https://www.halvorsen.blog

Visual Studio/C# and DAQ

Exemplified using DAQ hardware and DAQmx from NI

Hans-Petter Halvorsen

Contents

- Introduction
- Data Acquisition (DAQ)
- NI DAQ Devices
- Visual Studio/C# Code Examples
  - Analog In
    - Battery Indicator Example
    - TMP36 Temperature Sensor Example
  - Analog Out
  - Digital I/O
    - LED Example (Digital Out)
    - Push Button Example (Digital In)

Introduction

Introduction

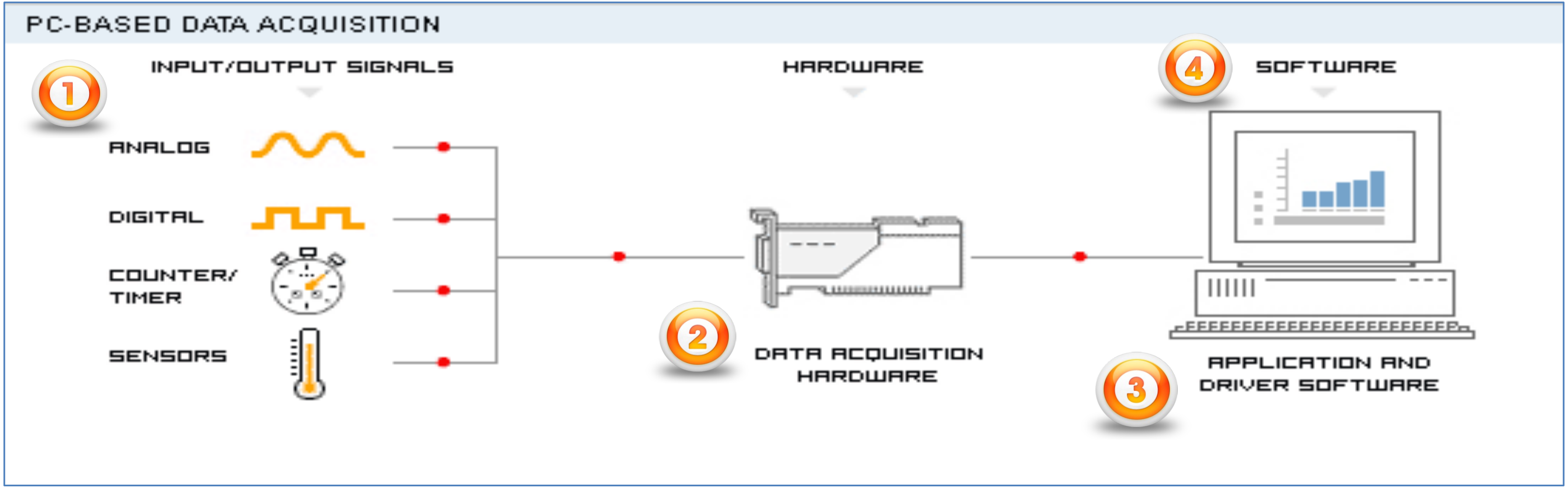
- The purpose is to read and write data using a **DAQ** device where we use Visual Studio and C#.
- We will exemplify by using a DAQ device from NI (previously National Instruments).
- We will use a DAQ device called USB-6008 (which is part of the **USB-600x** low-cost series).
- DAQ devices from NI use the **NI-DAQmx driver**.
- Examples shown will work on all DAQ devices from NI that are using the DAQmx driver (which is many!).
- The principles used can can also be applied on other DAQ hardware from other vendors.

Data Acquisition (DAQ)

Hans-Petter Halvorsen

Table of Contents

Data Acquisition (DAQ)


PC-BASED DATA ACQUISITION
1
INPUT/OUTPUT SIGNALS
HARDWARE
4
SOFTWARE
ANALOG
DIGITAL
COUNTER/ TIMER
SENSORS
2
DATA ACQUISITION HARDWARE
3
APPLICATION AND DRIVER SOFTWARE

A DAQ System consists of 4 parts:

1. Physical input/output signals, sensors
2. DAQ device/hardware (we will use NI USB-6008)
3. Driver software (NI DAQmx in our case)
4. Your software application (Application software) – We will use Visual Studio/C#

DAQ Device

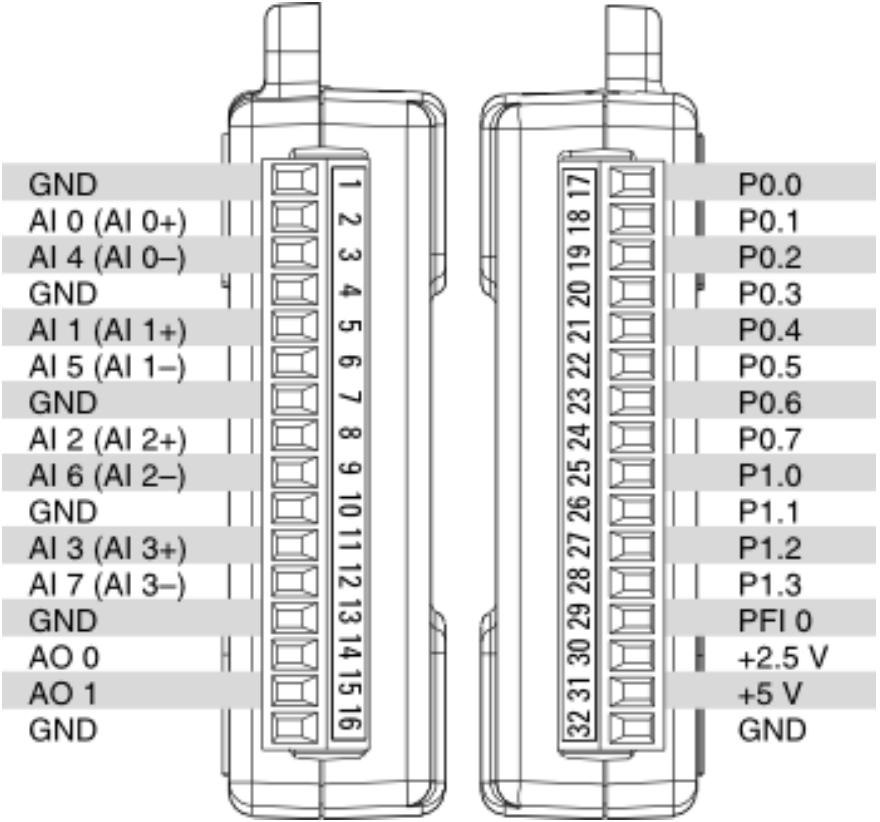
- A DAQ device can be used to read data from Sensors, e.g., a Temperature Sensor (Analog In)
- Or when we want to control something (Analog/Digital Out), e.g., a Heater, Pump, Valve, Light/Dimmer, etc.
- A DAQ device has typically Digital and Analog Channels
- 4 different types of Signals:
  - Analog Outputs (AO)
  - Analog Inputs (AI)
  - Digital Outputs (DO)
  - Digital Inputs (DI)

Analog Channels typically have values between 0-5V/0-10V
Digital Channels are either 0/False (~0V) or 1/True (~2-5V)

NI DAQ Devices

Table of Contents

USB-6008



GND	1	17	P0.0
AI 0 (AI 0+)	2	18	P0.1
AI 4 (AI 0–)	3	19	P0.2
GND	4	20	P0.3
AI 1 (AI 1+)	5	21	P0.4
AI 5 (AI 1–)	6	22	P0.5
GND	7	23	P0.6
AI 2 (AI 2+)	8	24	P0.7
AI 6 (AI 2–)	9	25	P1.0
GND	10	26	P1.1
AI 3 (AI 3+)	11	27	P1.2
AI 7 (AI 3–)	12	28	P1.3
GND	13	29	PFI 0
AO 0	14	30	+2.5 V
AO 1	15	31	+5 V
GND	16	32	GND

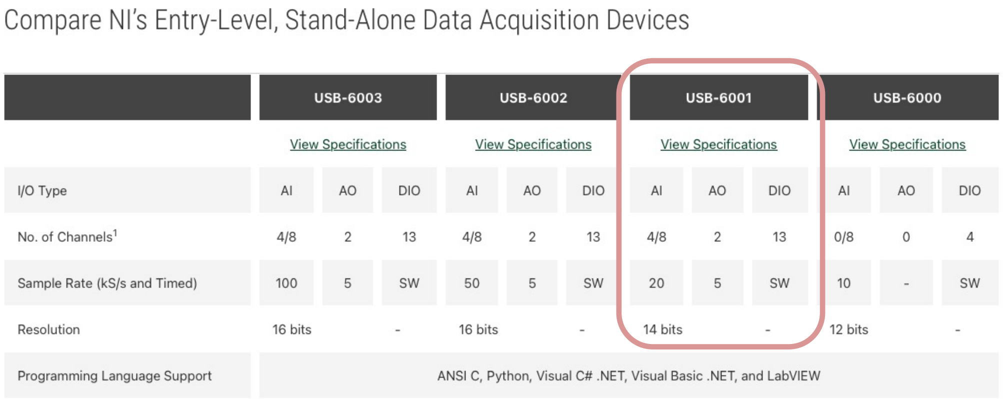
https://www.ni.com/docs/en-US/bundle/usb-6008-specs/page/specs.html

USB-600x DAQ series

Entry-Level, Plug-and-Play USB Data Acquisition

You depend on accurate measurements to make key decisions and discoveries, and NI's plug-and-play, USB multifunction I/O devices deliver quality measurements at an entry-level price.

USB-6008 has been replaced with newer versions like USB-6000, **USB-6001**, USB-6002 and USB-6003 which have similar functionality as USB-6008 and they all work in the same manner, and they all use the NI-DAQmx driver



Compare NI's Entry-Level, Stand-Alone Data Acquisition Devices

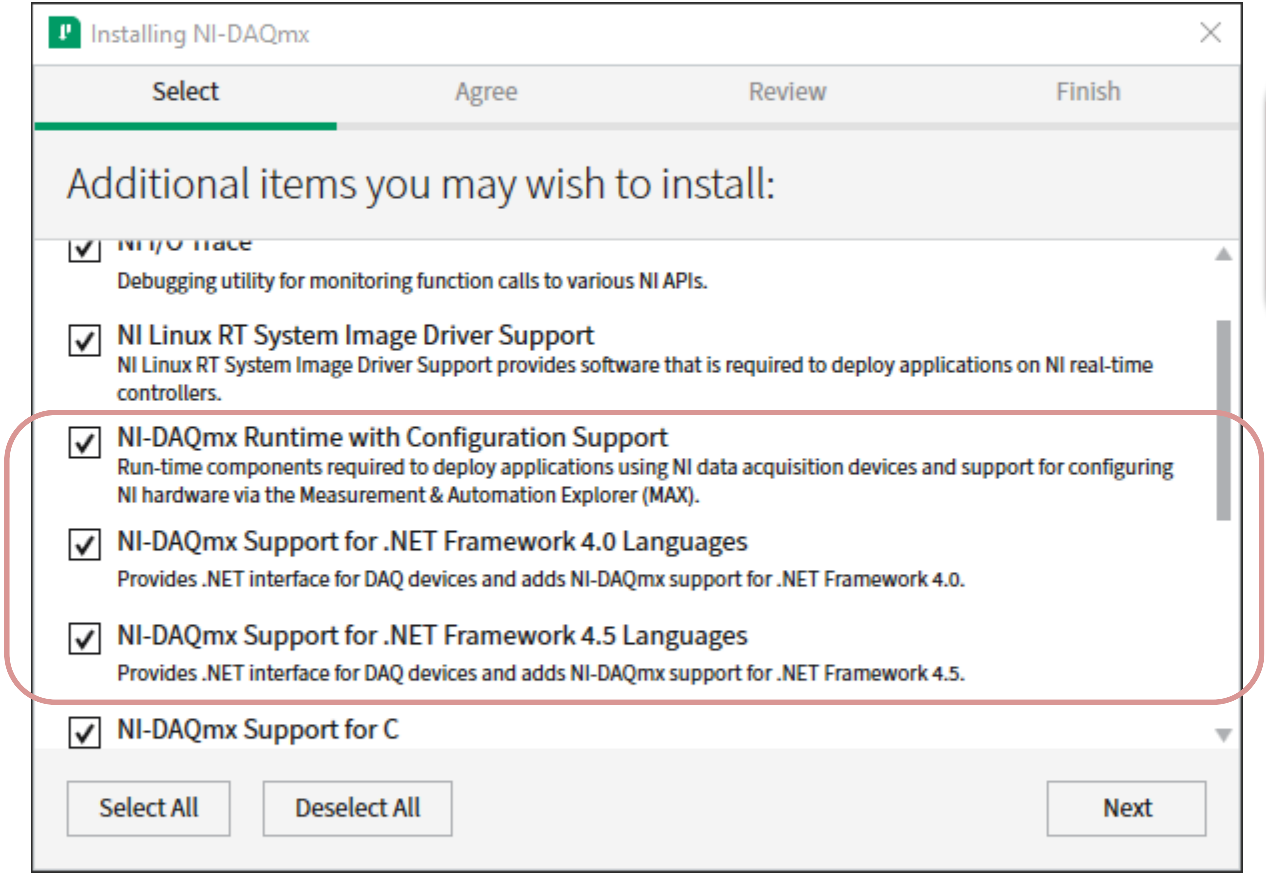
	USB-6003			USB-6002			USB-6001			USB-6000		
	View Specifications			View Specifications			View Specifications			View Specifications		
I/O Type	AI	AO	DIO	AI	AO	DIO	AI	AO	DIO	AI	AO	DIO
No. of Channels[1]	4/8	2	13	4/8	2	13	4/8	2	13	0/8	0	4
Sample Rate (kS/s and Timed)	100	5	SW	50	5	SW	20	5	SW	10	-	SW
Resolution	16 bits		-	16 bits		-	14 bits		-	12 bits		-
Programming Language Support	ANSI C, Python, Visual C# .NET, Visual Basic .NET, and LabVIEW											

https://www.ni.com/en-no/shop/data-acquisition/entry-level-usb-daq.html

NI-DAQmx

- NI-DAQmx is the driver software you use to communicate with and control your DAQ devices made by NI
- NI-DAQmx can be used with LabVIEW, Visual Studio/C#, Python, MATLAB, etc.
- NI-DAQmx can be downloaded for free (but you need of course to buy a NI-DAQmx compatible DAQ device if you don’t have one already)
- www.ni.com/downloads

NI-DAQmx Installation



Installing NI-DAQmx

Additional items you may wish to install:

☑ NI I/O Trace
Debugging utility for monitoring function calls to various NI APIs.

