Electronic Throttle Control for the University of Missouri FSAE Formula Car

Gage Crowder

Prepared for

ECE 4220 – Real Time Embedded Computing, University of Missouri

April 1, 2015

Abstract – The proposed project is electronic throttle control for the University of Missouri FSAE formula car. Having electronic throttle control increases responsiveness of the throttle by eliminating the delay caused by mechanical linkages. In addition it reduces the overall weight of the vehicle and decreases the number of mechanical service visits [1]. This project provides that electronic solution. The project consists of a control unit, various sensors, and a throttle actuator. The control unit monitors sensors that relay back information about the engine as well as driver input. The control unit then provides control signals to various parts of the car to accurately control acceleration.

Index Terms – Drive-by-wire, electronic throttle, electronic throttle control, FSAE ETC, vehicle driving, vehicle dynamics.

Introduction

The University of Missouri formula car team builds a scale race car every year and enters into a design and racing competition [2]. However, to stay competitive the car must be as advanced as possible. This includes aerodynamics, power output, electronics, and control [2]. With that in mind an electronic throttle control system is a great addition to next year's car. This adds one more system to the car to help gain valuable points towards the teams overall score.

Electronic throttle control differs from that of a mechanical control system; the throttle is monitored and controlled by an embedded system as opposed to mechanical linkages. According to [1] this has many benefits over the conventional system including: reducing the number of moving parts, reducing overall weight, increasing operational accuracy, and a decrease in the number of service visits for mechanical service. These are all beneficial design considerations when designing the car.

By implementing this project, an embedded system along with a small number of hardware devices, the formula car attains all the advantages mentioned above. There is currently no system that the team has previously used or designed so this project will be the first of its kind for the team. The short term goals for this project are to have all the hardware interfacing operational, and to meet as many of the time constraints as possible. The long term goal of this project is to have the system fully operational to meet the FSAE design standards to allow use of the project in next year's car. I expect that these goals are both realistic and attainable.

Proposed Implementation and Specifications

The following section includes figures and descriptions of my implementation as well as system specifications.

