Software **Test P**rocedure (STPr)

**for**

**North Shore Extension**

**Version 3.0 approved**

**Prepared by Garrett Grube, Jeff Deely, Ritesh Misra, Spencer Worms, Xavier Torgerson**

**Blue Team**

**12/15/2016**

**Table of Contents**

**Table of Contents**

**Revision History**

**1.** **Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and Abbreviations

1.4 References

**2.** **CTC Office**

**3. Track Controller**

**4. Track Model**

**5. Train Model**

**6. Train Controller**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Garrett Grube | 10/30/16 | Document Created | 1.0 |
| Xavier Torgerson | 10/31/16 | Track Model test cases | 1.1 |
| Garrett Grube | 11/2/16 | Train Controller Test Case | 1.2 |
| Ritesh Misra | 11/2/16 | CTC Test Cases | 1.3 |
| Spencer Worms | 11/2/16 | Train Model Test Cases | 1.4 |
| Jeff Deely | 11/2/16 | Track Controller Test Cases | 1.5 |
| Spencer Worms | 12/10/16 | Tested Train Model | 2.0 |
| Garrett Grube | 12/11/16 | Test Train Controller | 2.1 |
| Spencer Worms | 12/14/16 | Finished | 3.0 |

# **Introduction**

## **Purpose**

The purpose of this documents is to outline a testing procedure for each module’s core functionality.

## **Scope**

This documents conveys the test plan each group member has for their individual module.

## **Definitions, Acronyms, and Abbreviations**

|  |  |
| --- | --- |
| **Term** | **Definition** |
| User | Someone who interacts with the system |
| Module | A piece of the distributed system |
| Track Model | A virtual representation of the track and the nearby infrastructure |
| Train Model | A virtual representation of the train |
| Wayside Controller | A programmable logic controller for the track model |
| Train Controller | A system for interfacing with the train controls |
| CTC Office | A centralized station for routing trains and adding and removing track |
| Vital | A safety critical system |
| Block | A piece of the track |
| TODO | A abbreviation denoting temporary incompleteness; it will be finished in the future |

## **References**

"Google Java Style Guide." *Github*. Google, n.d. Web. 28 Sept. 2016.

"North Shore Extension Codebase." *Github*. N.p., n.d. Web. 16 Sept. 2016.

Profeta, Joseph. "Project Definition." Software Engineering Lecture. Benedum, Pittsburgh. 8 Sept.

2016. Lecture.

Wiegers, Karl Beatty Joy. "Software Requirements (3rd Edition) (Developer Best Practices)."

*9780735679665:*. Microsoft Press, 01 Jan. 1970. Web. 28 Sept. 2016.Overview

# CTC Office

2.1 **Test: Sending Suggestions**

2.1.1 Purpose

* Ensure that the wayside controller can properly receive suggestions
* Ensure that the CTC module can handle the wayside controller rejecting the suggestion and accepting the suggestion

2.1.2 Procedure

1. Set values for suggestion
2. Send safe and unsafe suggestion to wayside controller
3. Receive appropriate response and handle properly

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Tracker panel immediately updated with information of train leaving the yard | Train tracker panel displays the block number that the train is on immediately |
| Block status shows occupied when prompted to show block status for block adjacent to the yard | Block status shows up as occupied, with the field filled with the trainID |
| If the suggestion is rejected, no values are changed and appropriate pop-up shows up | Sent a suggestions while a train was still on the yard block, and nothing popped up/no trains were created. |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/12/2016 | Ritesh Misra | Pass |

* + - 1. Additional notes or encountered bugs



2.2 **Test: Changing Block Status**

2.2.1 Purpose

* Ensure that the CTC office can disable and enable blocks
* Ensure that the CTC office can properly deal with trains close to disabled or enabled blocks

2.2.2 Procedure

1. Enable a block with a stopped train right behind it
2. Disable a block with a moving train right behind it

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| In the case of enabling the block, the train should then be given the instruction to continue moving by the track circuit (indirectly through the CTC) | Was not able to test since the train would move right past disabled blocks anyway. |
| In the case of disabling the block, the train should be given the instruction to stop as soon as the block is disabled and should only be allowed to move when the block is enabled again. | Again, the train would completely ignore a disabled block and continue to move across the block. |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/14/16 | Ritesh Misra | Fail |

