

Programowanie systemów autonomicznych

Blok obieralny: Inteligentne Systemy Autonomiczne



Instytut Informatyki Stosowanej
Stefanowskiego 18/22
al. Politechniki 11

Tomasz Jaworski
Piotr Duch

Platforma jezdna

Platforma jezdna

Parametry:

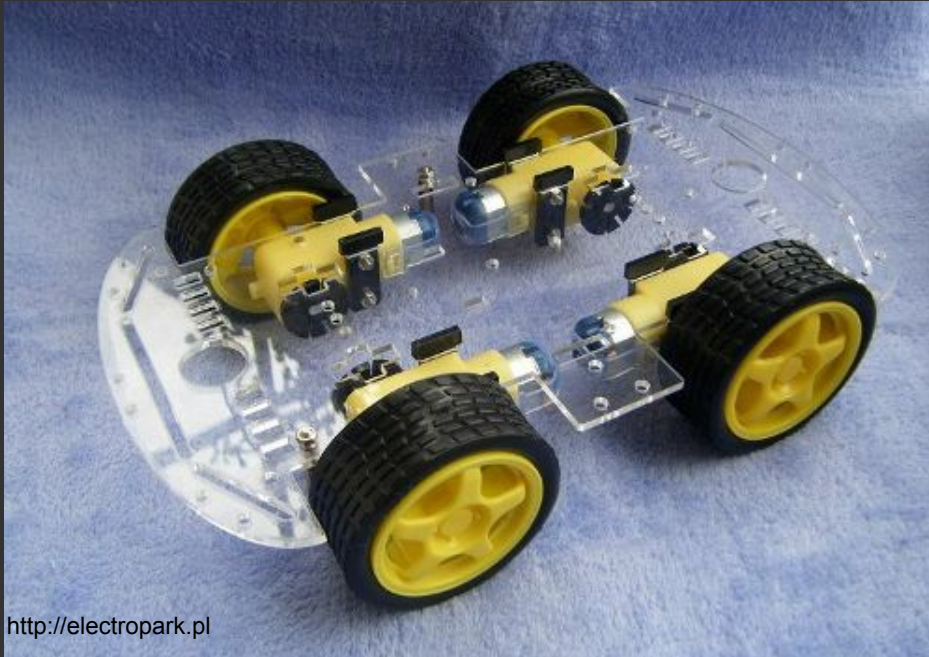
