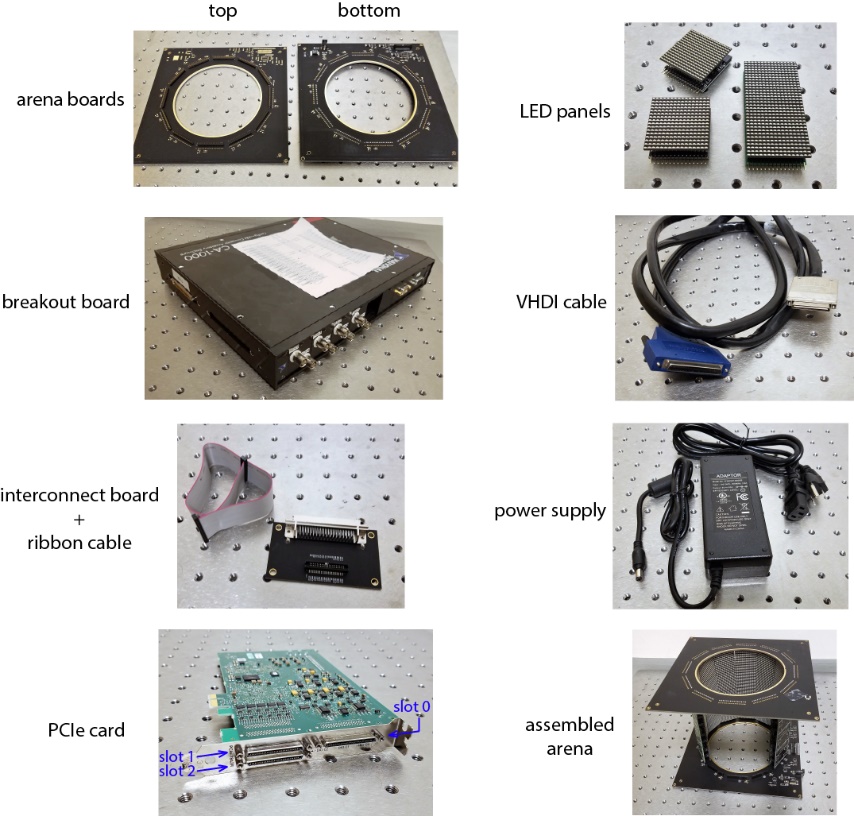
This document details how to set up a G4 LED display system and briefly describes the various software tools developed for this system. These tools can be used to generate visual stimuli, run experiments, and analyze the acquired results. Some of the software tools described later in this document do not require a physical LED arena set up and attached to the computer in order to be used; for example, the Motion Maker scripts can be used to generate and visualize patterns without any G4 hardware attached, and the Data Analysis scripts only require the TDMS log files generated during an experiment in order to analyze and plot data. Other tools, such as PControl and the Protocol Designer scripts, will only be fully functional when connected to a G4 Display system.

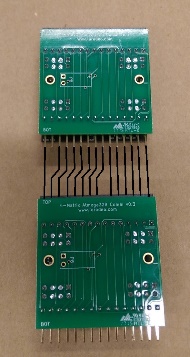
1. **Hardware Setup**

**What you need:**

* Windows desktop PC with a PCIe-7842R PCI card installed
  + Card available here: <https://www.ni.com/en-us/support/model.pcie-7842.html>
* G4 arena boards (1 bottom board and 1 top board)
* LED panels, already programmed (see ‘G4\_Display\_Tools/G4 Panel Programming’)
* NI Breakout board (optional)
* VHDCI cable(s)
* Interconnect board with ribbon cable
* Desktop power supply (5V, 10A)
  + Available here: <https://www.adafruit.com/product/658>



**Setting up:**

* Stack the LED panels into columns of the desired length and insert the columns into the G4 arena board (so that LED panel columns are sandwiched in between top and bottom boards).
  + A blank space can be created in columns of panels by using

wires to connect the pins between distant rows of panels

(e.g. between row 1 and 3) as shown in the picture to the

right. Blank spaces can be used to point cameras or other

equipment into the arena through the panel columns.

* Connect the arena to the interconnect board using the ribbon cable, and connect the interconnect board to the PCIe card (slot 1) using the VHDCI cable
* (optional) connect a breakout box to the PCIe card (slot 0) with another VHDCI cable
* Connect the power supply to the G4 arena bottom board.

