GE107: TINKERING LAB
PROJECT REPORT

AIR MOUSE

GROUP 5 - TEAM 4

Yashasav Prajapati 2021CSB1143

Vavadiya Harsh 2021CSB1139

Shashank Kumar 2021CSB1131

Vaibhav Kumar 2021MCB1219

Vikalp dhalwal 2021CSB1140

Rishik Vardhan 2021CSB1142

Brief Introduction:

In this project, our main objective is to demonstrate air mouse using Arduino. Here we will use MPU6050 sensor as accelerometer and gyroscope. Air mouse is one of the rotational mouse. A Rotational mouse is a type of computer mouse which attempts to expand traditional mouse functionality. The objective of rotational mice is to facilitate three degrees of freedom for human-computer interaction. There have been several attempts to develop rotating mice, using a variety of mechanisms to detect rotation.

Some of the mechanisms to detect rotation are:

- 2 Balls and 4 sensor sets
- Mechanical ring and rotary encoder
- Gyroscopes and accelerometers

Generally air mouse is the one which adopts gyroscopes and accelerometer sensors. Here In our project we will construct an air mouse using arduino and MPU6050 sensor and use python script to parse the input data and manipulate the cursor.

In our project, we used Arduino UNO. The Arduino UNO is an open-source micro-controller board based on the Microchip ATmega328P micro-controller. It is similar to the Arduino Nano and Leonardo. The ATmega328 on the board comes preprogrammed with a bootloader that allows uploading new code to it without the use of an external hardware programmer.

In our project we basically have three push buttons in which two of them have functionality of left button and right button of a traditional mouse. And the third button is the mouse button. On the screen, cursor will move only if this button is on. And further the cursor will move accordingly as the combined setup of breadboard and MPU6050 is tilted in a direction after calibration. So these are our projects main functionalities.

Software Specifications

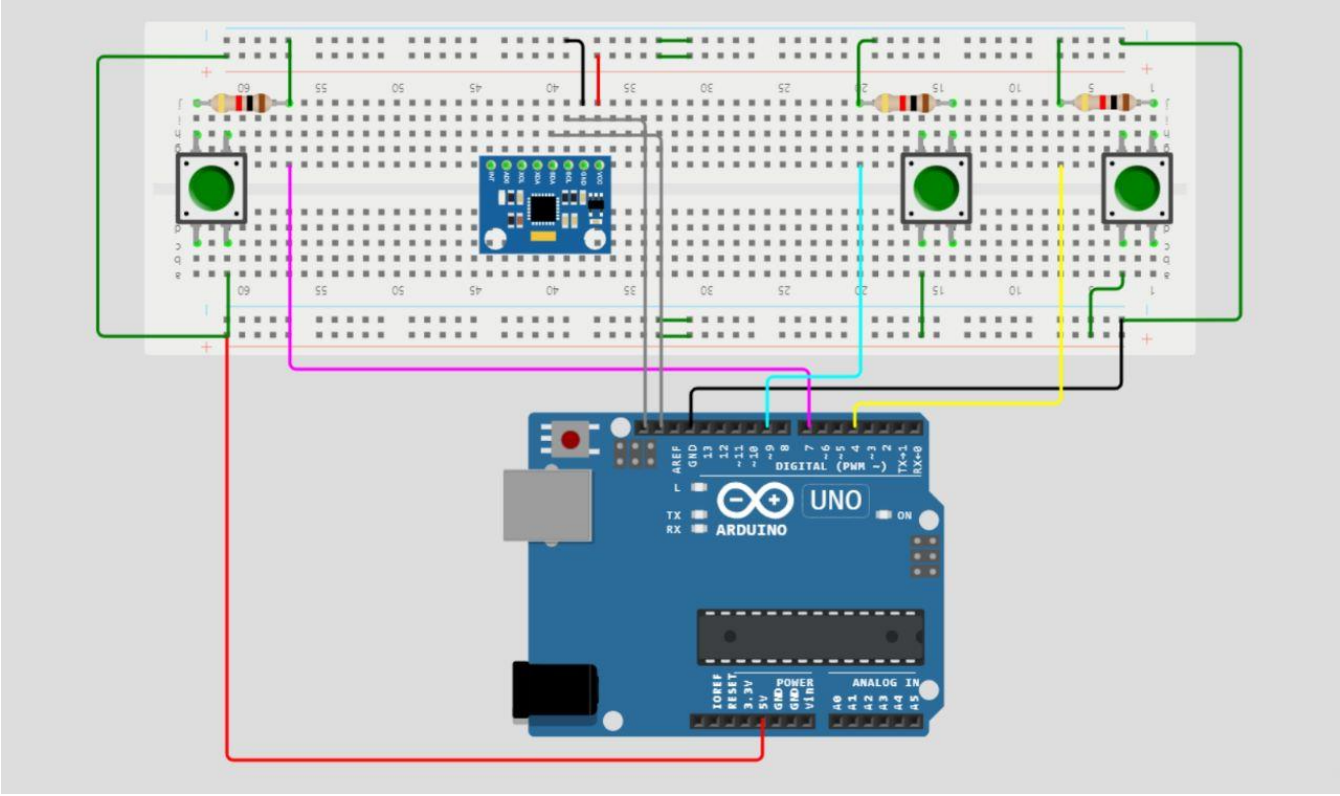
- Arduino IDE software
- I2C module for interaction between Arduino and MPU6050
- <MPU6050_tockn.h> library for Gyro communication

