Cypress Robot Kit

Final Report

Team Members:
Alvin Wu
Byung Joo Park
Todd Nguyen

Teaching Assistant:
Katherine O'Kane

ECE 445
Group #5

December 7, 2016

Abstract

The Programmable System-on-Chip (PSoC) made by Cypress is an alternative to hobbyist microcontrollers, such as the Arduino or Raspberry Pi, that provides similar capabilities, but is perhaps even more powerful with its programmable hardware blocks. In this report, the development of a robot kit whose purpose is to display the capabilities of the PSoC is outlined in detail. The main component that is designed is a shield where the PSoC, sensors, and motors can be easily plugged in achieving a plug-and-play product. The robot can be controlled manually through Bluetooth Low Energy (BLE) as well as autonomously through its obstacle avoidance and line following modes. This report provides details for both the hardware and software design of the system and verification of its main modules.

Contents

1 Introduction

1.1 Purpose

In the past decade, the availability and innovation of microcontrollers and 3D printing have created a ”Maker Movement” [1]. Most notably, both the Arduino and Raspberry Pi have seen great popularity. This movement has led to the emergence of a new market, namely makers and hobbyists. The goal of this robot kit is to highlight the PSoC as an alternative to controllers such as those mentioned above with the purpose of interesting the maker community in the PSoC and its potential. In addition, many off-the-shelf components are utilized in order to achieve a ”kit-able” and inexpensive product for hobbyists. The hardware and software design for this kit will also be open-sourced allowing makers to not only explore the capabilities of the PSoC, but also choose to add new features or build the product themselves from scratch as a learning experience.

1.2 Objectives

1.2.1 Goals and Benefits

- “Kit-able” design with easy construction
- Inexpensive to make it more accessible to hobbyists (Under $150)
- Plug-and-play experience to demonstrate PSoC capabilities

1.2.2 Functions and Features

- Mountable shield that provides a power system, motor control, and sensors for PSoC
- Remote control using Bluetooth Low Energy (BLE) and phone application
- Ability to switch between manual control and autonomous modes on the phone application
- Autonomous obstacle avoidance using sonar sensor
- Autonomous line following with IR emitting sensors

2 Hardware Design

2.1 Block Diagram

Our design consists of five main modules as shown in Figure 1: power system, sensors, control, motor control, and wireless control. The power system provides 6V to the motors and 5V to the rest of the system. Sensors include an ultrasonic sensor and two IR sensors. The PSoC is in the center taking both sensor and wireless inputs and outputting to the motors. Motor control consists of two H-bridges controlling two brushed DC motors. Finally, wireless control includes the BLE module (built into the PSoC) as well as the phone application and its communication protocol.

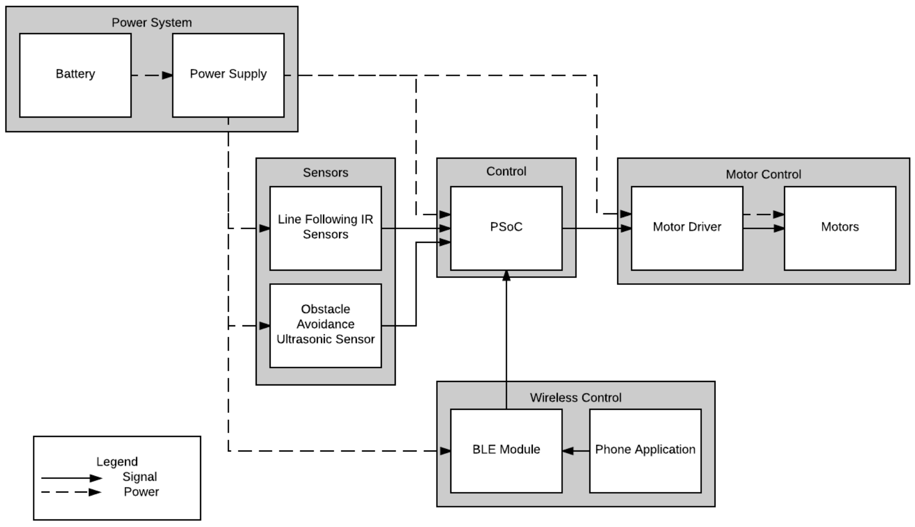


Figure 1: Block diagram showing how the different modules are connected.

2.2 Power System

2.2.1 Power Supply

This module consists of three LT1376 step-down switching regulators (rated at 1.5A maximum current output each) which steps down the input voltage and outputs the desired voltages (5V and 6V) to other parts of the system. We have selected these regulators in order to meet the specification for the stall current of the motors (800mA). Three separate voltage regulators are used to meet the maximum current-draw for our selected components. One regulator will take in 9.6V DC input from the NiMH battery and supply 5V to the PSoC, BLE Module, and sensors as shown in the schematic in Figure 2. The other two voltage regulators will also take in 9.6V DC but supply 6V DC to the H-bridge and motors as shown in Figure 3.

The module also contains an under voltage lockout (UVLO) prevention feature that will shut down drawing power from the battery if the battery voltage drops below 8V.

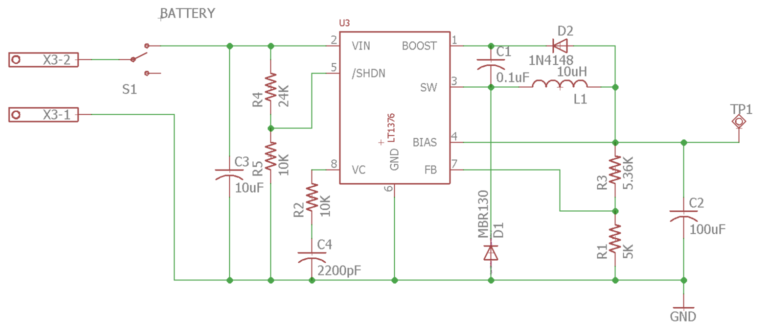


Figure 2: Circuit schematic for 5V power supply [2].

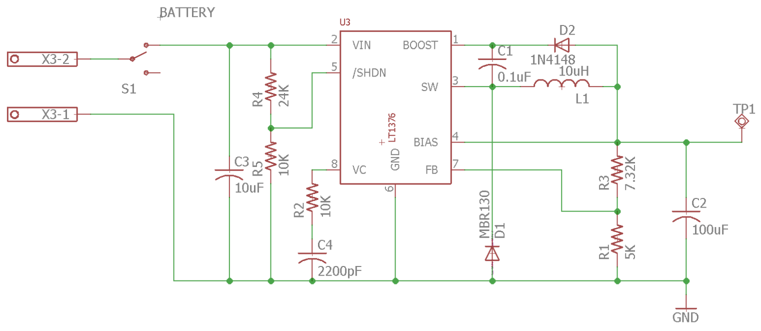


Figure 3: Circuit schematic for 6V power supply [2].

