List of Deliverables for Dynamic Parking Signage Project

ELEC 491 Capstone Design Project

Team PL-89

Geoff Goodwin-Wilson
Timothy Cheng
Hiu Lok Cheung
Yuyang He
Jared West

TABLE OF CONTENTS

Glossary

COVID-19	Coronavirus
CPEN	Computer Engineering
ELEC	Electrical Engineering
IDE	*Integrated Development Environment*
PCB	*Printed Circuit Board*
SEEDS	*Social Ecological Economic Development Studies*
UBC	*University of British Columbia*
UBC PAS	*UBC Parking and Accessing Services*
USB	*Universal Serial Bus*

1. Summary

To complete the capstone project 89: UBC Parking Dynamic Parking Signage, we will deliver the following list to the UBC Parking and Access Services by April 24th of 2020:

Mandatory Course Documents

- Requirement Specification documentation
- Design documentation
- Validation documentation
- Project video

Project Specific Deliverables

Hardware:

- 3 Dynamic Parking Sign prototypes
  - 1 without display
  - 2 with display
- PCB Design files
- Solidworks design files
- Bill of materials

Software:

- Source code of:
  - Embedded system for prototype to display an image
  - Web application and server to communicate to the cellular development board
  - Cellular dev board that works with a data plan and SIM card provide by UBC PAS
  - Display dev board to display an image, commented to aid readability

Documentations:

- Developer manual
  - Signage initialization and troubleshooting

Files:

- Zipped up folder of source code (ours and legacy or previous year's i.e. 2018-2019 we are 2019-2020), additional documentation we made, resources we used, and previous capstone team's milestones.
    - Please see Appendix A in the Developer manual for file structure and content. In this appendix, we illustrate what each folder contains.
- Power Budget Spreadsheet
- Credentials
    - Accounts (AWS, ST, etc)
    - Permissions and keys to AWS IoT Core and IoT devices

2. Final Prototype

Due to the pandemic of COVID-19 in 2020, social distancing intervention was implemented and UBC canceled in-person classes. From March 1st to March 15th, students began to feel unsafe to come to school as the virus was spreading around the world but had not been assessed as a high risk to the Canadian population. On March 16th, UBC announced that in person classes are canceled and all classes are to be completed online. In our capstone course CPEN/ELEC 491, teams were no longer permitted to meet in person to complete their projects. Our team is hardware-focused and was severely impacted by the new policies in place. We lost nearly 4 weeks of in-person development time to troubleshoot and complete the prototype (hardware) and there were 2 weeks of uncertainty before the full online transition when students felt unsafe to come to school. After March 16th, all assignments were online and students (general population) were to remain home unless exercising or grocery shopping (essential needs) in a valiant effort to flatten the curve. Students needed time to acclimate as productivity was low from staying home all day and every day. Some students moved home amidst nations quickly closing their borders to slow down the movement of people and thereby the spread of COVID-19.

Amidst this turmoil, our team continued to work on our prototypes but were not able to meet our promise of delivering two working prototypes. We have since changed our approach to the project by documenting as much as possible to ease the handoff process to the next capstone team. We want to provide our client with the most value by helping the next team quickly pick up where we were forced to stop working in person.

Although we have promised 2 working prototypes initially, we were building 3 prototypes for redundancy in case a prototype fails or components fail. All hardware development was finished when the course announced the teams were not able to collaborate in-person.

3. System Deliverables

3.1 Credentials

Although we were against using our personal Github accounts to ease the handoff process to the next team, we were left without a choice to use our personal GitHub account to cut down unnecessary wait time. We hoped to use a Github account owned by SEEDs to manage the source code to lessen the complications of transitioning for next year's capstone team. We used our team Gmail account to sign up for various hardware vendors to access resources such as IDEs, datasheets, and step-by-step tutorials. Along with credentials to the team Gmail account, we have included the list of accounts we created to access such resources. This is located in files/credentials that we have also handed in.

Permission files (.pem) are required to ssh into servers to interact with the AWS server on the command line. We included this in credentials/keys. Each IoT device also has its own .pem files (private key).

3.2 Source code

Our source code is all stored in GitHub. The SEEDS coordinator, Jonathan Kew, was unable to secure us a PRO account for us. We required this arrangement because free GitHub accounts only allow up to 3 collaborators on a private repository and the team consists of 5 members. However, we were able to use the student account of one team member, Timothy Cheng, which allowed 5 collaborates on a private repository. UBC PAS mentioned they were planning on setting up a GitHub account for future development that can be easily passed on from team to team. Although we were not able to figure out the details before the course deadline as we were no longer allowed to meet in person (COVID-19), we are working to migrate the code to their account during the summer of 2020. After this, we will update this document.

3.2.1 Embedded System for Sign

The source code for the sign is written in C to display an image. It is in files/Github/dynamicparking-master/dynamicparking-master/Firmware/Try. So far the image is displayed imperfectly, with the top part of the screen unable to function as shown in Figure 1.

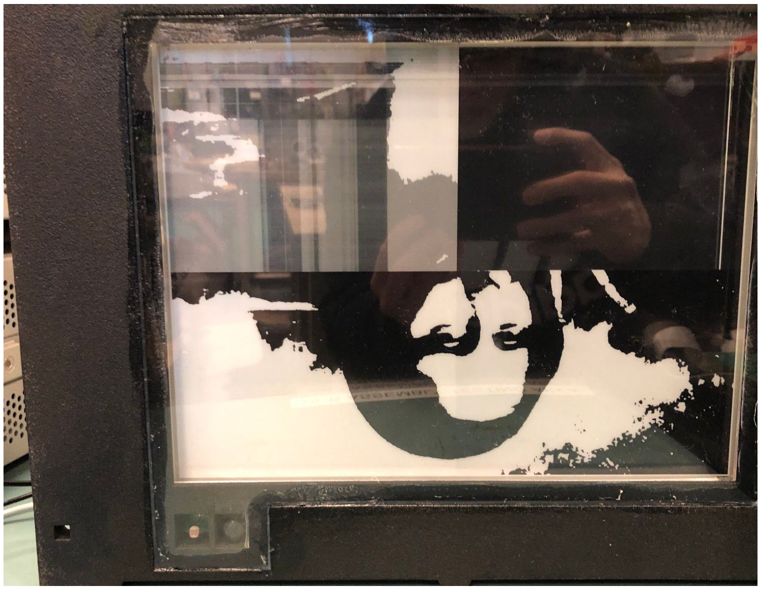

Figure 1. The Top Part of Prototype Screen Unable to Function

3.2.2 Management System

The AWS IoT Core is just a GUI on the AWS platform. It requires a developer to interact with buttons instead of code so this part requires no code, just credentials to the account.

The web app source code is in files/Github/capstonewebapp.

3.3 Developer Manual

We have included a developer manual in our deliverables to help next year's capstone team more easily transition into the development phase. Our predecessor did not leave any documentation behind so we decided to include detailed documentation on what tools we used (such as IDEs, setup, software dependencies, and helpful resources in the form of videos and websites) and how to use them.

4.Hardware Deliverables

4.1 Dynamic Parking Sign Prototypes

Our team will deliver our final progress on the dynamic parking sign prototype(s). During the course of this capstone course, we have fabricated 3 prototypes of the dynamic parking sign. All of the prototypes are identical in design. Note that the third prototype does not have an e-ink display.

4.2 PCB Design Files

The PCB was developed using Altium Designer, which is available to all UBC ECE students for free. All of the Altium design files are included as deliverables, so the design will be able to be continued by the next designer. A pdf file of the schematics will also be provided for convenience. These files will be located on our project GitHub repository called “dynamicparking” in the “Hardware” folder.

4.3 Solidworks Design Files

The enclosure and hardware assembly were designed using Solidworks, which is also available to all UBC students. The design files for every part of the hardware assembly will be included in our deliverables. These files are quite large in size, so they will be delivered on a USB stick with the prototypes.

4.4 Bill of Materials

A complete bill of materials will be included for our project, including part descriptions, where each part was bought, and part numbers for each part. This will assist the next designer in referencing parts that were brought to develop the hardware prototype.