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

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.

Zakres niniejszego projektu obejmuje kompleksowe opracowanie optycznego pomiaru temperatury, który łączy optymalne rozwiązania zarówno w obszarze sprzętowym, jak i programowym. Projekt podzielony jest na dwie główne części: część sprzętową i programową. Pod uwagę wzięte zostaną także różnorodne istotne czynniki, które wpływają na sposób wykorzystania zbudowanego urządzenia. Kluczowe dla projektu jest nie tylko samo działanie pirometru, lecz także wpływ środowiska, w którym jest użytkowane, i kwestia racjonalnej minimalizacji kosztów utworzenia w pełni funkcjonalnego systemu pomiarowego.

Po wstępie i założeniach projektowych następuje szczegółowy opis części sprzętowej oraz programowej systemu. Raport obejmuje proces uruchomienia i kalibracji urządzenia wraz z wynikami pomiarów testowych. Dla użytkowników przygotowano instrukcję obsługi. Całość uzupełniają podsumowanie, dodatki oraz bibliografia.

# 2. Wprowadzenie

**Temperatura** jest fundamentalnym parametrem fizycznym, który charakteryzuje stan energetyczny układu termodynamicznego. Stanowi ona miarę średniej energii kinetycznej cząsteczek materii, objawiającej się poprzez ich chaotyczny ruch i drgania. W praktyce inżynierskiej i życiu codziennym temperatura jest kluczowym parametrem determinującym kierunek przepływu energii cieplnej między ciałami oraz wpływającym na właściwości fizyczne materiałów.

W układzie SI przyjęto skalę Kelwina (K) jako podstawową jednostkę temperatury, definiując ją poprzez stałą Boltzmanna i energię kinetyczną cząsteczek. Historycznie wykształciły się również inne skale temperatur - skala Celsjusza (°C), powszechnie stosowana w Europie, oraz skala Fahrenheita (°F), popularna głównie w Stanach Zjednoczonych. Każda z tych skal ma swoje charakterystyczne punkty odniesienia, co przedstawiono w tabeli 2.1.

| **Punkt charakterystyczny** | **Skala temperatur** | | |
|---|---|---|---|
| | **Fahrenheit** | **Kelvin** | **Celsjusz** |
| *Punkty podstawowe* | | | |
| Zero bezwzględne | -459,67 | 0 | -273,15 |
| Zero Fahrenheita | 0 | 255,37 | -17,78 |
| *Punkty charakterystyczne wody* | | | |
| Zamarzanie | 32 | 273,15 | 0 |
| Wrzenie | 212 | 373,15 | 100 |
| *Temperatura biologiczna* | | | |
| Średnia temp. ciała człowieka | 98,2 | 309,8 | 36,6 |

Tabela 2.1: Porównanie podstawowych skal temperatury

Szczególne znaczenie w fizyce ma pojęcie zera bezwzględnego (0 K), które stanowi

teoretyczną granicę najniższej możliwej temperatury. W tej temperaturze ustaje ruch termiczny cząsteczek, a układ osiąga minimum energii. Jest to stan nieosiągalny w praktyce, choć współczesne laboratoria potrafią zbliżyć się do niego na milionowe części Kelwina.

Temperatura odgrywa kluczową rolę w procesach wymiany ciepła. Zgodnie z drugą zasadą termodynamiki, energia cieplna przepływa samorzutnie tylko w jednym kierunku - od ciał o wyższej temperaturze do ciał o temperaturze niższej. Proces ten trwa do momentu osiągnięcia równowagi termodynamicznej, czyli wyrównania temperatur wszystkich ciał uczestniczących w wymianie ciepła. Ta fundamentalna zasada ma istotne znaczenie w projektowaniu systemów pomiarowych i kontroli procesów przemysłowych.

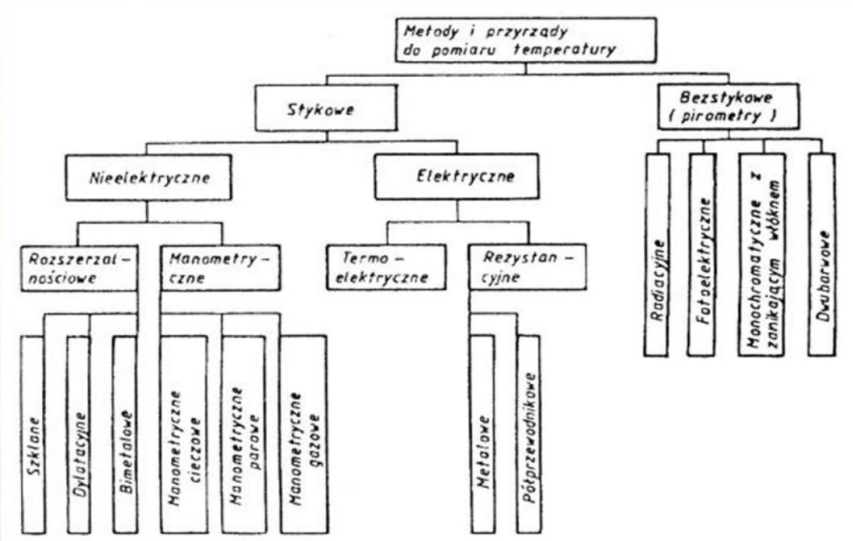
W metrologii optycznej pomiar temperatury opiera się na fundamentalnej własności materii - emisji promieniowania termicznego. Każde ciało o temperaturze wyższej od zera bezwzględnego emituje promieniowanie elektromagnetyczne, którego charakterystyka spektralna i intensywność są ściśle powiązane z jego temperaturą. Współczesne metody pomiaru wykorzystują to zjawisko na kilka sposobów.