In order to regulate to 5V and 6V DC, the appropriate resistor values that make up the voltage divider from Figure 2 and 3 must be selected. The following voltage divider equation is taken from the LT1376 datasheet where R1 is the upper resistor and R2 is the bottom resistor [2]:

$$R1 = \frac{R2(V_{out} - 2.42)}{2.42} \tag{1}$$

The value of $5k\Omega$ is suggested for R2 in the datasheet. Using this value of R2 we solve for R1 for $V_{out} = 5V$ and $V_{out} = 6V$ respectively:

$$\begin{aligned} R1_{5V} &= \frac{5k\Omega(5V - 2.42)}{2.42} \\ &= 5.33k\Omega \end{aligned} \tag{2}$$

$$\begin{aligned} R1_{6V} &= \frac{5k\Omega(6V - 2.42)}{2.42} \\ &= 7.397k\Omega \end{aligned} \tag{3}$$

Therefore, we picked resistor values of $5.36k\Omega$ and $7.32k\Omega$ for 5V and 6V respectively.

2.2.2 Battery Selection

The robot and its modules are powered by a 9.6V NiMH battery from Tenergy. We have a target system lifespan of one hour of average use. Average use includes driving the robot around without stalling, and collecting data from the various sensors we have attached. Average and maximum power consumption of each module is listed below. Both the average and maximum values are taken from each part's datasheet. If an average current is not listed, we just make both average and maximum current the same.

Module	Average	Maximum
PSoC4	14mA	14mA
Motors (per motor)	300mA	800mA
IR Sensor (per sensor)	25mA	25mA
Ultrasonic Sensor	15mA	15mA
Switching Regulator (per chip)	2.5mA	6mA
H-bridge (per chip)	0.85mA	2.5mA

Our results are divided by 0.90 because of the switching power supply's efficiency rating based off the datasheet. Then the maximum power P_{max} and average power P_{Avg} can be calculated as follows, where i_{max} and i_{avg} represent the currents of each respective column:

$$\begin{aligned} P_{max} &= \sum_{\forall i_{max}} i_{max} \\ &= (14mAh + 800 * 2mAh + 2 * 25mAh + \\ &15mAh + 3 * 6mAh + 2 * 2.5mAh)/0.90 \\ &= 1875mAh \end{aligned} \tag{4}$$

$$\begin{aligned} P_{Avg} &= \sum_{\forall i_{avg}} i_{avg} \\ &= (14mAh + 2 * +300mAh + 2 * 25mAh + \\ & 15mAh + 3 * 2.5mAh + 2 * 0.85mAh)/0.90 \\ &= 761mAh \end{aligned} \tag{5}$$

In order for us to have one hour of battery life, we need to have a battery with capacity of at least 761 mAh for the average case. Because of that, we picked a 9.6V battery, 2000 mAh NiMH battery. With this calculation, our worst-case lifespan will be about 1 hour. However, the robot should never be stalling for more than a few seconds at a time. The user should pay careful attention when operating the robot to make sure this is the case.

2.3 Motor Control

2.3.1 Motor Driver

This module consists of two ZXBM5210 Full-Bridge DC motor drivers (rated at 1A current output). A single motor driver circuit is shown in Figure 4. This H-bridge chip was chosen because it could withstand the 800mA stall current of the motor. The two H-bridges are each provided 6V by their own voltage regulators. Each motor is controlled by its own H-bridge.

The PSoC is used to send the correct outputs to the H-bridges in order to control the direction and speed of each motor. Pulse width modulation (PWM) is how we control speed. The direction can be controlled by swapping the input pin that receives the PWM. Figure 5 shows how the PWM is set up in hardware in the PSoC. The demultiplexers send the PWM signal to the correct output pins for the motors, depending on the control registers. The pin outputs required are detailed below in section 3.4.

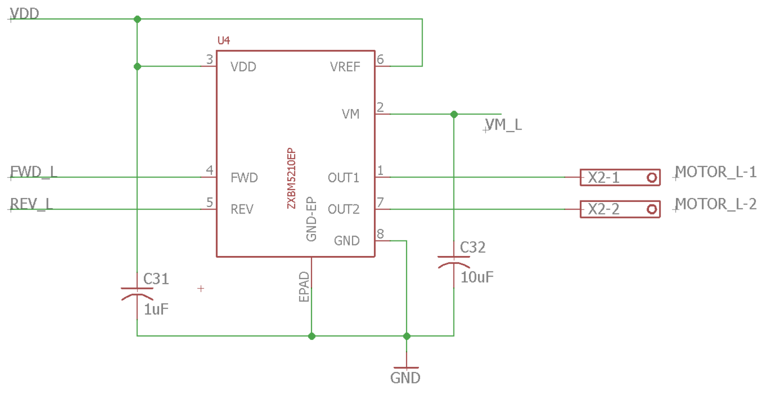


Figure 4: Circuit schematic for one motor driver circuit.

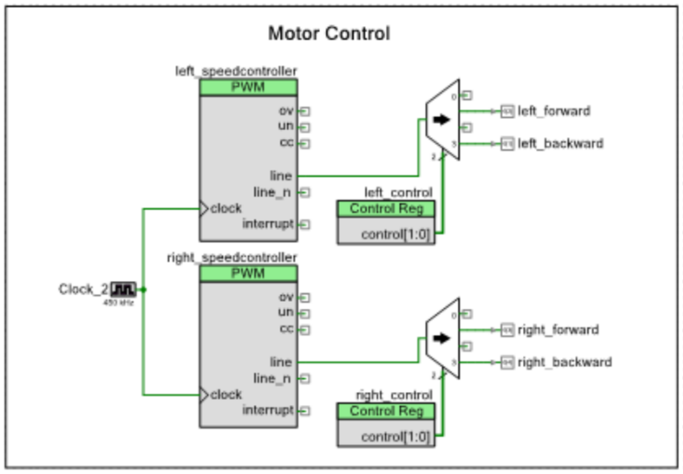


Figure 5: PWM and demultiplexer block to control motors in PSoC.

2.3.2 Motor Selection

The Tamiya 70097 dual motor gearbox kit to drive the robot. The motors used in this kit are two brushed DC motors each with a 800mA stall current while being powered by 6V. These motors are each powered with 6V and controlled by their own H-bridge. Each motor has their own H-bridge and voltage regulator because of the 800 mA stall current and 70 mA free-run current. The motors are in a differential drive configuration, and turning is accomplished by moving the motors in opposite directions. We chose this motor because it uses a higher voltage, and has a smaller stall current.

2.4 Sensors

2.4.1 IR Sensors

This module consists of two QRE1113 SparkFun Line Sensors. These sensors are powered with 5V and provide analog outputs between 0V and 5V representing the amount of reflection detected. This data is used to distinguish white from black which in turn controls which motors to run to allow line following. The sensors are placed side by side and attached to the bottom of the robot,in the front, and facing the ground. This data is sent to the PSoC GPIO pins through the shield and utilized to allow line following capabilities when line-following mode is selected from the phone application through BLE.

