**University of British Columbia**

**Electrical and Computer Engineering**

**Course Name: ELEC 291 Electrical Engineering Design Studio I W2021**

**Instructor: Calvino-Fraga, Jesus**

**Section: L2B**

**Project Name: Coin Picking Robot**

**Authors: Mehmet Berke Karadayi (77766392), John Ye (43883347), Shuo Wu (30408751), Xuchen Zhang (61531547), Jack Ye (12815999), Sizhen Yan (22164982)**

| **Name** | **Percentage Points** | **Signature** |
| --- | --- | --- |
| **John Ye** | **103** | **John Ye** |
| **M.Berke Karadayi** | **95** | **M.Berke Karadayi** |
| **Shuo Wu** | **102** | **Shuo Wu** |
| **Xuchen Zhang** | **97** | **Xuchen Zhang** |
| **Jack Ye** | **101** | **Jack Ye** |
| **Sizhe Yan** | **102** | **Sizhe Yan** |
| **TOTAL** | **600** |  |

**Date of Submission: April 8, 2022**

# **2. Table of Contents**

1. Title page
2. Table of Contents
3. Introduction
4. Investigation

A) İdea Generation

B) Investigation Design

C) Data Collection

D) Data Synthesis

E) Analysis of Results

1. Design
2. Use of Process
3. Need and Constraint Identification
4. Problem Specification
5. Solution Generation
6. Solution Evaluation
7. Detailed Design
8. Solution Assessment
9. Live-Long Learning
10. Conclusions
11. References
12. Appendix

# **3. Introduction**

The main objective of the project is to build a robot that detects and picks coins controlled by the PIC32 microcontroller system. The robot will move around the surface delimited by the perimeter wire. The robot will pick randomly placed coins with the help of an electromagnet and micro servos to pick coins. A number of the coins will be displayed on the LCD screen, and a LED light to show if a coin is being detected or not. This will allow the user to easily visualize the number of coins we picked and whether we have detected a coin or not.

The project consists of two main parts, which are software and hardware. In the software block, we maintained a connection between hardware and breadboard, which includes some function to detect and picks coins. C programming language is used to implement wires, the functionality of servo motors and the moving of the robot.

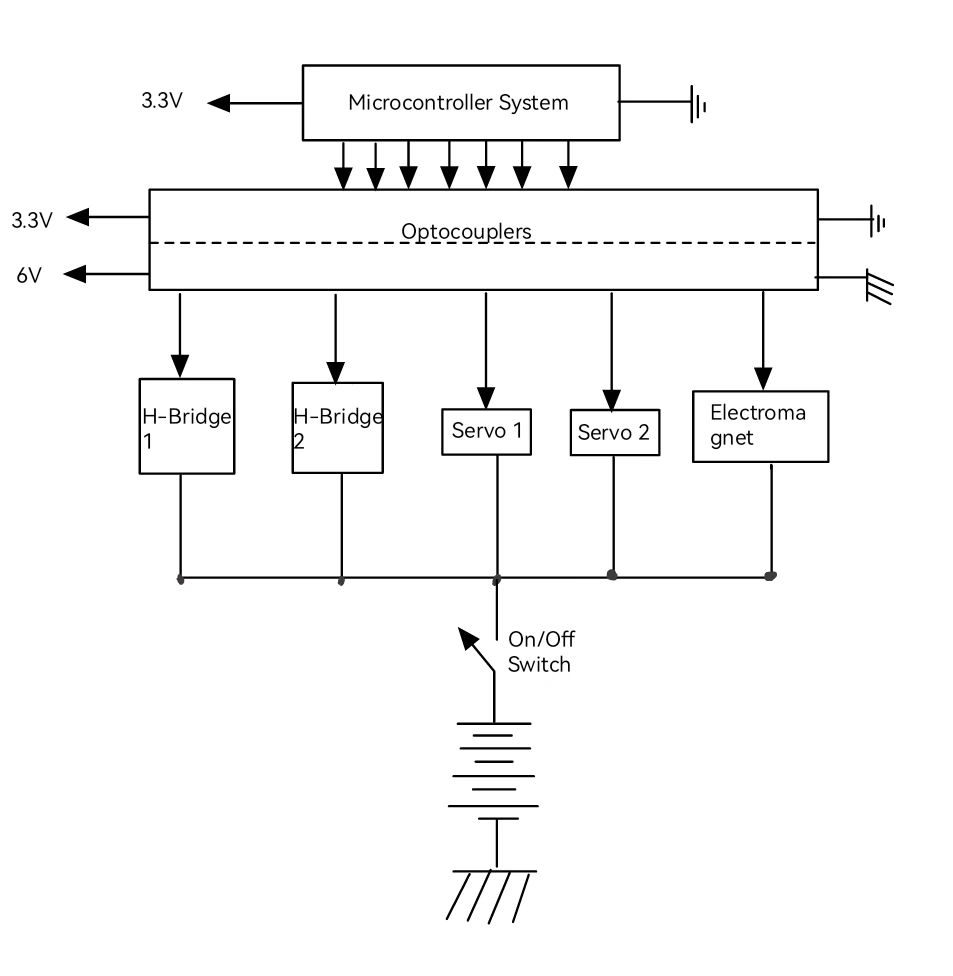


Figure 7:Hardware block diagram

# **4. Investigation**

1. **Idea Generation**

Our project group created a discord channel to discuss the project. We brainstormed before the lab hours to plan out the details. We divided into three groups to make an assembly part. Each member had a specific task to present in the next lab section. Then, we created both coin picking and robot assembly. Apart from lab hours, we met in a common room at the weekends to discuss the process, and we worked on the coding part by visual studio code online and Github. Therefore, each member can interact effectively.

The work is split between three groups in the following ways: micro servos, metal detector/perimeter detector, and servo motor.

1. **Investigation Design**

We mainly focused on the professor’s slides and videos to analyze the significant details of the design. We gathered our data from online research, lecture slides and some tests performed in the lab with the lab equipment, and this will be discussed in detail in the Data Collection section. All of us attended lab hours to work on the design and ask for any help from the professor and teaching assistant.

1. **Data Collection**

Our team collected data during the testing and debugging stages of our design through the use of lab tools and previous lab programs. Listed below are some of the parameters we need to measure:

* 1. Servo arm angle measurement
  2. Inductor’s oscillating frequency when different coins are under metal detector
  3. The speed in the cars turning
  4. The voltage of perimeter detectors with and without active perimeter.

1. **Data Synthesis**

Our team synthesized data and information to reach appropriate conclusions through consistent testing of our soldering oven controller, obtaining results through our methods detailed in Data Collection, and comparing the consistency of those results to expected values from design documents and circuit diagrams.

We tested our individual components before implementing them into our main robot.