1. **Software Setup**

Download and install the latest version of MATLAB. Required toolboxes are ‘instrument control’ and ‘statistics and machine learning’. A recommended toolbox is ‘parallel computing’.

Download and install the G4 Host LabVIEW executable. <https://www.dropbox.com/s/mywy2a3gb6vxhec/HHMI%20G4%20Host%28Ver1-0-0-230%29%20with%20installer.zip?dl=0>

Download (or clone) the G4\_Display\_Tools GitHub repository. <https://github.com/JaneliaSciComp/G4_Display_Tools>

**Add paths for G4\_Display\_Tools in MATLAB**. On the MATLAB home tab, click ‘set path’, then ‘add with subfolders’, then navigate to the location where you downloaded G4\_Display\_Tools. Save and close the set path window after the paths have been added.

**Configure the arena size for the display controller**. Open the Display controller configuration file in located in C:\Program Files (x86)\HHMI G4\Support Files\HHMI Panels Configuration.ini in a text editor such as notepad. Make sure that the “number of rows” field has the correct number of rows based on the size of LED arena you have built (e.g. for an arena that has 12 LED panel columns with each column 4 panels tall, the number of rows is 4.) The “number of columns” field should be the number of columns that the arena can support – not how many columns have been populated with LED panels – which is 12 for standard LED arenas.

**Configure the arena size for the PControl software**. In MATLAB, open G4\_Display\_Tools\PControl\_Matlab\userSettings.m and set the "NumofRows" and “NumofColumns” fields to the same values set for the display controller. If the arena will be mounted upside-down (i.e. if the arena will be mounted with the bottom board on the top and vice versa), set "flipUpDown = 1" and "flipLeftRight = 1". Otherwise if the arena will be mounted right-side up, leave both = 0.

**Configure the arena size for the Motion Maker software**. In MATLAB, run configure\_arena (located in G4\_Display\_Tools\Motion\_Maker\_G4\support\configure\_arena.m). Make sure the # of rows/columns of panels is correct. If you are using G4 panels consisting of 16x16 LEDs, set the panel size to be 16. If you are using a G4 arena consisting of 12 LED columns that would fully enclose the cylindrical arena (e.g. the 12” arena), set the arena circumference to 12. If you are using a differently-sized arena, such as the open-form 12/18 cylindrical arena (where 18 panel columns would be needed to fully enclose the cylinder), set the circumference to 18. If the center of the arena (located between columns 6 and 7) is not oriented directly “forward” from the center of the cylinder, use the “arena rotations” to account for that, otherwise some motion types will not be oriented correctly.

1. **Verifying that the setup is working**

After the hardware and software setup is complete, open MATLAB and run PControl\_G4. Two windows will open: a LabVIEW window followed by a MATLAB GUI. Once the PControl\_G4 MATLAB GUI has opened, click on the “arena” tab and click “all on”. If all pixels turn on, then the system has been set up successfully and you can continue into the next section for an overview of the software tools to begin more advanced operation of the display system.

If the arena does not turn on, try these common troubleshooting steps:

* Check that the connection between the interconnect board and the VHDCI cable is good, as the VHDCI cables can sometimes need a very tight fit to make all the connections.
* After running PControl\_G4, check the LabVIEW window to see if the green light labelled “dequeue timeout” is lit. If it is, it may be that the transfer speeds between the PCIe card and the computer’s memory is too slow. If the computer is relatively new/fast, one possible cause of this problem has been noted with newer Dell workstations, which can be fixed by updating the BIOS. Regardless of the computer make/model, it may be worth updating the computer’s BIOS and seeing if that helps, which can be done by finding your PC’s manufacture support webpage and downloading the latest BIOS installer (e.g. for Dells: <https://www.dell.com/support/home/us/en/04>).
* Using a voltmeter, check that the arena board is being supplied with 5 V as expected.
* Some issues in the past have been caused by mistakes in the arena board assembly. The connectors between the arena board and the LED panels have sometimes been placed on the wrong side of the arena board or have had the gendered 15-pin connectors switched between the top and bottom arena boards. To see if this is the case, remove all of the LED panels from the arena board and plug one column back in, but inserted backwards (where the LEDs are facing to the outside of the arena). If an “all on” command turns on the LEDs in this case, then the connectors were placed incorrectly.

