



Using USB-6008 in Visual Studio/C#

Hans-Petter Halvorsen, M.Sc.

Contents

- Visual Studio/C#
- What is DAQ?
- Using USB-6008 in C#
- Analog In
- Analog Out
- Using Timer, Charts, etc.



USB-6008
I/O Module

Software

- Visual Studio/C#
- DAQmx Driver
- DAQmx Driver can be downloaded for free from Internet





DAQ

Data Acquisition

Hans-Petter Halvorsen, M.Sc.

DAQ – Data Acquisition

Your App created
with C#

Sensors, etc.

PC-BASED DATA ACQUISITION

1 INPUT/OUTPUT SIGNALS

ANALOG



DIGITAL



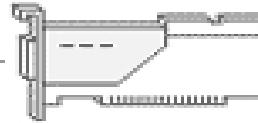
COUNTER/
TIMER



SENSORS



HARDWARE



2 DATA ACQUISITION
HARDWARE

4 SOFTWARE



3 APPLICATION AND
DRIVER SOFTWARE

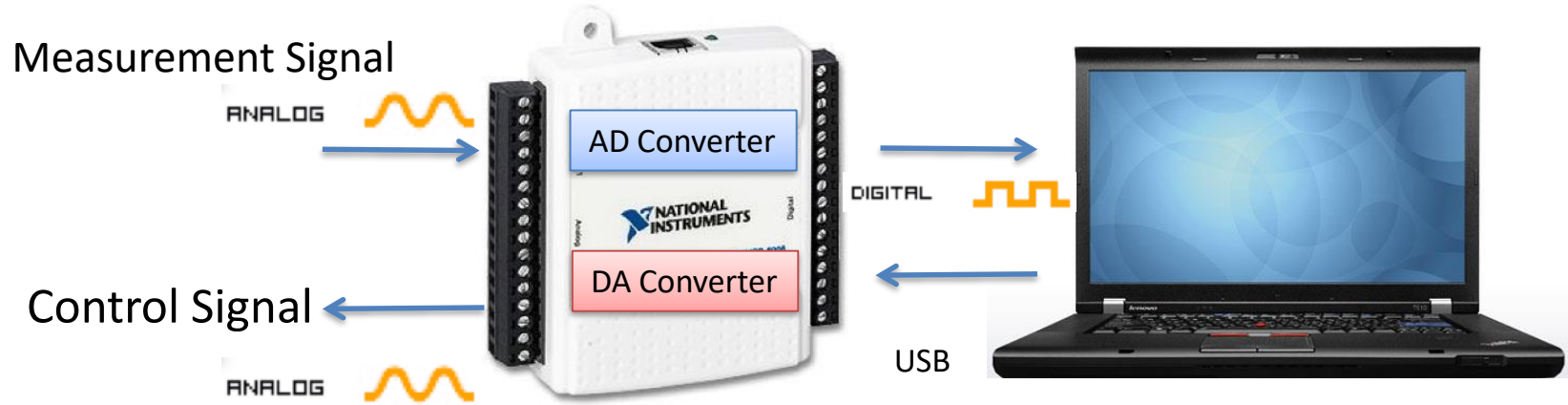
A DAQ System consists of 4 parts:

1. Physical input/output signals, sensors
2. DAQ device/hardware
3. Driver software
4. Your software application (Application software)

NI USB 6008 DAQ Device

NI DAQmx Driver

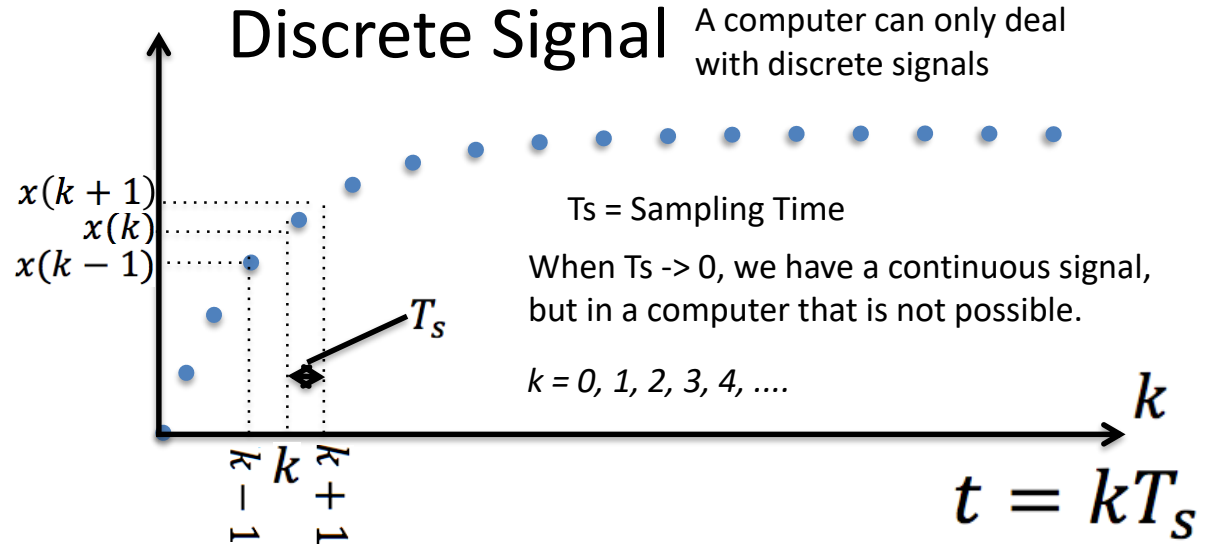
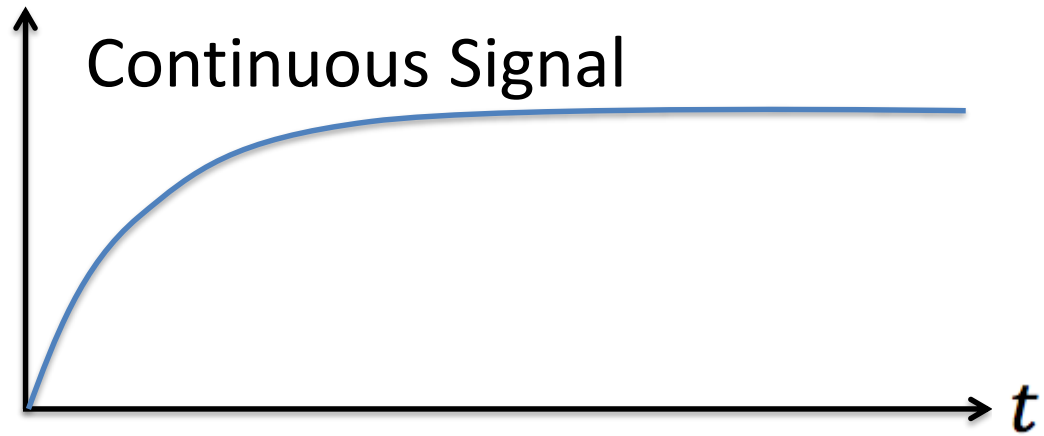
AD & DA Converters



