

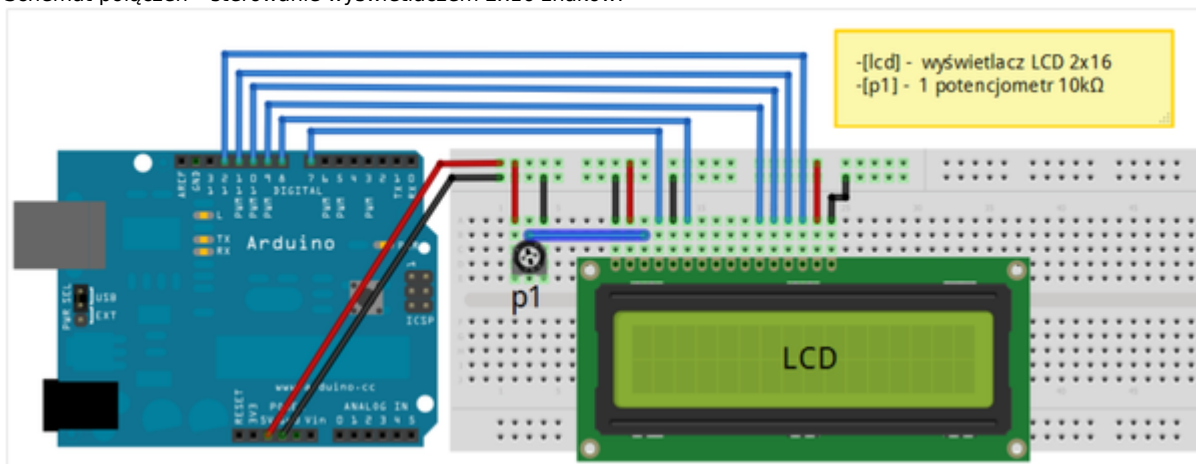


Nazwa implementacji: Pomiar czasu i wyświetlacz LCD - podłączenie i sterowanie LCD zgodnym z HD44780

Autor: Krzysztof Bytow

Opis implementacji: Stworzenie prostego stopera i zegara używając modułu-interfejsu. Wykorzystanie podstawowych funkcji do sterowania i prezentacji czasu na wyświetlaczu LCD jak i ekranie monitora. Zasada działania i używania bibliotek.

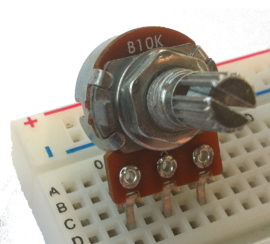
Schemat połączeń – sterowanie wyświetlaczem 2x16 znaków:



Zalecenia:

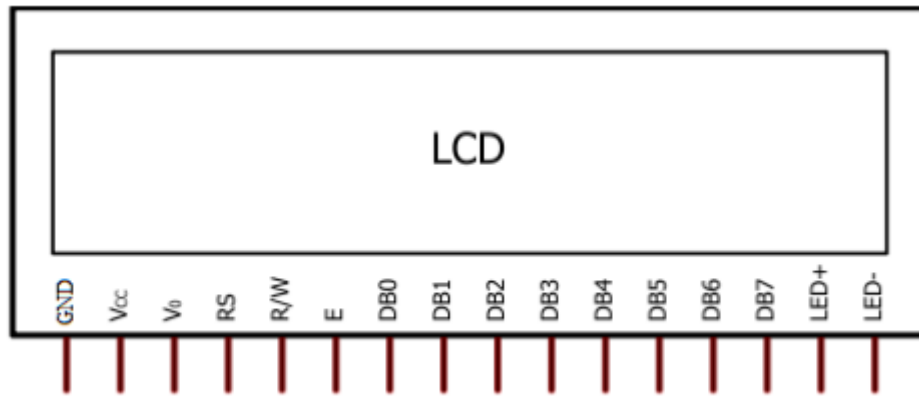
- w celu podłączenia zasilania dla +5 V używać przewodów o kolorze ciepłym (np.: czerwony)
- w celu podłączenia masy GND używać przewodów w kolorze zimnym (np.: niebieski, czarny)
- w celu regulacji ostrości wyświetlacza LCD należy zmienić wartość potencjometru do uzyskania czytelnych znaków na ekranie LCD
- dodatkowe komendy do sterowania wyświetlaczem dostępne są pod adresem: <http://arduino.cc/en/Reference/LiquidCrystal?from=Tutorial.LCDLibrary>

Uczeń/Uczennica po zestawieniu połączeń zgłasza nauczycielowi gotowość do sprawdzenia układu i wszystkich połączeń.



potencjometr 10kΩ →

Wyświetlacz alfanumeryczny LCD ze sterownikiem zgodnym z HD44780.



