

Appendix 1 Guidelines for NI PCI-6036E Card

Used with LabVIEW, MATLAB/Simulink/Data Acquisition Toolbox

1. Introduction to PCI 6036E and Accessories

NI PCI-6036E



Figure 1 NI PCI 6036E Card

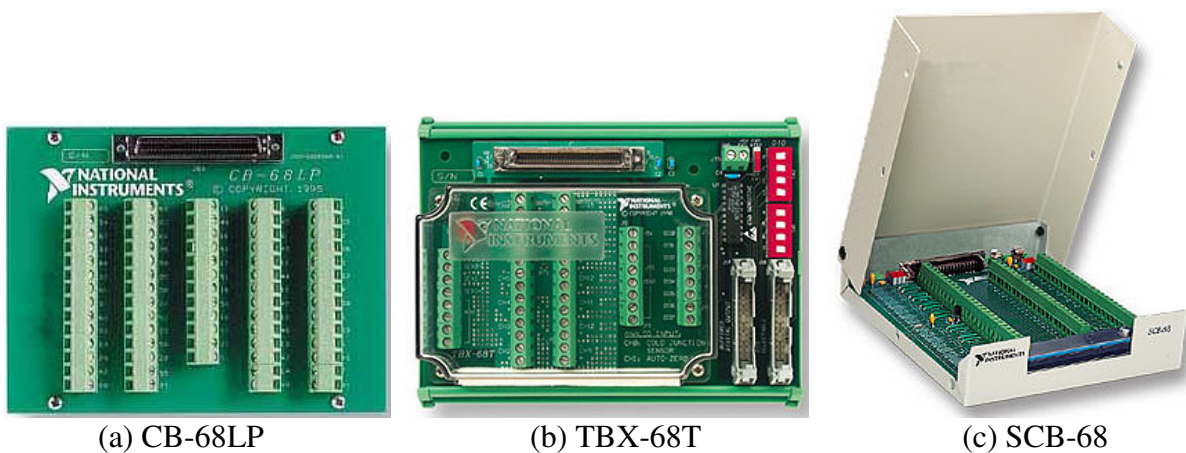


Figure 2 Screw terminal blocks: (a) CB-68 LP, (b) TBX-68 and (c) SCB-68

AI 8	34	68	AI 0
AI 1	33	67	AI GND
AI GND	32	66	AI 9
AI 10	31	65	AI 2
AI 3	30	64	AI GND
AI GND	29	63	AI 11
AI 4	28	62	AI SENSE
AI GND	27	61	AI 12
AI 13	26	60	AI 5
AI 6	25	59	AI GND
AI GND	24	58	AI 14
AI 15	23	57	AI 7
AO 0	22	56	AI GND
AO 1	21	55	AO GND
NC	20	54	AO GND
P0.4	19	53	D GND
D GND	18	52	P0.0
P0.1	17	51	P0.5
P0.6	16	50	D GND
D GND	15	49	P0.2
+5 V	14	48	P0.7
D GND	13	47	P0.3
D GND	12	46	AI HOLD COMP
PFI 0/AI START TRIG	11	45	EXT STROBE
PFI 1/AI REF TRIG	10	44	D GND
D GND	9	43	PFI 2/AI CONV CLK
+5 V	8	42	PFI 3/CTR 1 SRC
D GND	7	41	PFI 4/CTR 1 GATE
PFI 5/AO SAMP CLK	6	40	CTR 1 OUT
PFI 6/AO START TRIG	5	39	D GND
D GND	4	38	PFI 7/AI SAMP CLK
PFI 9/CTR 0 GATE	3	37	PFI 8/CTR 0 SRC
CTR 0 OUT	2	36	D GND
FREQ OUT	1	35	D GND

