

Zachary Hayes

Pratham Vadhulas

Internet of Things

4/4/2024

Project Proposal

IoT System:

The idea our team wishes to implement is a home security system. The goal of the smart system is to be able to detect and deter intruders from entering a room. This will be possible with the Arduino Uno R4 as there are multiple sensors and actuators on the market that can assist in the creation of this system. Home security systems can be quite expensive and intrusive, so by designing our own, it becomes truly secure. Using the ESP32 onboard wifi chip, we will be able to monitor various things via our phone or web interface. The home operator will be able to disable the system via the interfaces as well. All of this will be demonstrated via a detailed cardboard room illustrating each feature.

Sensors/Actuators:

There are many sensors and actuators that can be utilized for this project. The team has come up with multiple different ideas for security measures. One involves the ***Grove PIR Motion Sensor***, which will be utilized to detect motion in the room and lead to the snapping of the ***Arducam*** to have eyes on the intruder. In addition to this, ***Ultrasonic Motion Sensors*** can be installed in the windows of the room to detect if the windows have been opened. Next, a ***Servo Motor*** could be installed to activate an aerosol can and release a form of deterrent on the intruder.

Lastly, an ***RFID reader*** can be installed next to the entrance of the room. This will allow the owner of the room to disable the system via scanning an RFID transponder.

Use Case:

There are a multitude of use cases for designing your own home security system. This is simply a demonstration of what is possible. The best reason to design security systems like such is that not even the security companies know how your system works. This protects you from nearly all threats. It also is much cheaper than a monthly subscription to the common services. Another use case for this would be to develop the system within a vehicle, as the ideas and practices transcend just home security.

Application/Protocol:

Web Interface

1. Live Data Display: Show real-time sensor data, and system logs.
2. System Control: Enable users to arm/disarm the system, adjust settings, and manage alerts.
3. User Management: (If time permits) Allow for user account creation and management.

Development Tools

1. Frontend: HTML, CSS, and JavaScript for creating the user interface.
2. Backend: Consider a lightweight backend framework if we need server-side logic for data storage or user authentication.

Communication Protocol

1. MQTT (Message Queuing Telemetry Transport):

- Advantages: It's lightweight and perfect for small devices like the Arduino Uno WiFi, plus it supports real-time communication.
- Security Measures: we plan to secure it with SSL/TLS for encrypted communication, enhancing data privacy and security.

2. HTTPS:

- While it offers secure HTTP communication through REST APIs, it's more resource-intensive. We will need to consider the performance impact on the Arduino Uno WiFi.

3. CoAP:

- This will be our last resort option for communicating with the Arduino as we aim to learn a new protocol.

References:

[Servo](#)

[RFID](#)

[Motion Sensor](#)

[Arducam](#)

[Distance Sensor](#)