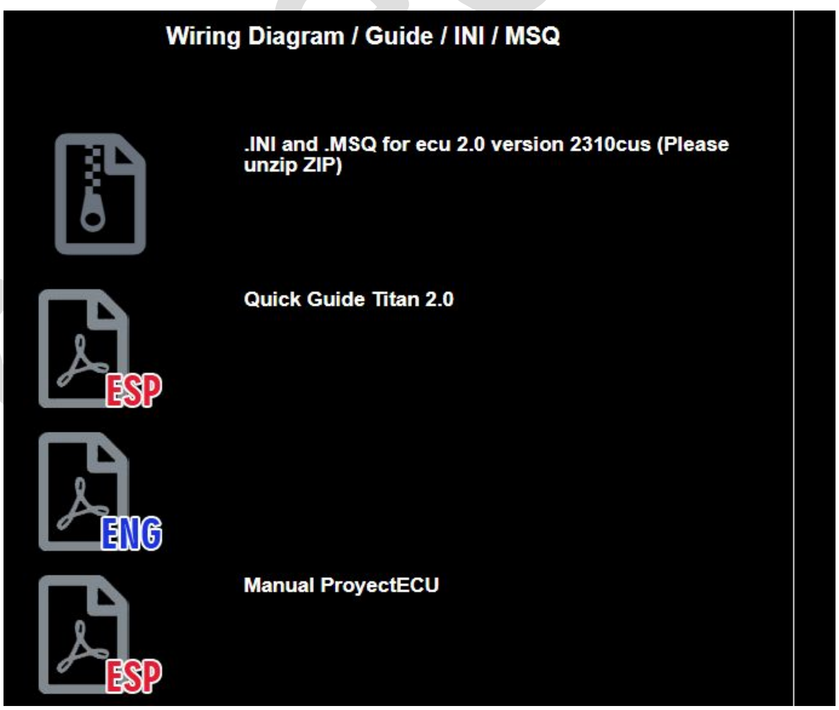


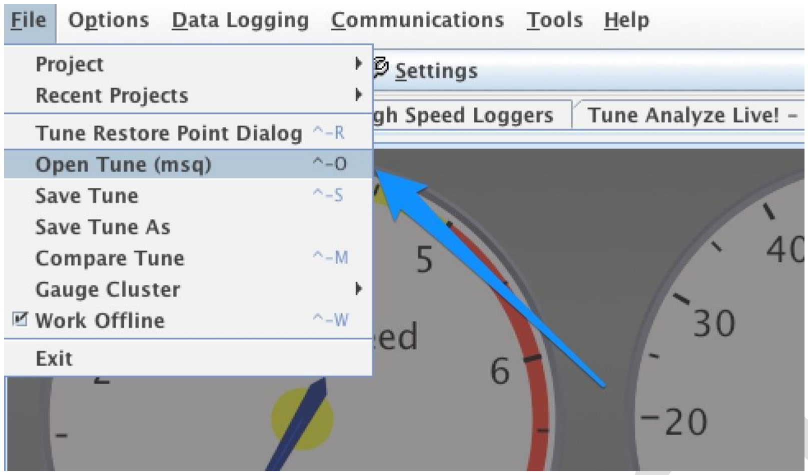
> **Note:** The Detect and Test port options require Tuner Studio version 3.0.60 or above to work correctly

Load base tune

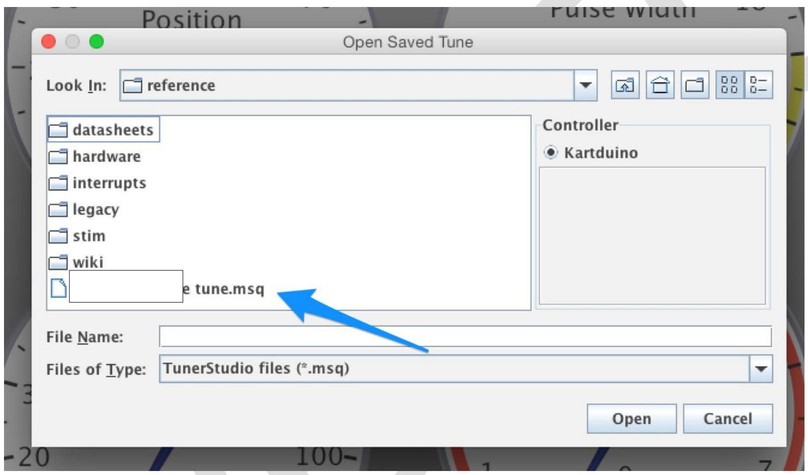
Once the project is created, you'll need to load in a base tune to ensure that all values are at least somewhat sane. Failure to do this can lead to very strange issues and values in your tune.

Normally a base tune is already loaded in the ECU, but also this basetune was downloaded alongside the .ini file from ProyecECU.com webpage.



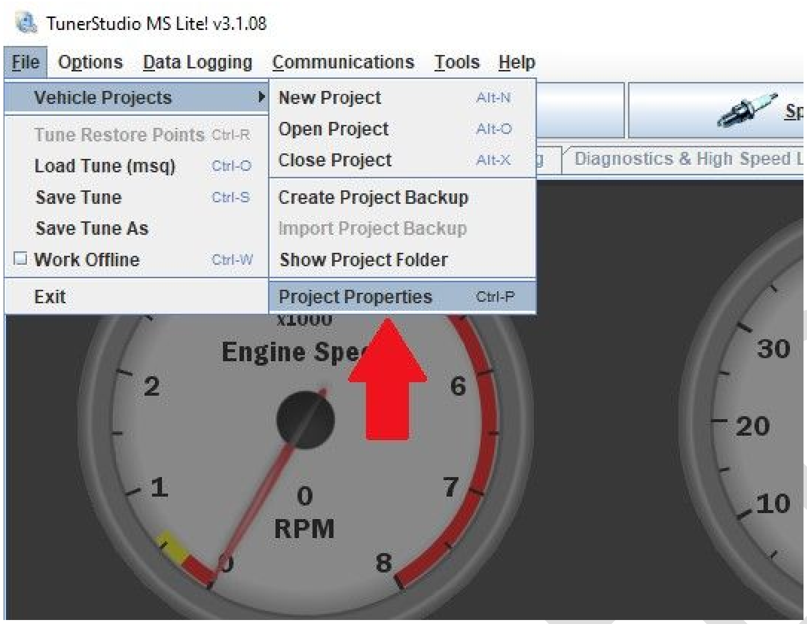


Look for the directory the base tune is located and open it:

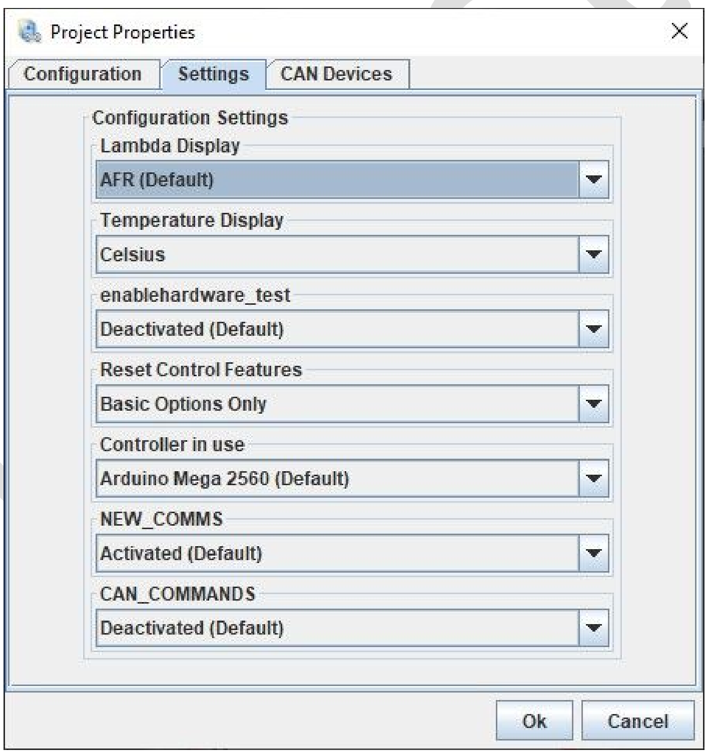


And that’s it! Tuner Studio should now attempt to connect to the ECU and show a real-time display of the ECU.

Configuring TunerStudio Project Properties



Once opened this page will be seen.



Settings Tab

The Settings tab does not affect the tune directly, but does change the way some things are displayed within Tuner Studio. Some menus are hidden by default, either for safety reasons or because they are still under development, and they can be enabled here.

Lambda Display

This changes whether the oxygen sensor reasons are shown in AFR (default) or Lambda.

Temperature Display

The temperature selection changes all degrees values within TunerStudio.

- Fahrenheit (Default)
- Celsius

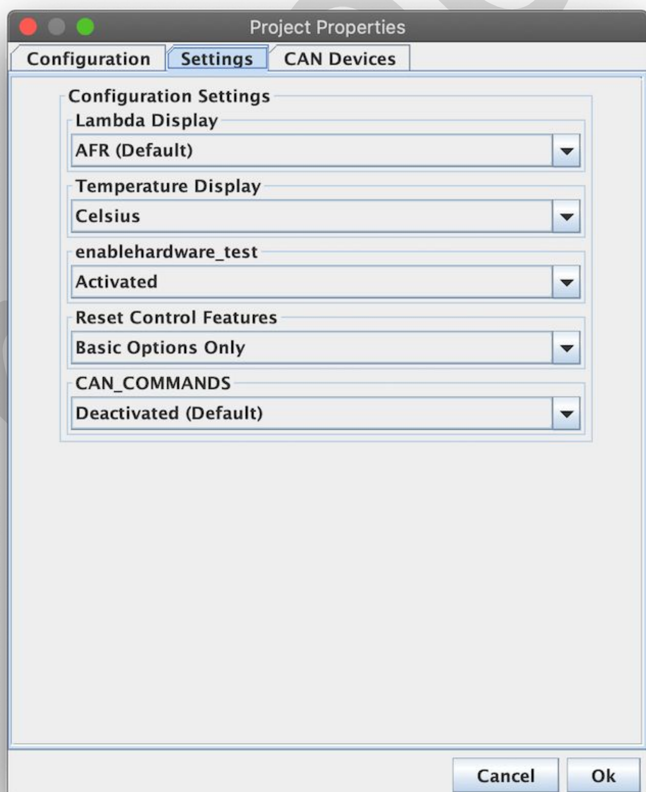
> Changing this value does not alter the values in tune at all, only which scale the values are displayed in

Enable Hardware Test

The hardware testing dialog allows you to manually turn the ignition and injection outputs on and off in order to test that the circuits are working. This can be dangerous if the outputs are connected to hardware however and so this dialog must be explicitly enabled.

> Please **ONLY** turn this on when the ECU is not connected to a vehicle

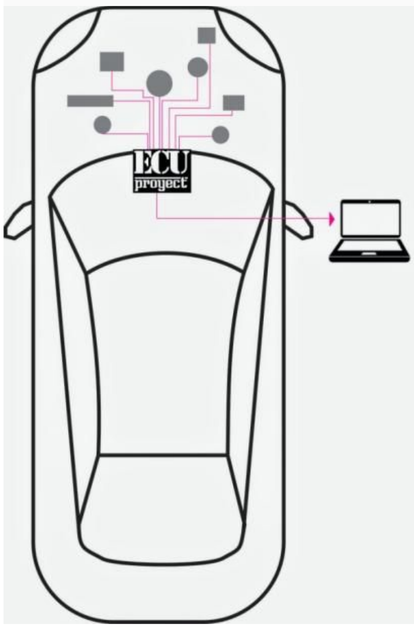
If Enabled, an additional Tab will appear on the tuning page

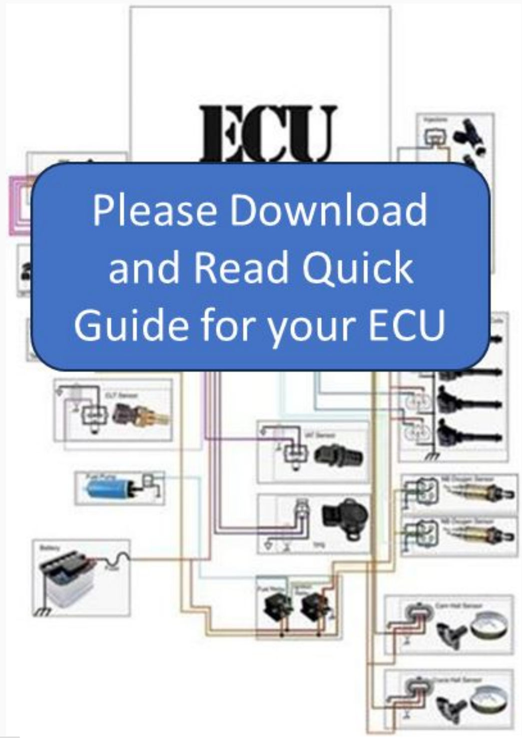


System Wiring Diagram

Wiring guide at a glance

The ECU can be configured in many ways depending on the engine, sensors, ignition, and fuel hardware used. For this reason, it is impossible to provide 1 single diagram that covers all scenarios, however the following is provided as a high level guide that can be used as a starting point. (Use ProyectECU Quick Guide).





Function Specific Diagrams

More detailed wiring guides for specific areas can be found below:

Injector Wiring

Introduction
The ECU contains from 2 to 8 injector control banks and is capable of supporting up to 8 cylinders with these.

Supported Injectors

The ECU supports High-Z (aka ‘high-impedance’ or ‘saturated’) injectors natively. Low-Z injectors are supported with the addition of resistors wired in series with the signal wires. High-Z injectors are typically those with a resistance greater than 8 Ohms.

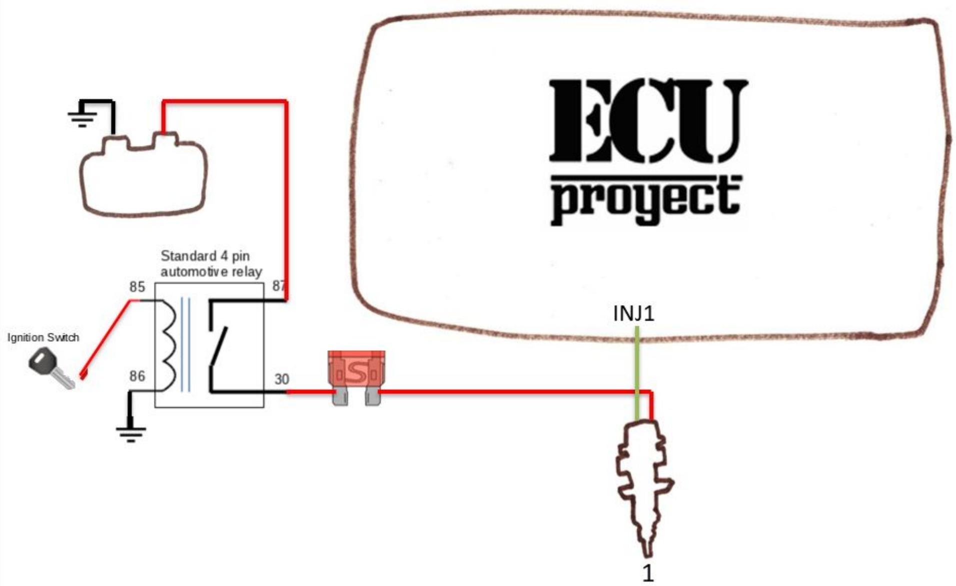
If “Low-Z” (“peak and hold” or PWM-controlled) injectors that are lower impedance are used, the wiring will require series resistors on each injector to avoid damaging the board with excessive current. The resistor ohms and watt rating can be calculated by Ohm’s Law or use an Internet calculator page such as the The ECU Injector Resistor Calculator.

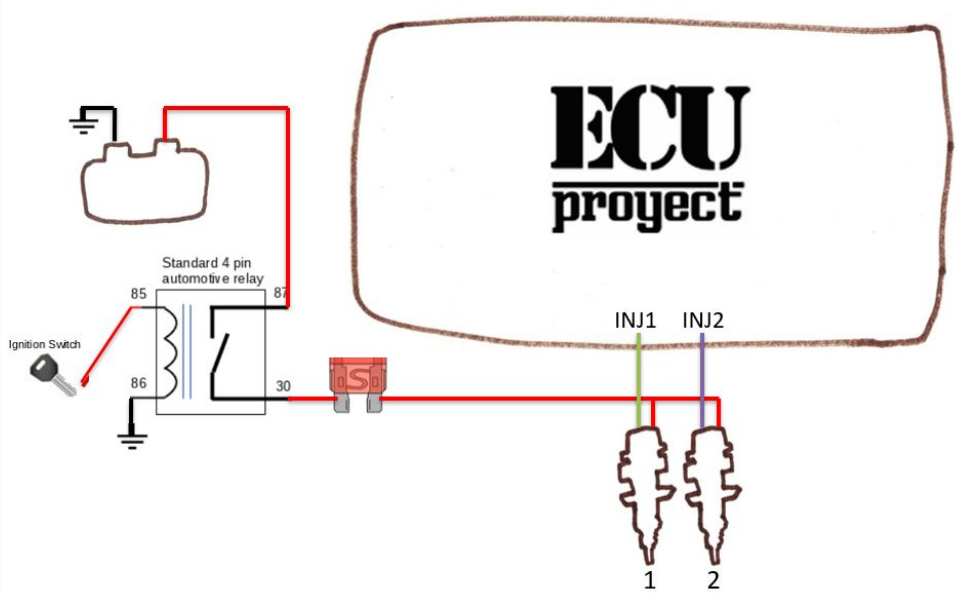
Layouts

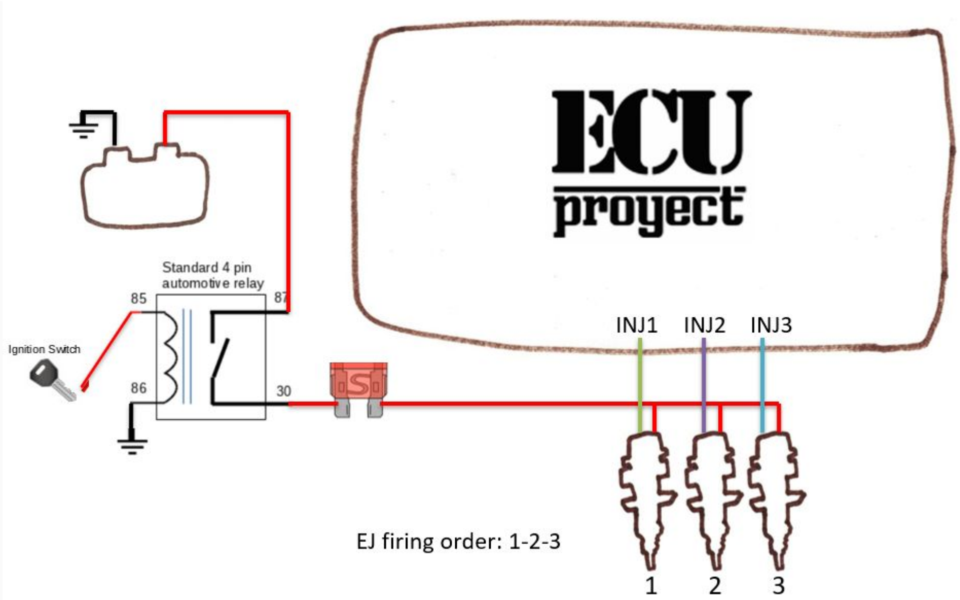
There is a number of ways that the injectors can be wired depending on your configuration and preference.

1, 2 and 3 cylinders:

For these configurations, each injector is connected to its own output of the ProyectECU board.

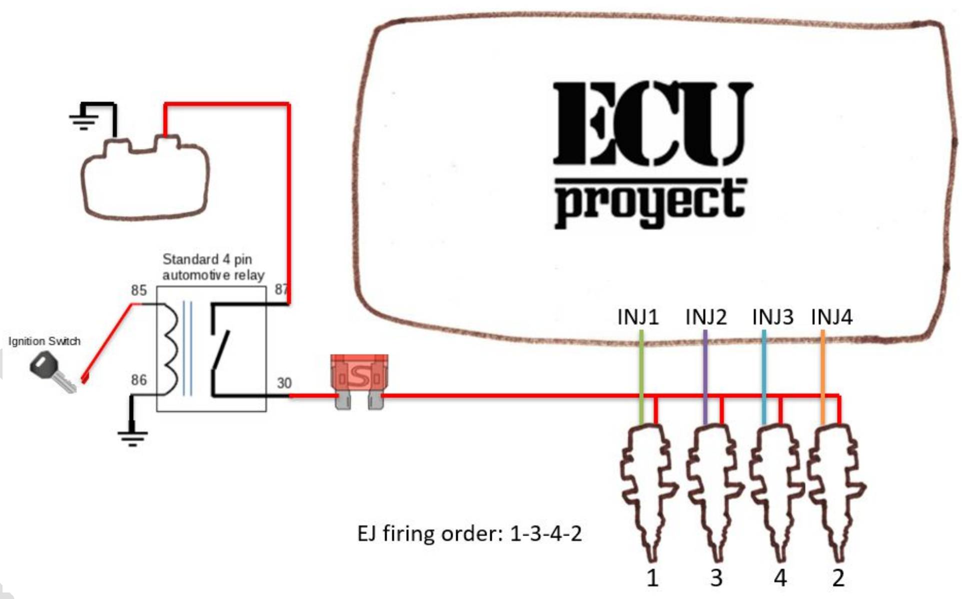






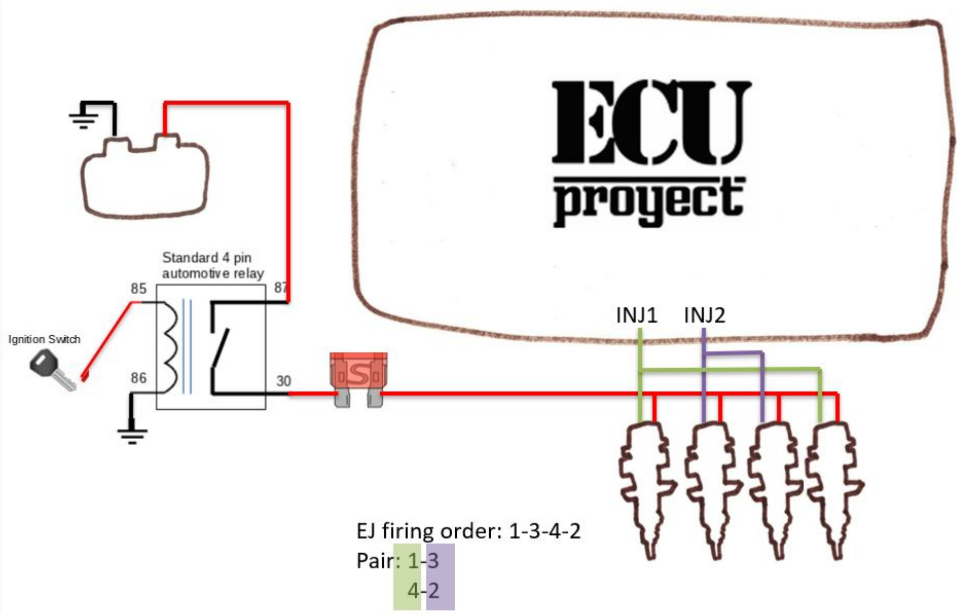
4 injectors
For 4 cylinders/injectors, there are 2 ways that these can be connected to the ECU:

Method1(Full sequential) (For ECUs with at least 4 injector channels) This method allows you to wire 1 injector per channel. The injector channels always fire in numerical order (ie 1, 2, 3, 4, 5, 6, 7, 8) so your injectors should be wired to take your firing order into account. Within Tuner Studio, this option can be enabled by selecting:
Settings -> *Engine Constants* -> *Injector Timing* -> *Sequential*



Method 2(Paired) In this method 2 injectors are connected to each injector channel. In this configuration, only 2 injector channels will be used. The injectors paired together must have their Top Dead Center (TDC) 360 crank degrees apart.

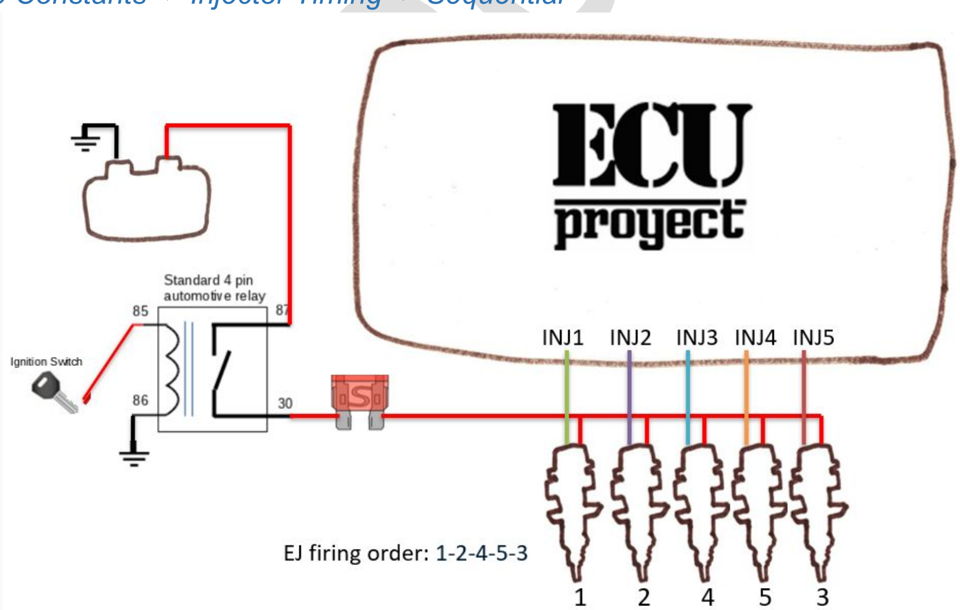
Settings -> *Engine Constants* -> *Injector Timing* -> *Paired*



5 injectors

Wire 1 injector per channel. (For ECUs with at least 6 injector channels). The injector channels always fire in numerical order (ie 1, 2, 3, 4, 5, 6, 7, 8) so your injectors should be wired to take your firing order into account. Within Tuner Studio, this option can be enabled by selecting:

Settings -> *Engine Constants* -> *Injector Timing* -> *Sequential*

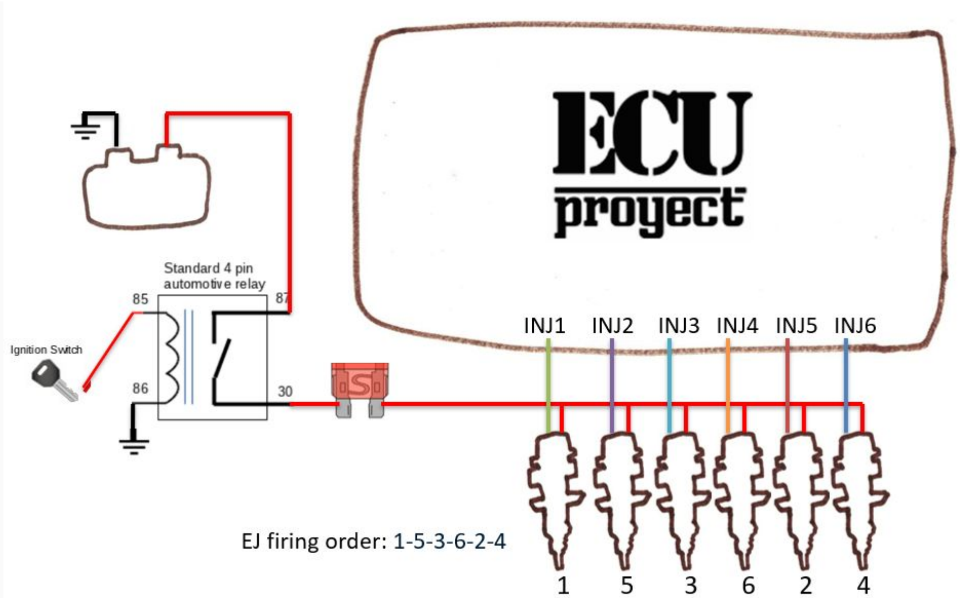


6 injectors

For 6 cylinders/injectors, there are 2 ways that these can be connected to the ECU:

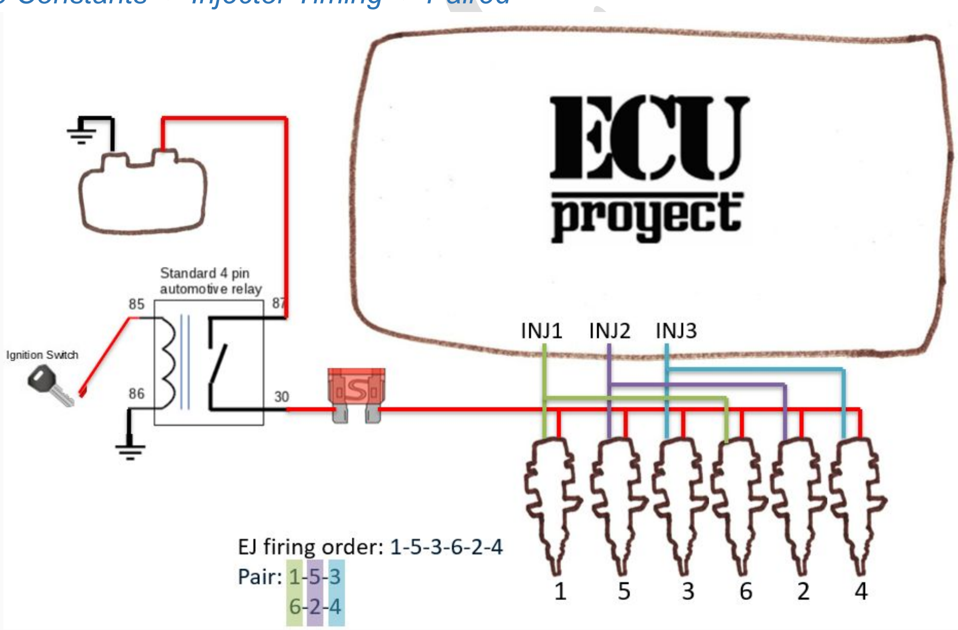
Method1(Full sequential) (For ECUs with at least 6 injector channels) This method allows you to wire 1 injector per channel. The injector channels always fire in numerical order (ie 1, 2, 3, 4, 5, 6, 7, 8) so your injectors should be wired to take your firing order into account. Within Tuner Studio, this option can be enabled by selecting:

Settings -> *Engine Constants* -> *Injector Timing* -> *Sequential*



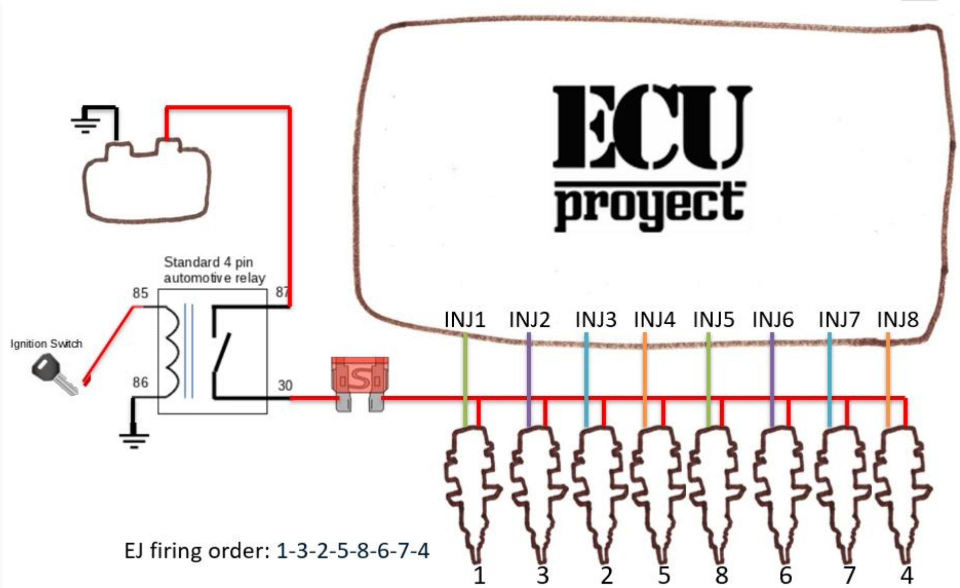
Method 2(Paired) In this method 2 injectors are connected to each injector channel. In this configuration, only 2 injector channels will be used. The injectors paired together must have their Top Dead Center (TDC) 360 crank degrees apart.

Settings -> *Engine Constants* -> *Injector Timing* -> *Paired*

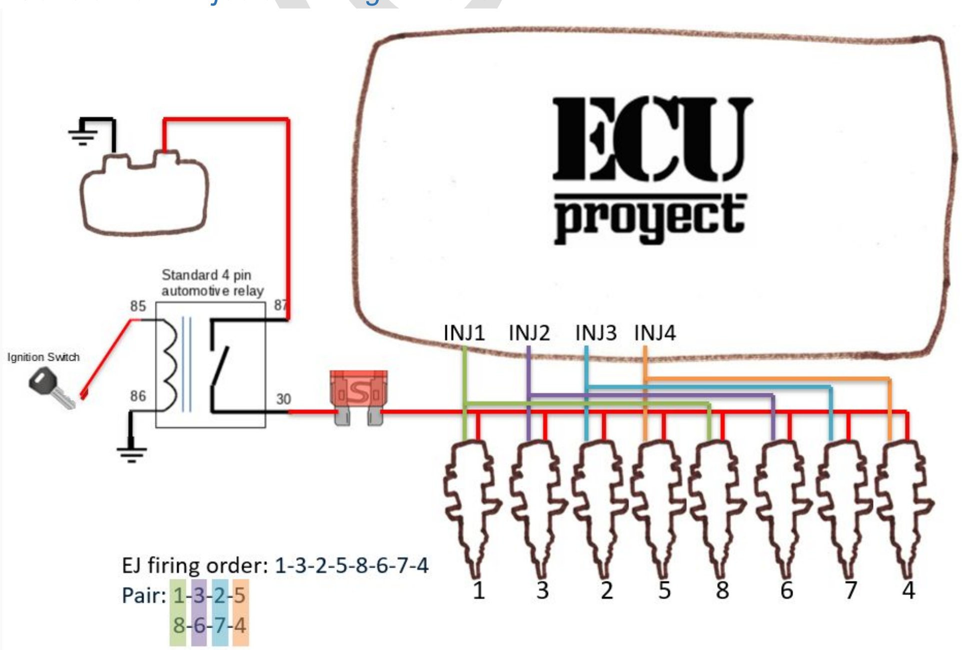


8 injectors
For 6 cylinders/injectors, there are 2 ways that these can be connected to the ECU:

Method1(Full sequential) (For ECUs with at least 8 injector channels) This method allows you to wire 1 injector per channel. The injector channels always fire in numerical order (ie 1, 2, 3, 4, 5, 6, 7, 8) so your injectors should be wired to take your firing order into account. Within Tuner Studio, this option can be enabled by selecting:
Settings -> *Engine Constants* -> *Injector Timing* -> *Sequential*



Method 2(Paired) In this method 2 injectors are connected to each injector channel. In this configuration, only 2 injector channels will be used. The injectors paired together must have their Top Dead Center (TDC) 360 crank degrees apart.
Settings -> *Engine Constants* -> *Injector Timing* -> *Paired*



Ignition Wiring

Overview

Ignition output configuration can be one of the most difficult areas of ECU wiring and one that often causes the most confusion. A large part of this complexity comes from the huge number of different ignition types that are available, with there being significant changes in the hardware used in the late 80s and throughout the 90s compared to newer designs.

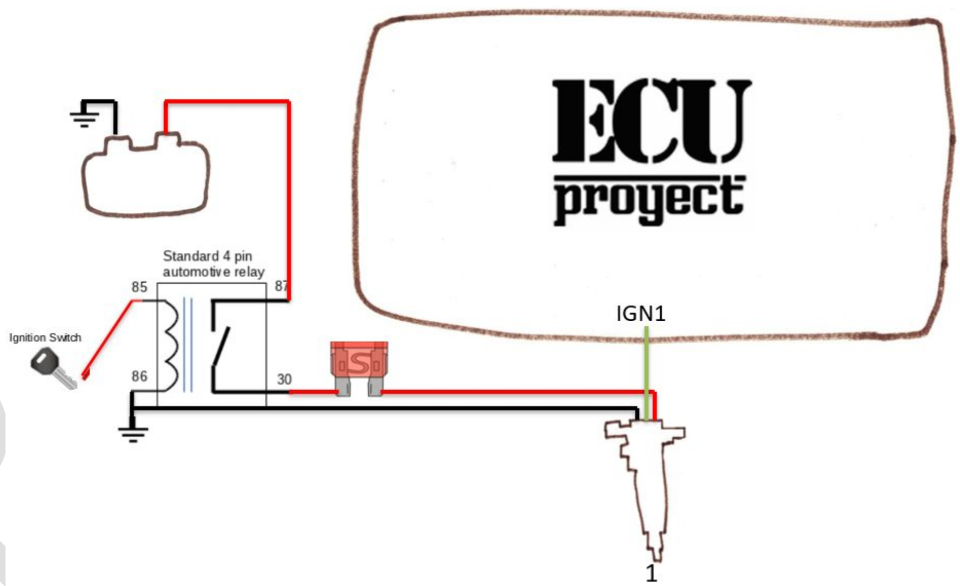
Whilst this guide does not cover all ignition styles and hardware, it does cover the most common scenarios. Generally, it is recommended (Where possible) to use newer styles of ignition hardware (Typically 'smart' Coil-on-Plug or Coil-Near-Plug) rather than utilizing separate ignition modules.

Sequential (Coil per cylinder) (Coil on Plug)

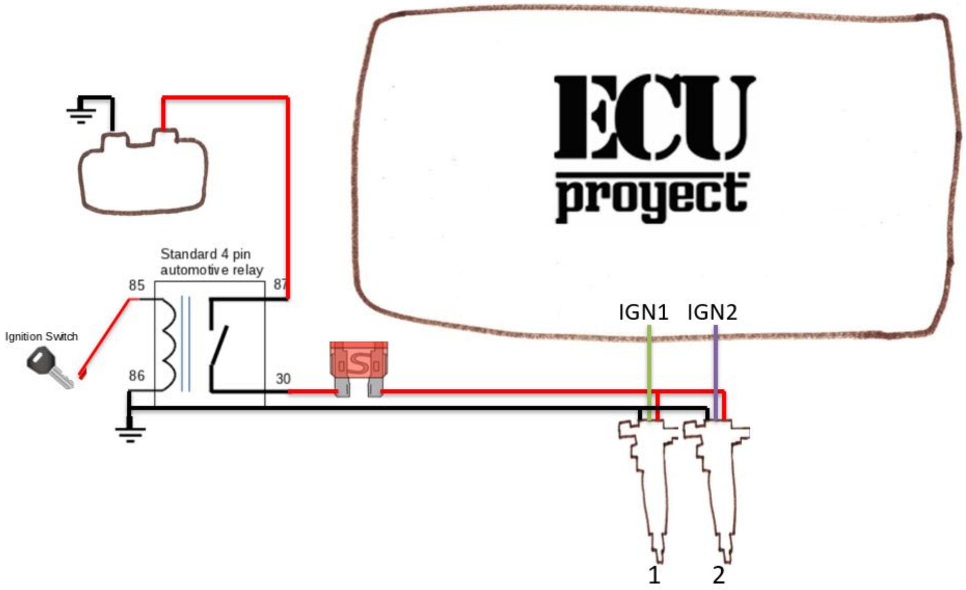
Sequential ignition control using individual coil per cylinder or Coil-on Plugs coils dramatically simplifies the ignition wiring. With this configuration, each coil (and subsequently each cylinder) connects to a single ignition output of the ECU, wired in the firing order.

