

Wydział Informatyki i Telekomunikacji  
Instytut Telekomunikacji Multimedialnej

# Komputerowe Systemy Pomiarowe

## Laboratorium

Miernik czasu reakcji

## 1. Opis zadania

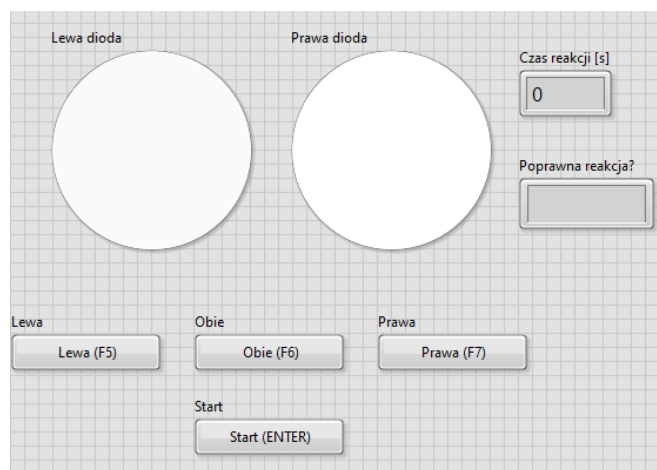
Celem zadania jest opracowanie aplikacji przeznaczonej do pomiaru czasu reakcji użytkownika.

Program po naciśnięciu przycisku *Start* rozpocznie odliczanie losowego odcinka czasu, następnie zaświeci wybraną losowo kombinację diod, w zależności od zaświeconej kombinacji diod użytkownik musi jak najszybciej nacisnąć odpowiedni przycisk decyzyjny: *Lewa*, *Prawa*, *Obie*. Program mierzy czas od chwili zaświecenia kombinacji diod do chwili naciśnięcia jednego z przycisków decyzyjnych, sprawdza również czy wciśnięto odpowiedni przycisk.

Podczas realizacji zadania poznajemy podstawy programowania w LabVIEW m.in.: tworzenie panelu czołowego, konfigurowanie diod LED, przycisków, wykorzystanie zmiennych lokalnych, określanie opóźnień, pomiar odcinka czasu. Zadanie pokazuje różnice pomiędzy wzorcami projektowymi wymuszonym przepływem danych, a pętlą zdarzeń.

### 1.1. Miernik czasu reakcji, wymuszony przepływ danych

Opracowanie programu należy rozpocząć od zaprojektowania panelu czołowego aplikacji. Przykładowy panel pokazano na rysunku 1.



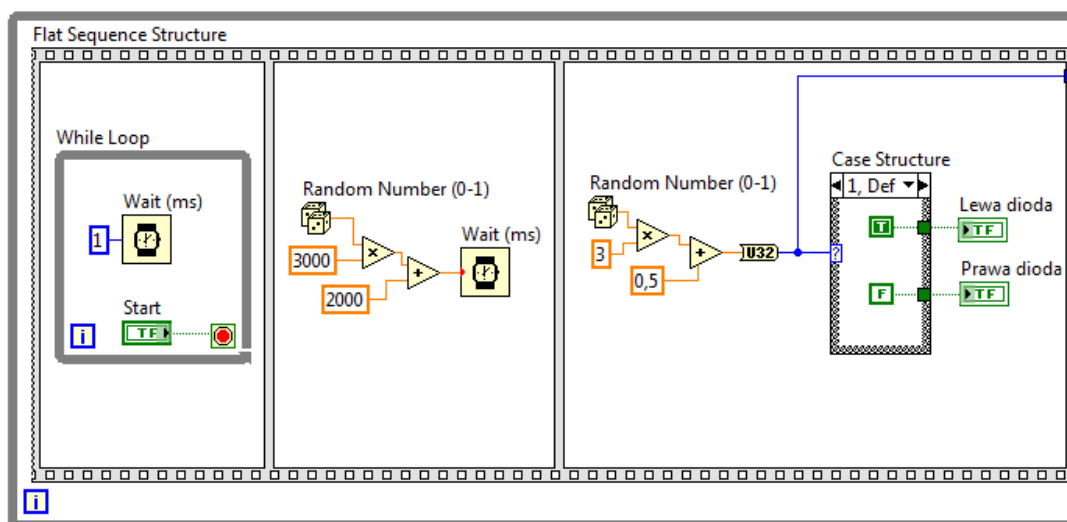
Rys. 1. Przykład wyglądu panelu użytkownika miernika czasu reakcji.