NC = No Connect

Figure 3 NI PCI-6036E pinouts

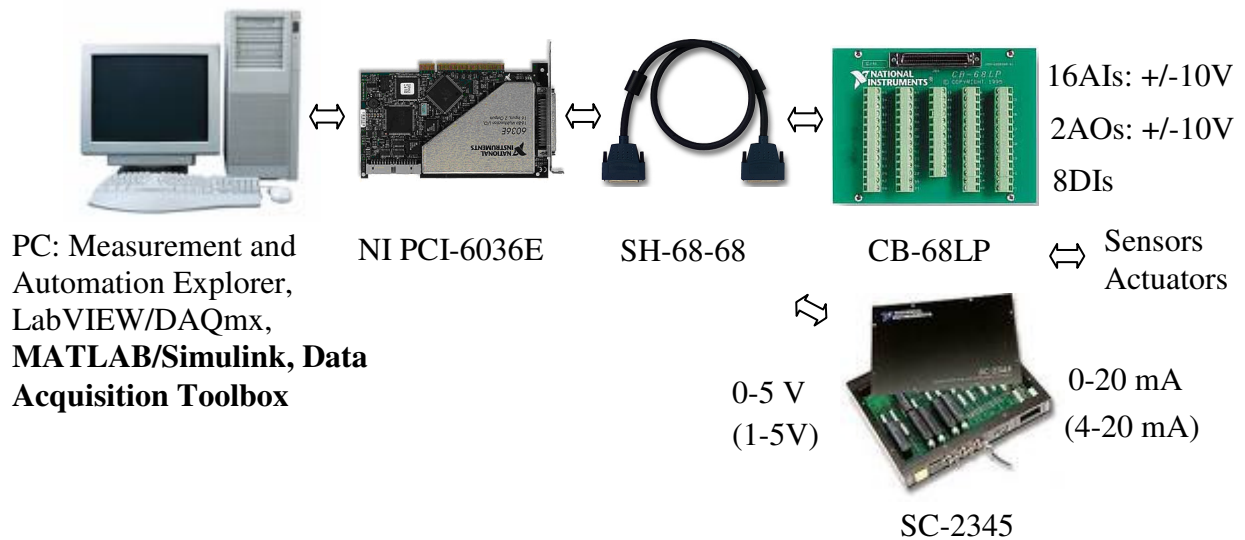


Figure 4 Installation of PCI-6036E and accessories

2. MATLAB/Simulink/Data Acquisition Toolbox

The following MathWorks software applications support NI PCI-6036E:

1. Data Acquisition Toolbox (working in MATLAB/Simulink)
2. Real-Time Windows Target (requires Real-Time Workshop)

Further information:

http://www.mathworks.com/products/product_listing/index.html

3. Data Acquisition Toolbox (MathWorks)

Data Acquisition Toolbox™ software provides a complete set of tools for analog input, analog output, and digital I/O from a variety of PC-compatible data acquisition hardware. The toolbox lets you configure your external hardware devices, read data into MATLAB® and Simulink® environments for immediate analysis, and send out data.

Data Acquisition Toolbox enables you to customize your acquisitions, access the built-in features of hardware devices, and incorporate the analysis and visualization features of MATLAB and related toolboxes into your design. You can analyze or visualize your data, save it for post-processing, and make iterative updates to your test setup based on your analysis results. Data Acquisition Toolbox allows you to use MATLAB as a single, integrated environment to support the entire data acquisition, data analysis, and application development process.



Figure 5 Data analysis and visualization with Data Acquisition Toolbox

Data Acquisition Toolbox also supports Simulink with blocks that enable you to incorporate live data or hardware configuration directly into Simulink models. You can then verify and validate your model against live, measured data as part of the system development process.

