Three Mouse Functions-

1_) Mouse Movement – Move The Curser To any Direction on the screen

2) Left Click - Used To Choose Files, Open Them

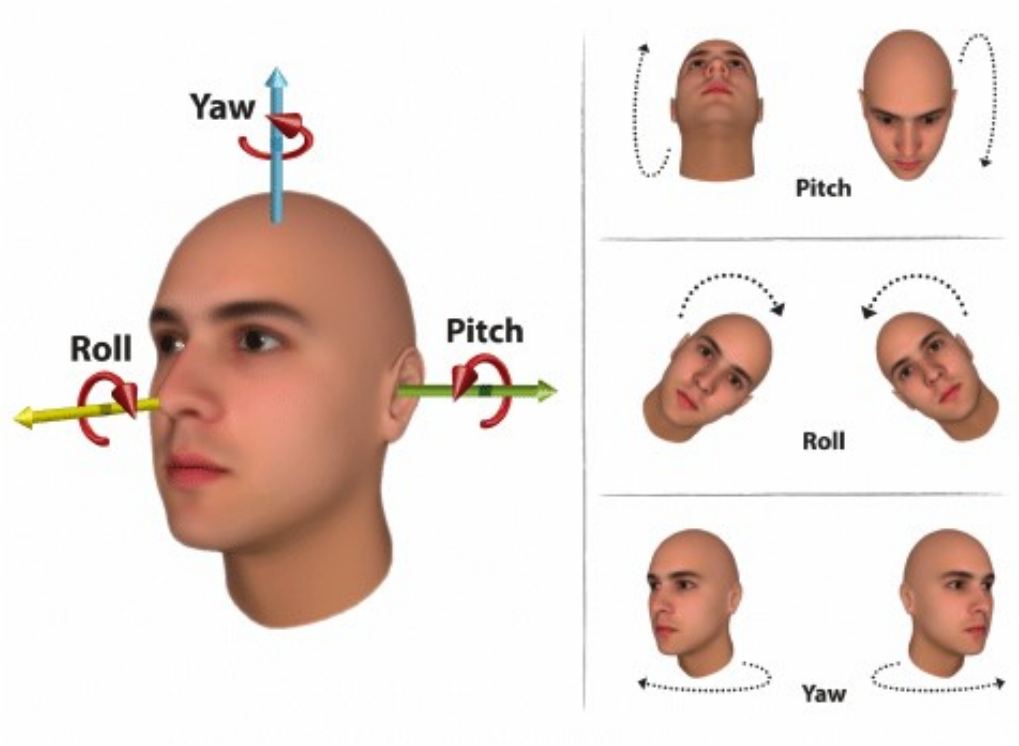
3) Right Click – used to Open File Opens (Rename, Detele, Move)

	FUNCTION	**TRANSLATION**
1)	Mouse Movement	Movement of the Head
2)	Left Click	Single Blow on Mic
3)	Right Click	Long Blow on mic(3Sec)

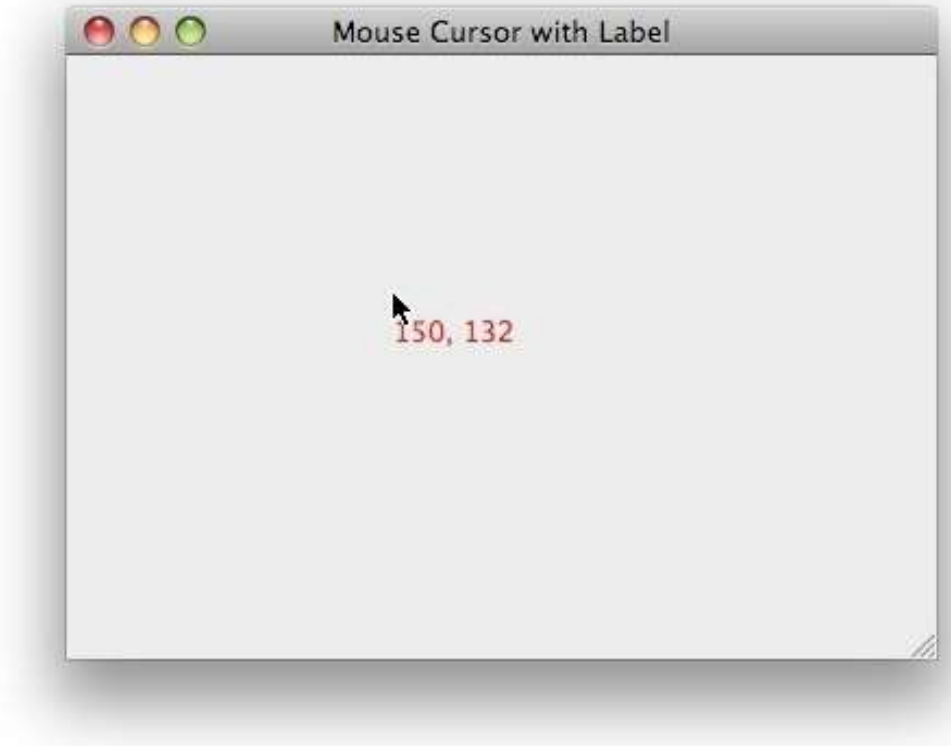
ALGORITHM REQUIRED OPERATIONS:

