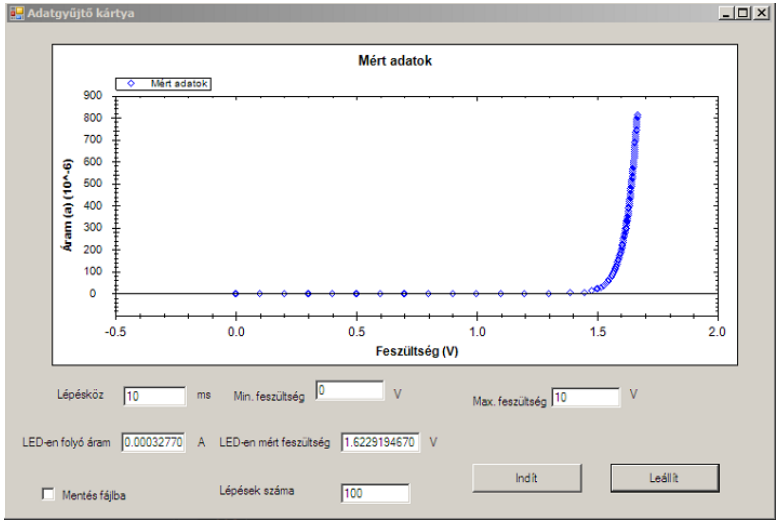
Physics laboratory for MSc students



Presenting:

Gergő Fülöp

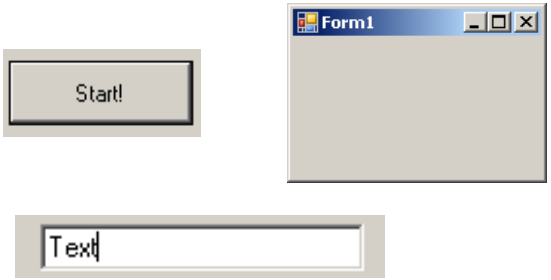
BME TTK Physics Department
2023/2024
Autumn semester

MŰEGYETEM 1782

Outline

Day 1

- Setting the goal
- Basics of C# programming
  - Making a GUI
  - Syntax, flow control, etc.
  - Debugging in Visual Studio
  - File I/O



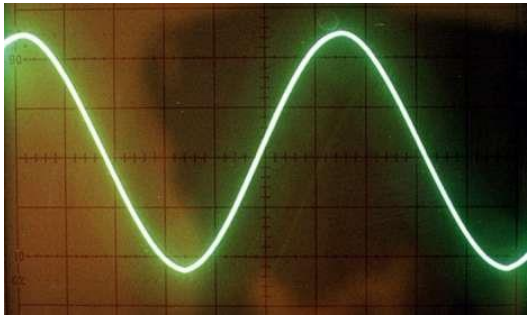
Day 2

- Connecting instruments via serial port
  - Communication with instrument
  - Display acquired data
- Solving complex measurement control and data analysis tasks
  - NI myDAQ data acquisition device

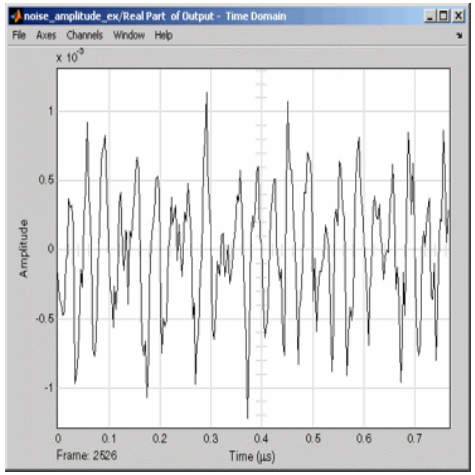
What is still missing…

Visualization of measurement data

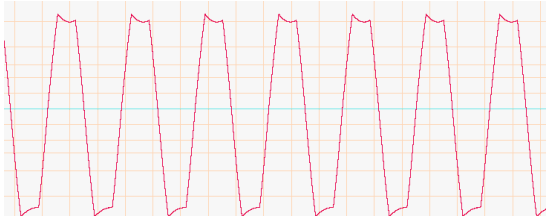
Is there a signal?



