

**DEPARTMENT OF INFORMATION TECHNOLOGY
FACULTY OF ENGINEERING & TECHNOLOGY**

IOT PROJECT REPORT

SUBJECT TITLE : INTERNET OF THINGS

SUBJECT CODE: 15IT422E

SUBMITTED TO: Prof Kayalvizhi Jayavel

FALL DETECTION SYSTEM

Sarnaam Musthafa Karipp RA1611008010086

Department of Information Technology



**SRM University, SRM Nagar, Kattankulathur-603203
Kanchipuram District, Tamil Nadu**

LINKS TO GITHUB AND YOUTUBE:

YouTube:

<https://www.youtube.com/watch?v=-hb-8XGw5ZU>

Github:

<https://github.com/sarnaam/falldetection>

ACKNOWLEDGEMENT

We would like to express my sincere gratitude to our IoT Professor Mrs Kayalvizhi Jayavel for being out there at every step of our course and guiding us all along the way to be capable of moulding our ideas into smart projects.

We would also thank Mr Hazim Hamza of Intel Corporation, Ireland who helped me a hands-on , weeks of explanations and demos which helped us visualize how IoT is being used in the industry .

We would also like to thank our parents, without whom I wouldn't be able to anything in any regard.

ABSTRACT

Fall detection is a major challenge in the public healthcare domain, especially for the elderly as the decline of their physical fitness, and timely and reliable surveillance is necessary to mitigate the negative effects of falls. This paper develops a novel fall detection system based on a wearable device. The system monitors the movements of human body, recognizes a fall from normal daily activities by an effective quaternion algorithm, and automatically sends request for help to the caregivers with the patient's location.

Falling is an external aspect that can lead to death for the elderly. With so many activities they can do will increase the likelihood of falling. A fall detection device is designed to minimize post-fall risk. An MPU6050 sensor with 3 axis accelerometer and 3 gyroscope axis is used to detect the activities of the elderly. This research is expected to recognize the falling forward movement, falling aside, falling backward, sitting, sleeping, squatting, upstairs, down stairs and praying. This project module is a combined integration of NodeMCU ESP8266 and MPU6050 for gyroscope and accelerometer for retrieving the fall detection details.

HARDWARE REQUIRED:

- 1 * Nodemcu ESP8266 module.
- 1 * MPU6050
- 1 * Breadboard
- Jumper Wires
- A Computer.
- A Smart phone.
- Micro USB Cable

SOFTWARE REQUIRED:

- Arduino IDE.
- ESP8266 library.
- Cayenne MQTT Library.
- MPU6050 Library.
- Cayenne MQTT Library.

MISCELLANEOUS:

- WiFi Internet connection.
- Soldering Tool.
- Soldering Wires.

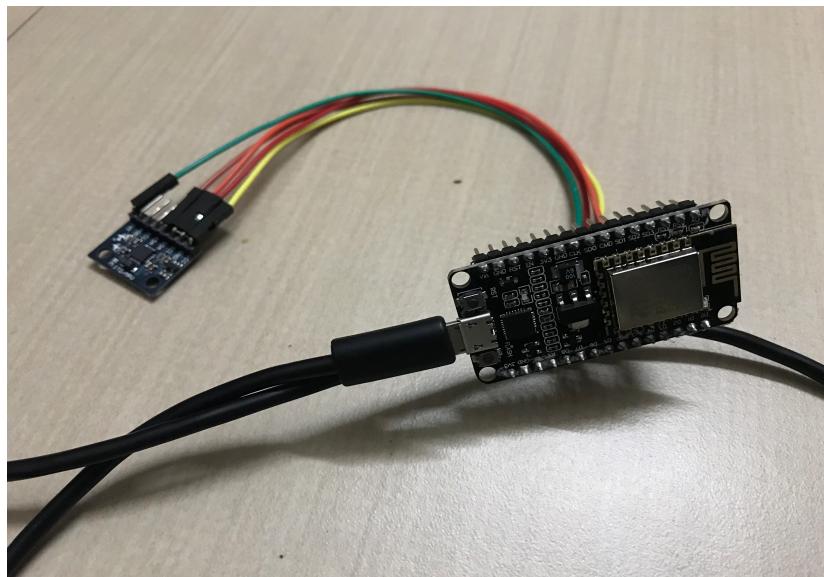
TOTAL COST OF COMPONENTS:

- Rs.1000-1500

SYSTEM OVERVIEW

The main component of the setup is the Nodemcu ESP8266 module. All the other hardware components are connected to the Nodemcu. The board is programmed in Arduino IDE and uses the ESP8266, MPU6050, Cayenne MQTT libraries. These libraries have been added to the Arduino IDE.