* + - 1. Additional notes or encountered bugs



2.3 **Test: Flipping Switches**

2.3.1 Purpose

* Ensure that the CTC office can flip switches when needed

2.3.2 Procedure

1. Set a train to move on the track right before a switch
2. Flip the switch position and check the train’s position after it crosses the appropriate block

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The train ends up on the appropriate block | Ran this on the beginning of the red line. The train waited at the yard until the switch was flipped, allowing it to move across.  Ran the same test again for switch 4 on the green line. The train ignored the switch this time, for some reason, and continued to go backwards in the direction of the yard. |
| The block occupied status is updated for the CTC and the appropriate wayside controller | In the case where the switch worked, the displayed information was properly updated. |
| If the switch needs to be flipped again for a new train coming through the track, it is flipped in advance | This does not happen, either through a malfunctional switch state suggestion from the CTC Office or an error in the Wayside Controller. |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/13/16 | Ritesh Misra | Fail |

* + - 1. Additional notes or encountered bugs



2.4 **Test: receiving block updates from Wayside**

2.4.1 Purpose

* Ensure that new block occupations are reported to the CTC

2.4.2 Procedure

1. Send a valid suggestion to the wayside controller
2. Look at the status of the block immediately after the yard
3. When that block is occupied, continue to the next block
4. Continue with the simulation speed set to a reasonable time

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Looking at the adjacent blocks continually shows the block being occupied and then not being occupied | Behaves as expected- the block status changes from occupied to unoccupied when the train moves (assuming the user refreshes when appropriate) |
| The speed of the train stays constant throughout the trial | The setpoint speed on the train GUI remains constant, but the actual speed fluctuates just a little bit. |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| TODO | TODO | TODO |

* + - 1. Additional notes or encountered bugs

2.5 **Test: simulation speed**

2.5.1 Purpose

* Ensure that increasing the simulation speed makes updates happen faster

2.5.2 Procedure

1. Run a simple suggestion through the system at 5 times speed. Measure the time it takes for the entire process.
2. Run the same suggestion again once the first one is over at 10 times speed. Measure the time it takes.

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The simulation should take half the time at 10x speed than at 5x. | The simulation is roughly two times faster, not exactly. |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Ritesh Misra | Pass |

* + - 1. Additional notes or encountered bugs



2.6 **Test: Train Rerouting**

2.6.1 Purpose

* Ensure that the CTC is able to handle the dispatcher requesting a new destination for a train already on a track

2.6.2 Procedure

1. Send a suggestion (preferably on the red track, since there are so many opportunities to reroute trains on it).
2. First, get the train to go to block K on the red track.
3. Once it has reached block K, tell it to go to block D.
4. When it reaches block D, change its destination from block K to block C.

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| When the train is on block K, make sure that its speed is 0. | The train does manage to get to K. |
| While the train makes its way to block K, the speed should be constant and the switch at A/E should be flipped to make it go to E. | The wayside controller does not toggle the switch by itself. |
| When the destination is changed, the switch at A/E should then be flipped and the train should make its way to block C. | This test condition was not reached due to the above malfunction. |

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Ritesh Misra | Fail |

# Track Controller

* 1. **Test: Train Route**
     1. Purpose
* Ensure that a train with a destination set by the CTC office will make it to it’s destination
  + 1. Procedure

1. Using the CTC instantiate a train and give it a destination

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The Track Model is updated with a new train near the yard | Pass |

1. Set the Track Controller’s UI view to display the track controller that has view of the yard
2. Follow the train as it moves through the system until it reaches its destination, swap controllers as necessary

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The train makes it to the correct destination track block with no issues | Pass |
| Each switch in the trains path is updated to the correct position as the train moves along the track | Pass |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/12/16 | Jeff Deely | Pass |

* + - 1. Additional notes or encountered bugs



* 1. **Test: Change Displayed Track Controller**
     1. Purpose
* Ensure that the track controller gui will change which section of track is displayed
  + 1. Procedure

1. Press the next track controller button (“>”)

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The track controller drop down menu shows the id of the next track controller (the last track controller has the first controller as its next) | Pass |
| The UI displays the track blocks under the jurisdiction of the current track controller | Pass |

1. Select a random track controller from the drop down menu

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The UI displays the track blocks under the jurisdiction of the current track controller | Pass |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/12/16 | Jeff Deely | Pass |

