**DIY Automatic Delivery Trolley**

by

Utchawin Namdee 5911697

Patipan Arunwattanawong 5912537

A report submitted in partial fulfillment of the requirements for

the degree of Bachelor of Engineering in

Mechatronics Engineering

Project Advisor

Asst. Prof. Dr. Narong Aphiratsakun

Examination Committee

Dr. Jerapong Rojanarowan, Dr. Wisuwat Plodpradista,

Assoc. Prof. Dr. Jiradech Kongthon, Mr. Sunchanan Charanyananda,

Mr. Amulya Bhattarai, Mr. Ehsan Ali

Assumption University

Vincent Mary School of Engineering

Thailand

October 2020

Approved by Project Advisor:

Name: Asst. Prof. Dr. Narong Aphiratsakun

Signature:

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Plagiarism verified by:

Name: Mr. Ehsan Ali

Signature:

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

This project will describe the automatic delivery trolley for sending documents to a specific station. This system is able to deliver documents without human interaction using trolley mechanisms. Various type of documents can be sent/receive to and from several stations. Moreover, the trolley system can be control using command issue via mobile application. The system consists of infrared (IR) sensor to make the trolley run along the black line and ultrasonic sensor to avoid colliding with objects or people. This project can be used in various physical/environmental situations. For example, sending the documents or medicine in the hospital, homes with elderly or disabled people. In addition, the scale can be developed or expanded to use other works like transporting various parts in the factory or transportation between cities.

**Table of Contents**

[Abstract 3](#_Toc48837776)

[1. Introduction 5](#_Toc48837777)

[1.1. Introduction of topic 5](#_Toc48837778)

[1.2. Project objective 5](#_Toc48837779)

[2. Project overview 5](#_Toc48837780)

[2.1. Initial study & Background study 5](#_Toc48837781)

[3. System 6](#_Toc48837782)

[3.1. Block diagram 6](#_Toc48837783)

[3.2. Controller 7](#_Toc48837784)

[3.2.1. Arduino Uno 7](#_Toc48837785)

[3.2.2. ESP8266 9](#_Toc48837786)

[3.3. Sensors 10](#_Toc48837787)

[3.3.1. IR Infrared Obstacle Avoidance Sensor Module 10](#_Toc48837788)

[3.3.2. Ultrasonic Sensor HC-SR04 11](#_Toc48837789)

[3.4. Driver and Motor 12](#_Toc48837790)

[3.4.1. Motor Driver Module L298N 12](#_Toc48837791)

[3.4.2. Motor 12](#_Toc48837792)

[3.5. Miscellaneous 13](#_Toc48837793)

[3.6. Mechanism Design 16](#_Toc48837801)

[3.7. Circuit diagram of DIY Automatic Delivery Trolley 16](#_Toc48837802)

[1.3. Flow Chart 17](#_Toc48837803)

[4. Conclusion 18](#_Toc48837804)

[4.3. Completed Work 18](#_Toc48837805)

[4.4. Future Work 18](#_Toc48837807)

[5. Reference 19](#_Toc48837808)

[6. Appendix 20](#_Toc48837809)

[6.1. Motor and Ultrasonic Code 20](#_Toc48837810)

# Introduction

## Introduction of topic

Nowadays, automation industrial plays significant role in the world as we can see in the daily life such as automatic door and vacuum cleaner robot. We can say that automation technology initiates other technology so that they have their own names and branch, for example, Robotics. We as a student would like to apply some automation technology to a daily life so that others can easily access to it. What we are going to do is DIY automatic delivery trolley.

## Project objective

Delivery something to someone might be a hard time if there is far distant between them. So, we develop this DIY automatic delivery trolley to solve this problem. The user will use smart phone application to control the Trolley to deliver the item from one place to another.

# Project overview

## Initial study & Background study

Currently, there are many types of controller, but we found 2 main types of controller that are suitable for our project: Arduino and Raspberry Pi.

