

Symulator alarmu przeciw włamaniowego

Założenia

Symulator alarmu wykrywa dźwięk.

Po uzbrojeniu alarmu użytkownik ma 10 sekund na wpisanie poprawnego kodu żeby rozbroić alarm.

Ewolucja założeń



Przełącznik



Czujnik ruchu

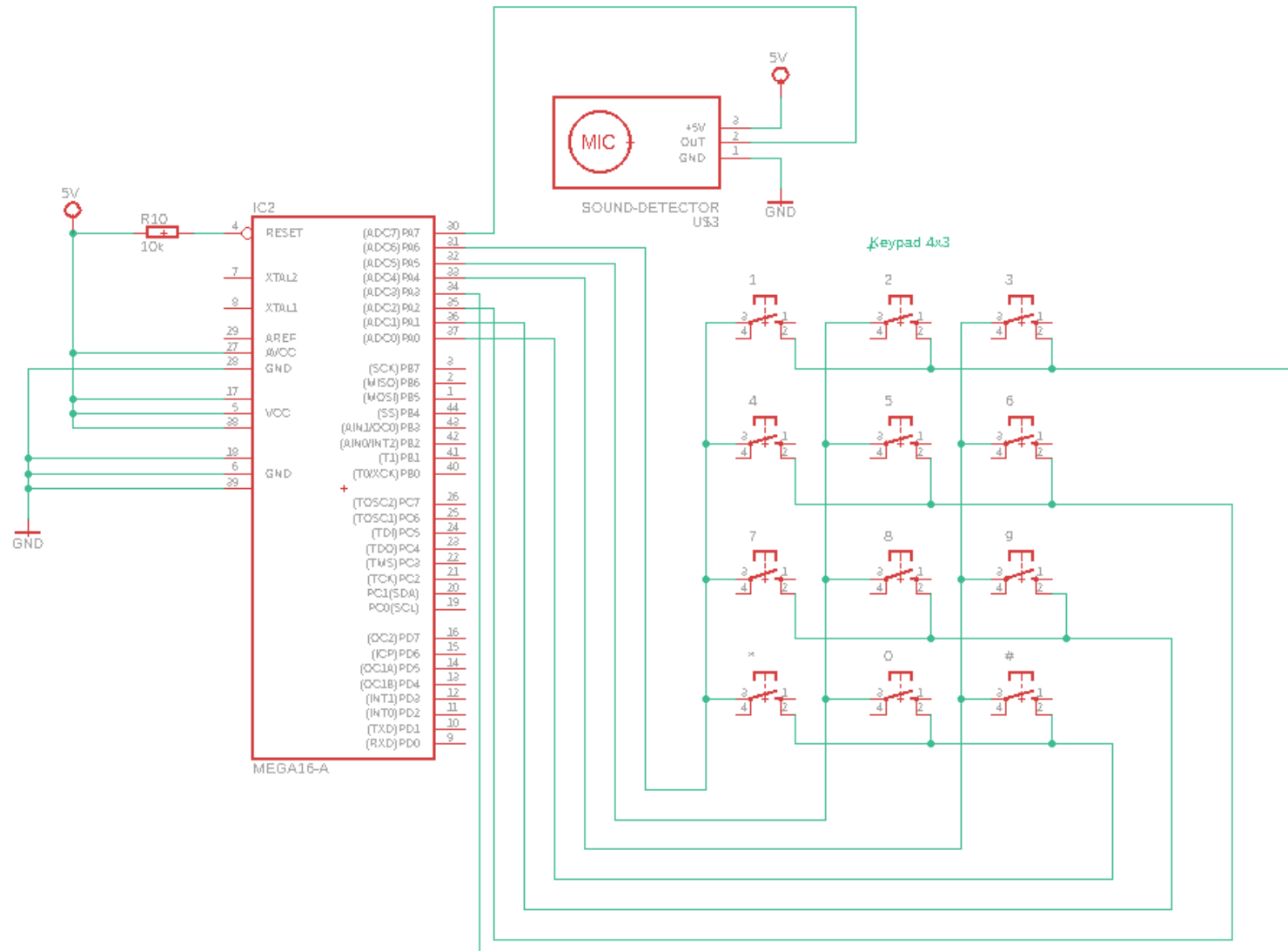


Syrena alarmowa

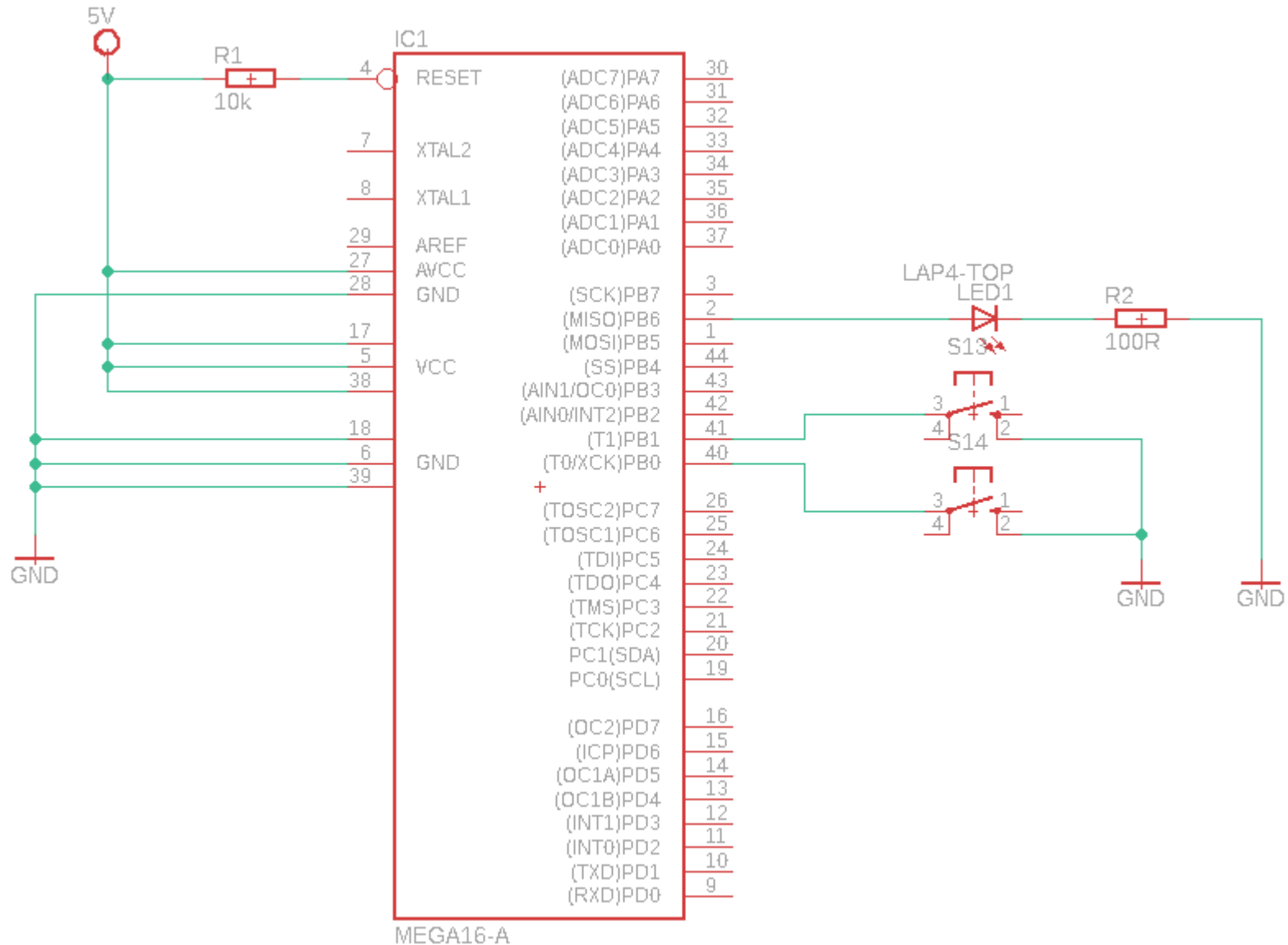


Buzzer

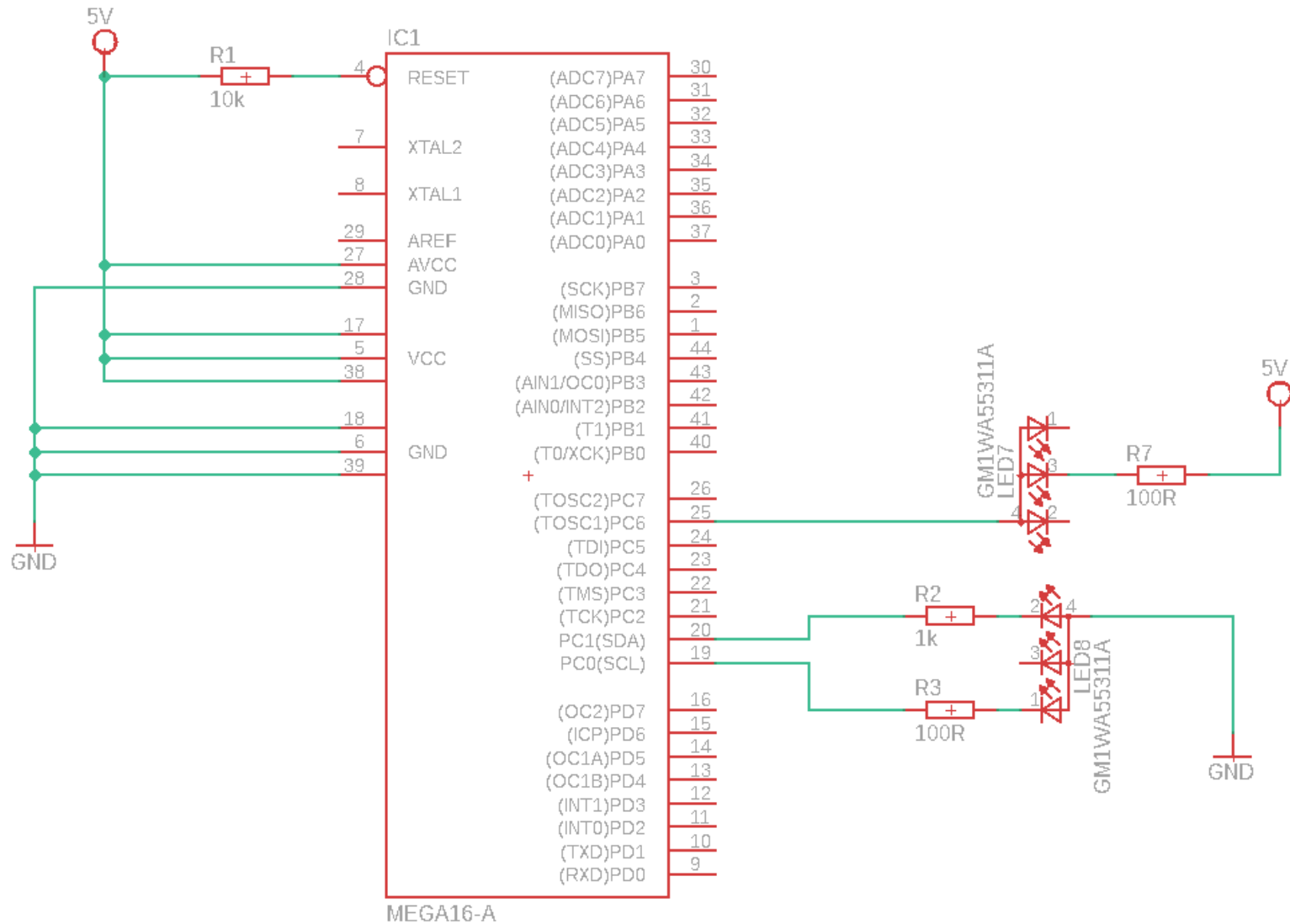
Schemat ideowy cz1



