

Komputerowe wspomaganie eksperymentu

5

Dr Piotr Sitarek

Katedra Fizyki Doświadczalnej, Politechnika Wrocławska

Temat na dziś

Kontrola urządzeń – klawiatura, mysz

Programowanie w środowisku (cd.)



ni.com

(część materiałów zaczerpnięta ze strony producenta)

Kontrola urządzeń

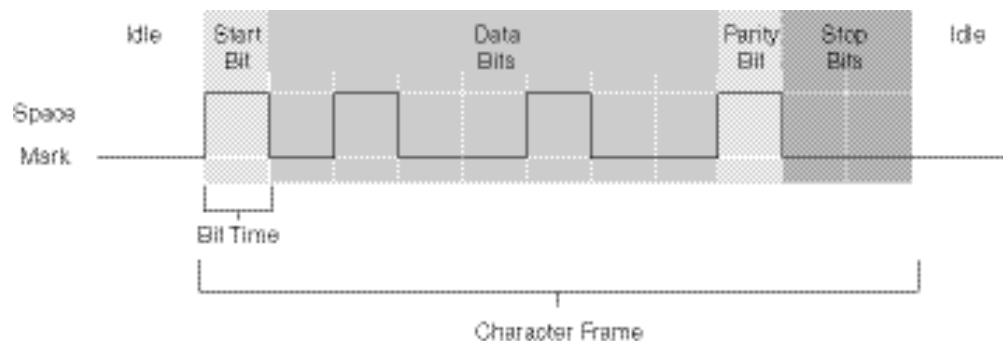
- GPIB
 - Serial
 - Modular Instruments
 - PXI Modular Instruments
 - Image Acquisition
 - Motion Control
 - USB
 - Ethernet
 - Parallel Port
-

GPIB

- General Purpose Interface Bus (GPIB)
- Interfejs (karta) GPIB zwykle używany jest do kontroli urządzeń zewnętrznych i akwizycji danych
- Cyfrowy 8-bitowy równoległy interfejs komunikacyjny
- IEEE 488.1 i 488.2 określają standardy dla GPIB
- Szybki transfer danych – 1MB/s lub więcej

Port szeregowy - serial

- Przesyła 1 bit na raz
- Zwykle nie jest potrzebne dodatkowe oprzyrządowanie – wbudowany w komputer (RS232 – najpopularniejszy)
- 4 parametry: szybkość transmisji (baud rate), rozmiar danych (data bits), parzystość (parity bit), stop bits



MAX

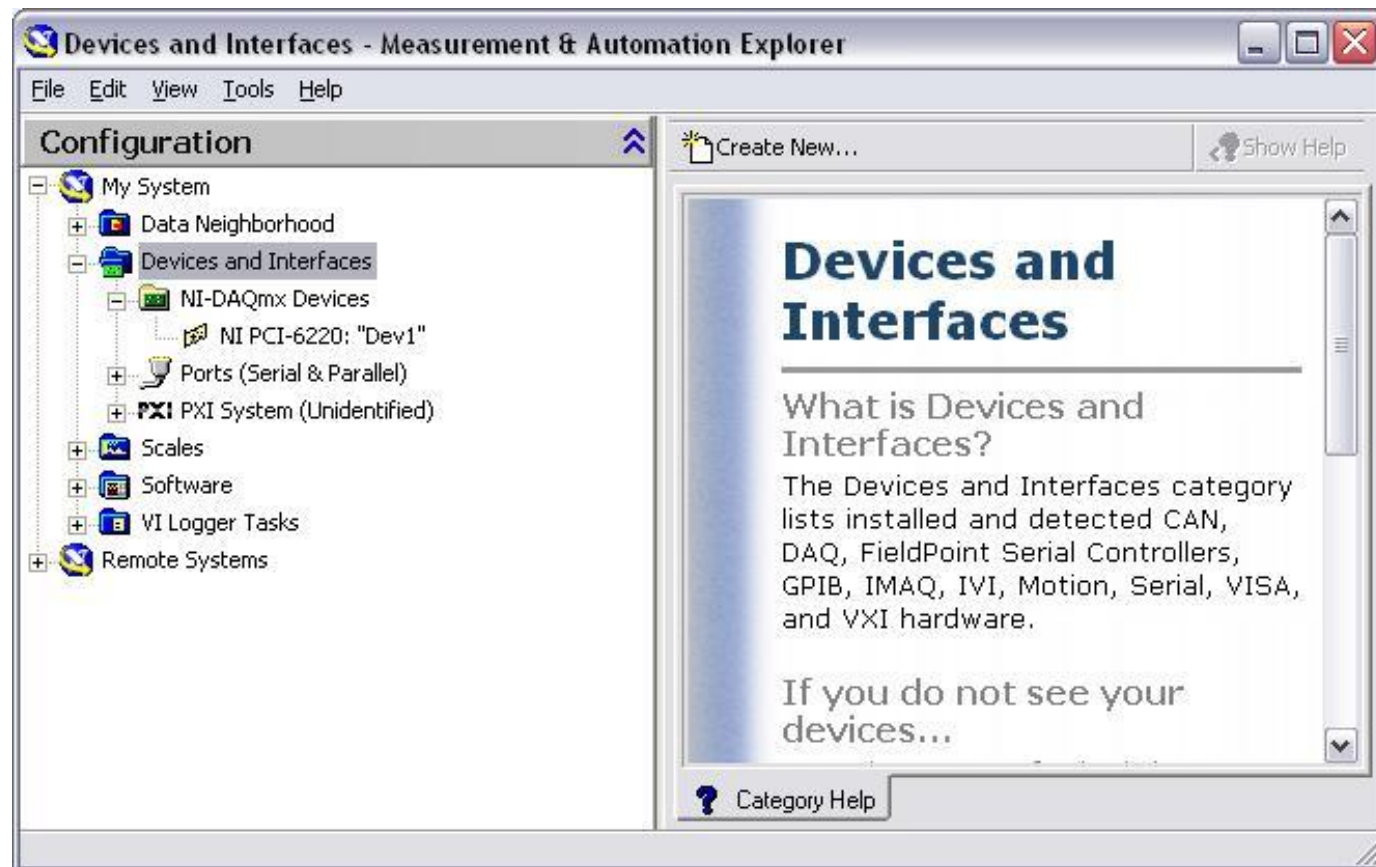
- MAX - Measurement & Automation Explorer
- MAX służy do konfiguracji i organizacji wszelkiego rodzaju urządzeń, np: NI DAQ, PCI/PXI, GPIB, IMAQ, IVI, Motion, VISA, VXI (także symulacji urządzeń)
- Symulacja urządzeń
- Testy