☑ NI Linux RT System Image Driver Support
NI Linux RT System Image Driver Support provides software that is required to deploy applications on NI real-time controllers.

☑ NI-DAQmx Runtime with Configuration Support
Run-time components required to deploy applications using NI data acquisition devices and support for configuring NI hardware via the Measurement & Automation Explorer (MAX).

☑ NI-DAQmx Support for .NET Framework 4.0 Languages
Provides .NET interface for DAQ devices and adds NI-DAQmx support for .NET Framework 4.0.

☑ NI-DAQmx Support for .NET Framework 4.5 Languages
Provides .NET interface for DAQ devices and adds NI-DAQmx support for .NET Framework 4.5.

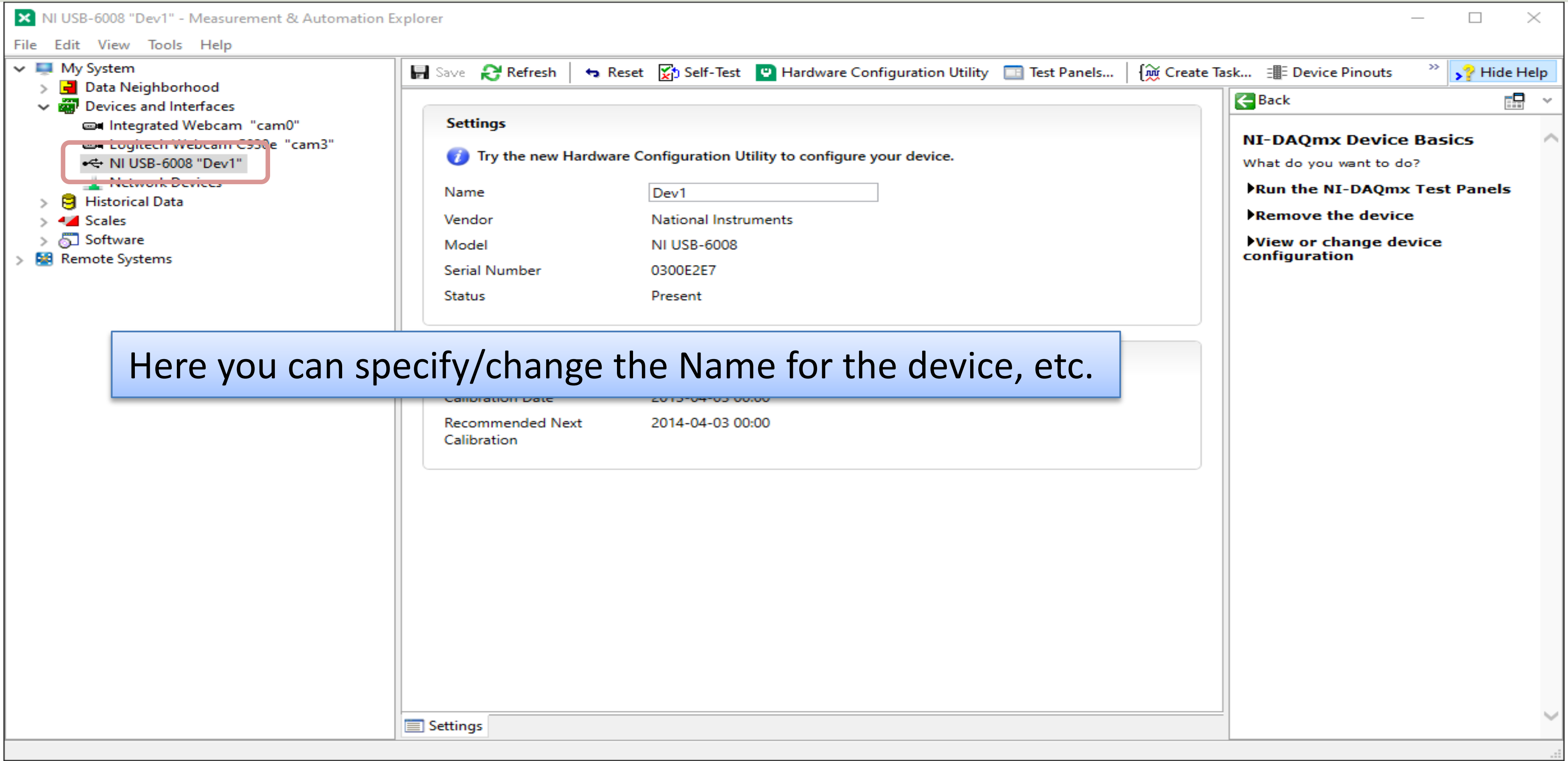
☑ NI-DAQmx Support for C

Make sure to add support for Visual Studio/.NET during installation of the NI-DAQmx software

Measurement & Automation Explorer (MAX)

- MAX is an application that automatically installs with the NI-DAQmx driver.
- With MAX, you can configure your NI hardware.
- MAX informs other programs which devices you have in your system and how they are configured.

Measurement & Automation Explorer (MAX)



Settings

Try the new Hardware Configuration Utility to configure your device.

Name	Dev1
Vendor	National Instruments
Model	NI USB-6008
Serial Number	0300E2E7
Status	Present

Calibration Date	2013-04-03 00:00
Recommended Next Calibration	2014-04-03 00:00

NI-DAQmx Device Basics

What do you want to do?

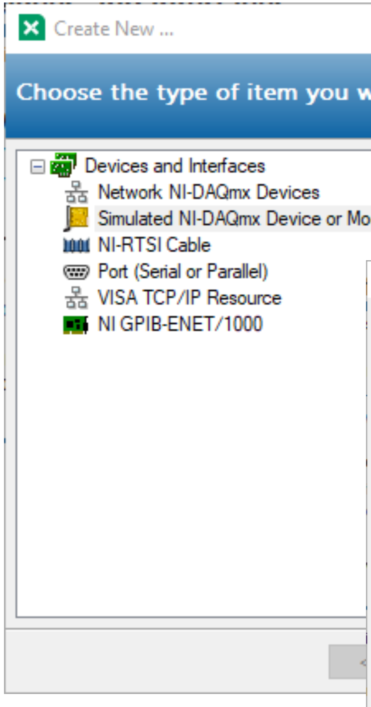
- Run the NI-DAQmx Test Panels
- Remove the device
- View or change device configuration

Here you can specify/change the Name for the device, etc.

NI-DAQmx Simulated Devices

To create an NI-DAQmx simulated device using MAX, complete the following steps:

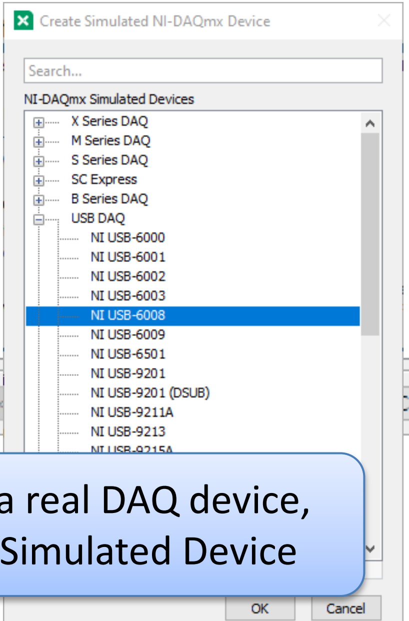
1. Right-click **Devices and Interfaces** and select **Create New**.
2. A dialog box prompts you to select a device to add. Select **Simulated NI-DAQmx Device or Modular Instrument** and click **Finish**.



Create New ...

Choose the type of item you want to add.

- Devices and Interfaces
  - Network NI-DAQmx Devices
  - Simulated NI-DAQmx Device or Modular Instrument
  - NI-RTSI Cable
  - Port (Serial or Parallel)
  - VISA TCP/IP Resource
  - NI GPIB-ENET/1000

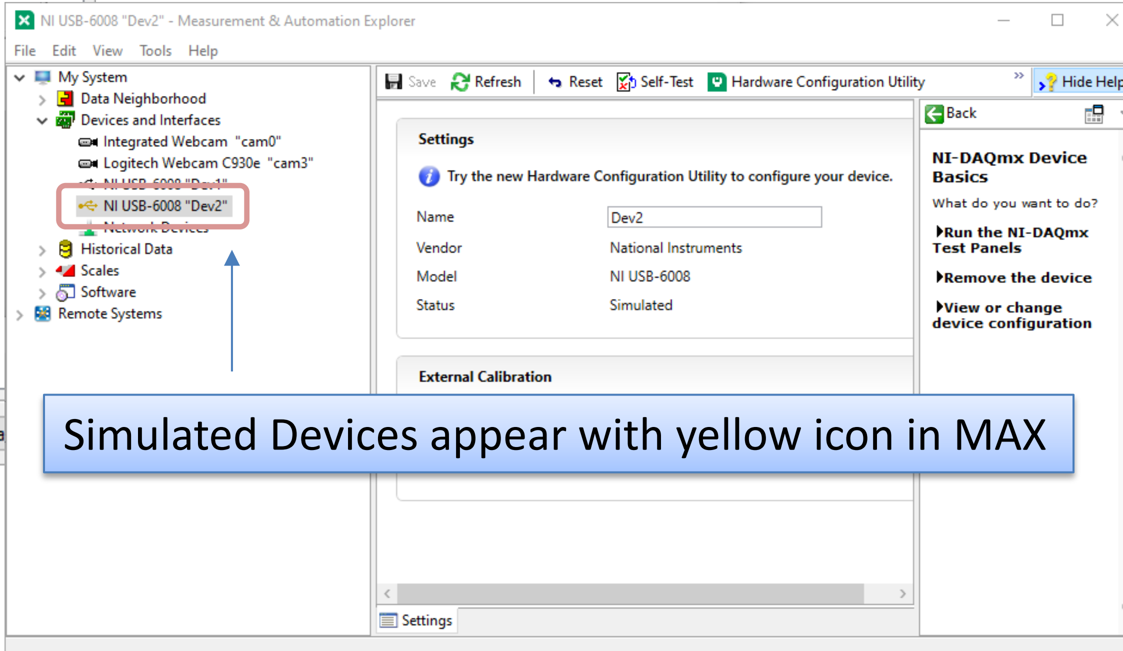


Create Simulated NI-DAQmx Device

Search...

NI-DAQmx Simulated Devices

- X Series DAQ
- M Series DAQ
- S Series DAQ
- SC Express
- B Series DAQ
- USB DAQ
  - NI USB-6000
  - NI USB-6001
  - NI USB-6002
  - NI USB-6003
  - NI USB-6008
  - NI USB-6009
  - NI USB-6501
  - NI USB-9201
  - NI USB-9201 (DSUB)
  - NI USB-9211A
  - NI USB-9213
  - NI USB-9215A



NI USB-6008 "Dev2" - Measurement & Automation Explorer

- My System
  - Data Neighborhood
  - Devices and Interfaces
    - Integrated Webcam "cam0"
    - Logitech Webcam C930e "cam3"
    - NI USB-6008 "Dev1"
    - NI USB-6008 "Dev2"
    - Network Devices
  - Historical Data
  - Scales
  - Software
- Remote Systems

Settings

Try the new Hardware Configuration Utility to configure your device.

Name	Dev2
Vendor	National Instruments
Model	NI USB-6008
Status	Simulated

External Calibration

NI-DAQmx Device Basics

What do you want to do?

