Robotic waiter for Restaurants

ECE 445 Design Document

Project #26
Cheng Jin
Jun Pun Wong
Kausik Venkat
TA: Xinrui Zhu

Contents

1. Introduction

1.1 Objective

Today, the shortage of manpower has become a huge conundrum for business owners, especially in the food and beverage industry. Even with the deployment of good chefs and experienced managers, restaurants tend to run into a chaos with insufficient servers. Therefore, a quick solution may be to have servers carrying heavier loads to increase efficiency. However, this solution quickly worsen as servers become fatigue and hence, unable to continue working.

Due to the fact that the main duties of servers are to fetch dishes from the kitchen to the customers, we propose to utilize an army of robots for this task. With the help of robots, the problem can be eliminated as robots can continue working tirelessly. Also, as servers are required to be trained, part-time workers tend to contribute to a wastage of resource. Furthermore as part-time workers have irregular schedules, restaurant owners may find themselves having less staff at a critical times. Consequently, to cope with a sudden surge of customers during festive seasons or the weekends, more robots can be deployed instead of having to hire more temporary staffs.

Ideally, we want to work towards a system that will benefit many people. It would be an automatic computer-controlled communication system between the various stakeholders - restaurant staff, patrons and the robotic waiters. There will be only need to a be a couple of stuff to assist the robots who would be able to take orders and deliver food, all on their own. But due to time constraints, we are restricted by what we can achieve in this course. We will only work on building a robot that can deliver food. This is because the software required to handle ordering is not very easy and software isn't a high priority in this class. We will also only implement one robot because the cost of the parts is quite high. And since we are only building one robot, most of the software will not take into account certain scenarios that would occur if multiple robots are present. Instead we assume that the various parameters and the location of the table is known ahead of time. Thus, when the robot is told to go to a table, the software is able to tell the coordinates of the table and perform a path-finding algorithm to look for an optimized route. We hope that this version is a first draft in an innovative product that transforms the dining industry. The robot will only deliver to one table at a time. In other words, it will only receive the subsequent orders when it stops in the kitchen.

1.2 Background

The research to optimize restaurant operations has been extensive. Therefore, robots have been used and is currently used in the market to cut cost and increase efficiency. In the China's northeastern Zhejiang province, a restaurant has deployed robotic waiters, each costs more than $9,400. [16]With the region's minimum monthly wage of $300, it may require some time for the restaurant to come to a breakeven. However, in long term, the restaurant may find itself saving money from the employment of less waiters.

In contrast, business owners in the USA may require less time than in China to come to a breakeven. This may be due to the much higher minimum wage. For example, with the D.C's minimum wage of $11.50, this translates to approximately $1,000 per month for every server a restaurant employs. Therefore, within a year, restaurant owners are guaranteed to save money with the assumption that no maintenance are required for the robot in that duration.

With the account of current robots costing at least $9000, we propose to reduce the costs of robots. In real applications, since the cost of our robots are much lower, it will be relatively easy to increase the number of working robots per unit area.

1.3 High-level requirements

- Send information from the kitchen's MCU to the robot's MCU and vice versa.
- Given a destination, the robot is able to calculate and navigate a path to its final destination.
- Kitchen's MCU is able to display the list of tables our robot is serving to on the LED display.

2. Design and Requirements

2.1 Design Block Diagram

The ecosystem for the robotic waiter requires 4 modules as shown in Figure 1. The robot itself will have a navigation module that will control the motors and calculate the path for the robot to take. It will be powered by its own power distribution unit which will power it with the proper 27V. And the current of motor has to be critically 1.6-1.8 A. The kitchen will also have its own central control unit; MCU 1. All requests from patrons would be received this command module and it will relay information to the robot. It is also powered by its own power distribution unit which will power it with 3V.

