BUTTER PASSING ROBOT

By

Yuchen He

Yu Jie Hsiao

Yuxiang Sun

Design Document for ECE 445, Senior Design, Spring 2018

TA: Xinrui Zhu

17 February 2018

Project No. 18

Contents

1. Introduction

1.1 Objective

For the senior design project, we are trying to build a small robot that can find butter on the table and bring it back. This idea was inspired by the famous sitcom “Rick and Morty”. In one of the episodes, Rick built a robot which could fetch the butter once it received a verbal command from its owner.[1] Such robot has the ability to move around on its own, detect target objects, recognize human speech and recognize faces. To make our project useful but also manageable, we will mainly try to implement a small autonomous vehicle with an object detection module. We will also assume that the butter is cube-shaped, yellow in color, without boxing and placed in a plate.

1.2 Background

As the artificial intelligence gets more and more popular nowadays, we see an increasing number of “intelligent” robots that can make people’s lives more convenient. For example, floor mopping robots can automatically wander around the room and absorb dust on the floor. Smart audio speakers can recognize voice commands and play the specified music. With that in mind, we decide to build a robot that can be useful in daily life by utilizing AI technology. To be more specific, we want our robot to have the ability to find and bring back butter on a dining table.

We did some research and found two existing projects online[2]. However, both these projects used existing robotic platform and did not showcase a solid software program. We plan to build a robotic platform that has the ability to run object detection program itself. That being said, computer vision will be an import aspect of our project. Thanks to the rapid development of machine learning in recent years, we have many existing frameworks and libraries (OpenCV, LeNet, Tensorflow…) to choose from. Our plan is to try those frameworks and find the most suitable one for our hardware platform.

1.3 High-level Requirement

- The vehicle can move by itself on a regular-sized (~ 2m*1m) table.
- The vehicle can detect the edge of the table and it will stop in order to prevent itself from falling
- The object detection program can distinguish yellow, cubed butter from other common breakfast objects (juice, bread… etc) and direct the vehicle toward butter. It should have an accuracy of at least 80%.

2. Design

2.1 Block Diagram

To be successful in operating, the butter passing robot requires three sections: a power supply, a control module and a robotic platform. The power supply consists of one Lithium battery as power source. It also contains a boost converter to ensure that all the electronic devices can operate under acceptable voltage range and two power switches to protect the system from current overload. The control module consists of a microcontroller and a Raspberry Pi. Connected with a camera, the Raspberry Pi will be used solely to analyze the image taken by its camera module. The microcontroller will be used to read inputs from the infrared sensor and the Raspberry Pi. It will then output appropriate signals via H-Bridge circuit to both motors. We plan to incorporate the microcontroller, the infrared sensor, the power switches and the boost converter on a PCB. The robotic platform(vehicle) consists of a plastic hook, a servo motor, two regular motors, two small tires and one chassis.

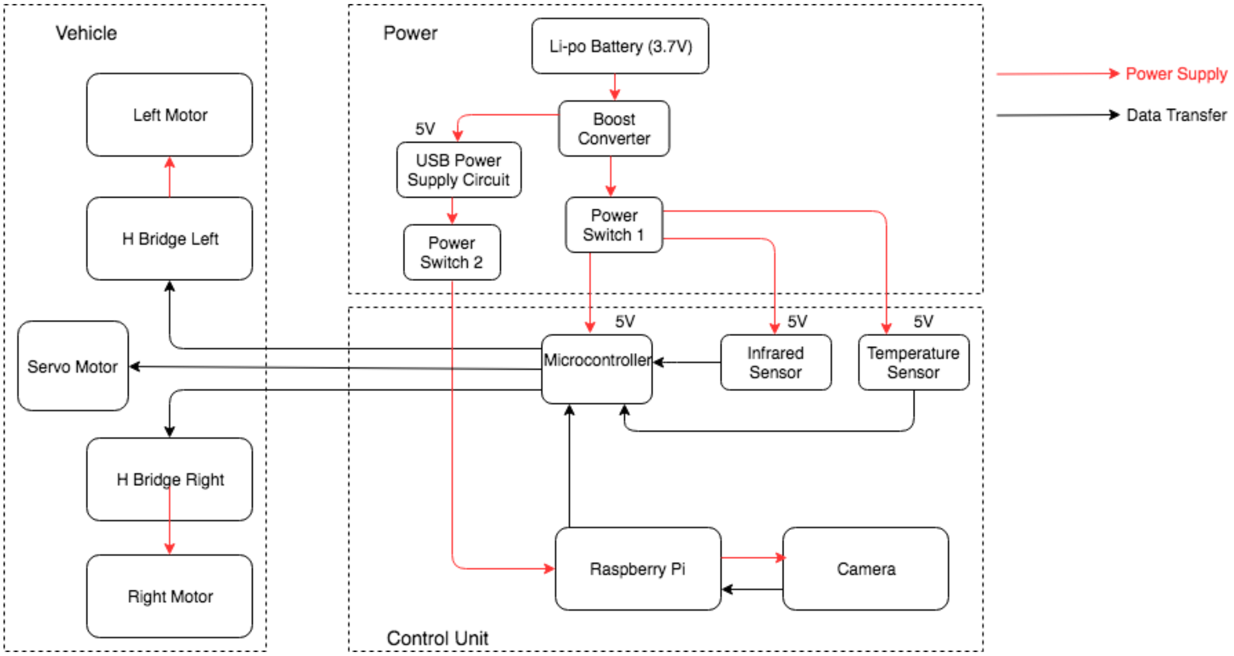

Fig. 1. Block Diagram

2.2 Physical Design

The physical design of our project is mainly based on the sparkfun vehicle[3]. The vehicle will be approximately $20cm$ in length, $13cm$ in width and $5cm$ in height. The vehicle will be equipped with a battery, a PCB board and a Raspberry Pi board, all of which will be placed on the chassis. Two infrared sensors will also be attached to the bottom of the vehicle, one in the front and the other in the back. The Raspberry Pi camera will be placed at the top chassis of the vehicle. We plan to glue it to the front center. ITo enable the vehicle to physically bring the butter back, we plan to add two small plastic hooks, which can rotate with help from the servo motor. The servo motor we plan to use is the Micro Servo from Sparkfun[26]. According to the datasheet, it can rotate around 160°, which satisfies our needs.