1 to 8 cylinders:
For these configurations, each ignitor is connected to its own output of the ProyectECU board.
Only for ECUs with the enough number of ignition outputs for each channel needed.

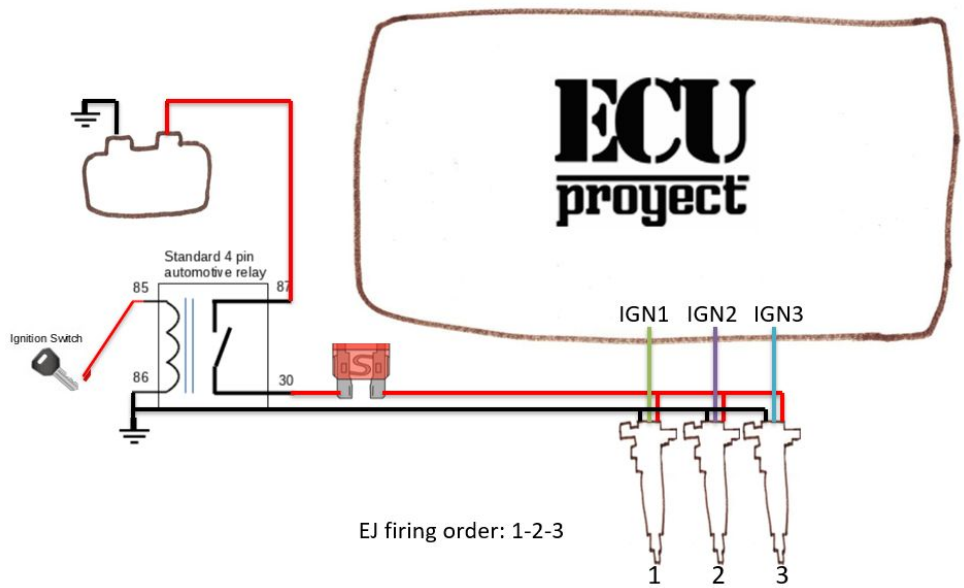
1 cylinder:



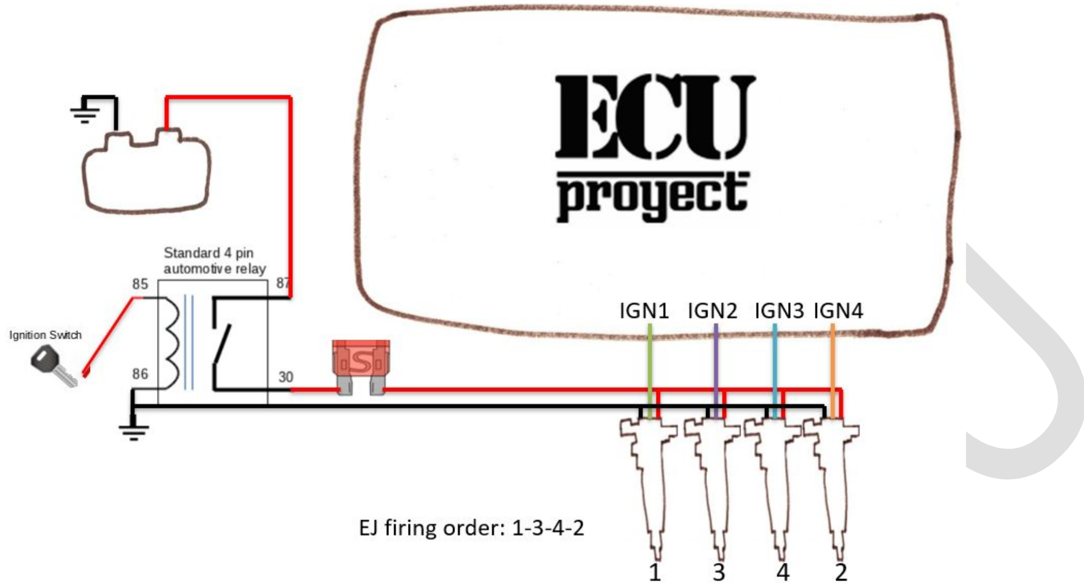
2 cylinder:



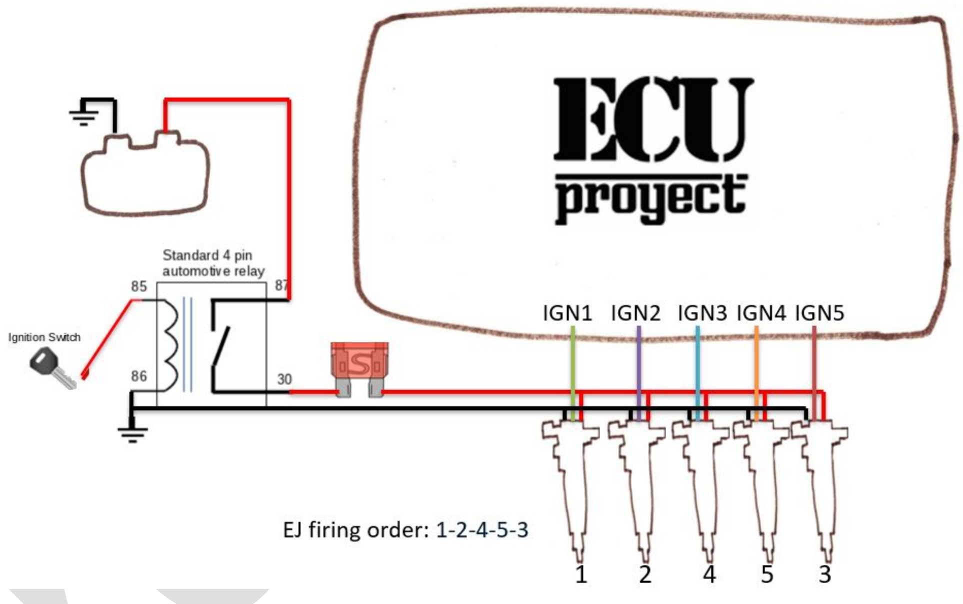
3 cylinder:



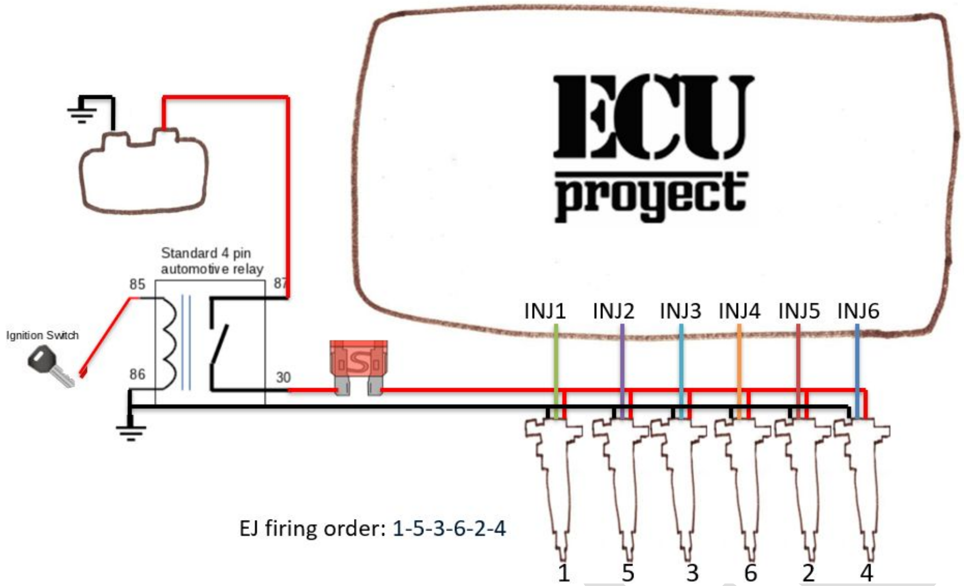
4 cylinder:



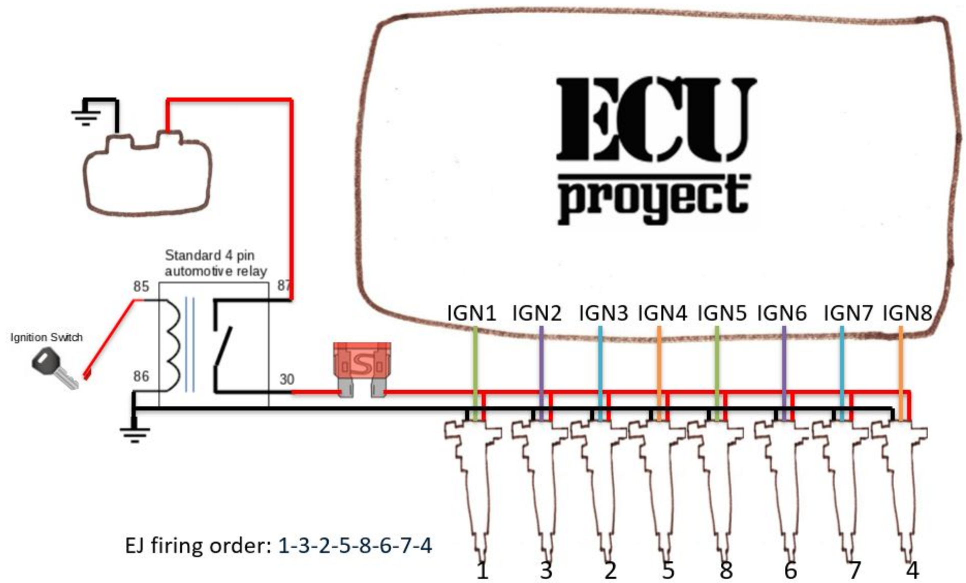
5 cylinder:



6 cylinder:



8 cylinder:



Wasted Spark

Wasted spark is a common means of controlling spark that requires only half the number of ignition outputs as there are cylinders, with 2 cylinder being attached to each output. EG: * 4 cylinder engine requires 2 ignition outputs * 6 cylinder engine requires 3 ignition outputs * 8 cylinder engine requires 4 ignition outputs

Wasted Spark has the advantage of not requiring any cam signal or input as it does not need to know the engine phase. This is possible by firing the ignition outputs once per revolution and pairing that output to 2 cylinders that are both at TDC (With one cylinder on compression stroke and the other on exhaust)

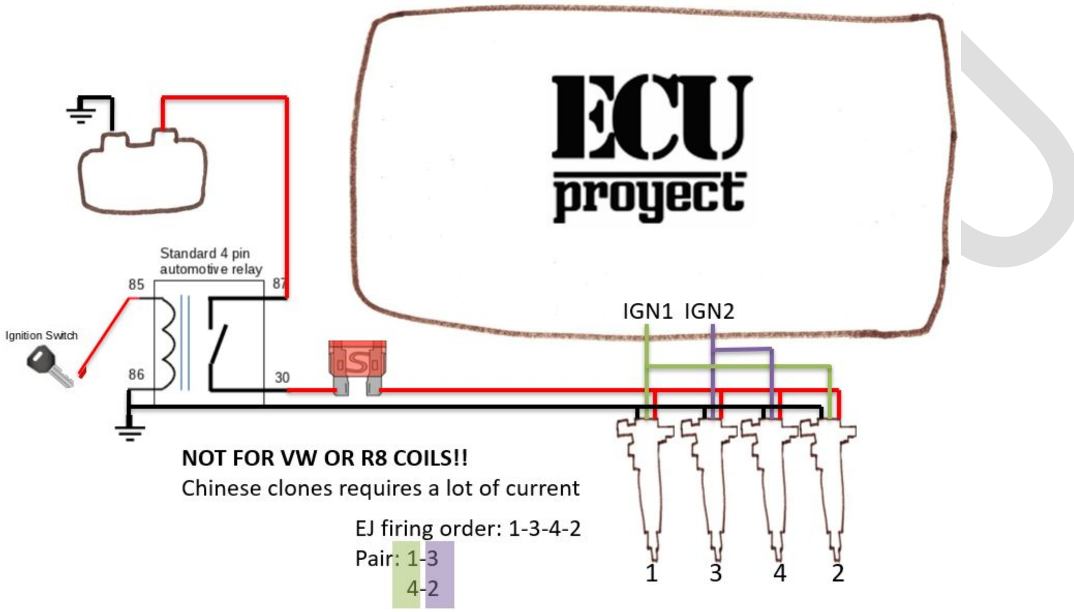
When using wasted spark, it is critical the correct pairs coils and/or spark plugs are joined together.

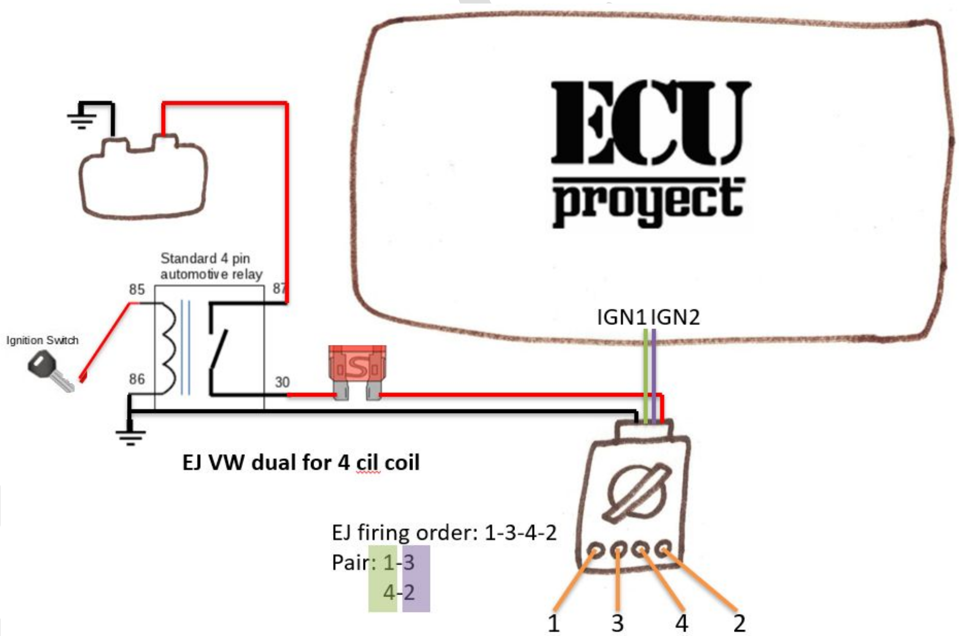
There are many dual poles, wasted spark coil packs available both with and without built in igniters.

Either are suitable for use with the ECU, but use of coils with built-in igniters is recommended.

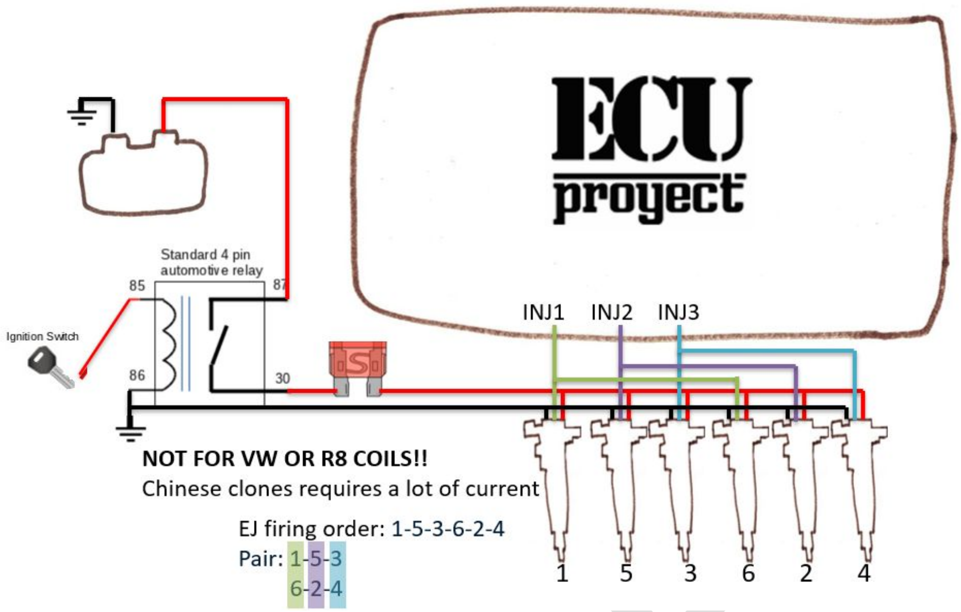
Warning: VW pencil type coils requires a lot of Ignition logical level current. So, using more than one per channel is not recommended.

4 cylinders wasted spark:

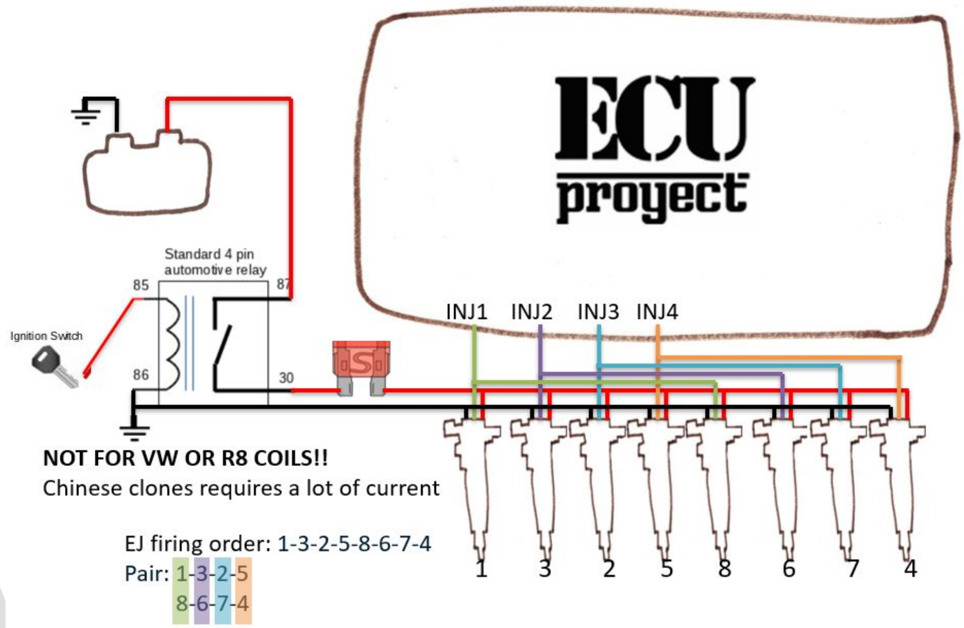




6 cylinders wasted spark:

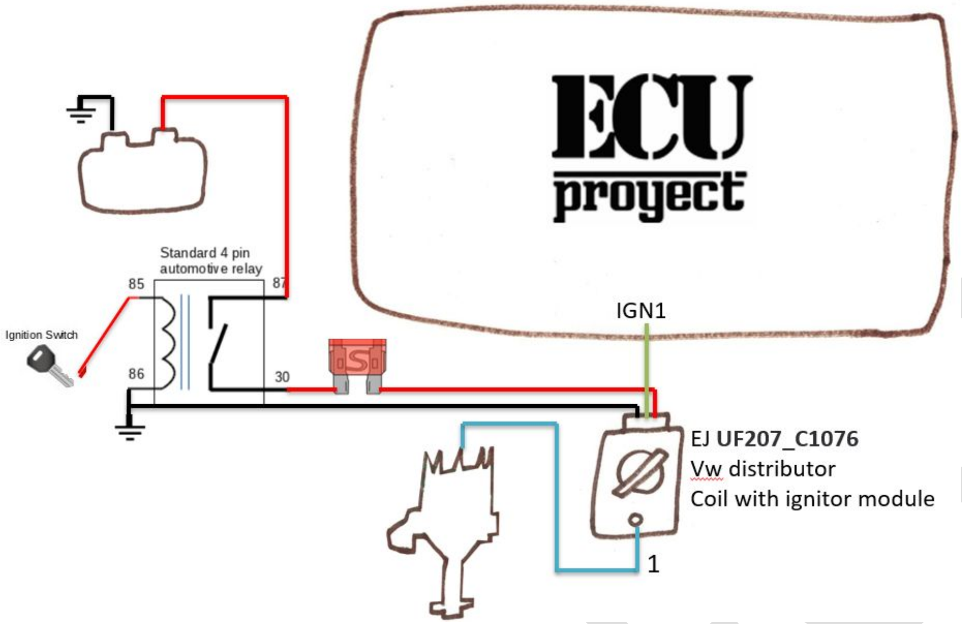


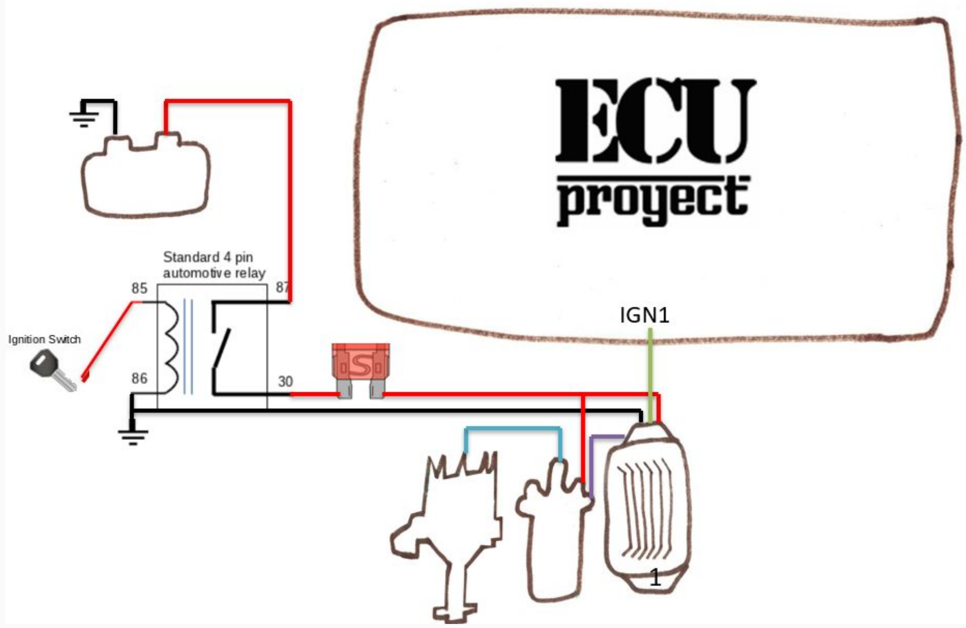
8 cylinders wasted spark:



Distributor

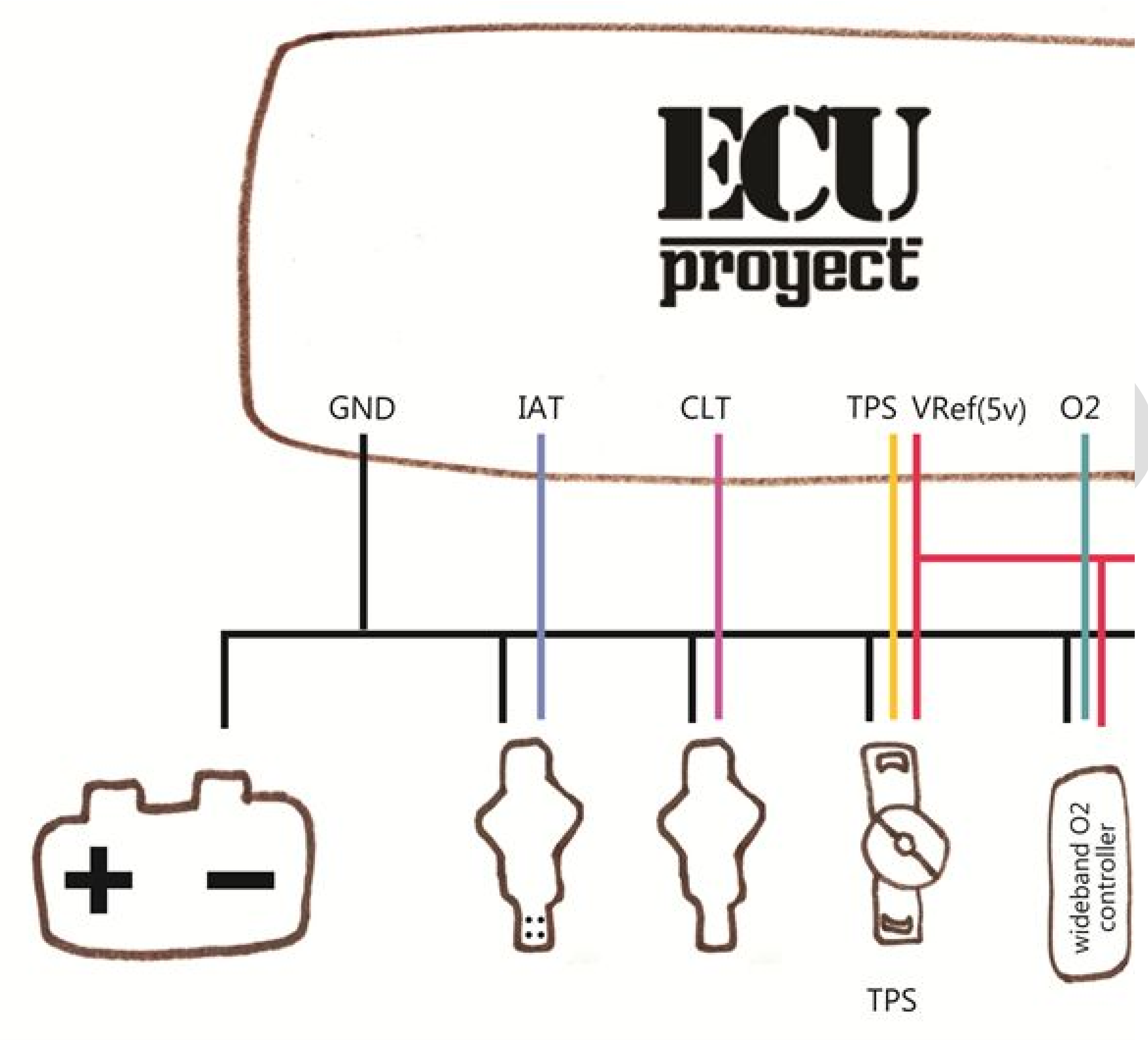
If a distributor remains in use, only a single output is required from the ECU. This should be fed into a single channel ignition module (Such as the common Bosch 124) which can then drive the coil.





Analog Sensor Wiring

Analog sensors provide data such as temperatures, throttle position and O2 readings to the ECU. The diagram below shows the typical wiring for these sensors.



Notes: The use of 2-wire temperature sensors is recommended. While 1-wire sensors will work, they are almost always considerably less accurate.
A 3-wire variable TPS is required. On/Off type TPS are not suitable.

Configuration

Important menus:

Settings Menu
Engine Constants
Injector Characteristics
Trigger Setup
IAT Density

Tuning Menu
Acceleration Wizard
AFR/O2
Rev Limits
Flex Fuel
Staged Injection

Spark Menu
Spark Settings
Dwell
IAT Retard

Starting/Idle Menu
Cranking
Warmup
Idle control
Idle
Idle Advance

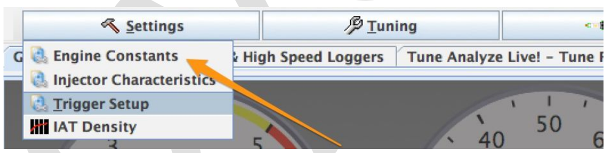
Accessories Menu
Thermo fan
Launch Control and Flat shift
Fuel pump
Boost Control
Tacho

Tools Menu
Sensor Calibration

Engine Constants

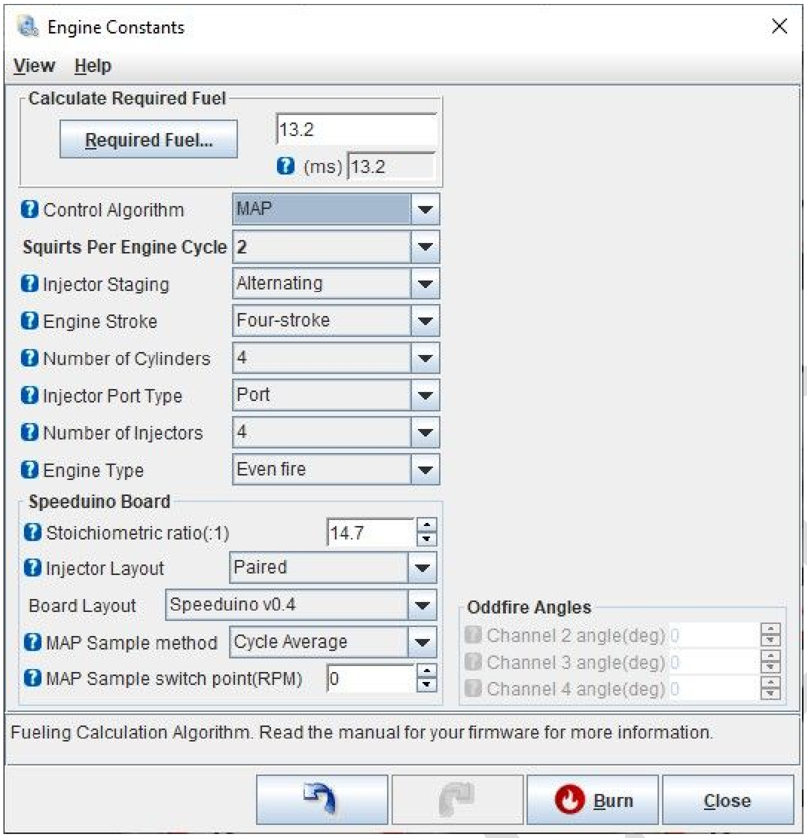
Overview

From the Settings menu, select Constants



Here you need to setup the engine constants. Fill out the fields in the bottom section before calculating the Required Fuel.

Configuration



Required Fuel Calculator

The required fuel calculator determines the theoretical fuel injection time that would be required at 100%VE. This is determined by knowing the engine capacity, the size and number of the fuel injectors and the number of squirts that will be performed in each cycle. Increasing this figure will lead to an overall increase in the amount of fuel that is injected **at all points** of the VE map (And vice versa).

> You should set all the values in the Settings section below before performing the Required Fuel calculation

Settings

- **Control Algorithm:** The load source that will be used for the fuel table
- **Squirts per Engine Cycle:** How many squirts will be performed over the duration of the engine cycle (Eg 720 degrees for a 4 stroke). most engines will not require values greater than 4. For sequential installations, this should be set to 2 with the injector staging set to 'Alternating'(Internally The ECU will adjust the squirts to 1)
  - Note that for 3 and 5 squirts, you must have a cam signal in addition to the crank.
- **Injector Staging:** This configures the timing strategy used for the injectors

  - **Alternating** (Recommended for most installs) - Injectors are timed around each cylinders TDC. The exact closing angle can be specific in the Injector Characteristics dialog.
  - **Simultaneous** - All injectors are fired together, based on the TDC of cylinder 1.

- **Engine stroke:** Whether the engine is 2 stroke or 4 stroke
- **Number of cylinders:** Number of cylinders in the engine. For rotary engines, select 4.
- **Injector Port Type:** Option isn't used by firmware. Selection currently does not matter
- **Number of injectors:** Usually the same as number of cylinders (For port injection)
- **Engine Type:** Whether the crank angle between firings is the same for all cylinders. If using an Odd fire engine (Ex Some V-Twins and Buick V6s), the angle for each output channel must be specific.
- **Stoichiometric ratio:** The stoichiometric ration of the fuel being used. For flex fuel, choose the primary fuel.
- **Injector Layout:** Specifies how the injectors are wired in

  - **Paired:** 2 injectors are wired to each channel. The number of channels used is therefore equal to half the number of cylinders.
  - **Semi-Sequential:** Semi-sequential: Same as paired except that injector channels are mirrored (1&4,2&3) meaning the number of outputs used are equal to the number of cylinders.
    Only valid for 4 cylinders or less.
  - **Sequential**: 1 injector per output and outputs used equals the number of cylinders. Injection is timed over full cycle. Only available for engines with 4 or fewer cylinders.
- **Board Layout:** Specifies the input/output pin layout based on which ProyectECU board you're using. For specific details see the first page of quick guide of the ECU, (16CH for 2.0 ecus, and UA4C for 1.0 ecus)
- **MAP Sample Method:** How the MAP sensor readings will be processed:

  - **Instantaneous:** Every reading is used as it is taken. Makes for a highly fluctuating signal, but can be useful for testing
  - **Cycle Average:** The average sensor reading across 720 crank degrees is used. This is of Event average are the recommended options for 4 of more cylinders
  - **Cycle Minimum:** The lowest value detected across 720 degrees is used. This is the recommended method for less than 4 cylinders or ITBs
  - **Event Average:** Similar to Cycle Average, however performs the averaging once per ignition event rather than once per cycle. Generally offers faster response with a similar level of accuracy.

- **MAP Sample switch point:** Instantaneous MAP sampling method is used below this RPM and the selected method is used above this RPM. Default value: 0 RPM. This can be used to improve low RPM throttle response, by using instantaneous MAP sample method around idle RPM for fastest MAP response and then switch to other methods at higher RPM to get rid of the MAP noise that instantaneous mode can have.

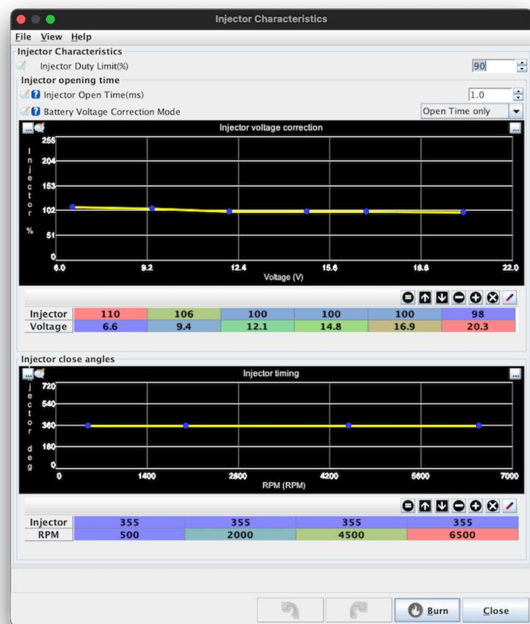
The Oddfire angles should only be used on oddfire engines (Primarily some specific V6s)

Injector Characteristics

Overview

Fuel injectors have unique hardware properties that must be accounted for within your tune. Ideally these will be provided as part of the specifications for your injectors, however in some cases the data may not be available or be difficult to find. Typical values are given below as starting points for these cases.

Settings



Variable	Typical value	Comment
Injector Open Time	0.9 - 1.5	The time the injector takes to open completely once triggered, plus the time necessary to close. This is specific to each injector type and version.
Injector Close Angles	355	This represents the angle (ATDC 0-720), relative to each cylinders TDC, that the injector squirt will end. This can be varied per channel (Including for semi-sequential wiring), but the default value of 355 is suitable for most applications.

Variable	Typical value	Comment
Injector Duty Limit	85%	The injector opens and closes once per crank revolution so, taking into account the open time of the injector, the duty cycle is

		limited to avoid this exceeding the revolution time. A value of 85% is recommended, but a higher value can be used for faster opening injectors. Note that once this duty cycle limit is reached, it will not be exceeded as the fuel injector cannot close and reopen fast enough to supply more fuel. This may potentially cause lean conditions at high RPM. If hitting this limit, strongly consider whether larger injectors are required.
Injector Voltage Correction	100%	The percentage the the injector pulse width is varied with changes in supply voltage. A value of 100% means no change to the pulse width.
Voltage Correction Mode	Open time only	Whether the voltage correction applies to just the opening time or the whole pulse width.

Trigger Setup

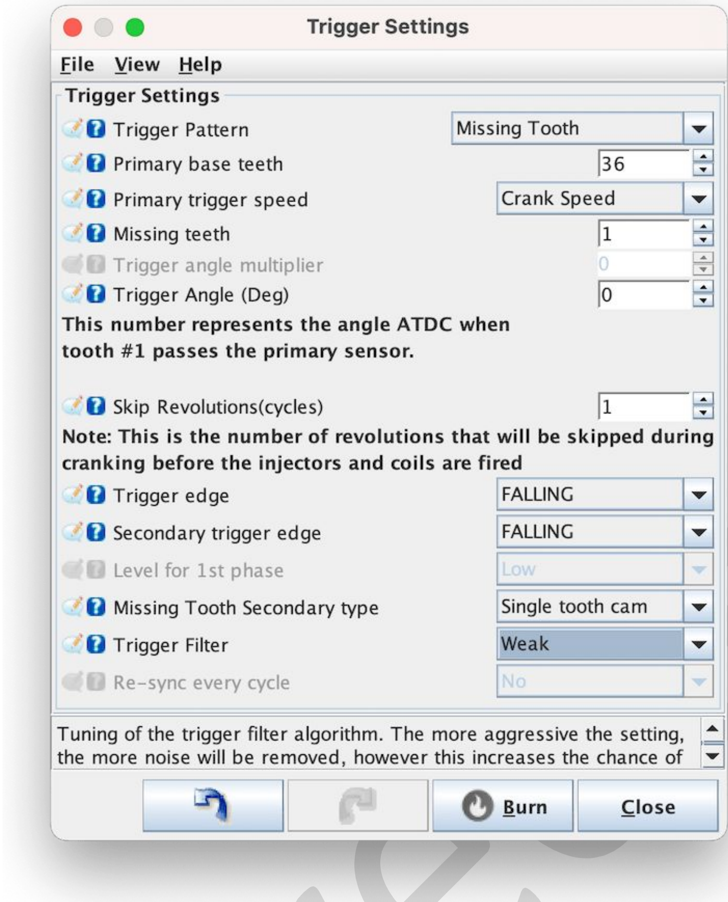
Overview

One of the most critical component so far in EFI setup, is the Crank Angle Sensor (Crank) and how it is used by the ECU. The Trigger settings dialog is where the trigger configuration is defined, and it is vitally important to have this correct before trying to start your engine.

With incorrect settings, you may have issues getting sync or see erratic RPM readings.

Note that many of the settings on this dialog are dependent on your configuration and it is therefore normal that some options maybe greyed out.

Trigger Settings



- **Trigger Pattern** - The pattern used by the crank/cam sensor setup on your engine. For a full list of the supported patterns, see the Decoders Section (at the end of this Document).
- **Primary Base teeth** - For patterns where the number of teeth are variable (missing tooth, dual wheel, etc), this number represents the number of teeth on the primary wheel. For missing tooth type wheels, this number should be the count as if there were no teeth missing.
- **Primary trigger speed** - The speed at which the primary input spins. It is closely related to the Primary Base teeth setting and indicates whether that number of teeth passes the sensor once every crank revolution or every cam revolution.
- **Missing teeth** - If using the missing tooth pattern, this is the size of the gap, given in ‘missing teeth’. Eg 36-1 has 1 missing tooth. 60-2 has 2 missing teeth etc. The missing teeth **MUST** be all located in a single block, there cannot be multiple missing tooth gaps around the wheel.
- **Trigger angle multiplier** - This option is used only on the Non-360 pattern.
- **Trigger angle** - The angle of the crank, **After Top Dead Centre (ATDC)**, when tooth #1 passes the sensor on the primary (crank) input. This setting is critical for The ECU to accurately know the current crank angle. See section below (‘Finding tooth #1 and trigger angle’) for further information on how to

determine this value. You should be using a timing light to confirm angle is correct once calculated. Without doing this your angle may be incorrect.

