

# Intelligent Building Energy Management System

Brian Lauer    Elliot Watkins  
Advisor: Dr. Suruz Miah

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

Tuesday, April 27, 2021

# Outline

1 Introduction

2 Architecture

3 Hardware

4 Software Implementation

5 Conclusion

# Outline

1 Introduction

2 Architecture

3 Hardware

4 Software Implementation

5 Conclusion

# Introduction

- Problems that iBEMS will address
  - 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 2 or 3 IoT devices connected to a building's energy supply
- Allow user to closely monitor energy usage for residential building

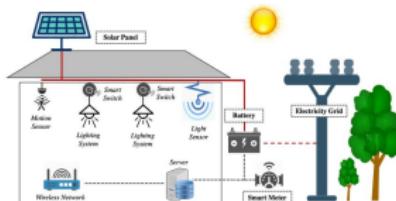


Figure 1: General BEMS

# Outline

1 Introduction

2 Architecture

3 Hardware

4 Software Implementation

5 Conclusion

# Architecture

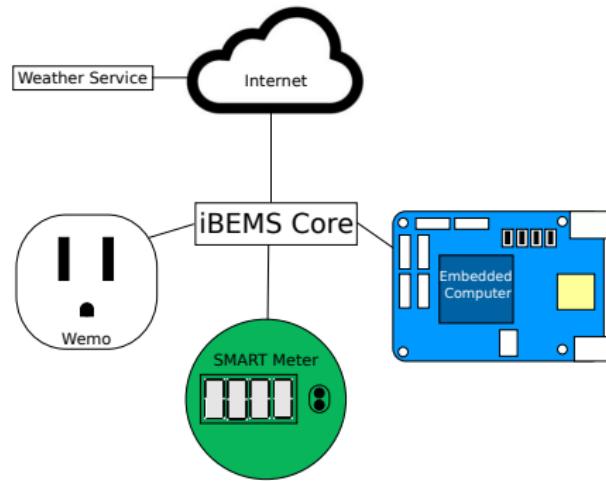
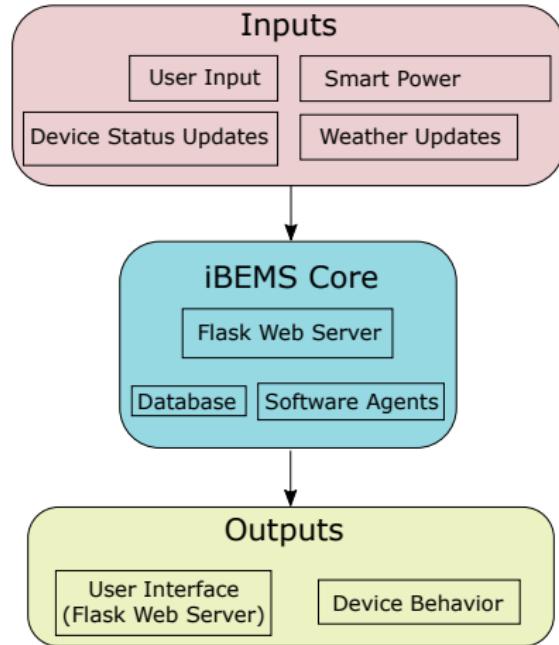


Figure 2: High Level Architecture

- iBEMS currently has support for 2 IoT devices (WeMo Insight Switch and embedded computer) and can download weather data for the user
- A microgrid with a smart meter could theoretically simulated and connected to by the BEMS core

