AUTONOMOUS INDOOR FOOD DELIVERY AGENT

DESIGN DOCUMENT

Team Members: Nishqa Sharma, Ignacio Ampuero González, Belén Castellote López

TA: Weihang Liang

1. INTRODUCTION

1.1 Problem and solution overview

The ECE Building at UIUC is home to one of the strongest ECE Departments in the United States. The students, faculty and staff at UIUC are extremely busy, working on research, teaching, or studying. However, everyone needs food, getting which can often be a time-consuming source of disturbance or a break of concentration. Placing orders on apps such as DoorDash and Grubhub can be expensive, especially for the students, since these apps charge about \$4-\$8 [1] per order on average. Moreover, tracking or waiting for an order from an external restaurant can also be a source of disturbance. Hence, our goal is to build an autonomous food delivery agent, which can take customer orders, accept payment, and deliver food within the ECE Building, allowing for substantially cheaper, quick and hassle-free delivery from The Daily Byte Café, located in the ECE Building.

Our solution is to build an autonomous food delivery agent with a metal body, with a compartment for food, which will be activated by The Daily Byte Café employees, once they receive an email with the food to deliver and the room where it should be delivered. The food will be introduced in the compartment, and then the cafeteria employee will send the orders to the robot so it can move to wherever the client is. The food will be delivered for a very minimal delivery fee.

1.2 Visual Aid

Just in order to understand how our project works, we've created the diagram from Figure 1 in order to explain in a visual way.

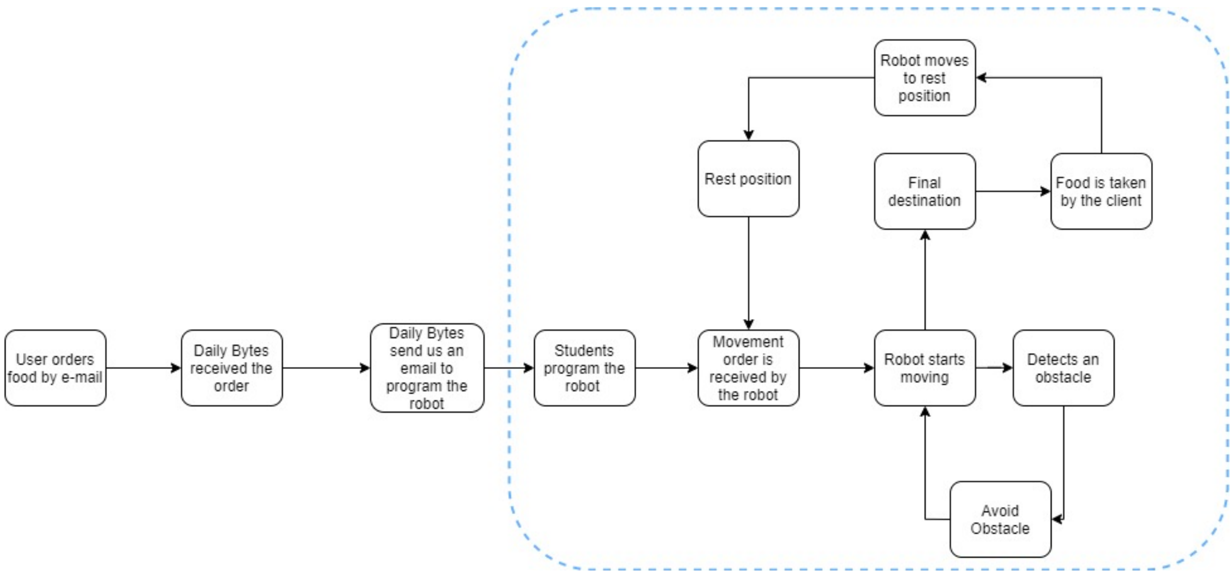

Figure 1: Visual Aid for Autonomous indoor food delivery

1.3 High Level Requirements

Our agent will be judged by the following criteria:

- The robot has an operation time of at least 15 minutes. Also, the maximum distance between 2 places in the ECE Building is around 600 meters, and we would like the robot to traverse that in about 9 minutes (540 seconds), for a maximum speed of about 1.1 m/s (4 km/hr).
- The robot stops 3-4 feet in front of the door of the room which the order was requested for.
- Robot must be able to detect the standing and moving obstacles in its path, while moving or standing, and adjust motors correspondingly. The expected accuracy is 70%.

