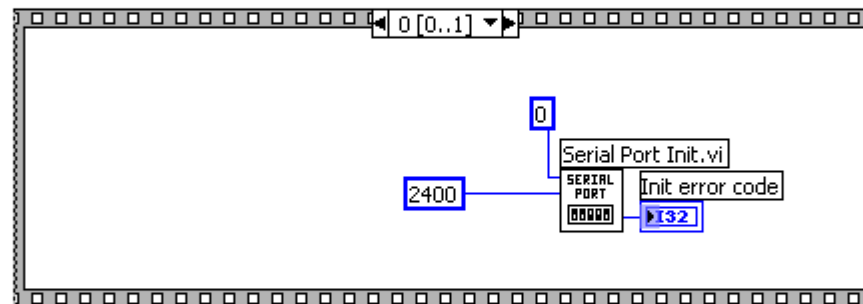


3 последовательные структуры:

Инициализация порта, чтение состояния прибора, передача команды

Инициализация порта: задание номера порта
и скорости передачи, вывод сообщения об ошибке

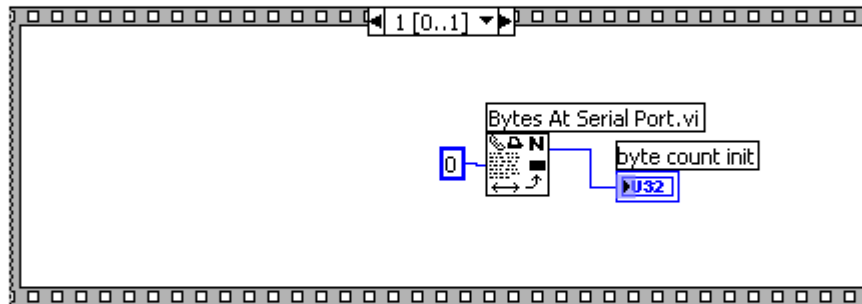


0 [0..2]

Инициализация порта:

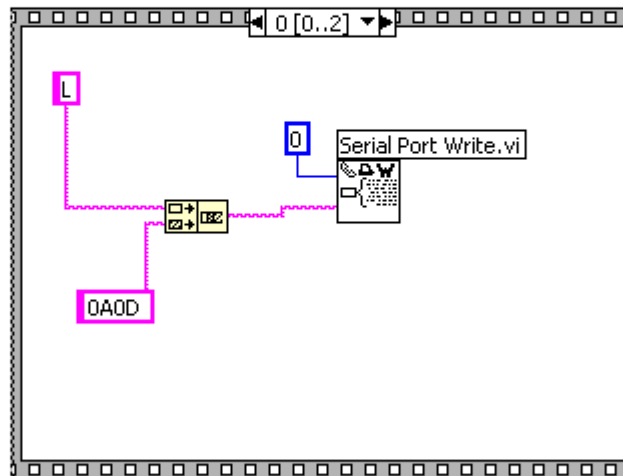
Вывод количества байт, переданных при инициализации

1 [0..1]



чтение состояния прибора

чтение состояния прибора:
передача команды «считать состояние» - символ **L**
и «перевод строки, возврат каретки» - **OAOD** (служебные)





