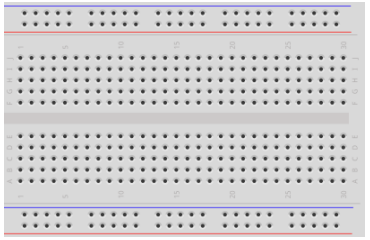
https://www.halvorsen.blog

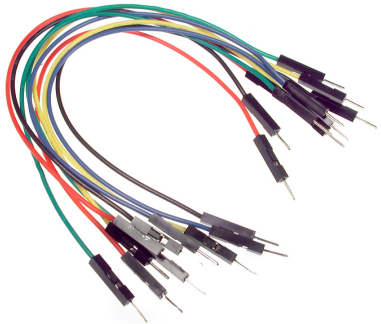
TMP36 Temperature Sensor

Hans-Petter Halvorsen

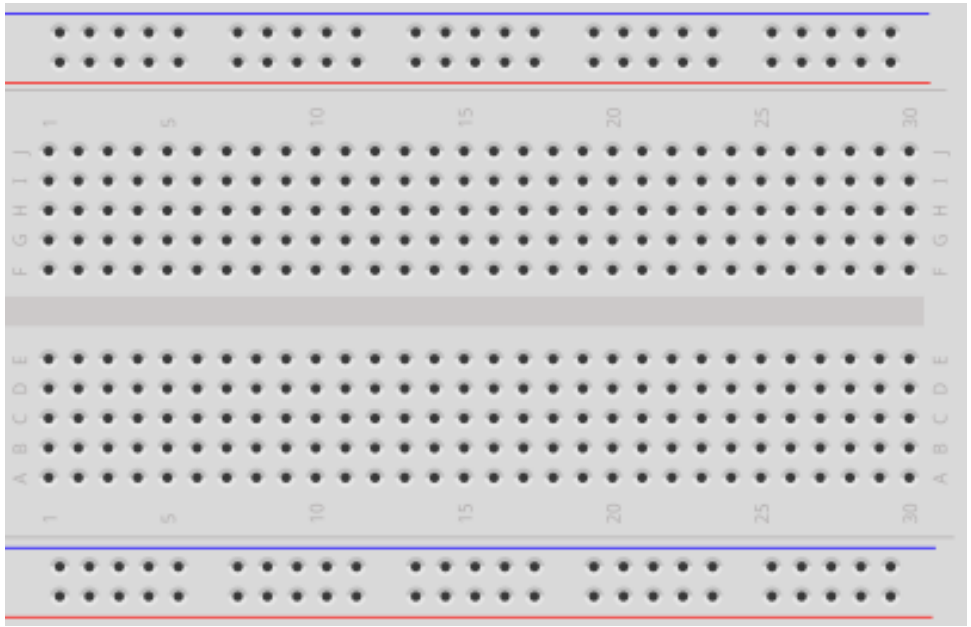
Hardware

- DAQ Device (e.g., USB-6008)
- Breadboard
- TMP36 Temperature Sensor
- Wires (Jumper Wires)