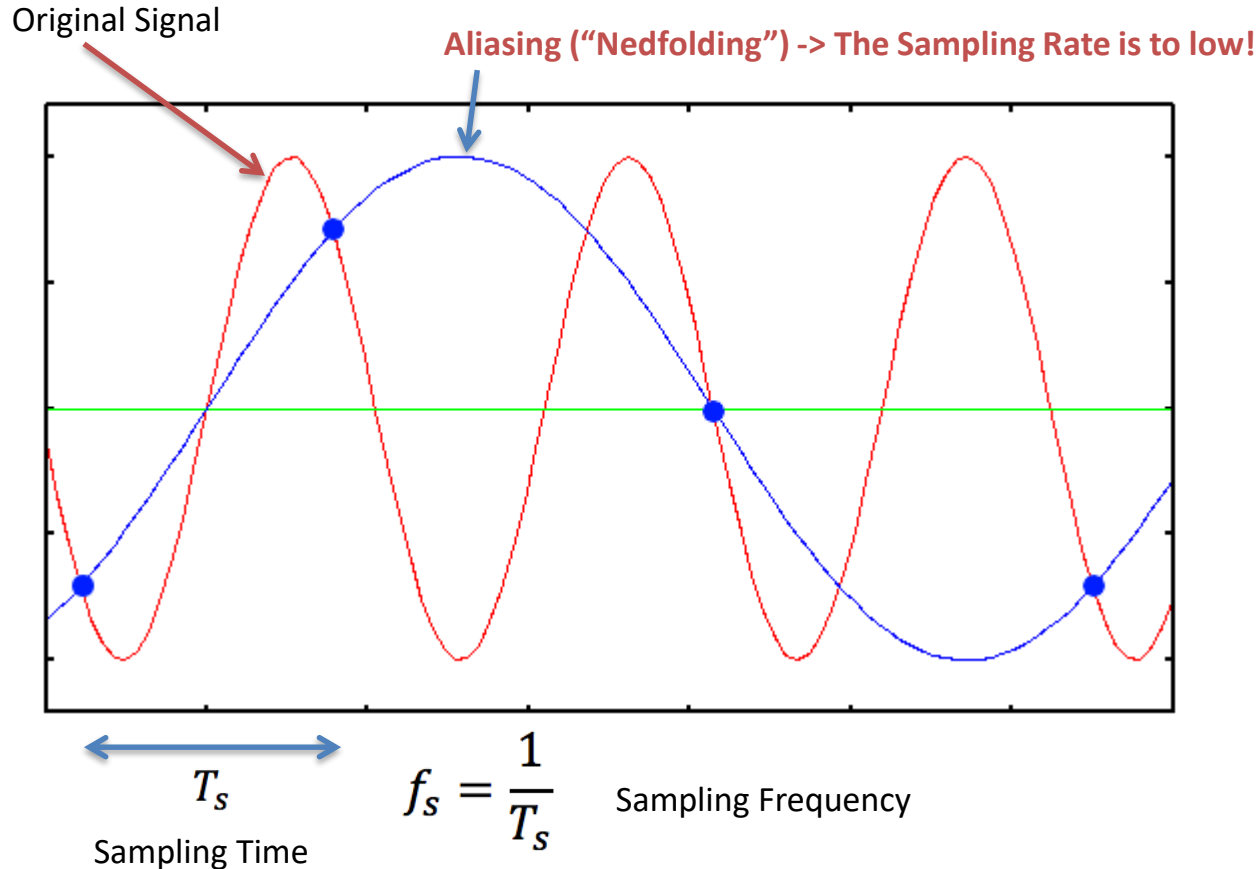
AD – Analog to Digital
DA – Digital to Analog

All Analog Signals needs to be converted to Digital Signals before the Computer can use them (AD Converter).

Continuous vs. Discrete Signals



Sampling and Aliasing





USB-6008

Hans-Petter Halvorsen, M.Sc.

How-To use USB-6008 with Visual Studio

USB-6008



USB



PC with Visual Studio

NI USB-6008 I/O Module

USB Connection



Specifications:

- 8 analog inputs, AI (12-bit, 10 kS/s, -10-10V)
- 2 analog outputs, AO (12-bit, 150 S/s, 0-5V)
- 12 digital I/O (DI/DO) 0-5V
- 32-bit counter

4 different types of Signals:

AO – Analog Output

AI – Analog Input

DO – Digital Output

DI – Digital Input

Note! **DAQmx** Driver is needed!!





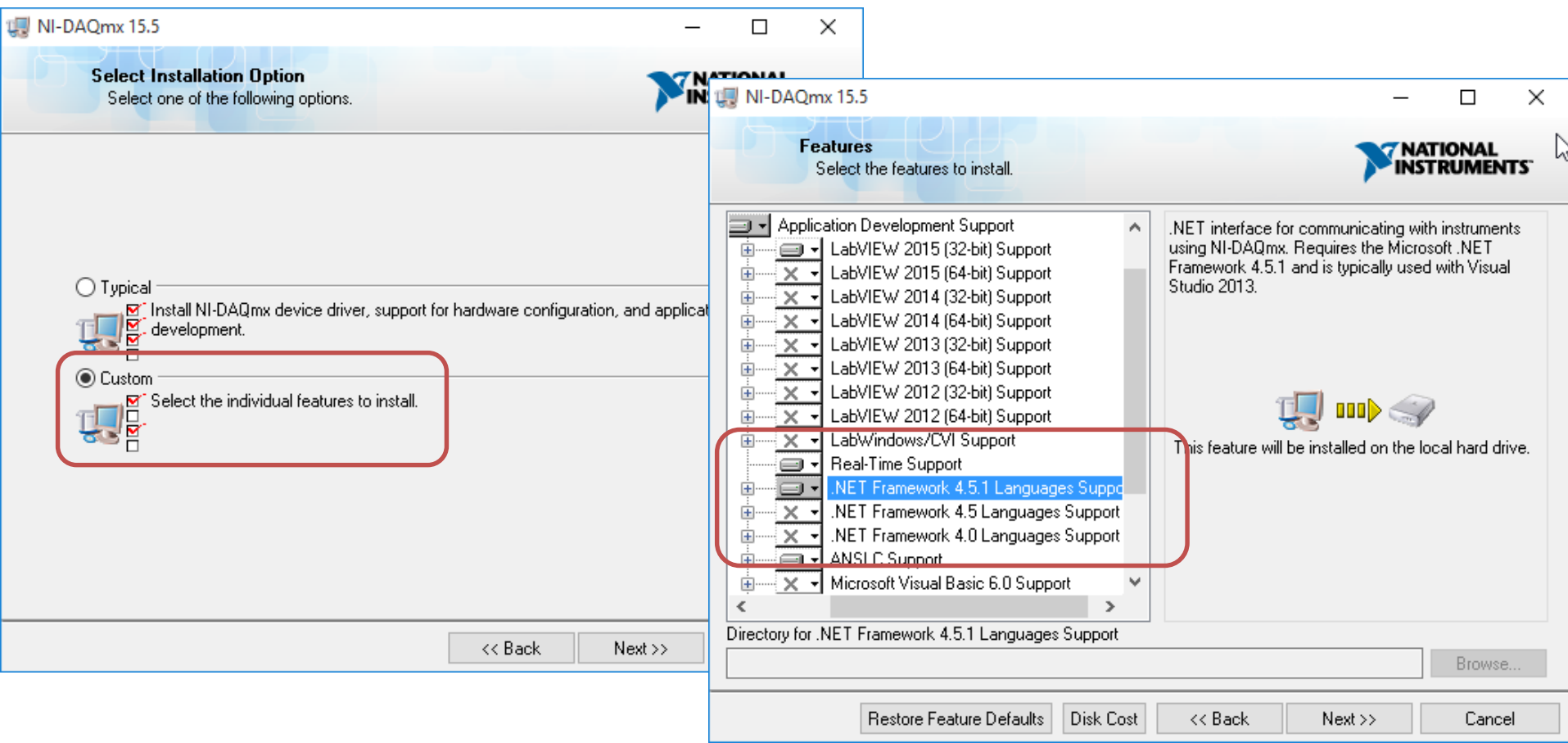
DAQmx Driver

Hans-Petter Halvorsen, M.Sc.

NI DAQmx Driver

- National Instruments provides a native .NET API for NI-DAQmx. This is available as a part of the NI-DAQmx driver
- In order to install the DAQmx API for C#, make sure to select “Custom” and then “.NET Support” when installing the DAQmx driver.
- Next, make sure that you select .NET Framework X.x Support for the version of .NET that your version of Visual Studio is using.

NI DAQmx Driver Installation



MAX – Measurement & Automation Explorer

NI USB-6008 "Dev1"

You may change the name ("Dev1")

C# Examples



Analog In

Hans-Petter Halvorsen, M.Sc.

