Arduino II - 21.10.2022

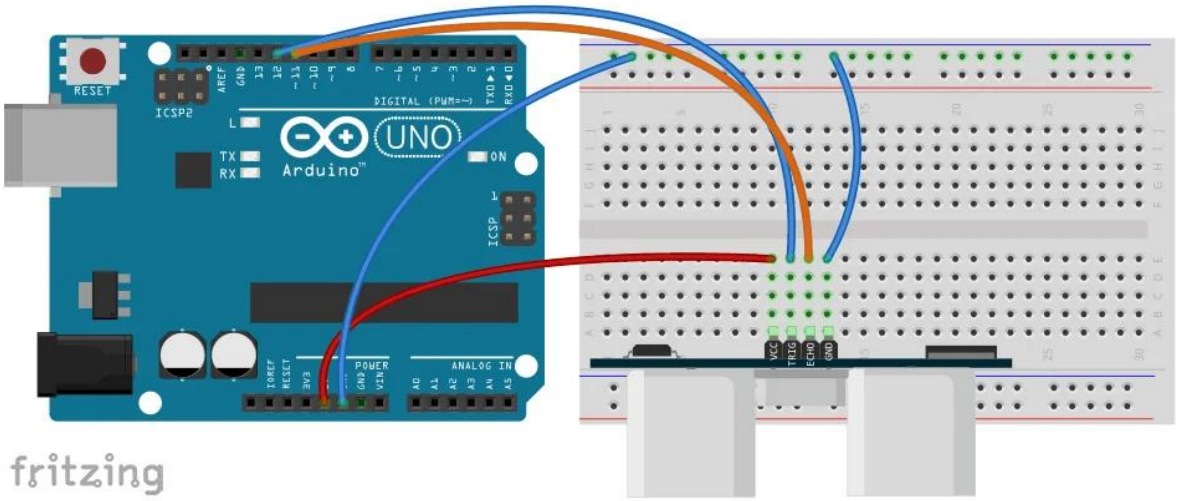
Zadanie 1. Czujnik ultradźwiękowy

Czujnik HC-SR04 składa się z nadajnika i odbiornika ultradźwięków. Na podstawie nasłuchiwania i odbijania się sygnału akustycznego (niesłyszalnego dla człowieka) można precyzyjnie określić odległość czujnika od przeszkody.

Trigger, to wejście wyzwalające. Gdy podamy na nie stan wysoki (przez min. 10 mikrosekund), to rozpocznie się pomiar odległości.

Natomiast z wyjścia echo odczytamy zmierzoną odległość. Maksymalny zasięg tego niepozornego układu deklarowany jest przez producenta na 4 m.

Miernik odległości (schemat podłączenia - VCC na 5V!, trigger pin 12, echo pin 11):



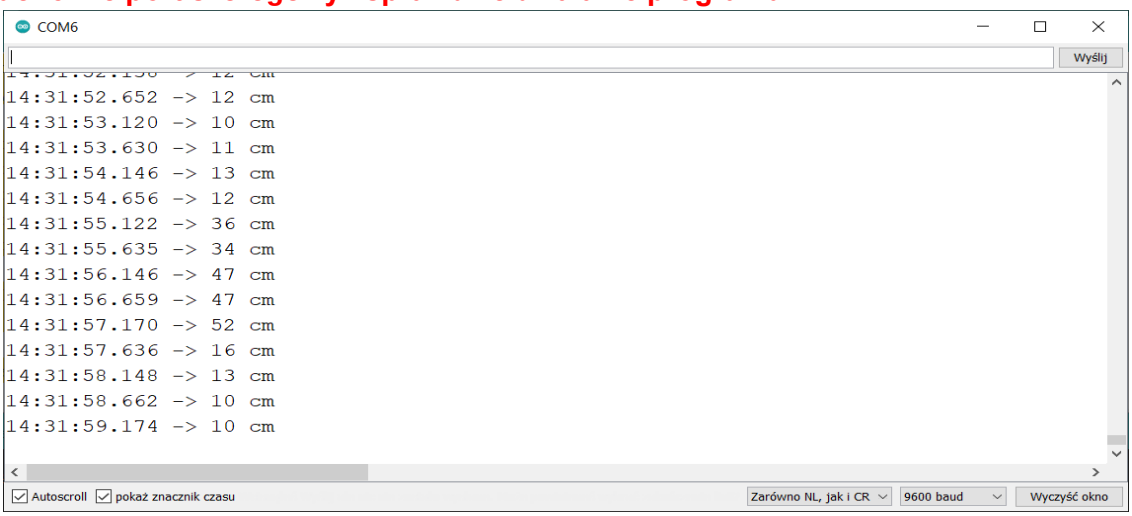
Przykładowy kod programu znajduje się na kolejnej stronie.