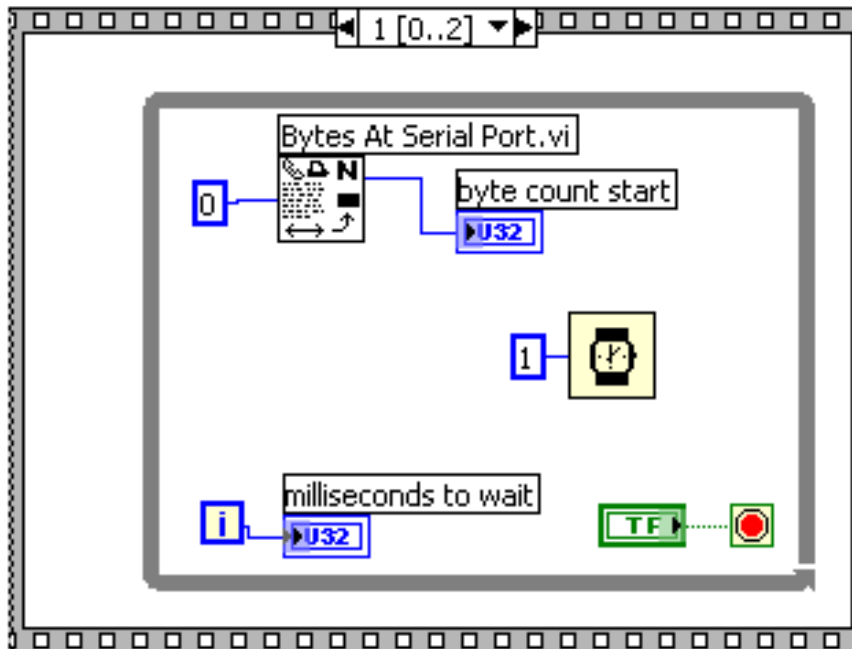
запуск цикла While

индикация числа переданных байт

\счетчика времени

индикация времени передачи команды

кнопка выхода из счетчика времени, если записи не происходит





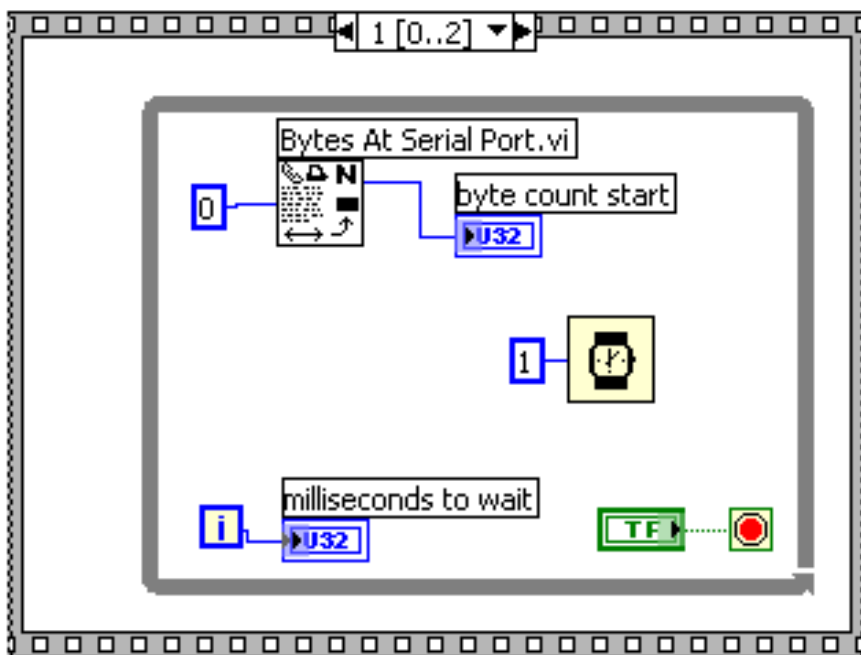
запуск цикла While

индикация числа переданных байт

запуск счетчика времени задержки между циклами While

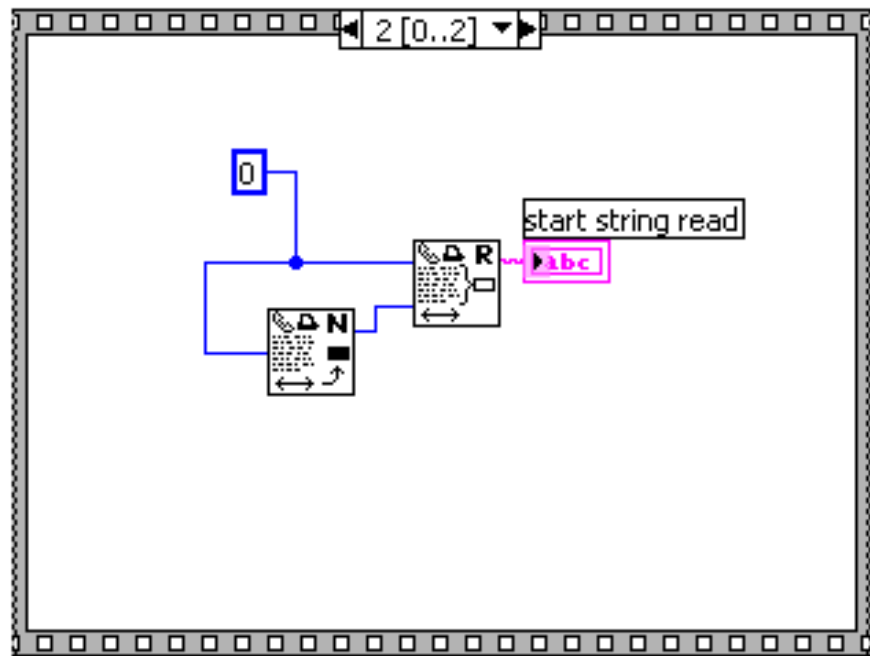
индикация циклов While – времени передачи команды

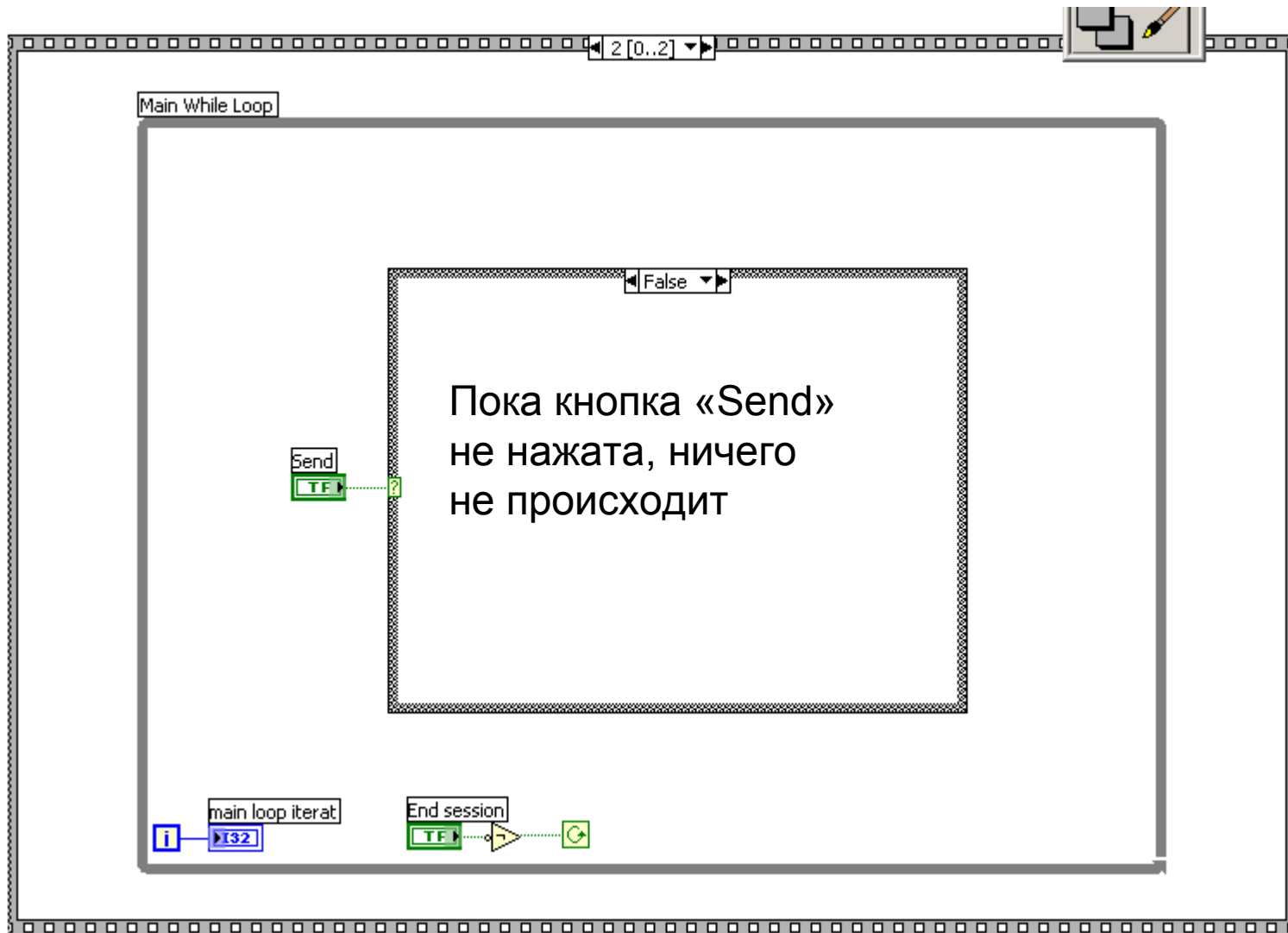
кнопка выхода из цикла, если оператор видит, что записи не происходит с





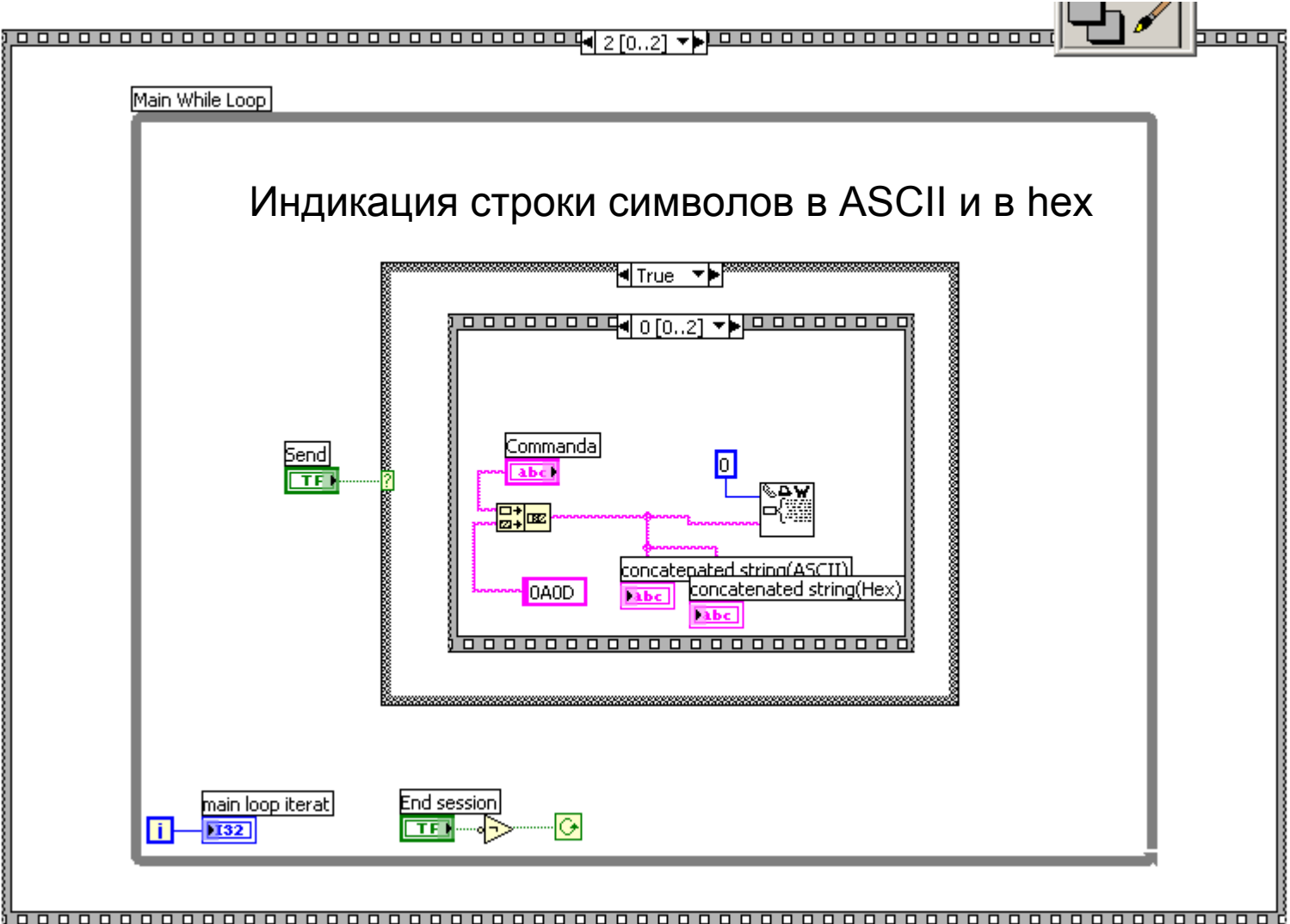
Чтение из порта строки символов, отвечающих на команду **L**



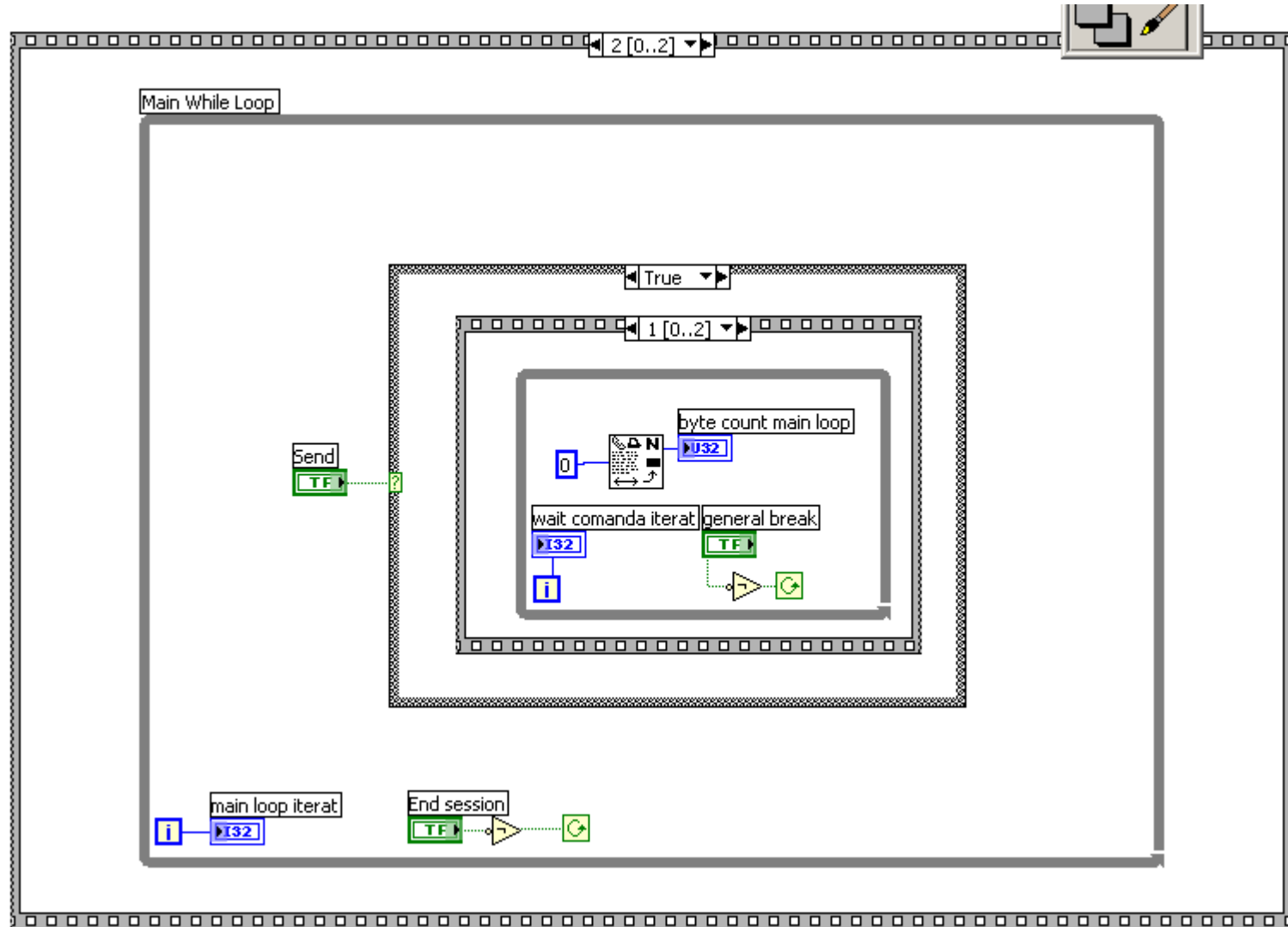


Запись в порт команд из поля ввода строки команды в цикле While и индикация числа циклов, выход из цикла - при нажатой кнопке «End session».