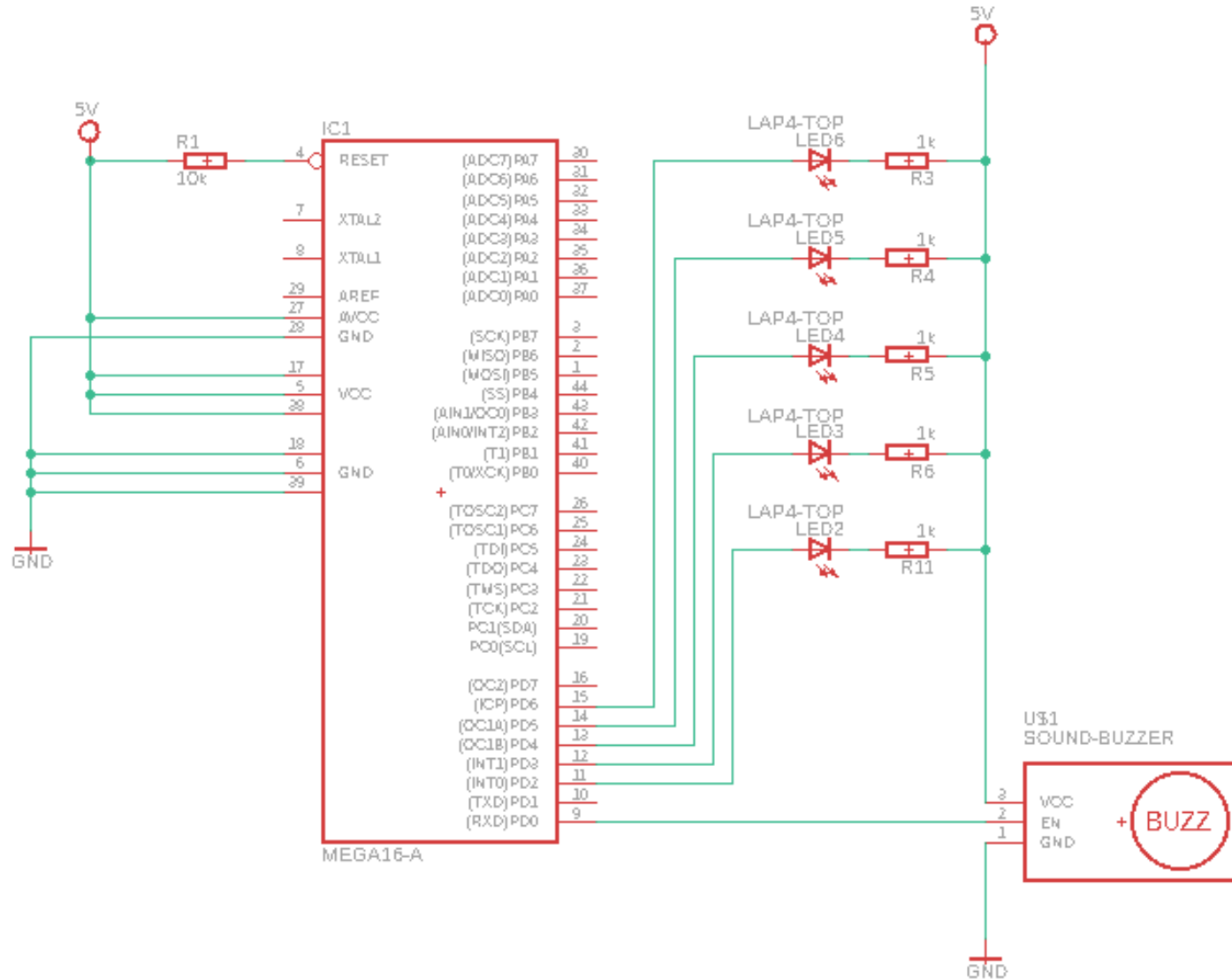
Schemat ideowy cz2



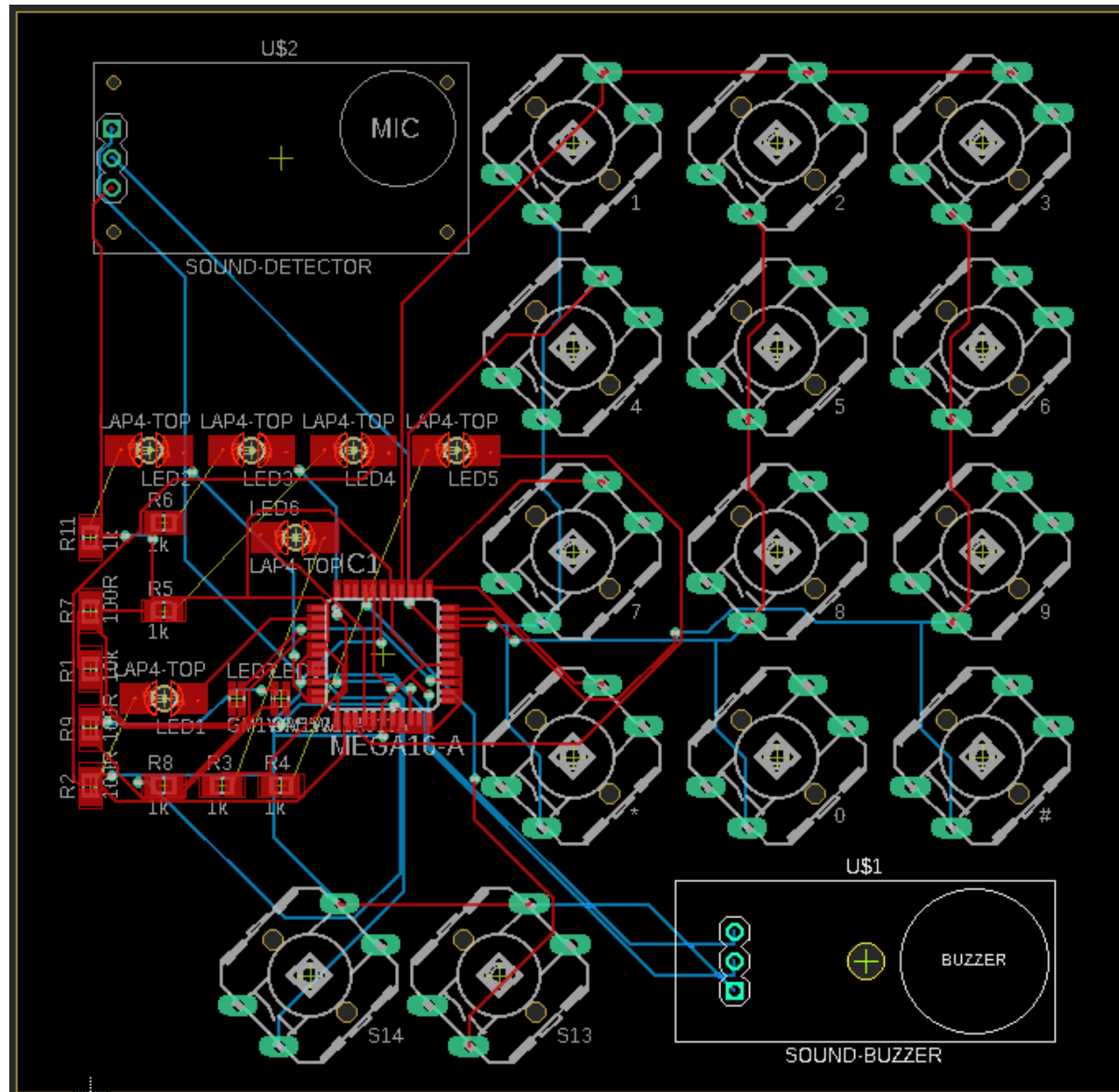
Schemat ideowy cz3



Schemat ideowy cz4



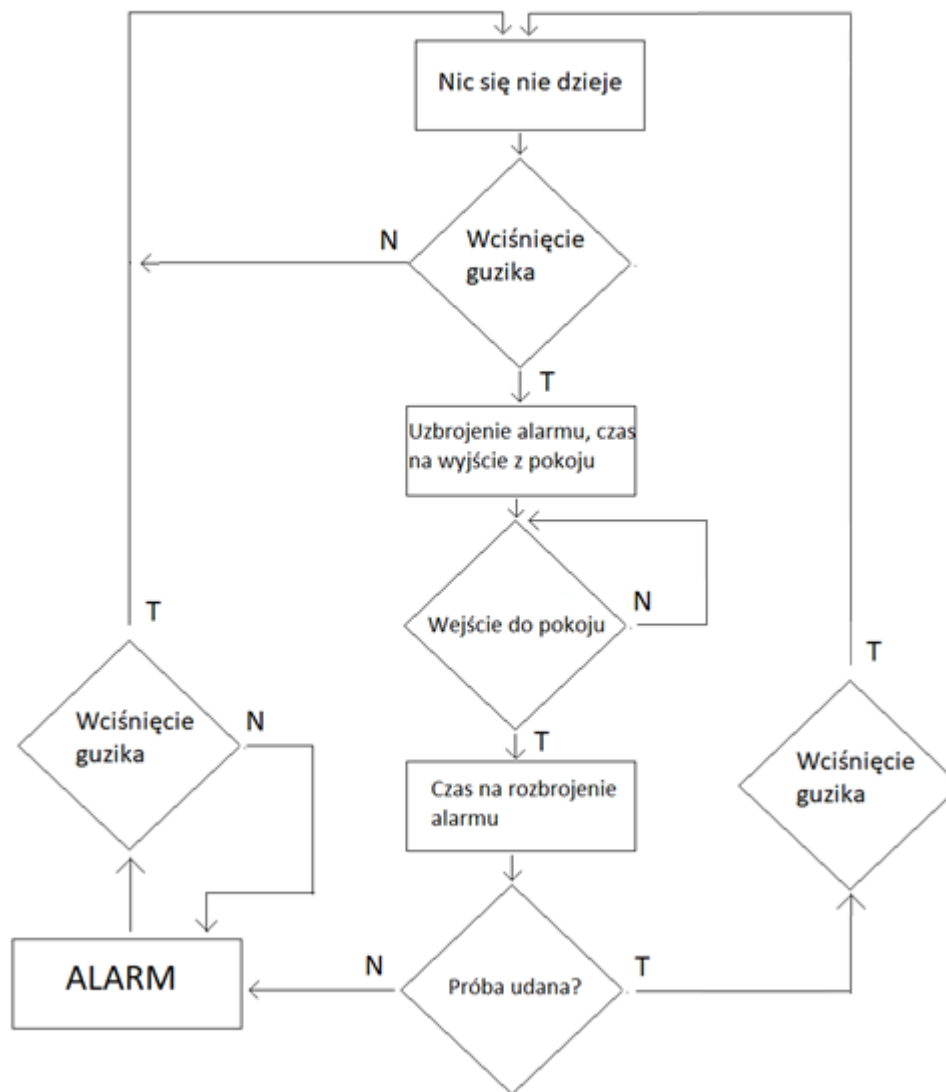
Płytki



Opis działania programu

Po włączeniu zasilania zapala się dioda zielona sygnalizująca zasilanie. Użytkownik jednym z przycisków uzbraja alarm. Po chwili czasu alarm jest gotowy do działania. Następnie gdy wykryje dźwięk zapalana jest dioda żółta. Użytkownik ma 10 sekund na wpisanie poprawnego kodu z klawiatury. Jeśli wpisze poprawny kod dioda zielona jest zapalana, w przeciwnym wypadku nastąpi sygnał dźwiękowy (buzzer) oraz świetlny (dioda czerwona).