Read Analog Values



We will read the voltage values on different batteries

Read Analog Signals with USB-6008

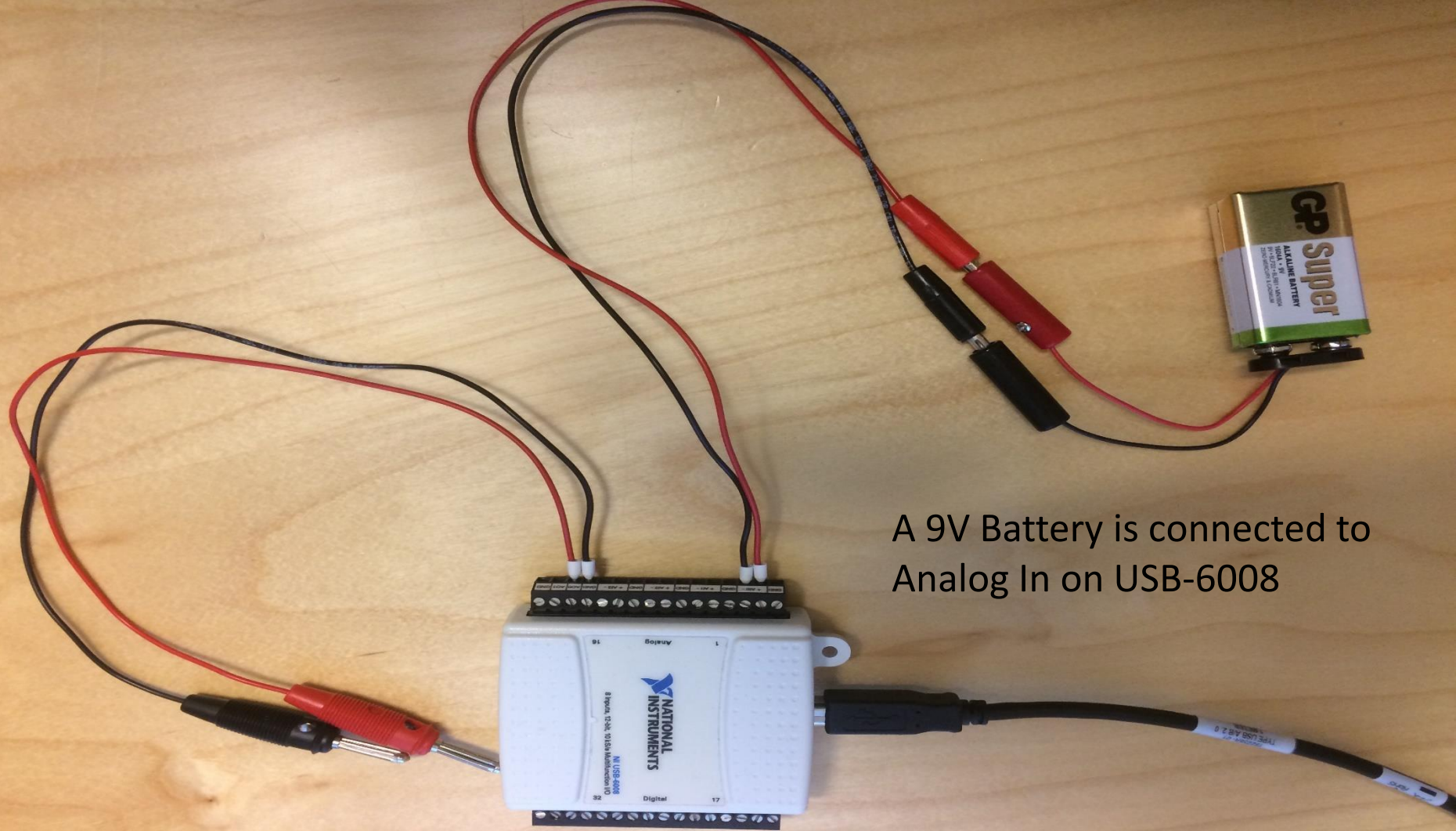
Multimeter



USB

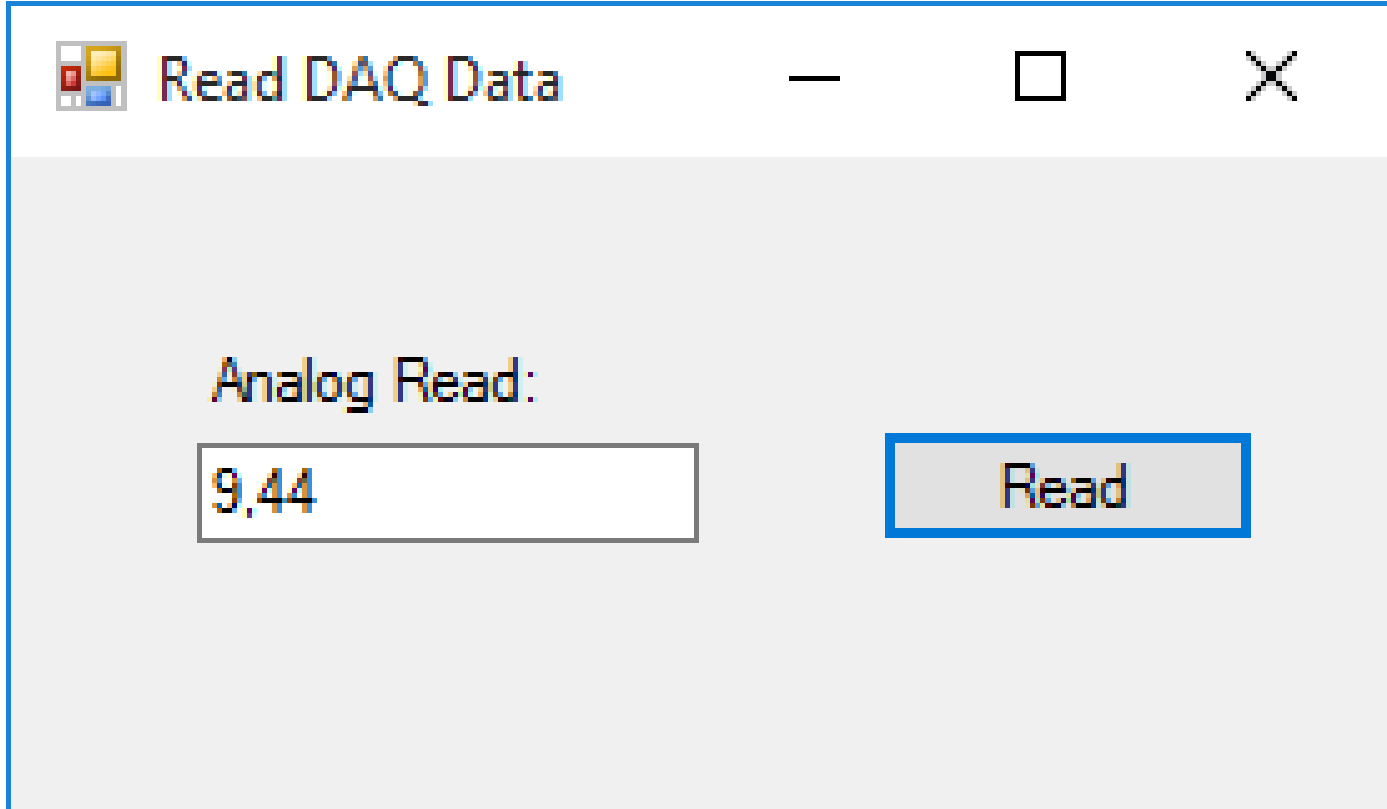


9V



A 9V Battery is connected to
Analog In on USB-6008

Read from USB-6008 DAQ Device



A screenshot of a Windows-style application window titled "Read DAQ Data". The window has a standard title bar with a minimize button, a maximize button (disabled), and a close button. The main content area is light gray and contains the text "Analog Read:" followed by a text box displaying the value "9.44". To the right of the text box is a button labeled "Read".

