

Politechnika Wrocławska, Katedra Inżynierii Biomedycznej

Systemy Pomiarowo-Diagnostyczne, laboratorium

Ćwiczenie 2 – Obsługa urządzeń pomiarowych z interfejsem szeregowym (VISA)

1. Cel ćwiczenia

Zapoznanie się z metodami obsługi urządzeń wyposażonych w interfejs szeregowy.

Nabywanie umiejętności realizacji urządzenia wirtualnego wykorzystującego multimetr oraz standard VISA.

Zapoznanie się z metodami tworzenia własnych modułów (podprogramów) w środowisku LabView.

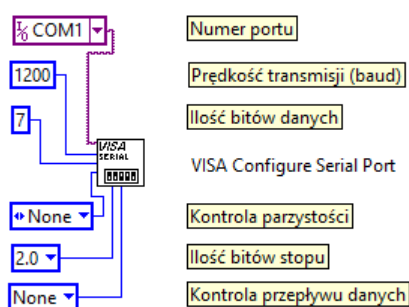
2. Wprowadzenie do ćwiczenia

a) Interfejs komunikacyjny

Do realizacji ćwiczenia zostanie wykorzystany multimetr z interfejsem szeregowym zgodnym ze standardem RS232. W celu nawiązania połączenia z urządzeniem należy skorzystać z dedykowanego przewodu dostarczonego przez producenta, zwracając uwagę na sposób podłączenia wtyku przewodu do multimetru (część urządzeń dostępnych na stanowiskach nie posiada fizycznego zabezpieczenia przed odwrotnym podłączeniem wtyku). Aby nawiązać połączenie z multimetrem konieczne jest skonfigurowanie parametrów interfejsu szeregowego w komputerze zgodnie z danymi zawartymi w instrukcji urządzenia:

- prędkość transmisji: 1200 baud
- ilość bitów danych w jednostce informacyjnej: 7
- ilość bitów stopu: 2
- brak kontroli parzystości i przepływu danych

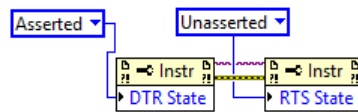
W środowisku Labview do skonfigurowania portu szeregowego, obsługi transmisji danych oraz wykrywania błędów wykorzystamy standard Virtual Instrument Software Architecture (VISA). Za konfigurację i rezerwację zasobów sprzętowych odpowiada moduł: *VISA Configure Serial Port* do którego należy doprowadzić informacje określające parametry transmisji. Przykładową konfigurację prezentuje rysunek 1.



Rysunek 1. Konfiguracja parametrów transmisji

Multimetry wykorzystywane na zajęciach laboratoryjnych wyposażone są w separator linii sygnałowych, który wymaga dwustronnego niezależnego zasilania. Ze względu na brak sprzętowej kontroli przepływu danych do tego celu można wykorzystać linie DTR i RTS (*Data Terminal Ready* i *Request to Send*). Niezależnie od wykorzystywanego środowiska programistycznego do prawidłowej pracy separatora wymagane jest nadanie tym liniom odpowiedniego stanu: DTR logiczny stan wysoki, RTS logiczny stan niski. W środowisku LabView dodatkowe parametry i stany interfejsu można odczytywać i nastawiać przy użyciu węzłów właściwości (*property node*) przypisanych do linii łączącej poszczególne bloki

standardu VISA (*VISA resource name*). Za kontrolę stanu linii DTR i RTS odpowiadają właściwości: *Serial Settings* -> *Modem Line Settings* -> *Line DTR State* oraz *Serial Settings* -> *Modem Line Settings* -> *Line RTS State*. Proponowany sposób wprowadzenia tych parametrów przedstawia rysunek 2.



