

<https://www.halvorsen.blog>

# Temperature Sensor TC-01 in Visual Studio

Visual Studio/C# Code Examples

Hans-Petter Halvorsen



# Contents

- Introduction
  - TC-01 Temperature Sensor
  - NI DAQmx
- Code Examples in Visual Studio/C#
  - Basic Example – Read a Temperature Value from the Sensor
  - Using a Timer

<https://www.halvorsen.blog>

# Introduction



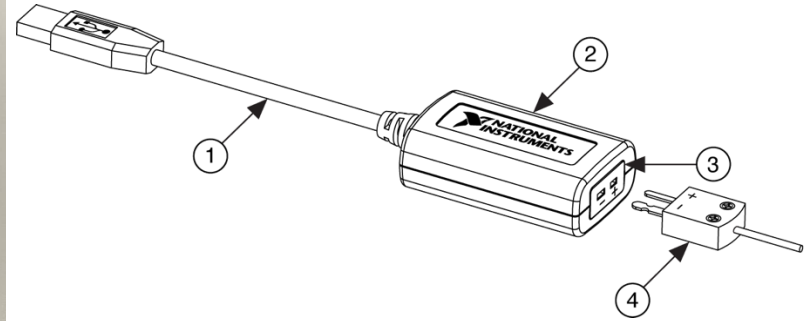
[Table of Contents](#)

Hans-Petter Halvorsen

# TC-01 Temperature Sensor

- TC-01 is a Temperature Sensor from NI/Emerson
- You connect it to your PC using a built-in USB A cable
- To use it in LabVIEW, C#, Python or MATLAB you need to install the **NI-DAQmx** driver. Free download from Internet.
- We will use Visual Studio/C#