Algorytm działania programu



Użyte elementy



PRZEWODY POŁĄCZENIOWE



KLAWIATURA MEMBRANOWA



PRZYCISKI



PŁYTKA STYKOWA



AVR ATMEGA 16



BUZZER



DIODY



CZUJNIK DŹWIĘKU



REZYSTORY



Zasilanie - 5V

Ciekawsze fragmenty kodu

Odczytanie kodu klawisza

```
uint8_t GetKeyPressed() {
    uint8_t r, c;

    PORTA |= 0x0F;

    for ( c = 0; c < 3; c++ ) {
        DDRA &= ~( 0x7F );
        DDRA |= ( 0x40 >> c );

        for ( r = 0; r < 4; r++ ) {
            if ( !( PINA & ( 0x08 >> r ) ) ) {
                return ( r*3 + c + 1); //zwrocenie kodu klawisza: 1->1, 2->2, ..., 10->'*', 11->0, 12->"#"
            }
        }
    }

    return 0xFF; //zwrocenie 255 gdy klawisz nie zostal wcisniety
}
```

Fragment funkcji getPassword()

```
if (timeOut() == 0) //czas sie nie skonczyl
{

    key = GetKeyPressed();
    if ( (key < 10 || key == 11) && (pos<4) )
    {
        switchOnDiode(pos);
        tab[pos] = key;
        pos++;
        _delay_ms( 50 );
    }
    else if (key == 10) //kasowanie znaku
    {
        if (pos != 0)
        {
            pos=pos-1;
            switchOffDiode(pos);
            _delay_ms( 50 );
        }
    }
    else if (pos == 4 && key==12) //zatwierdzenie kodu
    {
        enterCode = 1;
    }
}
```

Liczenie czasu wpisywania kodu

```
int timeOut()  
{  
    if (TCNT1 >= 15625) //15625 - jedna sekunda  
    {  
        TCNT1 = 0; //Zerowanie Timera  
        sec++;  
        if (sec >= 10)  
        {  
            return 1;  
        }  
    }  
    return 0;  
}
```

