

**GESTURE AND VOICE CONTROL COMPUTER(by using python machine learning & ardunio)**

**REPORT OF MAJOR PROJECT SUBMITTED FOR PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF COMPUTER APPLICATION**

**NAME: -**Sanjay Ghosh **UNIVERSITY ROLL NO: -** 11701016010

**REG.NO: -** 161170510015 OF 2016-2017

**UNDER THE SUPERVISION OF**

**MRS. SATABDWI SARKAR**

**Asst. Professor**

**DEPT. OF COMPUTER APPLICATION**

**RCC INSTITUTE OF INFORMATION TECHNOLOGY**

**AFFILIATED TO MAULANA ABUL KALAM AZAD UNIVERSITY OF TECHNOLOGY**

**CANAL SOUTH ROAD, BELIAGHATA, KOLKATA-700015**

**RCC INSTITUTE OF INFORMATION TECHNOLOGY**

**KOLKATA-700015, INDIA**



**CERTIFICATE**

The report of the project titled “GESTURE & VOICE CONTROL COMPUTER submitted by Sanjay Ghosh (Roll no: - 117010160010) of MCA 6th semester 2019 has been prepare under my/our supervision for the partial fulfillment of the requirements for MCA degree in Maulana Abul Kalam Azad University Technology. The report is hereby forwarded.

----------------------------------

[Internal Supervisor]

[Dept of Computer Application]

[RCCIIT, Kolkata]

-------------------------------

[ Head of the Department]

Dept. Of Computer Application

RCC Institute of Information Technology, Kolkata

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to **Asst. Mrs. Satabdwi Sarkar**, Department of Computer Application, RCC Institute of Information Technology and whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest she took in advising us, for the books and reference materials provided for the moral support extended to us.

I am also indebted to our Head of the Department (MCA), **Mr. Arup Kumar Bhattacharya** for his unconditional help and inspiration.

Signature of the student with date:

**SANJAY GHOSH** ………………………………………

**Registration no: -** 161170510015 OF 2016-2017

**Roll no**: - 11701016010

**RCC INSTITUTE OF INFORMATION TECHNOLOGY**

**Kolkata-700015, INDIA**



**CERTIFICATE OF ACCEPTANCE**

The report of the project titled **GESTURE & VOICE CONTROL COMPUTER** submitted by Sanjay Ghosh (Roll No: 11701016010) of MCA 6th Semester of 2018-2019, is hereby recommended to be accepted for the partial fulfillment of the requirements for MCA degree in Maulana Abul Kalam Azad University of Technology.

**Name of the examiner(s) Signature with date**

* + - 1. .................................................…………………………
      2. .................................................….......................

**ABSTRACT**

The purpose of gesture and voice recognition in Computers has always been the minimization of the distance between the physical world and the digital world. The way humans interact among themselves could be implemented in communication with the digital world by interpreting gestures via mathematical algorithm. Numerous ways and algorithms have been proposed and implemented to achieve the goal of gesture & voice recognition and its use in communicating with the digital world. Gestures can be tracked using hand movements, accelerometers and more. On the hand, Voice is recognized by Google Speech Reorganization and the reply is performed by GTTS (Google Text To Speech). This project deals with the design and implementation of a gesture voice-controlled computer using Arduino Uno with ATMEGA32 processor and a laptop loaded with Windows 10 operating system along with low cost hardware requirements. The system can be broadly classified into two components: The Hardware part consisting of Arduino Microcontroller, the ultrasonic sensors HC-SR04 and a computer, preferably a laptop, and the software part consists of Arduino IDE and Python 3.7 IDLE with PyAutoGUI , Speech Recognition module installed. Keywords, Arduino, Gesture, Laptop, PyAutoGUI, Python, Ultrasonic

**index**

**TOPICS PAGE NO**

1. Introduction………………………………………………………………………… 1
2. Objective.................................................................................. 2
3. Feasibility Study........................................................................ 3
4. Purpose of the project…………………………………………………………… 4
5. Software Requirement Specification.......................................... 5
6. Hardware and Software Requirement....................................... 5
7. Design...................................................................................... 6
8. Working Process…………………………………………………………………… 12
9. Workflow diagram………………………………………………………………. 14
10. Sample Coding...................................................................... 15
11. Future scope of project…………………………………………………… 29
12. Conclusion............................................................................ 30
13. Reference............................................................................ 31

**INTRODUCTION**

Human Interface Device or HID is software application that represents information to an operator or user about the state of a process, to accept and implement the operators control instructions. It includes electronic components for signaling and controlling automation systems. The interface comprises of hardware and software that helps in communication and exchange of information between the user (human operator) and the machine.

We normally use LED indicators, Switches, Touch Screens and LCD Displays as a part of HMI devices. Another way to communicate with machines like Robots or Computers is with the help of Hand Gestures and Voice.

Instead of using any Input devices such as keyboard and Mouse, we can use our hand gestures and voice to control certain functions of a computer like play/pause any multimedia, move left/right in photo slide show, scroll up/ down in a web page , open web page , play song on YouTube , get current time , date and weather, get current news , information about anything.

**OBJECTIVE**

Recently **Gesture controlled Laptops or computers** are getting very famous. This technique is called motion which enables us to control certain functions on our computer/Laptop by simply waving our hand in front of it. It is very cool and fun to do it, but these laptops are really priced very high. So, in this project let us try building our own **Gesture and Voice control Laptop/Computer by combining the Power of Arduino and Python**

We will use one **Ultrasonic sensor** to determine the position of our hand to increase or decrease the sound and **Gyro Sensors** to determine the movement of x, y and z axis **to move the air mouse. We made our project wireless for more efficient so we can control our computer from maximum six-meter distance.**

The concept behind the project is very simple. We will place one GYRO sensor and which will help the computer/laptop work properly by understanding our hand gesture. To perform actions on our computer we used Python **pyautogui** library. The commands from Arduino are send to the computer through serial port (USB). This data will be then read by python which is running on the computer and based on the read data an action will be performed. Voice is recognized by **Google Speech Reorganization** and the reply is performed by **GTTS (Google Text To Speech)**.

