

Wireless-Enabled Smart-Lights Hub Prototype

Senior Design December 2019 Team 16

Manimaran Govindarasu – Adviser
Ravikumar Gelli - Client

Logan Zasada – Team member
Alexander Beaver – Team member

Aaron Ramsey – Team member
Ryan Bush – Team member

Motivation

Iowa State's PowerCyber lab is known for its innovative approaches to grid management and cyber security. Because cyber-security is a relatively new field of interest, the PowerCyber lab provides training for Industry and would like to outreach to spread cyber security awareness. The system that they use for industry trainings and would like to use for outreach currently resides in the PowerCyber lab on the third floor of Coover. This system currently utilizes a light display that is synced to a power grid simulation. The problem, is that the light display is connected to an automation controller that is difficult to move and has limited space for lights. To remedy this, senior design group DEC-19-16 has been developing wireless-enabled smart light modules that will replace the light display. By using these light modules, the PowerCyber lab will be able to improve industry trainings by increasing the number of simulation components. The lab will also expand its ability to do outreach as they can bring the light modules anywhere and perform demonstrations.

Design Requirements

Functional Requirements

- Scalable for 100+ light modules
- Magnetically mountable
- Individually addressed modules
- Battery life of 8+ hours
- Physical size approximately equal to existing lights
- Charging circuit on board module
- Interface with existing program for light control
- Programmable for different colors
- User interface mechanism on module
- Reconfigurable mapping of light modules to simulation components

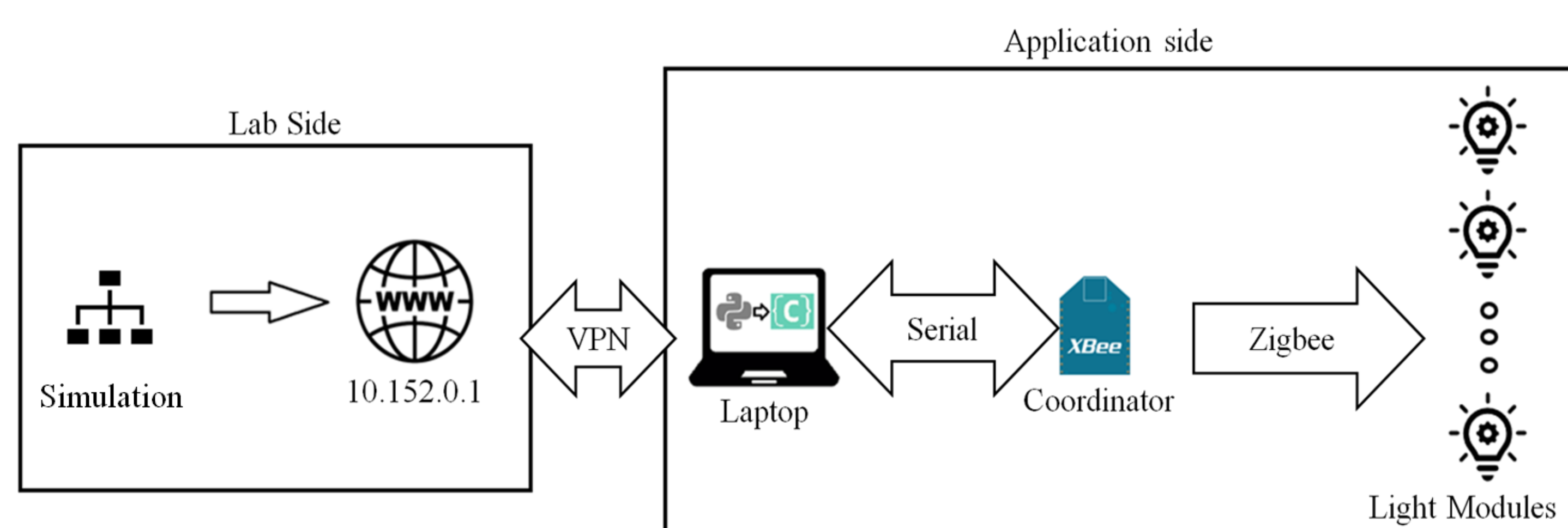
Non-Functional Requirements

- Aesthetically pleasing
- Clean finish
- Easy to transport
- Easy to configure

Engineering Constraints

Our project was constrained by time, requested communication standard, and available testing and measurement equipment. For time, we had 2 semesters to do the project which made it difficult to properly prototype, document, and design our project. Our client requested we use Zigbee, which constrained our project because a few members of our team had experience in other communication standards, but not in ZigBee. Finally, there was some testing and measurement equipment we wanted to use to adequately test our devices that wasn't available to us. Most notably, a variable load to test our battery charging circuit.

