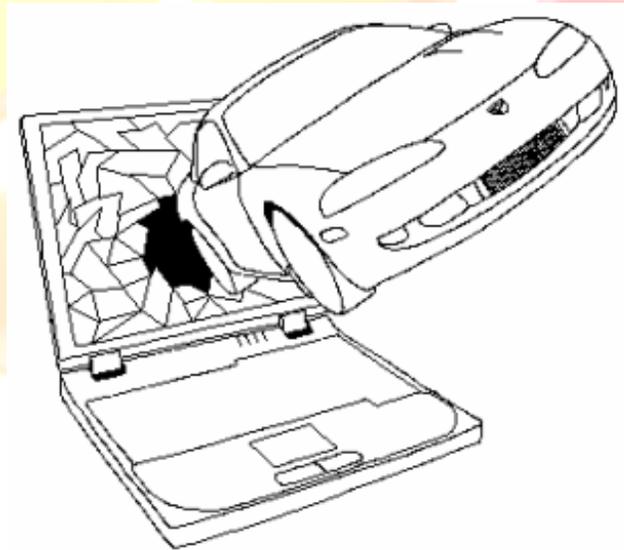


DSMLink

V2.5

Unleash the power of your DSM



December 12, 2008



Foreword

DSMLink is the result of several years worth of effort, literally. Hundreds of man-hours, into the thousands, were put into the development of this product. Many, many nights were spent pouring over disassemblies and testing little changes here and there until everything was understood and complete.

Fortunately, we have enjoyed every minute of it. We are DSM enthusiasts ourselves; we wouldn't be doing this if we didn't enjoy it. We have put together a product that we feel provides a unique and powerful set of tuning tools. We use it on our own cars on a regular basis. We have refined *DSMLink* over the last few years to include the features and functions that we feel really offer the best compromise between ease of use and "tuneability".

That last sentence might need a little explanation. *DSMLink* is not a full-fledged, standalone ECU replacement product. It does not provide every single little piece of adjustment you could possibly add to the stock ECU. There are countless things we could throw into *DSMLink* to provide all sorts of tweaks and adjustments, but that isn't the design philosophy of *DSMLink*.

Our goal is to produce a product that provides the DSM community with an enhanced data logging facility (MUCH needed in the '95+ models) and all the fuel and timing adjustments that about 90% of that group would ever need without being overly complicated. At the same time, we do not want to seriously limit the tuneability of the car by over simplifying the interface. We're striving for a nice compromise. We think *DSMLink* fills that gap, and we hope you agree.

Special Thanks

The laptop portion of *DSMLink* is open source. That is, we provide all the source code necessary to build or modify the laptop portion of *DSMLink* on your own. You can also review the code to find bugs, add features, or to develop a completely separate client version of *DSMLink* if you wish. The following folks have done just that.

MARTIN CHECK – Martin implemented the horsepower and torque calculators included with *DSMLink*. These routines were derived from work done by John P. and Anil R. on a standalone excel spreadsheet type of thing. Martin took those calculations and ported them into *DSMLink*. Many thanks go to Martin for this work. The routines are surprisingly accurate and provide a very useful feature in *DSMLink*.

HAL LANDRY – Hal has completed work on a Palm version of the *DSMLink* client. It's available for **FREE** download to registered *DSMLink* users. Many people are now leaving their laptops at home. Many, many thanks go to Hal for all his hard work. Few people will be able to appreciate how much work is involved in getting something like this up and running. We certainly do!

WARNING

DSMLink software is intended for off road use only. Without a doubt, use of this product will alter your wide-open throttle (WOT) emissions. *DSMLink* has not passed any sort of certification to qualify it for use on public streets.

DSMLink provides a mechanism with which the user can alter the fuel and timing control of the factory electronic control unit (ECU). Used wisely on a properly equipped vehicle, this can have big benefits at the track. But if *DSMLink* is used unwisely or on a car that does not have adequate support systems such as an ample fuel supply, you can **DESTROY** your engine! There is no question about this. You can configure *DSMLink* to destroy your engine.

The user is expected to have a thorough understanding of the tuning process. This product is not for the beginner. *DSMLink* is intended for the individual who already knows things like what injector pulse width means and how it relates to fuel supply and airflow during the tuning process. If you are not comfortable with these concepts, then *DSMLink* is not the product for you. It's far too easy to destroy your motor with a careless flick of the mouse... Use with care and at your own risk. ECMTuning cannot be held liable for any damage done to your vehicle through the use of *DSMLink*.

Nitrous note: Please be aware that a master control switch is also highly recommended for use in conjunction with *DSMLink's* nitrous controls. You do NOT want a spurious nitrous control signal to get through to your nitrous and fuel solenoids while you're cranking, for example. Use a master control switch along with *DSMLink's* controls to help prevent potential damage.

OBD compatibility

DSMLink communicates with the ECU over a serial connection. The port the ECU uses for this communication is the same one used by the built-in factory protocol provided by Mitsubishi. Maintaining compatibility with the factory mode of communication introduces a few issues.

First, the ECU normally looks for some specific signaling sequence to begin factory mode communication. *DSMLink* needs to "break" the ECU out of this mode to establish its own mode of communication. Once the ECU is in *DSMLink* mode, the factory mode will not function until the ECU is restarted (the key turned off for more than 5 seconds, then back on). In addition, on the '95-'99 platform, the *DSMLink* cable does not function for certain OBDII programs such as OBDTool nor do the standard OBDII cables function as *DSMLink* cables. For the '90-'94 platform, it may be that other cables will work with *DSMLink*, but we can't guarantee this and we make no effort to ensure compatibility.

Table of Contents

<u>FOREWORD.....</u>	<u>3</u>
<u>SPECIAL THANKS</u>	<u>3</u>
<u>WARNING.....</u>	<u>4</u>
<u>OBD COMPATIBILITY.....</u>	<u>4</u>
<u>QUICK START.....</u>	<u>6</u>
<u>APPLICATION USAGE.....</u>	<u>8</u>
File menu	8
Edit menu	8
<i>Notes and Default Notes</i>	8
<i>Options</i>	8
Display menu	11
<i>Basics</i>	11
<i>Values dialog</i>	11
<i>Other</i>	13
ECU menu.....	14
<i>Timing</i>	15
<i>Fuel</i>	16
<i>Airflow/MAF Compensation</i>	18
<i>VE Table</i>	20
<i>RPMs</i>	21
<i>TPS</i>	22
<i>Antilag</i>	23
<i>Knock</i>	23
<i>Nitrous</i>	24
<i>Dash</i>	25
<i>Miscellaneous</i>	26
<i>DTCs</i>	27
<i>Captured values</i>	28
<u>DISPLAYABLE VALUES.....</u>	<u>30</u>
<u>CHIP SPECS.....</u>	<u>31</u>

Quick start

This section is intended to give the antsy user a quick peek at a typical session with *DSMLink*. First, of course, you need to connect the supplied *DSMLink* cable. One end connects to the serial port of your laptop; the other connects to the diagnostic connector on the car. On the '95-'99 platform, the diagnostic connector can be found under the dash, in the area between the steering column and the inner kick panel. On the '90-'94 platform, it can be located next to where the driver's left leg would normally rest.

Once the cable is connected, turn on the ignition in the car (no need to run the engine) and start up the *DSMLink* application. Figure 1 shows the *DSMLink* startup screen.

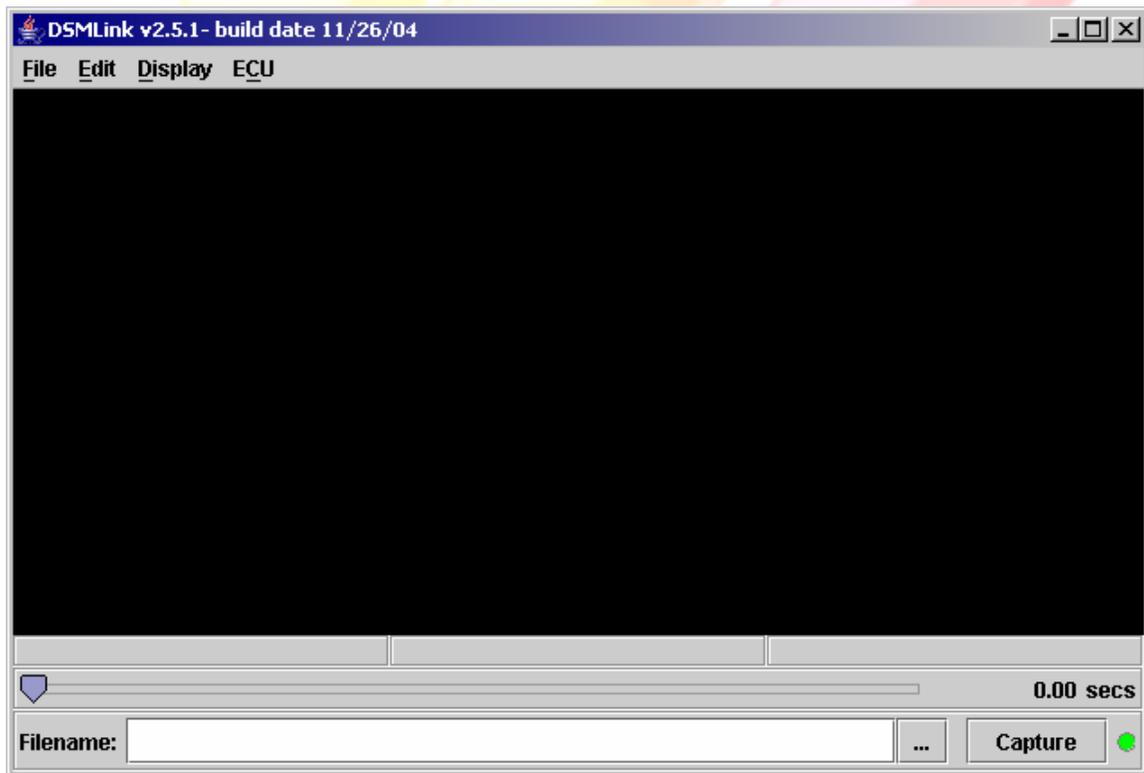


Figure 1: Initial *DSMLink* Screen

If all is well with the setup, then all you have to do to start a capture is to press the **[Capture]** button in the lower right hand corner of the screen. This will establish a connection with the ECU and start a log file in the current *DSMLink* directory. Alternatively, you can select a different file name by clicking the “...” button next to the **[Capture]** button before starting the capture.

