

**Week 1: Research Report**  
Student: *Craig Harrison*  
Advisor: *Dr. Richard Eason*

This is the first report on the redesign of the software and computer system portion of the SST (Scour Shave and Trim) robotic shoe sole trimmer. The robotic trimmer and controlling software were first designed over thirteen years ago; since that time the 66 Mhz processor, DOS operating system, and ISA buses the software was originally designed for have become obsolete and difficult to repair. This project is focused mainly on porting the DOS based Borland C++ software to the GNU C++ compiler under Linux, rewriting portions of the software to use an updated motor controller card, and redesigning the graphical user interface and error handling mechanisms to make full use of the Linux system. These upgrades will enhance the usability of the system and allow it to function in the face of new technology for years to come.

The majority of the development time this past week has been spent replacing DOS and/or Borland specific code with standard or POSIX compliant C++. As this project was originally completed before the C++ standard library or even the C++ language itself was standardized, much of the existing code makes use of Borland specific libraries. Some libraries, such as `cstring.h` and `stringstream.h`, had nearly compatible alternatives available in the standard library; however, many other libraries were missing entirely. The missing libraries involved such important features as precision timers and the graphics routines.

In order to correctly replace the Borland functions and classes with routines available on current system the full collection of Borland 4 reference manuals as well as the Turbo C reference manuals were employed. Between the reference materials and observing the context of missing routines it was possible to search locations such as <http://cplusplus.com> and the Linux man pages for libraries with compatible functionality. It ended up that all of the existing code that made use of the old non-standard C++ classes were able to be rewritten using classes from the ANSI C++ standard library and POSIX C functions. It is hoped that by adhering to strict standardization rules this software will be capable of running on the majority of POSIX compliant systems far into the future.

Current work is focused mainly on replacing the old input/output code with software that emulates the majority of the old functions on present-day hardware. Eventually the majority of this will be rewritten and improved upon, but the initial goal is to be able get a port that is as close as possible to the existing functional software running on new machines. To this end a motley assortment of different methods are being applied. Keyboard routines such as `kbhit()` and `getch()` have either been pulled directly from, or are rewritten using, the standard `ncurses` C library. A new, portable graphics class has been written using the cross platform Allegro library. The DOS graphics routines previously provided by Borland's Turbo C have been replaced by small routines that utilize the new graphics class to provide the old functionality.

The motor controller cards currently in use by the old system are no longer commercially available; luckily the newer cards still employ a backwards compatible API. The difficulty here lies in setting up the Linux system with drivers to communicate with the motor controller PCI cards. In order to best test these new drivers an older Linux computer with the now obsolete ISA slots is being set up to use the old hardware with new software.

**References:**

(1987). *Borland C++: Programmer's Guide*. California: Borland International, Inc.

*C++ Library Reference*. Retrieved June 5, 2008 from <http://cplusplus.com/reference>

Shawn Hargreaves (2007). *Allegro: A Game Programming Library*. Retrieved June 9, 2008 from <http://allegro.cc/manual>