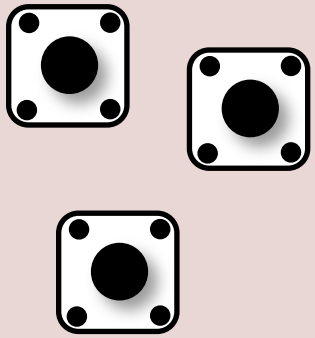
https://www.halvorsen.blog



Push Buttons with Python

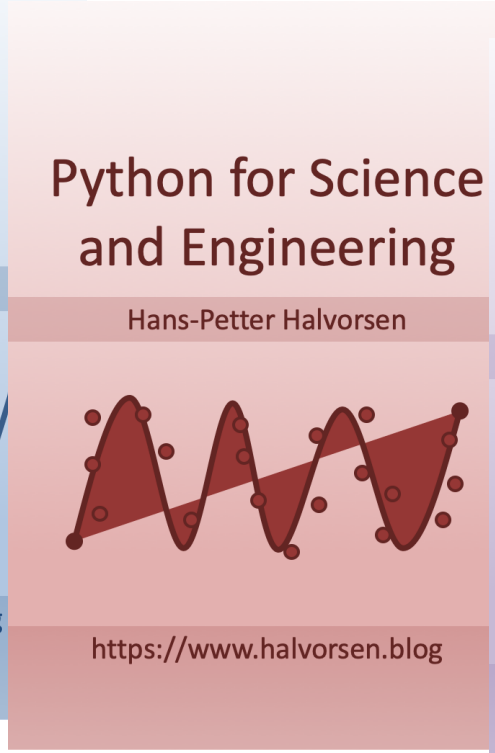
Exemplified by using NI USB-6008 I/O Module

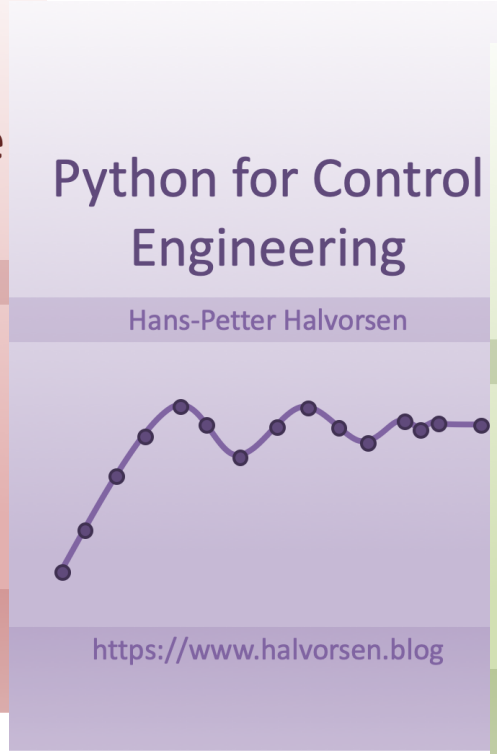
Hans-Petter Halvorsen

Free Textbook with lots of Practical Examples

https://www.halvorsen.blog/documents/programming/python/

Additional Python Resources





https://www.halvorsen.blog/documents/programming/python/

Contents

- DAQ and I/O Modules
- NI-DAQ
- Push Buttons
- Python Examples

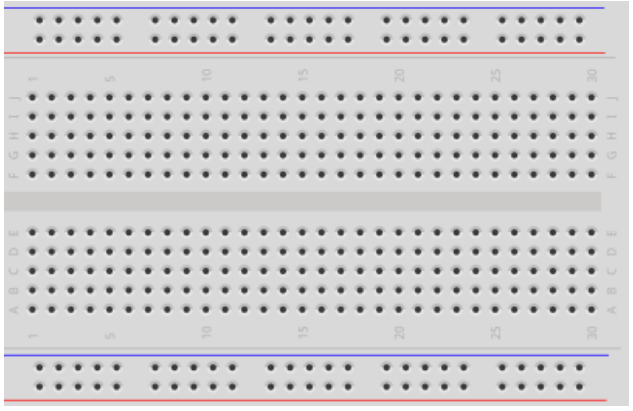
Note! The Python Examples provided will work for all NI-DAQ Devices using the NI-DAQmx Driver, which is several hundreds different types. We will use the NI USB-6008 DAQ Device or I/O Module as an Example

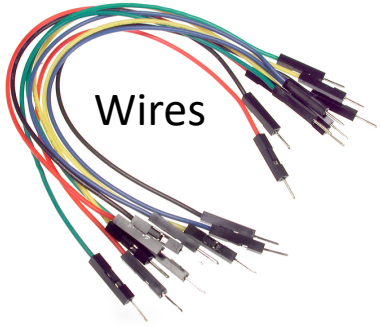
Equipment

USB-6008 (or similar DAQ Device)

