ARDUINO reciever 1

void setup() {

Serial.begin(115200); // Initialize the serial communication

}

void loop() {

if (Serial.available() > 0) { // If there's data available on the serial port

String message = Serial.readStringUntil('\n'); // Read the message up to the newline character

message.trim(); // Remove any leading/trailing whitespace from the message

// Print the parsed message to the serial monitor

Serial.print("Received message: ");

Serial.println(message);

Serial.println(message[0]);

}

}

Arduino Receiver 2

#include <Servo.h>

Servo servo;

int go = 0;

void setup() {

Serial.begin(115200); // Initialize the serial communication

servo.attach(9);

}

void loop() {

if (Serial.available() > 0) { // If there's data available on the serial port

String message = Serial.readStringUntil('\n'); // Read the message up to the newline character

message.trim(); // Remove any leading/trailing whitespace from the message

// Print the parsed message to the serial monitor

Serial.print("Rec mess: ");

Serial.println(message);

go = message[0];

go = go - 48;

// Serial.print("go: ");

// Serial.println(go);

if(go == 1)

{

servo.write(90);

delay(100);

}

else if(go == 2)

{

servo.write(179);

delay(100);

}

else if(go == 0)

{

servo.write(0);

delay(100);

}

// Serial.print("go: ");

// Serial.println(go);

// go = 0;

}

if(go == 1)

{

servo.write(90);

delay(100);

}

else if(go == 2)

{

servo.write(179);

delay(100);

}

else if(go == 0)

{

servo.write(0);

delay(100);

}

Serial.print("go: ");

Serial.println(go);

// go = 0;

}

Python transmitter 1

#!/usr/bin/env python3

import time

import serial

speed = 0

motor1\_dir=1

motor1\_speed=0

motor2\_dir=1

motor2\_speed=0

control\_string = ""

control\_string += str(motor1\_dir)+"," + str(motor1\_speed) +","+ str(motor2\_dir) +"," + str(motor2\_speed) + "\n"

def initialize():

global control\_string

global motor1\_speed

global motor1\_dir

global motor2\_dir

global motor2\_speed

ser = serial.Serial(port='/dev/ttyUSB0',baudrate=115200,parity=serial.PARITY\_NONE,stopbits=serial.STOPBITS\_ONE,bytesize=serial.EIGHTBITS,timeout=0)

while True:

control\_string = str(1) +","+ str(0) +","+str(0) +","+str(0)+"\n"

ser.write(str.encode(control\_string))

time.sleep(0.1)

if \_\_name\_\_ == '\_\_main\_\_':

initialize()

Python transmitter 2

import pyttsx3

import speech\_recognition as sr

import controller as cnt

import time

import serial

import threading

control\_string = ""

control\_string += str(0)+"," + str(0) +","+ str(0) +"," + str(0) + "\n"

ser = serial.Serial(port='COM6',baudrate=115200,parity=serial.PARITY\_NONE,stopbits=serial.STOPBITS\_ONE,bytesize=serial.EIGHTBITS,timeout=0)

engine=pyttsx3.init('sapi5')

voices=engine.getProperty('voices')

engine.setProperty('voices', voices[0].id)

def speak(text):

engine.say(text)

engine.runAndWait()

# speak("This is Subodh Robot")

def takeCommand():

r=sr.Recognizer()

with sr.Microphone() as source:

print("Listening...")

audio=r.listen(source)

try:

print("Recognize.........")

query=r.recognize\_google(audio, language='en-in')

except Exception as e:

print("Try Again.......")

return "None"

return query

def send\_serial(go):

control\_string = str(go) +","+ str(0) +","+str(0) +","+str(0)+"\n"

print(control\_string)

ser.write(str.encode(control\_string))

time.sleep(0.1)

def main\_call():

go = 0

while True:

query=takeCommand().lower()

if 'first floor' in query:

print("First Floor...")

speak("First Floor...")

go = 1

send\_serial(go)

elif 'second' in query:

print("Second Floor...")

speak("Second Floor...")

go = 2

send\_serial(go)

elif 'ground floor' in query:

print("Ground floor...")

speak("ground floor...")

go = 0

send\_serial(go)

elif 'exit' in query:

break

if \_\_name\_\_=="\_\_main\_\_":

main\_call()