ECE 445

SENIOR DESIGN LABORATORY

FINAL PAPER

Infineon's Robotic Car

Team #30

KAI-CHIEH LIANG
(kcliang2@illinois.edu)
WEI-JUI CHEN
(wjchen2@illinois.edu)
SAHARSH BALLAE
(sballae2@illinois.edu)

Professor: Arne Fliflet
TA: David Null

Dec 6, 2023

Abstract

The following report describes the precise method to build a line following robotic car with obstacle avoidance and voice recognition capabilities using Infineon's PsoC6 micro-controller. This project seeks to demonstrate the cutting edge processing capabilities of Infineon's PsoC6 micro-controller by conducting multiple computationally intensive tasks on a single chip. The research methodology involves a meticulous examination of the Infineon PsoC6 micro-controller's capabilities, encompassing programming, integration, and real-world testing. The main findings of this research achieve the successful integration of line following, obstacle avoidance, and voice recognition simultaneously on the Infineon PsoC6 micro-controller. The inclusion of speed feedback control sub-modules not only enhances the robot's scalability but also contributes to achieving higher speeds with precision. The radar submodule, while employed in its rudimentary form for basic obstacle avoidance, demonstrates potential for more advanced applications, laying the foundation for future intelligent obstacle detection systems. The significance of these findings lies in advancing the field of robotics by showcasing the capabilities of the Infineon PsoC6 micro-controller and providing a blueprint for future creations of multifunctional robotic systems. The successful implementation of this project not only contributes to the existing body of knowledge in robotics but also explores the feasibility of integrating complex functionalities onto a single micro-controller, thereby reducing the overall complexity of robotic systems.

Contents

List of Figures

List of Tables

1 Introduction

1.1 Challenge

Infineon's PSoC6 is a dual core, ultra-low-power MCU with 2MB flash, 1MB RAM, and up to 150MHz designed for IoT usage according to the datasheet [1], but doesn't have a device that showcases the MCU's capabilities.

1.2 Background

Infineon has been developing microcontrollers for many years [2], with over eight families dedicated to different users. In 2016, Cypress (now under Infineon) had also asked a ECE 445 team to design a line following robot car before. However, Infineon had developed newer modules and hopes to have a robot that could demonstrate the machine learning capabilities of the new PSoC6 MCU.

1.3 Solution

We aim to build a two-wheeled robotic car with object detection capabilities from the radar and will be able to follow a taped line using IR sensors, while recognizing voice commands detected by the microphone on the dev board to control its speed while not compromising the line following objective and also avoid obstacles placed on the line during real time using mostly Infineon products.

1.4 Visual Aid

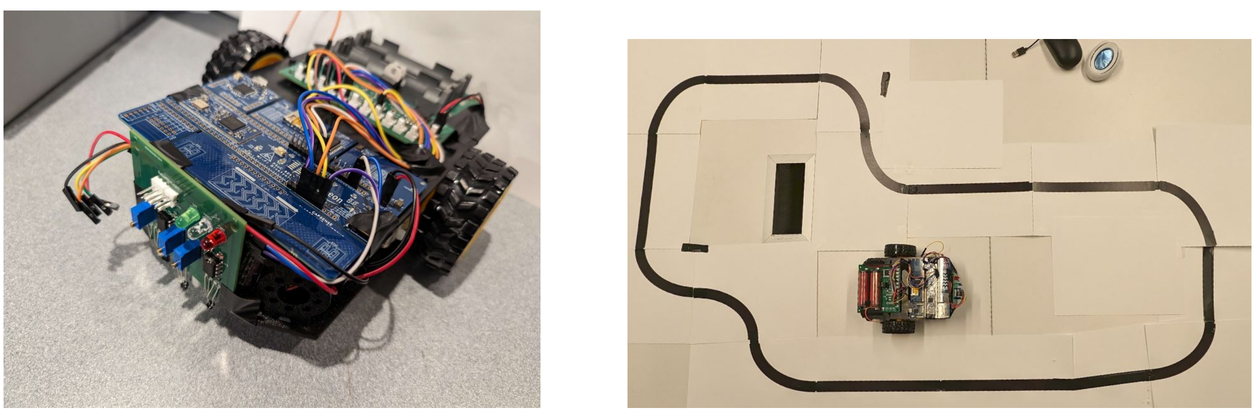

Figure 1: Photos of Final Product and Track Used to Test

1.5 High Level Requirements

1. Voice Command: The robot should be able to register one word commands including "Foward", "Reverse", "Stop", "Speed up/down" and translate the commands into the robot's motion with a 90% success rate.
2. Obstacle Avoidance: The radar should recognize obstacles as small as a 5cm cube within 2-10cm to the radar, and turns the robotic car around with a 90% success rate.
3. Line Following: The robotic car should be able to follow a 5m taped track with a minimum radius of curvature of 25cm at up to 70% speed and less than 7.5cm off track with a 90% success rate
4. Speed Control: The control module should be able to control the motor speed continuously from 0-100% with less than 2 seconds of setting time.
5. Battery Life: The onboard rechargeable battery should be able to 15x30x5 cm car (estimated 500g) at 90% speed for at least 10 minutes.

1.6 Block Diagram

In our final product, we omitted the charging circuit which will be discussed later.

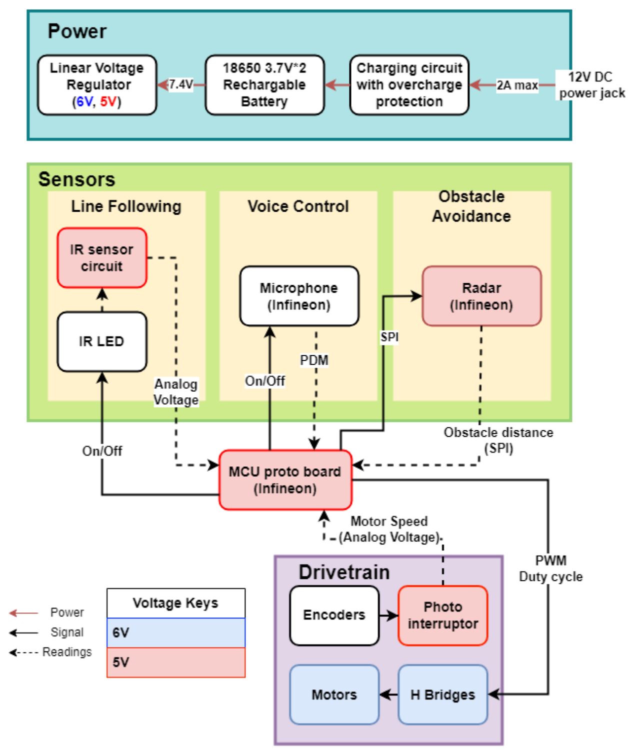

Figure 2: Block Diagram

2 Design

2.1 Design Choices

1. Power: Removal of Recharge Circuit due to safety concerns with cell balancing
2. Microcontroller: Removal of Reverse capability due to inconsistent line following
3. Sensor: Lowered number of IR sensors from five to three since there was no benefit brought by added sensors
4. Drive Train: Removal of closed loop speed feedback control due to low motor speed

2.2 Power Subsystem

Our power comes from two 3.7 V lithium ion rechargeable batteries connected in series to achieve a total voltage output of 7.4 V, which goes through three preset voltage regulators to output the required voltage and current for all the electronic components.