Measurement
& Automation

MAX

Dodawanie i konfigurowanie urządzeń



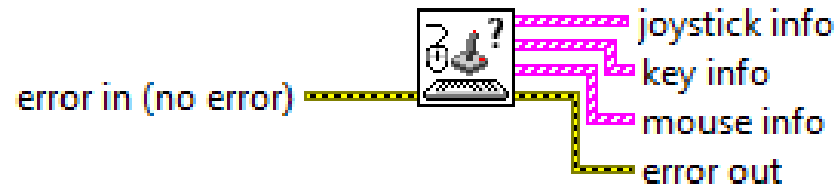
Data Acquisition Device - DAQ

Symulacja urządzenia na poziomie sterowników,

- generacja,
 - filtrowanie,
 - itp.
-

Klawiatura, mysz, pad

Query Input Devices.vi



Obtains information about the devices connected to the computer.

Initialize Keyboard.vi [Intialize Keyboard.vi]



Opens a reference to and initializes the keyboard connected to the computer.

Initialize Mouse.vi



Opens a reference to and initializes the mouse connected to the computer.

Initialize Joystick.vi



Opens a reference to and initializes a joystick device at the index you specify.

Acquire Input Data.vi



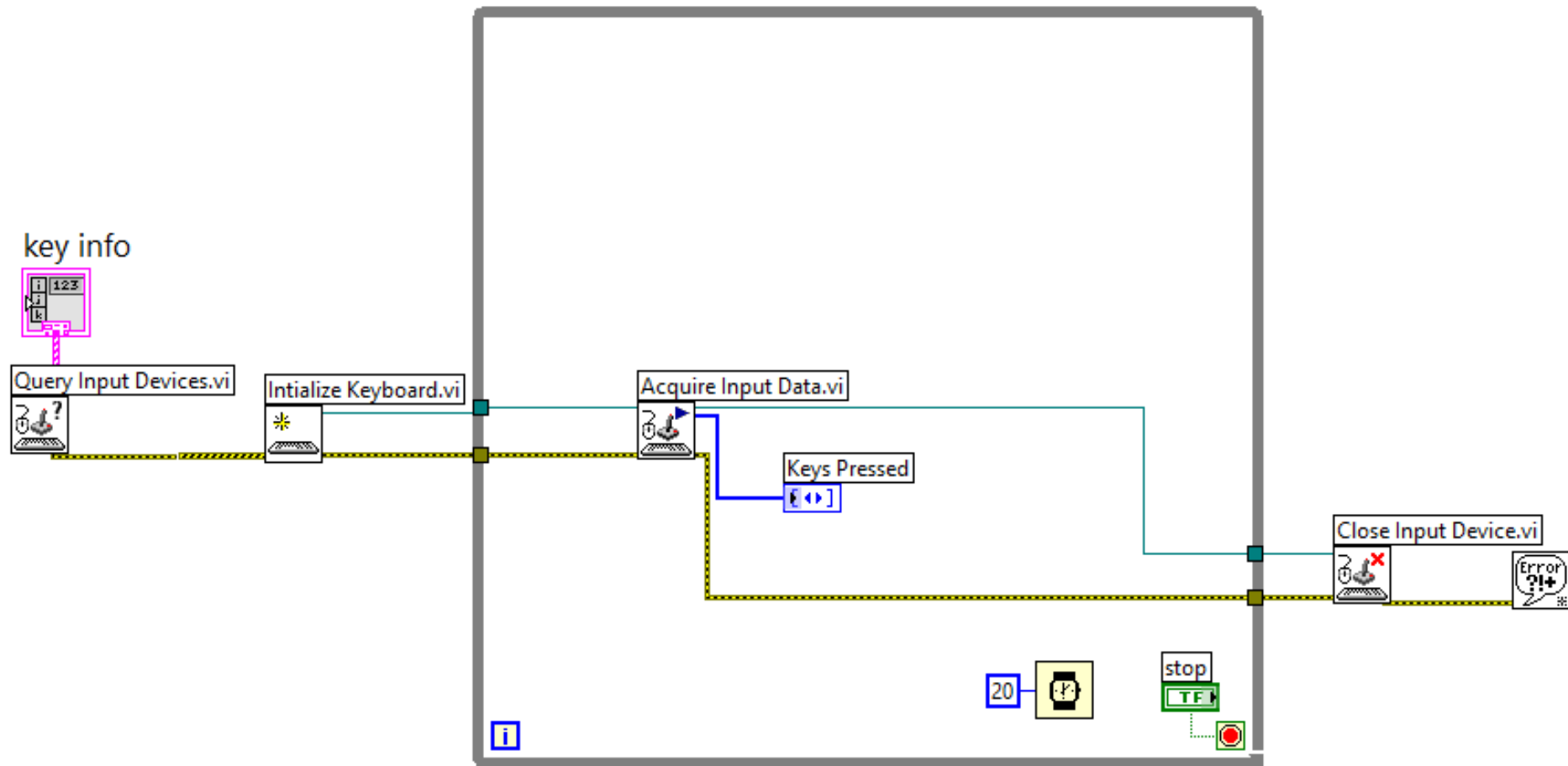
Returns data about the device connected to the computer. Wire data to the **device ID** input to determine the polymorphic instance to use or manually select the instance.