Rysunek 2. Sterowanie stanem linii DTR i RTS (zasilanie separatora)

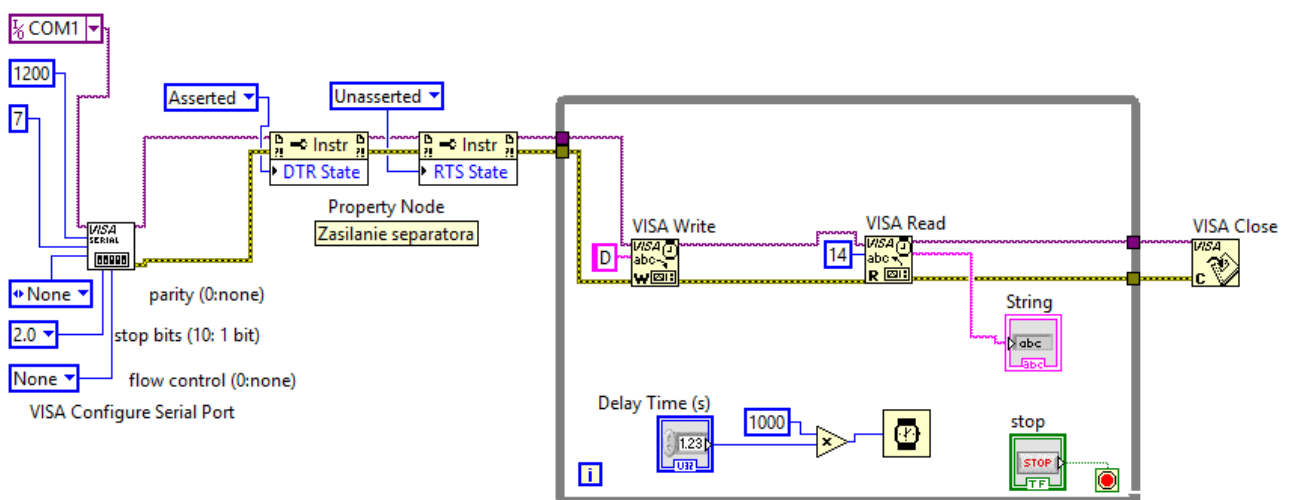
b) Obsługa multimetru

W tym ćwiczeniu do obsługi nadawania i odbioru danych za pośrednictwem skonfigurowanego interfejsu korzystamy z elementów: *VISA Write* i *VISA Read*. Informacje przesyłane są w postaci tekstowej jako dane typu *string*. W zależności od typu multimetru dostępne są różne polecenia jakie może realizować urządzenie. Podstawowe polecenie przesłania aktualnego wyniku wraz z jednostką realizowane jest po odebraniu przez multimetr znaku „D”. Odpowiedź z multimetrów dostępnych na zajęciach ma zawsze stałą długość i wynosi ona 14 znaków (wolne miejsca w odpowiedzi dopełniane są znakami spacji). Wynik przesyłany jest wraz z oznaczeniem jednostki zależnym od aktualnego trybu pracy urządzenia pomiarowego i zakończony jest znakiem CR (*carriage return*), przykładowe odpowiedzi wyglądają następująco:

- DC-1.9999 V [CR]
- 1.9999Mohm[CR]

Aby możliwe było wykonywanie dalszych obliczeń z wykorzystaniem wyniku zawartego w odpowiedzi należy przeprowadzić analizę otrzymanej wartości typu *string*. Do tego celu można wykorzystać bloki konwersji typów danych (np. *Fract/Exp String To Number*, wymaga wydzielenia danych liczbowych) lub bloki analizy i konwersji (np. *Scan from String*, wymaga określenia formatu poszukiwanych danych). Po przeprowadzeniu konwersji wyniku z postaci tekstowej na liczbową należy go przemnożyć przez współczynnik zgodny z jednostką zawarta w odpowiedzi (np. jeśli jednostką jest kOhm to wynik jest mnożony przez 1000).

Propozycja wykorzystania omówionych powyżej elementów standardu VISA przedstawiono na rysunku 3.



Rysunek 3. Propozycja realizacji podstawowej komunikacji z multimetrem

Uwaga: W programach korzystających z komunikacji z urządzeniami zewnętrznymi nie należy przerywać pracy programu poprzez wciśnięcie przycisku *Abort execution* znajdującego się na górnej listwie poleceń. Program powinien być zakończony przyciskiem umieszczonym na panelu operatora, a w diagramie powinna być przewidziana procedura

zakończenia komunikacji i zwolnienia zasobów sprzętowych (w systemie VISA tę funkcję spełnia blok *VISA Close* widoczny po prawej stronie na rysunku 3).

c) *Rezystancyjne czujniki temperatury*

Urządzenie opracowywane w ramach ćwiczenia będzie pełniło funkcję wirtualnego termometru z możliwością obsługi dowolnego rezystancyjnego czujnika temperatury. Po przeprowadzeniu wstępnej analizy danych otrzymanych z multimetru program będzie wykonywał konwersję wyniku pomiaru rezystancji na temperaturę zgodnie z opcjami wybranymi na panelu operatora.

