Politechnika Wrocławska

Katedra Metrologii Elektronicznej i Fotonicznej

Laboratorium Optoelektroniki i Fotoniki

Nazwa kursu:

Metrologia optyczna 2 - laboratorium

Temat projektu:

Optyczny Pomiar Temperatury - Pirometr

Autorzy projektu:

Grupa nr 4

Piotr Rosiński, 262016
Patryk Niczke, 260832
Przemysław Lis, 276922

Wydział Elektroniki, Fotoniki i Mikrosystemów
Kierunek: Elektronika

Miejsce i rok: Wrocław, 2024

Spis treści

1. Wstęp

2. Wprowadzenie

Metrologia optyczna stanowi obecnie jeden z najważniejszych narzędzi pomiarowych w nauce i przemyśle stale zwiększając swoje znaczenie. Bezdotykowy pomiar temperatury rewolucjonizuje precyzję kontroli procesów technologicznych, badań naukowych i diagnostyki medycznej. Szczególną zaletą tych rozwiązań jest możliwość wykonywania pomiarów w warunkach, które dotychczas stanowiły wyzwanie – w przypadku obiektów szybko się poruszających, materiałów o ekstremalnych temperaturach lub gdy klasyczny kontakt pomiarowy mógłby zakłócić naturalne właściwości badanego obiektu i wprowadzić zaburzenie do pomiaru. Celem niniejszego projektu jest opracowanie i implementacja pirometru – zaawansowanego urządzenia do bezdotykowego pomiaru temperatury wykorzystującego technologię podczerwieni. Projekt został zrealizowany w oparciu o czujnik MLX90614, który zapewnia odpowiednią precyzję i stabilność pomiarów w założonym zakresie temperatur. Sercem systemu jest popularna płytka mikrokontrolerowa, Arduino UNO, która stanowi centrum sterujące całego urządzenia. Płytka Arduino UNO oparta jest na 8-bitowym mikrokontrolerze ATmega328P, który zapewnia różnorodne funkcje, takie jak 14 cyfrowych pinów wejścia/wyjścia czy 6 analogowych wejść. Dzięki swojej prostocie i wszechstronności, Arduino UNO jest często pierwszym wyborem dla wielu, nieco mniej wymagających obliczeniowo projektów [1]. Kod źródłowy projektu został napisany w języku C/C++, z wykorzystaniem open-sourcowych bibliotek ułatwiających programowanie kluczowych komponentów, w tym wyświetlacza LCD opartego na standardzie HD44780. HD44780 to standardowy kontroler wyświetlaczy LCD. Został opracowany przez firmę Hitachi w latach 80. XX wieku i jest powszech-

nie stosowany w alfanumerycznych wyświetlaczach dot-matrix [2]. W projektach wykorzystujących tę technologię często stosowana jest biblioteka LiquidCrystalI2C, która upraszcza interakcję z wyświetlaczami LCD podłączonymi do mikrokontrolerów takich jak Arduino poprzez interfejs I2C. Dzięki tej bibliotece możliwe jest łatwe sterowanie wyświetlaczem oraz wyświetlanie tekstu i danych w sposób efektywny i intuicyjny.

Inicjalizacja omawianego wyświetlacza LCD z wykorzystaniem biblioteki LiquidCrystalI2C zajmuje zaledwie kilka linijek kodu źródłowego:

```
1 #include <LiquidCrystal_I2C.h>
2 LiquidCrystal_I2C lcd(0x27, 16, 2); // Adres I2C,
      liczba kolumn, liczba wierszy
```

Zakres niniejszego projektu obejmuje kompleksowe opracowanie bezdotykowego systemu pomiarowego temperatury, który łączy optymalne rozwiązania w każdym omawianym później aspekcie projektowym. Projekt podzielony jest na dwie główne części: część programową i konstrukcyjną. Pod uwagę wzięte zostaną także różne istotne czynniki, które wpływają na sposób wykorzystania zbudowanego urządzenia. Istotne dla projektu jest nie tylko samo działanie pirometru, lecz także wpływ środowiska w którym jest użytkowane i kwestia sensownej minimalizacji kosztów utworzenia w pełni funkcjonalnego systemu pomiarowego.

3. Założenia

3.1 Założenia funkcjonalne

3.2 Założenia konstrukcyjne

Funkcjonalność kompletnego urządzenia pozwala na bezproblemowy i możliwie najprostszy w realizacji bezdotykowy pomiar temperatury. Urządzenie dokonuje w czasie rzeczywistym pomiaru temperatury danej powierzchni z wykorzystaniem czujnika MLX90614, wyświetlając wynik na wspomnianym wyświetlaczu LCD.

Za pomocą dołączonej 4-przyciskowej klawiatury można:

- zwiększać wartość emisyjności
- zmniejszać wartość emisyjności
- przywrocić początkową wartość emisyjności wynoszącą 1
- zmieniać jednostkę w której wyświetlany jest wynik wciskając raz za razem przycisk: stopnie Celsjusza, stopnie Fahrenheita, stopnie Kelvina.

Urządzenie wykonane zostało na płycie ewaluacyjnej, która umożliwia korzystanie z urządzenia, minimalizując ryzyko jakiegokolwiek uszkodzenia urządzenia. Układ opierając się konstrukcyjnie na płycie działa stabilnie i daje opcję bezpiecznego przetransportowania przyrządu.

Wykonany w ramach projektu przyrząd pomiarowy składa się z:

- bezdotykowego czujnika temperatury MLX90614, który umożliwia pomiar temperatury obiektu w zakresie -70° do 380°C. Pomiar jest podawany z dokładnością do 0,5°C w zakresie 0-50°C, lub 4°C dla skrajnych wartości zakresu. Natomiast dla temperatury czujnika zakres wynosi od -40°C do 85°C [3].

