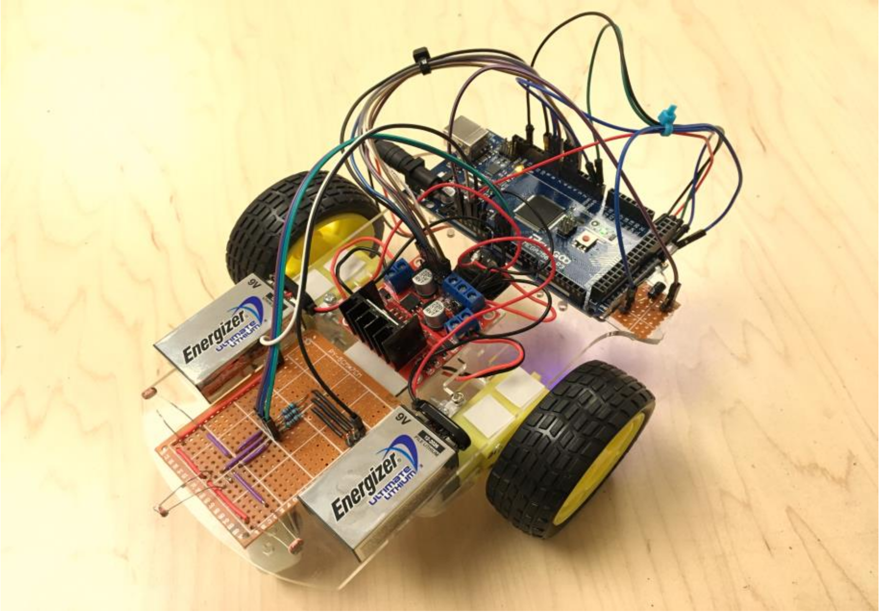
ECE 103

DYO Final Report



Team 09
Risto Rushford, cbr3@pdx.edu
Shengxin Ma, sm26@pdx.edu
Sohil Narayan, sohil@pdx.edu

Introduction

At the beginning of the Spring 2017 quarter, Team 09 was paired together to determine what type of project they wanted to pursue to fulfill the ECE 103 project requirements. Noah Page, who has since dropped out of the course, suggested that we do the Light Tracker variant of the project, but with an Arduino board. Upon some initial research on Arduino vs LabJack, the team (which also consisted of Risto Rushford and Shengxin Ma, agreed to use Arduino as the project's DAQ. Risto then suggested that instead of a stationary device, since the team was doing Arduino, why not make it into a light tracking robot instead. Thus, the Catbot prototype robot was conceived.

After Team 09 made the initial designs, ordered necessary parts and submitted the project proposal, Noah dropped the class. Risto and Shengxin continued to develop the project, and just after the first milestone Sohil Narayan joined the team to take Noah's place. With the latest addition to the team, the project could move forward significantly with many improvements on the original design, functionality, and programming aspects of the project with each team meeting. The developed project is outlined below.

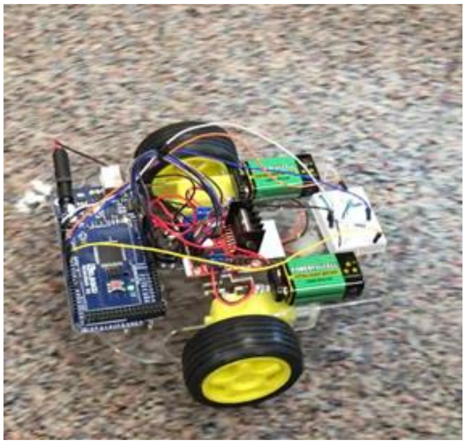
Hardware

Description

The Team 09 DYO light tracking device, named "CatBot" in honor of the pastime of entertaining cats with lasers, is fully functional and capable of interaction with a computer with a user interface via Bluetooth.

CatBot is built on a standard three-wheel chassis, with two motorized wheels and one support wheel. The motorized wheels are set just behind the center of gravity mounted on the underside of the board. On top of the board, centered between the two motors is a L298N Motor Drive Controller Board Module Dual H Bridge DC Stepper, which takes power from two 9V batteries (one for each motor) which are connected to a power switch, and relays commands to the motors from the Arduino board.

The Arduino board itself sits on the back of the wheel chassis, with its own battery supply mounted on the board just underneath it. In addition to the motor drive controller board, the Arduino is connected to a perf board at the front of the chassis with three photocell receptors which determine which direction the robot will travel, and another perf board behind it containing a Bluetooth module which provides serial port communication with a synced computer.



A 3D printed shell was originally planned to house the circuitry. The 3D model was made using AutoDesk's Fusion 360 Student Edition and saved as a .stl file which will be included with the report for consideration. The shell was on queue to be 3D printed at the L.I.D. but the staff did not keep on top of the queue and the item would not have been finished before the project demonstration as originally promised by L.I.D. management.

Schematic

There are three main circuit components in the project.

- Light detection, multi-photocell circuit
- Motor control module
- Bluetooth setup (using 3.3V as power input due to the limit number of 5V pin on the main board)

**All diagrams are listed in the Appendix D.*

Bill of Materials

- Elegoo Mega 2560 Arduino clone
- Emgreat® Motor Robot Car Chassis Kit
- 2 3-12V DC gear motors with wheels
- Qunqi L298N Motor Drive Controller Board Module Dual H Bridge DC Stepper
- 3 CdS photo resistors
- 3 220 Ohm resistors
- 2 5x7cm perf-boards (one modified for fit)
- 9V battery in enclosure with switch
- 2 non-enclosed 9V batteries in parallel with switch
- Adafruit BlueFruit EZ-Link Bluetooth module
- 1uF capacitor
- Jumper wires

Software

Description

The core Arduino sketch was developed for the robot to take input based on light sensitivity and perform basic actions in response. The program reads analog input from the photo resistors and compares the values. If the forward-facing photocell has the least resistance, then the robot will move forward. However, if the right photocell has the least resistance, then the right motor will stop, and the left motor will continue until the robot has turned toward the brightest light source and the forward-facing photocell again has the least resistance. The process is similar for the left facing photocell. Additional functionality was added later for analog motor control, serial input for receiving commands, and diagnostic functions to test individual components.

The program has basic diagnostic functionality. The diagnostic utility is a standalone program which uses PDCurses to provide advanced terminal output. The utility prints the analog values read from the photoresistors and listens for keyboard input. Upon receiving keyboard input, the program sends serial data to trigger component test functions in the Arduino sketch. The diagnostic utility may be used to check the analog readings from the three photocells, test steering or forward movement, disable movement, and test bidirectional analog motor control. Additionally, the utility works over a generic serial interface, so it can communicate using Serial-over-USB as well as the simulated serial port provided by the BlueFruit EZ-Link Bluetooth module.