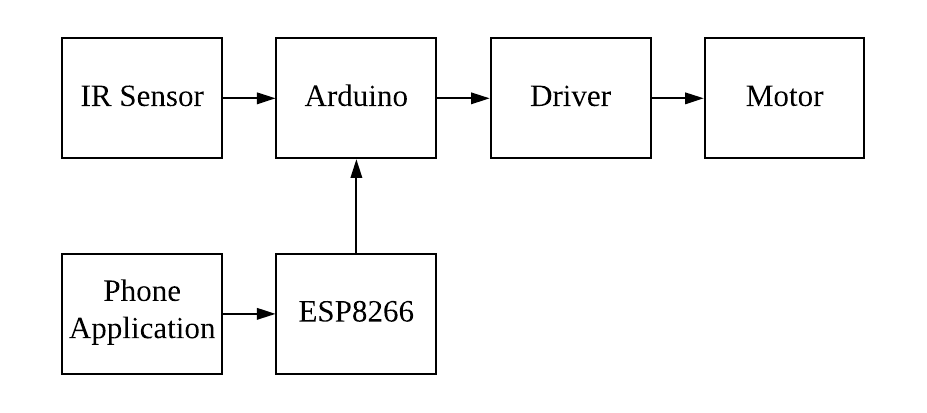
Raspberry Pi resembles a mini PC but use an OS that is developed from Linux. For the Raspberry Pi can use Python C++ to write the command.

Arduino is a microcontroller which can only work in accordance with the program that we wrote and there is no built-in OS. It is designed to be economical, small, does not require additional equipment for uploading sketches. there is also a port to connect with external sensors more than Raspberry Pi. There are various protocols which is a standard for connecting with external hardware for example I2C, SIP, UART including both Digital and Analog port. Importantly, it works more specifically type than Raspberry Pi for example real time control Arduino is more suitable than Raspberry Pi. In addition, Arduino is also designed the system to prevent over-voltage better than Raspberry Pi.

# System

## Block diagram

Figure 1: Block diagram of the system



Ultrasonic

The block diagram shown in figure 1 depicts how our system works. Firstly, the microcontroller, Arduino, will receive the data from sensors and ESP8266. Then Arduino will send it to driver. Lastly, the driver will drive the motor.