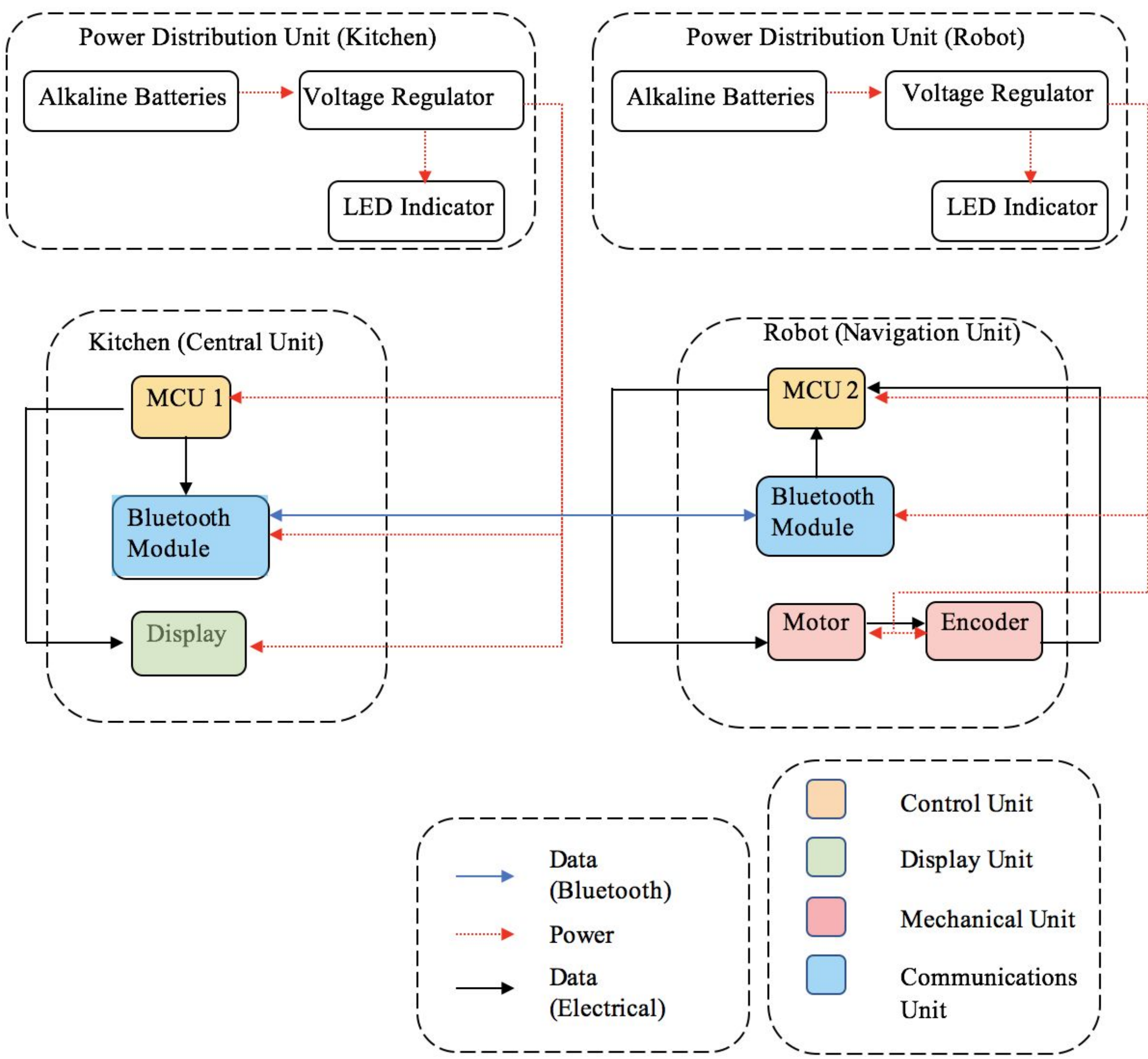

Figure 1: Block Diagram

2.2 Physical Design

This is the side view of our robot. It will be supported by 2 main driver wheels that are connected to stepper motors. There are six caster wheels. Those will help support the robot's weight and stabilize the robot. By using stepper motors, we are able to precisely determine the distance the robot will travel.

The base will be used to house our electronics like the MCU and the bluetooth receiver/transmitter. The load tray will be used to carry food to the patrons in the restaurant.

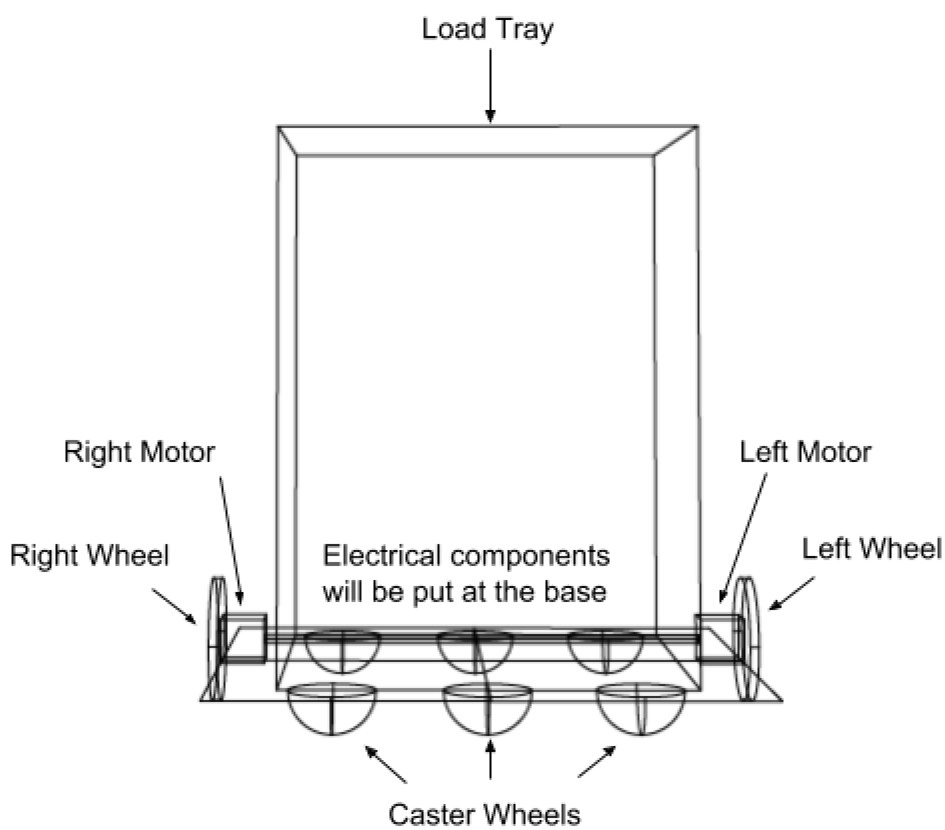

Figure 2: Side of entire robot

These is front side view of the robot. This diagram will help illustrate the dimensions of our robot.

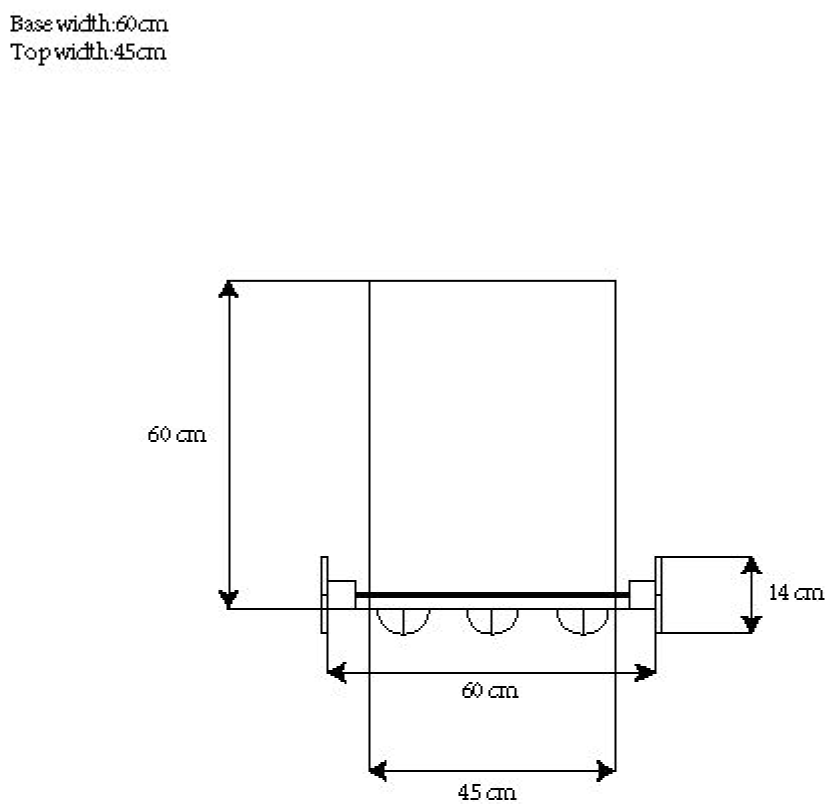

Figure 3: Detailed diagram of side robot

2.3 Power Distribution Unit

The power distribution unit will be responsible to power up the devices for any time it is needed.

