upload software revisions. (*Roger Enns*)

⇒ **MS Palm** - to tune and datalog with a Palm (*Roger Enns*)

To tune all the parameters of MegaSquirt so that your engine runs the best it can, you will need to do the following::

(1) First, learn to use MegaTune,
(2) Next, set the constants,
(3) Get the engine started and idling,
(4) Then set the PWM criteria,
(5) Then set the cold start and warm-up enrichments,
(6) Then set the VE table,
(7) Set the acceleration enrichments,
(8) Check that certain resistors are not getting too hot while driving.

We will go through each of these steps in turn.

Using MegaTune

MegaTune is the Windows 95 and later configuration editor for the MegaSquirt EFI controller. It allows all of the parameters to be modified and has a real-time VE table editor, which allows a vehicle passenger to tune the engine while driving.

The front page shows eight gauges, the left four of which are the major inputs to MegaSquirt and the last four are the output pulse width and resulting duty cycle for the two injector banks.

Pulse width is the measure in milliseconds of how long the injector is opened for each pulse, regardless of how many times it is opened in a cycle. Duty cycle gives the percentage of time the injector is open irrespective of individual pulse duration.

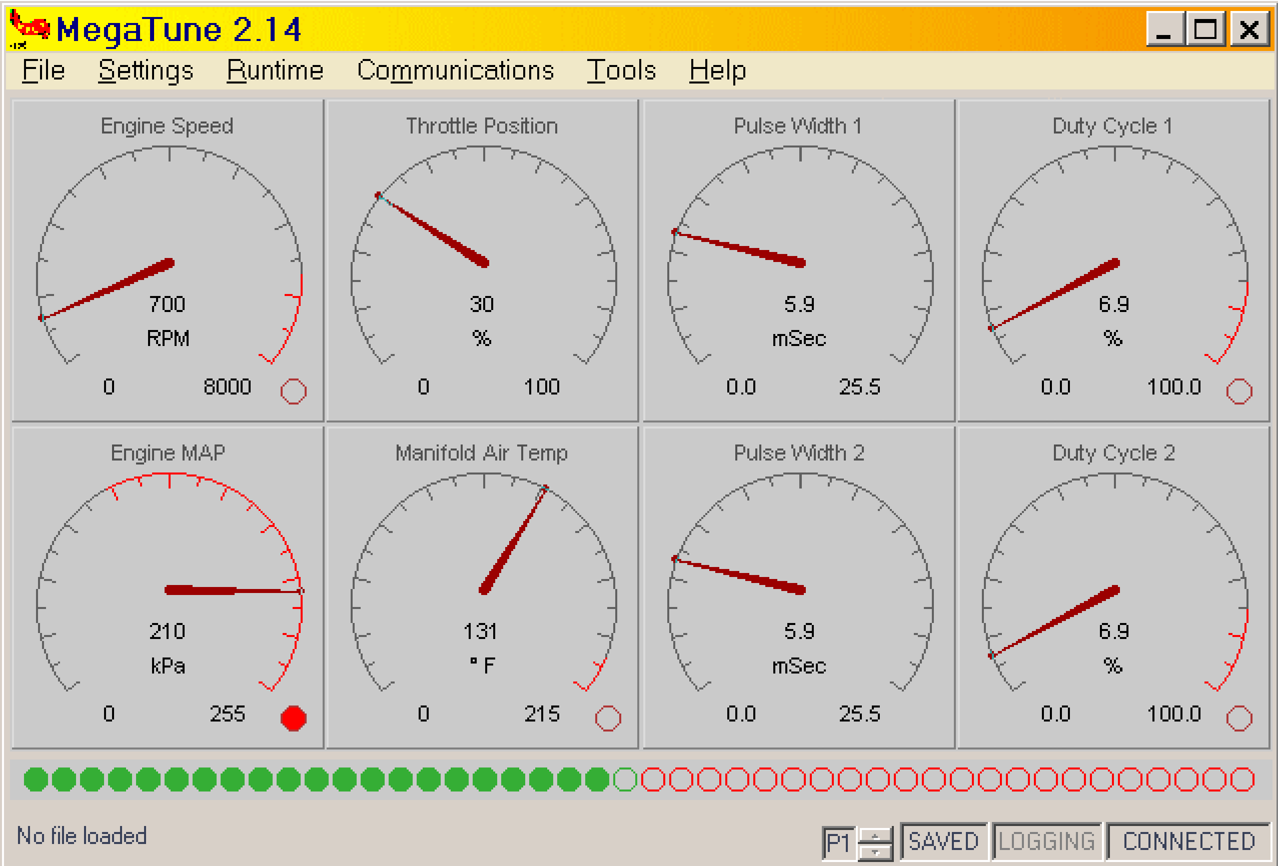
The bar gauge across the bottom of the window shows the oxygen sensor reading. The scale is determined by egoGauge value in the Tuning section of the megatune.ini file. This same setting controls the analog and bar gauges on the tuning screen. The first value of this setting controls the lowest voltage displayed on the gauges, the second number controls the highest and the optional third value specifies the “alert” value, above which the LEDs are red.

The bottom of the front page contains a status bar. The current file name (used for Save operations) is displayed in the left part of the status bar, followed by “saved” status. When the memory image has been modified since the last Open or Save operation, this entry shows “SAVED” in bold face.

MegaTune allows you to save and restore configurations as disk files. Use the Open, Save and Save As entries to do this.

Setting the Constants

Before attempting to start your MegaSquirt equipped engine, you will need to set a number of parameters that determine how MegaSquirt injects fuel. These include the injector open time, Req_Fuel, injector control criteria, PWM criteria, EGO characteristics, etc.. These constants are either calculated, or based on the configuration of your system.



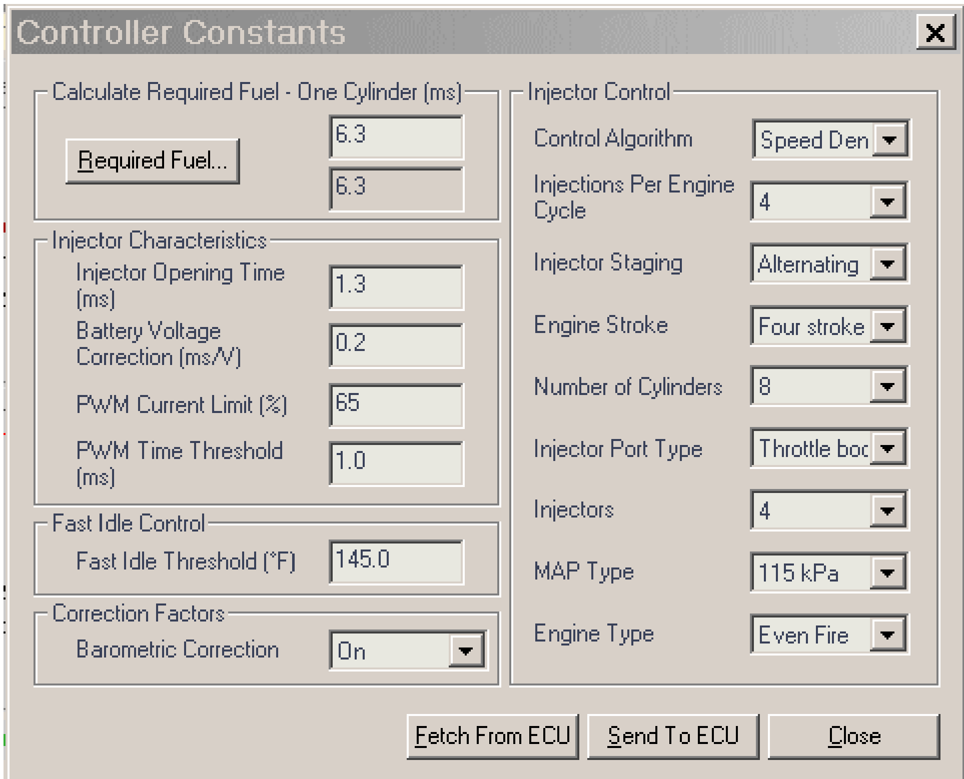
On the Settings/Constants screen:

If you have low impedance injectors,

- set the PWM time threshold to 1.0 ms,
- and the PWM % to 75%.

You will tune these after getting the engine running. See "Setting the PWM Criteria" in this section. **Failure to perform these steps can result in damage to your injectors.** If you have high-impedance injectors, set these vales to 25.4 ms and 100%, and you do not need to tune them further.

Control Algorithm lets you choose between Speed Density and Alpha-N. In all cases, you should choose speed density unless you have a good reason to do otherwise, and understand how this will change your tuning efforts. All tuning advice in this manual is based on the speed-density algorithm. Alpha-N uses the throttle position ('alpha') and RPM ('N') to calculate the amount of fuel to inject as opposed to using the manifold absolute pressure (MAP) and RPM to calculate the amount of fuel to inject. Alpha-N is useful for long duration cams where the resolution of manifold air pressure (map) would be small. It is also useful to get smoother idle on engines that have erratic map values. MegaSquirt be converted from its default **'speed-density'** calculations to **'alpha-N'** which uses RPM , temperature and TPS only. You must have v2.0 of the embedded software installed. Start up the tuning software, go to the Constants dialog and change speed density to Alpha-N. Re-map your VE table.



You will no longer use the MAP sensor for estimating the load on the engine -- the throttle position and rpm are used instead. This can help with cams with long duration and/or a lot of overlap, as they have low and variable vacuum at idle, making tuning very difficult.

Required Fuel - One Cylinder (ms) is top field of the Constants Req_Fuel dialog. It should contain the injector pulse width, in milliseconds, required to supply the fuel for a single injection event at stoichiometric combustion and 100% volumetric efficiency. In order to come up with this value, MegaTune provides a calculator that will suffice for 99% of applications (those for which it will not work generally require changes to the MegaSquirt controller code itself, and that is beyond the scope of this manual).

Injector Opening Time (ms) is the amount of time required for the injector to go from a fully closed state to a fully opened state when a 13.2 v signal is applied. Since fuel injectors are electro-mechanical devices with mass, they have latency between the time a signal is applied and the time they are in steady-state spraying mode.

The current MegaSquirt controller code makes the assumption that NO fuel is injected during the opening (and closing) phases, so making this value larger will enrich the mix and will have a much greater effect at low pulse widths. MS also uses this value as an additive constant in pulse width calculation, thus making this the lower limit for pulse width.

Injections per Engine Cycle set the number of squirts you want per engine cycle. You want this to be set so that your idle pulsewidth is no less than 2.0 ms, if possible, and your Req_Fuel is less than 12-15 milliseconds. These values allow proper tuning of the idle mixture while maintaining the ability to apply enrichments (acceleration, warm-up, etc.) under full throttle. This is the total injector events that you wish to occur for every engine cycle (360 degrees for two stroke engines and 720 for four strokes).

Injector Staging values for injector staging are simultaneous or alternating. If you wish for all your injectors to fire at once, select simultaneous. More likely you will choose alternating, as this helps even out the pressure fluctuations in the fuel rail.

Engine Stroke values for engine stroke type are two-stroke or four-stroke. MegaSquirt uses engine stroke to determine how many degrees are in an engine cycle.

Number of Cylinders is the count of the cylinders on your engine. If you are unsure how many cylinders your engine has, you shouldn't be installing MegaSquirt on it. This value is actually the number if ignition events per cycle sent to the ignition input on the controller.

Injector Port Type is used to select the type of injector that you are using, throttle body or multi-port.

MAP Type values for this may be selected from the option menu, and are either 115 kPa or 250 kPa. All Version 2 MegaSquirt's have the 250 kPa MAP sensor. This should be auto-detected from MegaSquirt, but if it is not, select the right one and hit "Send to ECU" right away.

On the Communications/Settings screen:

Port
The communications port number should correspond to the port to which the MegaSquirt controller is attached.

Timer Interval (ms)
The timer interval dictates how frequently the runtime and tuning displays are updated. An interrupt is generated at the specified interval, and the real time data is pulled down from the MS controller. Use 100-200 ms to start; you can try to smaller values (ex. 50 ms) if your computer is fast enough.

Verify ECU Communications
Click this button to attempt communications with the MegaSquirt controller. Success will be reported.

<u>Get the Engine Started and Idling</u>

You start, naturally enough, by getting your engine started. The first time MegaSquirting typically starts after about 5 minutes of alternating various REQ_FUEL numbers (on the tuning screen) and a bit of cranking. If you have been at it much longer than this, you ought to investigate other sources of problems before continuing to try to start the engine.