- **Skip revolutions** - The number of revolutions the engine should perform before the Sync flag is set. This can help prevent false sync events when cranking. Typical values are from 0 to 2
- **Trigger edge** - Whether the primary signal triggers on the Rising or Falling edge.
  - **The ECU have a VR Conditioner and require specific setting:**
    - RISING
- **Secondary trigger edge** - Whether the secondary signal triggers on the Rising or Falling edge
- **Missing Tooth Secondary type** - Cam mode/type also known as Secondary Trigger Pattern.
- **Level for 1st phase** - Only active with “Poll level” cam decoder. The level of the cam trigger input will be checked at crank tooth #1 and this defines if the level is supposed to be High or Low at 1st phase of the engine.
- **Trigger filter** – A time based software filter that will ignore crank/cam inputs if they arrive sooner than expected based on the current RPM. The more aggressive the filter, the closer to the expected time the filter will operate. Higher levels of filtering may cause true pulses to be filtered out however, so it is recommended to use the lowest setting possible.
- **Re-sync every cycle** - If set to yes, the system will look for the sync conditions every cycle rather than just counting the expected number of teeth. It is recommended that this option should be turned on, however if you have a noisy crank/cam signal you may need to turn it off as it can cause sync to drop out occasionally. Once The ECU has full sync it will continue to run in full sequential mode unless sync loss on crack trigger occurs.

Finding tooth #1 and trigger angle

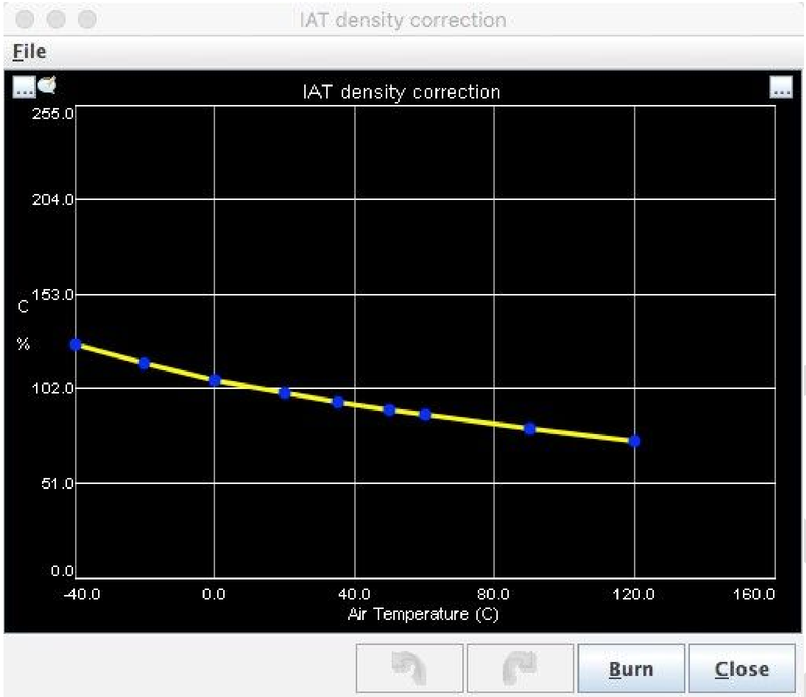
Please refer to the Trigger Patterns and Decoders for the trigger that you are using.

IAT Density

Overview

The IAT density curve represents the change in oxygen density of the inlet charge as temperature rises.

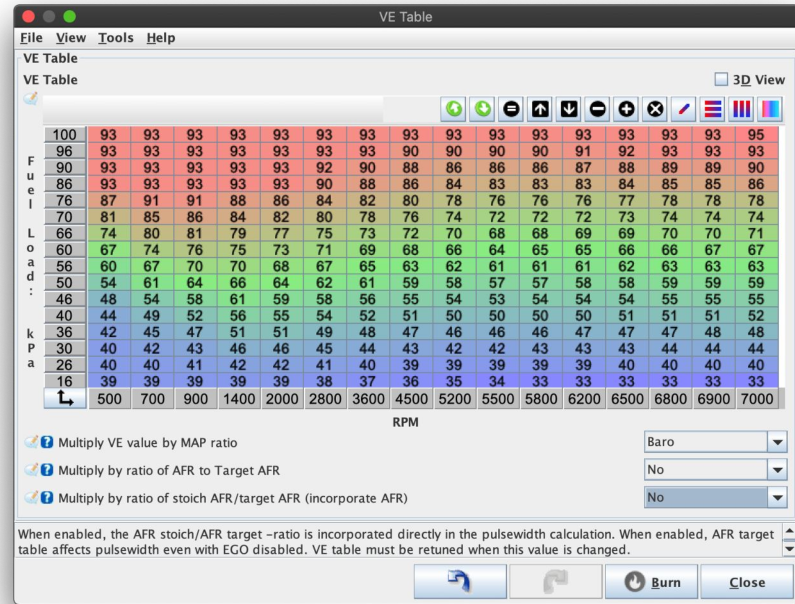
Example Curve



This default curve approximately follows the ideal gas law and is suitable for most installations, however if you are seeing very high inlet temperatures (Either due to heat soak in the engine bay or from turbocharging) the you may need to adjust the hot end of this curve.

Fuel (VE) table

The fuel or VE table is the primary method of controlling the amount of fuel that will be injected at each speed/load point.



Fuel Load: kPa \ RPM	500	700	900	1400	2000	2800	3600	4500	5200	5500	5800	6200	6500	6800	6900	7000
100	93	93	93	93	93	93	93	93	93	93	93	93	93	93	93	95
96	93	93	93	93	93	93	93	90	90	90	90	91	92	93	93	93
90	93	93	93	93	93	92	90	88	86	86	86	87	88	89	89	90
86	93	93	93	93	93	90	88	86	84	83	83	83	84	85	85	86
76	87	91	91	88	86	84	82	80	78	76	76	76	77	78	78	78
70	81	85	86	84	82	80	78	76	74	72	72	72	73	74	74	74
66	74	80	81	79	77	75	73	72	70	68	68	69	69	70	70	71
60	67	74	76	75	73	71	69	68	66	64	65	65	66	66	67	67
56	60	67	70	70	68	67	65	63	62	61	61	61	62	63	63	63
50	54	61	64	66	64	62	61	59	58	57	57	58	58	59	59	59
46	48	54	58	61	59	58	56	55	54	53	54	54	54	55	55	55
40	44	49	52	56	55	54	52	51	50	50	50	50	51	51	51	52
36	42	45	47	51	51	49	48	47	46	46	46	47	47	47	48	48
30	40	42	43	46	46	45	44	43	42	42	43	43	43	44	44	44
26	40	40	41	42	42	41	40	39	39	39	39	39	40	40	40	40
16	39	39	39	39	39	38	37	36	35	34	33	33	33	33	33	33

Configuration

The fuel map is a 3D, interpolated table that uses RPM and fuel load to lookup the desired VE value. The fuel load axis is determined by whether you are using Speed Density (MAP kPa) or Alpha-N (TPS) for your fuel load (See Engine_Constants).

The values in this table represent a percentage of the Required Fuel amount that will be injected when the engine is at a given speed/load point.

Options

- **MultiplyVEvaluebyMAPratio:** Enabling this option 'flattens' the fuel table by multiplying the value in the current speed/load point by the MAP value divided by either the Baro value (in kPa) or a fixed 100%. Using the Baro option adjusts fueling based on barometric reading, but for better results it's recommended to use the Barometric Correction curve instead.
  - You can tune with or without this option enabled, but it is generally recommended to be turned on as it will allow for simpler and more predictable tuning results.
  - For new tunes it is recommended to use the Fixed option

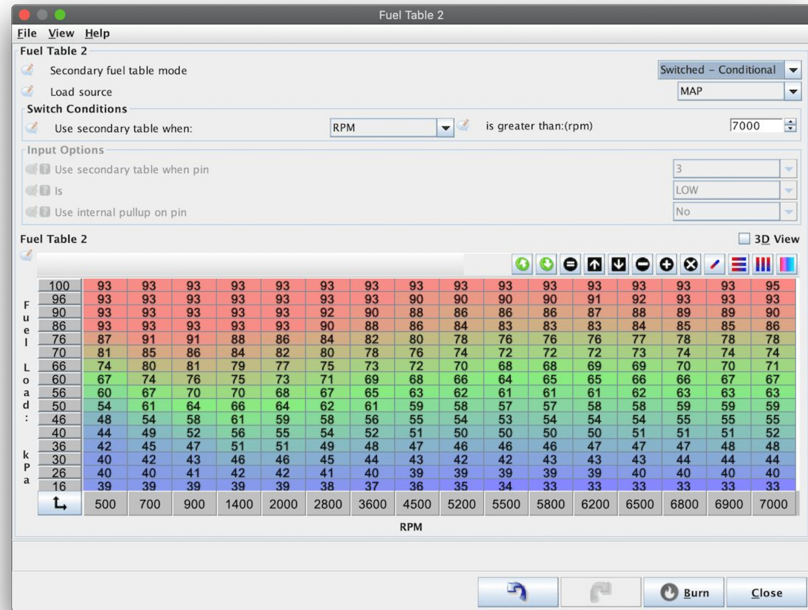
> **Warning:** Changing this value will require retuning of the fuel map!

- **Multiply by ratio of AFR to Target AFR:** This option is normally set to No for most setups. It allows basic closed loop feedback by adjusting the base fuel amount according to how faraway from the target AFR the engine is currently running (in %). If the AFR/O2 Sensor type is set to Disabled then this setting will have no impact on the fuel calculation.
- **Multiply by ratio of stoich AFR to target AFR ('IncorporateAFR')**: By enabling this setting AFR target is incorporated to pulse width calculation. This makes VE table a better representation of actual

VE, without AFR targets greatly affecting numbers. After VE table has been tuned, one can adjust an area richer or leaner just from AFR target table, basically without need to touch VE table.

> **Warning:** Changing this value will require retuning of the fuel map!

Secondary Fuel table



The ECU also has the ability to use a secondary fuel table which allows for blended and switched mode fueling. There are 2 blended modes and 2 switched modes available.

Blended fuel modes work in conjunction with the primary fuel table to come up with a single, combined VE. Switched fuel modes are where either the primary or secondary fuel table is used, but not both at the same time. Which table is being used at any given time can be configured based on either an external input (Ex dash switch) or set via certain conditions.

Multiplied %

This is a blended fuel mode (ie it uses both the primary and secondary fuel tables together) that allows for different load and RPM axis to be combined. Commonly this is used for having primary and secondary fuel tables with different load sources (**Ex:** Primary map using TPS and secondary map using manifold pressure).

This mode is often used on engines with Individual Throttle Bodies (ITBs) to allow TPS and MAP based tables to be combined.

The final fuel value is derived from treating both values (Primary and Secondary) as percentages and multiplying them together.

Example 1 • **Primary Fuel table value:** 75 • **Secondary fuel table value:** 100 • **Final value:** 75

Example 2 • Primary Fuel table value: 80 • **Secondary fuel table value:** 150 • **Final value:** 120

Example 3 • Primary Fuel table value: 90 • **Secondary fuel table value:** 80 • **Final value:** 72

Added

This is a blended fuel mode that is very similar to the above Multipled %mode. The only difference between the two is that instead of multiplying the values from the primary and secondary tables, the 2 are added together.

This is a less commonly used mode, but is an alternative in the same setups that you would use Multiplied %

Switched - Conditional

Conditional switched mode will allow use of the 2nd fuel table when a certain value goes above a defined level. The available switching values are:

- RPM
- Ethanol content
- MAP
- TPS

Depending on the desired outcome, this can be used to expand the resolution of the main fuel table, automatically handle alternate fuels or as an alternative ITB mode (Particularly if running boosted ITBs).

Switched - Input based

Input based switch mode let's you change the fuel table that is in use via an external input to the ECU. The options required are:

- The (Processor) pin that the input is connected to
- The polarity of this input (IE Is the secondary fuel table used with the signal is high or low). For a standard ground switching input, this should be LOW
- Whether to use the internal pullup on this input. For a standard ground switching input, this should be Yes

Acceleration Enrichment (AE)

Acceleration Enrichment (AE) is used to add extra fuel during the short transient period following a rapid increase in throttle. It performs much the same function as an accelerator pump on a carbureted engine, increasing the amount of fuel delivered until the manifold pressure reading adjusts based on the new load.

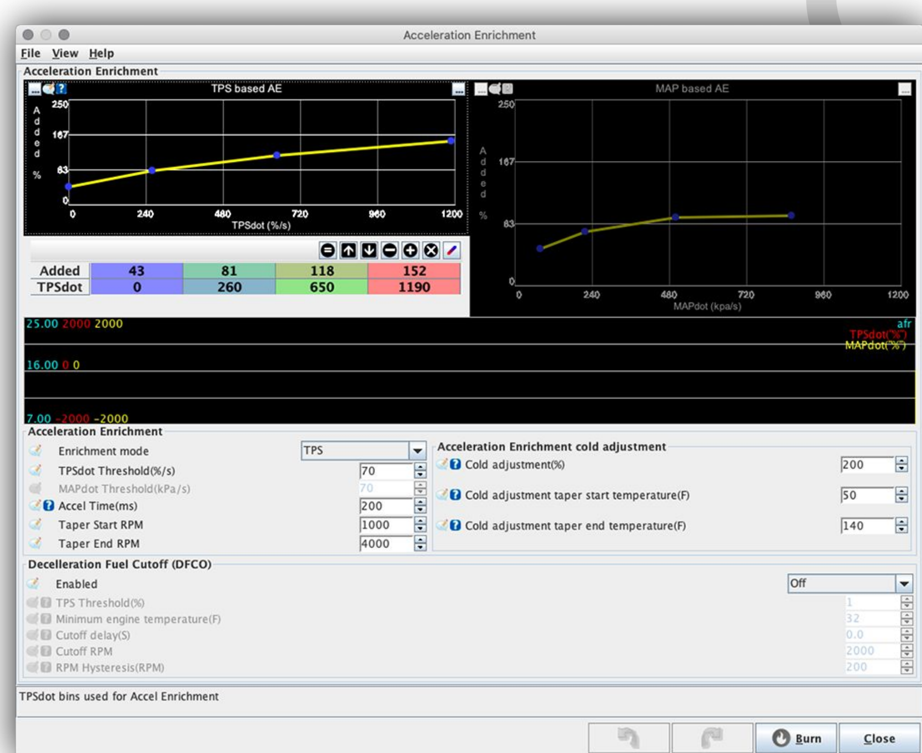
> To operate TPS based AE correctly, you must have a variable TPS installed and calibrated.

Theory

Tuning of acceleration enrichment is based on the rate of change of the throttle position, a variable known as TPSdot (TPS delta over time). This is measured in %/second, with higher values representing faster presses of the throttle and values in the range 50%/s to 1000%/s are normal. Eg:

- 100%/s = pressing the throttle from 0% to 100% in 1 second
- 1000%/s = pressing the throttle from 0% to 100% in 0.1s

TPSdot forms the X axis of the acceleration curve, with the Y axis value representing the % increase in fuel.



Tuning

The enrichment curve included with the base the ECU tune is a good starting point for most engines, but some adjustment is normal depending on injector size, throttle diameter etc.

In most cases, tuning of the AE curve can be performed in a stationary environment, though dyno or road tuning is also possible. Fast and slow blips of the throttle should be performed and the effect on the AFRs monitored using the live line graph on the AE dialog. This graph shows both TPSdot and AFR values in sync with each other, making adjustments to the correct part of the AE curve simpler to identify.

If you find that the AFR is initially good, but then goes briefly lean, you should increase the 'AccelTime' setting, with increments of 10-20ms recommended.

False triggering

In cases where the TPS signal is noisy, spikes in its reading may incorrectly trigger the acceleration enrichment. This can be seen in a log file or on a live dash in TunerStudio by the activation of the 'TPS Accel' indicator when there is no (or little) throttle movement occurring.

Should this occur (and assuming that the TPS wiring cannot be corrected to reduce noise) then the false triggers can be prevented from triggering AE by increasing the "TPSdot Threshold" value. This should be increased in increments of ~5%/s, pausing between each increase to observe whether AE is still being incorrectly activated.

Fields

- **EnrichmentMode** Chose whether to use Throttle Position Sensor or Manifold Absolute Pressure for acceleration enrichment.
- **TPSdot Threshold** Percentage of throttle position change per second required to trigger acceleration enrichment. For example, if set to 70, the throttle position must change at a rate of 70% per second for acceleration enrichment to become active.
- **MAPdotThreshold** Same as TPSdot Threshold, but applies when using MAP enrichment mode.
- **Accel Time** Duration of acceleration enrichment. Once enrichment is triggered, it will last this many milliseconds.
- **Taper Start RPM, Taper End RPM** Scales the enrichment taper at different RPMs. If RPM is less than or equal to Start RPM, enrichment will be 100% of the calculated enrichment value, based on the TPSdot (or MAPdot) value seen. If RPM is greater than or equal to End RPM, enrichment will be 0%. As RPM increases, the total amount of required enrichment decreases. Enrichment is scaled linearly in between these values.
- **ColdAdjustment** Scales the acceleration enrichment percentage linearly based on coolant temperature. At Start Temperature, adjustment will be equal to the Cold Adjustment field (%). At End Temperature, adjustment will be 0%.
- **Deceleration Fuel Cutoff** Stops injecting fuel when: *RPM is above **Cutoff RPM** TPS is below **TPS Threshold** Engine temperature is above **Minimum engine temperature** The above conditions are met for **Cutoff delay** seconds* ** RPM Hysteresis can be adjusted to account for fluctuating RPM conditions to prevent accidental DFCO.

AFR/O2 (Closed loop fuel)

AFR/O2 (for **A**ir:**F**uel **R**atio), dialog controls the closed loop fuel control, used for adjusting injector load based on input from an exhaust oxygen sensor (O2 sensor). In conjunction with the AFR Table, the closed loop AFR system will compare the actual O2 reading with the current target fuel ratio and make adjustments accordingly.

Use of a wideband sensor and controller is **strongly** recommended, however basic functionality is possible with a narrowband sensor if this is not available.

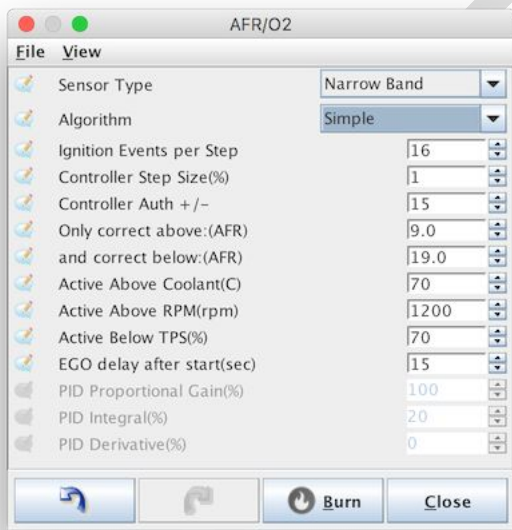
Note that closed loop fuel control is not a replacement for a poor tune. Many good configurations do not used closed loop control at all or only allow it very small adjustment authority.

Settings

The ECU supports 2 closed loop algorithms, each intended for different configurations:

1. **Simple** - A time based 'target chasing' algorithm where the amount of fuel adjustment is dependent on how long the reading has been lean or rich compared to the current target. This algorithm can work with widebands and with narrowband sensors where only basic rich/lean information is available. In particular, this algorithm performs poorly if you have a fuel map that is not close to complete. If you have this enabled and are seeing oscillations in the pulse width and/or AFRs, even when cruising, then you should disabled closed loop control until the base fuel MAP is better tuned.
2. **PID** - This is closed loop algorithm and will provide better results when combined with a wideband sensor and tuned correctly.

Common variables



- **Sensor type** - Narrowband or wideband, depending on hardware configuration. Narrowband sensor should be of the 0-1v type, wideband sensors should have a 0-5v signal. Wideband sensors need to be calibrated in the Tools->Calibrate AFR Table dialog
- **Algorithm** - See above for description of each algorithm available
- **Ignition events per step** - The AFR adjustment calculation will be performed every this many ignition cycles. Changes to closed loop adjustment typically have some lag before their impact is registered by the O2 sensor and increasing this value can take this lag into account. Typical values are 2-5.

- **Controller step size -**
- **Controller Auth** - The maximum % that the pulse width can be changed through this closed loop adjustment. Recommended value is no more than 20%.
- **Correctabove/belowAFR** - The AFR range that closed loop adjustments will be applied within.

This range is typically limited by the sensor and controller in use.

- **ActiveaboveCoolant** – Closed loop should only operate once engine is up to operating temperature. This value should be set to match the engines standard operating temp.
- **ActiveaboveRPM** – Closed loop adjustments should generally not be made at idle. Use this value to specify when adjustment should begin being made.
- **Active below TPS** - Above this TPS value, closed loops adjustments will be disabled
- **EGO delay after start** - All O2 sensors require a warmup period before their readings are valid. This varies based on the sensor in use, but 15s is a safe value in most cases.

PID only variables

- **P/I/D** - PID Proportional Gain, Integral and Derivative percentages.

These options are in addition to the Simple conditions and specify the parameters of the closed loop operation.

Limiters

The ECU includes a spark based rev limited with both hard and soft cuts.

The soft cut limiter will lock timing at an absolute value to slow further acceleration. If RPMs continue to climb and reach the hard cut limit, ignition events will cease until the RPM drop below this threshold.

> **Note** As this is spark based limiting, fuel only installs cannot use the rev limiter functionality

Settings

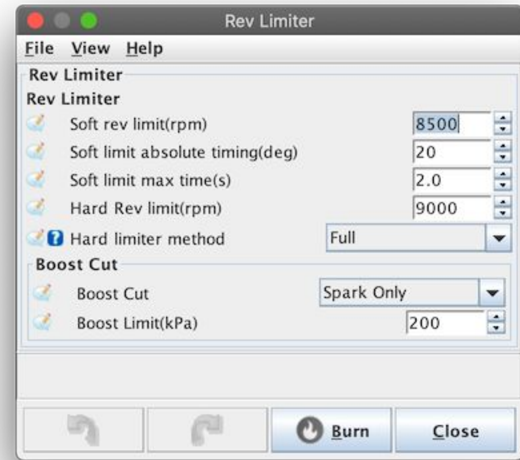


Figure 35: Rev limiter settings

- **Soft rev limit:** The RPM at which the soft cut ignition timing will be applied over.
- **Soft limit absolute timing:** Whilst the engine is over the soft limit RPM, the ignition advance will be held at this value. Lower values here will have a greater soft cut affect.
- **Soft limit max time:** The maximum number of seconds that the soft limiter will operate for. If the engine remains in the soft cut RPM region longer than this, the hard cut will be applied.
- **Hard rev limiter:** Above this RPM, all ignition events will cease.

Flex Fuel

Overview

The ECU has the ability to modify fuel and ignition settings based on the ethanol content of the fuel being used, a practice typically known as flex fueling. A flex fuel sensor is installed in the feed or return fuel lines and a signal wire is used as an input on the ECU board.

As ethanol is less energy dense, but also has a higher equivalent octane rating, adjustments to the fuel load and ignition timing are required.

Hardware

The ECU uses any of the standard GM/Continental Flex fuel sensors that are widely available and were used across a wide range of vehicles. These were available in 3 different units, all of which are functionally identical, with the main difference being only the physical size and connector. The part numbers for these are:

- Small - #13577429
- Mid-size - #13577379

- Wide - #13577394 (Same as the mid-size one, but with longer pipes)

All 3 use a variant of the Delphi GT150 series connector. You can use a generic GT150 connector, but you will have to clip off 2 tabs from the side of the sensor.

Part numbers:

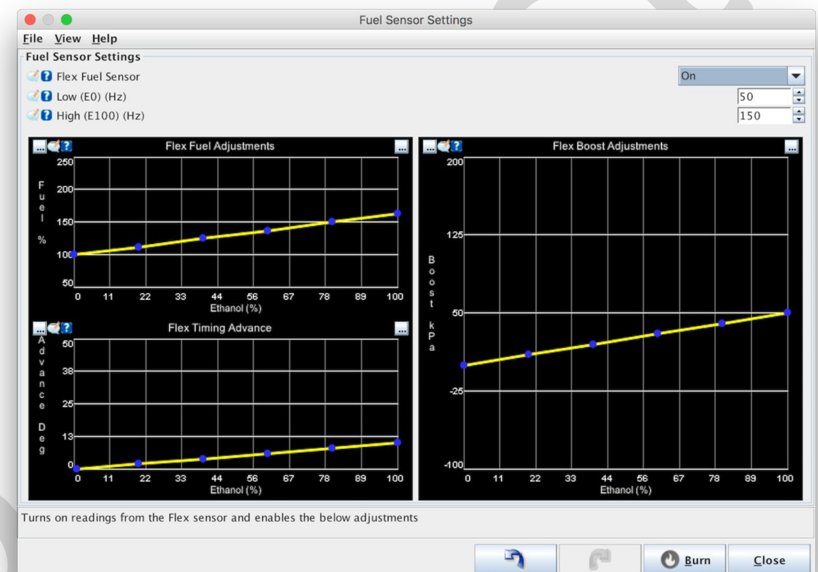
- Housing (#13519047)
- Pins (#15326427)
- Seal (#15366021)

Alternatively, there is a GM part for a harness connector, part number 13352241: http://www.gmpart sdirect.com/oe-gm/13352241

Wiring

All units are wired identically and have markings on the housing indicating what each pin is for (12v, ground and signal) The ECU normally has a pullup resistor on the signal, but please refer to the Quick Guide first.

Tuning



- **Sensor frequency -** The minimum and maximum frequency of the sensor that represent 0% and 100% ethanol respectively. For standard GM/Continental flex sensors, these values are 50 and 150
- **Fuelmultiplier%-** This is the additional fuel that should be added as ethanol content increases. The Low value on the left represents the adjustment to the fuel map at 0% ethanol and will typically be 100% if the base tune was performed with E0 fuel. If the base tune was made with E10 or E15 however, this value can be adjusted below 100%. The high value represents the fuel multiplier at

100% ethanol (E100) and the default value of 163% is based on the theoretical difference in energy density between E0 and E100. Tuning of this value may be required.

- **Additionaladvance-** The additional degrees of advance that will be applied as ethanol content increases. This amount increases linearly between the low and high values and is added after all other ignition modifiers have been applied.

Staged Injection

Overview

The ECU has the ability to control a secondary fuel stage for engines that have 2 sets of injectors, typically of different capacities. Whilst there are few stock engines that come with secondary injectors (the notable exception being many Mazda rotaries) secondary staged injection is a common modification, in particular used whenever large injectors are required, but where it is desirable to keep smaller injectors for smoother low RPM performance.

Hardware Configuration

The hardware configuration of the staging outputs depends greatly on the board in use, the engine itself and the fuel injector arrangement.

The table below outlines the number and channel configuration of the fuel channels required based on the cylinder count and fuel mode:

	1	**2**	**3**	**4**	**5**	**6**	**8**
Sequential	**Min Inj#:** 2**Pri:** 1**Sec:** 2	**Min Inj#:** 4**Pri:** 1/2**Sec:** 3/4	**Min Inj#:** 6**Pri:** 1/2/3**Sec:** 4/5/6	**Min Inj#:** 8**Pri:** 1/2/3/4**Sec:** 5/6/7/8	**Min Inj#:** 6**Pri:** 1/2/3/4/5**Sec:** 6	**Min Inj#:** 7**Pri:** 1/2/3/4/5/6**Sec:** 7	N/A
Other	As above	**Min Inj#:** 2**Pri:** 1**Sec:** 2	**Min Inj#:** 4**Pri:** 1/2/3**Sec:** 4	**Min Inj#:** 4**Pri:** 1/2**Sec:** 3/4	As above	**Min Inj#:** 6**Pri:** 1/2/3**Sec:** 4/5/6	**Min Inj#:** 8**Pri:** 1/2/3/4**Sec:** 5/6/7/8

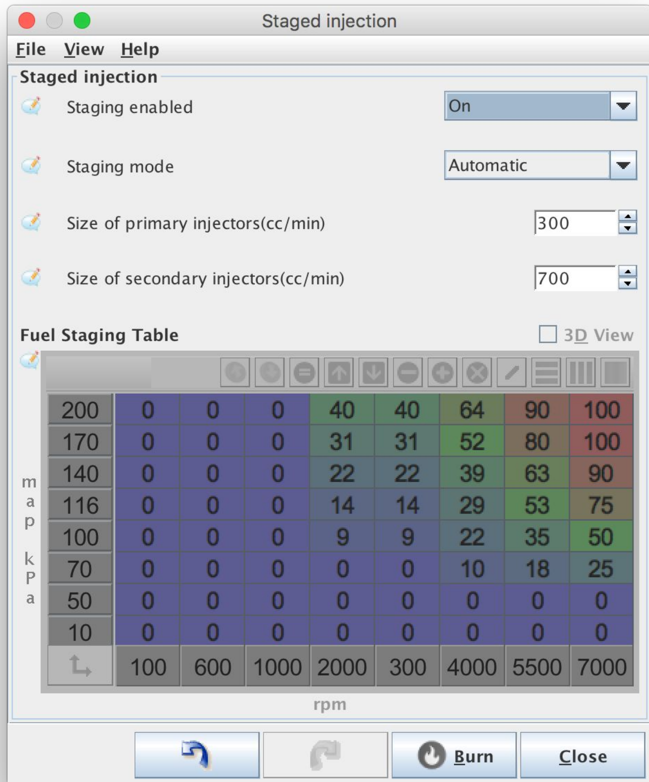
Configuration

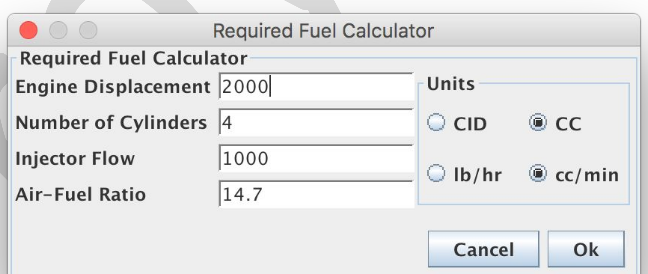
No matter which control strategy is chosen, you must enter the sizing of the primary and secondary injectors in order to allow The ECU to know the split in the overall fueling.

> **CRITICAL** - The req-Fuel value in the Engine Constants MUST be updated when staged injection is turned on. **When staging is in use, the value entered in the req_fuel calculator MUST be equal to the sum of both the primary and secondary injector sizes.** Failure to set these values correctly will result in excessive rich or lean conditions.

Ex:

- **Primary Injectors :** 300cc • **Secondary Injectors :** 700cc
    - **Value entered into the req_fuel calculator :** 1000cc





Control methods

The ECU provides 2 staging control modes, each with their own strengths and weaknesses. In most cases it is recommended to start with the Automatic mode, which only requires tuning of the standard VE table, and reviewing to see if you get the desired outcome. Only if this can't be tuned to give a satisfactory fuel split would changing to the manual table tuning be recommended.

Automatic staging When using the automatic staging method, the ECU takes into account the full capacity of the injectors (ie the sum of the 2 injector stages) and will perform a split of these itself. With this method, the user can simply tune the VE table in the same manner as if only a single set of injectors were used and the system takes care of the rest.

In this mode, The ECU will attempt to use the primary injectors up to their 'Injector Duty Limit' (As configured in the Injector Characteristics dialog. When staging is being used, it is recommended that this limit should be no higher than 85%. Once the primary injectors reach this duty limit, The ECU will begin to perform any further fueling from the secondary injectors. In this way, the VE table is all that is required for tuning as the system will take care of allocating the current fuel load to the best injectors.

Table control Table control allows the use of a manual 8x8 map that indicates what percentage of thefuelloadwillbeperformedbythe**secondary**injectors-0%=Secondaryinjectorsdisabled-100% = Primary injectors disabled

It is important to note that the values in this table do NOT correspond directly to the split of the duty cycle or pulse width. They represent the percentage of the total fuel load that the secondaries will be asked to perform. The affect this value has on the pulse width depends on the ratio of the primary and secondary injector capacities.

One disadvantage of the table tuning method is that it does not allow for the full fuel load of the primary and secondary injectors to be used simultaneously. As the table is a split of the total fuel load, as one set of injectors performs more, the other will perform less.

Wiring The wiring of injectors depends on the number of cylinders, the number of channels available on your ECU and whether you are using sequential fueling.

Example Assuming a 4 cylinder even fire engine, the injectors are to be wired in pairs.

Primary injectors on outputs 1 and 2. The secondary on outputs 3 and 4.

For other setups see the Hardware Configuration section above

Spark Settings

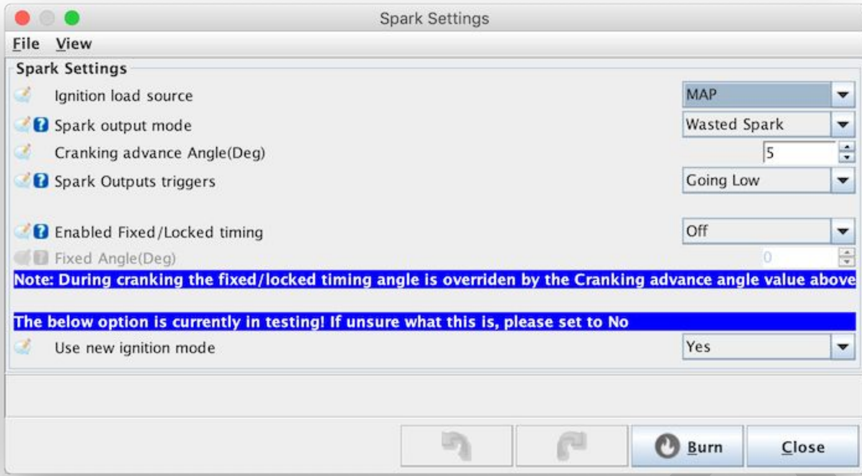
Overview

The Spark settings dialog contains the options for how the ignition outputs will function, including which of the IGN outputs are used and how. They are critical and incorrect values will result in an engine not starting and in some cases damage to hardware is possible. This dialog also contains a number of options for fixing the ignition timing for testing and diagnosis.

Please ensure you have reviewed these settings prior to attempting to start your engine.

The base map included will only be used to start the engine, and after staring to tune this map is extremely important, professional tune is highly recommended.

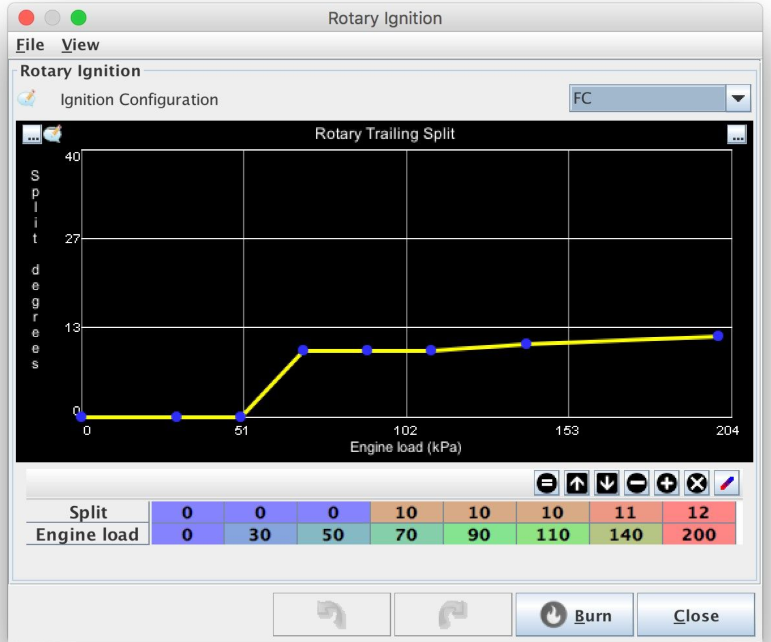
Settings



- **SparkOutputmode** - Determines how the ignition pulses will be outputted and is very specific to your ignition wiring. **Note that no matter which option is selected here, ignition signals ALWAYS fire in numerical order (ie1->2->3->4) up to the maximum number of outputs**. The firing order of the engine is accounted for in the wiring order.
  - **Wasted Spark** - Number of ignition outputs is equal to half the number of cylinders and each output will fire once every crank revolution. One spark will therefore take place during the compression stroke and the other on the exhaust stroke (aka the 'wasted' spark). This method is common on many 80s and 90s vehicles that came with specific wasted spark coils, but can also be used with individual coils that are wired in pairs. Wasted spark will function with only a crank angle reference (Ex a missing tooth crank wheel with no cam signal)
  - **SingleChannel** - This mode sends all ignition pulses to IGN1 output and is used when the engine contains a distributor (Typically with a single coil). The number of output pulses per (crank) revolution is equal to half the number of cylinders.
  - **WastedCOP** - This is a convenience mode that uses the same timing as the 'Wasted Spark' option, however each pulse is sent to 2 ignition outputs rather than one. These are paired IGN1/IGN3 and IGN2/IGN4 (ie When IGN1 is high, IGN3 will also be high). As this is still a wasted spark timing mode, only crank position is required and there will be 1 pulse per pair, per crank revolution. This mode can be useful in cases where there are 4 individual coils, but running full sequential is either not desired or not possible (Eg when no cam reference is available).
  - **Sequential** - This mode is only functional on engines with 4 or fewer cylinders.
  - **Rotary** - See below for full detail.
- **Crankingadvance** - The number of absolute degrees (BTDC) that the timing will be set to when cranking. This overrides all other timing advance modifiers during cranking.

- **Spark output triggers** - **THIS IS A CRITICAL SETTING!** Selecting the incorrect option here can cause damage to your igniters or coils. Specifies whether the coil will fire when the ignition output from the ECU goes HIGH or goes LOW. The VAST majority of ignition setups will require this to be set **GOING LOW** (ie the coil charges/dwells when the signal is high and will **fire** when that signal goes low). Whilst GOING LOW is required for most ignition setups, there are some configurations that perform the dwell timing on the ignition module and fire the coil only when they receive a HIGH signal from the ECU.
- **Fixed Angle** - This is used to lock the ignition timing to a specific angle for testing. Setting this to any value other than 0 will result in that exact angle being used (ie overriding any other settings) at all RPMs / load points, except during cranking (Cranking always uses the above Cranking Advance setting). This setting should be set to 0 for normal operation.

Rotary modes



The ECU supports the ignition configurations found on FC/FD RX7 and RX8 engines and this option becomes available when the Rotary ignition mode is selected above. The leading / trailing split angle can be set as a function of the current engine load.

- **FC** - Outputs are configured for the Leading/Trailing setup that was used on FC RX7s. Wiring is:
  - **IGN1** - Leading (wasted) sparks
  - **IGN2** - Trailing spark
  - **IGN3** - Trailing select
  - **IGN4** - Not used
- **FD** - Uses the same wasted spark signal for both leading sparks as FC, but individual signals for the trailing sparks. Wiring is:
  - **IGN1** - Leading (wasted) sparks
  - **IGN2** - Front rotor trailing
  - **IGN3** - Rear rotor trailing
  - **IGN4** - Not used
- **RX8** - Individual outputs are used for each spark signal. Wiring is:
  - **IGN1** - Front rotor leading
  - **IGN2** - Rear rotor leading
  - **IGN3** - Front rotor trailing **–** **IGN4** - Rear rotor trailing

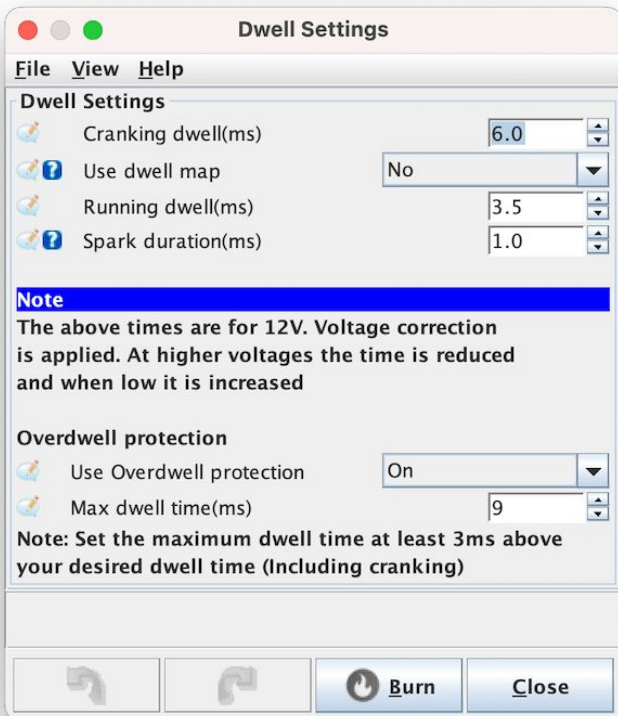
Dwell Control

Overview

The dwell control dialog alters the coil charging time (dwell) for The ECU's ignition outputs. Care should be taken with these settings as igniters and coils can be permanently damaged if dwelled for excessive periods of time.