Read DAQ Data

Analog Read:

9.44

Read

Simple DAQ in C# with DAQmx

```
using NationalInstruments.DAQmx;

...
Task analogInTask = new Task();
AIChannel myAIChannel;

myAIChannel = analogInTask.AIChannels.CreateVoltageChannel(
    "dev1/ai0",
    "myAIChannel",
    AITerminalConfiguration.Differential,
    0,
    5,
    AIVoltageUnits.Volts
);

AnalogSingleChannelReader reader = new
    AnalogSingleChannelReader(analogInTask.Stream);

double analogDataIn = reader.ReadSingleSample();

txtAnalogIn.Text = analogDataIn.ToString("0.00");
```



Analog In Example

```
using NationalInstruments.DAQmx;
```

```
...
```

```
Task analogInTask = new Task();
```

```
AIChannel myAIChannel;
```

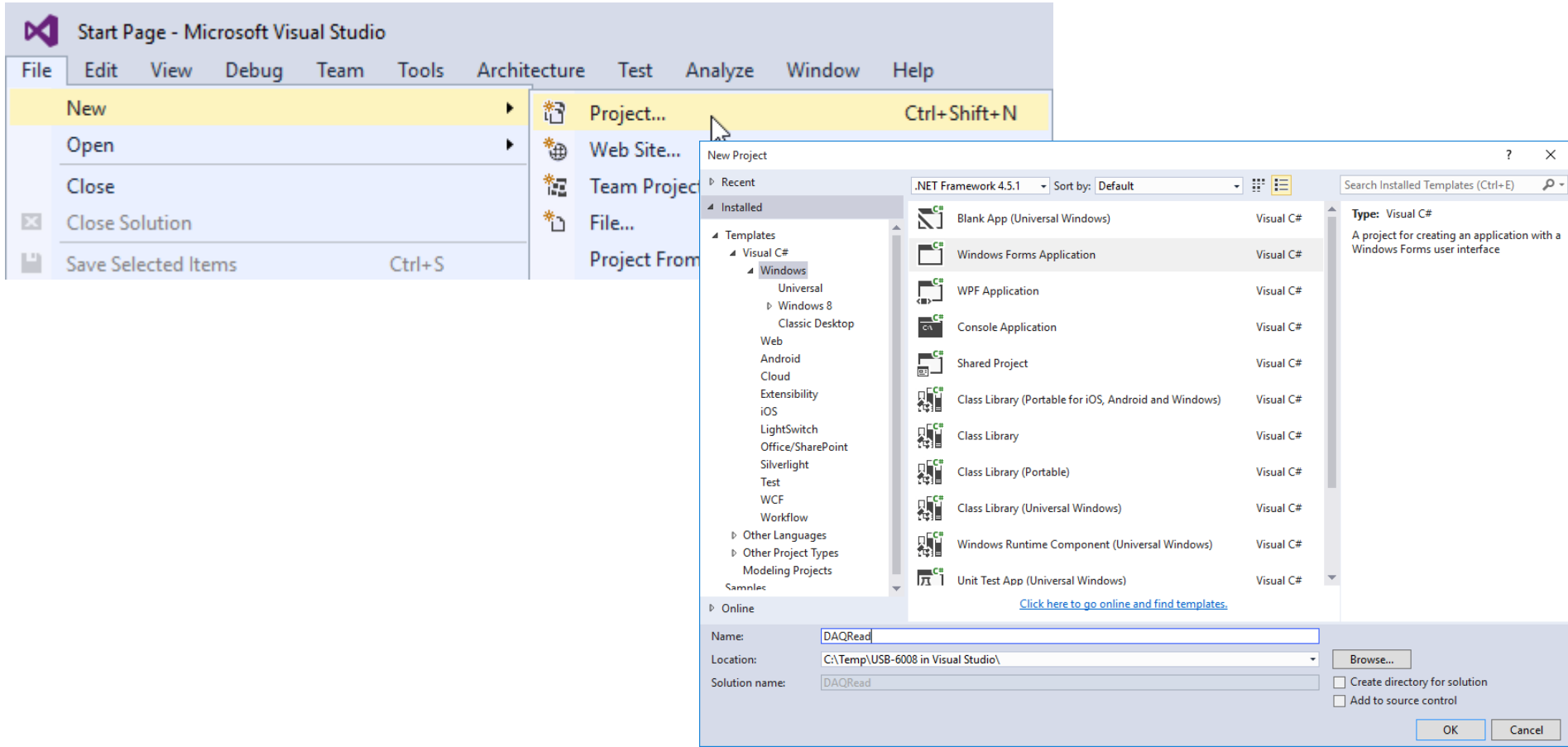
```
myAIChannel = analogInTask.AIChannels.CreateVoltageChannel(  
    "dev1/ai0",  
    "myAIChannel",  
    AITerminalConfiguration.Differential,  
    0,  
    5,  
    AIVoltageUnits.Volts  
);
```

```
AnalogSingleChannelReader reader = new  
    AnalogSingleChannelReader(analogInTask.Stream);
```

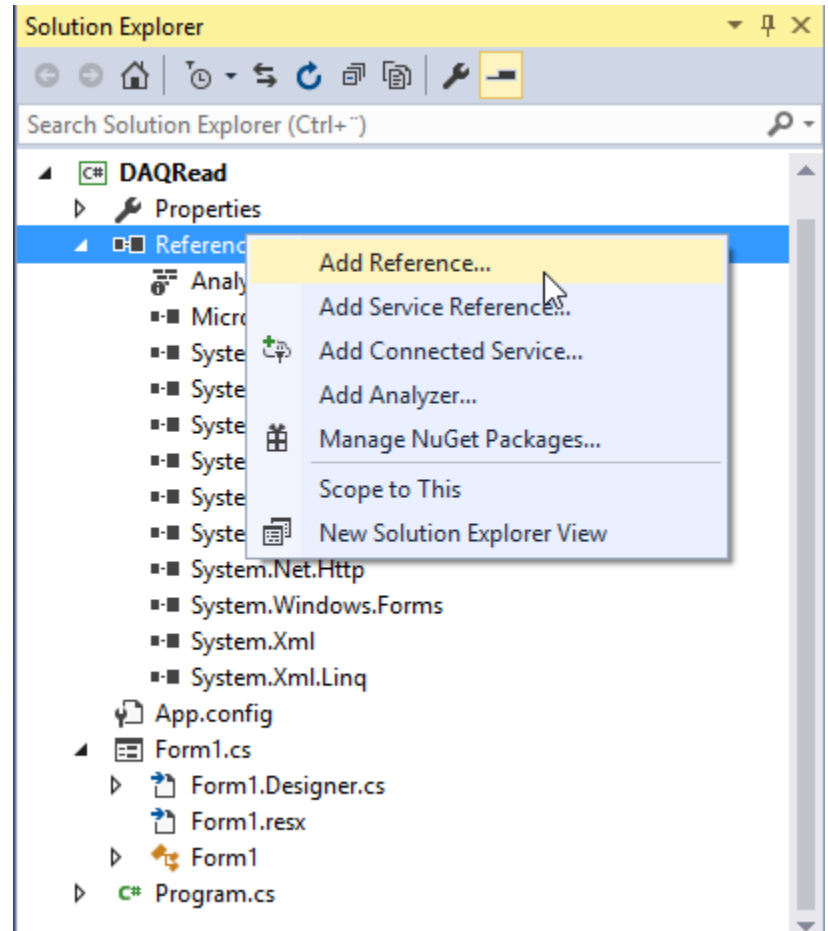
```
double analogDataIn = reader.ReadSingleSample();
```

```
txtAnalogIn.Text = analogDataIn.ToString("0.00");
```

