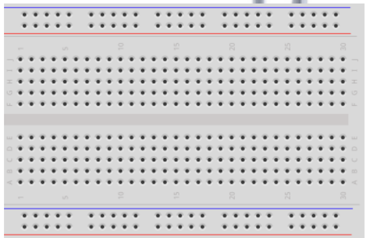
https://www.halvorsen.blog

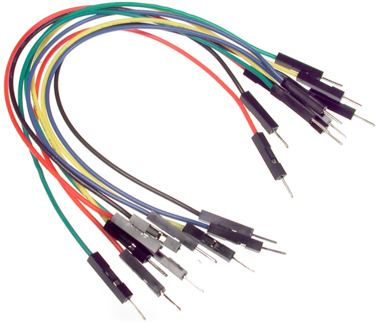
Light Sensor

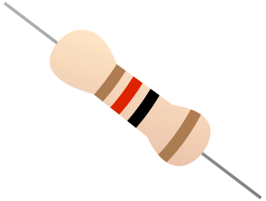
Hans-Petter Halvorsen

Hardware

- DAQ Device (e.g., USB-6008)
- Breadboard
- Light Sensor
- Resistor, $R = 33k\Omega$
- Wires (Jumper Wires)

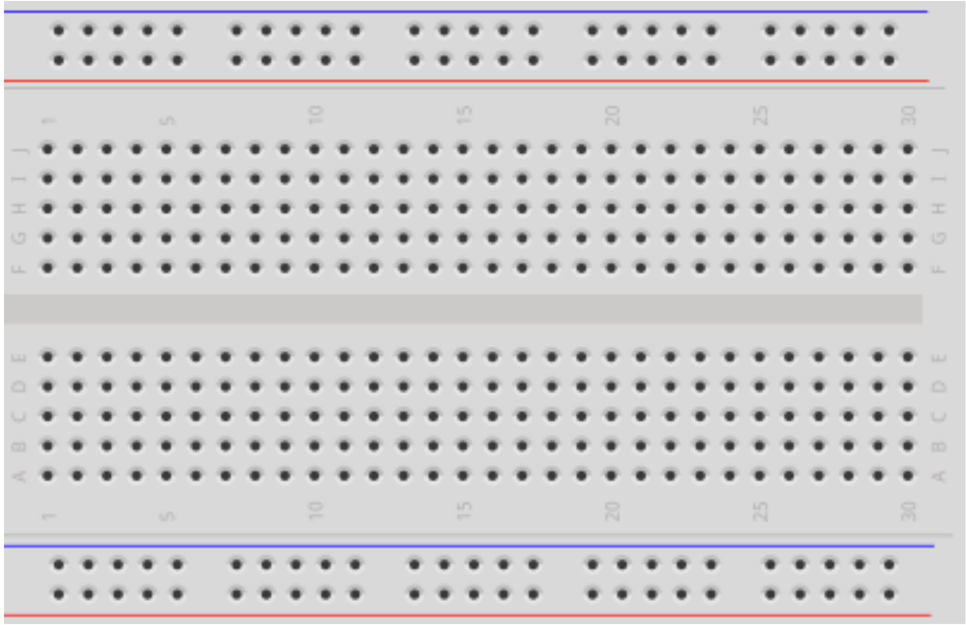


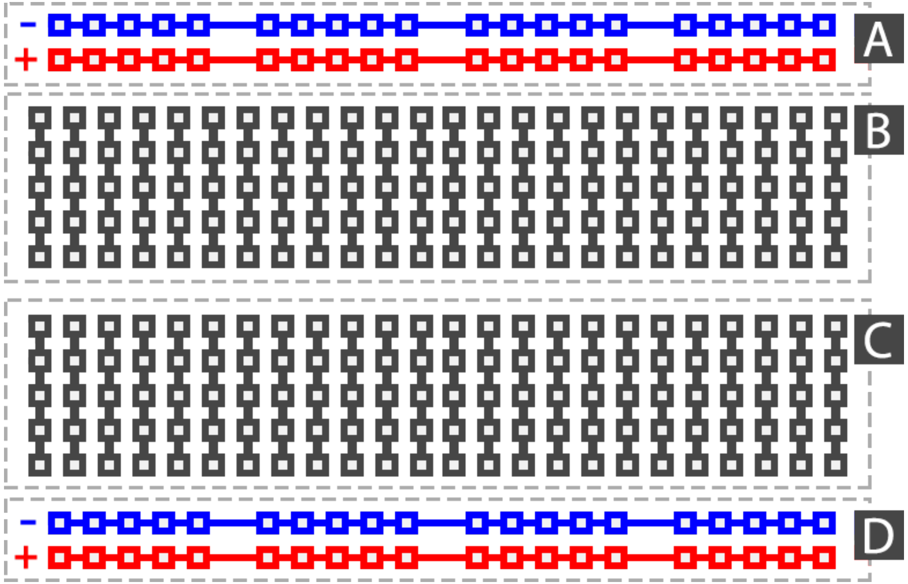




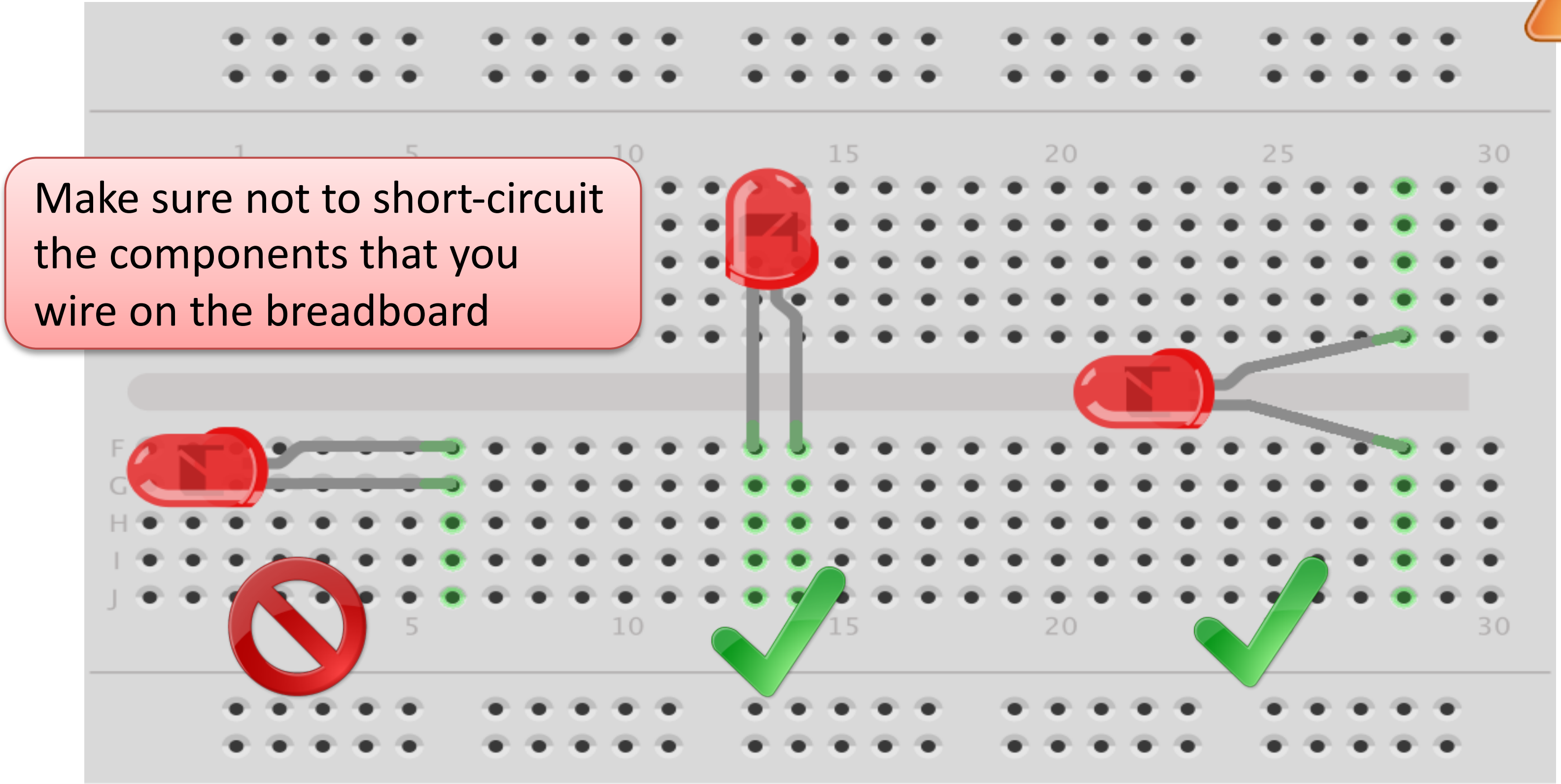
Breadboard

A breadboard is used to wire electric components together





Breadboard Wiring



