



Politechnika Białostocka
Wydział Elektryczny
Katedra Elektrotechniki, Energoelektroniki i Elektroenergetyki

Instrukcja
do pracowni specjalistycznej z przedmiotu

**Programowanie mikrokontrolerów
w języku wysokiego poziomu 1**

Kod przedmiotu: **TS1F1008**

(studia stacjonarne)

**ARDUINO - WSPÓŁPRACA PLATFORMY ARDUINO
Z URZĄDZENIAMI ZEWNĘTRZNYMI , CZ. 1**

Numer ćwiczenia

PMC_09

Autor:
dr inż. Jarosław Forenc

Białystok 2024

Spis treści

1. Opis stanowiska	3
1.1. Stosowana aparatura	3
1.2. Oprogramowanie	3
2. Wiadomości teoretyczne.....	3
2.1. Wstęp.....	3
2.2. Buzzer (brzęczyk piezoelektryczny)	4
2.3. Fotorезystor	5
2.4. Ultradźwiękowy czujnik odległości	7
2.5. Czujnik natężenia światła	8
3. Przebieg ćwiczenia.....	11
4. Literatura.....	12
5. Pytania kontrolne	12
6. Wymagania BHP.....	13

Materiały dydaktyczne przeznaczone dla studentów Wydziału Elektrycznego PB.

© Wydział Elektryczny, Politechnika Białostocka, 2024 (wersja 1.1)

Wszelkie prawa zastrzeżone. Żadna część tej publikacji nie może być kopiowana i odtwarzana w jakiegokolwiek formie i przy użyciu jakiegokolwiek środków bez zgody posiadacza praw autorskich.

1. Opis stanowiska

1.1. Stosowana aparatura

Podczas zajęć wykorzystywany jest komputer klasy PC z systemem operacyjnym Microsoft Windows 10/11 oraz platforma Arduino wraz z zestawem czujników.

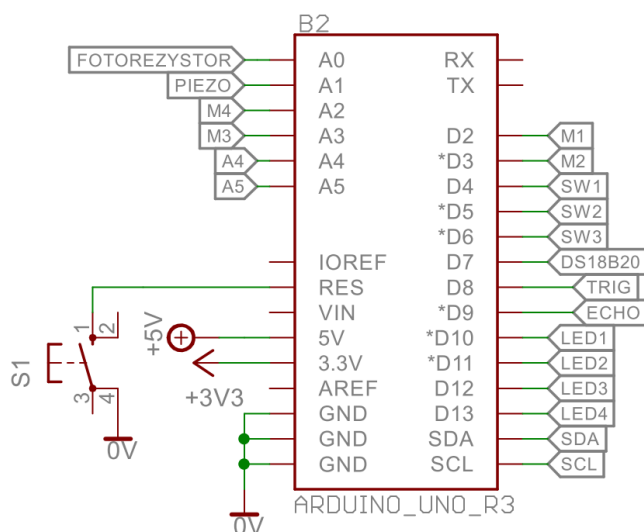
1.2. Oprogramowanie

Na komputerach zainstalowany jest edytor kodu źródłowego Visual Studio Code 1.92 (lub nowszy) wraz z rozszerzeniem (PlatformIO IDE for VSCode).

