International Journal for Modern Trends in Science and Technology, 8(02): 122-126, 2022
Copyright © 2022 International Journal for Modern Trends in Science and Technology
ISSN: 2455-3778 online
DOI: https://doi.org/10.46501/IJMTST0802020
Available online at: http://www.ijmtst.com/vol8issue02.html

IoT based Digital Attendance System using RFID & ESP32

Tushar Singh[1*] | Dr. Aasha Chauhan[1] | Megha Dewan[1] | Dr. Alok Agarwal[1]

[1]Department of ECE, Aravali College of Engineering and Management, Neharpar Faridabad, Faridabad, India.
[*]Corresponding Author: aashachauhan07@gmail.com

To Cite this Article
Tushar Singh, Dr. Aasha Chauhan, Megha Dewan and Dr. Alok Agarwal. IoT based Digital Attendance System using RFID & ESP32. International Journal for Modern Trends in Science and Technology 2022, 8 pp. 122-126. https://doi.org/10.46501/IJMTST0802020

Article Info
Received: 15 January 2022; Accepted: 05 February 2022; Published: 12 February 2022.

ABSTRACT

IoT based RFID Attendance System is designed and elaborated in this paper. Earlier attendance in any organization like schools, colleges, private offices were written and recorded on registers. A person was required for recording and maintaining that attendance. However, that was a nice fashion as written proof was required those days. But there were many disadvantages with that process. That's why digitization of attendance came into picture. After that problem of wastage of paper get solved. All data was stored in digital format in some computers. But still some problem was there in this scheme also. Those problems were like if that computer having that attendance data malfunctions, then there is a high chance of having data loss. Thus Internet of Things (IoT) came in picture. Now, attendance is directly recorded on the internet. So, this reduces the chance of data loss in any case. Microprocessor ESP32 module is used to design this attendance system. Programming for ESP32 module is written in Embedded C language using an IDE called Arduino IDE. We have used some major libraries such as RFID 522 library, AdafruitMQTT library, Wi-Fi library etc.

KEYWORDS: Embedded C language, ESP32 module, Internet of Things (IoT), IDE, RFID

1. INTRODUCTION

In IoT based RFID Attendance System, there are three major parts. First one is Internet of Things (IoT), a way to connect our devices to the world. IoT is a cloud based system which allows internet to work according to us. IoT is usually implemented for embedded systems to connect it with the virtual world. Second major part of the microprocessor ESP32 module. This module acts as a brain of this system. This module has various features like Bluetooth support, WiFi support, etc, which allows it to connect with the internet. It has different GPIO pins to connect with the other modules. We just have to program it according to our need. The third major part is RFID module. This module detects the RFID tag on the cards. It is close-range device which works on tapping the card on it.

2. Working Operation

RFID Attendance System works on the principle of radio-frequency detection. Whenever a card is placed near the reader module. Then, the reader module sends Radio waves to the card. The tag placed inside the card responds on this signal and reflects back the information stored in it. There are two types of tags active and passive tags. Active tags can send Radio waves to the reader whenever they come near the reader. They have their own power source embedded with it to generate Radio waves. Whereas passive tags don't have any power source to generate Radio waves. They respond only when the reader sends Radio waves to them, i.e., they reflect back the Radio waves came from the reader with information stored in them. Generally, for small purposes, passive tags are more popular. After reading the information from the tag, the RFID reader sends that information to ESP32 module. Then, ESP32 module changes that information in the readable format and does different comparisons on it to identify the owner of the card. After this process, ESP32 module sends that user's roll number to a specific website via internet. For this purpose, we chose a protocol called MQTT (MQ Telemetry Transport). This website is where we store the attendance records. This whole process takes few microseconds to be done. But we add some time delay of 1 or 2 seconds to avoid data clash.

