Smart Home Security System

Prepared by: Rohit Rokka
Tutor: Mahdi Saki

Table of Contents

Executive Summary

Smart home security system project is intended to create a security system that will use in order protect homeowners from intruders. This executive summary provides an overview of the Smart Home Security System project utilizing ThingSpeak as a key component. The project aims to develop a robust and efficient security system that enhances the safety and protection of a smart home environment.

The system architecture consists of multiple hardware components, including sensors and a microcontroller such as Arduino. These components work together to monitor and control different aspects of the smart home security system. In this project, an MPU sensor and Arduino Uno Rev2 Wi-Fi have been chosen as the main components to detect and respond to security threats effectively. With the increasing popularity of smart home technology, it is crucial to address security concerns and provide homeowners with a reliable and efficient security solution. The Smart Home Security System project focuses on implementing a robust system that not only detects potential security threats but also responds effectively to mitigate risks.

Introduction

Project Overview

The proposed project will be able to detect motion of anyone and send an alert message to the homeowners. The project aims to develop a comprehensive Smart Home Security System that leverages the capabilities of ThingSpeak, an IoT analytics platform, to enhance the safety and protection of a smart home environment. The implementation of the Smart Home Security System encompasses both hardware and software aspects. The hardware implementation involves setting up the MPU sensor and Arduino Uno Rev2 Wi-Fi, ensuring proper connections and compatibility. The software implementation focuses on programming the Arduino board, utilizing libraries and frameworks to enable seamless communication between the MPU sensor and the Arduino Uno Rev2 Wi-Fi.

Purpose of the document

The purpose of this document is to apprise about the project goals and determined, its complexity, its risks, its aims to achieve and the feedback acknowledging the various challenges and considerations that need to be considered.

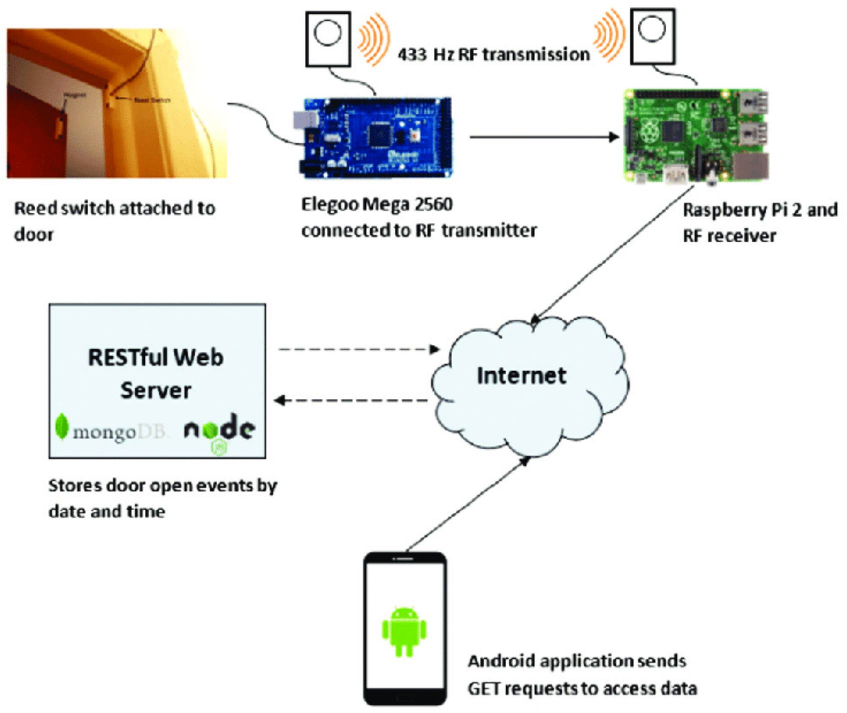
Background

Smart home security system is a major factor for ensuring the safety and project of the smart home environment and its occupants. The primary goal is to implement robust security measures to prevent unauthorized access, detect potential threats, and respond effectively to security breaches.