Przykładowy kod programu:

```
#define trigPin 12
#define echoPin 11

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT); //Pin, do którego podłączymy trig jako wyjście
  pinMode(echoPin, INPUT); //a echo, jako wejście
}

void loop() {
  Serial.print(zmierzOdleglosc());
  Serial.println(" cm");

  delay(500);
}

// FUNKCJA MIERZACA ODLEGLOSC W CM
int zmierzOdleglosc() {
  long czas, dystans;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  czas = pulseIn(echoPin, HIGH);
  dystans = czas / 58;

  return dystans;
}
```

pulseIn - prosta funkcja w Arudino służąca pomiarowi czasu trwania impulsu podanego na wejście.

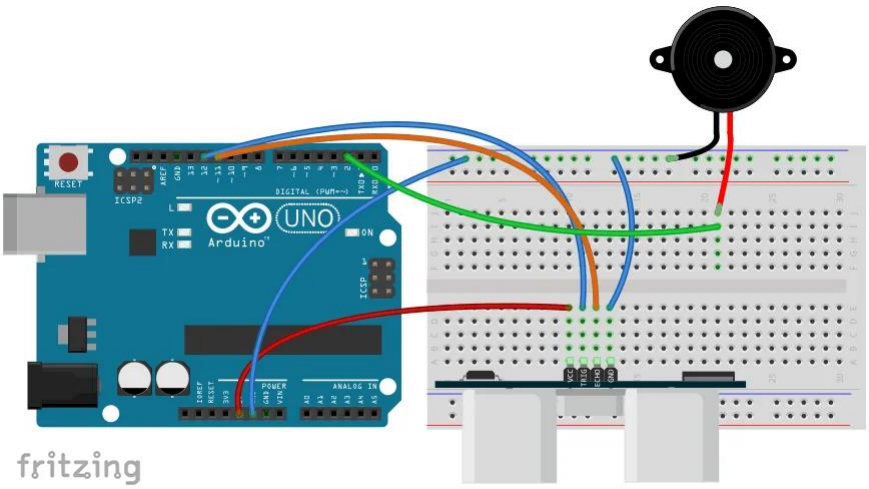
Uruchomić port szeregowy i sprawdzić działanie programu.



Zadanie 2. Buzzer

Buzzer ma dwa wyjścia + i -. Po podłączeniu na Vcc i z masą zacznie wydawać dźwięk. Buzzer jest elementem dwubiegunowym.

Schemat podłączenia:



Sprawdzić, czy obiekt znajduje się w określonej odległości od czujnika. Jeśli tak to włączy się alarm.

Przykładowy kod programu znajduje się na kolejnej stronie.

Przykładowy kod programu:

```
#define trigPin 12
#define echoPin 11

void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT); //Pin, do którego podłączymy trig jako wyjście
  pinMode(echoPin, INPUT); //a echo, jako wejście
  pinMode(2, OUTPUT); //Wyjście dla buzzera
}

void loop() {
  zakres(10, 25); //Włącz alarm, jeśli w odległości od 10 do 25 cm od czujnika jest przeszkoda
  delay(100);
}

int zmierzOdleglosc() {
  long czas, dystans;

  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  czas = pulseIn(echoPin, HIGH);
  dystans = czas / 58;

  return dystans;
}

void zakres(int a, int b) {
  int jakDaleko = zmierzOdleglosc();
  if ((jakDaleko > a) && (jakDaleko < b)) {
     digitalWrite(2, HIGH); //Włączamy buzzer
  } else {
     digitalWrite(2, LOW); //Wyłączamy buzzer, gdy obiekt poza zakresem
  }
}
```

UWAGA: Są 2 rodzaje buzerów. Jeden wymaga stałego sygnału 0 - brak, 1 - dźwięk. Jeśli będzie pykać/lekko „strzelać” należy zmienić kod obsługujący buzzer zgodnie z modyfikacjami na następnej stronie.