Przedstawiony panel można zaprojektować w następujący sposób:

- Uruchomić LabVIEW, utworzyć nowy plik VI, menu *File/New VI*.
- Przejść do okna panelu *Front Panel*.
- Na panelu umieścić dwie diody LED oraz cztery przyciski (ang. *Buttons*) z palety *Controls / Boolean*.
- Umieścić wyświetlacze: numeryczny *Numeric Indicator* z palety *Controls / Numeric* oraz tekstowy *String Indicator* z palety *Controls / String & Path*.
- Zmienić etykiety (*label*) umieszczonych obiektów, w tym celu należy po najejchaniu na nie kursorem myszy (kursor w kształcie strzałki) dwukrotnie kliknąć.
- Powiększyć diody LED i zmienić ich kolory. W celu zmiany koloru diody należy kliknąć na niej prawym przyciskiem myszy (PPM) i wybrać opcję *Properties*, następnie w zakład-

ce *Appearance* w polu *Colors* ustawić kolory dla diody świecącej (stan *ON*, *True*) i zgaszonej (stan *OFF*, *False*).

- Przypisać do przycisków wirtualnych przyciski klawiatury komputera. W tym celu należy na każdym przycisku kliknąć PPM i wybrać opcję *Properties*, następnie w zakładce *Key Navigation* na liście w polu *Toggle* wybrać odpowiednie przyciski klawiatury (jak pokazano na rysunku 1).
- Zapisać plik VI nadając nazwę *001\_miernikl\_xx.vi* (gdzie *xx* inicjały twórcy programu).
- Przejść do okna *Block Diagram*. Na rysunku 2 pokazano pierwszy fragment kodu aplikacji, który generuje losowy odcinek czasu i losową kombinację załączania diod.



Rys. 2. Pierwszy fragment kodu aplikacji do pomiaru czasu reakcji

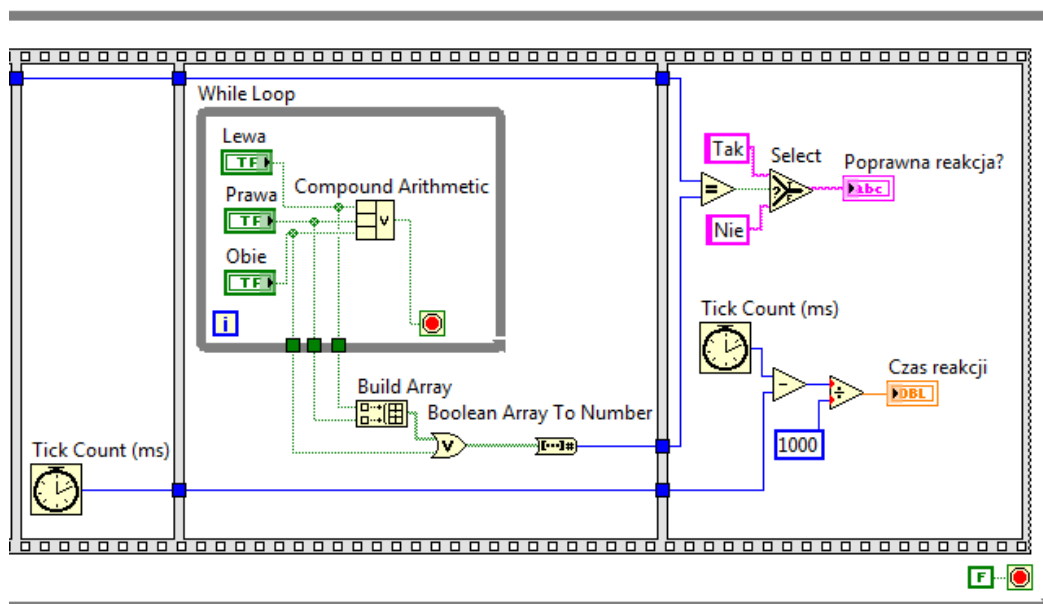
Wszystkie elementy diagramu znajdują się w paletach należących do *Functions* / *Programming*.

- Do wymuszania przepływu danych należy wykorzystać strukturę *Flat Sequence* z palety *Structures*. W diagramie narysować pierwszą ramkę tej struktury.
- W jej wnętrzu narysować pętlę *While Loop* z tej samej palety.
- W pętli *While Loop* umieścić przycisk *Start* i element opóźniający *Wait (ms)* z palety *Timing*. Wartość opóźnienia ustawić na 1 ms klikając PPM na wejściu tego obiektu i wybierając opcję *Create / Constant*. Wciśnięcie przycisku *Start* spowoduje wyjście z pętli *While Loop* i realizację kolejnych ramek struktury *Flat Sequence*.
- Dodać kolejną ramkę struktury *Flat Sequence* klikając PPM na jej dowolnej krawędzi i wybierając opcję *Add Frame After*.
- W tej ramce będzie umieszczony fragment kodu wyznaczający losowe odcinek czasu od 2 do 5 s. W tym celu: umieszczamy funkcję *Random Number (0-1)* z palety *Numeric*, która generuje liczby losowe z przedziału od 0 do 1. Następnie wstawiać obiekty mnożący i sumujący (paleta *Numeric*), elementy programu połączyć zgodnie z rys. 2. Obiekty 3000 i 2000 tworzymy klikając PPM na odpowiednich wejściach i wybierając opcję *Create / Constant*.
- Dodać kolejną ramkę *Flat Sequence* i umieścić w niej fragment kodu losującego kombinację załączającą diody LED. Aby losowo generować liczby 1, 2 i 3 należy wartości generowane przez funkcję *Random Number (0-1)* przemnożyć przez 3, dodać 0,5 i przekonwer-

tować na liczbę całkowitą bez znaku (*Unsigned*). Do konwersji wykorzystujemy obiekt *To Unsigned Long Integer (U32)* z palety *Numeric / Conversion*.

