

# ProjectECU Manual

This manual covers the hardware (sensors, wiring, etc.), software configuration, and setting items related to the operation of the ProjectECU. When getting started with ProjectECU, especially if this is your first time installing and configuring an electronic engine management system, this manual will help you understand the capabilities of ProjectECU and how it should be installed, both in terms of hardware and software/firmware.

While this document will help provide information related to ProjectECU setup, it does not cover advanced engine tuning, fuel/ignition strategies, etc. For this we will be having free video talks on the ProjectECU page

## Getting

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

**A ProjectECU board-** This is the muscle of the ECU and contains all the controllers and IO circuitry. Also this is the brain of the ECU and contains the processor, memory and storage. ProjectECU connects to interface with the vehicle's wiring harness

**Firmware:** This is the system software that runs on the processor 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 proceeding to installation on the vehicle. The configuration of the software in ProjectECU can be completed without the need for it to be installed in the vehicle and this allows the software and options available to be explored.

## About This Manual

This documentation is continually growing and this means that you may find gaps in the documentation where little information is currently provided. Feel free to post in the ProjectECU group if there's anything you critically (or even not so critically) need missing.

## Getting Started with ProjectECU ProjectECU

hardware and software can be configured in many ways, so understanding the requirements and configuration steps can be daunting initially.

The links below will help you get an overview of each area, how it should be configured, and how it relates to the general configuration.

Hardware requirements: what hardware you will need to work with ProjectECU (sensors, wiring, injectors, coils, etc.)

Specific ECU information:

ProjectECU Spartan

ProjectECU Titan

ProjectECU Gladiator

ProjectECU Carbumate

*ProjectECU Mercenary*

*ProjectECU Blackbox*  
*ProjectECU Mini*

Working with TunerStudio software

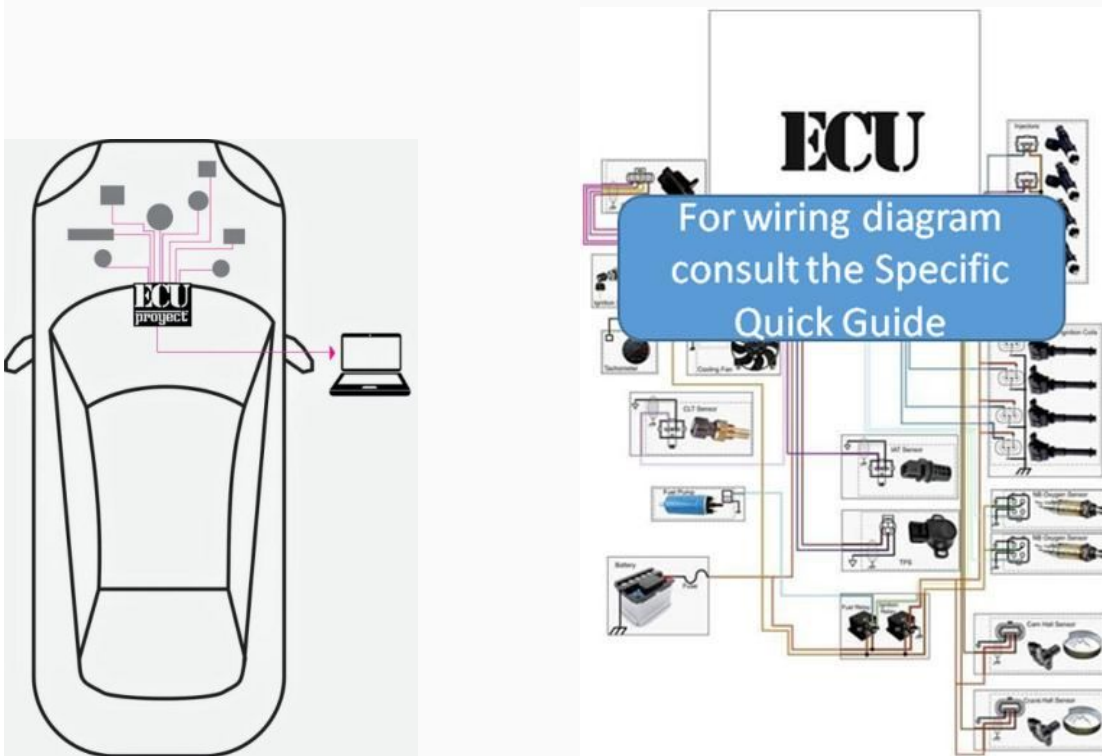
Connecting to TunerStudio

Configuring a Project in TunerStudio

# System Wiring Diagram Wiring

## Guide

At a glance, ProjectECU 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 ProjectECU Quick Guide).



## Function Specific Diagrams

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

## Injector Wiring

### Introduction

ProjectECU contains 2 to 4 injector control banks and is capable of supporting up to 8 cylinders with these.

## Supported Injectors

ProjectECU supports High-Z injectors (also known as 'high impedance' or 'saturated' injectors) natively. Low Z injectors are supported by adding resistors connected in series with the signal wires. High Z injectors are typically those with a resistance greater than 8 ohms.

If lower impedance "Low-Z" ("Peak and Hold" or PWM controlled) injectors are used, the wiring will require series resistors on each injector to avoid damaging the injectors or the board with excessive current. The resistance of ohms and watts can be calculated according to Ohm's law, or use a calculator page on the internet.

## Design

There are several ways to connect the injectors depending on your configuration and preference.

### 1, 2, and 3 injectors

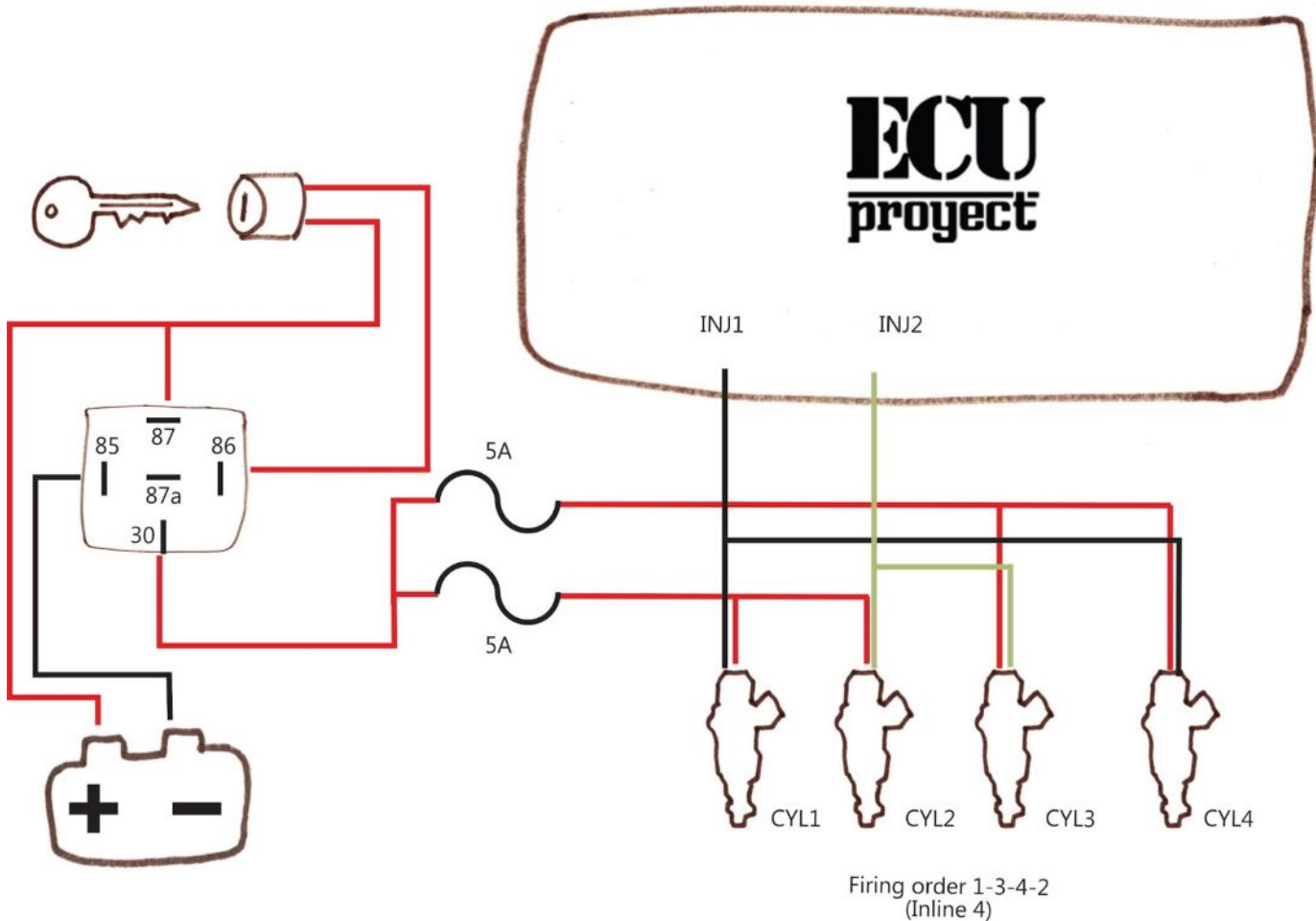
For these configurations, each injector is connected to its own output on the ProjectECU board.

### 4 injectors

For 4 cylinders, there are 2 ways to connect them to ProjectECU:

#### Method 1 (semi-sequential)

The standard method is the same as the one used for 6 or 8 cylinder configurations, where 2 injectors are connected to each injector channel. In this configuration, only 2 injector channels will be used. Matched injectors must have their top dead center (TDC) centers 360 crankshaft degrees apart.

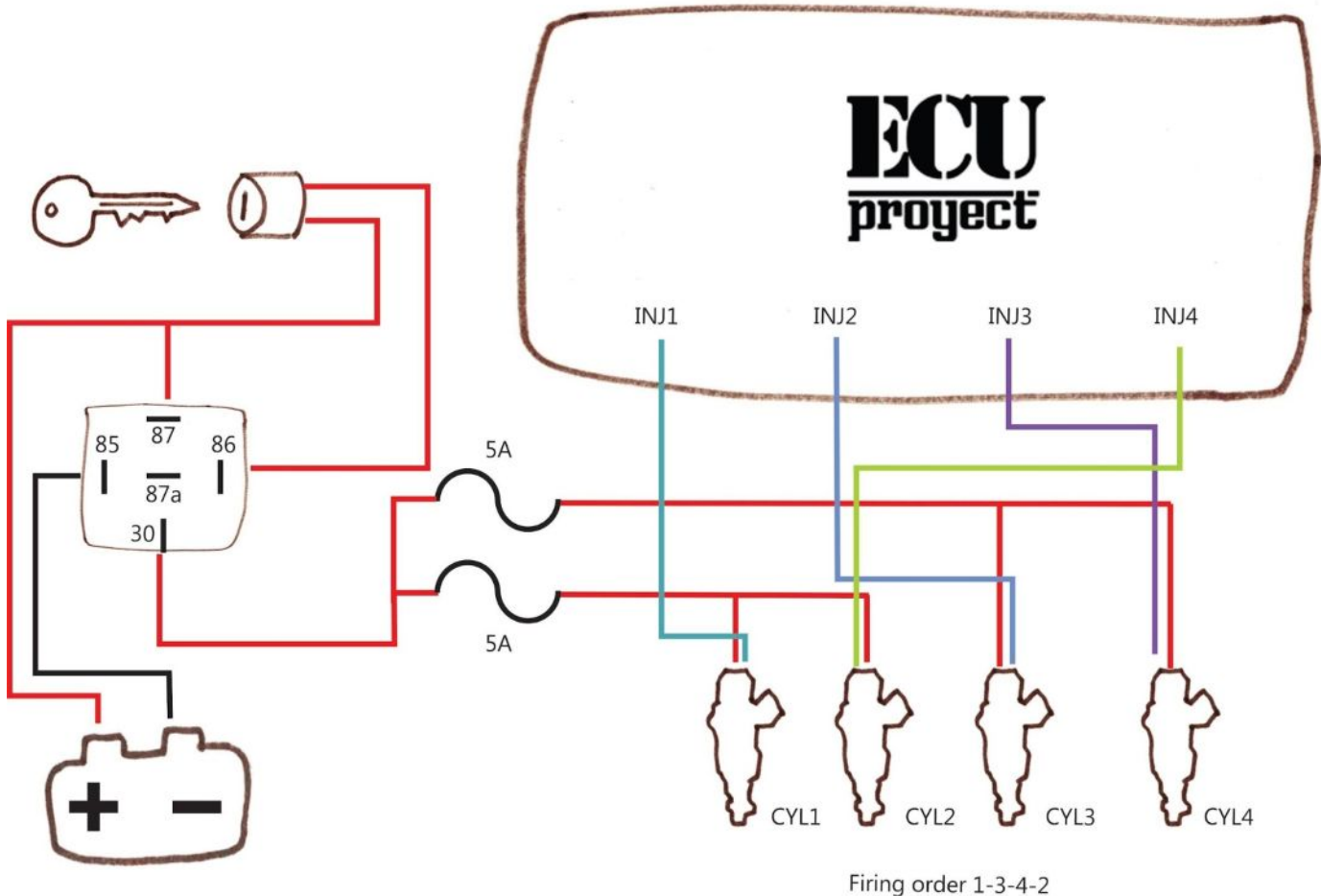


#### Method 2 (full sequential)

**WARNING!** If this mode is activated and you do not have a shaft sensor, the ecu will not have RPM detection as it will be waiting for the shaft position reading. **ONLY SELECT IF YOU HAVE A TREE SENSOR CONFIGURED AND CONNECTED!**

This method is only available on 4 cylinder / 4 injector applications and allows you to connect 1 injector per channel. Injector channels always fire in numerical order (ie 1, 2, 3, 4), so your injectors must be connected to account for their firing order. Within TunerStudio, this option can be enabled by selecting:

Settings -> Engine Constants -> Injector Timing -> Semi-sequential



### **5 injectors**

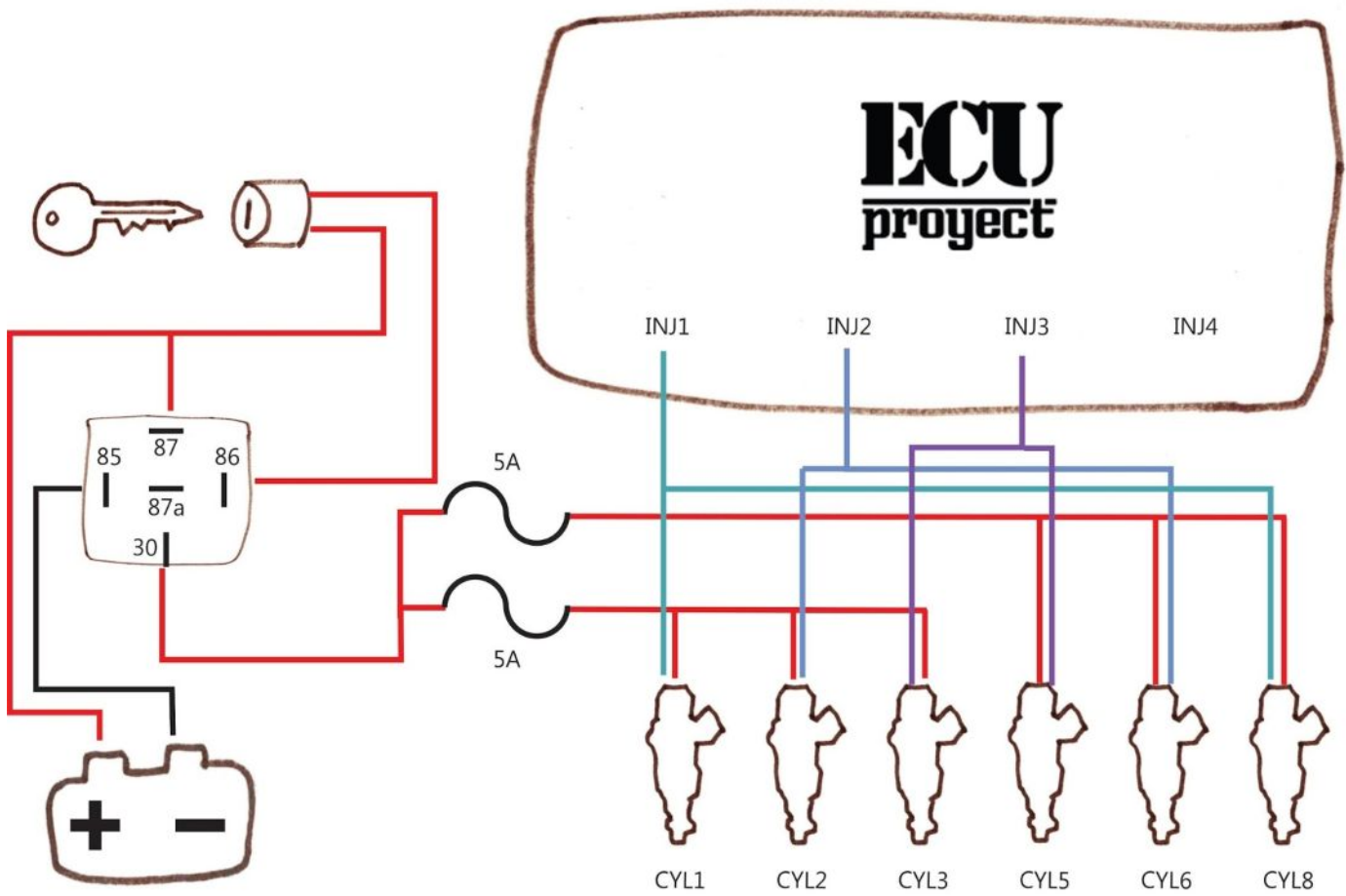
5 cylinder setups must be connected to use all 4 injector outputs with 2 injectors sharing the output #3. For the typical 5-cylinder in-line firing order (1-2-4-5-3), injectors 4 and 3 would join at the outlet of injector 3.

### **More than 5 injectors.**

For configurations with more than 4 injectors, the number of outlets used will be equal to half the number of injectors.