- wyświetlacza LCD HD44780 z dołączonym konwerterem I2C

- układu sterującego komponentami i przetwarzającymi dane pomiarowe uzyskiwane z czujnika tj. mikrokontrolera Arduino Uno

- 4-przyciskowej klawiatury

Urządzenie działa w odpowiednio szerokim zakresie temperatur, od -20°C do +70°C, aby umożliwić wykorzystanie w różnych warunkach otoczenia. Zostało ono zaprojektowane tak, aby mogło funkcjonować w umiarkowanej wilgotności tj. do bezpiecznej wartości 60% przy temperaturze 25°C, tak by zminimalizować ryzyko kondensacji i uszkodzeń komponentów.

Urządzenie zasilane jest napięciem 5V (możliwe także zasilanie poprzez port USB 2.0). Urządzenie nie zostało przetestowane pod kątem pracy w trudniejszych warunkach środowiskowych. Użyte materiały są odporne na korozję oraz działanie łagodnych substancji chemicznych, co może mieć znaczenie w przypadku zastosowań przemysłowych bądź laboratoryjnych.

Przed wdrożeniem urządzenia do użytku przeprowadzono testy środowiskowe, upewniając się, że spełnia wszystkie założenia dotyczące warunków pracy. Omawiane założenia środowiskowe są kluczowe dla zapewnienia niezawodności i trwałości urządzenia, a także dla jego prawidłowego działania w różnych warunkach otoczenia.

Projekt został opracowany z uwzględnieniem minimalizacji kosztów komponentów z jednoczesnym zachowaniem optymalnej do zastosowań jakości.

Koszty projektu obejmują:

- Czujnik MLX90614 – koszt jednostkowy w przedziale 50–80 zł w zależności od wybranego dostawcy.

- Wyświetlacz LCD HD44780 z konwerterem I2C – koszt jednostkowy w przedziale 20–30 zł.

- Mikrokontroler Arduino Uno – koszt jednostkowy około 100 zł.

- Klawiatura 4-przyciskowa – koszt jednostkowy w przedziale 3–10 zł.

- Laminat miedziowy – koszt jednostkowy w przedziale 20–30 zł.

- Śruby i elementy łączące - koszt około 5-10 zł.

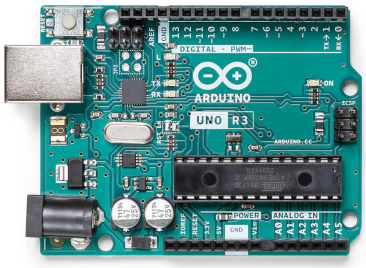
- Poliwęglan stanowiący obudowę – koszt jednostkowy około 30-40 zł

Całkowity koszt komponentów to około 220–28 zł, co czyni projekt relatywnie niedrogim i dość przystępnym cenowo w realizacji. Warto jednak zauważyć, że koszty projektu mogą ulec zmianie w zależności od wybranych dostawców i ilości zakupionych komponentów.

Urządzenie wykonane jest z części ogólnodostępnych. W jego strukturze nie znajdują się żadne elementy, których pozyskanie mogłoby być problematyczne. Największą część kosztów urządzenia stanowi mikrokontroler Arduino Uno. Jest to najdroższy element, ale jednocześnie kluczowy dla działania całego urządzenia. Istnieją tańsze alternatywy, które pozwolą zredukować koszt urządzenia o około 40%. Zamiast oryginalnego mikrokontrolera Arduino Uno można zastosować klon, który jest dostępny na rynku w cenie około 30 zł. Warto jednak zauważyć, że jakość klonów może być niższa niż oryginalnego produktu, co może wpłynąć na stabilną pracę i trwałość

urządzenia. Istnieją zastosowania, gdzie ze względu na całowitą cenę urządzenia, zastosowanie klonu mikrokontrolera może być uzasadnione. W przypadku omawianego projektu, zastosowanie oryginalnego mikrokontrolera Arduino Uno jest zalecane ze względu na jego niezawodność.

4. Opis części sprzętowej



Arduino Uno to popularny mikrokontroler wykorzystywany w projektach elektronicznych i robotyce. Jest to wszechstronna platforma programistyczna oparta na mikrokontrolerze ATmega328P. Mikrokontroler ten posiada 32KB pamięci Flash, 2KB pamięci RAM, 14 cyfrowych pinów wejścia/wyjścia, 6 pinów wejścia analogowego, zegar taktowany z częstotliwością 16MHz, interfejs USB, złącze zasilania 5V oraz złącze programowania ISP. Arduino Uno jest kompatybilny z wieloma dodatkowymi modułami, co pozwala na rozbudowę funkcjonalności. Mikrokontroler ten jest wykorzystywany w projekcie jako główny kontroler systemu.

Programowanie Arduino Uno odbywa się w dedykowanym środowisku Arduino IDE, które wykorzystuje język bazujący na C++. Dzięki rozbudowanej bibliotece funkcji i dużej społeczności użytkowników, realizacja nawet zaawansowanych projektów jest stosunkowo prosta [4]

Czujnik MLX90614 to zaawansowany, bezdotykowy termometr na podczerwień, który umożliwia pomiar temperatury obiektów w szerokim zakresie. Działa na napięciu zasilania od 3V do 3.6V i komunikuje się za pomocą interfejsu I2C, co czyni go łatwym w integracji z różnymi systemami, takimi jak Arduino czy inne mikrokon-

trolery. Dzięki swojej konstrukcji, MLX90614 znajduje zastosowanie w wielu dziedzinach, w tym w medycynie do pomiaru temperatury ciała, w systemach klimatyzacji oraz w automatyzacji przemysłowej [5].

Pokazana na zdjęciu klawiatura to membranowa klawiatura numeryczna, składająca się z czterech przycisków oznaczonych cyframi od 1 do 4. Charakteryzuje się prostą budową, elastyczną taśmą zakończoną złączem z pinami, co umożliwia łatwe podłączenie do mikrokontrolera lub innych urządzeń elektronicznych. Tego typu klawiatury są często wykorzystywane w prostych projektach elektronicznych, takich jak panele sterujące, systemy wprowadzania kodów czy interfejsy użytkownika w urządzeniach DIY. Dzięki niskiej cenie i kompaktowym rozmiarom, są popularnym wyborem wśród hobbystów i studentów elektroniki.