Запись строки команды, введенной предварительно в поле «Commanda», происходит при переключения состояния кнопки «Send»

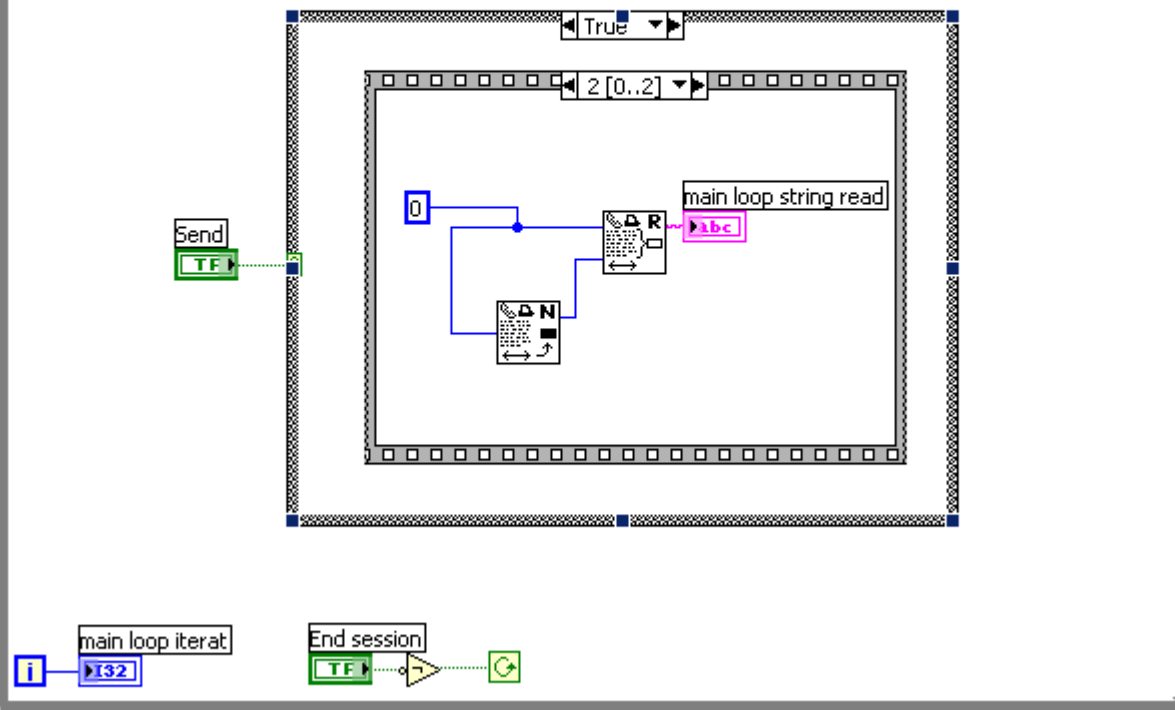


Запись в порт команды во внутреннем цикле While, индикация числа байт,
индикация числа циклов
Выход из внутреннего цикла при нажатии кнопки «generate break»

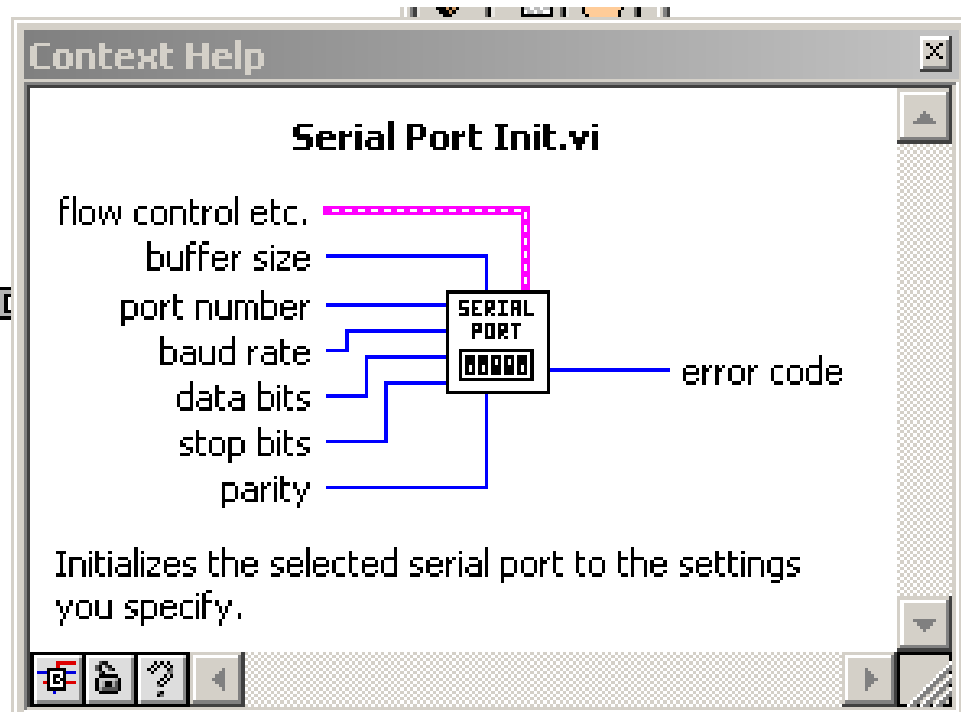
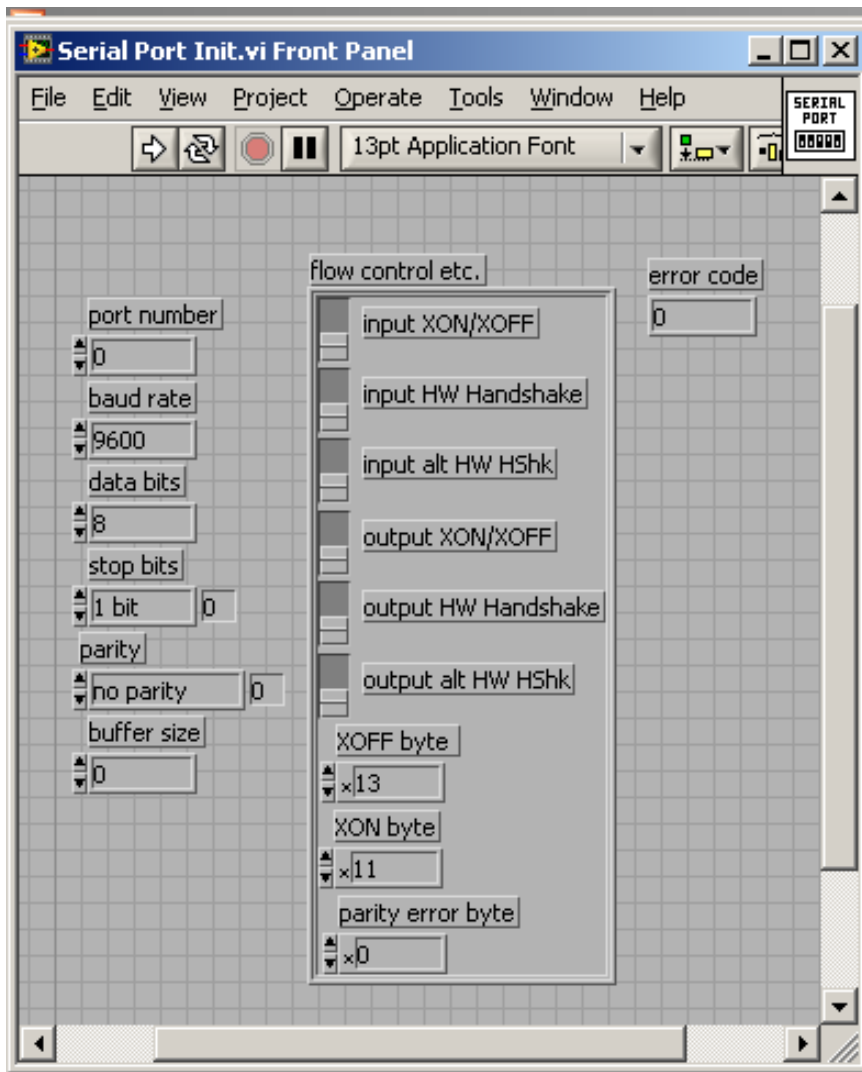