From the April 2017 firmware onwards, dwell will automatically reduce when the configured duration is longer than the available time at the current RPM. This is common in single channel ignition configurations (Eg 1 coil with a distributor) and in particular on higher cylinder count engines.

Settings



Note: Both the running and cranking dwell times are nominal values, assumed to be at a constant voltage (Usually 12v). Actual dwell time used will depend on the current system voltage with higher voltages having lower dwell times and vice versa. See section below on voltage correction.

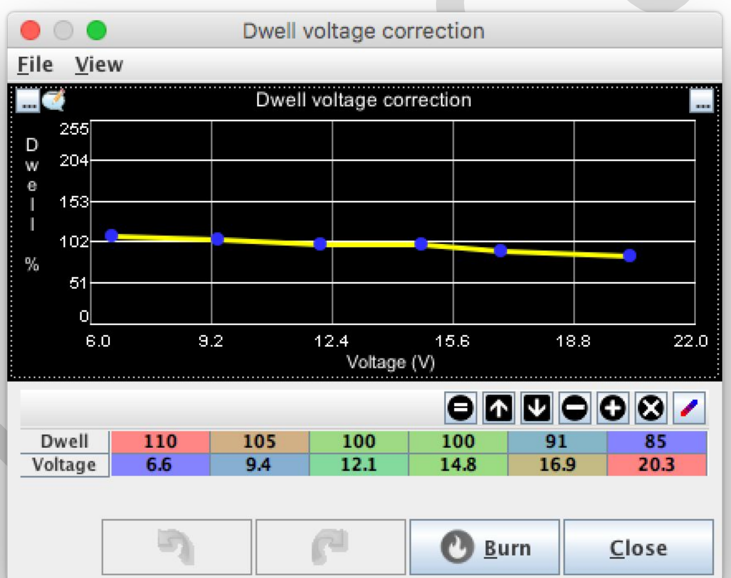
- **Crankingdwell**-The nominal dwell time that will be used during cranking. Cranking is defined as being whenever the RPM is above 0, but below the 'Cranking RPM' values in the Cranking dialog
- **Use dwell map** - By default this is set to "No" and The ECU will use fixed running dwell value (With a voltage correction applied). If different dwell values are required across engine RPM/load range, this can be set to Yes and separate Dwell table defines running dwell value.
- **Running dwell** - The nominal dwell that will be used when the engine is running normally.

- **Sparkduration** - The approximate time the coil takes to fully discharge. This time is used in calculating a reduced dwell when in time limited conditions, such as mentioned above on single coil, high cylinder count engines. The limited dwell time is calculated by taking the maximum revolution time at the given RPM, dividing by the number of spark outputs required per revolution and subtracting the spark duration. Outside of those conditions, this setting is not used.
- **Over dwell protection** - The over dwell protection system runs independently of the standard ignition schedules and monitors the time that each ignition output has been active. If the active time exceeds this amount, the output will be ended to prevent damage to coils. This value should typically be at least 3ms higher than the nominal dwell times configured above in order to allow overhead for voltage correction.

Voltage correction

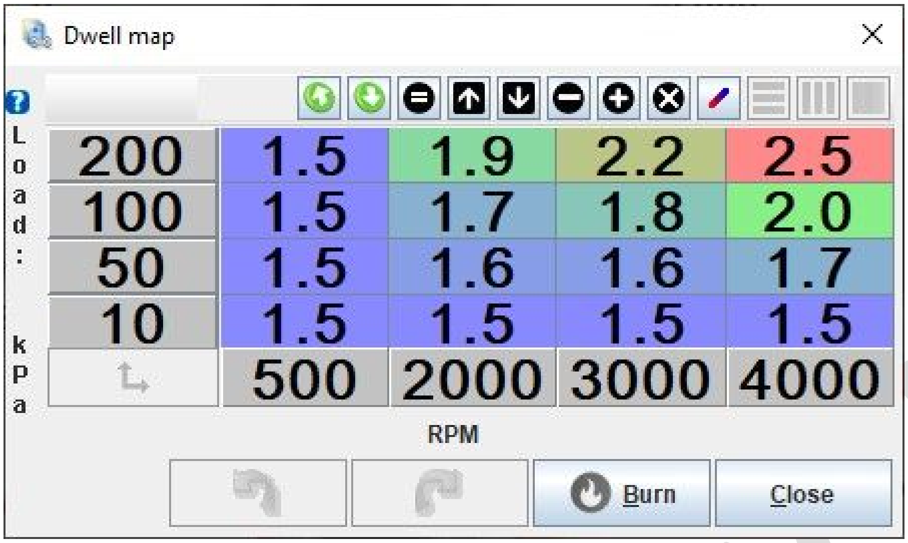
As the system voltage rises and falls, the dwell time needs to reduce and increase respectively. This allows for a consistent spark strength without damaging the coil/s during high system voltage conditions. It is recommended that 12v be used as the 'nominal' voltage, meaning that the Dwell % figure at 12v should be 100%.

The correction curve in the base tune file should be suitable for most coils/igniters, but can be altered if required.



Dwell map

If "Use dwell map" is set to "Yes" at dwell settings, this map will be available to allow for variable Running dwell based on ignition load and RPM values. The voltage correction will be applied on top of these map values.

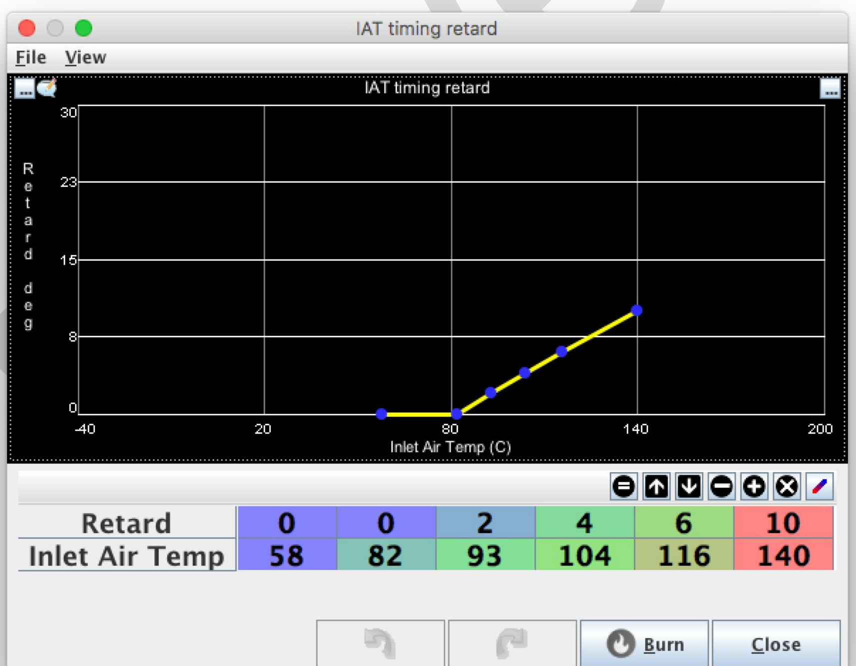


Temperature based timing changes

Changes in Inlet Air Temperature (IAT), in particular significant increases whilst under boost, can require ignition timing to be pulled. The IAT retard settings allow for this timing adjustment.

Example

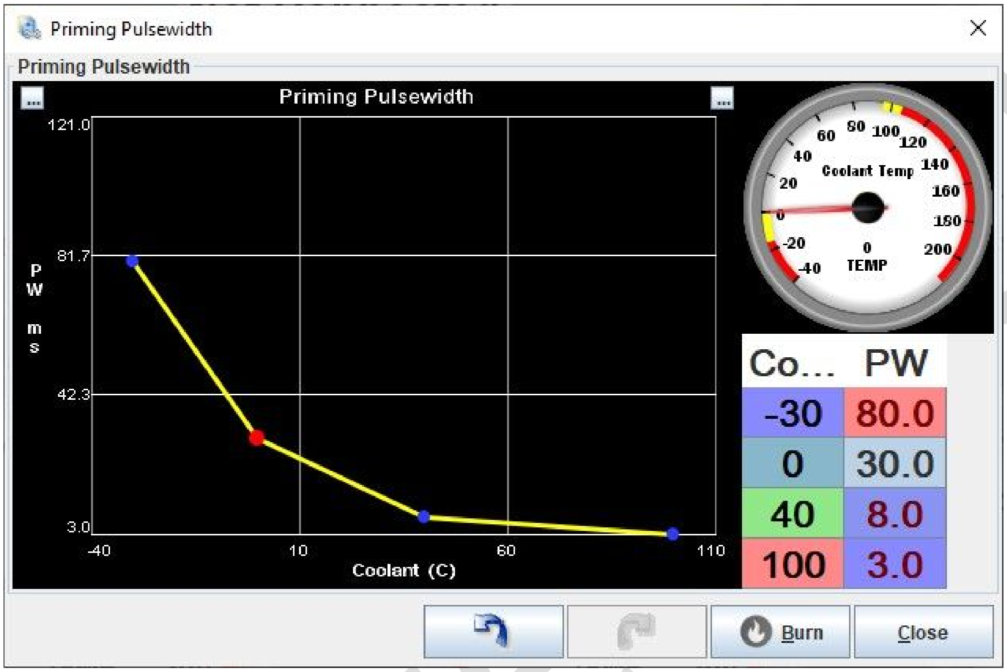
Exact settings will be engine dependent, but pulling of ignition timing beyond 100*C is a common scenario.



Priming pulsewidth

Priming Pulsewidth - Upon power up, The ECU will fire all injectors for this period of time. This pulse can be used to clear out air that may have entered the fuel lines or help the engine start easier by providing engine with fuel before it's cranked over.

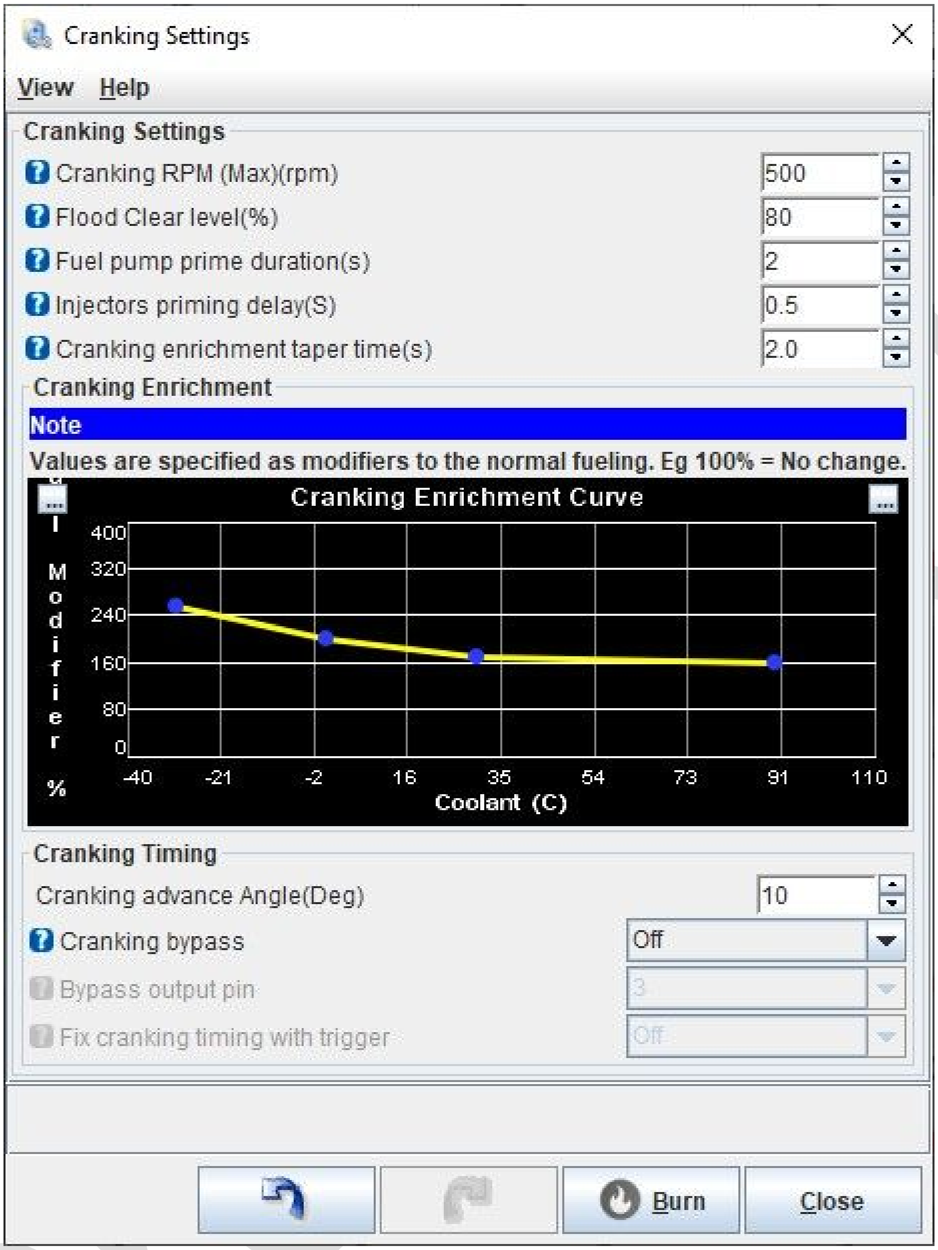
Usually priming pulsewidth is kept short but especially with low vaporization fuels (e85 etc.) longer priming pulse widths are required for easy starting of the engine. Regardless of what fuel is used, keep this value as low as possible to avoid flooding the engine. Start tuning from low priming pulsewidths and try longer pulsewidths until the engine starts easiest. Usually lower engine temps require longer priming pulsewidths.



Overview

Cranking conditions during starting typically requires multiple adjustments to both fuel and ignition control in order to provide smooth and fast starts. The settings on this dialog dictate when the ECU will consider the engine to be in a cranking/starting condition and what adjustments should be applied during this time.

Settings



- **Cranking RPM (Max)** - This sets the threshold for whether The ECU will set its status to be cranking or running. Any RPM above 0 and below this value will be considered cranking and all cranking related adjustments will be applied. It's generally best to set this to be around 100rpm higher than your typical cranking speed to account for spikes and to provide a smoother transition to normal idle.
- **Flood Clear level** - Flood clear is used to assist in removing excess fuel that has entered the cylinder/s. Whilst flood clear is active, all fuel and ignition events will be stopped, and the engine can be cranked for a few seconds without risk of starting or further flooding. To trigger flood clear, the RPM must be **below** the above Cranking RPM setting and the TPS must be **above** the threshold of this setting.
- **Fuel pump prime duration** - When the ECU is first powered on, the fuel pump output will be engaged for this many seconds in order to pressurize the fuel system. If the engine is started in this time, the

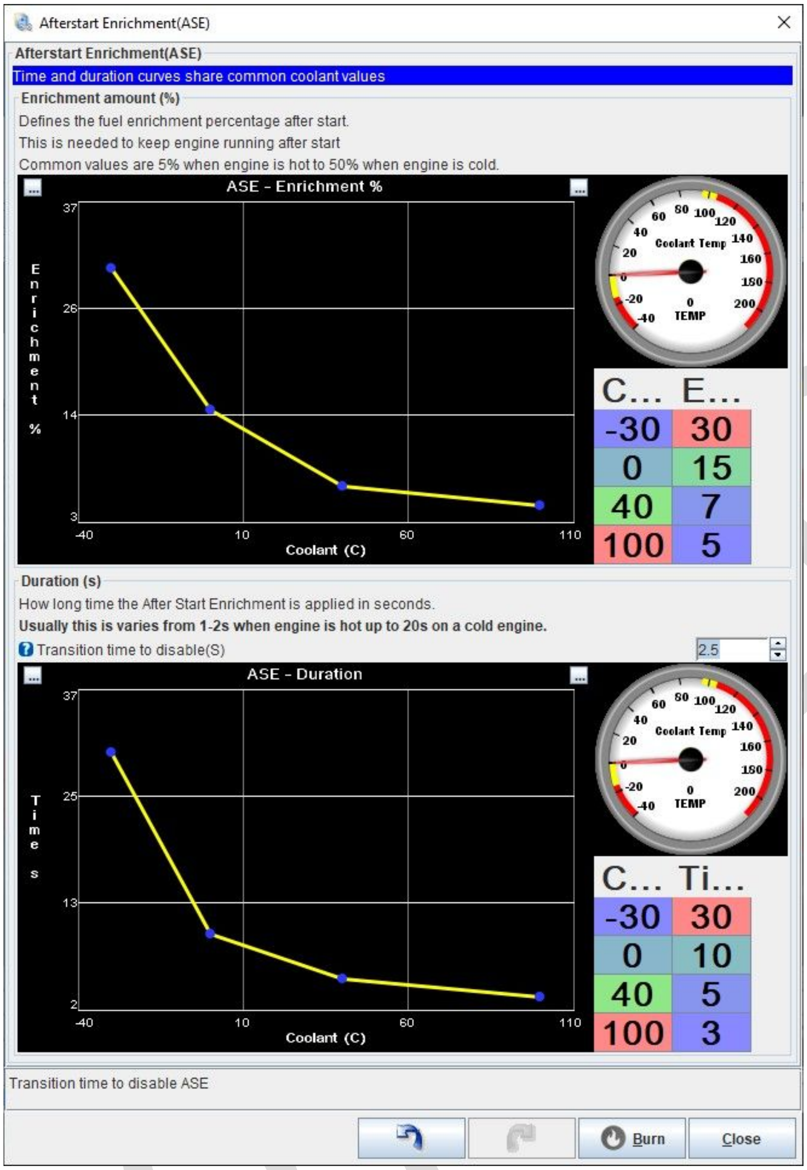
pump will simply keep running, otherwise it will be turned off after this period of time. Note that fuel pump priming only occurs at system power on time. If you have USB connected, The ECU remains powered on even without a 12v signal.

- **Injectors priming delay** - Upon power up, The ECU will fire all injectors for short period of time. (See Priming Pulsewidth) This setting sets the delay to priming after fuel pump is on and is used to wait for fuel line to get pressurized correctly.
- **Cranking enrichment taper time** - Taper time from cranking enrichment to ASE or run (after engine has started).
- **Crankingenrichment**-Whilst cranking is active (See Cranking RPM above), the fuel load will be increased by this amount. Note that as a standard correction value, this cranking enrichment **is in addition** to any other adjustments that are currently active. This includes the warmup enrichment etc.
- **CrankingadvanceAngle** - Whilst cranking the ignition advance from the spark table is ignored and engine uses this ignition advance value instead.
- **CrankingBypass**-This option is specifically for ignition systems that have a hardware cranking ignition option. These systems were used throughout the 80s and early 90s and allowed ignition timing to be fixed and controlled by the ignition system itself. Once the engine is determined to be running (via the cranking RPM setting) the output is raised HIGH to enable ECU timing control. With this option you can specify an output pin that will be set HIGH when the engine is running.
  The pin number specified is the PROCESSOR pin number.
- **Fixcrankingtimingwithtrigger** - Some (usually low resolution) trigger patterns are designed to align one of their pulses with the desired cranking advance. This is typically 5 or 10 degrees BTDC. When enabled, The ECU will wait for this timed input pulse before firing the relevant ignition output (A dwell safety factor is still applied incase this pulse is not detected). This option is only made available when a trigger pattern that supports this function is selected (See Trigger Setup).

Overview

Afterstart Enrichment (ASE) is a separate fuel modifier that operates over and above the WUE for a fixed period of time, after the engine first starts. Typically, this is a few seconds long period where a small enrichment can help the engine transition smoothly from cranking to idling.

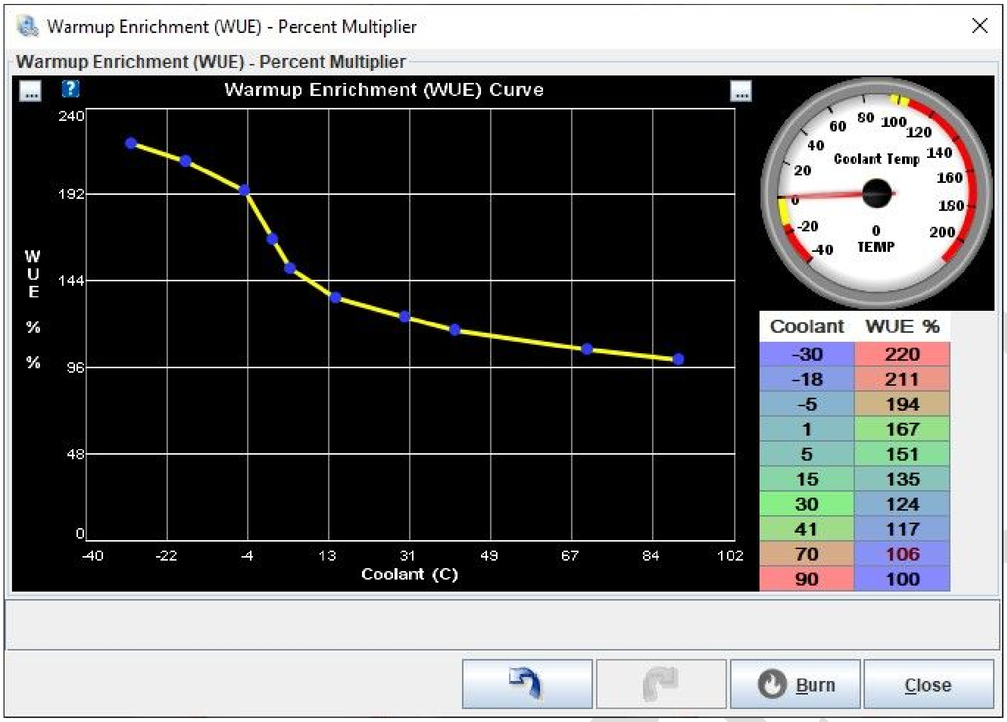
Settings



- **ASE - Enrichment %** - This curve sets the amount of enrichment during ASE period in percentage based on coolant temp. Typically 50% enrichment is required with cold engine and 5% with warm engine.
- **Transition time to disable** - After the ASE duration has passed, the enrichment amount will taper to zero smoothly to avoid sudden changes to AFR. This sets the time for how long the taper to zero will be. Typically few seconds.
- **ASE-Duration**-This curve sets the how long the ASE is applied in seconds. Typically1-2seconds is enough when engine is hot and 20 seconds when engine is cold.

Warmup curve

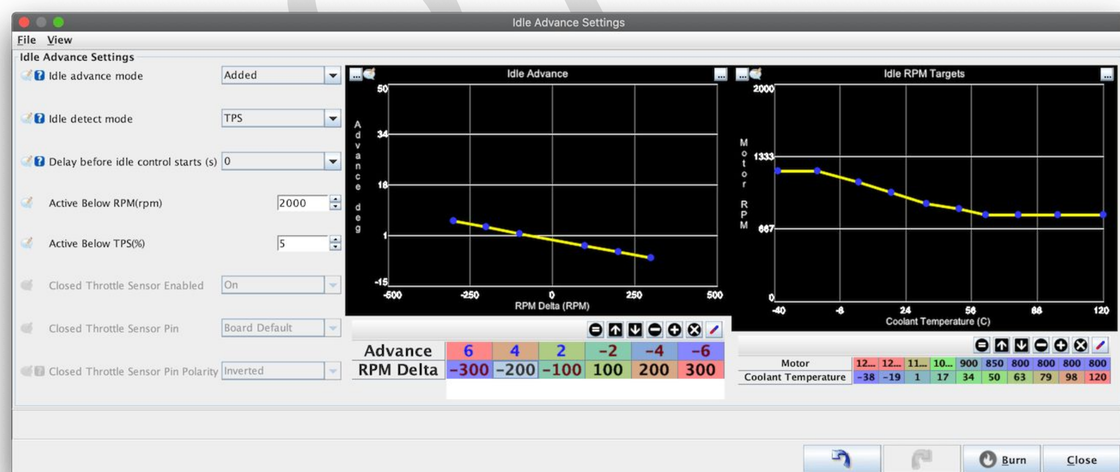
The Warm Up Enrichment (WUE) curve represents the additional fuel amount to be added whilst the engine is coming up to temperature (Based on the coolant sensor). The final value in this curve should represent the normal running temperature of the engine and have a value of 100% (Representing no modification of the fuel from that point onwards).



Idle advance control

Idle speed can be controlled without the use of an idle valve (IACV) by adjusting timing. This feature references the same idle RPM target curve that is used by the closed loop idle control and will then adjust the advance based on the error between current and target RPM.

Settings



- **Idle advance mode**
  - **Added** - This is the most common mode and will alter the regular advance amount by adding (or subtracting) a certain number of degrees based on the amount of RPM delta (Between target and actual RPMs).
  - **Switched** - The ignition advance will switch to the values in the idle advance curve rather than adjusting the normal advance values.
- **Idle detect mode** - This setting specifies how the ECU determines whether it is at idle or not. Most commonly this is based on a variable TPS and a specific TPS%, but if a closed throttle switch (CTPS) is available, this may be used instead.
- **Delay before idle control** - This allows the idle RPM to settle during deceleration before the ignition advance is changed.
- **Active below** - Maximum RPM that the idle advance control will be active under
- **Active Below** - If the idle detect mode is set to TPS, this is the throttle position that the control will be active below.
- The following 3 settings are only used if idle detection uses a CTPS input:
  - **CTPS enabled** - Whether to use a CTPS input.
  - **CTPS Pin** - The Processor pin that the CTPS is connected to.
  - **CTPS Polarity** - Whether idle is indicated by the input being pulled to ground (Normal) or indicated by the input being pulled to 5v (Inverted). In Normal mode, the internal pullup will be enabled.

Idle Advance curve

This curve specifies the amount of timing adjustment (Added mode) or the absolute advance amount (Switched mode) that will be used based on the delta (error) from target RPM.

The RPM delta is equal to: [Idle Target RPM] - [Current RPM]

Generally timing will be added (positive values) in order to try and increase RPM and timing will be removed (Negative values) to reduce RPM.

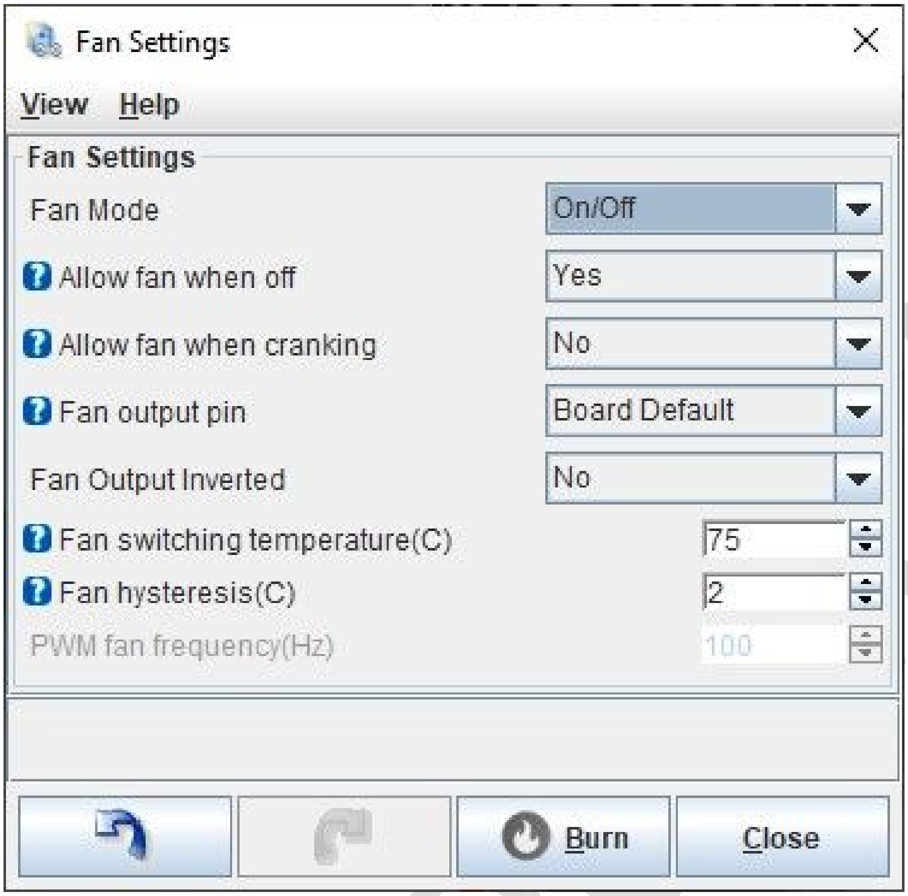
Idle RPM target curve

This curve specifies what the desired idle RPM is based on the current coolant temperature. This table is shared with the idle air control if that is being used in conjunction with idle advance control.

Thermo fan

Control of a cooling (thermo) fan is available through the Thermo fan dialog.

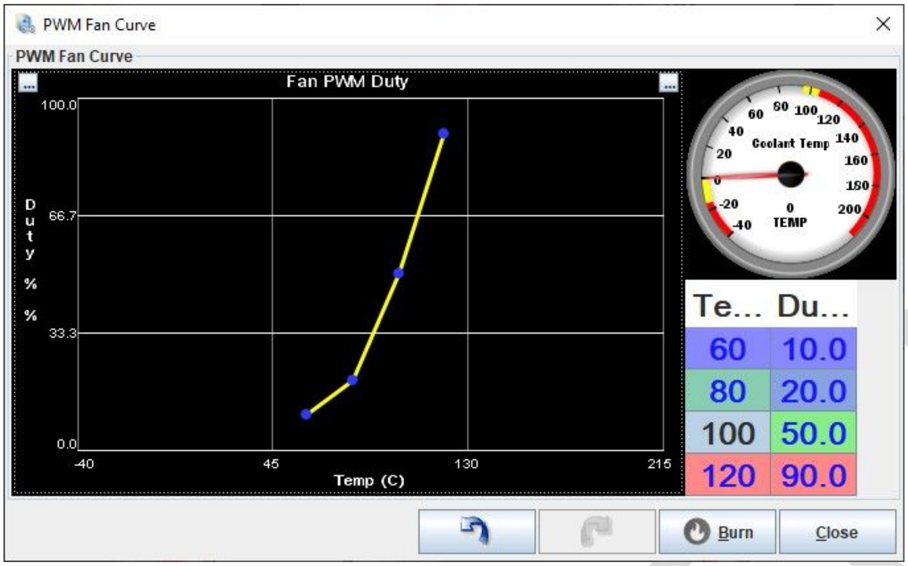
Settings



- **Fan mode** - On/Off or PWM. Set this to Off if fan output is not used (PWM only available in separate module).
- **Allow fan when off** - Whether the fan will run when the engine is not running.
- **Allow fan when cranking** - Whether the fan will run when engine is cranking.
- **Output pin** - The processor pin that the fan control will use. In most cases this should be left as Board Default
- **Output inverted** - Most setups will use No for this setting, but if you have a fan circuit that flips the output, the polarity can be reversed with this setting.
- **Fan switching temperature** - The temperature above which the fan will be turned on.
- **Fanhysteresis** - The number of degrees below the fan set point that fan will be turned off. This is used to avoid oscillation around the set point resulting in the fan turning on and off rapidly.
- **PWMFanfrequency** - Sets the PWM fan output frequency. See the fan controller specifications for the correct frequency.

> The ECU fan output is control signal only. Not capable of driving fan motor directly. So relay is required to turn fan on and off or separate fan controller in case of PWM fan.

PWM Fan Curve



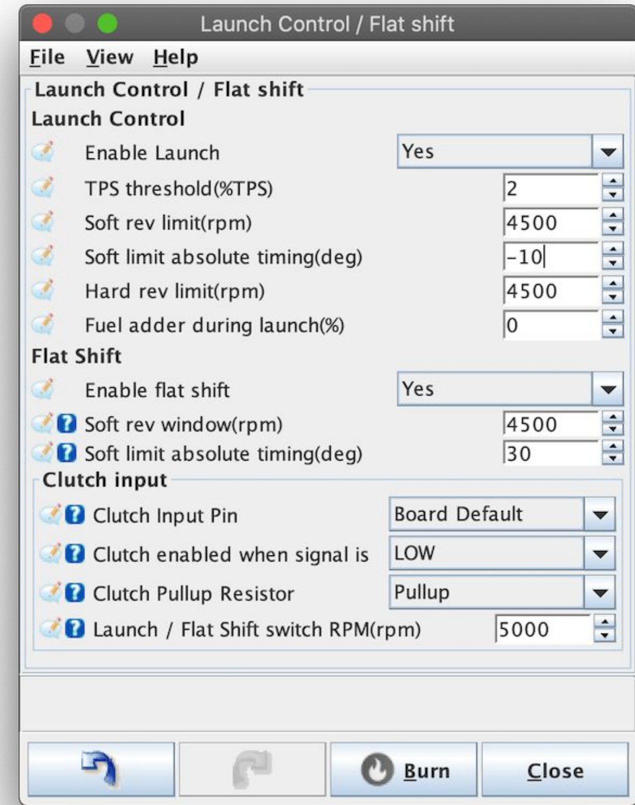
PWM Fan Curve sets the fan duty based on engine coolant temp. Duty range is 0-100%, but note that depending on the fan controller, the duty range can be different. In example 10-90% or the fan controller will go to fault state. See the fan controller specifications for the valid range.

Launch Control & Flat Shift

The ECU features a 2-step launch control combined with a flat shift feature. These are each dependent on a clutch switch (Usually a ground switching type) being wired in.

Setup

Both the 2-step and flatshift modes have hard and soft cut states. When under soft cut, the ignition timing will be altered to reduce the RPM acceleration, though this is generally not sufficient to stop or limit RPM rising. Under hard cut, the ignition signal is stopped completely until the RPMs drop.



Launch

- **TPSThreshold** – A minimum value for the launch engagement. The limiter will only be engaged above this RPM. Typical values are 1%-3% TPS, depending on how much noise is on your signal • **Soft rev limit** - The RPM at which the timing will be adjusted to slow RPM increase.
- **Soft limit absolute timing** - The **absolute** timing that will be used once the soft RPM limit is reached. This overrides all other timing adjustments at this time.
- **Hard rev limit** - The RPM at which the ignition signal will be cut entirely.
- **Fuel adder during launch** - A percentage modifier to the current pulse width to add extra fuel when launch (soft or hard) is active. This can aid in building boost on turbo setups at launch.
  time

Flat shift

- **Soft rev window** - This is an RPM window below the Launch / Flat shift **switch** RPM point during which an alternative timing will be applied. Typicalvaluesare100to1000rpm.

- **Soft limit absolute timing** - The absolute timing that will be used when in the flat shift soft RPM window.

Clutch settings

Both launch and flat shift require a clutch input in order to activate. This is generally a ground active type switch attached behind the clutch pedal.

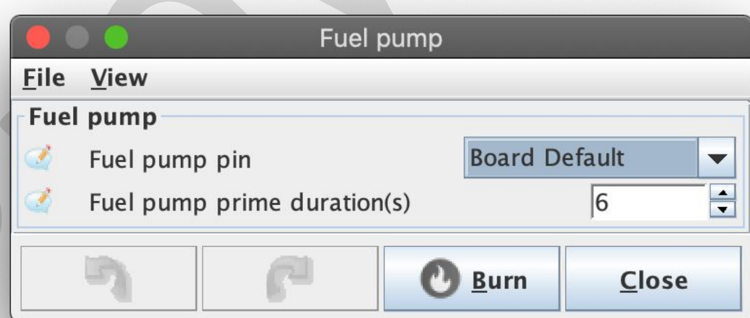
- **Clutch input pin** - The Processor pin that the switch is wired to. Most setups should leave this as the Board Default.
- **Clutch enabled when switch is** - The polarity of the clutch input. Typically this should be set to LOW for a switch that connects to ground when activated.
- **Clutch pullup resistor** - Whether the internal pullup will be enabled on this input. Typically this should be set to Open for 2.0 ECUs and Pullup for 1.0.
- **Launch / Flat shift switch RPM** - The ECU will use the RPM point the clutch is engaged at to determine whether it is in launch mode or flat shift. If the clutch is pressed above this RPM value, it will be assumed to be a flat shift, below it will be considered a launch.

> The engagement point of the clutch switch can make a significant difference in the application of launch control. The switch should trigger as close to the clutches take up point as possible for the fastest response.

Fuel pump

Fuel pump control is a simple but important function performed by the ECU. Currently The ECU does not perform variable (PWM) pump control. Can only be connected to a relay. DO NOT CONNECT DIRECTLY TO FUEL PUMP.

Settings



- **Fuel pump pin** - The Processor pin that the fuel pump output is on. In most cases this should be left to Board Default unless you have a specific reason to change this.

- **Primeduration** - How long (In seconds) the fuel pump should run when the system is first powered up. Note that this is triggered **when the ECU is powered on**, which will not always be the same as when the ignition is turned out.

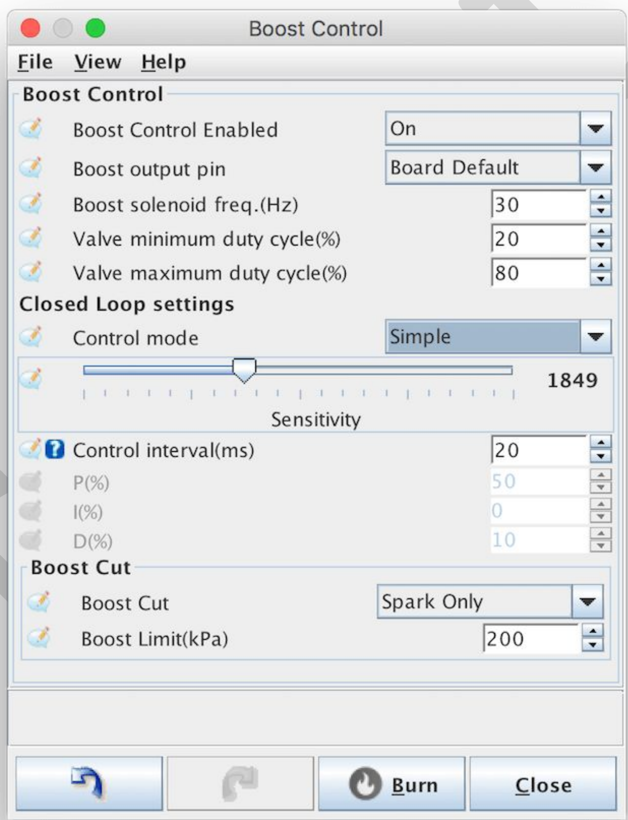
Boost Control

The ECU has an on board closed loop boost controller than can be used to regulate standard single turbo setups.

Most 3 or 4 port boost solenoids can be used, with frequencies between 15Hz and 500Hz supported. Any of the on board high current outputs can be directly connected to the solenoid and is controlled via a boost target table and PID tuning. Over boost limiting is also available.