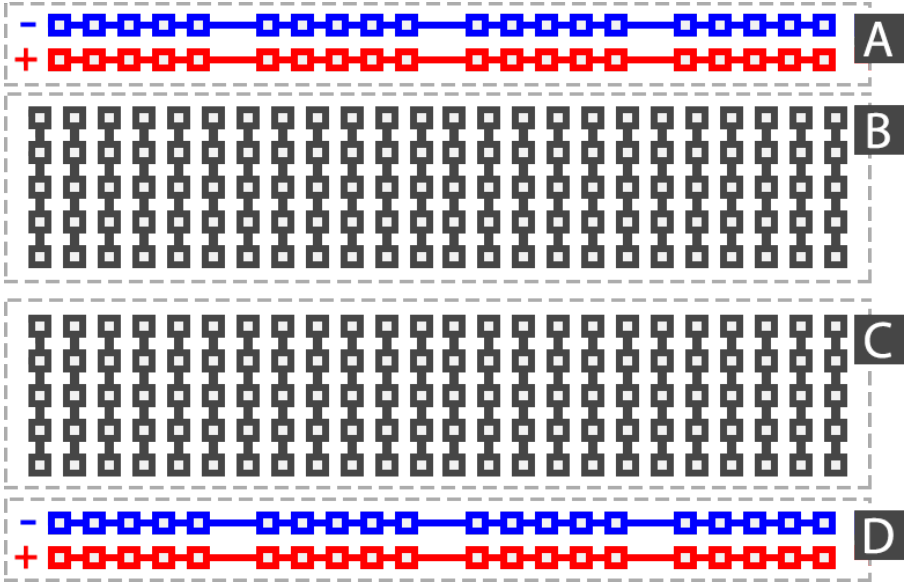




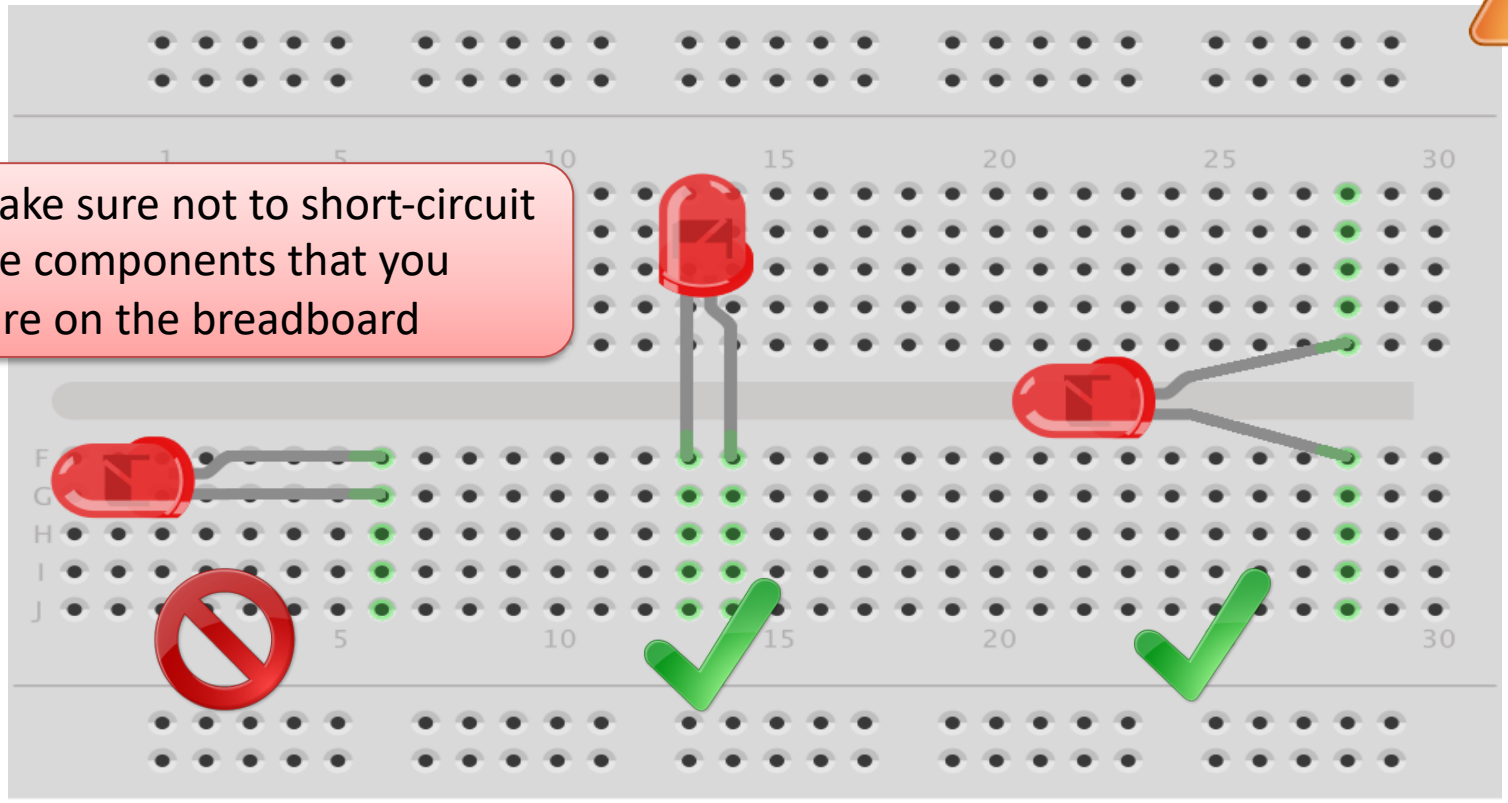
Breadboard



A breadboard is used to wire electric components together



Breadboard Wiring



The Breadboard is used to connect components and electrical circuits

fritzing

https://www.halvorsen.blog

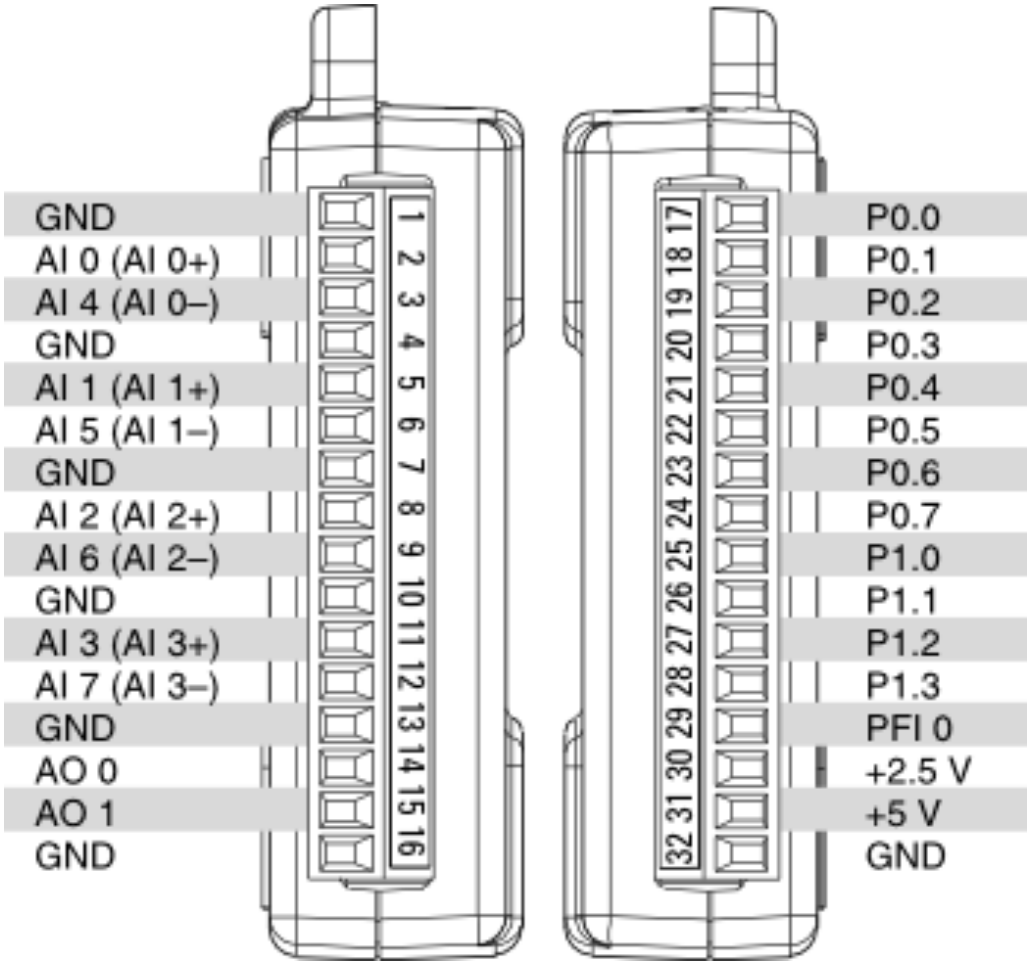
USB-6008

Hans-Petter Halvorsen

USB-6008


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

I/O Pins

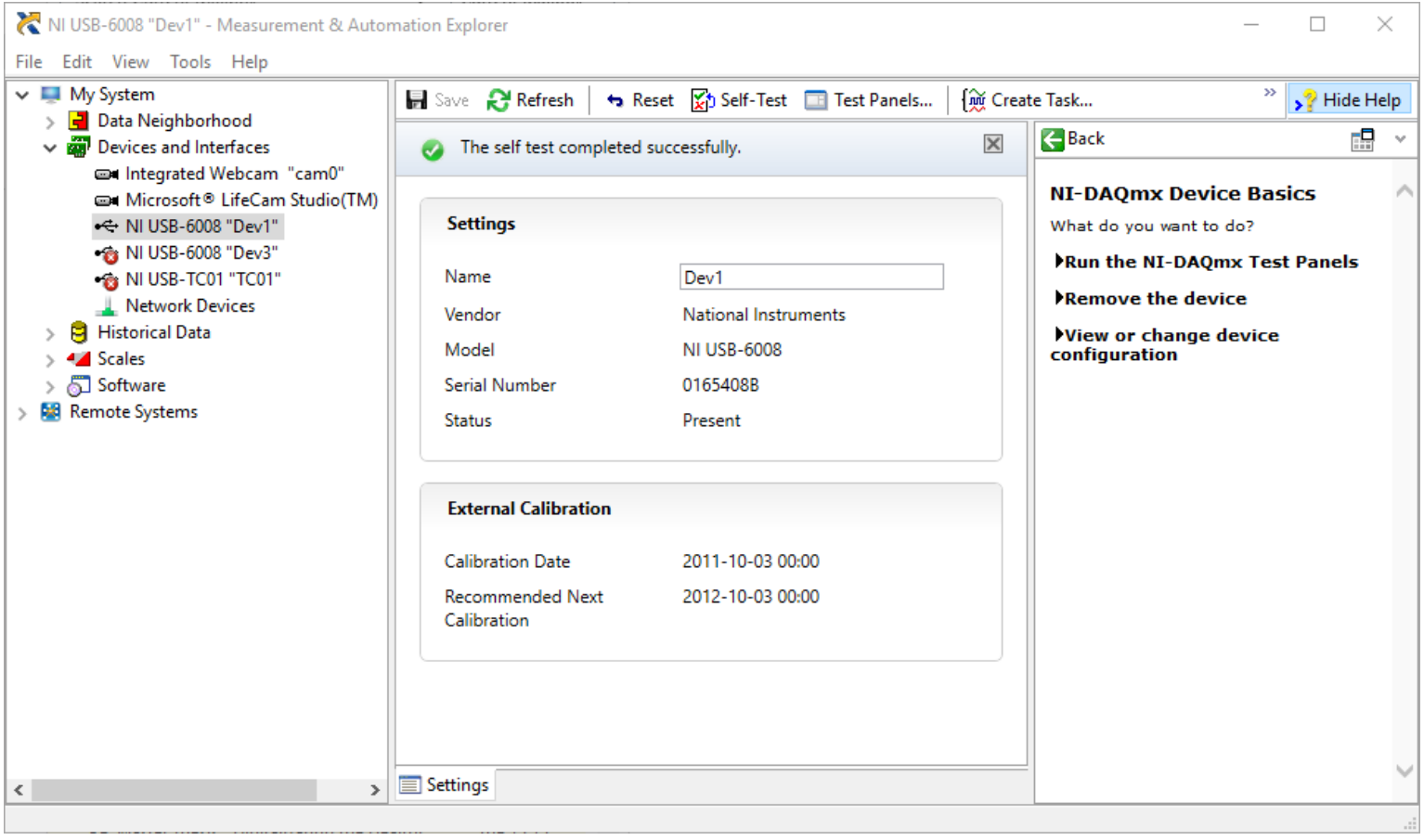

GND
AI 0 (AI 0+)
AI 4 (AI 0–)
GND
AI 1 (AI 1+)
AI 5 (AI 1–)
GND
AI 2 (AI 2+)
AI 6 (AI 2–)
GND
AI 3 (AI 3+)
AI 7 (AI 3–)
GND
AO 0
AO 1
GND
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32
P0.0
P0.1
P0.2
P0.3
P0.4
P0.5
P0.6
P0.7
P1.0
P1.1
P1.2
P1.3
PFI 0
+2.5 V
+5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

https://www.halvorsen.blog

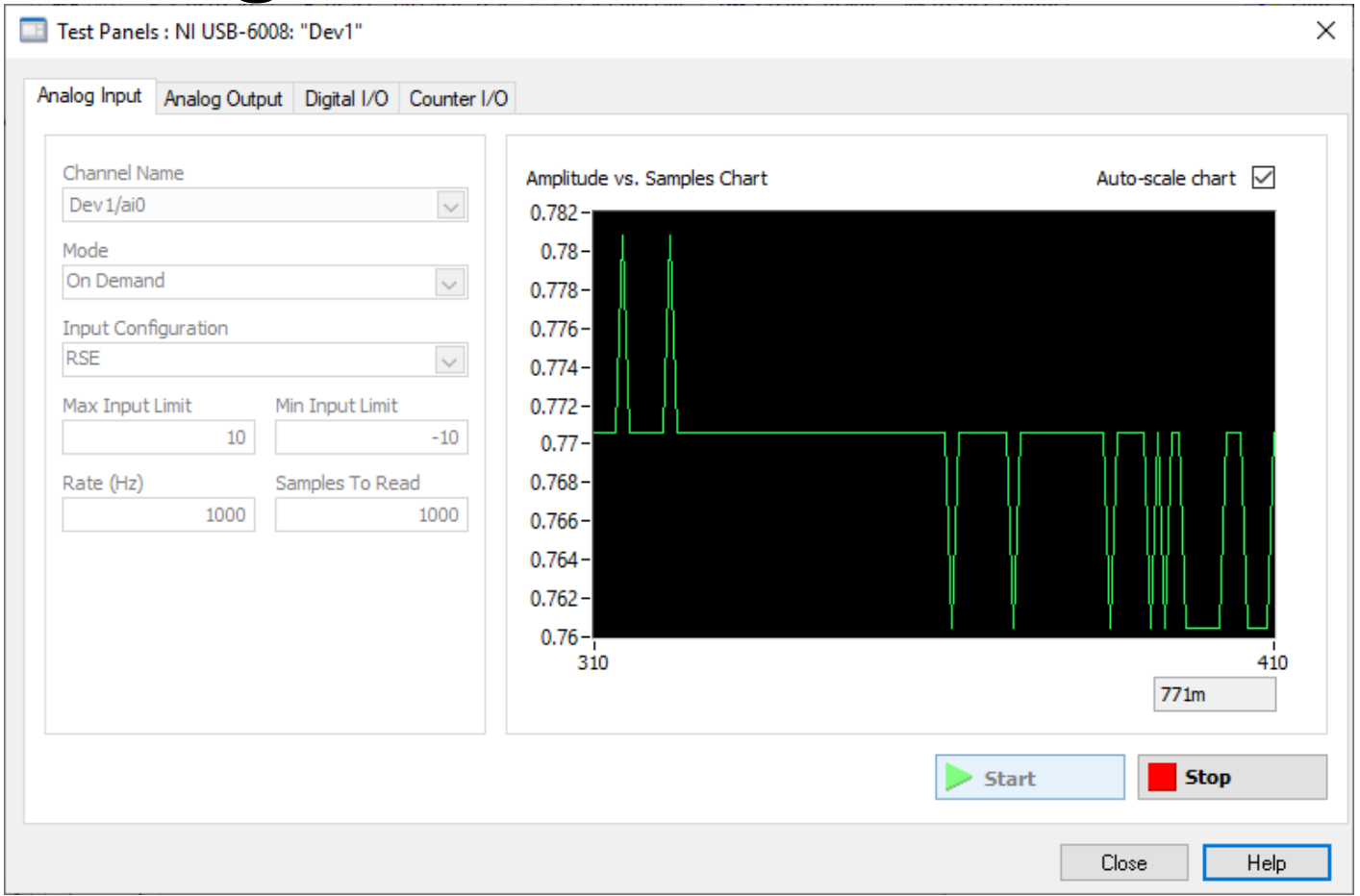
DAQmx

Hans-Petter Halvorsen

Measurement & Automation Explorer (MAX)