**Logic Flow included in Appendix E1 and E2, all source codes are uploaded along with the report.*

Problems & Solutions

Team 09 has encountered several challenges over the course of this project. For the sake of this report document, one main problem will be described for each stage of development for the project.

1. *Project Proposal*

At the beginning of the quarter, no one in Team 09 had experience using Arduino, only LabJack as a DAQ. The team solved this by having each member acquire his own Arduino, or Arduino clone, to use for practice and eventually to serve as the computing hardware of the project.

2. *Milestone 1*

After the proposal to use an Arduino had been submitted and the project concept was approved for development, one of the team members ceased all communication. Shengxin and Risto were left to learn to use Arduino on their own. Fortunately, because of the steps taken to solve the previous listed problem, they had their own Arduino boards and accompanying kits to practice with. There was a point of confusion in determining how to program the board to send the correct signals to the DC motors, but Risto and Shengxin took turns writing code to test the motor functions until finally it was made to work.

3. *Milestone 2*

At this point, Sohil had joined the team to replace Noah. Risto, who previously made a test circuit for the photocell array had figured out how to combine the code into a diagnostic that also ran the motors. There were some initial problems in the responsiveness of the device. Sohil made his first major contribution by helping the team to debug the problem. Just prior to Milestone 2 the team was ready to integrate a Bluetooth module and begin developing a user interface.

4. *Final Report / Demonstration*

The user interface has turned out to be the single greatest challenge of this project. Initially, the team researched the possibility of using an alternative to the Arduino IDE which could be used to upload the Arduino sketch file and then use command line prompts to control the device. This turned out be problematic, and the best solution was to code the program in C++ writing to the serial port. With this innovation, the team was able to continue using the standard Code::Blocks IDE and, using the pdcurses library, create a user interface which communicated directly with the Arduino even though the Arduino itself was programmed using the Arduino IDE.

Discussion

According to Risto, the ECE 103 class project went much than the ECE 102 project before it. The lesson learned was that when working as part of an Engineering Team, it is important for each team member to be reliable in the development of a product. Each team member by the end of the quarter was committed to developing the project with quality work and seeing it through to completion.

Each person had strengths and weaknesses that they brought to the team. Because each team member kept an open mind about learning how the other team members did things, each person could be challenged and improve in their knowledge and understanding of the engineering process and particularly on how to write C programs.

Signature Paragraph

Member Contributions

Shengxin -

- Designed and coded test circuit for the motor control module
- Document meeting details on Trello
- Final editing and review of each report
- Setup the chassis board
- Plot the circuit diagram for the motor control, photocell and the Bluetooth wireless connection
- Plot the flowchart for the Arduino code and wireless interface control

Sohil -

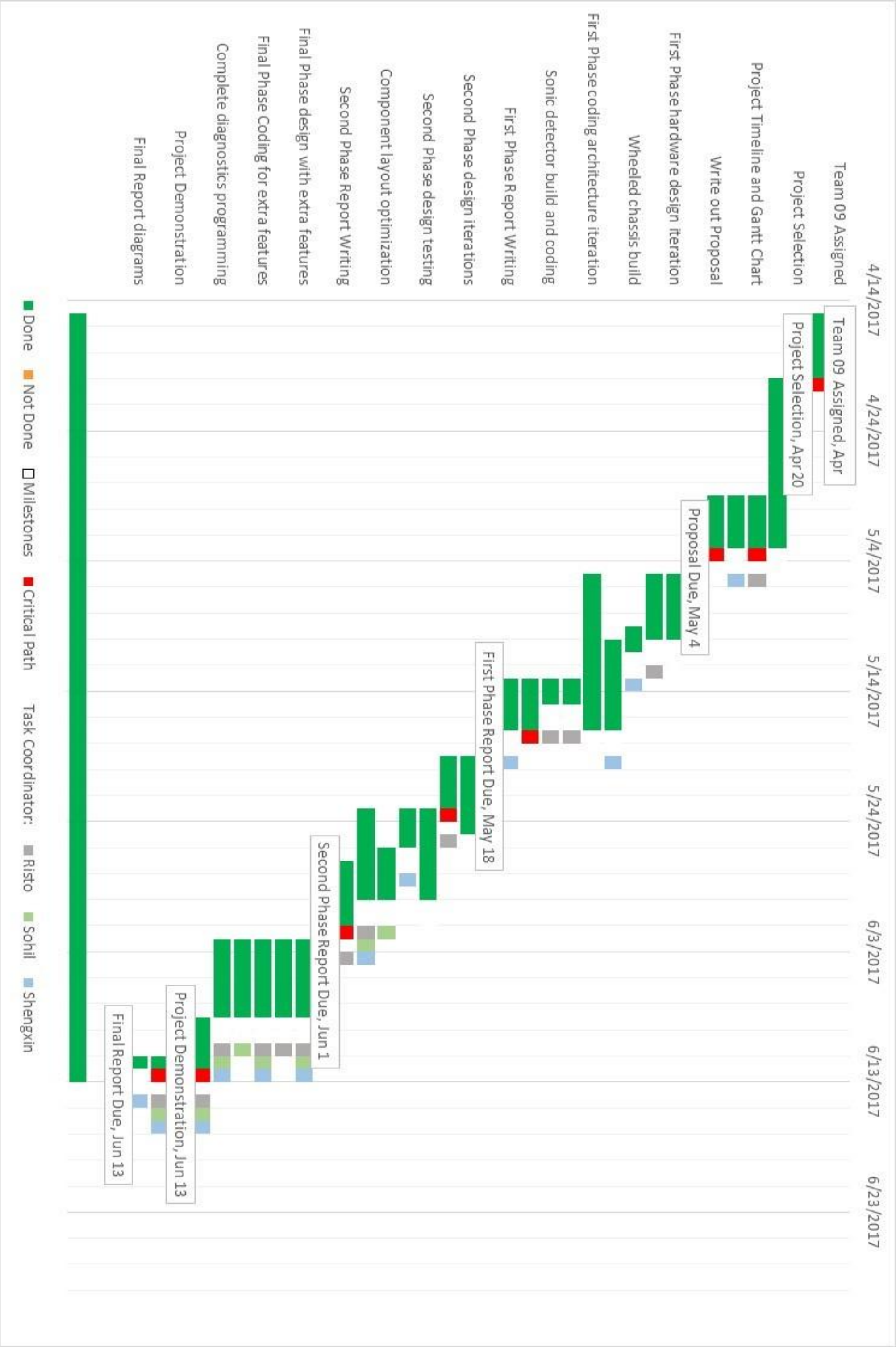
- **Designed and soldered primary photoresistor analog sensor PCB** (pictured in Appendix C, Fig 2)
- Researched PDCurses/NCurses libraries
- Researched serial communication methods using C++
- **Programmed debug and diagnostic interface using C++ and PDCurses** (See CodeBlocks project: CatBotDiag.cbp > main.cpp)
- **Programmed final version of Arduino sketch** including analog motor control, serial input and output, and diagnostic functions. (See Arduino sketch: catBotV2.ino)
- Various design improvements and fixes to robot chassis components
    - Rearranged components
    - Cleaned up wiring
    - Added power switch inline with motor power
    - Soldered motor leads and improved connectivity to motor driver board
    - Added adhesive to motor mounts