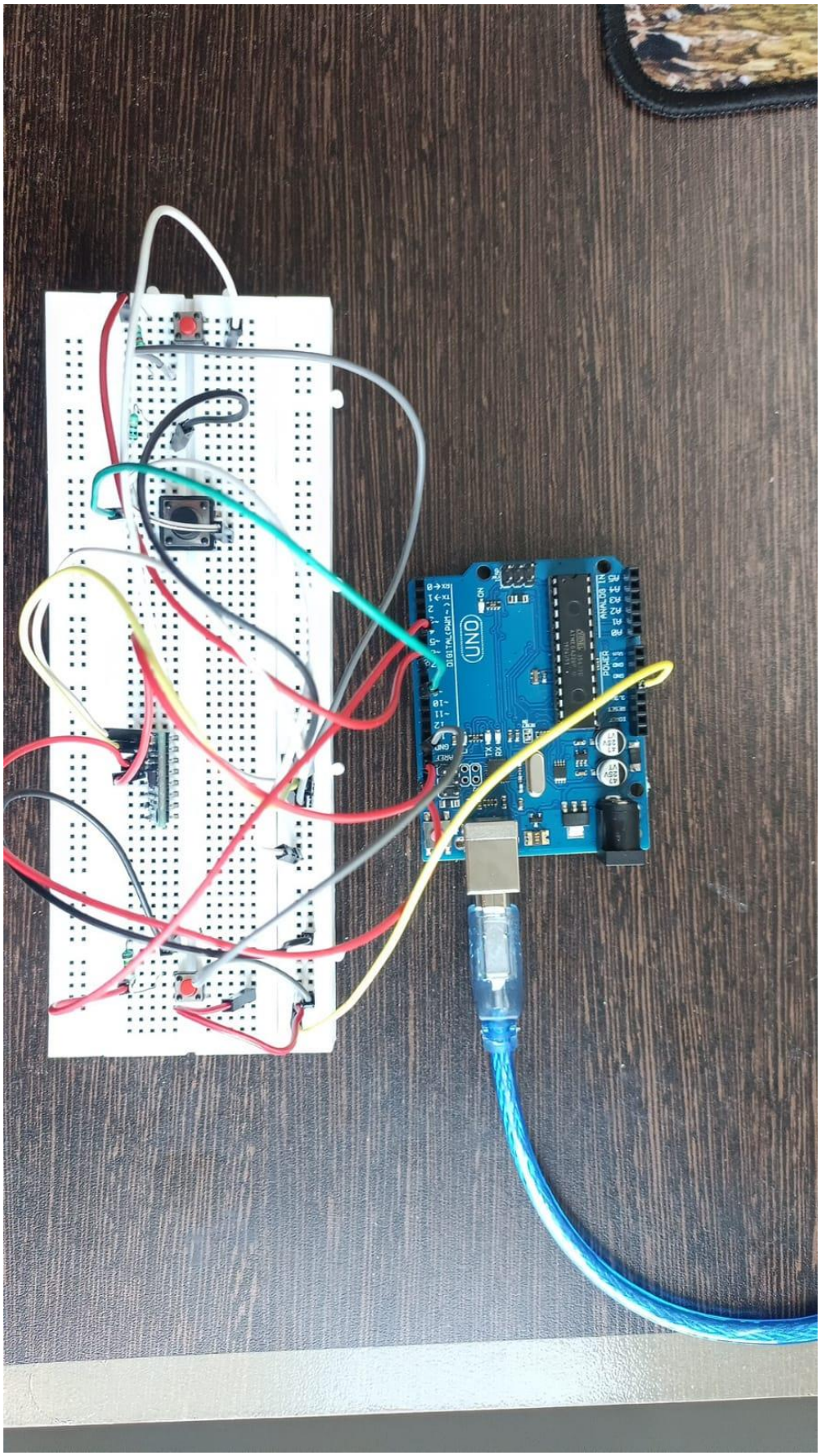
- <Wire.h> library for I2C communication