To measure the voltage of the inductor, we tested it separately on a board with the circuit for us to measure its voltage. We also used the equipment in the lab to limit the effect of the oscillating which will be present in the vpp signal in the oscilloscope.

To measure the frequency change when an inductor gets close to a coin, we simply wire up our circuits in another board and connect it with our inductors on cars and record the data that shows in the putty.

Since the moving angle of our servos is only 180 degrees, and in this project, we need to use the full 180 degrees, so we tested the brunch of the angle of the servo and physically adjusted the servo angle to fit the needs.

1. **Analysis of Results**

Our team appraised the validity of conclusions through rigorous testing and troubleshooting using a multimeter and comparing to expected results. Data were synthesized via testing throughout our design process.

Although we implemented lots of tests and measurements throughout the whole process, we found that the data we measured in the early stages are slightly off during the final stage when we put all the components onto the board because of the effect of the difference in voltages and noises. To solve this problem, we did lots of data debugging after we integrated the car.

**5. Design**

1. **Use of Process**

To create a good design, we grasp the engineering design project that is introduced to improve the project. We all know that the success of the project requires a well-design procedure, which is a great way to begin. As a group, we first tried to understand the major and minor challenges of the project. We investigated the prior labs, which are relevant to this project at some point. We also looked at lecture slides and videos to have specific information about the project that we need to overcome. Once we made plans based on specific information, we tried to design a prototype of the coin-picking robot to see if it is good enough for the design needs. If it does not satisfy the needs and requirements, we would re-analyze the information and make a new plan for the design.

Once we determined the prototype of our required design, we added extra functionality to make our solution more efficient and better.

1. **Need and Constraint Identification**

We analyzed the professor’s lecture slides and videos in detail to grasp all relevant information and requirements on the process. After a detailed analysis of these specifications, we thought about the design direction of the project and the basic functions that the product should have in the end.

Regarding specific requirements of the project, our product should have the following functions:

1. Coin detecting
2. Map perimeter detecting
3. Machine arm for picking up the coin
4. Function to move properly
5. LED to light up when picking up a coin
6. LCD screen to show the number of coins we have picked
7. **Problem Specification**

To enhance users’ visual experience, we added an LCD screen showing the number of the coins we have picked and a LED light to show if a coin is being detected or not. This will allow the user to easily visualize the number of coins we picked and whether we have detected a coin or not. Counting numbers with the speaker to enhance the auditory experience.

Distinguish the type of coins we are picking up, this problem is associated with the type of metal and amount of material used to make the coin. The solution generation for this additional feature will be based on this quantitative test.

1. **Solution Generation**

To meet stakeholders' requirements and need, our group generated the following solution to fulfill the needs and requirements.

**LCD:** The code from atmega8051 can be modified so that LCD be used by the pic32 microncontroller.

**LEDs:** Using a pin from PIC 32 to light up when the coin detected signal is 1.

Speaker: Adding a register component as a memory box and saving the data of the voice of several coins in it. Set an int variable “Number” to determine the number the speaker will make.

**Coin Types disfiguration:** To accomplish this function we need to measure the average frequency every unit time to see the value fits what type of coins.

**Coin Detector:** Our idea is to measure the frequency change of an inductor when it comes close to a metal coin because it will change the magnetic field which will be reflected as a change in the oscillator frequency.

**Perimeter Detector:** To detect if we reach a map edge or not, we simply measure the voltage out of the peak detecting inductors since the inductors will be affected when they get close to a perimeter wire.

**Servos:** The professor’s provided an example for controlling servos. However, we need to control two servos in the program.

**Motors:** The ideal generated at the beginning is about the functions needed for the motors, the wheels need to be able to move forward, backward, and change the direction of moving if perimeter or coin is detected. For turning the direction, one wheel needs to hold still and the other wheel will move.

**Electromagnet:** In order to meet the requirements of grabbing coins, we are going to use electromagnets to achieve this function. Electromagnets use electromagnetic attraction generated by current-carrying coils of iron to attract metal coins. After we referred to the professor's courseware, we made the electromagnetic coil according to the steps. Copper wires are wound around the outside of the core to make conducting windings and lead to both ends of the coil to connect the circuit board. An electromagnet becomes magnetic when a current passes through the coil. The pins of the electromagnetic coil and the power supply are linked on the circuit board via LTV-846 and N-MOSFET. And we added a diode between the two pins of the electromagnetic coil. It is worth noting in this section that we need to use THE TVT-846 to link the pins of the electromagnetic coil to the power supply. If not used, this can cause the N-MOSFET to overheat and the magnetic force of the electromagnetic coil to decrease.

1. **Solution Evaluation**

It was difficult to evaluate the potential solution without creating the design. The most difficult part of this project was connecting software with hardware from our perspective. While improving the robot, we faced crucial errors in the picking of coins. We found that it was difficult to arrange the correct range to pick up the coin by controlling the hardware and software. We spend much time on this to properly solve these errors.

For the solution generated during the last part for wheels, we notice that if one wheel holds still and the other wheel moves will make the radius of the turning circle too big, and the friction of the wheel might not be able to support the torque, the robot may slide on the testing field.

Another serious problem is that when we turn on the power switch, the voltage converter LM7805 and the P-MOSFET and N-MOSFET of the metal detector begin to heat up, and the brightness of the LED lamp begins to weaken gradually, and the metal detector works abnormally. Only the wheels and motors work. We spent a lot of time checking the links in the electronic circuits. After making sure there was no problem with the circuit board links, we removed the circuit boards and began to check the battery links inside the body. Finally, we found that the positive pole of the 9V battery clip was connected to the screws of the car body, resulting in a short circuit in a part of the circuit.

The solution to this problem is to wrap the 9V battery clip with electrical tape, which can prevent the battery from contacting other metal parts. At the same time, we fixed the point.

