

## Synchronizacja barierowa startu struktury czasowej

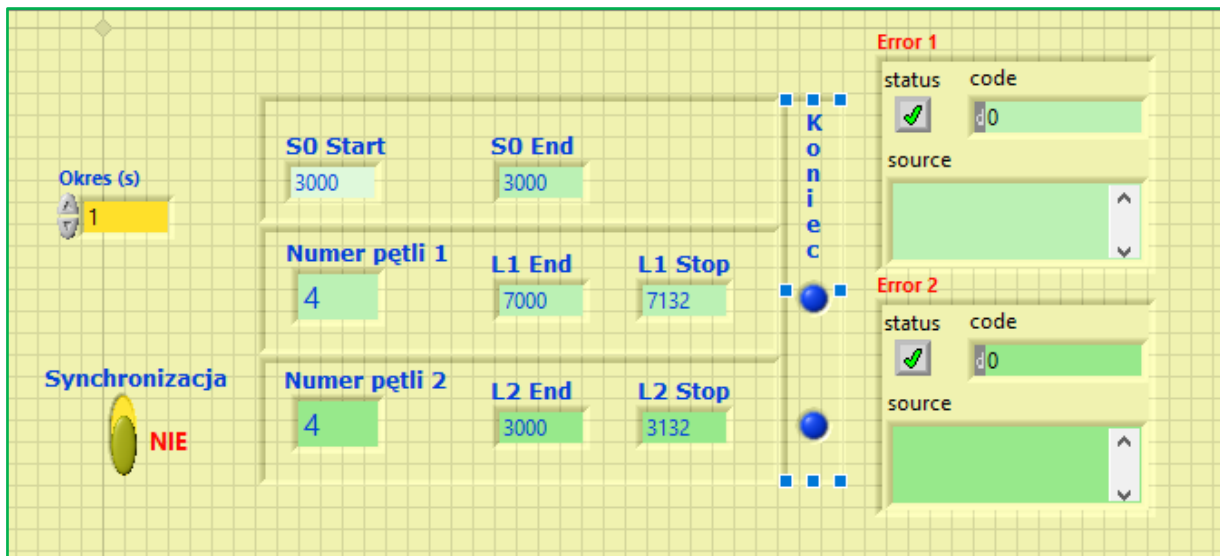
Podczas wykonywania ćwiczenia przyda się wiedza dotycząca pracy z kontrolerem kasety PXI, dlatego dobrze jest przeczytać punkt C z instrukcji do poprzedniego ćwiczenia.

### Wykonanie aplikacji z synchronizacją struktur czasowych

Program ma podawać czasy wykonania pętli czasowych zsynchronizowanych lub niesynchronizowanych ze sobą. Poza funkcją synchronizującą pętle czasowe, trzeba będzie użyć zmiennych lokalnych oraz inicjalizować wskaźniki, by po uruchomieniu programu wszystkie wskazywały wartości domyślne.

Tworzymy nowy, pusty **projekt** i na komputerze **docelowym** (PXI) tworzymy nowy pusty vi. Nadajemy im jakieś nazwy poprzez zapisanie na dysku, np. Nazwisko\_prac7.

Poniżej przedstawiona będzie kolejność budowania panelu czołowego, więc proszę wpierw przeczytać, a później budować. Panel czołowy ma wyglądać mniej więcej tak:

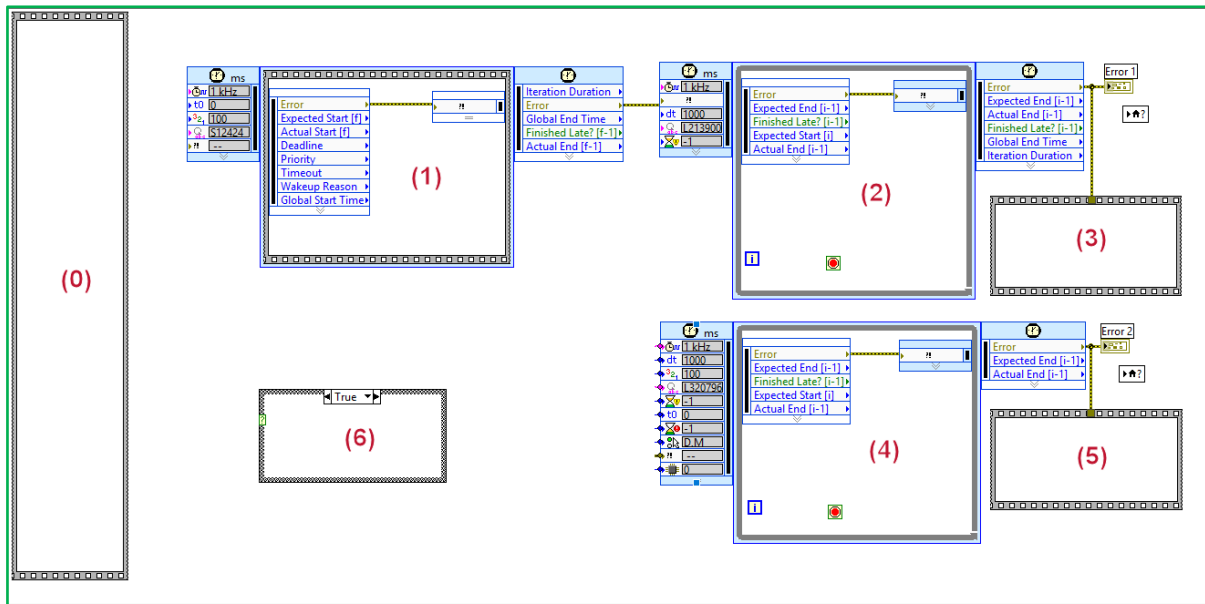


Poniżej znajduje się diagram blokowy, na którym umieszczono jedynie wymagane (ponumerowane) pętle i struktury oraz zaznaczono propagację błędów.

Potrzebne będą dwie **pętle** czasowe *Timed Loop* (2) i (4), których startowanie będzie mogło być synchroniczne, następnie jedna **sekwencja** czasowa *Timed Sequence* (1), która będzie poprzedzać pierwszą pętlę czasową oraz trzy zwykłe struktury sekwencyjne *Flat Sequence* (0), (3) i (5). Potrzebna będzie również jedna struktura *case* (6).

Struktura (1) będzie opóźniać start pętli czasowej (2). Pętla (2) będzie mogła być zsynchronizowana z pętlą (4). Funkcja synchronizująca będzie używana w zależności od przełącznika **Synchronizacja** i skorzysta ze struktury *case* (6). Sekwencje (3) i (5) zostaną użyte przy liczeniu końcowych parametrów czasowych wykonania programu, natomiast sekwencja (0) będzie dobrym miejscem na umieszczenie wszystkich funkcji inicjalizujących działanie programu.

## SYSTEMY CZASU RZECZYWISTEGO PRACOWNIA 8



### Teraz po kolei:

- a) Proszę ustawić na diagramie blokowym **sekwencje czasową (1)**, następnie, po prawej, **pętlę czasową (2)** i, jeszcze bardziej po prawej, zwykłą sekwencję **(3)**. Należy korzystać we wszystkich strukturach czasowych z tego samego zegara taktującego: w tym celu należy umieścić na lewo od struktury czasowej **(1)** źródło czasu **Create Timing Source** (funkcję tę można znaleźć tam, gdzie struktury czasowe) i połączyć jego wyjście *name (out)* z wejściami *Source Name* sekwencji **(1)** i pętli czasowej **(2)**. Wyjście *Error (out)* źródła łączymy z wejściem *Error* sekwencji czasowej. Ponieważ struktury te mają być wykonywane po kolei, należy połączyć wyjście błędów z sekwencji czasowej z wejściem błędów pętli czasowej oraz, analogicznie, wyjście błędów pętli czasowej z brzegiem zwykłej sekwencji **(3)**. To ostatnie podłączenie wymusi uruchomienie zwykłej sekwencji dopiero po zakończeniu pętli czasowej. Należy również połączyć wejścia i wyjścia błędów wewnątrz struktur – tak, jak na diagramie powyżej.
- b) Sekwencje czasową **(1)** proszę wyposażyć w opóźnienie początkowe (np. 3 s) – do wejścia *Offset* trzeba podpiąć stałą równą 3000 (w milisekundach). Strukturze tej można dać również nazwę: do wejścia *Structure Name* podłączmy stałą z nazwą **S0**.
- c) Wskaźnik **S0 Start** ma wskazywać początek wykonywania struktury czasowej **(1)**. Dlatego wyjście wewnątrz tej sekwencji *Actual Start [f]* trzeba połączyć ze wskaźnikiem **S0 Start**. Natomiast wskaźnik **S0 End** ma wyświetlać czas ukończenia tej struktury: wyjście zewnętrzne *Actual End [f-1]* łączymy ze wskaźnikiem **S0 End**.
- d) Pętli czasowej **(2)** nadajemy nazwę **L1**. Wejście *Source Name* już mamy podpięte (opisane w punkcie a). Musimy jeszcze podpiąć okres wykonywania się tej pętli. W tym celu wyjście kontrolki **Okres (s)** wpierw mnożymy przez 1000 (by sekundy zmienić na milisekundy), a wynik podpinamy do wejścia *Period* Pętli **L1**. We własnościach kontrolki **Okres (s)** proszę ustalić możliwe wartości od 0.5 do 20 s z krokiem zmiany 0.5 s. Wewnątrz pętli należy połączyć *Actual End [i-1]* z wyświetlaczem **L1 End**. Następnie, aby wyświetlić czas ukończenia wykonywania pętli **(2)** **zewnętrzne** wyjście *Actual End [i-1]* łączymy ze zmienną lokalną wyświetlacza **L1 End**. Pętla powinna wykonywać się zawsze czterokrotnie, aktualny numer wykonywanej pętli powinien być wypisany na wyświetlaczu **Numer pętli 1** (wyświetlana numeracja iteracji powinna się zaczynać od liczby 1, a nie 0). Aby umieścić na panelu czołowym wskaźnik błędu **Error 1**, należy kliknąć w wyjście *Error* pętli **(2)** i w zakładce *Create* wybrać *Indicator* – utworzy się odpowiedni wyświetlacz.

