LabVIEW DAQmx
Communicate with NI DAQ Devices in LabVIEW

Hans-Petter Halvorsen
Contents

• **Introduction**

• **Practical Examples**
  – **USB-600x DAQ Device**
    • Use DAQ Assistant in LabVIEW
    • Configure DAQ Settings using MAX
    • Use “Low-level” DAQmx VIs in LabVIEW
  – **TC-01 Thermocouple Temperature Device**
    • Use DAQ Assistant in LabVIEW
    • Configure DAQ Settings using MAX
    • Use “Low-level” DAQmx VIs in LabVIEW
Introduction
To use DAQ Hardware in LabVIEW we need to use the DAQmx driver.

We will use the following DAQ hardware to exemplify:
- **TC-01** Thermocouple Temperature Device
- **USB-600x** or similar DAQ Devices with Analog and Digital Channels
  - Note! Other DAQ devices from NI will work in the same manner since they all use the same DAQmx driver.

This Tutorial will show different ways to read data from these DAQ devices.

The features and principles shown here will be the same for other DAQ hardware from other vendors as well.
# USB-600x

Entry-Level, Plug-and-Play USB Data Acquisition

You depend on accurate measurements to make key decisions and discoveries, and NI’s plug-and-play, USB multifunction I/O devices deliver quality measurements at an entry-level price.

Compare NI’s Entry-Level, Stand-Alone Data Acquisition Devices

<table>
<thead>
<tr>
<th></th>
<th>USB-6003</th>
<th>USB-6002</th>
<th>USB-6001</th>
<th>USB-6000</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O Type</td>
<td>AI</td>
<td>AO</td>
<td>DIO</td>
<td>AI</td>
</tr>
<tr>
<td>No. of Channels</td>
<td>4/8</td>
<td>2</td>
<td>13</td>
<td>4/8</td>
</tr>
<tr>
<td>Sample Rate (kS/s and Timed)</td>
<td>100</td>
<td>5</td>
<td>SW</td>
<td>50</td>
</tr>
<tr>
<td>Resolution</td>
<td>16 bits</td>
<td>-</td>
<td>16 bits</td>
<td>-</td>
</tr>
</tbody>
</table>

DAQ and I/O Devices

The Video will focus on USB-600x and Analog Input (AI). The programming principles are the same for both devices (and other similar devices from NI) and the different Channel options. The Tutorial (PDF) will show examples using both devices.

I/O Devices have typically 4 different types of channels:
- Analog In (AI)
- Analog Out (AO)
- Digital In (DI)
- Digital Out (DO)
To use DAQ hardware in LabVIEW we need to use the DAQmx driver. It can be downloaded for free.

You can use MAX to test and configure your DAQ device.

Measurement and Automation Explorer (MAX) is installed as part of the DAQmx software.
DAQmx in LabVIEW

To use DAQ hardware in LabVIEW we need to use the DAQmx driver. It can be downloaded for free.

Different options using DAQmx

You have different options when setting up and connecting the DAQ device using DAQmx and LabVIEW:

1. Use the “DAQ Assistant” in LabVIEW (Configuration through a Wizard)
2. Configure DAQ Settings using MAX
3. Use the “Low-level” DAQmx VIs in LabVIEW (Full control of all details in your code)
Practical LabVIEW Examples

Table of Contents
LabVIEW Examples

You have different options when setting up and connecting the DAQ device using DAQmx and LabVIEW:

1. Use the “DAQ Assistant” in LabVIEW (Configuration through a Wizard)
2. Configure DAQ Settings using MAX
3. Use the “Low-level” DAQmx VIs in LabVIEW (Full control of all details in your code)
USB-600x
I/O Devices have typically 4 types of channels:

- Analog In (AI)
- Analog Out (AO)
- Digital In (DI)
- Digital Out (DO)

This Tutorial will focus on Analog In. The programming principles are the same for the other types.
Open MAX and make sure the USB-600x is installed properly. Here you can also test to see if the device is working properly.
DAQ Assistant

USB-600x

Hans-Petter Halvorsen
Different options using DAQmx

You have different options when setting up and connecting the DAQ device using DAQmx and LabVIEW:

1. Use the “DAQ Assistant” in LabVIEW (Configuration through a Wizard)
2. Configure DAQ Settings using MAX
3. Use the “Low-level” DAQmx VIs in LabVIEW (Full control of all details in your code)
Here, a 1.5V Battery is connected to Analog Input Channel 0 (AI0)
DAQ Assistant

Select the measurement type for the task.