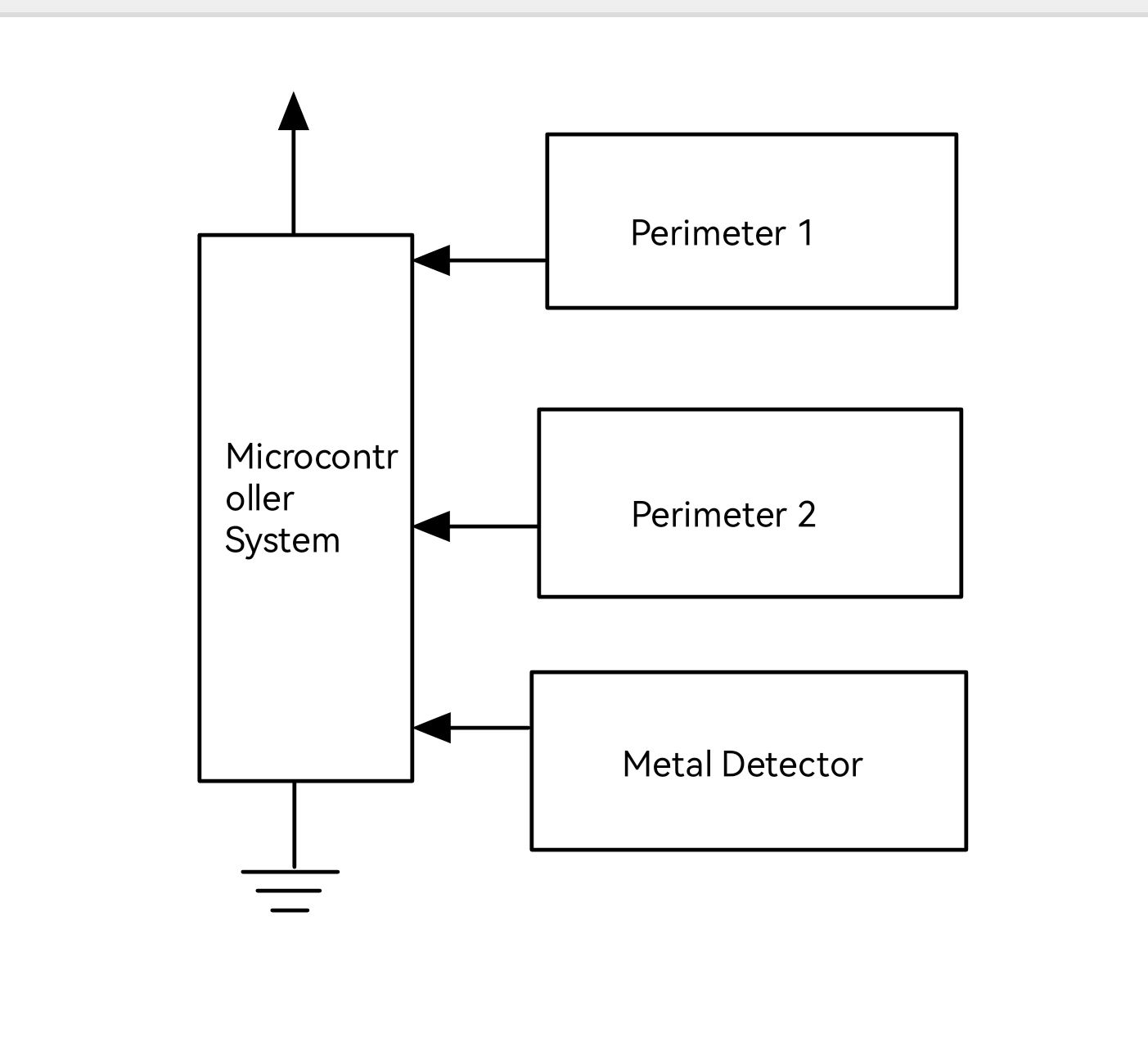
When we are testing the servo arm on the robot, we found that it was unresponsive. We tried the same software on the testing board, and the servo works as expected. This findings gave us a suspicion that the motor was interfering with the servo’s signal. To fix this, we implemented a second octocoupler to separate the signal of the pic32 microntroller from the noises of the motor. This implementation also allows for the increase in effectiveness of the electromagnet.

When we solved the 9V battery short circuit problem, we found a signal interference between the metal detector and the motor. When the metal detector is attached to the circuit board, the motor gear rotates in the opposite direction as before the metal detector was attached. At the same time, one of the edge detector's electromagnetic induction coils is not working. This results in our robot not being able to sense the boundary at certain angles and going out of the boundary. This problem also caused the voltage output of this part of the amplifier to be abnormal. After group discussion and calculation, we suspect that the type of parallel capacitor at both ends of the electromagnetic induction coil of the metal detector is wrong, which affects the frequency of induction.

After communicating with the professor, we measured the frequency at both ends of the capacitor through an oscilloscope and finally decided to replace the original 100nF capacitor with a 1nF capacitor. After replacing the new capacitor, the whole circuit can run normally.

1. **Detailed Design**
2. **Hardware**

| Parts | Quantity |
| --- | --- |
| PIC32 Microcontroller | 1 |
| Electromagnet | 1 |
| Coin Picker assembly | 1 |
| Servo motors | 2 |
| Servo wheel | 2 |
| Inductors | 3 |
| Metal Detector | 1 |
| MOSFET | 6 |
| Perimeter Detector | 1 |

****

We spent most of our time on the hardware. Individually, the hardware components are quite straightforward, but piecing them together brings many issues. Fortunately, we were able to solve those issues in the end.

**Metal Detector**

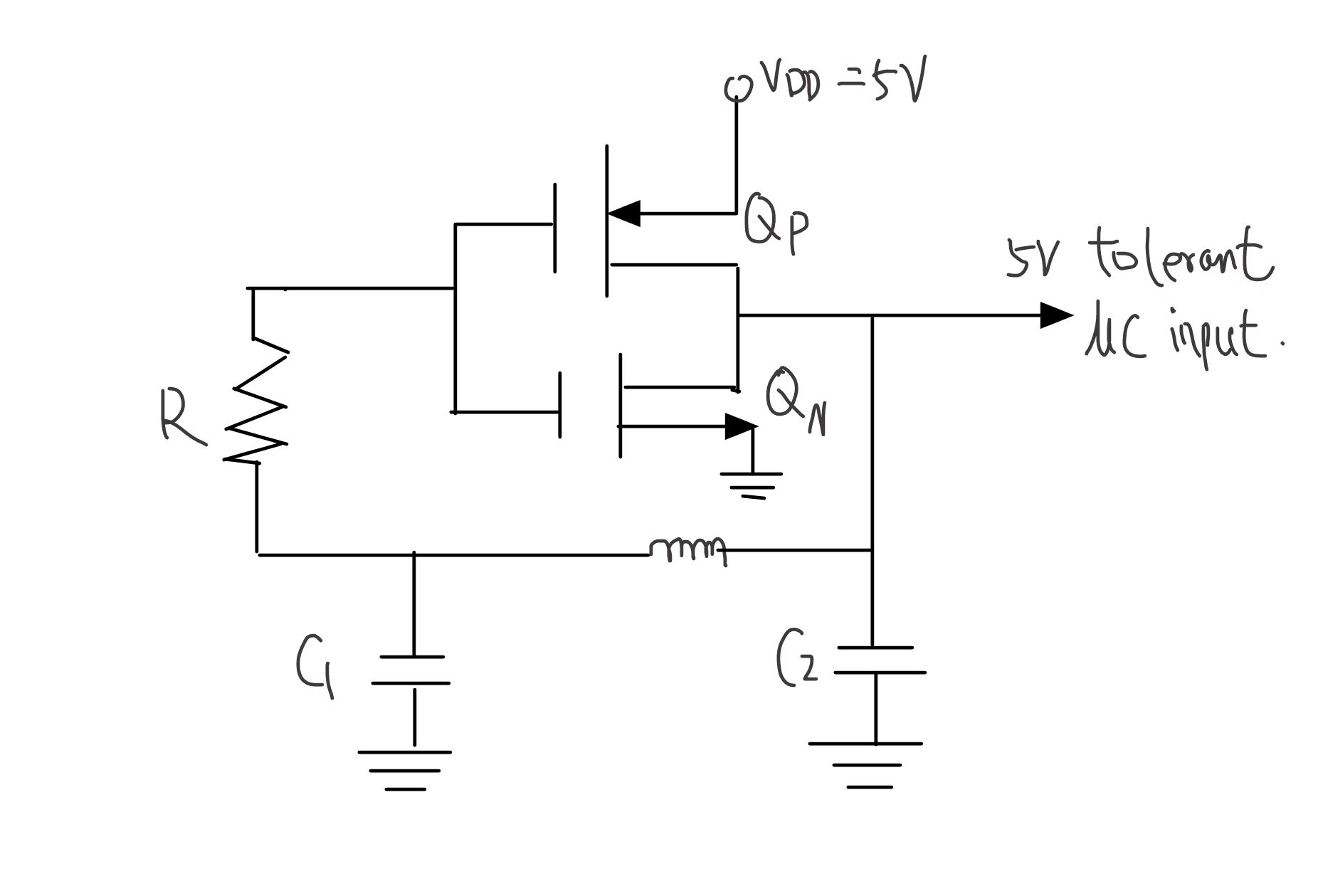
Metal detector consists of A CMOS inverter (as a NOT gate), a resistor, two capacitors and an inductor. The theory of metal detecting is that when the coins come close to the inductor, it changes the value of the inductor which causes the frequency to decrease. All the components should be installed on the breadboard except the inductor. The inductor is installed at the front of the bottom of the car to detect the metal in the forward direction easily. After installing the inductor under the car, connect it with two capacitors seriously by wires. 

