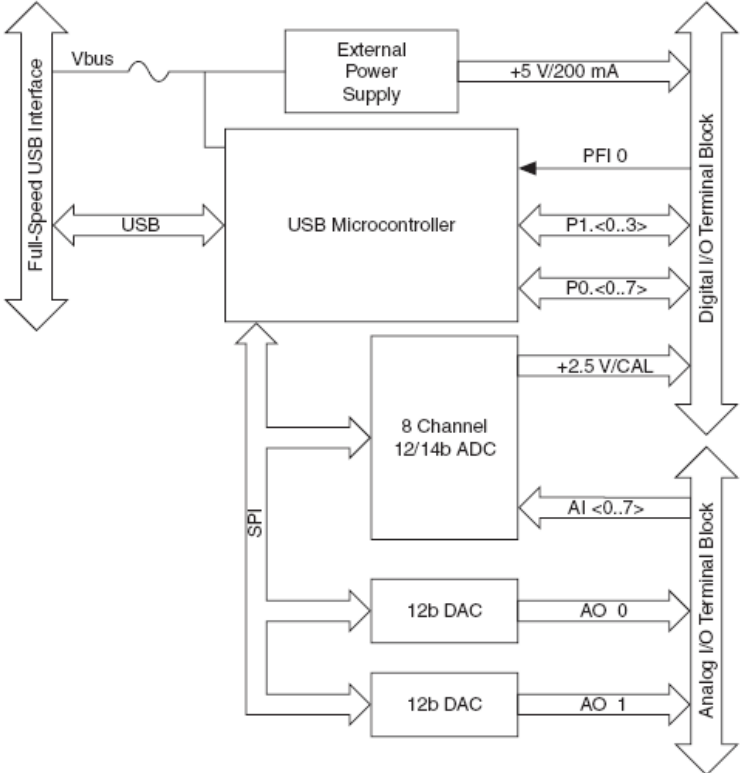


Obsługa kart I/O

Karta NI USB-6008 posiada:

- osiem wejść analogowych (AI),
- dwa wyjścia analogowe (AO),
- 12 cyfrowych wejść-wyjść (DIO),
- 32-bitowy licznik.

Schemat blokowy karty



Podstawowe parametry karty NI USB-6008:

- rozdzielczość wejść (AI) - 12 bitów (differential), 11 bitów (single-ended),
- maksymalna częstotliwość próbkowania wejść (AI) - Single Channel 10 kS/s, Multiple Channels (Aggregate) 10 kS/s,
- konfiguracja wejść-wyjść (DIO) - Open collector.

Karty pomiarowe firmy National Instruments dostarczane są wraz sterownikami dla środowiska Windows.

Większość urządzeń do akwizycji danych posiada cztery standardowe elementy wymagające konfiguracji i obsługi: wejścia analogowe, wyjścia analogowe, wejścia-wyjścia cyfrowe, liczniki. Dodatkowo w niektórych systemach znajdują się wejścia pozwalające na zewnętrzne wyzwalenie akwizycji i synchronizację.

Sterowniki do kart National Instruments

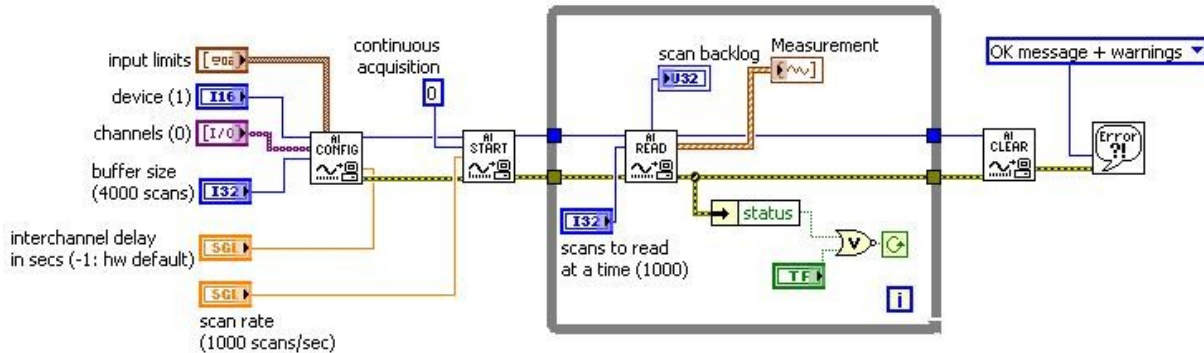
Karty firmy National Instruments posiadają rozbudowane wewnętrzne układy sterujące. W celu obsługi tych kart dostępne są sterowniki NI-DAQ dostarczające rozbudowany zestaw funkcji oraz przykłady pozwalające na obsługę kart z poziomu różnych języków programowania, takich jak:

