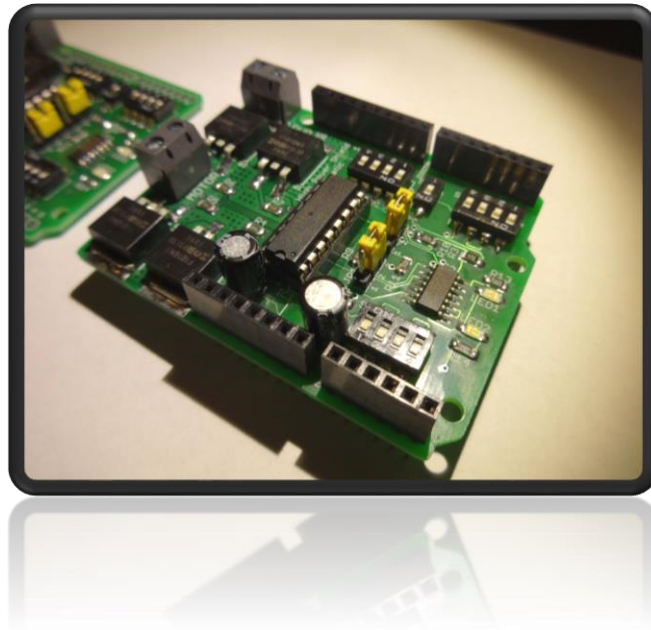


# *Arduino Power Shield*

**Moduł Arduino do sterowania silnikami dużej mocy i pomiaru prądu**

## **ERAmatic**



## 1. Opis ogólny

W odpowiedzi na potrzeby szybko rozwijającego się rynku prototypowania z użyciem platformy Arduino opracowaliśmy moduł Arduino Power Shield. Shield - to nakładka na płytkę bazową Arduino, rozszerzająca jego podstawowe możliwości. W sprzedaży lista takich nakładek jest obszerna - od modułów czujników, płytki przekaźnikowe, sterowanie silnikami krokowymi po moduły komunikacji bezprzewodowej. Siłą Arduino jest szybki start projektowanego układu docelowego z możliwie najmniejszym nakładem czasu na budowanie biblioteki lub projektowanie płytek PCB.

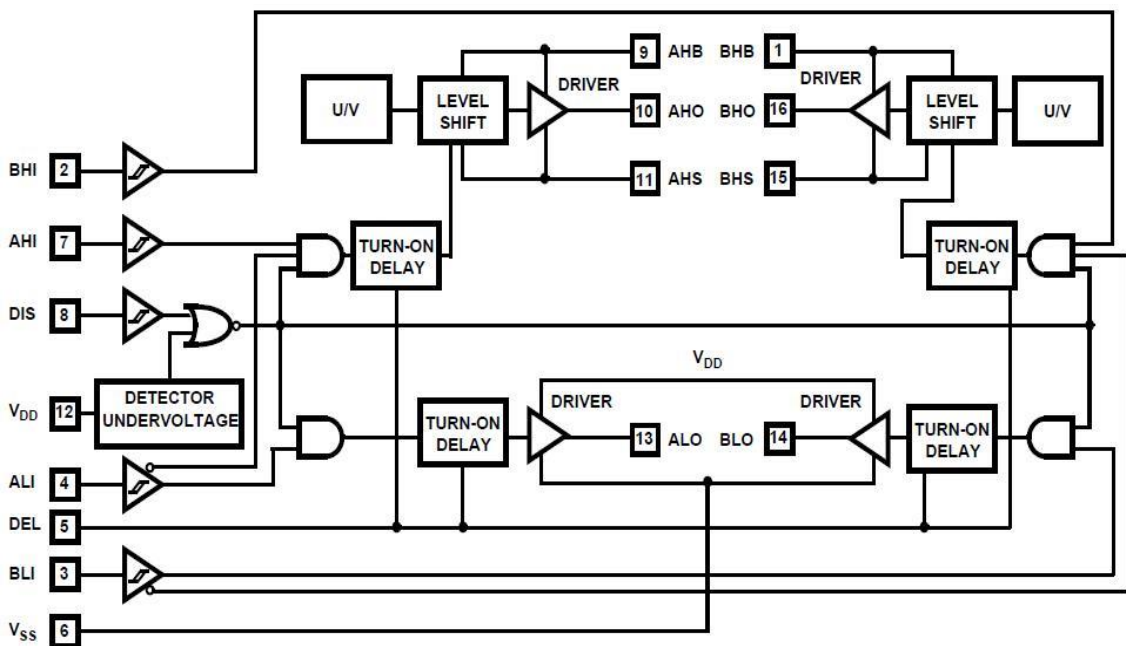
Teraz wprowadzamy własny unikalny Shield. Arduino Power Shield to nakładka do sterowania silnikami lub obciążeniami prądu stałego dużej mocy. Jak dużej mocy? W podstawowej wersji dla napięć do 48VDC oraz prądów rzędu do 80A. Praktycznie oznacza to możliwość sterowania mocami rzędu 1-2kW w pik. Jednak zaleca się nie przekraczania wartości 20A, ponieważ układ pomiaru prądu działa w tych granicach. Zaprojektowany układ posiada czujnik prądu silnika co umożliwi sterowanie np.: algorytmem PID momentem silnika lub ciągły pomiar prądu i odłączenie sterowania przez ograniczenie.