**FEASIBILITY STUDY**

This section aims to judge the cost requirement and value to be obtained. The types of feasibilities we used are:

Technical feasibility: Before the project has started, we thought of the technologies we would be using. we used many kinds of sensors and a power generator and microcontroller to depict the vision we wanted to show. We used breadboard and wires with some sensors to show the physical presentation and used Arduino to perform the programming part which would connect the hardware devices and the functionalities among them.

Resource feasibility: In order to check the resources, we need to perform the task with the help from the internet to get the information related to that. We also need to keep the concept of programming and a good knowledge of microcontroller. In order to perform the above tasks, we needed some computers where we can perform the programming part and quite a number of microcontrollers.

Schedule feasibility: We almost completed our project within 3-4 months which we were given and got enough time to testing that if everything is working or not.

**PURPOSE OF THE PROJECT**

* User convenient
* User can interact with computer from a certain distance
* It also helps to entertain physically disabled people
* User can move any 3D object by moving the air mouse
* Cost effective
* Interact with virtual reality and game
* We can voice command the device to perform certain tasks.

**HARDWARE AND SOFTWARE REQUIREMENT**

**Hardware Components:**

* Arduino Uno
* ULTRASONIC SENSORS
* MPU-6050
* APDS 9000
* FEW CONNECTING WIRES
* A LAPTOP WITH INTERNET CONNECTION

**Development environment:**

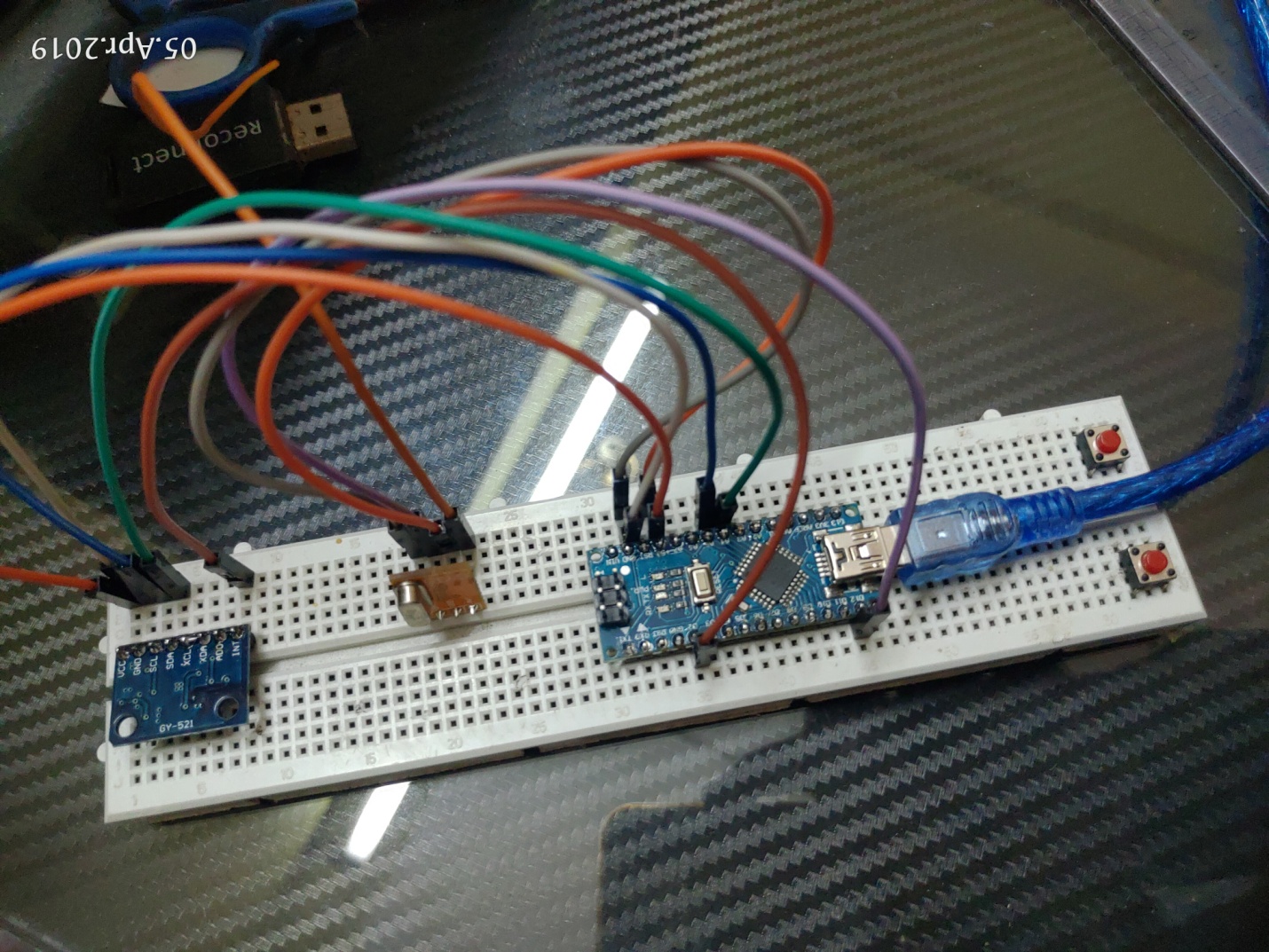
Software requirements:

* Windows 10
* Arduino ide
* Python idle

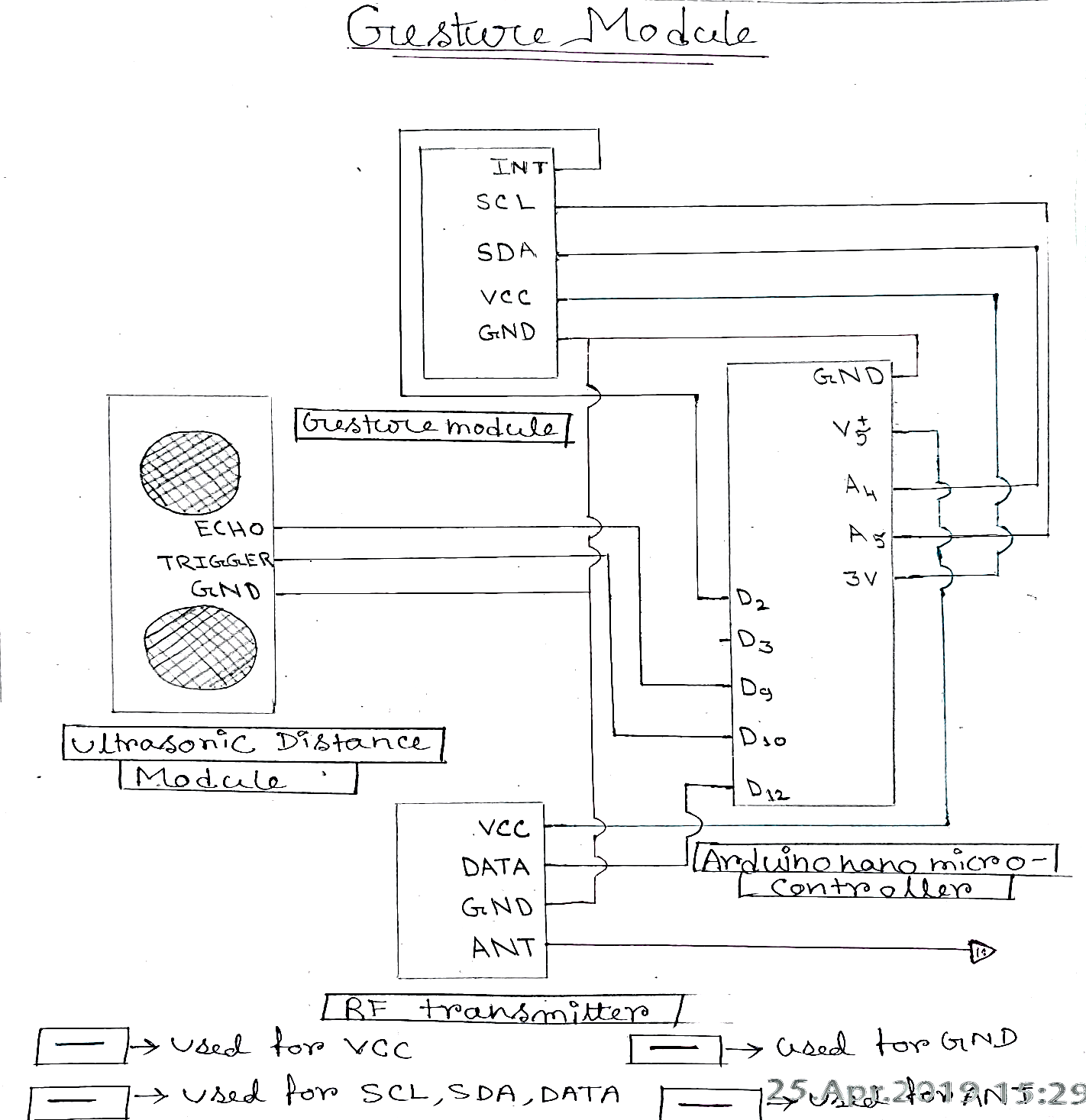
**Recommended System Requirements(min)**