Design Approach



System Components

Simulation

- Simulates power grid
- Can be used to simulate grid attacks
- Connects to server in the lab

Laptop

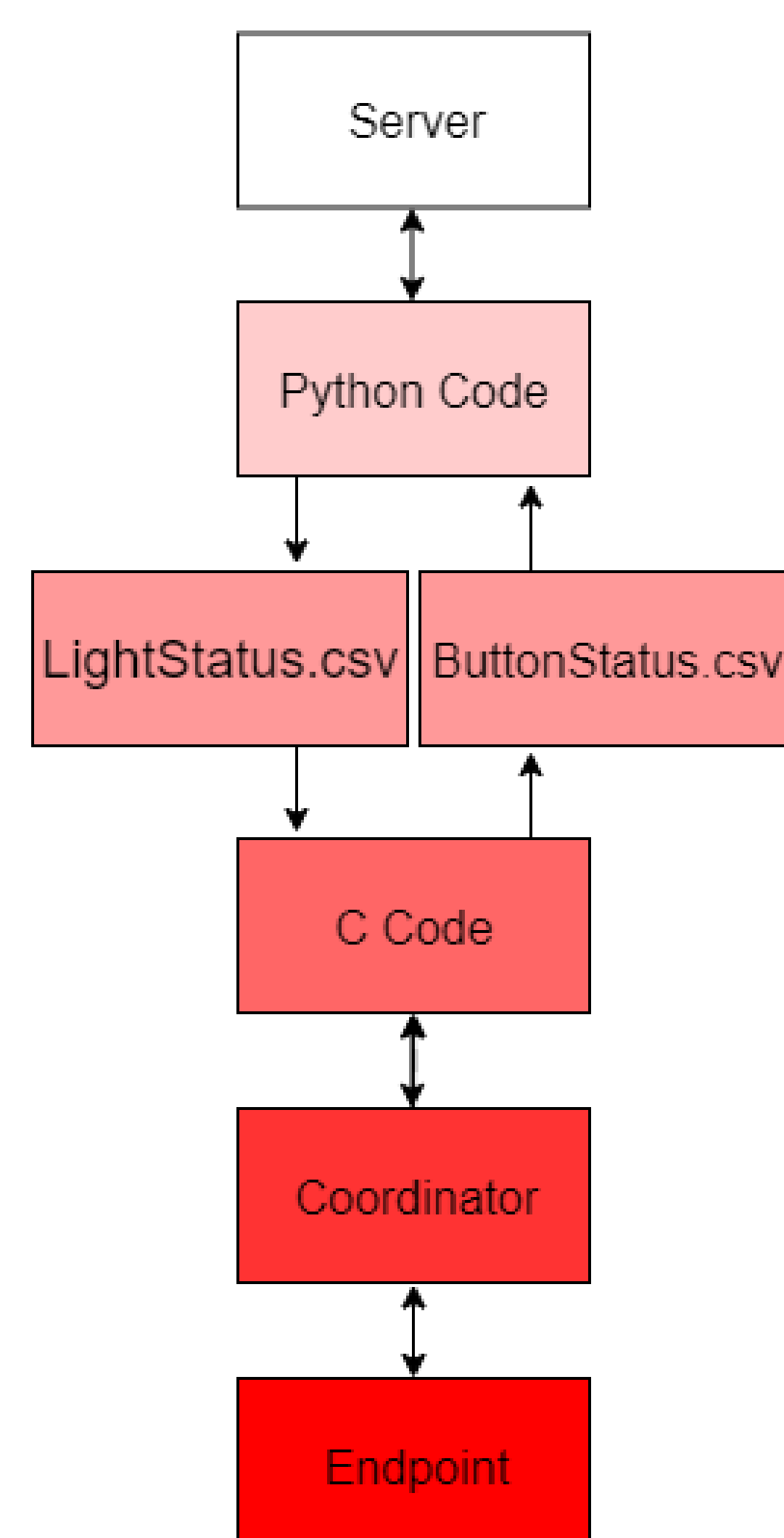
- Connects to lab server via vpn
- Connects to coordinator ZigBee device via USB cable
- Can be taken any where
- Runs program to control light modules

Coordinator Zigbee

- Connects to laptop via USB cable
- Sends commands to light modules via Zigbee

Light Modules

- Connects to laptop via USB cable
- Sends commands to light modules via Zigbee



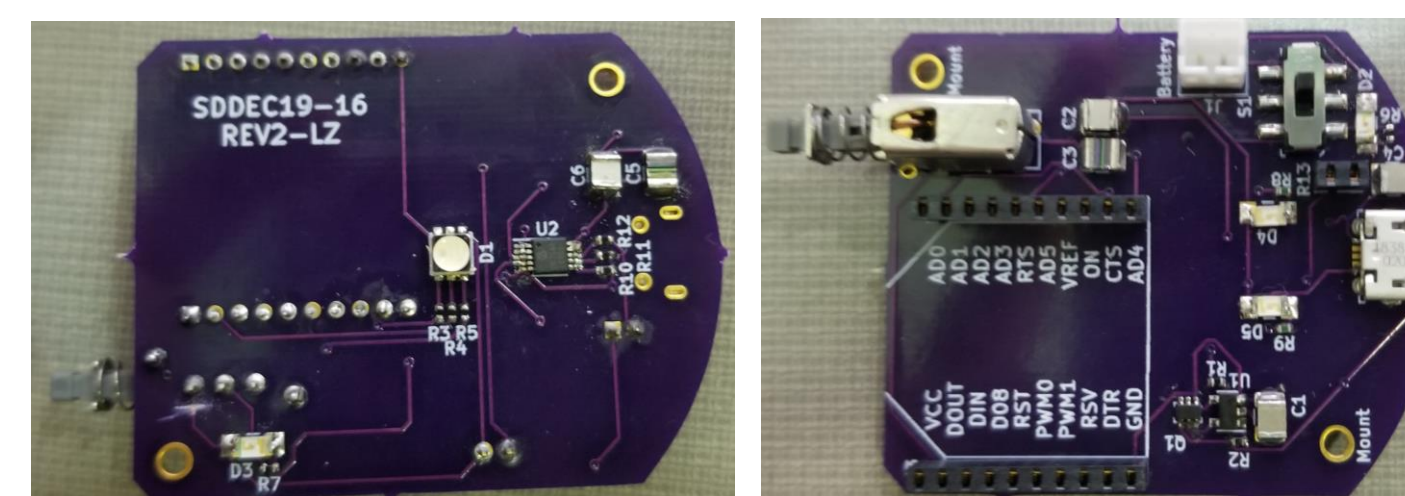
Technical Details

Hardware

Our main hardware task is to create multiple light modules. These modules must take in data from the XBee device, then control an LED based on the data that the XBee sees. We also added a button functionality, where pressing a button sends a communication from our module to the coordinator.

To achieve this task, we designed a PCB (pictured below) that contained the XBee device, the main LED as well as status LEDs, and some charging and battery protection circuitry.

Inputs and outputs to the PCB are: Battery, main power switch, toggle button, charging port, thermistor (to monitor battery temp), the Port for the XBee device, and the LEDs.



Software

Languages used: Python and C. A Python script acts as a middleman between the server and the C code. Its main job is to write and read CSV files that have mappings from lights to their red, green, and blue light statuses

The C script interfaces with the XBee coordinator through a serial connection. Its job is to send serial commands to the coordinator using the data in one CSV file. If the light status on the coordinator was to change, the C program reads the serial from the coordinator and populates a second CSV file for the Python script to read.

Libraries used: The Windows API for file reads and writes.

Development tools: GNU Project debugger for C core dump analysis and XTCU for programming the XBee coordinator and other Zigbee devices.