- C,
- C#
- Visual Basic,
- Labview.

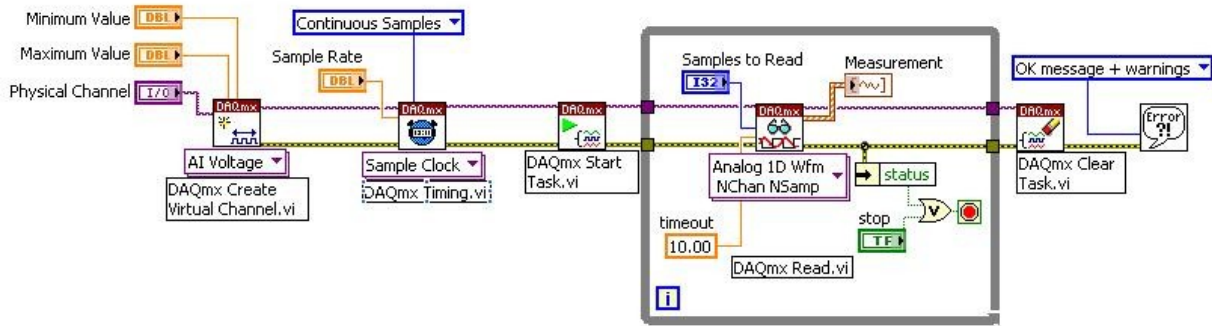
Sterowniki te umożliwiają wykorzystanie różnych zestawów funkcji API:

- Traditional NI-DAQ,
- NI-DAQmx,
- NI-DAQmx Base.

Traditional NI-DAQ jest starszą biblioteką funkcji API, która jest obecnie nie rozwijana. NI-DAQmx jest najnowszą wersją API, która posiada zmienioną koncepcję obsługi kart pomiarowych w stosunku do poprzedniej biblioteki. NI-DAQmx Base jest uproszczoną wersją biblioteki NI-DAQmx.



Rys. 2. Przykład programu w środowisku Labview wykorzystującego Traditional NI-DAQ



Rys. 3. Przykład programu w środowisku Labview wykorzystującego NI-DAQmx.

Na rysunkach 2 i 3 przedstawione są przykładowe programy utworzone w środowisku Labview. Program wykorzystujący Traditional NI-DAQ ma typową strukturę która składa się z bloku konfiguracyjnego, rozpoczynającego akwizycję, zbierającego dane oraz bloku kończącego akwizycję. W przypadku programu wykorzystującego NI-DAQmx, struktura jest inna i rozpoczyna się od utworzenia wirtualnego kanału, konfiguracji zadania oraz jego uruchomienia.

Po zainstalowaniu sterowników konfiguracja karty pomiarowej może być realizowana poprzez:

- program Measurement & Automation Explorer dostarczany wraz ze sterownikami karty,
- program utworzony w środowisku Labview,
- programu utworzonego w jednym z obsługiwanych języków programowania.

Measurement & Automation Explorer

Measurement & Automation Explorer jest programem, który pozwala na:

- ustawienie konfiguracji karty,
- przetestowanie działania karty,
- utworzenie kanałów wirtualnych oraz zadań,
- ustawienie skalowania sygnałów.