* + - 1. Additional notes or encountered bugs



* 1. **Test: Detect Track Failure, Change Track Model**
     1. Purpose
* Ensure that the track controller can detect a failed block in the track model
  + 1. Procedure

1. Using the Track controller tester place a failure on any block

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Track model will update with the new information | Pass |

1. Update all track controllers
2. Select the track controller that has jurisdiction over the failed block

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The Track Controller GUI will show that the authority of track blocks next to the failed block are 0 miles | Pass |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/14/16 | Jeff Deely | Pass |

* + - 1. Additional notes or encountered bugs



* 1. **Test: PLC’s Boolean Logic**
     1. Purpose
* Ensure that the PLC will provide correct updates to various train locations and track states
* Ensure that the UI corresponds with PLC changes

# Procedure

1. Using the Track Controller UI upload the PLC file so that it is set to the correct track controller
2. Using the CTC’s UI instantiate a new train from the yard, set its destination to move through the section of the track controller being tested
3. As it moves through the section watch the Track Controller UI for changes

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Every case specified in the PLC code document is carried out as the train moves through the track | Pass |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/13/16 | Jeff Deely | Pass |

* + - 1. Additional notes or encountered bugs



# Track Model

The following test procedures test proper functionality of the train controller

## Test: Load track and inspect blocks

* + 1. Purpose
* Ensure that the track data is properly loaded
* Ensure the UI track corresponds to the proper track data
  + 1. Procedure

1. Click the plus in the bottom right corner

2. Select Track Data (.csv) from file picker

3. Click on each block to ensure it matches the row in the (.csv)

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Selected Block shows expected block identity | Selected Block returns proper Block |
| Block parameters match load file for each block identity | Blocks match data from excel |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/14/2016 | Xavier Torgerson | Pass |

Additional notes or encountered bugs:



## Test: Track Model modification persistence

* + 1. Purpose
* Ensure changes made in other modules are reflected in the UI
* Ensure changes in the UI are reflected in other modules
  + 1. Procedure

1. Modify the Track Model by invoking a block failure from the Wayside controller

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Red indicator should show block failure in the Track model | Block failure is shown in Track Model however no longer through indicator |

2. Fix the block by clicking the repair button

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Red UI indicator should turn green | Block can be repaired but a green indicator is no longer shown |
| Wayside control should no longer see the block as failed | Pass |

## Test: Train presence detection

* + 1. Purpose
* Ensure the Track Model can properly locate the position of the trains on the track
* Ensure the new position information is relayed to the Track Model in a timely fashion
  + 1. Procedure

1. Send train to random block
2. Record every block where presence is detected until the train reaches it’s destination

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Continuous path of blocks from the origin to the destination | Works independently of CTC office |

3. Repeat at step 1 until train has traversed every block

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Every block number should be recorded through repeating step 2 | Every route into and out of each switch was taken successfully |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/14/2016 | Xavier Torgerson | Pass |

Additional notes or encountered bugs:



## Test: Send Speed and Authority to Trains

* + 1. Purpose
* Ensure Trains are getting the appropriate speed and authority
* Ensure the speed and authority are updated in a timely fashion
  + 1. Procedure

1. Use CTC office to make a new train
2. Set the speed and authority
3. Monitor the speed and authority propagate from the Wayside Controller to the Block by watching the block speed and authority near train update in the Track Model

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Block speed and authority near the train is updated when the CTC office changes speed and authority | CTC sends suggestion to wayside which is reflected in the Track Model |
| Block speed and authority are less than or equal to what the CTC office set for the train | The suggestion is reflected on each Block |

1. Monitor the speed and authority pass from the Track Model to the Train Model

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model speed and authority reflect the current block’s speed and authority | The train is receiving the appropriate speed/authority of it’s block |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/14/2016 | Xavier Torgerson | Pass |

Additional notes or encountered bugs:



# Train Model

The following test procedures test proper functionality of the train model

## Test: Test Mode

* + 1. Purpose
* Ensure that the test mode can update values of the train
  + 1. Procedure

1. Initialize a train model
2. Activate the test mode
3. Set the power request to zero
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model does not increase the speed of the train | Train Model maintains speed of zero |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## Test: Basic Velocity Calculations

