SAVITRIBAI PHULE PUNE UNIVERSITY

A MINI-PROJECT REPORT ON

FALL DETECTION SYSTEM FOR ELDERLY PEOPLE

SUBMITTED TOWARDS THE
PARTIAL FULFILLMENT OF THE REQUIREMENTS OF

FOURTH YEAR SEMESTER II OF ENGINEERING

(Computer Engineering)

BY

APOORVA KHUSPE	**4143**
MRUGAKSHI CHIDRAWAR	**4153**
PRATIK PATIL	**4159**

Under The Guidance of

Prof. P.S. Vidap

DEPARTMENT OF COMPUTER ENGINEERING
Pune Institute of Computer Technology Dhankawadi,
Pune-411043

Pune Institute of Computer Technology

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the Mini-Project Entitled

FALL DETECTION SYSTEM FOR ELDERLY PEOPLE

Submitted by

Apoorva Khuspe	4143
Mrugakshi Chidrawar	4153
Pratik Patil	4159

is a bonafide work carried out by Students under the guidance of **Prof. P.S. Vidap** and it is submitted towards the partial fulfillment of the requirement of **Fourth Year Computer Engineering Semester II** of **Savitribai Phule Pune University**.

Prof. P.S. Vidap	Prof. M. S. Takalikar
Internal Guide	**H.O.D**
Dept. of Computer Engg.	**Dept. of Computer Engg.**

Date :

Place : Pune

Contents

FALL DETECTION SYSTEM FOR ELDERLY PEOPLE

Abstract:

Fall detection is a major challenge in the public healthcare domain. Falls for the elderly can be especially serious, as they can lead to serious health issues. Making a system to alert when a fall occurs is helpful, as it provides peace of mind to both the wearer and their loved ones.

Nowadays, fall detectors are commonly included in smartwatches, wearable fitness trackers, and other types of wearables. Fall detection can also be done by a surveillance camera, also known as a computer-vision-based fall detection system. There are several kinds of fall detection methods and the algorithm each uses is what makes them unique and accurate.

Keywords: Fall Detection, Arduino Uno, MPU6050, Safety of elders

1 INTRODUCTION

Adults 65 years of age or older experience higher rates of falling and are generally at a higher risk for falls. Falls and fall related injuries represent a significant threat to the health and independence of adults 65 years of age and older. Falls can have severe consequences such as injury or death; in 2010 in the United States, 21,649 older adults died from fall related injuries. Even if a fall does not result in a physical injury, it can often produce fear of falling resulting in a decrease in mobility, participation in activities, and independence. Fall detection technologies enable rapid detection and intervention for individuals who have experienced a fall. This ability could reduce the physical and mental damage caused not only by the fall but time after a fall before discovery.

1.1 Motivation

Falls represent a significant threat to the health and independence of adults 65 years of age and older. As a wide variety and large amount of passive monitoring systems are currently and increasingly available to detect when an individual has fallen, there is a need to analyse and synthesize the evidence regarding their ability to accurately detect falls to determine which systems are most effective.

Nowadays, fall detectors are commonly included in smartwatches, wearable fitness trackers, and other types of wearables. Fall detection can also be done by a surveillance camera, also known as a computer-vision-based fall detection system. There are several kinds of fall detection methods and the algorithm each uses is what makes them unique and accurate. In most fall situations, the body leans to the side and touches the ground with high acceleration. So, an algorithm must detect a fall in these situations when there is a rapid change of position in a very short amount of time. Commonly, fall detection systems use a gyroscope and an accelerometer. A gyroscope is used to determine an orientation and an accelerometer provides the information about the angular parameter as three-axis data. But we also need to decide a threshold so that the system can differentiate between a fall and normal activity.

1.2 Purpose

The purpose of developing fall detection system is to computerize the alertness mechanism in times of accidents related to old people. Another major purpose is that the near and dear ones will be immediately alerted about any possible mishappening and they can take appropriate medical treatment on time.

2 PROBLEM STATEMENT AND SCOPE

2.1 Problem Statement

To develop an effective system for monitoring and detecting fall of elder people so that the surrounding people can provide quick help to them.

2.2 Scope

The scope of the project is : The gyroscope and accelerometer in the fall detector system will detect any uncommon movement of the person, and if detected, would trig ger signals of 3 types namely fair, moderate and serious according to the sudden change in position and speed of the person.

3 DESIGN AND ANALYSIS OF SYSTEM

3.1 Design of ERTOS System

3.1.1 Module Information

This system has the following modules :

1) MPU6050 accelerometer and gyroscope breakout module

2) Arduino UNO

3) Jumper Wires (Generic)

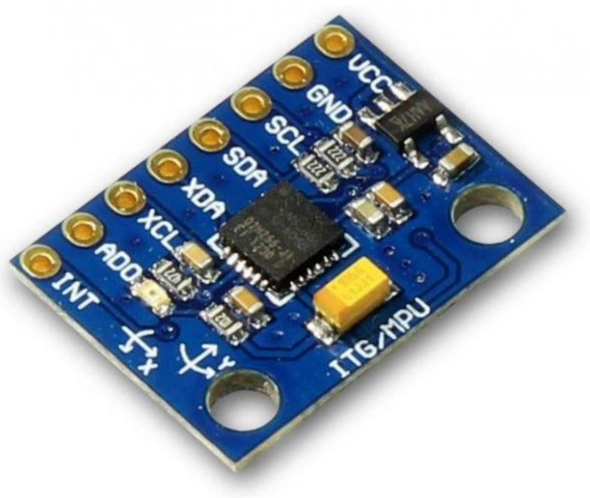
4) Arduino IDE

5) Type B Cable connector

6) PC

3.1.2 Module Description

1) **MPU6050 accelerometer and gyroscope breakout module :** The MPU6050 IMU has both 3-Axis accelerometer and 3-Axis gyroscope integrated on a single chip. The gyroscope measures rotational velocity or rate of change of the angular position over time, along the X, Y and Z axis. On the other hand, the MPU6050 accelerometer measures acceleration in the same way as explained in the previous video for the ADXL345 accelerometer sensor. Briefly, it can measure gravitational acceleration along the 3 axes and using some trigonometry math we can calculate the angle at which the sensor is positioned. So, if we fuse, or combine the accelerometer and gyroscope data we can get very accurate information about the sensor orientation. The MPU6050 IMU is also called six-axis motion tracking device or 6 DoF (six Degrees of Freedom) device, because of its 6 outputs, or the 3 accelerometer outputs and the 3 gyroscope outputs.

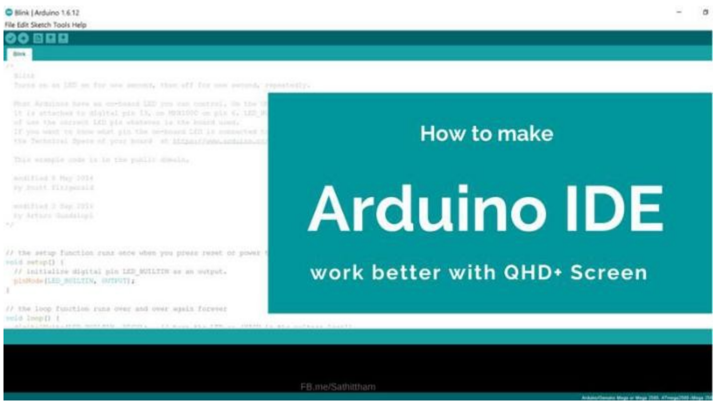


2) **Arduino UNO** : The Arduino UNO is an open-source microcontroller board based on the Microchip ATmega328P microcontroller and developed by Arduino.cc. The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable. It can be powered by a USB cable or by an external 9 volt battery, though it accepts voltages between 7 and 20 volts.

3) **Jumper Wires (Generic)** : Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

4) **Arduino IDE** : The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.



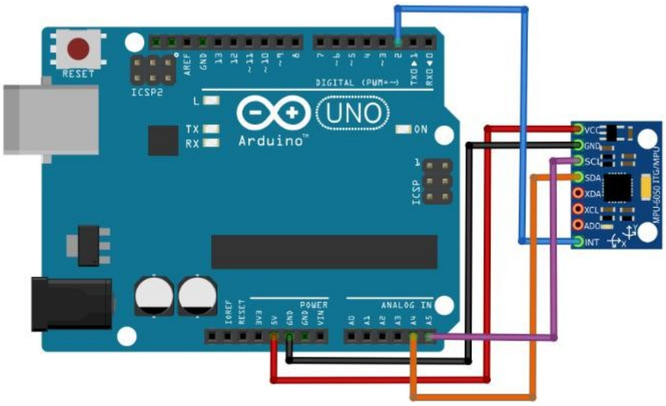
5) **Type B Cable Connector** : A Type B cable connector is used to connect Arduino UNO to the PCs USB port.

6) **PC** : A PC or Laptop is need to perform coding and dumping the code into Arduino UNO.

3.2 ERTOS Design Methodology Steps

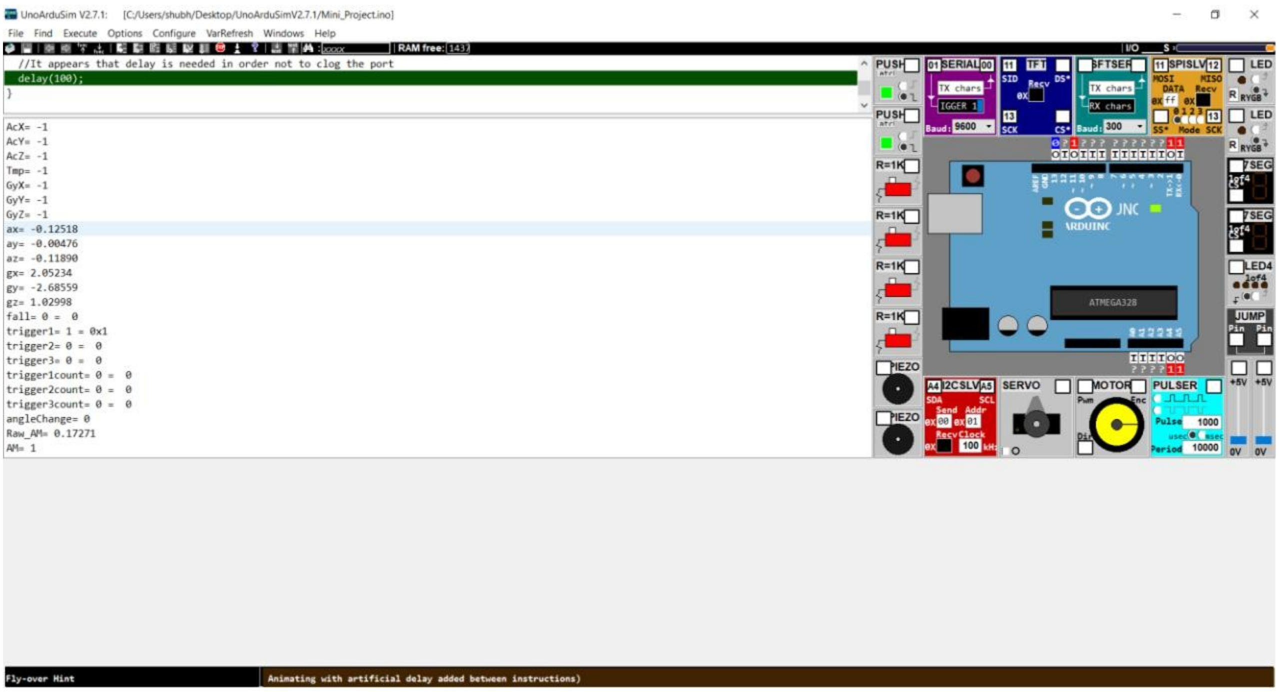
Device and Component Integration

1. The device and ccomponents are included in the diagram.



Application Development

1. The application is developed according to the circuit diagram and the core is dumped into the Arduino.
2. After dumping the code, the program is executed.
3. The Output of the real-time conditions are presented on the display scereen.



4 CIRCUIT DIAGRAM

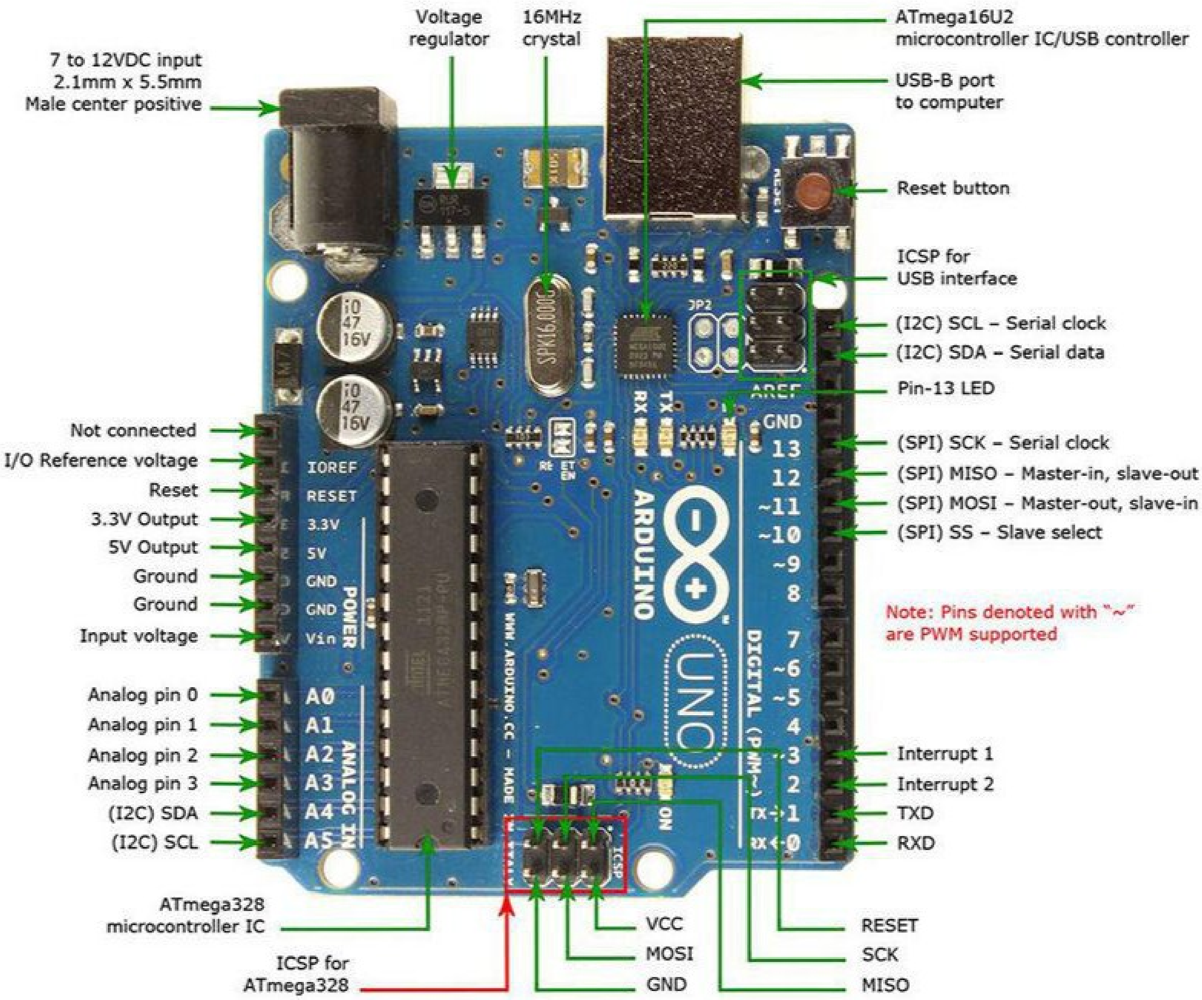


Figure 1: Arduino UNO

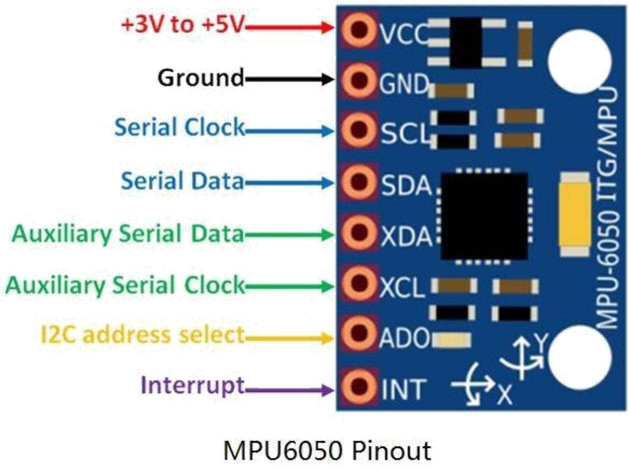


Figure 2: MPU6050 accelerometer and gyroscope breakout module

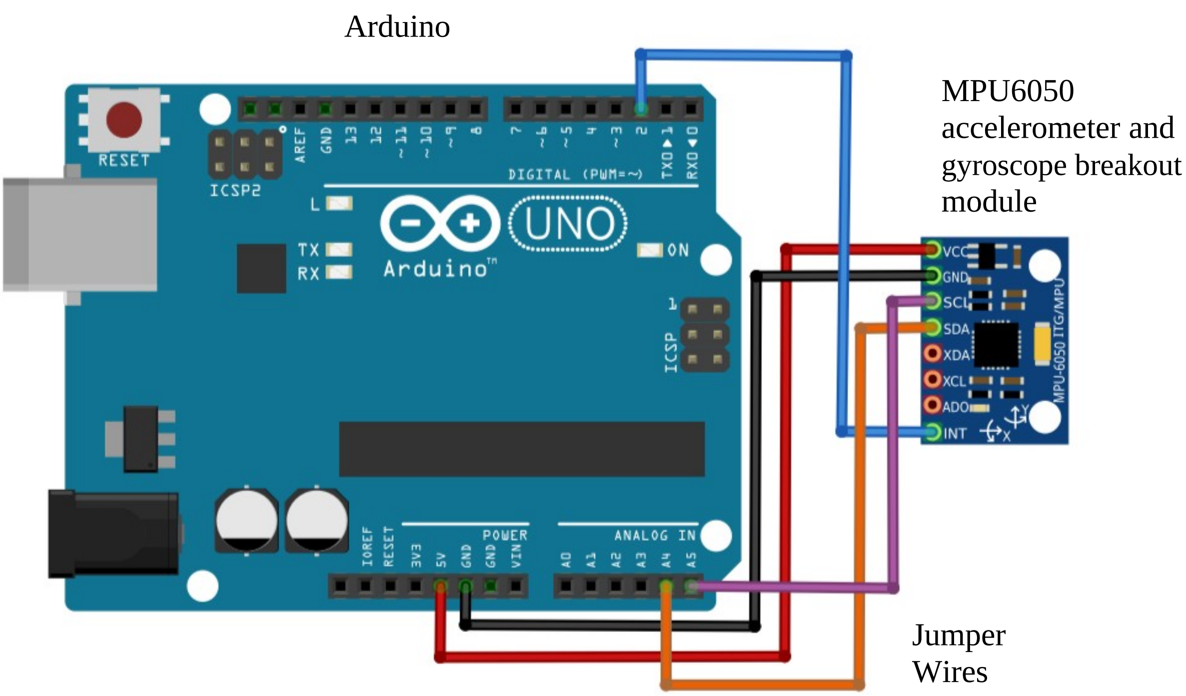


Figure 3: Circuit Diagram for proposed system

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

Ultimately the result of the implementation of this project will lead to reduction in the number of small and medium scale accidents experienced by elders. A desktop program for monitoring and detecting fall of elders or even infants is developed. This would definitely help the elders to get timely help due to the presence of precision systems . This will ensure safety and will help the loved and near ones of the elders to be alert.

5.2 Future Work

The system can be enhanced using different technology such as an Android based application for mobile monitoring system with the help of WiFi controlled modules to access information from any part. Also, features such as audio and video recording can be added to obtain detailed information about the present condition of the elders.

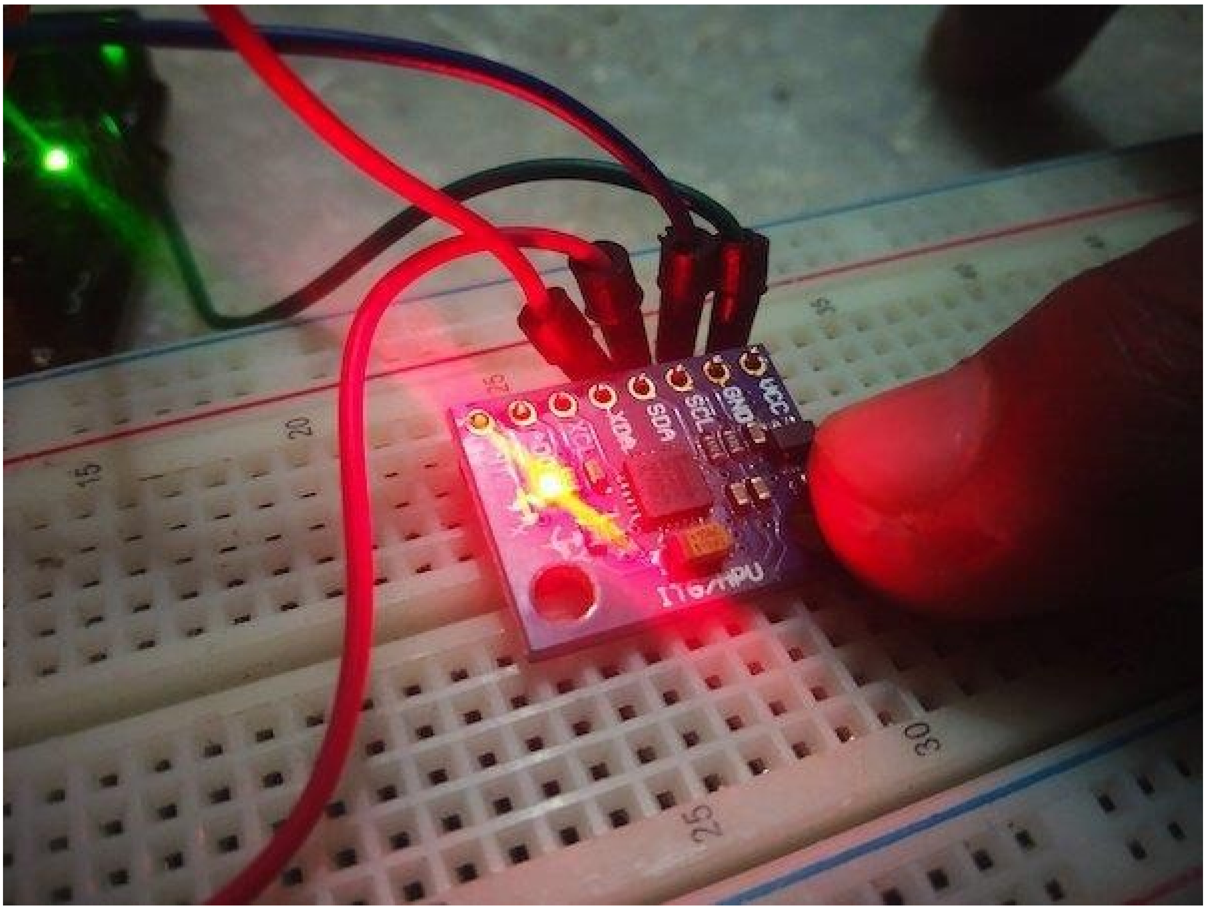
References

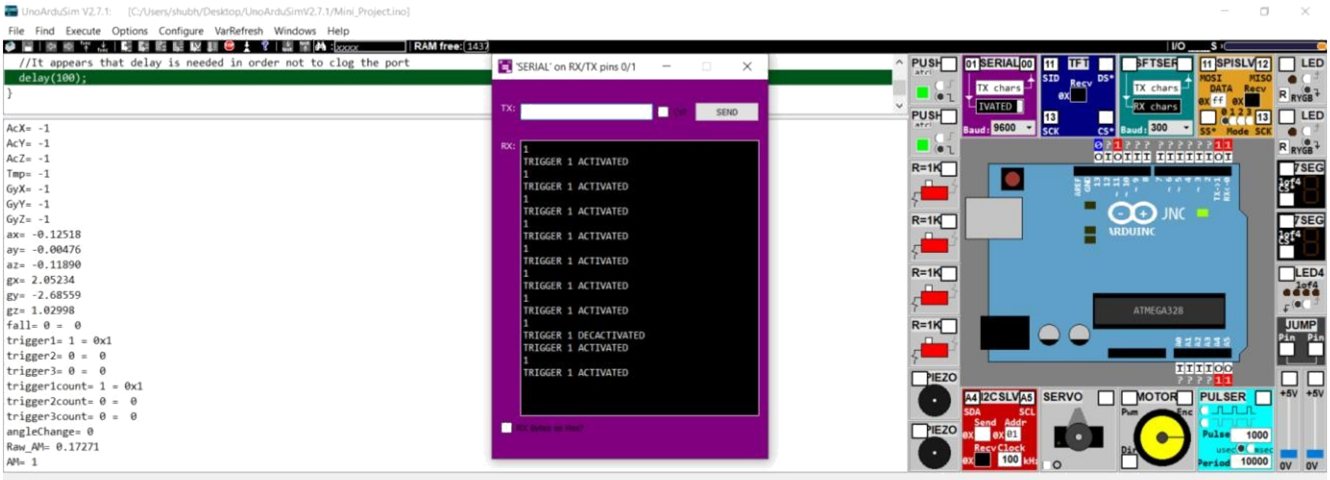
[1] "How to build a Fall Detector with Arduino", https://maker.pro/arduino/tutorial/how-to-build-a-fall-detector-with-arduino

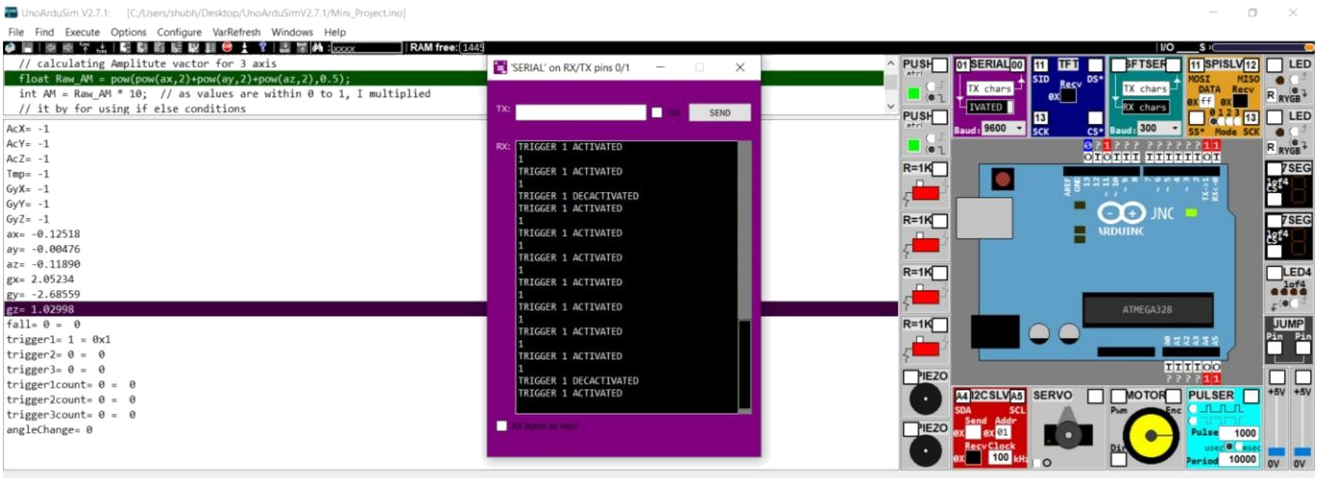
[2] "Arduino and MPU6050 Accelerometer and Gyroscope Tutorial", https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/

[3] "How to Interface Arduino and the MPU 6050 Sensor", https://maker.pro/arduino/tutorial/how-to-interface-arduino-and-the-mpu-6050-sensor

IMPLEMENTATION & TESTING







SOURCE CODE

Upload the below Sketch to Arduino UNO

```
#include<Wire.h>
const int MPU_addr=0x68; // I2C address of the MPU-
6050 int16_t AcX=0,AcY=0,AcZ=0,Tmp,GyX=0,GyY=0,GyZ=0;
float ax=0, ay=0, az=0, gx=0, gy=0, gz=0;

//int data[STORE_SIZE][5]; //array for saving past data
//byte currentIndex=0; //stores current data array index (0-255)
boolean fall = false; //stores if a fall has occurred
boolean trigger1=false; //stores if first trigger (lower threshold) has occurred
boolean trigger2=false; //stores if second trigger (upper threshold) has occurred
boolean trigger3=false; //stores if third trigger (orientation change) has occurred

byte trigger1count=0; //stores the counts past since trigger 1 was set
true byte trigger2count=0; //stores the counts past since trigger 2 was
set true byte trigger3count=0; //stores the counts past since trigger 3
was set true int angleChange=0;


void setup(){
Wire.begin();
Wire.beginTransmission(MPU_addr);
Wire.write(0x6B); // PWR_MGMT_1 register
//Wire.write(0); // set to zero (wakes up the MPU-6050)
Wire.endTransmission(true);
Serial.begin(9600);
pinMode(11, OUTPUT);
digitalWrite(11, HIGH);
}
void loop(){
mpu_read();
//2050, 77, 1947 are values for calibration of accelerometer
/  values may be different for you ax = (AcX-2050)/16384.00;
ay = (AcY-77)/16384.00; az =
(AcZ-1947)/16384.00; //270,
351, 136 for gyroscope gx =
(GyX+270)/131.07;
gy = (GyY-351)/131.07;
gz = (GyZ+136)/131.07;
/  calculating Amplitute vactor for 3 axis
float Raw_AM = pow(pow(ax,2)+pow(ay,2)+pow(az,2),0.5);
int AM = Raw_AM * 10; // as values are within 0 to 1, I multiplied
/  it by for using if else conditions Serial.println(AM); //Serial.println(PM);
//delay(500);

if (trigger3==true)
{ trigger3count++;
//Serial.println(trigger3count);
if (trigger3count>=10){
angleChange =
pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5); //delay(10);
Serial.println(angleChange);
if ((angleChange>=0) && (angleChange<=10)){ //if orientation changes remains between
0 -10 degrees
fall=true; trigger3=false; trigger3count=0;
Serial.println(angleChange);
}
else{ //user regained normal orientation
trigger3=false; trigger3count=0;
Serial.println("TRIGGER 3 DEACTIVATED");
}
}
}

if (fall==true){ //in event of a fall detection
```

```
Serial.println("FALL DETECTED");
digitalWrite(11, LOW);
delay(20);
digitalWrite(11, HIGH);
fall=false;
/   exit(1);
}
if (trigger2count>=6){ //allow 0.5s for orientation
change trigger2=false; trigger2count=0;
Serial.println("TRIGGER 2 DECACTIVATED");
}
if (trigger1count>=6){ //allow 0.5s for AM to break upper threshold
trigger1=false; trigger1count=0;
Serial.println("TRIGGER 1 DECACTIVATED");
}
if (trigger2==true){ trigger2count++; //angleChange=acos(((double)x*(double)bx+

(double)y*(double)by+(double)z*(double)bz)/(d ouble)AM/(double)BM);

angleChange = pow(pow(gx,2)+pow(gy,2)+pow(gz,2),0.5); Serial.println(angleChange);
if (angleChange>=30 && angleChange<=400){ //if orientation changes by between 80-100
d egrees
trigger3=true; trigger2=false; trigger2count=0;
Serial.println(angleChange);
Serial.println("TRIGGER 3 ACTIVATED");
}
}
if (trigger1==true)
{ trigger1count++;
if (AM>=12){ //if AM breaks upper threshold
(3g) trigger2=true;
Serial.println("TRIGGER 2 ACTIVATED");
trigger1=false; trigger1count=0;
}
}
if (AM<=2 && trigger2==false){ //if AM breaks lower threshold
(0.4g) trigger1=true;
Serial.println("TRIGGER 1 ACTIVATED");
}
//It appears that delay is needed in order not to clog the
port delay(100);
}

void mpu_read(){
Wire.beginTransmission(MPU_addr);
Wire.write(0x3B); // starting with register 0x3B (ACCEL_XOUT_H)
Wire.endTransmission(false);
Wire.requestFrom(MPU_addr,14,true); // request a total of 14 registers
AcX=Wire.read()<<8|Wire.read(); // 0x3B (ACCEL_XOUT_H) & 0x3C (ACCEL_XOUT_L)
AcY=Wire.read()<<8|Wire.read(); // 0x3D (ACCEL_YOUT_H) & 0x3E (ACCEL_YOUT_L)
AcZ=Wire.read()<<8|Wire.read(); // 0x3F (ACCEL_ZOUT_H) & 0x40 (ACCEL_ZOUT_L)
Tmp=Wire.read()<<8|Wire.read(); // 0x41 (TEMP_OUT_H) & 0x42 (TEMP_OUT_L)
GyX=Wire.read()<<8|Wire.read(); // 0x43 (GYRO_XOUT_H) & 0x44 (GYRO_XOUT_L)
GyY=Wire.read()<<8|Wire.read(); // 0x45 (GYRO_YOUT_H) & 0x46 (GYRO_YOUT_L)
GyZ=Wire.read()<<8|Wire.read(); // 0x47 (GYRO_ZOUT_H) & 0x48 (GYRO_ZOUT_L)
}
```