Close Input Device.vi

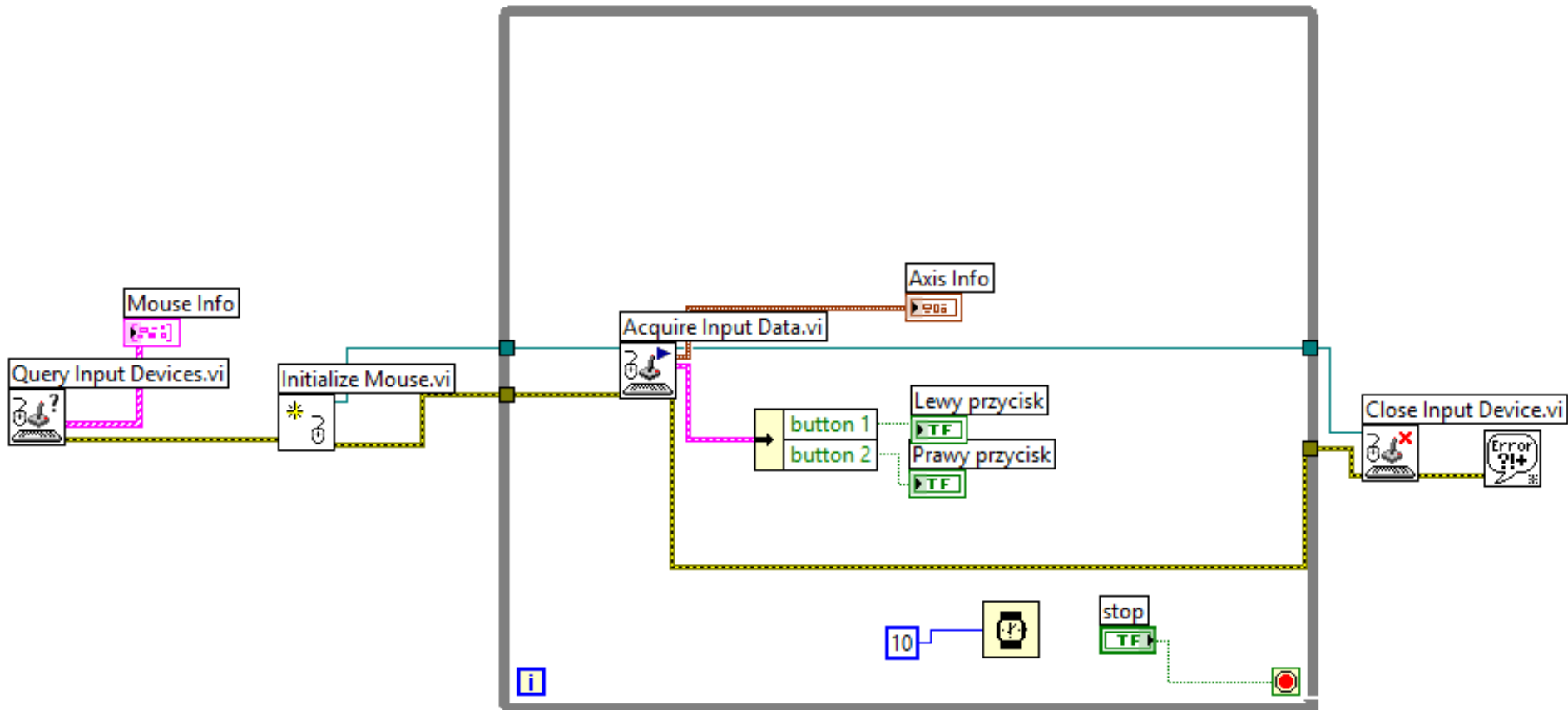


Closes the device you specify in **device ID**. Wire data to the **device ID** input to determine the polymorphic instance to use or manually select the instance.

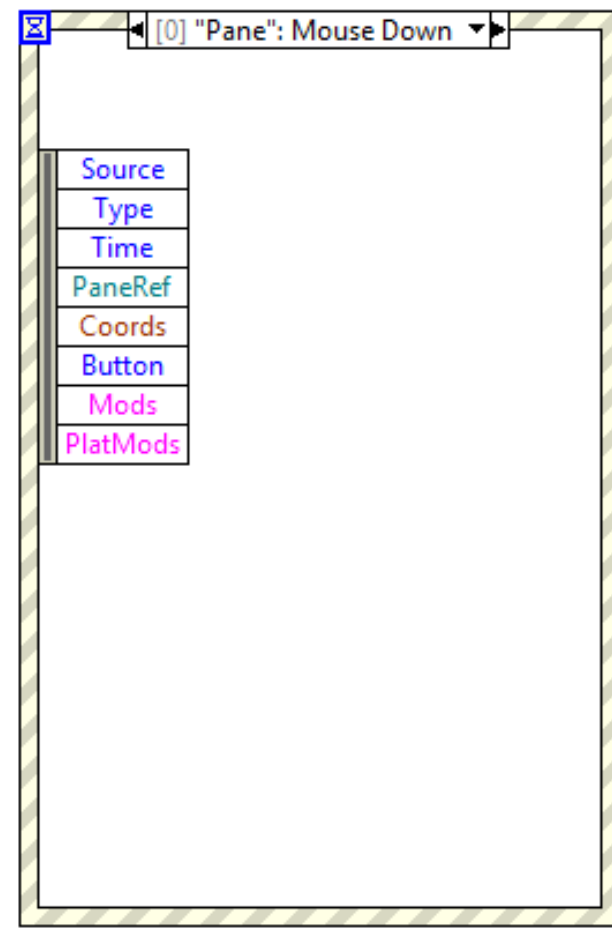
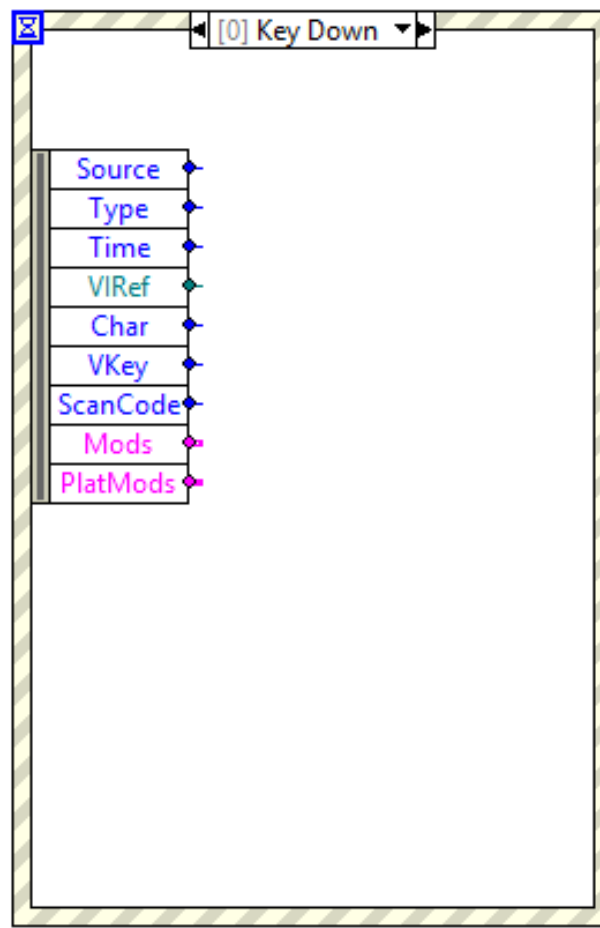
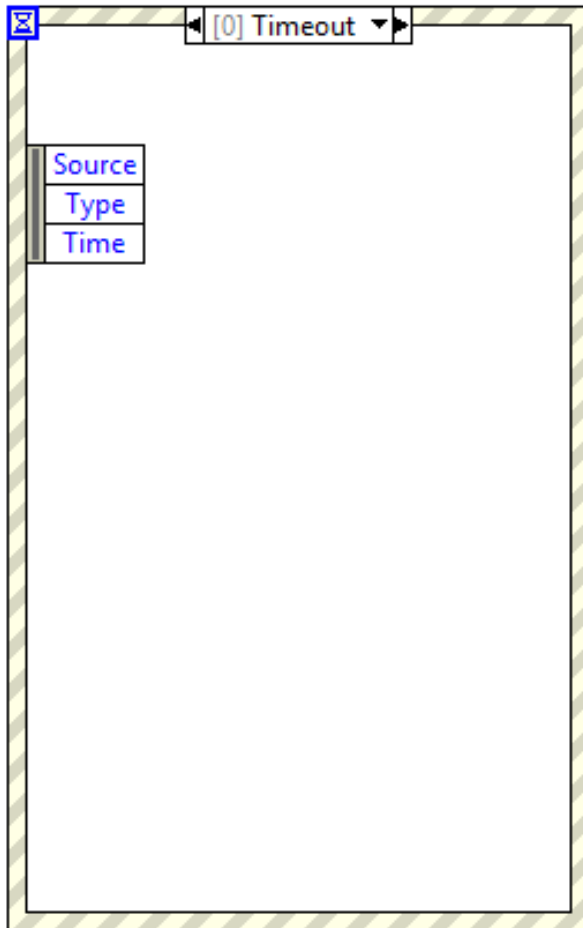
Klawiatura, mysz, pad