The Breadboard is used to connect components and electrical circuits

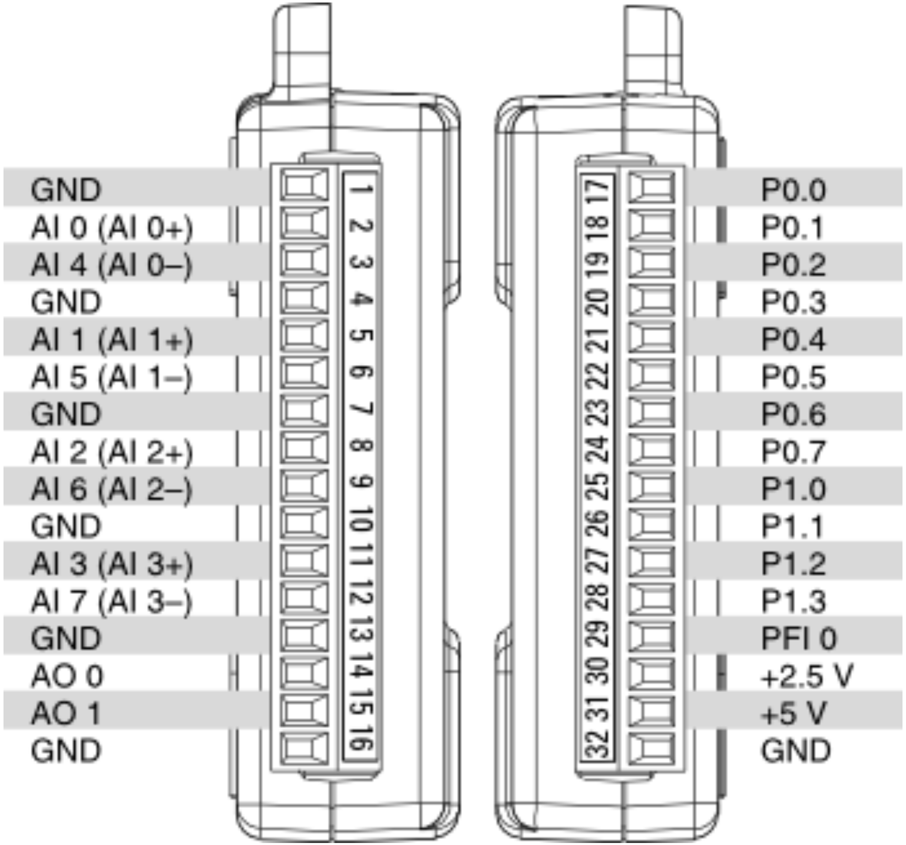
https://www.halvorsen.blog

USB-6008

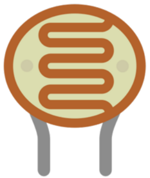
Hans-Petter Halvorsen

USB-6008


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


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

Light Sensor



or

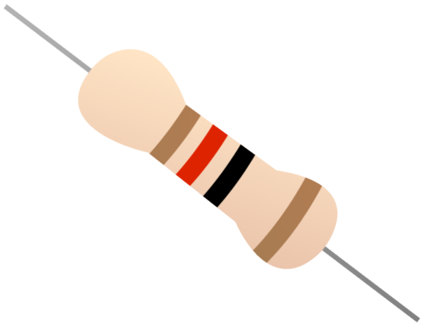
Light sensor, Photocell (Photo resistor), LDR (ligh dependent resistor)

A light sensor / photocell is a sensor used to detect light.

