Speeduino Thoughts – Part One

Since I started out looking for an ECU replacement for my 1985 GL1200 Limited Edition Fuel Injected model motorcycle, I have found there to be no lack of information or product out there.

The reason for my search for an ECU replacement/upgrade is that everything fails in time, and without an ECU my bike is literally a boat anchor. It would be a lot of work to convert the engine from CFI to a carbureted version, and the ECU, LCD dash, and travel computer are all integrated. There are some functions that do not rely on the ECU, but these also do not affect the operation of the engine.

There is no direct aftermarket replacement for an OEM ECU. There are many that come close, but this is mainly because the ECU has been tried and perfected for that specific application, and this is primarily in the auto industry.

Choosing an aftermarket ECU is quite complex and requires significant research to determine applicability. Expertise in electronics, getting to know what you have and need, and cost all figure into your decision making process. My research has found that the best advice is to get to know what you have and learn as much about it as possible, decide on the aftermarket unit you will be using and do the same with it. You must understand the aftermarket ECU requirements – software/firmware/hardware – what each criterion means and how each interacts.

Having mentioned the above, it is by no means beyond the DIY project, but a realistic timeline and implementation schedule is needed.

The motorcycle market is a very small market for aftermarket ECUs. Most motorcycle projects involving an aftermarket ECU are for conversion projects from a carbureted model to using fuel injection. Most only involve fuel and ignition, and some with a turbo installation. This is quite different from my requirement as I mentioned above. I intend to keep as much of the OEM functionality as possible. To do this I will be required to implement a “Full Monty” so to speak in that the aftermarket ECU will be as close to a plug and play unit as possible.

I started by delving into the Megasquirt family of aftermarket ECUs mainly because that is what I found to be prevalent on the Goldwing forums. I did a lot of research into the Megasquirt, how it works, what product would be best for my application and the cost of the various options.

The Megasquirt has been around for some time, is a fairly mature product, and the cost is representative of the features that are required. The software/firmware is mature enough that development is only being looked at for the higher end Megasquirt units. The literature and documentation available is considerable, as are the forums that follow the Megasquirt family. With enough research and digging, you can find answers to just about any question imaginable.

I posted my project on one of the forums I follow, the CX500 Forum. I received a post mentioning the Speeduino project and if I had looked into it. This post was instrumental in the direction I have chosen to proceed in.

Looking into the Speeduino project, I found it to be intriguing, challenging, and of course in my budget range. It is by far the most dynamic ECU project out there. It is a young project, started and based in Australia. It is an open source project with lots of people using/changing the base software/firmware to develop there own applications. Many people have developed and produced their own interface boards to fit the application and space constraints.

The specific aims of this project are:

- Low barrier to entry specifically price and availability of hardware, clear, well documented code, easily accessible software development
- Capabilities/Features
- Simplicity of development

The Speeduino project goal is to keep the project as simple as possible and to delve into as many different area applications as possible. No weird build environments, no knowledge of assembly needed, favour simplicity over performance and keep the entry barrier as inexpensive as possible. This is being achieved by keeping the hardware/software/firmware sides of the system covered under open licenses.

The Speeduino project primarily uses the Arduino Mega 2560 microprocessor, as the “brains” for the ECU unit. There is an interface board attached that connects all the inputs/outputs and ancillary boards necessary for the ECU to operate successfully. The choice of the Arduino microprocessor that you would choose is directly related to your budget, expertise and what you want to accomplish.

There are third party microprocessors (main boards) and accessory boards available for use. One of the third party boards is the NO2C, No Overhang Two Channel, microprocessor. It has a very good feature suite and is good for small spaces.

Another possibility depending on your application is the UA4C Seawolf Sea Fox. This unit has 4 ignition/injection channels.

The documentation for the Speeduino project is not as comprehensive as that for the Megasquirt family. This project has not been around for as long. The onus is on you to do the research necessary to make the Speeduino work for your application. You can find answers to many of your questions through research and perusing on line. The Speeduino WIKI is a must read (several times) and the Speeduino forum is a wealth of information as well.

