|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| logoKM | logoKM | | POLITECHNIKA ŚLĄSKA  WYDZIAŁ ELEKTRYCZNY  KATEDRA MECHATRONIKI | logoKM |
| Instrukcja do ćwiczenia laboratoryjnego | | | | |
| Przedmiot: | | Przetwarzanie i Wizualizacja Danych Pomiarowych | |  |
| Symbol ćwiczenia: | | PiWDP10 | |  |
| Tytuł ćwiczenia: | | **Struktury programistyczne: struktura Event** | | |

**SPIS TREŚCI**

|  |  |  |
| --- | --- | --- |
|  | **SPIS RYSUNKÓW** | 2 |
| 1. | **CELE ĆWICZENIA** | 3 |
| 2. | **WPROWADZENIE** | 3 |
| 2.1. | **Pętle równoległe** | 3 |
| 2.2. | **Struktura master/slave** | 3 |
| 3. | **LABORATORYJNE STANOWISKO BADAWCZE** | 7 |
| 3.1. | **Obiekt badany** | 7 |
| 3.2. | **Urządzenia dodatkowe** | 7 |
| 3.3. | **Oprogramowanie** | 7 |
| 4. | **PROGRAM ĆWICZENIA – WYKAZ ZADAŃ DO REALIZACJI** | 7 |
| 5. | **PRZYKŁAD REALIZACJI ZADANIA – zatrzymanie pętli równoległych za pomocą jednego przycisku *STOP*** | 4 |
| 6. | **RAPORT** | 9 |
| 7. | **PYTANIA** | 9 |

# SPIS RYSUNKÓW

|  |  |  |
| --- | --- | --- |
| 1. | Tworzenie zmiennej lokalnej. | 3 |
| 2. | Widok zmiennej globalnej: powłoka, zmienna i panel zmiennej. | 4 |
| 3. | Widok szablonu zmiennej funkcjonalnej. | 5 |
| 4. | Zmienna funkcjonalna realizująca dwa zadania zapisu i odczytu. | 5 |
| 5. | Przykład pracy równoległej z występowaniem efektu Race Condition. | 6 |
| 6. | Zabezpieczenie przed efektem Race Condition z zastosowaniem zmiennej funkcjonalnej. | 7 |
| 7. | Zastosowanie zmiennej typu FVG do odmierzania przedziałów czasu. | 9 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

1. **CELE ĆWICZENIA**

Celem ćwiczenia jest poznanie się z:

* Obsługą struktury Event do obsługi zdarzeń wywołanych przez obiekty panelu
* Programowaniem zdarzeń oraz filtrów w strukturze Event.

1. **WPROWADZENIE**
   1. **Pętle równoległe**
   2. **Struktura master/slave**
2. **LABORATORYJNE STANOWISKO BADAWCZE**
   1. **Obiekt badany**

- Środowisko programistyczne LabVIEW,

* 1. **Urządzenia dodatkowe**

- brak,

* 1. **Oprogramowanie**
* LabVIEW 2013 lub nowszy

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

Kolejne kroki do wykonania podczas zajęć:

* Zastosowanie struktury Event:

- otworzyć projekt „*System pomiarowy DAQ*”,

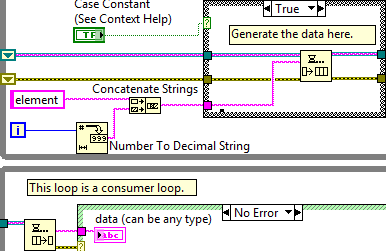
- otworzyć nowy plik z szablonu **File** 🡪 **New…** „ *Producer/Consumer Design Pattern (Data)*”

- zapisać plik pod nazwą „*main\_rownolegle.vi*”,

- w górnej pętli zamienić stałą typu **boolean** na kontrolkę i zamienić jej akcję na „*Latch When Pressed*” lub „*Latch When Released*”,\*

- w dolnej pętli umieścić wskaźnik typu **string** i podpiąć do odpowiedniego tunelu,

- dopisać fragment kodu podany niżej:



- sprawdzić działanie programu uruchamiając aplikację,

- sprawdzić działanie programu uruchamiając aplikację z opcją **Highlight Execution**,

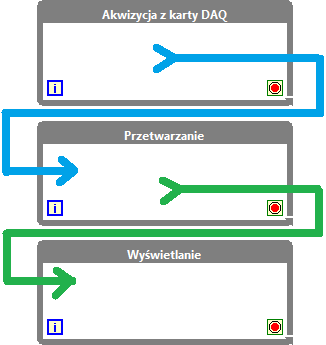
- sprawdzić działanie programu po wciśnięciu przycisku **STOP**!

