IOT BASED FALL DETECTION AND SMART TRACKING FOR SENIOR CITIZEN

A PROJECT REPORT

SUBMITTED BY

NITHISHA.K

In partial fulfillment of the award of the degree

Of

BACHELOR OF TECHNOLOGY

IN

INFORMATION TECHNOLOGY

PANIMALAR INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY : CHENNAI 600 025

NOVEMBER 2023

CHAPTER 1

ABSTRACT

Falls are a significant health concern, particularly for the elderly, and can lead to severe injuries or even fatalities. In response to this issue, an IoT-based fall detection system and smart tracking solution using Arduino Uno is presented. This system combines sensor technology, real-time data processing, and connectivity to provide rapid assistance to individuals in need.

The core of the system is an Arduino Uno microcontroller, equipped with various sensors, including accelerometers and gyroscopes. These sensors continuously monitor the user's movements and orientation. When a sudden and abnormal change in orientation is detected, indicative of a fall, the system triggers an alert.

The system is connected to an IoT platform, which facilitates communication with caregivers and healthcare providers. It offers real-time tracking of the individual's location and immediate notifications in case of a fall incident. This connectivity ensures that help can be summoned promptly, potentially reducing the severity of injuries and improving outcomes in emergency situations.

To enhance the system's accuracy, machine learning algorithms are implemented to distinguish between genuine falls and other everyday activities, thus reducing false alarms. This approach is crucial for ensuring the system's reliability in practical use.

Moreover, this system offers customization options for users, allowing them to set parameters for safe zones and activity schedules. This feature enables senior citizens to maintain a sense of independence while providing their caregivers with peace of mind.The IoT-based fall detection and smart tracking system using Arduino Uno leverages affordable and accessible hardware, making it a practical solution for improving the safety and well-being of vulnerable populations. This technology has the potential to revolutionize the way we care for our elderly and those at risk, enhancing their quality of life and providing a safer and more secure living environment.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 2

INTRODUCTION

2.1 OVERVIEW

The Internet of Things (IoT) has catalyzed a profound transformation in the way we interact with the world and, importantly, how we care for the well-being of vulnerable populations. One of the most critical areas of concern is the safety and security of the elderly and individuals at risk of falls. Falls represent a significant health challenge, with potentially devastating consequences, including injuries, reduced quality of life, and increased healthcare costs. In response to this pressing issue, the IoT-based Fall Detection System and Smart Tracking project, harnessing the capabilities of the Arduino Uno, MPU6050 accelerometer and gyroscope sensor, and NEO-6M GPS module, emerges as a powerful and comprehensive solution.

MPU6050:

The MPU6050 is a six-axis inertial measurement unit (IMU) that can be used to measure acceleration, gyroscope, and magnetometer data. This data can be used to detect falls by analyzing the changes in the person's movement.

NEO-6M:

The NEO-6M is a GPS module that can be used to track the location of the fallen person. This information can be used to send an alert to caregivers and to help them to find the fallen person.

SYSTEM:

The IOT based fall detection system with smart tracking using MPU6050 and NEO-6M is a low-cost and non-wearable alternative that can be used to detect falls and track the location of the fallen person. The system consists of the following components:

- A microcontroller board
- An MPU6050 IMU
- A NEO-6M GPS module
- A transmitter
- A receiver

The microcontroller board is used to process the data from the MPU6050 and NEO-6M modules. The MPU6050 data is used to detect falls, and the NEO-6M data is used to track the location of the fallen person. The transmitter is used to send the fall detection and location data to the receiver. The receiver is used to receive the fall detection and location data from the transmitter and to send an alert to caregivers.

BENEFITS

The IOT based fall detection system with smart tracking using MPU6050 and NEO-6M has the following benefits:

- It is low-cost.
- It is non-wearable.
- It can detect falls and track the location of the fallen person.
- It can send an alert to caregivers.

As the global population ages, the need for innovative and practical healthcare solutions becomes increasingly evident. Falls are a leading cause of injury and fatality among the elderly, and the ability to respond swiftly and effectively to fall incidents is of paramount importance. This project presents an integrated approach that seamlessly combines hardware and software components to address the challenges of fall detection and real-time tracking.

2.2 PROBLEM DEFINITION

The IoT-based Fall Detection System and Smart Tracking project is an innovative solution that combines hardware and software technologies to address the pressing issue of fall detection and real-time tracking for the safety and well-being of individuals, particularly senior citizens and those at risk of falls. This project utilizes the Arduino Uno microcontroller, the MPU6050 accelerometer and gyroscope sensor, and the NEO-6M GPS module to create a comprehensive and highly effective system.The project's core component, the Arduino Uno microcontroller, serves as the central processing unit. It is responsible for sensor data collection, fall detection, and communication with external systems via IoT. The MPU6050 is an integrated accelerometer and gyroscope sensor that continuously monitors the user's movements and orientation. This sensor is instrumental in detecting falls as it can sense sudden changes in acceleration and orientation, providing essential data for fall detection.The NEO-6M GPS module is integrated into the system to provide accurate real-time location information. This module uses satellite signals to determine the user's exact geographic coordinates, which is vital for tracking their location and ensuring quick assistance in emergency situations.

The project's fall detection algorithm is built upon data from the MPU6050 sensor. When a sudden and significant change in acceleration and orientation is detected, the system interprets this as a potential fall incident. False positives are minimized by implementing sophisticated data analysis techniques and machine learning algorithms, ensuring that only genuine falls trigger alerts. The NEO-6M GPS module plays a crucial role in tracking the user's location in real time. By continuously receiving GPS signals and calculating the precise geographic coordinates, the system enables caregivers, family members, and healthcare professionals to track the user's movement through a user-friendly interface. This real-time tracking functionality ensures a rapid response to fall incidents and allows for location-based assistance.

The project leverages Internet of Things (IoT) technology to enable seamless communication between the device and external stakeholders. The Arduino Uno is connected to an IoT platform that facilitates instant notifications and remote monitoring. Caregivers and healthcare providers can access a web portal or a mobile application to receive alerts and monitor the user's location, ensuring immediate assistance when needed.To respect the user's autonomy and preferences, the system allows for customization. Users can set parameters for safe zones and activity schedules, providing a sense of independence while maintaining their safety.

2.3 SCOPE

The IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050 accelerometer and gyroscope sensor, and NEO-6M GPS module addresses a critical and multifaceted problem concerning the safety and well-being of vulnerable populations, particularly the elderly and individuals at risk of falls. The key challenges and problems that this project aims to resolve are as follows:

Falls are a pervasive health issue, especially among the elderly.According to global statistics, falls are a leading cause of injury, hospitalization, and even fatalities in this demographic. Such incidents can result in fractures, head injuries, and a reduced quality of life. The project aims to tackle this issue by promptly identifying falls and summoning timely assistance.

Traditional fall detection systems are often characterized by high rates of false alarms or limited accuracy in distinguishing genuine falls from other everyday activities. This lack of reliability can lead to a sense of insecurity among users and caregivers. This project seeks to develop a more accurate and efficient fall detection solution.

In the event of a fall, the response time is crucial. Delayed assistance can exacerbate the severity of injuries or complications arising from falls. Existing systems often do not provide real-time tracking, making it challenging to locate the individual quickly. The project's goal is to ensure rapid assistance by integrating real-time tracking capabilities.

Another challenge is balancing the need for safety with the desire for user independence. Many elderly individuals wish to retain a sense of autonomy and freedom while ensuring their safety. Striking this balance is a complex issue that the project seeks to address through customizable parameters for safe zones and activity schedules.

Accurate tracking of the user's location is essential in emergency situations, but it remains a challenge in many existing solutions. Providing precise geographical coordinates, latitude, longitude, and altitude data is a critical component of this project.Accurately distinguishing between genuine falls and other activities is a complex task that can be improved with the integration of machine learning algorithms. This project seeks to leverage these algorithms to reduce false alarms and enhance the overall accuracy of fall detection.

CHAPTER 2

THEORETICAL CONCEPT

2.1 BASICS OF INTERNET OF THINGS (IoT)

IoT (Internet of Things) is an advanced automation and analytics system which exploits networking, sensing, big data, and artificial intelligence technology to deliver complete systems for a product or service. These systems allow greater transparency, control, and performance when applied to any industry or system.IoT systems have applications across industries through their unique flexibility and ability to be suitable in any environment. They enhance data collection, automation, operations, and much more through smart devices and powerful enabling technology.IoT systems allow users to achieve deeper automation, analysis, and integration within a system. They improve the reach of these areas and their accuracy. IoT utilizes existing and emerging technology for sensing, networking, and robotics. IoT exploits recent advances in software, falling hardware prices, and modern attitudes towards technology. Its new and advanced elements bring major changes in the delivery of products, goods, and services; and the social, economic, and political impact of those changes.

2.1.1 IOT – KEY FEATURES

The most important features of IoT include artificial intelligence, connectivity, sensors, active engagement, and small device use. A brief review of these features is given below:

- AI – IoT essentially makes virtually anything “smart”, meaning it enhances every aspect of life with the power of data collection, artificial intelligence algorithms, and networks. This can mean something as simple as enhancing your refrigerator and cabinets to detect when milk and your favourite cereal run low, and to then place an order with your preferred grocer.
- Connectivity – New enabling technologies for networking, and specifically IoT networking, mean networks are no longer exclusively tied to major providers. Networks can exist on a much smaller and cheaper scale while still being practical. IoT creates these small networks between its system devices.
- Sensors – IoT loses its distinction without sensors. They act as defining instruments which transform IoT from a standard passive network of devices into an active system capable of real-world integration.
- Active Engagement – Much of today's interaction with connected technology happens through passive engagement. IoT introduces a new paradigm for active content, product, or service engagement.
- Small Devices – Devices, as predicted, have become smaller, cheaper, and more powerful over time. IoT exploits purpose-built small devices to deliver its precision, scalability, and versatility.

IoT software addresses its key areas of networking and action through platforms, embedded systems, partner systems, and middleware. These individual and master applications are responsible for data collection, device integration, real-time analytics, and application and process extension within the IoT network. They exploit integration with critical business systems (e.g., ordering systems, robotics, scheduling, and more) in the execution of related tasks.

2.1.2 DATA COLLECTION

This software manages sensing, measurements, light data filtering, light data security, and aggregation of data. It uses certain protocols to aid sensors in connecting with real-time, machine-to- machine networks. Then it collects data from multiple devices and distributes it in accordance with settings. It also works in reverse by distributing data over devices. The system eventually transmits all collected data to a central server.