1)Addition
2)Subtraction
3)Multiplication
4)Division

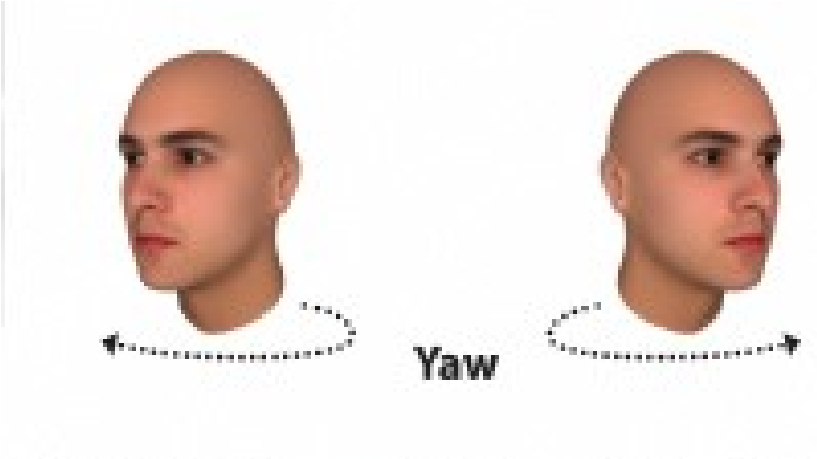
Function 1) MOVING CURSER – HEAD MOVEMENT



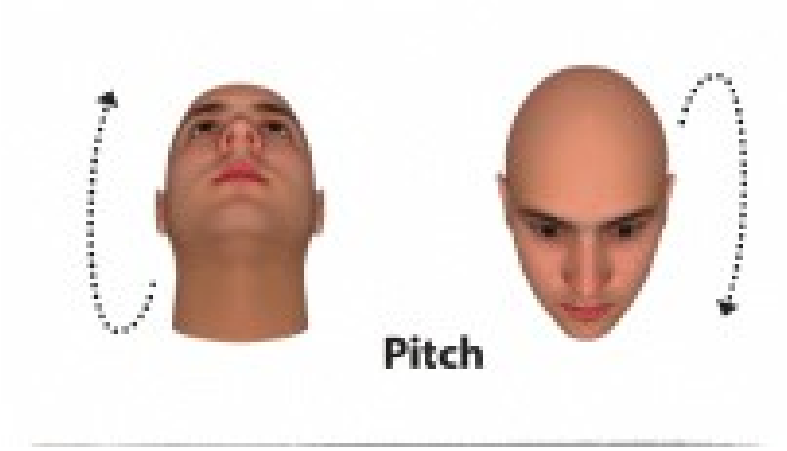
i. OUR HEAD MOVEMENT DIRECTIONS

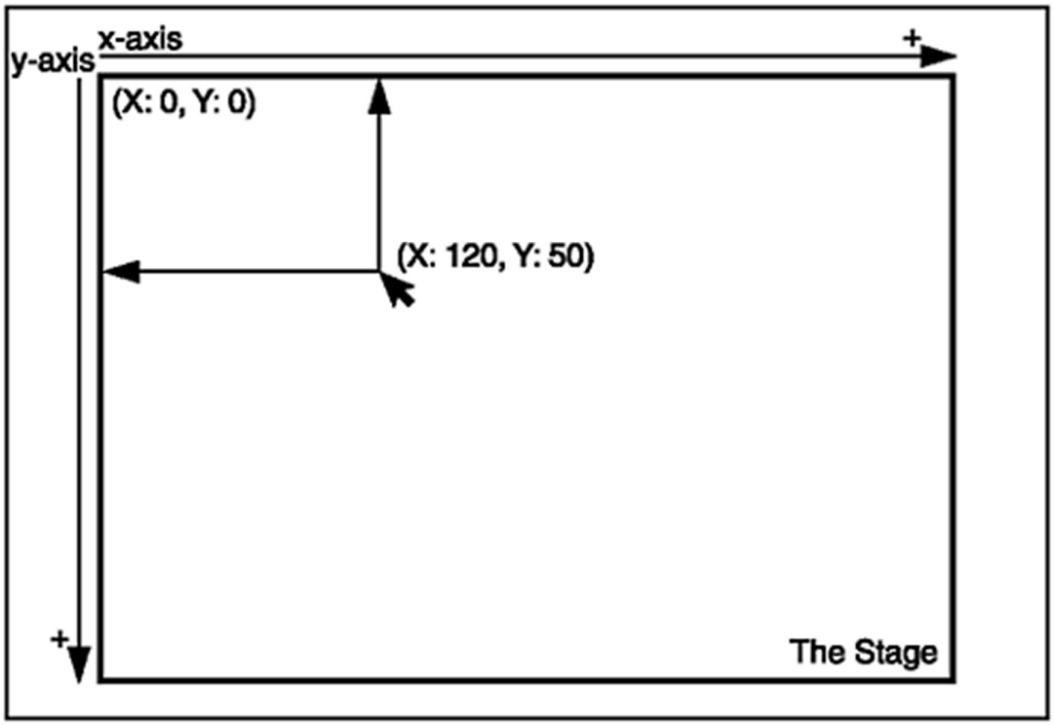


X AXIS:



Y- AXIS :