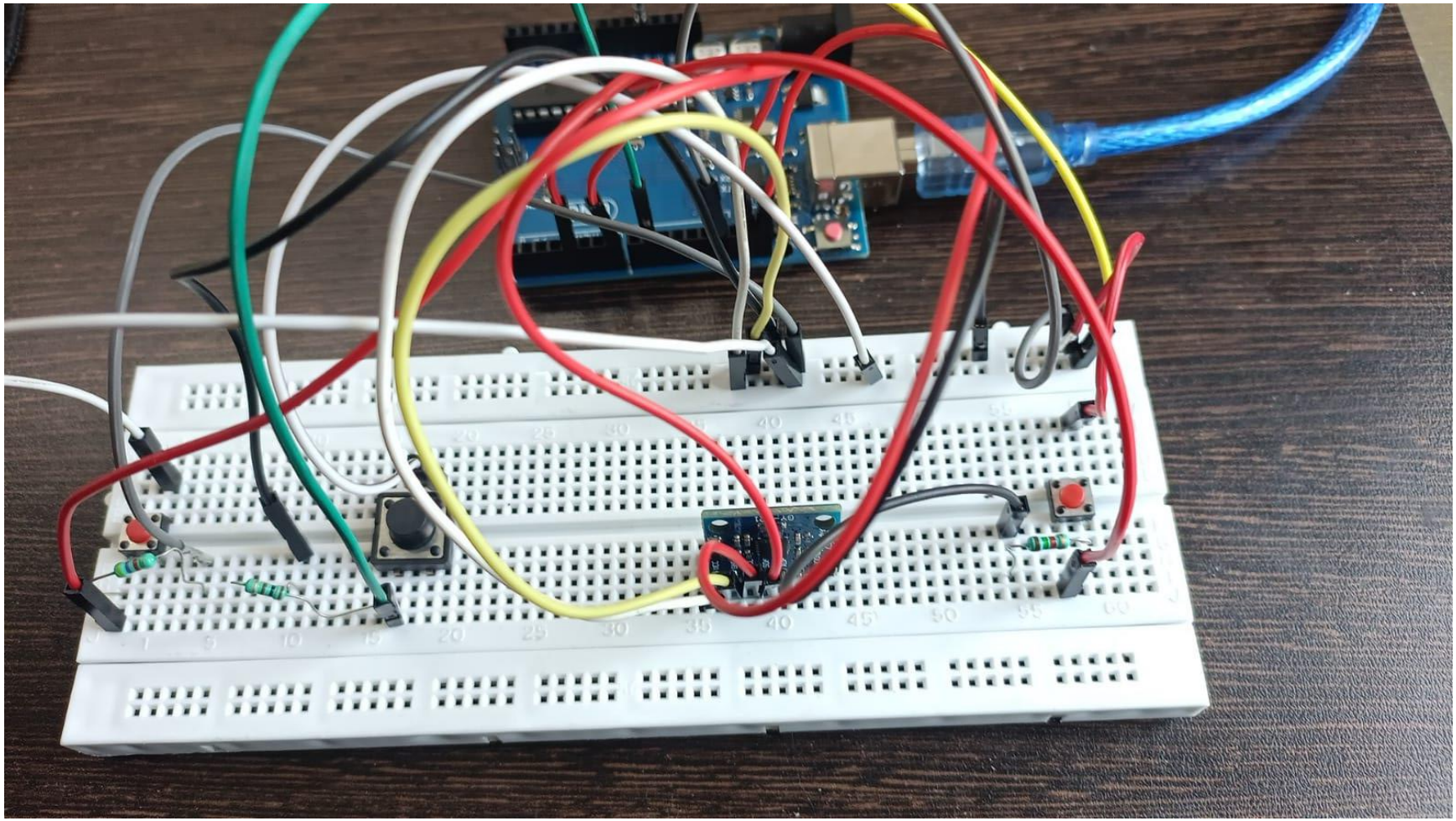
Hardware Specifications

- Arduino UNO(ATmega328P microchip)
- with USB cable
- MPU6050
- 3x PUSH BUTTONS
- Jumper wires
- Bread Board
- 3x Resistors

Circuit




UNO



Code

```
#include <MPU6050_tockn.h>   // Download and install this library https://github.com/tockn/MPU6050_tockn
#include <Wire.h>

#define LeftB     7          // Left Button Pin
#define RightB    4          // Right Button Pin
#define MouseB    9          // Mouse Enable Button Pin

MPU6050 mpu6050(Wire);
int X,Y;                     // Data Variables for mouse co-ordinates
int OX,OY;                   // Angle Variables for calucating gyroscope zero error
```

Here we include <MPU6050_tockn_n> and <Wire.h> libraries which contain functions like mpu6050.begin(), mpu6050.calcGyroOffsets(), mpu.update(), mpu6050.getAngleX(), mpu6050.getAngleY(), mpu6050.getAngleZ() etc. And defining 7 as left button pin i.e LeftB, defining 4 as right button pin i.e RightB, defining 9 as Mouse enable button pin i.e MouseB. And declared variables for mouse coordinates and gyroscope output angles.

```
void setup() {

  Serial.begin(9600);                    // Initialise Serial communication
  Wire.begin();                          // Initialise I2C communication
  mpu6050.begin();                       // Initialise Gyro communication
  Serial.println("Starting the program"); // Identifier "STARTM" for start mouse
  mpu6050.calcGyroOffsets(true);         // Setting Gyro offsets
  mpu6050.update();                     // Command to calculate the sensor data before using the get command
  OX = mpu6050.getAngleX();              // Getting angle X Y Z
  OY = mpu6050.getAngleY();


  pinMode(LeftB,INPUT);         // Setting Pinmode for all three buttons as INPUT
  pinMode(RightB,INPUT);