## Controller

### 

### Arduino Uno

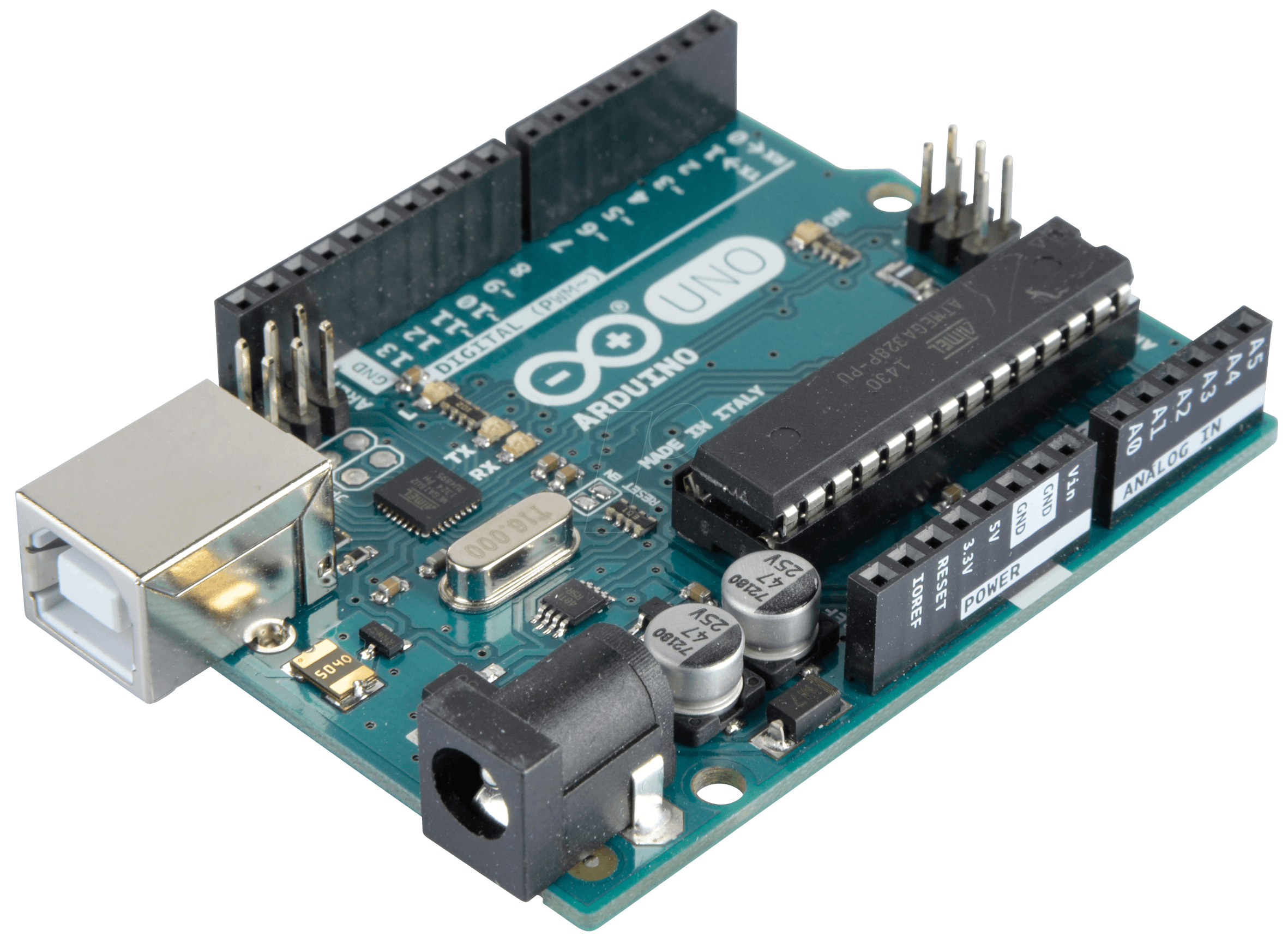


Figure 2

The Microcontroller that we are going to use is Arduino UNO since it is a beginner friendly and easily to code. Arduino UNO, operating at 5V, comes with 14 Digital input and 6 Analog input which is more than enough for this project. Using Arduino UNO make us applied what we have learned so far in the classroom into the real world. Furthermore, Arduino UNO are used in various of projects around the world so that we can have many references and support as we want. We use Arduino UNO to be the main controller.

**The table of specification**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Remarks** |
| **1** | Microcontroller | ATmega328P |
| **2** | Operating Voltage | 5 V |
| **3** | Input Voltage  (Recommended) | 7-12 V |
| **4** | Digital I/O Pins | 14 (of which 6 provided  PWM output) |
| **5** | PWM Digital I/O Pins | 6 |
| **6** | Analog Input Pins | 6 |
| **7** | DC Current per I/O Pin | 20 mA |
| **8** | DC Current for 3.3V Pin | 50 mA |
| **9** | Flash Memory | 32 KB |
| **10** | SRAM | 2KB |
| **11** | EEPROM | 1KB |
| **12** | Clock Speed | 16Hz |

### ESP8266

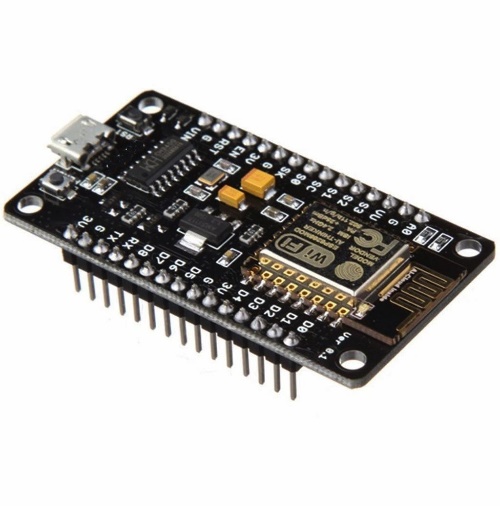


Figure 3

The ESP8266 is another microcontroller that we use in this project. This MCU has a Wi-Fi receiver so that we can communicate it via application in the smart phone. The communication will start from the application to ESP8266 via WiFi and then back to Arduino UNO via serial communication. We use this controller to connect the application to Arduino for entering commands via the application.

