

<https://www.halvorsen.blog>

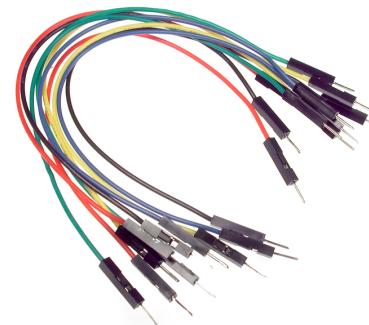
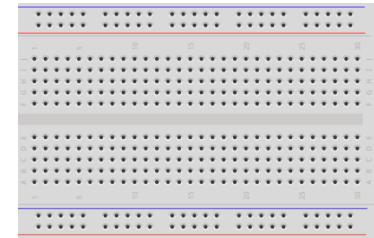


TMP36 Temperature Sensor

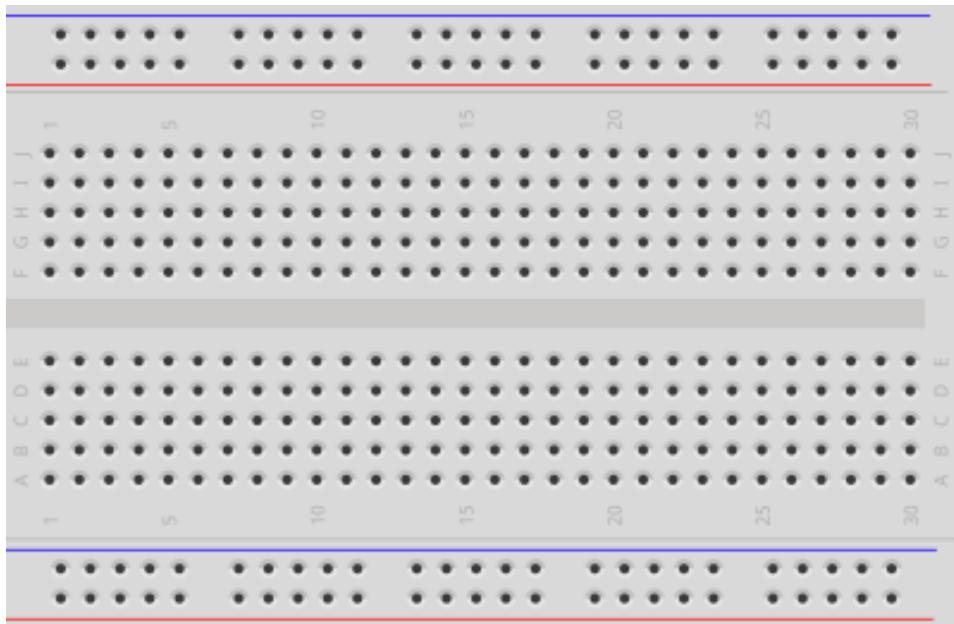
Hans-Petter Halvorsen

Hardware

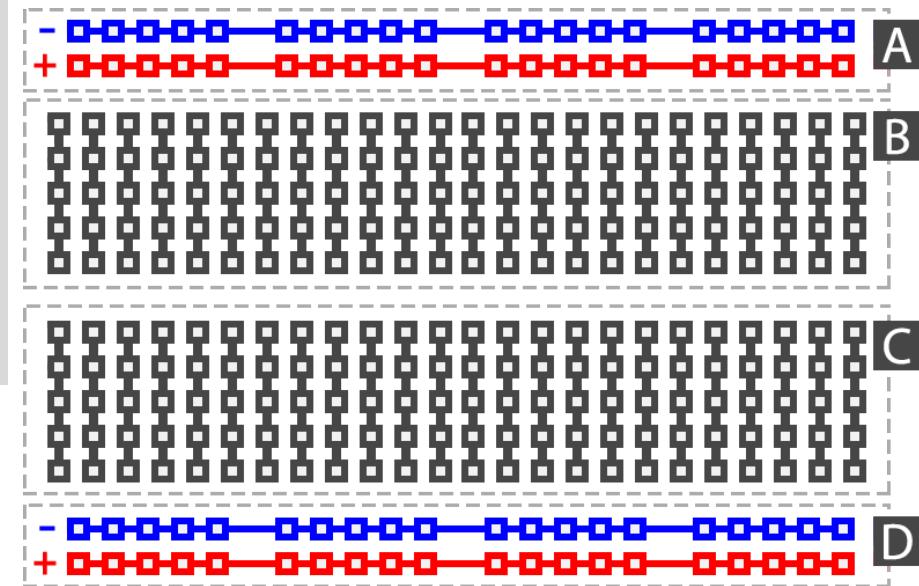
- DAQ Device (e.g., USB-6008)
- Breadboard
- TMP36 Temperature Sensor
- Wires (Jumper Wires)