Estimating SNR

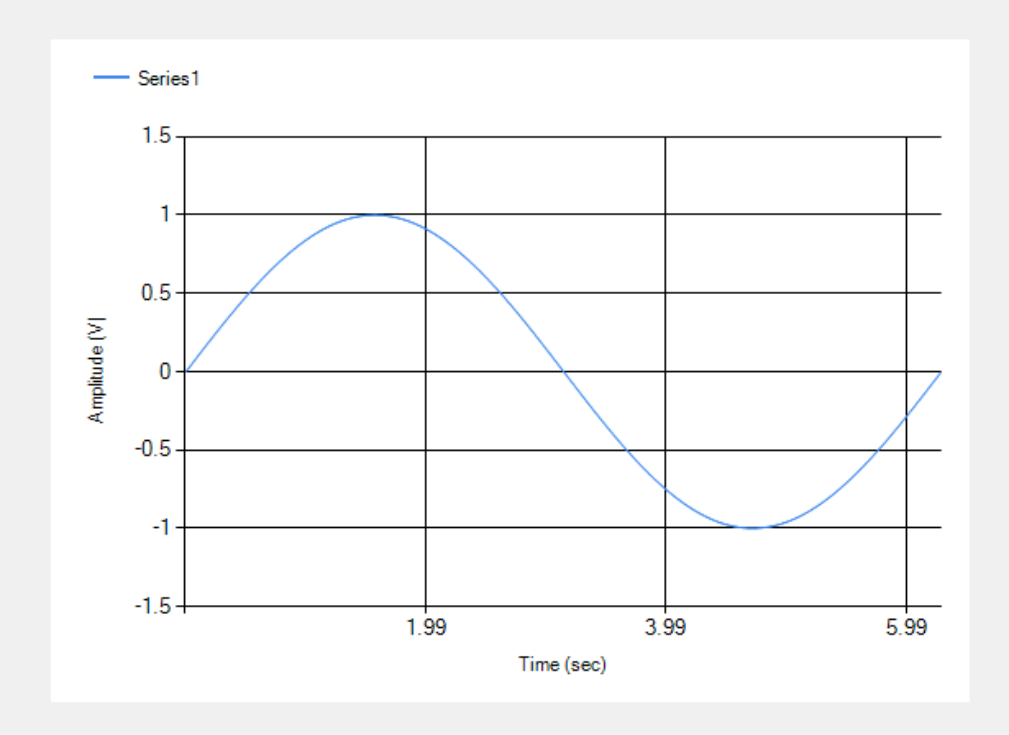


Identifying malfunction



What is still missing…

Real-time plotting

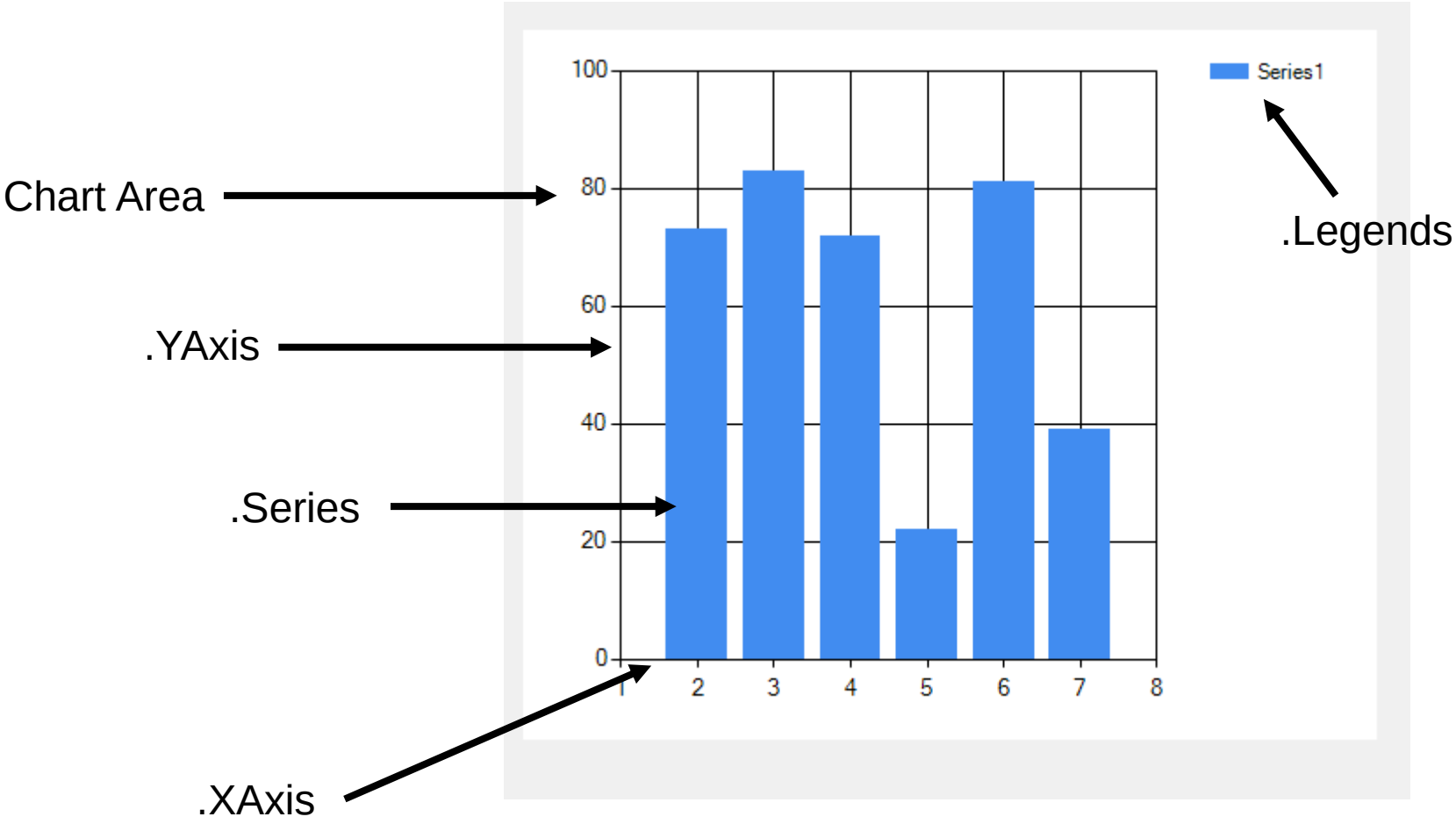


Plot axis labeled properly

- Name of the quantity
- Units of the quantity

Chart

Inserting a Chart control: Toolbox → Data → Chart



Components of the Chart

```
using System.Windows.Forms.DataVisualization.Charting;
```

Chart Area Title
Titles[0]

Properties	
.Text	Title of the plot

Warning: the Titles collection is empty by default. Add a Member either in the Properties editor, or programmatically (e.g. chart1.Titles.Add("Title");)

Components of the Chart: x and y axis

```
using System.Windows.Forms.DataVisualization.Charting;
```

chart1.ChartAreas[0].AxisX

Axis	
.Title	Axis label
.Minimum	Minimum value of the axis (Double.NaN – autoscale)
.Maximum	Maximum value of the axis (Double.NaN – autoscale)
.IsLogarithmic	Log / Lin scale
.LabelStyle.Format	Number format (e.g. "0.00“ for fixed 2 digits after decimal)

https://docs.microsoft.com/en-us/dotnet/standard/base-types/custom-numeric-format-strings?redirectedfrom=MSDN

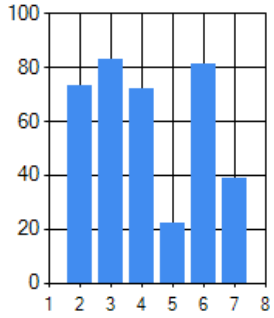
Warning: setting a *Minimum* value higher than the *Maximum* value results in an *Exception*.