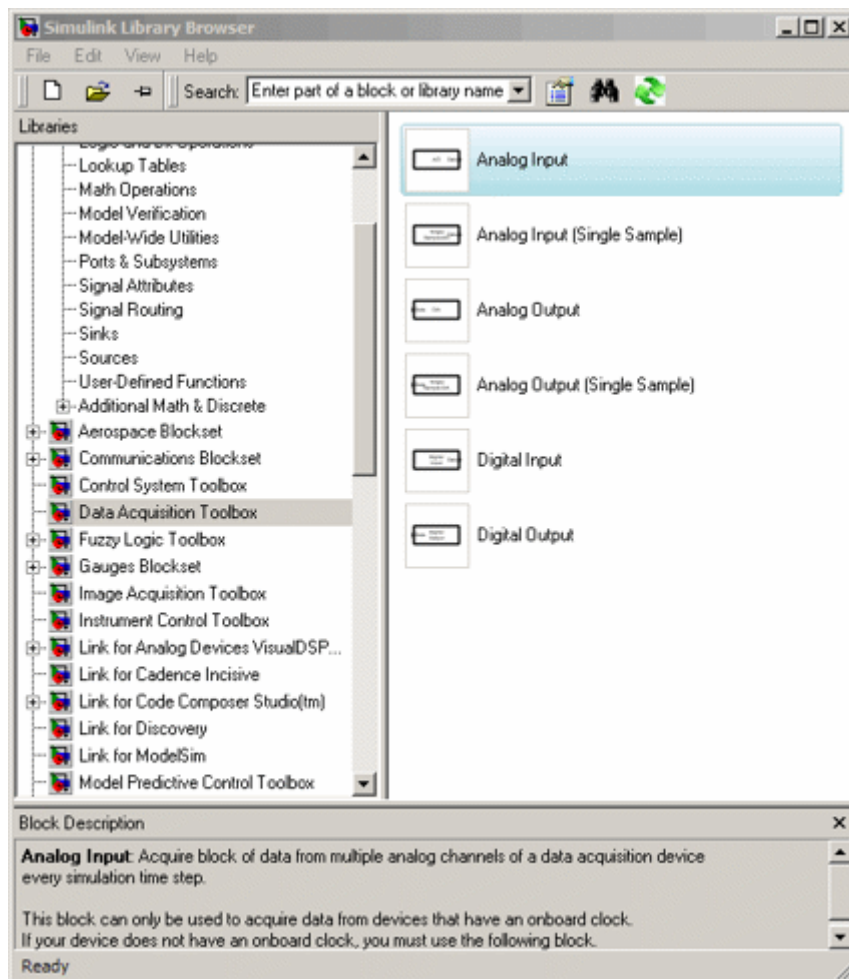
More information:

<http://www.mathworks.com/products/daq/>

4. Procedures to Create a Simulink Model for Data Logging and Generating

The toolbox block library contains six blocks:

- **Analog Input** — Acquire data from multiple analog channels of a data acquisition device.
- **Analog Input (Single Sample)** — Acquire a single sample from multiple analog channels of data acquisition device.
- **Analog Output** — Output data to multiple analog channels of a data acquisition device.
- **Analog Output (Single Sample)** — Output a single sample to multiple analog channels of data acquisition device
- **Digital Input** — Acquire the latest set of values from multiple digital lines of a data acquisition device.
- **Digital Output** — Output data to multiple digital lines of a data acquisition device.



4.1 Create a Simulink Model for Data Logging

Step 1: Open the Data Acquisition Toolbox Block Library

Step 2: Create a New Model

- Step 3: Add the Analog Input Block to the Model
- Step 4: Add a Scope to the Model
- Step 5: Specify Block Parameters
- Step 6: Connect the Blocks
- Step 7: Run the Simulation
- Step 8: Look at the Data in the Scope

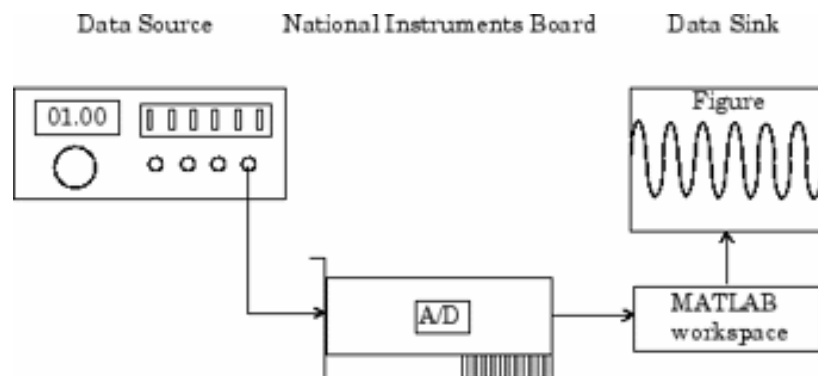
4.2 Create a Simulink Model for Data Generating

- Step 1: Open the Data Acquisition Toolbox Block Library
- Step 2: Create a New Model
- Step 3: Add the Analog Output Block to the Model
- Step 4: Add a Constant Block/Slider Gain to the Model
- Step 5: Specify Block Parameters
- Step 6: Connect the Blocks
- Step 7: Run the Simulation
- Step 8: Measure the signal

5. Sample MATLAB Codes

Acquiring Data with a National Instruments Board

The setup for this task is shown below.