- w pętli konsumenta dać opóźnienie czasowe sterowane za pomocą kontrolki, uruchomić aplikację i sprawdzić działanie programu w trzech przypadkach:

* Pętla producent działa szybciej niż pętla konsument,
* Obydwie pętle działają z tą samą szybkością,
* Pętla producent działa wolniej niż pętla konsument (chwilowe działanie),

- opracować strukturę pętli równoległych, pracujących w strukturze producent/konsument

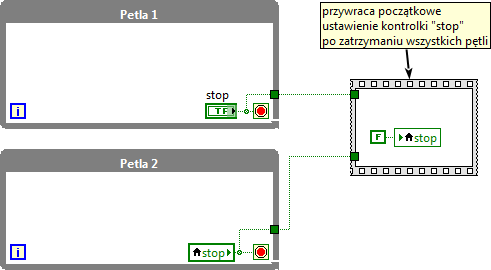
Zgodnie z rysunkiem:



- wypełnić pętle kodem realizującym odpowiednie zadania (podpowiedź: można skorzystać z kodu zapisanego w odpowiednich stanach maszyny stanów),

1. **PRZYKŁAD REALIZACJI ZADANIA – filtracja zdarzenia zamknięcia aplikacji**

W celu zatrzymania kilku pętli równoległych jednym przyciskiem *stop* należy posłużyć się zmiennymi lokalnymi do tego przycisku. Jest to jedno z klasycznych zastosowań zmiennych lokalnych. W przypadku użycia zmiennych z kontrolkami typu **boolean**, kontrolki te nie mogą wykonywać akcji typu **Latch**. Należy więc zmienić działanie przycisku na wybraną metodę **Switch**. W celu przywrócenia stanu przycisku *stop* do wartości początkowej można zastosować klatkę struktury **Flat Sequence** wykonywanej po zakończeniu wszystkich pętli.



1. Zastosowanie zmiennej lokalnej „stop” do zatrzymania dwóch pętli równoległych.
2. **RAPORT**

Raport z przeprowadzonego ćwiczenia laboratoryjnego powinien zawierać opis kolejnych czynności wykonywanych w trakcie realizacji ćwiczenia, zrzuty ekranu dokumentujące wykonane kroki oraz zanotowane parametry konfiguracyjne kart DAQ (mogą być zawarte  
w tabeli)

1. **PYTANIA**
2. Omówić metody programowania równoległego.
3. Wymienić metody przesyłania danych między pętlami równoległymi, podać zalety i wady każdej metody.
4. Co to jest „sztuczna zależność struktur”?

# LITERATURA

1. LabVIEW Core 1 Course manual.
2. LabVIEW Core 1 Exercise book.
3. LabVIEW Core 2 Course manual.
4. LabVIEW Core 2 Exercise book.
5. https://www.youtube.com/watch?v=iNm0zWY7o8g&list=PLUnVykytJXxPxm5u0vRKpPRVVg2u\_WgFg&index=3
6. https://www.youtube.com/watch?v=RuIN31rSO2k

Opracowanie: Marek Kciuk

# ZADANIA DO REALIZACJI PRZED PRZYSTĄPIENIEM DO ĆWICZENIA

- ukończyć zadania z instrukcji nr 7.

# ZADANIA DODATKOWE PO WYKONANIU ĆWICZENIA

- uzupełnić strukturę danych oraz działanie programu „*Main\_Maszyna.vi*”, żeby były przechowywane wektory próbek wartości *Temperatura* i *Średnia*. – zadanie ma na celu przygotowanie programu do uzupełnienia stau „*SAVE*”.

- zmodyfikować działanie aplikacji tak, żeby sekwencja: Measure, Analyze, Display była zapętlona przez zadany okres czasu, z możliwością wcześniejszego wyjścia.

- zmodyfikować program tak, żeby przycisk „*Zakończ Pomiar*” znajdował się w strukturze kontrolki „Akcje”.

# ZAŁĄCZNIKI

## Opóźnienia i zarządzanie czasem działania aplikacji