Project Complexity

The complexity of the project is expected to be medium level in terms of connectivity due to the need of connection of hardware and software components to synchronize with each other.
The key source of the complexity was the wiring connection for connecting between microcontroller and the sensor. Different devices use different communication protocols, such as Wi-Fi, Bluetooth, Zigbee, or Z-Wave. Integrating these devices into a cohesive security system requires expertise in handling multiple protocols and ensuring interoperability between them. Managing the data generated by smart home security devices, including sensor data, video feeds, and user information, requires careful consideration. Ensuring data privacy, security, and compliance with relevant regulations can be complex, particularly when dealing with sensitive information. In terms of system reliability and redundancy, building a reliable and fault-tolerant smart home security system requires redundancy and failover mechanisms. Implementing backup systems, redundancy in communication channels, and ensuring system availability during power outages or network disruptions can be complex. Programming microcontroller in order to read the data from the sensor and sending it to thingspeak dashboard was complex coding.

System design


433 Hz RF transmission
Reed switch attached to door
Elegoo Mega 2560 connected to RF transmitter
Raspberry Pi 2 and RF receiver
RESTful Web Server
mongoDB
node
Stores door open events by date and time
Internet
Android application sends GET requests to access data

Source: Hoque 2019

The project system design provides enhanced security and monitoring capabilities using the MPU6050 sensor and Arduino Uno Rev2 controller. The system architecture revolves around the seamless integration of these components, along with the utilization of the ThingSpeak IoT platform and Twilio for SMS alerts. The MPU6050 sensor is connected to the Arduino board. The sensor captures motion and orientation data, which is then processed by the Arduino board. The Arduino board acts as the brain of the system, controlling the data flow and communication. The Arduino board has a built in WIFI module which enables wireless communication that allows the system to connect to the IoT cloud-based platform service. The Arduino board securely transmits the processed sensor data to our IoT platform for storage and visualization where users can monitor real-time sensor data, visualize trends, and set up customized alerts.

System requirements

Hardware design

MPU6050 Sensor

For the motion capture I have chosen MPU6050 a 6-axis motion tracking module that combines a 3-axis accelerometer and a 3-axis gyroscope. Its integration into your smart home security system adds crucial motion and orientation sensing capabilities. The MPU6050 sensor measures acceleration and rotation rates in three dimensions. It provides precise motion data, including tilt, shake, and movement detection. Its compact size and low power consumption make it suitable for embedded applications like your smart home security system. The MPU6050 sensor is connected to the micro controller using the appropriate wiring scheme so that it can capture and monitor various motion related data points.

Arduino Uno ReV2

The main brain of the system for this project is the microcontroller Arduino Uno ReV2 based on ATmega4809. It combines the features of the classic Arduino Uno with built-in Wi-Fi connectivity, making it suitable for IoT applications. It serves as the central processing unit and control unit for my project. It runs the firmware or code that composes the operation of the various components, including the MPU6050 sensor and Wi-Fi module. The Arduino Uno Rev2 Wi-Fi facilitates the integration of the MPU6050 sensor into our system. It communicates with the sensor using the I2C protocol, retrieves the motion and orientation data, and performs necessary calculations or analysis on the received data. The Arduino board processes the sensor data formats it according to the required protocol or format and transmits it to the designated endpoints, into our IoT platform for visualization and storage. The Arduino Uno Rev2 Wi-Fi is compatible with a wide range of shields and modules, which can be added to expand its capabilities. This provides flexibility for future enhancements or additions to our smart home security system.