The MPU6050 module with gyroscope and accelerometer is directly connected to the Nodemcu using a micro USB cable. This MPU6050 is used to detect the fall achieved when hits the condition and is signalled by the programmed NodeMCU ESP8266 with a trigger as an Email using Cayenne support.



Circuit Of NodeMCU and MPU6050

Programming NodeMCU and Setting Up Cayenne Dashboard:

- Import MPU6050 Library to the Arduino IDE
-
- Fill your cayenne token, ssid, and wifi password to the code.
-
- Upload falldetection code to your NodeMCU
-
- Setup your cayenne dashboard, add a custom widget, choose two state widget and give it name, choose virtual pin V8 for connectivity, and press add widget.
-
- Now press the trigger, give trigger's name, on if add the widget added before, on then fill email with your email, and press save.

Code

```
#include <Wire.h>
#include <MPU6050.h>
#define CAYENNE_PRINT Serial
#include <CayenneMQTTESP8266.h>

MPU6050 mpu;

boolean freefallDetected = false;

char ssid[] = "Sarnaam";
char wifiPassword[] = "sarn1234";

char username[] = "f3ee8980-4e2a-11e9-9622-9b9aecba453";
char password[] = "af47a27852e8b4a1c8081e8b9c37c1e080765ff8";
char clientID[] = "df55b780-5877-11e9-81a2-d1fdd4219210";

void setup()
{
    Serial.begin(115200);
    Cayenne.begin(username, password, clientID, ssid, wifiPassword);
    Serial.println("Initialize MPU6050");
```

```
while(!mpu.begin(MPU6050_SCALE_2000DPS, MPU6050_RANGE_16G))  
{  
    Serial.println("Could not find a valid MPU6050 sensor, check wiring!");  
    delay(500);  
}  
  
mpu.setAccelPowerOnDelay(MPU6050_DELAY_3MS);  
  
mpu.setIntFreeFallEnabled(true);  
mpu.setIntZeroMotionEnabled(false);  
mpu.setIntMotionEnabled(false);  
  
mpu.setDHPFMode(MPU6050_DHPF_5HZ);  
  
mpu.setFreeFallDetectionThreshold(17);  
mpu.setFreeFallDetectionDuration(2);  
  
checkSettings();  
  
attachInterrupt(12, doInt, RISING);  
}  
  
void doInt()
```

```
{  
    freefallDetected = true;  
}  
  
void checkSettings()  
{  
    Serial.println();  
  
    Serial.print(" * Sleep Mode:      ");  
    Serial.println(mpu.getSleepEnabled() ? "Enabled" : "Disabled");  
  
    Serial.print(" * Motion Interrupt:  ");  
    Serial.println(mpu.getIntMotionEnabled() ? "Enabled" : "Disabled");  
  
    Serial.print(" * Zero Motion Interrupt:  ");  
    Serial.println(mpu.getIntZeroMotionEnabled() ? "Enabled" : "Disabled");  
  