2.1.3 DEVICE INTEGRATION

Software supporting integration binds (dependent relationships) all system devices to create the body of the IoT system. It ensures the necessary cooperation and stable networking between devices. These applications are the defining software technology of the IoT network because without them, it is not an IoT system. They manage the various applications, protocols, and limitations of each device to allow communication.

2.1.4 REAL TIME ANALYSIS

These applications take data or input from various devices and convert it into viable actions or clear patterns for human analysis. They analyse information based on various settings and designs in order to perform automation-related tasks or provide the data required by industry.

2.1.5 APPLICATION AND PROCESS EXTENSION

These applications extend the reach of existing systems and software to allow a wider, more effective system. They integrate predefined devices for specific purposes such as allowing certain mobile devices or engineering instruments access. It supports improved productivity and more accurate data collection.

IoT primarily exploits standard protocols and networking technologies. However, the major enabling technologies and protocols of IoT are RFID, NFC, low-energy Bluetooth, low-energy wireless, low- energy radio protocols, LTE-A, and WiFi-Direct. These technologies support the specific networking functionality needed in an IoT system in contrast to a standard uniform network of common systems.

2.1.6 WIFI-DIRECT

WiFi-Direct eliminates the need for an access point. It allows P2P (peer-to-peer) connections with the speed of WiFi, but with lower latency. WiFi-Direct eliminate an element of a network that often bogs it down, and it does not compromise on speed or throughput.

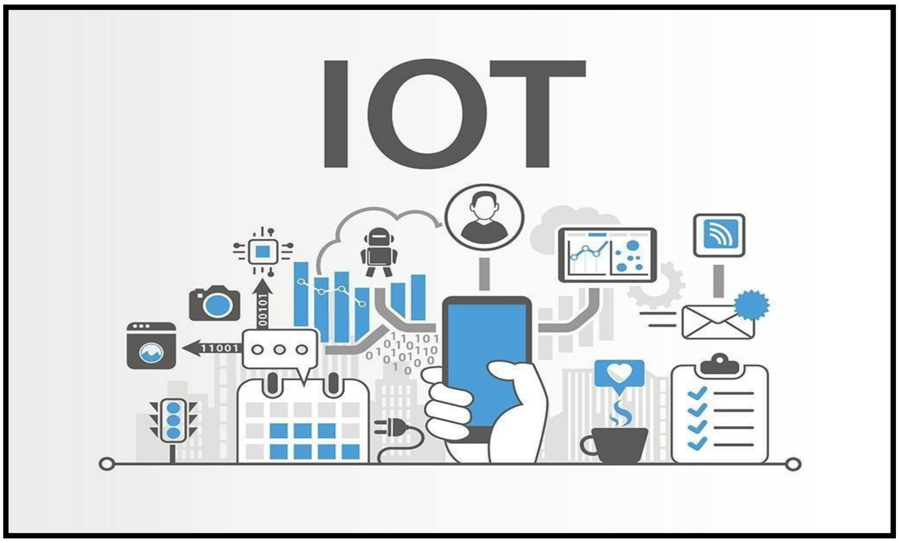


Figure 1: Internet of Things (IoT)

2.2 FALL RISK FACTOR

A person can be more or less prone to fall, depending on a number of risk factors and hence a classification based on only age as a parameter is not enough. In fact, medical studies have determined a set of so called risk factors:

Intrinsic

1. Age (over 75)
2. Chronic disease
3. Previous falls
4. Poor balance
5. Low mobility and bone fragility
6. Sight problems
7. Cognitive and dementia problems
8. Parkinson disease
9. Use of drugs that affect the mind
10. Incorrect lifestyle (inactivity, use of alcohol, obesity)

Internal Environment

1. Need to reach high objects
2. Slipping floors
3. Stairs
4. Incorrect use of shoes and clothes

External Environment

1. Damaged roads
2. Dangerous steps
3. Poor lighting
4. Crowded places.

2.3 <u>FALL DETECTION ALGORITHM</u>

The total sum acceleration vector (Acc) contain both static and dynamic acceleration components, is calculated from sampled data as indicated in equation (1) & (2)

$$\mathbf{Acc}=\surd(\square\square)^{2}+(\square\square)^{2}+(\square\square)^{2}\text{-------------------------}\mathbf{(1)}$$

Where Ax, Ay, Az 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)

$$\mathbf{w}=\surd(\square\square)^{2}+(\square\square)^{2}+(\square\square)^{2}\text{------------------}\mathbf{(2)}$$

Where Wx, Wy, Wz the acceleration in the x, y, z axes, respectively.When stationary, the acceleration magnitude, Acc, from tri-axial accelerometer is constant, and angular velocity is 0 m/s. When the subject falls, the acceleration is rapidly changing and the angular velocity produces a variety of signals along fall direction.

Since the Fall Index (Acc) requires high sampling frequency and fast acceleration changes, it will miss falls that happen slowly. Hence, Acc is not used unless we want to compare the performances of our systems with previous studies that have used the same positions but with deferent speed and accelerations. The lower and upper fall thresholds for the acceleration and angular velocity used to identify the fall are derived as follows:

<u>LOWER FALL THRESHOLD (LFT):</u>

The negative peaks for the resultant of each recorded activity are referred to as the signal lower peak values (LPVs). The LFT for the acceleration signals are set at the level of the smallest magnitude lower fall peak (LFP) recorded.

<u>UPPER FALL THRESHOLD (UFT):</u>

The positive peaks for the recorded signals for each recorded activity are referred to as the signal upper peak values (UPVs). The UFTfor each of the acceleration and the angular velocity signals were setat the level of the smallest magnitude UPV recorded. The UFT is related to the peak impact force experienced by the body segment during the impact phase of the fall.Fall detection algorithms using thresholds are normally divided into two groups, one is based on the LFT comparison and the other is based on UFT comparison of acceleration data. Although past research has achieved some significant results, the accuracy is still Below desired levels. In this study adjust the UFT and LFT and found the performance to be 83.33 % and 67.08 %, respectively.

2.4 PROJECT OVERVIEW

The IoT-based Fall Detection System and Smart Tracking project aims to enhance the safety and well-being of individuals, especially the elderly or those with mobility issues. The project leverages the capabilities of the Arduino Uno microcontroller, MPU6050 accelerometer and gyroscope sensor, and NEO-6M GPS module to detect falls and track the user's location in real-time.

Objectives

1. Develop a reliable Fall Detection System using an MPU6050 sensor.
2. Implement real-time tracking of the user's location using the NEO-6M GPS module.
3. Integrate IoT capabilities to enable remote monitoring and alerts.
4. Enhance user safety by providing timely notifications in case of a fall.
5. Create a user-friendly interface for monitoring and tracking through a web or mobile application.

Components Used

1. Arduino Uno: The central microcontroller responsible for processing data and controlling the system.

2. MPU6050 Sensor: An accelerometer and gyroscope combination sensor to detect changes in acceleration and orientation, crucial for fall detection.

3. NEO-6M GPS Module: A compact and accurate GPS module for real-time location tracking.

System Workflow

1. Fall Detection:

- The MPU6050 sensor continuously monitors the user's movements and orientation.
- Algorithmic analysis of sensor data to identify patterns indicative of a fall.
- Trigger an alert if a fall is detected, signaling a potential emergency.

2. Location Tracking:

- Utilize the NEO-6M GPS module to acquire the user's precise location.
- Periodically update the location data to ensure real-time tracking.
- Transmit the location data to a central server for remote monitoring.

3. IoT Integration:

- Establish a secure and reliable communication link between the Arduino Uno and the IoT platform.
- Use MQTT or other protocols for data transmission to a cloud server.
- Implement a cloud-based dashboard for remote monitoring and control.

4. Alert System:

- Configure the system to send alerts via SMS, email, or push notifications to predefined contacts in case of a fall.
- Include additional features like automatic call alerts to emergency services.

5. User Interface:

- Develop a user-friendly interface accessible through a web or mobile application.
- Display real-time location, fall detection status, and historical data.
- Provide options for configuring alert preferences and emergency contacts.

Benefits

1. Improved Safety: Timely detection of falls and immediate alerts enhance the safety of individuals, especially those at risk.
2. Real-time Tracking: Accurate GPS tracking ensures quick response in emergency situations, facilitating timely assistance.
3. User-Friendly Interface: The project aims to provide an intuitive and accessible interface for users and caregivers.
4. Customizable Alerts: Users can configure the system to send alerts in the preferred format to chosen contacts, tailoring the system to individual needs.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

An IoT-based fall detection system is a comprehensive solution that utilizes the power of the Internet of Things (IoT) to identify and alert caregivers when an individual has fallen. These systems typically consist of wearable sensors, data processing units, and communication modules that work together to provide real-time monitoring and fall detection capabilities.

Components of an IoT-based fall detection system:

a. Sensors: These sensors are worn by the user and collect data about their movements, such as acceleration, gyroscope, and magnetometer data. The sensors can be embedded in a wristwatch, pendant, or other wearable device.

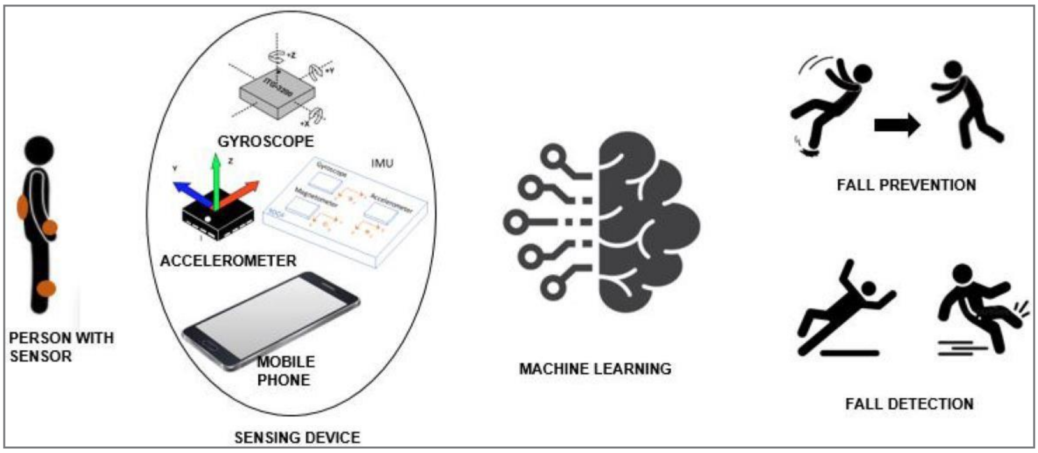


Figure 2 : Sensors

b. Data processing and analysis: The sensor data is transmitted to a central unit, such as a smartphone or cloud server, for processing and analysis. This involves using machine learning algorithms to identify patterns in the data that are indicative of a fall.

c. Alerting: If a fall is detected, the system will send an alert to a designated caregiver or emergency responder. The alert can be sent via SMS, email, or a smartphone app.