  pinMode(MouseB,INPUT);

  if(OX < 0){               // Inverting the sign of all the three offset values for zero error correction
    OX *= (-1);}
  else{
    OX = (OX-OX)-OX;}

  if(OY < 0){
    OY *= (-1);}
  else{
    OY = ((OY-OY)-OY)+10;}


}
```

This is the setup function, Here we first begin serial moniter and set boudrate as 9600, and then initialize I2C communication and Gyro communication using Wire.begin() and mpu6050.begin() and calibrate the sensor using

mpu6050.calcGyroOffsets() keeping the combined setup of breadboard and MPU6050 sensor flat and next calculate and store the angles in Gyro variables using mpu.update(), mpu6050.getAngleX() and mpu6050.getAngleY() functions. If we comment out the mpu.update() line then angles in OX and OY wont be updated. And after that set pinmodes to all three buttons as input and next invert the sign of all three offset values for zero error correction. This completes the setup functions.

```
void loop() {
  mpu6050.update();
  X = OX + mpu6050.getAngleX();
  Y = OY + mpu6050.getAngleY();


  if(digitalRead(MouseB) == HIGH){
    Serial.println("DataToMove,"+String(X)+','+String(Y));
    delay(100);
  }

  delay(50);
  if(digitalRead(LeftB) == HIGH){
    delay(100);
    Serial.println("DATAB,L");
  }
  if(digitalRead(RightB) == HIGH){
    delay(100);
    Serial.println("DATAB,R");
  }
}
```

And now this is the void loop part. Here in first three lines we will obtain the updated values of the angles and compute the change in mouse coordinates using getAngle and update functions. And next we will check if mouse enable button is on. And if it is on print on serial monitor "DatatoMove" and X, Y values and access these values on python script and for left click and right click we will print "DATAB,L" and "DATAB,R" respectively.