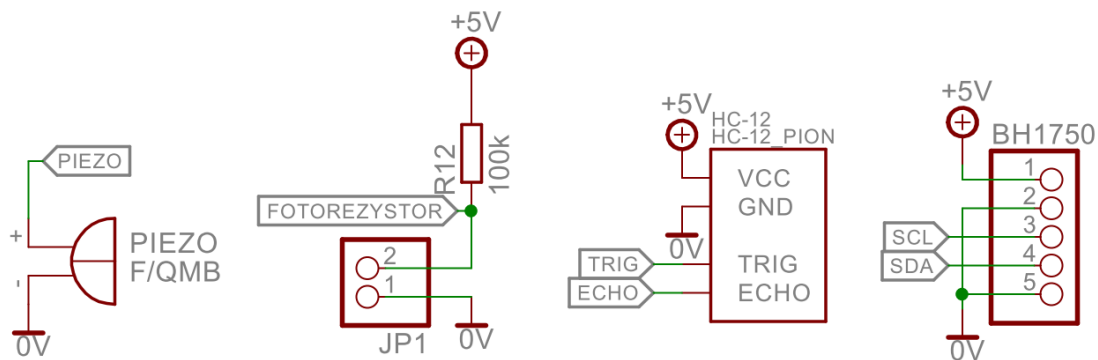
2. Wiadomości teoretyczne

2.1. Wstęp

W instrukcji zostanie przedstawione wykorzystanie następujących elementów i czujników współpracujących z modułem Arduino: buzzera (brzęczyka piezoelektrycznego), fotorezystora, ultradźwiękowego czujnika odległości (HC-SR04) oraz czujnika natężenia światła (BH1750). Schemat podłączenia elementów i czujników do modułu Arduino przedstawiają Rys. 1 i Rys. 2.



Rys. 1. Schemat podłączenia wyprowadzeń modułu Arduino



Rys. 2. Schemat podłączenia elementów i czujników

2.2. Buzzer (brzęczyk piezoelektryczny)

Buzzer umożliwia odtwarzanie dźwięków o określonej częstotliwości. W poniższym programie, po naciśnięciu przycisku **SW1**, odgrywana jest melodia składająca się z trzech dźwięków o różnej częstotliwości.

Program obsługujący buzzer.

```
#include <Arduino.h>

#define BUZZER_PIN A1
#define SW1_PIN 4

void setup() {
  pinMode(SW1_PIN, INPUT);
}

void loop() {
  if(digitalRead(SW1_PIN) == LOW)
  {
    delay(200);
    tone(BUZZER_PIN, 1000);
    delay(200);
    tone(BUZZER_PIN, 500);
    delay(200);
    tone(BUZZER_PIN, 750);
    delay(200);
    noTone(BUZZER_PIN);
  }
}
```

Zgodnie z Rys. 1 buzzer jest podłączony do pinu analogowego **A1**. Definiujemy dwie stałe preprocesora: **BUZZER_PIN** oraz **SW1_PIN**, które będą przechowywać numery odpowiednich pinów. W funkcji **setup()** ustawiamy pin **A1** jako wejście (**INPUT**). W funkcji **loop()** sprawdzamy stan wciśnięcia przycisku za pomocą funkcji **digitalRead()**. Jeśli stan jest niski (**LOW**), to odtwarzamy dźwięki, trzykrotnie wywołując funkcję **tone()**. Funkcja ta ma dwa argumenty: pierwszy określa numer pinu, do którego podłączony jest buzzer, a drugi - częstotliwość dźwięku w Hz. Długość trwania poszczególnych dźwięków definiujemy za pomocą funkcji **delay()**. Wyłączenie odtwarzania dźwięku następuje poprzez wywołanie funkcji **noTone()** z argumentem określającym numer pinu.

2.3. Fotorezystor

Fotorezystor jest to element półprzewodnikowy, którego rezystancja zmienia się w zależności od natężenia padającego na niego światła. Wraz ze wzrostem natężenia oświetlenia rezystancja fotorezystora maleje. Poniższy program pokazuje sposób wykorzystania fotorezystora podłączonego do platformy Arduino.