Using the Test Panel in MAX



https://www.halvorsen.blog

TMP36

Hans-Petter Halvorsen

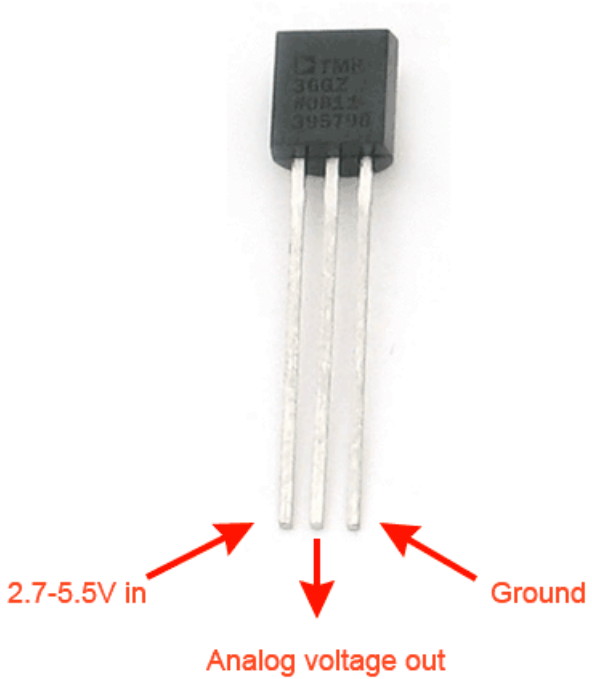
TMP36

TMP is a small, low-cost temperature sensor and cost about $1 (you can buy it “everywhere”)

TMP36

Temperature measurement range	-40...+125 °C
Accuracy	±2 °C
Power supply	2.3...5.5 V
Package	TO-92
Temperature sensitivity, voltage	10 mV/°C

TMP36 Temperature Sensor

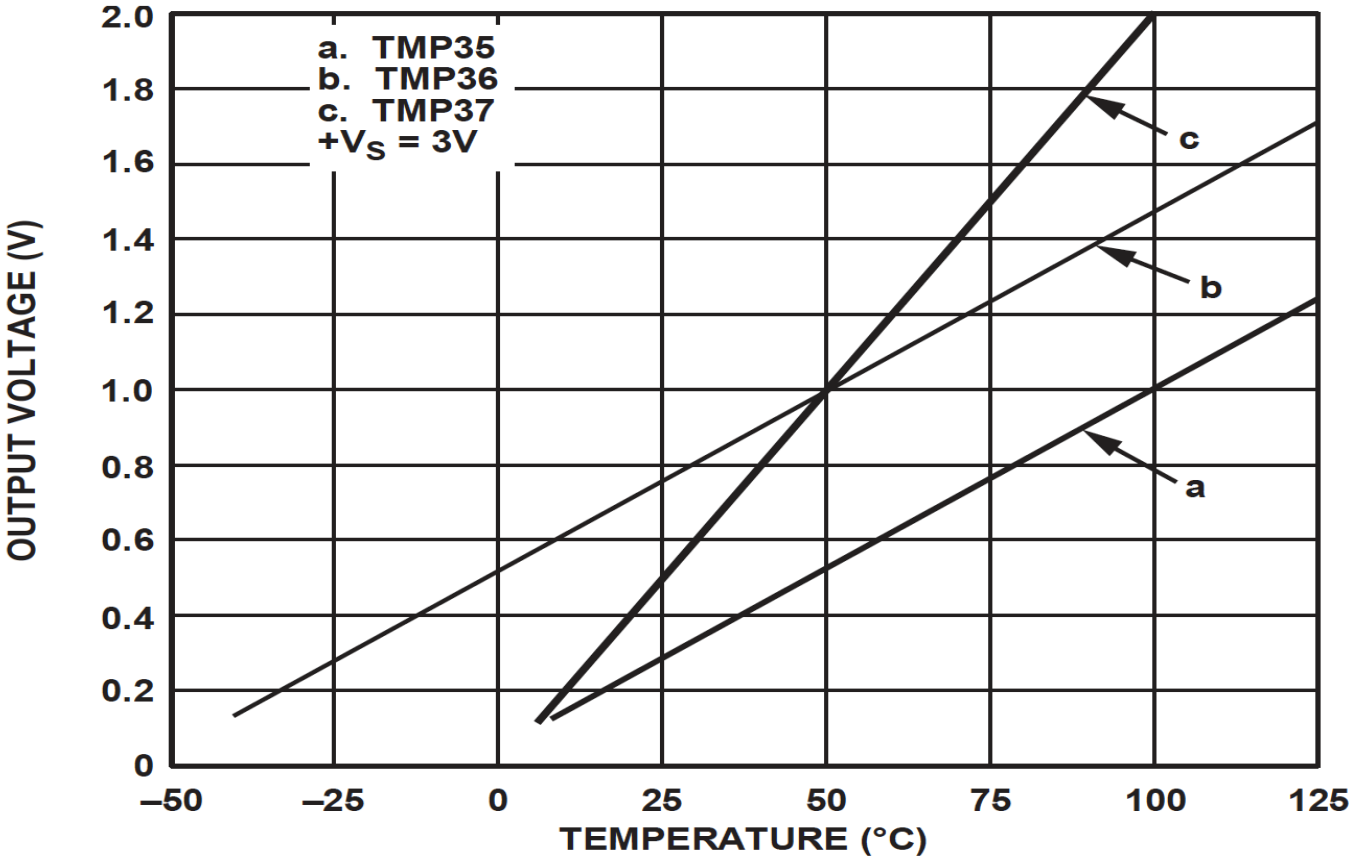


A Temperature sensor like TM36 use a solid-state technique to determine the temperature.

They use the fact as temperature increases, the voltage across a diode increases at a known rate.

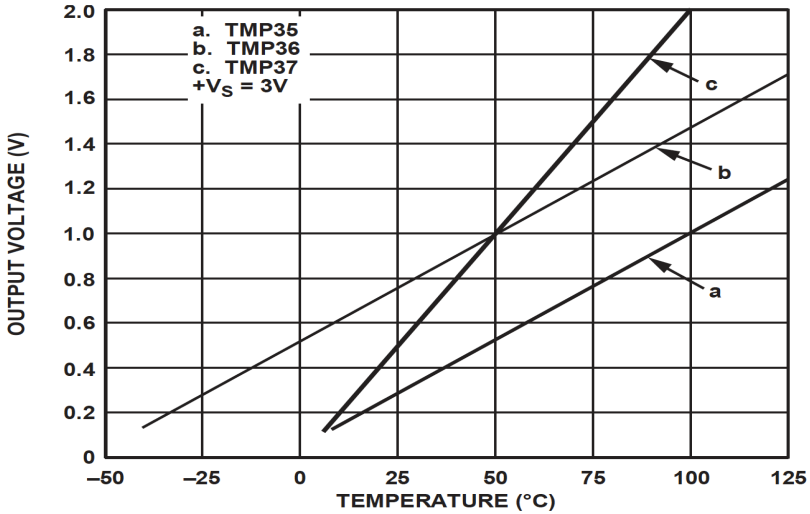
https://learn.adafruit.com/tmp36-temperature-sensor

TMP36 Datasheet



http://no.rs-online.com/webdocs/14cd/0900766b814cd0a1.pdf

Linear Scaling



This gives:

$$y - 25 = \frac{50 - 25}{1 - 0.75}(x - 0.75)$$

Then we get the following formula:

$$y = 100x - 50$$

Convert form Voltage (V) to degrees Celsius

From the Datasheet we have:

$$(x_1, y_1) = (0.75V, 25°C)$$
$$(x_2, y_2) = (1V, 50°C)$$

There is a linear relationship between Voltage and degrees Celsius:

$$y = ax + b$$

We can find a and b using the following known formula:

$$y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$$

Celsius to Fahrenheit Conversion

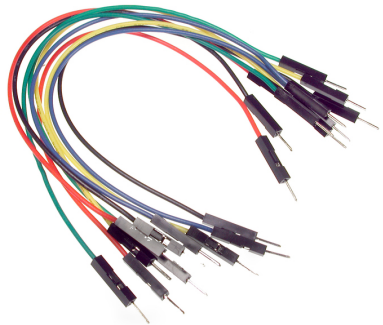
In Norway we typically use Celsius as temperature unit, while in US they use Fahrenheit.

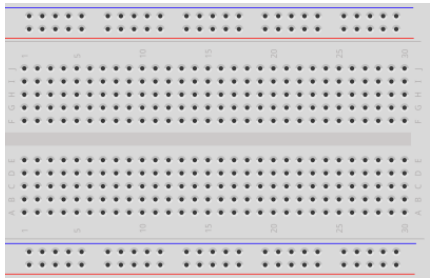
Conversion between these are as follows:

