RFID Based Smart Classroom Monitoring System

Nayem Sarkar[*], Neamul Islam Fahim[†], Tairin Islam[‡], Shah Merajur Rahman[§], and Rabbi Islam Yeasin[§]
Dept. of Computer Science and Engineering
United International University, Dhaka, Bangladesh
011203032[*], *011202216*[†], *011203012*[‡], *011201356*[§], and *011201067*[§]

***Abstract*— In this digital era, we have to do everything manually in classroom which is also time consuming. We have to turn off the lights, fix rooms temperature manually.In our project we used RFID technology thats offers auto attendance, instantly can see faculty's name and course information, auto control of temperature of the room, auto light on system. Our project will save a lot of time and hassle. Don't have to do things manually. Everything in the classroom will be automated**

RFID; Smart; Classroom; Monitoring; System

I. Project Overview

Technological improvements in the modern educational landscape have changed traditional teaching approaches. The RFID-based Smart Classroom Monitoring System is a novel solution that uses Radio-Frequency Identification (RFID) technology to improve classroom administration and monitoring procedures.

This system's fundamental goal is to automate attendance tracking, decreasing manual labor and boosting accuracy. The solution offers seamless and automated attendance recording by integrating RFID tags or cards within student IDs. This automation not only saves class time but also provides a more accurate depiction of student presence, allowing educators to alter teaching tactics based on attendance patterns [1].

Furthermore, the RFID-based access control feature acts as a strong security precaution in educational environments. It successfully prevents unwanted entrance to classrooms, resulting in a safe learning environment. This feature offers teachers and administrators with real-time information about classroom occupancy, allowing them to effectively regulate access during ongoing classes [2].

The centralized database of the system securely saves attendance records and related information. This data warehouse enables complete reporting and analytics capabilities. These capabilities can be used by educators to generate attendance records, track student involvement, and discover attendance trends. Such insights allow for data-driven decision-making, which assists educators in improving instructional approaches and tackling attendance-related issues [3].

The RFID-based Smart Classroom Monitoring System takes a revolutionary approach to classroom management by seamlessly integrating RFID technology into educational institutions. Its multiple functions expedite attendance monitoring, strengthen security measures, and give useful data insights, helping educators to promote a more engaging and secure learning environment.

II. Component List

- Arduino UNO
- RFID Sensor
- LCD Display (20x4)
- I2C LCD Adapted (Display module)
- Relay single channel (Keyes -SRLY)
- DHT 11 Sensor Module (Keys)
- Infrared LED
- Infrared Receiver
- 2N2222 NPN Transistor
- 470k Resistor
- Buzzer
- LED -(G,R)
- Bread Board Mini (color)
- Jumper Set (all types)
- Male Header Connector Pin (single)
- Female Header Connector Pin (single)
- Wire (AC)
- Holder
- AC Bulb
- Double Sided Foam Tape
- **Arduino UNO**
  Description: It's a popular microcontroller board based on the ATmega328P microcontroller.It features digital and analog input/output pins that allow users to interact with various electronic components and sensor.

- **RFID Sensor**
  Description: Radio-Frequency Identification (RFID) sensors are used to identify and track tags or cards wirelessly. They consist of an RFID reader and compatible tags that can be read from a distance without

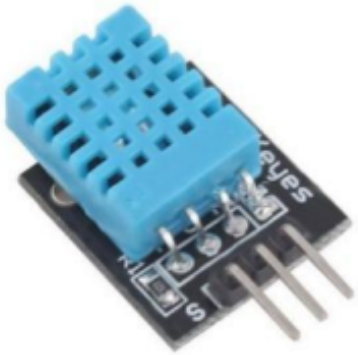
  direct contact.

- **LCD Display (20x4)**
  Description: This Liquid Crystal Display (LCD) can display 20 characters per line across 4 lines. It's commonly used to output information in projects due to its readability and clarity.

- **I2C LCD Adapted (Display module)**
  Description: It's an adapter for connecting an LCD display using the I2C (Inter-Integrated Circuit) protocol, which reduces the number of wires needed for communication between the display and microcontroller.