Figure 7: The perimeter detector circuit

**Perimeter detector**

The main structure of the perimeter detector consists of the amplifier, RC filter and a tank circuit. Two perimeter detectors are installed at the bottom of the car perpendicularly to make sure we always detect the wire no matter the car reaches the edge from what angle.

The tank circuit is a parallel connection between an inductor and a capacitor. When the reactance of the capacitor and inductor are equal to each other, there is only one stable frequency. In this project, the frequency should be 16kHz which is equal to the frequency of the AC source. Based on the equation, the capacitor should be 98.946 nf, which we use 100nf. When the inductor gets close to the perimeter wire, the voltage out of the peak detector increases. Since the peak of voltage is too small to read and analyze by the microcontroller, the amplifier is used to enlarge the signal. The ideal non-inverting gain is over 48, and we used one 2k ohm and 100k ohms resistor to achieve 500% gain.

**Servo Arm and LCD**

The servo arm needs three connections: VCC, GND and PWM signal. We connect the VCC to 6V and we connect the PWM signal to pin RB15 and RB14 of the pic32 microcontroller. This allows the servo arm to move to a specified signal.

The LCD requires 10 connections: 2 pins connecting to GND, 1 pin connecting to VCC, 1 pin connecting to GND with a 2k resistor, 4 data pins connecting to RB13, RB12, RB10 and RA4, RS connecting to RA2, E connecting to RA3. The LCD is set to 4-bit mode so only 4 data pins are required.

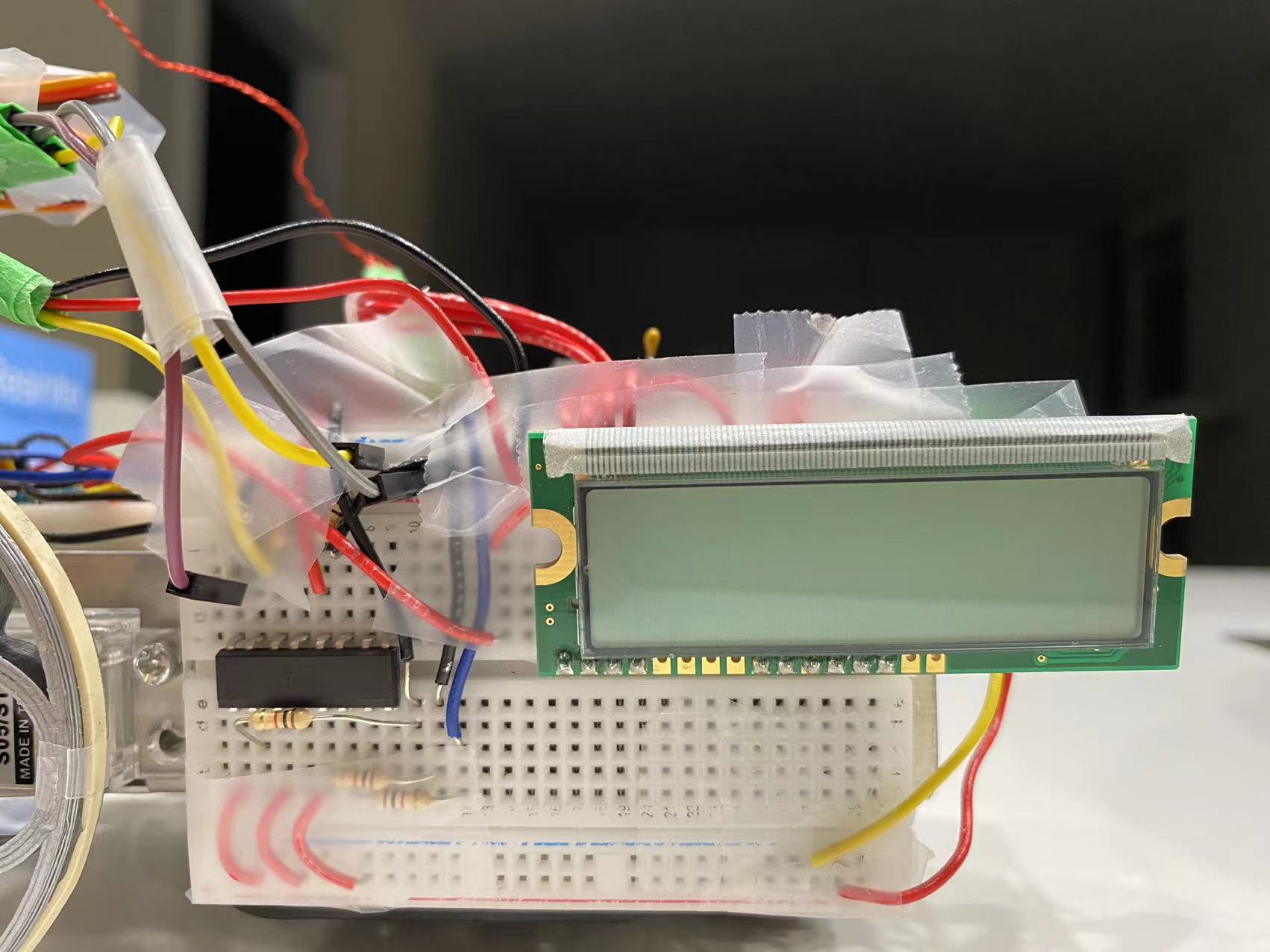


Figure 1

The Connection of Servo Arm and LCD. Note that an optocoupler is used to keep the noises from the motor from the signals of the servo arm.