You start tuning by just getting it to idle properly by adjusting the speed [with the throttle stop and/or FIdle solenoid], and mixture [with the VE table or Req_Fuel]. There' s no danger of harming anything, as there' s not enough load on the engine to build the heat that would melt anything. And the engine will idle on a very wide range of mixtures, so it is not too hard to get it started, you just play around with the REQ-fuel until it fires. Afterwards you can reset your Req_Fuel and adjust your VE to get the same idle pulsewidth.

Setting the PWM Criteria

To tune the PWM [pulse width modulation] values for your engine, you need to know what kind of injectors you have- low impedance or high-impedance. If you are running **high-impedance injectors** (greater than 10 Ohms), then set the PWM time to a number like 25.4, in essence you are disabling the PWM mode. This allows full voltage to the injectors throughout the pulsewidth.

For **low-impedance injectors** (less than 3 Ohms), you need to limit the current to avoid overheating the injectors. To do this, there is a period of time that you apply full battery voltage [peak] current, then switch over to a lower current-averaged [hold] current, i.e. peak and hold.

To run low-impedance injectors with the PWM current limit mode, you need to set two parameters - the "**PWM Current Limit %**" and the "**Time Threshold for PWM Mode**" - both are on the ' constants' page. The current limit % is the percent duty cycle when the current limit is invoked. The time threshold is the amount of time from when the injector is first opened until the current limit is activated.

Start with 75% PWM and 1.0 millisecond time threshold. Once you get idling, then first adjust the PWM duty cycle down in 1% increments until you notice a change in idle quality(be sure to hit the "send to ECU" button each time you change the value). This is the point where the current limit is too much and the injectors are not being held fully open. Then move the value back up 3 - 5% (for example, if the idle falters at 45%, then put in a number of 48% to 50%) and move on to adjusting the time threshold. Lower the time threshold by 0.1 milliseconds at a time until the idle quality deteriorates. Then increase it 0.3 ms.You can go back and forth adjusting the duty cycle and time threshold alternately to get the optimum values for your set-up.

On the car this is very easy to do and only takes a few minutes. And, at idle, the overall injector pulsewidths are small compared to their close time, so this will allow you to adjust the values. In other words, adjust the PWM current limit before taking the car out on the street where injector pulsewidths become high.

Now, repeat the two steps again until you converge on a set of numbers that work for your set-up. Also, for some setups, 75% may be too low, so they will need to increase this value - same for the time threshold. **Use PWM time threshold values greater than about 1.5 to 1.7 milliseconds only with great caution - you can burn out your injectors!**

Setting the Cold Start and Warm-Up Enrichments

If it is cold out, you have to figure out the cold start enrichments/warm up enrichments right away to keep the engine running as it warms up. During cranking mode (defined when RPM is less than 400), MegaSquirt shoots out cranking pulsewidths which come from a linear interpolation of two end-point values defined by the user, one at - 40 degrees F and one at 170 degrees F.

For Bruce' s Jag with a Chevy 350, he has values of 2 milliseconds at 170 degrees F and 10 milliseconds at -40 degrees F. The values for your combination are likely different, though generally should follow a similar pattern. During cranking, MegaSquirt injects one pulse for every ignition event, so for the Chevy 350 8-cylinder it shoots out 8 times for 720 degrees crankshaft, all injectors. So, at 170 degrees, the effective amount of fuel per cylinder is 8 * 2 = 16 milliseconds. For - 40 degrees it is 10 * 8 = 80 milliseconds.

Once the engine fires up (defined by engine RPM greater than 400 RPM), the engine goes into an afterstart enrichment. The afterstart enrichment starts out at a user-defined percentage enrichment value (20 % for the Bruce' s Chevy motor), and ramps down to 0% after so many ignition trigger events, user-defined (Bruce uses 200 for this number for his Chevy engine). This is an enrichment above the normal warm-up enrichment, which is temperature dependent. You can also set your Fast Idle Threshold (degrees F) if you have installed a fast idle solenoid. Enter the temperature below which you wish to turn on the fast idle solenoid. This is independent of any warm-up enrichment. A typical value is 145 Fahrenheit.

With the above numbers, the engine fires immediately, every time, just like any new OEM fuel injected car. Bruce says it took him a while to converge to these numbers, especially the afterstart enrichment, which needs to be just right, or the engine will run rough or stall immediately after starting.

You can do this as the engine warms by adjusting the warm-up bins, loading it to the ECU, and noting the effect on idle quality. It will take several starts {from a cold soak} to get this close. Then you can play around with revving the engine in neutral and adjusting the mix to stoichiometric. Up to here it easy enough to do without an O2 sensor by adjusting for maximum vacuum (lowest MAP kPa) ant any given rpm.

Setting the VE Table

Have someone ride with you and bring up the tuning screen. See where the "dot" hangs around when you are under load - this is where you need to focus on tuning. Use the up-arrow+shift to richen the VE values - enrich (with increased VE number) the four corners around where the dot is - give each corner five up-arrow-shifts, and see if this helps. Turn off the O2 closed-loop mode by setting the step size to zero. Watch the O2 gauge on the tuning screen and use this as feedback for rich and lean. The 02 gauge may move to fast from rich to lean to be able to tune. Another strategy that works is to turn on EGO correction, and then tune using the EGO correction gauge rather than the EGO voltage gauge. If correction is below 100%, then raise VE to raise correction and so on.

Alternatively, use the O2 sensor makes the driving part of the setting up much easier, as you can datalog and use MegaTweak to get the VE table set up near enough to perfectly with a few easy drives up and down the street, a bit more tuning, and you are ready to go a bit harder. You do not go harder if there' s any problems [typically a backfire means too lean, sluggish revving means too rich]. Figure out why, and fix it.

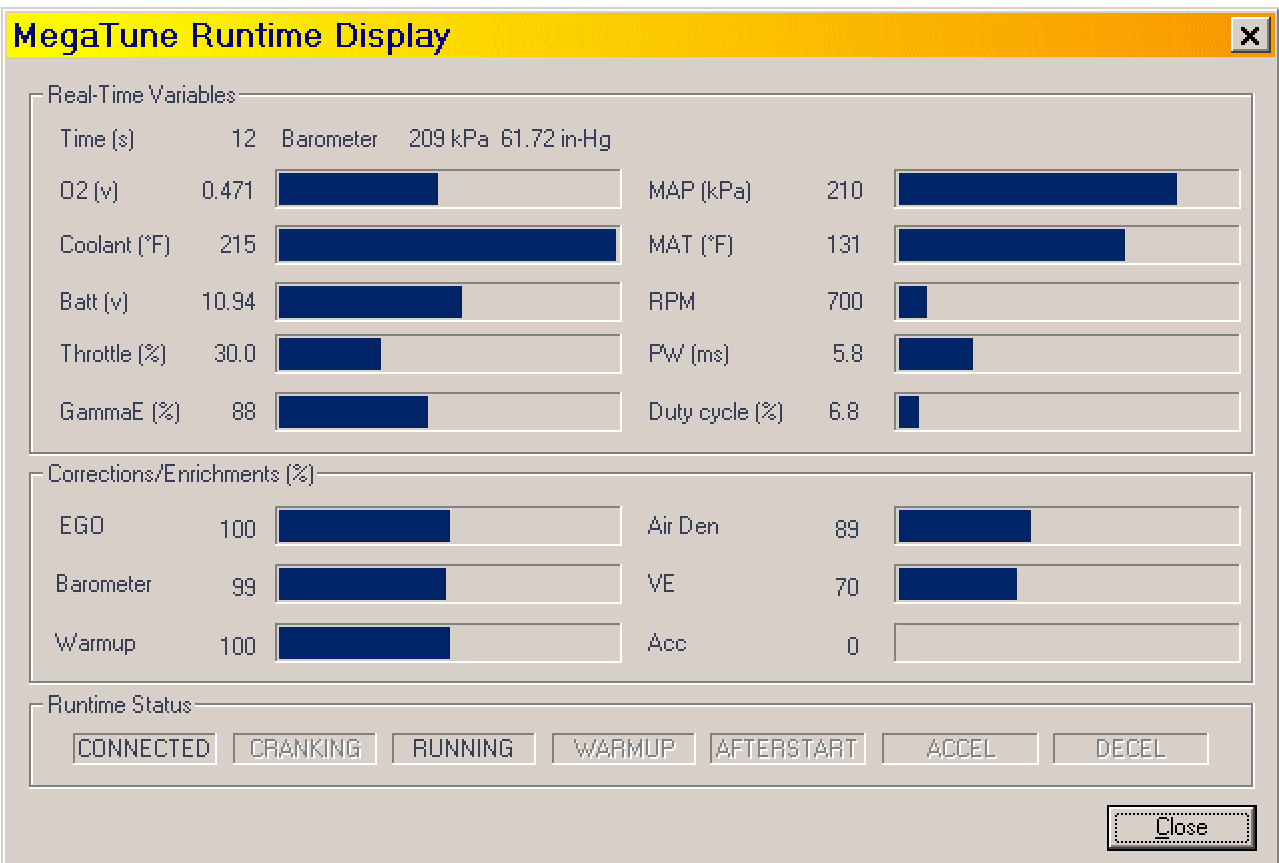
The tuning numbers within the 8x8 grid are gamma values composed of (lambda * VE), where lamba is the ratio of the air/fuel ratio to stoichiometric, and VE is the volumetric efficiency. Values beyond the table bounds are extrapolated at the boundary value, so the surface beyond the table is "flat". Note that you can change the RPM and MAP bins to suit your operating ranges. We customarily call them just VE to simplify the discussion. Very roughly, the pulse width is computed by taking VE as a function of rpm and load (load is MAP in speed density, or throttle angle in alpha-n) and:

PW = VE(rpm,load) * MAP * temperature_correction * Req_Fuel

In other words, the mass of the air is computed using the ideal gas law (pv=nrt) and then that result is combined with a characteristic number for a given injector.

Datalogging and MSTweak3000

Datalogging allows you to create a running record of the MegaSquirt real-time variables. Once you have enabled datalogging (by clicking on the Datalog menu item on the File list), MegaTune polls the MegaSquirt controller when any of the front page, runtime display or tuning screen are active, and writes this data to a file. The file has a comma-separated value format and defaults to having an extension of “xls”, so Microsoft Excel will open them automatically. The datalogs can be used as input to the MSTweak3000 program to automatically correct your VE table. See the MSTweak folder at the Yahoo site.



When datalogging is enabled, the second status box contains a bold “LOGGING” indicator. The rightmost indicator contains either greyed-out “CONNECTED”, meaning that MegaTune is not communicating with MegaSquirt, dark “CONNECTED”, indicating that MegaTune and MegaSquirt communications is working properly, or dark “RESET n”, indicating that the MS controller has been erroneously re-booted n times since MegaTune started talking to it.

Note: the Logging item under the Communications menu is a different function, and does not need to be selected to enable datalogging of the real time variables. See the MegaTune help menu for details on the Comm Logging function.

If you can drive the car at all, start datalogging. Look through the log and MAP-RPM pairs that are near grid points in your VE map, when they O2 sensor reading is significantly below 0.500 (say 0.014), jack up the VE at that point by 10%. When the O2 sensor is significantly above 0.500 (say 0.825) then drop it down by 10%. A couple runs around the block should get things running pretty well.

Or you can use MSTweak3000, which will sort through your datalogs and suggest what VE points need to be changed. Everybody (who is running and datalogging) should try MegaTweak 3000 - it is very powerful and easy to use. It allows you to read in your datalogs, get rid of outlier points, then generate a new VE map. You can pick new 'bin' values - rpm especially (at the peaks and valleys of your filtered datalog) - then calculate a new VE map with the push of a button! No more gazing with crossed eyes at Excel spreadsheets to pick data for tweaking your VE table.

What MegaTweak3000 does is determine what VE entry value will give you a AFR of 14.7, based on the O2 transition point recorded in the file. MegaTweak3000 gives you a RegenVE, which is the VE value for 14.7:1. If you want another AFR, you can estimate it by taking the RegenVE value and multiply by the ratio of the stoichiometric over the desired AFR. For example, if MegaTweak3000 gives you a value of 50% for the VE table, and you want 12.5:1 instead, then 50 * (14.7 / 12.5) = 59 - this is what you plug into the VE table. See the MegaTweak software and help file for more information.

Don't get hung up on actual AFR numbers - for the example above to work, everything else must be dead on, including the injector offset, injector battery voltage correction, REQ_Fuel for your injector flow rate, and air temperature correction. It will get you close enough with the resolutions we are working with, but remember that the only AFR you can nail down with a NB O2 sensor is 14.7:1, everything else is an estimate from this point. If you have a WB-O2 sensor, then you can read the AFR directly from the sensor output voltage and use those results to tweak your VE table.