Since these IR sensors are analog sensors, an Analog to Digital Converter (ADC) is needed. This is done with an ADC hardware block in PSoC. Figure 6 shows how this is implemented in PSoC Creator. Specifically, only channel 0 and channel 1 are enabled and each channel is wired to an input pin.

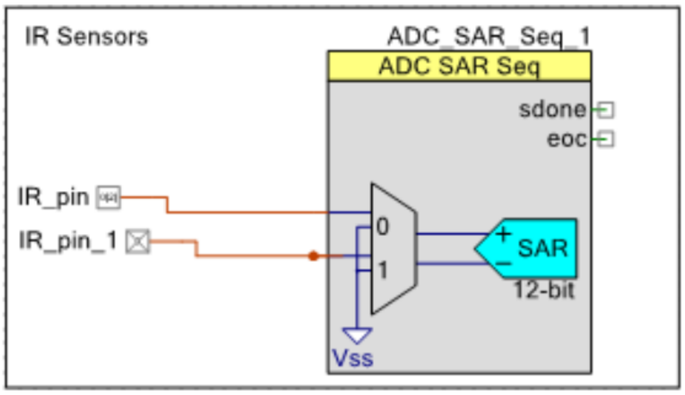


Figure 6: ADC hardware block used to convert the analog IR sensor signals into usable digital signals.

2.4.2 Ultrasonic Sensor

The HC SR-04 sensor provides us with information about the distance of possible obstacles in front of the robot with a range of 2cm to 400cm. It is powered by 5V and communicates with the PSoC through two I/O pins interfaced through the shield.

One more hardware component related to the ultrasonic sensor is necessary. Specifically, one timer is needed to measure how long the ECHO pin is held high which will then be converted into a distance in software. Figure 7 shows how this is implemented in PSoC with a Timer Counter. There is one output pin named TRIGGER_pin as well as an input pin called ECHO_pin which is connected to the timer. The timer counter is a one-shot up counter set to capture mode. When the trigger pin is toggled by the sensor, the timer counter will start counting up in microseconds. When the ECHO pin goes back low (0V), an interrupt will be generated at which point the distance can be calculated.

The period of the timer counter is chosen as follows. Equation 6 states the distance conversion as stated by the data sheet [3]. Let t be the time of the pulse recorded from the ultrasonic sensor. I set it equal to the max distance the sensor can theoretically read which is 500 cm.

$$\begin{aligned} \frac{t}{58} &= 500 \\ t &= 29000 \end{aligned} \tag{6}$$

Therefore, since the normal distance ranges we are concerned with are only within the 30cm range, it is safe to set the period of the counter to just above the value calculated in Equation 1, or $2^{15} - 1 = 32767$.

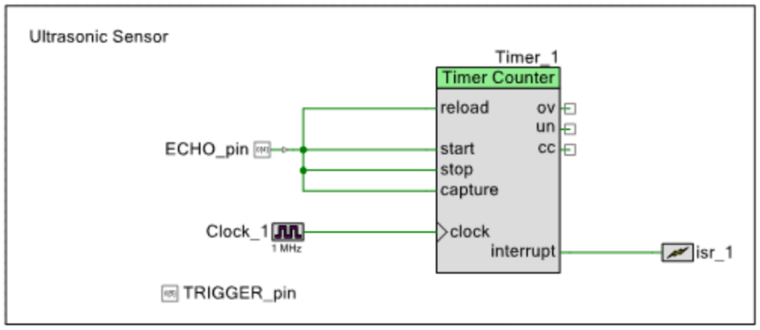


Figure 7: Implementation of Timer Counter for ultrasonic sensor.

2.5 Robot Chassis

The Tamiya 70098 Track and Wheel set is used as a base for the robot, and movement is accomplished through the tank treads on the set. The Tamiya 70097 gearbox and motors that we described above powers this section, and turning will be controlled as described above, with a differential drive. The two gearbox outputs each drive one side of the drive. The PSoC and shield are mounted on top of this kit. Sensors and motors can then be connected to the shield. Figure 8 shows a picture of the mobile platform.

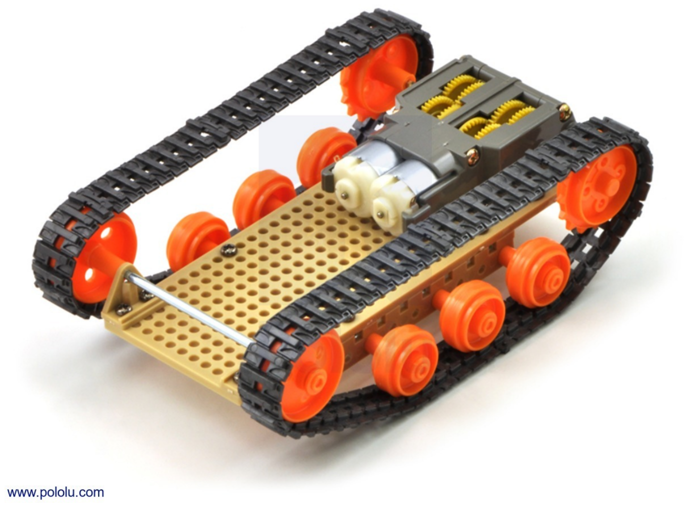


Figure 8: Chassis of robot kit. The PSoC, shield, and sensors will mount on top of this base. Photo from Polulu [4].

3 Software Design

3.1 Software Flowchart

Figure 9 illustrates the high level flow control of the main loop. At each iteration, BLE events are processed and the sensors are read. The state of the robot is then read and the corresponding case is executed. At any time, if the phone gets disconnected, the loop will catch that in the beginning and wait for a connection before proceeding.

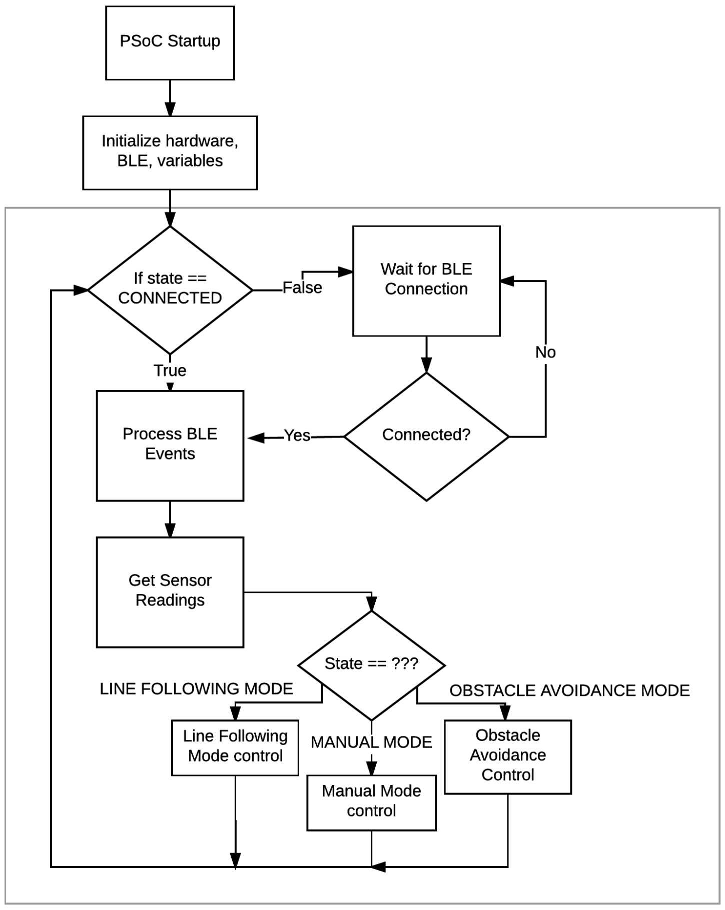


Figure 9: High-level flowchart of the main program loop. Box surrounding the actual loop.