Wyświetlacz LCD z konwerterem I2C oparty na sterowniku HD44780 to popularne rozwiązanie do wyświetlania tekstu w projektach elektronicznych. Dzięki wbudowanemu konwerterowi I2C znacznie uproszczona jest komunikacja z mikrokontrolerem, ponieważ wymaga jedynie dwóch linii sygnałowych (SDA i SCL), za-

miast standardowych 6-8 w przypadku klasycznego podłączenia. Wyświetlacz obsługuje różne konfiguracje, najczęściej spotykane to 16x2 (16 znaków na 2 liniach) lub 20x4 (20 znaków na 4 liniach). Sterownik HD44780 umożliwia łatwe sterowanie wyświetlanymi znakami oraz tworzenie niestandardowych symboli. Dzięki czytelnemu interfejsowi i szerokiemu wsparciu w bibliotekach do Arduino, Raspberry Pi i innych platform, wyświetlacz ten jest chętnie używany w projektach takich jak panele kontrolne, wskaźniki statusu czy urządzenia IoT.

5. Opis części programowej

Czujnik temperatury MLX90614 został podłączony do mikrokontrolera Arduino za pomocą interfejsu I2C. W celu komunikacji z czujnikiem została wykorzystana biblioteka Wire.h. W celu sprawdzenia poprawności połączenia z czujnikiem został napisany program, który odczytuje temperaturę z czujnika i wyświetla ją na monitorze szeregowym i wyświetlaczu LCD.

Wyświetlacz LCD HD44780 podłączono z wykorzystaniem konwertera pracującego na interfejsie I2C. Do komunikacji z wyświetlaczem została użyta biblioteka LiquidCrystal_I2C.h.

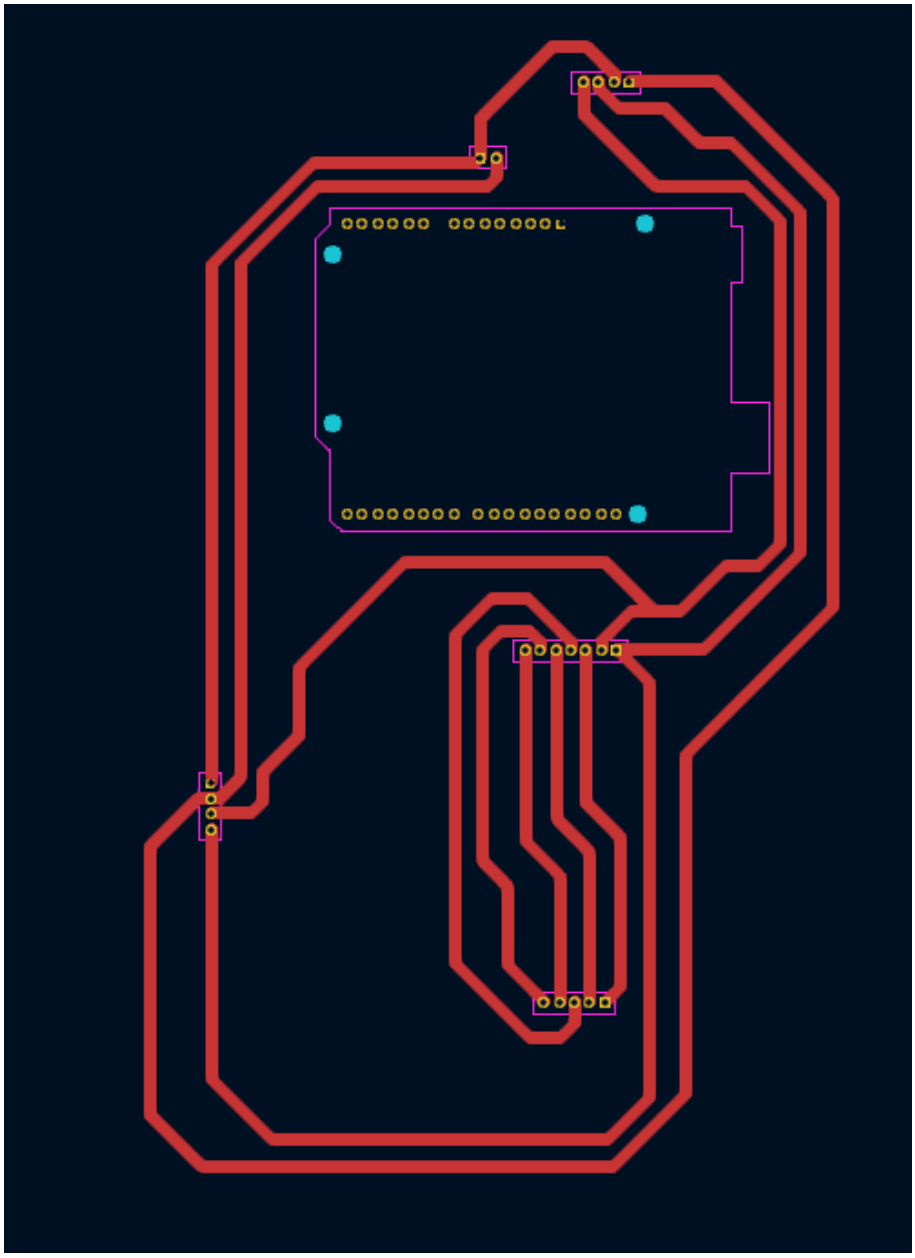
W celu synchronicznej współpracy wyświetlacza LCD i czujnika temperatury z mikrokontrolerem Arduino, został napisany program, który cyklicznie odczytuje temperaturę z czujnika i wyświetla ją na wyświetlaczu LCD. Program został napisany w języku C/C++ z wykorzystaniem bibliotek Wire.h i LiquidCrystal_I2C.h.

Po przetestowaniu komponentów na płytce prototypowej, zaprojektowano docelową płytę ewaluacyjną z wykorzystaniem oprogramowania KiCad 8.0. Z wykorzystaniem wspomnianego oprogramowania, ułożono układ ścieżek na płytce drukowanej oraz rozmieszczono złącza w odpowiednich miejscach.

