Wykład 7
20.11.2026 r.
Struktura Event

Katarzyna Gwóźdź

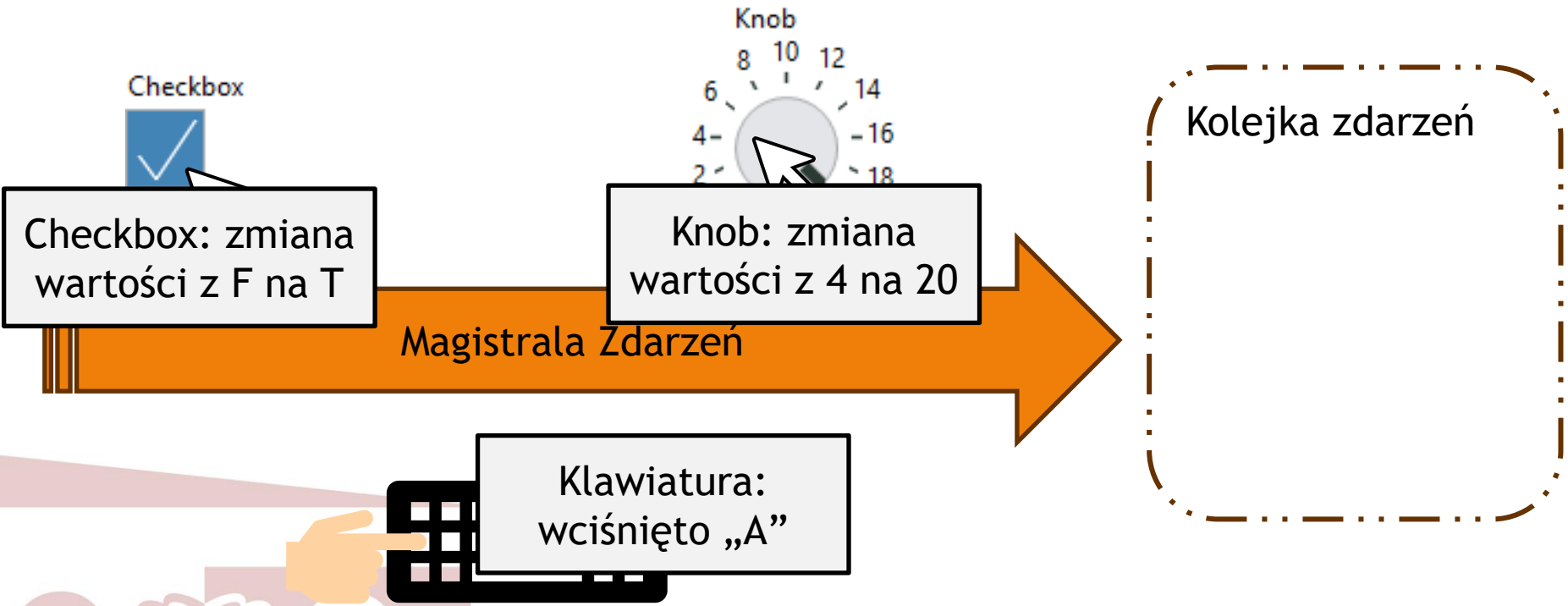
Event - wydarzenie

Niesynchroniczna notyfikacja, że coś się wydarzyło.

Źródła eventów:
Interfejs użytkownika,
Urządzenia wejściowe/wyjściowe,
Inne elementy programu.

Przykłady eventów:
Zmiana wartości kontrolki,
Naciśnięcie guzika,
Naciśniecie przycisku myszy,
Zamknięcie programu, i inne…

Event - wydarzenie



Event - źródła

Obsługiwane przez LabVIEW

- Zmiana wartości kontrolki
- Kliknięcie na przycisk
- Zmiana rozmiaru okna

Interfejs programowy



- Kliknięcie myszą
- Poruszenie kursorem
- Naciśnięcie przycisku na klawiaturze

Interfejs sprzętowy



- Programowe wywołanie zdarzenia w jednej pętli, by zatrzymać inną pętlę

Program



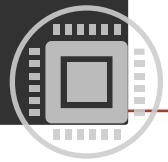
- Zmiana konfiguracji ekranu
- Zlecono wyłączenie komputera