Risto -

- Functional project manager.
- Report writer
- Designed and updated Gantt chart using EXCEL.
- Designed and coded test circuit for photocell array.
- Designed and coded test circuit for sonic sensor (not used in final product due to time constraints).
- Designed .stl file for shell casing to be 3D printed (due to time constraints and availability of 3D printers, not used on final product). See Appendix F for design images.
- Combined circuit and code for photocell array with motor shield.
- Added BlueTooth module to chassis
- Bought some chips

Appendix A

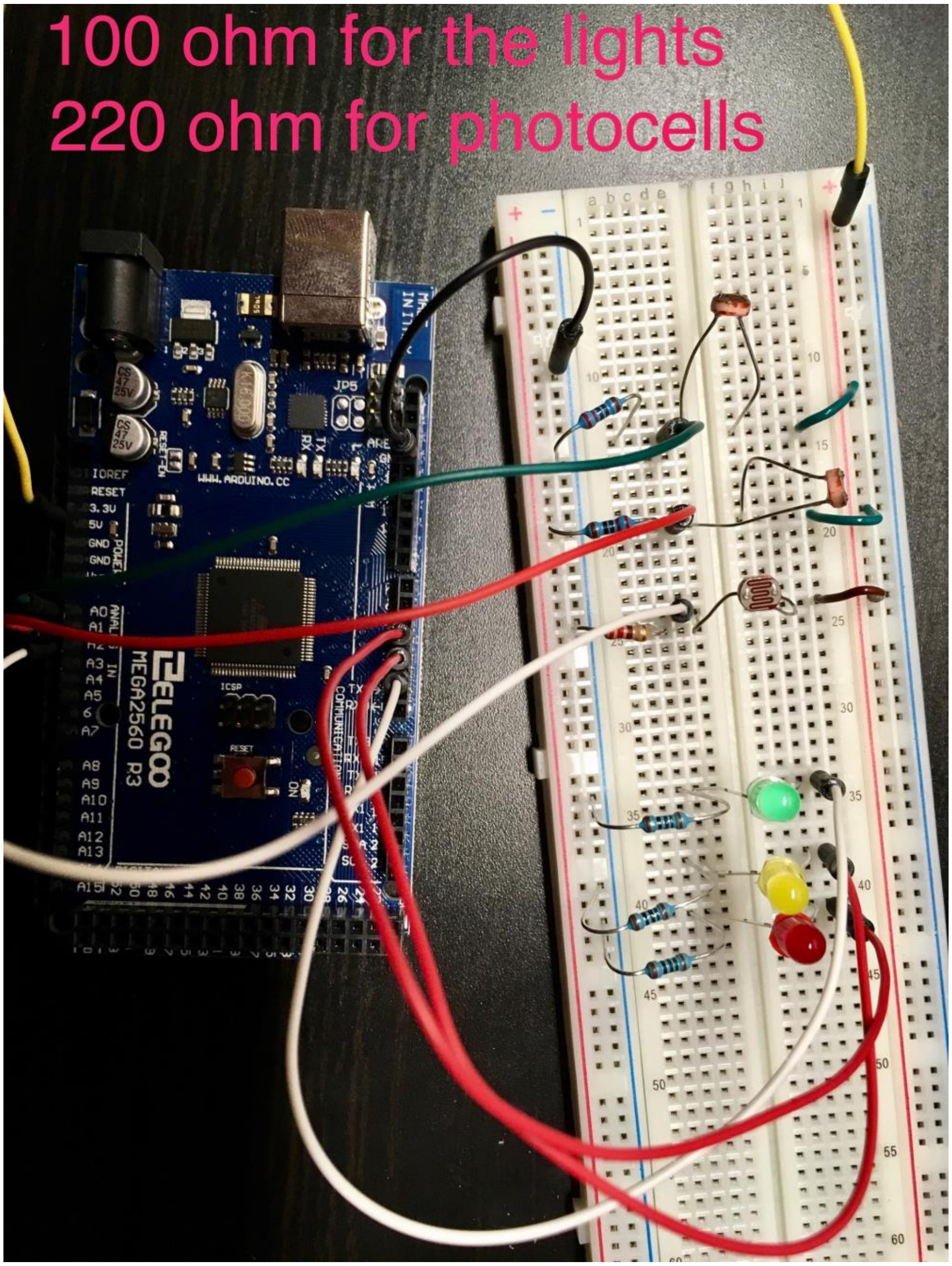
Team 09 Updated Gantt Chart (June 12th):



Appendix B1

Links to video documentaries on project development:

