Embedded Systems

RFID ATTENDANCE SYSTEM

Harsha Vardhan V M - 20Z217

Kabilan K K - 20Z223

Prednya Ramesh - 20Z239

Shri Shivathmika V - 20Z251

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE ENGINEERING

of Anna University

March 2023

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

PSG COLLEGE OF TECHNOLOGY

(Autonomous Institution)

COIMBATORE – 641 004

Introduction:

RFID technology is a wireless communication method that uses radio frequency signals to identify and track objects.

In an RFID attendance system, each student is given an RFID tag, which is a small device that contains a unique identification code. The RFID tag can be carried by the student, or it can be embedded in a student ID card or other item that the student carries.

To take attendance, the RFID reader is used to scan the RFID tags of the students as they enter the classroom. The RFID reader sends the identification codes of the RFID tags to the computer, which then records the attendance of the corresponding students.

Problem Statement:

Traditional attendance systems typically rely on methods such as sign-in sheets when employees arrive and leave work. These methods can be prone to errors and can be easily falsified. Additionally, they often require manual data entry and analysis, which can be time-consuming and prone to errors.

There are several other problems associated with traditional attendance systems such as lack of mobility and flexibility, failure to accurately capture information, lack of security and data privacy and limited integration with other systems.

Due to these challenges, many organizations have started to move towards digital and automated attendance systems, which can provide more accurate and secure tracking of employee attendance and one of them is the RFID attendance system. The others are biometric fingerprints or even the NFS attendance system.

Purpose:

An RFID-based attendance system makes use of RFID technology to mark student's attendance before every lecture in the institute. The students have to use an RFID tag to mark their presence on the RFID reader. The data is stored in the attendance system with high accuracy and efficiency.

However, managing the daily attendance of thousands of students is quite a challenging task for teachers. Traditional means of registering daily students' attendance may result in errors & tremendous manual work.

An RFID based attendance system can be a great solution to overcome such challenges as it automates the students' attendance process & enables teachers and parents to track & monitor students' activities effortlessly.

Besides student attendance management, the RFID attendance system can also be used to track the attendance of faculty or staff to simplify the payroll management procedure.

Product Features:

RFID attendance systems use radio frequency identification technology to track and record the attendance of employees or students. Some of the product features of RFID attendance systems include:

- RFID reader: The system includes an RFID reader that can read the information stored on RFID tags or cards.
- RFID tags/cards: The system uses RFID tags or cards that contain unique identification information for each user.
- Attendance tracking: The system tracks the attendance of users by reading the RFID tags or cards and recording the time and date of entry and exit.
- Real-time data: The system provides real-time data on attendance, which can be viewed by authorized personnel.

Hardware Design:

Implementation Details:

We've developed an attendance system with Arduino nano,NodeMCU,RFID Scanner using MQTT Broker.The RFID scanner scans the RFID tag and sends the data to a cloud platform called adafruit.io through the WiFi module.

1. The RFID scanner is connected to the Arduino Nano and NodeMCU microcontroller.
2. The RFID scanner library and the necessary code are uploaded to the microcontroller to enable it to read the unique identification number on the RFID tag and store it in its memory.
3. The NodeMCU is configured to connect to a Wi-Fi network and Adafruit.io using appropriate libraries. A feed is set up on Adafruit.io to receive attendance data from the NodeMCU.
4. When an individual enters the room or building, the RFID tag is scanned using the RFID scanner, and the unique identification number on the tag is read by the scanner and sent to the microcontroller. The ID number is received by the microcontroller and stored in its memory. The attendance data is sent to the NodeMCU by the microcontroller and then forwarded to Adafruit.io.
5. The attendance data is received by Adafruit.io and stored in the appropriate feed. The attendance data can be analyzed and visualized using Adafruit.io's dashboard or exported to other data analysis tools. The process is repeated for every individual entering and leaving the room or building.
6. At the end of the day or session, the attendance data can be exported from Adafruit.io or analyzed in real-time to generate reports or calculate total attendance

The Arduino Nano, NodeMCU board, and MFRC522 RFID are commonly used components in an RFID-based attendance system. Here's how each component works and its purpose in the attendance system:

1. **Arduino Nano:** The Arduino Nano is a microcontroller board based on the ATmega328P chip. It is a compact and cost-effective board that is ideal for controlling the RFID scanner and managing the attendance data. The purpose of the Arduino Nano in an attendance system is to interface with the RFID scanner and collect attendance data. It can also send the collected data to other devices for storage and analysis.
2. **NodeMCU Board:** The NodeMCU is an open-source development board that is based on the ESP8266 Wi-Fi chip. It allows for easy integration with Wi-Fi networks and IoT platforms such as Adafruit.io. The purpose of the NodeMCU board in an attendance system is to connect the Arduino Nano to a Wi-Fi network and send attendance data to Adafruit.io. It enables remote access to attendance data and provides real-time analysis and visualization of the data.
3. **MFRC522 RFID Scanner:** The MFRC522 RFID scanner is a low-cost RFID reader that operates at 13.56 MHz. It can read and write to RFID tags and cards, and it can detect the unique identification number of each tag. The purpose of the MFRC522 RFID scanner in an attendance system is to scan the RFID tags of individuals entering or leaving the room or building. It sends the scanned ID number to the Arduino Nano for processing and storage.

Together, the Arduino Nano, NodeMCU board, and MFRC522 RFID scanner form a complete attendance tracking system. The MFRC522 RFID scanner reads the unique ID number of each RFID tag, and the Arduino Nano collects and processes the data.The NodeMCU board is connected to the Arduino Nano to provide connectivity to the internet or a local network. The Arduino code can be programmed to send attendance data to a web server or a database using the NodeMCU board. The attendance data can then be accessed by teachers or employers to track the attendance of their students or employees.

Overall, the Arduino Nano, NodeMCU board, and MFRC522 RFID reader are used together to create an attendance system that can be used in schools, colleges, or businesses.

Coding:

1. Initialize users and storing in firebase database:

```
#include <ESP8266WiFi.h>
#include <SPI.h>
#include <RFID.h>
#include "FirebaseESP8266.h"  // Install Firebase ESP8266 library

#define FIREBASE_HOST "test2-55006-default-rtdb.firebaseio.com"
//Without http:// or https:// schemes
#define FIREBASE_AUTH "TyzEBsqLLVs4gCzx0iE1GBHYFooThdw90EJYjpbe"
RFID rfid(D8, D0);       //D8:pin of tag reader SDA. D0:pin of tag
reader RST
unsigned char str[MAX_LEN]; //MAX_LEN is 16: size of the array

const char ssid[] = "Harsha's iPhone";
const char pass[] = "12345678";

String uidPath= "/";
//Define FirebaseESP8266 data object
FirebaseData firebaseData;

void connect() {
  Serial.print("checking wifi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
    delay(1000);
  }

  Serial.println("\n connected!");
}

void setup()
{

  Serial.begin(9600);
  WiFi.begin(ssid, pass);
```

```
  SPI.begin();
  rfid.init();
  connect();
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
}
void pushUser (String temp)    //Function to check if an identified
tag is registered to allow access
{
   Serial.println("PUSHING USER ID: "+temp);

    Firebase.setInt(firebaseData, uidPath+"users/"+temp,0);
}
void loop() {
  if (rfid.findCard(PICC_REQIDL, str) == MI_OK)   //Wait for a tag to
be placed near the reader
  {
    Serial.println("Card found");
    String temp = "";                             //Temporary
variable to store the read RFID number
    if (rfid.anticoll(str) == MI_OK)              //Anti-collision
detection, read tag serial number
    {
      Serial.print("The card's ID number is : ");
      for (int i = 0; i < 4; i++)                 //Record and
display the tag serial number
      {
        temp = temp + (0x0F & (str[i] >> 4));
        temp = temp + (0x0F & str[i]);
      }
      Serial.println (temp);
      pushUser (temp);     //Check if the identified tag is an
allowed to open tag
    }
    rfid.selectTag(str); //Lock card to prevent a redundant read,
removing the line will make the sketch read cards continually
  }
  rfid.halt();
}
```

2. Fetching from firebase database and displaying the data in the application:

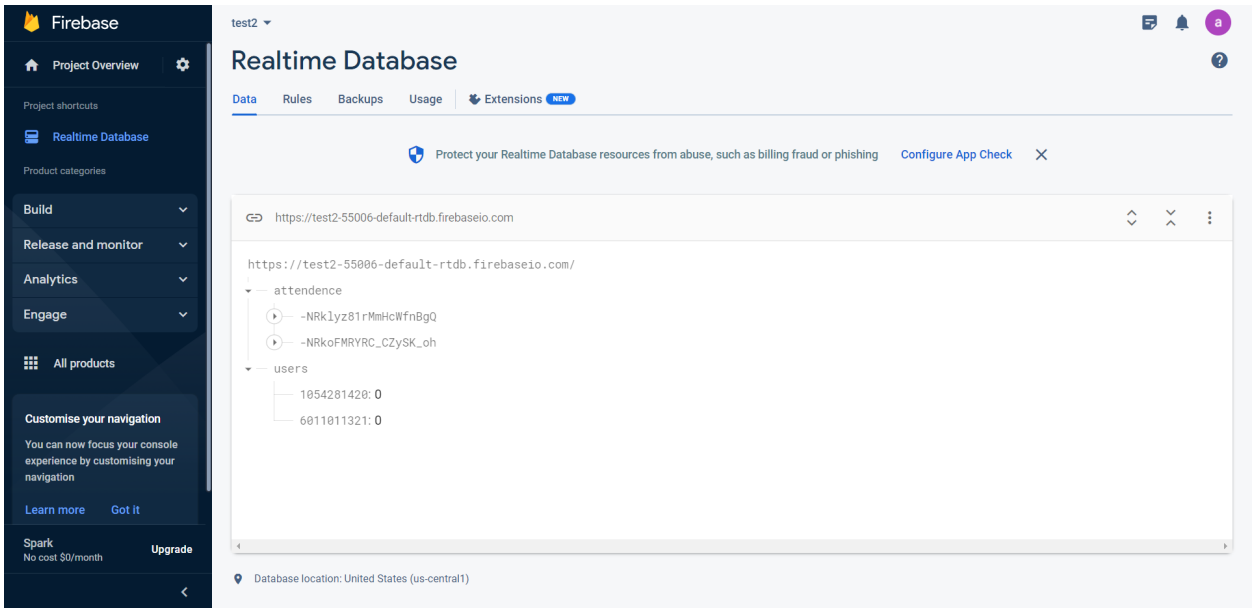
```
#include <ESP8266WiFi.h>
#include <Wire.h>
#include <SPI.h>
#include <RFID.h>
#include "FirebaseESP8266.h"  // Install Firebase ESP8266 library
#include <NTPClient.h>
#include <WiFiUdp.h>

#define FIREBASE_HOST "test2-55006-default-rtdb.firebaseio.com"
//Without http:// or https:// schemes
#define FIREBASE_AUTH "TyzEBsqLLVs4gCzx0iE1GBHYFooThdw90EJYjpbe"
RFID rfid(D8, D0);       //D10:pin of tag reader SDA. D9:pin of tag
reader RST
unsigned char str[MAX_LEN]; //MAX_LEN is 16: size of the array

WiFiUDP ntpUDP;
const long utcOffsetInSeconds = 19800; //(UTC+5:30)
NTPClient timeClient(ntpUDP, "pool.ntp.org");

const char ssid[] = "Harsha's iPhone";
const char pass[] = "12345678";

String uidPath= "/";
FirebaseJson json;
//Define FirebaseESP8266 data object
FirebaseData firebaseData;

unsigned long lastMillis = 0;
// const int red = D4;
// const int green = D3;
String alertMsg;
String device_id="device11";
boolean checkIn = true;

void connect() {
  Serial.print("checking wifi...");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
```

```
    delay(1000);
  }

  Serial.println("\n connected!");
}

void setup()
{

  Serial.begin(9600);
  WiFi.begin(ssid, pass);

  SPI.begin();
  rfid.init();

  timeClient.begin();
  timeClient.setTimeOffset(utcOffsetInSeconds);
  connect();
  Firebase.begin(FIREBASE_HOST, FIREBASE_AUTH);
  Firebase.reconnectWiFi(true);
}
void checkAccess (String temp)    //Function to check if an
identified tag is registered to allow access
{

    if(Firebase.getInt(firebaseData, uidPath+"/users/"+temp)){

      if (firebaseData.intData() == 0)         //If
firebaseData.intData() == checkIn
      {
          delay(1000);

          json.add("time",String(String(timeClient.getDay())+"
"+String(timeClient.getFormattedTime())));
          json.add("id", device_id);
          json.add("uid", temp);
          json.add("status",1);

          Firebase.setInt(firebaseData, uidPath+"/users/"+temp,1);
```

```
          if (Firebase.pushJSON(firebaseData, uidPath+ "/attendence",
json)) {
            Serial.println(firebaseData.dataPath() +
firebaseData.pushName());
          } else {
            Serial.println(firebaseData.errorReason());
          }
      }
      else if (firebaseData.intData() == 1)   //If the lock is open
then close it
      {
          delay(1000);

          Firebase.setInt(firebaseData, uidPath+"/users/"+temp,0);

          json.add("time",String(String(timeClient.getDay())+"
"+String(timeClient.getFormattedTime())));
          json.add("id", device_id);
          json.add("uid", temp);
          json.add("status",0);
          if (Firebase.pushJSON(firebaseData, uidPath+ "/attendence",
json)) {
            Serial.println(firebaseData.dataPath() +
firebaseData.pushName());
          } else {
            Serial.println(firebaseData.errorReason());
          }
      }

    }
    else
    {
      Serial.println("FAILED");
      Serial.println("REASON: " + firebaseData.errorReason());
    }
}
void loop() {
  timeClient.update();
```

```
  if (rfid.findCard(PICC_REQIDL, str) == MI_OK)   //Wait for a tag to
be placed near the reader
  {
    Serial.println("Card found");
    String temp = "";                             //Temporary
variable to store the read RFID number
    if (rfid.anticoll(str) == MI_OK)              //Anti-collision
detection, read tag serial number
    {
      Serial.print("The card's ID number is : ");
      for (int i = 0; i < 4; i++)                 //Record and
display the tag serial number
      {
        temp = temp + (0x0F & (str[i] >> 4));
        temp = temp + (0x0F & str[i]);
      }
      Serial.println (temp);
      checkAccess (temp);     //Check if the identified tag is an
allowed to open tag
    }
    rfid.selectTag(str); //Lock card to prevent a redundant read,
removing the line will make the sketch read cards continually
  }
  rfid.halt();

  delay(500);
}
```

Output:


Firebase
Project Overview
Project shortcuts
Realtime Database
Product categories
Build
Release and monitor
Analytics
Engage
All products
Customise your navigation
You can now focus your console experience by customising your navigation
Learn more
Got it
Spark
No cost $0/month
Upgrade
test2
Realtime Database
Data
Rules
Backups
Usage
Extensions
NEW
Protect your Realtime Database resources from abuse, such as billing fraud or phishing
Configure App Check
https://test2-55006-default-rtdb.firebaseio.com
https://test2-55006-default-rtdb.firebaseio.com/
attendence
-NRklyz81rMmHcWfnBgQ
-NRkoFMRYRC_CZySK_oh
users
1054281420: 0
6011011321: 0
Database location: United States (us-central1)

References:

https://www.researchgate.net/publication/348163330_STUDENT_ATTENDANCE_USING_RFID_SYSTEM

https://ieeexplore.ieee.org/document/5356360

https://www.javatpoint.com/embedded-system-project-rfid-based-attendance-system

Conclusion:

In conclusion, the RFID-based smart attendance system using NodeMCU and Firebase is a great example of how IoT and cloud computing technologies can be used to automate traditional manual processes. By using RFID technology, the system eliminates the need for manual attendance marking, thereby saving time and reducing errors.

The system works by using an RFID reader module connected to a NodeMCU board, which reads the unique ID number of an RFID tag or card. This information is then sent to a Firebase real-time database using WiFi connectivity. The data can be viewed in real-time on the Firebase console, making it easy to track attendance and generate reports.

Overall, this system is an excellent example of how IoT and cloud computing technologies can be used to streamline manual processes, improve efficiency, and reduce errors. It can be used in schools, colleges, and offices to mark attendance automatically and accurately, thereby reducing the workload of teachers, professors, and administrators.