|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| logoKM | logoKM | | POLITECHNIKA ŚLĄSKA  WYDZIAŁ ELEKTRYCZNY  KATEDRA MECHATRONIKI | logoKM |
| Instrukcja do ćwiczenia laboratoryjnego | | | | |
| Przedmiot: | | Przetwarzanie i Wizualizacja Danych Pomiarowych | |  |
| Symbol ćwiczenia: | | PiWDP1 | |  |
| Tytuł ćwiczenia: | | **Wprowadzenie do akwizycji danych w środowisku LabVIEW** | | |

**SPIS TREŚCI**

|  |  |  |
| --- | --- | --- |
|  | Spis rysunków | 2 |
| 1. | Cele ćwiczenia | 3 |
| 2. | Podstawowe wiadomości | 3 |
| 3. | Laboratoryjne stanowisko badawcze | 5 |
| 3.1. | Obiekt badany | 5 |
| 3.2. | Urządzenia dodatkowe | 5 |
| 3.3. | Oprogramowanie | 5 |
| 4. | Program ćwiczenia - wykaz zadań do realizacji | 6 |
| 5. | Przykład realizacji zadania – programowanie określonej trajektorii ruchu | 6 |
| 6. | Raport | 6 |
| 7. | Pytania | 7 |
|  | Literatura | 7 |

**SPIS RYSUNKÓW**

|  |  |  |
| --- | --- | --- |
| 1. |  |  |
| 2. |  |  |
| 3. |  |  |
| 4. |  |  |
| 5. |  |  |
| 6. |  |  |
| 7. |  |  |
| 8. |  |  |
| 9. |  |  |

* program narzędziowy MAX
* testowanie karty DAQ w MAX,
* tworzenie wirtualnej karty
* testowanie wirtualnej karty DAQ w MAX,
* metody konfiguracji zadania pomiarowego (TASK) i skali
* w MAX
* podstawy programowania w LabVIEW
* pojęcia Front panel i diagram code
* paleta kontrolek i paleta funkcji
* Kontrolki i wskaźniki
* skrót klawiszowy Ctrl+E
* konfiguracja pomiaru za pomocą DAQ assistant

podgląd biblioteki DAQmx (wyszukanie stałej skali)

**1. CELE ĆWICZENIA**

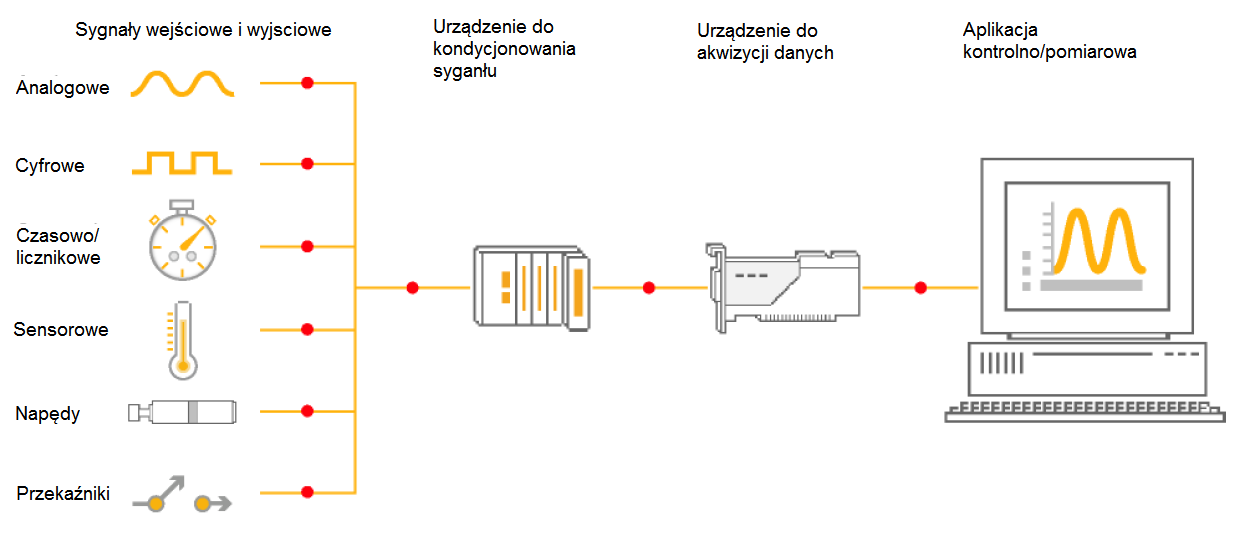
Celem ćwiczenia jest zapoznanie się z następującymi zagadnieniami:

* Zapoznanie się z obsługą programu narzędziowego NI MAX (Measurement & Automation Explorer),
* Sprawdzenie działania karty DAQ w programie NI MAX,
* Zapoznanie się z możliwościami symulacji kart pomiarowych w NI MAX,
* Zapoznanie się z podstawowymi pojęciami związanymi z programowaniem w LabVIEW,
* Zapoznanie się z podstawowymi skrótami klawiaturowymi przyspieszającymi pracę w LabVIEW,
* Metody tworzenia zadań (task) w LabVIEW.

**2. PODSTAWOWE WIADOMOŚCI**

LabVIEW jest graficznym środowiskiem programistycznym języka G. Nazwa języka pochodzi od wyrazu **Graphic** (grafika/rysunek). Pisanie programów w środowisku LabVIEW odbywa się na zasadzie łączenia kolejnych węzłów **nodes** (bloków kodu) za pomocą przewodów **wires**.

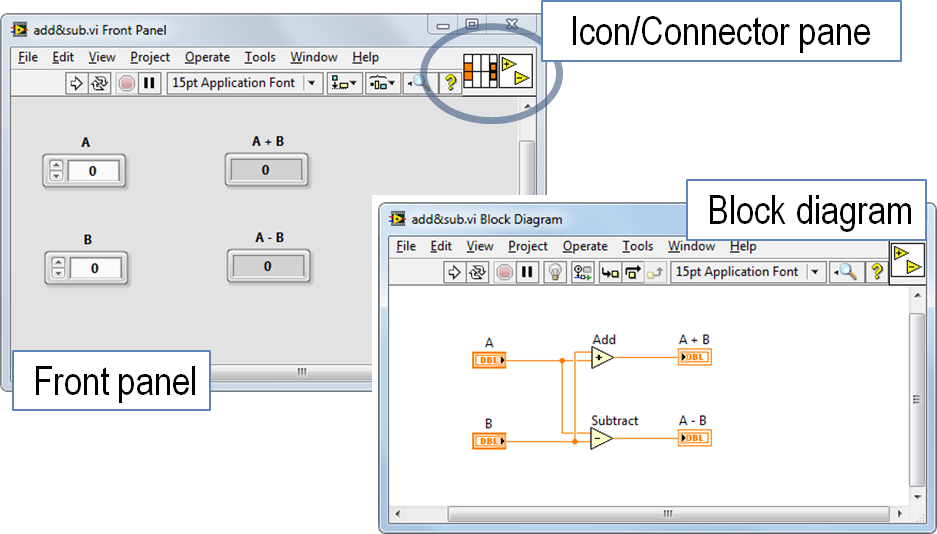
Podstawową ideą działania środowiska LabVIEW jest pojęcie instrumentu wirtualnego (**Virtual Instrument**), który użytkownik klasycznych przyrządów pomiarowych będzie w stanie obsłużyć bez dodatkowego, specjalistycznego przeszkolenia, bazując na doświadczeniu obsługi klasycznych przyrządów pomiarowych. Idea ta ma także odzwierciedlenie w rozszerzeniu plików środowiska LabVIEW: *vi* – *virtual instrument*. Struktura przyrządu wirtualnego została przedstawiona na Rys. 1.



1. Struktura wirtualnego przyrządu pomiarowego.

Struktura podstawowego pliku – vi składa się z trzech składników przedstawionych na Rys. 2:

* Panel (**Front Panel**) – jest interfejsem użytkownika, umieszcza się na nim kontrolki i wskaźniki, do których będzie miał dostęp użytkownik programu.
* Diagram (**Block Diagram**) – zawiera kod programu – cała funkcjonalność pisanego programu jest zaimplementowana w postaci diagramu znajdującego się w tym oknie. Poza kontrolkami i wskaźnikami w oknie diagramu znajdują się stałe oraz węzły, połączenia między nimi realizowane są za pomocą przewodów.
* Ikona i panel połączeń (**Icon** and **Connector Pane**), struktura programu w LabVIEW jest hierarchiczna – każdy program VI może być użyty jako podprogram (**subVI**) w bardziej ogólnym programie VI. Elementem identyfikującym subVI w diagramie nadrzędnego VI jest ikona. Natomiast strukturę wejść i wyjść wprowadzających dane do i z subVI określa struktura panelu połączeń.



1. Struktura pliku VI.

Elementami GUI do wymiany informacji między użytkownikiem a programem są kontrolki (**Controls**) i wskaźniki (**Indicators**) umieszczone na panelu. Kontrolki wprowadzają informacje do programu, wskaźniki prezentują użytkownikowi wyniki działania programu. Wszystkie kontrolki i wskaźniki dostępne w LabVIEW znajdują się w palecie kontrolek (**Controls Palete**) wyświetlanej prawym przyciskiem myszy (*PPM*) w oknie panelu.

Odpowiednio użycie PPM w oknie diagramu otworzy paletę funkcji (**Functions Palete**) zawierającą wszystkie dostępne funkcje. Paleta funkcji oraz paleta kontrolek podzielone są na zakładki. Przykłady palet funkcji i kontrolek przedstawiono na.

|  |  |
| --- | --- |
| a)  loc_env_controls_palette.bmp | b)  loc_env_functions_palette.bmp |
| 1. Widok palety kontrolek (a) oraz palety funkcji (b). | |

* podstawy programowania w LabVIEW
* skrót klawiszowy Ctrl+E
* konfiguracja pomiaru za pomocą DAQ assistant

**3. LABORATORYJNE STANOWISKO BADAWCZE**

**3.1. Obiekt badany**

.

**3.2. Urządzenia dodatkowe**

Brak

**3.3. Oprogramowanie**

* LabVIEW 2013 lub nowszy

**4. PROGRAM ĆWICZENIA – WYKAZ ZADAŃ DO REALIZACJI**

Ćwiczenie 1:

* Zapoznanie się z elementami składowymi stanowiska robota.

**5. PRZYKŁAD REALIZACJI ZADANIA**

Szablon programu poruszania napędami manipulatora,

**6. RAPORT**

Raport z przeprowadzonego ćwiczenia laboratoryjnego powinien zawierać:

* Krótki opis robota.

**7. PYTANIA**

1. Wymienić elementy struktury i omówić funkcje przyrządu wirtualnego.

2. Porównać funkcjonalność klasycznych i wirtualnych przyrządów pomiarowych.

3. Omówić podstawowe elementy oraz funkcjonalność struktury pliku VI .

4. Czy jest wymagane tworzenie dedykowanej ikony dla każdego VI?

**LITERATURA**

1. Honczarenko J.: *Roboty przemysłowe. Budowa i zastosowanie.* Wydawnictwa Naukowo-Techniczne, Warszawa 2004.

Opracowanie: Marek Kciuk

**ZAŁĄCZNIKI**

1. **Zastosowanie sprzętowej transmisji – pętla sprzężenia zwrotnego**

Prosty program wykorzystujący sprzętowy moduł transmisji szeregowej (UART) do realizacji pętli sprzężenia zwrotnego sterownik SSC-32 -> mikrokontroler.

Pętla sprawdzająca wysyła zapytanie („Q”) do sterownika czy zakończył ostatni ruch, odpowiedź sterownika zapisana jest do zmiennej ans („+” – ruch w trakcie realizacji, „ . ” – zakończono wszystkie ruchy.

ans var byte ‘deklaracja zmiennej sprawdzającej

ans = "."

enablehserial ‘konfiguracja I uruchomienie sprzętowego UARTa

sethserial H38400, H8DATABITS, HNOPARITY, H1STOPBITS

main: ‘program główny sterujący napędem #16

gosub wait\_1

hserout ["#16P", dec 800, $D]

gosub wait\_1

hserout ["#16P", dec 2200,"S", dec 200, $D]

goto main

wait\_1: ‘pętla sprawdzająca zakończenie ruchu

hserout ["Q",$D]

hserin [ans]

pause 5

debug [ans,$D]

if ans="+" then goto wait\_1

pause 22

return

1. **Program scalonego detektora kolorów**

'ustawienie wyprowadzen

Out con 6

S3 con 0

S2 con 7

S1 con 2

S0 con 3

LED con 1

SSC con 15

'deklaracja stalych

pRED con 15

pGREEN con 12

pBLUE con 9

mSPEED con 400

'deklaracja zmiennych

RED var word

GREEN var word

BLUE var word

'Inicjalizacja

High S1

High S0

'program glowny

Main:

Gosub Color\_Detect

Gosub Color\_Show

goto Main

Color\_Detect: 'Detekcja trzech skladowych koloru

High LED

pause 5

Low S2

Low S3

count Out, pRED, RED

High S3

count Out, pBLUE, BLUE

High S2

count Out, pGREEN, GREEN

pause 5

Low LED

RETURN

Color\_Show: 'Wyswietla watrosci koloru w trybie pracy "DEBUG"

debug[" r: ", dec RED]

debug[" g: ", dec GREEN]

debug[" b: ", dec BLUE]

gosub New\_Line

RETURN

New\_Line: 'Przechodzi do nowej lini w trybie "DEBUG"

debug[$D]

RETURN

1. **Dane techniczne serwonapędów**

HS-475HB

|  |  |  |  |
| --- | --- | --- | --- |
| |  | | --- | | **Detailed Specifications** | | **Control System:** +Pulse Width Control 1500usec Neutral **Required Pulse:** 3-5 Volt Peak to Peak Square Wave **Operating Voltage:** 4.8-6.0 Volts **Operating Temperature Range:** -20 to +60 Degree C **Operating Speed (4.8V): 0.23sec/60 degrees at no load Operating Speed (6.0V): 0.18sec/60 degrees at no load Stall Torque (4.8V): 61 oz/in. (4.3kg.cm) Stall Torque (6.0V): 76 oz/in. (5.5kg.cm) Operating Angle:** 45 Deg. one side pulse traveling 400usec **360 Modifiable:** Yes **Direction:** Clockwise/Pulse Traveling 1500 to 1900usec **Current Drain (4.8V):** 8mA/idle and 150mA no load operating **Current Drain (6.0V):** 8.8mA/idle and 180mA no load operating **Dead Band Width:** 8usec **Motor Type:** 3 Pole Ferrite Motor **Potentiometer Drive:** Indirect Drive **Bearing Type:** Top Ball Bearing, Lower Bushing **Gear Type:** Karbonite Gears **Connector Wire Length:** 11.81" (300mm) **Dimensions:** 1.50" x 0.8"x 1.4" (41 x 20 x 37mm) **Weight:** 1.4oz (39g) | | HS-475HB_schem**Rys. 6.** Rysunek silnika HS-475HB |
| HS-85BB   |  | | --- | | **Detailed Specifications** | | **Control System:** +Pulse Width Control 1500usec Neutral **Bearing type:** Single Top Ball Bearing **Operating Voltage:** 4.8-6.0 Volts **Operating Speed (4.8V): 0.16sec/60 degrees  Operating Speed (6.0V): 0.18sec/60 degrees  Stall Torque (4.8V): 42 oz/in. (3kg.cm) Stall Torque (6.0V): 49 oz/in. (3.5kg.cm) Motor Type:** 3 Pole Ferrite Motor **Dimensions:** 1.1" x 0.45" x 1.15 " (29mm x 13mm x 30mm) **Weight:** 0.7oz (19.2g) | |  |