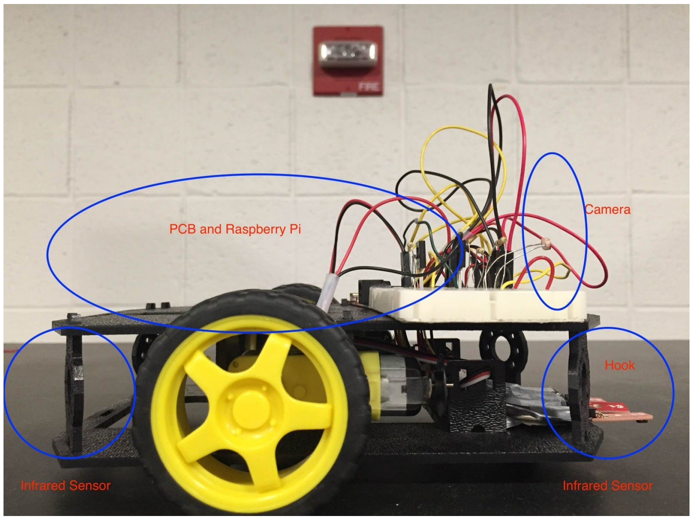

Fig. 2. Physical Design Sketch

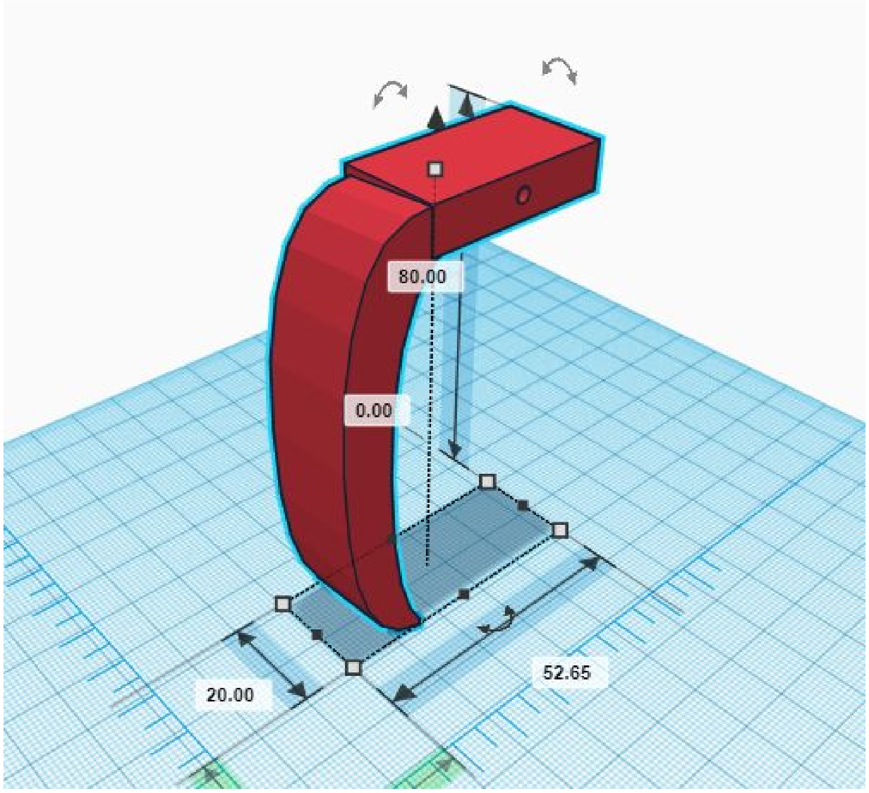

Fig. 3. Plastic Hook Sketch (Dimension all in mm)

2.3 Power Supply

The power supply module provide power to all the hardware components. It consists of two lithium batteries, a boost converter, two power switches and a USB power-supply circuit for the Raspberry Pi board.

2.3.1 Lithium Battery

To ensure that our robot can work independently without any extra power chord, we plan to use two lithium batteries to supply power. Each lithium battery[4] can supply 3.7V and has a capacity of $3400\ mAh$. The batteries will be connected in parallel to avoid saturation. They will hold total power of $3400\ mAh * 2 = 6800\ mAh$. They will be able to keep the whole system running for about $6800\ mAh / 2225\ mA \approx 3.05\ h$.

2.3.2 Boost Converter

Since the lithium battery will discharge over time, and all the hardware components operate under voltage ranges different from the battery voltage, we will need a boost converter to supply a stable voltage input to the remaining parts. And according to the datasheet, the Raspberry Pi and infrared sensor operate under 5V, the microcontroller operates within 1.8V ~ 5V and the temperature sensor operates within 2.7V ~ 5.5V. Thus, a 5V output from the boost converter should be sufficient for the whole system.

Requirement	Verification
Given input in the range of 2.5V to 3.7V, the boost converter must provide constant 5V output to satisfy the needs of all components.	We plan to apply different voltages (2.5V ~ 3.7V) to the boost converter using function generator in lab. And we will use the multimeter to measure the output voltage of the regulator, which should always be 5V.

Table. 1. Boost Converter R&V[5]

2.3.3 Power Switch

To make sure that the current going into all the components will not damage those components, we plan to place two power switches in the power module. The power switch we plan to use is the TPS2020 Power Distribution Switch[6]. This power switch is able to limit the output current when the load exceeds the threshold. The switch will be shut if a short circuit caused high power dissipation and increased the temperature.

Requirement	Verification
The power switch is able to limit the output current even if the input current is exceeding the threshold.	We plan to connect the power switch with resistors on the breadboard. And we will use function generator to apply different voltages across the resistor. We will use a multimeter to measure the output current from the switch, which should be below the threshold.