Cechy modułu Arduino Power Shield :

- możliwość sterowania prądem / momentem oraz kierunkiem obrotów silnika - pełny mostek typu H
- dedykowane złącze do podłączenia silnika prądu stałego
- dedykowane złącze do podłączenia zasilania dla silnika
- 4 grupy dip-switchy pozwalające na wybranie jednego z wejść Arduino do sterowania lub jednoczesne sterowanie kilku silników
- pomiar prądu silnika w zakresie +-20A (inne zakresy dostępne po uprzednim kontakcie)
- możliwość sterowania w zakresie do 48VDC oraz do 80A (inne zakresy dostępne po uprzednim kontakcie)
- diody LED wskazujące prawidłowe zasilania +5V oraz +12V
- możliwość wysyłania pomiaru prądu przez interfejs USB (port COM)

## 2. Zasada działania

Moduł do sterowania silników dużej mocy został zaprojektowany na bazie wieloletniego doświadczenia z układami mostkowymi, ich zabezpieczania oraz emisyjności. Zasada działania mostka polega na odpowiednim załączaniu poszczególnych elementów kluczujących. W przypadku mostka H użytego w module są to tranzystory MOSFET. Jako sterownik użyto układ scalony HIP4082, którego schemat budowy zaczerpnięty z noty katalogowej przedstawiono na rysunku poniżej:



Schemat ideowy działania układu HIP4082

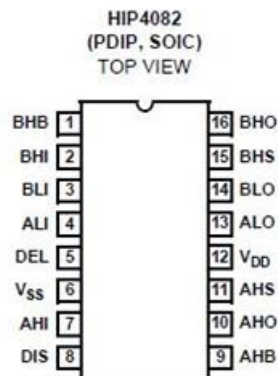
Układ ten jest przeznaczony do sterowania tranzystorami unipolarnymi MOS-FET poprzez sygnały PWM o częstotliwości do 200kHz.

Podstawowymi parametrami układu HIP4082IBZ są:

- zasilanie 12V DC,
- częstotliwość przełączania do 200kHz,
- zasilanie układu bootstrap maksymalnie 95V,
- przy obciążeniu silnikami o 1000pF pojemności, czas przełączania wynosi typowo 15ns,

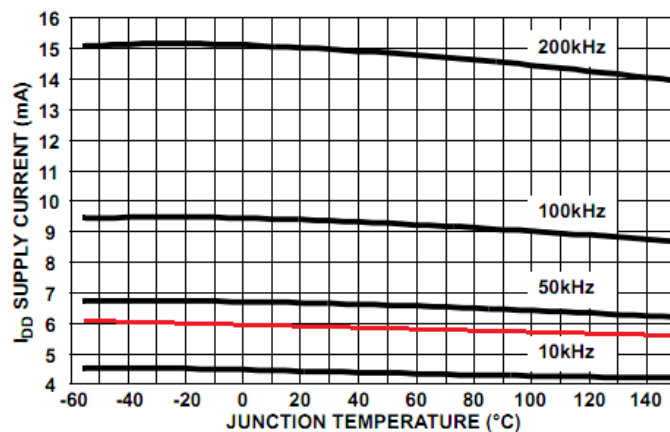
- zabezpieczenie nad prądowe,
- zabezpieczenie przed krótkotrwałymi impulsami wysokonapięciowymi.