**Podstawowe metody pomiaru temperatury** można sklasyfikować na dwie główne kategorie: stykowe i bezstykowe jak pokazano na rysunku 2.1. Kluczowy jest sposób oddziaływania czujnika z obiektem. Metody stykowe wymagają fizycznego kontaktu czujnika z powierzchnią obiektu, co pozwala na przekazywanie ciepła poprzez przewodzenie. Z kolei metody bezstykowe opierają się na detekcji promieniowania cieplnego emitowanego przez obiekty, które mają temperaturę wyższą niż zero absolutne (0 K). Każdy obiekt emituje to promieniowanie, co umożliwia pomiar bez konieczności bezpośredniego dotyku.

Technologie bezstykowe, takie jak pirometry podczerwieni czy kamery termowizyjne, zrewolucjonizowały sposób monitorowania temperatury, oferując szybkie, precyzyjne i bezpieczne pomiary nawet na odległość. Są one nieocenione w zasto-

sowaniach przemysłowych, medycznych czy naukowych, gdzie bezpośredni kontakt z obiektem może być trudny, niebezpieczny lub wręcz niemożliwy. Warto jednak pamiętać, że w przypadku ciał stałych i cieczy promieniowanie cieplne pochodzi głównie z warstwy przypowierzchniowej, dlatego kluczowe jest uwzględnienie współczynnika emisyjności materiału, aby uniknąć błędów w interpretacji wyników.

Co więcej, rozwój technologii bezstykowych wpływa również na minimalizację wpływu na badane obiekty. W przypadku delikatnych materiałów czy żywych organizmów, metody bezstykowe eliminują ryzyko uszkodzenia czy zniekształcenia wyników pomiarów.




Rysunek 2.1: Podstawowe metody pomiaru temperatury [1]

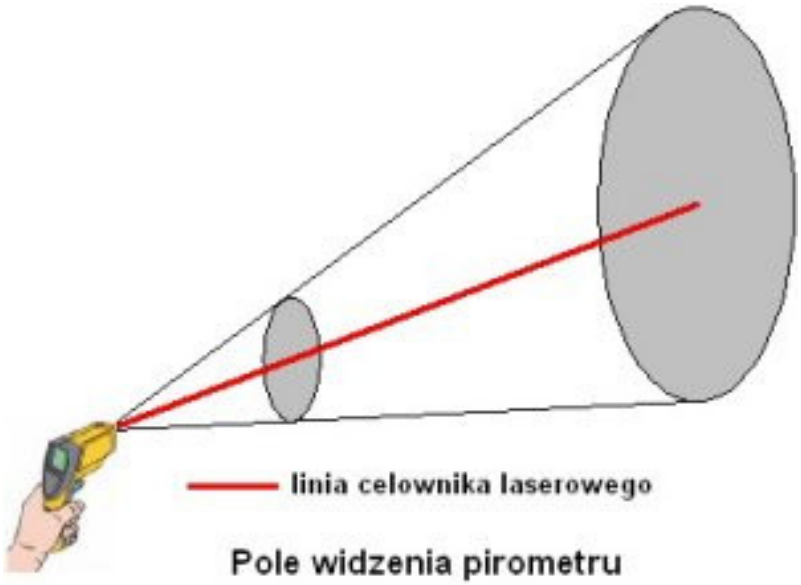
Współczesne rozwiązania pomiarowe idą jednak o krok dalej, integrując zaawansowane algorytmy przetwarzania obrazu termicznego oraz techniki sztucznej inteligencji. Dzięki temu możliwe jest nie tylko dokładniejsze monitorowanie temperatury, ale także automatyzacja procesów decyzyjnych. Przykładowo, systemy oparte na AI mogą analizować trendy temperaturowe w czasie rzeczywistym, przewidywać potencjalne awarie w maszynach czy optymalizować zużycie energii w inteligentnych budynkach. To otwiera nowe możliwości w zarządzaniu energią, diagnostyce medycznej czy nawet w badaniach klimatycznych.

## Zasada działania pirometru

Każdy przedmiot materialny emituje promieniowanie podczerwone (cieplne), niewidoczne dla oczu, ale wyczuwalne np. przy zbliżeniu ręki do gorącego żelazka. Natężenie tego promieniowania jest tym większe, im wyższa jest temperatura przedmiotu. Pirometr mierzy natężenie promieniowania podczerwonego dochodzącego od przedmiotu do jego obiektywu. Zmierzoną wielkość promieniowania przyrząd przelicza na odpowiadającą jej temperaturę przedmiotu i pokazuje wartość tej temperatury na wyświetlaczu.

## Pole widzenia pirometru

Standardowy przenośny pirometr umożliwia pomiar średniej temperatury powierzchni kołowej o średnicy kilku centymetrów z odległości 1 metra. Wraz ze zwiększaniem się odległości pirometru od mierzonego obiektu, rośnie również średnica obszaru, z którego urządzenie rejestruje promieniowanie podczerwone, co doskonale obrazuje rysunek 2.2.

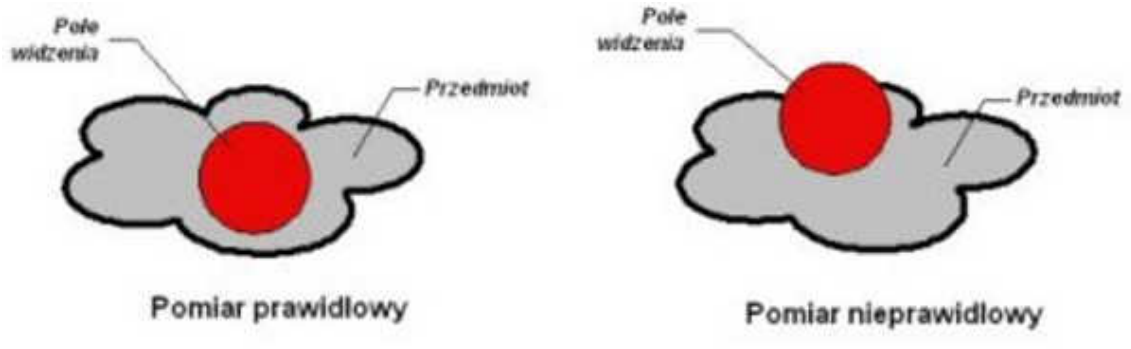



Rysunek 2.2: Zależność pola widzenia od odleglości do mierzonego obiektu [2]

Aby ułatwić precyzyjne zlokalizowanie obszaru pomiarowego, pirometry są zazwyczaj wyposażone w celownik laserowy. Promień lasera widoczny jest jako czerwona plamka na powierzchni mierzonego obiektu. Jej położenie wyznacza środek koła, które określa pole widzenia pirometru.