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

Simulated Devices appear with yellow icon in MAX

If you don't have a real DAQ device, you can create a Simulated Device

https://www.halvorsen.blog

Visual Studio/C# Code Examples

Hans-Petter Halvorsen

Table of Contents

https://www.halvorsen.blog

Analog In

Hans-Petter Halvorsen

Table of Contents

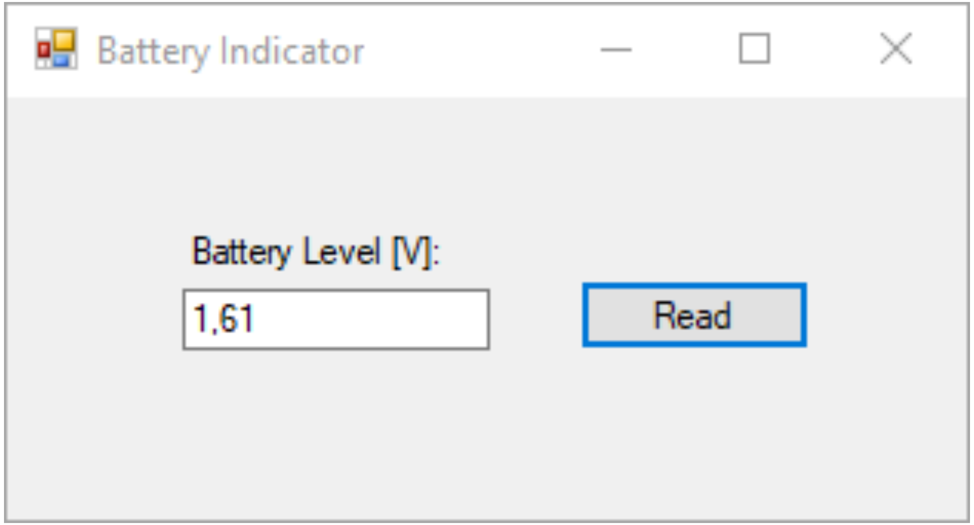
Analog In Example

Battery Indicator Example

Hans-Petter Halvorsen

Table of Contents

Battery Indicator Example

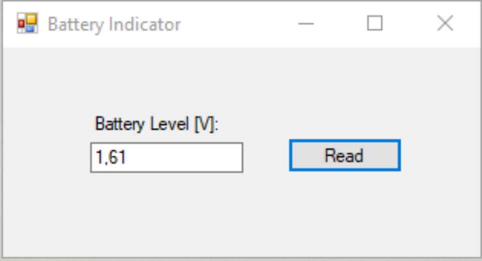


Battery Level [V]: 1,61

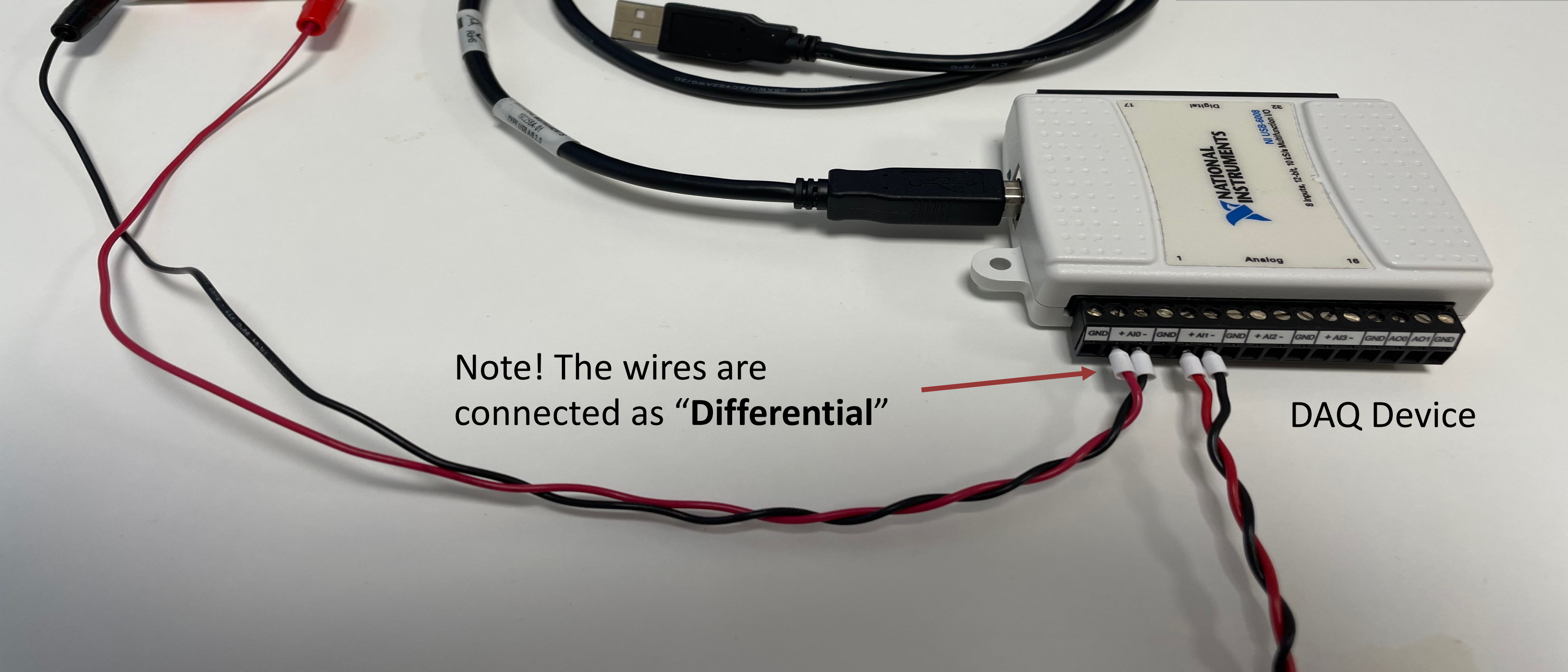
We start with a basic Example just reading the Voltage Value from a 1.5V battery that is connected to the DAQ device

1.5V Battery

Connect USB Cable to PC

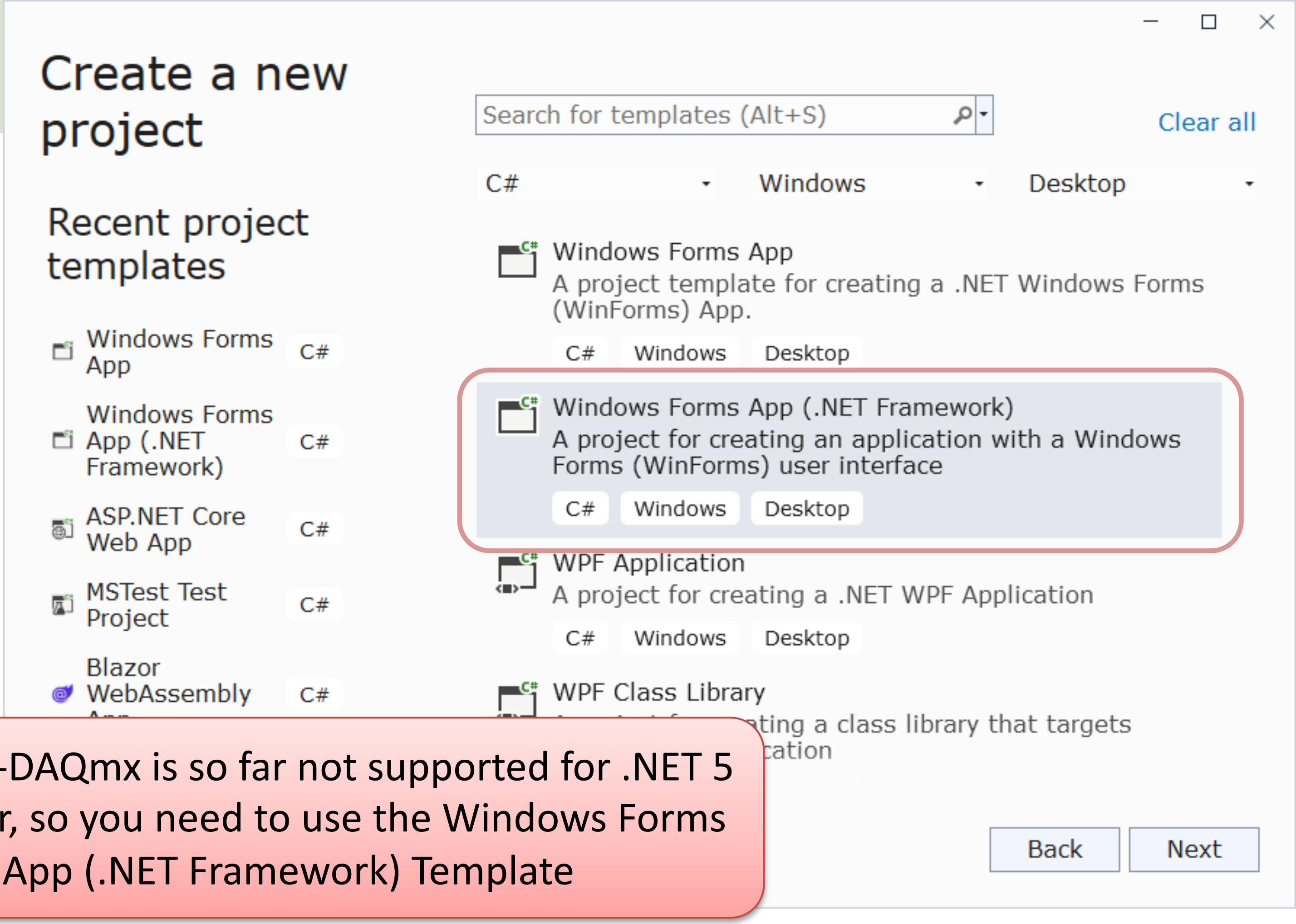


Battery Level [V]: 1,61


NATIONAL INSTRUMENTS
NI USB-6008
8 inputs, 12-bit, 10 kS/s Multifunction I/O
Digital
Analog
GND + AI0 - GND + AI1 - GND + AI2 - GND + AI3 - GND AO0 AO1 GND

Note! The wires are connected as “**Differential**”

DAQ Device



Create a new project

Search for templates (Alt+S)

Clear all

C# | Windows | Desktop

Recent project templates

- Windows Forms App — C#
- Windows Forms App (.NET Framework) — C#
- ASP.NET Core Web App — C#
- MSTest Test Project — C#
- Blazor WebAssembly App — C#

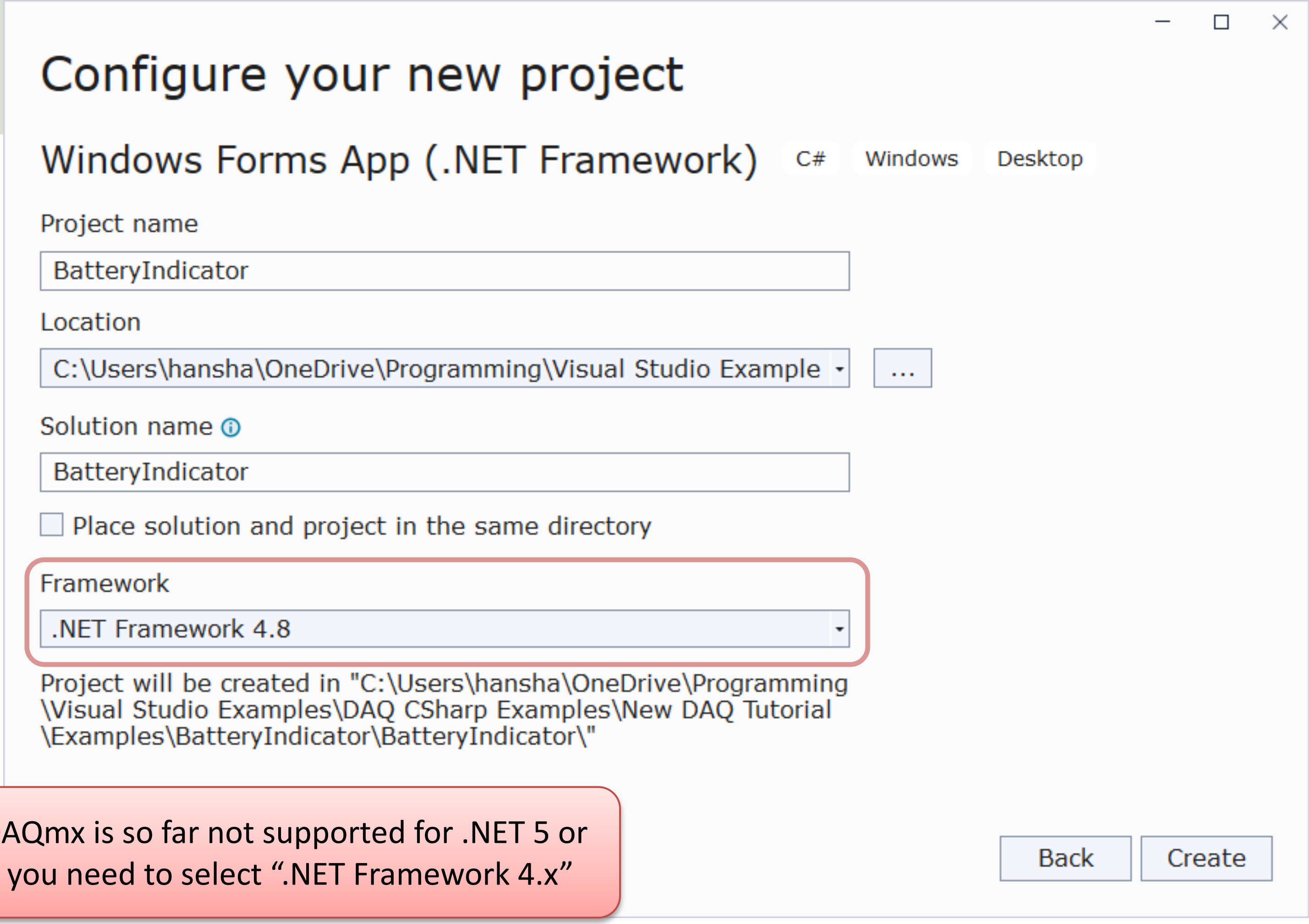
Windows Forms App
A project template for creating a .NET Windows Forms (WinForms) App.
C# · Windows · Desktop

Windows Forms App (.NET Framework)
A project for creating an application with a Windows Forms (WinForms) user interface
C# · Windows · Desktop

WPF Application
A project for creating a .NET WPF Application
C# · Windows · Desktop

WPF Class Library
...ting a class library that targets ...cation

Note! NI-DAQmx is so far not supported for .NET 5 or higher, so you need to use the Windows Forms App (.NET Framework) Template



Configure your new project

Windows Forms App (.NET Framework)

Project name

BatteryIndicator

Location

C:\Users\hansha\OneDrive\Programming\Visual Studio Example

Solution name ⓘ

BatteryIndicator

□ Place solution and project in the same directory

Framework

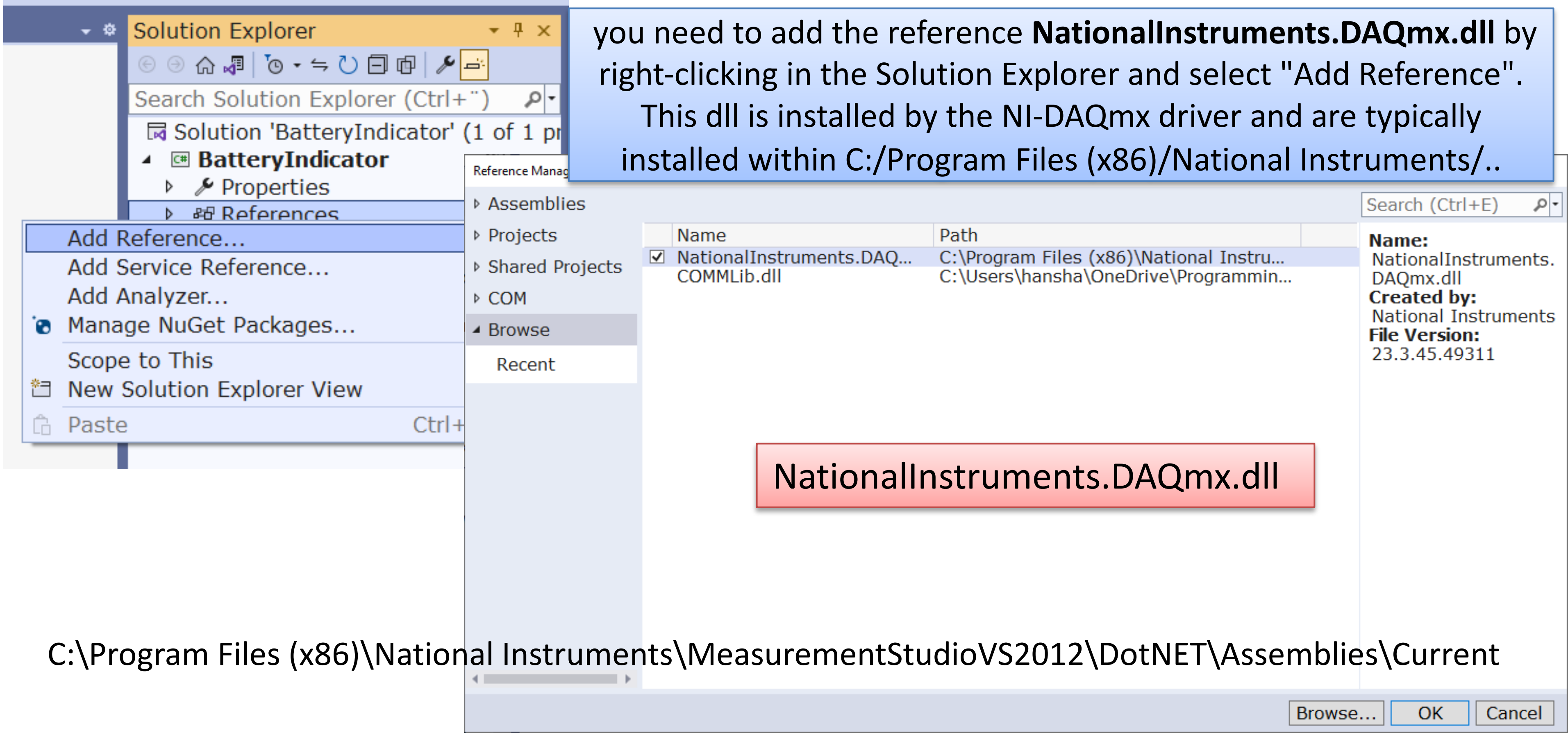
.NET Framework 4.8

Project will be created in "C:\Users\hansha\OneDrive\Programming\Visual Studio Examples\DAQ CSharp Examples\New DAQ Tutorial\Examples\BatteryIndicator\BatteryIndicator\"

Note! NI-DAQmx is so far not supported for .NET 5 or higher, so you need to select ".NET Framework 4.x"

Add Reference

you need to add the reference **NationalInstruments.DAQmx.dll** by right-clicking in the Solution Explorer and select "Add Reference". This dll is installed by the NI-DAQmx driver and are typically installed within C:/Program Files (x86)/National Instruments/..