Settings

The ECU's boost control uses a PID algorithm with 2 modes of operation, Simple and Full. Each has their own advantages and disadvantages, as outlined below

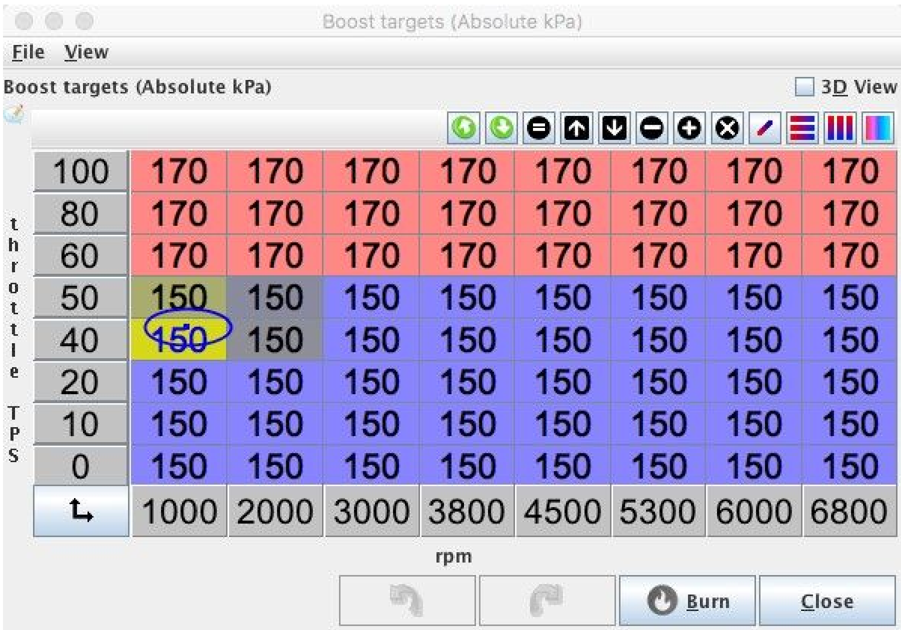


In Simple mode, the PID values themselves are controlled by the ECU itself and a sensitivity slider is used to adjust how aggressive the output duty cycle will be set. The simple mode can be easy and fast to setup, however has the downside that to avoid overboost, the sensitivity may need to be set low, which can increase lag.

Boost Cut

Boost Cut is a safety setting that will cut engine cycles (fuel, spark or both) if the boost level exceeds a certain figure.

Target table



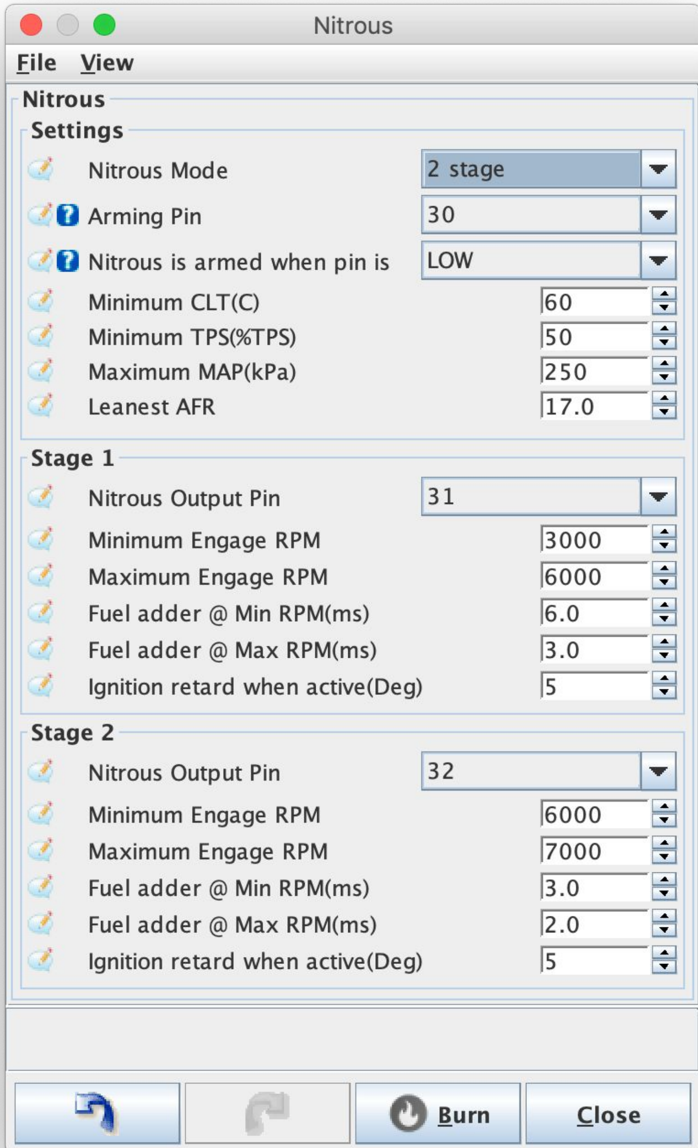
throttle TPS								
100	170	170	170	170	170	170	170	170
80	170	170	170	170	170	170	170	170
60	170	170	170	170	170	170	170	170
50	150	150	150	150	150	150	150	150
40	150	150	150	150	150	150	150	150
20	150	150	150	150	150	150	150	150
10	150	150	150	150	150	150	150	150
0	150	150	150	150	150	150	150	150
rpm	1000	2000	3000	3800	4500	5300	6000	6800

The boost map function varies depending on whether open or closed loop boost control has been selected.

- In closed loop mode, this map serves as a target table. The values in the map are the desired boost pressures (in kPa). In closed loop mode, these target values can optionally be modified by a flex fuel value if available.
- In open loop mode, the map values are the duty cycle percents that will be used.

Nitrous Control

The ECU contains a 2 step nitrous control system for controlling valves and making fueling adjustments for dry setups. The 2 stages operate independently and can overlap (ie both run at the same time) if needed.



Activation Settings

- **Nitrous Mode:** Whether 1 or 2 stages will be used
- **Arming Pin:** The Processor Pin to be used for arming the nitrous control.
- **Arming pin polarity:** What pin state is considered to be armed. Generally this will be LOW for a ground switching input
- **Minimum CLT:** The minimum coolant temperature that the stages will activate at
- **Minimum TPS:** The minimum TPS that the stage will activate at
- **MaximumMAP:** A protection to ensure that the nitrous will not activate above a certain level of boost
- **Leanest AFR:** Nitrous will only activate if the AFR is (And remains) below this value

Stage Settings

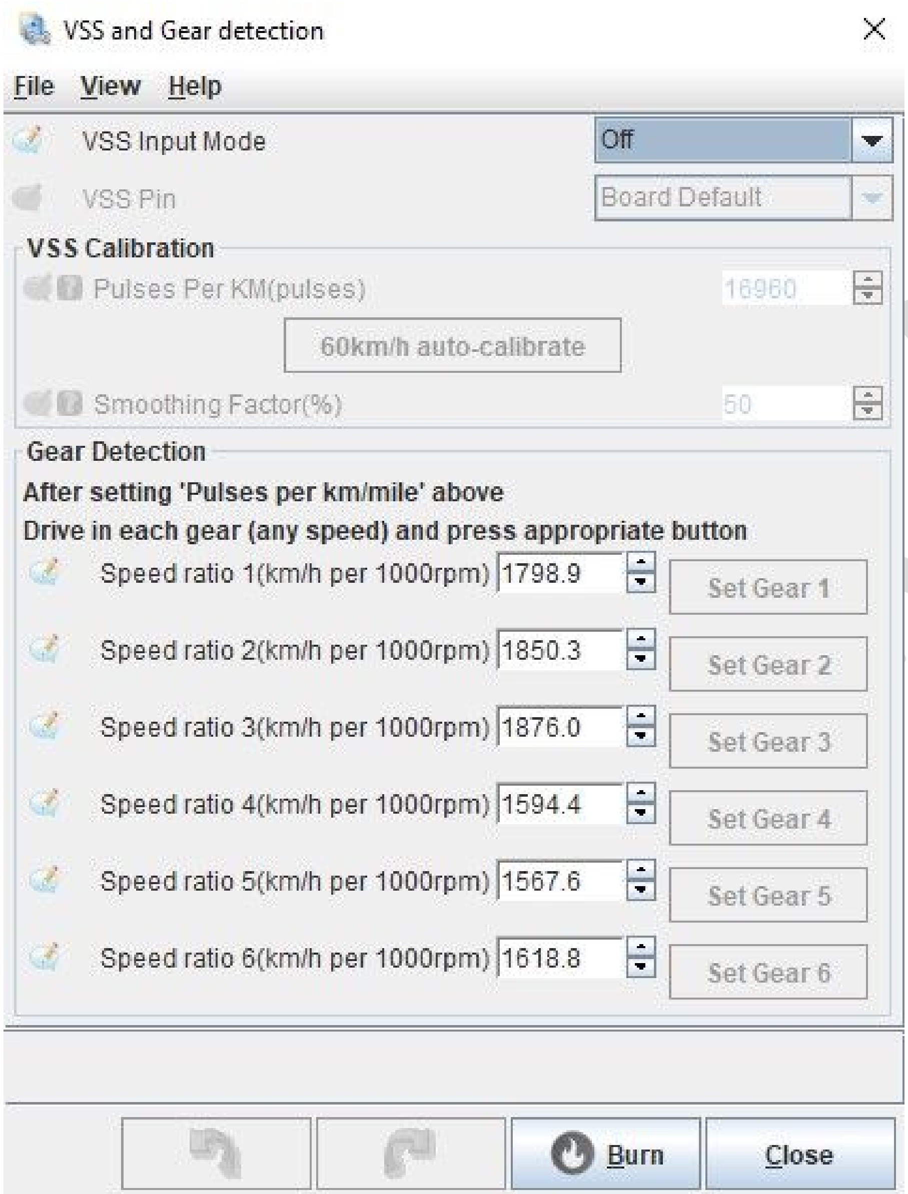
The settings for each stage are identical and allow for the 2 stages to run individually or joinly overlapping during a given RPM window.

- **Nitrous output Pin:** The (Processor) pin that will be driven high when the stage is active.
- **Minimum Engage RPM:** The RPM at which the stage will begin
- **Maximum Engage RPM:** The RPM at which the stage ends
- **Fuel adder @ Min/Max RPM:** The amount of fuel to be added at the minimum and maximum RPM points.
  - The fuel adder amount will be scaled between these 2 values as the RPMs rise
  - A calculator for these fuel adder values can be found at: https://bit.ly/3a0e9WU
- **Ignition retard when active:** An ignition modifier to be used to retard timing when the stage is active
  - Note that the retard values are cumulative. If both stages are active then the total retard amount will be the sum of both stages.

VSS and Gear Detection

The ECU includes Vehicle Speed Sensing option that senses speed by measuring pulses in the ECU input. Other VSS input options aren't yet supported.

Settings



- **VSSinputmode** – Select between “Pulses per KM” or “Pulses per mile” depending on which one is preferred. If VSS is not used, set this to Off.
- **VSS Pin** - Select what input pin is used for VSS signal. Only HS type pins will work for this function.
- **PulsesPerKM(pulses)** - You can set manually how many pulses at VSS input equal one kilometer travel distance. Or you can drive speed of 60km/h and click “60km/h auto-calibrate” button to set pulses per km setting automatically.
- **SmoothingFactor(%)** – A smoothing factor to help reduce noise in the VSS signal. Typical values are between 0 and 50 ### Gear Detection > Gear detection should only be calibrated once VSS is working correctly and should be done with a passenger for safety!.

Once VSS is working accurately, gear detection can also be configured. To calibrate this: 1. Place car in 1st gear and begin driving 2. Once RPM reaches approx. 2500 in this gear, press the Set Gear 1 button 3. Repeat above steps for each gear (Pressing the appropriate button each time)

Variable Valve Timing (VVT)

The ECU has an on board VVT controller than can be used to regulate one or two camshafts. VVT output can adjust valve timing or lift usually by controlling solenoid that uses oil pressure to change cam timing/lift.

Supported VVT modes are On/Off, open loop PWM and closed loop PWM.

VVT modes

On/Off

In On/Off mode the VVT output is either On or Off depending on the load and RPM. This is suitable control mode for simple VVT systems in older engines. MAP or TPS can be selected as load source. VVT control table is used to define when VVT output is on or off. Value 100 on the table defines that output is on and any other value sets the output off. For simplicity it’s recommended to use values 100 and 0 in the VVT control table to represent on and off (0% duty and 100% duty). This mode can be used for example in BMW single vanos engines and Honda VTEC engines.

Open loop PWM

In Open loop PWM mode the VVT output uses Pulse Width Modulation to adjust the cam timing. MAP or TPS can be selected as load source and also output frequency is selectable. Output duty is defined bytheVVTcontroltablesothatvalueonthemapisdirectlytheVVToutputduty. VVToutputhas0.5% duty accuracy and the available frequency range is 10-510Hz

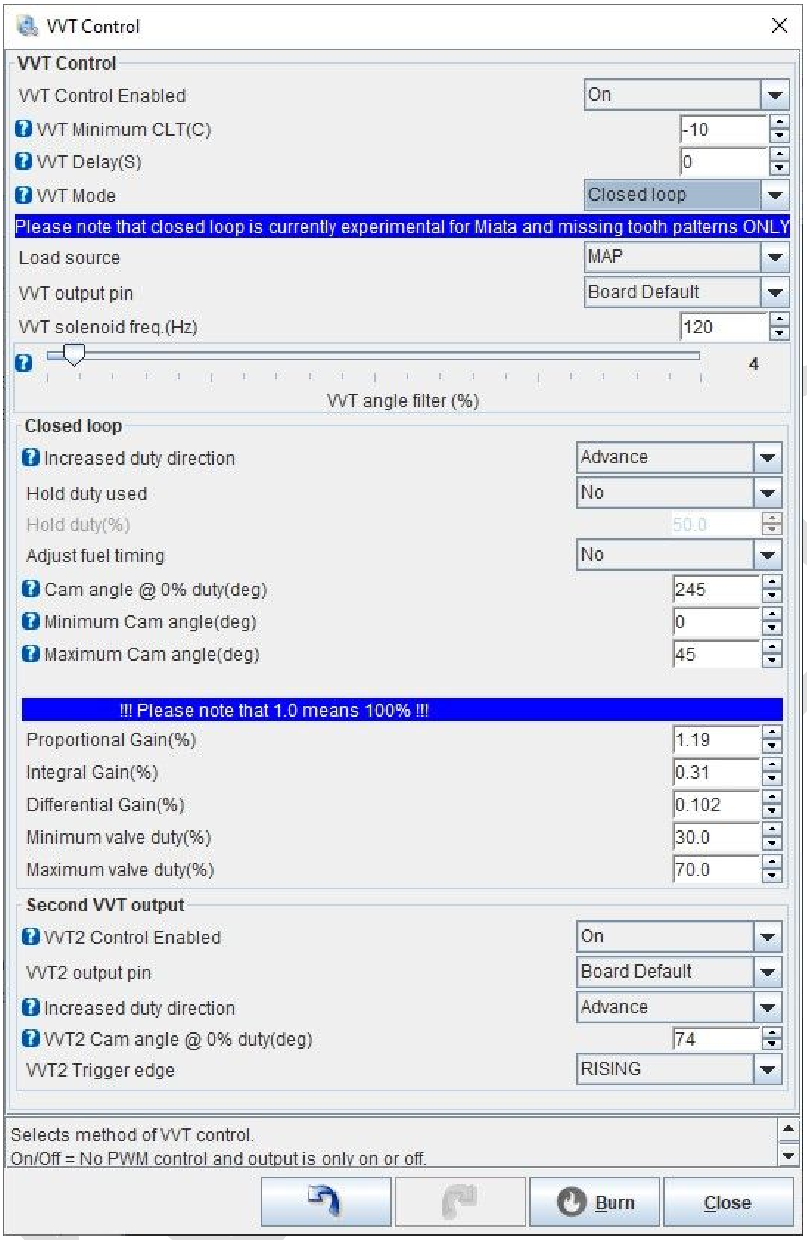
Closed loop PWM

Closed loop PWM mode also uses Pulse Width Modulation for VVT output to adjust the cam timing.

But in this mode the VVT control table is used as cam angle target table. VVT control algorithm uses PID loop to keep the cam angle at the target value using the VVT output duty. Setting up the closed loop VVT is way more in depth than On/Off or Open Loop modes. But yields to better cam control if the engine supports this kind of VVT mode. This mode can be used for example in BMW dual vanos engines and Ford ST170.

> **Note:** Currently Closed loop VVT control is experimental feature and it only works for Miata, Missing tooth and ST170 trigger patterns.

Settings



General

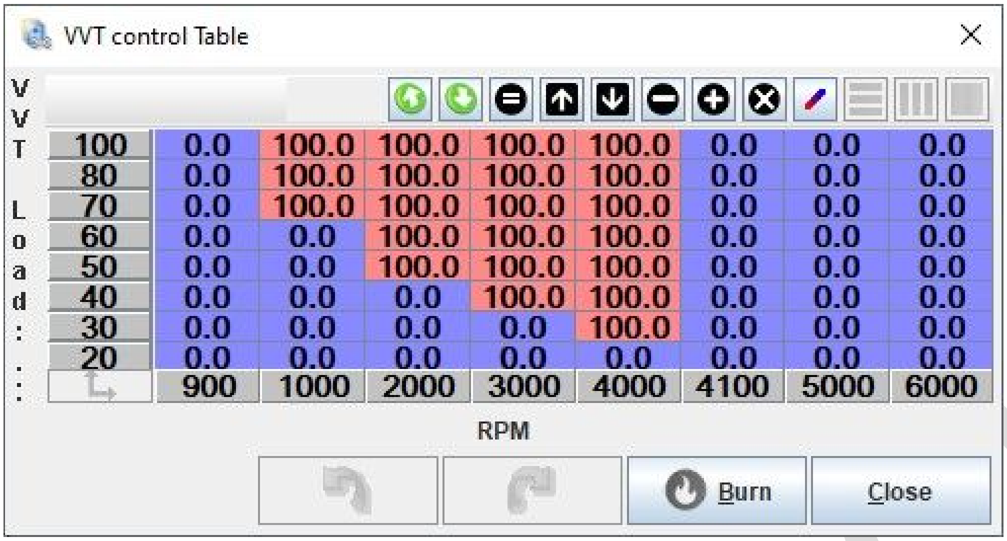
- **VVT Control Enabled** - If VVT isn't used, set this to Off.
- **VVT Minimum CLT(C)** - Minimum coolant temp to activate VVT.
- **VVT Delay(S)** - Time to wait after reaching minimum coolant temp (additional time for oil warmup).
- **VVT Mode** - For selecting one of the three VVT modes.
- **Load source** - This defines the Y-axis (Load axis) of the VVT control table. Available options for load are MAP and TPS.
- **VVToutputpin** - For selecting VVT output pin. "Board default" uses the VVT output pin specific for your board and it's the correct setting for most of the setups. But also other pins for VVT output are available.
- **VVT Solenoid freq.(Hz)** - This sets the VVT output frequency. Available frequency range is 10510Hz.

- **VVT angle filter (%)** - Adjustable filter for cam angle reading. Start with low filter values from 2 to 10, and increase filtering amount if the cam angle reading is noisy. The cam angle reading works in all three VVT modes if the trigger patterns also supports closed loop VVT. ### Closed loop
- **Increased duty direction** - Sets the closed loop control direction. If higher solenoid duty advances cam, set this to "Advance". If on the other hand, more duty retards the cam, set this to "Retard".
- **Hold duty used** - In some VVT systems, specific solenoid duty is used to hold the current cam angle. Use this setting to enable the hold duty.
- **Hold duty (%)** - Set the desired cam angle holding duty. Usually around 50%.
- **Adjust fuel timing** - By enabling this, fuel injection timing is adjusted based on the cam angle.
- **Camangle@0%duty(deg)** - This setting is used to bring the cam angle reading to usable 1-99 degrees range. First use Open Loop mode to figure out the cam angle reading at 0% duty and then switch to closed loop and write the open Loop 0% duty cam angle reading here. When doing that, the cam angle reading at 0% duty will show 0. Now you can fine tune this value so that the VVT angle reading is in the 1-99 range. You might need to adjust this value bigger amount if the cam angle reading goes to negative values when duty increases. For example, if 100% cam angle reading is -35, lower this value at least by 36. So that the both ends of the adjustment are withing the 1-99 range. Also make sure that the cam angle readings stay within the1-99 range at higher RPM too. Belt stretch etc. can affect the cam angle reading, even though duty stays the same.
- **Minimum Cam angle(deg)** - Safety limit for minimum expected cam angle value. If cam angle gets smaller or equal to this, it triggers VVT error state, closed loop adjustment is disabled and VVT output duty drops to 0%. Start by using 0 degrees and fine tune if needed.
- **MaximumCamangle(deg)** - Safety limit for maximum expected cam angle value. If cam angle gets bigger than this, it triggers VVT error state, closed loop adjustment is disabled and VVT output duty drops to 0%. Start by using 100 degrees and after everything is dialed in, set this to slightly higher value than the biggest cam angle reading is in your setup. ### Second VVT Output
- **VVT2 Control Enabled** - To enable second VVT control. This uses mainly the same settings as the primary VVT control. For thesettingsthatareavailableforVVT2, see descriptions above. Set this to Off if not used. > **Note:** Currently Closed loop VVT control for second VVT output is only available for missing tooth trigger pattern with single tooth on the cam.
- **VVT2 Trigger edge** - Set the second cam input to trigger on falling or rising edge.

VVT duty cycle

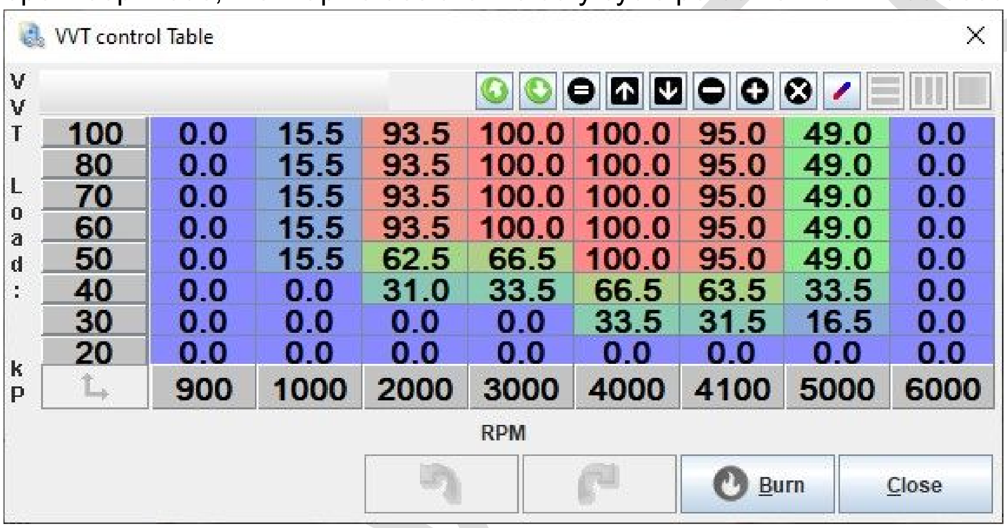
The VVT control table function varies depending on whether on/off, open or closed loop VVT mode has been selected.

- InOn/Offmode,100 is taken as "output on" and any other values represents "output off". Values 0 and 100 are recommended to use in this mode.



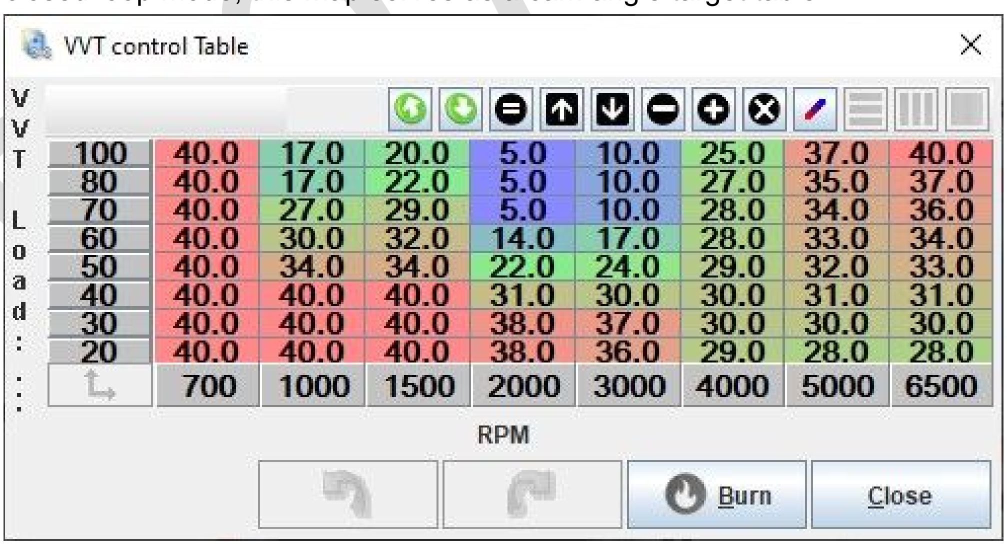
VVT Load	900	1000	2000	3000	4000	4100	5000	6000
100	0.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0
80	0.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0
70	0.0	100.0	100.0	100.0	100.0	0.0	0.0	0.0
60	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0
50	0.0	0.0	100.0	100.0	100.0	0.0	0.0	0.0
40	0.0	0.0	0.0	100.0	100.0	0.0	0.0	0.0
30	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

- In open loop mode, the map values are the duty cycle percents that will be used



VVT Load: kP	900	1000	2000	3000	4000	4100	5000	6000
100	0.0	15.5	93.5	100.0	100.0	95.0	49.0	0.0
80	0.0	15.5	93.5	100.0	100.0	95.0	49.0	0.0
70	0.0	15.5	93.5	100.0	100.0	95.0	49.0	0.0
60	0.0	15.5	93.5	100.0	100.0	95.0	49.0	0.0
50	0.0	15.5	62.5	66.5	100.0	95.0	49.0	0.0
40	0.0	0.0	31.0	33.5	66.5	63.5	33.5	0.0
30	0.0	0.0	0.0	0.0	33.5	31.5	16.5	0.0
20	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

- In closed loop mode, this map serves as a cam angle target table.



VVT Load	700	1000	1500	2000	3000	4000	5000	6500
100	40.0	17.0	20.0	5.0	10.0	25.0	37.0	40.0
80	40.0	17.0	22.0	5.0	10.0	27.0	35.0	37.0
70	40.0	27.0	29.0	5.0	10.0	28.0	34.0	36.0
60	40.0	30.0	32.0	14.0	17.0	28.0	33.0	34.0
50	40.0	34.0	34.0	22.0	24.0	29.0	32.0	33.0
40	40.0	40.0	40.0	31.0	30.0	30.0	31.0	31.0
30	40.0	40.0	40.0	38.0	37.0	30.0	30.0	30.0
20	40.0	40.0	40.0	38.0	36.0	29.0	28.0	28.0

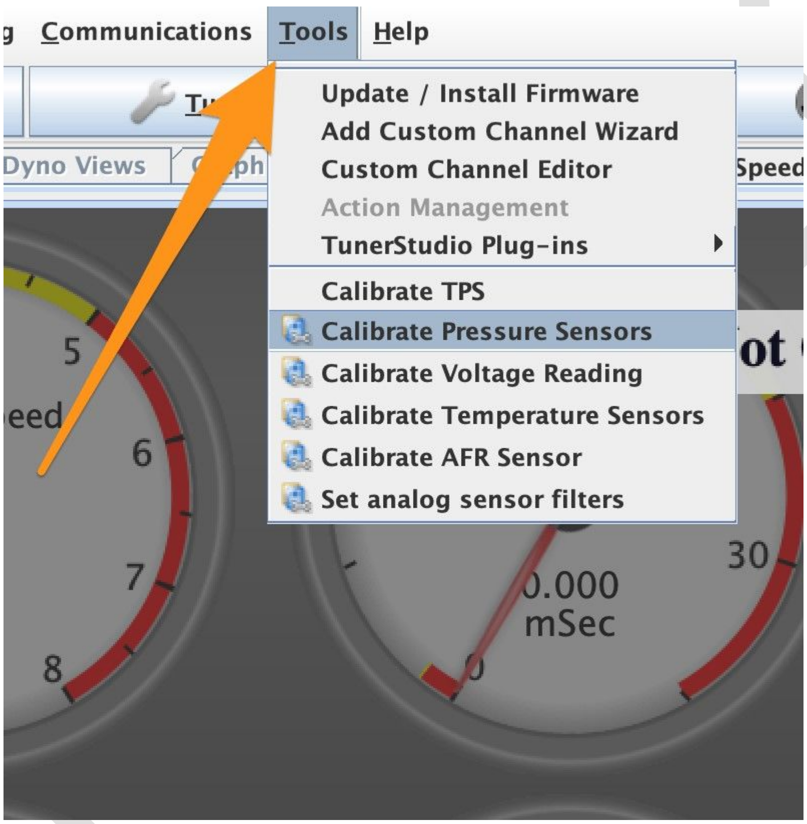
Sensor Calibration

Before your The ECU can correctly interpret the signals from the sensors, it must know which sensors you are using. Inputting this information into TunerStudio (TS) writes the correct calibration to your the ECU. It is necessary to perform this step before you can effectively check your The ECU build. Note that this is not tuning your system, but only telling it how to understand the signals from the sensors.

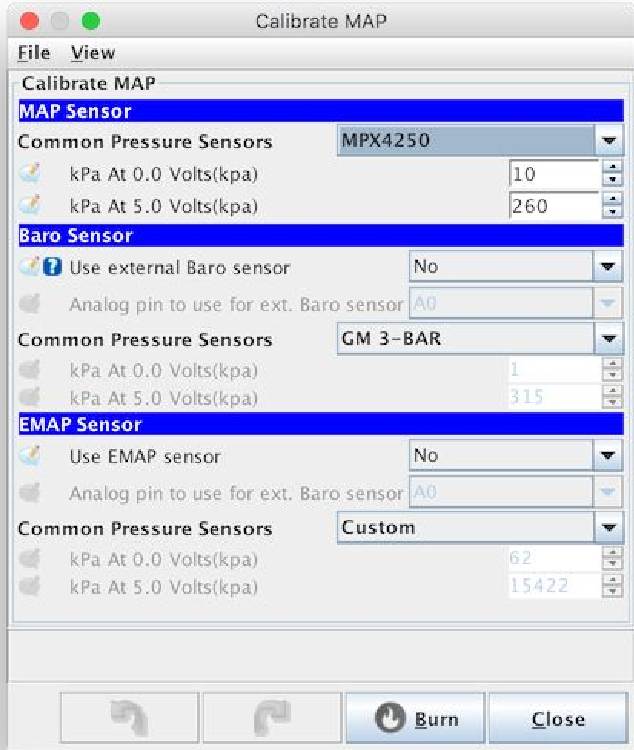
This should be completed after completing the Settings for your engine. Your computer must be connected to your The ECU through TS to perform the calibrations.

MAP Sensor

Open the **Tools** menu:



Select **Calibrate Pressure Sensors**, the window below will open:



Select your MAP Sensor from the drop down list. If you used the MAP sensor in the bill of materials, this will be the MPXH6400AP. If you are using another MAP or one from the engine manufacturer, select it from the list. Click **Burn** to send the information to the ECU.

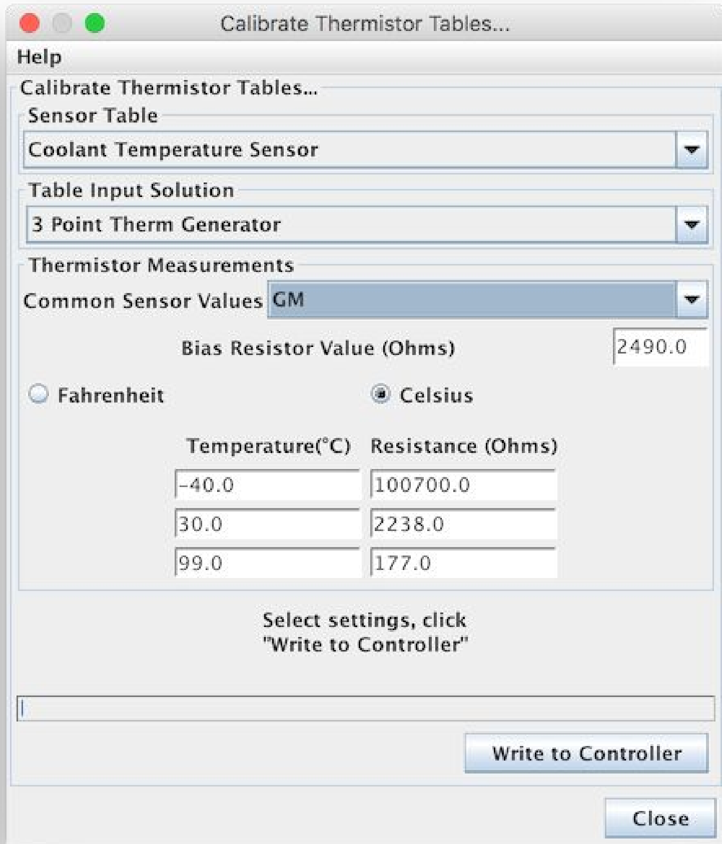
If used, the external Baro and EMAP (exhaust pressure) sensors can be calibrated in the same manner.

Coolant and Intake Temperature Sensors

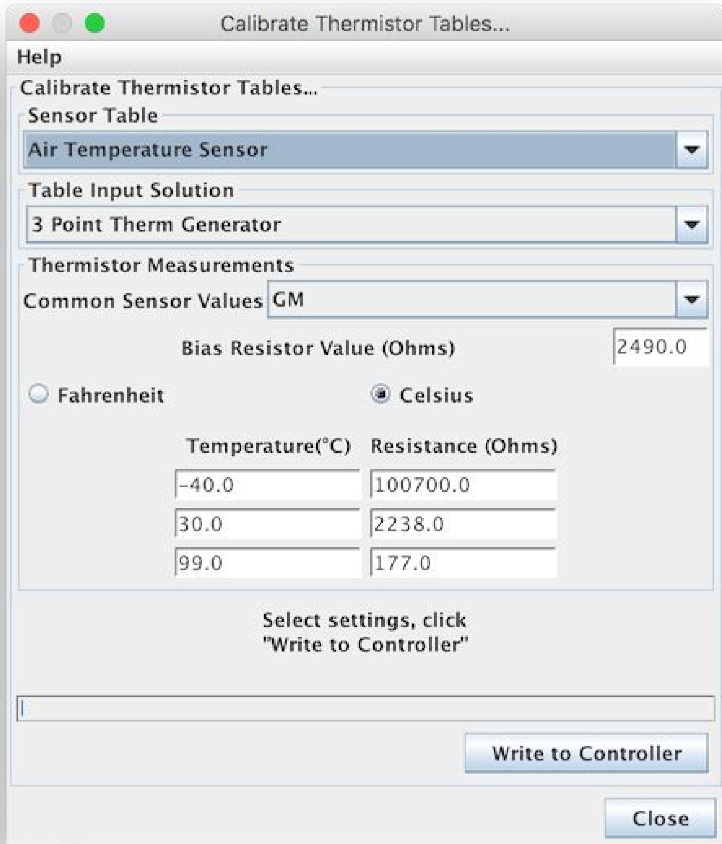
Open the **Tools** menu and select **Calibrate Thermistor Tables**:

The sensor selected will be the **Coolant Temperature Sensor**. Select your sensor from the **Common Sensor Values** drop-down list. This will place the correct values into the temperature and resistance charts and the Bias resistor value. If your sensor is not listed, see Entering Custom Values below.

Note that the standard The ECU build is to have a 2490 ohm bias resistor, which is standard for sensors used by most manufacturers. If your sensor requires another value, you may need to change resistor R3 to the correct value for your sensor. You can try overriding the Bias Resistor Value with 2490 ohms, but check to be sure your sensor reads correctly in TS.



The same calibration can then be performed for the Inlet Air Temperature (IAT) sensor by changing the **Sensor Table** to **Air Temperature Sensor**:



Select your sensor from the **Common Sensor Values** drop-down list. This will place the correct values into the temperature and resistance charts and the Bias resistor value. Click **Write to Controller** to send this information to the ECU. If your sensor is not listed, see Entering Custom Values below.

Note that the standard The ECU build is to have a 2490 ohm bias resistor, which is standard for sensors used by most manufacturers. If your sensor requires another value, you may need to change resistor R3 to the correct value for your sensor. You can try overriding the Bias Resistor Value with 2490 ohms, but check to be sure your sensor reads correctly in TS.

Entering Custom Values

Some sensors are not listed in the tables for the common sensors. If yours is not listed, you will need to enter the values into the fields yourself. You will need two bits of information: 1. The value of your bias resistor (2490 for ProyectECUs), and 2. The resistance of your sensor at three different temperatures.

The sensor resistance can be generated by measuring the resistance of the sensor in ambient air, putting it in a freezer and then in boiling water. You will need a good multimeter and an accurate thermometer that measures -10C to 100C (14°F to 212°F). It is best to use jumper wires to allow the resistance of the sensor to be read without holding it in your hand (some sensors react quickly to temperature changes). Some sensors react slowly to temperature changes, so allow the sensor at least 10 minutes to reach a stable temperature, and then record the temperature and resistance observed.

In the **Calibrate Thermistor Tables** screen, first ensure the correct temperature unit is selected (**F** or **C**). Then record the bias resistor value and the temperature/resistance values in the fields. Click **Write to Controller** to send this information to your ECU.

Note that his procedure can also be used to enter the values of resistance on simulators for testing and troubleshooting. Two points should be remembered if you use simulator values – first, never enter zero for resistance. Although your simulator may go to zero, enter some small value above zero, say 10 ohms. Entering zero leads to false values in the firmware. Second – remember to enter the correct sensor values before installing your the ECU!

Oxygen Sensor

Open the **Tools** menu again and select **Calibrate AFR Table**:

Select your **Oxygen Sensor** from the **Common Sensor Values** drop-down list. If you are using a custom Oxygen Sensor controller, select **Custom Linear WB** and then you can enter the values for **Volts** and **AFR** at two points (should be published in the controller manual).

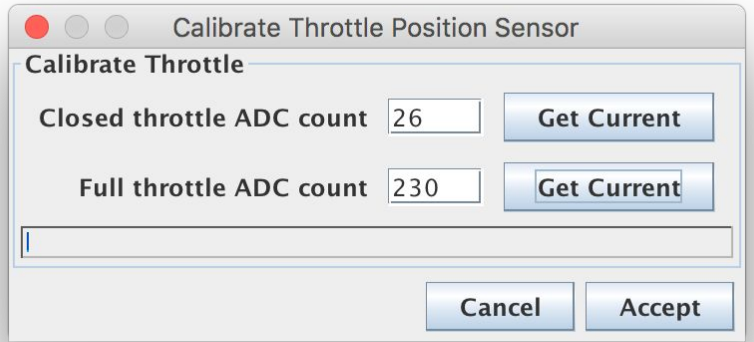
Click **Write to Controller** to send this information to your ECU.

This will set up your The ECU so that you can also run simulations to check your build before installation.

Throttle Position Sensor

Before The ECU can work correctly with your engine, you will also need to Calibrate the Throttle Position Sensor. This must be done using the throttle body and TPS used on the engine. It is best to do this while the throttle body is installed on the engine.

Open the **Tools** menu and select **Calibrate TPS**:



With the throttle closed, click the **Get Current** button beside the Closed Throttle ADC count field. Then move the throttle to full open and hold it there. Then click the **Get Current** button beside the Full Throttle ADC count field.

Click **Accept** to save the information to the ECU.

Auxiliary IO Configuration

The ECU firmware also supports the reading of up to 16 additional input channels (hardware inputs depends on the ECU). These inputs can be either Analog or Digital Pins on the Processor (or other mcu in use) .