Main While Loop

Чтение символов из порта, число символов задано как входной параметр..



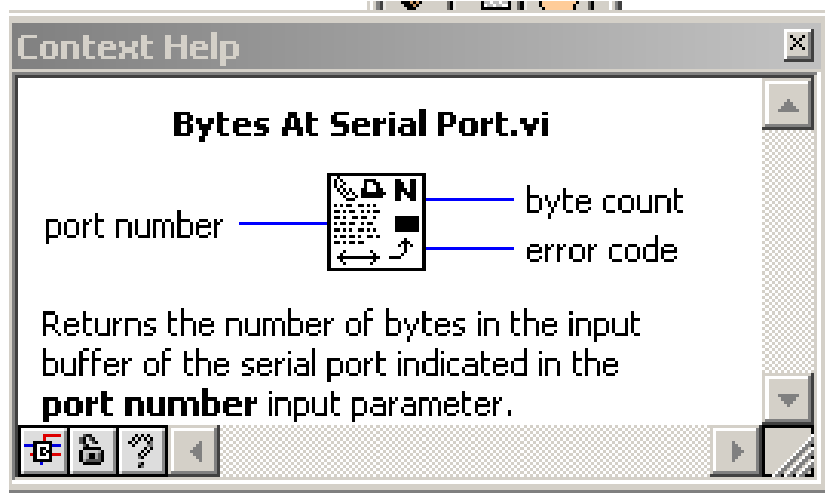
Таким образом, применены встроенные в LabVIEW функции:



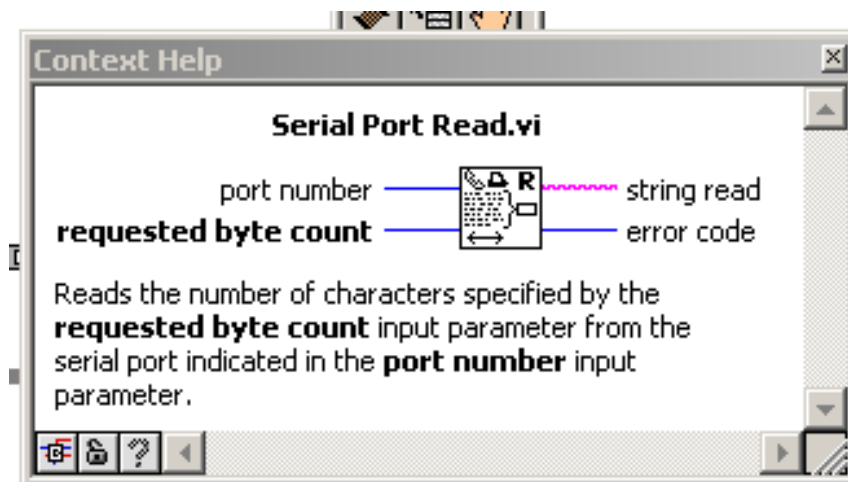
Инициализация порта:

- номер порта,
- скорость передачи данных
- число бит данных
- число стоповых бит
- контроль четности
- размер буфера в байтах
- протокол обмена, те. управление потоком данных
- код ошибки

Таким образом, применены встроенные в LabVIEW функции:

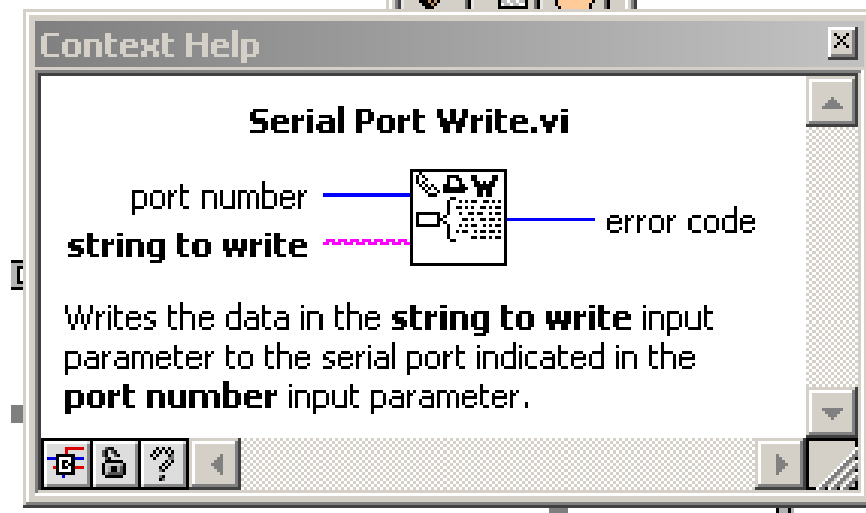


Возвращает количество принятых во входной буфер порта байтов, Используется вместе с функцией Serial Port Read



Считывает заданное число байтов из порта в строку чтения

Таким образом, применены встроенные в LabVIEW функции:



Записывает данные в виде строки записи в порт

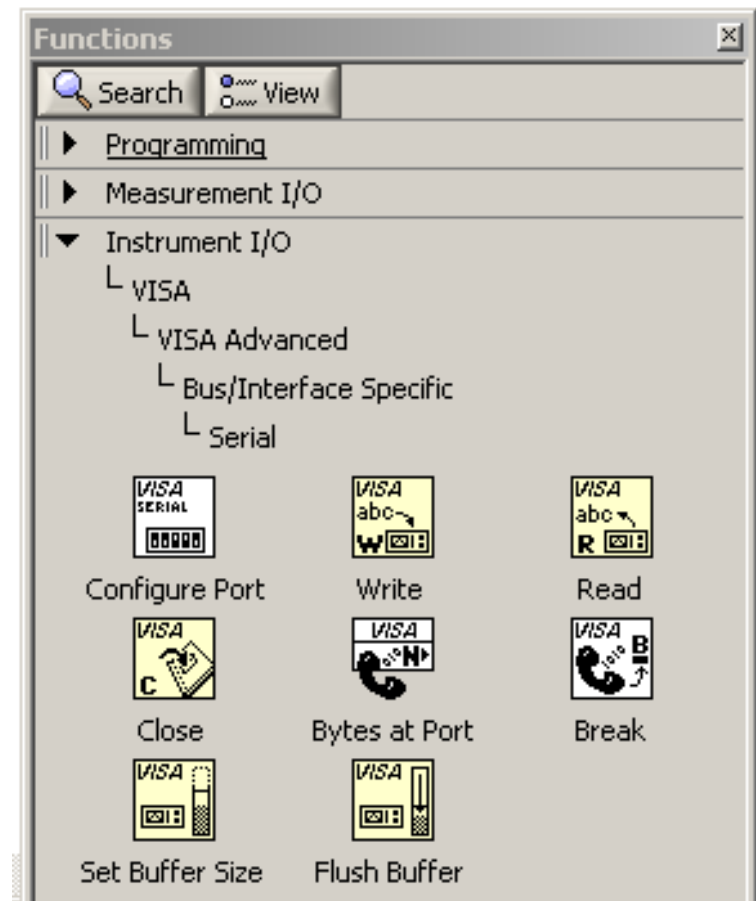
Другие функции:

Serial Port Break – приостанавливает работу порта, формируя обрыв линии,

Close Serial Driver – корректно закрывает порт и освобождает ресурс

В поздних версиях LabVIEW применяется стандартный программный интерфейс приложение API VISA (Virtual Instrument Software Architecture):

- для интерфейса RS-232 в отдельной палитре имеется специальный набор функций VISA, аналогичных функциям драйвера последовательного интерфейса:



VISA Configure Serial Port

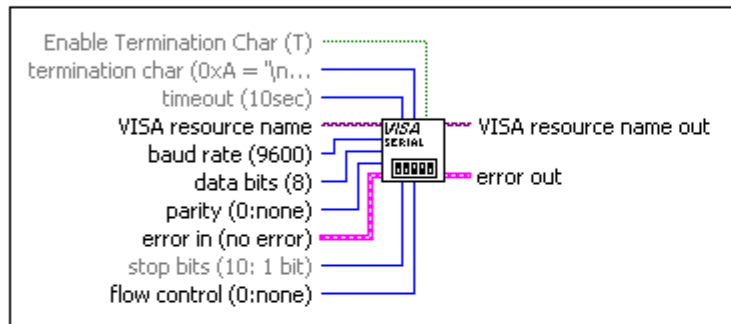
Initializes the serial port specified by **VISA resource name** to the specified settings. The VISA class you wire to the **VISA resource name** input determines the polymorphic instance to use. [Example](#)

Use the pull-down menu to select an instance of this VI.

Select an instance

Place on the block diagram Find on the **Functions** palette

VISA Configure Serial Port



TF

Enable Termination Char prepares the serial device to recognize **termination char**. If TRUE (default), the VI_ATTR_ASRL_END_IN attribute is set to recognize the termination character. If FALSE, the VI_ATTR_ASRL_END_IN attribute is set to 0 (None) and the serial device does not recognize the **termination char**.

U8

termination char calls for termination of the read operation. The read operation terminates when the **termination char** is read from the serial device. 0xA is the hex equivalent of a linefeed character (\n). Change the **termination char** to 0xD for message strings that terminate with a carriage return (\r).

U32

timeout sets the timeout value for the write and read operations.

I70

VISA resource name specifies the resource to be opened. This control also specifies the session and class. Refer to [VISA Resource Name Control](#) for more information.

U32

baud rate is the rate of transmission. The default is 9600.

U16

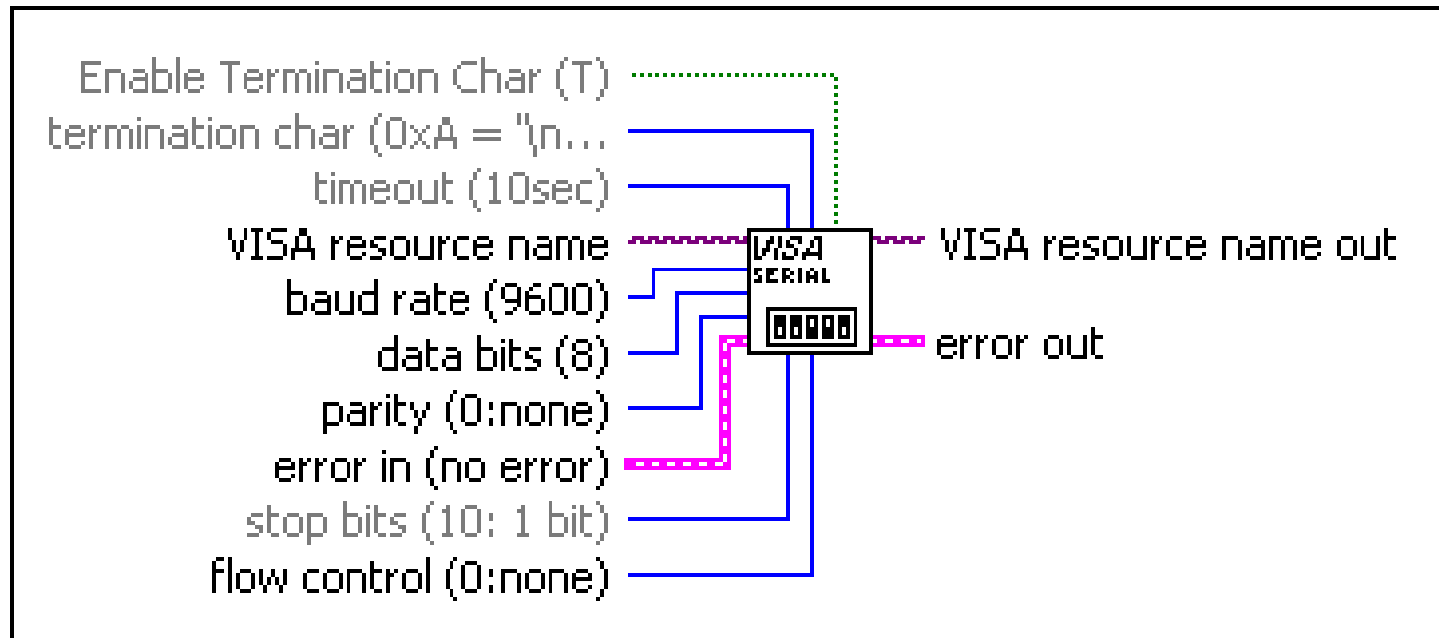
data bits is the number of bits in the incoming data. The value of **data bits** is between five and eight. The default value is 8.

U8

parity specifies the parity used for every frame to be transmitted or received.