2.3.1 Alkaline Battery

3 9V 560 mAh batteries will be used as the main power source for the robot. Alkaline batteries are used because they do not require any special disposal method and they are among the cheapest option[4]. However, the downside of alkaline batteries are that they have very high internal resistance and thus, reduces their runtime duration. Therefore, we might have to use multiple batteries. Besides that, misuse of power from the battery may cause damage or failure to our components and thus, protective circuits are required to reduce the probability of such events.

For the kitchen's power system, one 9V 560 mAh battery will be used. This is because the dropout voltage of the voltage regulator is 2V. Henceforth, a minimum input voltage for a guarantee of 5V output has to be at least 7V.

2.3.2 Voltage Regulator

The voltage regulator circuit supplies stable DC voltage to other units. For the voltage regulator circuit on the robot in Figure 5, the 5V voltage regulator and the 24V voltage regulator will take in 27V from the three 9V batteries. The 5V voltage regulators provide ±5% of 3V to power the bluetooth transmitter and receiver.

Requirements	Verification
A) For the kitchen unit, the 5V voltage regulator must provide 5V(±5%) given a 9V(±10%) input. The 3.3V voltage regulator has to provide 3.3V(±5%) from an input voltage of 5V(±5%). B) For the robot unit, the 24V voltage regulator must provide 24V(±5%) given an input voltage of 27V(±5%). The 5V voltage regulator has to provide 5V(±5%) given an input voltage of 27V(±5%). The 3.3V voltage regulator has to provide 3.3V(±5%) from an input voltage of 5V(±5%).	- For the kitchen unit, connect the output of the 5V voltage regulator to an oscilloscope. Connect an input voltage of 9V to the 5V voltage regulator. - Measure the voltage. Repeat for the 3.3V voltage regulator with an input of 5V. It should provide stable voltage of 5V (±5%) for and 3.3V (±5%) respectively. - For the robot unit, connect the output of the 24V voltage regulator to an oscilloscope. Connect an input voltage of 27V to the 24V voltage regulator. - Measure the voltage. Repeat the same for 5V voltage regulator. Repeat for the 3.3V voltage regulator with an input of 5V. It should provide stable voltage of 24V(±5%), 5V (±5%) and 3.3V (±5%) respectively.
The 24V voltage regulator in robot must provide stable current which is 1.5A(±5%).	- Connect the regulator with its battery power source. Discharge the battery for 5 minutes. - Use current meter to measure the output current. The current must be remain 1.5A.
Must maintain temperature of chip below 150°C at a peak current draw of 1.8A.	- While doing the measurement of 1, use infrared thermometer to measure the IC temperature. Ensure that it will not be higher than 150°C.

Figure 4: R&V table for the voltage regulator

2.3.3 LED Indicator

LEDs will be placed on the robot to indicate that the robot's power is switched on.

Requirements	Verification
LED viewing angle has to be at least 40 degrees from 1m.	- Connect 7 LEDs in parallel and connect them with 210mA current source. Stand 1m away from the LED. Ensure the LED can be seen clearly.

Figure 5: R&V table for LED indicator

2.4 Control Unit

The control unit will be responsible in receiving and processing of data.

2.4.1 Micro-Controller Unit (in kitchen)

This MCU will be inside the kitchen. It will send out instructions to the robot on its next destination. It will also be connected to a LED display that will display the tables, our robot is currently "serving".

Requirement	Verification
The delay of data transmission between bluetooth and MCU should be within 2s.	- connect both a bluetooth receiver and transmitter to 2 different arduinos. - measure the timestamp as the receiver sends a signal - measure the timestamp as the transmitter receives the signal on the other MCU - the difference in the timestamps should be lesser than 2s

Figure 6: R&V table for MCU in kitchen

2.4.2 Micro-Controller Unit (in robot)

This MCU will be on the robot. It will serve 2 purposes. One of them would be to connect with the bluetooth transmitter and receiver unit (inside the robot) and gather instructions from the MCU in the kitchen on where to go next. Then, it would use that information to calculate the path it should take to reach its destination.

The other purpose is to gather data from the encoders that are connected to the motors and ensure that the robot moves the correct distance and at the velocity we want to.

Requirements	Verification
The delay of data transmission between bluetooth and MCU should be within 2s.	- connect both a bluetooth receiver and transmitter to 2 different arduinos. - measure the timestamp as the receiver sends a signal - measure the timestamp as the transmitter receives the signal on the other MCU - the difference in the timestamps should be lesser than 1s
Time to calculate path is smaller than 5s.	- run the path-finding software on the arduino and time it electronically. - the time taken should be lesser than 5s

	- try this with multiple table set-ups to ensure that the path-finding algorithm does not just pass the test because of an easy setup
Error in movement in x-axis should be less than ±0.5° for every 5 meters moved	- attach a fine laser pointer to one side of the robot - ensure that there are no physical obstruction in the path of the laser and that it is firmly attached to the robot - face the robot towards a white wall - mark the point where the laser is hitting the wall on with a pencil - move the robot 5 meters - mark the new point where the laser is hitting the wall - using the distance between the 2 points, calculate the angle of error. You might need to use small-angle approximation
Error in movement in y-axis should be less than ±0.5cm for every 5 meters moved	- mark the starting point of the robot clearly - let the robot move for a specific distance. Let's call that x. - once robot stops, measure distance robot has moved and calculate actual travelled distance. Let's called that x' - x' should be ±0.1% away from x
Error in angle when turning should be ±0.5°	- make a line or mark on the floor exactly along one side of the robot - let the robot turn 90° - now, make a line or mark on the floor exactly along the side chosen earlier - calculate the angle when those two lines meet. It should be ±0.5° of 90°

Figure 7: R&V table for MCU in robot

2.5 Communication Unit

The communication unit will be responsible for the conveying of information from the control unit to the respective components.

