

LabWindows/CVI

(použité informačné zdroje National Instruments)

doc. Ing. Ján Šaliga, PhD.

KEMT FEI TU Košice

Čo je LabWindows/CVI

- Vývojové prostredie založené na jazyku ANSI C vytvorené National Instruments
- Jednoduchý a rýchly vývoj aplikácií pre spracovanie signálov, meranie, generovanie, archiváciu a komunikácie
- Kompatibilné s Visual C a Borland C
- Vlastný alebo externý kompilátor
- Knižnice funkcií pre rôzne aplikácie – zrýchlenie a zjednodušenie programovania

Základné vlastnosti a možnosti

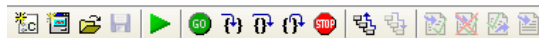
- Prehľadné prostredie - pracovný priestor (workspace) s jednoduchým ovládaním
- Automatické generovanie typických kostier programových štruktúr (main, for, while, ...) a deklarácií a iných súčastí programov
- Interaktívny výkon knižničných funkcií
- Rýchla orientácia v knižniciach a vyhľadávanie funkcií
- Kontextový pomocník (help) v každom kroku programovania
- ...

Pracovné okno (workspace window)

- Nastavenie cez Option»...

The screenshot displays the National Instruments LabVIEW workspace window for a project named 'nonInfit.c'. The workspace is divided into several panes:

- Source Files:** Shows the project structure with files like nonInfit.c, nonInfit.h, and nonInfit.uir.
- Code Editor:** Contains the C source code for the fitting function and main function. The code includes comments and function calls like `FitCurveProcedure` and `RunUserInterface`.
- nonInfit.uir:** A graphical user interface window titled 'Nonlinear Fit Example'. It features a plot area showing data points and a fitted curve. The plot has a grid and axes ranging from 0 to 127 on the x-axis and -1.00 to 1.00 on the y-axis.
- Control Panel:** Located below the plot, it includes controls for 'Points' (set to 128), 'Data Template' (set to 'Damped Sinusoid'), 'Variance from Ideal' (set to 0.050), and 'Fit Data To' (set to 'Damped Sinusoid').
- Function Selection:** Below the control panel, there are fields for 'Data Template Function' and 'Fit Function', both set to $Y(x) = \exp(-a \cdot x) \cdot \cos(w \cdot x)$. There are also fields for 'Ideal Parameter a' (0.03000), 'Ideal Parameter w' (0.20000), 'Fit Parameter a', 'Fit Parameter w', and 'Mean Squared Error' (0.00000).
- Buttons:** 'Fit Curve', 'Help', and 'Quit' buttons are visible.
- Watch and Variables:** At the bottom, there are empty 'Watch' and 'Variables' panes for monitoring program execution.



NonInfit

- Source Files
 - nonInfit.c
- Include Files
 - nonInfit.h
- User Interface File:
 - nonInfit.uir

```

<-1> nonInfit.c
}
return 0;
}

void FitCurveProcedure(void)
{
    GetCtrlVal (mainpnl, MAINPNL_IDEALPAR1, &idealpar1);
    GetCtrlVal (mainpnl, MAINPNL_IDEALPAR2, &idealpar2);
    coefs[0] = idealpar1;
    coefs[1] = idealpar2;

    DeleteGraphPlot (mainpnl, MAINPNL_G

    seed++;
    if (seed > 50) {
        seed = 1;
    }
    GaussNoise (numpoints, noiselevel,

    for (i=0; i<numpoints; i++) {
        xwave[i] = i;
        if (functtemplate == 0) datawav
        if (functtemplate == 1) datawav
        datawave[i] = datawave[i] + noi

    PlotY (mainpnl, MAINPNL_GRAPH, data
        VAL_THIN_LINE, VAL_EMPTY_SQUA

    if (functfit == 0) {
        NonLinearFit (xwave, datawave,
            1, &meansqerror);
        SetInputMode (mainpnl, MAINPNL_
    }
    else {
        NonLinearFit (xwave, datawave,

```

nonInfit.uir

Nonlinear Fit Example

Points: 128

Data Template: Damped Sinusoid

Variance from Ideal: 0.050

Fit Data To: Damped Sinusoid

Data Template Function: $Y(X) = \text{Exp}(-a^*X) * \text{Cos}(w^*X)$

Fit Function: $Y(X) = \text{Exp}(-a^*X) * \text{Cos}(w^*X)$

Ideal Parameter a: 0.03000

Ideal Parameter w: 0.20000

Fit Parameter a: 0.03000

Fit Parameter w: 0.20000

Mean Squared Error: 0.00000

Auto Fit Curve:

Buttons: Edit Curve, Help, Quit

Libraries

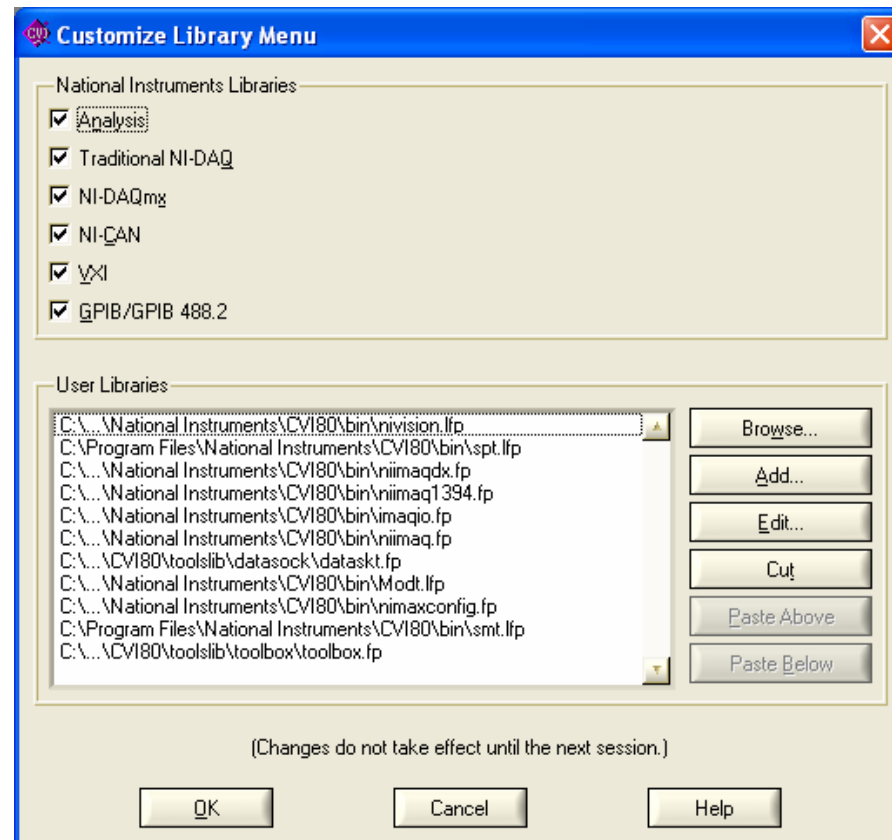
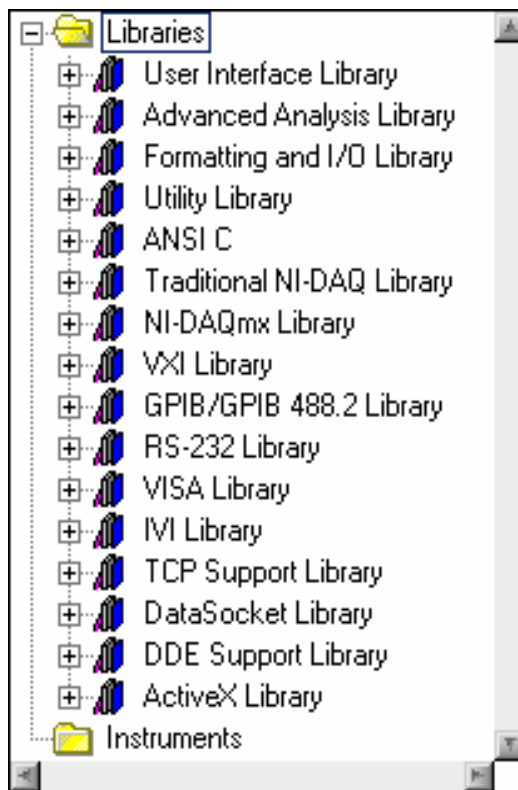
- User Interface Libr
 - Panels
 - Menu Structure
 - Controls/Graph
 - Pop-up Panels
 - InstallPopUp
 - RemovePop
 - Message/Pr
 - File/Director
 - FileSelec
 - MultiFileS
 - DirSelect
 - AddToFil
 - GetFilePc
 - ClearFilef
 - Graph Popu
 - Font Popups
 - GetSystemP
 - SetSystemP
- KeyPress Even
- Callback Functi
- User Interface I
- Printing
- Mouse and Cur
- Rectangles anc
- Bitmaps

Name	Value	Type

Name	Value	Type
Globals (nonInfit_dbg.exe)		
nonInfit.c Statics (nonInfit_dbg.exe)		
mainpnl	1	
numpoints	128	
noiselevel	0.0500000000000000	
functfit	1	
functtemplate	1	
FitCurveProcedure (nonInfit_dbg.exe)		

Knižničné okno (library tree)

- Prehľad výber – výber – info o polohe funkcie
- Prispôsobenie aktuálneho súboru knižníc cez Library»Customize, zobrazenie cez View»...



Štandardné knižnice

- **User Interface Library**—Functions for creating and controlling a graphical user interface.
- **Analysis Library (Base Package)/Advanced Analysis Library (Full Development System)**—Functions that operate on arrays to simulate and analyze large sets of numerical data quickly and efficiently.
- **Formatting and I/O Library**—Functions for inputting and outputting data to files and manipulating the format of data in a program.
- **Utility Library**—Functions that perform various operations, including using the system timer, managing disk files, launching another executable, and using multiple threads in a program.
- **ANSI C Library**—The ANSI C standard library functions.
- **VXI Library, GPIB/GPIB 488.2 Library and RS-232 Library** —Functions for communicating with and controlling VXI devices, General Purpose Interface Bus (GPIB) and RS-232 ports using interrupt-driven I/O.
- **VISA Library** —Functions for controlling VXI, GPIB, serial, and other types of instruments.
- **TCP Support Library** —Functions that provide a platform-independent interface to the reliable, connection-oriented, byte-stream, network communication protocol.
- **Internet Library** —Functions that communicate with and receive files and commands from remote servers.
- **DDE Support Library** —Functions that you can use to create an interface with other Windows applications using the DDE standard.
- **ActiveX Library** —Functions that create and control ActiveX servers.
- **DIAdem Connectivity Library** —Functions that you can use to log test data in National Instruments DIAdem file format (.tdm).
- **.NET Library** —Functions that facilitate calling .NET assemblies.

