Smart Security for Museum Display Cases

Presentation by:

Yarden Horowitz
Eden Sheiko

Introduction to the problem:

- Museums face the challenge of protecting valuable artifacts and artworks within display cases from theft or damage.
- Existing security systems often lack real-time monitoring and are unable to provide comprehensive data for analysis.

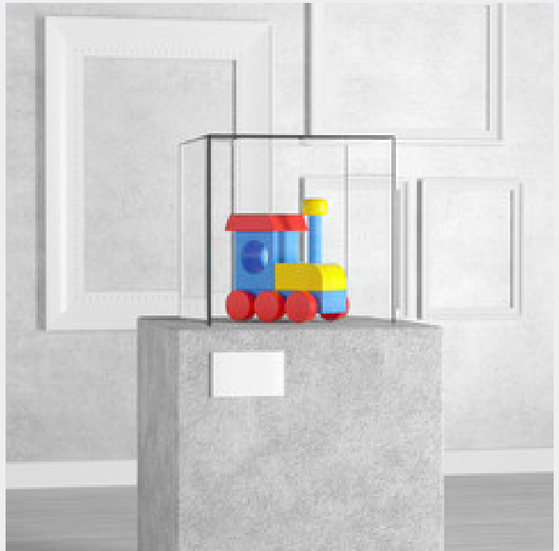
Motivation behind the project:

- Develop a cost-effective and efficient solution to enhance the security of museum display cases. Implement an IoT-based system that allows real-time monitoring and provides actionable insights for preventive measures.

- Protecting valuable artifacts in museum display cases is crucial, and our IoT Arduino-embedded project aims to provide a secure solution with sensors.

Protecting valuable artifacts in museum display cases is crucial, and our IoT Arduino-embedded project aims to provide a secure solution with sensors.

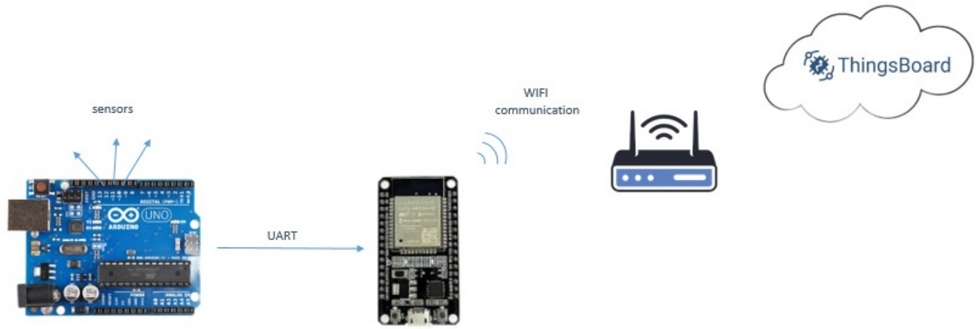
Our system allows real-time monitoring of display case conditions through a web-based dashboard.



System Design

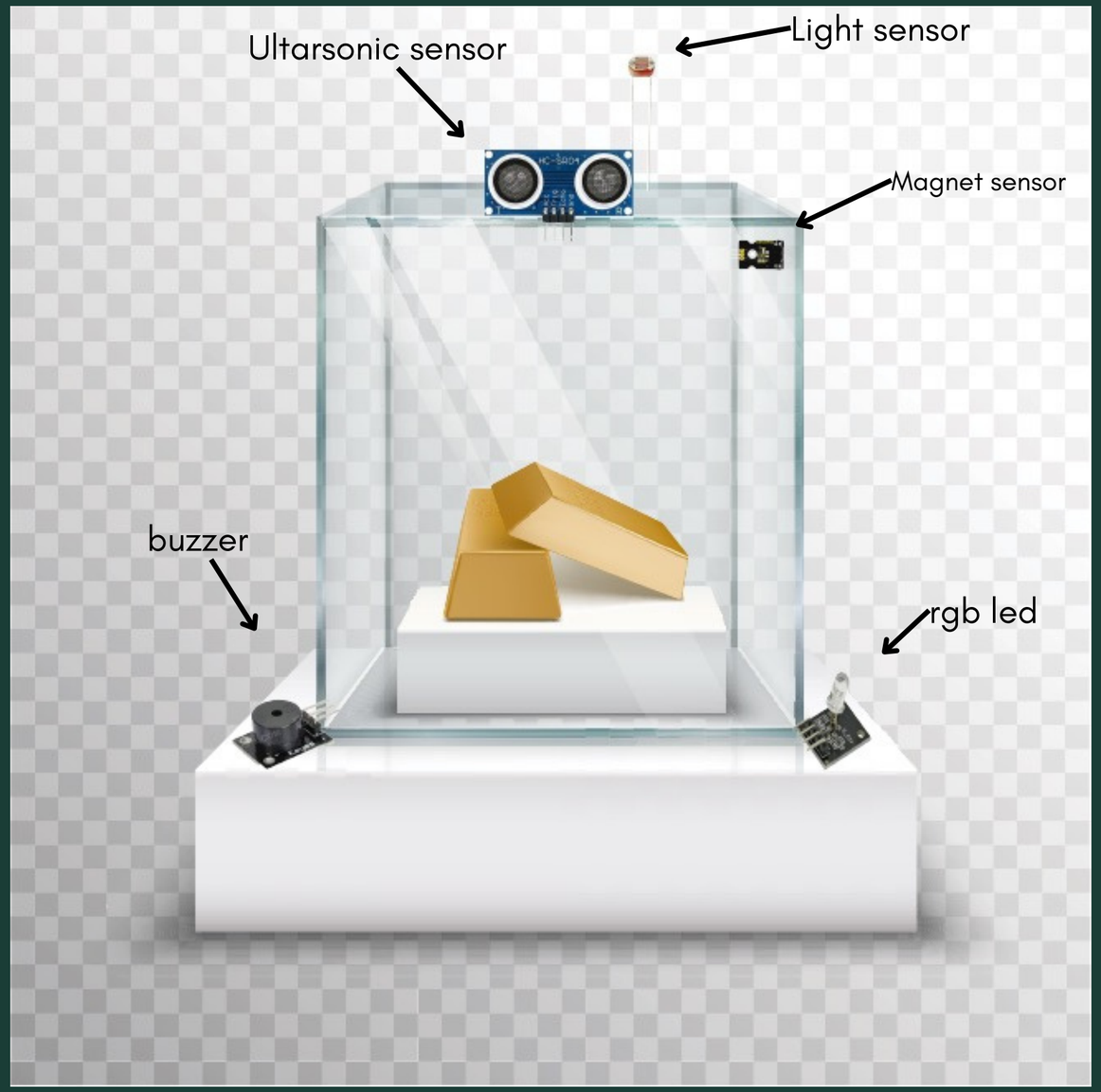
System Architecture:

connecting the Arduino to the sensors, actuators, and to the ESP32 Gateway device to establish a connection between the local Wi-Fi network and the ThingsBoard cloud platform.



Hardware Components:

Core componnent: Arduino Uno microcontroller
Sensors: Ultrasonic sensor,Magnetic sensor, Light sensor.
Actuators: buzzer sensor alarm system, LED indicators.
Gateway device: ESP32 with network connectivity.


Ultarsonic sensor
Light sensor
Magnet sensor
buzzer
rgb led

Implementation Details

we connected the Magnet, Ultrasonic, and light sensors to the Arduino Uno.
the idea is to have stages of deterrence-
the Ultrasonic is measuring the distance of the people who are standing close to the display case and shows different color according to the distance:
distance > 100 - green
20 < distance < 100 - blue
distance < 20 - red

when the distance is on red state - the buzzer starts working

Implementation Details

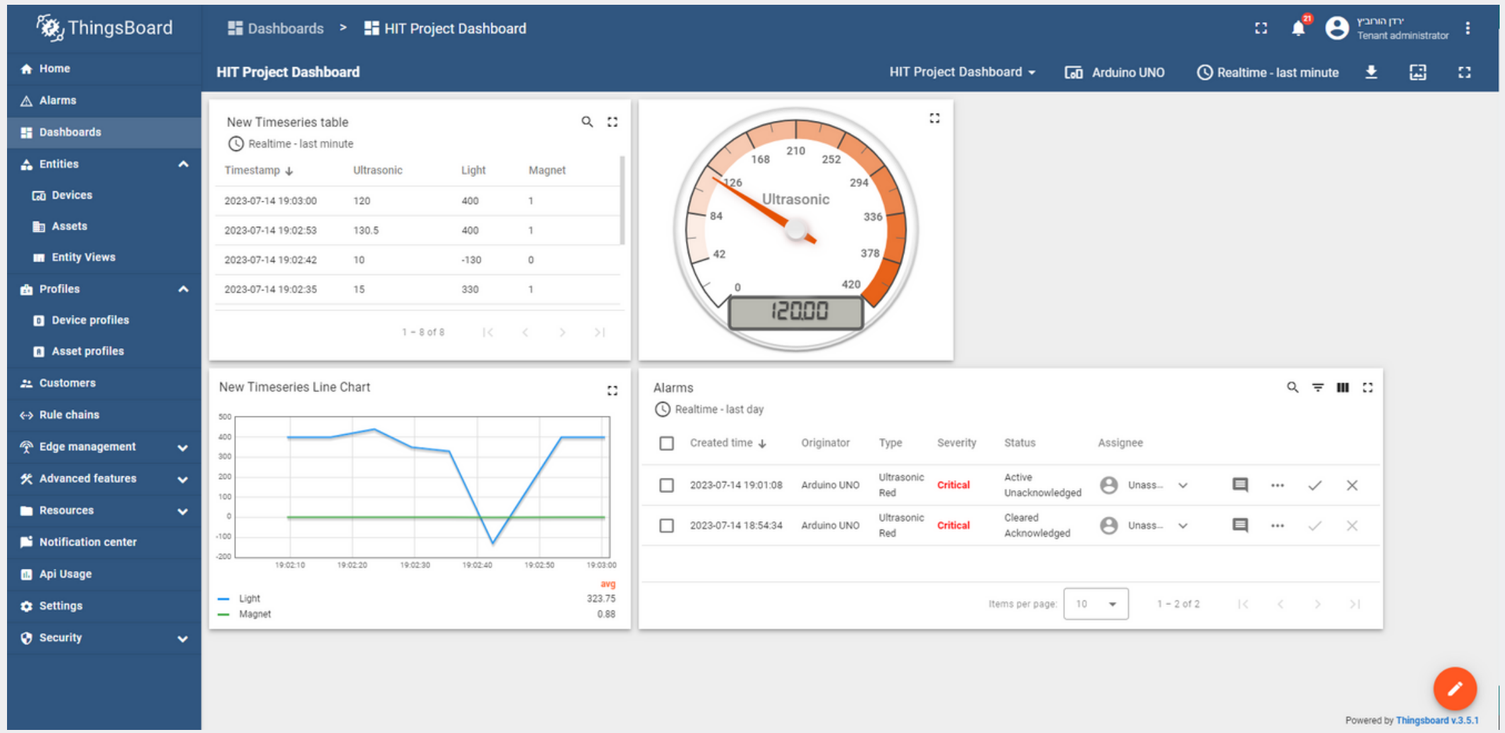
We also use the light sensor as an additional identifier, when it is on dark mode we will check between 2 options-
1. Turn off the light in the museum - the museum is closed and the ultrasonic is on green mode
2. The ultrasonic is on blue/red mode, and in combination with the light sensor it is possible to understand that there is a threat in the environment

And the most critical sign - is the disconnection of the magnet inside the box.
When the magnet disconnects, it goes from position 1 to position 0 - and this is a sign that the thief has opened the display case.

In such a situation, an alert is sent on Thingsboard to the security personnel.

Implementation Details

Communication and Cloud Integration-
the Arduino IDE sends the data in UART to the ESP32 gateway, and the ESP is configured to connect with the ThingsBoard cloud platform and transmit sensor data.
in the Thingsboard dashboard, we can see the time-series Line Chart of the sensors, time-series table, alarms, and analog gauge of the Ultrasonic.



ESP32_UART.ino

```
26
27
28  WiFiClient wifiClient;
29  HttpClient httpClient(wifiClient, tbServer, tbPort);
30
31  void setup() {
32    Serial.begin(9600);
33    Serial2.begin(9600, SERIAL_8N1, RXp2, TXp2);
34    connectToWiFi();
35  }
36
37  void loop() {
38      // Check if connected to Wi-Fi, retry connection if necessary
39    if (WiFi.status() != WL_CONNECTED) {
40      connectToWiFi();
41    }
42    Serial.println("Message Received From Arduino By UART: ");
43    Serial.println(Serial2.readString());
44    uartString = Serial2.readString();
45    payload = uartString.c_str();  // Converting String to const char*
46
47    // Send telemetry data to ThingsBoard
48    sendTelemetry(payload);
49
50    delay(5000); // Send data every 5 seconds
51  }
52
53  void connectToWiFi() {···
61  }
62
63  void sendTelemetry(const char* data) {···
82  }
83
```

```
if(distance >= GREEN_DISTANCE) {
  setColor(0, 255, 0); // Green
  noTone(BUZZER_PIN);  // Turn off the buzzer
} else if(distance >= BLUE_DISTANCE && distance < GREEN_DISTANCE) {
  setColor(0, 0, 255); // Blue
  noTone(BUZZER_PIN);  // Turn off the buzzer
} else if(distance < RED_DISTANCE) {
  setColor(255, 0, 0); // Red
  tone(BUZZER_PIN, BUZZER_FREQUENCY); // Turn the buzzer on at 1kHz
}
```

```
// Create a JSON document with capacity for the data
StaticJsonDocument<200> doc;

// Add the sensor data to the document
doc["Ultrasonic Distance"] = distance;
doc["Light Sensor Value"] = lightValue;
doc["Magnetic Sensor Value"] = magneticValue;
doc["KY-018 Light sensor value"] = lightValueKY018;

// Serialize the document as JSON and send it to the Serial
serializeJson(doc, Serial);
Serial.println();
```

Challenges

Challenges faced during the project:

- Sensor calibration and fine-tuning to ensure accurate readings.
- Establishing reliable communication between Arduino and the gateway device.
- Integrating with the ThingsBoard cloud platform and configuring data visualization.

How challenges were addressed:

- Conducting extensive testing and calibration to optimize sensor performance.
- Implementing error handling and retry mechanisms for communication reliability.
- watching video guilds of ThingsBoard integration

conclusion

In conclusion, our IoT Arduino-embedded museum display case security system has successfully enhanced museum artifact security through real-time monitoring, cloud connectivity, and is also used as a sign of deterrence to people around her

We have gained valuable insights into practical considerations and the value of real-time monitoring.

Thank you for taking the time to learn about our innovative museum display case security solution!

Special thanks to:
Vladi Budnitski
Dr. Yakov Damatov

Creators:
Eden Sheiko - edensheiko@gmail.com
Yarden Horowitz - yarden218@gmail.com