Requirement	Verification
The bluetooth transmitter and receiver should be able to transmit data for a distance of 0.05m to 8m without the presence of physical barriers between them. [3]	-Place the bluetooth transmitter 0.05m away from receiver. Send a signal from the MCU connected to the transmitter. Make sure the receiver can receive the data correctly and stably. Repeat this several times in the circumference of a 0.05m radius circle.

	-Place the bluetooth transmitter 8m away from receiver. Send a signal from the MCU connected to the transmitter. Make sure the receiver can receive the data correctly and stably. Repeat this several times in the circumference of a 6m radius circle.

Figure 8: R&V table for communication unit

2.5.1 Bluetooth Module in kitchen (central unit)

The transmitter sends the signals from kitchen's MCU to the robot's MCU. It will only send the next destination to the robot's bluetooth module, then it goes to idle mode. Only when the bluetooth receiver receives a signal confirming that the robot is back at its origin, it sends the subsequent orders to the robot.

2.5.2 Bluetooth Module in the robot (navigation unit)

The transmitter sends the signal confirming that the robot is back at its origin to the kitchen's bluetooth receiver to trigger the MCU to send the subsequent orders. The receiver receives the subsequent orders from the kitchen's MCU.

2.6 Mechanical Unit

The mechanical unit is responsible for the movement of the robot according to the instruction from the control unit.

2.6.1 Motor

The motors on the robot is a bipolar stepper motor. It can provide sufficient power for the robot's load and is the among cheapest option. We will be using two motors on the robot. Each of the motor is connected with a wheel on the robot. This is so that robot is able to change direction by rotating its wheels in opposite direction. The motor require 20-25 V voltage and 1.6-1.8 A current. [4]

Requirement	Verification
the motors can provide enough power to carry about 2kg weight in 10(±20%) RPM (about 10 meter per meter.	- Add 2kg load onto the robot and get it to move 10m - Measure the time taken by a stopwatch. Ensure the speed is 2.5m (±20%) per minute.

Figure 9: R&V table for Motor

2.6.2 Encoder

The encoder will be connected to the motors. It will be used to calibrate the rotation speed of the wheels as well as calculate the distance the robot has moved (using number of rotations of the wheel).

Requirement	Verification
The encoder can count the rotations of wheel accurately. The acceptable error rate of result is within ±5%.	- get the robot to move a distance of 2-3m. Measure this distance accurately. - while the robot is moving, data from the encoder connected to the MCU must be recorded to find the total rotations the wheels have done. - using the number of rotations, calculate the distance and compare it with the actual distance (measured earlier) - the difference should be within 5%

Figure 10: R&V table for Encoder

2.7 Display Unit

A LED display will be used to display the patrons the robot is handling.

2.8 Risk Analysis

There are a couple major risk factors in our project. The first one is the effectiveness of the bluetooth transmission and receiver system. This transmission system will allow signals to be sent from tables to the main MCU in the kitchen and send instructions from the MCU in the kitchen to the robot. Thus if there is a breakdown in this communication, it will majorly derail our project. So we intend to prototype a simpler version of this and confirm that we are able to send messages across correctly.

The other risk factor is in the navigation of our robot. If the robot successfully gets an instruction from the main MCU in the kitchen, it has to be able to identify its location, calculate the root to its next destination, move towards it and stop correctly. The risk factor here is regarding the precise movement of our robot. If our robot does not stop correctly, that could build up and result in a bigger error if the robot has multiple turns to its destination. This could even result in, the robot not reaching its destination and maybe even colliding with another table. To avoid this, we plan on calibrating the robot such that its calculations are precise and it stops at the correct place.