Once the **[Capture]** button is pressed, a connection is attempted with the ECU. This is the point where errors in your configuration are most likely to crop up. These errors include things like the serial port not being enabled in the laptop BIOS, permissions under your operating system not being quite right, or just a loose cable connection either

to the laptop or to the car's diagnostic connector.

If your serial port is not defined to be COM1, then you will want to change this setting by going to the **Edit->Options** dialog. This will bring up a dialog box that can be used, among other things, to change which COM port (or tty device under Linux) will be used by *DSMLink* for communication with the ECU.

Unfortunately, if hardware configuration errors exist, it's going to be largely up to the user to figure them out. Laptop configurations vary widely and we simply cannot diagnose every single hardware problem. *DSMLink* has been tested on several different configurations without any problems, as long as the laptops were configured properly. We expect this to be the case for most other laptops as well.

Once the connection is established and a data capture has been started, the **[Capture]** button will change to a **[Stop]** button. When you're ready to stop the capture, click the **[Stop]** button in the lower right hand corner. Doing this will stop the stream of data coming out of the ECU and will reset the graph to the start of the capture.

The capture has been saved to the file listed at the bottom of the screen. The slider bar just above the filename box at the bottom of the screen can be used to navigate the data. You can either use the mouse/trackpad or the keyboard to move the slider. For accelerated movement with the keyboard, hold the **Ctrl** key down while you use the left and right arrows. To change the data displayed on the graph, use the **Display->Values** dialog.

Application Usage

File menu

The **File** menu needs little explanation except for the **Export...** item. The data that gets exported is the data currently selected for display via the **Display->Values** dialog. As explained in other parts of this document, the data contained within a logfile and the data actually calculated and displayed on the screen are two very different things. The **Export** operation of *DSMLink* writes the set of data currently displayed on the screen in the graph panel to a CSV (comma separated values) file appropriate for importing into Excel or some other spreadsheet type application.

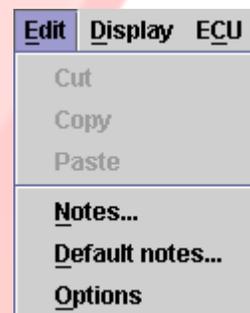


Edit menu

Notes and Default Notes

Notes are a way of adding comments to a log file. These comments can be any free-form text you want to enter. The log file contains a date and time stamp that can be viewed in the **Notes** dialog.

Default notes are used when a datalog is first created. The lifetime of a set of default notes is one day. **Default notes** are useful for recording things like the name of the event you're attending that day and the general weather conditions. These notes are automatically put into the notes section for every new datalog you capture during the course of the day. You can, of course, edit the notes for each log file later to add more detail.



Options

The **Capture** tab of the **Options** dialog is shown in Figure 2. The **Capture** panel lists a set of options that affect what *DSMLink* does during a capture session.

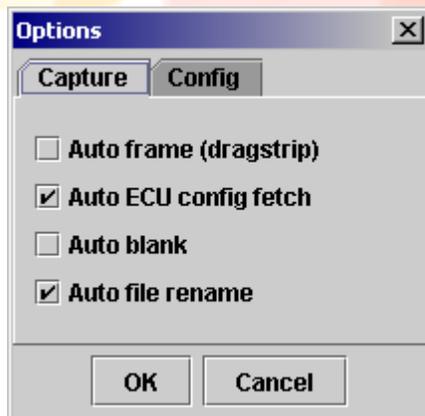


Figure 2: Options->Capture Tab

Auto frame selects a feature that attempts to automatically frame a data capture

assuming it was a pass at the drag strip. This keeps you from having to locate the start and end of a pass and clip accordingly. It also reduces the size of the data capture during the capture process because the capture won't actually start until *DSMLink* determines it's an appropriate time. At the end of a run, *DSMLink* will detect that your speed has stopped increasing and will frame accordingly.

Auto ECU config fetch enables a feature that causes all ECU configuration data to be read in as soon as a connection is established with the ECU. In addition, all ECU configuration data will be saved with each datalog so you can always see exactly how the ECU was configured at the time a particular datalog was captured.

Auto blank disables the graph panel during a capture. This can be useful on really slow laptops where displaying the graph in real-time might not be practical. In extreme cases, data loss might occur if the laptop is not able to keep up with the demands of rendering a graph while capturing the stream of data coming from the ECU. That has not been observed in practice, but it could happen.

Auto file rename will not over write an existing file when a capture session is started. It will attempt to construct a new filename by appending a number to the end of the current filename.

Figure 3 shows the **Options->Config** tab. This tab is used to select which COM port or tty device to use to talk with the ECU. The two rate parameters are used to tune the *DSMLink* display routines to match the capabilities of the PC on which it's being run. The rate is specified in units of data packets received from the ECU. So the instance shown in Figure 3 is configured to update the graph panel once for every four packets received from the ECU. The text panel will be updated once for every 20 packets.

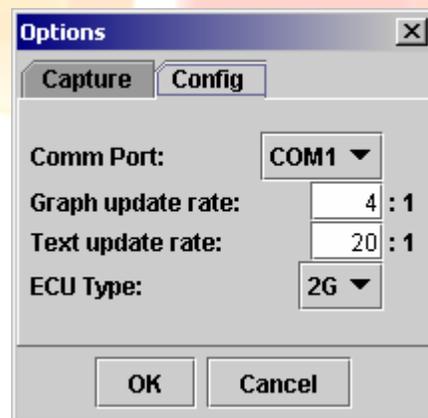


Figure 3: Options->Config Tab

Displaying text is more expensive than updating the graph in terms of CPU resources. So if you observe problems with performance while displaying real-time (stream) data from the ECU, first try to adjust the text update rate and then the graph rate. Note that these values have no effect on the display of data once it has been stored in memory. These

only affect the real-time display of streaming data. Note also that if you have a decently fast laptop, you can get better (smoother, more useable) screen update rates by lowering these values.

The last parameter (**ECU Type**) specifies the default ECU type *DSMLink* should use when attempting to establish a connection with the ECU. This must be set to match the ECU type or else connection failures will occur.

NOTE TO USERS UPGRADING: The **Car** and **Dyno** tabs have been removed. You can now adjust display properties per display routine. So, for example, to adjust the variables used for the A/F ratio estimation, you can simply right click on the A/F item in the graph legend and select **Props**. Or, select the item in the **Display->Values...** dialog and then click **Props**. Either way, you will be presented with a property editor that can be used to tweak that specific item.



Display menu

Basics

The **Display** menu provides operations that act on the capture data and/or adjust the display of that data. The first two entries, **Graph** and **Blank**, simply display the graph or blank the screen. These are rarely used, but do sometimes come in handy. The **Scale...** item displays a dialog box that can be used to adjust the time scale of the displayed data. The units are in seconds to display on the screen at one time.

Values dialog

The **Values...** item displays a dialog box with which you can select the values to display on the graph. This is a little more complicated than it sounds. First, the values you can display will depend on the data that was captured. For example, you cannot display the status of the idle switch if that value was not captured in the log. The values to be captured by the ECU can be configured using the **ECU->Captured values** menu item described in the next section. If an item does not appear in the **Values** dialog, then it's because the data required to calculate that value was not captured.



Figure 4 shows a sample **Display->Values** dialog.