Table. 2. Power Switch R&V

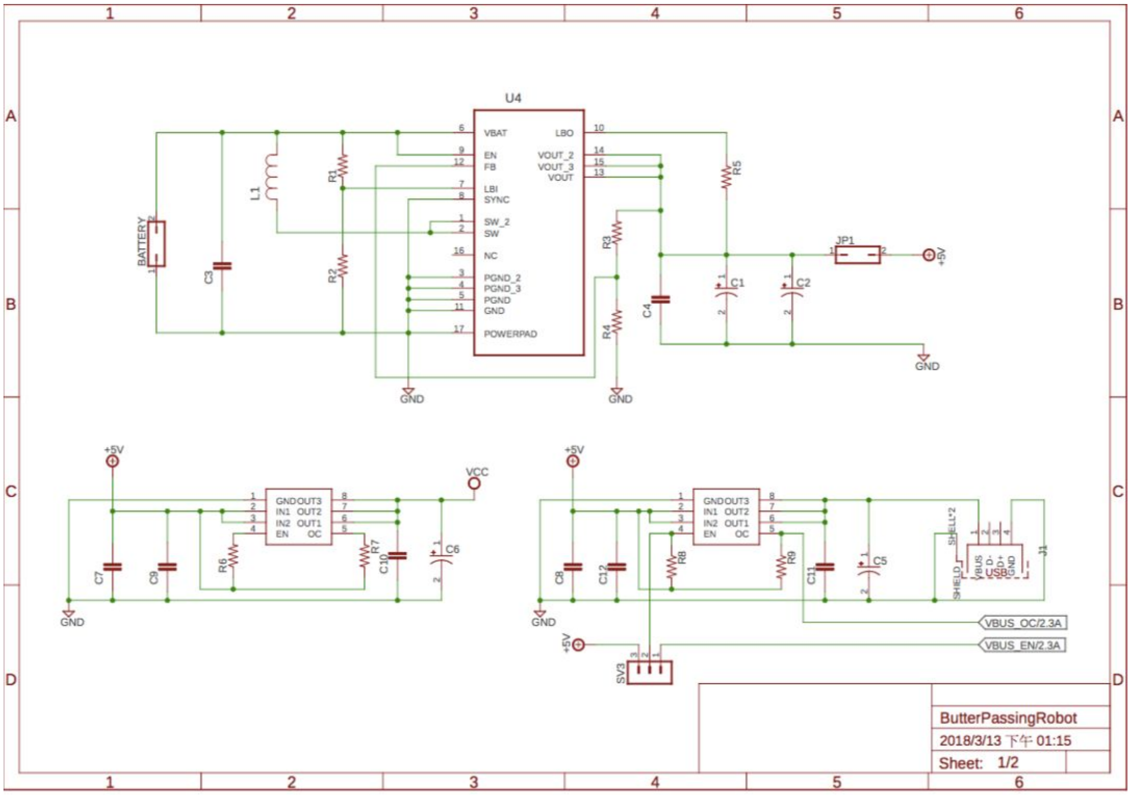

Fig. 4. Schematic for Power Module

2.4 Vehicle

The vehicle is the “body” of our project. It will perform all the physical work: moving toward the butter and bringing the butter back. It receives instructions and power from the control module.

2.4.1 Motor Module

We plan to use H-Bridge circuits to drive our motors. The advantage of a H-Bridge circuit is that it can drive the motor in both directions. This will give us the ability to rotate the vehicle about its center and to drive the vehicle backwards. Below is an example diagram using H-Bridge circuit with a L293 motor driver.

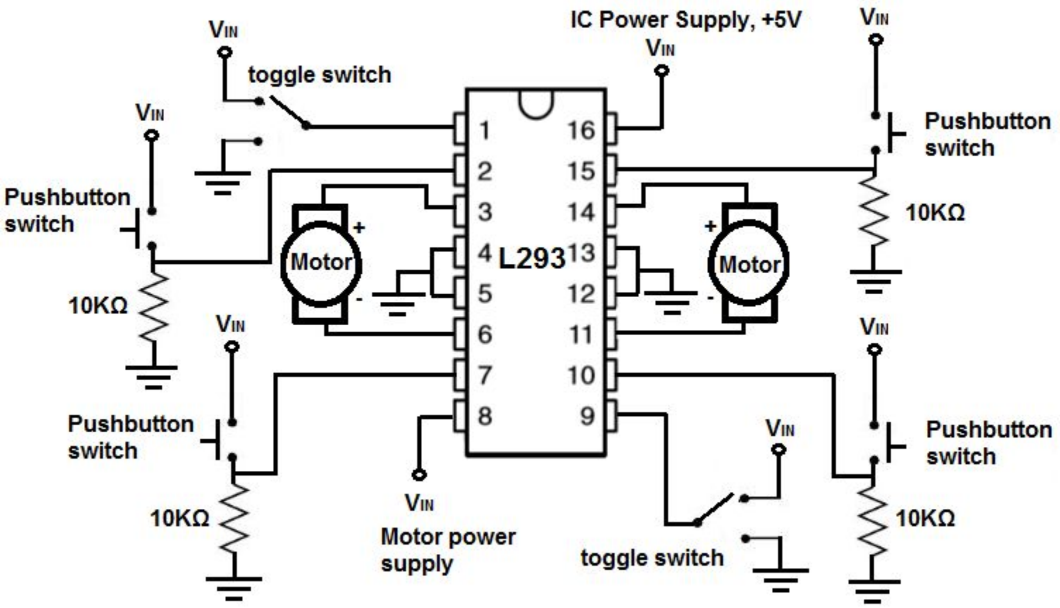

Fig. 5. H-Bridge Example[6]

Enable	**Logic Pin 1**	**Logic Pin 2**	**Result**
High	Low	High	Forward
High	High	Low	Reverse
High	Low	Low	Stop
High	High	High	Stop
Low	/	/	Off

Table. 3. Modes in H-Bridge Circuit

The enable pins control the motors. If they are connected to ground, then neither of the motors can be operated. Through the H-bridge circuit, if the logic pin 2 of either motor is high, then the corresponding motor will spin in a forward direction. Similarly, if the logic pin 1 of either motor is high, then the corresponding motor will spin in a reverse direction. Once two logic pins are both low or high, then the motor will shut off. This is how we control the forward and reverse movement of motors.

Requirement	Verification
The vehicle can move straight forward and backward. The angle offset should be within ±10°.	We plan to write a test program. it will apply a constant voltage V to both motors for 10 seconds. We will draw the path on the table and measure the angle offset using a protractor.

The vehicle can rotate about its center. The center should only shift within 2 cm every time the vehicle rotates 60°.	We will write a test program and load that to the microcontroller. It will perform a rotation of the vehicle every 10 seconds. And we measure the center shifting using a ruler.

Table. 4. Motor Module R&V

2.5 Control Module

The control module is the “brain” of our project. It is powered by the power supply module and it outputs signal to the vehicle module.

2.5.1 Microcontroller

We plan to use an ATmega328p microprocessor manufactured by Microchip Technology for this project. It will receive voltage readings from the Raspberry Pi, the infrared sensor, the temperature sensor, and the current-limiting resistor. Inputs from the first two parts contribute to the functionality of our project, while inputs from the last two components provide safety precautions for our project. By our estimation, we need 11 pins to connect to all the devices. According to the datasheet[7] of ATmega328p, this microcontroller as 22 general-purpose I/O pins, which is sufficient for our project.

