



Politechnika
Wrocławska

Katedra Metrologii Elektronicznej i Fotonicznej

Optoelektronika 2 - Projekt

Jednokierunkowa Transmisja Światłowodowa

Grupa nr **P21**

Yaroslav Basiuk 257786

Yevhen Shpatakov 276739

Wrocław 2025

Wydział Elektroniki,
Fotoniki i Mikrosystemów

$\hbar\omega$

Spis treści

1 Wstęp	3
1.1 Światło	4
1.2 Zjawisko Fotoelektryczne	4
1.3 Laser półprzewodnikowy	5
2 ZAŁOŻENIA	6
2.1 Założenia projektowe	6
2.2 Założenia konstrukcyjne	6
3 OPIS CZEŚCI SPRZĘTOWEJ	7
3.1 Schemat blokowy	7
3.2 Nadajnik	7
3.2.1 Schemat ideowy	7
3.2.2 Dobór komponentów	9
Mikrokontroler	9
Zasilanie	11
Wzmocnienie	11
3.2.3 Montaż układu	12
3.3 Odbiornik	13
3.3.1 Schemat ideowy	13
3.3.2 Dobór komponentów	14
Mikrokontroler oraz Zasilanie	14
Odbiornik światłowodowy	14
Wyświetlacz	15
3.3.3 Montaż układu	16
4 OPIS CZEŚCI PROGRAMOWEJ	17
4.1 Algorytm główny	17
4.1.1 Nadajnik	17
4.1.2 Odbiornik	19
4.2 Opis kluczowych funkcji	21
4.2.1 Nadajnik	21

Deklaracja bibliotek i obiektów	21
Funkcja <code>setup()</code>	21
Funkcja <code>loop()</code>	22
4.2.2 Odbiornik	23
Deklaracja bibliotek i obiektów	23
Funkcja <code>setup()</code>	23
Funkcja <code>loop()</code>	24
5 Uruchomienie, pomiary testowe i kalibracja	26
5.1 Uruchomienie	26
5.2 Pomiary testowe i kalibracja	26
5.3 Specyfikacja urządzenia	27
5.3.1 Nadajnik	27
5.3.2 Odbiornik	27
6 INSTRUKCJA OBSŁUGI	28
6.1 Nadajnik	28
6.1.1 Opis złącz	28
6.1.2 Uruchomienie i konfiguracja	28
6.2 Odbiornik	29
6.2.1 Opis złącz	29
6.2.2 Uruchomienie	30
6.3 Przesyłanie danych.	31
7 PODSUMOWANIE	33
7.1 Przedmiot pracy	33
7.2 Osiągnięte wyniki	33
7.3 Napotkane problemy	33
7.4 Wizje na przyszłość	33
8 BIBLIOGRAFIA	34
9 DODATKI	35
9.1 Nadajnik	35
9.2 Odbiornik	36

1 Wstęp

Rozwój technologii optoelektronicznych wpływa na wiele dziedzin współczesnego życia, w tym na telekomunikację, przemysł, medycynę oraz systemy zabezpieczeń. Jednym z kluczowych obszarów zastosowania tej technologii jest światłowodowa transmisja danych, która umożliwia szybkie i niezawodne przesyłanie informacji na duże odległości. Projekty dotyczące jednokierunkowej transmisji światłowodowej znajdują swoje zastosowanie tam, gdzie wymagana jest wysoka izolacja sygnału w celu zapewnienia bezpieczeństwa transmisji oraz ochrony przed potencjalnymi zagrożeniami, takimi jak nieuprawniony dostęp czy interferencje elektromagnetyczne.

Celem niniejszego projektu jest zbadanie zasad działania oraz potencjalnych zastosowań jednokierunkowej transmisji światłowodowej, a także przeprowadzenie analizy parametrów technicznych związanych z tego typu systemami. Realizacja projektu pozwoli na zrozumienie kluczowych aspektów projektowania układów optycznych i pomoże włączyć zdobytą wiedzę do praktycznych zastosowań.

Jednokierunkowa transmisja światłowodowa, zwana również izolowaną transmisją optyczną, polega na przesyłaniu sygnału w jednym kierunku bez możliwości jego odbioru lub zakłócenia przez zewnętrzne źródła. Tego typu transmisja jest realizowana za pomocą dedykowanych światłowodów oraz komponentów optoelektronicznych, takich jak diody laserowe, fotodiody czy optyczne izolatory. Główne zalety technologii światłowodowej obejmują wysoką przepustowość, niską utratę sygnału oraz odporność na zakłócenia elektromagnetyczne.

Rozwój systemów jednokierunkowej transmisji znajduje szczególne zastosowanie w infrastrukturze krytycznej, takiej jak sieci energetyczne, systemy nadzoru i monitorowania, a także w branży wojskowej i medycznej. W takich aplikacjach kluczowe znaczenie ma bezpieczeństwo danych oraz eliminacja ryzyka zakłóceń. W niniejszym projekcie omówione zostaną zasady działania systemów optycznych, przeanalizowane zostaną ich zalety oraz ograniczenia, a także zaprezentowane zostaną możliwości ich zastosowania w praktyce.

1.1 Światło

Światło w ujęciu potocznym odnosi się do widzialnej części promieniowania elektromagnetycznego, odbieranego przez ludzkie oko, z zakresem długości fal wynoszącym 380–780 nm (czasem podaje się węższy zakres, np. 400–700 nm). W nauce natomiast światło to część promieniowania optycznego, obejmującego ultrafiolet, światło widzialne i podczerwień, z zakresem fal od 100 nm do 1 mm, zgodnie z prawami optyki geometrycznej i falowej.

Przykłady pokazujące szerszy zakres światła:

- Substancje barwiące mogą pływać pod wpływem bliskiego ultrafioletu.
- Wzrost temperatury można zaobserwować w niewidocznych częściach widma.
- Zwierzęta widzą zakresy światła wykraczające poza ludzką percepcję.

Światło w ujęciu współczesnej optyki wykazuje dualizm korpuskularno-falowy, będąc zarówno falą elektromagnetyczną, jak i strumieniem fotonów. W próżni światło porusza się z prędkością c , wynoszącą 299 792 458 m/s, która zmniejsza się w ośrodkach materialnych w zależności od ich współczynnika załamania. Nauka zajmująca się światłem to optyka.

1.2 Zjawisko Fotoelektryczne

Efekt fotoelektryczny (fotoemisja) to zjawisko fizyczne polegające na emisji elektronów z powierzchni materiału pod wpływem promieniowania elektromagnetycznego o odpowiedniej częstotliwości. Elektrony emitowane w tym procesie nazywane są fotoelektronami, a ich energia kinetyczna zależy od częstotliwości światła, a nie jego natężenia.

Zjawisko to występuje w dwóch formach: zewnętrznej, gdy elektrony są emitowane na zewnątrz materiału, oraz wewnętrznej, gdy promieniowanie powoduje przenoszenie nośników ładunku między pasmami energetycznymi, co prowadzi do fotoprzewodnictwa. W gazach może zachodzić fotojonizacja.

Wyjaśnienie efektu fotoelektrycznego przez Alberta Einsteina w 1905 roku, oparte na hipotezie kwantów światła (fotonów), było kluczowe dla rozwoju teorii korpuskularnofalowej materii. Einstein otrzymał za to Nagrodę Nobla w 1921 roku.

1.3 Laser półprzewodnikowy

Laser półprzewodnikowy, nazywany również diodą laserową, to urządzenie wykorzystujące złącze p-n jako ośrodek czynny, w którym emisja promieniowania następuje pod wpływem przepływającego prądu elektrycznego. Charakteryzuje się kompaktowymi wymiarami, możliwością łatwej modulacji prądu o wysokiej częstotliwości (nawet do gigaherców) oraz zdolnością generowania promieniowania w szerokim zakresie długości fal, od podczerwieni do fioletu.

Działanie opiera się na rekombinacji promienistej w złączu p-n, która uwalnia fotony, oraz na wzmacnianiu promieniowania w rezonatorze Fabry'ego-Perota. Aby doszło do akcji laserowej, konieczne jest osiągnięcie progu prądowego. Wady, takie jak szerokie widmo czy zmienność parametrów pod wpływem temperatury, można ograniczyć dzięki zastosowaniu układów z ujemnym sprzężeniem zwrotnym.

Pierwsze obserwacje elektroluminescencji w złączu półprzewodnikowym miały miejsce w 1952 roku, a już w 1962 roku skonstruowano pierwszy laser półprzewodnikowy na bazie arsenku galu (GaAs). Lasery półprzewodnikowe mogą być złączowe, w których emisja odbywa się w obszarze złącza, lub bezzłączowe, wykonane z jednorodnego materiału. Znajdują szerokie zastosowanie, m.in. w telekomunikacji światłowodowej i urządzeniach optycznych.

2 ZAŁOŻENIA

2.1 Założenia projektowe

Celem projektu jest stworzenie prostego systemu transmisji światłowodowej tekstu w trybie jednokierunkowym, umożliwiającego przesyłanie danych z komputera do wyświetlacza tekstowego. Przesyłany typ danych `String`.

