CSE 360

FALL 2023

Group No: 01

Project Name: Elderly Fall Detection System

Group members:

TAHSIN ASHRAFEE SUSMIT (20301088)

ISRATUL HASAN (20301072)

TANZEELA MARIAM (19101193)

FARHANA REZA (16301025)

MD. WASEQ ALVI (20101153)

Submitted to:

Mehedi Hasan Emo
Lecturer
Department of Computer Science and Engineering

Md. Mahadi Hasan
Lecturer
Department of Computer Science and Engineering

Submitted on:

13-11-2023

Introduction

Falls, especially among the older demographic, provide substantial safety risks, frequently resulting in injuries, enduring impairments, or even mortalities. The real-time detection of falls enables prompt reaction, which has an opportunity to reduce the severity of injuries or expedite medical assistance. The achievement of addressing this urgent matter has been attainable through the use of basic electrical components, facilitated by technological advancements. The focus of this idea is the development of a rudimentary fall detection system utilising the Arduino UNO platform in conjunction with the MPU6050 sensor, a widely used gyroscope and accelerometer module, and a PIR motion sensor to interact with humans. The objective is to identify abrupt changes in motion or positioning that are indicative of a fall event. The aforementioned device might be conceptualised as a wearable or attachable apparatus that, upon sensing a fall event, initiates an alarm mechanism. The primary objective is to provide a cost-effective, highly efficient, and readily reproducible system that provides a sense of security to the user and their carers.

Application Area

Fall detectors may be used in many sectors. Hospitals and rehabilitation centres can use the device to help patients recover from surgery or conditions that impede movement. Rapid fall detection might save the lives of senior care facility residents who roam or have conditions like Alzheimer's. The elderly who live alone can use the device to notify their caretakers or family members of emergencies swiftly. This detector can improve safety on construction sites and warehouses as well as in healthcare. Integration with smart home systems or personal emergency response systems (PERS) can show the device's flexibility and usefulness as a whole.

Technology and Tools

- **Arduino UNO:** The central microcontroller board that processes sensor data and determines if a fall has occurred.
- **MPU6050:** A gyroscope and accelerometer combo that will provide orientation and acceleration data to detect falls.
- **PIR MOTION SENSOR:** PIR sensors provide a cost-effective and privacy-conscious solution, and their proper configuration helps mitigate false alarms
- **Arduino IDE (Software):** The primary software environment tailored for Arduino platforms, facilitating code writing, compilation, and upload processes. It provides an intuitive interface, combined with essential tools and features, allowing users to bring their electronic projects to life and debug them efficiently.

Programing Language

Arduino uses a variant of the C++ programming language. The code is written in C++ with an addition of special methods and functions. Moreover, when we create a 'sketch', it is processed and compiled into machine language.

Working Mechanism of Sensors

The MPU6050 is a motion processing unit that combines a 3-axis accelerometer and a 3-axis gyroscope into a single integrated circuit. By employing the piezoelectric effect, the accelerometer is capable of quantifying linear acceleration along the X, Y, and Z axes, therefore discerning between static forces such as gravity and dynamic forces arising from movement or vibrations. Simultaneously, the gyroscope measures angular velocity, detecting alterations in orientation along the roll, pitch, and yaw axes, by utilising the Coriolis effect. In order to enhance the efficiency of connection with microcontrollers, the MPU6050 utilises the I2C protocol to transfer processed data related to acceleration and rotational velocity. The device's flexibility is underscored by its inherent features for detecting motion, estimating direction, and recognising free-fall, rendering it a widely favoured option for applications that need accurate motion analysis or activity monitoring.

The Passive Infrared (PIR) motion sensor functions by sensing alterations in infrared radiation within its designated range. The PIR sensor consists of pyroelectric sensors and a Fresnel lens, enabling it to detect changes in infrared radiation that is released by heated objects, such as humans. When the phenomenon of movement takes place, the aforementioned sensors perceive and record these alterations, subsequently producing an electrical signal as an indication of the detected motion. The signal produced by the sensor has the capability to initiate a range of actions, including the activation of lights or the sounding of alarms. The use of a PIR sensor in a fall detection system enhances its capacity to detect movement, hence facilitating the identification of alterations or the lack of motion. This feature contributes to the system's ability to recognise possible fall incidents when integrated with other sensors.

