Quick Start Guide

**OSC1-36**

*Portable 36-channel microLED Driver System*

**1 Introduction**

The University of Michigan's OSC1-36 chip serves to trigger user-defined optical stimulation on up to 36 different channels. Matlab GUI and library is available for interacting with OSC1-36 stimulation system. The open-source code for the GUI interface is available at GitHub link: https://github.com/YoonGroupUmich/OSC136H. We invite you to post issues and solutions there as well, which will help our scientific community as we have limited resources for supporting this device.

**2** **System Setup**

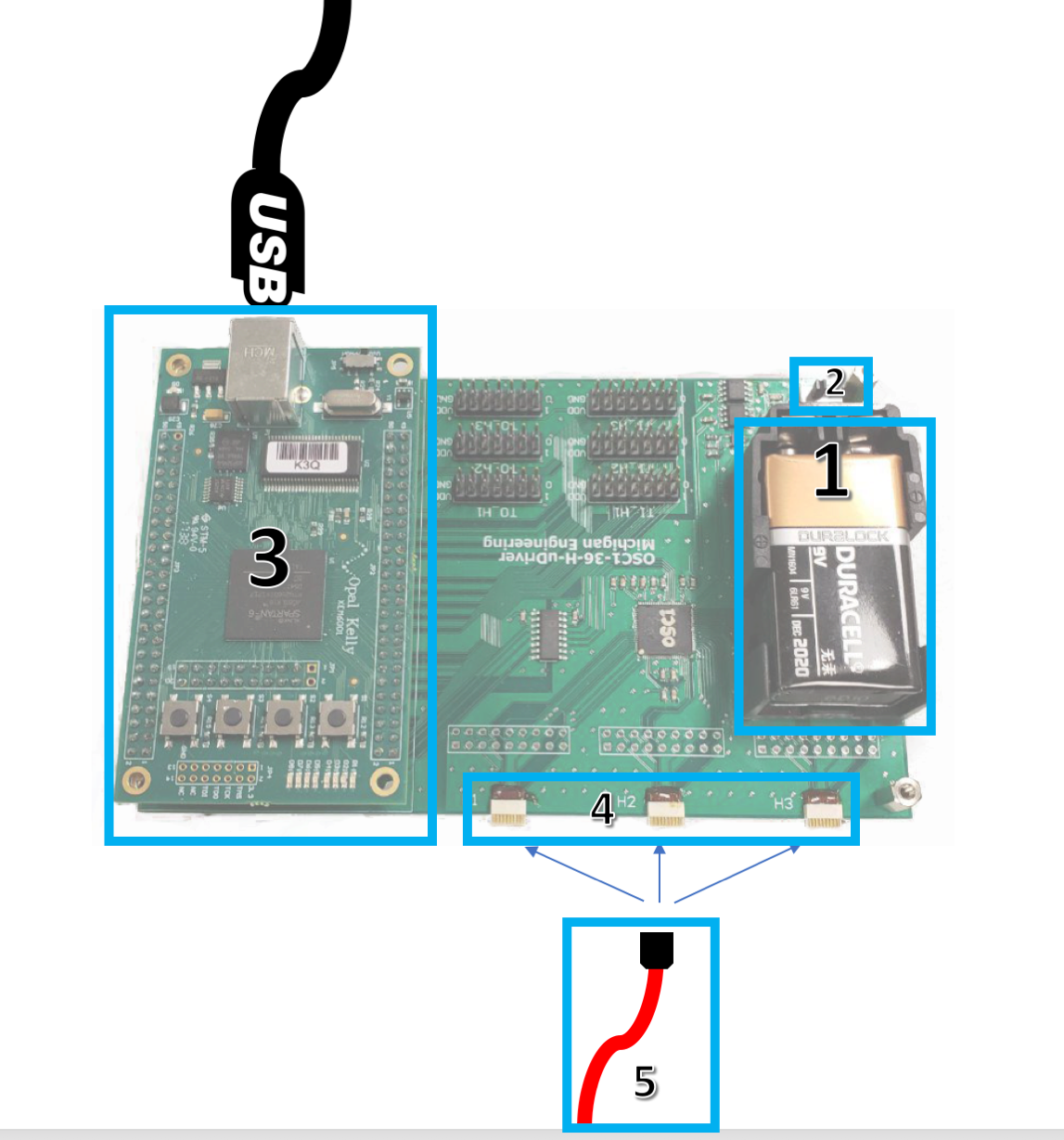
**2.1 Included items**

(1 pcs) USB-A to USB-B Cables

(1 pcs) OSC1 System w/ XEM-6001 FPGA

(1 pcs) 18-pin 3-meter Headstage Cable

**2.2 Device Setup**

****

**Step 1:** Install 9V battery (not included).

**Step 2:** Power Device on with Switch. An LED light indicates the system is on.

**Step 3:** Plug the included USB cable to the device and the computer with which you are using the GUI tool.

**Step 4:** Choose the appropriate system-side connector (labeled H1, H2, H3) based on your GUI settings. Each connector can drive 12 channels.

**Step 5:** Plug the included Headstage cable into your headstage.

*Note: The pins labeled TX\_H1-3 are triggers corresponding with H1-3 outputs which can be used to overcome the high latency of USB for more precise temporal control over stimulation, if desired. However, this feature is not supported in our GUI tool and would require a custom setup to be exploited.*

*Please see section 3.1 to install the GUI tool.*

**3** **GUI Interface**

**3.1 Installation**

To use the OSC136H library requires the installation of Opal Kelly FrontPanel (available on GitHub), Matlab 2015b (or newer), and the MinGW compiler for Matlab. The OSC136H library is currently only compatible with Windows 64-bit.

### Installing Opal Kelly Frontpanel

Use the included driver to install Opal Kelly Frontpanel on your PC.

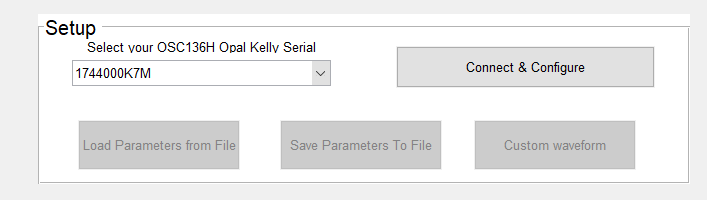
### Installing MinGW for Matlab

MinGW is available for all Matlab versions 2015b and newer, but requires a workaround for any version 2017a and previous. The work around is described in the Bug Report in the following link:

<https://www.mathworks.com/matlabcentral/fileexchange/52848-matlab-support-for-mingw-w64-c-c++-compiler>

**3.2 Connection**

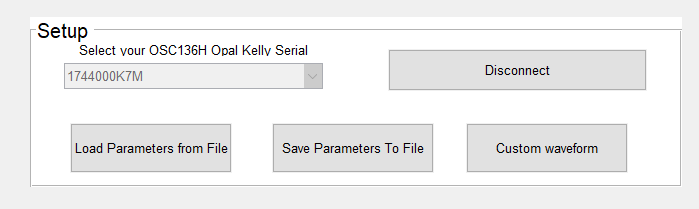
The user interface is frozen when there is no board connecting to the current PC. When the GUI window is open, it will actively detect devices before connecting to one. Click on ‘Connect & Configure’ and choose ‘OSC1\_36\_H\_uDriver\_Control.bit’ that can be downloaded along with the other source files.



*Note:* *There might be multiple Opal Kelly Boards connecting PC. Be sure to connect to the FPGA embedded on OSC1-36 by checking the serial number shown in the text box.*

