FALL DETECTION USING WEARABLE SENSOR

Submitted by

Team 10

Palguna Chaithanya Kurakula,

(1001974583)

Pranaya Reddy Nalabolu,

(1001965344)

Sreya Kotha,

(1002007076)

Maneesh Reddy Polamreddy,

(1001967226)

Professor: Vp Nguyen

INTRODUCTION:

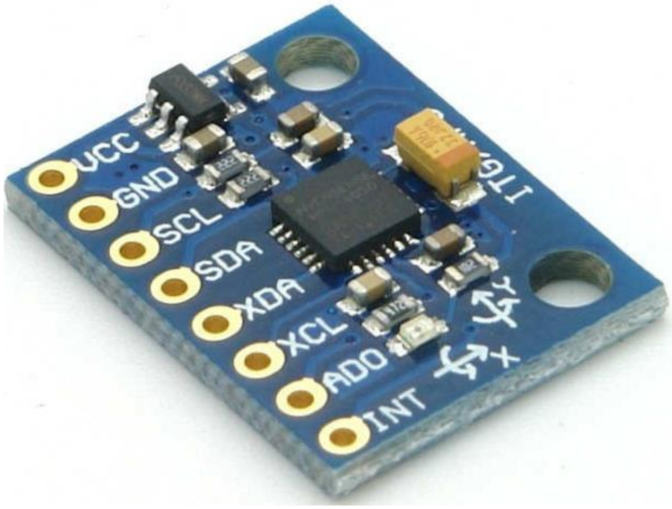
Elderly persons frequently fall and suffer fatal or life-threatening injuries as a result. Every time a fall is detected, it can alert the concerned person or a family member, lowering the likelihood that medical assistance would be delayed. As a result, numerous different kinds of automated fall detection systems have been developed. These days, smartwatches, fitness trackers, and other wearables all come equipped with fall detectors. We're going to use the NodeMCU and MPU6050 sensor module to develop a fall detection device. The MPU6050 sensor module has an accelerometer and a gyroscope. The accelerometer offers data on the angular parameter, such as the three-axis data, while the gyroscope is utilized to detect orientation. We will contrast the amplitude of the acceleration with the threshold value in order to identify the fall. The device will notify the individual concerned through the buzzer if fall is detected.

PROJECT OVERVIEW:

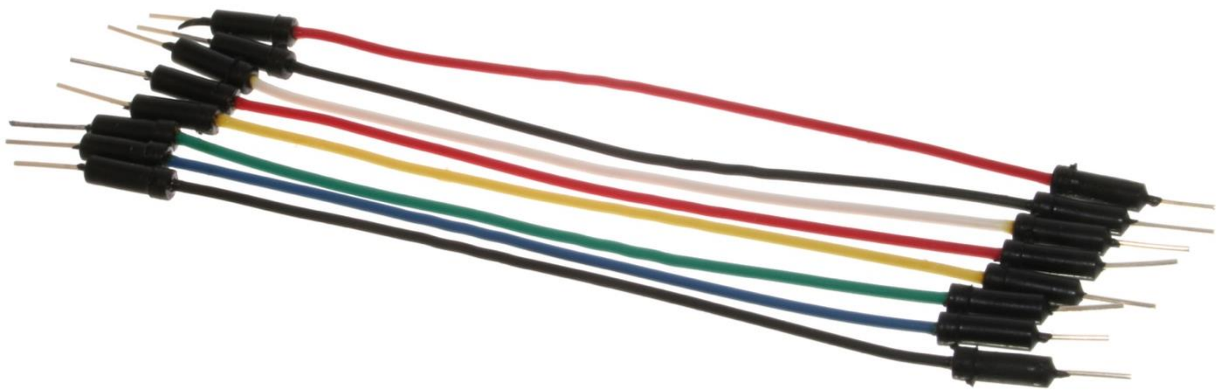
Hardware Required:

NodeMCU ESP8266:
NodeMCU is an open-source platform based on ESP8266 which can connect objects and let data transfer using the Wi-Fi protocol. In addition, by providing some of the most important features of microcontrollers such as GPIO, PWM, ADC, and etc., it can solve many of the project's needs alone.