	Name	Path
☑	NationalInstruments.DAQ...	C:\Program Files (x86)\National Instru...
	COMMLib.dll	C:\Users\hansha\OneDrive\Programmin...

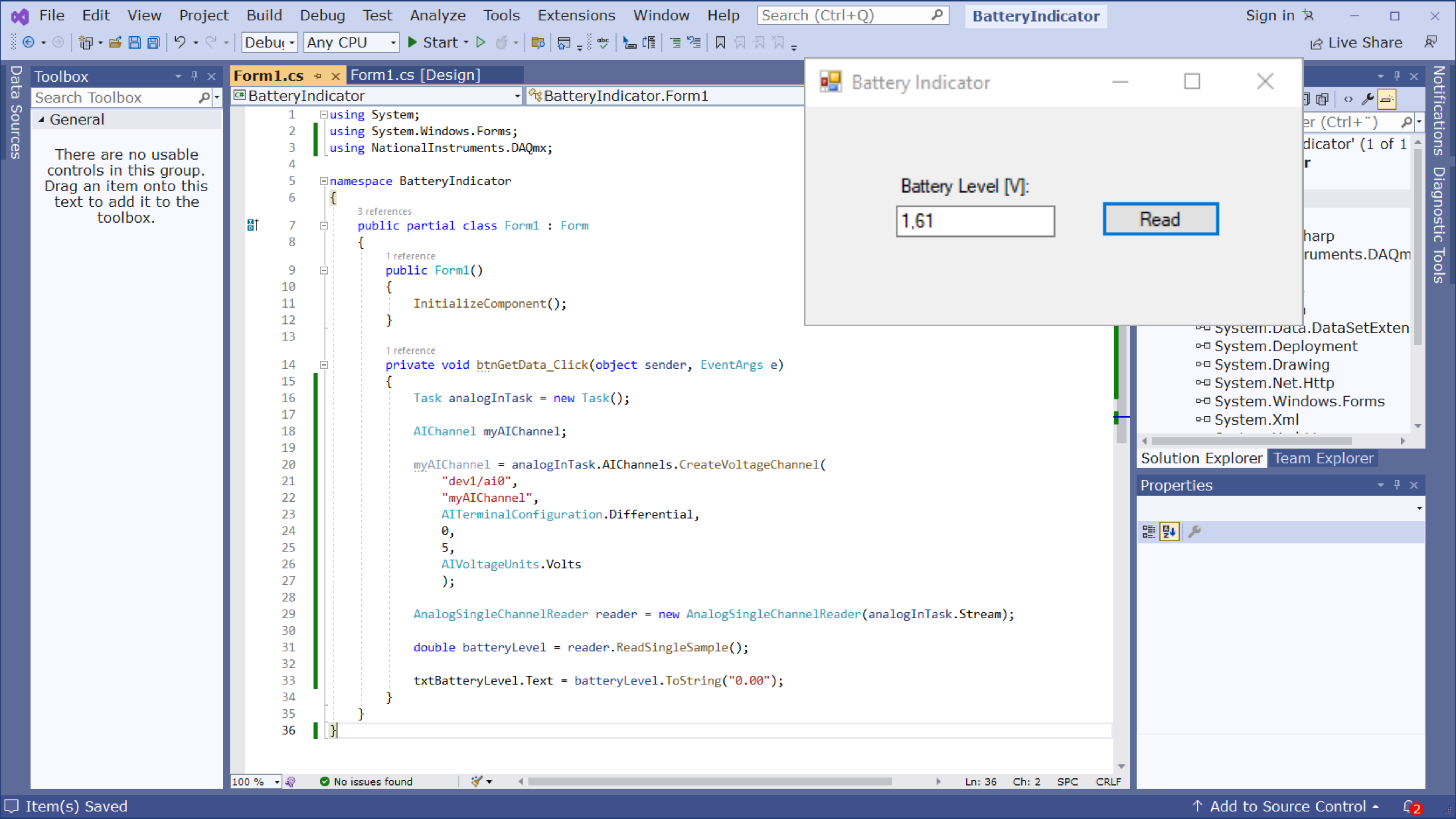
Name: NationalInstruments.DAQmx.dll
Created by: National Instruments
File Version: 23.3.45.49311

NationalInstruments.DAQmx.dll

C:\Program Files (x86)\National Instruments\MeasurementStudioVS2012\DotNET\Assemblies\Current

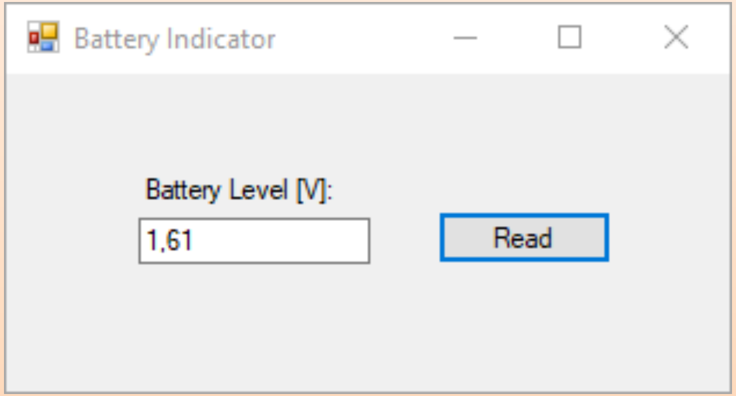
```
..
using NationalInstruments.DAQmx;
..
Task analogInTask = new Task();

AIChannel myAIChannel;

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

AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

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

We can choose between “RSE” and “Differential”. We have used **Differential** wiring in this example



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

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

        private void btnGetData_Click(object sender, EventArgs e)
        {
            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 batteryLevel = reader.ReadSingleSample();

            txtBatteryLevel.Text = batteryLevel.ToString("0.00");
        }
    }
}
```

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

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

      private void btnGetData_Click(object sender, EventArgs e)
      {
         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 batteryLevel = reader.ReadSingleSample();