## Pomiary przy użyciu pirometru

Aby wykonać prawidłowy pomiar, pole widzenia pirometru nie może wychodzić poza przedmiot, którego temperaturę mierzymy. W przeciwnym razie pirometr będzie zbierał promieniowanie podczerwone nie tylko z przedmiotu, ale także z otoczenia ($t_a$), co spowoduje błędny wynik pomiaru. Na rysunku 2.4 przedstawiono prawidłowe oraz nieprawidłowe ustawienie pola widzenia pirometru względem powierzchni mierzonego obiektu.




Rysunek 2.3: Prawidłowy i nieprawidłowy pomiar z wykorzystaniem pirometru [2]

## Współczynnik emisyjności powierzchni a dokładność pomiaru

Materiały mają różną zdolność wysyłania promieniowania podczerwonego ze swojej powierzchni. Właściwość ta zależy od gładkości i barwy powierzchni. Materiały o powierzchniach matowych i ciemnych lepiej emitują promieniowanie podczerwone niż materiały o powierzchniach gładkich i jasnych.

Współczynnik emisyjności określa się w zakresie od 0 do 1. Przykładowo, współczynnik emisyjności powierzchni cegły wynosi 0.85, a powierzchni polakierowanej czarnym lakierem matowym 0.97. Aluminium ma współczynnik emisyjności 0.07. W celu otrzymania prawidłowego wyniku pomiaru pirometrem należy wartość współczynnika emisyjności danej powierzchni wprowadzić do pamięci wewnętrznej przyrządu.

# Prawa fizyczne i wzory

**Prawo Wiena** (2.1) określa zależność między temperaturą ciała doskonale czarnego a długością fali, przy której emituje ono najwięcej promieniowania. Formuła ta jest wykorzystywana do określania temperatury na podstawie emitowanego promieniowania.

$$\lambda_{MAX} = \frac{b}{T} \tag{2.1}$$

gdzie:

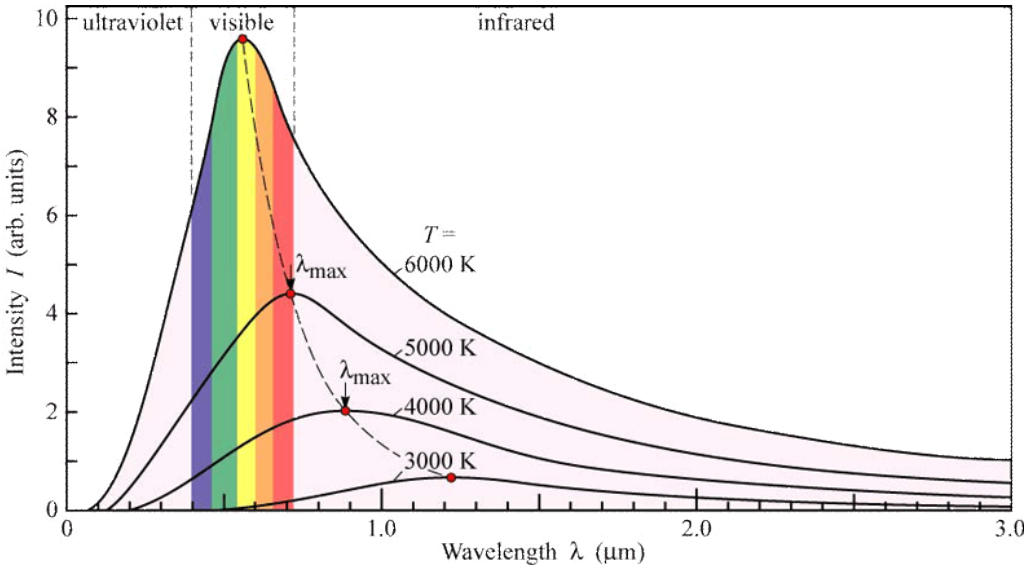
- $\lambda_{MAX}$ – długość fali o maksymalnej emisji,
- $T$ – temperatura,
- $b$ – stała Wiena.

**Prawo Stefana-Boltzmanna** (2.2) mówi, że całkowita energia promieniowania ciała doskonale czarnego jest proporcjonalna do czwartej potęgi jego temperatury

$$E = \sigma T^4 \tag{2.2}$$

**Ciało doskonale czarne** to teoretyczny obiekt fizyczny, który całkowicie pochłania padające na niego promieniowanie elektromagnetyczne, niezależnie od długości fali. Nie odbija ani nie przepuszcza żadnego promieniowania. W praktyce ciało

doskonale czarne jest również idealnym emitentem promieniowania cieplnego, co oznacza, że emituje maksymalną możliwą ilość promieniowania dla danej temperatury. Jego emisyjność wynosi 1.




Rysunek 2.4: Rozkład energii w widmie promieniowania ciała doskonale czarnego w różnych temperaturach

[3]

**Współczynnik emisyjności** ($\varepsilon$) opisuje, jaką część energii promieniowania emitowanego przez ciało doskonale czarne emituje dane ciało w tej samej temperaturze. Wartość współczynnika emisyjności $\varepsilon$ jest zazwyczaj dostarczana przez producenta.

Współczynnik emisyjności obliczony na podstawie zmierzonych temperatur:

$$\varepsilon = \frac{T_{O_1}^4 - T_{A_1}^4}{T_{O_2}^4 - T_{A_2}^4}$$

gdzie:

- $T_{O_1}$ – zmierzona temperatura obiektu (w kelwinach),
- $T_{A_1}$ – zmierzona temperatura otoczenia (w kelwinach),
- $T_{O_2}$ – rzeczywista temperatura obiektu (w kelwinach),
- $T_{A_2}$ – rzeczywista temperatura otoczenia (w kelwinach).