MegaTweak3000 will not tune your MegaSquirt for you (yet) but it will suggest new VE table entries for you. It is a tool to help you visualize the VE map and choose better VE values and to better place the MAP and RPM bins for your engine. Keep in mind you need a good O2 sensor to do this though (the WB-O2 sensor will work great).

To start, set the O2 +/- limit to 100% for a really rough map, and if your map is more or less tuned in then 50-70% will work too (keep this high though for tuning).

The critical settings are O2 step% and ignition events per step.

·1 When tuning anything in the lower RPM range (1000-3000 rpm) set the **step% = 1** and **ignition events = 32** (2000 rpm with a V6 = 100 events per second = about 3% change in a second).

·2 Then when tuning the higher rpm' s with a rough map, **step% = 3** and **ignition events = 64** (about 3.5 changes per second at 4500 rpm).

·3 When the map is tuned better set the **step% = 1** and **ignition events = 72** which gives the closed loop control some more stability and allows for better fine tuning at the higher rpm' s.

For fine tuning keep the O2 adjustments per second between 3 and 5. And roughing in maps between 5 and 10 (depends on how good the O2 sensor is, if it' s old go lower).

Once the map is tuned in, set the:

- **O2 +/- limit = 5%** (it can go higher depending on how questionable the map is)
- **step% = 1**, and
- ignition events to a value that would switch about 4x a second at your average cruising speed.

You can calculate your:

O2 adjustments per second = ((rpm/120) * cylinders) / ignition events per step

Setting the Acceleration Enrichments

After you have the VE table dialled in, then start adjusting the acceleration enrichment. You may want to try a short acceleration shot time (like 0.2 ms) and jack up the accel enrichment bins. Decel setting of 100% means no cut. 1% means reduce the pulse width by 99%, to 1% of what it normally would be. The low MAP part of your VE table is probably a touch lean, so the NB sensor drops below stoichiometric. If the car doesn' t buck too hard, you are close to correct settings. If it bucks and stumbles, then it' s going too lean and you need to richen that part of the table.

Before tuning decel [or accel] be sure you have your VE table close first! To get the VE table set up, set the delta-TPS setting very high (30v/s or something like that) so that **TPS enrich/enlean** never kicks in. Then, (in steady state) set up VE table.

Check Certain Resistors

If you want your MegaSquirt to be reliable, do not skip this step. You need to tune two of the resistors to values appropriate for your ignition and injector set-up.

Check the temperature of R10 (kit supplied 390 ohm, 1/2 watt, orange-white-brown), which is used in the ignition input circuit from the coil. It should not be too hot to touch with your fingertip. If it is too hot, the value of this resistor may have to be changed depending on application - start with the supplied value (390 ohms), and if runs hot while running, then increase it' s value, in steps, up to 1K (like 470 ohms, 560 ohms, 680 ohms, 1K). Use 1/2 watt resistors.

Check the temperature of R32 (270 ohm, 1/2 watt). This resistor is used in the flyback circuit

Data Log Format

The data log contains a header line indicating the contents of each column:

Seconds - The seconds part of MS' s internal clock. Since this is a one byte value, it wraps around at 255.
RPM - The engine speed in RPM, to the nearest 100 RPM (you will never see anything but zero for the last two digits).
MAP - The calculated MAP sensor pressure in kPa. The correct value for this uses the transfer function embedded in MegaTune (or preferably read from your kpafactor.inc file).
O2 - The O2 sensor value in volts.
EngineBit - Engine mode, a decimal representation of a binary number, see below. 0 = 0000 0000, 1 = 0000 0001, 2 = 0000 0010, ... ,17 = 0001 0001, 33 = 0010 0001, and so on. To understand them, first you have to convert the EngineBit numbers to their 8-bit binary representation. The numeric value has no intrinsic meaning, just the setting of each bit. The least significant bit (the one on the far right in the binary number) indicates "engine running", the next one indicates cranking mode, and so on as below. The bits are numbered from **right to left**, i.e. 6-5-4-3-2-1.
Gego - The EGO gamma in percent.
Gair - The air density correction gamma in percent.
Gwarm - The warm-up enrichment gamma in percent.
Gbaro - The barometric correction factor gamma in percent.
Gve - The current VE as computed inside MS.
Gammae -The total gamma enrichment in percent.
PW – The pulse width in milliseconds.

MegaSquirt Datalog Engine Bit Code

Bit	Mode	0	1
1	Running	0=not running	1=running
2	Cranking	0=engine not cranking	1=engine cranking
3	Start-up Enrichment	0=not in start-up warm-up	1=in warm-up enrichment
4	Warm-up Enrichment	0=not in warm-up phase	1=in warm-up
5	TPS Acceleration Enrichment	0=not in TPS acceleration mode	1=TPS acceleration mode
6	TPS Deceleration Enrichment	0=not in deceleration mode	1=in deceleration mode

Engine Bit Examples

Bit	Binary	Means:
1	000001	"running"
2	000010	"cranking"
5	000101	"running and start-up enrichment"
9	001001	"running and warm-up enrichment"
13	001101	"running and start-up enrichment and warm-up enrichment"
17	010001	"running and TPS acceleration enrichment"
25	011001	"running and warm-up and TPS acceleration enrichment"
33	100001	"running and TPS deceleration enrichment"

to control the closing of your injectors. It should not be too hot to touch with your fingertip. If it is too hot, the value of this resistor can be increased, or the zener D21 can be replaced with a lower breakdown value.

Tuning Issues

If you have a very long duration cam in your motor, and It idles poorly, you may be able to get it to idle better. Often a rough idle may be caused by lean air/fuel ratios. This is really is more of a cam issue than a fuelling issue. The exhaust valve is held open later into the intake stroke and the intake opens earlier near the end of the exhaust stroke. At low speeds and relatively high intake vacuums you get more exhaust contamination of the fresh air/fuel charge. As you get more contamination of the air/fuel charge you typically need a richer mixture to get it to ignite and burn properly.

This means you probably cannot run a stoichiometric [chemically correct] mixture of 14.7:1 with your long duration cam. You need to run richer. So you tune your idle by ' ear' rather than with a narrow band EGO [oxygen] sensor. And make sure you are not allowing EGO correction at idle if you have a rowdy cam! It will be trying to “correct” your mixture back to a lousy idle. If your engine will not idle well at stoichiometric mixtures, set the EGO Active Above RPM to a few hundred RPM above your idle speed. This will ensure that MegaSquirt does not try to lean the mixture back to stoichiometric to compensate for your adjustments.

Another tip you can try if you have a large overlap cam is to pinch off the MAP hose slightly while the engine is idling, and see if the idle quality improves. If so, then try a restriction in the MAP vacuum line. This has the effect of damping the vacuum pulsation the MAP sensor sees. You will have to experiment with restrictor sizes to see what works for your system.

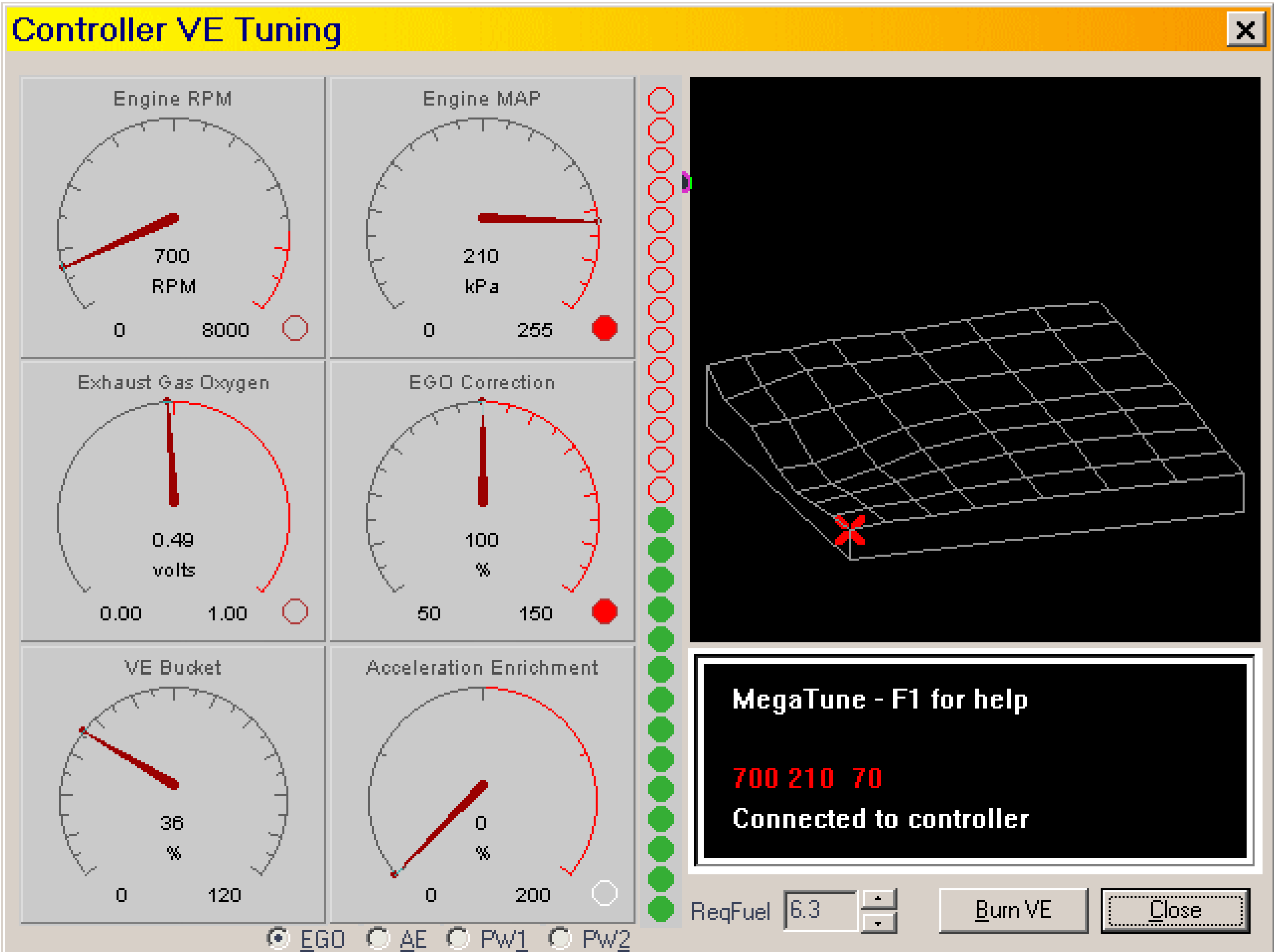
A few more things to try:

1) Check your VE table entries right around the idle point - if the RPMs or MAP fluctuate, then you can get rolling idles, etc. You may have to move some of the bins around to bracket the idle RPM/MAP region, and keep flat VE values within this.

2) If you run low-impedance injectors, you need to tune your PWM current limit. Start with 75% PWM and 1.0 millisecond time threshold. Once you get idling, then first adjust the PWM duty cycle down until you notice a change in idle quality, then move the value back up 3 - 5%. Do the same with the time threshold. On the car it is very easy to do and only takes a few minutes. And, at idle, the overall injector pulsewidths are small compared to their close time, so this will allow you to adjust the values. In other words, adjust the PWM current limit before taking the car out on the street where injector pulsewidths become high.

Your engine will idle at a certain vacuum. It might help on a street use motor on the VE map to use a lower point for starting MAP than idle vacuum. For example you can have yours set at 20 even though you idle at 27 or so. This allows you to run less fuel on overrun deceleration and coast events (not just for a second like the TPS will do). This allows you a saving of 3-4 MPG on average driving and you might be able to run more advanced timing under this vacuum.