We will use the threshold-based detection method to check if a parameter is over a threshold during a time interval. Falling causes a big acceleration shift in a split second, and the individual lies stationary for some time without changing orientation. Knowing these details lets us build our algorithm. We will first gather accelerometer data and then determine acceleration magnitude. We require acceleration magnitude to determine how rapidly velocity varies, since acceleration gives us the rate of change.

$$|a| = \sqrt{a_x^2 + a_y^2 + a_z^2}$$

The equation shown above represents the amount of acceleration, with the variables ax, ay, and az denoting the acceleration along the X-, Y-, and Z-axes accordingly, based on the gathered data. Once the amount of acceleration has been calculated, it is necessary to determine if its value falls below the lower threshold. The lower threshold represents the minimal threshold within a predetermined range, while the value progressively increases until reaching the high threshold. Next, it is imperative to verify if it surpasses the upper barrier within a certain time frame, such as 500 milliseconds. If all of the aforementioned circumstances are identified as being true, it becomes necessary to assess whether there is a shift in orientation within a timeframe of 500 milliseconds. Typically, this holds true if the aforementioned conditions are indeed met. Subsequently, an assessment is conducted to ascertain the persistence of orientation over a certain duration. If the answer is affirmative, then a decline occurred. The presence of a loop in a programme entails that in the event of a false condition at any given iteration, the programme will go back to its initial state and re-execute.

Connection with ICs

The circuit is constructed with an Arduino UNO microcontroller, an MPU6050 accelerometer and gyroscope breakout module, and a PIR (Passive Infrared) motion sensor. In the process of data collection, the MPU6050 module is utilised to provide analogue input to the Arduino UNO. The MPU6050 module outputs six values, consisting of three accelerometer values and three gyroscope readings. The technology employed in this system is based on microelectromechanical systems (MEMS) utilising sensors, with communication facilitated by the I2C protocol. Sensors of this nature have extensive use in several domains such as smartphones, robotics, 3D modelling, unmanned aerial vehicles (UAVs), and other related fields. The MPU6050 consists of a triaxial gyroscope, a triaxial accelerometer, a digital motion processor, and an integrated temperature sensor. The integration of a three-axis magnetometer through the I2C connection results in a comprehensive output of nine-axis motion fusion.