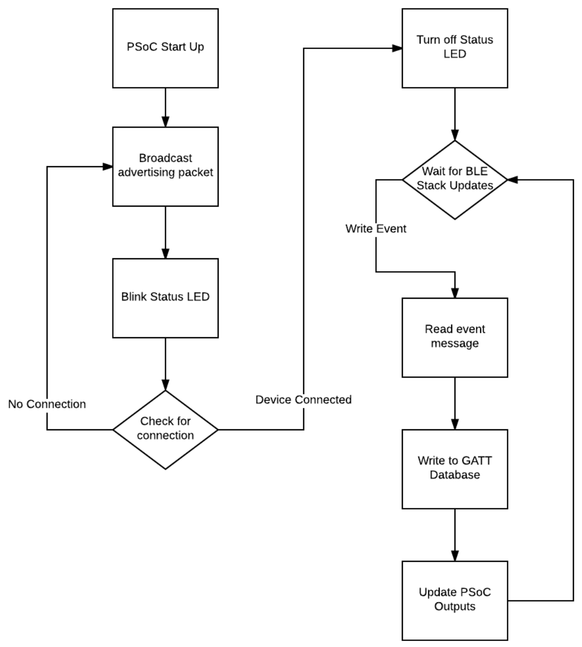


Figure 10: Flowchart detailing the process for receiving commands on the BLE board.

3.2 Wireless Control

This module consists of the Cypress CY8C-142 PSoC4 BLE module, which attaches to the robot shield. The BLE module is powered by 5V and provides serial communication to the PSoC and receive commands from another BLE device (ie. phone application). Commands sent through the BLE module are decoded in the PSoC and sent to the appropriate motor controller. A status LED on the shield indicates the connection state of the BLE module (ie. connected or not connected).

Processing BLE events on the PSoC works as follows. When the PSoC board is turned on, it initializes to the not connected state. The BLE module sends out advertising packets and the status LED blinks to alert the user. Once a device connects to the BLE module, the LED is turned off. Inside the BLE module, there is a Generic Attributes (GATT) database, which holds information about the characteristics of the device, such as the state. The BLE module then waits for write requests to the GATT database. Once an event is received, the GATT database is updated and the PSoC outputs are updated. If, at any time, the device disconnects from the other BLE device, the PSoC returns to the disconnected state and blinks the status LED. The flowchart displayed in Figure 10 illustrates this loop that runs on the PSoC to process and send BLE commands.

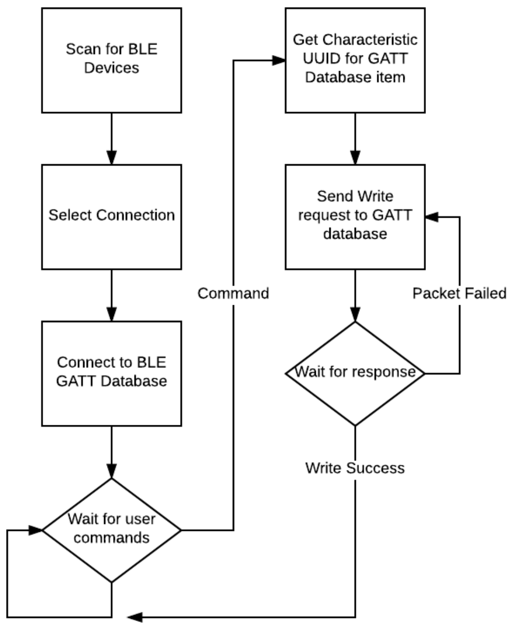

Scan for BLE Devices
Select Connection
Connect to BLE GATT Database
Wait for user commands
Command
Get Characteristic UUID for GATT Database item
Send Write request to GATT database
Packet Failed
Wait for response
Write Success

Figure 11: Flowchart detailing how the mobile application communicates with the BLE module.

On the other end, the phone application work as follows as illustrated in Figure 11. The application searches for the correct device and sets up communication with the GATT database. Once we receive a command from the application, we go the characteristic we want to write to, and then send a write request to the BLE module by using its UUID. These UUIDs were set when we initialized the GATT Database on the BLE module. Once we receive a success response, we can send more write requests.

3.3 Mobile Application

The application is designed for Android and utilizes the smart phone's built-in BLE module. The app can communicate with the robot and switch between its different modes. In the autonomous modes, the robot will control itself, but still send distance readings from the ultrasonic sensor and colors detected by the line sensors back to the phone from the sensors. In the manual control mode, the phone will act as a controller for the robot. Users will be able to control the motors on the robot to drive, and receive feedback from the on-robot sensors. Figure 12 shows the user interface of the designed app.

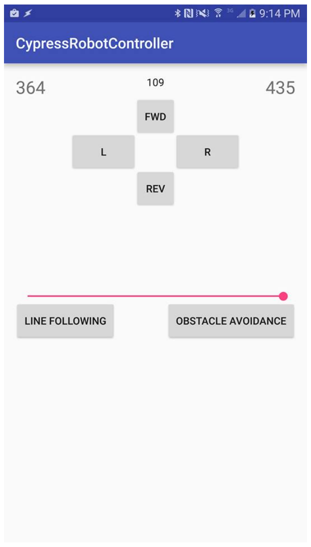
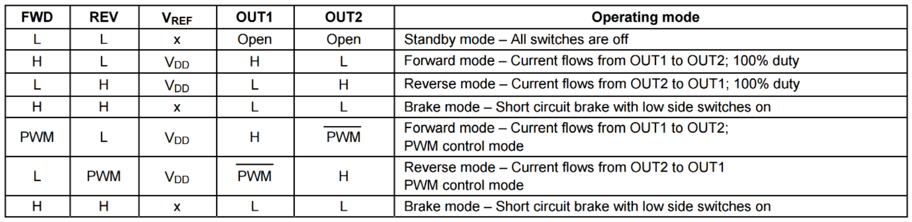


Figure 12: User interface of application that controls the robot.