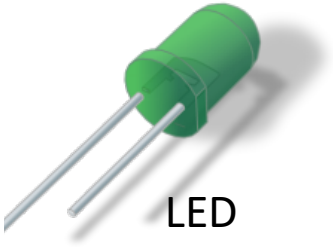
Push Button

Breadboard

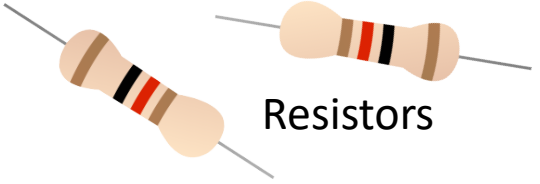




Wires



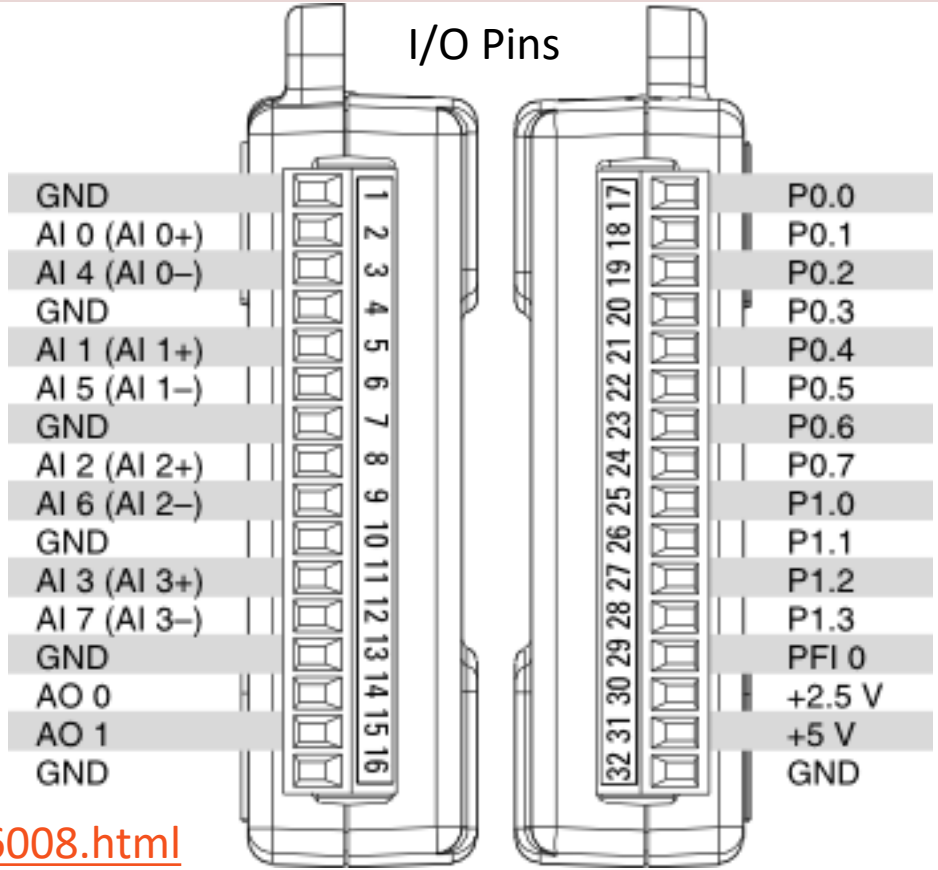
LED



Resistors

NI USB-6008

We will use NI USB-6008 in our examples



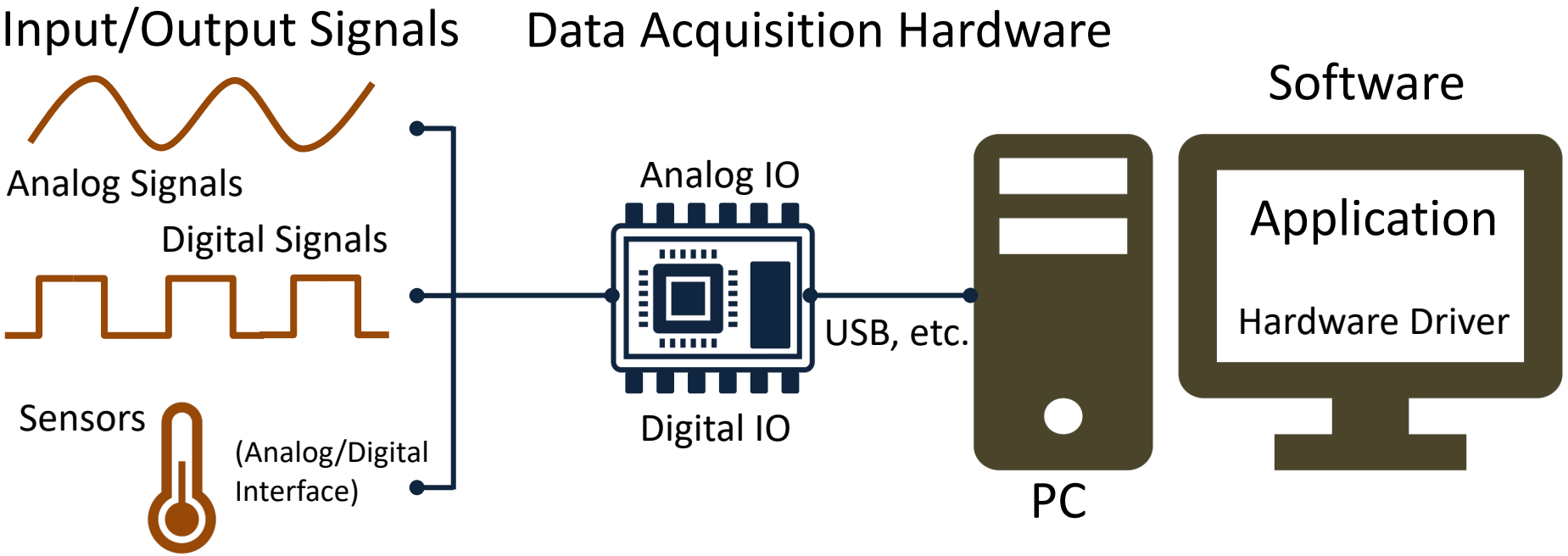
http://www.ni.com/en-no/support/model.usb-6008.html

NI DAQ Device with Python

How to use a NI DAQ Device with Python

Python Application	Your Python Program
nidaqmx Python Package Free	Python Library/API for Communication with NI DAQmx Driver
Python Free	Python Programming Language
NI DAQmx Free	Hardware Driver Software
NI DAQ Hardware	In this Tutorial we will use USB-6008 DAQ Device or I/O Module