A task is a collection of one or more virtual channels with timing, triggering, and other properties. To have multiple measurement types within a single task, you must first create the task with one measurement type. After you create the task, click the Add Channels button to add a new measurement type to the task.

Select the physical channel(s) to add to the task.

If you have previously configured global virtual channels of the same measurement type as the task, click the Virtual tab to add or copy global virtual channels to the task. When you copy the global virtual channel to the task, it becomes a local virtual channel. When you add a global virtual channel to the task, the task uses the actual global virtual channel, and any changes to that global virtual channel are reflected in the task.

If you have TEDS configured, click the TEDS tab to add TEDS channels to the task.

For hardware that supports multiple channels in a task, you can select multiple channels to

Configure the task.

Communication Mode
Select "Serial" for communication with the device.

Timing Settings
Acquisition Mode: Use "4 Sample (On Demand)" for the acquisition mode.

Voltage Input Setup
Signal Input Range:
Max: 5
Min: 0

Terminal Configuration
Differential

Samples to Read: Use values to read.
Rate (Hz): Use rate to read.

Finish, Cancel
Do you get an Error like this when trying to use the DAQ Assistant?

In “LabVIEW 2022 Q3” to “LabVIEW 2023 Q3” there is a bug with the DAQ Assistant. **To fix that you need to install the LabVIEW 2021 SP1 core component.** See information on this web page from NI:

https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q0000019gTMCAY&l=en-NO

Another solution is to use, e.g., the lower level DAQ functions in LabVIEW, see upcoming examples in this Tutorial.

I have been in contact with NI, and it is expected that this error will be fixed in upcoming versions of LabVIEW

This Bug/Issue has been fixed in LabVIEW 2024 Q1 and NI-DAQmx 2024 Q1 (and newer)
Convert from Dynamic Data
While Loop
Number of Decimals and Units

Make sure to select a proper **Number of Decimals**. Typically, a DAQ Device/Sensor don’t have an Accuracy with 10 Decimals. Make sure to read the Datasheet for the selected Hardware.

Always make sure to add a proper **Unit**. A Value without a Unit makes no Sense!
Reading Multiple Channels
Reading Multiple Channels
Configure DAQ Settings using MAX USB-600x
Different options using DAQmx

You have different options when setting up and connecting the DAQ device using DAQmx and LabVIEW:

1. Use the “DAQ Assistant” in LabVIEW (Configuration through a Wizard)
2. Configure DAQ Settings using MAX
3. Use the “Low-level” DAQmx VIs in LabVIEW (Full control of all details in your code)
Configure DAQ Settings using MAX
Configure DAQ Settings using MAX

Measuring Voltage

Most measurement devices are designed for measuring, or reading, voltage. The common voltages that are measured are DC and AC.

DC voltages are useful for measuring phenomena that can change slightly over time, such as temperature, pressure, or strain.

AC voltages, on the other hand, are waveforms that constantly increase, decrease, and reverse polarity. Most powerlines deliver AC voltage.

Depending on your configuration, some of these timing modes may not be available.

1 Sample (On Demand) specifies that the task acquire or generate one sample.
1 Sample (HW Timed) specifies that the task acquire or generate one sample on an edge of a hardware clock.

Note: SignalExpress does not support 1 Sample (HW Timed) mode.

N Samples specifies that the task acquire or generate a finite number of samples, specified by Samples To Read/Write.
Configure DAQ Settings using MAX
While Loop

- **Block Diagram**: The diagram shows a while loop with a task named `MyVoltageTask` that reads analog data from channel 1 with 1 sample. The loop waits for 1000 ms and stops when a stop button is pressed. The analog data is displayed on the front panel.

- **Front Panel**: The analog value `A10 [V]` is currently set to 1.6.
While Loop v2 – Start/Stop Task

**DAQmx Start Task (VI)**

Owning Palette: DAQmx - Data Acquisition VIs and Functions

Installed With: NI-DAQmx

Transitions the task to the running state to begin the measurement or generation. Using this VI is required for some applications and is optional for others. If you do not use this VI, a measurement task starts automatically when the DAQmx Read VI runs. The **autostart** input of the DAQmx Write VI determines if a generation task starts automatically when the DAQmx Write VI runs.