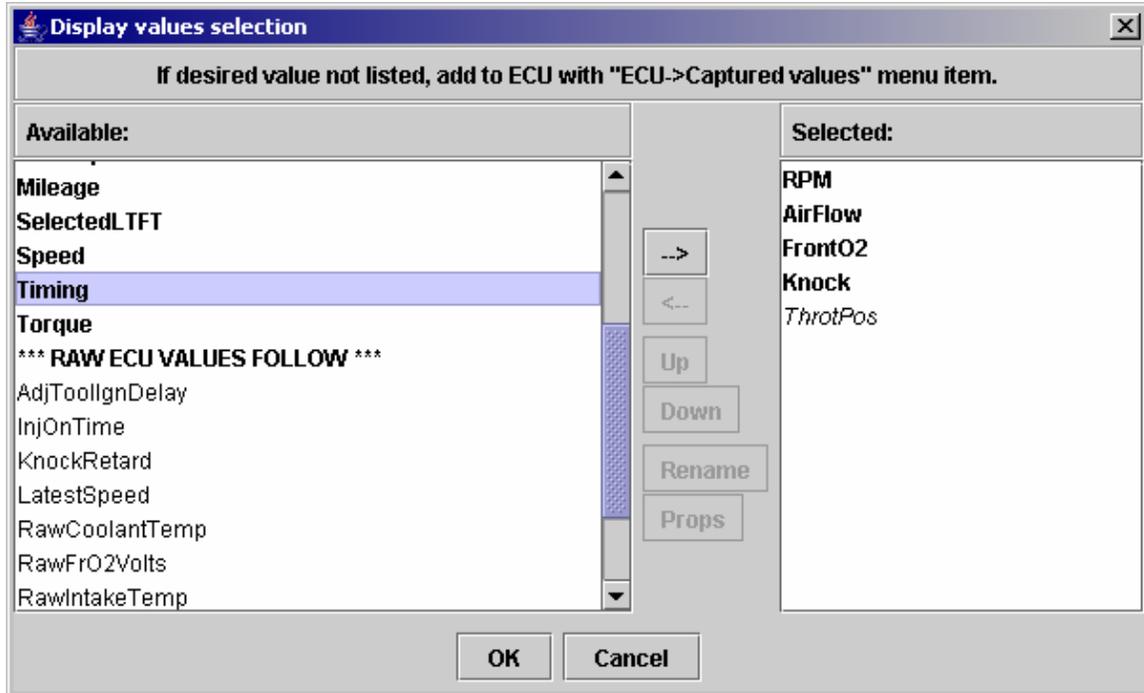


Figure 4: Display->Values Dialog

The items on the left are the values that are available for display based on the data contained within the file currently loaded in memory. The **bold** faced entries are calculated (human readable) values. These are the values most people will be interested in most of the time. The non-bold faced entries in the **Available** list are the raw values captured from the ECU. Rarely are these raw values ever needed, but from time to time they do prove useful, so the **Values** dialog box allows them to be displayed.

DSMLink attempts to remember the user's preferred set of display items, even if those items are not displayable with the current datalog. An example in Figure 4 is the *ThrotPos* value in the list on the right hand side. It's shown in italic font to indicate that it's being included in the list only because it's an item that was used previously, but that it's not really displayable like the other items in the list because the current datalog does not include throttle position data.

The list on the right hand side of the **Values** dialog contains the items currently selected for display in the graph panel. The order of these values determines the color that each entry will be graphed in. For this reason, the **[Up]** and **[Down]** buttons can be used to reorder the entries to suit your preference.

The **Props** button can be used to adjust customizable display property values per display item. An example would be the HP estimation routine. To estimate HP, the routine needs to know some specifics about your car. Figure 5 shows the HP display properties.

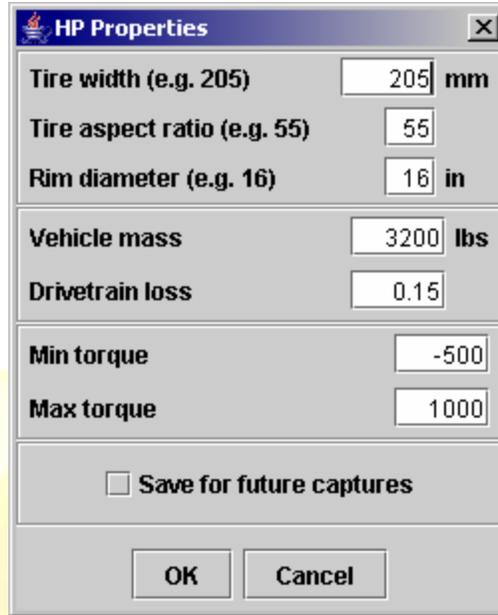


Figure 5: HP Display Properties

The **Save for future captures** checkbox can be used to specify that these values are **your** values as opposed to temporary. Unless you're viewing someone else's datalog, you would normally want to check this box. Checking it saves the values you entered as local defaults to be used any time you start a new capture or open a datalog that does not include overrides to these values. The default checked state of this item is determined as a best guess by *DSMLink*.

The **Rename** button is a little more obscure than the rest. If you want to capture a custom sensor like, say, the GM3Bar sensor, you would normally select **GM3Bar** when adding that sensor to the ECU's list of capture items (**ECU->Captured values**). If you do that, *DSMLink* asks where the **GM3Bar** is connected. For example, if it's connected to the RearO2 input on a '95-'99 ECU, you would select **RearO2** from the list. *DSMLink* will then record that your datalog includes **GM3Bar** data and everything will be fine. However, if you mistakenly select **RearO2** from the initial list instead of **GM3Bar**, then your datalogs will indicate that they include **RearO2** data, not **GM3Bar**. This will prevent you from displaying **GM3Bar** data on the screen because *DSMLink* simply doesn't know that you have **GM3Bar** data in the datalog. It still thinks you just have **RearO2** data from a factory rear O2 sensor.

To fix that situation, the **Rename** button was added. Using it in the scenario presented above, you would select **RearO2** from the display items and then click on the **Rename** button. *DSMLink* will present you with a list of available items to rename the selected entry to, including **GM3Bar**.

Other

The **Display->Auto frame** menu item performs the same operation as the auto frame option described in the **Options** dialog section on page 8. Basically, this menu item

attempts to “frame” the data currently loaded into memory so that very little extraneous data appears before or after the real data from the run. Of course, this assumes the data being viewed is a pass at the drag strip...

The framing operation works by locating a large block of time where your speed was continuously increasing. It looks for the time when that block started, moves back in time a few seconds and chops the data at that point. It then looks forward to find your peak speed (presumably your speed at the end of the run), then adds a few seconds, and chops the data at that point. The result is a fairly well trimmed dataset representing a drag strip pass with very little “noise” before or after the pass. Use of this type framing operation produces very consistent datalogs that can be quickly reviewed because they’re all scaled and clipped similarly.

The **Smooth RPM** menu item is intended to smooth out a stretch of RPM data so as to remove unwanted “glitches”. Some display values and some data exports will want a single gear’s worth of RPM data and they’ll want that RPM data to be continuously increasing. That’s what **Smooth RPM** attempts to do. In order to do that, the data needs to be trimmed first so that a single gear’s worth of data is all that remains. If **Smooth RPM** finds too much variation in the data to be reasonably “smoothed”, it will tell you so.

The **Normalize time** and **Normalize RPM** menu items both attempt to make the time or RPM values conform to a standardized scale. The idea is to provide a way to compare data from multiple runs on the same graph at the same time (using Excel, GNUPlot, or some similar tool). In order to get the data to line up exactly on the same scale, you would “normalize” that scale. These menu items attempt to do that by interpolating all the values as necessary to arrive at a standardized scale. Like the **Smooth RPM** menu item, **Normalize RPM** expects the RPM data to be continuously increasing. If the data has too much variation in RPM, you will be told about it.

Clip to start and **Clip to end** basically remove the data from the current position to the start and end of the log file, respectively.

ECU menu

The ECU menu lists all the things that can be done with the ECU. The **Online** menu item simply initializes the communication protocol with the ECU. This is useful to verify connectivity with the ECU without actually doing anything else. The **Online** menu item changes to **Offline** once a connection with the ECU has been established. You can use the **Offline** menu item if you want to manually force a disconnect from the ECU and free the COM port for use by another application.

The **StreamOn** and **Capture** menu items tell the ECU to start streaming data. The difference is that the data is written directly

ECU	
<u>Offline</u>	F8
<u>T</u> iming...	F5
<u>F</u> uel...	F6
<u>A</u> irflow...	
<u>R</u> PMs...	
<u>T</u> PS...	
<u>A</u> ntilag...	
<u>K</u> nock...	
<u>N</u> itrous...	
<u>D</u> ash...	
<u>M</u> iscellaneous...	
<u>D</u> TCs...	
Load/Save/Type	▶
<u>S</u> tr <u>ea</u> m On	F11
<u>C</u> apture	F12
<u>C</u> aptured <u>v</u> alues...	F7

to a file during a **Capture** operation, but it's simply stored in memory with **StreamOn**. The **StreamOn** function is useful when all you want to do is look at some data from the ECU while debugging a problem with the car in the garage, for example.

With the **Capture** menu item, a file is created in the last referenced directory and data is written directly to it while being displayed on the graph. The filename is determined by the contents of the **Filename** box at the bottom of the screen and the **Auto rename** option in the Options dialog. If no name is in the **Filename** box, then a name will be created automatically for you. If a name is provided and the **Auto rename** option is disabled, then that file will be overwritten. If the **Auto rename** option is enabled and a name is in the **Filename** box, then that file will be created if it doesn't exist. If a file already exists with that name, then a new name will be generated by either updating the sequence number at the end of the name if one is there or appending a sequence number if one is not already there.