- Wybór losowej kombinacji diod zrealizować za pomocą struktury *Case Structure* z palety *Structures* w taki sposób, aby wybór ramki nr 1 załączał lewą diodę, wybór ramki nr 2 załączał prawą diodę, a wybór ramki nr 3 załączał obie diody.
- W tym celu narysować strukturę *Case Structure*, do jej selektora dołączyć wyjście obiektu *To Unsigned Long Integer (U32)*. Wybrać ramkę 0 tej struktury i usunąć ją klikając PPM na jej krawędzi i wybierając opcję *Delete This Case*. Ramkę 1 ustawić jako domyślną klikając PPM na jej krawędzi i wybierając opcję *Make This The Default Case*. We wnętrzu tej ramki umieścić dwie stałe logiczne z palety *Boolean*: *True* i *False*. Odpowiednio połączyć stałe logiczne z terminalami diod. Powielić tę ramkę klikając PPM i wybierając *Duplicate Case*. Powstanie wówczas ramka 2. Zmienić wartości obu stałych logicznych na przeciwne. Ponownie powielić ramkę (powstanie ramka 3) i obie stałe logiczne ustawić na *True*.

Na rysunku 3 pokazano drugi fragment kodu aplikacji, która jest odpowiedzialna za pomiar czasu reakcji oraz weryfikację poprawności wciśnięcia odpowiedniego przycisku. Również w tym przypadku wszystkie obiekty znajdujące się na diagramie znajdują się w paletach należących do *Functions / Programming*.



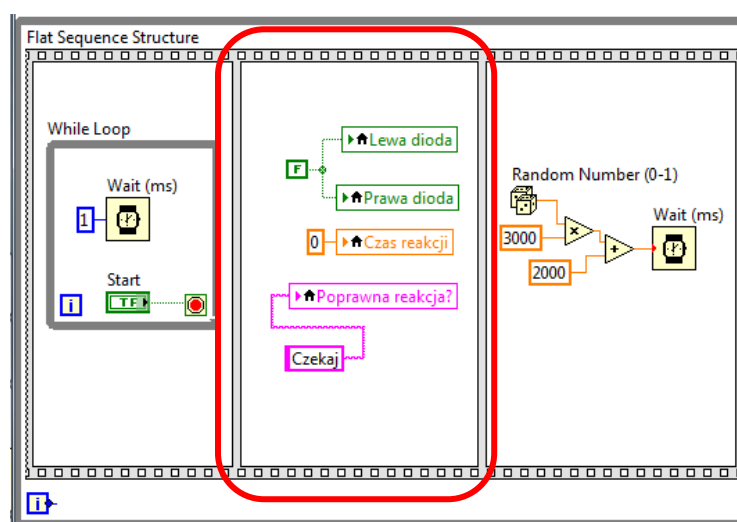
Rys. 3. Drugi fragment kodu aplikacji do pomiaru czasu reakcji

- Tworzymy kolejną ramkę struktury *Flat Sequence*. Umieszczamy w niej znacznik czasu *Tick Count (ms)* z palety *Timing*, który zwraca wartość licznika milisekund.
- Dodajemy kolejną ramkę *Flat Sequence* i umieszczamy w niej obiekty sprawdzające wciśnięcie przycisków decyzyjnych *Lewa* (dioda), *Prawa* (dioda) lub *Obie* (diody). Terminale przycisków umieszczamy w pętli *While Loop* wraz z obiektem *Compound Arithmetic* z palety *Boolean*. Łączymy jak pokazano na rysunku 3. Aby obiekt *Compound Arithmetic* miał odpowiednią liczbę wejść należy go rozciągnąć o jedną pozycję w dół. Pętla *While Loop* w tej ramce będzie wykonywana do chwili wciśnięcia dowolnego przycisku decyzyjnego.

- Do weryfikacji poprawności wciśniętego wykorzystujemy obiekt *Build Array* z palety *Array*, obiekt sumy logicznej oraz obiekt konwersji *Boolean Array To Number*. Obiekty łączymy w sposób przedstawiony na rys. 3.
- Tworzymy kolejną ramkę struktury *Flat Sequence*, w niej umieszczamy terminale wyświetlaczy *Poprawna reakcja?* i *Czas reakcji*.
- Tworzymy warunek sprawdzający poprawność wciśniętego klawisza. Do oka diagramu wstawiamy obiekty *Equal?* i *Select* z palety *Comparasion* oraz dwa obiekty *Constant* z palety *String*. Obiektom *Constant* nadajemy nazwy *Tak* i *Nie*. Obiekty łączymy zgodnie z rysunkiem 3.
- Umieszczamy drugi znacznik czasu *Tick Count (ms)*, którego wartość odejmujemy od umieszczonego wcześniej i różnicę dzielimy przez 1000 aby uzyskać wynik pomiaru w sekundach.
- Całą strukturę *Flat Sequence* umieszczamy wewnątrz pętli *While Loop*, do wejścia warunku *Loop Condition* dołączyć stałą logiczną *False*, klikając PPM na *Loop Condition* i wybierając opcję *Create Constant*.

