Gesture Controlled PC Mouse

Aim: To make a wireless computer mouse that is controlled by hand gestures.

Components Used:

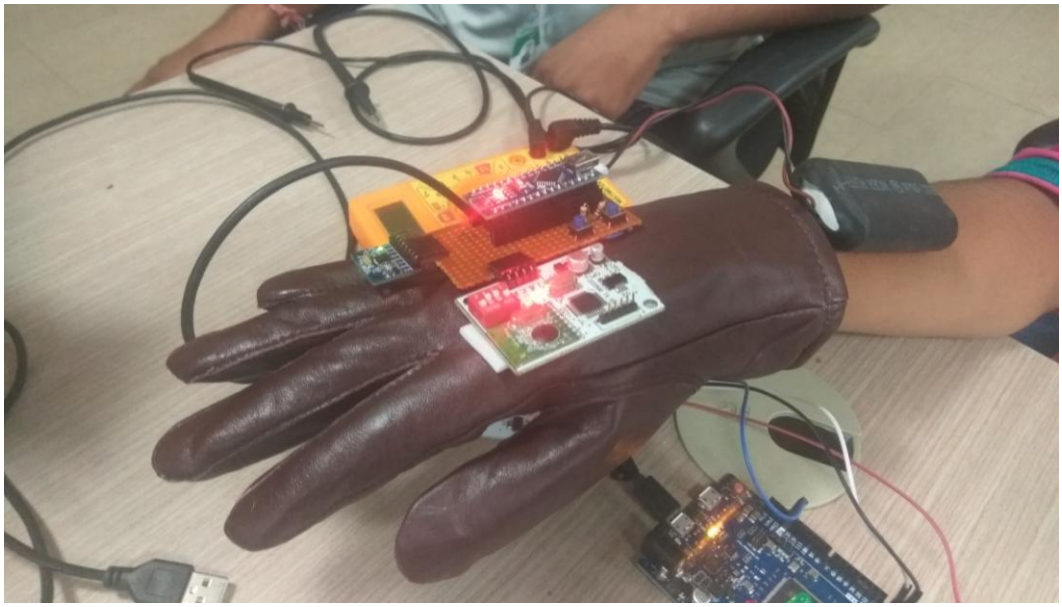
Sno	Component Name	Quantity	Price	Purchased From
1	Arduino Due	1	1500	Amazon
2	Arduino Nano	1	400	Amazon
3	CC2500 RF Transceiver	2	300	Lema Labs
4	GY-521 Module	1	275	Amazon
5	Push Button	1	5	Ritchie Street
6	Jumper wires	-	-	Ritchie Street

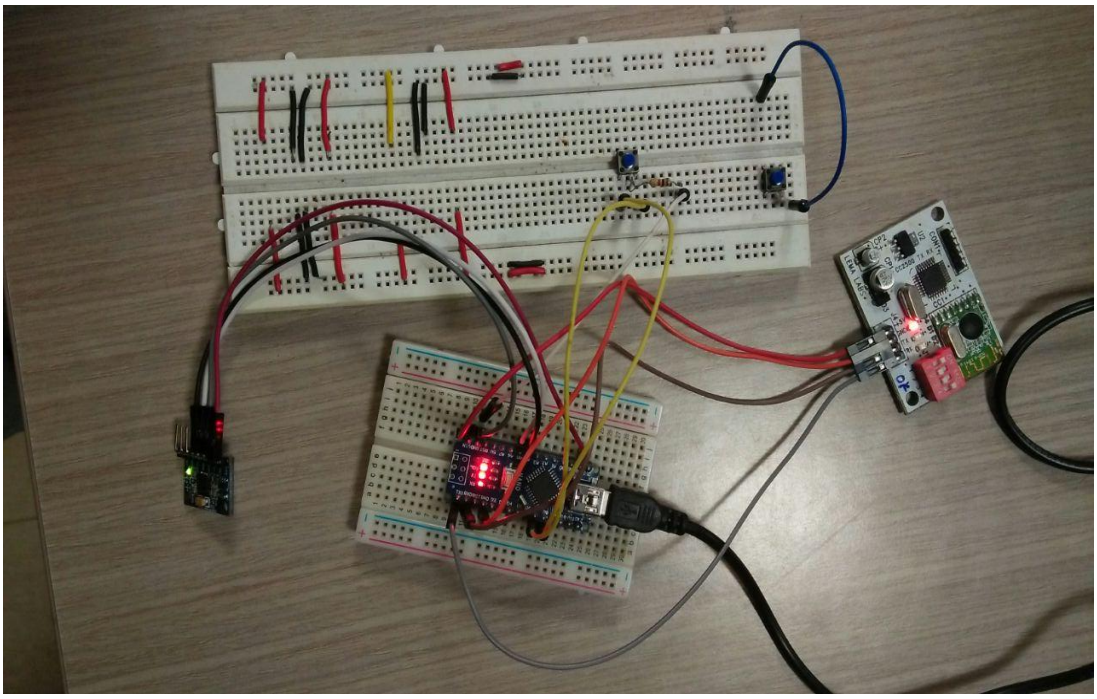
The System:

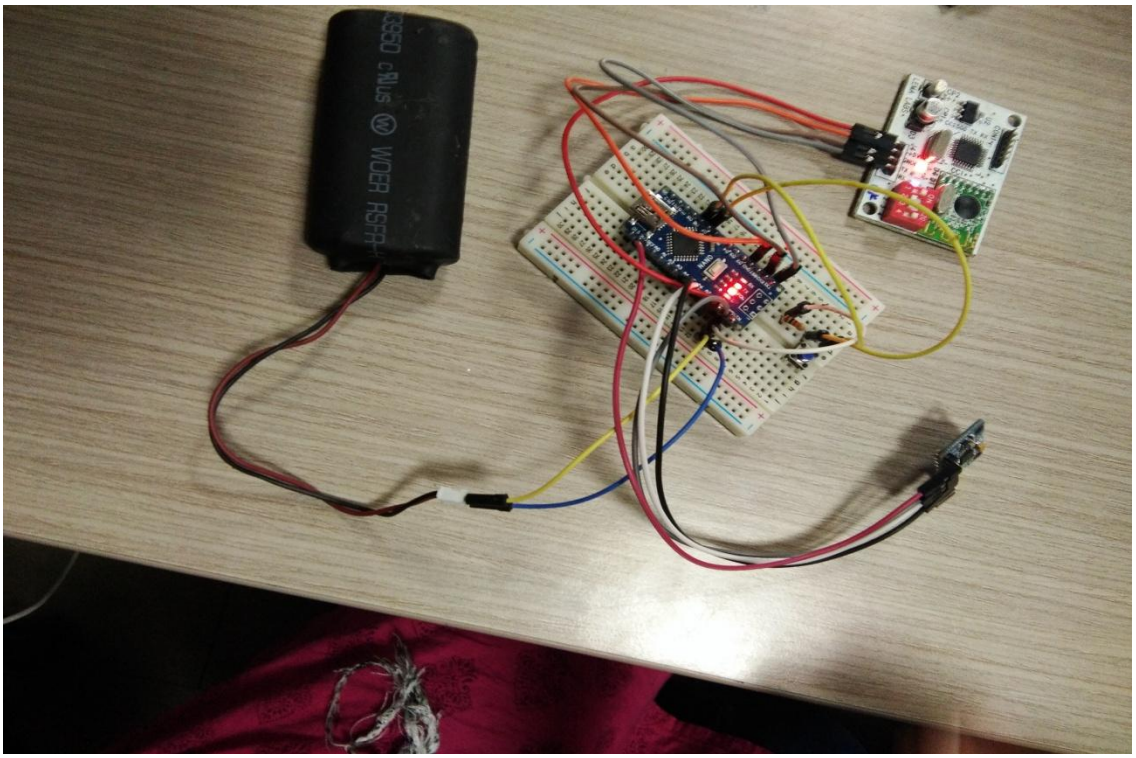
Top : Hand Module (next page)

Bottom : PC Module (next page)

Below : Final Hand Module with soldered breadboard







Circuit Connections:

1) Hand Module:-

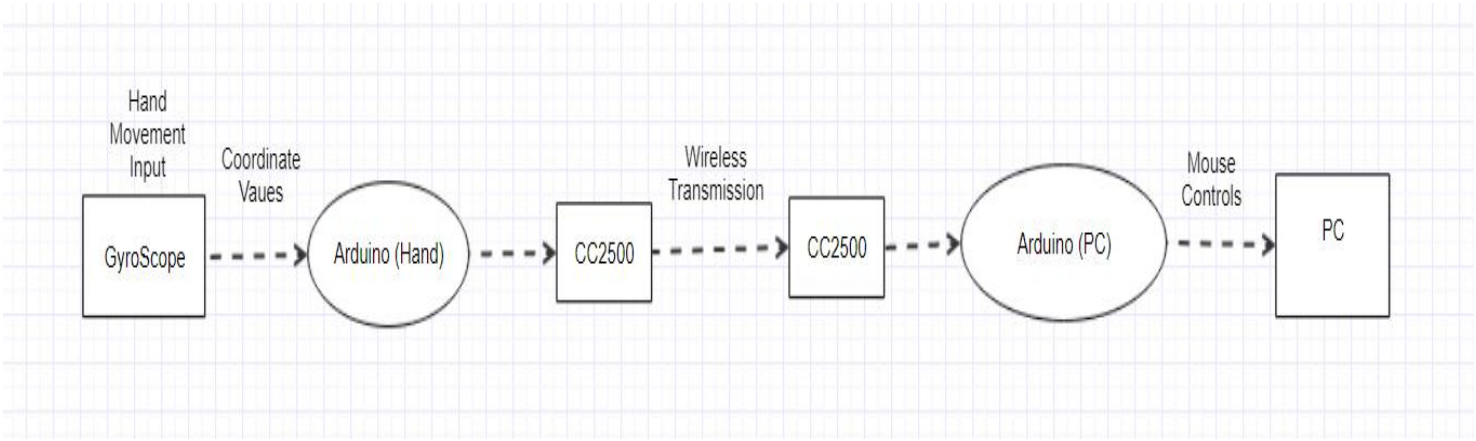
- Connect the Vcc and Gnd pins of the CC2500 to the battery's +ve and –ve poles of the power source.
- Connect Rx of the CC2500 to the Tx of the Arduino Nano and the Tx to Rx.
- Connect the Vcc ad Gnd of the GY-521 module to the +ve and –ve poles of the power source.
- Connect the SDA and SCL pins of the Gyro module to the SDA and SCL pins of the Arduino (Pin D4,D5 for the Nano)
- Wire a push button, with one end at Vcc and the other going into any digital input pin of the Arduino, this will be used as a trigger for mode change.
- Wire another push button to trigger a mouse click.
- The other pins aren't required for this project.

2) PC Module:-

- Use an Arduino which has Keyboard/Mouse emulator like Due or Leonardo.
- Connect the Vcc and Gnd pins of the CC2500 to the battery's +ve and –ve poles of the power source.
- Connect the Native port of the Arduino to the PC for Mouse/Keyboard emulation.
- For uploading codes use the Programming port.
- Connect two LED's to two different digital pins of the Arduino and ground them.

Construction :

- This system consists of 2 Arduino microcontroller boards paired with each other wirelessly using CC2500 RF module.
- The microcontroller on the hand has an Accelerometer/Gyroscopic module that helps in tracking movement and position of the hand.
- A button is provided for clicking.
- An additional button is provided to toggle the different modes – Mode 1, Mode 2, Mode 3.
- The microcontroller attached to the PC is used to receive inputs and send out mouse/keyboard commands to the computer.



~General Block Diagram~

Working :

- The system works with data sourced from the accelerometer.
- This data is used to find the orientation of the users hand to detect the gesture.
- Ex: A hand twisted 90 degrees left will read (-)15000 on its corresponding axis.
- These values are linearly mapped to the mouse movement.
- The more the bend the more the mouse moves left.
- Similarly the up, down and right movements are mapped to their respective Mouse cursor movements
- Keyboard inputs can also be given in this fashion.

Roadblocks :

- Accelerometer and Gyroscopic values readings contain a lot of noise, thus producing junk values.
- Wireless communication via RF is short range and at times very inaccurate

Future Scope :

- The Gesture Controlled mouse can be further be made more accurate by using precision sensors.
- The packaging can be made more compact using a dedicated PCB for this.
- The product can replace the conventional mouse given it can exhibit more types of gestures which are easy to understand and perform.

Code :

1) PC Module

```
#include<Mouse.h>
#include<Keyboard.h>
int st=1;                                   //Status of last arrow key used
int mode=3;                                 //Mode 1 - Mouse , 2 - Keyboard , 3 - Disable
int pr=1;
# define mo Serial3                         //Define Serial 3 as mo

void setup()
{
    mo.begin(9600);
    pinMode(A0,OUTPUT);                     //Set as output pins for led indication
    pinMode(A1,OUTPUT);
    }

    void kb(char val)                       //Keyboard function for Mode 2 - Keyboard
    {
            Keyboard.begin();               //Start Keyboard data transmission
            int y = (int(val))%10;
             if(y==0&&st==2)
            {
                    Keyboard.press(216);    //Left key press
                    st=1;
                    Keyboard.release(216);
             }
```

```
 else if(y==9&&st==2)
  {
              Keyboard.press(215);                          //Right key press
              st=1;
              Keyboard.release(215);
}
else if(y==4||y==5)
          st=2;
Keyboard.end();                                             //End keyboard transmission
 }
void mouse(char val)                                        //Mouse function for Mode 1 - Mouse
{
             Mouse.begin();                                 //Mouse begin
          int x = (int(val))/10;
          int y = (int(val))%10;
          x = map(x,0,9,10,-10);
          y = map(y,0,9,-10,10);
          Mouse.move(y,x,0);
            Mouse.end();                                    //End mouse
}

void loop()
{
          while(mo.available())                             //Check if data is available
           {
                    char a = mo.read();
 if(int(a)==120&&pr!=120)                                   //Check for a mode change
```

```
{
        if(mode==1)
            mode=2;                         //Cyclically assign modes
        else if(mode==2)
            mode=3;
        else
            mode=1;
}
else if(int(a)==121)                        //Check if button click is requeste
    Mouse.click();                          //If true click it
if(mode==1)
    kb(a);
else if(mode==2)
    mouse(a);
else if(mode==3)
    nothing();
if(mode==3)                                 //If mode 3 , turn off all LEDs
{
  digitalWrite(A0,0);
  digitalWrite(A1,0);
}
else if(mode==2)                            //For mode 2 turn on one LED
{
  digitalWrite(A0,1);
  digitalWrite(A1,0);
}
else if(mode==1)                            //For mode 1 turn on the other LED
```

```
    {
      digitalWrite(A0,0);
      digitalWrite(A1,1);
    }
    pr = int(a);

  }
}
```

‘

2) Hand Module

```
#include "I2Cdev.h"
#include "MPU6050.h"
#if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
#include "Wire.h"
#endif
MPU6050 accelgyro;
Int16_t ax, ay, az;
int16_t gx, gy, gz;
#define OUTPUT_READABLE_ACCELGYRO
#define LED_PIN 13
bool blinkState = false;
int mode = 3;

void setup() {
    #if I2CDEV_IMPLEMENTATION == I2CDEV_ARDUINO_WIRE
```

```
        Wire.begin();
  #elif I2CDEV_IMPLEMENTATION == I2CDEV_BUILTIN_FASTWIRE
  Fastwire::setup(400, true);
  #endif
  Serial.begin(9600);
  pinMode(8, OUTPUT);
  digitalWrite(8, HIGH);
  pinMode(A0, OUTPUT);
  digitalWrite(A0, HIGH);
  pinMode(7, INPUT);                                    //Switch for Mode selection
  pinMode(14, OUTPUT);
  pinMode(A1,INPUT);                                    //Switch for click
  Serial.println("Initializing I2C devices...");
  accelgyro.initialize();
  Serial.println("Testing device connections...");
  Serial.println(accelgyro.testConnection() ? "MPU6050 connection successful" :
"MPU6050 connection failed");
}


void loop() {
  accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
  delay(5);
  if (digitalRead(7) == HIGH)                           //Check if switch mode button is pressed
  {
     Serial.print(char(120));
     delay(100);
   }
```

```
  else if (digitalRead(A1)==1)                    //Check if click button is pressed
  {
     Serial.print(char(121));
   }
   else {
     int x1 = map(ax, -15000, 15000, 0, 9);       //Mapping values of gyroscope to mouse X axis
     int y1 = map(ay, -15000, 15000, 0, 9);       //Mapping values of gyroscope to mouse Y axis
     int z1 = x1 * 10 + y1;                        //Converting data to a single character
     Serial.print(char(z1));                      //Sending the data to the other Arduino

  }

}
```

~~~~~~
~~~~~~