Program obsługujący fotorezystor.

```
#include<Arduino.h>

#define PHR_PIN A0
int value = 0;

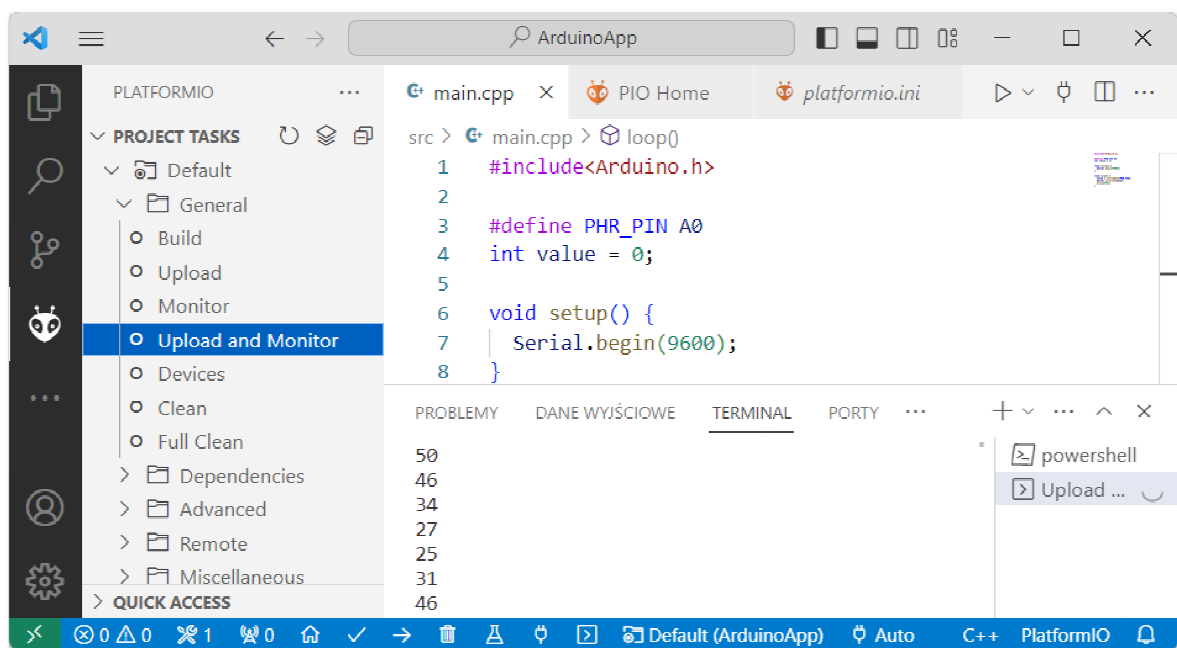
void setup() {
  Serial.begin(9600);
}

void loop() {
  value = analogRead(PHR_PIN);
  Serial.println(value);
  delay(250);
}
```

Fotorezystor jest podłączony do analogowego wyjścia **A0** (Rys. 1). Arduino posiada 10-bitowy przetwornik analogowo-cyfrowy. Napięcie w zakresie od 0 do

5 V doprowadzone do wejścia analogowego Arduino jest przetwarzane przez przetwornik na liczbę całkowitą z przedziału od 0 do 1023. Liczbę tę można odczytać, wywołując funkcję **analogRead()**. W powyższym programie definiujemy stałą preprocesora **PHR_PIN**, która przechowuje numer pinu **A0**. Następnie deklarujemy zmienną **value** do przechowywania wartości odczytanej z wejścia analogowego. Do wyświetlenia tej wartości wykorzystano okno monitora portu szeregowego. W związku z tym w funkcji **setup()** wywoływana jest funkcja **Serial.begin()**, która ustawia prędkość transmisji dla portu szeregowego. W funkcji **loop()** wartość odczytana z przetwornika analogowo-cyfrowego jest wyświetlana w oknie monitora portu szeregowego za pomocą funkcji **Serial.println()**.

Obserwacja wyników w oknie monitora portu szeregowego wymaga innego sposobu uruchomienia programu. Należy kliknąć ikonę **PlatformIO**, a następnie wybrać opcję **PROJECT TASK** → **Default** → **General** → **Upload and Monitor**. Po przesłaniu programu do modułu Arduino i jego uruchomieniu, wartości odczytywane z portu **A0** będą wyświetlane w oknie **TERMINAL** (Rys. 3).



Rys. 3. Obserwacja wyników uruchomienia programu w oknie monitora portu szeregowego

2.4. Ultradźwiękowy czujnik odległości

Ultradźwiękowy czujnik odległości HC-SR04 umożliwia pomiar odległości od obiektu poprzez wysyłanie i odbieranie fali ultradźwiękowej. Czujnik ten składa się z dwóch sparowanych ze sobą elementów: nadajnika i odbiornika, oraz układu sterowania i generatora sygnału pomiarowego. Nadajnik wysyła sygnał ultradźwiękowy, który odbija się od obiektu i wraca do odbiornika. Odległość od obiektu jest obliczana na podstawie różnicy czasu, który upływa od chwili wysłania sygnału pomiarowego, do chwili jego odbioru. Poniżej przedstawiony jest przykładowy program obsługujący ultradźwiękowy czujnik odległości.

Program obsługujący ultradźwiękowy czujnik odległości HC-SR04.

```
#include<Arduino.h>

#define TRIG_PIN 8
#define ECHO_PIN 9

long duration;
int distance;

void setup() {
  pinMode(TRIG_PIN, OUTPUT);
  pinMode(ECHO_PIN, INPUT);
  Serial.begin(9600);
}

void loop() {
  digitalWrite(TRIG_PIN, LOW);
  delayMicroseconds(2);
  digitalWrite(TRIG_PIN, HIGH);
  delayMicroseconds(10);
  digitalWrite(TRIG_PIN, LOW);
  duration = pulseIn(ECHO_PIN, HIGH);
  distance = duration * 0.034 / 2;
  Serial.print("Odlegosc: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(250);
}
```

Wyprowadzenie **Trig** czujnika jest wejściem sygnału wyzwalającego cykl pomiarowy i jest podłączone do pinu nr **8** modułu Arduino. Wyprowadzenie **Echo** czujnika jest wyjściem sygnału odwzorowującego czas przemieszczania się fali ultradźwiękowej i jest podłączone do pinu nr **9** modułu Arduino. Po podaniu na wejście **Trig** stanu wysokiego o czasie trwania 10 μs , wbudowany generator wysyła na nadajnik czujnika sygnał ultradźwiękowy o częstotliwości 40 kHz przez czas 200 μs . Po zakończeniu wysyłania tego sygnału pin **Echo** zmienia stan na wysoki, rozpoczynając pomiar. Wyemitowana fala ultradźwiękowa jest wysyłana i gdy napotka na drodze obiekt, zostaje odbita w stronę odbiornika. Zmierzony czas trwania stanu wysokiego na pinie **Echo** (za pomocą funkcji **pulseIn()**) jest proporcjonalny do odległości czujnika od przeszkody. Odległość (w cm) jest obliczana na podstawie wzoru:

$$d = \frac{t [\mu\text{s}] \cdot v \left[\frac{\text{cm}}{\mu\text{s}} \right]}{2}$$

gdzie: **d** - zmierzona odległość w cm, **t** - zmierzony czas w μs , **v** - prędkość dźwięku w powietrzu ($v = 340 \text{ m/s} = 0,034 \text{ cm}/\mu\text{s}$)

2.5. Czujnik natężenia światła

Cyfrowy czujnik natężenia światła BH1750 umożliwia pomiar natężenia światła zarówno w pomieszczeniach, jak i na otwartej przestrzeni, w zakresie od 1 do 65535 luksów. Czujnik ten komunikuje się z modułem Arduino za pomocą interfejsu **I2C**, wykorzystując dwie linie: linię danych - **SDA** (Serial Data Line) oraz linię zegarową - **SCL** (Serial Clock Line). Przykładowy program pokazujący sposób wykorzystania czujnika natężenia światła przedstawiony jest poniżej.