Płyta ewaluacyjna definiuje rozmiar urządzenia, wynoszący 208 mm x 146 mm, po dodaniu obudowy z płyty poliwęglanowej wysokość urządzenia ma wartość 46 mm.

Płytę ewaluacyjną wykonano z użyciem tradycyjnej technologii termotransferowej, a wytrawianie laminatu przebiegło z użyciem chlorku sodu. Po wykonaniu

odpowiednich otworów w płycie, umieszczono złącza i przewody metodą lutowania THT. Wszystkie elementy urządzenia połączone zostały śrubami oraz tulejami mosiężnymi z gwintami w rozmiarze M3. Całość została obudowana dwiema płytami poliwęglanowymi o grubości 3 mm.



Rysunek 5.1: Rozkład ścieżek na płytce drukowanej

6. Uruchomienie, kalibracja

Poniżej przedstawiono zestaw wzorów opisujących wymianę ciepła przez promieniowanie oraz związane z nimi parametry fizyczne:

- σ – stała Stefana-Boltzmanna, określająca intensywność promieniowania ciała doskonale czarnego,
- ϵ – współczynnik emisyjności (od 0 do 1), opisujący zdolność ciała do emitowania promieniowania w stosunku do ciała doskonale czarnego,
- S – powierzchnia ciała emitującego promieniowanie,
- T_{env} – temperatura otoczenia w stopniach Celcjusza (C),
- T_{meas} – zmierzona temperatura obiektu stopniach Celcjusza (C),
- T_{real} – rzeczywista temperatura obiektu stopniach Celcjusza (C).

1. Moc promieniowania cieplnego emitowanego przez ciało:

$$P = \sigma \cdot \epsilon \cdot S \cdot \left(T_{\text{env}}^4 - T^4\right)$$

2. Równanie równowagi cieplnej opisujące emisję promieniowania:

$$\epsilon \cdot T_{\text{env}}^4 - \epsilon \cdot T_{\text{real}}^4 = T_{\text{env}}^4 - T_{\text{meas}}^4$$

3. Współczynnik emisyjności obliczony na podstawie temperatur:

$$\epsilon = \frac{T_{\text{env}}^4 - T_{\text{meas}}^4}{T_{\text{env}}^4 - T_{\text{real}}^4}$$

Na podstawie dwóch różnych temperatur wyznaczono emisyjność badanego obiektu.

Wzory zostały zastosowane do obliczenia wartości emisyjności:

$$\epsilon = \frac{30.15^4 - 69.31^4}{30.15^4 - 70.1^4} = 0.9541$$

Dokładny wynik obliczenia przed zaokrągleniem wynosi: 0.9540835302172766, po zaokrągleniu do czterech miejsc po przecinku wynik to: 0.9541.

7. Pomiary testowe

W celu weryfikacji poprawności działania pirometru, przeprowadzono test porównawczy z wykorzystaniem wzorcowanego pirometru przemysłowego Sonel DIT-200 wraz z opcją pomiaru temperatury metodą stykową. Test przeprowadzono w kontrolowanych warunkach laboratoryjnych w przedziale temperatury 25 - 30°C. Zakres temperatury mierzonego obiektu wynosi od 35°C do 160°C. Obiektem pomiarowym jest płyta grzejna o mocy 800W, pomalowanego na kolor czarny matowy.

Procedura testowa:

1. Przygotowanie stanowiska składającego się z urządzenia testowanego, pirometru przemysłowego wyposażonego w pomiar metodą optyczną oraz stykową oraz obiektu odwzorowującego wymagane nastawy temperatury.

2. Wykonanie pomiarów temperatury w zakresie od 35°C do 160°C w odstępach co 5°C oraz zarejestrowanie ich w arkuszu kalkulacyjnym.

3. Dwukrotne powtórzenie pomiarów. Każdy pomiar był wykonywany trzykrotnie, a wyniki uśredniano w celu zwiększenia dokładności pomiarów.

Temperatura z Arduino [°C]	**Termometr stykowy [°C]**	**Pirometr przemysłowy [°C]**
160	160	164.7
155	156.1	159.3
150	151.8	156.6
145	146.9	152.9
140	142.5	147.2
135	137.2	142.1
130	132.5	139.2
125	127.8	131.3
120	122.3	126.3
115	117.4	120.5
110	112.7	115.6
105	106.9	110.7
100	101	105
95	94.8	100
90	89.7	96.2
85	83.8	92.4
80	79.3	88.5
75	73.3	84.1
70	68.7	80.1
65	63.8	75.7
60	58.7	70.7
55	54.1	66
50	51	60.4
45	45.4	54.2
40	41.1	42.1
35	37	36.1

Tabela 7.1: Porównanie pomiarów temperatury dla różnych urządzeń.

Rysunek 7.1: Przebieg procedury testowej

8. Instrukcja obsługi dla użytkownika

8.1 Krótki opis pirometru i jego przeznaczenia

Poniższy pirometr jest projektem naukowo-badawczym, służącym do bezkontaktowego pomiaru temperatury oraz wyświetlania jej w czasie rzeczywistym na wbudowanym wyświetlaczu LCD. Urządzenie oferuje możliwość wyboru wyświetlania temperatury według skali Celsjusza, Fahrenheita oraz Kelwina. Ponadto, miernik umożliwia dostosowanie emisyjności w zakresie od 0.00 do 1.00.

Obudowa urządzenia jest niepełna, charakteryzująca się prostym, schludnym i nowoczesnym wyglądem. Ze względu na delikatne elementy, urządzenie jest nieodpowiednie do użytkowania przez dzieci. Posiada klasę odporności IP10, co oznacza, że nie należy narażać go na kontakt ze skutkami opadów atmosferycznych oraz przypadkowym zalaniem. Czyszczenie urządzenia powinno odbywać się z użyciem ściereczki nasączonej wodą, ewentualnie z dodatkiem delikatnego detergentu. W przypadku uszkodzenia urządzenia lub jego nieprawidłowego działania, należy skontaktować się z producentem lub oddać urządzenie do punktu naprawczego.