Requirement	Verification
The microcontroller must stop driving the motor when the reading from the infrared sensor is lower than 4V.	We plan to simulate the infrared sensor input using a function generator and use a multimeter to measure the microcontroller’s output. When the input voltage is less than 4V, the reading on the multimeter should be 0V.
The microcontroller must stop driving the motor when the reading from the temperature sensor is above 1V.	We plan to simulate the temperature sensor input using a function generator and use a multimeter to measure the microcontroller’s output. When the input voltage is greater than 1V, the reading on the multimeter should be 0V.

Table. 5. Microcontroller R&V

Type	Pin Number	Pin Name	Description
Digital Output	2	PD0	connected to H-Bridge Pin 2 to control left motor
Digital Output	3	PD1	connected to H-Bridge Pin 7 to control left motor
Power	7	VCC	connected to voltage regulator
Ground	8	GND	connected to ground
Analog Input	12	PD6	connected to Infrared sensor output
Analog Input	13	PD7	connected to Infrared sensor output
Digital Output	17	PB3	connected to H-Bridge Pin 10 to control right motor

Digital Output	18	PB4	connected to H-Bridge Pin 15 to control right motor
Power	20	AVCC	connected to voltage regulator
Analog Input	28	PC5	connected to temperature sensor output

Table. 6. Microcontroller Pin Layout

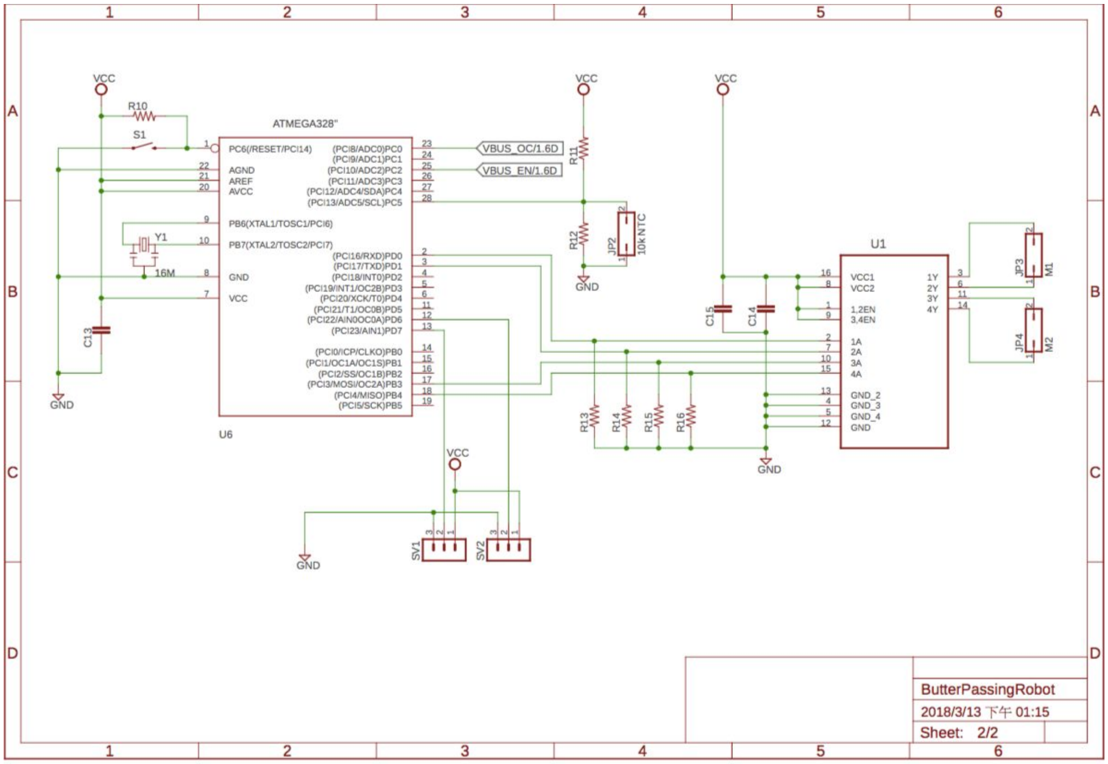

Fig. 6. Schematic of Microcontroller and H-Bridge Chip

2.5.2 Object Detection Module with Raspberry Pi and Camera

We plan to use a Raspberry Pi 3B motherboard and its corresponding v2 camera module for this project. Those two components are crucial since they are in charge of providing our robot accurate information about the butter. The Raspberry Pi is powered by the power supply unit, through the USB power supply circuit. The v2 camera module will be directly powered by the Raspberry Pi board. The v2 camera module will be placed at the front of the vehicle so that it can take photos of the environment the vehicle is facing.

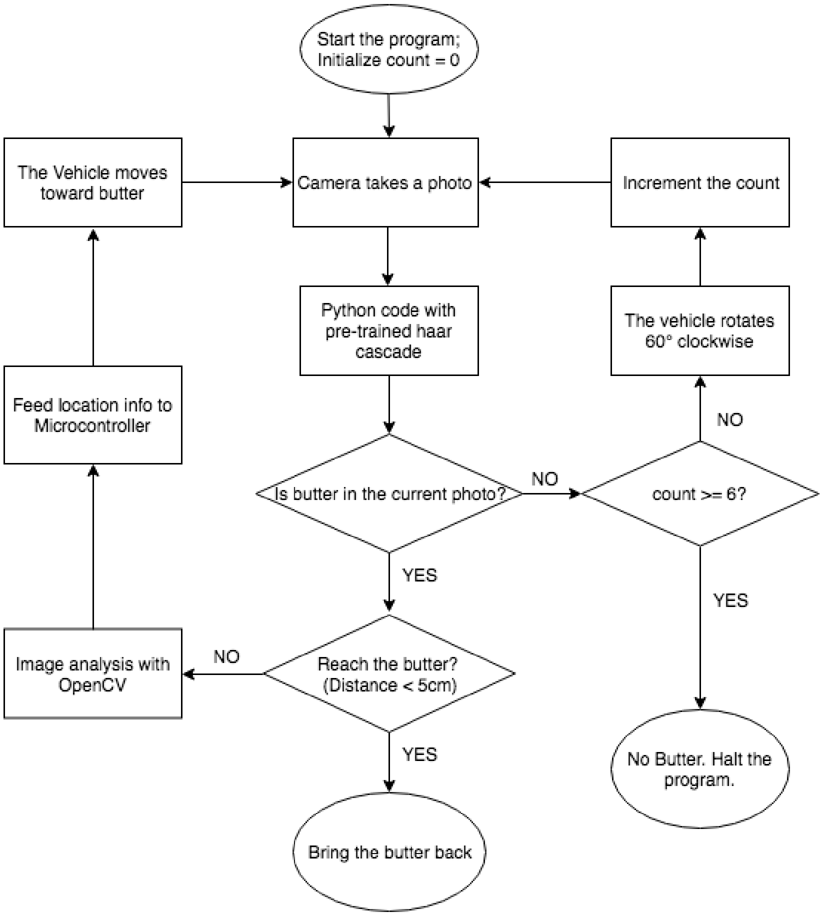
Start the program; Initialize count = 0
Camera takes a photo
The Vehicle moves toward butter
Increment the count
Python code with pre-trained haar cascade
The vehicle rotates 60° clockwise
Feed location info to Microcontroller
NO
Is butter in the current photo?
NO
count >= 6?
YES
YES
Image analysis with OpenCV
NO
Reach the butter? (Distance < 5cm)
YES
No Butter. Halt the program.
Bring the butter back