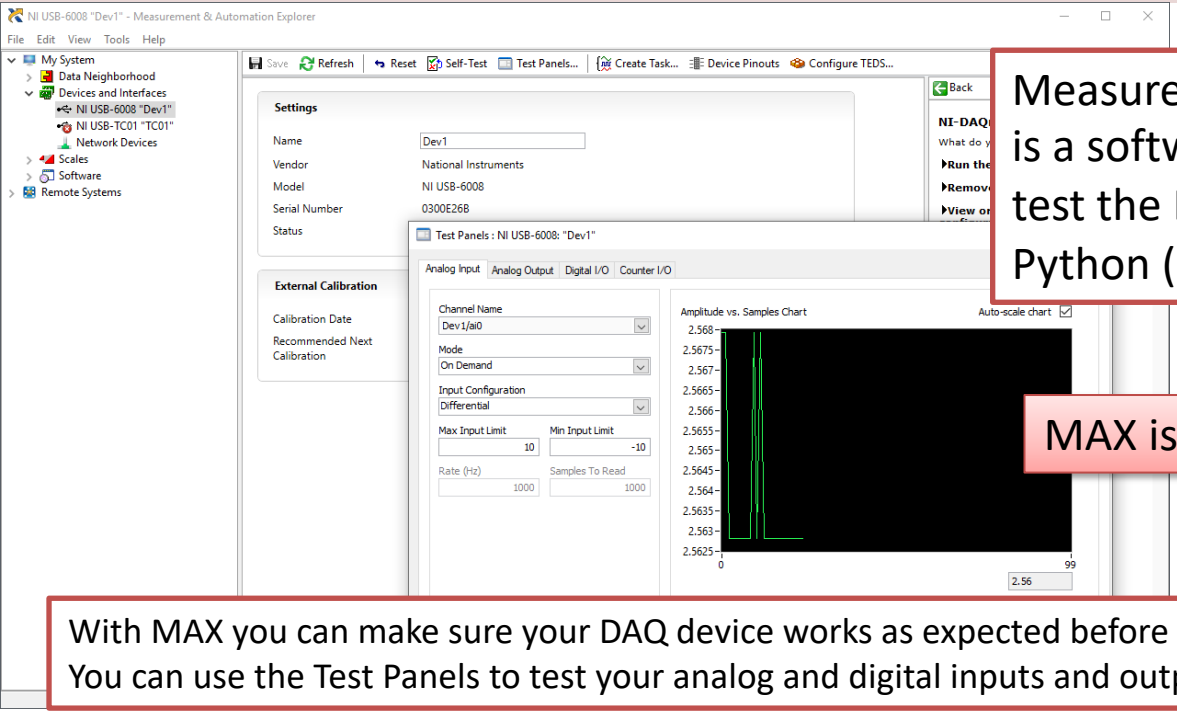
DAQ System


Input/Output Signals
Data Acquisition Hardware
Software
Analog Signals
Digital Signals
Sensors
(Analog/Digital Interface)
Analog IO
Digital IO
USB, etc.
PC
Application
Hardware Driver

NI-DAQmx

- NI-DAQmx is the software you use to communicate with and control your NI data acquisition (DAQ) device.
- NI-DAQmx supports only the Windows operating system.
- Typically you use LabVIEW in combination with NI DAQ Hardware, but the NI-DAQmx can also be used from C, C#, Python, etc.
- The NI-DAQmx Driver is Free!
- Visit the ni.com/downloads to download the latest version of NI-DAQmx

Measurement & Automation Explorer (MAX)



Measurement & Automation Explorer (MAX) is a software you can use to configure and test the DAQ device before you use it in Python (or other programming languages).

MAX is included with NI-DAQmx software

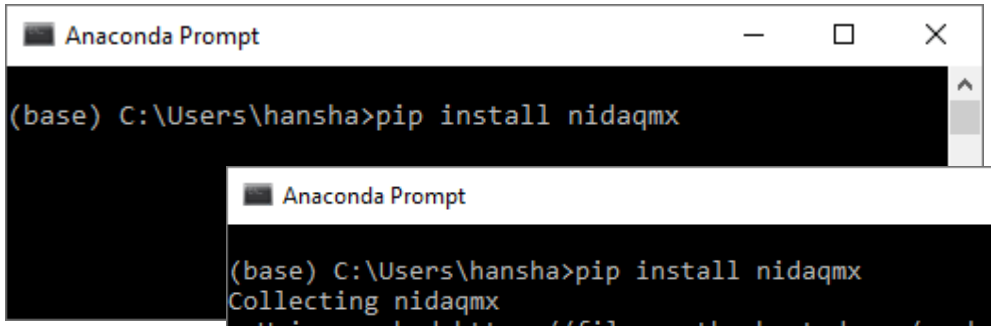
With MAX you can make sure your DAQ device works as expected before you start using it in your Python program. You can use the Test Panels to test your analog and digital inputs and outputs channels.

