IoT HTR: User Guide

This document will provide the user with a detailed explanation of how to run the software with the test file.

1. Booting up the software
    a. Create a directory in your system named “LCS Project”
    b. In the LCS Project directory, insert the IoTEngine.py file and the sensordata.csv file
    c. Open IoTEngine.py in your favorite editor to view the code and documentation comments. This will help you understand the functionality of the code
    d. Open the terminal and cd into the directory “LCS Project”
    e. Run the command “python3 IoTEngine.py” in the terminal
        i. If this does not work, run “python IoTEngine.py”
    f. You will be prompted for a username; enter “conductor”
    g. You will be prompted for a password; enter “password”
    h. The GUI display will begin

2. Editing the test file
    a. If you would like to change the test data, open the “sensordata.csv” file. You will see a csv file with data simulating what is received by the sensor data
        i. The data file has rows representing the Sensor Heading and column representing the data from the sensor
        ii. The IoT Engine will receive one column of data every 5 seconds
        iii. When the IoT Engine has parsed through the entire file, the engine assumes that the ride has ended
    b. Manipulating Object Distance
        i. Tracks how far the nearest object is to the train
        ii. To view the object error message displayed by the IoT Engine, set the Object Distance to be less than or equal to 1000 meters. For no error message set the Object Distance to be greater than 1000 meters.
    c. Manipulating Wheel Speed
        i. Calculates the wheel speed in rotations per minute
        ii. To view the slippage error message displayed by the IoT Engine, set the GPS Speed to be 20 mph and the Wheel Speed to be 100 rpm (9.8125 mph). For no error message set the GPS Speed to be 5 mph and the

Wheel Speed to be 100 rpm (9.8125 mph). Other combinations where the wheel speed is greater or less than the expected speed can be used as well, this is just one example.

d. Manipulating Moving
    i. Checks if the nearest object is moving
    ii. To view the object error message displayed by the IoT Engine about a moving object, set Moving to be TRUE and the Object Distance to be less than 1000 meters. To view the error message displayed by the IoT Engine about a stationary object set Moving to be FALSE and the Object Distance to be less than 1000 meters. Other combinations can be used as well, this is just one example.
e. Manipulating Humidity
    i. Checks the humidity level outside to check for slipping
    ii. To view the slippage error message displayed by the IoT Engine, set the Humidity to be greater than 50. For no error message set the Humidity to be less than or equal to 50.
f. Manipulating Precipitation Level
    i. Checks the precipitation level outside to check for slipping
    ii. To view the slippage error message displayed by the IoT Engine for snow, set the Precipitation Level to be greater than 4 and the Temperature to be less than 32. To view the error message displayed by the IoT Engine for rain, set the Precipitation Level to be less than or equal to 4 and the Temperature to be greater than or equal to 32.
g. Manipulating Gate
    i. Checks if there is a gate present
    ii. To view the gate error message displayed by the IoT Engine, set the Gate to be TRUE and Object Distance to be between 100 and 1500 meters. For no error message, set the Gate to be FALSE and Object Distance to be greater than or equal to 1000 meters.
    iii. To view the initial horn error message displayed by the IoT Engine, set the Gate to be TRUE and Object Distance to be between 1500 and 1700 meters. To view the horn error message displayed by the IoT Engine when the train has arrived at the gate, set the Gate to be TRUE and Object Distance to be between less than 100 meters.
h. Manipulating Connectivity
    i. Check that there is connectivity to the internet connection

ii. To establish a connection to WiFi/Cellular, set Connectivity to be 1, and the program will exit and reroute the conductor to the normal train system. To continue on the IoT Engine, set the Connectivity to be 0.