Senior Capstone Project I

Project Proposal

Project Title: Development of an Intelligent Building Energy Management Platform

Brian Lauer Elliot Watkins Advisor: Dr. Suruz Miah

Department of Electrical and Computer Engineering
Bradley University
1501 W. Bradley Avenue
Peoria, IL, 61625, USA

Monday, November 30, 2020

Outline

1. Introduction
2. Background Study
3. Functional Requirements
4. System Architecture
5. Preliminary Work
6. Parts List
7. Timeline and Milestones

Introduction

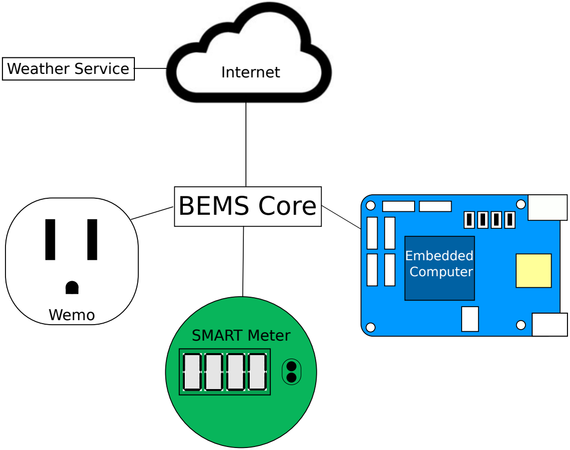


Figure 1: High level architecture

Introduction

- BEMS Core will interface with as many devices as it has support for
  - Wemo and Embedded Computer will be fully functional by end of semester
  - User will be able to control these devices directly from the Web Server
- In the Spring, we hope to simulate a microgrid with MATLAB Simscape and an HVAC System with Simulink
  - The microgrid simulation will allow us to virtually measure a power meter
  - An HVAC System in Simulink could be controlled with a Linear Quadratic Regulator (LQR) based algorithm that can regulate the building temperature
- If time allows, we will also have the BEMS Core ping a weather service for updates on severe weather as well

Background Study

- An existing BEMS described in [?] incorporates its own sensors for:
  - temperature
  - humidity
  - luminosity
  - air quality
  - motion
- The data from these sensors is fed to the BEMS core where decisions are made for the HVAC system and lighting of the building
- While we will only be simulating HVAC with Simscape, our BEMS core is similar to this project in that we are using IoT as a communication platform

Background Study

- MPC (Model Predictive Control) used in [?]
- Building control split into two levels for building automation improvement
- Examples of the usage of MPC given are temperature control and interaction with smart grids

Background Study

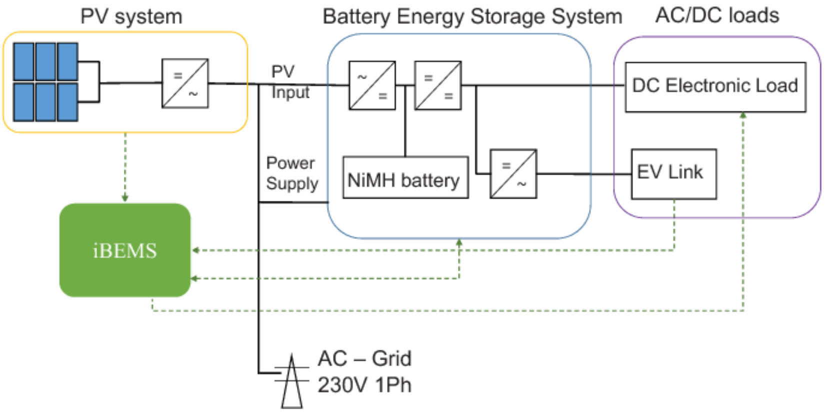


Figure 2: Photovoltaic (Battery Energy Storage) BES system simulated in [?]

