

<b>LABORATORIUM - ELEKTRONIKI</b>
<b>Układy mikroprocesorowe cz.2</b>

### PRZEBIEG ĆWICZENIA

1. Wybrać z dostarczonych przez prowadzącego następujące elementy
  - Układ Arduino Mega
  - Płytki prototypowa
  - Wyświetlacz 2X16
  - Potencjometr 10K $\Omega$
  - Przewody połączeniowe
  - czujnik ruchu PIR
  - czujnik płomienia
2. Przy pomocy znalezionych elementów zbudować na płytce stykowej układ z Rysunku 1, układ pomocniczy znajduje się na końcu instrukcji w załączniku 1

### IDEA MONTAŻU UKŁADÓW PROTOTYPOWYCH NA PŁYTKCE STYKOWEJ:

Połączenia wewnątrz płytki stykowej są zrealizowane tak jak na niej zaznaczono, czyli zwarte po 5 w kolumnach. Dodatkowo obie zewnętrzne (górną i dolną) linie są zwarte w trzech grupach, po 20, 15 i 20 wtyków.

Idea montażu polega na tym, że elementy, które mają się ze sobą łączyć muszą mieć odpowiednie nóżki/piny wetknięte w tę samą 5-wtykową kolumnę lub którąś z dłuższych linii.

To, czego nie da się połączyć bezpośrednio (dotyczy to przede wszystkim tranzystorów<sup>1</sup>) można zewrzeć używając dostępnych przewodów (jako zworek).

- 2.1. Po zmontowaniu układu należy zgłosić gotowość nauczycielowi prowadzącemu, jeżeli prowadzący zaakceptuje wykonanie połączeń, można wtedy przystąpić do dalszej części ćwiczenia
- 2.2. Podłączyć Arduino do gniazda USB
- 2.3. Uruchomić program Arduino
- 2.4. W ustawieniach wybieramy odpowiedni model Arduino na jakim będziemy pracować W tym celu wybieramy pasek "Narzędzia" następnie "Płytki" i wybieramy odpowiedni model
- 2.5. W ustawieniach wybieramy odpowiedni model Arduino na jakim będziemy pracować.  
W tym celu wybieramy pasek "Narzędzia" następnie "Płytki" i wybieramy odpowiedni model
- 2.6. W ustawieniach wybieramy odpowiedni model procesora Arduino na jakim będziemy pracować.  
W tym celu wybieramy pasek "Narzędzia" następnie "Procesor" i wybieramy odpowiedni model
- 2.7. W ustawieniach wybieramy odpowiedni port komunikacyjny Arduino na jakim będziemy pracować.

W tym celu wybieramy pasek "Narzędzia" następnie "Port " i wymieramy odpowiedni

- 2.8. Uzupełniamy kompilator o poniżej zaprezentowany kod, zamiast znaków zapytania należy podać numery portów Arduino

```
//code

#include <LiquidCrystal.h> //Dołączenie wbudowanej biblioteki do obsługi
wyświetlacza
LiquidCrystal lcd (??, ??, ??, ??, ??, ??); //Informacja o podłączeniu wyświetlacza
//(RS, E,D4,D5,D6,D7)

void setup() {
  lcd.begin(16, 2); //Deklaracja typu wyświetlacz
  lcd.setCursor(0, 0); //Ustawienie kursora na pozycji (0,0)
  lcd.print("Linijka 1"); //Wyświetlenie tekstu
  lcd.setCursor(0, 1); // Ustawienie kursora na pozycji (0,1)
  lcd.print("Linijka 2"); //Wyświetlenie tekstu
}

void loop() {
}
```

- 2.9. Po przepisaniu poprawnie kodu wybieramy opcję "Skic" następnie "Weryfikuj/Kompiluj", jeżeli program zostanie skompilowany poprawnie możemy przystąpić do wgrania naszego programu na Arduino, wówczas wybieramy opcję "Skic" następnie "Wgraj", po wgraniu na wyświetlaczu powinniśmy uzyskać napisy :

```
Linijka 1
Linijka 2
```

- 2.10. **Należy również za pomocą potencjometru ustawić odpowiedni kontrast !!!**