Pętla główna

```
int main( void ) {

    init(); // sekcja inicjalizacji peryferiów
    //zmienne sterujące prace programu
    int beep = 0;
    int alarm = 0;
    int end = 0;

    TCCR1B |= ((1 << CS10) | (1 << CS11)); //Ustawia timer z preskalerem Fcpu/64

    while ( 1 ) {

        if ( !( PINB & ( 1 << PB1 ) ) ) { //uzbrojenie alarmu
            _delay_ms( 200 ); //czas na uzbrojenie alarmu
            PORTC &= ~( 1 << PC6 ); //załączenie diody niebieskiej
            alarm = 0;
            while ( alarm == 0 ) {

                if ( !( PINA & ( 1 << PA7 ) ) ) { //jesli wystapi dzwiek
                    PORTC |= ( 1 << PC6 ); //wylaczenie diody niebieskiej -> koniec uzbrojenia alarmu

                    PORTD &= ~( 1 << PD2); //dioda zolta -> czas na wpisanie alarmu
                    getPassword(); //wpisanie hasla
                    PORTD |= ( 1 << PD2); //wylaczenie diody zoltej

                    if ( checkPassword() == 1 ) { //dobre haslo

                        PORTC |= (1<<PC0); //dioda zielona -> dobre haslo
                        end = 0;
                        while (end == 0)