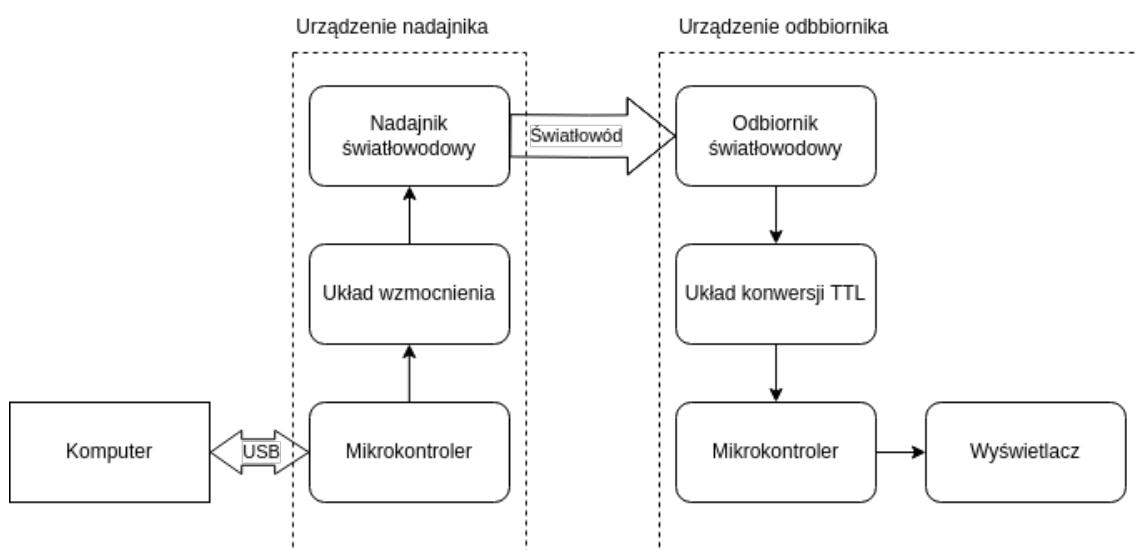
2.2 Założenia konstrukcyjne

Do realizacji projektu wykorzystano dostępne komponenty optoelektroniczne: nadajnik HFBR-1414Z oraz odbiornik HFBR-2412Z, które pracują na długości fali określonej przez ich specyfikację (około 820 nm). Transmisja odbywa się za pomocą światłowodu wyposażonego w złączki ST. Dane tekstowe są przesyłane z komputera przez interfejs serial do Arduino Nano połączonego z nadajnikiem, a następnie przesyłane przez światłowód do odbiornika podłączonego do drugiego Arduino Nano. Ostatecznie odebrane dane są wyświetlane na wyświetlaczu LCD 16x2. Zasilanie wszystkich elementów zapewnia interfejs USB, co upraszcza konstrukcję i minimalizuje potrzebę dodatkowych komponentów. Połączenie elementów odbywa się w sposób możliwie najprostszy, z naciskiem na funkcjonalność i łatwość montażu. Projekt został zaplanowany jako podstawowy układ transmisji światłowodowej, co pozwala na zdobycie doświadczenia w pracy z technologią światłowodową.

3 OPIS CZEŚCI SPRZĘTOWEJ

3.1 Schemat blokowy

Na rysunku 1 przedstawiono schemat blokowy jednokierunkowego systemu transmisji światłowodowej. System rozpoczyna się od komputera, który pełni funkcję źródła danych tekstowych. Dane te są przesyłane przez interfejs USB do mikrokontrolera zintegrowanego z nadajnikiem. Mikrokontroler przetwarza dane i przekazuje je do nadajnika światłowodowego, który konwertuje sygnał elektryczny na optyczny. Następnie sygnał optyczny jest przesyłany przez światłowód do odbiornika, gdzie następuje jego konwersja z powrotem na sygnał elektryczny. Sygnał ten trafia do układu konwersji TTL, który dostosowuje poziomy logiczne do mikrokontrolera odbiornika. Ostatnim etapem jest wyświetlacz, który prezentuje dane tekstowe użytkownikowi.



Rysunek 1. Schemat blokowy projektu.

3.2 Nadajnik

3.2.1 Schemat ideowy

Na rysunku 2 przedstawiono schemat nadajnika światłowodowego. Nadajnik zawiera laser półprzewodnikowy w obudowie kompatybilnej ze złączką ST. Prąd pracy lasera wynosi około 100 mA, podczas gdy wyprowadzenia GPIO mogą dostarczyć maksymalnie 20 mA, dlatego konieczne jest zastosowanie układu wzmacnienia.

Układ wzmacnienia zrealizowano za pomocą bramek logicznych, zintegrowanych z tranzystorem NPN na wyjściu, dzięki temu układ charakteryzuje się dużą impedancją wejściową oraz niską impedancją wyjściową, co dostarcza potrzebny prąd. Maksymalny prąd wynosi 100 mA, lecz zdecydowano się na minimalny prąd umożliwiający poprawną pracę lasera, zgodnie z jego charakterystyką prądowo-napięciową, aby zminimalizować ryzyko uszkodzenia. Do zasilania lasera użyto rezystora pull-up, którego wartość obliczono według następującego wzoru:

$$R2 = \frac{V_{CC} - V_F}{I_F} \quad (1)$$

gdzie: $R2$ — rezistor pull-up, V_{CC} — napięcie zasilania, $V_F = 1.65$ [V] — napięcie przewodzenia lasera półprzewodnikowego (*ang. Forward Voltage*), zgodnie z dokumentacją, I_F — prąd pracy lasera (maksymalnie 100 [mA]). Parametry wybrano w następujący sposób: $V_{CC} = 5$ [V] (napięcie USB), $I_F = 45$ [mA].

Podstawiając powyższe wartości do równania 1, można przeprowadzić obliczenia:

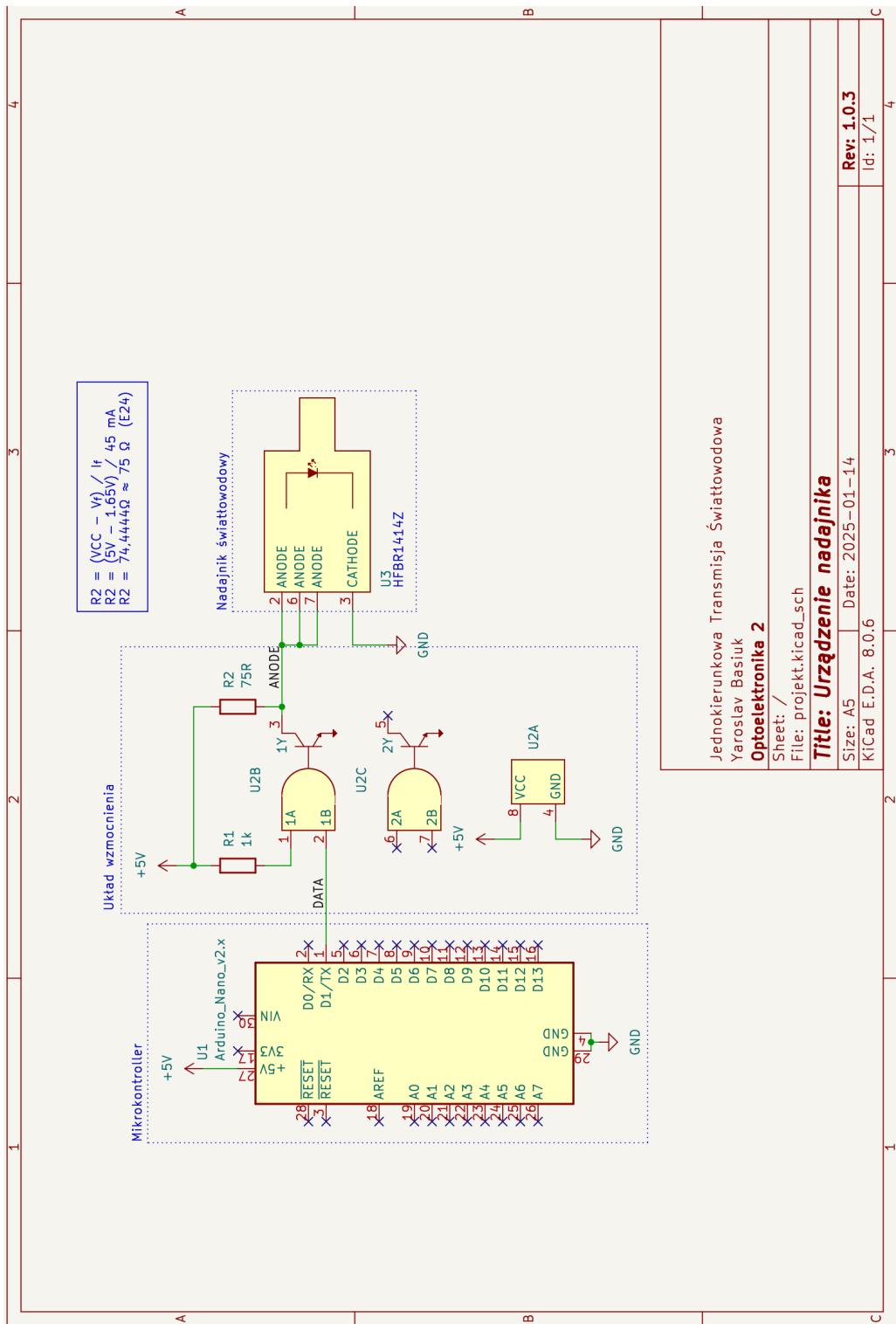
$$R2 = \frac{5[V] - 1.65[V]}{45[mA]} = 74.44444[\Omega] \approx 75[\Omega]. \quad (2)$$

W równaniu 2 wyznaczona wartość $R2$ wynosi 74.44444 [Ω]. W praktyce rezistor o takiej rezystancji nie jest dostępny. Jest możliwość uzyskania takiej wartości za pomocąłączenia kilku rezystorów, natomiast celem jest uproszczenie konstrukcji, dlatego została dobrana najbliższa rezystancja z szeregu E24, która wynosi 75 [Ω].