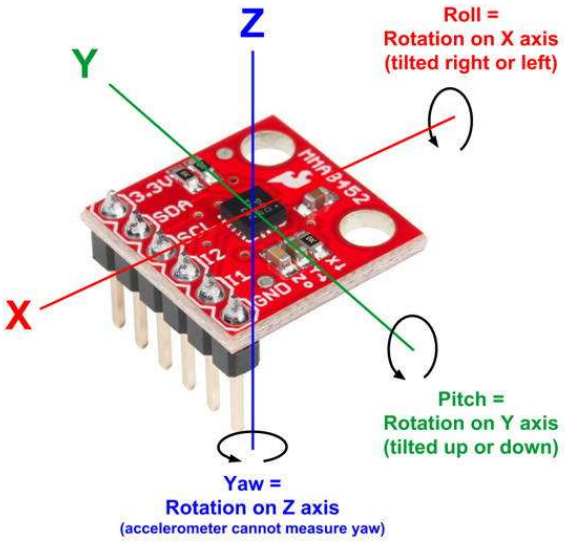


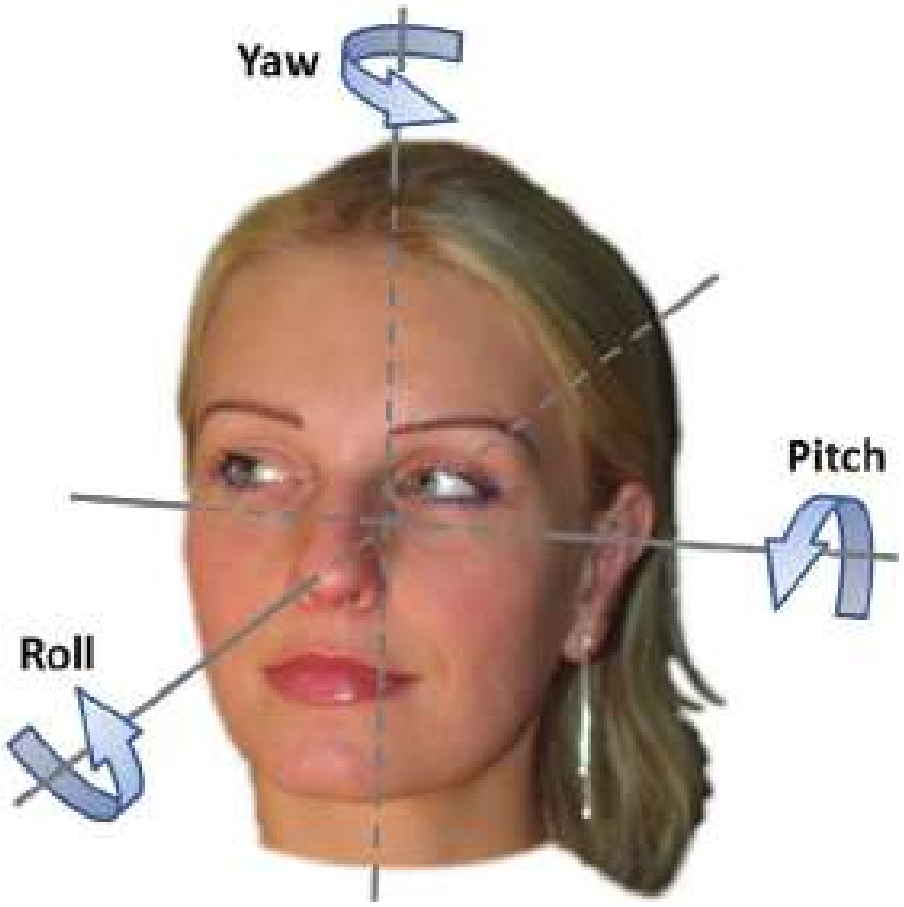


HOW DO WE MEASURE THIS MOVEMENT?

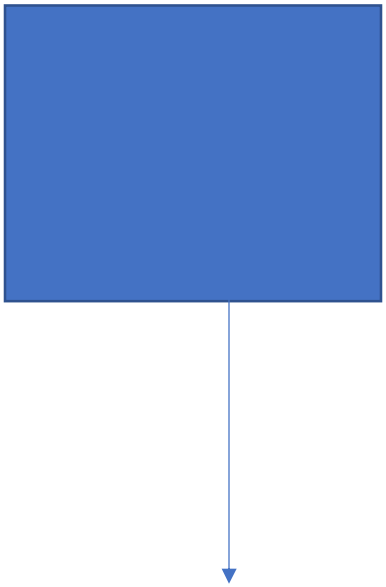
OUR SENSOR:

Accelerometer::





1) STICKING THE SENSOR TO YOUR HEAD – AS THE HEAD MOVES THE SENSOR MOVES ACCORDINGLY.
2) IT MEASURES THE CHANGE IN ITS MOTION ACROSS THE X AXIS AND Y AXIS.
3) HEADMOVEMENTS IS MEASURED



X,Y,Z

(3 NUMBERS, EVERY 1 MILLISECONDS)

EXAMPLE OF MOVING OUR HEAD IN 5 SECONDS

1)MOVE OUR HEAD UPWARDS –

(X,0,Z) -> (x,20,Z)

2) MOVE HEAD TO RIGHT

(0,Y,Z) - >(30,Y,Z)

ANALYSING ACCELEROMETER DATA-

FORMAT:(three Numbers)

+X, + Y, + Z

1)MOVING OUR HEAD UPWARDS: Y Changes WITH + SIGN

2) Moving HEAD DOWNWARDS: Y Changes WITH – SIGN

3) Moving Head To the RIGHT: X Changes WITH – SIGN

4) MOVING Head To LEFT: X Changes WITH + SIGN

ALGORITHM:

1) **Input Variables X = current X value of sensor ;Y=current Y value of sensor to the current sensor values.**
   Initialise prevX = 0; prevY=0;
2) **Process accelX = (X – prevX)/2;**
   Process accelY = (Y – prevY)/2;

3) **Processing the Scale: accelX = accelX * scalerX;**

   accelY = accelY * scalerY;

   //AccelX and AccelY are the final Standard Mouse Coordinates

```
int x, y;
float Result;
void main(){
printf(“Enter x”);
scanf(“%d”, &x);
printf(“Enter y”);
scanf (“%d”, &y);//INPUTING

Result = x+y; //PROCESSING
Result = Result * 2;

Printf(“THE SUM IS
%d”,Result);//OUTPUTING



}
```

Formulas:

Acceleration =

Change In Speed = $\frac{\text{Final Speed – Initial Speed}}{\text{TIME}}$

Angular Acceleration =

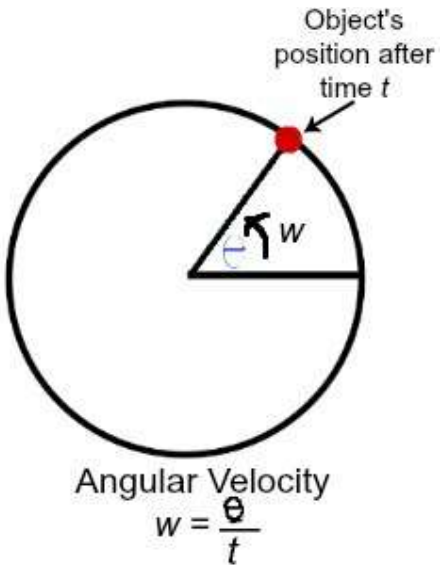
Change In Angle = $\frac{\text{Final Angle - Initial Angle}}{\text{Time(2)}}$

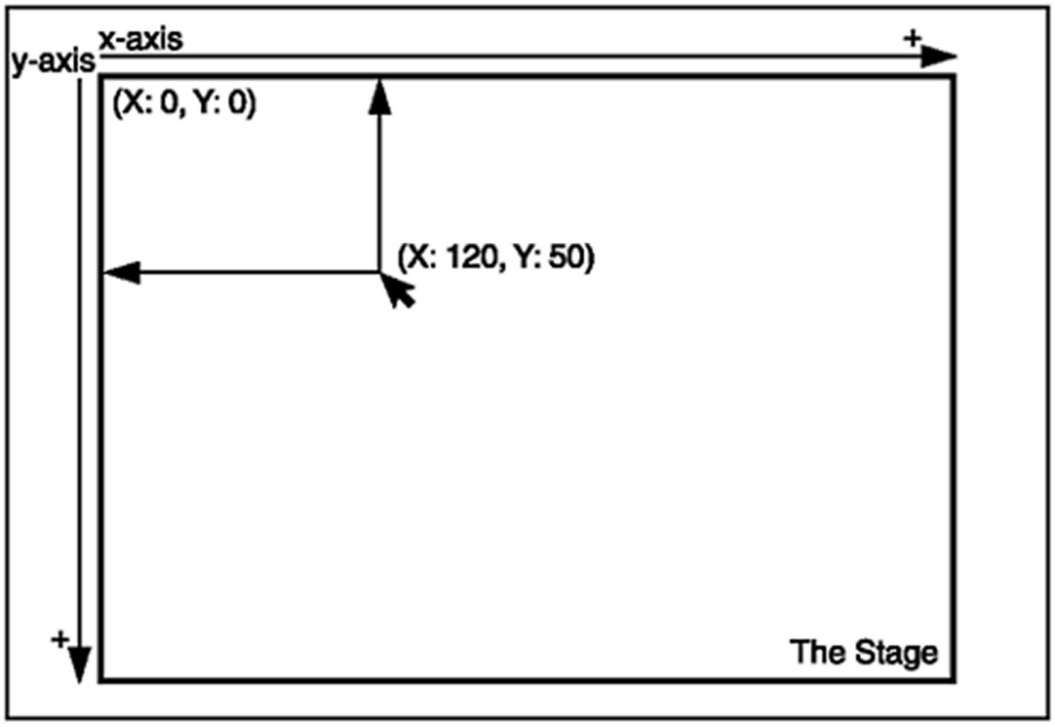
Example: A object moves from 20km/sec to 40km/sec in 5 seconds. Then the change in speed or acceleration:

$$\frac{40 - 20}{5} = 5$$

Example: A object changes angle from 60deg to 20deg in 5 seconds. Then the change in speed or acceleration:

$$\frac{60 - 20}{5} = 8$$





THE PROCESSING:

1 Case: The Person Moves His Head To the Right in 2 Sec.

2 Case: The person Moves His to the Right in 5 Seconds.

Initialise Variables:

Int X – Current angle of our Head in X Axis

Int Y – Current Angle of Our Head In Y Axis

Int prevX – Last angle of our Head in X Axis

Int prevY – Last Angle of our head in Y Axis

Int accelX – acceleration in X Axis

Int accelY – Acceleration in Y Axis

Constant scalerX = 10;

Constant ScalerY = 8;

Data From Sensor: (Sample)

0 , 0

0, +20 – (Moved Our Head Up)

-30, -10 – (Cross Downwards)