$$T_F = \frac{9}{5} T_C + 32$$

Necessary Equipment

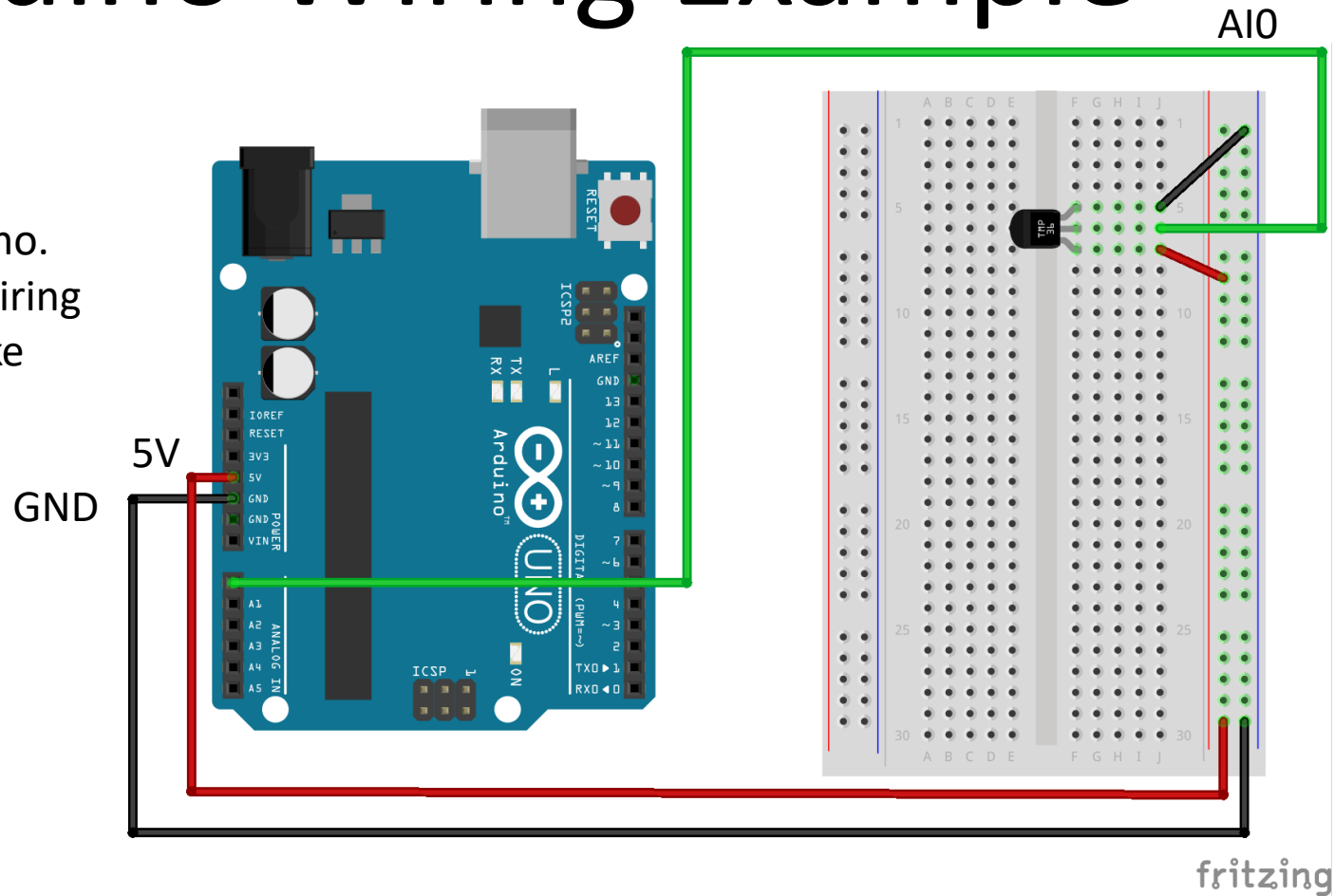
- PC
- DAQ Module, e.g., USB-6008
- Breadboard
- TMP36
- Wires (Jumper Wires)



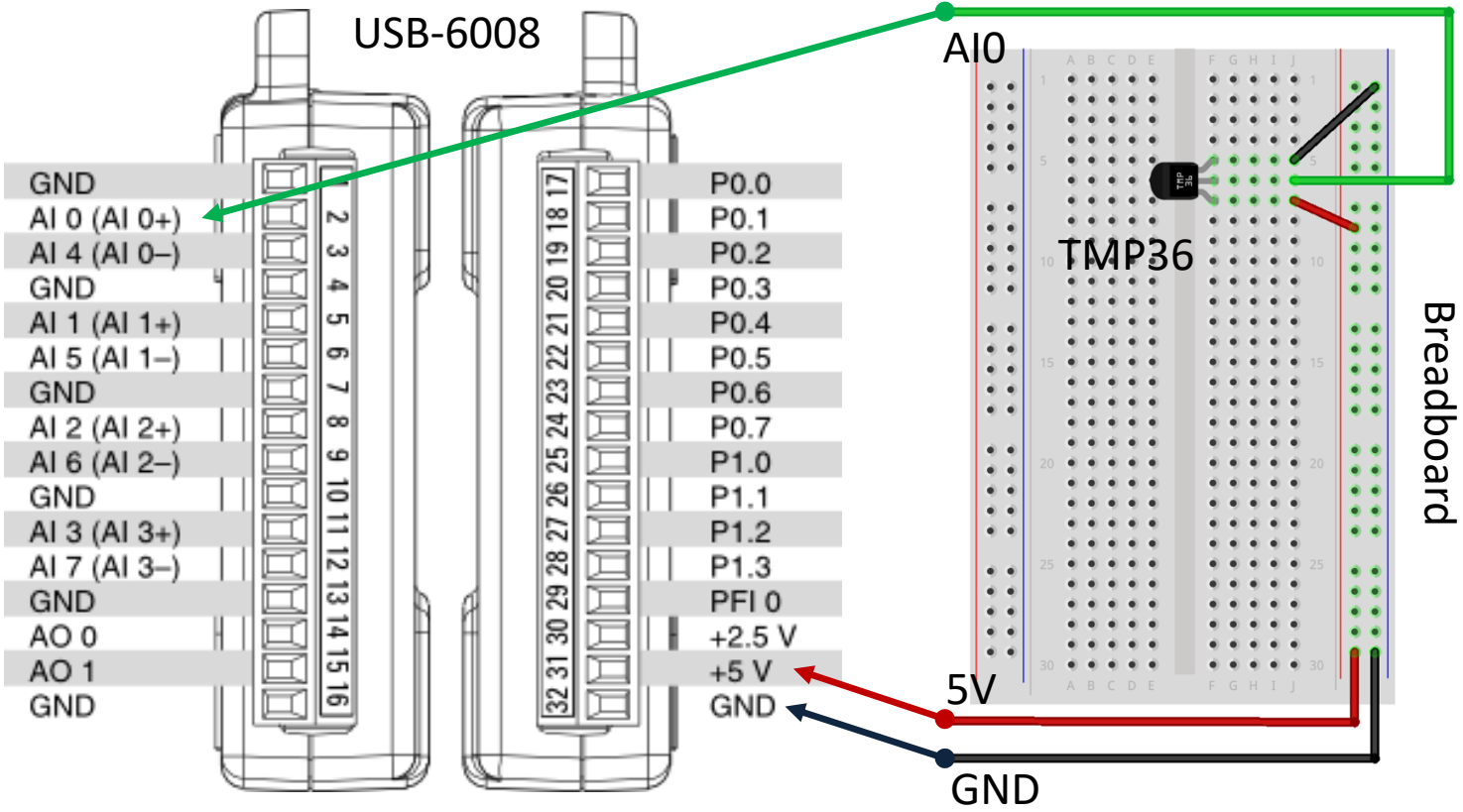


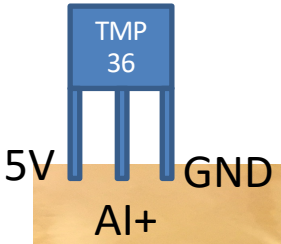
Arduino Wiring Example

Here you see a wiring examples using Arduino. You make the same wiring using a DAQ device like USB-6008 or similar.



USB-6008 Wiring Example





USB-6008 Wiring Example

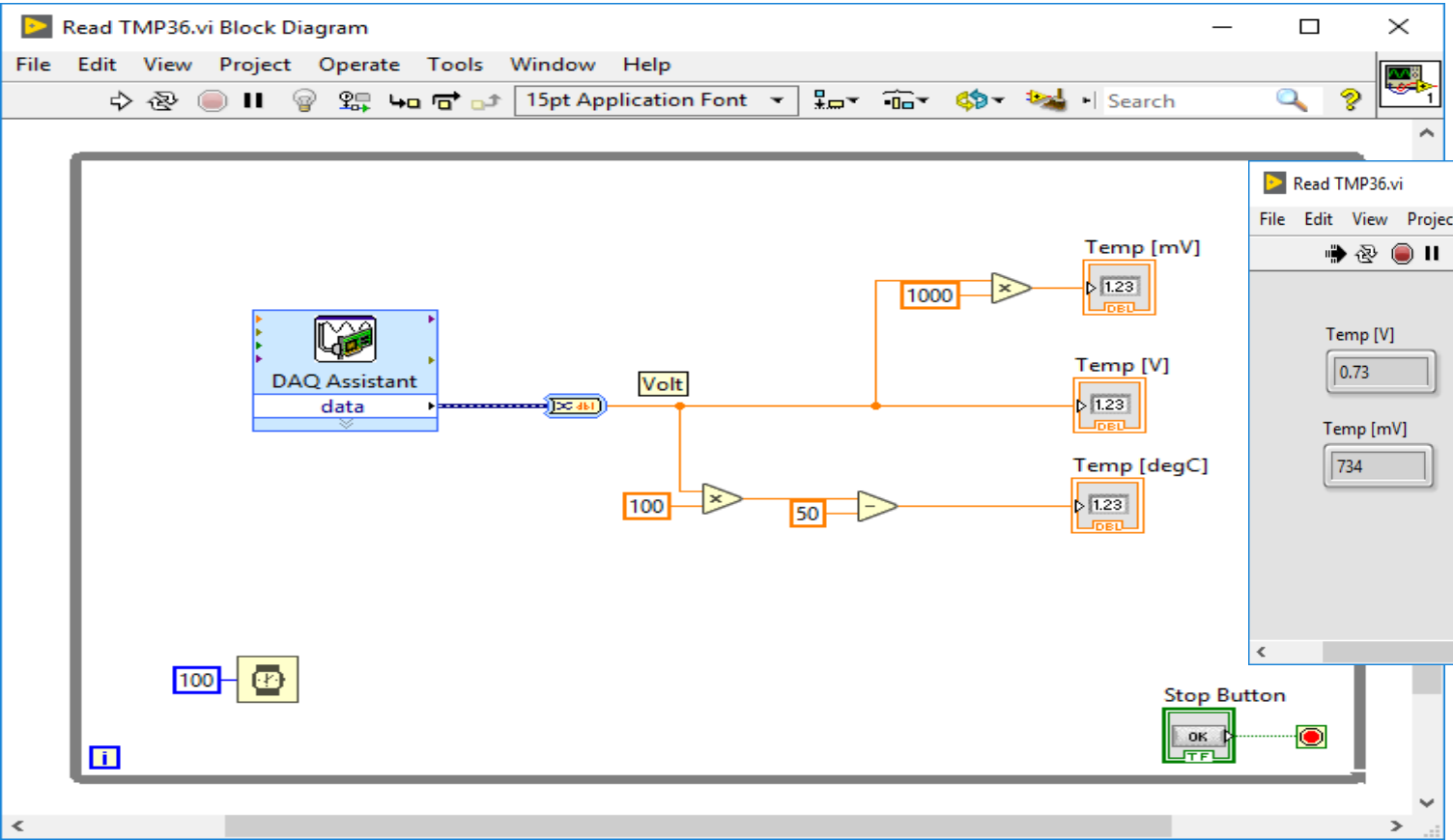
We connect the TMP36 to LabVIEW using a USB DAQ Device from National Instruments, e.g., USB-6001, USB-6008 or similar. I have used a breadboard for the wiring.

