

CS452 Kernel 4

Christoph Ulshoefer (20751216) and Thomas Broadley (20522223)

February 16, 2018

1 Running the kernel

Execute the following commands in RedBoot to start the kernel:

```
load -b 0x00218000 -h 10.15.167.5 "ARM/csulshoe/k4.elf"  
go
```

Our K4 ELF file is stored at `/u3/cs452/tftp/ARM/csulshoe/k4.elf` in the CS Student Computing Environment. Our K4 source code can be found at <https://git.uwaterloo.ca/csulshoe/cs452/tree/k4-done>.

The kernel initializes for the first four to five seconds of execution. Please don't quit the kernel during this period. If you do, you will likely see an assertion failure on the next run.

2 Commands

Command	Description
<code>tr <train_number> <train_speed></code>	Set a train's speed.
<code>rv <train_number></code>	Reverse a train.
<code>sw <turnout_number> <turnout_direction></code>	Switch a turnout.
<code>go</code>	Start the train controller.
<code>stop</code>	Stop the train controller.
<code>q</code>	Quit.

3 Implementation

3.a Interrupt-driven I/O

The operations required to make interrupt-driven I/O are divided as follows:

- Kernel
 - Initialize UARTs and VIC.
 - When a task calls `AwaitEvent`, enable the appropriate interrupt in the UART.
 - When a receive interrupt occurs, disable the interrupt in the UART and ready the task that called `AwaitEvent` on that interrupt type (if necessary).

- When a modem status interrupt occurs and the CTS bit is set, disable the interrupt in the UART. If a transmit interrupt occurred previously, ready the task that called `AwaitEvent`.
- When a transmit interrupt occurs, disable the interrupt in the UART. If a modem status interrupt with the CTS bit set occurred previously, ready the task that called `AwaitEvent`.
- Notifier
 - Call `AwaitEvent`.
 - On returning from `AwaitEvent`, send a message to the appropriate I/O server - we set up the `AwaitEvent` semantics so that notifiers always have a UART ready to transmit/receive characters. This design allows us to use relatively generic notifier and server implementations.
- Receive server
 - Call `Receive`.
 - If the received message is from the server’s notifier, read the data from the UART, then send it to the first queued task. If there are no tasks queued, store the data in a buffer.
 - If the received message is from a task that called `Getc`, respond with the first buffered byte of data. If there are no buffered data, queue the task.
- Transmit server
 - Call `Receive`.
 - If the received message is from the server’s notifier, write the first buffered byte to the UART. If there are no buffered data, keep the notifier reply-blocked.
 - If the received message is from a task that called `Putc`, write the byte to the UART and reply to the notifier if the UART is not transmitting. Otherwise, store the byte in a buffer.

3.b A0 task structure

We created four I/O servers, one for each element of $\{\text{train, terminal}\} \times \{\text{transmit, receive}\}$. Each server has a notifier.

At the beginning of execution, the first user task creates a switch resetter. This task switches all turnouts to curved except for 154 and 156.

The first user task handles user input. Once a user has typed in a command, the first user task sends it to the command dispatcher, which writes bytes to the train or spawns a task to execute the command.

In particular, we spawn separate tasks to reverse trains and switch turnouts. These actions require timing multiple sends to the train over a period of milliseconds or seconds. To do this, these tasks call `Delay`. If the command dispatcher instead called `Delay`, it wouldn’t be possible to accept user commands while a train was reversing or a turnout was being switched.

The command dispatcher also sends a message to the track state controller on each command. This task is the central repository of information about the track and trains, including train speeds and directions, turnout states, and sensor data. Another task, the sensor data secretary, reads sensor data from the train receive I/O server and passes it to the track state controller.

Finally, three view tasks write to the terminal: the clock, sensor, and turnout views. The sensor and turnout views poll the track state server every 60 milliseconds, then update the view when the track state changes.

4 Extra features

4.a Building with GCC 7

For the CS Student Computing Environment, we switched to GCC 7. If you wish to build the kernel yourself, you need to download this new version (available [here](#)) and unpack it in your `$HOME` directory. Using link-time optimization, we get a 67% speedup compared to GCC 4.0.2. To make link-time optimization work correctly, we needed to steal a script from the Linux kernel (`gcc-ld`). We included the necessary hints that we used other authors' code in the repository and even included Linux's GPL.