2. DESIGN

2.1 Block Diagram

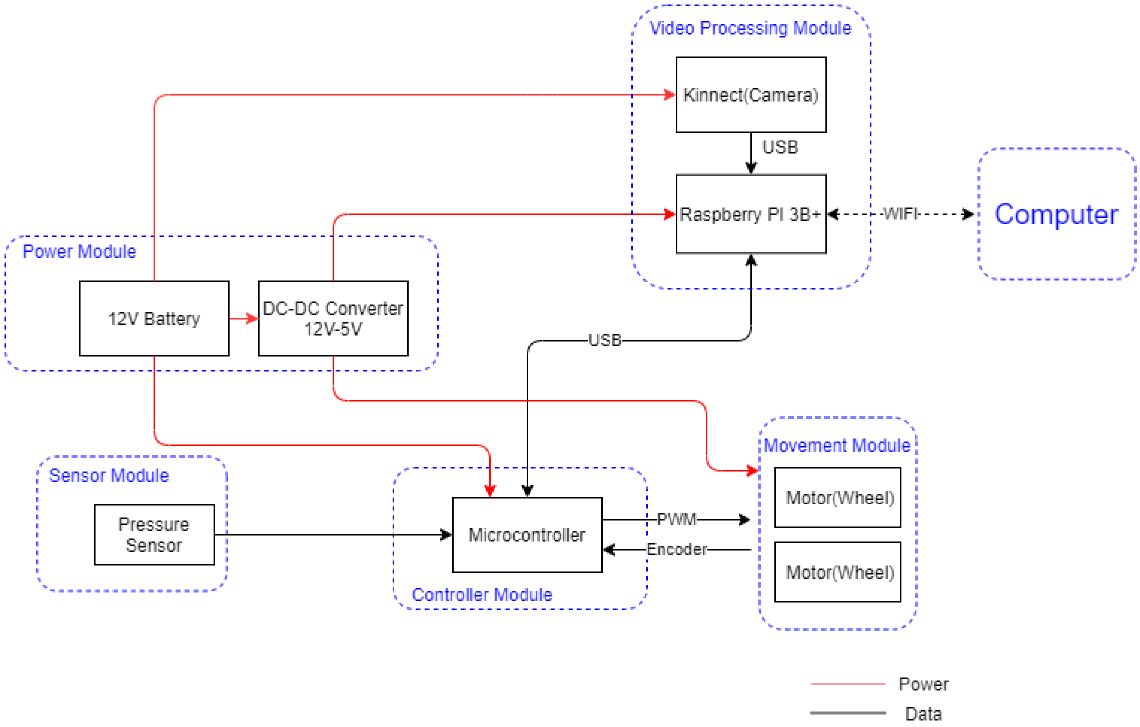

Figure 2: Block Diagram

1. The power subsystem:

   This block will contain a 12-Volt battery and a voltage regulator, that will ensure a constant voltage of 5V for the PCB and the Raspberry Pi.

2. The driving subsystem

   This block will contain 2 wheels in the back, and also a third spin wheel in the front. The 2 back wheels will be driven by two 5V DC Gear Motor Encoder. Each motor will have an encoder attached to it, which will be used to track position and orientation.

3. The sensor subsystem

   This block will contain a pressure sensor that will be used to calculate when food has been placed on or removed food from the cart.

