

<b>Imię i nazwisko</b>	<b>Maciej Filipiak</b>
<b>Temat</b>	<b>Interfejs użytkownika dla czujnika temperatury</b>

### *Funkcjonalność programu*

Projekt polegał na stworzeniu systemu do akwizycji i wizualizacji danych z czujnika temperatury podłączonego do Arduino. Wykorzystałem LabVIEW do komunikacji z mikrokontrolerem poprzez interfejs VISA, a także do przetwarzania i prezentacji danych w czasie rzeczywistym. Program umożliwia monitorowanie temperatury, sterowanie procesem pomiarowym oraz analizę zebranych danych.

### *Użyty sprzęt:*

- **Arduino Uno** – mikrokontroler zarządzający pomiarem temperatury
- **KA-Nucleo-Weather** – ekspander z czujnikiem temperatury
- **Komputer z LabVIEW** – odbierający i przetwarzający dane

### *Opis działania:*

1. **Nawiązanie komunikacji** – Użyłem VISA do otwarcia połączenia między komputerem a Arduino, umożliwiając przesyłanie danych pomiarowych.
2. **Sterowanie pomiarem** – W zależności od stanu logicznego przycisku Start/Stop, program sprawdzał, czy mikrokontroler przesłał nowe dane.
3. **Odczyt danych** – Pobierałem wartości temperatury za pomocą **VISA Read**, a następnie przetwarzałem je do odpowiedniego formatu.
4. **Przetwarzanie danych** – Wykorzystałem **Match Pattern** do ekstrakcji wartości liczbowych i konwersji z formatu string na DBL, umożliwiając dalsze operacje matematyczne.
5. **Akwizycja danych (Data Acquisition)** – Dane były gromadzone w czasie rzeczywistym i integrowane z systemem pomiarowym.
6. **Integracja hardware** – Połączenie Arduino z LabVIEW poprzez VISA pozwoliło na pełną synchronizację urządzenia i oprogramowania.
7. **Debugowanie i testowanie** – Podczas pracy nad projektem testowałem komunikację VISA oraz analizowałem poprawność przetwarzania danych, optymalizując działanie algorytmów.
8. **Zamykanie połączenia** – Po zakończeniu działania program czyścił bufor i zamykał połączenie z mikrokontrolerem.

### *Front Panel – interfejs użytkownika*

- **Okienko temperatury** – wyświetlało wartość liczbową temperatury w stopniach Celsjusza
- **Wykres temperatury** – przedstawiał zmiany temperatury w funkcji czasu
- **Termometr** – graficzna reprezentacja aktualnej temperatury
- **Dioda LED** – sygnalizowała aktywny proces pomiaru
- **Przycisk Start/Stop** – umożliwiał rozpoczęcie i zatrzymanie pomiaru
- **Przycisk End** – kończył działanie programu

### *Dodatkowe zastosowane funkcjonalności*

- **Shared Variables** – użyłem do współdzielenia danych pomiędzy różnymi procesami w LabVIEW
- **Obsługa sterowników (drivers)** – pobierałem i konfigurowałem sterowniki dla Arduino, aby umożliwić poprawną komunikację
- **Generatory sygnałów** – testowałem algorytmy, używając wirtualnych generatorów sygnałów w LabVIEW, aby symulować różne wartości temperatury

Dzięki zastosowaniu LabVIEW, system pozwalał na elastyczne zarządzanie pomiarami temperatury oraz łatwą analizę danych. Projekt był dobrą okazją do integracji sprzętu i oprogramowania oraz do nauki pracy z interfejsem VISA, akwizycją danych i debugowaniem systemów pomiarowych.