# 3. Założenia

## 3.1 Założenia funkcjonalne

Funkcjonalność urządzenia umożliwia szybki i intuicyjny bezdotykowy pomiar temperatury. W czasie rzeczywistym urządzenie mierzy temperaturę danej powierzchni za pomocą czujnika MLX90614, a wynik jest wyświetlany na ekranie LCD.

## 3.2 Założenia konstrukcyjne

Podstawowe komponenty urządzenia:

- **Czujnik temperatury MLX90614** – umożliwia precyzyjny pomiar temperatury obiektu w zakresie od -70°C do 380°C. Pomiar jest podawany z dokładnością do 0,5°C w zakresie 0–50°C lub 4°C dla wartości skrajnych. Zakres temperatury czujnika wynosi od -40°C do 85°C [4].
- **Mikrokontroler Arduino Uno** – steruje komponentami urządzenia i przetwarza dane z czujnika.
- **Wyświetlacz LCD HD44780** z konwerterem I2C – służy do prezentowania wyników pomiarów.
- **4-przyciskowa klawiatura** – umożliwia interakcję z urządzeniem.

## Płyta ewaluacyjna

Urządzenie zostało zbudowane na płycie ewaluacyjnej, co zapewnia stabilność pracy i minimalizuje ryzyko uszkodzeń. Konstrukcja umożliwia bezpieczny transport urządzenia.

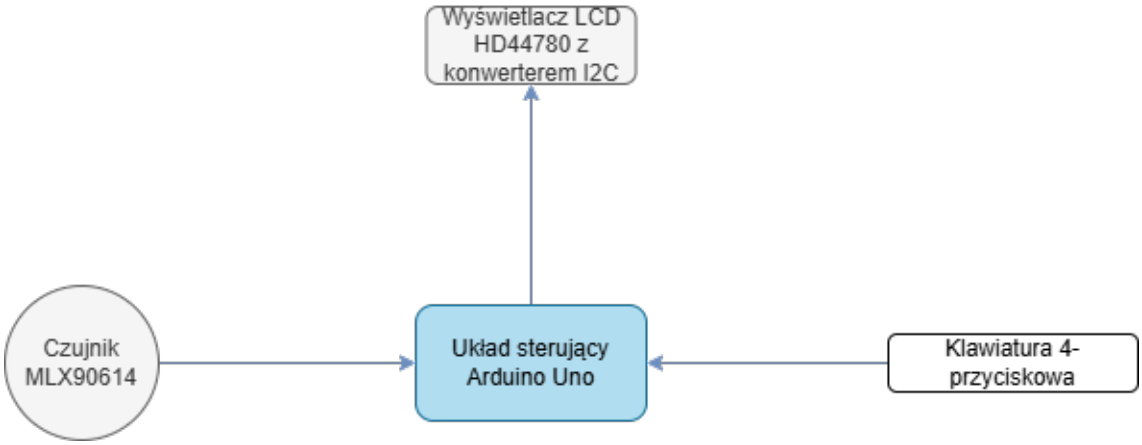
## Środowisko pracy

Urządzenie jest zdolne do pracy w szerokim zakresie temperatur od -20°C do +70°C, co pozwala na jego wykorzystanie w różnych warunkach otoczenia.

## Zasilanie

Zasilanie urządzenia odbywa się poprzez napięcie 5V, z opcją wykorzystania portu USB co najmniej w wersji 2.0. Urządzenie nie zostało przystosowane do pracy w ekstremalnych warunkach środowiskowych.

# 4. Opis części sprzętowej



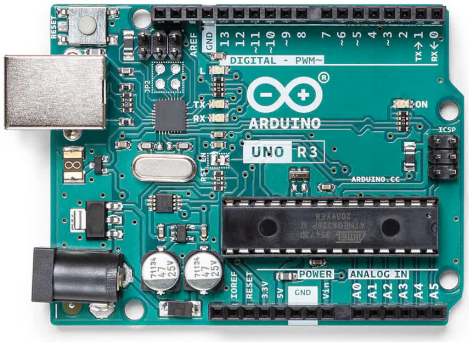

Rysunek 4.1: Schemat blokowy skonstruowanego urządzenia

## 4.1 Arduino Uno

Arduino Uno pokazany na rysunku 4.2, 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 [5].

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.




Rysunek 4.2: Mikrokontroler Arduino Uno [5]

**Zadanie:** Arduino Uno pełni funkcję jednostki centralnej systemu, integrując wszystkie kluczowe komponenty urządzenia. Jest odpowiedzialne za przetwarzanie danych pomiarowych z czujnika temperatury MLX90614, obsługę logiki systemu oraz sterowanie wyświetlaczem LCD. Dzięki temu użytkownik może w czasie rzeczywistym obserwować wyniki pomiarów i dokonywać ich konfiguracji za pomocą klawiatury. Arduino Uno zapewnia stabilne i niezawodne działanie, gwarantując synchronizację między komponentami. Ponadto, mikrokontroler umożliwia szybką reakcję na polecenia użytkownika, takie jak zmiana wartości emisyjności czy zmiana jednostek temperatury.

Dzięki swojej wszechstronności i otwartości Arduino Uno idealnie nadaje się do prototypowania urządzeń pomiarowych, takich jak projektowane w tym systemie rozwiązanie.

## 4.2 Czujnik MLX90614

Czujnik MLX90614 pokazany na rysunku 4.3, 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 mikrokontrolery. Dzięki swojej konstrukcji, MLX90614 znajduje zastosowanie w wielu dziedzinach, w tym w medycynie do pomiaru temperatury ciała, w systemach klimatyzacji oraz w automatyzacji przemysłowej [6].