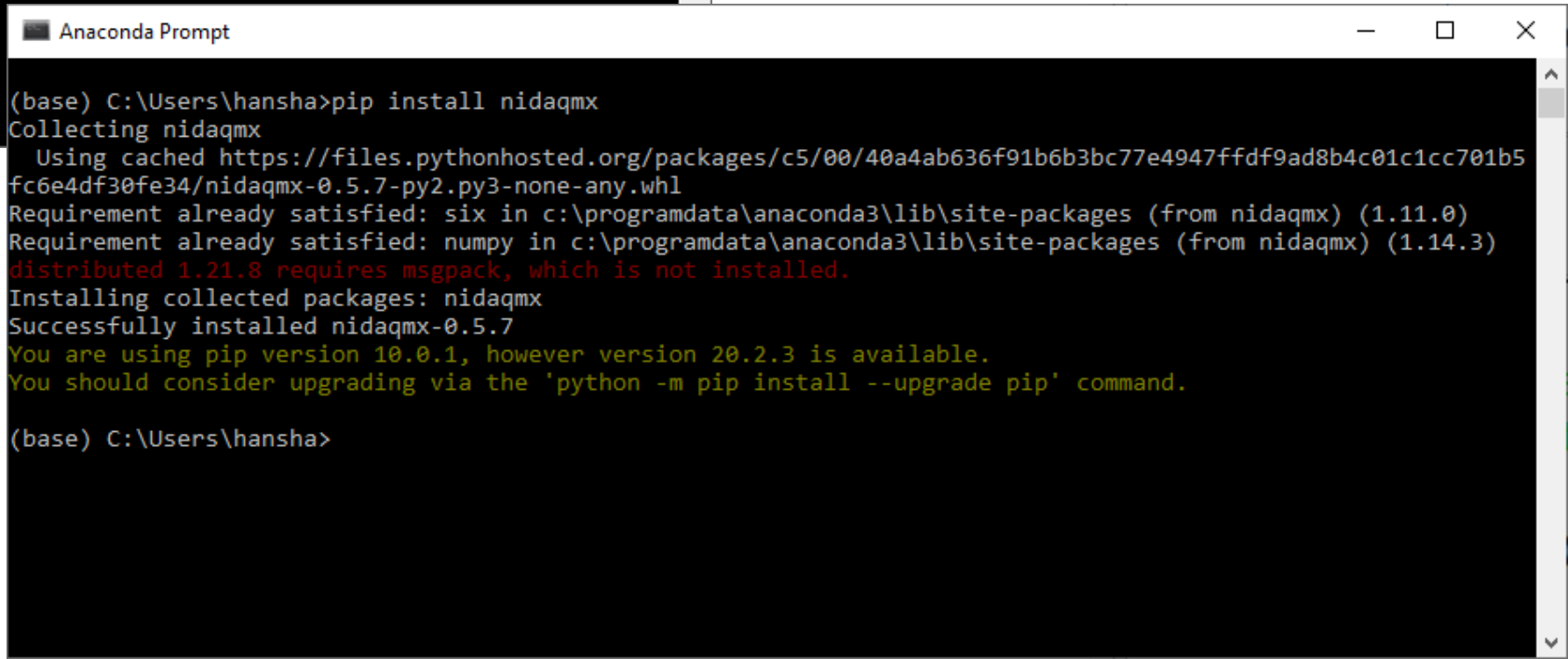
nidaqmx Python API

- Python Library/API for Communication with NI DAQmx Driver
- Running **nidaqmx** requires NI-DAQmx or NI-DAQmx Runtime
- Visit the ni.com/downloads to download the latest version of NI-DAQmx
- nidaqmx can be installed with **pip**:
  `pip install nidaqmx`
- https://github.com/ni/nidaqmx-python

nidaqmx Python Package

Installation



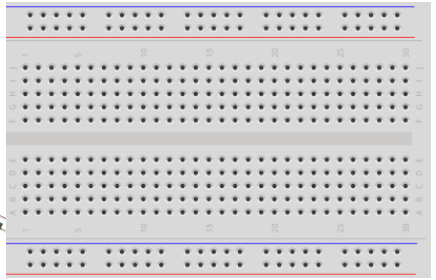


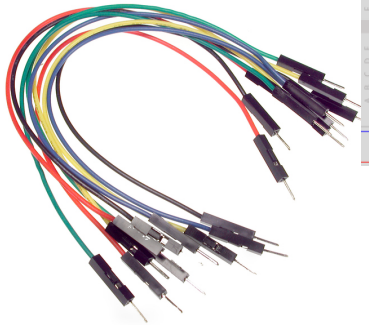
Push Button with Python

Hans-Petter Halvorsen

Necessary Equipment

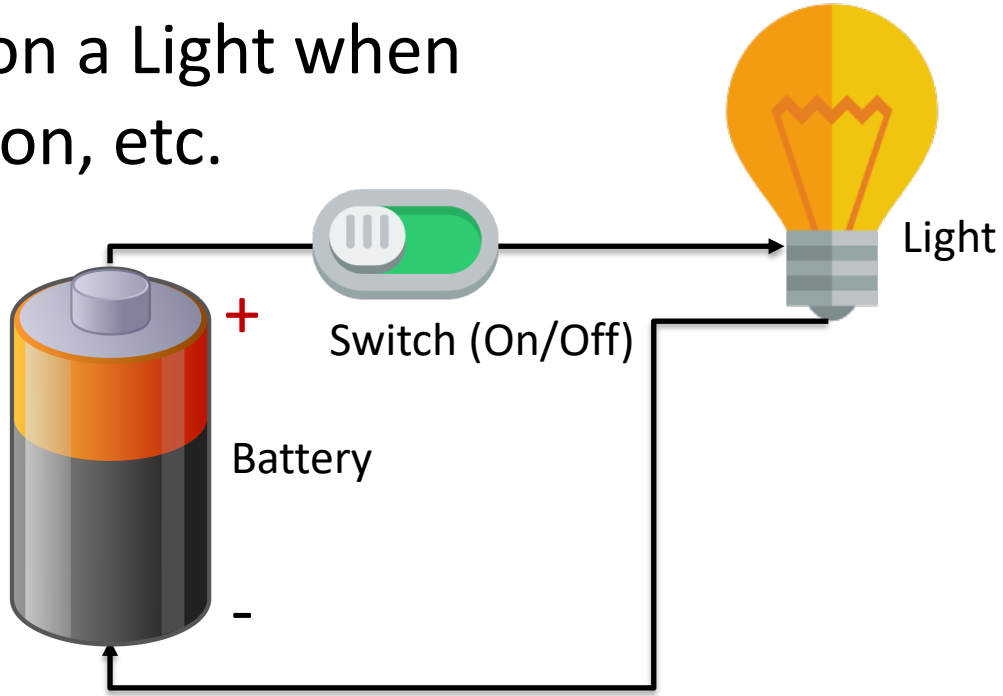
- DAQ Device (e.g., USB-6008)
- Breadboard
- Push Button
- Wires (Jumper Wires)