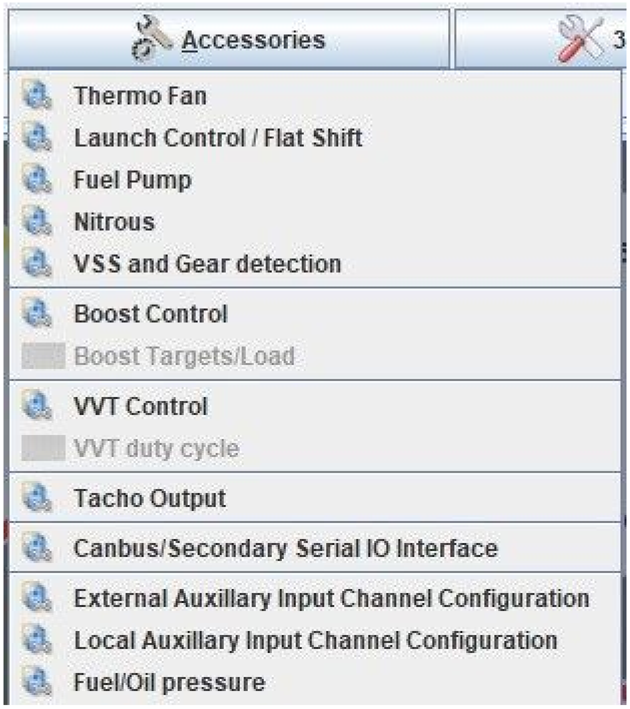
How to Use

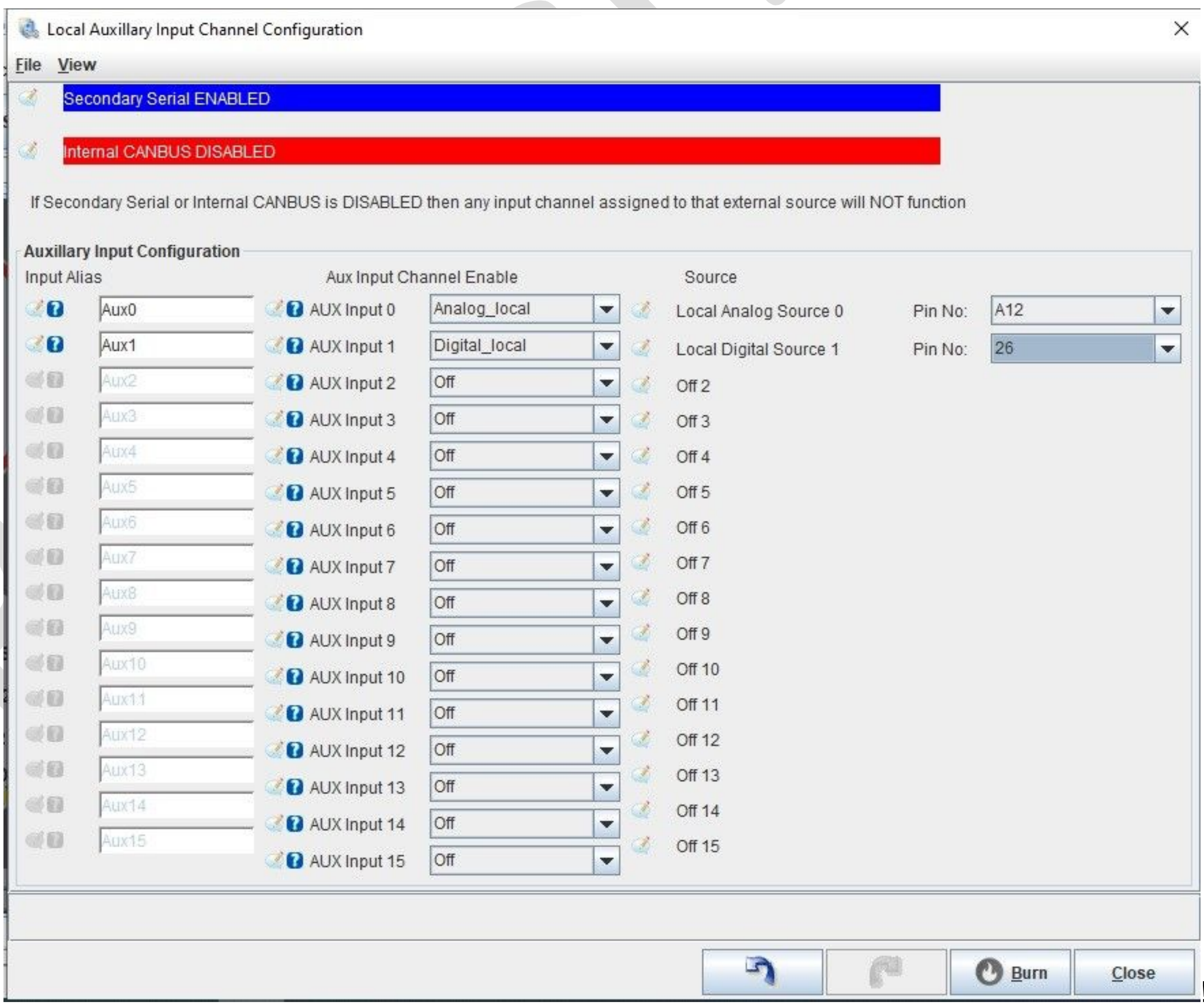
The configuration is mainly split into two categories,

- **Local MCU Pin -** How to Configure to use a Local MCU pin
- **External Data Source -** How to Configure to use a External Data Source

How to Configure to use a Local MCU pin

The configuration page is accessed from the Accessories drop down within TunerStudio, select the "local Auxiliary input channel configuration" option will be opened.


Accessories
Thermo Fan
Launch Control / Flat Shift
Fuel Pump
Nitrous
VSS and Gear detection
Boost Control
Boost Targets/Load
VVT Control
VVT duty cycle
Tacho Output
Canbus/Secondary Serial IO Interface
External Auxillary Input Channel Configuration
Local Auxillary Input Channel Configuration
Fuel/Oil pressure


Local Auxillary Input Channel Configuration
File View
Secondary Serial ENABLED
Internal CANBUS DISABLED
If Secondary Serial or Internal CANBUS is DISABLED then any input channel assigned to that external source will NOT function
Auxillary Input Configuration
Input Alias
Aux Input Channel Enable
Source
Aux0 AUX Input 0 Analog_local Local Analog Source 0 Pin No: A12
Aux1 AUX Input 1 Digital_local Local Digital Source 1 Pin No: 26
Aux2 AUX Input 2 Off Off 2
Aux3 AUX Input 3 Off Off 3
Aux4 AUX Input 4 Off Off 4
Aux5 AUX Input 5 Off Off 5
Aux6 AUX Input 6 Off Off 6
Aux7 AUX Input 7 Off Off 7
Aux8 AUX Input 8 Off Off 8
Aux9 AUX Input 9 Off Off 9
Aux10 AUX Input 10 Off Off 10
Aux11 AUX Input 11 Off Off 11
Aux12 AUX Input 12 Off Off 12
Aux13 AUX Input 13 Off Off 13
Aux14 AUX Input 14 Off Off 14
Aux15 AUX Input 15 Off Off 15
Burn
Close

In the above image the first two channels have been configured as an local analog and a local digital input respectively.

- **Input Alias -** This is a user defined Alias name (up to 20 characters) for the input channel.
- **Aux Input Channel Enable** - This Enables/Disables the input channel.
- **PIN** - Is the pin selected (only for local options).

InputAlias The input alias can be any ascii character name the user wishes up to 20 characters long. This can also be left as the default.

Aux input Channel Enable Options here are:

- **OFF -** Channel is disabled.
- **EXT/CAN** - The channel is assigned to an external data source.

(this option is only visible if CAN_COMMANDS is enabled in project properties. See here for further information).

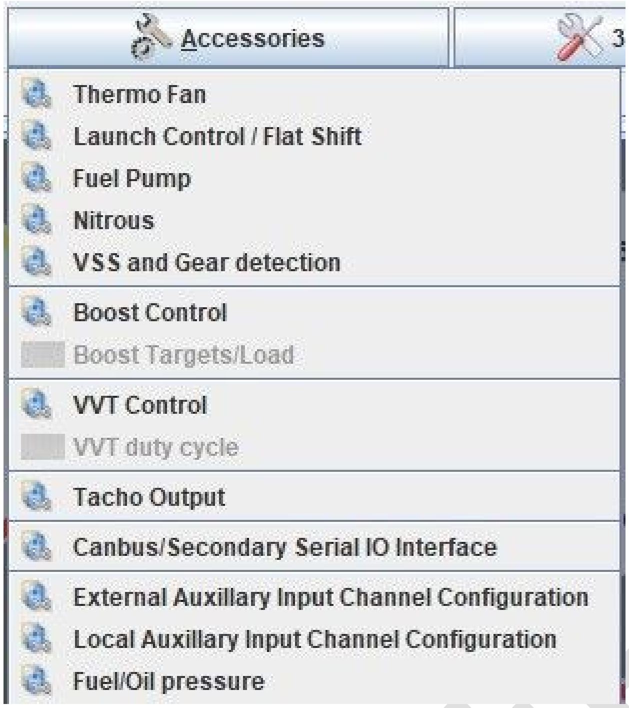
- **Local_analog -** Select a local analog mcu pin.
- **Local_Digital** - Select a local digital mcu Pin.

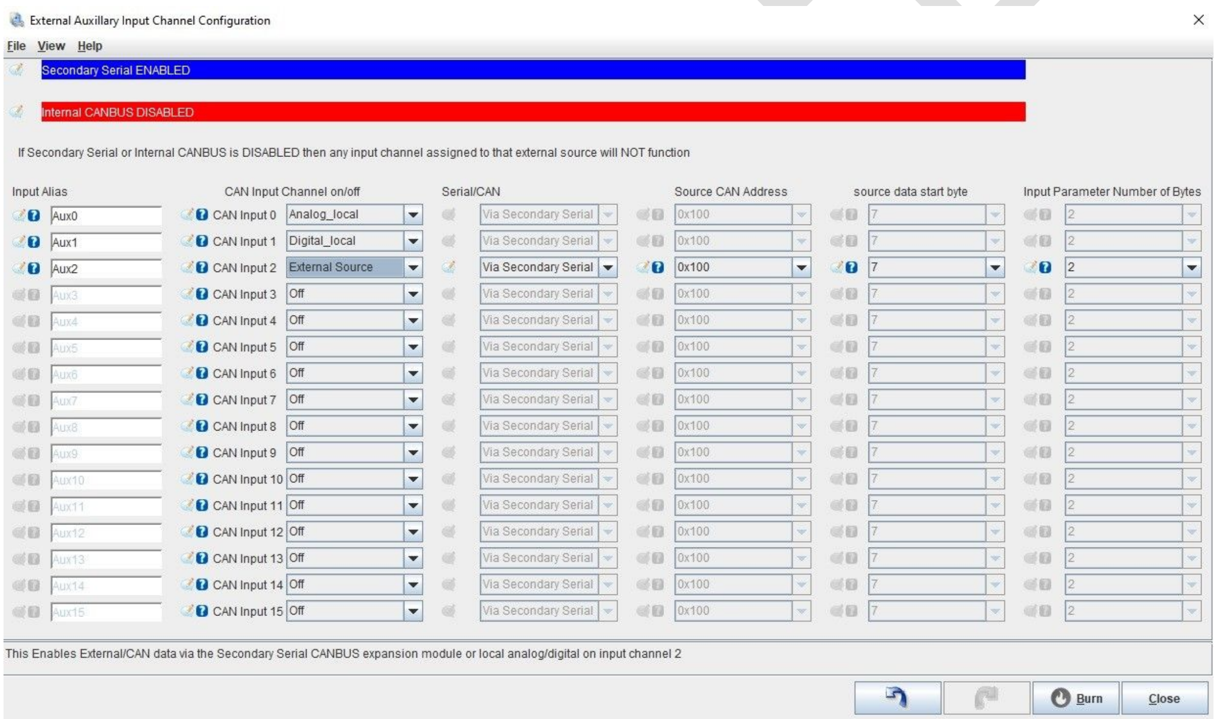
PINThis setting is only available for local mcu pin selections. It is the actual mcu pin name.

How to Configure to use an External Data Source

To use the Auxiliary input channels for external data the Secondary IO must be enabled. See here for further information on how to do this.

The configuration page is accessed from the Accessories drop down within TunerStudio, select the "External Auxiliary input channel configuration"
option will be opened.





For External data inputs to be active the “Enable External Data Input” option must be enabled.

In the above image the first three channels have been configured as an local analog and a local digital input and a External input respectively.

- **Input Alias** - This is a user defined Alias name (up to 20 characters) for the input channel
- **External Aux Input Channel Enable** - This Enables/Disables the input channel
- **Source CAN Address** - Is Real Can address of the source device

- **Source Data Start Byte** - Is the first byte (of the 8bytes in a canbus command) where the data can be found.
- **Input Parameter Number of Bytes** - Is the number of bytes the data is stored in (lsb first).

InputAlias The input alias can be any ascii character name the user wishes up to 20 characters long. This can also be left as the default.

External Aux input Channel Enable   Options here are:

- **OFF** - Channel is disabled.
- **EXT/CAN** - The channel is assigned to an external data source.
- **Local_analog** - Select a local analog mcu pin.
- **Local_Digital** - Select a local digital mcu Pin.

Source CAN Address This is the Hex address of the remote Device.

Source Data Start Byte A can data command has up to 8 bytes. This value sets the first data byte the data value begins at.

Input Parameter Number of Bytes The data byte can be made from a single byte or two (word or 16bit value)

Supported trigger patterns

Missing Tooth Pattern

Overview

A missing tooth crank trigger is used as standard equipment by a number of OEMs, most notably Ford, but is also very popular as an aftermarket fitment.

It is comprised of crank wheel with a given number of evenly spaced teeth, and one or more 'missing' teeth. Common values are typically 60-2, 36-1, 24-1, 12-1 and 4-1 where the first number represents the total number of teeth the wheel would have if there were none missing. The second number after a dash "-" indicates the number of teeth missing.

> **Note**: If there is a third number (e.g., 36-1-1), the missing teeth are not sequential, and this decoder does not apply. Do not confuse counts with slashes "/", as numbers following slashes represent cam teeth—not missing teeth. Wheels with "+" indicate added teeth rather than missing, and again this decoder does not apply.

Applications

Missing tooth crank wheels can be used on virtually any engine and is one of the more popular aftermarket options. It provides very good resolution in the higher tooth count versions (Eg 36-1 and 60-2) without being CPU intensive to decode.

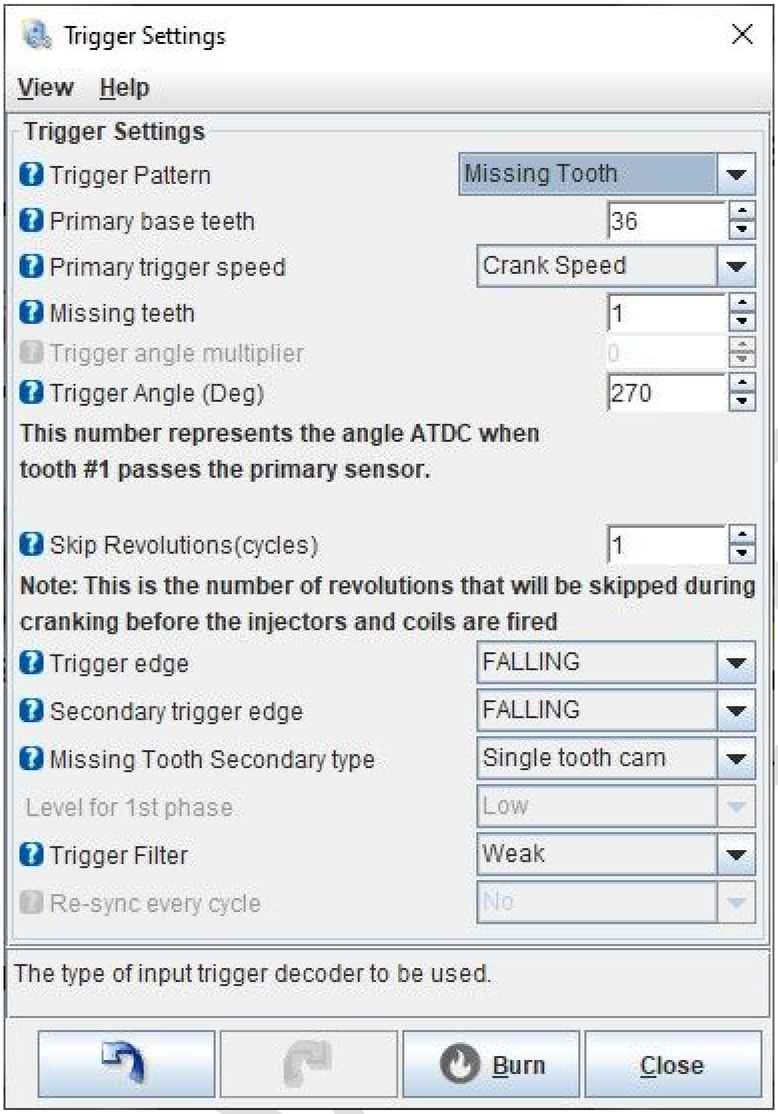
Timing Requirements

The missing tooth crank and cam decoders require that the wheel is spinning at roughly the same speed throughout the rotation. For single missing tooth decoders: If the next tooth does not come within 1.5 * The Delta Time of the last 2 teeth it is assumed we just observed the missing tooth. For morethanonemissingtoothdecoderthereisabitmoreleewayifthenexttoothdoesnotcomewithin 2 * The Delta Time of the last 2 teeth it is assumed we just observed the missing teeth.

Usually this can be fixed by ensuring that the starter motor has enough current available to power through any harder spots through the rotation / opening closing cams / engine accessories.

If the starter motor is good and getting the right voltage, ensure the mechanical components of the engine are correct.

Tuner Studio Configuration



Fields:

- Primarybaseteeth: This is the number of teeth the wheel would have if there were none missing. Eg a 36-1 wheel has only 35 actual teeth, but you would enter 36 into this field.
- MissingTeeth: The size of the 'gap' in the number of teeth. These missing teeth must be situated in a single block (ie there's only a single gap in the teeth).
- Trigger Angle: This is the angle in crank degrees **AFTER** TDC (ATDC) of the first tooth following the gap.

Timing Setting

The trigger angle can be found using the following steps:

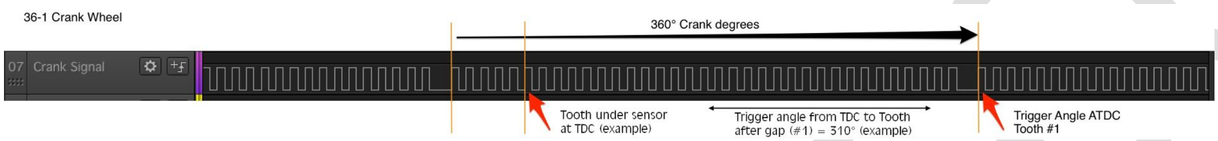
1. Set the crankshaft at TDC 0° (Cylinder 1) with a tool by hand.
2. Rotate the crankshaft (running direction) until the first tooth after the missing teeth is under the sensor.
3. Measure how many degrees the crankshaft rotated. This is the value to enter as the trigger angle.

Sequential operation

The missing tooth decoder supports sequential operation if an additional cam input is present. If Sequential mode is selected for either the fuel timing or spark timing, the system will expect to see a cam signal and will not sync correctly without this. Note that this is ONLY the case if sequential is selected for one or both of fuel and spark timing.

This cam signal should take the form of 4-1 cam trigger wheel or a single pulse every complete cycle. This can be a short tooth or a half moon type arrangement, provided that electrically there is only a single rising (or falling) pulse per cycle.

Trigger Diagram



Missing tooth (Cam speed)

The missing tooth cam – speed trigger is a ECU innovation, that permits function similar to a dual wheel setup, thereby allowing sequential or wasted spark operation from cam-mounted or distributor wheels. The operation is based on both Missing Tooth and Dual Wheel. It is suggested to read those sections first for familiarization as this section will only highlight the fundamental differences to those common decoders.

This decoder is comprised of a single cam-speed wheel in the same configuration as a crank-mounted missing-tooth wheel. The number of teeth ***must*** be evenly divisible into 720°. As it rotates at half crank speed, the sensor reads half the wheel teeth on each 360° crank revolution, and the remaining teeth on the next crank rotation. A single missing tooth will appear on only one of the two crank rotations, and is then used as a phase indicator, much as the dual-wheel system uses the cam signal.

Applications

Missing tooth cam or distributor wheels can be used with cam or distributor wheel modification or fabrication as no OEM systems use it originally. The wheel ***must*** have at least as many teeth as cylinders, ***not*** including the missing tooth. This generally requires double the number of teeth as cylinders or more. As many teeth, slots, or other readable features (sensor targets) as possible in the limited space is recommended in order to satisfy this requirement, and to maximize resolution. The sensor must be capable of reliably reading smaller or closely-spaced teeth.

Due to typically limited teeth, only half the teeth being read on each revolution, and the potential for reduced accuracy due to timing drive wear; the timing accuracy may be reduced in comparison to crank wheel systems. A figure of error cannot be predicted here as the wear or 'slop' of a given engine will be unique. However, it should be reasonable to assume the timing error will not exceed theaccuracyofanOEM-equivalentcam-drivensystemsuchastypicaldistributorsystems, or possibly better due to more sensor targets.

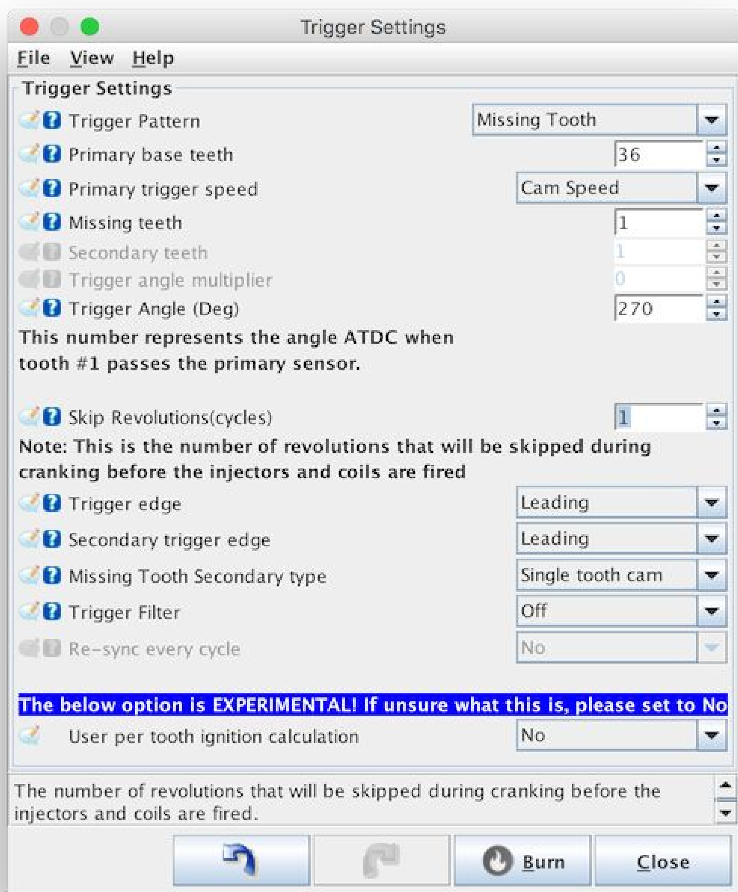
Timing Requirements

The missing tooth crank and cam decoders require that the wheel is spinning at roughly the same speed throughout the rotation. For single missing tooth decoders: If the next tooth does not come within 1.5 * The Delta Time of the last 2 teeth it is assumed we just observed the missing tooth. For morethanonemissingtoothdecoderthereisabitmoreleewayifthenexttoothdoesnotcomewithin 2 * The Delta Time of the last 2 teeth it is assumed we just observed the missing teeth.

Usually this can be fixed by ensuring that the starter motor has enough current available to power through any harder spots through the rotation / opening closing cams / engine accessories.

If the starter motor is good and getting the right voltage ensure the mechanical components of the engine are correct.

Tuner Studio Configuration



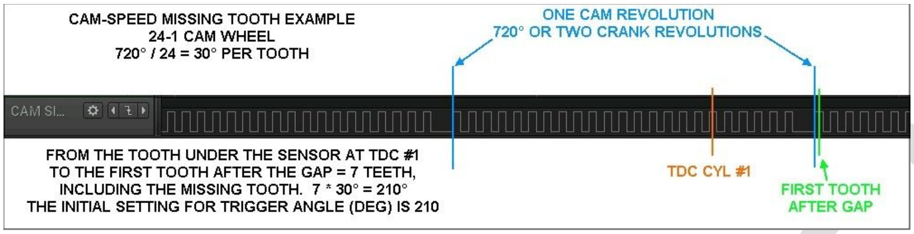
Fields:

- **Primarybaseteeth**: This is the number of teeth the wheel would have if there were none missing, e.g. a 36-1 wheel has only 35 actual teeth, but you would enter 36 into this field.
- **Missing Teeth:** The size of the 'gap' in the number of teeth. These missing teeth must be situated in a single block (ie there's only a single gap in the teeth). One missing tooth is recommended.
- **TriggerAngle**: This is the angle in **crank degrees AFTER** TDC (ATDC) of the first tooth following the gap. This number ranges from -360° to +360°.
- **Cam Speed**: Ensure this box is checked for this cam-speed system.

Timing Setting

The trigger angle is set in CRANK degrees, not cam.

Trigger Pattern



Dual Wheel

A dual wheel trigger is one where there is a primary multi-tooth wheel combined with a secondary single pulse to provide location information. The primary input should contain no missing teeth. Both pulses can run at either cam or crank speed, but sequential operations requires that the secondary pulse is located on the cam. The design of the secondary trigger can vary (Eg a single short tooth, half-moon wheel etc), provided it only provides a single pulse per revolution.

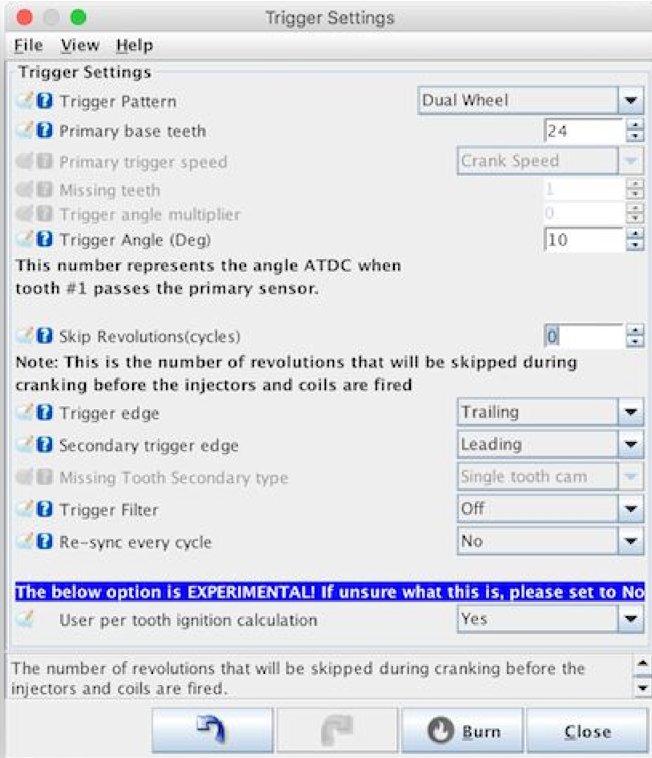
As with other arbitrary tooth count wheels, the number of teeth must evenly divide into 360 (or 720 if running at cam speed).

Tooth #1 is defined to be the first tooth on the primary wheel AFTER the pulse on the secondary wheel.

Applications

Dual wheel triggers are standard fitment on a number of Euro make cars, particularly those from VW and Audi. They are also a popular aftermarket fitment due to their simplicity and ease of fitment.

Tuner Studio Configuration



Fields:

- **Primary base teeth**: This is the number of teeth on the primary input wheel. If the primary wheel is located on the cam (or is otherwise running at cam speed), divide it's teeth by two for this setting
- **Trigger Angle**: This is the angle in crank degrees **AFTER** TDC (ATDC) of the first tooth on the primary input, following the single pulse on the secondary input.
- **Triggeredge**: Whetherthetriggerwillbetakenfromtheleading(rising)ortrailing(falling)edge of the primary input
- **Secondary trigger edge**: As above, but for the secondary input
- **Re-sync every cycle**: Whether the system will reset the sync level every time the secondary input is seen. This can be useful for noisy crank triggers that otherwise my lose sync permanently and not recover until a restart.

Timing Setting

The trigger angle can be found by placing the engine at TDC, then calculating how far it must be rotated until the first primary tooth after the secondary pulse.

Sequential operation The missing tooth decoder supports sequential operation if the secondary input is running at cam speed. If Sequential mode is selected for either the fuel timing or spark timing, the system will expect that the secondary input is running at cam speed and will only provide half the output pulses if this is not the case.

This cam signal should take the form of a single pulse every complete cycle. This can be a short tooth or a half moon type arrangement, provided that electrically there is only a single rising (or falling) pulse per cycle.

Basic Distributor

The Basic Distributor trigger is a very simplistic decoder that expects input like what a traditional distributor outputs. That is, 1 pulse per cylinder per cycle.

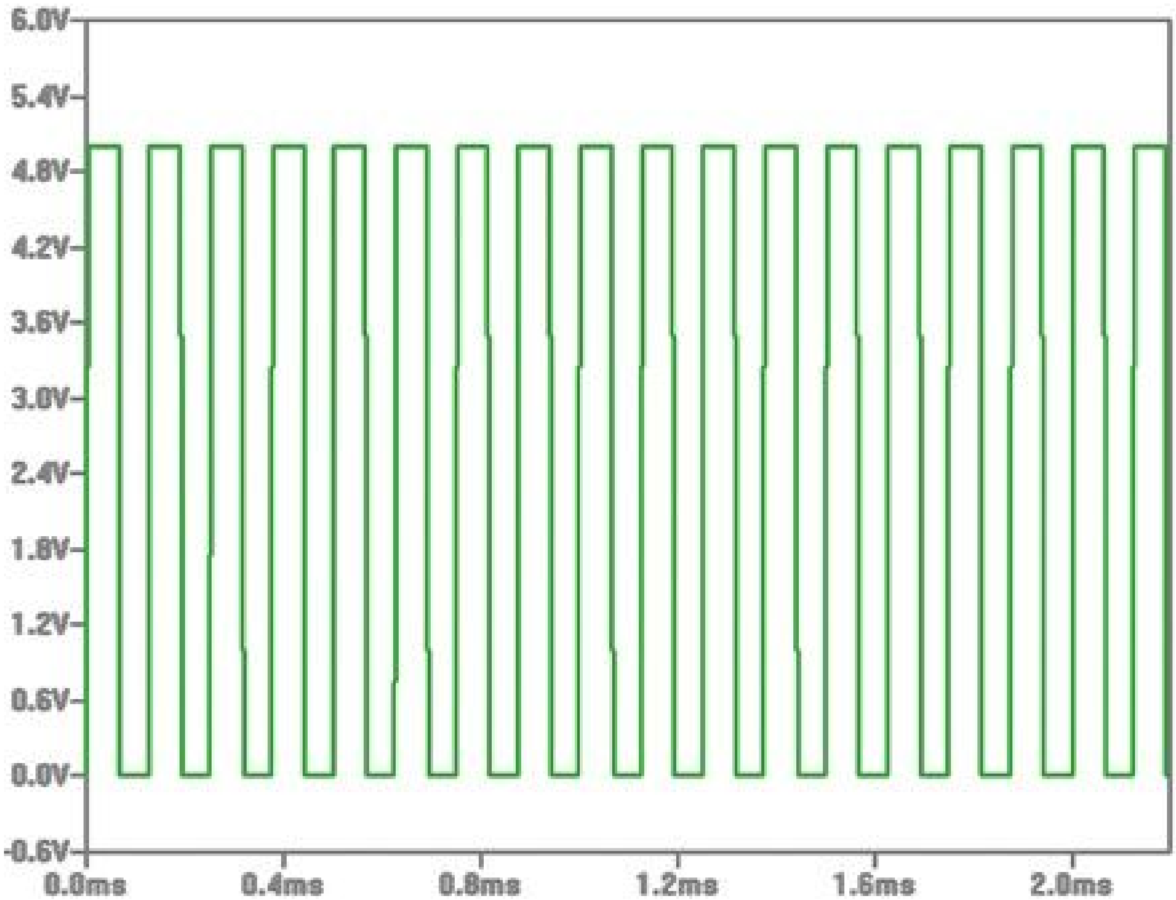
For this reason the signal lacks any cylinder position signal and so without a missing/added tooth or camshaft signal reference The ECU cannot calculate crankshaft angle, phase of cycle, or cylinder assignment. **A distributor must be used to route the resulting sparks to the correct cylinders** (With the exception of single cylinder engines).

The signal can be as simple as the breaker points from an old pre-electronic distributor, to a crankshaft wheel without any abnormal, extra, or missing slots, provided it is conditioned appropriately to 0v-5v. Most who have installed aftermarket tachometers are familiar with the simplicity of the signal with the only variation being the number of pulses in each crankshaft rotation.

A note on resolution

By its nature the resolution of a simple distributor wheel is quite low. The exact resolution depends on the number of cylinders (The more cylinders, the higher the resolution), but even with an 8 cylinder engine there are only 4 pulses per revolution. This can impact timing accuracy if running ignition control from The ECU and in most cases upgrading to a higher resolution trigger wheel is strongly recommended.

Trigger Signal



GM 7X

This decoder uses a GM trigger wheel with six notches spaced evenly apart and one uneven notch. The uneven notch is counted as #3 with a total of seven notches.

4G63 Pattern (not recommended)(install a 24tooth cas disc instead)

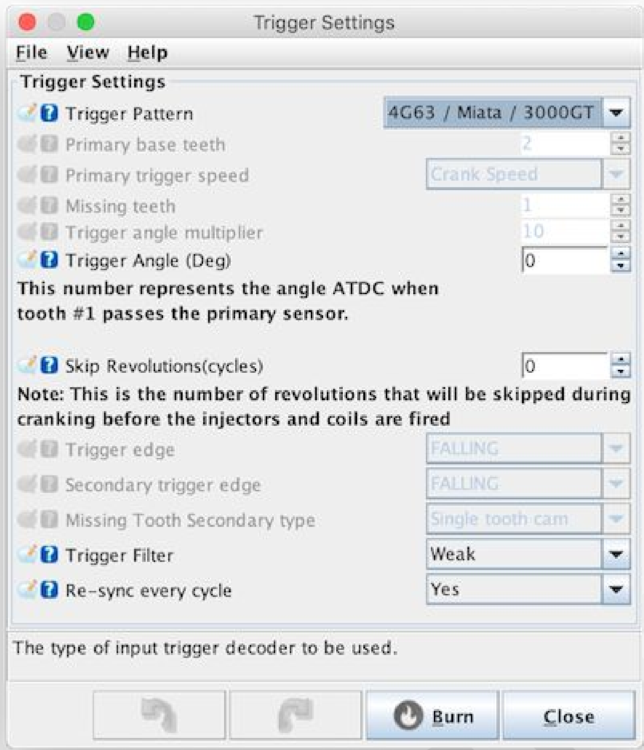
The 4g63 trigger is used across a large number of both Mitsubishi and Mazda 4 cylinder engines. See below for applications.

It is comprised of crank and cam signals that are provided by either a hall sensor or an optical sensor.
The signal is electrically the same in both cases.

Applications

- Mitsubishi Lancer
- NA Miata / MX-5 (Up to 1997)

Tuner Studio Configuration

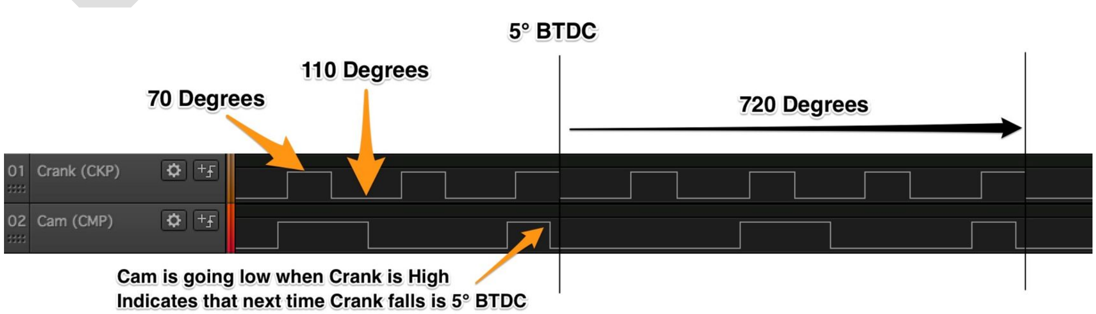


> **NOTE** Within the Cranking options dialog, ensure that the Fix cranking timing with trigger option is turned **ON**

Timing adjustment

In most cases altering the trigger angle should not be required, however there is some small variation between the OEM versions of this trigger so some minor adjustment may be needed. Once you have the engine started, set a fixed ignition angle, and check the timing with a timing light. If this is a few degrees out (<20°), adjust the trigger angle here. If this is more than 20° out, there may be a larger problem.

Trigger Pattern



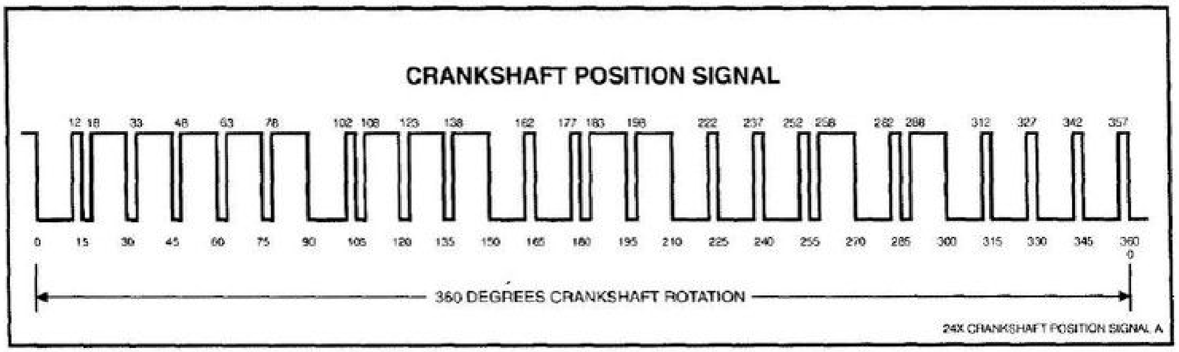
GM 24X

Overview

This a 24 tooth wheel with 12 wide teeth and 12 narrow teeth. The narrow provides 3 degrees of pulse while the wide provides 12. All of the falling edges are 15 degrees apart. This decoder uses the falling edges, requiring the cam signal to determine crank angle.

> The “Dual wheel” decoder set to 24 teeth and FALLING edge should be used rather than the dedicated 24X decoder. An updated version of the dedicated 24X decoder remains a WIP.

