

Temat: Funkcje bez argumentów

Do tej pory nasz kod umieszczaliśmy w funkcjach `setup() {}` lub `loop() {}`. Pierwsza z nich dotyczyła ustawień, a druga była nieskończoną pętlą główną, która wykonywała się cały czas. Przykładowo, aby migać diodą podłączoną do pinu nr 13 należało napisać taki program:

```
1 void setup() {
2   pinMode(13, OUTPUT); //Konfiguracja pinu 13 jako wyjście
3 }
4
5 void loop() {
6   digitalWrite(13, HIGH); //Włączenie diody
7   delay(1000); //Odczekanie 1 sekundy
8   digitalWrite(13, LOW); //Wyłączenie diody
9   delay(1000); //Odczekanie 1 sekundy
10 }
```

Wyobraźmy sobie sytuację, w której nasz program jest bardzo rozbudowany, a miganie chcemy wykorzystać jako potwierdzenie wybranych operacji. Szybko okaże się, że wielokrotne powielanie fragmentu odpowiedzialnego za włączanie i wyłączanie diody jest **zasochłonne dla programisty**. Co gorsza, sprawia to również, że cały **program jest znacznie trudniejszy w analizowaniu**.

Deklarując zmienną możemy odwoływać się do niej wielokrotnie w łatwy sposób. Gdybyśmy mogli zapisywać pod *łatwą nazwą* całe ciągi operacji, to programy byłyby zdecydowanie czytelniejsze. Również ewentualne zmiany byłyby łatwiejsze. Tutaj pomocne będą właśnie **funkcje**.

W powyższym przykładzie za miganie diody odpowiedzialny jest fragment od wiersza 6 do 9. Możemy go "wyciągnąć" poza `loop() {}` i stworzyć z niego osobną funkcję.

Na początku konieczne jest zadeklarowanie nazwy funkcji oraz jej typu. Robimy to np.: pod funkcją `loop() {}`

Dalej otwieramy **nawias klamrowy** i w jego wnętrzu umieszczamy kod, który ma się wykonać w momencie wywołania funkcji:

```
1 void setup() {
2   pinMode(13, OUTPUT); //Konfiguracja pinu 13 jako wyjście
3 }
4
5 void loop() {
6   zamigajLED();
7 }
8
9 void zamigajLED() {
10  digitalWrite(13, HIGH); //Włączenie diody
11  delay(500); //Odczekanie 0,5 sekundy
12  digitalWrite(13, LOW); //Wyłączenie diody
13  delay(500); //Odczekanie 0,5 sekundy
14 }
```

Zwróć uwagę, że w pętli loop() {} wpisujemy nazwę funkcji (już bez przedrostka void). Działanie takie nazywamy **wywołaniem funkcji**.

Ćwiczenie:

Wykorzystując własny zestaw arduino przećwicz powyższy sposób deklaracji i wywołania funkcji. W przypadku braku zestaw odwiedź stronę <https://forbot.pl/blog/kurs-arduino-czujnik-odleglosci-hc-sr04-funkcje-id4290> i obejrzyj efekt działania tego programu.