System operacyjny



- Podłączono głośniki
- Odłączono dysk USB

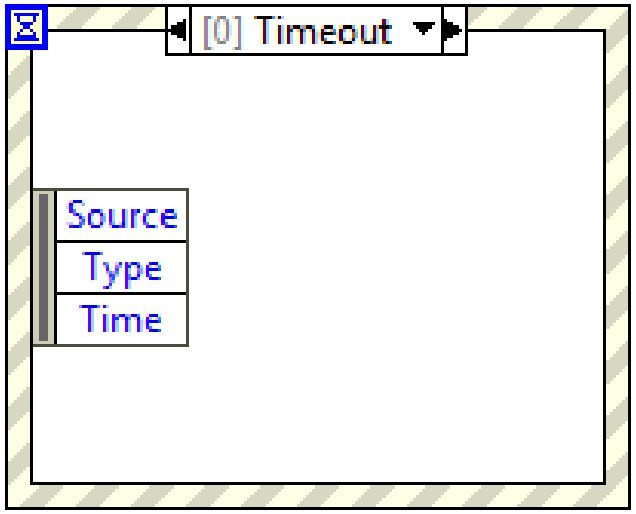
Sprzęt



Struktura Event

Struktura w Labview do obsługi eventów.

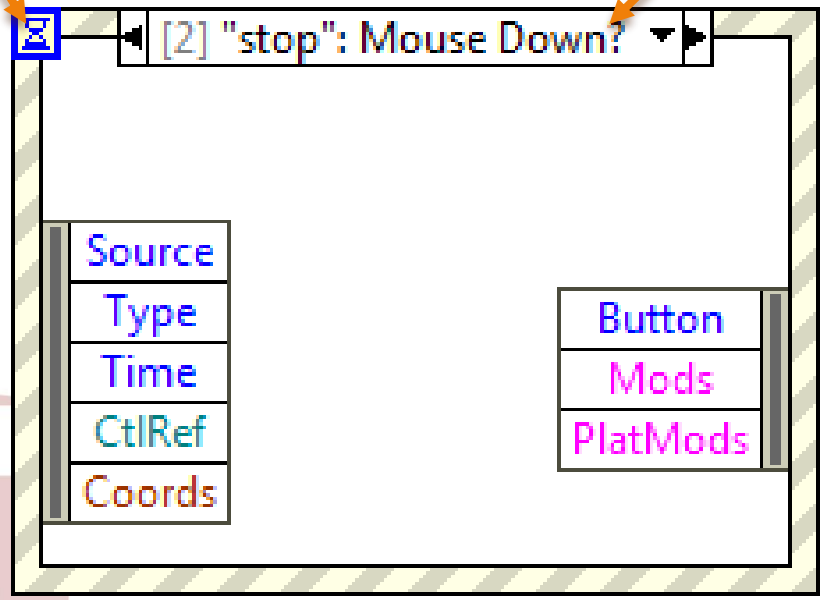
- Wykonywanie kodu uzależnione od występowania wydarzeń.
- Kontroluje wydarzenia w czasie.
- Tylko jedno wydarzenie może nastąpić w tym samym momencie.
- Każdy case kontroluje inne wydarzenie (jedno lub kilka o tym samym typie danych).



Struktura Event

Timeout – czas oczekiwania na wydarzenie, wartość domyślna (-1) to oczekiwanie bez końca.

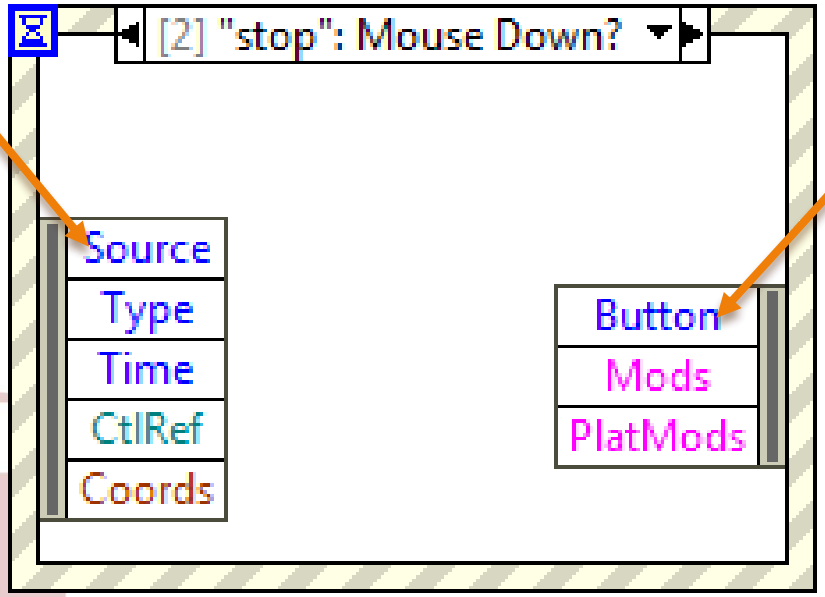
Selektor struktury Event – pokazuje który case wydarzeń obsługujemy.