Rysunek 4.3: Czujnik MLX90614 [6]

**Zadanie:** Czujnik MLX90614 stanowi kluczowy element systemu, umożliwiający precyzyjny, bezkontaktowy pomiar temperatury. Wykorzystuje technologię detekcji promieniowania podczerwonego, pozwalając na dokładny odczyt temperatury powierzchni bez konieczności fizycznego kontaktu z badanym obiektem.

## 4.3 Wyświetlacz LCD 16x2

Wyświetlacz LCD z konwerterem I2C pokazany na rysnuku 4.4, oparty jest 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), zamiast 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) [7]. 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.


Rysunek 4.4: Wyświetlacz LCD44780 z konwerterem I2C [7]

**Zadanie:** Wyświetlacz LCD 16x2 pełni funkcję interfejsu wizualnego, umożliwiając prezentację wyników pomiarów temperatury oraz dodatkowych informacji, takich jak ustawiona wartość emisyjności i wybrana jednostka temperatury (Celsjusz, Fahrenheit lub Kelvin).

## 4.4 4-przyciskowa klawiatura

Pokazana na zdjęciu 4.5 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.


Rysunek 4.5: 4-przyciskowa klawiatura [8]

Za pomocą dołączonej 4-przyciskowej klawiatury można:

- zwiększać wartość emisyjności,
- zmniejszać wartość emisyjności,
- przywrócić początkową wartość emisyjności wynoszącą 1,
- zmieniać jednostkę, w której wyświetlany jest wynik, wciskając kolejno przycisk (stopnie Celsjusza, Fahrenheita, Kelvina).

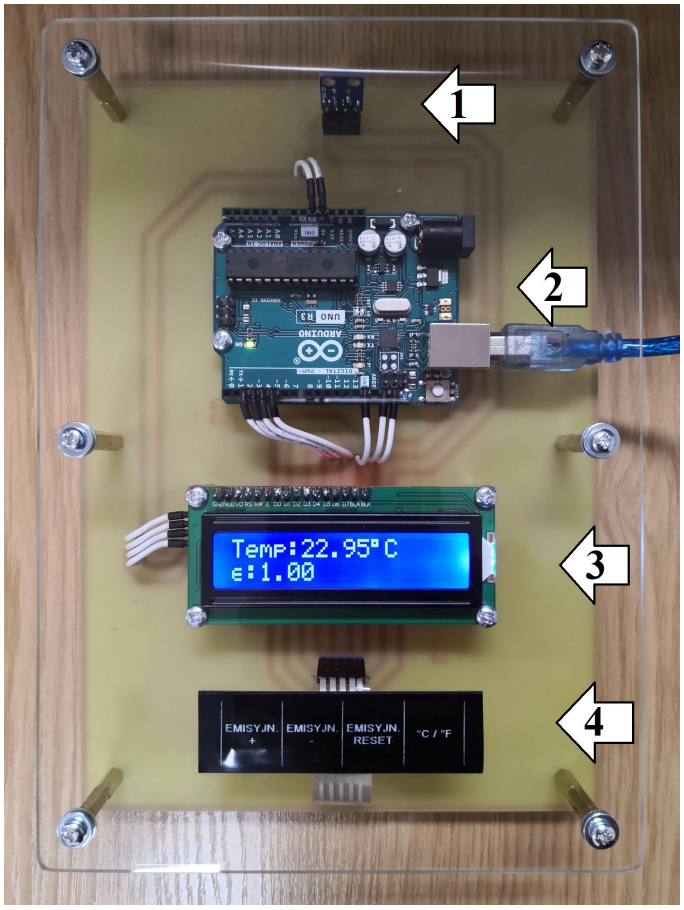
**Specyfikacja:** Prosty układ czterech przycisków, połączony z wejściami Arduino.

# Rzeczywisty układ

Na zdjęciu 4.6 przedstawiono rzeczywiste wykonanie urządzenia, którego schemat blokowy zaprezentowano na rysunku 4.1. Całość została zamontowana na płytce drukowanej i zabezpieczona przezroczystą osłoną, co zapewnia ochronę i wygodę obsługi.

Fotografia 4.6 przedstawia panel górny urządzenia. Najważniejsze jego elementy:

1. Czujnik MLX90614
2. Układ sterujący Arduino Uno
3. Wyświetlacz LCD HD44780 z konwerterem I2C
4. Klawiatura 4-przyciskowa