I have found that a lot of the answers to my questions as to why certain requirement(s) are a certain way are rooted in the auto industry. This is to be expected as the primary reason for the Speeduino project is rooted in the auto industry.

Some issues that have been put forward are using the Speeduino as a “piggyback” unit, keeping the OEM ECU so that the OEM ECU maintains the signals to the ancillary components and Speeduino operates the engine. I have thought of this as a possibility, but feel that a complete replacement will be able to maintain the ancillary units and the readouts. Time will tell with this.

The piggyback unit could be the NO2C unit because it has 2 injector and ignition channels considering that this is exactly what is installed. The issue here is that any additional changes such as going to full sequential ignition or fuel injection would require further expense and considering I live in Canada this is never an inexpensive option. I have viewed YouTube videos where the NO2C board has been used to achieve full sequential ignition/fuel injection. Not for the faint of heart, but can be done.

The Speeduino interface board that I have decided on is the Version 4.0.4. This board is designed primarily to facilitate a plug and play aspect when changing to the Speeduino. There is a version 3 board but it is for a different application, much like the NO2C. Suffice it to say that each board and configuration is for a specific application, and as such, doing your homework is essential.

The version 4.0.4 interface board has a very rich feature architecture. This board supports:

- 16x16 3D fuel and ignition maps, with base of either TPS (Alpha-N) or MAP (Speed Density)
- Supports up to 8 cylinders fuel and ignition with 4 channels of fuel and 4 channels of ignition outputs
- 1, 2 (Even fire only), 3 and 4 cylinder engines with full sequential fuel and ignition
- 6 (even fire only) and 8 cylinder engines are supported with wasted spark and 2 squirts per cycle
- 6x6 3D individual cylinder trim on engines up to 4 cylinders
- After Start Enrichment
- Rev limiting (Spark based, hard and soft)
- Cranking specific enrichment, dwell timing and advance
- General logging through TunerStudio
- High speed tooth logging
- TPS calibration through TunerStudio
- Sensor calibration through TunerStudio (Coolant, IAT and O2)
- Warm Up Enrichment (WUE)
- TPS based acceleration enrichment
- Tacho output
- Fuel pump activation/deactivation (With priming)
- Over dwell and over duty protection
- Battery voltage compensation for dwell and injectors
- Modular wheel decoder support. Included decoders:
- Missing tooth (Eg 36-1, 60-2 etc)
- Dual wheel (Evenly spaced teeth on crank, single tooth on cam)
- Basic distributor
- GM 7X
- GM 24X
- 4g63 aka 4/2
- 'Jeep 2000'
- Audi 135
- Miata 99-05
- Honda D17 (12+1)
- Nissan 360
- Subaru 6/7
- Taking requests...
- Open and closed loop idle control (PWM and Stepper)
- Closed loop boost control
- Open loop VVT control
- Deceleration fuel cut off (DFCO)
- Launch control
- Flex fuel
- O2 based autotune (Registered version of TunerStudio required)

This is a very comprehensive list of features that will allow a person to start small and progress with changes as the project develops. As I have mentioned, I intend to do an ECU upgrade/replacement that will be as close to plug and play as possible.

Up to this point, I have been researching the CFI system of my motorcycle to get an understanding of what the individual components do and how each interacts with the system. I feel I have achieved this and in doing so, have continued onto the next step which is to determine how the OEM CFI system can interface with an aftermarket ECU.

This has been a serious learning curve. There is no aftermarket ECU system out there that compares to what Honda achieved in the mid 1980s. Honda's CFI system design is the forerunner of its PGM-FI system that has not been significantly changed.

The CFI system on my 1985 GL1200 Limited Edition Fuel Injected motorcycle is Honda's third iteration of this system. This system is used in the 1986 GL1200 Special Edition – Injected motorcycle as well.

The OEM CFI system was first implemented on the 1982 CX500 turbo motorcycle with the second iteration on the 1983 CX650 turbo. Each iteration received some refinements depending on the application. In this respect my motorcycle does not have a turbo application so the OEM ECU programming was not used.