2.9 Schematics

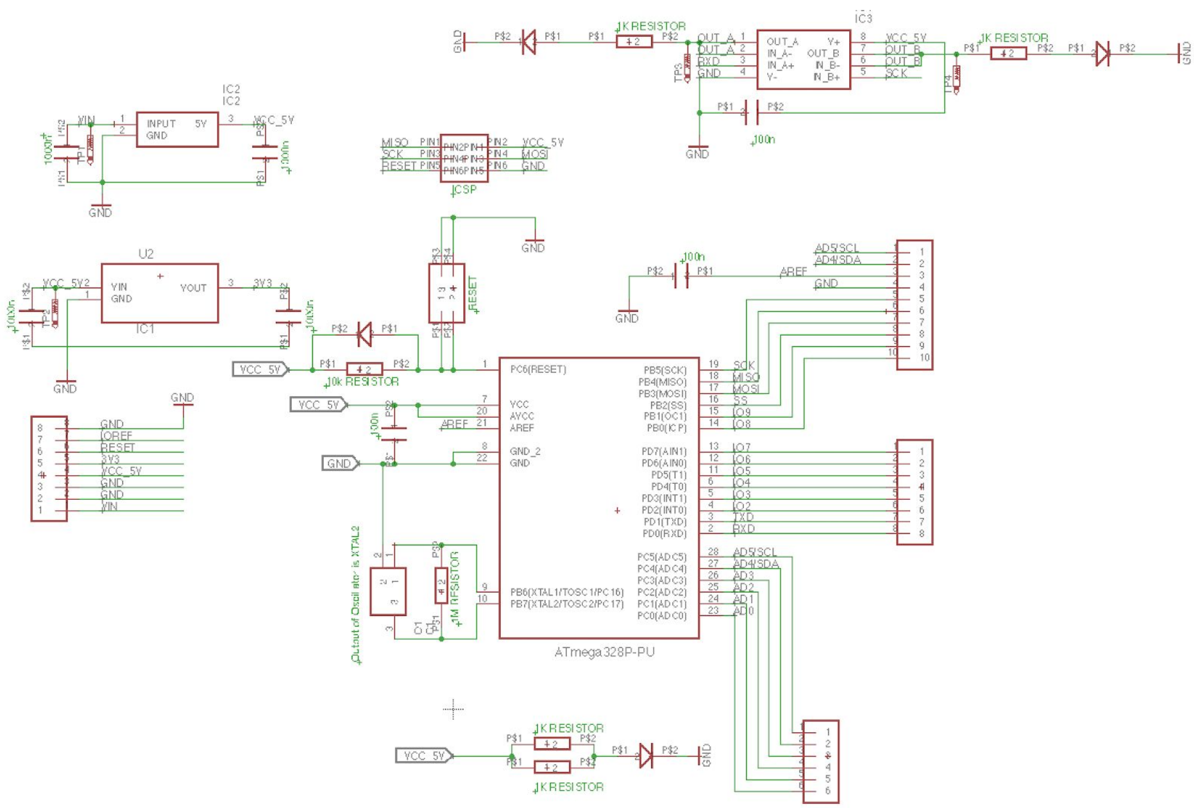

Figure 11: PCB schematic of the Control Unit (Kitchen)

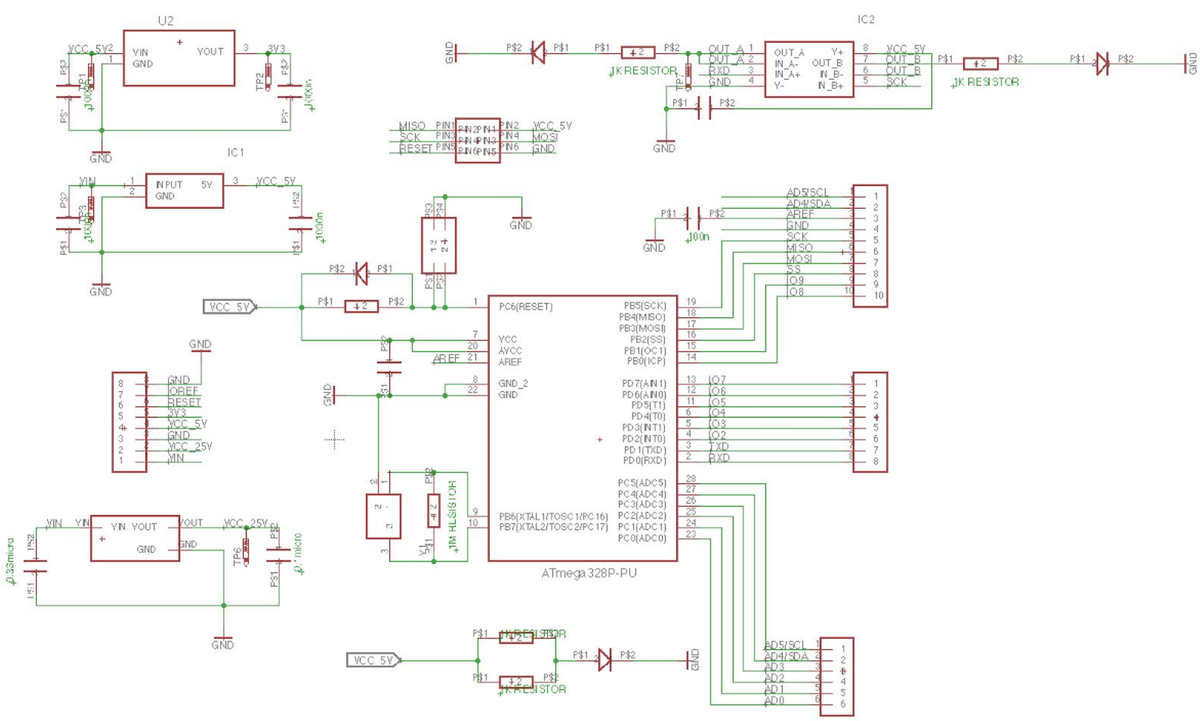

Figure 12: PCB Schematic of the Navigation Unit (Robot)

2.10 Software

2.10. 1 Path-finding algorithm

The path-finding algorithm is very critical to the robot. It will be given a destination location from the MCU in the kitchen and it will calculate the shortest way to navigate there. We will use Breadth-First Search, a graph-based search algorithm to identify the quickest path the robot can take. This algorithm will take into account the positions of the tables in the restaurant and identify a path that avoids them. Because of time constraints, in our project, the table's positions will be hard-coded into the robot. The flowchart below, figure 12, demonstrates how BFS would work.

