How to build an EEC Definition file – Part 1 (The Basics)

By Jaysen Anderson & Jason Bolger

There are a couple of ways to go about building a definition file to suit your desired EEC. This document attempts to cover the basics.

Ok lets get started, here is the procedure in point form.

- Dump your EEC bin (if you cant already get your bin from the net)
- Define parameters in your EEC (or find a definition for your EEC on line)
- Start editing parameters
- Burn a copy of your modified bin to a chip
- Plug it in and your done

Easy hey? Well maybe not as easy as it sounds!

Something to note is that even if your EEC bin hasn't been dumped or defined, you can "generally" run a bin from the same model car on your ECU. For instance Jaysen has an EF 5ltr with a 6DFC ELXR8 manual ECU. A couple of mates did manual conversions on EL's, one with a 6DGC & one with a 6DGD, both being auto ECU's, but they can be converted to manual ECU's with the J3 ROM chip. This has been tested, but if you're unsure as to how your bin is setup, i.e. some EF's run 56k code bins instead of 88k, you might get yourself in to trouble, and do so at your own risk.

Another thing is that you should actually have a need to edit your bins, or you might just end up with a pile of molten metal, or a car that generally runs bad and chews fuel. When we say you should have a reason, we don't mean so you can remove the 180kph speed limiter for street use, this is just dumb! Don't get us wrong, we like spirited driving too, but to speed excessively is DANGEROUS, so keep it for the track. Most people will not need speed limiters adjusted unless they really do have a race car, or they are just plain dumb.

Method 1

The first and the easiest way is to find an existing definition that has been written for you style of ECU (ie EECIV 5ltr MAF V8) and then use the binary that the definition is designed for, to locate scalers, flags/switches & maps/tables.

For Example:

- Start with GUFB definition file
- GUFB suits the A9L Mustang ECU
- A9L is an EECIV, with 5 distinct banks, 2 of code and 3 of null data
- 6DFC Aussie EL XR8 is very similar in code.

Use the def file to find the parameters in the A9L binary with a hex reader, then find the same or similar parameters in the EEC you are trying to define. A thing to note here is that even though some of the parameters may change a little, the code around the data all follows a similar pattern, so don't restrict your search to a word or double word, look a few lines above and below in the code to make sure that you are in fact looking at the right thing.

Method 2

The second way is a lot harder, and you need to be a lot more competent with assembler code (which both J's are not!) You need to disassemble the code, but the 8061 & 8065 processors are proprietary to Ford, so you need to use a close cousin the 8096, and fudge your way through from there. This is very time consuming. We do not know enough about it to comment effectively, luckily however, the 2 types of EECV's used in Aussie Falcons have already had a good deal defined on them, so all you need to do if your EEC hasnt been defined yet, is use method 1 above.

Method 3

There is a 3rd way, but we would advise against it. Look through the binary with your hex editor, find what you think is a map or a table, change it and see what happens. This sort of trial and error is how EEC hacking started, but it can cause great amounts of harm if not done with finesse. Scalars are even harder to find as they are only a byte or word in length, so good luck with that.

Links

Here are some sites with software, hardware and info that we think will help you.

http://sourceforge.net/projects/hexplorer/ My (Jaysen's) preferred hex editor the as bytes are colour coded according to their value
http://tunerpro.markmansur.com/downloadApp.htm A very user friendly graphical user interface that is easy to create and modify definition files (TunerPro calls its def files xdf's)
http://tunerpro.markmansur.com/downloadBinDefs.htm This is where you can start by downloading bins and def files, the ones here are ready to use, but you can modify them to suit your application.
http://www.techcore.com.au/eec/ A collection of Aussie EEC bins & definitions that are ready to be used and modified created by Jason, as well as the locally sourced hardware, ie chips, burners ect.
http://www.moates.net/ This is an overseas option for the hardware (although at greater cost).
http://forum.tunerpro.net/ A helpful forum
http://www.moates.net/phpforum/index.php A helpful forum
http://eectuning.org/forums/ A helpful forum
http://tunerpro.markmansur.com/tutorials/UnderstandingHex.htm If you dont know how to read hex