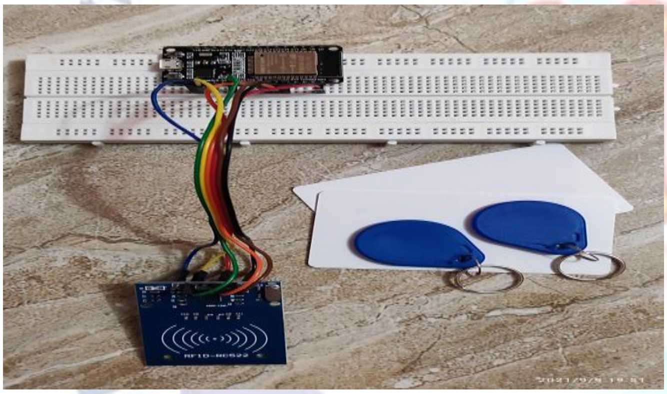
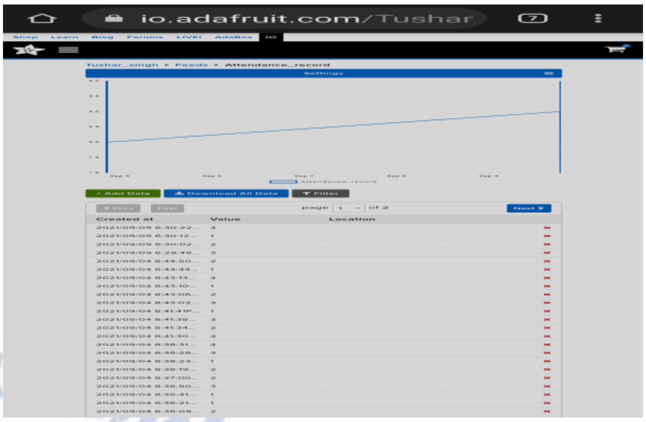


Figure 1: IoT based RFID Attendance System Circuit

3. Monitoring System

In this system, the RFID tags enable the school/college management people to supervise the student movement in and out of the campus. When RFID tags pass through the RFID reader in read range zone, then system will record the data from the RFID tags to the database systems. Laziness on the part of students, nonchalance to school work, extra social activities that have no importance in aiding the objectives of the institution and a lot more, may prevent students from attending lectures. Sequel to these, lecturers and administrators in most developing countries have had to come up with ways to ensure a healthy participation from students, and make sure that the student lecturer interactive relationship is kept intact.



io.adafruit.com/Tushar

Tushar_singh > Feeds > Attendance_record

page 1 of 2

Created at	Value	Location
2021/09/09 6:30:22...	4	
2021/09/09 6:30:12...	1	
2021/09/09 6:30:02...	2	
2021/09/09 6:29:49...	3	
2021/09/04 8:44:50...	2	
2021/09/04 8:44:44...	1	
2021/09/04 8:43:13...	4	
2021/09/04 8:43:10...	1	
2021/09/04 8:43:06...	2	
2021/09/04 8:43:02...	3	
2021/09/04 8:41:41P...	1	
2021/09/04 8:41:38...	3	
2021/09/04 8:41:34...	2	
2021/09/04 8:41:30...	4	
2021/09/04 8:38:31...	4	
2021/09/04 8:38:28...	3	
2021/09/04 8:38:23...	1	
2021/09/04 8:38:19...	2	
2021/09/04 8:37:00...	2	
2021/09/04 8:36:50...	3	
2021/09/04 8:36:41...	1	
2021/09/04 8:36:21...	1	
2021/09/04 8:36:09...	2	

Figure 2: Website for IoT based RFID Attendance System

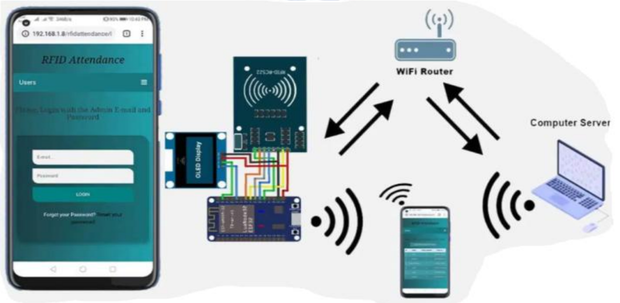


Figure 3: IoT based RFID Attendance Monitoring System using ESP32

4. Attendance Monitoring System

The process of attendance is done by using RFID technology as shown in Figure 4. In this system each student has an RFID Tag to do presence, student put RFID Tag near RFID Reader, then ID result from RFID reader will be sent to microcontroller and compare it with the student data stored in memory, memory serves to store the data of the student's name of the course, if the data ID is a lecture participant then the student's name will be displayed on the LCD Display as well if the student data is not listed it will be informed through LCD Display that the student unregistered, using Wi-Fi module microcontroller can send student attendance data to cloud database by using internet network, data already accommodated in cloud database can be seen in real time by teacher, student and even parent, so that student presences can be monitored from anywhere in real time using Internet of Things (IoT).

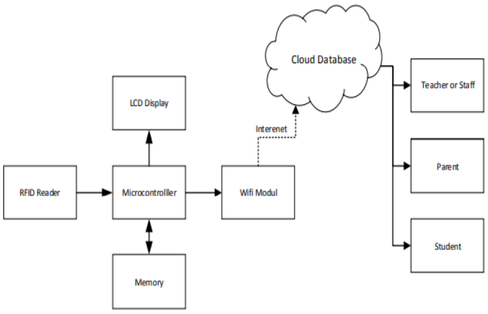


Figure 4: Block diagram of Attendance Monitoring System

5. Flow Chart of RFID System

To display the results of presence data the user must login first, the login process determines the view that can be seen later, the user is divided into two kinds of login as a teacher or administration, and login as a student or parent which is the output display for teachers and administration, where in this view attendance data from students are grouped by subject and class, while in Figure 5 is the output display for students and parents, it only displays data from students who login only, the name of the students displayed depending on the user's students and parents, so that each student and parents can only see their own presence data only.

In RFID systems, an item is tagged with a tiny silicon chip plus an antenna collectively called a tag. The tag can be mobile or stationary and be scanned by stationary or mobile readers respectively, using radio waves. In each scanning case, a reader must scan the tag for the data it contains and then send that information to a database, which interprets the data stored on the tag.

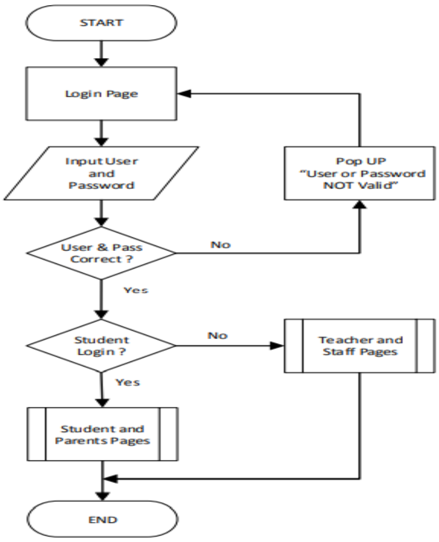


Figure -5 Flowchart System

6. ESP32 Programming

Programming for ESP32 module is written in Embedded C language using an IDE called Arduino IDE. Generally, this IDE provides various libraries to work with different modules like RFID module, etc. In this program, we have used some major libraries such as RFID522 library, Adafruit_MQTT library, Wi-Fi library.

7. Coding

```
#include <MFRC522.h>
#include <SPI.h>
#include <WiFi.h>
#include "Adafruit_MQTT.h"
#include "Adafruit_MQTT_Client.h"
/************************ WiFi Access Point *******************************/
#define WLAN_SSID "Redmi Note 9 Pro Max"
#define WLAN_PASS "nahi_malum"
/************************ Adafruit.io Setup *******************************/
#define AIO_SERVER     "io.adafruit.com"
// Using port 8883 for MQTT
#define AIO_SERVERPORT  1883
// Adafruit IO Account Configuration
// (to obtain these values, visit https://io.adafruit.com and click on Active Key)
#define AIO_USERNAME "Tushar_singh"
#define AIO_KEY "aio_WVih76blhDmsaqoBPqFOJ2iExHWU"
/*********** Global State (you don't need to change this!) *****************/
// WiFiFlientSecure for SSL/TLS support
WiFiClient client;
// Setup the MQTT client class by passing in the WiFi client and MQTT server and login details.
Adafruit_MQTT_Clientmqtt(&client, AIO_SERVER, AIO_SERVERPORT, AIO_USERNAME, AIO_KEY);
//Creating an RFID object to communicate with module
#define SDA_PIN   21
#define RST_PIN   4
MFRC522 rfid(SDA_PIN, RST_PIN);
MFRC522::MIFARE_Key key;
String tag;
intval;
```

```
/****************************** Feeds
***************************************/
// Setup a feed called 'att' for publishing.
// Notice MQTT paths for AIO follow the form:
<username>/feeds/<feedname>
Adafruit_MQTT_Publishatt =
Adafruit_MQTT_Publish(&mqtt, AIO_USERNAME
"/feeds/Attendance_record");
/************************** Sketch Code
************************************/
void setup() {
Serial.begin(115200);
delay(1000);
WiFi.disconnect();
delay(500);
// Connect to WiFi access point.
Serial.println(); Serial.println();
Serial.print("Connecting to ");
Serial.println(WLAN_SSID);
WiFi.begin(WLAN_SSID, WLAN_PASS);
while (WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
Serial.println();
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
SPI.begin();
rfid.PCD_Init();
}
void loop() {
// Ensure the connection to the MQTT server is alive
(this will make the first
// connection and automatically reconnect when
disconnected). See the MQTT_connect
// function definition further below.
MQTT_connect();
if (!rfid.PICC_IsNewCardPresent()) {
return;
}
if (!rfid.PICC_ReadCardSerial()) {
return;
}
Serial.print("Card UID: ");
for (byte i = 0; i<rfid.uid.size; i++) {
tag.concat(String(rfid.uid.uidByte[i] < 0x10 ? "0" : " "));
tag.concat(String(rfid.uid.uidByte[i], HEX));
}
tag.toUpperCase();
Serial.println(tag);
if (tag == " 59 AA 66 E5") {
Serial.println("Tushar");
val = 1;
}
else {
if (tag == " 27 39 2C A1") {
Serial.println("Nikita");
val = 2;
}
else {
if (tag == " 2703 31 A1") {
Serial.println("Anamika");
val = 3;
}
else {
if (tag == " B7 C6 F7 67") {
Serial.println("Sandeep");
val = 4;
}}}}
tag = "";
delay(1000);
rfid.PICC_HaltA();
rfid.PCD_StopCrypto1();
// Now we can publish stuff!
Serial.print(F("\nSendingval "));
Serial.print(val);
if (! att.publish(val)) {
Serial.println(F(" Failed"));
} else {
Serial.println(F(" OK!"));
}
// wait a couple seconds to avoid rate limit
// delay(1000);
}
// Function to connect and reconnect as necessary to the
MQTT server.
// Should be called in the loop function and it will take
care if connecting.
voidMQTT_connect() {
int8_t ret;
// Stop if already connected.
if (mqtt.connected()) {
return;
}
Serial.print("Connecting to MQTT... ");
uint8_t retries = 3;
while ((ret = mqtt.connect()) != 0) { // connect will return 0 for connected
Serial.println(mqtt.connectErrorString(ret));
Serial.println("Retrying MQTT connection in 5 seconds...");
mqtt.disconnect();
delay(5000);  // wait 5 seconds
retries--;
if (retries == 0) {
// basically die and wait for WDT to reset me
Serial.println("Can't connect, reset the module");
while (1);
}}
Serial.println("MQTT Connected!");
}
```

8. CONCLUSION

IoT based RFID Attendance System is designed, in which attendance is directly recorded on the internet. Which reduces the chance of data loss? For designing this attendance system, ESP32 module is used and programming is written in Embedded C language using an IDE called Arduino IDE. The Adafruit_MQTT allows the program program to send only numerical values to the website. So, we can send only roll numbers of any student or employee rather than sending their names.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] K. Ding, P. Jiang. IEEE/CAA Journal of AutomaticaSinica, 5, 1:128–138(2017).

[2] A. Singh, S. Meshram, T. Gujar, P.R. Wankhede. Baggage tracing and handlingsystem using RFID and IoT for airports. International Conference on Computing, Analytics and Security Trends (CAST), (Pune, India, 2016).

[3] T. Sharma, S.L. Aarthy. An automatic attendance monitoring system using RFID andIOT using Cloud. 2016 Online International Conference on Green Engineering andTechnologies (IC-GET), (Coimbatore, India, 2016).

[4] H.M. Nguyen, S.H. Kim, D.T. Le, S. Heo, J. Im, D. Kim. Optimizations for RFID-based IoT applications on the Cloud. 5th International Conference on the Internet ofThings, (Seoul, South Korea, 2015).

[5] O.T. Arulogun, A. Olatunbosun, O.A. Fakolujo, O.M. Olaniyi. IJSER, 4, 2:1–9(2013).

[6] S. Nainan, R. Parekh, T. Shah. IJCSI, 10,1(2013).

[7] O.G., Chiagozie, O.G. Nwaji. Academic Research International, 2,2:168–183 (2012.

[8] Z. Singhal, R.K. Gujral, International Journal of Computer Applications, 39,3:37–41 (2012).

[9] J.D. Irawan, S. Prasetio, S.A. Wibowo. IP based module for building automation system. In: Proceedings of second international conference on electrical systems, technology and information 2015 (ICESTI 2015). F. Pasila, Y. Tanoto, R. Lim, M. Santoso, N. Pah (Eds), Singapore: Springer (2016). pp. 337–343).

[10] N. Saparkhojayev, S. Guvercin, IJCSI, 9,3: 227–230(2012).