ELECOMP CAPSTONE SUMMIT
FRIDAY, MAY 11, 2018

THE UNIVERSITY OF RHODE ISLAND

THINK BIG WE DO℠

Blank-Out Sign Connectivity Over Cellular Network

Creating a connection between blank-out signs and users via cellular module

Team Members: Justin McGrath (CPE), Alex Kenney (ELE)

Technical Director: David Connolly

Project Motivation

This project is all centered around Blank-Out Signs, which are signs that are used on the roads that offer different messages to motorists. This works by lighting up LEDs that are implemented in the signs in different sections such that they display a message. An example of this would be the LED signs show in toll booths that represent which lanes are open or closed. These messages are predefined so a sophisticated control apparatus is not necessary in order to change the message shown. SESA designed the BOS to operate with a simple dry-contact closure driven by a low-cost microcontroller. This alternative to full-featured DMS makes BOS lack the means for communication. This means that the only way to change the message on the sign or to check for an error. The expensive installation of hardwired communications infrastructure necessary to control and monitor signs negates the cost advantages of the BS products and renders remote use economically untenable.

Key Accomplishments

Arduino Mega2560 Development- The Arduino Mega 2560 is a small microcontroller that allows the user to send and store information. We had to find a way to power this processor and use its pins to relay information between the website and the signs while using a connected modem. To solve this problem we decided to use the SkyWire 4G LTE cellular modem combined with the Arduino, shown in *Figure 3.*

Daily Logs- We created a logger.php with multiple functions to be able to create daily logs and echo back the logs created. We then updated the server files to be able to use logger.php to log processes in addition to creating a JavaScript logger to log client-side events, thus ensuring all events (both client-side and server-side) are captured in one daily log text file. Server system errors that are not application-specific are logged in separate files.

Refining the Login Process- We edited the logout.php, login.php, home.php, and session.php to ensure that they successfully work together to log a user in (given that they know their credentials) and then keep them logged in as they redirect to different pages. A method for registering completely new users was also added.

Register Signs to a User- A registration page was added for users who are logged in so that they have the ability to register new signs. This registration includes adding the id of the new sign to the correct row in the customer database as well as adding a new row in the sign database.

Server Set Up- Switching from GoDaddy.com to AWS meant the server and databases had to be set up once again and the web application needed to be re-uploaded to the server. The AWS server now has all necessary installations and uploaded files and can perform the needed tasks, such as running a socket to receive and transmit data.

Module Connection- None of this project works if we can't connect the module to the TCP socket and send queries to the database. Our group successfully tested the socket connection between the server and the sign module. The server can receive HTTP requests sent by the sign module and send back a response according to the database values for that sign.

Setting I/O Pins- The Arduino continuously queries the server for new database information. When a customer uses the user interface to change database values, the socket will push out these new values to the corresponding sign module the next time it asks for updated information. The sign module uses the received information to set I/O pins accordingly; ensuring only one pin is set high at one time.

BOS Integration- The sign module has successfully been integrated into a blank-out sign for testing the final product. This has proven that the front and back end of the server work together to send updated information to the Arduino and consequently change the message displayed on the sign to the desired message.

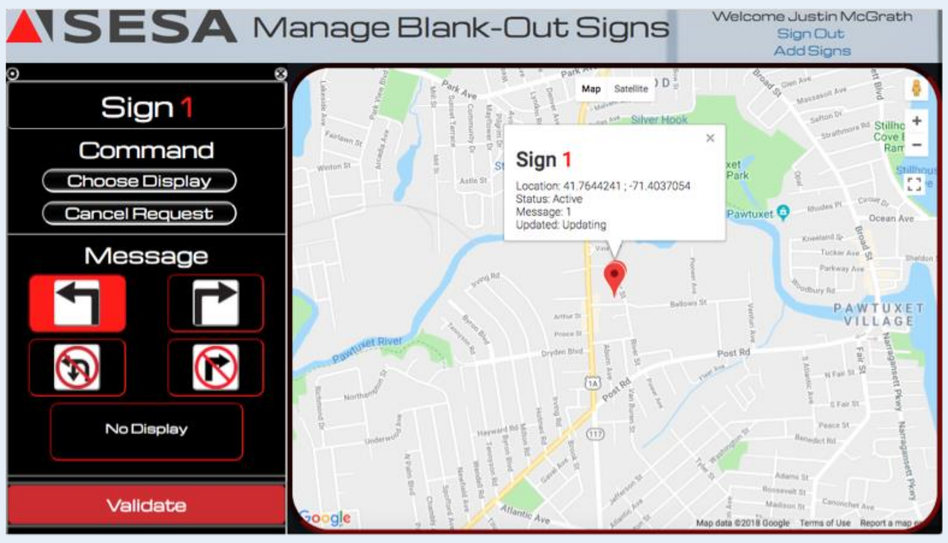


Fig. 4 Image displays the user interface of the web application after logging into the web application on a desktop (top) or on mobile (right), clicking on a marker, and setting a new message for the sign.