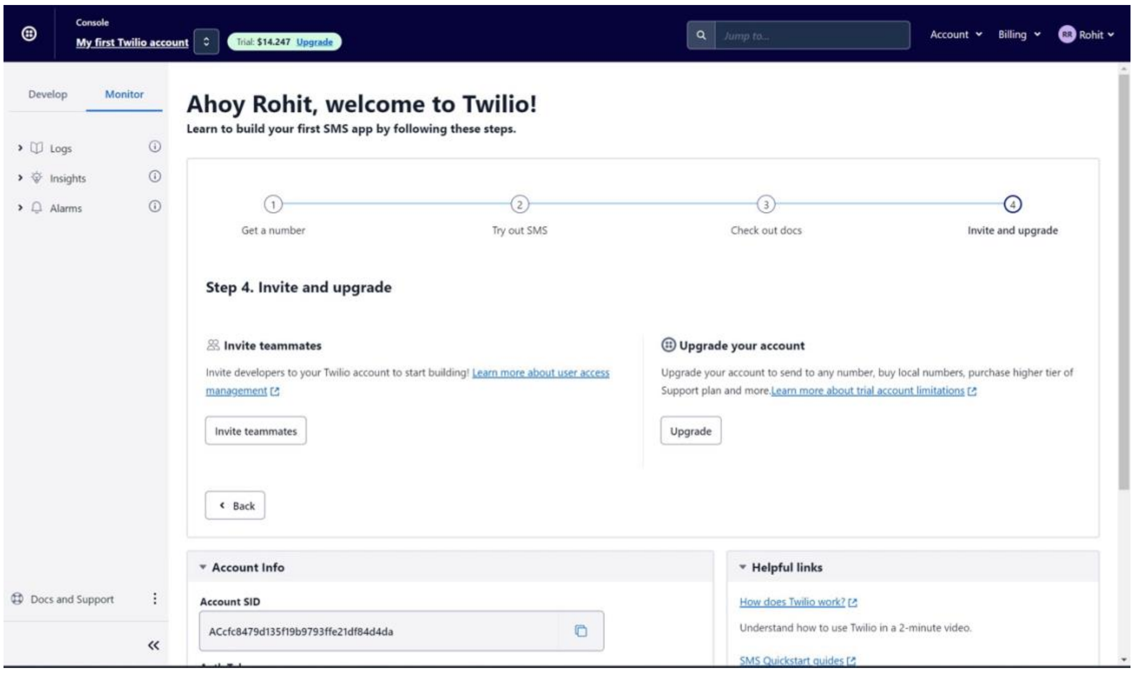
Software design

ThingSpeak

At the beginning of the project, the initial plan was to connect to AWS cloud service but because of the complexity of configuring and managing its services.

ThinkSpeak is like AWS, an IoT platform that allows us to collect, analyse, and visualize sensor data in real-time. It provides an easy-to-use interface for storing and retrieving data, as well as tools for creating custom visualizations and setting up alerts. ThingSpeak offers an Arduino library that enables seamless integration with the Arduino Uno Rev2 Wi-Fi microcontroller. This library provides functions and methods to connect to the ThingSpeak platform, transmit data, and perform other operations. To send our MPU sensor data to ThingSpeak, we have established connection between our microcontroller and the ThingSpeak Servers. ThingSpeak allows us to configure alerts based on specific conditions or thresholds. we have set up triggers that send notifications or alerts via SMS notification and other supported channels. The homeowner will receive an alert when motion is detected outside predefined hours or when an unusual activity pattern is observed.

Twilio



Ahoy Rohit, welcome to Twilio!

Learn to build your first SMS app by following these steps.

1 Get a number — 2 Try out SMS — 3 Check out docs — 4 Invite and upgrade

Step 4. Invite and upgrade

Invite teammates

Invite developers to your Twilio account to start building! Learn more about user access management

Invite teammates

Upgrade your account

Upgrade your account to send to any number, buy local numbers, purchase higher tier of Support plan and more.Learn more about trial account limitations

Upgrade

Back

Account Info

Account SID

ACcfc8479d135f19b9793ffe21df84d4da

Helpful links

How does Twilio work?

Understand how to use Twilio in a 2-minute video.