Na rysunku 2 układ scalony U2 reprezentuje dwie bramki logiczne AND. Wykorzystano tylko jedną bramkę logiczną. Dodając rezistor R1, uzyskano stan wysoki na wejściu 1A bramki U2B. Dzięki temu bramka ta pełni funkcję wtórnika logicznego (*ang. Buffer*). W rezultacie układ wzmacnienia charakteryzuje się dużą impedancją wejściową, co zapobiega przeciążeniu pinu D1 w mikrokontrolerze U1.

Gdy na wejściu bufora (U2B) znajduje się logiczne 0, tranzystor jest zamknięty, a prąd płynie z zasilania przez rezistor R2 do lasera (U3). Oznacza to, że przy stanie logicznym 0 laser świeci. Natomiast gdy na wejściu bufora pojawia się logiczna 1, tranzystor otwiera się i zwiera potencjał węzła, do którego podłączona jest anoda lasera, do 0 [V]. Wówczas prąd płynący przez rezistor R2 omija laser (U3).

Podsumowując, układ wzmacnienia odwraca sygnał — przy logicznej 1 laser U3 nie emituje światła, a przy logicznym 0 laser świeci, przesyłając światło do światłowodu.



Rysunek 2. Schemat połączeń urządzenia nadajnika.

3.2.2 Dobór komponentów

Mikrokontroler

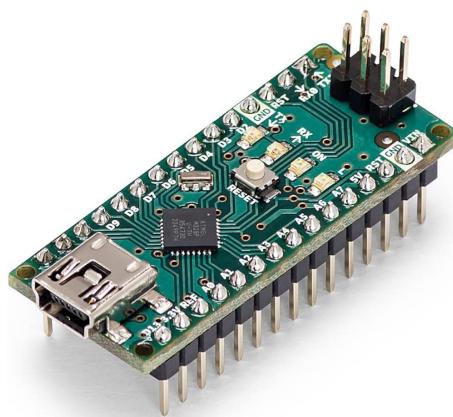
Do nadajnika i odbiornika zostaje wybrane dwie platformy Arduino Nano, które jest

przedstawione na rysunku 3. Są to kompaktowe płytki mikrokontrolera oparte na układzie ATmega328p (wersja 3.x), stanowiąca jednostkę sterującą systemu. Jest kompatybilna z Arduino IDE i zasilana napięciem 5 V przez gniazdo USB Mini-B lub piny zasilania. Mikrokontroler w nadajniku odpowiada za przesyłanie danych światłowodem za pośrednictwem protokołem UART. W odbiorniku odpowiada za dekodowanie odebranych danych oraz wyświetlanie wyników na ekranie LCD.

Parametry Arduino NANO:

- Mikrokontroler: ATmega328p
- Napięcie zasilania: 5 V
- Pamięć FLASH: 32 KB
- Pamięć SRAM: 2 KB
- Taktowanie zegara: 16 MHz
- Porty komunikacyjne: UART

Wybrano ten mikrokontroler z uwagi na jego cenę (jest najtańszy z dostępnych na rynku) oraz łatwość w obsłudze. Ma też wbudowany konwerter USB/TTL, dzięki czemu można przesyłać dane z komputera bez używania pośrednich komponentów.



Rysunek 3. Mikrokontroler Arduino NANO.;

Zasilanie

Podstawą każdego urządzenia jest zasilanie. W projekcie, którego głównym celem było zminimalizowanie liczby komponentów zasilano przez port USB, platformy Arduino. Dodatkowo, dostępny jest pin 5 V umożliwia dostarczanie zasilania do pozostałych komponentów.

Rozwiązanie to uzasadnia dużą dostępnością ładowarek do smartfonów wyposażonych w wyjście USB, co ułatwia zasilanie urządzenia. Układy mogą być również zasilane bezpośrednio z komputera. Standard USB 2.0 przewiduje maksymalny prąd wynoszący 500 mA, podczas gdy nadajnik używany w projekcie pobiera maksymalnie 100 mA.

Do zasilania układu wykorzystano kabel USB-A — miniUSB, przedstawiony na rysunku 4.



TopElektronik.eu

Rysunek 4. Kabel zasilający typu mini-B.

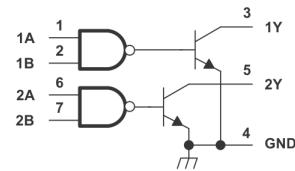
Wzmocnienie

W układzie użyto bramki logicznej AND, oraz bipolarny tranzystor NPN. Ostatecznie został wybrany układ scalony SN75452BP od Texas Instruments. Wybrano ten układ scalony, z uwagi na wbudowany tranzystor NPN, dzięki czemu jest minimalizacja ilości komponentów. Na rysunku 5 jest przedstawiony dany układ. Strukturą wewnętrzną producent przedstawia w dokumentacji na przykładzie bramek NAND, natomiast w wybranym układzie są bramki logiczne AND.

Do transmisji danych wybrano nadajnik HFBR1414Z z powodu jego kompatybilnością ze złączką ST, który jest przedstawiony na rysunku 6. Specyfikacja lasera jest podana w tabeli 1.



(a) Obudowa.



(b) Struktura wewnętrzna na przykładzie bramek logicznych NAND.

Rysunek 5. Układ scalony SN75452BP.

Tabela 1. Podstawowe parametry nadajnika światłowodowego HFBR1414Z, gdzie Min. - wartość minimalna, Typ. - wartość typowa, Max. - Wartość maksymalna.

Parametr	Symbol	Min.	Typ.	Max.	Warunki
Napięcie przewodzenia	V_F	1.48	1.70	2.09	$I_F = 60 \text{ mA dc}$
Napięcie wsteczne	V_{BR}	1.8	3.8	—	$I_F = -100 \mu\text{A dc}$
Długość fali szczytowej	I_P	792	820	865	—



Rysunek 6. Nadajnik światłowodowy HFBR1414Z.

3.2.3 Montaż układu

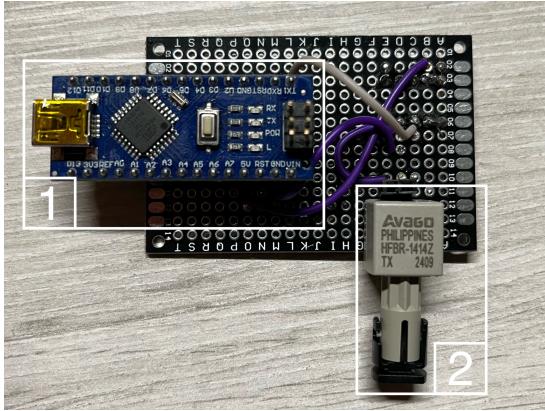
Do montażu obu układów użyto płytka prototypowej.

Na rysunku 7 pokazano zmontowany układ nadajnika.

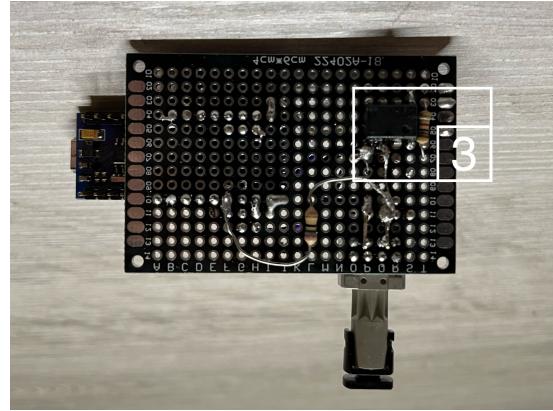
Opis zaznaczonych elementów:

1. Mikrokontroler Arduino NANO.
2. Nadajnik światłowodowy.
3. Układ wzmacnienia.

Podsumowując, układ jest zmontowany tak samo jak się zakładało.



(a) Widok z góry na układ nadajnika.



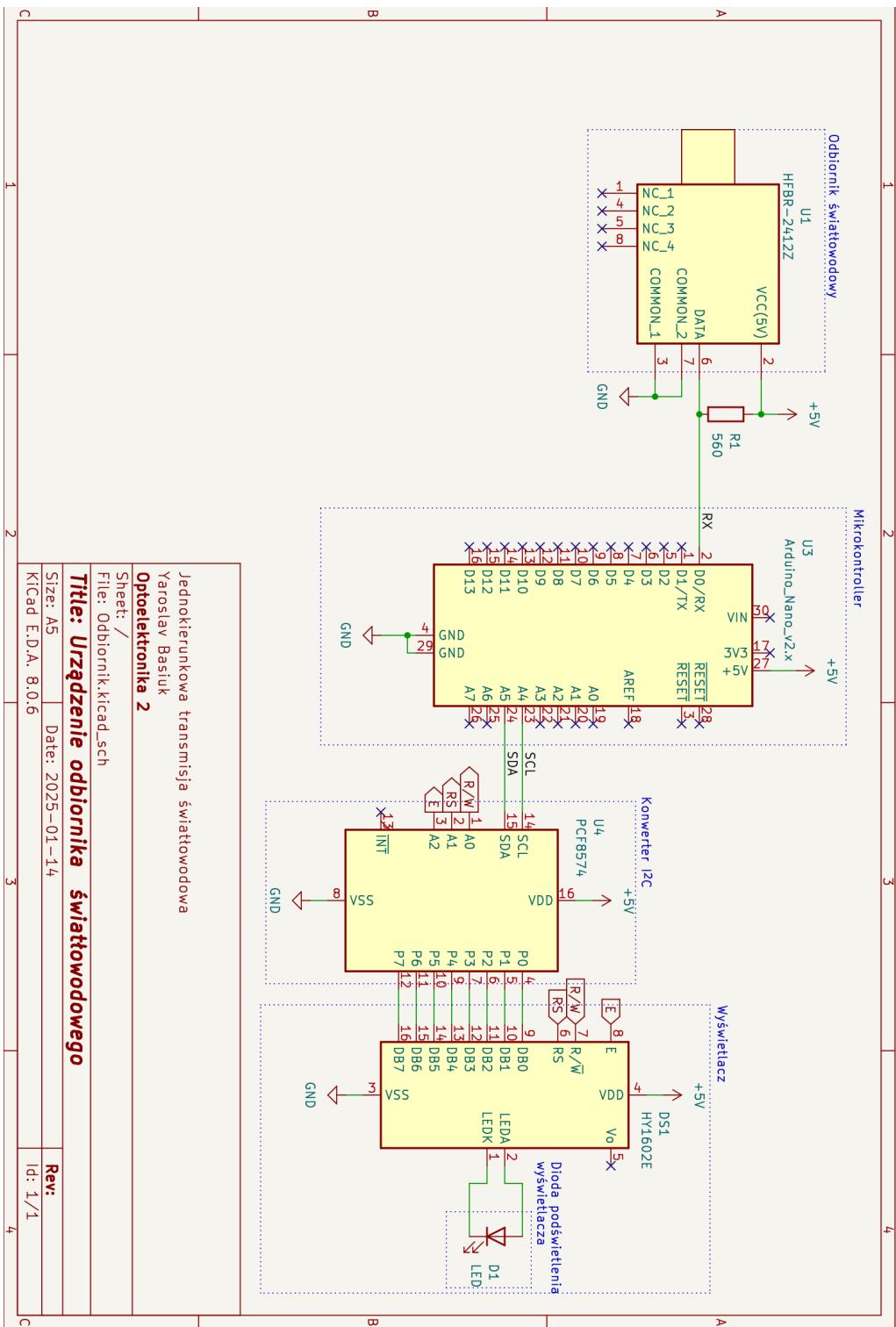
(b) Widok z dołu na układ nadajnika.

Rysunek 7. Widoki układu nadajnika: z góry i z dołu.

3.3 Odbiornik

3.3.1 Schemat ideowy

Na rysunku 8 jest przedstawiony schemat ideowy odbiornika. Dzieli się on na 4 sekcje. Odbiornik światłowodowy reprezentuje fotodiodę podłączaną do wzmacniacza w obudowie kompatybilnej z złączem ST. Kiedy światło trafia na fotodiodę, potencjał na wyjściu DATA, jest równy 0 [V]. Kiedy odbiornik nie jest oświecony, to potencjał na tym wyjściu jest równy napięciu zasilania. Do poprawnego działania odbiornika jest dobrany pull-up rezystor R1. Szczegóły doboru tego rezystora, są opisane w sekcji 5.2 dotyczącej kalibracji. Dzięki rezystorowi R1 na rysunku 8 następuje ponowne odwrócenie sygnału, ponieważ sygnał jest wcześniej odwracany w układzie wzmacnienia nadajnika. W efekcie, gdy mikrokontroler U1 na rysunku 2 podaje stan wysoki, mikrokontroler U2 na rysunku 8 również odbiera stan wysoki (5 V). Wyświetlacz jest podłączony za pośrednictwem konwertera I2C. Domyślnie ten wyświetlacz jest sterowany interfejsem równoległym, 8 bitów do przesyłu danych, 1 bit RS tryb polecenia, 1 bit ReadWrite i 1 Enable. Łącznie wymagane jest 11 bitów, żeby zwolnić wyjścia GPIO Arduino jest użyty konwerter I2C, który zajmuje tylko 2 piny (pin zegara oraz pin danych). Dzięki zastosowaniu konwertera osiągnięty jest uproszczony montaż.



Rysunek 8. Schemat połączeń urządzenia odbiornika.

3.3.2 Dobór komponentów

Mikrokontroler oraz Zasilanie

Mikrokontroler oraz zasilanie są identyczne jak w nadajniku.

Odbiornik światłowodowy

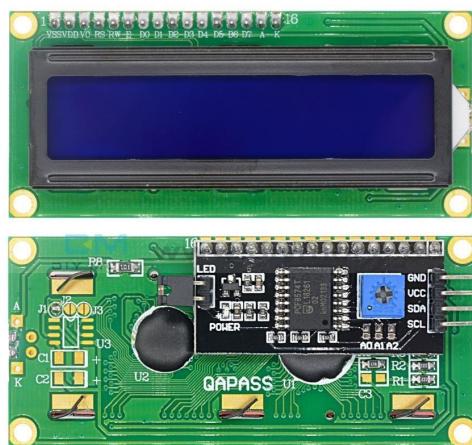
W odbiorniku jest użyty czujnik HFBR2412Z. Wybrano dany czujnik, z uwagi na jego kompatybilność ze złączką ST jak w nadajniku. Ponadto ma wbudowany wzmacniacz, który wzmacnia sygnał z fotodiody, dzięki temu dodając tylko PULLUP rezystor, na wyjściu jest uzyskany sygnał kompatybilny z logiką TTL.



Rysunek 9. Odbiornik światłowodowy HFBR2412Z

Wyświetlacz

Na rysunku 10 przedstawiono wyświetlacz LCD, który obsługuje wyświetlanie 32 znaków w układzie 2×16, wraz z konwerterem I2C. Konwerter umożliwia sterowanie ekranem za pomocą dwóch linii komunikacyjnych: SDA i SCL. Układ zasilany jest napięciem 5 V. Na płytce wyświetlacza znajduje się potencjometr, który pozwala na regulację kontrastu.



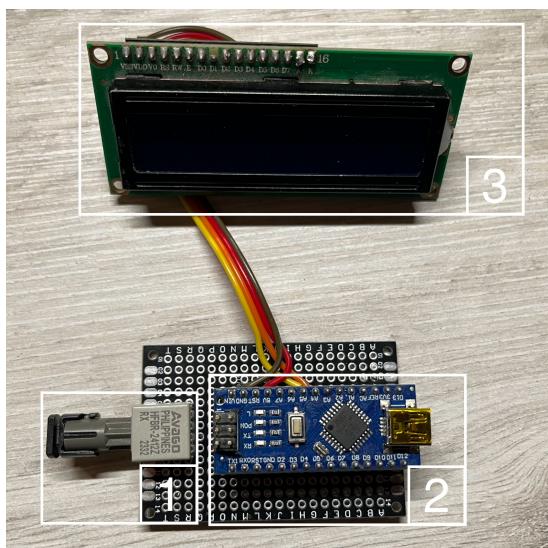
Rysunek 10. Układ wyświetlacza LCD z konwerterem I2C.

3.3.3 Montaż układu

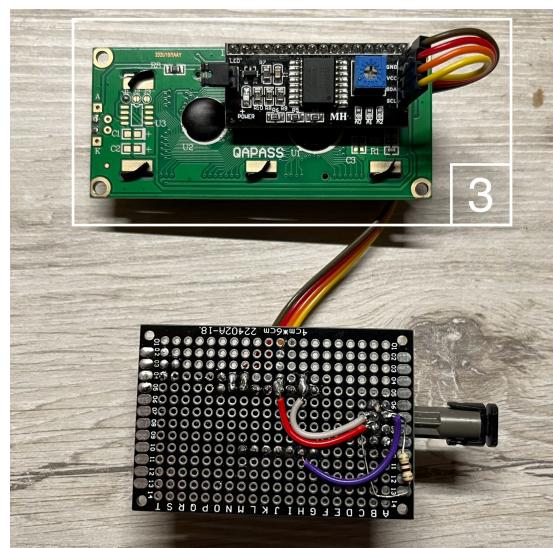
Na rysunku 11 przedstawiono górny i dolny widok zmontowanego odbiornika.

Dokładny opis zaznaczonych elementów rysunku ??:

1. Odbiornik światłowodowy.
2. Arduino NANO.
3. Wyświetlać 1602 z wbudowanym konwerterem I2C.



(a) Widok z góry na układ odbiornika.



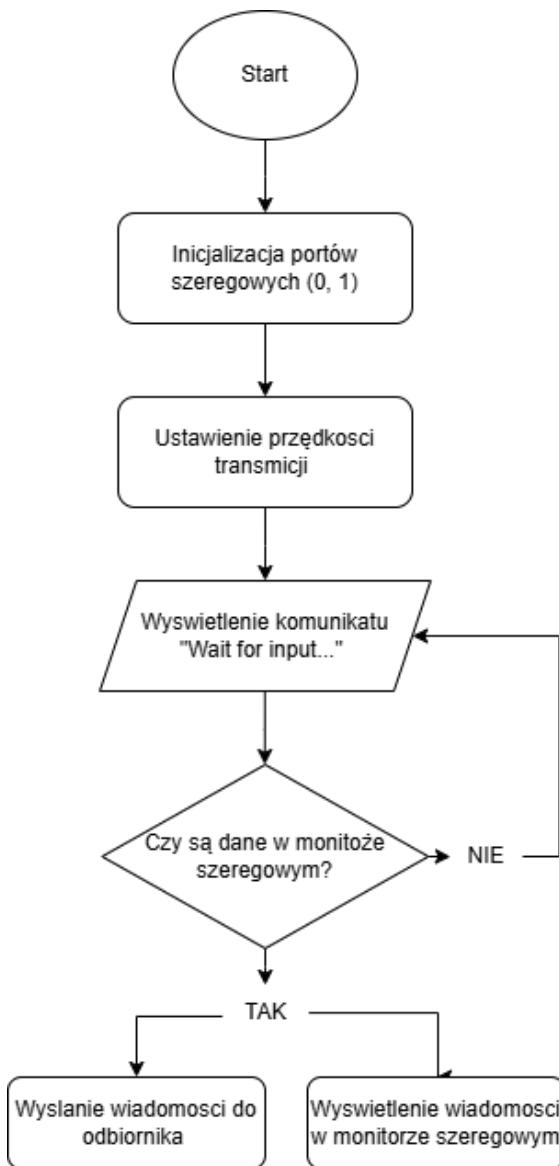
(b) Widok z dołu na układ odbiornika.