* + 1. Purpose
* Ensure that when a positive power request is sent into the train model, the train accelerates
* Ensure that the train will not accelerate past the max velocity
* Ensure that the train will not accelerate past the max acceleration
* Ensure that both the services brake and the emergency brake both decrease the speed of the train
  + 1. Procedure

1. Initialize a train model
2. Activate Test Mode
3. Set power request to a large positive value (9999999)
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model increases the speed of the current train by 1.12mph from 0 to 1.12mph | Train speed is now 1.1 (rounded down from 1.12) |

1. Continue submitting the power until the velocity is increased to at least 43.50mph
2. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model does not have a velocity greater than 43.50mph | Speed does not go greater than 43.5 mph |

1. Activate the service brake
2. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model decreases speed of the train by at least 2.68 mph | Speed decreases more than 2.68 because of friction |

1. Activate the emergency brake
2. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model decreases the speed of the train more than the service brake, by at least 6.1mph | Speed decreases about 4 mph more than service break |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## Test: Complex Velocity and Velocity Edge cases

* + 1. Purpose
* Ensure that the train decreases in speed due to gravity
* Ensure that a negative grade increase velocity and a positive grade decreases velocity
* Ensure that a negative power request is set to be a zero power request
  + 1. Procedure

1. Initialize a train model
2. Activate Test Mode
3. Set power request to a large negative value (-999999)
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model’s velocity remains at zero and does not become negative | The velocity maintains at zero |

1. Set power request to a large positive value(9999999)
2. Repeatedly increase the power until it is at max velocity
3. Set the power request to zero
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model’s velocity decreases from the max velocity | Train speed decreases due to friction |

1. Set the grade to a negative number
2. Set the power request to zero
3. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model’s velocity decreases less than it decrease from gravity | With a negative grade, velocity decreases less |

1. Set the grade to a positive number
2. Set the power request to zero
3. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model’s velocity decreases more than it decreases from gravity | With a negative grade, velocity decreases more |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## Test: Update Speed Authority Request

* + 1. Purpose
* Ensure that the train can receive a speed and authority request
* Ensure that the train will update the requested speed and authority
  + 1. Procedure

1. Initialize the system
2. Create a train with an initial speed request of 30mph and authority request of 2 miles
3. Create a new speed and authority request with a speed of 10mph and authority of 1 mile
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The Train Model’s authority is 1 mile and not 2 miles | Authority is correct |
| The Train Model’s requested speed is 10mph | Requested speed is correct |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## Test: Brake Failure Error

* + 1. Purpose
* Ensure that the Brake Failure error case works
  + 1. Procedure

1. Initialize a train model
2. Activate Test Mode
3. Activate Train Engine Failure
4. Set power request to a large positive value (9999999)
5. Submit the power request multiple times, until the current velocity is at the maximum velocity
6. Activate the service brake record and the decrease in speed
7. Activate the emergency brake and record the decrease in speed
8. Activate the brake failure and the service brake and record the decrease in speed
9. Activate the emergency brake and brake failure and record the decrease in speed
10. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The Train Model decreases in speed less when the brake error is on with the service brake than with the brake error being off with the service brake | Train model does not include service brake deceleration when brake failure is active |
| The Train Model decreases in speed less when the brake error is on with the emergency brake than with the brake error being off with the emergency brake | Train model does not include emergency brake deceleration when brake failure is active |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## 

## Test: Signal Pickup Failure Error

* + 1. Purpose
* Ensure that the Signal Pickup Failure error case works
  + 1. Procedure

1. Initialize the system
2. Activate Signal Pickup Failure
3. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| The Train Controller receives a signal pickup error message | Train controller registers singal pickup failure |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## Test: Train Engine Failure Error

* + 1. Purpose
* Ensure that the Train Engine Failure error case works
  + 1. Procedure

1. Initialize a train model
2. Activate Test Mode
3. Activate Train Engine Failure
4. Set power request to a large positive value (9999999)
5. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model does not increase the speed of the train | Train model decreases because no power and friction |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



## Test: Receive Train Data

* + 1. Purpose
* Ensure that the Train Model can update using data passed from the Train Controller and Track Model
  + 1. Procedure

1. Initialize the system
2. Create a train and have it go to a block with a non-zero grade
3. Have the Train Controller alter the states of the lights and doors
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Model interface does not have a zero grade shown | Train Model has the correct grade |
| Train Model interface has the altered states of the lights and doors shown | Train model interface shows updated values |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/10 | Spencer Worms | Pass |