Program obsługujący czujnik natężenia światła BH1750.

```
#include <Arduino.h>
#include <BH1750.h>
#include <Wire.h>

BH1750 LightMeter(0x23);
```



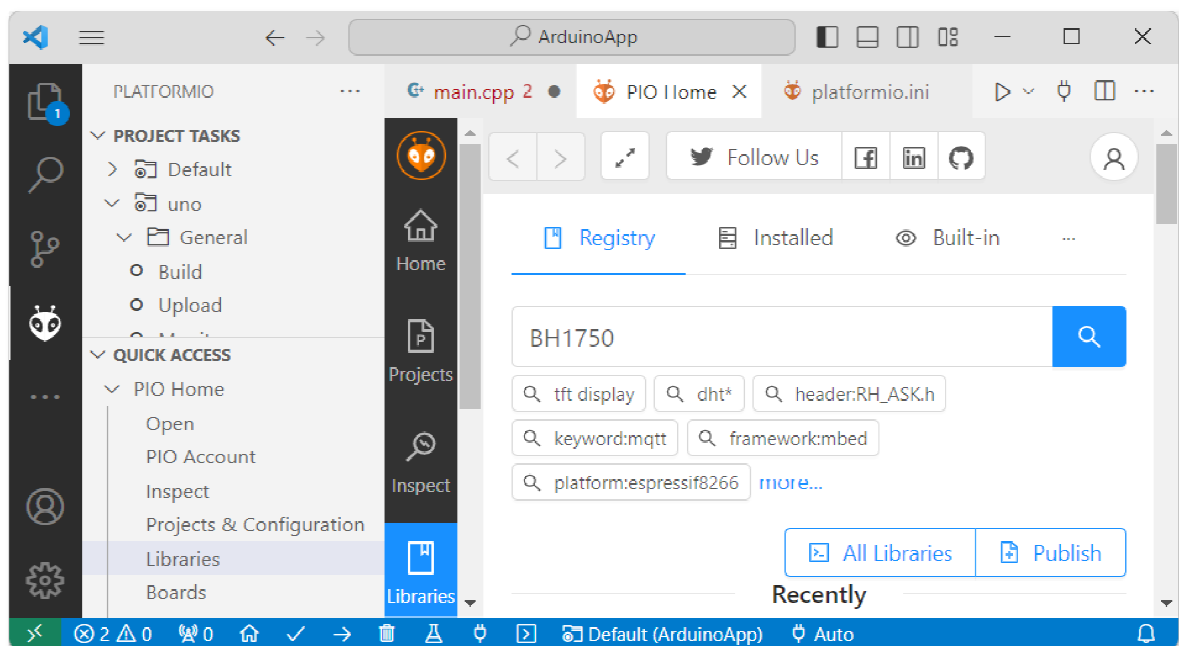
```

void setup() {
  Serial.begin(9600);
  Wire.begin();
  LightMeter.begin();
}

void loop() {
  float lux = LightMeter.readLightLevel();
  Serial.print("Natężenie światła: ");
  Serial.print(lux);
  Serial.println(" lx");
  delay(250);
}

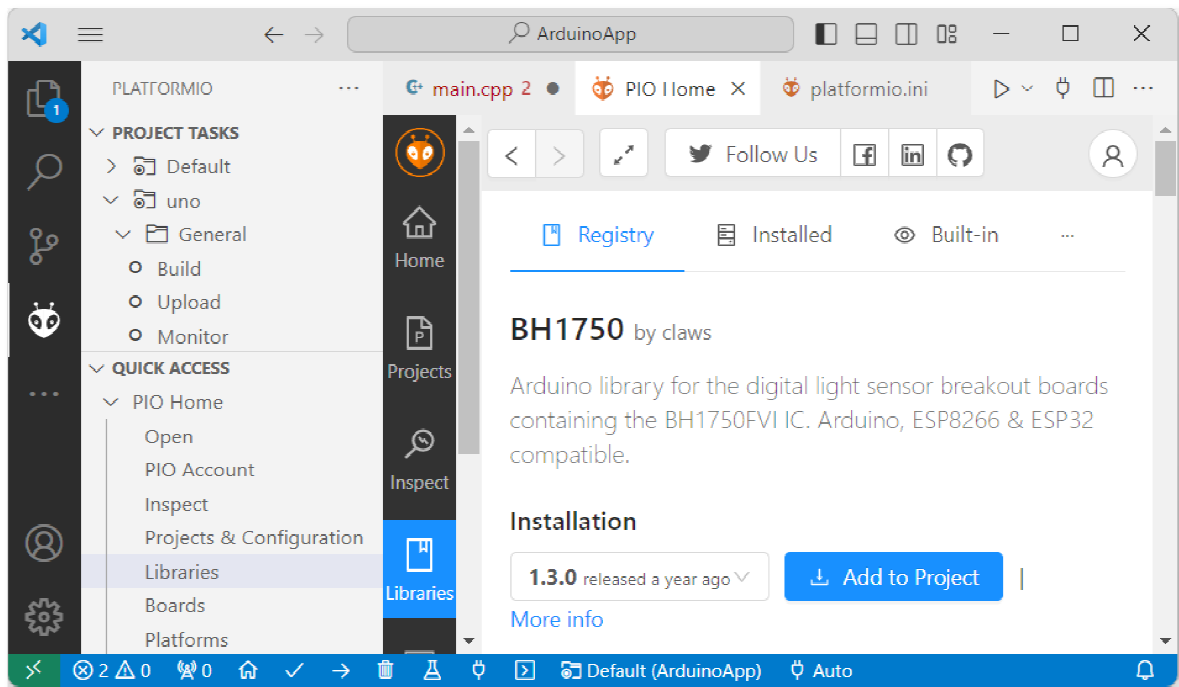
```