2.2.1 Rechargeable Batteries and Battery Holders

To only use linear step down regulators, we require a battery capable of providing at least 6 V at at least 3 Amps for power to all the different components.

If we were to use a lower voltage battery, we would have to use step-up/boost converters that would add unnecessary complexity to our design. To meet these requirements, we decided to use two 3.7 V lithium ion rechargeable batteries connected in series. In particular, the INR-18650-P28A [3] perfectly suits our needs as it outputs 3.7 V at maximum 35 A with a total capacity of 2800 mAh.

To put our batteries on our PCB close to the components that draw power from it, we require mounts to place them on the PCB as. To suit our needs, we chose the 1043P Battery holder module from Keystone Electronics. It is through hole mountable which is much easier to solder on to our PCB and uses THM Polarized contacts to ensure that the battery is placed in the correct orientation with its polarity matching every time.

The charge time for the battery is extremely fast, being only 1 hour when connected to the separate charger. During our entire time working on our project, we did not have to recharge our battery once which serves to show that the total capacity of 2.8Ah is sufficient for use.

2.2.2 Voltage Regulator

For our different modules, we need to supply the correct voltages to the motors and other parts of the circuit. Our batteries can only supply a fixed 7.4 V at 3 Amps while each DC motor requires a maximum of 6 V at 1.5 Amps while everything else on the circuit requires 5 V at various Amps. To meet these power requirements, linear voltage regulators are needed to convert the fixed input voltage to our varying output voltages.

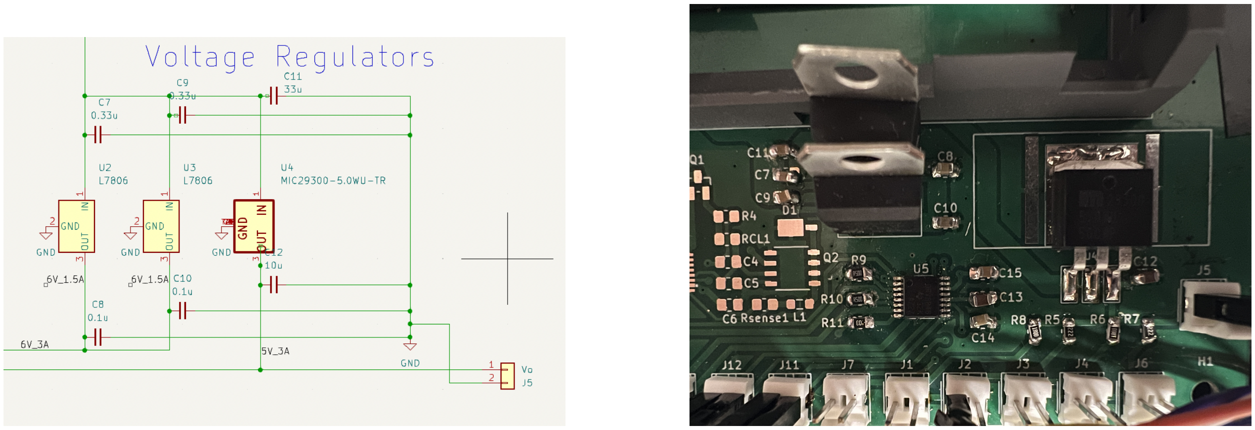

Figure 3: Voltage Regulators

For the motors, we chose two L7806ABV linear voltage regulators connected in parallel as it is capable of converting a varying input voltage to a fixed six V at three Amps output voltage and current as pictured on the left side of Figure 3 which fulfills the power requirements of each DC motor.

For the rest of the circuit, a single MIC29300 linear voltage regulator suffices as it is capable of converting a varying input voltage to a fixed 5 V at 3 Amps output voltage and current as pictured on the right side of Figure 3 which fulfills the power requirements of the rest of the circuit.

2.2.3 Removal of Battery Management IC

When choosing our battery management IC, we went with the LTC4006EGN-6 [4] battery management IC since it seemed to meet our power needs perfectly but we decided to remove it after our TAs brought up safety concerns due to the absence of a cell balancing feature. Without a cell balancing feature to ensure that the charge supplied by the battery management IC is distributed evenly across both cells, we cannot guarantee equal charge rates between the two cells which would lead to cell charge imbalances.

This charge imbalance is a safety concern since if lithium cells are overheated or overcharged, they are prone to accelerated cell degradation. They can catch fire or even explode as a thermal runaway condition can occur if a lithium ion cell voltage exceeds 4.2 V by even a few 100 mV [5]. Due to time constraints and because recharge ability was not a main focus of our project, we decided to remove the battery management component altogether as shown in Figure 4.

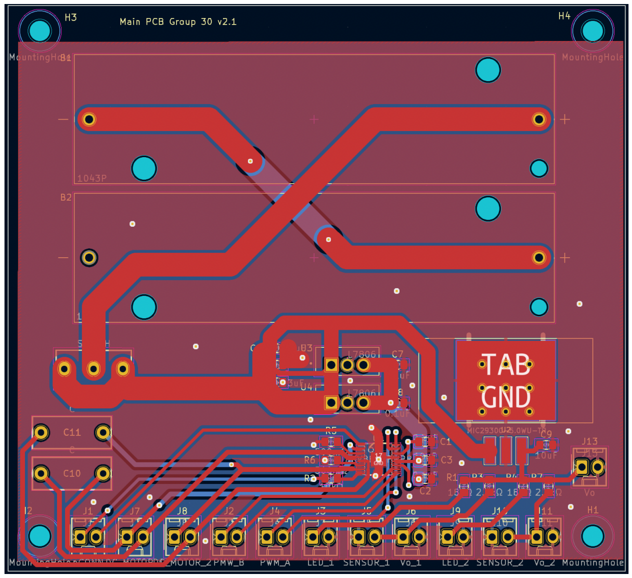

Figure 4: PCB Design of Main Board which Includes the Power and Drive Train Subsystems

2.3 Micro controller Subsystem

The micro controller subsystem would include the PSoC6 MCU and other components on the development board. The dev board shown in figure 5 takes a 5V USB input, which we will be supplying through the batteries.