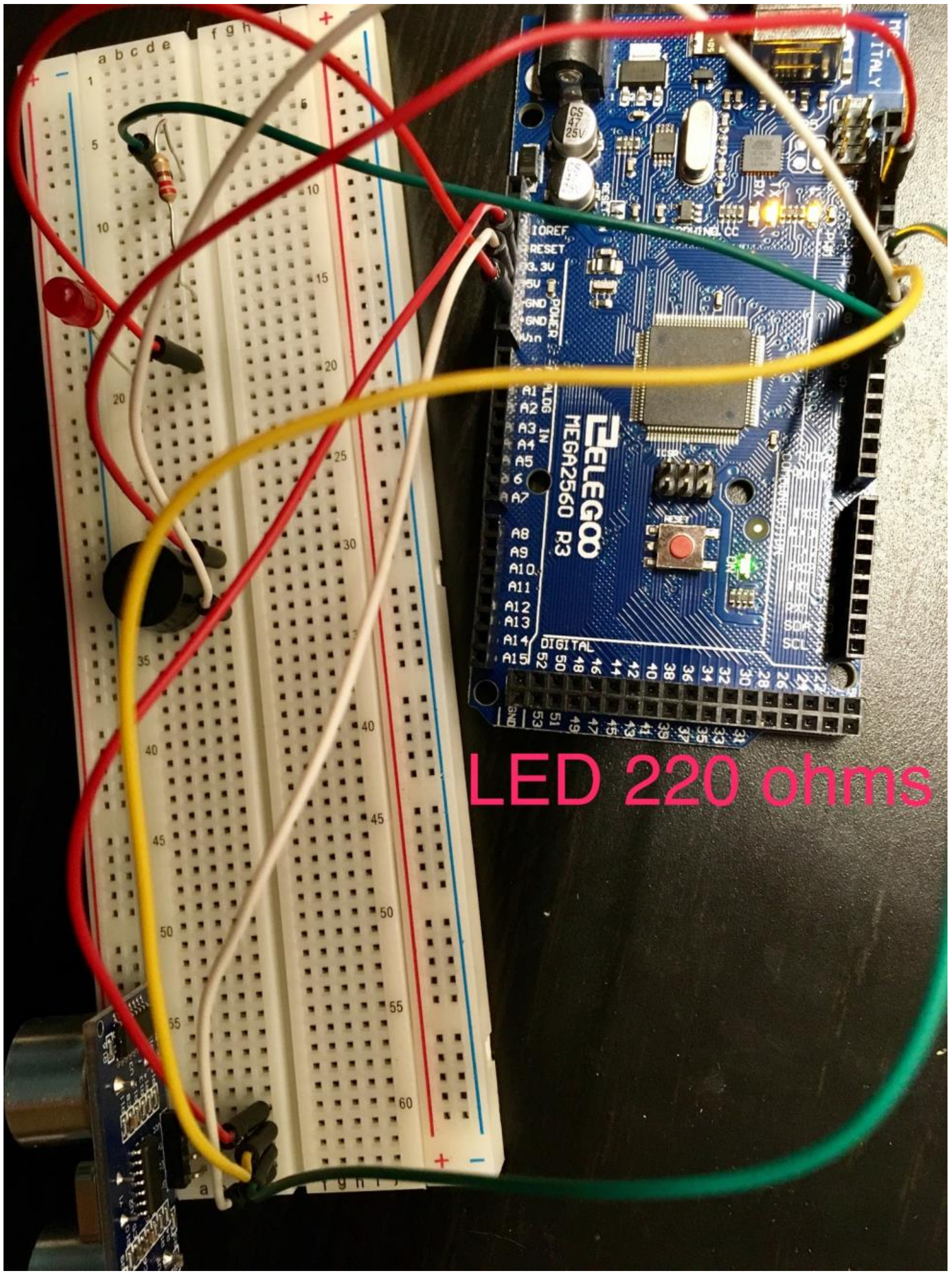
Testing the photocell sensors:
https://drive.google.com/open?id=0B4jCAOlsh45Db0ZmRERHSDh1bWc



100 ohm for the lights
220 ohm for photocells

Appendix B2

Testing the sonic sensor:
https://drive.google.com/file/d/0B4jCAOlsh45DSms2UWVabmVDSkE/view?usp=sharing



LED 220 ohms

Appendix C

Circuit diagram for the motor control module (High bridge L298N), (the L298N module in the diagram missing several pins for PWM)

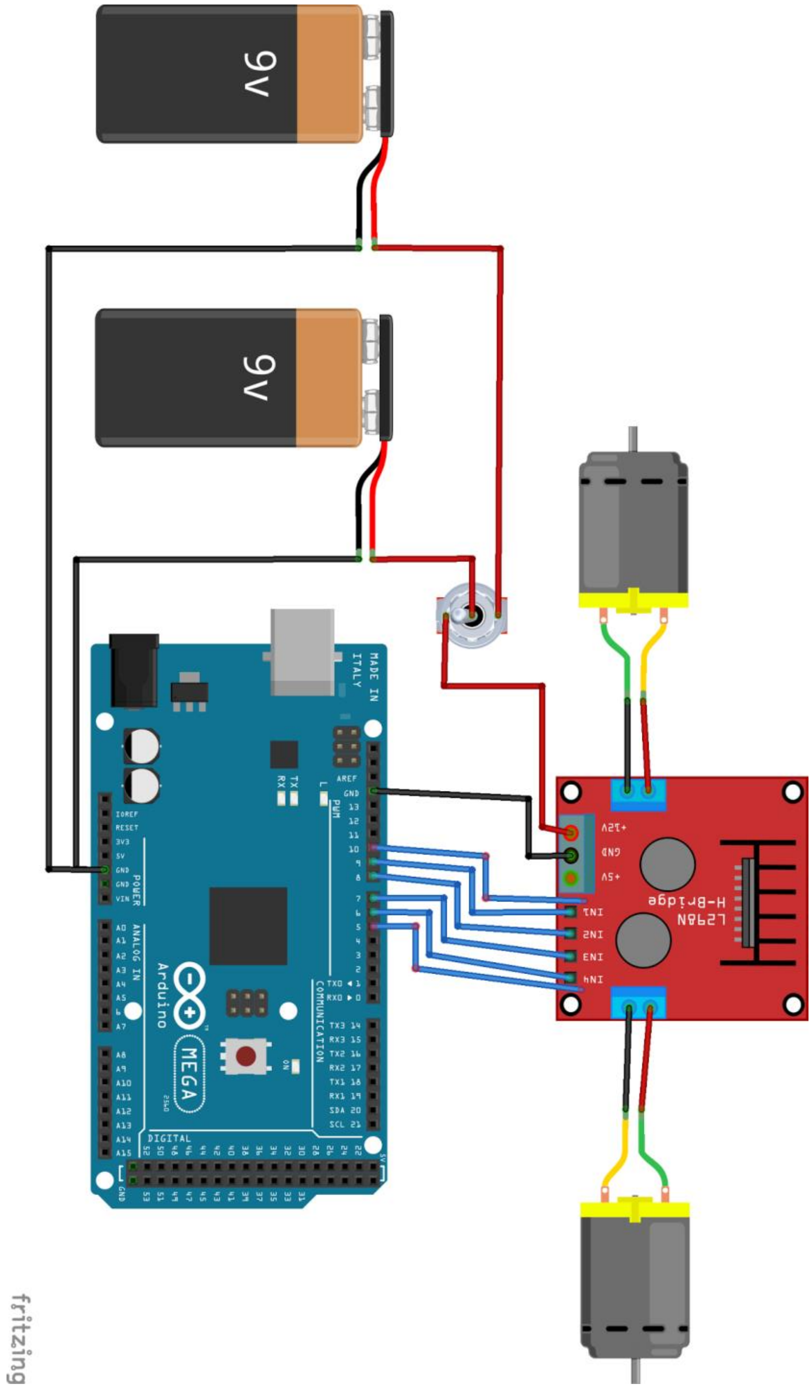


Figure 1

Circuit diagram for the photocell setup (220 ohm resistor, photocell):