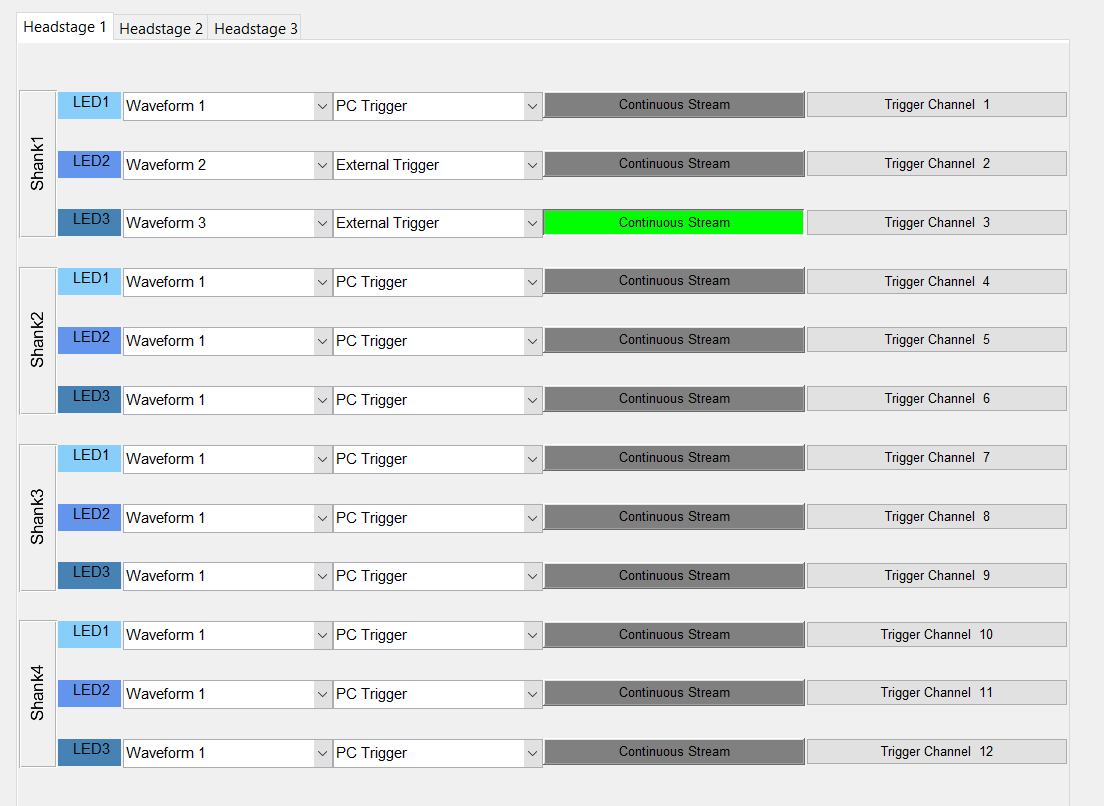
**3.3 Configuration**

The user can define up to 4 different square waveforms and 1 custom waveform that can further be selected to trigger on different channels. The trigger type can be set to external trigger or PC trigger for the square waveforms and can only be PC triggered for the custom waveform. Users can manually type in the parameters for square waveforms and select mode for each channel. Also, the parameters can be initialized by properly formatted configuration files. The system can be initialized by configuration files and can also save current configurations to a config file. An example of a properly formatted configuration file is given in [example\_config.txt](https://github.com/YoonGroupUmich/OSC136H/blob/master/example_config.txt) in GitHub source folder. You can always save you configuration with ‘Save Parameters To File’ or reload your pre-saved configuration with ‘Load Parameters From File’ (Be sure to turn off all the continuous stimulation before changing the configuration).



**3.4 Trigger Stimulation**

After the configuration is finished, the user now can trigger the optical stimulation on up to 36 channels with the pre-defined waveforms. There are 4 square waveforms and 1 custom waveform to be selected from. Two types of trigger mode are available to trigger the stimulation of the selected waveform: One-shot Trigger and Continuous Trigger.



**3.4.1 One-shot Trigger**

The one-shot trigger will be activated after the pushbutton *Trigger Channel #* is pressed. The one-shot trigger will assign a train of pulses on the specific channel with parameters defined by the selected waveform. Each of the channels will work independently until it finishes the pre-defined number of cycles.

**3.4.2 Continuous Trigger**

The continuous trigger will be activated after the toggle button *Continuous Stream* is pressed. The continuous trigger will assign a continuous wave of pulses with defined period, duty cycle and amplitude of the selected waveform. But the waveform will be continuous until the toggle button is triggered again, which means the number of pulses is always neglected when continuous trigger is activated on a certain channel.

Note that continuous trigger has a higher priority than one-shot trigger. When continuous trigger is activated, the one-shot trigger push button of the triggered channel will not have any effect. Each of the channels will work independently until the toggle button of that channel is pressed again.