Data<0 -> Negative (if(x<0) Else if(x==0) else(

Data>0 - > Positive

1Step: prevX = 0; prevY = 0;

x = 0; y = 20;

accelX = (0 – 0)/2 = 0/2 = 0;

accelY = (20 – 0)/2 = 10; //accelY = (-20-0)/2 = -10

prevX = x = 0; prevY = y=+20;//This Gives Data For Next

accelY = 10 x **ScalerY = 100**;//Translation of Head Angles to Pixels

accelX = 0 x **ScalerX = 0;**

Coordinates – (0,100)

Case

step 2:

prevX = 0; prevY =+ 20;

x = -30; y = -10;

accelX = (-30-0)/2 = -15

accelY = (-10 – 20)/2 = -15

Understanding Computer Scales And Pixels:

1080 Pixels

720 Pixels

480 Pixels

- Our Computer Takes the coordinates of the curser with Pixels Unit;
- A Regular Computer Has Pixel Lengths upto 1000 to 800 Pixels

Data Of Sensor in a range of time:

0,0
0,0
0,0
0,0
0,0
0,0
0,10
0,10
0,10
0,0
0,-10
0,-10
5,0
4,0

4,0
3,0

Function 2 and 3: Left Click And Right Click

We use our mouth to make a blow on the mic to make a Left click and Right Click on the Screen.

Left Click: A Short Blow produces a left click

Right Click: A Long Blow Produces a Right Click.

How Do We Capture this Blow:

Sensor: ***AUDIO SENSOR with MIC***

This sensor captures the sound in the mic and gives the loudness in scale of 0 – 1024 numbers.

The Number Increases with Loudness.

1) Whether we are blowing or not.
   Sol: If the Value is Above 600, we are blowing on mic.
   If the value is belove 600, then we are not blowing.
   Threshold Value = 600 This helps in differentiating.
2) Whether the blow was long or short for left click or right click.

Sol: if the interval of change in values is more, or the set of values above 600 is more, the blow is longer.

If the set is between 3 and 5, then it is a short blow for LEFT CLICK.
If the set is between 5 and 10, the it is a long blow for Right Click.

0

0

0

0

620

605

650

0

0

0

0

620

605

650

640

630

670

650

0

0

0

0

0

THE
PROJECT:

The parts That Provide Data as **Input**

1) Accelerometer - Movement of our Head
   MPU6050 Accelerometer + Gyroscope
2) Mic Sensor – Duration and Blowing Status
   RM008 SOUND SENSOR

Processing UNIT

1) **ARDUINO PRO MINI – uses Microcontroller Atmega328 - FOR THE MOUSE UNIT**

2) **ARDUINO MICRO – RECEIVER UNIT**

Communication Part:

Bluetooth Module:

HC05 x 2 Units

One is connected to the mouse, another is connected receiving Side.

WireLess Mouse Has Two Parts:

a) **The Mouse**

b) The Receiving Pendrive That is connected to the computer. - Receiver

Parts Of The Project:

The Mouse Module:

1) MPU6050 – Accelerometer
2) RM008 – Sound Sensor
3) Arduino Pro Mini – Microcontroller
4) HC05 – Bluetooth Transmitter
5) Power Source/Battery – 5V Output
6) Over-The-Ear Headphone

The Receiver module:

1) Arduino Micro – Microcontroller
2) HC05 – Bluetooth Receiver
3) USB CABLE


L

SOFTWARE: ARDUINO IDE

Language: C

Program Structure:

```
#include
//declartions
void setup() {
//RUNS AT START
}
void loop() {
 //RUNS AFTER START IN LOOP
//RUNS 1000 Times per second
}
```

```c
#include <stdio.h>
#include <mouseproject.h>
Int xglobal;
Float y;

Void function_name(int x, int y);

int main() {

   printf("Hello, World!");


   }

Void function_name(int x, int y){
Int z;
Z = x+y;
Printf(“%d”, z);
}
```

ARDUINO FUNCTIONS/MICROCONTROLLER FUNCTION:

- pinMode(2, OUTPUT); //setsup the mode of the given pin
- digitalWrite(2, HIGH);//SWITCH FOR PIN
- delay(1000);
- analogRead()

step1: led_2 should be on, led_3, Led_4 off

step2: wait .5 second

step3: led_3 on, led_2 led_3,off

step4: wait .5 seconds

step5: led_4 on, led2, led3 off

step6:wait 0.5 seconds

-All MICROCONTROLLERS HAVE A RUNNING FREQUENCY THAT OPERATE IN CYCLES

- 16 MHz

```
#include <stdio.h>
Int x;
Int main(){ //void setup
Printf
Scanf
file
While(1){//void loop
Printf(“hi”)
```

}

}

Project Electronics:

Preconcepts:

VCC – + ve INPUT POWER pin

5 volts

GND - -ve Input power

TXD-

RXD- COMMUNICATION LINES for Bluetooth

A0 – ANALOG PINS – Read Data From Mic Sensor

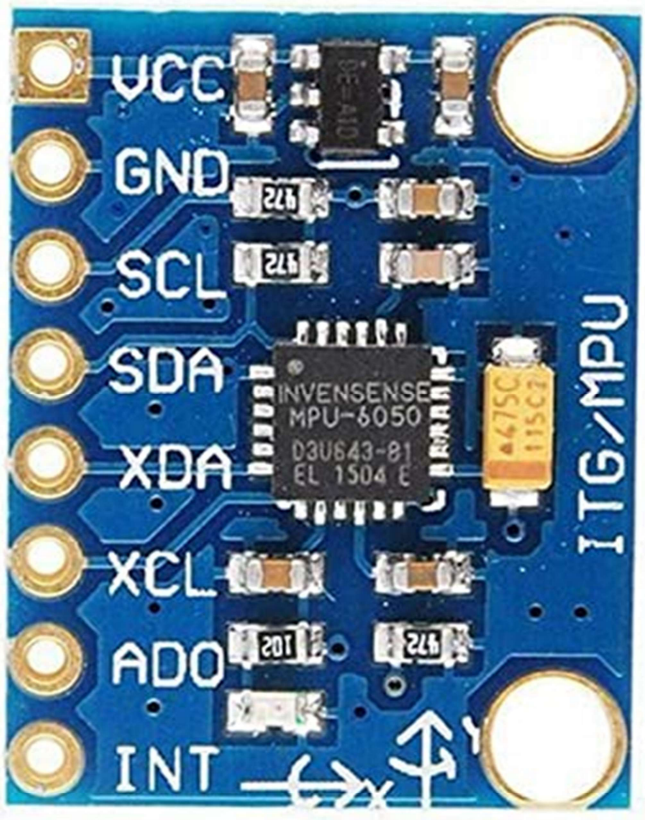
SDA

SCL – Communications lines for MPU6050 Accleretor – I2C

14,15 –

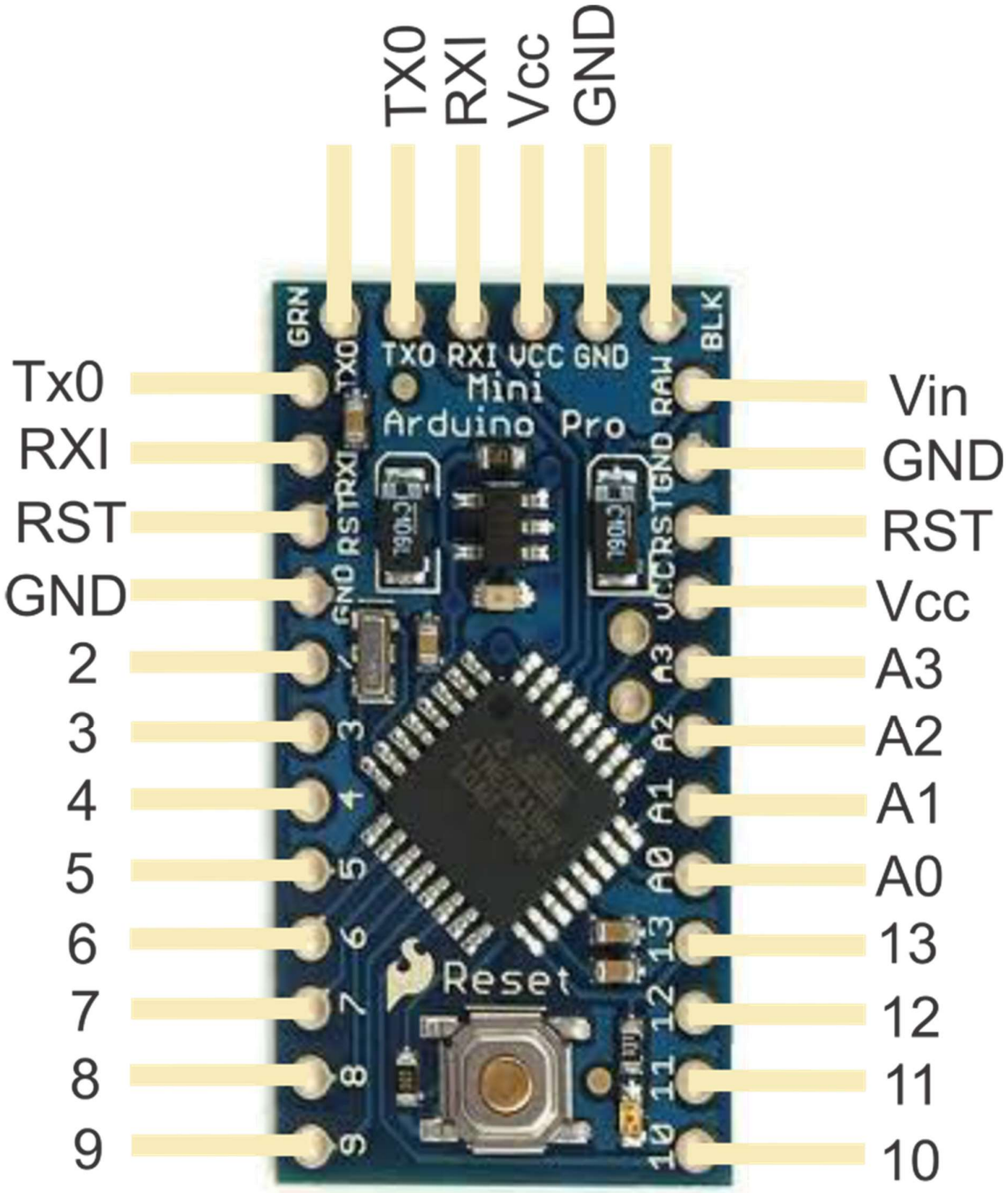
I2C Com. – Com bet Accerelometer n Micrrocontroller – (SCL,SDA,Int)