Po uruchomieniu programu pojawia się okno zawierające po lewej stronie drzewo przedstawiające konfigurację sprzętową i programową systemu.

NI-DAQmx Devices - Measurement & Automation Explorer

File Edit View Tools Help

Configuration

My System

- Data Neighborhood
- Devices and Interfaces
 - NI-DAQmx Devices
 - NI PCI-6036E: "Dev1" (Create New NI-DAQmx Device...)
 - NI USB-6008: "Dev2" (Find Network NI-DAQmx Devices)
 - PXI System (Unidentified)
 - Scales
 - Software
 - IVI Drivers
 - Remote Systems

NI-DAQmx Devices	Device Type
NI PCI-6036E: "Dev1"	Plug-in Device (NI-DAQmx...)
NI USB-6008: "Dev2"	Plug-in Device
	RTSI Cable
	NI-DAQmx SCXI Chassis
	NI-DAQmx SCC Connector Block
	NI-DAQmx TEDS Interface
	NI-DAQmx Simulated Device

Attributes

Back

NI-DAQmx Device Basics

What do you want to do?

- Run the NI-DAQmx Test Panels
- Remove the device
- View or change device properties

W gałęzi Devices and Interfaces/NI-DAQmx Devices widoczne są skonfigurowane karty pomiarowe.

W celu dodania nowej karty należy kliknąć prawym klawiszem myszy na polu NI-DAQmx Devices i w dostępnym menu wybrać typ urządzenia lub dokonać przeszukania systemu w celu odnalezienia wszystkich zainstalowanych urządzeń.

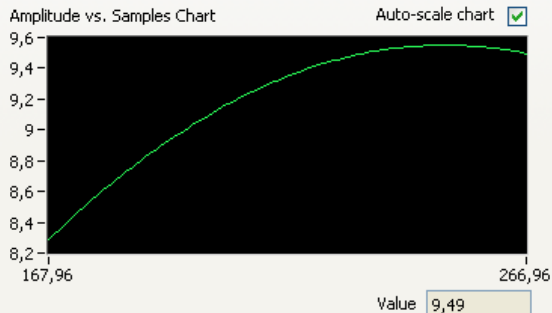
Jeżeli testowane jest tylko oprogramowanie i nie ma w systemie dostępnej karty, można skorzystać z symulowanych urządzeń wybierając NI-DAQmx Simulated Device.

Jeżeli karta jest skonfigurowana można po wybraniu tej karty uaktywnić zakładkę Test Panels i przetestować działanie poszczególnych układów dostępnych na karcie

Test Panels : NI PCI-6036E: "Dev1"

Analog Input Analog Output Digital I/O Counter I/O

Channel Name	Max Input Limit	Rate (Hz)
Dev1/ai0	10	1000
Mode	Min Input Limit	Samples To Read
On Demand	-10	1000
Input Configuration		
Differential		



