HILO App

The HILO app is designed as a user-friendly software to control a HILO spinning machine. The functionality is limited (for now) to controlling the thickness and amount of twist of spun yarn. Users can select a HILO device connected to their computer (via a USB serial port) or use a simulator. The app is written in Processing and can be launched from the Processing IDE, or exported as a stand-alone app for MacOS, Windows or Linux.

Keyboard Controls

The app is made to be used with the mouse or touchpad. A few keyboard shortcuts were left in, which can be helpful when using or debugging.

* **c**​ - connect to HILO
* **C**​ (uppercase, Shift + c) - disconnect from HILO
* **Spacebar**​ - start or stop spinning
* **l**​ (lowercase L) - print the list of available serial ports to the console (for debug)
* **I**​ (uppercase i, Shift + i) - load a new image (only in pattern mode)

Settings Files

The app is provided with a set of configuration files, within the ​**data/settings/**​ folder. The files are written in plain text using a JSON format. They can be opened and edited using a simple text editor (e.g. TextEdit, Notepad) and are easy enough to understand and modify. The original names and locations should be kept, as the software is looking for these exact files at the exact location. Descriptions of each file and contents follow.

App Settings

Filename: ​**HILO\_AppSettings.json**

This file contains basic settings for the app’s behavior and appearance. This includes the size of the viewport window and a reference for the file from which the app’s text is loaded.

* **prefPort**​:thenameofthepreferredserialport,withindoublequotes.Forinstance **"COM4"**​ on Windows or ​**"/dev/tty.usbserial001"**​ on Mac OS. You can leave it blank, as just a pair of double quotes ​**""**​. Whenever you successfully connect to a HILO machine via serial port, the software sets that port as the preferred port and updated the settings file.
* **fullscreen**​:ifsetto​**true**​(noquotes)theappwillstartinfullscreenmodeandignore the ​**width**​ and ​**height**​ settings. If set to ​**false**​ (no quotes) the app’s window size is determined by ​**width**​ and ​**height**​.
* **langFile**​:thenameofthefile(inquotes)containingtheapp’stext,bydefault **"HILO\_AppText\_EN.json"**​. It would be possible to use a different file, so the app can be translated to other languages.
* **width**​and​**height**​:positiveintegernumbersdeterminingthesizeoftheapp’swindow, in pixels. If the ​**fullscreen**​ option is set to ​**true**​ then ​**width**​ and ​**height**​ are ignored.

Machine Settings

Filename: ​**HILO\_MachineSettings.json**

This file contains settings for the HILO machine. Every time a connection is established, these values are set in the machine. The values can’t be changed in the UI. They are pre-determined through tests using the in-house test app.

* **stepsPerCm**​:numberofstepsinthedraftingmotorcorrespondingtoonecentimeterof yarn. A positive floating point number, such as ​**12.34**​ or ​**8.0**​. You can determine this number by dividing the total number of steps in a full rotation (typically 200) by the perimeter of the drafting roll, in centimeters.
* **deliverySpeedSteps**​:thespeedinstepspersecond(positiveintegervalue,like​**300**​) for the delivery speed. Essentially, the speed at which the machine is running.
* **draftingSpeedPercMax**​:thehighestvalueforthespeedofthedraftingroll,as a percentage of the delivery speed. A positive integer between 0 and 100, like ​**70**​. This corresponds to the drafting speed for thick yarn.
* **draftingSpeedPercMin**​:thelowestvalueforthespeedofthedraftingroll,asa percentage of the delivery speed. A positive integer between 0 and 100, like ​**20**​. This corresponds to the drafting speed for thin yarn.
* **spindleSpeedStepsMax**​:thespeedinstepspersecond(positiveintegervalue,like​**700**​) for the spindle, corresponding to the maximum amount of twist in spin mode.
* **spindleSpeedStepsMin**​:thespeedinstepspersecond(positiveintegervalue,like​**200**​) for the spindle, corresponding to the minimum amount of twist in spin mode.

HILO Devices

A HILO device is represented as an abstract in the app, as a ​**HILOInterface**​ (found in the tab **HILO\_BaseClasses**​). This defines a common set of operations for any HILO device, and allows us to use an actual device (communicating through a serial port); or a “simulator” so that we can test and demonstrate the app without needing a machine or other hardware. In the future we may wish to define other devices such as remote/networked HILO machines and clusters, or different versions of the HILO machine.

The simulator acts for most (but not all) purposes like an actual machine. As mentioned above, it is meant to replace an actual machine during app testing and quick demos. The simulator class ​**HILOSimulator**​ is defined in the tab ​**HILO\_Simulator**​, whereas the actual HILO (which connects to a serial port and communicates with a machine using the HILO protocol) is implemented as class ​**HILODevice**​ in the tab ​**HILO\_Device**​.

A device or simulator should be assigned a callback handler (implementing the **HILOCallbackHandler**​ interface, see the ​**HILO\_BaseClasses**​ tab) which gets notified of changes in the state of HILO and acts accordingly. The actual handler used by the app for HILO events is an ​**AppHILOCallbackHandler**​ (implementing the ​**HILOCallbackHandler**​ interface) and can be found in the tab ​**App\_Callbacks\_HILO**​.