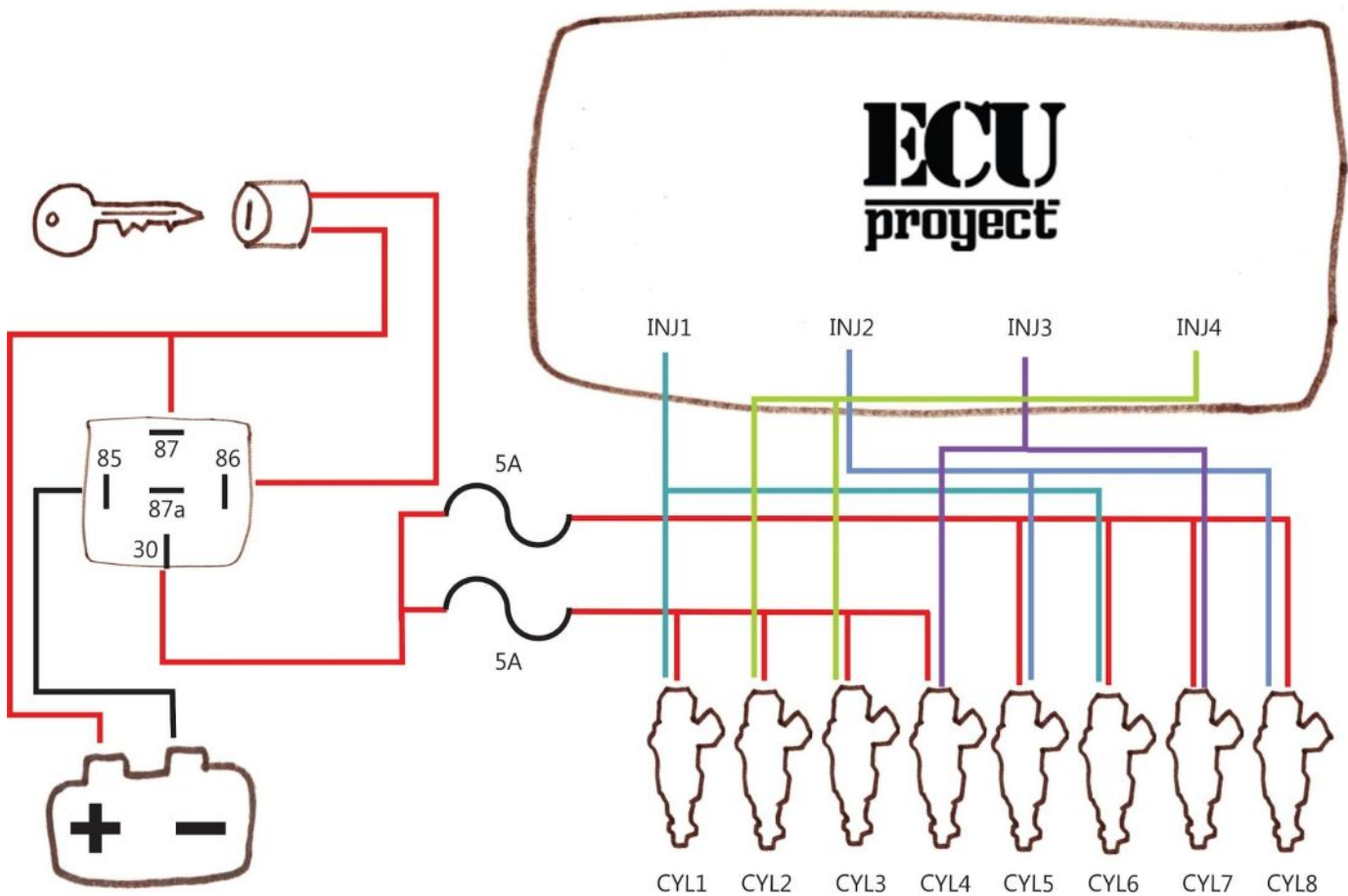
Examples:

### **6 cylinders**



Firing order 1-5-3-6-2-4  
(Inline 6)

8 cylinders



Firing order 1-8-7-2-6-5-4-3  
(Eg LS V8)

In line with the above, this configuration requires each injector output to be connected to 2 injectors. The injectors must be grouped in opposite pairs, that is, cylinders whose top dead centers are 360 degrees apart.

## Ignition Wiring 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 stems from the large number of different ignition types available, with significant changes in hardware used in the late 1980's and throughout the 1990's compared to newer designs.

While this guide does not cover all ignition styles and hardware, it does cover the most common scenarios. It is generally recommended (wherever possible) to use newer styles of ignition hardware (typically "smart" coil or close coil) rather than using separate plug-in modules.

### WastedSpark

Wasted Spark is a common means of spark control that requires only half the number of ignition outlets as there are cylinders, with 2 cylinders connected to each outlet. For example:

4-cylinder engine requires 2 ignition outlets

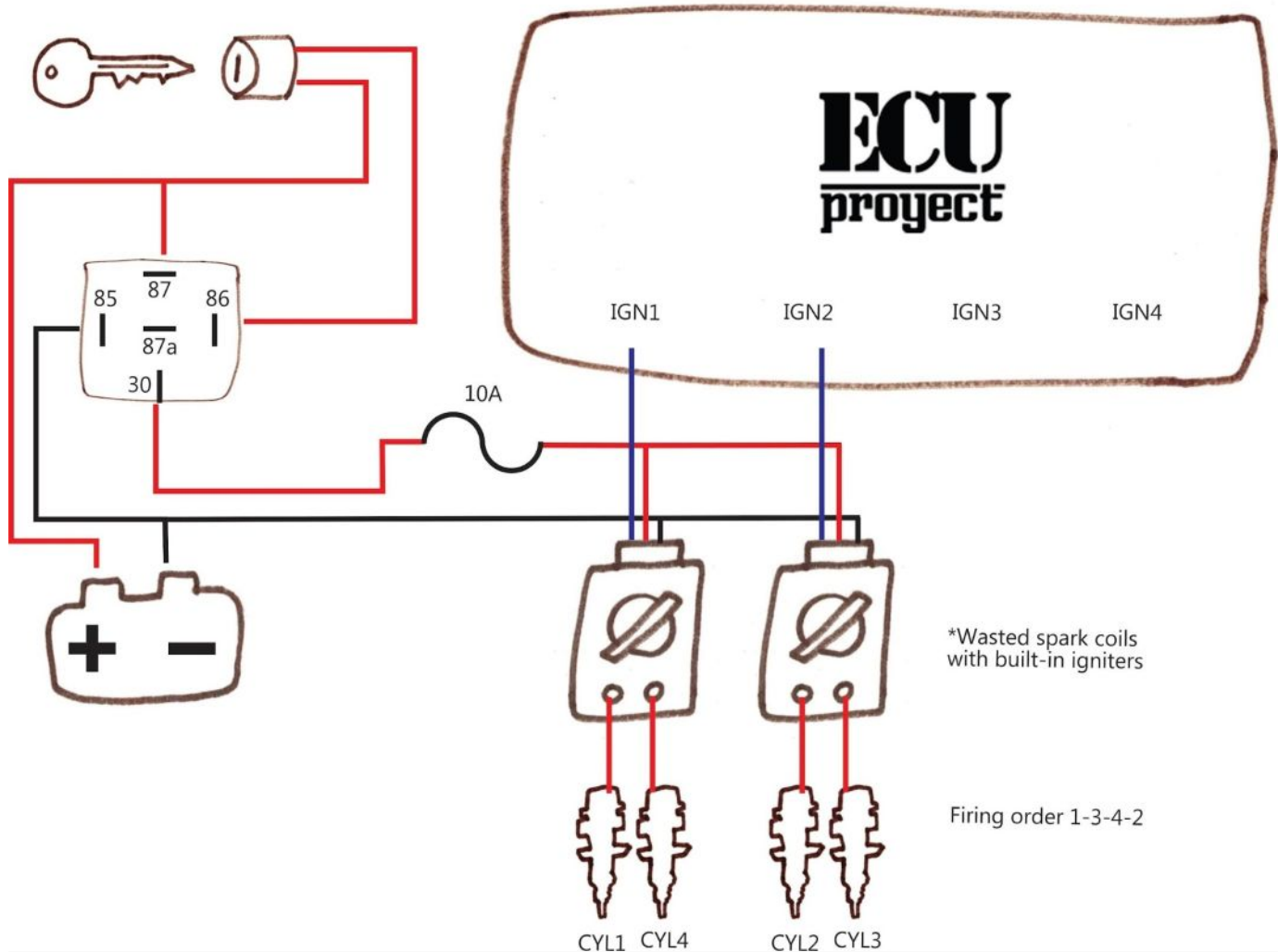
6-cylinder engine requires 3 ignition outlets

8-cylinder engine requires 4 ignition outlets

WastedSpark has the advantage of not requiring any cam signals or inputs as it does not know the phase of the motor. This is possible by firing the ignition outputs once per revolution and matching that output to 2 cylinders that are both at TDC (with one cylinder on the compression stroke and the other on the exhaust)

When using a wasted spark it is critical that the correct pairs of coils and/or spark plugs are mated.

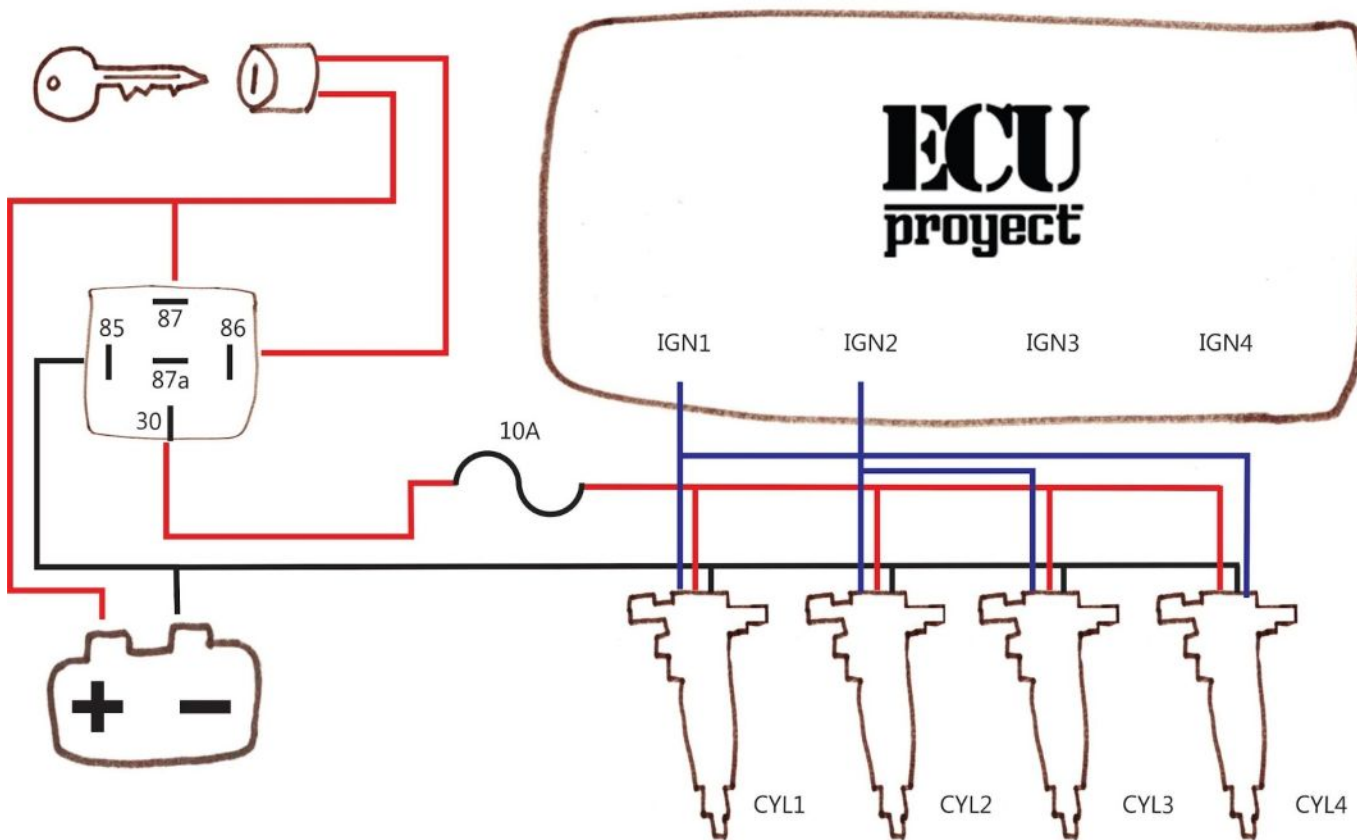
There are many two pole wasted spark coil packs available with and without integrated igniters. Either is suitable for use with ProjectECU, but the use of coils with integrated igniters is recommended.



Note: The example above uses 'smart' coils with integrated igniters. DO NOT connect high current (dumb) coils without adding an ignition module or pickup.

### **Coilon Plug (Pencil Coils or Single Coils)**

As an alternative to a double pole wasted spark coil, single coils can be used on plug units in a wasted spark configuration.

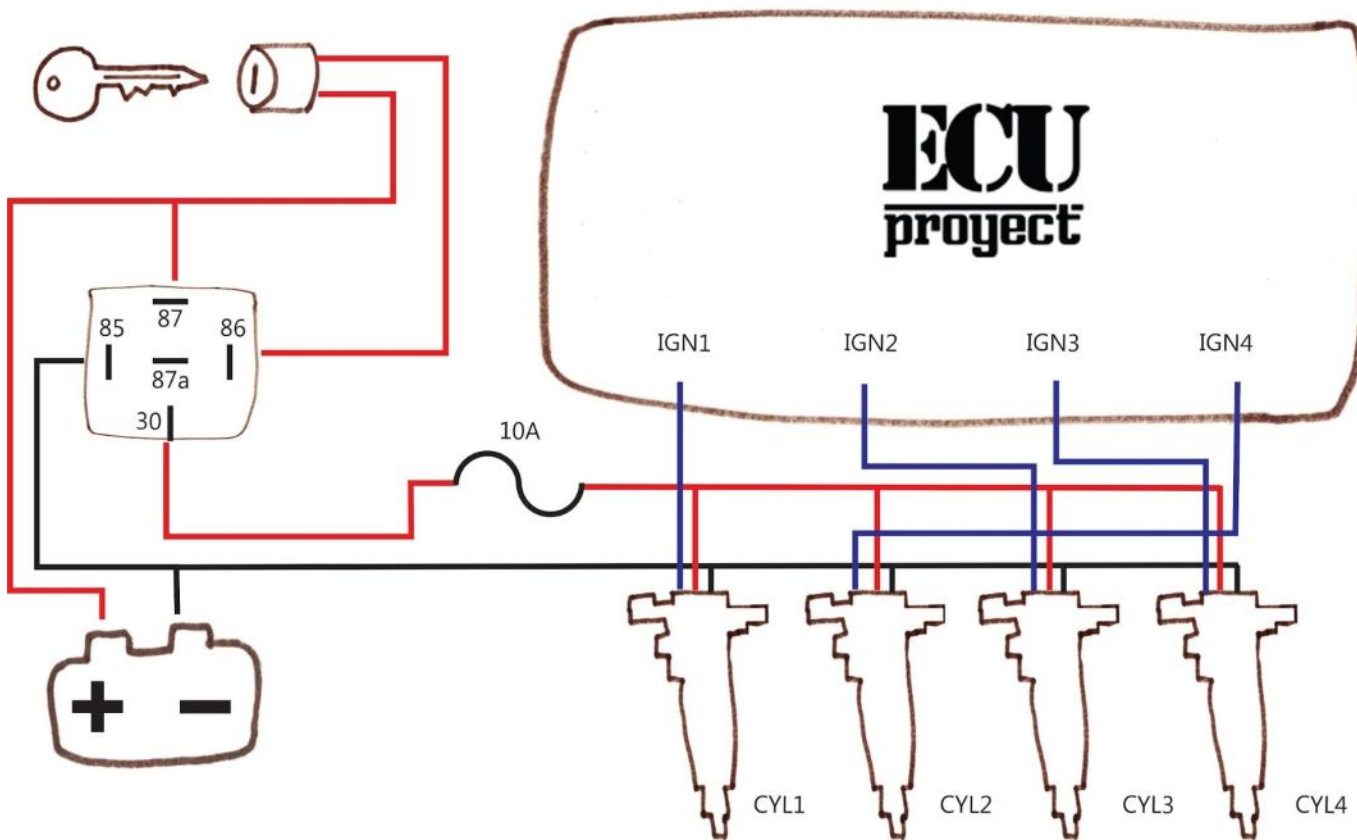


Firing order 1-3-4-2

Note: The examples above use 'smart' coils with integrated igniters. DO NOT connect dumb COPs (2-pin) without adding a

### **Sequential Ignition Module (COP) Sequential**

control using individual coils or Coil-on dramatically simplifies ignition wiring. With this configuration, each coil (and later each cylinder) connects to a single ignition output, connected in firing order.

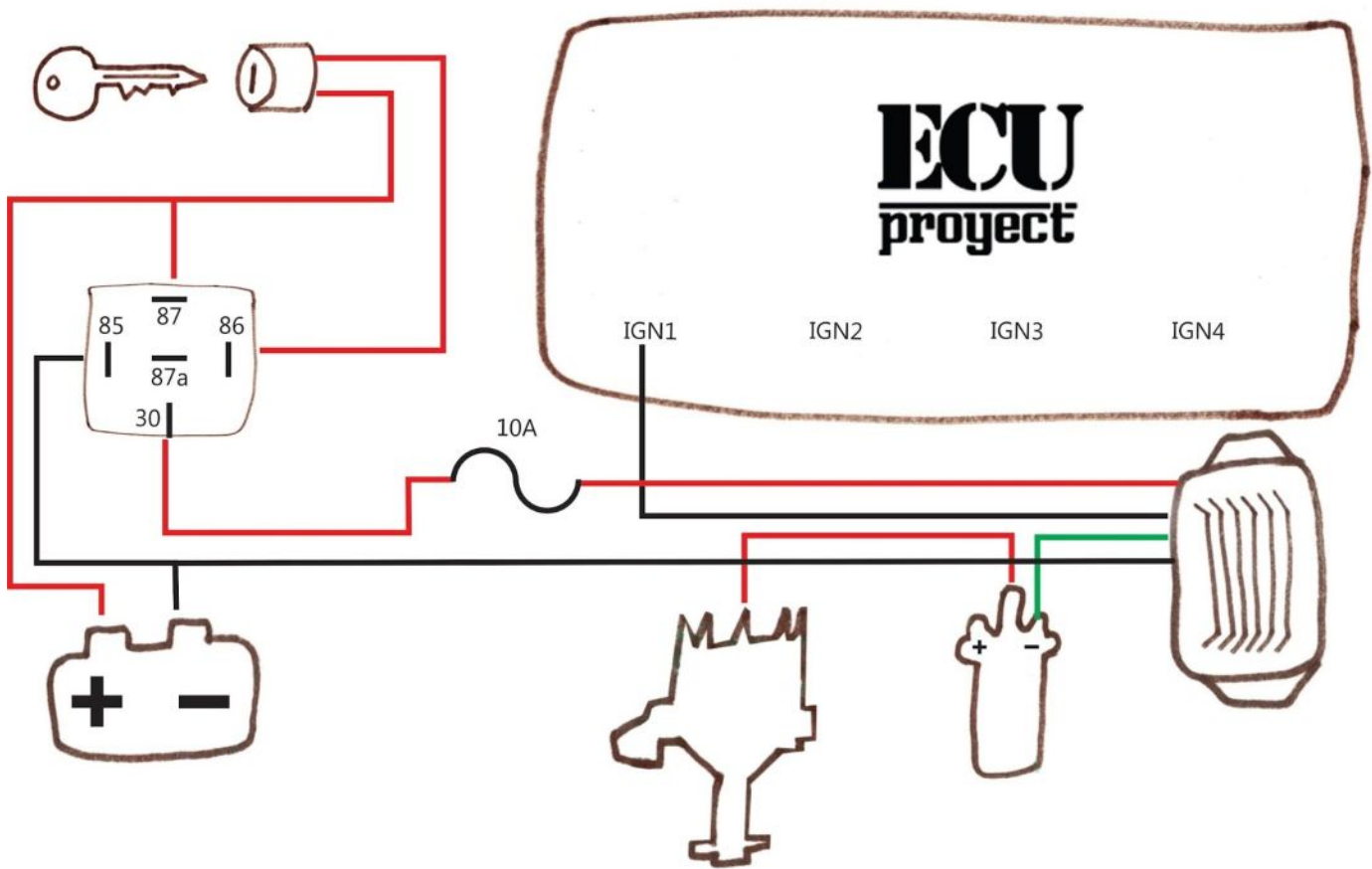


Firing order 1-3-4-2

Note: The example above uses 'smart' coils with integrated igniters. DO NOT connect dumb COPs (2-pin) without adding an ignition module or pickup.