Components of the Chart: Series

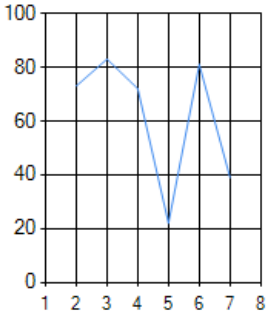
```
using System.Windows.Forms.DataVisualization.Charting;
```

Chart1.Series[0]

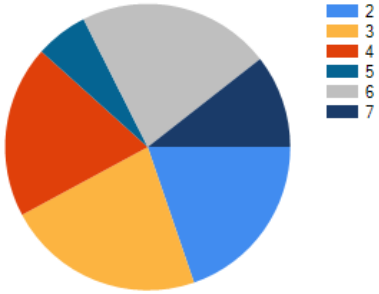
Methods:	
.Points.AddXY(X, Y)	Add a data point to the plot specified by x and y coordinates
Properties:	
.ChartType	Appearance of the plot, e.g. SeriesChartType.Line



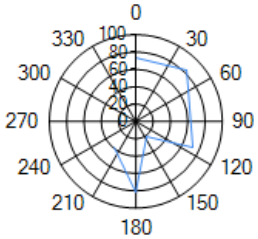
Column



Line



Pie



Polar

Software timing with a Timer object

Timer:

```
using System.Windows.Forms;
```

Properties	
Interval	Time between two ticks in milliseconds
Enabled	Tells whether the timer is running

Events	
Tick	The method which runs repeatedly

Example for plotting with the Chart control

```
private void Form1_Load(object sender, EventArgs e)
        {
            chart1.Titles.Add("Sine wave");

            chart1.ChartAreas[0].AxisX.Title = "Time (sec)";
            chart1.ChartAreas[0].AxisY.Title = "Amplitude (V)";

            chart1.ChartAreas[0].AxisX.LabelStyle.Format = "0.##";
            chart1.ChartAreas[0].AxisY.LabelStyle.Format = "0.##";

            chart1.Legends[0].Docking = Docking.Top;

            chart1.Series[0].ChartType = SeriesChartType.Line;
        }

        private void buttonPlot_Click(object sender, EventArgs e)
        {
            double x,y;
            for(int i=0; i<1000; i++)
            {
                x = (double) i/1000*2*Math.PI;
                y = Math.Sin(x);
                chart1.Series[0].Points.AddXY(x,y);
            }
        }
```

Accessing data points of the Chart

n*th* point pair

```
DataPoint P = chart1.Series[0].Points.ElementAt(n);
```

X value:

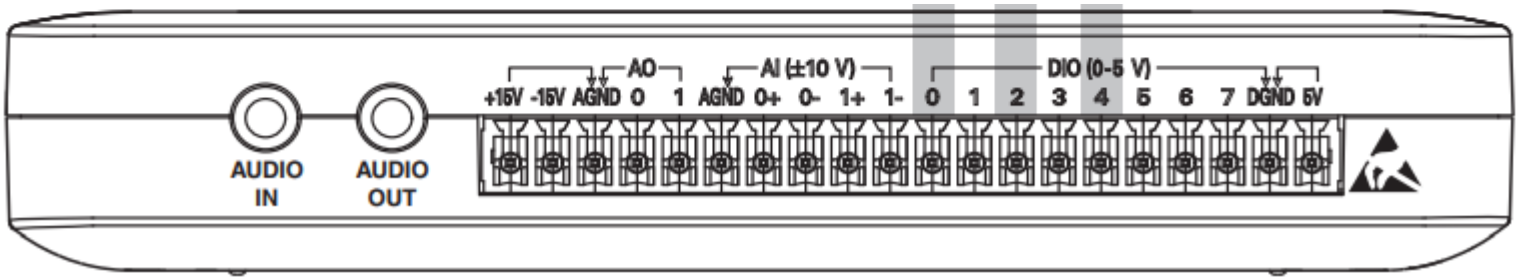
```
P.XValue
```

Y value:

```
P.Yvalues[0]
```

NI myDAQ

- Digital input/outptut
- Analog input/output
  - Range: ±10 V or ±2 V
  - Resolution: 16 bits
  - Max. sampling rate: 200 kS/s
  - Input: differential or single-ended
- Digital multimeter
  - Voltage, current, resistance



Manual (pdf): https://www.ni.com/pdf/manuals/373061g.pdf

DAQmx workflow

Create Task → Set up the Task → Create Reader/Writer

Read/Write samples

Dispose of the Task

- Type of input
  - Analog
  - Digital
- Number of channels:
  - Single
  - Multi
- Number of samples:
  - Single → software timing (with a Timer)
  - Multi → hardware timing

Programming the myDAQ

Add reference manually in the solution explorer
Add the using directive below to the top of your code

```
using NationalInstruments.DAQmx;
```

Example: analog input (single sample)

Declarations:

```
NationalInstruments.DAQmx.Task InTask;
AnalogSingleChannelReader analogReader;
```

Creating the task and the reader object:


```
InTask = new NationalInstruments.DAQmx.Task();

InTask.AIChannels.CreateVoltageChannel("myDAQ1/ai0", "",
        AITerminalConfiguration.Differential, -10, 10, AIVoltageUnits.Volts);

analogReader = new AnalogSingleChannelReader(InTask.Stream);
```

Reading a single sample:

```
MeasuredValue = analogReader.ReadSingleSample()
```

Disposal:

```
InTask.Dispose();
```

Example: analog output (single sample)

Declarations:

```
NationalInstruments.DAQmx.Task OutTask;
AnalogSingleChannelWriter analogWriter
```

Creating the task and the writer object:

```
OutTask = new NationalInstruments.DAQmx.Task();

OutTask.AOChannels.CreateVoltageChannel("myDAQ1/ao0", "",
        -10, 10, AOVoltageUnits.Volts);

analogWriter = new AnalogSingleChannelWriter(OutTask.Stream);
```

Check

NI MAX

Writing a single sample:

bool autoStart

```
analogWriter.WriteSingleSample(true, data)
```

Disposal:

```
OutTask.Dispose();
```

Programming the myDAQ: reading/writing multiple samples

Input and output sampling with hardware timing:

```
InTask.Timing.ConfigureSampleClock("", SampleRate,
       SampleClockActiveEdge.Rising, SampleQuantityMode.FiniteSamples, numPnts);


OutTask.Timing.ConfigureSampleClock("", SampleRate,
       SampleClockActiveEdge.Rising, SampleQuantityMode.FiniteSamples, numPnts);
```

Writing multiple values:

```
analogWriter.WriteMultiSample(true, outData);
```

Reading multiple values:

```
InTask.Start();

double[] inData = new double[numPnts];
inData = analogReader.ReadMultiSample(numPnts);
```

https://fizipedia.bme.hu/index.php/Oszcilloszk%C3%B3p_program_myDAQ-al

Using the SerialPort

Add a SerialPort object in the Designer view from the Toolbox.