Liczba silników: 4 zredukowane na 2

Zasilanie: 4-9 V

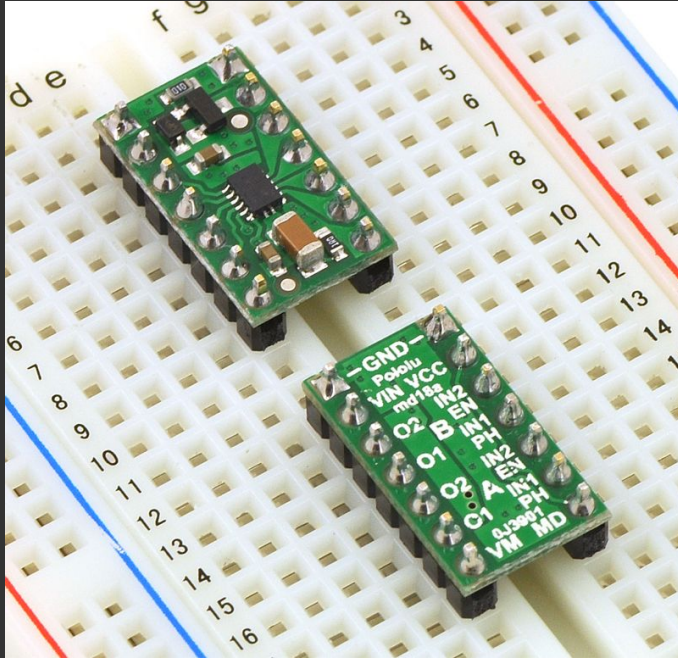
Sterowanie: Mostek H

Pomiar prędkości: enkodery Halla

Pobór prądu: do 200mA przy obciążeniu

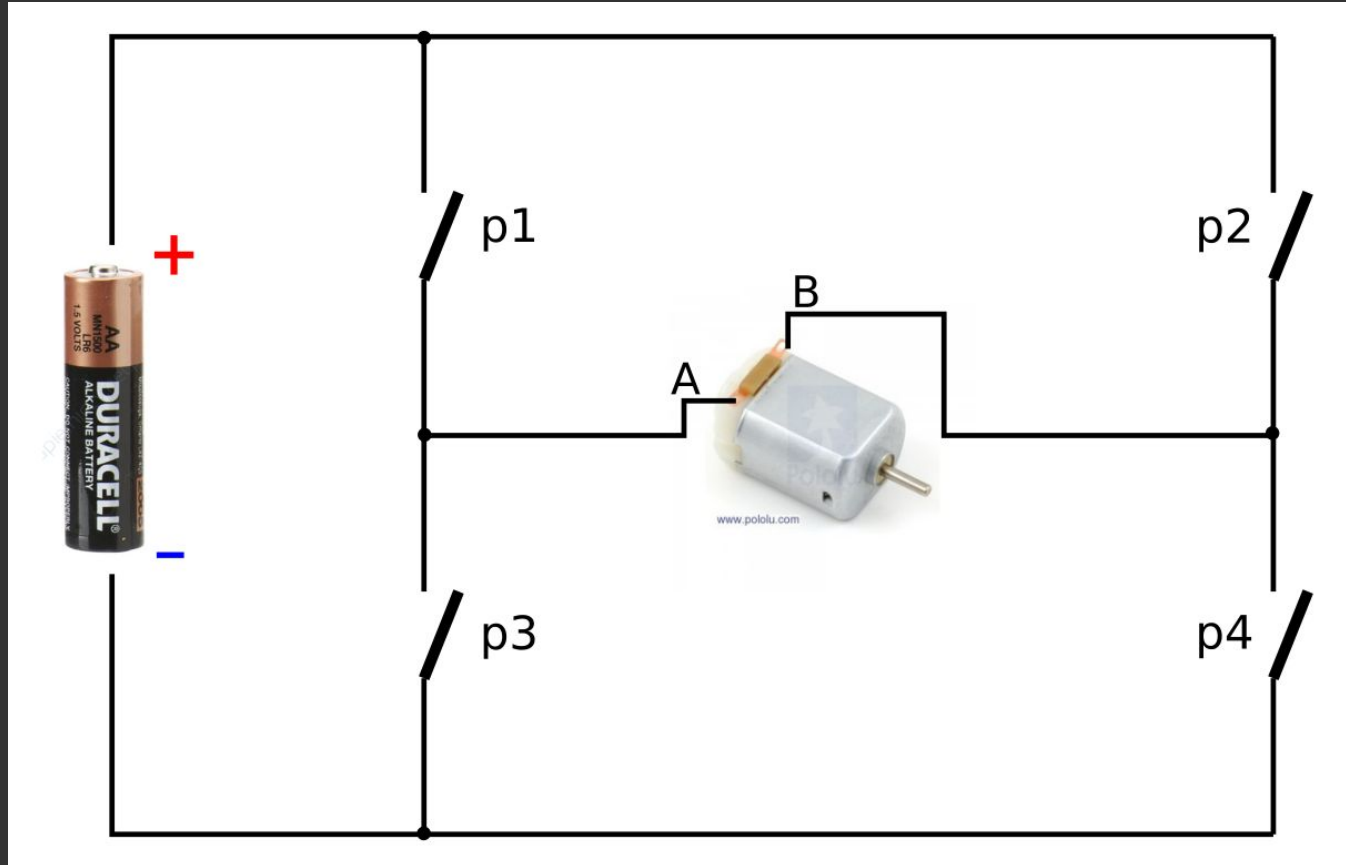


Platforma jezdna - elementy sterująco - kontrolne