         txtBatteryLevel.Text = batteryLevel.ToString("0.00");
      }
   }
}
```



Battery Level [V]: 1,61

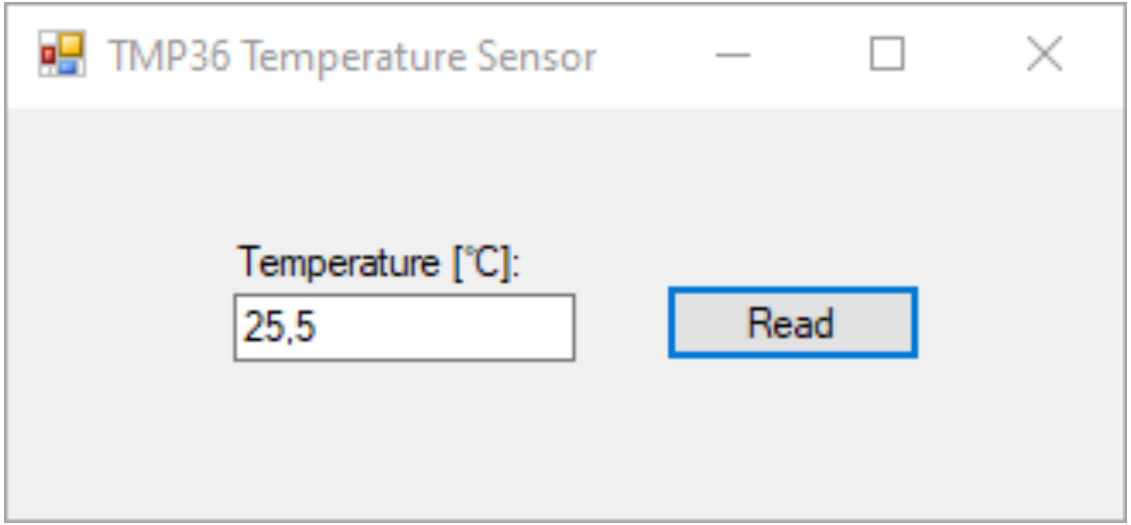
Analog In Example

TMP36 Temperature Sensor Example

Hans-Petter Halvorsen

Table of Contents

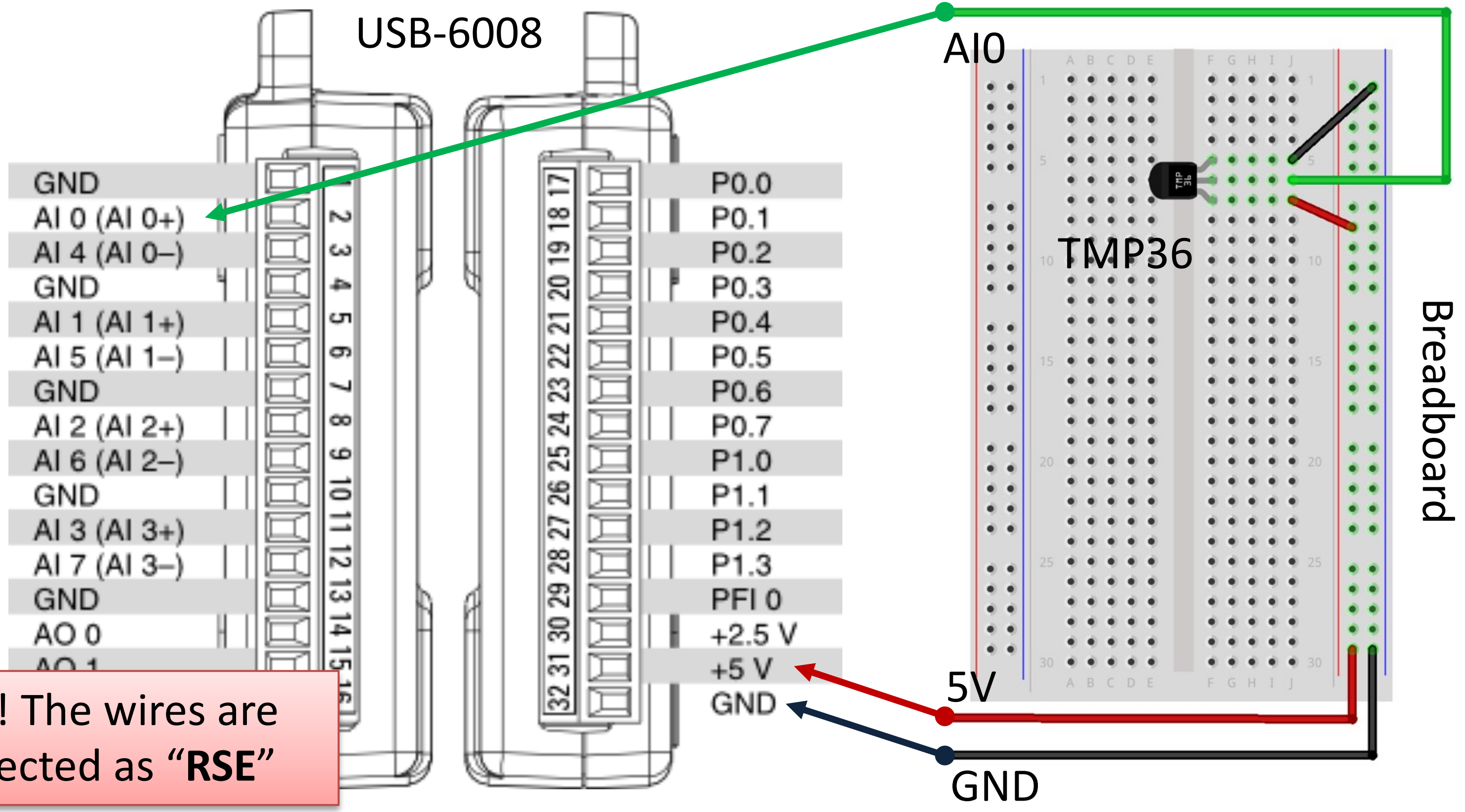
TMP36 Temperature Sensor Example



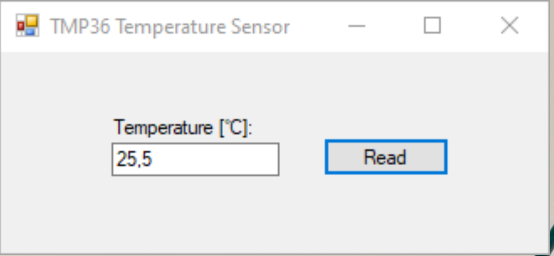
Temperature [°C]: 25,5

In this example we will use a TMP36 Temperature Sensor and read from the DAQ device and calculate the Temperature value in degrees Celsius.

TMP36 Wiring Example

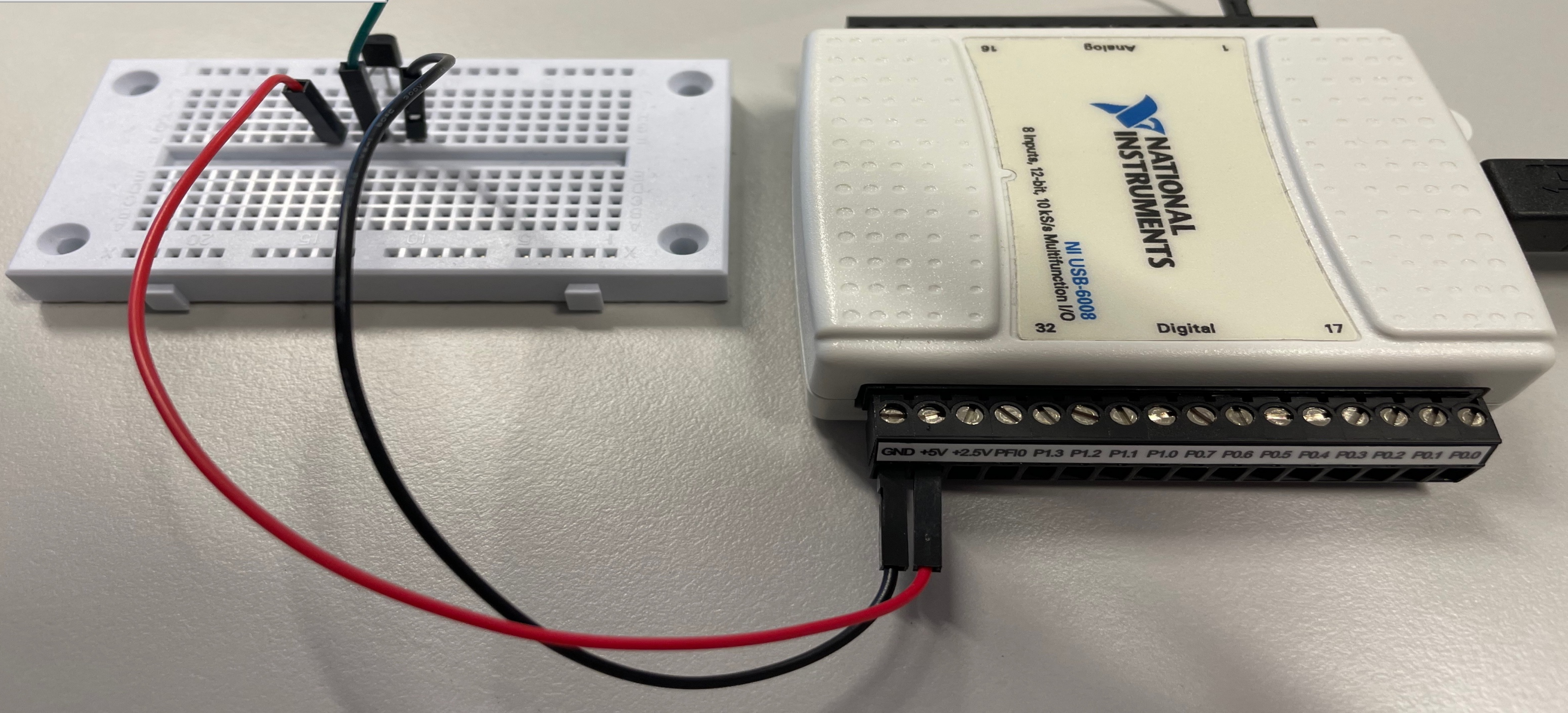


Note! The wires are connected as “**RSE**”



TMP36 Temperature Sensor

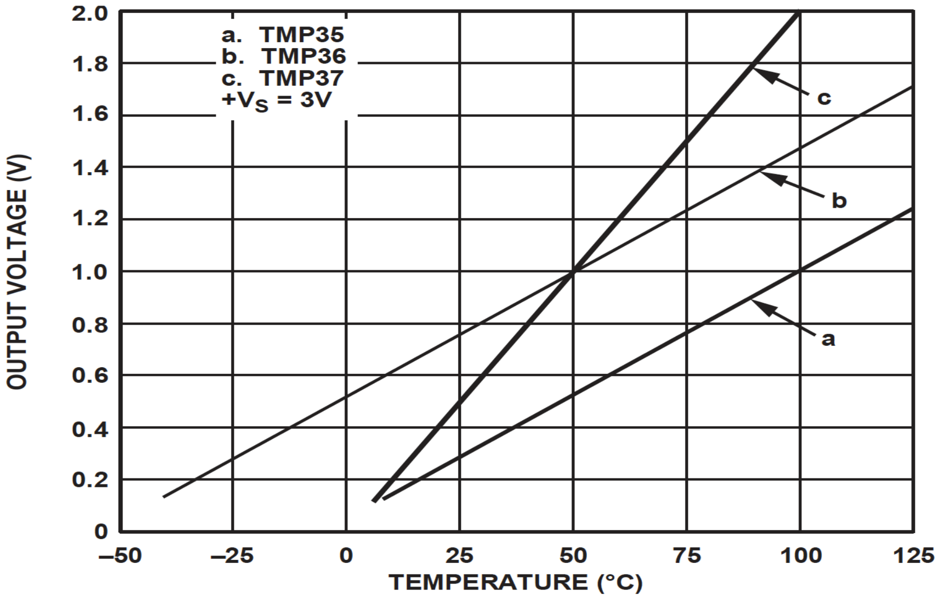
Temperature [°C]: 25,5



Note! The wires are connected as “**RSE**”

TMP36 Temperature Sensor

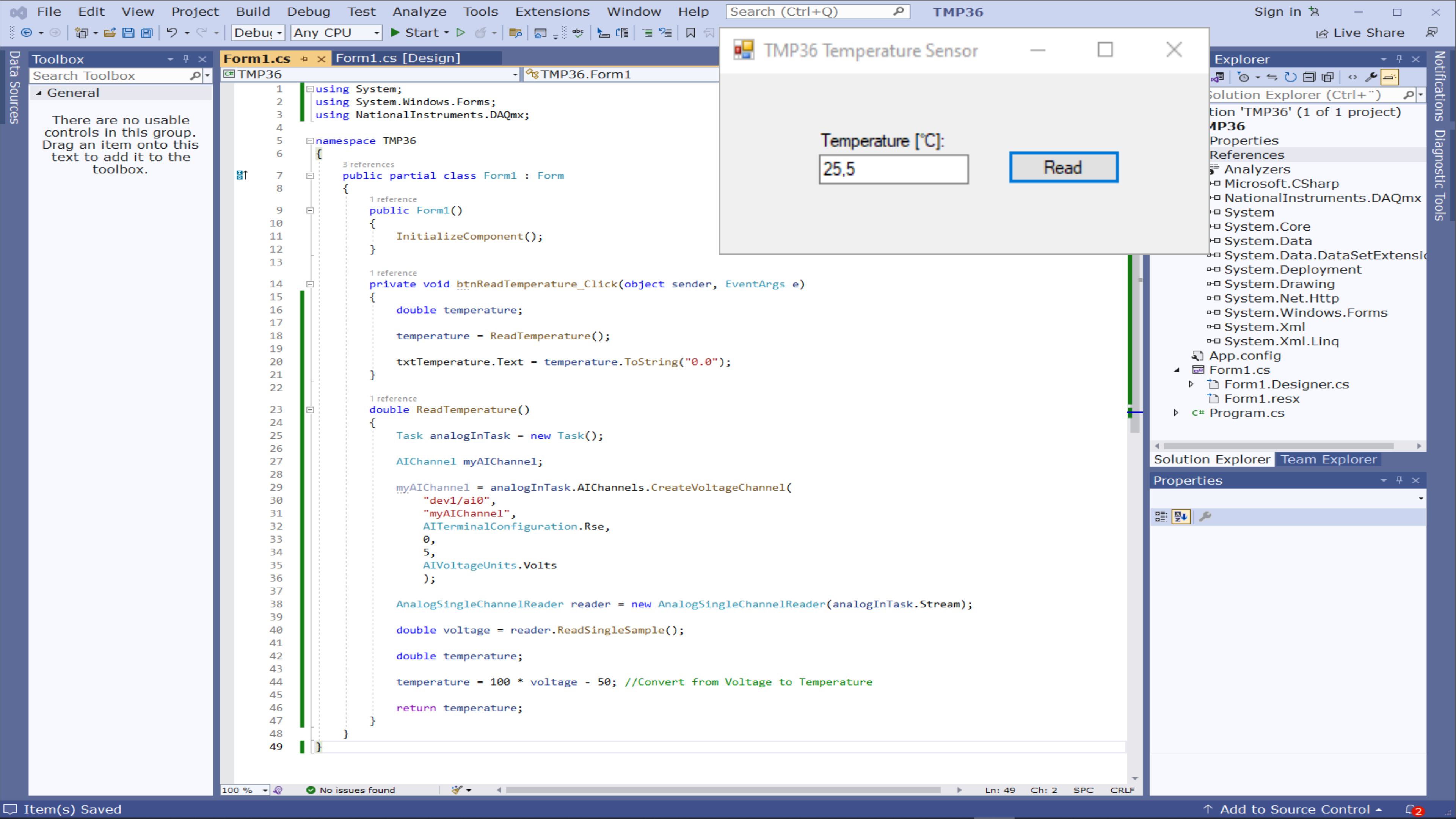
Figure from Datasheet:



Formula for converting from Voltage to Temperature in Degrees Celsius:

$y = 100x - 50$ where x is the value read from the DAQ device in voltage

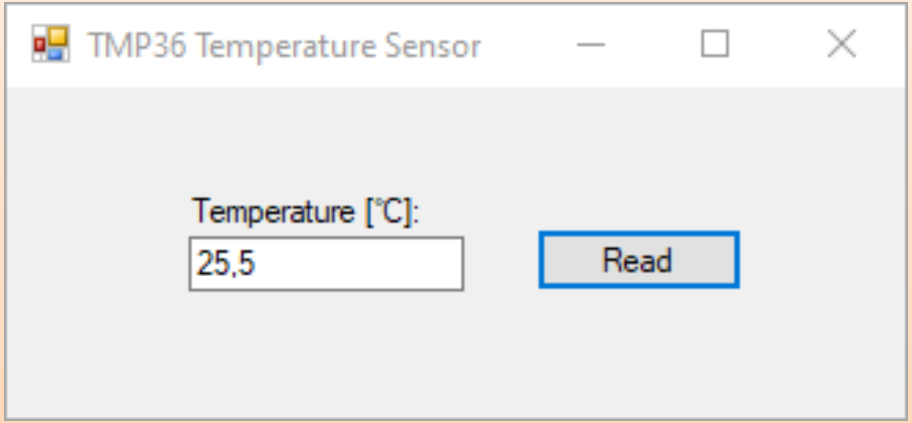
```
double ReadTemperature()
{
  Task analogInTask = new Task();

  AIChannel myAIChannel;

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

  AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

  double voltage = reader.ReadSingleSample();

  double temperature;

  temperature = 100 * voltage - 50; //Convert from Voltage to Temperature

  return temperature;
}
```



Temperature [°C]: 25,5

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

namespace TMP36
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnReadTemperature_Click(object sender, EventArgs e)
        {
            double temperature;

            temperature = ReadTemperature();

            txtTemperature.Text = temperature.ToString("0.0");
        }

        double ReadTemperature()
        {
            Task analogInTask = new Task();

            AIChannel myAIChannel;

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

            AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

            double voltage = reader.ReadSingleSample();

            double temperature;

            temperature = 100 * voltage - 50; //Convert from Voltage to Temperature

            return temperature;
        }
    }
}
```

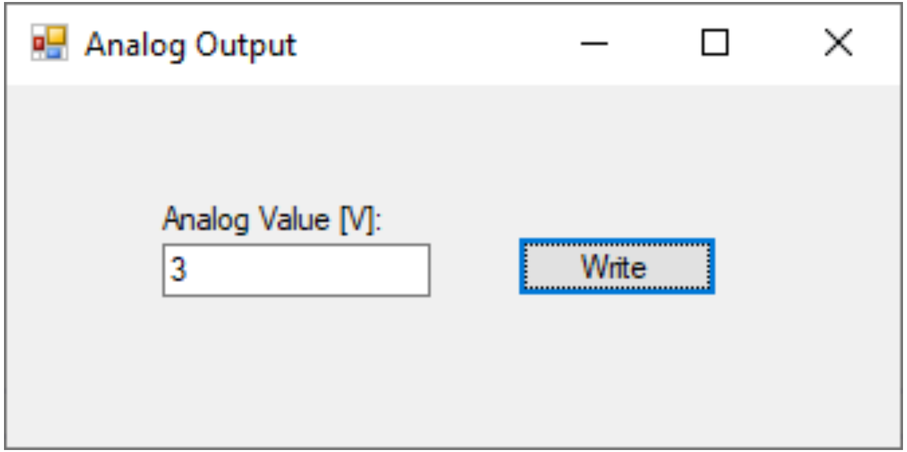
```
using System;
using System.Windows.Forms;
using NationalInstruments.DAQmx;

namespace TMP36
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void btnReadTemperature_Click(object sender, EventArgs e)
        {
            double temperature;

            temperature = ReadTemperature();

            txtTemperature.Text = temperature.ToString("0.0");
        }

        double ReadTemperature()
        {
            Task analogInTask = new Task();

            AIChannel myAIChannel;

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

            AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

            double voltage = reader.ReadSingleSample();

            double temperature;

            temperature = 100 * voltage - 50; //Convert from Voltage to Temperature

            return temperature;
        }
    }
}
```



