Synchronizacja procesów-semafory i rendezvous

Proszę wpierw zapoznać się z programami przykładowymi Semaphores.vi oraz Rendezvous.vi. Można je uruchomić albo na komputerze głównym albo na docelowym – to nie ma znaczenia dla ich funkcjonalności.

1. *Semaphores.vi*

W LabVIEW moduły semaforów znajdują się w zakładce *Functions> Programming> Synchronization> Semaphore.* Prezentowany program generuje **niejednocześnie** trzy różne przebiegi (sinusoidalny, trójkątny i piłokształtny). Każdy z tych przebiegów jest generowany przez 2s, a kolejność generowania kontrolowana jest z wykorzystaniem semaforów. Proszę zapoznać się z własnościami stosowanych funkcji oraz działaniem całego programu. Proszę zwrócić uwagę na nietypową metodę zatrzymywania programu.

Ciekawostka: Inicjalizacja (wymazywanie zawartości) diagramów graficznych dokonuje się w tym programie w inny sposób, niż dotychczas poznaliśmy. Proszę kliknąć w menu **File** okna panelu i wybrać **VI Properties.** Następnie w rozwijanym menu **Category** należy wybrać **Execution.** Znajdziemy tam zaznaczoną opcję **Clear indicators when called**, która powoduje czyszczenie wyświetlaczy przy uruchomieniu programu.

2. *Rendezvous.vi*

Program ten jest bardzo podobny w konstrukcji do poprzedniego. Wykorzystuje on funkcje *Rendezvous*, które wymuszają jednoczesny start wykonywania wybranych części programu. Funkcje te znajdują się w *Functions> Programming>* Synchronization> *Rendezvous*. Każdy sygnał generowany jest przez inną ilość czasu: 1s, 2s i 4s. Analogicznie, jak poprzednio, proszę zapoznać się z opisem używanych funkcji oraz z działaniem programu.

3. Własny program z użyciem funkcji synchronizujących

Tworzymy projekt *Nazwisko_Prac_B_sem*, w którym na komputerze docelowym (kontrolerze PXI) umieścimy dwa programy, np. Nazwisko*_B_1.vi* i *Nazwisko_B_2.vi*. Ponieważ oba programy będą bardzo do siebie podobne, więc najlepiej jest je pisać równocześnie. Opis ćwiczenia zakłada taką metodę pisania.

Poniżej przedstawione są po kolei obiekty, które należy umieścić w programie od lewej strony do prawej, **zapewniając przedstawioną kolejność ich wykonywania**:

1. Bloczek rozpoczynający śledzenie programu (*TraceTool Start Trace*) z buforem 1e6 i śledzeniem szczegółowym. Ten bloczek będzie obecny tylko w programie *Nazwisko_B_1.vi* - śledzenie i tak będzie obejmować oba programy.
2. W obu programach tworzymy *Rendezvous* o nazwie np. RenB – taki napis podpinamy do wejścia *Name*. Dzięki temu dalsze części obu programów zaczną się wykonywać równocześnie.
   Nie jest problemem to, że tworzymy Rendezvous (lub semafor) dwukrotnie – pierwsza wywołana funkcja utworzy obiekt, a druga nie zrobi nic, bo obiekt już istnieje.
3. Bloczek oczekiwania na synchronizację *Rendezvous*.
4. *TraceTool Log User Event* ze zdarzeniem o ID = 0, a w programie *Nazwisko_B_2.vi* o ID = 3.
5. W obu programach tworzymy semafor o nazwie np. SemB.

6. W obu programach wstawiamy pętlę czasową o nazwie P-1 (odpowiednio P-2) zawierająca trzy wewnętrzne ramki. Okres wykonywania obu pętli ma wynosić 10 μs – odpowiednie źródło czasu wybieramy klikając prawym klawiszem myszki w lewe wejścia pętli i wybierając *Configure Input Node*. Później będziemy jeszcze zmieniać okres pętli, ich priorytety oraz będziemy wymuszać, by wykonywały się albo na tym samym, albo na różnych procesorach.