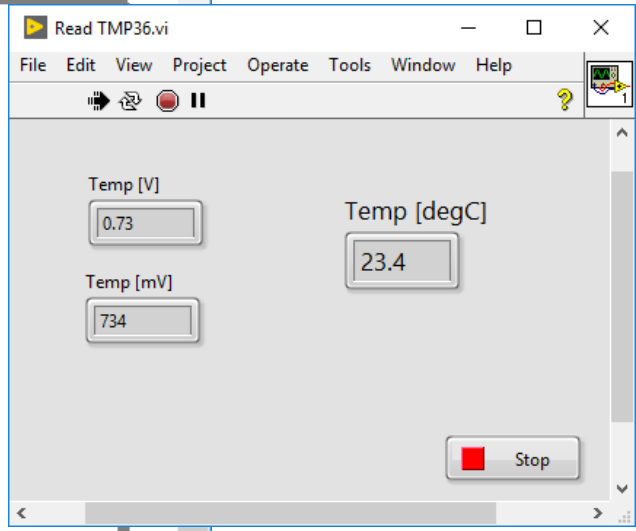
Pseudo Code

A typical scenario:

1. Read Signal from DAQ Device (0-5V)
2. Convert to degrees Celsius using information from the Datasheet
3. Show/Plot Values in your Application GUI
4. Save Data to a Database

LabVIEW Example





Arduino Example

```
const int temperaturePin = 0;

float adcValue;
float voltage;
float degreesC;

void setup()
{
  Serial.begin(9600);
}

void loop()
{

  adcValue = analogRead(temperaturePin);

  voltage = (adcValue*5)/1023;

  degreesC = 100*voltage - 50;

  Serial.print("ADC Value: ");
  Serial.print(adcValue);

  Serial.print("  voltage: ");
  Serial.print(voltage);

  Serial.print("  deg C: ");
  Serial.println(degreesC);

  delay(1000);
}
```

Convert from ADC-value (0-1023) to Voltage (0-5V)