Start Stop

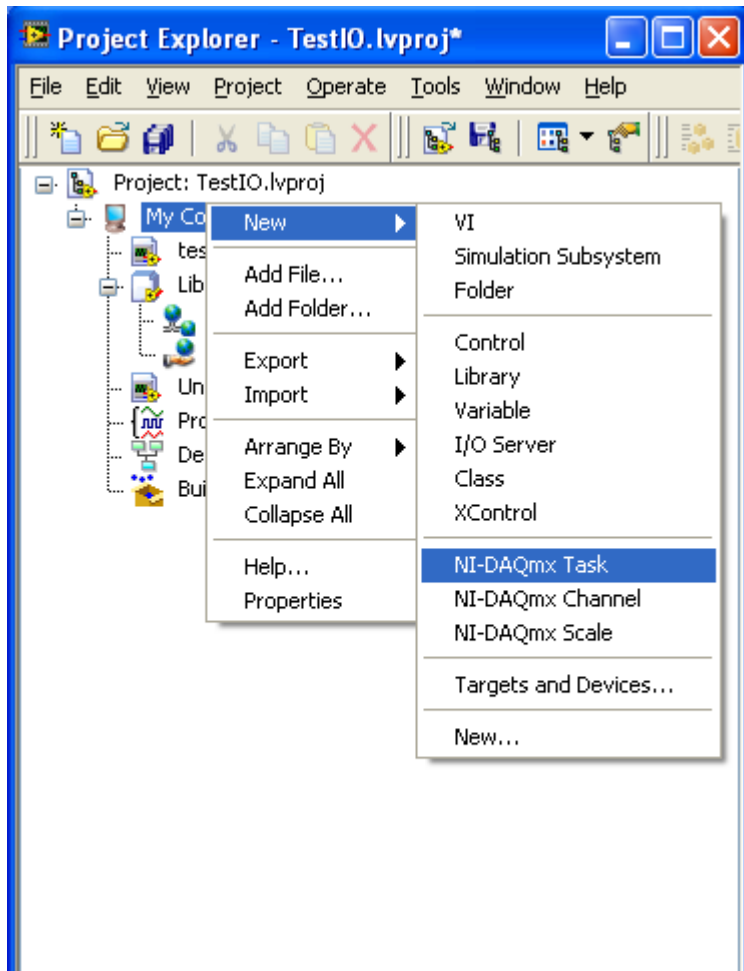
Help Close

Obsługa kart pomiarowych w Labview

W przypadku programowania z wykorzystaniem środowiska Labview dostępne są różne metody obsługi kart pomiarowych:

- oparta o wykorzystanie bloku DAQ Assistant, który pozwala na konfigurację karty za pomocą kreatora,
- wykorzystująca bloki programistyczne z biblioteki NI-DAQmx,
- kanały wirtualne i zadania można także tworzyć i konfigurować w programie Measurement & Automation Explorer, a następnie wykorzystywać w programie Labview.

Tworzenie wirtualnych kanałów, zadań oraz konfiguracji skalowania możliwe jest do realizacji na poziomie projektu, co widoczne jest na rysunku. Wszystkie elementy utworzone na poziomie projektu dostępne są dla każdego VI utworzonego w tym projekcie. Konfiguracja odbywa się w taki sam sposób jak w programie Measurement & Automation Explorer.



Wszystkie funkcje do obsługi kart pomiarowych w środowisku Labview dostępne są w bibliotece Measurement IO → NI-DAQmx.



Funkcje te są dostępne tylko wtedy, gdy są zainstalowane sterowniki NI-DAQmx. Sterowniki te są instalowane jako opcja dodatkowa przy instalacji Labview.

Aktualna wersja dostarczana jest także wraz z kartą pomiarową. Natomiast najnowszą wersję można pobrać z witryny firmy National Instruments.

Przy instalacji nowej wersji sterowników należy zawsze sprawdzić, z jakimi wersjami Labview i systemów operacyjnych dana wersja współpracuje.

Wykorzystanie bloku DAQ Assistant

DAQ Assistant jest graficznym interfejsem służącym do interaktywnego tworzenia, edycji i uruchamiania kanałów wirtualnych i zadań wykorzystującym bibliotekę NI-DAQmx. W ramach biblioteki NI-DAQmx tworzone są wirtualne kanały składające się z fizycznego kanału na urządzeniu DAQ oraz informacji o konfiguracji tego kanału, takich jak konfiguracja wejścia, zakres wejściowy, skalowanie. Zadaniem w sterownikach NI-DAQmx jest zbiór kanałów wirtualnych połączony z informacją o czasie oraz informacjami dotyczącymi wyzwalania i ilości zbieranych próbek.



Po umieszczeniu w programie bloku DAQ Assistant nie ma on widocznych wejść i wyjść co świadczy, że kanały i zadanie z nim związane nie są skonfigurowane. Należy kliknąć na nim dwa razy w celu uruchomienia kreatora pozwalającego na skonfigurowanie kanałów i zadania.

Create New Express Task...

NI-DAQ™
DAQ Assistant




Select the measurement type for the task.