We are using python instead of C++ because arduino uno don't have <mouse.h> library. So we extract the input data on serial and move the curser or perform click operations using python.

And coming to python code,

```
import serial
from pynput.mouse import Button, Controller

mouse = Controller()
```

Here first we import serial which helps us to access data on serial. And now import Button and Controller from pynput.mouse which helps us to perform click operations and move the curser respectively. Access Controller().

```
try:

  ser = serial.Serial('/dev/ttyACM0',baudrate = 9600)

  while 1:                                                    # Whi
```

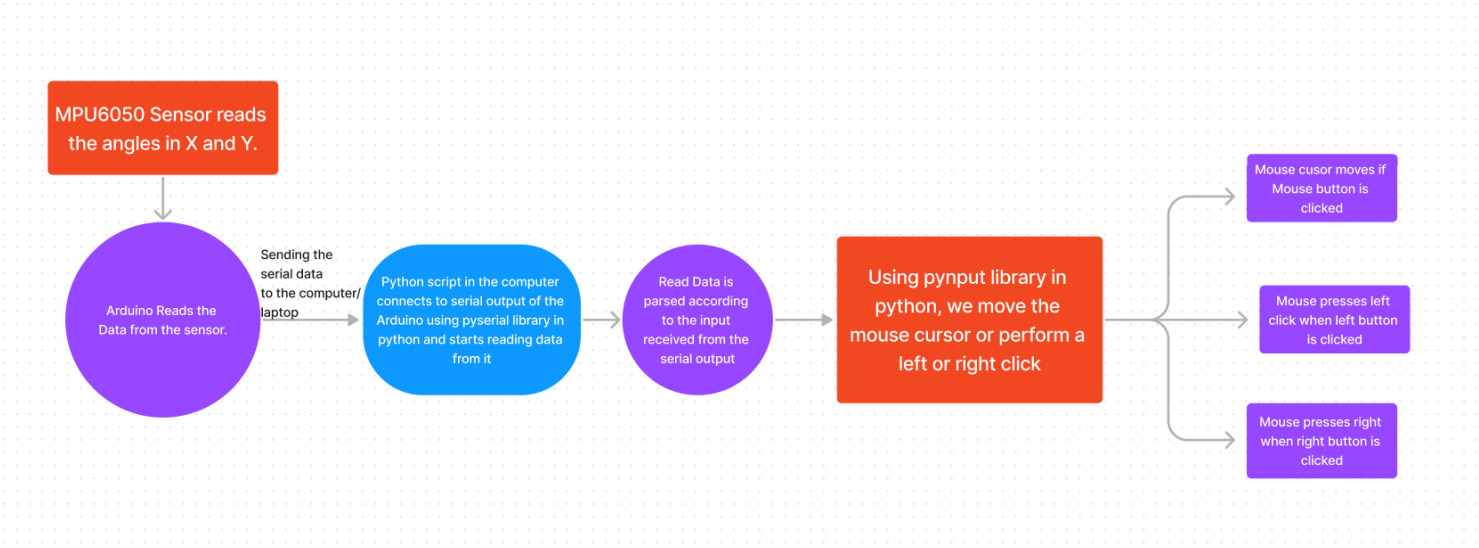
```
except:
  print("Mouse not found or disconnected.")
  k=input("Press any key to exit.")
```

Here we use try and except. If there is an error in try part it wont excecute it and go to except part and print “Mouse is not found or disconnected”. And in try part First we set the serial port number and Baudrate as 9600 i.e the baudrate we set in serial.begin() in our ino code. If we set a different value, then we may observe an unexpected behavior.

```
while 1:
    dump = ser.readline()
    dump = str(dump)
    dump = dump[2:-5]
    data = dump.split(',')
    print(data)
    if data[0] == "DataToMove":
      mouse.move(int(data[1]), int(data[2]))
    if data[0] == "DATAB":
            if data[1] == 'L' :
              mouse.press(Button.left)
              mouse.release(Button.left)
            if data[1] == 'R' :
                    mouse.press(Button.right)
                    mouse.release(Button.right)
```

