



Politechnika Wrocławska

Katedra Metrologii Elektronicznej i Fotonicznej

Nazwa kursu:

Metrologia optyczna - laboratorium

Temat projektu:

Bezdotykowy pomiar temperatury za
pomocą pirometru opartym na
czujniku MLX90614

Autorzy projektu:

inż. Piotr Rosiński

inż. Patryk Niczke

inż. Przemysław Lis

Wydział Elektroniki, Fotoniki i Mikrosystemów

Kierunek: Elektronika

Miejsce i rok: Wrocław, 2024

Spis treści

1	Wstęp	3
1.1	Wprowadzenie	3
1.2	Cel projektu	3
1.3	Zakres projektu	4
2	Założenia projektowe	5
2.1	Opis założeń funkcjonalnych	5
2.2	Opis założeń konstrukcyjnych	5
2.3	Opis założeń środowiskowych	6
2.4	Opis założeń ekonomicznych	7
3	Charakterystyka wykorzystanych komponentów sprzętowych	9
3.1	Mikrokontroler Arduino Uno	9
3.2	Czujnik temperatury MLX90614	10
3.3	4-przyciskowa klawiatura	10
3.4	Wyświetlacz LCD z konwerterem I2C HD44780	11
4	Analiza struktury zastosowanego oprogramowania i sposobu połączenia komponentów	12
4.1	Połączenie z czujnikiem temperatury MLX90614	12
4.2	Połączenie z wyświetlaczem LCD HD44780	12
4.3	Synchroniczna współpraca LCD i czujnika temperatury z wykorzystaniem mikrokontrolera Arduino	13
4.4	Wykonanie płyty ewaluacyjnej oraz konstrukcja gotowego urządzenia naukowo-badawczego	13

5	Opis wzorów fizycznych	15
5.1	Wzory	16
6	Wykonanie testów i dokonanie odpowiednich pomiarów	17
6.1	Przebieg testów	17
6.2	Tabela porównawcza pomiarów	18
7	Instrukcja użytkowania	19
7.1	Krótki opis pirometru i jego przeznaczenia	19
7.2	Dane techniczne	20
7.3	Obsługa urządzenia	21
7.4	Opis budowy urządzenia	21
8	Podsumowanie i Wnioski	25
8.1	Potencjalne przyczyny błędów	26
	Bibliografia	32

Rozdział 1

Wstęp

1.1 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.

1.2 Cel projektu

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 płytki mikrokontrolerowa, Arduino UNO, która stanowi centrum sterujące całego urządzenia. Płytki Arduino UNO oparta jest na 8-bitowym mikrokontrolerze ATmega328P, który zapewnia róż-

norodne 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 powszechnie 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 linii kodu źródłowego:

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

1.3 Zakres projektu

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.

Rozdział 2

Założenia projektowe

2.1 Opis założeń funkcjonalnych

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

2.2 Opis założeń konstrukcyjnych

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 $0,5^{\circ}\text{C}$ w zakresie $0-50^{\circ}\text{C}$, 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

2.3 Opis założeń środowiskowych

Urządzenie działa w odpowiednio szerokim zakresie temperatur, od -20°C do $+70^{\circ}\text{C}$, 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.

2.4 Opis założeń ekonomicznych

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

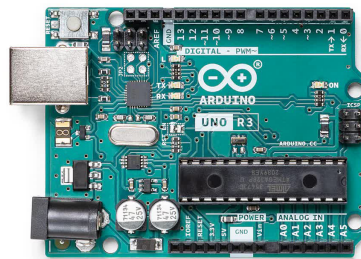
Całkowity koszt komponentów to około 220–28 zł, co czyni projekt relatywnie niedrogim i dość przystępnym cenowo w realizacji. Warto jednak zauważyć, że koszty projektu mogą ulec zmianie w zależności od wybranych dostawców i ilości zakupionych komponentów.

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

Rozdział 3

Charakterystyka wykorzystanych komponentów sprzętowych

3.1 Mikrokontroler Arduino Uno



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]

3.2 Czujnik temperatury MLX90614



Czujnik MLX90614 to zaawansowany, bezdotkowy 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 [5].