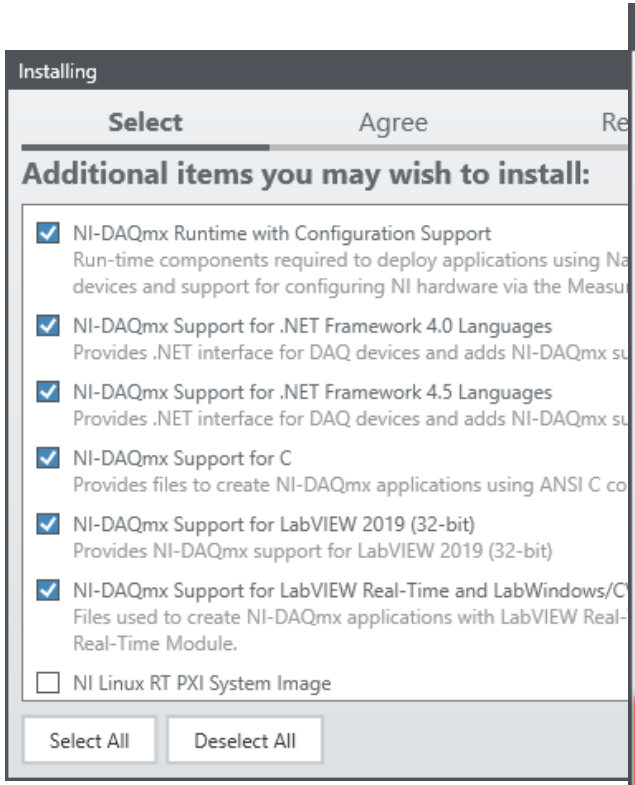
Convert from Voltage to degrees Celsius

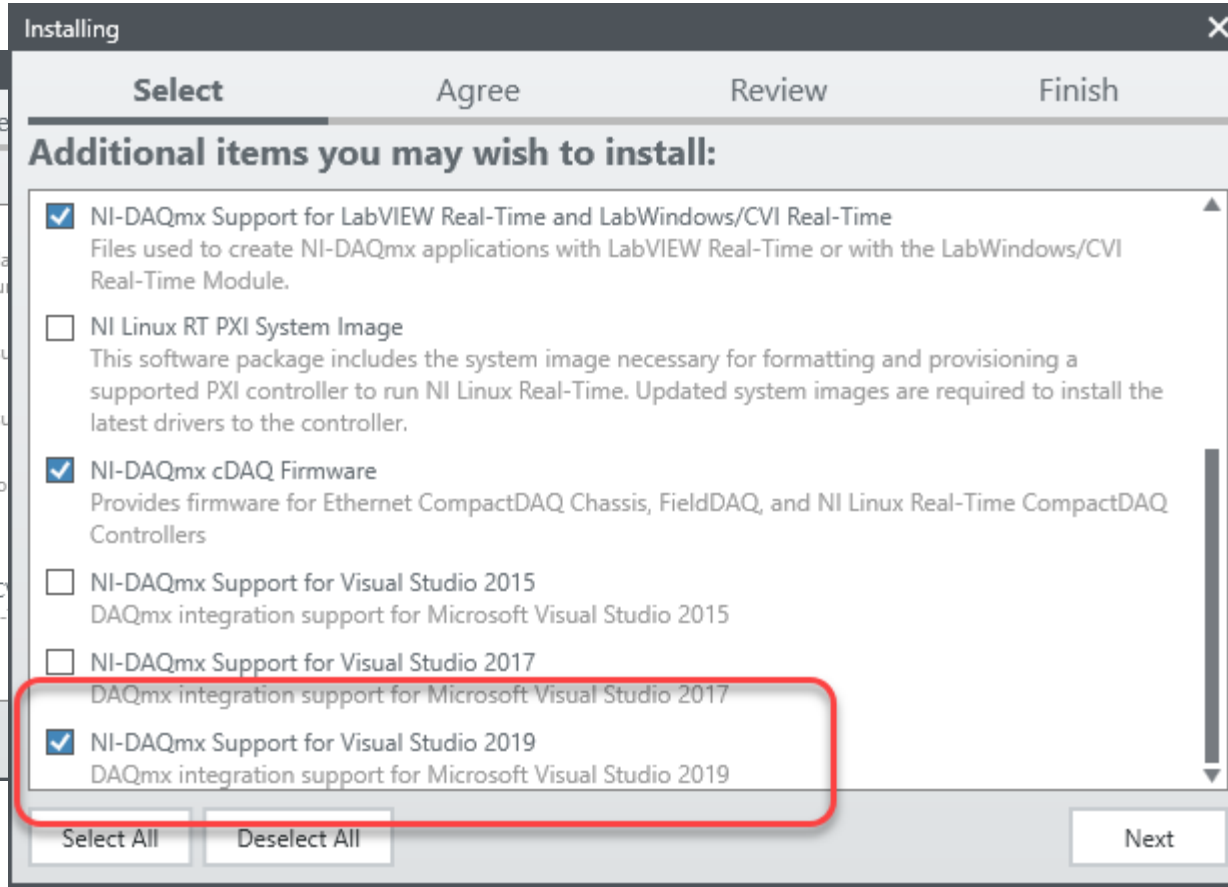
https://www.halvorsen.blog

Visual Studio

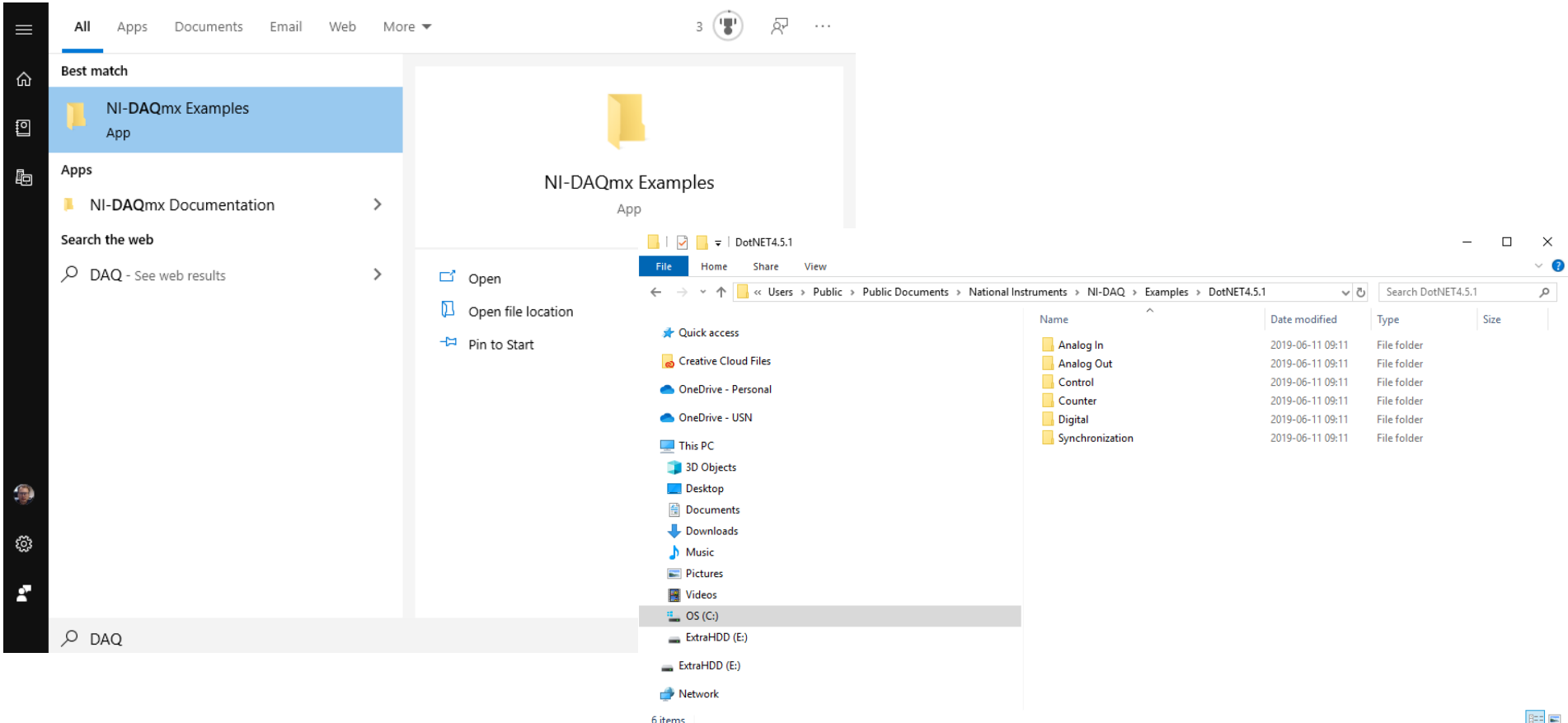
Hans-Petter Halvorsen

NI-DAQmx Driver



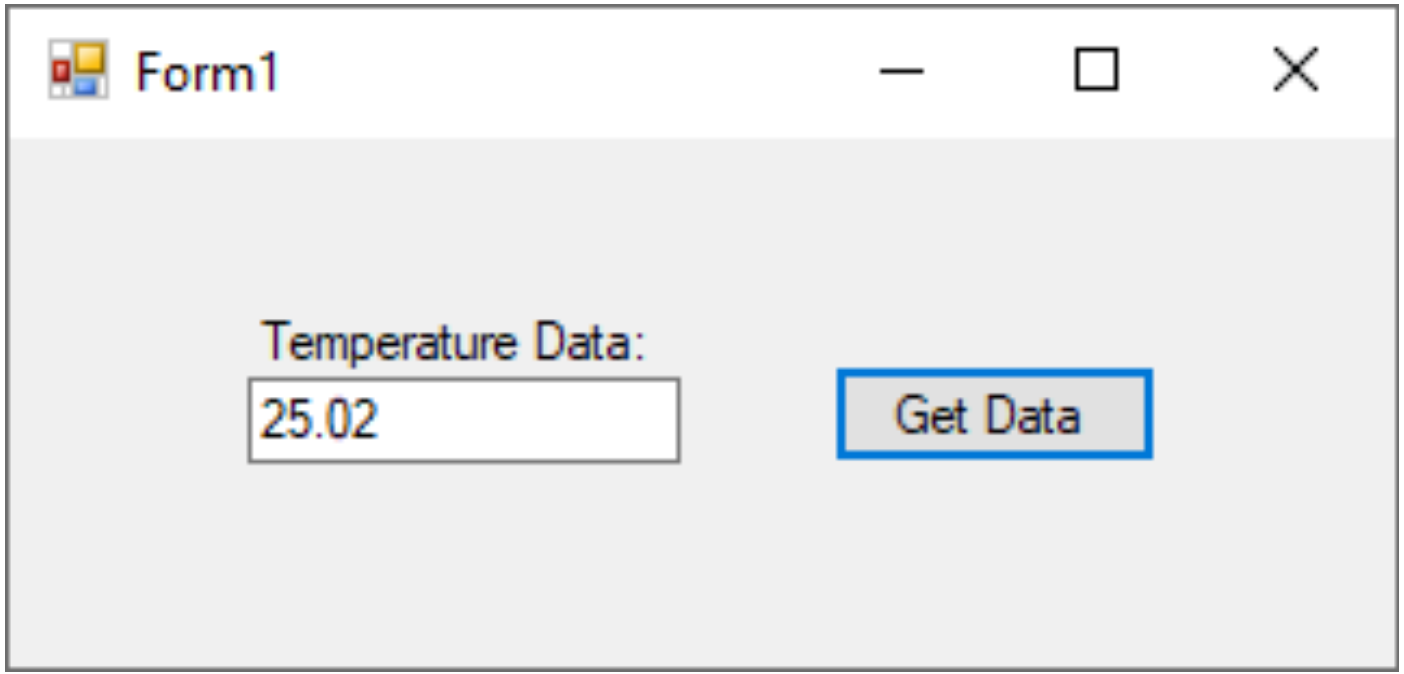


NI-DAQmx Examples



Application

We will create the following Application in Visual Studio:



Create a new project

Search for templates (Alt+S)

Clear all

C# | Windows | Desktop

Recent project templates

- Windows Forms App (.NET Framework) — C#
- ASP.NET Core Web Application — C#
- ASP.NET Web Application (.NET Framework) — C#
- ASP.NET Web Application (.NET Framework) — Visual Basic
- Windows Forms App (.NET Core) — C#
- Python Application — Python

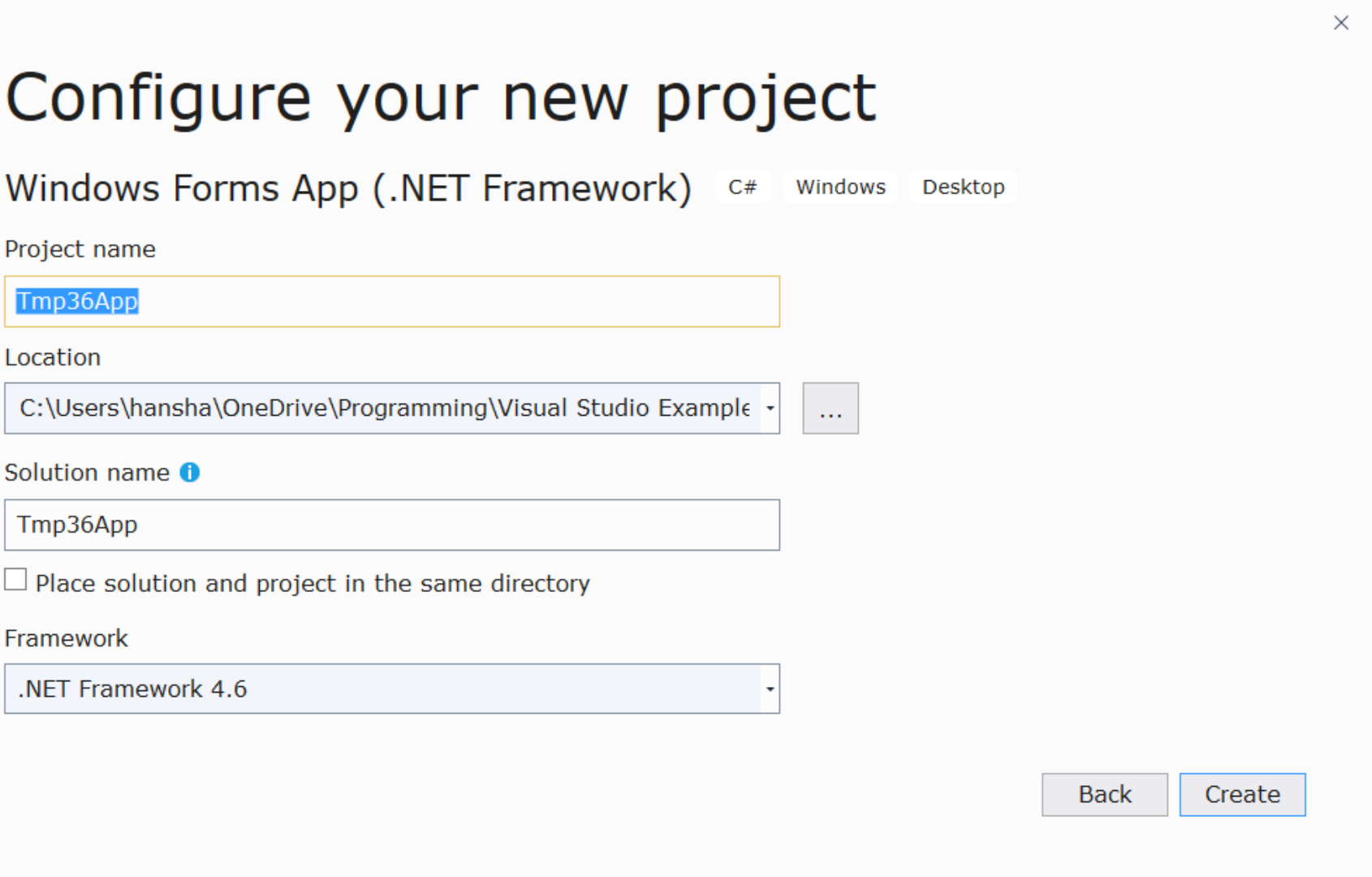
NUnit Test Project (.NET Core)
A project that contains NUnit tests that can run on .NET Core on Windows, Linux and MacOS.
C# Linux macOS Windows Desktop Test Web