Uruchomić i sprawdzić działanie aplikacji. Czy program działa poprawnie?

Okazuje się, że jest „pamiętany” stan diod z poprzedniej iteracji. To powoduje, że nie można poprawnie zareagować w przypadku gdy ponownie zostanie wyświetlona taka sama kombinacja diod. Aby naprawić ten błąd należy po naciśnięciu przycisku *Start* „zgasić diody”, w tym celu wykorzystujemy odwołania lokalne do stałych *Lewa dioda* i *Prawa dioda*.



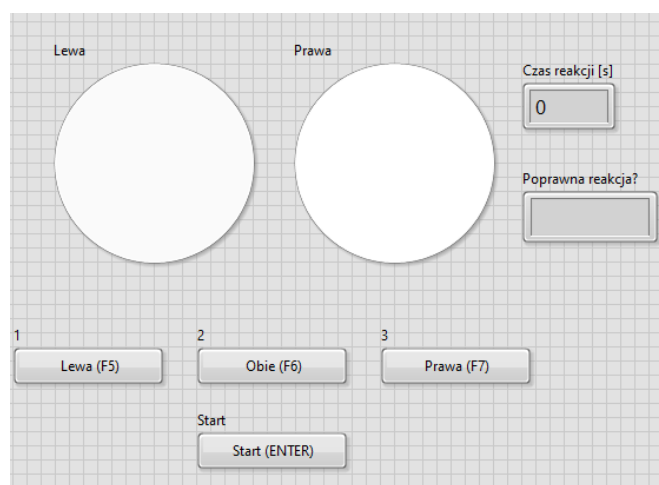
Rys. 4. Dodatkowa ramka ze zmiennymi lokalnymi

- W diagramie wstawiamy nową ramkę pomiędzy ramką z przyciskiem *Start*, a ramką odliczającą opóźnienie. Należy kliknąć PPM na krawędzi pomiędzy tymi ramkami w wybrać opcję *Insert Frame*.
- Utworzyć zmienne lokalne (ang. *Local Variable*) klikając na wybranych obiektach PPM i wybierając opcję *Create / Local Variable*. Utworzone obiekty umieścić w utworzonej poprzednio ramce. Do wejść zmiennych dołączyć odpowiednie stałe jak pokazano na rys. 4.
- Zapisać program, uruchomić i sprawdzić jego działanie.

## 1.2. Miernik czasu reakcji, pętla zdarzeń

Opracowany w poprzednim punkcie program działa prawidłowo, ale nie jest optymalny pod kątem wykorzystania zasobów sprzętowych. Stan przycisku *Start* jest odczytywany 1000 razy na sekundę, a pętla odczytująca stan przycisków decyzyjnych wykonuje się znacznie szybciej. Lepszym wyborem dla tego typu aplikacji jest **wzorec programowy pętla zdarzeń**.

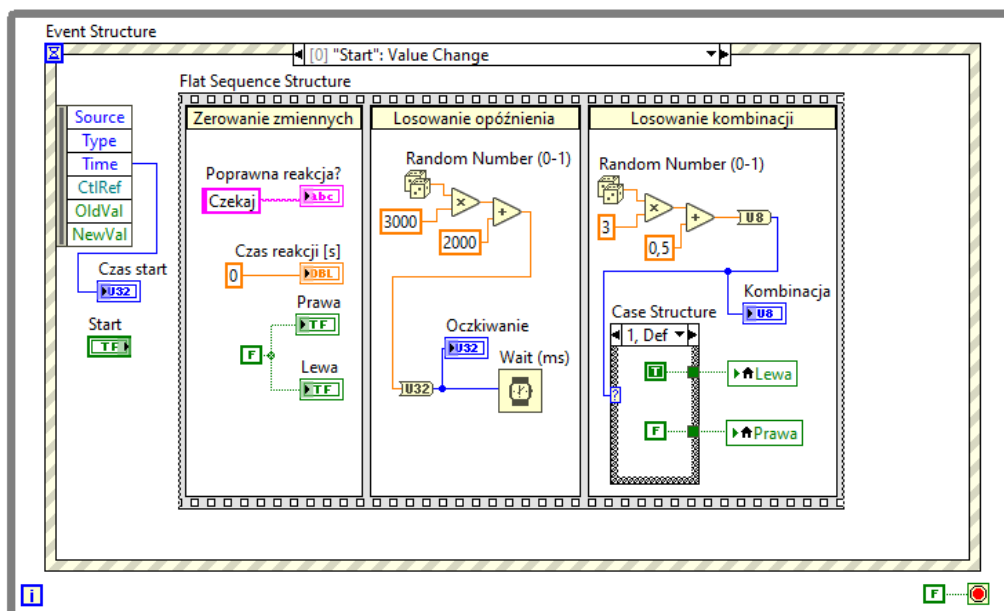
- Opracować nowy program lub zmodyfikować poprzedni zapisując jego kopię pod nową nazwą *002\_petlaZdarzen\_xx.vi* (gdzie xx inicjały twórcy programu).
- Jak poprzednio opracowanie programu należy rozpocząć od zaprojektowania panelu użytkownika. Panel powinien wyglądać podobnie jak wcześniej, z jedną różnicą. Należy zmienić etykiety (*Label*) przycisków decyzyjnych jak to pokazano na rysunku 5.