Okno projektu (project window)

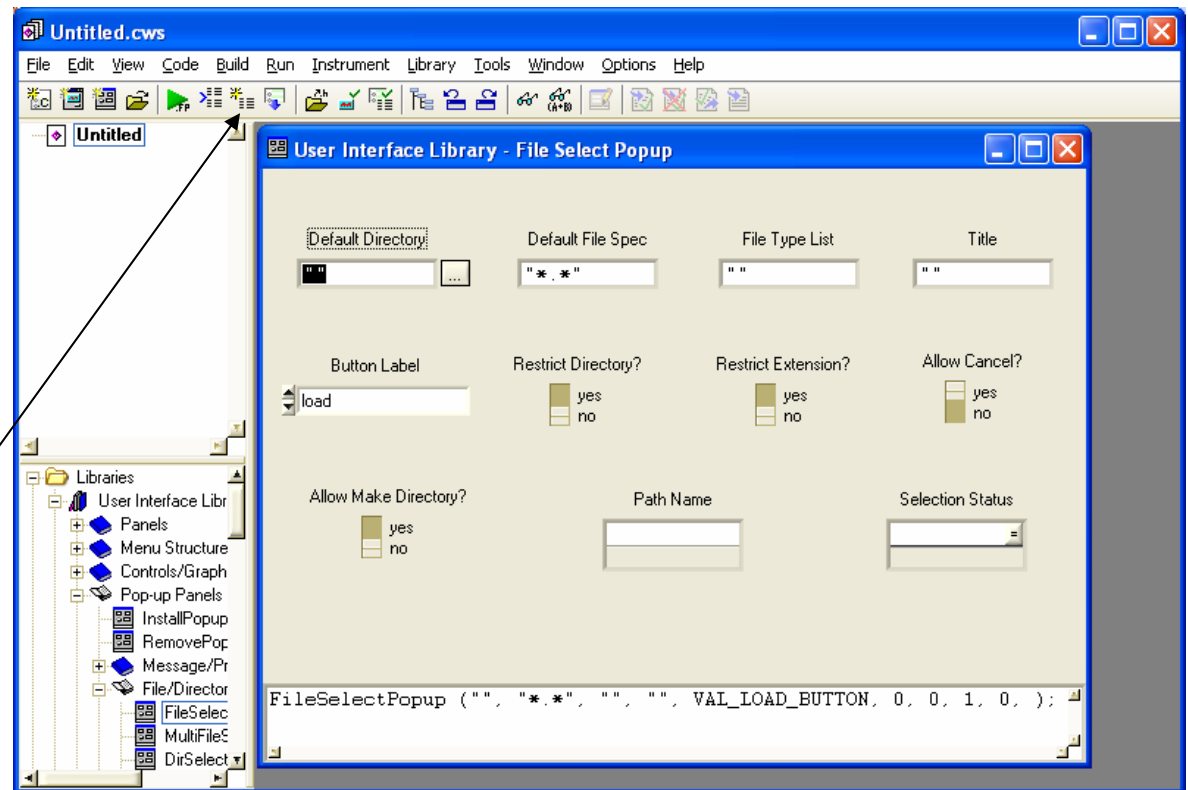
- Projekt = zoznam všetkých zdrojových (.c), deklaračných (.h), grafických (.uir), a ďalších súborov (.obj, .lib, .fp, ...) , ktoré spolu vytvárajú aplikáciu
- Vytvorenie nového/načítanie existujúceho cez menu File
- Nastavenie zobrazenia cez View»Columns alebo iné položky vo View
- V pracovnom priestore môže byť súčasne otvorených viacero projektov – aktívny je zvýraznený (dvojklik)

Zdrojové okno (source window)

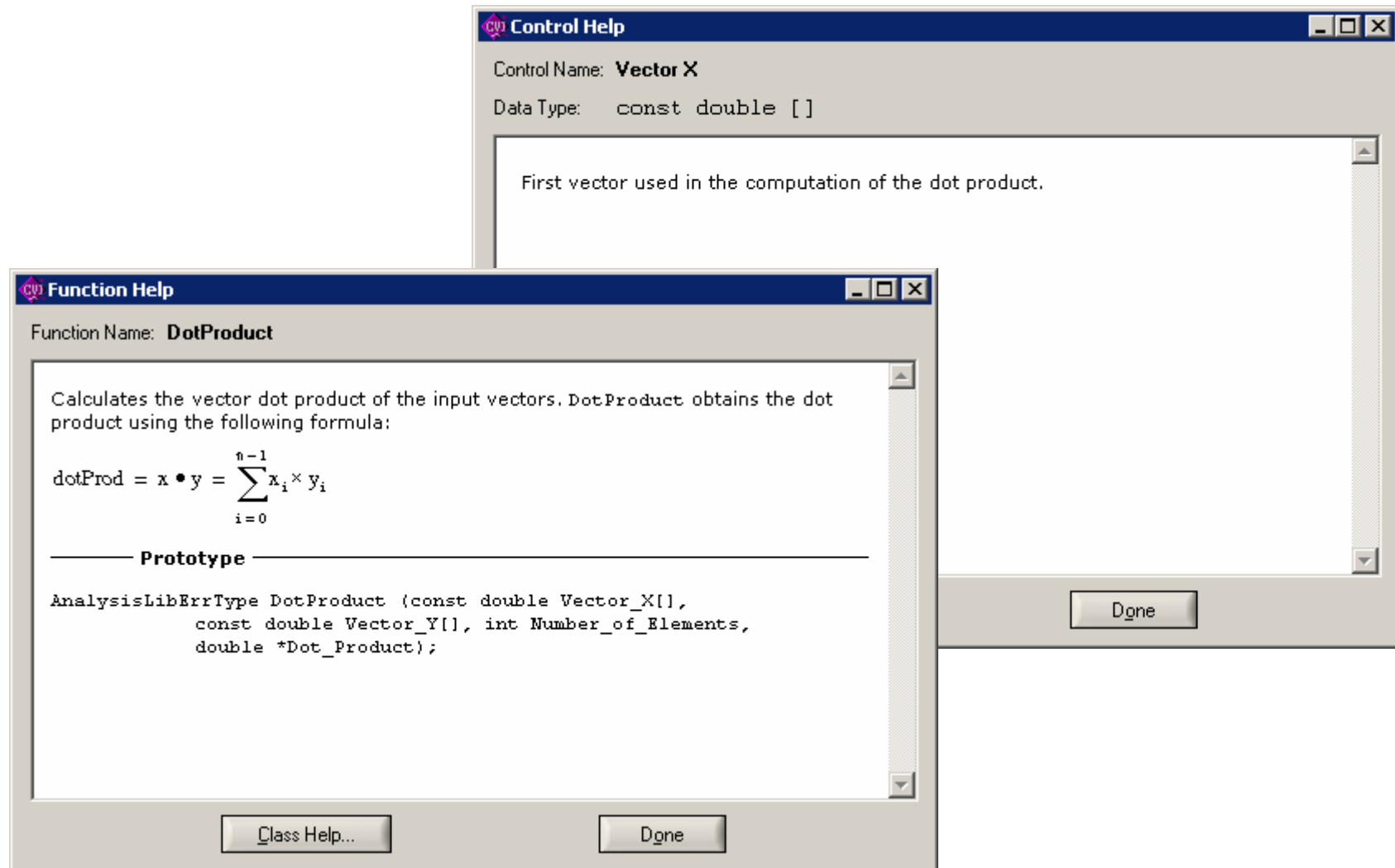
- Vytvorenie nového/načítanie existujúceho cez menu File
- Editácia
 - Vkladanie pripravených skeletoňov: Edit » Insert Construct
 - Vkladanie funkcií
 - priame písanie s inteligentným dopĺňaním (prvé písmená»pravé tlačidlo»Recall Function Panel
 - Výber z knižnice (knižničné okno alebo ponuka Library»...)
 - Deklarácie premenných: priamy zápis alebo poloautomatizované cez Function Panel vybranej funkcie
 - ...
- Ladenie
 - Vkladanie prerušení (breakpoint) – klik do ľavého stĺpca
 - Rôzne úrovne krokovania Run»... alebo ikony na lište
 - Sledovanie hodnôt premenných (watch) – menu pod pravým tlačidlom myši
 - ...

Panel funkcie (Function Panel)

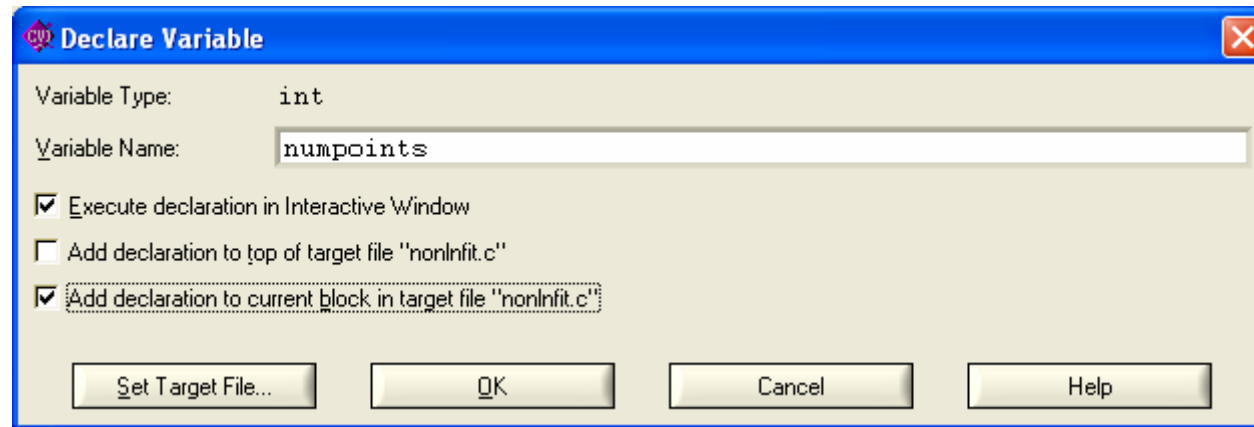
- K dispozícii ku každej funkcii z knižníc cez menu pod pravým tlačidlom myši
- Vyplnenie parametrov do políčk – pravé tlačidlo=help
- Deklarácia premennej, vloženie existujúcej premennej cez Code» alebo ikony na lište



Pomocník k položce, funkci, ...



Deklarácia premennej (aj Ctrl-D),



Interaktívny výkon inštrukcie

Interactive Execution

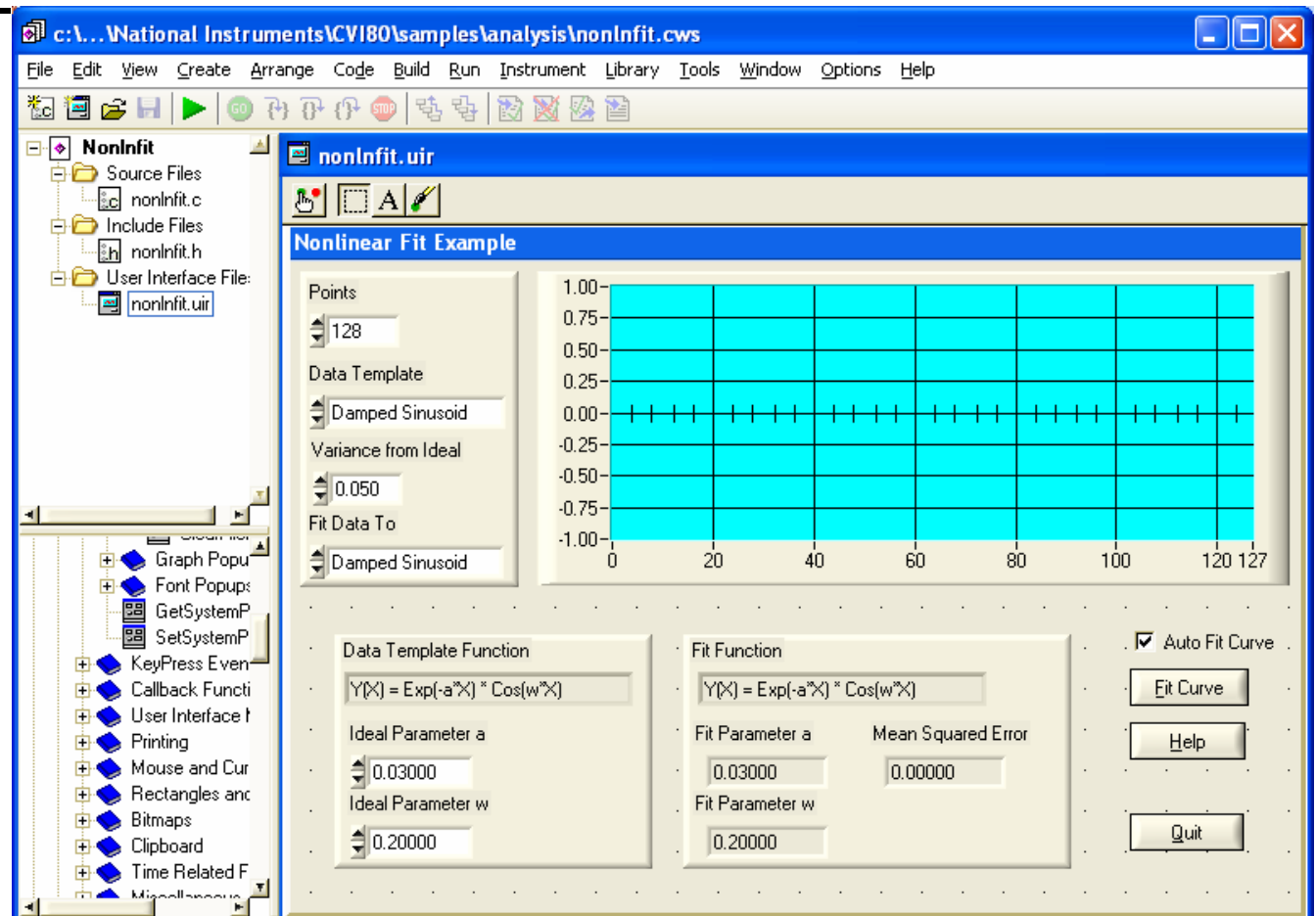
File Edit View Build Run Instrument Library Tools Window Options Help

```
1 #include <ansi_c.h>
2 #include <utility.h>
3 Cls();
4 printf("Welcome to the LabWindows/CVI Basics I Course.");
5
```