Temperature [°C]: 25,5

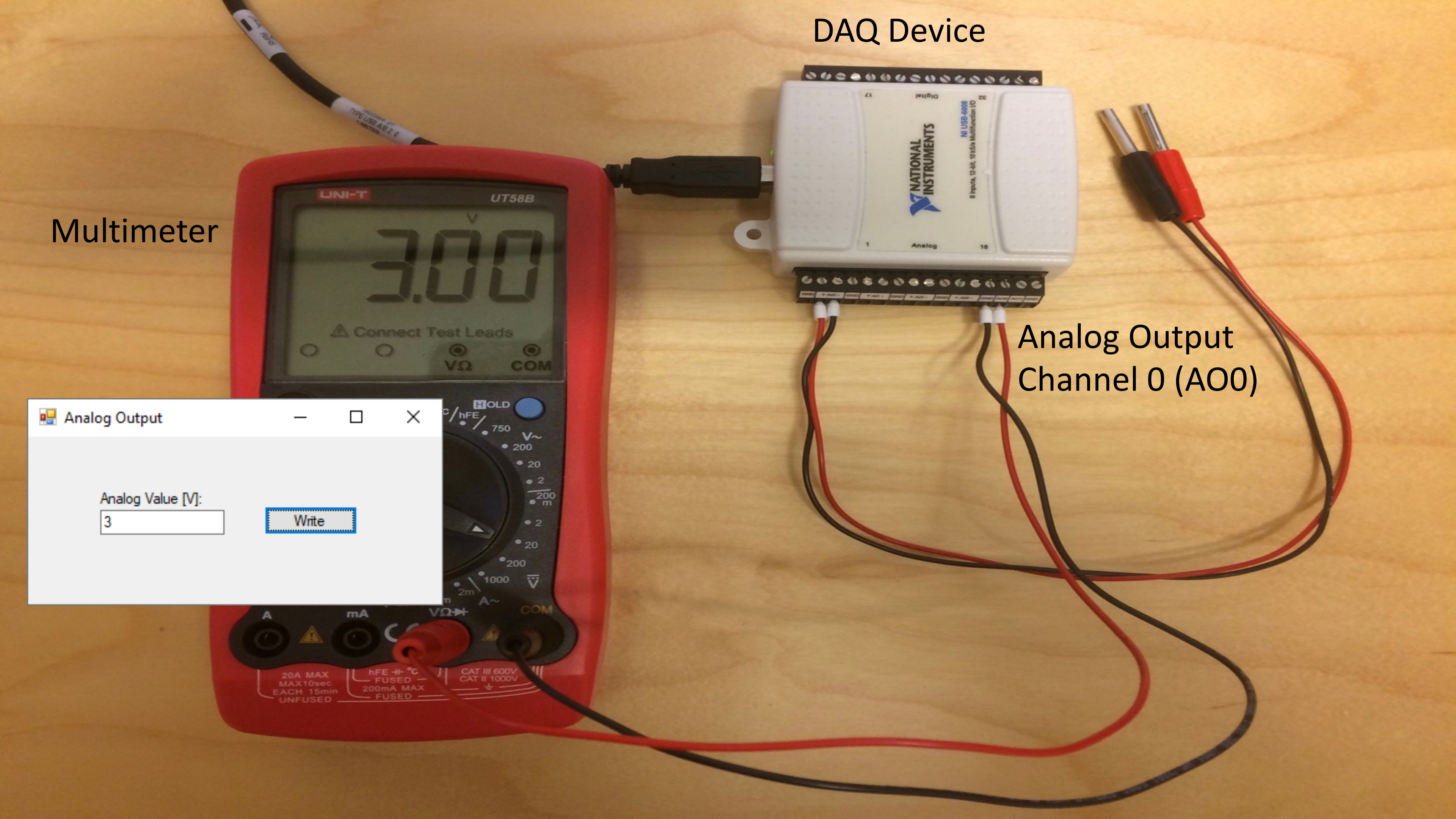
Analog Out

Basic Analog Out Example



Analog Value [V]: 3

This Analog Out Example write a Value to the Analog Out 0 Channel (AO0) on the DAQ device. We can connect a Multimeter to see if the Application works as expected



DAQ Device

Multimeter

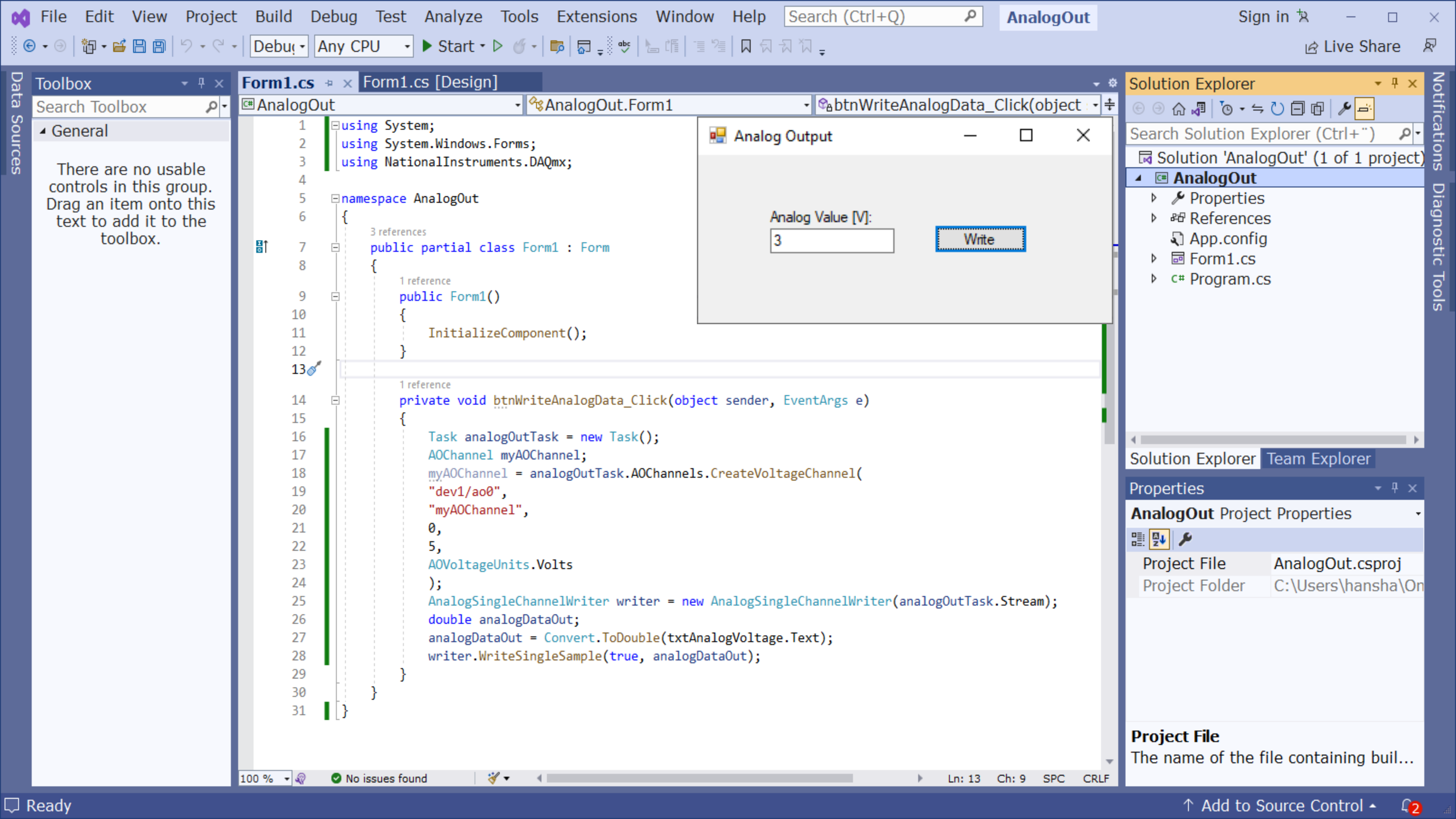
Analog Output Channel 0 (AO0)

Analog Output

Analog Value [V]: 3

Write

```
..
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(txtAnalogVoltage.Text);

writer.WriteSingleSample(true, analogDataOut);

..
```



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

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

        private void btnWriteAnalogData_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(txtAnalogVoltage.Text);
            writer.WriteSingleSample(true, analogDataOut);
        }
    }
}
```

Analog Output

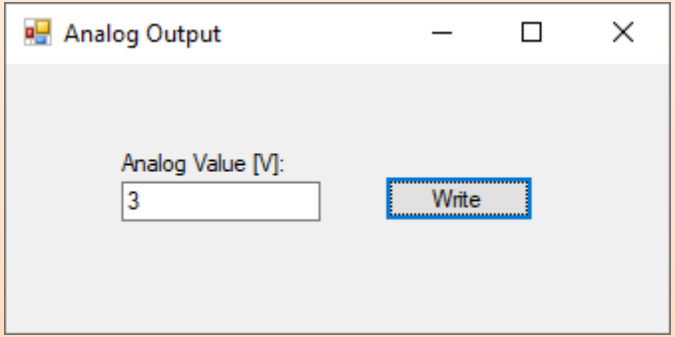
Analog Value [V]: 3

Write

Project File: AnalogOut.csproj

Project Folder: C:\Users\hansha\On

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

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

        private void btnWriteAnalogData_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(txtAnalogVoltage.Text);
            writer.WriteSingleSample(true, analogDataOut);
        }
    }
}
```



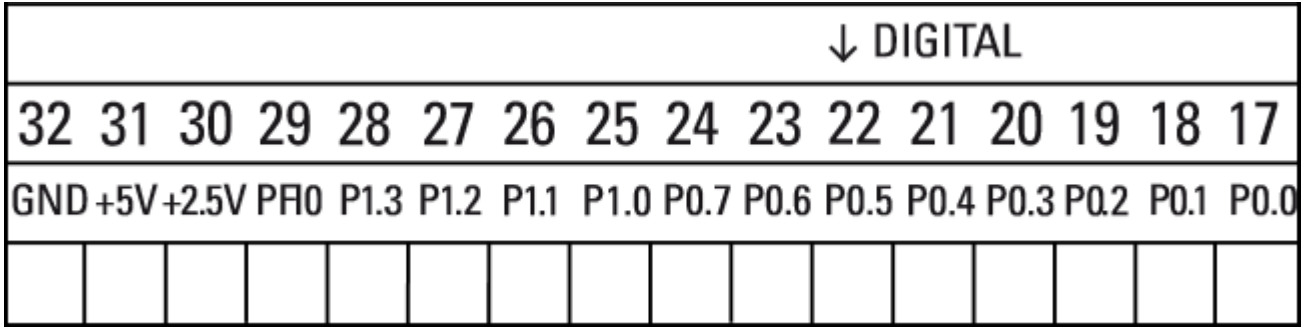
Analog Value [V]: 3

Digital I/O

Hans-Petter Halvorsen

Table of Contents

Digital Chanels



↓ DIGITAL

32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17
GND	+5V	+2.5V	PFI0	P1.3	P1.2	P1.1	P1.0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0

Dev1/Port0/line0:7

P0.<0..7> Port 0 Digital I/O Channels 0 to 7 — You can individually configure each signal as an input or output.

Dev1/Port1/line0:3

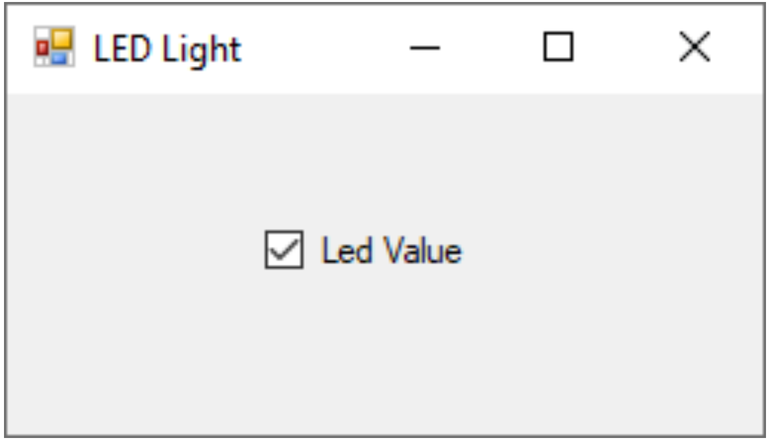
P1.<0..3> Port 1 Digital I/O Channels 0 to 3 — You can individually configure each signal as an input or output