Na rysunku poniżej przedstawiono wygląd układu oraz oznaczenie wyprowadzeń.



Sterownik HIP4082 - opis wyprowadzeń

Pobór prądu przez sterownik HIP4082IBZ przedstawiono poniżej. Na podstawie wykresu, działający z częstotliwością 33kHz układ wraz ze sterowaniem PWM pobiera ok. 6mA prądu.



Pobór prądu dla sterownika HIP4082IBZ dla częstotliwości przełączania 33kHz

Kolorem czerwonym zaznaczono wartość prądu pobieranego przez sterownik HIP4082IBZ w funkcji temperatury. Jak widać układ należy do grupy urządzeń energooszczędnych.

Opis wyprowadzeń układu scalonego HIP4082:

BHB – Zasilanie układu „bootstrap” w kanale B

BHI – Wejście sygnału sterującego kanał B, wejście high, które steruje wyjściem BHO

BLI – Wejście sygnału sterującego kanał B, wejście low, które steruje wyjściem BLO

ALI – Wejście sygnału sterującego kanał A, wejście low, które steruje wyjściem ALO

DEL – rezystor opóźniający załączenie sterownika

V<sub>SS</sub> – Masa układu, GND

AHI – Wejście sygnału sterującego kanał A, wejście high, które steruje wyjściem AHO

DIS – Wejście cyfrowe Disable

AHB – Zasilanie „bootstrap” w kanale A

AHO – Wyjście sterujące pracą tranzystora w kanale A high

AHS – Wejście sygnału sprawdzającego „bootstrap” w kanale A

V<sub>DD</sub> – Zasilanie 12V sterownika

ALO – Wyjście sterujące pracą tranzystora w kanale A low

BLO – Wyjście sterujące pracą tranzystora w kanale B low

BHS – Wejście sygnału sprawdzającego „bootstrap” w kanale B

BHO – Wyjście sterujące pracą tranzystora w kanale B high

Układ DC (dioda – kondensator) ulegają przeładowaniu koniecznemu doysterowania tranzystorów. Sterownik HIP4082 ustawia sygnały wyjściowe na tranzystory unipolarne MOS-FET IRF1010NS lub innych o podobnych parametrach. IRF1010ns to tranzystory o parametrach przedstawionych w tabelicy poniżej:

| Parametr                          | Wartość        | Jednostka |
|-----------------------------------|----------------|-----------|
| Prąd pobierany (max)              | 85             | [A]       |
| Wydzielana moc                    | 180            | [W]       |
| Temperatura pracy                 | od -55 do +175 | [°C]      |
| Spadek napięcia dren-źródło       | 55             | [V]       |
| Rezystancja przejścia dren-źródło | 11             | [mΩ]      |
| Czas załączania                   | 13             | [ns]      |
| Czas narastania                   | 76             | [ns]      |
| Czas wyłączenia                   | 39             | [ns]      |

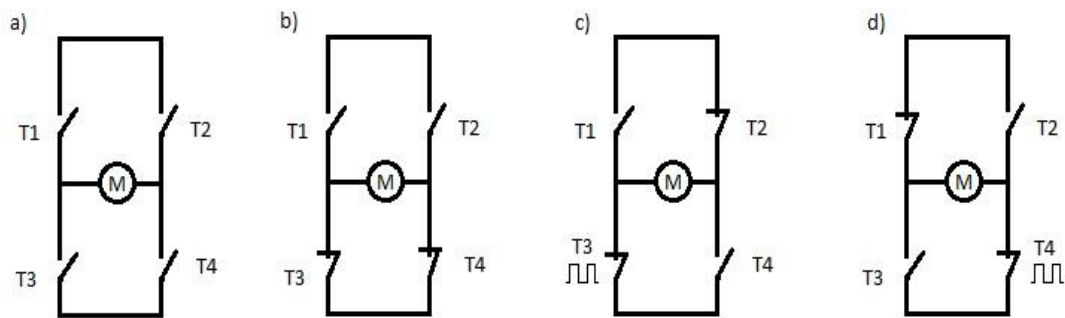
Wykorzystane tranzystory posiadają obudowę SMD (Surface Mount Devices – urządzenia przygotowane do montażu powierzchniowego). Pozwoliło to na zminimalizowanie rozmiarów płytki drukowanej. Sterowanie modułu odbywa się poprzez 3 sygnały wysyłane z mikrokontrolera (lub innej platformy szybkiego prototypowania) poprzez układ bramkowy. Sygnałami tymi są : sygnał PWM do sterowania prędkością, sygnał kierunku DIR oraz sygnał włączenia/wyłączenia DISABLE. Sterownikiem HIP4082IBZ również posiada obudowę SMD.