- **Relay single channel (Keyes -SRLY)**
  Description: A relay module that can control higher voltage and current devices using a lower voltage signal. It acts as a switch controlled by the microcontroller.

- **DHT 11 Sensor Module (Keys)**
  Description: The DHT11 sensor measures temperature and humidity and outputs digital signals. It's a low-cost sensor suitable for basic weather monitoring



  applications.

- **Infrared LED**
  Description: Emitting infrared light, these LEDs are used in applications like remote controls, proximity sensors, and communication systems that work with IR

  signals.

- **Infrared Receiver**
  Description: This component receives and decodes infrared signals. It's used in devices like IR remote controllers and communication modules that utilize IR

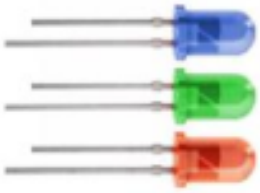
  communication.

- **2N2222 NPN Transistor**
  Description: A versatile transistor used for amplification and switching applications in electronic circuits. It's known for its low cost and wide availability.

- **470k Resistor**
  Description: A resistor with a resistance value of 470,000 ohms. It can be used for current limiting, voltage division, or biasing purposes in electronic circuits.

- **Buzzer**
  Description: An audio signaling device that generates sound when an electrical signal is applied. It's used for alarms, notifications, and various audio indications.

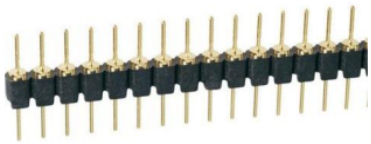
- **LED-(G,R)**
  Description: Light-emitting diodes emitting green and red lights. LEDs are used for visual indicators and status displays in electronic projects.



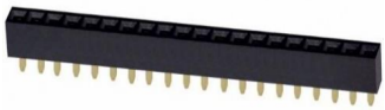
- **Breadboard Mini (color)**
  Description: A solderless prototyping board used for creating temporary electronic circuits. It allows easy placement and connection of electronic components for testing and experimentation.

- **Jumper Set (all types)**
  Description: Assorted jumper wires used to create connections between various components on a breadboard or within a circuit.

- **Male Header Connector Pin (single)**
  Description: Pins with male connectors used for making electrical connections on circuit boards or modules.



- **Female Header Connector Pin (single)**
  Description: Receptacles with female connectors that accept male header pins, enabling secure connections between components.



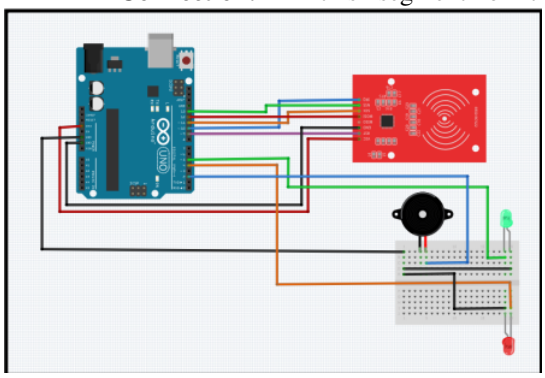
- **Wire (AC)**
  Description: Electrical wire designed for alternating current (AC) applications, typically used to connect components in AC circuits.

- **Holder**
  Description: A component used for supporting or securing other components in place, providing structural stability.

- **AC Bulb**
  Description: A light bulb designed to operate on alternating current (AC) power sources.
- **Double Sided Foam Tape**
  Description: This tape has adhesive on both sides and is commonly used for mounting or affixing components securely in place.

III. Implementation

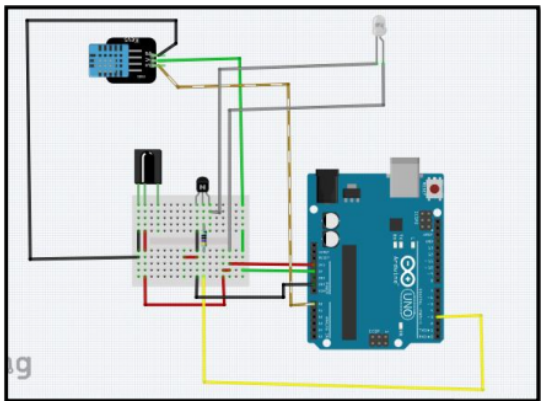
RFID Connection: In this segment of the circuit,



- We have used:
  - A breadboard.
  - A buzzer.
  - RFID sensor.
  - LED lights.