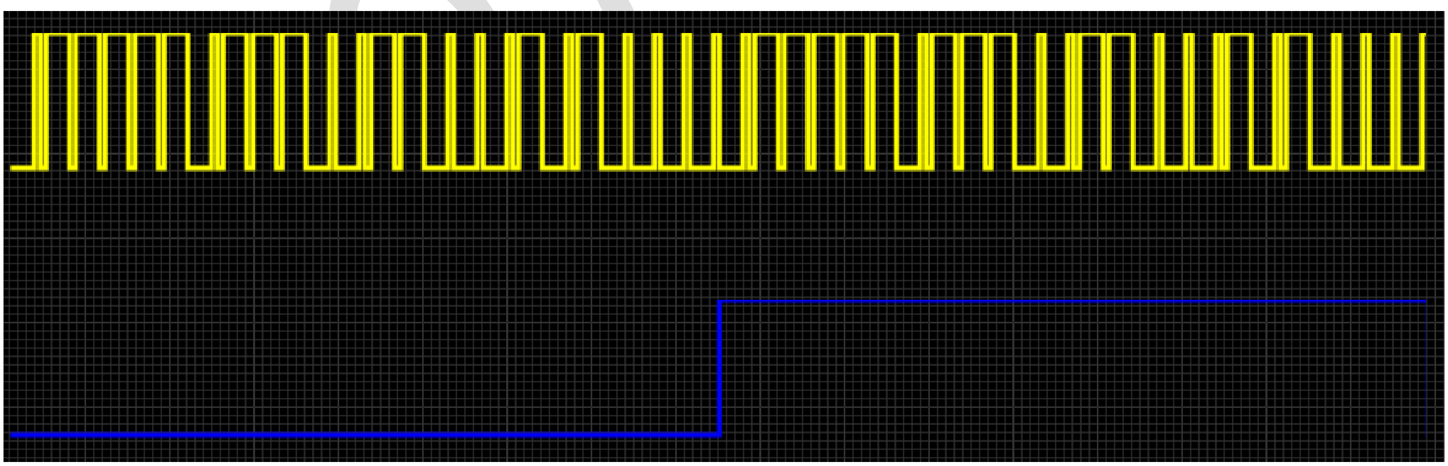
Trigger Signal



Crankshaft Position Sensor Signal

Overview

There are two signals one from the crank wheel and the other from the cam. The crank wheel puts out a series of four pulses every 120 degrees. Each of the four pulses is 20 degrees apart and lasting only 2 degrees. The cam wheel pluses once every 360 degrees or 720 crank degrees. The pulse last for 180 degrees or 360 crank degrees.



Harley Evo

The Harley EVO pattern is used on V-Twin engines from ’86 through to ’99.

This pattern will work on all injected EVO engines.

Overview

The Honda D17 decoder applies to the Honda engine family using a 12+1 crankshaft wheel. The 4+1 camshaft signal is not currently supported with The ECU, but as of Oct'23 development including vtec support is underway. Without the cam signal, all standard fuel and ignition modes up to semi sequential and wasted-spark are supported.

Applications

- TBA

Tuner Studio Configuration

Timing adjustment

In most cases altering the trigger angle should not be required, however there is some small variation between the OEM versions of this trigger so some minor adjustment may be needed. Once you have the engine started, set a fixed ignition angle and check the timing with a timing light. If this is a few degrees out (<20°), adjust the trigger angle here. If this is more than 20° out, there may be a larger problem.

Trigger Pattern

The crank trigger wheel consists of 12 evenly spaced teeth plus 1 additional 13th tooth which provides position information. The first tooth after this 13th one is considered Tooth #1

Miata 99-05

From MY99 onwards, Miatas moved to a new trigger patter that, whilst similar to that used on the 4g63, is more tolerant to noise and does not rely on both edges of a tooth being tracked. Crucially it also permits movement of the cam signal relative to the crank signal which is required due to the addition of variable cam timing in these engines. Sync can be determined in the same way regardless of if the variable cam is at it's maximum or minimum movement.

The trigger consists of a 4 tooth wheel located on the crankshaft and a 3 tooth wheel on the cam. The teeth on both wheels are unevenly spaced.

Applications

NB Miatas from 1999 until 2005.

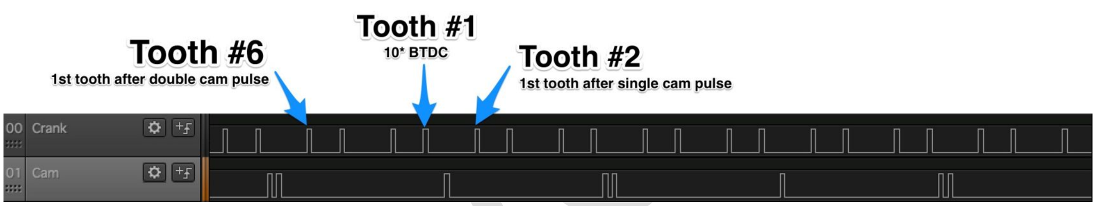
Tuner Studio Configuration

- The trigger angle should not need changing once this pattern has been selected (ie Make sure it is set to 0)
- Both trigger edges should be set to RISING
- For most installs, Trigger filtering set to Off or Weak is sufficient.
- In the Starting/Idle -> Cranking Settings dialog ensure the follow ing options are turned on:
  - 'Fix cranking timing with trigger'
  - 'Use new ignition mode'

Trigger Pattern

The crank wheel contains 4 teeth, separated by an alternating 70 and 110 degrees.

Syncisdeterminedbycountingthenumberofsecondary(cam)pulsesthatoccurbetweentheprimary (crank) pulses and can be confirmed at 2 points in the cycle. The first crank pulse after 2 cam pulses is tooth #6 and the first crank pulse after a single cam pulse is tooth #2. Tooth #1 is located at 10 degrees BTDC and cannot be identified directly, only relative to teeth #2 and #6. As the camshaft timing is moved as part of the VVT, the secondary pulses remain within the same 'window' relative to the primary pulses. Sync can therefore before confirmed at all loads and speeds, no matter what VVT value is being currently used.



Non-360 Decoder

This is a variation of the dual-wheel decoder that can be used with tooth counts that do not divide evenly into 360° This decoder system is usually unique to a particular brand or engine series, and therefore has previously been assigned a name to identify the type, such as the Audi 135 decoder. While this "uneven divisor" decoder can be used with a variety of tooth counts, not all tooth counts can be used with this system.

Nissan 360

The Nissan 360 CAS trigger is used across a large number of both4 and 6 cylinder Nissan engines. See below for applications.

The trigger is comprised of a wheel, running at cam speed, that has 360 windows and is read by an optical sensor. Each window therefore represents 2 crank degrees. For location information, there is also an inner ring of windows, equal to the number of cylinders (ie 4 windows on 4 cylinder engines, 6 windows on 6 cylinder engines).

NOTE: There are multiple versions of the 4 cylinder CAS and not all are currently supported. Each known version is described below:

1. Pattern 1 - Has a single unique inner window with all others being identical. Not currently supported.
2. Pattern 2 - The unique slot sizes are in opposing pairs. This is partially supported.

3. Pattern 3 - Each inner window has a unique size. Typically 4, 8, 12, 16 on 4 cylinder engines and 4, 8, 12, 16, 20, 24 on 6 cylinders. This is supported.

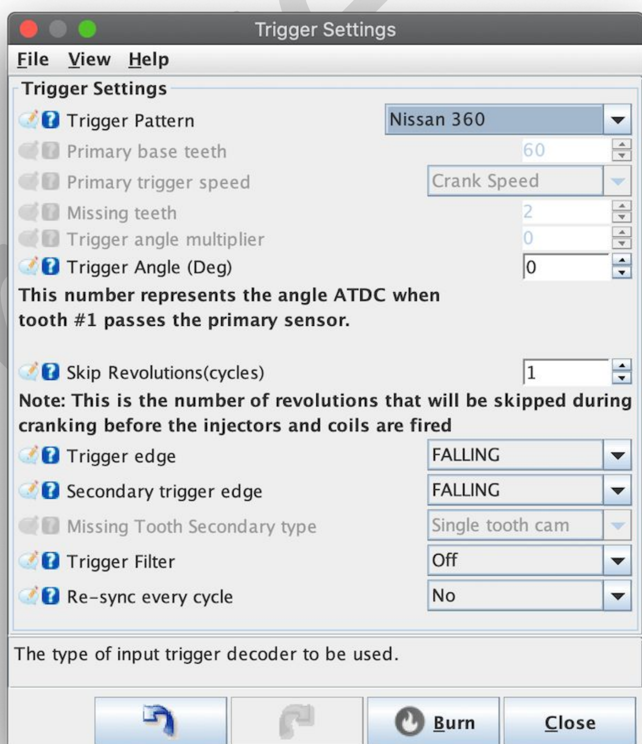
Applications

- CA18 - Believed to have pattern 3
- SRxx Redtop - Believed to be pattern 3
- SRxx Blacktop (early) - Believed to be pattern 1
- SRxx Blacktop (notch) - Believed to be pattern 1
- FJ20 - Believed to have pattern 1
- RB30 - Believed to have pattern 1
- RB25/26 - Believed to all have pattern 3

Tuner Studio Configuration

- Set both Trigger edge to Trailing
- Trigger Filter: off
- Re-sync every cycle: yes

NOTE: If you are still not seeing any RPM signal or Sync try reversing the CAM and CRANK signals on the IDC40. These settings are confirmed for the 4-8-12-16 wheel.



Timing adjustment

In most cases altering the trigger angle should not be required, however there is some small variation between the OEM versions of this trigger so some minor adjustment may be needed. Once you have the engine started, set a fixed ignition angle and check the timing with a timing light. If this is a few degrees out (<20°), adjust the trigger angle here. If this is more than 20° out, there may be a larger problem.

Trigger Pattern

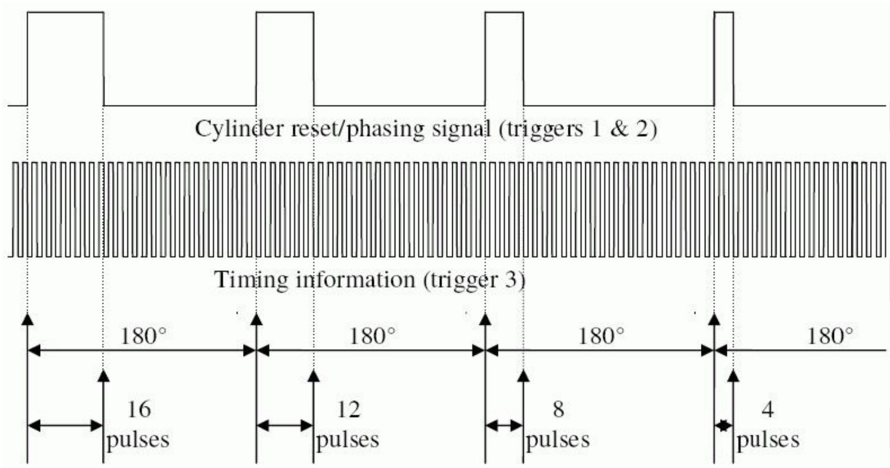


Figure 20: Stylised Nissan cam angle sensor waveform

Daihatsu +1

Overview

The Daihatsu +1 triggers are used across a number of 3 and 4 cylinder engines from Daihatsu. See below for applications.

It is comprised of a single cam signal provided by either a hall sensor. This should be fed into the RPM1 input on the ECU

Applications

- TBA (3 cylinder)
- TBA (4 cylinder)

Tuner Studio Configuration

Simply select the Daihatsu +1 trigger option.

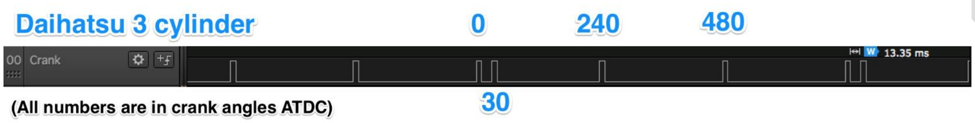
Timing adjustment

In most cases altering the trigger angle should not be required, however there is some small variation between the OEM versions of this trigger so some minor adjustment may be needed. Once you have the engine started, set a fixed ignition angle and check the timing with a timing light. If this is a few degrees out (<20°), adjust the trigger angle here. If this is more than 20° out, there may be a larger problem.

Trigger Pattern

In 3 cylinder engines, there are 3 evenly spaced teeth at 0, 240 and 480 crank degrees. There is an additional (+1) tooth located at 30 crank degrees to provide position info.

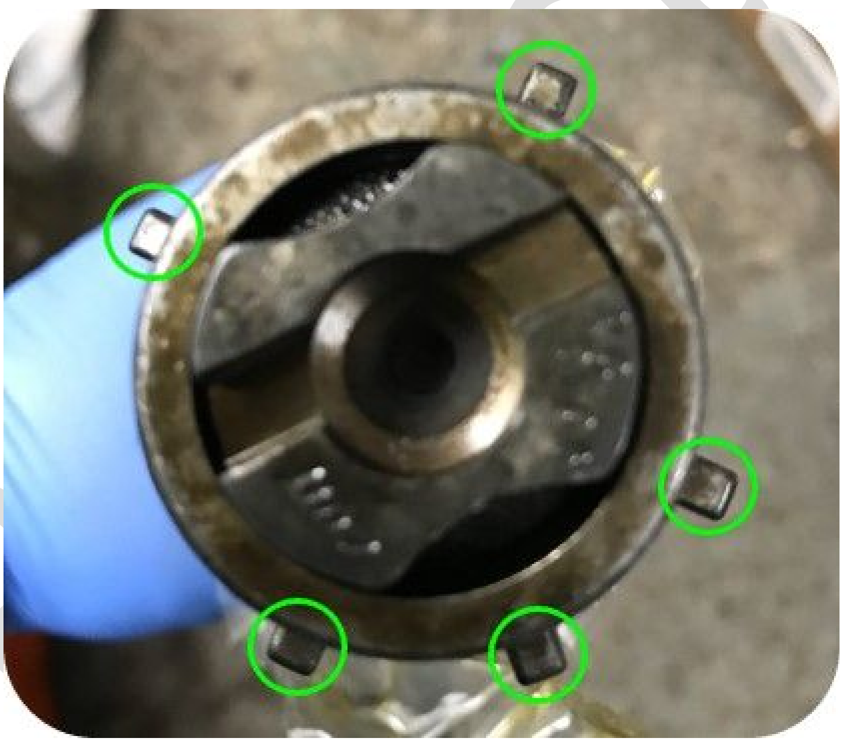
The 4 cylinder is the same, except with 4 evenly spaced teeth. The 5 teeth are therefore located at 0, 30, 180, 360 & 540 (Crank degrees, ATDC)



Ford ST170

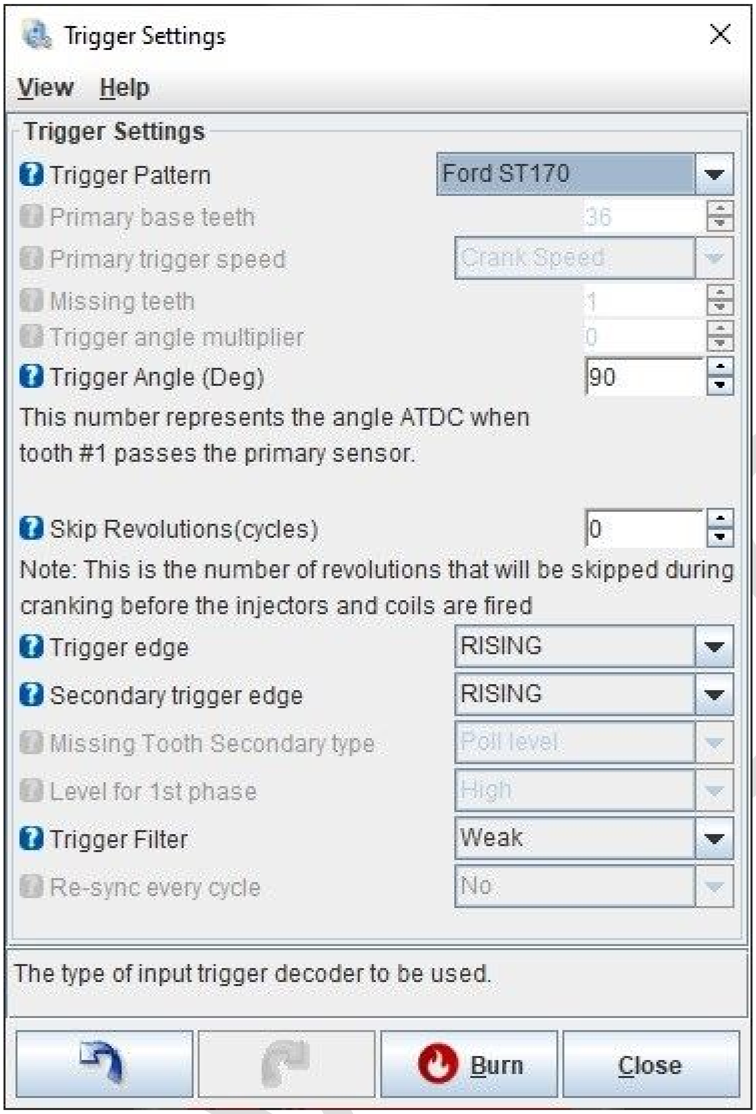
Overview

Ford ST170 trigger pattern consists of 36-1 missing_tooth pattern on crank and 5-tooth cam trigger wheel for sequential operation and VVT.

Cam pattern: 

Applications

- Ford Focus ST170 **Tuner Studio Configuration**



Fields:

- Trigger Angle: This is the angle in crank degrees **AFTER** TDC (ATDC) of the first tooth following the gap

Timing Setting

TBD

Sequential operation

TBD

Trigger Diagram

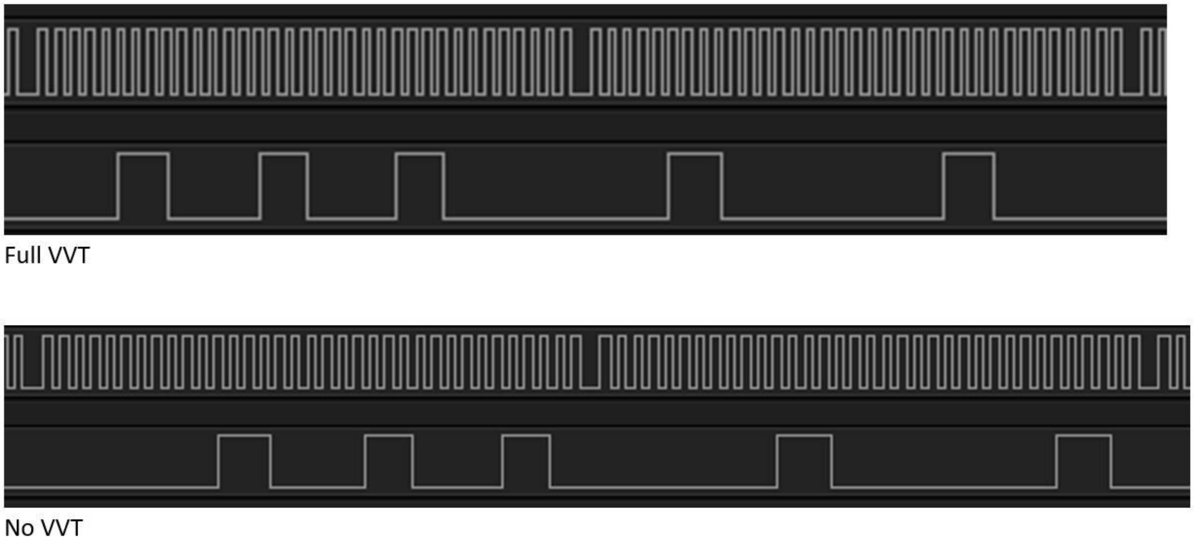


Figure 98: ST170 VVT trigger

Subaru 36-2-2-2

The 36-2-2-2 wheel is common on many 4 and 6 cylinder Subaru engines from approx. 2000 onwards. It utilizes a crank trigger wheel containing a nominal 36 teeth, spaced 10 crank degrees apart, and 3 groups of 2 missing teeth. These missing tooth groupings allow for sync to be determined within at most 1/2 a crank turn.

Early wheels were VR triggered however after the switch to variable valve timing, Subaru switched to Hall sensors. Most configurations are paired with one or two 4-1 cam sensors, however these are not required for sync on The ECU.

> Note that there are 2 variations of the 36-2-2-2 pattern, the H4 and the H6. Whilst visually very similar, the patterns have different groupings of teeth and are not compatible. **Support for the H6 variant of this trigger was added in the 202103 firmware and will not work on earlier versions**

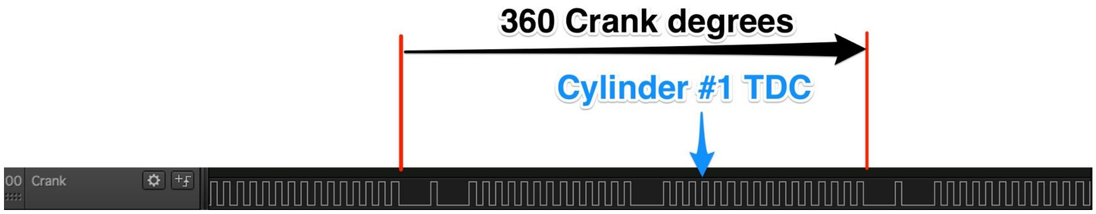
Configuration

- **Trigger Angle:** 0 • **Trigger Edge:** FALLING • **Secondary Trigger edge:** N/A • **Skip Revolutions:** 1
- **Trigger Filter:** Weak (Depending on install)

Trigger Pattern

The3setsof2missingteetharelocatedsuchthatonegroupisonitsownandtheothertwoarelocated adjacent to one another, with a single tooth in between. Sync can be determined by detecting the missing 2 teeth, then seeing if there is another set of missing teeth immediately after it.

Cylinder 1 TDC compression happens on the fourth tooth after the single gap. The ECU watches for any missing tooth period, then waits to confirm whether it is followed by another. Sync can therefore be determined in this manner at 2 points in a single crank revolution.

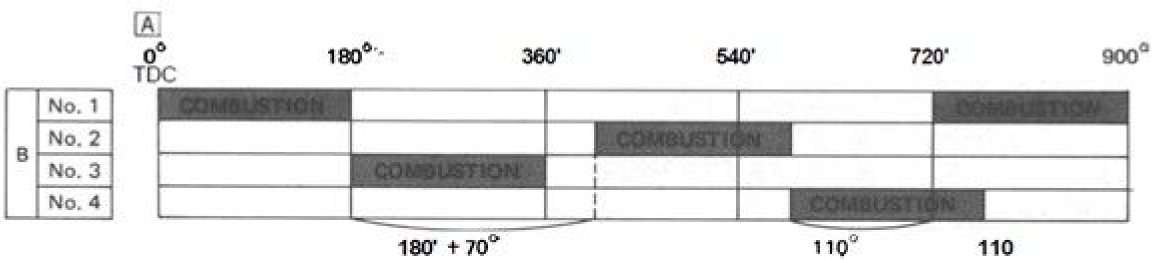


H4 Pattern Note: Many diagrams and trigger wheel images available online show the wheel from the backside, making it show as rotating counterclockwise. For the correct orientation, when looking at the front of the engine, the wheel spins clockwise.

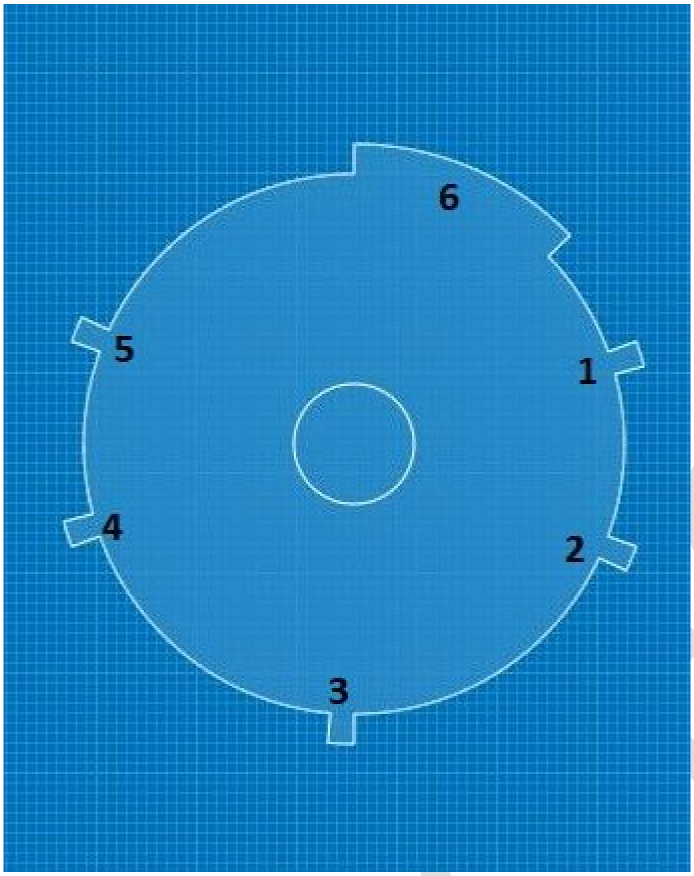
Yamaha VMax 1990+

Overview

TheYamahaVmaxisaV4enginewith70degreesbetweenthecylinderheads. This makes it an oddfire engine since combustion is not always after the same number of degrees. The picture below shows the ignition pattern for this engine:



As can be seen, combustion occurs after 180, 250, 180 and 110 degrees. The early Yamaha Vmax bikes (from -85 to -89) used four pick-ups and a TCI controller, this trigger will not work for the old Vmaxes, perhaps these can use the basic distributor and run off the ignition pulse. From 1990, the Yamaha Vmax uses a digital ignition which has one pick-up and uses the pattern as shown below:



The flywheel runs counter-clockwise and the beginning of lobe 6 (it's left side) is TDC of cylinder 1. To identify all lobes:

- Lobe 1 is the firing point without advance for cylinder 2
- Lobe 2 is the max advance for cylinder 3.
- Lobe 3 is the firing point without advance for cylinder 3 and max advance for cylinder 4.
- Lobe 4 is the firing point without advance for cylinder 4
- Lobe 5 is the max advance for cylinder 1.
- Lobe 6 is the firing point without advance for cylinder 1 and max advance for cylinder 2.

We don't care about the max advance lobes but do use them to sync the flywheel signal. To sync the signal, we have to find the wide lobe (6). This is done by triggering on both the RISING and FALLING edges of each lobe. By determining the time difference, we can find the wide lobe. To quickly sync (reducing cranking time), we want to provide a synced signal as soon as #1 is seen (instead of waiting for 6 to come by again). This is why we start counting from that lobe. To offset the fact that we start the rotation with #1 instead of #6, the trigger angle is set at 70.

Applications

This is the first trigger built to sync on a widelobe (instead of a missing tooth), so it could inspire others do adapt it for similar flywheels.

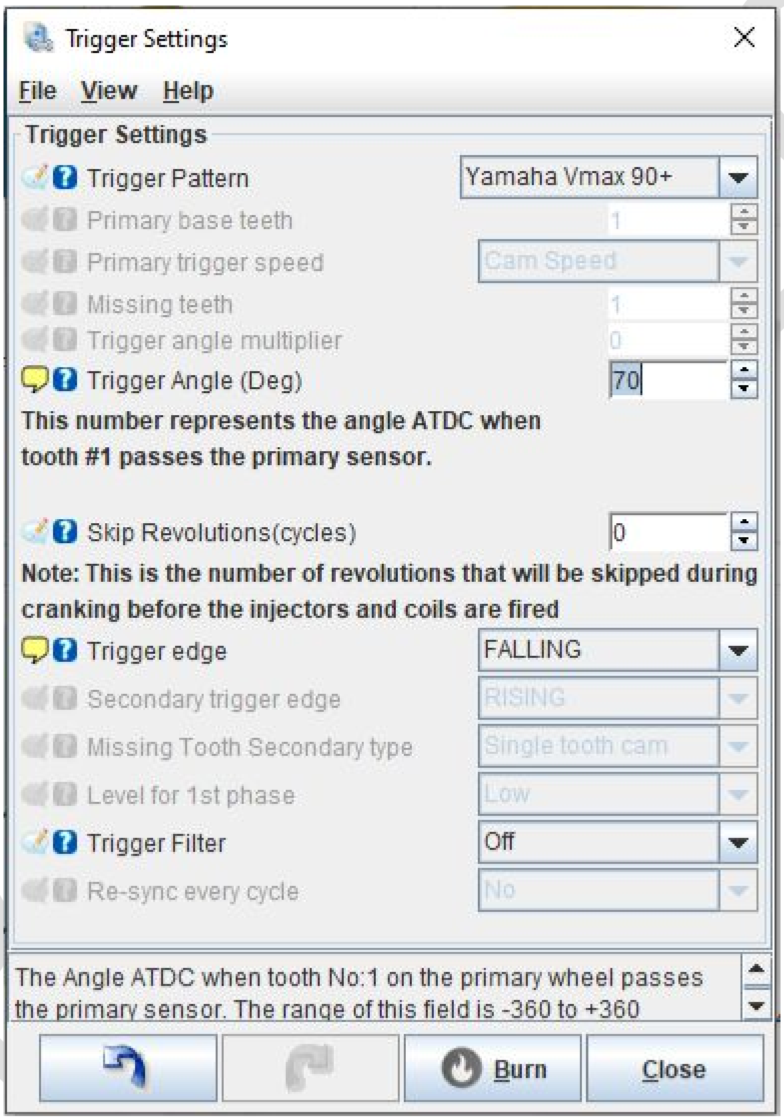
Timing Requirements

The small lobes are 5 degrees, the big one is 45. However, when cranking, the signal might not be strong enough to show the entire 45 degrees of the wide lobe. Therefore, if a lobe is seen that is twice the size of the last, it is considered the wide lobe. To ensure the trigger filter works correctly, the distance between the lobes is taken into account when calculating the triggerFilterTime.

Hardware modification

The signal from the pick-up is quite noisy. Therefore, it requires 10K resistors on the VR+ and VR- line, and the R10 on the VR-conditioner to be equipped with a 220nf ceramic capacitor to filter the generator noise.

Tuner Studio Configuration



Fields:

- **Trigger Angle:** Since we sync on the first lobe after the sync lobe, this should be 70
- **Trigger edge:** For an inverting vr-conditioner, use FALLING, for a non-inverting use RISING
- **Trigger filter:** Set to your preference, but Aggressive is likely going to cause sync issues

Further details

Tested in fuel-only mode, however on the bench (Ardustim) it also shows a good ignition signal. To setup the ignition at the correct odd fire angles, setup channel2 to 180 degrees, channel 3 to 430 degrees and channel 4 to 610 degrees. The wiring for the ignition should adhere to the firing order as shown above.

Sequential operation

No sequential operation is possible since there is no cam signal to work with.

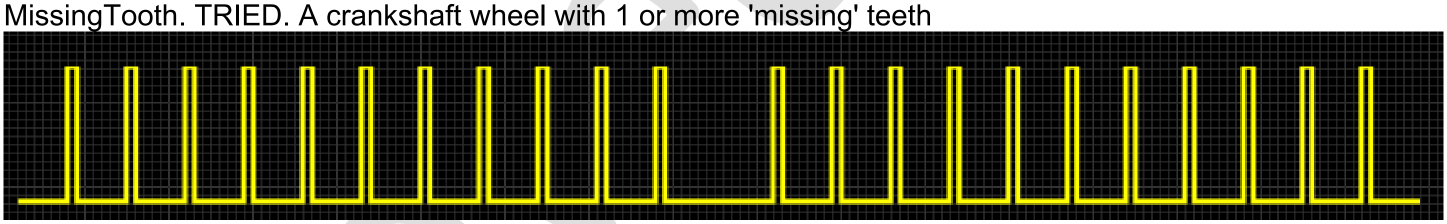
Fast Resume of other Patterns

List of supported crankshaft/shaft patterns

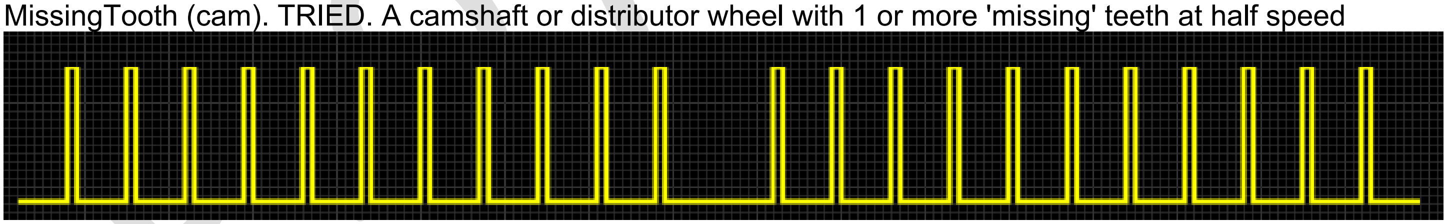
The ECU supports an increasing number of crankshaft and CAS decoders. This includes some of the most common OEM configurations, as well as aftermarket preferred ones (like MissingTooth tone wheels).

The following list includes all that are currently supported. Each leads to a page with details on using the decoder (These pages are a work in progress)

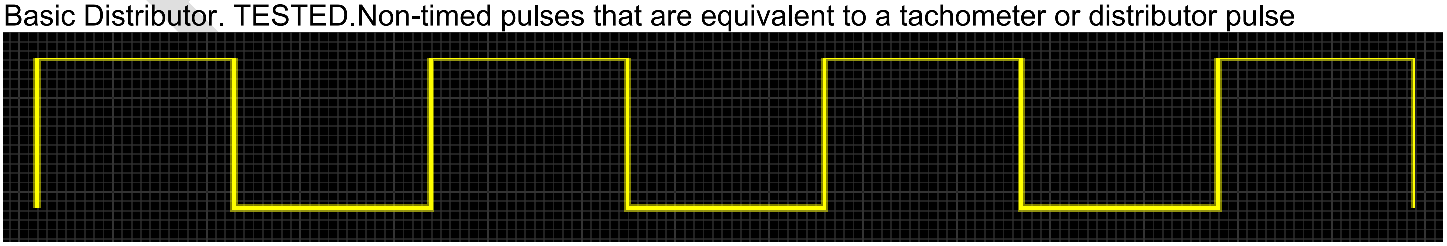
MissingTooth. TRIED. A crankshaft wheel with 1 or more 'missing' teeth



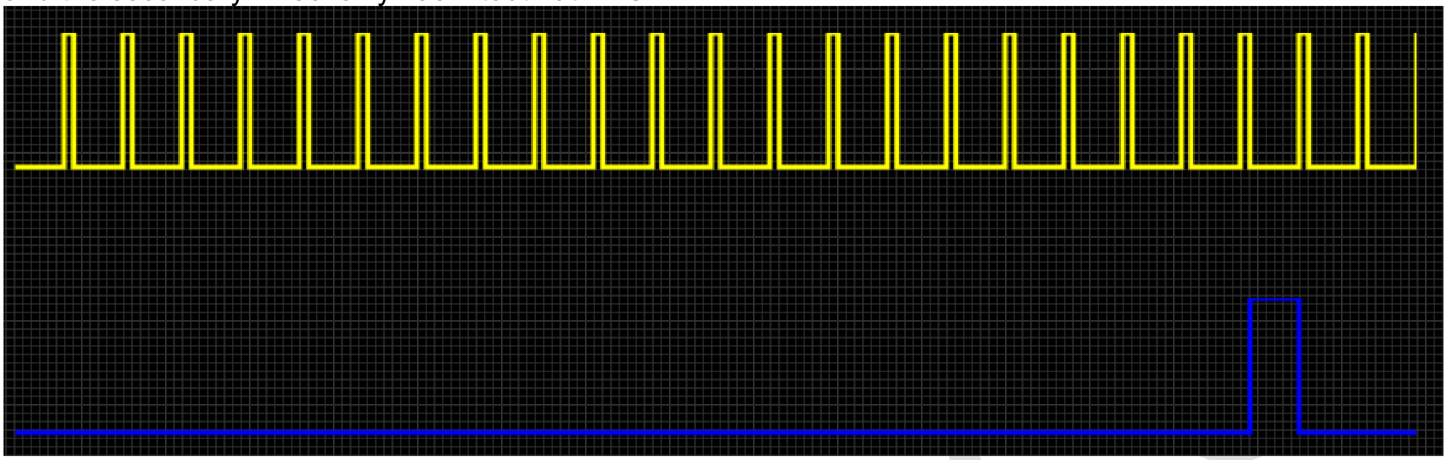
MissingTooth (cam). TRIED. A camshaft or distributor wheel with 1 or more 'missing' teeth at half speed



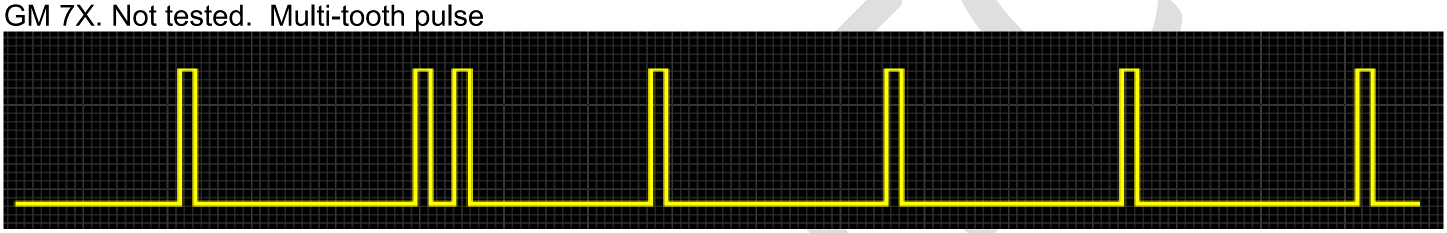
Basic Distributor. TESTED.Non-timed pulses that are equivalent to a tachometer or distributor pulse



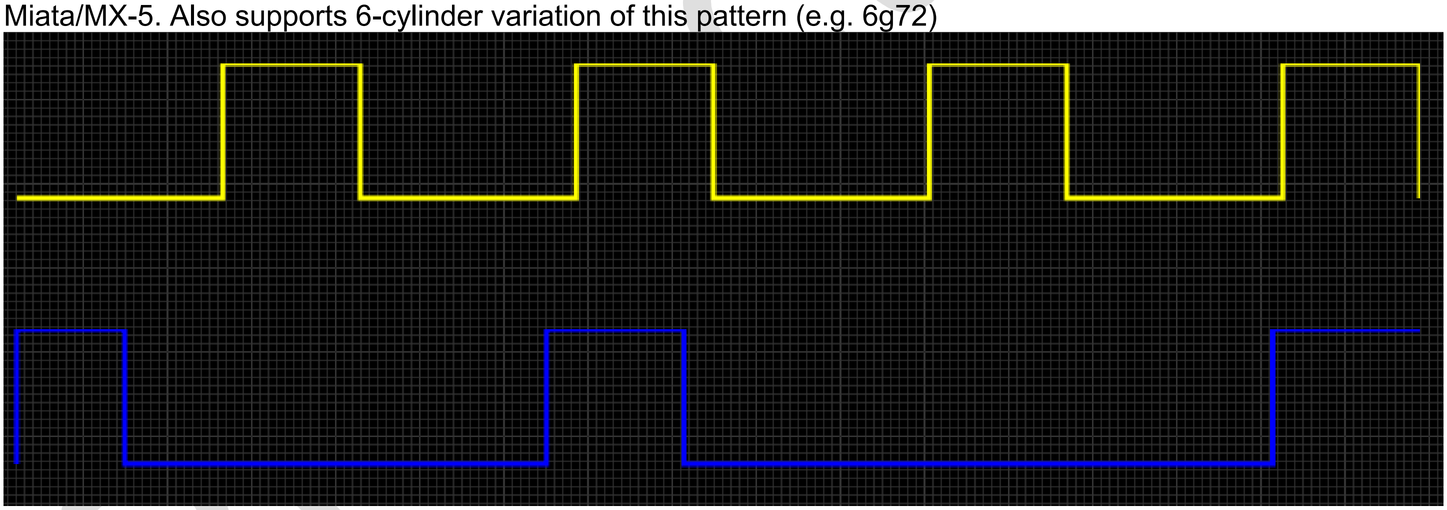
Dual Wheel. TRIED. Two combined signals from two different wheels, the primary wheel has no missing teeth, and the secondary wheel only has 1 tooth at TDC