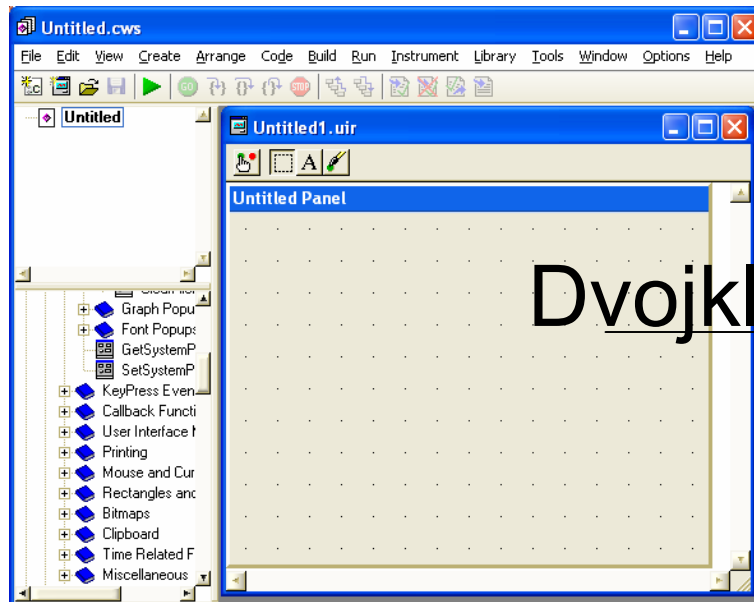
1/5 35 CS Ins

GUI a editor grafického rozhrania

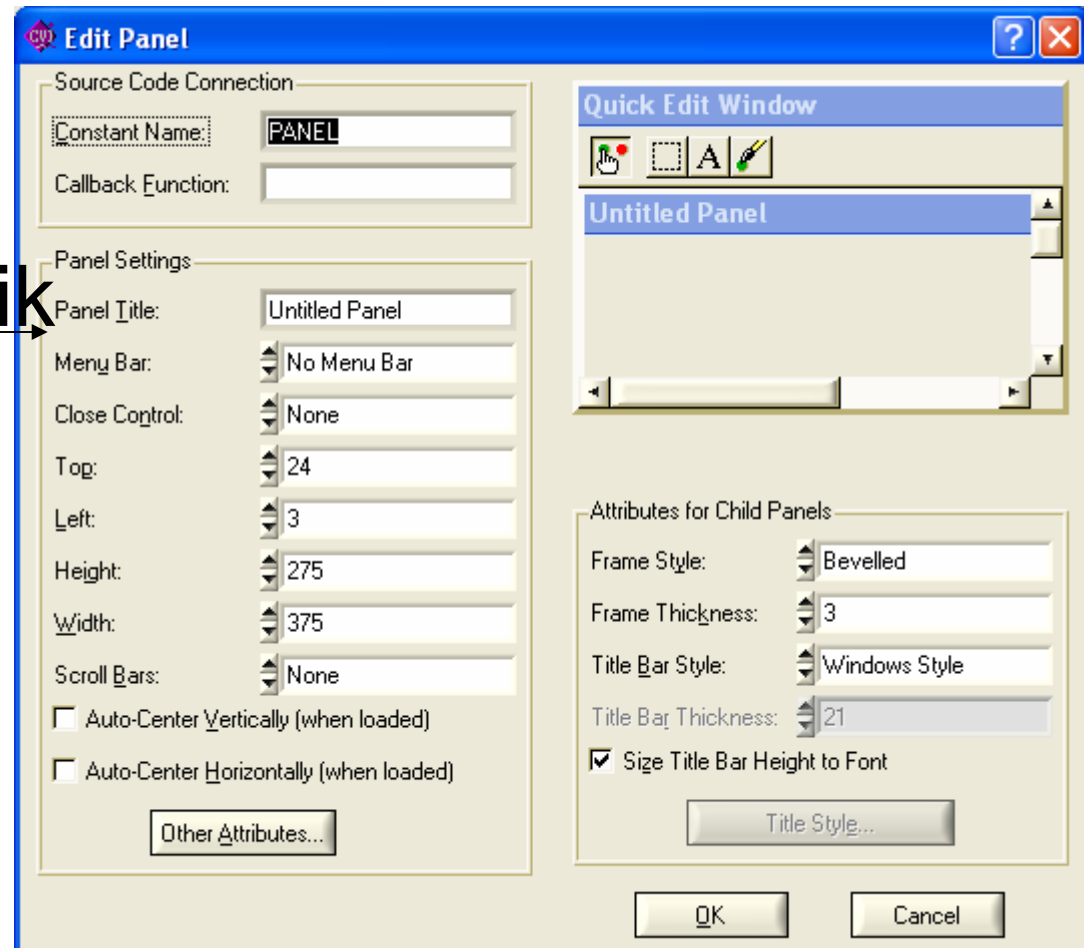
- Vytvorenie nového/načítanie existujúceho cez menu File
- Dizajn: vytvorenie panelu a vloženie grafiky z knižničných grafických prvkov
- Nastavenie ich vlastností a väzby na zdroj .C cez ich panely (dvojklik)



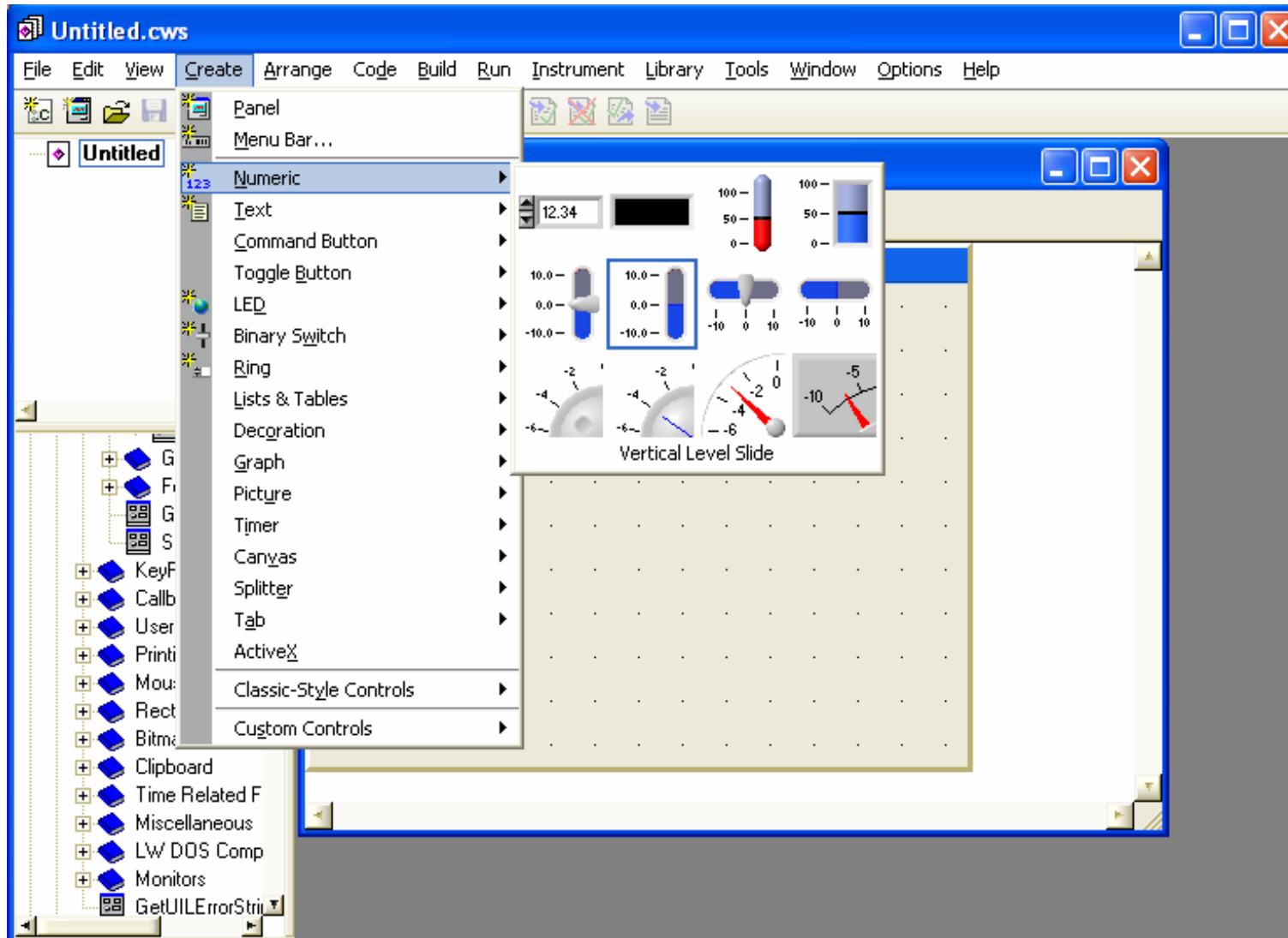
Nastavenie vlastností panelu



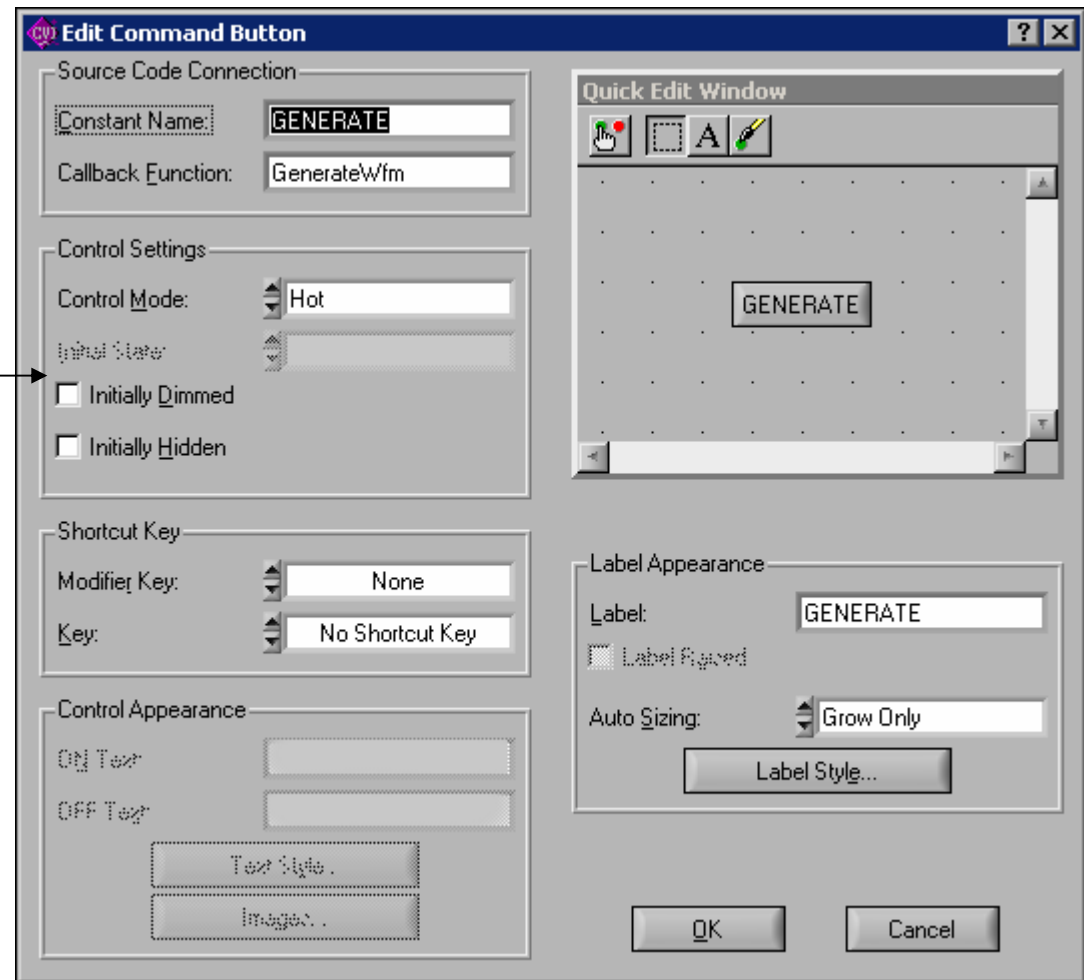
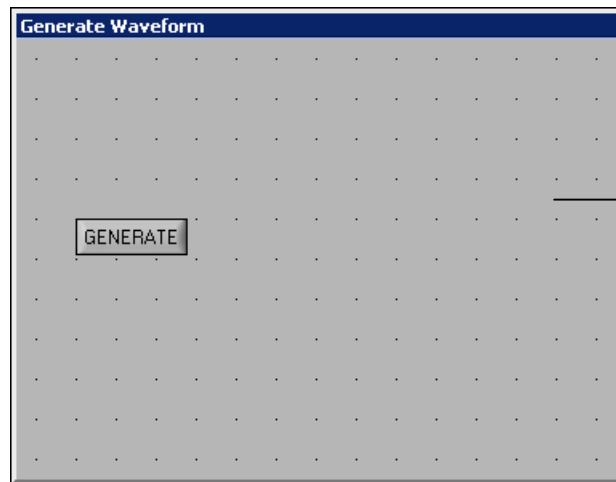
Dvojklik