Configuring the Data Acquisition Session

For this example, you will acquire 1 second of data on one input channel. The board is set to a sampling rate of 10 kHz, which is well above the frequency of interest. After you connect the input signal to the board, you will trigger the acquisition one time using a manual trigger.

You can run this example by typing `daqdoc4_2` at the MATLAB Command Window.

1. **Create a device object** — Create the analog input object `AI` for a National Instruments board. The installed adaptors and hardware IDs are found with `daqhwinfo`.

```
AI = analoginput('nidaq', 'Dev1');
```

2. **Add channels** — Add one channel to `AI`.
3. `chan = addchannel(AI, 0);`

- 4. Configure property values** — Assign values to the basic setup properties, and create the variables `blocksize` and `Fs`, which are used for subsequent analysis. The actual sampling rate is retrieved because it might be set by the engine to a value that differs from the specified value.

```
5. duration = 1; %1 second acquisition
6. set(AI, 'SampleRate', 10000)
7. ActualRate = get(AI, 'SampleRate');
8. set(AI, 'SamplesPerTrigger', duration*ActualRate)
9. set(AI, 'TriggerType', 'Manual')
10. blocksize = get(AI, 'SamplesPerTrigger');
    Fs = ActualRate;
```

- 11. Acquire data** — Start `AI`, issue a manual trigger, and extract all data from the engine. Before `trigger` is issued, you should begin inputting data from the function generator into the data acquisition board.

```
12. start(AI)
13. trigger(AI)
    wait(AI, duration + 1)
```

The `wait` function pauses MATLAB until either the acquisition completes or the time-out elapses (whichever comes first). If the time-out elapses, an error occurs. Adding 1 second to the duration allows some margin for the time-out.

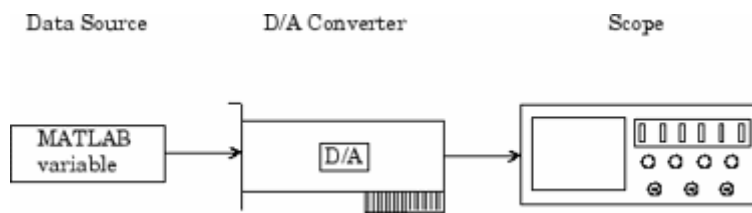
```
data = getdata(AI);
```

- 14. Clean up** — When you no longer need `AI`, you should remove it from memory and from the MATLAB workspace.

```
15. delete(AI)
    clear AI
```

Outputting Data with a National Instruments Board

In this example, sine wave data is generated in the MATLAB workspace, output to the D/A converter on a National Instruments board, and displayed with an oscilloscope. The setup is shown below.



You can run this example by typing `daqdoc6_2` at the MATLAB Command Window.

- 1. Create a device object** — Create the analog output object `AO` for a National Instruments board. The installed adaptors and hardware IDs are found with [daqhwinfo](#).

```
AO = analogoutput('nidaq', 'Dev1');
```

- 2. Add channels** — Add one channel to `AO`.

```
chan = addchannel(AO, 0);
```

3. **Configure property values** — Define an output time of four seconds, assign values to the basic setup properties, generate data to be queued, and queue the data with one call to `putdata`.

```
4. duration = 4;
5. set(AO, 'SampleRate', 10000)
6. set(AO, 'TriggerType', 'Manual')
7. ActualRate = get(AO, 'SampleRate');
8. len = ActualRate*duration;
9. data = sin(linspace(0, 2*pi*500, len))';
   putdata(AO, data)
```

10. **Output data** — Start `AO`, issue a manual trigger, and wait for the device object to stop running.

```
11. start(AO)
12. trigger(AO)
    wait(AO, 5)
```

13. **Clean up** — When you no longer need `AO`, you should remove it from memory and from the MATLAB workspace.

```
14. delete(AO)
    clear AO
```