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

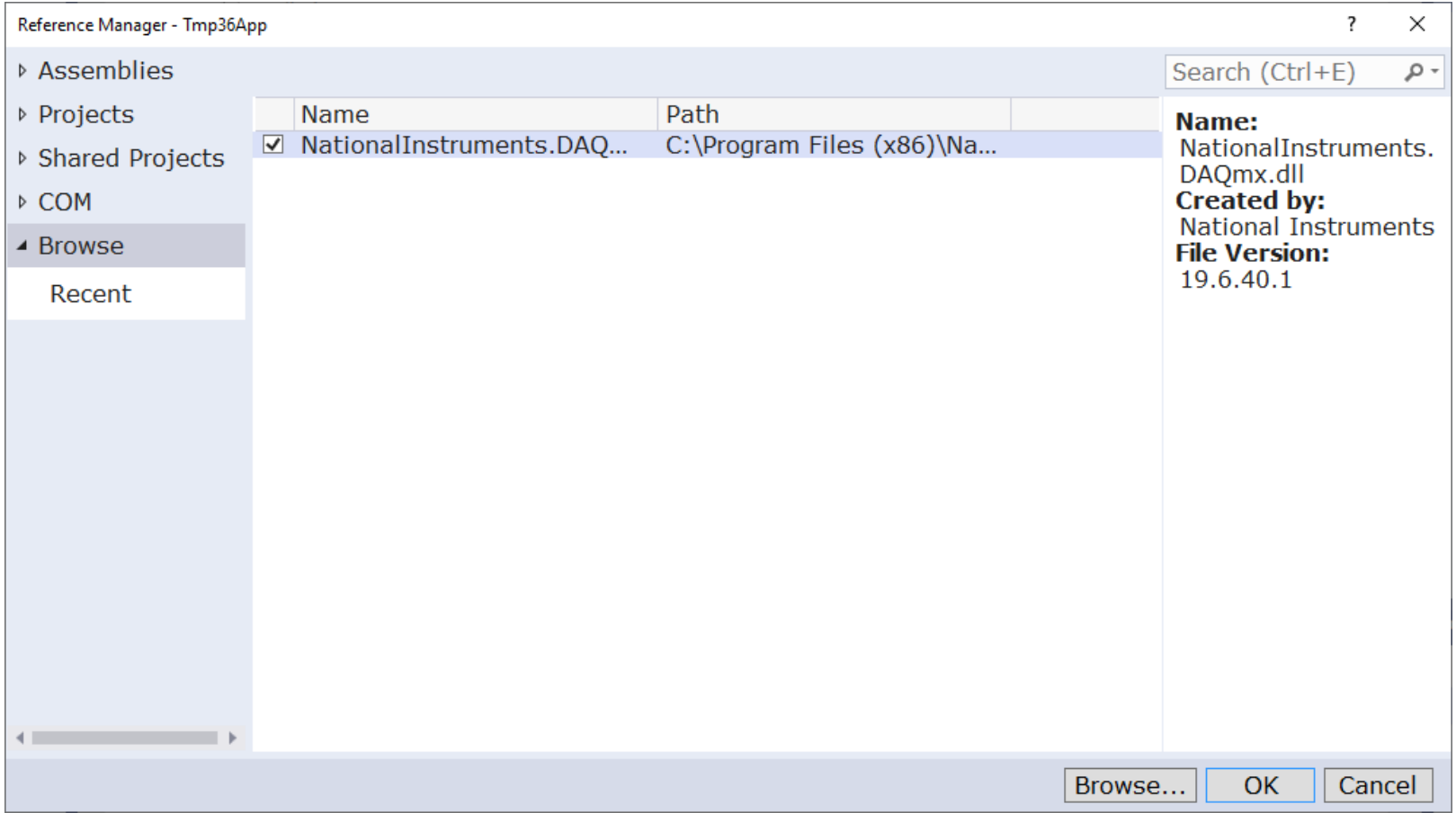
WPF App (.NET Framework)
Windows Presentation Foundation client application
C# Windows Desktop

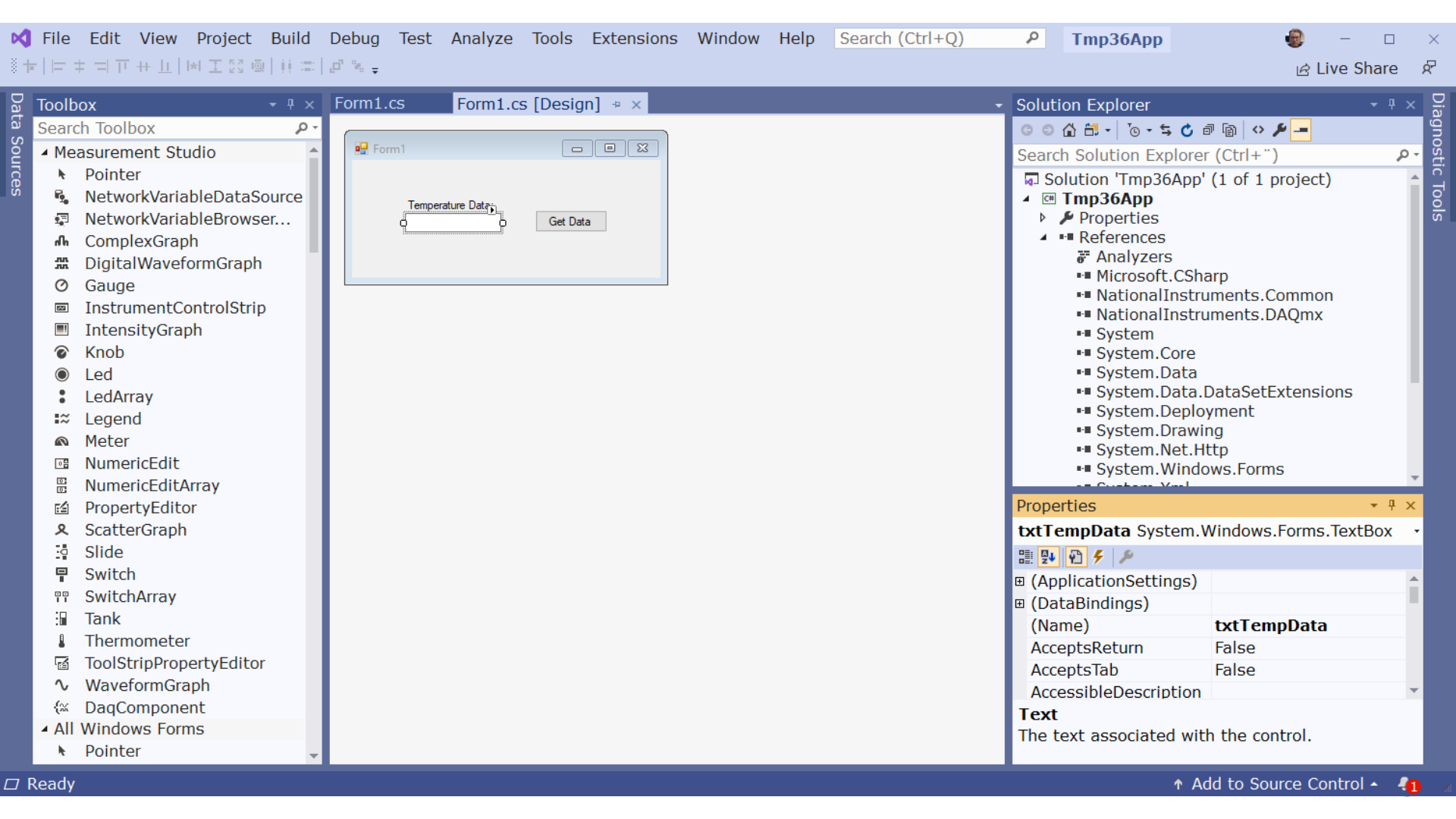
WPF App (.NET Core)

Back Next

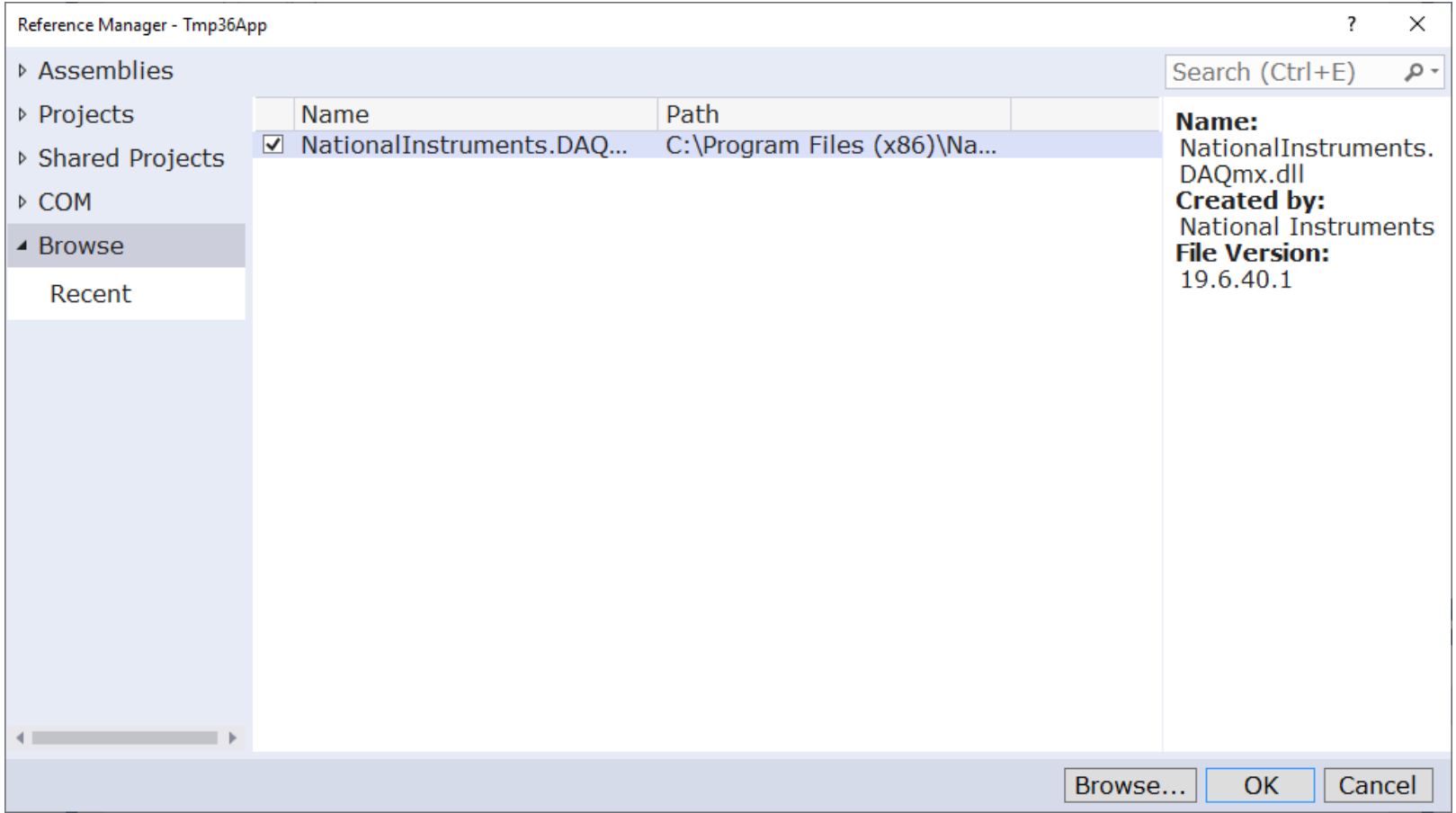

Configure your new project
Windows Forms App (.NET Framework)
C#
Windows
Desktop
Project name
Tmp36App
Location
C:\Users\hansha\OneDrive\Programming\Visual Studio Example
...
Solution name
Tmp36App
Place solution and project in the same directory
Framework
.NET Framework 4.6
Back
Create