On the other hand, you may want to do the opposite. You can increase the VE values just to the left and above idle. You can make them really rich [say double the idle VE value, to keep the car from stalling. This seems to work really well, if the engine starts to stumble, the PW



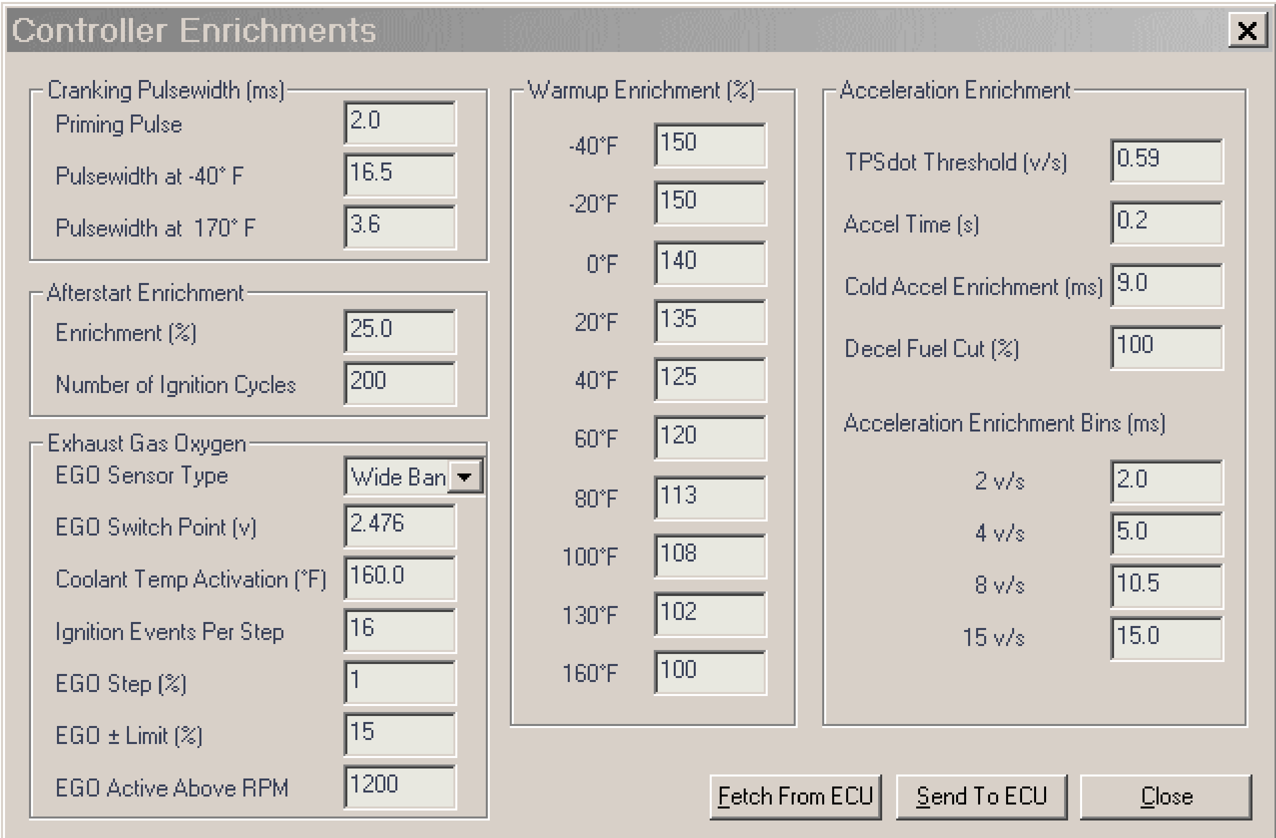
goes up and it recovers.

By working with the RPM and MAP bins, you should be able to work out a set of values that lets you run lean at cruise and decel [where the RPM is above idle, and MAP below idle], but rich when stalling [RPM below idle].

On the warm up enrichments screen, the warm up enrichment only goes to 160F. The 160 degree bin value of the enrichment (which should ideally be 100%) is used at all temperatures above 160 degrees F.

The system compensates automatically for any amount of idle solenoid bypass air because of the effect it has on the MAP value [i.e. the vacuum in the manifold is lowered by the bypass air, this is sensed by the Manifold Absolute Pressure sensor, and the processor decides to inject more fuel. The effect is exactly the same as if you had cracked open the throttle a bit. The fuel goes around the throttle plates, which are never truly closed. They are set at the opening required for the slowest throttle speed desired for the engine under optimal conditions, which leave plenty of room for the fuel to get by. The fast idle air then adds to this baseline amount of air to raise the idle speed.

In some circumstances, you may want to run without oxygen sensor feedback, called “open loop”. The best way of forcing MegaSquirt to run open loop is to change the O2 sensor step



to 0 [zero] on the enrichments screen. It will still log the O2 voltage, but not do anything about it.

With the version 2.00 code the MegaSquirt Fuel Pump output is programmed with a priming pulse option to shut off the unit in case of an engine stall, etc. It turns on the pump immediately when the power is applied and shuts it off 2 seconds later if the engine is not running. If you set the width of priming pulse in MegaSquirt to zero, then the system defaults to not turning on the pump until the first tach pulse. If this field is non-zero, then when the key is turned on, the injectors will fire once with a duration specified by the priming pulse field, and the pump is also activated, and will stay on for two seconds if there is no tach activity, or for as long as there tach activity plus two seconds. The version 1 software only runs the fuel pump when it sees activity on the tach input. So, if you just turn on the key and do not crank the engine, the pump is off. When you start cranking, the pump comes on, and times out two seconds after the last pulse on the tach signal.

On start-up MegaSquirt records the ambient barometric pressure. The barometer correction multiplier to VE increases as pressure decreases. If it the ambient baro pressure is low (high altitude) the algorithm adds fuel. This is mostly because at a given MAP, the engine will flow more air with less exhaust back pressure and therefore needs more fuel at higher altitudes. Once running the MAP sensor determines fuel based on you VE table entries which are then scaled by the baro correction recorded at start-up. The correction values used by MegaSquirt came from a code disassembly of a 1990 Corvette ECU.

If your MegaTune displays bizarre values for the barometer on the runtime display, 76 kpa, for example, you may be resetting while running. MegaTune has a check that detects most resets by watching the seconds value. If the seconds goes to zero from any value other than 255, then it signals a reset with an audible “beep” and sets a counter visible on the lower right corner of the screen, where it normally says **"connected"**

As well, you can check the datalog seconds count - make sure it counts up to 255, then rolls over to zero and continues again and again. If you get shorter counts (like say 56 then a rollover) then the processor is resetting. Note that most of the time on the car you will not notice that the reset has occurred because it happens so fast.

What happens when the engine has a running reset (when the engine is running and the processor resets), then it grabs the barometer near the beginning of the MegaSquirt processor boot procedure. If the engine is running, then it will grab engine vacuum and use this for barometer.

For normal operation the processor comes up so fast that it has grabbed the MAP value before the engine has a chance to start cranking, much less running.

You need Windows 95 and a conventional serial port to communicate with MegaSquirt. USB will NOT work, however some people have reported that they have been successful using a USB-serial adapter. Just about any computer that' s capable of booting Windows 95 (or better) will be fast enough, but get the fastest machine you think is reasonably priced.

Other Tuning Software and Platforms

MS-Palm is available from the files section of the MegaSquirt yahoo! site. It has worked well for some. MS-Palm uses the **HOTPaw basic**, and does data logging. It is limited to about 60 datalog lines or about 15 seconds or so at 4 Hz. It writes the data out as Memopad entries, which are limited in size to 4k. The source code is there for any and all to hack away. It would be easy to cut back on the variable list, to get a log time up over a minute if someone so desired. MS-Palm does allow editing of VE table, enrichment bins, etc. as well. It worked fine with V2 code, but has not been tested yet with the new DT code. Or you can use MSMiniTune, which requires the **NSBasic** files from the files section, but it may not be fully functional.

You might be able to use a Mac to tune MegaSquirt. Blake Qualley has successfully ran PC Configurator and MegaTune hooked up to the MegaSquirt board with the Stimulator, on a Mac using VirtualPC and Win98. Everything worked great. He uses a Mini DIN 8 to DB9 cable, selects the Mac serial port as COM1, and "shared" the Mac hard drive with Win98 as volume (F). His set-up includes an Airport wireless network (802.11b), sharing the DSL connection, setting up VirtualPC to utilize various resources (e.g. Mac serial/printer port as COM1, Ethernet, faking the video and sound cards, etc). MegaTune runs great and there is no lag between turning a MegaStim pot and seeing the results on the screen. Blake says he would definitely recommend VirtualPC to other Mac users over purchasing a PC just to run MegaTune and other MegaSquirt related apps.

Other Hardware for Your MegaSquirt

MegaStimulator

The Stimulator is a small board which plugs into the connector of the MegaSquirt. It simulates all the sensor the inputs the MegaSquirt would normally see and provides power to the MegaSquirt. The Stimulator also allows you to monitor the MegaSquirt injection pulses [actual], fuel pump relay operation, and fast idle solenoid output with four LEDs. On the MegaSquirt, there are three LEDs which allow you to monitor the remaining functions (injection pulse [commanded], warm-up enrichment, and acceleration enrichment).

It allows you to:

- verify that the unit is working prior to installation in the car,
- become familiar with the software prior to tuning.
- determine if software changes are having their anticipated effect before attempting to run them on an engine, and finally,
- diagnose problems that may crop up after installation, because you will be able to verify if the computer is (or is not) working as intended.

There are 4 LEDs on the Stimulator, and 3 on the MegaSquirt. On the Stimulator, the 4 LEDs are:

- **Injector #1** - lights when the first injector bank is grounded [firing].
- **Injector #2** - lights when the second injector bank is grounded [firing].
- **Fuel Pump** - lights when the fuel pump relay is grounded.
- **FIdle** - lights when the Fast Idle solenoid is activated.

On the MegaSquirt, there are three LEDs, which are:

- **Injector** - lights when the **either** injector bank is instructed to be grounded [firing].
- **Warm-Up** - lights when Warm-up Enrichment (WE) is activated.
- **Acceleration** - lights when Acceleration Enrichment (AE) is activated.

Note that these all do separate things.

The injector LED on the MegaSquirt lights when either Injector bank is **commanded** to fire, while the injector LEDs on the Stimulator light when each injector bank actually **is** grounded. The megasquirt LED will flash synchronously with the Stimulator Injector LEDs in simultaneous mode, and will flash twice as fast as either LED in alternating mode.

The Warm-up Enrichment values are separate entries in the software from the Fast Idle Threshold value, so these two LEDs will generally light at different, though similar, coolant temperatures.

The Fuel Pump LED is light whenever the fuel pump relay is grounded. Since the Stimulator puts out about 1 pulse per second minimum, and the MegaSquirt leaves the pump on for 2 seconds after the last ignition event, the fuel pump LED should be light whenever the Stimulator is plugged into the MegaSquirt.

By using these LEDs, adjusting the RPM, coolant and air temp., EGO, and throttle potentiometers on the Stimulator, and viewing the PCC Configuration or MegaTune tuning software, you ought to be able to test every function of your MegaSquirt, except MAP function [for which you can suck and blow...] at any time.

If you have assembled your Stimulator, and everything seems to be in the right place, but it doesn' t work properly, check the 9V battery for power. The 9 volt battery may not supply enough current for some or all of MegaSquirt' s functions. Even a fresh battery may not work.

You can wire the Stimulator to a 9 to 12 volt wall mount 110 volt AC adapter. Cut the plug off the end, check the polarity of the resulting wires, and connect them appropriately to the Stimulator. You can then run the Stimulator/MegaSquirt combination for as long as you want without the complications of dying batteries!

The Stimulator will work with the Wing diode installed and the Flyback modifications - the John zener is the deal breaker. If you have D8 installed, jumper it to get a low voltage tach signal to work. Works with a 5v tach signal.

It is normal that the fuel pump led on the Stimulator board is always on, even when RPM is at 0. This is because you cannot turn off the tach pulses on the Stimulator board. Even when the RPMs read zero while hooked on the Stimulator board, the Stimulator board is putting out about one pulse/sec, enough to keep the fuel pump alive.

When you are tuning you will likely want to hit some specific RPMs for some reason. With RPMC you notice you can go from not reading any RPM to something in the neighborhood of 18000 RPM. (Not may engines will do that but the MegaStimulator can do some pretty nifty things!). You do all this in 270 degrees or so of rotation. Next RPMF will allow you to work up and down around a specific RPM a little more easily. In use start by setting RPMF in the middle of its range. Then get near the desired RPM with RPMC the Fine Tune to exactly your desired RPM with RPMF.

T*o* simulate pressure [vacuum] while testing your MegaSquirt, you can use a syringe. A short length of polyurethane tubing and a 20ml syringe works just right as the final component of a stimulator. This set-up can pull decent vacuum and generate up to 100kpa of turbo boost. Or you can alternately suck and blow.

Relay Board

The relay kit provides is a central place for all of the required relays, fuse protection, and external wiring. It was developed in a response to a few burned boards due to mis-wiring. MegaSquirt gets its power form the car' s 12 battery, and distributes it through appropriate fuses and relays to the various actuators (injectors, fuel pump, fast isle solenoid, etc.), as well as providing a central wring terminal for all of the other MegaSquirt components, such as sensors.

The relay board is not absolutely necessary, but makes it easier to hook up the wiring to MegaSquirt and makes it less likely that you will fry something in MegaSquirt through incorrect wiring. Whether you need the relay kit depends on your ability and what you are comfortable with. The MegaSquirt connector can be tricky to wire, but if you are careful then there will be no problems. With the relay kit, you still have to run a cable from the relay box underhood to the MegaSquirt unit (not underhood, as recently reported), but then you have a nice terminal block to run all of the engine wiring.