The Arduino will store the whole code from the PC and execute the instructions as given in the code. First of all, when a RFID is scanned on the sensor, it will detect the ID type. If it is a student ID, it will switch on the lights and AC of the classroom, and take the attendance for the first student. If the ID belongs to the second student, it will only take attendance as the lights and fan are already switched on. For faculty IDs, it will display the faculty name, course name, total students enrolled in the course, and the duration of the class on the display. The green LED will flash on successful ID card reading, while the red light will blink in case of any error.

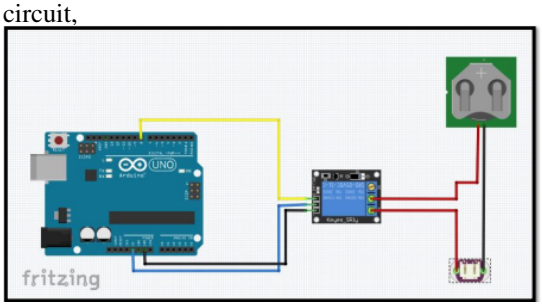
Temperature Module Connection: In this segment of the circuit,



- We have used:
  - MOSFET.
  - IR receiver.
  - IR transmitter LED.
  - 470 kΩ resistor.
  - DHT 11.

The IR receiver will store the signal from the AC remote initially and pass it to the AC through the IR transmitter. DHT 11 is used for monitoring the classroom temperature, and the code will control the AC by analyzing temperature data.

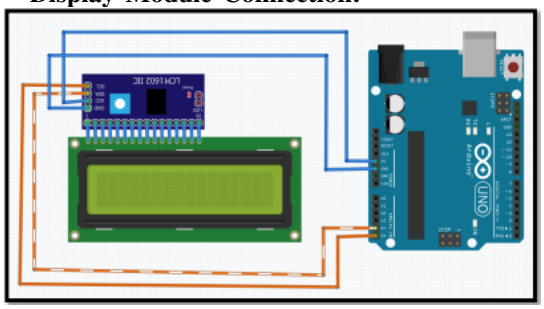
Light Controlling Connection: In this segment of the circuit,



- We have used a relay which is for switching.

The lights and fans will be in series connection. The relay is designed to be normally closed. When anyone scans the RFID, it passes a high signal to turn on the lights and fans.

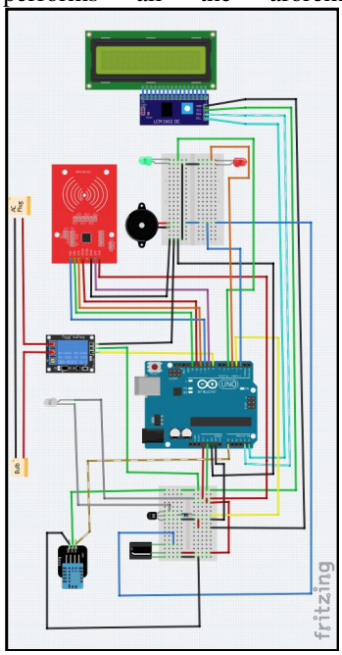
Display Module Connection:



The display will show:

- The Faculty name.
- Course code.
- Total number of students.
- Class hours.

Final Circuit: Here is the final circuit that performs all the aforementioned tasks together.



IV. Future scope of work

We would like to Improve our project for future. The door will open automatically . Auto user identification can be other improvement.