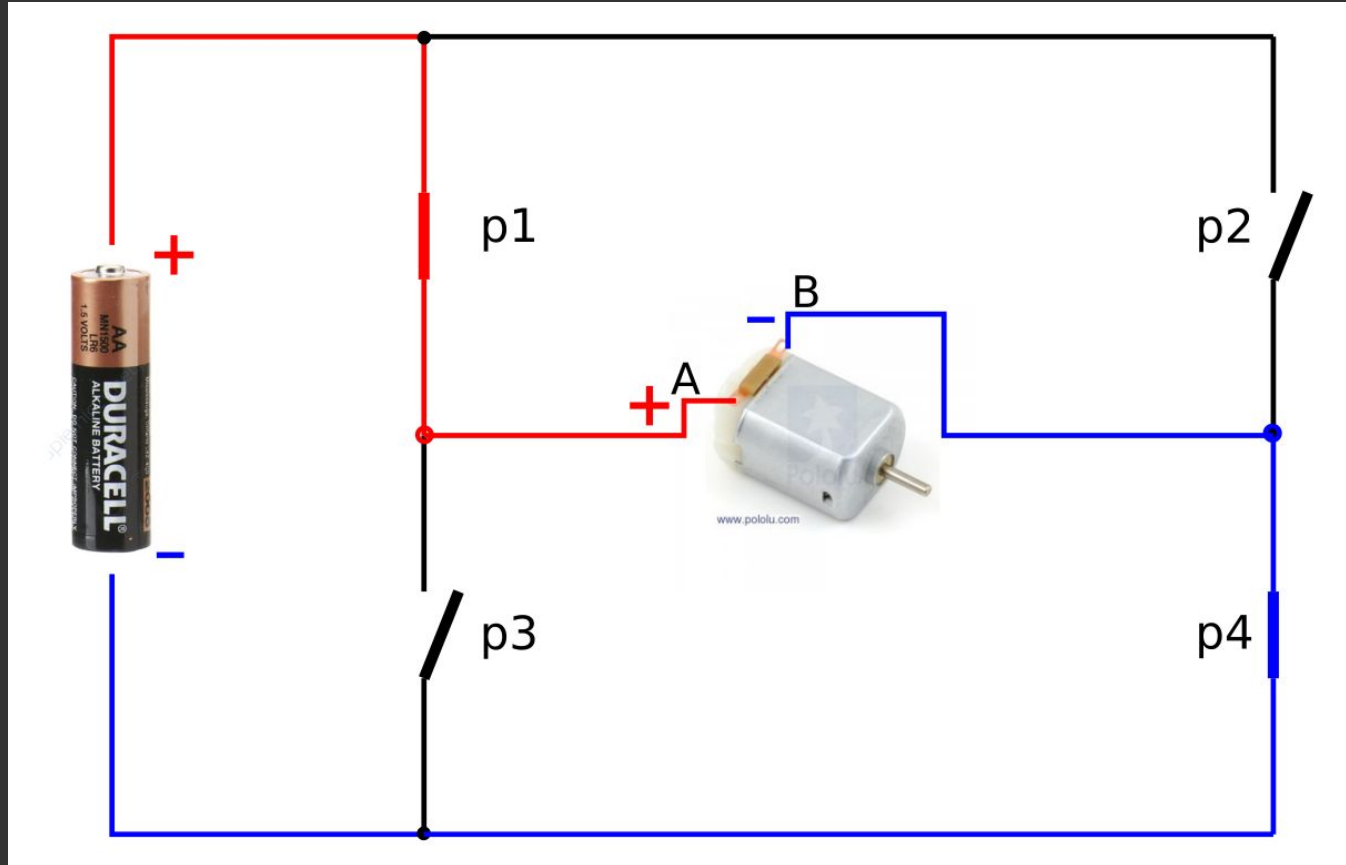


Silnik DC - podstawy sterowania

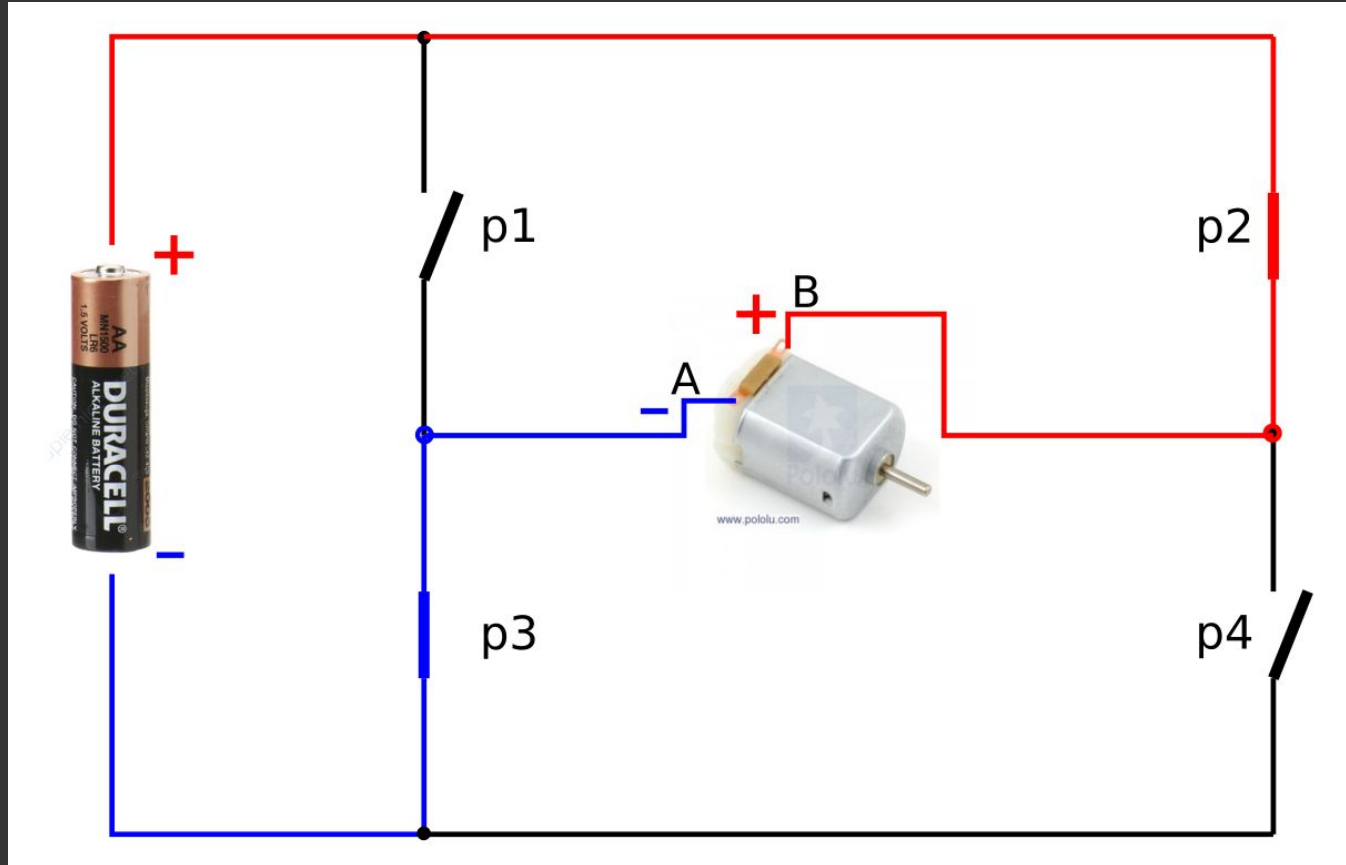
Mostek H: schemat ogólny



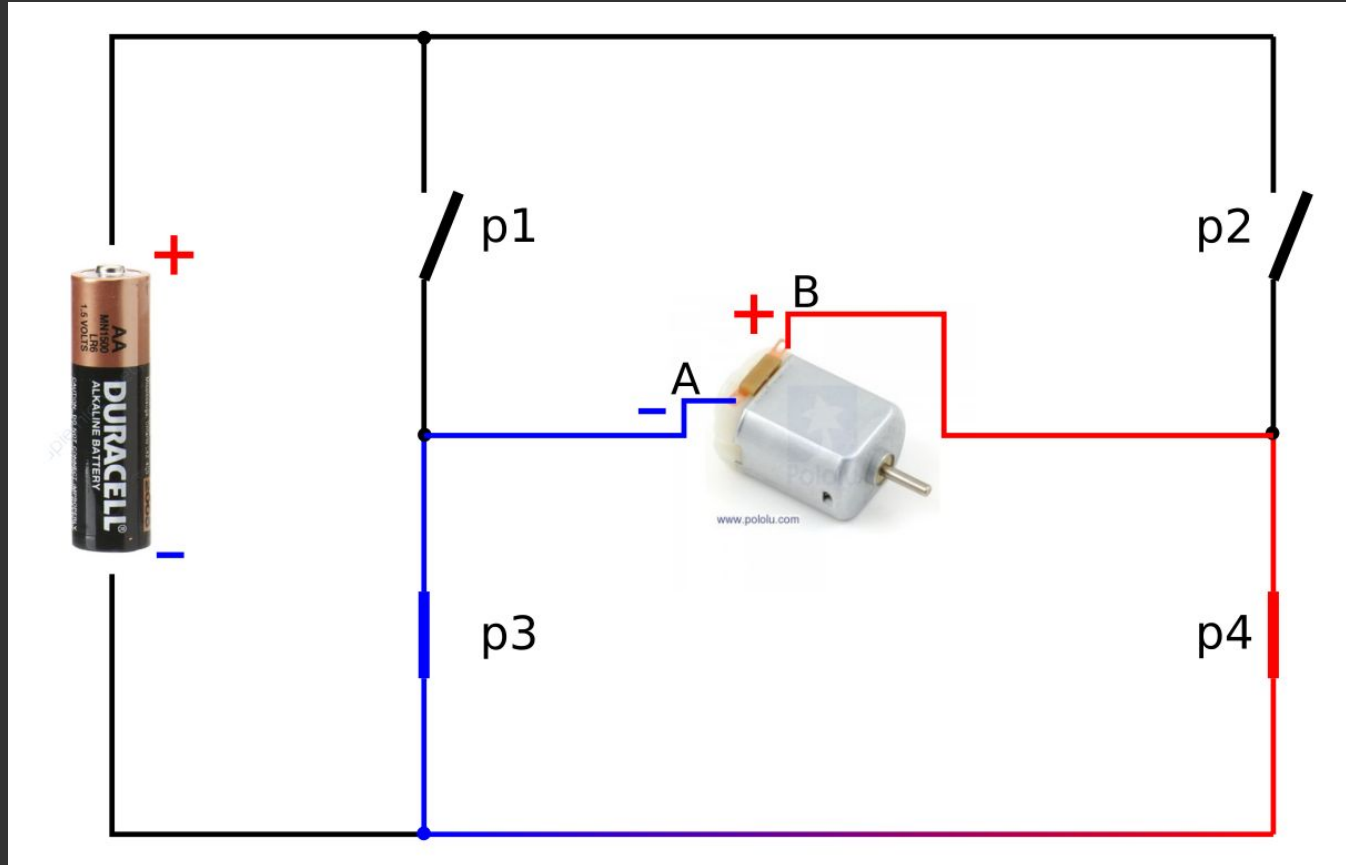
Mostek H: ruch “w lewo”



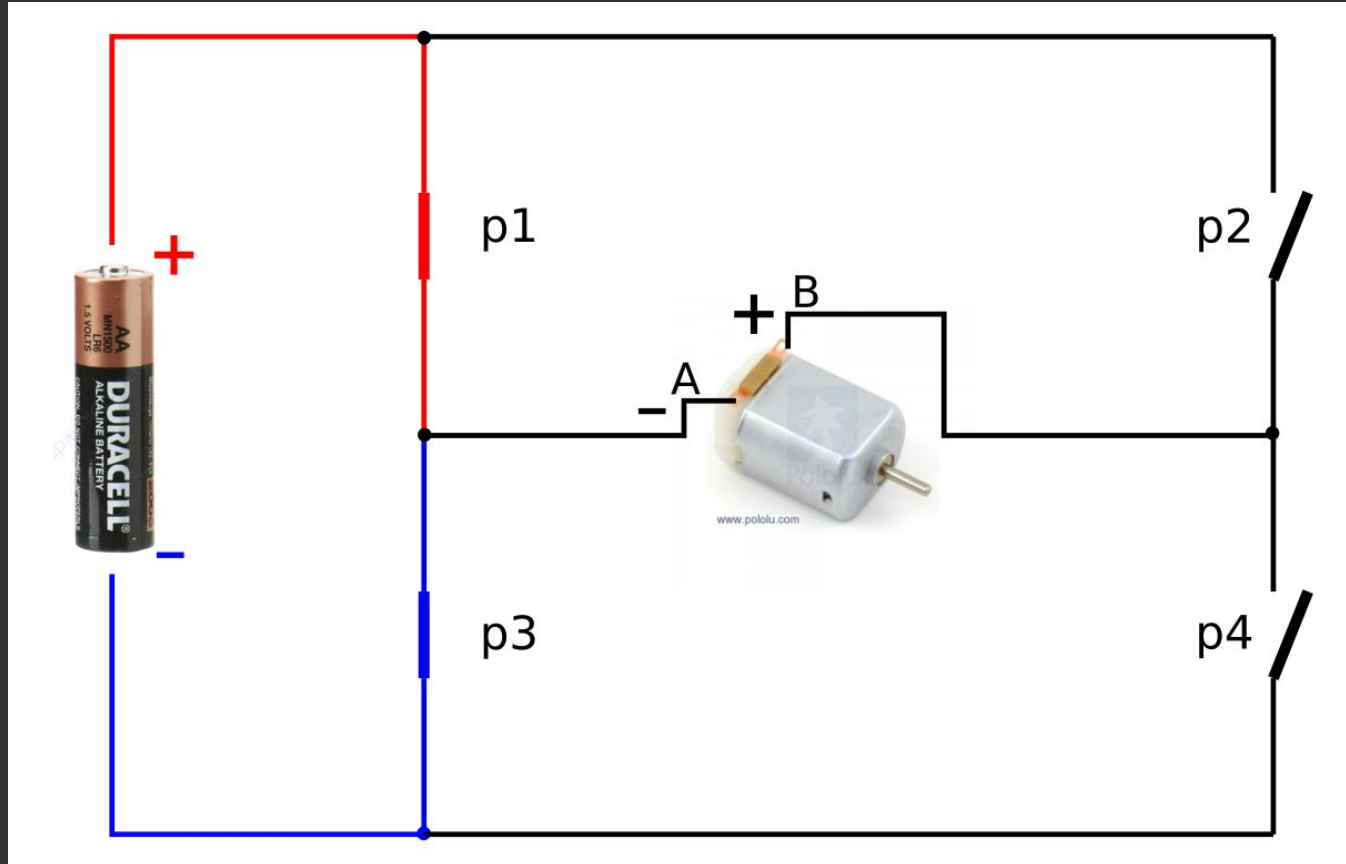
Mostek H: ruch “w prawo”