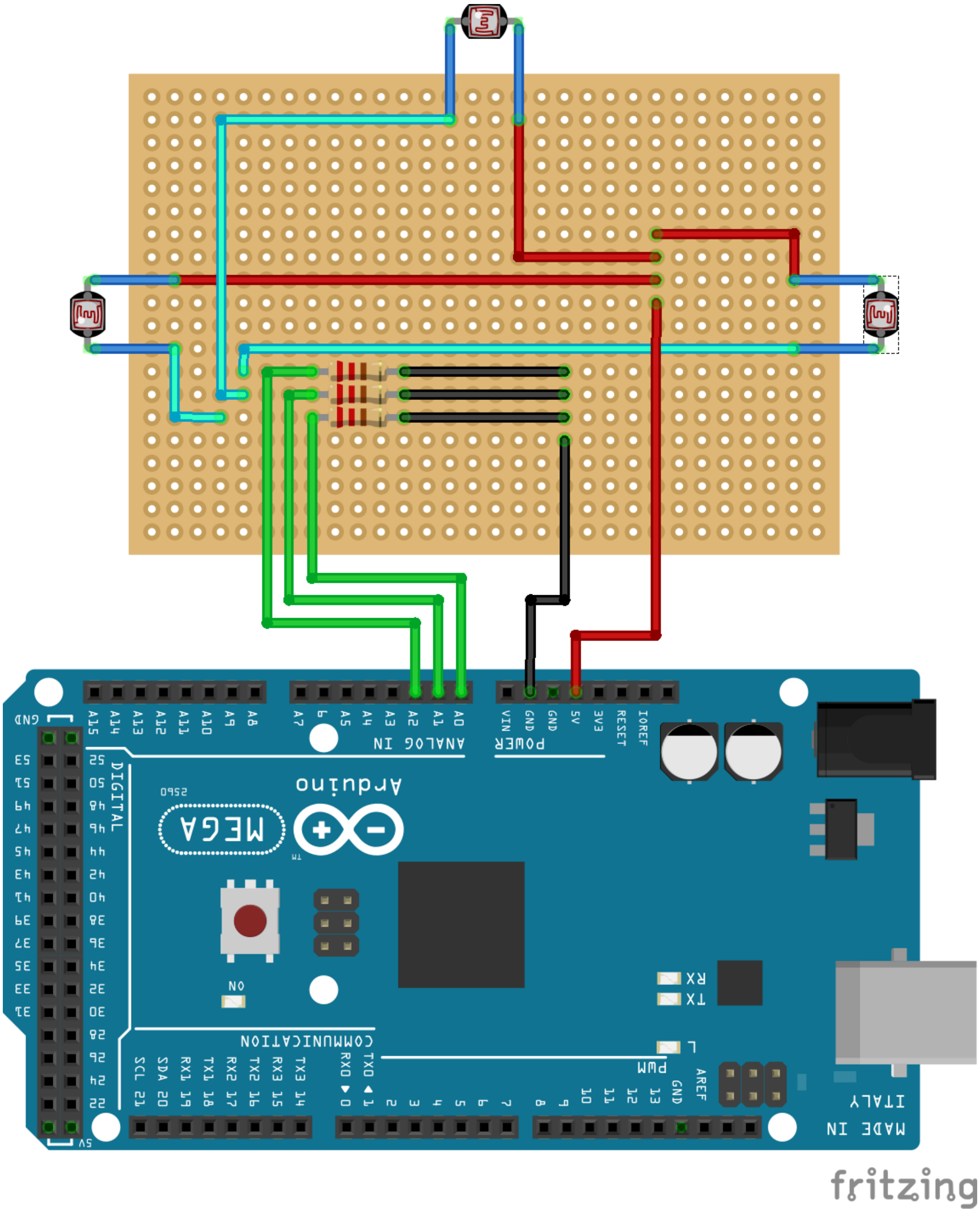
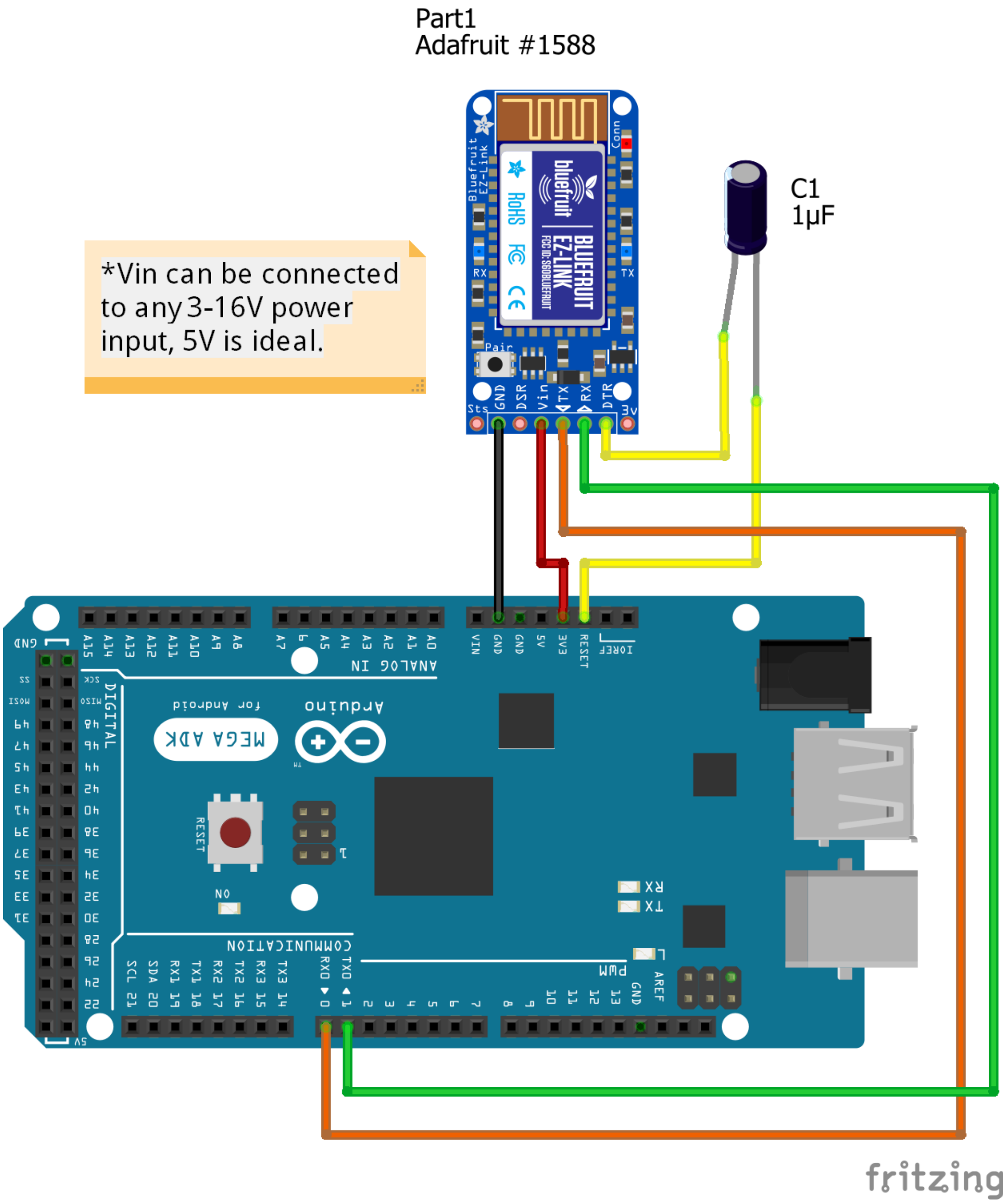