Modyfikacje w kodzie:

a) w funkcji **setup()**

```
void setup() {
  Serial.begin (9600);
  pinMode(trigPin, OUTPUT); //Pin, do którego podłączymy trig jako wyjście
  pinMode(echoPin, INPUT); //a echo, jako wejście
  pinMode(5, OUTPUT); //Wyjście dla buzzera zmiana 2 pin na 5
}
```

b) w funkcji **zakres()**

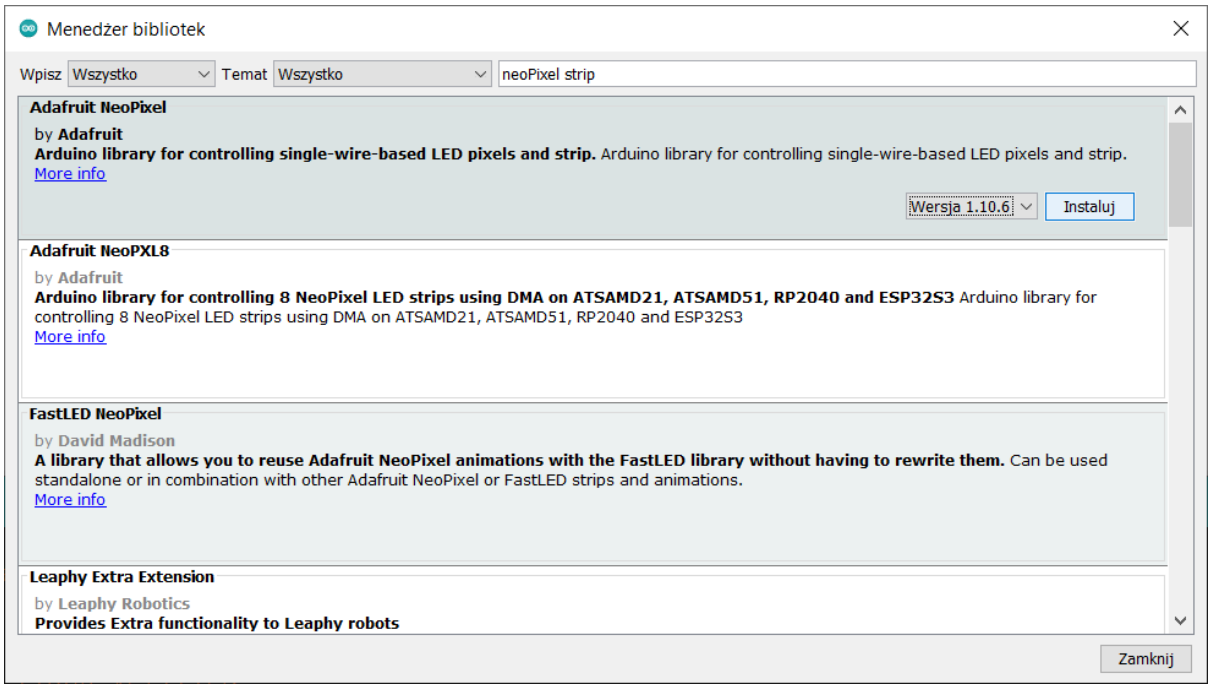
```
void zakres(int a, int b) {
  int jakDaleko = zmierzOdleglosc();
  if ((jakDaleko > a) && (jakDaleko < b)) {
     analogWrite(5, 230); //Włączamy buzzer 255 maksymalna wartość (zmiana funkcji z digitalWrite(2, HIGH)
  } else {
     analogWrite(5, 0); //Wyłączamy buzzer, gdy obiekt poza zakresem (zmiana funkcji z digitalWrite(2, LOW)
  }
}
```

Poniższa tabela oznacza, dla których pinów Arduino można ustawić piny do sterowania PWM.

Board	PWM Pins	PWM Frequency
Uno, Nano, Mini	3, 5, 6, 9, 10, 11	490 Hz (pins 5 and 6: 980 Hz)

Zadanie 3. Obsługa taśmy LED

Do zarządzania taśmą LED potrzebna jest biblioteka neoPixel strip. Sprawdzić czy ta biblioteka jest zainstalowana (np. Poprzez sprawdzenie jej dostępności w zakładce Przykłady).