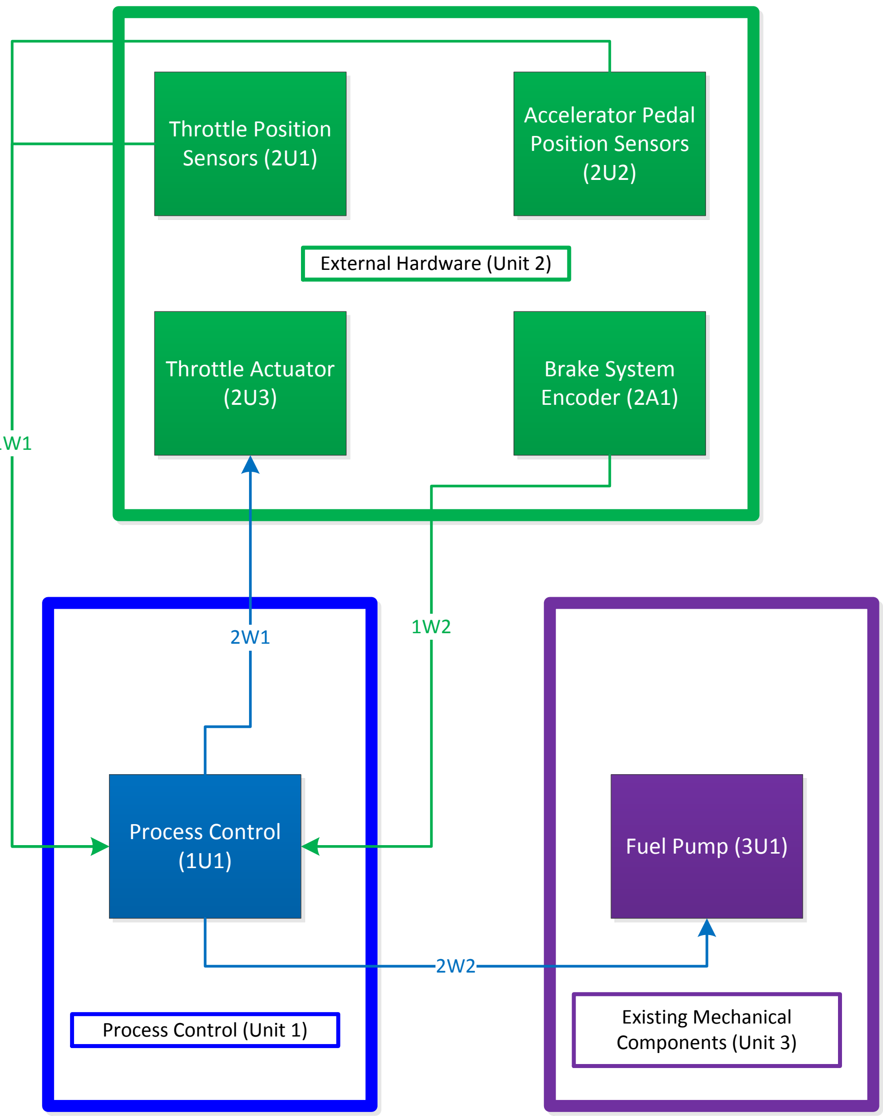

Figure 1: Functional block diagram for the electronic throttle control

Functional Blocks:

The following describes the functional blocks as shown in *Figure 1: Functional block diagram for the electronic throttle control*. Some design constraints and specifications were taken from [2] to ensure it meets minimum safe operating requirements for use by the FSAE team.

Process Control (Unit 1)

The following describes the functional blocks for *Process Control (Unit 1)*.

1U1 – Process Control

The process control unit is an Arduino Uno powered by the existing 12V battery on the car. It is responsible for monitoring all the external sensors as well as providing a control signal to the throttle actuator and fuel pump. The process control unit will cut power to *2U3 – Throttle Actuator* and *3U1 – Fuel Pump* as defined under *2A1 – Brake System Encoder*. And power to *2U3 – Throttle Actuator* will be cut as defined under *2U1 – Throttle Position Sensors* and *2U2 – Accelerator Pedal Position Sensors*.

External Hardware (Unit 2)

The following describes the functional blocks for *External Hardware (Unit 2)*.

2U1 – Throttle Position Sensors

The throttle position sensors consist of two individual sensors mounted to the throttle body of the car's engine. It communicates with *1U1 – Process Control* via an analog signal.

2A1 – Brake System Encoder

The brake system encoder is a physical hardware device already on the car. It communicates with *1U1 – Process Control* via a digital signal. The normal operating voltage range for the sensors is .5V – 4.5V. If the signal goes outside the acceptable range of values power to the electronic throttle control will be shut down immediately. Power to *2U3 – Throttle Actuator* and *3U1 – Fuel Pump* will be shutdown given any of the following: the mechanical brakes are actuated and *2U1 – Throttle Position Sensors* indicate the throttle is open by more than a permitted amount. This will occur within 1sec. If the error between the indicated and target throttle position vary by more than 10% for more than 1sec. This will also occur within 1sec. Or, during periods of heavy braking where the throttle is open by more than 10%.

2U2 – Accelerator Pedal Position Sensors

The accelerator pedal position sensors are actuated by a foot pedal. There are two separate pedal position sensors. If there is a difference of more than 10% the power to the electronic throttle will be shut down immediately. The pedal position sensors communicate with *1U1 – Process Control* via an analog signal. The normal operating voltage range for the sensors is .5V – 4.5V. If the signal goes outside the acceptable range of values power to the electronic throttle control will be shut down immediately.

2U3 – Throttle Actuator

The throttle actuator is a small servo motor mounted to the throttle body of the car's engine. It is responsible for opening and closing the throttle to allow more or less fuel and air into the engine. It communicates with *1U1 – Process Control* via a digital PWM signal.

Existing Mechanical Components (Unit 3)

The following describes the functional blocks for *Existing Mechanical Components (Unit* 3).

3U1 – Fuel Pump

The fuel pump is a mechanical device already installed on the car and provides fuel to the engine. Power to the fuel pump is controlled by *1U1 – Process Control*. Power to the fuel pump will be interrupted within .5sec during hard braking situations as dictated by *2A1 – Brake System Encoder*. It communicates with *1U1 – Process Control* via a digital control signal.

