ELEC 291 Project 2 Report

Coin Picking Robot

Section L2B: Tuesday/Thursday 8 am - 11 am

Submission Date: April 8th, 2025

Group #: B03

Name	Student #	Grade	Signature
Taran Gill	14113278	105	
Jadon Lao	89583587	110	
Aiman Ishmam	22510614	110	Aiman
Yu Chen (Jerry) Liu	67589390	50	Jerry
Jeremy Fernandez	65199846	120	
Junyoung Park	56848930	105	

Table of Contents

1.0 Introduction

In this project, we will be designing a coin picking robot as well as a remote to control the robot. The robot's main functionalities will include an automatic mode that will automatically detect and collect coins within a set perimeter, and a manual mode that collects coins while being controlled by the remote control. The team's design approach was to first assemble and work on the individual components, then combine the various components together and ensure base functionality, and then work on bonus features. For circuit and system block diagrams please refer to *Appendix A to H*.

The following is a summary of the base functionalities as well as the bonus features of the project:

1. Basic functionality
    a. Metal detector
    b. Perimeter detector
    c. Coin picker
    d. Automatic / Manual modes
2. Bonus features
    a. Ultrasonic sensor that allows the robot to avoid obstacles on its way.
    b. Rear LEDs that indicate which way the robot is rotating or moving backward.
    c. 7 segments display that shows how many coins have been picked up.

2.0 Investigation

2.1 Idea Generation

Since this project has a massive amount of approaches to reach the same end goal, our first idea generating task was to figure out which steps and approaches we would take for the project. To ensure that our project followed a general outline we worked on our base functionalities as well as generated ideas for our bonus features.

We decided that our approach should involve the PIC32 family MX130F064B for the controller and the 8051 family EFM8LB1. The reasoning for choosing this is because we were more comfortable with these microcontrollers. The rest of the process consisted of working on robot mechanics first, then developing both the hardware and software of both the controller and the robot.

As a result of this process, we decided that our group would generate bonus feature ideas contributing to an easier operation for the user. Examples of this include:

- Adding headlights/backlights
- Potential LEDs for Joystick movement
- More accurate way to count coins

2.2 Investigation Design

Throughout the project, the team conducted experiments, collected data, and validated results to ensure that everything was properly functioning. We would need to use experimentation and modes of verification to ensure that both the robot and controller are working in sync. Some examples of this being used in design investigation involve:

- Radio Testing → Joystick and Servo Testing
- Electromagnet Testing → Perimeter Testing & Coin Picker
- Robot and Controller Code Development
- Accuracy Development – Ensuring Coins are picked up

For the Radio Testing, we sent numbers counting upwards between the master and the slave to ensure that communication between the radios worked. Afterwards, we were able to test our Joystick and Servo by sending information depending on the positioning of the joystick. As for the Electromagnet Testing, we were testing the hardware using oscilloscopes and multimeters to gauge the voltage we should set our thresholds for the perimeter and magnet should be. Both of these contributed to the development of the code, where extensive testing is done through flashing the code.

2.3 Data Collection

<u>PuTTy:</u> The PIC32 collects analog-to-digital (ADC) readings from the horizontal and vertical movements of the joystick, converts the ADC readings to voltage and wirelessly transmits the voltage data to the EFM8 via the JDY-40 wireless transmitter. After testing voltage levels at different joystick positions, code verification was checked on PuTTy.

<u>Oscilloscope:</u> The team tested a variety of coins and metallic objects and observed the period change of the Colpitts oscillator circuit on the oscilloscope.

Multimeter: In order to check when either microcontroller pins are outputting the right voltage or whether the voltage source was working, we connected a multimeter to specific spots in the circuit to spot errors.

Voltage Generator: We used the voltage generator to check whether the circuit was the problem or whether the batteries used were running low.

2.4 Data Synthesis

The ADC readings from the joystick on the PIC32 board are converted to voltage, mapped to the respective movement command (forward, backward, left, or right). Both the direction and the ADC value are displayed on the liquid crystal display (LCD). Additionally, the ADC and voltage values, along with the frequency of the Colpitts Oscillator circuit, are displayed on PuTTY during debugging. This data was analyzed to ensure that the correct frequency values were used for coin detection and that the appropriate voltage was detected at the perimeter of the coin collection area. Any discrepancies in ADC readings or frequency variations were used to fine-tune the system parameters, ensuring accurate movement control and reliable detection.