**The table of Specification**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Remarks** |
| **1** | Microcontroller | Xtensa Single-Core 32-Bit  L106 |
| **2** | 802.11 b/g/n Wi-Fi | Yes, HT20 |
| **3** | Typical Frequency | 80 MHz |
| **4** | SRAM | 160 KBytess |
| **5** | Flash | SPI Flash, up to  16 Mbytes |
| **6** | Operating Voltage | 2.5V ~ 3.6V |
| **7** | Operating Current | Average value: 80 mA |
| **8** | Software PWM | 8 Channels |
| **9** | GPIO | 17 |
| **10** | SPI/I2C/I2S/UART | 2/1/2/2 |
| **11** | ADC | 10-bit |
| **12** | Working Temperature | -40ºC-125ºC |
| **13** | Security | WPA/WPA2 |

## Sensors

### ผลการค้นหารูปภาพสำหรับ ir infrared obstacle detection sensor pngIR Infrared Obstacle Avoidance Sensor Module

Figure 4

IR Infrared Obstacle Avoidance Sensor Module, working under voltage of 3.3 to 5 V DC, is an optical sensor for detecting obstructions or black surfaces. Receiver will always be able to receive the light signal from Emitter transmitter if the surface is not black and the value is 0. On the other hand, emitter sent the signal to the black surface Therefore, the Receiver cannot receive the signal from reflection then the value is 1. From this project we use them to detect black line in order to make the trolley run along the black line. We use IR infrared to detect the black line is defined.

### Ultrasonic Sensor HC-SR04



Figure 5

Ultrasonic Sensor HC-SR04 is used for detecting obstacles in front to prevent accidents.We use ultrasonic to prevent collisions with objects or living things.

**The table of Specification**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Remarks** |
| **1** | Operating Voltage | 5 V DC |
| **2** | Operating Current | 15 mA |
| **3** | Operating Frequency | 40 Hz |
| **4** | Maximum Range | 4 m |
| **5** | Minimum Range | 2 cm |
| **6** | Measuring Angle | 15 degree |
| **7** | Trigger Input Pulse Width | 10 µS |

## Driver and Motor

### Motor Driver Module L298N

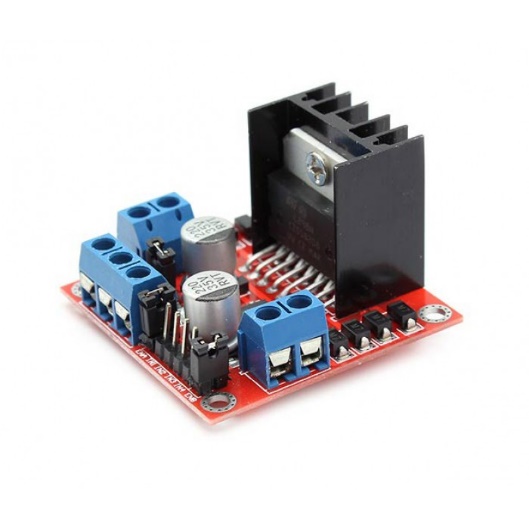


Figure 6

Motor Driver Module L298N is used for motor driving, speed control and direction control. We will use the module to control speed and direction along the black line. The operating voltage is 7 to 35 V DC and the maximum current that this driver can handle is 2 A.

### Motor



Figure 7

Motors are used for running trolley. For the project we decide to use 2 motors to run the trolley.

**The table of Specification**

|  |  |  |
| --- | --- | --- |
| **No.** | **Description** | **Remarks** |
| **1** | Operating Voltage | 3-6 V DC |
| **2** | Reduction Ratio | 1:120 |
| **3** | Output torque | 15KG |
| **4** | Output shaft diameter | 5.5 mm |
| **5** | Speed | 45±10% - 100±10% rpm |

## Miscellaneous

### Metal bar with screw hold



### Switch



### Wires



### Batteries





### Magnets



### Wheels



### 

## Mechanism Design

This is the 3D design of DIY Automatic Delivery Trolley and stations.