Rys. 5. Etykiety przycisków decyzyjnych.

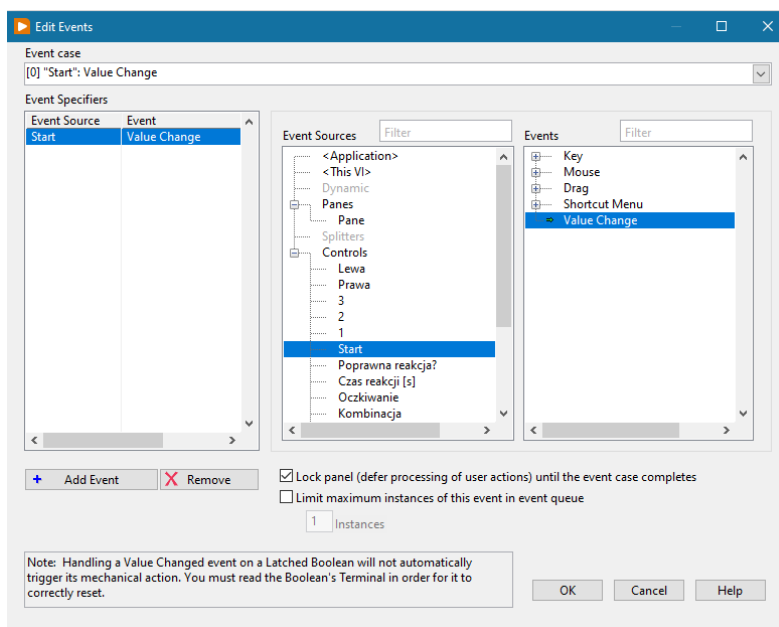
- Przycisk *Lewa* powinien mieć etykietę 1, przycisk *Obie* etykietę 3, a przycisk *Prawa* etykietę 2. Później można ukryć te etykiety na panelu klikając na przyciskach PPM i odznaczając opcję *Label* na liście *Visible Items*. Zmienić także etykiety diod: z *Lewa dioda* na *Lewa* oraz z *Prawa dioda* na *Prawa*.
- Przyjmujemy, że program będzie rozpoczynał odliczanie czasu naciśnięcia przycisku *Start* i zatrzymywał po naciśnięciu przycisku decyzyjnego. Należy zmienić sposób działania wszystkich przycisków klikając na nich PPM i wybierając opcję *Mechanical Action / Latch When Pressed*.
- Przejść do okna diagramu.
- W diagramie narysować ramkę struktury *Event Structure* z palety *Structures*.
- Na rysunku 6 pokazano kod programu z wyświetloną zawartością ramki *Start* struktury *Event Structure*.
- We wnętrzu struktury *Event Structure* umieścić terminal przycisku *Start*.
- Narysować pierwszą ramkę struktury *Flat Sequence*, ramka ta będzie służyła do „zerowania” kontrolerek wyświetlających dane. Umieścić w niej terminale diod LED oraz wyświetlaczy *Czas reakcji [s]* i *Poprawna reakcja?*. Dołączyć do nich odpowiednie stałe, wg rysunku 6, klikając na ich wejściach PPM i wybierając opcję *Create / Constant*.

- Dodać kolejną ramkę *Flat Sequence*. Umieścić w niej fragment kodu odpowiedzialny za generowanie losowego odcinka czasu.
- Dodać trzecią ramkę *Flat Sequence*. Umieścić w niej fragment kodu odpowiedzialny za generowanie losowej kombinacji włączania diod LED.



