E491 - Project Proposal

Fully Programmable Engine Control Unit

David Tolsma

November 7, 2019

1 Abstract

The ECU is an integral part of a modern engine system. It is responsible for taking in sensor data from many places in the vehicle including air temperature, engine temperature, manifold pressure, oxygen sensor data, crank angle and various other inputs, processing the data, then determining the correct amount of fuel to inject, when to trigger a spark, as well as controlling other critical engine components.

This project aims to create a fully programmable engine controller for four-cycle automotive and small engine applications.

2 Project Users, Problem Definition, Goals, and Features

Many commercial after-market ECUs exist such as MegaSquirt, LinkECU, or Haltec. These products tend to be very expensive costing excess of $1000 and are closed source, preventing user customization or modifications to the hardware or software. In response to these issues, several open source projects have been started with only a handful being in active development and use. Currently these projects include Speeduino and rusEFI. Both projects each have around 30-50 known good installations, are in current development, and are feature rich.

Speeduino was developed to be a shield for an Arduino Mega, and in light of that design choice is limited by lack of processing power, advanced timer and communications peripherals (such as CAN), and limited IO. The project does not anticipate moving to another platform in the future, so while it is a great platform for basic engine management, it has limited room to grow.

rusEFI has taken a different design methodology and rather than using an aging ATMEGA platform has opted for a much more modern ARM based processor. This processor has much more processing power, much better peripheral support, and has enough resources to move from a traditional programming scheme to a real time operating system which allows for much easier management of the large number of sensors, outputs, and communications protocols that the ECU needs to manage. While this project has moved to a more modern platform and has a large number of software features, it is still using a development board and shield configuration and has a slightly under-featured hardware component.

This project aims to take inspiration from both projects and create a standalone ECU that helps fill in some of the gaps that both of these projects leave open. Neither projects have an integrated knock sensor or an integrated wide-band O2 sensor support, but both rely on after thought add-on boards. One of the major goals in this project is to bring all of the hardware necessary for engine control onto a single board.

Goals for this project include integrating a wide-band O2 controller, support for four cylinder injection and ignition, an integrated pre-ignition (knock) sensor input, manifold absolute pressure (MAP) inputs, CAN bus inputs and outputs, as well as all the standard needs such as general purpose high current outputs for devices such as fans and fuel pumps.

3 Project Constraints

Due to being in an automotive environment, the parts choices are limited to those that are rated for appropriate temperature ranges as well as being robust enough to survive in the electrically noisy automotive electrical system. In order to replace an existing stock ECU the size of the project must be properly housed in an enclosure that is small enough to be a drop in replacement for a stock controller.

The most significant constraint on the project is time. Many of the desired features will have dedicated hardware components such as an integrated knock sensor, which will in turn need firmware written to control it. The software component of this project will be quite involved, and getting the software to the point of even a very basic engine management may take a majority of the time allotted to software development. Many software features such as "throttle tip in" (a feature where extra fuel is added when the throttle is rapidly opened) can significantly improve engine response when the accelerator is rapidly depressed. This feature is not necessary for very basic engine management, but is an essential feature for a finished product. Due to time restraints, many of these more advanced control algorithms may need to be placed on a stretch goal list rather than making them essential features to complete. A basic and testable project is the end goal by the end of the year, with additional hardware and software features to be integrated afterward.

4 Project Responsibilities

Many automotive systems are safety critical and the drive-train system is no exception. Creating a project such as this that is both controlling an expensive piece of hardware as well as keeping the operator out of harms way, there is a lot of responsibility to create a project with as little risk as possible. All of the parts in in the design should be rated for the high and low operating temperatures that are needed in an automotive setting, as well as having robust input and output protection in order to insure that that noise or issues elsewhere in the vehicle do not cause the ECU to fail or malfunction. The need for a safe system also forces the quality of the code to be high with very good error handling systems to ensure as safe of operation as possible.

5 Bibliography

[1] Speeduino.com, 2019. [Online].
Available: https://speeduino.com/wiki/index.php/Speeduino. [Accessed: 25- Oct- 2019]

[2] Rusefi.com, 2019. [Online].
Available: https://rusefi.com/. [Accessed: 25- Oct- 2019]