

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.

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

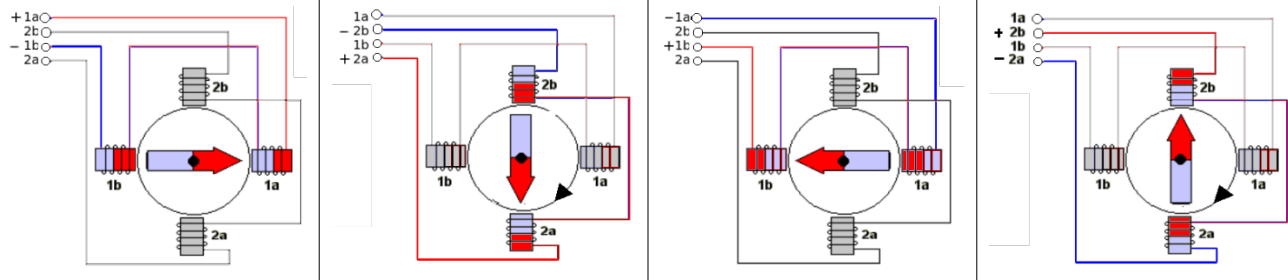
Przy użyciu instrukcji *while*, *if* oraz *delay* stworzyć cyfrową wersję pięciokątnej kości do gry:

- 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.

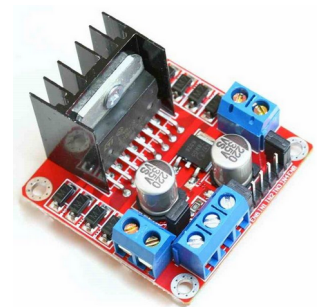
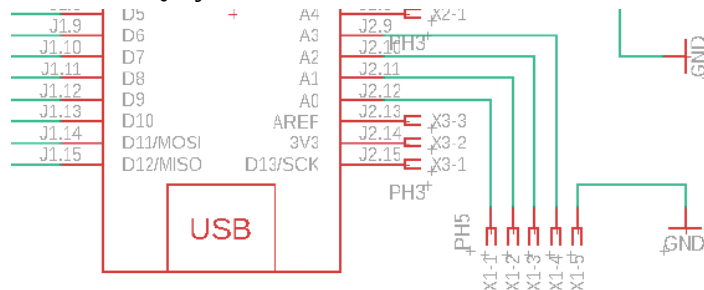
SILNIKI KROKOWE (przypomnienie)

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.

Zasada działania silnika krokowego (bipolarnego)



Zadanie 8.1 Podłączyć mostek H wraz z silnikiem do PH5.



Po uruchomieniu Arduino IDE wybrać przykładowy szkic: **Plik** → **Przykłady** → **Stepper** → **OneRevolution**. W programie zmienić deklarację `Stepper myStepper(stepsPerRevolution, 8, 9, 10, 11);` na `Stepper myStepper(stepsPerRevolution, A0, A2, A1, A3);` oraz `myStepper.setSpeed(60);` na `myStepper.setSpeed(10);`.

Przeanalizować program oraz zweryfikować obserwacje. Następnie napisać taki, który po wciśnięciu przycisku „lewego”, spowoduje ruch o jeden krok w lewo, natomiast po wciśnięciu przycisku „prawego”, ruch w prawo. Następnie, zadeklarować zmienną, która będzie przechowywała obecną ilość kroków jaką przebył silnik i wyświetlać ją po każdym wywołanym kroku (wciśnięciu przycisku).

Jeśli zostanie wciśnięty trzeci przycisk silnik ma powrócić do położenia początkowego (wykonać odpowiednią ilość kroków w odpowiednim kierunku). Jeśli silnik znajduje się w położeniu zerowym, włączyć diodę.

Zadanie 8.2 (wyłącznie na zajęcia)

Do wyboru, jeden z modeli:

CNC

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

a) 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)?

b) Zmodyfikować program w taki sposób, by wykonywał kwadrat. Laser ma być włączony tylko podczas wykonywania ruchu w osi X.

c) Wykorzystując pętlę **while** lub **for** wykonać ruch w linii prostej pod kątem 45°.

d) Utworzyć nowy program (lub zmodyfikować poprzedni), wykonujący kwadrat obrócony o 45°. Laser ma być włączony tylko dla dwóch boków.

Termostat

Termostat składa się z wcześniej poznanego termometru podłączonego do portu A3 (DS18B20, biblioteki **OneWire** oraz **DallasTemperature**) dwóch przycisków (porty 11 i 12) oraz dwóch przełączników. Do przełączników podłączone są: wentylator (przełącznik podłączony do portu A0) oraz grzałka (opornik 12Ω; przełącznik podłączony do A1).

- Napisać program wysyłający do komputera temperaturę odczytaną przez termometr.
- Do powyższego dodać funkcję ręcznego uruchamiania grzania oraz chłodzenia. Uniemożliwić uruchamianie grzania w przypadku gdy temperatura przekroczy 35 °C. Wysyłać na komputer informację o aktualnym stanie chłodzenia oraz grzania.
- Zmodyfikować powyższe o automatyczne grzanie – zadeklarować zmienną t o wartości 30. Jeśli temperatura na termometrze przekroczy wartość t , uruchamiane jest chłodzenie. Jeśli temperatura znajduje się poniżej t uruchamiane jest grzanie.
- Zmodyfikować powyższe w taki sposób, że przyciski pozwalają na zmianę wartości t . Uwaga! Temperatura nadal nie może przekroczyć 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(ileft, 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).

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/
arduino.cc/reference/en/language/structure/control-structure/for/

Licencja MIT
Patrik Król
v3.2 (pre)