Urządzenie składa się z następujących elementów:

- dwóch paneli poliwęglanowych o grubości 3 mm stanowiących obudowę,
- płyty ewaluacyjnej,
- modułu Arduino Uno pełniącego rolę serca urządzenia,
- czujnika pirometrycznego,
- klawiatury,
- wyświetlacza LCD.

8.2 Dane techniczne

- **Prędkość próbkowania:** 1/sekundę,
- **Zasilanie:** USB-B 5V,
- **Pobór prądu:** >50 mA,
- **Klasa odporności:** IP10,
- **Zakres pomiaru temperatury:** od -30 do 150 °C,
- **Dopuszczalna wilgotność względna bez kondensacji:** 5–95%,
- **Temperatura pracy oraz przechowywania:** od -10 do 50 °C,
- **Wymiary:** 160 x 200 mm,
- **Waga:** 350 g.

8.3 Obsługa urządzenia

Po podłączeniu urządzenia do zasilania za pomocą kabla USB-B, jest ono natychmiast gotowe do pracy i wykonuje pomiary. Należy skierować przednią część urządzenia (w której znajduje się element pomiarowy) na badany obiekt. Wyniki pomiarów są wyświetlane na ekranie LCD w czasie rzeczywistym.

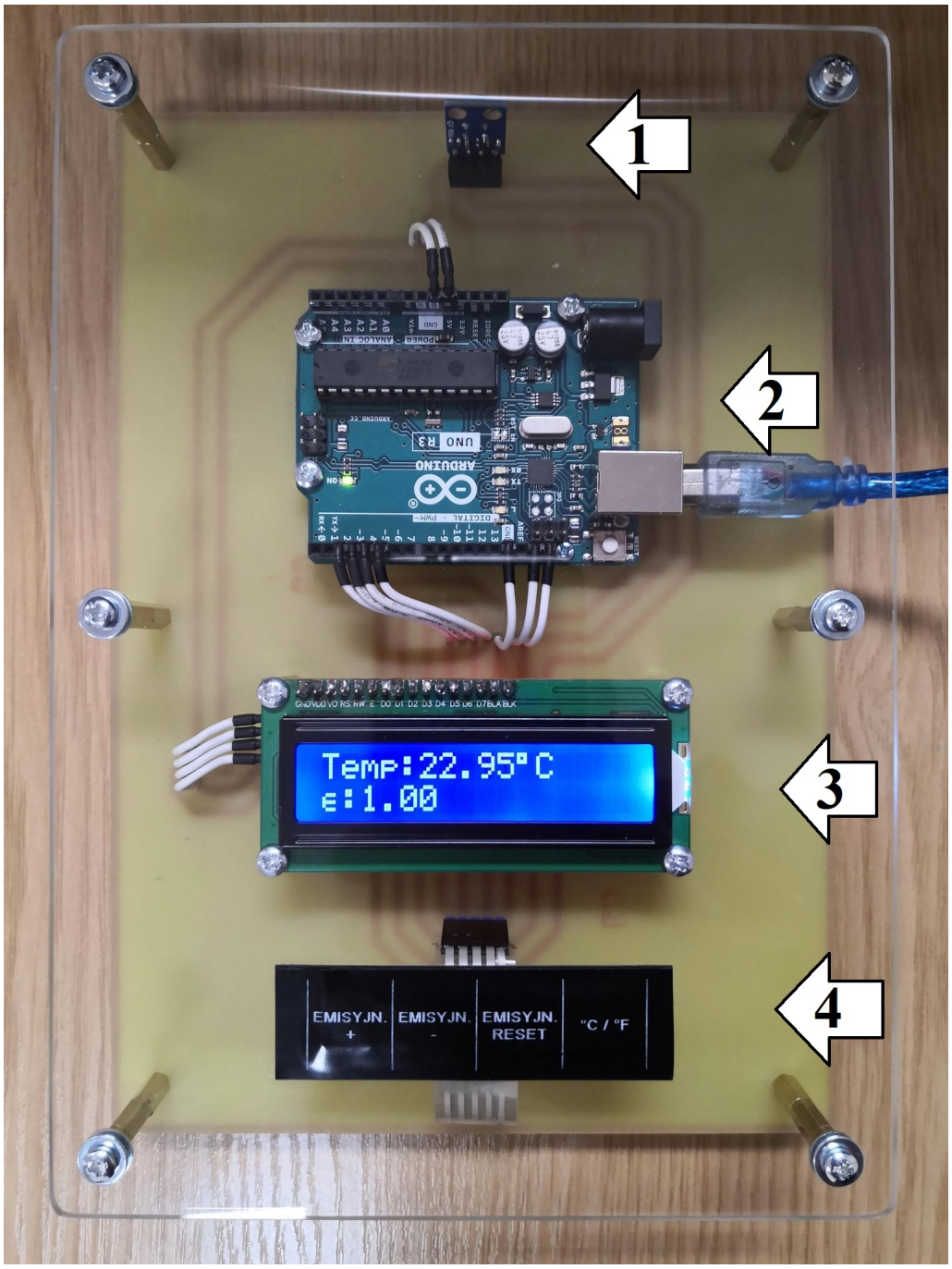
Ustawienie emisyjności w zakresie od 0.00 do 1.00 odbywa się za pomocą klawiszy `EMISYJN. +` oraz `EMISYJN. -`. Tabela emisyjności typowych materiałów znajduje się na rysunku 4. Prawidłowe ustawienie emisyjności ma istotny wpływ na dokładność pomiarów i powinno być dostosowywane przy każdej zmianie badanej powierzchni. W celu przywrócenia wartości domyślnej emisyjności (1.00), należy użyć przycisku `EMISYJN. RESET`.

Zmianę skali temperatury pomiędzy stopniami Celsjusza, Fahrenheita oraz Kelwina wykonuje się przyciskiem `°C / °F`. Po zakończeniu pomiarów należy odłączyć przewód zasilający od urządzenia. Pirometr powinien być przechowywany w suchym miejscu.