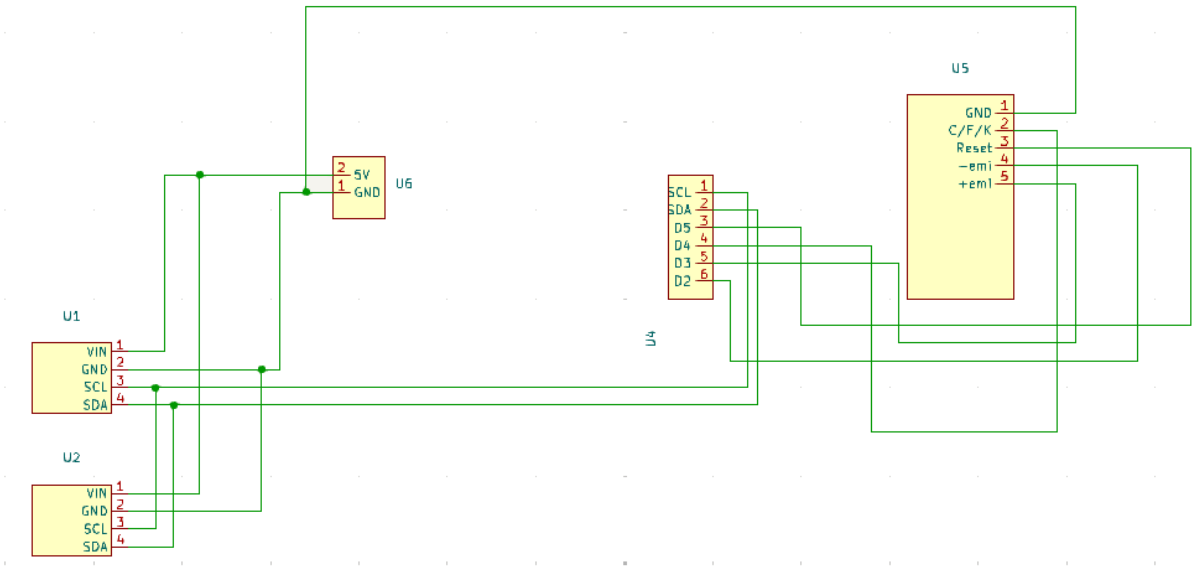



Rysunek 4.6: Panel górny urządzenia

# Część mechaniczna

W oparciu o schemat 4.1 zaprojektowano specjalnie dostosowaną płytkę PCB, której schemat ideowy widoczny jest na rysunku 4.7, natomiast układ ścieżek na płytce drukowanej można dostrzeć na rysunku 4.8. Łączy ona wszystkie podzespoły w jednej strukturze, gwarantując stabilność połączeń i zmniejszając ryzyko uszkodzeń obwodu. Dodatkowo pozwala na optymalne i przejrzyste rozmieszczenie elementów. Schemat przedstawiony na rysunku 4.7 ilustruje sposób połączenia kluczowych komponentów urządzenia.

- **Arduino Uno** oznaczone jako U5
- **Czujnik MLX90614** oznaczony jako U1
- **Wyświetlacz LCD z interfejsem I2C** oznaczony jako U2
- **Konwerter I2C dołączony do wyświetlacza** oznaczony jako U5
- **Klawiatura 4-przyciskowa** oznaczona jako U3

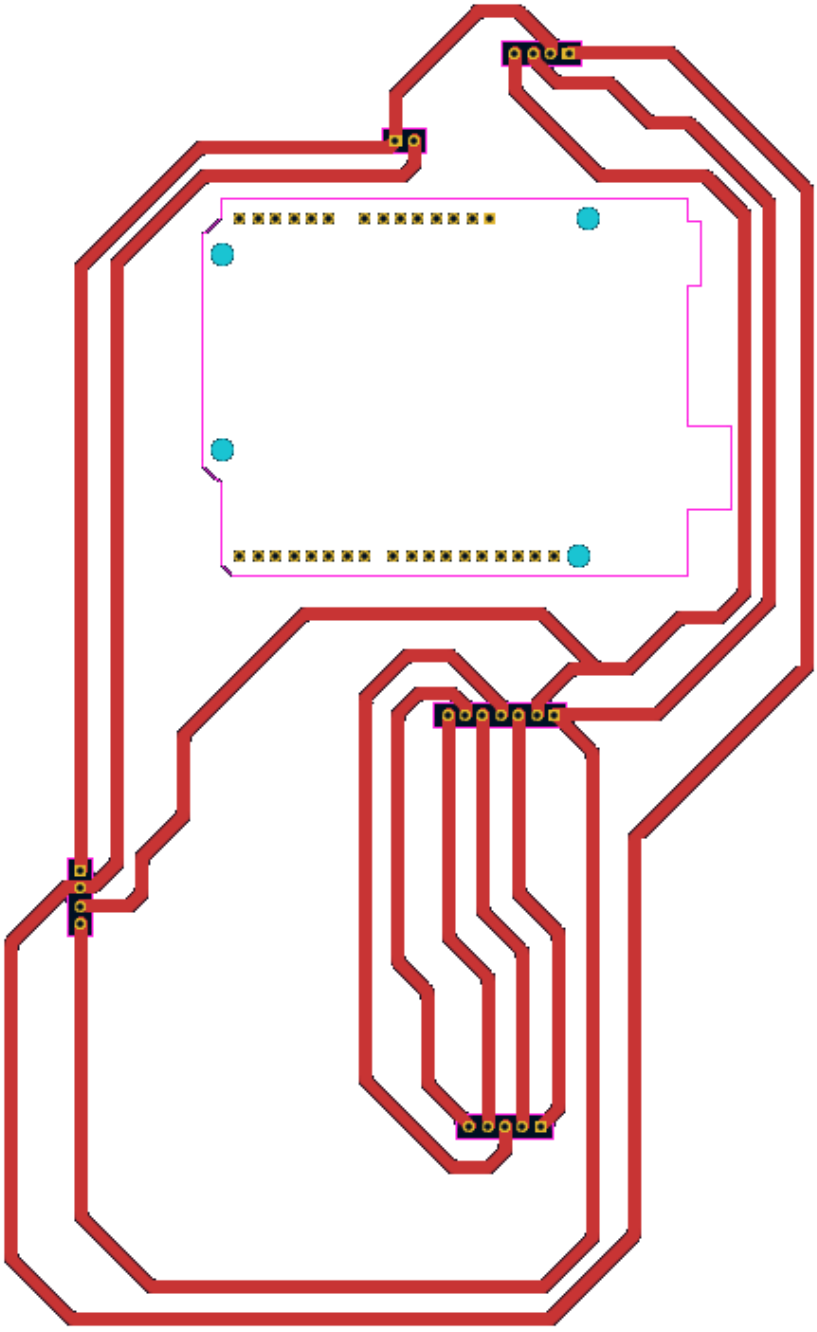



Rysunek 4.7: Schemat połączeń między komponentami

Połączenia między elementami opierają się głównie na magistrali I2C oraz standardowych liniach sygnałowych, co minimalizuje liczbę przewodów i upraszcza konstrukcję układu. Czujnik temperatury MLX90614 został podłączony do mikrokontrolera Arduino za pomocą interfejsu I2C. Wyświetlacz LCD HD44780 podłączono z wykorzystaniem konwertera pracującego na interfejsie I2C.