Knižnice grafických prvkov



Nastavenie vlastností prvkov



Väzba GUI na .c

- Každý prvok môže byť indikátorom alebo ovládacím prvkom.
- Meno prvku (premenná) = vzniká naviazaním mien panelu a prvku
- Indikátor (napr. graf, LED, ...) – odosielanie dát/ zistenie-nastavenie vlastností podľa mena prvku cez vhodné knižničné funkcie z User interface library.
- Ovládač (Control, napr. tlačidlo, potenciometer, menu, ...)
 - zistenie stavu čítaním cez funkcie z User interface library
 - Zistenie/nastavenie vlastností cez funkcie z User interface library
 - **Automatické spustenie (vyvolanie) obslužnej rutiny (podprogramu) = program riadený udalosťami: zvolenej v okne vlastností prvku – tzv. Callback (kostra automaticky vygenerovaná prostredím cez Code»Generate»...**

Niektoré UIR funkcie

Panels

Load Panel	<code>LoadPanel</code>
Load Panel (Extended)	<code>LoadPanelEx</code>
Display Panel	<code>DisplayPanel</code>

Pop-up Panels»Message/Prompt Popups

Message Popup	<code>MessagePopup</code>
----------------------	---------------------------

Controls/Graphs/Strip Charts»General Functions

Set Control Value	<code>SetCtrlVal</code>
Get Control Value	<code>GetCtrlVal</code>
Set Control Attribute	<code>SetCtrlAttribute</code>
Get Control Attribute	<code>GetCtrlAttribute</code>

User Interface Management

Run User Interface	<code>RunUserInterface</code>
Quit User Interface	<code>QuitUserInterface</code>

Controls/Graphs/Strip Charts»Graphs and Strip Charts» Graph Plotting and Deleting

Plot Y	<code>PlotY</code>
Delete Graph Plot	<code>DeleteGraphPlot</code>

Controls/Graphs/Strip Charts»Graphs and Strip Charts»

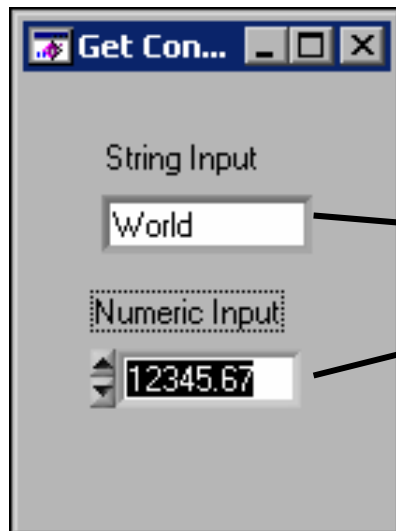
Strip Chart Traces	
Plot Strip Chart	<code>PlotStripChart</code>
Clear Strip Chart	<code>ClearStripChart</code>

Kostru programu pre GUI

```
1  #include <userint.h>
2  #include "genwave.h"
3
4  static int panelHandle;
5
6  int main ()
7  {
8      panelHandle = LoadPanel (0, "genwave.uir", PANEL);
9      if (panelHandle < 0) {
10         MessagePopup("", "Error Loading Panel");
11         return -1;
12     }
13     else
14         DisplayPanel (panelHandle);
15
16     .
17     . /* program code */
18     .
19     return 0;
20 }
21
```

21/21 1 C Ins

Načítanie hodnoty



```
1 #inc
2 #inc
3
4 static
5 static double number;
6 static char buffer[100];
7
8 int main ()
9 {
10     panelHandle = LoadPanel (0, "getvalue.uir", PANEL);
11     if (panelHandle < 0) {
12         MessagePopup("", "Error Loading Panel");
13         return -1;
14     }
15     else
16         DisplayPanel (panelHandle);
17
18     GetCtrlVal(panelHandle, PANEL_STRIN, buffer);
19     GetCtrlVal(panelHandle, PANEL_NUMIN, &number);
20     return 0;
21 }
22
```

Globals		
setvalue.c		
panelHandle	1	int
number	12345.6700000000	double
buffer	009A0950 "World"	char [100]

Nastavenie hodnoty

The image shows a GUI window titled "Set Cont..." and a code editor window. The GUI window has two sections: "String Output" with a text box containing "Hello" and "Numeric Output" with a text box containing "1.23". The code editor shows C code with two arrows pointing from the code to the GUI elements. The first arrow points from line 20, `SetCtrlVal(panelHandle, PANEL_STROUT, buffer);`, to the "Hello" text box. The second arrow points from line 21, `SetCtrlVal(panelHandle, PANEL_NUMOUT, number);`, to the "1.23" text box.

```
1  #include <userint.h>
2  #include "setvalue.h"
3
4  static int panelHandle;
5  static double number;
6  static char buffer[100];
7
8  int main ()
9  {
10     panelHandle = LoadPanel (0, "setvalue.uir", PANEL);
11     if (panelHandle < 0) {
12         MessagePopup("", "Error Loading Panel");
13         return -1;
14     }
15     else
16         DisplayPanel (panelHandle);
17
18     strcpy (buffer, "Hello");
19     number = 1.23;
20     SetCtrlVal(panelHandle, PANEL_STROUT, buffer);
21     SetCtrlVal(panelHandle, PANEL_NUMOUT, number);
22     return 0;
23 }
24
```

Nastavenie vlastností prvku

```
1  #include <userint.h>
2  #include "SetAttributes.h"
3
4  static int panelHandle;
5
6  int main ()
7  {
8      panelHandle = LoadPanel (0, "SetAttributes.uir", PANEL);
9      if (panelHandle < 0) {
10         MessagePopup("", "Error Loading Panel");
11         return -1;
12     }
13     else
14         DisplayPanel (panelHandle);
15
16     SetCtrlAttribute (panelHandle, PANEL_COMMANDBUTTON,
17                     ATTR_CMD_BUTTON_COLOR, VAL_BLUE);
18     SetCtrlAttribute (panelHandle, PANEL_COMMANDBUTTON,
19                     ATTR_TOP, 20);
20     return 0;
21 }
22
```

The image illustrates the effect of applying different attributes to a button in a dialog box titled "Set Control Attribute". The dialog box contains a single button labeled "CHANGE MY ATTRIBUTES".

- The top screenshot shows the button with its default appearance (white background, black text).
- The middle screenshot shows the button with the attribute `ATTR_CMD_BUTTON_COLOR, VAL_BLUE` applied, resulting in a blue background and white text.
- The bottom screenshot shows the button with the attribute `ATTR_TOP, 20` applied, resulting in the button being positioned higher on the dialog box.

Arrows point from the corresponding code lines in the C program to the respective screenshots, demonstrating how the code changes the button's appearance and position.

Riadenie udalosťami

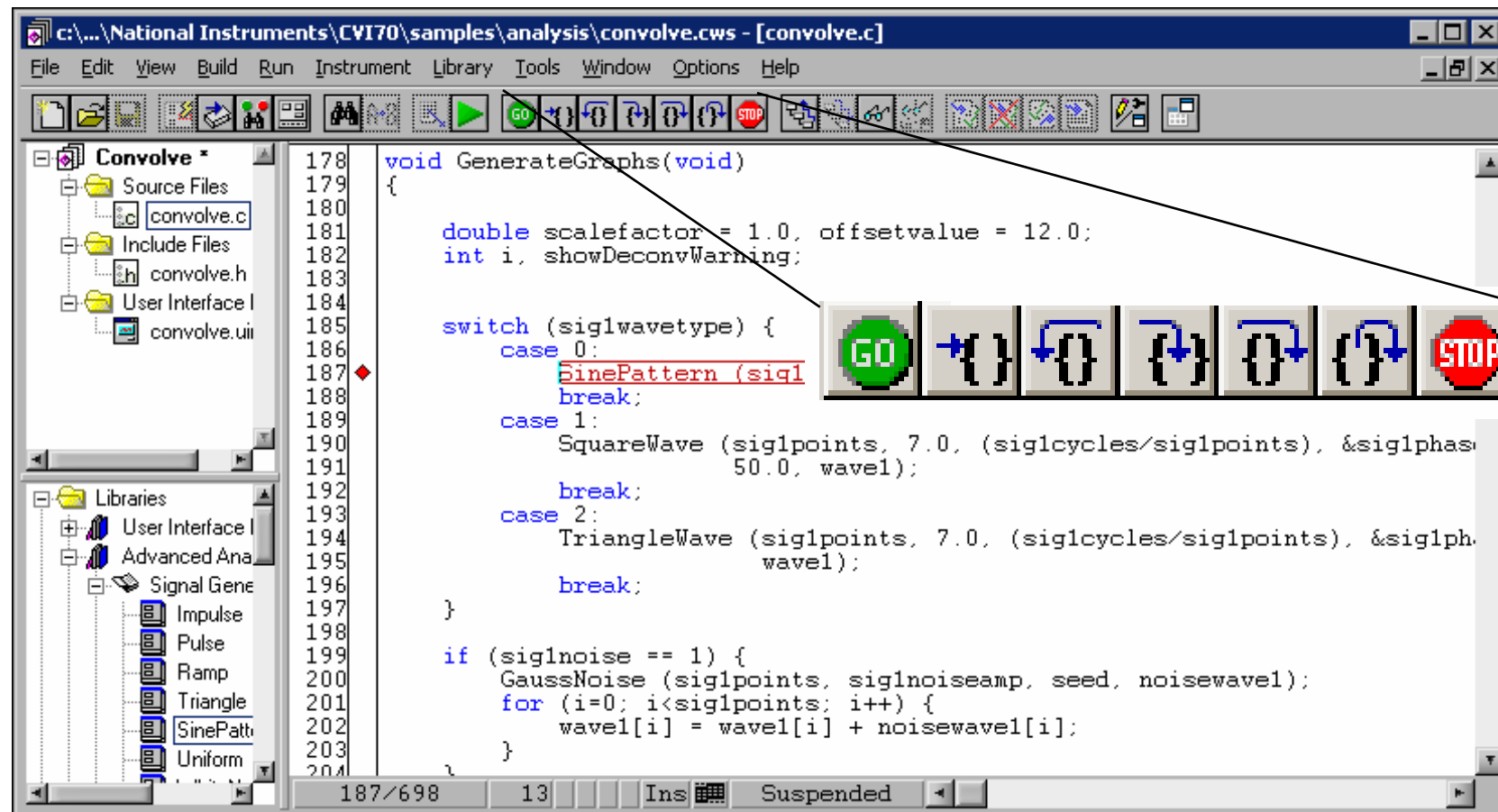
- Udalosť (event) na ovládacom prvku spôsobí volanie príslušnej Callback funkcie
- Systém riadenia programu udalosťami sa aktivuje funkciou **RunUserInterface**
- Udalosti sa ukladajú do fronty
- Počas výkonu Callbacku ostatné udalosti (Callback funkcie) čakajú.
- Ak je potrebné v tomto čase umožniť vyvolanie iného callbacku, použite **ProcessSystemEvents**, ak je potrebné vykonať vykreslenie **ProcessDrawEvents**
- Ukončenie riadenia programu udalosťami sa vykoná pomocou **QuitUserInterface**

```
1  int stopFlag = 1;
2
3  void RunUserInterface ( )
4  {
5      while (stopFlag)
6          ProcessSystemEvents();
7  }
8
9  void QuitUserInterface ( )
10 {
11     stopFlag = 0;
12 }
```