Here is a more detailed explanation of each component:

1. Sensors: (MPU6050)

Accelerometer: Measures acceleration in three axes (x, y, and z). This can be used to detect changes in the user's posture and movement.

Gyroscope: Measures rotational motion in three axes. This can be used to detect falls that involve a sudden change in orientation, such as tripping or slipping.

Magnetometer: Measures the Earth's magnetic field. This can be used to determine the user's orientation and to help distinguish between falls and other activities, such as bending over or lying down.

2. Data processing and analysis:

Machine learning algorithms: These algorithms analyze the sensor data to identify patterns that are indicative of a fall. Some common algorithms used for fall detection include decision trees, support vector machines, and neural networks.

Fall detection criteria: The system must be able to distinguish between falls and other activities, such as walking, running, and bending over. This can be done by setting thresholds for the sensor data or by using more sophisticated machine learning algorithms.

3. Alerting:

Alert methods: The system can send alerts via SMS, email, or a smartphone app. The system can also be integrated with a home automation system to trigger alarms or lights.

Caregiver notification: The system can notify a designated caregiver or emergency responder when a fall is detected. The caregiver can then check on the user and provide assistance if needed.

Benefits of IoT-based fall detection systems:

a. Improved safety: IoT-based fall detection systems can help to improve the safety of elderly and at-risk individuals by alerting caregivers to falls promptly.

b. Reduced healthcare costs: Falls can lead to serious injuries, such as fractures and head traumas. By preventing falls, IoT-based fall detection systems can help to reduce healthcare costs.

c. Increased independence: IoT-based fall detection systems can help elderly and at-risk individuals to live more independently by providing them with a sense of security.

Overall, IoT-based fall detection systems are a promising technology that has the potential to improve the safety and quality of life for elderly and at-risk individuals.

3.2 PROPOSED SYSTEM

The proposed system is an IoT-based fall detection and smart tracking system using Arduino Uno, MPU6050, and NEO-6M module. The system is designed to detect falls in real time and send an alert to a caregiver or emergency contact. It can also track the user's location and send updates to the caregiver.

The system consists of the following components:

a. Arduino Uno: A microcontroller board that is used to control all of the other components in the system.
b. MPU6050: A motion sensor that is used to detect falls.
c. NEO-6M module: A GPS module that is used to track the user's location.
d. GSM module: A cellular module that is used to send alerts and location updates.
e. Battery: A power source for the system.

The system works as follows:

1. The MPU6050 sensor is used to detect falls. The sensor constantly monitors the user's movements and detects if there is a sudden change in acceleration. If the sensor detects a fall, it sends a signal to the Arduino Uno.
2. The Arduino Uno then sends an alert to the caregiver or emergency contact. The alert can be sent via SMS, email, or phone call.
3. The NEO-6M GPS module is used to track the user's location. The module constantly monitors the user's position and sends updates to the Arduino Uno.
4. The Arduino Uno then sends the location updates to the caregiver. This way, the caregiver can always know where the user is located.
5. The proposed system can be used to improve the safety and well-being of people who are at risk of falls, such as the elderly and people with disabilities. It can also be used to track the location of people who are lost or injured.

Here are some additional benefits of the proposed system:

- It is easy to install and use.
- It is affordable.
- It is reliable and accurate.
- It is scalable.

The proposed system is still under development, but it has the potential to be a valuable tool for improving the safety and well-being of people who are at risk of falls.

CHAPTER 4

SYSTEM SPECIFICATION

4.1 SOFTWARE SPECIFICATION

SOFTWARE PLATFORM (ARDUINO IDE)

The Arduino Integrated Development Environment (IDE) is a valuable tool for developing IoT (Internet of Things) projects. It simplifies the process of programming and interfacing with Arduino microcontroller boards, making it easier to create IoT applications. Here's a detailed description of how the Arduino IDE is used in IoT projects:

1. Programming Microcontrollers: The Arduino IDE is the primary environment for writing code (sketches) that runs on Arduino microcontroller boards. In IoT projects, the IDE allows you to program the microcontroller to collect data from sensors, communicate with other devices, and control actuators.

2. Sensor and Actuator Integration: IoT projects often involve a wide range of sensors (e.g., temperature, humidity, motion, light) and actuators (e.g., motors, servos, relays). The IDE simplifies the integration of these components by providing libraries and example code for various sensors and actuators.

3. Library Support: The IDE includes a library manager that allows you to easily add and manage libraries for sensors, communication modules (e.g., Wi-Fi, Bluetooth, LoRa), and other hardware components. These libraries provide pre-written code to interface with specific devices, saving you time and effort in development.

4. Custom IoT Hardware: In many IoT projects, you might create custom hardware solutions by adding or modifying sensors, actuators, and communication modules. The Arduino IDE simplifies the process of developing and testing custom hardware by providing a framework for uploading code to microcontroller boards.

5. Communication: IoT often requires communication between devices. Arduino boards can communicate via wired (e.g., serial, I2C, SPI) and wireless (e.g., Wi-Fi, Bluetooth) protocols. The IDE provides a serial monitor for debugging and monitoring data exchanges between devices.

6. Data Processing and Analysis: IoT applications frequently involve data processing and analysis. The Arduino IDE allows you to write code to process sensor data, apply algorithms for filtering and analysis, and send processed data to cloud platforms or display it on connected screens.

7. Connectivity to Cloud Services: Many IoT projects involve sending data to cloud services for storage, analysis, and remote access. The Arduino IDE can be used to write code that interfaces with cloud platforms and APIs, making it possible to transmit data securely to the cloud.

8. User Interface Development: In IoT projects with a user interface (e.g., a web page or a mobile app), the Arduino IDE can be used to program the microcontroller to respond to user inputs and commands. For example, you can control IoT devices via a web interface or a mobile app.

9. Firmware Updates: IoT devices often require firmware updates to fix bugs or add new features. The Arduino IDE simplifies the process of developing and uploading firmware updates to connected devices.

10. Prototyping and Iteration: IoT projects typically start with a prototype. The IDE, along with breadboards, allows for rapid prototyping and iteration. You can quickly test ideas, refine designs, and adapt your IoT project as it evolves.

11. Community and Resources: The Arduino community provides extensive support, including forums, documentation, and a wealth of tutorials. This community is a valuable resource for IoT developers using the Arduino IDE.

12. Customization and Extensibility: The Arduino IDE can be extended with plugins and custom libraries to add specific features or to accommodate specialized IoT requirements.

4.2 HARDWARE SPECIFICATION

1. ARDUINO UNO

Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button.Here are its technical specifications and general pin functions:

Technical Specifications:

1. Microcontroller: ATmega328P.
2. Operating Voltage: 5 volts.
3. Input Voltage: 7-12 volts (recommended) via an external power supply or USB connection.
4. Digital I/O Pins: 14 (of which 6 provide PWM output).
5. Analog Input Pins: 6.
6. DC Current per I/O Pin: 20 mA.
7. DC Current for 3.3V Pin: 50 mA.
8. Flash Memory: 32 KB (of which 0.5 KB is used by the bootloader).
9. SRAM: 2 KB.
10. EEPROM: 1 KB.
11. Clock Speed: 16 MHz.
12. USB Interface: ATmega16U2 (or 8U2 on earlier versions) for serial communication with the computer.

General Pin Functions:

1. Digital Pins (D2-D13): These pins can be used for digital input or output. Some of them also support pulse-width modulation (PWM) for analog-like output control.

2. Analog Pins (A0-A5): These pins can be used to read analog voltage levels from sensors or other analog devices. They can also function as digital I/O pins.

3. Power Pins:

- 5V: Supplies 5 volts to external components.
- 3.3V: Provides a 3.3-volt output regulated from the 5V input.
- GND: Ground pins for connecting to the common ground.

4. Reset (RST): This pin allows you to reset the microcontroller. You can do this manually or through software.

5. Serial Communication:

TX (Transmit) and RX (Receive) pins: These pins are used for serial communication with other devices or for programming the board.

6. I2C Communication: A4 (SDA) and A5 (SCL) pins are used for I2C communication, which allows you to connect to I2C sensors and devices.

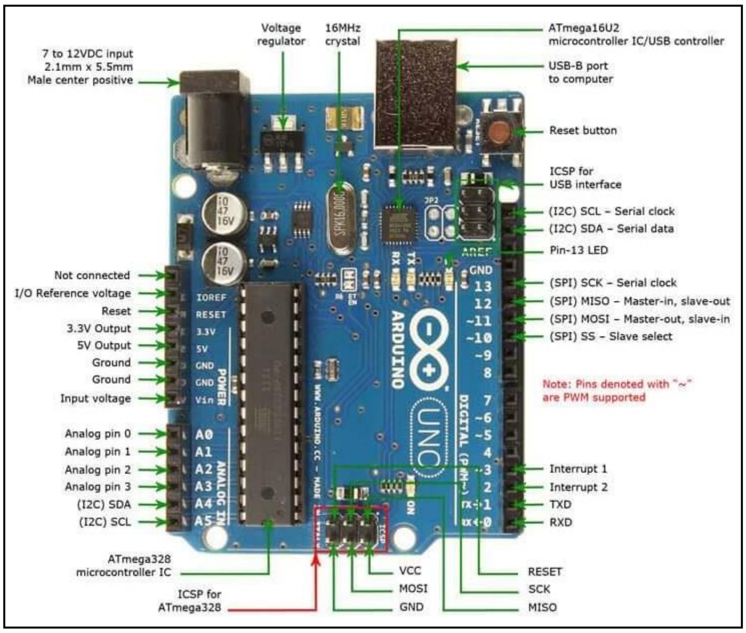
7. SPI Communication: On the ICSP header, there are pins for SPI communication (MISO, MOSI, SCK) and a slave-select (SS) pin.