                        {
                            if ( !( PINB & ( 1 << PB0 ) ) ) { // po naciśnięciu przycisku powrot do stanu początkowego
                                PORTC &= ~( 1<<PC0));
                                end=1;
                            }
                        }
                    }
                }
            }
        }
    }
}
```

```

    } else { //zle haslo
        PORTC |= 1<<PC1; //dioda czerwona
        beep = 1;
        PORTD &= ~( 1 << PD0 ); //właczenie buzzera

        while ( beep == 1 ) {

            if ( !( PINB & ( 1 << PB0 ) ) ) { //po naciśnięciu przycisku powrot do stanu początkowego

                PORTD |= ( 1 << PD0 );
                beep = 0;
                PORTC &= ~( 1<<PC1);

            }

        }
        alarm = 1;

    }

}

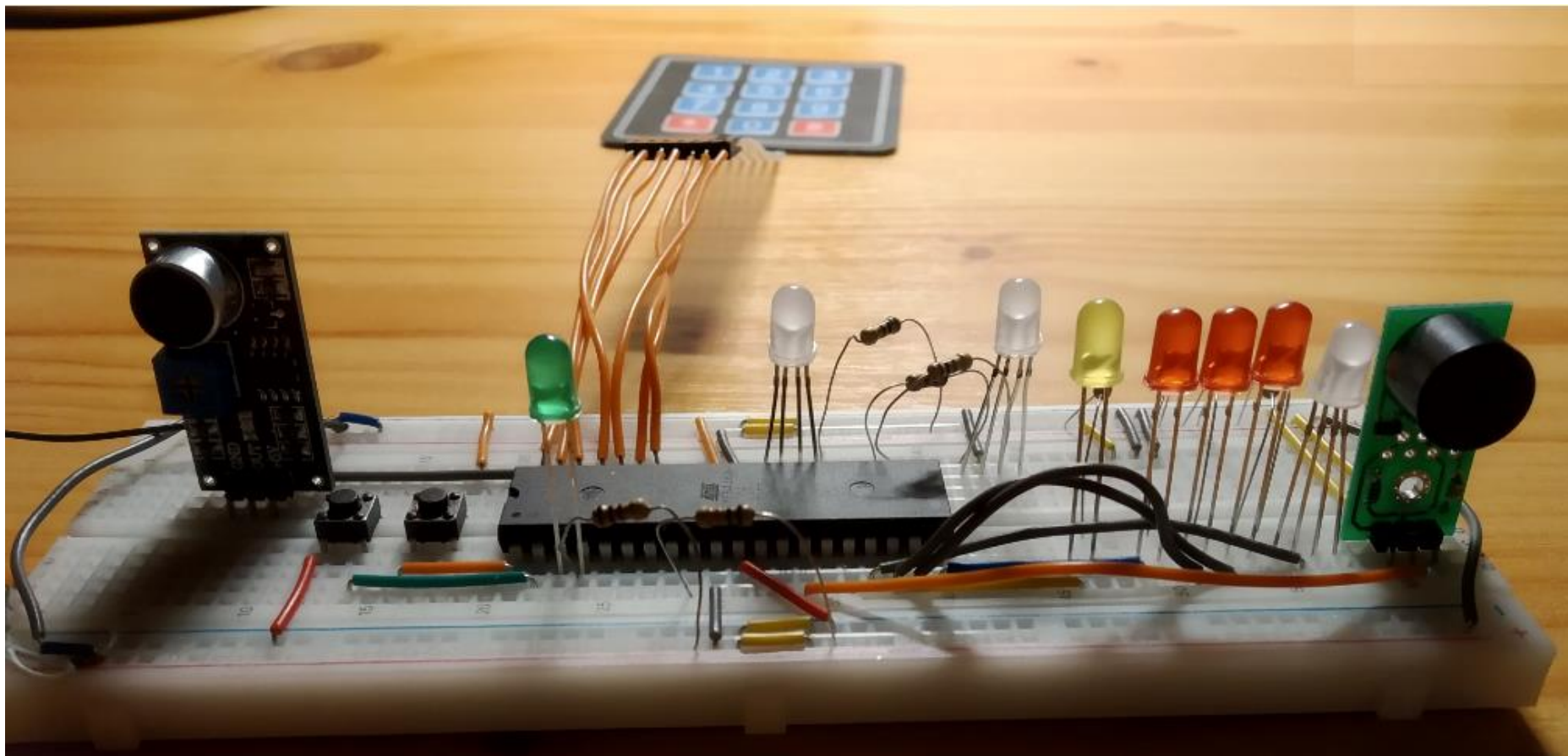
PORTC |= ( 1 << PC6 ); //wylaczenie diody niebieskiej

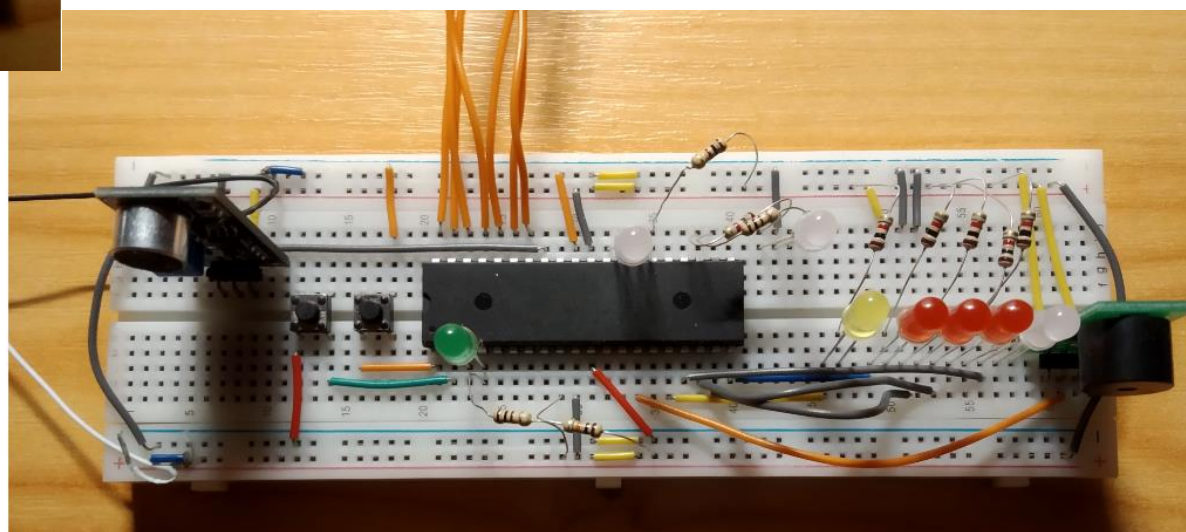
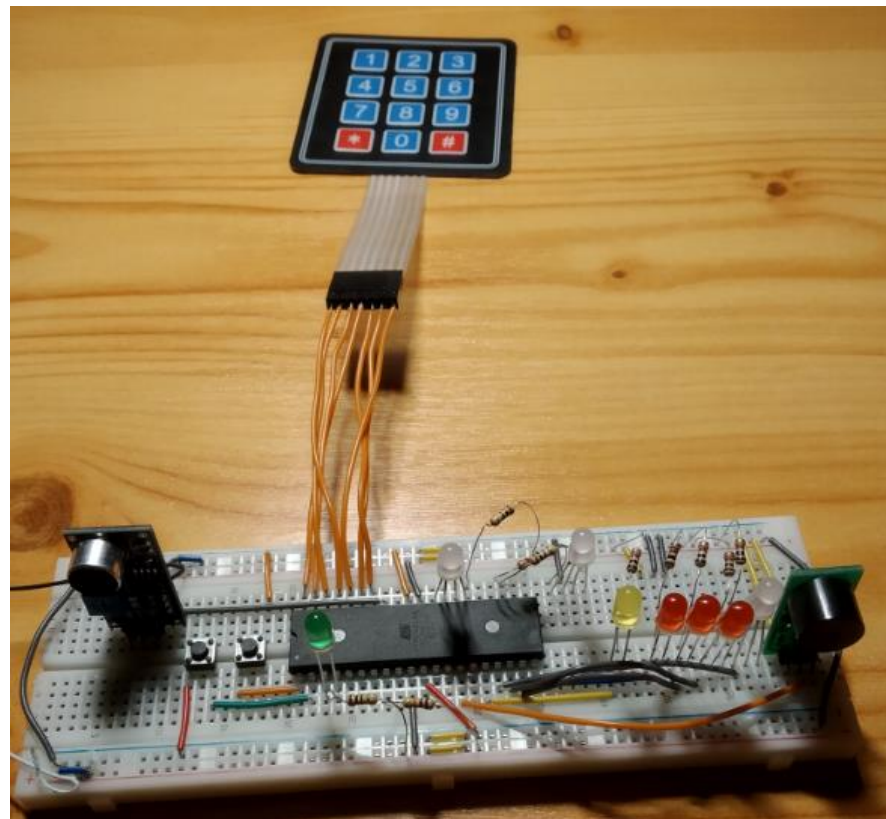
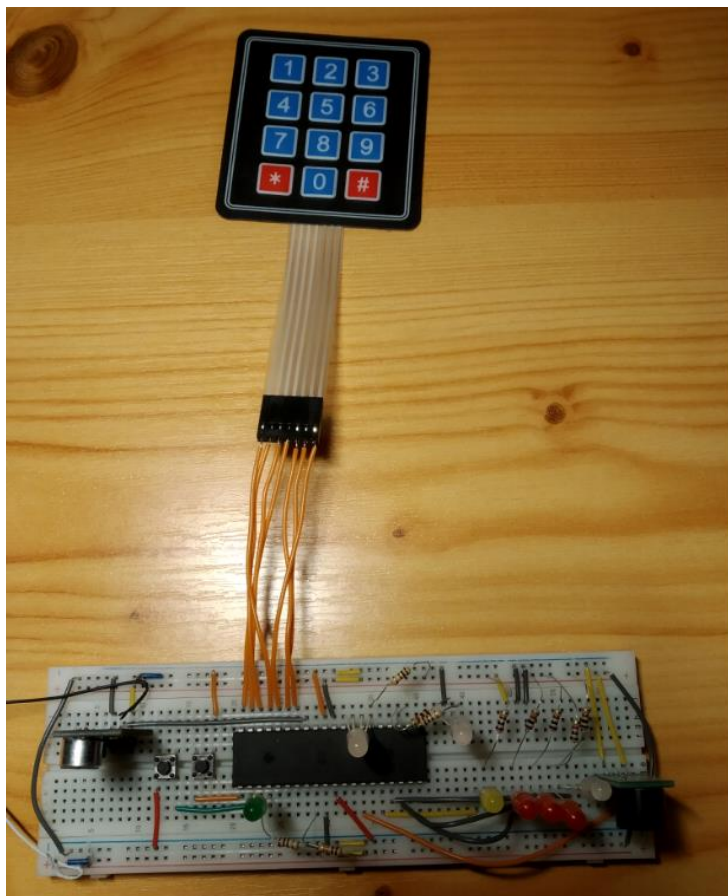
}

}

```


Zdjęcia





Bibliografia

- <https://botland.com.pl>
- <https://atnel.pl/>

Dziękuję za uwagę