2.5 Analysis of Results

Because many factors depended on specific parts of the robot to remain constant, it was complicated to attain validity for components. The voltage in the battery drains, leading to many components of the robot having inconsistent values. We dealt with this and verified it by extensive testing and going through trial and error till the results were acceptable. One scenario is an example:

Various types of coins were tested on the inductors at the bottom of the robot to see how the different coin types and sizes impact the resting frequency of the Colpitts oscillator circuit. The frequency range is roughly 59k-61k depending on initialization, while metal spikes may be a 0.4k - 1.5k increase. Our code was calibrated accordingly.

3.0 Design

3.1 Use of Process

At the start of our project, we developed an approach that would ensure that we had the base features done at first. We broke down the overall approach into smaller objectives: Robot Mechanics, Circuitry of Robot base and Controller, JDY40 usage and EFM8/PIC32 programming. This subdivided approach allowed for us to decompose the main objective into individual components for the team to work on.

During the circuit assembly, we referred to many circuit designs such as an optocoupler circuit and Colpitts Oscillator circuit to assemble parts of the circuit. By utilizing circuit designs that were given to us and integrating that into our design, we were able to develop our own design how we wanted it. Through testing with multimeter and oscilloscopes, we appropriately identified and solved issues between the software and hardware.

3.2 Need and Constraint Identification

Our following needs and constraints are identified as follows:

- Ensure the robot can detect the perimeter wire, which carries a 20V AC signal at 16.0 kHz, and autonomously return inside the boundary once detected, without requiring any remote control assistance.
- Enable the robot to autonomously collect 20 coins placed within the perimeter, with the system shutting down completely after a successful collection. The robot should allow no more than three false positives during operation.
- Play a sound on a speaker with an intensity that increases proportionally to the strength of the metal detection signal, providing intuitive audio feedback.
- Design the robot to function independently without requiring a constant connection to a laptop, instead relying on a battery-powered system integrated into the remote-controlled car.
- Due to limited space on the robot's main breadboard, an additional breadboard was attached to accommodate the wiring required to power the EFM8 microcontroller board using the 5V battery system.

3.3 Problem Specification

Battery Longevity in the Remote-Controlled Car:

- Observed rapid depletion of batteries during testing and operation.
- Power sources affected:
    1. A 9V battery powers all components on the slave board.
    2. Four 1.5V batteries power the EFM8 microcontroller board.
- Testing observations:

   1. After 10 minutes of operation, the four 1.5V batteries dropped from 6.5V to 5.8V.
   2. Voltage drop led to reduced performance, especially in frequency detection accuracy, perimeter detection accuracy, and overall system responsiveness.
- Implemented solution:
   1. Replaced old 9V battery and four 1.5V batteries with fresh ones before final demonstration.

Inconsistent ADC Readings in the Metal Detection Circuit:

During the testing phase, we encountered a few issues, which are listed as follows, including a solution for each issue:

- The base frequency fluctuated significantly during testing.
- This instability occurred when using a 1kΩ resistor in the circuit. This was fixed by replacing the 1kΩ resistor with a 100kΩ resistor, stabilizing the base frequency.
- Detection Inconsistency:
   1. At times, the coin placed under the inductor was not detected.
   2. This was due to excessive distance between the coin and the inductor.
- Mechanical Adjustment:
   1. Zip ties holding the inductor were cut to lower its position.
   2. Tape was added to further reduce the distance between the inductor and the coin, improving frequency change detection and more consistent ADC readings.

3.4 Solution Generation

In addition to the core functionality of the coin-picking robot, we implemented a few enhancements to improve usability, feedback, and automation:

1. **Tail Light Direction Indicators**
   - LEDs are positioned to mimic the conventional tail lights found in standard vehicles.
   - The left or right LED lights up to indicate the direction of movement (left or right turn).
   - When the car reverses, both LEDSs follow the standard automotive convention for indicating reverse motion.
   - This enhanced the realism and user experience by simulating actual car lighting behaviour.
   - Provides visual feedback to easily determine the direction in which the car is moving.

2. **Coin Counter Display**
   - Integrated a 7-segment display to show the number of coins picked up by the car.
   - Mechanism:
     1. Each time the servo motor activates to pick up a coin, the system increments a counter.
     2. The updated count is displayed in real time on the 7-segment display.
   - Counter Behaviour:
     1. The count increments by one with each successful coin pickup.
     2. Once the count reaches 20, it resets to zero and begins counting again, following the coin collection requirements defined in the project scope.

