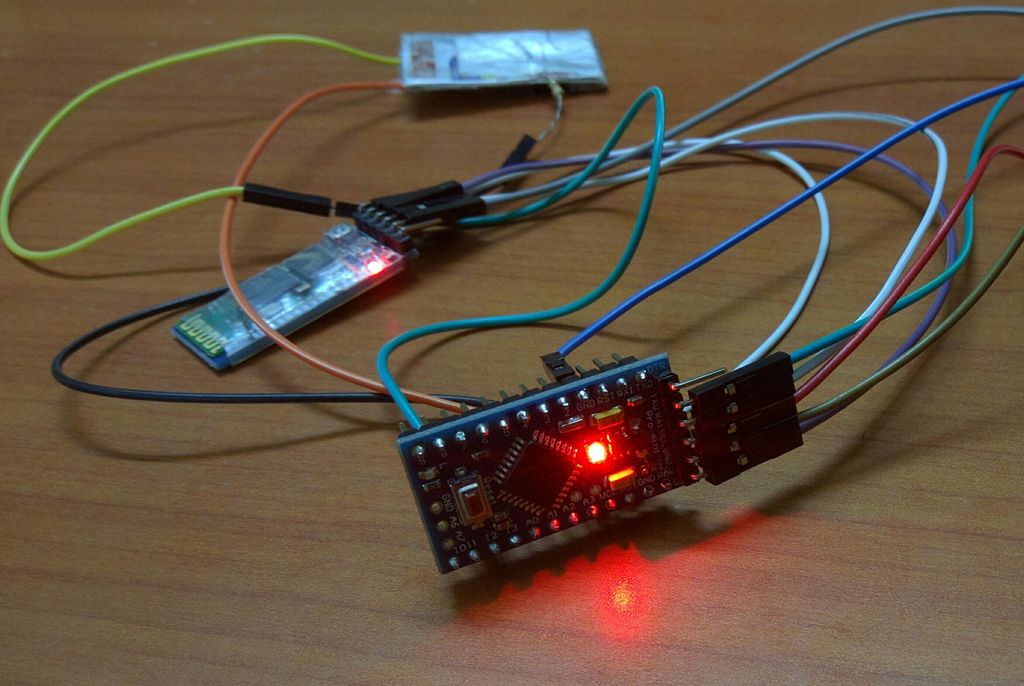
**PEDOMETER USING ARDIUNO**

BY-VARUN AGARWAL(207)

DEEP PATEL(210)

MALAY PATIL(211)

ABHISHEK KODAD(220)



COMPONENTS OF ARDUINO

POWER

Every Arduino board needs a way to be connected to a power source. The Arduino UNO can be powered from a USB cable coming from your computer or a wall power supply that is terminated in a barrel jack.

PINS

The pins on your Arduino are the places where you connect wires to construct a circuit ,probably in connection with a breadboard and some wire. They usually have black plastic ‘headers’ that allow you to just plug a wire right into the board. The Arduino has several different kinds of pins:

* **GND (3)**: Short for ‘Ground’. There are several GND pins on the Arduino, any of which can be used to ground your circuit.
* **5V (4) & 3.3V (5):** The 5V pin supplies 5 volts of power, and the 3.3V pin supplies 3.3 volts of power. Most of the simple components used with the Arduino run happily off of 5 or 3.3 volts.
* **Analog (6)**: The area of pins under the ‘Analog In’ label (A0 through A5 on the UNO) are Analog In pins. These pins can read the signal from an analog sensor and convert it into a digital value that we can read.
* **Digital (7)**: Across from the analog pins are the digital pins (0 through 13 on the UNO). These pins can be used for both digital input (like telling if a button is pushed) and digital output (like powering an LED).
* **PWM (8)**: You may have noticed the tilde (~) next to some of the digital pins (3, 5, 6, 9, 10, and 11 on the UNO). These pins act as normal digital pins, but can also be used for something called Pulse-Width Modulation (PWM), these pins are able to simulate analog output (like fading an LED in and out).
* **AREF (9)**: Stands for Analog Reference. Most of the time you can leave this pin alone. It is sometimes used to set an external reference voltage (between 0 and 5 Volts) as the upper limit for the analog input pins.

