C#.NET 2010 example: DG1Arb

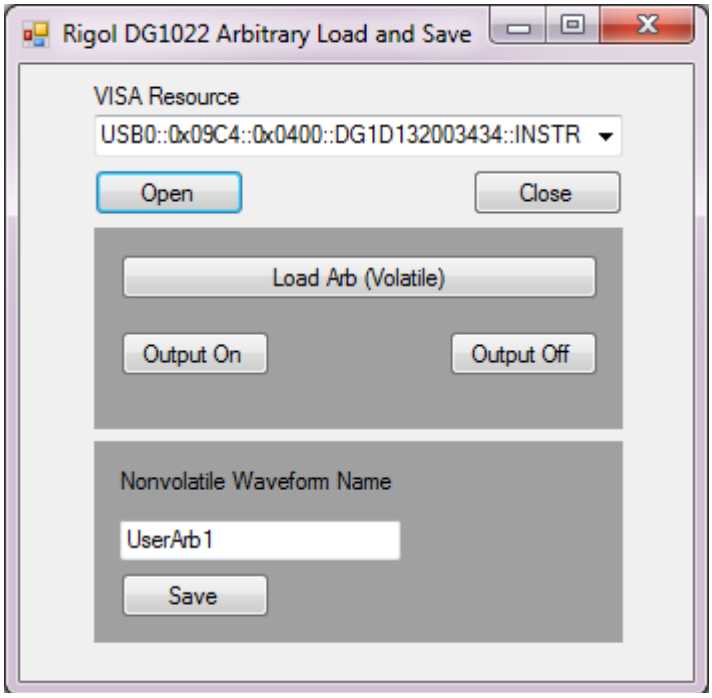
*Date:*09/12/2011

Solution: We have created a small example to show how to remotely program the Rigol DG1022.

This example allows you to establish a message-based VISA session with the instrument, load an aribitrary waveform into volatile memory, enable/disable the outputs, and save the waveform to nonvolatile memory.

It utilizes .NET framwork 4.0 and National Instruments VISA .

Here is the design of the form:



Added Controls
Qty 1 Combo box:

cboVISARsrc - Holds instrument VISA resource ID.

Qty 6 Buttons:

btnOpen - Opens VISA session
btnClose - Close VISA session
btnLoadArb VISA session
btnOutpOn – Enables instrument output
btnOutpOff - Disables instrument output
btnSaveArb – Saves named arbitrary waveform to nonvolatile memory

