[1]

IKU Electronic Control Unit

Takuya Seaver, EE, Jack Walter, CSE, Yongjie Yang, EE, and Xueteng Qian, CSE

***Abstract*—As global temperatures are on the rise the need for new improvements in the way we travel increase. In order to achieve the goal of halting the effects of global warming, the IKU electronic control unit tests the limits of how many miles a vehicle can achieve on one gallon of gasoline. This device consists of two main components. The first component being the hardware that will be connected to the engine where it will convert information of one type to another, so that the other component, the software can process the data. The software will then send a command signal to the fuel injector. In the end, the fuel injector will spray fuel into the engine for the correct amount of time so that the engine can efficiently use its fuel source and get the most miles out of it.**

I. INTRODUCTION

A. *Significance*

Global warming is a topic that is in the news, on the internet, and affects the entire world. The most abundant greenhouse gas in our atmosphere is CO2[2]. The largest source of these rising CO2 emissions is in the transportation industry[3]. Until the electric car is made more attainable for people then lowering the amount of CO2 emissions created by the transportation industry is an intermediary solution.

B. *Context and Existing Products*

The UMass Super Mileage Vehicle team or SMV team for short, is a team of mechanical engineering students that test the limits of the modern-day combustion engine. The number of miles that can be brought out by one of these engines is mainly determined by the efficacy of the vehicle's method of mixing fuel and air to produce the optimal fuel-to-air ratio. Several options are available on the market today such as an off-the-shelf electronic control unit. Another option is to use a carburetor which mixes air and fuel to produce the desired ratio. These methods have been tried before, but what they lack is customization. Our solution differs in the fact that it allows for the user to modify the device so that it can be tuned to your exact needs. As for the SMV team, their vehicle can have a fully tunable Electronic Control Unit (ECU) to get the most miles per gallon.

C. *Societal Impacts*

The SMV team would be our main use case. They need an ECU that can be modified, so that they can perform better at the yearly competition that they compete in. This product could positively affect them by performing better than their current off-the-shelf ECU. One downside to our product is that it does not eliminate CO2 emissions. Our product will still be the control system for a fuel injection system that burns fossil fuels. There will still be some sort of CO2 emissions, even though not as much. This is seen as only an intermediary step to achieving less carbon emissions in our atmosphere.

D. *Requirements Analysis and Specifications*

We were given some details as to the requirements of the design. The system needed to be tunable and customizable. For this, there needs to be documentation provided for the modifications to be made to the tables that the data is stored in. This data is to be modified using information that is observed when the O2 sensor is mounted. The system needs to be able to take in information from the O2 sensor when needed for tuning. Safety is also an important factor. If the system fails and is not injecting fuel correctly then the system needs to be able to be turned off manually. Also, since the processes within the engine happen very fast then our system needs to be able to keep up. The requirements and specifications are listed below in Table 1.

Requirement	**Specification**	**Value**
Tuning	O2 Sensor Mountable	N/A
Responsive	Duration	<=10ms
Safety	Switch	ON/OFF
Frequency	RPM Limit	Up to 6000RPM
Modifiable	Modify Tables	Instructional Documentation

Table 1: Requirements and Specifications

II. DESIGN

A. *Overview*

As an intermediary solution, an Electronic Control Unit (ECU) is a good way to solve the climate change issue. An ECU is an integral part of the throttle body fuel injection engine, because it can easily provide the accurate amount of fuel and air into the engine that increases the power. Besides, a carburetor is also a device that can mix air and fuel to inject the engine (*Appendix A*).

Thus, we design an ECU. See the ECU block diagram in Figure 1. Firstly, we synthesized input signals because we have to simulate the signals from sensors. For the car, an ECU can read signals from the crankshaft sensor, O2 sensor, manifold air pressure sensor and intake air temperature sensor. After reading the input signals, then we have to design our ECU develop the software by using a microcontroller, in this case an ATtiny817. Besides this, through an integrated circuit level shifter and

[1]

T. Seaver from Plymouth, MA (e-mail: tseaver@umass.edu).
J. Walter from Westwood, MA (e-mail: jackwalter@umass.edu).
Y. Yang from Linzi, China (e-mail: yongjieyang@umass.edu).
X. Qian from Beijing, China (e-mail: xqian@umass.edu).

[2] See [1] in the References Section

[3] See [2] in the References Section

driver, the control signal from the MCU can precisely control the fuel injector by accurate pulse width.