Timing

The **ECU->Timing** dialog is shown in Figure 6. Adjustment of timing with this dialog is pretty straightforward. Adjustments are made at 500-RPM points from 500 to 8000 RPM. Moving a slider "up" one tick attempts to advance the ignition timing by one degree at that RPM point. No changes are made to the ECU until the OK button is pressed. Unlike fuel adjustments, ignition timing adjustments made to the ECU's map with *DSMLink* are always active, regardless of airflow or throttle position.

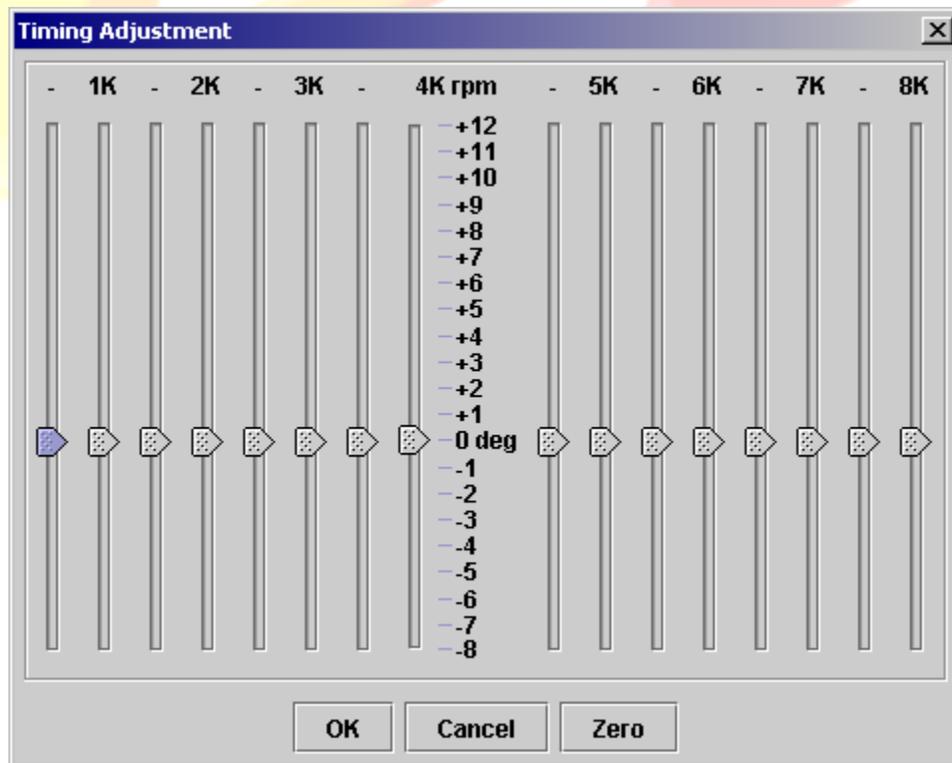


Figure 6: ECU->Timing Dialog

Fuel

The ECU->Fuel dialog is shown in Figure 7. Fuel adjustments are just a little more complicated than the timing adjustments. There's simply more to do with fuel than with timing. There are three main parts of the **Fuel** dialog: global fuel, dead time, and the typical RPM-based adjustments that you might expect to see. Global fuel and dead time are adjustments that get applied all the time. These adjustments are provided to compensate for aftermarket injectors. RPM-based adjustments are only made during certain modes of open loop operation in the ECU, primarily the WOT open loop mode.

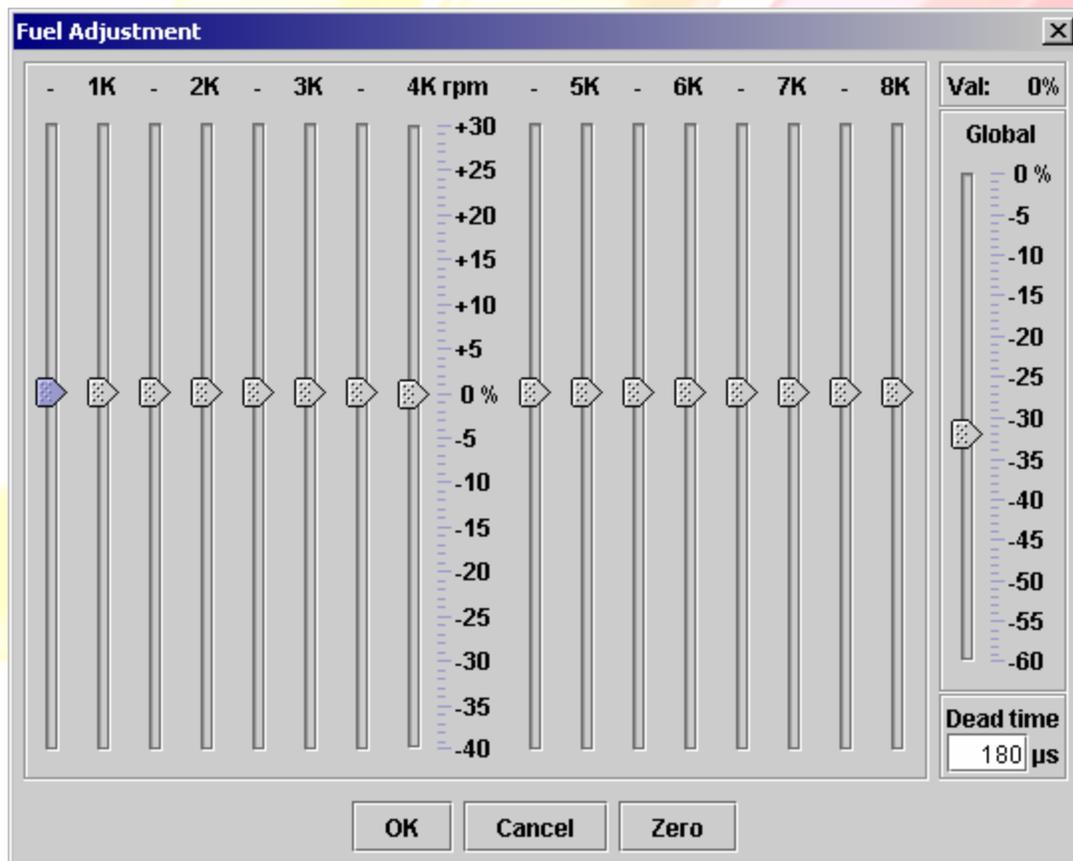


Figure 7: ECU->Fuel Dialog

Basic Global and Deadtime

Aftermarket injectors are sized “larger” than the factory injectors so as to provide more fuel in the same amount of time. This scaling of fuel flow is easy to calculate and deal with. Simply divide the old flow rate by the new rate and subtract 1 to estimate the “global” adjustment you need in order to compensate for the new injectors. This does assume both injectors are rated at the same pressure. As an example, consider the purchase of 720 cc/min injectors.

$$\text{Adjustment} = 450 / 720 - 1.0 = 0.625 - 1.0 = -0.375 \text{ or } -37.5\%$$

So in order to scale the ECU’s fuel curve down to compensate for 720 injectors, we would need to reduce the ECU’s fuel calculations by 37.5%. But there’s a trade off with larger injectors. You don’t just get more fuel flow; you also get slower response.

The ECU takes into account the response characteristics of the factory injectors when calculating how long to “fire” them. It first calculates an appropriate time from airflow, knock, and all sorts of other data. It then adds an amount of time based on battery voltage to account for how long it will take the injector to physically open up after being told to do so. This injector response time is referred to as dead time, dwell time, saturation time, and probably a whole slew of other names. It’s a characteristic of injector design and size.

The **Dead time** area of the **Fuel Adjustment** dialog is where you can make adjustments to this dead time figure. The value entered into this area gets added to the value calculated by the ECU. While there is a way to calculate what this value should be, a trial and error approach tends to work best. Use the long-term fuel trim (SelectedLTFT) value calculated by the ECU as your guide. The following outlines a good procedure for getting the dead time right for your setup.

1. Car must be fully warmed up then must be operated at cruise for about 10 minutes followed by idle for 5 minutes to get stable LTFT values.
2. Compare the LTFT at idle and cruise. You want both to be right about zero (+/- 5% or so).
3. If the two are about the same, dead time setting is good.
4. If they are equal but non-zero, adjust the global setting.
5. If idle LTFT is more positive than cruise LTFT, increase the dead time, less, decrease.

The following table can be used as a rough guide when trying to dial in new injectors. The dead time values are provided as examples of what we have found to work well in the cars we have tested or have been submitted to us by our users. Your particular set of injectors may need slightly different values, but these should be close.

Size (cc/min)	Global (%)	Dead time (µs)
RC 550	-18%	180
Hahn 625	-18%	180
Denso 660	-31%	180
Denso 720	-38%	450
PTE 580	-16%	180
PTE 680	-30%	210
PTE 780	-40%	315
PTE 880	-46%	330
FIC 650	-31%	315
FIC 750	-40%	315

FIC 850	-47%	315
FIC 950	-53%	315

A minor quirk in the implementation of the dead time value adjustment requires that any value entered into this dialog must be evenly divisible by 15. The software will automatically make appropriate adjustments for values you enter that are not evenly divisible by 15.

