

Niezbędne informacje do
wykonania ćwiczenia 1

Struktura programu

```
1 #define moj_port 9           //aliasy - nadawanie własnych nazw portom
2 #include <LiquidCrystal.h>   // dołączanie bibliotek np do obsługi wyświetlacza
3                               // <-----komentarze dla pojedynczego wiersza
4 /*
5  *                           Komentarze dla wielu wierszy
6  */
7
8 void setup() {
9     // instrukcje wykonywane jednokrotnie po resecie procesora
10
11 }
12
13 void loop() {
14     // instrukcje wykonywane w pętli po resecie procesora
15 }
16
17 void MojaProceduraWywolanaProgramowo()
18
19 {
20
21 }
```

Elementy programu

```
#define mojaNazwaPortu 22 // aliasy nadawanie własnych nazw portom
#define mojaNazwaIIPortu 23
#define mojaNazwaIIIPodciagnietegoPortu 24
```



Zwróć uwagę, że nie ma tutaj nawiasów i znaków „;”

```
6 //-----// instrukcje wykonywane jednokrotnie -----
7 void setup() {
8 // instrukcje wykonywane raz // <---komentarze w pojedynczym wierszu
9 /*
10 * komentarze obejmujące kilka wierszy
11 */
12 pinMode(mojaNazwaPortu, OUTPUT);
13 pinMode(mojaNazwaIIPortu, INPUT);
14 pinMode(mojaNazwaIIIPodciagnietegoPortu, INPUT_PULLUP); // Podciągnięcie portu do +5V
15 pinMode(2, INPUT_PULLUP);
16
17
18
19 pinMode(2, INPUT_PULLUP);
20 attachInterrupt(digitalPinToInterrupt(2), Przerwanie, FALLING); // deklaracja obsługi przerwań na porcie 0, procedura obsługi, FALLING - zbocze opadające
21 // FALLING - wyzwalane zboczem opadającym
22 // RISING - wyzwalane gdy stan wejścia zmienia się z LOW na HIGH
23 // CHANGE - wyzwalane gdy stan wejścia zmienia się na przeciwny
24 // LOW - wyzwalanie stanem niskim
25
26 interrupts(); //załączenie obsługi przerwań
27
28 digitalWrite(mojaNazwaPortu, LOW); // Ustawienie portu w stan niski 0 logiczne
29 digitalWrite(mojaNazwaPortu, HIGH); // Ustawienie portu w stan niski 1 logiczna
30 }
```

Pętla główna programu i podprocedury

```
31 //-----// główna pętla programu -----
32 void loop() {
33     // instrukcje wykonywane w pętli
34
35     noInterrupts();           // wyłączenie obsługi przerwań
36     delay (1000);             // opóźnienie wykonania kodu o 1 sekundę
37     if (digitalRead(mojaNazwaIIPortu) == 0) // instrukcja warunkowa oraz zczytanie stanu portu
38     {
39                                     // wykonanie instrukcji jeśli stan portu będzie 0
40     MojaProcedura();           // wywołanie procedury poza główną pętlą programu
41
42 }
43
44
45
46 }
47 //-----// Podprocedury i procedury obsługi przerwań-----
48 void Przerwanie()             //Procedura przerwania
49 }
50 }
51 |
52
53 void MojaProcedura() // procedura wywoływana w programie
54
55
56 {
57 }
```