Signals:

The following describes the signal paths as shown in *Figure 1: Functional block diagram for the electronic throttle control*.

Process Control (Unit 1)

The following describes the signal paths for *Process Control (Unit 1)*.

1W1

This is a wired analog input signal from *2U1 – Throttle Position Sensors* and *2U2 – Accelerator Pedal Position Sensors* to *1U1 – Process Control*. This allows the process control to monitor the current throttle position and the current accelerator pedal positon.

1W2

This is a wired digital input signal from *2A1 – Brake System Encoder* to *1U1 – Process Control*. It monitors the brake system encoder for error states that require cutting power to one or more assemblies.

External Hardware (Unit 2)

The following describes the signal paths for *External Hardware (Unit 2)*.

2W1

This is a wired digital PWM output signal from *1U1 – Process Control* to *2U3 – Throttle Actuator*. This controls where the throttle should be positioned.

2W2

This is a wired digital output signal from *1U1 – Process Control* to *3U1 – Fuel Pump*. This controls the power to the fuel pump.

Software Architecture:

The following section includes figures and descriptions of the software architecture.

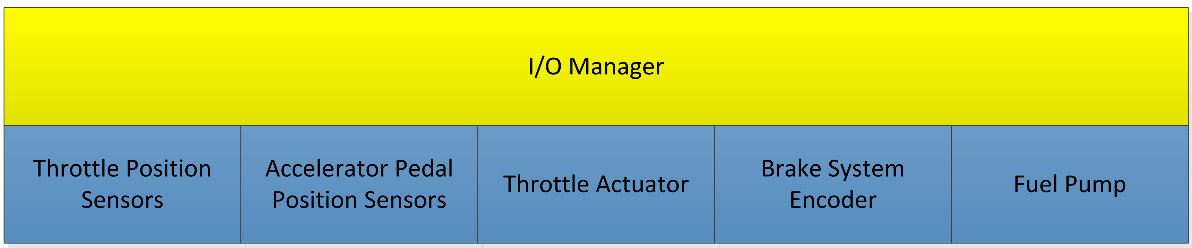

Figure 2: Software architecture, technology stack diagram

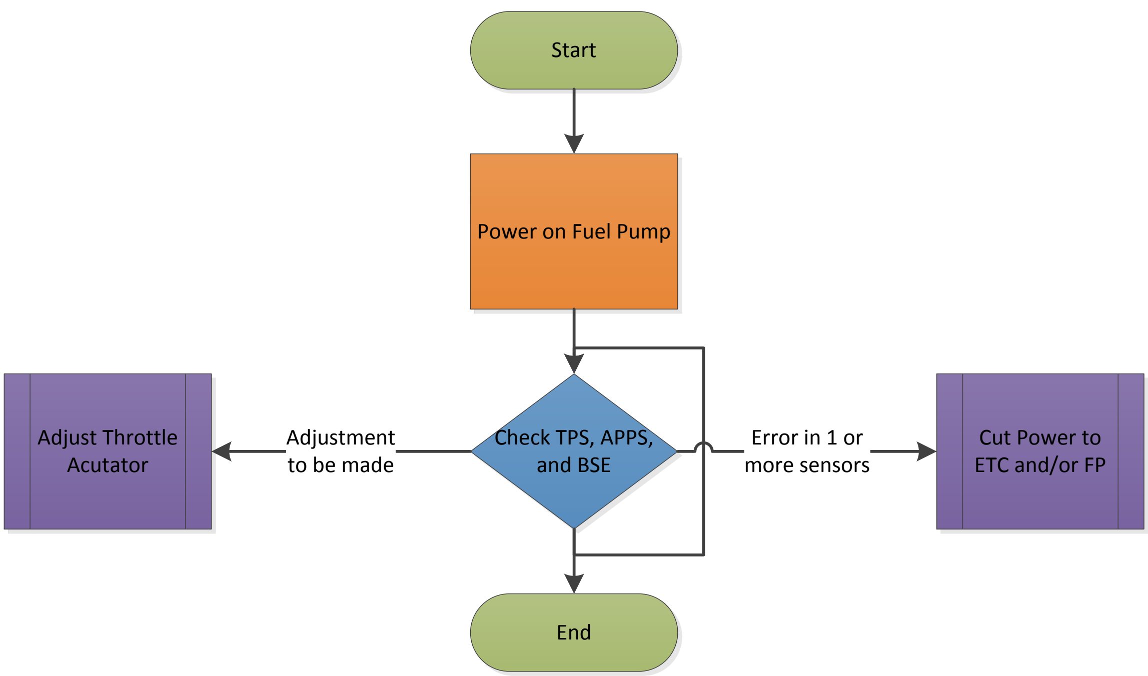

Figure 3: Software flow diagram

I/O Manager

The I/O manager controls the input and output of *Process Control (Unit 1)* to ensure smooth and consistent operation. It serves as control to the individual device drivers. It takes inputs from *2U1 – Throttle Position Sensors*, *2A1 – Brake System Encoder*, and *2U2 – Accelerator Pedal Position Sensors*, and provides output to *2U3 – Throttle Actuator* and *3U1 – Fuel Pump*.

Throttle Position Sensors

This is a device driver that handles the input from *2U1 – Throttle Position Sensors* and provides output to the *I/O Manager*. This communicates the current position of the throttle back to the process control. This requires a clear signal from *2U1 – Throttle Position Sensors* to *1U1 – Process Control*.

Accelerator Pedal Position Sensors

This is a device driver that handles the input from *2U2 – Accelerator Pedal Position Sensors* and provides output to the *I/O Manager*. This communicates the current position of the throttle pedal back to the process control. This requires a clear signal from *2U2 – Accelerator Pedal Position Sensors* to *1U1 – Process Control*.

Throttle Actuator

This is a device driver that handles the output to *2U3 – Throttle Actuator*. This provides the digital output that controls the servo (throttle) to move to the correct position. This requires a clear signal from *1U1 – Process Control* to *2U3 – Throttle Actuator*.

