

Manual for TENS-gated stimulation with NI-DAQ device and Matlab

This is a manual for assembling an interface between a commercially available TENS device, NI-DAQ device and Matlab, including a Matlab script example of TENS-gated cued sipping.

Requirements

1. Data acquisition device: [National Instruments USB-6501](#), 24-ch programmable 5V TTL or 3.3 V digital I/O, 8.5 MA Product number: 779205-01
2. DAQ drivers: Get drivers for NI-DAQ, “[NI-DAQmx 19.6](#)” You will mainly use the “NI Device Monitor” application.
3. Matlab (2017b, 64 bit) with [Data Acquisition Toolbox](#)
4. Matlab Data Acquisition Toolbox Support Package for National Instruments NI-DAQmx Devices, for Matlab 2014A and later. My experience was that the installation consists of multiple steps and I had to restart the installation multiple times to get it to complete. To install via the add-on explorer menu in Matlab or directly on [their webpage] (<https://www.mathworks.com/hardware-support/nidaqmx.html>). Make sure the NI device is not connected or NI software running when installing this support package.
5. TENS device with extra unused channel, for example [this 4-channel model](#)

Hardware connections:

The NI-DAQ device ports can fit regular electrode connectors or you can cut off the connectors and strip the wire. Plug the red connector into P2.7 and the black connector into GND next to it (Figure 1). Use some electrical tape to secure the connector.

TENS device settings:

EMS settings, duty cycle of 30 seconds on, 30 seconds off, 0.25 ms-duration monophasic square wave pulses at 25 Hz, ramp of 4 s.

Test NI-daq device acquisition:

After installing the NI-DAQ drivers, start the “NI Device Monitor” from the Start menu and open it from your system tray (Figure 2a). Select “Test this device” (Figure 2b) and switch to the “Counter IO” tab (Figure 2c). Press “Start” (Figure 2d) and turn on the TENS device and set channel 4 to “1”. Watch the “counter”. With a 30 second block of 25 Hz stimulation

the counter should go up to 750, but it misses the first 21 and last 21 pulses during the ramp time, coming to a total of 708 edges counted. This means that the first edge is counted ~1 second after the block started.

Matlab script:

Configure the data acquisition session with the **daq.createSession('ni')** and **addCounterInputChannel** commands. Note that if you have multiple NI devices registered on your computer, you may need to change the device number (see the "NI Device Monitor to find the right device number). Use **startForeground** to reset the counters (**resetCounters** does not work). Use **inputSingleScan** to wait for the first counted edge.

%Written by Maria G. Veldhuizen 21-07-2020

*%script to offset cued sipping of a liquid stimulus (or other
%behavior or perceptual stimulation) from TENS stimulation blocks with a
%duty cycle of 30 seconds on, 30 seconds off, 0.25 ms-duration monophasic
%square wave pulses at 25 Hz*

*%dependency on Matlab data acquisition toolbox functions,
%nidaq drivers,
%nidaq toolbox for matlab
%see manual for links to products and connection details*

%% preparation section

% prepare reading trigger from TENS device

```
s = daq.createSession('ni');  
    try % assign channel and suppress warning  
        ch = addCounterInputChannel(s, 'Dev1', 'ctr0', 'EdgeCount');  
    catch  
    end
```

%just to be safe, reset the counter

```
    try % reset counter and suppress error message  
        startForeground(s); % seems to reset the counter, unlike resetcounters  
    catch  
    end
```

% prepare audiofiles

```
[r,fs]=audioread('sip.m4a'); % Load sip cue  
pause(.5);  
sip = audioplayer(r, fs);  
[r2,fs]=audioread('ready.m4a'); % Load ready cue (to indicate start of sipping  
block)  
pause(.5);  
ready = audioplayer(r2, fs);
```

```

%pre-allocate variables
TENSblocktime = zeros(1,6);
sipblocktime=zeros(1,6);
siptime = zeros(1,55);
k=1;

% wait for user input
h = warndlg('When you are ready, hit ok','Wait...');
uiwait(h);

%% experiment run section
timecont=clock;%log start time of run

for j = 1:10
    %% wait for stimulator to start
    disp('waiting for TENS to start');
    start=inputSingleScan(s);%need to have one first to get the counter
    started right
        while start == 0
            start=inputSingleScan(s);
        end
        display('TENS started');
        TENSblocktime(j) = etime(clock,timecont);% Log time
        pause(29); % wait for tVNS block to end
        play(ready);% play sound to signal get ready
        display('sip block started');
        sipblocktime(j) = etime(clock,timecont); % Log time
        pause(3);
        for i = 1:5
            play(sip);%play sound to signal to swallow
            siptime(k) = etime(clock,timecont); % Log time
            k=k+1;
            pause(5);% give time to swallow
        end
        try % reset counter and suppress error message
            startForeground(s);
        catch
        end
    end
end
display('end of experiment');

```

License and attribution

CC-0. If you would like to use this code in your project, please cite:

Maria G. Veldhuizen, TENS-gated-matlab-trigger, (2020), GitHub repository,
<https://github.com/mariaveldhuizen/TENS-gated-matlab-trigger>

BibTeX entry:

@misc{Veldhuizen2020TENS, author = {Veldhuizen, Maria G.}, title = {TENS-gated-matlab-trigger}, year = {20120}, publisher = {GitHub}, journal = {GitHub repository}, howpublished = {}, commit = {9e570159467f049fc4ace5d460f57f6848c82c80} }

This code is hosted publicly at <https://github.com/mariaveldhuizen/TENS-gated-matlab-trigger> and is supported by the following grant: TÜBİTAK 2232 International Fellowship for Outstanding Researchers grant no 118C299 to Maria G. VELDHUIZEN