

Student: *Craig Harrison*
Advisor: *Dr. Richard Eason*

Progress:

This past week has seen great progress in terms of the getting the new Linux code working with the trimming machine. The new software had previously been timing out when the first blocking command, a Find Index (FI) command, was sent to one of the stylus motors. In the functioning DOS code this is a short wait while a motor moved to its home position; under the Linux software the motor was not moving and therefore never reached home. Comparing the I/O logs of the working output and the Linux code up to the timing out command revealed nothing: both of the software systems were sending the same commands and, even more telling, receiving the same replies. However, when I displayed the raw motor controller replies to a Tell Position (TP) command next to what the software was returning from our `getRawPosition()` function I discovered that they did not match up. It turns out that the code was casting the raw array of bytes returned by the card into a more sensible array of structures. However, the GNU C++ compiler was padding the first element of the structure where the Borland compiler did not. Some web searches turned up a “`#pragma pack(1)`” directive that allows us to set the packing level of portions of the code. Upon incorporating this change the machine was able to home all but one of the motors successfully, as shown in Figures 1 and 2 below.



Figure 1: Motors in starting positions



Figure 2: Stylus and X-Axis in home positions

New Motor Controller Cards:

The computer currently connected to SST machine in Lewiston is currently refusing to recognize that it has a hard drive, so in the down time I have been begun work on wrapper functions for the new motor controller cards. Currently our code is attempting to use the old motor controller cards with new software, but eventually these old cards need to be phased out in favor of new PCI cards. The card manufacturer claims that the new cards are backwards compatible, so for the most part the transition

will be seamless. However, the new cards have a supplied driver and library which we would like to use yet is not compatible with the old API. To this end I have been writing wrapper functions that will translate the old `dcxcmd()` and `dcxrpy()` calls into the new `pmccmd()` and `dcxrpy()` calls, respectively.

New Graphical User Interface:

Two days this past week were spent looking into GTK as the possible core of our new touchscreen GUI interface. After reading a portion of the extensive `gtkmm` online documentation I wrote up a small test application using the `gtkmm` library (the C++ wrapper for GTK) and the Cairo drawing library. The framework seems very well suited for what we wish to do; the only difficulty that I foresee involves the way the SST software demands to be in control its process nearly 100% of the time, requiring our GUI to be running as a separate thread as it wants to be in complete control as well. The most feasible solution may be a Unix pipe between the two separate processes, as well as possibly the use of some of the undefined signals. A screenshot of the demo GUI written this week is shown in Figure 3 below.

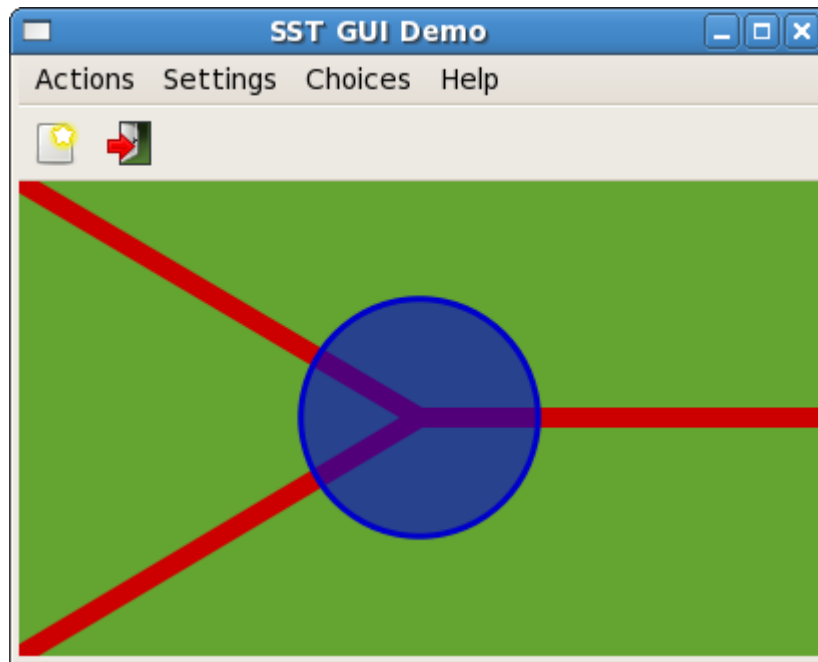


Figure 3: GUI demo of `gtkmm` and Cairo