# TC-01 Temperature Sensor

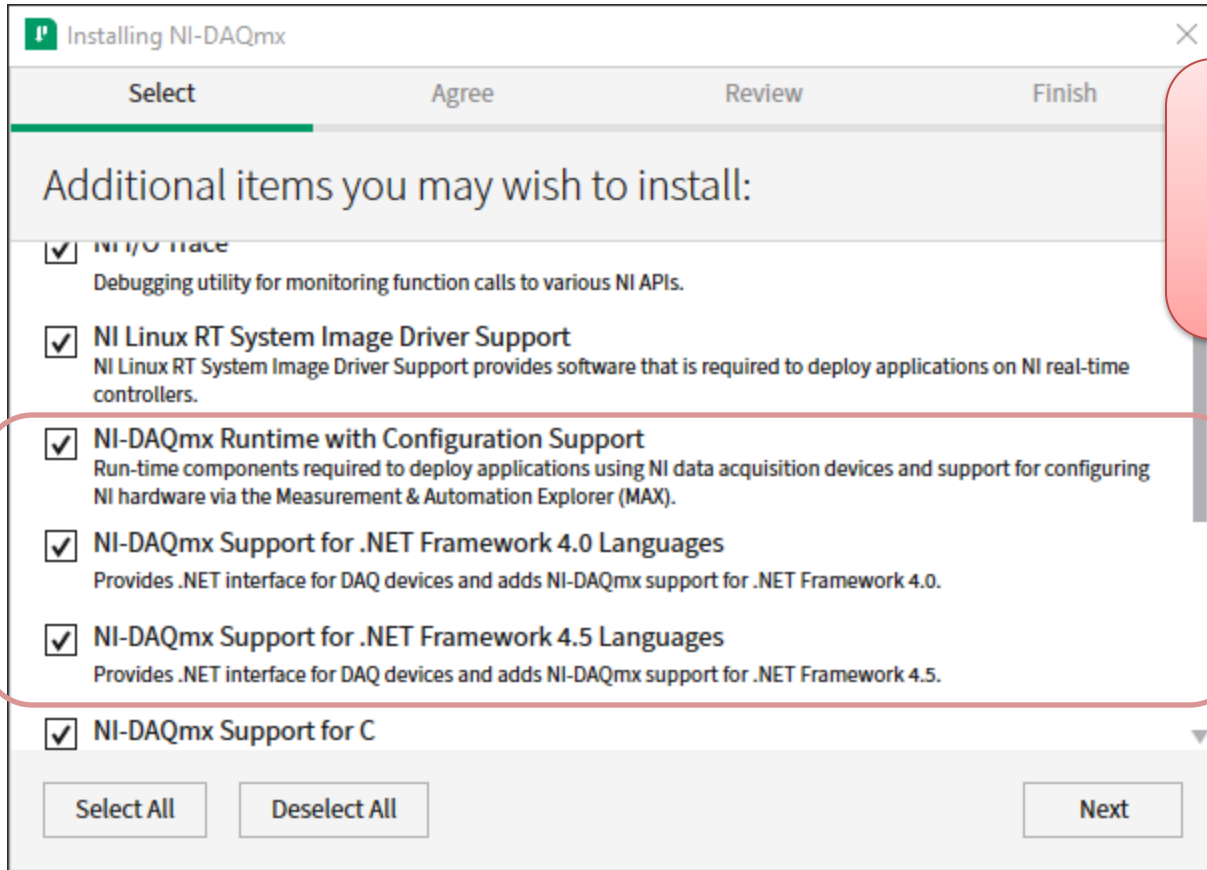


1. USB Cable
2. NI USB-TC01 Device
3. Thermocouple Input
4. Thermocouple Sensor

# 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, such as TC-01 Thermocouple Temperature Sensor.
- [www.ni.com/downloads](http://www.ni.com/downloads)

# NI-DAQmx Installation



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

# MAX

The screenshot shows the NI Measurement & Automation Explorer (MAX) interface. The title bar reads "NI USB-TC01 'Dev1' - Measurement & Automation Explorer". The left sidebar shows a tree view with "My System" expanded, containing "Data Neighborhood", "Devices and Interfaces", "Network Devices", "Historical Data", "Scales", "Software", and "Remote Systems". Under "Devices and Interfaces", "NI USB-TC01 'Dev1'" is selected. The main pane displays the "Settings" for this device, with a message: "Try the new Hardware Configuration Utility to configure your device." Below this is a table of device properties:

Name	Dev1
Vendor	National Instruments
Model	NI USB-TC01
Serial Number	017EAF07
Status	Present

The interface also includes a menu bar (File, Edit, View, Tools, Help), a toolbar with icons for Save, Refresh, Self-Test, Hardware Configuration Utility, Test Panels..., Create Task..., Configure TEDS..., and Hide Help, and a "Back" button.

Note the Name of the Device. You need to use that name in the C# code. But you can change the default Name here in MAX



<https://www.halvorsen.blog>

# Code Examples



[Table of Contents](#)

Hans-Petter Halvorsen

# Create a new project

## 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#

Search for templates (Alt+S)

Clear all

C# Windows Desktop

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  
A class library that targets

**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**

Back

Next

# Configure your new project

Windows Forms App (.NET Framework) C# Windows Desktop

Project name

Temperature Sensor

Location

C:\Users\hansha\source\repos

Solution name ⓘ

Temperature Sensor

Place solution and project in the same directory

Framework

.NET Framework 4.8

Project will be created in "C:\Users\hansha\source\repos\Temperature Sensor\Temperature Sensor"

