

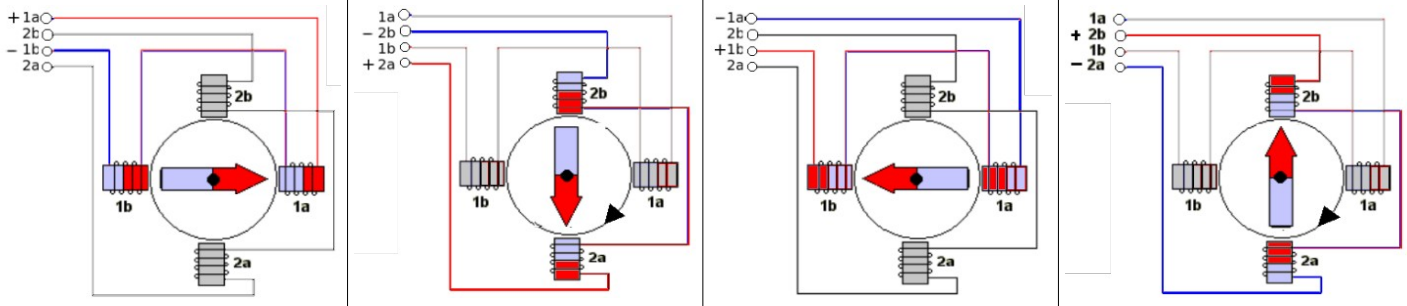
# A3. Arduino jako urządzenie sterujące

**PRZED ZŁOŻENIEM I PODŁĄCZENIEM PONIŻSZYCH UKŁADÓW NALEŻY POPROSIĆ O WERYFIKACJĘ.**

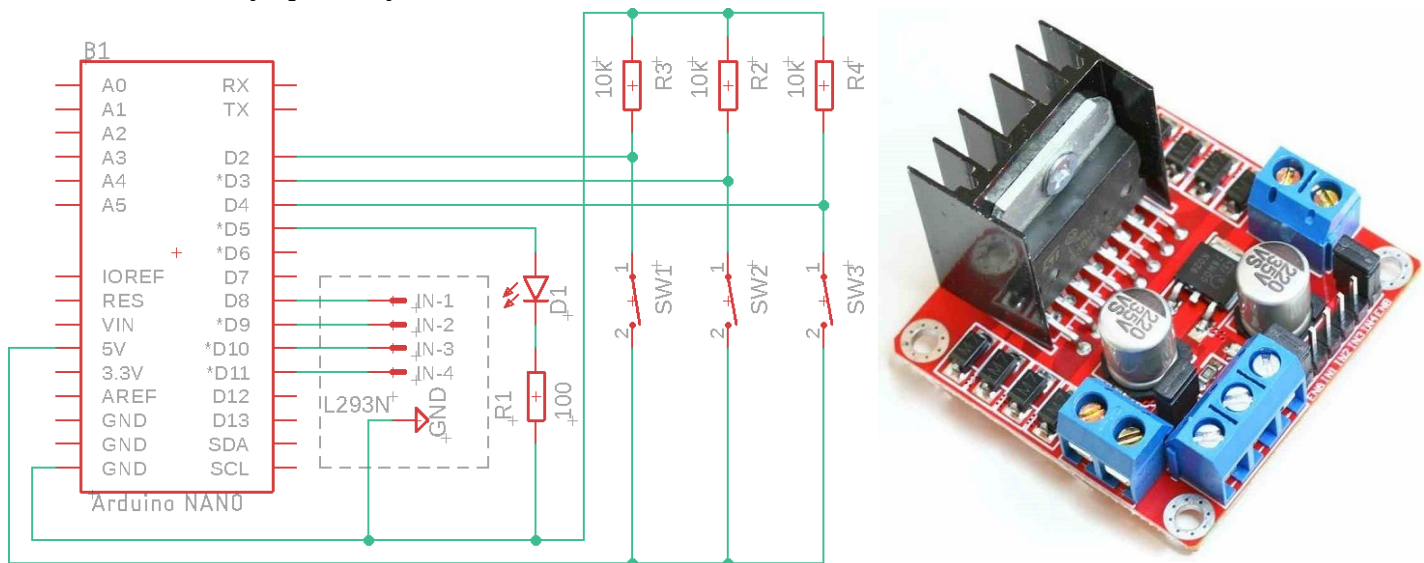
## SILNIKI KROKOWE

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 A3.1** Złożyć poniższy układ.



L293N

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, 8, 10, 9, 11);` oraz `myStepper.setSpeed(60);` na `myStepper.setSpeed(10);`.

Przeanalizować program oraz zweryfikować obserwacje. Następnie napisać taki, który po wciśnięciu przycisku 1, spowoduje ruch o jeden krok w lewo, natomiast po wciśnięciu przycisku 2, 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 zerowy, włączyć diodę.

## Instrukcja warunkowa `while`

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

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

## 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 A3.2

Dostępne są trzy modele na zajęciach (w kolejności odzwierciedlającej poziom trudności): światła drogowe, wózek oraz termostat. Należy wykonać przynajmniej jeden z projektów.

## Światła drogowe

Model światła 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 światła drogowych.
- **Dodatkowe:** Zmodyfikować poprzedni program o przyciski dla pieszych (skracające czas oczekiwania na zielone światło).

## Wózek

Model wózka składa się z dwóch czujników szczelinowych (porty 9 i 10), dwóch przycisków (porty 2 i 3) oraz układu umożliwiającego sterowanie silnikiem. Czujniki wysyłają logiczne 1, jeśli wykryją przedmiot. Przyciski wysyłają logiczne 0 jeśli są wciśnięte.

Czujniki i przyciski należy ustawić jako `INPUT_PULLUP`, czyli np. `pinMode(ileft, INPUT_PULLUP)`;

Układ sterujący silnikiem podłączony jest do portów A0 oraz A1 w taki sposób, że:

jeśli na port A0 podamy stan wysoki, to na pierwszej nodze silnika pojawi się napięcie,

jeśli na port A0 podamy stan niski, to na pierwszej nodze silnika pojawi się 0V.

Podobnie w przypadku portu A1 oraz drugiej nogi silnika.

- Napisać program wysyłający na port szeregowy stan przycisków oraz czujników szczelinowych.
- Napisać program, który, jeśli zostanie wciśnięty przycisk 1, uruchomi ruch wózka w lewo na 1 s., jeśli przycisk 2, ruch wózka w prawo na 1 s. Następnie zmodyfikować program w taki sposób, że wózek zatrzymuje się nie po sekundzie, ale w momencie pojawienia się sygnału na odpowiednim czujniku szczelinowym.
- Napisać program, który po wciśnięciu któregoś z przycisków, uruchamia ruch wózka w prawo, a po najechaniu na prawą krańcówkę (czujnik szczelinowy), wraca do krańcówki lewej, gdzie pozostaje do ponownego wciśnięcia przycisku.

## 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 przekaźników. Do przekaźników podłączone są: wentylator (przekaźnik podłączony do portu A0) oraz grzałka (opornik 12Ω; przekaźnik podłączony do A1).

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

- 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.

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  
www.instructables.com/id/How-to-use-the-L298-Motor-Driver-Module-Arduino-Tu/

Patryk Król  
v1.2