Digital Out

Digital Out Example

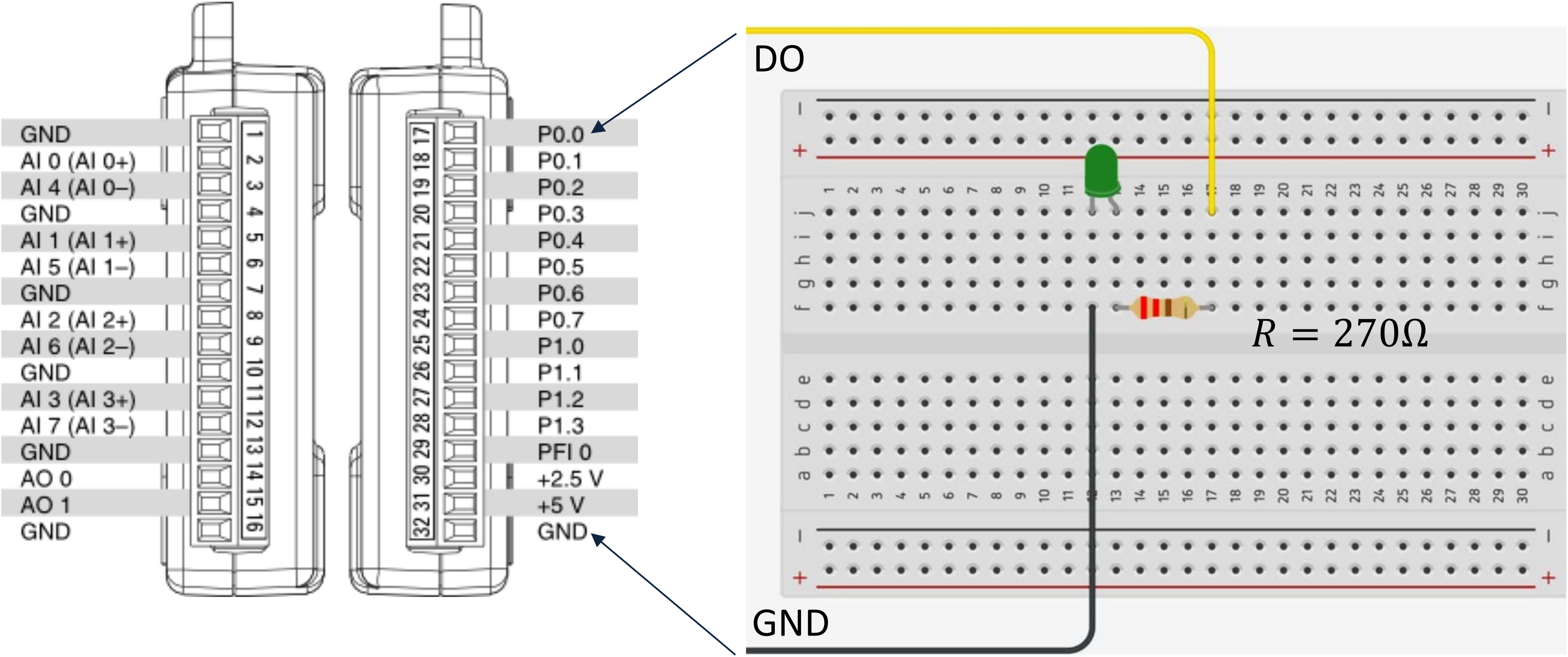
LED Example

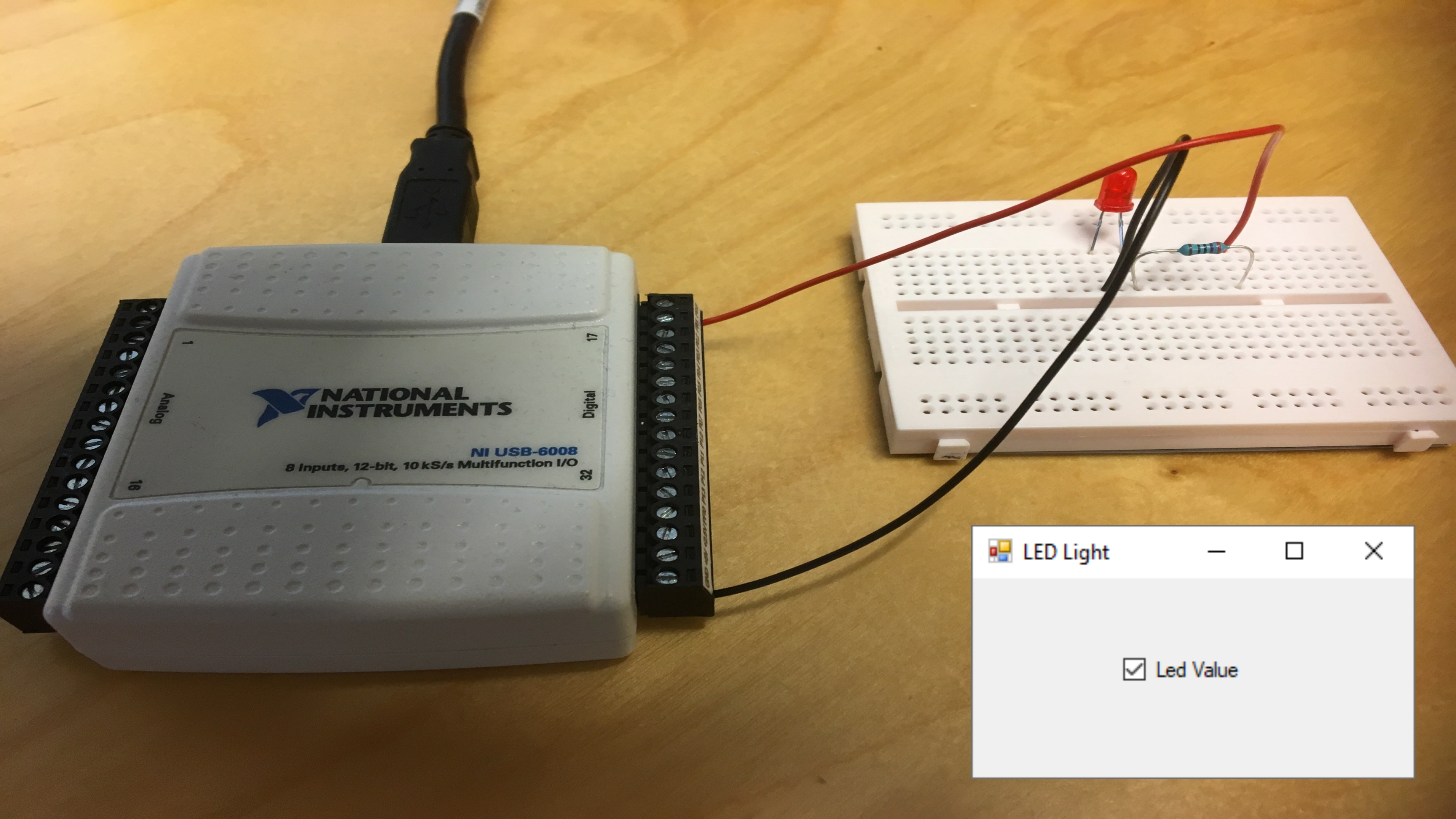
Basic Digital Out Example



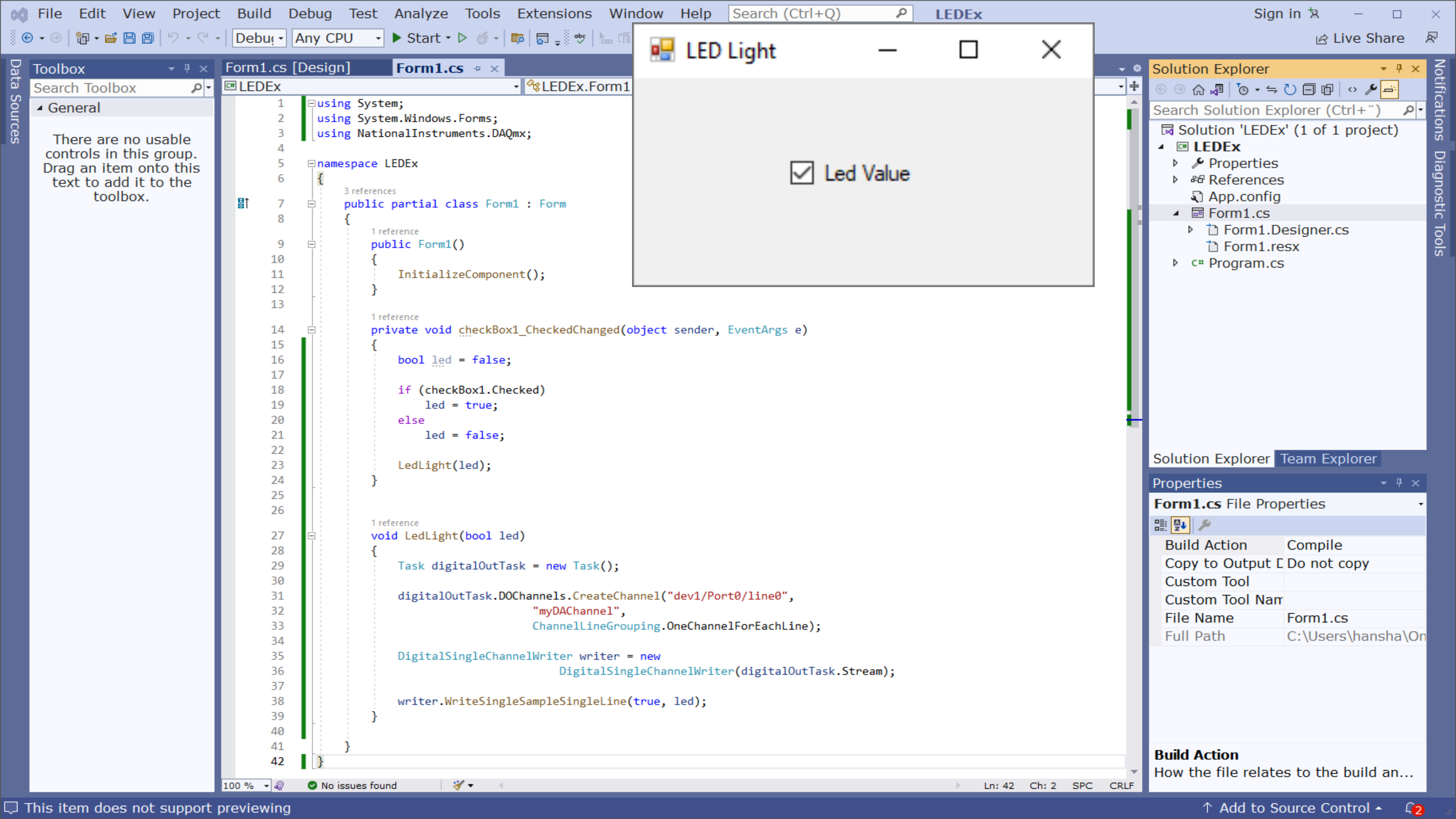
This Digital Out Example writes a Value to the Digital Out Port 0, Line 0 on the DAQ device. We can connect a Multimeter to see if the Application works as expected or we can connect a LED, etc.

Wiring Example




NATIONAL INSTRUMENTS
NI USB-6008
8 Inputs, 12-bit, 10 kS/s Multifunction I/O
1
17
Analog
Digital
16
32
LED Light
☑ Led Value

```
void LedLight(bool led)
{
    Task digitalOutTask = new Task();

    digitalOutTask.DOChannels.CreateChannel("dev1/Port0/line0",
                "myDAChannel",
                ChannelLineGrouping.OneChannelForEachLine);

    DigitalSingleChannelWriter writer = new
                 DigitalSingleChannelWriter(digitalOutTask.Stream);

    writer.WriteSingleSampleSingleLine(true, led);
}
```



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

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

        private void checkBox1_CheckedChanged(object sender, EventArgs e)
        {
            bool led = false;

            if (checkBox1.Checked)
                led = true;
            else
                led = false;

            LedLight(led);
        }


        void LedLight(bool led)
        {
            Task digitalOutTask = new Task();

            digitalOutTask.DOChannels.CreateChannel("dev1/Port0/line0",
                                "myDAChannel",
                               ChannelLineGrouping.OneChannelForEachLine);

            DigitalSingleChannelWriter writer = new
                                 DigitalSingleChannelWriter(digitalOutTask.Stream);

            writer.WriteSingleSampleSingleLine(true, led);
        }

    }
}
```

Properties

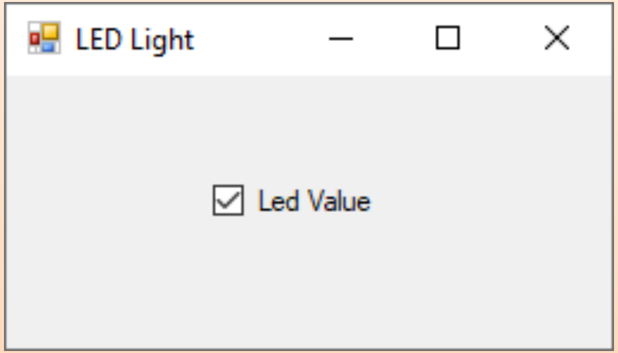
Form1.cs File Properties

Build Action	Compile
Copy to Output D	Do not copy
Custom Tool	
Custom Tool Nam	
File Name	Form1.cs
Full Path	C:\Users\hansha\On

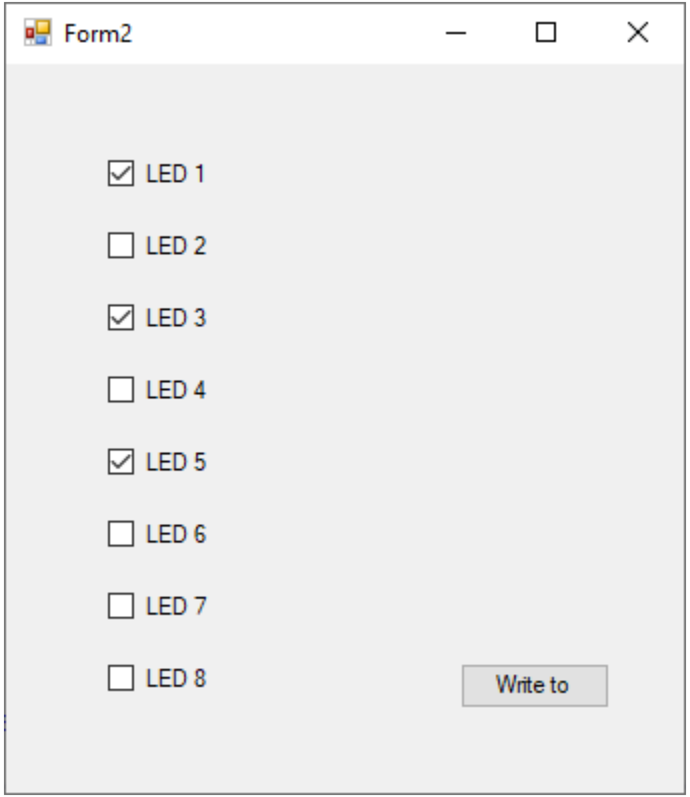
Build Action
How the file relates to the build an...

This item does not support previewing

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

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

    private void checkBox1_CheckedChanged(object sender, EventArgs e)
    {
      bool led = false;

      if (checkBox1.Checked)
        led = true;
      else
        led = false;

      LedLight(led);
    }


    void LedLight(bool led)
    {
      Task digitalOutTask = new Task();

      digitalOutTask.DOChannels.CreateChannel("dev1/Port0/line0",
                 "myDAChannel",
                 ChannelLineGrouping.OneChannelForEachLine);

      DigitalSingleChannelWriter writer = new DigitalSingleChannelWriter(digitalOutTask.Stream);

      writer.WriteSingleSampleSingleLine(true, led);
    }
  }
}
```



Multiple LEDs



☑ LED 1

□ LED 2

☑ LED 3

□ LED 4

☑ LED 5

□ LED 6

□ LED 7

□ LED 8

If you don’t have 8 LEDs, use a Multimeter to check the voltage value on the Digital Output Channels on the DAQ Device

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

namespace LEDApp
{
   public partial class Form2 : Fo
   {
      public Form2()
      {
         InitializeComponent();
      }
      private void btnWriteDaq_Click(object sender, EventArgs e)
      {
         Task digitalOutTask = new Task();

         digitalOutTask.DOChannels.CreateChannel("dev1/Port0/line0:7",
                                   "myDAChannel", ChannelLineGrouping.OneChannelForAllLines);

