Beacon-Controlled Autonomous Network for Trains (BCAN'T)

ECE 445 Design Document

Jordi Pakey-Rodriguez
Prithvi Garimalla
Susan Chen
Group 5
TA - Jacob Bryan

1 Introduction

1.1 Objective

While model trains have become more and more advanced over the last few decades, they are still entirely controlled by a human operator. These human operators are solely responsible for ensuring that no harm comes to the train. A lapse in oversight could be disastrous, causing damages that would be very costly to repair or replace. Model trains currently do not have the ability to detect obstacles or other trains directly in front of them. If not stopped in time, a crash resulting in derailment or physical damage is likely to occur. Additionally, derailment may occur when the trains are traveling at high speeds on a curved tracks. The trains have no recourse for these situations other than for the operator to manually stop or slow the train to avoid an accident.

Our solution is to give each locomotive the ability to respond to the track on its own by detecting obstacles and reacting to programmable beacons placed on the track. A time of flight sensor would be placed at the front of the train to detect obstacles directly in front of it. Small beacon modules placed on the track would be individually programmable to send out signals to the train to let it know what speed it should travel at. Essentially, the end goal is to have the train sense potential issues along the track that would cause collision or derailment and react to avert such problems.

1.2 Background

Currently, the National Model Railroad Association (NMRA) defines standards for Digital Command Control (DCC) for model trains. DCC allows for digital control of individual trains with respect to their lights, speed, direction, sound, and much more[8]. The downside to DCC is that smooth operation is completely dependent on the human operator and controls must be manually set instead of having the train react to its environment autonomously. In addition, there is quite a learning curve when installing DCC and trying to set up a system without prior experience can be quite challenging.

Our solution would be a convenient alternative to those who wish to have speed control for their locomotives. The sensors integrated with the locomotive would also provide some measure of autonomous protection against human negligence by avoiding obstacles and preventing derailment.

1.3 High-Level Requirements

- Must be able to operate indefinitely on existing methods of track power, both analog (DC) and bipolar Digital Command Control (DCC).

- Must be able to detect oncoming obstacles on the track and stop to avoid derailment or damage.
- Must be able to read customizable speed limit beacons that are powered by the tracks and adjust its speed accordingly.

2 Hardware Design

Our solution consists of two independent systems: the beacon board and the locomotive controller. The beacons are individual tiny PCB's on the track that constantly send out a signal and are read as the train passes over them. The locomotive controller consists of three PCB's on the train and is responsible for reading the beacons, identifying obstacles, and controlling the motor. The block diagram is shown in Figure 1.

2.1 Block Diagram

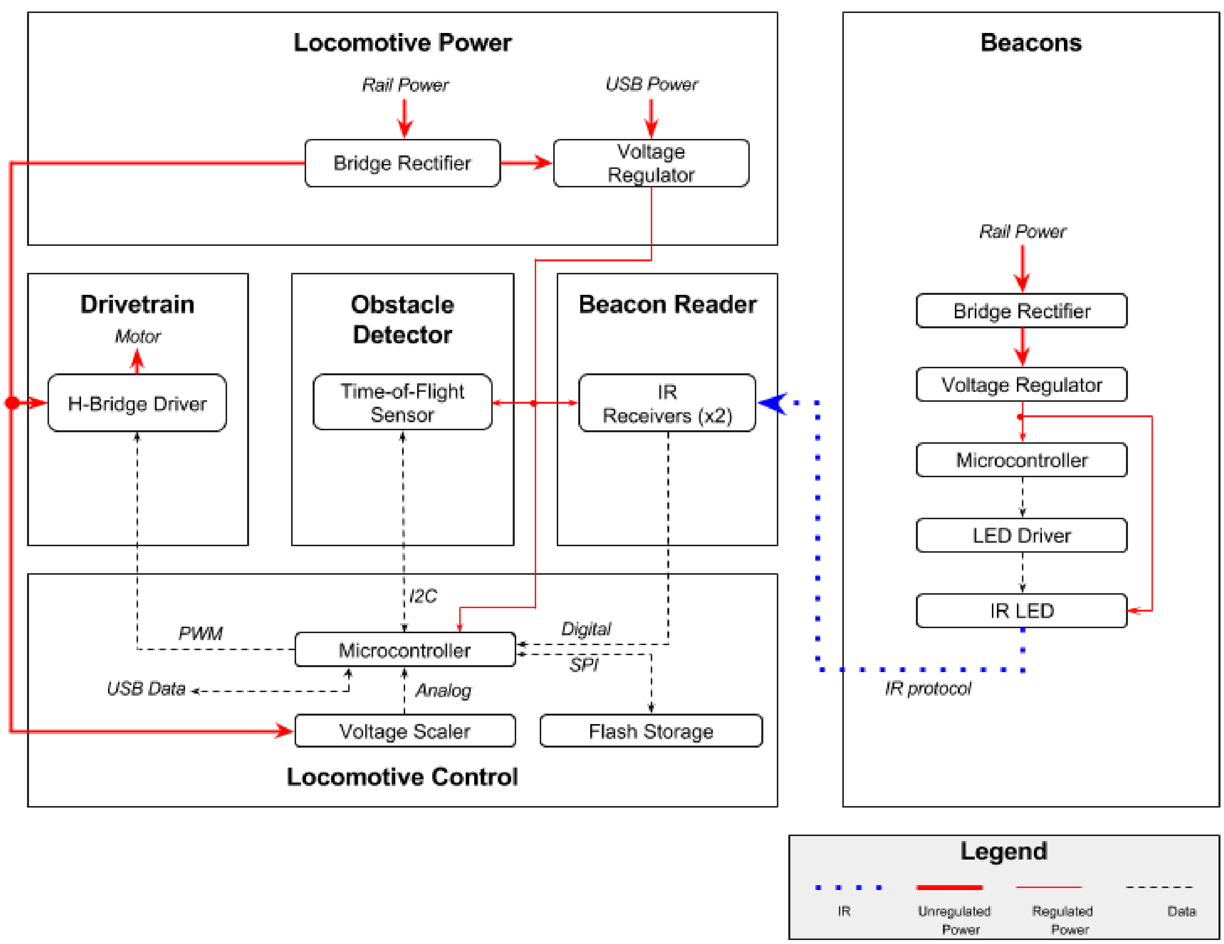

Figure 1: Block Diagram

2.2 Beacon

The beacon is small board that sits between two track ties as shown in green in Figure 2. A bridge rectifier and a voltage regulator are used to provide 2.8V power to the microcontroller, LED driver, and IR LED. The microcontroller will control the IR LED to continuously broadcast the board's ID for the IR receiver on the locomotive board to read.