Silnik sterowany modułem może znajdować się w czterech stanach pracy. Pierwszym z nich jest stan, gdzie bazy wszystkich tranzystorów są w stanie niskim. Przy wysokim stanie sygnału DISABLE wszystkie tranzystory są w stanie otwartym. Oznacza to, że zaciski silnika nie są podpięte do żadnego potencjału. Na rysunku 1.a przedstawiono taką sytuację.

Drugim stanem jest stan hamulca. Jest to stan, w którym sygnał DISABLE jest w stanie niskim oraz sygnał PWM jest stały i ma niski stan. Powodują to, że wirnik silnika nie porusza się. Zaciski silnika są podłączone do zacisków masy. Na rysunku 1.b przedstawiono taką sytuację. Silnik wówczas trudniej obrócić a gdy takie sterowanie zostanie wykonane podczas ruchu – silnik bardzo szybko zatrzyma się.

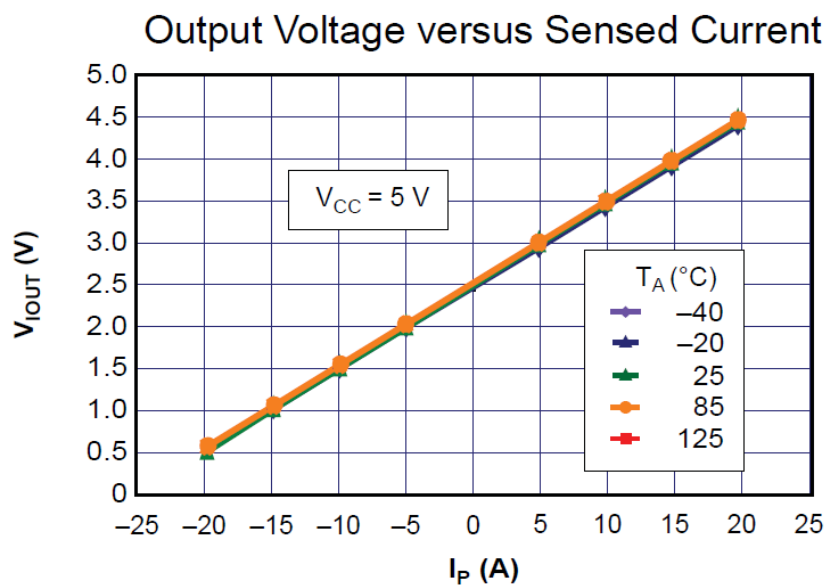
Trzecim stanem jaki może wystąpić w silniku to obrót w lewo. Sytuacja taka wystąpi gdy DISABLE = 0 oraz na sygnał wejściowy DIR zostanie ustawiony umownie na stan niski oraz na wejście PWM podano zmodulowany sygnał. Na rysunku 1.c przedstawiono taką sytuację.

Czwartym ostatnim stanem jest stan obrotu w prawo. Sytuacja taka wystąpi gdy DISABLE = 0 oraz na sygnał wejściowy DIR zostanie ustawiony umownie na stan wysoki oraz na wejście PWM podano zmodulowany sygnał. Na rysunku 1.d przedstawiono taką sytuację.