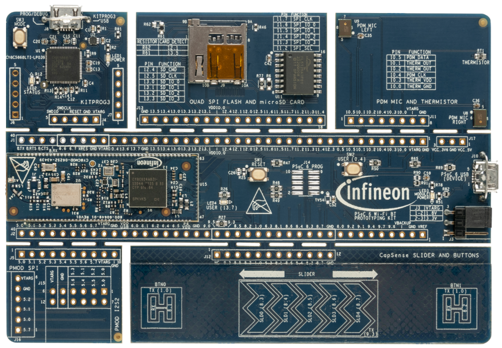

Figure 5: Photo of Our Dev Board

2.3.1 Modus Toolbox

One of the requirements laid out by the pitcher was to use Modus toolbox, the environment Infineon developed to program the MCU in C. The MCU would take in analog voltage from IR sensors, SPI from radar, and PDM from microphones. It should then determine how the motors' behaviors should be altered. Infineon provides two options to implement our application, as shown in figure 6, either using Hardware Abstraction

Layer, which we chose because it's easier and more versatile, or the Peripheral Driver Library. We started from the example programs, defined GPIO pin names in the device configurator, and initialized HAL blocks which reserves the used resources.

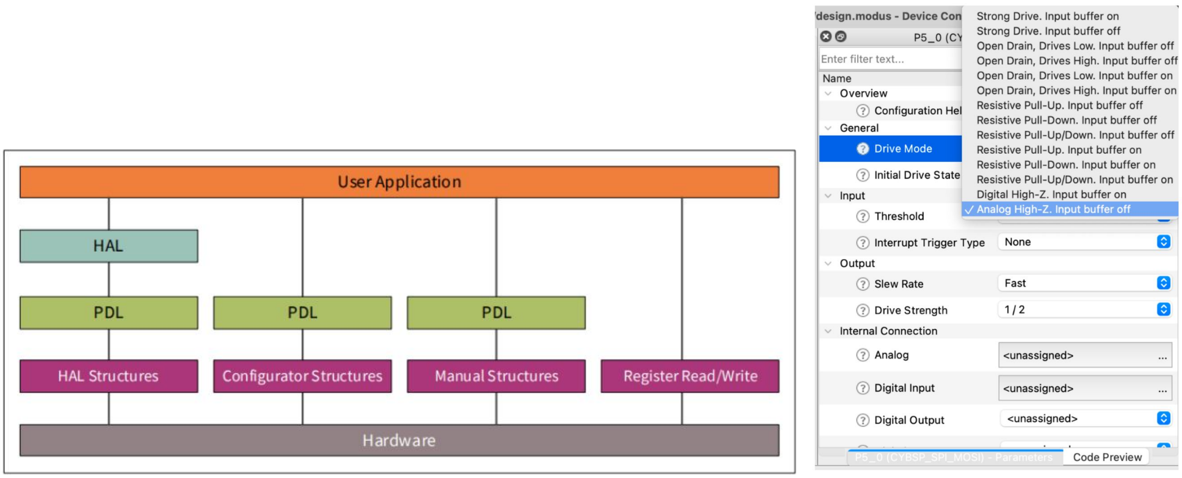

Figure 6: Left: Possible Methods of Implementing Applications Right: Available Pin Modes Shown in Device Configurator

Another detail was the modes for each GPIO pin, as shown on the right in figure 6. For each different signal, we used different modes. For digital signals like the ones from radar and the microphone, we used digital high z. For the analog signals, instead of using the analog mode and implementing ADC, we used pull down mode to convert the analog signal to digital.

The programs for each function are modularized, and each one works in their own thread to prevent one function blocking the all the computing resources.

2.4 Sensor Subsystem

2.4.1 Overview

The sensor subsystem includes IR sensors for line following, microphones for voice command, and radar sensors for obstacle avoidance. Output measurements of sensors provide MCU with its current configuration with respect to the environment, which is necessary for the MCU to control the of the robot car.

2.4.2 IR System

IR sensors are used for line following. Each IR sensor unit is composed of an IR LED as transmitter, a photo diode as receiver, and an op amp used as a comparator to produce the analog voltage signal as pictured below in diagram 7.

Output voltage level changes according to the intensity of light detected by photo diode with this equation:

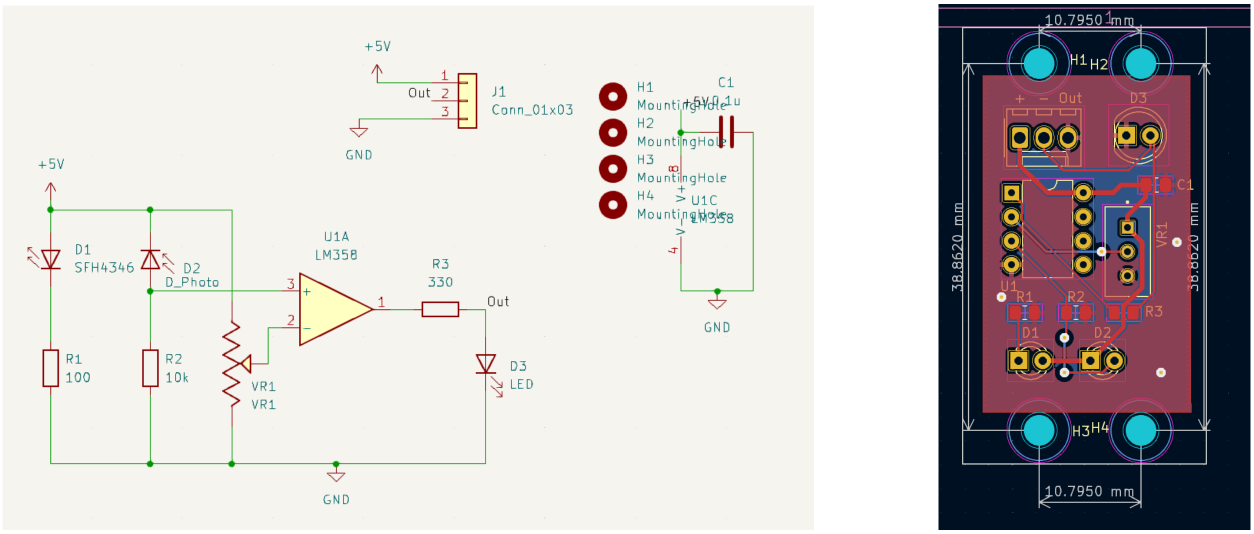

Figure 7: Schematic and PCB design of IR Sensor Circuit

$$V_{out} = A_{OL}(V_+ - V_-) \tag{1}$$

This helps us identify the surface color right below the sensor, since white surface will reflect much more light than black surface. In the end, we updated R2 with a larger resistance value (91k) to give a bigger tuning range. The current through the photodiode when it is on is fixed, so a larger R2 will give a larger V_+, meaning that we can have a larger range of V_- for the op-amp to be on. That ultimately increased the maximum $V_+ - V_-$ from around a few 10 mV to few 100 mV. On the other hand, too big of this resistance would result in a slower response time. We then connect the circuit output to the a GPIO pin on the MCU set at pull down mode to determine the relative location of the car to the track.