   Pętla czasowa podzielona jest wewnętrznie na 3 ramki oraz ma się wykonywać czterokrotnie. Jej zawartość jest następująca:

   - Lewa ramka zawiera żądanie semafora, a **po nim** zdarzenie *TraceTool Log User Event* o ID = 1 (odpowiednio ID = 4).
   - W ramce środkowej znajdują się bloczki, które będą odpowiedzialne za zatrzymanie pętli po wykonaniu 4 iteracji.
   - Prawa ramka zawiera zdarzenie *TraceTool Log User Event* o ID = 2 (odpowiednio ID = 5), a **po nim** uwolnienie semafora.

7. Za pętlą czasową umieszczamy uwolnienie referencji do semafora, a następnie do Rendezvous.
8. Na samym końcu programu *Nazwisko_B_1.vi* umieszczamy bloczek wysyłający prześledzony program na komputer główny (**169.254.148.100)**.

Otwieramy *Trace Viewer* i definiujemy kolory zdefiniowanych flag. Wyłączamy wszystkie flagi systemowe (można czasowo włączyć flagi *Loop Done* i zaobserwować kończenie pętli i sekwencji w pętli).

Należy wpierw uruchomić *Nazwisko_B_2.vi,* a po nim *Nazwisko_B_1.vi,* by nie śledzić niepotrzebnie czasu pomiędzy uruchomieniem przez nas programów.

Proszę

- zaobserwować pozycje flag dla różnych okresów wykonywania pętli (10 μs, 100 μs, 200 μs).
- uruchamiać pętle bądź na tym samym, bądź na różnych procesorach.
- zmienić priorytety pętli i zaobserwować zachowanie.

4. Podłączenie urządzenia pomiarowego

Teraz wracamy do zadania z pracowni 9 i zmieniamy je zastępując generator sygnału rzeczywistym urządzeniem pomiarowym zainstalowanym w kasecie PXI. Jeżeli nie mamy już tego programu, to można ściągnąć z pegaz.uj.edu.pl pliki Target_9.vi i Host_9.vi, włączyć je do projektu i tylko dodać zmienne współdzielone, by wyświetlić sygnał na wyświetlaczu.

Usuwamy generator z programu **Target_9.vi**. W jego miejsce należy wstawić funkcję obsługującą urządzenia (**DAQ Assist**, czyli Asystent DAQ, który jest najprostszą funkcją umożliwiającą na stworzenie systemu akwizycji danych), która znajduje się w *Functions> Measurement I/O> DAQmx – Data Acquisition*
lub w *Functions> Express> Input.*
Po ułożeniu bloczka na diagramie blokowym otworzy się okno umożliwiające jego konfigurację.

a) Konfiguracja *DAQ Assistant*.

Będziemy odbierać sygnały analogowe przychodzące na pierwsze wejście analogowe karty PXI-6221, więc wybieramy *Acquire Signals> Analog Input> Voltage>* kanał ai0 urządzenia PXI-6221. Klikamy *Finish.* Na następnym oknie, jako *Terminal Configuration* wybieramy RSE (*ground-Referenced Single-Ended*, czyli sygnał mierzony względem wspólnej masy). Jako *Acquisition Mode* wybrać należy *N Samples, Samples to read* = 1000, *rate (Hz)* = 1000 lub 1k. Zamykamy okno klikając OK. Parametry próbkowania można też podawać na bloczek z zewnątrz: *rate* (ilość pomiarów, których w ciągu sekundy dokonuje układ ADC na karcie 6221) oraz *number of samples* (ilość próbek przekazana w ciągu sekundy) możemy podpiąć do stałych o wartości równej 1000. Do wejścia *stop (F)* zmienną STOP. Wyjście *data* ze zmienną DaneGen.

b) Odbierany sygnał należy przekazać do programu **Host_9.vi** i wyświetlić na tym samym wykresie, co wcześniej sygnał z generatora. Jeżeli znajduje się tam również pisanie do pliku, co robiliśmy na pracowni 9, to należy je albo usunąć, albo zakomentować.
c) Można poeksperymentować z reżymem pracy Asystenta DAQ i zmienić go z *N Samples*, na ciągły.