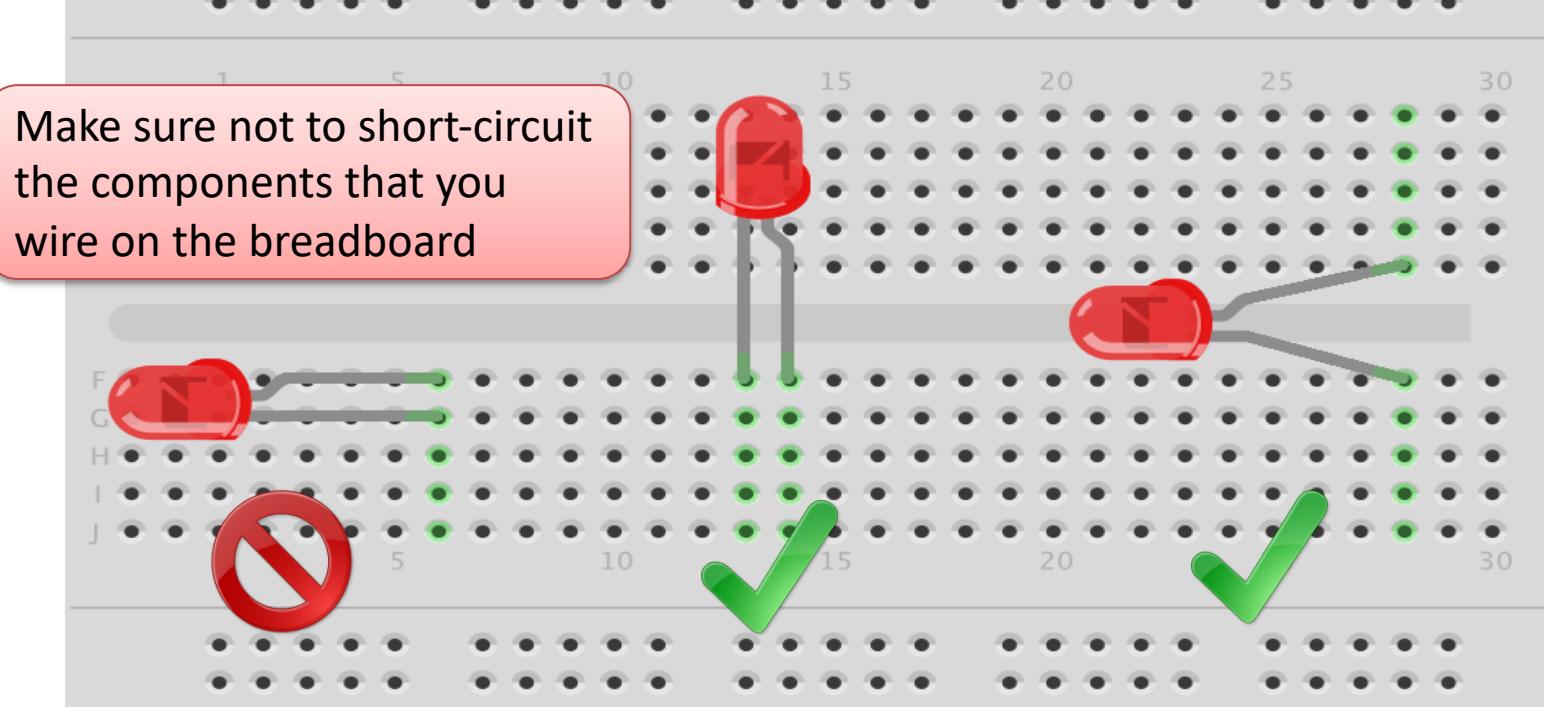
Breadboard



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>



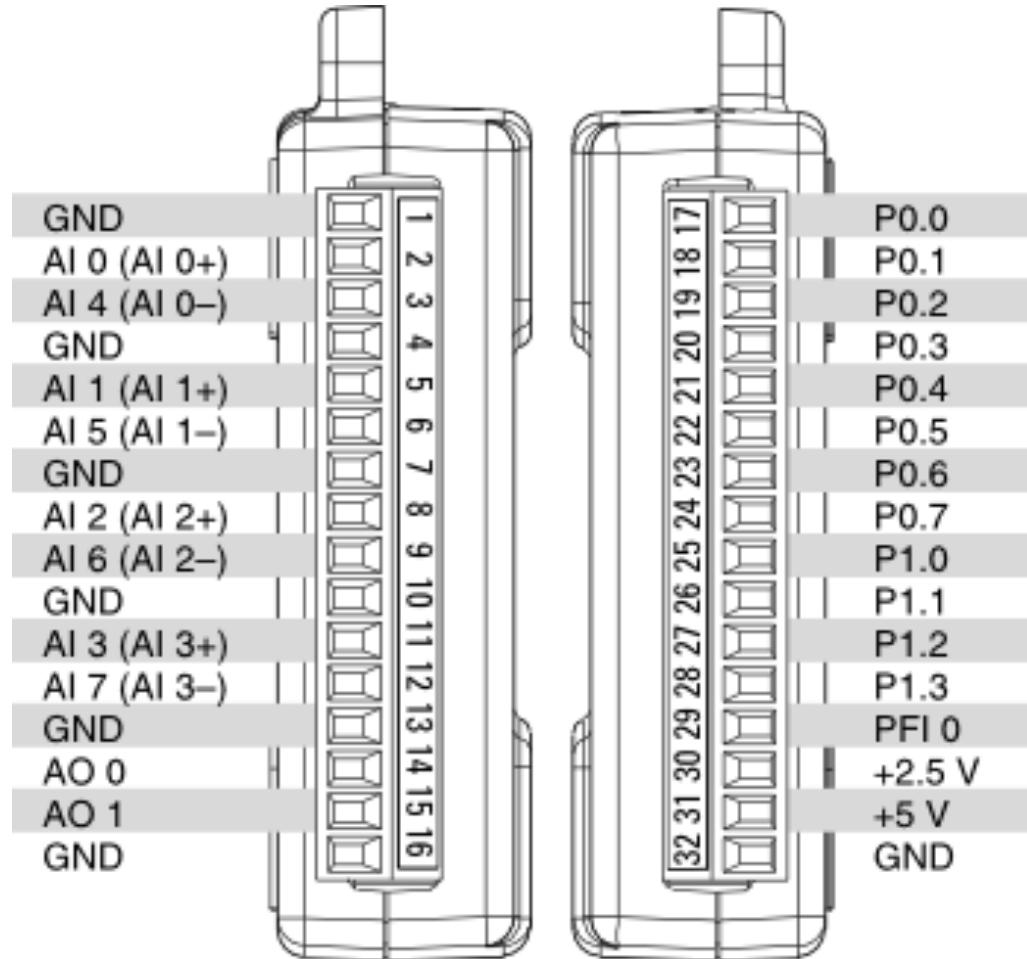
USB-6008

Hans-Petter Halvorsen

USB-6008



I/O Pins



<https://www.halvorsen.blog>



DAQmx

Hans-Petter Halvorsen

Measurement & Automation Explorer (MAX)

The screenshot shows the NI Measurement & Automation Explorer (MAX) interface. The left sidebar displays a tree view of system components, including My System, Data Neighborhood, Devices and Interfaces (with entries for Integrated Webcam "cam0", Microsoft® LifeCam Studio(TM), NI USB-6008 "Dev1", NI USB-6008 "Dev3", NI USB-TC01 "TC01", Network Devices, Historical Data, Scales, Software, and Remote Systems). The main pane shows a successful self-test message: "The self test completed successfully." Below this are two expandable sections: "Settings" and "External Calibration".

Settings

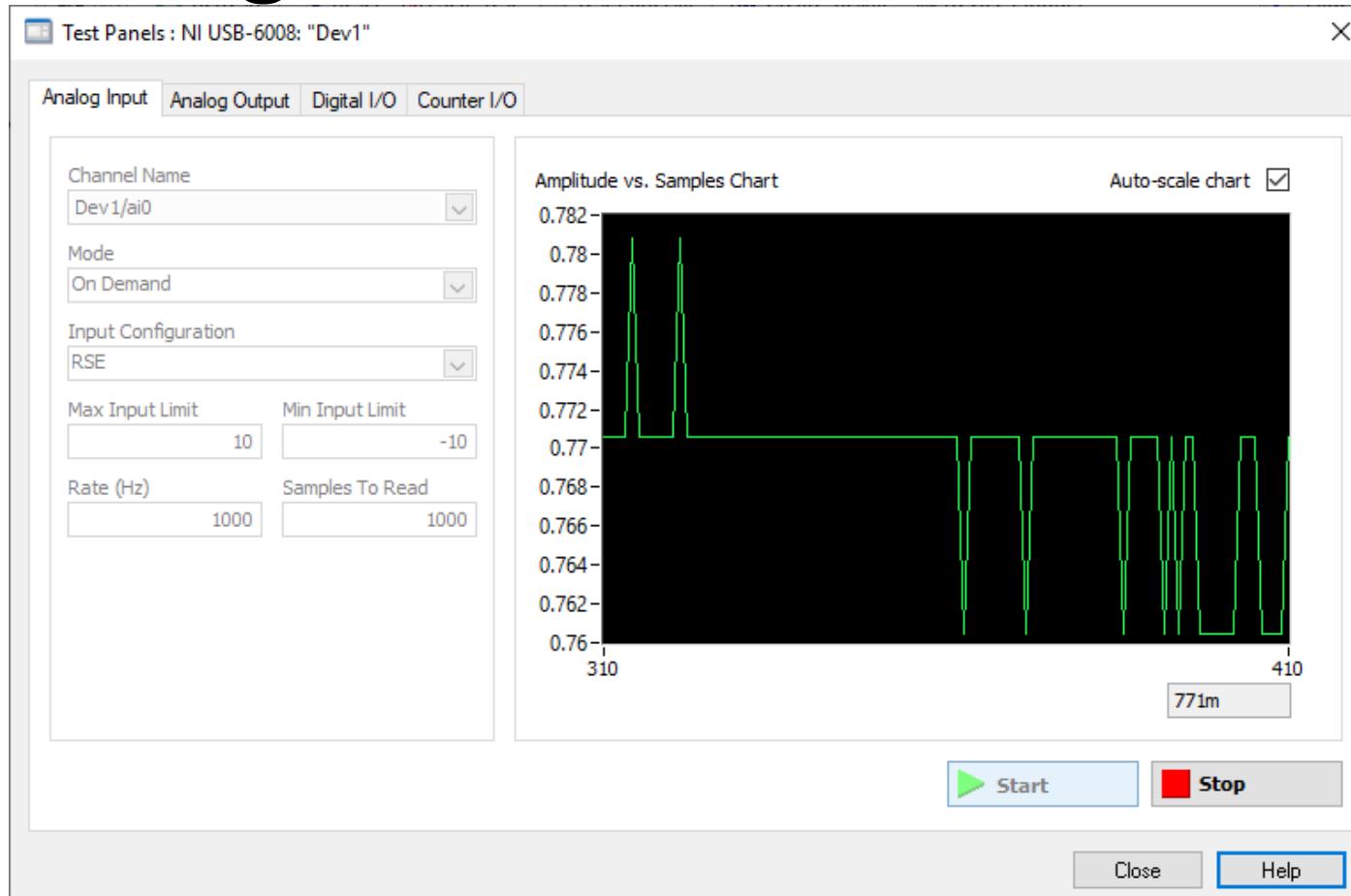
Name	Dev1
Vendor	National Instruments
Model	NI USB-6008
Serial Number	0165408B
Status	Present