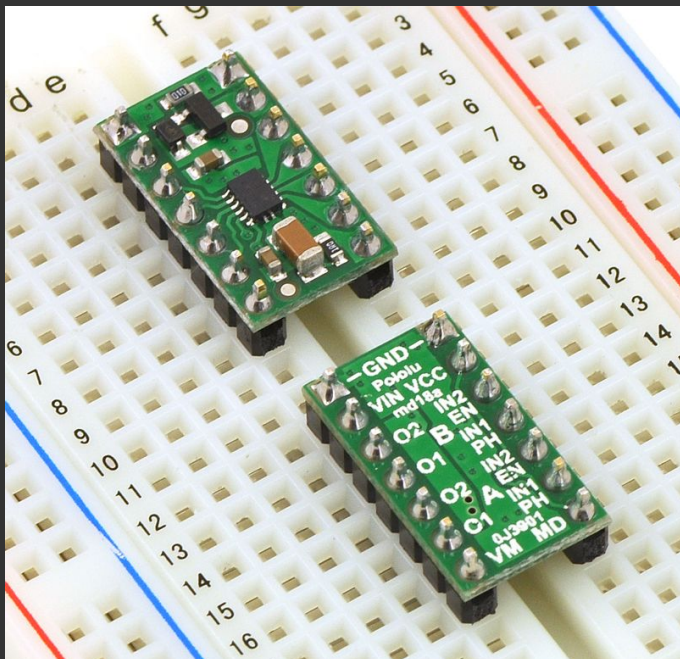
Mostek H: aktywne hamowanie



Mostek H: Stan niedozwolony



DRV8835: podwójny sterownik silnika DC



Moduł POLOLU 2135

Parametry DRV8835:

Liczba silników: 2

Architektura: 2x mostek H pełny

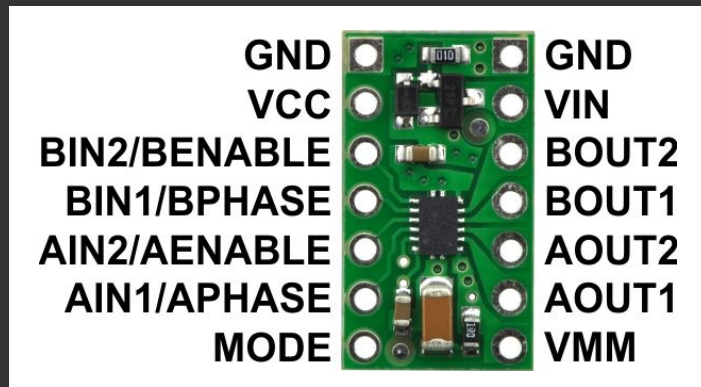
Zasilanie mostka: 0-11 V

Zasilanie logiki: 3,3 V oraz 5 V

Sterowanie: IN/IN oraz PHASE/PWM

Prąd wyjścia: 1,2 A ciągły; 1,5 A chwilowo (do 5 sek)

DRV8835: opis wyprowadzeń



GND - masa

VCC - zasilanie logiki

VIN - zasilanie silnika

AOUTx - wyjście silnika A

BOUTx - wyjście silnika B

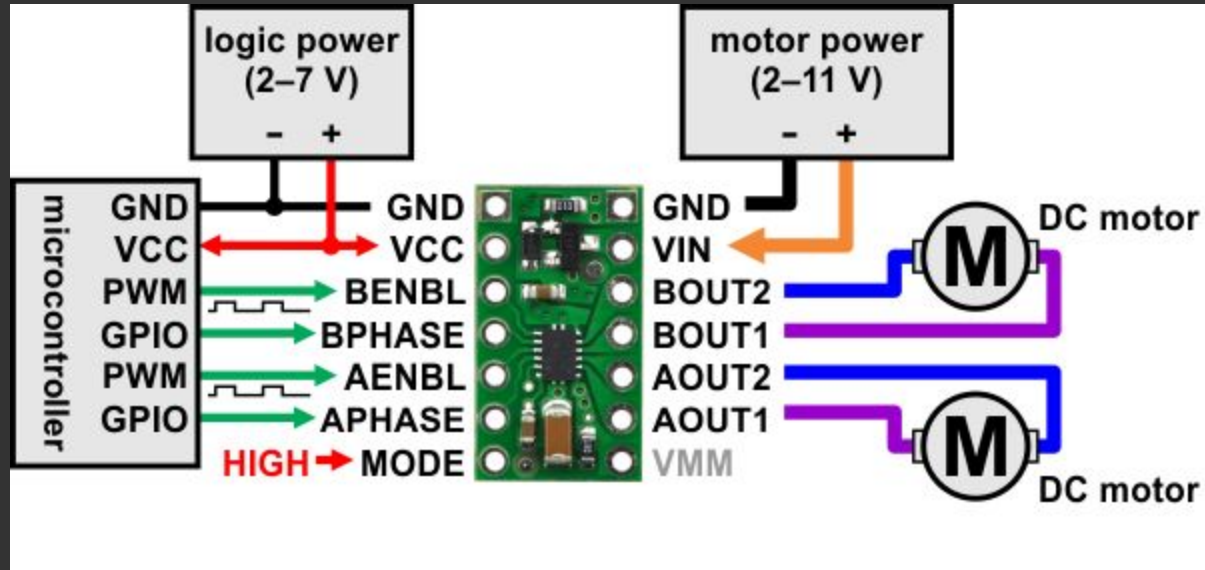
MODE - tryb pracy

LOW = IN/IN (wejścia AINx oraz BINx)

HIGH = PHASE/ENABLE (wejścia APHASE/AENABLE oraz BPHASE/PENABLE).

Domyślnie MODE = LOW

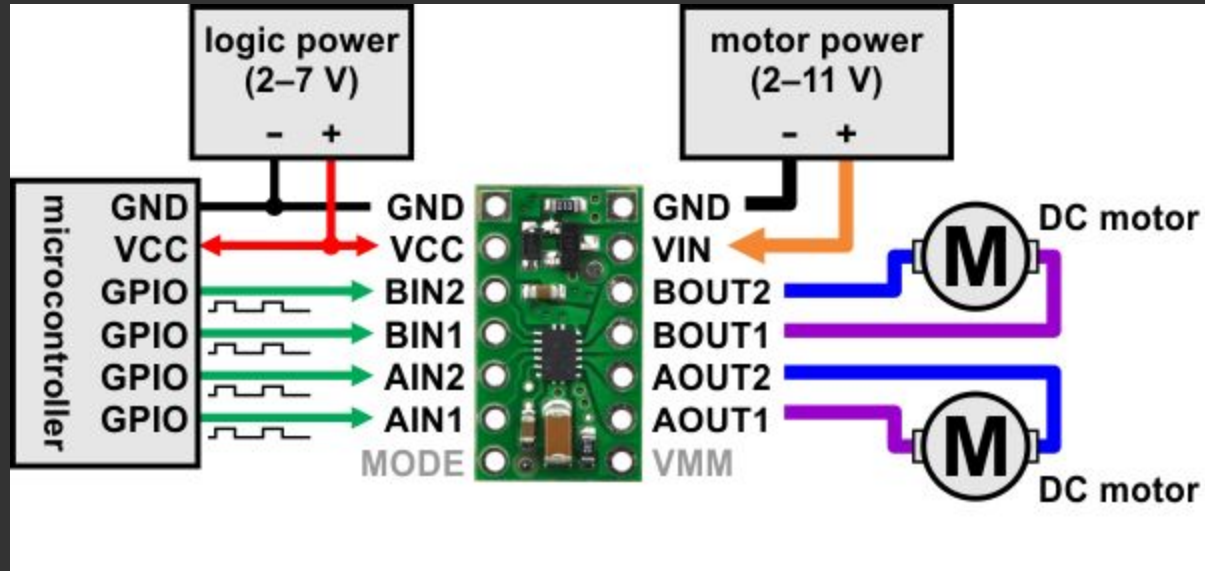
DRV8835: tryb **PHASE/ENABLE** (MODE=HIGH)



