User Interface for the Laser Acceptance Tests of the Quartz Bars for the EIC's DIRC Detector

Tyler Lemon, Mary Ann Antonioli, Peter Bonneau, Aaron Brown, Pablo Campero, Brian Eng, George Jacobs,
Mindy Leffel, Marc McMullen, and Amrit Yegneswaran
Physics Division, Thomas Jefferson National Accelerator Facility, Newport News, VA 23606

July 25, 2023

This note presents the user interface developed for the laser acceptance tests of the quartz bars for the EIC's DIRC detector.

The laser test station for the quartz bars will use an Arduino-based photodiode readout device to measure current responses. A linear motorized stage will move the quartz bar under test in to the laser beamline and another will move the quartz bar out. During initial alignment and characterization of the test station, photodiode values, stage positions, and the in-or-out-of-the-laser-beamline status will be logged. To facilitate data logging, a user interface, Fig. 1, has been developed.

The user interface was developed in Python with the library *Tkinter*, which provides an interface to the user interface software, *Tk*. With *Tkinter*, the user interface was created by programmatically placing widgets—*Tkinter's* nomenclature for items placed on the screen— and configuring the widgets for control or monitoring of the test station devices. There are widgets for ADC connection and readout; stage connections, control, and readback; bar position indicators; and log file setup and logging action.

The widgets for logging are labeled A1 and A2, Fig. 1. A1 is a text box and a browse button. The text box can be used to input a datalogging file path or the browse button can be used to open a file browsing window to navigate to and select the file. A2 is a *LOG* button that initiates logging.

Once the *LOG* button is clicked, the Python program checks whether a log file path was entered in the text box; if no path was entered, a pop-up window is displayed stating that no log file path was entered. The pop-up window asks whether to halt data logging or continue with a default file name *YYYY-MM-DD_hhmmss.dat*, where YYYY-MM-DD is the year, month, and date, and hhmmss is the current time in the 24-hour format with no separators between the hours, minutes, and seconds. Once a file path is input or the default file name is opted, the program checks whether the file exists. If the file does not exist, another pop-up window opens notifying that the file does not exist and enquires whether to continue by creating the file or to stop the logging process. If the response is *continue*, the program creates a new file and logs the photodiodes' currents, stage positions, and whether the quartz bar is in or out of the laser beamline; data is in CVS format. After writing the data to the selected log file, the file is closed. This procedure of opening a file whenever the user wants to write to it and then closing it once the write action is completed was chosen to prevent data corruption if the overall program is forced to close by an unforeseen error, or by forced closing of the program through Task Manager or a request at the terminal.

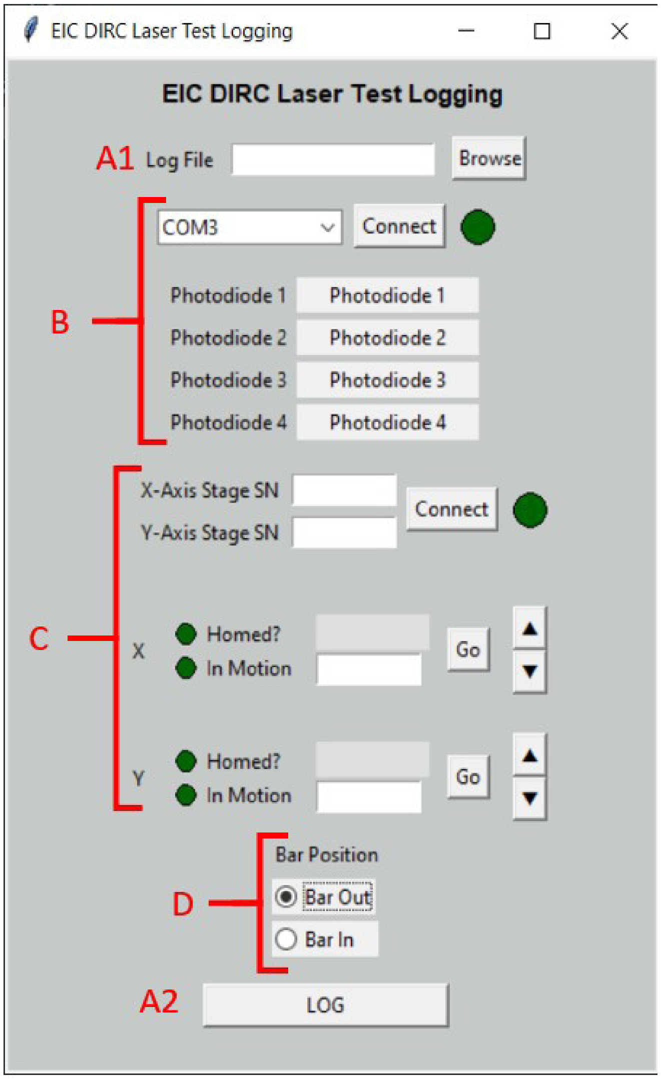

FIG. 1. Screenshot of the user interface, prior to connection to any device.

The widgets for configuring the connection to the photodiode readouts and displaying the photodiode values are labeled B in Fig. 1. The photodiodes' responses are quantified using ADS1115 16-bit ADCs read by an Arduino Uno microcontroller that is accessed by the user interface over a serial connection. Once a selection is made from the dropdown box and the connection is made using the *Connect* button, the photodiodes' data will be displayed in the appropriate widgets.

For control and readback of the motorized stages, several widgets are used, labeled C in Fig. 1. Stage controls enable selection of the axis by entering in the stages' serial numbers (SN). Once the user interface successfully connects to the stages, their statuses are shown by Boolean LEDs and numerical indicators. Each stage has two Boolean LEDs. One LED indicates whether the stage has completed the initialization homing process—moving to its lowest position and using that position as a reference as to where zero is on the stage. The other LED indicates whether the stage is in motion. Additionally, each stage has a numerical readback and numerical input for monitoring and controlling the current location, after pushing the *Go* button, and each stage has a set of up and down arrow buttons to initiate a "jog" action on the stage to move it in or out of the laser beam by a preset increment.

Label D, Fig. 1, of the user interface allows the selection of *Bar In* or *Bar Out* of the test beam; this control widget is only used to indicate in the data file the quartz bar position when the log button is clicked.

To conclude, the user interface for monitoring and controlling laser acceptance test equipment has been developed. During initial setup, alignment, and testing of the setup, the user interface will be used to log photodiode current values, quartz bar position, and linear motorized stage positions to a CSV formatted file, which can then be input into the analysis program to calculate the percentage of laser light transmitted through the quartz bar.