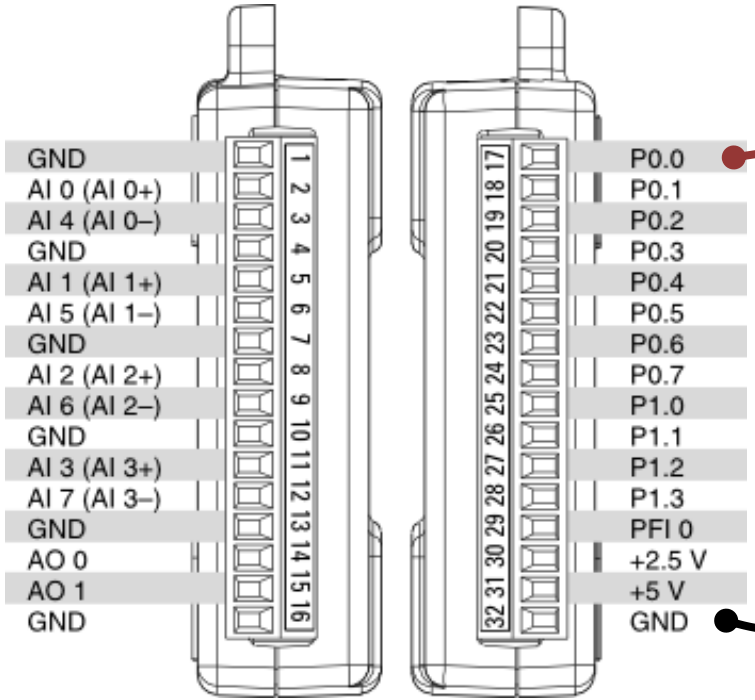
Push Button/Switch

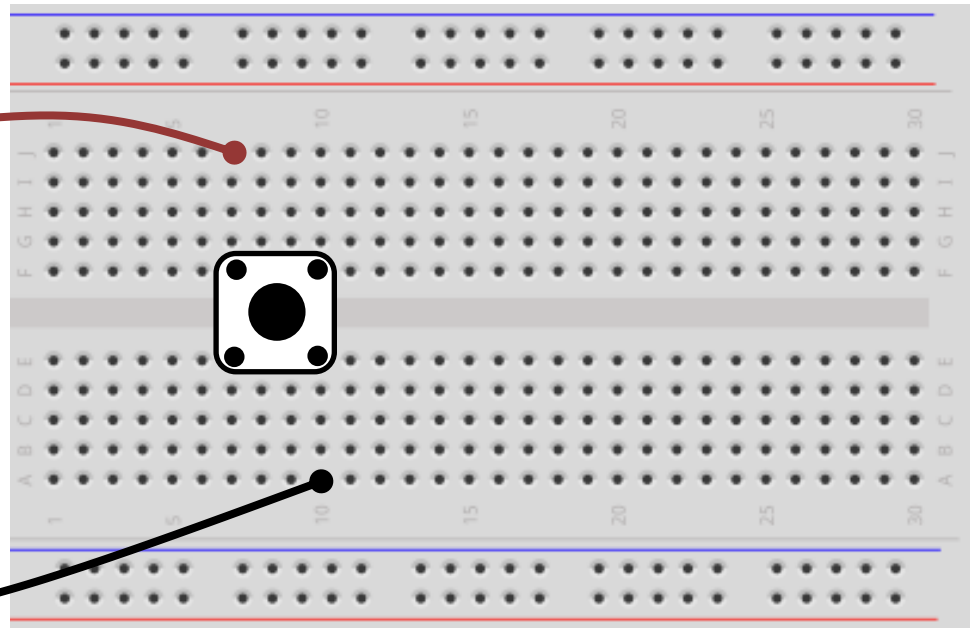
- Pushbuttons or switches connect two points in a circuit when you press them.
- You can use it to turn on a Light when holding down the button, etc.