* Windows 7
* 512 MB RAM
* 30 GB Hard Disk
* Processor intel i3

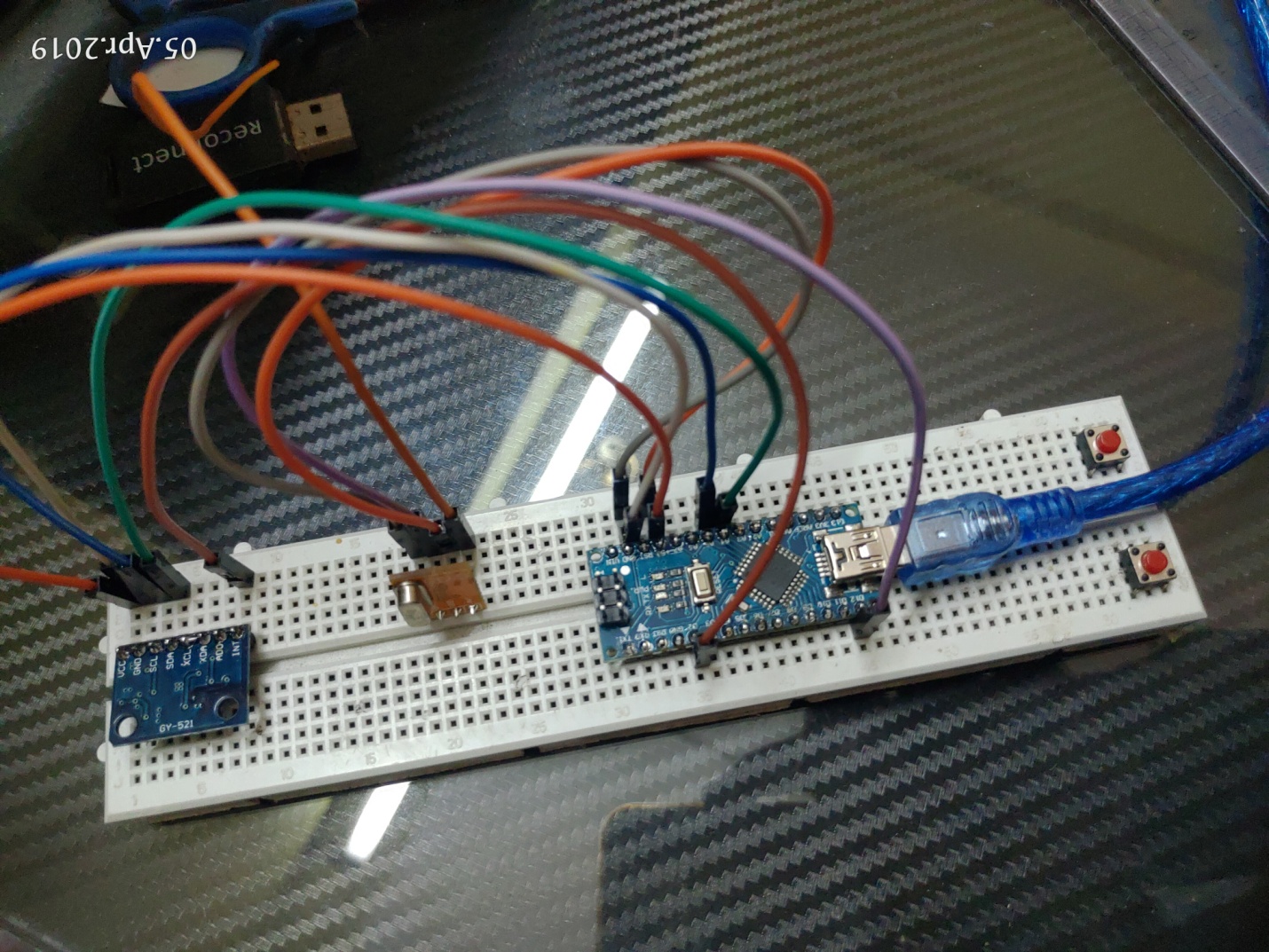
**GESTURE MODULE**



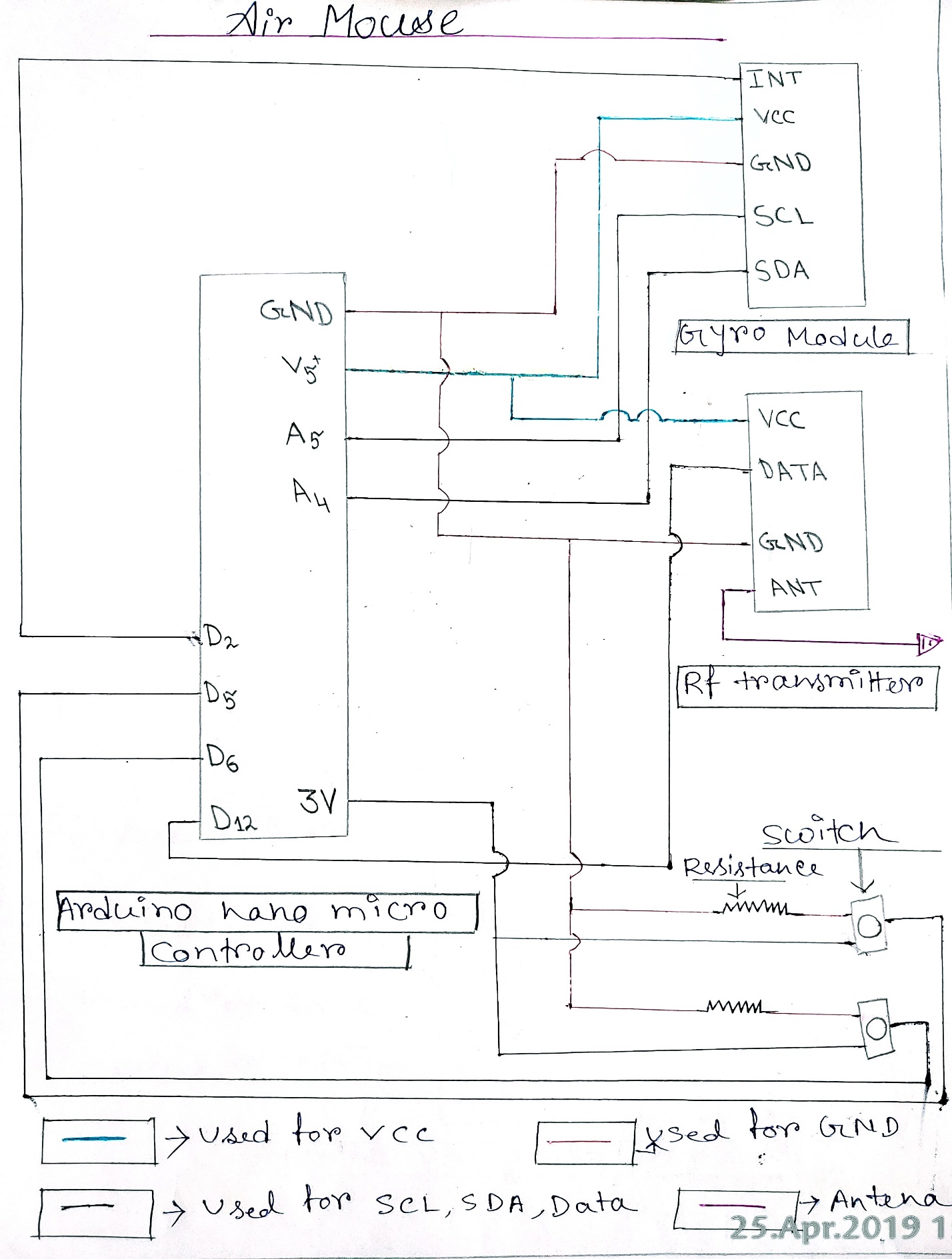
**CIRCUIT DIAGRAM OF GESTURE CONTROL**



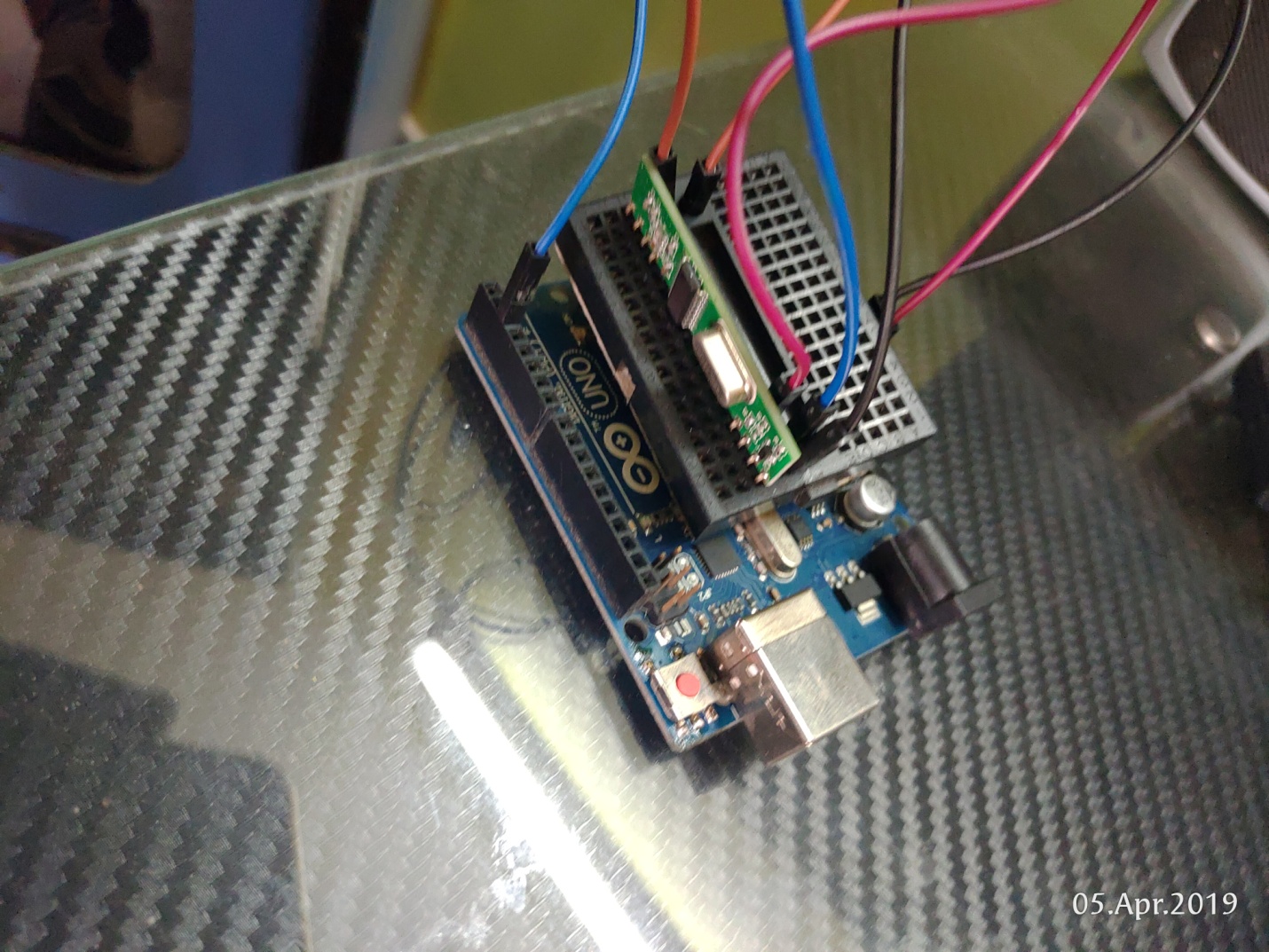
**AIR MOUSE**



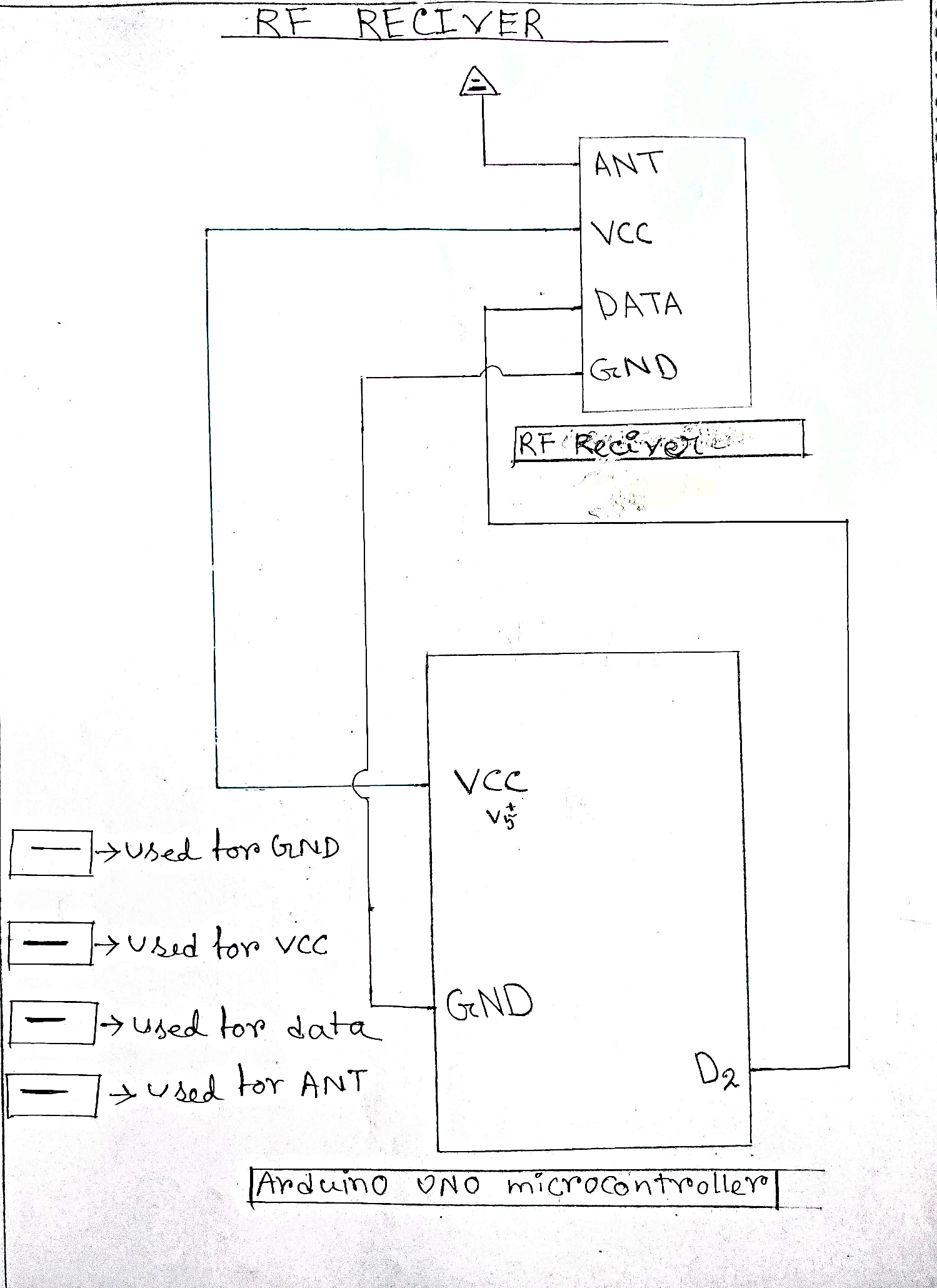
**CIRCUIT DIAGRAM OF AIR MOUSE**



**RF RECIVER**



**CIRCUIT DIAGRAM OF RF RECIVER**



**Working process**

The Gesture sensor should be programmed to read the hand gesture of users. The procedure is explained below.

In this program we have programmed some **actions** as a demo for **media player (**VLC player, Groove music, Photo).

**Action 1:**When our hands are placed the near sensor then the video in VLC player should Play/Pause.

**Action 2:**When we do left swipe above the sensor then the video should Fast Forward one step.

**Action 3:**When we do right swipe above the sensor then the video should Rewind one step.

**Action 4:**When we do down swipe and up swipe on the sensor then the next or previous track is start.

**Action 5:**When we placed our hand far from the sensor then the media shows in full screen.

In this program we also programmed some **actions** as a demo for **browsers (google chrome)**