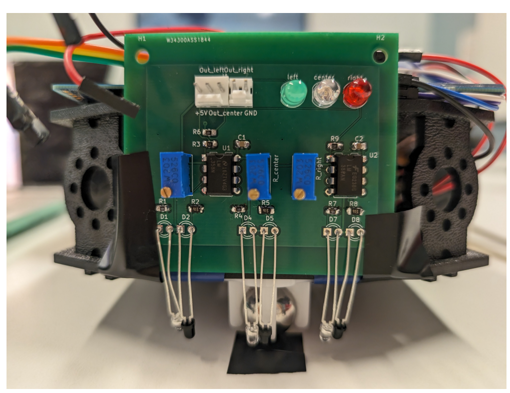

Figure 8: Front view of assembled IR sensor

Our plan was to place five IR sensor units at the front end of our chassis (labeled one to five from left to right), with unit three at the center and two more on each side 1cm away from each other. Each IR unit will be placed 0.5cm above the ground. Our tape will be 1.9cm in width. When the car is centered on the line, only unit three detects black surface. Side units detects black when car is off center and robot will turn accordingly. The outer two units are for large turns and inner two units are for following straight lines. In the end, however, since the car was not running at a high speed, it turned out only using one level of turn was enough to follow the line, so we only implemented three IR sensor pairs.

2.4.3 Voice Control System

Hardware The two PDM microphones on the top right of Infineon Development board are used to receive voice signal (Figure 5). It outputs a PDM digital signal to the MCU for voice recognition. The microphones require a supply voltage between 1.62V - 3.60V, which will be supplied internally by the development board.

Software Our algorithm for voice command detection is based on Cyberon Dspotter Machine Learning Solution. This on-chip machine learning solution is able to perform a two-stage voice command detection, with first level as trigger and second level as action command. We create customized Dspotter model with our required voice commands and fine-tuned the model to best suit our use case.
This model breaks each command into phoneme and look for the correct phoneme pattern during speech recognition. Each voice command is assigned a Command Id that could be used in our main program. Alternative pronunciation for "reverse" command is included to accommodate different accents. Model could be fine-tuned with the major parameters, including Confidence Reward, SG Difference Reward, and Energy Threshold. Confidence Reward determines how easily a command is to be recognized, SG Difference Reward determines how much the command is different from silence/garbage, and Energy Threshold determines the minimum energy level required to be recognize as a command. We adjusted the value of each command's Confidence reward and SG Difference Reward to achieve the maximum sensitivity without high false positive rate and adjusted the overall Energy Threshold to reduce the affect of noise. The interface and parameter values can be seen in Figure 9.

To integrate the model with our program, we included two libraries: cyberon_asr and CybModelInfo. As for APIs, we used cyberon_asr_init() to initialize to model and cyberon_asr_process() to process each voice command. Duration of silence is used to determine ending of a voice command. Once a voice command is completed, we use a variable pdm_pcm_flag to flag the new entry and trigger cyberon_asr_process().

2.4.4 Radar System

To perform obstacle avoidance, the most basic data we need will be the obstacle distance. The Intermediate Frequency signal will have information such as the phase and frequency difference between the sent and received signals which we can derive the distance from.

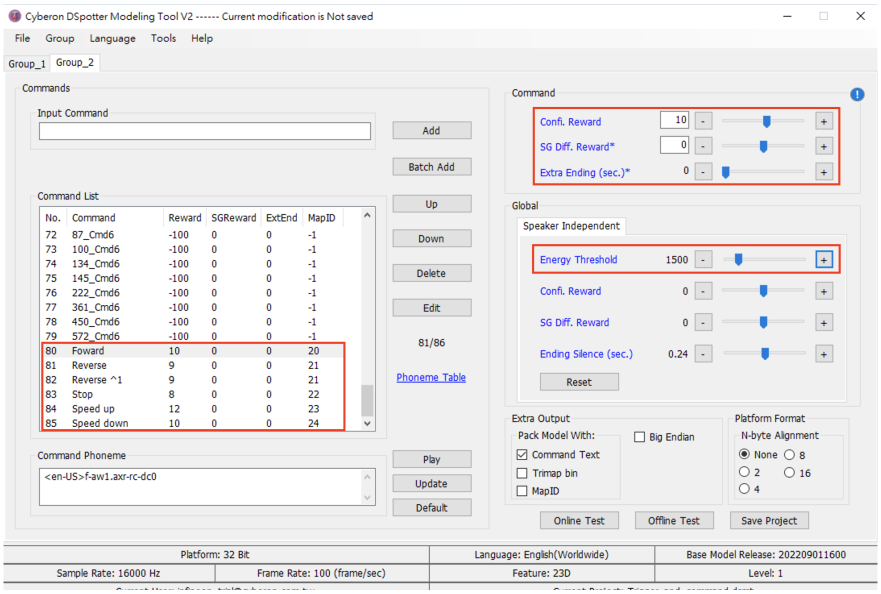

Figure 9: Dspotter model customization interface

By measuring the time delay τ between the transmitted and received signals, we can estimate the distance between the object and the sensor: $D = \frac{\tau}{2c}$. On the other hand, knowing the phase difference between two receiver antennas could tell us the angle . We would also be able to calculate the Doppler speed by the difference in frequency of the sent and received signal. Also, the radar was desgine to be mounted on a stationary object and we found the vibrations while the car is moving made the sensor misread sometimes.

From our objective, it made sense to have all the modules on a single MCU. However, we had trouble establishing a connection from the dev board to either of our radar boards, one demo board designed to work with a PC for processing and one wing board with its own PSoC6 MCU. We found we would need 10 total connections, including both I2C and SPI, and it was challenging to make such a number of stable connections on the dev board in addition to other modules' connections, especially if we would need both I2C and SPI since not any GPIO pin has the resources required for those connections. We found a workaround, which is to use the wing board with its own MCU and output a single signal indicating an obstacle detected in range. We also found that the orientation of the obstacle and the material changes the strength of the reflected signal even at the same distance. Therefore in the end we used our hands instead of a stationary object as the radar was designed to detect humans. We also mounted the radar a bit higher than the car to avoid the radar picking up signal reflected off the car itself.

