

# CS452 A0 Documentation

Jason Sun 20387090

May 11, 2015

## 1 Program operation

To operate the program, first load the executable in redboot and execute it:

```
load -b 0x0021800 -h 10.15.167.5 /u/cs452/tftp/ARM/j53sun/kernel.elf
go
```

Valid commands are:

```
tr train_number train_speed
rv train_number
sw switch_number switch_direction
```

To build it from source, go to cs452/kernel folder and run the makefile:

```
j53sun@ubuntu1204-004:~/cs452/kernel$ clear ; make libs && make clean && make
```

This automatically builds and copies it to my directory

```
/u/cs452/tftp/ARM/j53sun/kernel.elf
```

## 2 Program structure

Each pass of the polling loop:

- Checks the clock. It uses the 32 bit timer (timer 3). Every tenth of a second the clock is redrawn. The polling loop runs about 1600 times per tenth second under no load, and about 1500 times when holding down carriage return.
- Polls the train sensors for changes (incomplete functionality). Sends a new sensor data request when all data bytes from previous sensor data request have been received. The data is then put into a buffer, which is allocated on the stack. I have gotten sensor readings, but this breaks the sending of other commands.

- Checks the send buffer. Sends a character from the send buffer, if there are items in the buffer and its timing delay has been reached. Two different timing delays exist:
  - One for the display terminal (115200 baud). A delay of 10 loops are used. The ARM process operates at 200Mhz, and the serial is 115Khz.  $115/200,000=0.000575$  bit over terminal per instruction, so 0.005 instruction per tenth second. Using 1500 loops per tenth second,  $0.005 * 1500 = 7.5$  CPU cycles as lower bound.
  - One for the train controller (2400 baud). A delay of 1500 loops (roughly 0.1s) are used to make sure the train controller isn't overwhelmed with commands.  $115200 / 2400 = 48$  times slower, we should send no more than 48 commands per second.  $1 / 48 = 0.02$  and so waiting 5x longer should be enough.
- Checks the receive buffer. If character is received, it is printed to terminal. User input is sent to be parsed if the user presses carriage return. Parsing validates the commands before filling the buffer to train with commands.

## 2.1 Known bugs

- Backspace glitches when space is pressed and the user attempts to backspace the space. The way the input buffer is written is it simply tokenizes user's input based on spaces pressed.

## 3 Source code location and md5 hashes

The root directory is

/u1/j53sun/cs452

And md5 hashes

55a8e491be14b5a6c2ab2c4e18dff3aa	kernel/kernel.a
cde2e8ae7f7fd8be78167195ccab647a	kernel/kernel.c
dd49706f2421250533861efb4b571884	kernel/kernel.h
6898c4e700229324a782fb52577107c0	kernel/Makefile
8f13fb8bede40fa7fa47de95f5bf4d8d	kernel/menu.c
3920e98f9f3fd8d0acf74d6de6be1530	kernel/mysteps.txt
9334189d105271131c9ba7b08a4215af	kernel/orex.ld
d1bf2d35ebbb3eaf222de8edec6dc22e	kernel/timer.c
dba36a6a6d4aa0e4df9f55293e49476e	include/bool.h
791a92535ef5baa9546ddc3de017db34	include/buffer.h
d32dda3f6cd59b210c03d1ed8332c581	include/bwio.h
c343907f259c821eb6f910d32ca325cd	include/circ_buffer.h
87e9fe1c0b91c71156b20cb17d045b2d	include/menu.h

```

3a55d4201635b4972cd319a4bbd5d6a9 include/screen.h
2f9e34c2a1e730561ce059cf3694d690 include/termio.h
c4b75907720d8d0c206f853a6b234f68 include/timer.h
6a6bbef2cca529750f7084c90f1769f2 include/trains.h
d5ae7a7af25c09b1ada872424e22c766 include/ts7200.h
6abec0cf31b762a695b5c23a223bbdcb src/buffer.c
88b4539f5cd0de2a8b055e30c06c7465 src/bwio.c
53daf42ef0d841cf245afd53cfd405cd src/commio.c
e17ae063330185a3a0a51a66b0cf11da src/Makefile
de032abed4c3344dd8b3ec0f73bd1d20 src/menu.c
a64100478c8fc0396e5073997cf7a31c src/screen.c
d3b035d2812ec7262eb731bfc93708d7 src/termio.c
80f9969727b6b4133a1348fe47d74a9f src/timer.c
d9857544f07c77e962093a557b60c0c9 src/trains.c

```

## 4 List of all files submitted

All the files in 3, plus this design document (pdf).

## 5 Question answers

### 5.1 Clock timing

Clock does not miss updates or lose time because the clock is updated every tenth of a second, and that gives the CPU about 1500 polling loop iterations.

### 5.2 Hardware to reply to a sensor query

Hardware replies to sensor query is broken. I can read the sensor data, but that breaks my sending output and I think I'd rather submit without the bug that makes my program unable to control the trains.