External Calibration

Calibration Date	2011-10-03 00:00
Recommended Next Calibration	2012-10-03 00:00

On the right side, there is a panel titled "NI-DAQmx Device Basics" with links to "Run the NI-DAQmx Test Panels", "Remove the device", and "View or change device configuration".

Using the Test Panel in MAX



<https://www.halvorsen.blog>



TMP36

Hans-Petter Halvorsen



TMP36



FRONT



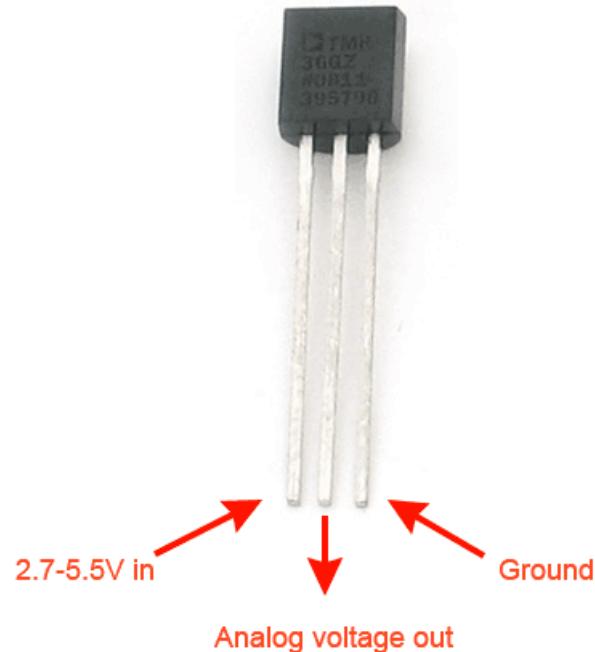
BACK

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

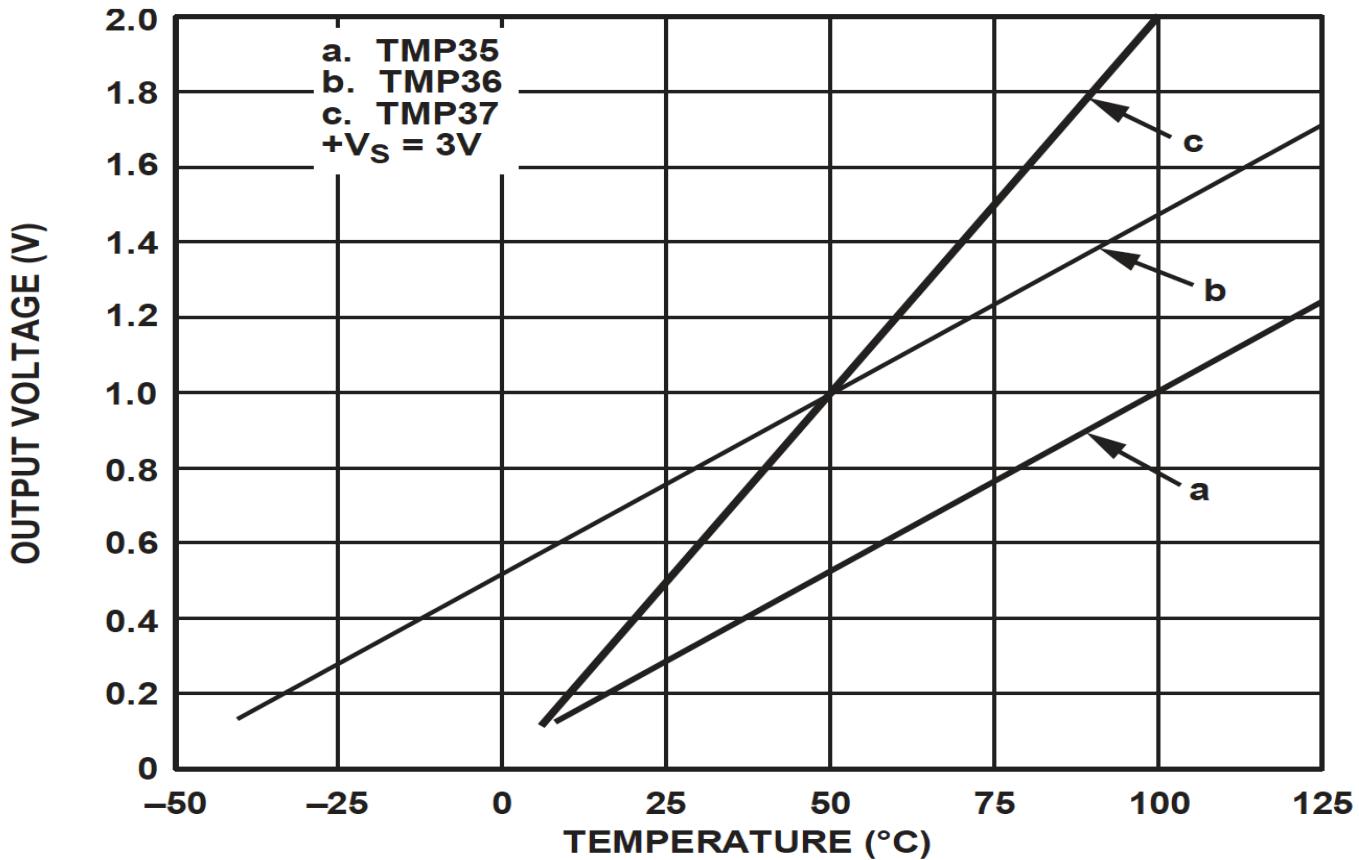
TMP36 Temperature Sensor



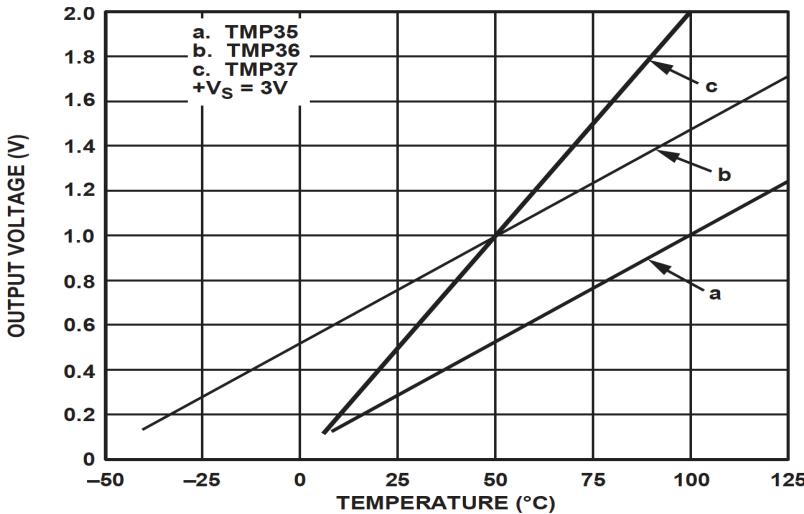
A Temperature sensor like TM36 use a solid-state technique to determine the temperature.

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

TMP36 Datasheet



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:

$$(x_1, y_1) = (0.75V, 25^\circ C)$$
$$(x_2, y_2) = (1V, 50^\circ C)$$

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

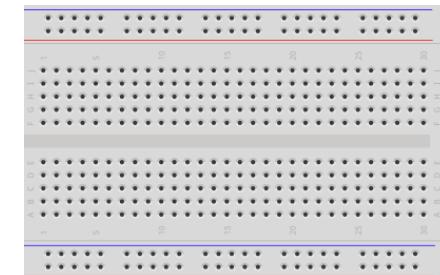
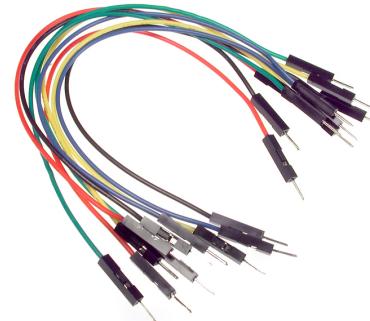
In Norway we typically use Celsius as temperature unit, while in US they use Fahrenheit.

Conversion between these are as follows:

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

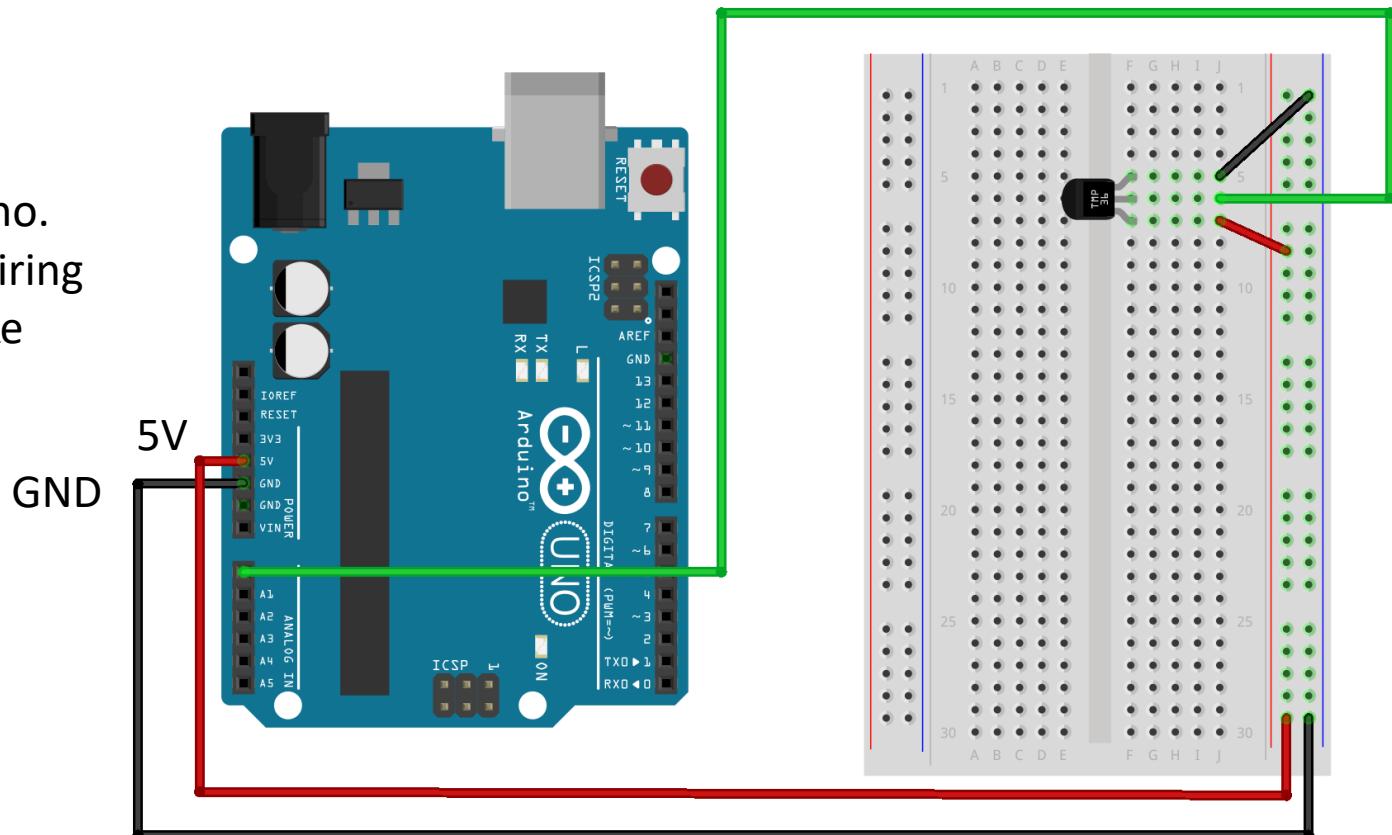
Necessary Equipment

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

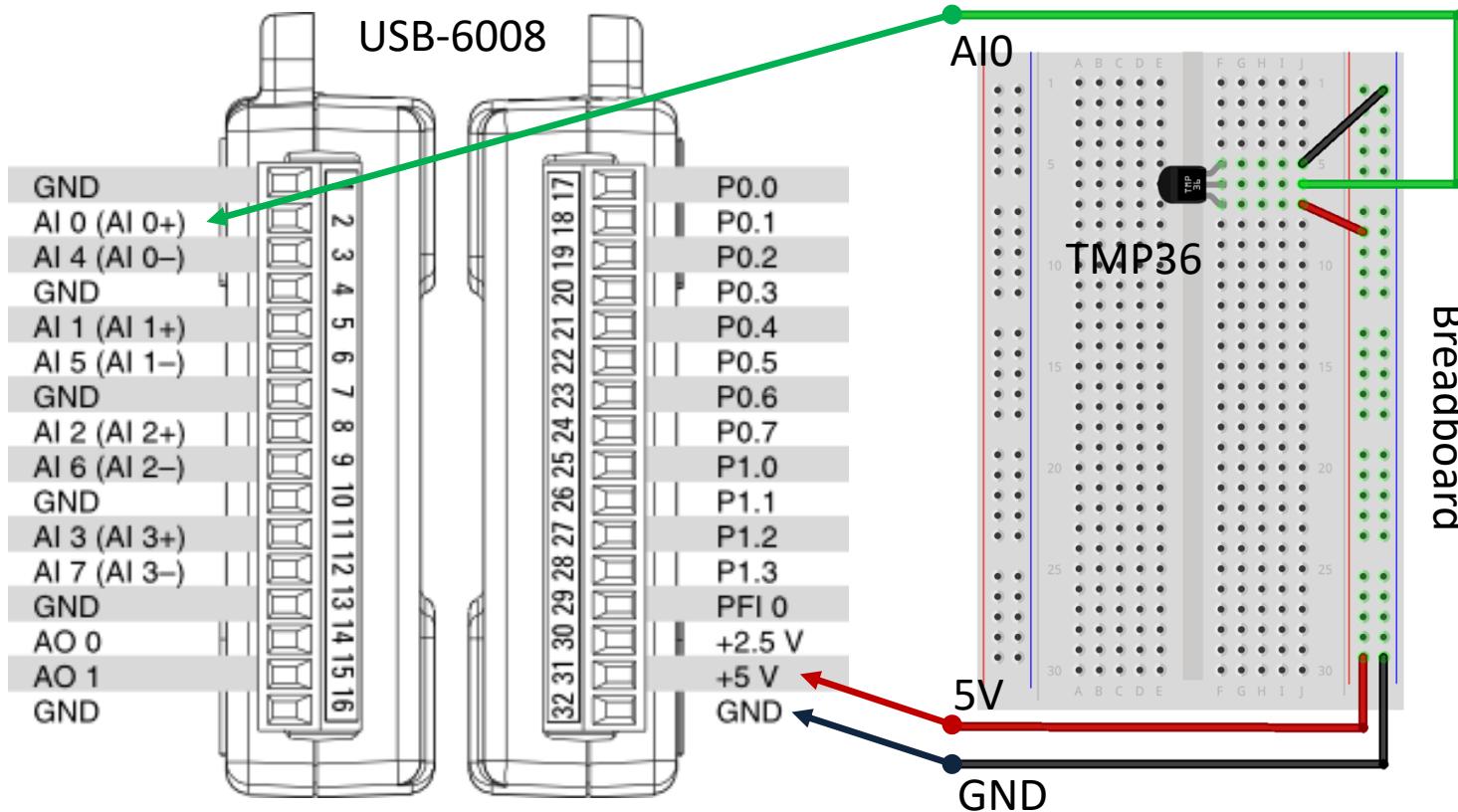


Arduino Wiring Example

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



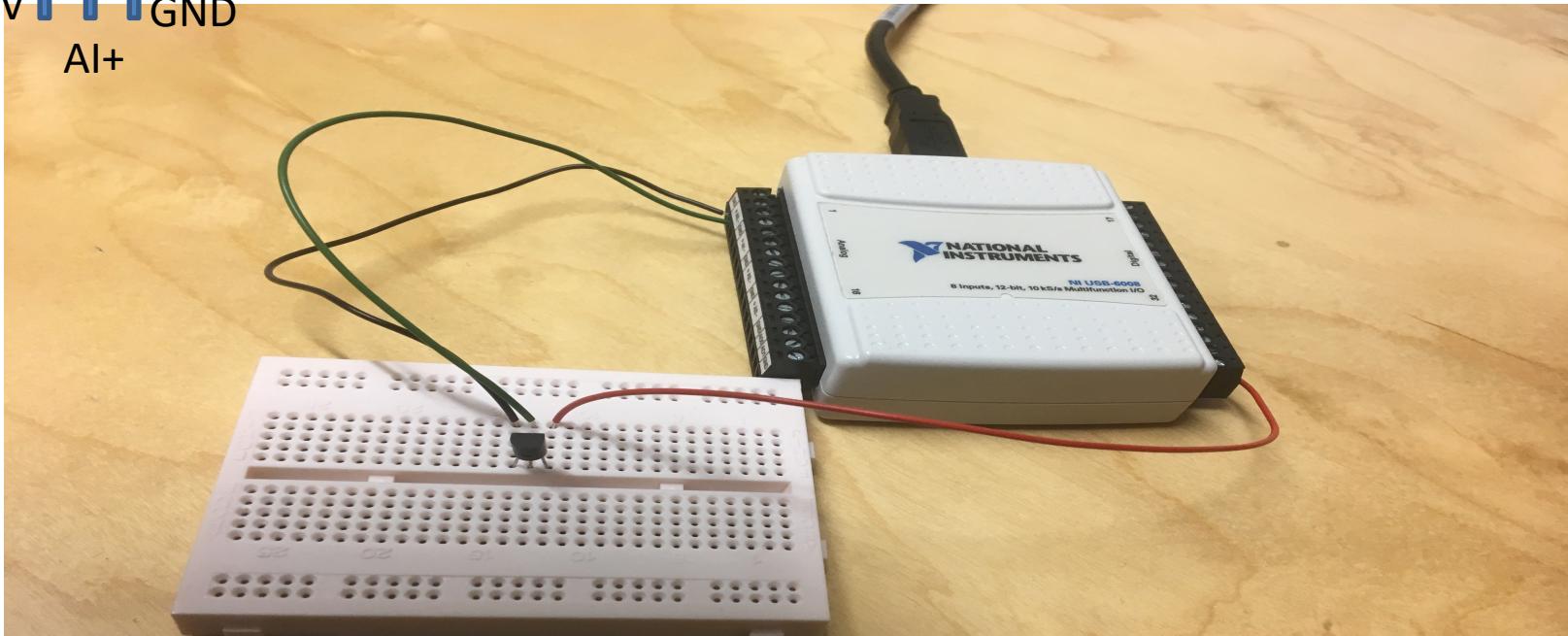
USB-6008 Wiring Example





5V GND
AI+

USB-6008 Wiring Example



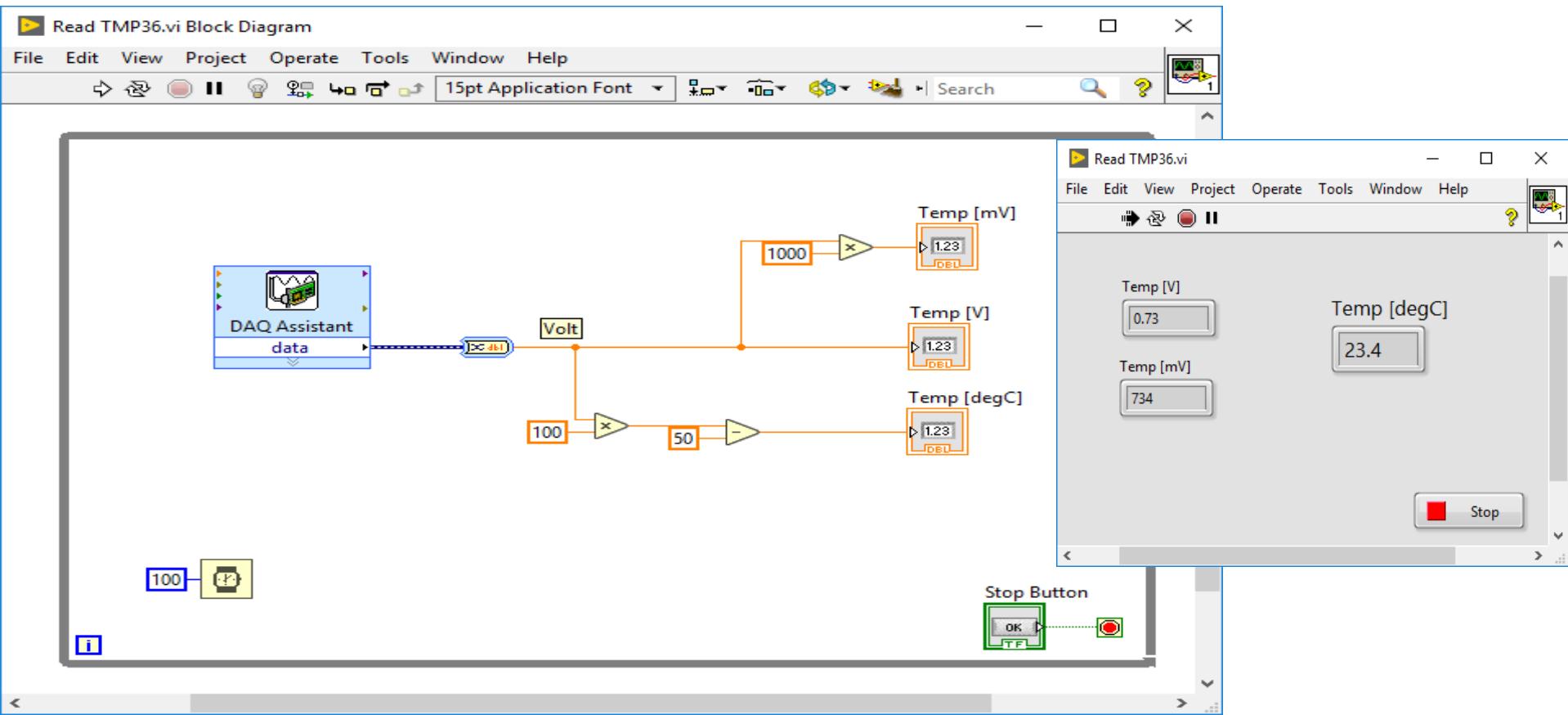
We connect the TMP36 to LabVIEW using a USB DAQ Device from National Instruments, e.g., USB-6001, USB-6008 or similar. I have used a breadboard for the wiring.

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

Installing

Select Agree Review Finish

Additional items you may wish to install:

NI-DAQmx Runtime with Configuration Support
Run-time components required to deploy applications using NI devices and support for configuring NI hardware via the Measu

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

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

NI-DAQmx Support for C
Provides files to create NI-DAQmx applications using ANSI C co