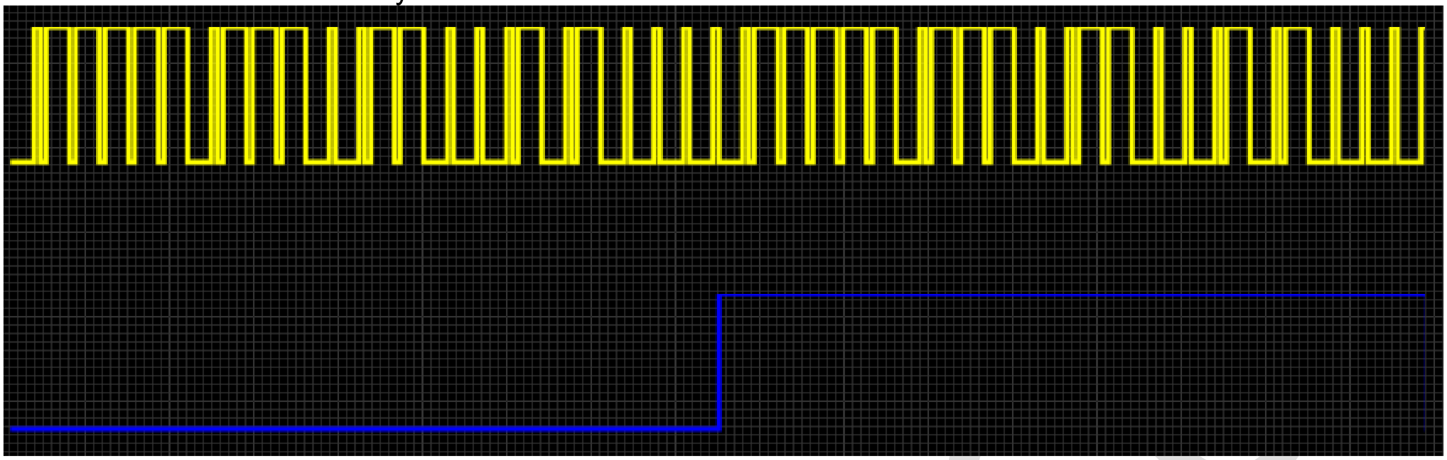
GM 7X. Not tested. Multi-tooth pulse



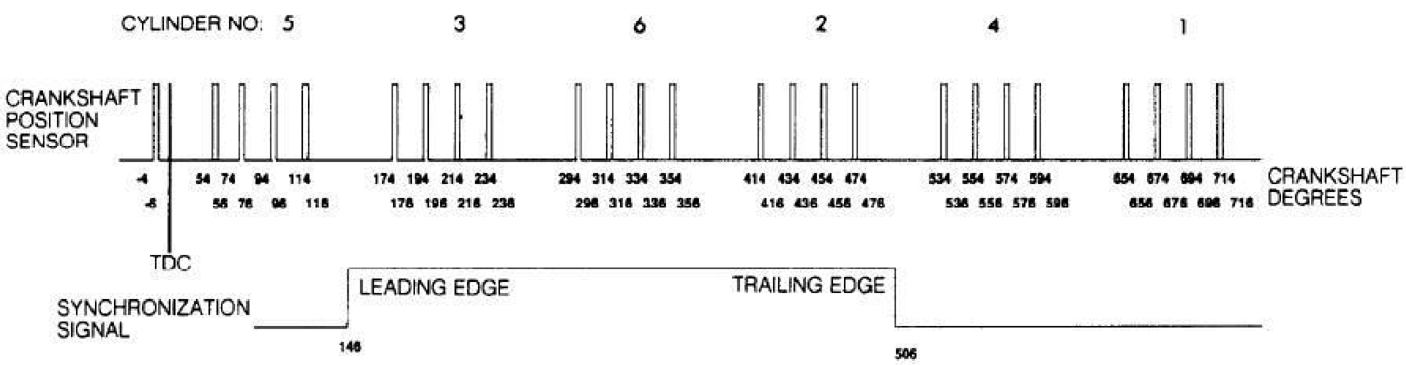
4G63. Complete. Not recommended but functional. As used in many 4 cylinder Mitsubishis and NA/NB Miata/MX-5. Also supports 6-cylinder variation of this pattern (e.g. 6g72)



GM 24X.Not tested. Commonly used on GM LS1 V8

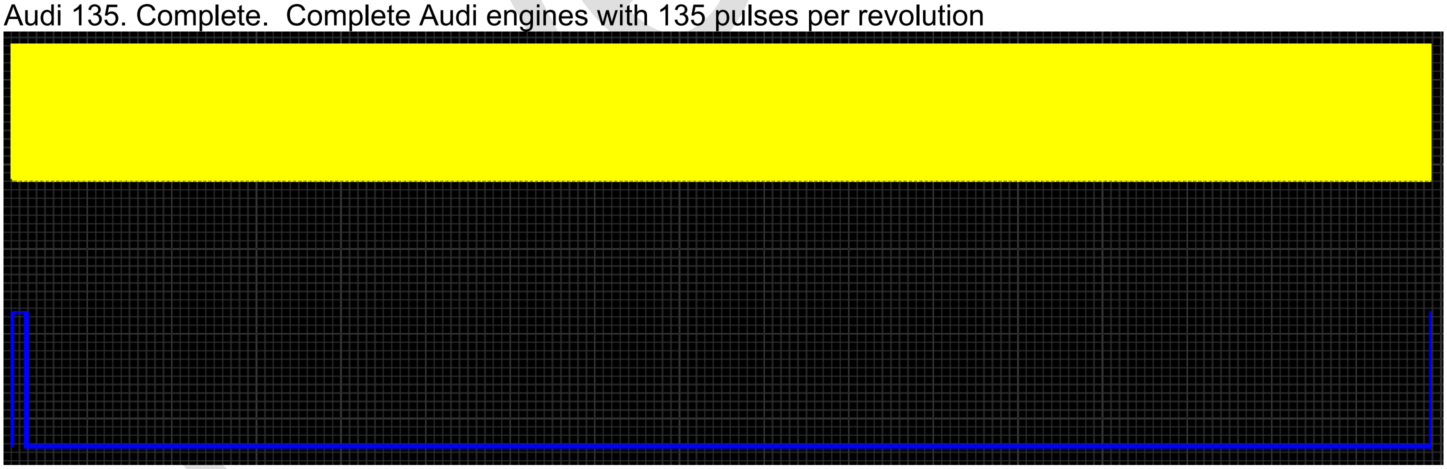


Jeep 2000.Complete. Jeep 6-cylinder engines from '91 to 2000

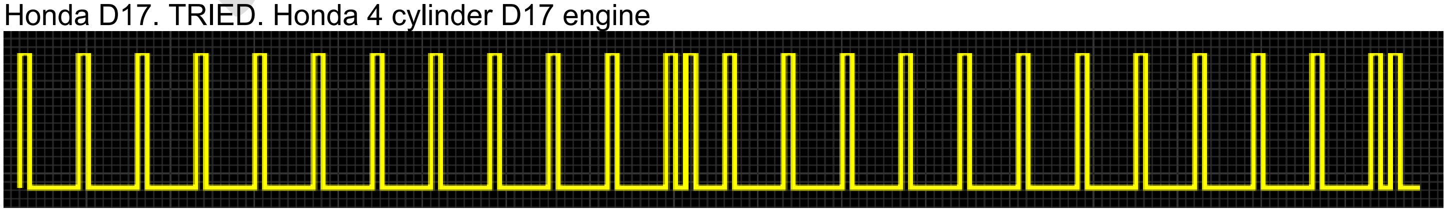


4.0L 6-CYLINDER INJECTION TIMING SCHEMATIC

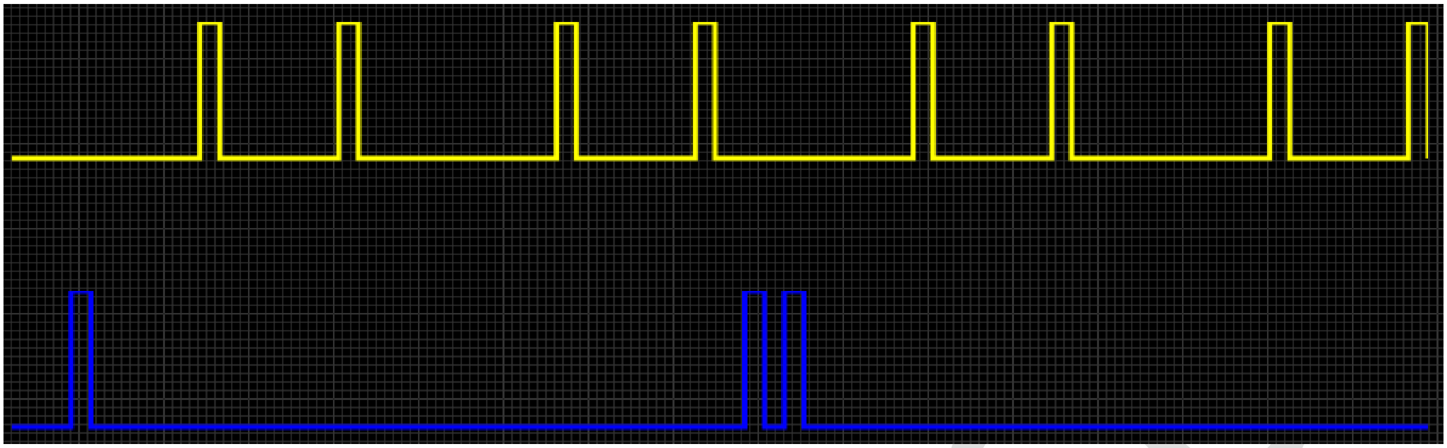
Audi 135. Complete. Complete Audi engines with 135 pulses per revolution



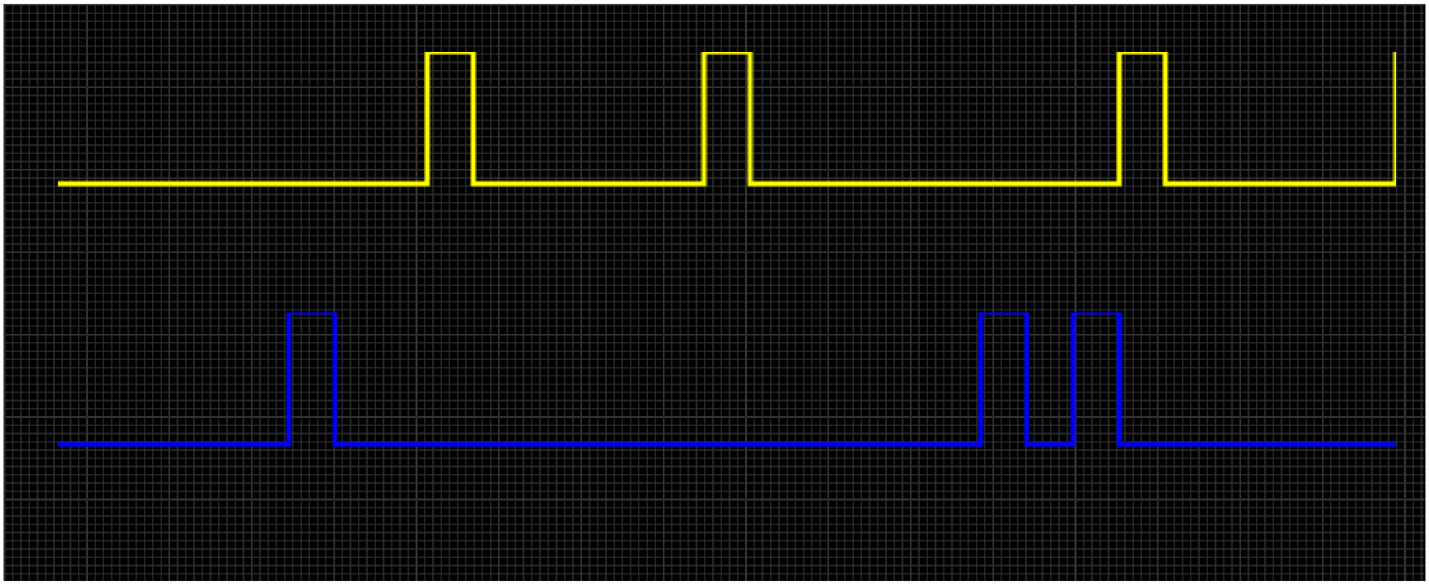
Honda D17. TRIED. Honda 4 cylinder D17 engine



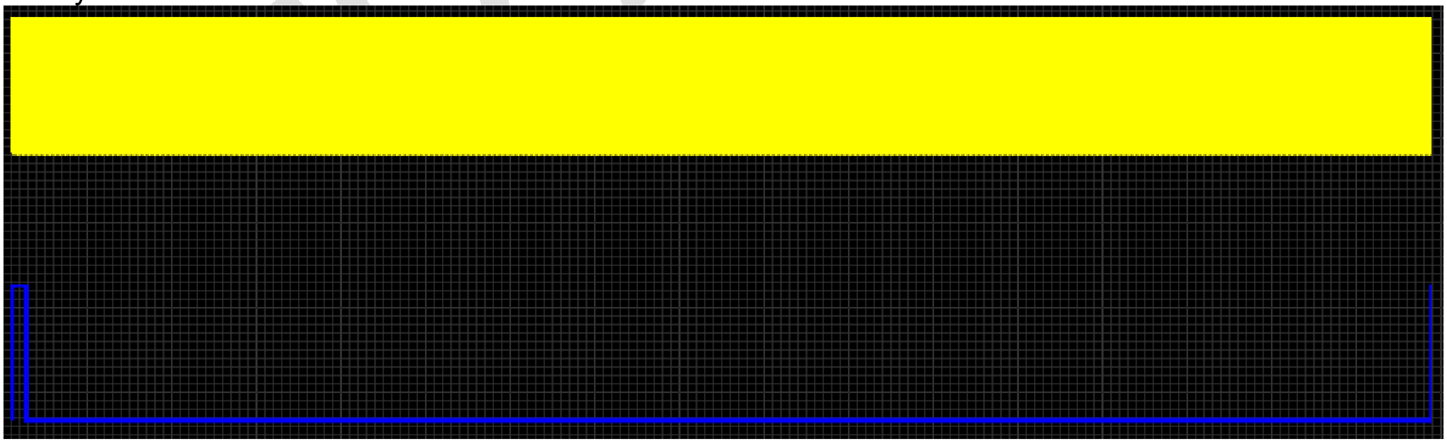
Miata 99.TESTED. 1.8L Miata/MX5 from '99 to '05



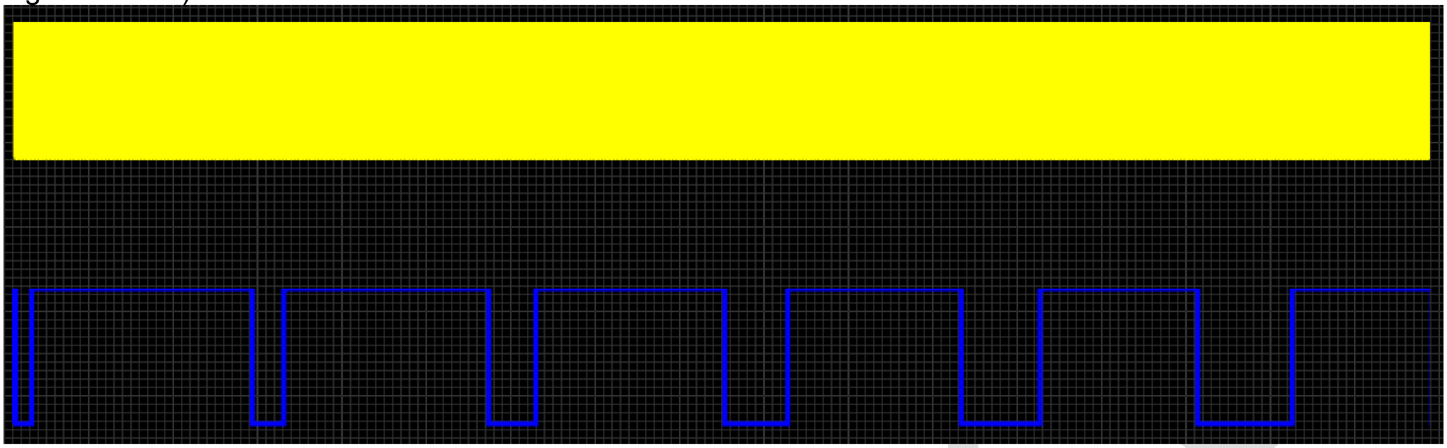
Mazda AU. Not tested. Used in the Mazda 323/Familia/Protegé



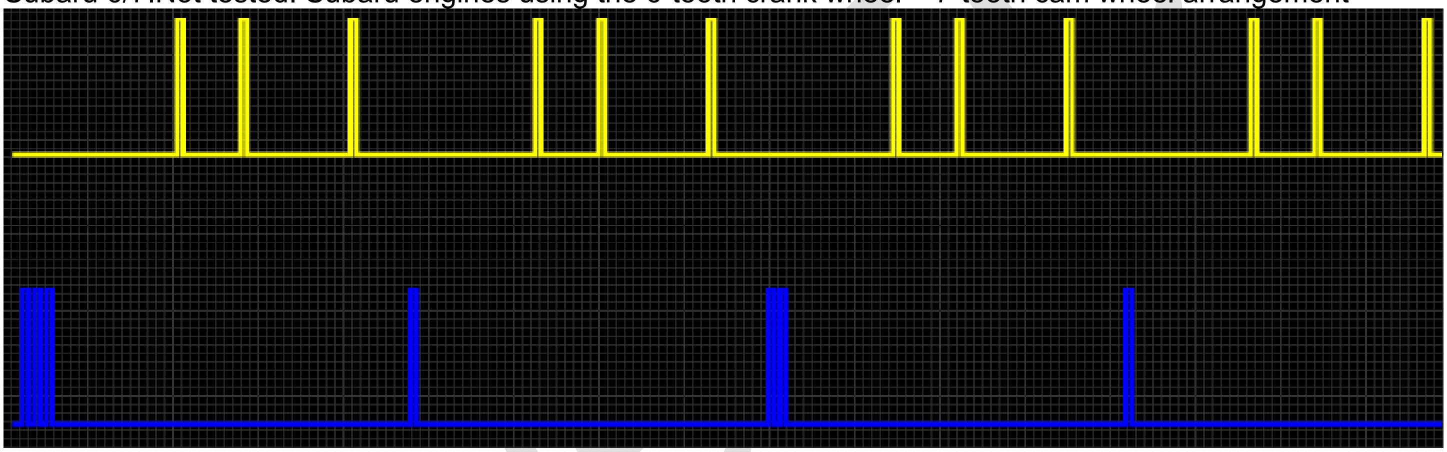
Non-360.Complete. A variation of the dual wheel decoder that can be used with tooth counts that do not divide evenly over 360



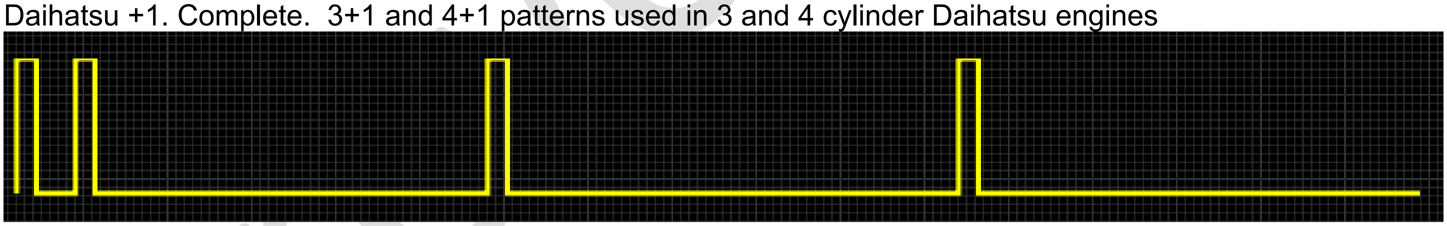
Nissan 360. In progress. 360 tooth cam wheel used on many 4 and 6 cylinder engines. (Only the few pulse signal is used)



Subaru 6/7.Not tested. Subaru engines using the 6-tooth crank wheel + 7-tooth cam wheel arrangement

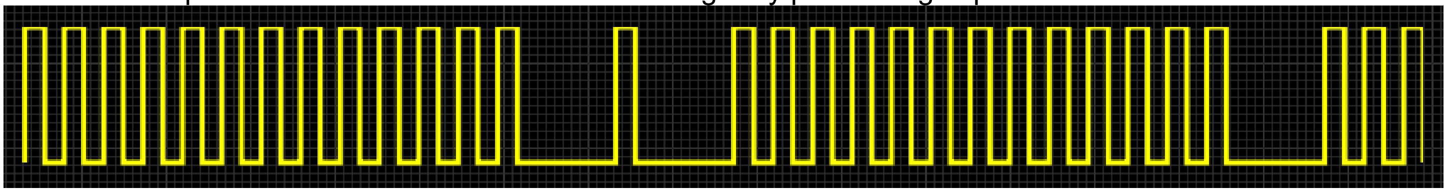


Daihatsu +1. Complete. 3+1 and 4+1 patterns used in 3 and 4 cylinder Daihatsu engines

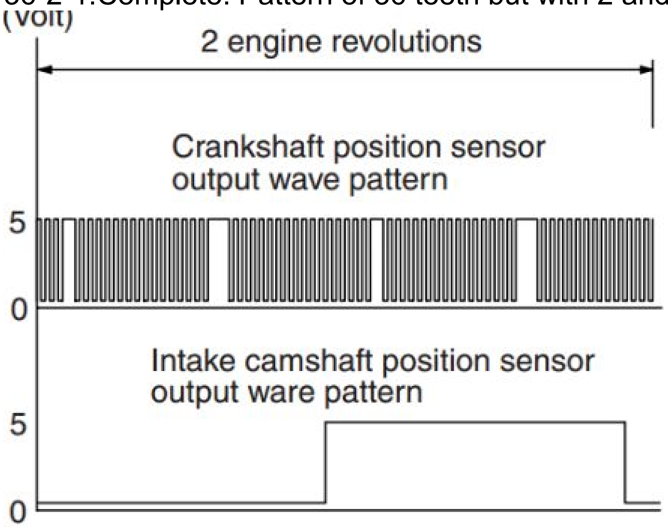


Harley. Complete. The Harley EVO pattern is used on V-Twin engines from '86 to '99. This pattern will work on all injected EVO engines.

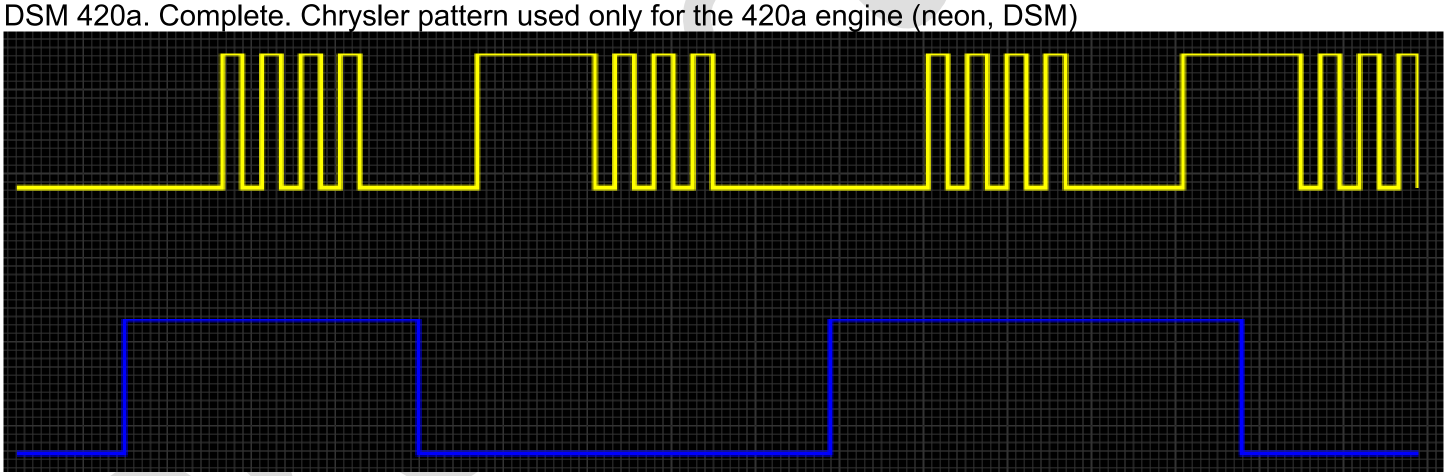
36-2-2-2.Complete. A 30-tooth wheel with three strategically placed large spaces. Used on some Subarus



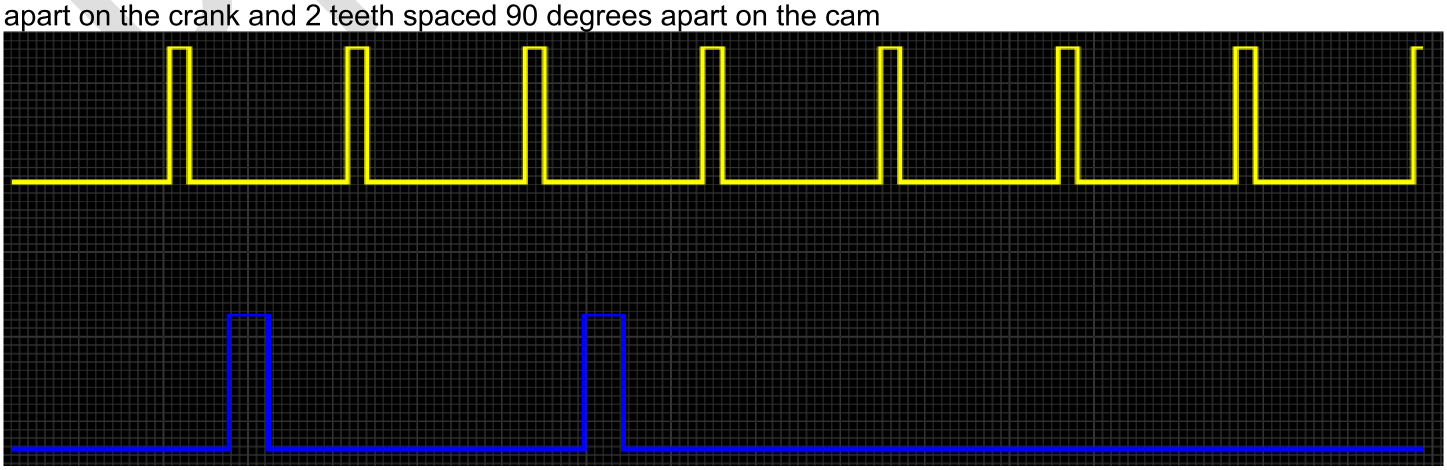
36-2-1.Complete. Pattern of 36 teeth but with 2 and 1 missing teeth. Used on the Mitsubishi 4B11



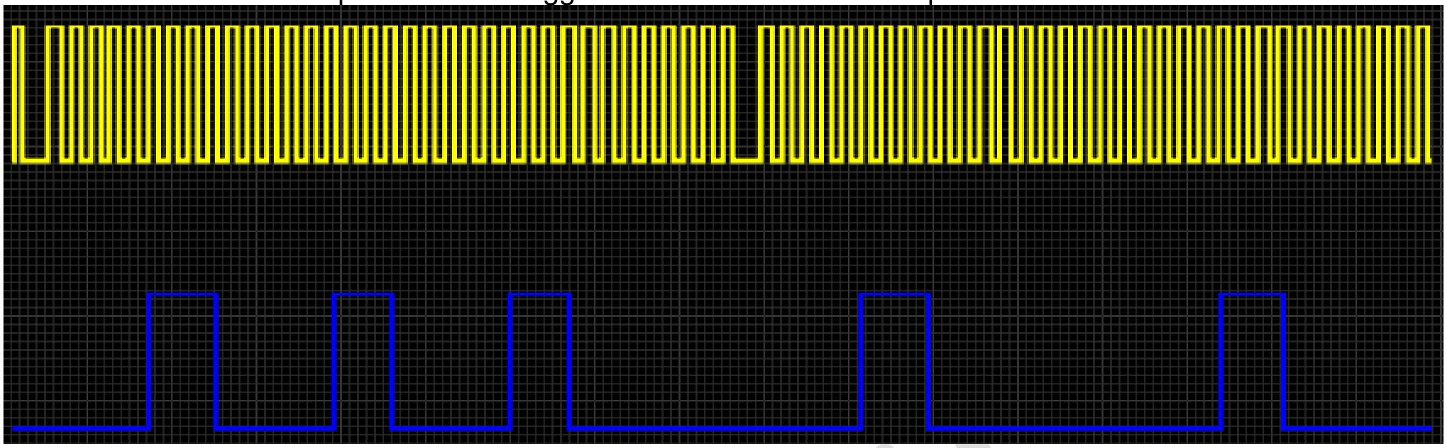
DSM 420a. Complete. Chrysler pattern used only for the 420a engine (neon, DSM)



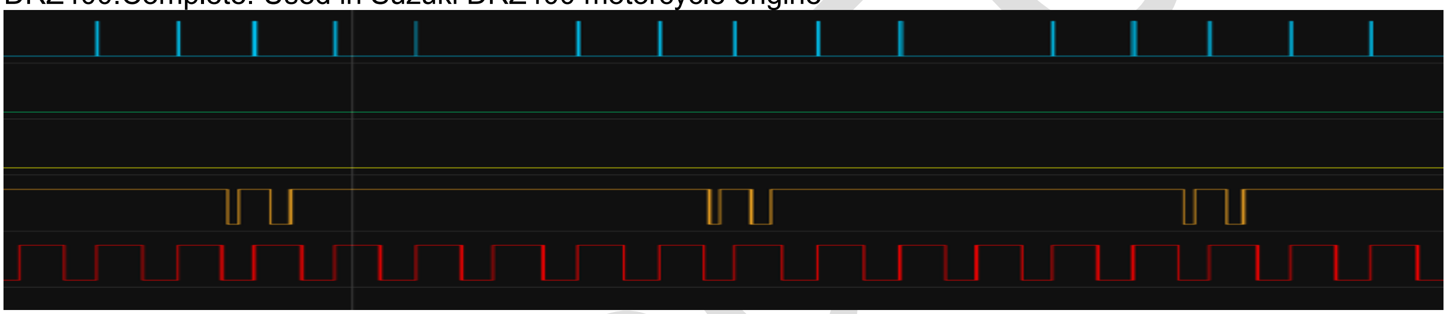
Webber-Marelli. Complete. Weber-Marelli 8+2 pattern configuration with 2 wheels, 4 teeth spaced 90 degrees apart on the crank and 2 teeth spaced 90 degrees apart on the cam



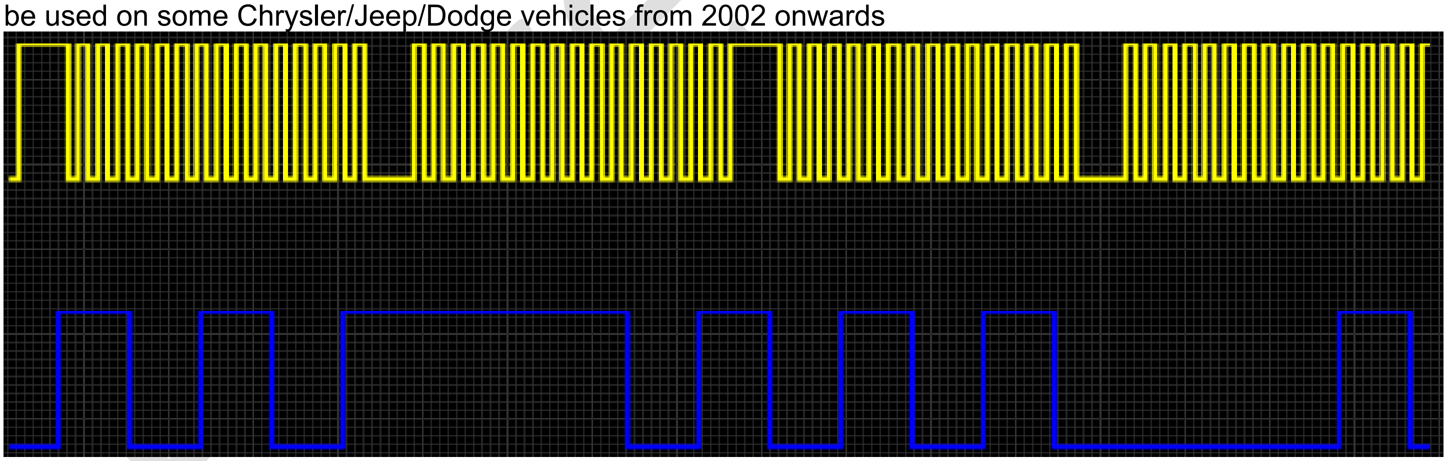
FordST170. Complete. A dedicated pattern for the 01-04 Ford Focus ST170/SVT engine. Standard 36-1 trigger wheel that runs at crank speed and 8-3 trigger wheel that runs at cam speed.



DRZ400.Complete. Used in Suzuki DRZ400 motorcycle engine



NGC. Tested on Bench. For 4, 6 and 8 cylinder engines equipped with the Chrysler NGC pattern. Appears to be used on some Chrysler/Jeep/Dodge vehicles from 2002 onwards



Suzuki K6A. Not tested. This decoder is based on the K6a decoder on a 3 cylinder engine with a trigger pattern on the camshaft consisting on 6+1 teeth. The teeth are not equally spaced on the decoder. 70 degrees prior to TDC on a cylinder their is a tooth.

Limitations
This decoder is designed around the 3 cylinder K6A engine, although the decoder does not hard code the number of cylinders. This should work for other K6 engines with the same trigger pattern but is unproven. Due to the spacing of the teeth on the cam, it is unlikely to work reliably on 4 cylinder engines.

Due to the pattern being cam based and with a low tooth count its essential that tooth base ignition is used.

Setup
Trigger Settings
Select the Trigger Pattern “K6A”.

Set the trigger angle to 0 degrees then use a timing light with locked ignition to establish the exact angle to use. If you change the trigger edge ensure you re-check the trigger angle.

NB you can only setup the “Trigger Angle (deg)” once you have completed all of the setup actions.

Spark Settings (Ignition)
“Enable per tooth timing” should be set to “yes”.

Renix 44 and 66 tooth decoder. Tested. Background
The Renix trigger wheels are required because neither tooth pattern divides exactly into 360 degrees. On the 44 tooth pattern each tooth is 8.1818 degrees wide, whilst the 66 tooth pattern is 5.4545 degrees. For the ECU to work accurately the trigger wheel teeth must divide exactly (be left with a whole number) into 360. Every 11 teeth for either decoder equals a whole number (11 * 8.1818 = 90, 11 * 5.4545 = 60). This trigger wheel therefore works by fooling the ECU into thinking the 44 tooth pattern only has 4 teeth (one every 90 degrees) and the 66 tooth pattern has 6 (one every 60 degrees).

Limitations
Both 44 and 66 tooth trigger wheels have nothing to indicate a unique point within 360 degrees. This means we can't establish if we're at 0 or 180 degrees. This presents two limitations,

This decoder can't do wasted spark or sequential ignition. If you require ignition you need to keep the distributor.
Fuel injection can not be timed to match the valve timing. You should therefore select as a minimum two squirts per engine cycle to ensure the injection event is within 180 degrees of ideal. This should be the ECU default. Injection timing is less critical than ignition timing so for none race vehicles this is acceptable.

Both 44 and 66 tooth use the same implementation. Whilst 44 tooth has been tested for many months before inclusion into the firmware the 66 tooth hasn't so could potentially have issues. If problems are found please log an issue on Github for it to be investigated.

Trigger setup - it is essential the trigger setup is correct for the decoder to work. Please ensure your tooth log matches the screen shot below before requesting help with the decoder.

Setup
Engine Constants
Squirts per Engine Cycle is set to 2 or higher

Number of cylinders dictates your trigger pattern.

4 cylinders equates 44 tooth pattern.
6 cylinders equates 66 tooth pattern

Trigger Settings
Select the Trigger Pattern “Renix”. A number of existing settings will be greyed out but display the values prior to the Renix pattern being selected. In the background the system sets up these values as it requires them.

NB you can only setup the “Trigger Angle (deg)” once you have completed all of the setup actions. To set this you should follow the standard guidance elsewhere in the Wiki.

Trigger Sensor (VR or Hall)
Due to the way the teeth are setup on the decoder its very important if a VR sensor is used the wiring, VR decoder and trigger edge are setup correctly. You are targeting a tooth log that looks like the one below.

Spark Settings (Ignition)
The ECU can currently only support spark output mode “Single Channel”. This means the sparks need to be routed via a distributor.

“Use new ignition mode” should be set to “yes”.

Future Enhancements
Some Renix installations also have available a cam signal. When this decoder was developed details of the cam signal timing were not available. If enough details can be provided across enough engines to prove the location of the cam signal is similar (within 180 degrees) this can be added into a future release. The addition of a cam signal would then allow wasted spark and sequential ignition and fuelling.

Rover MEMS decoder. Tested. Rover developed MEMs (Modular Engine Management System) which was used on a number of British vehicles with petrol engines. This included ‘A’ series, ‘O’ Series, ‘T’ series and ‘K’ Series engines. Due to the range of engines being used by a number of manufactures you can find the MEMS solution in Rovers, LandRovers, Caterham's, Elise's, Morgan's and many kit cars.

During its lifetime Rover implemented multiple trigger patterns with the MEMs system. This decoder supports the 5 known patterns and will automatically identify and decode the pattern. The patterns are described below based on the number of teeth that occur sequentially broken up by gaps in the pattern. The gaps are represented by the ‘-’.

3-14-2-13-
2-14-3-13-
11-5-12-4-
17-17-
7-10-6-9-

The decoder also adds supports for the secondary (cam) 5-3-2- tooth pattern used by Rover as well as normal single tooth patterns.

Additionally, functionality has been added to the 17-17- pattern so if a secondary (cam) signal is present it will allow accurate spark signal to be generated at the appropriate time. The secondary signal needs to be between 360 and 720 degrees in the 720 degree cycle.

Limitations
The pattern with 17-17- can not do wasted spark due to no unique identifier within the 360 degrees engine cycle. If you require spark this must be done via a distributor with ignition set to single channel. This pattern is also limited on fuelling and must be used with at least 2 squirts per engine cycle.

The code currently only supports 4 ignition on cylinder engines, if their is a demand for 6 or 8 cylinder please log an ‘issue’ in github and the code can be extended.

Setup
Engine Constants
Squirts per Engine Cycle is set to 2 or higher for the trigger pattern 17-17-

Trigger Settings

Select the Trigger Pattern “Rover MEMS”. A number of existing settings will be greyed out but display the values prior to the Rover MEMS pattern being selected. In the background the system sets up these values as it requires them.

NB you can only setup the “Trigger Angle (deg)” once you have completed all of the setup actions. To set this you should follow the standard guidance elsewhere in the Wiki.

Spark Settings (Ignition)
If you have the trigger wheel 17 teeth gap 17 teeth gap the ECU can currently only support spark output mode “Single Channel”. This means the sparks need to be routed via a distributor.

All other patterns can support wasted spark and if a secondary trigger edge (cam) signal is present support sequential.

“Enable per tooth timing” should be set to “yes”.

Toyota 3 tooth cam
Availability
Under Development - tested on Lexus I6 engines successfully, waiting on V8 dual cam VVT2 testing to complete.

Background
Extends the missing tooth decoder used on Toyota engines to include decoding the 3 tooth cam pattern for both cam1 and cam2. This includes supporting VVT1 and VVT2.

Limitations
to be confirmed.

Setup
missing tooth decoder, select Toyota 3 tooth decoder.