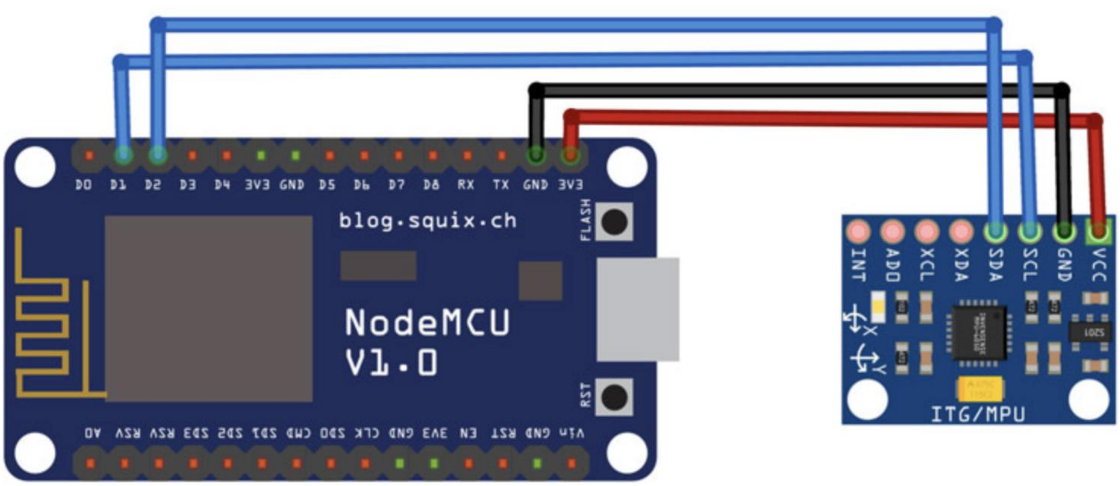
MPU6050 Accelerometer:
The MPU6050 sensor module is a full 6-axis Micro-Electro-Mechanical Systems (MEMS) device that can measure a variety of motion-related data, including acceleration, velocity, direction, displacement, and gyroscope. In addition to this, it features a separate temperature sensor built inside the chip.
The MPU6050 module is compact, uses little power, has a high repetition rate, is shock-tolerant, and is inexpensive for end users. With its I2C bus and auxiliary I2C bus interface, the MPU6050 can easily obstruct the operation of other sensors like magnetometers and microcontrollers.



Jumper Wires:
These wires are used to connect the microcontroller and the sensor. There are male to male as well as male to female jumper wires for IOT modules to jump between headers of the board.



Circuit Diagram:



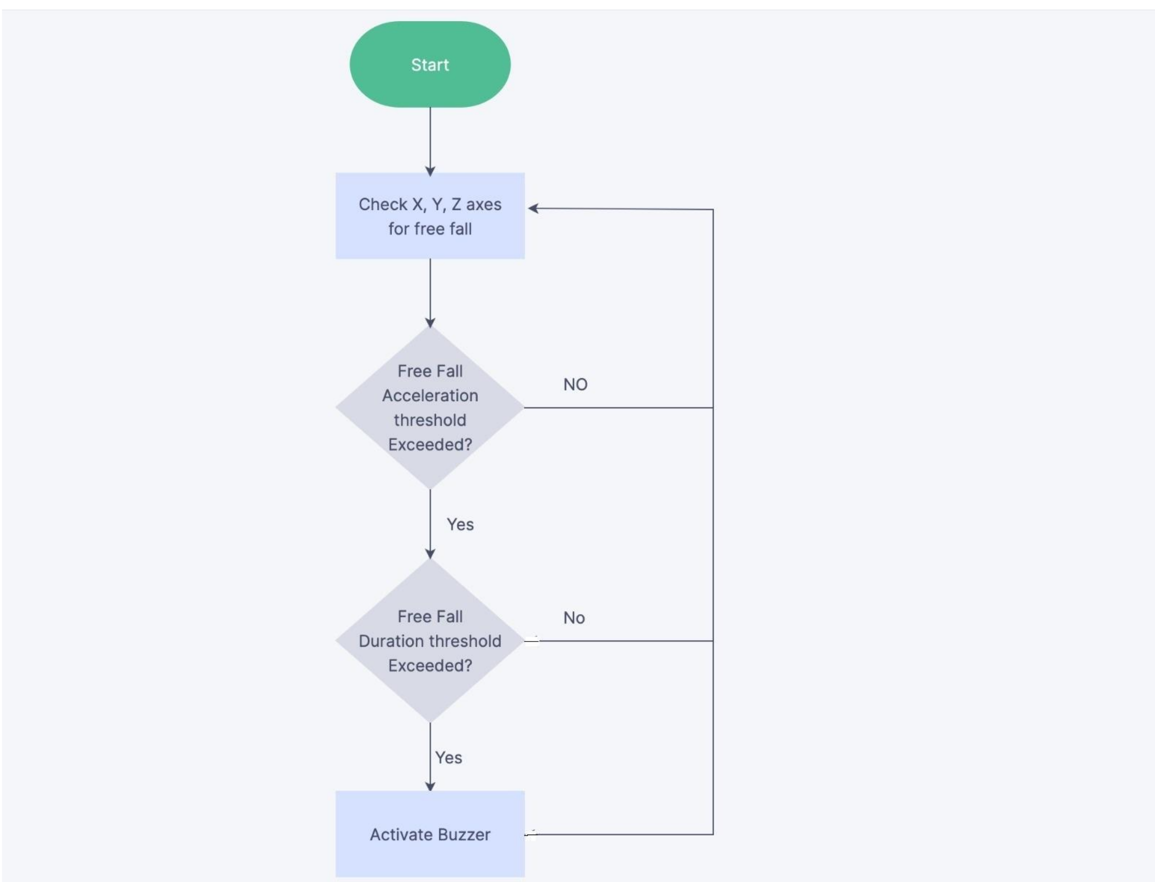
Since the MPU6050 operates on the I2C protocol, connecting NodeMCU and MPU6050 just requires two wires. While the VCC and GND pins of MPU6050 are linked to 3.3V and GND of NodeMCU, respectively, the SCL and SDA pins of MPU6050 are connected to NodeMCU's D1 and D2 pins.