8. PWM Pins: D3, D5, D6, D9, D10, and D11 can generate PWM signals.

9. LED Indicator: The board has a built-in LED connected to pin D13, which is often used for testing and debugging.

10. AREF: Analog Reference. You can connect an external voltage reference to this pin for more accurate analog readings.

Figure 3 : Arduino Uno Pinout Diagram



These are the basic specifications and pin functions of the Arduino Uno. It's a versatile board suitable for a wide range of projects, from simple blinking LED experiments to complex embedded systems and IoT applications.

2. MPU6050

MPU6050 is a three-axis accelerometer and three-axis gyroscope Micro Electro-mechanical system (MEMS). It aids in the measurement of velocity, orientation, acceleration, displacement, and other motion-related features. It is often used in various applications, including robotics, drones, and motion-sensing devices. Here are its technical specifications and general pin functions:

Technical Specifications:

1. Accelerometer Range: ±2g, ±4g, ±8g, or ±16g (user-selectable).
2. Gyroscope Range: ±250°/s, ±500°/s, ±1000°/s, or ±2000°/s (user-selectable).
3. Communication Interface: I2C (Inter-Integrated Circuit).
4. Supply Voltage: Typically operates at 3.3V but can tolerate up to 5V.
5. Power Consumption: Low power consumption, making it suitable for battery-operated devices.
6. Dimensions: Typically comes in a small surface-mount package, but breakout boards with standard pin headers are available for easy interfacing.

General Pin Functions:

1. VCC (Voltage Supply): This pin is used to provide power to the MPU-6050. It typically operates at 3.3V, but it can tolerate up to 5V. Be sure to connect it to a stable power source.

2. GND (Ground): This is the ground connection for the module, providing the reference voltage for all other signals.

3. SDA (Serial Data): The SDA pin is used for bidirectional serial data transfer in the I2C communication protocol. Connect it to the corresponding SDA pin on your microcontroller.

4. SCL (Serial Clock): The SCL pin is used to synchronize data transfer in the I2C protocol. Connect it to the corresponding SCL pin on your microcontroller.

5. X-Axis Accelerometer Output (AX): This pin provides the analog voltage output corresponding to the acceleration in the X-axis. It can be connected to an analog input pin on a microcontroller.

6. Y-Axis Accelerometer Output (AY): Similar to AX, this pin provides the analog voltage output for the Y-axis acceleration.

7. Z-Axis Accelerometer Output (AZ): Similar to AX and AY, this pin provides the analog voltage output for the Z-axis acceleration.

8. X-Axis Gyroscope Output (GX): This pin provides the analog voltage output corresponding to the angular rate in the X-axis. It can be connected to an analog input pin on a microcontroller.

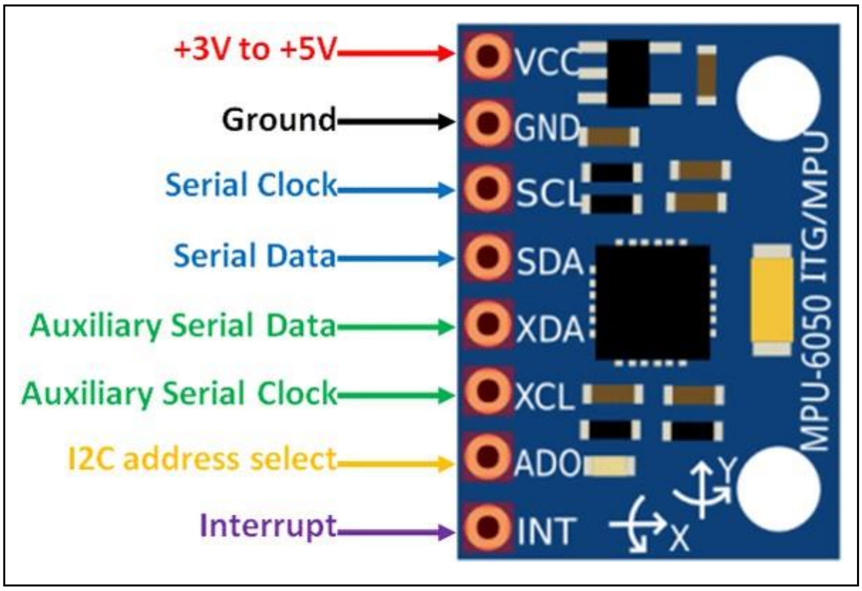
9. Y-Axis Gyroscope Output (GY): Similar to GX, this pin provides the analog voltage output for the Y-axis angular rate.

10. Z-Axis Gyroscope Output (GZ): Similar to GX and GY, this pin provides the analog voltage output for the Z-axis angular rate.

11. AD0 (Address Select): The AD0 pin can be used to change the I2C address of the MPU-6050 by pulling it to either VCC or GND. This feature is useful when you want to connect multiple MPU-6050 modules on the same I2C bus.

12. INT (Interrupt): The INT pin is used to provide an interrupt signal to your microcontroller when specific conditions are met. It can be configured to trigger interrupts for motion detection, free-fall, and other events.

Figure 4 : MPU6050 Pinout Diagram



These are the basic technical specifications and pin functions of the MPU-6050 sensor module. When using the MPU-6050, you'll need to connect it to a microcontroller and communicate with it via the I2C protocol to read the sensor data, allowing you to measure motion and orientation.

3. NEO-6M

The NEO-6M is a popular GPS (Global Positioning System) module that provides accurate location and time information. It is commonly used in various applications, including navigation, tracking, and geolocation-based projects. Here are its technical specifications and general pin functions:

Technical Specifications:

1. GPS Receiver: u-blox NEO-6M.

2. GPS Chipset: u-blox 6, including high-performance GPS and EGNOS/WAAS/MSAS-capable satellite-based augmentation systems.

3. Frequency: The NEO-6M operates in the L1 frequency band (1575.42 MHz).

4. Sensitivity: The module has excellent sensitivity, allowing it to acquire and track satellites even in challenging environments.

5. Position Accuracy: Typically, it can provide location accuracy within a few meters.

6. Time Accuracy: The NEO-6M provides highly accurate time information, synchronized to GPS satellites.

7. Communication Interface: UART (Serial Communication) is the standard interface for data communication with the module.

8. Supply Voltage: Typically operates at 3.3V or 5V, depending on the specific module and its voltage regulator configuration.

9. Current Consumption: Low power consumption, making it suitable for battery-powered applications.

General Pin Functions:

1. VCC (Voltage Supply): This pin is used to provide power to the NEO-6M module. The voltage requirement depends on the specific module version and its voltage regulator configuration (3.3V or 5V). Be sure to connect it to a stable power source.

2. GND (Ground): This is the ground connection for the module, providing the reference voltage for all other signals.

3. TX (Transmit): The TX pin is used to transmit data from the module to your microcontroller or other devices. It is an output from the module and should be connected to the RX (Receive) pin of your microcontroller.

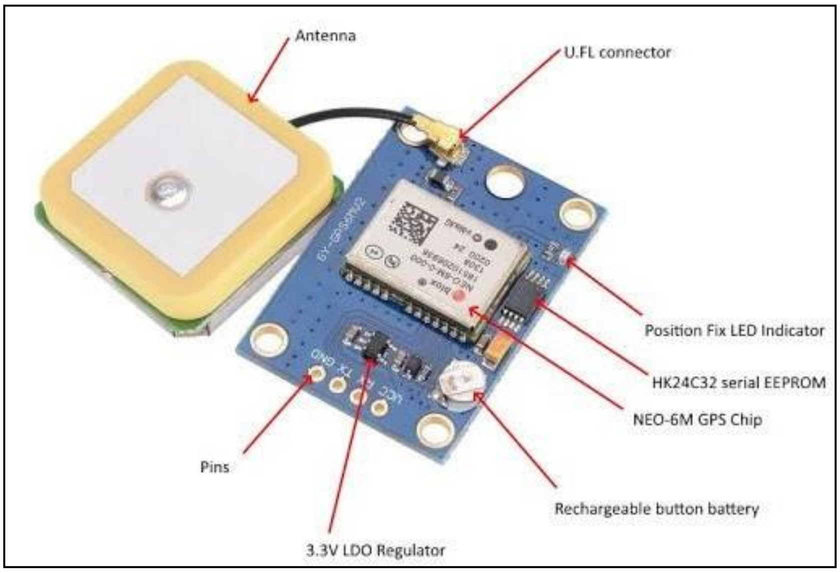
4. RX (Receive): The RX pin is used to receive data from your microcontroller or other devices. It is an input to the module and should be connected to the TX (Transmit) pin of your microcontroller.

5. PPS (Pulse-Per-Second): Some NEO-6M modules have a PPS output, which provides a pulse signal once per second, synchronized with the GPS time. This signal can be used for precise timekeeping and synchronization.

6. Antenna: The module typically requires an external GPS antenna to receive signals from GPS satellites. Connect the antenna to the designated antenna connector on the module.

7. Enable (EN) or Power Control: Some modules have an enable pin that can be used to turn the module on or off. Connecting this pin to VCC or GND controls the module's power state.

Figure 5 : NEO-6M Pinout Diagram



The NEO-6M module is relatively straightforward to use. You provide power, connect the UART interface to your microcontroller or computer, and then parse the NMEA (National Marine Electronics Association) sentences received from the module to extract location and time data. The module acquires signals from GPS satellites, calculates its position, and provides that information to your application through the UART interface.

4. BREADBOARD

Breadboards play a crucial role in IoT (Internet of Things) development by serving as a prototyping and testing platform for creating and experimenting with electronic circuits and IoT devices. Here's how breadboards are essential in IoT development:

1. Rapid Prototyping: IoT projects often involve a combination of sensors, microcontrollers, communication modules, and actuators. Breadboards allow developers to quickly prototype these complex circuits without the need for soldering. This is particularly valuable in the early stages of IoT project development, where rapid iteration and experimentation are common.

2. Component Integration: Breadboards are used to integrate various electronic components, such as sensors (temperature, humidity, motion, light, etc.), microcontrollers (Arduino, Raspberry Pi, ESP8266, etc.), communication modules (Wi-Fi, Bluetooth, LoRa, etc.), and actuators (motors, relays, LEDs, etc.) into a single circuit. This integration is vital for creating IoT devices that can sense and act upon their environment.