3.4 Motor Control

Figure 13 shows the logic control used to control the speed and direction of each motor from the H-bridge datasheet. The speed can be controlled from the BLE application using a sliding bar and this will in turn increase or decrease a PWM global value in the code. To drive a motor, this PWM is written to the FWD pin of the motor while the REV pin is held low at 0V. To change motor directions, the PWM will be applied to the REV pin while the FWD pin will be held low at 0V.

To move forward, both motors will be sent a PWM signal to their FWD pin and their REV pin will be sent a low signal of 0V. To move backwards, both motors will be sent a PWM signal to their REV pin and their FWD pin will be sent a low signal of 0V. To turn the robot, one motor will be sent a PWM value while the other motor has its FWD and REV pins held low (0V). This will make it so one motor moves forwards (or backwards) and the other does not drive at all allowing the robot to turn.



FWD	REV	V_{REF}	OUT1	OUT2	Operating mode
L	L	x	Open	Open	Standby mode – All switches are off
H	L	V_{DD}	H	L	Forward mode – Current flows from OUT1 to OUT2; 100% duty
L	H	V_{DD}	L	H	Reverse mode – Current flows from OUT2 to OUT1; 100% duty
H	H	x	L	L	Brake mode – Short circuit brake with low side switches on
PWM	L	V_{DD}	H	$\overline{PWM}$	Forward mode – Current flows from OUT1 to OUT2; PWM control mode
L	PWM	V_{DD}	$\overline{PWM}$	H	Reverse mode – Current flows from OUT2 to OUT1 PWM control mode
H	H	x	L	L	Brake mode – Short circuit brake with low side switches on

Figure 13: Motor control logic table from datasheet [5].

3.5 Autonomous Movement

In addition to manual control through the phone application, this robot kit also offers two autonomous modes which can be selected by buttons in the phone application.

The first autonomous mode is line-following mode. In this mode, the robot will utilize the data from the two IR sensors to follow a black line on a white background. The algorithm is outlined in Figure 14. Upon selecting line-following mode on the phone application, the robot will continue following the black line until another mode is selected.

In obstacle avoidance mode, the robot will utilize data from the ultrasonic sensor to avoid crashing into obstacles. To get a distance reading, the trigger pin will be sent a high signal of 5V for at least 10us. Upon reading high pulse of 5V from the echo pin, a counter will be started to measure how long the echo pin is held high. This time measurement will then be converted into cm based off of constants found in the datasheet. Upon sensing an obstacle, the motor control should turn away to avoid it. The data from this sensor will be utilized when the obstacle avoiding mode is selected from the phone application through BLE.

Figure 15 outlines the obstacle avoidance algorithm. The robot will demonstrate autonomous movement by moving in a random direction until it encounters an obstacle. Upon obstacle detection, it will turn away from the obstacle and continue moving until it reaches another one. Obstacle detection will happen at a threshold of approximately 10cm. The robot will continue to autonomously avoid obstacles until another mode is selected.

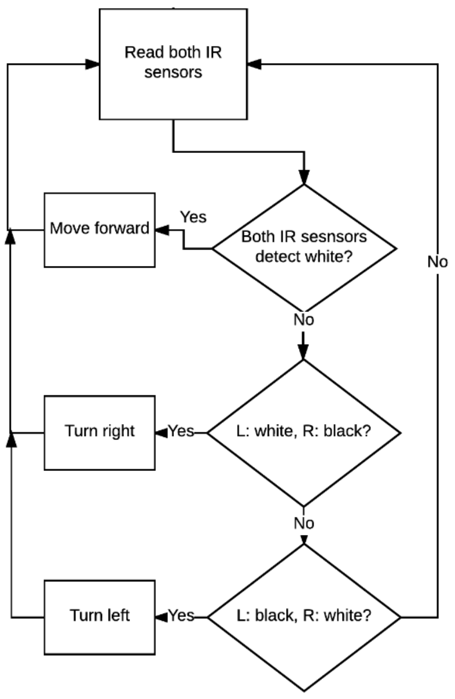


Figure 14: Flowchart detailing line following algorithm.

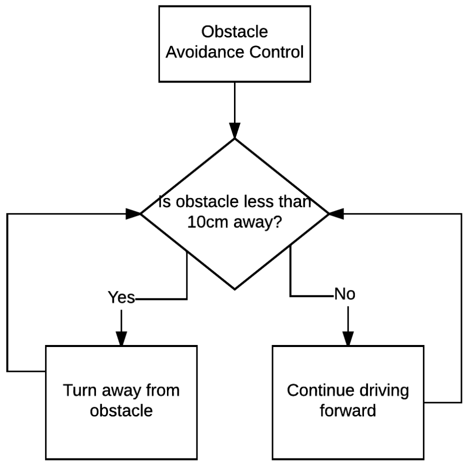


Figure 15: Flowchart detailing obstacle avoidance algorithm.

4 Verification

4.1 Power System

We verified our power system by probing multiple test points and measuring voltages and currents using the digital multimeter. First, we measured the voltage of the NiMH battery after fully charging and observed 10.6V, which is higher than rated voltage. This is due to overcharging and the voltage dropped to and settled around 9.6V as discharging power continuously.

To verify power regulators for 5V and 6V outputs, first we connected 6.3Ω load resistor to the 5V output and measured the voltage and current. We observed 5.01V and 798mA from the multimeter. Then we conducted similar test with two 6V output with 7.5Ω load resistor. We observed voltages of 5.988V and 5.99V, and currents of 795mA and 796mA respectively.

Lastly we verified the battery life by running the robot in line following autonomous mode for one hour without shutting off. We observed that the battery successfully power the robot for one hour itself without losing power.

4.2 Motor Control

To verify the requirement that our H-Bridge circuit can control a motor in the forward and reverse direction, we sent the correct output to the H-Bridge through the PSoC. We outputted LOW to REV, and HIGH to FWD and verified that the motor was spinning forward. We then sent the opposite commands, and verified that the motor was spinning backwards. To verify variable speed, we used PSoC to output a PWM signal with varying duty cycles (0, 25, 50, 75, and 100%) and verified that the robot moved at different speeds at different duty cycles.

To verify that the H-Bridge chip was able to supply the maximum current that our motor would draw, we placed an 7.5Ω resistor across the output pins, then verified that the voltage potential across the resistor was 5.9-6.1V.

4.3 Line Following IR Sensors

To verify that our Line Following IR Sensors could classify black and white colors, we first put the robot on top of a white poster board to read the values for white. We repeated this for the black color. From these numbers, a threshold was determined. Specifically, we classify any value over 1000 to be black and any value under to be white. The values we streamed to our Android application helped us verify that we could indeed distinguish white from black.

4.4 Obstacle Avoidance Ultrasonic Sensor

To verify that we could read the distance a robot was from an obstacle, we used a large box as an obstacle. The distance was streamed onto our Android app. A ruler was placed on the ground perpendicular to the box and the robot was tested at each centimeter interval from 5-400cm. At most, only an error of 1cm occurred at some intervals and therefore we could ensure our robot would be able to detect an obstacle within this range.