Program wise, we used a example program - human presence detection, provided by Infineon. We were able to find the function's declaration but not its definition, so the most we could do was to tune the parameters they provided, as labeled in figure 10. Macro threshold is essentially the threshold of the reflected signal strength, so it is related to the trigger distance. We set this to the highest as the program allowed to ensure the

```c
static const xensiv_radar_presence_config_t default_config =
{
    .bandwidth                         = 460E6,
    .num_samples_per_chirp             = XENSIV_BGT60TRXX_CONF_NUM_SAMPLES_PER_CHIRP,
    .micro_fft_decimation_enabled      = false,
    .micro_fft_size                    = 128,
    .macro_threshold                   = 2.0f,
    .micro_threshold                   = 12.5f,
    .min_range_bin                     = 1,
    .max_range_bin                     = 1,
    .macro_compare_interval_ms         = 250,
    .macro_movement_validity_ms        = 1000,
    .micro_movement_validity_ms        = 4000,
    .macro_movement_confirmations      = 0,
    .macro_trigger_range               = 1,
    .mode                              = XENSIV_RADAR_PRESENCE_MODE_MACRO_ONLY,
    .macro_fft_bandpass_filter_enabled = true,
    .micro_movement_compare_idx       = 5
};
```

Figure 10: Provided Configuration Struct in Example Program

radar wasn't picking up reflections from environment. The max range bin was set to the same as min range bin so the radar would only have two output states: on/off. The mode was changed to macro only instead of micro if macro since we didn't need micro movements and they slow down the reset time of the radar. The fft filter is enabled to filter out the vibrations from the car, as the vibrations have a different frequency than the actual signal.

2.5 Drive Train Subsystem

The two motors are bidirectional due to the addition of h bridges to allow current to flow in both directions through the motor. A photo interrupter will monitor motor speeds and provide feedback for use with speed modulation. Each individual motor should have variable continuous speed following capabilities from 0-100% based on the control signals from the MCU. It should also be able to rotate in both directions to enable turning.

2.5.1 H-bridge

The H-bridge is the heart of the Drive Train subsystem and is responsible for the behavior of the two DC motors. The H-bridge can configure the motors to achieve four different modes, to coast, to move forward, to reverse, and to brake. The H-bridge can also achieve speed control since it is provided with a Pulse Width Modulated signal from the MCU. The duty cycle of the PWM can be used to achieve varying analogous speed control.

To meet our DC motor power requirements, we chose the DRV8833PWP Dual H-Bridge Motor Driver chip. The DRV8833PWP is ideal for our requirements as it can handle a wide power supply voltage from 2.7 to 10.8V while outputting an average current of 1.5Amps. Since the chip has two H-bridges inside, a single chip can be used to drive both of our DC motors as seen in Figure 11.

2.5.2 DC Motors

The biggest criteria for choosing the motors are the maximum rpm and maximum torque, as that would determine the car's maximum speed and maximum acceleration. To reduce

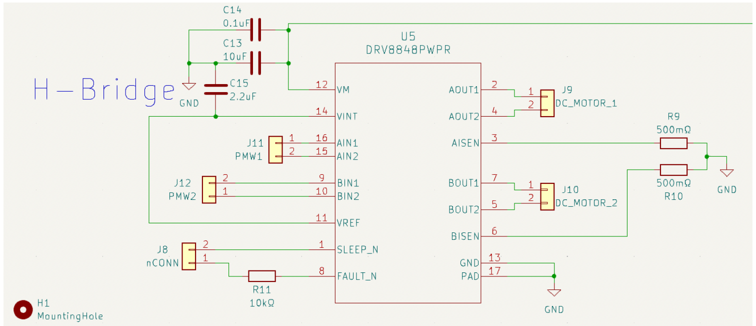

Figure 11: H-bridge Schematic

costs, we decided to stick with plastic DC TT motors. Out of the plastic TT motors, the highest rpm we could find was 250 without load, and that would be $V_{max} = \frac{250}{60} * \pi * D_{wheels} = 0.85m/s$. The relationship between the maximum torque and the maximum acceleration when the car is moving is as follows:

$$F_{max} = \frac{\tau_{max}}{D_{wheel}} = m_{car} * a_{max} + f_{kinetc} \tag{2}$$

The final robotic car weighs around 500g, thus, with estimating the kinetic friction coefficient at 0.15 and wanting an acceleration at least $1m/s^2$, the torque would have to be at least 80mNm. The Adafruit 3777 motors have 78 nNm, which is slightly under our expected value, but higher torque models would be a lot pricier and with smaller max rpm.

2.5.3 Photo Interrupter

To ensure that the motors are operating at the desired Revolutions per minute(RPM), speed control feedback is required. Since the DC motors are dual shafted, a Photo Interrupter is mounted to the shaft that the wheel is not on to monitor the RPM. The Photo interrupter can be divided into three parts, the Encoder, the IR LED, and the IR sensor.

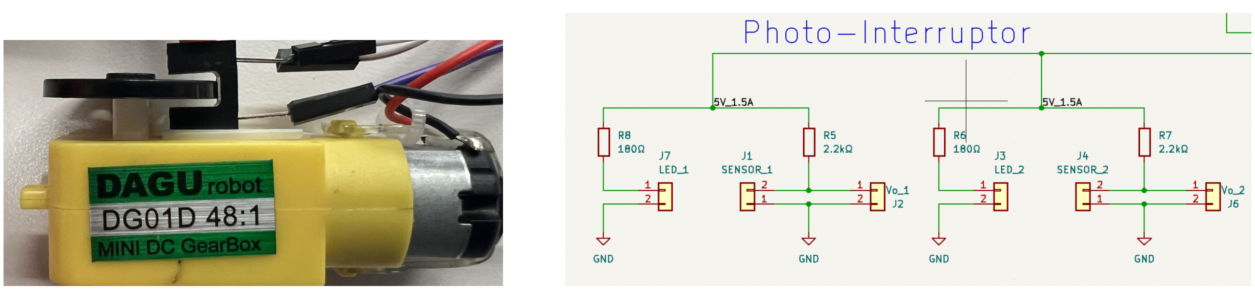

Figure 12: Photo Interrupter mounted on motor

The Encoder is a plastic disc that has a number of notches, that allows IR light to pass through when it is rotating. The Photo Detectors with encoders in-between their gaps are mounted to the motor as shown in Figure 12 to calculate the rotation ticks. Its working principle is quite simple as it is essentially made up of an LED and a photo detector. Whenever the light from the LED passes through the slits of the encoder, the photo detector gets excited and its analog voltage changes, the time difference between two consecutive lows or highs can be measure by the MCU which calculates the actual RPM of the motor according to formula 3 where $clock\_freq$ is the internal clock frequency used bu the MCU to measure ticks.

$$\text{RPM} = \frac{60.0}{\text{tick} \times 40.0} \times \text{clock\_freq} \tag{3}$$

2.5.4 Speed Feedback Control

Our hypothesis was that there will be a lot of friction and load on the motors, hence we would need to ensure that the actual speed of the motors closely follows our reference speed for good line following performance hence, a feedback control loop would be needed.