Struktura Event

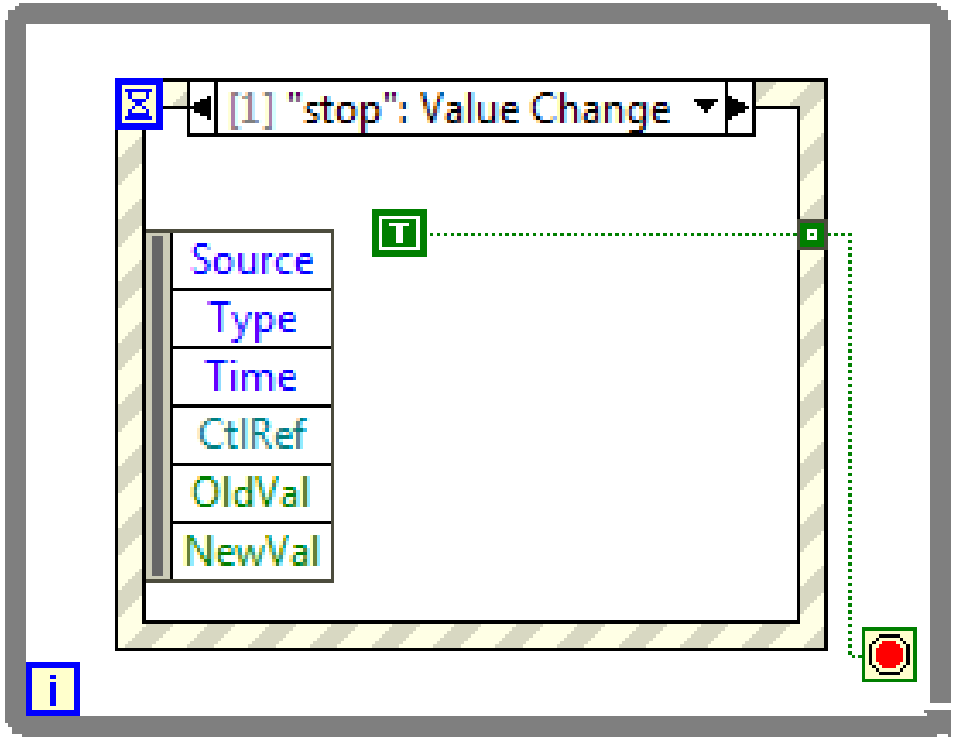
Event Data – wyświetla parametry wydarzenia, np. czas wydarzenia, współrzędne kursora.