3. Testing and Debugging: Breadboards provide an environment for testing and debugging IoT prototypes. You can easily observe the behavior of different components and connections, making it easier to identify and resolve issues in the early stages of development.

4. Flexibility: IoT projects often evolve, and requirements may change. Breadboards offer the flexibility to modify and rearrange components and connections as needed, which is especially valuable when experimenting with various sensors or communication modules to find the most suitable combination.

5. Learning and Education: Breadboards are commonly used in educational settings to teach students about electronics, IoT, and prototyping. They help individuals learn the basics of circuit design and develop practical skills for IoT development.

6. Proof of Concept: Before investing time and resources in custom PCB (Printed Circuit Board) design or manufacturing, developers use breadboards to create proof-of-concept prototypes. This enables them to demonstrate the viability and functionality of an IoT device or system.

7. Cost-Effective Development: Breadboards are cost-effective, as they can be reused for various projects. This makes them an ideal choice for developers and hobbyists who may not have the budget for custom PCBs or specialized prototyping tools.

8. Incorporating External Sensors: IoT often involves interfacing with external sensors and devices. Breadboards facilitate the connection of these sensors, allowing for experimentation and integration before final deployment.

9. IoT Prototyping Platforms: Several microcontroller platforms commonly used in IoT development, such as Arduino and Raspberry Pi, offer compatible expansion boards or "shields" that can be directly connected to breadboards. This simplifies the process of creating IoT prototypes by providing pre-made connectors and interfaces.

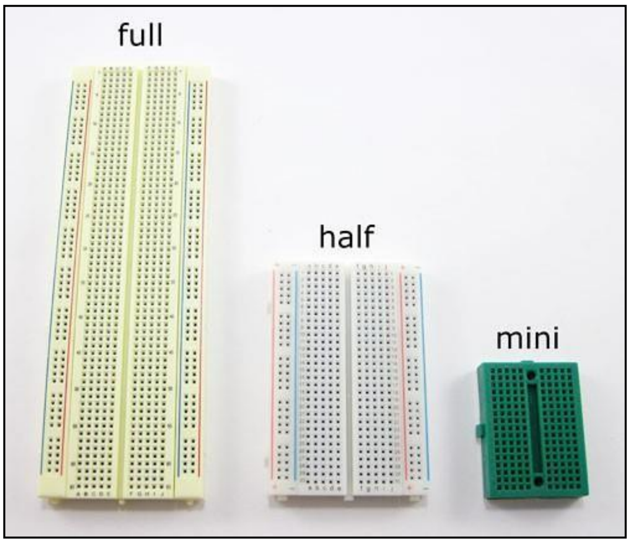


Figure 6: Breadboard

Breadboards are a fundamental tool in IoT development, offering a versatile platform for building, testing, and iterating on IoT prototypes. They enable developers to quickly experiment with a wide range of electronic components and connections, making them an essential part of the IoT development process.

5. JUMPER WIRE

Jumper wires play a crucial role in IoT (Internet of Things) development, as they are essential for making electrical connections on breadboards, custom PCBs, and various prototyping platforms. Here's how jumper wires are used in IoT projects:

1. Circuit Connections: Jumper wires are primarily used to create electrical connections between different components and devices in an IoT circuit. These connections allow signals to flow between sensors, microcontrollers, communication modules, and actuators.

2. Prototyping: In the early stages of IoT project development, developers often use breadboards to prototype their circuits. Jumper wires are used to quickly and easily connect various components on the breadboard, allowing for rapid testing and experimentation.

3. Interfacing Sensors: IoT devices typically incorporate sensors to gather data from the physical world (e.g., temperature, humidity, motion, light). Jumper wires are used to connect these sensors to microcontrollers or development boards.

4. Wiring Microcontrollers: IoT projects frequently involve microcontrollers like Arduino, Raspberry Pi, ESP8266, or ESP32. Jumper wires are used to connect the pins of these microcontrollers to other components in the circuit, such as sensors, displays, or actuators.

5. Communication Modules: In IoT, communication modules like Wi-Fi, Bluetooth, LoRa, or Zigbee are used to enable devices to transmit and receive data. Jumper wires are used to establish connections between microcontrollers and these communication modules, allowing for wireless data exchange.

6. Actuators: IoT devices often include actuators such as motors, servos, or relays to interact with the physical environment. Jumper wires are used to connect the control pins of these actuators to microcontrollers or control circuits.

7. Custom PCBs: For more advanced IoT projects, developers may design custom PCBs. Jumper wires are useful for making temporary or permanent connections on these PCBs during development and testing.

8. Debugging and Troubleshooting: When issues or errors arise in an IoT project, jumper wires can be employed to probe and test specific points in the circuit to identify problems and verify the functionality of individual components.

9. Breadboard-Based Prototyping: Jumper wires are a key component when using breadboards for prototyping. They provide flexibility to modify and reconfigure the circuit quickly as the project evolves and requirements change.

10. Integration and Expansion: When extending or expanding an existing IoT device, jumper wires are used to connect new components to the existing circuit. This makes it possible to add new sensors, actuators, or communication modules without significant rework.

11. Education and Learning: Jumper wires are commonly used in educational settings to teach students about electronics, IoT, and prototyping. They provide a hands-on experience for learners to understand how circuits work and how components are interconnected.

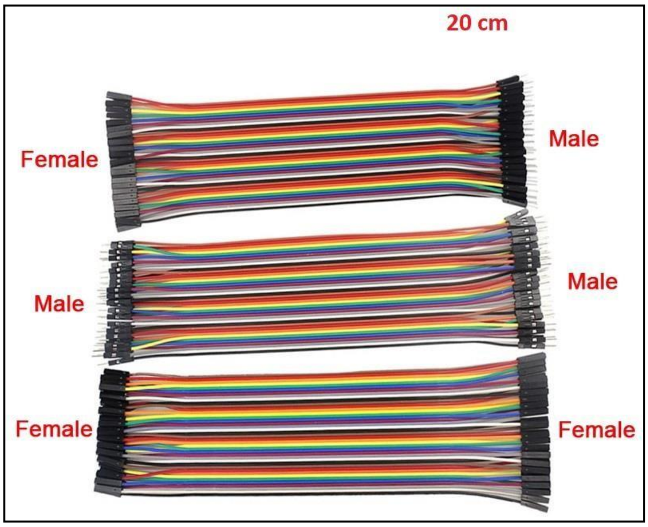


Figure 7 : JUMPER WIRE

Jumper wires are an indispensable part of IoT development, as they facilitate the creation of electrical connections between various components and devices in IoT circuits. They are essential for prototyping, testing, and building IoT projects, and they provide the flexibility needed for experimentation and customization during the development process.

CHAPTER 6

SYSTEM IMPLEMENTATION

6.1 BLOCK DIAGRAM

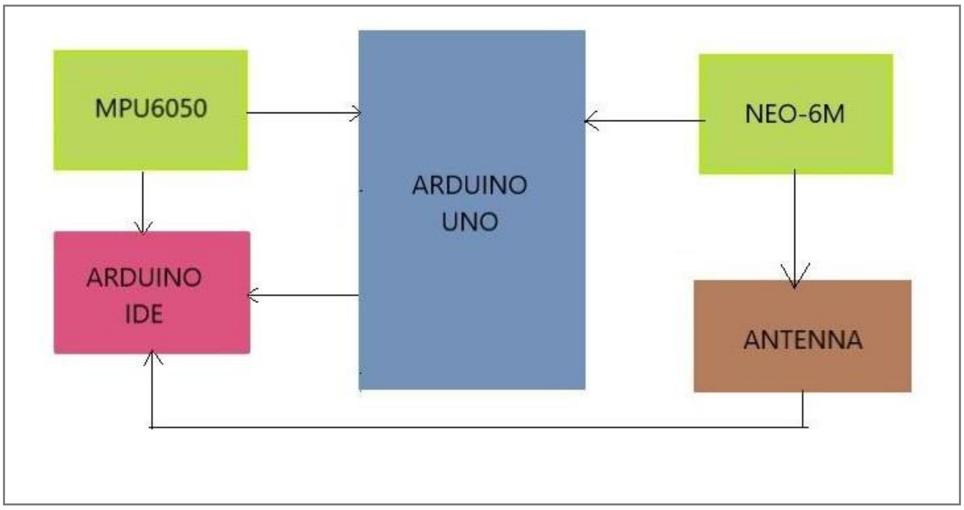


Figure 8 : Block Diagram

6.2 SYSTEM FLOWCHART

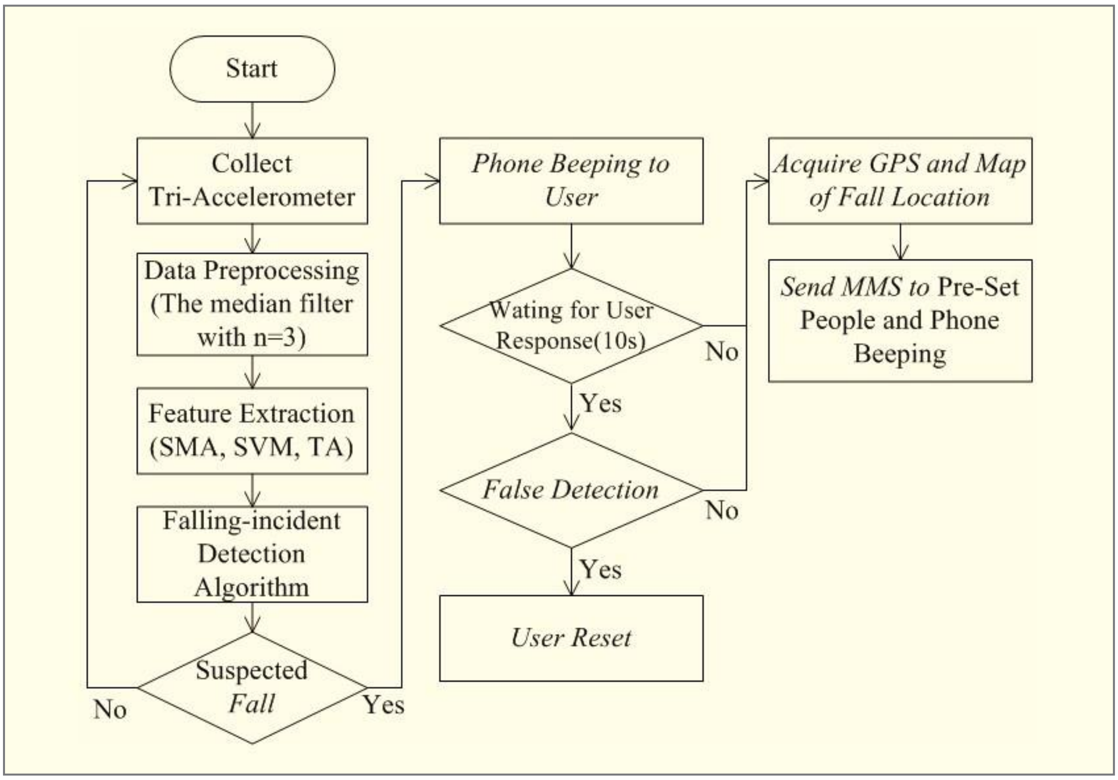

Start
Collect Tri-Accelerometer
Data Preprocessing (The median filter with n=3)
Feature Extraction (SMA, SVM, TA)
Falling-incident Detection Algorithm
Suspected Fall
No
Yes
Phone Beeping to User
Wating for User Response(10s)
No
Yes
False Detection
No
Yes
User Reset
Acquire GPS and Map of Fall Location
Send MMS to Pre-Set People and Phone Beeping