Taśmę LED podłączyć następująco:
DI(DIN) - PIN 6
VCC - PIN 5V
GND - GND

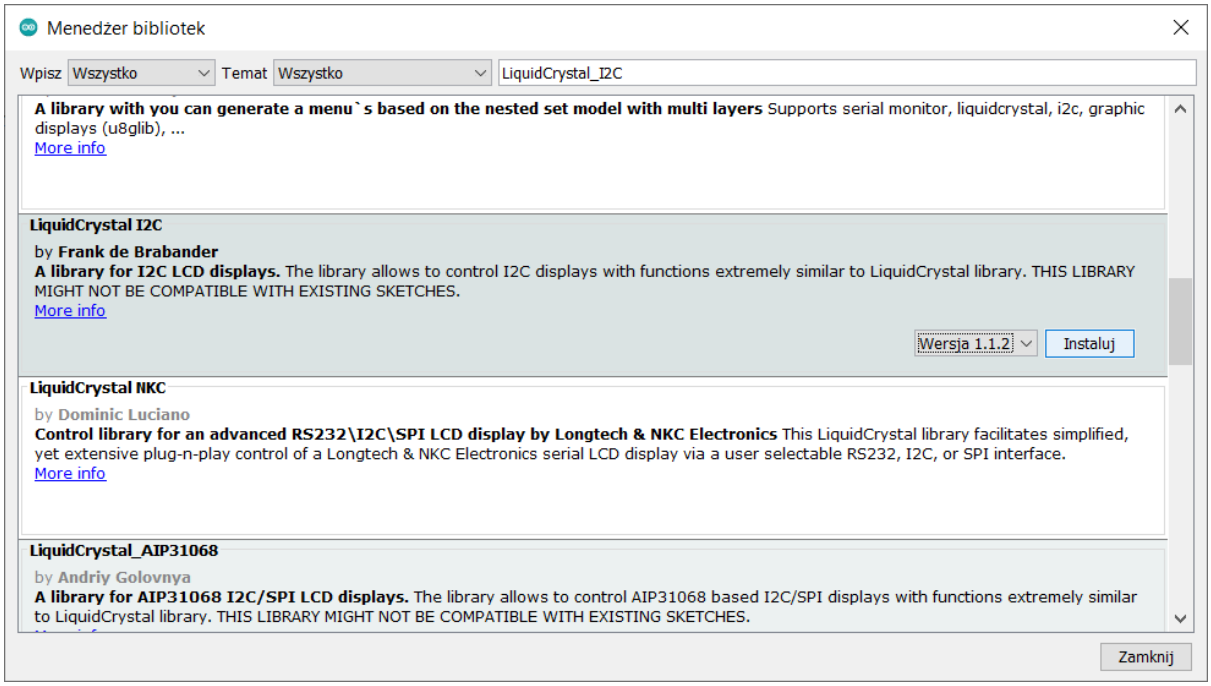
Przykładowy kod programu znajduje się na kolejnej stronie.

Przykładowy kod programu:

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
  #include <avr/power.h>
#endif
#define PIN        6
#define NUMPIXELS 8

Adafruit_NeoPixel pixels(NUMPIXELS, PIN, NEO_GRB + NEO_KHZ800);
#define DELAYVAL 500

void setup() {
#if defined(__AVR_ATtiny85__) && (F_CPU == 16000000)
  clock_prescale_set(clock_div_1);
#endif

  pixels.begin();
}

void loop() {
  pixels.clear();