0	no parity (default)
1	odd parity

VISA Configure Serial Port



Инициализирует последовательный порт, определяемый с помощью входа VISA resource name -

Enable Termination Chare - если True, то готовность к распознаванию символа разрешения (termination chare – 0xA и 0xD)

Timeout – время ожидания выполнения операций записи или чтения

Untitled 1 Block Diagram *

File Edit View Project Operate Tools Window Help

13pt Application Font

Globals) (All global members)

Tools

PICKING

VISA Configure Serial Port (Instr).vi Front Panel

File Edit View Project Operate Tools Window Help

13pt Application Font

VISA resource name: COM3 (dropdown menu open with LPT1 and Refresh options)

timeout (10sec): 10000

termination char (0xA = '\n' = LF): \A

Enable Termination Char (T): ON

VISA resource name out: % (dropdown menu)

data bits (8): 8

parity (0:none): None

stop bits (10: 1 bit): 1.0

flow control (0:none): None

error in (no error): status [checked] code 40 source

error out: status [checked] code 40 source

New



- Files
- Projects
- Workspaces
- Other Documents

- ATL COM AppWizard
- Cluster Resource Type Wizard
- Custom AppWizard
- Database Project
- DevStudio Add-in Wizard
- Extended Stored Proc Wizard
- ISAPI Extension Wizard
- Makefile
- MFC ActiveX ControlWizard
- MFC AppWizard (dll)
- MFC AppWizard (exe)
- New Database Wizard
- Utility Project
- Win32 Application
- Win32 Console Application
- Win32 Dynamic-Link Library
- Win32 Static Library

Project name:
AnotherSample

Location:
C:\CURS-11\AnotherSample

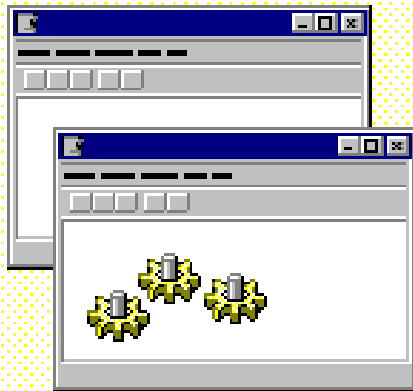
- Create new workspace
- Add to current workspace
- Dependency of:
[Empty dropdown]

Platforms:
 Win32

OK Cancel

What kind of DLL would you like to create ?

- An empty DLL project.
- A simple DLL project.
- A DLL that exports some symbols.



< Back

Next >

Finish

Cancel

File Edit View Insert Project Build Tools Window Help



(Globals)

(All global members)

DllMain



```
// AnotherSample.cpp : Defines the entry point for the DLL application.
```

```
//
```

```
#include "stdafx.h"
```

```
BOOL APIENTRY DllMain( HANDLE hModule,  
                      DWORD ul_reason_for_call,  
                      LPVOID lpReserved  
                      )
```

```
{  
    return TRUE;  
}
```

```
|
```

ClassView FileView

Build Debug Find in Files 1 Find in Files 2



(Globals)

(All global members)

testFunction

Workspace 'AnotherSample': 1 project(s)

AnotherSample files

- Source Files
 - AnotherSample.cpp
 - StdAfx.cpp
- Header Files
- Resource Files
- ReadMe.txt

```
// AnotherSample.cpp : Defines the entry point for the DLL application.
//
#include "stdafx.h"

BOOL APIENTRY DllMain( HANDLE hModule,
                      DWORD  ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    return TRUE;
}

__declspec(dllexport)
void testFunction(long a, long b, long* sum, long* prod)
{
    *sum = a + b;
    *prod = a * b;
}
```

ClassView FileView

Sample.vi Block Diagram *

File Edit View Project Operate Tools Window Help

13pt Application Font

Sample.vi Front Panel

File Edit View

String

C:\C...

Stri

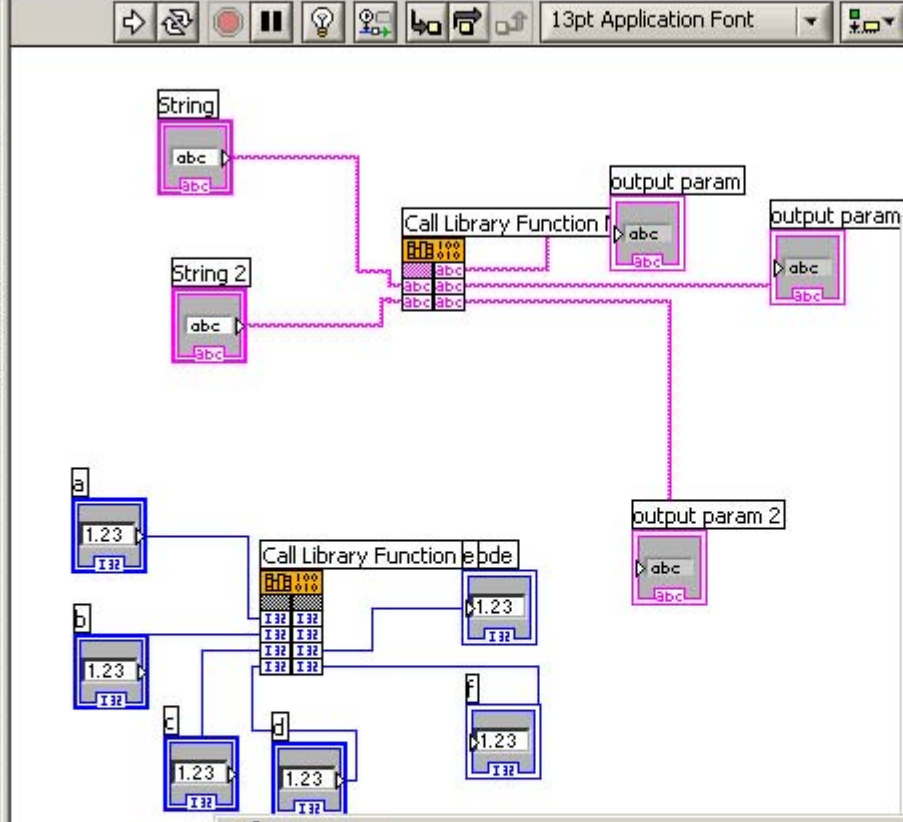
C:\C...

a

0

b

0



Tools

Разметк

Примени

Макеть

Connectivity

Libraries & Executables

Libraries & Ex...

Call Library F...

Code Interfa...

System Exec.vi

.NET

Input Device ...

ActiveX

Windows Reg...

Functions

Programming

Structures

Numeric

Comparison

File I/O

Synchronization

Measurement I/O

Instrument I/O

Mathematics

Signal Processing

Data Communication

Connectivity

Express

Favorites

Select a VI...

Array

Boolean

Timing

Waveform

Graphics & So...

Cluster & Vari...

String

Dialog & User...

Application C...

Report Gener...