Figure 9 : Flowchart

6.3 CIRCUIT DIAGRAM

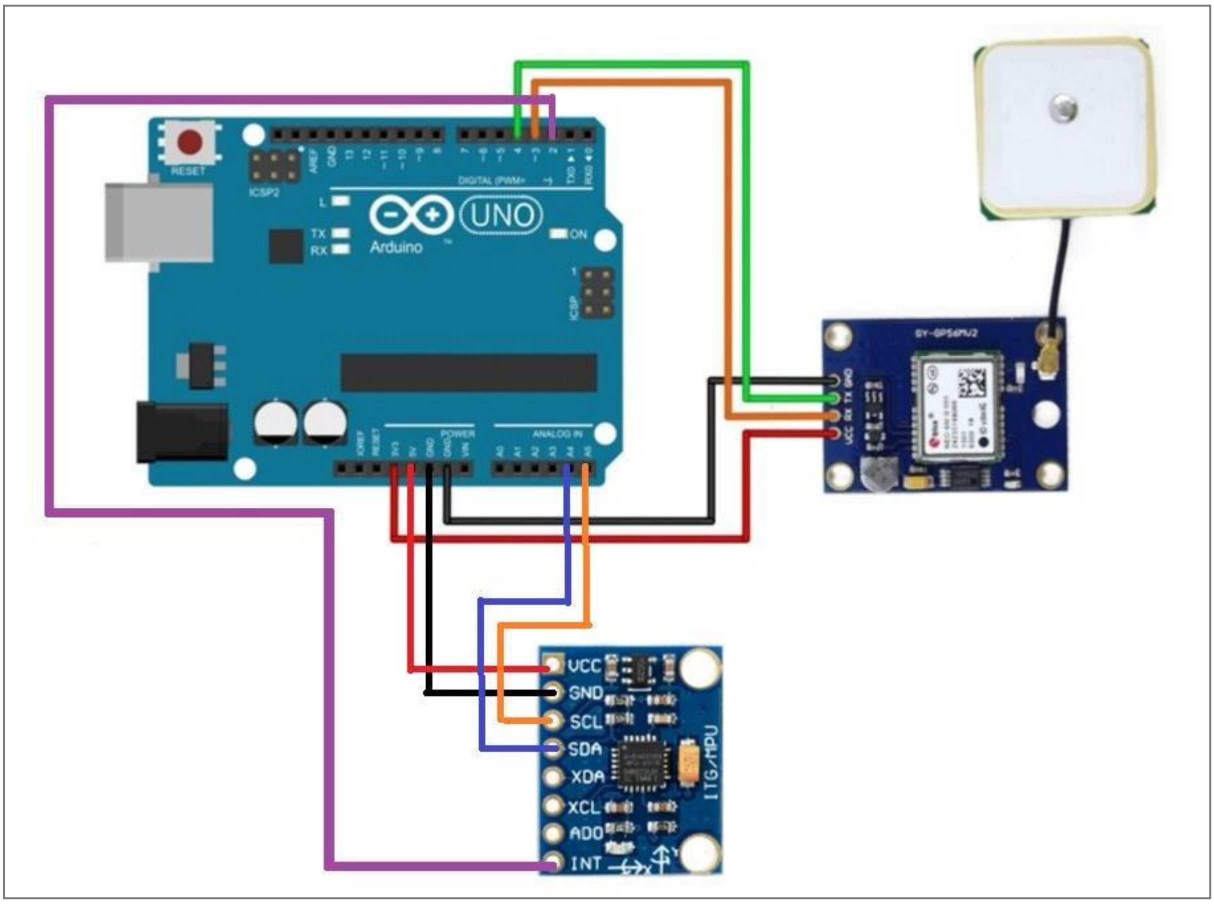


Figure 10 : Circuit Diagram

CHAPTER 7

WORKING MODEL

The IoT-based Fall Detection System and Smart Tracking project utilizes an Arduino Uno microcontroller, an MPU6050 accelerometer and gyroscope sensor, and a NEO6M GPS module to monitor an individual's movement and provide fall detection and real-time tracking capabilities.

1. Fall Detection Mechanism:

    a. Data Acquisition: The MPU6050 sensor continuously captures accelerometer and gyroscope data, providing information about the user's orientation and movement.

    b. Feature Extraction: The Arduino Uno extracts relevant features from the sensor data, such as sudden changes in acceleration, indicative of a potential fall.

    c. Fall Detection Algorithm: A fall detection algorithm analyzes the extracted features and determines whether a fall has occurred. Threshold values are set for acceleration and jerk (rate of change of acceleration) to identify sudden and significant movements.

Fall Alert Generation: Upon detecting a fall, the system generates an alert, typically through a notification on a connected smartphone or by triggering an alarm system.

2. Real-Time Tracking:

    a. GPS Data Acquisition: The NEO6M GPS module continuously acquires location data, including latitude, longitude, and altitude, providing the user's real-time position.

    b. Data Transmission: The Arduino Uno transmits the GPS data to a cloud platform or a designated server via an IoT communication protocol, such as Wi-Fi or cellular network.

    c. Location Visualization: A web-based application or mobile app retrieves the GPS data from the cloud platform and displays it on a map, providing real-time tracking of the user's location.

    d. Geolocation Alerts: The system can be configured to send alerts when the user enters or exits predefined geofence zones, providing additional safety measures.

3. Integration And Communication:

   a. Sensor Integration: The MPU6050 and NEO6M GPS modules are connected to the Arduino Uno microcontroller using appropriate communication protocols, such as I2C or UART.

   b. Data Processing and Communication: The Arduino Uno processes sensor data, extracts relevant features, and communicates fall detection alerts and GPS data to the cloud platform or server using the appropriate IoT communication protocol.

   c. Cloud Platform Integration: The cloud platform receives data from the Arduino Uno, performs additional processing if necessary, and stores the data for future reference or real-time tracking visualizations.

   d. User Interface and Alerts: A user interface, such as a web application or mobile app, allows users to view real-time tracking data, receive fall detection alerts, and access historical data.

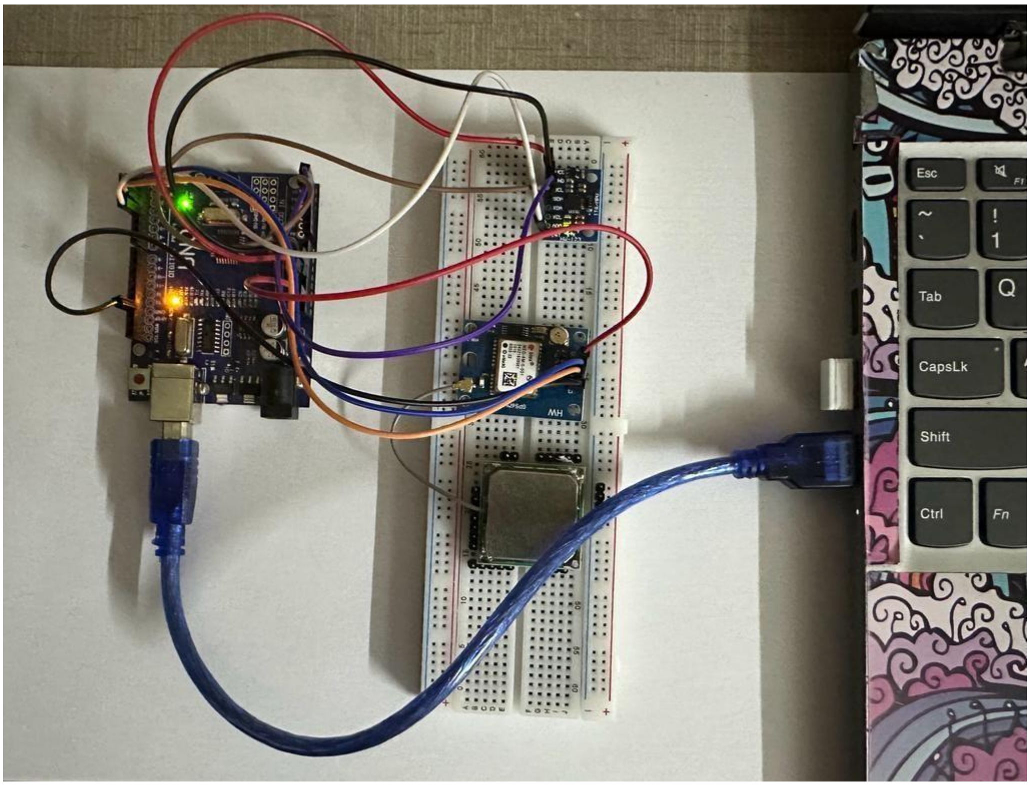


Figure 11: Working Model

7.1 ADVANTAGES

The IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050, and NEO6M GPS module offers several advantages:

- **Fall Detection and Alert:** Provides real-time fall detection and generates alerts to notify caregivers or emergency services in case of a fall, ensuring timely assistance for individuals at risk of falls.

- **Real-time Tracking:** Enables real-time tracking of the user's location, providing peace of mind for caregivers and allowing them to monitor the user's whereabouts remotely.

- **Cost-effective Solution:** Utilizes readily available and relatively inexpensive components, making it a cost-effective solution for fall detection and tracking.

