

```

#include <SPI.h>
#include <MFRC522.h>
#include <LiquidCrystal.h>
#include <LiquidCrystal_I2C.h>

#define SS_PIN 10 // ESP32 pin GIOP5
#define RST_PIN 9 // ESP32 pin GIOP27
#define RELAY_PIN 8 // ESP32 pin GIOP8 connects to relay

int totalColumns = 16;
int totalRows = 2;

LiquidCrystal_I2C lcd(0x27, totalColumns, totalRows);

int tap = 0;
int charging_time = 30000;
int display_delay = 2000;
unsigned long time_now = 0;

//All Strings
String welcomeMessage = "MAJOR PROJECT";
String titleMessage = "ADVANCE STRUCTURE OF CHARGING PORTS IN RAILWAYS";
String groupinfoMessage = "PROJECT CREATED BY GROUP NO. 16 FINAL YEAR ELECTRICAL DEPARTMENT";
String tapMessage = "TAP YOUR POWER ID FOR CHARGING";
String invalidMessage = "USE A VALID POWER ID FOR CHARGING";
String chargebreakMessage = "CURRENTLY CHARGING NOT AVAILABLE FOR 30 SECONDS";

//time reverse variables
int S = 60; // count seconds
int M = (2*60); // count minutes
int H = 0; // count hours

MFRC522 rfid(SS_PIN, RST_PIN);

byte authorizedUID[4] = {0xE9, 0x29, 0x74, 0xE9};

void scrollMessage(int row, String message, int delayTime, int totalColumns) {
  for (int i=0; i < totalColumns; i++) {
    message = " " + message;
  }
  message = message + " ";
  for (int position = 0; position < message.length(); position++) {
    lcd.setCursor(0, row);
    lcd.print(message.substring(position, position + totalColumns));
    delay(delayTime);
  }
}

```

```

void setup() {
  Serial.begin(9600);
  SPI.begin(); // init SPI bus
  rfid.PCD_Init(); // init MFRC522
  pinMode(RELAY_PIN, OUTPUT); // initialize pin as an output.
  digitalWrite(RELAY_PIN, HIGH); // deactivate the relay

  lcd.init();
  lcd.clear();
  lcd.backlight();

  lcd.setCursor(1,0);
  lcd.print(welcomeMessage);
  scrollMessage(1, titleMessage, 250, totalColumns);
  Serial.println("ADVANCE STRUCTURE OF CHARGING PORTS IN RAILWAYS");

  //delay(display_delay);

  lcd.setCursor(1,0);
  lcd.print(welcomeMessage);
  scrollMessage(1, groupinfoMessage, 250, totalColumns);
  Serial.println("PROJECT CREATED BY GROUP NO. 16 ELECTRICAL
DEPARTMENT");

  //delay(display_delay);

  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("INSTRUCTION:");
  scrollMessage(1, tapMessage, 250, totalColumns);

  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("TAP POWER ID");
  Serial.println("Tap an RFID/NFC tag for Charging");
}

void loop() {
  if (rfid.PICC_IsNewCardPresent()) { // new tag is available
    if (rfid.PICC_ReadCardSerial()) { // NUID has been readed
      MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);

      if (rfid.uid.uidByte[0] == authorizedUID[0] &&
          rfid.uid.uidByte[1] == authorizedUID[1] &&
          rfid.uid.uidByte[2] == authorizedUID[2] &&
          rfid.uid.uidByte[3] == authorizedUID[3]) {
        //Serial.println("Authorized Tag");

        lcd.setCursor(1,0);

```

```

lcd.print("AUTHORIZED ID");
delay(display_delay);

}

else{
  tap = 1;
  lcd.setCursor(0,0);
  lcd.print("UNAUTHORIZED ID");

  //Serial.println("Un-authorized Tag");
  delay(display_delay);

  lcd.setCursor(0,0);
  Serial.println("Use a Valid Tag for Charging");

  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("TAP POWER ID");
  Serial.println("Tap an RFID/NFC tag for Charging");
}

if(tap == 0){
  digitalWrite(RELAY_PIN, LOW); // activate the relay for 3 MINUTES

  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("CHARGING ON");

  for (int t = (S+M); t >= 0; t--){

    if (rfid.PICC_IsNewCardPresent()) { // new tag is available
      if (rfid.PICC_ReadCardSerial()) { // NUID has been readed
        MFRC522::PICC_Type piccType = rfid.PICC_GetType(rfid.uid.sak);

        if (rfid.uid.uidByte[0] == authorizedUID[0] &&
            rfid.uid.uidByte[1] == authorizedUID[1] &&
            rfid.uid.uidByte[2] == authorizedUID[2] &&
            rfid.uid.uidByte[3] == authorizedUID[3]) {
          break;
        }
      }
    }
  }

  delay(1000);
  lcd.clear();
  lcd.setCursor(2,0);
  lcd.print("CHARGING ON");
}

```

```

    lcd.setCursor(4, 1);
    lcd.print(t);

    lcd.setCursor(8, 1);
    lcd.print("sec");
}

tap = 1;

//delay(charging_time);

digitalWrite(RELAY_PIN, HIGH);
tap = 0;

lcd.clear();
lcd.setCursor(2,0);
lcd.print("CHARGING OFF");

delay(display_delay);

lcd.clear();
lcd.setCursor(0,0);
lcd.print("NOTE:");
scrollMessage(1, chargebreakMessage, 250, totalColumns);

lcd.clear();
lcd.setCursor(1,0);
lcd.print("CHARGING BREAK");
lcd.setCursor(3,1);
lcd.print("RIGHT NOW");
delay(charging_time);

lcd.clear();
lcd.setCursor(2,0);
lcd.print("TAP POWER ID");
Serial.println("Tap an RFID/NFC tag for Charging");

}
else{
    digitalWrite(RELAY_PIN, HIGH); // deactivate the relay
    tap = 0;
}
}

rfid.PICC_HaltA(); // halt PICC
rfid.PCD_StopCrypto1(); // stop encryption on PCD
}
}

```