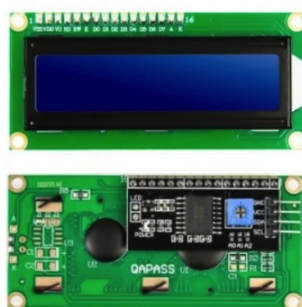
3.3 4-przyciskowa klawiatura

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.

3.4 Wyświetlacz LCD z konwerterem I2C HD44780



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), 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). 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.

Rozdział 4

Analiza struktury zastosowanego oprogramowania i sposobu połączenia komponentów

4.1 Połączenie z czujnikiem temperatury MLX90614

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.

4.2 Połączenie z wyświetlaczem LCD HD44780

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

4.3 Synchroniczna współpraca LCD i czujnika temperatury z wykorzystaniem mikrokontrolera Arduino

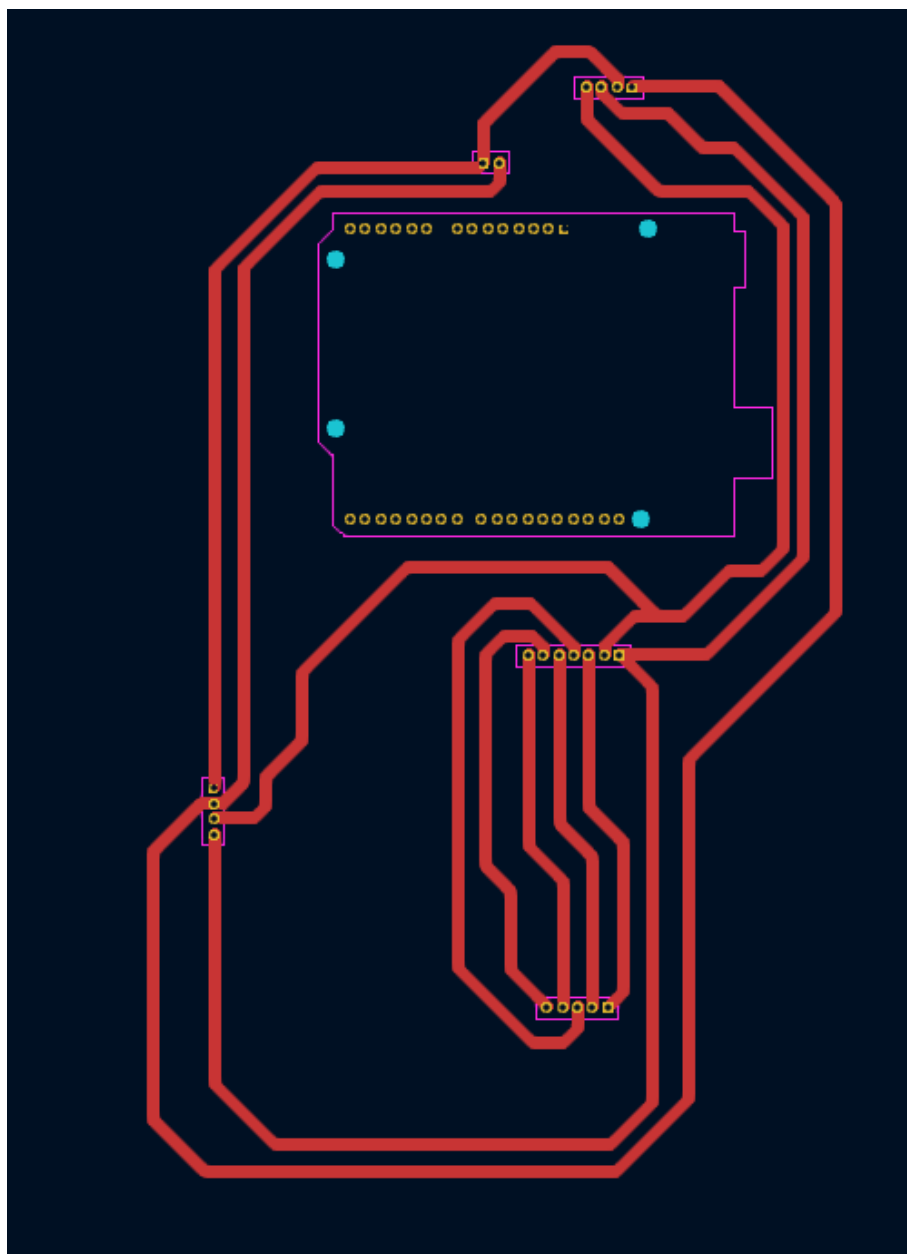
W celu synchronicznej współpracy wyświetlacza LCD i czujnika temperatury z mikrokontrolerem Arduino, został napisany program, który cyklicznie odczytuje temperaturę z czujnika i wyświetla ją na wyświetlaczu LCD. Program został napisany w języku C/C++ z wykorzystaniem bibliotek Wire.h i LiquidCrystal_I2C.h.