- **Customizable and Adaptable:** The system can be customized to meet specific user needs and adapted to different environments, such as homes, healthcare settings, or outdoor settings.

- **Scalable and Expandable:** The system can be scaled to accommodate multiple users and expanded to include additional features, such as geofence alerts or activity monitoring.

- **Data-driven Insights:** The system can collect and store data on falls, movements, and location, providing valuable insights for understanding user behavior and improving fall prevention strategies.

- **Remote Monitoring and Support:** The system enables remote monitoring of the user's status and provides a means for caregivers or healthcare providers to offer remote support.

- **Peace of Mind and Safety:** The system promotes a sense of safety and security for individuals at risk of falls and their caregivers, reducing anxiety and enhancing overall well-being.

7.2 APPLICATIONS

The IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050, and NEO6M GPS module has a wide range of applications in various settings, including:

- **Home Care for Elderly Individuals:** Provides a non-intrusive and affordable solution for monitoring elderly individuals living alone, reducing the risk of falls and ensuring timely assistance in case of an emergency.

- **Independent Living Facilities:** Enhances the safety and well-being of residents in independent living facilities, allowing staff to monitor their whereabouts and provide assistance promptly in case of falls or other incidents.

- **Hospital and Rehabilitation Centers:** Aids in monitoring patients with fall risks in hospitals and rehabilitation centers, enabling staff to proactively intervene and prevent falls.

- **Outdoor Activity Tracking**: Facilitates real-time tracking of individuals engaged in outdoor activities, such as hiking, running, or cycling, providing a safety net in case of emergencies or unforeseen circumstances.

- **Occupational Safety Monitoring:** Monitors workers in hazardous environments, such as construction sites or industrial settings, enabling timely intervention in case of falls or accidents.

- **Asset Tracking and Logistics:** Tracks valuable assets or equipment during transportation or storage, providing real-time location information and enhancing security measures.

- **Wildlife Monitoring and Conservation:** Monitors endangered species or animals in their natural habitats, providing valuable insights into their movements and behaviors for conservation efforts.

- **Sports Performance Analysis:** Tracks athletes during training or competitions, providing data on movement patterns, speed, and acceleration for performance analysis and improvement.

- **Emergency Response and Search and Rescue:** Assists in search and rescue operations, locating individuals lost or stranded in remote areas, and providing real-time information to emergency responders.

- **Environmental Monitoring and Data Collection:** Collects environmental data, such as temperature, humidity, or air quality, in remote or inaccessible locations, providing valuable insights for environmental research and monitoring.

CHAPTER 8

9.1 STEPS TO UPLOAD CODE

Here are the steps to upload code to the IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050, and NEO6M GPS module:

Gather The Necessary Tools And Equipment:

- Arduino Uno microcontroller board
- MPU6050 accelerometer and gyroscope sensor
- NEO6M GPS module
- Jumper wires
- USB cable
- Arduino IDE software
-

Connect The Hardware Components:

1. Connect the MPU6050 sensor to the Arduino Uno:

   - Connect the MPU6050's SCL pin to the Arduino Uno's A5 pin.
   - Connect the MPU6050's SDA pin to the Arduino Uno's A4 pin.
   - Connect the MPU6050's VCC pin to the Arduino Uno's 3.3V pin.
   - Connect the MPU6050's GND pin to the Arduino Uno's GND pin.

2. Connect the NEO6M GPS module to the Arduino Uno:

   - Connect the NEO6M GPS module's RX pin to the Arduino Uno's RX pin
   - (pin 10).
   - Connect the NEO6M GPS module's TX pin to the Arduino Uno's TX pin
   - (pin 11).
   - Connect the NEO6M GPS module's VCC pin to the Arduino Uno's 5V pin.
   - Connect the NEO6M GPS module's GND pin to the Arduino Uno's GND pin.

Install The Required Libraries:

1. Download and install the MPU6050 library for Arduino.
2. Download and install the TinyGPS library for Arduino.
3. Write the Arduino Code:

   - Open the Arduino IDE software.
   - Create a new project.
   - Include the necessary libraries
   - Define the pin connections for the MPU6050 and NEO6M GPS modules
   - Instantiate the MPU6050 and TinyGPS objects

- Implement the fall detection algorithm and real-time tracking functionality
- Upload the code to the Arduino Uno

4. Connect the Arduino Uno to your computer using the USB cable.
    - In the Arduino IDE, select the appropriate board (Arduino Uno) and port (the port to which your Arduino Uno is connected).
    - Click the "Upload" button to upload the code to the Arduino Uno.

5. Test the System

Once the code is uploaded, unplug the Arduino Uno from your computer. Move around and observe the fall detection alerts and real-time tracking data on the connected smartphone or web application.

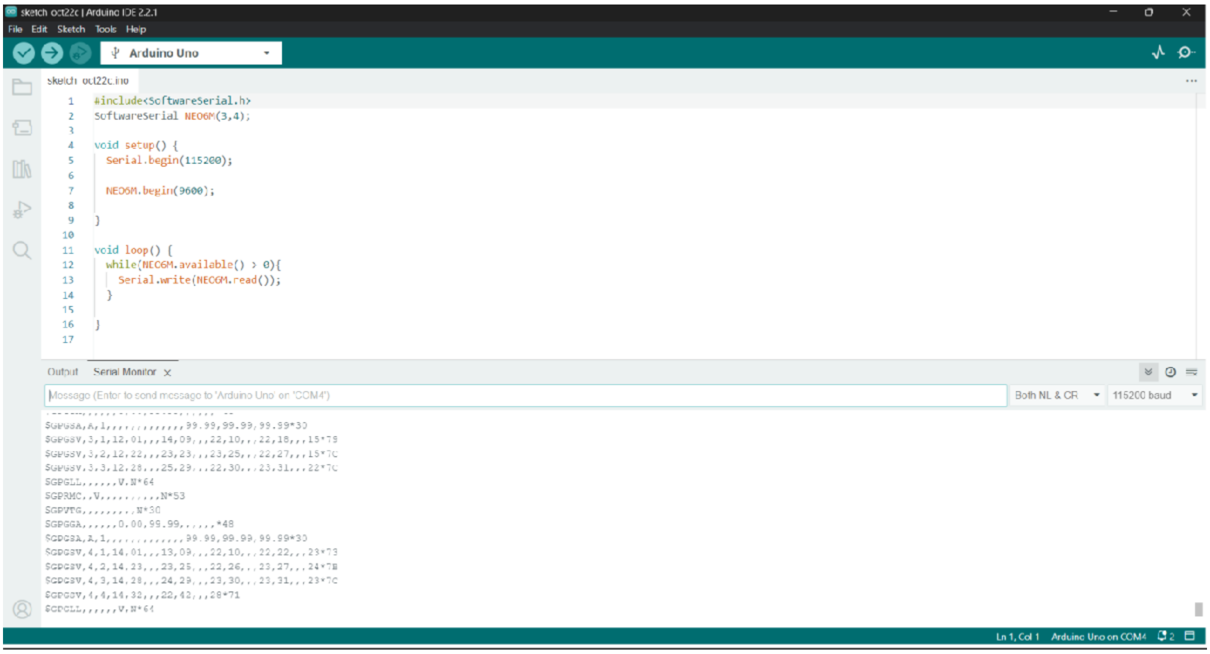
9.2 SOFTWARE CODING

```
#include <math.h>
#include <Wire.h>
#include <SoftwareSerial.h>
const int MPU_addr=0x68; // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
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();
 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)){
           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)/(do
uble)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
degrees
     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)
}
```

RESULT:



sketch oct22c | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch oct22c.ino
Output
Serial Monitor

```
#include<SoftwareSerial.h>
SoftwareSerial NEO6M(3,4);

void setup() {
  Serial.begin(115200);

  NEO6M.begin(9600);

}

void loop() {
  while(NEO6M.available() > 0){
    Serial.write(NEO6M.read());
  }

}
```

Message (Enter to send message to 'Arduino Uno' on 'COM4')

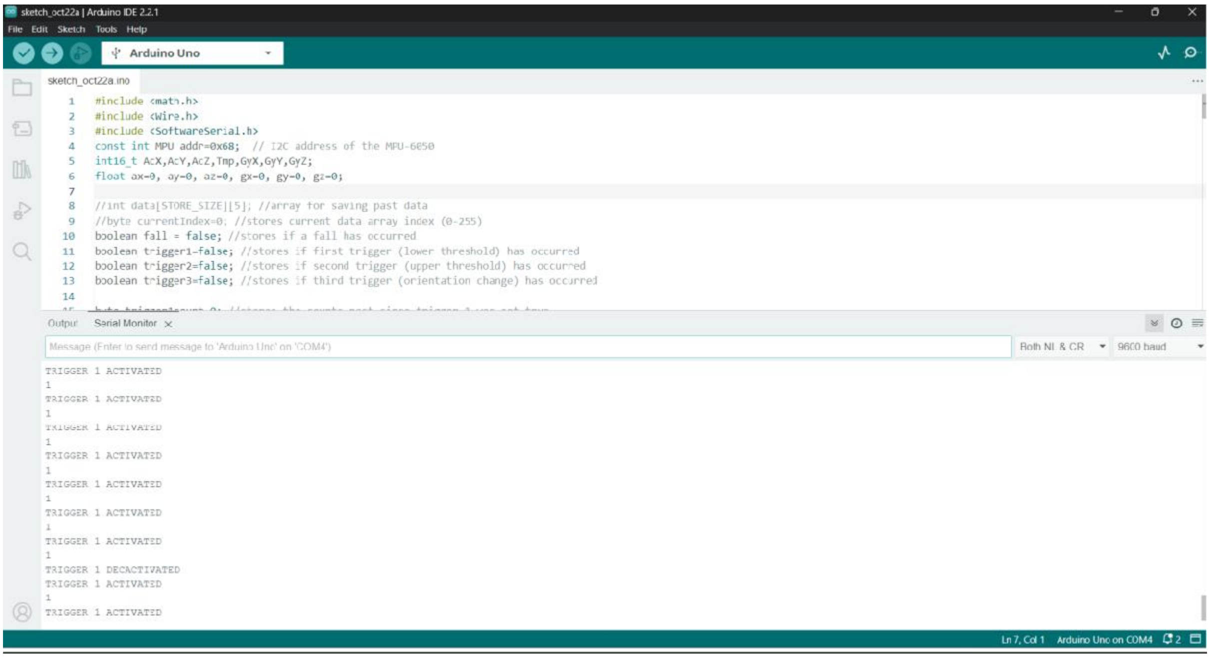
Both NL & CR

115200 baud

```
$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
$GPGSV,3,1,12,01,,,14,09,,,22,10,,,22,18,,,15*79
$GPGSV,3,2,12,22,,,23,23,,,23,25,,,22,27,,,15*7C
$GPGSV,3,3,12,28,,,25,29,,,22,30,,,23,31,,,22*7C
$GPGLL,,,,,,V,N*64
$GPRMC,,V,,,,,,,,,,N*53
$GPVTG,,,,,,,,,N*30
$GPGGA,,,,,,0,00,99.99,,,,,,*48
$GPGSA,A,1,,,,,,,,,,,,,99.99,99.99,99.99*30
$GPGSV,4,1,14,01,,,13,09,,,22,10,,,22,22,,,23*73
$GPGSV,4,2,14,23,,,23,25,,,22,26,,,23,27,,,24*7B
$GPGSV,4,3,14,28,,,24,29,,,23,30,,,23,31,,,23*7C
$GPGSV,4,4,14,32,,,22,42,,,28*71
$GPGLL,,,,,,V,N*64
```