Do obsługi czujnika natężenia światła BH1750, konieczne jest zainstalowanie dodatkowej biblioteki. W tym celu klikamy ikonę **PlatformIO** i wybieramy opcję **QUICK ACCESS → PIO Home → Libraries**. W oknie wyszukiwania wpisujemy nazwę czujnika BH1750 (Rys. 4).



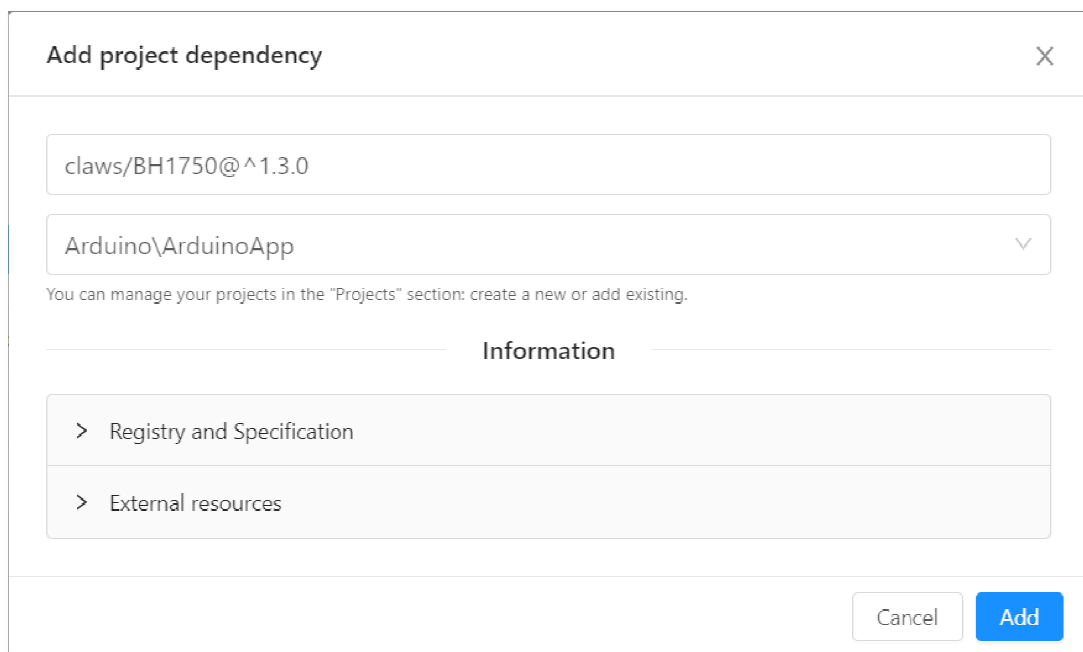
Rys. 4. Wyszukanie biblioteki obsługującej czujnik natężenia światła BH1750