Hardware Setup

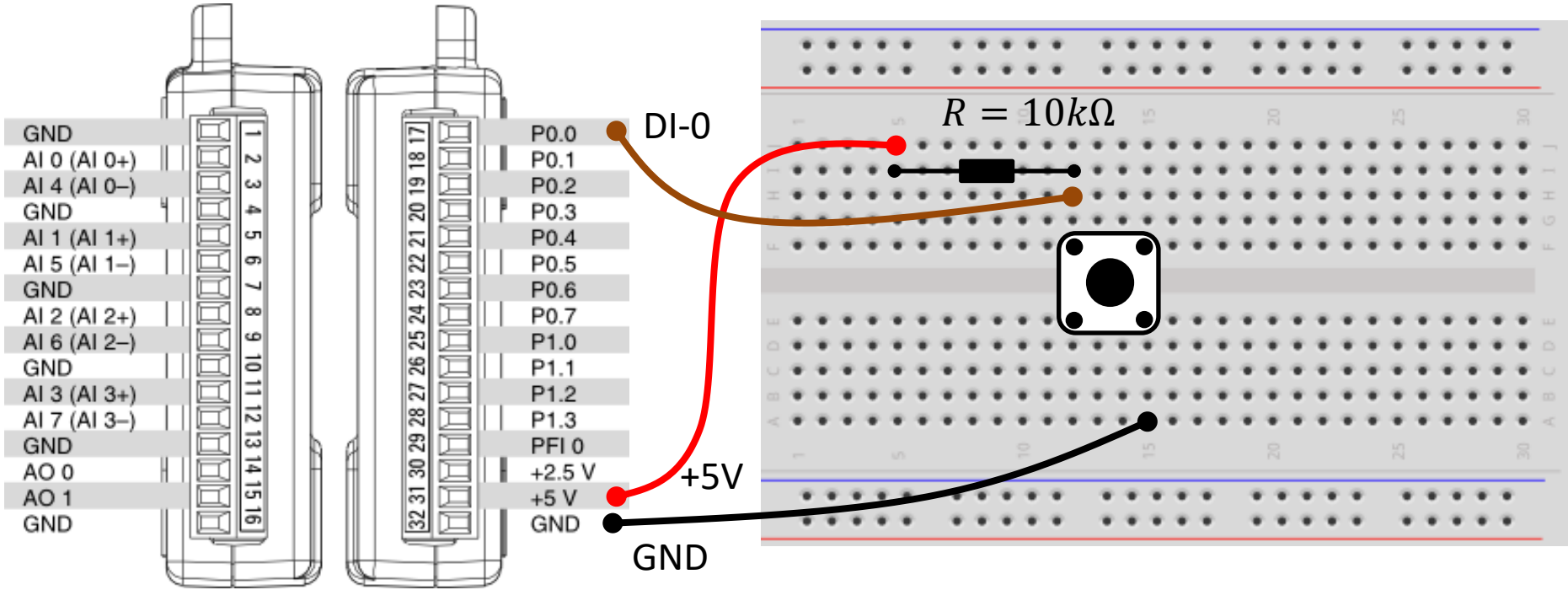
Using built-in 4.7 kΩ **Pull-up Resistor**