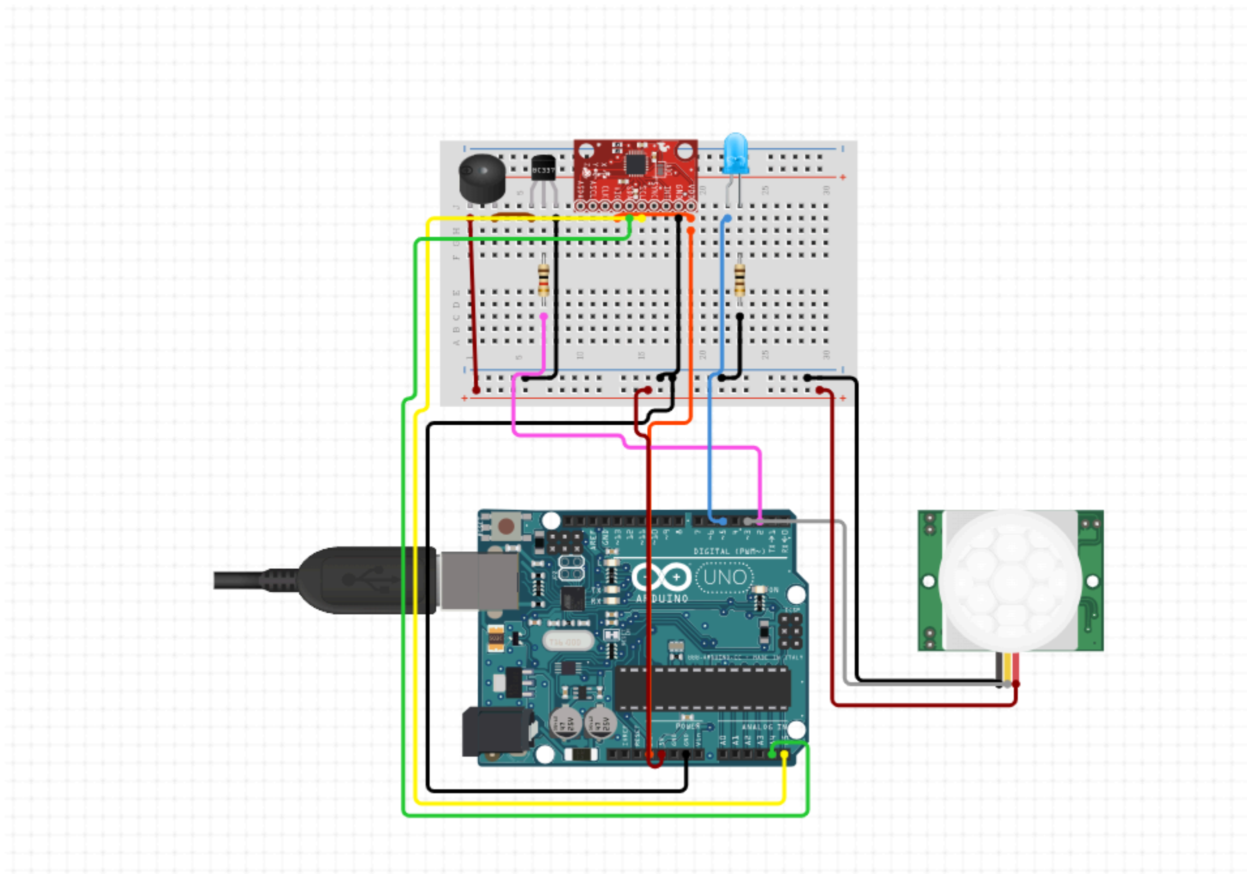


Figure: Connecting the Arduino UNO, the MPU6050 and PIR MOTION sensor

Position the Passive Infrared (PIR) sensor in a suitable area that maximises its ability to detect motion efficiently. Conduct an assessment to determine the extent and precision of its capabilities, making necessary modifications accordingly. The output of the sensor may be utilised to initiate a range of activities, such as generating notifications, triggering alarm systems, or incorporating it into a fall detection system to spot motion that potentially signifies a fall event. The process of connecting is rather straightforward. The data receiving is facilitated by using the A4 and A5 pins of the Arduino UNO. Moreover, the Arduino's digital pin 11 serves as an output pin that is set to a high state when detecting a fall. The pin may be utilised to establish a connection between an Wi-Fi module or bluetooth module. The subsequent task involves creating a recipe on IFTTT for a button and thereafter verifying the hardware.

Data Flow from Sensors through ICs to I/O Devices has been described in four steps. They are:

1. MPU6050 senses the movement and orientation.
2. Arduino UNO reads the data from MPU6050.
3. The algorithm processes the data.
4. If a fall is detected, an output can be sent - e.g., to an LED or a buzzer .