4.b PutBytes and Printf

We realized early on that we would need a way to write multiple bytes atomically to a UART. In our K4 implementation, each UART is written to by multiple tasks. If we called `Putc` repeatedly to send long strings of characters, bytes from different tasks would be interleaved, resulting in garbled output.

We implemented two primitives to allow for atomic multi-byte writes: `PutBytes` and `Printf`. `PutBytes` copies a given number of bytes from a character buffer to the transmit server's buffer. `Printf` does the same, but takes a null-terminated formatting string and variadic arguments instead.

4.c Task descriptor reuse

Our kernel now allows new tasks to reuse zombie tasks' task descriptors. New tasks reuse the first task descriptor not already in use, which is determined by finding the most significant 1 in a bitmap. To make sure that task IDs in userspace are unique for the entire runtime of the kernel, each task descriptor has a generation that starts at 1. When a task is destroyed, its generation is incremented. The user task ID of a task is calculated to be `generation * MAX_TASKS + kernel_tid`.

4.d Multiple reverse

When the user inputs a reverse command, a task is created that will reverse the train. This task sends a stop command for the train, delays for an amount of time based on the train's speed before stopping, reverses the train, and re-accelerates it to its previous speed. Since each reverser task operates independently, multiple trains can be reversed simultaneously. This design would not have been possible without the ability to reuse task descriptors.

5 File and repository hashes

The commit before adding this report had SHA `6ce41646c2e800872744d16a7835f52bf16fe77f`.

5.a MD5 hashes

Here is a list of the MD5 hashes of the files included in our repository.

```
git ls-files | grep -v 'test/googletest' | xargs md5sum
```

3baa32ac2b0d0342c6975830905a3da0	.gitignore
4bbdec35a4b9dc9ca8615b1e321de502	.gitmodules
cf46893a7a1e5a2258d7a55c26e88647	.travis.yml
cdbd7fbeb502f2adcba5aeb12b7e8e8c	Doxyfile
a8104cb719e6b2ae7f775a66dae8976e	Makefile
8dcf3e360da847b858398e8eb0d610c7	PULL_REQUEST_TEMPLATE.md
ce94962b2dddb2290171768c11a7376	README.md
ffffb471c0ef031de7290e9803a651aac	gcc-ld
d7810fab7487fb0aad327b76f1be7cd7	gcc-ldCOPYING
1773eb6cf890af6f5502d958d405b2d6	include/bwio.c
1506ac36c03fc3c02a62124aec4dbf20	include/bwio.h
1e06c47ed88775686330932eb570b056	include/io.h
420d4fdff40e298e855ad37d87af6197	include/myio.c
93cdb176a9db6bf6073d3f9ec5996df7	include/myio.h
0ff6bee28590e1b949a208361291deab	include/mytimer.c