A **task** is a collection of one or more virtual channels with timing, triggering, and other properties.


To have **multiple measurement types** within a single task, you must first create the task with one measurement type. After you create the task, click the **Add Channels** button to add a new measurement type to the task.

Acquire Signals

[-] Analog Input

 Voltage

 Temperature

 Strain

 Current


 Resistance

 Frequency

 Position

 Sound Pressure

 Acceleration

 Force

 Pressure

 Torque

< Back

Next >

Finish

Cancel

W oknie należy wybrać, czy sygnał ma być mierzony, czy generowany oraz określić rodzaj mierzonego sygnału, co będzie miało wpływ na jednostki i współczynniki skalowania.

Następnie należy wybrać kanały które mają być wykorzystywane w ramach tworzonego zadania.

Select the physical channel(s) to add to the task.

If you have previously configured [global virtual channels](#) of the same measurement type as the task, click the **Virtual** tab to add or copy global virtual channels to the task. When you copy the global virtual channel to the task, it becomes a local virtual channel. When you add a global virtual channel to the task, the task uses the actual global virtual channel, and any changes to that global virtual channel are reflected in the task.

If you have TEDS configured, click the **TEDS** tab to add TEDS channels to the task.

For hardware that supports [multiple channels](#) in a task, you can select multiple channels to add to a task at the same time.

Physical

Supported Physical Channels

- Dev1 (DAQCard-6036E)
 - ai0
 - ai1
 - ai2
 - ai3
 - ai4
 - ai5
 - ai6
 - ai7
 - ai8
 - ai9
 - ai10
 - ai11
 - ai12
 - ai13

<Ctrl> or <Shift> click to select multiple channels.

< Back

Next >

Finish

Cancel

Po zatwierdzeniu pojawia się okno, w którym możemy dokonać konfiguracji szczegółowej utworzonego zadania oraz poszczególnych kanałów, a także uruchomić go w celu przetestowania czy prawidłowo zostało skonfigurowane.

Configuration

Triggering

Advanced Timing

Logging

Channel Settings



Details >>

Voltage

Click the Add Channels button (+) to add more channels to the task.

Voltage Input Setup



Settings



Calibration

Signal Input Range

Max

Min

Scaled Units

Volts

Terminal Configuration

Differential

Custom Scaling

<No Scale>

Timing Settings

Acquisition Mode

N Samples

Samples to Read

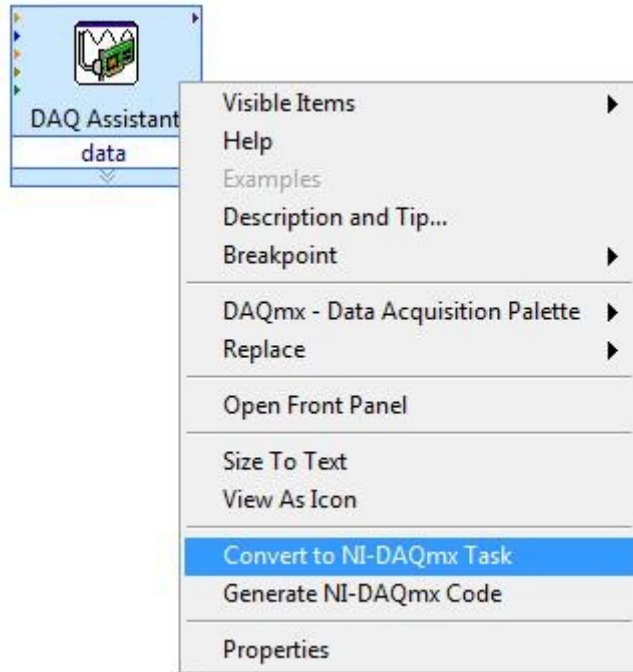
100

Rate (Hz)