Housing

To house the module and meet the "aesthetically pleasing" requirement, we house the modules in a circular white plastic housing. This housing screws together from the bottom, and has a glass dome on top to diffuse light. Inside, the PCB and battery are properly secured, and there is a magnet to allow the device to hang on a whiteboard or other magnetic board.



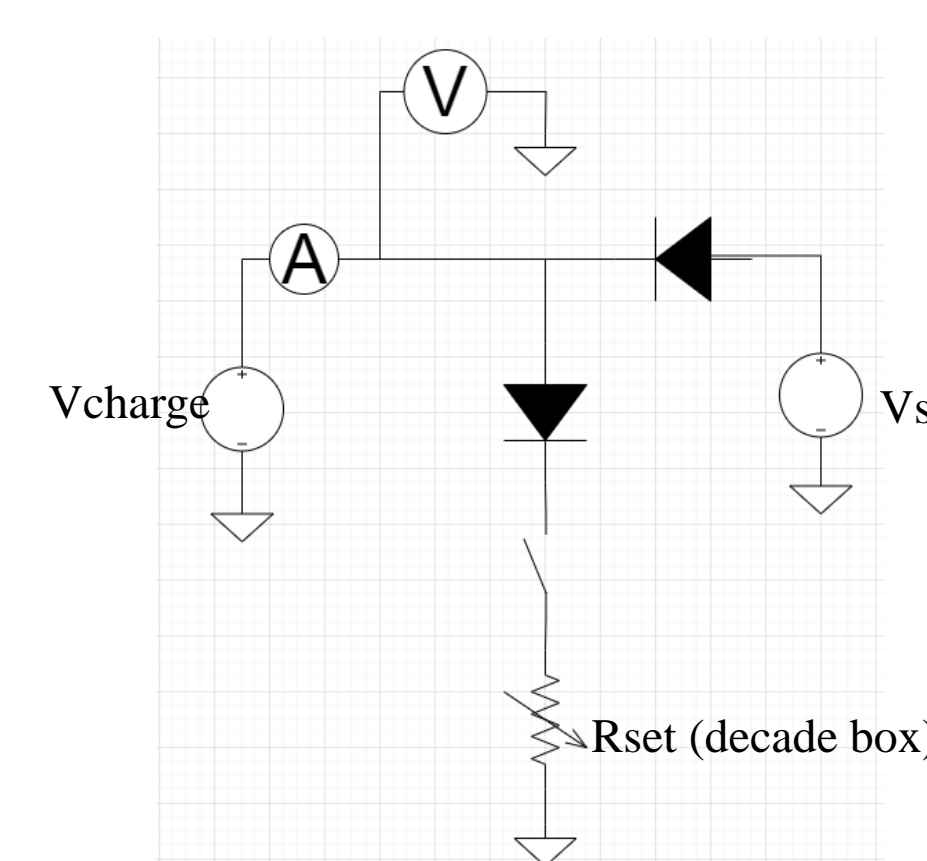
Intended Users

There are two main groups of intended users. The primary users are the members of the PowerCyber Lab, a state-of-the-art testbed to protect the nation's power grid from cyber-attacks. The secondary users are industry partners and the general public. This interaction will consist of the secondary users observing the the demonstration and even being able to press a button on the modules to simulate a cyber-attack.

Testing

Hardware

To test the hardware, breadboards were used created mock circuits using breadboard to verify the functionality of proposed circuits. Digi-XTCU was used to create firmware for the radio modules for the purpose of testing hardware. To test the charging circuit, the schematic seen below was used. Where Vset was used to set the charging voltage and Rset was used to simulate the batteries resistance



Software

Virtual serial ports were created to facilitate testing when there were no available devices to use. Before moving to a live environment, mocked XBee commands were sent to the serial port and the system response was monitored and adjusted accordingly. Once the system works accurately enough, the testing moves to a live environment with a light active.

Operating Environment

This whole system will be used in a conference-type environment. The PowerCyber lab members will be in a conference room with a whiteboard or other magnetic board. They will project a map on to the board, and place our light modules at certain points of interest that represent a part of the power grid system.

Relevant Standards

- IPC-2221B
 - A generic standard on PCB designs
- ZigBee / IEEE 802.15.4
 - Low power and low data rate communication protocols