The resistance decreases with increasing light intensity (stronger light).

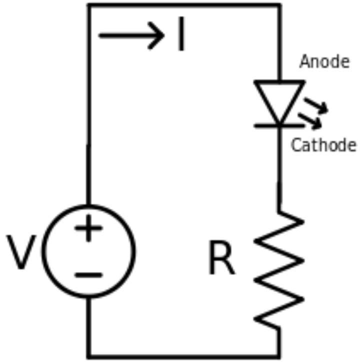
According to Ohms law $U = RI$ the voltage will then get lower when the light gets brighter

Resistors



Resistance is measured in Ohm (Ω)

Resistors comes in many sizes, e.g., 220Ω , 270Ω, 330Ω, 1kΩm 10kΩ, ...

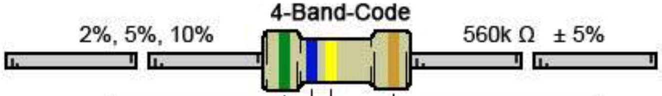


The resistance can be found using Ohms Law
$U = RI$

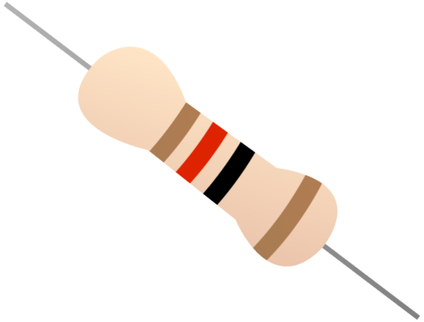
https://en.wikipedia.org/wiki/Resistor

Electrical symbol: 