Set up the communication parameters (instrument-specific, do only once):

```
serialPort1.PortName = "COM1";
serialPort1.BaudRate = 9600;
serialPort1.DataBits = 8;
serialPort1.StopBits = System.IO.Ports.StopBits.One;
serialPort1.Parity = System.IO.Ports.Parity.None;
```

Check

NI MAX

Open the port:

```
serialPort1.Open();
```

Send a command to the instrument:

```
serialPort1.WriteLine("*idn?");
```

Read the answer from the instrument:

```
ValuetextBox.Text = serialPort1.ReadLine();
```

Repeat as needed

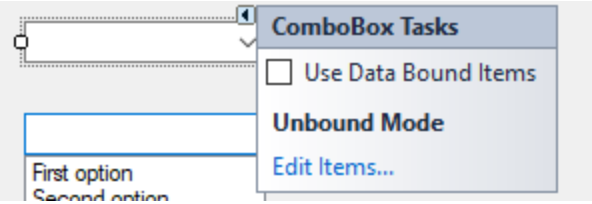
Close the port:

```
serialPort1.Close();
```

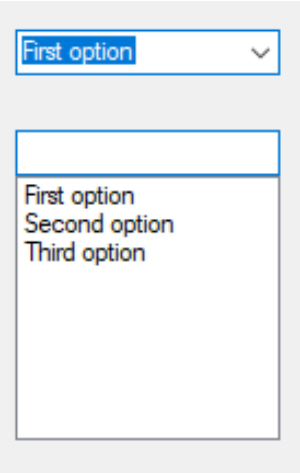
https://fizipedia.bme.hu/index.php/DMM

ComboBox

- Populate the options:
  - Through the GUI, or
  - Items Collection (e.g. comboBox1.Items)



- DropDownStyle
  - DropDown
  - Simple



- Get selected item

```
string selected = comboBox1.GetItemText(comboBox1.SelectedItem);
```

NumericUpDown

- Properties
  - Value
  - Minimum
  - Maximum
  - DecimalPlaces
  - Increment
- Note: type is decimal

Programming exercises

1. Read and plot the (x,y) point pairs in the file `suruasd.txt` with a **Chart** control. Set the type to SeriesChartType.Point. Set the color to green. (Download the data file from Fizipedia.)

Download and open the two-panel template project (*"kétpaneles")* from Fizipedia. Use it for the exercises below, in 3 separate projects.

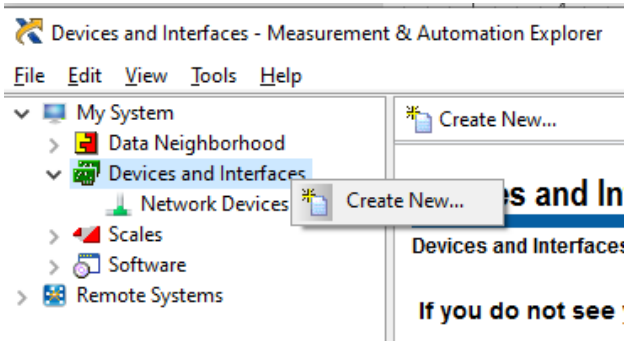
2. Create a **Timer** object with an interval of 100 ms. Upon each tick, read a **single sample** using the analog input of the simulated myDAQ. Use a chart to visualize the data real-time.

3. Create a function generator using the analog output of the myDAQ with **hardware timing**. The user selects a waveform (sine, triangle, square), a frequency and an amplitude. Display the generated waveform in a Chart.

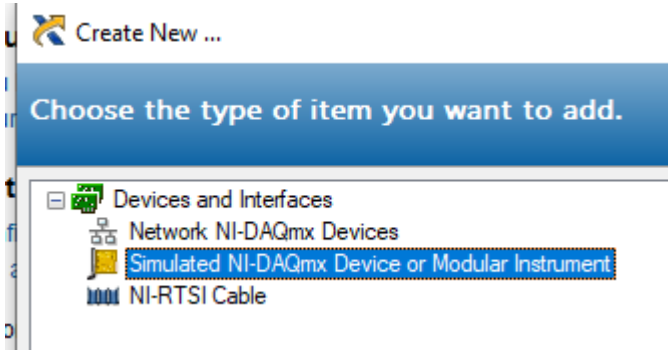
4. Using the analog input of the myDAQ with **hardware timing**, create an oscilloscope (read **multiple samples** at once). Display the measured waveform on a **Chart**.