Functional Requirements

- The minimum viable product will be to connect to a Wemo Insight Smart Plug and a Embedded Computer (BeagleBone Blue)
  - The WeMo Plug will simply be turned on and off from the Web Server. The BEMS Core will also record its power usage
  - The only functionality needed on the BeagleBone Blue will be PWM input to the motor drivers
- A database (Apache Cassandra) will be implemented for recording power usage

System Architecture

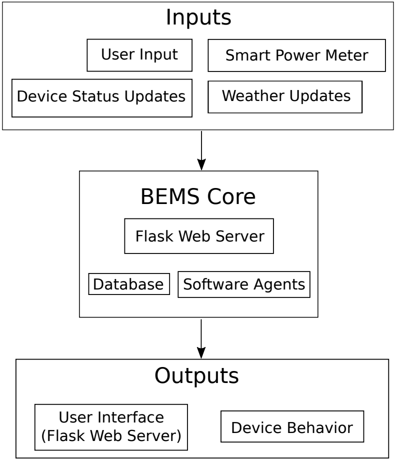


Figure 3: Functional block diagram of the BEMS

System Architecture - Inputs

- Users
  - Adjust the operation of a device (turn up the temperature of an AC unit, for instance)
  - Turn a device on or off
- Smart power meters will be able to connect the platform and give real-time updates of the building's overall power usage
- Connected Devices
  - The Wemo Switch and Embedded Computer will have master-slave relationship with the BEMS Core
  - The BEMS Core will send commands through the router to download power data
- The BEMS Core regularly receives weather updates from a weather service

System Architecture - BEMS Core

- Python and Javascript
  - Python is used to control devices on the back end
  - Javascript is used to run the web page (user interface)
- Flask Web Framework
  - Handles rendering device metadata and time-series data to the web page
  - Routes URLs to view functions
- Apache Cassandra and SQLite databases
  - Apache Cassandra is a NoSQL database management system for storing time-series device data
  - SQLite is a simple relational database management system that will be used to store device metadata
- Bootstrap and JQuery
  - Bootstrap is a CSS framework for creating modern web pages
  - JQuery is a Javascript library to help with DOM traversal, event handling, and AJAX requests

System Architecture - Outputs

- BEMS core will output all data from connected devices to be accessed through the web server via a browser (accessed by any device that can browse the web)
  - On/Off Status
  - PWM Setting
- When prompted by user input, the BEMS core will run a python script to plot energy usage of a given device on a given day; this plot will appear on the active devices page
- Warning on the Web Server home page about potential energy loss due to storms

BRADLEY University

Preliminary Work - Design

- The platform incorporates multiple agents to help facilitate communication between various parts of the software
- Discovery agent
  - Traverses through available API files and uses the findDevicesAPI call to locate devices on the network
  - Later on auto discovery will be implemented to allow devices to be shown in the software whenever a new supported device joins the network
- Control agent
  - General agent for all supported devices, capable of retrieving and modifying the status of a device available on the network through API calls
  - Supports starting up a separate thread to periodically query the device for information like ON/OFF status and power consumption