Resistor Color Codes

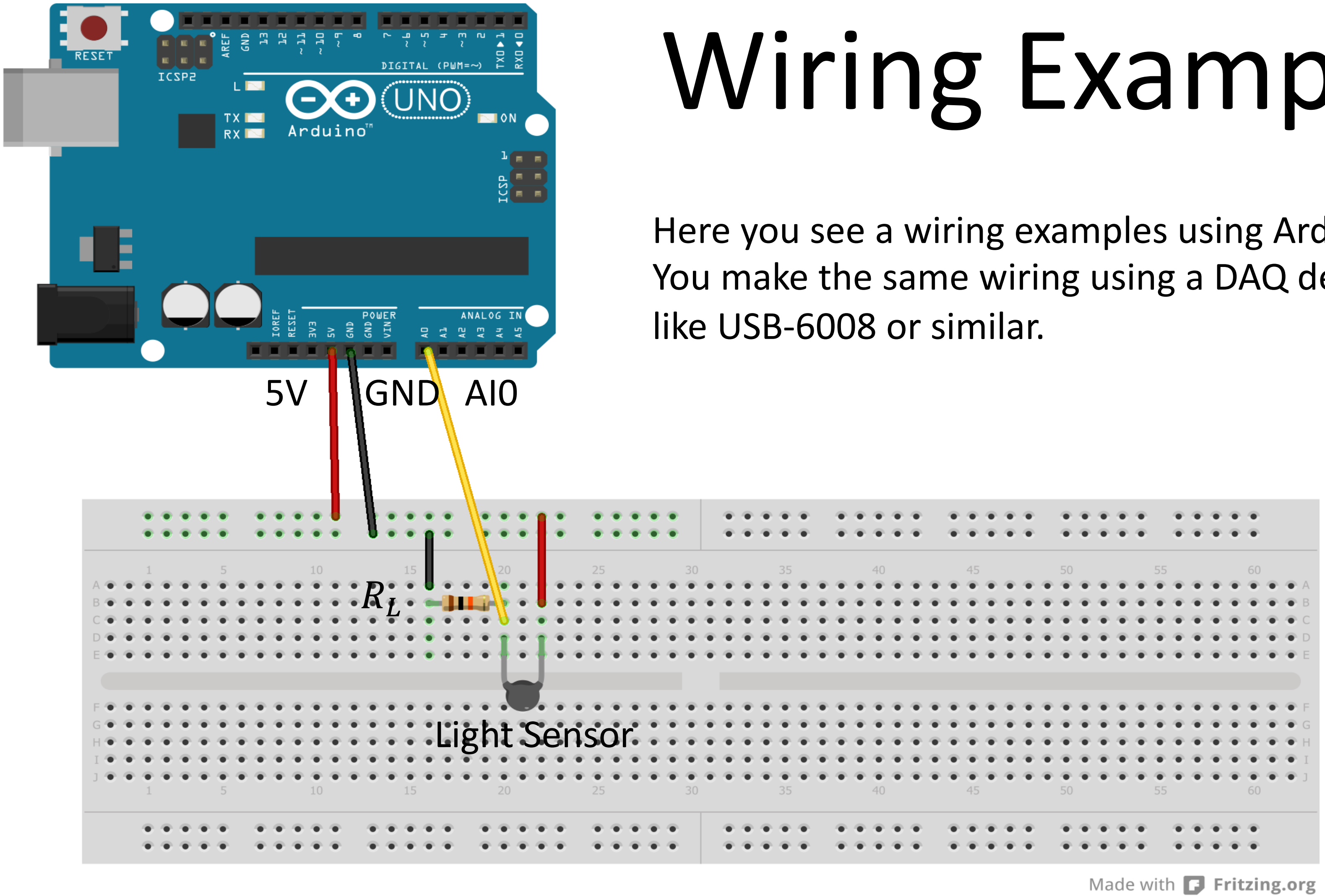


COLOR	1^{ST} BAND	2^{ND} BAND	3^{RD} BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8		± 0.05%
White	9	9	9		
Gold				0.1Ω	± 5% (J)
Silver				0.01Ω	± 10% (K)



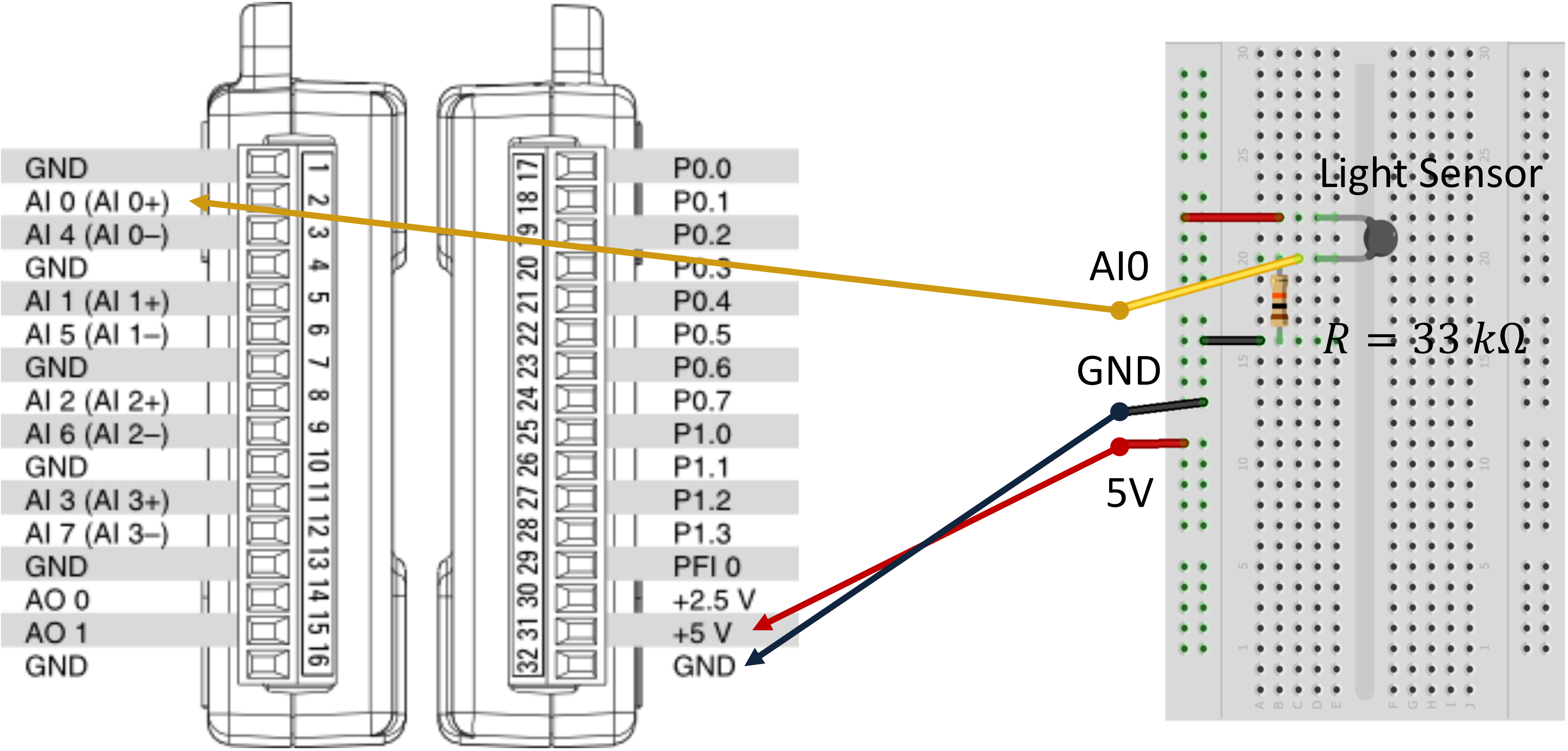
Resistor Calculator: http://www.allaboutcircuits.com/tools/resistor-color-code-calculator/