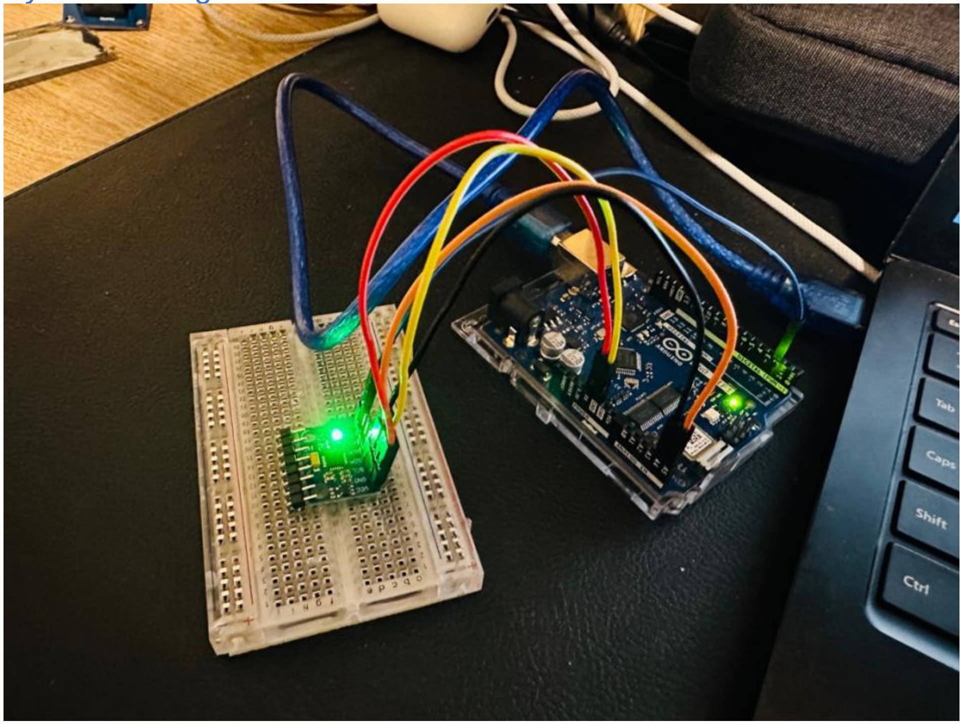
SMS Quickstart guides

For the cloud communications platform to provide APIs for sending and receiving text messages (SMS) and handling other communication tasks programmatically, we have setup Twilio account, it basically offers a simple and reliable way to incorporate SMS functionality into our smart home security system. We coded Twilio API to our Arduino board that allow us programmatically send SMS messages into the respected registered devices. Proper error handling and retry mechanisms were coded when sending SMS alerts using Twilio. This ensures that in case of network issues or temporary failures, the system attempts to resend the alerts or handles the errors gracefully.

Communication protocol

Since Arduino Uno Rev2 Wi-Fi was used in the project, the communication between the microcontroller, sensors and other components were achieved wirelessly. The built-in Wi-Fi capability allows the Arduino board to connect to our home network or directly to the internet. The communication protocol used over the wireless network is typically based on the Internet Protocol (IP). This means that devices within your network, including the Arduino board and the ThingSpeak platform, will have unique IP addresses assigned to them. The communication between these devices happens using IP-based protocols. For the data transmission, subscription, and request/response mechanisms, MQTT (Message Queuing Telemetry Transport), and HTTP protocols are being used.

System Configuration



To ensure proper functionality and integration, Various system configurations were performed. Firstly, we have linked our MPU6050 sensor and Arduino board with jumper wires to establish the connection between them and then the microcontroller to respected device using USB cable. Ensuring that the Arduino code includes the necessary configurations for integrating the components which includes setting up the communication protocols (such as HTTP or MQTT) between the Arduino Uno Rev2 Wi-Fi and ThingSpeak, as well as configuring the API requests to send sensor data to ThingSpeak and SMS alerts via Twilio.

Device Registration and Authentication

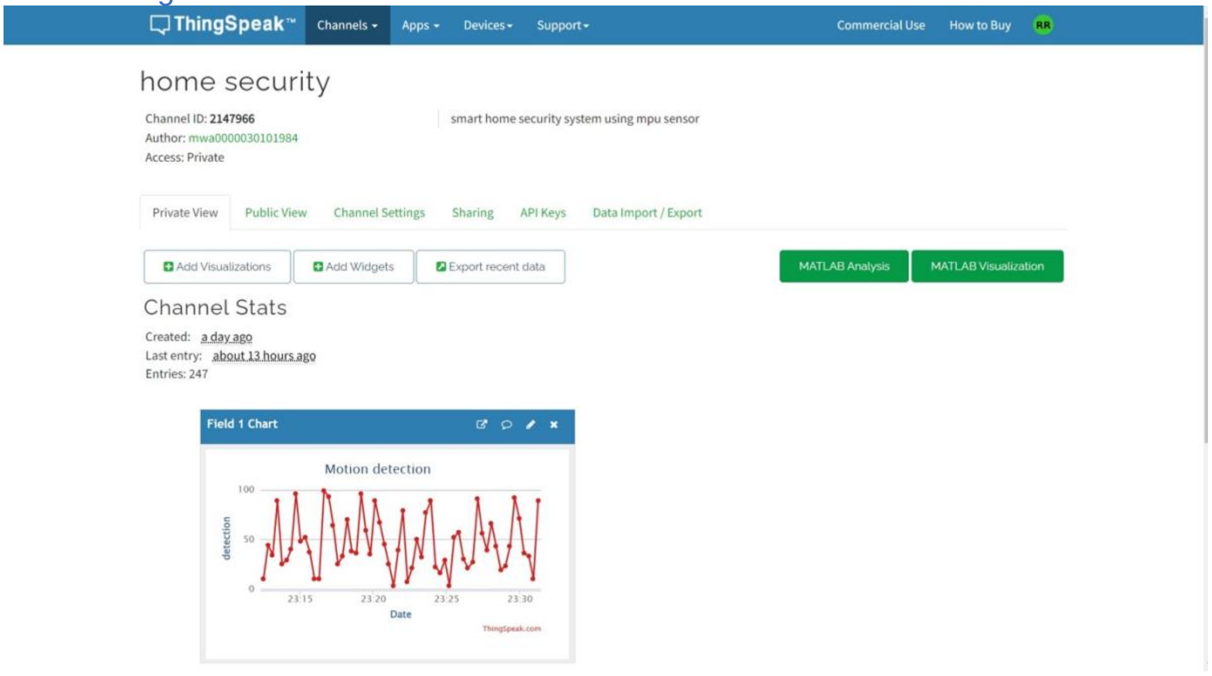
Device registration and authentication are important aspects of our smart home security system which ensure that only authorized devices can access and interact with our system. Necessary information such as device identifiers, unique device keys and mac addresses were collected to enrol the devices. SSl/TLS encryption protocols were implemented to ensure the communication between the devices and the central system is secured. This helps prevent unauthorized access and data interception.