Fig. 7. Control Module Flowchart

We will first train an object detection program using either Harr Cascade Classifier or Convolutional Neural Network. There are several frameworks and libraries available to us, e.g. OpenCV, TensorFlow, LeNet… We plan to test with each of them and find the most suitable one based on the requirements listed in the following table.

Requirement	Verification
The program can detect yellow, cubed butter in the size smaller than 12cm * 3cm * 3cm[10]. The accuracy rate should be 80% at least.	We will write a test program asking the detector to run with at least 500 positive images of cubed butter (either automatically created by classifier or manually taken by us). We will re-train our detector until the 80% accuracy is met.
The program can distinguish butter from other kitchen objects of similar color: orange juice, honey mustard. The accuracy rate should be 80% at least.	We will write a test program asking the detector to run with at least 500 negative images of orange juice or honey mustard. We will re-train our detector until the 80% accuracy is met.

The program can finish processing an image in 0.5 s, so that the motor can always act in time	We will import the datetime library in python and time the our detection program on Raspberry Pi.
The programs should output an angle between the vehicle and the target. And this angle should be within ±10° from the actual angle.	We will place the butter and Raspberry Pi camera both on the table. Then we will compare the actual angle measured by a protractor with the angle calculated by our detector.

Table. 7. Raspberry Pi and Camera R&V

Since we know the field of view of our camera[11], we can derive the angle using the pixel index. This method is not perfect geometrically. However, it will still be useful in providing an approximation since our vehicle will be constantly moving and adjusting the angle.

Figure 5 is a top view of our vehicle's working environment. The value our program should calculate and output is ∠ACB. The known value is ∠ECF representing the field of view of our camera. Figure 6 is a camera view of our vehicle's working environment. The rectangle represents the target object detected by our program. (x, y) represents the starting pixel of the target rectangle and (w, h) represents the dimension of the target rectangle. And (x_c, y_c) represents the center pixel of the image. This pair can be calculated given the resolution of the image:

$$x_c = \frac{w_{image}}{2}, y_c = \frac{h_{image}}{2}$$

Equation. 1. Image Dimension

The equation we plan to use is the following:

$$\angle ACB = \frac{(x + \frac{w}{2}) - x_c}{W_{image}} \cdot \angle ECF$$

Equation. 2. Angel Calculation

A positive output value means the vehicle should turn right, while a negative value means the vehicle should turn left.