Wiring Example

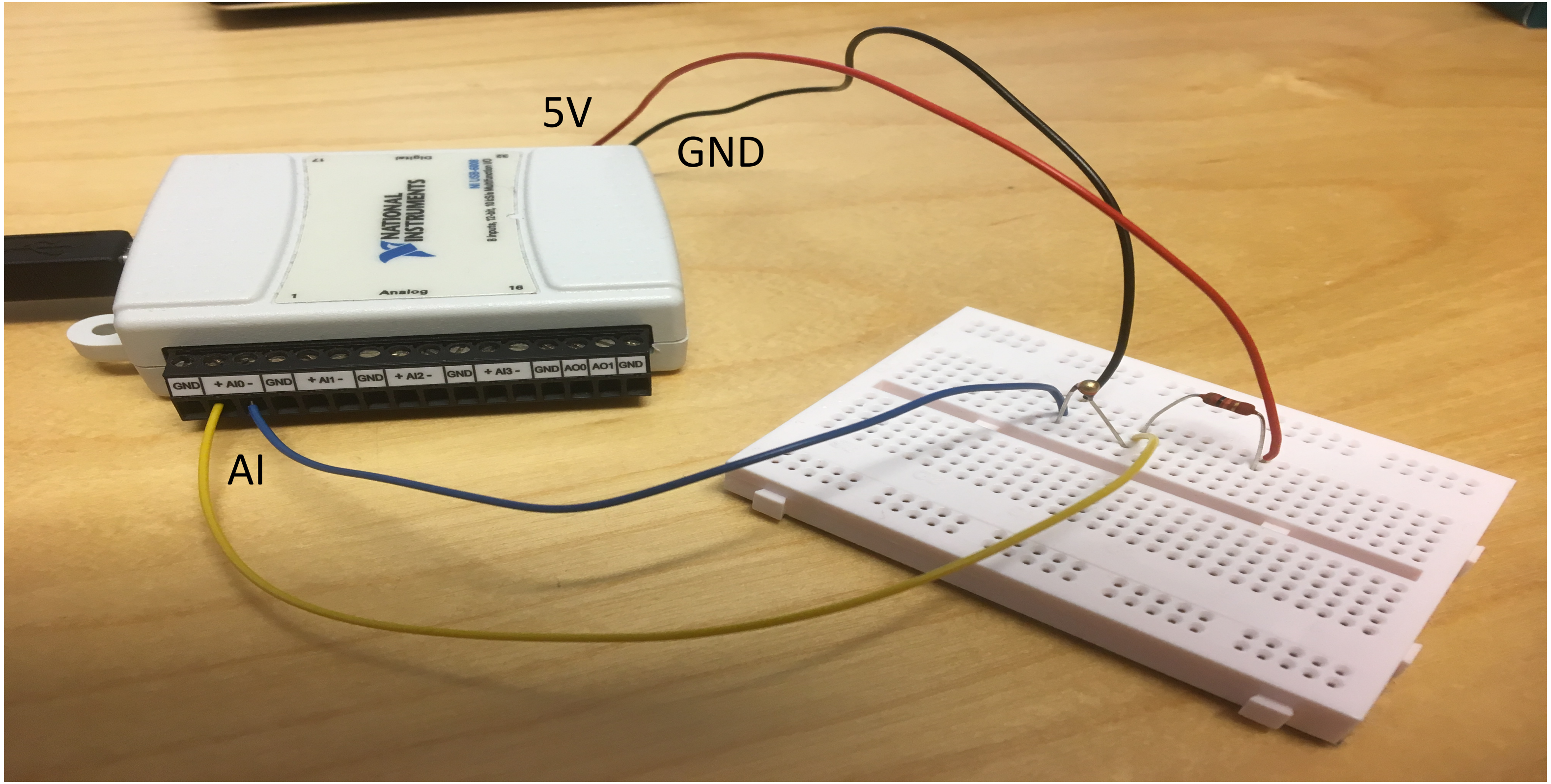


Here you see a wiring examples using Arduino. You make the same wiring using a DAQ device like USB-6008 or similar.

Wiring Example



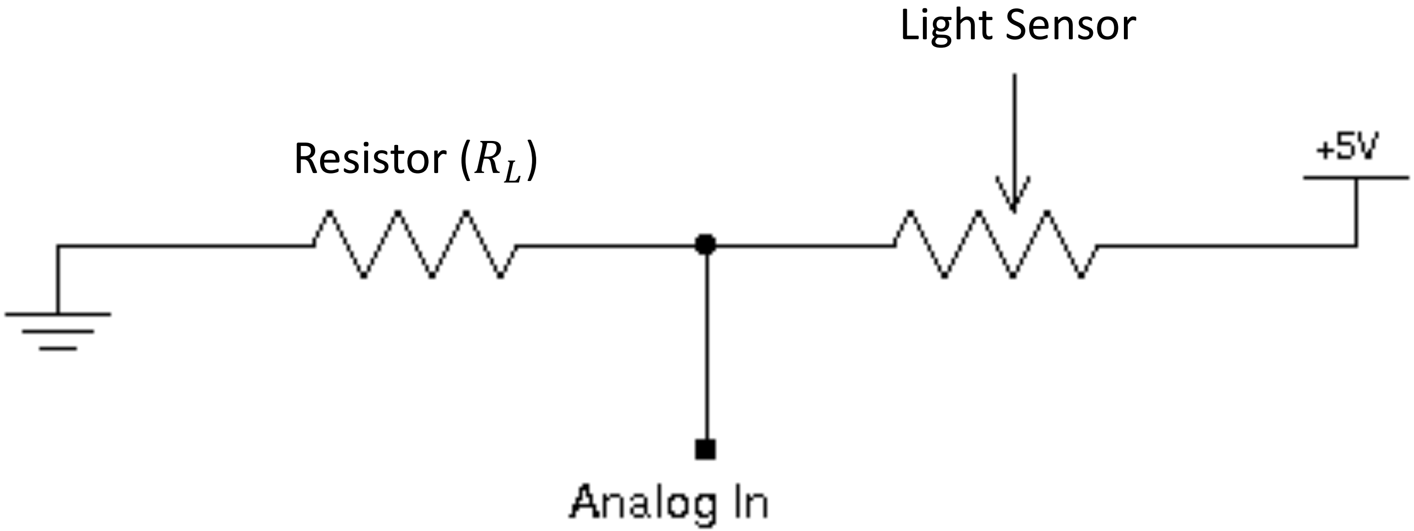