### **Distributor**

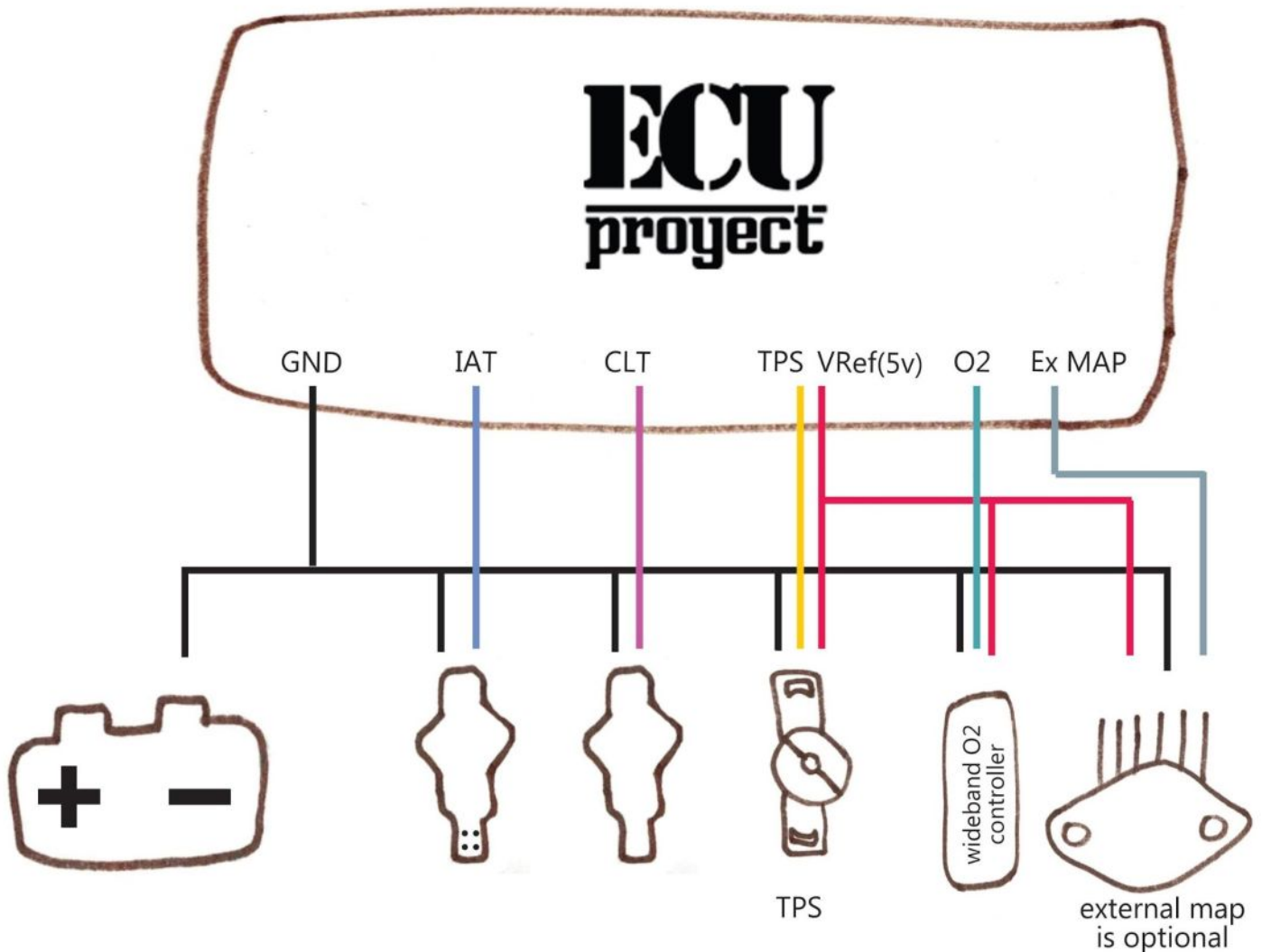
If a distributor remains in use, only a single output from the ECU is required. This needs to be fed to a single channel ignition module (such as the common Bosch 124) which can then drive the coil.



## Sensor

### Wiring Analog Sensor Wiring Analog

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



Notes: Use of 2-wire temperature sensors is recommended. While 1-wire sensors will work, they are almost always considerably less accurate.

The external MAP sensor in the diagram above is optional and can be omitted if integrated MAP is used. Alternatively, an external Baro sensor can be added in the same way as an external MAP.

A 3-wire variable TPS is required. On/off type TPS are not suitable.

## Settings

Index of settings 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

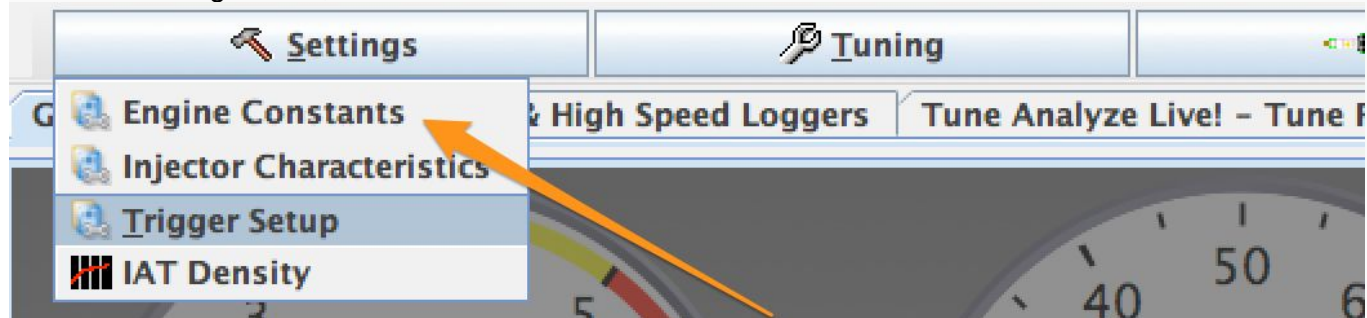
## Settings Menu

### Engine constants Engine

Constants

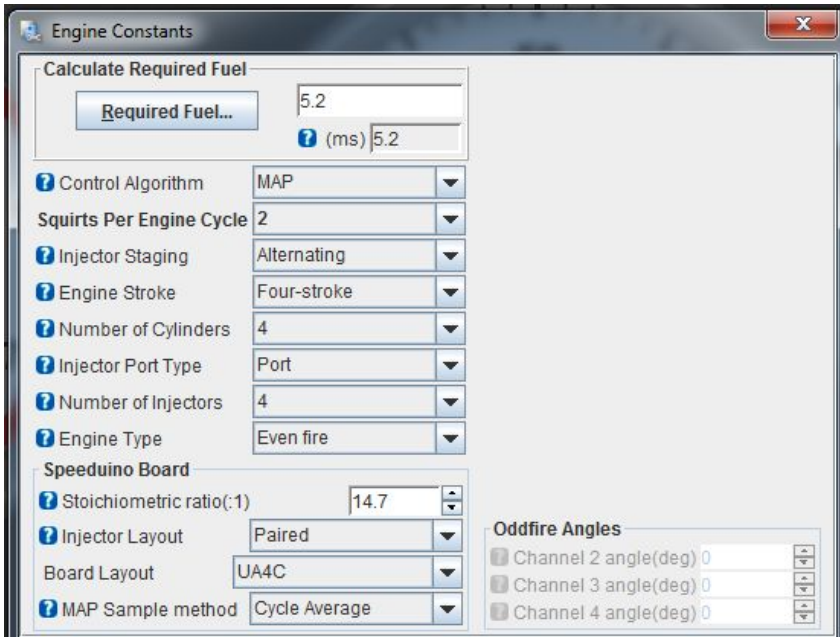
#### Overview

From the Settings menu, select Constants.



Here you need to set the motor constants. Complete the fields in the section below before calculating the fuel required.

### Setting Description



**Control Algorithm:** The load source to use for the fuel table.

**Squirts per EngineCycle** - How many squirts will be made over the duration of the engine cycle (eg 720 degrees for a 4 cycle engine). 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 ProjectECU will set the squirts to 1)

Note that for squirts 3 and 5, you should have a cam signal in addition to the crankshaft signal.

**InjectorStaging** – Sets the timing strategy used for the injectors

**Staggering** (recommended for most installations) – The injectors are timed around each cylinder TDC. The exact closing angle can be specified in the Injector Characteristics dialog.

**Simultaneous** - All injectors fire together, based on TDC of cylinder 1.

**EngineStroke** - Whether the engine is 2 or 4 stroke.

**Number of cylinders:** number of cylinders in the engine. For rotary engines, select 4.

**Injector Port Type:** The option is not used by the firmware. The selection currently does not matter.

**Number of injectors:** generally the same as the number of cylinders (for port injection).

**EngineType:** Whether the crank angle between firing is the same for all cylinders. If you are using an odd engine (eg some V-Twins and Buick V6's), the angle for each outlet channel must be specific.

**InjectorLayout** - how injectors are connected

**Paired:** 2 injectors are wires 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 the injector channels fire at the same time (1 and 4, 2 and 3), which means that the number of outlets used is equal to the number of cylinders. Only valid for 4 cylinders or less.

**Sequential:** 1 injector per outlet and the outlets used are equal to the number of cylinders. The injection is timed throughout the cycle. Only available for engines with 4 cylinders or less. REQUIRES SHAFT SENSOR IF SHAFT SENSOR IS NOT DETECTED DO NOT OPEN RPMs.

**BoardLayout** - Specifies the input/output pin layout based on the ProjectECU board you are using. For specific details of these assignments please refer to the quick guide for each ECU model (usually UA4C).

**MAP SampleMethod** - How the MAP sensor readings will be processed:

*Instantaneous* - Each reading is used as it is taken. Produces a highly fluctuating signal, but can be useful for *CycleAverage testing*: the average sensor reading over 720 crank degrees is used. This is the average of the event, they are the recommended options for 4 or more cylinders.

*CycleMinimum* - The lowest value detected in 720 degrees is used. This is the recommended method for less than 4 cylinders or ITB

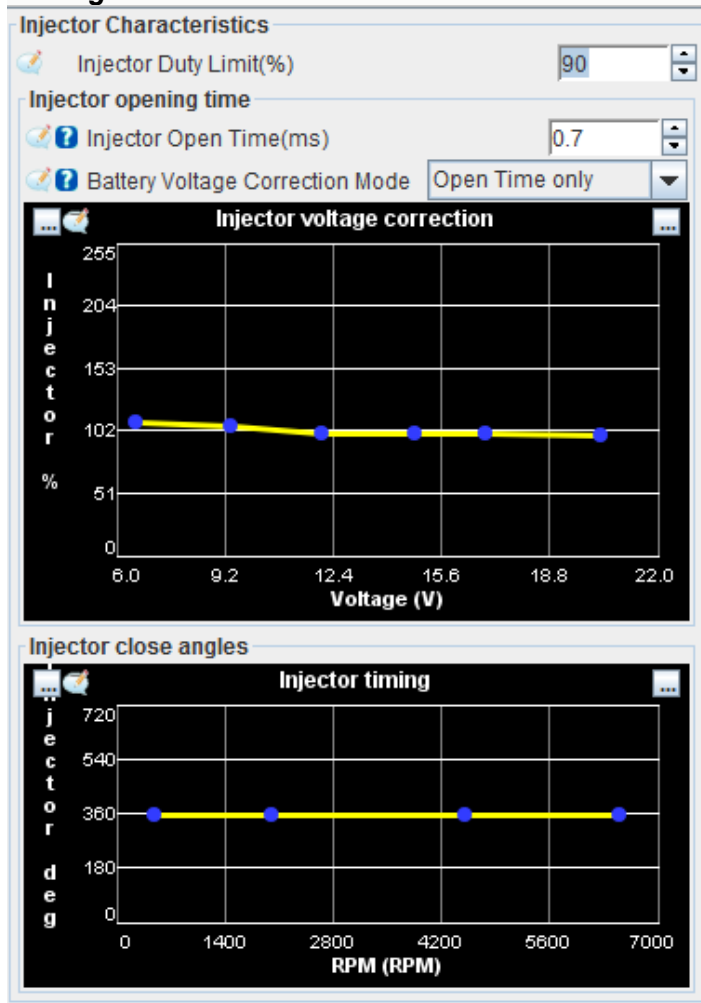
*EventAverage*: Similar to CycleAverage, however it averages once per firing event instead of once per cycle. It generally offers a faster response with a similar level of accuracy.

## InjectorCharacteristics

### Overview

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

### Settings



**InjectorDutyLimit**(Typically 85%): The injector opens and closes once per startup revolution, so taking into account the injector open time, the duty cycle is limited to prevent this from 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 quickly enough

to deliver more fuel. This can cause lean conditions at high RPM. If you reach this limit, seriously consider whether larger injectors are required.

**Injector Open Time**(Typically 0.7 - 1.5): The time it takes for the injector to fully open once it is activated, plus the time it takes to close. This is specific to each injector type and version. See Fuel Calculation for how this value is used, and the Adjustment section for adjustments to this value.

**Battery Voltage Correction Mode:** Normally Open time only is selected, this is to select if we want correction, only in opening or in opening and closing of injectors.

**InjectorVoltageCorrection**(Typically 100%): The percentage of injector pulse width varies with changes in supply voltage. A value of 100% means no change in pulse width.

**InjectorCloseAngles** (Typically 355): This represents the angle, relative to each cylinders TDC, that the injector jet will end. This can be varied per channel (including semi-sequential wiring), but the default value of 355 is suitable for most applications.

## TriggerSetup

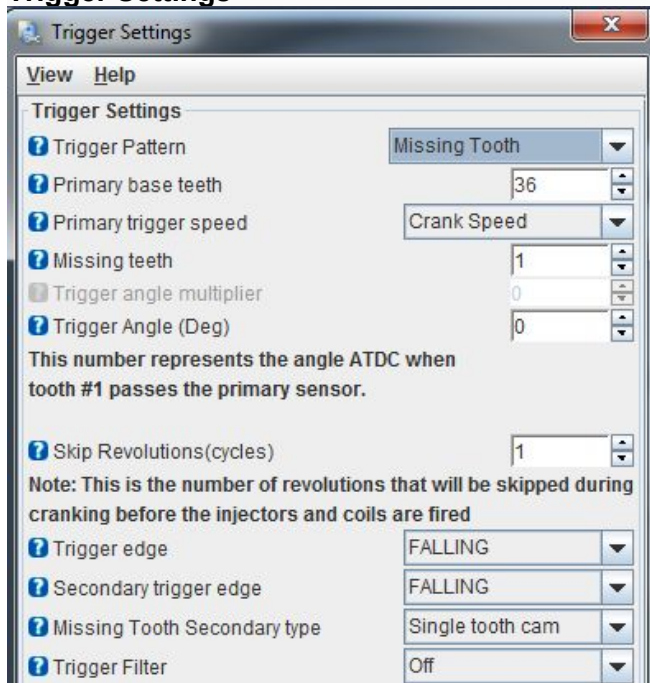
### Overview

One of the most critical components of an EFI setup is the Crank Angle Sensor (CKP or CAS) and how it is used by the ECU. The TriggerSetup dialog is where the trigger settings are defined and it is vitally important to have this correct before attempting to start the motor.

With an incorrect setting, you may have trouble synchronizing or seeing erratic RPM readings.

Note that many of the settings in this dialog depend on your settings, and therefore it is normal for some options to be grayed out.

### Trigger Settings



**TriggerPattern** - The pattern used by the crank/cam sensor settings on your engine. For a complete list of supported patterns, see the Decoders appendix.

**Primary Base teeth:** For patterns where the number of teeth is variable (missing tooth, double wheel, etc.), this number represents the number of teeth on the primary wheel. For wheels with missing teeth, this number should be the count, as if no teeth were missing.

**Primarytrigger speed** - The speed at which the primary trigger rotates. It is closely related to the primary base tooth configuration and indicates whether that number of teeth passes the sensor once every crank revolution or every cam revolution.

**Missingteeth** - If using the missing tooth pattern, this is the size of the gap, given in 'missing teeth'. For example, 36-1 has 1 missing tooth. 60-2 has 2 missing teeth, etc. The missing teeth **MUST** be located on a single block, there cannot be multiple spaces of missing teeth around the wheel.

**Secondaryteeth:** As in the previous case, but for the secondary entrance. This input is always assumed to run at cam speed.

**Triggeranglemultiplier-**

**Triggerangle:** The crank angle, after top dead center (ATDC), when tooth #1 passes the sensor on the primary (crankshaft) input. This setting is critical for ProjectECU to accurately know the current crankshaft angle. See the section below ('Finding Tooth #1 and Activation Angle') for more information on how to determine this value.

## Trigger options

**Skiprevolutions** - The number of revolutions the motor must make before the timing flag is set. This can help prevent false sync events on boot. Typical values are 0 to 2.

**Triggeredge** - Whether the primary signal is triggered on the rising or falling edge

**Secondarytrigger edge** - Whether the secondary signal is triggered on the rising or falling edge

**Triggerfilter** - A time-based software filter that will ignore crank/cam inputs if they arrive sooner than expected based on current RPM. The more aggressive the filter, the closer to the expected filter run time. However, higher levels of filtering can cause true pulses to be filtered out, so it is recommended to use the lowest setting possible

**Re-synceverycycle** - If set to yes, the system will look for sync conditions every cycle in instead of counting the expected number of teeth. It is recommended that this option be turned on, however if you have a noisy crank/cam signal you may need to turn it off as it can cause the timing to be off occasionally.

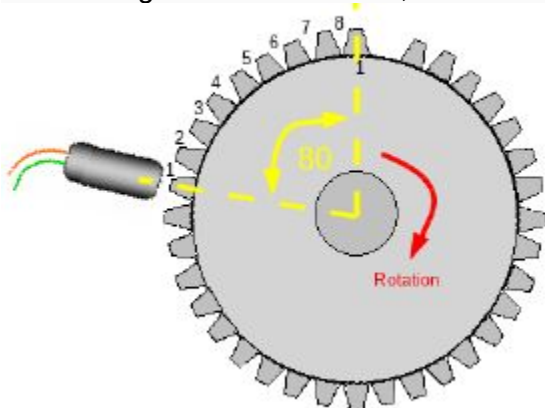
## Finding tooth #1 and firing angle

First of all, it is necessary to take into account that the ecu works with ATDC firing angle, which means that increasing the angle means advancing the ignition timing. And decreasing the angle means retarding ignition. It is possible that you come from a tuning environment where it is the opposite. Please take this into account.

Firing angle can be determined in 2 ways:

- Put the engine at TDC (TDC) and check the angle between the ckp sensor and the missing tooth.
- Carrying out an estimative search as explained in the video "Finding firing angle OLD SCHOOL style"

For the Engine at TDC method, the following figure can help:



A 36-1 wheel is shown, to obtain the angle per tooth only divide:  $360/\text{teeth}$ .

Angle per tooth:  $360/36 = 10$ . We have that the sensor is BEFORE by 8 teeth.

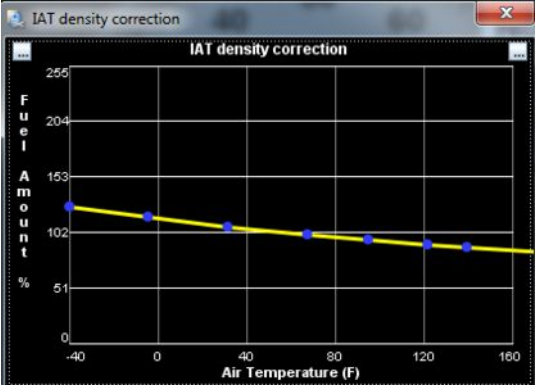
8 teeth means 80 degrees. Because the sensor is before the missing tooth we get BEFORE (negative angle -), AFTER (positive angle +).