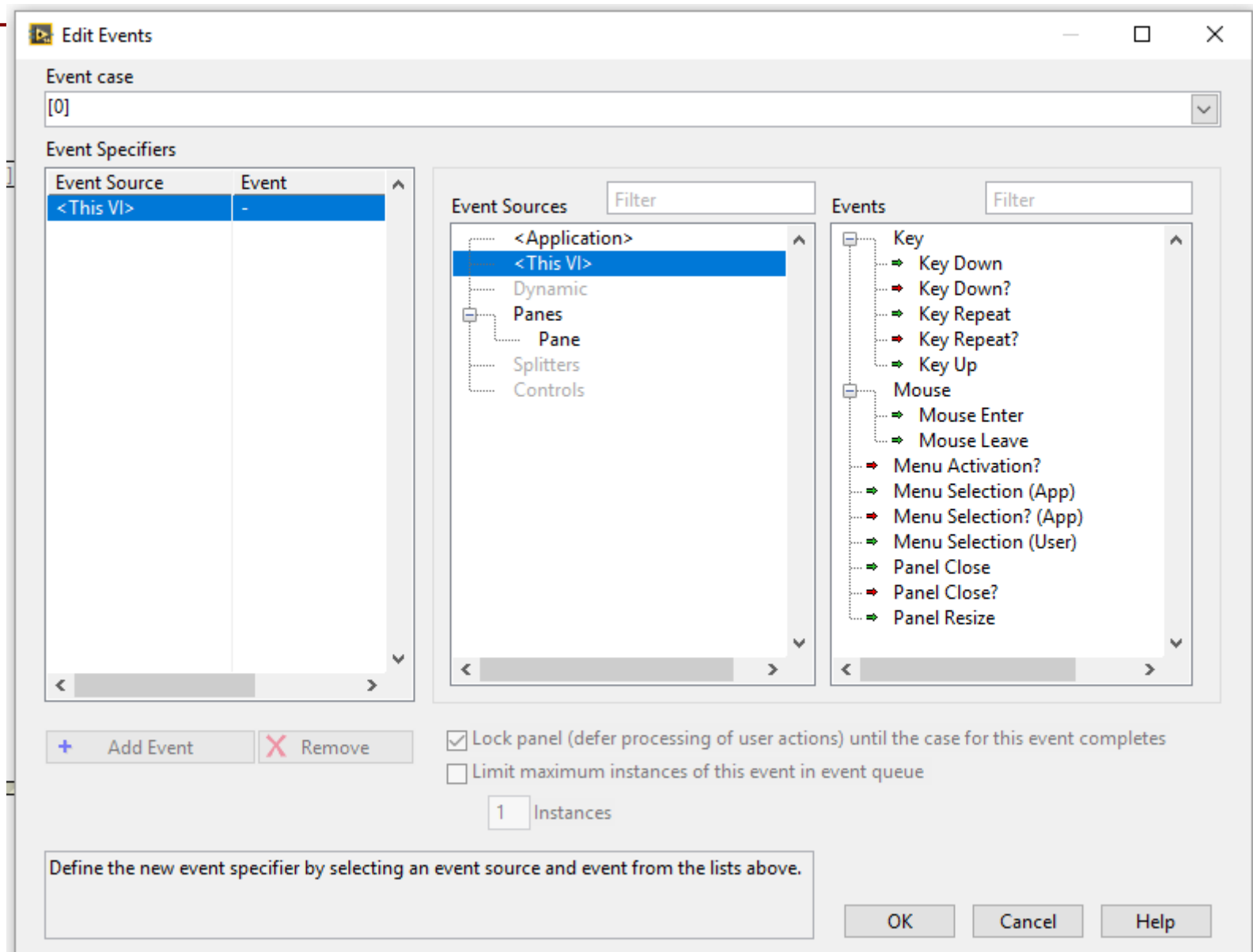
Klawiatura, mysz, pad



Struktura „event”



Struktura „event”



Struktura „event”

Edit Events

Event case
[0] "Pane":

Event Specifiers

Event Source	Event
Pane	-

Event Sources

- <Application>
- <This VI>
- Dynamic
- Panes
 - Pane
- Splitters
- Controls

Events

- Mouse
 - Mouse Down
 - Mouse Down?
 - Mouse Enter
 - Mouse Leave
 - Mouse Move
 - Mouse Up
 - Mouse Wheel
- Shortcut Menu
- Pane Size

Struktura „event”

Pane Events

Requires: Base Development System

View the [class hierarchy](#).