LabVIEW Help

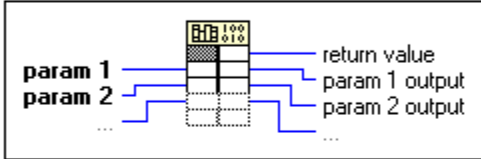
Hide Locate Back Forward Options

Contents Index Search Favorites

- LabVIEW Help
- Finding Example VIs
- Glossary
- LabVIEW 8.0 Features and Changes
- Activating LabVIEW Software
- Using Help
- LabVIEW Documentation Resources
- Quick Tips
- Getting Started
- Fundamentals
- VI and Function Reference
 - Programming VIs and Functions
 - Measurement I/O VIs and Functions
 - Instrument I/O VIs and Functions
 - Mathematics VIs
 - Signal Processing VIs
 - Data Communication VIs and Functions
 - Connectivity VIs and Functions
 - .NET Functions
 - ActiveX Functions
 - Input Device Control VIs
 - Libraries & Executable VIs
 - Call Library Function Node**
 - Code Interface Nodes
 - System Exec
 - Code Interface Nodes
 - Pipes VIs
 - Port I/O VIs
 - Source Control VIs
 - Windows Registry VIs

Call Library Function Node

Calls a DLL or shared library function directly. The Call Library Function Node supports a large number of data types and calling conventions. You can use it to call functions in most standard and custom-made DLLs and shared libraries. If you want to call a DLL that contains [ActiveX](#) objects, use the [Automation Open](#) VI with the [Property Node](#) and the [Invoke Node](#). The Call Library Function Node is expandable and shows data types for the wired inputs and outputs, similar to the [Bundle](#) function. [Details](#) [Example](#)



Place on the block diagram Find on the **Functions** palette

POLY param 1..n are example input parameters of the library function.

POLY return value is an example return value of the library function.

POLY param 1..n output are example output parameters of the library function.

Call Library Function Node Details

The Call Library Function Node consists of pairs of input and output terminals. You can use one or both terminals. If the node does not return **return value**, the top pair of terminals is unused. Each additional pair of terminals corresponds to a parameter in the parameter list of the function you call in top-to-bottom order. You pass a value to the function by wiring to the left terminal of a terminal pair. You read the value of a parameter after the function call by wiring from the right terminal of a terminal pair.

Right-click the node and select **Configure** from the shortcut menu to display the [Call Library Function](#) dialog box, which you can use to specify the library name or path, function name, calling conventions, parameters, and return value for the node. When you click the **OK** button, the node automatically

Tools

(Globals)

Workspace 'AnotherS

Sample.vi Fro

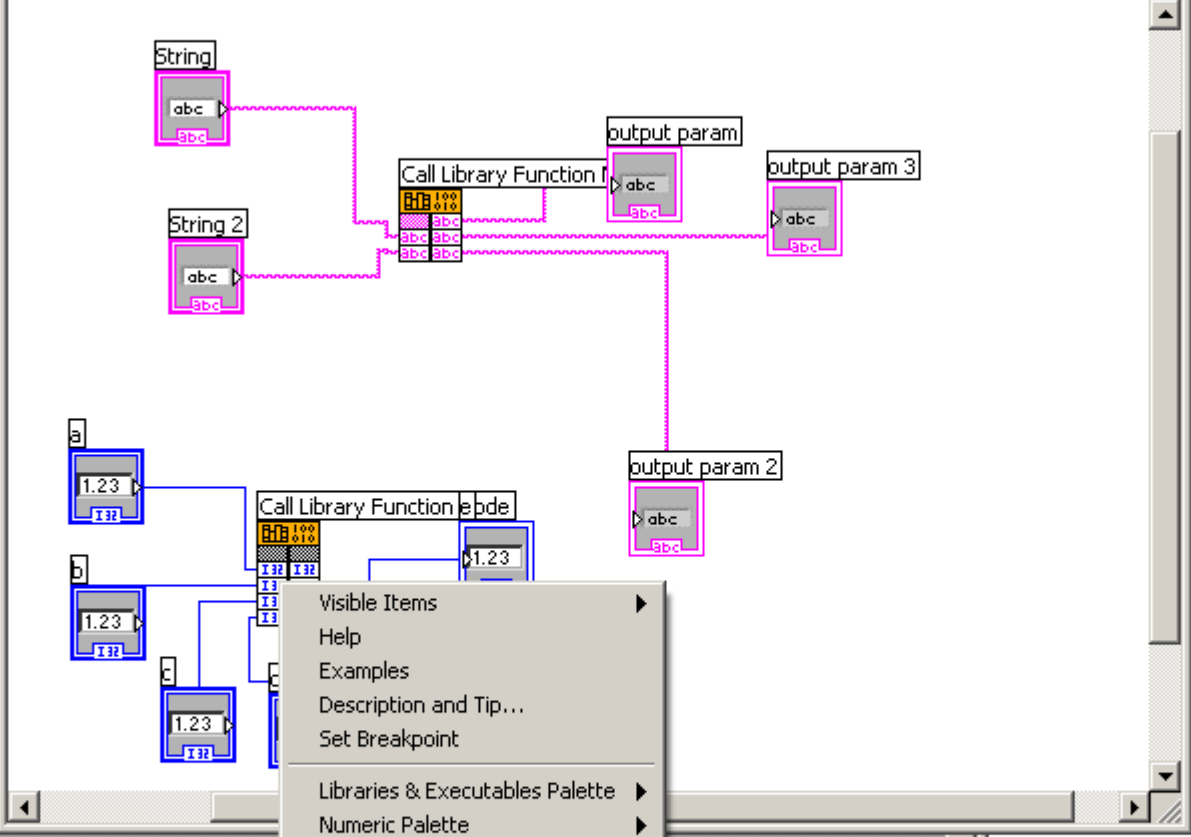
File Edit View

String

C:\C

Stri

C:



- Visible Items
- Help
- Examples
- Description and Tip...
- Set Breakpoint
- Libraries & Executables Palette
- Numeric Palette
- Create
- Replace
- Configure...
- Create .c File...

AnotherSample.dll - 0 error(s), 0 warning(s)



Call Library Function



Library Name or Path

C:\Curs-11\AnotherSample\Debug\AnotherSample.dll

Browse...

Function Name

testFunction

Run in UI Thread

Calling Conventions

C

Parameter

return type

Type

Void

Add a Parameter Before

Add a Parameter After

Delete this Parameter

Function Prototype:

```
void ?testFunction@@YAXJJPAJ0@Z(long arg1, long arg2, long *arg3, long *arg4);
```

OK

Cancel

Help

(Globals)

LL application.

Call Library Function

Library Name or Path: C:\Curs-11\AnotherSample\Debug\AnotherSample.dll Browse...

Function Name: testFunction Run in UI Thread

Calling Conventions: C

Parameter: return type return type

Type: Void arg1 arg2 arg3 arg4

Parameter Before Parameter After Delete this Parameter

Function Prototype:
void ?testFunction@@YAXJJPAJ0@Z(long arg1, long arg2, long *arg3, long *arg4);

OK Cancel Help

a
0
b
0

AnotherSample.dll - 0 error(s), 0 warning(s)

Call Library Function [X]

Library Name or Path:

Function Name:

Calling Conventions:

Parameter:

Type:

Data Type:

Pass:

Function Prototype:

```
void ?testFunction@@YAXJJPAJ0@Z(long arg1, long arg2, long *arg3, long *arg4);
```