4. The Video Processing Subsystem

   This block will contain the Kinect Xbox camera and a Raspberry Pi 3B+. The Kinect will transmit the video via USB to the Raspberry. The Raspberry Pi will have ROS installed within it, which will process the video and hold the mapping (2D map) and path planning libraries.

5. The microcontroller subsystem

   This block will contain the PCB with the microcontroller ATmega2560. This block will receive a velocity command from the Raspberry Pi and it translates this command into a velocity command for each motor. The PCB drives the motors through PWM . The speed of each motor is calculated with the information of the encoders of each motor and sent back to the Raspberry Pi..

2.2 Physical Design

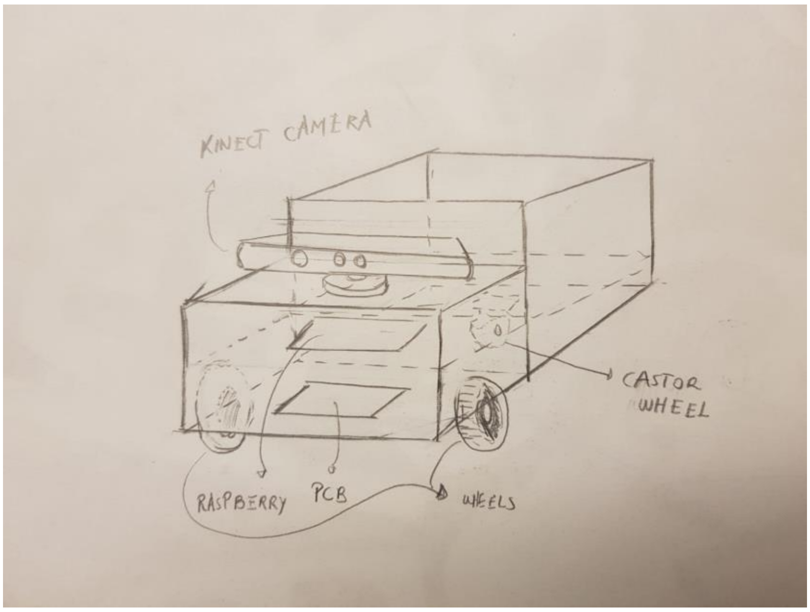

Figure 3: Physical Design

We plan to have a robot body of 0.8’ x 0.8‘base and 0.7’ in height made of Aluminum and lined with double reflective insulation material (used commonly by almost all restaurants in the USA to maintain food temperature). Also, Aluminum is a bad conductor of heat, and will help preserve the food temperature as well. There will be 2 wheels and one castor wheel (8 cm of diameter), which will be used to drive the vehicle using differential drive. The PCB will consist of the microcontroller and electronic circuit in order to control the pressure sensor and the motors (12V DC Gear Motor Encoder). The PCB will be powered by Li 12-V batteries, as well as the Kinect Xbox Camera and the Raspberry Pi 3B. Thinking about the pieces we have, the weight should be close to 2 kgs. The inside will be made of wood, for supporting the structure.

2.3 Power subsystem

The power is provided by 12 V to the camera and the PCB, but a voltage regulator is needed for the motor encoders and Raspberry Pi, because they must be at 5 V.

2.3.1 12 V Li-Battery

The battery should be able to work at least for 15 minutes while all the vehicle is moving around, with the voltage not going down more than 11.7V (the battery should be working in voltages from 11.7-12.3V). In order to analyze the battery working, we should estimate how many mA are passing through it while it's working. The microcontroller works with aprox. 100mA, the camera with 200mA and the voltage regulator that might be working at 1.3A (if we are looking at the Raspberry and the encoders). Approximately, the current will be 1600mA, while the device is working. Attending to our High-Level requirements, the battery should have a capacity of at least 400mAh. Because of this reason, we decided to use a 12V Li-Battery 3000mAh, because it fitted the requirements and at the same time it was the cheapest one.