Rysunek 11. Widoki układu nadajnika: z góry i z dołu.

4 OPIS CZEŚCI PROGRAMOWEJ

4.1 Algorytm główny

4.1.1 Nadajnik



Rysunek 12. Diagram programu nadajnika

Diagram przedstawia algorytm działania nadajnika.

1. Start

Algorytm rozpoczyna się od symbolu "Start", który reprezentuje początek działania programu.

2. Inicjalizacja portów szeregowych (0, 1)

Na początku wykonywana jest konfiguracja portów szeregowych:

- Wbudowany port szeregowy `Serial` jest inicjalizowany, co umożliwia komunikację z komputerem przez monitor szeregowy.
- Wirtualny port szeregowy `mySerial`, przypisany do pinów RX (0) i TX (1), również zostaje skonfigurowany.

3. Ustawienie prędkości transmisji

Prędkość transmisji dla obu portów szeregowych została ustalona na 9600 bodów.

4. Wyświetlenie komunikatu "Waiting for input..."

Po inicjalizacji portów szeregowych w monitorze szeregowym wyświetlany jest komunikat „Waiting for input...”, który informuje użytkownika o gotowości nadajnika do odbierania danych.

5. Czy są dane w monitorze szeregowym?

Algorytm sprawdza, czy w monitorze szeregowym dostępne są dane do odczytu:

- Jeśli dane są dostępne, przechodzi do kolejnego kroku.
- Jeśli dane nie są dostępne, algorytm wraca do oczekiwania.

6. Odczytanie wiadomości

Odczytywane są dane przesłane przez monitor szeregowy do momentu napotkania znaku nowej linii. Dane te są przechowywane w zmiennej tekstowej `message`.

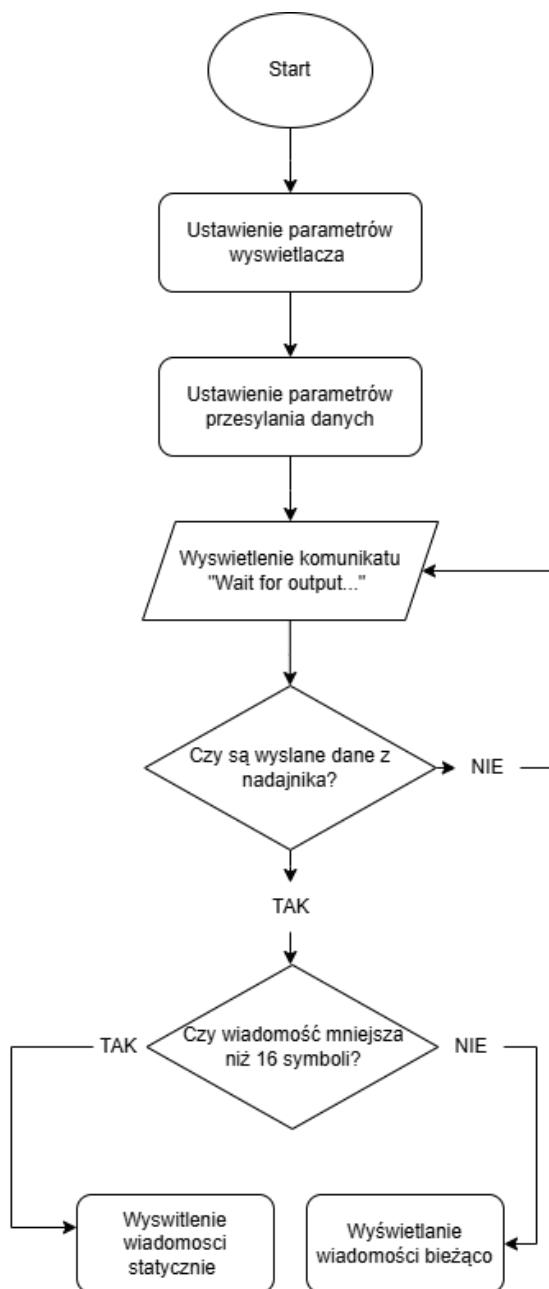
7. Wyświetlenie wiadomości w monitorze szeregowym

Odczytana wiadomość jest wyświetlana w monitorze szeregowym, co pozwala użytkownikowi zweryfikować przeslaną wiadomość.

8. Pętla programu

Po przetworzeniu wiadomości algorytm wraca do kroku 5, gdzie ponownie sprawdza dostępność danych w monitorze szeregowym. Dzięki temu nadajnik działa w sposób ciągły, przetwarzając dane w czasie rzeczywistym.

4.1.2 Odbiornik



Rysunek 13. Diagram programu odbiornika

Diagram przedstawia algorytm działania odbiornika.

1. Start

Algorytm rozpoczyna swoje działanie.

2. Inicjalizacja wyświetlacza

Wyświetlacz LCD jest przygotowywany do pracy: włączane jest podświetlenie, ustawiany kursor i czyszczony ekran.

3. Konfiguracja portów szeregowych

Porty szeregowe są inicjalizowane, umożliwiając komunikację z monitorem szeregowym oraz odbieranie danych z nadajnika.

4. Wyświetlenie komunikatu „Waiting for msg”

Na ekranie LCD i w monitorze szeregowym pojawia się komunikat informujący o gotowości odbiornika.

5. Sprawdzanie danych z nadajnika

Algorytm sprawdza, czy z nadajnika przesłano dane:

- Jeśli dane są dostępne, są odczytywane.
- Jeśli brak danych, algorytm wraca do oczekiwania.

6. Analiza długości wiadomości

Po odczytaniu wiadomości sprawdzana jest jej długość:

- Jeśli długość nie przekracza 16 znaków, wiadomość jest wyświetlana statycznie.
- Jeśli wiadomość jest dłuższa niż 16 znaków, wyświetlana jest dynamicznie, przewijając się w drugim wierszu ekranu LCD.

7. Wyświetlanie wiadomości

Odebrana wiadomość jest prezentowana na ekranie LCD: statycznie (dla krótkich wiadomości) lub dynamicznie (dla długich wiadomości).

8. Pętla programu

Po przetworzeniu wiadomości algorytm wraca do sprawdzania dostępności danych, działając w sposób ciągły i przetwarzając kolejne wiadomości w czasie rzeczywistym.

4.2 Opis kluczowych funkcji

4.2.1 Nadajnik

Deklaracja bibliotek i obiektów

Biblioteka `SoftwareSerial` umożliwia emulowanie portu szeregowego na dowolnych cyfrowych pinach mikrokontrolera. Komunikuje się za pomocą protokołu UART (*Universal Asynchronous Receiver-Transmitter*), który jest standardem dla dwukierunkowej transmisji szeregowej danych.

```
9 #include <SoftwareSerial.h>
```

Deklaracja portów UART

```
12     SoftwareSerial mySerial(0, 1); // RX (D0), TX (D1)
```

- 0 → Pin RX (odbiór danych).
- 1 → Pin TX (wysyłanie danych).

Te piny (D0 i D1) są używane przez wbudowany sprzętowy UART (`Serial`), który jest zarezerwowany dla komunikacji z komputerem z *Serial Monitor* w Arduino IDE.

Funkcja `setup()`

Funkcja `setup()` jest wywoływana tylko raz po uruchomieniu lub resetie mikrokontrolera. Ustawia początkowe warunki pracy mikrokontrolera, takie jak konfiguracja portów komunikacyjnych i wyświetlanie komunikatów w *Serial Monitor*. Kluczowe działania w tej funkcji:

- Inicjalizuje sprzętowy UART, który umożliwia komunikację z komputerem przez port szeregowy (np. wyświetlanie danych w *Serial Monitor*). Prędkość transmisji (*baud rate*) jest ustawiona na 9600 bodów.

```
15     Serial.begin(9600);
```

- Inicjalizuje wirtualny port szeregowy stworzony za pomocą biblioteki `SoftwareSerial`. Również działa z prędkością transmisji 9600 bodów.

```
16     mySerial.begin(9600);
```

- Wyświetla komunikat w *Serial Monitor*, informując użytkownika, że nadajnik jest gotowy do odbierania danych.

```
17      Serial.println("Waiting_for_input...");
```

Funkcja **loop()**

Funkcja **loop()** jest wywoływana wielokrotnie w pętli. Odpowiada za główną logikę programu i wykonywanie działań w czasie rzeczywistym. Kluczowe działania w tej funkcji:

- Sprawdza, czy są dostępne dane do odczytu w buforze UART. Funkcja zwraca liczbę bajtów oczekujących na odczyt.

```
22      if (Serial.available() > 0) {
```

- Odczytuje dane przesłane przez *Serial Monitor* do momentu napotkania znaku nowej linii (*\n*). Dane są przechowywane w zmiennej typu **String** i interpretowane jako wiadomość tekstowa.

```
24      String message = Serial.readStringUntil('\'\n'');
```

- Wyświetla odczytaną wiadomość w *Serial Monitor*, aby użytkownik mógł zobaczyć, co zostało odebrane.

```
27      Serial.println(message);
```

4.2.2 Odbiornik

Deklaracja bibliotek i obiektów

```
9 #include <LiquidCrystal_I2C.h>
```

- Używana do obsługi wyświetlacza LCD podłączonego przez interfejs I2C. - Umożliwia łatwą kontrolę wyświetlania tekstu na LCD.

```
10 #include <SoftwareSerial.h>
```

- Umożliwia utworzenie wirtualnego portu szeregowego na dowolnych cyfrowych pinach.
- W tym kodzie port `mySerial` jest przypisany do pinów:

- 0 → RX (odbiór danych).
- 1 → TX (wysyłanie danych).

```
13 LiquidCrystal_I2C lcd(0x27, 16, 2);
```

- Inicjalizuje obiekt `lcd` dla wyświetlacza LCD z adresem I2C 0x27, który ma 16 kolumn i 2 wiersze.

```
16 SoftwareSerial mySerial(0, 1); // RX (D0), TX (D1)
```

- Tworzy wirtualny port szeregowy przypisany do pinów 0 (RX) i 1 (TX).

