

## Krokový motor + driver

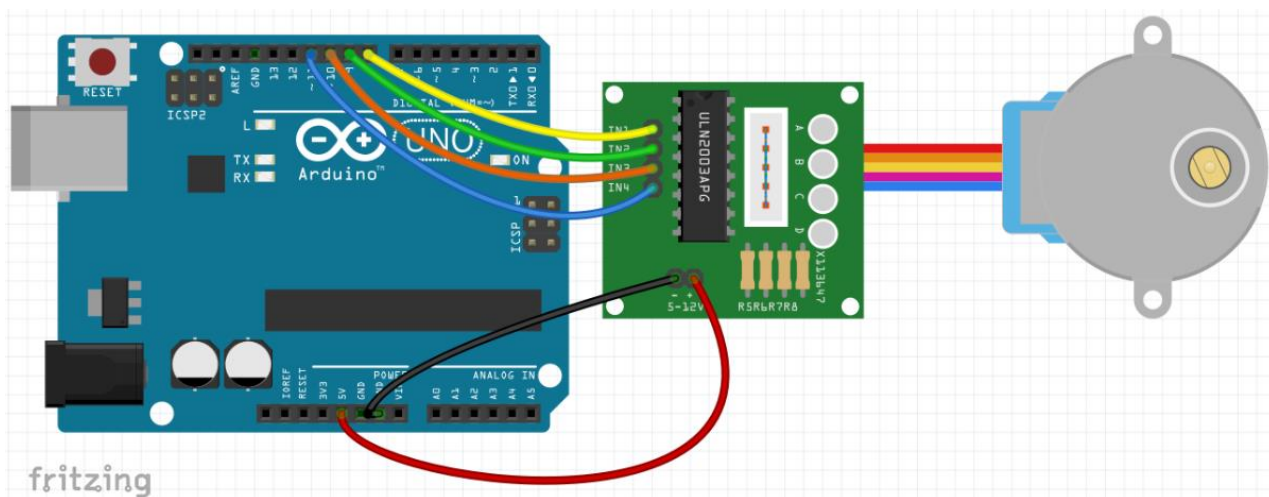
### 1. POPIS A SPECIFIKACE

Modul krokového motorku s budičem. Lze použít s vývojovými kity Arduino/Genuino, Raspberry PI a spoustou dalších. Obsahuje 4 indikační LED diody.

Hlavní čip	ULN2003
Napájecí napětí	5 VDC
Max. provozní proud	500 mA
Motor	28BYJ-48
Převodový poměr	1:64
Úhel kroku	5,625/64
Průměr šasi	28 mm
Rozměry modulu (mm)	35 x 30 x 10
Délka kabelu	23 cm



### 2. ZAPOJENÍ



00101  
01001  
00001

## 3. UKÁZKA PROGRAMU

Zapojení a ukázka programu převzata z: <http://navody.arduino-shop.cz/navody-k-produktum/krokovy-motor-a-driver.html>

```
// Krokový motor a driver

// čísla pinů pro digitální výstupy
const int in1 = 8;
const int in2 = 9;
const int in3 = 10;
const int in4 = 11;

// proměnná pro nastavení rychlosti,
// se zvětšujícím se číslem se rychlost zmenšuje
int rychlost = 1;
//
int uhel = 360;

void setup() {
  // inicializace digitálních výstupů
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {
  // plná rotace o 360 stupňů = 512 volání
  // funkce rotacePoSmeru() či rotaceProtiSmeru()
  for(int i=0;i<(uhel*64/45);i++){
    rotacePoSmeru();
  }
  // pauza po dobu 1 vteřiny
  delay(1000);

  for(int i=0;i<(uhel*64/45);i++){
    rotaceProtiSmeru();
  }
  // pauza po dobu 1 vteřiny
  delay(1000);
}
// zde následují funkce pro volání jednotlivých
// kroků pro otočení po či proti směru hodinových
// ručiček
void rotacePoSmeru() {
  krok1();
  krok2();
  krok3();
  krok4();
  krok5();
  krok6();
  krok7();
  krok8();
}
void rotaceProtiSmeru() {
  krok8();
  krok7();
  krok6();
  krok5();
  krok4();
  krok3();
  krok2();
  krok1();
}
// každý krok obsahuje výrobcem dané pořadí
// pro správné spínání motoru a následnou
// pauzu, kterou určujeme rychlost otáčení
void krok1() {
```

```

digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(rychlost);
}
void krok2() {
digitalWrite(in1, HIGH);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(rychlost);
}
void krok3() {
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, LOW);
digitalWrite(in4, LOW);
delay(rychlost);
}
void krok4() {
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(rychlost);
}
void krok5() {
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, LOW);
delay(rychlost);
}
void krok6() {
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, HIGH);
digitalWrite(in4, HIGH);
delay(rychlost);
}
void krok7() {
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(rychlost);
}
void krok8() {
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
digitalWrite(in3, LOW);
digitalWrite(in4, HIGH);
delay(rychlost);
}
}

```