Figure 2

Circuit diagram for the bluetooth module (BlueFruit EZ-Link):

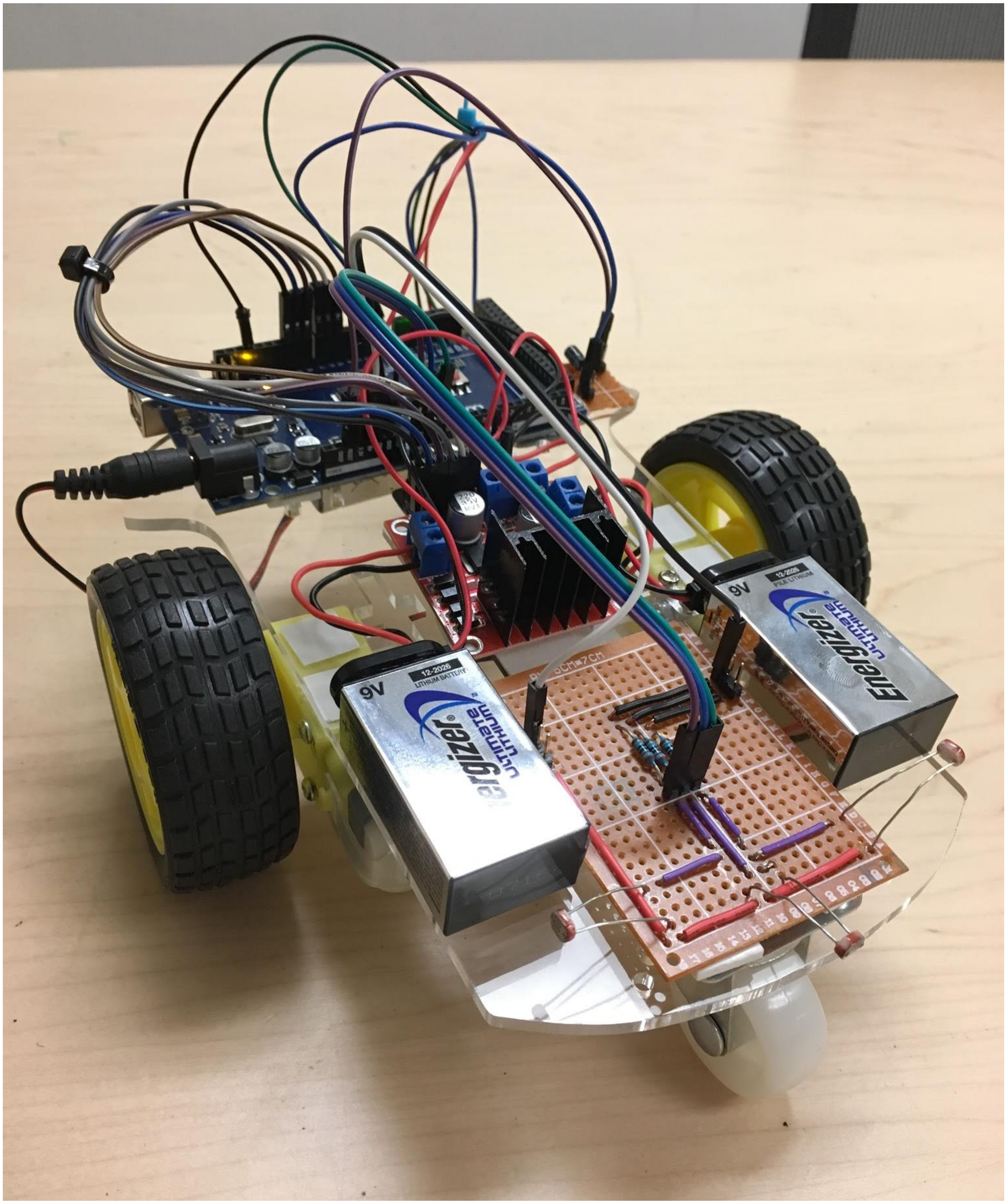


*Vin can be connected to any 3-16V power input, 5V is ideal.