SYSTEMY CZASU RZECZYWISTEGO  
PRACOWNIA 8

- e) Końcowa struktura sekwencyjna (3) oznajmi koniec wykonywania tej części programu. Odpowiedni wskaźnik diody, znajdujący się na panelu czołowym, należy na diagramie blokowym umieścić wewnątrz struktury sekwencyjnej (3) z podłączoną stałą logiczną *True*.
- f) Wewnątrz struktury (3) trzeba wypisać również rzeczywisty czas, w którym skończyło się wykonywanie tej części programu, czyli tego toru struktur. W tym celu trzeba w pierwszej inicjalizować niezależny pomiar czasu: Wewnątrz struktury (0) należy umieścić funkcję *Tick Count (ms)*, która znajduje się w *Functions > Programming > Timing* oraz połączyć ją ze wskaźnikiem o jakiejś nazwie np. **czas\_pocz**, który można umieścić gdziekolwiek na panelu czołowym. Jest on potrzebny tylko po to, by móc użyć jego zmiennej lokalnej, można więc go ukryć (jego wskazania są nieinteresujące) klikając w niego na diagramie blokowym i wybierając *Hide Indicator*. Natomiast w końcowej strukturze sekwencyjnej (3) trzeba teraz tylko odjąć aktualny czas dany przez umieszczoną tam kolejną funkcję *Tick Count (ms)* od czasu początkowego, czyli od zmiennej lokalnej wskaźnika **czas\_pocz**. Wynik tego odejmowania podajemy na wskaźnik **L1 Stop**.
- g) Wszystkie wskaźniki, które do tej pory pojawiły się na diagramie powinny być inicjalizowane na początku wykonywania programu. Jak już to robiliśmy, tworzymy ikony *Reinitialize to Default* i umieszczamy na początku, wewnątrz struktury (0).
- h) Proszę umieścić na diagramie blokowym drugi tor struktur sekwencyjnych, czyli pętlę czasową (4) (o nazwie **L2**) i strukturę sekwencyjną (5) – najlepiej pod tymi z pierwszego toru. Analogicznie, wejście *Source Name* pętli czasowej proszę podpiąć do tego samego źródła czasu, co poprzednie struktury czasowe oraz podać tę samą wartość okresu wykonywania się pętli **L2**, co **L1**. Na panelu czołowym umieszczamy wskaźniki **Numer pętli 2**, **L2 End**, **L2 Stop**, dolną diodę oraz wskaźnik błędu **Error 2**. Pętla ta powinna się również wykonywać czterokrotnie, a wskaźnik **Numer pętli 2** powinien funkcjonować tak samo, jak w pierwszej pętli czasowej. Wszystkie połączenia wewnątrz struktur (4) i (5) są analogiczne, jak w (2) i (3) z tym, że używamy właściwych wskaźników.
- i) Wszystkie wskaźniki inicjalizujemy w (0). Aby mieć pewność, że struktura (0) wykona się na początku, łączymy linie błędu któregośkolwiek z modułów inicjalizujących z wejściem błędu funkcji *Create Timing Source*.
- j) Ostatnim elementem tworzonego programu jest synchronizacja barierowa pętli czasowych. Bez zsynchronizowania oba tory (S0+L1 oraz L2) działają niezależnie od siebie. Aby je zsynchronizować wykorzystamy nazwy, które nadaliśmy pętlom (L1 i L2) – przypomnę, że podpinamy do ich wejść *Structure Name* stałe z odpowiednimi nazwami. Następnie należy umieścić w strukturze (6), w części **True** struktury *case*, funkcję *Synchronize Timed Structure Starts.vi* którą można znaleźć tam, gdzie struktury czasowe. Do wejścia *Replace (T)* możemy podłączyć stałą *True*, do wejścia *synchronization group name* jakąś nazwę (np. L12), a do wejścia *timed structure names* trzeba podłączyć nazwy pętli czasowych połączone w tablicę (jak na rysunku powyżej). Można również do wejścia *timeout (ms)* podpiąć stałą równą -1 (czekanie na pętle do skutku). Struktura *case* (6) jest kontrolowana przez kontrolkę **Synchronizacja**. Przypadek *False* tej struktury należy pozostawić pusty, nie ma wtedy żadnej synchronizacji. Na innych zajęciach poznamy inne funkcje umożliwiające na synchronizację barierową.
- k) Otrzymany program proszę wykonać w modzie **Synchronizacja – TAK** oraz **Synchronizacja – NIE** i porównać jego działanie. Czy rozumiemy, dlaczego takie wartości czasu są wyświetlane dla różnych okresów (zwłaszcza, 0.5 s, 1 s, 1.5 s, 2 s itp.)?