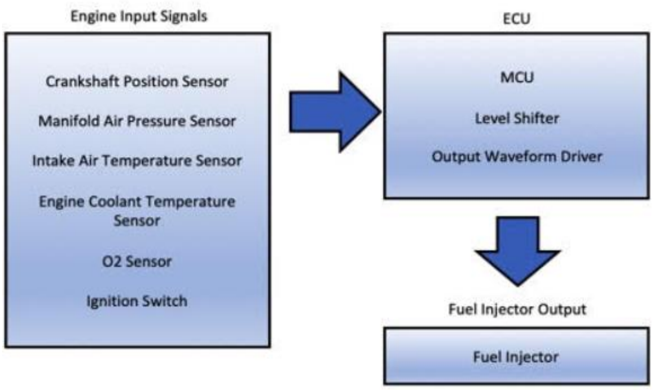

Figure 1: Block Diagram of ECU

For specifications in block 1, the ECU system should get a real-time processing of input data signals for fuel injection. And the system also should be able to get the feedback capabilities with the O2 sensor for optimization. In block 2, the microcontroller (MCU) can read signals supporting up to 6000 RPMs and able to have a correct pulse width to inject the correct amount of fuel within ~10ms for the specification. In addition, eventually, the hardware part of the level shifter and output waveform driver can easily and precisely be ready to drive a fuel injector by the control signal from the MCU.

B. Engine Input Signals

Looking at Figure 1, the first big block is about simulated signals from four kinds of sensors and one switch. The crankshaft sensor measures the rotation speed (RPMs); the manifold air pressure (MAP) sensor measures the air density and air mass flow rate in the engine; the intake air temperature sensor monitors the temperature of the air entering the engine; the O2 sensor monitors how much unburned oxygen exits from engine. In addition, the ignition switch is a mechanical switch to control the open and close function of the engine to inject fuel. So, when we synthesized the signals, we used the Arduino Uno r3 to generate the input signals. To be specific, we also used potentiometers which are connected to the Arduino to modify the speed. From this part, we learned how to simulate the signals and how to write the Arduino code from what we learned in the course. Then we used an oscilloscope to watch the input signals when we changed the potentiometers for testing. *Details see Appendix B.*

C. MCU

Lookup Table

	LOAD							
RPM	0	20	25	30	40	60	80	100
1500	3.2	3.2	3.2	3.2	0	0	0	0
2000	3.4	4.8	5.2	6	7.6	7.5	7.5	7.5
2500	3.2	3.5	1.8	2.1	6	6.5	7.2	7.5
3000	3.2	5.2	4	3.5	4.8	5.5	6.5	7.2
3500	3.2	5.5	6.4	4.5	5.2	5.5	6.8	7.2
4000	3.2	4.4	4.7	5.1	6	6.2	6.9	7.2
4500	3.2	4.5	5	5	6.4	6.6	6.9	7.2
5000	3.2	4.7	5.1	5.3	7	7.1	7.2	7.3
5500	3.2	3.2	3.2	5.5	6.6	7	7.2	7.4
6000	3.2	3.2	3.2	5.4	6.2	6.8	7.4	7.8

Figure 2: Lookup Table of MCU

Looking at Figure 1, there is a microcontroller (MCU) in the ECU. One of the most important parts in ECU is to develop the software in our ATtiny817 microcontroller. First of all, looking at Figure 2, we have to consider the mathematics function of the relationships between the data of the sensors to find the output pulse width in the lookup table. Then we write the C code by using the Atmel Studio 7 IDE. Then the software can read the synthesized signals from the input and give a correct pulse width of the output. So, we accepted the coding idea from our embedded systems class. It really helps us to experience and understand the programming for the microcontroller. For testing this part, we also used an oscilloscope to watch the output signals when we changed the potentiometers. *Details see Appendix B.*

D. Level Shifter and Driver

Looking at Figure 1, there are two parts about level shifter and output waveform driver. The function of level shifter and driver is reading the control signal from MCU and giving the correct reverse pulse width to control the fuel injector, which is supported by a 12V power supply. It means when the voltage of the control signal is LOW(0V), the injector will inject fuel, and when the control signal voltage is HIGH (12V), then the injector will be closed and not inject fuel. We learned the method to design integrated circuits from our electronics course and this project provides an interesting and skilled experience for us. Especially for driving an inductive load, which is the internal circuit of fuel injector. Then for testing this part, we used an oscilloscope to compare the input and output pulse width to see whether or not it is reversed and has the same width. Also, connecting the whole thing to a fuel injector to see if the integration of the whole system is working correctly.

III. Project Management

Referring to Table 2 below.