Figure 3

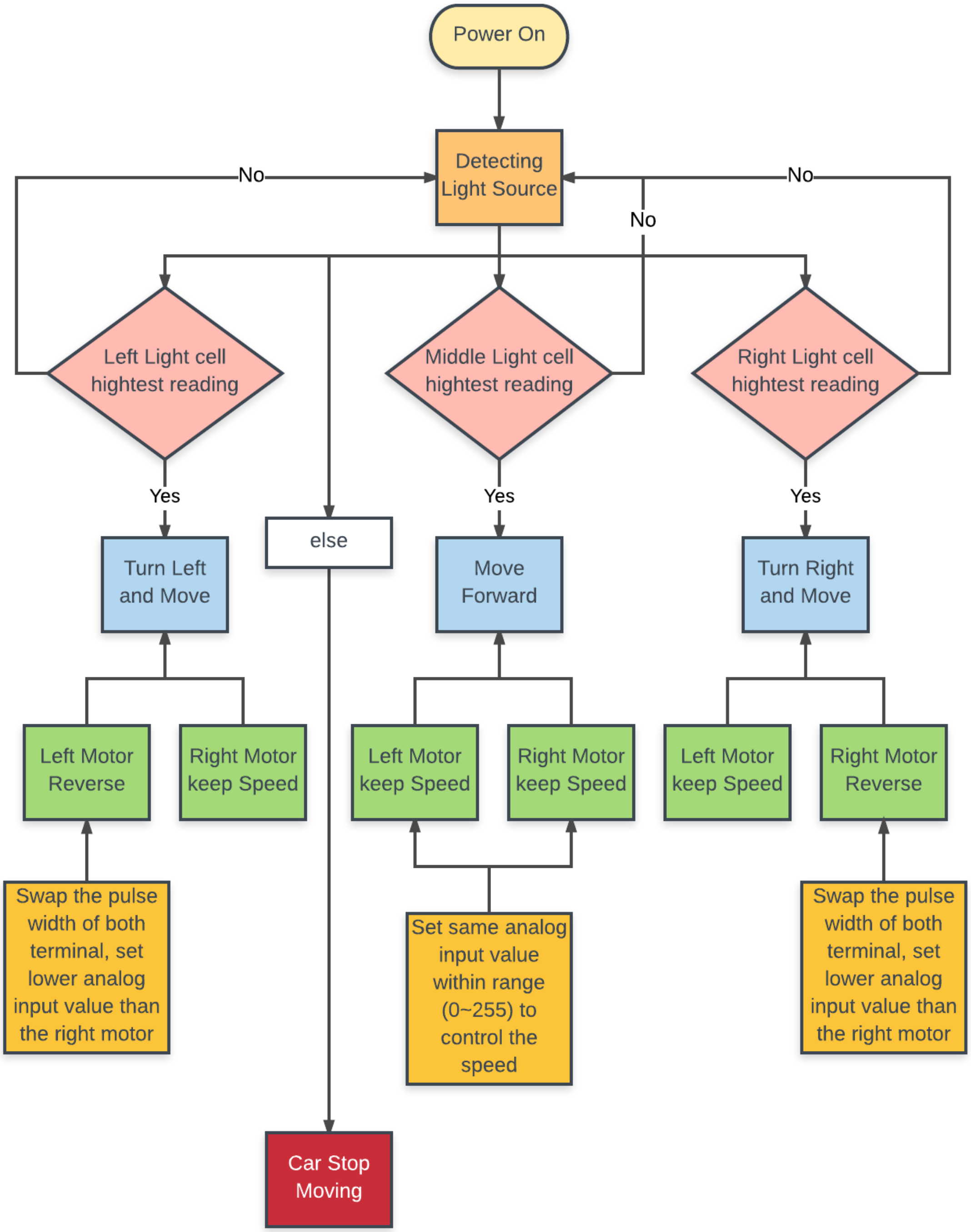
Appendix D

Final assembling:



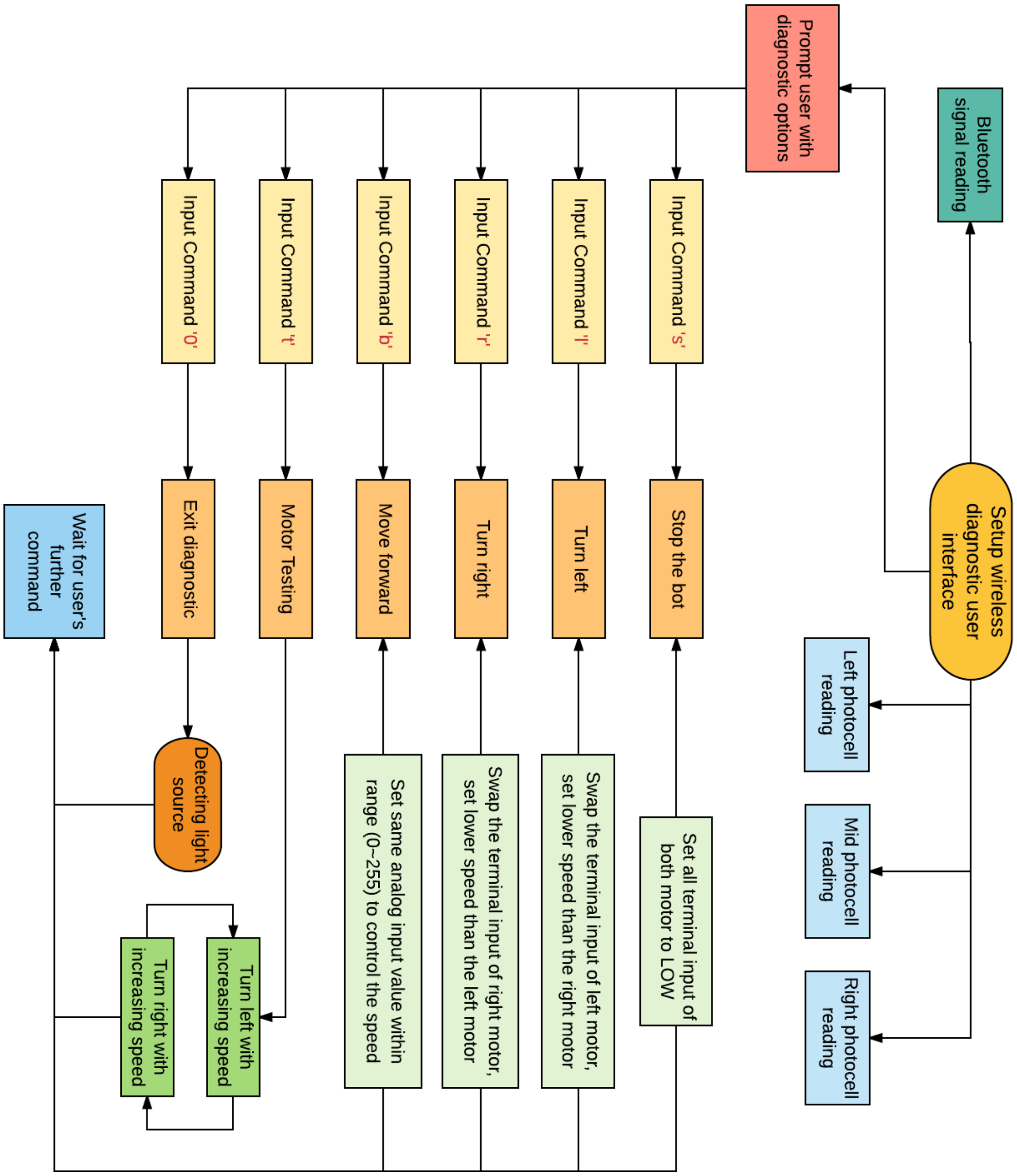
Appendix E1

Flow chart of the light detection and motor direction combined code (without wireless diagnostic command)


Power On
Detecting Light Source
No
No
No
Left Light cell hightest reading
Middle Light cell hightest reading
Right Light cell hightest reading
Yes
Yes
Yes
else
Turn Left and Move
Move Forward
Turn Right and Move
Left Motor Reverse
Right Motor keep Speed
Left Motor keep Speed
Right Motor keep Speed
Left Motor keep Speed
Right Motor Reverse
Swap the pulse width of both terminal, set lower analog input value than the right motor
Set same analog input value within range (0~255) to control the speed
Swap the pulse width of both terminal, set lower analog input value than the right motor
Car Stop Moving