**Optocoupler**

We used two Octocoupler to interface the output of the microcontroller to the electronics. The working principle of the optical coupler is to encapsulate the luminescent device and the optical receiver in the same shell. We usually need to link the output pin of the control chip to the input pin of the luminescent device and connect the output pin of the optical receiver to the electronic component. When the control chip generates a certain voltage, the luminescent and receiver start to work to control the operation of electronic components. In other words, it is a device that converts electrical signals into light signals and then into electrical signals. The unique working principle of the optical coupler can isolate the input and output well, which makes it have a good anti-interference ability. In this project, we used LTV-846.

**Servo Motors**

Each of the servo motors consists of two sets of H-bridges, the H-bridge is made by N-type and P-type MOSFETs, these two sets of MOSFETS allow the current to flow in both directions: one current direction makes the servo rotates clockwise and the other direction make the servo rotates counterclockwise. However, with only the H-bridges, the voltage of the signal is not big enough to make the wheel rotate. As a result, we implement it with Optocouplers. The signal will be amplified after it crosses the Optocouplers. As a section summary, one servo motor needs 2 sets of MOSFETS and one Optocoupler to make the amplified signal transmit back and forth.

1. **Software**

Many aspects of the software are inspired by Professor’s lecture slides and sample pic32 projects. We also found useful functions online to aid the software development. Those will be listed below and in the references.

**- Servo Arm**

The servo arm requires a PWM-driven signal which uses interrupts to move the arm. This requires the built-in timer in pic32. (Codes for these two parts are from the professor’s sample code).

Once the servo arm is correctly initialized, it can then be controlled by the pic32. To change the position, we just need to input a number in the range of 60 to 260. (Codes in figure 2 of appendix)

While testing the robot, we found that the arm was not reliable in picking up the 2 dollar coin due to the coin’s weight and the arm’s speed. To fix this, we implement a simple function that changes the value of the PWM signal by 1 degree every 6 milliseconds by using a while loop.

**- Blinking LED**

The LED was initialized like a normal output pin. To achieve a blinking effect, the code above toggles the LED on and off with a half-second delay while moving the arm. (See figure 2)

**- Printing on LCD**

The pic32 only has 28 pins available. To save on pins, we decided to configure the LCD in 4-bits mode. We follow the same initialization of LCD on atmega8051 (provided by the professor). The LCD requires 2 digits hexadecimal inputs, and since the pins on pic32 that connect to the data pins are configured as a 1-bit output, we found a function online that extracts the bits from a hexadecimal number.

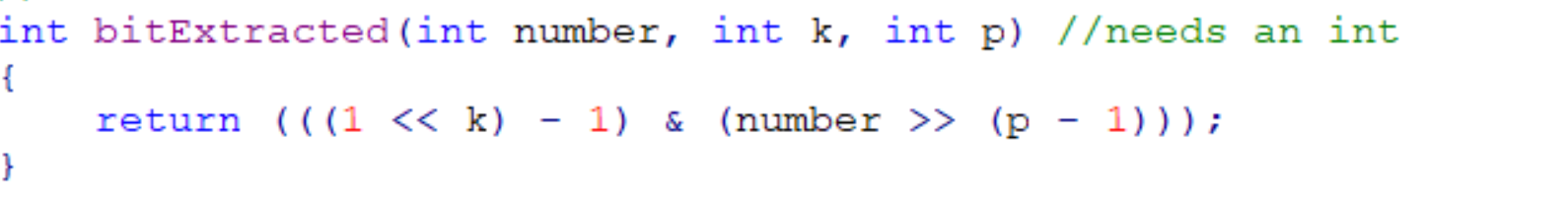


Figure 7

Once we obtained the function that sends data to the LCD, we can then use the LCDprint function to print the messages on our robot.

**- Detecting Coins**

Though the theory is to measure the frequency change, through experimenting, we found a value f the metal detector without any coin and we use that as our threshold value. This function would return 1 to the function which gives a signal to other functions that a coin is detected. (The details of this function can be found in getCoin() in the source code, figure 3).

**- Detecting Perimeter**

To measure the voltage generated by the perimeter detector, we used an op amplifier here and we used Pin AN5 from the PIC 32 chip. We used an Edgecounter here as a timer.

Through experimenting, we were able to determine a threshold value , which is the highest potential difference when no perimeter wires are nearby. With this value, we were able to update the value currently detected by the robot and compare it to this threshold value. When the voltage is greater than the threshold, the robot will perform the sequence that allows the robot to back up and turn to avoid going out of bounds. (detail codes can be found in figure 4 of appendix)

**- Stopping the robot when 20 coins are picked**

Here we set a variable Coins as an index for counting the number of coins we have collected. When this index gets to 20 which means the process has been completed, we wrote a program for the robot to dance by controlling the wheel and stop moving after a while.

**- Motors (control the movement of the robot)**

The code composed is shown in figures 3,4,5 in the appendix**.** Moreover, changing the direction of the robot can be implemented by making the wheels rotate in different directions. The purpose of “*waitms(50);*” is to make sure the signal will not be changed until the instruction is finished. Also, “MoveSlow()”; function used the feature of the servo motors: when the signal changes from 1 to 0, the servo motors will not be locked(standing still), it will keep moving because of the momentum of inertia.

1. **Solution Assessment**

To further increase the sensitivity of the metal detector, we added a sponge between the inductor and the car to lower the inductor.

We found that when testing the inductor without a power source from the board with the help of an oscilloscope in the lab, the voltage generated by the inductor is very obvious ranging from 0V to 3V and the circuit worked perfectly fine except for the part of the amplifier since we did not charge the chip. However, after we turn on the switch for the power supply, the output voltage would increase to 3.3V despite the effect of an AC-powered wire. After some debugging, we found that the error located at the metal detected part, the resonance frequency produced by the metal detector part was around 16kHz which directly had a significant disturbance for this part of the functionality. After we changed a capacitor to higher the frequency of the metal detector part, the perimeter detector worked reliably.

However, we found our data-updating date was really slow, so we modified the counter program in the edge counter function to shorten the update time gap and this correspondingly changed metal detecting which makes all the detecting functions work more efficiently.

**6. Live-Long Learning**

In the process of building a coin-picking robot, our group learned new concepts and developed these to apply to the robot. These are divided into two parts, which are the software and hardware section. Our experience from various courses helped us to develop the robot.