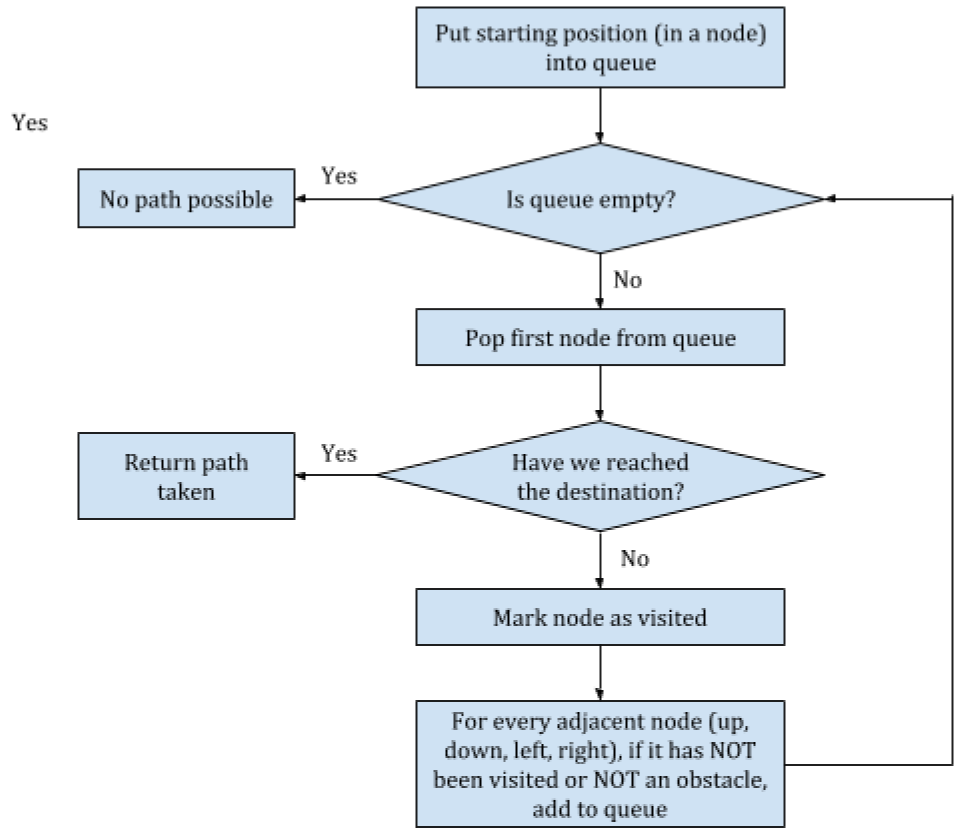

Figure 13: Flowchart demonstrating how Breadth-First Search

The first thing we would do is to get the layout of the restaurant. We would map the restaurant to a x & y coordinate system. We would account for the boundaries (i.e. walls) and the locations of the tables. We would also add a 'home base' for the robot. This is where the robot would 'rest' when it has no orders to deliver. Generally, it would make sense for this to be closer to the kitchen/staff. Once we have these values, we would program them onto the MCU. This would be a one-time task unless any of the values change.

The MCU in the kitchen would always be aware whether the robot is completing a task or is waiting for a task. When the robot is waiting for a task, the MCU in the kitchen sends new destination coordinates. The robot will act on those instructions and move onto its next task accordingly. The path would be calculated using a Breadth-First Search algorithm using its starting location, which the robot will always be aware of, and its destination, which it would get from the kitchen.

The calculated route would be returned as an array. Two indices would be read at once. The first index would contain the instruction that the robot should take, either to move forward or to turn. The second index would contain a parameter that would be useful for one of the previous functions. For move forward function, distance would be necessary. The function would calculate the duration and the velocity for the stepper motors based on the distance the robot has to travel. The turn function would look for an angle parameter. Turning left would 90, turning right would be -90 and turning 180° would be 180. The function would throw an error for any other value passed. So taking two indices at a time, the robot make either a straight drive or turn on its spot until it has reached its destination.

2.10.2 Bluetooth transmission protocol

We will be integrating bluetooth modules in our product. One bluetooth module will be used for each unit; the bluetooth module will be used for both transmitting and receiving. In the kitchen, the transmitter is used to transmit table number determined by the user interface. The table numbers are pre-set in the software. On the other hand, the receiver will be used to receive a signal when the robot returns to its origin. This signal triggers the MCU and tells it that the robot is ready to receive the subsequent order.

In contrast, the transmitter in the robot will be used to transmit the aforementioned signal that triggers the kitchen's MCU when the robot returns to its origin. Then the receiver in the robot will receive the table number determined by the user interface and relays that information to the onboard MCU.

In this project, we will be using the HM-10 bluetooth module. It is fairly low power consumption as it runs on BLE 4.0 (Bluetooth Low Energy). In order to establish a connection, the MCU sending the signal has to go into master mode and the MCU receiving the signal has to go into slave mode[5]. Once the master recognizes the slave, it will send a connection request. Once the connection has been established, their roles become master slave and slave device. Now, the master slave is able to send information to the slave device. The LED on the bluetooth module will also be turned on to verify that a connection is made.

2.11 Tolerance Analysis

Because our design is mostly focused on the accuracy of robots' navigation, we have to analyze the possible inaccuracies which come from the module. Our goal is to reduce our error margin as much as possible.

First, stepper motor has stop accuracy. In other words, the motor may stop at a position slightly farther or nearer than a desired position. According to datasheet, the stop accuracy of our motor is ±1 degree. As shown in the graph below(which is not the graph of our motor), the stop accuracy of the motor varies according to the shaft angle, but it won't be more than a range of degree(for our motor, this range is less than 1 degree one data sheet)

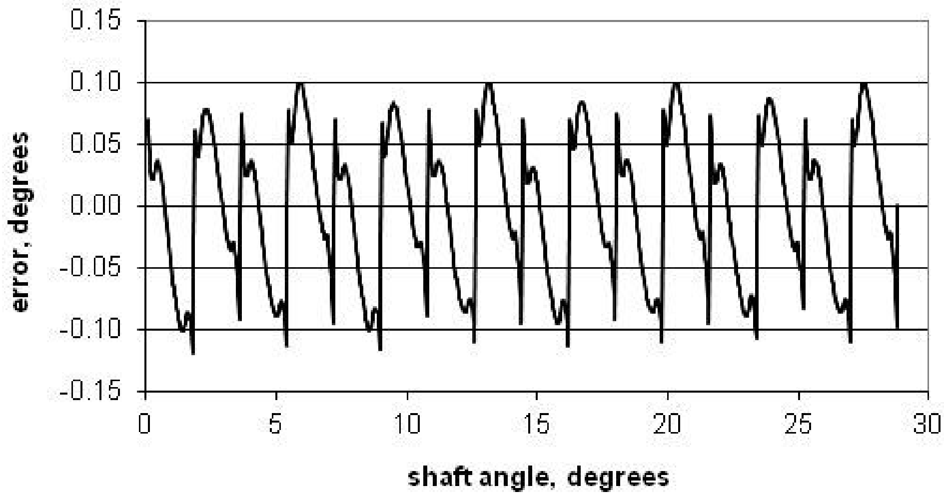

Figure 14: Encoder error vs shaft angle

$$\frac{\omega_A}{\omega_B} = R \qquad (2.1)$$

Therefore when the motor stops, it should give an error of 1 degree or less. The gear ratio will further decrease the error by equation 2.1. Since the gear ratio in our motor is 19:1, the actual stop error in our motor is around 0.017368°. Our wheel's diameter is 15 cm. Therefore, with the account of 1 degrees in error, the additional length travelled by our robot is 0.1377 mm.[7] By the way, because our robot won't stop too much times in one run, So the error will be relative small to 10 cm in total.

