

LABORATORIUM - ELEKTRONIKA

Komunikowanie wyświetlaczy LCD z układami mikroprocesorowymi na platformie Arduino

1. Cel ćwiczenia

Celem ćwiczenia jest zaznajomienie studenta z budową systemów opartych na platformie Arduino. W ramach ćwiczenia student uczy się komunikować z różnego typu wyświetlaczami oraz zapoznaje się ze środowiskiem programistycznym Arduino.

2. Trochę teorii

Z punktu widzenia osoby początkującej, Arduino jest gotowym zestawem uruchomieniowym opartym na mikrokontrolerze z rodziny AVR. Zestaw ten stworzony jest według konkretnych założeń, dzięki czemu:

- nie wymaga zewnętrznego programatora;
- współpracuje z dedykowanym kompilatorem;
- można dokupić do niego „nieskończoną” liczbę płytek rozszerzających (takich jak np. sterowniki silników, wyświetlacze, moduły wykonawcze, czujniki i wiele innych).

Projekt zaczął być rozwijany w 2005 roku, we Włoszech. Od tej pory zgromadził rzeszę zwolenników czy wręcz fanatycznych użytkowników. Od samego początku Arduino było przygotowywane z myślą o osobach, które nie miały wcześniej zbyt wiele wspólnego z programowaniem mikrokontrolerów. Przystępne środowisko, przyjazna składnia oraz niska cena sprawiły, że Arduino stało się niezwykle popularne.

Arduino jest platformą typu **Open Hardware**. Oznacza to, że powszechnie dostępne są wszelkie materiały potrzebne do stworzenia własnego zestawu rozwojowego działającego w tym standardzie. Z tego powodu znaleźć można wiele różnych płytek zgodnych z Arduino.

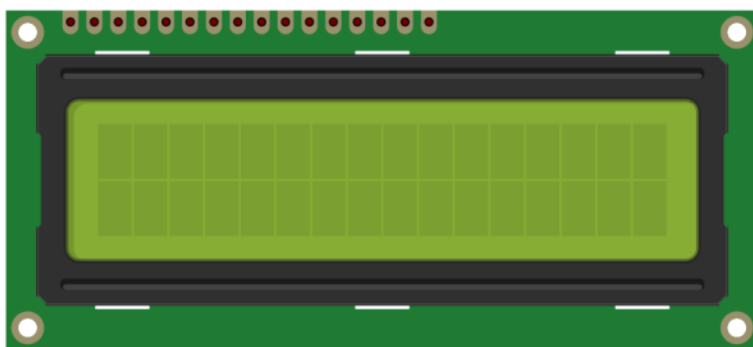
Arduino Mega jest z jedną z najbogatszych wersji. Płytką zawiera mikrokontroler ATmega2560, wyposażony w 54 wejść/wyjść cyfrowych, z czego 15 można wykorzystać jako wyjścia PWM (np. do sterowania silnikami) oraz 16 wejść analogowych. Układ taktowany jest sygnałem zegarowym o częstotliwości 16 MHz, posiada 256 kB pamięci programu Flash oraz 8 kB pamięci operacyjnej SRAM.

Specyfikacja:

- napięcie zasilania: 7-12 V,
- mikrokontroler: ATmega 2560,
- maksymalna częstotliwość zegara: 16 MHz,
- pamięć SRAM: 8 kB,
- pamięć Flash: 256 kB (8 kB zarezerwowane dla bootloadera),
- pamięć EEPROM: 4 kB,
- piny I/O: 54,
- kanały PWM: 15,
- liczba wejść analogowych: 16 (kanały przetwornika A/C o rozdzielczości 10 bitów),
- interfejsy szeregowo: 4xUART, SPI, I2C,
- zewnętrzne przerwania,
- dioda LED - pin 13,
- gniazdo USB A - do programowania,
- złącze DC 5,5 x 2,1 mm - do zasilania.