NI-DAQmx Support for LabVIEW 2019 (32-bit)
Provides NI-DAQmx support for LabVIEW 2019 (32-bit)

NI-DAQmx Support for LabVIEW Real-Time and LabWindows/C
Files used to create NI-DAQmx applications with LabVIEW Real-Real-Time Module.

NI Linux RT PXI System Image

NI-DAQmx cDAQ Firmware
Provides firmware for Ethernet CompactDAQ Chassis, FieldDAQ, and NI Linux Real-Time CompactDAQ Controllers

NI-DAQmx Support for Visual Studio 2015
DAQmx integration support for Microsoft Visual Studio 2015

NI-DAQmx Support for Visual Studio 2017
DAQmx integration support for Microsoft Visual Studio 2017

NI-DAQmx Support for Visual Studio 2019
DAQmx integration support for Microsoft Visual Studio 2019

Select All Deselect All

Next

NI-DAQmx Examples

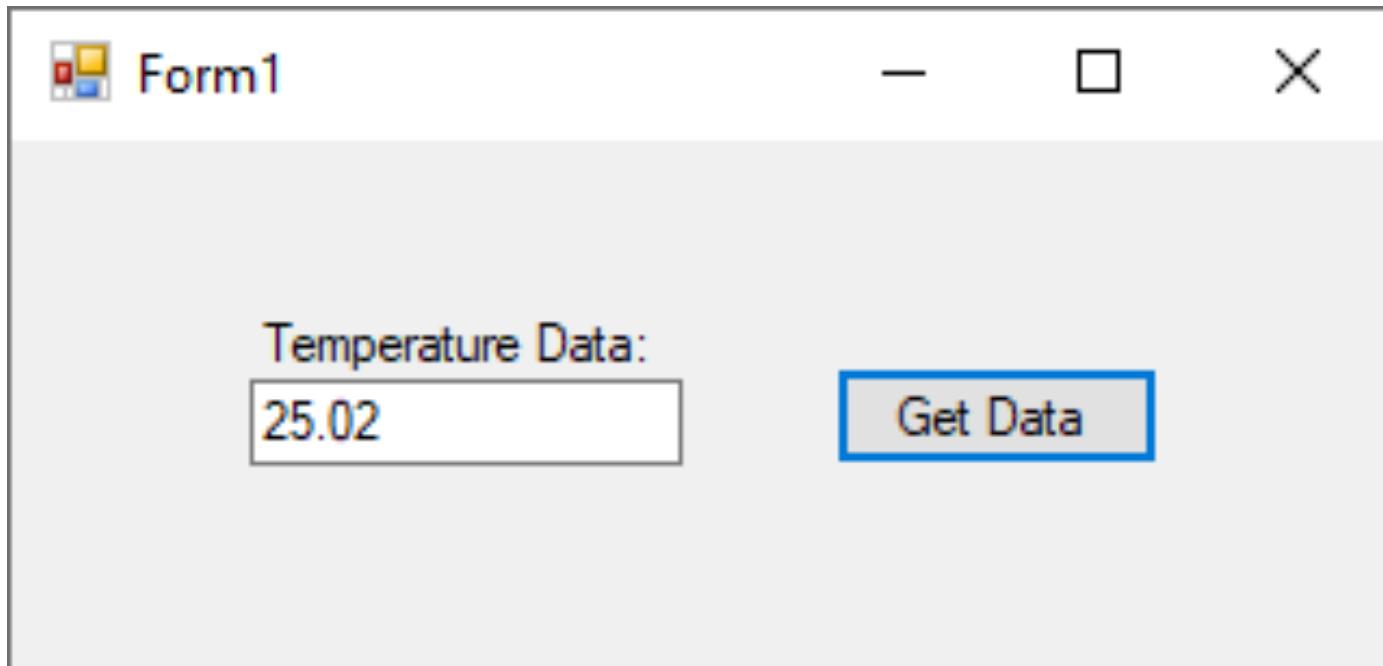
A screenshot of a Windows operating system interface. On the left, there is a vertical taskbar with icons for File Explorer, Task View, Start, and other system functions. A search bar at the bottom left contains the text "DAQ". The main area shows a search results pane on the left with "Best match" results: "NI-DAQmx Examples" (App) highlighted in blue, "NI-DAQmx Documentation", and "DAQ - See web results". The main content area displays the "NI-DAQmx Examples" folder, which is an "App". The folder icon is a yellow folder with a blue ribbon-like shape. The folder name is "NI-DAQmx Examples" and it is located at "C:\Users\Public\Public Documents\National Instruments\NI-DAQ\Examples\DotNET4.5.1". The file list shows the following items:

Name	Date modified	Type	Size
Analog In	2019-06-11 09:11	File folder	
Analog Out	2019-06-11 09:11	File folder	
Control	2019-06-11 09:11	File folder	
Counter	2019-06-11 09:11	File folder	
Digital	2019-06-11 09:11	File folder	
Synchronization	2019-06-11 09:11	File folder	

The file list also includes a "File" menu with options: Open, Open file location, and Pin to Start. Below the file list, there is a navigation bar showing the path: C:\Users\Public\Public Documents\National Instruments\NI-DAQ\Examples\DotNET4.5.1. The right side of the screen shows a standard Windows desktop environment with icons for OneDrive, This PC, and various system icons.

