NidaqServer

Michael Stephan

September 28, 2017

1 Introduction

Digital Input/Output capabilities of certain I/O cards from National Instruments. Currently supported devices are: PCI-6221 (Fries lab) and PCI-6503 (Singer lab). Both cards provide 3 digital ports each representing 8 digital lines.

There is a Matlab class that provides most of the fuctionality from within Matlab.

Prior to sending any pipe commands to the server, you have to initialize the pipe:

```
NidaqServer.Connect();
```

2 Subsystems

2.1 Reward

There are two mechanisms to drive the reward line (`port2/line3`).

1. auto reset event named `Reward`. Whenever this event is set to the signaled state, the server generates a pulse of a given length at the output line. The default duration is 100 ms. It can be changed though a pipe command. After connecting to the `kernel32` event:

   ```
   reward = IPCEvent('Reward');
   ```

   you may generate a reward pulse by setting the event to the signaled state:

   ```
   reward.trigger();
   ```

   Optionally you may have the server send out an event marker code for each reward event. Use a pipe command to chose the marker code ($\neq 0$) to use and to enable this feature:

   ```
   NidaqServer.SetRewardCode(code);
   ```

2. Send a command to the server defining a sequence of up to 8 output pulses (and the time intervals between them).

At the end of a reward pulse or sequence the server (starting with version 1.0.4.0) sets the `kernel32` auto reset event `RewardDone` to the signaled state. You may use this event to wait for the end of the reward signal.

2.2 Event Markers

`port0` and `port1` are used to output 16 bits of event marker information. The marker value is supplied to the server through a pipe command. The output lines are strobed with a positive pulse on `port2/line7` (PCI-6221) or `port2/line0` (PCI-6503) respectively.

2.3 Digital Input Events (Lever etc.)

This feature signals `kernel32` events on certain changes of selected digital input lines of `port2`. There are two distinct modes you can define for a line:

1. "on/off" mode. In this mode you define two distinct manual reset events. On a 0 to 1 transition of the input line the "on"-event is set to the signaled state and the "off"-event is reset to unsignaled. Accordingly on a 1 to 0 transition of the input line the "off"-event is set to the signaled state and the "on"-event is reset to unsignaled.

2. "pulse" mode. In this mode you define a single auto reset event. This event is set to the signaled state whenever a 0 to 1 transition is detected on the according input line.

The configuration of this subsystem is done in two steps. First you have to define the input lines by specifying the respective event names. Lines added with two event names will operate in the "on/off" mode whereas lines with only one event name will operate in "pulse" mode. After adding all lines to be monitored you have to start the subsystem. The server will then begin to monitor the defined lines and signal the respective events. Once the subsystem is started the configuration can't be changed anymore.

The lines that can be used for this subsystem depend on the daq-card in use:

PCI-6221: Any line (0...7) of `port2` can be used. The chosen lines are configured as input lines. All other lines are unaffected.

PCI-6503: Only lines 4–7 of `port2` can be used. These lines are always configured as input lines by the server. They can't be used as output lines.