Another interesting feature that Honda included in its FI motorcycles OEM ECU had a diagnostic program and a light indication system for CFI system error messages. This is in my opinion the forerunner to the OBD units that are used today. With this feature, Honda incorporated a “Fuel System” indicator light on the LCD dash that is the equivalent of the check engine light in more modern FI applications. The use of a “Fuel System” warning light is good; however, it is a bit misleading because it has nothing to do with the fuel system.

The CFI system of the 1985 and 1986 Goldwing FI motorcycles interfaces with the travel computer and LCD dash. These two components are reliant on the ECU for engine data to facilitate a lot of the various travel computer and LCD dash features.

The information/documentation regarding the integration of the various components, as well as the ECU programming, is not readily available or even available. In this regard, you have to be your own “Sherlock Holmes” and do considerable research to find your answers, or test the various components to get the answers.

It is unfortunate that the expense of FI and the development of better carburation on motorcycles put the brakes on further Goldwing FI installations. My reading has found that Honda did not revisit FI on motorcycles until the late 1990s and for the Goldwing, not until the GL1800 in 2001. The resurgence of FI on Honda motorcycles is mostly due to emission standards, motorcycles tend to follow a close second to the auto industry.

I have had to learn about trigger wheels, square wave versus sine waves, coolant/air temp sensors, IAC system impacts, O2 sensors (won't find one on the early FI systems), flood clear levels (don't know if Honda included this feature), injector latency (dead time), coil dwell times, spark times, and a host of other considerations.

The CFI system on my motorcycle has a crank sensor with an 8 tooth trigger wheel (no missing teeth) and two cam sensors. There is no aftermarket ECU unit that uses two cam sensors. Having an 8 tooth trigger wheel can be used, but a 12 or more toothed wheel is recommended. This gives more signal pulses to the ECU for better operation. You must ensure that the number of teeth on the trigger wheel evenly divides into 720. A trigger wheel with more than 8 teeth will also provide a better number in degrees for the tuning software.

This only scratches the surface of the information and understanding you must get to achieve success with a project such as this. Another aspect of this project and any project dealing with converting to FI is to give yourself lots of time. I have given myself a year to learn about the in and outs of the Speeduino project and the various components of the OEM CFI system before I implement it on my 1985 Goldwing.

There are quite a few items/components that need to be purchase to support my project such as a wideband O2 sensor, the Speeduino ECU components, learning about the tuning software, loading the applicable software/firmware, and others. If you are doing a conversion, there are other requirements such as finding FI components and installing these, no simple task, and takes a lot of dedication.

The Speeduino project uses TunerStudio as a tuning interface. It operates on the various software platforms and provides configuration, tuning and logging capabilities. There is a free version of TunerStudio, but it is recommended to purchase a licence to use all the features of TunerStudio. Purchasing a licence also supports the development of this invaluable tuning software package.

MegaLog Viewer is another recommended software program to have.

I must mention at this time that TunerStudio is also used with the Megasquirt family of ECUs. There are similarities, but overall, what is done for Megasquirt ECUs may or may not be applicable to the Speeduino project.

A good example of this is the Cranking Settings for the Speeduino project are completely different from those required for the Megasquirt ECUs. Definitions, calculations, and other such requirements may be used interchangeably, but again, caution is advised. Do your homework first to confirm if what is being used for the Megasquirt ECU is applicable to the Speeduino project.

Having the correct investigative tools will assist in getting your project up and operating successfully. A tach/dwell meter, timing light with an integral timing advance feature, inexpensive oscilloscope for sensor signals, possibly a logic analyzer as well. These and other test equipment can make your project and research much easier. A lot of this test equipment is relatively inexpensive and can be found on Amazon, eBay, and Ali Express to mention a few sites.

My journey so far has been quite interesting a good learning experience. There is a lot of information available you just have to apprise yourself of it and make sure it is applicable to your project.