    Serial.print(" * Free Fall Interrupt:   ");  
    Serial.println(mpu.getIntFreeFallEnabled() ? "Enabled" : "Disabled");  
  
    Serial.print(" * Free Fal Threshold:    ");  
    Serial.println(mpu.getFreeFallDetectionThreshold());  
  
    Serial.print(" * Free FallDuration:     ");  
    Serial.println(mpu.getFreeFallDetectionDuration());
```

```
Serial.print(" * Clock Source:      ");
switch(mpu.getClockSource())
{
    case MPU6050_CLOCK_KEEP_RESET:   Serial.println("Stops the clock and
keeps the timing generator in reset"); break;
    case MPU6050_CLOCK_EXTERNAL_19MHZ: Serial.println("PLL with external
19.2MHz reference"); break;
    case MPU6050_CLOCK_EXTERNAL_32KHZ: Serial.println("PLL with external
32.768kHz reference"); break;
    case MPU6050_CLOCK_PLL_ZGYRO:    Serial.println("PLL with Z axis gyroscope
reference"); break;
    case MPU6050_CLOCK_PLL_YGYRO:    Serial.println("PLL with Y axis gyroscope
reference"); break;
    case MPU6050_CLOCK_PLL_XGYRO:    Serial.println("PLL with X axis
gyroscope reference"); break;
    case MPU6050_CLOCK_INTERNAL_8MHZ: Serial.println("Internal 8MHz
oscillator"); break;
}
```

```
Serial.print(" * Accelerometer:      ");
switch(mpu.getRange())
{
    case MPU6050_RANGE_16G:        Serial.println(" +/- 16 g"); break;
    case MPU6050_RANGE_8G:         Serial.println(" +/- 8 g"); break;
    case MPU6050_RANGE_4G:         Serial.println(" +/- 4 g"); break;
```

```
case MPU6050_RANGE_2G:           Serial.println("+- 2 g"); break;
}

Serial.print(" * Accelerometer offsets:   ");
Serial.print(mpu.getAccelOffsetX());
Serial.print(" / ");
Serial.print(mpu.getAccelOffsetY());
Serial.print(" / ");
Serial.println(mpu.getAccelOffsetZ());

Serial.print(" * Accelerometer power delay: ");
switch(mpu.getAccelPowerOnDelay())
{
    case MPU6050_DELAY_3MS:        Serial.println("3ms"); break;
    case MPU6050_DELAY_2MS:        Serial.println("2ms"); break;
    case MPU6050_DELAY_1MS:        Serial.println("1ms"); break;
    case MPU6050_NO_DELAY:         Serial.println("0ms"); break;
}

Serial.println();
}

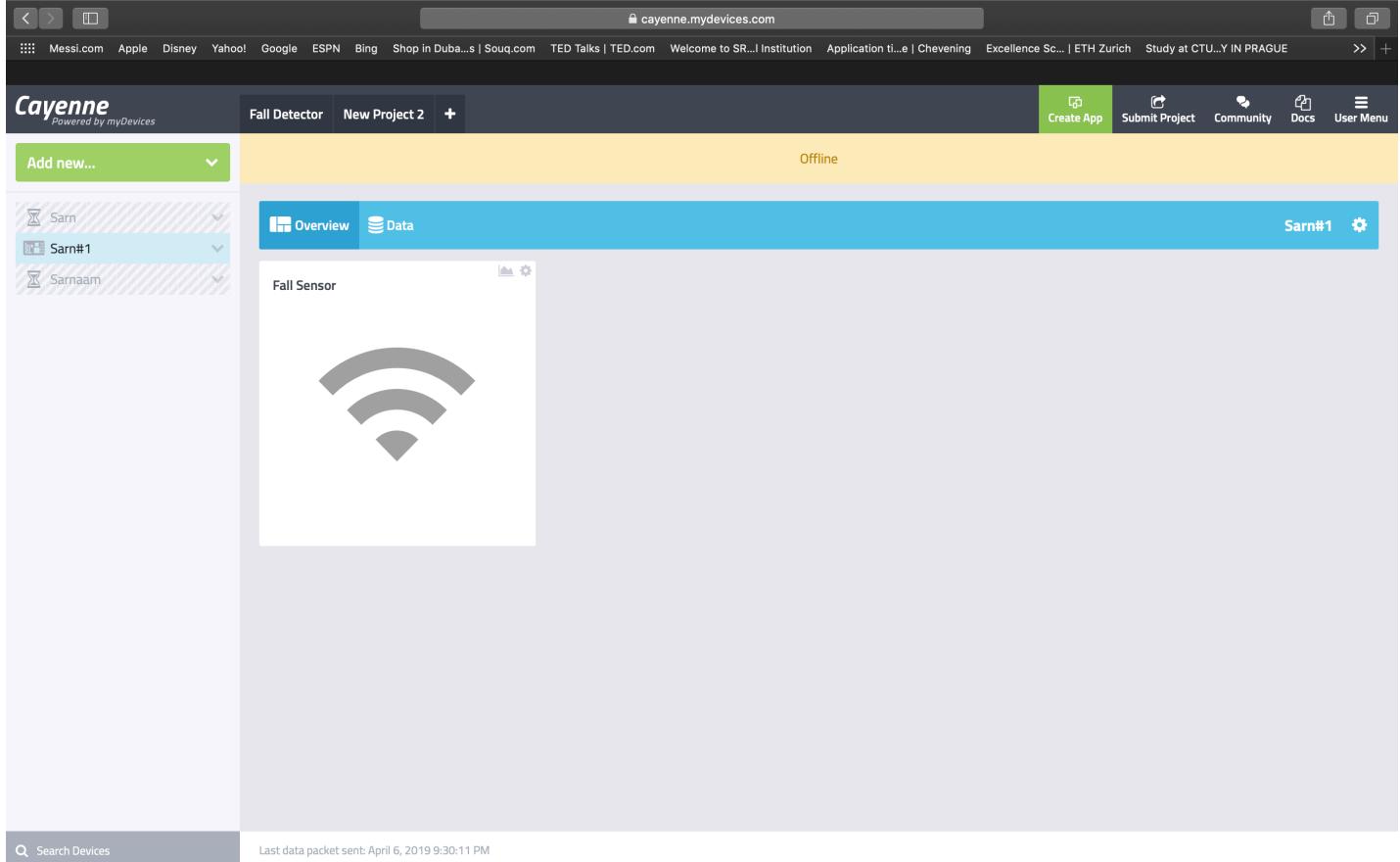
void loop()
{
```

```
Cayenne.loop();  
}  
  
CAYENNE_OUT(V8)  
{  
    Vector rawAccel = mpu.readRawAccel();  
    Activites act = mpu.readActivites();  
  
    if (freefallDetected)  
    {  
        Cayenne.virtualWrite(V8, act.isFreeFall); //virtual pin  
        delay(100);  
    }  
}
```