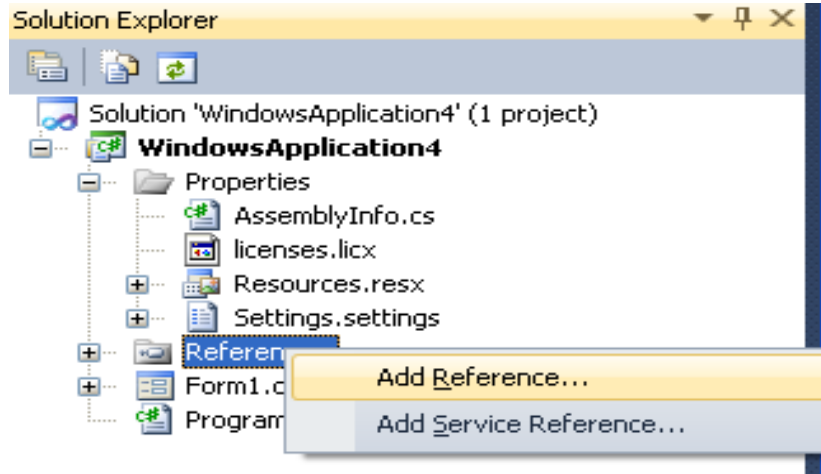
WinForm App



Add Assembly References



Add References to the DAQmx Driver in Visual Studio



Select «Browse» and Find
`NationalInstruments.DAQmx.dll`

`C:\Program Files (x86)\National Instruments\...`

We also need to add the following Namespaces:

```
using NationalInstruments.DAQmx;
```

```
NationalInstruments.DAQmx.dll
```

DAQRead - Microsoft Visual Studio

File Edit View Project Build Debug Team Tools Architecture Test Analyze Window Help

Quick Launch (Ctrl+Q)

Debug Any CPU Start

Form1.cs [Design] Form1.cs App.config

Toolbox

Search Toolbox

- All Windows Forms
 - Pointer
 - BackgroundWorker
 - BindingNavigator
 - BindingSource
 - Button
 - CheckBox
 - CheckedListBox
 - ColorDialog
 - ComboBox
 - ContextMenuStrip
 - DataGridView
 - DataSet
 - DateTimePicker
 - DirectoryEntry
 - DirectorySearcher
 - DomainUpDown
 - ErrorProvider
 - EventLog
 - FileSystemWatcher
 - FlowLayoutPanel
 - FolderBrowserDialog
 - FontDialog

Read DAQ Data

Analog Read:

Read

Solution Explorer

Search Solution Explorer (Ctrl+)

- Solution 'DAQRead' (1 project)
 - DAQRead
 - Properties
 - References
 - App.config
 - Form1.cs
 - Form1.Designer.cs
 - Form1.resx
 - Form1
 - Program.cs

Properties

Form1.cs File Properties

Build Action	Compile
Copy to Output Directory	Do not copy
Custom Tool	
Custom Tool Namespace	
File Name	Form1.cs

Build Action

How the file relates to the build and deployment processes.

```

using System;
using System.Windows.Forms;
using NationalInstruments.DAQmx;

namespace DAQRead
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnRead_Click(object sender, EventArgs e)
        {
            Task analogInTask = new Task();

            AIChannel myAIChannel;

            myAIChannel = analogInTask.AIChannels.CreateVoltageChannel(
                "dev1/ai0",
                "myAIChannel",
                AITerminalConfiguration.Differential,
                0,
                10,
                AIVoltageUnits.Volts
            );

            AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

            double analogDataIn = reader.ReadSingleSample();

            txtDaqValue.Text = analogDataIn.ToString("0.00");
        }
    }
}

```

DEMO

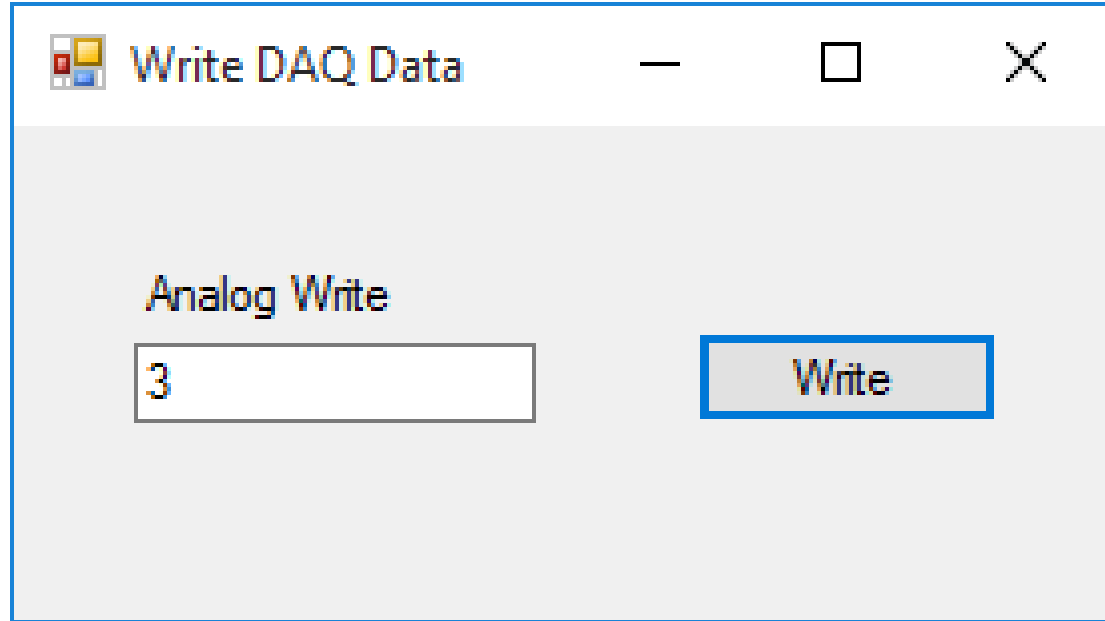
C# Examples



Analog Out

Hans-Petter Halvorsen, M.Sc.

Write to USB-6008 DAQ Device



Write DAQ Data

Analog Write

3

Write

Simple DAQ in C# with DAQmx



```
using NationalInstruments.DAQmx;
```

```
...
```

```
Task analogOutTask = new Task();
```

```
AOChannel myAOChannel;
```

```
myAOChannel = analogOutTask.AOChannels.CreateVoltageChannel(  
    "dev1/ao0",  
    "myAOChannel",  
    0,  
    5,  
    AOVoltageUnits.Volts  
);
```

```
AnalogSingleChannelWriter writer = new  
    AnalogSingleChannelWriter(analogOutTask.Stream);
```

```
double analogDataOut;
```

```
analogDataOut = Convert.ToDouble(txtAnalogOut.Text);
```

```
writer.WriteSingleSample(true, analogDataOut);
```

Analog Out Example

```
using NationalInstruments.DAQmx;

...
Task analogOutTask = new Task();
AOChannel myAOChannel;

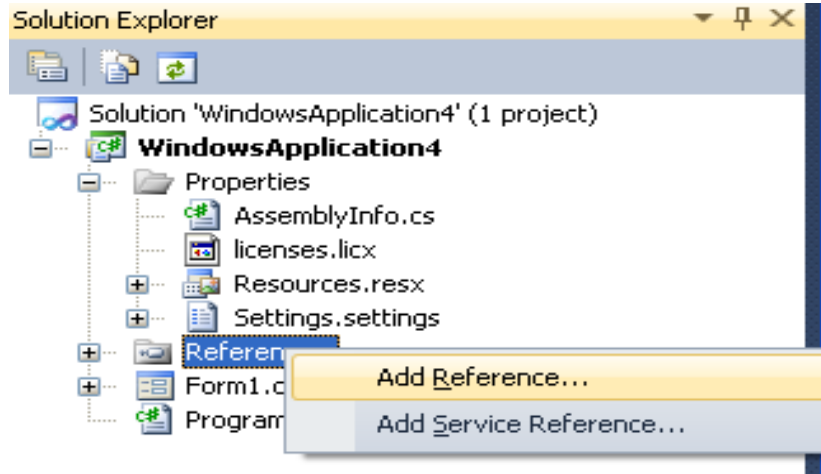
myAOChannel = analogOutTask.AOChannels.CreateVoltageChannel(
    "dev1/ao0",
    "myAOChannel",
    0,
    5,
    AOVoltageUnits.Volts
);

AnalogSingleChannelWriter writer = new
    AnalogSingleChannelWriter(analogOutTask.Stream);

double analogDataOut;
analogDataOut = Convert.ToDouble(txtAnalogOut.Text);

writer.WriteSingleSample(true, analogDataOut);
```