Funkcja `setup()`

Funkcja `setup()` jest uruchamiana raz po włączeniu mikrokontrolera lub jego resetie. Służy do inicjalizacji i konfiguracji ustawień początkowych.

Kluczowe działania:

- Inicjalizuje wbudowany port UART do komunikacji z komputerem przez port USB.

```
19     Serial.begin(9600);
```

- Inicjalizuje wirtualny port szeregowy stworzony przez `SoftwareSerial`. Prędkość transmisji wynosi 9600 bodów.

```
20     mySerial.begin(9600);
```

- Inicjalizuje wyświetlacz LCD i przygotowuje go do pracy.

```
22     lcd.init();
```

- Włącza podświetlenie LCD, dzięki czemu wyświetlany tekst jest widoczny.

```
23     lcd.backlight();
```

- Czyści wyświetlacz, usuwając poprzednio wyświetlane dane.

```
24     lcd.clear();
```

- Ustawia kurSOR na pierwszym wierszu i wyświetla komunikat „Waiting for msg”.

```
25     lcd.setCursor(0, 0);
```

```
26     lcd.print("Waiting_for_msg");
```

- Wyświetla komunikat w Serial Monitor, informując użytkownika, że odbiornik jest gotowy do pracy.

```
27     Serial.println("Waiting_for_output...");
```

Funkcja loop()

Funkcja `loop()` jest wykonywana w pętli i odpowiada za odbieranie oraz wyświetlanie danych w czasie rzeczywistym.

Kluczowe działania:

- Sprawdzanie dostępności danych:

```
32     if (Serial.available() > 0)
```

- Odczytanie danych:

```
34     String data = Serial.readStringUntil('\n');
```

- Przetwarzanie wiadomości:

```
35     String message = data.substring(0, data.length() -  
        1);
```

- Wyświetlenie statycznego komunikatu na LCD:

```
37     lcd.clear();
38     lcd.setCursor(0, 0);
39     lcd.print("Message:");
```

- Wysłanie wiadomości przez SoftwareSerial:

```
40     mySerial.println(message);
```

- Wyświetlenie wiadomości w Serial Monitor:

```
41     Serial.println("Message_received:" + message);
```

- Obsługa krótkich wiadomości (<= 16 znaków):

```
43     if (message.length() <= 16) {
44         lcd.setCursor(0, 1);
45         lcd.print(message);
46     }
```

- Obsługa długich wiadomości (> 16 znaków):

```
49     String displayMessage = message + " ";
50     for (int i = 0; i < displayMessage.length(); i++) {
51         lcd.setCursor(0, 1);
52         lcd.print(displayMessage.substring(i, i + 16));
53         delay(263);
54     }
55     lcd.setCursor(0, 1);
56     lcd.print(" ");
```

5 Uruchomienie, pomiary testowe i kalibracja

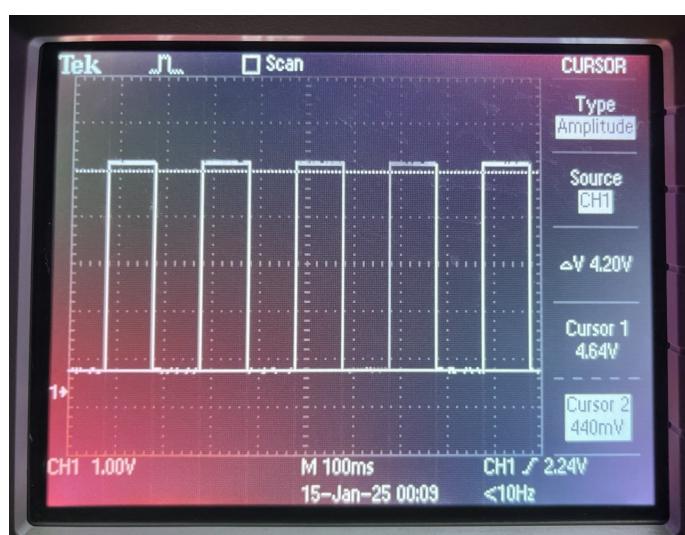
5.1 Uruchomienie

Pierwsze uruchomienie nadajnika skończyło się niepowodzeniem. Przez przypadek, laser światłowodowy podłączono bez rezystora R2, z rysunku 2, co spowodowało spalenie lasera.

Po uruchomieniu odbiornika, dane nie były odbierane z nadajnika. Postanowiono wyjście odbiornika światłowodowego podłączyć do oscyloskopu, i zaświecić mocnym źródłem światła. Kiedy światłem świecono na odbiornik, było widać spadek napięcia na wyjściu, wnioskując tym, że odbiornik "reaguje na światło".

5.2 Pomiary testowe i kalibracja

Procesowi kalibracji podlegał tylko odbiornik. Nadajnik był wzbudzony sygnałem prostokątnym, co umożliwiało sprawdzenie poziomu napięć w różnych stanach na odbiorniku. Na wyjściu dobrano wartość rezystora pull-up, który służy do dobierania poziomów napięć do konkretnej logiki. Na początku jego wartość wynosiła $1\text{ k}\Omega$. W tym przypadku przy stanie wysokim napięcie wynosiło 3,2 V, a przy niskim 2,8 V. Niestety, te poziomy nie były zgodne z TTL, inaczej mówiąc, mikrokontroler nie był w stanie odczytywać danych. Po dobraniu odpowiedniego rezystora najlepsza wartość wynosiła $550\text{ }\Omega$. Jak pokazano na rysunku 14, przy stanie wysokim wartość napięcia wynosi 4,7 V, a przy niskim 0,4 V.



Rysunek 14. Pomiar poziomów napięć na odbiorniku za pomocą kurSORów na oscyloskopie

5.3 Specyfikacja urządzenia

W ramach projektu zaprojektowano i wykonano dwa podstawowe moduły: nadajnik oraz odbiornik, których specyfikacja została przedstawiona w tabelach 2 i 3.

5.3.1 Nadajnik

Tabela 2 przedstawia kluczowe parametry nadajnika. Moduł ten zasilany jest napięciem 5V, a maksymalny pobór prądu wynosi 100 mA. Główne zadanie nadajnika to emisja światła lasera o długości fali 820 nm, co umożliwia transmisję danych z częstotliwością bitową 96 kb/s.

Tabela 2. Specyfikacja nadajnika

Parametr	Symbol	Wartość
Napięcie zasilania	V_{IN}	5V
Maksymalny prąd poboru	I_{MAX}	100mA
Długość fali lasera	λ	820nm
Częstotliwość bitowa	f_b	96 kb/s

5.3.2 Odbiornik

Specyfikacja odbiornika została przedstawiona w tabeli 3. Moduł odbiorczy również zasilany jest napięciem 5V, jednak jego maksymalny pobór prądu jest mniejszy i wynosi 50 mA. Podobnie jak nadajnik, odbiornik przetwarza dane z częstotliwością bitową 96 kb/s, zapewniając płynną komunikację w układzie transmisji światłowodowej.

Tabela 3. Specyfikacja odbiornika

Parametr	Symbol	Wartość
Napięcie zasilania	V_{IN}	5V
Maksymalny prąd poboru	I_{MAX}	50mA
Częstotliwość bitowa	f_b	96 kb/s

6 INSTRUKCJA OBSŁUGI

6.1 Nadajnik

6.1.1 Opis złącz

Na rysunku 1 jest pokazany nadajnik światłowodowy. Który w sobie ma 2 złącza:

1. Jest to złącze zasilania oraz do odbioru danych z komputera.
2. Jest to złącze światłowodowe ze złączką ST.

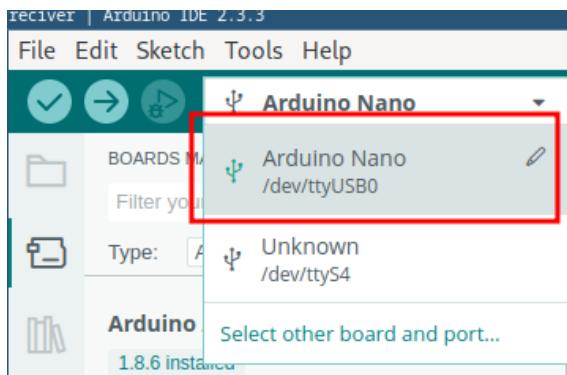


Rysunek 1. Nadajnik światłowodowy.

