



AKADEMIA GÓRNICZO-HUTNICZA

Sprawozdanie z projektu

**„Alarm z czujnikiem światła działającym jako
czujnik ruchu”**

Technika mikroprocesorowa 2

Elektronika i Telekomunikacja, rok III

Mikołaj Telec

1. Cel projektu

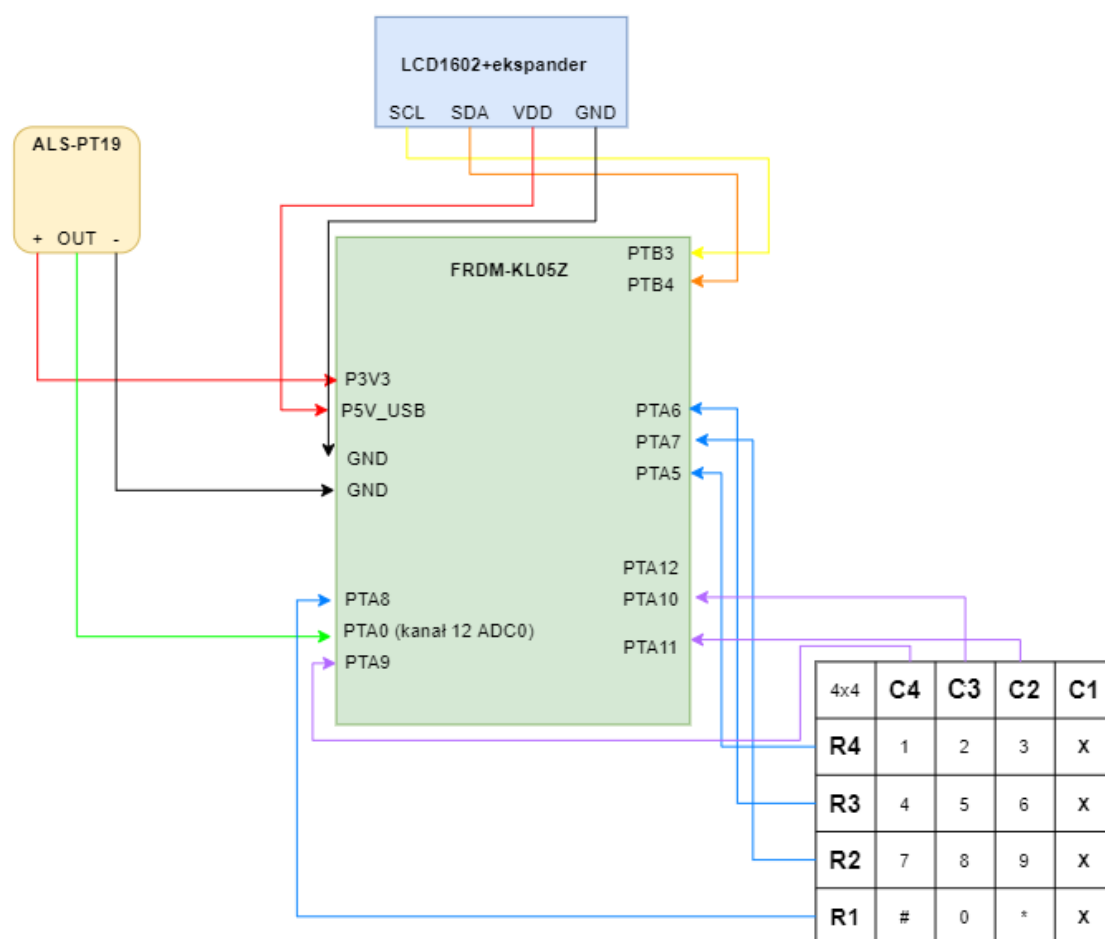
Celem projektu było stworzenie alarmu wykrywającego ruch przy pomocy czujnika światła. Alarm miał mieć możliwość uzbrajania i rozbrajania za pomocą szyfru oraz dwa tryby: użytkownika i administratora. Włamanie miało być sygnalizowane na diodach LED.

2. Użyte elementy oraz schemat połączeń

W projekcie oprócz zestawu FRDM-KL05ZJ wykorzystałem następujące elementy:

- wyświetlacz LCD1602A z ekspanderem I²C PCF8574,
- klawiatura 4x4,
- czujnik światła ALS-PT19.

Schemat blokowy połączeń:



Podłączenie elementów jest identyczne jak na zajęciach laboratoryjnych:

- Klawiatura:
 - C4 – PTA9, C3 – PTA10, C2 – PTA11, C1 – nie używam
 - R4 – PTA5, R3 – PTA6, R2 – PTA7, R1 – PTA8
- Czujnik światła:
 - + - P3V3, - - GND, OUT - PTA0 (kanał 12 ADC0).
- LCD1602A + ekspander:
 - SCL – PTB3, SDA – PTB4, VDD – P5V_USB.

3. Działanie aplikacji

a) Wstępne ustawienia

Przed rozpoczęciem korzystania z aplikacji należy odpowiednio ustawić czujnik. Światło docierające do czujnika powinno być na tyle duże, by napięcie na końcówce OUT było większe od wartości referencyjnej będącej granicą między włamaniem, a brakiem włamania. W kodzie programu domyślnie ustawiona jest ona na **1.0 V**. Dobrałem ją na podstawie pomiarów napięcia czujnika przy oświetleniu światłem dziennym. Napięcie referencyjne można zmienić poprzez zmianę wartości zmiennej globalnej **LIGHT_VALUE**, znajdującej się na samym początku pliku main.c (w komentarzu podane są przykładowe wartości, które używałem).

b) Instrukcja

Czynności w aplikacji wykonywane są przy pomocy 4 cyfrowych PIN-ów. Wpisany PIN można potwierdzić tylko wtedy, gdy zostaną wpisane 4 cyfry. **Potwierdzenia dokonuje się przy pomocy przycisku „#” (S13)**. W każdej chwili można **usunąć wpisany kod poprzez wciśnięcie przycisku „*” (S15)**. Powoduje to wyczyszczenie bufora ze wszystkich wpisanych znaków.

Aplikacja zawiera dwa tryby:

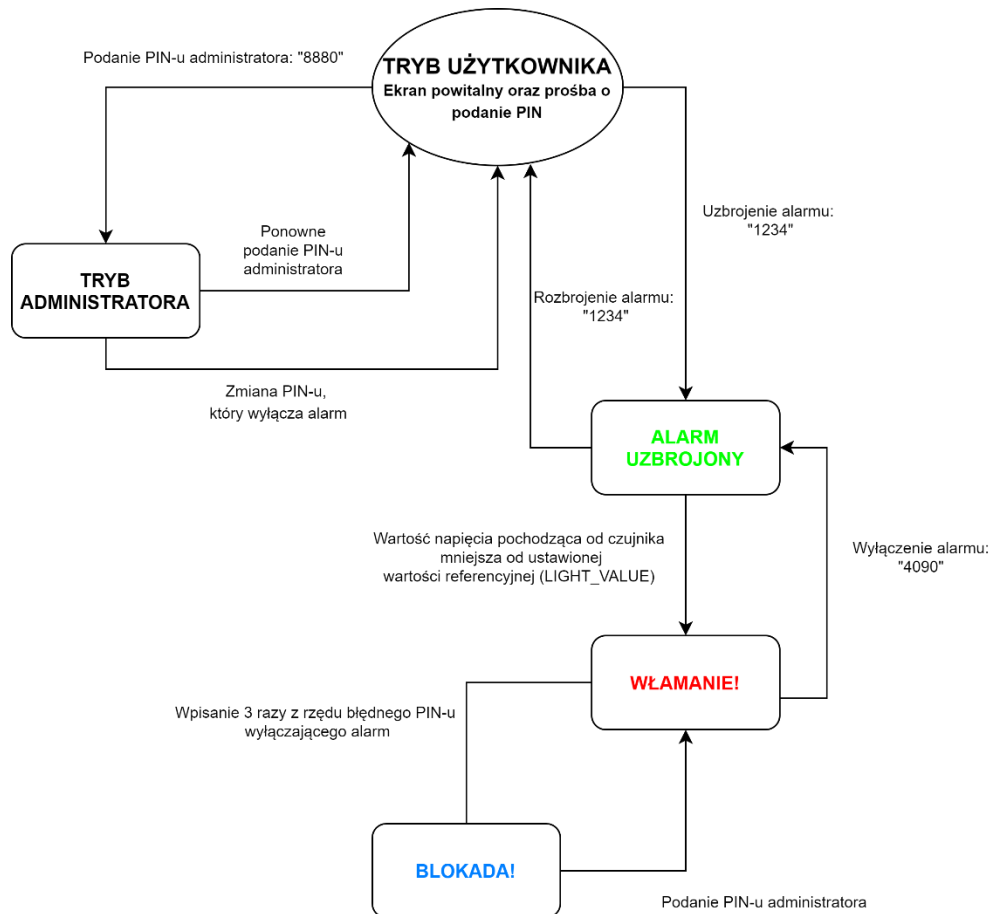
- **użytkownika:**
 - z tego trybu można przejść do trybu administratora (wpisanie kodu administratora) lub uzbroić alarm (wpisanie kodu uzbrajającego).
- **administratora**
 - w tym trybie można dokonać zmiany PIN-u wyłączającego uruchomiony alarm. Nowy PIN powinien różnić się od aktualnie używanych kodów. Ponowne podanie kodu administratora, zamiast nowego PIN-u powoduje powrót do trybu użytkownika,
 - uruchomienie tego trybu sygnalizowane jest **wszystkimi diodami LED**.

PIN-y ustawione domyślnie:

- „8880” – **kod administratora** – używany do uruchomienia trybu oraz wyłączenia blokady spowodowanej przekroczeniem dozwolonej ilości prób,
- „1234” – **kod uzbrajający/rozbrajający alarm**,
- „4090” – **kod wyłączający uruchomiony alarm** – ten kod może zostać zmieniony w trybie administratora.

Gdy alarm zostanie uzbrojony, zapala się **zielona dioda LED** oraz czujnik zaczyna dokonywać pomiarów światła. W momencie, gdy wartość napięcia otrzymanego od czujnika światła będzie mniejsza od ustawionej wartości referencyjnej, od razu zostanie uruchomiony alarm. Sygnalizowany on jest **czerwoną diodą LED**. Alarm zostanie wyłączony po podaniu poprawnego kodu. Należy pamiętać, że wyłączenie alarmu, nie powoduje zaprzestania dokonywania pomiaru przez czujnik światła. Oznacza to, że jeśli wyłączymy alarm i czujnik wykryje ruch, to alarm zostanie uruchomiony ponownie. Jeśli 3 razy z rzędu zostanie podany zły PIN, następuje blokada, która jest sygnalizowana **niebieską diodą LED**. Można ją wyłączyć wpisując kod administratora. Po odblokowaniu alarmu wciąż trzeba wyłączyć alarm odpowiednim PIN-em.

Opis działania w skróconej formie:



4. Pliki źródłowe

- **main.c** – główny plik,
- **keypad.h, keypad.c** – biblioteka z obsługą klawiatury,
- **utils.h** – zawiera definicje użytych PIN-ów oraz makra,
- **LEDs.h, LEDs.c** – obsługa diod RGB,
- **ADC.h, ADC.c** – obsługa ADC0,
- **pit.h, pit.c** – obsługa licznika PIT0,
- **i2c.h, i2c.c** – obsługa I2C,
- **lcd1602.h, lcd1602.c** – obsługa wyświetlacza. Zawiera stworzone przeze mnie funkcje wyświetlające stosowne komunikaty.