

# Politechnika Wrocławska, Katedra Inżynierii Biomedycznej

## Systemy Pomiarowo-Diagnostyczne, laboratorium

### Ćwiczenie 4 – Pomiar zdalny

#### 1. Cel ćwiczenia

Zapoznanie się z metodami organizacji transmisji danych w systemach pomiarowych.

Nabywanie umiejętności opracowania protokołu komunikacyjnego dla urządzeń pomiarowych pracujących w konfiguracji *single-master/multi-slave* oraz zastosowanie opracowanego protokołu do wykonania zdalnych pomiarów.

Utrwalenie wiedzy i umiejętności z zakresu obsługi urządzeń pomiarowych w środowisku LabView.

#### 2. Wprowadzenie do ćwiczenia

Ćwiczenie przeprowadzone jest na dwóch stanowiskach komputerowych. Jedno ze stanowisk pełni rolę nadrzędną, za pomocą aplikacji przygotowanej dla tego stanowiska operator zarządza procesem pomiarowym. Drugie stanowisko jest wyposażone w urządzenie pomiarowe (karta pomiarowa) i jego aplikacja umożliwia zdalny dostęp do zasobów sprzętowych tego stanowiska.

Założono, że system ma umożliwiać dwukierunkową komunikację dzięki której będzie można: realizować funkcje zdalnego odczytu pojedynczych wyników oraz serii wyników pomiarowych, dokonywać zmiany nastawy przetwornika cyfrowo-analogowego oraz przysyłać krótkie komunikaty tekstowe. Aplikacja ma być opracowana niezależnie dla urządzenia nadrzędnego i podrzędnego. Pomimo, że ćwiczenie jest realizowane na dwóch stanowiskach założono, że oprogramowanie ma umożliwiać rozbudowę systemu o dodatkowe urządzenia podrzędne. System ma pracować w konfiguracji *single-master/multi-slave* (jedno urządzenie nadrzędne/kontroler i wiele urządzeń podrzędnych).

#### 3. Zasady komunikacji

Realizacja komunikacji szeregowej pomiędzy urządzeniami wymaga ustalenia podstawowych zasad transmisji danych. W tym ćwiczeniu między stanowiskami komputerowymi będą przesyłane dane tekstowe (*string*) za pośrednictwem interfejsu RS232 lub połączenia sieciowego z wykorzystaniem protokołu TCP/IP. Dla uproszczenia analizy przesyłanych danych założono, że będą one przesyłane jako znaki zgodne z kodowaniem ASCII (dotyczy to także wyników i liczb).

Aby móc analizować odebrane dane oraz przeprowadzać podstawową weryfikację poprawności transmisji informacje powinny być przesyłane w ustalonej kolejności i formacie. Dane łączone są w zestawy zwane ramkami komunikacyjnymi według ściśle określonych reguł. W ramach tego ćwiczenia należy zgodnie z założeniami zadania opracować własny zestaw reguł, które będą stanowiły podstawę prostego protokołu komunikacyjnego.

Jako inspirację do opracowania własnego protokołu można wykorzystać powszechnie używany protokół MODBUS (tryb ASCII). W tym protokole transmisja jest zawsze inicjowana przez urządzenie nadrzędne. Ogólny schemat budowy ramki komunikacyjnej tego protokołu przedstawiony jest na rysunku 1.

Start	Adres	Kod funkcji	Dane	Suma (LRC)	Koniec
:			...		CR LF

Rysunek 1. Schemat ramki komunikacyjnej Modbus ASCII

#### Adresowanie:

Każde pole przedstawione w dolnym wierszu tabeli z rysunku 1 to jeden symbol z tablicy ASCII. Ramka rozpoczyna się znakiem specjalnym, po którym przesyłany jest adres urządzenia docelowego. Adresowanie umożliwia użycie większej