Event	Description
Mouse Down	Generated when you click the mouse button on a pane. Details
Mouse Down?	Generated when you click the mouse button on a pane. Details
Mouse Enter	Generated when the cursor enters the bounds of the pane. Details
Mouse Leave	Generated when the cursor leaves the bounds of the pane. Details
Mouse Move	Generated when you move the mouse over a pane. Details
Mouse Up	Generated when you release the mouse button on a pane. LabVIEW does not generate this event if a shortcut menu appears when you click the mouse button. Details
Mouse Wheel	Generated when you scroll the mouse wheel over a pane. Details
Pane Size	Generated when the user sizes the pane by clicking and dragging the window frame, clicking and dragging a splitter bar, maximizing the front panel, or when the user restores the pane to its original size from a maximized state. Details
Shortcut Menu Activation?	Generated when the user right-clicks blank space on the front panel to display the shortcut menu. Details
Shortcut Menu Selection (User)	Generated when the user selects a user-defined shortcut menu item from the shortcut menu of the pane. Details

State Machine

Maszyna stanów – State Machine

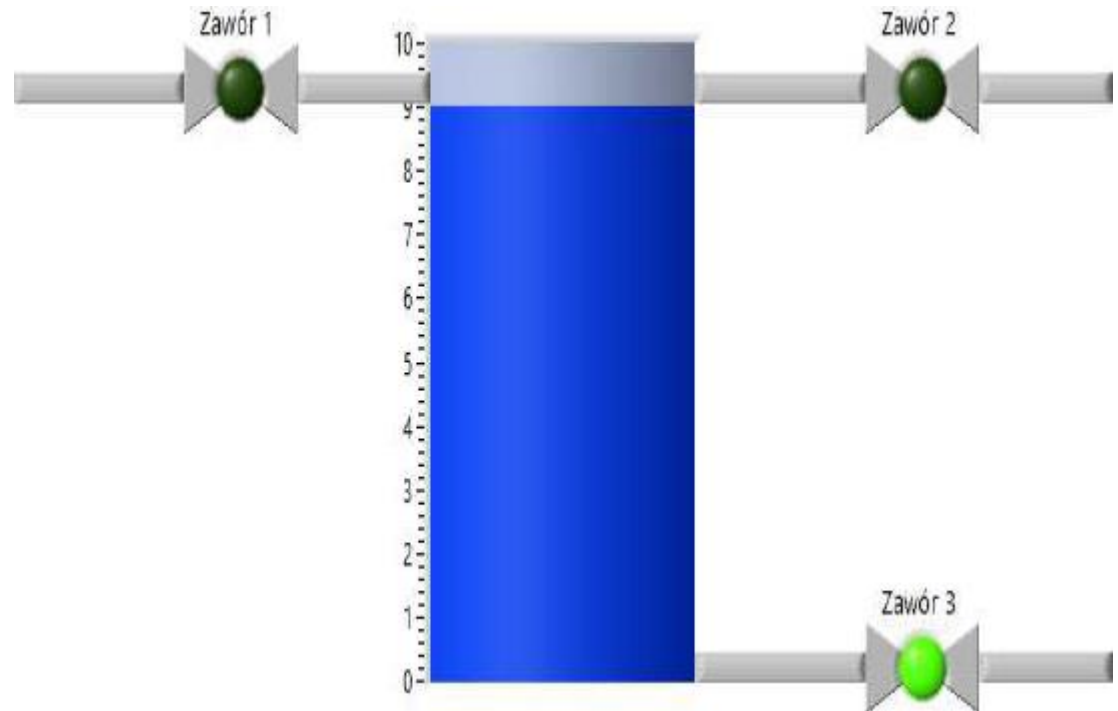
Diagram maszyny stanowej – jest znacznie większą realizacją matematycznej koncepcji automatu skończonego w zastosowaniach informatyki wchodzącej w skład języka UML. Diagram opisuje szereg stanów przez jakie pod wpływem interakcji może przejść obiekt w systemie informatycznym. Pokazuje możliwe stany obiektu, od stanu początkowego do stanu końcowego oraz przejścia, które powodują zmianę tego stanu. (wikipedia)

State Machine

Maszyna stanów jest używana w aplikacjach, w których istnieją rozróżnialne stany.

- Każdy stan może prowadzić do innego- pojedynczego lub wielokrotnego stanu lub może zakończyć przebieg procesów.
- Zasada działania maszyny stanów polega na wprowadzaniu danych przez użytkownika lub obliczeń wewnątrz aktualnego stanu w celu określenia do jakiego stanu przejść.
- Następny stan procesu jest przechowywany i przesyłany do kolejnego kroku procesu za pomocą rejestrów przesuwnych (Shift Register) lub węzła Feedback Node.

State Machine



1. Otworzyć zawory 1 i 2
2. Napęlnić zbiornik
3. Zamknąć zawory 1 i 2
4. Otworzyć zawór 3
5. Opróżnić zbiornik
6. Zamknąć zawór 3
7. Powtarzać proces do naciśnięcia klawisza STOP

State Machine

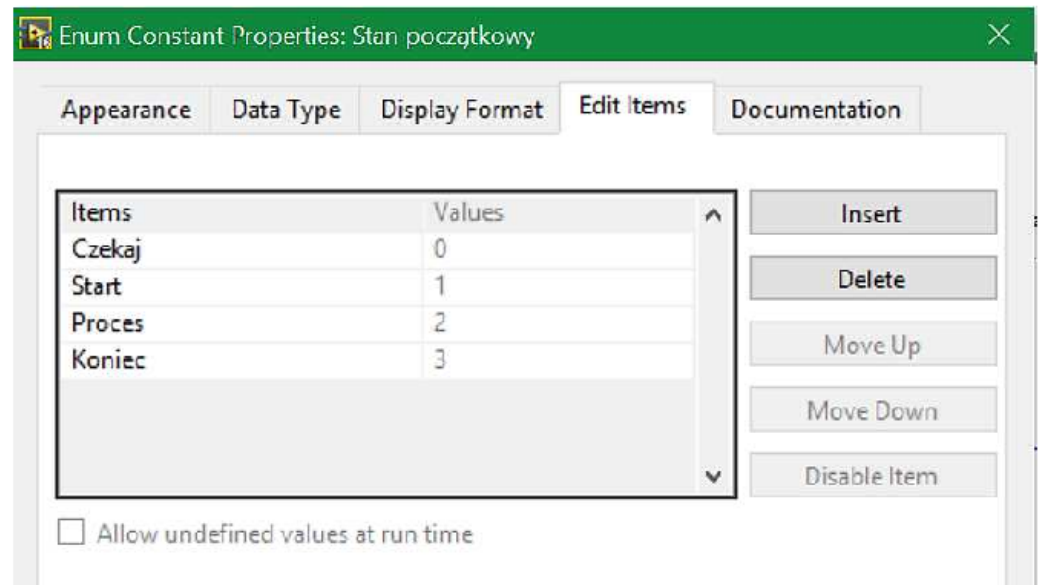
Należy zdefiniować wszystkie stany występujące w całym procesie (np. kontrolka **Enum**):