Application

We will create the following Application in Visual Studio:



Create a new project

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

Search for templates (Alt+S)



[Clear all](#)

C#

Windows

Desktop



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

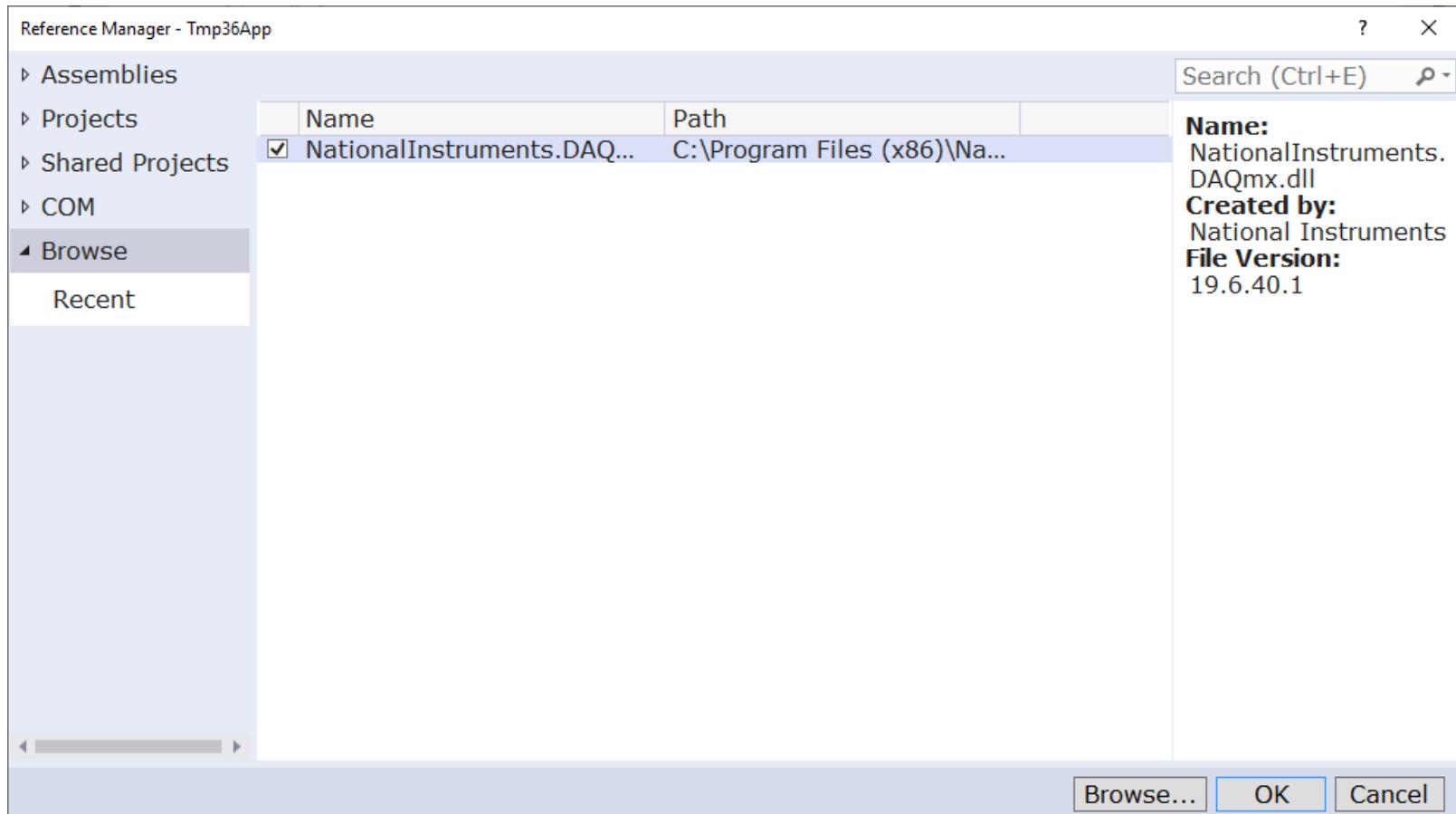
Framework

.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 Diagnostic Tools

Toolbox

Form1.cs [Design]

Form1.cs

Form1

Temperature Data

Get Data

Search Toolbox

Measurement Studio

- Pointer
- NetworkVariableDataSource
- NetworkVariableBrowser...
- ComplexGraph
- DigitalWaveformGraph
- Gauge
- InstrumentControlStrip
- IntensityGraph
- Knob
- Led
- LedArray
- Legend
- Meter
- NumericUpDown
- NumericEditArray
- PropertyEditor
- ScatterGraph
- Slide
- Switch
- SwitchArray
- Tank
- Thermometer
- ToolStripPropertyEditor
- WaveformGraph
- DaqComponent

All Windows Forms

- Pointer

Solution Explorer

Search Solution Explorer (Ctrl+F)