4.5 Wireless Control

We verified the phone application by running the app on the prototype robot. Once we connected, we saw that the robot did not move until we used the directional movement buttons, verifying that the robot was in Manual Mode. We tested each direction (forward, reverse, rotate left, and rotate right) and changed speed in order to verify manual control. Then, we used the Line Following and Obstacle Avoidance buttons to put the robot in the specified modes, and verified that the behavior of the robot was correct for the specified mode.

To test the disconnection safety feature, we put the robot into autonomous obstacle avoidance mode so that we could see the robot moving without us sending commands to it. We then disconnected our phone from the robot, and we verified that the robot stopped moving upon sensing the disconnection.

4.6 Shield Integration

Shield integration was verified mostly by visual inspection. First the PSoC BLE module must fit the two header connector sockets on the shield therefore the module is easily removable for installing and flashing. Also the shield itself was visually inspected so that it is not wider than the base, and that mounting holes aligned with the holes of the base for assembling together.

4.7 Software Requirements

The autonomous obstacle avoidance mode was verified by pressing the obstacle avoidance mode on the Android app and sending the robot toward a box. We observed that at 15cm (the threshold we set for this test) the robot detected the obstacle and turned away from it.

The autonomous line following mode was verified by pressing the line following mode on the Android app on our test track. The test track was made from white poster board and black electrical tape. The robot was able to successfully navigate the full circular track repeatedly.

5 Conclusion

5.1 Accomplishments

We were able to design a PCB with all the modules we wanted from our original goal. Our program and Android application were able to control the robot successfully with different operating modes. Finally we presented our fully functioning project at the demonstration. We also met our budget set by Cypress with the cost of the final product.

5.2 Uncertainties

One issue that we had was with our H-Bridge Chip. Though we were able to control direction and speed, we found that starting the motors at full speed or close to full speed (90% Duty Cycle and above) causes the motor behave erratically. The motor begins to stall as and overheats the chip. We fixed the issue by limiting the initial speed by lower than 90% Duty Cycle, however we are still uncertain about the erratic behavior of the chip when starting at the full speed.

5.3 Ethics

Our project follows the IEEE Code of Ethics with the following [6]:

1. To accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment.

The design specifications will check for the safety of the public, and all potentially dangerous items will be disclosed to the public below in the safety statement.

3. To be honest and realistic in stating claims or estimates based on available data.

All the above claims, such as the battery life, will be based on accurate calculations and have been disclosed in this document.

5. To improve the understanding of technology; its appropriate application, and potential consequences.

The goal of our product is to provide a safe base for utilizing the capabilities of PSoC, and to further improve other peoples' knowledge of the product.

6. To maintain and improve our technical competence and to undertake technological tasks for others only if qualified by training or experience, or after full

disclosure of pertinent limitations.

Design for this product has been undertaken with caution, and only done after understanding of the design.

7. To seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others

This project has been reviewed by the Teaching Assistants, Professors, and our peers in the Senior Design class, as well as Cypress Semiconductors, our project's sponsor. All resources that have been used in this document have been cited.

5.4 Future Work

In the future, we would like to improve the mechanical design, make it easier to add peripherals, and open-source our project. Currently, we use a spare PCB to hold our battery. We would like to provide more mounting options for our shield, and find batteries that fit underneath our robot, or provide a place to mount the battery.

We would also like to make it easier to add peripherals by making our breakout pins easier to access, and provide power options for those breakouts. Right now, signals can be outputted from those pins, but there is no dedicated 5V and Ground pin for those breakouts; future revisions of the board would have these. The last planned task is to open-source our project. The code would be cleaned up and made easily expandable, then all code and circuit schematics would be made available online for free, and provide some documentation with them. This would make it very easy for hobbyists to explore PSoC capabilities.

6 Cost

6.1 Labor

Name	Hours Invested	Hourly Rate	Total Cost = Rate*Hours*2.5
Alvin Wu	250	$30	$18,750
Byung Joo Park	250	$30	$18,750
Todd Nguyen	250	$30	$18,750
Total	750		$56,250

6.2 Parts

Part Name	Part Number	Unit Cost	Quantity	Total
PSoC 4 BLE Module	Cypress CY8CKIT-142	$9.97	1	$9.97
Twin-Motor Gearbox Kit	Tamiya 70097	$8.95	1	$8.95
Universal Plate Set	Tamiya 70098	$6.25	1	$6.25
Track and Wheel Set	Tamiya 70098	$6.85	1	$6.85
DC Motors	Polulu item #1117	$1.79	2	$3.58
Sparkfun Line Sensor Breakout (Analog)	QRE1113	$2.95	2	$5.90
Ultrasonic Sensor	HC-SR04	$3.95	1	$3.95
H-Bridge	ZXBM5210	$1.12	2	$2.24
Voltage Regulator	LT1376	$6.53	3	$19.59
9.6V NiMH Battery	Tenergy battery pack	$14.95	1	$14.95
Battery charger for 9.6V battery		$7.99	1	$7.99
Tamiya battery connector		$3.86	1	$3.86
10uH Inductor	VLF4014AT-100MR90	$1.08	3	$3.24
Schottky diode	MBR130T1G	$0.31	3	$0.93
1uF capacitor	CL31B105KOFNNNE	$0.10	3	$0.30
10uF capacitor	1206YC106KAT2A	$0.17	3	$0.51
Yellow LED	LTST-C230YKT	$0.21	1	$0.21
10uF tantalum capacitor	F931D106MAA	$0.38	3	$1.14
100uF tantalum capacitor	TL8A0107M010C	$1.12	3	$3.36
0.1uF capacitor	C1206F104K3	$0.33	3	$0.99
2200pF capacitor	C1206C222K5	$0.13	3	$0.39
Diode	1N4148W-7-F	$0.15	3	$0.45

4.99K Ω resistor	ERJ-8ENF4991V	$0.10	3	$0.30
7.32K Ω resistor	ERJ-8ENF7321V	$0.10	2	$0.20
5.36K Ω resistor	ERJ-8ENF5361V	$0.10	1	$0.10
10K Ω resistor	ERJ-8GEYJ103V	$0.02	3	$0.06
330 Ω resistor	ERJ-2GEJ330X	$0.10	1	$0.10
Rocker switch, 3pin SPDT	Pololu item # 1406	$0.75	1	$0.75
Connector header	SBH11-PBPC-D12	$0.38	1	$0.38
Connector header	SBH11-PBPC-D12	$0.72	1	$0.72
Connector terminal 2 position, 3.5mm	Pheonix contact #1984617	$0.41	2	$0.82
PCB fabrication		$12.60	1	$12.60
Total				$121.27

6.3 Grand Total

Parts	Labor	Grand Total
$121.27	$56,250	$56,371.27

References

[1] MakerFaire. The Maker Movement. Available:
http://makerfaire.com/maker-movement/

[2] Linear Technology. LT1376 Datasheet. Retrieved from
http://cds.linear.com/docs/en/datasheet/13756fd.pdf