In software design, we believe that APSC 160 course was really helpful to write the suitable code for this project. We learned the C programming language in this course. It helps us to understand the code, and also write our code in some programs.

CPSC 259 course was also helpful in terms of software design. We learned data structures and algorithm, which contains pointers, trees, linked lists etc. These concepts were useful to understand the idea of codes.

We also think that CPEN 211 course was helpful to understand some logic of microcomputer terms. The logic behind interrupting and the connection of software on microcontrollers were some concepts that we used during the project with the help of the course. They helped us have a good foundation to understand the logic of code.

In the hardware assembly part, ELEC 201 helped us how to use the circuit board and reasonably. The course provides us master of the connection method of some basic electronic components.

ELEC 211 was also helpful for the hardware section. We learned the principle of electromagnetics within the scope of the course.

In addition, ELEC 291 directly guides us to provide suitable design projects. It helped us grasp the use of most of the basic equations and told us how all the hardware works. This increases the probability of success in the design and production of our projects.

Thanks to this course for helping us to learn different concepts. We still need to continue to improve ourselves to be ready for upcoming projects, which might be a capstone conference to design one of the best projects.

**7. Conclusions**

In this project, we build a coin-picking robot that detects and picks coins controlled by the PIC32 microcontroller system. The system picked randomly placed coins with the help of an electromagnet and micro servos to choose coins. Moreover, an LCD screen is added to show the number of the coins, and a LED light is used to show if a coin is being detected or not.

We approximately spent a total of 60-65 hours on this project. During the time, 5-7 hours were spent on discussion, 10 hours were spent on both assembly parts, and 15 hours were spent on the coding part. The last 15-20 hours were spent on hardware, demo and adjusting the extra functionality.

**8. References**

Dr. Jesus Calvino-Fraga P.Eng. Project 2: Coin Picking Robot, pp. 2-18, March 18, 2022.

Dr. Jesus Calvino-Fraga P.Eng. Laboratory 6 - The PIC32 Microcontroller System, pp. 2-7, March 18, 2022.

**9. Appendix**

Project Source Code (handed in on canvas)

Project Demo Video (handed in on canvas)

**Metal detector:**

(No)Stable Count=(7400-7500)

2$ (7200-6800)

1$ (7100-6700)

25 cents(7200-6700)

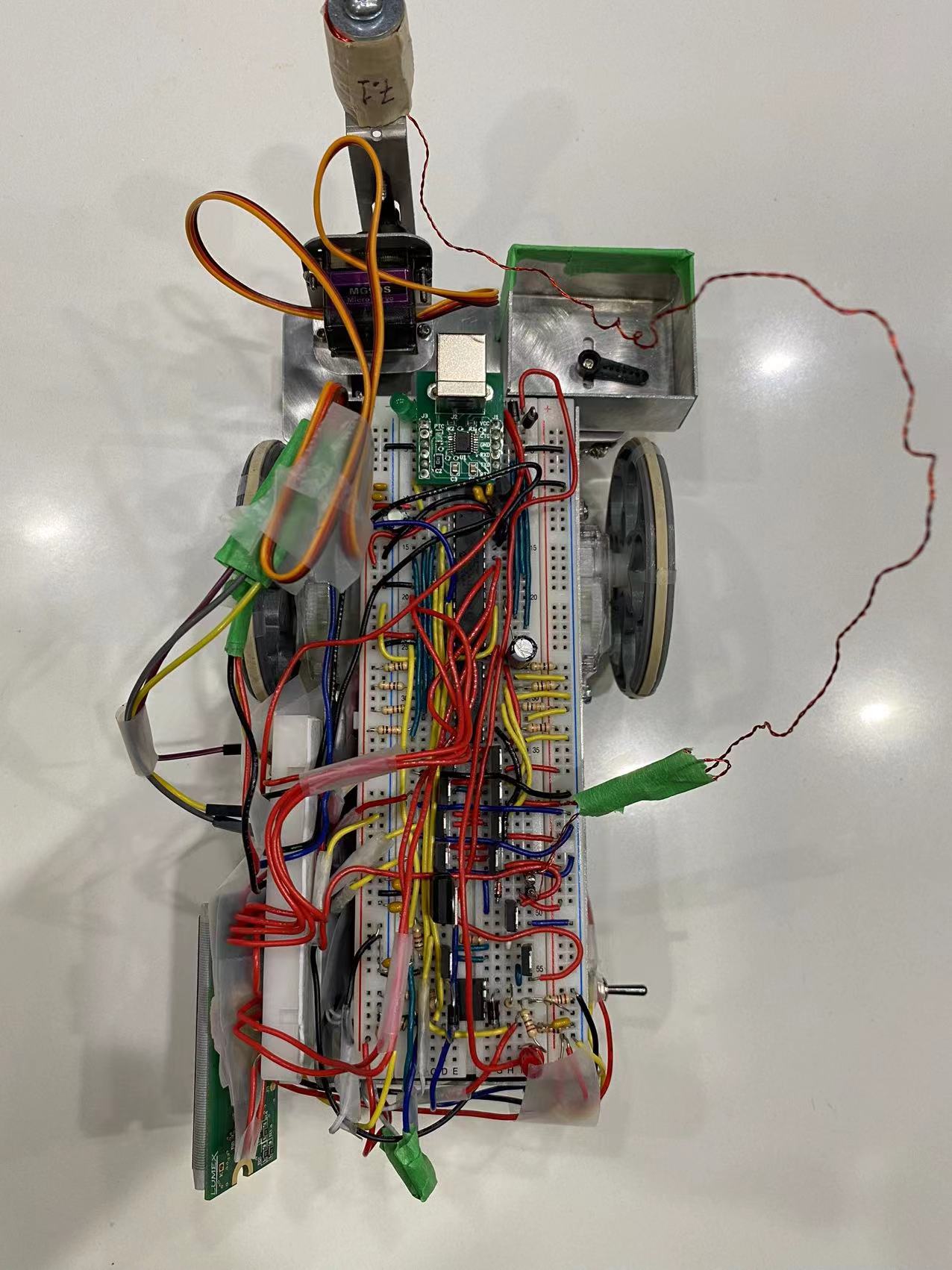
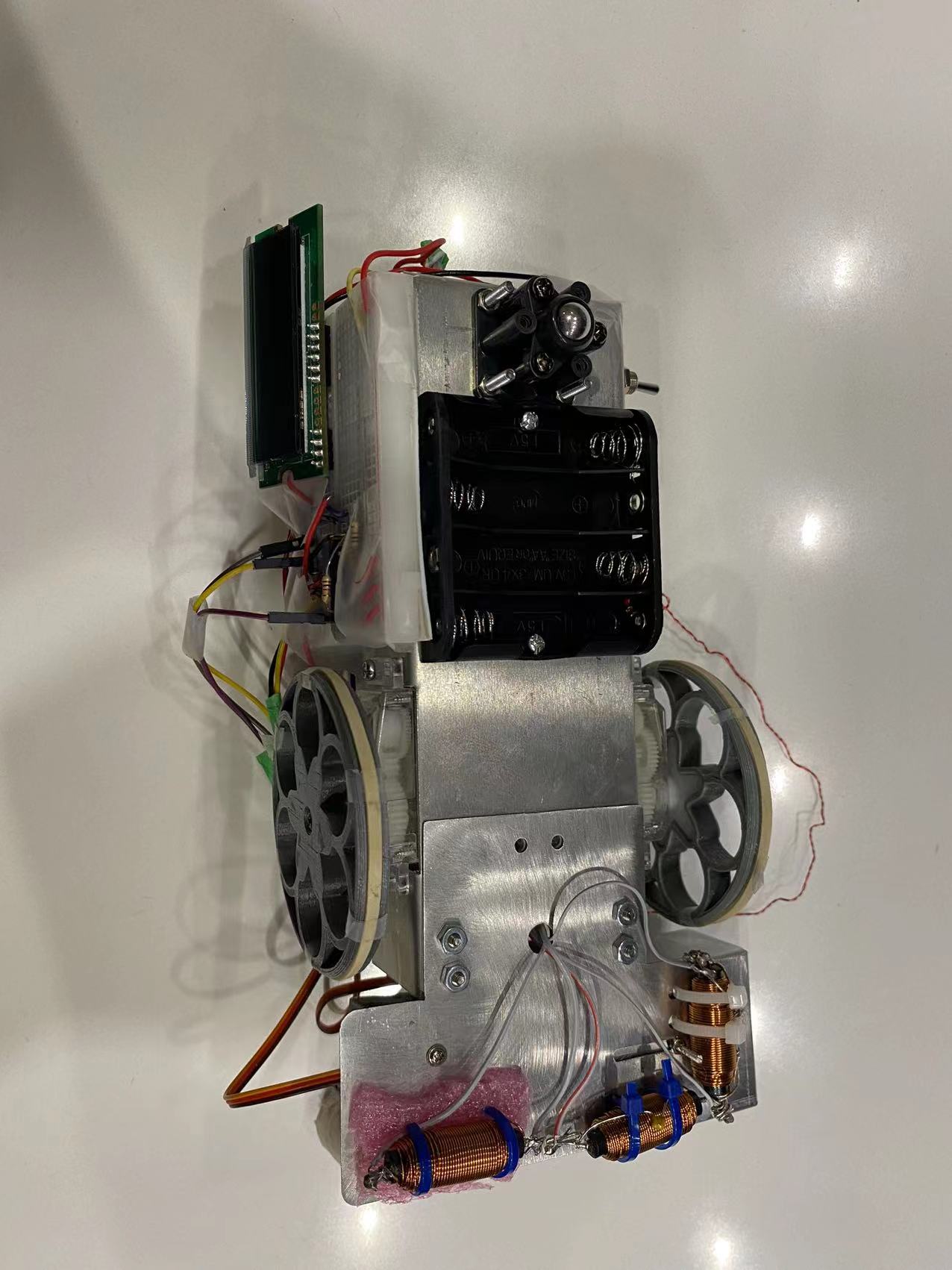
10 cents(7150-6700)