However, after actual implementation and testing, we found that even in the absence of precise speed control, the robot had no issues following the line leaving our speed feedback superfluous, hence we did not use the RPM measure from the Photo Interrupter in any comparator for speed feedback control.

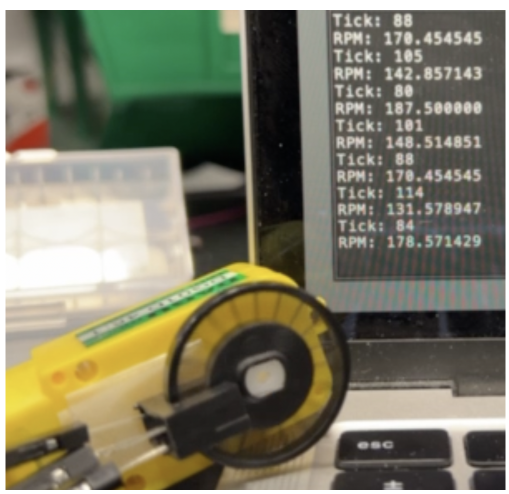

Figure 13: RPM measure from Photo Interrupter

3 Verification

3.1 Power

Requirements	Verification
• The two different types of voltage regulators should be able to provide the correct corresponding output voltage and current when a load of 3A is applied across each one.	• Connect the batteries to the voltage regulators and apply two separate 3 Amp loads across each voltage regulator. • Monitor the voltage regulators' output voltage with an Oscilloscope and ensure that the voltage stays within the corresponding voltages of 5V +/- 0.2 and 6V +/- 0.2. **Result** Testing without load for the 5V voltage regulator resulted in no voltage drop across it, adding a load in the from of the MCU fixed this issue resulting in a load of 5V +/- 0.1. The 6V voltage regulator created a voltage drop without a load applied across it resulting in a load of 6V +/- 0.1.

3.2 MCU

Requirements	Verification
• Each program should be modularized and not block up each other.	• Try to have multiple functions runnning at the same time and see if any program gets halted. **Result** All programs except radar was able to work on the same chip together, but we simply took 1 output signal from the radar.

3.3 Sensor

Requirements	Verification
• Each IR Sensor unit must output voltage above the same threshold voltage of 2 V +/- 0.1 on white surface and output voltage below threshold voltage of 2 V +/- 0.1 on black surface.	• Place the IR unit perpendicular and 0.5cm away from the white surface, and make sure the reading on voltmeter is below the threshold voltage. Vice versa for black. • Repeat step for all five units. **Result** The output was fixed at 1.7V by the LED, and the MCU was able to register that as a logic high.
• Microphone able to detect voice command 2m away.	• Program the development board with PSoC™ 6 MCU: PDM-to-PCM example in ModusToolbox. • Give the five voice commands 2 meters away from the microphone. **Result** For two tests of each command, the LED turned on each a voice command is given, and turns off without a voice command.
• Voice command recognition should be able to recognize the five voice commands even when motor noise is present.	• Run our voice command recognition program on the Infineon development board. • Place a running motors at each side of the development board and give voice commands. **Result** Voice commands were identified with 80% accuracy in 20 entries of random command.

Requirements	Verification
• Radar sensor has to measure the distance of obstacle within 1 meter in the range of ±40°vertically and horizontally within 5% error rate.	• Run the program for radar module on the development board. • Place obstacle in front of the radar and measure the distance from the obstacle to the radar. **Results** The results varied with the material and orientation of the obstacle, so we tried to put a hand perpendicular. The distance had an average around 33cm, but with a +/- 5cm variation as seen in appendix B
• Radar sensor can perform accurate measurement while the sensor is moving.	• Move the radar toward the obstacle and away from the obstacle while keep track of the measurement reading. • Make sure the measured distance decreases as radar move closer to the obstacle at fixed velocity and check if the readings are roughly linear. **Result** The radar readings weren't too consistent even when stationary, but we used fft filter and check radar didn't get triggered simply by vibrations.

3.4 Drivetrain

Requirements	Verification
• The speed sensors' response time should be continuously outputting and updating the tick signal at least fast enough to capture 250 RPM.	• At steady motor speed, the speed sensors should output the ticks corresponding to the correct speed which is RPM*40 ticks per rotation (+/- 5 RPM) **Result** Changing the speed of the motors by controlling the PWM signal from the MCU resulted in the expected values in a tolerance of +/- 10. Due to movement from the motors, some ticks would be registering twice with RPMs exceeding 7500. Filtering out ticks below 10 as false positives fixed this issue.
• The DC motor should go up to 180 RPM with load at max power to reach a speed of 0.6m/s	• Power the motor at 6V and 100% duty cycle. Check the max current drawn on an oscilloscopes. • With the chassis assembled, let the wheels run at full speed for 5 seconds and measure the distance travelled. **Result** Providing maximum current to the motors resulted in a distance of 2.87m +/- 0.1, suggesting a lower speed of around 0.574m/s. This may be due to a number of reasons such as friction, variable power draw from motors, or not moving in a exactly straight line.

4 Cost and Schedule

4.1 Cost Analysis

4.1.1 Labor

Each group member will be expected to work around 2.5 hours per day over a 50 day period on the project with a salary of $42/hr.

$$\$42/hr * 2.5 * 50 = \$4,200$$

For 3 group members, the total labor cost will come up to $12,600.