Rys. 6. Ramka *Event Structure* przycisku Start

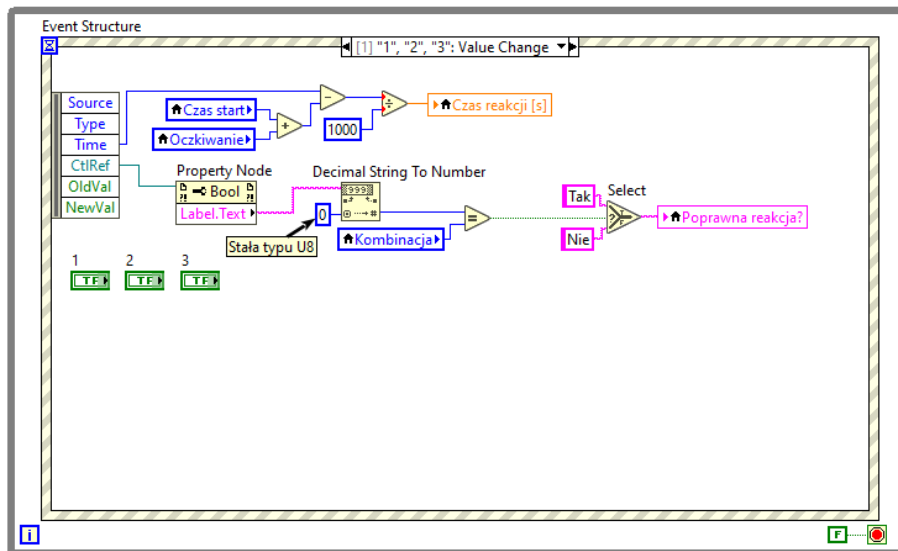
- Ustawić zdarzenie, które będzie rozpoczynało wykonanie zawartości ramki *Event Structure*. Kliknąć PPM na krawędzi tej struktury i wybrać opcję *Edit Events Handled by This Case*. Z listy źródeł zdarzeń (*Event Sources*) wybrać kontrolkę *Start* (rys. 7) i na liście zdarzeń (*Events*) zaznaczyć opcję *Value change*. Upewnić się, że jest zaznaczona opcja blokowania panelu na czas realizacji tej ramki (*Lock panel*) i zamknąć okno edycji zdarzeń klikając na przycisk *OK*.



Rys. 7. Wybór zdarzeń struktury *Event Structure*



- Ponieważ dane pomiędzy ramkami struktury *Event Structure* będą przekazywane za pomocą zmiennych lokalnych, to należy dołączyć dodatkowe kontrolki wyświetlające (*Indicator*) do:
  - wyjścia zwracającego czas powstania zdarzenia *Time* – *Czas start*,
  - wejścia obiektu opóźniającego *Wait (ms)* – *Oczekiwanie*,
  - wejścia struktury *Case Structure* – *Kombinacja*.
- Wstawiamy je klikając PPM na odpowiednich wejściach lub wyjściach i wybierając opcję *Create / Indicator*.
- Te trzy kontrolki nie muszą być wyświetlane na panelu czołowym, dlatego można je ukryć klikając na nich PPM w diagramie i wybierając opcję *Hide Indicator*; lub na panelu czołowym *Front Panel* klikając PPM na kontrolce, a z menu wybierając opcję *Advanced / Hide Indicator*.
- Dodać kolejną ramkę struktury *Event Structure*, której zawartość będzie wykonywana po naciśnięciu jednego z przycisków decyzyjnych. W tym celu należy kliknąć PPM na krawędzi *Event Structure* i wybrać opcję *Add Event Case...* W wyświetlonym oknie konfiguracji zdarzeń dodać zdarzenia *Value change* dla kontrolek (przycisków) 1, 2 oraz 3.
- Rysunek 8 przedstawia kod programu z aktywną ramką przycisków decyzyjnych struktury *Event Structure*, zawartość tej ramki wykonuje się po naciśnięciu, któregośkolwiek z przycisków decyzyjnych: *Lewa*, *Prawa*, *Obie*. W tej ramce jest również obliczana wartość czasu reakcji i sprawdzana poprawność wciśnięcia przycisku decyzyjnego.



Rys. 8. Ramka *Event Structure* przycisków decyzyjnych

- Wewnątrz tej ramki umieścić terminale przycisków *Lewa*, *Prawa*, *Obie*, czyli terminale 1, 2 i 3.
- Aby wyznaczyć czas reakcji należy od czasu zarejestrowania zdarzenia naciśnięcia przycisku decyzyjnego odjąć sumę czasu zwolnienia przycisku *Start* – *Czas start* i czasu losowego opóźnienia – *Oczekiwanie*, wartości te odczytujemy wykorzystując zmienne lokalne.
- Zmienne lokalne możemy tworzyć umieszczając w diagramie obiekt *Local Variable* z palety *Structures*, następnie ustawiamy na nim kursor myszy, klikamy LPM i z listy wybieramy, którą kontrolkę zmienna ma reprezentować. Drugi sposób tworzenia zmiennej lokal-