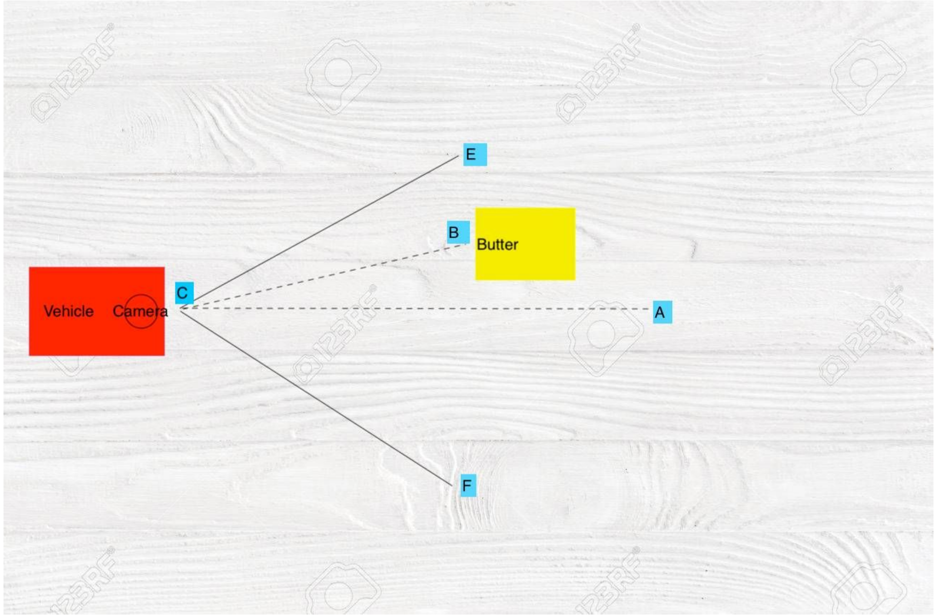

Fig. 8. Top View of the Table

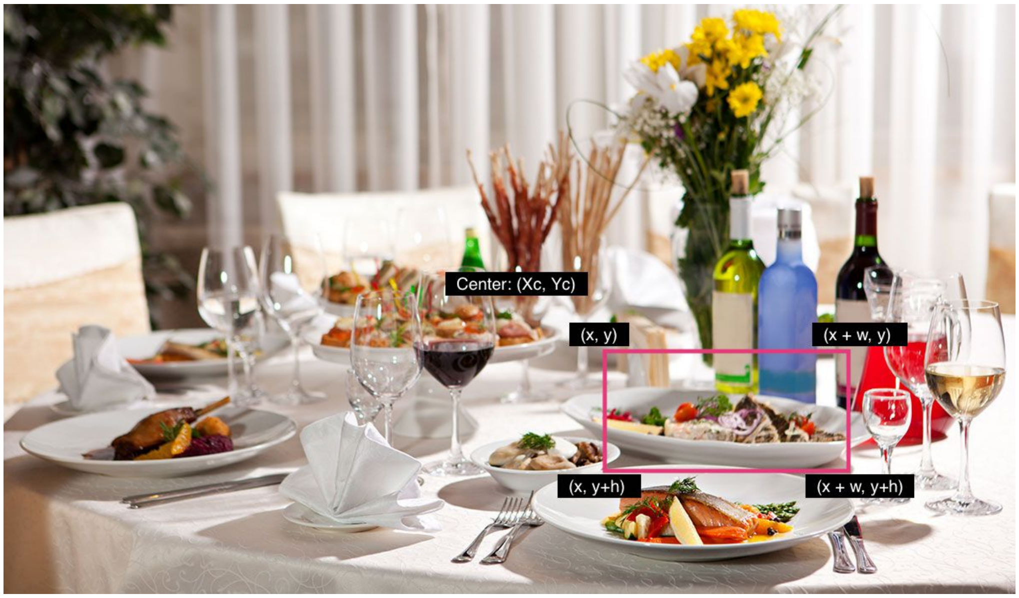

Fig. 9. Camera View of the Table

2.5.3 Infrared Sensor

We plan to install two infrared sensors on our vehicle, one in the front, another in the back. They are used to make sure our vehicle never falls off the table. It can also be used to detect whether the vehicle reaches the plate containing butter, since the color and texture of the plate will be different from that of the table. When the infrared sensor outputs values below a threshold, the microcontroller should stop stop the motor. The infrared sensors are powered by the power module and their OUT pin will be connected to the microcontroller.

Requirement	Verification
The infrared sensor should be able to distinguish gray floor, which is about 80 cm away, from the white table, which is about 1 cm away.	According to previous work with this sensor, it will output around 4.3 V over white surface, and 4.85 V over black surface. The difference is quite distinguishable. [13] We plan to place this sensor near objects of different colors, and use a multimeter in lab to record the voltage reading.
The infrared sensor should be able to distinguish white paper plate with decorating patterns from the white table.	We plan to place this sensor near the table and the white paper plate, and use a multimeter in lab to record the voltage reading. The voltage readings should be different.

Table. 8. Infrared Sensor R&V

2.5.4 Temperature Sensor

We include a temperature sensor in our project for safety precaution. Since we are using a Lithium battery to power the whole system, we need to make sure the temperature of the battery is within a safe range. According to experience of previous groups, that range should be from 0°c to 40°c.[14] The temperature sensor will be powered by the power supply module and its output will be used by the microcontroller. According to the datasheet, the temperature sensor has a detecting range of -5°c to 125°c, with ±2°c accuracy. And the temperature sensor has a scale factor of 10 mV/°C. Assume the temperature of the battery raises from 20°c (room temperature) to 40°c (dangerous), then the voltage reading will increase by 0.2V, which is detectable by our microcontroller.

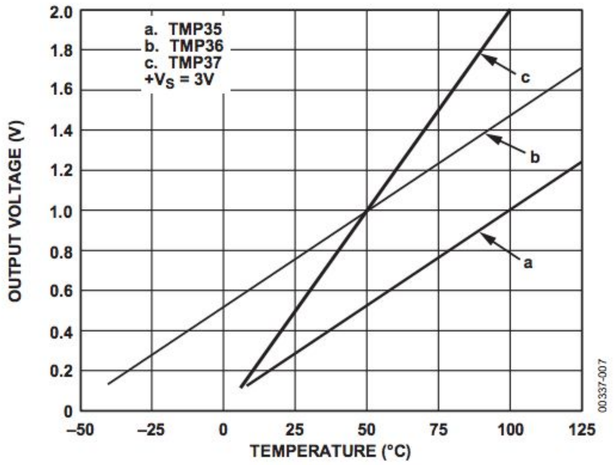

Fig. 10. Temperature Sensor Characteristics[15]

2.6 Tolerance Analysis

The most important requirement for our project is the running time of the object detection program. If the software program has a time complexity or space complexity that's too large, the Raspberry Pi will spend more time processing each image. If the vehicle cannot receive up-to-date information about the butter's location, it will just continue in its old path and end up somewhere further away.

We did some research online and found a project where a Raspberry Pi with its camera was used for face recognition.[17] From this video, it takes around 4 seconds for the Raspberry Pi to finish processing an image taken by the v2 camera module. However, the Raspberry Pi used in this video is an older model. According to another article we found online[18], the model we use (Raspberry Pi 3B) has a benchmark score that's 50% better than the older model. So our estimation is that the same face detection will probably take around 2.5 seconds to finish on our Raspberry Pi. And we believe we can reduce this time further by reducing image size and feeding more manually created, specific-to-our-environment positive images. We ran the object detection program on 10 images and timed the process. As can be seen in the table below, object detection program can finish much faster on a smaller image. However, we also have to take into account the accuracy of object detection. Given a fixed detection window size, detecting objects in a smaller image will generally produce a lower false positive rate since there are fewer potential windows to check.

Image Size	Number of Images	Running Time
820 * 616	10	34.10s
410 * 308	10	2.45s
205*154	10	0.65s