If you do not use the DAQmx Start Task VI and the DAQmx Stop Task VI when you use the DAQmx Read VI or the DAQmx Write VI multiple times, such as in a loop, the task starts and stops repeatedly. Starting and stopping a task repeatedly reduces the performance of the application.

```
task/channels in  --------> DAQmx  --------> task out
    error in  |       |       |    error out
```

Increase speed by using Start Task VI
While Loop v2 – Start/Stop Task
Using “Low-level” DAQmx VIs

USB-600x
Different options using DAQmx

You have different options when setting up and connecting the DAQ device using DAQmx and LabVIEW:

1. Use the “DAQ Assistant” in LabVIEW (Configuration through a Wizard)
2. Configure DAQ Settings using MAX
3. Use the “Low-level” DAQmx VIs in LabVIEW (Full control of all details in your code)
Using “Low-level” DAQmx VIs
In the previous example we just used the default setting. If you need to change some of the default setting, just right-click on the select input and create a constant.
Configure Additional Settings
While Loop
While Loop with Start/Stop Task

Increase **speed** by using “Start Task” and “Stop Task” VIs outside the While Loop
Final Application with Error Handling

All Applications should have proper Error Handling

Further Improvements: Use the **State Machine** programming principle in your Application
Improved Solution: Create and use SubVIs

See Code for the 2 SubVIs on next page
Reading Multiple Channels
Can be put into a SubVI

Further Improvements: Use the **State Machine** programming principle in your Application
TC-01 Thermocouple
Open MAX and make sure the TC-01 is installed Properly. Here you can also test to see if the device is working properly.
DAQ Assistant

TC-01 Thermocouple
DAQ Assistant
The TC-01 device has only one Analog Input called ai0.

Select CJC Source = “Built-in”
Do you get an Error like this when trying to use the DAQ Assistant?

In “LabVIEW 2022 Q3” to “LabVIEW 2023 Q3” there is a bug with the DAQ Assistant. To **fix that you need to install the **LabVIEW 2021 SP1 core component**. See information on this web page from NI:

https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q0000019gTMCAY&l=en-NO

Another solution is to use, e.g., the lower level DAQ functions in LabVIEW, see upcoming examples in this Tutorial.

Workaround:
- Install LabVIEW 2021 SP1
- LabVIEW 2021 SP1 does not need to be activated, it is sufficient to install LabVIEW 2021 on your system and not activate it
- Installing just the LabVIEW 2021 SP1 run-time engine will not be sufficient, the LabVIEW ADE needs to be installed.

I have been in contact with NI, and it is expected that this error will be fixed in upcoming versions of LabVIEW

This Bug/Issue has been fixed in LabVIEW 2024 Q1 and NI-DAQmx 2024 Q1 (and newer)
Convert from Dynamic Data
While Loop

Typically, you log Data inside a While Loop
Number of Decimals and Units

Make sure to select a proper **Number of Decimals**. Typically, a Temperature Sensor don’t have an Accuracy with 10 Decimals. Make sure to read the Datasheet for the selected Sensor.

Always make sure to add a proper **Unit**. A Value without a Unit makes no Sense!
Configure DAQ Settings using MAX TC-01 Thermocouple

Table of Contents

Hans-Petter Halvorsen
Configure DAQ Settings using MAX
Configure DAQ Settings using MAX
Configure DAQ Settings using MAX
While Loop
Increase **speed** by using “Start Task” and “Stop Task” VIs outside the While Loop.
While Loop v2 – Start/Stop Task
Using “Low-level” DAQmx VIs

TC-01 Thermocouple

Hans-Petter Halvorsen
Using “Low-level” DAQmx VIs
Configure Additional Settings

In the previous example we just used the default setting. If you need to change some of the default setting, just right-click on the select input and create a constant.
Configure Additional Settings
While Loop
Increase speed by using “Start Task” and “Stop Task” VIs outside the While Loop
All Applications should have proper Error Handling

Further Improvements: Use the **State Machine** programming principle in 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