Numer	Nazwa	Opis
1	GND	Masa
2	Vcc	Napięcie zasilania +5V
3	Vo	Regulacja kontrastu wyświetlacza
4	RS	Wybór rodzaju informacji: 0 - komenda 1 - dane
5	R/W	Kierunek transmisji: 0 - wysłanie komendy lub danych 1 - odczyt stanu lub danych
6	E	Sygnał zezwalający: 0 - wejścia nieaktywne 1 - wykonanie działania
7	DB0	Linia danych D0
8	DB1	Linia danych D1
9	DB2	Linia danych D2
10	DB3	Linia danych D3
11	DB4	Linia danych D4
12	DB5	Linia danych D5
13	DB6	Linia danych D6
14	DB7	Linia danych D7
15	LED+	Anoda diody podświetlającej
16	LED-	Katoda diody podświetlającej

Sterowanie wyświetlaczem

```
#include <LiquidCrystal.h>          // podłączamy bibliotekę
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); //definiujemy zmienną lcd, przez którą będziemy się
//komunikować z modułem; podajemy, do których
```



```
                                //pinów podłączyliśmy się w Arduino
void setup()
{
  lcd.begin(16, 2);
}
void loop()                      // główna pętla
{
  lcd.clear();                   // czyścimy znaki na wyświetlaczu
  lcd.print(" !!! SWOI !!!");    // wyświetlamy napis
  lcd.setCursor(4, 1);          // ustawiamy kursor kolumna (wartości od 0 do 16)
                                // wiersz (wartości od 0 do 1)
  lcd.print(" Witaj ");         // wyświetlamy napis
  delay(2000);                  // czekaj 2 sekundy
}
```

Znaki specjalne – na przykładzie znaku stopnia wykorzystywanego przy temperaturze:
(generator znaków specjalnych umieszczono w serwisie e-Swoi.pl).

```
#include <LiquidCrystal.h>      // podłączamy bibliotekę
LiquidCrystal lcd(7, 8, 9, 10, 11, 12); //definiujemy zmienną lcd, przez którą będziemy się
                                //komunikować z modułem; podajemy, do których
byte znak[8] =
{
  B11100,
  B10100,
  B11100,
  B00000,
  B00000,
  B00000,
  B00000,
  B00000
};
void setup()                    // początkowa konfiguracja – część przygotowująca
{                               // działania
  lcd.createChar(1, znak);
  lcd.begin(16, 2);
}
void loop()                     // główna pętla
{
  lcd.print(" !!! SWOI !!!");    //wyświetlamy napis SWOI
  lcd.setCursor(0, 1);          //ustawiamy kursor w pozycji (kolumna, wiersz 0 lub 1)
  lcd.print(" Temp=38");        // wyświetlamy napis
  lcd.write(1);                 // wyświetlenie wcześniej zdefiniowanego znaku
  delay(3000);                  // odczekanie 3 sekund
  lcd.clear();                  // wyczyszczenie wyświetlacza
  lcd.print(" Test ");          // wyświetlenie napisu Test
  delay(1500);                  // odczekanie 1,5 sekundy
  lcd.clear();                  // wyczyszczenie wyświetlacza
}
```





Wyczyść

Generator kodu dla nowych znaków LCD Arduino

```
#include <LiquidCrystal.h>

LiquidCrystal lcd(7, 8, 9, 10, 11, 12);

byte newChar[8] = {
    B11100,
    B10100,
    B11100,
    B00000,
    B00000,
    B00000,
    B00000,
    B00000
};

void setup()
{
    lcd.createChar(0, newChar);
    lcd.begin(16, 2);
    lcd.write(0);
}

void loop()
{
}
```