You cannot install the Relay board lid with the relays installed. The relay board is designed to be open-top. This is not a problem under the hood or inside the passenger compartment. We only included the top of the case in the relay kit because it was easier to ship (not having to take apart all of the cases). Now, if you want, you could potentially cut out the top lid for the relays, an extra-curricular activity.

Also, you may want to spray on conformal coating after the PCB is soldered up - be sure to cover up the sockets with tape before spraying. And you should drill a very tiny hole inside of the top and bottom flange of the case to allow moisture to escape if mounted underhood.

There is complete details on how to make up the DB-37 cable between the MS box and the relay kit on the Power/Relay board page of MegaSquirt website. This shows you (with pictures) how to make the cable with individual wires made into a bundle.

There is one DB37 hood supplied with each ECU kit but none with the Digi-Key relay board parts. The hood and female connector is also not mentioned on the power relay kit BOM list. We only included one hood, mainly because the ones we could get in quantity were plastic and would not survive underhood temperatures. Also, many people are soldering the wiring directly on the relay board, which means that they do not need the connector at all. If you are going the connector-on-relay board route, try get a metal DB-37 hood locally.

Willette Programmer

The Willette programmer is a standalone programmer for the MC68HC908GP processor. The power is obtained from a PC serial port. The board is 2.25 inch square has a ZIF socket for the processor and a ICP header for target board programming. The complete kit was $30US. The programmer group buy is over but spares are being shipped on a 1st pay basis.

The kit is complete except for the 40 pin socket. In order to program blank chips for the MegaSquirt you would need to purchase a 40 pin Dip socket. Partial kit price is $25US and includes shipping to North American destinations.

More info is at the site: http://groups.yahoo.com/group/68hc908pgm. **The second group buy still has spare units left over, but there are no plans for a third group buy. So get yours now if you think you might want one.**

Note that MegaSquirt partial kits come with the processor pre-programmed. You do not need the Willette program to run MegaSquirt or perform standard upgrades on these processors. However, if you purchase a blank chip from any other supplier than the group buy, or if you want to extensively rework the MegaSquirt code, you **need** this programmer. See the Yahoo 68hc908 group for more info.

The steps to program a new processor documented are listed below:

Make sure you have v 1.20 of the Prog08sz software, newer versions have some configuration problems. Get the good version from at http://groups.yahoo.com/group/68hc908pgm/files/software/prog08sz.zip.

1. For MegaSquirt code (not tomtek ignition), assemble both the **boot_r12.asm** and **megasquirt.asm** into their respective ".s19" files. See how to do this in the ' CODE' section of the MegaSquirt FAQ. (Tom' s ignition code and dual table MS code already contain the boot loader, so you only get one file.)

2. Start up **prog08sz**, you might have to retry several times to connect. If you can' t get a connection, read the 908 archives for some hints on debugging.

3. Often the connect problem is low voltage due to a wimpy serial port, so Jim says eliminate this problem by connecting an external power source. (e.g., a 9 volt battery) to the circuit. Connect the + to the banded side of D2 (or D1) and the - to a ground point (e.g., the mounting lug on the 9 pin connector).

4. Select the 908_gp32.08p module when asked to "Specify Programming Algorithm to Use!".

5. If this chip has been programmed before, you need to erase it (if in doubt, do this anyhow). Click on the button with the "pencil erasing" icon (6th from the left) to Erase Module. When this is done, it says "Erasing. Module has been erased" in the lower left corner of the screen, after which you should verify the chip with Blank Check Module (the button beside Erase Module). You should get a message saying "Erased."

6. Click on "SS Specify S Record" in the window on the middle left [or the ' diskette' button in the menu], and select "**boot_r12.s19**".

Click on "Program Module" [on the menu on the left, or use the button beside the ' diskette' /Specify Record button]. After a few seconds, it should complete.

Then you can then click on "VM Verify Module" [on the menu on the left, or use the button beside the "Program Module" button]. You should get a message that the module was ' verified' .

7. Click on Specify S Record as above, this time selecting "**megasquirt.s19**" Follow this with "Program Module", then "Verify Module".

Note that you can consolidated steps 6 and 7 by including the boot_r12 code when assembling megasquirt.asm, just uncomment the last line in megasquirt.asm before you assemble it.

8. Disconnect the serial cable from the programmer, unlatch the ZIF lever, remove the chip, and you should be ready to insert the chip into a MegaSquirt controller and run.

MegaJolt

MegaJolt *Light* is a Bowling and Grippo developed ignition controller. It works with a Ford EDIS module and a 36-1 crank sensor. You can run the MegaJolt *Light* by itself (you will need a MAP and coolant sensor) or housed with MegaSquirt (which can share sensors). Find more information about MegaJolt at: http://www.bgsoflex.com/mjl/mjl_edis.html

The EDIS module provides the timing critical functionality to the coil pack. The processor takes temperature, MAP and TPS to make timing adjustments. Ford EDIS comes in 4, 6 and 8 cylinder varieties. No use to the Audi 5 pot. It is not compatible with an odd fire V6. This an intermediate step towards the full MegaJolt ION sensing, still under development, and is offered to provide an interim (though full featured) ignition solution for those Megasquirters who are looking for a ignition. MegaJolt *Light* requires the 36-1 toothed wheel used by the Ford. Bruce and Al are actively testing this at present. To be released to the rest of us soon.

MegaJolt ION sensing ignition is going to be a universal full specification igntion comtroller. Using ionisation currents in the burnt fuel to make fine timing adjustments based upon peak pressure point. This is still some way off being completed, at least several months.

You are going to need 1 EDIS module that can handle the appropriate number of cylinders (EDIS-4, EDIS-6 or EDIS-8) and a 36 - 1 wheel. Both of these will be mandatory with MegaJolt *Light*, which can run the EDIS module from 0 degrees to 59 degrees of advance. Eventually Bruce and Al are planning to develop a generic ignition module which will handle any type wheel and have coil on plug, but this will be a while off.

More information on the Ford EDIS modules can be found at:

http://www.alternativeauto.com/waterbox/wb-archives/edis.html

For a schematic for MegaJolt look at www.bgsoflex.com/mjl/mjl_edis.html. The MegaJolt *Light* board has knock-sensor input and a full stepper IAC motor driver on-board. When the circuit is fully tested Bruce and Al will put up the schematics on this site.

MegaJolt Light can be housed in the same box as MegaSquirt. For the full-up MegaJolt, it will require its own separate box.

Embedded Software Upgrade Instructions

Hyperterminal

Here are instructions for using Hyperterm.exe for updating the embedded software (all Windows installations has the Hyperterminal application - use a "find-file" function to search for "hypertrm.exe"):

1) Put the **bootloader jumper pin** in, hook up a serial cable from the computer to the MegaSquirt box, and fire up Hyperterminal (Hypertrm.exe).

> Note: you do not use the PC Configurator to perform the firmware upload - this is done with the terminal program Hyperterm (or similar, see below), and do not have PC Configurator, MegaTune, or any other serial data application running on your PC.

At this point, leave the MegaSquirt unit un-powered.

2) You will see a Hyperterminal screen when you execute - type in any name you want in the box, and make sure the Red telephone with the little yellow telephone next to it is highlighted.

3) A new window appears. On the "Connect Using" selection, select either COM1 or COM2, depending on your serial connection.

4) Then another window with comm port settings pops up. Select 9600 baud, 8 data bits, no parity, 1 stop bit, and set flow control to "None" - this is very important. Hyperterm terminal is now up.

5) Apply power to the MegaSquirt board, then hit <enter>. You should see the "Boot>" prompt appear in the terminal screen.

6) Type "H" and you will be shown the options available in bootloader mode.

7) Hit "W" for "Wipe" - this erases the entire flash array (except the section running the bootloader).

8) Hit "P" for "Program," which will respond with the text "wait ...".

9) Then under the "Transfer" menu, select "Send Text File..." (do not use the "Send file" mode - you are sending a text file).

10) Set "Files of type" to "All files" and select the .s19 file containing your assembled code (megasquirt.s19). You will see no activity on the screen for about 40 seconds.

11) Then the prompt will come back, which means the operation is finished.

12) Turn off power to MegaSquirt and shut down Hyperterminal.

13) Remove the jumper, and you are ready to go.

Download.exe

OR if you' d rather use something simpler than Hyperterminal,

1) Grab the latest megasquirt.s19 file from the www.bgsoflex.com website.

2) Get http://groups.yahoo.com/group/megasquirt/files/MegaTune/tools.zip. This will contain a file called "download.exe". Extract download.exe wherever you saved the .s19 file.

3) Start up an MS-DOS command window [i.e. click the Start button, chose Run, then type in **command** and press ' enter'], change the current directory (cd\) to the directory containing download.exe and the .s19 file.

4) Short out R6, boot up the MegaSquirt and type: **download megasquirt.s19**

5) Watch it report status as the download proceeds. It will automatically reboot MegaSquirt, so your LEDs will start flashing on the MegaStim, just like usual.

6) Pull the jumper off R6 and start tuning.

EasyTherm

OR simpler yet,

Use EasyTherm. It includes the most recent embedded software versions for MegaSquirt. The brief instructions are in the EasyTherm help file.

MegaSquirt Glossary

MT - *MegaTune*, a Windows-base configurator program for the MegaSquirt EFI controller.

MS - *MegaSquirt*, used in this document to refer to the MegaSquirt fuel injection controller or it' s embedded software.

AFR - *Air Fuel Ratio*, the mass ratio of air to fuel in the combustion chamber. See NB- and WB-EGO sensors, below.

CTS - *Coolant Temperature Sensor*. Usually the CTS is an NTC (Negative Temperature Coefficient) thermistor, or a resistor whose resistance varies with temperature (NTC means the resistance goes down as the temperature goes up.

Duty Cycle – A number indicating the amount of time that some signal is at full power. In the context of MegaSquirt, duty cycle is used to describe the amount of time that the injectors are on, and to describe the "hold" part of the peak and hold injector drivers (see Low Impedance Injectors, below).

EGO Sensor - *Exhaust Gas Oxygen sensor*, used to describe the sensor in the exhaust that measures the lean/rich state of the AFR. Used to control the via a feedback algorithm called 'closed loop'.

FIdle - *Fast Idle*. A device used to control idle speed with additional air supplied by a vacuum solenoid. MegaSquirt has a simple on-off fast idle control, and does not have the ability to drive a PWM IAC (Idle Air Control) device.

High Impedance Injectors - Fuel injectors designed to work with a simple switch in a 12 volt circuit, no special signal conditioning is required to drive them. The resistance of a high impedance injector is about 10-15 ohms.

IAC – *Idle Air Control*. Typically a"stepper motor". MegaSquirt does not have the circuitry to control these.

IAT sensor - *Intake Air Temperature sensor*, same as MAT, see below.

Low Impedance Injectors - Fuel injectors that are designed to run at a much lower current than would be supplied by a direct 12 volt connection. They require a special signal that is initially at full current (4-6 amps, a.k.a. "peak current") for about 1.0-1.5 ms, but then drops down to about 1 amp ("hold current") for the rest of the opening pulse. The resistance of a low-impedance injector is typically 1-3 ohms.

MAP sensor - *Manifold Absolute Pressure sensor*. Measure the absolute pressure in the intake manifold (related to the engine vacuum), to determine the load on the engine and the consequent fuelling requirements. The standard MAP sensor in MegaSquirt$^{2.0}$ is the MPX4250 (2.50 BAR or 21 psig).

MAT Sensor - *Manifold Air Temperature sensor*, the same as IAT. The MAT circuit is identical to the CTS circuit, see CTS, above.

NB-EGO Sensor - *Narrow Band EGO sensor*, gives a switch at the stoichiometric ratio (the chemically correct mixture of air and fuel), but unreliable for AFR other than stoichiometric.