Network Setup and Connectivity

Network setup and connectivity are crucial for establishing communication between devices and ensuring the smooth operation of the system. First of all, home network was properly configured and ready to support the connectivity requirements of the project which includes Wi-Fi coverage, internet connection.

Implementation

Data Storage and dashboard



home security

Channel ID: 2147966
Author: mwa0000030101984
Access: Private

smart home security system using mpu sensor

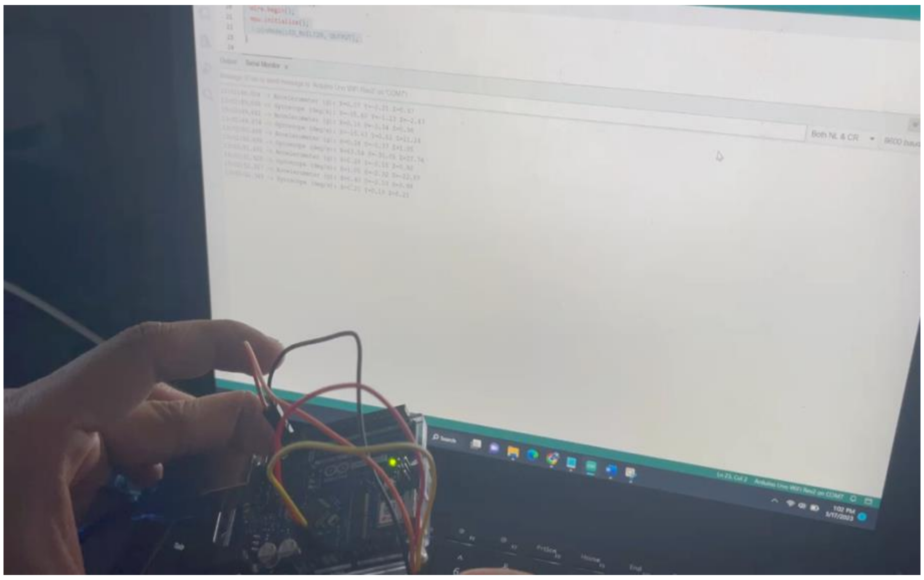
Channel Stats

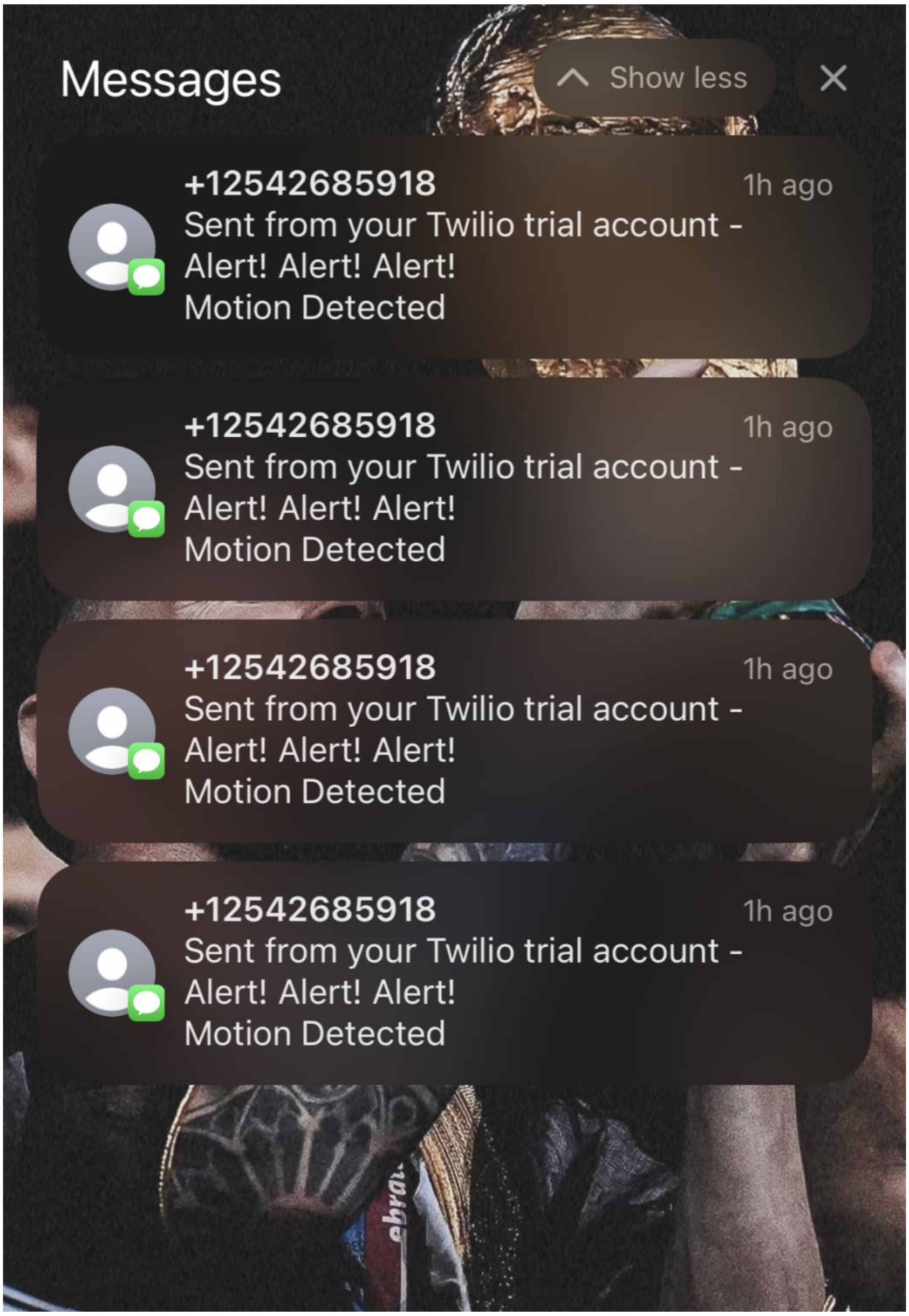
Created: a day ago
Last entry: about 13 hours ago
Entries: 247

ThingSpeak will provide a platform to store and visualize the sensor data. We have created a channel that is linked to the central processing system. There is motion detection field into the channel which will store our motion captured data. ThingSpeak stores incoming sensor data in the designated fields of our channel using a time-series database approach. When data is sent to ThingSpeak, it is associated with a specific channel and field(s) within that channel. When the Arduino Uno Rev2 Wi-Fi microcontroller sends sensor data to ThingSpeak, it includes the data values for each field in the HTTP request. ThingSpeak captures the data and stores it in its database, associating it with the corresponding channel and field(s). Each data entry is timestamped with the time of arrival, allowing for time-based analysis and visualization. Once the data is retrieved, we can analyse and visualize it to gain insights into our smart home security system. Homeowners can perform statistical analysis, variance detection or trend analysis.

Testing

We have performed a demonstration testing to ensure that when the motion is detected by the sensor, it will send it to IoT platform and send an alert message to the homeowners.




Messages

Show less

+12542685918 1h ago
Sent from your Twilio trial account -
Alert! Alert! Alert!
Motion Detected

+12542685918 1h ago
Sent from your Twilio trial account -
Alert! Alert! Alert!
Motion Detected

+12542685918 1h ago
Sent from your Twilio trial account -
Alert! Alert! Alert!
Motion Detected

+12542685918 1h ago
Sent from your Twilio trial account -
Alert! Alert! Alert!
Motion Detected

Security Risks and Issues

The are various potential security risks and issues that may arise to the project system.