1. **Overview of G4\_Display\_Tools**

**Motion\_Maker\_G4**

This set of scripts and GUIs can be used to design patterns (primarily for displaying motion, rather than pictures of objects) on the G4 display. Patterns are generated using the Motion\_Maker\_G4.m script based on input parameters that describe the desired motion. These scripts output two types of pattern files: The first type is a .mat file which contains the created pattern matrix and all the pattern parameters so that it can be easily read back into MATLAB. The second type is a .pat file containing a binary vector of the pattern that can be quickly accessed by the Display Controller. Only the .pat file is necessary to be displayed on a G4 arena, though the .mat file is needed to be easily loaded back into MATLAB for viewing, debugging, or for creating experiments with the G4\_Protocol\_Designer (described later). See G4\_Display\_Tools\Motion\_Maker\_G4\About Motion Maker.docx for more details.

**Function\_Maker\_G4**

This set of scripts and GUIs allow for the design and creation of analog output functions and position functions, to be used in conjunction with displaying patterns on a G4 display. Position functions control what frame of the selected pattern is displayed for every refresh cycle (when the display system is operating in position function mode), operating at a rate of either 500 or 1000 Hz (1-bit or 4-bit patterns, respectively). Analog output functions control the voltage of the analog output channels of the G4 system (accessed easily with the optional breakout box) in a way that is synchronized to the display refresh cycle, operating at 1000 Hz regardless of the pattern refresh rate. Similar to Motion\_Maker\_G4, functions are created using the Function\_Maker\_G4.m script based on input parameters that describe the desired function. These scripts output two types of files: The first type is a .mat file which contains the created function array and all the function parameters so that it can be easily read back into MATLAB. The second type is either a .afn (for analog output functions) or .pfn (for position functions) file containing a binary vector of the function that can be quickly accessed by the Display Controller.

**PControl\_G4**

Developed by Jinyang Liu, PControl\_G4 allows for communication between the G4 display system and MATLAB by establishing a TCP connection between the two. A communication channel can be opened using the connectHost function, and messages can be translated and sent to the display using Panel\_com. Commands for displaying patterns and using functions in various modes can be sent, provided that an ‘experiment folder’ has been created and specified. Experiment folders can be made by manually selecting pre-made pattern and function files using the design\_exp GUI. Running PControl\_G4 automatically connects to the G4 display and opens a window where an experiment folder can be specified and various commands can be sent, including commands for displaying the patterns included in the experiment folder. Finally, examples of custom-written patterns and functions are also included in this set of scripts.

**G4\_Protocol\_Designer**

Developed by Lisa Taylor, these scripts and GUIs allow for designing, visualizing, and running experimental protocols using patterns and functions that have already been created. An experimental structure can be created and visualized using the G4\_Experiment\_Designer GUI, where pre-made pattern and function files can be selected and organized. Experimental protocols can be validated within the GUI and saved as .g4p files. The G4\_Experiment\_Conductor GUI can run experimental protocols and display information on the current experiment progress in real-time. See G4\_Display\_Tools\G4\_Protocol\_Designer\User Instructions.docx for more details on how to use these scripts.

**G4\_Example\_Experiment\_Scripts**

These scripts – using many of the functions described in the previous tools – demonstrate an entirely script-based solution for creating patterns, functions, and experiment folders, as well as creating and running experiments with the G4 system.

**G4\_Data\_Analysis**

These scripts can be used to read data logged and acquired by the G4 display system into MATLAB. Each experiment (marked by ‘start log’ and ‘stop log’ commands) outputs a folder of log files in .TDMS format, which can be read and converted into a MATLAB struct using the G4\_TDMS\_folder2struct function. These log files contain data and timestamps corresponding to the frames displayed during that experiment as well as the commands received over TCP. Any active analog output and analog input channels are also logged by both voltage and corresponding timestamp. Additional scripts are included for further processing, analyzing, and plotting data from two example categories of experiments – a tethered fly walking on an air-suspended ball, and a tethered flying fly monitored with a wingbeat analyzer. An example of a full data analysis pipeline is shown in the test\_G4\_Data\_Analysis.m script.