DRV8835: tryb **PHASE/ENABLE** (MODE=HIGH)

Simplified drive/brake operation with MODE=1 (PHASE/ENABLE)				
xPHASE	xENABLE	xOUT1	xOUT2	operating mode
1	PWM	L	PWM	reverse/brake at speed PWM %
0	PWM	PWM	L	forward/brake at speed PWM %
X	0	L	L	brake low (outputs shorted to ground)

DRV8835: tryb **IN/IN** (MODE=LOW lub wiszący)



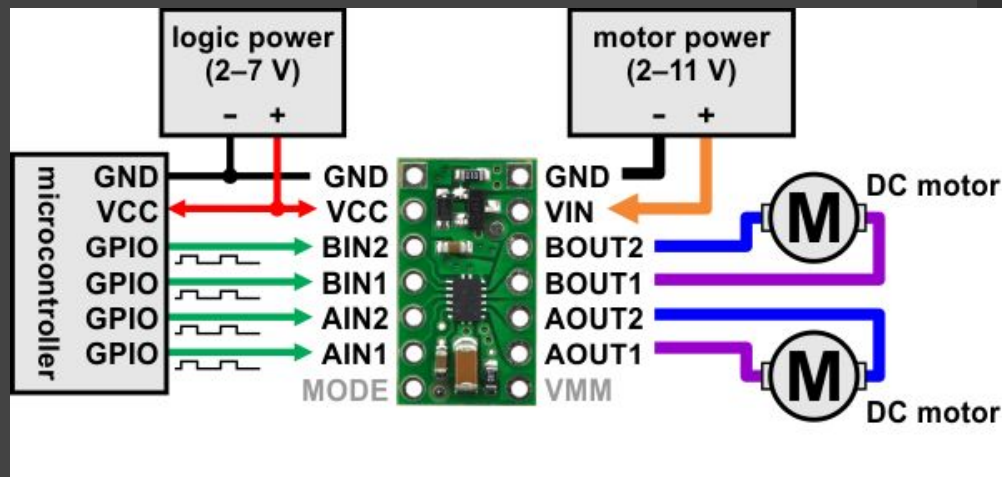
DRV8835: tryb **IN/IN** (MODE=LOW lub wiszący)

Drive/coast or drive/brake operation with MODE=0 (IN/IN)				
xIN1	xIN2	xOUT1	xOUT2	operating mode
0	0	OPEN	OPEN	coast (outputs off)
0	PWM	L	PWM	reverse/coast at speed PWM %
PWM	0	PWM	L	forward/coast at speed PWM %
PWM	1	L	PWM	reverse/brake at speed 100% – PWM %
1	PWM	PWM	L	forward/brake at speed 100% – PWM %
1	1	L	L	brake low (outputs shorted to ground)

DRV8835: płynna zmiana prędkości silników

```
void setup() {  
  pinMode(11, OUTPUT);  
  pinMode(13, OUTPUT);  
  
  digitalWrite(11, HIGH); // AIN1; tryb forward/break  
  digitalWrite(13, HIGH); // AIN2  
  
  for (int i = 0; i < 255; ++i) {  
    analogWrite(10, i); // BIN1; tryb reverse/coast  
    analogWrite(12, i); // BIN2  
    delay(50);  
  }  
  digitalWrite(11, LOW);  
  digitalWrite(13, LOW);  
  for (int i = 0; i < 255; ++i) {  
    analogWrite(10, i);  
    analogWrite(12, i);  
    delay(50);  
  }  
}
```

// 10=SERVO1, 11=SERVO2, 12=SERVO3, 13=SERVO4



Enkoder obrotów RS030



Czujnik RS030

Czujnik: efekt Halla

Źródło pola: okrągły magnes
neodymowy, 8 biegunów

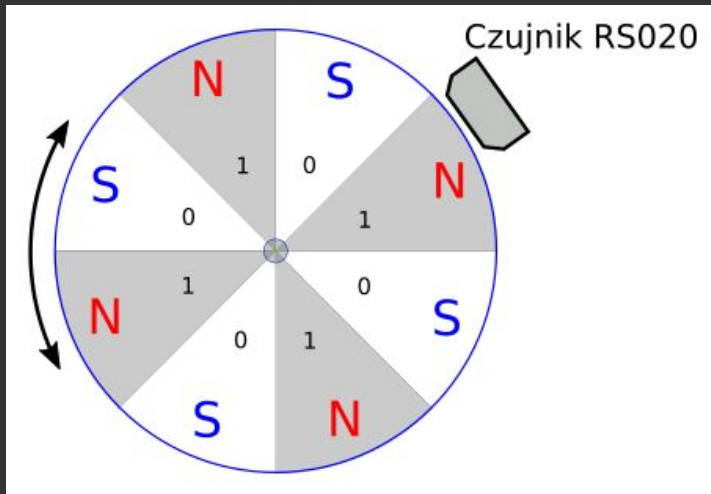
Wyjście: otwarty dren

(wymaga rezystora podciągającego do
VCC rzędu 10k)