- Data breaches: one of the primary concerns on smart home security system is the potential for unauthorized access to the system's data. This can occur if proper security measures are not in place, such as weak passwords, unencrypted data transmission, or vulnerabilities in the network or cloud services used.

- Social engineering attacks: people might attempt to mislead users into providing sensitive information or granting unauthorized access. Such as phishing attacks.
- System malfunction: Potential technical issues and malfunctions might occur into the system that could impact its security. So, we have ensured regular tests, updates maintain the hardware and software components to minimize the risk of system failures.
- Physical Security: the sensor devices and microcontroller can be compromised. So, we need to ensure that physical security measures are in place such as alarms or surveillance systems to protect against tampering and theft.

Planning and Timeline

Week	Task	Notes
1	Discussed possible research topics for the project	
2	Decided research topic	
3	Started researching tools	Submitted project proposal on Moodle
4	Continued with the researching tools	
5	Begin Researching appropriate sensors and microcontrollers	
6	Continued with researching appropriate sensors and microcontrollers. Prototyping	Submitted Literature Review
7	Started searching for IoT platform.	
8	Begin all aspects of project in earnest. Ordered MPU6050 Sensor and Arduino Uno	Submitted practical project plan and performed early-stage presentation
9 (including holidays)	Connected MPU sensor and Arduino Board	
10	Started working on ThingSpeak and twilio for APIs to send Alerts.	
11	- Complete Networking for data. - Display output on the Arduino processing unit.	

	- Encountered an electrical issue with connecting camera to the Arduino uno board. - Got feedback from tutor for camera and focused on motion sensor.	
12	- Programmed necessary coding to the arudino board to make it ready for the demonstration.	Delivered final stage presentation and demonstration.

Conclusion

In conclusion, the implementation of a smart home security system using the MPU6050 sensor and Arduino Uno Rev2 Wi-Fi presents a robust solution for enhancing the security and monitoring capabilities of a home. Throughout the project, several key components were integrated, including the sensor, microcontroller, ThingSpeak for data storage and visualization, and Twilio for SMS alert notifications. The system design focused on ensuring the reliable gaining of sensor data, secure transmission to ThingSpeak, and real-time alerting through Twilio. By leveraging the capabilities of these platforms, users can effectively monitor their home's security status remotely and receive instant notifications in the event of any suspicious activities.

The testing phase played a crucial role in ensuring the functionality, reliability, and security of the system. Through comprehensive unit testing, integration testing, security testing, performance testing, and user acceptance testing, potential issues and vulnerabilities were identified and addressed. This rigorous testing approach connects confidence in the system's ability to perform as intended and deliver a seamless user experience. There are various of risks and issues that we might encountered such as data breaches, social engineering etc, Overall, the implementation of this smart home security system provides users with enhanced peace of mind, remote monitoring capabilities, and immediate alerting mechanisms. By combining hardware components, software integration, and cloud-based platforms, the project demonstrates a successful fusion of technology and security to protect and safeguard the home environment.

Future Expansion

This project offers a solid substance for further improvements and capabilities. Some of them are:

i. Camera integration: Incorporate surveillance cameras can be installed and wired into the system to enable real-time video monitoring and recording. This expansion can provide visual evidence and improve the overall security coverage of the home.

ii. Voice control Integration: Integrate voice assistants such as Amazon Alexa or Google Assistant to enable voice control of the security system. This expansion offers hands-free operation and enhances the user's interaction with the system.

iii. Machine Learning and AI: Exploring the integration of machine learning algorithms and artificial intelligence techniques to enhance the system's capabilities that includes advanced irregularity detection, activity pattern recognition, or predictive analysis for proactive security measures.

Bibliography

Johnson, A. 2022, 'Smart Home Security: Protecting Your Home in the Digital Age', Home Security Magazine, 15 January 2022, https://www.homesecuritymagazine.com/smart-home-security/protecting-your-home-in-digital-age (accessed 10 May 2023).

Smith, J. 2020, The Smart Home Revolution: Enhancing Security and Convenience, 2nd edn, ABC Publications, New York.

HowToMechatronics (2019). Arduino and MPU6050 Accelerometer and Gyroscope Tutorial - HowToMechatronics. [online] HowToMechatronics. Available at: https://howtomechatronics.com/tutorials/arduino/arduino-and-mpu6050-accelerometer-and-gyroscope-tutorial/.

circuitdigest.com. (n.d.). *How Does the MPU6050 Accelerometer & Gyroscope Sensor Work and Interfacing It With Arduino*. [online] Available at: https://circuitdigest.com/microcontroller-projects/interfacing-mpu6050-module-with-arduino.