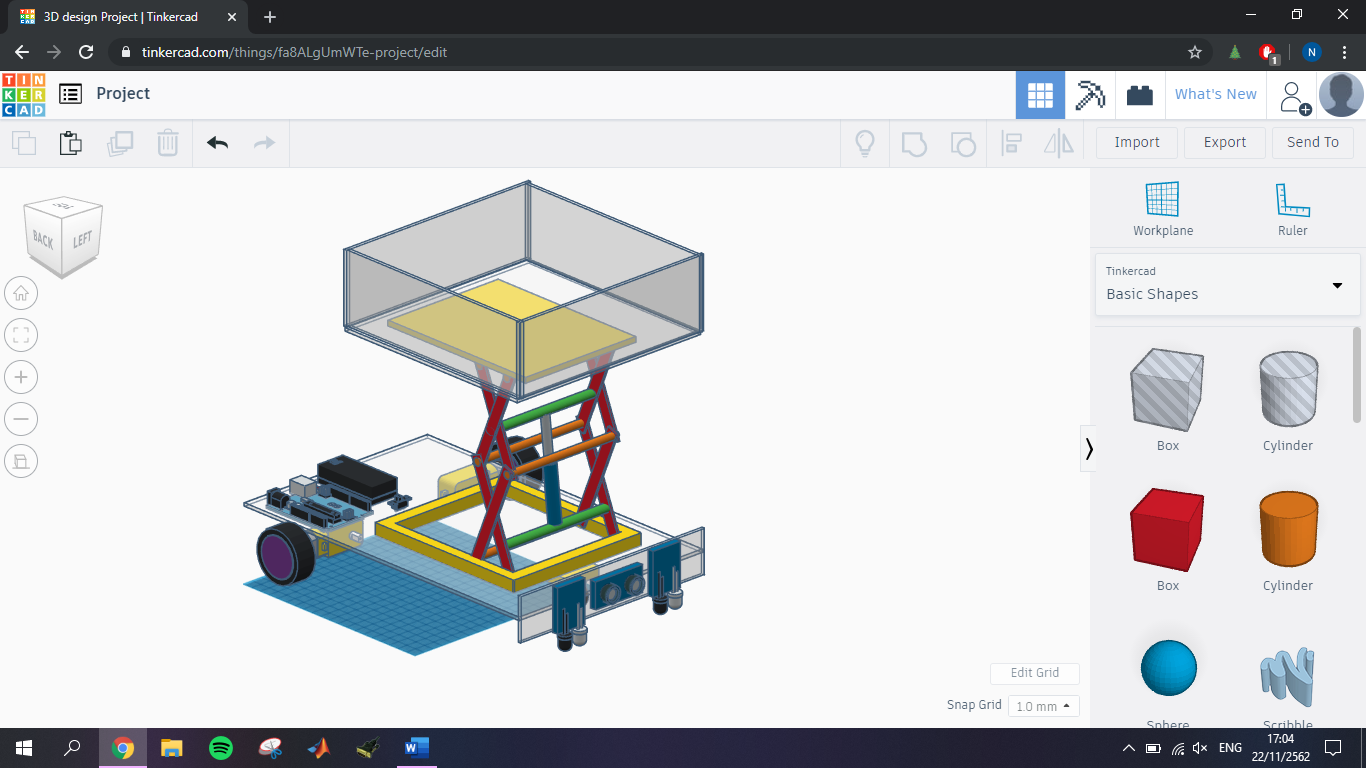
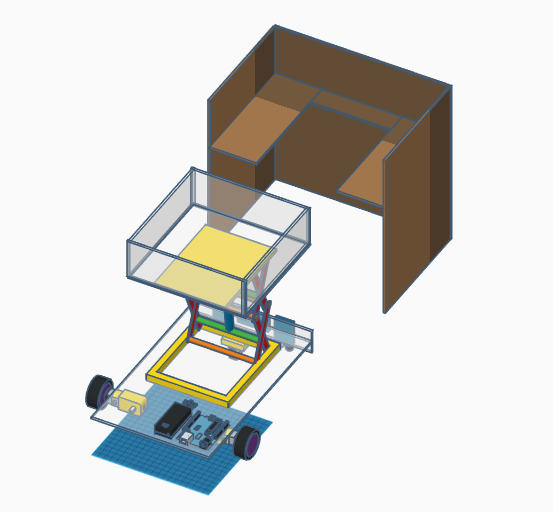