13/13 1 C Ins

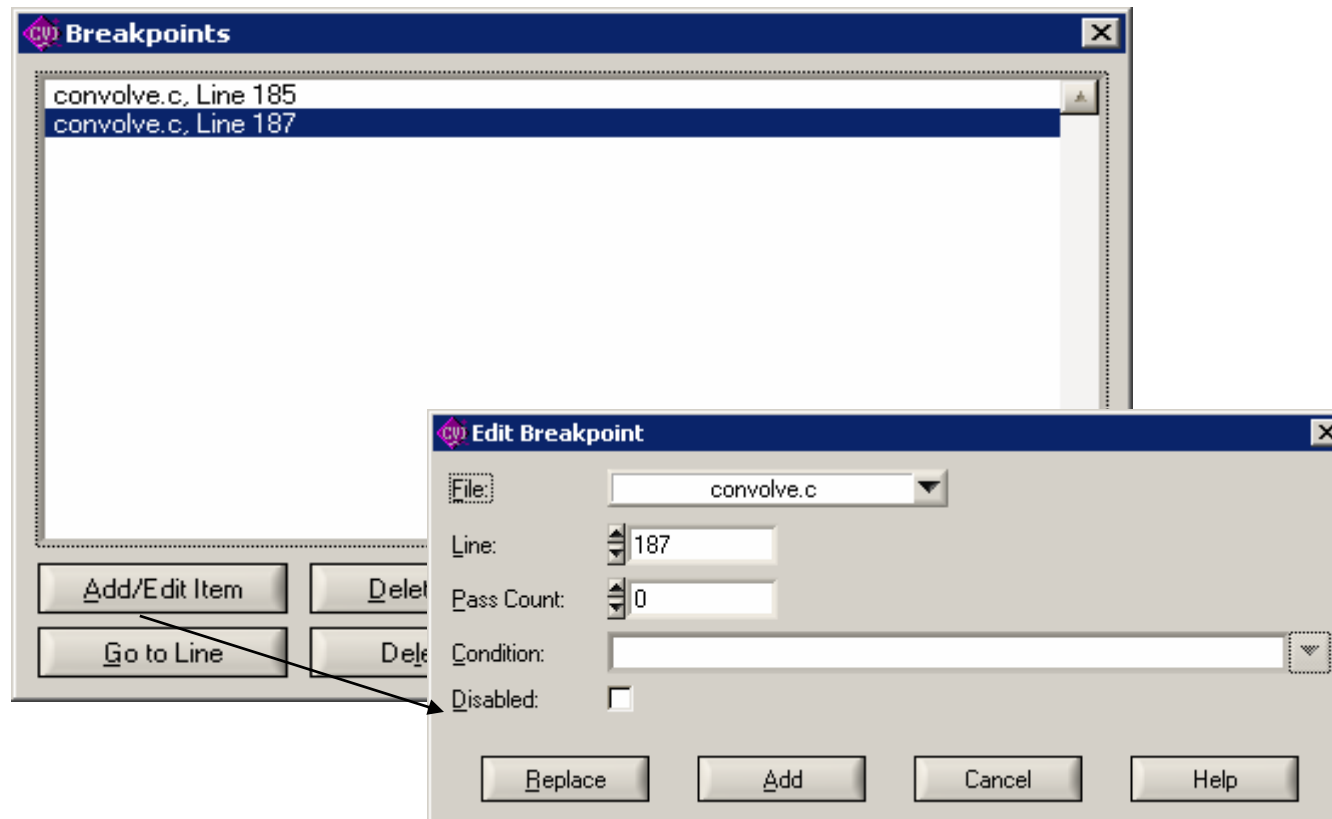
Ladenie

- Krokovanie



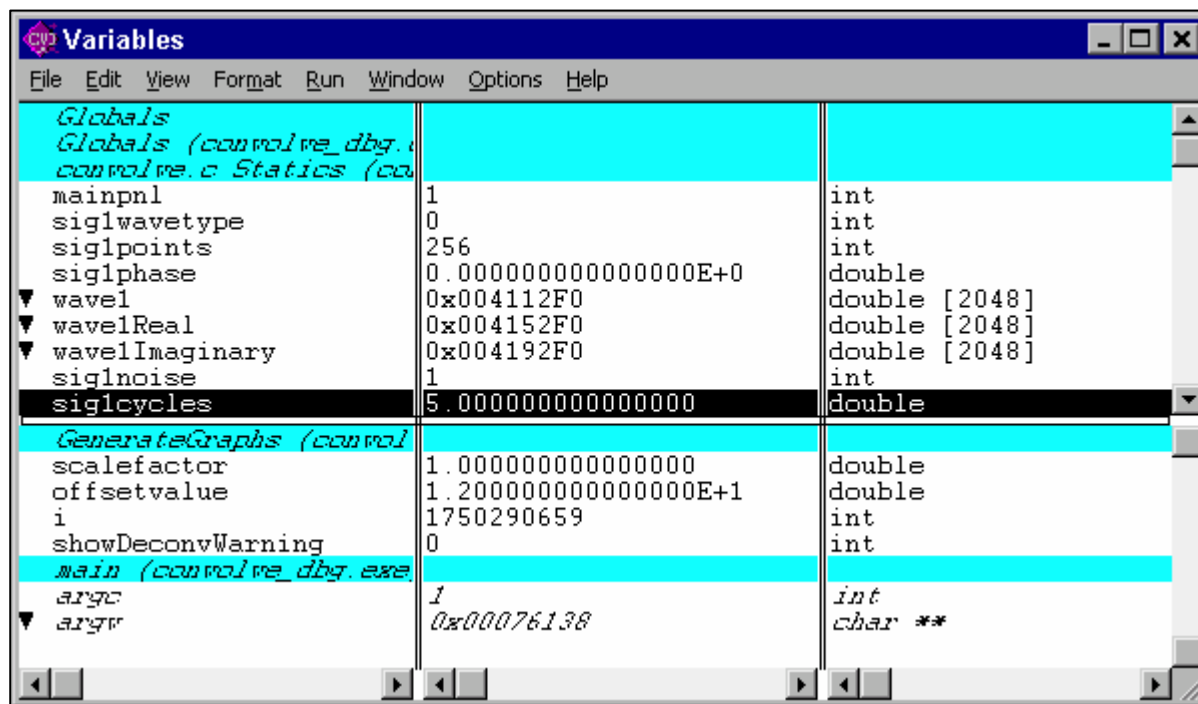
Podmienený breakpoint

- Nastavte jednoduché prerušenie (klik – stĺpec vľavo v Source window
- Zvoľte Run» Breakpoints



Sledovanie hodnôt premenných

- Z menu Window»Variables



The screenshot shows a 'Variables' window with a menu bar (File, Edit, View, Format, Run, Window, Options, Help) and a list of variables. The 'sigcycles' variable is selected and highlighted in black. The table below represents the data shown in the window:

Variable	Value	Type
<i>Globals</i>		
<i>Globals (convolve_dbg.c)</i>		
<i>convolve.c Statics (convolve.c)</i>		
mainpnl	1	int
sigwavetype	0	int
siglpoints	256	int
siglphase	0.0000000000000000E+0	double
▼ wavel	0x004112F0	double [2048]
▼ wavelReal	0x004152F0	double [2048]
▼ wavelImaginary	0x004192F0	double [2048]
siglnoise	1	int
sigcycles	5.0000000000000000	double
<i>GenerateGraphs (convolve_dbg.c)</i>		
scalefactor	1.0000000000000000	double
offsetvalue	1.2000000000000000E+1	double
i	1750290659	int
showDeconvWarning	0	int
<i>main (convolve_dbg.exe)</i>		
argc	1	int
▼ argv	0x00076138	char **

- Typ: presuň kurzor nad premennú v *.c – zobrazí sa jej hodnota

Komunikácia s hardvérom

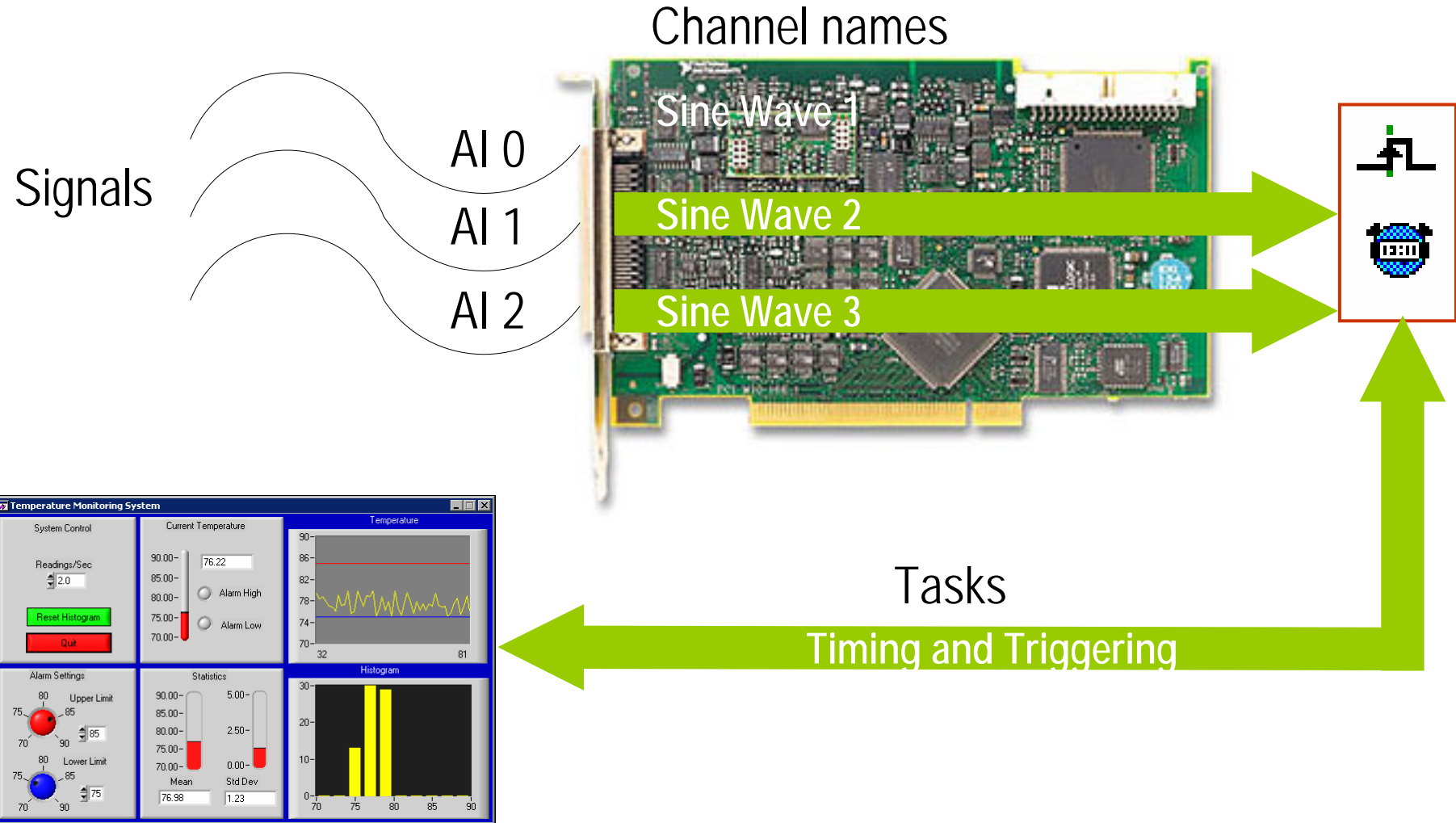
- Princíp: volanie služieb z ovládačov integrovaných v OS (driver – obyčajne .dll) priamo alebo cez vysokoúrovňové funkcie v statických knižniciach (.lib). Drivery od výrobcu hardvéru
- Multifunkčné karty od NI – driver v NI DAQmX (staršie traditional NI DAQ)
- Špeciálne funkcie jednotlivé rozhrania (RS, GPIB, ...) ¹
- VISA – knižnica zovšeobecnených komunikácií po ľubovoľnom rozhraní¹