Enkoder obrotów RS030

Czujnik RS030

Czujnik: efekt Halla
Źródło pola: okrągły magnes
neodymowy, 8 biegunów



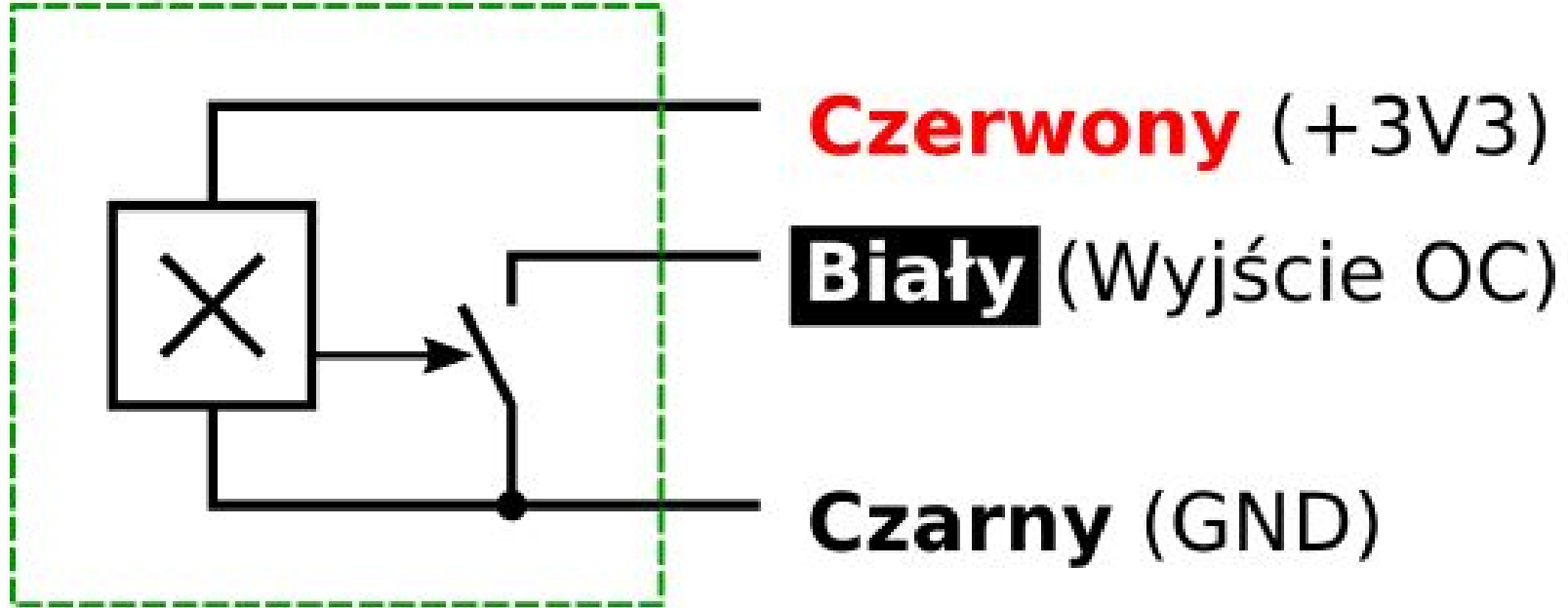
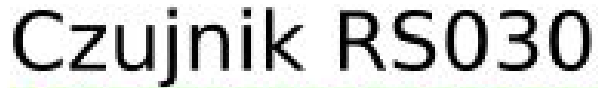
Enkoder obrotów RS030 - pomiar prędkości koła

```
volatile int i = 0, cnt = 0, velocity = 0;
volatile unsigned long last_measurement;
void counter() {
    i++;
    cnt++;
    if (last_measurement + 1000 <= millis()) {
        last_measurement = millis();
        velocity = cnt;
        cnt = 0;
    }
}

void setup() {
    Serial.begin(9600);
    pinMode(44, INPUT);           // 44 = USOUND_ECHO
    attachInterrupt(digitalPinToInterrupt(44), counter, CHANGE);
}

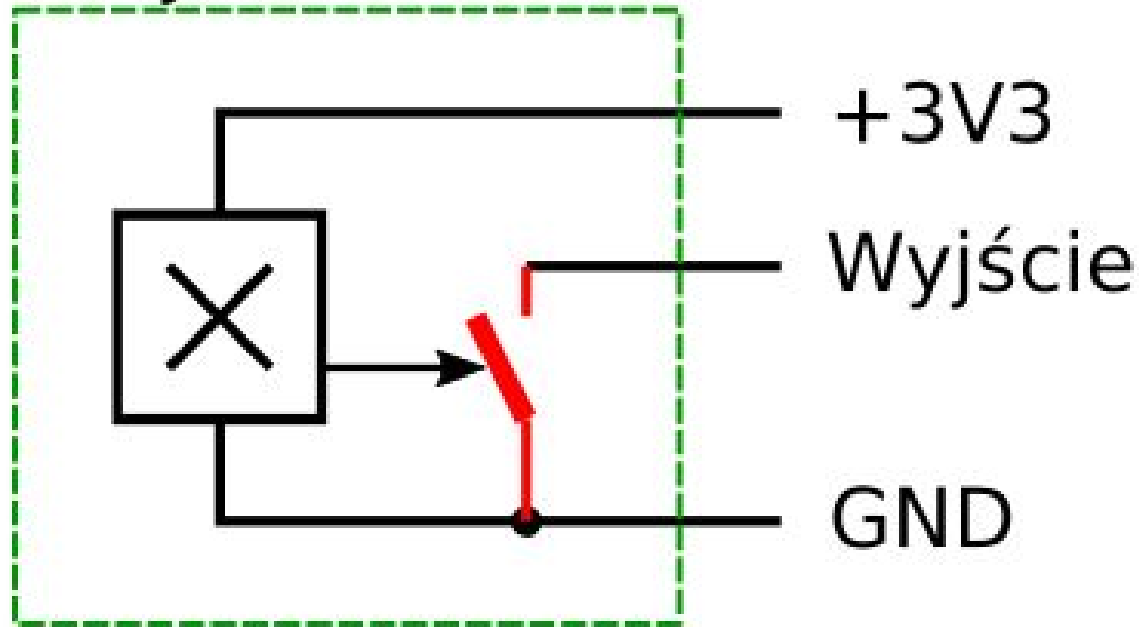
void loop() {
    char buf[32];
    sprintf(buf, "i=%d; vel=%.2f RPM\n", i, (float)velocity * 60.0f / 8.0f);
    Serial.print(buf);
    delay(250);
}
```