Add References to the DAQmx Driver in Visual Studio



Select «Browse» and Find
`NationalInstruments.DAQmx.dll`

`C:\Program Files (x86)\National Instruments\...`

We also need to add the following Namespaces:

```
using NationalInstruments.DAQmx;
```

```
NationalInstruments.DAQmx.dll
```

DAQ in C# with DAQmx – Analog Out



```
private void btnWriteAnalogOut_Click(object sender, EventArgs e)
{
    Task analogOutTask = new Task();

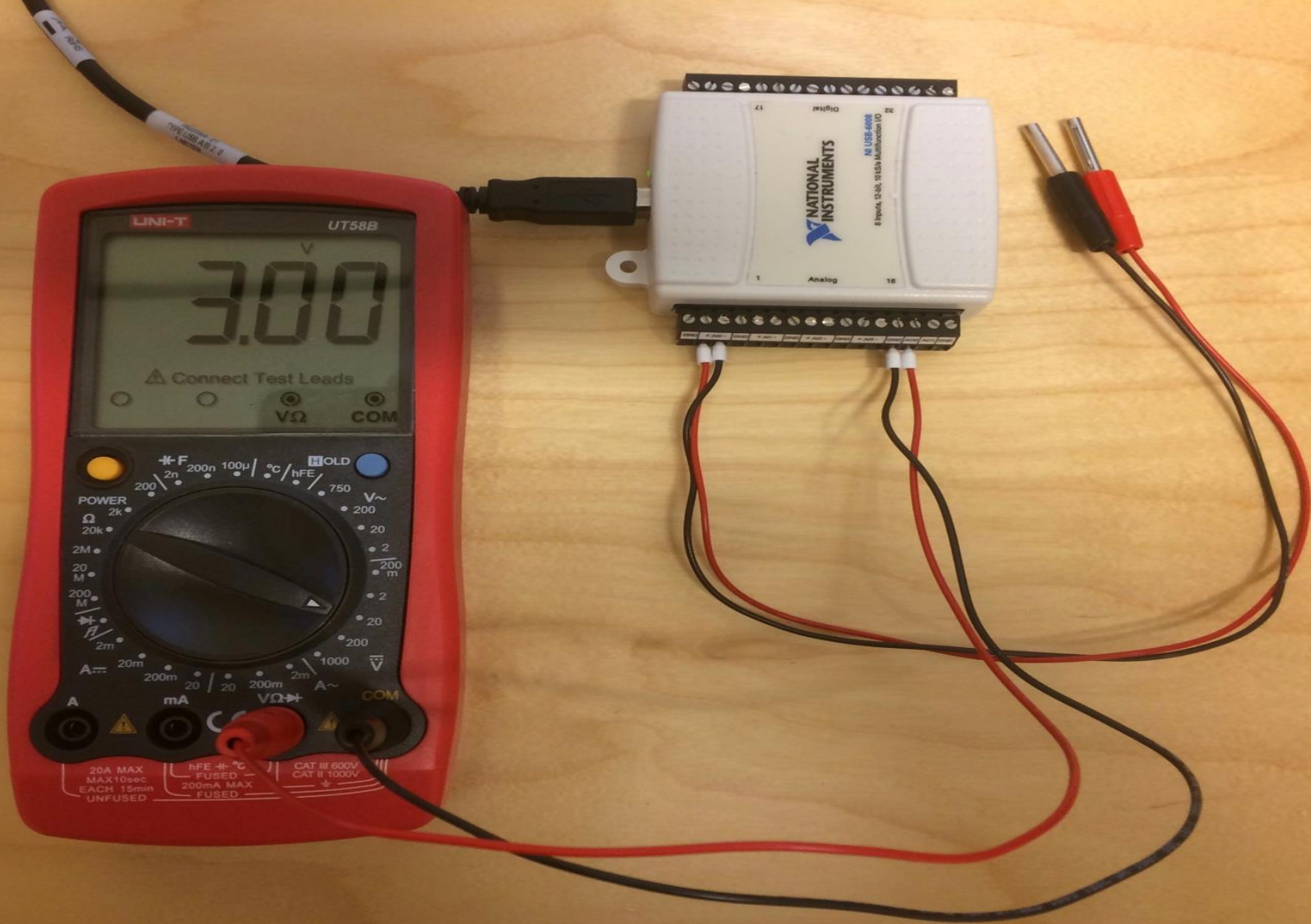
    AOChannel myAOChannel;

    myAOChannel = analogOutTask.AOChannels.CreateVoltageChannel(
        "dev1/ao0",
        "myAOChannel",
        0,
        5,
        AOVoltageUnits.Volts
    );

    AnalogSingleChannelWriter writer = new
        AnalogSingleChannelWriter(analogOutTask.Stream);

    double analogDataOut;
    analogDataOut = Convert.ToDouble(txtAnalogOut.Text);

    writer.WriteSingleSample(true, analogDataOut);
}
```



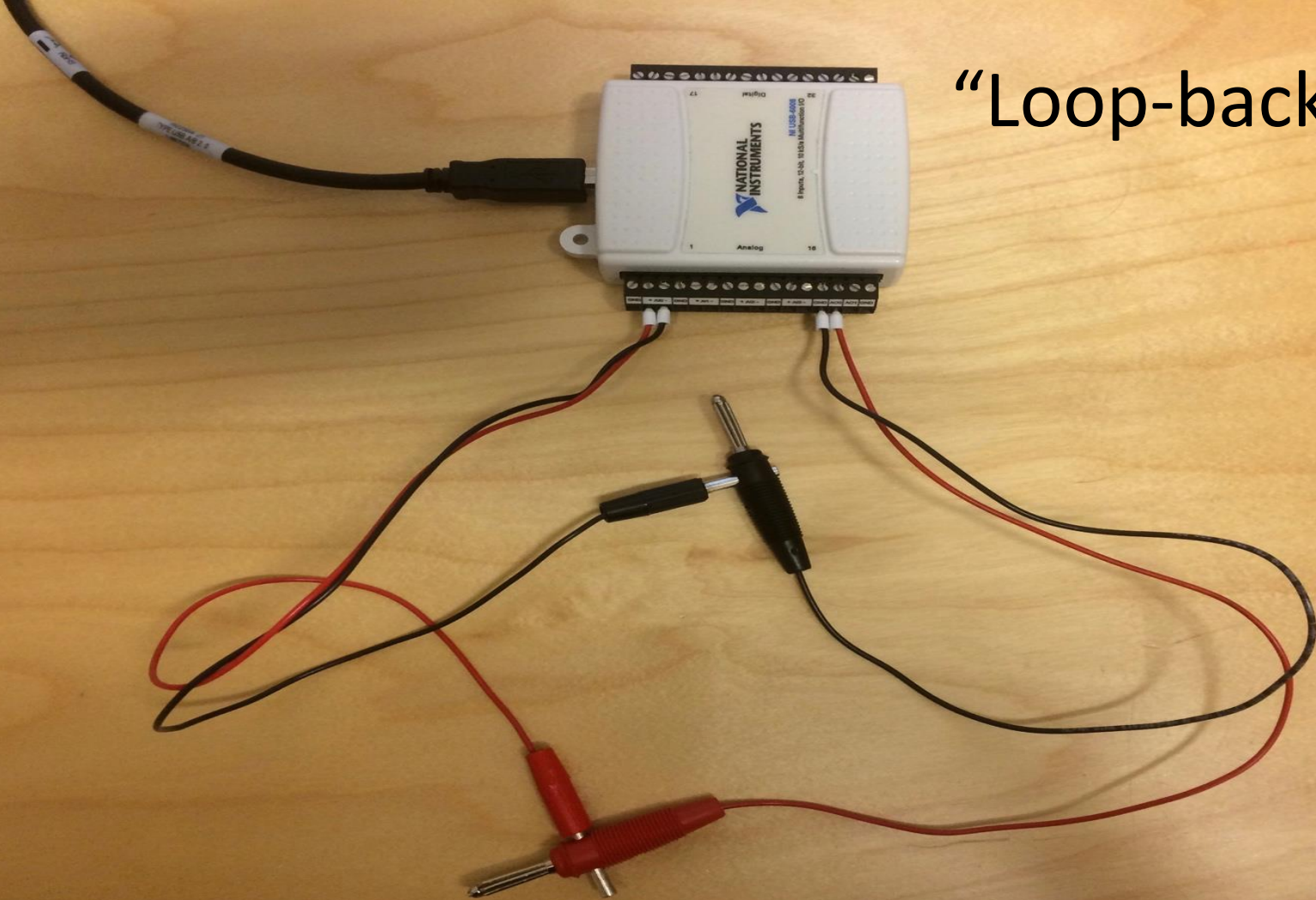
DEMO



Analog Out + Analog In

Hans-Petter Halvorsen, M.Sc.