¹pozri príslušné prednášky

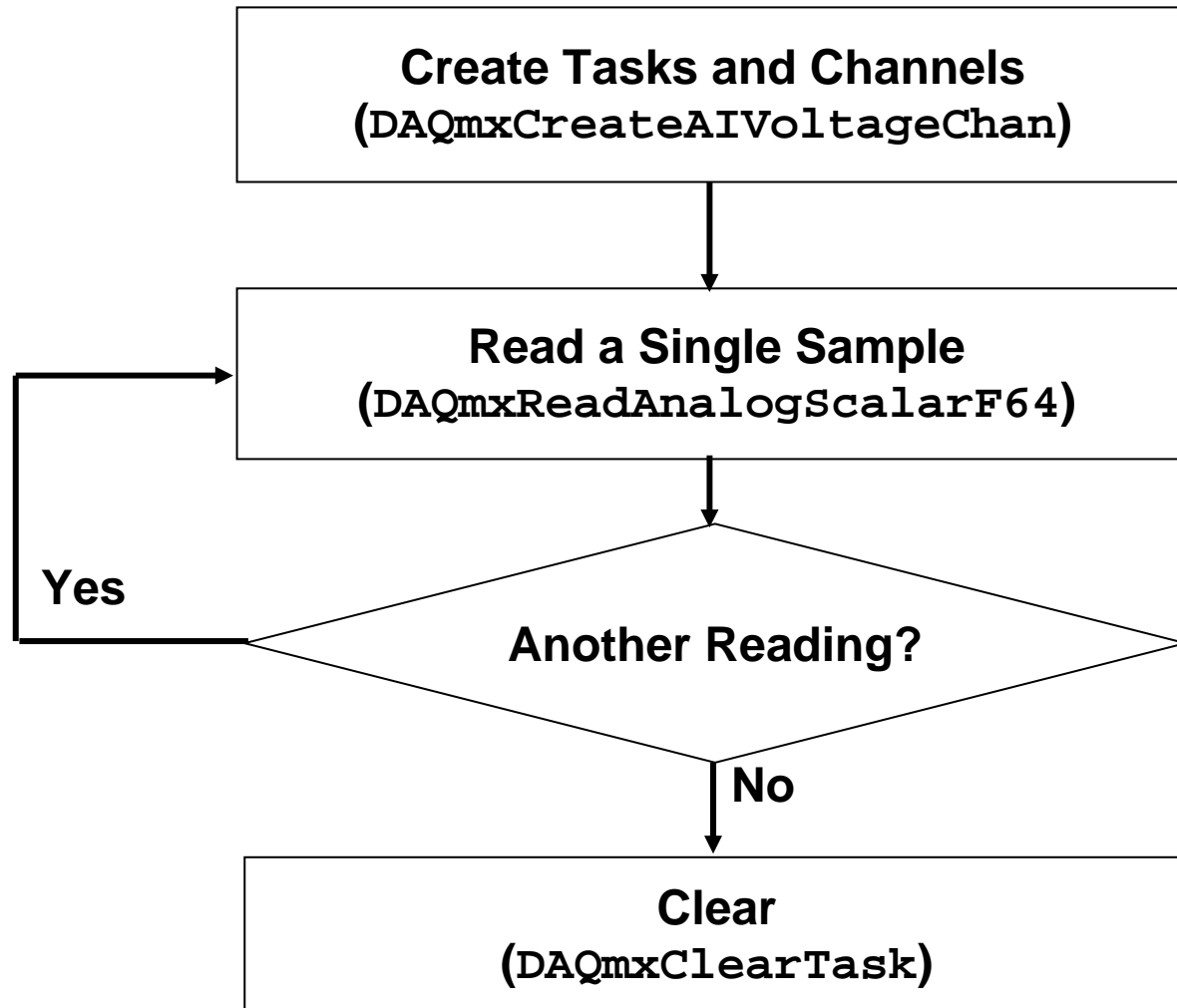
Komunikácia s multifun. DAQ

- Nainštalovaná a skonfigurovaná karta (aj virtuálna) – Measurement and automation explorer – priradené označenie napr. „Dev1“
- Samotný program:
 - „manuálne“ zostavenie programu z funkcií v knižnici DAQmX
 - Tools»Create/Edit DAQmX Task

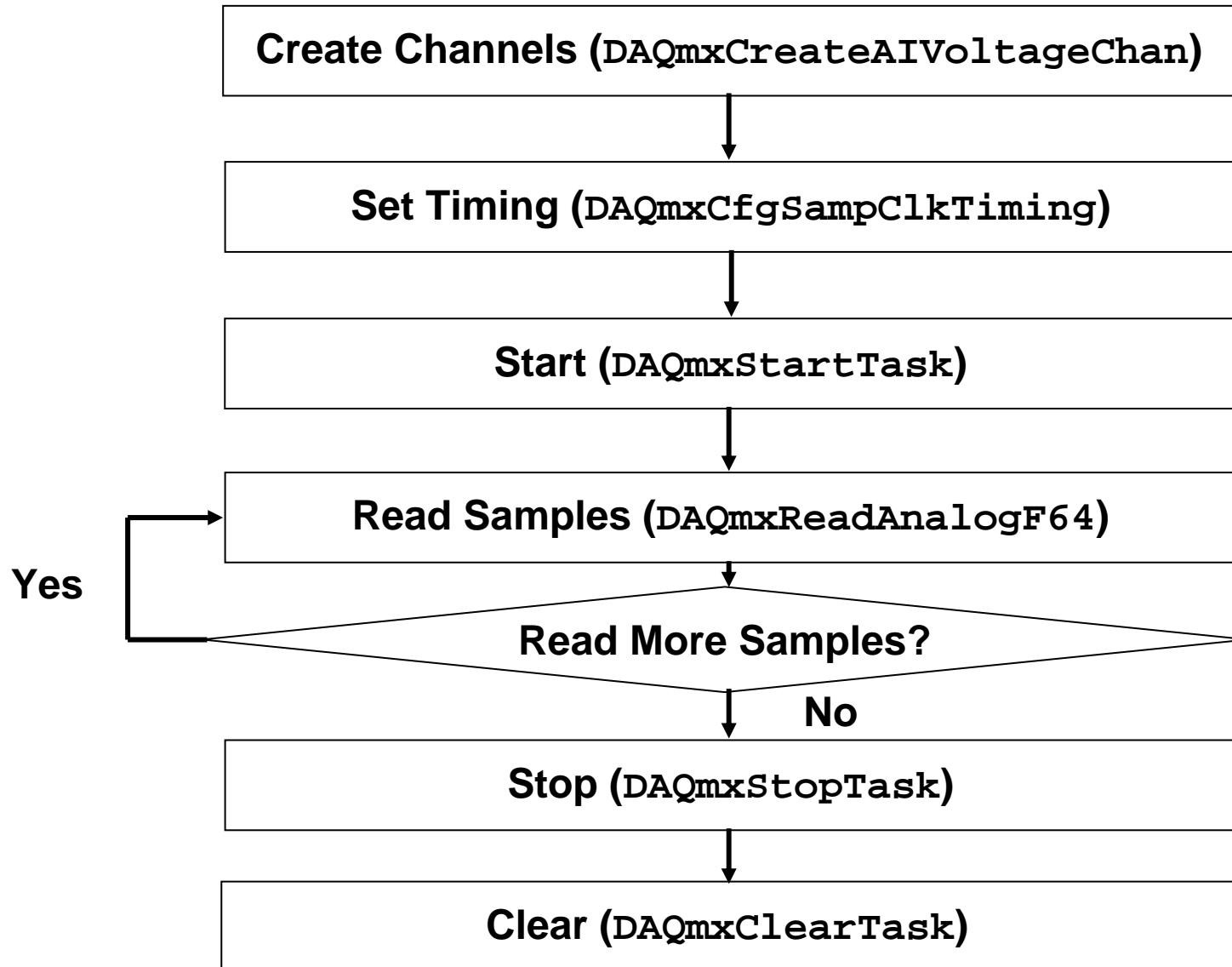
Kanály a úlohy



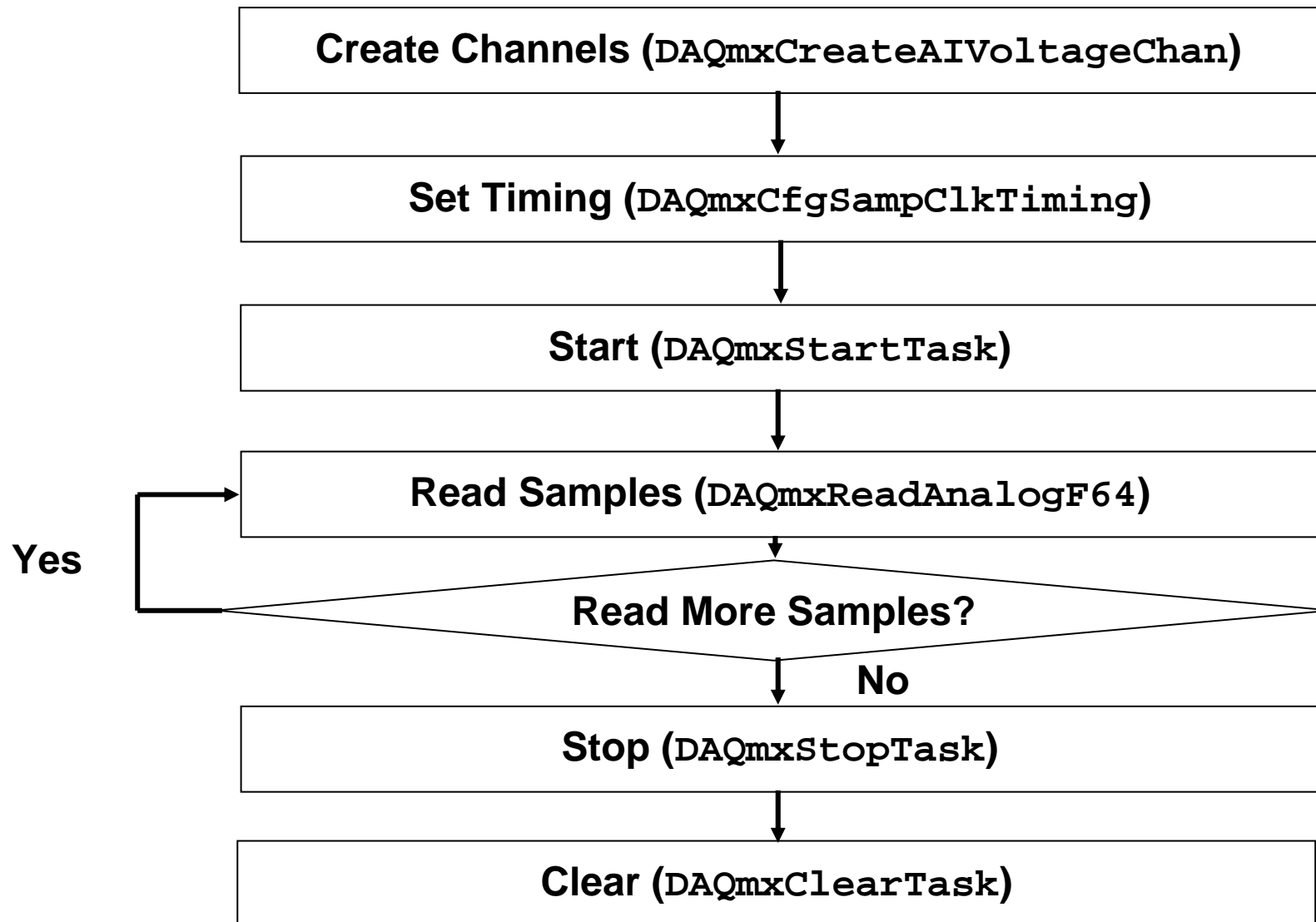
Snímanie jednej hodnoty (vzorky)