Also note that other long term fuel trim variables do exist. These can be used instead of SelectedLTFT if you wish. Depending on your ECU type, these will include LTFT Lo, LTFT Mid, and LTFT Hi. These are the individual long term fuel trim variables inside the ECU. Which one is used at a given point in time is determined by logic inside the ECU. It's mostly based on airflow. Low airflow typically seen at idle will cause LTFT Lo to be used while airflow typically seen under cruise conditions will usually result in the use of LTFT Mid (LTFT Hi on 1Gs). Whichever value is selected for use by the ECU is what you will see in the SelectedLTFT variable.

Fuel pressure

If you have adjusted your fuel pressure beyond the factory default, you will need to take this into account when calculating the global fuel scale and, to some extent, the dead time as well. Increasing the base fuel pressure increases the flow rate of the injector and slightly increases the expected dead time values as well. The increase in flow rate can be calculated like so:

$$\text{FlowFactor} = \text{sqrt}(\text{NewPress} / \text{OldPress})$$

For example, let's say you have increased base pressure from the factory value of 43.5psi (2G) to 55psi:

$$\text{FlowFactor2G} = \text{sqrt}(55 / 43.5) = 1.1244$$

Note that a manual 1G will use 37psi instead of 43.5psi:

$$\text{FlowFactor1G} = \text{sqrt}(55 / 37) = 1.2192$$

Applying this adjustment to the example above running 720s gives the following global fuel adjustment:

$$\text{Adj2G} = 450 / 720 / 1.1244 - 1.0 = 0.5559 - 1.0 = -0.444 \text{ or } -44.4\%$$

$$\text{Adj1G} = 450 / 720 / 1.2192 - 1.0 = 0.5126 - 1.0 = -0.487 \text{ or } -48.7\%$$

The table below is provided as a reference for calculating all this stuff.

Model	Trans	Injector size	Base Pressure
1G	M/T	450 cc/min	37 psi
1G	A/T	390 cc/min	43.5 psi
2G	A/T	450 cc/min	43.5 psi

2G	M/T	450 cc/min	43.5 psi
----	-----	------------	----------

Airflow/MAF Compensation

The **ECU->Airflow** dialog is probably the single most complicated piece of the *DSMLink* application from the user's standpoint. It's intimidating because the user is required to understand some fundamental car concepts that, up to this point, could have very well been totally ignored. Let's start with the easy part first, **MAF Compensation**.

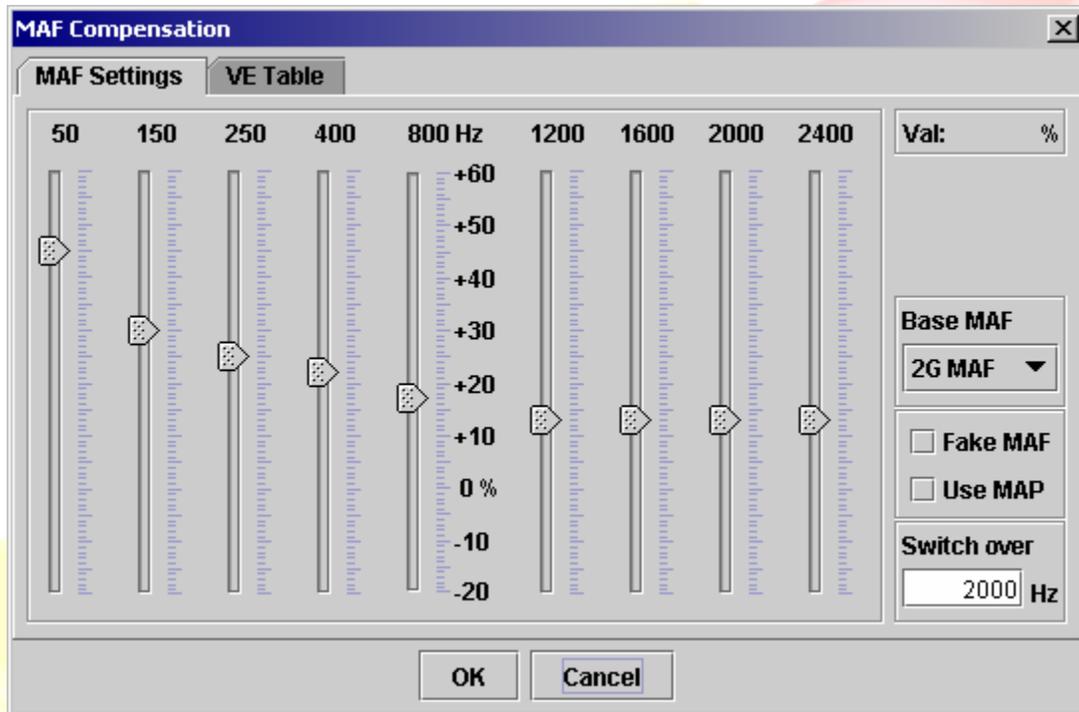


Figure 8: ECU->MAF->MAF Settings Dialog

Note that the **Base MAF** section is not present or needed on the 2G version of *DSMLink*. On 1Gs, this area is used to select the type of MAF installed in the car. If you have a GM MAF installed along with a translator, then you would select the factory MAF type the translator is supposed to emulate (1G or 2G). If you have a 2G hacked MAF installed, then select 2G MAF as the base type and enter appropriate airflow adjustments to account for the hacked state of the MAF as described more below.

The **MAF Compensation** dialog provides percentage adjustments at convenient intervals to apply to the MAF frequency coming into the ECU. What might not be expected is that this table is index by MAF frequency itself. Modifications to the factory MAS will tend to affect things differently based on the airflow rate through the MAS. Slower moving air has less inertia and as a result will be more likely to change direction. When less restrictive paths become available around the center metering section of the MAS (e.g., hacking or bypass tube), slow moving air will be far more likely to take those alternate

paths than will air moving somewhat faster.

The **MAF Compensation** configuration illustrated in Figure 8 represents the values one might expect to use with a hacked 2G MAS. That is, a 2G MAS with the lower and both side honeycombs removed. When airflow metering of this configuration was tested side by side with that of an unhacked 2G MAS, the compensations illustrated above were calculated to provide a good match between the two.

The **Fake MAF** and **Use MAP** options are both closely related. When **Fake MAF** is selected, the signal coming from the real MAF is ignored above the entered **Switch over** point. Instead, a MAF signal is “simulated” by extrapolating from the conditions observed at the **Switch over** point.

If **Use MAP** is selected, *DSMLink* will factor in the value it sees on the MDP sensor line as though it were a manifold pressure (boost) signal. On 1Gs, this feature uses the EGR temp input. This assumes, of course, that you have fed an appropriate pressure signal into the appropriate input line to the ECU. If **Use Map** is not selected, *DSMLink* just uses a hard-coded value for pressure under the assumption that your boost is holding pretty steady beyond the **Switch over** point.

Once the **Switch over** point is exceeded, the MAF sensor is completely ignored until the calculated airflow drops well below the clamp point. So as your revs and estimated airflow continue to increase, *DSMLink* will continue to ignore the MAF sensor. Once you've let up on the gas, the calculated airflow will drop below the clamp point and *DSMLink* will switch back to using the real MAF signal.

VE Table

Both modes (with and without the MAP sensor) use the volumetric efficiency (VE) table to drive the estimation of airflow once the MAF has been "clamped". To understand how VE plays a role in this, let's consider a simplified model of airflow through an engine.

$$flow = pressure * volume * RPM * VE$$

Basically, a given volume (displacement) pressurized to a certain pressure (boost) will only hold so much air. Repeating that process over a period of time (RPM) produces an airflow rate. The VE figure is used to adjust that theoretical limit to match the real limit once everything in the airflow system on a car is taken into consideration.

The default VE table provided with *DSMLink* is illustrated in Figure 9. The values shown there were calculated from readings taken off one of our cars. That car had an ExtrudeHoned intake manifold and ported 2G head. Since the VE table is used relative to the clamp point, it's probably good for most applications with stock manifolds. For a sheet metal intake setup, you'll probably want to tweak the upper RPM VE entries a bit.