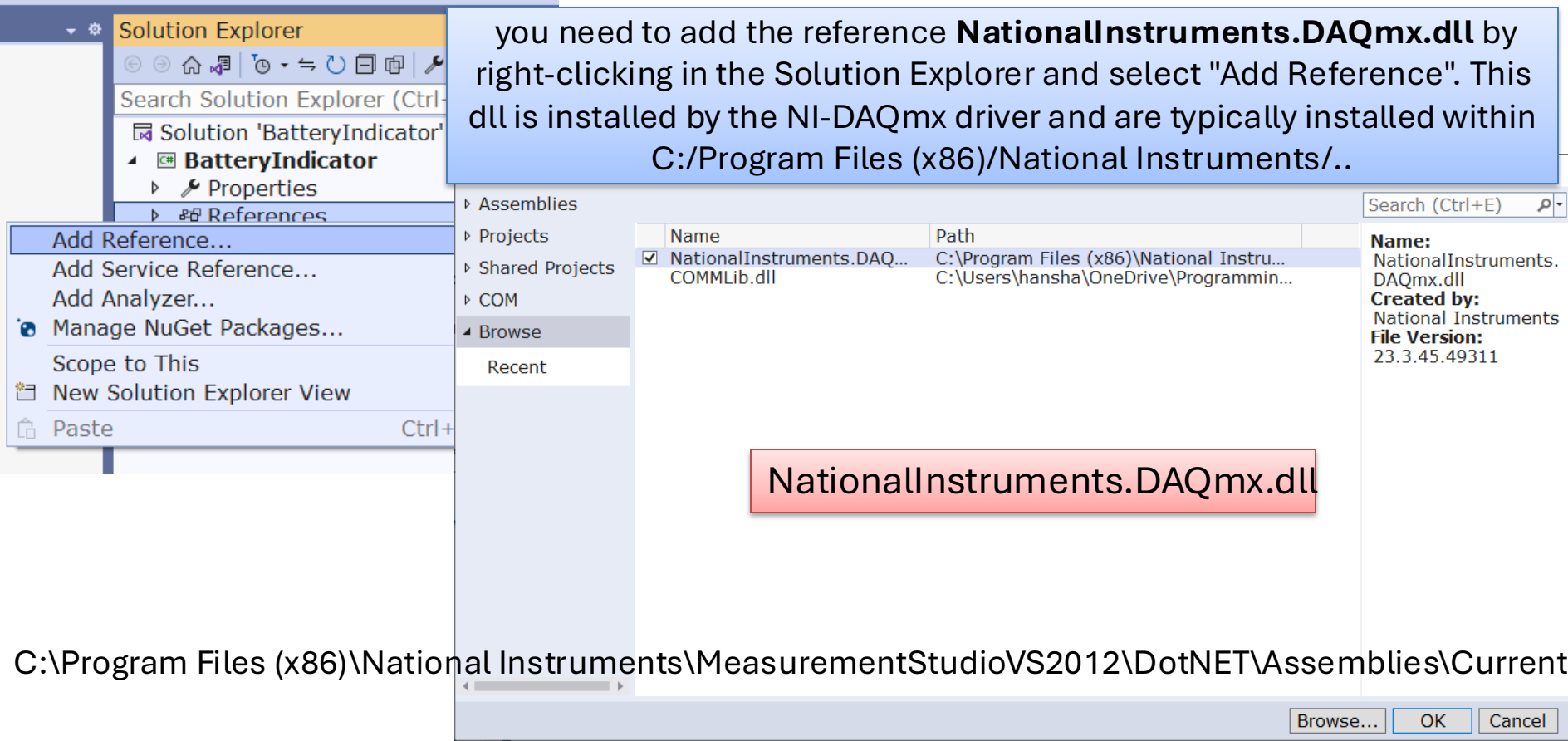
Back

Create

**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/..**



The screenshot shows the 'Add Reference' dialog box in Visual Studio. The 'Browse' tab is selected, and the file list shows 'NationalInstruments.DAQmx.dll' selected. The file path is C:\Program Files (x86)\National Instruments\MeasurementStudioVS2012\DotNET\Assemblies\Current. The dialog also shows the file's name, creator (National Instruments), and version (23.3.45.49311).

Name	Path
<input checked="" type="checkbox"/> NationalInstruments.DAQ... COMMLib.dll	C:\Program Files (x86)\National Instru... 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