In this case the angle will be -80 degrees.

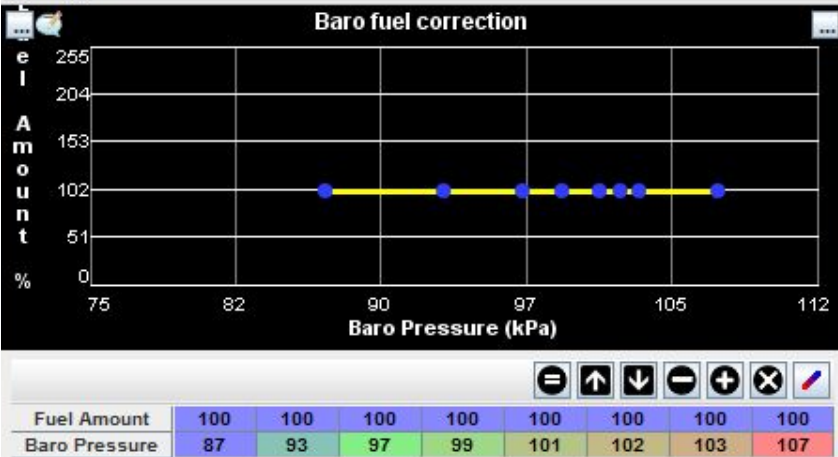
# IAT Density

## Overview

The IAT density curve represents the change in oxygen density of the inlet air as temperature increases. The default curve roughly follows the ideal gas law and is suitable for most installations, however if you experience very high inlet temperatures (either due to heat in the engine bay or from turbocharging) it is possible need to adjust the hot end of this curve.



# Baro Correction



## Overview

The barometric density curve represents the change in barometric pressure per change in height. Normally a barometric sensor is not used as the ECU "self-calibrates" each time it is turned on. But in the event that you need to compete in a "Hill Climb" type race or with sudden changes in altitude from the start to the end of the race, installing a barometric sensor will be a great help to maintain the tune during altitude changes.

# TuningMenu

## AccelerationEnrichment Acceleration

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

To function properly you must have a variable TPS installed and calibrated. OR Have a MAP installed and calibrated.

## Theory

Throttle enrichment adjustment is based on the rate of change of throttle position, a variable known as TPSdot (TPS delta over time). This is measured in %/second, with higher values representing faster throttle pressures and values in the range of 50%/s to 1000%/s being normal. For example:

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.

## Parameters

**EnrichmentMode-** You can choose between using TPS or MAP as a source to measure the sudden change of throttle.

**TpsDOTthreshold/MAPdotthreshold-**Parameter that allows us to filter small changes and filter them to avoid shooting fuel where it is not required.

**Accel Time-** Injection time after the AE routine was triggered, this allows tuning to avoid areas with excess or lack of fuel.

**TaperStart RPM-** Minimum RPMs at which the AE routine will be taken into account.

**TaperEnd RPM-**Maximum RPMs at which the AE routine will be taken into account.

**AE coldadjustment-**Scales the throttle enrichment percentage linearly based on coolant temperature. At the start temperature, the setting will be equal to the cool setting field (%). At the final temperature, the setting will be 0%.

## Tuning

Acceleration Enrichment

View Help

Acceleration Enrichment

TPS based AE

MAP based AE

Added %

TPSdot (%/s)

MAPdot (kPa/s)

Acceleration Enrichment

Enrichment mode: TPS

TPSdot Threshold(%/s): 70

MAPdot Threshold(kPa/s): 70

Accel Time(ms): 200

Taper Start RPM: 1000

Taper End RPM: 5000

Acceleration Enrichment cold adjustment

Cold adjustment(%): 100

Cold adjustment taper start temperature(F): 32

Cold adjustment taper end temperature(F): 140

Deceleration Fuel Cutoff (DFCO)

Enabled: Off

TPS Threshold(%): 1

Minimum engine temperature(F): 122

Cutoff delay(S): 1.0

Cutoff RPM: 1500

RPM Hysteresis(RPM): 200

TPSdot bins used for Accel Enrichment

Left Arrow Right Arrow Burn Close

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

In most cases, AE curve adjustment can be performed in a stationary environment, although road or dynamometer adjustment is also possible. Quick and slow throttle taps should be made and the effect on AFRs monitored using the live line graph in the AE dialog. This graph shows the TPSdot and AFR values in sync with each other, making settings in the correct part of the AE curve easier to identify.

If you find that the AFR is initially good, but then dips briefly, you should increase the Accel Time setting, with recommended increments of 10-20ms.

### **False Trigger**

In cases where the TPS signal is noisy, spikes in its reading can incorrectly trigger throttle enrichment. This can be seen in a log file or live script in TunerStudio by activating the 'TPS Accel' flag when there is little or no throttle movement.

If this occurs (and assuming the TPS wiring cannot be corrected to reduce noise), false triggers can be prevented from firing AE by increasing the "TPSdot Threshold" value. This should increase in ~5%/s increments, pausing between each increase to observe if AE is still triggering incorrectly.

### **Deceleration Fuel Cutoff (DEFCO) Deceleration fuel cutoff**

This is a function focused on fuel saving, its main function is to completely cut off the injection when the ProjectECU detects that it is in deceleration mode.

**TPS threshold**-The TPS value below which DFCO will be active. Typical value is 5%-10%, but a higher value may be required if the TPS signal is noisy.

**Minimum engine temperature**- Minimum temperature value to start the DEFCO operation, with this unnecessary cuts are avoided with the engine cold.

**Cutoff**The RPM above which DFCO will be active. Usually set a few hundred RPM above full idle speed.

**RPM Hysteresis**-Hysteresis for DFCO RPM. 200-300 RPM is typical for this, however a higher value may be necessary if RPM fluctuates around the cutting speed.

### **AFR/O2**

AFR / O2 (paire: fuel ratio) dialog box controls closedLoop fuel control, used to adjust 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 to the current target fuel ratio and adjust accordingly.

The use of a wideband sensor and controller is strongly recommended, however basic functionality is possible with a narrowband sensor if one is not available.

Please note that closed loop fuel control is not a replacement for poor tuning. Many good setups do not use closed-loop control or only allow you very little adjustment authority.

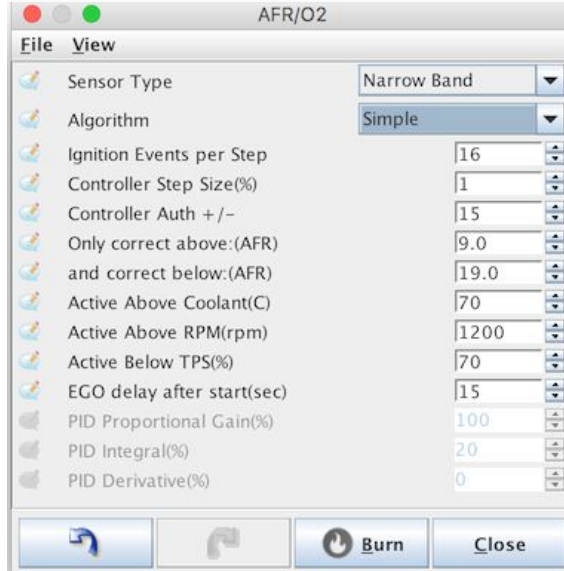
### **Configurations**

ProjectECU supports 2 closed loop algorithms, each targeting different configurations:

**Simple** - a time based 'goal chasing' algorithm where the amount of fuel trim depends on how long the reading has been lean or rich compared to the current target. This algorithm is best suited for narrow band sensors where only basic rich/lean information is available. In particular, this algorithm performs poorly if you have a fuel map that is nowhere near complete. If you have this enabled and you see oscillations in pulse width and/or AFR, even when cruising, then you should disable closed loop control until the base fuel map is better tuned.

**PID** – This is the preferred closed loop algorithm and will provide the best results when combined with a wideband sensor and tuned correctly.

### Common variables



The screenshot shows a software window titled "AFR/O2" with a menu bar containing "File" and "View". Below the menu bar is a list of configuration options, each with a small icon to its left. The options and their current values are:

Option	Value
Sensor Type	Narrow Band
Algorithm	Simple
Ignition Events per Step	16
Controller Step Size(%)	1
Controller Auth + / -	15
Only correct above:(AFR)	9.0
and correct below:(AFR)	19.0
Active Above Coolant(C)	70
Active Above RPM(rpm)	1200
Active Below TPS(%)	70
EGO delay after start(sec)	15
PID Proportional Gain(%)	100
PID Integral(%)	20
PID Derivative(%)	0

At the bottom of the window are four buttons: a left arrow, a right arrow, a "Burn" button with a flame icon, and a "Close" button.

**Sensor type:** narrowband or wideband, depending on the hardware configuration. Narrowband sensor must be 0-1v type, wideband sensors must have 0-5v signal. Broadband sensors must be calibrated in the Tools->Calibrate Table AFR

**Algorithm dialog** - see description above for each available algorithm

**Initiation events per step** - AFR setting calculation will be performed every few ignition cycles. Changes to the closed loop setting generally have some delay before the O2 sensor registers its impact and by increasing this value you can account for this delay. Typical values are 2-5.

**Controller step size-**

**Controller Auth:** The maximum percentage that the pulse width can be changed through this closed-loop setting. The recommended value is not more than 20%.

**Correct above/below AFR:** The AFR range in which the closed loop adjustments will be applied. This range is generally limited by the sensor and controller in use.

**Activate above Coolant** - The closed loop should only run once the engine reaches operating temperature. This value should be set to match the standard operating temperature of the motor.

**Activate above RPM:** Closed loop adjustments should generally not be made at idle. Use this value to specify when the adjustment should begin.

**Activate below TPS** – Above this TPS value, closed loop settings will be disabled

**EGO delay after start** – All O2 sensors require a warm-up period before their readings are valid. This varies depending on the sensor in use, but 15s is a safe value in most cases.

### Variable for PID mode



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

## Overview

Configure engine protection limiters

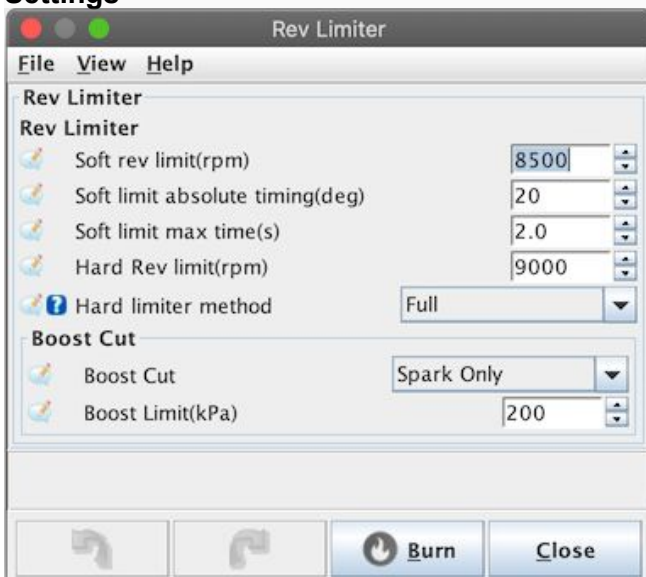
### ProjectECU

includes an ignition cut-based rev limiter with hard and soft cuts.

The soft cut limiter will lock the time to an absolute value to further reduce throttling. If the RPM continues to rise and reaches the hard cutoff limit, the ignition events will cease until the RPM falls below this threshold.

Note As this is a spark based limitation, Fuel Only installations cannot use the rev limiter functionality.

## Settings



Soft RPM Limit – The RPM at which the soft cut ignition timing will be applied.

Absolute Soft Limit Time: As long as the engine is above the soft limit RPM, the ignition advance will remain at this value. Lower values here will have a more smooth cutting effect.

Soft Limit Max Time – The maximum number of seconds the soft limiter will work for. If the motor remains in the soft cut RPM region for longer than that, hard cut will be applied.

Hard Rev Limiter - Above this RPM, all ignition events will cease.

## Flex Fuel

Flexible fuel / ethanol content sensor configuration.

### Overview

ProjectECU 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 fuel. A flex fuel sensor is installed in the fuel feed or return lines and a signal wire is used as an input on the ProjectECU board.

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

### hardware

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

Small - #13577429

Medium - #13577379

Wide - #13577394 (same as the medium size, 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 need to cut 2 tabs on the sensor side.

Connector Part Numbers:

Housing (#13519047)

Pins (#15326427)

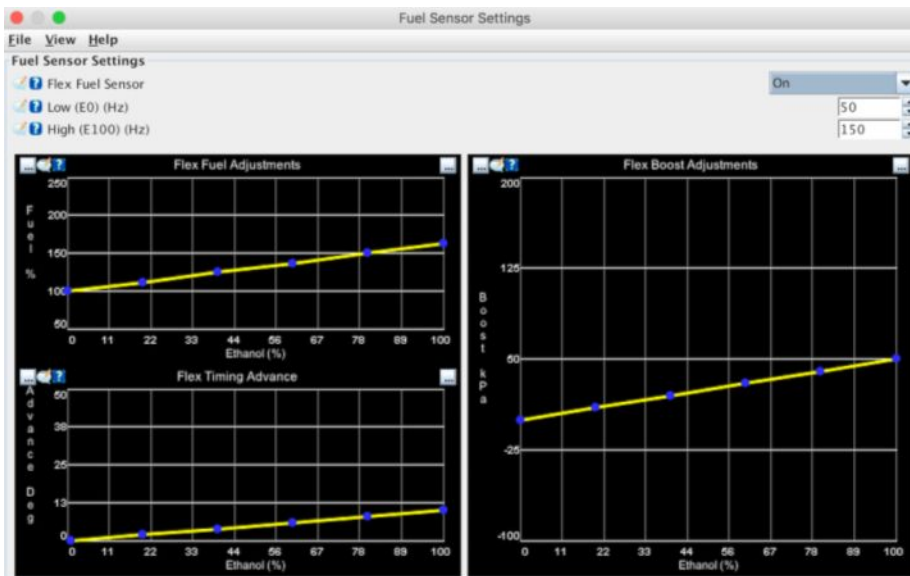
Seal (#15366021)

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

### Connection

All units are wired identically and have markings on the housing indicating what each pin is for (12v, ground and signal), ProjectECUs are provided with a cable for direct sensor signal connection.

### tuning



**Sensor frequency** The minimum and maximum frequency of the sensor representing 0% and 100% ethanol, respectively. For standard GM/Continental flex sensors, these values are 50 and 150

**Fuel multiplier%** - This is the additional fuel that must be added as the ethanol content increases. The low value on the left represents the fuel map tune with 0% ethanol and will typically be 100% if the base tune was done on E0 fuel. However, if the base tune was made with E10 or E15, this value can be set 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. This value may need to be adjusted

**Additional advance:** The additional degrees of advance to be applied as the ethanol content increases. This amount increases linearly between the low and high values and is added after all other power-up modifiers have been applied.

## Staged injection

ProjectECU has the ability to control a secondary fuel stage for engines that have 2 sets of injectors, usually of different capacities. There are few stock engines that come with secondary injectors, the notable exception being many Mazda rotaries. Secondary staged injection is also a common modification, used wherever large injectors are required, but where it is desirable to keep smaller injectors for smoother low RPM performance.

### Configuration

No matter what control strategy is chosen, you must enter the size of the primary and secondary injectors to let ProjectECU know the split in the overall fueling.

**CRITICAL:** The value of req\_fuel in the engine constants **MUST** be updated when staged injection is activated. When using Staged injection mode, the value entered into the req\_fuel calculator **MUST** be equal to the sum of the primary and secondary injector sizes.

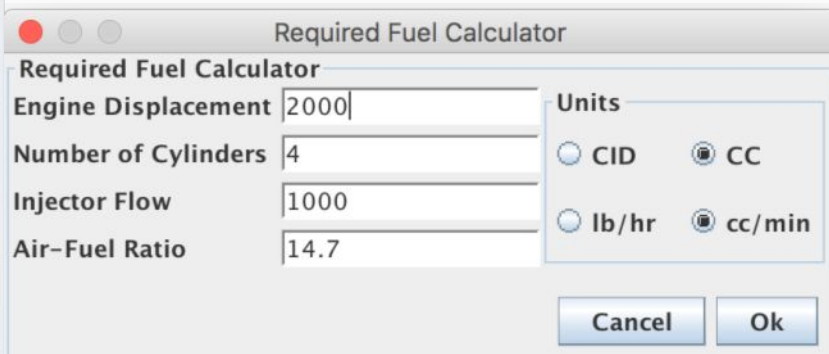
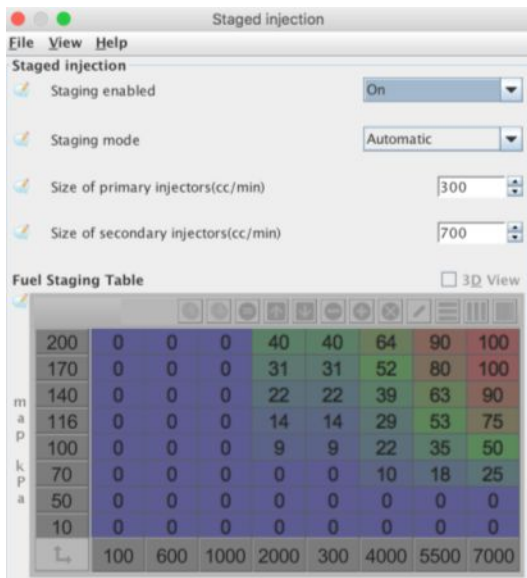
Failure to set these values correctly will result in excessive fuel rich or lean conditions.

For example:

Primary injectors: 300cc

Secondary injectors: 700cc

Value entered in the req\_fuel calculator: 1000cc



## Control methods

ProjectECU offers 2 preparation control modes, each with its own strengths and weaknesses. In most cases, it is recommended to start with Automatic mode, which only requires tuning the standard VE table, and check to see if you get the desired result. Only if this cannot be adjusted to provide satisfactory fuel splitting would a switch to manual table adjustment be recommended.

## Table control

The table control allows the use of an 8x8 manual map that indicates what percentage of the fuel load will be carried out by the secondary injectors

0% = secondary injectors disabled

100% = primary injectors disabled

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

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

## Automaticstaging

ProjectECU has an automatic staging method that takes into account the total capacity of the injectors (ie the sum of the 2 stages of the injector) and will perform a division of these by itself. With this method, the user can simply adjust the VE table in the same way as using a single set of injectors and the system will take care of the rest.

In this mode, ProjectECU will attempt to use the primary injectors up to their 'Injector Service Limit' (as configured in the Injector Characteristics dialog). When staging is used, it is recommended that this limit not be higher than 85 %.Once If the primary injectors reach this duty limit, ProjectECU will start to do more fueling of the secondary injectors.This way the VE table is all that is required for tuning as the system will take care of it. of assigning the current fuel load to the best injectors

Note that the dead time of the 2 sets of injectors is currently assumed to be the same

# SparkMenu

## SparkSettings Configuration

options for the ignition outputs

### Overview

The SparkSettings dialog Spark config contains the options for how the ignition outputs will work, including which of the IGN outputs is used and how. These are or values Incorrect parameters and parameters will cause the motor not to start and in some cases may lead to damaged coils. This dialog also contains a number of options to correct the power-up timing for testing and diagnostics.

Make sure you have checked this setting before attempting to start your engine.