Table. 9. Detection Time vs Image Size

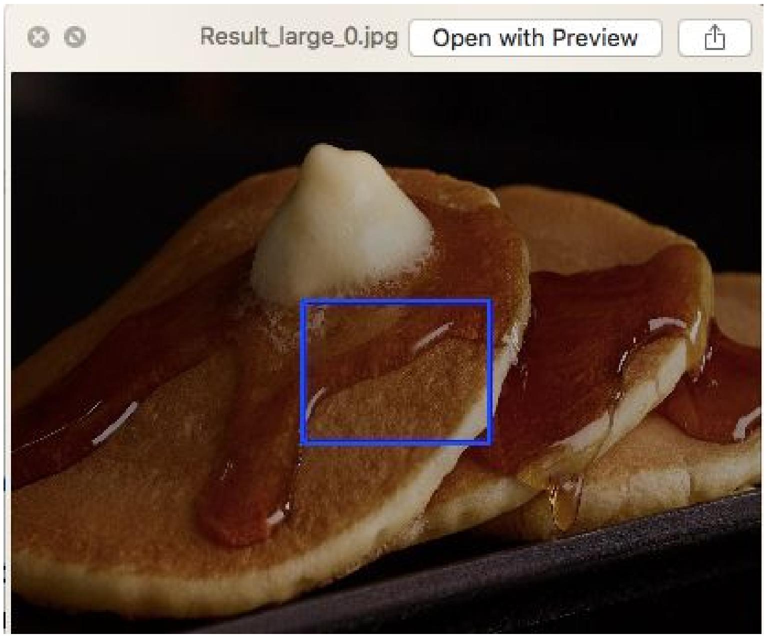

Fig. 11. 410*308 Image without False Positive Detection

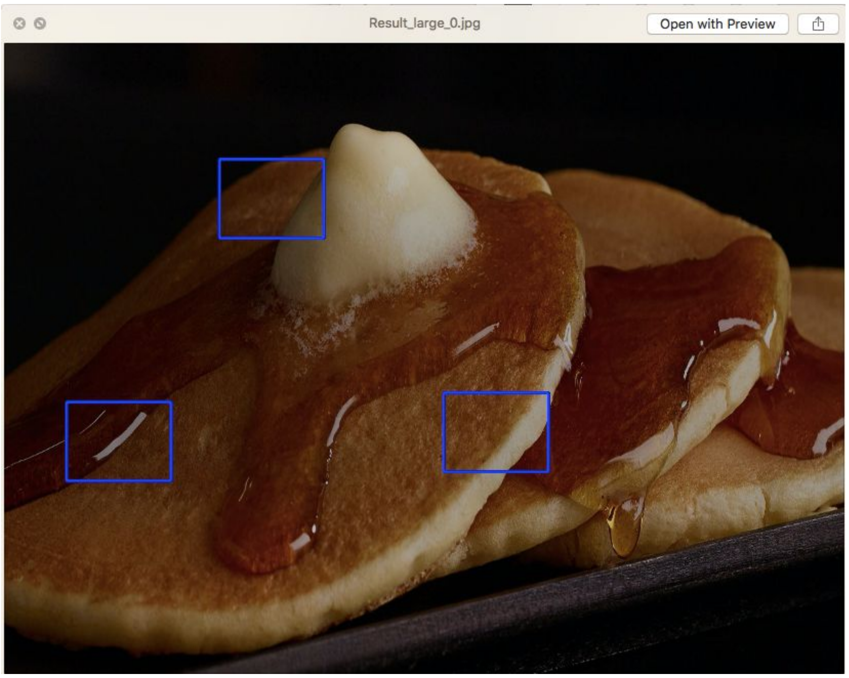

Fig.12. 820*616 Image with False Positive Detection

Also, our Raspberry Pi has a Wi-Fi module. According to this test[19], the Wi-Fi speed can reach up to 35 Mbits/s. The file size of an image taken by Raspberry Pi camera is around 2.5 MB under full quality. Thus, the Raspberry is capable of transmitting a full-size, full quality image in 0.6 seconds. ($2.5\ MB / ((35\ Mbits/s)/8) = 0.6s$) We can further reduce this by shrinking the image. Given the strong connectivity and fast image-transmitting capability, it's also possible for us to run the object detection program on a cloud server (e.g. AWS Linux instance) and send the result back.

With all these options available to us, we plan to first try different object detection frameworks and compare their time complexity. Then we will pick the fastest one and transfer that trained classifier/CNN to our Raspberry Pi. Then we will time the object detection process on Raspberry Pi and decide whether to utilize other computing resources.

3. Cost and Schedule

3.1 Cost Analysis

The total cost of this project would be the sum of our labor cost and the cost to purchase hardware parts.

3.1.1 Labor

According to Engineering Career Services at UIUC[20], the average starting salary for Engineering graduates from UIUC is $71,856. Dividing this number by working hours per year: 40*52 = 2080, we get the average hourly rate: $35. However, since we all haven't graduated yet, it would be more realistic to estimate our hourly rate to $30. Given the design of our project, we assume each of us will work 10 hours per week for 13 weeks. Thus, the total labor cost of this project would be:

$$2.5 \ * \ \$30/hr/person \ * \ 3 \ persons * \ 10 \ hrs/week \ * \ 13 \ weeks \ = \ \$29250$$

3.1.2 Parts

To successfully implement this project, we need to purchase some hardware parts. The following table gives detailed information about the cost of each component. We also include the spare parts we need for development and testing here.

Part	Manufacturer	Retail Cost ($)	Count	Total Cost
Raspberry Pi 3B Motherboard	Raspberry Pi Foundation	35.91	1	35.91
Raspberry Pi Camera Module V2	Raspberry Pi Foundation	26.45	1	62.36
ATMEGA328P	Microchip Technology	2.20	2	66.76
Sparkfun QRE1113 Infrared Sensor[21]	Sparkfun	2.95	4	78.56
Li Battery	Panasonic	15.99	2	110.54
Shadow Chassis[22]	Sparkfun	12.95	1	123.49
Motor[23]	Sparkfun	1.95	2	127.39
TMP36 Temperature Sensor	Sparkfun	1.50	1	128.89
L7805 Voltage Regulator	STMicroelectronics	0.95	5	133.64
SN754410 H-Bridge Chip	Texas Instruments	2.35	2	138.34

Table. 10. Cost of Hardware Parts

3.1.3 Grand Total

The total cost of our project would be:

$$\$29250 + \$138.34 \ = \$29388.34$$

3.2 Schedule

Week	Objectives	Yuchen	Yuxiang	Yu Jie
02/19	Prepare for Mock Design Review; Finish Design Document	1.) Write the cost, schedule and block diagram section 2.) Requirement and Verification	1.) Write the objective, background and high-level requirement 2.) Requirement and Verification	1.) Write the functional overview and ethics section 2.) Requirement and Verification
02/26	Research different platforms for object detection; Prepare for Design Review	Train a classifier for butter detection and test the computing speed	Research the H-bridge circuit	Research different hardware parts, design the circuit protection scheme
03/05	Finalize PCB design; Test the parts on breadboard; Get familiar with soldering	Train another object detection scheme using CNN, time the processing	Finalize PCB design. Test different parts on breadboard.	Finalize PCB design. Experiment with different sensors output range.
03/12	Finalize on the object detection method; Transfer the desired one onto Raspberry Pi; Test the first PCB board	Compare different software and transfer the most suitable program onto Raspberry Pi	Test the circuit on the PCB board, check against schematic and simulation	Test the functionality of different sensors on board
03/19	Integrate the Raspberry Pi with microcontroller	Test software program output with manually fed images simulating the demo environment	Test signals going into microcontroller and adjust the input to H-bridge circuit	Test the functionality of the vehicle: rotation, movement
03/26	Check the functionality of the vehicle	Improvise on software;	Find issues with prototype, revise and may re-submit for final PCB design	Test all power supply, revise and may re-submit for final PCB design
04/02	Assemble all different parts together	Finish software code. Debug code to control prototype precisely	Work on physical installment of the project	Work on assembling final PCB and ensuring that power is delivered to PCB while it operates and spins
04/09	Debug	Final debugging and ensure that everything is operating correctly	Final debugging and ensure that everything is operating correctly	Final debugging and ensure that everything is operating correctly

