

# NidaqServer

Michael Stephan

June 7, 2018

## 1 Introduction

The aim of this server is to provide a unified interface to the digital Input/Output capabilities of certain I/O devices from National Instruments through standard Windows kernel32 mechanisms. Currently supported devices are: PCI-6221 (Fries lab), PCI-6503 (Singer lab) and USB-6353. All devices provide 3 digital ports each representing 8 digital lines. (The USB-6353 provides 24 additional lines which currently are not used).

There is a Matlab class (**DaqServer**) that provides most of the functionality from within Matlab.

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

```
DaqServer.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 through a pipe command:

```
DaqServer.SetRewardTime(timems);
```

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:

```
DaqServer.SetRewardCode(code);
```

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

```
DaqServer.Reward([100 500 100]);
```

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.

You may query the server for the total reward time (in ms). This query will also reset the time counter within the server. Thus the returned time is the total time the reward line was active since the last query (or the start of the server). Matlab:

```
DaqServer.GetTotalTime()
```

## 2.2 Event Markers

16 bits of event marker information are output through `port0` and `port1` on PCI-6221 and PCI-6503. With USB-6353 `port0/line0:15` are used. 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/USB-6353) 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 device in use:

**PCI-6221/USB-6353:** 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.

## A Commands

To control the server the following commands may be sent via a kernel32 pipe named `\\.\pipe\NidaqServerPipe`. The commands are sent as a stream of bytes.

- [1 **line** **pulseEventName**] Add a line in “pulse” mode. See section 2.3.  
Matlab example: `DaqServer.AddLine(6, 'ButtonPress');`
- [2 **line** **onEventName** **offEventName**] Add a line in “on/off” mode. See section 2.3.  
Matlab example: `DaqServer.AddLine(6, 'ButtonPress', 'ButtonRelease');`
- [3] Start the digital input subsystem.  
Matlab: `DaqServer.Start();`
- [4 **time**] Set the length (in ms) for pulses generated by reward events as a 16 bit unsigned integer. See point 1. in section 2.1.
- [5 **time1** ... **timeN**] Generate a reward pulse sequence. Specify the times (in ms) for the pulses and pauses between them as 16 bit unsigned integers. See point 2. in section 2.1. The number of durations given must be odd.
- [6 **marker**] Set the event marker output lines according to the 16 bit unsigned integer number **marker**. See section 2.2.
- [7 **marker**] Set the event marker code (16 bit unsigned integer) to output when a reward event is detected. See point 1. in section 2.1.
- [8] Get the total reward time (in ms) since the last query. You have to read 4 bytes from the pipe (as an `uint32`) after issuing this command.