- nej wykorzystuje menu podręcznego obiektu, dla którego chcemy utworzyć zmienną lokalną. W celu utworzenia klikamy PPM na obiekt dla którego chcemy utworzyć zmienną lokalną i wybierając *Create / Local Variable*. Domyślnie zmienne lokalne po utworzeniu są w trybie *Write* (możliwa jest zmiana ich wartości). Aby odczytać wartość zmiennej należy zmienić jej tryb na *Read* klikając na niej PPM i wybierając opcję *Change To Read*.
- Struktura *Event Structure* za pośrednictwem terminala *CtlRef* zwraca referencję do kontroli, która wywoła zdarzenie. Ta właściwość zostanie wykorzystana do identyfikacji naciśniętego przycisku decyzyjnego.
  - Do wyjścia terminala *CtlRef* podłączyć wejście *reference* funkcji *Property Node* z palety *Application Control*. Następnie kliknąć LPM na polu *Property* i wybrać opcję *Label / Text*. Wówczas obiekt *Property Node* będzie zwracał etykietę naciśniętego przycisku. Etykieta jest typu tekstowego (*String*), korzystając z obiektu *Decimal String To Number* z palety *String / Number / String Conversion*, należy przekonwertować ją na typ numeryczny *U8*. Do wejścia *default (0 I32)* tego obiektu dołączyć stałą numeryczną i ustawić jej typ na *U8*, (opcja *Representation / U8*). W tym miejscu wyjaśnia się przyczyna zmiany etykiet przycisków decyzyjnych na wartości 1, 2 i 3.
  - Za pomocą funkcji *Equal?* z palety *Comparison* porównać numer etykiety wciśniętego przycisku decyzyjnego z wartością zmiennej *Kombinacja*, jeżeli są one równe to oznacza to, że wciśnięto właściwy przycisk.
  - Strukturę *Event Structure* otoczyć pętlą *While Loop* i dodać przycisk wyjścia z pętli.
  - Zapisać program, uruchomić i zapoznać się z jego działaniem. Porównać działanie tego programu z programem wykorzystującym wymuszony przepływ danych.

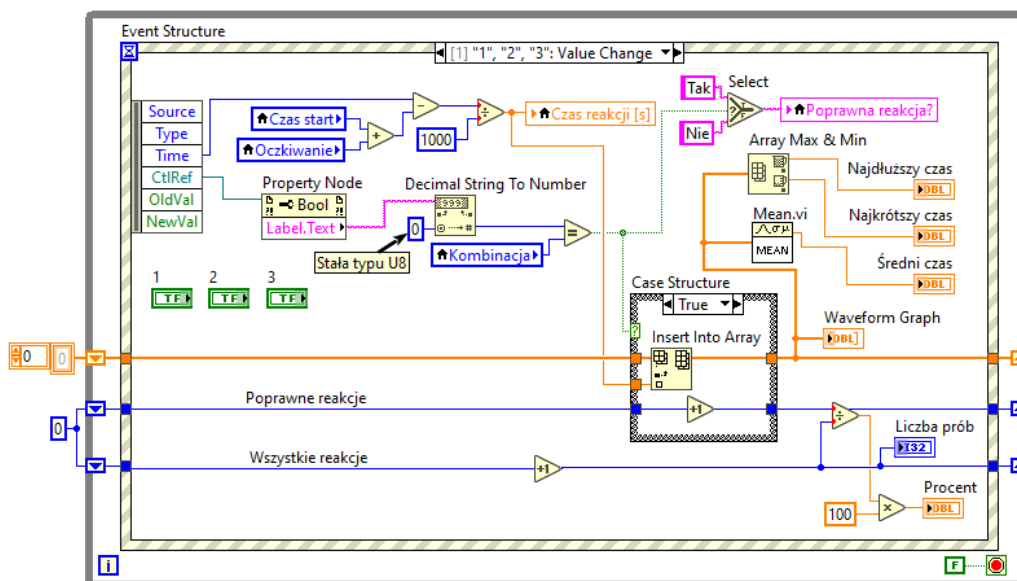
## 2. Statystyka pomiarów czasu reakcji (zadanie do samodzielnej realizacji)

Zwiększyć funkcjonalność programu o statystykę prezentującą wartości: minimalną, maksymalną i średnią czasu reakcji. Obliczyć także wyrażony w % stosunek poprawnych reakcji do wszystkich reakcji. Statystyka powinna być zerowana po uruchomieniu programu oraz po naciśnięciu przycisku *Reset*. Wartości czasu reakcji zaprezentować także na wykresie. Nazwa programu *003\_petlaZdarzenStat\_xx.vi* (gdzie xx inicjały twórcy programu).

**Do realizacji tego zadania można wykorzystać przedstawione poniżej wskazówki.**

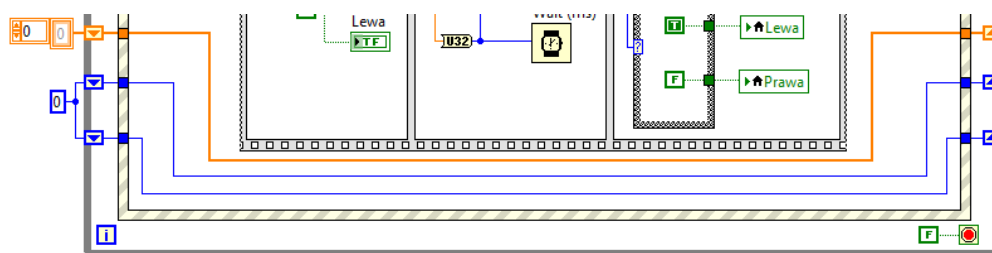
Na rysunku 9 przedstawiono zmodyfikowaną ramkę *Event Structure* przycisków decyzyjnych z dodatkowymi funkcjami.