liczby urządzeń w ramach jednej sieci komunikacyjnej. Ramka interpretowana jest tylko wtedy gdy adres odbiornika i adres zamieszczony w ramce są zgodne. Opracowując własne zasady tworzenia ramki można przyjąć jako znak początkowy dowolny znak z podstawowej tablicy ASCII. Dla uproszczenia należy przyjąć że adres urządzenia jest jednocyfrowy i zostanie zapisany jako jeden znak z zakresu 0-9. W konfiguracjach sieciowych z jednym niezmiennym urządzeniem nadrzędnym zazwyczaj przypisuje się temu urządzeniu jeden z góry określony adres. W protokole Modbus przesyłany adres jest adresem urządzenia podrzędnego niezależnie od kierunku transmisji danych (czyli niezależnie od tego czy dane są wysłane przez urządzenie nadrzędne czy podrzędne). Podczas realizacji ćwiczenia należy przyjąć jednak inne podejście, w którym przesyłany adres jest adresem urządzenia docelowego, a następnie należy wykorzystać ten adres do sprawdzenia czy urządzenie powinno interpretować polecenia zawarte w ramce (należy sprawdzić zgodność adresu odebranego z adresem wybranym dla urządzenia, dotyczy to także urządzenia nadrzędnego).

#### *Funkcje i potwierdzenia:*

Kolejne pole ramki powinno zawierać kod funkcji, dla uproszczenia można przyjąć że funkcje reprezentowane są jednym symbolem ASCII. Lista funkcji jest podana w założeniach zadania. Są to funkcje umożliwiające przestanie pojedynczego wyniku pomiaru, serii wyników pomiaru, nastawienia przetwornika DAC lub przesłania krótkiej informacji tekstowej. Pierwsze dwie funkcje pełnią rolę zapytań, przesłane w odpowiedzi wyniki mogą być wykorzystane jako potwierdzenie odbioru polecenia. Kolejne dwie funkcje pełnią rolę nastaw i dla uzyskania potwierdzenia odebrania tych nastaw należy odesłać ramkę potwierdzającą zawierającą znak początkowy, adres urządzenia nadrzędnego oraz dowolny wybrany ciąg znaków który będzie interpretowany jako potwierdzenie odbioru nastawy.

#### *Dane:*

Pole danych zależne jest od wykorzystywanej funkcji. Dla uproszczenia można przyjąć, że ma ono stałą długość dla odpowiedzi zawierających wyniki lub nastawę napięcia dla DAC. Odpowiedź zawierającą wyniki serii pomiarowej (wyniki pomiaru napięcia na 4 wejściach analogowych karty) należy przesłać jako jeden ciąg znaków w polu danych i przed prezentacją na panelu urządzenia nadrzędnego podzielić go na 4 niezależne wyniki. Wyniki serii można wysyłać jako zestaw znaków o ustalonej długości lub/i oddzielać wyniki separatorem. W przypadku wiadomości tekstowej (ostatnia funkcja) długość pola danych nie jest określona, ma być ona jednak nie dłuższa niż 90 znaków.

#### *Suma:*

W protokole należy też zaimplementować proste zabezpieczenie pozwalające sprawdzić czy ramka została przesłana w całości lub nie została połączona z inną ramką komunikacyjną. Założono, że test ten będzie znacznie prostszy niż w standardowych protokołach i będzie polegał na podliczeniu ilości znaków w ramce. Ilość znaków jest dopisana za polem danych, dla uproszczenia przyjęto że jest ona zapisana zawsze za pomocą dwóch znaków i reprezentuje ilość znaków w całej ramce (łącznie z polem sumy).

#### *Zakończenie ramki:*

W ćwiczeniu założono, że po sumie ilości znaków nie będzie dodatkowego pola kończącego ramkę.

#### *Wykorzystanie biblioteki VISA:*

Jeśli w trakcie realizacji ćwiczenia będzie wykorzystana biblioteka VISA i interfejs RS232 można skorzystać z nieużywanego we wcześniejszych ćwiczeniach węzła właściwości, który udostępni informację o ilości znaków gotowych do odebrania przez funkcję *VISA Read*. W tym celu należy kliknąć prawym przyciskiem myszy na linii *VISA resource name* (górną linią łączącą funkcje VISA) i wybrać opcję *Create->Property for Instr Class->Serial settings->Number of bytes at serial port*. Wykorzystanie informacji o ilości bajtów gotowych do odebrania jest konieczne ze względu na zmienną długość ramki komunikacyjnej oraz może okazać się przydatne przy wykrywaniu błędów komunikacyjnych. Przy analizie błędów komunikacji pomocna może się okazać także opcja *timeout* ustawiana podczas konfiguracji interfejsu w funkcji *VISA Serial*. Należy także pamiętać, że do prawidłowej komunikacji z użyciem interfejsu RS232 niezbędne jest ustalenie takich samych parametrów transmisji na obu stanowiskach.