**Action 1:** when we do right swipe above the sensor then the next page will open.

**Action 2:** when we do left swipe above the senor then the previous page will be loaded.

**Action 3:** when we do down swipe above the sensor then the page moves in downward.

**Action 4**: when we do up swipe above the sensor then the page moves in upwards.

**Action 5:** when we placed our hand far or near from the sensors then the screen will be zoom in and zoom out respectively.

In this programme we also programmed some actions as a demo for **Power Point.**

**Action 1:** when we do right swipe and left swipe above the sensor to change the slides.

**Action 2**: when we do down swipe and up swipe above the sensor to move the slides in downward and upward direction.

**Action 3:** when we placed our hands near to the sensor the ppt will start.

**Action 4:** For exit from the ppt we placed our hand far from the sensor.

In this project we also programmed for controlling the device by our **voice.**

**1. open xyz.com**: Replace xyz with any website**.**

**2.Send email/email:** Follow up recipient’s name etc**.**

**3.Current weather in (city name):** Tell us the current weather of the city.

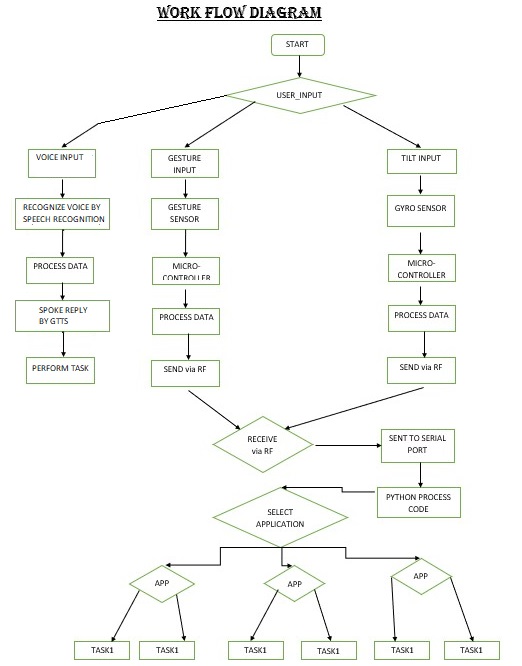