4.1.2 Parts

Description	Manufacturer	Part Number	Quantity	Extended Price
PSoC 62S2 Wi-Fi Bluetooth Prototyping Kit	Infineon	CY8CPROTO-062S2-43439	1	$56.25
P28A 18650 2800mAh 35A Battery	Molicel	INR-18650-P28A	2	$9.98
BATT HOLDER 18650 1 CELL PC PIN	Keystone Electronics	36-1043P-ND	2	$6.52
IC BATT CHG LI-ION 2CELL 18QFN	Monolithic Power Systems Inc.	1589-MP2672GD-0000-ZCT-ND	1	$2.71
CAP CER 1UF 6.3V X7R 0402	Murata Electronics	490-13339-1-ND	1	$0.10
CAP CER 10UF 16V X5R 0805	Murata Electronics	490-6473-1-ND	2	$0.32
CAP CER 22UF 4V X5R 0603	Murata Electronics	490-5526-1-ND	4	$0.64
CAP CER 0.1UF 6.3V X5R 0201	Murata Electronics	490-3167-1-ND	7	$0.70
RES SMD 30K OHM 0.1% 1/16W 0402	Susumu	RG10P30KBCT-ND	1	$0.48
MP6K000 0.05%	VPG Foil Resistors	MP6K000-ND	1	$0.44
FIXED IND 1.5UH 2.6A 0.06OHM SMD	Murata Electronics	490-10648-1-ND	1	$0.27
IC REG LINEAR 5V 3A TO263-3	Microchip Technology	MIC29300-5.0WU-CT-ND	1	$4.68
IC REG LINEAR 6V 1.5A TO220AB	STMicroelectronics	497-1444-5-ND	2	$1.48
IC MOTOR DRIVER PAR 16HTSSOP	Texas Instruments	296-29394-5-ND	1	$2.96
CAP CER 2.2UF 6.3V X6S 0402	Murata Electronics	490-10012-1-ND	6	$0.60
CAP CER 10000PF 10V X7R 0201	Murata Electronics	490-3194-1-ND	1	$0.10
GEARMOTOR 200 RPM 3-6V DC	Adafruit Industries LLC	1528-2589-ND	2	$5.90
TT MOTOR ENCODER (PACK OF 2)	Adafruit Industries LLC	1528-2615-ND	1	$0.95
SENSOR OPT SLOT PHOTOTRANS MOD	Lite-On Inc.	160-2210-ND	2	$2.36
INFRARED LED	N/A	ILED-8	10	$9.30
Photo Resistor	N/A	PID 161	10	$11.70
LM358N LOW POWER OPERATIONAL AMPLIFIER	N/A	LM358N	7	$6.51
10K SINGLE TURN 1/2WATT TRIMPOT	N/A	3352W-1-103	5	$7.50

Figure 14: List of parts and cost

The total cost for the parts is $132.45. 5% shipping cost adds $6.63 and 10% sales tax adds $13.25 for a total parts cost of $152.33.

4.1.3 Grand Total

Adding the cost of labor and the cost of the various parts, we have a grand total cost of $12752.33.

4.2 Schedule

Week	Task	Person
10/1 - 10/7	Order parts for prototype	Everyone
	Power system prototyping	Wei-Jui
	IR sensor prototyping	Kai-Chieh
	Drive train prototyping	Saharsh
10/8 - 10/14	1st PCB design and order	Everyone
	Radar module testing	Kai-Chieh
	Cyberon ML model testing	Wei-Jui
	Start motor control	Saharsh
10/15 - 10/21	Car assembly and module integration	Everyone
	Finalize motor control	Saharsh
	Voice control implementation	Wei-Jui
	Line following implementation	Kai-Chieh
10/22 - 10/28	Obstacle avoidance implementation	Kai-Chieh
	2nd PCB design and order	Everyone
	Chassis design and 3D print	Wei-Jui
	Advance feature research	Saharsh
10/29 - 11/4	Advance feature design and implementation	Everyone
11/5 - 11/11	Advance feature implementation and testing	Everyone
11/12 - 11/18	Minor bugs fixing and fine tuning	Everyone
11/26 - 12/2	Demo	Everyone

Table 1: Planned schedule

5 Conclusion

5.1 Accomplishments

We were able to meet most of our high level requirements. The car was able to carry out all the specified commands with around 80% success rate. The radar was able to register people's hands around 15 cm away and turn the car around 95% of the time. The line following was also successful 90% of the time, getting stuck only when the turn is too fast. We had five levels of speed controllable through voice with a min of 30% duty cycle and max of 70% duty cycle with just an open loop control. The rechargeable batteries were able to last at least a 3 hours. We only didn't have the radar running on the same MCU.

5.2 Uncertainties

We faced four major challenges throughout the project. First, safety concern of the charging circuit. Our charging IC was not capable of cell balancing when charging two batteries in series. Therefore, we decided to charge our batteries with commercial charger. Second, low accuracy of voice command recognition in the presence of motor noise. This is due to motor noise and human voice share the same frequency range of 100-300Hz. Because of the space limit of our chassis, we could only place the microphones near the motors where noise is unavoidable. Improvement would only be possible if we have external microphones mounted elsewhere. Third, line following in reverse mode. We only have IR sensors at the front of the chassis, so IR sensors will not be able to provide information in time for our robot to follow the line backward. The problem could be solved by adding another copy of IR sensors at the back of the chassis. However, we implemented U-turn when reverse is needed to minimize large modifications to hardware. Fourth, our radar was not able to be connected to the dev board. We found out we needed 10 connections and both SPI and I2C set up, which would be difficult to fit onto our dev board as many pins are reserved. So we simply used the radar board and its MCU as a separate module and outputted a signal when the radar is triggered.

5.3 Future Work

The biggest challenge for future work would be integrating the radar to the Dev board MCU, as doing so would require more stable connections and modifications to the programs. To make the product look more sophisticated, custom designed and 3D printed chassis with mounting for all the electronics would be helpful. Also, we could try to improve the motor speed and make line following look even cooler. We can either add more motors or simply swap out the plastic gearbox with metal ones that have higher rpm. Another cool feature to implement could be adding another radar sensor to perform cruise control around obstacles.

5.4 Ethics

As per the IEEE Code of Ethics [6], our design must be developed sustainably and comply with ethical designs while ensuring the privacy of others and to disclose any potential factors that might risk the safety of the public and environment. Our project fits the IEEE Code of Ethics 1.2 as we are creating a demo-able project to teach the public about the electronics and machine learning, while promoting interest in STEM among children. Any criticisms of our technical design, our claims stated based on our best estimates of the available data or our proper credit of others' contribution will be sought, accepted and acted upon to ensure lawful conduct in our professional activities. Any actions that undermine the ethical and moral integrity of the team will be circumvented. All members will work in an environment of acceptance and non-discrimination.

5.5 Safety

The most possible potential safety hazards we identify are the physical safety of people and surroundings if the robotic car goes out of control and safety related to power and battery. To mitigate the potential hazards, we should test the robotic car in an open field or an enclosed area. A stop button easily accessible should also be installed on the robotic car. For power related hazards, our commercial batteries and charger have built in protection. When charging the batteries, at least one team member would be next to it. The batteries would be stored in fire proof bags when not used and ideally stored away not fully charged.

References

[1] Infineon. “”CY8CPROTO-062S2-43439 PSoCTM 62S2 Wi-Fi Bluetooth® prototyping kit guide”.” (2023), [Online]. Available: https://www.infineon.com/dgdl/Infineon-CY8CPROTO-062S2-43439_PSoC_62S2_Wi-Fi_Bluetooth_prototyping_kit_guide-UserManual-v02_00-EN.pdf?fileId=8ac78c8c85ecb34701865c87426d3fc3 (visited on 09/28/2023).

[2] Infineon. “”The evolution of TriCore”.” (2023), [Online]. Available: https://www.infineon.com/cms/en/product/promopages/historyoftricore/#:~:text=Everything%20started%20in%201999%2C%20when,instant%20hit%20towards%20Powertrain%20applications. (visited on 09/28/2023).

