Bruin Formula Racing Fall 2019 Project Specification

Ken Suzuki, Rahul Salvi, Patrick Chau

Last Updated: 9/22/2019

1 Introduction

The goal of this project is to develop a smart relay and fuse box. This box shall have indicators detailing the status of each fuse (blown or unblown), as well as communicate over the CAN bus to the ECU, following the AEMnet specification. In addition, the box shall also house each relay connected to actuators on the car, the full list of which will be detailed in the following sections. The control for the relays will be provided from the ECU.

Skills which will be gained from participating in this project:

· PCB design

· Microcontroller Basics

· An understanding of the CAN Bus

· Finding vendors and intelligently picking out parts based on use case

· Gaining familiarity with electronics on the vehicle

2 Main Requirements

There are several functionalities that we wish to achieve with this project. These are the overarching goals for the project and ideally we achieve all of them.

1. The relay box must have be able to handle 12V.
2. The relay box must have capacity to contain 12 fuses.
3. The relay box must contain 7 relays which can be replaced.
4. Fuses must be easily removable.
5. The box must be able to connect to the CAN bus.
6. LEDs must be visible on the outside of the box to indicate whether or not each fuse is blown. LEDs will be off when fuse is okay. LEDs will be on when the fuse is blown and continuity is lost.
7. USB connection to the microcontroller (if one is to be used) should be easily accessible for programmability.
8. The box should be able to be completely enclosed as well as opened for examination.
9. The box should have as minimal of a form factor as feasible.
10. All diagnostic information will be relayed over the CAN bus (without overlapping with already designated message IDs present on the bus).
11. The budget for the project is ideally $100.

3 Details

The reason for 12 fuses is that each one protects an electronic component from being damaged by a large current. The fuses used will just be typical automotive blade fuses so a part number for a fuse holder which can be soldered to a PCB will need to be found. The purposes for the fuses and their required current values are as follows.

1. O2 sensor - 2A
2. Brake Light - 5A
3. ECU Power - 2A
4. Fuel Pump Power - 15A

5. Fan Power - 15A
6. Injector Power - 5A
7. Coil Power - 10A
8. EFI Main Power - 20A
9. EFI Main Control - 3A
10. Flash Enable - signal, low current
11. 2 auxiliary fuses

Additionally, 6 actuators need to be controlled, the names of which are as follows. Previously we used relays (P/N 1432772-1) that worked well but different control methods (Power FETs for example) can be used.

1. Fuel Pump - 12VDC, 30A
2. Fan - 12VDC, 30A
3. Injector - 12VDC, 30A
4. Coil - 12VDC, 30A
5. EFI Main - 12VDC, 30A
6. Fuse Box Power - 12VDC, 30A
7. Starter Motor - 12VDC, 70A

Some things are not specified in this document but still need to be determined, such as how the box will pin out, what type of connectors are to be used, and specifically how the box will communicate over the CAN bus(both in hardware and in software).

4 Documentation and Softwares

Some softwares that can be used while doing this project. Information on usage can easily be found with a quick Google search, but download links are added for convenience:

1. EagleCAD: Widely used EDA made by Autodesk, free for Students for 3 years unrestricted.Link
2. KiCAD: Hobbyist level EDA, open source and completely free for everyone with full functionality. Link
3. Altium CircuitMaker: recently went free, professional level EDA. Link
4. LTSpice: circuit simulation software, widely used in industry and a great software to know. Large library of components Link
5. Arduino IDE: IDE made specifically for working with Arduino, has a good number of base libraries installed. Of course, you can always use your IDE of choice when coding. Link

Some documents which might be useful:

1. Infinity ECU Manual: Inside this manual has example wiring schematics which shows how the relays should be connected together and how fuses should be put together. Link
2. AEMnet CAN Protocol: this is the specification for the CAN messages the ECU will send out. Link
3. Cooper-Bussmann mVEC: this is a product that's already produced and used that's very similar to what we're designing. Could be used to get ideas on implementation. Link

5 References

Listed below are useful reference materials for self-study. Feel free to read through them when you have time and if you have any questions about something you read, feel free to reach out to us.

1. CAN Protocol Overview: For those unfamiliar with CAN, this is a good overview Link
2. CAN Protocol Specification: For those interested in knowing the ins and outs, you can read the specification straight from Bosch, the company which developed the CAN protocol. Link