* + - 1. Additional notes or encountered bugs:



# Train Controller

The following test procedures test proper functionality of the train controller

## Test: Emergency Passenger Brake

* + 1. Purpose
* Ensure that the state of the emergency passenger brake can be toggled via the train controller
* Ensure that change of state is updated in the train model
* Ensure that the emergency passenger brings the train model to zero speed
  + 1. Procedure

1. Instantiate a train controller / train model pair

2. Press and hold the emergency passenger brake button on the train controller

3. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Controller shows “Emergency Passenger Brake = ON” | Emergency Passenger Brake = ON |
| Train Model shows “Emergency Passenger Brake = ON” | Emergency Passenger Brake = ON |
| Train Model shows “Train speed = 0 mph” | Train speed = 0 mph |

4. Release the the emergency passenger brake button

5. Use the train controller to request power

6. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Controller shows “Emergency Passenger Brake = OFF” | Emergency Passenger Brake = OFF |
| Train Model shows “Emergency Passenger Brake = OFF” | Emergency Passenger Brake = OFF |
| Train Model shows “Train speed > 0 mph” | Train speed > 0 mph |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs



## Test: Train Speed Request

* + 1. Purpose
* Ensure that the train controller can request a setpoint speed
* Ensure that the train controller translates this speed to a power request
* Ensure that the power request is transferred to the train model
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Use the speed slider on the train controller to request 20 mph
3. Allow the train time to come to speed
4. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Controller shows “Speed Request = 20mph” | Speed Request = 20mph |
| Train Controller shows “Power Request > 0 mw” | Power Request > 0 mw |
| Train Model shows “Current Speed = 20 mph” | Current Speed = 20 mph |
| Train Model shows “Current Power > 0 mw” | Current Power > 0 mw |

1. Use the speed slider on the train controller to request zero speed
2. Allow the train time to come to a full stop
3. Verify:

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train Controller shows “Speed Request = 0 mph” | Speed Request = 0 mph |
| Train Controller shows “Power Request = 0 mw” | Power Request = 0 mw |
| Train Model shows “Current Speed = 0 mph” | Current Speed = 0 mph |
| Train Model shows “Current Power = 0 mw” | Current Power = 0 mw |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Train Doors

* + 1. Purpose
* Ensure the state of the right and left train doors can be toggled via the train controller, between “open”, “closed”, and “fail”
* Ensure that the change of train door state is updated in the train model
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Use the train controller to set the left and right train doors to “open”
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Right Doors = Open” | Right Doors = Open |
| Train controller shows “Left Doors = Open” | Left Doors = Open |
| Train model shows “Right Doors = Open” | Right Doors = Open |
| Train model shows “Left Doors = Open” | Left Doors = Open |

1. Use the train controller to set the left and right train doors to “closed”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Right Doors = closed” | Right Doors = closed |
| Train controller shows “Left Doors = closed” | Left Doors = closed |
| Train model shows “Right Doors = closed” | Right Doors = closed |
| Train model shows “Left Doors = closed” | Left Doors = closed |

1. Use the train controller to set the left and right train doors to “fail”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Right Doors = fail” | Right Doors = fail |
| Train controller shows “Left Doors = fail” | Left Doors = fail |
| Train model shows “Right Doors = fail” | Right Doors = fail |
| Train model shows “Left Doors = fail” | Left Doors = fail |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Train Lights

* + 1. Purpose
* Ensure that the train controller is capable of change the state of the train lights
* Ensure that the change of train light state is reflected in the train model
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Use the train controller to set the train lights to “on”
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train Lights = on” | Train Lights = on |
| Train model shows “Train Lights = on” | Train Lights = on |

1. use the train controller to set the train light to “off”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train Lights = off” | Train Lights = off |
| Train model shows “Train Lights = off” | Train Lights = off |

1. use the train controller to set the train light to “fail”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train Lights = fail” | Train Lights = fail |
| Train model shows “Train Lights = fail” | Train Lights = fail |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Train AC

* + 1. Purpose
* Ensure that the train controller is capable of changing the state of the train AC
* Ensure that the change of state is reflected in the train model
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Use the train controller to set the train ac to “on”
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train AC = On” | Train AC = On |
| Train model shows “Train AC = On” | Train AC = On |