Event Filter – pozwala nadpisywać parametry zdarzenia

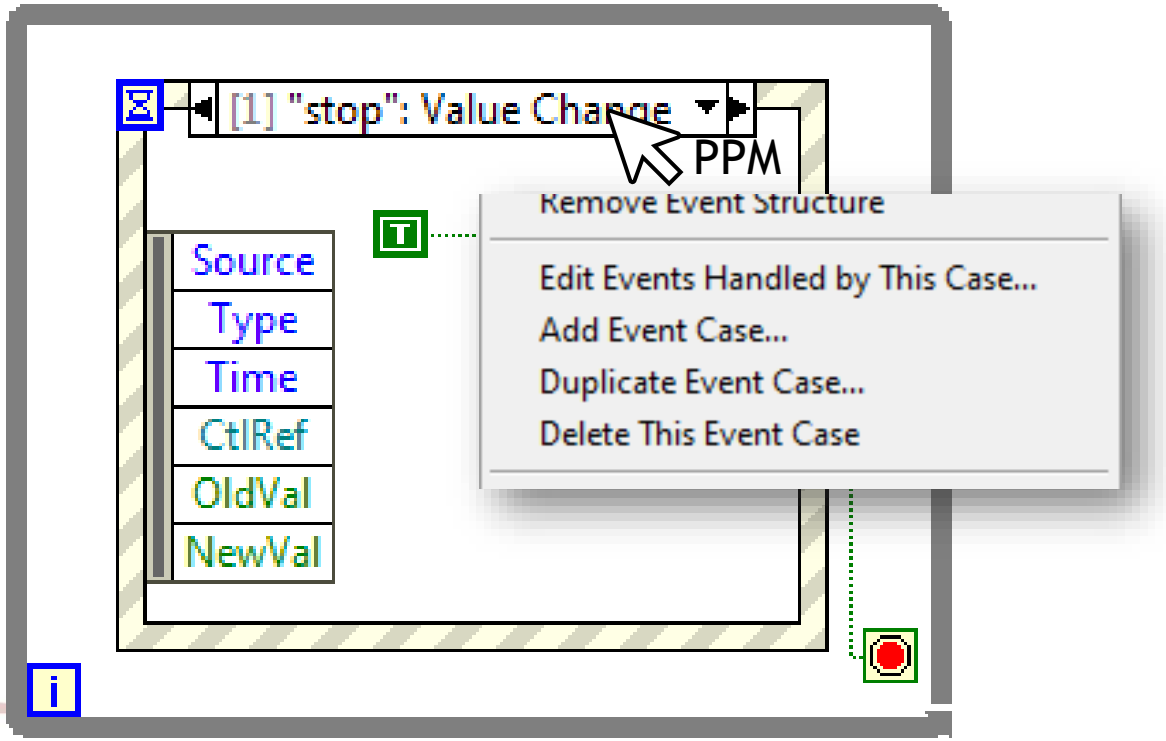


Implementacja struktury Event

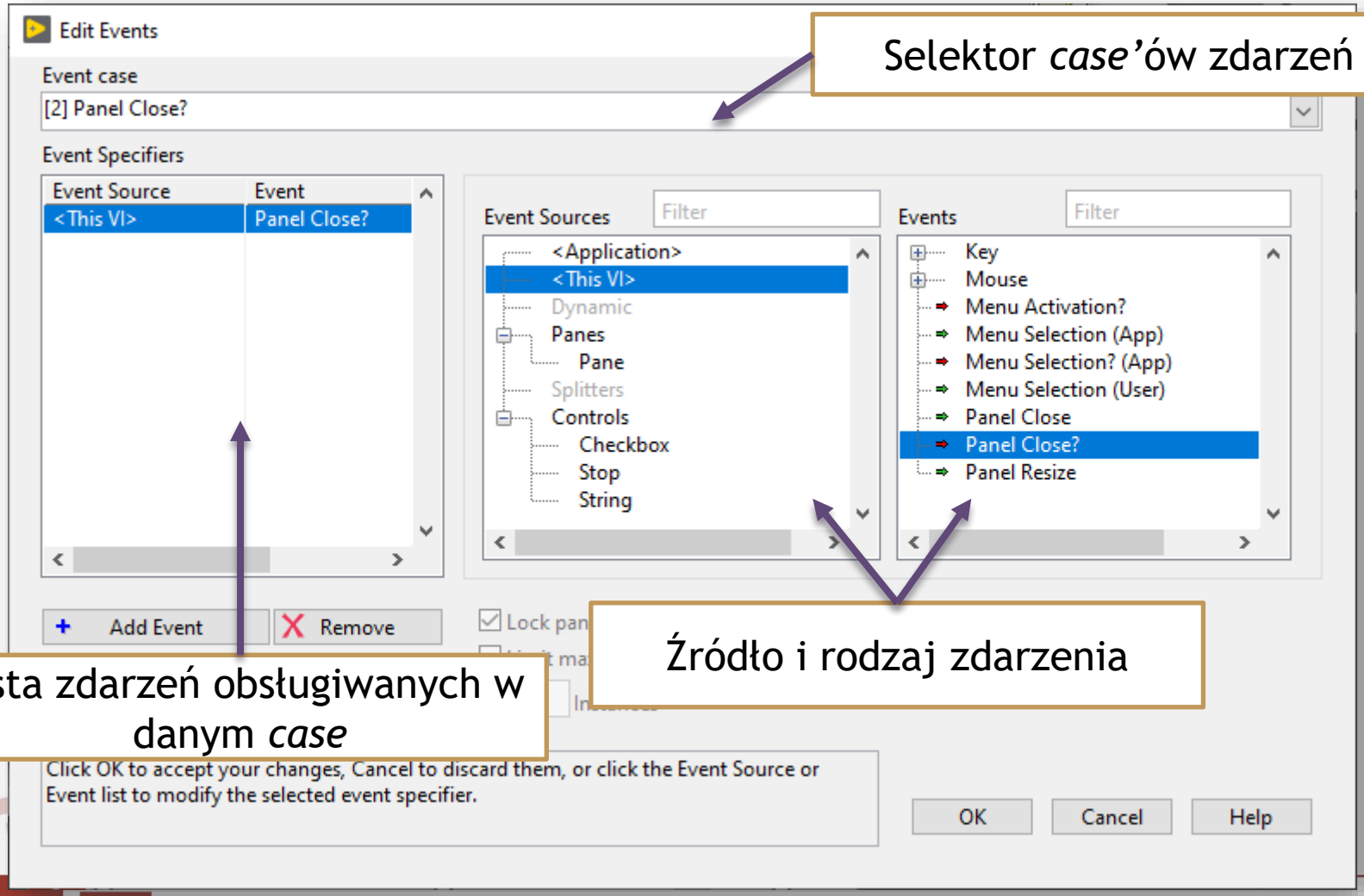
- Zwykle umieszczamy strukturę Event w pętli While.
- Podczas jednej iteracji obsługiwane jest jedno wydarzenie.
- Gdy nie występują żadne wydarzenia, nie są wykonywane żadne działania, program czeka.