Hardware (Local)
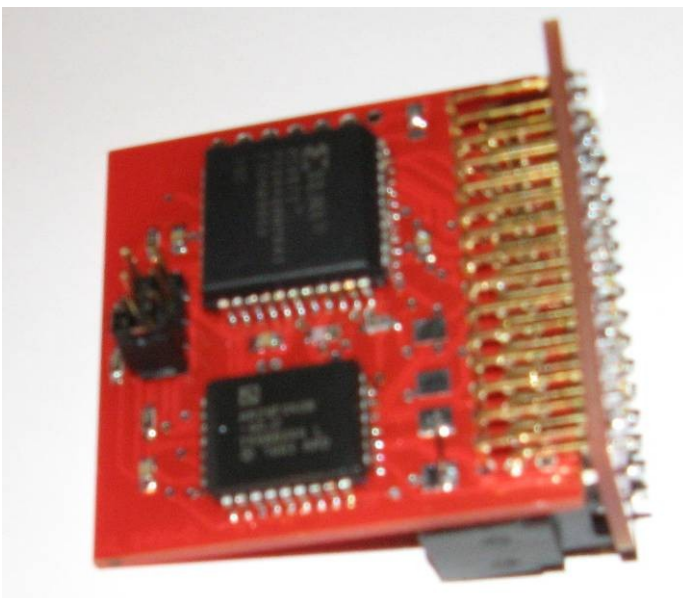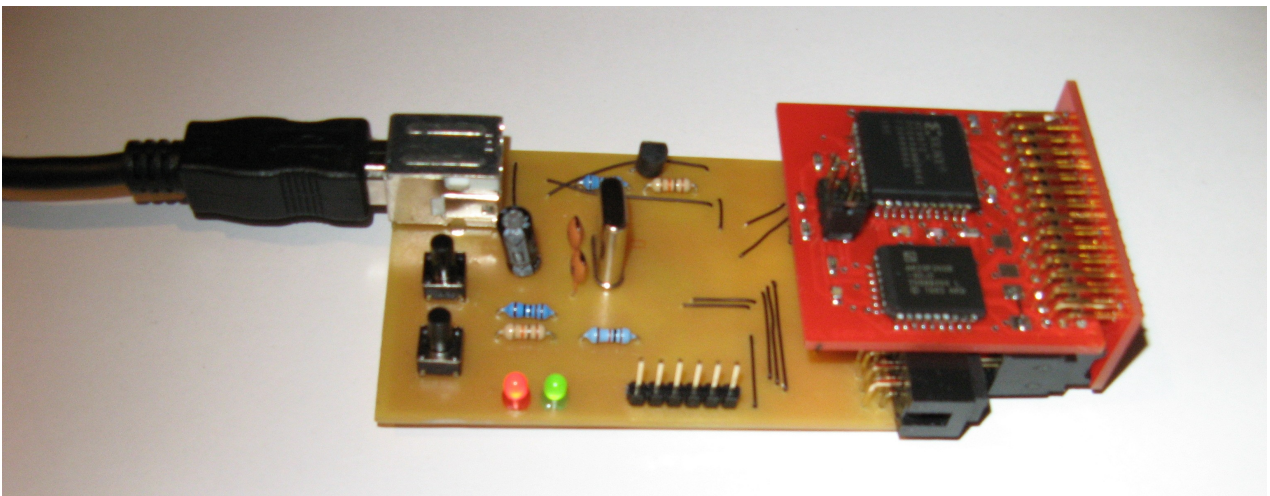
Above: Andrew Tarabaras' J3 Programmer with a J3 chip installed; Left: a J3 chip.

Downloading Binaries

Each piece of hardware has its own operations to follow, so seek the directions specified by the manufacturer. However here is a basic outline on how its done.

Obtaining the binary

When you first open your EEC case, your J3 port will look like this. (complements **frankieh**)

Notice the thermal paste on the PCB, and when you wipe that off, there is hot glue that also needs to be removed. Jaysen has found a paddle pop stick works well for this, but any method you can devise to get the connector as clean as possible will do. If you fail at this step, you will run in to continual problems down the track.

This is what they look like when they have been cleaned.

Plug in your hardware.

Thats about it. Next is to define and edit the bin (which is covered in another docco) and then use your hardware to burn your new bin to a chip. Plug your chip in to your now cleaned J3 port, and go for a squirt!

JA & JB