Admin (2021). *Smart Activity Tracker using MPU6050 and Arduino*. [online] IoT Projects Ideas. Available at: https://iotprojectsideas.com/smart-activity-tracker-using-mpu6050-and-arduino/ [Accessed 30 May 2023]

ResearchGate. (n.d.). *(PDF) Design and Implementation of Smart Home Security System*. [online] Available at: https://www.researchgate.net/publication/293173717_Design_and_Implementation_of_Smart_Home_Security_System.

Bhasker, M. (n.d.). *SMART HOME SECURITY SYSTEM USING ARDUINO AND IoT*. [online] Available at: https://ijcrt.org/papers/IJCRT2108220.pdf

K.Satheeshkumar, Ithkumar, N.A., P.A.Gopinath, Ithkumar, S.R. and J.Chandramohan, M. (2018). Implementation of Smart Home Automation and Security System Using Arduino and Wi-Fi through Android Application. *International Journal of Engineering Research & Technology*, [online] 5(13). Available at: https://www.ijert.org/implementation-of-smart-home-automation-and-security-system-using-arduino-and-wi-fi-through-android-application.

Appendix

For Network Connection

```
#include <WiFiNINA.h>
#include <Wire.h>
#include <ThingSpeak.h>
#include <MPU6050.h>

// replace with your network credentials
const char* ssid = "your_SSID";
const char* password = "your_PASSWORD";

// ThingSpeak details
unsigned long channelID = YOUR_CHANNEL_ID;
const char* writeAPIKey = "YOUR_WRITE_API_KEY";
const char* server = "api.thingspeak.com";

// create an instance of the MPU6050 class
MPU6050 mpu;

void setup() {
  Serial.begin(9600);

  // initialize the MPU6050 sensor
  Wire.begin();
```

For data capture

```
#include <Wire.h>
#include <MPU6050.h>

MPU6050 mpu;

void setup() {
  Serial.begin(9600);
  Wire.begin();
  mpu.initialize();
}

void loop() {
  mpu.update();
  float accX = mpu.getAccX();
  float accY = mpu.getAccY();
  float accZ = mpu.getAccZ();

  float accMagnitude = sqrt(pow(accX, 2) + pow(accY, 2) + pow(accZ, 2));

  if (accMagnitude > 5.0) {  // Change this threshold to adjust sensitivity
    Serial.println("Motion detected!");
    delay(5000);  // Delay to avoid repeated detection
  }
}
```

For IoT

```
#include <SPI.h>
#include <Ethernet.h>
#include <ThingSpeak.h>

// Ethernet shield pin configuration
#define W5100_CS_PIN 10
#define SDCARD_CS_PIN 4

// ThingSpeak channel details
unsigned long channelID = YOUR_CHANNEL_ID;       // Replace with your channel
ID
const char *writeAPIKey = "YOUR_WRITE_API_KEY";  // Replace with your Write
API Key

// Sensor data variables
float sensorValue1;
float sensorValue2;

// Initialize the Ethernet client
EthernetClient client;

void setup() {
  // Initialize serial communication
  Serial.begin(9600);

  // Initialize Ethernet shield
  Ethernet.begin(mac);

  // Set the ThingSpeak server
  ThingSpeak.begin(client);
}

void loop() {
  // Read sensor values
  sensorValue1 = analogRead(A0) * 0.0048828125;  // Example: Analog sensor
connected to A0
  sensorValue2 = analogRead(A1) * 0.0048828125;  // Example: Analog sensor
connected to A1

  // Print sensor values
  Serial.print("Sensor 1: ");
  Serial.println(sensorValue1);
  Serial.print("Sensor 2: ");
  Serial.println(sensorValue2);

  // Update ThingSpeak channel
  ThingSpeak.setField(1, sensorValue1);
  ThingSpeak.setField(2, sensorValue2);
  int httpCode = ThingSpeak.writeFields(channelID, writeAPIKey);

  // Check if update was successful
  if (httpCode == 200) {
    Serial.println("Data sent to ThingSpeak successfully.");
  } else {
    Serial.println("Error sending data to ThingSpeak. HTTP error code: " +
String(httpCode));
  }

  // Wait for a few seconds before sending the next update
  delay(5000);
}
```