Requirements	Verification
1. The battery should have an autonomy of at least 400mAh, with a remaining voltage above 11.7V	1. We will test this voltage during the building phase of the robot, by connecting a voltmeter/oscilloscope to the battery in parallel.

Table 1: R&V 12V Li-Battery

2.3.2 Voltage regulator (5V)

The voltage regulator is needed because the encoders from the motor and the Raspberry Pi 3B must work at 5 V. It is important that the voltage and the current works the same way as the encoders and Raspberry Pi, 5V/5% and within 0-50mA for the encoders and 0-2.5 A for the Raspberry Pi. The aforementioned specifications can be found at: Raspberry Pi 3 Model B+

Requirements	Verification
1. Provides 5V/5% from a 11.7-12.3V source. 2. Can operate at currents within 0-3A	1. A) We will verify the voltage provided by the regulator using a voltmeter/oscilloscope in parallel to the voltage regulator 2. A) To measure the current, we will make use of an ammeter/oscilloscope to measure the current and make sure that it lies in the 0-3 A range

Table 2: R&V Voltage regulator (5V)

2.3.3 Voltage regulator (12 V)

Requirements	Verification
1. Provides 5V/5% from a 11.7-12.3V source. 2. Can operate at currents within 0-2A	1. A) We will verify the voltage provided by the regulator using a voltmeter/oscilloscope in parallel to the voltage regulator 2. A) To measure the current, we will make use of an ammeter/oscilloscope to measure the current and make sure that it lies in the 0-2 A range

Table 3: R&V Voltage regulator (12V)

2.4 Driving subsystem

This subsystem is responsible for making the autonomous robot move. For this, we will need the help of two engines that spin two wheels connected to them. Besides, to control the speed and direction we need these engines to have a gyroscope and an accelerometer. We will also have the help of a third wheel, which in this case will be rotating.

2.4.1 Motor

Requirements	Verification
1. No-Load Current: $\leq 0.15A$ 2. Rated Voltage: 12V/1% 3. The encoders will need a voltage of 3.3V/+5%-5V/5%	1. A) We will measure the current passing through the motor by connecting a multimeter in series with each of the motors in turn, and the current passing through them in the "no-load" state, ie, the motors are not running, should be less than or equal to 0.15 A. 2. A) Since we are using a voltage regulator to supply the voltage to the motor, which is supposed to supply a steady voltage, we expect the tolerance to be low. To measure this voltage, we will measure the voltage across each motor (both will be connected in parallel) using a multimeter in parallel with each motor in turn 3. A) We will make use of a voltage regulator to supply 5V to the encoder. We will verify the voltage by using a multimeter in parallel to the encoder connections.

Table 4: R&V motor

2.5 Sensor subsystem

This subsystem is composed of a single sensor, which will be a pressure sensor and will be used to detect when the order has been removed so that the robot can return to its rest place. The detected signal will be sent to the microcontroller.

2.5.1 Load Sensor

The load sensor doesn't need to be accurate, but it must distinguish weight for over 200 grams and under 200 grams.

Requirements	Verification
1. The load sensor will need to measure with a tolerance of 10%	1. A) Choose something that weighs 200 grams. B) Measure it with the microcontroller to see if it is working or not.

Table 5: R&V load sensor

2.6 The Video Processing and Computer Vision subsystem

This subsystem is composed of the Kinect Xbox camera, which will collect all the video data, and the Raspberry Pi, which will have all the code to process this data and make decisions about the navigation of the robot

2.6.1 Raspberry Pi 3 Model B

Requirements	Verification
1. The Raspberry Pi will require a voltage of 5V/5% 2. It will require a current of 2.5 A/5%	1. A) We will verify the voltage provided by the regulator using a voltmeter/oscilloscope in parallel to the voltage regulator 2. A) To measure the current, we will make use of an ammeter/oscilloscope to measure the current and make sure that it lies in the 0-3 A range