IMPLEMENTATION:

This section describes the hardware platform used to build a prototype of the system and the software running on the smart device and on the external sensing unit when provisioned.

Project Approach:

Here first we connect the NodeMCU microcontroller with the MPU6050 IMU sensor using jumper wires, code is dumped using the USB into the sensor using the breadboard. When fall is detected, a buzzer will be activated alerting the individual concerned.



Hardware Implementation:

Both dynamic and static acceleration components are included in the overall acceleration vector Acc. is calculated from sampled data as indicated in Eq. (1)

$$Acc = \sqrt{(A_x)^2+(A_y)^2+(A_z)^2} \quad (1)$$

Where A_x, A_y, A_z are the acceleration in the x, y, z axes, respectively.
Similarly, to the acceleration, the angular velocity is calculated from sampled data as indicated in Eq. (2)

$$w=\sqrt{(w_x)^2+(w_y)^2+(w_z)^2} \quad (2)$$

Where w_x, w_y, w_z the acceleration in the x, y, z axes, respectively.

Tri-axial accelerometers' measured acceleration, Acc, is constant at rest, and their measured angular velocity, 0o/s, is zero. When a subject falls, the acceleration changes quickly, and the angular velocity generates a range of signals along the fall direction. Slow-moving falls will be missed by the Fall Index (Acc) since it demands a high sample frequency and rapid acceleration changes. As a result, Acc is rarely employed until we want to compare the performances of our systems with those of earlier research that used the same positions but different speeds and accelerations. The following equations are used to obtain the lower and upper fall thresholds for the acceleration and angular velocity used to identify the fall.

Lower fall threshold (LFT): The negative peaks for each recorded activity's outcome are known as the signal's lower peak values (LPVs). The level of the smallest magnitude lower fall peak (LFP) recorded serves as the LFT for the acceleration signals.
Upper fall threshold (UFT): The recorded signals' positive peaks are referred to as the signals' upper peak values for each recorded activity (UPVs). The level of the smallest magnitude UPV collected was used to set the UFT for each acceleration and angular velocity signal. The UFT is related to the maximum impact force that a body segment experiences during the impact phase of a fall.

The two categories of threshold-based fall detection algorithms are typically based on the LFT comparison and the UFT comparison of acceleration data, respectively. Even though previous research has produced some noteworthy outcomes, the accuracy is still below expectations. Adjusting the UFT and LFT in this investigation revealed that the performance was 83.33% and 67.08%, respectively. The flowchart of our algorithm, which was incorporated into the Arduino UNO program in C, is shown in Figure 3. This program was created to read an analogue variable from its port as an additional adjustment to fix the upper and lower ACC.

Software Used:

Arduino IDE:

The Arduino Integrated Development Environment - or Arduino Software (IDE) - contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions and a series of menus. To upload programs and communicate with them, it establishes a connection with the Arduino hardware.

Programming for NodeMCUesp8266 can be done by using this IDE and programming. The language is basically based on computer languages C, C++ and java. So, in the NodeMCU chip we are going to program using Arduino IDE.

Demonstration:

We choose the correct board and COM port before uploading our code to the board. Go to Tools > Board and select ESP8266 Module.