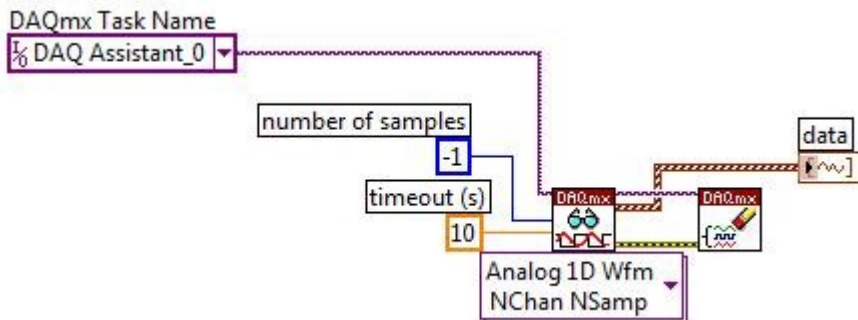
1k

Po zakończeniu konfiguracji w bloku umieszczonym na diagramie pojawiają się wejścia i wyjścia, które można wykorzystać do komunikacji z utworzonym zadaniem.

Utworzony i skonfigurowany blok możemy wykorzystywać w programie. Istnieje także możliwość przekonwertowania go na zadanie lub wygenerowania kodu.



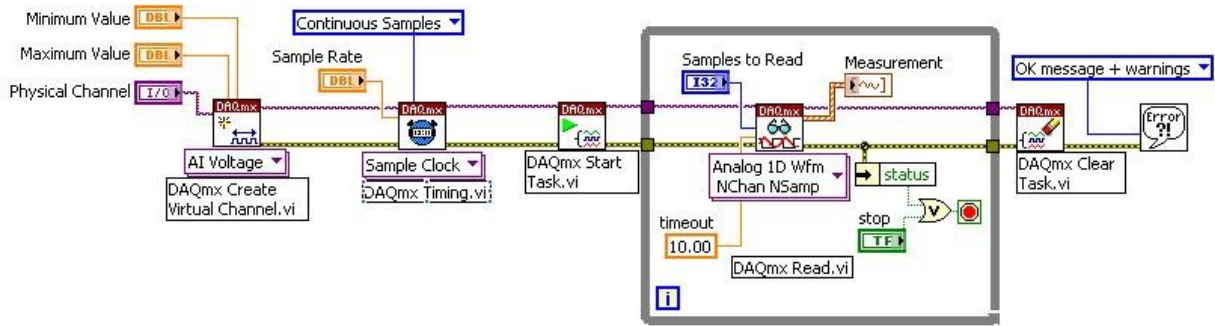
W tym przypadku struktura programu ulega uproszczeniu w stosunku do programu tworzego , gdyż należy tylko uruchomić odpowiednie zadanie, co przedstawione jest na rysunku.



Rys. Uruchomienie zadania skonfigurowanego w bloku DAQ Assistant.

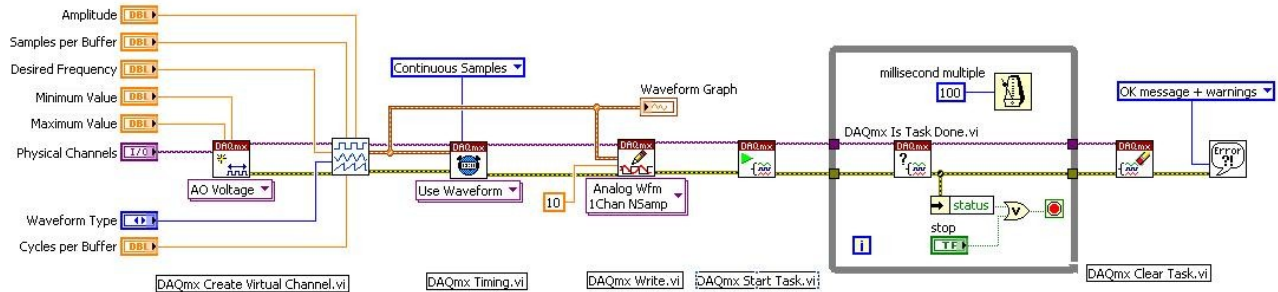
Sposób ten jest wygodny w przypadku pracy na stanowisku na którym wykonujemy pomiary. Jeżeli program jest przenoszony na inne komputery, wadą tego rozwiązania jest konieczność konfiguracji zadania oddzielnie na każdym z komputerów. W tym przypadku lepszym rozwiązaniem jest tworzenie kanałów wirtualnych i zadań z poziomu kodu programu.