0776bbc8e09735c39ee6ad1a5465b08b	include/mytimer.h
b48706b75307a44de39b64a85a4297a0	include/rawio.c
2314714decf98c1fde63b96f2ffaf2ab	include/rawio.h
a51b441bd12f84e40333d2dfde291780	include/timer_data.h
74c63a11fc85af6c146dbcb5fa19b14a	k2benchmark.txt
51f1ce428a9a07d7b7b394f9c8302279	kernel/asm/kernToUser.s
d1d5fe7833e97870de0e1539f8581727	kernel/asm/startup.s
dc310ba0937f8818a8735c4fed086051	kernel/asm/trap.s
770dc066a65a334181fb303745f14dad	kernel/include/labenv/timer.c
a696fe702d49142a7a1d9961e0817256	kernel/include/labenv/timer.h
38cfc8c969ea2246fc27635304fd1947	kernel/include/labenv/ts7200.h
ee90c5ee9fadbe5f7cf574db5241d256	kernel/include/versatilepb/timer.c
eb6c4ecdbda72702ca44cc8132804d9b	kernel/include/versatilepb/timer.h
bb1fd1bb42d39293c3ae09aeaba5636a	kernel/include/versatilepb/versatilepb.h
0421e442c4abc4951d2224f4d88ad8eb	kernel/src/cp_vec.c
503fb56ff1929bcc8d7a6b3d19b2524e	kernel/src/cp_vec.h
a37742c39070e70a5836f3855627a8a1	kernel/src/events.c
ba6a3c142b44633cc80c765f36939f62	kernel/src/events.h
f3d286fecda3d548e6dadebb162c9454	kernel/src/handle_abort.c
21d914b9ba8147299149e9f527e21b2c	kernel/src/iio.c
aaa4154d4b5ab18bb78651cfcb13f886	kernel/src/iio.h
a4d4ae8c486bb55716afef1215b77818	kernel/src/interrupt.c
f105abe4e0d376eb9a0e700e20a9e626	kernel/src/interrupt.h
a0946900f956766d853fa86ea49f1628	kernel/src/kassert.c
8baadeeeeb32c820cce4b6420efcb92b	kernel/src/kassert.h
c7f303448c61cd0b620060ef10915cb8	kernel/src/kusage_stats.c
c1ed6afa70cb921f87c0e90770edb87b	kernel/src/kusage_stats.h
56ff6f6f508b29ede42e0cf2831ca2c7	kernel/src/multitasking/messaging.c
8da39e0fbf99d840a126cad76ea4ed57	kernel/src/multitasking/messaging.h
d8f5950a0cac94759ad5200fb735ddd1	kernel/src/multitasking/ready_queue.c
acdaabac90092feb2d815d93daa6c160	kernel/src/multitasking/ready_queue.h
149c5145f09e141dd7a5c68ba1bd3c2f	kernel/src/multitasking/schedule.c
a11a9befd765f6c6c048f6b84860d0f9	kernel/src/multitasking/schedule.h
5864d87dbe1c10fd564ccd73a3b06a6d	kernel/src/multitasking/scheduler.c
2a6005ff964276bd1f83ba93f2e06b86	kernel/src/multitasking/scheduler.h
1ff609f65cd9531a647ce6575c039b6f	kernel/src/multitasking/send_queue.c
d2fd55f45ab57e8e5c75a5a072bb74ec	kernel/src/multitasking/send_queue.h
88d1923ffec75d0b7f28afa37b1eae2	kernel/src/multitasking/task.c
b20b570ef1305a2e44efb45b8919064f	kernel/src/multitasking/task.h
4ad860fd0d43e6a9cf29eacf89e2b06d	kernel/src/queue.h
689635daad9af76d40efb0d072e9c984	kernel/src/syscall/syscall.c
0d26653864f2251c8a286335fc5d4c2a	kernel/src/syscall/syscall.h
8c6fd98546a9c2e609b33e3acf63f84a	kernel/src/vic.c