Code

```
// Include Libraries
#include "Arduino.h"
#include "Buzzer.h"
#include "LED.h"
#include "MPU6050.h"
#include "Wire.h"
#include "I2Cdev.h"
#include "PIR.h"


// Pin Definitions
#define BUZZER_PIN_SIG	2
#define LEDB_PIN_VIN	5
#define PIR_PIN_SIG	3



// Global variables and defines
int16_t mpu6050Ax, mpu6050Ay, mpu6050Az;
int16_t mpu6050Gx, mpu6050Gy, mpu6050Gz;
// object initialization
Buzzer buzzer(BUZZER_PIN_SIG);
LED ledB(LEDB_PIN_VIN);
MPU6050 mpu6050;
PIR pir(PIR_PIN_SIG);


// define vars for testing menu
const int timeout = 10000;	//define timeout of 10 sec
char menuOption = 0;
long time0;
```

```
// Setup the essentials for your circuit to work. It runs first every time your circuit is powered with electricity.
void setup()
{
    // Setup Serial which is useful for debugging
    // Use the Serial Monitor to view printed messages
    Serial.begin(9600);
    while (!Serial) ; // wait for serial port to connect. Needed for native USB
    Serial.println("start");

    Wire.begin();
    mpu6050.initialize();
    menuOption = menu();

}

// Main logic of your circuit. It defines the interaction between the components you selected. After setup, it runs over and over again, in an eternal loop.
void loop()
{


    if(menuOption == '1') {
    // Buzzer - Test Code
    // The buzzer will turn on and off for 500ms (0.5 sec)
    buzzer.on();       // 1. turns on
    delay(500);             // 2. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter delay in milliseconds.
    buzzer.off();      // 3. turns off.
    delay(500);             // 4. waits 500 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter delay in milliseconds.
    }
    else if(menuOption == '2') {
    // LED - Basic Blue 5mm - Test Code
    // The LED will turn on and fade till it is off
    for(int i=255 ; i> 0 ; i -= 5)
    {
        ledB.dim(i);                      // 1. Dim Led
        delay(15);                               // 2. waits 5 milliseconds (0.5 sec). Change the value in the brackets (500) for a longer or shorter delay in milliseconds.
    }
    ledB.off();                        // 3. turns off
```

```
    }
    else if(menuOption == '3') {
    // SparkFun MPU-6050 - Accelerometer and Gyro - Test Code
    mpu6050.getMotion6(&mpu6050Ax, &mpu6050Ay, &mpu6050Az, &mpu6050Gx, &mpu6050Gy,
&mpu6050Gz);   //read accelerometer and gyroscope raw data in three axes
    double mpu6050Temp = ((double)mpu6050.getTemperature() + 12412.0) / 340.0;
    Serial.print("a/g-\t");
    Serial.print(mpu6050Ax); Serial.print("\t");
    Serial.print(mpu6050Ay); Serial.print("\t");
    Serial.print(mpu6050Az); Serial.print("\t");
    Serial.print(mpu6050Gx); Serial.print("\t");
    Serial.print(mpu6050Gy); Serial.print("\t");
    Serial.print(mpu6050Gz); Serial.print("\t");
    Serial.print(F("Temp- "));
    Serial.println(mpu6050Temp);
    delay(100);

    }
    else if(menuOption == '4') {
    // Infrared PIR Motion Sensor Module - Test Code
    bool pirVal = pir.read();
    Serial.print(F("Val: ")); Serial.println(pirVal);

    }

    if (millis() - time0 > timeout)
    {
        menuOption = menu();
    }

}



// Menu function for selecting the components to be tested
// Follow serial monitor for instrcutions
char menu()
{

    Serial.println(F("\nWhich component would you like to test?"));
    Serial.println(F("(1) Buzzer"));
    Serial.println(F("(2) LED - Basic Blue 5mm"));
    Serial.println(F("(3) SparkFun MPU-6050 - Accelerometer and Gyro"));
```

```
  Serial.println(F("(4) Infrared PIR Motion Sensor Module"));
  Serial.println(F("(menu) send anything else or press on board reset button\n"));
  while (!Serial.available());

  // Read data from serial monitor if received
  while (Serial.available())
  {
    char c = Serial.read();
    if (isAlphaNumeric(c))
    {

      if(c == '1')
                    Serial.println(F("Now Testing Buzzer"));
              else if(c == '2')
                    Serial.println(F("Now Testing LED - Basic Blue 5mm"));
              else if(c == '3')
                    Serial.println(F("Now Testing SparkFun MPU-6050 - Accelerometer and
Gyro"));
              else if(c == '4')
                    Serial.println(F("Now Testing Infrared PIR Motion Sensor Module"));
      else
      {
        Serial.println(F("illegal input!"));
        return 0;
      }
      time0 = millis();
      return c;
    }
  }
}
```