### Settings

Spark Settings

Ignition load source: MAP

Spark output mode: Wasted Spark

Cranking advance Angle(Deg): 5

Spark Outputs triggers: Going Low

Locked timing

Enabled Fixed/Locked timing: Off

Fixed Angle(Deg): 0

Note: During cranking the fixed/locked timing angle is overridden by the Cranking advance angle value above

New Ignition Mode

This option is currently will improve accuracy on most compatible triggers  
However if timing issues are encountered, please disable this

Use new ignition mode: Yes

**Spark Output mode** - Determines how spark pulses will be output and is very specific to spark wiring. Note that no matter which option is selected here, the power-up signals are ALWAYS fired in numerical order (ie 1->2->3->4) up to the maximum number of outputs. The firing order of the engine is taken into account in the wiring order.

**WastedSpark:** The number of ignition outlets is equal to half the number of cylinders and each outlet will fire once every revolution of the crank. Therefore, one spark will occur on the compression stroke and the other on the exhaust stroke (also known as the 'wasted' spark). This method is common on many 80's and 90's vehicles that come with specific wasted spark coils, but can also be used with individual coils that are connected in pairs. The wasted spark will work only with a crank angle reference (eg a missing crank wheel with no cam signal)

**Single Channel** - This mode sends all ignition pulses to the IGN1 output and is used when the engine contains a distributor (usually with a single coil). The number of output pulses per revolution (crank) is equal to half the number of cylinders.

**Wasted COP** - This is a handy mode that uses the same amount of time as the 'Wasted Spark' option; however, each pulse is sent to 2 turn-on outputs instead of one. These are IGN1/IGN3 and IGN2/IGN4 paired (ie when IGN1 is high, IGN3 will also be high). Since this is still a wasted spark timing mode, only the crank position is required and there will be 1 pulse per torque, per crank revolution. This mode can be useful in cases where there are 4 individual coils, but it is not desirable or possible to execute the entire sequence (for example, when no tree reference is available).

**Sequential** - This mode only works on engines with 4 cylinders or less.

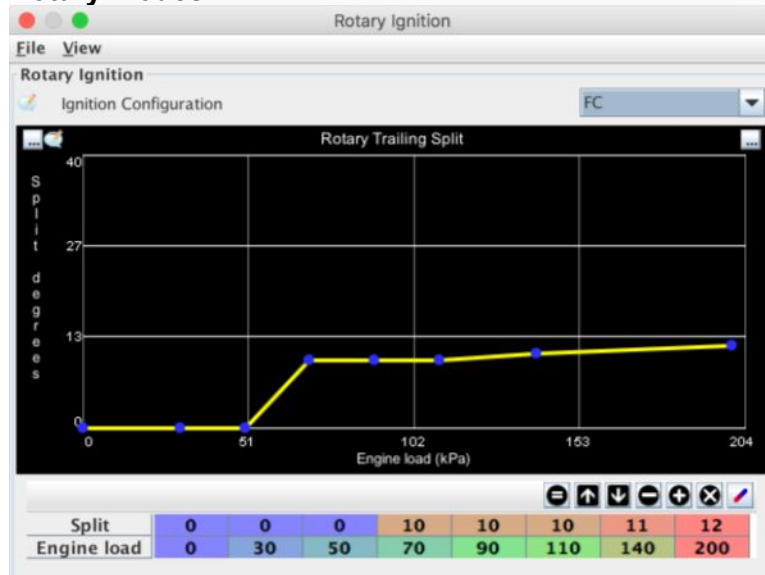
**Rotary** - See below for more details

**Start Advance** - The number of absolute degrees (BTDC) that the time will be set to when starting. This overrides all other timing advance modifiers during startup.

**Spark output triggers:** THIS IS A CRITICAL SETUP! Selecting the wrong option here may cause damage to your ignitors or coils. Specifies whether the coil will fire when the ProjectECU ignition output is HIGH or LOW. The vast majority of ignition configurations will require this to be set LOW (ie the coil charges/stays when the signal is high and will fire when that signal is low). While LOW LOW is required for most ignition settings, there are some settings that do the dwell time in the ignition module and fire the coil only when they receive a HIGH signal from the ECU.

**FixedAngle** - Used to lock the ignition timing to a specific spark advance for the test. Setting this to any value other than 0 will result in that exact angle being used (i.e. overrides any other advance settings) at all RPM/load points except during startup (Startup always uses the advance setting). Previous boot advance). This setting must be set to 0 for normal operation.

## Rotary modes



ProjectECU currently only supports the ignition configuration used on FC RX7 engines. Support for FD and RX8 ignition configurations is not available. Front/rear split angle can be set based on current engine load.

FC: The outputs are set to the start/end settings that were used in FC RX7. The wiring is:

- IGN1 - Leading (wasted) sparks
- IGN2 - Trailing spark
- IGN3 - Trailing select
- IGN4 - Not used

FD - Uses the same wasted spark signal for the two initial sparks as FC, but individual signals for the final 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. The wiring is:

- IGN1 - Front rotor leading
- IGN2 - Rear rotor leading
- IGN3 - Front rotor trailing
- IGN4 - Rear rotor trailing

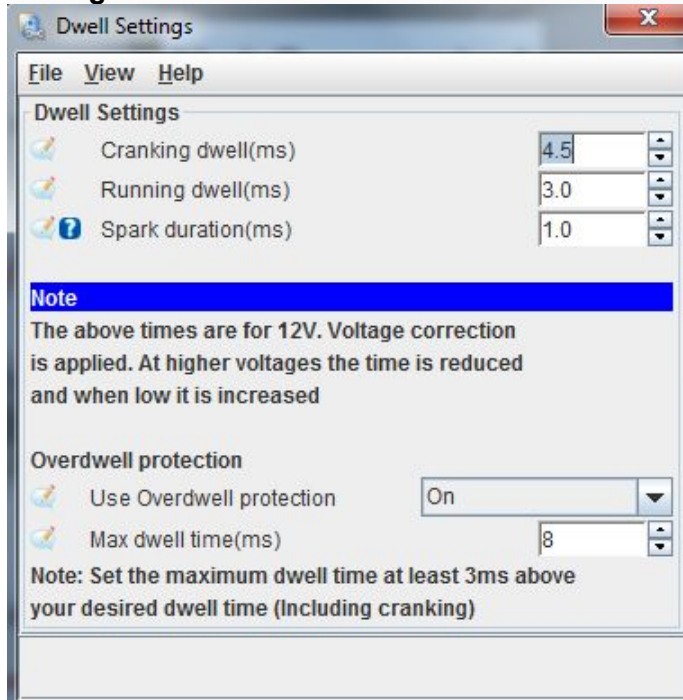
## Dwell

## Overview

The Dwell control dialog alters the coil charge time for the ProjectECU ignition outputs. Care must be taken with these settings, as modules and coils can be permanently damaged if left on for excessive periods of time.

The Dwell will automatically decrease when the configured duration is greater than the time available at the current RPM. This is common in single channel ignition configurations (eg 1 coil with a distributor) and particularly in larger cylinder count engines.

## Settings



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

**Crankingdwell** - The nominal dwell time to be used during cranking. Running is defined as when the RPM is above 0, but below the 'Start RPM' values in the startup dialog

**Running dwell:** Nominal dwell time to use when the motor is running normally.

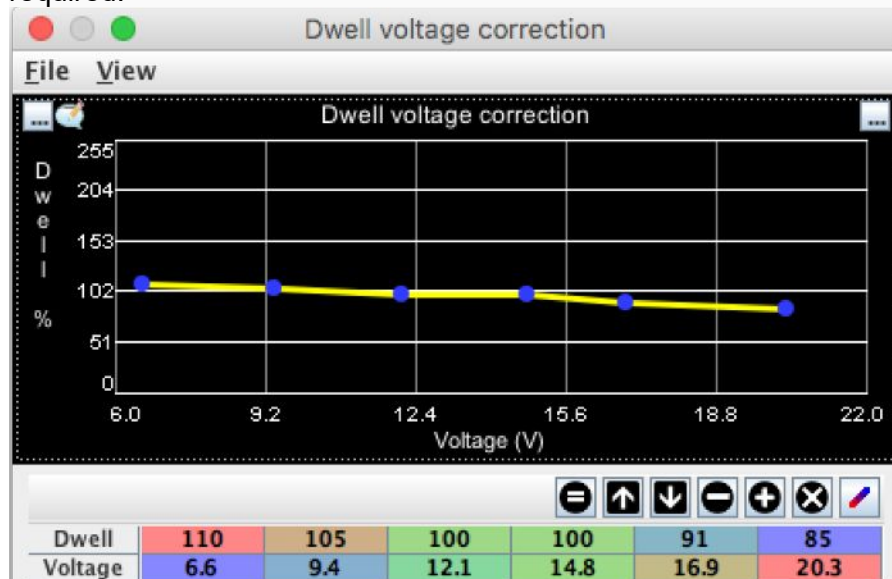
**Sparkduration** – The approximate time it takes for the coil to fully discharge. This time is used to calculate a reduced dwell time under time limited conditions, as mentioned above on high cylinder count single coil motors. Limited dwell time is calculated by taking the maximum revolution time at 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.

**Overdwellprotection:** The overdwell protection system works independently of the standard power-on times and monitors how long each power-on output has been active. If the active time exceeds this amount, the output will end to prevent damage to the coils. This value should typically be at least 3ms higher than the previously configured nominal dwell times to allow for nominal charging and voltage correction.

## VoltageCorrection

As the system voltage increases and decreases, the Dwell time needs to decrease and increase respectively. This allows for a constant spark force without damaging the coil/s during high system voltage conditions. It is recommended that 12v be used as the 'nominal' voltage, which means the % hold at 12v figure should be 100%.

The correction curve in the base Tune file should be suitable for most coils/ignitors, but can be modified if required.

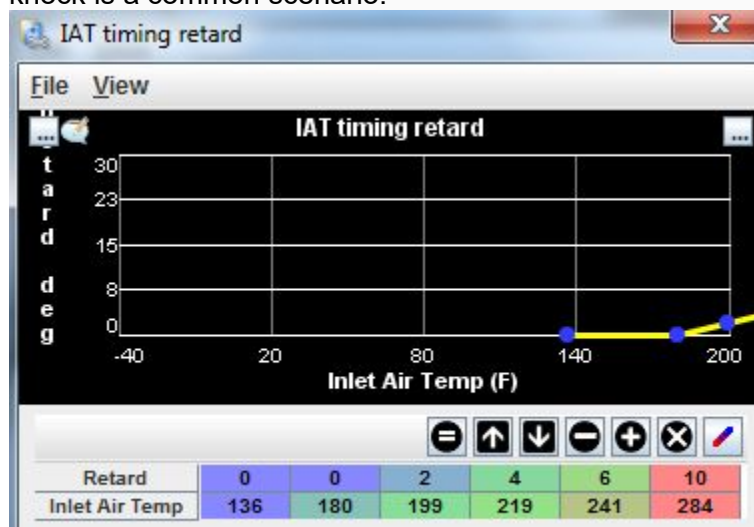


## IAT retard

Changes in intake air temperature (IAT), particularly significant increases during boost, may require ignition timing to be retarded. The IAT delay setting allows for this timing adjustment.

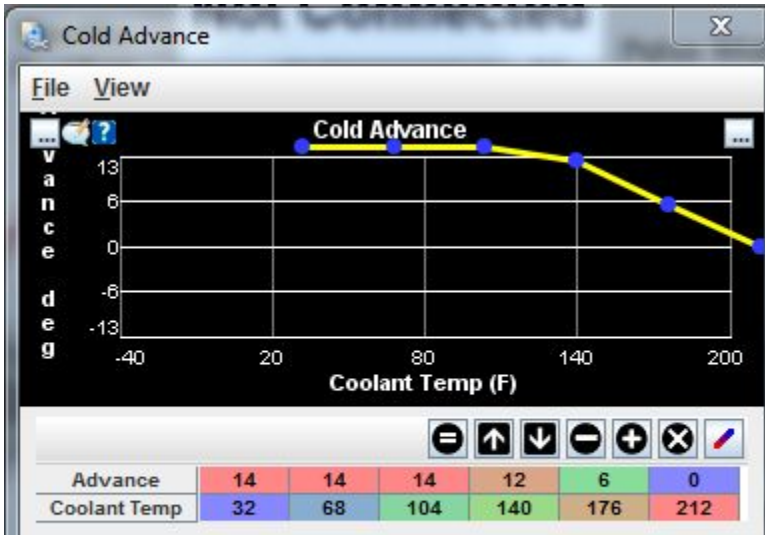
### Example

The exact settings will depend on the engine, but delaying the ignition timing beyond 100°C to eliminate pre-knock is a common scenario.



## Cold Advance

Its function is to add a little time advance while the engine is still cold, this to start the cold slow combustion, and avoid having low engine response when cold.



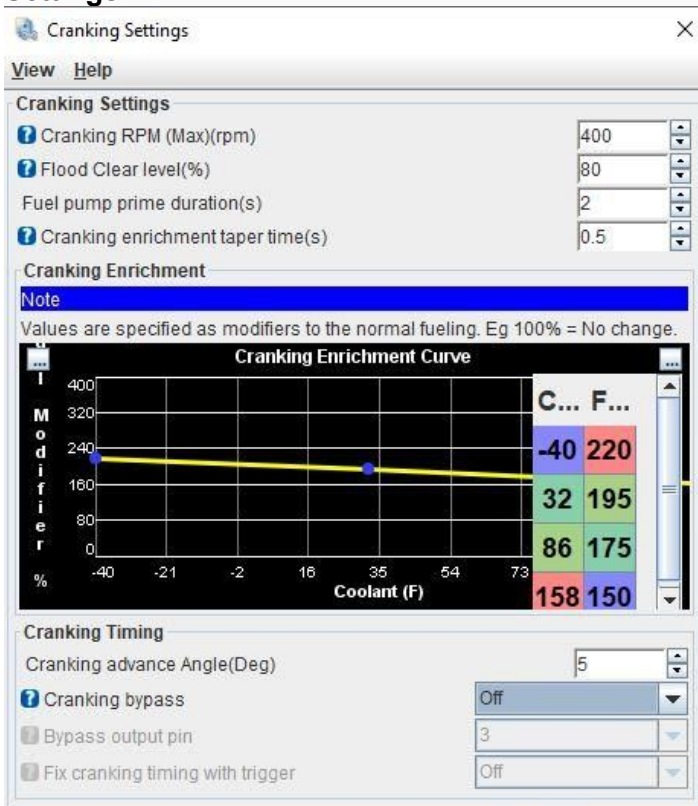
# Starting/Idle Menu

## Starting

### Overview

conditions typically require multiple adjustments to both fuel and ignition control to provide smooth and quick starts. The settings in this dialog determine when ProjectECU will consider the engine to be in a run/start condition and what settings should be applied during this time.

### Settings



**Cranking RPM** - This sets the threshold for whether ProjectECU will run in cranking or running mode. Any RPM above 0 and below this value will be considered boot and all boot related settings will be applied. In

general, it's best to set it to about 100 rpm higher than your typical cranking speed to account for spikes and provide a smoother transition to normal idle

**Flood Clear level - Flood** clearing is used to help eliminate excess fuel that has entered the cylinder(s). While Floodclear is active, all fuel and ignition events will stop and the engine can be cranked for a few seconds without risk of additional starts or flooding. To activate Floodclear, the RPM must be below the previous Cranking RPM setting and the TPS must be above the threshold of this setting.

**Fuel pump prime duration** – When ProjectECU is powered up for the first time, the fuel pump output will turn on for these seconds to pressurize the fuel system. If the engine starts at this point, the pump will simply keep running; otherwise it will turn off after this period of time. Note that fuel pump priming only occurs at system start-up.

**PrimingPulsewidth** – Upon power up, ProjectECU will fire all injectors for this period of time. This pulse is NOT designed as an initial fuel charge, but is to clean out any air that may have entered the fuel lines. It should be kept short to prevent engine flooding.

**Crankingenrichmenttaper time:** transition time between cranking mode and engine on mode, this is to avoid sudden changes between routines.

**Crankingenrichment** - While the crank is active (see Crank RPM above), the fuel load will increase by this amount. Note that as a standard correction value, this bootstrap enrichment is in addition to any other settings that are currently active. This includes warm-up enrichment, etc.

**Cranking Bypass** - This option is specifically for ignition systems that have a hardware ignition option. These systems were used during the 1980's and early 1990's and allowed the ignition timing to be set and controlled by the ignition system itself when active (via an input wire). With this option, you can specify an output pin that will be set HIGH when the system is booting. The specified pin number is the Processor pin number.

**Fixcrankingtimingwithtrigger:** Some (usually low resolution) trigger patterns are designed to align one of their pulses with the desired cranking timing. This is typically 5 or 10 degrees BTDC. When enabled, ProjectECU will wait for this timed input pulse before triggering the corresponding ignition output (a safety factor of permanence is still applied in case this pulse is not detected). This option is only available when a trigger pattern that supports this feature is selected (see Trigger Settings)

## Warmup

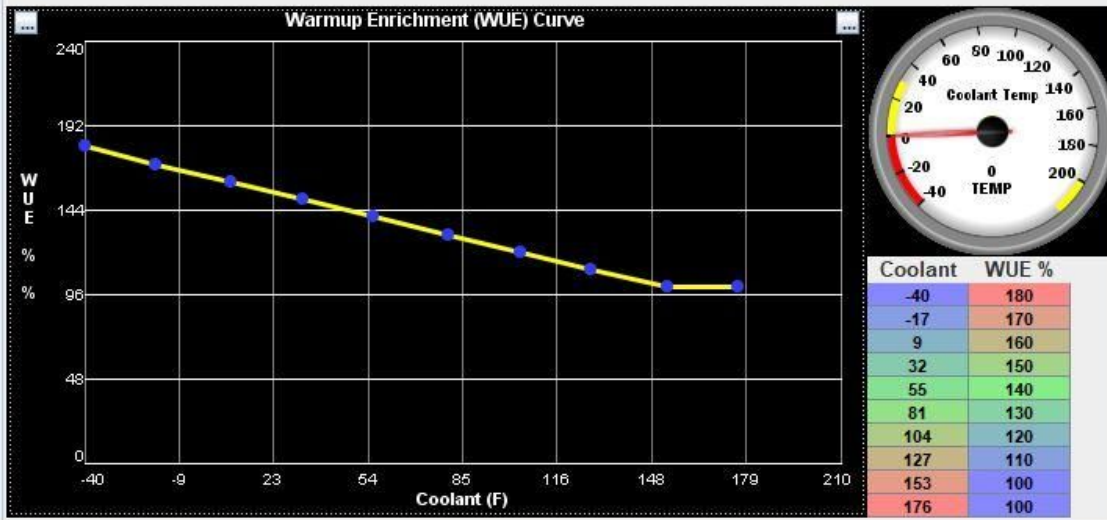
### Overview

The Warmup Enrichment (WUE) dialog contains settings related to the post-start period (i.e., after starting) but before the engine has reached normal operating temperature. Allow fueling modifications during this time for smooth cold driving.

### Settings

#### Warmup Enrichment (WUE) - Percent Multiplier

Final enrichment value must be 100%.