Apart from that, there are also error margins in the encoder calculations. There errors in the encoder can be divided into: radial error, shift error and eccentric error.

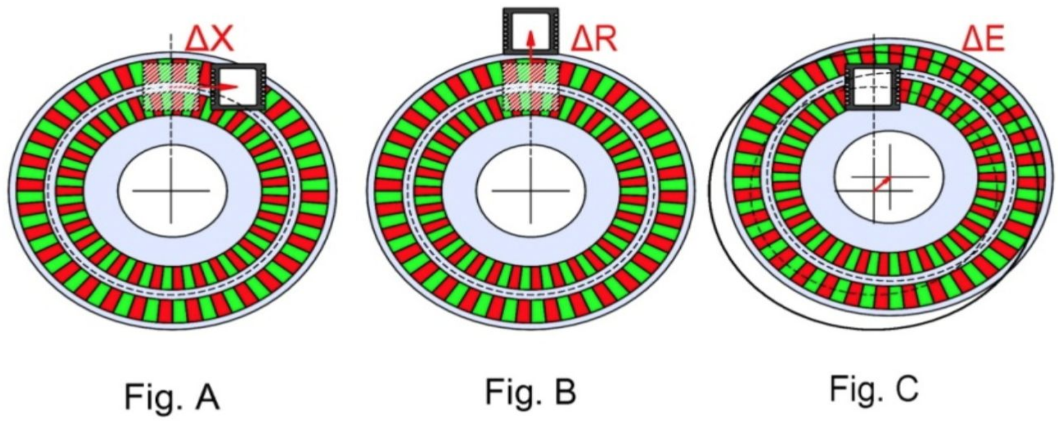

Figure 15: Stepper motor accuracy curves

$$ERR_{RAD} = 90\circ * (\frac{D+\Delta R}{D} - 1) \quad (2.2)$$

Radial error can be calculated by using the above equation, 2.2. We want the radial error to be within 1 mm for a 15mm-diameter encoder. This is so that the error calculation by the encoder, ERR is less than 0.6 °. The shift error should be within 2 mm for total. The shift error will not affect the phase difference for the calculation of the absolute value. Next, ΔE is measured by eccentrically mounted measurement target (ΔE) that causes the device to wobble. The smaller the diameter, the more the pitch of the measurement target changes. A long-wave error occurs that reduces the absolute measurement accuracy. We can estimate the error by using the equation, 2.3.

An eccentricity error of 10 µm leads to a phase error of 1.4° referenced to the sine period, or to an angle error of 0.05° referenced to the mechanical rotation. Therefore, it shows that ERR_{WOB} will be influenced by a lot. Henceforth the error of this part must be within 10 µm.[8] These errors caused by encoder can be decreased by converting the data in encoder to position of rotor. Because the encoder has grooves to regulate every step, for a whole period the error will not be accumulated. What we want to know is the final error and that will be smaller than 1 mm according to above calculation for rotating one time.

$$ERR_{WOB} = 360\circ * (\frac{\Delta E}{2*p}) \quad (2.3)$$

$$E_d = \frac{\omega_A}{\omega_B} \quad (2.4)$$

$$E_b = \frac{b_{actual}}{b_{nominal}} = \frac{90\circ}{90\circ-\alpha} \quad (2.5)$$

Third, the robot has system errors when it runs. The errors are mainly caused by two kinds of system errors: E_d caused by actual differences of diameter of two wheels and E_b caused by unsureness of efficient wheel distance. In reality operation of robot, E_d basically influences the path when robot goes straight line, while E_b influences the robot when it turns around a fixed points. E_d can be calculated using equation 2.4 where D_R is the diameter of right wheel, and D_L is the diameter of left wheel. We want this ratio to be 1(±0.01%). That means the difference of the wheels should be smaller than 0.01%.

E_b can be described using equation 2.5 where b_{actual} is the actual distance between two wheels, $b_{nominal}$ is the nominal distance between two wheels and α is the error of angle. We want the angle to be smaller than 0.05 °. So the value of E_b should be smaller than 1.00055586 and bigger than 1. That means the error of wheel distance should be smaller than 0.056%. And if we want to increase the accuracies for long-period use, we can add compensations in software.[9]

This error will be biggest problem in our project. Considering the difference of ground will increase the error, we will fix the demo place in lab. And we can use encoder to recalculate position, we can do some calibrations in software. And we just control the error to be smaller than 1% of total length, which is about 10 cm. To cope with this, the robot will also reset its position before the next order when it returns from its current trip to deliver the current order.

3. Costs

Manual Labor

Team member	Hourly Rate	Total Hours	Multiplier	Total Cost
Cheng Jin	$40	150	2.5	$15000
Jun Pun Wong	$40	150	2.5	$15000
Kausik Venkat	$40	150	2.5	$15000
			Total labor cost:	$45000

Figure 16: Labor Cost calculations table

Machine shop cost:
$55/hour * 5 hours = $275

Parts cost:

Part	Item cost	Quantity	Total Cost
Driver wheel	$9.90	2	$19.80
Caster wheel	$1.00	2	$2.00
Stepper motor	$39.90	2	$78.80
LED indicator	$0.20	14	$2.80
Bluetooth receiver/transmitter	$8.99	2	$17.98
ATMega328P (kitchen)	$7.99	1	$7.99
ATMega2560 (robot)	$12.71	1	$12.71
AA battery	$3.59	2	$7.18
AA battery holder box case	$1.99	1	$1.99
9V battery	$6.55	3	$14.10
9V battery holder box case	$4.99	1	$4.99
LCD 20x4 character display	$19.95	1	$19.95
Power button	$5.95	2	$11.90
Encoder	$2.12	2	$4.24
Stepper motor driver	$33.95	2	$67.90
		Total parts cost:	$292.59

Figure 17: Parts total cost calculation table

Total Cost = $45000 + $275 + $279.49 = **$45554.49**

4. Schedule

Week of	Cheng Jin	Jun Pun	Kausik
2/19	Read up on stepper motors and wheels to buy. Work on design document.	Build schematics for both the MCUs. Work on the design document.	Talk to machine shop and finalize physical design. Work on design document
2/26	Buy stepper motors and remaining parts. Confirm design with machine shop	Read up on bluetooth receiver and transmitter and buy those parts	Read up on MCUs, LED and display screen and buy those parts.