USB-6008 Wiring Example



We connect the Sensor 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.

Wiring

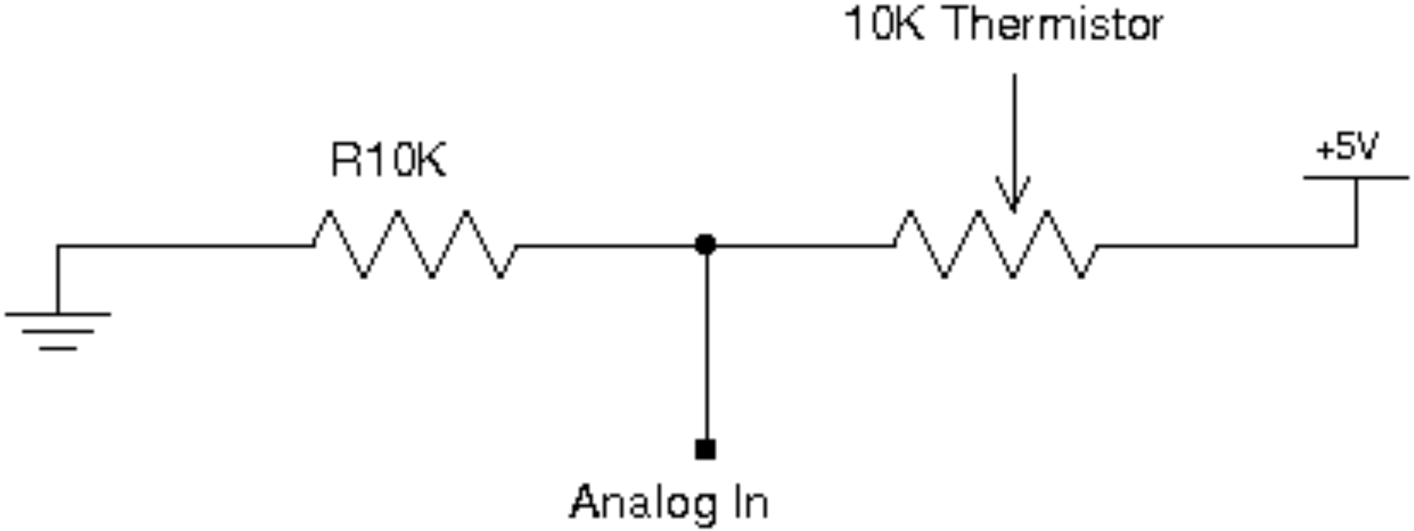
The wiring is called a "Voltage divider":