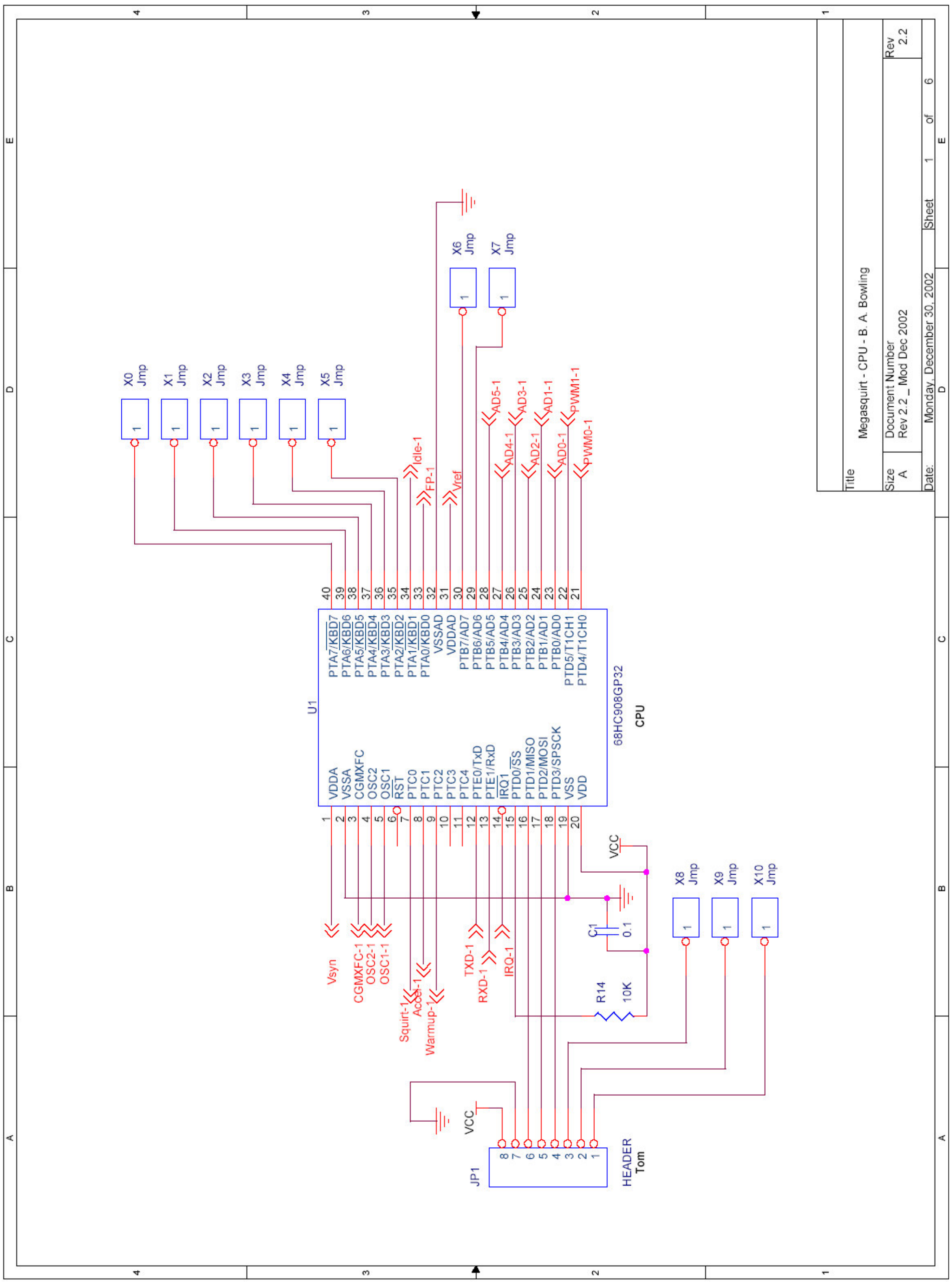
P&H Injectors - *Peak and hold injectors*; see Low Impedance injectors.

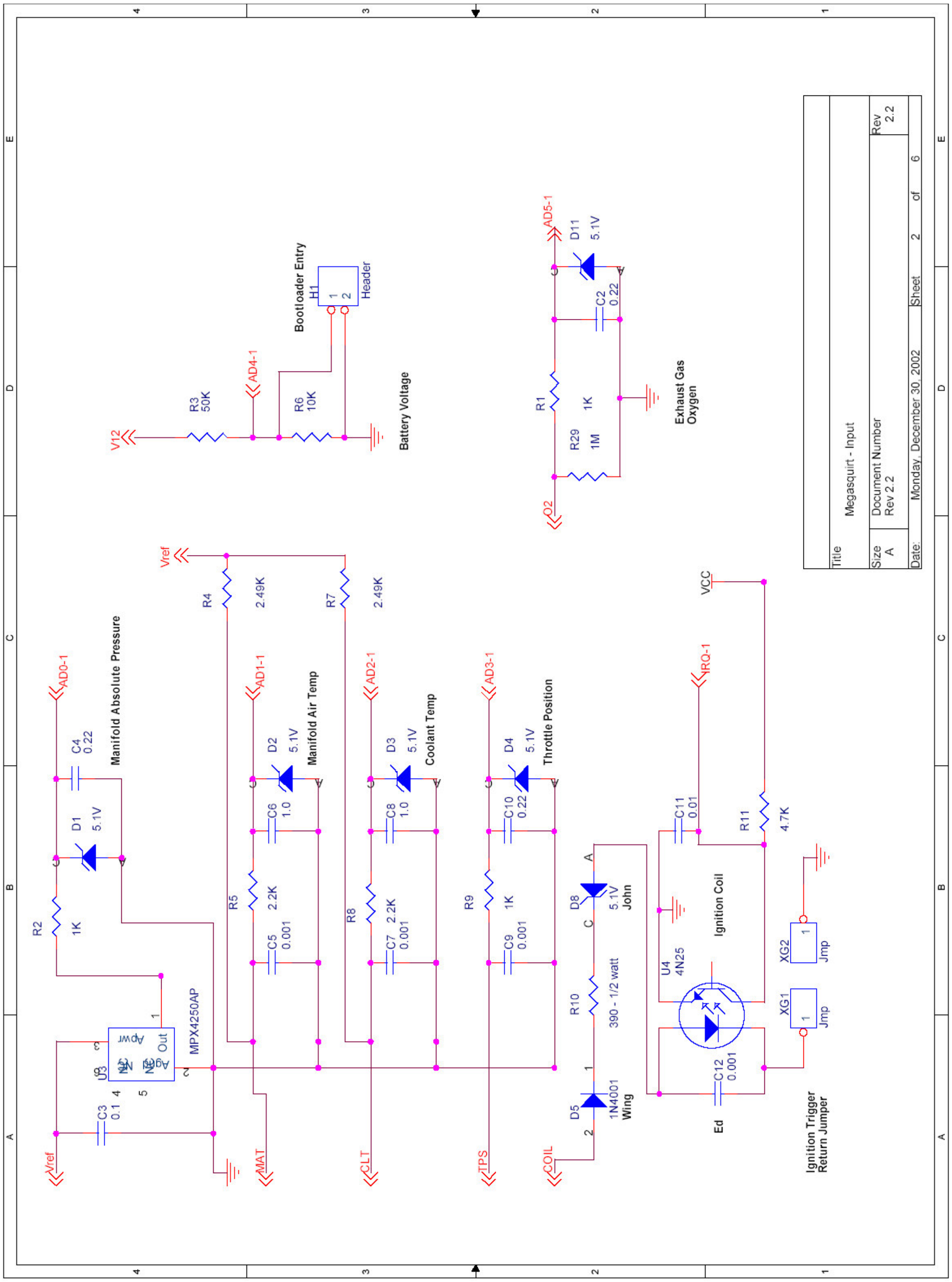
Pulse Width Modulation (*PWM*) - A signal with a fixed pulse width (frequency), which is turned on for part of the pulse. The percent of time that the signal is on is called its duty cycle. PWM is used to control voltage (and consequently current) to fuel injectors.

TPS - *Throttle Position Sensor,* a voltage divider that gives information to MS about throttle opening, from which it computes rate of throttle opening for acceleration enrichment.

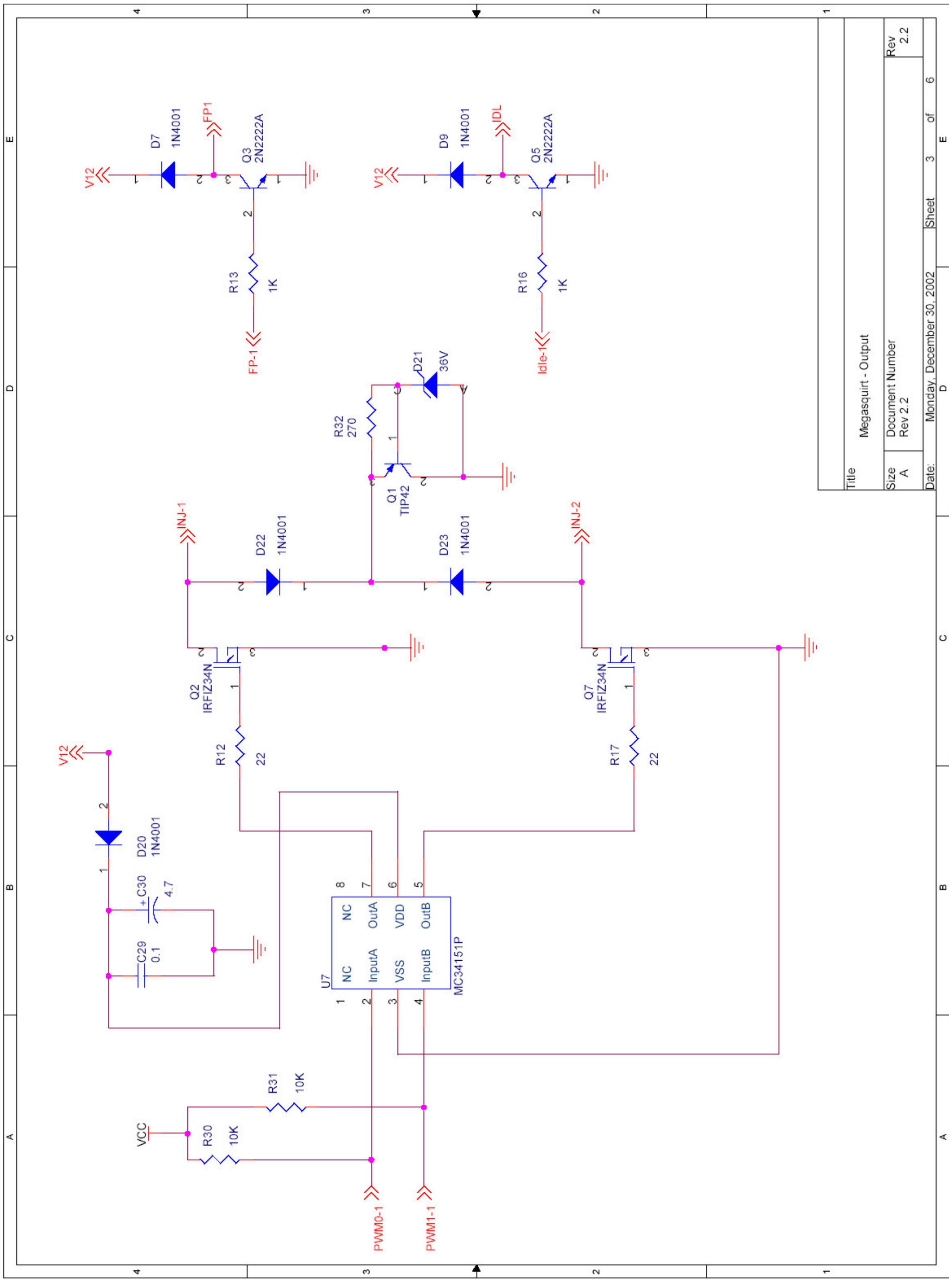
VE - *Volumetric Efficiency*. The actual amount of air being pumped by the engine as compared to its theoretical maximum. A 200 cubic inch motor will theoretically move 200 cubic inches of air in one cycle at 100% efficiency. If the engine is actually running at 75% VE, then it will move 150 cubic inches of air on each cycle.