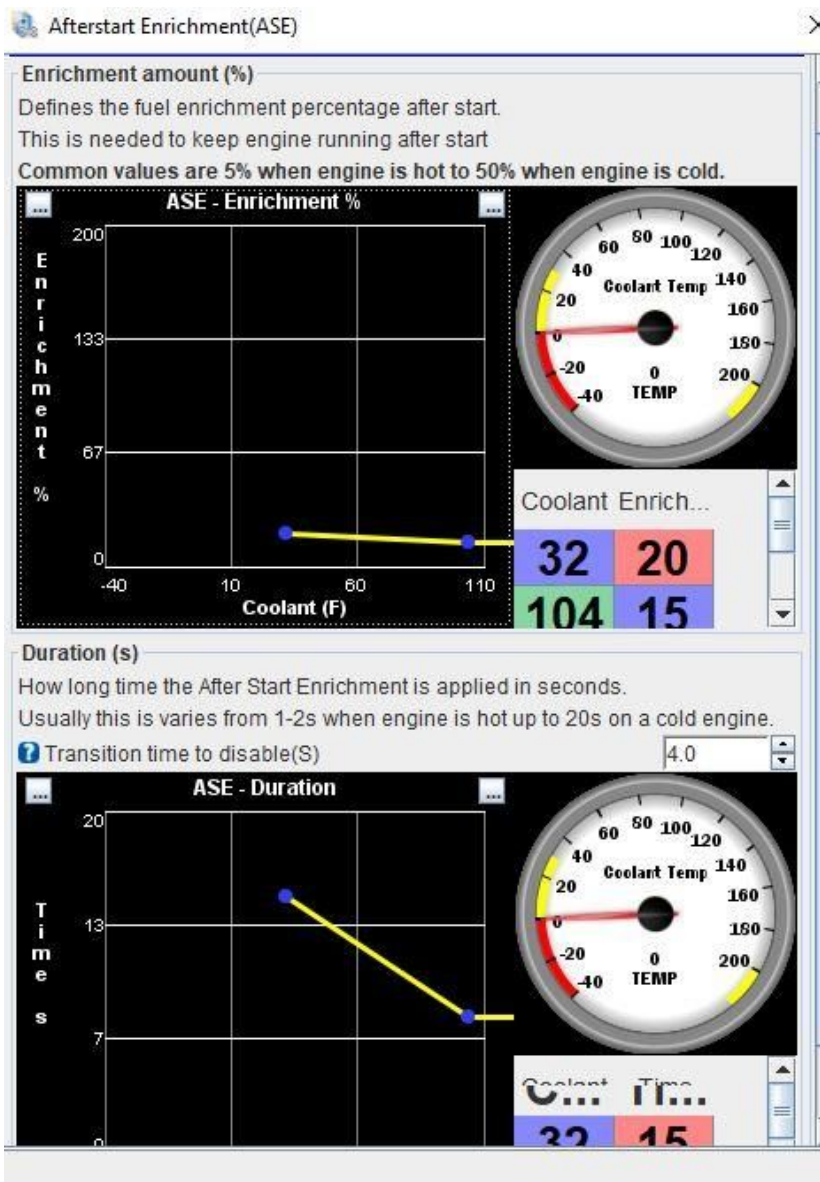
#### Warmup curve

This curve represents the amount of additional fuel that will be added while the engine is coming up to temperature (according to the coolant sensor). The final value on this curve should represent the normal operating temperature of the engine and have a value of 100% (representing no change in fuel beyond that point).

#### Afterstart Enrichment Afterstart

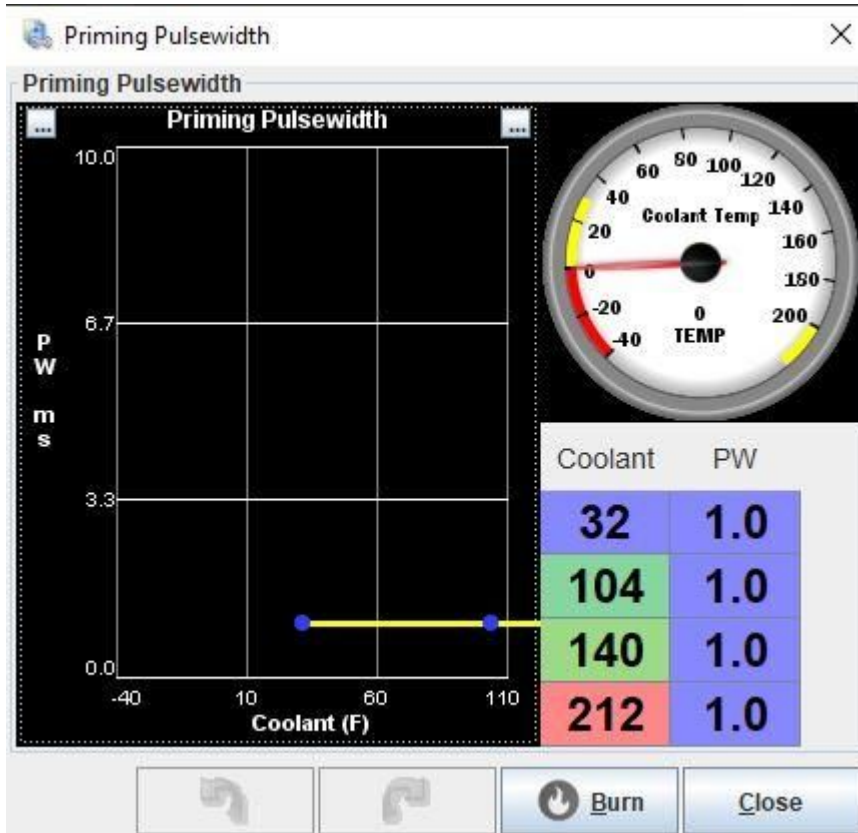
Enrichment (ASE) is a separate fuel modifier that operates above the WUE for a fixed period of time after the engine is first started. Typically, this is a 3-10 second period where a little enrichment can help the engine make a smooth transition from starting to idle.

There is a table by temperature to define extra amount of fuel % and duration of ASE, which allows precise adjustments.



## PrimingPulsewidth

The priming pulse is used to remove air from the injectors prior to startup. The table allows adjustment according to the motor temperature.



## AccessoriesMenu

### Thermo Fan

Control of a cooling fan relay.

#### Overview

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

Please note that currently only fan on/off control is possible (not fan speed control using PWM).

### Settings

The Fan Settings dialog box contains the following settings:

- Fan Mode:** On/Off
- Allow fan when off:** No
- Allow fan when cranking:** No
- Fan output pin:** Board Default
- Fan Output Inverted:** No
- Fan switching temperature(F):** 187
- Fan hysteresis(F):** 2

**Fan mode:** currently only on/off mode is supported

**Allow fan when off:** defines if it will be possible for the fan to turn on with the motor off

**Allow fan whencracking:** define if it will be possible for the fan to be on during motor start

**Fan output pin:** The Processor pin that will be used for fan control. In most cases, this should be left as Board Default

**Fan output inverted** - Most configurations will use No for this configuration, but if you have a fan circuit that inverts the output, the on/off polarity will be set to No. you can invert with this setting

**Fan temperature:** the temperature above which the fan will turn on.

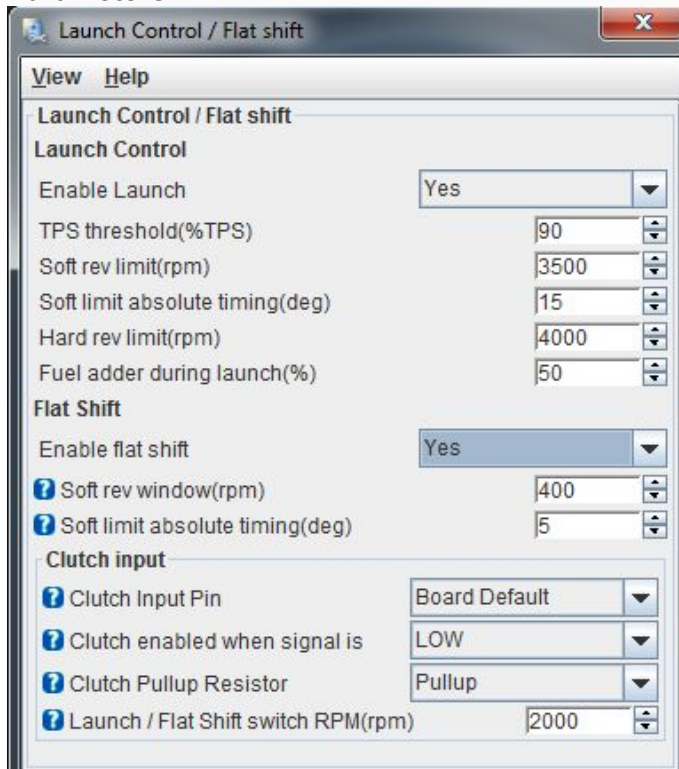
**Fan hysteresis** – The number of degrees below the fan setpoint that will turn off. This is used to prevent oscillation around the set point which results in the fan turning on and off rapidly.

## Launch control/flat shift

### Overview

Allows you to adjust the launch mode as well as being able to adjust rev trim to gear change.

### Parameters



### Launch control

**EnableLaunch:** Allows you to activate or deactivate the Launch function.

**TPS threshold (%TPS):** Allows you to configure the launch control activation window by TPS position, the TPS position must exceed the value of this parameter.

**Softrevlimit (rpm):** Defines at how many rpms the soft type cut will occur (alternating cut between cylinders, with advance delay).

**Softlimitabsolutetiming (deg):** Defines the ignition advance that the ECU will command when the soft cut mode is activated (it will ignore other parameters or tables, it is absolute mode and allows negative advance numbers).

**Hardrevlimit (rpm):** Defines at how many rpms the hard cut will occur (total spark cut).

**Fuel adderduringlaunch (%):** Defines how much percentage of extra fuel will be administered during launch mode.

### Flat Shift

**Enable flat shift:** Allows you to activate or deactivate the cutoff function at gear change.

**Softrevwindow (rpm):** The number of RPMs below the flat shift point where the soft limit (also known as the soft limit window) will be applied. Recommended values are 200-1000

**Softlimitabsolutetiming:** The absolute time (BTDC) to be used within the Flat shift window.

## Clutch Input

Allows you to configure the digital input for activation of the Launch signal.

**Clutch input pin:** The Processor pin that the clutch switch is connected to. This is NOT the pin of the connector, but the pin it is related to on the Processor. It is recommended to use Board Default.

**Clutch enabled when signal is:** Whether the signal is High or Low when the clutch pedal is activated. For a switch input to ground (most clutch switches), this should be LOW.

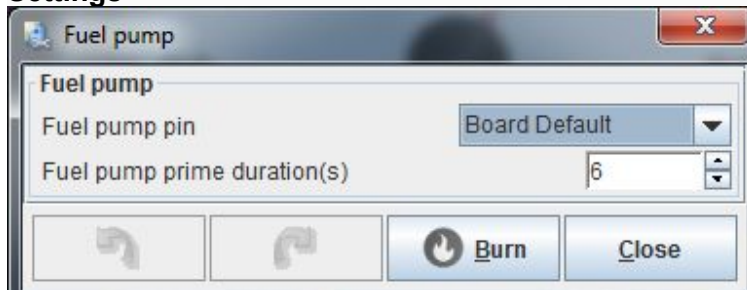
**Clutch pullup resistor:** Whether the internal pullup resistor is enabled or left floating. For a switch input to ground (most clutch switches), select Pullup.

**Launch / flat shift switch:** The RPM switch point that determines whether an engaged clutch is for launch control or flat shift. Below this figure an engaged clutch is considered to be for launch, above this figure an active clutch input will be considered a flat shift. This should be set at least several hundred RPMs above idle.

## Fuel Pump Fuel

pump control is a simple but important function performed by the ECU. Currently ProjectECU does not perform variable pump control (PWM), but it can activate a relay for the fuel pump.

### Settings



**Fuel pump pin:** The CPU pin to which the fuel pump output is assigned. In most cases, this should be left at Default unless you have a specific reason to change this.

**Prime duration** – How long (in seconds) the fuel pump should run when the system is first turned on. Note that this is activated when the ECU is turned on, which will not always be the same as when the ignition is turned off.

## Boost Control

Open and Closed Loop Boost Control Settings

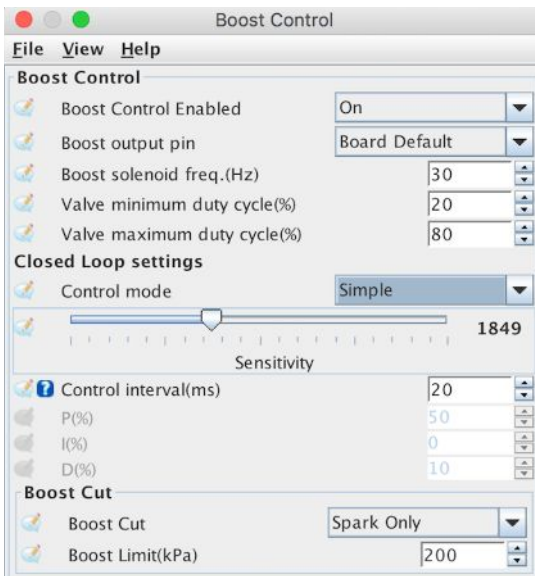
### Overview

ProjectECU has an integrated closed loop Boost controller that can be used to regulate standard single turbo settings.

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

### Configurations

The ProjectECU Boost control uses a PID algorithm with 2 modes of operation, Simple and Full. Each has its own advantages and disadvantages, as described 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. Simple mode can be quick and easy to set up, however it has the drawback that to avoid overboost, you may need to set the sensitivity low, which can increase lag.

## Destination Table

Boost targets (Absolute kPa)

File View

Boost targets (Absolute kPa) 3D View

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

Buttons: Undo, Redo, Burn, Close

The Boost Map function varies depending on whether Open Loop or Closed Loop Boost control is selected.

In closed loop mode, this map serves as a destination table. The values on the map are the desired boost pressures (in kPa). In closed loop mode, these target values can be optionally modified by a flex fuel value, if available.

In open loop mode, the map values are the percentages of the duty cycle that will be used.

## Tacho

ProjectECU has a tachometer output with basic parameters.

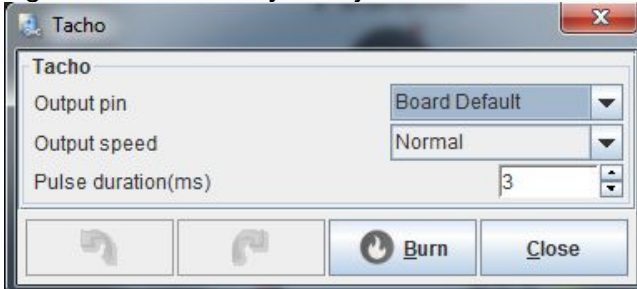
## Settings

Output, output speed, and pulse width settings are available only.

Output pin- the Processor pin that the Tach output is assigned to. In most cases, this should be left at Default unless you have a specific reason to change this.

**Output Speed-** It is used to configure the number of output pulses to half (for some tachometers that mark twice the RPMs).

**Pulse duration-** Pulse width duration in milliseconds, this parameter is for tachometers where you lose the signal, it is necessary to adjust to thinner or thicker pulses so that the tachometer detects perfectly.



## Tools Menu

### Sensor Calibration

Steps to Calibrate Analog Sensors

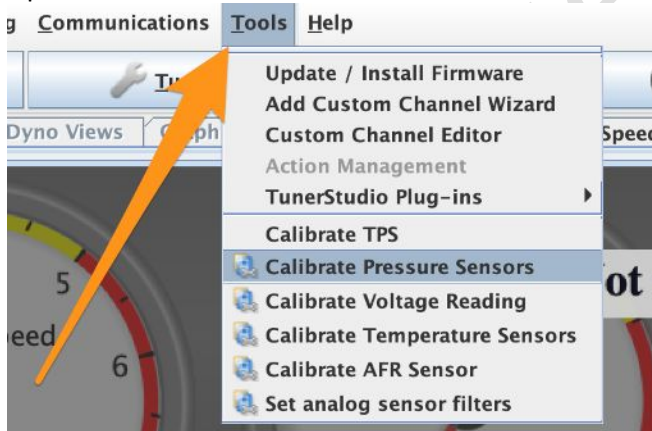
#### Sensor Calibration

Before your ProjectECU can correctly interpret sensor signals, it must know which sensors you are using. Entering this information into TunerStudio (TS) writes the correct calibration to your ProjectECU. This step is required before you can effectively verify your ProjectECU installation. Note that this is not Tuning your system, but just telling you how to understand the signals from the sensors.

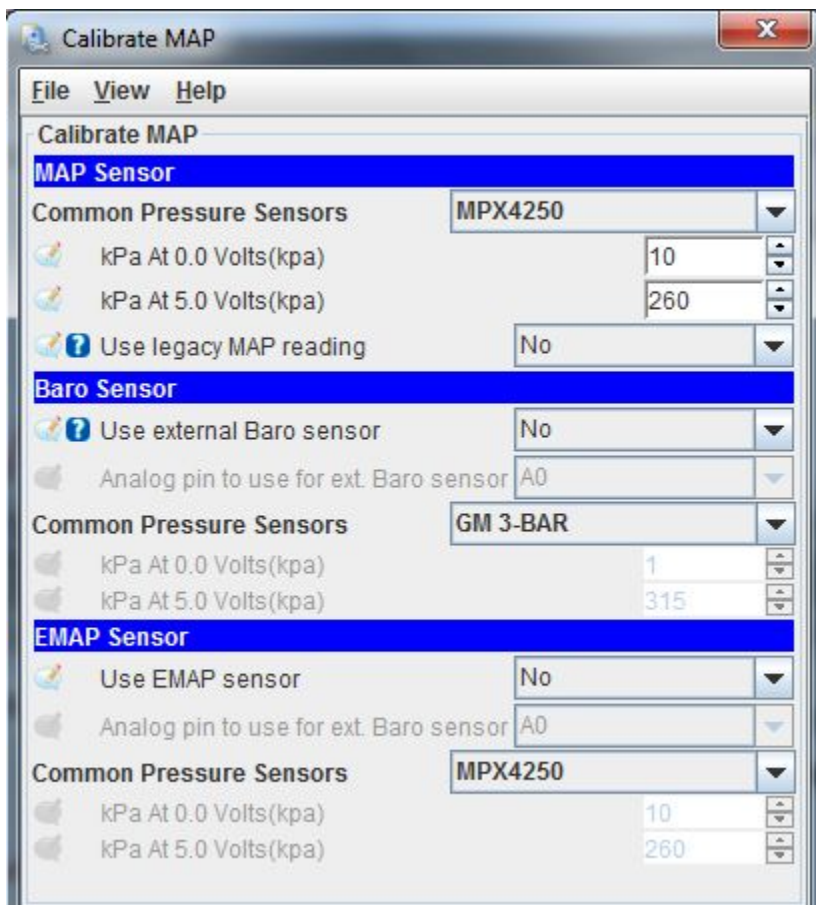
This should be done after completing your engine setup. Your Laptop must be connected to your ProjectECU through TS to perform the calibrations.

#### MAP Sensor (calibrate Pressure Sensors)

Open the Tools menu:



tools\_menu Select Calibrate Pressure Sensors, the following window will open:



Select your MAP sensor from the drop-down list. If you have doubts about which one to use, ask in the ProjectECU group or in the page. If you are using another MAP or one from the engine manufacturer, select it from the list. Click Save to send the information to your ProjectECU.

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

**Use legacy MAP reading:** Select if you already had a tune done in an old version of ProjectECU. With this the tune will not be affected by the new firmware version.

### Inlet and Coolant Temperature Sensors (calibrate Temperaturesensors)

Open the Tools menu and select Calibrate Thermistor Tables:

The selected sensor will be the coolant temperature sensor. Select your sensor from the Common Sensor Values drop-down list. This will put the correct values in the temperature and resistance tables and the bias resistance value. If your sensor is not listed, see Entering Custom Values below.