1. Use the train controller to set the train ac to “off”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train AC = off” | Train AC = off |
| Train model shows “Train AC = off” | Train AC = off |

1. Use the train controller to set the train ac to “fail”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train AC = fail” | Train AC = fail |
| Train model shows “Train AC = fail” | Train AC = fail |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Train Heat

* + 1. Purpose
* Ensure that the train controller is capable of changing the state of the train heat
* Ensure that the change of state is reflected in the train model  
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Use the train controller to set the train heat to “on”
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train Heat = on” | Train Heat = on |
| Train model shows “Train Heat = on” | Train Heat = on |

1. Use the train controller to set the train heat to “off”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train Heat = off” | Train Heat = off |
| Train model shows “Train Heat = off” | Train Heat = off |

1. Use the train controller to set the train heat to “fail”
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Train AC = fail” | Train AC = fail |
| Train model shows “Train AC = fail” | Train AC = fail |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Advertisement

* + 1. Purpose
* Ensure that the train controller is capable of making an advertisement
* Ensure that the change of state is reflected in the train model
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Use the train controller to enable advertisement
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Advertisement = on” | Advertisement = on |
| Train model shows “Advertisement = on” | Advertisement = on |

1. Use the train controller to disable advertisement
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Advertisement = off” | Advertisement = off |
| Train model shows “Advertisement = off” | Advertisement = off |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Emergency Brake

* + 1. Purpose
* Ensure that the train controller is capable of triggering an emergency brake
* Ensure that the change of state is reflected in the train model
* Ensure that the emergency brake brings the train to zero speed
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Request a train speed of 20 mph and allow time for the train to come to speed
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Speed Request = 20 mph” | Speed Request = 20 mph |
| Train controller shows “Current Speed = 20 mph” | Current Speed = 20 mph |
| Train model shows “Current Speed = 20 mph” | Current Speed = 20 mph |

1. Use the train controller to enable the emergency brake. Allow the train time to come to a full stop.
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Emergency Brake = on” | Emergency Brake = on |
| Train controller shows “Current Speed = 0 mph” | Current Speed = 0 mph |
| Train model shows “Emergency Brake = on” | Emergency Brake = on |
| Train model shows “Current Speed = 0 mph” | Current Speed = 0 mph |

1. use the train controller is disable the emergency brake
2. Request a train speed of 20 mph. Allow the train time to come to speed.
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Emergency Brake = off” | Emergency Brake = off |
| Train controller shows “Current Speed = 20 mph” | Current Speed = 20 mph |
| Train model shows “Emergency Brake = off” | Emergency Brake = off |
| Train model shows “Current Speed = 20 mph” | Current Speed = 20 mph |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:



## Test: Service Brake

* + 1. Purpose
* Ensure that the train controller is capable of changing the state of the service brake
* Ensure that the change of state is updated in the train model
* Ensure that the service brake decreased train speed
  + 1. Procedure

1. Instantiate a train controller / train model pair
2. Request a train speed of 20 mph and allow time for the train to come to speed
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Speed Request = 20 mph” | Speed Request = 20 mph |
| Train controller shows “Current Speed = 20 mph” | Current Speed = 20 mph |
| Train model shows “Current Speed = 20 mph” | Current Speed = 20 mph |

1. Use the train controller to enable the service brake
2. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Service Brake = on” | Service Brake = on |
| Train controller shows “Current Speed < 20 mph” | Current Speed < 20 mph |
| Train model shows “Service Brake = on” | Service Brake = on |
| Train model shows “Current Speed < 20 mph” | Current Speed < 20 mph |

1. Use the train controller to disable the service brake
2. Allow time for the train to reach its setpoint speed
3. Verify

|  |  |
| --- | --- |
| Expected Results | Actual Results |
| Train controller shows “Service Brake = off” | Service Brake = off |
| Train controller shows “Current Speed = 20 mph” | Current Speed = 20 mph |
| Train model shows “Service Brake = off” | Service Brake = off |
| Train model shows “Current Speed = 20 mph” | Current Speed = 20 mph |

* + 1. Results

|  |  |  |
| --- | --- | --- |
| Date Completed: | Name of Tester: | Test Result (Pass/Fail): |
| 12/11/16 | Garrett Grube | Pass |

Additional notes or encountered bugs:

# 