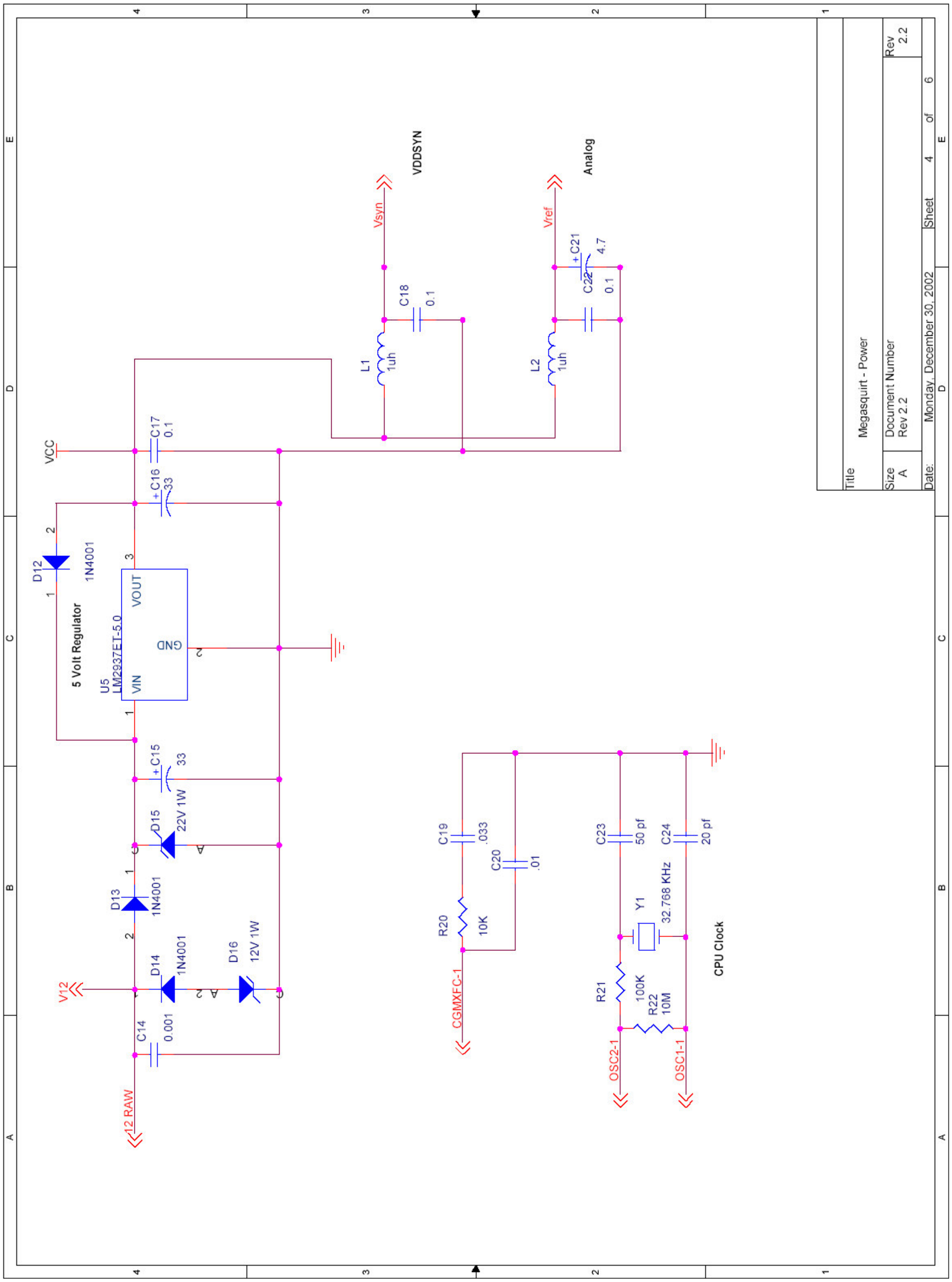
WB-EGO Sensor - *Wide Band EGO sensor*, can be used to derive real AFR data with mixtures from 10:1 to 20:1, i.e. anything you are likely to be interested in.

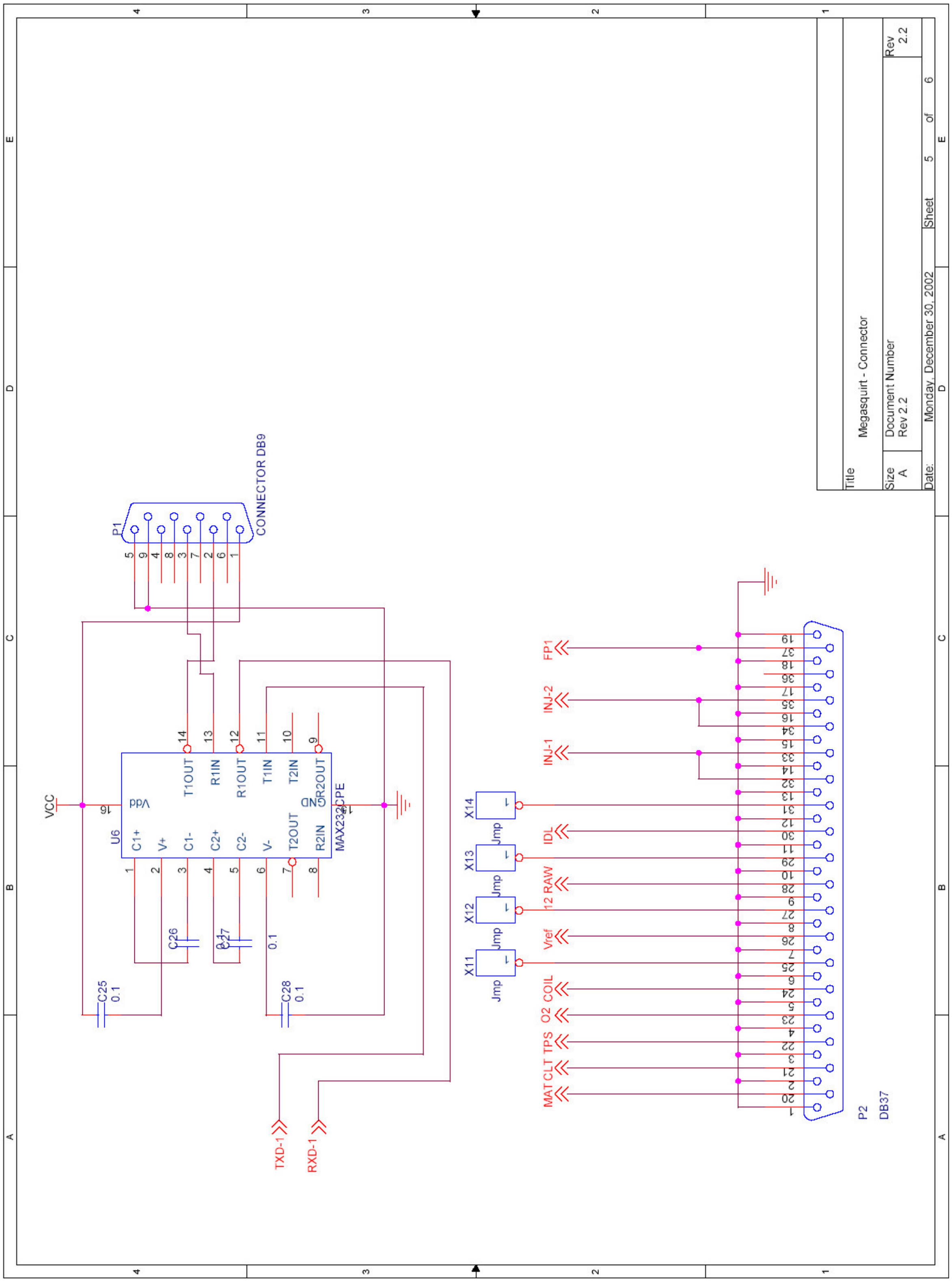

U1
68HC908GP32
CPU
VDDA
VSSA
CGMXFC
OSC2
OSC1
RST
PTC0
PTC1
PTC2
PTC3
PTC4
PTE0/TxD
PTE1/RxD
IRQ1
PTD0/SS
PTD1/MISO
PTD2/MOSI
PTD3/SPSCK
VSS
VDD
PTA7/KBD7
PTA6/KBD6
PTA5/KBD5
PTA4/KBD4
PTA3/KBD3
PTA2/KBD2
PTA1/KBD1
PTA0/KBD0
VSSAD
VDDAD
PTB7/AD7
PTB6/AD6
PTB5/AD5
PTB4/AD4
PTB3/AD3
PTB2/AD2
PTB1/AD1
PTB0/AD0
PTD5/T1CH1
PTD4/T1CH0
X0 Jmp
X1 Jmp
X2 Jmp
X3 Jmp
X4 Jmp
X5 Jmp
X6 Jmp
X7 Jmp
X8 Jmp
X9 Jmp
X10 Jmp
Idle-1
FP-1
Vref
AD5-1
AD4-1
AD3-1
AD2-1
AD1-1
AD0-1
PWM1-1
PWM0-1
Vsyn
CGMXFC-1
OSC2-1
OSC1-1
Squirt-1
Accel-1
Warmup-1
TXD-1
RXD-1
IRQ-1
C1 0.1
R14 10K
VCC
JP1
HEADER
Tom
Title
Megasquirt - CPU - B. A. Bowling
Size A
Document Number
Rev 2.2 _ Mod Dec 2002
Rev 2.2
Date: Monday, December 30, 2002
Sheet 1 of 6

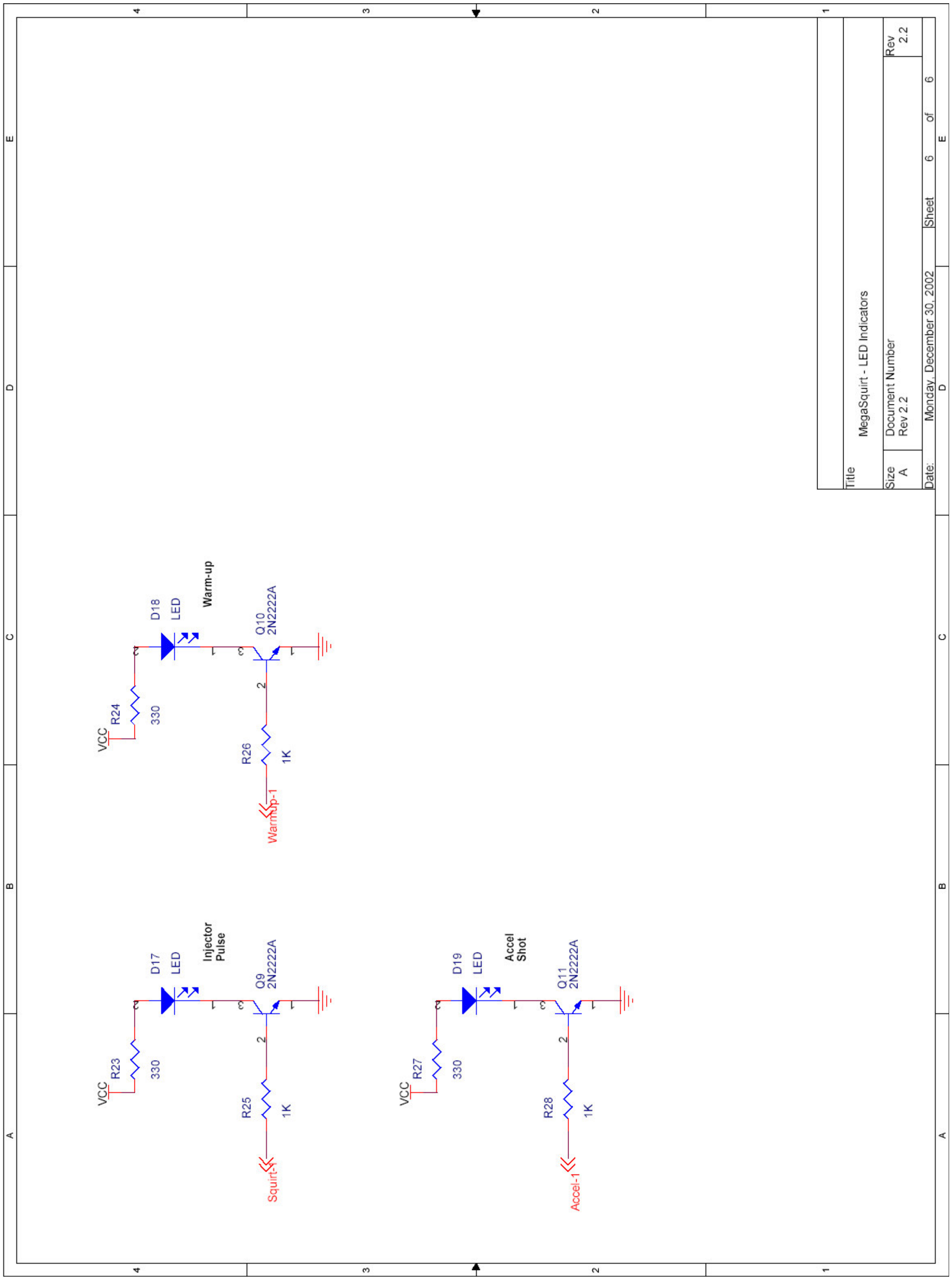


Manifold Absolute Pressure
Manifold Air Temp
Coolant Temp
Throttle Position
Battery Voltage
Bootloader Entry
Exhaust Gas
Oxygen
Ignition Coil
Ignition Trigger
Return Jumper
MPX4250AP
Title
Megasquirt - Input
Size
A
Document Number
Rev 2.2
Rev
2.2
Date:
Monday, December 30, 2002
Sheet
2
of
6