Stan początkowy

• Czekaj ▾

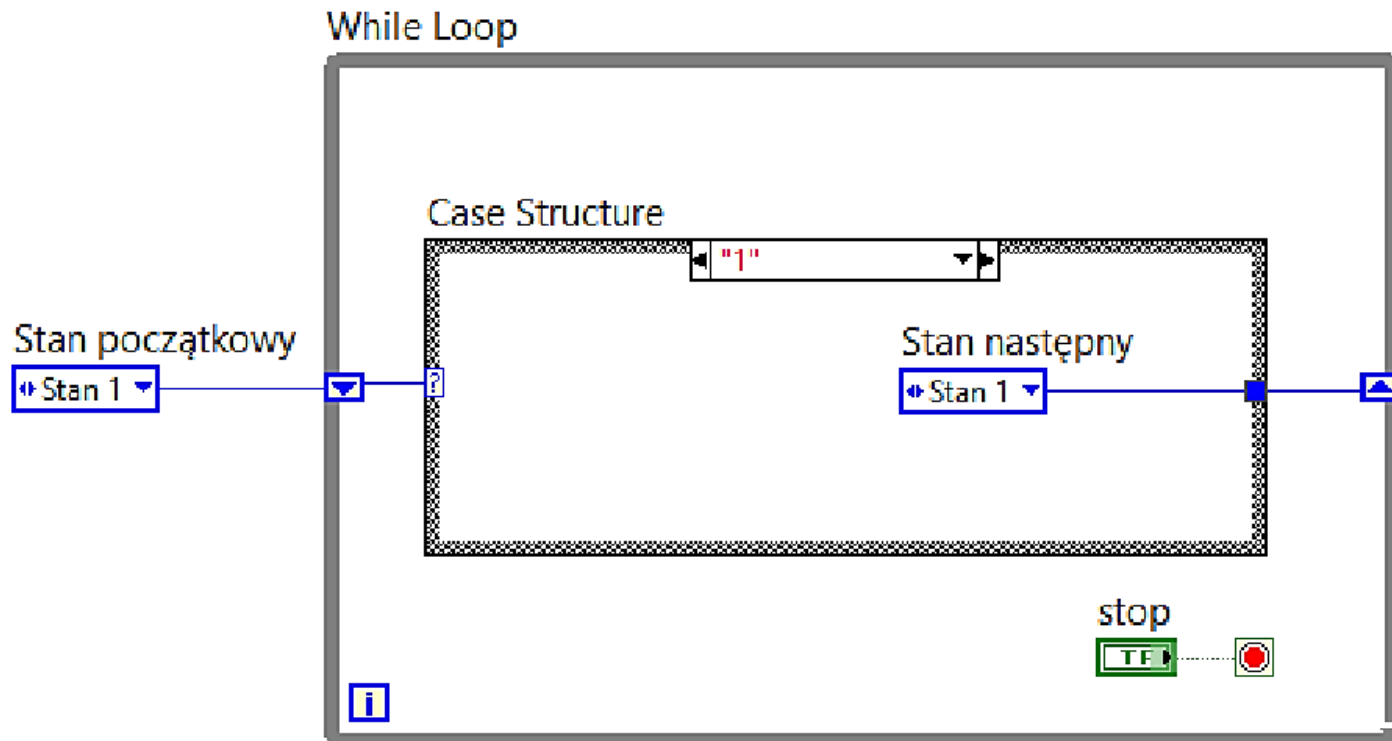
Stan początkowy

- ✓ Czekaj
- Start
- Proces
- Koniec



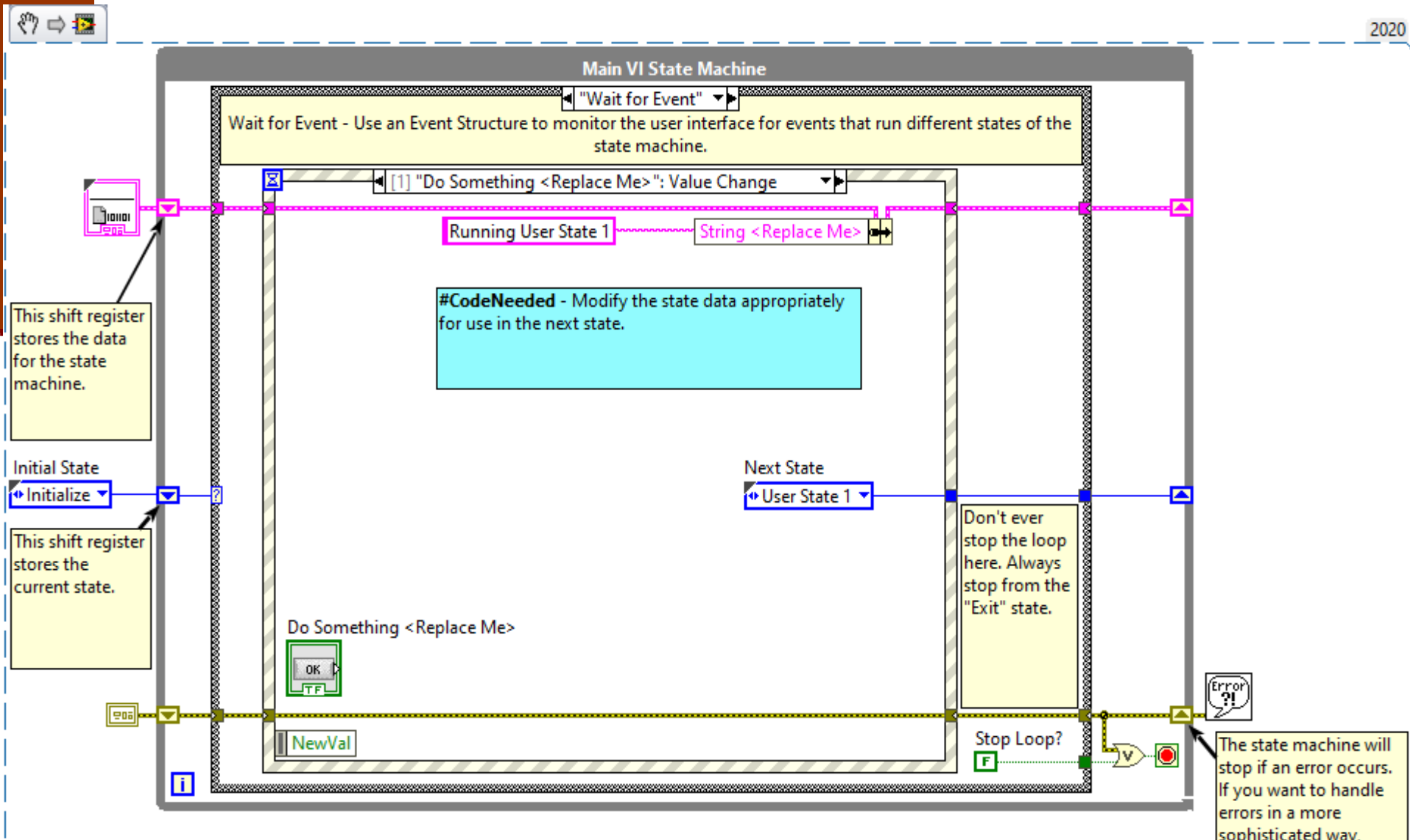
State Machine

Struktura podstawowa



State Machine

2020



Pad

Przyciski
kierunkowe

Przyciski akcji

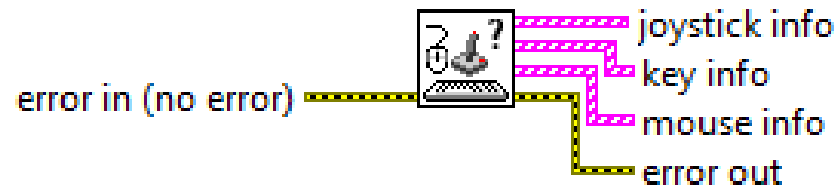
Gałki analogowe



Urządzenie sterujące wykorzystywane w grach komputerowych.

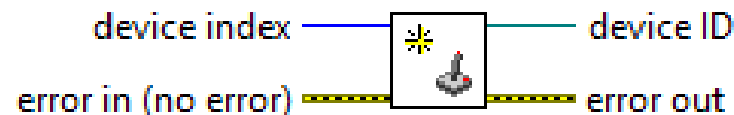
Pad

Query Input Devices.vi



Obtains information about the devices connected to the computer.

Initialize Joystick.vi



Opens a reference to and initializes a joystick device at the index you specify.

Acquire Input Data.vi



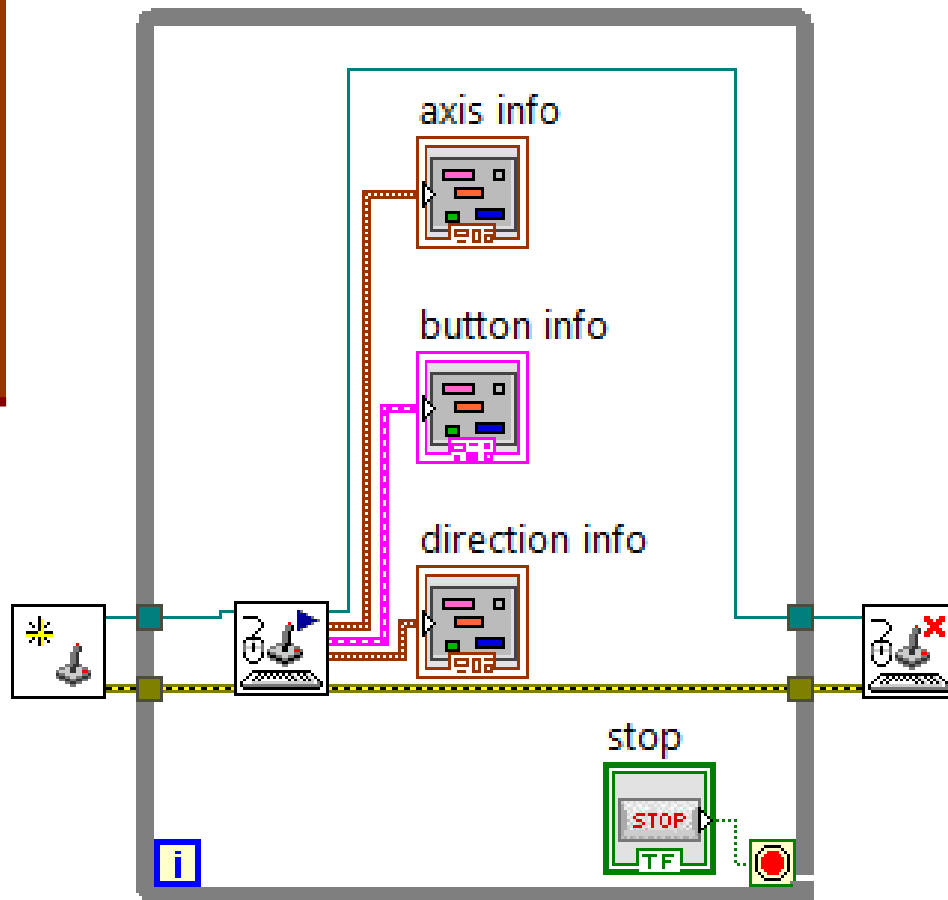
Returns data about the device connected to the computer. Wire data to the **device ID** input to determine the polymorphic instance to use or manually select the instance.

Close Input Device.vi



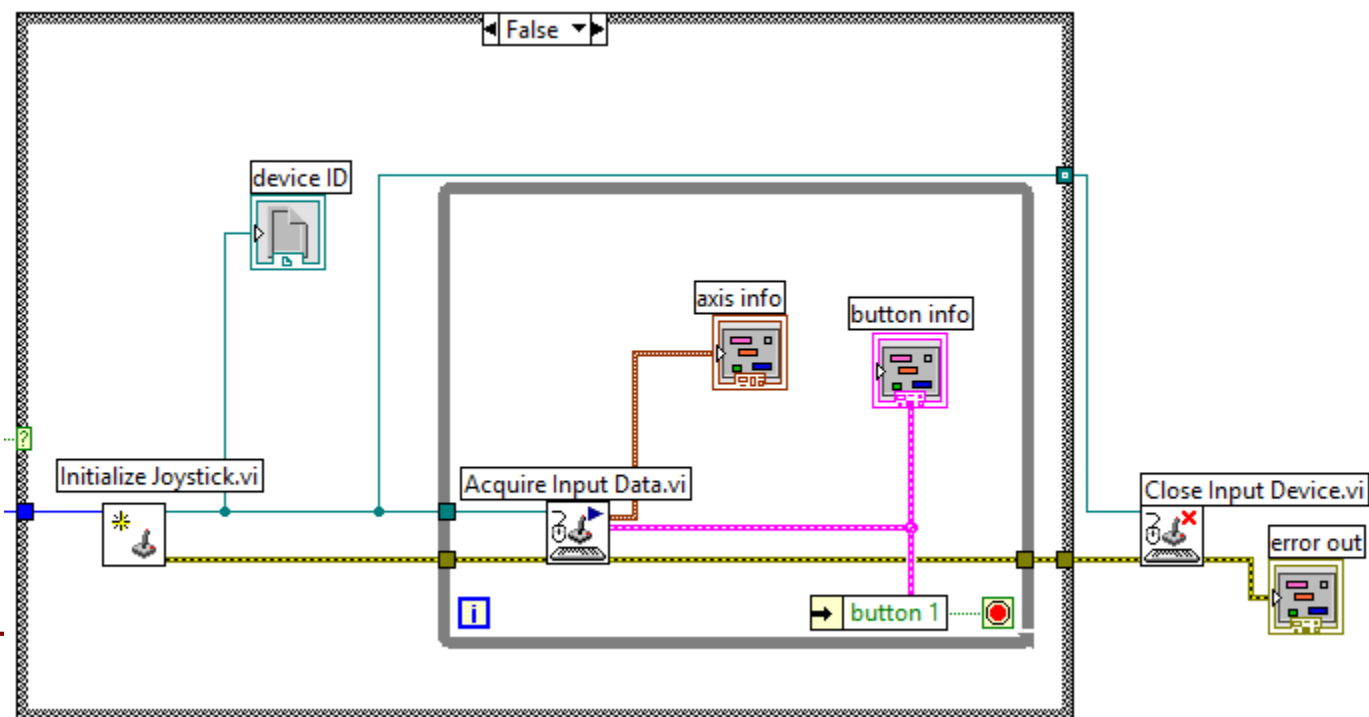
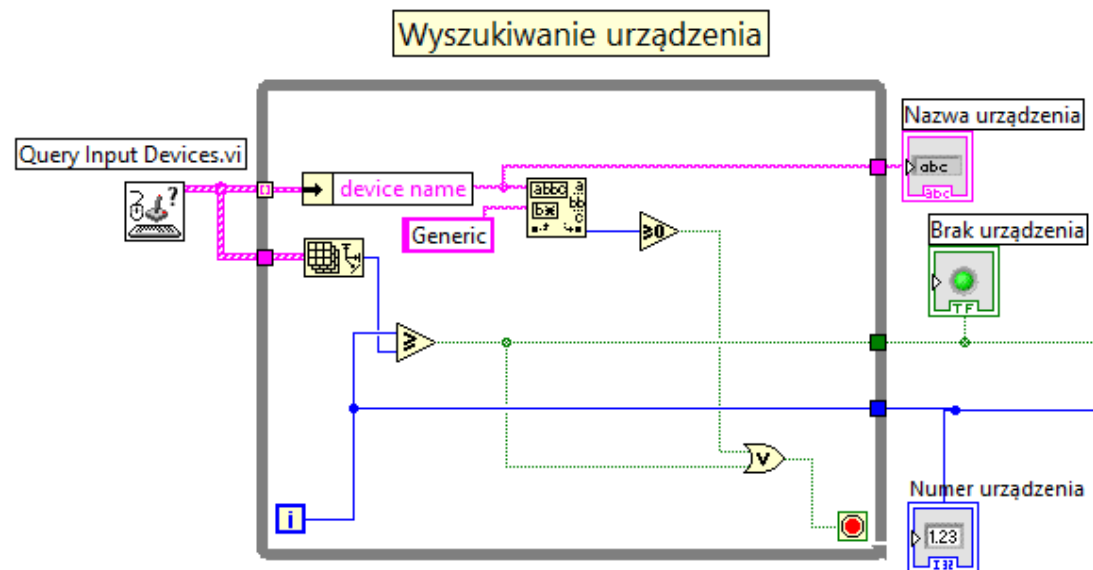
Closes the device you specify in **device ID**. Wire data to the **device ID** input to determine the polymorphic instance to use or manually select the instance.

Pad



axis info	button info	direction info
X axis 0	button 1	pov 1 0
Y axis 0	button 2	pov 2 0
Z axis 0	button 3	pov 3 0
X axis rotation 0	button 4	pov 4 0
Y axis rotation 0	button 5	
Z axis rotation 0	button 6	
U axis 0	button 7	
V axis 0	button 8	
	button 9	
	button 10	

Ciąg „Generic” może trzeba zamienić na inny ciąg, powiązany z nazwą pada. Bardzo często wystarczy też wprowadzić numer urządzenia – 0. Po określeniu numeru urządzenia – czyli jak pad będzie właściwie działał – górna pętla przestaje być potrzebna.



Zadanie 7.1

- korzystając ze struktury „event” (a nie przykładu ze str. 10!) napisać program:
 - wyświetlający wpisany z klawiatury znak – pomocna może być funkcja „type cast”
 - rozszerzyć program o możliwość pisania łańcuchów i kasowania znaków – pamiętać o czyszczeniu bufora danych po uruchomieniu programu i usuwaniu niepotrzebnych/niewidocznych znaków
 - uzupełnić/zmodyfikować strukturę tak aby poprawnie działał przycisk „stop”

Zadanie 8.1

- napisać program sterowany przy użyciu padu (przykład na poprzednim slajdzie) – jeśli pad jest niedostępny to może być klawiatura - klawisze, lub mysz – przyciski i koło (przykłady wg. str. 10 i 11 - Wykład 7), umożliwiające wykorzystanie urządzenia sterującego do kontroli programu, np.:
 - rysowania figury geometrycznej – odpowiednio oprogramować zmianę rozmiaru, poruszanie się, zmianę koloru i innych parametrów
 - kontroli nad obiektem 3D – jak powyżej
 - mini gry (np. prosty snake)
 - inne pomysły mile widziane
-