4.4 Wykonanie płyty ewaluacyjnej oraz konstrukcja gotowego urządzenia naukowo-badawczego

Po przetestowaniu komponentów na płytce prototypowej z wykorzystaniem oprogramowania KiCad 8.0 zaprojektowano układ ścieżek na płytce drukowanej.

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 4.1: Rozkład ścieżek na płycie drukowanej

Rozdział 5

Opis wzorów fizycznych

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).

5.1 Wzory

1. Moc promieniowania cieplnego emitowanego przez ciało:

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

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.

Rozdział 6

Wykonanie testów i dokonanie odpowiednich pomiarów

6.1 Przebieg testów

W celu weryfikacji dokładności 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. Obiekt pomiarowy jest płytą grzejącą o mocy 800W pomalowanego na kolor czarny matowy.

/vspace12pt

Procedura testowa:

1. Przygotowano stanowisko testowe składające się z urządzenia testowego, termometru stykowego oraz pirometru przemysłowego.
2. Dokonywano pomiarów w 5-stopniowych odstępach temperatury, zaczynając od 160°C i stopniowo schodząc do 35°C.
3. Każdy pomiar był wykonywany trzykrotnie, a wyniki uśredniano w celu minimalizacji błędów.

6.2 Tabela porównawcza pomiarów

Temperatura z Ardu- ino [°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 6.1: Porównanie pomiarów temperatury dla różnych urządzeń.



Rysunek 6.1: Przebieg procedury testowej

Rozdział 7

Instrukcja użytkowania

7.1 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.

7.2 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.

7.3 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.