Note that ProjectECU's standard construction is to have a 2490 ohm pull-up resistor, which is standard for sensors used by most manufacturers. If your sensor requires another value, you may need to change the resistor to the correct value for your sensor. Puede intentar mover el valor de resistencia de polarización con 2490 ohmios, pero asegúrese de que su sensor lea correctamente en TS.

Calibrate Thermistor Tables...

Help

Calibrate Thermistor Tables...

Sensor Table  
Coolant Temperature Sensor

Table Input Solution  
3 Point Therm Generator

Thermistor Measurements  
Common Sensor Values GM

Bias Resistor Value (Ohms) 2490.0

☐ Fahrenheit ☒ Celsius

Temperature(°C)	Resistance (Ohms)
-40.0	100700.0
30.0	2238.0
99.0	177.0

Select settings, click "Write to Controller"

Write to Controller

Close

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

Calibrate Thermistor Tables...

Help

Calibrate Thermistor Tables...

Sensor Table  
Air Temperature Sensor

Table Input Solution  
3 Point Therm Generator

Thermistor Measurements  
Common Sensor Values GM

Bias Resistor Value (Ohms) 2490.0

☐ Fahrenheit ☒ Celsius

Temperature(°C)	Resistance (Ohms)
-40.0	100700.0
30.0	2238.0
99.0	177.0

Select settings, click "Write to Controller"

Write to Controller

Close

Select your sensor from the Common Sensor Values drop-down list. This will put the correct values in the temperature and resistance tables and the bias resistance value. Click Write to Controller to send this information to your ProjectECU. If your sensor is not listed, see Entering Custom Values below.

Note that ProjectECU's standard construction is to have a 2490 ohm pull-up resistor, which is standard for sensors used by most manufacturers. If your sensor requires another value, you may need to change the resistor to the correct value for your sensor. You can try overriding the bias resistor value with 2490 ohms, but make sure your sensor reads correctly at TS.

### Entering Custom Values

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

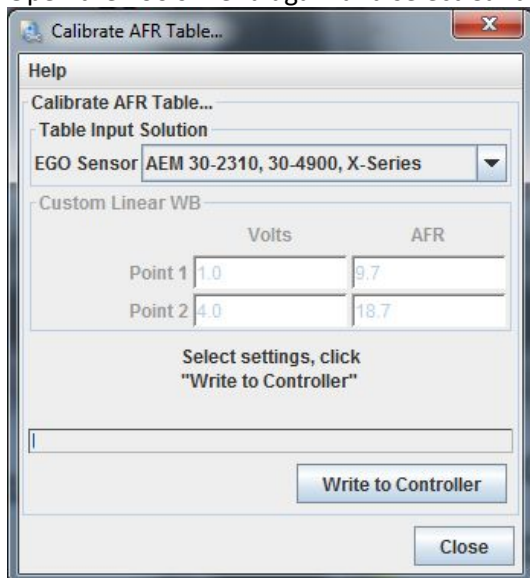
Sensor resistance can be generated by measuring the resistance of the sensor in ambient air, placing it in a freezer, and then in boiling water. You will need a good multimeter and an accurate thermometer that measures -10C to 100C (14°F to 212°F). It is best to use jumper wires to allow the resistance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

On the Calibrate Thermistor Tables screen, first make sure you select the correct temperature unit (F or C). Then record the bias resistor value and temperature/resistance values in the fields. Click Write to Controller to send this information to your ProjectECU.

Note that your procedure can also be used to input resistance values into simulators for testing and troubleshooting. There are two points to remember if you use simulator values: First, never enter zero for the resistor. Although your simulator can go to zero, enter a 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 ProjectECU!

### Calibrate Oxygen Sensor (calibrate AFR sensor)

Open the Tools menu again and select Calibrate AFR Table:



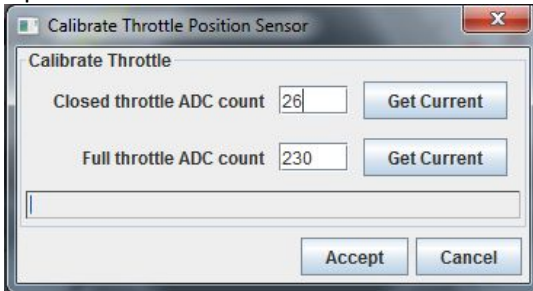
Select your oxygen sensor from the Common Sensor Values drop-down list. If you are using a custom oxygen sensor controller select Custom Linear WB and then you can enter the Volts and AFR values at two points (should be published in the controller manual).

Click Write to Controller to send this information to your ProjectECU.

### Throttle Position Sensor (calibrate TPS)

Before ProjectECU can work properly 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 next to the Closed throttle ADC count field. Then move the throttle to full open and hold it there. Then click the Get Current button next to the Full throttle ADC count field.

Click OK to save the information to ProjectECU.

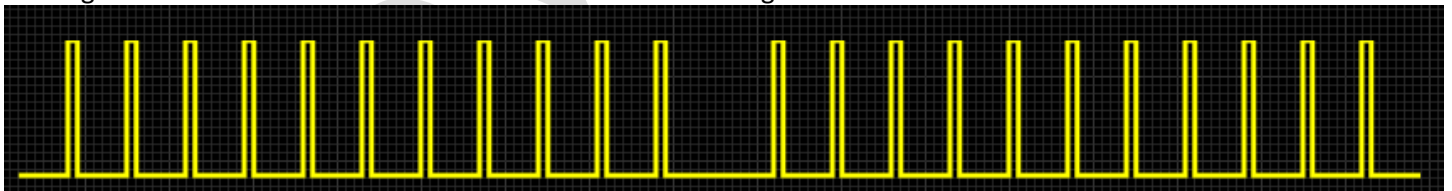
## CKP/CAM Sensor Patterns

### Supported Crankshaft/Shaft Patterns List

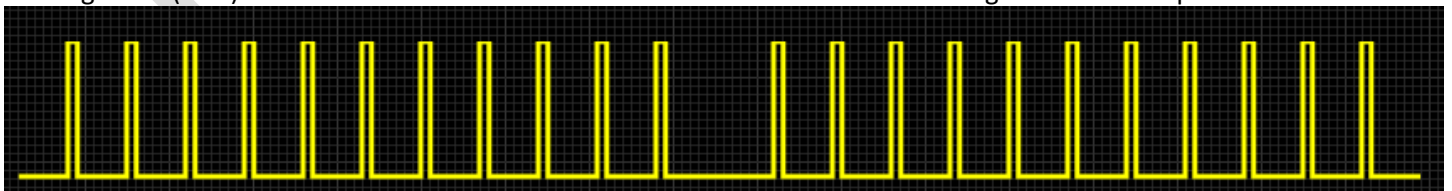
ProjectECU supports an increasing number of crankshaft/shaft and CAS decoders. This includes some of the most common OEM configurations, as well as aftermarket favorites (such as MissingTooth tone wheels).

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

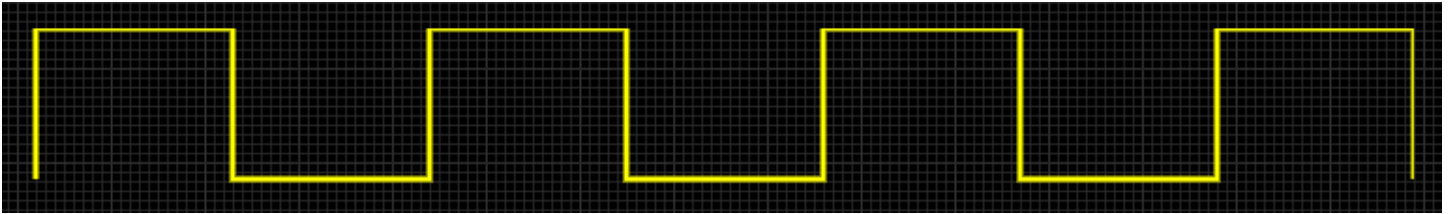
MissingTooth. TRIED. A crankshaft wheel with 1 or more 'missing' teeth



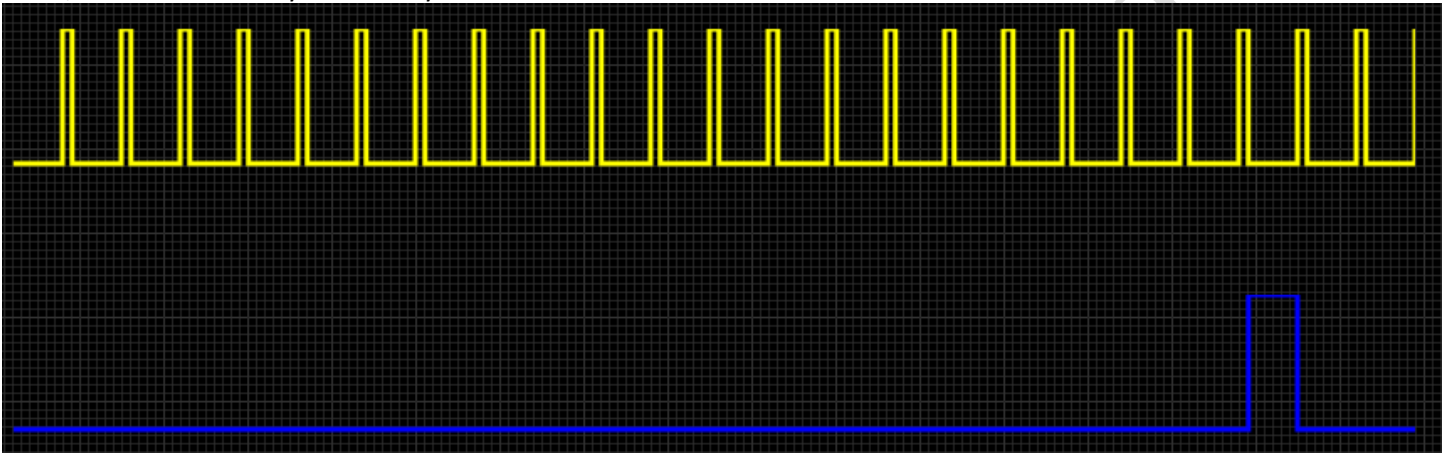
MissingTooth (cam). TRIED. A camshaft wheel or distributor with 1 or more 'missing' teeth at half speed



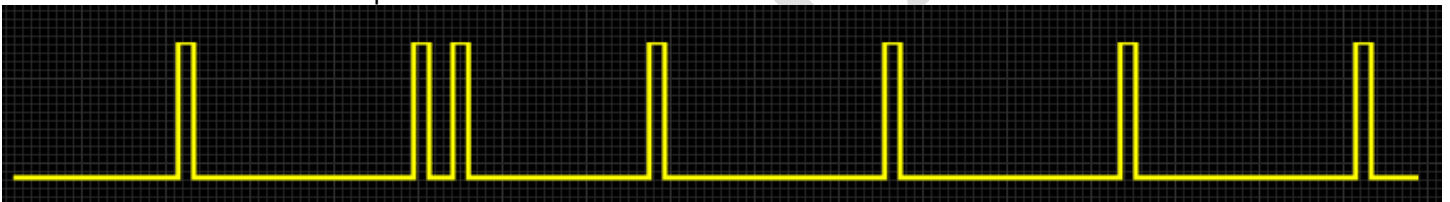
Basic Distributor. TESTED. Untimed pulses that are equivalent to a tachometer or



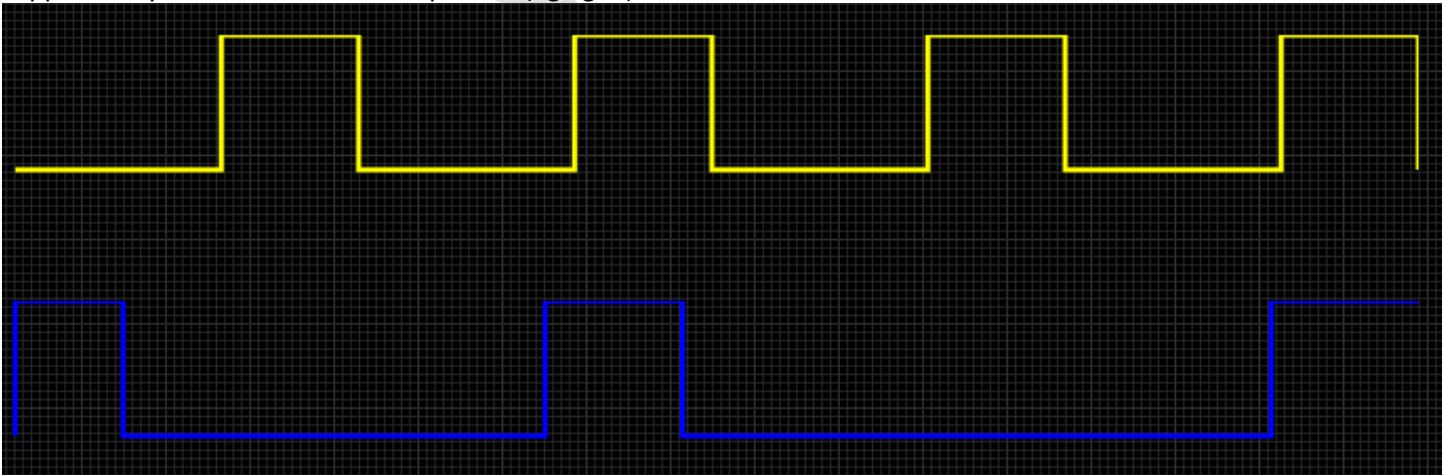
Dual Wheel distributor pulse. TRIED. Two signals combined from two different wheels, the primary wheel has no missing teeth, and the secondary wheel only has 1 tooth on TDC



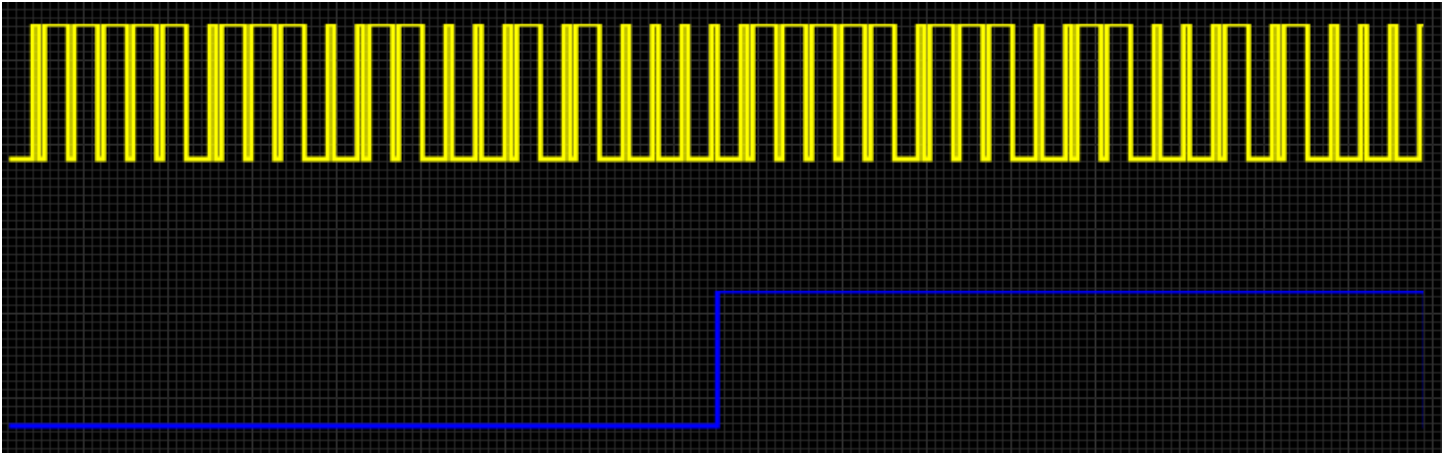
GM 7X. Not tested.multi-tooth pulse



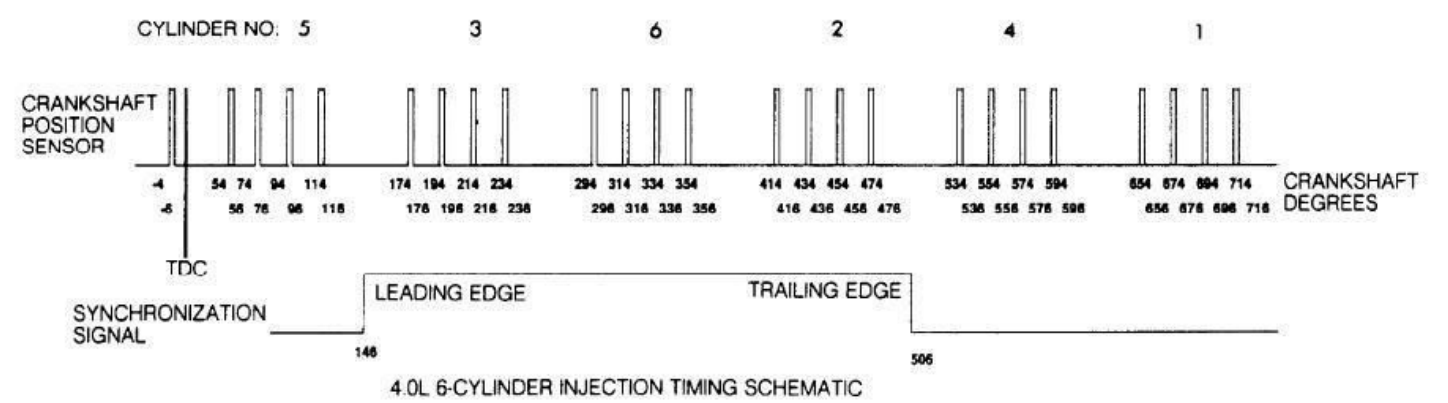
4G63Fill in. Not recommended but functional. As used on many 4 cylinder Mitsubishi's and NA/NB Miata/MX-5's. Also supports 6 cylinder variation of this pattern (eg 6g72)



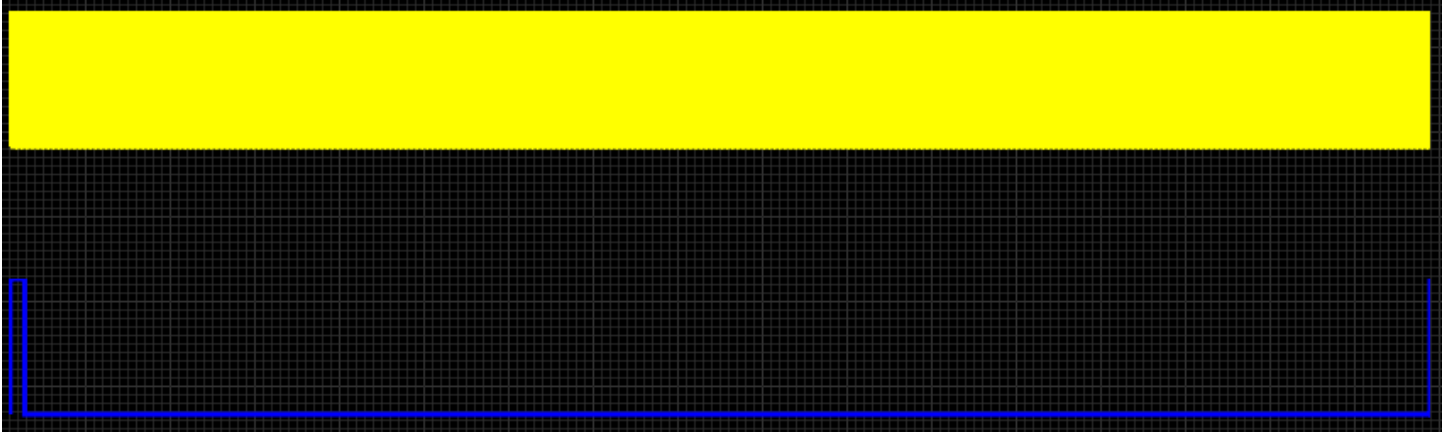
GM 24X. Not tested. Commonly used in GM LS1 V8