[https://en.wikipedia.org/wiki/Voltage_divider]

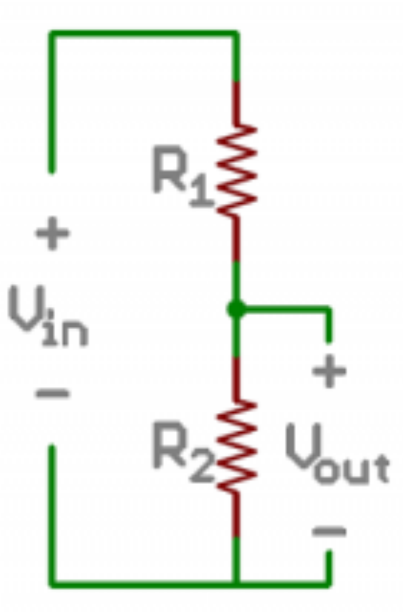
Wiring

The wiring is called a “Voltage divider”:



[https://en.wikipedia.org/wiki/Voltage_divider]

General Voltage Divider



$$V_{out} = V_{in} \cdot \frac{R_2}{R_1 + R_2}$$

https://learn.sparkfun.com/tutorials/voltage-dividers/all

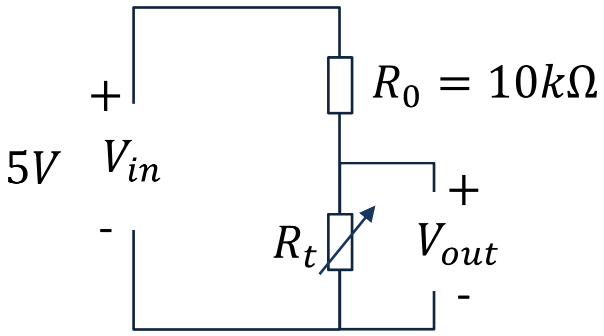
Voltage Divider for our system

Voltage Divider Equation:

$$V_{out} = V_{in} \frac{R_t}{R_0 + R_t}$$

We want to find R_t:

$$R_t = \frac{V_{out} R_0}{V_{in} - V_{out}}$$



R_t - 10k Thermistor. This varies with temperature. From Datasheet we know that $R_t = 10k\Omega$ @25°C

<u>Steps:</u>

1. We wire the circuit on the Breadboard and connect it to the DAQ device
2. We measure V_{out} using the DAQ device
3. We calculate R_t using the Voltage Divider equation
4. Finally, we use Steinhart-Hart equation for finding the Temperature

Lux

Illuminance (lux)	Surfaces illuminated by
0.0001	Moonless, overcast night sky (starlight)[4]
0.002	Moonless clear night sky with airglow[4]
0.05–0.3	Full moon on a clear night[5]
3.4	Dark limit of civil twilight under a clear sky[6]
20–50	Public areas with dark surroundings[7]
50	Family living room lights (Australia, 1998)[8]
80	Office building hallway/toilet lighting[9][10]
100	Very dark overcast day[4]
150	Train station platforms[11]
320–500	Office lighting[8][12][13][14]
400	Sunrise or sunset on a clear day.
1000	Overcast day;[4] typical TV studio lighting
10,000–25,000	Full daylight (not direct sun)[4]
32,000–100,000	Direct sunlight

Design a Luxmeter Using a Light Dependent Resistor: https://www.allaboutcircuits.com/projects/design-a-luxmeter-using-a-light-dependent-resistor/