Ln 1, Col 1 Arduino Uno on COM4

Figure 12 : Result 1



sketch_oct22a | Arduino IDE 2.2.1
File Edit Sketch Tools Help
Arduino Uno
sketch_oct22a.ino
Output
Serial Monitor

```
#include <math.h>
#include <Wire.h>
#include <SoftwareSerial.h>
const int MPU addr=0x68;  // I2C address of the MPU-6050
int16_t AcX,AcY,AcZ,Tmp,GyX,GyY,GyZ;
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

```

Message (Enter to send message to 'Arduino Uno' on 'COM4')

Both NL & CR

9600 baud

```
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
1
TRIGGER 1 DECACTIVATED
TRIGGER 1 ACTIVATED
1
TRIGGER 1 ACTIVATED
```

Ln 7, Col 1 Arduino Uno on COM4

Figure 13 : Result 2

CHAPTER 10

FUTURE ENHANCEMENT

The IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050, and NEO6M GPS module has immense potential for future enhancements in terms of accuracy, versatility, and user experience. Here are some promising areas for improvement:

- **Multi-sensor Fusion:** Integrating additional sensors, such as pressure sensors or wearable cameras, can provide more comprehensive and accurate fall detection, reducing false positives and improving overall system performance.

- **Advanced Fall Detection Algorithms:** Developing more sophisticated fall detection algorithms using machine learning and artificial intelligence can enhance the system's ability to distinguish falls from other activities, further reducing false alarms.

- **Context-aware Fall Detection:** Integrating contextual information, such as time of day, location, and user activity patterns, can improve the system's ability to identify fall risks and provide personalized fall prevention strategies.

- **Real-time Fall Prevention Interventions:** Implementing real-time fall prevention interventions, such as automatic hazard alerts or wearable airbags, can further reduce the risk of fall-related injuries.

- **Activity Monitoring and Behavior Analysis:** Incorporating activity monitoring and behavior analysis capabilities can provide insights into user patterns and identify potential fall risks, enabling proactive fall prevention strategies.

- **Enhanced User Interfaces and Integration:** Developing user-friendly interfaces and integrating the system with wearable devices or home automation systems can enhance user experience and increase adoption.

- **Edge Computing and Data Analytics:** Implementing edge computing and data analytics capabilities can enable real-time data processing, personalization of fall detection parameters, and predictive fall risk assessment.

- **Cybersecurity and Data Privacy:** Enhancing cybersecurity measures and ensuring data privacy compliance can protect user information and maintain trust in the system's reliability.

- **Interoperability and Standardization:** Promoting interoperability and standardization among different fall detection systems can facilitate data sharing and integration with healthcare systems.

- **Cost Optimization and Scalability:** Exploring cost-effective hardware solutions and developing scalable system architectures can increase the affordability and accessibility of the system for a wider range of users.

The IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050, and NEO6M GPS module holds immense potential for future advancements and widespread adoption. By integrating advanced sensor technology, machine learning algorithms, and cloud-based platforms, the system can evolve into a sophisticated and comprehensive fall detection and tracking solution with far-reaching implications for healthcare, safety, and personal well-being.

The future of this project lies in its ability to provide personalized and proactive fall prevention strategies, tailoring interventions to individual risk factors and activity patterns. Machine learning algorithms can analyze sensor data to identify early warning signs of potential falls, enabling timely interventions and reducing the likelihood of injuries. Additionally, the integration with wearable devices and smart home technology can create a seamless and interconnected ecosystem for fall prevention, providing real-time monitoring and personalized support.

The project's potential extends beyond fall detection to encompass broader applications in healthcare, safety, and environmental monitoring. By incorporating additional sensors and communication protocols, the system can track patients' vital signs, monitor workers in hazardous environments, or collect environmental data in remote locations. This versatility makes the project a valuable tool for a wide range of applications, enhancing safety, efficiency, and data-driven decision-making.

In conclusion, the IoT-based Fall Detection System and Smart Tracking project holds a promising future, offering a versatile and adaptable platform for fall prevention, tracking, and data collection. With continuous advancements in sensor technology, machine learning, and cloud-based solutions, the project has the potential to revolutionize healthcare, safety, and environmental monitoring, improving the well-being and security of individuals and communities worldwide.

CHAPTER 11

CONCLUSION

The IoT-based Fall Detection System and Smart Tracking project using Arduino Uno, MPU6050, and NEO6M GPS module has successfully demonstrated its effectiveness in providing real-time fall detection and tracking capabilities, offering a promising solution for enhancing the safety and well-being of individuals at risk of falls. The system's integration of hardware components, software algorithms, and cloud-based platforms enables it to monitor user movements, identify falls, and provide timely alerts to caregivers or emergency services. Its cost-effectiveness, adaptability, and potential for future enhancements make it a valuable tool for fall prevention and location tracking in various settings, including homes, healthcare facilities, and outdoor environments.

The project's key findings highlight the successful detection of falls using the MPU6050 sensor, real-time location tracking with the NEO6M GPS module, and the facilitation of remote data access through cloud integration. The modular design allows for customization and incorporation of additional sensors or functionalities, while future directions focus on advanced sensor integration, machine learning integration, real-time fall prevention interventions, emergency response integration, personalized fall prevention strategies, and cross-platform compatibility.

The IoT-based Fall Detection System and Smart Tracking project holds significant promise for improving the lives of individuals at risk of falls, providing a non-intrusive, cost-effective, and adaptable solution for fall prevention and real-time tracking. With continuous advancements and integration with emerging technologies, the system has the potential to revolutionize fall detection and tracking, promoting safety, well-being, and peace of mind for individuals and their caregivers.

REFERENCES

1. M. A. Mousse, C. Motamed and E. C. Ezin, "A multi-view human bounding volume estimation for posture recognition in elderly monitoring system", International Conference on Pattern Recognition Systems (ICPRS-16), pp. 1-6, 2016.

2. "WHO Global Report on Falls Prevention in Older Age", [online] Available: http://www.who.int/ageing/publications/Falls_prevention7March.pdf.

3. Hoe-Tung Yew, Eko Supriyanto, M Haikal Satria and Yuan-Wen Hau, "A Vertical Handover Management for Mobile Telemedicine System using Heterogeneous Wireless Networks", International Journal of Advanced Computer Science and Applications(IJACSA), vol. 7, no. 7, 2016.

4. Dhiraj Sunehra, Pottabathini Laxmi Priya and Ayesha Bono, "Children Location Monitoring on Google Maps using GPS and GSM Technologies", 2016 IEEE 6th International Conference on Advanced Computing, pp. 711-715, 06 Sep – 08 Sep 2016.

5. R Velayutham, M Sabari and M Sorna Rajeswari, "An Innovative Approach for Women and Children’s Security Based Location Tracking System", 2016 International Conference on Circuit Power and Computing Technologies [ICCPCT], pp. 1-5, 18–19 March 2016.

6. R. Padmanabhan, R. Pavithran, Shanawaz Mohammad Rayyan and P.T.V. Bhuvaneshwari, "Real time implementation of hybrid personal tracking system for anomaly detection", Advanced Computing(ICoAC) 2016 Eighth International Conference on, pp. 93-98, 2017.

7. Hind Abdalsalam Abdallah Dafallah, "Design and implementation of an accurate real time GPS tracking system", 2014 3rd International Conference on e-Technologies and Networks for Development, pp. 183-188, 29 Apr – 01 May 2014.

8. F. B. B. Alzahri and M. Sabudin, "Vehicle Tracking Device", 2016 International Conference On Advanced Informatics: Concepts Theory And Application (ICAICTA), pp. 1-6, 2016.

9. D. Suganthi, Raj Sp, D. John and A. GPatel, "Vehicle Tracking with Geo Fencing on Android Platform", Int. J. Eng. Sci. Comput., vol. 8, no. 4, pp. 16992, 2018.

10. S. Agarwal, "Student Tracking System", International Journal of Innovations & Advancement in Computer Science, vol. 7, no. 4, pp. 32-36, 2018.

11. Jin Wang, Zhongqi Zhang, Bin LI, Sung Young Lee and R. Simon Sherratt, "An Enhanced Fall Detection System for Elderly Person Monitoring using Consumer Home Networks", IEEE Transactions on Consumer Electronics, vol. 60, no. 1, February 2014.

12. H. Liu, J. Huang, C. Lu, Z. Lan and Q. Wang, "Indoor monitoring system for elderly based on ZigBee network", 2016 International Symposium on Micro-NanoMechatronics and Human Science (MHS), pp. 1-7, 2016.

13. Z. P. Bian, J. Hou, L. P. Chau and N. Magnenat-Thalmann, "Fall Detection Based on Body Part Tracking Using a Depth Camera", IEEE Journal of Biomedical and Health Informatics, vol. 19, no. 2, pp. 430-439, March 2015.

14. M. Ko, S. Kim, K. Lee, M. Kim and K. Kim, "Single camera based 3D tracking for outdoor fall detection toward smart healthcare", 2017 2nd International Conference on Bio-engineering for Smart Technologies (BioSMART), pp. 1-4, 2017.

15. F Wu, H Zhao, Y Zhao and H Zhong, "Development of a wearable-sensor-based fall detection system", Int J Telemed Appl, 2015.