ESP32 RFID Access Control System

Arwa BENCHEIKH

October 15, 2025

Project Overview

This project implements a smart access control system using an ESP32 with multiple integrated features: RFID-based user identification, temperature and humidity sensing (DHT11), relay control, NeoPixel visual feedback, MQTT communication, local SPIFFS logging, Google Sheets integration, and a Node-RED dashboard for live monitoring and control. The system uses FreeRTOS tasks to efficiently manage simultaneous operations.

Hardware Components

- ESP32-WROVER-KIT
- MFRC522 RFID Reader
- DHT11 Temperature & Humidity Sensor
- Relay Module (for door lock control)
- NeoPixel LED Strip (8 pixels)
- Active Buzzer

Wiring Connections

- MFRC522:
  - RST: GPIO4
  - SDA: GPIO5
  - MOSI: GPIO23, MISO: GPIO19, SCK: GPIO18
  - VCC: 3.3V, GND: GND
- DHT11: Data pin GPIO14, VCC: 3.3V, GND: GND
- Relay: GPIO15

- Buzzer: GPIO16
- NeoPixel Strip: GPIO17, 5V/3.3V depending on strip spec

Key Features

- Read RFID cards and determine access permissions from a local user list or server.
- Relay control for door lock operation with timed automatic off.
- NeoPixel LEDs provide visual feedback: green for access granted, red for denied.
- Buzzer feedback for access events.
- Read temperature and humidity periodically from DHT11 sensor.
- FreeRTOS tasks for parallel operations: WiFi management, MQTT communication, DHT sampling, RFID scanning, and log syncing.
- MQTT communication for real-time event publishing and remote relay control.
- Local logging to SPIFFS in NDJSON format with log rotation.
- Google Sheets webhook integration to record access events online.
- Node-RED dashboard for visualization, control, and user management.

Software Requirements

- PlatformIO IDE or VSCode with PlatformIO extension
- Arduino framework for ESP32
- Required libraries:

```
miguelbalboa/MFRC522@^1.4.12
adafruit/Adafruit NeoPixel@^1.15.1
adafruit/RTClib@^2.1.4
adafruit/Adafruit BusIO@^1.17.2
knolleary/PubSubClient@^2.8
adafruit/DHT sensor library@^1.4.6
bblanchon/ArduinoJson@^7.4.2
```

PlatformIO Configuration

```
[env:esp-wrover-kit]
platform = espressif32
board = esp-wrover-kit
framework = arduino
lib_deps =
    miguelbalboa/MFRC522@^1.4.12
    adafruit/Adafruit NeoPixel@^1.15.1
    adafruit/RTClib@^2.1.4
    adafruit/Adafruit BusIO@^1.17.2
    knolleary/PubSubClient@^2.8
    adafruit/DHT sensor library@^1.4.6
    bblanchon/ArduinoJson@^7.4.2
upload_port = COM10
monitor_port = COM10
monitor_speed = 115200
```

WiFi and MQTT Management

- WiFi reconnect handled automatically in a dedicated FreeRTOS task.
- NTP time synchronization for accurate timestamps.
- MQTT client reconnects automatically and subscribes to:
  - `arwa/esp32/relay/cmd` – for remote relay commands
  - `arwa/esp32/user/control` – to refresh users or sync logs
- Access events (`arwa/esp32/rfid/access`) and DHT readings (`arwa/esp32/dht11`) are published to MQTT.

SPIFFS Log Handling

- Access events and sensor readings are saved locally in NDJSON format.
- Logs automatically rotate if size exceeds 50 KB.
- Offline events are queued and sent to Google Sheets when WiFi reconnects.

Google Sheets Integration

- Events are sent via POST request with JSON payload:

```
{
    "uid": "<RFID_UID>",
    "name": "<User Name>",
    "status": "<granted/denied>",
    "timestamp": "<YYYY-MM-DD HH:MM:SS>",
    "secret": "mysupersecret123"
}
```

- Webhook URL is configurable in the code.

Node-RED Dashboard

- Provides real-time monitoring of:
  - Access events and RFID logs
  - Relay states
  - DHT11 sensor data
- Allows manual refresh of user list and syncing of offline logs.
- Example HTTP request node to fetch users from server:

```
[
    {
        "id": "8886a8a76d9ced34",
        "type": "http request",
        "z": "2a331b5cdcea25a9",
        "name": "Fetch Users from Server",
        "method": "GET",
        "ret": "obj",
        "url": "https://script.google.com/macros/s/AKfycbzZ_iIvN-
           w7AClYiusBbfyLnC4jUHnRSNBfnrTYwBKEzmH5kkTedtRxDsKqgEy0H5qM
           /exec",
        "wires": [["1364f00b75114339"]]
    }
]
```

FreeRTOS Tasks

- **WiFiTask** – handles connection, reconnection, and NTP sync.
- **MQTTTask** – maintains MQTT connection and publishes queued events.
- **DHTTask** – reads temperature and humidity periodically, queues events.

- **RFIDTask** – scans RFID cards, checks permissions, triggers relay and feedback.
- **SyncTask** – periodically fetches users from server and attempts to upload unsent logs.

Setup Instructions

1. Install PlatformIO IDE or VSCode with PlatformIO extension.
2. Connect ESP32 to your PC via USB.
3. Open the project folder in PlatformIO.
4. Ensure the correct COM port is set in `platformio.ini`.
5. Click `Build` and then `Upload` to flash the ESP32.
6. Open the Serial Monitor to view debug messages.
7. Access Node-RED dashboard for live monitoring.

Usage

1. Power on the ESP32.
2. Scan RFID tags to test access control.
3. Observe LED and buzzer feedback.
4. View sensor readings and events on Node-RED.
5. Send commands via MQTT or dashboard to control relay or refresh users.

License

This project is open-source and free to use for educational and personal purposes.

Author: Ben Cheikh Arwa , Electrical Engineering student student @ ENIT