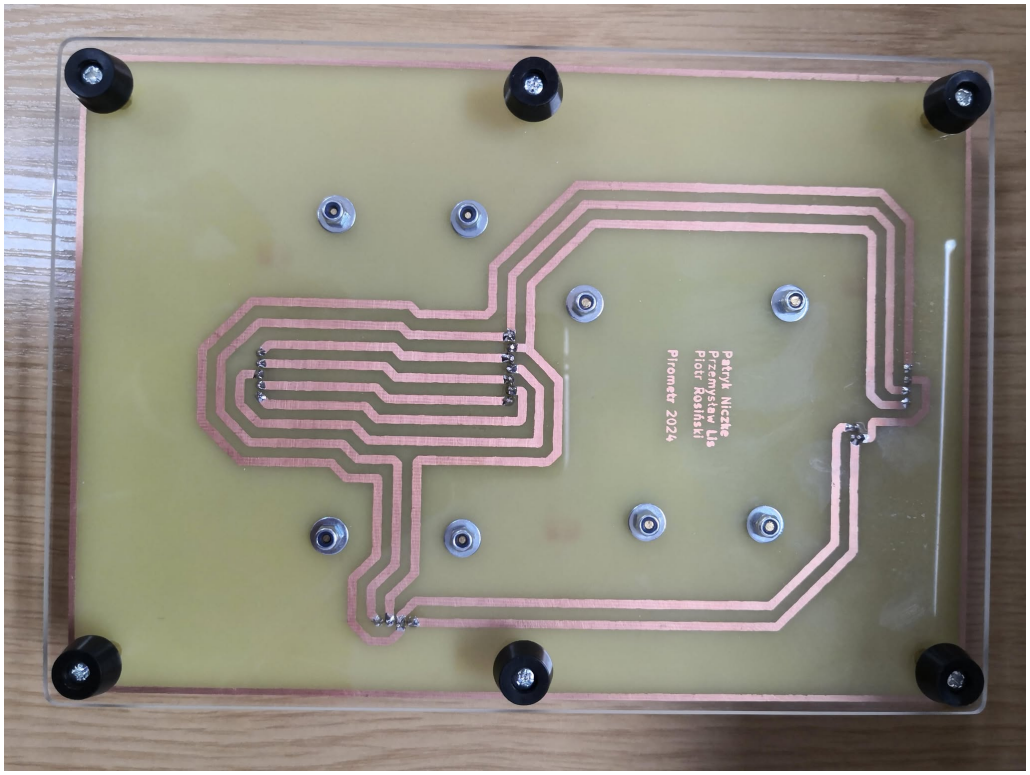
8.4 Opis budowy urządzenia

Rys.1 przedstawia panel górny urządzenia. Najważniejsze jego elementy to:

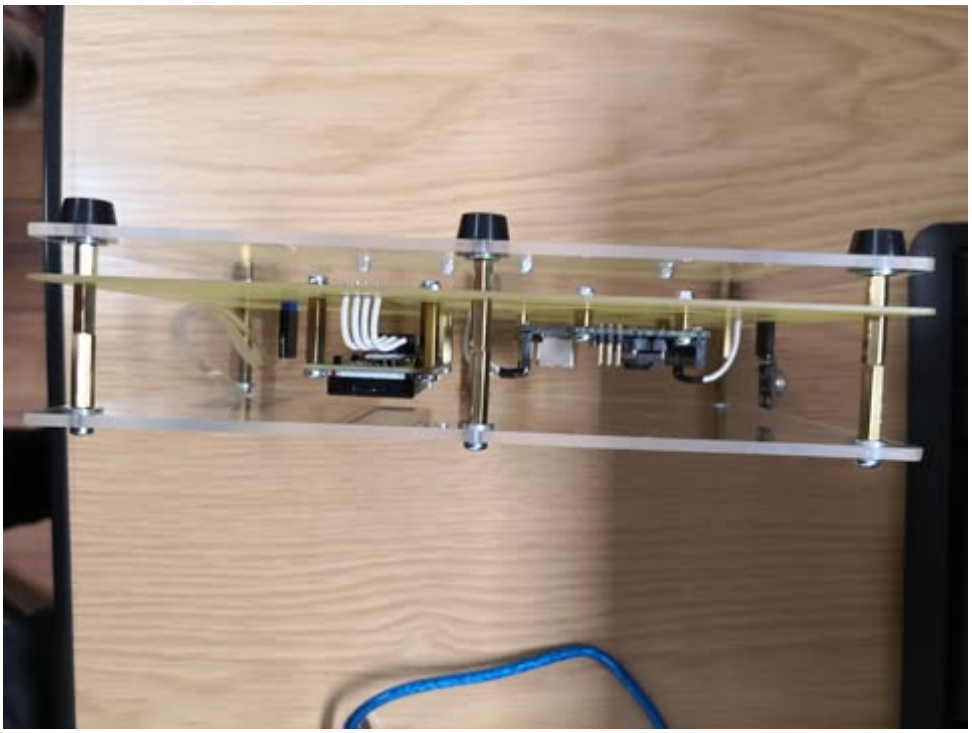
1. Element pirometryczny służący do pomiaru temperatury.
2. Moduł Arduino Uno z gniazdem zasilającym urządzenie oraz interfejsem USB-C.
3. Wyświetlacz LCD z niebieskim podświetleniem.
4. Klawiatura służąca do wprowadzania ustawień urządzenia.



