CS Capstone Problem Statement

November 28, 2018

HyperRail App

Prepared for

OPENS Lab

Chet Udell

Prepared by

Group 25

Vincent Nguyen
Adam Ruark
Yihong Liu

Abstract

Manually collecting environmental data in a large space is time consuming and tedious. Through the HyperRail application, one can remotely configure an environmental sensor package to move along a railway in a space, collect data, and send it back to the user wirelessly, thus automating the data acquisition process.

CONTENTS

1 Introduction

Climate change is causing many adverse effects on the environment across the globe. As a result, monitoring the environment has become more important in order to detect and thoroughly understand changes in the environment, and to be able to adapt or take action. However, frequently traversing through rugged terrain or dangerous environments, such as steep mountains, dense forests, and deep caverns, can often be difficult or unfeasible. On the safer side, monitoring agricultural areas, such as greenhouses and vineyards, often becomes monotonous and very time consuming to do manually. Although the use of drones can help oversee certain areas, their cost can make them inefficient or the terrain itself can prevent their use.

Using a small railway, an automated environmental sensor package, which contains a variety of different sensors, can traverse through a space, collecting information as it travels along the railway. The sensor package's settings can be customized through a custom-developed application, called the HyperRail application. The application currently allows the user to specify the speed of the package, current length of the rail, and the size of the spool used for the motors. It also monitors the position of the sensor package by calculating the number of motor steps it has taken and updates the position display in real-time. However, it currently requires a direct connection to the sensor package itself, making the HyperRail's use limited to small and contained environments to avoid external interference.

The goal of the project is making the rail system accessible to anyone with internet access. Instead of requiring a direct connection to the micro-controller to control the rail system, the user should be able to use their personal computer or mobile phone to interact with the rail system, making it easier for user to operate.

2 Proposed Solution

2.1 Description

Our proposed solution is to create a web application that communicates with a central server, which in turn communicates with the sensor system. This web application will be mobile friendly to work on Android and iOS in addition to personal computers. The web application will let the user configure schedules for the sensors to run, set up intervals for the sensors to collect data, allow the user to manually move the sensors along the rail as needed, and also display the current status of the sensor system. Once a user saves a configuration, these settings will be uploaded to a central server where it will be deployed to the corresponding sensor system. Ideally, all communication between the HyperRail sensor system and the user interface would be through the central server. However, there may be a setup where we want to connect directly from the web application to the sensor system directly; this is still up for debate. This configuration allows the user to access the device from anywhere in the world, and the central server abstracts the two endpoints from each other. Furthermore, this ensures a unified experience.

Scheduling runs will be controlled and initiated by the central server by sending signals to the HyperRail sensor system with locations to go to, rate of travel, and any points of interest to monitor along the rail. The data collected by the sensors will be uploaded to the central server and can be viewed by the web application.

In addition to building the web application, we will also be optimizing the current applications response time and precision. For the project, we will be primarily working with the Arduino IDE for hardware interactions, and using HTML, C, or Python for the web application. Features that need further development or optimization include a more detailed, yet intuitive interface and data precision.

2.2 Stretch Goals

One stretch goal is to allow the sensor package and HyperRail App to interact with existing actuators in the space. Because actuators are machines that perform simple movements on objects, such as valves or switches, this interaction will give the application additional capabilities depending on the application of the actuators.

Another stretch goal is the capability to configure a fleet of sensor systems to use the same configuration. This allows for quick configuration of a series of sensors that are intended for the same purpose. Another feature that we would like to include is the capability to save and load configurations to allows users to share settings across devices. This would make the application even more portable for users and allow the sensor system to be truly configured from anywhere.

3 Performance Metrics

This project will be considered complete once we develop a functional HyperRail application. Primarily, there must be a web application that is usable on Android, iOS, and personal computers that can connect to the HyperRail system and upload configurations to it from anywhere. Additionally, this application should be able to automate runs on a scheduling system and set up events for sensors to scan. The application should also display the current status of the HyperRail system, including position along the rail and velocity, and data collected from it. The source code for this project must also be well documented to allow other teams to take over and enhance this project in future.

Some additional features that we potentially want to add, but are not mandatory for a completed project include configuring a multitude of HyperRail systems at once, improving application performance to ensure an enjoyable user experience, the ability to allow the user to collect more types of data, and the option to save configurations for future use.