Estimated Cost Analysis

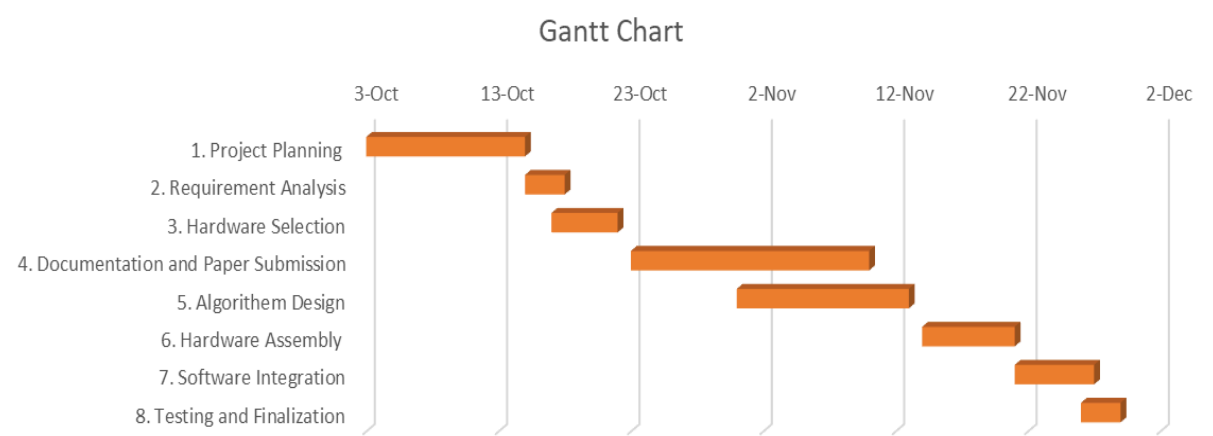
Elements	Cost in Taka	Bought From
Arduino UNO	840	RoboticsBD
MPU6050	200	RoboticsBD
PIR MOTION SENSOR	95	RoboticsBD
Active Buzzer 5V	19	RoboticsBD
Led, Resistor, Jumper Wire, Breadboard	324	RoboticsBD

Total: Roughly **1500 Taka**

Responsibilities of Each Member in Tabular Format:

Name	ID	Details
TAHSIN ASHRAFEE SUSMIT	20301088	Planning, Coding, Hardware, Report Writing
ISRATUL HASAN	20301072	Report Writing, Diagrams, Coding, Hardware
TANZEELA MARIAM	19101193	Requirement Analysis, Report Writing, Coding
FARHANA REZA	16301025	Planning, Analysis, Report Writing, Coding
MD. WASEQ ALVI	20101153	Planning, Requirement Analysis, Coding.

Workplan (Gantt Chart)



	8. Testing and Finalization	7. Software Integration	6. Hardware Assembly	5. Algorithem Design	4. Documentation and Paper Submission	3. Hardware Selection	2. Requirement Analysis	1. Project Planning
Start Date	26-Nov	21-Nov	14-Nov	31-Oct	23-Oct	17-Oct	15-Oct	3-Oct
Days to Complete	3	6	7	13	18	5	3	12

Conclusion

When it comes to monitoring and detecting falls, this simple fall detector offers a solution that is not only effective but also cost-effective. This is a significant benefit. Even though this is a very simple design, it has the potential to be expanded with extra features like wireless alerts and interaction with a number of different emergency response systems. The efficiency and cost-effectiveness of the basic model make it accessible to a wide range of users, particularly those who require such monitoring due to age or medical conditions.Furthermore, the addition of wireless notifications ensures that caregivers or family members can respond promptly in case of a fall, enhancing the user's safety and well-being. The possibility of integration with emergency response systems elevates the device to a higher level of reliability, offering peace of mind to both users and their support networks. Thus, it can be a valuable asset ensuring the safety and security of individuals prone to falls.

References

1. Interface MPU6050 Accelerometer and Gyroscope Sensor with Arduino by Last Minute Engineers
   https://lastminuteengineers.com/mpu6050-accel-gyro-arduino-tutorial/?fbclid=IwAR0Ig5aYy3IekJuGCu9ut0cXPE9nXAAzXnFtbH8HoGJpjXOM7u6ppnWxCUY
2. How to Build a Fall Detector With Arduino by MAKER PRO
   https://maker.pro/arduino/tutorial/how-to-build-a-fall-detector-with-arduino?fbclid=IwAR2vv-OnzPCGpcpi9qFkep0HGcINIAmVnR6R9uBu7hwdaI2q64pulbYgFFA