Table 6: R&V Raspberry Pi 3 Model B

2.6.2 Camera

Requirements	Verification
1. The Kinect needs a 12V/1% stabilized voltage	1. Since we are using a voltage regulator to supply the voltage to the motor, which is supposed to supply a steady voltage, we expect the tolerance to be low. To measure this voltage, we will measure the voltage across each motor (both will be connected in parallel) using an oscilloscope

Table 7: R&V camera

2.7 Controller Module

This module will be responsible for controlling the wheels and the sensor, while at the same time will be communicating with the Raspberry Pi 3B, which will send the information for letting the microcontroller know how it should work with the wheels (speed and direction).

2.7.1 Microcontroller

For this project, the microcontroller is going to communicate with the motor by PWM, and it must be able to move both motors at different speeds (in order to choose the direction). It also must be able to receive information from the pressure sensor.

Finally, we decided to use an ATMEGA microcontroller, because we need how we should work with that, and it wasn’t more expensive than any other microcontroller.

Requirements	Verification
1. Motors must be able to work at different speeds 2. Information from the sensor must be sent correctly	1. and 2. A) Create a testbench in order to send signals to both motor and sensor, with different parameters. B) Test if the motors are running at the correspondent speed and if the sensor is working properly with each weight.

Table 8: R&V microcontroller

Risk Analysis

The weakest link in this design is the Video Processing Module. This is because it forms the crux of the robot, and it needs to process the constant data that it will get from the Kinect Xbox camera. In addition to that, it also needs to plan the path both locally and globally, specify what to do in case one fails, which can be a tough decision to make, and avoid dynamic obstacles on the fly. Hence, it is central to the project, heavy in computation and dependent on other modules for functioning. Hence, it is the biggest risk to the success of the project.

2. 9 Tolerance Analysis

The most critical module in our design is the Video Processing Module. This module is required to take the video data from the Kinect Camera, which will then be processed by all the code from the libraries mentioned in section 2.6.2 as well as our glue code on the Raspberry Pi.

For this module to function correctly, the main requirements from the corresponding aforementioned sections (apart from writing the correct code) are:

- The Raspberry Pi will require a voltage of 5V/5%
- The Raspberry Pi will require a current of 2.5 A/5%
- The Kinect needs a 12V/1% stabilized voltage

Hence, if these requirements are met, the project should be able to be completed with a significantly lowered chances of glitches, because it will mean that the camera and Raspberry Pi will function at maximum efficiency, and not lose power in the middle of execution.

Since the camera needs a steady voltage of 12V/1%, we decided to use a battery labelled as 12V, which provides power between 11.7V-12.3V. In order to accomplish a supply as steady and as accurate as possible, we decided to use a voltage regulator for 12 V. In addition to that, we have decided to measure the supplied voltage using a multimeter in parallel, so that we know that we are providing an adequate supply to the camera. Also, the Kinect draws a current of about 1.0A/5% (The Kinect power supply can provide that amount at 12V.)

Therefore, for the camera, our total error in power will be given by using the formula: $\frac{\Delta z}{z} = \frac{\Delta x}{x} + \frac{\Delta y}{y} + \ldots$

Where x is the voltage=5V, Δx is the error in voltage=0.12 V, y is the current=1.0 A, Δy is the error in current=0.054V, which we round to 0.05 V. Also, z is the total power consumed without error, which is 12V*1A=12 W.

Therefore, the error in power is given by 12(0.12/5 + 0.05/1.08) = 0.84 W.

The power demanded by the Kinect is 12 W, so the error in power for the Kinect is 0.84/12*100=0.1%, which is an acceptable amount of error to have for the camera power.