- Do obliczeń statystycznych wykorzystać tablicę zarejestrowanych czasów reakcji. Do tablicy należy dodawać tylko czasy zmierzone dla poprawnych reakcji. Tablicę zbudować wykorzystując rejestr przesuwający *Shift Register* i funkcję *Insert Into Array* z palety *Array*. Rejestr przesuwający tworzymy klikając PPM na bocznej krawędzi pętli *While Loop* i wybierając opcję *Add Shift Register*.



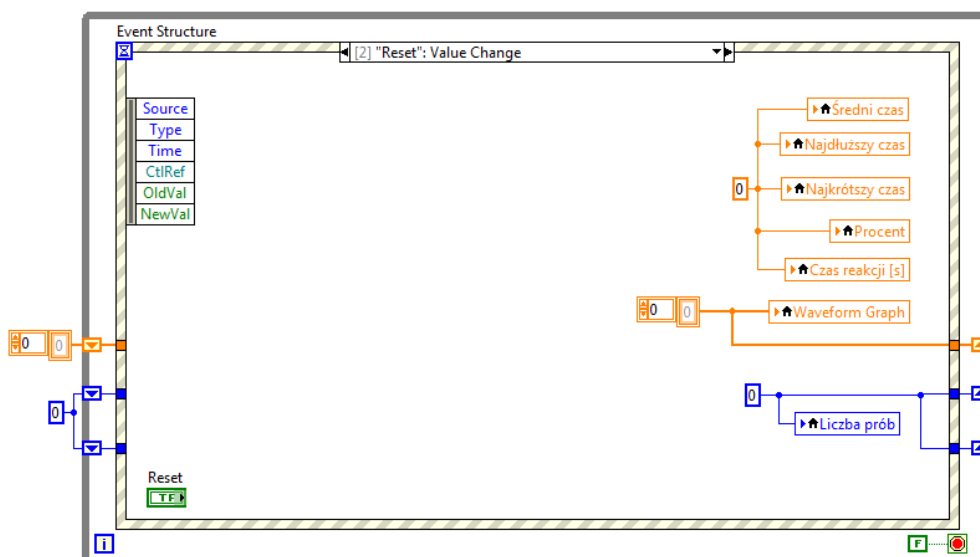
Rys. 9. Zmodyfikowana ramka *Event Structure* przycisków decyzyjnych

- Do wyznaczenia wartości maksymalnej i minimalnej wykorzystać obiekt *Array Max & Min* z palety *Array*.
- Do wyznaczenia wartości średniej wykorzystać obiekt *Mean.vi* z palety *Functions / Mathematics / Probability & Statistics*.
- Do wyznaczenia stosunku poprawnych reakcji do wszystkich reakcji wykorzystać dwa rejestry przesuwające. Rejestr zliczający wszystkie wciśnięcia przycisków decyzyjnych powinien być inkrementowany przy każdym wykonaniu ramki, a rejestr poprawnych reakcji tylko po wciśnięciu odpowiedniego przycisku decyzyjnego.
- W strukturze *Case Structure* w ramce *False* (ramka niewidoczna na rys. 9) dane z wejścia są bezpośrednio przepisywane na wyjście.
- Podobnie należy bezpośrednio ze sobą połączyć wejścia i wyjścia rejestrów w ramce *Event Structure* obsługującej naciśnięcie przycisku *Start* (rys. 10).



Rys. 10. Połączenia rejestrów w ramce obsługującej naciśnięcie przycisku *Start*

- Na panelu użytkownika umieścić *Waveform Graph* z palety *Graph*.
- Do resetowania danych statystycznych umieścić na panelu użytkownika dodatkowy przycisk z palety *Boolean*, nadać mu etykietę *Reset*.
- W diagramie programu utworzyć dodatkową ramkę struktury *Event Structure* do obsługi wciśnięcia przycisku *Reset*. Wygląd tej ramki pokazano na rysunku 11.
- Zapisać program nadając mu nową nazwę *003\_Statystka\_xx.vi* następnie uruchomić program i sprawdzić jego działanie.



Rys. 11. Ramka *Event Structure* obsługująca przycisk *Reset*

### 3. Zawartość sprawozdania

Sprawozdanie z ćwiczeń powinno zawierać:

- pliki z programami opracowanymi podczas zajęć (w formie elektronicznej),
- podpisane zrzuty ekranów uruchomionych programów z krótkim opisem,
- krótki opis nowych poznanych w trakcie ćwiczenia obiektów z palety *Functions*,
- uwagi i wnioski.