Cayenne Dashboard, Widgets and Triggers

We give our WiFi SSID and password inside the code before compiling and uploading it to the Nodemcu. Also in order to connect to the Cayenne database, we give the Cayenne username, Cayenne password and Cayenne client ID so that it can connect via MQTT.

We then create the two state widget to make triggers in it. The trigger states that when a fall is signalled by the gyro and accelerometer module, it send a detected information to NodeMCU board connected to the Cayenne network which then provides a Triggered notification to the mail given on the trigger.



Fall Sensor

General

Widget Name
Fall Sensor

Channel
V8

Choose Widget
Choose Widget

Choose Display
Icon

Choose Icon
 **Wi-Fi**

Choose Unit
Digital (0/1)

Remove**Save**

cayenne.mydevices.com

Messi.com Apple Disney Yahoo! Google ESPN Bing Shop in Duba...s | Souq.com TED Talks | TED.com Welcome to SR...I Institution Application ti...e | Chevening Excellence Sc... | ETH Zurich Study at CTU...Y IN PRAGUE >> +

Cayenne Powered by myDevices

Add new... ▾

Triggers

My Triggers New Trigger

if  **Sarn#1** **then** notify...

Sarnaam - Sarn#1

Last ran 4/6/19 9:30 PM Ran 3 times

Fall

Search Devices

Screenshot of the Cayenne web interface showing a trigger configuration:

Triggers

Fall

if Sarn#1
Sarnaam - DigitalActuator - Channel V8

Fall Sensor

On (1)
 Off (0)

then notify...