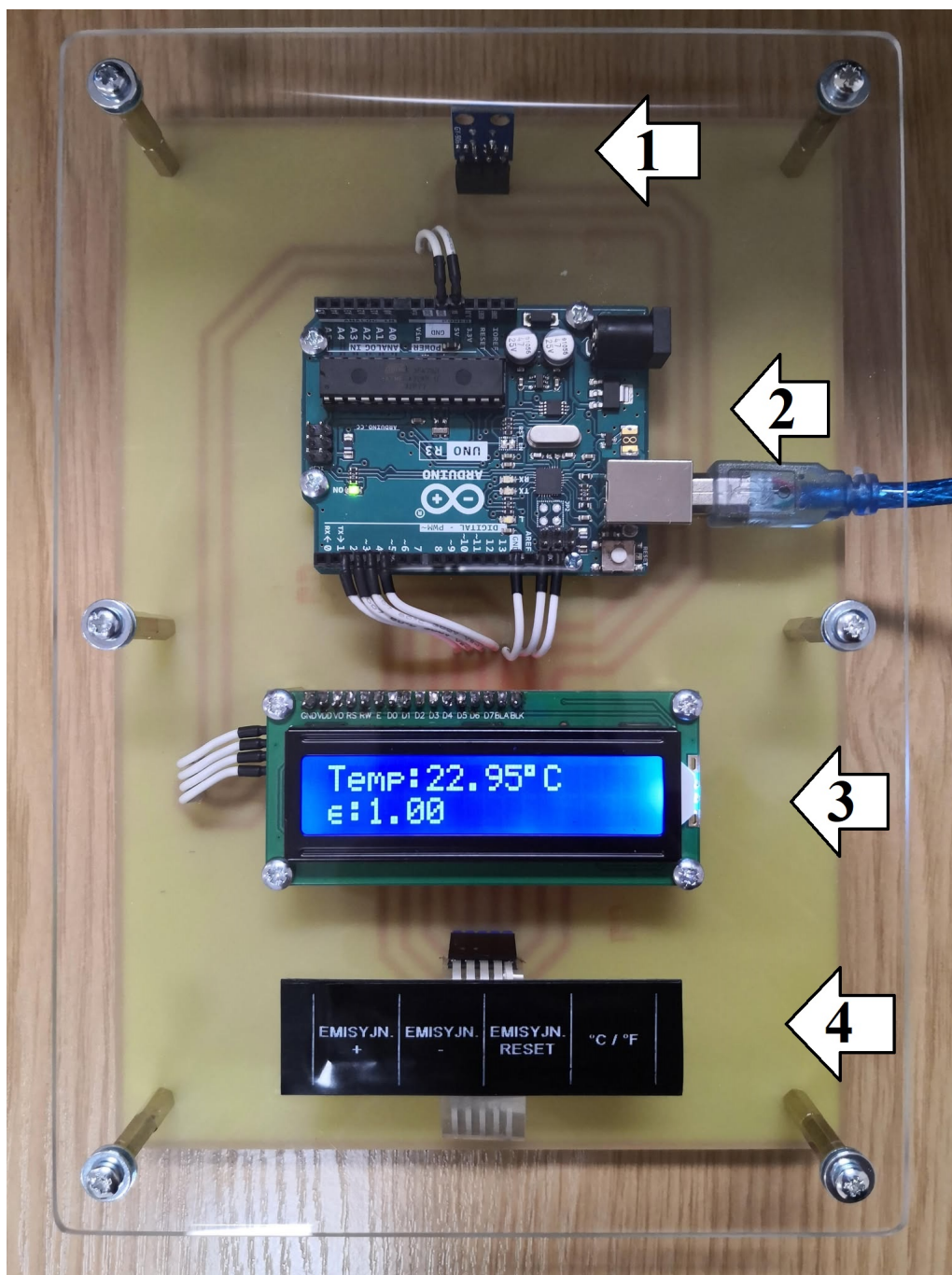
Ustawienie emisyjności w zakresie od 0.00 do 1.00 odbywa się za pomocą klawiszy EMISYJN. + oraz EMISYJN. -. Tabela emisyjności typowych materiałów znajduje się na rysunku 4. Prawidłowe ustawienie emisyjności ma istotny wpływ na dokładność pomiarów i powinno być dostosowywane przy każdej zmianie badanej 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.

