

```
//projekt MIFARE
```

```
//dolaczanie bibliotek obslugi RFID, RTC, LCD
```

```
#include <SPI.h>
```

```
#include <MFRC522.h>
```

```
#include <Wire.h>
```

```
#include <DS3231_Simple.h>
```

```
#include <LiquidCrystal_I2C.h>
```

```
//inicjowanie obiektow
```

```
DS3231_Simple Clock;
```

```
LiquidCrystal_I2C lcd(0x27,16,2); // ustawienie LCD na adresie 0x27 - wyswietlacz 16 znakow i 2 linie
```

```
MFRC522 rfid(10, 9);
```

```
MFRC522::MIFARE_Key key;
```

```
//deklaracja zmiennych globalnych
```

```
const byte UID[] = {0x45, 0x04, 0x0E, 0x88}; // tablica binarna UID karty uprawnionej do otwarcia zamka
```

```
#define czas_trwania 5000 // czas opoznienia w [ms] uruchomienia elektrozamka
```

```
boolean stan = false; // zmienna typu logicznego 0/1
```

```
unsigned long czas; // zmienna typu calkowitego unsigned LONG
```

```
void setup() {  
  Serial.flush(); //czyszczenie bufora portu szeregowego  
  Serial.begin(9600); // ustawienie przepustowosci portu szeregowego  
  SPI.begin();  
  rfid.PCD_Init();  
  pinMode(2, OUTPUT); //ustawienie wyjscia cyfrowego pin D2 dla diody sygnalizacyjnej LED CZERWONA  
  pinMode(3, OUTPUT); //ustawienie wyjscia cyfrowego pin D3 dla diody sygnalizacyjnej LED ZIELONA  
  pinMode(4, OUTPUT); //ustawienie wyjscia pin D4 dla buzzera  
  pinMode(5, OUTPUT); //ustawienie wyjscia pin D5 dla przekaznika 1-kanalowego  
  Clock.begin();  
  lcd.init(); // inicjalizacja ekranu LCD  
  lcd.backlight(); // ustawienie podswietlenia  
  lcd.clear(); // czyszczenie ekranu LCD  
  lcd.print("Inicjalizacja!"); // wyswietlenie komunikatu startowego na ekranie LCD.  
  delay(100); // opoznienie 100ms  
}  
  
//.....GLOWNA PETLA PROGRAMU.....  
void loop() {
```

```
if (rfid.PICC_IsNewCardPresent() && rfid.PICC_ReadCardSerial()) //odczyt karty Mifare
{
if (rfid.uid.uidByte[0] == UID[0] && //sprawdzenie odczytanych wartosci z nr UID karty uprawnionej
    rfid.uid.uidByte[1] == UID[1] &&
    rfid.uid.uidByte[2] == UID[2] &&
    rfid.uid.uidByte[3] == UID[3])
{
Serial.print("Pracownik nr 1 (uprawniony OTWIERAM)- ");
odczyt_rtc (); //odczyt aktualnego czasu z modulu RTC
lcd.clear(); //czyszczenie ekranu LCD
odczyt_uid (); //wywołanie funkcji odczyt UID karty
lcd.setCursor(0, 1); //ustawienie kursora w lini drugiej
lcd.print("Karta nr 1"); //wyswietlenie komunikatu na ekranie LCD
stan = true; //ustawienie zmiennej stan=1
czas = millis() + czas_trwania; //wprowadzenie interwalu 5sek - delay(5000);
} else
{
Serial.print("Pracownik nr 2 (nie uprawniony)- ");
odczyt_rtc ();
lcd.clear();
```

```
odczyt_uid ();  
lcd.setCursor(0, 1);  
lcd.print("Karta nr 2");  
stan = false;  
}  
rfid.PICC_HaltA();  
rfid.PCD_StopCrypto1();  
}
```

```
if (stan && czas < millis())
```

```
    stan = false;      //zmiana stan=0 ustawiana po splnieniu obu warunkow jednocześnie - operator AND
```

```
digitalWrite(3, stan); //uruchomienie diode LED ZIELONA 1-ON 0-OFF - w zaleznosci od wartosci zmiennej STAN
```

```
digitalWrite(2, !stan); //uruchomienie diode LED czerwonej - negacja wartosci zmiennej STAN
```

```
digitalWrite(5, stan); //wlacza lub wylacza przekaznik zgodnie z wartoscia zmiennej STAN
```

```
digitalWrite(4, !stan); //uruchomienie buzzer tj wartosc 0 wzbudza tongenerator (dlatego negacja zmiennej STAN
```

```
}
```

```
//..... KONIEC GLOWNEGO BLOKU PROGRAMU.....
```

```
// ponizej definicje funkcji:
```

```
void odczyt_uid() {
```

```
lcd.setCursor(0, 0); //ustawienie kursora w wierszu pierwszym - dane wypisywane na LCD
lcd.print("UID: ");
lcd.print(rfid.uid.uidByte[0], HEX);
lcd.print(":");
lcd.print(rfid.uid.uidByte[1], HEX);
lcd.print(":");
lcd.print(rfid.uid.uidByte[2], HEX);
lcd.print(":");
lcd.print(rfid.uid.uidByte[3], HEX);
lcd.print(" ");
}

void odczyt_rtc() {
    DateTime MyDateAndTime; // utworzenie zmiennej do przechowywania danych czasowych
    MyDateAndTime = Clock.read(); //odczytanie danych z zegara RTC
    Serial.print("Data: 20"); // wypisywanie danych na monitorze portu szeregowego
    Serial.print(MyDateAndTime.Year);
    Serial.print("-");
    Serial.print(MyDateAndTime.Month);
    Serial.print("-");
```

```
Serial.print(MyDateAndTime.Day);  
Serial.print(" Czas: ");  
Serial.print(MyDateAndTime.Hour);  
Serial.print(":");  
if (MyDateAndTime.Minute<10)  
    {  
        Serial.print("0");  
    }  
Serial.print(MyDateAndTime.Minute);  
Serial.print(":");  
if (MyDateAndTime.Second<10)  
    {  
        Serial.print("0");  
    }  
Serial.println(MyDateAndTime.Second);  
}
```