Preliminary Work - Design

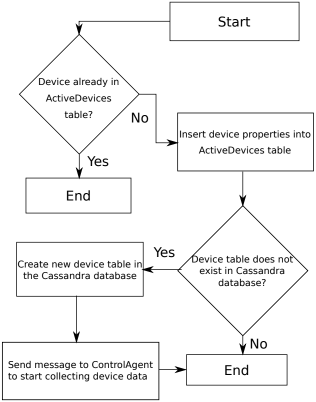


Figure 4: `DiscoveryAgent.setDeviceToActive()` flow chart

Preliminary Work - Design

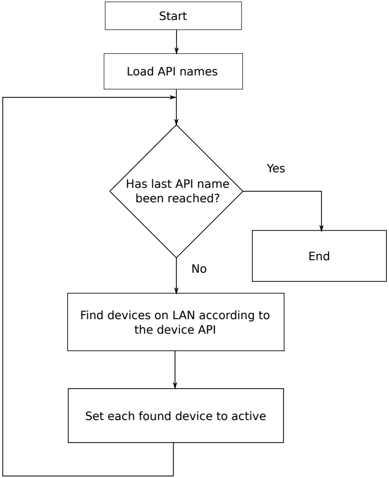


Figure 5: `DiscoveryAgent.searchForDevices()` flow chart

Preliminary Work - Design

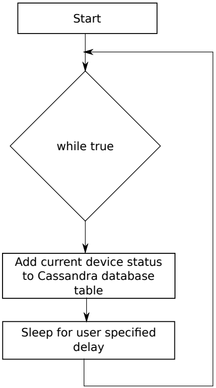


Figure 6: `ControlAgent.periodicQueryBehavior()` flow chart

Device APIs

- setState - Setter for Wemo Insight Switch and motor controllers on Embedded Computer
- getState - Returns power usage and on/off status for Wemo Insight Switch and PWM setting for Embedded Computer
- findDevices - Returns IP address of supported devices on the network
- findMetadata - Returns manufacturer, name, and MAC address of supported devices
- For Wemo Insight Switch, IP multicasting and Universal Plug and Play are used to send messages
- We are using a Beaglebone Blue for an embedded computer and so far have only used IP multicasting for communication

Experimental Setup

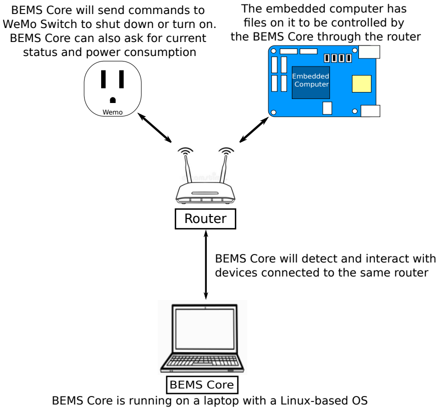


Figure 7: Lab Setup for Testing and Development

Preliminary Work - Experimental Activities

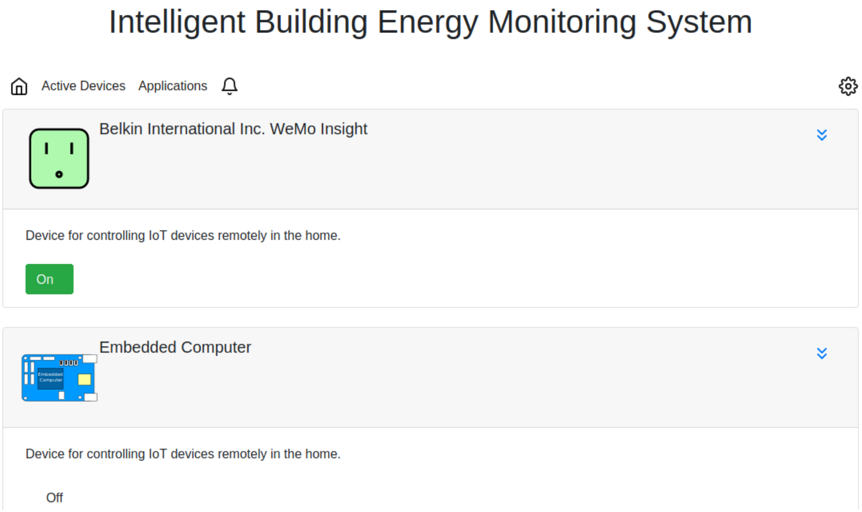


Figure 8: Screenshot of Web Server (Active Devices Page)