Creating a simulated myDAQ

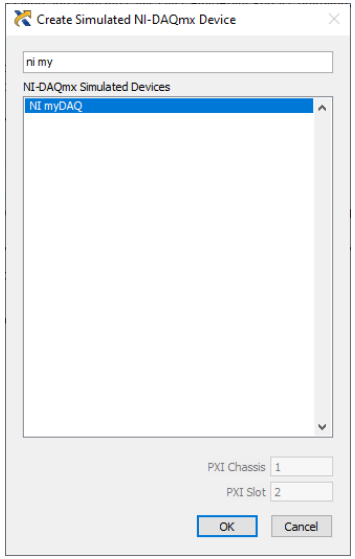
1. Run **NI MAX** (Measurement and Automation Explorer)
2. Right click on **Devices and Interfaces → Create New**…

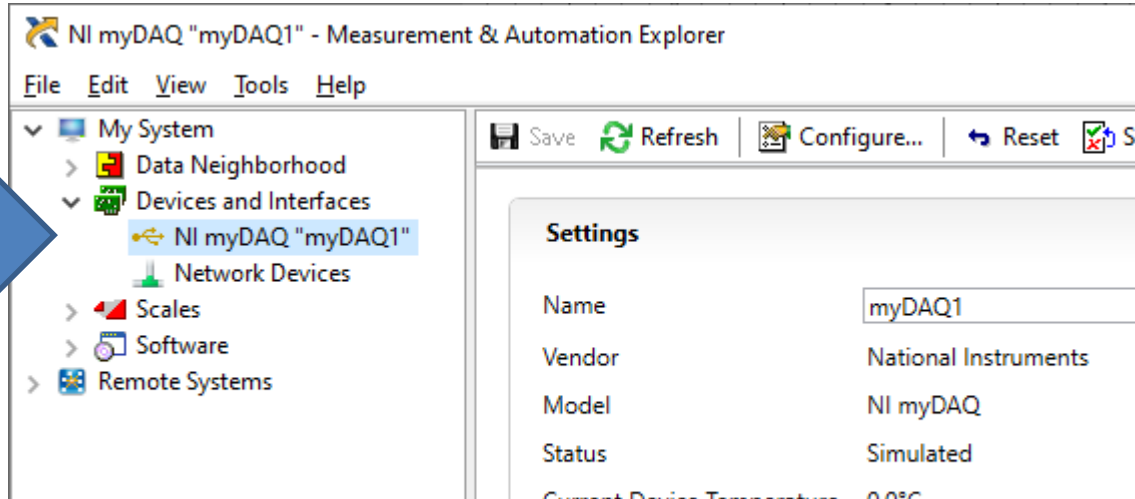


3. Choose type: **Simulated NI-DAQmx Device or Modular Instrument**



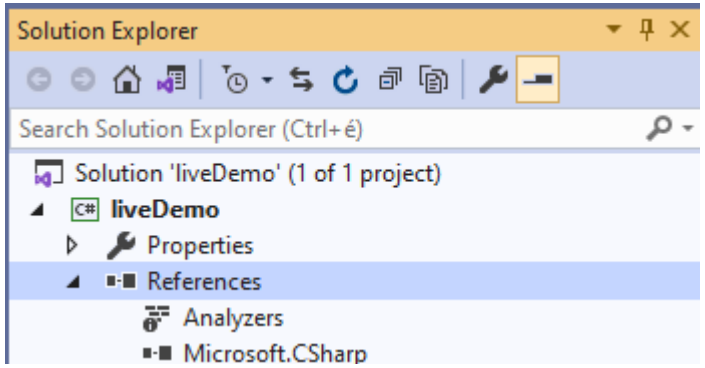
4. Search for myDAQ, select **NI myDAQ**.





Using the .NET library for the myDAQ

1. Start a new project in **Visual Studio**.
2. In the **Solution explorer**, right click on **References → Add reference**.



3. Search for DAQ, select and add **National Instruments DAQmx**.