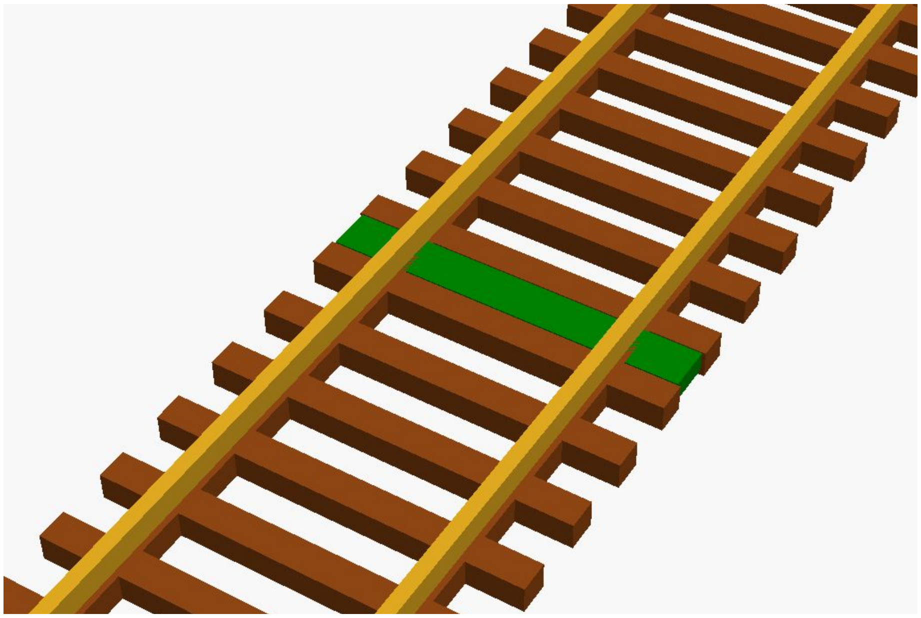

Figure 2: Beacon Rendering

2.2.1 Schematic

The schematic for the beacon board is shown in Figure 3. The shunt resistor R5 will be populated if we find that the LED is unable to switch fast enough for clean 38kHz modulation. The calculations for R2 and R4 are shown in Calculation 1 and Calculation 2, respectively.

Calculation 1:

R_2, gate resistor:

$$Ohm'sLaw : R = \frac{V}{I}$$
$$I/Oinputvoltage, V_o = 2.8V$$
$$Max\ current\ per\ I/O\ pin = 40mA$$
$$Choose\ I_1 = 28mA$$
$$R_1 = \frac{2.8}{0.028} = 100\Omega$$

Calculation 2:

R_4, IR LED series resistor:

$$Forward\ voltage\ of\ IR\ LED,\ V_f = 1.2V$$
$$Supply\ voltage,\ V_d = 2.8V$$
$$Continuous\ DC\ current,\ I_3 = 34mA$$
$$Using\ Ohm's\ Law :$$
$$I_3 = \frac{V_d - V_f}{R_3}$$
$$= \frac{2.8 - 1.2}{R_3} = 34mA$$
$$R_3 = 47\Omega$$

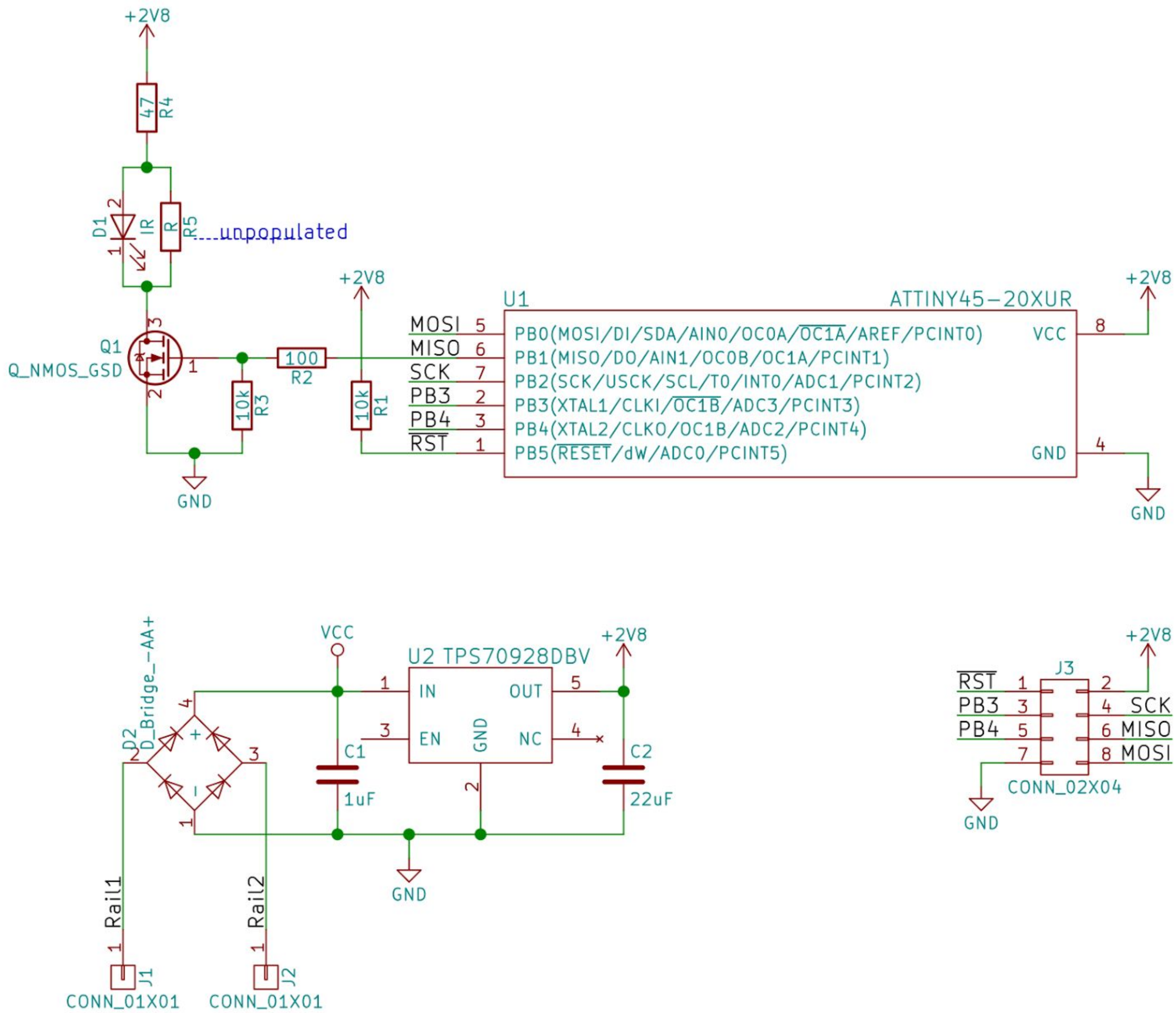

Figure 3: Beacon Board Schematic

2.2.2 Bridge Rectifier

Since the voltage regulator only takes positive voltage, the full-bridge rectifier is needed rectify the power signals from the rails. It will consist of four schottky diodes in a bridge configuration and rectify voltages ranging from -16V to +16V.

2.2.3 Voltage Regulator

A linear voltage regulator will take an input from the rectified rail power and bring the voltage down to 2.8V from an input voltage of up to 16V. It will provide the power needed for the MCU and its peripherals.

2.2.4 Microcontroller

The microcontroller used will be an ATtiny45 8-bit AVR processor. It will use a GPIO pin to control the LED driver circuit to continuously broadcast the board's unique ID using the IR protocol.

2.2.5 LED Driver

The LED driver circuit consists of an N-channel MOSFET, a gate resistor, and a pull down resistor. The MOSFET will conduct current when the I/O pin from the microcontroller is set to high. The gate resistor is used to prevent damage due to overcurrent to the microcontroller and the pull down resistor is used to ensure the MOSFET is turned off when the voltage at the gate is left floating.

2.2.6 IR LED

The 940nm IR LED will be used to broadcast a unique ID that the receiver on the locomotive can read. It will be controlled by the LED driver circuit using PWM. The viewing half-angle for the LED can be calculated using Equation 1.

$$\theta_{\frac{1}{2}} = \arctan\left(\frac{V_{max}T}{2H_{min}}\right)$$

Equation 1

where V_{max} is the max speed of the train, T is the length of one ID message, and H_{min} is the minimum distance from the beacon board LED to the IR receivers on the train.

2.2.7 Requirements and Verifications

Requirements	Verifications
1. Must be able to be powered from rail power of up to ±16V and draw up to 75mA. 2. Must be able to modulate the LED between 38kHz - 40kHz with a peak wavelength of 940nm - 950nm. 3. Must have a view angle θ on the axis parallel to the track given by Equation 1 (see IR LED section for description)	1. Connect the board to a bench power supply set at +16V. Verify the LED is lit using a camera. Repeat with -16V. 2. Connect TSOP57238 IR receiver output to an oscilloscope and point it at the IR LED[6]. Verify that the signal is received. 3. With the datasheet value for θ and the distance H, solve for VT. Place the receiver at a horizontal distance VT / 2 and a height H from the LED and verify with the oscilloscope that the ID is read.

Table 1: R&V for Beacon Board

2.3 Locomotive Power

2.3.1 Schematic

The power circuit shown in Figure 4 consists of BD3575HFP-TR voltage regulator as well as a CDBHM240L-HF full bridge rectifier.

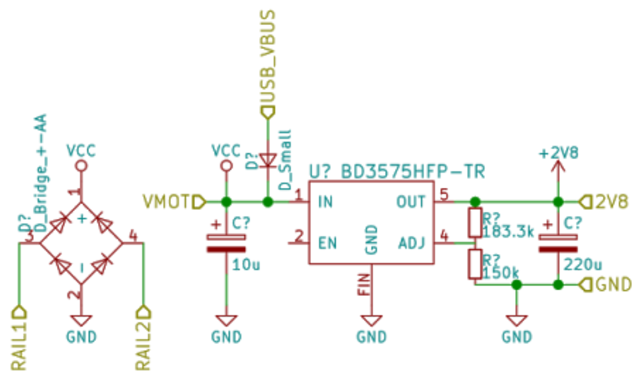

Figure 4: Power Schematic

2.3.2 Bridge Rectifier

The full-bridge rectifier is needed to rectify power signals from the rails before they are routed to the voltage regulator, voltage scaler, and motor driver. It will consist of four schottky diodes in a bridge configuration and rectify voltages ranging from -16V to +16V.

2.3.3 Voltage Regulator

The rectified voltage will be fed into a linear voltage regulator in order to bring the voltage down to 2.8V. The 2.8V source will be used to power the microcontroller, time-of-flight sensor, IR receiver, and the voltage scale circuit. Calculation 3 shows how the output voltage is calculated for the BD3572FP linear regulator.

Calculation 3:

$$V_{out} = 1.26\frac{R_1 + R_2}{R_1}$$

$$\frac{2.8}{1.26} = \frac{R_1 + R_2}{R_1}$$

$$Using\ manufacturer's\ recommended\ R_1\ value,\ R_1 = 150k\Omega :$$

$$R_2 = 183.3k\Omega$$

2.3.4 Tolerance Analysis

The voltage regulator is a critical component of our project and must regulate the rail voltage properly. If the voltage regulator does not operate within our requirements, there is a risk of inaccurate readings from the time-of-flight sensor which has an optimal operating voltage of 2.7-2.9V. Without accurate readings, the train may hit obstacles on the track. Also, the voltage scaler circuit includes a single supply operational amplifier that is not designed to operate at voltages much lower than the regulated output. Without an accurate reading of the rail voltage, the correct speed cannot be calculated with the microcontroller. If the voltage regulator were to fail, components could be damaged if the voltage was not regulated low enough or not turn on at all if the voltage was regulated too low.

The resistor divider at the output pin and the adjustment pin of the linear regulator determines the output voltage. Therefore, we will use resistors with 1% tolerance to minimize the range of potential output voltages. Table 2 shows the nominal values for each resistor and ±1% for each resistor value. Table 3 shows the output voltage calculated using Calculation 3 and the percent difference from the required 2.8V.

Resistor Tolerance	Resistor Value (kΩ)
Nominal R1	150
Nominal R2	183.3
R1+1%	151.5
R2+1%	185.133
R1-1%	148.5
R2-1%	181.467

Table 2: Resistor Tolerance Values

R1(kΩ)	R2(kΩ)	Vout(V)	% Difference
150	183.3	2.79972	0.01
150	185.133	2.815117	0.54
150	181.467	2.784323	0.56
151.5	183.3	2.784475	0.55
151.5	185.133	2.79972	0.01

151.5	181.467	2.76923	1.10
148.5	183.3	2.815273	0.55
148.5	185.133	2.830825	1.10
148.5	181.467	2.79972	0.01

Table 3: Vout calculated with 1% tolerance resistors

From Table 3, the worst case scenario would produce a voltage with a percent difference of 1.10%. This would produce output voltages of either 2.769V or 2.821V, which would fall well within our requirement of regulating to 2.8±0.2V. It can be concluded that if we used 1% tolerance resistors for the voltage regulator, our design would still function correctly.

2.3.5 Requirements and Verifications

Requirements	Verifications
1. Must be able to accept voltages up to ±16V from the rail supply and output 2.8±0.2V and supply up to 280mA	1. Connect the locomotive to maximum rail power, verify the 2.8V output is active with a multimeter. Reverse the orientation of the train on the track and verify the 2.8V output again. Then put a 10Ω 1 Watt resistor across the 2.8V output and measure the current.
2. Must be able to output rectified rail power if the rail supply is connected	2. Connect the locomotive to rail power. Using a multimeter measure if the input to the motor driver is a rectified version of the rail power.
3. Must be able to accept voltages between 4.4V-5.2V from the USB supply and output 2.8±0.2V power and supply up to 280mA	3. Remove the train from the track and connect USB power. Verify the 2.8V output is active with a multimeter. Then put a 10Ω1 Watt resistor across the 2.8V output and measure the current.

Table 4: R&V for Locomotive Power

2.4 Locomotive Control

The locomotive control unit consists of a microcontroller and a USB data connection. If there is a USB connection the control unit will enter a programming mode, otherwise it will run the normal track operation. This will allow operators to map the beacon IDs to one of 128 different speeds[3]. The microcontroller receives a voltage input via the ADC pin from the voltage scale circuit and outputs a PWM signal to the drivetrain to control the speed. The IR receiver sends beacon IDs to the microcontroller in order to determine what speed the train should go. It will also stop the train if the time-of-flight sensor detects an obstacle in its path.

2.4.1 Schematic

The main part of the control circuit is the STM32F4 microcontroller. It is connected to a M95512-RDW6TP EEPROM for storage of the beacon mappings. The control schematic is shown in Figure 5.

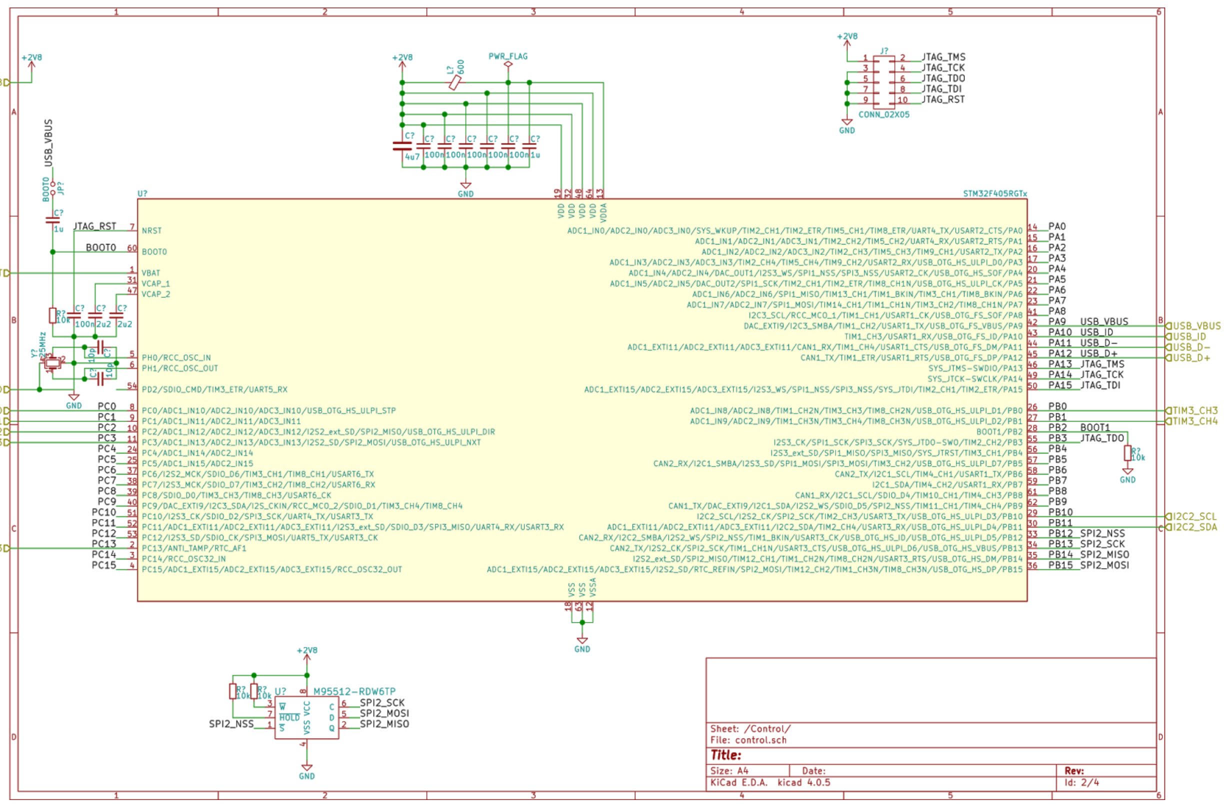

Figure 5: Control Schematic

2.4.2 Microcontroller

The microcontroller will be a STM32F405 32-bit ARM processor. It will communicate with the time-of-flight sensor via I2C, the IR receiver digitally, and control the motor with PWM signals. It will also store configuration data for the beacons in external memory.

2.4.3 Flash Storage

An EEPROM is used to store the beacon mappings. It communicates with the microcontroller over SPI. EEPROM was chosen since we can address individual bytes, which will make performing binary search through the dataset for the beacon mappings more efficient.

2.4.3 Voltage Scaler

The voltage scaler will be a voltage divider circuit to scale down the rectified rail voltage to a lower voltage that can be read by the microcontroller's ADC. The output of the voltage scaler circuit will have a voltage follower in order to not load down the voltage divider circuit. The rectified voltage ranges from 0-16V and the microcontroller ADC can read voltages from 0-1.8V. Two resistors will divide the voltage to output a range of 0-250mV, which will give around 568 different readings for a 12-bit precision ADC. The calculations for the resistor values for the voltage divider are shown in Calculation 4.

Calculation 4

$$Voltage\ divider\ rule:\ V_{out} = V_{in}\frac{R_2}{R_1 + R_2}$$

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2}$$

$$\frac{0.250}{1.8} = \frac{0.250}{15.75 + 0.25}$$

$$R_1 = 15.75k\Omega \quad R_2 = 0.25k\Omega$$

2.4.4 Requirements and Verifications

Requirements	Verifications
1. Must be able to act as a USB Mass storage device when connected to a PC	1. With the final microcontroller firmware, connect it to a PC to put it into programming mode. Copy a valid mapping file to the Mass Storage Device the microcontroller presents itself as. Then without unplugging the microcontroller, verify the file can be read.
2. Must be able to store beacon ID mappings between power cycles	2. Repeat Verification 1, with the beacon mappings set between two IDs for which there are physical beacons that correspond to turning the lights on and the turning the lights off. Then unplug and reconnect the USB. Power the beacons, then hold the locomotive over the beacons and verify the correct actions occur.
3. Must be able to scale a voltage range of 0-16V down to 0-2.8V and be read by	3. Write and flash a test program that

the microcontroller's ADC with at least 10 bit precision.	displays the current current ADC reading over the serial connection to the computer. Power the board over USB while connecting the rail power input to a bench power supply and sweep between 0V-27V and ensure the microcontroller is accurately displaying at least 128 distinct speed steps.

Table 5: R&V for Locomotive Control

2.5 Drivetrain

The locomotive drivetrain unit consists of the motor controller that will interface the PWM signal from the microcontroller with the rectified rail power and use that to power the motor.

2.5.1 Schematic

The ZXMHC6A07N8TC is a dual P-MOSFET dual N-MOSFET H-bridge IC that we are using to control the motor. The 4N25 are optoisolators used for isolation and as gate drivers. The drivetrain schematic is shown in Figure 6.

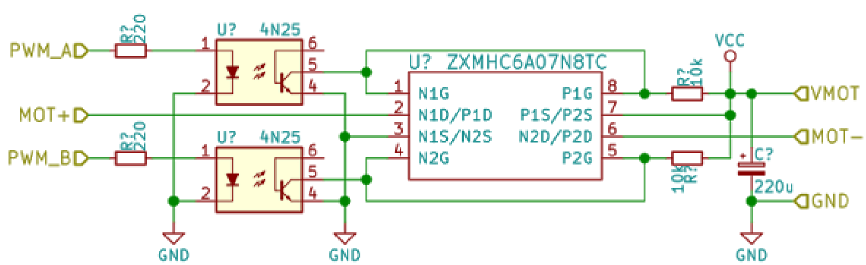

Figure 6: Drivetrain Schematic

2.5.2 Motor Controller

The motor controller will be used to control the speed and direction of the DC train motor.

2.5.3 Requirements and Verifications

Requirements	Verifications
1. Must be able to take a direction signal	1. Write and flash a test program to the

and a PWM signal from the microcontroller and control the motor accordingly.	locomotive microcontroller that will cycle through 0%, 25%, 50%, 75%, and 100% duty cycles on the PWM line with the direction line both high and low for each. Verify the output of the motor controller with an oscilloscope.

Table 7: R&V for Drivetrain

2.6 Obstacle Detector

The obstacle detector will detect any obstruction that could cause the train to derail and cause an emergency stop.

2.6.1 Time of Flight (TOF) Sensor

The TOF sensor will use a laser to measure whether or not the path in front of the train is clear or if the train is approaching an unexpected obstacle.

2.6.2 Requirements and Verifications

Requirements	Verifications
1. Must have a 16.5mm breadth of view at 100mm from the front of the train	1. Write and flash a test program to the microcontroller that will display over the PC serial connection the values read from the sensor. Place an object at a distance of 100mm in front of the sensor and 8.25mm off the center axis of the train and perpendicular to the track. Move to 8.25mm off the center axis in the opposite direction. Verify both times the object is detected.
2. Must have a sample rate of 10Hz	2. With the same sample program from verification 1, ensure the output is updating 10 times a second by recording the number of updates in 10 seconds and dividing by 10.

Table 8: R&V for Obstacle Detector

2.7 Beacon Reader

The beacon reader consists of two IR receivers mounted underneath the train on either side. They will read beacons as the train passes over them. With two receivers, the microcontroller can infer the direction the train is moving and execute different actions accordingly (i.e. going into a bend vs coming out of a bend).

2.7.1 IR Receiver

The IR receiver will read the beacons as the train passes over them.

2.7.2 Requirements and Verifications

Requirements	Verifications
1. Must be able to sense a 38kHz modulated 940nm IR beam and output a demodulated digital signal	1. Connect a beacon to rail power. Write and flash a test program that will display over the PC serial connection the value read from the IR receiver. Verify that the value read matches the beacon's ID.

Table 9: R&V for Beacon Reader

3 Software Design

3.1 IR Protocol

The IR protocol will use 38kHz modulation present in many televisions and other consumer devices. We chose this frequency because sensors for it are widely available. Our protocol uses Manchester encoding according to the G.E. Thomas convention with a bit period of 960 μs and data length of 16 bits[7]. The timing diagram for the protocol is shown in Figure 7.

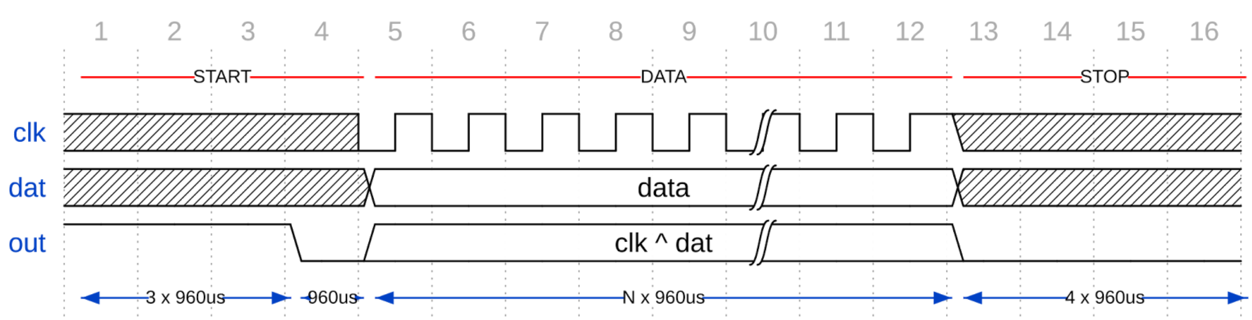

Figure 7: IR Protocol Timing Diagram

3.2 Beacon Firmware

The beacon firmware is a very simple loop that continuously broadcasts the beacon ID using the IR protocol.

3.2.1 Pseudocode

```
while (1) {
    // Start of message
    on(PERIOD * 3);
    off(PERIOD);

    // Transmit unique ID using manchester code, MSB first
    // 0 - rising edge
    // 1 - falling edge
    for (i = LENGTH - 1; i >= 0; i--) {
        bit = (ID >> i) & 1;
        for (clk = 0; clk <= 1; clk++) {
            if (clk ^ bit) {
                on(PERIOD / 2);
            }
            else {
                off(PERIOD / 2);
            }
        }
    }

    // End of message
    off(PERIOD * 4);
}
```

3.3 Locomotive Firmware

The Locomotive has two modes of operation: track mode and programming mode. It also reacts to Pin Change Interrupts from the IR receiver. At boot, the MCU checks to see if USB is connected. If so, it enters programming mode and presents itself as a USB Mass Storage Device. When the user starts to write a configuration file to the locomotive, the MCU will buffer each beacon mapping and verify they are being written in ascending order since at lookup-time, it will perform a binary search through the mappings to find the correct one.(if an errored mapping comes in, the MCU will stop storing more mappings).

When in track mode, the MCU will continuously check the Obstacle Detector, stopping the train if there is one, then whether a beacon ID is ready, in which case it looks up the appropriate mapping and executes it, and finally make any motor adjustments as necessary.

The IR receiver also works by invoking a Pin Change Interrupt every time its value changes. Using our IR Protocol and the repeated nature of the data we are sending, we need only store one message length worth of pin changes to calculate the beacon ID. When we have the requisite data, we set a “Ready Flag” which will trigger the main event loop to lookup and execute the mapping.

3.3.1 Flow Charts

The flowchart for the locomotive operation is split into two sections, the main locomotive operation is shown in Figure 8 and the interrupt control flow is shown in Figure 9.

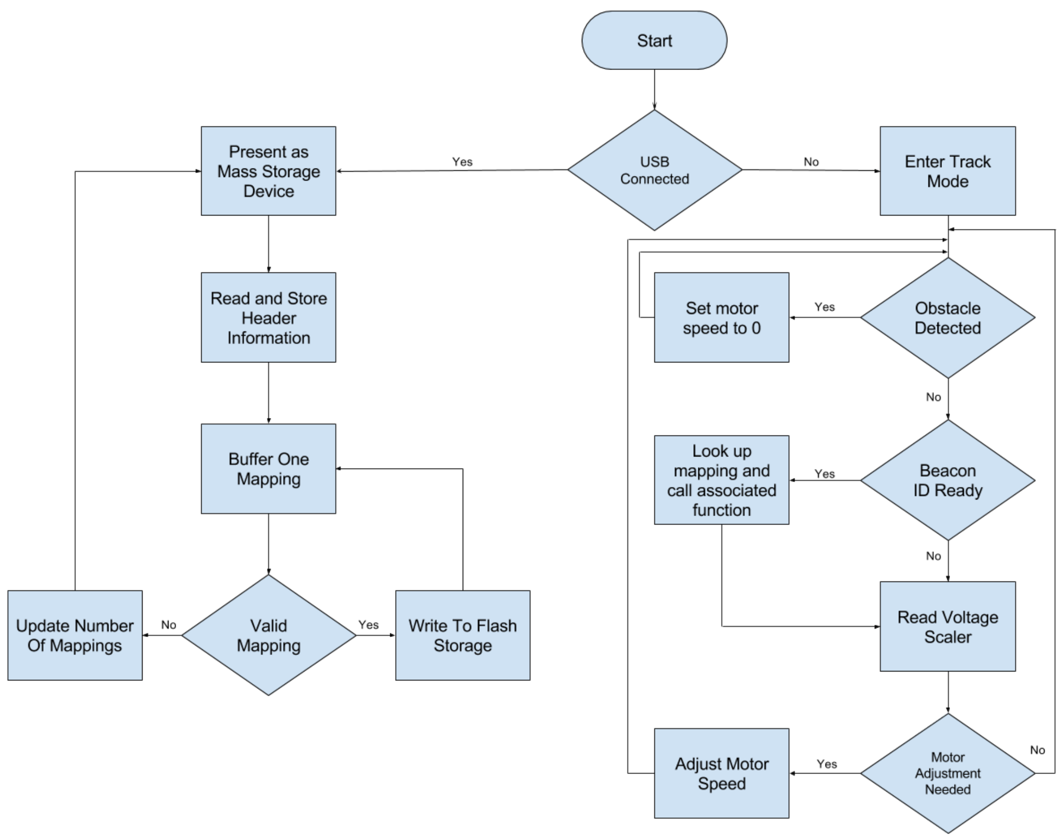

Figure 8: Main Locomotive Control Flow

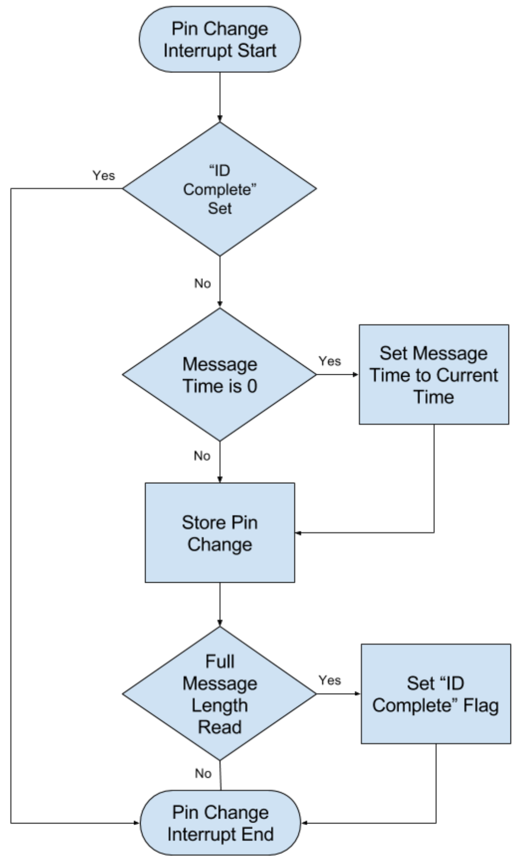

Figure 9: Interrupt Control Flow

3.3.2 Feature List

- Mount as USB MSD device if USB plugged on boot
- Stores the beacon mappings on flash memory
- Reads beacons through Pin Change Interrupts
- Can stop the train if an obstacle is detected
- Can continuously adjust the motor speed and direction based on track conditions.

3.4 PC Software

3.4.1 Prototype

Figure 10 is a mockup of the web interface (hosted online here) to program the beacons. Once specifying the ID, the user can select the instances where the mapping applies and then the actual action the beacon corresponds to.

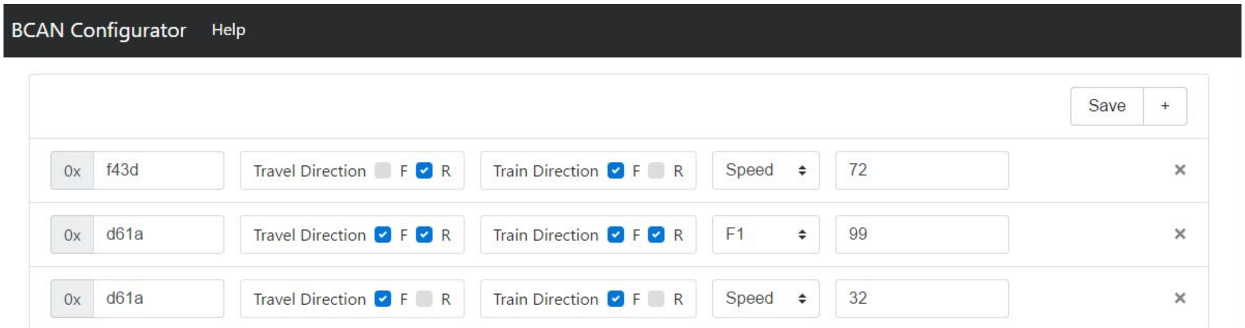

Figure 10: Prototype of Beacon Configurator

3.4.2 Feature List

- Allows user to add any number of beacon mappings
- Each mapping consists of a beacon ID, description, direction of travel, direction of drain, function, and additional parameters for that function
- Clicking "Save" downloads a file that can be copied to the USB Mass Storage device to program the mappings

4 Cost and Schedule

4.1 Cost

The grand total for producing our beacon board is $45,757.86. The breakdown between labor and component costs can be seen below in Tables 10 and 11.

4.1.1 Labor

Name	Hourly Rate	Total Hours	Total	Total*2.5
Prithvi Garimalla	$36	170	$6120	$15,300
Susan Chen	$36	170	$6120	$15,300
Jordi Pakey-Rodriguez	$25	210	$5250	$13,125
Team Cost	$97	550	$17,490	$43,725

Table 10: Labor Cost Estimates

4.1.2 Parts

Description	Manufacturer	Part #	Quantity	Cost
ATtiny45	ATMEL	ATTINY45-20XUR	1	$1.19
IR LED	EVERLIGHT	IR19-21C/TR8	1	$0.50
Rectifier	Infineon	BAS4002A-RPP	1	$0.41
2.8V Voltage Regulator	Texas Instruments	TPS70928DVB	1	$1.37
ARM Processor	STMicroelectronics	STM32F405RGT6	1	$10.58
EEPROM	STMicroelectronics	M95512-RDW6TP	1	$0.95
IR Receiver	Vishay	TSOP57238TT1	2	$4.46
Rectifier	Comchip	CDBHM240L-HF	1	$0.84
2.8V Voltage Regulator	Rohm	BD3575HFP-TR	1	$3.17
TOF Sensor	STMicroelectronics	VL6180X	1	$6.00
Op-Amp	STMicroelectronics	LM358DR	1	$0.47
Optoisolator	Lite-On	4N25	2	$0.96
H-Bridge	Diodes Incorporated	ZXMHC6A07N8TC	1	$2.04

Table 11: Component Cost Estimates

4.2 Schedule

Week	Prithvi	Jordi	Susan
2/26	Start writing Locomotive firmware	Start PCB layout	Start Locomotive Control Prototype on breadboard
3/5	Finish and start debug of firmware	Finish PCB layout and submit for manufacturing	Test firmware on breadboard prototype
3/12	Continue debugging prototype/start soldering PCB's if arrived		
3/19	Spring Break/Individual Progress Reports		

3/26	Start debugging on actual train	Finish soldering/ start debugging on actual train
4/2	Finish debugging (4/7 goal for full functionality)	
4/9	Start preparing for mock demo/presentation	
4/16	Start final presentation/practice presentation	Start final paper/practice presentation
4/23	Final demo and presentation practice/start final paper	
4/30	Finish final paper	

Table 12: Component Cost Estimates

5 Ethics and Safety

There are two potential safety concerns with our project, however we believe they are both within the acceptable tolerances for a general consumer product. The first of these is that the rails of the train track are electrically charged so one could potentially shock themselves. However, we are not changing this characteristic in our design and therefore are not creating any additional safety hazards. This is a well understood risk of model trains so responsible usage dictates they should be kept out of reach of small children or pets. The second is that we are using a time-of-flight sensor that uses a laser. The laser used is Class 1, which means that it is safe for use under normal conditions and while operating within the manufacturer's specifications[5]. We fully intend on staying within those guidelines.

In accordance with the IEEE code of ethics, we will only endeavor to take on technical tasks we feel we are qualified for and seek advice from our mentors for the skills we lack. We will seek feedback and criticism for our work and continuously work to correct our mistakes. We will strive for honesty and properly credit individual contributions to our project. Most importantly, we will support and encourage each other to follow this code of ethics.

6 References

[1] "Electrical Standards for Digital Command Control", *http://www.nmra.org*, 2006. [Online]. Available: http://www.nmra.org/sites/default/files/standards/sandrp/pdf/s-9.1_electrical_standards_2006.pdf. [Accessed: 08- Feb- 2017].

[2] "NMRA STANDARDS S-9 Electrical", *http://www.nmra.org*, 1984. [Online]. Available: http://www.nmra.org/sites/default/files/standards/sandrp/pdf/s-9_1984.08.pdf. [Accessed: 08- Feb- 2017].

[3] "NMRA STANDARD", *www.nmra.org*, 2012. [Online]. Available: http://www.nmra.org/sites/default/files/s-9.2.1_2012_07.pdf. [Accessed: 08- Feb- 2017].

[4] "USB in a NutShell - Chapter 2 - Hardware", *Beyondlogic.org*, 2010. [Online]. Available: http://www.beyondlogic.org/usbnutshell/usb2.shtml. [Accessed: 08- Feb- 2017].

[5] "VL6180X", *www.st.com*, 2016. [Online]. Available: http://www.st.com/content/ccc/resource/technical/document/datasheet/c4/11/28/86/e6/26/44/b3/DM00112632.pdf/files/DM00112632.pdf/jcr:content/translations/en.DM00112632.pdf. [Accessed: 08- Feb- 2017].

[6]"IR Receiver Modules for Remote Control Systems," in Vishay, 2017. [Online]. Available: http://www.vishay.com/docs/82434/tsop572.pdf. [Accessed: Feb. 20, 2017].

[7]"Manchester Coding Basics," in *Atmel*, 2015. [Online]. Available: http://www.atmel.com/images/atmel-9164-manchester-coding-basics_application-note.pdf. [Accessed: Feb. 20, 2017]

[8]"DCC working group," in *National Model Railroad Association*, National Model Railroad Association. [Online]. Available: http://www.nmra.org/dcc-working-group. [Accessed: Feb. 21, 2017].