Preliminary Work - Experimental Activities

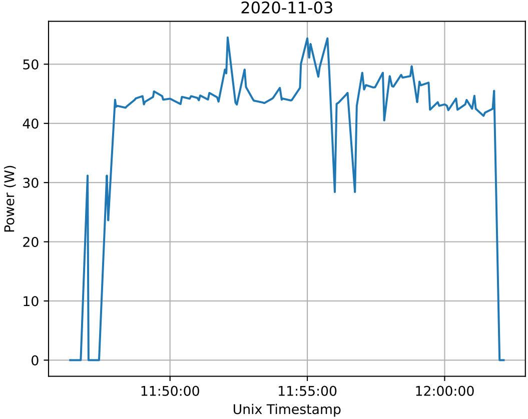


Figure 9: Plot of WeMo Switch power usage 2020-11-03

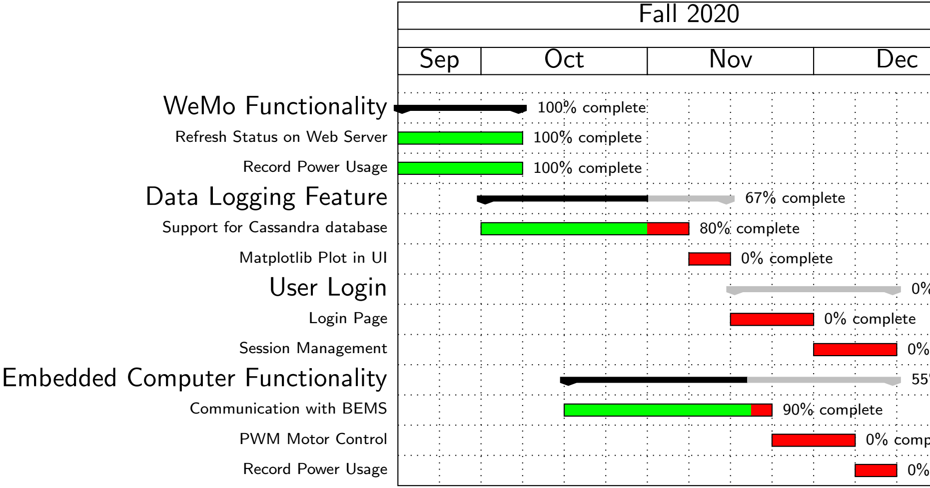
Preliminary Work - Experimental Activities

- Up to this point, work has been completed on the web user interface which was developed with the Bootstrap CSS framework and the JQuery Javascript framework
- When a request is sent to the active devices URL, the discovery agent will ping all possible supported devices on the network and newly discovered devices will appear on the active devices page
- the refresh button will cause the position of the slide switch switch shown to change depending on whether the WeMo switch is on or off
- The top nav bar will link to other pages including the home page, applications page, notifications and settings page (active devices is currently the only page implemented)

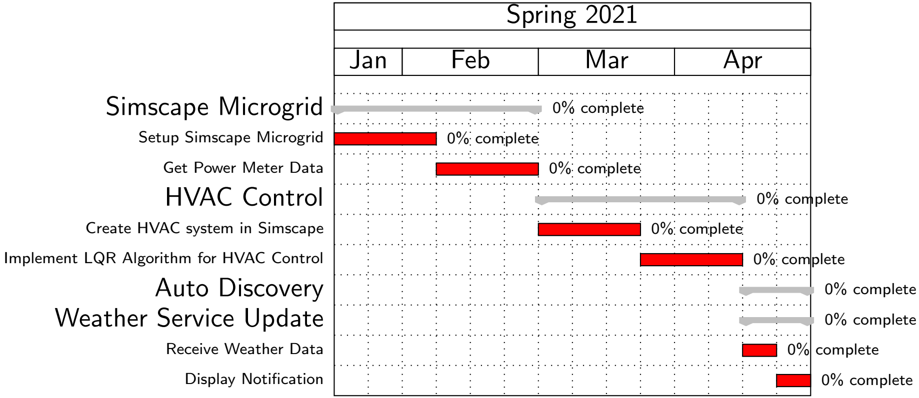
Parts List

- WeMo Insight Smart Plug
- BeagleBone Blue
  - Octavo OSD3358 Microprocessor
  - Wifi/Bluetooth
  - IMU/Barometer
  - Power Regulation and state-of-charge LEDs for a 2-cell LiPo
  - H-Bridges
  - Connectors for 4 DC motors and 8 Servos
- JST-ZH connector (for BeagleBone Blue Motor Connections)
- DC Motor
- ECE department laptop
- ECE department lab computers

Timeline and Milestones



Timeline and Milestones



Summary

- Development of a platform to control devices
- Ability to read device status (power usage, state)
- Smart grid simulation using Simscape and Matlab
- HVAC control algorithm using Linear Quadratic Regulator (LQR)

References I

- BEMOSS 3.5
  https://github.com/bemoss/BEMOSS3.5
- Flask Documentation
  https://flask.palletsprojects.com/en/1.1.x/
- Datastax Python 3 Cassandra Driver
  https:
  //docs.datastax.com/en/developer/python-driver/3.24/
- Apache Cassandra Database
  https://cassandra.apache.org/

BRADLEY University

References