PRZEBIEG ĆWICZENIA

1. Wybrać z dostarczonych przez prowadzącego następujące elementy:
 - układ Arduino Mega,
 - płytki prototypowa,
 - wyświetlacz 2X16,
 - potencjometr 10K Ω ,
 - przewody połączeniowe.
2. Z pomocą płytki stykowej połączyć platformą Arduino z wyświetlaczem 2x16, kompatybilnym ze sterownikiem HD44780, zgodnie z wytycznymi poniżej:



Opis poszczególnych wejść wyświetlacza (od lewej do prawej):

1. **GND** – masa – 0V
2. **V_{cc}** – zasilanie dodatnie - 5V
3. **V₀** – regulacja kontrastu – środkowa nóżka potencjometru
4. **RS** – wybór rejestrów (komenda, dane) – **Arduino PIN 2**
5. **RW** – wybór opcji odczyt/zapis - 0V (ponieważ interesuje nas tylko wysyłka danych **do** wyświetlacza)
6. **E** – zezwolenie na zapis do rejestrów - **Arduino PIN 3**
7. **D0** – dane - NC
8. **D1** – dane - NC
9. **D2** – dane - NC
10. **D3** – dane - NC
11. **D4** – dane - **Arduino PIN 4**
12. **D5** – dane - **Arduino PIN 5**
13. **D6** – dane - **Arduino PIN 6**
14. **D7** – dane - **Arduino PIN 7**
15. **V_{pod}** – zasilanie dodatnie podświetlenia – jedna zewnętrzna nóżka potencjometru i **5V**
16. **GND_{pod}** – masa podświetlenia – druga zewnętrzna nóżka potencjometru i **0V**

Piny od 1 do 3 służą do zasilania układu, od 4 do 14 do sterowania, zaś pod 15 i 16 znajduje się wewnętrzna dioda świecąca, która podświetla ekran. Wyświetlacze kompatybilne ze sterownikiem HD44780 mogą komunikować się z otoczeniem w trybie **4-bitowym oraz 8-bitowym**.

W pierwszym z nich konieczne jest 7 połączeń Arduino <-> wyświetlacz. Natomiast w przypadku trybu 8-bitowego należy zrealizować ich aż 11. Korzystając z mniejszej liczby wyprowadzeń zachowujemy całość możliwości wyświetlacza.

3. Po zmontowaniu układu należy...
- 3.1. Podłączyć Arduino do gniazda USB.
- 3.2. Uruchomić środowisko Arduino.
- 3.3. W ustawieniach wybieramy odpowiedni model Arduino (czyli po prostu ten, na którym będziemy pracować). W tym celu wybieramy "Narzędzia", następnie "Płytkę" i wybieramy odpowiedni model.
- 3.4. W ustawieniach wybieramy port komunikacyjny, na którym będziemy pracować.
W tym celu wybieramy "Narzędzia", następnie "Port" i wybieramy odpowiedni port.
- 3.5. Uzupełniamy szkic o poniżej zaprezentowany kod, przy czym zamiast znaków zapytania należy podać odpowiednie numery portów Arduino (w kolejności).

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(?,?,?,?,?);
void setup()
{
  lcd.begin(16,2);
}
void loop()
{
  lcd.setCursor(0,0);
  lcd.print("Lubie placki");
}
```

- 3.6. Po przepisaniu kodu można załadować kod do mikrokontrolera. Na wyświetlaczu powinien pojawić się właściwy napis, należy też pamiętać, by za pomocą potencjometru ustawić odpowiedni kontrast.

Uwagi:

Jak widać na załączonym kodzie, we fragmencie „setup” umieszczamy ten kod, który ma się wykonać jednokrotnie, natomiast w sekcji „loop”, ten który będzie się wykonywał w sposób zapętłony.

Zadania do wykonania:

- 4.1. Korzystając z załączonych opisów instrukcji biblioteki LiquidCrystal.h należy poprawić napis „Lubie placki” na „Lubię placki”.
- 4.2. Korzystając z załączonych opisów instrukcji biblioteki LiquidCrystal.h oraz listy komend z tej biblioteki należy zrealizować program wykonujący „coś ciekawego” na minimum dwóch zmieniających się ekranach. Może w tym pomóc funkcja „delay(ms)”, gdzie „ms” to czas pauzy w milisekundach.
Od tego, co zaproponuje grupa (w sensie na ile będzie to zaskakujące i ciekawe „technicznie”) zależy ocena ćwiczenia.

Opracowanie ćwiczenia: Paweł Chwietczuk