<https://www.halvorsen.blog>

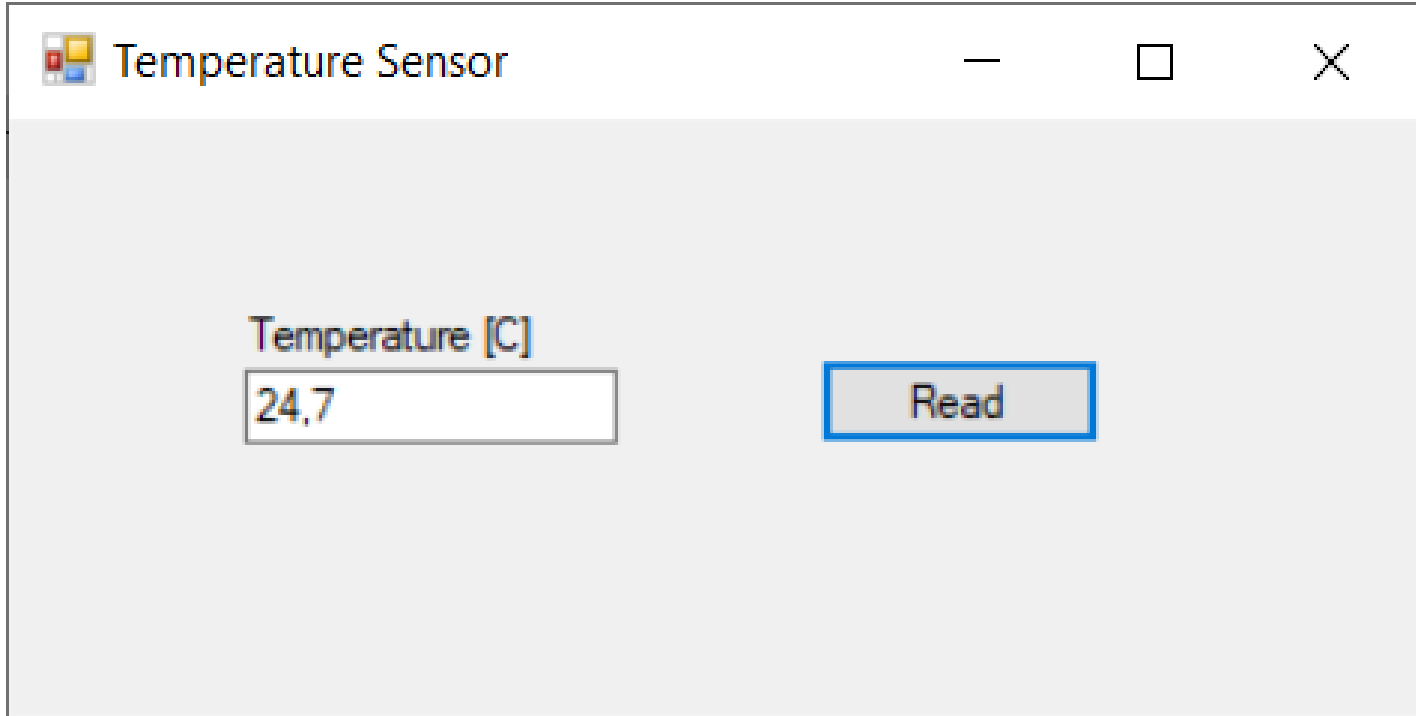
# Basic Example



[Table of Contents](#)

Hans-Petter Halvorsen

# User Interface



# Add Namespace

```
using NationalInstruments.DAQmx;
```

# ReadTemperature()

```
double ReadTemperature()
```

```
{
```

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

```
    AIChannel myAIChannel;
```

```
    myAIChannel = analogInTask.AIChannels.CreateThermocoupleChannel(
```

```
        "dev1/ai0",
```

```
        "Temperature",
```

```
        0,
```

```
        100,
```

```
        AIThermocoupleType.J,
```

```
        AITemperatureUnits.DegreesC
```

```
    );
```

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

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

```
    return temperature;
```

```
}
```



# Button Event Handler()

```
private void btnReadTemperature_Click(object sender, EventArgs e)
{
    double temperature;
    temperature = ReadTemperature();
    txtTemperature.Text = temperature.ToString("0.0");
}
```

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

```
namespace Temperature_Sensor
```

```
{
    public partial class Form1 : Form
```

```
{
    ..
```

```
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.CreateThermocoupleChannel(
```

```
        "dev1/ai0",
        "Temperature",
        0,
        100,
        AIThermocoupleType.J,
        AITemperatureUnits.DegreesC
    );
```

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

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

```
    return temperature;
```

```
}
```

```
}
```