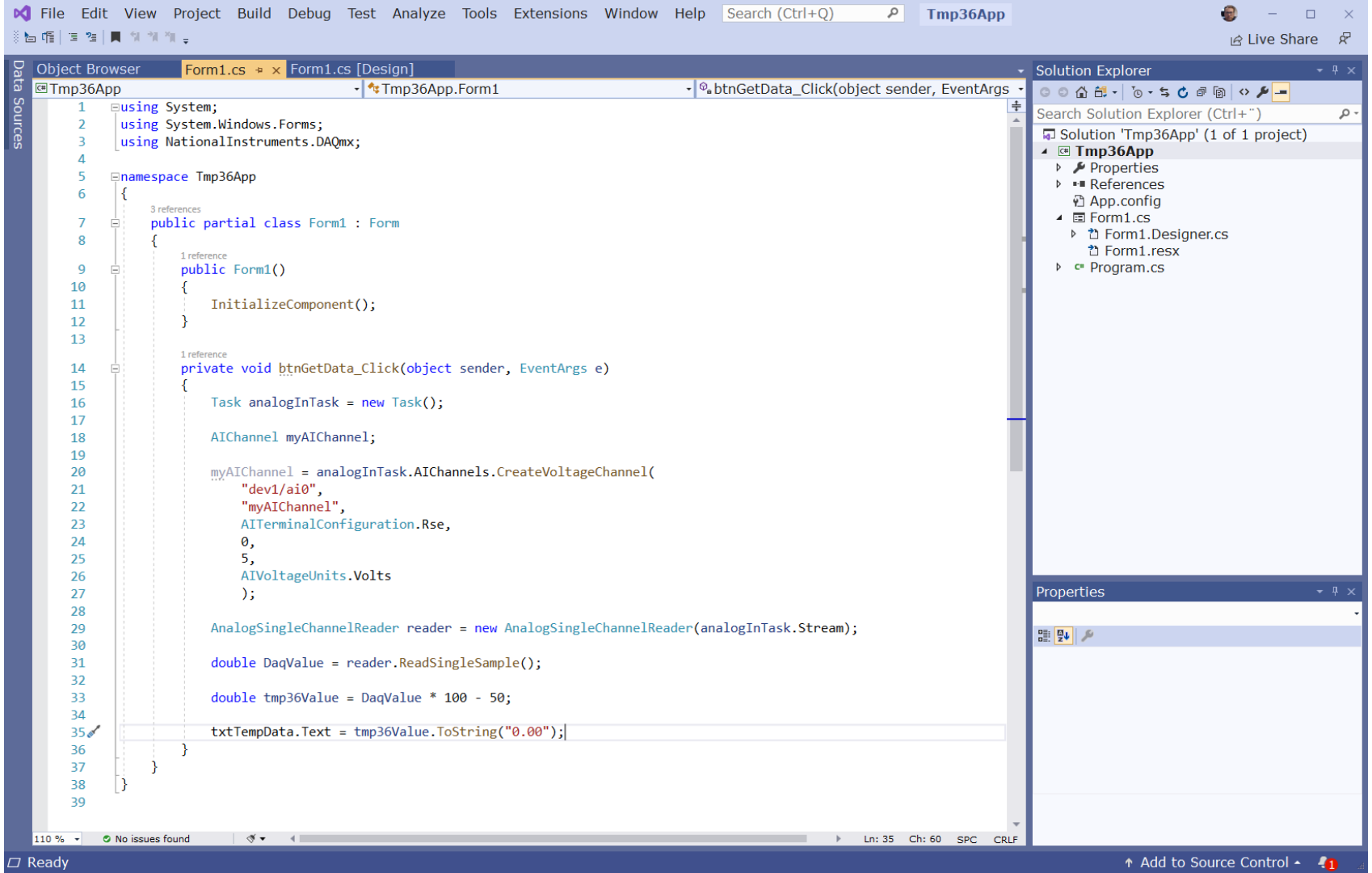
NationalInstruments.DAQmx.dll




File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Tmp36App
Live Share
Data Sources
Toolbox
Search Toolbox
Measurement Studio
Pointer
NetworkVariableDataSource
NetworkVariableBrowser...
ComplexGraph
DigitalWaveformGraph
Gauge
InstrumentControlStrip
IntensityGraph
Knob
Led
LedArray
Legend
Meter
NumericEdit
NumericEditArray
PropertyEditor
ScatterGraph
Slide
Switch
SwitchArray
Tank
Thermometer
ToolStripPropertyEditor
WaveformGraph
DaqComponent
All Windows Forms
Pointer
Form1.cs
Form1.cs [Design]
Form1
Temperature Data
Get Data
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Tmp36App' (1 of 1 project)
Tmp36App
Properties
References
Analyzers
Microsoft.CSharp
NationalInstruments.Common
NationalInstruments.DAQmx
System
System.Core
System.Data
System.Data.DataSetExtensions
System.Deployment
System.Drawing
System.Net.Http
System.Windows.Forms
Properties
txtTempData System.Windows.Forms.TextBox
(ApplicationSettings)
(DataBindings)
(Name) txtTempData
AcceptsReturn False
AcceptsTab False
AccessibleDescription
Text
The text associated with the control.
Diagnostic Tools
Ready
Add to Source Control

NationalInstruments.DAQmx.dll

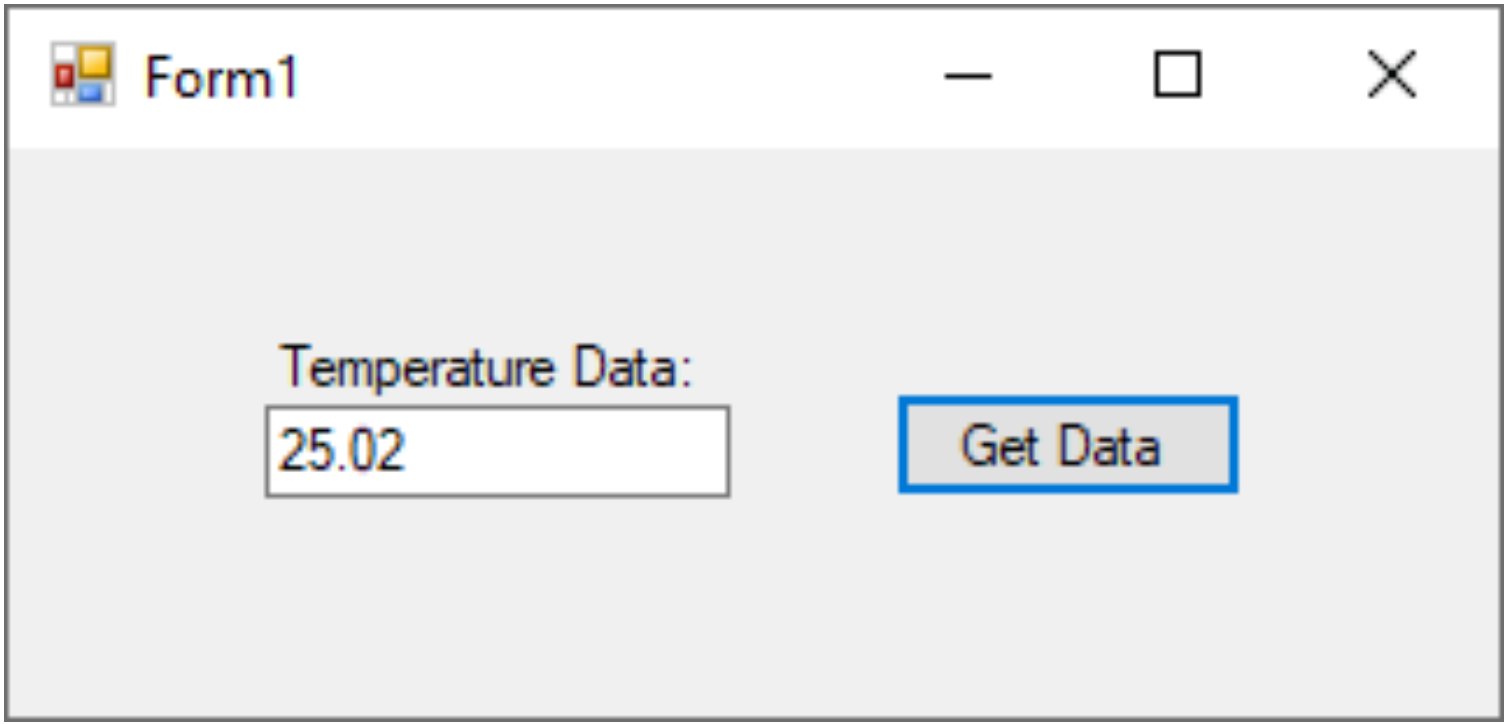



File Edit View Project Build Debug Test Analyze Tools Extensions Window Help
Search (Ctrl+Q)
Tmp36App
Live Share
Data Sources
Object Browser
Form1.cs
Form1.cs [Design]
Tmp36App
Tmp36App.Form1
btnGetData_Click(object sender, EventArgs
using System;
using System.Windows.Forms;
using NationalInstruments.DAQmx;
namespace Tmp36App
{
3 references
public partial class Form1 : Form
{
1 reference
public Form1()
{
InitializeComponent();
}
1 reference
private void btnGetData_Click(object sender, EventArgs e)
{
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 DaqValue = reader.ReadSingleSample();
double tmp36Value = DaqValue * 100 - 50;
txtTempData.Text = tmp36Value.ToString("0.00");
}
}
}
110 %
No issues found
Ln: 35 Ch: 60 SPC CRLF
Solution Explorer
Search Solution Explorer (Ctrl+;)
Solution 'Tmp36App' (1 of 1 project)
Tmp36App
Properties
References
App.config
Form1.cs
Form1.Designer.cs
Form1.resx
Program.cs
Properties
Ready
Add to Source Control

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

namespace Tmp36App
{
   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.Rse,
            0,
            5,
            AIVoltageUnits.Volts
            );

         AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);

         double DaqValue = reader.ReadSingleSample();

         double tmp36Value = DaqValue * 100 - 50;

         txtTempData.Text = tmp36Value.ToString("0.00");
      }
   }
}
```

Final Application

Now we are ready to Run (F5) our Application:



Improvements

- Create and use separate **Classes** and in general improve the C# code
- Use a **Timer** in order to read values at specific intervals
- Plot values in a **Chart**
- Save Data to a **Database**
- Save Data to a **Text File**
- etc.

Good luck with your Application

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog