Creating a Web Application to Monitor Battery Status

Tiffany Valitis & Raychel Beasley

Abstract

In this paper we explain how we chose a web application to display battery diagnostics as a viable project, and explain in depth why an application was needed for this function. We will also introduce the use of javascript, html and css as tools to display an existing JSON in the Building Wide Intelligence (BWI) project in a more aesthetic manner. Lastly, we will propose updates and future modifications to extend the functionality of the application.

Introduction

In our project we created a web application to display the current state of each segbot, without the need to be physically near them. The state information provided in the web application includes current battery level, approximate battery life left, the floor the segbot is located on, and the current or previous user. It also displays the order they need to be charged in, based on how much battery each has remaining.There was an existing system in place that sent emails when a robot was low on battery, but the emails were only sent to a small list of people, and relied on a robot's battery profile. However, there was only one robot, Marvin, with an accurate battery profile, and the information was not readily accessible to everyone in the lab.We pulled information from the existing system, a mentor's project, and created battery profiles for the other robots. Then, we compiled it into an easy-to-use web application everyone in the lab can access.

Background & Related Work

A lot of our project uses existing programs and systems already implemented on the robots. We used Max Svetlik's programs to create battery profiles for the robots, in order to determine how much battery life was remaining. This information was then published to the same JSON that Walter Sagehorn uses for his Virtour website, from which we pull the information for our own website. Though no outside work exactly replicates this project, other researchers have had similar goals , such as Osentoski, et al. Their paper "Robots as web services: Reproducible experimentation and application development using rosjs," assisted in this project regarding the structure and capabilities of rosjs and the use of web applications in conjunction with ROS. Many sections were directly relevant to understanding the web application. Another work, by D. Schulz, et al, " Web interfaces for mobile robotics in public places," described using the internet, as "unpredictable" making the point that when you make something public, such as the web application described throughout, you cannot know your audience. This idea is why this project includes a graphical user interface to display the data in an understandable way. "Robot web tools: Efficient messaging for cloud robotics," authored by R. Toris, et al, discusses the use of JavaScript, and HTML5 in creating interactive websites, and discusses the growth in the past decade of using these tools for web interfaces. As the article stated, this has become the "new standard" which is why this project, which displays battery diagnostics for the building wide intelligence (BWI) project, is relevant for the use of the segbots and the growth of the building wide intelligence (BWI) project in UTCS. Another outside publication used was "Robot web tools [ros topics]," Alexander, Brandon, et al. In the article, the authors describe the pros and cons of ROS, and details the usefulness of rosbridge and rosjs in enabling non-ROS client processes (such as web interfaces) to interoperate with ROS processes. The last relevant publication used in researching this web application was,"Rosbridge: Ros for non-ros users,"Crick, Christopher, et al. In the article Rosbridge: ROS for Non-ROS Users, the authors describe rosbridge and rosjs, a middleware abstraction layer over ROS also researched in the previous article, and the accompanying JavaScript library they have created to allow people unfamiliar with not only ROS, but most of robotics, to use robots and have a gentle introduction into the field.

Technical Approach

Before beginning work on this application we needed to figure out what had previously been done by other researchers in the segbot lab. We discussed our project and proposed approach with several past CS research students. These students assisted us in figuring out what has previously been accomplished. One post doctoral, Justin Hart, suggested several key elements from his observations on what the Building Wide Intelligence (BWI) might need as it expanded over time. This led to the development of our project. Previous mentors and researchers had made ROS nodes to collect battery data and form battery profiles, as well as ways to read different information from the web. We utilized existing methods and extended the functionality in an easy to read application.

In order to make our web application, we needed to learn the languages HTML, CSS, and JavaScript. Using a variety of online tools, we learned basic HTML5 and CSS, which we used to format the web page. We used JavaScript to parse and process the data from a JSON, created by Walter Sagehorn for the Virtour application. His JSON (Figure 1) already contained multiple elements we needed. We added segments of code his existing program structure to include the remaining battery life data. In addition, we used JQuery to create buttons and add some finishing touches to our webpage.

[{"Name":"leela","User":"legolas","IP":"10.147.131.148","X":"0.03
{"Name":"bender","User":"legolas","IP":"10.147.249.175","X":"-40.

Fig. 1 - This is a segment of the JSON nixons-head by Walter Sagehorn.

For the ROS elements of our project, we relied on existing programs to collect and process the data we needed, however, the process of modifying and using pre-existing code resulted in unexpected complications. First, to create battery profiles, we had to connect with Max Svetlik, who has moved on to other research endeavours outside of the Building Wide Intelligence (BWI) lab. Max had previously created a node to collect battery data, and a program to process the data and generate a function, which could then be used to calculate how much time a battery had left. However, due to an error, which we narrowed down to either Max's code or a problem with the Arduino hardware on the robot, accurate battery voltage wasn't being published. The data we collected from running a V2 segbot (Bender) for approximately 10 hours was 12.5V. When we attempted to provide a best fit curve estimate to the graph the result was a straight which was useless for any further calculations (Figure 2). Leela's data was corrupted, so we created a curve from old data from Marvin, provided to use by Max. We had to convert the function from Python to C++, and code it into the node that published the JSON, in order to be able to receive the battery data.

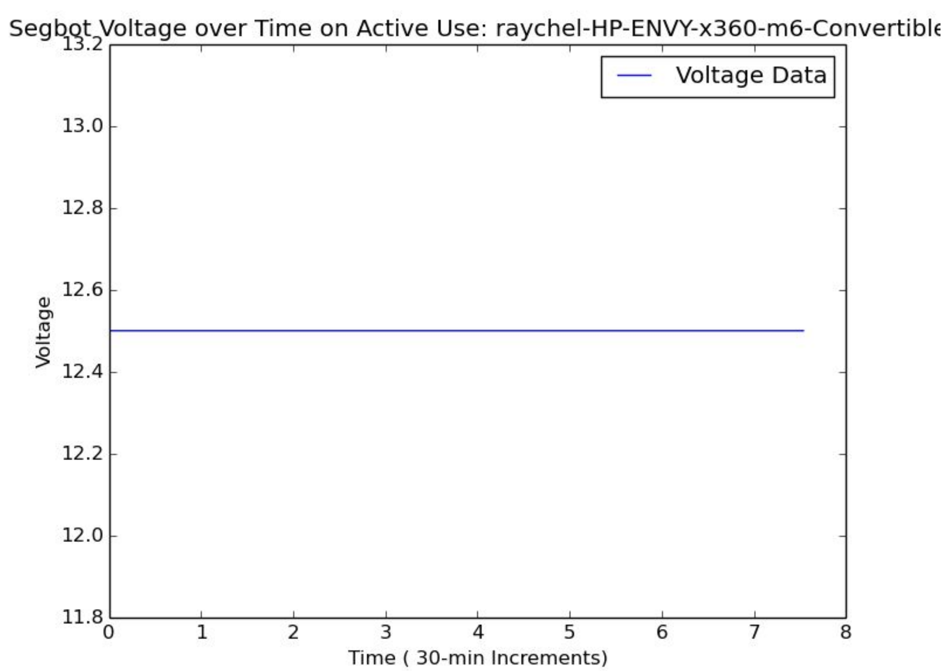

Fig.2 -This graph shows the segbot Bender's day 2 bag data graphed. It remained 12.5 volts for 10 hours.

Another issue we encountered was where to host the images we used. We chose to use GitHub, as it provides a usable link for javascript. Initially we also used GitHub to host the web application itself, however, as of June 2016, non-custom domains are secured by redirecting http requests to https. This proved troublesome, as the JSON, nixons-head, is http. To get around this the application is now, temporarily, hosted through Raychel Beasley's UTCS lab account, which provides the domain http://www.cs.utexas.edu/~ray0237/. In the future the BWI project will have a more permanent URL provided for the application.

Evaluation & Examples

We decided to approach the evaluation side of this project in three phases. The first phase consisted of deciding where each element data for the proposed web application was located. To do this we had to parse the pre-existing JSON file for the data it already provided to us. Phase two was where we did the actual collecting of the needed data. To do this we not only had to change the JSON to C++, but we also had to store battery data and later create a profile from it. This phase consisted of a great deal of debugging, as the profiler demonstrated flaws in the collected data files. To evaluate the data we attempted to create a graph (Figure 3) with the equation -A * exp(Kx) + C, which would create a curve of best fit for the declining voltage from the segbots. This evaluation led to the errors in the data described above. The third phase was displaying the final data collected. Using conditionals, we changed the images on the application based on said collected data. The conditionals themselves could be evaluated through other students projects. To save time, when other students ran their projects on the segbots, we were able to view whether or not our data changed. We also used test code, by hard coding values, to test the application apart from parsing the JSON data. As described in the technical approach, parts of this project had to be adapted as problems arose, because of this, as can be expected, our proposed and final project differ in areas. The proposed goal and the final goal for this project remained the same, to create a web application that included the current states of the segbots. However; as the project progressed the methodology originally proposed was altered from a python programmed GUI to Javascript, HTML5 and CSS. This was largely because, originally, we did not realize what tool was best for parsing information from a JSON. The layout of the web application also changed some, but that can be accredited to how the program itself naturally developed (Figure 4).

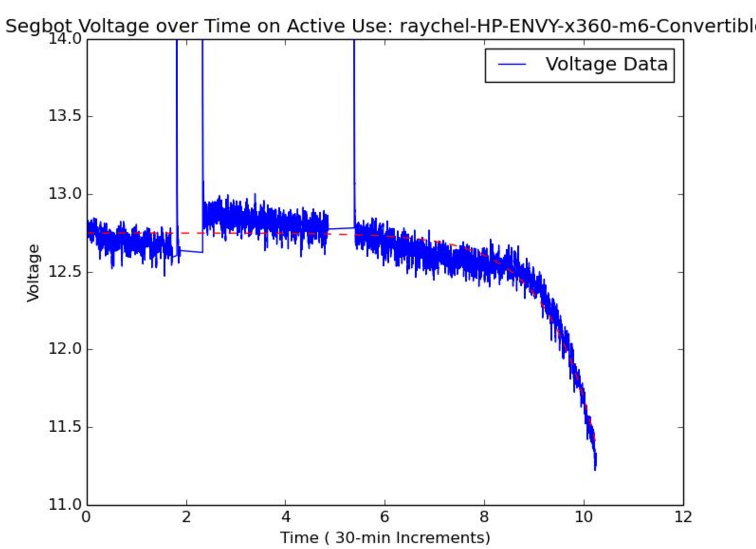

Fig.3 - Graph of the used voltage data (Marvin's)

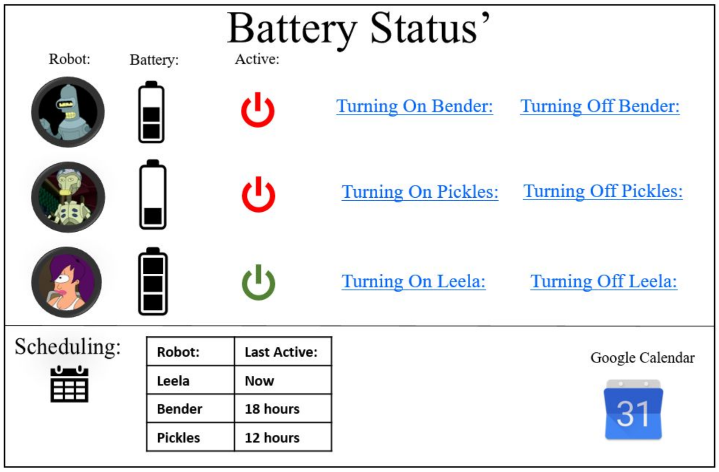

Fig.4 - Original mock up of web application.

Conclusion & Future Work

See Figure 5. Our initial goal was to create a web application to display the states of the segbots utilized in the Building Wide Intelligence (BWI) project within the University of Texas Computer Science department. We succeeded in this goal. Our final result includes the name of two different segbots, the third usable segbot, Pickles, was left off because it is next in line to be decommissioned, estimated battery percentage (via fractions of a battery image), last or current user, current floor, estimated time remaining and order to be charged. Though unforeseen issues led to parts of the battery profile being less accurate, specifically for the v3 Leela, there is now an existing application that can be expanded upon in later semesters, both aesthetically and with more data. These expansions include functionality such as the use of Google Calendar to extend the charging order of the segbots to formal scheduling. This could include alerts as to what segbot should be on at a certain time, and what events, such as Explore UT are coming up that would change the normal run time of the robots. Expansion would also include more accurate battery profiles, though Bender and Marvin's (the profile currently being used) should not differ greatly, Leela being a newer model, specifically, will need to be changed and re-implemented. To allow time flexibility when collecting a new profile (remember that Bender took 10 hours to collect) a merging option is needed. This way it is similar to update the profiles on a more regular basis as the batteries age. Lastly, security could be added to allow only members of the Building Wide Intelligence (BWI) project to view the application, specifically if a scheduling option is made available in order to keep outside individuals from purposefully, or accidentally, making changes to the set schedule.

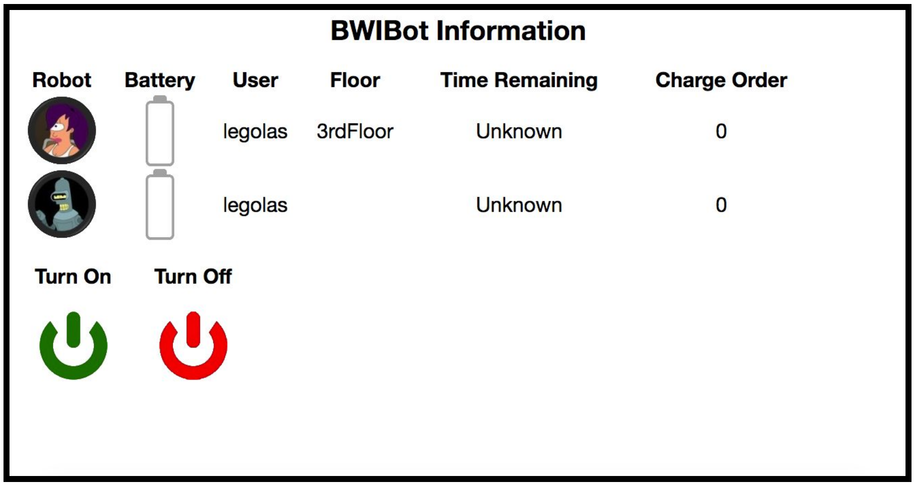

Fig.5- Final Web application

Citations

Alexander, Brandon, et al. "Robot web tools [ros topics]." *IEEE Robotics & Automation Magazine* 19.4 (2012):20-23.

http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6377438&tag=1

Crick, Christopher, et al. "Rosbridge: Ros for non-ros users." Proceedings of the 15th International Symposium on Robotics Research. 2011.

https://pdfs.semanticscholar.org/9e0f/01c9a3a2530aa7e7044a4f18aec5f80265df.pdf

D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers. Web interfaces for mobile robotics in public places. IEEE Robotics and Automation Magazine, 49
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=833575

R. Toris, J. Kammerl, D. V. Lu, J. Lee, O. C. Jenkins, S. Osentoski, M. Wills, and S. Chernova, "Robot web tools: Efficient messaging for cloud robotics," in Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on, pp. 4530–4537, Sept 2015.
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7354021

S. Osentoski, G. Jay, C. Crick, B. Pitzer, C. DuHadway, and O. C. Jenkins, "Robots as web services: Reproducible experimentation and application development using rosjs," in Robotics and Automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011, pp. 6078–6083
http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5980464