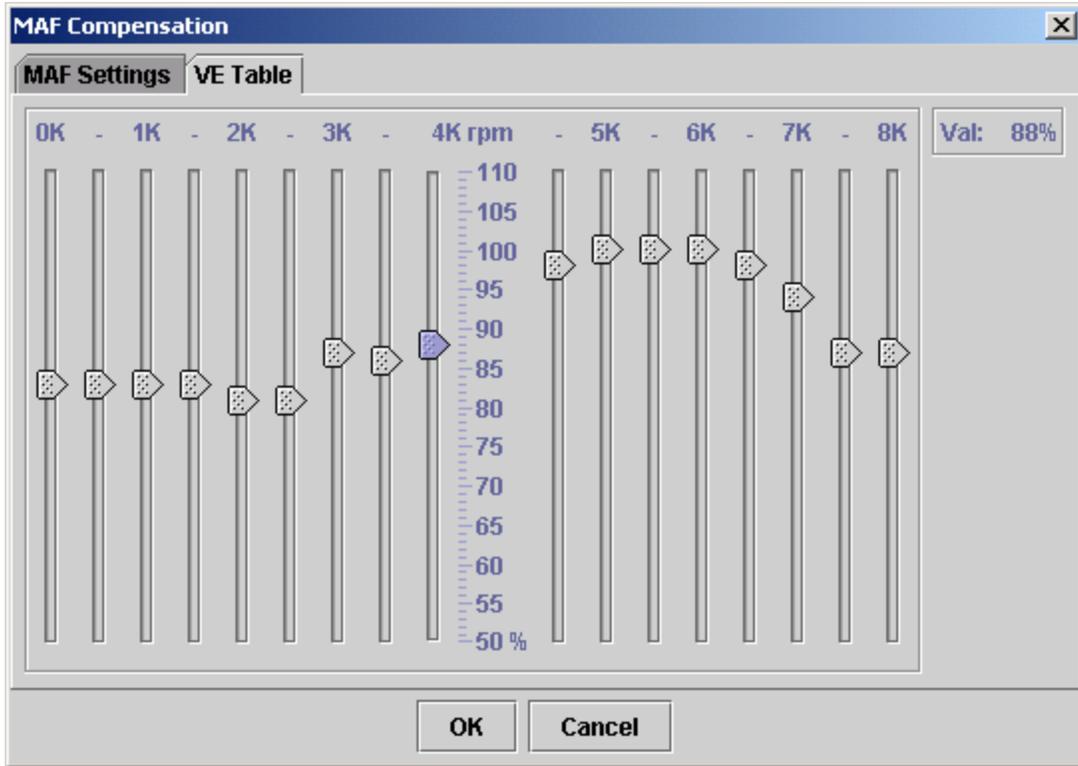


Figure 9: ECU->MAF->VE Table Dialog

To calculate a more precise VE table for your particular setup, first make sure you're running a boost level low enough to keep MAF Hz below your target clamp point (probably around 2700hz on a 2G MAF and 2100hz on a 1G MAF). Then deselect **Fake MAF**, drop your clamp point to some low value like 500hz, and log both MAFComp and MAFSD. MAFComp is the MAF sensor reading after compensations have been applied to it from the MAF Compensation table. MAFSD is MAFComp after VE adjustments have been applied (as well as factoring in pressure if you have **Use MAP** selected). Both of these are calculated even when they're not being used. Adjust the VE table until MAFSD replicates MAFComp. This process will effectively map out the volumetric efficiency of your setup.

RPMs

The **ECU->RPMs** dialog is shown in Figure 10. Idle, launch, and rev limit should have fairly obvious application. Select **Enabled clutch cut** if you have connected the clutch switch to the ECU as outlined in the *DSMLink* instruction sheet that came with the wire. Once enabled and connected properly, you can select a shift rev limit that will be active anytime the clutch pedal is in and vehicle speed is greater than **Launch Spd**. **Launch Spd** is used to determine the point at which the rev limit changes from **Launch** to the full **Rev limit**. A value slightly lower than the speed at which you'd expect the car to be moving in 1st gear at the selected **Launch RPM** seems to work well. Entering a **Launch Spd** value that's too high will effectively make the **Launch** limit your new **Rev limit** in

1st gear because the car won't reach the shut off speed under normal conditions by the time the **Launch** rev limit is reached.

Coast FC adj is used to tweak the lower limit on coasting fuel cut. Enter a value in here when you select an idle point greater than 950 RPM and observe any odd behavior while coasting down or letting the revs drop. As a general rule of thumb, you should enter the difference between your selected idle point and 950. So if you select an idle of 1100 RPM, you might try entering 150 into the **Coast FC adj** box.

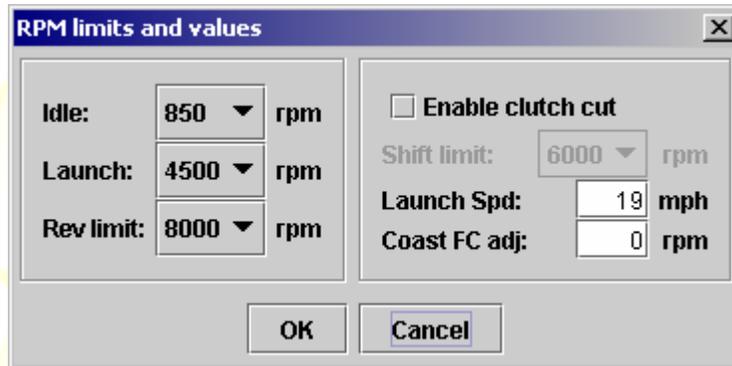


Figure 10: ECU->RPMs Dialog

TPS

The **ECU->TPS** dialog is shown in Figure 11. Aftermarket TPS units don't always output the same readings as the stock unit. To adjust for this, select **Adjust TPS** and enter an appropriate offset and scale. The **TPS Offset** value is added to **RawThrotPos** inside the ECU to adjust this value, then the scale factor is applied. To estimate appropriate offset and scale values, log **RawThrotPos** while the throttle is both fully closed and fully open. You want to see **RawThrotPos** values around 32 or so when the throttle is fully closed. Alternatively, you can view **ThrotPos** and make sure it's reading close to 0% with the throttle closed. When fully open, you want **ThrotPos** to be greater than 90%.

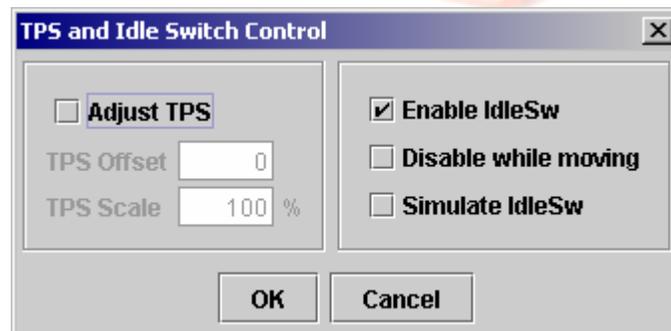


Figure 11: ECU->TPS Dialog

Select **Enable IdleSw** if you want the ECU to see an active reading from the idle switch (either simulated or real). Deselect this option if you want the ECU to always see the idle

switch off. If selected, then you can optionally disable its operation while moving or you can simulate the value based off TPS. Each of these combinations is useful in different scenarios. You might want to simulate the idle switch if yours isn't quite working right but your TPS is. You might want to disable it entirely if you just don't care at all about idle or coasting fuel cut.

Antilag

First a foreword about the anti-lag system (ALS) implemented by *DSMLink*. This is not an all out rally or road course version of ALS. We do not provide a means of activating throttle bypass devices, nor do we use manifold pressure in any sort of closed loop control of the ALS. You can think of this as drag style ALS if you'd like. The dialog used to configure *DSMLink* anti-lag settings is shown in Figure 12.

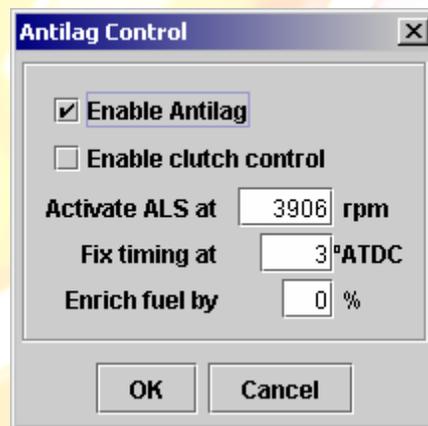


Figure 12: ECU->Antilag Dialog

ALS is only active with the throttle down at least half way and only during launch conditions (speed less than 5 mph). Optionally, it can also be activated during shifts with the clutch in (speed greater than 20mph) by checking **Enable clutch control**. Like the shift rev limit, it's assumed that you have wired in the clutch switch for this functionality to work properly.

The control parameters for ALS include a starting RPM point, below which ALS will not be activated. The user can specify the ignition advance value to use during ALS operation (a value between 3 and 10° ATDC works well) and a percentage of fuel enrichment to apply. We've not seen much difference at all with fuel enrichment, but your mileage may vary.

Knock

The primary purpose of the **ECU->Knock** dialog shown in Figure 13 is to provide a way for users with persistent "phantom knock" to work around their problem. Phantom knock is typically observed in part throttle conditions where true detonation problems are highly unlikely. The cause of this problem can be a variety of things and the symptoms are very

annoying. This is such a common problem that we've decided to provide a tool to help work around it.

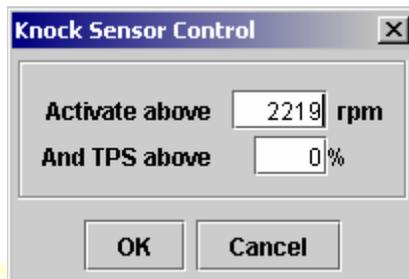


Figure 13: ECU->Knock Dialog

Don't be fooled with this dialog, though. It's not applicable only to phantom knock conditions. You can easily configure this thing to effectively disable the knock sensor entirely. Choose your values carefully. Those values shown above are equivalent to stock operation of the knock sensor on a 2G. 1Gs do not include the RPM disable point from the factory.