## Płytka PCB

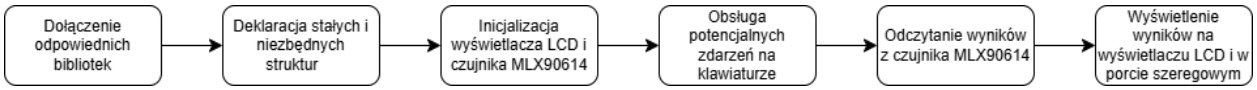
Po przetestowaniu komponentów na płytce prototypowej, zaprojektowano docelową płytę ewaluacyjną z wykorzystaniem oprogramowania KiCad 8.0. Wykonano układ ścieżek na płytce drukowanej oraz rozmieszczono złącza w odpowiednich miejscach. Płyta ewaluacyjna definiuje rozmiar urządzenia, wynoszący 208 mm x 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 4.8: Rozkład ścieżek na płytce drukowanej

# 5. Opis części programowej

Oprogramowanie pirometru zostało napisane w języku C++ z wykorzystaniem środowiska Arduino IDE. Program odpowiada za odczyt temperatury z czujnika MLX90614, wyświetlanie wyników na ekranie LCD 16x2 oraz obsługę 4-przyciskowej klawiatury do regulacji emisyjności/zmiany jednostki. Kod został zoptymalizowany pod kątem intuicyjnej obsługi i wysokiej dokładności pomiarów. Podstawowy algorytm działania przedstawiono na schemacie blokowym 5.1.




Rysunek 5.1: Schemat blokowy skonstruowanego urządzenia

## 5.1 Biblioteki i wstępna konfiguracja

Na początku kodu są załączone biblioteki:

`#include <Arduino.h>` – standardowa biblioteka Arduino.

`#include <Adafruit_MLX90614.h>` – obsługa czujnika MLX90614.

`#include <LiquidCrystal_I2C.h>` – sterowanie LCD przez I2C.

`#include <math.h>` – funkcje matematyczne (`pow()`, `sqrt()`).

## Definiowanie przycisków

Numery pinów dla czterech przycisków:

```
#define BTN1 2
#define BTN2 3
#define BTN3 4
#define BTN4 5
```

Przyciski podłączone do pinów cyfrowych 2–5.

## Konfiguracja wyświetlacza LCD

```
LiquidCrystalI2C lcd(0x27, 16, 2);
```

Adres I2C: `0x27`, wyświetlacz: 16 kolumn, 2 wiersze.

## Definiowanie znaku $\epsilon$

```
byte epsilon[8] = {
    B00000, B00000, B01110, B10000,
    B11110, B10000, B01110, B00000
};
```

Znak $\epsilon$ (8x5 pikseli) do wyświetlenia na LCD.

## Inicjalizacja czujnika MLX90614

```
Adafruit_MLX90614 mlx = Adafruit_MLX90614();
float ems = 1.0;
int tempScale = 0;
```

- `mlx` – obiekt czujnika MLX90614.
- `ems = 1.0` – emisyjność (domyślna wartość).
- `tempScale` – skala temperatury: 0 (°C), 1 (°F), 2 (K).

## 5.2 Połączenie z klawiaturą

### Odczyt stanów przycisków

```
int BTN1V = digitalRead(BTN1);
int BTN2V = digitalRead(BTN2);
int BTN3V = digitalRead(BTN3);
int BTN4V = digitalRead(BTN4);
```

Funkcja `digitalRead()` odczytuje stan każdego z przycisków (BTN1 do BTN4). Zwraca 0 (przycisk wciśnięty) lub 1 (przycisk nie wciśnięty). Zmienna `BTN1V`, `BTN2V`, `BTN3V` i `BTN4V` przechowują odpowiednio stany przycisków 1, 2, 3 i 4.

### Zwiększenie emisyjności (emissivity)

```
if (!BTN1V && ems < 1.0) {
    Serial.println("Increased emissivity");
    ems += 0.01;
    if (ems > 1.0) ems = 1.0;
}
```

Jeśli przycisk `BTN1` jest wciśnięty i emisyjność (`ems`) jest mniejsza niż 1.0, program:

- Wyświetla w monitorze szeregowym komunikat `"Increased emissivity"`.
- Zwiększa wartość emisyjności o 0.01.
- Upewnia się, że emisyjność nie przekroczy 1.0 (maksymalna dopuszczalna wartość).

### Zmniejszenie emisyjności

```
if (!BTN2V && ems > 0.0) {
    Serial.println("Decreased emissivity");
    ems -= 0.01;
    if (ems < 0.0) ems = 0.0;
}
```

Jeśli przycisk `BTN2` jest wciśnięty i emisyjność (`ems`) jest większa niż 0.0, program:

- Wyświetla w monitorze szeregowym komunikat `"Decreased emissivity"`.
- Zmniejsza wartość emisyjności o 0.01.
- Upewnia się, że emisyjność nie spadnie poniżej 0.0 (minimalna dopuszczalna wartość).

## Resetowanie emisyjności

```
if (!BTN3V) {
    Serial.println("Emissivity reset");
    ems = 1.0;
}
```

Jeśli przycisk `BTN3` jest wciśnięty, program:

- Wyświetla w monitorze szeregowym komunikat `"Emissivity reset"`.
- Ustawia wartość emisyjności na 1.0.

## Zmiana jednostki

```
if (!BTN4V) {
    tempScale = (tempScale + 1) % 3;
    Serial.println(tempScale == 0 ? "Switched to Celsius"
        :
```

```
4                         (tempScale == 1 ? "Switched to
                             Fahrenheit" :
5                         "Switched to Kelvin"));
6       delay(300);
7   }
```

Jeśli przycisk `BTN4` jest wciśnięty, program:

- Zmienia zmienną `tempScale`, aby cyklicznie przechodziła przez trzy wartości (Celsius, Fahrenheit, Kelvin). Zmienna `tempScale` jest inkrementowana, a następnie obliczany jest reszta z dzielenia przez 3, co daje efekt cyklicznego przełączania między 0, 1, 2.
- Wyświetla odpowiedni komunikat w monitorze szeregowym:
  - `"Switched to Celsius"` – jeśli `tempScale` wynosi 0.
  - `"Switched to Fahrenheit"` – jeśli `tempScale` wynosi 1.
  - `"Switched to Kelvin"` – jeśli `tempScale` wynosi 2.
- Funkcja `delay(300)` wprowadza opóźnienie 300 ms, aby przycisk nie wywołał wielu zmian w krótkim czasie.