Since the Raspberry Pi requires 5V/5%, we will be using the same 12V battery with a 5V voltage regulator, so that we can again get a supply as accurate and as constant as possible. We will also measure this voltage with a multimeter in parallel with the Raspberry Pi, to make sure that we are providing an adequate supply to the

We will also measure the current supplied to the Raspberry Pi, to make sure that it falls in our desired range of 2.5A/5%, using a multimeter in series with the Raspberry Pi.

We performed the same calculation that we did for the camera, using the formula and method above, and our error came out to be 1.25%, which again is acceptable for this application.

2.10 Schematics

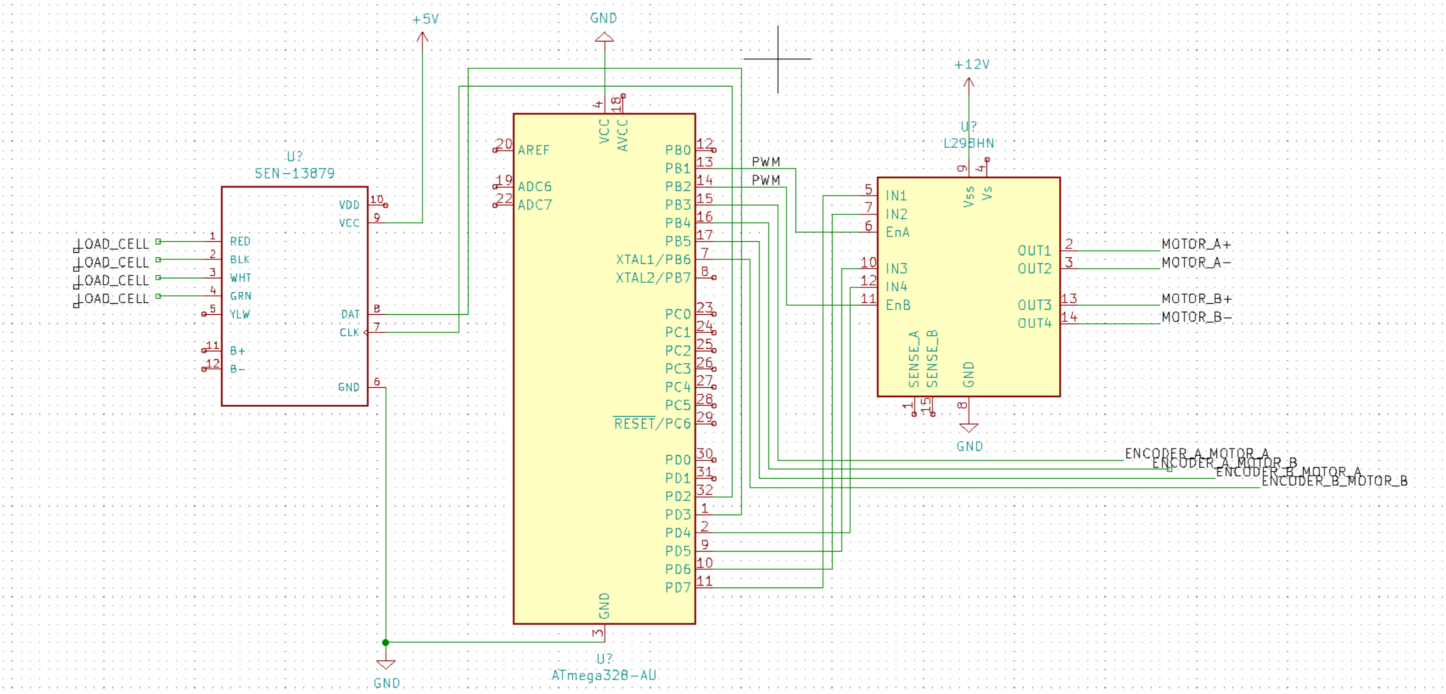

Figure 4: Schematics for the microcontroller

The schematic is not the final one and it is something conceptual. It is just a representation of what the final schematic could be. In it, it appears the microcontroller for the Arduino UNO (ATmega328), a dual full bridge driver (L298) for the motors and External Input Sensor Amplifier (SEN-13879) for the load sensor.

2.10 Algorithm Analysis

First, the user will place the order on the user interface found on the front-end computer. The order will be emailed to the Daily Byte, who will prepare it and notify the front end via email as well. Once the order is prepared, the front end will transmit the room number to deliver to the robot over Wi-Fi, as well as a signal to say that the order is ready to pick up.

Once the robot has these, it will make use of a Kinect Xbox video camera to implement Visual SLAM, which will be used to localize the robot and map the surroundings. To implement this, we will be making use of the “gmapping” ROS library. Once our localization and mapping is done, we will use our maps to plan the path using “local planner” ·

Once the robot has decided which path to take, it will start moving from its current location to the Daily Byte, by using the motor and wheel driving code from the microcontroller. Once it reaches the Daily Byte, a staff member will place the order onto the food compartment. Once the order has been placed in, the robot will use the same procedure as above to navigate to the room, and stop in front of the room, where the customer will pick it up.

2.11 Algorithm Flowchart

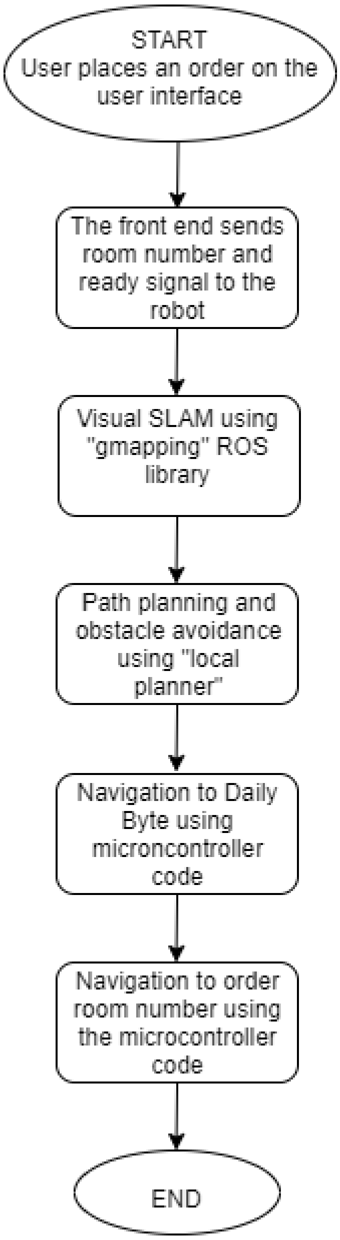

Figure 5: Algorithm Flowchart

3. Cost and Schedule

3.1 Cost Analysis

- Labor: our goal is that we all work the same number of hours and optimizing the time in the best possible way. For this, it will be necessary a previous distribution of tasks that make our work a place for a complete project and in which the high-level requirements are met. As we are assuming that for the project we are working as engineers without experience, a reasonable salary should be 30$/hour. Assuming we are going to work for 10 weeks from now and we've been working for 4 weeks, that's a total of 14 weeks, 10hours/week.

Total cost = 3 *2.5*14*30*10 = **31,500 $** for Labor cost

- Parts: Table 9 includes all the parts that need to be bought to build the robot.

Description	Part #	Quantity	Cost/unit ($)
Kinect Camera	1	1	24.39
Wheel+Motor_Encoder Combination (CHIHAI)	2	2	20
Castor Wheel	3	1	5
12 V Li-Battery	4	1	15.96
Raspberry Pi 3.0 B	5	1	35
Microcontroller (Arduino)	6	1	11.85
Load sensor	7	1	7
Voltage regulator (5V)	8	1	7
Voltage regulator (12V)	9	1	2.74
Total for all the parts			**148.94**

Table 9: Cost for each part

- Sum of cost:

Total for the Labor	31500 $
Total for the Parts	148.94$
Overall Cost	**31,648.94$**