Nitrous

The **ECU->Nitrous** dialog is shown in Figure 14. The meaning of the parameters should be fairly obvious. **IMPORTANT NOTE:** Please be aware that a master control switch is also highly recommended for use in conjunction with *DSMLink's* nitrous controls. You do NOT want a spurious nitrous control signal to get through to your nitrous and fuel solenoids while you're cranking, for example. Use a master control switch along with *DSMLink's* controls to help prevent potential damage.

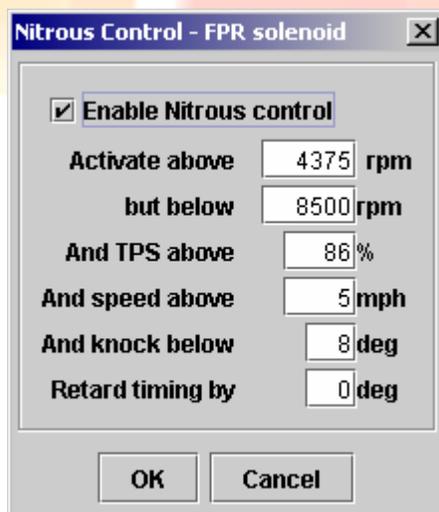


Figure 14: ECU->Nitrous Dialog

To activate the nitrous and fuel solenoids, *DSMLink* takes over control of the fuel

pressure regulator solenoid line. When activated (by selecting **Enable Nitrous control**), this line no longer behaves like the factory intended. Instead, the line will be deactivated (open) while the nitrous parameters are not satisfied and then activated when they are. Additional logic inside the ECU will prevent a state change within 100ms of the previous state change. Also, when clutch switch control is activated (either in the **RPMs** or **ALS** dialog), nitrous controls will be deactivated as well when the clutch switch indicates that the clutch pedal is in. And, finally, nitrous controls are deactivated when any of the rev limits are reached.

Note that the **Retard timing** feature is only present on the 1G version of *DSMLink* at the time of this writing.

Dash

The **ECU->Dash** dialog is shown in Figure 15.



Figure 15: ECU->Dash Dialog

Use this dialog to change the point at which the check engine light is flashed to indicate excessive knock or high coolant temps. **Coolant Offset** is basically added to the true coolant temp indicated by the sensor. This can be useful for adjusting the way the ECU responds to coolant temps, for example to activate fan control sooner when using a lower thermostat. The coolant temp value logged by *DSMLink* and the one used to flag the check engine light is still the true coolant temp read from the temp sensor.

The **Boost Gauge** drop down can be used to change the values displayed on the factory boost gauge. **NOTE:** The boost gauge does not display anything useful unless the engine is actually running. Until then, the boost gauge will simply display a fixed value.

Miscellaneous

The **ECU->Miscellaneous** dialog as seen on 2Gs is shown in Figure 16; the 1G version is shown in Figure 17. Operation of the **Devices** and **Solenoids** sections should be obvious. Click the buttons and the appropriate line is activated. The one exception is with the **FPS** button. If nitrous controls are enabled, FPS operation with the **Miscellaneous** dialog will be disabled. If you wish to test FPS solenoid activation via the **Miscellaneous** dialog, then you must first disable nitrous controls **and** enable **Normal FPS** operation.

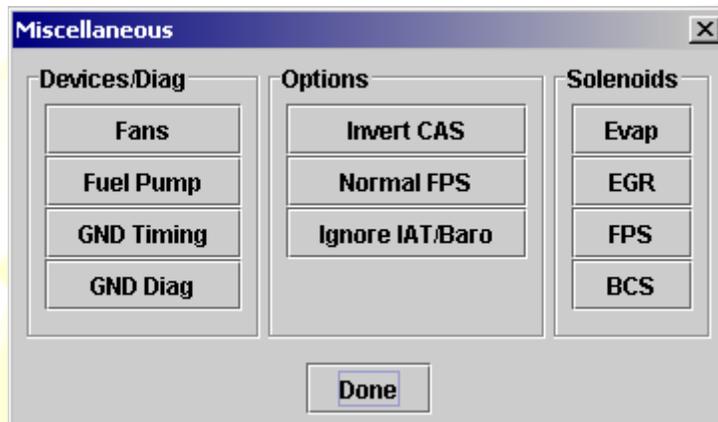


Figure 16: ECU->Miscellaneous Dialog (2G)

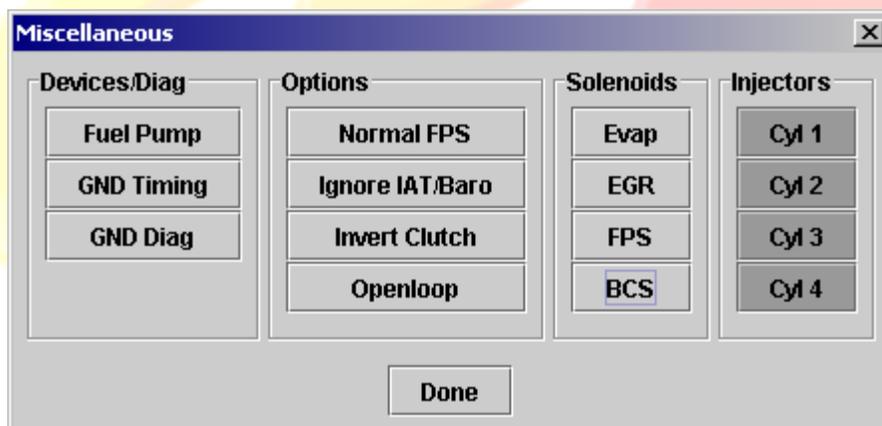


Figure 17: ECU->Miscellaneous Dialog (1G)

The **Signals/Opts** section needs a little more explaining. When selected, **GND Timing** will cause the ECU to behave as though the timing check connector, located under the hood along the firewall in '95/'96 model years, were grounded. Check the shop manual for conditions where this might be useful in diagnostics. Similarly, the **GND Diag** button will do the same for the diagnostic pin on the OBD connector. Again, check with the shop manual as this can be useful in various diagnostic situations, in particular when trying to check/set base timing or base idle.

When selected, **Ign IAT/Baro** will cause the ECU to ignore the intake air temp and

barometric pressure signals for purposes of calculating airflow. Instead, the ECU will use hard coded values of 80F and sea level. These values were chosen based on the values observed with the popular GM MAFT unit. By letting the ECU use software versions of those values, the input lines coming into the ECU are freed up for logging other things.

Normal FPS returns the fuel pressure solenoid operation to standard, factory behavior. This is disabled by default because of the potential conflict with nitrous controls. We don't want a freshly powered up ECU to default to stock operation of a line that might well be connected to and controlling a pair of nitrous and fuel jets.

2G specific

The **Invert CAS** option logically inverts the operation of the cam angle signal coming into the ECU. This should be used when running a '95 ECU in a car that does not have a '95/'96 style cam angle sensor. The **Fans** button will activate both fans on full high.

1G specific

The **Invert clutch** option can be used to tell the ECU that you have your clutch wire installed on the starter switch (the one activated when the clutch pedal is fully depressed to the floorboard) instead of the *DSMLink* standard of the cruise switch (the one activated as soon as you press in on the clutch pedal). The meaning of these two switches is logically inverted, so *DSMLink* needs to know which one you have the wire connected to.

The **OpenLoop** option locks the ECU in open loop mode (ignoring the front O2 sensor input). This option is really only intended for a track car that wishes to use the front O2 sensor input into the ECU for datalogging something other than the front O2 sensor.

DTCs

Figure 18 shows a sample **ECU->DTCs** dialog box. The first section shows the DTCs currently stored in the ECU. The bottom section shows the DTC checks currently enabled. Note that 1Gs will have a different set of DTC enabler checkboxes.

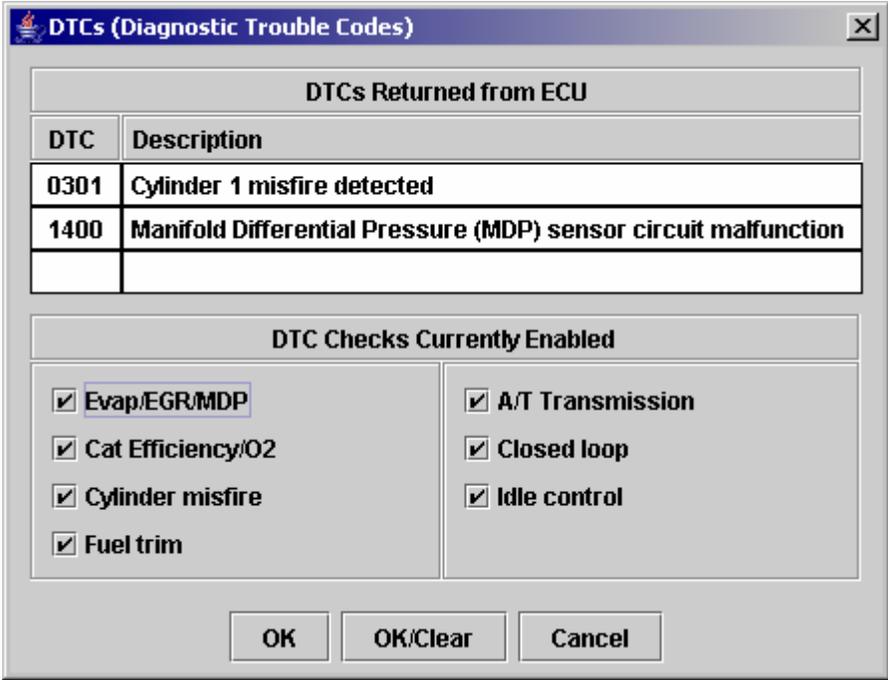


Figure 18: ECU->DTCs Dialog

Captured values

Figure 19 shows the ECU->Captured values dialog box that can be used to add or remove elements from the ECU's data stream. The difference between stream data and the data displayed on the graph is explained in the

Display menu section on page 11. The items listed on the left hand side of the dialog (**Display** items) are the items available to you for display. The bold-faced items are the ones that *DSMLink* can calculate and display based on the stream data listed on the right hand side (**ECU list**).