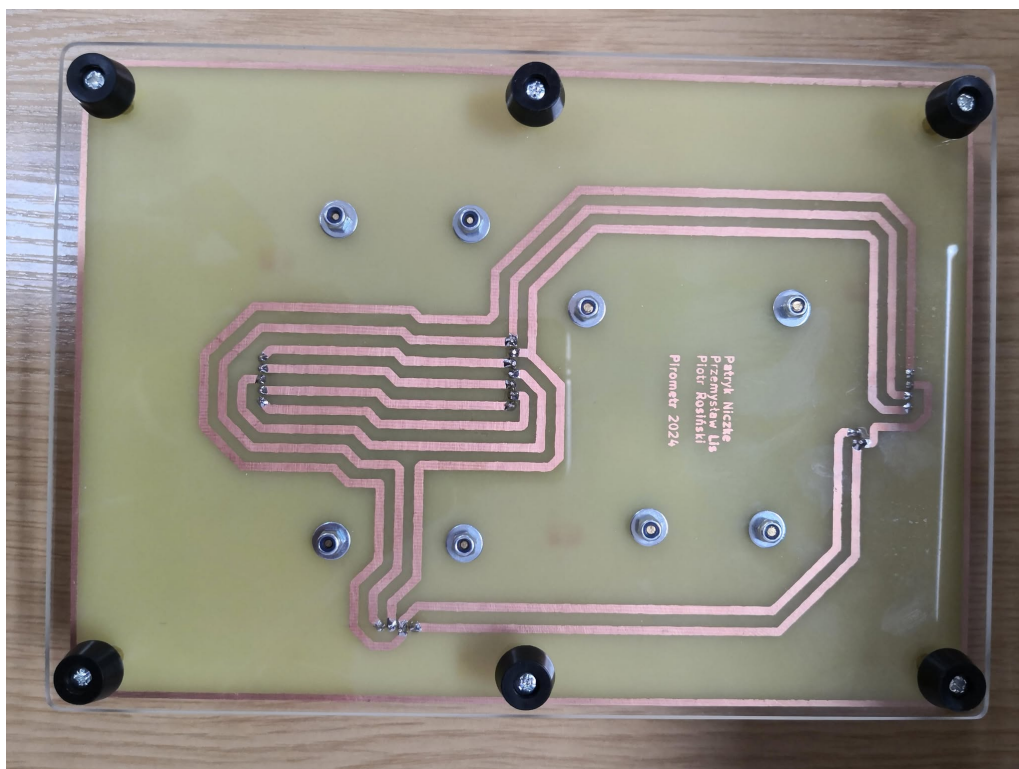
7.4 Opis budowy urządzenia

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.