Enkoder obrotów RS030 - podłączenie



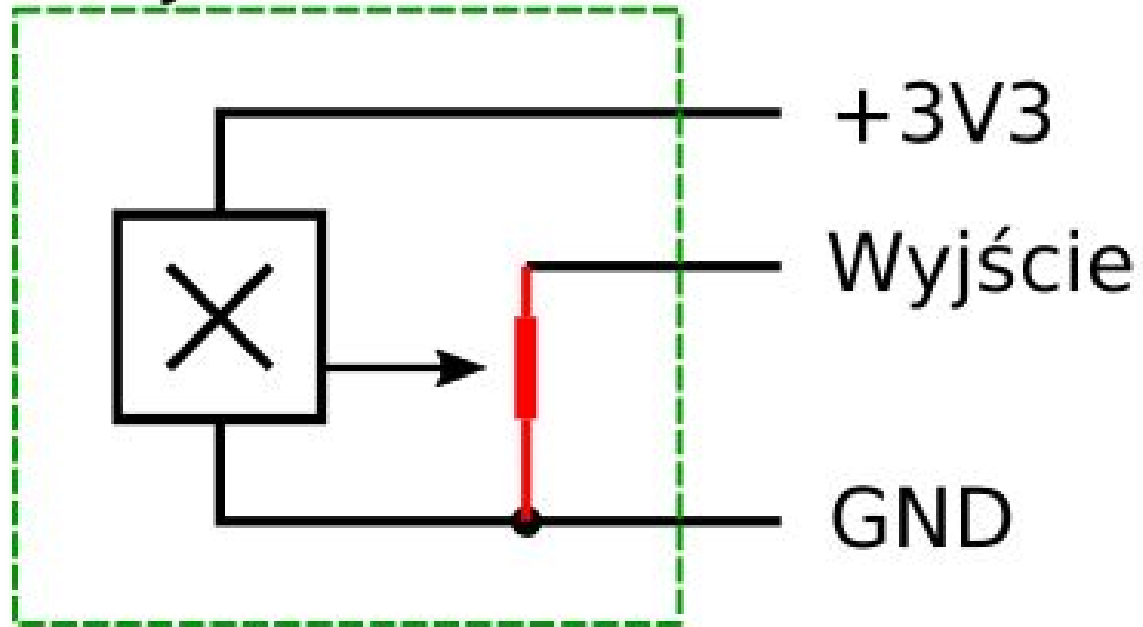
Enkoder obrotów RS030 - połączenie

Czujnik RS030



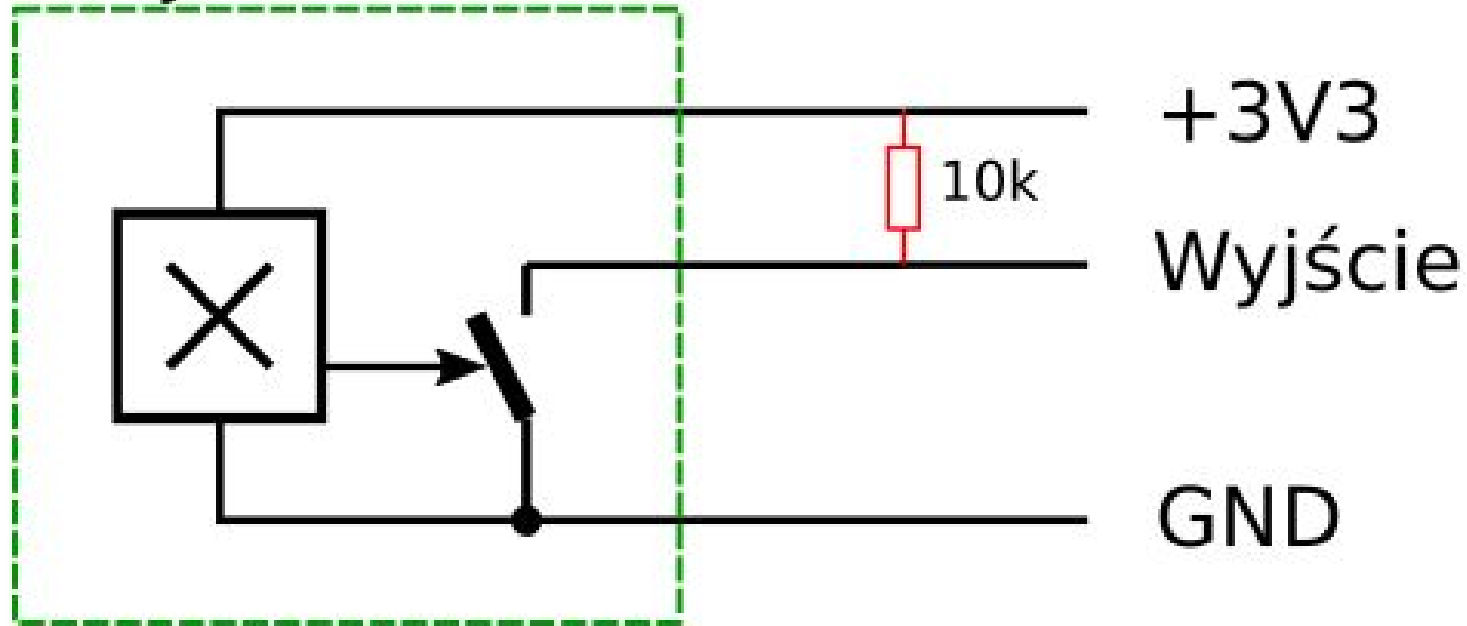
Enkoder obrotów RS030 - połączenie

Czujnik RS030



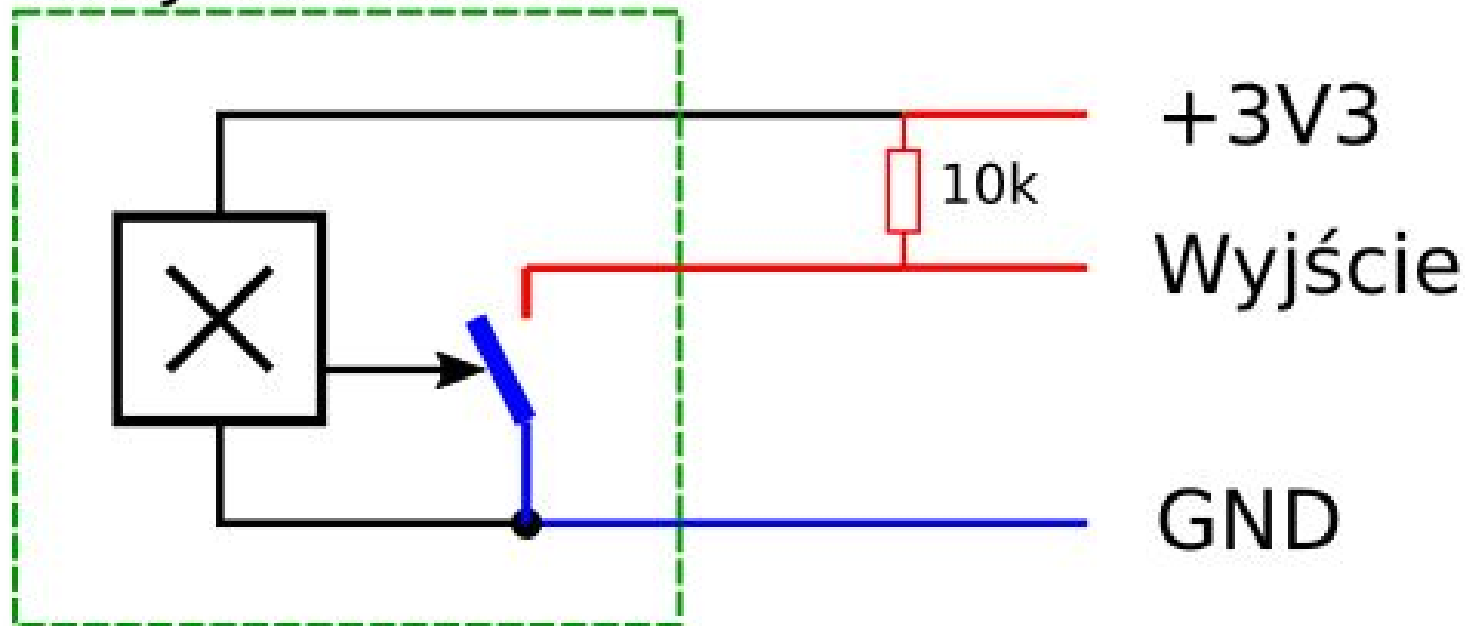
Enkoder obrotów RS030 - połączenie

Czujnik RS030



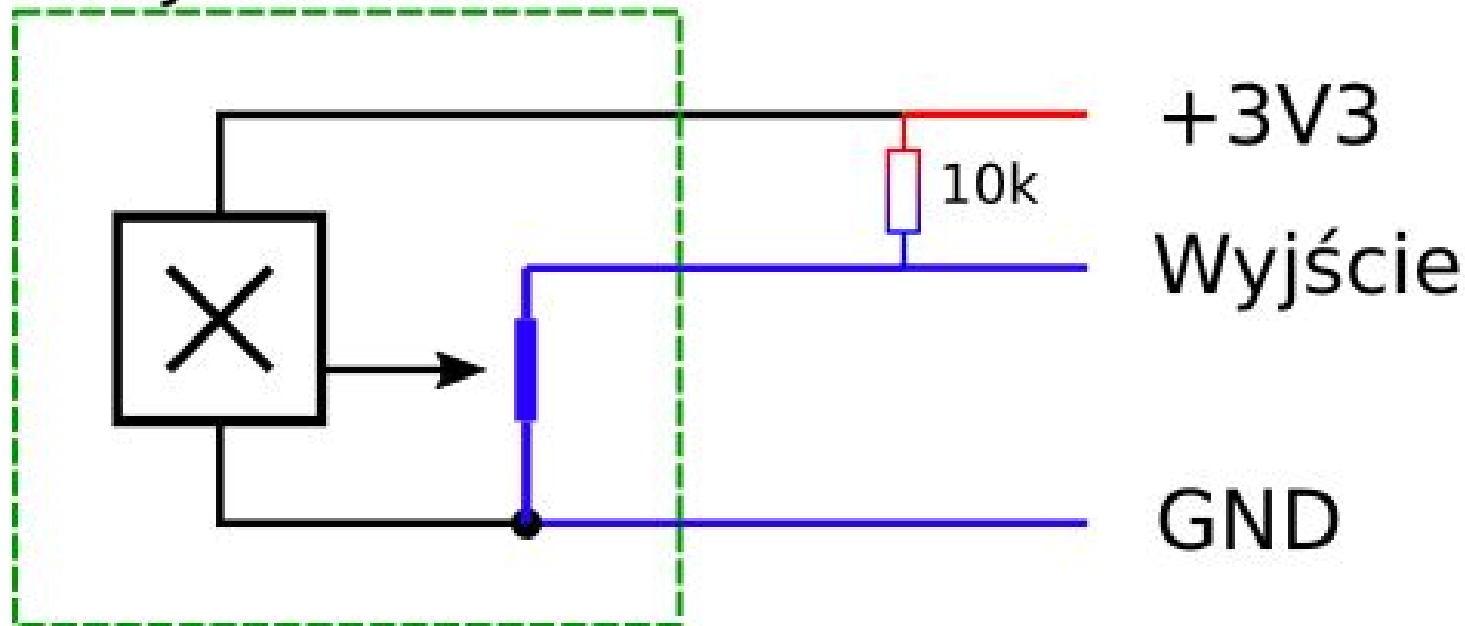
Enkoder obrotów RS030 - połączenie

Czujnik RS030



Enkoder obrotów RS030 - połączenie

Czujnik RS030



Dziękuję za uwagę

Serwonapęd