**4. Hello**: Greets user**.**

**5.News for today:** Reads top news of the day.

**6.Play a song:** Play song in you tube**.**

**7.Time:** Current system time.

**8.Tell me** about xyz: Tells about xyz**.**



Sample code

#include <ESP8266WiFi.h>

* Gesture Transmit

#include <Wire.h>

#include <SparkFun\_APDS9960.h>

#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

// Pins

//sda->A4,scl->A5 on NANO

#define APDS9960\_INT 2 // Needs to be an interrupt pin

#define Transmit 12//RF transmit pin

const int trigger1 = 9; //Trigger pin of Ultasonic Sensor

const int echo1 = 10; //Echo pin of Ultasonic Sensor

// Constants

// Global Variables

SparkFun\_APDS9960 apds = SparkFun\_APDS9960();

int isr\_flag = 0;

long time\_taken;

int dist,distL,distR;

/\*###Function to calculate distance###\*/

void calculate\_distance(int trigger, int echo)

{

digitalWrite(trigger, LOW);

delayMicroseconds(2);

digitalWrite(trigger, HIGH);

delayMicroseconds(10);

digitalWrite(trigger, LOW);

time\_taken = pulseIn(echo, HIGH);

dist= time\_taken\*0.034/2;

if (dist>60){

dist = 60;

}

//Serial.println(dist);

if (dist>1&&dist<5){

Serial.println("VOLDN");

mySwitch.send(8, 24);