## 5.3 Połączenie z czujnikiem temperatury MLX90614

### Odczyt temperatury

```
1 float ObjTemp = mlx.readObjectTempC();
2 float AmbientTemp = mlx.readAmbientTempC();
3 float correctedTemp = correctTemperature(ObjTemp,
     AmbientTemp, ems);
```

W tych liniach kodu odczytywane są dwie temperatury z czujnika MLX90614:

- `ObjTemp`: temperatura obiektu, którą czujnik odczytuje w stopniach Celsjusza.
- `AmbientTemp`: temperatura otoczenia, również odczytywana w stopniach Celsjusza.

Następnie obliczana jest `correctedTemp`, czyli poprawiona temperatura obiektu na podstawie zmierzonej temperatury obiektu, temperatury otoczenia i wartości emisyjności, wykorzystując funkcję `correctTemperature()`.

```
if (isnan(correctedTemp)) {
    Serial.println("Read error: Temperature NaN");
    correctedTemp = 0.0;
}
```

Jeśli funkcja `correctTemperature()` zwróci wartość "Not a Number"(NaN), co oznacza, że wystąpił błąd w obliczeniach, program wyświetla komunikat o błędzie na monitorze szeregowym, a zmienna `correctedTemp` jest ustawiana na 0.0.

## 5.4 Połączenie z wyświetlaczem LCD HD44780

## Inicjalizacja wyświetlacza LCD

```
lcd.init();
lcd.clear();
lcd.backlight();
lcd.setCursor(0, 0);
lcd.createChar(0, epsilon);
```

Po zainicjowaniu wyświetlacza za pomocą `lcd.init()`, wyświetlacz zostaje wyczyszczony (`lcd.clear()`), a podświetlenie zostaje włączone (`lcd.backlight()`). Kursor jest ustawiany na początek pierwszego wiersza (`lcd.setCursor(0, 0)`), a funkcja `lcd.createChar()` pozwala na zdefiniowanie niestandardowego znaku, np. symbolu epsilon, który może być użyty w wyświetlaczu.

```
lcd.setCursor(0, 0);
lcd.print("Temp: ");
lcd.print(displayTemp);
lcd.print(scaleLabel);
```

W tym fragmencie kodu tekst jest wyświetlany na LCD. Funkcja `lcd.setCursor(0, 0)` ustawia kursor na początku pierwszego wiersza, a następnie wyświetlany jest tekst "Temp: ", wartość temperatury oraz jednostka skali temperatury (°C, °F, K).

```
lcd.setCursor(0, 1);
lcd.write(byte(0));
lcd.print(" : ");
lcd.print(ems);
```

Drugi wiersz wyświetlacza pokazuje wartość emisyjności. Funkcja `lcd.write(byte(0))` wyświetla wcześniej zdefiniowany niestandardowy znak (np. epsilon), a następnie na ekranie pojawia się wartość emisyjności (`ems`).

# 6. Uruchomienie, kalibracja

Po podłączeniu urządzenia do zasilania za pomocą kabla USB, włączy się ono automatycznie. Czujnik nie wymaga przeprowadzenia kalibracji. W przypadku braku odczytów na wyświetlaczu, należy dostosować kontrast przy pomocy potencjometru umiejscowionego po lewej stronie ekranu. Wszelkie problemy z komunikacją z czujnikiem lub wyświetlaczem będą przekazywane przez interfejs szeregowy, którego transmisję można monitorować za pomocą programu Arduino IDE.

# 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

## 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.

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.

## 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.

# 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.

# 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.

# Bibliografia


[1] „Metody i urządzenia do pomiaru temperatury i ciśnienia” - Dr inż. Andrzej Kłabut

https://slideplayer.pl/slide/5307334/17/images/4/Metody+pomiaru+temperatury.jpg

[2] „Pirometr - poradnik użytkowania” - MERA-SP.PL

https://mera-sp.pl/blog/rozwiazania/pirometry-poradnik-uzytkowania

[3] „Rozkład energii w widmie promieniowania ciała doskonale czarnego w różnych temperaturach”

https://weblab.deusto.es/olarex/cd/kaernten/BBR_PLnew_27.09.2013/rozkad_energii_w_widmie_promieniowania_ciaa_doskonale_czarnego_w_rnych_temperaturach.html

[4] Korneliusz Jarzębski - „Pirometr z czujnikiem MLX90614ESF-BAA”

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

[5] Arduino - „Arduino Uno Rev3”

https://store.arduino.cc/products/arduino-uno-rev3/

[6] ElektroWeb - „Pirometr termometr bezdotykowy MLX90614 GY-906”

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

[7] Productos y Tecnología - „LCD 16X2 Fondo Azul/Verde + I2C”

https://marboltec.com/producto/lcd-16x2-fondo-azul-alta-calidad/

[8] AVT Sklep - „Klawiatura membranowa - 4 klawisze 1x4 - numeryczna - samoprzylepna do Arduino”

https://sklep.avt.pl/pl/products/klawiatura-membranowa-4-klawisze-1x4-numeryczna-samoprzylepna-do-arduino-185330.html

# Dodatki

## Pełny listing kodu źródłowego

```
#include <Arduino.h>
#include <Adafruit_MLX90614.h>
#include <LiquidCrystal_I2C.h>
#include <math.h>

// Define buttons
#define BTN1 2
#define BTN2 3
#define BTN3 4
#define BTN4 5

// Set I2C address for the LCD (change if needed, e.g., 0x27)
LiquidCrystal_I2C lcd(0x27, 16, 2); // 16 chars, 2 lines

// Custom epsilon character
byte epsilon[8] = { B00000, B00000, B01110, B10000, B11110, B10000, B01110, B00000 };

```

```
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);
```

# Program ćwiczenia dla studentów