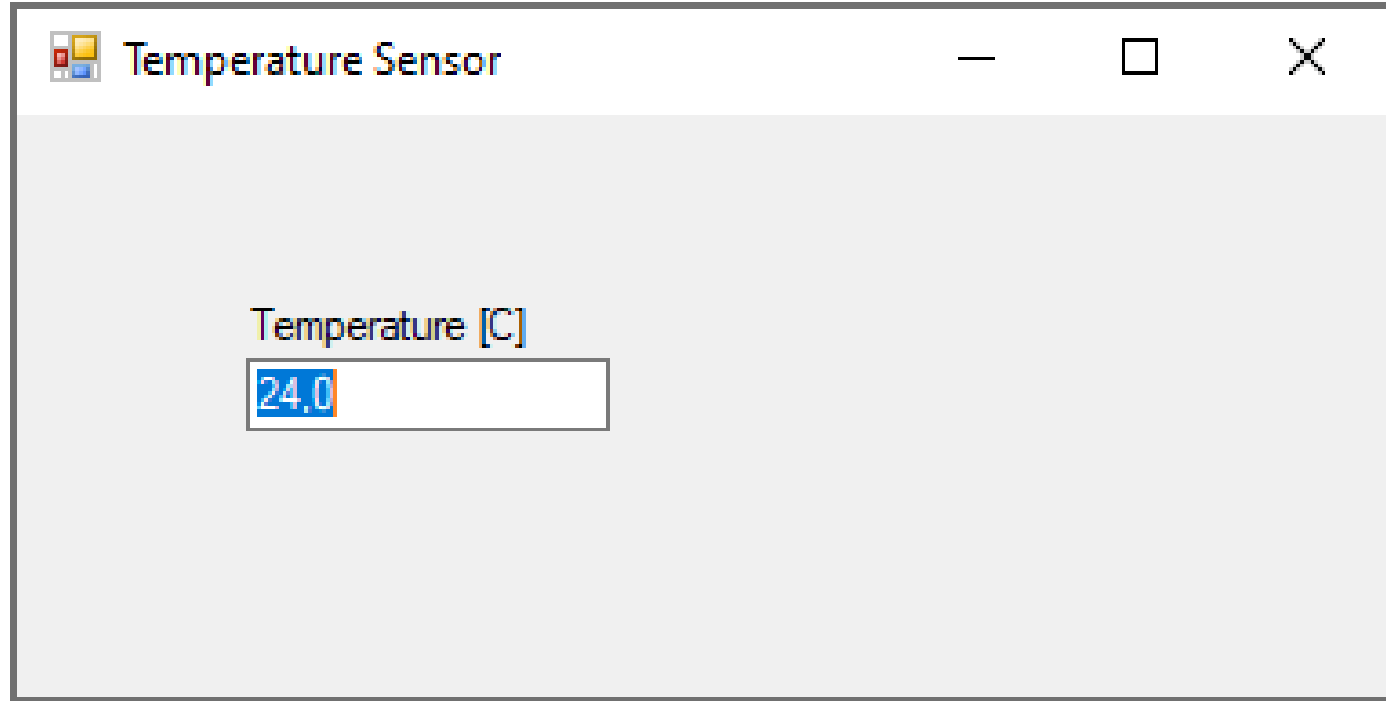
```
}
```

# Use a Timer for Continuous Temperature Readings



Hans-Petter Halvorsen

# User Interface



# Timer

```
private void timer1_Tick(object sender, EventArgs e)
{
    double temperature;
    temperature = ReadTemperature();
    txtTemperature.Text = temperature.ToString("0.0");
}
```

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

```
namespace Temperature_Sensor
```

```
{
```

```
    public partial class Form1 : Form
```

```
    {
```

```
        public Form1()
```

```
        {
```

```
            InitializeComponent();
```

```
            timer1.Start();
```

```
        }
```

```
        private void timer1_Tick(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.CreateThermocoupleChannel(
```

```
                "dev1/ai0",
```

```
                "Temperature",
```

```
                0,
```

```
                100,
```

```
                AIThermocoupleType.J,
```

```
                AITemperatureUnits.DegreesC
```

```
            );
```

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

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

```
            return temperature;
```

```
        }
```

```
    }
```

```
}
```

# Hans-Petter Halvorsen

University of South-Eastern Norway

[www.usn.no](http://www.usn.no)

E-mail: [hans.p.halvorsen@usn.no](mailto:hans.p.halvorsen@usn.no)

Web: <https://www.halvorsen.blog>

