

8. Arduino jako sterownik

Instrukcja warunkowa **while**

Instrukcja warunkowa **while** jest wykonywana tak długo, jak długo jest spełniony jest **warunek**.

```
while(warunek){  
    //wykonywane, gdy spełniony warunek  
}
```

Instrukcja warunkowa **for**

Instrukcja **for** służy do powtarzania bloku instrukcji ujętych w nawiasy klamrowe. Licznik (zwyczajowo zmienna *i*) jest zwykle zwiększany (*i++*) lub zmniejszany (*i--*) o jeden i używany do końca pętli.

```
for(int i = 0; warunek; i++){ //początkowa wartość i wynosi 0.  
    //wykonywane, gdy spełniony warunek  
    //i++ powoduje każdorazowe zwiększenie i o jeden po wykonaniu bloku instrukcji  
}
```

Nowy tryb (**pinMode**): **INPUT_PULLUP**

Dotychczas poznane tryby **INPUT** oraz **OUTPUT** ustawiały port arduino jako wejście lub wyjście. Istnieje jeszcze jeden tryb – **INPUT_PULLUP**, tryb ten pozwala na użycie przycisku bez dodatkowych elementów (w ten sposób podłączone są przyciski w poniższych modelach). Po ustawieniu odpowiednio portu (**pinMode**(port, **INPUT_PULLUP**);), przycisk ten, gdy wciśnięty, będzie podawał na port 0V, w przeciwnym wypadku będzie tam się pojawiało 5V.

Nowy typ zmiennej: **float**

Dotychczas stosowane były zmienne typu int, zmienne int mogą przechowywać wartość od -32768 do 32767. Oznacza to, że próba zapisania wartości np. temperatury 32.7 °C spowoduje jej *obcięcie* do postaci 32. Do zapisu wartości zmiennoprzecinkowej może posłużyć zmienna typu float.

Zadanie 8.1 (wersja na symulator, gdy niedostępny sprzęt fizyczny)

Przy użyciu instrukcji *while*, *if* oraz *delay* stworzyć cyfrową wersję pięciociennej kości do gry:

- po wciśnięciu przycisku rozpoczyna się losowanie,
- losowanie polega na stworzeniu zmiennej reprezentującej ilość „oczek”, a następnie dodaniu do niej „losową” ilość razy 1,
- po każdym zwiększeniu zmiennej należy sprawdzić czy nie jest większa od ilości „oczek” (ilości diod). Jeśli liczba jest większa, należy od niej odjąć ilość „oczek”. Następnie wyświetlić (zapalić odpowiednią diodę) wynik i poczekać np. 10 ms.

Zadanie 8.1 (wyłącznie na zajęcia)

Do wyboru, jeden z modeli:

Termostat

Termostat składa się z termometru, dwóch przycisków oraz dwóch przełączników sterujących wentylatorem oraz grzałką.

- Z menu należy wybrać **Plik** → **Przykłady** → **WTD** → **Termostat**.
- Przeanalizować program i zweryfikować.
- Wprowadzić zmienną np. *limit* o wartości 30 °C oraz zaprogramować arduino tak, by uruchamiało grzanie poniżej wartości *limit* oraz chłodzenie powyżej.
- Zmodyfikować powyższe w taki sposób, że przyciski (porty 11 i 12, należy użyć trybu INPUT_PULLUP) pozwalają na zmianę wartości *limit*. Zmiana limitu powinna być wyświetlana na komputerze. Limit nie może przyjąć wartości wyższej niż 35 °C.

Światła drogowe

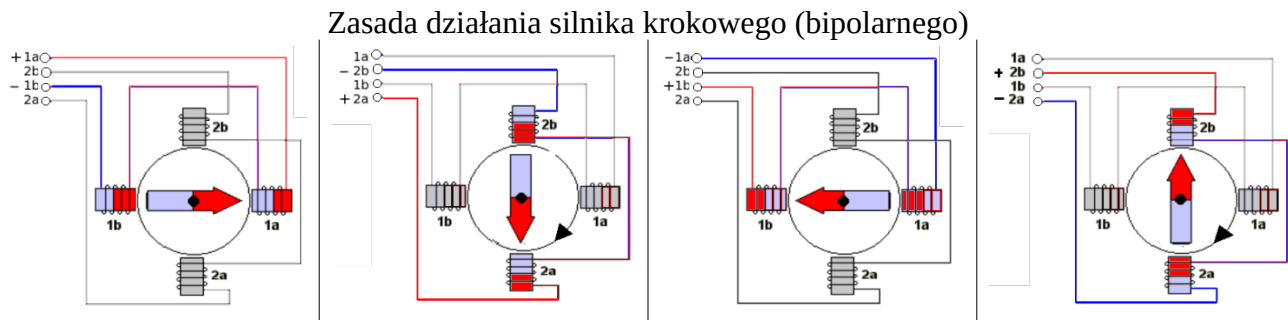
Model świateł drogowych składa się z dwóch przycisków oraz odpowiedniej ilości diod. Przyciski podłączone są do portów A0 oraz A1. Diody podłączone są do portów 3-12.

Przyciski należy ustawić jako INPUT_PULLUP, czyli np. pinMode(left, INPUT_PULLUP);

- Napisać program, który zapalał po kolei pojedyncze diody. Ustalić do którego portu podłączone jest które światło.
- Napisać program, który realizuje typową sekwencję świateł drogowych.
- **Dodatkowe:** Zmodyfikować poprzedni program o przyciski dla pieszych (skracające czas oczekiwania na zielone światło).

SILNIKI KROKOWE (przypomnienie)

Silnik krokowy – silnik elektryczny, w którym impulsowe zasilanie prądem elektrycznym powoduje, że jego wirnik wykonuje za każdym razem ruch obrotowy o ściśle ustalony kąt. Kąt obrotu, zależnie od budowy silnika – jest to zwykle wartość od kilku do kilkudziesięciu stopni.



CNC

Model frezarki CNC składa się z dwóch silników krokowych oraz lasera.

- Z menu należy wybrać **Plik** → **Przykłady** → **WTD** → **CNC**. Przeanalizować program oraz wgrać go i zweryfikować obserwacje. Następnie rozbudować program o zerowanie maszyny po uruchomieniu arduino. Czy dałoby się to zrobić *ładniej* (poprzez rozbudowę układu)?
- Zmodyfikować program w taki sposób, by wykonywał kwadrat. Laser ma być włączony tylko podczas wykonywania ruchu w osi X.
- Wykorzystując pętlę **while** lub **for** wykonać ruch w linii prostej pod kątem 45°.
- Utworzyć nowy program (lub zmodyfikować poprzedni), wykonujący kwadrat obrócony o 45°. Laser ma być włączony tylko dla dwóch boków.

Źródła:

Autodesk Eagle, Version 9.3.2, © 2019 Autodesk, Inc. All rights reserved
electronics.stackexchange.com/questions/70643/how-to-reverse-rotation-direction-of-stepper-motor
[instructables.com/id/How-to-use-the-L298-Motor-Driver-Module-Arduino-Tu/](https://www.instructables.com/id/How-to-use-the-L298-Motor-Driver-Module-Arduino-Tu/)
arduino.cc/reference/en/language/structure/control-structure/for/

Licencja MIT
Patryk Król
v3.2