5 cents(7000-6500)

**Map:**

In the map:0V

Touch the map:0.1V

**Pictures of our cars (bottom views and top views)**

**C code part**

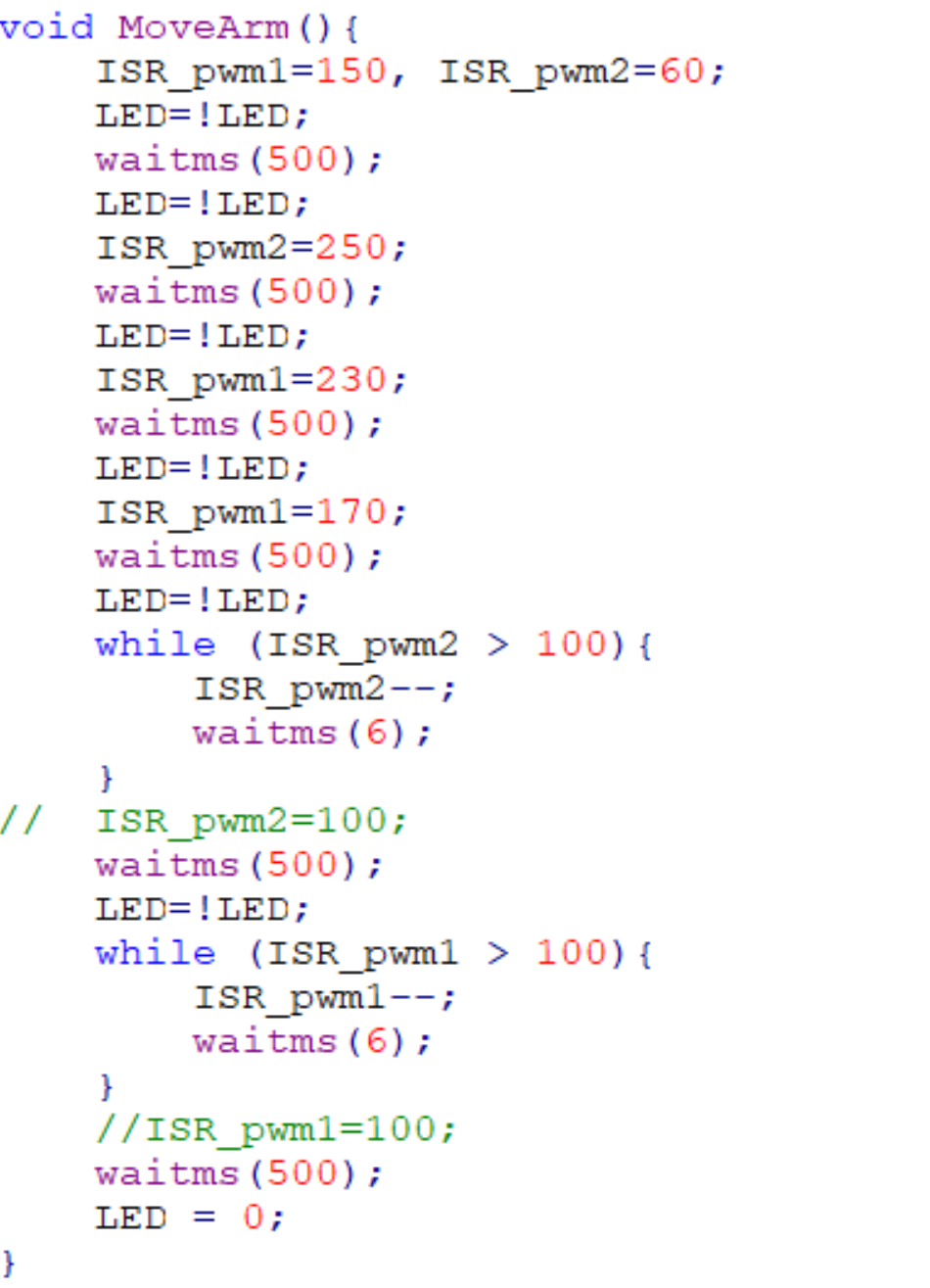


Figure 2