Obsługa wejść analogowych



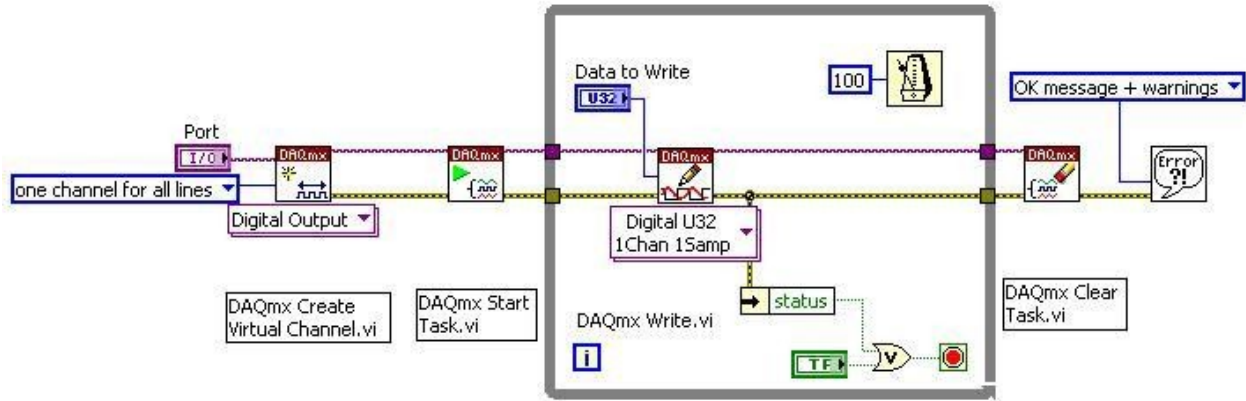
NI-DAQmx: Cont. Acq&Graph Voltage-Int Clk.vi

Obsługa wyjść analogowych



NI-DAQmx: Cont Gen Voltage Wfm-Int Clk.vi

Obsługa wyjść cyfrowych



NI-DAQmx: Write Dig Port.vi

Obsługa w języku Python

```
from PyDAQmx import *; import numpy
dev="cDAQ1Mod1"; acqtime=60; chan=2; freq=100000.0
numsamples=int(freq*acqtime); timeout=acqtime+10
taskHandle = TaskHandle(); read = int32()
data = numpy.zeros((numsamples*chan,), dtype=numpy.float64)

# Konfiguracja zadania, kanałów i próbkowania
DAQmxCreateTask("",byref(taskHandle))
DAQmxCreateAIVoltageChan(taskHandle,dev+"/ai0","",DAQmx_Val_Cfg_De
fault,-10.0,10.0,DAQmx_Val_Volts,None)
DAQmxCreateAIVoltageChan(taskHandle,dev+"/ai1","",DAQmx_Val_Cfg_De
fault,-10.0,10.0,DAQmx_Val_Volts,None)
DAQmxCfgSampClkTiming(taskHandle,"",freq,DAQmx_Val_Rising,DAQmx_Val
FiniteSamps,numsamples)

DAQmxStartTask(taskHandle) # uruchomienie zadania akwizycji
DAQmxReadAnalogF64(taskHandle,numsamples,timeout,DAQmx_Val_GroupBy
ScanNumber,data,numsamples*chan,byref(read),None)
DAQmxStopTask(taskHandle) # zatrzymanie zadania akwizycji
DAQmxClearTask(taskHandle)
np.save('d:/badania/proba_AC_jlt_80_no_C_test1.npy',data)
```