Serwonapędy/serwosilniki

Parametry:

Zasilanie: 4-9 V

Sterowanie: RC PWM

Moment: 1-100 kg/cm

Podziały:

Elektronika: cyfrowa i analogowa

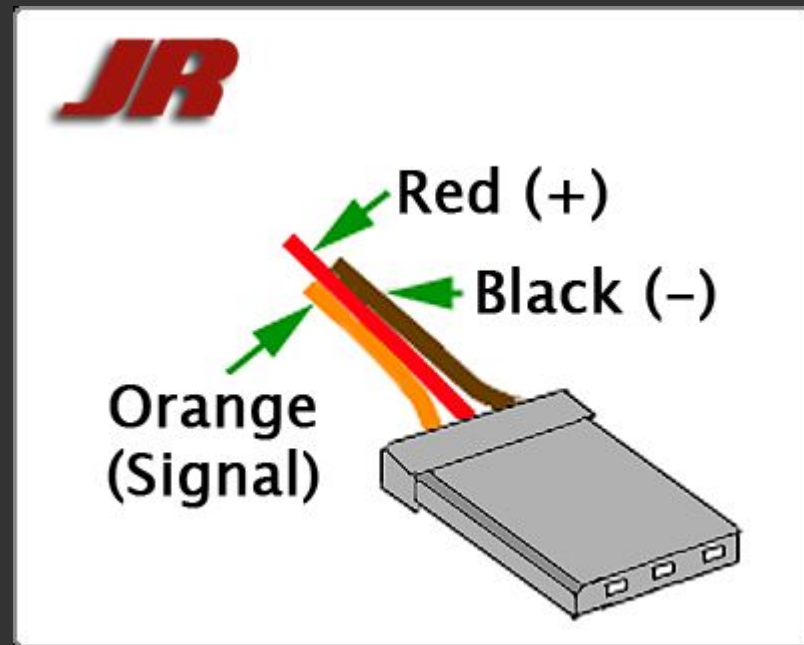
Mechanika: plastikowa i metalowa

Gabaryty (i parametry): micro, mini, standard, large, monster

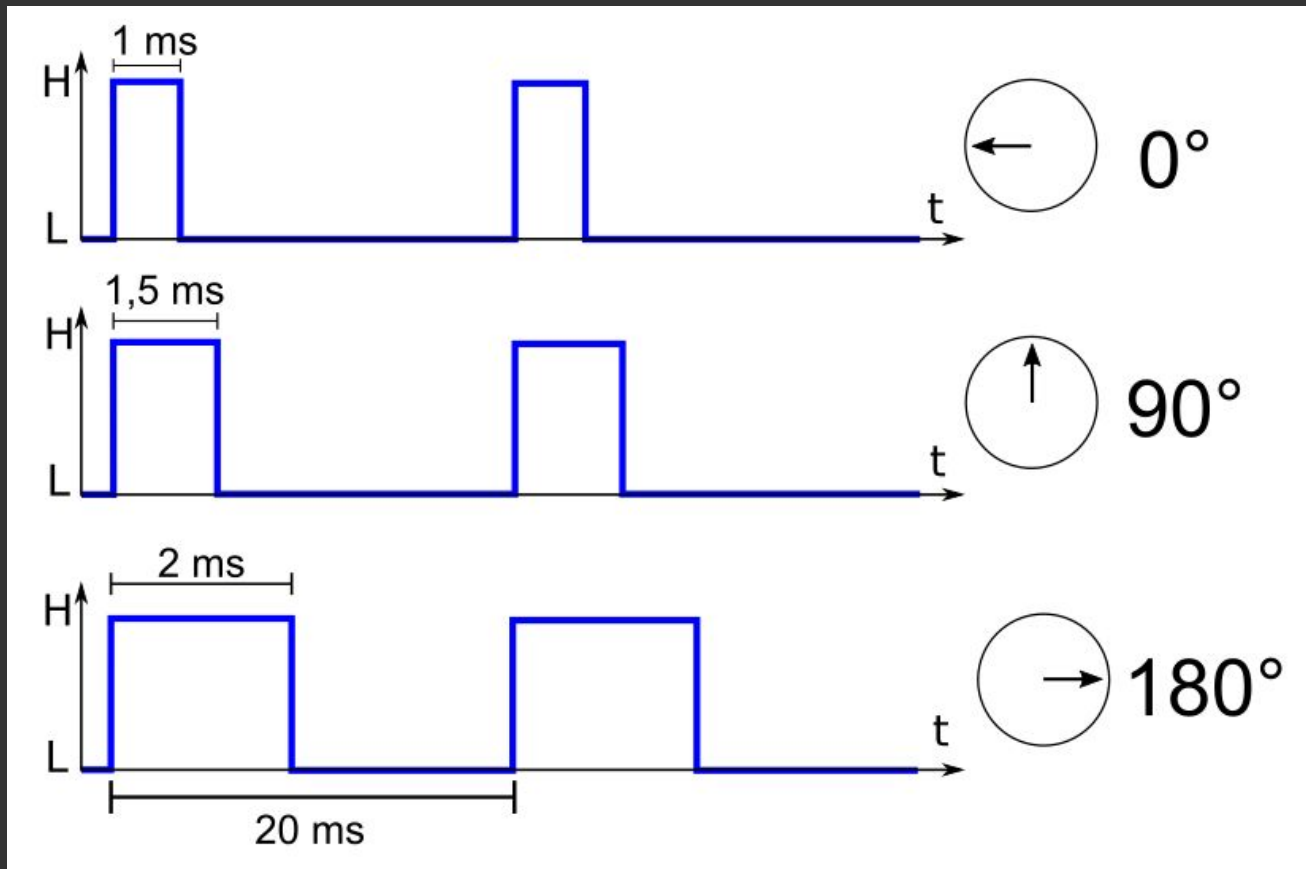
Serwonapęd: złącze zasilające i sygnałowe

Trzy kontakty:

1. Masa (-) - czarny (zimny)
2. Zasilanie (+) - czerwony (gorący)
3. Sygnał (S) - pomarańczowy/żółty



Serwonapęd: sterowanie



Dziękuję za uwagę