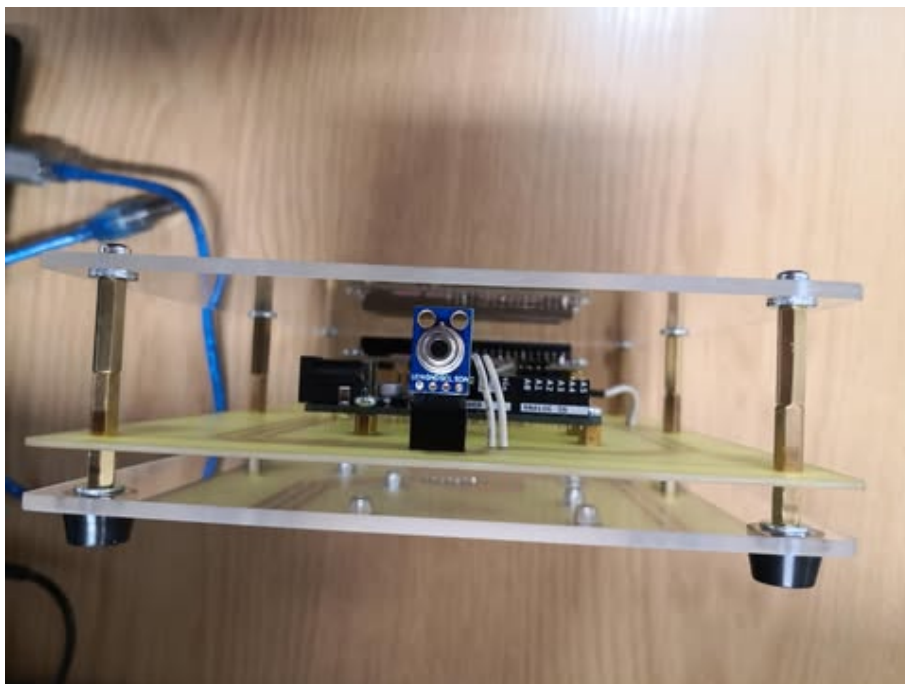
Rysunek 7.1: Panel górny urządzenia



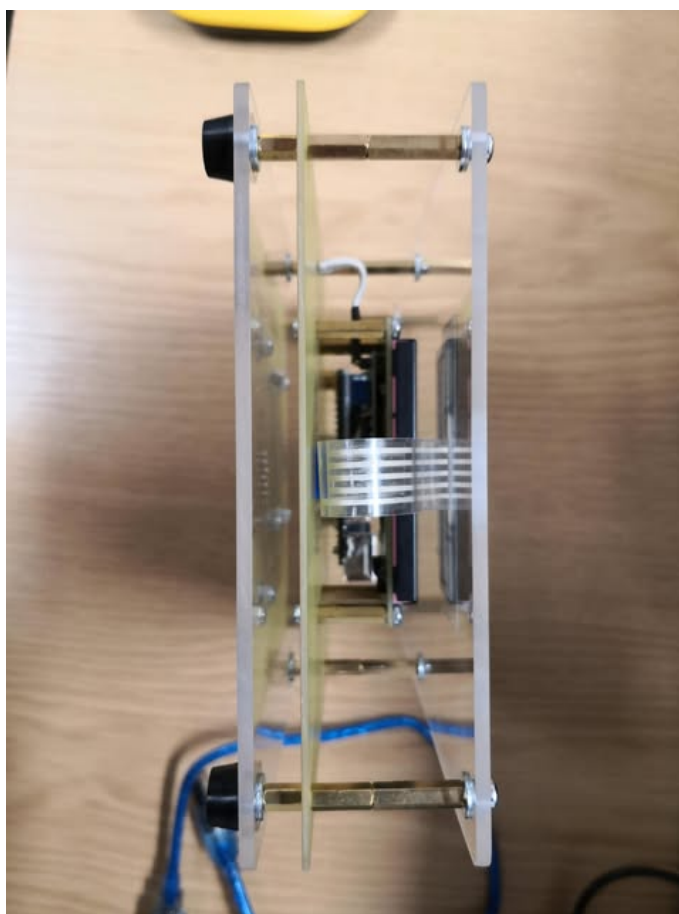
Rysunek 7.2: Panel tylny urządzenia



Rysunek 7.3: Widok z boku



Rysunek 7.4: Widok z boku



Rysunek 7.5: Widok z boku

Rozdział 8

Podsumowanie i Wnioski

Wyniki testów zostały przedstawione w tabeli 6.1. 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, błędami ustawień emisyjności lub ograniczoną rozdzielczością czujnika MLX90614.
- Termometr stykowy wykazuje większe różnice w niższych temperaturach, co może wynikać z bezwładności termicznej sondy stykowej.
- 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.
- Pirometr Arduino wykazuje dobrą powtarzalność wyników. Różnice w stosunku do termometru stykowego są niewielkie i pozostają względnie stałe w całym zakresie pomiarowym, co świadczy o stabilności działania czujnika MLX90614.

- Urządzenie testowe działa stabilnie w całym zakresie temperatur i wykazuje akceptowalny poziom zgodności z urządzeniami wzorcowymi.

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

8.1 Potencjalne przyczyny błędów

- **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
3  #include <Adafruit_MLX90614.h>
4
5  #include <LiquidCrystal_I2C.h>
6
7  #include <math.h>
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000

```

```

10     #define BTN4 5
11
12     // Set I2C address for the LCD (change if needed, e.g
13     //    ., 0x27)
14
15     LiquidCrystal_I2C lcd(0x27, 16, 2); // 16 chars, 2
16     //    lines
17
18     // Custom epsilon character
19
20     byte epsilon[8] = { B00000, B00000, B01110, B10000,
21     //    B11110, B10000, B01110, B00000 };
22
23
24     // Initialize MLX90614 sensor
25
26     Adafruit_MLX90614 mlx = Adafruit_MLX90614();
27
28     float ems = 1.0;
29
30     // Default emissivity
31
32     int tempScale = 0; // 0 - Celsius, 1 - Fahrenheit, 2
33     //    - Kelvin
34
35
36     float correctTemperature(float measuredTemp, float
37     //    ambientTemp, float emissivity) {
38
39         float measuredTempK = measuredTemp + 273.15;
40
41         float ambientTempK = ambientTemp + 273.15;
42
43         float trueTempK = pow((pow(measuredTempK, 4) - (1
44         //    - emissivity) * pow(ambientTempK, 4)) /
45         //    emissivity, 0.25);
46
47         return trueTempK - 273.15;
48     }
49
50
51     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         ;
90         correctedTemp = 0.0;
91     }
92
93     float displayTemp = correctedTemp;
94     char scaleLabel = 'C';
95     bool showDegreeSymbol = true;
96
97     if (tempScale == 1) {
98         displayTemp = correctedTemp * 9.0 / 5.0 +
99         32.0;
100        scaleLabel = 'F';
101    } else if (tempScale == 2) {
102        displayTemp = correctedTemp + 273.15;
103        scaleLabel = 'K';
104        showDegreeSymbol = false;
105    }
106
107    Serial.print("Temperature: ");
108    Serial.print(displayTemp);
109    Serial.print(" ");
110    Serial.print(scaleLabel);
111    Serial.print("\nEnvironment Temperature: ");
112    Serial.print(AmbientTemp);
113    Serial.print("\nEmissivity: ");
114    Serial.println(ems);

```

```
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);
```


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>