f694733a3095dac0e74e18521ea1aa17	kernel/src/vic.h
910ddd9774bc89c04f804f76669354ce	lib/a0codes.h
bec5edf5b40af4209c289f8c83f248ef	lib/attributes.h
59506a2fc8c1e78f1287e2d9eb454b34	lib/benchmark.c
3b7d15e91a197c4a1a4cdfbe674f88b1	lib/benchmark.h
a5c0e34dc13c5856ce5572101ebe778b	lib/buffer.h
7949f4bf8c52321c0bcf49e3b67817ff	lib/buffertypes/char_buffer.c
8f75f97be05b4e5fa3ceb739f3a6d61b	lib/buffertypes/char_buffer.h
0dd2910bf5314d67758f17c17ef855ee	lib/buffertypes/int32_t_buffer.c
81ddf219ea1acff14e8237a88742c258	lib/buffertypes/int32_t_buffer.h
950644b2e998ed44f473d921d5db044e	lib/codes.c
dbcecff713b5e55b43f3a44db4d006a7	lib/codes.h
d41d8cd98f00b204e9800998ecf8427e	lib/constants.c
06e84818bb23369072347e52dc27ab46	lib/constants.h
38424f2856725ba4ce4919eefe5e3142	lib/crash.h
608c3732de6719e206bf1c76772bd549	lib/crash.s
dd7ed407227be8ff5e72530a2e86ada4	lib/event_data.h
471dbe582614af849e8bf58bc16458df	lib/messages.h
ca86a48b51092733767b9cd6e4e341b4	lib/priority_queue.h
637a0b453266e950654aaada20cc75b1	lib/terminal.h
af9cdf9014bec08c6df74cead3ca105d	lib/track/README.txt
7707b57cb3a6c3703c0e3da3acbef763	lib/track/parse_track.py
1e06c8505279ed9f4d10136465c81618	lib/track/parts_tracka
a408ef1736b356024bf0dc2bc05d98b0	lib/track/parts_trackb
b3aa33d41e5837b8f82619f71c114b2c	lib/track/track_data.c
aa902c027517deb27775e25b3797d207	lib/track/track_data.h
350483959800639957e7d760bff9e7a4	lib/track/track_node.h
50e0b1150b39a2425cb2180c33e0e57f	lib/track/tracka
dab2764f1d7f07aa454bad2ec01158b8	lib/track/trackb
0ef99a8a75c948a16a9da24712286b4c	lib/tststdlib.c
654f8e7b011be9480c2a5b42bf927957	lib/tststdlib.h
e80f2789ca24d327bac466b79ccc5290	lib/tstring.c
2a7117fdcad90f1e3029ad923292f15	lib/tstring.h
06e369b76c2d125b61ac6a95d7d31309	lib/usage_stats.c
2c7495e2cf58fe43659b7ac7fed0b7ee	lib/usage_stats.h
3ae763c0ae09e92d537e6ee8c104c720	lib/user_command.c
82701933a67d4fe1780b19ed63b7ed05	lib/user_command.h
065363a2bb67430493d34347fcc888f5	main.c
04a99fa4e57230b2b2b7a525a1105097	main.ld
d9cd792c1413ac79bf45f799ae903f42	reports/a0-csulshoe.pdf
2d81bda8e6453ea87e964edefb8d37c5	reports/k1.pdf
3c476edf2886bdb295d9f5e37a748d9c	reports/k2.pdf
e962ebaab43701b2baa697adfbe05f7d	reports/k3.pdf
1c7a429e1ced54dbb23098d3078a30e8	stats.md
da4d2d5a9d3d7971c83928f051ef6b58	test-resources/assert.c
c1e9354e62486019325b3d7e096f17b4	test-resources/assert.h
931cdaa500cc0a45aaabe98149ec389b	test/Makefile
d41d8cd98f00b204e9800998ecf8427e	test/e2e/__init__.py
13944380c0629327e27c828f8ac39c06	test/e2e/qemu_tcp_wrapper.py
f1e50c9027d900c43570595e642fd518	test/e2e/snapshots/k2.txt
57e0cf79450fa3e8dc465928d0e82668	test/e2e/snapshots/k3.txt
2135a39d326e969a7a1aff8ed796464c	test/e2e/snapshots/printf_happy_path.txt
5c894b95e44a5b8d0e850fc07fa45f93	test/e2e/test_clock.py
dfc8a0cafcb834fad4ee9896366d132e	test/e2e/test_commandparsing.py
8f749bae092472967cd325ee379e4d03	test/e2e/test_getcputc.py
4b9ad74f1f6d4a0c985c47d3a0803d96	test/e2e/test_interrupts.py
939ce9936dc62dba256a3a3b18f4ecd2	test/e2e/test_kernel_demo.py
207f9c96a203fb1408236396eb8c4da1	test/e2e/test_messaging.py