Na liście wyników wyszukiwania wybieramy pierwszą bibliotekę o nazwie **BH1750 by claws** i instalujemy ją, klikając przycisk **Add to Project** (Rys. 5).



Rys. 5. Instalacja biblioteki BH1750 by claws

W kolejnym oknie wybieramy projekt, do którego ma być dodana biblioteka **BH1750 by claws** (Rys. 6).



Rys. 6. Wybór projektu, do którego ma być dodana biblioteka BH1750 by claws

Po kliknięciu przycisku **Add**, w pliku konfiguracyjnym **platformio.ini** zostanie dodany wiersz:

```
lib_deps = claws/BH1750@^1.3.0
```

zaś w pliku `.cpp` z kodem źródłowym zostaną dodane dwie biblioteki:

```
#include <BH1750.h>  
#include <Wire.h>
```

3. Przebieg ćwiczenia

Na pracowni specjalistycznej należy wykonać wybrane zadania wskazane przez prowadzącego zajęcia. W różnych grupach mogą być wykonywane różne zadania.

1. Napisz program wyświetlający w oknie monitora portu szeregowego wartość zwracaną przez fotorezystor podłączony do modułu Arduino. Sprawdź i podaj w sprawozdaniu zwracaną wartość przy różnych warunkach oświetlenia fotorezystora: gdy jest zasłonięty, przy normalnym oświetleniu oraz przy oświetleniu lampą telefonu komórkowego.
2. Napisz program wyświetlający w oknie monitora portu szeregowego wartość zwracaną przez ultradźwiękowy czujnik odległości podłączony do modułu Arduino. Stosując ultradźwiękowy czujnik odległości, zmierz i podaj w sprawozdaniu wysokość, szerokość oraz głębokość biurka, przy którym pracujesz. Porównaj otrzymane wyniki z pomiarami wykonanymi przez inne grupy. Porównaj także otrzymane wyniki z pomiarami wykonanymi innymi przyrządami, np. miarą składaną drewnianą (metrówką), taśmą mierniczą.
3. Napisz program wyświetlający w oknie monitora portu szeregowego wartość zwracaną przez cyfrowy czujnik natężenia światła podłączony do modułu Arduino. Sprawdź i podaj w sprawozdaniu zwracaną wartość przy różnych warunkach oświetlenia: gdy czujnik jest zasłonięty, przy normalnym oświetleniu oraz przy oświetleniu lampą telefonu komórkowego. Sprawdź, czy oświetlenie

biurka, przy którym pracujesz, jest odpowiednie (znajdź w Internecie wymaganą wartość natężenia oświetlenia).

