



Wydział Nauk Ekonomicznych i Technicznych

Marcin Białecki
Hubert Gwóźdź
Adrian Tretyn

Tworzenie alarmu przy pomocy Arduino

Praca zaliczeniowa
mgr inż. Arkadiusz Kubacki

Konin 2020

1. Temat: Tworzenie alarmu przy pomocy Arduino.

2. Wstęp

Celem projektu było stworzenie alarmu przy pomocy Arduino. Projekt był oparty na przerwaniach. Alarm był załączany kiedy czujnik światła wykrył natężenie światła poniżej określonego poziomu.

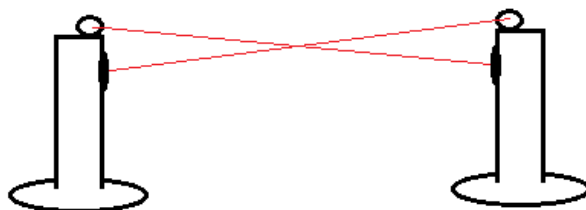
Zestaw wykorzystany do projektu:

- płytka Arduino (2 sztuki),
- laser (2 sztuki),
- fotorezystor (2 sztuki),
- rezystory 1 k Ω (2 sztuki),
- stojak na lasery i fotorezystory (2 sztuki),
- kabelki.

3. Opis projektu

Alarm oparty na przerwaniach, załącza się kiedy czujnik światła wykrywa natężenie światła poniżej określonego poziomu.

Działanie:



Na stojakach ustawione są dwa lasery, których wiązki światła trzeba ustawić tak, aby padały na fotorezystory umieszczone obok laserów. Do każdego stojaka przypada po jednej płytce Arduino, która odpowiada za właściwe zinterpretowanie sygnałów.

Arduino Uno z modułem Multi Function Shield odpowiada za uruchomienie alarmu w przypadku przerwania wiązki lasera wysyłanej z drugiego Arduino.

Przy starcie laser ciągle świeci ułatwiając ustawienie laserów w odpowiedniej pozycji. Po wciśnięciu przycisku A3 na płytce z modułem Multi Function Shield laser przestaje świecić i sprawdza czy wiązka z drugiego lasera pada cały czas na fotorezystor, w przypadku kiedy tak nie jest, załącza się alarm.

Druga płytka Arduino Uno zaprogramowana jest w taki sposób, że po otrzymaniu 3 mignięć lasera z pierwszego Arduino, wyłącza swój laser. Po 5 takich sygnałach, laser ponownie się włącza.

W celu oszczędzania energii oba Arduino wchodzi w stan uśpienia, jeśli nie zarejestrują żadnych działań użytkownika, bądź wiązka laserowa nie zostanie przerwana. Kod obu płytek oparty jest na przerwaniach, które wznowiają działanie Arduino kiedy zajdzie taka potrzeba.

Arduino Uno z modułem Multi Function Shield posiada 3 przyciski, które odpowiadają za:

- A1 - włączenie drugiego lasera (wysyła 3 sygnały laserem do drugiego Arduino)
- A2 - wyłączenie drugiego lasera (wysyła 5 sygnałów laserem do drugiego Arduino)
- A3 – aktywacja alarmu po pierwszym uruchomieniu oraz wyłączanie alarmu w przypadku jego aktywowania