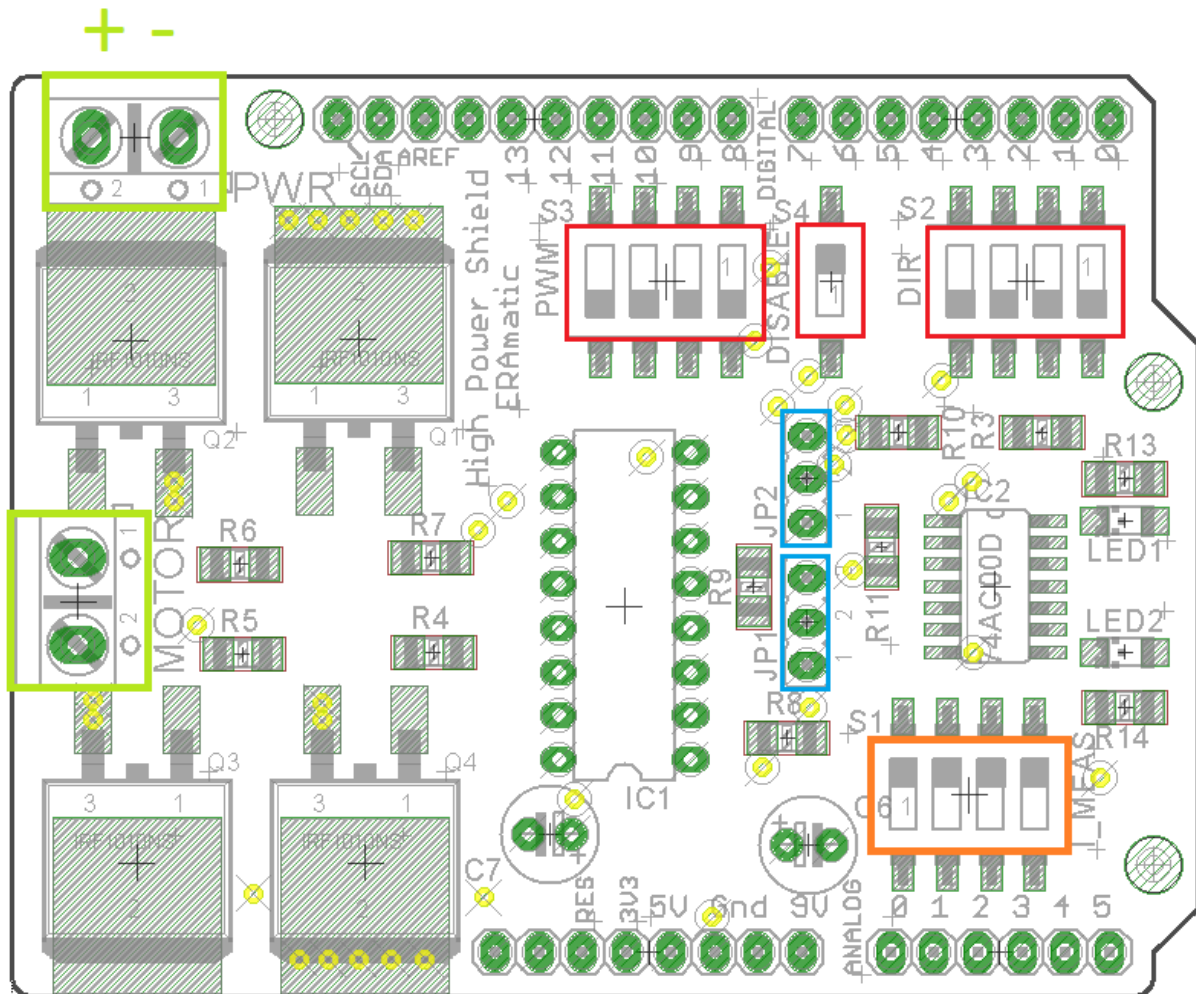
Rys. 1 Schematyczna zasada działania mostka H. a) stan bez załączonych tranzystorów, b) stan hamulca, c) obrót w lewo, d) obrót w prawo

Pomiar prądu realizowany jest dedykowanym układem ACS712. Układ ten składa się z dokładnego, bez offsetowego, liniowego obwodu Hall-a ze ścieżką miedzi znajdującą się blisko powierzchni pomiarowej. Prąd płynący przez tę ścieżkę miedzi generuje pole magnetyczne, które konwertuje układ Hall-a w IC jako proporcjonalne napięcie. Dokładność urządzenia jest zmaksymalizowana poprzez bliskość sygnału magnetycznego do przetwornika Hall. Pomiar prądu poprzez napięcie jest stabilny i dokładne w szerokim zakresie. Poniżej przedstawiono napięcie wyjściowe (napięcie pomiarowe) w zależności od prądu mierzonego dla napięcia zasilania 5V w różnych temperaturach pracy:



### 3. Możliwe konfiguracje

Rozmieszczenie elementów na płycie przedstawiono na rysunku poniżej:



**Kolor czerwony:**

- selektory (dip-switch) do wyboru sygnałów sterowania:

- PWM – sygnał wypełnienia – prędkości (do wyboru PIN: 8,9,10,11)
- DISABLE – sygnał sterowania włączeniem/wyłączeniem całego mostka (PIN:7)
- DIR – sygnał kierunku (do wyboru PIN: 0,1,2,3)

**Kolor pomarańczowy:**

- selektory (dip-switch) do wyboru sygnału pomiarowego:

- I\_MEAS – sygnał pomiaru prądu (do wyboru PIN ANALOG: 0,1,2,3)

**Kolor zielony:**

- złącze zasilające (w zakresie od 12V DC do 48V DC) polaryzacja podpisana na rysunku

- złącze do podłączenia silnika DC, polaryzacja obojętna



## Kolor niebieski:

- zworki podciągnięcia sygnału DISABLE i DIR do masy lub zasilania

- JP1 (DIRECTION)
  - Zworka założona do strony ANALOG – podłączenie sygnału kierunku DIR przez rezystor do masy (GND)
  - Zworka założona do strony DIGITAL – podłączenie sygnału kierunku DIR przez rezystor do 5V
  - Zworka założona tylko na środkowy PIN lub brak zworki – kierunek DIR jest sterowany z pinów Arduino
- JP2 (DISABLE)
  - Zworka założona do strony ANALOG – podłączenie sygnału DISABLE przez rezystor do masy [GND](mostek aktywny)
  - Zworka założona do strony DIGITAL – podłączenie sygnału DISABLE przez rezystor do 5V (mostek nieaktywny)
  - Zworka założona tylko na środkowy PIN lub brak zworki – sygnał DISABLE jest sterowany z pinów Arduino, domyślnie mostek nieaktywny

## 4. Przykładowy program Arduino Uno

W celu sprawdzenia modułów przygotowano program na platformę Arduino UNO. Podłączenie:

- DIGITAL 2 – sygnał kierunku DIR
- sygnał włączenia/wyłączenia DISABLE nie użyty, zworka w pozycji włączenia
- DIGITAL 9 – sygnał PWM
- ANALOG 0 – sygnał wejścia analogowego dla czujnika do pomiaru prądu

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);

  pinMode(9, OUTPUT);
  pinMode(2, OUTPUT);
  digitalWrite(2, HIGH);
  digitalWrite(9, LOW);
}
int speed_of_motor=0;
byte dir=0;
byte dir2=0;

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
```

```

// print out the value you read:

Serial.println(sensorValue-512);
delay(1);          // delay in between reads for stability

if(speed_of_motor<255 && dir==0)
{
  speed_of_motor++;
}
else
{
  if(speed_of_motor==255)
  {
    dir=1;

  }
  if(dir==1)
  {
    speed_of_motor--;
  }
  if(speed_of_motor<=0)
  {speed_of_motor=0;dir=0;
if(dir2==0)
{
  dir2=1; digitalWrite(2, LOW);

}
else
{
  dir2=0; digitalWrite(2,HIGH);
}
}
}

  analogWrite(9, speed_of_motor);

  delay(90);

}

```

Działanie programu:

Po resecie / uruchomieniu kierunek ustawiany jest w stan wysoki. Rozpoczyna się modulacja PWM w zakresie 0-255 na wyjściu DIGITAL 9. Z interwałem 90ms zwiększana jest prędkość. Po dojściu do wartości granicznej rozpoczyna się powolne dekrementowanie prędkości. Po zatrzymaniu następuje ustawienie stanu kierunku na niski. Ponownie rozpoczyna się powolna inkrementacja – dojście do prędkości maksymalnej i dekrementacja. Następnie zmieniany jest ponownie sygnał kierunku i cały proces rozpoczyna się od nowa. W tym czasie pomiar prądu silnika wysyłany jest na przy pomocy wbudowanego portu USB. Opcję tę można sprawdzić przy pomocy Serial Monitora w oprogramowaniu Arduino.