Goal	**Status**
Correctly Synthesize Input Signals	Accomplished
Calculate Pulse Width Time Based on Input Data Calculations and Lookup	Partly Accomplished

Table Values	
Inject the Correct Amount of Fuel	In Progress

Table 2: MDR Goals

There were several goals set in place for MDR. Some of which were not accomplished. Synthesizing input signals to mimic those of an engine was accomplished. Calculating pulse width has not been functioning correctly, yet we have a lookup table and are able to use our input data for our equations even though we have not been able to get a correct pulse width signal. Since our signal is not correct then the correct amount of "fuel" is not being injected. Until we figure out the issue and are able to get a correct output from our microcontroller then these last two goals are not fully accomplished. That will be fixed in the coming weeks, so that we are still on schedule.

Looking at Figure 3, we can see what our upcoming schedule looks like. Before we begin to integrate real engine components, we will need to look at our current issues and make sure that our system works for basic functionality. Once the problems are fixed then it will be on to real engine component integration which will be led by Takuya. Simultaneously the refinement of our code to be able to coincide with our faster moving input signals will be worked on by Jack and Xueteng. As these two processes are being figured out, we will begin to design our PCB which will be the responsibility of Yongjie. In the next block we will be analysis of power consumption which will be worked on by Takuya. Analysis of our algorithms which will be part of the software teams' responsibilities. Also, we hope to get a PCB prototype printed by this time, so we can get ready to test it on some real engine components. The next few task items will be worked on by the whole team. As a team, we will need to go over the best course of action for optimization of our system. What can be worked on? What works well enough, so that we can focus on items that don't work as well? These are the types of questions we will be thinking of when it comes time to make this decision. The next few steps are to go along with our goal of providing documentation for our system. We believe that proper documentation is good practice, but also better for our consumer. The better the documentation that we can provide the better they will know the product. Finally, we will be working on our CDR presentation. This is important to add into the schedule so that we aren't rushing ourselves in the end to get a presentation together to present our progress. We don't have any plans to change how our team is organized now. Things seem to be running smoothly overall. For a further description on how our team works see Appendix C.

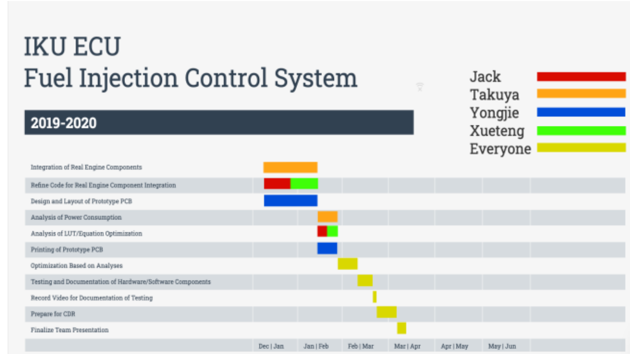

Figure 3: Gantt Chart

IV. Conclusion

The current state of the project is good. There are some key things to focus on soon. Getting the system to work for basic functionality is of the utmost importance. Once that is settled then we can focus on engine component integration. That will provide a lot of work for us to do. Going from perfectly working synthesized signals to somewhat unstable and possibly unreliable engine components could become problematic if enough time isn't allocated to troubleshooting any issues that may arise. To achieve our goal of using actual engine components we will need a large amount of time dedicated to it. Asking questions to the SMV team will also be important since none of us have worked with these parts before. Time is of the essence and to be where we want to be as a group then near-perfect time management needs to be the focus in the future.

Acknowledgment

As a group, we would like to thank our advisor, Prof. Xia, for working around our schedule and meeting with us when needed. Also, we would like to thank our evaluators, Prof. Hollot and Prof. Gong, for providing good constructive criticism. All points were made clear in their feedback and all were taken into consideration.

References

[1] "Inventory of U.S. Greenhouse Gas Emissions and Sinks," *EPA*, 11-Apr-2019. [Online]. Available: https://www.epa.gov/ghgemissions/inventory-us-greenhouse-gas-emissions-and-sinks. [Accessed: 17-Dec-2019].

[2] "Sources of Greenhouse Gas Emissions," *EPA*, 13-Sep-2019. [Online]. Available: https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions. [Accessed: 17-Dec-2019].

Appendix

A. *Design Alternatives*

Compared to the ECU, carburetors were used because they are inexpensive. However, the ECU has special valves which can electronically control the open and close state of the fuel injector, and the carburetors don't have this functionality. In addition, an important feature of ECU is that it can read signals through mechanical sensors. Sensors ensure that the exact amount of fuel is injected at a given time.