In this part we are keeping an infinite while loop. Next we are reading from the serial port and converting byte data into string. And next dump = dump[2:-5] will clean the raw data from the serial port by deleting the unwanted first two characters and last five characters. Next data = dump.split(',') will split the data in the dump according to the comas. And store it in the array of strings "data". Next we try to perform the operations according to the strings in "data". If data[0] is "DataToMove", then move the curser using mouse.move(_,_) by the strings in data[1] and data[2] (after converting the numbers from strings to int. And if data[0] is "DATAB". consider the character data[1], and if it is 'L', then perform left click using Button i.e by the commands, mouse.press(Button.left) and mouse.press(Button.left). and if data[1] is 'R' then perform right click using Button i.e by the commands, mouse.press(Button.Right) and mouse.press(Button.right).

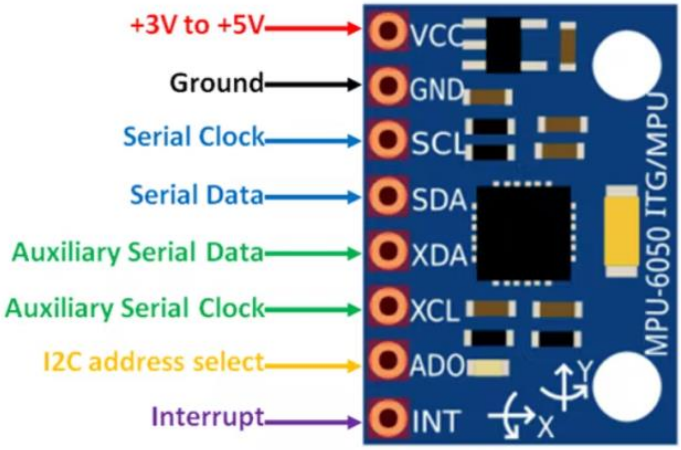
Data-Flow Diagram



Components Information

MPU6050 - MPU6050 is a 6-axis motion tracking devise. MPU6050 is a Micro Electro-mechanical system (MEMS), it consists of three-axis accelerometer and three-axis gyroscope. It helps us to measure velocity, orientation, acceleration, displacement and other motion like features. MPU6050 consists of Digital

Motion Processor (DMP), which has property to solve complex calculations. MPU6050 consists of a 16-bit analog to digital converter hardware. Due to this feature, it captures three-dimension motion at the same time. For interfacing with Arduino, it uses the I2C module. Its main feature is that it can easily combine with accelerometer and gyro. It has 8 pins but our project we only need 4. They are VCC, GND, SCL and SDA.



Testing Details

We tested the air mouse with our laptop, the speed/sensitivity can be easily modified by changing the delay in the .ino file. The initial calibration done was good enough that pressing the Mouse button did not move the cursor if the air mouse was not moved. The delay in clicking and actually performing the click in the system was negligible.

Conclusions and future improvements

By this course and air mouse project, we learned about arduino, its IDE environment and its syntax sensors, we learned about MPU6050 sensor, I2C communication, outputs of MPU6050 sensor etc. Basically, we learned that generally in dealing with arduino and sensors we obtain our output values by sending a particular input and manipulate the outputs and use it in our desired project accordingly. And coming to air mouse, we basically learned the technological history of human computer interaction developed from the traditional mouse. There are several ways for human computer interaction in the modern era of electronics. In the present day, computers motion recognition is used very efficiently for playing games. The work done in this paper presents a simple and

low-cost device for the movement of cursor on computer screen or to rotate the three-dimensional images. For the movement of cursor the data of accelerometer sensor (according to hand movement) is fed to the controller unit and after processing it is sent serially to the computer. The developed device is used for all applications as accomplished with mouse. The proposed device senses the end user action with the help of accelerometer and push buttons. The air mouse is comfortable to handle, and does not considerably obstruct entering. It functions completely as serial mouse available in the market. This innovative approach improves the end users experience with day-to-day task and playing games in computer.