Table 10: Total Cost Analysis

4. SCHEDULE

Week	Nacho	Nishqa	Belén
2/24/20	Version 1 Schematic circuit	Version 1 Schematic circuit	Version 1 Schematic circuit
3/2/20	Version 2 Schematic circuit and first version PCB design	Version 2 Schematic circuit and first version PCB design	Version 2 Schematic circuit and first version PCB design
3/9/20	Mapping using gmapping	Mapping using gmapping	First version PCB design and improvement PCB design
3/16/20	Mapping using gmapping	Mapping using gmapping	Microcontroller design
3/23/20	Test and debug the gmapping library software module	Start working on local planner for path planning and obstacle avoidance	Microcontroller test and bug fix
3/30/20	Local planner for path planning and obstacle avoidance	Local planner for path planning and obstacle avoidance	Code to control the motors
4/6/20	Local planner for path planning and obstacle avoidance	Local planner for path planning and obstacle avoidance	Code to control the motors
4/13/20	Work on testing the whole vehicle together	Test and debug local planner software module	Code to control the motors and implement on the real motor
4/20/20	Bug Fix+Integrated System test	User Interface and Bug Fix	Bug Fix+Integrated System Test
4/27/20	Prepare final presentation and begin final report	Prepare final presentation and begin final report	Prepare final presentation and begin final report

Table 11: Schedule and distribution of assignments

5. DISCUSSION OF ETHICS AND SAFETY

Since our agent is going to be carrying items from the Daily Byte, it will most likely carry liquid beverages. We will have to make sure that those do not come in contact, with the electrical components, which could cause short circuits, which would not be significant enough to hurt anyone, but would cause the robot to malfunction, or potentially stop working completely until the circuit were repaired. To prevent this, we will add an extra layer of casing around the electronic components present on the chassis, compliant with the IP67 regulatory standard.

We will also have to make sure that the navigation is accurate, so that no people or property present in the ECE Building are hurt by the body of the navigating robot. This is in alignment with section 1.2 of the ACM Code of Ethics [7]. We intend to do this by adding temporary stops and emergency stops, so that if the robot is unable to assess where to turn in a situation, it will come to a halt rather than collide with a person or property.

Although we will take full care to only use the video stream data for the purposes of navigation, it is possible that this data could be intercepted and used maliciously over the Wi-Fi for gathering private data from the ECE Building. This would go against sections 1.1, 1.2 and 1.6 of the ACM Code of Ethics [7], because it would violate the privacy and safety of individuals in the surroundings of the product. To avoid this, the transmitted video data can be encrypted in future versions of the project, so that it cannot be used maliciously.

5. CITATIONS

[1] Helling., Brett, et al. “DoorDash - Everything You Need To Know About This Delivery Service.” Ridester, 2 Feb. 2020, www.ridester.com/doordash-on-demand-food-delivery-service/.

[2] “Bear Robotics, Inc.: Future of Restaurants.” Bear Robotics, www.bearrobotics.ai/products/.

[3] “Savioke.” Savioke, www.savioke.com/.

[4] vestacookingrobot001, and vestacookingrobot001. “China's 1st Cooking Robot: Robot Waiter: Dishwasher Supplier - VestaCookingRobot.” Vesta Cooking Robot Co., Limited, 13 Jan. 2020, www.vestacookingrobot.com/.

[5] mpetrova92. “These Butler Bots Are Building Resumes at Companies like FedEx and Residence Inn.” CNBC, CNBC, 25 Aug. 2017, www.cnbc.com/2017/08/24/fedex-and-hotels-and-hire-autonomous-delivery-robots.html.

[6] https://www.mdpi.com/1424-8220/19/24/5480/pdf

[7] “The Code Affirms an Obligation of Computing Professionals to Use Their Skills for the Benefit of Society.” Code of Ethics, www.acm.org/code-of-ethics.