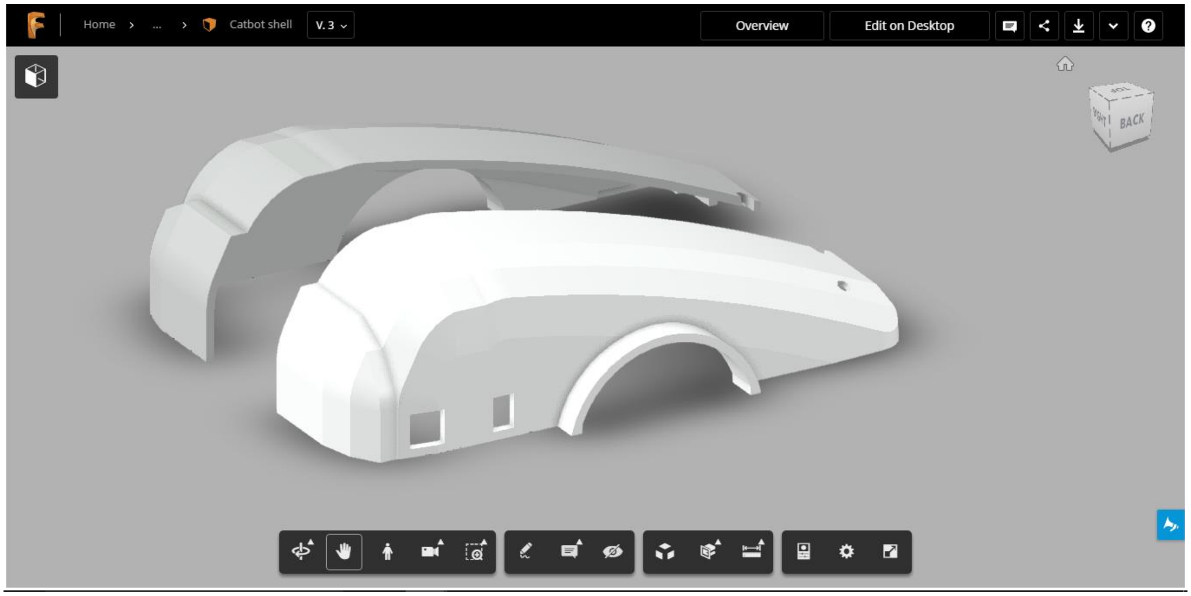
Appendix E2

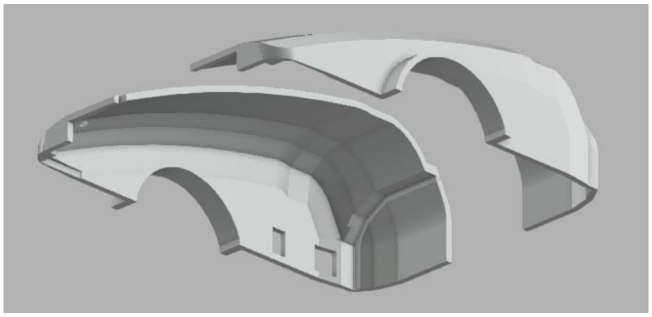
Flow chart of the wireless diagnostic user interface:

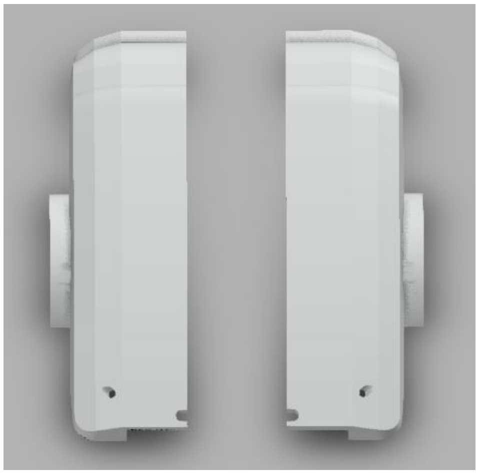

Bluetooth signal reading
Setup wireless diagnostic user interface
Left photocell reading
Mid photocell reading
Right photocell reading
Prompt user with diagnostic options
Input Command 's'
Input Command 'l'
Input Command 'r'
Input Command 'b'
Input Command 't'
Input Command '0'
Stop the bot
Turn left
Turn right
Move forward
Motor Testing
Exit diagnostic
Set all terminal input of both motor to LOW
Swap the terminal input of left motor, set lower speed than the right motor
Swap the terminal input of right motor, set lower speed than the left motor
Set same analog input value within range (0~255) to control the speed
Turn left with increasing speed
Turn right with increasing speed
Detecting light source
Wait for user's further command

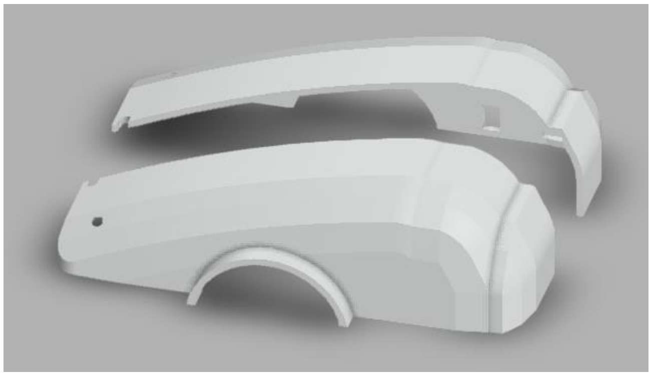
Appendix F

Images of CatBot outer shell design (not installed as of project demonstration) on AutoDesk Fusion 360.









References

"Tutorial - L298N Dual Motor Controller Module 2A and Arduino." *tronixlabs,* 22 Nov. 2014, https://tronixlabs.com.au/news/tutorial-l298n-dual-motor-controller-module-2a-and-arduino/. Accessed 6 June 2017.

"HOW TO: control DC Motors with Arduino + L298N." *LessonStudio on YouTube,* 7 Dec. 2014 https://www.youtube.com/watch?v=kv-9mxVaVzE. Accessed 6 June 2017.

"Installing Fritzing." *Fritzing,* 2 June 2016, http://fritzing.org/download/?donation=0. Accessed 8 June 2017.

"Getting and Using Arduino Serial Input." *Starting Electronics,* 19 March 2015, https://startingelectronics.org/software/arduino/learn-to-program-course/19-serial-input/. Accessed 10 June 2017.

Lady Ada. "Introducing Bluefruit EZ-Link Breakout." *Adafruit.com*, 21 Sep. 2016, https://cdn-learn.adafruit.com/downloads/pdf/introducing-bluefruit-ez-link.pdf. Accessed 10 June 2017.

"Adding PDCurses to MinGW." *Computing Tidbits,* 6 June, 2016, http://cects.com/adding-pdcurses-to-mingw/. Accessed 11 June 2017.

"Install a Library under CodeBlocks for MinGW." June 2017, http://ccm.net/faq/29303-install-a-library-under-codeblocks-for-mingw#indicate-the-compiler-where-to-find-the-h-files. Accessed 11 June 2017.