## Cz.2. Czujniki z wyjściem cyfrowym (czujnik ruchu, czujnik płomienia, czujniki wody)

- 3.1. Odłączamy złącze usb od układu Arduino.
- 3.2. Podłączamy czujnik ruchu do płytki prototypowej zgodnie z parametrami znajdującymi się w wstępie teoretycznym, Wyjście z czujnika podłączamy do wolnego wejścia cyfrowego następnie w programie zastępujemy znaki zapytania numerem portu na Arduino, przed uruchomieniem prosimy nauczyciela prowadzącego o sprawdzenie poprawności połączeń.
- 3.3. Podłączmy Arduino do portu usb i dokonujemy konfiguracji podłączenia w dostępnym oprogramowaniu, następnie uzupełniamy program o kod zawarty poniżej, kompilujemy i sprawdzamy działanie czujnika.

```

//code

#include <LiquidCrystal.h> //Dołączenie biblioteki
LiquidCrystal lcd(2, 3, 4, 5, 6, 7); //Informacja o podłączeniu nowego wyświetlacza
#define PIR ?? // podaj pin do którego podpięty został sygnał z czujnika

void setup() {
  lcd.clear(); // czyszczenie lcd
  lcd.begin(16, 2); //Deklaracja typu
  pinMode(PIR, INPUT_PULLUP); //PIR jako wejście
}

void loop() {

  if (digitalRead(PIR) == HIGH) { //Jeśli wykryto ruch
    lcd.clear();
    lcd.setCursor(0, 0); //Ustawienie kursora
    lcd.print("wykryto ruch"); //Wyświetlenie tekstu
    delay(500); // odczekaj
  }
  else
  {
    lcd.clear();
    lcd.setCursor(0, 0); //Ustawienie kursora
    lcd.print("brak ruchu"); //Wyświetlenie tekstu
    delay(500);
  }
}

```

3.4. Po poprawnie wykonanej poprzedniej części zadania, przystępujemy do podłączenia kolejnego czujnika, w tym celu odłączamy Arduino od portu usb, dokonujemy zmiany czujnika oraz sposobu wyświetlania komunikatów stosownie do zastosowanego czujnika.

### Cz.3. Czujnik ultradźwiękowy HC-SR04

- 4.1. Odłączamy złącze usb od układu Arduino.
- 4.2. Podłączamy czujnik ultradźwiękowy do płytki prototypowej zgodnie z parametrami znajdującymi się w wstępie teoretycznym, Wyjście oraz wejście (**ECHO, TRIG**) z czujnika podłączamy do wolnych wejść cyfrowych następnie w programie zastępujemy znaki zapytania numerem portu na Arduino, przed uruchomieniem prosimy nauczyciela prowadzącego o sprawdzenie poprawności połączeń.
- 4.3. Podłączmy Arduino do portu usb i dokonujemy konfiguracji podłączenia w dostępnym oprogramowaniu, następnie uzupełniamy program o kod zawarty poniżej, kompilujemy i sprawdzamy działanie czujnika.

```

//code

#define trigPin ??
#define echoPin ??
#include <LiquidCrystal.h>
LiquidCrystal lcd(2, 3, 4, 5, 6, 7);

void setup() {
  lcd.begin(??, ??);
  lcd.setCursor(??, ??);
  pinMode(trigPin, OUTPUT); //Pin, do którego podłączymy trig jako wyjście
  pinMode(echoPin, INPUT); //a echo, jako wejście
  lcd.clear();
}

void loop() {
  long czas, dystans;
  digitalWrite(trigPin, LOW); // ustawienie stanu niskiego na pinie trig
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  czas = pulseIn(echoPin, HIGH); // funkcja licząca czas

  Dopisz linijkę obliczającą dystans zgodnie ze wzorem podanym poniżej

  lcd.print(dystans);
  lcd.print(" cm");

  delay(500);
  lcd.clear();
}

```

Metodę pomiaru opisuje prosta zależność, w której mierzony jest czas (T1) w sekundach od momentu wyzwolenia sygnału na TRIG do momentu pojawienia się odpowiedzi na pinie ECHO:

$$\text{odległość} = (T1 * \text{prędkość rozchodzenia się dźwięku}) / 2$$

# Załącznik 1