RESET BUTTON

### Arduino has a reset button **(10)**. Pushing it will temporarily connect the reset pin to ground and restart any code that is loaded on the Arduino. This can be very useful if your code doesn’t repeat, but you want to test it multiple times.

### POWER LED INDICATOR

Just beneath and to the right of the word “UNO” on your circuit board, there’s a tiny LED next to the word ‘ON’ **(11)**. This LED should light up whenever you plug your Arduino into a power source. If this light doesn’t turn on, there’s a good chance something is wrong.

TX RX LEDs

TX is short for transmit, RX is short for receive. These markings appear quite a bit in electronics to indicate the pins responsible for serial communication. There are two places on the Arduino UNO where TX and RX appear – once by digital pins 0 and 1, and a second time next to the TX and RX indicator LEDs **(12)**. These LEDs will give us some visual indications whenever our Arduino is receiving or transmitting data.

MAIN IC

Integrated Circuit **(13)** can be considered as the brain of our Arduino. The main IC on the Arduino is slightly different from board type to board type, but is usually from the ATmega line of IC’s from the ATMEL company. This can be important, as you may need to know the IC type (along with your board type) before loading up a new program from the Arduino software. This information can usually be found in writing on the top side of the IC. If you want to know more about the difference between various IC’s, reading the datasheets is often a good idea.

VOLTAGE REGULATOR

The voltage regulator **(14)** is not actually something you can (or should) interact with on the Arduino. But it is potentially useful to know that it is there and what it’s for. The voltage regulator does exactly what it says – it controls the amount of voltage that is let into the Arduino board. It allows around 20V.

SENSOR

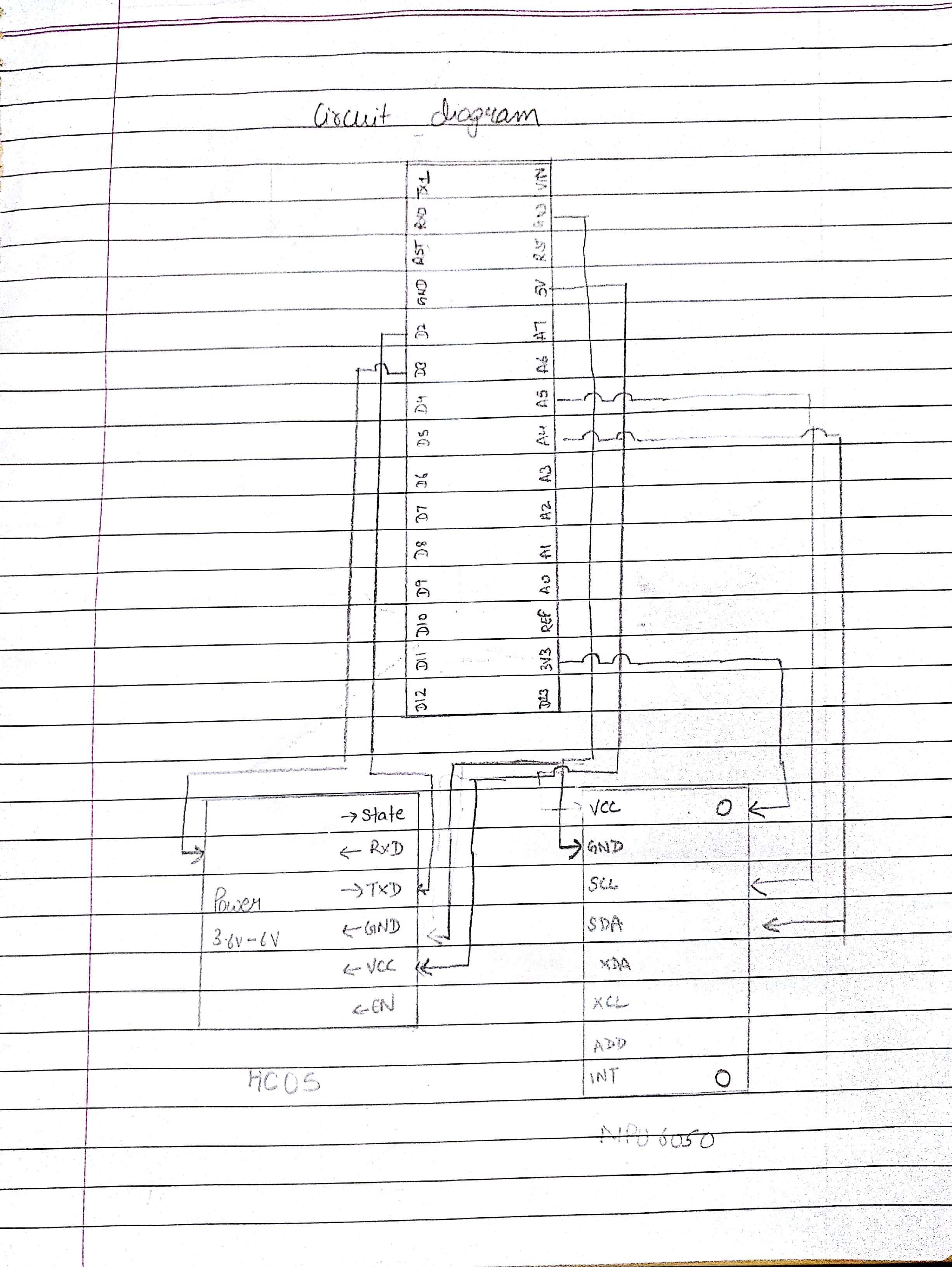
MPU-6050

IMU sensors like the MPU 6050 are used in self-balancing robots, UAVs, smartphones, and more. IMU sensors help us get the position of an object attached to the sensor in three-dimensional space. These values are usually in angles to help us to determine its position. They are used to detect the orientation of smartphones, or in wearable gadgets like the Fitbit, which uses IMU sensors to track movement..

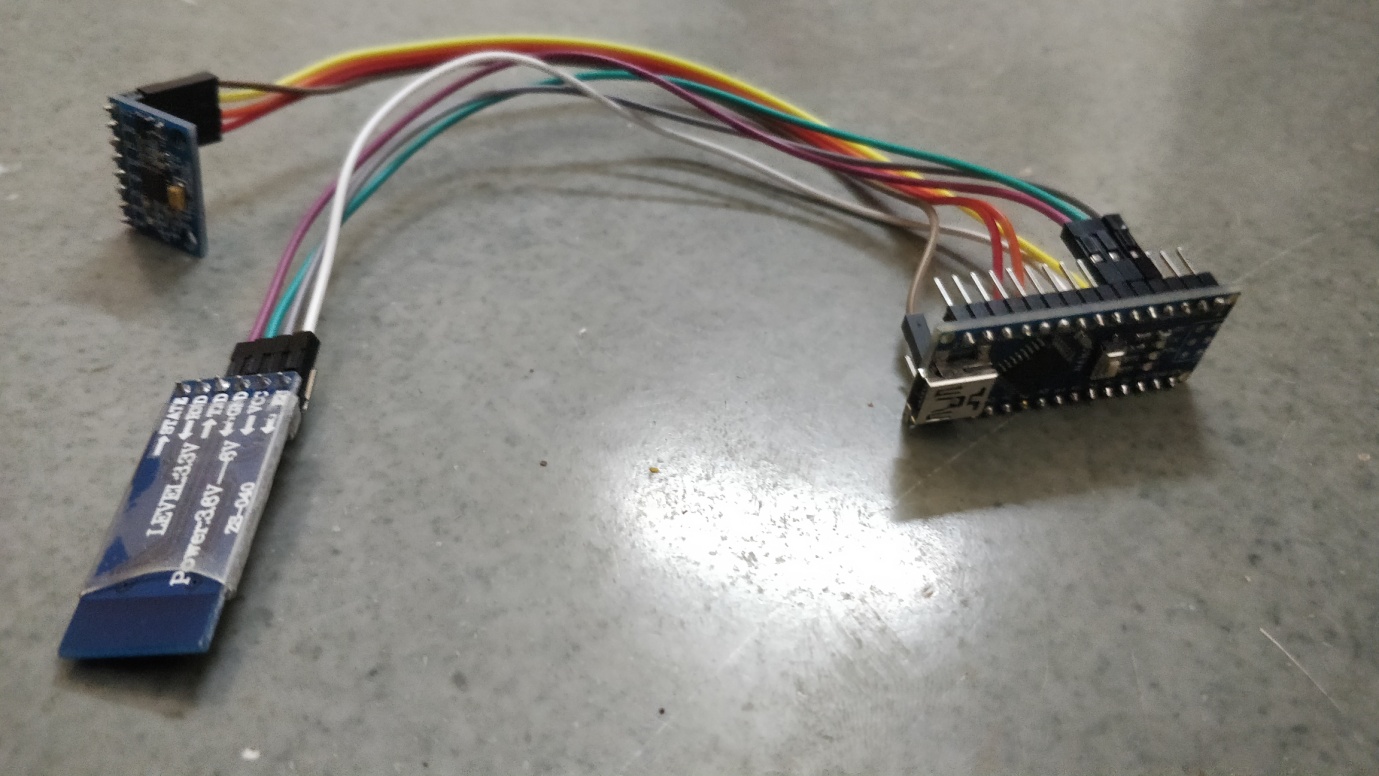
GYRO SENSOR

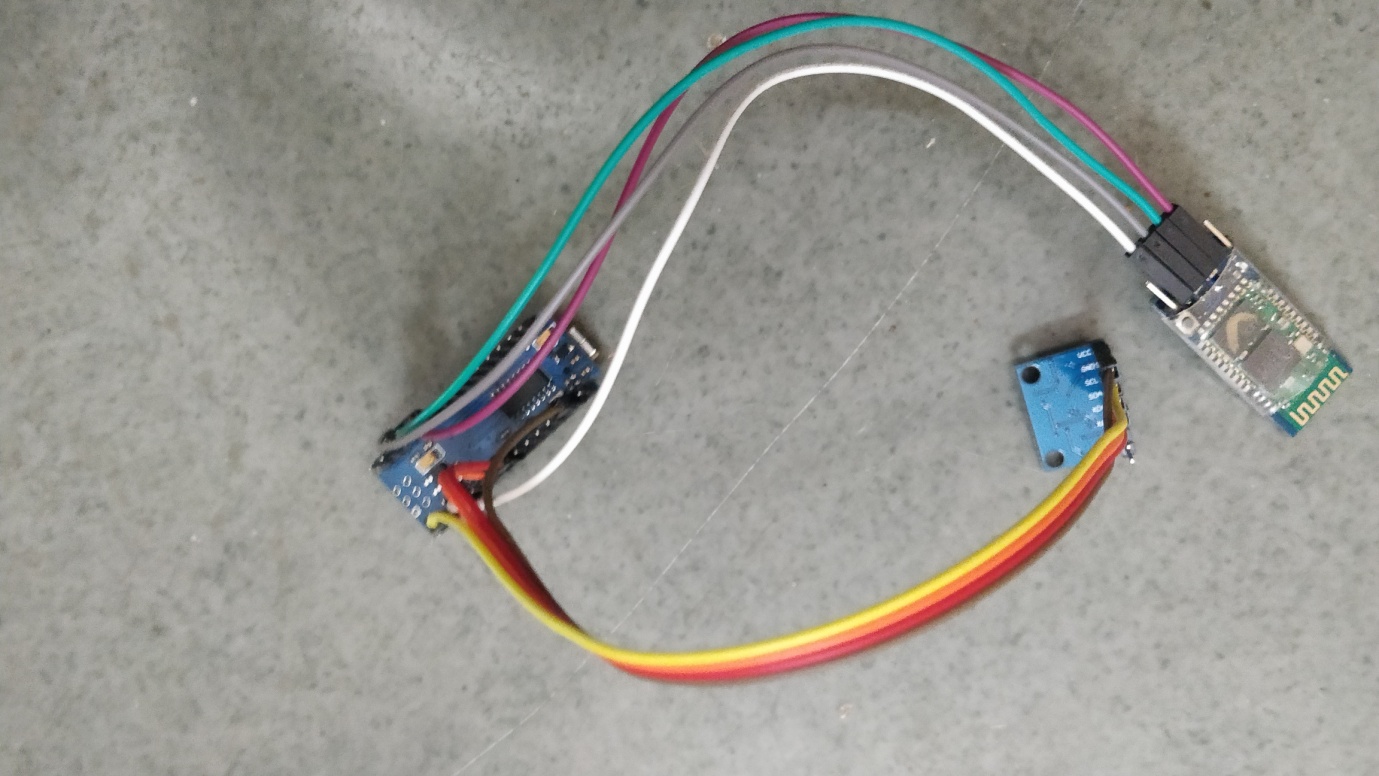
A gyroscope is a device that uses Earth's gravity to help determine orientation. Its design consists of a freely-rotating disk called a rotor, mounted onto a spinning axis in the center of a larger and more stable wheel

CIRCUIT DIAGRAM



SNAPS OF WORKING PROJECT





ARDUINO CODE :

#include <Wire.h>                                       //library to connect 2 two wire interface

#include <MPU6050.h>                                    //for interfacing the mpu6050 sensor

#include <SoftwareSerial.h>                             //comunicate with arduino serial monitor and BT module

SoftwareSerial BTSerial(2, 3);                          //connecting HC-05 for transmission BTserial(TX , RX)

MPU6050 mpu;

/\*connections

mpu6050 :

VCC -> 3.3V

GND -> gnd

SCL -> A5

SDA -> A4

HC-05 :

VCC -> 5V

GND -> GND

TXD -> D2

RXD -> D3

\*/

unsigned long timer = 0;

float timeStep = 0.01;

float x= 0;

float y = 0;

float z = 0;

float a=0;

int step1;

void setup() {

  Serial.begin(9600);

  BTSerial.begin(9600);

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

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

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

    delay(500);

  }

  mpu.calibrateGyro();

  mpu.setThreshold(3);

}

void loop()

{

  float temp = mpu.readTemperature();

  a=(x+y)/2;

  timer = millis();

  Vector norm = mpu.readNormalizeGyro();

  x = x + norm.YAxis \* timeStep;                                          //raw data of roll

  y = y + norm.XAxis \* timeStep;                                          //raw data of pitch

  z = z + norm.ZAxis \* timeStep;                                          //raw data of yaw

  delay((timeStep\*1000) - (millis() - timer));

  if((x+y)/2!=a){

    step1++;

    float dist=step1\*1.2;

    Serial.print("  steps : ");

    Serial.print(step1);

    Serial.print("  temp : ");

    Serial.print(temp);

    Serial.print("  dist : ");

    Serial.println(dist);

    BTSerial.print("\n");

    BTSerial.print(" steps : ");

    BTSerial.print(step1);

    BTSerial.print(" temp : ");

    BTSerial.print(temp);

    BTSerial.print(" dist : ");

    BTSerial.print(dist);

    BTSerial.print(" ft");

    delay(800);

  }

}

APPLICATIONS OF THE PROJECT

SMART WATCHES

SMART PHONES

GOOGLE FIT

Pacer Pedometer & Weight Loss Coach

CONCLUSION

We have successfully made pedometer with the sensors named MPU6050 (GYRO SENSOR) which can count the steps and also heart rate with the help of smart watches and smart phones.

FUTURE SCOPE

It can be used for monitoring activities on smartwatches also at affordable prices .

It can be used to control their health problems.