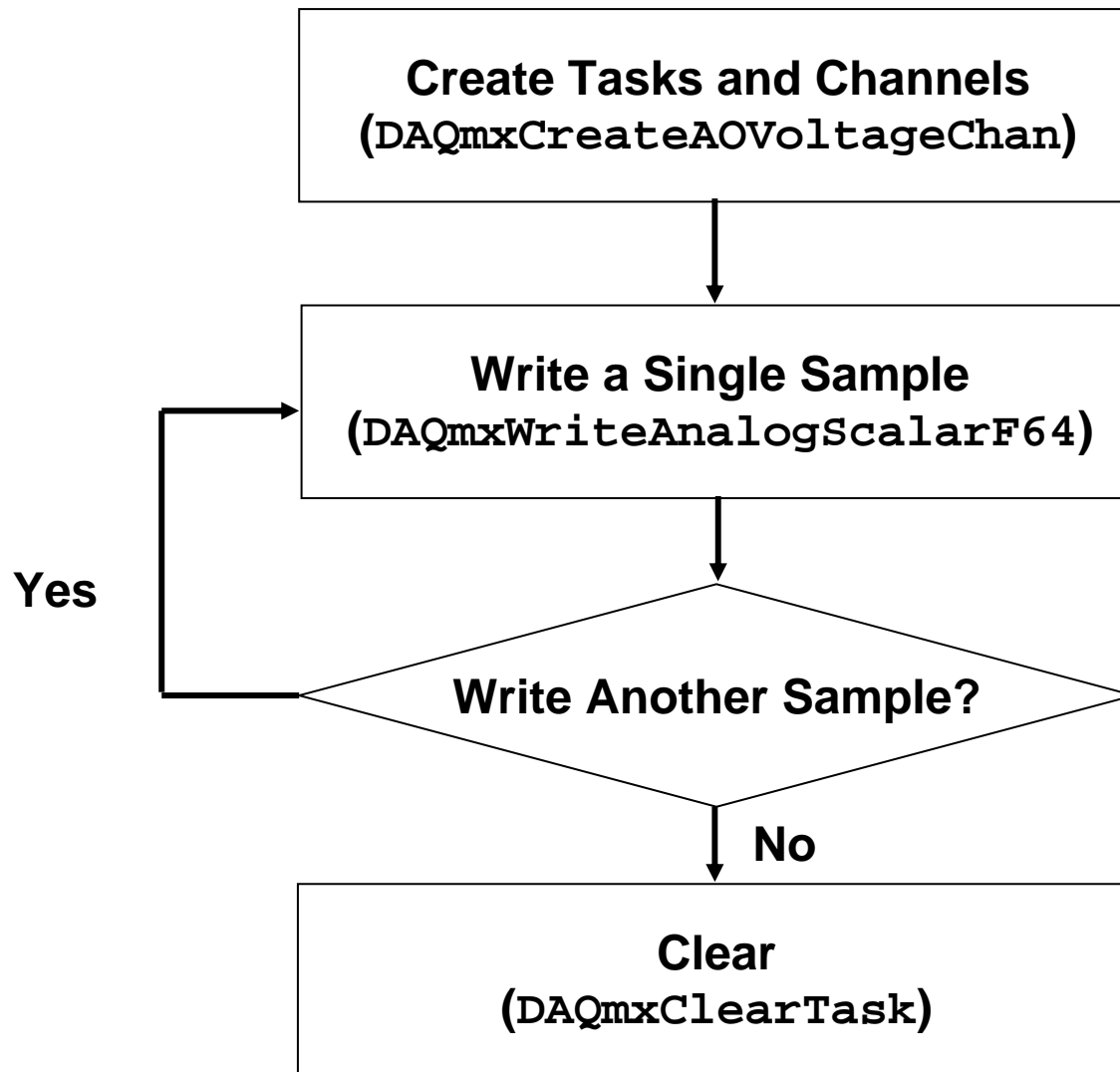
Viacbodové/hodnotové snímanie



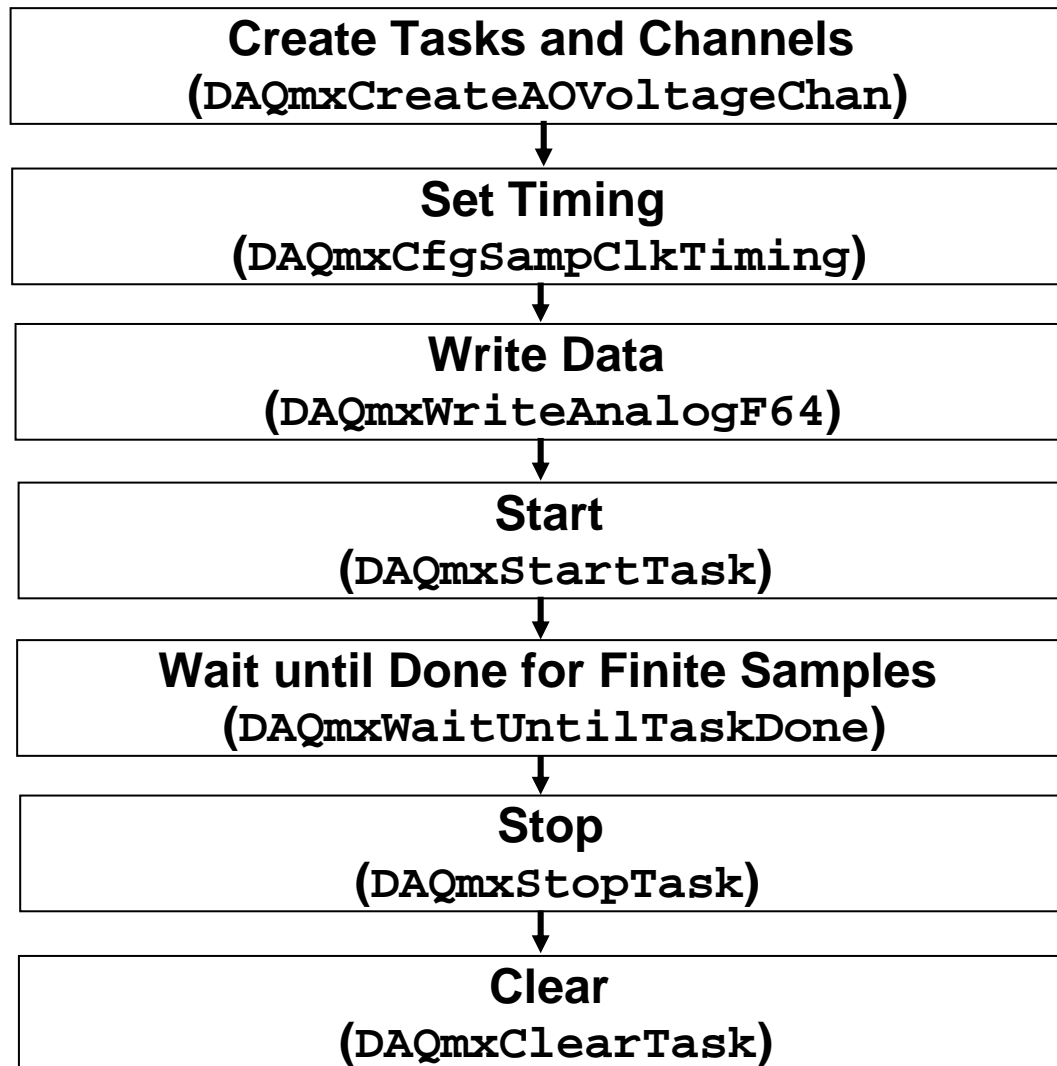
Kontinuálne analógové snímanie



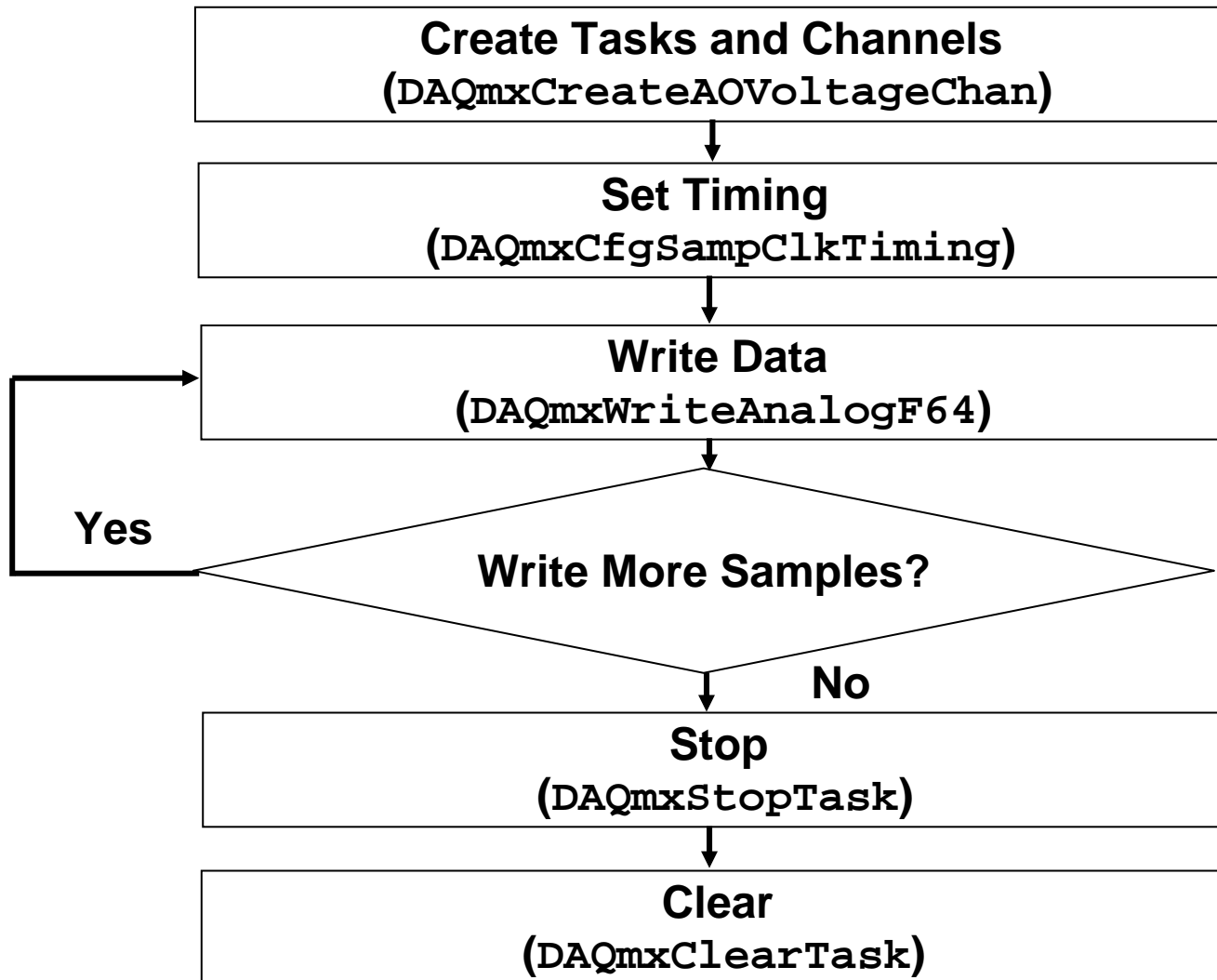
Jednoduchý analógový výstup



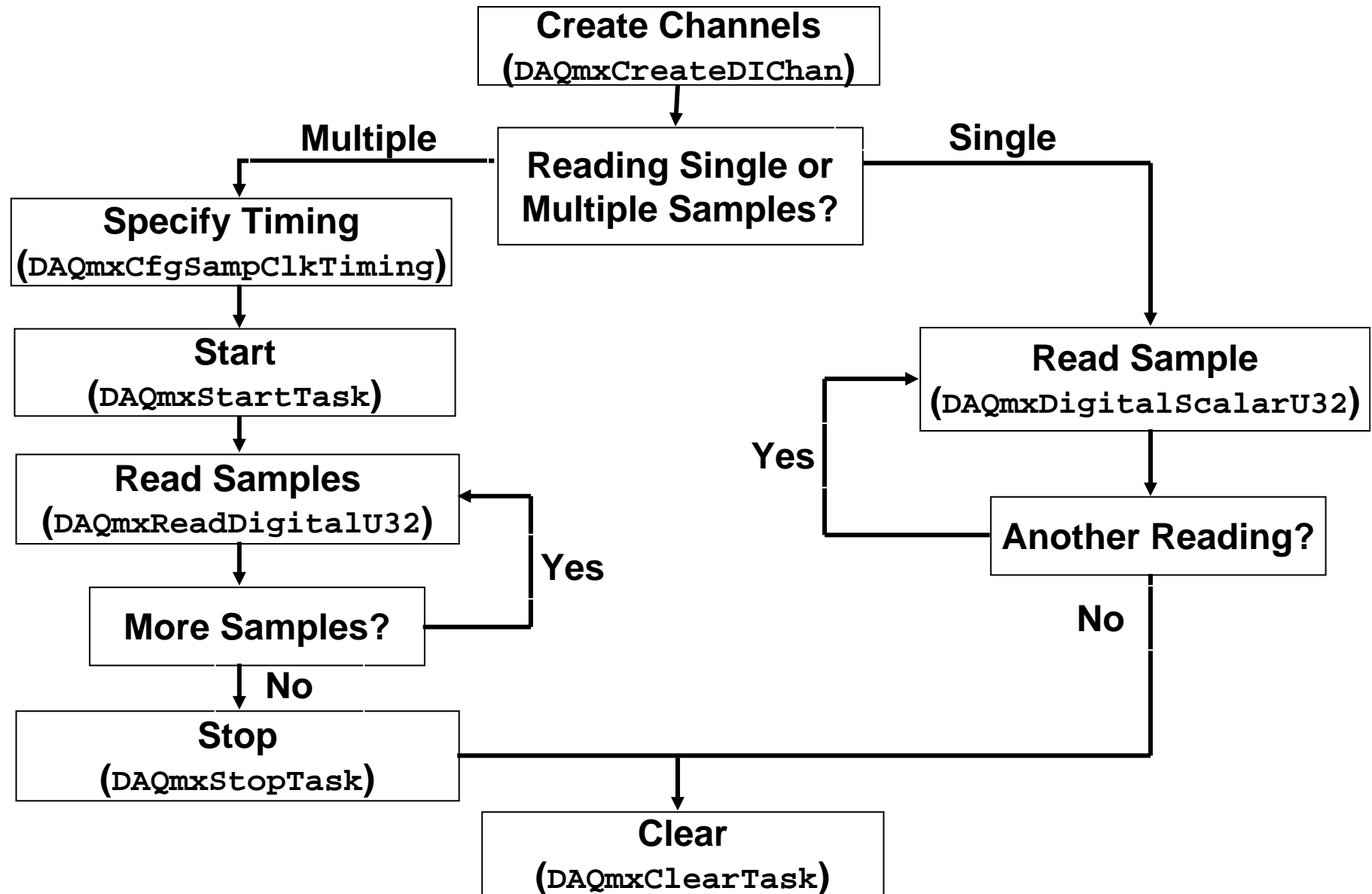
Viachodnotový analógový výstup



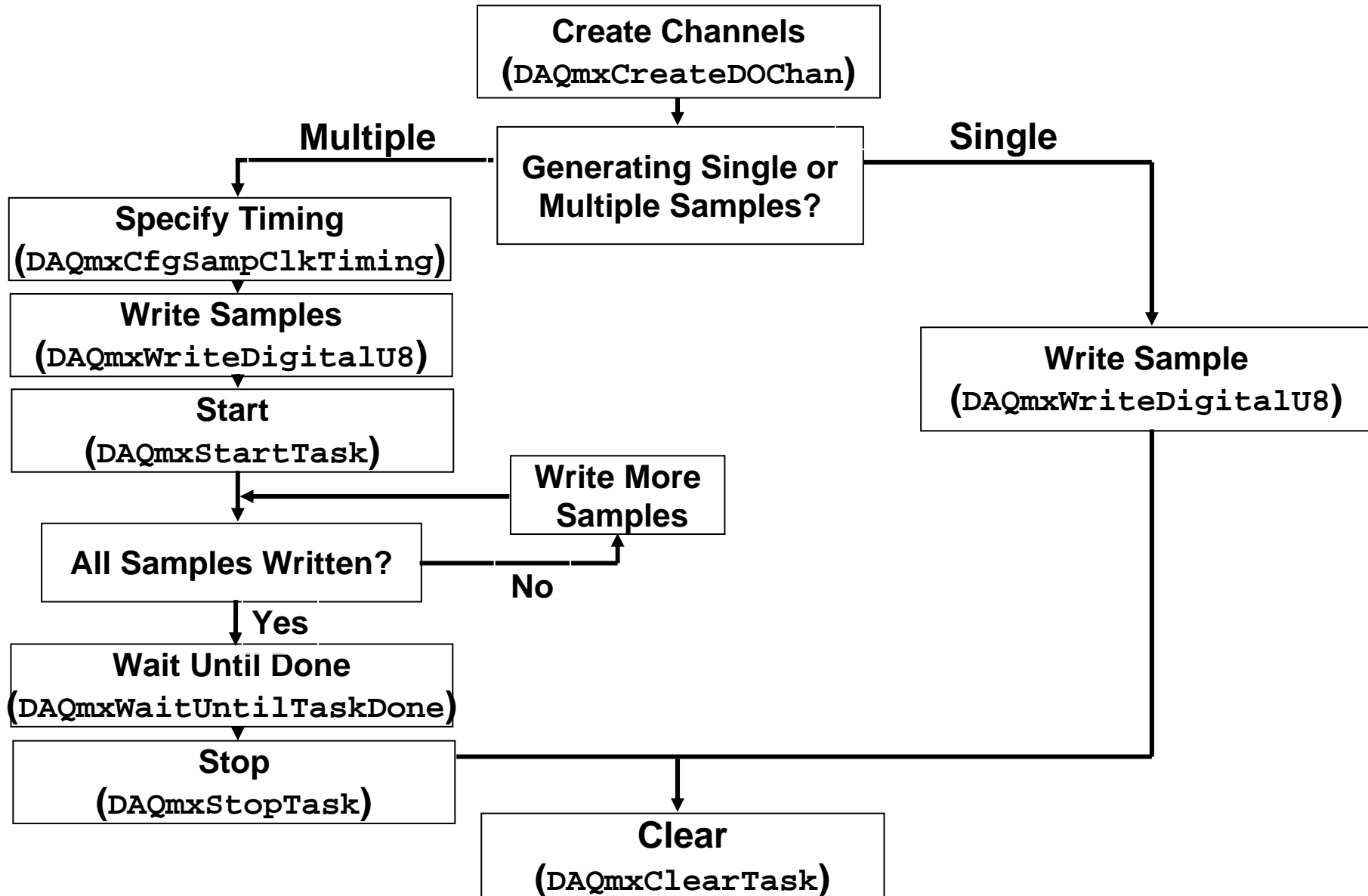
Kontinuální viacbodový analógový výstup



Digitálny vstup



Digitálny výstup

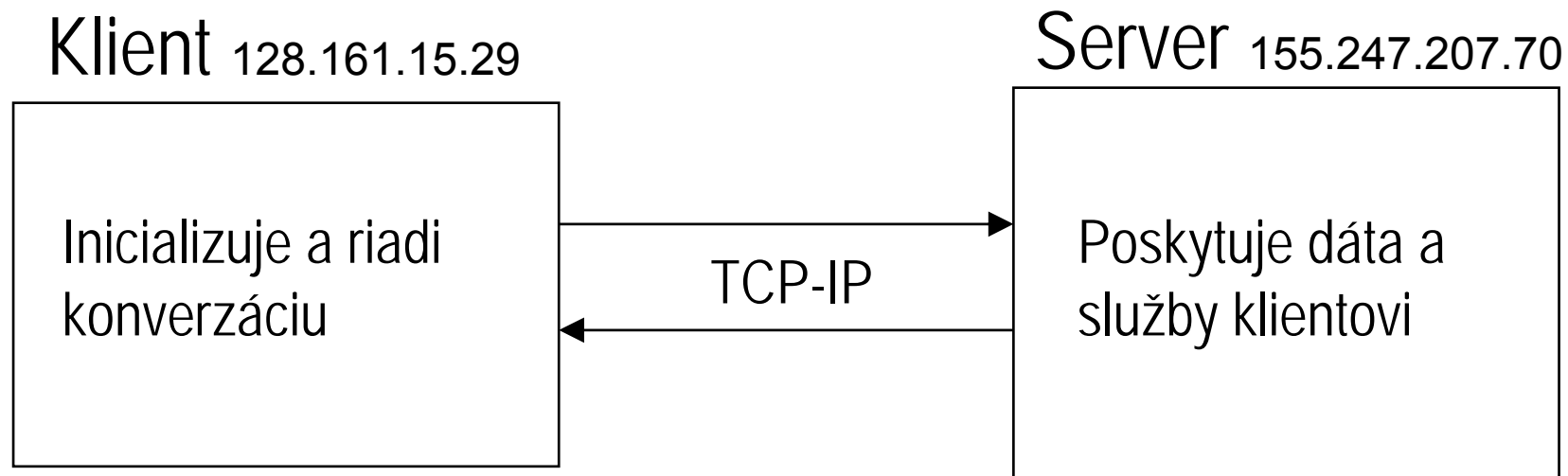


Komunikácia po Internete

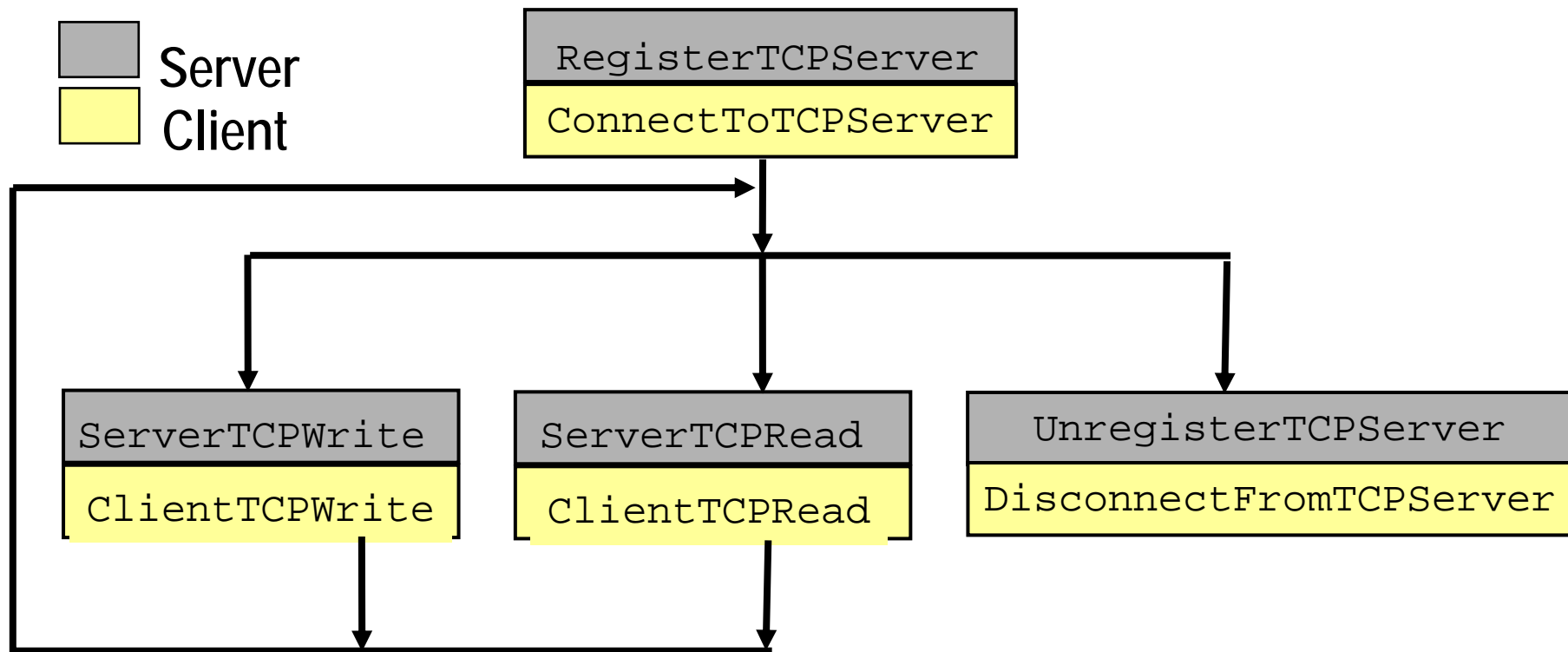
- Základné funkcie na úrovni protokolov TCP/IP v knižnici TCP:
 - Klient, server, podporné a doplnkové funkcie
- Vysokoúrovňové funkcie v knižnici Internet:
 - ftp, telnet, email, web

Princíp komunikácie

- Server aj klient musia mať svoju IP adresu
- Server po spustení (registrácii) čaká na pripojenie klienta
- Po pripojení je komunikácia veľmi jednoduchá ako čítanie a zápis dát do súboru = server dostáva požiadavky od klienta, ktoré vykonáva a výsledky posiela klientovi



Základné komunikačné funkcie



- Výhodná inštalácia Callback funkcie pre server

```
int TCPcallback (int handle, int xType,  
                int errCode, void *callbackData);
```

xType = udalosť, napr. pripojenie/odpojenie klienta, boli prijaté dáta od klienta, ...

Čo ďalej?

- Naštudovať Getting start ...
- Naštudovať príklady v adresári Examples a na web stránke www.ni.com
- Tvorit' jednoduché programy