The **ECU list** shows the raw values that the ECU must capture in order to calculate and display the bold-faced items shown in the **Display** list. Bold-faced items in the **ECU list** are the items that you have added to the core set of items that are always logged by the ECU. You can only remove items from the **ECU list** that are bold-faced. The other items in the **ECU list** are always present in the data stream.

Keep in mind that as you add items to the ECU list, the frame rate of your capture session will be reduced accordingly. For example, if your frame rate is 20 fps with just the core data being capture, then your frame rate will reduce to roughly 15 fps if you were to add another 25% more data to the frame (about 4 new values).

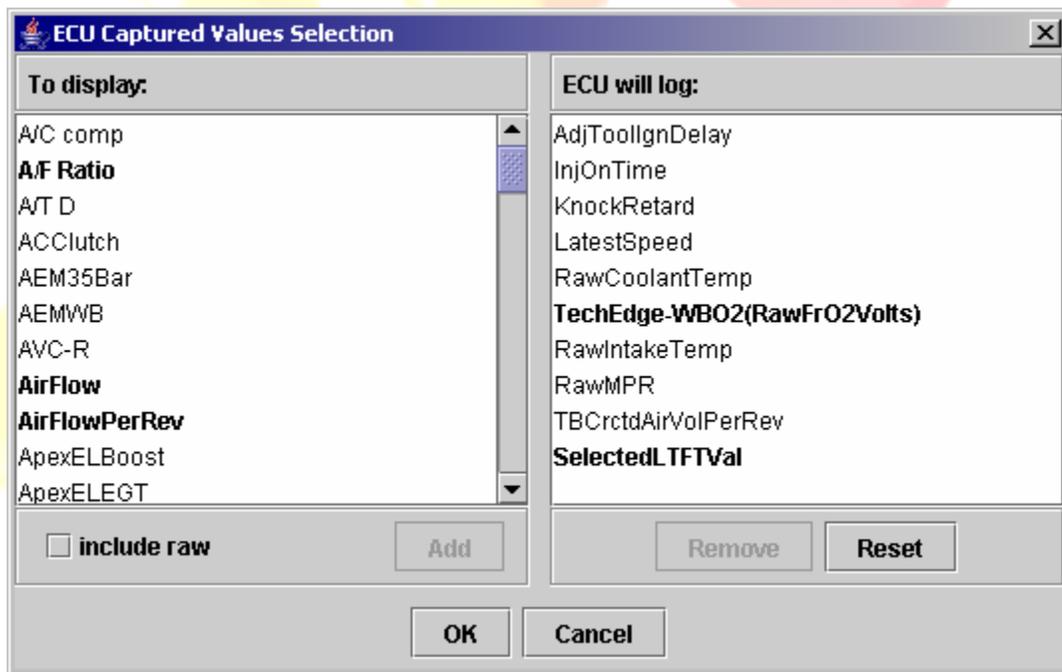


Figure 19: ECU->Captured Values Dialog

Displayable values

Name	Units	Notes
A/C comp	-	AC switch in dash panel is active.
A/F Ratio	-	Estimated air/fuel ratio. Uses values entered in the options dialog.
A/T D	-	Automatic trans in drive.
ACClutch	-	ECU has engaged the AC clutch.
AirFlow	g/s	Airflow expressed in gm/sec.
AirFlowLb	Lb/min	Airflow expressed in lb/min.
AirFlow Per Rev	g/rev	Airflow expressed per revolution. More accurate way of viewing airflow used by the ECU.
AVC-R	psi	Boost pressure assuming the AVC-R pressure sensor is fed into the MDP sensor input.
Baro	in/Hg	Pressure reading from sensor in MAS.
Battery	v	Battery voltage according to the ECU.
CoolTmp	F	Coolant temp seen by ECU.
FourBar	psi	Boost pressure assuming a 4-bar sensor is fed into the MDP sensor input.
FrontO2	v	Front O2 sensor voltage.
FuelPmpRly	-	ECU has activated the fuel pump relay.
FuelPresSol	-	ECU has activated the fuel pressure solenoid.
GM3Bar	psi	Boost pressure assuming GM 3-bar sensor is fed into the MDP sensor input
HP	hp	Calculated HP based on RPM acceleration, gearing, weight, and drag
ISCPosition	-	Idle stepper motor position
IdleSw	-	Idle (throttle closed) switch active.
InjDuty	%	Injector duty cycle calculated from current injector pulsewidth and engine RPM.
InjOn	ms	Injector "on time", expressed in milliseconds.
IntTemp	F	Intake temp from MAS sensor.
KeyStart	-	Indicates that the key is in the start position (i.e., cranking).
Knock	deg	Knock retard expressed in degrees of ignition retard.
LTFT	%	The selected long-term fuel trim value used by the ECU during closed loop operation.
LTFT Lo	%	The low-airflow LTFT value.
LTFT Mid	%	The mid-airflow LTFT value.
LTFT Hi	%	The high-airflow LTFT value. (1G only)
LearnedIdleAdjust	-	Learned idle adjustment
LearnedIdleAdjustAC	-	Learned idle adjustment with AC on
MAF	Hz	MAF Hz used by the ECU after all adjustments are applied
MAFComp	Hz	MAFRaw after just MAF compensation is applied (no SD applied)
MAFRaw	Hz	Raw Hz reading straight from the MAF into the ECU
MAFSD	Hz	MAFRaw after MAF compensations and SD adjustments are applied
MDPSensor	-	MDP sensor reading
MIL	-	Check engine light on.
Mileage	mpg	Basic mileage estimate using injector pulsewidth and speed
O2Htr2	-	Rear O2 heater circuit indicator.
O2Htr1	-	Front O2 heater circuit indicator.
OLFT	%	The resulting open loop enrichment after the user-defined adjustment has been applied.
Octane	-	Octane rating as viewed by the ECU based on long term knock
OffsetCoolTmp	F	Coolant temperature with offset applied from Dash dialog.
OpenLoop	-	Indicates when the ECU is in open loop mode. When false, the ECU is in closed loop mode.
P/S load	-	Power steering pump active.
RadiatorFan	-	ECU has activated radiator fan.
RawKnock1,2,3,4	-	RawKnock values seen by the ECU per cylinder. Cylinder #s don't necessarily line up.
RearO2	v	Rear O2 sensor voltage.
RPM	rpm	Engine RPM.
Speed	mph	Speed sensor reading.
STFT	%	Short-term fuel trim calculated and used by the ECU during closed loop operation.
StockOLFT	%	The factory open loop enrichment value applied to the airflow-based fuel calculation.
TPSvolts	v	Throttle position sensor reading, expressed as voltage
ThrotPos	%	Throttle position sensor reading, expressed in percentage of full open.
Timing	deg	Ignition timing, expressed in degrees BTDC.
Torque	ft-lb	Torque estimate based on airflow...not accurate.
UserOLFT	%	User selected fuel adjustment.
WBO2	-	Wideband sensor fed into MDP sensor line. Assumed to be TechEdge unit.
WGSolenoid	-	ECU has activated the wastegate solenoid.

Chip Specs

DSMLink includes a set of changes to the factory EPROM that are not really covered in any of the previous sections of this manual. These changes are really internal-only and are not accessible in any way from the *DSMLink* laptop software. The table below lists these changes for your reference.

Change	Description
Knock MIL	Flashes the check engine light if timing is pulled back more than 5 degrees due to knock.
Fuel cut	Removes the airflow fuel cut.
Speed limiter	Removes the speed limiter.
Octane reset	Resets the long-term octane value on every restart of the car.
BCS flicker	Removes the boost control solenoid (BCS) flicker between 2300 and 3900 RPMs.
Auto shift timing	Retains ignition timing in between WOT shifts on automatics.
Cranking fuel cut	If the throttle is wide open while cranking, injectors are shut off. This seems to help clear out flooded conditions produced while attempting to start in REALLY cold environments.
OBDII MIL bug	Fixes the OBDII MIL bug in the '95 ECU. Mitsubishi did a recall on the '96 ECUs but offered no fix for the '95.