https://en.wikipedia.org/wiki/Lux

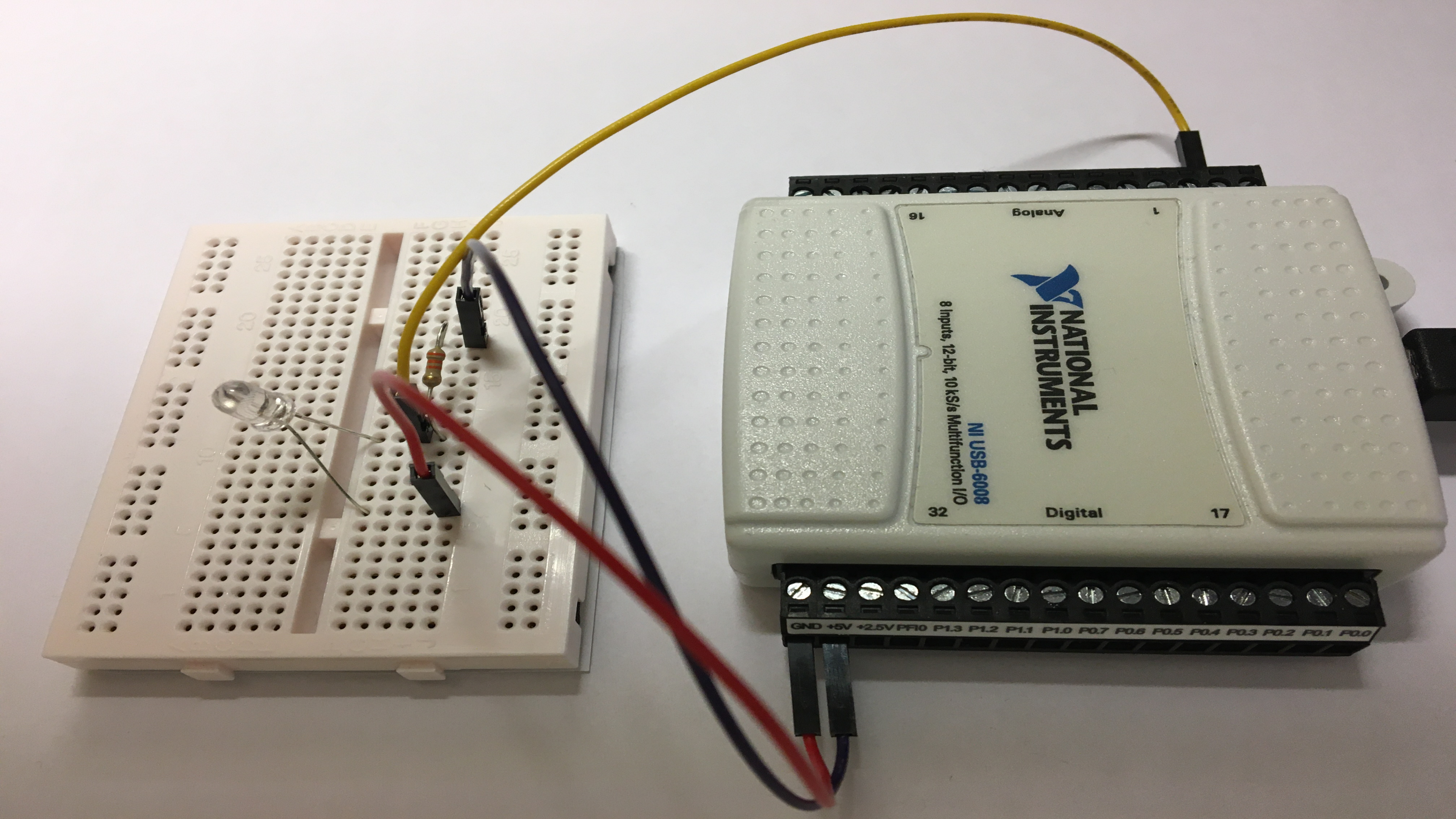
Code

1. Get $\boldsymbol{V_{out}}$ from the DAQ device
2. Calculate $\boldsymbol{R_t} = \frac{V_{out}R_0}{V_{in}-V_{out}}$
3. Find a relationship (a Formula) between R_t and Lux **Lux = f($\boldsymbol{R_t}$)**
4. Calculate the Lux value using your formula **Lux = f($\boldsymbol{R_t}$)**
5. Present the **Lux** value in the User Interface

https://www.halvorsen.blog

Visual Studio

Hans-Petter Halvorsen


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

We will use a **Timer** to read new values at a specific time interval

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

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

                        chart1.Series.Clear();
                        chart1.Series.Add("My Data");
                        chart1.Series["My Data"].ChartType = SeriesChartType.Line;

                        timer1.Interval = 1000;
                        timer1.Start();
                }

                private void timer1_Tick(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();

                        txtLightData.Text = DaqValue.ToString("0.00");

                        chart1.Series["My Data"].Points.AddY(DaqValue);

                }
        }
}
```

```
public Form1()
{
    InitializeComponent();

    chart1.Series.Clear();
    chart1.Series.Add("My Data");
    chart1.Series["My Data"].ChartType = SeriesChartType.Line;

    timer1.Interval = 1000;
    timer1.Start();
}
```

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

private void timer1_Tick(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();

        txtLightData.Text = DaqValue.ToString("0.00");

        chart1.Series["My Data"].Points.AddY(DaqValue);

}
```

Improvements

- Create and use separate **Classes** and in general improve the C# code
- Find a relationship between the voltage signal you read from the DAQ device and **Lux**, which is the official unit for measuring light
  - You can use a Lux measurement device as a reference. You can also download a Lux meter App on your Smart Phone (for free)
- 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