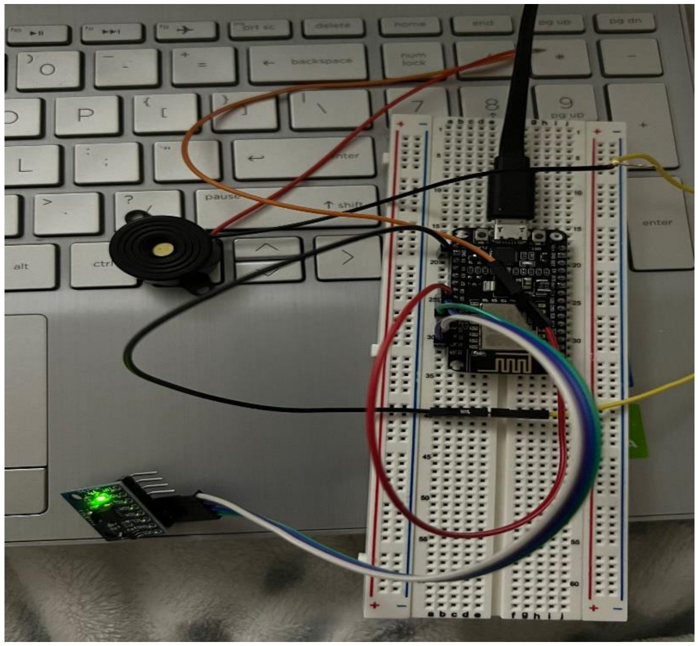
Then, go to Tools > Port and select the appropriate port through which the board is connected.
To upload the code to ESP8266 development board, click the upload button.

In the Arduino IDE, open up the serial monitor and set the baud rate to 115200. Now jerk the MPU6050 sensor downwards with some force. The fall will be detected, if the magnitude is greater than the threshold value set by us. We can view it in the serial monitor as follows:

```
1
TRIGGER 1 ACTIVATED
TRIGGER 1 DECACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
3
3
FALL DETECTED
Connecting to maker.ifttt.com
Requesting URL: /trigger/fall_detect/with/key/dfqfJoPY-FnaOvhF9IDzez

closing connection
```

Hardware setup:



Contribution:

Our project’s aim is to find a suitable yet cheaper alternative to expensive fall detection modules available in the market. Currently in the market such fall detecting IOT devices are very expensive, so we decided to make it at least affordable for elderly.

Tasks assigned to each individual:
Palguna Chaithanya: Hardware setup, Code for the microcontroller and report.
Sreya Kotha: Hardware setup, Code for the microcontroller and report.
Pranaya Reddy: Hardware setup, Code for the microcontroller and report.
Maneesh Reddy: Hardware setup, Code for the microcontroller and report.

CONCLUSION:

The several fall feature parameters of the 6 axes acceleration were introduced and applied according to the algorithm. Possible falls were chosen through the simple threshold and then applied to the MPU to solve the problems such as deviation of interpersonal falling behavioral patterns and similar fall actions. As a summary, our project presents a holistic solution for fall detection by immediately alerting the concerned individual with a buzzer.

LESSONS LEARNT:

- Start early when we are trying to do something new.
- While in the process of doing project, deep understanding of the things required for our project is necessary.
- Correct understanding of modules used in implementing the prototype.
- Algorithms to identify the difference between the exercise and the fall. Understanding the hardware components and make use of it.
- As we didn't differentiate between exercise and fall, we further want to use the machine learning algorithm to differentiate between the events and improve our project by sending an SMS to the registered mobile number.
- We can use deep neural networks and IMU sensor to get actual fall detected.
- Currently our understanding was different so learnt that trying something new need more time than the time required to know some project.
- Connecting the Wi-Fi module(microcontroller) to the mobile device was a big task since open Wi-Fi is unstable
- The sensor we used was so sensitive so the results may vary.
- Better understanding of the key technical challenges that we are trying to resolve in the project

We learnt so many things while doing this project such as team involvement, proper implementation of our idea and time management.

REFERENCES:

1. He W, Goodkind D, Kowal P (2016) U.S. Census Bureau, International Population Reports, P95/16-1, An Aging World: 2015. U.S. Government Publishing Office, Washington, DC
2. Yacchirema DC, Sarabia-Jácome D, Palau CE, Esteve M (2018) A Smart System for sleep monitoring by integrating IoT with big data analytics. IEEE Access, p 1
3. Robie K (2010) Falls in older people: risk factors and strategies for prevention. JAMA 304(17):1958–1959
4. Practical fall detection based on IoT technologies: A survey author links open overlay Nassim Mozaffaric Javad Rezazadeh Reza Farahbakhsh Samaneh Yazdanic Kumbesan Sandrasegaranb (2019).
5. Fall detection system for elderly people using IoT and Big Data Author links open overlay Diana Yacchirema Jara Suárez de Pugaa Carlos Palaua Manuel Esteve (2018).

GitHub link:

https://github.com/kpchaithanya/Special-Topics-in-Networking-Final-Project.git

YouTube Link:

https://youtu.be/01w-rhucjoo