**3.4.3 External Trigger**

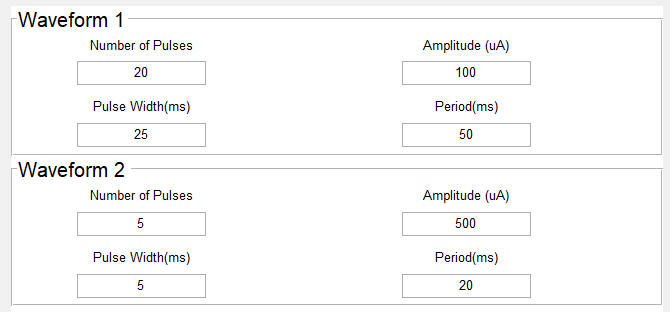
In addition to trigger the stimulation by mouse-clicking, OSC1-36 also supports triggering on a rise-edge TTL pulse. If selected ‘External Trigger’ by replacing the default ‘PC Trigger’, the channel will activate one-shot trigger if the corresponding External Trigger channel receives a rise-edge TTL pulse.

*Note: Each rise-edge of TTL pulse will reset the one-shot trigger (in the case that TTL pulse has a higher frequency than the triggered waveform). Selecting Continuous Mode will also overwrite the external trigger function, since the channel will be triggered continuously.*

**3.5 Limitations**

There are certain limitations of the OSC136H user interface to fit in the design of the firmware. These include some requirement of the waveform generated and other requirement on the custom waveform.

**3.5.1 Square Waveform**



**Number of Pulses**

Valid number of pulses for the square waveform parameter includes integers that range from 0 to 63.

**Amplitude**

Valid amplitude for the square waveform parameter includes integers that range from 0 to 1023 (μA).

**Pulse Width**

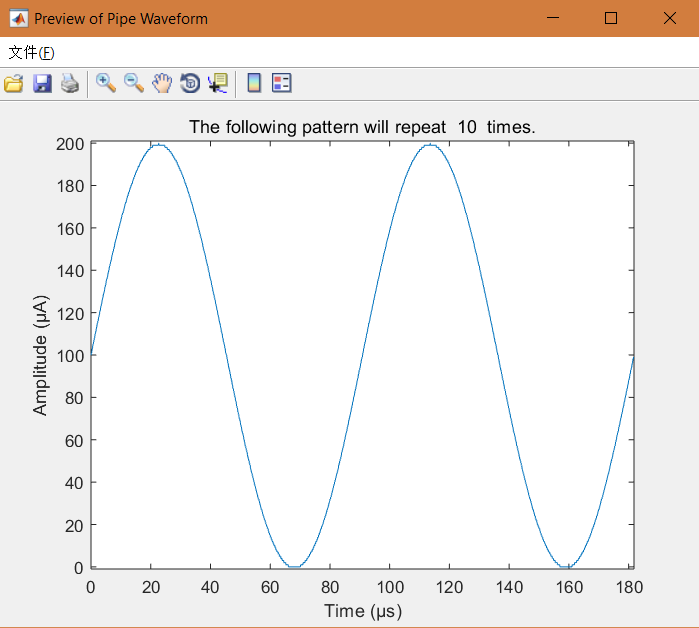
Valid pulse width for the square waveform parameter includes multiples of 2.5 in range 0 to 637.5 ms.

**Period**

Valid period for the square waveform parameter includes multiples of 5 in range 0 to 1275 ms.

**3.5.2 Custom Waveform**

User can customize their waveform by initializing a Custom Waveform. The user must assign each amplitude at every period to design the custom waveform (See a sine waveform example in *custom\_waveform.cwave* and the generating script *write2file.m*). After loading the custom waveform, previewing the waveform is supported with desired *Amp vs. time*.



**Number of Pulses**

Valid number of pulses for the custom waveform parameter includes integers that range from 0 to 8192.

**Input data size**

The valid size of the array that forms the custom waveform includes integers that range from 0 to 32767, i.e. the maximum number of sampling amplitude for a single period in custom waveform is 32767.

**Artifact**

Discrete current changes may cause detectable artifacts in your recording signal. Please test in saline before use.