4. Napisz program, w którym diody LED sygnalizują odległość ultradźwiękowego czujnika odległości od obiektu. Jeśli odległość jest większa niż 20 cm, to powinna świecić się zielona dioda LED. Jeśli odległość jest mniejsza lub równa 20 cm, ale większa niż 10 cm, to powinna świecić się tylko żółta dioda LED. Jeśli odległość jest mniejsza lub równa 10 cm, to powinna świecić się tylko czerwona dioda LED.
5. Napisz program, w którym buzzer sygnalizuje odległość ultradźwiękowego czujnika odległości od obiektu. Jeśli odległość jest większa niż 20 cm, to buzzer nie wydaje żadnego dźwięku. Jeśli odległość jest mniejsza lub równa 20 cm, ale większa niż 10 cm, to buzzer wydaje przerywany dźwięk. Jeśli odległość jest mniejsza lub równa 10 cm, to buzzer wydaje ciągły dźwięk.

4. Literatura

- [1] Prata S.: Język C. Szkoła programowania. Wydanie VI. Helion, Gliwice, 2016.
- [2] Wrotek W.: Arduino od podstaw. Helion, Gliwice, 2023.
- [3] Monk S.: Arduino dla początkujących. Podstawy i szkice. Helion, Gliwice, 2019.
- [4] Evans M., Noble J., Hochenbaum J.: Arduino w akcji. Helion, Gliwice, 2014.
- [5] Kernighan B.W., Ritchie D.M.: Język ANSI C. Programowanie. Wydanie II. Helion, Gliwice, 2010.
- [6] <https://code.visualstudio.com/> - Visual Studio Code
- [7] <https://www.arduino.cc/reference/en/> - Arduino Language Reference

5. Pytania kontrolne

1. Wyjaśnij, w jaki sposób można obsłużyć buzzer w programie.

2. Wyjaśnij, w jaki sposób można obsłużyć fotorezystor w programie.
3. Wyjaśnij, w jaki sposób można obsłużyć ultradźwiękowy czujnik odległości w programie.
4. Wyjaśnij, w jaki sposób można obsłużyć czujnik natężenia światła w programie.

6. Wymagania BHP

Warunkiem przystąpienia do praktycznej realizacji ćwiczenia jest zapoznanie się z instrukcją BHP i instrukcją przeciwpożarową oraz przestrzeganie zasad w nich zawartych.

W trakcie zajęć laboratoryjnych należy przestrzegać następujących zasad.

- Sprawdzić, czy urządzenia dostępne na stanowisku laboratoryjnym są w stanie kompletnym, nie wskazującym na fizyczne uszkodzenie.
- Jeżeli istnieje taka możliwość, należy dostosować warunki stanowiska do własnych potrzeb, ze względu na ergonomię. Monitor komputera ustawić w sposób zapewniający stałą i wygodną obserwację dla wszystkich członków zespołu.
- Sprawdzić prawidłowość połączeń urządzeń.
- Załączenie komputera może nastąpić po wyrażeniu zgody przez prowadzącego.
- W trakcie pracy z komputerem zabronione jest spożywanie posiłków i picie napojów.
- W przypadku zakończenia pracy należy zakończyć sesję przez wydanie polecenia wylogowania. Zamknięcie systemu operacyjnego może się odbywać tylko na wyraźne polecenie prowadzącego.
- Zabronione jest dokonywanie jakichkolwiek przełączeń oraz wymiana elementów składowych stanowiska.
- Zabroniona jest zmiana konfiguracji komputera, w tym systemu operacyjnego i programów użytkowych, która nie wynika z programu zajęć i nie jest wykonywana w porozumieniu z prowadzącym zajęcia.

- W przypadku zaniku napięcia zasilającego należy niezwłocznie wyłączyć wszystkie urządzenia.
- Stwierdzone wszelkie braki w wyposażeniu stanowiska oraz nieprawidłowości w funkcjonowaniu sprzętu należy przekazywać prowadzącemu zajęcia.
- Zabrania się samodzielnego włączania, manipulowania i korzystania z urządzeń nie należących do danego ćwiczenia.
- W przypadku wystąpienia porażenia prądem elektrycznym należy niezwłocznie wyłączyć zasilanie stanowiska. Przed odłączeniem napięcia nie dotykać porażonego.