

RFID Attendance System Using Arduino Uno and ESP8266

Project Overview

The **RFID Attendance System** helps automate attendance marking by using **RFID cards**. The system reads the unique **RFID tag ID** from student cards, and upon a successful match, it logs the attendance into a **cloud database** (e.g., Firebase, ThingSpeak, or Google Sheets). This system can be connected to an **Arduino Uno** for controlling the RFID reader, while the **ESP8266** provides Wi-Fi connectivity for sending data remotely.

Objectives

- ✓ **Scan RFID cards** to mark attendance.
- ✓ **Send attendance data** to a cloud platform (Firebase or Google Sheets).
- ✓ **Store attendance records** and monitor them remotely.
- ✓ **Display student details** (name, ID) when attendance is marked.
- ✓ **Real-time attendance tracking** via a web interface.

Components Required

1. **Arduino Uno** – Main controller for the RFID reader.
2. **ESP8266 Wi-Fi Module** – Provides Wi-Fi connectivity.
3. **RC522 RFID Module** – Reads RFID tags/cards.
4. **RFID Tags/Cards** – Unique identifiers for students.
5. **LCD Display (16x2 or 20x4)** – Displays attendance information.
6. **Jumper Wires & Breadboard** – For connections.
7. **5V Power Supply** – Powers the Arduino and peripherals.
8. **Cloud Platform (Firebase, Google Sheets)** – Stores and displays attendance records.

How the System Works

1. The **Arduino Uno** interfaces with the **RFID reader (RC522)** to scan the RFID tag.
2. The **RFID tag ID** is matched with a pre-programmed database to identify the student.
3. Upon successful match, the system **marks attendance** and sends the data to a **cloud database** via the **ESP8266 Wi-Fi module**.
4. **Student details** (name, ID) are displayed on the **LCD** after attendance is marked.
5. The **attendance records** are stored remotely and can be accessed through a **web interface**.

Circuit Diagram

RC522 RFID Module to Arduino Uno

RC522 Pin Arduino Uno Pin

SDA	D10
SCK	D13
MOSI	D11
MISO	D12
IRQ	Not connected
GND	GND
RST	D9
3.3V	3.3V

ESP8266 to Arduino Uno (Serial Communication)

ESP8266 Pin Arduino Uno Pin

RX	TX
TX	RX
GND	GND
3.3V	3.3V

LCD Display to Arduino Uno

LCD Pin Arduino Uno Pin

VSS	GND
VDD	5V
VO	Potentiometer (optional for contrast)
RS	D7
RW	GND
E	D6
D4-D7	D4, D5, D6, D7

Arduino Code for RFID Attendance System

This code reads the RFID tag and sends attendance data to **Firestore**.

```

#include <SPI.h>
#include <MFRC522.h>
#include <ESP8266WiFi.h>
#include <FirebaseESP8266.h>

// Wi-Fi Credentials
const char* ssid = "Your_WiFi_Name";
const char* password = "Your_WiFi_Password";

// Firebase Credentials
#define FIREBASE_HOST "Your_Firebase_Host"
#define FIREBASE_AUTH "Your_Firebase_Auth_Token"

// RFID Setup
#define RST_PIN 9
#define SS_PIN 10
MFRC522 mfrc522(SS_PIN, RST_PIN);

// Firebase setup
FirebaseData firebaseData;

void setup() {
  Serial.begin(9600);
  SPI.begin(); // Initiate SPI bus
  mfrc522.PCD_Init(); // Initialize the RFID reader
  WiFi.begin(ssid, password); // Connect to Wi-Fi

  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }
  Serial.println("Connected to Wi-Fi!");

  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH); // Connect to Firebase
}

void loop() {
  if (mfrc522.PICC_IsNewCardPresent()) {
    if (mfrc522.PICC_ReadCardSerial()) {
      String rfidTag = "";
      for (byte i = 0; i < mfrc522.uid.size; i++) {
        rfidTag += String(mfrc522.uid.uidByte[i], HEX);
      }

      Serial.print("RFID Tag: ");
      Serial.println(rfidTag);

      // Send attendance data to Firebase
      String path = "/attendance/" + rfidTag;
      Firebase.setString(firebaseData, path, "Present");

      // Check for errors in writing to Firebase
      if (firebaseData.failed()) {
        Serial.println("Firebase failed: " + firebaseData.errorReason());
      } else {
        Serial.println("Attendance recorded for RFID: " + rfidTag);
      }
    }
  }
}

```

```
        delay(2000); // Wait for a moment before reading again
    }
}
```

How to Use the RFID Attendance System

1. **Upload the code** to Arduino Uno using the Arduino IDE.
 2. **Connect the system** to Wi-Fi and Firebase.
 3. **Scan the RFID card** by placing it near the RFID reader.
 4. **Check the Firebase database** for attendance updates.
 5. **View the data** in real-time through Firebase or Google Sheets.
 6. The LCD will **display** the attendance status when RFID is scanned.
-

Features & Benefits

- ✓ **Automated attendance** – Quickly marks students' attendance via RFID cards.
 - ✓ **Remote monitoring** – View and track attendance records online.
 - ✓ **Real-time updates** – Data is updated instantly in Firebase.
 - ✓ **No manual input** – Eliminate manual attendance marking.
 - ✓ **Affordable & scalable** – Easy to set up and expand to more students.
-

Future Enhancements

- Mobile App Integration** – Create an app for real-time attendance monitoring.
- Multiple RFID card support** – Use multiple RFID cards per student for flexibility.
- Time-based attendance** – Record the exact time a student checks in.
- Security features** – Add fingerprint or facial recognition for added security.