#### 4. Założenia do zadania

- 1) Komunikacja odbywa się między dwiema aplikacjami komputerowymi uruchomionymi na dwóch stanowiskach komputerowych, jedno z nich pełni rolę nadrzędną (*Master*) a drugie podrzędną (*Slave*). Sterowanie procesem pomiarowym odbywa się tylko za pośrednictwem aplikacji nadrzędnej, aplikacja podrzędna umożliwia natomiast zdalny dostęp do urządzenia pomiarowego.
- 2) Ramka komunikacyjna ma zawierać następujące pola:
  - a) Znak początkowy (wybrać samodzielnie)
  - b) Adres urządzenia docelowego (w zakresie od 0 do 9, gdzie 0 jest zarezerwowane dla *Master*)
  - c) Kod/symbol funkcji (opisane poniżej, kody/symbole wybrać samodzielnie)
  - d) Pole danych (o długości i zawartości zależnej od typu funkcji)
  - e) Ilość znaków w całej ramce (dla kontroli poprawności transmisji, przyjmując że to pole zawiera zawsze dwa znaki)
- 3) Funkcje protokołu [2,5 pkt za każdą funkcję]:
  - a) Zapytanie o pojedynczy wynik pomiaru (wejście AIN3 urządzenia *Slave*)
  - b) Zapytanie o serię wyników pomiarów (wejścia AIN0-AIN3 urządzenia *Slave*, wysyłane jako jeden *string*)
  - c) Wysłanie pojedynczej nastawy napięcia (dla przetwornika *DACO* urządzenia *Slave*, z potwierdzeniem odbioru)
  - d) Przesłanie komunikatu tekstowego do wybranego urządzenia *Slave* (do 90 znaków, z potwierdzeniem odbioru)
- 4) Właściwości protokołu:
  - a) Protokół ma obsługiwać tryb ASCII
  - b) Komunikacja jest inicjalizowana zawsze przez urządzenie *Master*
  - c) Urządzenie *Master* ponawia zapytanie/polecenie po odczekaniu 1 sekundy w przypadku gdy: [2pkt]
    - i) Urządzenie *Slave* nie odpowiada (timeout)
    - ii) Ramka odpowiedzi ma nieprawidłową długość
    - iii) Ramka odpowiedzi ma zły znak początkowy
  - d) Ponawianie zapytania/polecenia jest sygnalizowane na panelu urządzenia *Master*
  - e) Odebrana ramka jest interpretowana pod warunkiem, że adres zawarty w ramce i adres urządzenia są zgodne
- 5) Ćwiczenie należy zrealizować korzystając z 2 stanowisk komputerowych połączonych interfejsem szeregowym lub wykorzystać sieć komputerową i protokół TCP. Jedno stanowisko pełni funkcję urządzenia *Master*, drugie urządzenia *Slave*:
  - a) Stanowisko *Slave*:
    - i) Ma możliwość zmiany swojego adresu (adresy 1-9)
    - ii) W zależności od wybranej funkcji:
      - funkcja a i b: po otrzymaniu zapytania odbiera wyniki pomiarów z karty pomiarowej (wejście AIN3 lub wejścia AIN0-AIN3), prezentuje je na pulpicie i wysyła odpowiedź z wynikami
      - funkcja c: ustawia odebraną wartość na wyjściu *DACO*, wyświetla ją na pulpicie i wysyła potwierdzenie
      - funkcja d: wyświetla komunikaty tekstowe przysłane przez urządzenie *Master* i wysyła potwierdzenie
    - iii) Prezentuje odebraną i wysłaną ramkę oraz sygnalizuje odbywającą się transmisję
    - iv) Sygnalizuje błąd w przypadku odebrania nieprawidłowej ramki
  - b) Stanowisko *Master*:
    - i) Umożliwia wysyłanie zapytań, nastaw i komunikatów do urządzenia *Slave* zgodnie z listą funkcji
    - ii) Pozwala na zmianę adresu urządzenia *Slave*, z którym odbywa się aktualnie komunikacja (1-9)
    - iii) Sygnalizuje odbywającą się komunikację na panelu
    - iv) Prezentuje odebraną i wysłaną ramkę
    - v) Sygnalizuje błąd w przypadku odebrania nieprawidłowej ramki
    - vi) Prezentuje odebrane wyniki pomiarów na panelu użytkownika (funkcja a i b)

Skala ocen (punkty : ocena)

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

Założenia bez przypisanej punktacji są wymagane do zaliczenia ćwiczenia.