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

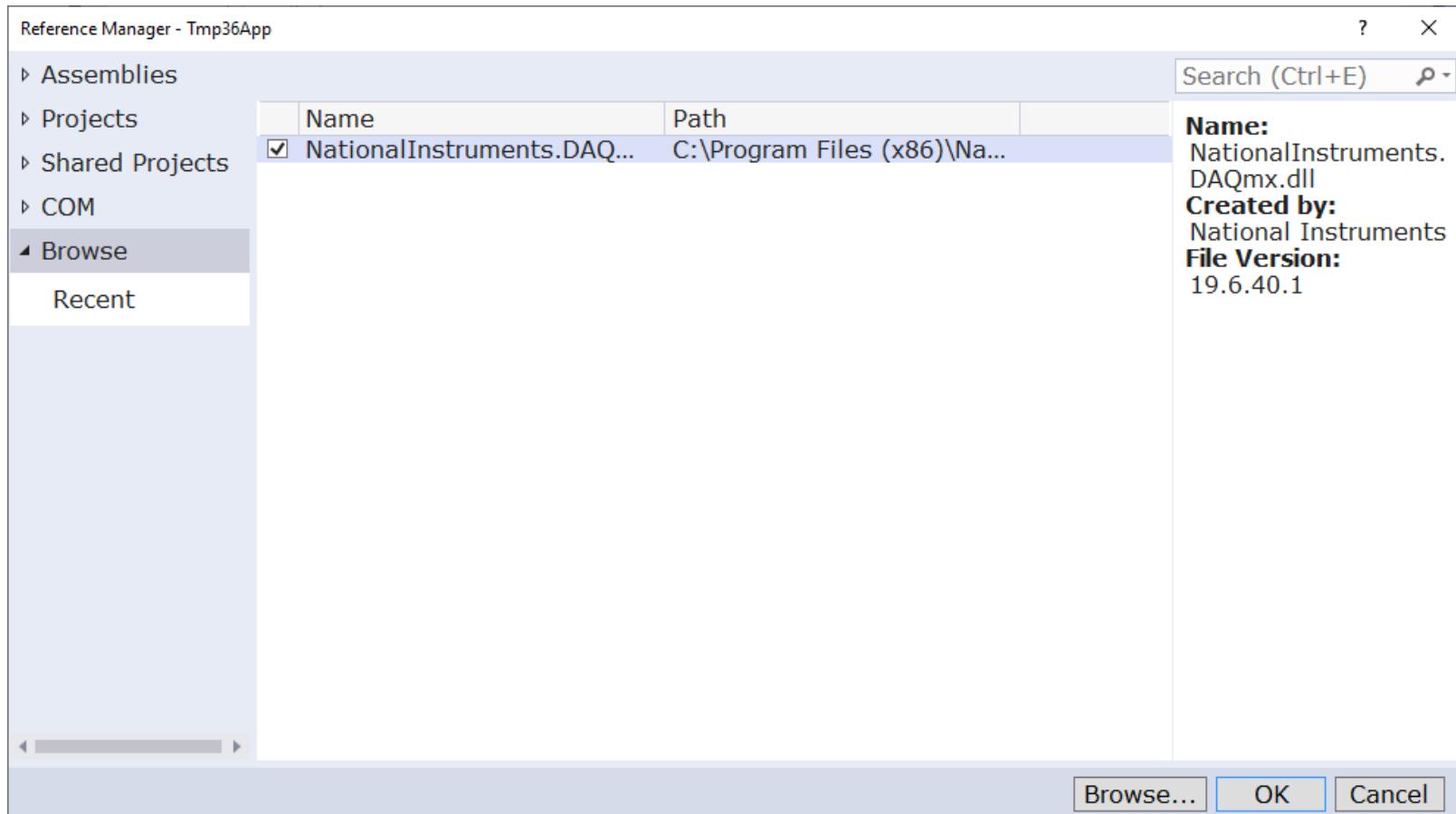
(ApplicationSettings)	
(DataBindings)	
(Name)	txtTempData
AcceptsReturn	False
AcceptsTab	False
AccessibleDescription	

Text

The text associated with the control.

Add to Source Control

NationalInstruments.DAQmx.dll



Object Browser Form1.cs [Design] Form1.cs [Design]

Tmp36App Form1

```
1 using System;
2 using System.Windows.Forms;
3 using NationalInstruments.DAQmx;
4
5 namespace Tmp36App
6 {
7     public partial class Form1 : Form
8     {
9         public Form1()
10        {
11            InitializeComponent();
12        }
13
14        private void btnGetData_Click(object sender, EventArgs e)
15        {
16            Task analogInTask = new Task();
17
18            AIChannel myAIChannel;
19
20            myAIChannel = analogInTask.AIChannels.CreateVoltageChannel(
21                "dev1/ai0",
22                "myAIChannel",
23                AITerminalConfiguration.Rse,
24                0,
25                5,
26                AIVoltageUnits.Volts
27            );
28
29            AnalogSingleChannelReader reader = new AnalogSingleChannelReader(analogInTask.Stream);
30
31            double DaqValue = reader.ReadSingleSample();
32
33            double tmp36Value = DaqValue * 100 - 50;
34
35            txtTempData.Text = tmp36Value.ToString("0.00");
36        }
37    }
38 }
```

Solution Explorer

Search Solution Explorer (Ctrl + F)

Solution 'Tmp36App' (1 of 1 project)

- Properties
- References
- App.config
- Form1.cs
- Form1.Designer.cs
- Form1.resx
- Program.cs

Properties

```
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);

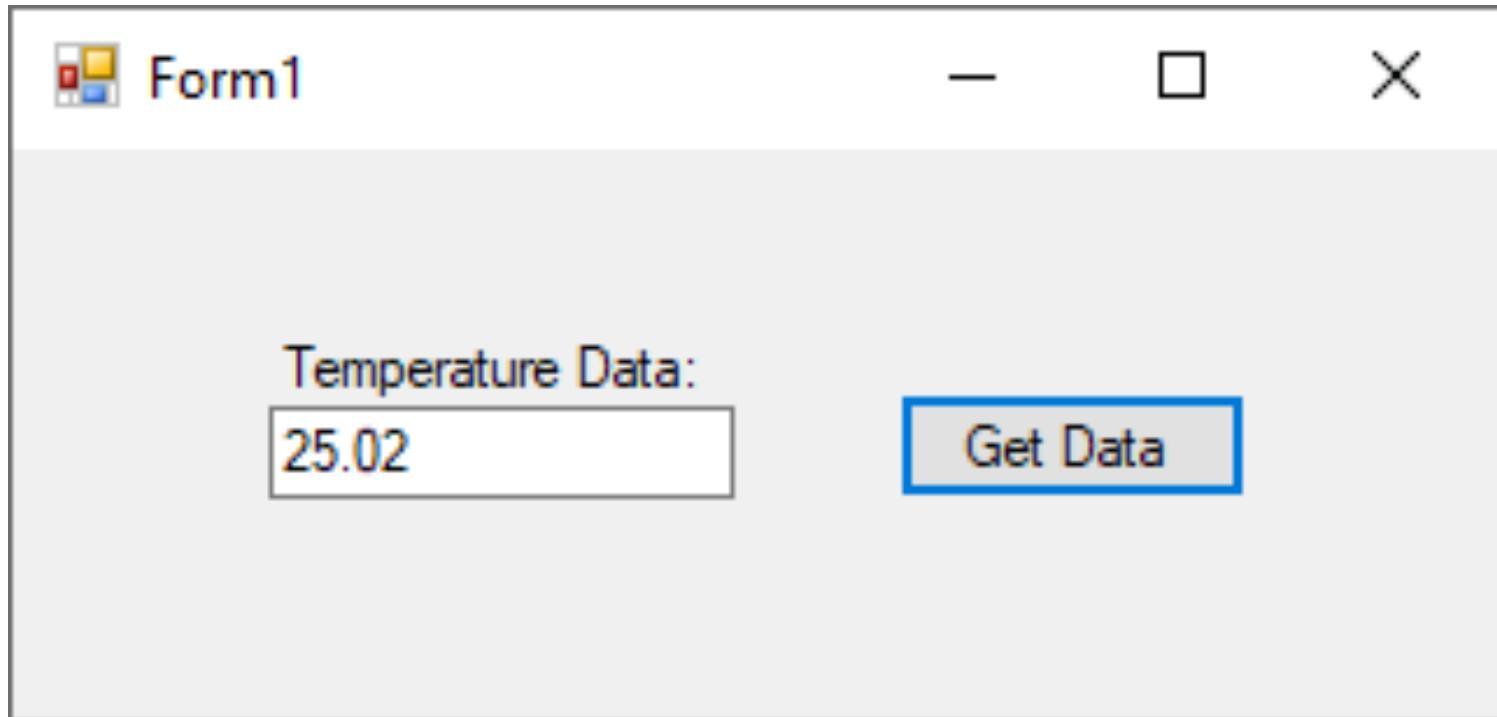
            double DaqValue = reader.ReadSingleSample();

            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>

