Development of 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

Thursday, October 19, 2020

Outline

Introduction

- Why BEMS?
  - Can help save on energy costs
  - Less impact on environment
- How can the IoT help?
  - WiFi integrated into most buildings
  - Our Web-based approach simplifies development and end-user experience
- Overall Goal
  - Create a platform in which users can login and access devices connected to a buildings energy supply
  - Allow the user to closely monitor energy usage throughout a commercial or residential building.

BRADLEY University

Introduction

Play Video

System Architecture

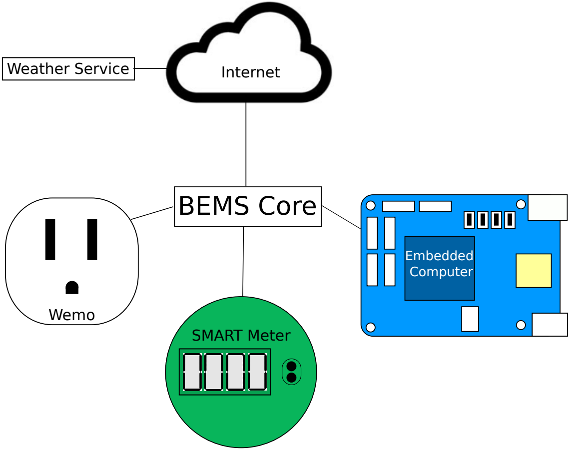

Figure 1: High level architecture

System Architecture - Supported Devices

- WeMo Switch
  - Turn on and off from web server
  - Record power usage month to month
  - Plot power usage for a given time frame using MatPlotLib library in Python
- Embedded Computer
  - Connect to BeagleBone Blue
  - Control motor drivers from web server
- Smart Meter
  - Cannot physically implement this
  - MATLAB has ability to simulate power meters

System Architecture

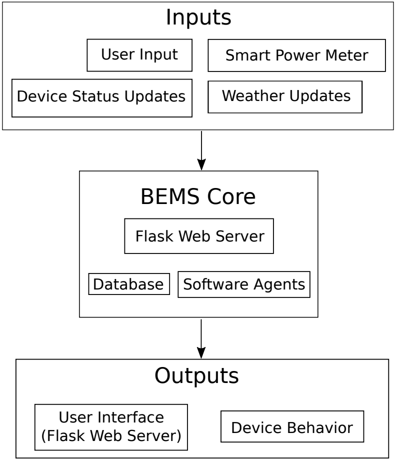

Figure 2: Functional block diagram of the BEMS

System Architecture - Inputs

- Users will adjust the operation of a device (turn up temperature of AC unit or turn on/off a light)
- Smart Power meters (BEMS core will need to receive power data from MATLAB scripts) will provide real-time power consumption for the entire building
- Connected devices will give status updates when prompted by the BEMS Core
- The BEMS core will scan a weather service website to receive updates on severe weather that may impact power transmission

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

BRADLEY University

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)
- All available data on any smart device will be able to be perused via the platform

BRADLEY University

Modes of Operation

- **Mode #0: Device discovery mode**: networked IoT devices will be automatically discovered on the local building network
- **Mode #1: Manual device control mode**: the state of each active device will be controllable through an easy-to-use web interface and command line application
- **Mode #2: Power reporting mode**: a page will be available for reporting overall building power consumption and individual device power consumption

End-Use Requirements

- Device behavior should be implemented to seamlessly support different status configurations
- Users (building admins, employees, home owners) will have the ability to see a plot of the past energy usage of a device
- Python's Matplotlib package will form the basis for the plotting functionality
- Support for new devices should be easily developed through multiple layers of abstractions

BRADLEY University

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/