[3] RobotShop. HC-SR04 Ultrasonic Range Finder. Available:
http://www.robotshop.com/en/hc-sr04-ultrasonic-range-finder.html

[4] Polulu. Tamiya 70100 Track and Wheel Set. Available:
https://www.pololu.com/product/106/#lightbox-picture0J2711;main-pictures

[5] Diodes Incorporated. ZXBM5210 Datasheet. Retrieved from
http://www.diodes.com/_files/datasheets/ZXBM5210.pdf

[6] IEEE Code of Ethics. Retrieved from
http://www.ieee.org/about/corporate/governance/p7-8.html

Appendix A: Requirements & Verification

Requirement	Verification
Power System [Total points: 10]	**Power System**
1. Battery must output above 9.6V ±0.1V with maximum current of 1600 mA. [0 points]	1. Verification Process for Item 1: (a) Attach 4.6Ω resistor as load. (b) Attach oscilloscope across load. (c) Measure voltage and verify that output is steady between 7.3-7.5V.
2. 5V DC Buck Converter must supply DC voltage of 5V±0.1V with maximum current of 800 mA. [4 points]	2. Verification Process for Item 2: (a) Attach 6.3Ω resistor as load. (b) Attach oscilloscope across load. (c) Measure voltage and verify that output is steady between 4.9-5.1V.
3. 6V DC Buck Converters must supply DC voltage of 6V±0.1V with maximum current of 800 mA. [4 points]	3. Verification Process for Item 3: (a) Attach 7.5Ω resistor as load. (b) Attach oscilloscope across load. (c) Measure voltage and verify that output is steady between 5.9-6.1V.
4.The robot system shall have an average battery life of at least one hour. [2 points]	4. Verification Process for Item 4: (a) Upload battery life test code onto PSoC which will drive the robot autonomously in obstacle avoidance mode. (b) Let the robot run for at least one hour. (c) Verify the robot is still running after one hour. Repeat (a)-(c) 2 more times.
Motor Control [Total points: 10]	**Motor Control**
1. H-bridge must be able to control each motor in forwards and backwards direction. [5 points]	1. Verification Process for Item 1: (a) From the PSoC, output LOW (0V) to the REV pin and HIGH (5V) to the FWD pin for both motors. (b) Verify that both motors are spinning in the forward direction. (c) From the PSoC, output LOW (0V) to the FWD pin and HIGH (5V) to the REV pin for both motors. (b) Verify that both motors are spinning in the backward direction.

Requirement	Verification
2. H-bridge must be able to control the speed of motors. [5 points] 3. H-bridge must be able to supply 6V$\pm$0.1V with a maximum current of 800mA for each motor. [0 points]	2. Verification Process for Item 2: (a) From the PSoC, output LOW (0V) to the REV pin for both motors. (b) Output 0, 25, 50, 75, 100% duty cycle PWM signal to the FWD pin sequentially. (c) Verify that the both motors are faster at each duty cycle. 3. Verification Process for Item 3: (a) Attach 7.5Ω resistor as load. (b) Attach oscilloscope across load. (c) Measure voltage and verify that output is steady between 5.9-6.1V.
Line Following IR Sensors [Total points: 5] 1. PSoC shall be able to classify between black and white colors from the IR sensor readings. [5 points]	**Line Following IR Sensors** 1. Verification Process for Item 1: (a) Place IR sensor over white surface. (b) Read ADC value of sensor and record. (c) Verify that the ADC value is less than 1024 (2.5V). (d) Place IR sensor over black surface. (e) Read ADC value of sensor and record. (f) Verify that the ADC value is greater than 1024 (2.5V).

Requirement	Verification
Obstacle Avoidance Ultrasonic Sensor [Total points: 5] 1. The PSoC shall calculate the correct distance in cm $\pm$ 1cm from 5cm up to at least 30cm from the ultrasonic sensor readings. [5 points]	**Obstacle Avoidance Ultrasonic Sensor** 1. Verification Process for Item 1: (a) Place the ultrasonic sensor at the beginning of a 30 cm ruler. (b) Place a flat and size-able obstacle such as a large notebook at the 5 cm mark, and verify the reading is within 4cm - 6cm (c) Repeat for each cm up to 30 cm
Wireless Control [Total points: 10] 1. Phone application can switch the mode of the robot from manual control, autonomous line following, and autonomous obstacle avoidance. [4 points] 2. In Manual Control mode, the user can move the robot forwards, backwards, rotate left, rotate right, and change the speed. [4 points]	**Wireless Control** 1. Verification Process for Item 1: (a) Connect to the robot through the BLE Application (b) Verify that the default state is manual control (c) Set the robot on a black line and switch the mode using the application to autonomous line following mode. Verify that the robot follows the line. (d) Using the application, switch the mode to manual mode again. Verify this by seeing if the robot has stopped moving. (e) Place the robot near wall or large obstacle. Using the application, switch the mode to autonomous obstacle avoidance mode. Verify this by seeing that the robot turns before hitting the wall. 2. Verification Process for Item 2: (a) Connect to robot through BLE application (b) Set mode to manual mode (c) Using the buttons on the phone application, move the robot forwards, backwards, rotate left, and rotate right, and verify that the commands correspond to the correct actions.

Requirement	Verification
3. Upon disconnection from the phone, the robot will turn off both drive motors and start broadcasting its BLE advertising packet. [2 points]	3. Verification Process for Item 3: (a) Establish connection between phone application and PSoC (b) Send forward command to robot (c) Disconnect phone and verify that the robot has stopped moving (d) Restart app and verify that the BLE module is broadcasting and can be connected to again.
Shield Integration [Total points: 5] 1. The PSoC module shall be able to be mounted on the top of the shield with compatible pin placements. [4 points] 2. The shield cannot be wider than the base of the robot kit (< 6cm ± 0.1cm), and longer (<16cm ± 0.1cm). [1 point]	**Shield Integration** 1. Verification Process for Item 1: (a) Place the PSoC module into the corresponding sockets on the shield. (b) Verify it fits correctly. 2. Verification Process for Item 2: (a) Measure the width and length of the PCB shield. (b) Verify that it is less than 6cm wide and less than 16cm long. (c) Visually inspect the shield when it is attached onto the robot and make sure it does not extend beyond the wheels
Software Requirements [Total points: 5] 1. When placed on white surface with black line in line following mode, the robot shall follow the black line. [2 points] 2. When in obstacle avoidance mode, the robot shall avoid all static obstacles. [3 points]	**Software Requirements** 1. Verification Process for Item 1: (a) Place robot on line-following track (b) Set the robot to the line-following state (c) Verify robot follows the black line 2. Verification Process for Item 2: (a) Set the robot to obstacle-avoiding mode (b) Let the robot run for one minute (c) Verify the robot does not crash and avoids obstacles