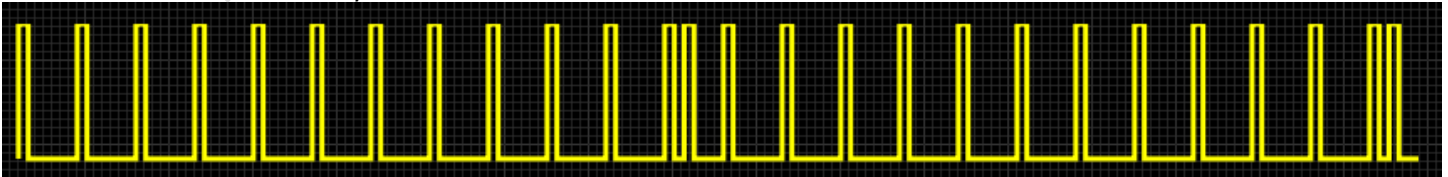
Jeep 2000.Complete. Jeep 6 cylinder engines from '91 to 2000



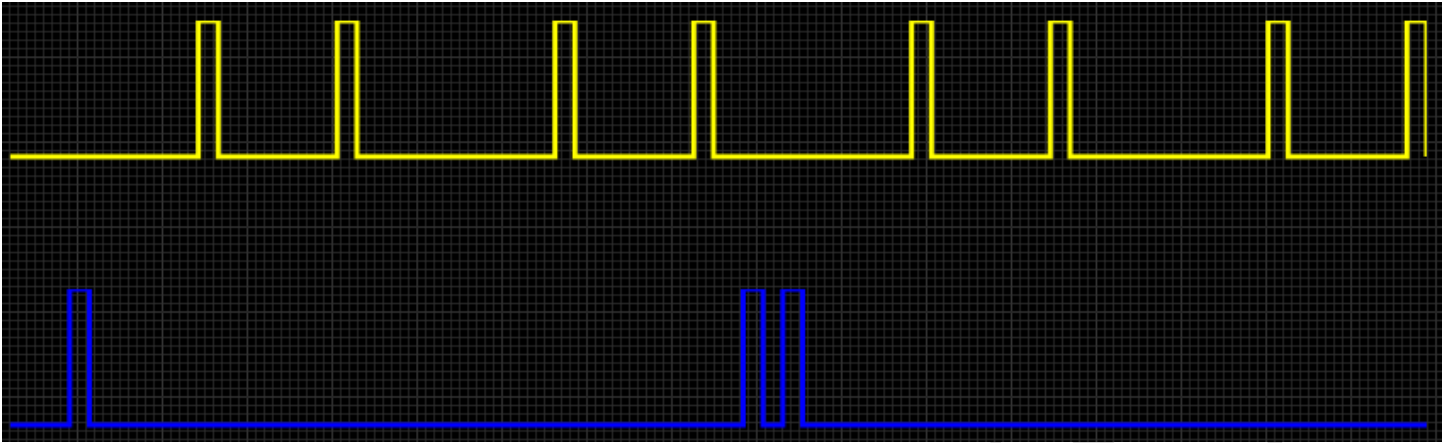
Audi 135. Complete. Complete Audi engines with 135 pulses per revolution



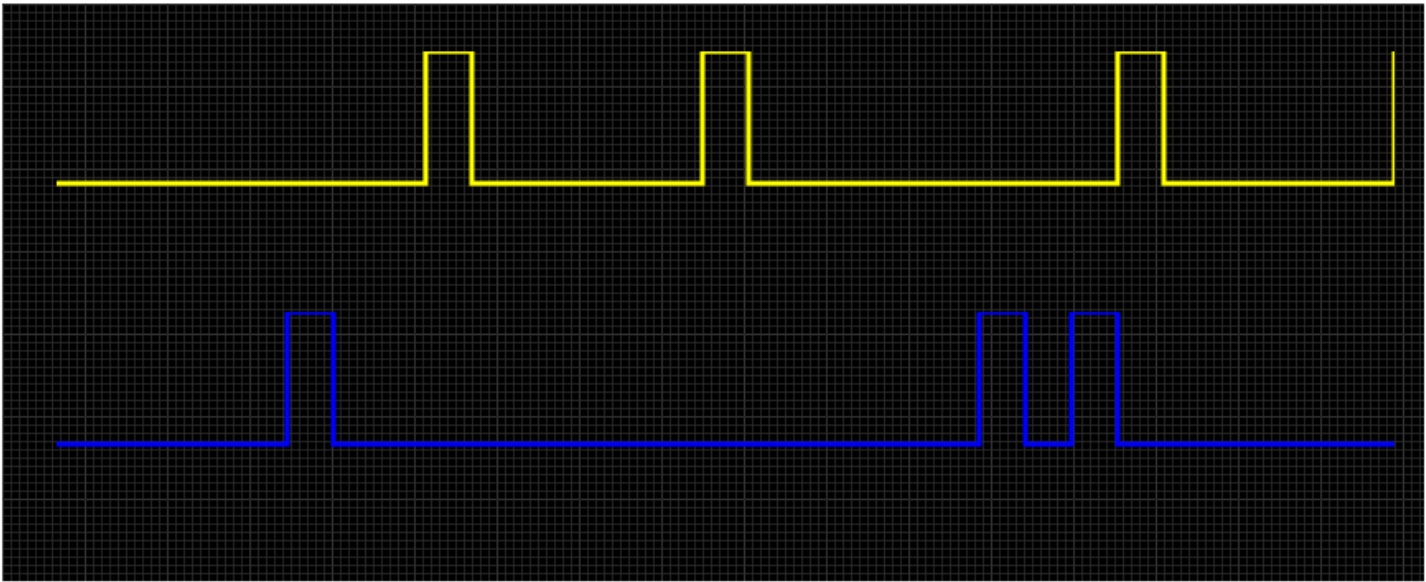
Honda D17. TRIED. Honda 4 cylinder D17



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



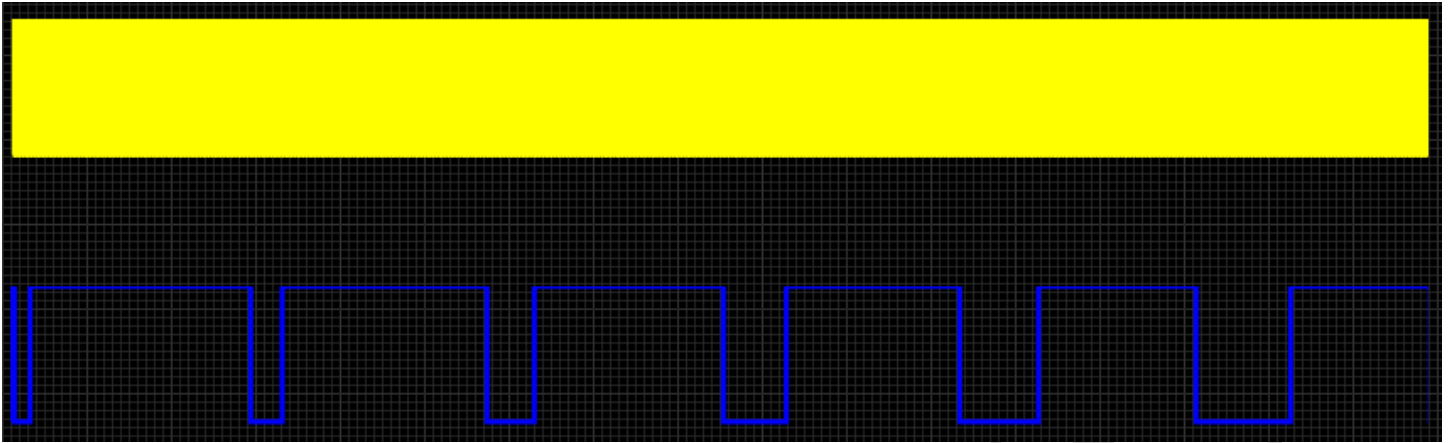
Mazda AU. Not tested. Used in the Mazda 323/Familia/Protegé



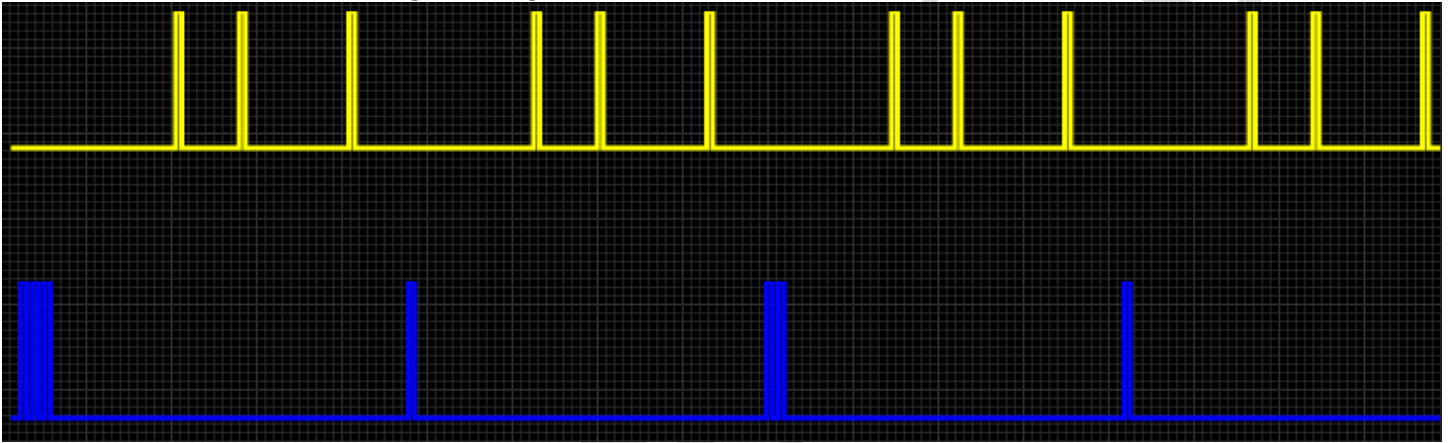
Non-360.Complete. A variation of the dual wheel decoder that can be used with tooth counts that are not evenly split on 360



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



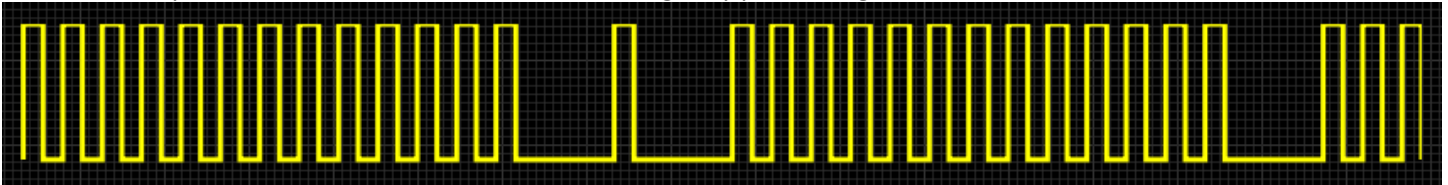
Daihatsu +1 cam wheel arrangement. Fill in. Patterns 3 + 1 and 4 + 1 used in Daihatsu 3 and 4 cylinder



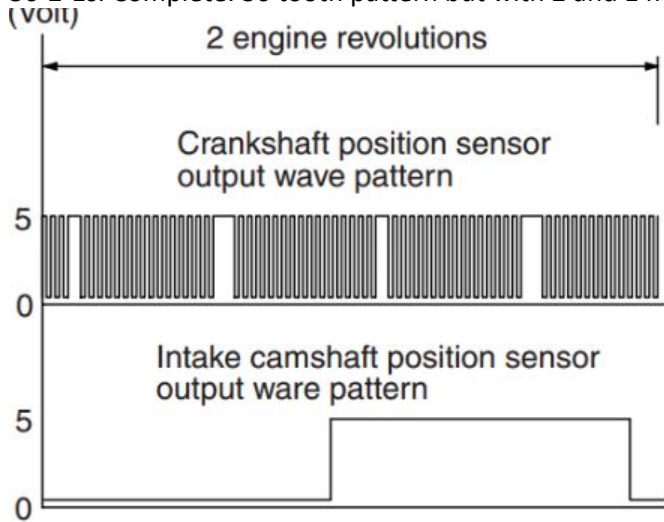
Harley engines. Complete. The Harley EVO pattern is used on V-Twin engines from '86 through '99. This pattern will work on all injected EVO engines.



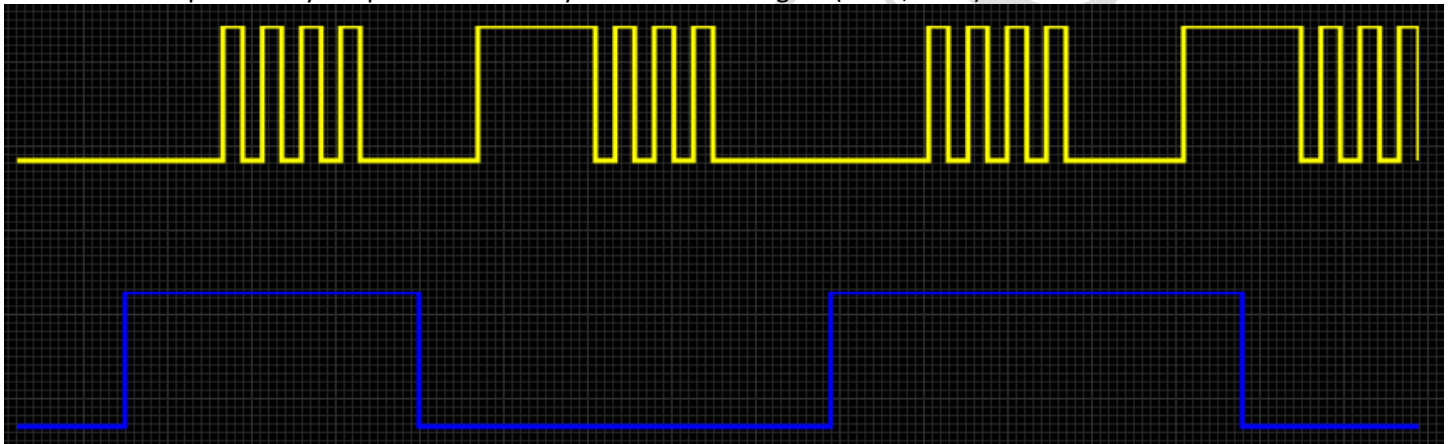
36-2-2-2. Complete. A 30-tooth wheel with three strategically placed large slots. Used on some Subaru



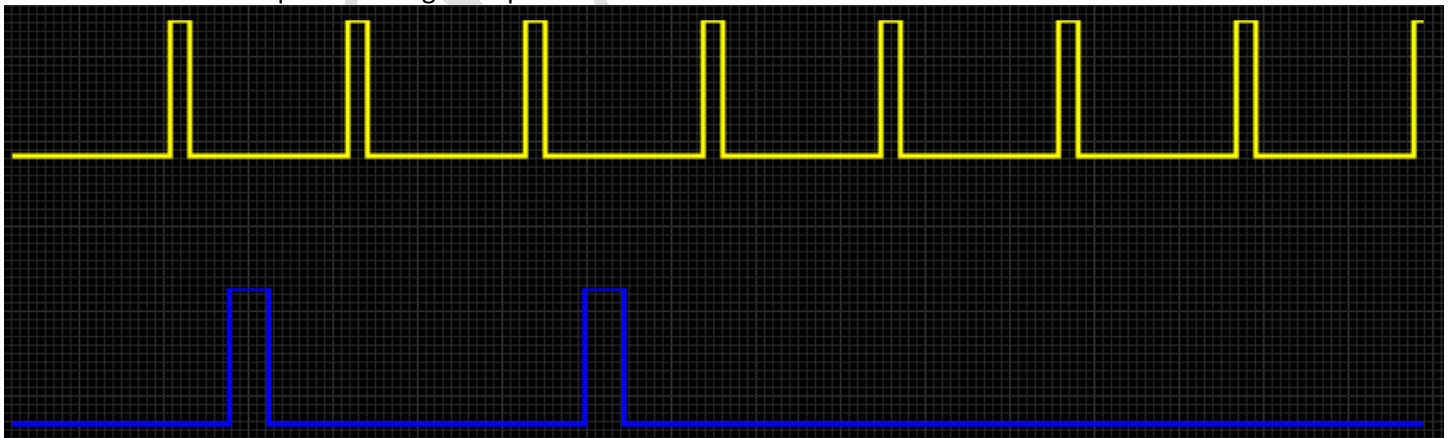
36-2-1s. Complete. 36 tooth pattern but with 2 and 1 missing teeth. Used in the Mitsubishi 4B11



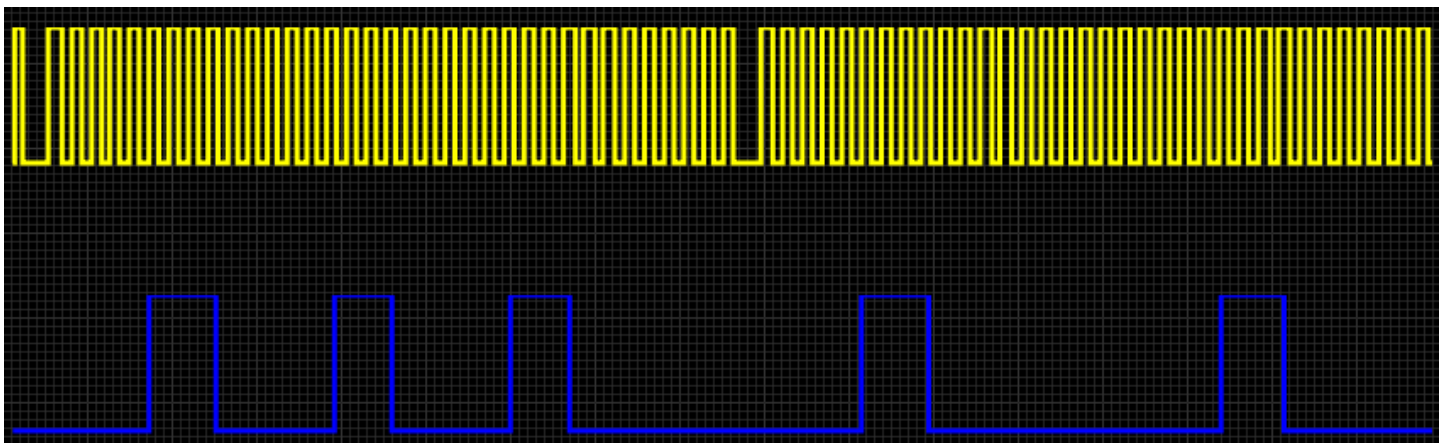
DSM 420a. Complete. Chrysler pattern used only for the 420th engine (neon, DSM)



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



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



DRZ400.Complete. It is used in Suzuki DRZ400



NGC motorcycle engine. Not tested. For 4, 6 and 8 cylinder engines equipped with the Chrysler NGC pattern. Seems to be used on some Chrysler/Jeep/Dodge vehicles from 2002 onwards