V12
D7
1N4001
FP1
Q3
2N2222A
R13
1K
FP-1
V12
D9
1N4001
IDL
Q5
2N2222A
R16
1K
Idle-1
R32
270
D21
36V
Q1
TIP42
INJ-1
D22
1N4001
D23
1N4001
INJ-2
Q2
IRFIZ34N
R12
22
Q7
IRFIZ34N
R17
22
V12
D20
1N4001
C30
4.7
C29
0.1
U7
NC
InputA
VSS
InputB
NC
OutA
VDD
OutB
MC34151P
VCC
R30
10K
R31
10K
PWM0-1
PWM1-1
Title
Megasquirt - Output
Size
A
Document Number
Rev 2.2
Rev
2.2
Date:
Monday, December 30, 2002
Sheet
3
of
6


VDDSYN
Analog
Vsyn
Vref
C18
0.1
C21
4.7
C22
0.1
L1
1uh
L2
1uh
VCC
C17
0.1
C16
33
D12
1N4001
5 Volt Regulator
U5
LM2937ET-5.0
VIN
GND
VOUT
C15
33
D15
22V 1W
D13
1N4001
D14
1N4001
D16
12V 1W
V12
C14
0.001
12 RAW
C19
.033
C20
.01
R20
10K
CGMXFC-1
C23
50 pf
C24
20 pf
Y1
32.768 KHz
R21
100K
R22
10M
OSC2-1
OSC1-1
CPU Clock
Title
Megasquirt - Power
Size
A
Document Number
Rev 2.2
Rev
2.2
Date:
Monday, December 30, 2002
Sheet
4
of
6


Title
Megasquirt - Connector
Size
A
Document Number
Rev 2.2
Rev
2.2
Date:
Monday, December 30, 2002
Sheet
5
of
6
P1
CONNECTOR DB9
U6
MAX232CPE
VCC
C25
C26
C27
C28
0.1
TXD-1
RXD-1
X11
X12
X13
X14
Jmp
FP1
INJ-2
INJ-1
IDL
12 RAW
Vref
COIL
O2
TPS
CLT
MAT
P2
DB37


VCC
R23
330
D17
LED
Injector Pulse
Q9
2N2222A
R25
1K
Squirt-1
VCC
R24
330
D18
LED
Warm-up
Q10
2N2222A
R26
1K
Warm-up-1
VCC
R27
330
D19
LED
Accel Shot
Q11
2N2222A
R28
1K
Accel-1
Title
MegaSquirt - LED Indicators
Size
A
Document Number
Rev 2.2
Rev
2.2
Date: Monday, December 30, 2002
Sheet 6 of 6

MegaSquirt 68HC908GP32 Memory Map

$0000 - $003F = I/O Registers: 64 Bytes
$0040 - $023F = RAM 512
$0240 - $7FFF = Unimplemented 32,192 bytes
$8000 - $FDFF = FLASH Memory: 32,256 bytes
$FE00 = SIM Break Status Register (SBSR)
$FE01 = SIM Reset Status Register (SRSR)
$FE02 = Reserved (SUBAR)
$FE03 = SIM Break Flag Control Register (SBFCR)
$FE04 = Interrupt Status Register 1 (INT1)
$FE05 = Interrupt Status Register 2 (INT2)
$FE06 = Interrupt Status Register 3 (INT3)
$FE07 = Reserved (FLTCR)
$FE08 = FLASH Control Register
$FE09 = Break Address Register High (BRKH)
$FE0A = Break Address Register Low (BRKL)
$FE0B = Break Status And Control Register (BRKSCR)
$FE0C = LVI Status Register (LVISR)
$FE0D - $FE0F = Unimplemented: 3 bytes
$FE10 - $FE1F = Unimplemented: 16 bytes Note: Reserved for compatibility with monitor code for A-Family parts
$FE20 - $FF52 = Monitor ROM: 307 bytes
$FF53 - $FF7D = Unimplemented: 43 bytes
$FF7E = Flash Block Protect Register (FLBPR)
$FF7F - $FFDB = Unimplemented: 93 bytes
$FFDC - $FFFF = Flash Vectors: 36 bytes

Some MegaSquirt Assembly Language Variables

ACMULT = Acceleration cold multiplication factor (percent/100)
adsel = ADC Selector Variable
aircor = Air density correction is computed from MAT.
asecount = Counter value for after-start enrichment counter - every ignition
AWC = After-start number of cycles
AWEV = After-start Warmup Percent enrichment add-on value
baro = The barometric pressure as measured by MegaSquirt.
barocor = Barometer Lookup Correction - percent, based on the initial MAP sensor reading.
batt = Battery Voltage ADC Raw Reading - counts
BATTFAC = Battery Gamma Factor
clt = Coolant Temperature ADC Raw Reading - counts (0 - 255)
coolant = Coolant temperature in Degrees F plus 40 (allows -40 degress to fit in integer)
CWH = Crank Enrichment at 170 F
CWU = Crank Enrichment at -40 F
ddra = Port A Data Direction Register
ego = Exhaust Gas Oxygen ADC Raw Reading - counts
egocorr = This is the correction factor computed from O2 sensor readings.
egocount = Counter value for EGO step - incremented every ignition pulse
egotemp = Coolant Temperature where EGO is active
egocountcmp = Counter value where EGO step is to occur
egodelta = EGO Percent step size for rich/lean
egolimit = Upper/Lower EGO rail limit (egocorr is inside 100 +/- Limit)
engine = Variable bit-field to hold engine current status
FASTIDLE = Fast Idle Temperature
gammae = Total Gamma Enrichments - percent
InjOpen = Injector Open Time
InjOCFuel = PW-correlated amount of fuel injected during injector open
INJPWM = Injector PWM duty cycle at current limit
INJPWMT = Injector PWM mmillisec time at which to activate.
kpa = MAP value in units of KPa
KPARANGEVE = VE Table MAP Pressure Bins for 2_D interp.
last_tps = TPS reading updated every 0.1 seconds
lmap = Manifold Absolute Pressure ADC last Reading
lmat = Manifold Air Temp ADC last Reading
lclt = Coolant Temperature ADC last Reading
ltps = Throttle Position Sensor ADC last Reading
lbatt = Battery Voltage ADC last Reading
lego = Last EGO ADC reading
map = Manifold Absolute Pressure ADC Raw Reading - KPa (0 - 255)
mat = Manifold Air Temp ADC Raw Reading - counts (0 - 255)
mms = 0.0001 second update variable
ms = 0.001 second increment

porta = Port A Data Register
portb = Port B Data Register
portc = Port C Data Register
PRIMEP = Priming pulses (0.1 millisec units)
pulseigncount = Ignition pulse counter
pw = The injector pulse width being used by MS to squirt fuel into your motor.
pwcalc = Computed pulse width - move into variable PW at pulse time
pw = Injector squirt time in 1/10 milleseconds (0 to 25.5 millisec) - applied
pw2= The other pw omparison (injector #2)
pwrun1 = Pulsewidth timing variable 1 - from 0 to 25.5ms
pwrun2 = Pulsewidth timing variable 2 - from 0 to 25.5ms
REQ_FUEL = Fuel Constant
RPMOXLIMIT = Minimum RPM where O2 Closed Loop is Active
rpm = Computed engine RPM - rpm/100
rpmch = Counter for high part of RPM
rpmcl = Counter for low part of RPM
rpmpl = Low part of RPM Period
rpmk = Constant for RPM = 12,000/ncyl - downloaded constant
rpmph = High part of RPM Period
rpmphl = last rpmph value (for odd-fire)
rpmpll = last rpmpl value (for odd-fire)
RPMRANGEVE = VE table RPM Bins for 2-D interpolation
rxoffset = offset placeholder when receiving VE/constants vis. SCI
secl = Time in seconds since MegaSquirt last booted. Low seconds - from 0 to 255, then rollover.
sech = High seconds - rollover at 65536 secs (1110.933 minutes, 18.51 hours)
squirt = Event variable bit field for Injector Firing.
tenth = 1/10th second
tmp1,...,tmp19 = Temporary storage.
tps = Throttle Position Sensor ADC Raw Reading - counts, represents 0 - 5 volts
tpsaccel = The acceleration enrichment.
tpsaclk = TPS enrichment timer clock in 0.1 second resolution
TPSAQ = TPS acceleration amount (fn TPSDOT) in 0.1 ms units
tpsacold = Cold acceleration amount (at -40 degrees) in 0.1 ms units
TPSASYNC = ***** TPS Acceleration clock value
TPSDQ = Deacceleration fuel cut
tpsfuelcut = TPS Fuel Cut (percent).
tpsthresh = Accel TPS DOT threshold
txcnt = SCI transmitter count (incremented)
txgoal = SCI number of bytes to transmit
txmode = Transmit mode flag
T1SCX_NO_PWM = No PWM
VE = 64 bytes for VE Table
vecurr = The current computed VE value determined by look up in the VETABLE using

RPM and MAP.
VOLTOXTARGET = O2 sensor flip target value
warmcor = The warmup correction factor applied due to start-up and coolant temperature status.
WWU = Warmup bins(fn temp)

MegaSquirt MC68HC908GP32 Instruction set:

ADC = Add with Carry
ADD = Add without Carry
AIS = Add Immediate Value (Signed) to Stack Pointer
AIX = Add Immediate Value (Signed) to Index Register
AND = Logical AND
ASL = Arithmetic Shift Left
ASR = Arithmetic Shift Right
BCC = Branch if Carry Bit Clear
BCLR n = Clear Bit n in Memory
BCS = Branch if Carry Bit Set
BEQ = Branch if Equal
BGE = Branch if Greater Than or Equal To
BGT = Branch if Greater Than
BHCC = Branch if Half Carry Bit Clear
BHCS = Branch if Half Carry Bit Set
BHI = Branch if Higher
BHS = Branch if Higher or Same
BIH = Branch if IRQ Pin High
BIL = Branch if IRQ Pin Low
BIT = Bit Test
BLE = Branch if Less Than or Equal To
BLO = Branch if Lower
BLS = Branch if Lower or Same
BLT = Branch if Less Than
BMC = Branch if Interrupt Mask Clear
BMI = Branch if Minus
BMS = Branch if Interrupt Mask Set
BNE = Branch if Not Equal
BPL = Branch if Plus
BRA = Branch Always
BRA = Branch Always
BRCLR n = Branch if Bit n in Memory Clear
BRN = Branch Never
BRSET n = Branch if Bit n in Memory Set
BSET n = Set Bit n in Memory
BSR = Branch to Subroutine
CBEQ = Compare and Branch if Equal
CLC = Clear Carry Bit
CLI = Clear Interrupt Mask Bit
CLR = Clear
CMP = Compare Accumulator with Memory
COM = Complement (One's Complement)
CPHX = Compare Index Register with Memory
CPX = Compare X (Index Register Low) with Memory
DAA = Decimal Adjust Accumulator
DAA = Decimal Adjust Accumulator (Continued)
DBNZ = Decrement and Branch if Not Zero
DEC = Decrement
DIV = Divide
EOR = Exclusive-OR Memory with Accumulator
INC = Increment
JMP = Jump
JSR = Jump to Subroutine
LDA = Load Accumulator from Memory
LDHX = Load Index Register from Memory
LDX = Load X (Index Register Low) from Memory
LSL = Logical Shift Left
LSR = Logical Shift Right
MOV = Move
MUL = Unsigned Multiply
NEG = Negate (Two's Complement)
NOP = No Operation
NSA = Nibble Swap Accumulator
ORA = Inclusive-OR Accumulator and Memory
PSHA = Push Accumulator onto Stack
PSHH = Push H (Index Register High) onto Stack
PSHX = Push X (Index Register Low) onto Stack
PULA = Pull Accumulator from Stack
PULH = Pull H (Index Register High) from Stack

........continued on next page

MegaSquirt MC68HC908GP32 Instruction set - cont.

PULX = Pull X (Index Register Low) from Stack
ROL = Rotate Left through Carry
ROR = Rotate Right through Carry
RSP = Reset Stack Pointer
RTI = Return from Interrupt
RTS = Return from Subroutine
SBC = Subtract with Carry
SEC = Set Carry Bit
SEI = Set Interrupt Mask Bit
STA = Store Accumulator in Memory
STHX = Store Index Register
STOP = Enable IRQ Pin, Stop Oscillator
STX = Store X (Index Register Low) in Memory
SUB = Subtract
SWI = Software Interrupt
TAP = Transfer Accumulator to Processor Status Byte
TAX = Transfer Accumulator to X (Index Register Low)
TPA = Transfer Processor Status Byte to Accumulator
TST = Test for Negative or Zero
TSX = Transfer Stack Pointer to Index Register
TXA = Transfer X (Index Register Low) to Accumulator
TXS = Transfer Index Register to Stack Pointer
WAIT = Enable Interrupts; Stop Processor