3.5 Solution Evaluation

Criteria	**Option 1: Basic Metal Detection Car**	**Option 2: Enhanced Metal Detection Car with Bonus Features**
User interface & feedback	No display or basic visual indicators	7-segment display shows coin count; green LED taillights indicate movement direction
Metal Detection Accuracy	Inconsistent due to frequency fluctuation and inductor distance	Improved with resistor change and better inductor positioning
Coin Counting Mechanism	Display it on the LCD on the slave board but not clearly visible to the user	Counter increments on each pickup and displays count on 7-segment display, resets at 20

After comparing the design options, we chose Option 2 as our final design because it was the option that met our design criteria more accurately:

- Improved User Interface: The inclusion of a 7-segment display and green LED tail lights provides intuitive, real-time feedback that significantly improves usability over the basic design.

- Reliable Metal Detection: By replacing the 1 kΩ resistor with a 100 kΩ resistor and optimizing the inductor's placement, Option 2 offers greater stability and accuracy in frequency detection.
- Clear Coin Count Feedback: Unlike Option 1, where the coin count was displayed on the LCD in a less visible location, Option 2 uses a front-facing 7-segment display to clearly show the number of coins collected, improving user awareness during operation.

3.6 Detailed Design

In this project, we used two microcontrollers with a master-slave protocol. The PIC32MX130 is the master in the controller and the EFM8LB1 in the coin-picking robot. The communication between the two microcontrollers was done using a JD40 radio signal.

The controller, which is depicted in *Appendix D*, uses the already mentioned PIC32 as well as the USB adapter BO230X for loading source code. Furthermore, as required for functionality, it also uses a 9v battery with an AAAA for voltage conversion to 5V. For movement functionality, it uses a joystick to send a voltage to the ADC in the PIC32. It also includes a speaker that uses PWM to make it beep depending on the strength of the metallic material*.

The robot is composed of 2 GWServo motors S05/STD to move around the perimeter. The two motors are powered by four 1.5-volt batteries, using a total voltage of 6 volts. The forward, backward, left, and right movements of the robot are controlled by four output pins on the EFM8. The output signals are sent to an H-bridge that alternates between logic 0 or 1, making the wheels move. For metal detection, we used a Colpitts oscillator, as shown in *Appendix A*, to detect metallic material. In order to detect the voltage when the robot is near the designated perimeter, we used two tank circuits, as shown in *Appendix B*. The picking coin mechanism is performed by two server motors and an electromagnet. When a coin is detected, the lower servo motor would rotate counterclockwise while the upper servo motor would descend. Once in the correct position, a signal is sent from the EFM8. As depicted in *Appendix D*, the signal would turn on the electromagnet, attracting the coin. After this, the upper servo motor rises, and the lower one turns clockwise. Once aligned with the bucket, the EFM8 deactivates the electromagnet, dropping the coin.

3.7 Solution Assessment

1. Evaluation based on requirements, needs, and constraints:

The coin-picking robot was assessed based on key functionality and design requirements:

- Accurate metal detection: The system must reliably detect metallic objects placed under the robot and respond accordingly, using a Colpitts oscillator-based metal detector.
- Real-time user feedback: The system must provide visual output for coin detection and movement status.
- Perimeter Detection: The robot must detect and stay within an AC perimeter defined by a function generator.
- Autonomous and Manual Operation: The robot must operate both autonomously to pick 20 coins within a perimeter and manually using a joystick.
- Tail Light Functionality: Two green LEDs mounted on the rear of the robot act as tail lights, indicating in which direction the robot is moving.

- Coin Counter Display: A 7-segment display connected to the slave board should display the number of coins the robot has picked up.

2. Testing Method and Results

Metal Detection Accuracy

Test: Multiple Canadian coins (5¢, 10¢, 25¢, $1, and $2) were placed under the inductor. The change in oscillator frequency was monitored on PuTTY and oscilloscope to understand its behaviour.

Result:

- With a 1 kΩ resistor, frequency readings were unstable. Replacing it with a 100 kΩ resistor significantly reduced the fluctuation.
- Lowering the inductor using cut zip ties and added tape brought it closer to the ground, improving detection reliability.
- The robot picked up all the coins without any difficulty and only encountered 3 false positives while picking up the coins during the demonstration.