Anticipated Best Outcome

The anticipated best outcome for the project is a production-ready, modularized cellular communication system interfaceable with the current BOS architecture. The module will allow asynchronous transmission of both the sign's status to a remote server as well as control operations from a web application. The web application will be secure and mobile friendly as well as be able to support multiple users with multiple signs. The end user will be able to manage their BOS, relay commands to other signs and allow the user to be able to check the status of their signs. When a fault is detected, the web application will send email alerts to the users.

Project Outcome

The Anticipated Best Outcome was not achieved; the radio communications units between signs were not purchased and tested.

Figures

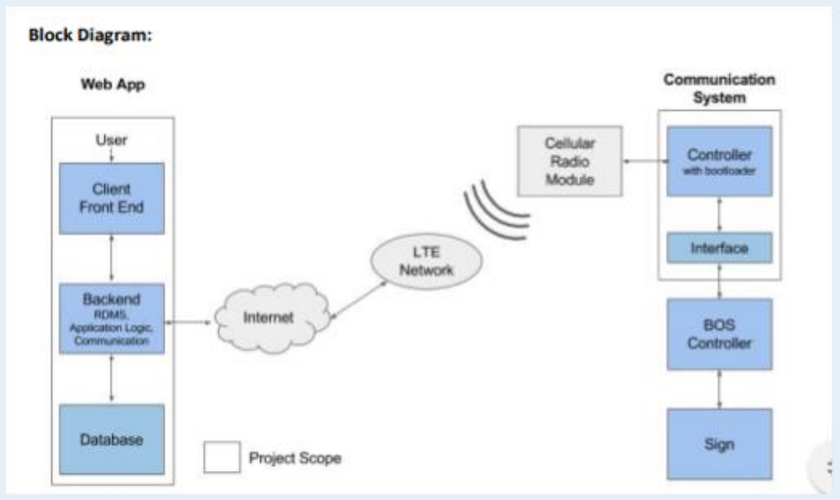
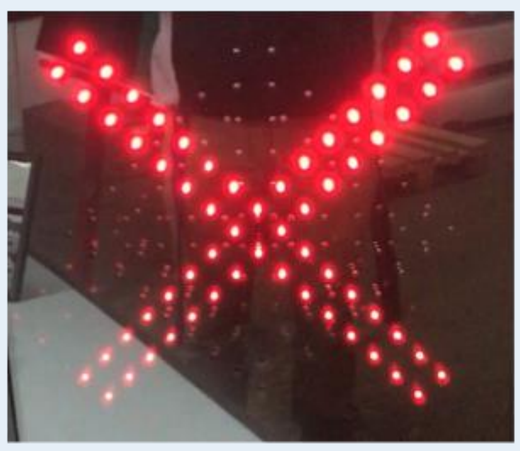
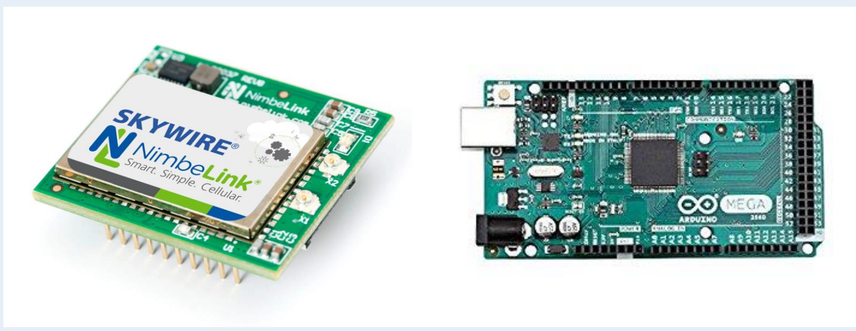


Fig. 1. Block Diagram of the system. This shows the step by step process of how the signs interact with the users, allowing them control over the messages that the sign displays.



Fig, 2. Blank Out- Signs that are to be retrofitted with the cellular module to allow connection to the server for receiving a new message.



Fig, 3. Above depicts the Skywire 4G LTE cellular modem (left) and Arduino Mega 2560 microcontroller (right) that we chose to use for communicating with the server.

ELECOMP Capstone Program Director: Professor Harish R D Sunak

email: sunak@ele.uri.edu telephone: 401-874-5859 ELECOMP Website: http://web.uri.edu/elecomp-capstone