V. Appendix

A. Project Code

```
#include <SPI.h>
#include <MFRC522.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include <IRLibSendBase.h>
#include <IRLib_HashRaw.h>
#include <dht.h>
#define SS_PIN 10
#define RST_PIN 9
#define LED_G 4
#define LED_R 5
#define BUZZER 7
#define dht_apin A0

dht DHT;

IRsendRaw mySender;

MFRC522 mfrc522(SS_PIN, RST_PIN);

LiquidCrystal_I2C lcd(0x27, 20, 4);
int relay = 8;
}
```

Listing 1. Project Code

```
void setup() {
  Serial.begin(9600);   // Initiate a
 serial communication
  SPI.begin();          // Initiate
SPI bus
  mfrc522.PCD_Init();   // Initiate
MFRC522
  pinMode(LED_G, OUTPUT);
  pinMode(LED_R, OUTPUT);
  pinMode(BUZZER, OUTPUT);
  noTone(BUZZER);
  Serial.println("Put your card to
the reader...");
  Serial.println();
  lcd.init();
  lcd.backlight();
  pinMode(relay, OUTPUT);
}


#define RAW_DATA_LEN 350
uint16_t rawData[RAW_DATA_LEN] = {
  9326, 2594, 554, 750, 558, 742,
562, 742,
  554, 750, 554, 746, 562, 742, 562,
 738,
  558, 1914, 558, 1914, 558, 746,
558, 742,
  566, 738, 554, 746, 562, 742, 562,
 1910,
  562, 738, 558, 746, 558, 746, 562,
 738,
  558, 746, 558, 746, 562, 738, 554,
 750,
  558, 742, 562, 742, 554, 746, 558,
 746,
  562, 742, 562, 738, 558, 746, 562,
 742,
  562, 738, 558, 746, 558, 742, 566,
 1906,
  566, 738, 554, 746, 562, 1910,
562, 1910,
  562, 742, 562, 738, 558, 746, 558,
 742,
  566, 738, 558, 746, 558, 742, 562,
 742,
  554, 746, 562, 742, 562, 742, 554,
 746,
  558, 746, 562, 742, 562, 738, 558,
 746,
  558, 1914, 558, 742, 566, 738,
558, 746,
  558, 742, 562, 742, 554, 746, 562,
 742,
  562, 742, 554, 1914, 558, 746,
558, 1914,
  558, 746, 562, 738, 558, 1914,
558, 746,
  558, 1910, 562, 1000
};
```

Listing 2. Project Code

```
    void loop() {
      DHT.read11(dht_apin);
      int temp = DHT.temperature;
      Serial.print("temperature = ");
      Serial.print(temp);
      Serial.println(" C");

      // Look for new cards
      if (!mfrc522.PICC_IsNewCardPresent
    ()) {
        return;
      }
      // Select one of the cards
      if (!mfrc522.PICC_ReadCardSerial()
    ) {
        return;
      }
      mySender.send(rawData,
    RAW_DATA_LEN, 36); //Pass the buffer
    ,length, optionally frequency
      Serial.println(F("AC Switched On")
    );

      //Show UID on serial monitor
      Serial.print("UID tag :");
      String content = "";
      //  byte letter;
      for (byte i = 0; i < mfrc522.uid.
    size; i++) {
        Serial.print(mfrc522.uid.uidByte
    [i] < 0x10 ? " 0" : " ");
        Serial.print(mfrc522.uid.uidByte
    [i], HEX);
        content.concat(String(mfrc522.
    uid.uidByte[i] < 0x10 ? " 0" : " "))
    ;
        content.concat(String(mfrc522.
    uid.uidByte[i], HEX));
      }
      Serial.println();
      Serial.print("Message : ");
      content.toUpperCase();
      if (content.substring(1) == "0A 9F
     FE BE")
      {
        Serial.println("Authorized
    access");
        Serial.println("Nur Hossain
    Rabbi - 011203016");
        Serial.println();
        delay(500);

        digitalWrite(LED_G, HIGH);
        tone(BUZZER, 500);
        delay(1500);
        digitalWrite(relay, HIGH);
```

Listing 3. Project Code

```
lcd.print("Faculty  Name ");

        lcd.setCursor(0, 1);
        lcd.print("Course-Code");

        lcd.setCursor(0, 2);
        lcd.print(" 01  ");

        lcd.setCursor(0, 3);
        lcd.print("Start - End");
        delay(300);
        noTone(BUZZER);
        //delay(5000);
        digitalWrite(LED_G, LOW);
      }

      else if (content.substring(1) == "
    AC 0F EF 5B")  //change here the UID
     of the card/cards that you want to
    give access
      {
        Serial.println("Authorized
    access");
        //    Serial.println("Boos -
    011203016");
        Serial.println();
        // delay(500);
        digitalWrite(LED_G, HIGH);
        delay(1500);
        digitalWrite(relay, LOW);  //
    turn the bulb off
        tone(BUZZER, 500);
        lcd.print("                    ");

        lcd.setCursor(0, 1);
        lcd.print("        Class       "
    );

        lcd.setCursor(0, 2);
        lcd.print("        Empty       "
    );

        lcd.setCursor(0, 3);
        lcd.print("                    ");
        // delay(300);
        noTone(BUZZER);
        //delay(5000);
        digitalWrite(LED_G, LOW);


      }

      else {
        Serial.println(" Access denied")
    ;
        digitalWrite(LED_R, HIGH);
        tone(BUZZER, 300);
        delay(1000);
        digitalWrite(LED_R, LOW);
        noTone(BUZZER);
      }
    }
```

Listing 4. Project Code

B. Complete Code

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include <time.h>

//creating  Student Structure
struct Student
{
    char RFID[10];
    int ID;
    char name[50];
};
struct StudentInfo
{
    char S_RFID[10];
    int ID;
    char name[50];
};

//creating  Faculty Structure
struct FacultyInfo
{
    char F_RFID[10];
    char name[50];
    int course[50][50];
};

//gloval variable
int totalStudent=0;//count student
char FacultyName[50]="";//Faculty Name
char CourseName[50]="";//Course Name
int day;
//functions
void AC_Con();//Function for AC Control
void Light_Controll();//function for  Light control
void ClassTimer();//function for classTimer
void Display();// Function for Display
void Attendence(struct StudentInfo SI[],struct tm tm);// function for Attendence
```

Listing 5. Complete Code

```
int main()//setup
{
    //initializing Variables
    int AC = 0;
    int Light = 0;
    char ID[8];


    time_t t;
    t = time(NULL);
    struct tm tm = *localtime(&t);
    int day=0,period;

    //initializing day
     if(tm.tm_mday== 1)
     {
         day = 1;
     }
     else if(tm.tm_mday== 2)
     {
         day = 2;
     }
     else if(tm.tm_mday== 4)
     {
         day = 3;
     }
     else if(tm.tm_mday== 5)
     {
         day = 5;
     }
```

Listing 6. Complete Code

```
//including Structure Data from txt files
     struct Student S[100];
     struct StudentInfo SI[100];
     struct FacultyInfo FCI[10];


     // Scanning Students Information
     FILE *fp1;
     fp1=fopen("Student.txt","r");
     for(int i = 0; i<10; i++)
     {
          fscanf(fp1,"%s",&S[i].RFID);
          fscanf(fp1,"%d",&S[i].ID);
          fscanf(fp1,"%s",&S[i].name);

     }
     fclose(fp1);

     // Scanning Faculty Information

     FILE *fp2;
     fp2=fopen("Facultycode.txt","r");
     for(int i = 0; i<10; i++)
     {
          fscanf(fp2,"%s",&FCI[i].F_RFID);
          fscanf(fp2,"%s",&FCI[i].name);

          for(int j = 0; j<4; j++)
          {
               for(int k = 0; k<4; k++)
               {
                    fscanf(fp2,"%d",&FCI[i].course[j][k]);
               }
          }
     }
     fclose(fp2);
     Display();
}
```

Listing 7. Complete Code

```
void loop()
{
     Display();
     char rfid[20];
     gets(rfid);// Scan RFID

     int j = 0;
//Removing Space from ID
     for (int i = 0; rfid[i] != '\0'; i++)
     {
          if (rfid[i] != ' ')
          {
               rfid[j] = rfid[i];
               j++;
          }
     }
     rfid[j] = '\0';


     //Searching/ Comparing ID with Student
     for(int i = 0; i<10; i++)
     {
          int Cmp = strcmp(rfid,S[i].RFID);
          if(Cmp == 0)
          {
               // if ID match,than assign student info to StudentInfo Structure for Attendance
               SI[totalStudent].S_RFID[] = S[i].RFID[];
               SI[totalStudent].ID[] = S[i].ID[];
               SI[totalStudent].name[] =S[i].name[];

               //Counting Students attending in the class
               totalStudent++;

               // Calling AC Control Class/Function
               AC_Con(AC);

               //Calling Light Control function
               Light_Controll(Light);
               Display();
          }
     }
     //Searching/ Comparing ID with Faculty
     for(int i = 0; i<10; i++)
     {
          int Cmp=strcmp(rfid,FCI[i].F_RFID);
          if(Cmp==0)
          {
               FacultyName=FCI[i].name[];
               CourseName=FCI[i].course[day][period]);
          }
     }
     Display();
}
```

Listing 8. Complete Code

```
void AC_Con(int x){
    if(x>=26){
        //AC on
        printf("\nAC ON");
    }else if(x <=19){
        //AC off
        printf("\nAC OFF");
    }else if(x = 0){
        printf("\nTemperature
Scanning Error");
    }else{
        printf("\nTemperature is OK
");
    }
}
void Light_Controll(int y){
    if(y=0){
        //Light on
        printf("\nLight turned ON")
;
    }else if(y = 1){
        //Light off
        printf("\nLight turned OFF"
);
    }else{
        printf("\nLight Scanning
Error");
    }
}
void Attendance(struct StudentInfo
SI[],struct tm tm){
    FILE *fp;
    fp=fopen("Attendance.txt","a");
    fprintf(fp, "Course Name - %s\n
",CourseName);
    fprintf(fp, "Current Date: %d-%
d-%d\n", tm.tm_mday, tm.tm_mon+1, tm
.tm_year+1900);
    fprintf(fp, "Current Time: %d:%
d:%d\n", tm.tm_hour, tm.tm_min, tm.
tm_sec);
    fprintf(fp, "\n\nTotal Student
- %d\n",totalStudent);
    for(int i = 0; i<10; i++){
        fprintf(fp,"\t%d)",i+1);
        fprintf(fp,"\t%d",SI[i].ID)
;
        fprintf(fp,"\t%s\n",SI[i].
name);
    }printf("\n02.Print Done
......!!!\n");
}
void Display(int a, int b){
    if(FacultyName[0]!= 0){
        CourseName
        FacultyName
        totalStudent
        ClassTimer();
    }printf("\nAmount of Student: %
d",a);
    printf("\nTime left %d",b);
}
```

Listing 9. Complete Code

```
void ClassTimer(){
    s = 5400;
    while(s > 0){
        hours = (s/3600);
        minutes = (s -(3600*hours))
/60;
        seconds = (s -(3600*hours)
-(minutes*60));
        tm->tm_hour = hours;
        tm->tm_min = minutes;
        tm->tm_sec = seconds;
        mktime(tm);
        printf("%02d:%02d:%02d\n",
tm->tm_hour, tm->tm_min, tm->tm_sec)
;
        s--;
        clock_t end_wait;
        end_wait = clock() + 1 *
CLK_TCK;
        while (clock() < end_wait)
{}
}
```

Listing 10. Complete Code

REFERENCES

[1] H. Gupta, A. Singh, and S. Gupta, "Rfid-based attendance system using arduino," *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, vol. 6, no. 3, March 2018.

[2] S. Jain, A. Sharma, and R. Aggarwal, "Smart classroom monitoring and controlling system using iot," *International Journal of Computer Applications*, vol. 182, no. 16, March 2018.

[3] R. Mohan and K. Chandrasekhar, "Rfid based smart attendance system with face recognition using raspberry pi," *International Journal of Scientific Research in Computer Science, Engineering, and Information Technology*, vol. 3, no. 4, September 2018.