04/16	Prepare for Mock Presentation	Mock demo preparation.	Prepare for mock demo and help with communications issues from the motors	Prepare for mock demo and help with communications issues with the controller
04/23	Finish up on final report and summarize the work	Work on presentation and paper	Work on presentation and paper	Work on presentation and paper
04/30	Prepare for Final Presentation	Finish up final papers	Finish up final papers	Finish up final papers

Table. 11. Schedule

4. Ethics and Safety

The major safety concern within our project is the use of a lithium battery. Risks include the thermal stability of active materials within the battery at high temperatures, fires and even explosions. Failure in performance can be caused by short-circuiting, poor execution of a design, or an unanticipated use or abuse of a project. To prevent any of those risks, we need to check our circuit design before actually implementing it. Also, we should always place our battery in a safe position in order to prevent it from violent collisions, which can damage the separator and cause the electrodes to touch. If the battery be pierced(either by accident or deliberately), then short circuit will happen. Based on data and experience from previous projects[24], we will also implement some circuit to constantly monitor the temperature of our lithium battery. If the temperature exceeds an optimal range, we will program the microcontroller to shut down the system.

Another safety concern of our project is current overflow. This can be caused by design flaw, mechanical issue like motor overheat or getting stuck. Current overflow can do damage to all the devices we have on the PCB board and can also potentially damage our Raspberry Pi. In order to prevent this from happening, we will add some current limiting resistor to limit the current. We will also feed the voltage reading across those resistors back into the Microcontroller. The microcontroller should be programmed to halt the whole system if the voltage exceeds a pre-calculated threshold.

Working in the ECE senior design lab with other groups also has safety risks. To prepare us for that, we all completed the lab training program and obtained the certificate. We will always keep safety precautions in mind and not impose danger to other students or ourselves.

We are responsible for the outcome and information of our project, following every rules in the IEEE code of ethics [25]. Specifically, considering the 7th rule of IEEE code of ethics, since we will require large amount of knowledge for improving and accomplishing this project, variable experiences and technical information related to will be needed from the Internet or books. We will pay respect and be grateful to others for their efforts; furthermore, plagiarism will not happen. Besides, the 9th rule should also be watched out. In that our robot may be touched directly, we will take care designing the physical part in case of current leaking which can cause injury to human.

Reference

[1] *Rick and Morty - You pass Butter*, Youtube. Available at: https://www.youtube.com/watch?v=X7HmltUWXgs

[2] *Rick and Morty butter passing robot*, Youtube. Available at: https://www.youtube.com/watch?v=7jGcqPJm5vw https://www.youtube.com/watch?v=TZbxWvc9O6M

[3] *Sparkfun RedBot Basic Kit*. Available at: https://www.sparkfun.com/products/retired/13166

[4] Panasonic NCR18650B Super Max Rechargeable Li-ion Battery, Amazon. Available at: https://www.amazon.com/Panasonic-NCR18650B-Rechargeable-Battery-3400mAh/dp/B00DHXY72O/ref=sr_1_1?ie=UTF8&qid=1519274078&sr=8-1&keywords=panasonic+li+battery

[5] *5-V Output, 1-A, 96% Efficient Boost Converter*, Texas Instruments. Available at: http://www.ti.com/product/TPS61032

[6] TPS2020 Single Power Distribution Switch, Texas Instruments. Available at: http://www.ti.com/product/TPS2020

[7] *H-Bridge Circuit,* Wikipedia. Available at: https://en.wikipedia.org/wiki/H_bridge

[8] *ATmega328P Complete Data Sheet.* Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-42735-8-bit-AVR-Microcontroller-ATmega328-328P_Datasheet.pdf

[9] *Raspberry Pi Camera Module V2, Sparkfun.* Available at: https://www.sparkfun.com/products/14028

[10] *Convolutional Neural Networks vs. Cascade Classifiers for Object Detection.* Available at: https://dzone.com/articles/cnn-vs-cascade-classifiers-for-object-detection

[11] *Size and Shape of Packaging, Butter, Wikipedia.* Available at: https://en.wikipedia.org/wiki/Butter

[11] *Field of View of Raspberry Pi V2 Camera, Raspberry Pi Forum.* Available at: https://www.raspberrypi.org/forums/viewtopic.php?f=43&t=154155

[12] *OpenCV Classifier Documentation*. Available at: https://docs.opencv.org/2.4.13.5/modules/objdetect/doc/cascade_classification.html#cascadeclassifier-cascadeclassifier

[13] *Final Project Report: LineFollowing, Lightsensitive Autonomous Vehicle*, Robert Altman and Yuchen He

[14] *Design Document*, ECE 445 Course Website. Available at: https://courses.engr.illinois.edu/ece445/guidelines/design-document.asp

[15] Temperature Sensor - TMP36, Sparkfun. Available at: https://www.sparkfun.com/products/10988

[16] *LED Current Limiting Resistors*, Sparkfun. Available at: https://www.sparkfun.com/tutorials/219

[17] *OpenCV Face Detection with Raspberry Pi - Robotics with Python p.7,* sentdex. Available at: https://www.youtube.com/watch?v=1I4gHpctXbU&t=965s

[18] *Raspberry Pi 3 vs Pi 2: What's the difference?* TrustedReviews. Available at: http://www.trustedreviews.com/opinion/raspberry-pi-3-vs-pi-2-2936374

[19] *Raspberry Pi network speed test: RPI2, RPI3, Zero, ZeroW (LAN&WiFi)*, Not Enough TECH. Available at: http://www.notenoughtech.com/raspberry-pi/raspberry-pi-internet-speed/

[20] *"Student Success",* Engineering Career Services. Available at: http://ecs.engineering.illinois.edu/outcomes/

[21] *SparkFun RedBot Sensor - Line Follower,* Sparkfun. Available at: https://www.sparkfun.com/products/11769

[22] *Shadow Chassis,* Sparkfun. Available at: https://www.sparkfun.com/products/13301

[23] *Hobby Motor - Gear*, Sparkfun. Available at: https://www.sparkfun.com/products/11696

[24] *Projects, Spring 2017,* ECE 445 Course Website. Available at: https://courses.engr.illinois.edu/ece445/projects.asp

[25] *IEEE Code of Ethics*. Available at: https://www.ieee.org/about/corporate/governance/p7-8.html

[26] *Servo - Generic (Sub-Micro Size), Sparkfun*. Available at: https://www.sparkfun.com/products/9065