  for(int i=0; i<NUMPIXELS; i++) {

    pixels.setPixelColor(i, pixels.Color(0, 150, 0));
    // kolory: Red - 255,0,0; Green - 0,255,0; Blue - 0,0,255
    pixels.show();
    delay(DELAYVAL);
  }
}
```

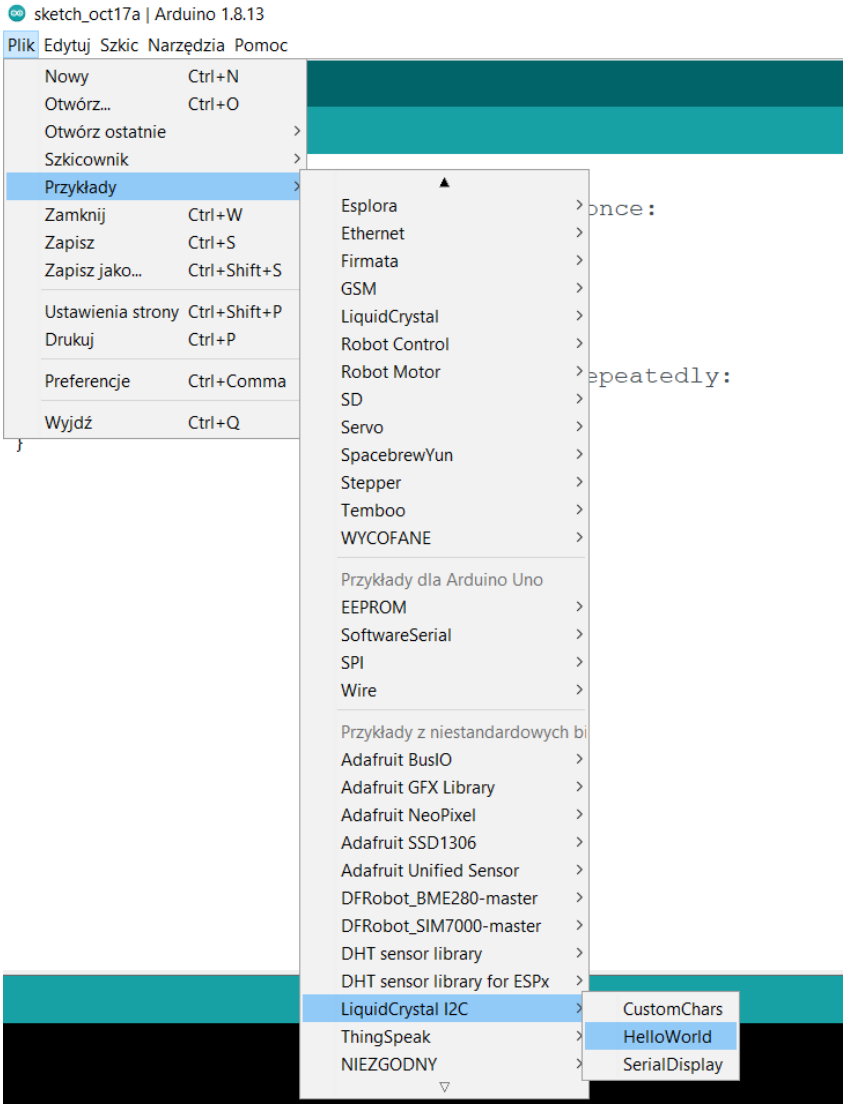
Zadanie 4. Wyświetlacz 16 znakowy, 2 segmentowy ze sterownikiem

Do zarządzania wyświetlaczem potrzebne jest biblioteka LiquidCrystal_I2C:



Podłączenie wyświetlacza:
VCC na 5V
GND
SDA -> SDA (druga strona płytki podpisane)
SCL -> SCL (druga strona płytki podpisane)

Wejść w pozycję z przykładami (rys. Kolejna strona). Uruchomić każdy na płytce Arduino i wyświetlaczu. Przeprowadzić modyfikacje w kodzie w celu lepszego zapoznania z działaniem wyświetlacza.


sketch_oct17a | Arduino 1.8.13
Plik Edytuj Szkic Narzędzia Pomoc
Nowy Ctrl+N
Otwórz... Ctrl+O
Otwórz ostatnie
Szkicownik
Przykłady
Zamknij Ctrl+W
Zapisz Ctrl+S
Zapisz jako... Ctrl+Shift+S
Ustawienia strony Ctrl+Shift+P
Drukuj Ctrl+P
Preferencje Ctrl+Comma
Wyjdź Ctrl+Q
Esplora
Ethernet
Firmata
GSM
LiquidCrystal
Robot Control
Robot Motor
SD
Servo
SpacebrewYun
Stepper
Temboo
WYCOFANE
Przykłady dla Arduino Uno
EEPROM
SoftwareSerial
SPI
Wire
Przykłady z niestandardowych bi
Adafruit BusIO
Adafruit GFX Library
Adafruit NeoPixel
Adafruit SSD1306
Adafruit Unified Sensor
DFRobot_BME280-master
DFRobot_SIM7000-master
DHT sensor library
DHT sensor library for ESPx
LiquidCrystal I2C
ThingSpeak
NIEZGODNY
CustomChars
HelloWorld
SerialDisplay
nce:
epeatedly:

Zadanie 5. Funkcja MILIS

Zapoznać się z artykułem:

https://forbot.pl/blog/kurs-arduino-ii-wielozadaniowosc-opoznienia-z-millis-id18418

Wykonać wybrane ćwiczenia (z uruchomieniem kodu na płytce) utrwalające wiedzę dotyczącą funkcji MILIS.

Zadanie 6. Przycisk monostabilny z obsługą MILIS

Stworzyć projekt z wykorzystaniem poznanych czujników/elementów wraz z przyciskiem monostabilnym z wykorzystaniem poznanej funkcji MILIS.