Figure 8

Figure 9

Figure 9

## Circuit diagram of DIY Automatic Delivery Trolley

## Flow Chart

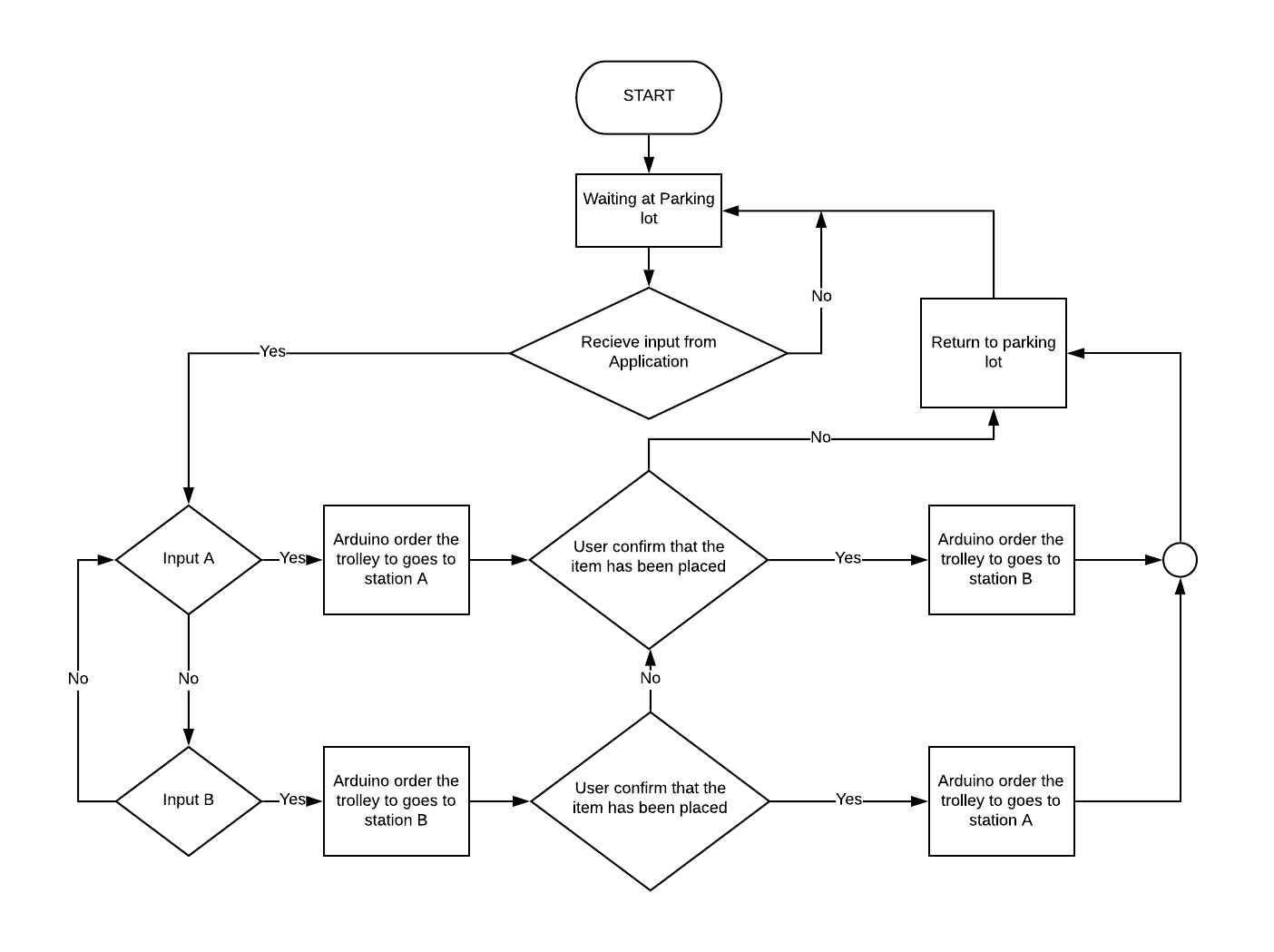


Figure 10

From the start, our trolley will wait at the parking spot. When it receives order from the application, they will go to the station according to the input which are input A goes to station A and input B goes to station B. The trolley will wait at the station for the user to confirm that the document has been placed at amount of time. If there is no confirmation, the trolley will return to the station. When the user has confirmed, the trolley will deliver the document to another station and then it will return to the parking spot and wait for another order.