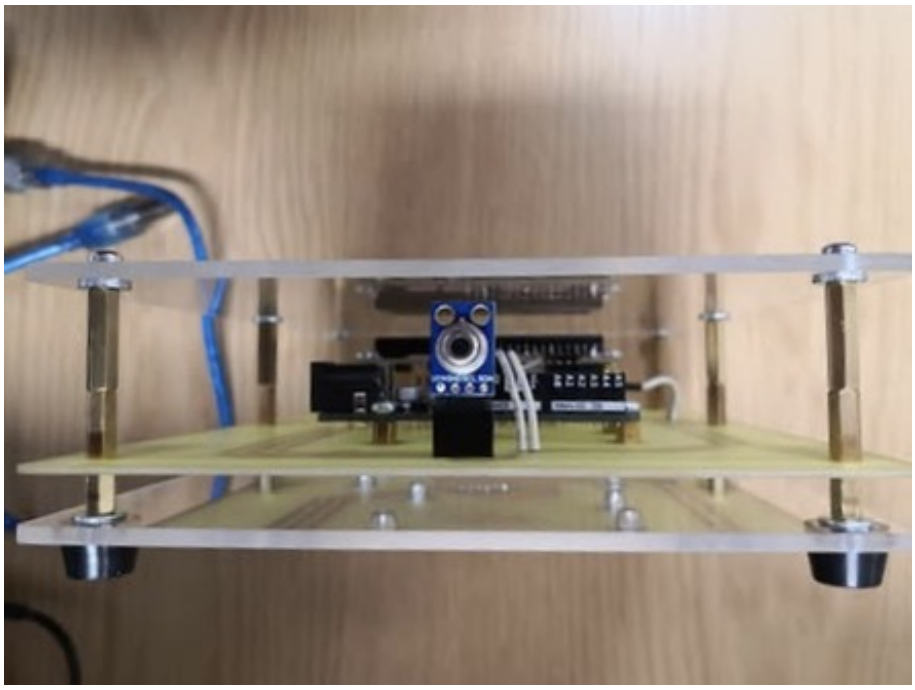
Rysunek 8.1: Panel górny urządzenia



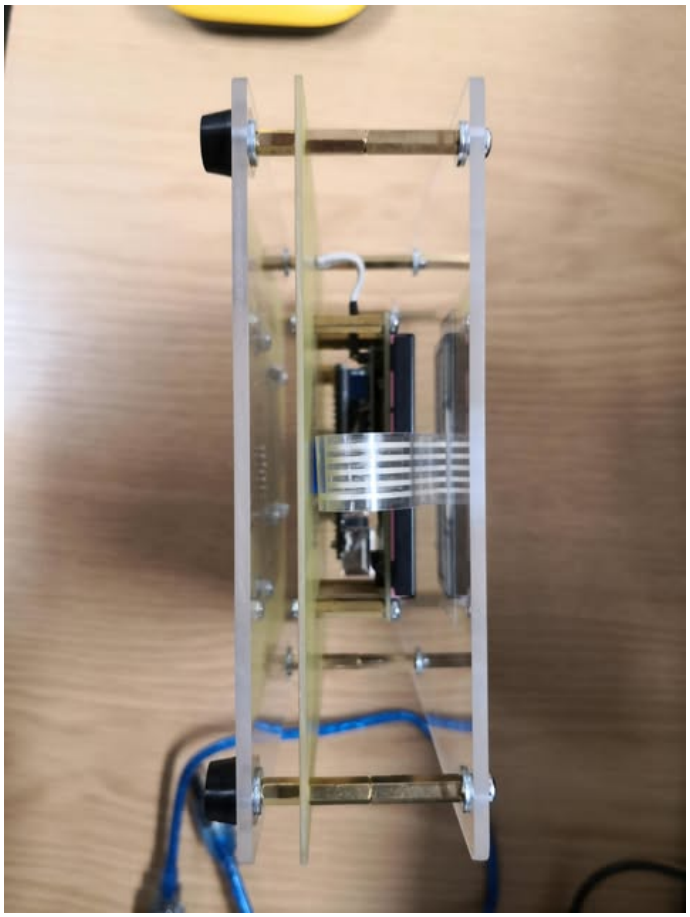
Rysunek 8.2: Panel tylny urządzenia



Rysunek 8.3: Widok z boku



Rysunek 8.4: Widok z boku



Rysunek 8.5: Widok z boku

9. Podsumowanie

Wyniki testów zostały przedstawione w tabeli 7.1. Obserwacje wskazują, że:

- Wyniki pirometru przemysłowego są zbliżone do wyników pirometru testowego, z odchyleniem nieprzekraczającym 4°C w najwyższych temperaturach.
- Pirometr przemysłowy, jako urządzenie profesjonalne, rejestruje wyższe wartości temperatur w całym zakresie. Rozbieżności te mogą być związane z niedoskonałością kalibracji pirometru Arduino, lub ograniczoną rozdzielczością czujnika MLX90614.
- Termometr stykowy wykazuje wyższe różnice w niższych temperaturach, co może wynikać z bezwładności termicznej sondy stykowej lub nierównomierności.
- Wraz ze wzrostem temperatury, różnice między wynikami z pirometru Arduino a pirometru przemysłowego stają się bardziej widoczne. Pirometr Arduino odnotowuje wartości niższe niż pirometr przemysłowy w wyższych zakresach temperatur (powyżej 120°C), co sugeruje możliwość systematycznych błędów wynikających z niedoskonałości kalibracji.
- Wraz ze wzrostem temperatury, różnice między wynikami z pirometru Arduino a pirometru przemysłowego stają się bardziej widoczne. Pirometr Arduino odnotowuje wartości niższe niż pirometr przemysłowy w wyższych zakresach temperatur (powyżej 120°C), co sugeruje możliwość systematycznych błędów wynikających z niedoskonałości kalibracji.

- Wraz ze wzrostem temperatury, różnice między wynikami z pirometru Arduino a pirometru przemysłowego stają się bardziej widoczne. Pirometr Arduino odnotowuje wartości niższe niż pirometr przemysłowy w wyższych zakresach temperatur (powyżej 120°C), co sugeruje możliwość systematycznych błędów wynikających z niedoskonałości kalibracji.

Podsumowując, pirometr spełnia założenia projektowe i może być stosowany do bezkontaktowego pomiaru temperatury w zakresie od 35°C do 160°C.