W ramach ćwiczenia przewidziano możliwość obsługi dwóch przykładowych czujników tj. PT100 oraz KTY81-210. Czujniki różnią się obszarem zastosowań, metodą przetwarzania i charakterystyką. Założono, że obliczenia dotyczące przetwarzania wyników pomiarów rezystancji na temperaturę będą wykonane dwiema metodami. Czujnik PT100 będzie wykorzystany jako przykład czujnika o charakterystyce opisanej zależnością matematyczną, natomiast czujnik KTY81-210 będzie przykładem czujnika, dla którego producent udostępnia tabelę z listą rezystancji i odpowiadających im temperatur.

Zależność dla czujnika PT100 jest podana w treści zadania, natomiast dla czujnika KTY należy skorzystać z informacji zawartych w nocie katalogowej (tabela dla czujnika KTY81-210, zależność pomiędzy temperaturą w stopniach Celsjusza a typową rezystancją).

Dla czujnika KTY sugerowane jest dopasowanie krzywej logarytmicznej w dowolnym programie zewnętrznym (np. Excel) i wprowadzenie uzyskanej zależności do diagramu programu w LabView. Osoby chętne mogą przeprowadzić proces dopasowania krzywej w środowisku LabView korzystając z funkcji z grupy *Mathematics->Fitting*.

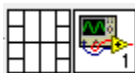
d) *Podprogramy – tworzenie własnych modułów (subVI)*

Konwersja wyników pomiarów zgodnie z założeniami zadania ma być wykonana w formie modułu (podprogramu *SubVI*). Moduły w środowisku LabView możemy tworzyć samodzielnie lub skorzystać z automatycznego tworzenia podprogramów. Samodzielne tworzenie daje nam większą kontrolę nad procesem powstawania podprogramu i terminali wejścia/wyjścia lecz jest bardziej czasochłonne.

Z metody automatycznej warto skorzystać przypadku prostych podprogramów, polega ona na zaznaczeniu wybranego fragmentu diagramu i zamianie na pojedynczy moduł z automatycznym utworzeniem terminali (po zaznaczeniu fragmentu wybieramy opcję *Edit->Create SubVI*). Jeśli chcemy od początku zdecydować o sposobie wyprowadzenia terminali lub wolimy tworzyć podprogramy od podstaw (a nie poprzez oddzielenie fragmentu większej całości) to możemy dowolny program zapisany z rozszerzeniem *vi* potraktować jako podprogram (moduł). Użycie zewnętrznego programu *vi* jako podprogramu odbywa się poprzez wybranie opcji *Select a VI...* z palety funkcji diagramu.

Aby możliwe było wprowadzanie i wyprowadzanie danych z modułu konieczne jest przypisanie kontrolerom i wskaźnikom znajdującym się na panelu opowiadających im terminali. Układ terminali i przypisania można dowolnie zmieniać korzystając z edytora połączeń znajdującego się w prawym górnym rogu okna służącego do edycji panelu operatora LabView (lewa strona rysunku 4). Przypisanie odbywa się poprzez kliknięcie na pole edytora i wybrany kontroler lub wskaźnik (terminal zmieni kolor na odpowiadający kontrolerowi/wskaźnikowi typ danych).

Możliwa jest też zmiana ikony reprezentującej moduł. Można to zrobić klikając na ikonę znajdującą się obok edytora terminali (prawa strona rysunku 4)



Rysunek 4. Edytor terminali i ikony modułu

Należy pamiętać że podprogramy zapisywane są jako dodatkowe pliki *vi*. Każdorazowe uaktualnienie podprogramu wymaga zapisania zmian, które automatycznie będą widoczne z poziomu programu nadrzędnego.