Qty 1 Text box

txtSaveFileName – Nonvolatile waveform name to save

Code:

```
using System;
using System.Diagnostics;
using System.Threading;
using System.Collections;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Windows.Forms;
using NationalInstruments.VisaNS;

// DG1Arb is a small application created to give
// users a VB.Net example of loading and saving an arbitrary waveform
// into the Rigol  DG1022 Arbitrary Waveform Generator.
//
// The commands are also highlighted in the Arb Load example of the DG1022 Programming
manual.
//
// This app utilizes .NET Framework 4.0 and the associated National Instruments VISA
Common and VISANS
// references.
//
// 1) Enter VISA resource ID.
//
// NOTE: A USB instrument resource ID will look like this:
// USB0::XXXX::YYYY::SSSSSS::INSTR
// where XXXX is the Vendor ID (VID), YYYYY is the Product ID (PID), and SSSS is the
product identifier. Typically, Serial Number.
//
// See your VISA documentation for more information on valid VISA resource ID formats.
//
// 2) Press Open button to created new message based VISA session
// 3) Press Load Arb to load the predefined arbitrary points into the instruments
volatile memory.
// 4) Press Output On/Off to control the instrument output state.
// 5) Enter the desired filename and press Save if you would like to save the waveform to
nonvolatile memory.
// 6) Press Close to end the VISA session prior to closing the active window.
//
// This application is provided as-is. No support or warranty is expressed or implied.
//
// Author: JC
// Company: Rigol Technologies, North America
// Date: 09.09.2011
//
```

```
namespace WindowsFormsApplication1
{
    public partial class Form1 : Form
    {
        private MessageBasedSession mbSession; //Create Message based session

        public Form1()
        {
            InitializeComponent();
        }


        private void btnOpen_Click(Object sender, EventArgs e)
        {
            string strVISARsrc = cboVISARsrc.Text; //Set VISA resource ID

            try
            {
   mbSession = (MessageBasedSession)ResourceManager.GetLocalManager().Open(strVISARsrc);
                //Instantiate new message based VISA session

            }
            catch (InvalidCastException)
            {
                MessageBox.Show("Resource selected must be a message-based session");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }



        private void btnClose_Click(System.Object sender, System.EventArgs e)
        {
            try
            {
                mbSession.Dispose(); //Close VISA session
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
```

```
private void btnLoadArb_Click(System.Object sender, System.EventArgs e)
        {
            try
            {
                mbSession.Write("FUNC USER"); //Select User Defined Waveform
                wait(10);
                mbSession.Write("FREQ 1000"); //Set repetition frequency
                wait(10);
                mbSession.Write("VOLT:UNIT VPP"); //Set voltage amplitude units
                wait(10);
                mbSession.Write("VOLT:HIGH 4"); //Set amplitude high
                wait(20);
                mbSession.Write("VOLT:LOW -4"); //Set amplitude low
                wait(20);
                mbSession.Write("DATA:DAC VOLATILE,8192,16383,8192,0");
                //Set amplitude of waveform points using bit value/point.
                //NOTE: This example is a 4 point arbitrary waveform.
                wait(10);
                mbSession.Write("FUNC:USER VOLATILE"); //Enable the volatile waveform.
                wait(10);

            }
            catch (InvalidCastException)
            {
                MessageBox.Show("Resource selected must be a message-based session");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }



        private void wait(int interval)
        //Loops for a specificied period of time (milliseconds)
        //
        {
            Stopwatch sw = new Stopwatch();
            sw.Start();

            Thread.Sleep(interval); //Use Thread Sleep to wait

            sw.Stop();
        }
```

```
        private void btnOutpOn_Click(System.Object sender, System.EventArgs e)
        {
            try
            {
                mbSession.Write("OUTP ON"); //Enable output
                wait(10);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }



         private void btnOutpOff_Click(System.Object sender, System.EventArgs e)
        {
            try
            {
                mbSession.Write("OUTP OFF");//Disable output
                wait(10);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        private void btnSaveArb_Click(System.Object sender, System.EventArgs e)
        {
            try
            {
                mbSession.Write("DATA:COPY " + txtSaveFileName.Text);
                //Save waveform to nonvolatile memory
                wait(10);
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

    }
}
```

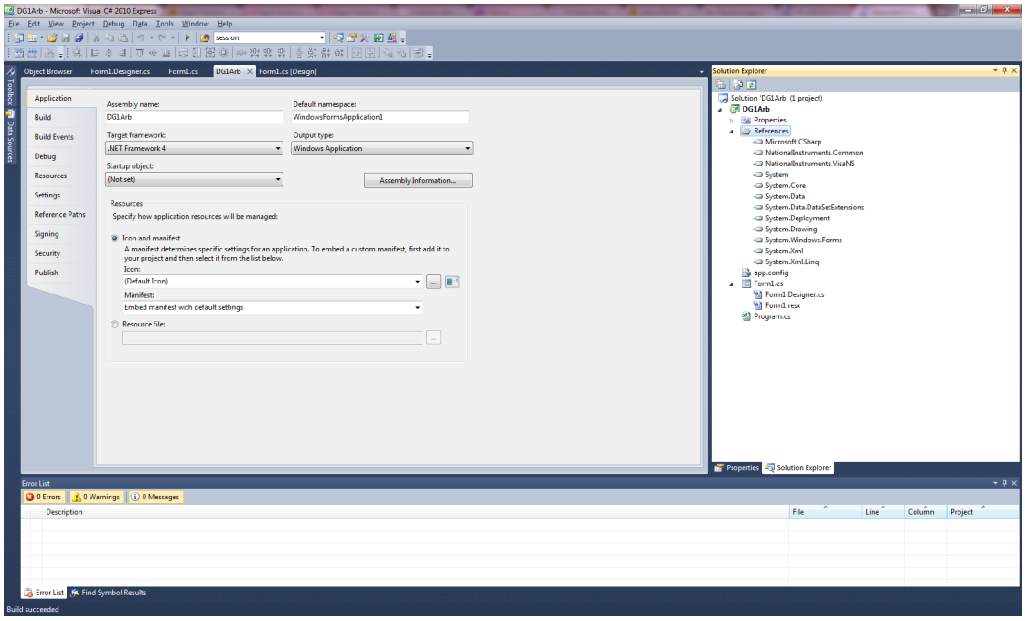


Figure 1: References and build properties.