CS 452 Milestone 1

Benjamin Zhao, Kyle Verhoog

Mar 08, 2018

Contents

Operation

The elf file is located at `/u/cs452/tftp/ARM/ktverhoo/milestone.elf` and can be initiated using the typical `load` command.

After `load`, wait a few seconds for tasks to initialize.

You should be presented with our interface featuring *windows*.

You can type commands described in the next section.

Commands

- `tr <tr#> <speed>` sets the given train to the given speed.
- `rv <tr#>` reverses the train.
- `rt <tr> <speed>` sets the track to be in a loop and starts the train moving.
- `go <start node> <end node>` tells the train to go to `<end node>`, `<start   node>` is not currently used for anything.
- `tk <tr> <starting node>` initializes the train with the system.
- `cls` clears the terminal.
- `cr <offx> <offy> <width> <height>` creates a dummy task with writes characters to its window to show off the window management.

Design

Train Track

Railway Manager

The Railway Manager acts mostly as an initializor for the track. It is responsible for spawning all track-related tasks and sharing common data like the track data structure.

Sensor Manager

The Sensor Manager is in charge of fetching sensor data from the train controller over UART1. It provides a subscription service in which tasks can subscribe to sensor events.

Sensor Manager puts to use `BLPutC` which is a blocking `PutC` to UART1 which permits it to know to a much better degree when the actual sensor poll byte has been transmitted.

Switch Manager

The Switch Manager manages all switch related actions on the track. This includes sending and maintaining the switch states of the track.

Other tasks can query it to gain information about the switches or to switch the states.

Train Manager

The Train Manager facilitates the basic train commands sent from the shell. It also is responsible for getting trains initialized with the Prediction Manager.

Prediction Manager

The Prediction Manager attempts to maintain knowledge about trains on the track. It attempts to track trains by keeping track of where they were last seen and are expected to be seen next. It can handle a single sensor miss, but otherwise is quite fragile.

For Milestone 1, it also manages calculating pathing data and sending required switch changes to the Switch Manager.

Display Manager

We figured that it would be worth it to have a structured way to present data to the screen both for presentation as well as for debugging.

Terminal Manager

Terminal Manager manages a set of windows. Tasks can request a window from the Terminal Manager and output from the task can be redirected to either its own window or a common-to-all-tasks logging window.

There is currently support (not enabled currently) to route input to windows other than the shell, depending on the cursor focus.

Shell

Shell is just the first task which registers to the terminal manager which is configured to accept input. It is currently a monstrosity which handles the parsing and executing of all commands.

IOServer

Blocking PutC

We added a blocking version of the `PutC` function which is very useful in applications like the Sensor Manager for when it polls to the train controller.

Milestone 1

Stopping

Calibrating

Pathing

Files/Hashes