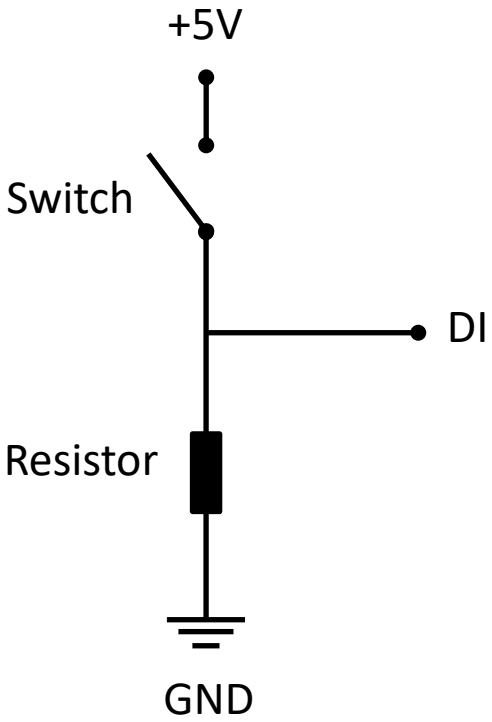
Hardware Setup

Using external Pull-up Resistor

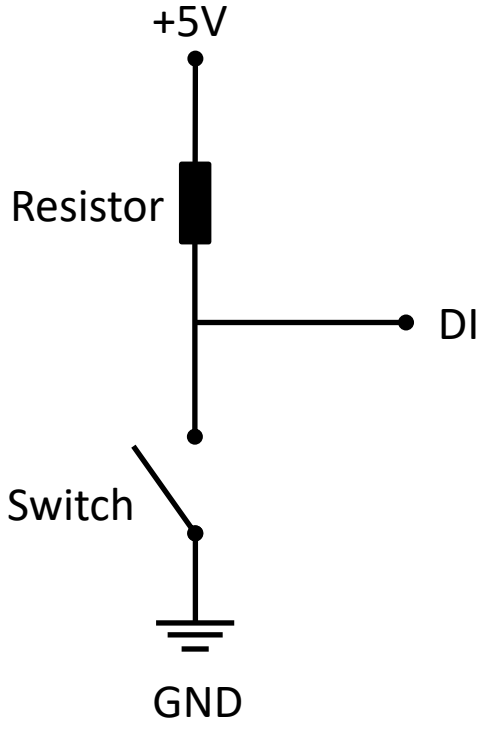


Pull-down/Pull-up Resistor

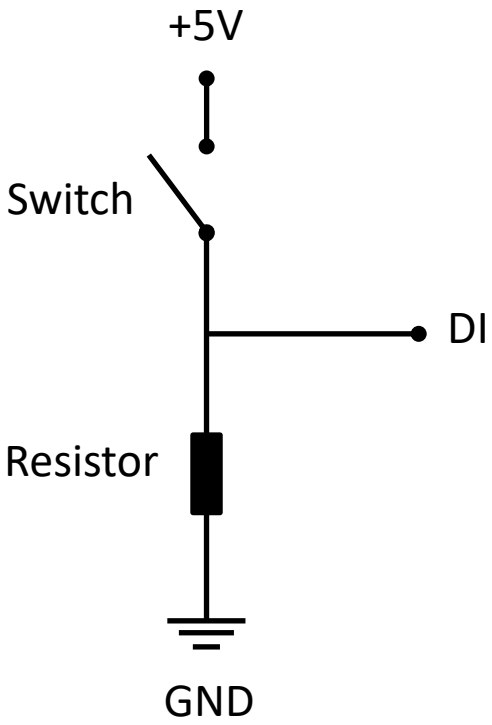
Pull-down Resistor



Pull-up Resistor



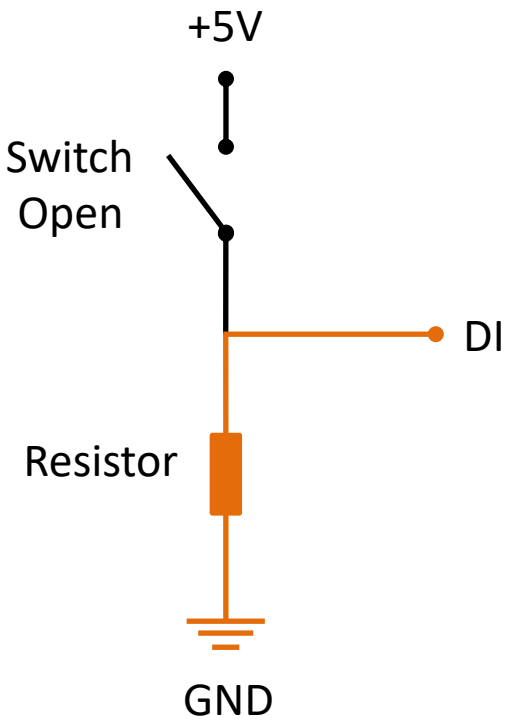
Pull-down Resistor



- When the pushbutton is open (unpressed) there is no connection between the two legs of the pushbutton
- This means the DI pin is connected to ground through the pull-down resistor and we read a **False** (Low).
- When the button is closed (pressed), it makes a connection between its two legs
- This means the DI pin is connected to +5V, so then we read **True** (High).

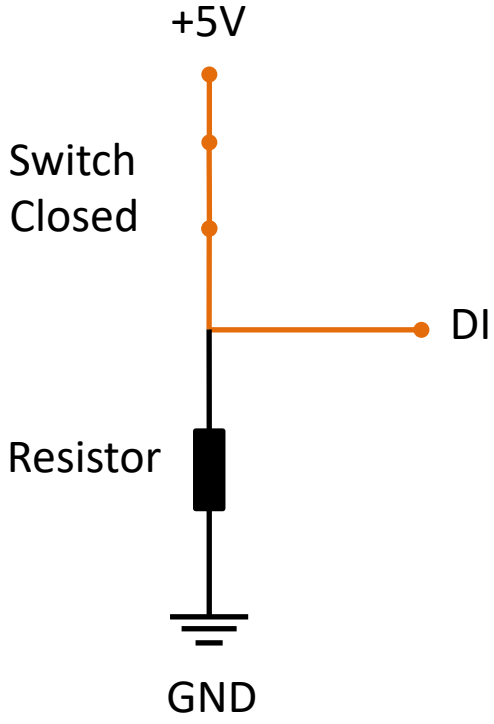
Pull-down Resistor

False/Low

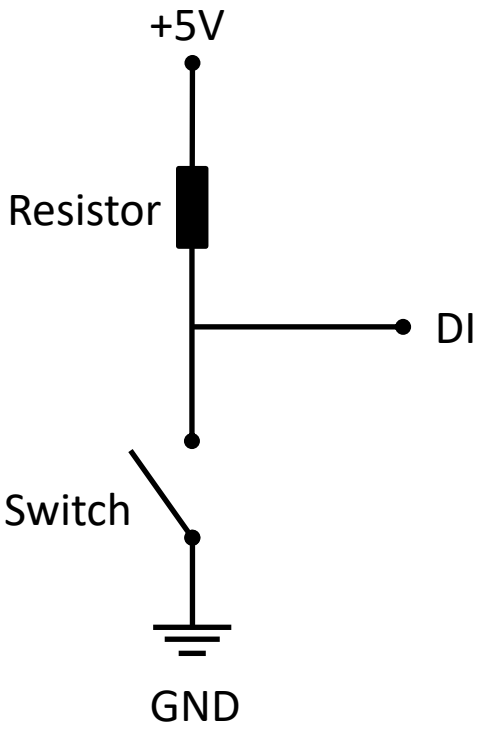


We Push the Button

True/High



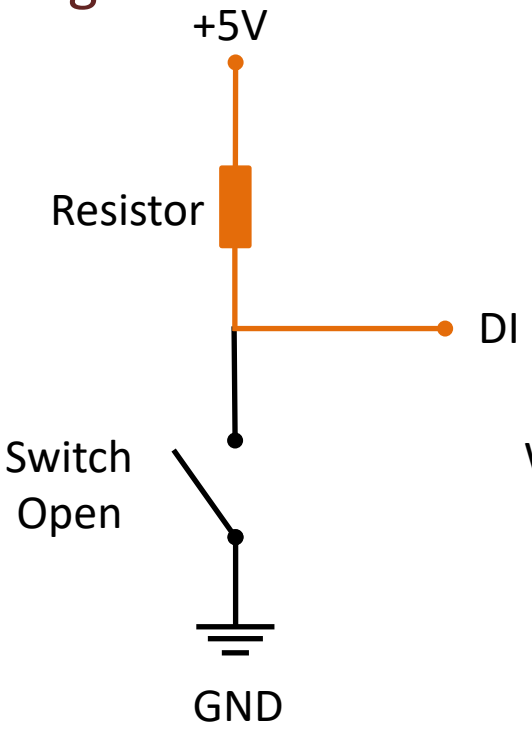
Pull-up Resistor



- When the pushbutton is open (unpressed) there is a connection between 5V and the DI pin.
- This means the default state is **True** (High).
- When the button is closed (pressed), the state goes to **False** (Low).