“Loop-back” Test



Write/Read Data using USB-6008

Write DAQ Data

Analog Write

3

Write

Read DAQ Data

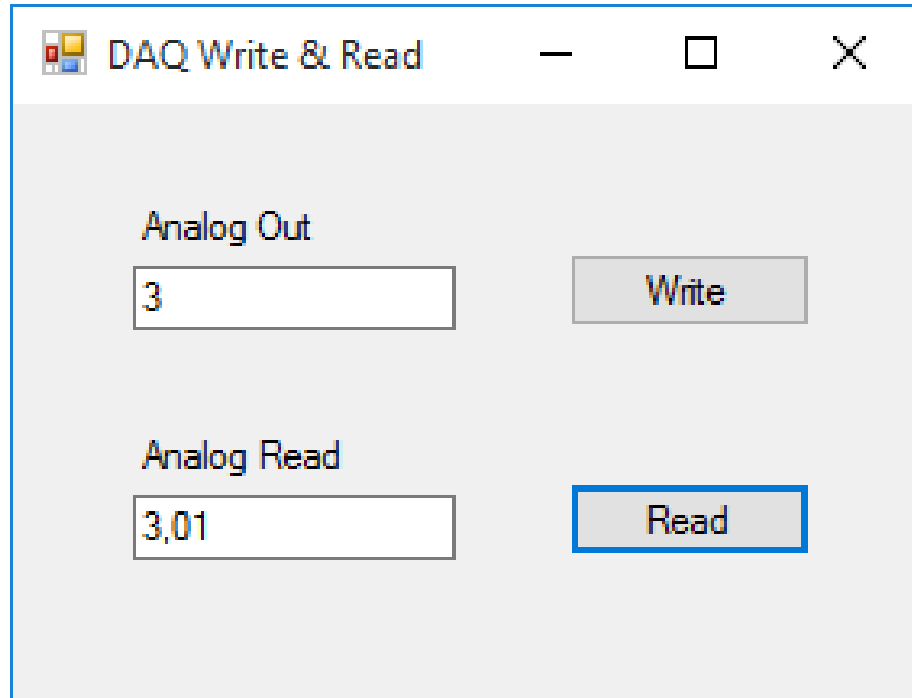
Analog Read:

3,01

Read

DEMO

Write/Read Data using USB-6008



The image shows a software window titled "DAQ Write & Read". It contains two main sections: "Analog Out" and "Analog Read". In the "Analog Out" section, there is a text input field containing the number "3" and a "Write" button. In the "Analog Read" section, there is a text input field containing the value "3,01" and a "Read" button. The "Read" button is highlighted with a blue border.

Section	Control	Value
Analog Out	Input Field	3
	Write Button	Write
Analog Read	Input Field	3,01
	Read Button	Read

DEMO



Improvements

Hans-Petter Halvorsen, M.Sc.


Improvements

- Using a Timer
- Trend/Plot the Data from the DAQ device in a Chart
- Create and Use separate Classes for implementing the DAQ code
- ...

Timer

In Visual Studio you may want to use a Timer instead of a While Loop in order to read values at specific intervals.



- 1  **Timer** Select the “Timer” component in the Toolbox
- 2 **Initialization:**

```
public Form1()
{
    InitializeComponent();

    timer1.Start();
}
```

Double-click on the Timer object in order to create the Event

- 4 **Timer Event:**

```
private void timer1_Tick(object sender, EventArgs e)
{
    ... //Read from DAQ Device
    ... //Formatting
    ... //Plot Data
}
```

Properties:

3

Properties	
timer1 System.Windows.Forms.Timer	
[ApplicationSettings]	
(Name)	timer1
Enabled	False
GenerateMember	True
Interval	100
Modifiers	Private
Tag	

You may specify the Timer Interval in the Properties Window

Structure your Code properly!!
Define Classes and Methods which you can use here

Read DAQ Values using a Timer

The screenshot displays the Microsoft Visual Studio IDE with the following components:

- Toolbox:** On the left, the 'General' tab is active, showing a message: "There are no usable controls in this group. Drag an item onto this text to add it to the toolbox."
- Code Editor:** The central pane shows the code for `Form1.cs` in Design mode. The code is as follows:

```
using System;
using System.Windows.Forms;
using NationalInstruments.DAQmx;

namespace DAQReadwithTimer
{
    3 references
    public partial class Form1 : Form
    {
        1 reference
        public Form1()
        {
            InitializeComponent();

            timer1.Interval = 1000;//ms
            timer1.Start();
        }

        1 reference
        private void timer1_Tick(object sender, EventArgs e)
        {
            Task analogInTask = new Task();

            AICannel myAICannel;

            myAICannel = analogInTask.AIChannels.CreateVoltageChannel(
                "dev1/ai0",
                "myAICannel",
                AITerminalConfiguration.Differential,
                0,
                10,
                AIVoltageUnits.Volts
            );

            AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

            double analogDataIn = reader.ReadSingleSample();

            txtDaqRead.Text = analogDataIn.ToString("0.00");
        }
    }
}
```
- Solution Explorer:** On the right, it shows the project structure for 'DAQReadwithTimer' (1 project), including `Properties`, `References`, `App.config`, `Form1.cs`, and `Program.cs`.
- Properties Window:** Below the Solution Explorer, the 'Properties' window is visible but empty.
- Test Explorer:** At the bottom left, the 'Test Explorer' tab is active.
- Status Bar:** The bottom status bar shows 'Ln 17', 'Col 61', 'Ch 61', 'INS', and a 'Publish' button.

```

using System;
using System.Windows.Forms;
using NationalInstruments.DAQmx;

namespace DAQReadwithTimer
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            timer1.Interval = 1000;//ms
            timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            Task analogInTask = new Task();
            AIChannel myAIChannel;

            myAIChannel = analogInTask.AIChannels.CreateVoltageChannel(
                "dev1/ai0",
                "myAIChannel",
                AITerminalConfiguration.Differential,
                0,
                10,
                AIVoltageUnits.Volts
            );

            AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

            double analogDataIn = reader.ReadSingleSample();

            txtDaqRead.Text = analogDataIn.ToString("0.00");
        }
    }
}