Brake System Encoder

This is a device driver that handles the input from *2A1 – Brake System Encoder* and provides output to the *I/O Manager*. This monitors input from the brake encoder to check if power to one or more assemblies should be switched off. This requires a clear signal from *2A1 – Brake System Encoder* to *1U1 – Process Control*.

Fuel Pump

This is a device driver that handles the output to *3U1 – Fuel Pump*. This provides the digital output that controls power to the fuel pump. This requires a clear signal from *1U1 – Process Control* to *3U1 – Fuel Pump*.

Color Scheme

The following outlines the color code used in *Figure 2: Software architecture, technology stack diagram*.

Yellow – Process management

Blue – Device Drivers

Class Relevance

This section includes a breakdown of different course concepts utilized by the project.

Interfacing with Hardware

Foremost the project incorporates interfacing with hardware. There are multiple sensors and motors that are being monitored and controlled. These all are checked and controlled in an efficient manner. Both analog and digital signals are used for the various inputs and outputs.

Embedded System Design

This project utilizes an Arduino Uno as the embedded computing device. Given the nature of the project a full embedded computer was not needed, but rather a microcontroller is the better option. While this does have its apparent drawbacks, limited CPU power, limited operating systems, etc., implementing the project with a microcontroller is the better approach. By referencing *Proposed Implementation and Specifications* it is clear a thorough amount of design work went into this project.

Interrupt Driven Programming

This project utilizes interrupts during execution. When monitoring the brake encoder and the other various sensors if certain conditions are met power is cut to one or more assemblies. This must occur within a set amount of time. Utilizing interrupts ensures no CPU power is wasted by polling for the conditions. It also ensures quick response time when the power off actions must occur.

Process Timeline

See the Gantt chart that accompanies this document for a complete project timeline.

References

[1] J. Fuller, "How Drive-by-wire Technology Works," HowStuffWorks, 28 April 2009. [Online]. Available: 2015. [Accessed 12 February 2015].

[2] "About Us | Mizzou FSAE," [Online]. Available: http://mizzoufsae.com/?q=node/2. [Accessed 02 February 2015].

[3] SAE International, "SAE Collegiate Design Series - SAE International," 17 September 2014. [Online]. Available: http://students.sae.org/cds/formulaseries/rules/2015-16_fsae_rules.pdf. [Accessed 12 February 2015].