[3] Molicel. “”INR-18650-P28A Datasheet”.” (), [Online]. Available: https://cdn.shopify.com/s/files/1/0481/9678/0183/files/INR18650P28A-V1-80093_900a9c5f-df75-494e-8d58-44e375041d8b.pdf?v=1663385054 (visited on 09/28/2023).

[4] L. Technology. “”LTC4006 Datasheet”.” (2021), [Online]. Available: https://www.analog.com/media/en/technical-documentation/data-sheets/4006fa.pdf (visited on 09/28/2023).

[5] B. Power. “”Why Proper Cell Balancing is Necessary in Battery Packs”.” (), [Online]. Available: https://www.batterypoweronline.com/blogs/why-proper-cell-balancing-is-necessary-in-battery-packs/.

[6] IEEE. “”IEEE Code of Ethics”.” (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

Appendix A Flow chart

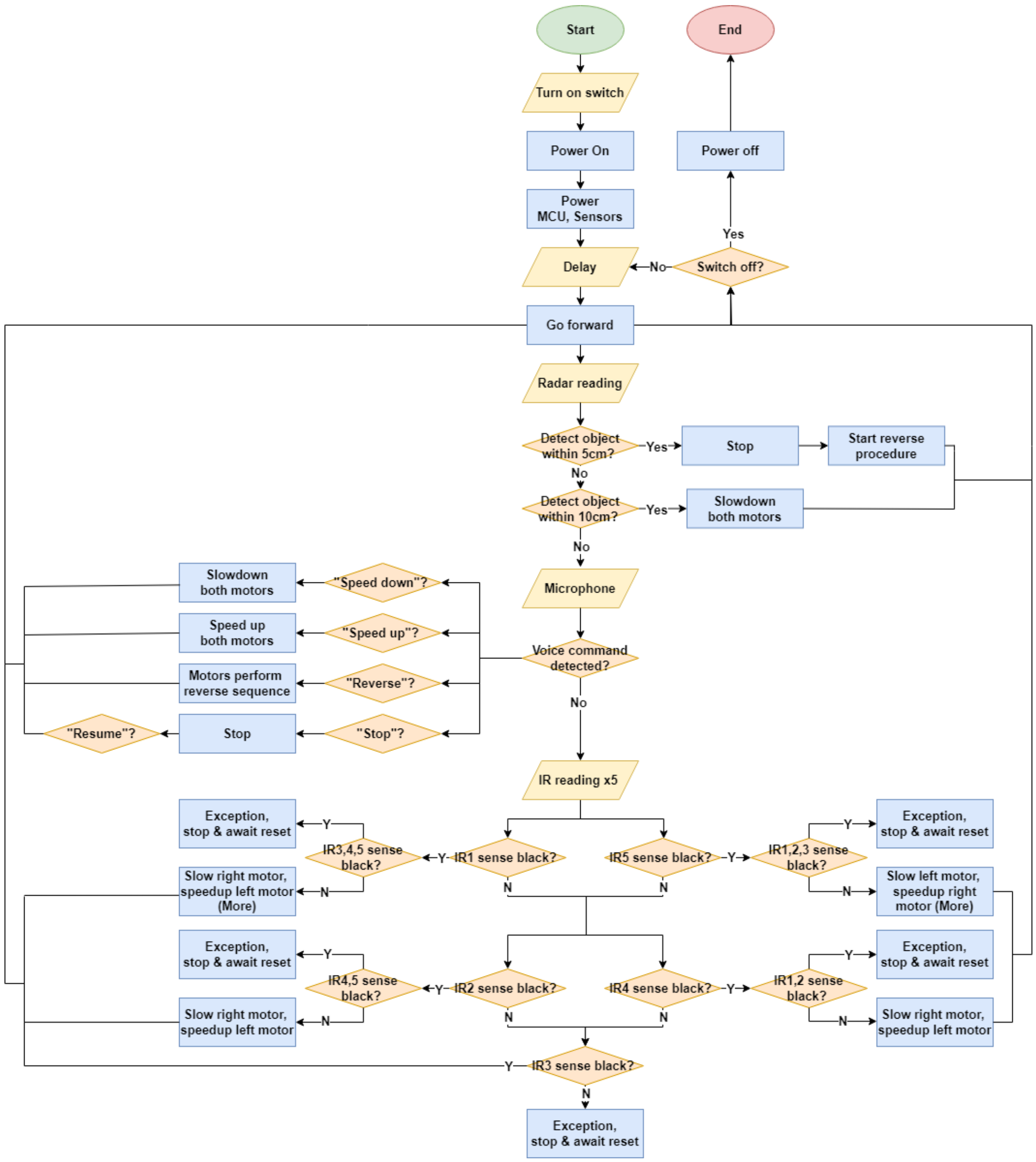

Figure 15: Flow Chart

Appendix B Detailed Verification Results

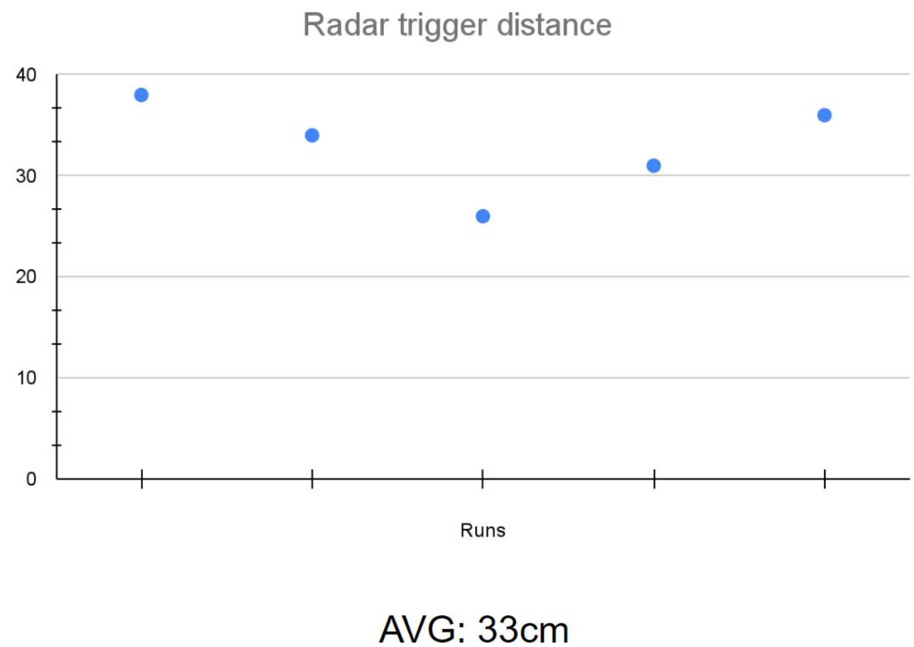

Figure 16: Results of Radar Trigger Distance