B. Testing Methods

Until now, we made progress on the basic functionality of the ECU system. Then we must test each part whether it can successfully work by itself. We mainly used an oscilloscope to watch the signals to check them.

For the engine input signals part, when we changed the resistance of the potentiometers, there is a changed wave showed on the oscilloscope.

And for the MCU part, we also used an oscilloscope to watch and analyze the pulse width of the different resistance of potentiometers. The wave should be a rectangular wave and the pulse width around about 10ms. In addition, we tried to use some data to see whether they are matched with the lookup table.

Last but not least, it was testing the level shifter and driver. Without other parts in the ECU, we used the function generator to build a simulated rectangular signal at 6000RPM (100Hz) and the high voltage being 3.3V to synthesize the output signal from our MCU. Then we checked the output signal of the level shifter and driver which looks like a rectangular wave with a high voltage of 12V, and it has the correct pulse width. This is the same as the input.

In conclusion, to test the whole system, we were connecting to a fuel injector. And the fuel injector will be injecting compressed air. Besides this, we also must watch the output pulse width of the fuel injector. We put a finger on the end of the fuel injector so that we could feel the intermittent air flow.

C. Team Organization

Our team is organized well. We meet every week with our advisor. We also meet throughout the week as a group to talk about what we've worked on or what we have planned. Attendance at our weekly meetings is great. It is very rare that we do not have everyone in attendance. Our expertise is split up from our majors. The two electrical engineering majors, Takuya and Yongjie, work as the hardware team. The two computer engineering majors, Jack and Xueteng, work as the software team. We tend to stick to our expertise, but since we all have a baseline knowledge of each other's work then we are never afraid to help each other out. We communicate through email, online messaging, and in person. Sometimes when communicating through email or online messaging isn't working then we will schedule a video call to get our points across without the obscurity of text messages. Overall, we help each other and get along well. There has not been a single time in which any of us have felt that our opinion or advice isn't wanted. Whenever one of us has some sort of idea or suggestion then we speak up and talk through it.

D. Beyond the Classroom

Takuya – Besides technical skills that I have needed to develop I think that the most important skills that I have applied from my professional life would be from a management perspective. I have used many of the techniques that I have seen in my professional life to make sure that we as a team have good communication and clear goals in mind. As for design, many resources have given me ideas as to how to approach the hardware problem. From coworkers at my internship to online resources, many have given me direction as to how to approach an engineering problem. I am always relating the issues that come up within our project to similar issues from my internship experience. Many of the issues with creating a product from the research and development phase is like what we are trying to accomplish within our project. It has all been very informative, especially from relating our issues to that of the engineering industry.

Jack – One of the main skills I have had to develop with this project is working with embedded systems. Most of the work I did was developing the software with Xueteng for our ATtiny817 microcontroller. We developed this software in C using Atmel Studio 7 IDE. I have had to become a lot more comfortable with reading data sheets for the microcontroller as well as understanding microcontroller concepts. Atmel Start's website has been very helpful as it has many examples of projects that help you get started with the ATtiny817. I have been drawing a lot of my experience in Computer Science Lab I & II as I worked with embedded systems in those classes. I can see a connection with this project and my life as a professional. I really enjoy programming and want to pursue that after I graduate; however, I don't know if I see myself working on embedded systems specifically.

Yongjie – I have obtained a lot of knowledge from our ECU project. As a hardware circuit designer, I mainly designed a circuit to amplify and modify a signal by using the topology circuit. For this project, I successfully designed a level shifter and driver, whose function is to let the ECU system drive the fuel injector. To design this part, I did a lot of research and changed different kinds of transistors repeatedly to optimize the circuit and minimize the noise signal as much as possible to get a precise output control signal from the ECU. In fact, each integrated circuit design potentially enhances my mastery of hardware knowledge. Furthermore, I believe in the future of the professional career life, derived from this project, I can develop more technical skills in experiencing this kind of teamwork.

Xueteng – For this project, I mainly did the ECU programming part with Jack. We used the ATtiny817 and programmed in C. The ability to debug is very important, because we have encountered lots of problems each time, we add a new function to the whole program. After MDR, we improved our debugging skills and we can kind of predict problems that might appear after we add new functions. The datasheet is one of the most important resources because it teaches us the functionality of each part and how to use them, and another good source is the sample code from the Atmel start website because we need lots of basic setup codes and it saves us lots of time. This project taught me that if I want to build something from scratch, I need to be patient and walk step by step from designing, building blocks, and assembling. Every step could have tons of questions, and the only solution is to face the problems at hand and solve them one at a time.