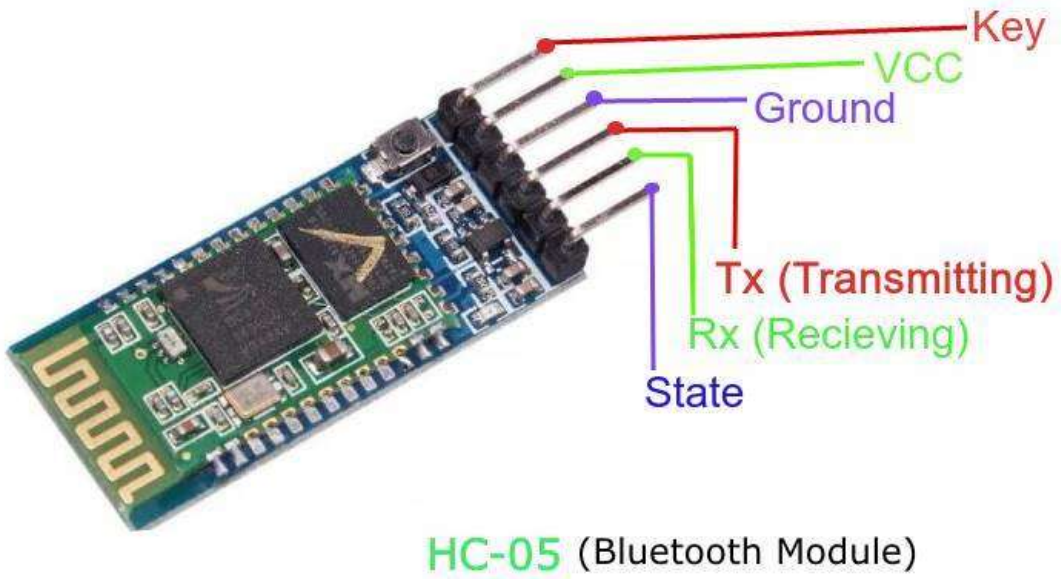
Serial Com – Bluetooth n Microcontroller(RXD, TXD)



ROBOMART.com™

NOTE:THE PRODUCT MAY BE DIFFER FROM IMAGE GIVEN ABOVE


TX0
RXI
Vcc
GND
Tx0
RXI
RST
GND
2
3
4
5
6
7
8
9
Vin
GND
RST
Vcc
A3
A2
A1
A0
13
12
11
10
GRN
BLK
TX0 RXI VCC GND
Mini
Arduino Pro
RAW
GND
RST
VCC
Reset



HC-05 (Bluetooth Module)

```
#include <Wire.h>
#include "I2Cdev.h"
#include "MPU6050_6Axis_MotionApps20.h"
#include "MPU6050.h"

MPU6050 accelgyro(0X69);//I2C addreess for mpu6050 is 0X69
```

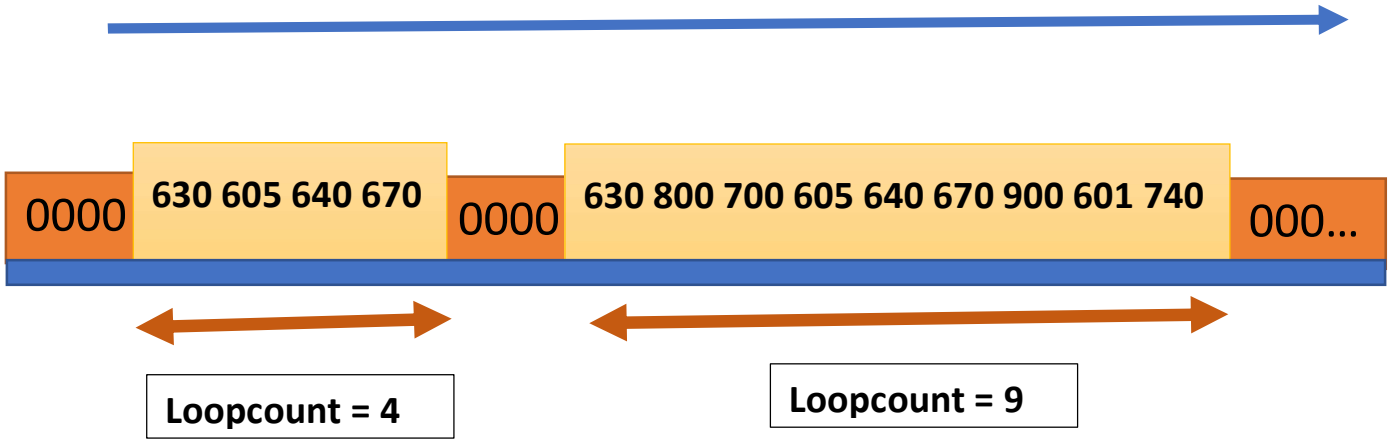
```
float X=0;
float Y=0;


float prevX=0;
float prevY=0;


float accelX=0;
float accelY=0;

float scalerX = 10;
float scalerY = 8;
```

```
int loopcount = 0;
int blowcount = 0;



void setup() {

  Wire.begin();// put your setup code here, to run once:
  Serial.begin(38400);
  accelgyro.initialize();
```

```
}

void loop() {
accelgyro.getAcceleration(&X, &Y,
&Z);



accelX = ((X - prevX)/2);
accelY = ((Y - prevX)/2);

prevX = X;
prevY = Y;

accelX = accelX * scalerX;
accelY = accelY * scalerY;
```

}

Click Processing Logic:



Outputing:

Mouse Coordinates Format: “X , Y”

Mouse Click Signal Format:

“1” – LEFT Click – Blow

“2” – RIGHT Click – Long Blow

Click Output Variable: blowcount(int)

Coordinates Variable:

mousecode(string): accelX + ”,” + accelY

Construction Flow:[METHODOLOGY]

1) Ideating
2) Model Designing and Planning
3) Components Search and Market Selection
4) Assign Microcontroller Pins on board
5) Prepare The Circuit Diagram
6) Start Making Connections, And Soldering
7) Powerup And Test Connections And Communications
8) Load The Our C Program
9) Assemble DESIGN On HEADPHONE
10) Glueing and Basing The Circuit Modules and Boards
11) Powering Up And **Prototype**

12) Test And Correction – **Beta Testing**

13) Revise and **Final Build**

PowerBank – 50gms – Biggest

Connection Details:

A5------> SDA[mpu6050]

A4----→ SCL[mpu6050]

Pin2----→INT[mpu6050]

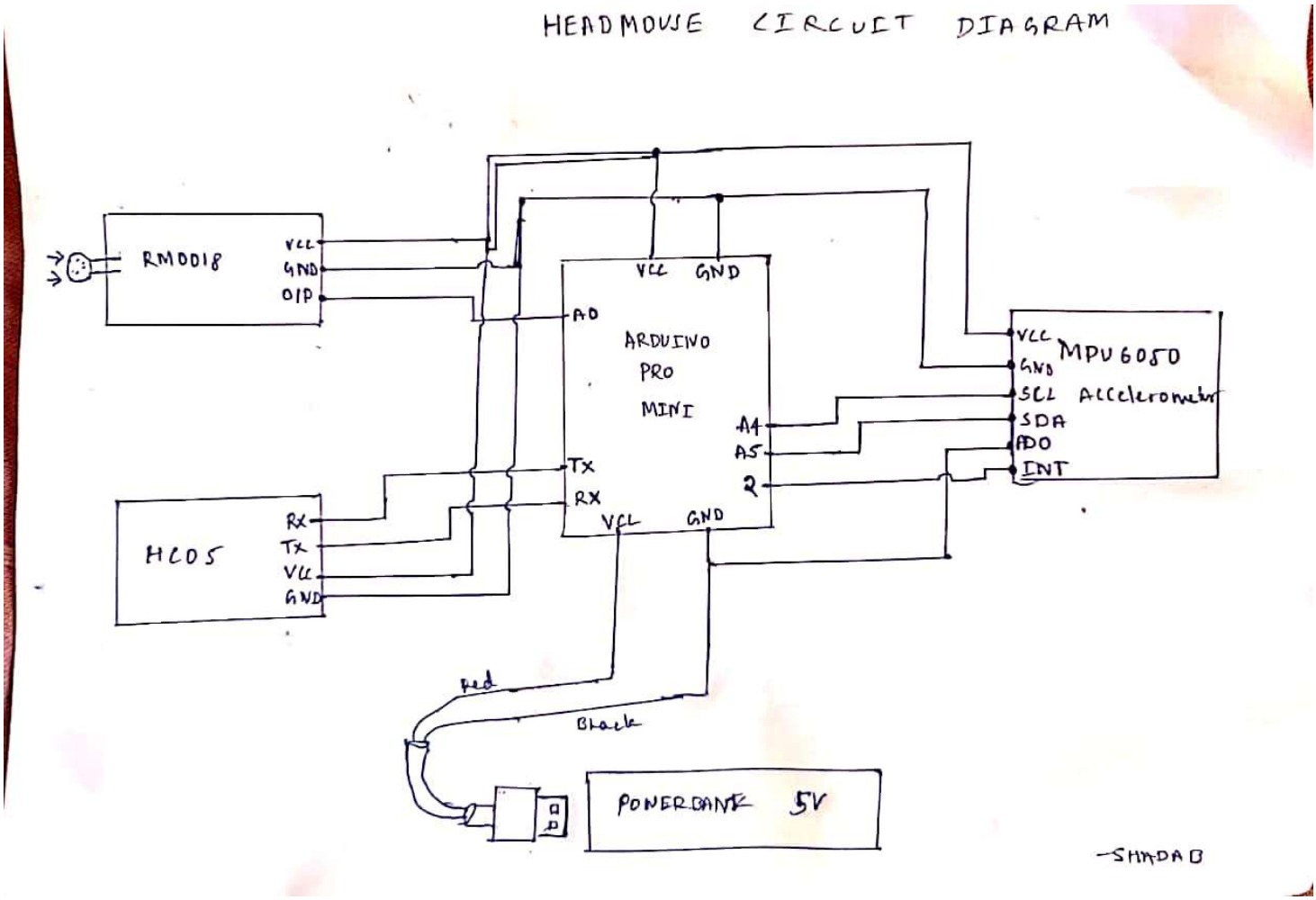
GND----→AD0[mpu6050]

Rx ----→Tx[Bluetooth]

Tx-----→Rx[Bluetooth]

A0-----→Mic Out[Rm00018]

A5,A4 – I2C Communication Pins-
Address- {0X69}



SOLDERING WIRES:

Soldering Station

Soldering Flux

Soldering Lead

Single Core Multi Strand Insulated Wire

Strip the Insulation

Wind the strands

Apply solder

Insert in the board

Final solder

<u>Connection Testing:</u>

-Short Circuits

-Components OverHeat

-Power Leakage Or Oversupply

1) ***Multimeter Terminal Test***
2) ***Test Programs to test Communication***