6.1.2 Uruchomnienie i konfiguracja

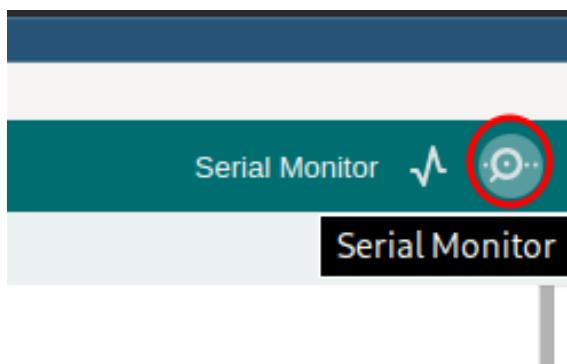
Dla uruchomienia wystarczy nadajnik podłączyć do komputera, do USB. Podłączyć światłowód do złącza nr 2 na rysunku 1.

W górnym rogu lewym rogu, jest pole do wyboru urządzenia. Należy kliknąć na niego, i wybrać pole "Arduino NANO". Jak pokazano na rysunku 2.



Rysunek 2. Wybór nadajnika do komunikacji.

Następnie trzeba otworzyć SERIAL MONITOR. Który jest w górnym lewym rogu, jak pokazano na rysunku 3.



Rysunek 3. Otwieranie Serial Monitora

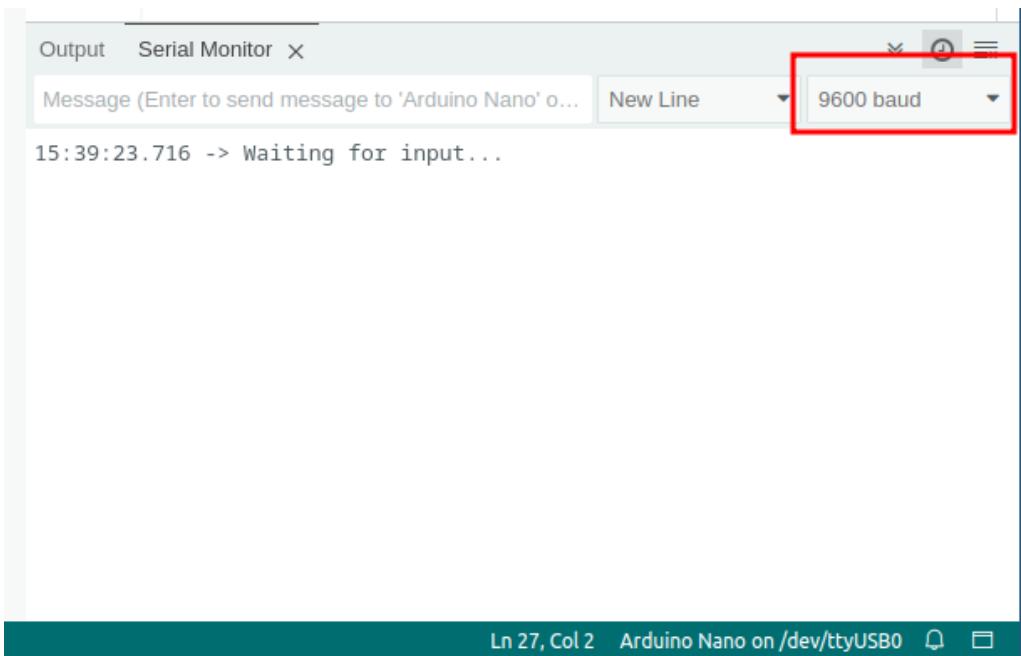
Po otwarciu Monitora szeregowego, należy wybrać częstotliwość 9600 baud (9600 bit/s), jak na rysunku 4. Także należy wybrać separator "NEW LINE" (jak na rysunku ??), znaczy to, że na końcu wiadomości jest znak nowej linii ('\n'). Jest to ważne, ponieważ odbiornik rozpoznaje koniec wiadomości tym znakiem.

6.2 Odbiornik

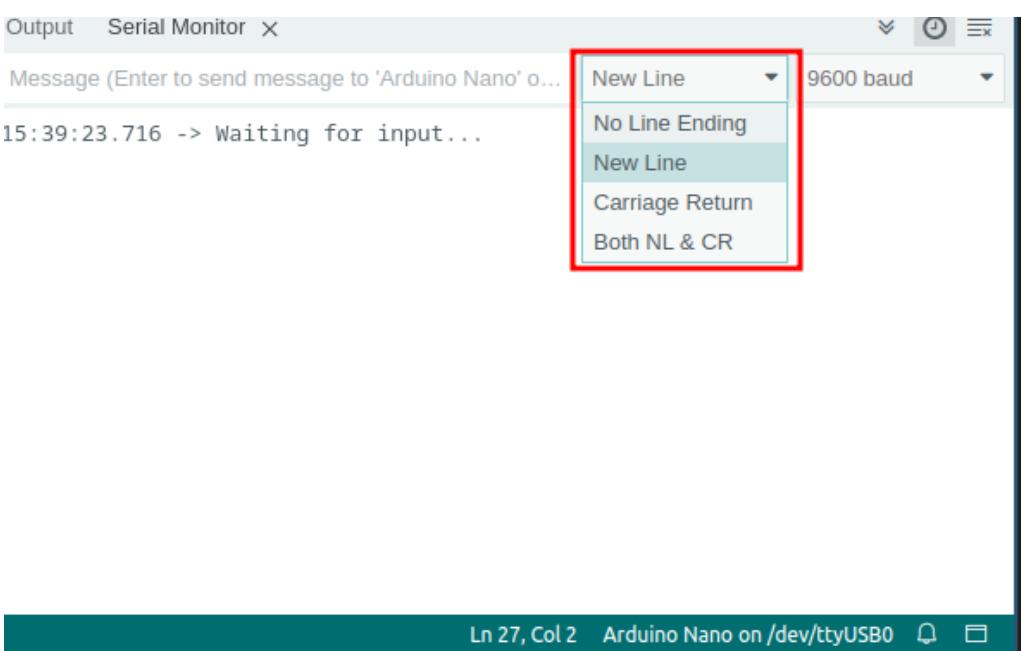
6.2.1 Opis złącz

Na rysunku 6 jest pokazany odbiornik światłowodowy. Na nim ponumerowane są dane elementy:

1. Złącze zasilanie wraz ze złączem komunikacji z komputerem (USB).
2. Złącze światłowodowe ze złączką ST.
3. Wyświetlacz.



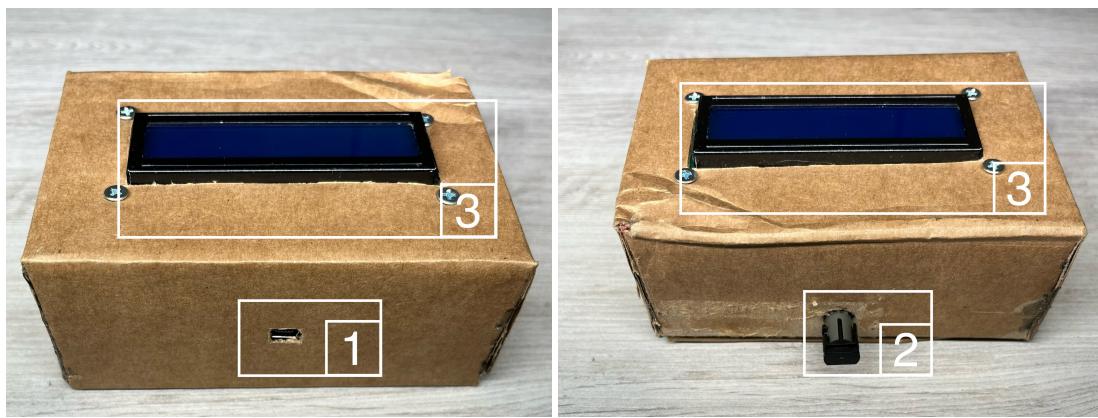
Rysunek 4. Wybór częstotliwości monitora szeregowego.



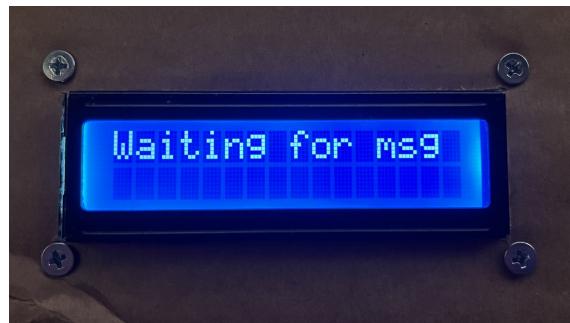
Rysunek 5. Wybór separatora dla przesyłania wyadomości

6.2.2 Uruchomienie

Do uruchomienia odbiornika, wystarczy podłączyć drugi koniec światłowodu do złączki nr 2 z rysunku 6, oraz podłączyć do zasilacza lub innego komputera za pomocą kabla mini usb złącze nr 1, z rysunku 6. W przypadku, jeśli odbiornik został uruchomiony poprawnie, na wyświetlaczu pojawi się komunikat: "Waiting for a msg.", jak na rysunku 7.



Rysunek 6. Odbiornik światłowodowy.



Rysunek 7. Oczekiwanie odbiornika na wiadomość.