Dodawanie wydarzeń



Dodawanie wydarzeń

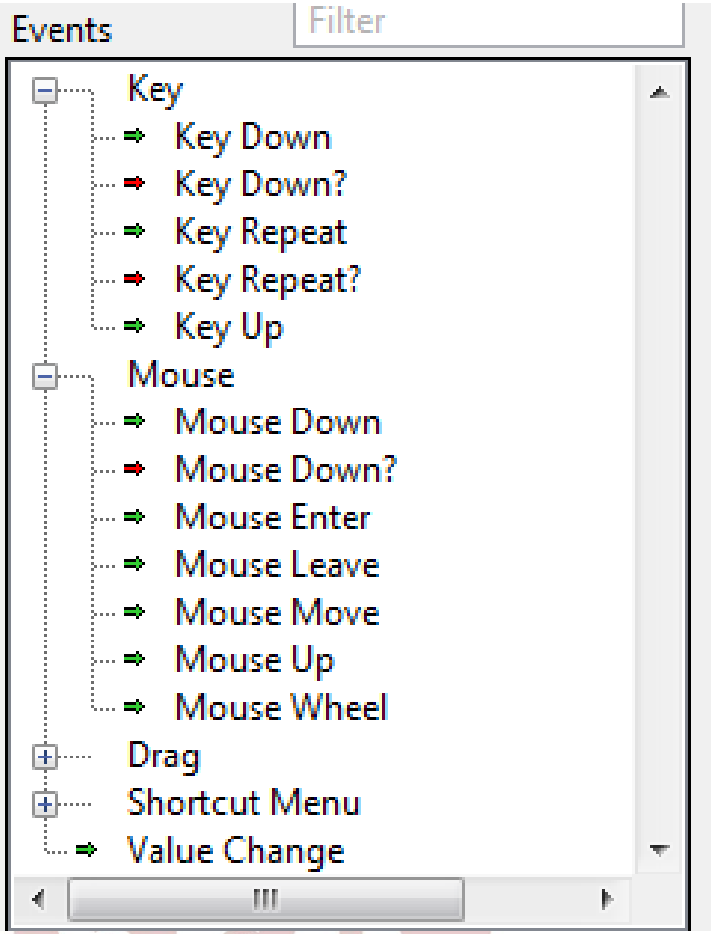


Selektor *case*'ów zdarzeń

Lista zdarzeń obsługiwanych w danym *case*

Źródło i rodzaj zdarzenia

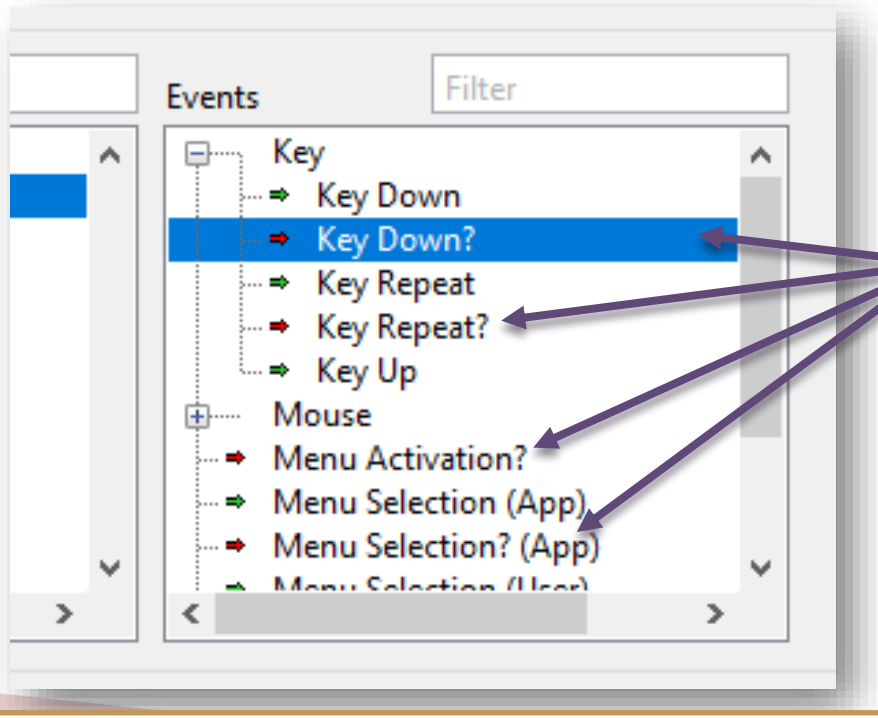
Rodzaje wydarzeń



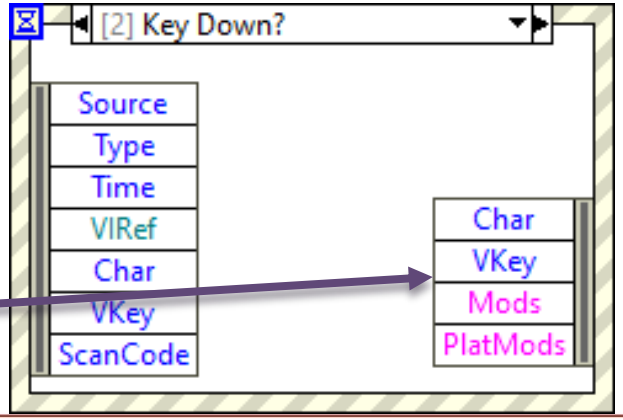
Powiadomienie o wydarzeniu: użytkownik wykonał akcję i wydarzenie zostało przetworzone przez Labview.

Filtrowane wydarzenia: użytkownik wykonał akcję, Labview nie przetworzył wydarzenia, można nadpisywać domyślne akcje.

Rodzaje wydarzeń



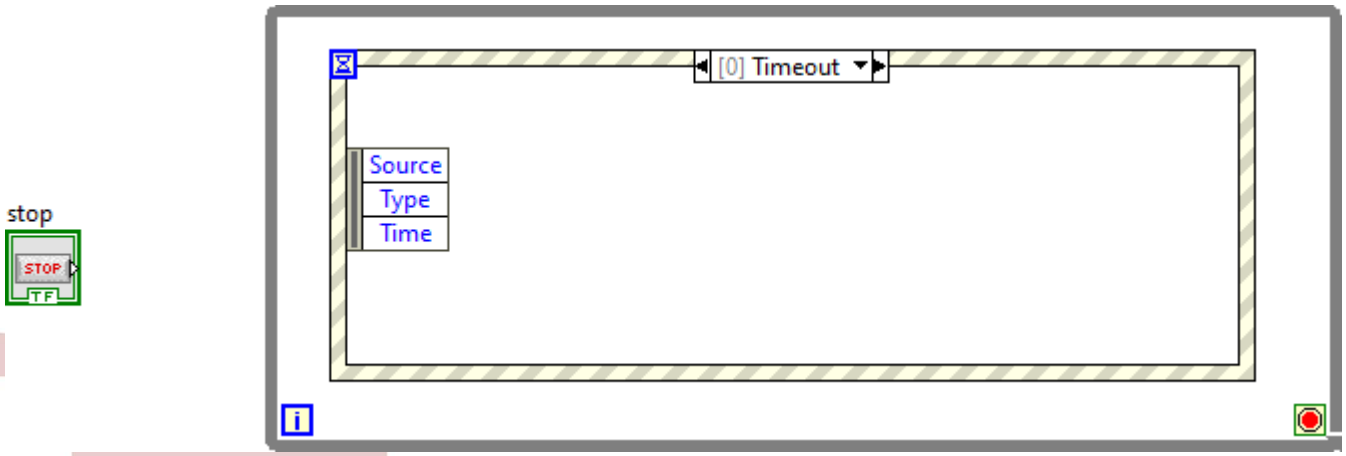
Zdarzenia z filtrem oznaczone są czerwoną strzałką i pytajnikiem ➔…?



Akcja użytkownika wywoła wykonanie kodu w odpowiednim case. Do dalszego przetwarzania akcji wykorzystane zostaną dane z *Event Filter*

Przykład - Obsługa przycisku STOP przy użyciu struktury event.

1. Tworzymy strukturę event w pętli while oraz przycisk STOP.

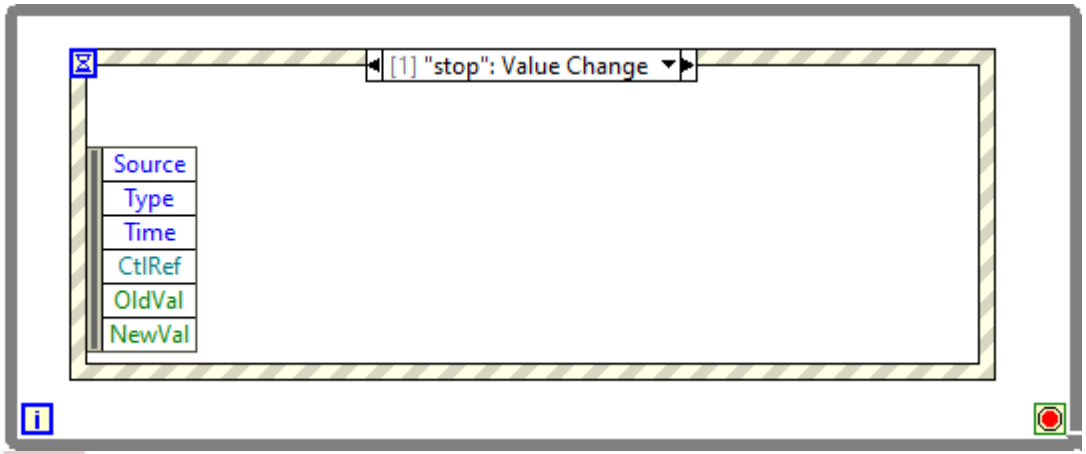


Przykład - Obsługa przycisku STOP przy użyciu struktury event.

2. Tworzymy event obsługujący przycisk stop. PPM -> Add event case. Jako źródło eventu wybieramy przycisk STOP. Jako event wybieramy zmianę wartości przycisku.

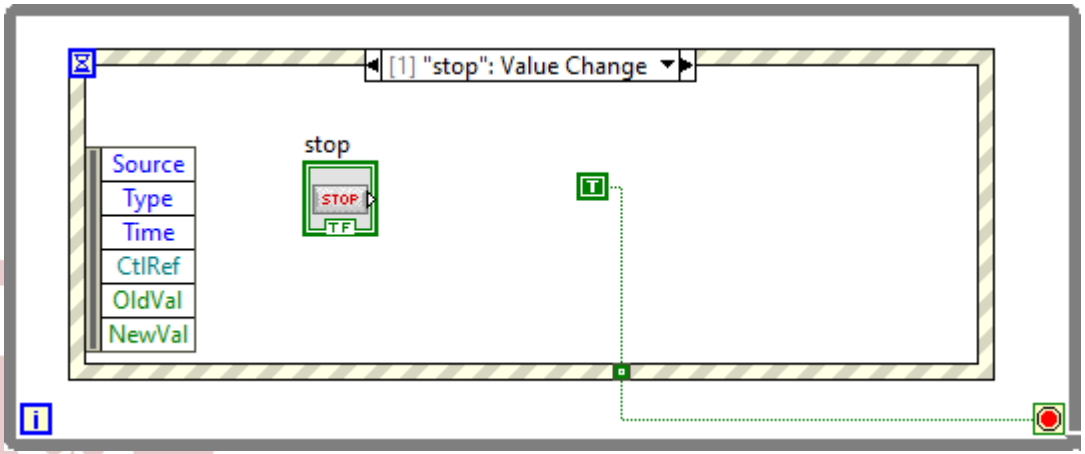
Po wciśnięciu przycisku STOP zostanie wykonana zawartość case „stop”: Value Change. W tym przypadku nie interesuje nas czy przycisk jest w stanie T/F, tylko fakt zmiany tej wartości.





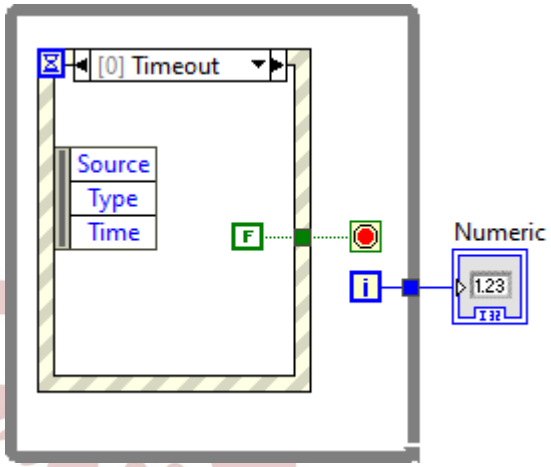
Przykład - Obsługa przycisku STOP przy użyciu struktury event.

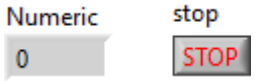
3. Za każdym razem, gdy przycisk zostanie wciśnięty program ma się zakończyć. W każdym innym casie należy podłączyć false.



Przykład - Obsługa przycisku STOP przy użyciu struktury event.

4. W tym przypadku nic nie zostało podłączone do Timeout – program czeka w nieskończoność na wydarzenie. Oznacza to, że pętla while nie wykonuje się, jest zawieszona do momentu, gdy pojawi się event – w tym przypadku naciśnięci przycisku stop. Iterator pętli zawsze wskazuje 0, ponieważ pętla zawsze wykona się tylko 1 raz.





Zadanie 1

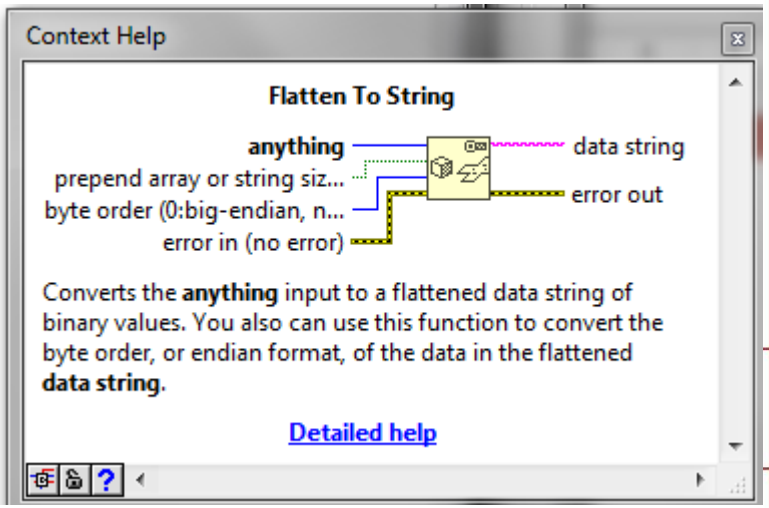
Napisz program, w którym naciśnięcie przycisku na front panelu spowoduje wylosowanie liczby z zakresu 0-28 i wyświetlenie jej w indykatorze.

Zadanie 2

Napisz program, który będzie śledził ruch kursora i wyświetlał jego współrzędne na wykresie wykorzystując strukturę event.

Zadanie 3

Napisz program, który będzie wyświetlał wszystkie naciśnięte klawisze na klawiaturze (można ograniczyć się do liter i cyfr) w indykatorze typu string przy wykorzystaniu struktury event. Zaprogramuj również możliwość resetowania tego indykatora oraz usuwania ostatniego znaku używając backspace.



Zadanie 4

Napisz program, który przy próbie zamknięcia programu za pomocą x wyświetli komunikat proszący o wciśnięcie przycisku stop.