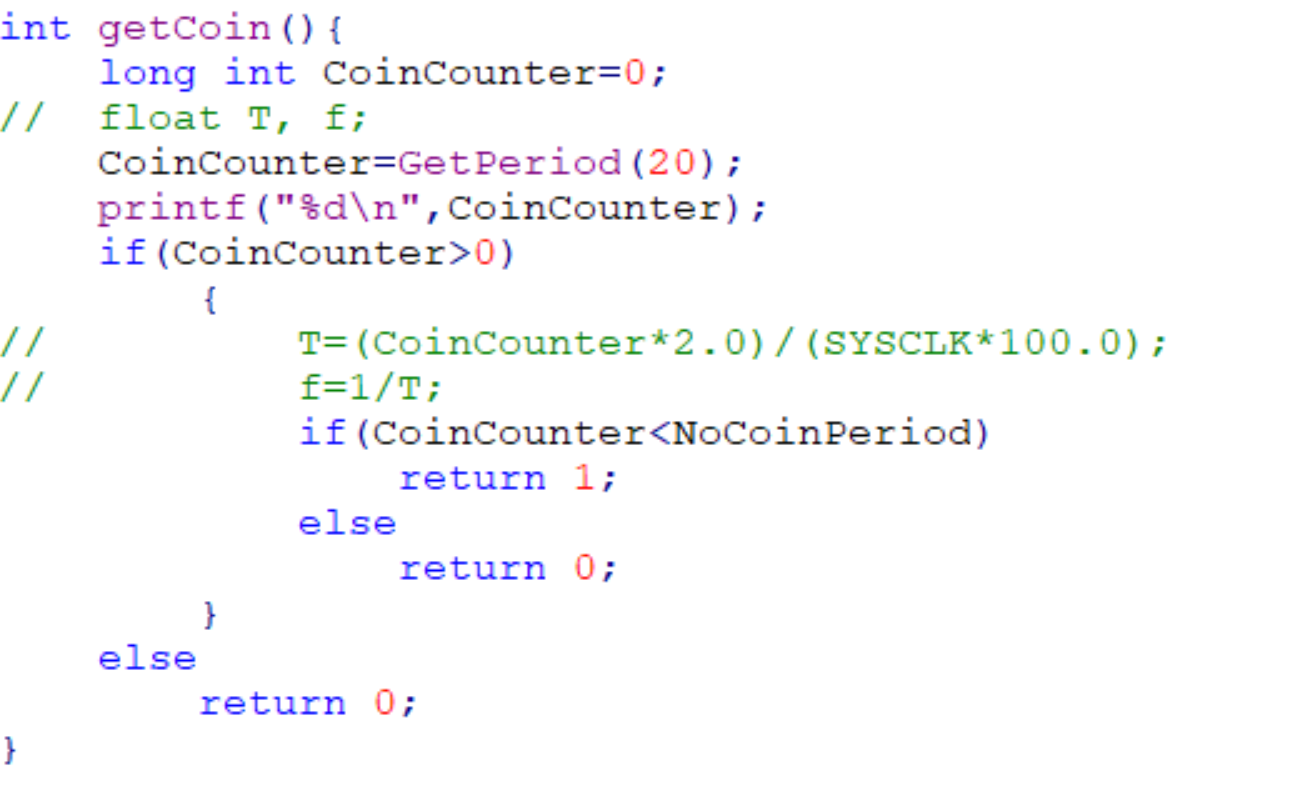
****

Figure 3

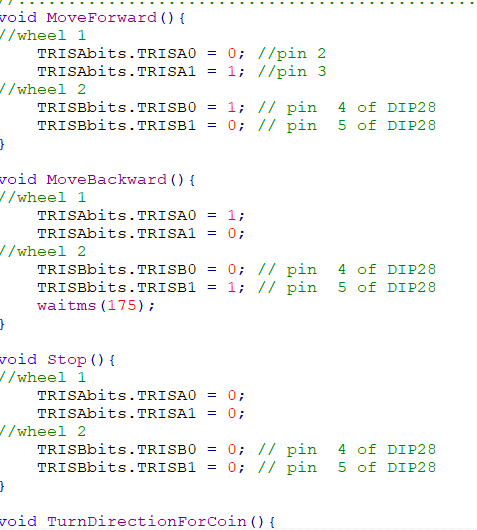
****

Figure 4

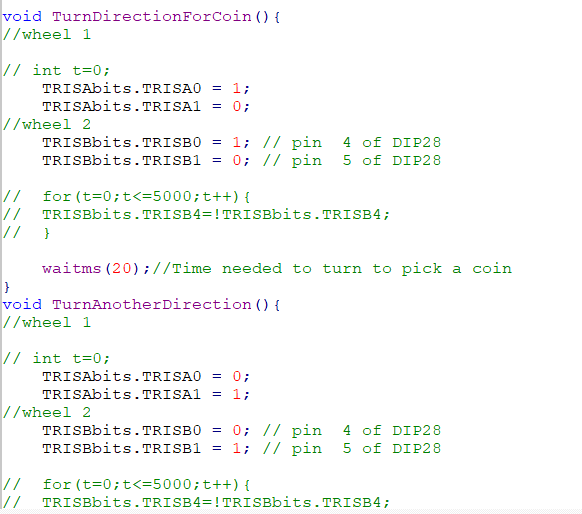


Figure 5

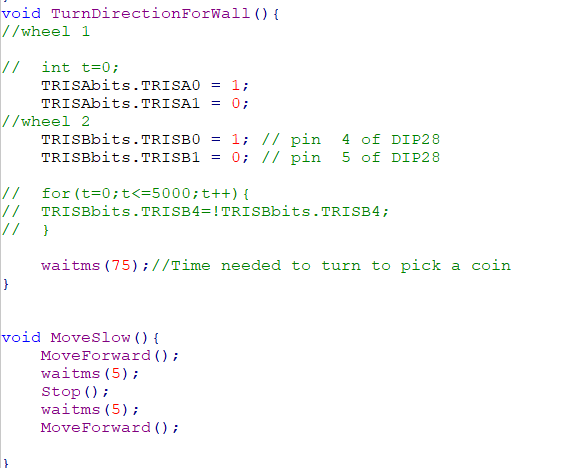


Figure 6