# Conclusion

## Completed Work

## Future Work

# Reference

[1] How do microcontrollers work? <https://internetofthingsagenda.techtarget.com/definition/microcontroller>

[2] What is Arduino?

<https://www.arduino.cc/en/guide/introduction>

[3] How to Make Line Follower Robot Using Arduino

<https://www.instructables.com/id/Line-Follower-Robot-Using-Arduino-2/>

[4] Line Follower Robot Arduino

<https://create.arduino.cc/projecthub/robocircuits/line-follower-robot-arduino-299bae>

[5] Simple Led Control With Blynk and NodeMCU Esp8266 12E

<https://www.instructables.com/id/Simple-Led-Control-With-Blynk-and-NodeMCU-Esp8266-/>

[6] Getting Started With NodeMCU V1.0 and Blynk App

<https://www.instructables.com/id/Getting-Started-With-NodeMCU-V10-and-Blynk-App/>

[7] 3D Design and Circuit diagram of DIY Automatic Delivery Trolley

<https://www.tinkercad.com>

[8] Block diagram

<https://www.draw.io/>

# Appendix

## Motor and Ultrasonic Code

void setup()

{ Serial.begin(9600);

pinMode(10,OUTPUT);

pinMode(9,OUTPUT);

pinMode(2,OUTPUT);

pinMode(3,OUTPUT);

pinMode(4,OUTPUT);

pinMode(5,OUTPUT);

pinMode(6,OUTPUT);

pinMode(7,OUTPUT);

pinMode(8,INPUT);

pinMode(12,OUTPUT);

pinMode(13,OUTPUT);

pinMode(A0,INPUT);

pinMode(A1,INPUT);

Serial.println("Config Ready");

}

void loop() {

char DIR;

int SPD,SCA;

long duration, distance;

digitalWrite(12, LOW);

delayMicroseconds(2);

digitalWrite(12, HIGH);

delayMicroseconds(10);

digitalWrite(12, LOW);

duration = pulseIn(8,HIGH);

distance = (duration/2) / 29.1;

if (distance >= 300) {

digitalWrite(13, LOW);

//Serial.print(distance);

//Serial.println(" cm");

delay(10);

}

else if (distance >= 150 && distance < 300) {

digitalWrite(13, HIGH);

delay(1000);

digitalWrite(13, LOW);

delay(1000);

//Serial.print(distance);

//Serial.println(" cm");

delay(1000);

}

else if (distance >= 50 && distance < 150) {

digitalWrite(13,HIGH);

delay(300);

digitalWrite(13,LOW);

delay(300);

//Serial.print(distance);

//Serial.println(" cm");

delay(300);

}

else {

digitalWrite(13,HIGH);

delay(300);

//Serial.print(distance);

//Serial.println(" cm");

delay(300);

}

//////// DC MOTOR /////////

if(Serial.available()>0) {

DIR = Serial.read();

SPD = Serial.parseInt();

SCA = map(SPD,0,100,0,255);

if (DIR == 'R') {

analogWrite(9,SCA);

digitalWrite(2,HIGH);

digitalWrite(3,LOW);

digitalWrite(4,HIGH);

digitalWrite(5,LOW);

Serial.print("TURN RIGHT ");

Serial.print(SPD);

Serial.println("%");

}

else if (DIR == 'L') {

analogWrite(9,SCA);

digitalWrite(2,LOW);

digitalWrite(3,HIGH);

digitalWrite(4,LOW);

digitalWrite(5,HIGH);

Serial.print("TURN LEFT ");

Serial.print(SPD);

Serial.println("%");

}

else if (DIR == 'U') {

analogWrite(10,100);

digitalWrite(6,HIGH);

digitalWrite(7,LOW);

Serial.println("LIFT UP ");

delay(5000);

analogWrite(10,0);

}

else if (DIR == 'D') {

analogWrite(10,100);

digitalWrite(6,LOW);

digitalWrite(7,HIGH);

Serial.println("LIFT DOWN ");

delay(5000);

analogWrite(10,0);

}

else {

analogWrite(9,0);

Serial.println("INPUT ERROR");

}

}

}