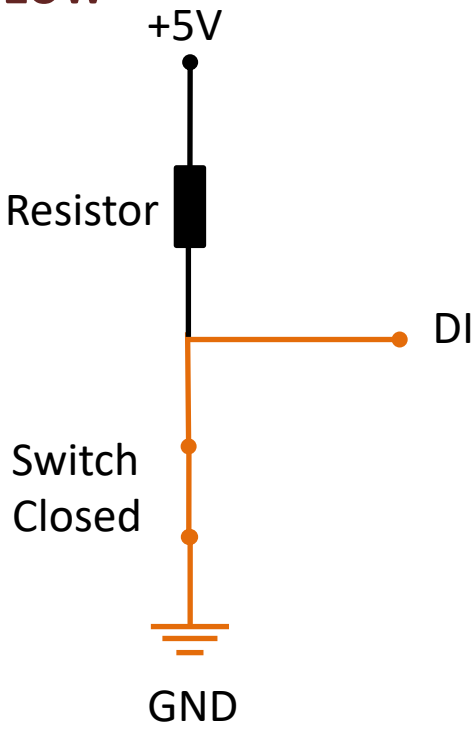
Pull-up Resistor

True/High



We Push the Button

False/Low



Pull-down/Pull-up Resistor

Why do we need a pull-up or pull-down resistor in the circuit?

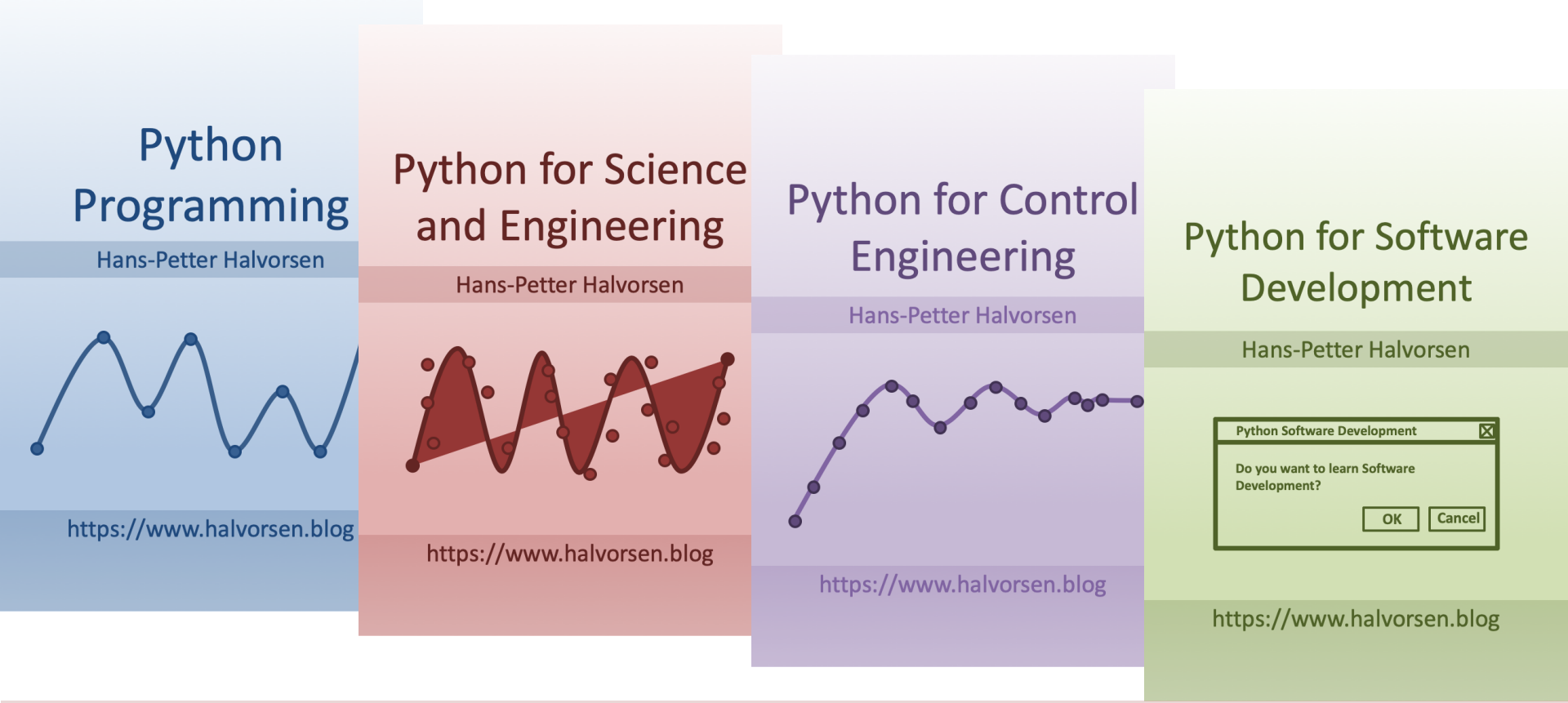
- If you disconnect the digital I/O pin from everything, the LED may blink in an irregular way.
- This is because the input is "floating" - that is, it will randomly return either HIGH or LOW.
- That's why you need a pull-up or pull-down resistor in the circuit.

Python

```
import nidaqmx
import time

task_di = nidaqmx.Task()
task_di.di_channels.add_di_chan("Dev1/port0/line0")
task_di.start()

N = 10
for k in range(N):
    buttonstate = task_di.read()

    if buttonstate != True:
        print("The Button is Pushed")
    else:
        print("Nothing")

    time.sleep(1)

task_di.stop
task_di.close()
```

Here you can do the magic, e.g., turn on a light, an engine or what ever. In this basic example I just print a message to the user.

Additional Python Resources



https://www.halvorsen.blog/documents/programming/python/

Hans-Petter Halvorsen

University of South-Eastern Norway

www.usn.no

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

Web: https://www.halvorsen.blog