Perimeter Detection

Test: The perimeter was constructed using wire to enclose an area of 0.5 m² and energized with an AC signal of 20 V peak-to-peak at 16.0 kHz. The frequency was determined using the resonant frequency formula:

$$f = \frac{1}{2\pi\sqrt{L.C}}$$

Result: When the robot approached the perimeter, it consistently detected the boundary and changed direction to remain within the defined area. The direction of movement after encountering the perimeter was randomized using the rand() function in C, ensuring the robot did not exit the perimeter regardless of the side it encountered. This randomized behavior added robustness to the boundary avoidance logic.

Autonomous Coin Collection Task

Test: 20 coins were randomly distributed inside the perimeter area. The robot was set to autonomous mode.

Result:

- The robot successfully collected 18-20 coins every time during the testing phase and demonstration.
- It remained inside the perimeter during all attempts.

Tail Light Functionality

Test: Two green LEDs were mounted on the rear of the robot to act as taillights, mimicking the conventional functionality of real vehicle lighting.

Result:

- When the robot moved forward, both LEDs remained on continuously.
- When reversing, the LEDs blinked at regular intervals to signal reverse motion.

Coin Counter Display:

Test: Each successful pickup incremented a counter stored in the microcontroller and displayed it on a 7-segment display.

Result:

- The display accurately reflected the number of coins collected.
- The counter resets to 0 upon reaching 20, as per the project constraint.

3. Strengths of our project:
    - Reliable coin detection and pickup system after optimizing oscillator and inductor placement.
    - Proper power management using dedicated batteries for motors and logic systems.
    - Robust autonomous navigation with accurate perimeter detection.
    - Fully functional coin counter logic integrated with servo activation.
    - Clear visual and auditory feedback: 7-segment counter, tail lights, and speaker audio cues.
    - Effective remote control system using JDY-40 radios and joystick inputs, ensuring seamless communication between master and slave.

4. Weaknesses of our project:
    - Battery drain occurred faster than expected during extended autonomous operation, requiring multiple replacements.
    - Detection distance sensitivity: Detection performance dropped sharply if the coin was slightly misaligned or too far from the inductor.
    - The robot struggled slightly with smaller coins like the dime due to a smaller change in frequency when detecting the dime.

4.0 Live-Long Learning

For the base functionality of the robot, the team learned to use an H-Bridge circuit to control the servo motor's rotation and the Colpitts Oscillator circuit to detect metal to find the coins. Additionally, the team learned how to build a perimeter detector circuit to detect the boundaries of the coin-collecting area for the automatic collection task. Our understanding of MOSFETs from ELEC201 and CPEN211 greatly helped us learn how the H-Bridge circuit works, while our knowledge of RLC circuits from ELEC201 helped us grasp the Colpitts Oscillator circuit more easily. The team's knowledge of the C programming language from CPSC259 helped greatly with understanding how to program the microcontrollers in this project and integrate PWM in C to control the motor's speed of rotation. Furthermore, learning about how to use Git for version control and collaborating with teammates on coding tasks on Github from CPEN211 labs helped us manage our code for this project when we had several team members working on different parts of the software at the same time. One new thing that we learned that wasn't covered in previous coursework was how to use the JDY-40 wireless transmitter to send and receive commands wirelessly between two microcontrollers.

5.0 Conclusions

Our project consists of a robot base and a remote controller. The robot has two modes: automatic and manual. In automatic mode, its task is to pick up 20 coins in the perimeter wire. While the robot is moving around, the frequency of the colpitts oscillator circuit changes when the coin gets close to the inductor. Then, the robot stops and activates the coin-picking mechanism. After the coin is picked up, it continues to go forward. When the perimeter wire gets close to the inductor for perimeter detection, voltage change is detected by the peak detector circuit, and it triggers the robot to stop, back up, and then turn at a random angle to the left. After it picks up 20 coins, it goes into manual mode. In manual mode, the user can operate the robot using a joystick on the remote controller. LCD displays the frequency of the Colpitts oscillator, and the buzzer beeps faster and faster when the coin gets closer. There are also two buttons on the remote control, one for triggering the coin-picking mechanism and one for switching modes. The signals are transmitted using a JDY-40 radio. EFM8 MCU is used for the robot base ,and PIC32 MCU is used for the controller. In addition to the basic functionalities described above, we have added a few more features for a better design. Firstly, we attached an ultrasonic sensor at the front of the robot to detect any obstacle in its way. If an obstacle is detected, it turns 360 degrees. Secondly, we used another EFM8 board to configure a 7-segment display to show how many coins have been picked up. Lastly, we attached two LEDs on the rear of the robot to indicate direction. During a project, we encountered several problems, and one of the major issues was inconsistent ADC readings in the metal detection circuit. The problem was the resistance value, which was too low for the circuit. We replaced the 1kΩ resistor with a 100kΩ resistor and stabilized the base frequency. With all this hard work, the project took more than 3 weeks to be completed.

Bibliography

Calviño-Fraga, J. (2024). *Project 2 – Slides from March 14th* [Lecture slides]. Electrical Engineering Design Studio I, University of British Columbia.

Calviño-Fraga, J. (2024). *Project 2 – ELEC291_Project_2_Coin_Picking_Robot_2025.pdf* [Lab handout]. Electrical Engineering Design Studio I, University of British Columbia.

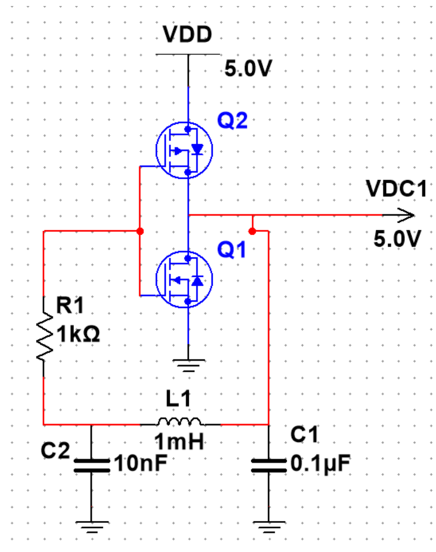
Calviño-Fraga, J. (2024). *Project 2 – Robot_assembly.pdf* [Lab handout]. Electrical Engineering Design Studio I, University of British Columbia.

Calviño-Fraga, J. (2024). *Project 2 – Coin_picker_assembly.pdf* [Lab handout]. Electrical Engineering Design Studio I, University of British Columbia.

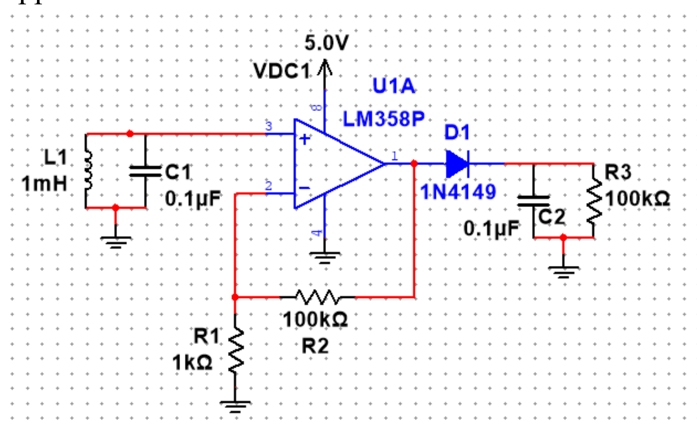
Calviño-Fraga, J. (2024). *Project 2 – Making_the_electromagnet.pdf* [Lab handout]. Electrical Engineering Design Studio I, University of British Columbia.

Appendices

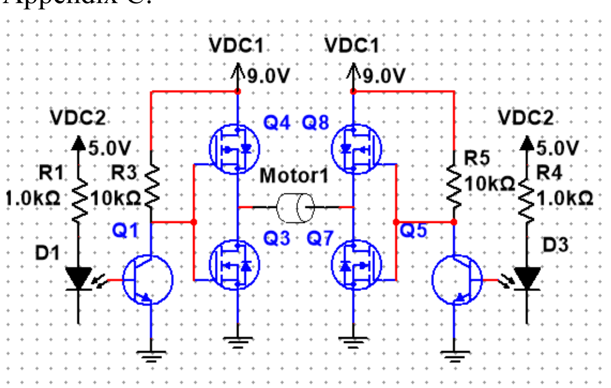
Appendix A:



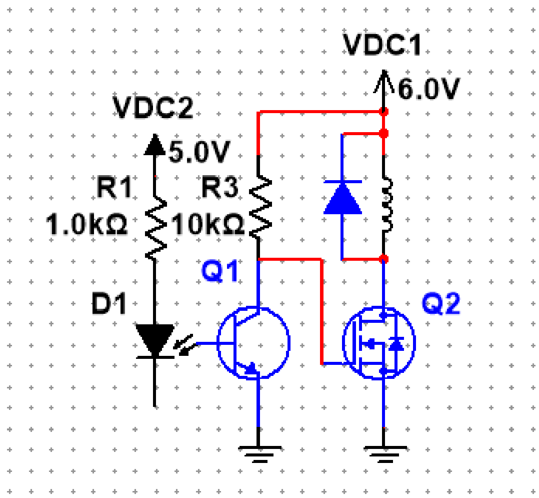
Appendix B:



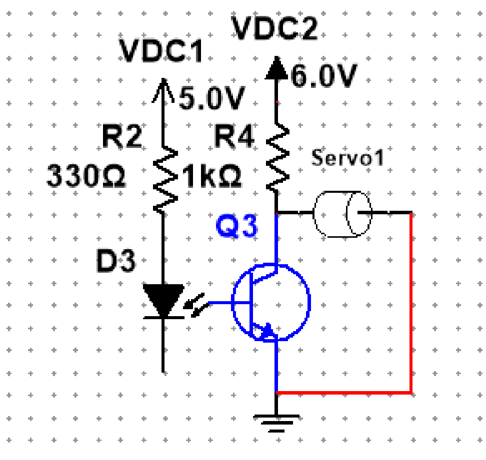
Appendix C:



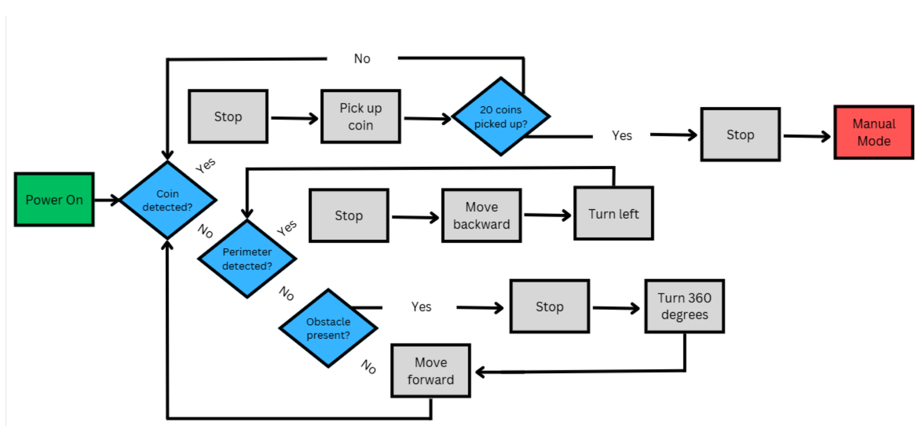
Appendix D:



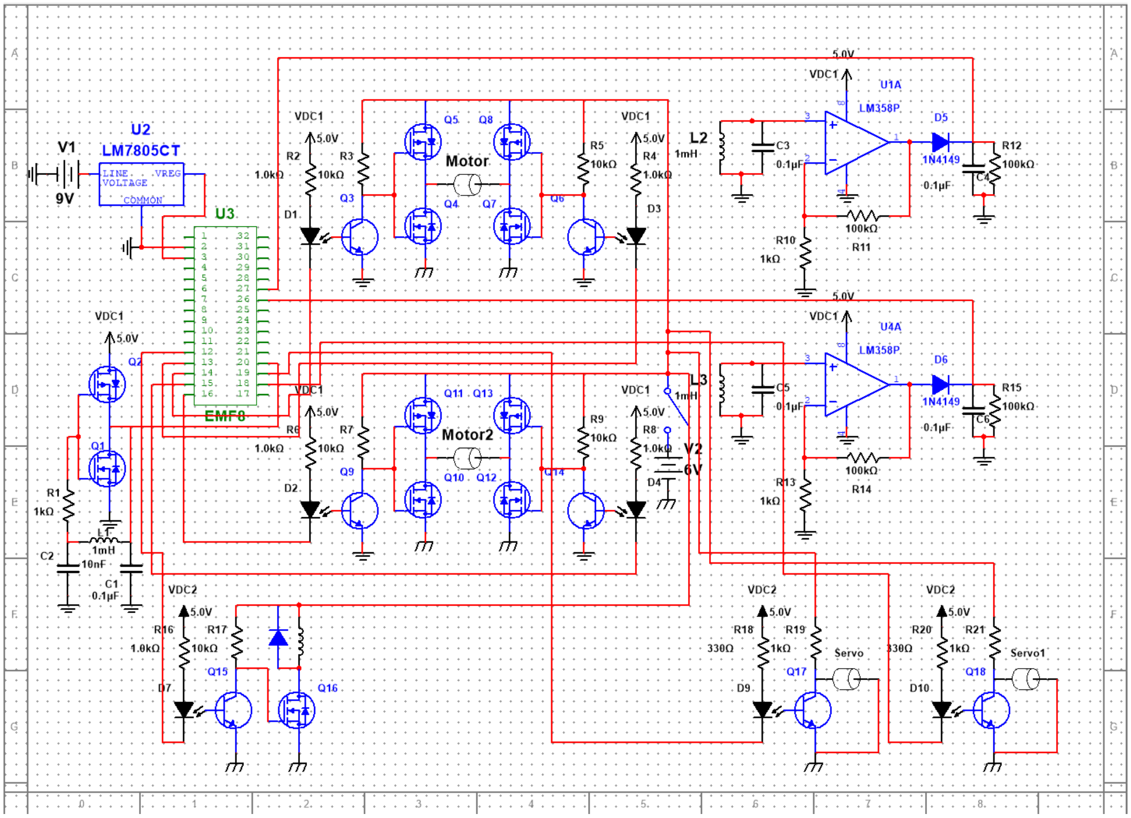
Appendix E:



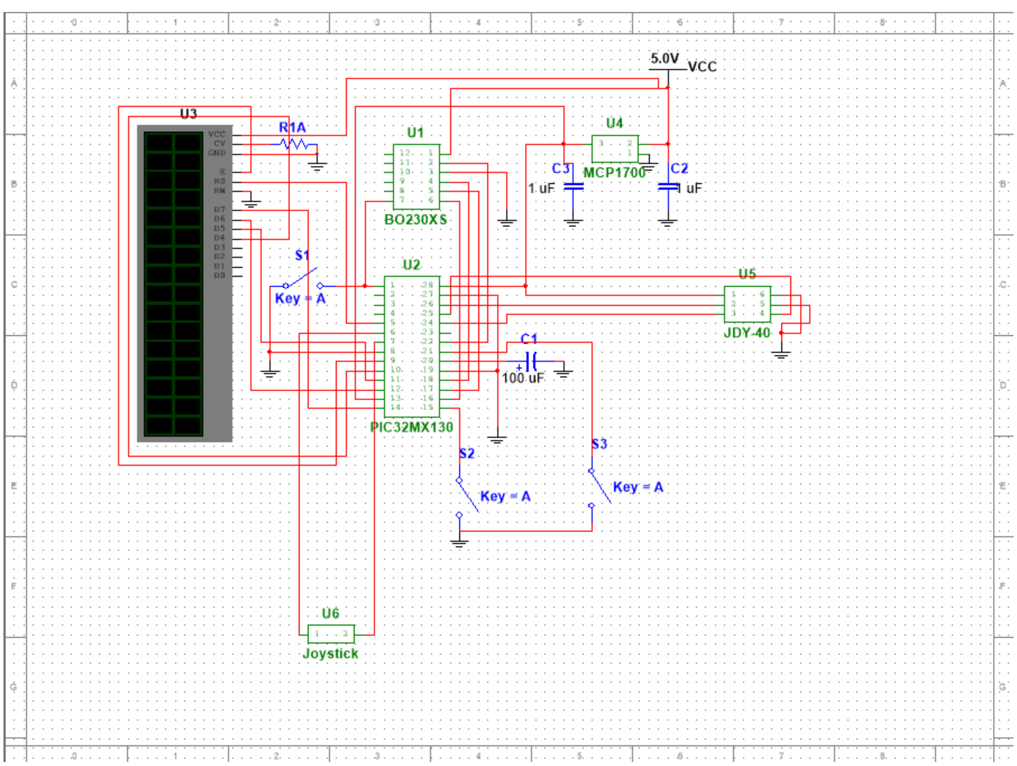
Appendix F:



Appendix G:



Appendix H:



Appendix I:
Robot Base Code

Appendix J:
7 Segment Display Code

Appendix K:
Controller Code