3. Założenia do zadania

- 1) Komunikacja z multimetrem [3 pkt]:
 - a) Program ma realizować dwukierunkową komunikację z multimetrem za pośrednictwem interfejsu szeregowego oraz funkcji VISA
 - b) Komunikacja ma być poprawnie zainicjalizowana po uruchomieniu programu oraz zakończona przed zatrzymaniem programu
 - c) Błędy w komunikacji mają być sygnalizowane na panelu użytkownika (*timeout*)
 - d) Program ma obsługiwać wszystkie zakresy pomiaru rezystancji, jakie udostępnia multimetr
 - e) Wynik poza zakresem ma być sygnalizowany na panelu i nie ma powodować zgłoszenia błędu w programie
 - f) Odebrany wynik pomiaru rezystancji ma być zamieniony na wartość wyrażoną w Ohmach (typ *Double*)
- 2) Konwersja wyników pomiaru rezystancji (wykonany jako SubVI):

- a) Przyrząd ma realizować funkcję przeliczenia rezystancji na temperaturę dla PT100 oraz KTY81-210 [3 pkt]
 - i) Dla PT100 skorzystać ze współczynników: $A = 3.9083 \cdot 10^{-3} \text{ } ^\circ\text{C}^{-1}$, $B = -5.775 \cdot 10^{-7} \text{ } ^\circ\text{C}^{-2}$, $R_0 = 100 \text{ } \Omega$ oraz zależności (tylko dla $T > 0^\circ\text{C}$, gdzie r jest zmierzoną wartością rezystancji):

$$T_{RTD}(r) = \frac{-A + \sqrt{A^2 - 4B\left(1 - \frac{r}{R_0}\right)}}{2B}$$

- ii) Dla KTY81-210:

Przeprowadzić dopasowanie krzywej do typowych wartości rezystancji dla temperatur z zakresu od -50 do 50°C (skorzystać z tabeli umieszczonej w nocie katalogowej oraz z arkusza kalkulacyjnego lub funkcji dopasowania dostępnych w LabView). Wyznaczone równanie krzywej wprowadzić do programu i wykorzystać do przeliczania rezystancji na temperaturę
- b) Konwerter wartości rezystancji na temperaturę ma być wykonany w formie modułu SubVI [3 pkt]
 - i) Wejścia:
 - rezystancja (typ DBL)
 - rodzaj czujnika (typ U8), $0 \rightarrow \text{Pt100}$, $1 \rightarrow \text{KTY81}$
 - precyzja wyniku konwersji (typ U8) – zaokrąglenie wykonane w diagramie SubVI
 - skala temperatury (typ BOOL): true – Celsjusza, false - Fahrenheita
 - ii) Wyjścia:
 - temperatura (typ DBL) – wyznaczona według zasad opisanych w pkt. 2a i zaokrąglona zgodnie z parametrem wejściowym określającym precyzję
 - nieprawidłowa konwersja (typ BOOL) – sygnalizowana gdy wynik dla PT100 jest mniejszy niż 0°C oraz dla KTY81 gdy wynik jest mniejszy od -50°C lub większy od 50°C

- 3) Obsługa wirtualnego urządzenia [3 pkt]
 - a) Udostępnić możliwość sterowania wszystkimi parametrami wejściowymi konwertera (poza rezystancją)
 - b) Prezentacja wyników:
 - i) Operator może wybrać czy program ma prezentować odczytaną z multimetru wartość rezystancji czy wyliczonej temperatury (jednostka wyniku powinna się zmieniać)
 - ii) Najnowszy wynik ma być prezentowany w polu odczytowym
 - iii) Historia 100 ostatnich wyników ma być prezentowana na wykresie
 - iv) Opis osi wykresu powinien odpowiadać aktualnemu trybowi pracy – temperatura lub rezystancja
 - v) Wykres jest czyszczony po zmianie trybu
 - c) Jeśli wyświetlana jest temperatura i konwerter zwrócił błąd to wynik nie jest widoczny w historii wyników
 - d) Nieprawidłową konwersję sygnalizować obiektem LED (sygnalizacja niezależna od błędu komunikacji)
 - e) Częstość odczytu i konwersji danych ma być regulowana w zakresie od 500ms do 10s
 - f) Po uruchomieniu programu przyjmowane są domyślnie wartości nastaw
 - g) Panel jest zorganizowany w czytelny i ergonomiczny sposób

6-6,5 pkt: 3 7-8 pkt: 3,5 8,5-9,5 pkt: 4 10-11 pkt: 4,5 11,5-12 pkt: 5