6.3 Przesyłanie danych.

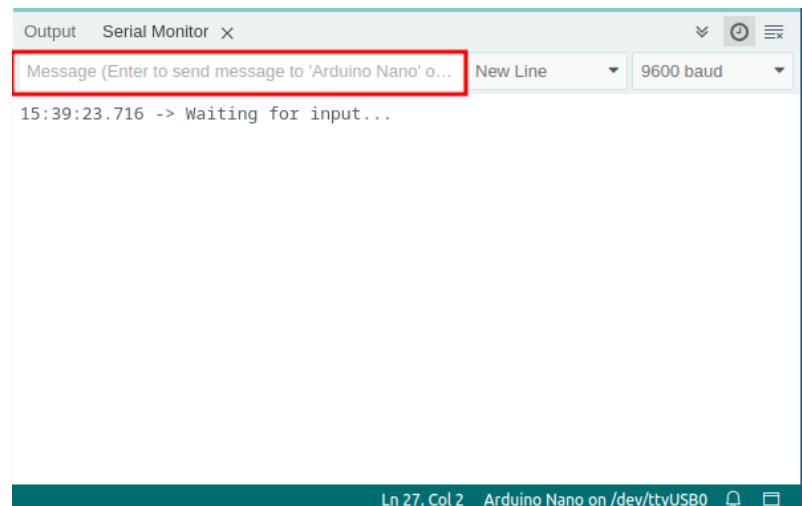
Do przesyłania danych do nadajnika, należy użyć odpowiedniego oprogramowania, do akwizycji, (np. PuTTY), natomiast zalecane jest Arduino IDE. Jeśli konfiguracja nadajnika jest poprawna, to nadajnik wyśle wiadomość do komputera dla użytkownika: "Waiting for input...". Jak na rysunku 8.



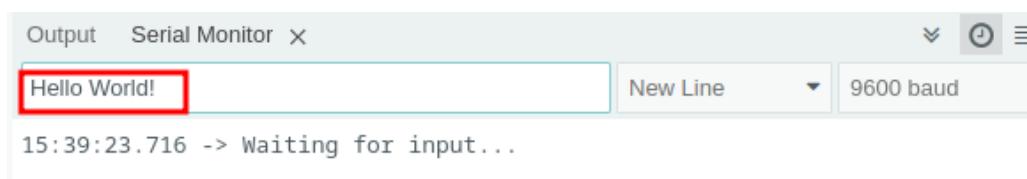
Rysunek 8. Oczekiwanie odbiornika na wiadomość od użytkownika

W polu tekstowym "Message", można wpisywać wiadomość do przesyłu po torze światłowodowym. Pole tekstowe jest pokazane na rysunku 9. Dla przykładu można przesłać "Hello World!", jak na rysunku 10.

Dana wiadomość powinna się wyświetlić na wyświetlaczu, jak przedstawiono na rysunku 11.



Rysunek 9. Pole tekstowe dla przesyłania wiadomości.



Rysunek 10. Przykład przesyłanie wiadomości.



Rysunek 11. Wyświetlenie odebranej wiadomości

7 PODSUMOWANIE

7.1 Przedmiot pracy

Założeniem projektowym było zaprojektowanie najprostszego komunikatora światłowodowego, przesyłanie danych tekstowych za pomocą światłowodu.

7.2 Osiągnięte wyniki

- Przesyłanie danych tekstowych za pomocą światłowodu.
- Zaprojektowany układ sterowania laserem światłowodowy.

7.3 Napotkane problemy

Jednym z problemów była dostępność nadajnika i odbiornika światłowodowego. Czekane są z zagranicy, i czas oczekiwania na dostawę wynosił 3 tygodnie.

Kolejny problem był, źle podłączenie nadajnika światłowodowego bez użycia rezystora, co skutkowało jego uszkodzeniem. Na dostawę kolejnego lasera oczekiwanie wynosiło kolejne 3 tygodnie.

7.4 Wizje na przyszłość

- Rezygnacja z platformy Artudio na korzyść STM32F4xx.
- Minimalizacja rozmiaru.
- Zmiana elementów na typ SMD.
- Dwukierunkowa transmisja.
- Integracja projektu w dronach.

8 BIBLIOGRAFIA

Literatura

- [1] Dokumentacja odbiornika i nadajnika: https://www.mouser.pl/datasheet/2/678/AV02_0176EN_2023_01_26-1827546.pdf
- [2] Dokumentacja układu wzmacnienia: https://www.ti.com/lit/ds/symlink/sn75452b.pdf?ts=1737462938236&ref_url=https%253A%252F%252Fwww.ti.com%252Fproduct%252FSN75452B%253Futm_source%253Dsupplyframe%2526utm_medium%253DSEP%2526utm_campaign%253Dnot_alldatasheet%2526DCM%253Dyes
- [3] Dokumentacja Arduino Nano: <https://docs.arduino.cc/resources/datasheets/A000005-datasheet.pdf>
- [4] Definicja światła: <https://pl.wikipedia.org/wiki/%C5%9Awiat%C5%82o>
- [5] Definicja zjawiska fotoelektrycznego: https://pl.wikipedia.org/wiki/Efekt_fotoelektryczny
- [6] Definicja lasera półprzewodnikowego: https://pl.wikipedia.org/wiki/Laser_p%C3%B3%C5%82przewodnikowy

9 DODATKI

9.1 Nadajnik

Listing 1. "Kod programu nadajnika"

```
1 /*
2 Autor: Yevhen Shpatakov
3 Wersja kodu: 2.3.4
4 Data napisania: 28.01.2025
5 Język: Arduino (C)
6 System operacyjny: Windows 11
7 Opis: Program nadajnika, który odbiera dane z monitora
szeregowego i wysyła je przez SoftwareSerial.
8 */
9 #include <SoftwareSerial.h>
10
11 // Definiowanie pinów RX i TX dla SoftwareSerial
12 SoftwareSerial mySerial(0, 1); // RX (D0), TX (D1)
13
14 void setup() {
15     Serial.begin(9600); // Inicjalizacja komunikacji przez
    monitor szeregowy (predkosc 9600 bodow)
16     mySerial.begin(9600); // Inicjalizacja SoftwareSerial z
    predkoscia 9600 bodow
17     Serial.println("Waiting_for_input..."); // Wyswietlanie
    komunikatu w monitorze szeregowym
18 }
19
20 void loop() {
21     // Sprawdzanie, czy dane sa dostepne w monitorze
    szeregowym
22     if (Serial.available() > 0) {
23         // Odczytanie wiadomosci z monitora szeregowego do
```

```

        momentu napotkania znaku nowej linii ('\n')

24     String message = Serial.readStringUntil('\n');

25

26     // Wyswietlenie odczytanej wiadomosci w monitorze
27     // szeregowym
28     Serial.println(message);
29 }
```

9.2 Odbiornik

Listing 2. "Kod programu odbiornika"

```

1 /*
2 Autor: Yevhen Shpatakov
3 Wersja kodu: 2.3.4
4 Data napisania: 28.01.2025
5 Język: Arduino (C)
6 System operacyjny: Windows 11
7 Opis: Program odbiornika, który odbiera dane przesłane
       przez nadajnik i wyświetla je na ekranie LCD.
8 */
9 #include <LiquidCrystal_I2C.h>
10 #include <SoftwareSerial.h>
11
12 // Inicjalizacja wyświetlacza LCD (adres I2C 0x27, 16
13 // kolumn, 2 wiersze)
14 LiquidCrystal_I2C lcd(0x27, 16, 2);
15
16 // Definiowanie pinów RX i TX dla SoftwareSerial
17 SoftwareSerial mySerial(0, 1); // RX (D0), TX (D1)
18
19 void setup() {
20     Serial.begin(9600);           // Inicjalizacja komunikacji
```

```

    przez monitor szeregowy (baud rate: 9600)

20 mySerial.begin(9600);           // Inicjalizacja
    SoftwareSerial z tym samym baud rate (9600)

21
22 lcd.init();                  // Inicjalizacja wyświetlacza
    LCD

23 lcd.backlight();             // Włączenie podświetlenia
    wyświetlacza

24 lcd.clear();                 // Wyczyszczenie ekranu LCD

25 lcd.setCursor(0, 0);         // Ustawienie kurSORA na
    pierwszym wierszu

26 lcd.print("Waiting_for_msg"); // WYSWIETLENIE
    poczatkowego komunikatu na LCD

27 Serial.println("Waiting_for_output..."); // WYSWIETLENIE
    komunikatu w monitorze szeregowym

28 }
29

30 void loop() {
31     // Sprawdzenie, czy dane są dostępne w monitorze
        szeregowym

32     if (Serial.available() > 0) {
33         // Odczytywanie danych z monitora szeregowego do
            momentu napotkania znaku nowej linii ('\n')

34         String data = Serial.readStringUntil('\n');
35         String message = data.substring(0, data.length() - 1);
            // Usunięcie ewentualnego ostatniego znaku nowej
            linii

36
37         lcd.clear();                   // Wyczyszczenie ekranu
            LCD

38         lcd.setCursor(0, 0);          // Ustawienie kurSORA na
            pierwszym wierszu

39         lcd.print("Message:");       // WYSWIETLENIE statycznej

```

```

    etykiety na LCD

40   mySerial.println(message);      // Wyslanie wiadomosci
        przez SoftwareSerial

41   Serial.println("Message_received:_ " + message); //
        Wyswietlenie odebranej wiadomosci w monitorze
        szeregowym

42

43   if (message.length() <= 16) {
44       // Jesli wiadomosc mieści się w 16 znakach, wyswietl
        ja bez przewijania w drugim wierszu
45   lcd.setCursor(0, 1);
46   lcd.print(message);
47 } else {
48     // Jesli wiadomosc jest dłuższa niż 16 znakow,
        przewin ja w drugim wierszu
49   String displayMessage = message + "_"; // Dodanie
        spacji dla płynnego przewijania
50   for (int i = 0; i < displayMessage.length(); i++) {
51     lcd.setCursor(0, 1); // Ustawienie kurSORA na
        drugim wierszu
52     lcd.print(displayMessage.substring(i, i + 16)); //
        Wyswietlenie okna 16 znakow
53     delay(263); // Dostosowanie predkosci przewijania
54   }
55   lcd.setCursor(0, 1);
56   lcd.print("_"); // Wyczyszczenie drugiego wiersza po
        przewijaniu
57 }
58 }
59 }
```