63428d6fbf558a456bfd1503dbffb217	test/e2e/test_mypriority.py
b137b49b04d3fd3a55f7aa94d500f799	test/e2e/test_nameserver.py
705f20c8af9c2bb6be26efcab7a8caac	test/e2e/test_printf.py
149af40366d412a08de9514a974823cb	test/e2e/test_segfaults.py
30e544606327baabd6afe99c229faf9f	test/e2e/test_test.py
3834f5e022525034f78b19b6d5b764fa	test/unit/test_all.cc
65a92d844a6d98056dff9024003fbfc	test/unit/test_all.h
916f5fab4c3bf56c62b8588863c9cd76	test/unit/test_buffer.cc
f5785dfb91f2c6c1050d6925a08f20c6	test/unit/test_buffer.h
584d33a90964dc237012ba401c475965	test/unit/test_clock_wait_queue.cc
825a054548d34057930fc5cbea517933	test/unit/test_clock_wait_queue.h
be7726ba1a922ec5ab26525114d7704d	test/unit/test_messaging.cc
f34aa46731d487e48a5cd1a9a681b518	test/unit/test_messaging.h
1909548396588628d189701a805e7bb4	test/unit/test_ready_queue.cc
917d4146f8727af1b9986cd4b94565a3	test/unit/test_ready_queue.h
ee004b6ffd40e6513204bdd28474af7f	test/unit/test_scheduler.cc
c31f7630705c36d377356eace6e86c9b	test/unit/test_scheduler.h
ebf57a23d6c248f7de1547e67b2900c7	test/unit/test_stdlib.cc
664c5ed933718c9c833e37a4663e6ef4	test/unit/test_stdlib.h
feb2d463f057a6d3b342f467e7682ce2	test/unit/test_task.cc
36889e34775aefba76462f7ede00349b	test/unit/test_task.h
0412ad5d98215489d254beeca159caf8	usr/clock.c
b4566bc63807ef8e5e47e70ae6140774	usr/clock.h
bf52251f80fa600ff9f49cc01f03c50a	usr/command_dispatcher.c
f0ef7e7e13932378e07f94fe95188da0	usr/command_dispatcher.h
c1a84bc2d0e1e77eda929b31f622dfb5	usr/idle.c
0850a1ff169d4bf316f23f252bd1a1a2	usr/idle.h
a2e207dab2f58b947d2427afd30b7414	usr/ioserver.c
a89d339c58c2eb8f9c54a6ced6845a70	usr/ioserver.h
686bfad6b5f7a70f10865e0bc29a5249	usr/k1.c
758b67d7c795e165ba195c0d39d6bbe0	usr/k1.h
b8923150be8054c9bfa75bd511f9bc21	usr/k2.c
d95a09d1de56dd554148b0b003d1a4a9	usr/k2.h
3efb1e872f6ea02246e5d87f3fccce8c	usr/k3.c
87e3b6299a0d9261e6a0a3dc7a0adaa2	usr/k3.h
fe6fba12efca85d680eb9e2e6dbff3a2	usr/k4.c
2a2c52643d0225988a451e1e8125d6a8	usr/k4.h
3f9b92e877e51c8a6556121920a974e6	usr/lib/clock_wait_queue.c
b9b4baf1f42585917df639bc883bdaf4	usr/lib/clock_wait_queue.h
f23ab005de96d571487ac46cd51672cd	usr/nameserver.c
3640a5039843896751af260a02e3a188	usr/nameserver.h
817140541814c39259444b4add668276	usr/notifier.c
a266f655714bf034e10ee80066a04045	usr/notifier.h
7c3d3b1b7553e85b5ac5c952b19eb0f2	usr/spam.c
c6b26aad9c868a310e2a61543f1b6d54	usr/spam.h
890ab352f3b5ace6b5ad00e693fd21b3	usr/test/awaitevent/test_rx_terminal.c
987024da0946d3bc51fa9b517aba1e4d	usr/test/awaitevent/test_rx_terminal.h
324862e41991fa243099d43763f7d490	usr/test/awaitevent/test_timer_interrupt.c
fd0230e03c6329d23cd59ce714d14841	usr/test/awaitevent/test_timer_interrupt.h
7609b1ef434e30e5c7d49477fb636423	usr/test/awaitevent/test_tx_terminal.c
c4db6de70684fda79da761d55c674458	usr/test/awaitevent/test_tx_terminal.h
31041ff5a0154e679718472f2cd150eb	usr/test/clock/clock_util.c
74a779c5688003aa346dc6d18d84a2d4	usr/test/clock/test_clock.h
4df05ca408a98b04ecb8e5a4647342ec	usr/test/clock/test_clock_accuracy.c
7b58840919f048032eab36dcaec632bf	usr/test/clock/test_clock_errors.c
7e5373809fd8b91e2c3aa4dee4872665	usr/test/clock/test_clock_syscall_accuracy.c
bed867b2ad5bb4aae38a223eb2a750ba	usr/test/clock/test_clock_syscall_errors.c
1377b67b1599f9e97f783013fecc80cc	usr/test/iio/test_getputc_errors.c