- **Emisyjność:** Ustawienia emisyjności mają kluczowy wpływ na wyniki pomiarów. Niedokładne dobranie tej wartości dla badanych materiałów może prowadzić do błędów w pomiarach.
- **Kalibracja:** Pirometr Arduino, jako urządzenie prototypowe, nie posiada profesjonalnej kalibracji fabrycznej, co wpływa na dokładność pomiarów.
- **Czujnik MLX90614:** Czujnik zastosowany w urządzeniu charakteryzuje się ograniczoną dokładnością w wyższych zakresach temperatur, co mogło wpłynąć na odchylenia w pomiarach.
- **Wpływ środowiska:** Czynniki takie jak wilgotność, temperatura otoczenia czy odbicia promieniowania podczerwonego mogą wprowadzać dodatkowe błędy.

```
1  #include <Arduino.h>
2  #include <Adafruit_MLX90614.h>
3  #include <LiquidCrystal_I2C.h>
4  #include <math.h>
5
6  // Define buttons
7  #define BTN1 2
8  #define BTN2 3
9  #define BTN3 4
```

```
10      #define BTN4 5
11
12      // Set I2C address for the LCD (change if needed, e.g
           ., 0x27)
13      LiquidCrystal_I2C lcd(0x27, 16, 2); // 16 chars, 2
           lines
14
15      // Custom epsilon character
16      byte epsilon[8] = { B00000, B00000, B01110, B10000,
           B11110, B10000, B01110, B00000 };
17
18      // Initialize MLX90614 sensor
19      Adafruit_MLX90614 mlx = Adafruit_MLX90614();
20      float ems = 1.0;
21      // Default emissivity
22      int tempScale = 0; // 0 - Celsius, 1 - Fahrenheit, 2
           - Kelvin
23
24      float correctTemperature(float measuredTemp, float
           ambientTemp, float emissivity) {
25          float measuredTempK = measuredTemp + 273.15;
26          float ambientTempK = ambientTemp + 273.15;
27          float trueTempK = pow((pow(measuredTempK, 4) - (1
                - emissivity) * pow(ambientTempK, 4)) /
               emissivity, 0.25);
28          return trueTempK - 273.15;
29      }
30
31      void setup() {
```

```
32          delay(200);
33          pinMode(BTN1, INPUT_PULLUP);
34          pinMode(BTN2, INPUT_PULLUP);
35          pinMode(BTN3, INPUT_PULLUP);
36          pinMode(BTN4, INPUT_PULLUP);
37          Serial.begin(9600);
38          lcd.init();
39          lcd.clear();
40          lcd.backlight();
41          lcd.setCursor(0, 0);
42          lcd.createChar(0, epsilon);
43          if (isnan(ems)) {
44              ems = 1.0;
45          }
46          if (!mlx.begin()) {
47              lcd.setCursor(0, 1);
48              lcd.print("MLX error!");
49              Serial.print("MLX error!\n");
50              while (1);
51          }
52      }
53
54      void loop() {
55          int BTN1V = digitalRead(BTN1);
56          int BTN2V = digitalRead(BTN2);
57          int BTN3V = digitalRead(BTN3);
58          int BTN4V = digitalRead(BTN4);
59
60          if (!BTN1V && ems < 1.0) {
```

```
61              Serial.println("Increased emissivity");
62              ems += 0.01;
63              if (ems > 1.0) ems = 1.0;
64          }
65
66          if (!BTN2V && ems > 0.0) {
67              Serial.println("Decreased emissivity");
68              ems -= 0.01;
69              if (ems < 0.0) ems = 0.0;
70          }
71
72          if (!BTN3V) {
73              Serial.println("Emissivity reset");
74              ems = 1.0;
75          }
76
77          if (!BTN4V) {
78              tempScale = (tempScale + 1) % 3;
79              Serial.println(tempScale == 0 ? "Switched to
                    Celsius" : (tempScale == 1 ? "Switched to
                    Fahrenheit" : "Switched to Kelvin"));
80              delay(300);
81          }
82
83          float ObjTemp = mlx.readObjectTempC();
84          float AmbientTemp = mlx.readAmbientTempC();
85          float correctedTemp = correctTemperature(ObjTemp,
                AmbientTemp, ems);
86
```

```
87          if (isnan(correctedTemp)) {
88              Serial.println("Read error: Temperature NaN")
                    ;
89              correctedTemp = 0.0;
90          }
91
92          float displayTemp = correctedTemp;
93          char scaleLabel = 'C';
94          bool showDegreeSymbol = true;
95
96          if (tempScale == 1) {
97              displayTemp = correctedTemp * 9.0 / 5.0 +
                    32.0;
98              scaleLabel = 'F';
99          } else if (tempScale == 2) {
100             displayTemp = correctedTemp + 273.15;
101             scaleLabel = 'K';
102             showDegreeSymbol = false;
103         }
104
105         Serial.print("Temperature: ");
106         Serial.print(displayTemp);
107         Serial.print(" ");
108         Serial.print(scaleLabel);
109         Serial.print("\nEnvironment Temperature: ");
110         Serial.print(AmbientTemp);
111         Serial.print("\nEmissivity: ");
112         Serial.println(ems);
113
```

```
114         lcd.setCursor(0, 0);
115         lcd.print("Temp:");
116         lcd.print(displayTemp);
117         if (showDegreeSymbol) {
118             lcd.print((char)223);
119         }
120         lcd.print(scaleLabel);
121 
122         lcd.setCursor(0, 1);
123         lcd.write(byte(0));
124         lcd.print(":");
125         lcd.print(ems);
126         delay(500);
```

10. Dodatki

Bibliografia

[1] Sanchez, Julio; Canton, Maria P. (2007) - „Microcontroller Programming: the Microchip PIC. CRC Press."

[2] ElektronikaB2B - „Arduino - jak wybrać i kupić?"

`https://elektronikab2b.pl/technika/50150-arduino-jak-wybrac-i-kupic`

[3] Korneliusz Jarzębski - „Pirometr z czujnikiem MLX90614ESF-BAA"

`https://www.jarzebski.pl/arduino/czujniki-i-sensory/pirometr-z-czujnikiem-mlx90614.html`

[4] Arduino - „Arduino Uno Rev3"

`https://docs.arduino.cc/hardware/uno-rev3/#features`

[5] ElektroWeb - „Pirometr termometr bezdotykowy MLX90614 GY-906"

`https://www.elektroweb.pl/pl/czujniki-temperatury/273-pirometr-termometr-bezdotykowy-mlx90614-gy-906.html`