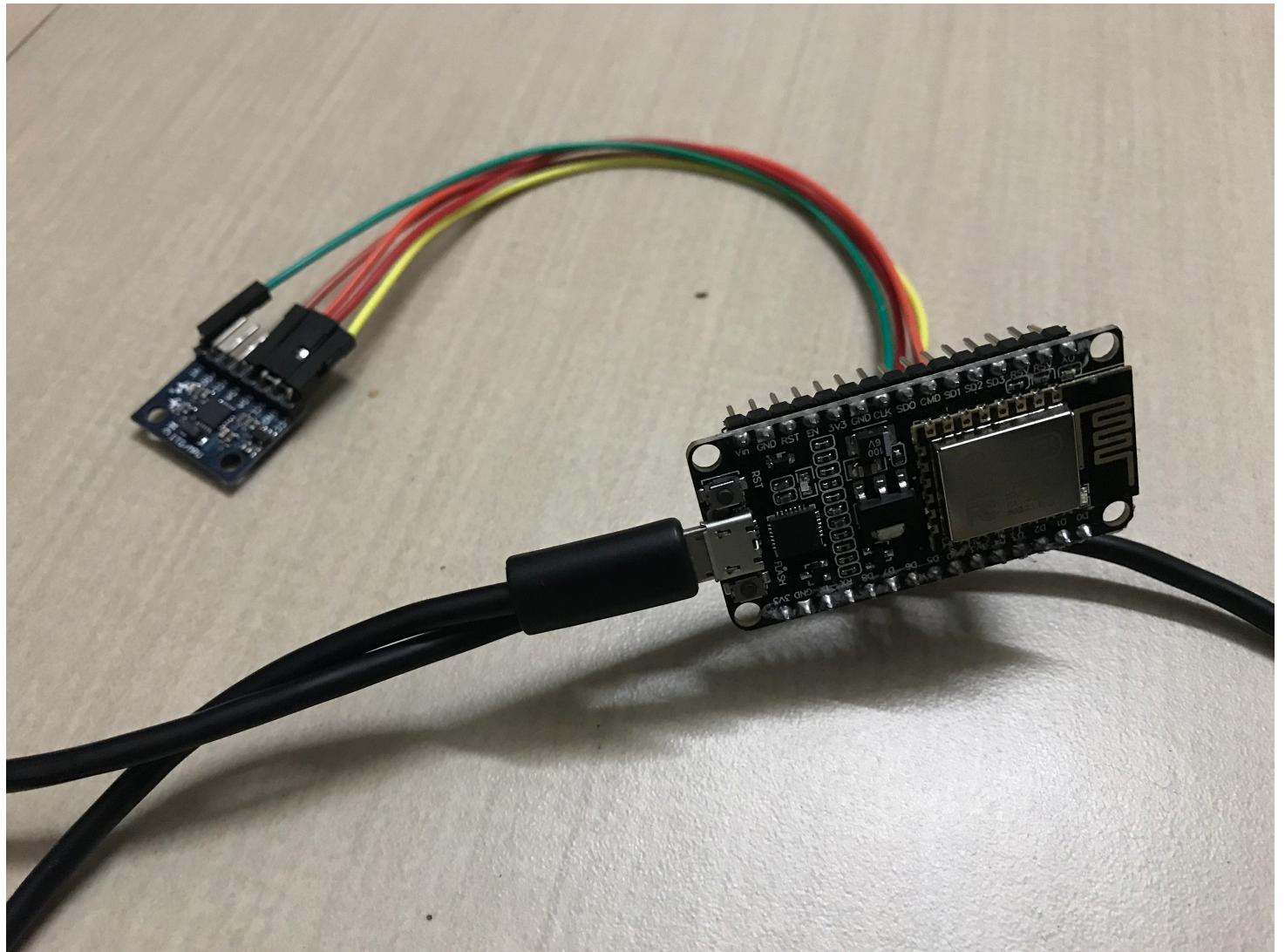
Add custom recipient

fall Detected
sarnaammustafa_mustafa@srmuniv.edu.in

Add more recipients?

Select All
 Send Text Message
(requires mobile phone number)
 Send Email

Cancel **Delete** **Save**



The screenshot shows a Gmail inbox with 544 unread messages. A specific email from "myDevices Cayenne" is highlighted with a red oval, containing the subject "Trigger Notification - Hi Sarnaam Musthafa, Your Sarn#1 is ...". Below it, another message from "Sarnaam Mustafa" is partially visible with the subject "Lab Copy". The inbox also contains messages from various sources like Instructables, Superset, and GoAir, with subjects ranging from "Install Complete" to "Low Fare Alert!". The interface includes standard Gmail controls like compose, search, and filter buttons.

From	Subject	Date
myDevices Cayenne	Trigger Notification - Hi Sarnaam Musthafa, Your Sarn#1 is ...	9:30 PM
Sarnaam Mustafa	Lab Copy	9:00 PM
myDevices	myDevices Cayenne - Install Complete - Hi Sarnaam Mustha...	8:02 PM
Instructables	FindyBot3000: Voice Controlled Organizer - Pallet Wood Cl...	Apr 4
Superset	New Notice - Volante Further Online Test - 04.04.19 - tnpstu...	Apr 4
SRM Communications	Session on "Academic Internship @ Singapore" - June 2019...	Apr 4
Superset	New Notice - Volante Online Test 2nd Batch List - tnpsuite ...	Apr 4
Superset	New Notice - Volante Online Test 1st Batch List - tnpsuite P...	Apr 4
6EGSTInvoice	TaxInvoice - KL1181903AW28501 - Hi, Please find attached ...	Apr 3
GoAir	Low Fare Alert! - Dear Smart Flyers, take a trip to your favou...	Apr 2
Instructables	Word Clock Controlled by 114 Servos - Kitchen Remodel on ...	Apr 2
Superset	New Notice - Campus Recruitment by Volante (2020 Batch)...	Apr 2
Coursera	Recommended: Machine Learning - Ready to learn somethi...	Apr 2
Superset	New Notice - FINAL REMINDER: RELIANCE INDUSTRIES PR...	Apr 1

Result :

Fall detection system using ESP8266 NodeMCU, MPU6050 for gyroscope and accelerometer module has been successfully developed and implemented.