606ca77a18d32ae87856761098065e97
96b46eae0fce398869696ba43e175bf2
943f5683a7afd3a053e5a864aa586a35
091124f3277913c6339829b031478ea7
93d9372b059af217848ce09cf45e69c4
cfa501bd3d1d8e66354b98cc1a1b635a
d262f35cce581eda07aff2484fd2d8b5
4e0595b514cecd036282b86b0d1ddc02
497af6748eae1fdaf6a437d80dcce337
f0abb91510a1c1433da5977ba2a1881d
6ac01327ddbffa06649f0c20ed04e8ec
76583e962409b9a90c62df1e97b74598
47c902d889407fee469b052bec91585c
a17bf4e6f1b05c814f24c35fe3cd5677
c85eef4e381e81e765ec21f4154f21d3
7b973bb3f9a91d9cd292229d0c90f11d
8bd72ff578d51d007fe156e698e1b48c
7101f020db73ffcf486747bebc6150b
0aca389c63629487a337b5f89a773e08
2d3f90279df9fe86d7b8241d0e2f7a15
9ae182d2a7d7d6a327d70d6593eea0de
fbb92e62ce4cdab26d53efd17ac977c5
90a28211c3a519e8eb53142e3443de96
97e4576bd06081ac956acb15838a51f6
0e290966f99895d1a336e40a7857b5bb
a8e19df150c1fa62d1f8b68cd13ea758
3d35f49914dd58470f91b47088a74abc
83869951c54ac75ae2d68223a8cde7fd
40ff8628c0cf51b8fbc2bff1a2ad6113
0f8b0f3069f380632a16a6351bb719ae
c527c363cc04ca1bf6da967debf5e3f5
e22f4b33d4b2703e4fb73230ed9d8195
83053e202339e79cbc203b0570f028a5
2ba8bd8df7780278a29b2e2936e93918
2752991c97a04dbb594be1f0baae4e11
cb3c4395433173066bdebfc0540d4ede
1a63accf05f4a4f7624a088732419811
823190edf8b32d3ffa4ed773ba01c434
82c1bd94a9c389c479989179e8c59e4c
286264369949a8e9101ad0e77885a334
28c9374b520a70e56972308b9e7f2afd
133ee01a098e4b4ad5a4488512e14e8d
a29ce20a04fb0cf141e001ddb09a8a65
6cae62c97dc3e1b1de204974a295e55a
45499fb865c9b64dfaf169deb0522088
3ffdc9be852e4ad6ef8064d500fbb8da
594127a7d25cbe54d542bdcfec4e4867
c7ab09e45873699d5bbb333a7641b99e
5ec406d1cd0713cdd4a8cc3517645eb3
bde55ae9edbb30ed8dde9efd9692e9de
a7668b81a997cb9d602415320f5d33f3
3f052cb7f6b25c5e80dca05f284701d3
c06605bea54e7505a4ceadee05e53a6a
89441cccbea450f485e451770d95f95
03a9a6a402360d35fe9d1aedfd8e98e7
usr/test/iio/test_getcputc_errors.h
usr/test/iio/test_getcputc_happypath.c
usr/test/iio/test_getcputc_happypath.h
usr/test/iio/test_printf.c
usr/test/iio/test_printf.h
usr/test/messaging/test_messaging.h
usr/test/messaging/test_messaging_basic.c
usr/test/messaging/test_messaging_exit_with_blocked.c
usr/test/messaging/test_messaging_fifo_send.c
usr/test/messaging/test_messaging_invalid_tid.c
usr/test/messaging/test_messaging_receive_before_send.c
usr/test/messaging/test_messaging_reply_target_zombie.c
usr/test/messaging/test_messaging_same_priority.c
usr/test/messaging/test_messaging_send_recipient_zombie.c
usr/test/messaging/test_messaging_sequence.c
usr/test/messaging/test_messaging_tree.c
usr/test/messaging/test_messaging_truncation.c
usr/test/nameserver/ns_tid.c
usr/test/nameserver/ns_tid.h
usr/test/nameserver/test_nameserver_happypath.c
usr/test/nameserver/test_nameserver_happypath.h
usr/test/nameserver/test_nameserver_too_many.c
usr/test/nameserver/test_nameserver_too_many.h
usr/test/nameserver/test_nameserver_wrapper_errors.c
usr/test/nameserver/test_nameserver_wrapper_errors.h
usr/test/test_message_benchmark.c
usr/test/test_message_benchmark.h
usr/test/test_mypriority.c
usr/test/test_mypriority.h
usr/test/test_runner.c
usr/test/test_runner.h
usr/test/test_undefined_handler.c
usr/test/test_undefined_handler.h
usr/train/reverser.c
usr/train/reverser.h
usr/train/sensor_secretary.c
usr/train/sensor_secretary.h
usr/train/switch_resetter.c
usr/train/switch_resetter.h
usr/train/switcher.c
usr/train/switcher.h
usr/train/track_state_controller.c
usr/train/track_state_controller.h
usr/user_data_abort.h
usr/views/clock_view.c
usr/views/clock_view.h
usr/views/sensor_view.c
usr/views/sensor_view.h
usr/views/turnout_view.c
usr/views/turnout_view.h
versatilepb.ld
versatilepb_e2e.ld
versatilepb_e2e_iio.ld
versatilepb_e2e_iiotmr.ld
versatilepb_e2e_tmr.ld