```

DEMO

Trending Data in Visual Studio



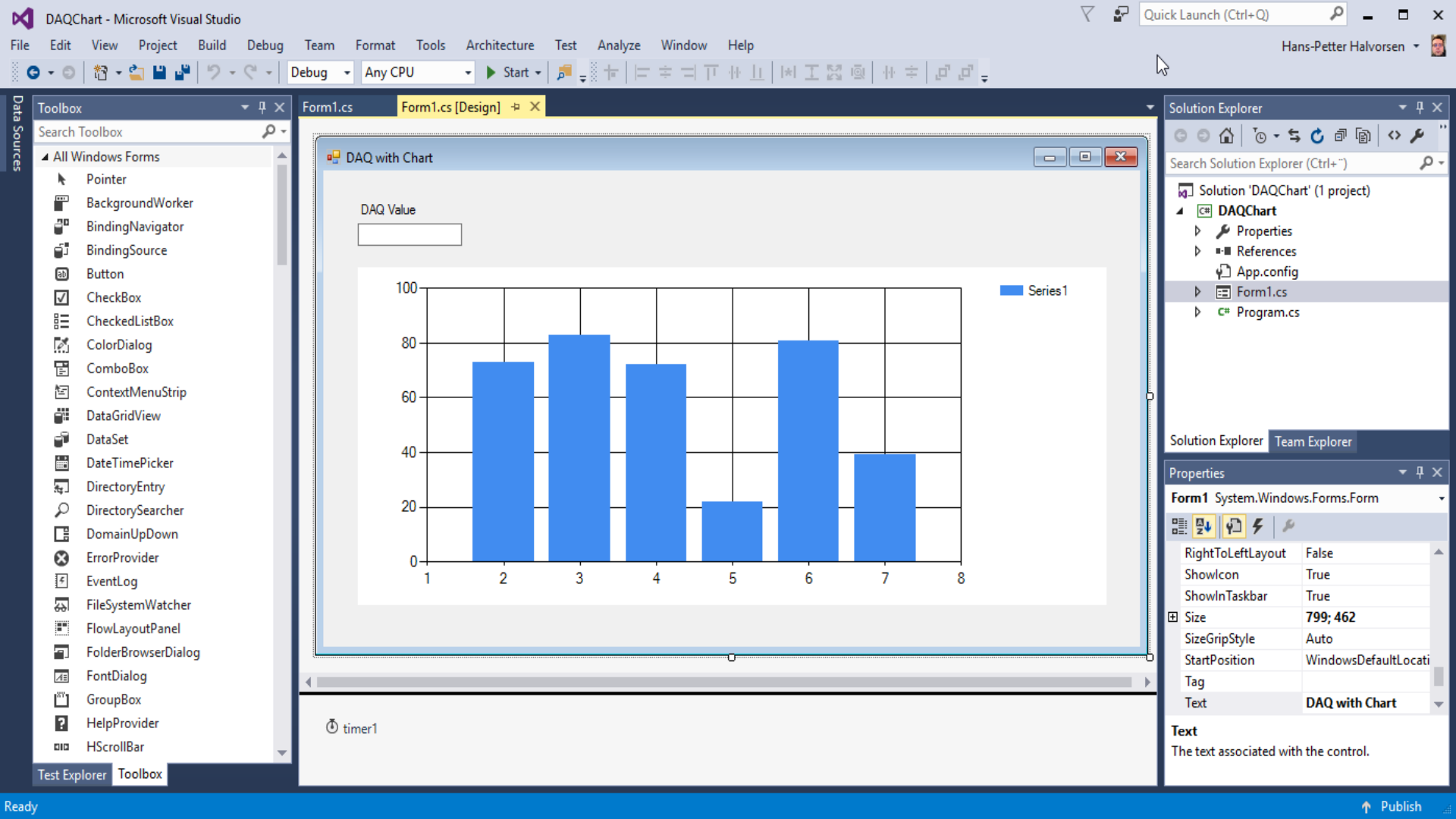
Visual Studio has a Chart control that you can use in Windows Forms or Web application (ASP.NET)

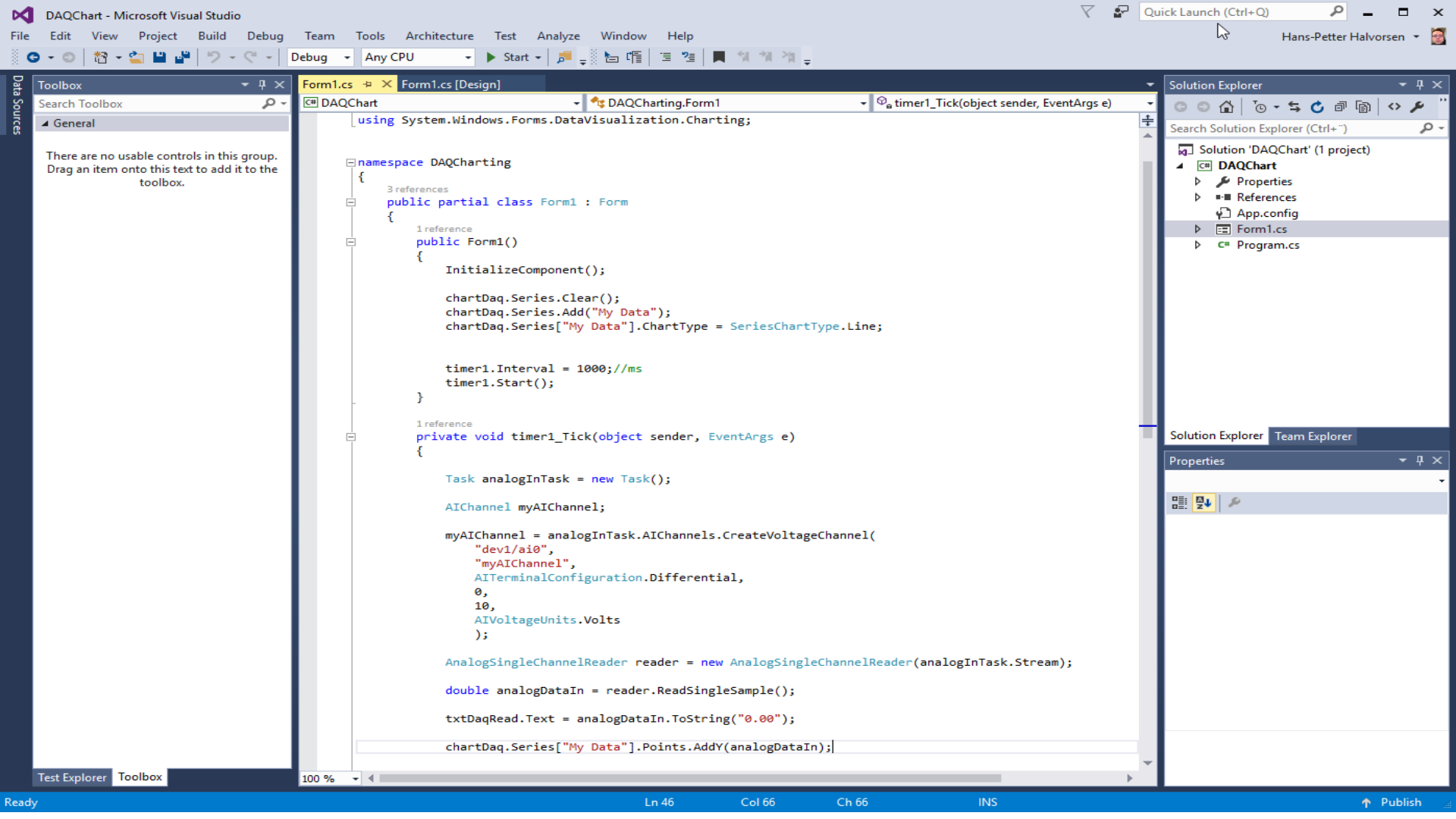
<https://msdn.microsoft.com/en-us/library/dd489237.aspx>

<http://www.i-programmer.info/programming/uiux/2756-getting-started-with-net-charts.html>

```
using System.Windows.Forms.DataVisualization.Charting;
...
chart1.Series.Clear();
chart1.Series.Add("My Data");
chart1.Series["My Data"].ChartType = SeriesChartType.Line;
...
int[] x = {1, 2, 3, 4, 5, 6, 7, 8};
int[] y = {20, 22, 25, 24, 28, 27, 24, 26};
for (int i = 0; i < x.Length; i++)
{
    chart1.Series["My Data"].Points.AddXY(x[i],y[i]);
}
```

Creating a Web App? Use the following Namespace instead:
System.Web.UI.DataVisualization.Charting





DEMO

Hans-Petter Halvorsen, M.Sc.



University College of Southeast Norway

www.usn.no

E-mail: hans.p.halvorsen@hit.no

Blog: <http://home.hit.no/~hansha/>