3/5	Work with the motors and encoders to ensure accuracy in distance moved by robot	Prototype with the bluetooth receiver and transmitter and work on sending and receiving signals	Prototype displaying information on the LED screen and controlling the LED indicators.
3/12	Double confirm kitchen MCU's PCB design with our finalized circuit plans and submit for purchase	Double confirm robot MCU's PCB design with our finalized circuit plans and submit for purchase	Start writing software needed for navigation and displaying information on LED screen
3/19	Spring break (Work on individual progress reports)		
3/26	Finish calibrating algorithm for distance robot moves	Finish implementing communication transfer protocol among bluetooth receiver & transmitter	Finish implementing algorithm for robot's navigation in restaurant
4/2	Complete verifications for motor, encoder	Complete verifications for bluetooth transmitter & receiver	Complete verifications for software, MCU & other parts
4/9	Work on integrating all the parts together	Work on integrating all the parts together	Work on integrating all the parts together
4/16	Debug any issues with the navigation system and prep for mock review	Debug any issues with communication module and prep for mock review	Debug any issues with software, MCU and prep for mock review
4/23	Start working on final paper and final demo (including verifications)	Start working on final paper and final demo (including verifications)	Start working on final paper and final demo (including verifications)
4/30	Be ready to submit final paper and give a good working demo of robot	Be ready to submit final paper and give a good working demo of robot	Be ready to submit final paper and give a good working demo of robot

Figure 18: Weekly Schedule

5. Ethics & Safety

We don't have too many safety risks in this project. One of the safety concerns is that we will have a power system on the robot which we need to regulate. We will likely use alkaline batteries which are generally safe but we need to design the power circuit properly so ill-fated incidents will not occur. This acts according to IEEE Code of Ethics Conduct #1 [9].

Also looking at IEEE Code of Ethics Conduct #9, we need to ensure the robot's navigation is not harmful to humans around it. To ensure that, we plan on controlling the speed of the robot to a fairly slow pace - slower than humans' walking speed. If somehow, the speed crosses a certain range that we deem risky, we will have some circuit breakers or safety measures that will completely stop power to the robot so that it does not harm anyone by moving extremely rapidly.

Despite designing a robot that is supposed to replace human workers, we have kept IEEE Code of Ethics Conduct #5 in mind. We believe that our project, in the long term, would help improve people's lives and advance modern technologies.

Lastly, we are also ensuring that we do not infringe anyone's intellectual property rights. By IEEE Code of Ethics Conduct #7, if we use the work of others, we will credit their contributions properly.

6. References

[1] "Absolute Encoder Design: Magnetic or Optical," iC-Haus GmbH, 2012, [Online]. Available at: http://www.ichaus.de/upload/pdf/WP6en_Magnetic_vs_optical_17092012.pd

[2] "Arduino – Sending A String Over Bluetooth Using The HM-10", Joshua Hrisko, Engineers' Maker Portal, 2017, [Online]. Available at: https://engineersportal.com/blog/2017/9/20/hm-10-bluetooth-module

[3] "ATMEGA328/P", Microchip, n.d., [Online]. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega328_P%20AVR%20MCU%20with%20picoPower%20Technology%20Data%20Sheet%2040001984A.pdf
f

[4] "Battery Technologies," Sparkfun, n.d. [Online]. Available: https://learn.sparkfun.com/tutorials/battery-technologies [Accessed: Feb. 1, 2018].

[5] "Bluetooth Interface with HM-10", Thrasyvoulos Karydis, 2015, [Online]. Available at: http://fab.cba.mit.edu/classes/863.15/doc/tutorials/programming/bluetooth.html

[6] "Calibration of systematic odometry errors for wheeled mobile robots," Bei Xuying, Ping Xueliag, Gao Wenyan, 2018, [Online]. Available at: http://www.arocmag.com/article/02-2018-09-001.html

[7] "CSTCE16M0V53-R0", Murata Electronics North America, n.d., [Online]. Available at:

https://www.murata.com/~/media/webrenewal/support/library/catalog/products/timingdevice/ceralock/p16e.ashx

[8] "Data sheet of motors," Pololu Electronics, n.d. [Online]. Available: https://www.pololu.com/product/1200 [Accessed: Feb. 7, 2018].

[9] "IEEE Code of Ethics," IEEE, n.d. [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html

[10] "LMV358MMX", Texas Instrument, n.d., [Online]. Available at: http://www.ti.com/lit/ds/symlink/lmv358-n.pdf

[11] "L78L24C", STMicroelectronics, n.d., [Online]. Available at: http://www.alldatasheet.com/datasheet-pdf/pdf/22640/STMICROELECTRONICS/L78L24C.html

[12] "MC7805CT", Micro Commercial Components, n.d., [Online]. Available at: http://www.mccsemi.com/up_pdf/MC7805CT(TO-220).pdf

[13] "MCP1700", Microchip, n.d., [Online]. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/20001826D.pdf

[14] "Restaurants in China are replacing waiters with robots," Business Insider, 2016. [Online]. Available:http://www.businessinsider.com/chinese-Restaurant-robot-waiters-2016-7

[15] "Stepper Motor Accuracy," Jeff Kordic, [Online]. Available: https://www.applied-motion.com/news/2015/10/stepper-motor-accuracy

[16] "Wi-Fi Direct vs. Bluetooth 4.0: A Battle for Supermacy," PCWorld, 2010. [Online]. Available: https://www.pcworld.com/article/208778/Wi_Fi_Direct_vs_Bluetooth_4_0_A_Battle_for_Supremacy.html [Accessed: Feb. 7 2018]

[17] "17HS19-1684S-PG19", Stepperonline Motors & Electronics, n.d., [Online]. Available at: https://www.omc-stepperonline.com/download/17HS19-1684S-PG19.pdf