}

else if (dist>8 && dist<15){

Serial.println("VOLUP");

mySwitch.send(7, 24);

}

}

/\*###Function to Detect Gesture###\*/

void handleGesture() {

if ( apds.isGestureAvailable() ) {

switch ( apds.readGesture() ) {

case DIR\_UP:

Serial.println("UP");

mySwitch.send(3, 24);

break;

case DIR\_DOWN:

Serial.println("DOWN");

mySwitch.send(4, 24);

break;

case DIR\_LEFT:

Serial.println("LEFT");

mySwitch.send(1, 24);

break;

case DIR\_RIGHT:

Serial.println("RIGHT");

mySwitch.send(2, 24);

break;

case DIR\_NEAR:

Serial.println("NEAR");

mySwitch.send(5, 24);

break;

case DIR\_FAR:

Serial.println("FAR");

mySwitch.send(6, 24);

break;

default:

Serial.println("NONE");

mySwitch.send(0, 24);

}

}

}

void interruptRoutine() {

isr\_flag = 1;

}

void setup() {

// Set interrupt pin as input

pinMode(APDS9960\_INT, INPUT);

// Ser pin for Ultrasonic

pinMode(trigger1, OUTPUT);

pinMode(echo1, INPUT);

// Initialize Serial port

Serial.begin(9600);

// Initialize interrupt service routine

attachInterrupt(0, interruptRoutine, FALLING);

// Initialize APDS-9960 (configure I2C and initial values)

if ( apds.init() ) {

//Serial.println(F("APDS-9960 initialization complete"));

} else {

//Serial.println("APDS-9960 initialization Error");

}

// Start running the APDS-9960 gesture sensor engine

if ( apds.enableGestureSensor(true) ) {

//Serial.println(F("Gesture sensor is now running"));

} else {

//Serial.println(F("Something went wrong during gesture sensor init!"));

}

mySwitch.enableTransmit(Transmit);

mySwitch.setRepeatTransmit(6);

}

void loop() {

if( isr\_flag == 1 ) {

detachInterrupt(APDS9960\_INT);

handleGesture();

isr\_flag = 0;

attachInterrupt(APDS9960\_INT, interruptRoutine, FALLING);

}

calculate\_distance(trigger1,echo1);

}

* Air Mouse Transmit

#include <Wire.h>

#include <MPU6050.h>

#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

MPU6050 mpu;

int Lclick = 11;

int Rclick = 13;

void setup()

{

Serial.begin(115200);

while(!mpu.begin(MPU6050\_SCALE\_2000DPS, MPU6050\_RANGE\_2G))

{

Serial.println("Could not find a valid MPU6050 sensor, check wiring!");

delay(500);

}

checkSettings();

mySwitch.enableTransmit(12);

mySwitch.setRepeatTransmit(5);

pinMode(Lclick, INPUT);

pinMode(Rclick, INPUT);

}

void checkSettings()

{

Serial.println();

Serial.print(" \* Sleep Mode: ");

Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");

Serial.print(" \* Clock Source: ");

switch(mpu.getClockSource())

{

case MPU6050\_CLOCK\_KEEP\_RESET: Serial.println("Stops the clock and keeps the timing generator in reset"); break;

case MPU6050\_CLOCK\_EXTERNAL\_19MHZ: Serial.println("PLL with external 19.2MHz reference"); break;

case MPU6050\_CLOCK\_EXTERNAL\_32KHZ: Serial.println("PLL with external 32.768kHz reference"); break;

case MPU6050\_CLOCK\_PLL\_ZGYRO: Serial.println("PLL with Z axis gyroscope reference"); break;

case MPU6050\_CLOCK\_PLL\_YGYRO: Serial.println("PLL with Y axis gyroscope reference"); break;

case MPU6050\_CLOCK\_PLL\_XGYRO: Serial.println("PLL with X axis gyroscope reference"); break;

case MPU6050\_CLOCK\_INTERNAL\_8MHZ: Serial.println("Internal 8MHz oscillator"); break;

}

Serial.print(" \* Accelerometer: ");

switch(mpu.getRange())

{

case MPU6050\_RANGE\_16G: Serial.println("+/- 16 g"); break;

case MPU6050\_RANGE\_8G: Serial.println("+/- 8 g"); break;

case MPU6050\_RANGE\_4G: Serial.println("+/- 4 g"); break;

case MPU6050\_RANGE\_2G: Serial.println("+/- 2 g"); break;

}

Serial.print(" \* Accelerometer offsets: ");

Serial.print(mpu.getAccelOffsetX());

Serial.print(" / ");

Serial.print(mpu.getAccelOffsetY());

Serial.print(" / ");

Serial.println(mpu.getAccelOffsetZ());

Serial.println();

}

void loop()

{

//Vector rawAccel = mpu.readRawAccel();

Vector normAccel = mpu.readNormalizeAccel();

int xval =normAccel.XAxis/3 + 5;

int yval =normAccel.YAxis/3 + 5;

int RCval = 0;//digitalRead(Rclick)

int LCval = 0;//digitalRead(Lclick)

int T = 1000\*xval + 100\*yval + 10\*RCval + LCval;

Serial.println(T);

mySwitch.send(T, 24);

//delay(1000);

//delay(10);

}

* **Receiver**

#include <RCSwitch.h>

RCSwitch mySwitch = RCSwitch();

void setup() {

Serial.begin(9600);

mySwitch.enableReceive(0); // Receiver on interrupt 0 => that is pin #2

}

void loop() {

if (mySwitch.available()) {

int value = mySwitch.getReceivedValue();

if (value == 0) {

} else {

Serial.println(value);

}

mySwitch.resetAvailable();

}

}

* Python code

import subprocess

import threading

import time

import serial

import serial.tools.list\_ports

import pyautogui

from win32gui import GetWindowText,GetForegroundWindow

from pushbullet import Pushbullet

try:

Arduino\_Serial = serial.Serial('com3',9600)

except:

print('Problem connection to Serial port')

VOLUP=7

VOLDN=8

RIGHT=1

LEFT=2

UP=3

DOWN=4

NEAR=5

FAR=6

#function for get data from serial

def getdata():

try:

value=int(str(Arduino\_Serial.readline())[2:6])

return value

except:

try:

value=int(str(Arduino\_Serial.readline())[2:3])

return value

except:

getdata()

#function for gesture

def gesture(status):

appName=GetWindowText(GetForegroundWindow()).split('- ')[-1]

#print(appName,status)

if status==VOLUP:

pyautogui.press('volumeup')

elif status==VOLDN:

pyautogui.press('volumedown')

elif appName=='Groove Music':

if status==RIGHT:

pyautogui.press('nexttrack')

elif status==LEFT:

pyautogui.press('prevtrack')

elif status==NEAR:

pyautogui.press('playpause')

elif status==UP:

pyautogui.hotkey('ctrl','T')

elif status==DOWN:

pyautogui.hotkey('ctrl','H')

elif appName=='VLC media player':

if status==RIGHT:

pyautogui.hotkey('ctrl','right')

elif status==LEFT:

pyautogui.hotkey('ctrl','left')

elif status==UP:

pyautogui.press('prevtrack')

elif status==DOWN:

pyautogui.press('nexttrack')

elif status==NEAR:

pyautogui.press('playpause')

elif status==FAR:

pyautogui.press('f')

elif appName=='Google Chrome':

if status==RIGHT:

pyautogui.press('browserforward')

elif status==LEFT:

pyautogui.press('browserback')

elif status==UP:

pyautogui.press('pageup')

elif status==DOWN:

pyautogui.press('pagedown')

elif status==NEAR:

pyautogui.hotkey('ctrl','+')

elif status==FAR:

pyautogui.hotkey('ctrl','-')

elif appName=='Power Point':

if status==RIGHT:

pyautogui.press('pagedown')

elif status==LEFT:

pyautogui.press('pageup')

elif status==UP:

pyautogui.press('pageup')

elif status==DOWN:

pyautogui.press('pagedown')

elif status==NEAR:

pyautogui.press('f5')

elif status==FAR:

pyautogui.press('esc')

elif appName=='Photos':

if status==RIGHT:

pyautogui.press('right')

elif status==LEFT:

pyautogui.press('left')

elif status==UP:

pyautogui.press('f5')

elif status==DOWN:

pyautogui.hotkey('ctrl','r')

elif status==NEAR:

pyautogui.hotkey('ctrl','+')

elif status==FAR:

pyautogui.hotkey('ctrl','-')

status=0

#function for air mouse

def mouse(value):

if value == 0:

value=1

else:

rvalue = value

a =int((rvalue)%10)

b =int((rvalue/10)%10)

c =int((rvalue/100)%10)

d =int((rvalue/1000)%10)

#print(d)

if (d == 2 ):

xReading = -30

elif (d == 3 ):

xReading = -15

elif (d == 4 ):

xReading = -5

elif (d == 7 ):

xReading = 5

elif (d == 8 ):

xReading = 15

elif (d == 9 ):

xReading = 30

else:

xReading = 0

if (c == 2 ):

yReading = 30

elif (c == 3 ):

yReading = 15

elif (c == 4 ):

yReading = 5

elif (c == 7 ):

yReading = -5

elif (c == 8 ):

yReading = -15

elif (c == 9 ):

yReading = -30

else:

yReading = 0

#print(xReading,yReading)

pyautogui.moveRel(xReading,yReading,duration=0)

#pyautogui.moveRel(yReading,xReading,duration=0.05)

#pushbullet authentication

try:

pb = Pushbullet("o.ix3hgbGjbdBaBnmY42mF3GNo3XouLS8a")

pb.delete\_pushes()#delete all pushes

except:

print('Problem connection to Messege Server')

#funttion for pushbullet shutdown

def pushbullet():

time.sleep(2)

#exeption handle for null push messege

try:

pushes = str(pb.get\_pushes()[0]).split(': ')[-1].strip('{''}'"'").upper()

print("Messege :",pushes)

except:

#print('NO Messege')

pushbullet()

if pushes=='SHUTDOWN':

pb.delete\_pushes()

subprocess.call(["shutdown", "/s"])

#print('shutdown')

elif pushes=='LOGOUT':

pb.delete\_pushes()

subprocess.call(["shutdown", "/l"])

#print('logout')

pushbullet()

#multithreading

try:

t1 = threading.Thread(target=pushbullet)

t1.start()

except:

print('Internet connection falior')

#print(pyautogui.KEYBOARD\_KEYS)

while True:

#time.sleep(5)

data=getdata()

#print(data)

if (data>=1000):

mouse(data)

else:

gesture(data)

#print('..')

**Future scope**

* The user will get more smooth performance from the device which will be calibrated by us.
* Drivers.exe file is used as our project device.
* For the betterment of security and reliability we will make our own servers to communicate the other devices.
* We gradually enhance the device so that we can use it in gaming and virtual world in future.
* We will make our project device an acceptable and proficient product.

CONCLUSION

This project enables us to control certain functions on our computer/Laptop by simply waving our hand in front of it. It is a innovative project which can limit our time consumption and can be easily accessible. So, in this project we represented the computer/laptop wirelessly and effortlessly waving our hand as well as voice command the desired device to perform certain task.

REFERENCE

* <https://www.python.org/>
* <https://www.arduino.cc/>
* <https://github.com/>
* <https://www.youtube.com/>
* <https://pypi.org/project/SpeechRecognition/>
* <https://pypi.org/project/pyttsx3/>
* <https://learn.sparkfun.com/tutorials/apds-9960-rgb-and-gesture-sensor-hookup-guide/all>
* <https://www.electronicwings.com/sensors-modules/mpu6050-gyroscope-accelerometer-temperature-sensor-module>
* <https://lxml.de/parsing.html>