4. Kody programu

4.1. Kod wgrany na płytkę Arduino Uno z modułem Multi Function Shield

```
//dodanie potrzebnych bibliotek
#include <PinChangeInterrupt.h>
#include <TimerOne.h>
#include <LowPower.h>
#include <MFShield.h>

//definiowanie nazw pinow
#define laserPin 5
#define fotoResPin A5
#define dioda1 13
#define dioda4 10
#define BUTTON1 A1
#define BUTTON2 A2
#define BUTTON3 A3
#define BUZZER 3

MFShield mfs; //tworzenie obiektu do obsługi modułu Multi Function Shield

volatile int fotoResStatus; //natezenie swiatla na fotorezystorze
volatile int timer = 0; //zmienna pozwalajaca wywoływac rozne funkcji co okreslony
czas
volatile int LaserOnStatus = 0; //zmienna ktora uaktywnia komunikacje z drugim
Arduino w celu wlaczenia lasera, przy ustawieniu na 1 zaczyna sie komunikacja, a
po zakonczeniu wysylania sygnalow zmienna ustawiana jest na 0
volatile int LaserOffStatus = 0; //zmienna ktora uaktywnia komunikacje z drugim
Arduino w celu wylaczenia lasera, przy ustawieniu na 1 zaczyna sie komunikacja, a
po zakonczeniu wysylania sygnalow zmienna ustawiana jest na 0
volatile bool alarmActivated = false; //aktywuje alarm
volatile int sleepMode = 0; //po przyjeciu okreslonej wartosci Arduino przechodzi
w stan uspienia

void setup() {
    Timer1.initialize(10000); //inicjalizacja timera na 10ms
    Timer1.attachInterrupt(actions); //przypisanie funkcji ktora bedzie wywoływana
co okreslony czas

    //ustawianie trybow kazdego z uzywanych pinow
    pinMode(dioda1, OUTPUT);
    pinMode(dioda4, OUTPUT);
    pinMode(laserPin, OUTPUT);
    pinMode(fotoResPin, INPUT);
    pinMode(BUTTON1, INPUT_PULLUP);
```

```

pinMode(BUTTON2, INPUT_PULLUP);
pinMode(BUTTON3, INPUT_PULLUP);

digitalWrite(diode1, LOW); //zapala diode1
digitalWrite(diode4, HIGH); //gasi diode4
digitalWrite(laserPin, HIGH); //aktywacja lasera

mfs.showDisplay(false); //wylaczenie wyswietlacza
fotoResStatus = analogRead(fotoResPin); //sprawdza natężenie światła padającego
na fotorezystor
mfs.display(fotoResStatus); //wyswietla na wyswietlaczu czterocyfrowym

while(digitalRead(BUTTON3)) //po uruchomieniu Arduino czeka na naciśnięcie
BUTTON3, dopoki nie wcisnie się tego przycisku alarm jest nieaktywny, można wtedy
ustawić lasery w odpowiedniej pozycji ponieważ ciągle świeci
{}

digitalWrite(diode1, HIGH); //gasi diode1
digitalWrite(laserPin, LOW); //wylacza laser

//ustawienie przerwań dla fotorezystora i trzech przycisków
attachPCINT(digitalPinToPCINT(fotoResPin), alarmOn, FALLING);
attachPCINT(digitalPinToPCINT(BUTTON1), LaserOn, RISING);
attachPCINT(digitalPinToPCINT(BUTTON2), LaserOff, RISING);
attachPCINT(digitalPinToPCINT(BUTTON3), alarmOff, RISING);
}

void loop() {
    if(sleepMode >= 1500){ //po 15 sekundach Arduino wejdzie w stan uspienia
        mfs.showDisplay(false); //wyłączenie wyswietlacza
        LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF); //wejście Arduino w stan
        uspienia, do czasu wystąpienia dowolnego przerwania
    }

    mfs.loop(); //odswierzanie wyswietlacza

    while(alarmActivated){ //załączenie alarmu dzwiekowego dopoki zmienna
    "alarmActivated" nie zostanie zmieniona na "false" poprzez odpowiednie przerwanie
        digitalWrite(BUZZER, LOW); //sygnał dzwiekowy Buzzera
        digitalWrite(diode4, LOW); //włączenie diody4
        delay(250); //odczekanie 250ms
        digitalWrite(diode4, HIGH); //wylaczenie diody4
        digitalWrite(BUZZER, HIGH); //wylaczenie sygnalu dzwiekowego Buzzera
        delay(250);
    }
}

void actions(){ //funkcja odpowiadająca za wywoływanie innych funkcji co określony
czas
    ++timer; //zwiększanie tej zmiennej pozwala zmierzyć czas aktywności Arduino
    ++sleepMode; //odlicza czas do uspienia Arduino

    if(timer%100 == 0){ //wyswietlanie co 1s stanu natężenia światła na
fotorezystorze
        fotoResStatus = analogRead(fotoResPin);
        mfs.display(fotoResStatus);
    }

    if(LaserOffStatus > 0) LaserOffFunction(); //aktywuje funkcję wylaczajaca laser
drugiego Arduino

```

```

    if(LaserOnStatus > 0) LaserOnFunction(); //aktywuje funkcje włączająca laser
    drugiego Arduino

    if(timer >= 2000000000) timer = 0; //zaczyna odliczanie od zera, kiedy zmienna
    "timer" jest bliska konca zakresu obsługiwanego przez typ intiger
}

void LaserOn(){
    LaserOnStatus = 1; //aktywuje funkcje włączająca laser drugiego Arduino
    sleepMode = 0; //resetuje odliczanie do uspienia
}

void LaserOnFunction(){ //funkcja wysyla 5 migniec laserem w ciagu 0.5s, ktore
    aktywują laser drugiego Arduino
    if(LaserOnStatus%10 == 0) //wylacza laser co 100ms
        digitalWrite(laserPin, LOW);
    else if (LaserOnStatus%10 == 5) //włącza laser co 100ms
        digitalWrite(laserPin, HIGH);

    if(LaserOnStatus >= 50) //limit czasu działania funkcji ustawiony na 0.5s
        LaserOnStatus = 0; //dezaktywuje funkcje
    else
        ++LaserOnStatus; //zwiększa licznik ilości wykonan funkcji, do czasu
osiągnięcia wartości 50
}

void LaserOff(){
    LaserOffStatus = 1; //aktywuje funkcje wyłączająca laser drugiego Arduino
    sleepMode = 0; //resetuje odliczanie do uspienia
}

void LaserOffFunction(){ //funkcja wysyla 3 migniec laserem w ciagu 0.6s, ktore
    dezaktywują laser drugiego Arduino
    if(LaserOffStatus%20 == 0) //wylacza laser co 200ms
        digitalWrite(laserPin, LOW);
    else if (LaserOffStatus%20 == 10) //włącza laser co 200ms
        digitalWrite(laserPin, HIGH);

    if(LaserOffStatus >= 60) //limit czasu działania funkcji ustawiony na 0.6s
        LaserOffStatus = 0; //dezaktywuje funkcje
    else
        ++LaserOffStatus; //zwiększa licznik ilości wykonan funkcji, do czasu
osiągnięcia wartości 60
}

void alarmOn(){
    alarmActivated = true; //aktywuje alarm
    digitalWrite(dioda4, LOW); //zapala diode4
    mfs.showDisplay(false); //wylacza wyswietlacz
    sleepMode = 0; //resetuje odliczanie do uspienia
}

void alarmOff(){
    alarmActivated = false; //dezaktywuje alarm
    digitalWrite(dioda4, HIGH); //gasi diode4
    sleepMode = 0; //resetuje odliczanie do uspienia
}

```

4.2. Kod wgrany na płytke Arduino Uno bez dodatkowych modułów

```
//dodanie potrzebnych bibliotek
```

```

#include <PinChangeInterrupt.h>
#include <LowPower.h>

//definiowanie nazw pinow
#define laserPin 2
#define fotoResPin A5

volatile byte laserInpulse = 0; //nalicza ilosc sygnalow lasera

void setup() {
    //ustawianie trybow kazdego z uzywanych pinow
    pinMode(laserPin, OUTPUT);
    pinMode(fotoResPin, INPUT);

    digitalWrite(laserPin, HIGH); //aktywacja lasera

    //ustawienie przerwania dla fotorezystora
    attachPCINT(digitalPinToPCINT(fotoResPin), laserOnOff, RISING);
}

void loop() {
    LowPower.powerDown(SLEEP_FOREVER, ADC_OFF, BOD_OFF); //wejście Arduino w stan
    uspienia, do czasu wystąpienia przerwania

    delay(1000); //czeka na naliczenie sygnalow
    if(laserInpulse == 3)
        digitalWrite(laserPin, LOW); //aktywuje laser
    else if(laserInpulse == 5)
        digitalWrite(laserPin, HIGH); //dezaktywuje laser
    laserInpulse = 0; //zeruje ilosc naliczonych impulsow lasera
}

void laserOnOff(){
    ++laserInpulse; //nalicza kazdy impuls
}

```

5. Wnioski

Jak można zauważyć mikrokontroler Arduino daje nam wiele możliwości do przydatnego działania. Potrzeba do tego "tylko" odpowiedniego kodu, kilka dodatkowych przedmiotów oraz tego co najważniejsze, czyli chęci do pracy i dobrego jej zaplanowania. Zrobienie tak działającego alarmu daje dużo przyjemności, ale też dużo doświadczenia i niekiedy może się przydać w codziennych zastosowaniach. Oczywiście taki mały mikrokontroler nie ma zastosowania w dużych firmach do wielkich projektów, ponieważ do takich są tworzone większe kontrolery na wzór Arduino, jednak do użytku domowego/osobistego jest wystarczający.