

Imię i nazwisko	Maciej Filipiak
Temat	Interfejs użytkownika dla czujnika temperatury

Funkcjonalność programu

Projekt polegał na stworzeniu modułowego, obiektowego systemu do akwizycji i wizualizacji danych z czujnika temperatury podłączonego do Arduino. LabVIEW zostało wykorzystane do komunikacji z mikrokontrolerem poprzez interfejs VISA oraz do przetwarzania i prezentacji danych w czasie rzeczywistym. Program umożliwia monitorowanie temperatury, sterowanie parametrami pomiarowymi, analizę trendów oraz eksport wyników.

Użyty sprzęt:

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

Architektura systemu:

Program został zaprojektowany w sposób modułowy, opierając się na architekturze obiektowej LabVIEW. Zastosowano warstwę abstrakcji sprzętowej (HAL), co umożliwia łatwe zarządzanie sterownikami i aktualizację sprzętu. Wymiana danych między modułami odbywa się za pomocą współdzielonych zmiennych (Shared Variables), co pozwala na rozszerzalność systemu i dystrybucję danych w wielu procesach.

Opis działania:

1. **Nawiązanie komunikacji** – Użycie VISA do otwarcia połączenia między komputerem a Arduino, umożliwiając przesyłanie danych pomiarowych poprzez interfejs USB.
2. **Sterowanie pomiarem** – Program kontroluje proces pomiarowy w zależności od stanu logicznego przycisku Start/Stop oraz parametrów użytkownika.
3. **Odczyt danych** – Pobieranie wartości temperatury za pomocą VISA Read oraz ich konwersja do odpowiedniego formatu.
4. **Przetwarzanie danych** – Użycie Match Pattern do ekstrakcji wartości liczbowych, konwersja formatu string na DBL, umożliwiając operacje matematyczne i analizę trendów.
5. **Akwizycja i logowanie danych** – System zbiera i zapisuje pomiary w czasie rzeczywistym z dodanymi znacznikami czasu (timestamping).
6. **Integracja sprzętu** – Pełna synchronizacja Arduino z LabVIEW poprzez VISA, ułatwiająca testowanie i debugowanie.
7. **Automatyczne zarządzanie błędami** – Implementacja narzędzi diagnostycznych do walidacji systemu i wychwytywania błędów komunikacyjnych.
8. **Zamykanie połączenia** – Po zakończeniu działania program czyści bufor i zamyka połączenie z mikrokontrolerem.

Interfejs użytkownika (Front Panel):

- **Okienko temperatury** – wyświetla wartość liczbową temperatury w stopniach Celsjusza.
- **Wykres temperatury** – prezentuje zmiany temperatury w funkcji czasu.
- **Termometr** – graficzna reprezentacja aktualnej temperatury.

- **Dioda LED** – sygnalizuje aktywny proces pomiaru.
- **Przyciski sterujące** – Start/Stop do zarządzania pomiarem oraz End do zakończenia programu.
- **Opcje konfiguracji** – możliwość ustawienia parametrów pomiarowych, takich jak częstotliwość próbkowania.
- **Eksport danych** – możliwość zapisania wyników w formacie CSV do dalszej analizy.

Dodatkowe funkcjonalności:

- **Współdzielone zmienne (Shared Variables)** – zapewniają płynną wymianę danych między modułami i procesami w LabVIEW.
- **Obsługa sterowników (Drivers)** – integracja odpowiednich sterowników do poprawnej komunikacji z Arduino.
- **Generatory sygnałów** – wirtualne źródła sygnałów pozwalające na testowanie algorytmów w warunkach symulacyjnych.
- **Skalowalna architektura** – możliwość łatwej rozbudowy systemu i integracji z nowymi komponentami sprzętowymi.

Dzięki zastosowaniu LabVIEW, system pozwalał na elastyczne zarządzanie pomiarami temperatury, analizę danych w czasie rzeczywistym oraz łatwą rozbudowę. Projekt stanowił cenną okazję do integracji sprzętu i oprogramowania, a także do nauki pracy z interfejsem VISA, akwizycją danych, debugowaniem systemów pomiarowych oraz tworzeniem obiektowych aplikacji w LabVIEW.