         DigitalSingleChannelWriter writer = new DigitalSingleChannelWriter(digitalOutTask.Stream);

         bool[] dataArray = new bool[8];
         dataArray[0] = chkLed1.Checked;
         dataArray[1] = chkLed2.Checked;
         dataArray[2] = chkLed3.Checked;
         dataArray[3] = chkLed4.Checked;
         dataArray[4] = chkLed5.Checked;
         dataArray[5] = chkLed6.Checked;
         dataArray[6] = chkLed7.Checked;
         dataArray[7] = chkLed8.Checked;

         writer.WriteSingleSampleMultiLine(true, dataArray);
      }
   }
}
```

https://www.halvorsen.blog

Digital In

Hans-Petter Halvorsen

Table of Contents

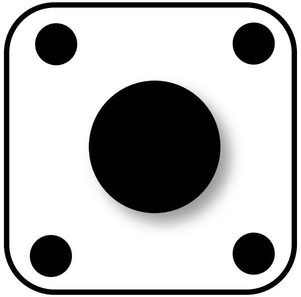
Digital In Example

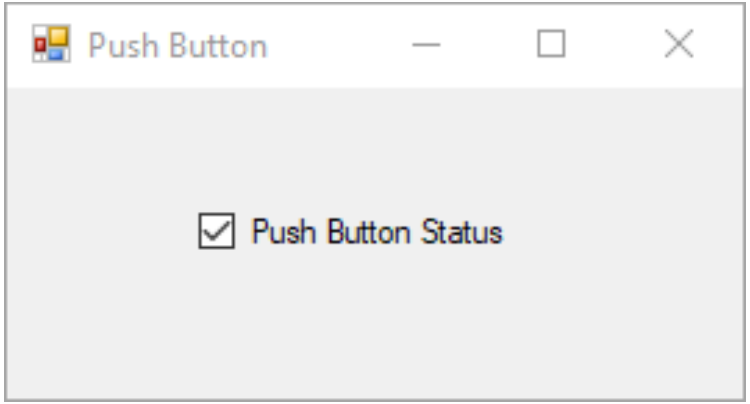
Push Button Example

Hans-Petter Halvorsen

Table of Contents

Push Button Example

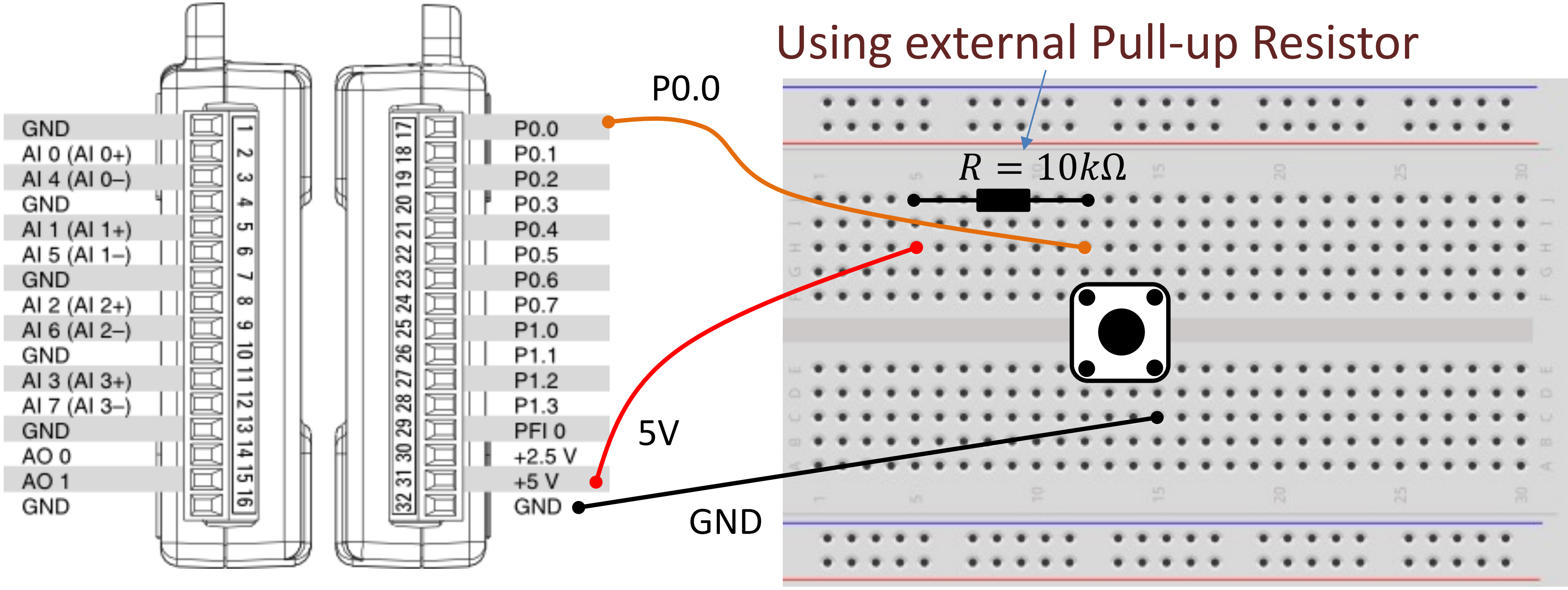


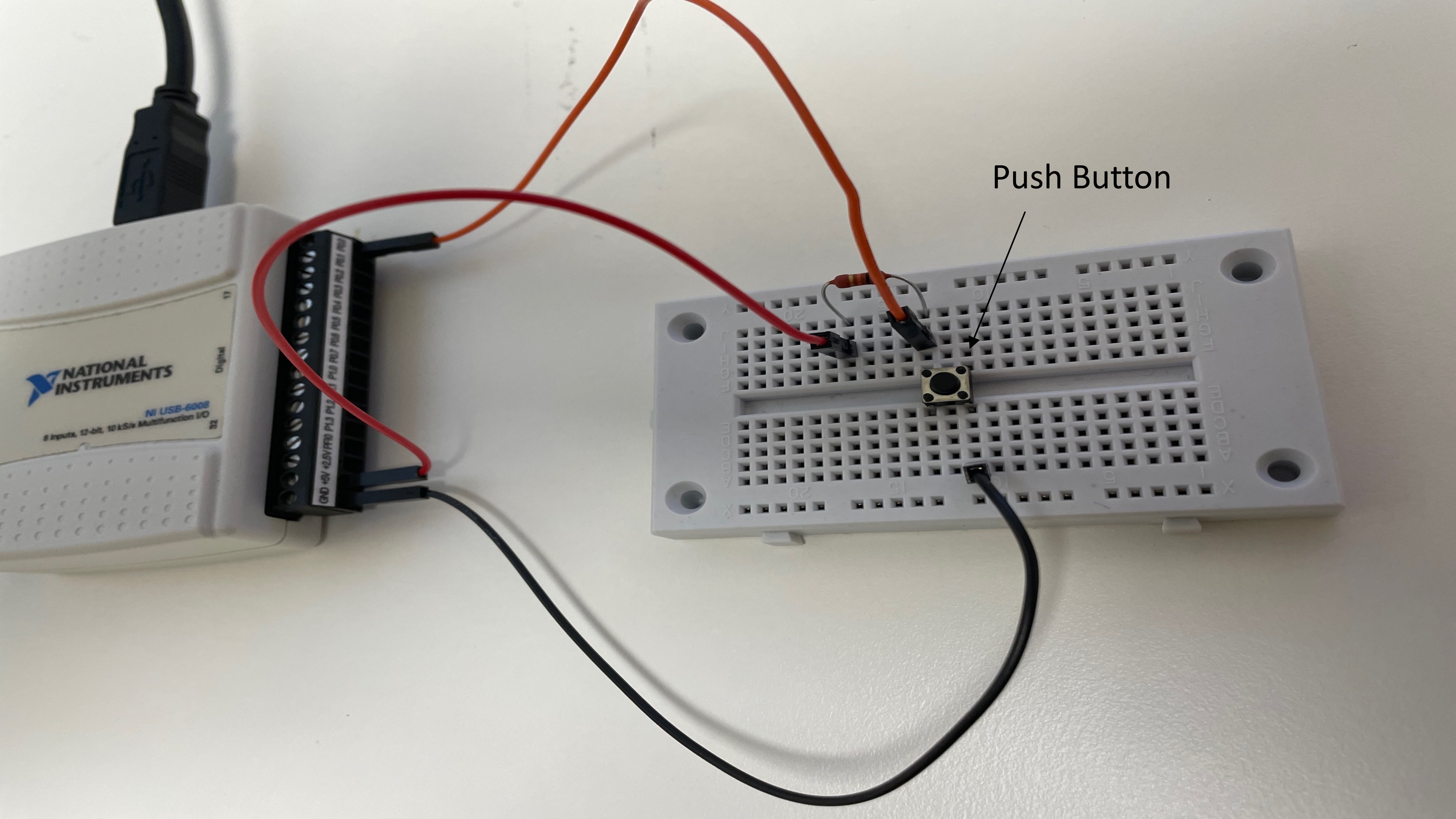


☑ Push Button Status

This Digital In Example shows how we can use a Push Button to set a Digital In to be False/Low (0V) or True/High (5V)

Wiring Example



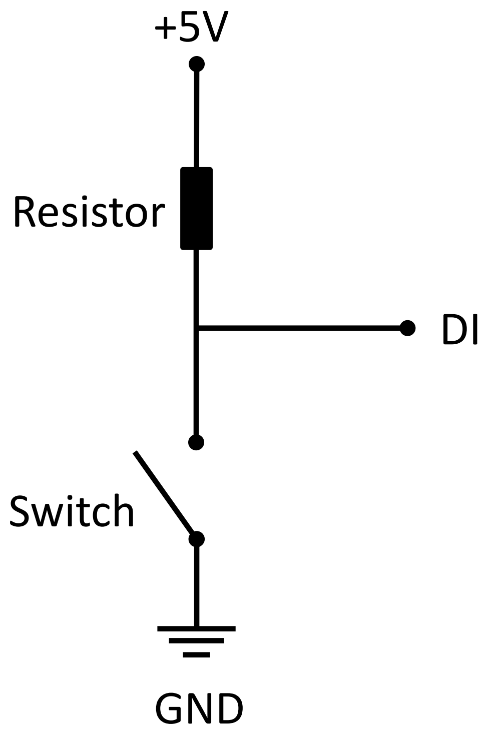

Push Button
NATIONAL INSTRUMENTS
NI USB-6008
8 inputs, 12-bit, 10 kS/s Multifunction I/O
Digital
GND +5V +2.5V PFI0 P1.3 P1.2 P1.1 P1.0 P0.7 P0.6 P0.5 P0.4 P0.3 P0.2 P0.1 P0.0

Pull-down/Pull-up Resistor

Why do we need a pull-up or pull-down resistor in the circuit?

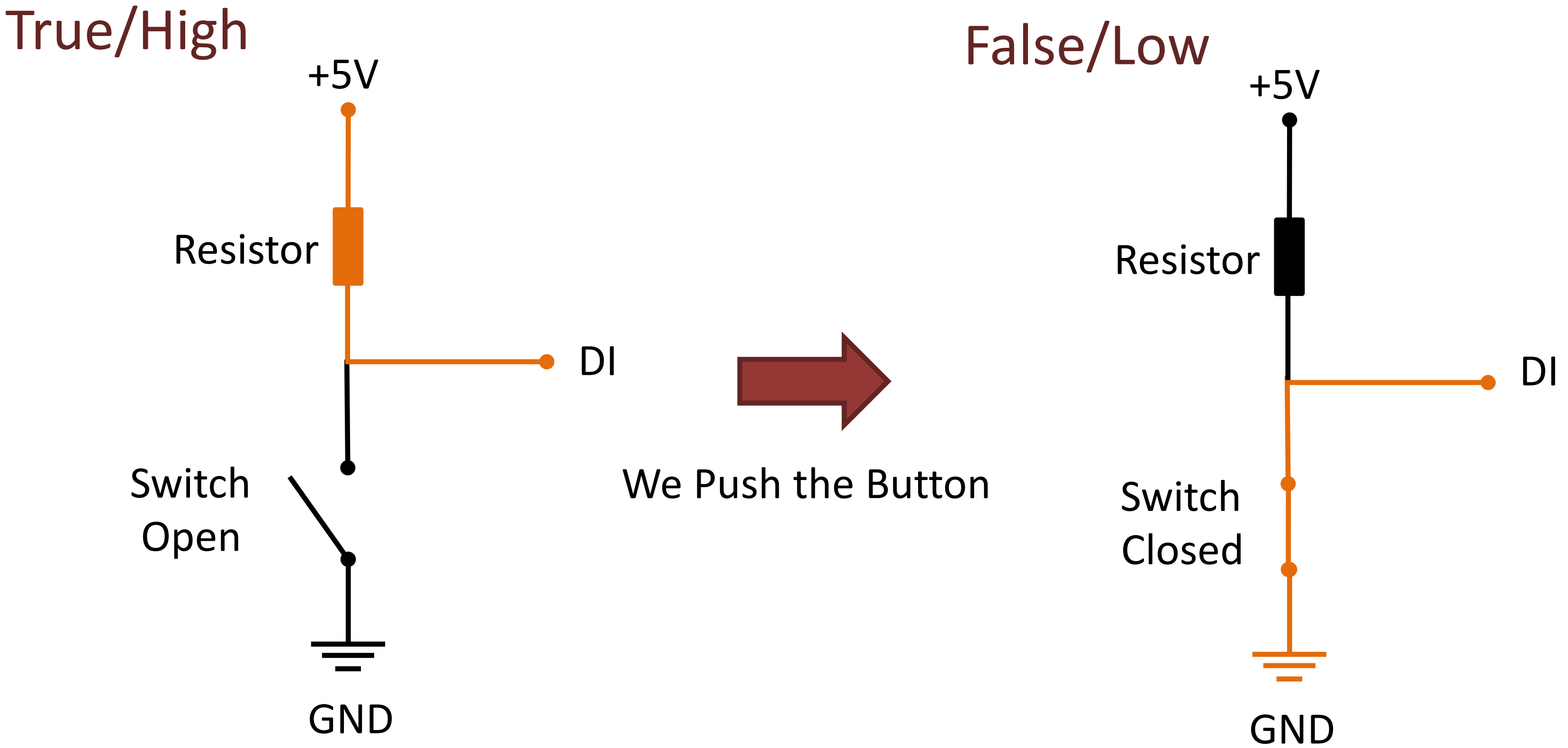
- If you disconnect the digital I/O pin from everything, it will behave in an irregular way.
- This is because the input is "floating" - that is, it will randomly return either HIGH or LOW.
- That's why you need a pull-up or pull-down resistor in the circuit.

Pull-up Resistor



- When the pushbutton is open (unpressed) there is a connection between 5V and the DI pin.
- This means the default state is **True** (High).
- When the button is closed (pressed), the state goes to **False** (Low).

Pull-up Resistor

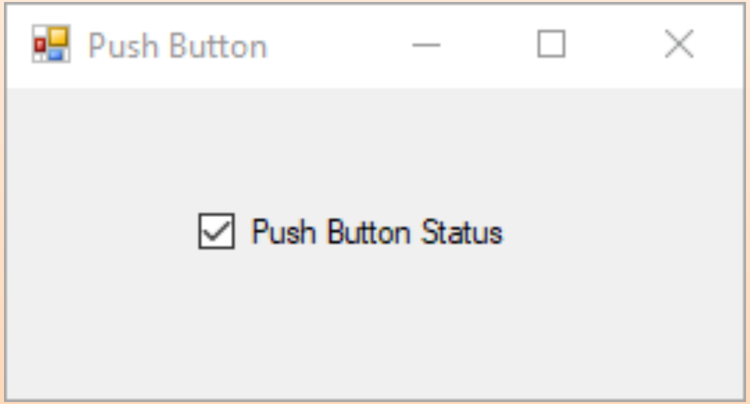


True/High

We Push the Button

False/Low

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

namespace PushButton
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
            timer1.Interval = 100;
            timer1.Start();
        }

        private void timer1_Tick(object sender, EventArgs e)
        {
            checkBox1.Checked = ReadPushButton();
        }

        bool ReadPushButton()
        {
            Task digitalInTask = new Task();

            digitalInTask.DIChannels.CreateChannel("dev1/Port0/line0",
                                    "myDIChannel",
                                    ChannelLineGrouping.OneChannelForEachLine);

            DigitalSingleChannelReader reader = new
                                        DigitalSingleChannelReader(digitalInTask.Stream);

            bool pushButton = reader.ReadSingleSampleSingleLine();

            return !pushButton;
        }
    }
}
```



☑ Push Button Status

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog