Howard University
Department of Electrical and Computer Engineering

EECE 404 - Senior Design II

Deliveroid
Final Report

Autonomous delivery robot

By
Shelton Allen
Jonathan Goberdhan
Conrad Blash

Graduate Advisor: Derrick Anang
Faculty Advisor: Charles Kim

Summary: The transport of physical documents and other various items in an office space can still be automated. Our solution design for this problem, the Deliveroid, is an autonomous robot that seeks to carry out this transport as easily and as efficiently as possible. Using knowledge of microcontrollers as well as the engineering process in general, our group of three electrical and computer engineers attempted to assemble this solution design but found that we were unable to complete our short-term objective due to time and intellectual constraints. Overall, the experience offered by this project will allow the three engineers to assess any engineering task much more realistically in the future.

Problem Statement: There exists a need in the workplace for a seamless and effortless exchange of physical documents and other items between co-workers or departments. Currently, to have such an exchange, people have to tediously take time out of their busy day and walk to the office of the person they wish to deliver to.

Primary Objective: Automate this delivery process with small robotic car able to transport these documents and items.

Short-Term Goal: Our short-term goal is to have the Deliveroid transport items between offices on one floor of a building.

Long-Term Goal: Our long-term goal is to have the Deliveroid transport items across multiple office floors and buildings.

Design Requirements: When initially preparing our design requirements we examined similar products to our idea that are currently on the market. We'll examine these products more in depth later in this report we formed the following constraints based on what we discovered.

From a socio-cultural perspective, we want the Deliveroid to be aesthetically pleasing and compact. A trait shared among all similar products on market is their clean, compact and unobtrusive design. In a workplace setting we felt that a similar approach would be applicable for our design. From an intellectual standpoint, we need the deliveroid to implement technology that we three, senior, Electrical and Computer Engineering majors, can understand. The technology we decided to use was multiple microcontroller devices with peripheral sensors. From a Financial standpoint, we saw that market price of a product very similar to ours, Roomba had a Market Sale Retail Price (MSRP) of around $200 USD. We wanted to, at the very least construct a single Deliveroid for under that MSRP. Finally, from a Time perspective, we had only 2.5 to 3 months to finish a Deliveroid design meaning we again had to make sure that the design wasn't overly complex and could easily be divided among us.

With these constraints in mind we then created more quantifiable design specifications. Looking at our socio-cultural constraint, we felt that the approximate dimensions should not exceed 2 feet in length, width or height. We also felt that as for sound emission while in full operation, Deliveroid, should not exceed 45 Decibels, talking volume. We came to this conclusion as most offices are quiet workplaces. In terms of power we felt that a rechargeable battery with 20 minute life should ample as most trips will simply be to destination and back to source, where it can be charged if low on battery. In terms of peripheral sensors mentioned earlier in this report, we want to have these sensors be able to have Deliveroid instantly correct navigation if off course, and avoid objects within 50 centimeters of the front. Next, since we are aiming for an improvement in the business office environment, we felt that Deliveroid should be able to move at 4 miles per hour, walking speed. If it is moving slower than walking speed, there won't be much improvement work time efficiency. A requirement we also considered was a 15-20 pound weight limit. This would be enough to support three mid-sized laptops and several documents at once. We were unable to implement this in our design due to the absence of a weight sensor. In addition to this, we also had planned on including a delivery notification system in the form of LED lights along the perimeter of the Deliveroid's frame that would flash upon delivery. This was also not implemented due to the absence of parts.

Standards & Requirements: There were three internationally recognized standards we tried to conform to when developing our solution design. Naturally, there are many more that we probably missed, and we were unable to test if our final design conformed to our three criteria. Nonetheless, we still examined these three Standards & Regulations so that should our design advance significantly in the design process, it would not be too much of a burden to have it conform to production standards.

The first standard we examined was Federal Communications Commission (FCC) part 15 dealing with interference of radio-frequency devices. It was very clear from the beginning of our design process that some wireless communication would be implemented in our final design. We wanted to keep FCC part 15 in mind due to this. The next standard we considered was International Electrotechnical Commission (IEC) 61000-4-2, which deals with electrostatic immunity testing. There will be many interconnected components used in our design and we felt that it is a safety hazard for potential users of Deliveroid to be using it untested for shock immunity. Third, we examined was International Organization for Standardization (ISO) 13849-1, which deals with safety of machinery. Deliveroid will be a smaller piece of machinery moving at a significant speed in the workplace. With this standard in mind our Faculty Advisor suggested making a more rounded frame for our solution implementation, in order to avoid unnecessary safety hazards.

Current Status of the Art:

The main engineering principles/concepts used in the design of the Deliveroid is a concept that has gotten a significant amount of recent traction - robot autonomy. In the case of our Deliveroid, its autonomous aspect lies in the fact that the delivery robot needs to be able to autonomously travel between departments in the workplace in order to transfer documents and other items between co-workers and/or departments. A fully autonomous Deliveroid would need to be able to gain and store information about its environment, move without human intervention, cause no harm to humans, property or itself. Some technologies that aid sensing the environment includes electromagnetic spectrum sensors, sound sensors, temperature sensors, distance sensors, altitude sensors and simple cameras. For example, distance sensors may aid autonomy by allowing the bot to "sense" an obstacle or person and avoid collision.

There have been many advances in the search of a solution that will allow for full autonomy. Starship Technologies have found a way to use Satellite positioning to create an autonomous food delivery robot. The robot is able to use its positioning system to find out what roads it should take to get to its destination and where each intersection is with great accuracy. Indoors, an engineer can make use of sonar sensing to sense objects and enable the robot to follow a specific path along a wall. Many commonly used technologies for getting robots to go to a specified destination simple guides the robot to the goal destination. This can be accomplished by putting an inductive loop or magnets in the ground underneath the robot and have it follow the path. Another method may be drawing a dark line on a lighter colored floor. This would create a contrast between the dark line and the floor. Using an array of IR sensors, the robot is able to calculate the reflectance of the surface underneath and thus try its best follow where there is a lower reflectance.

These technologies, however, do not create a fully autonomous robot. This is why many designers are seeking to use light detection and ranging sensing (lidar) in order for the robot to create a 3D map of its surroundings. By using such a technology, the Deliveroid would be able to gather details about its surroundings and store them on a storage medium for later use. The Deliveroid would be able to learn about new environments and thus self-localize itself in that environment and be able to self-calculate paths to a certain destination. Another technology available today is called ROS (Robot Operating System). ROS provides device drivers, visualizers and libraries that can better enable a robot to function as intended. ROS also provides a navigation technology that collects data from odometry and sensor streams and outputs commands that tell the robot what to do and in what direction. The navigation stack can also be configured for the particular shape of the robot which enables it to perform at a higher level.

Within the last 5 years robotic delivery service has become more of a prominent feature with new products being released on an almost monthly basis. One such product is a robot built by Effidence SAS, a French based company, called PostBOT. Post bot is being tested in a german city and is supposed to help with carrying mail items for deliveries by carrying up to three hundred and thirty pounds and it moves and gets its sense of direction by tracking the movements of the legs of the deliveryman automatically following behind. It can also work in all weather conditions and can navigate obstacles by using computer vision software to identify object and move around them. It has four wheels and it is rectangular in shape with a height of about 5 feet.

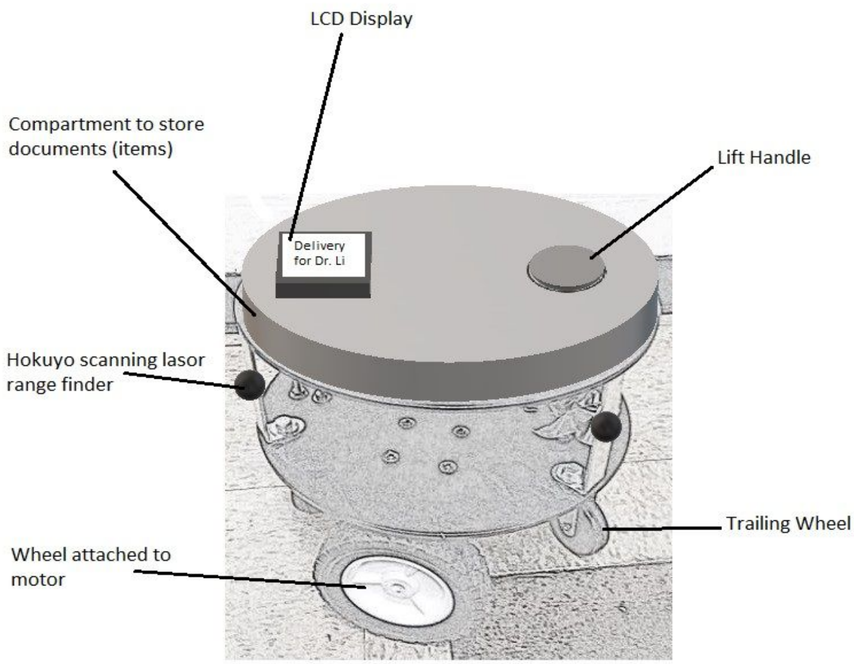
Another robot named Gita built by Piaggio an italian company is a personal delivery device that looks similar to an exercise ball in shape can carry up to forty pounds and moves at a maximum speed of twenty two mph. It lasts for eight hours and has a unique design of two wheels over the entire body which allows for zero turning radius. It was designed to be a maneuverable as possible in highly condensed cities and can go any place that is wheelchair accessible. It has multiple cameras embedded around its body and a storage bay with a fingerprint locking mechanism. It was designed to make carrying around heavy everyday items easier.

China has also deployed another delivery robot earlier this year made by Jingdong. It can carry 5 packages at once and travels up to twelve mph and can climb a twenty five degree incline as well. It is a four wheeled robot that costs almost one hundred thousand dollars to manufacture. When delivering it sends a text notification at its destination to the recipient and is uses face recognition technology or a pin code in order to access contents.

Deployed in dc is a food delivery robot developed by Starship technologies in London. A 2 feet tall robot deliver meals and groceries short distances around the city. The robots navigate by using cameras and gps technology and builds internal maps of the city through exploring new areas so that it can compute faster was to get to its destination the next time around. It can identify stop signs and can tell when a car is passing and can wait till the path is clear to cross without getting crushed. The robot comprises of six wheels with a compartment that has a code lock to open.

Proposed Designs:

Design 1:

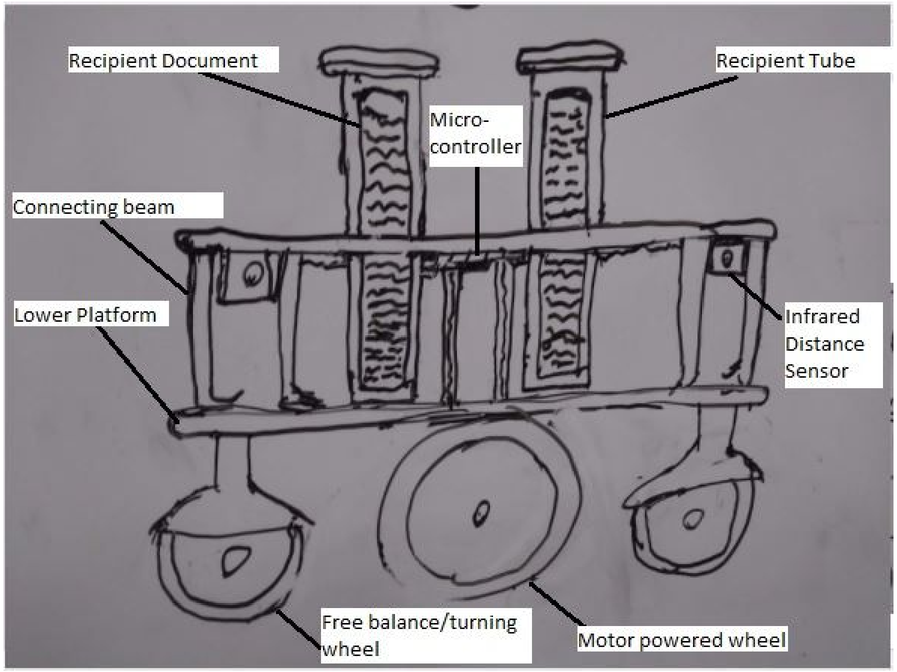


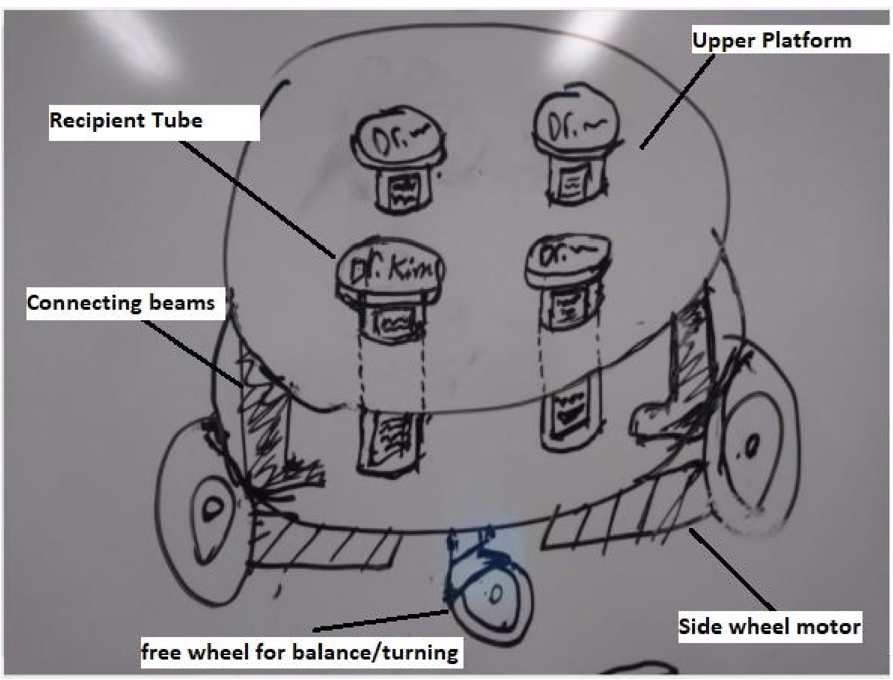
Description: This first design incorporated the wheeled robot frame left behind by the VIP team, BusBot. That frame included Two Plastic plates connected by small iron bars and bolts, with two motors, two motorized wheels and two free-moving wheels for turning; all beneath the bottom plate. This design includes a lid compartment for the documents on the top with LCD display pad for delivery status and user interface. This design as a microcontroller that connects to wifi and runs on ROS with scanning laser for distance sensing.

Pros: This design was relatively clean of clutter and used relatively state of the art technology. It also requires very little physical modification to a framework we already possess. wheel positioning allows for zero point turns

Cons: The relatively new technology implemented along with ROS was out of our area of expertise and did not fall in line with our Time and intellectual constraints.

Design 2:

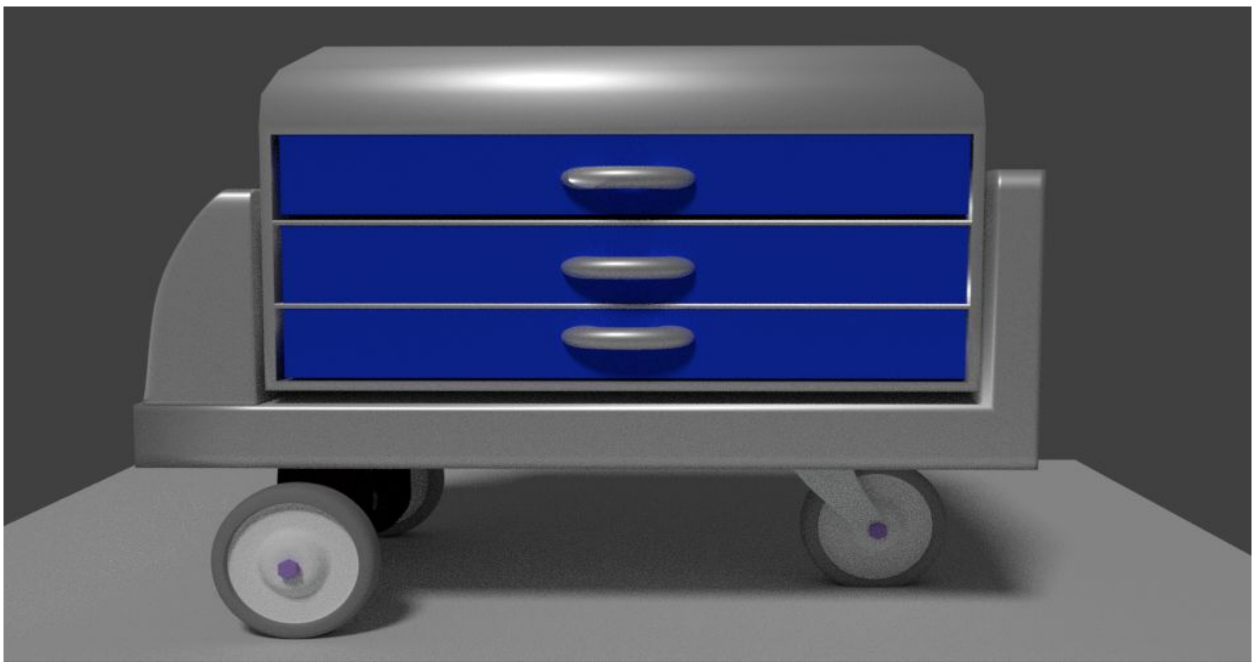




Description: The above design is also based on the framework left behind by VIP team BusBot. In this design holes are made in the upper plastic plate that would be able to hold parcel tubes that can hold documents. The two motor powered wheels and two free turning wheels are retained from the BusBot design. All sensors, wireless modules and microcontrollers are kept underneath the top plastic plate in order to adhere to safety and electrostatic discharge regulations (to some degree). There are only 4 parcel tubes present and they are manually labelled by the users of this Deliveroid.

Pros: Uses technology that our group is quite familiar with and also uses a framework that our group was in possession of, wheel positioning allows for zero point turns.
Cons: This design requires permanent modification to our only framework immediately on hand. It also is limited in the amount of users that can access it due to physically named recipient tubes.

Design 3:



Description: This design uses a completely different framework from the BusBot. There are only 3 wheels available with two motor powered wheels and 1 free turning back wheel. The set of drawers in the middle allow simple transport of documents and other items. Distance sensors are along the perimeter of the design. A physical shock sensor lies between the front shield and the drawer in the center. The drawers are detachable and are held in place with vertical rails along the back of front shield and the back of the rear bumper. A weight sensor lies at the base of of the interior of the lower gray frame and is able to detect a load weight. The microcontrollers and batteries are kept in a sealed compartment above the top drawer. Wiring to components such as the motors, shock sensor and distance sensors start in the top compartment and run along the interior of the drawers in the form of a bus detachable with the drawers. The portion of the bus that stays in the lower frame fans out to all sensors along the interior of the lower frame.

Pros: Excels in many of our design requirements. Uses technology we were somewhat familiar with. Better meets standards and regulations established early in the design process.

Cons: Uses a completely new frame we'd have to build from scratch, uses many different components in conjunction and will be difficult to program.

Top Designs:
Our top solution design, we felt was design 3 as it completely outshines the frame modifications to the BusBot. Though design 2 was a close second in terms of feasibility, due to the inability acquire time in the campus's machine shop, we were forced to not use any of our three designs. After this realization we proceeded to simply create software that would allow for the base functionality of our original design requirements on the hardware we had available.

Solution Implementation:
The following block diagram (fig. 1) is the hardware level flow of how we originally planned to have to Deliveroid execute 1 delivery.

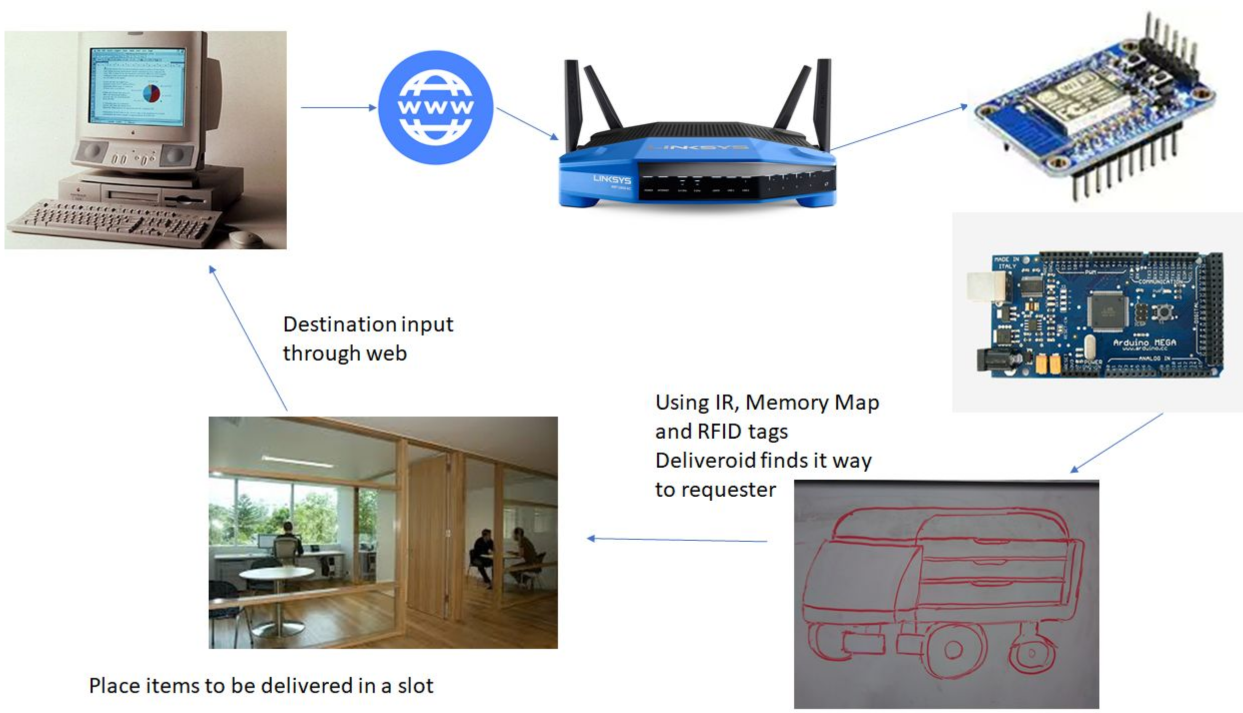


Fig. 1 Hardware block diagram 1

After the destination is sent to the wireless module it moves to the destination and awaits for further requesters/destinations.

The following figure (fig. 2)shows our initial software block diagram before adjustments were made due to lack of available parts

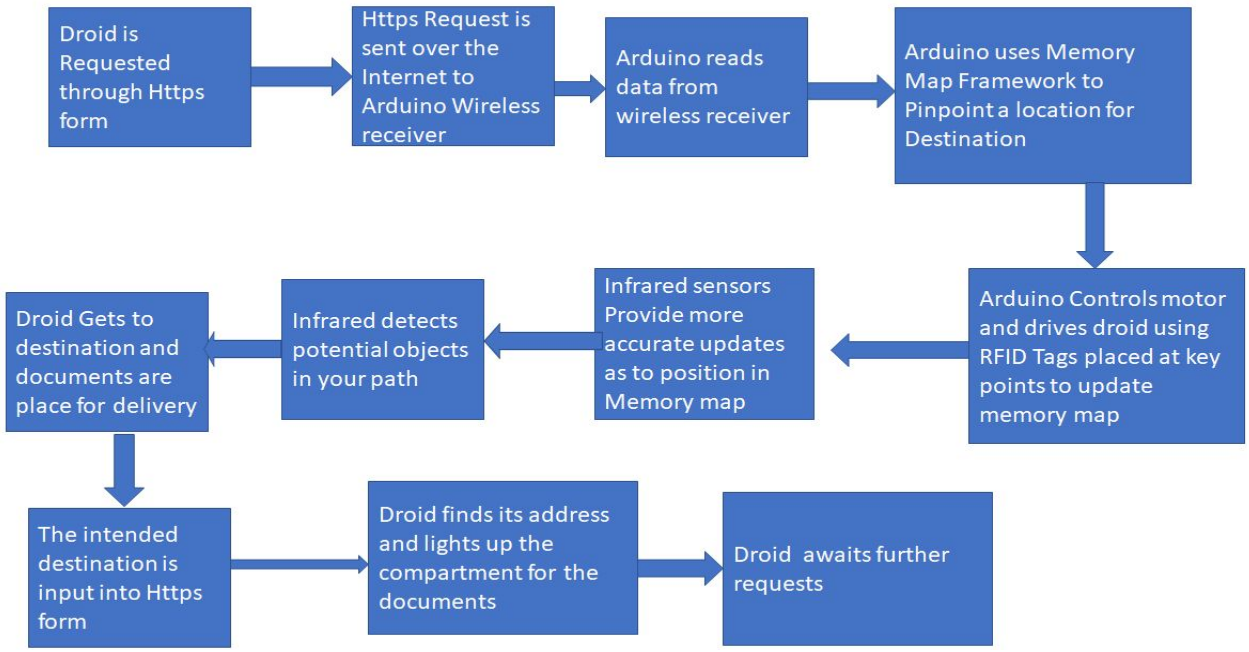


Fig. 2 Software block diagram 1

The below figure (Fig. 3) shows an adjusted software block diagram flow for deliveries

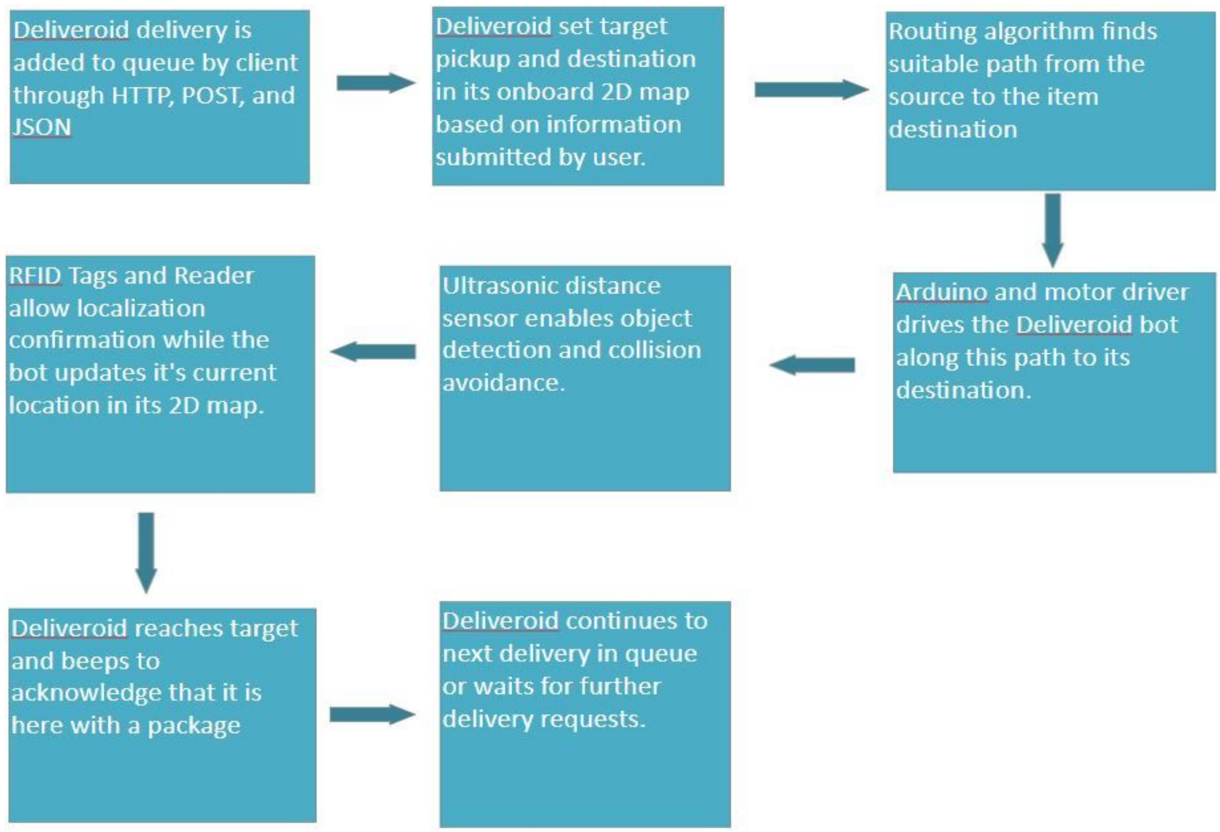


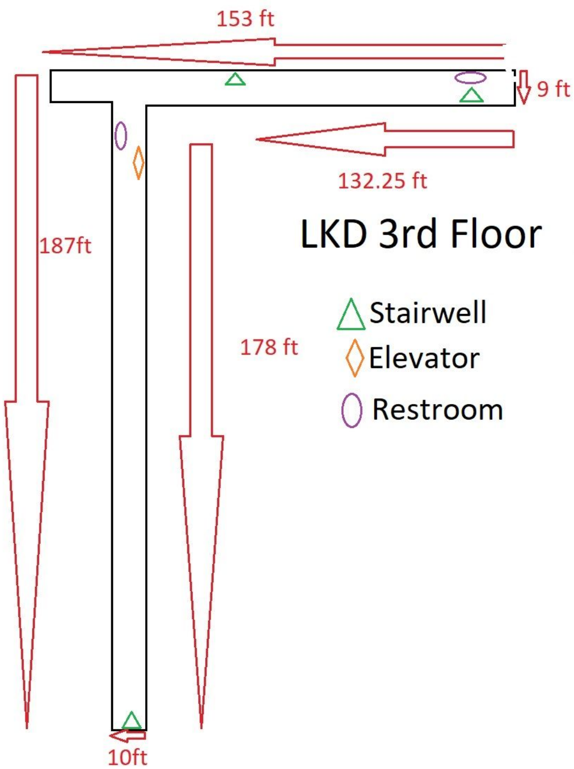
Fig. 3 Software block diagram 2

The difference between fig. 2 and fig. 3 lies in the initial state. Like the hardware block diagram, (fig. 1), fig. 2 navigates to the office to pick up from, and then navigates to the delivery office. In our final implementation we simply had the Deliveroid move from source to destination rather than from current position, to source, to destination.

With these flow diagrams in mind we proceeded to divide up the tasks available as such.

Conrad:

Conrad was responsible for creating the Example 2D map that would be programmed into the main microcontroller (Arduino Mega 2560 in fig. 6). He also programmed the map into the microcontroller and how the Deliveroid would traverse through said map. He programmed the Ultrasonic distance sensor as well. Below is the figures showing the planned example map (fig. 4) and the 2D map fitting (fig. 5).



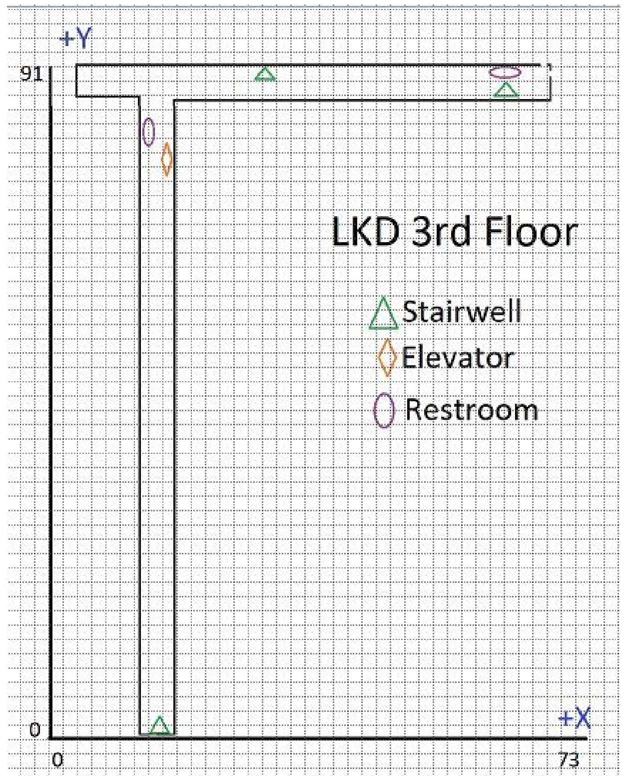


Fig. 4 Rough Lewis k Downing Hall F3 Map

Fig. 5 X-Y plane fitting of Fig. 4

Explanation: After finding exact measurements of our simulation environment, Conrad then fitted the environment's map to an X-Y plane and doubled the effective size of one (x,y) coordinate from 1 square foot to 2 square feet. Then using conditional if statements in the programming of the board, he created boundaries which the Deliveroid should not able to reach. As for traversal, he used more conditional statements to calculate when the deliveroid should: turn left, turn right, reverse direction, and go forward. The Ultrasonic sensor was programmed using example code found on the internet and was integrated into a side microcontroller (Arduino Uno) that would communicate with and stop map progression in the main microcontroller whenever an object came within 50 centimeters of the sensor.

Shelton:

Shelton was responsible for implementing network features. This included programming of a ESP8266 wifi module and the Frontend and Backend development for a client interface with the Deliveroid. He programmed the RFID tag sensor as well. The figures below are a pin mapping of the wifi module (fig. 6) and the client interface web page (fig. 7).

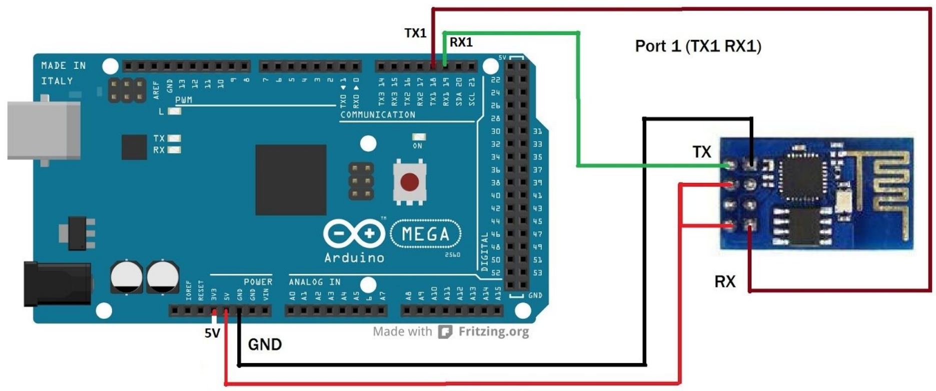


Fig. 6 WiFi module pin mapping to arduino mega

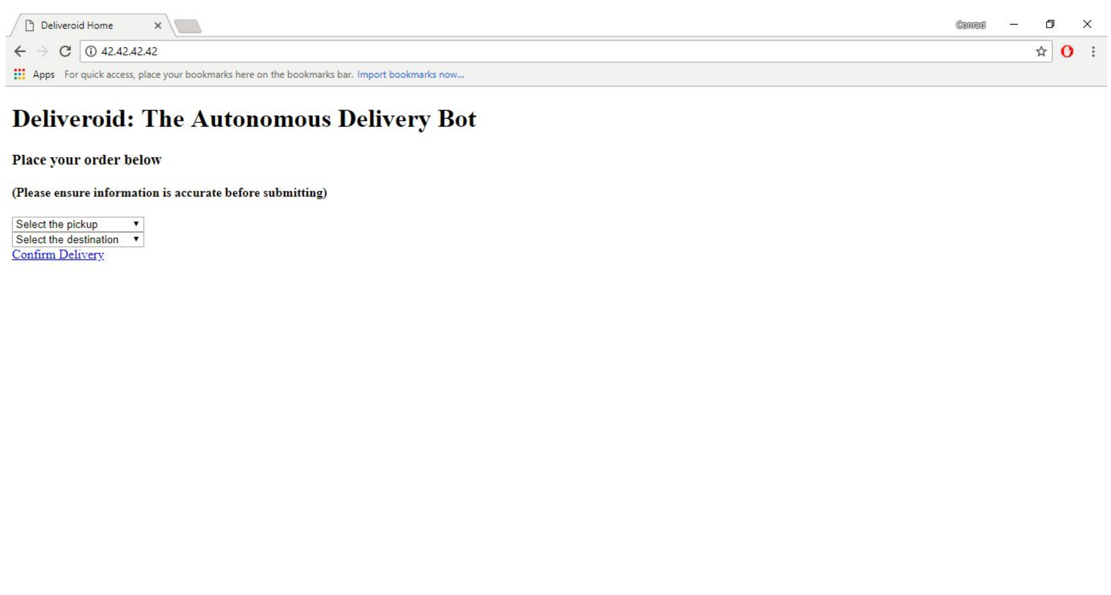


Fig. 7 Web page for request sending

Explanation: Using JS and JSON programming, Shelton created a client webpage that is able submit a source and destination value to the main microcontroller through the main microcontroller's wifi module that creates a private network. The webpage is only accessible on the wifi module's private network, and must be programmed into the module separately from the board. This request system was then integrated into the main microcontroller's 2D mapping by Shelton as well. As for the RFID tag reader, Shelton programmed the side microcontroller to accept in the list of destinations available from the main microcontroller and have the RFID reader recognize a tag that will tell the side microcontroller to send a signal to the main microcontroller to stop advancement through the 2D map.

Jonathan:
Jonathan was responsible for selecting the motor driver board to interface with the DC motors available from the BusBot frame and the microcontrollers. He was also in charge of programming said driver board (L298N) and finding a the optimal power source for it. He then calculated and programmed optimal turn and advance times for wheel operation. The figure below, (fig. 8) shows the pin mapping of the driver board-microcontroller interface.

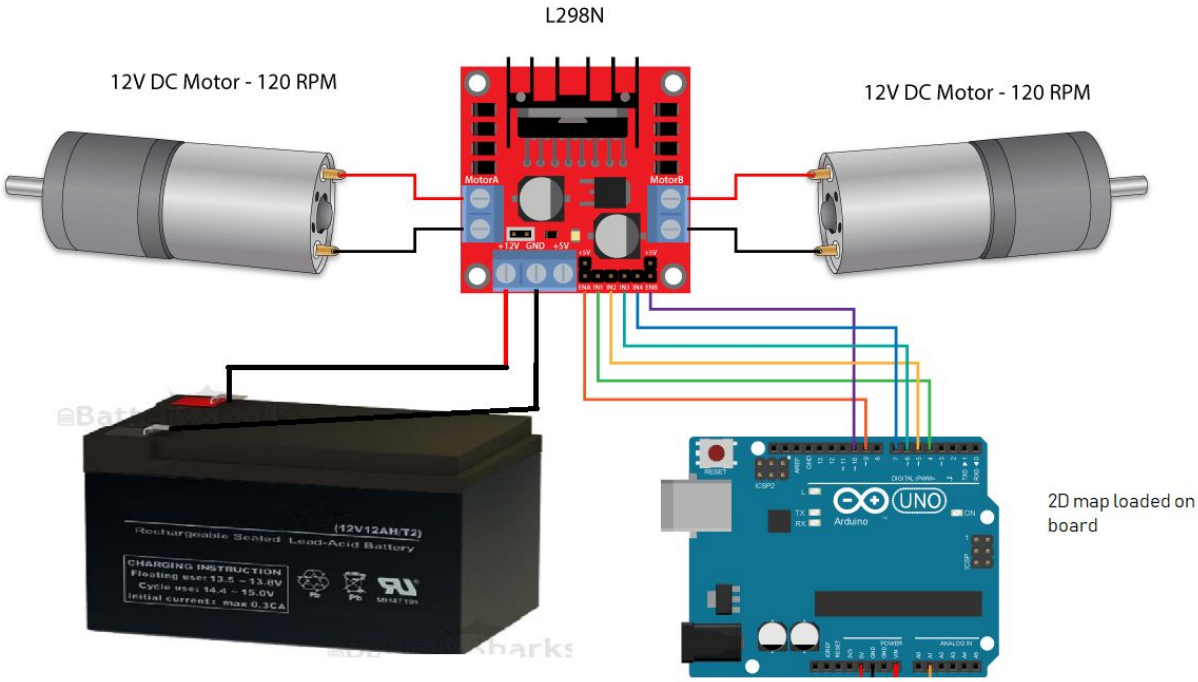


Fig. 8 The L298N linked to the motors, battery and Arduino uno

Explanation: After researching and acquiring the correct Motor Driver board for our DC motors, Jonathan tested the capabilities of different power sources and found that an industrial grade lead acid 12-volt battery to be enough to run the motor driver and motors. However, connecting two of these 12-volt batteries in parallel yields full power, optimal motor performance. Using an Arduino Uno for testing, Jonathan measured various wheel and motor revolution times and created motor driver programming that is functional for both 1 12 volt battery and 2 12 volt batteries. He then integrated this programming into the map present on the main microcontroller.

Components: Below is a list of the technology and components used in the solution design with the exception of the plastic and metal frame (Tbl. 1). A simplified view of the all the technologies in conjunction is also below (fig. 9), along with a visual of the final solution design (fig. 10). There were several components that were vital to how the team implemented the Deliveroid. These included a fast master microcontroller - the Arduino Mega 2560. The Arduino

Mega 2560 features 54 digital I/O pins, 16 analog inputs, 4 hardware serial ports (two of which were used in the Deliveroid implementation), 256 KB of flash memory and a clock rated at 16 Mhz. With the minimal response time necessary to prevent damage to the robot, it's surroundings or any person in its vicinity, the Arduino Mega was able to deliver near instantaneous reactions to different situations that arose.

Tbl. 1

Component name	Quantity Used
Arduino Uno	1
Arduino Mega 2560	1
Ultrasonic Distance Sensors	3
Motor Driver Board	1
12V 7.5AH DC Battery	1 OR 2
125Khz RFID Reader	1
125Khz RFID Tags	2
12V DC Motor	2
ESP8266 WiFi Dev Board	1

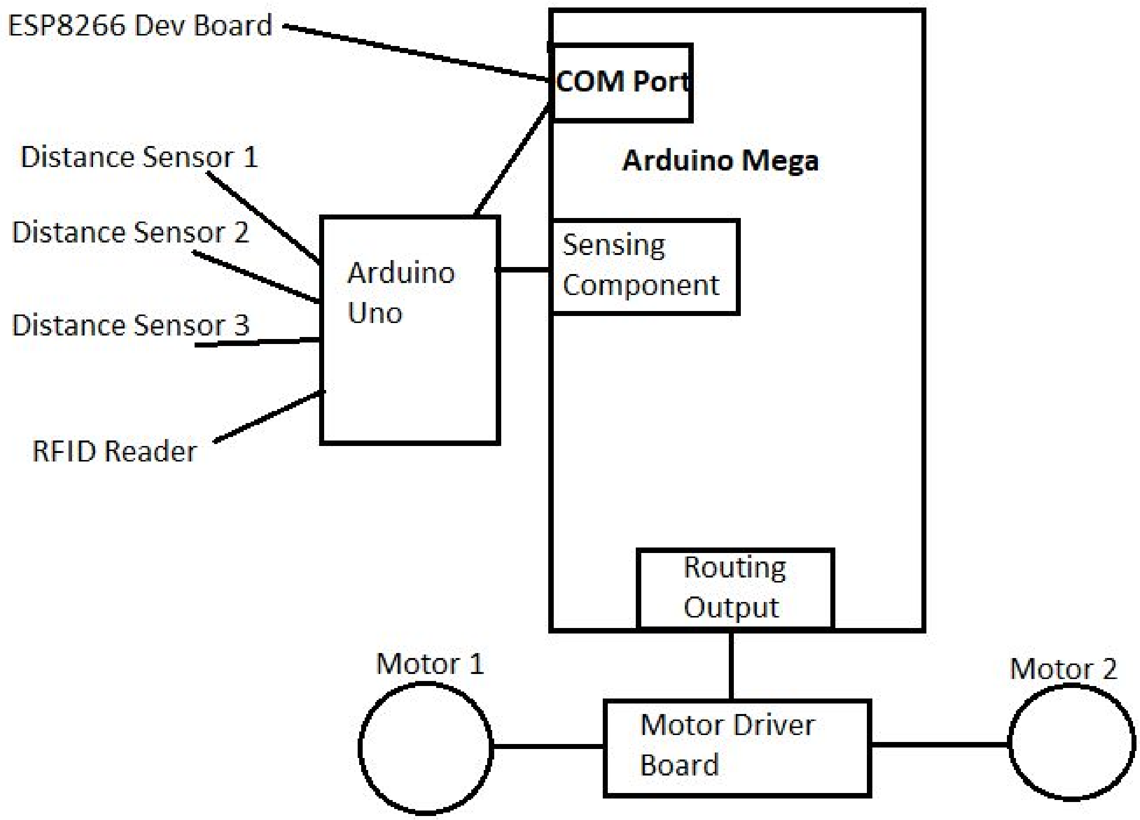


Fig. 9 simplified diagram of all major components

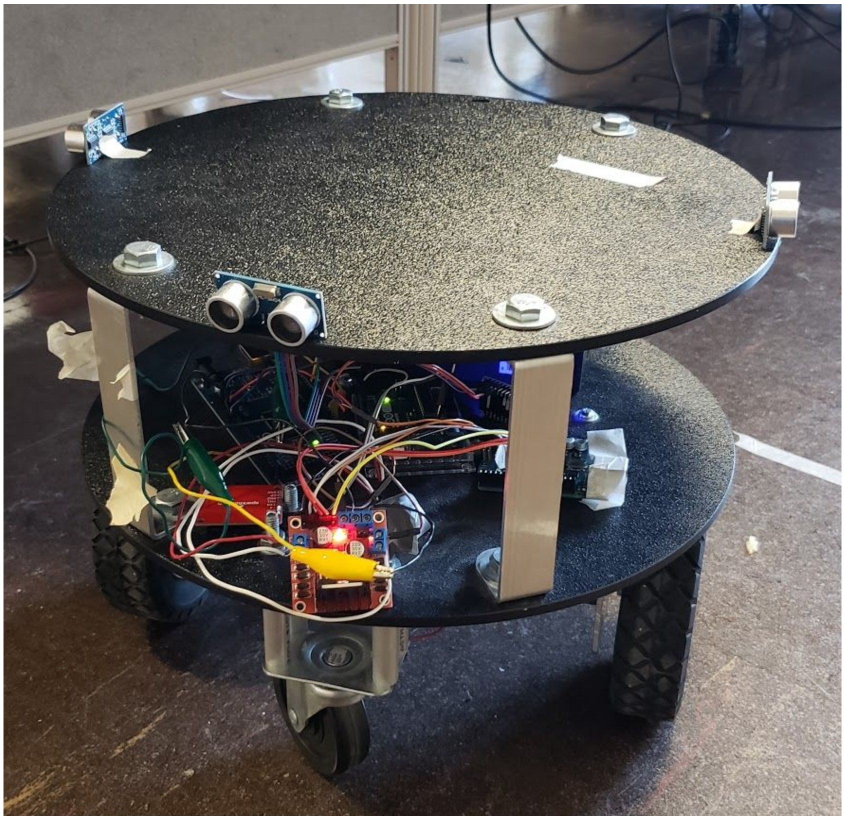


Fig. 10 Final product of solution design

Conclusion: Complete assembly of the solution design was very last minute. As one of the most integral components in testing, the motor driver board, was absent from solution design development until the weekend before presentation day. Without the motor driver board were unable to control the DC motors and thus were confined to testing the different components of the design separately. Upon finally receiving the driver board it took quite a bit of time figure out the specifics of its operation and we only had 1-2 days to test the final design as a whole, with every block of programming integrated.

Thus, though we were close, we were unable to complete our short-term objective due to the burden of our time constraint. These are the specifics of what went wrong. First is the power. Due to the one 12 volt battery we had available not being enough to reach full motor power, one of the two wheels would often slow down or stop inconsistently. This made programming exact 90 degree or 180 degree turns almost impossible. Second is the dual microcontroller interface. As mentioned in the solution implementation section, the main microcontroller is the arduino mega 2560 while the side microcontroller is the arduino uno. The uno was home to all distance and RFID tag readers. The implementation of side board was place so that the sensors that could potentially stop the Deliveroid would act as soon as they detect the threat/possible collision.

While we did confirm the correct operation of these “collision detectors”, we were unable to have the main microcontroller receive commands to stop from the side microcontroller. This could have been remedied with more testing time. Our last problem that needed more time to fix was the steady state functionality of the main microcontroller. All of the code that we combined from our components is on the code framework that is Conrad’s 2D map. This 2D map was never tested for non-terminating scenarios. This in combination with the large amount of source code on the main microcontroller, (2D map, traversal, motor driving, wifi receiving), made it very time consuming to troubleshoot problems.