Appendix B: Shield Schematic

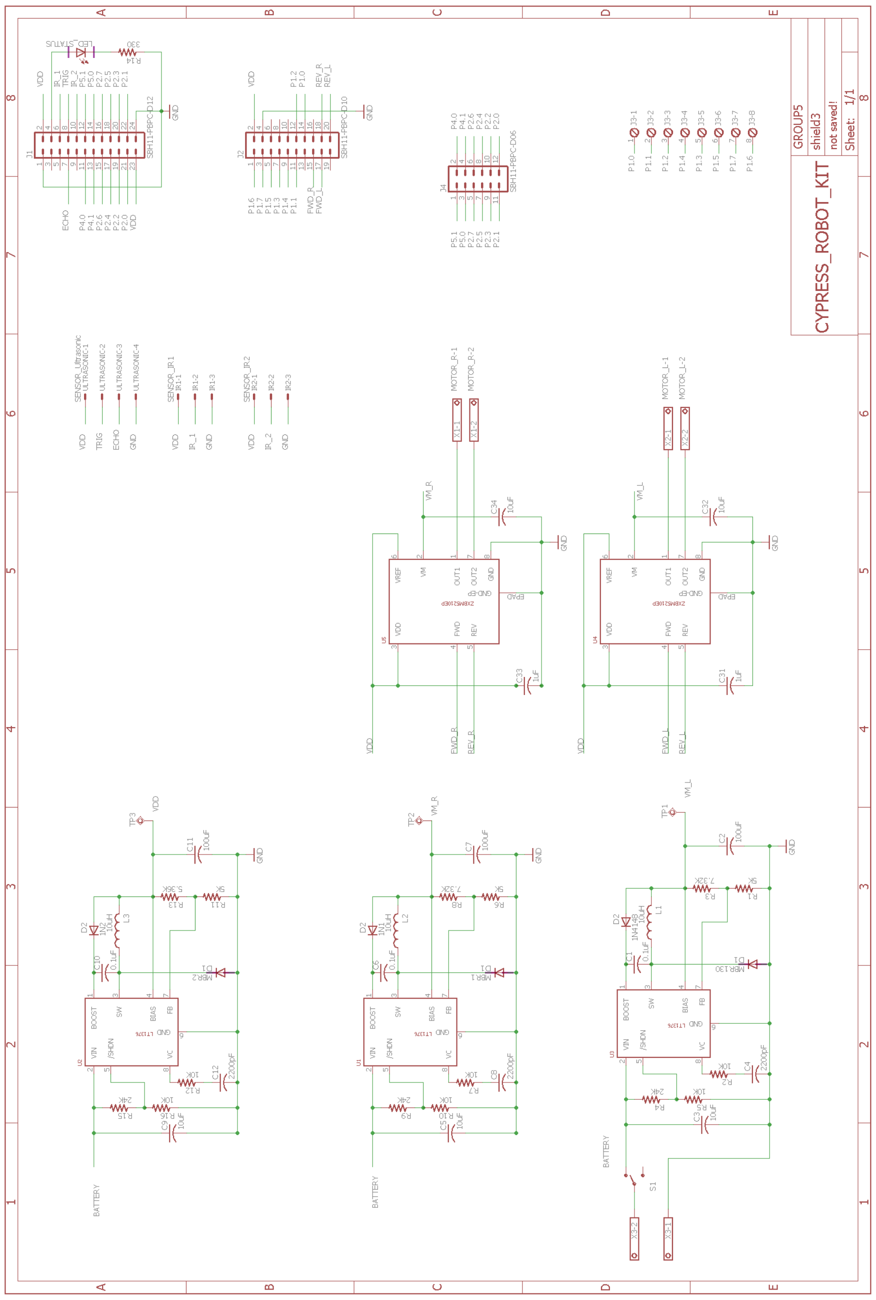


Figure 16: Circuit schematic of the overall system

Appendix C: PSoC Schematic

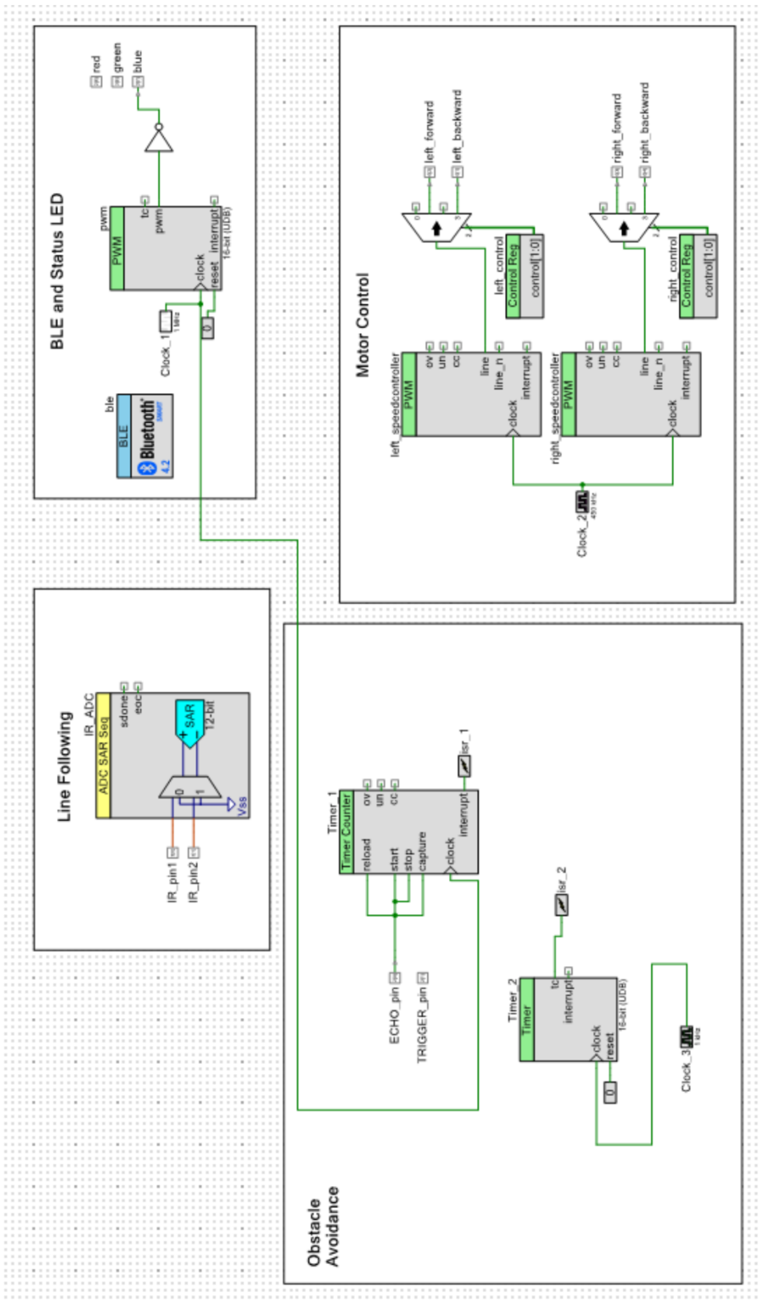


Figure 17: Top level hardware design in PSoC Creator.