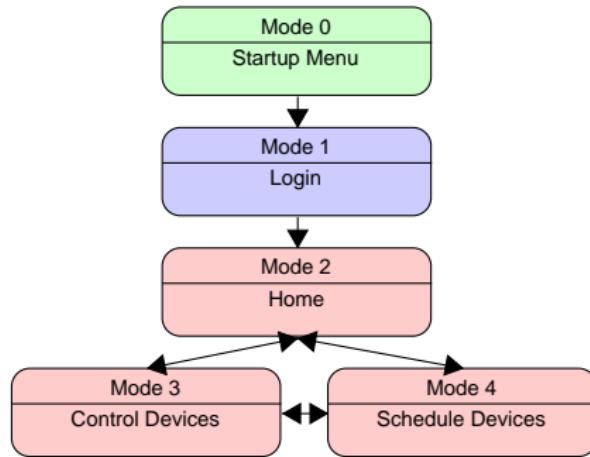
# Architecture



- iBEMS receives input from users, connected devices, and will automatically download weather data
- The core of iBEMS consists of a web server, a database and software agents which handle interactions with connected devices
- Power data, device status, and schedules for each device will be output to the web server. Additionally, connected devices respond to commands from the iBEMS

Figure 3: Functional Block Diagram

# Operational Modes



- ➊ Startup menu
- ➋ Login - system administrators can add and configure users
- ➌ Home - view weather data
- ➍ Controlling devices
- ➎ Scheduling devices

Figure 4: Operation Modes Interaction

# Agent Communication Architecture

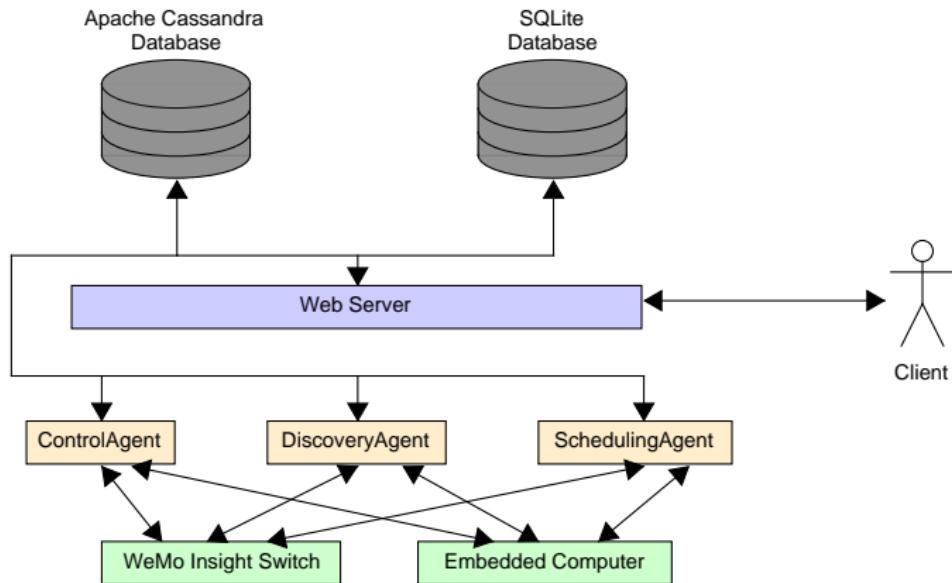


Figure 5: Interconnection between system components

# Outline

1 Introduction

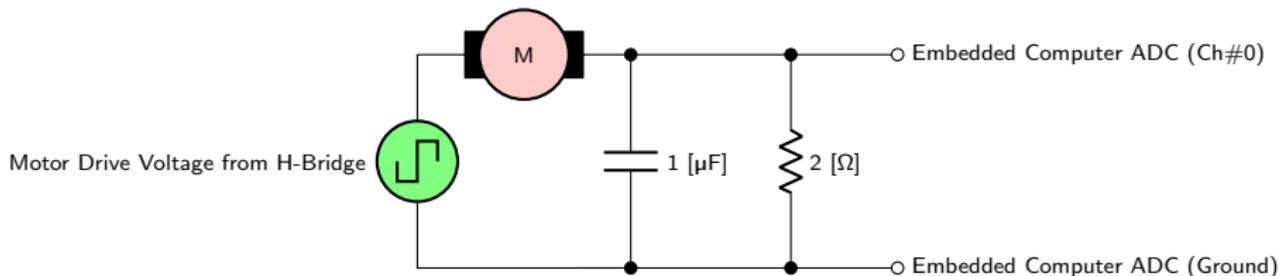
2 Architecture

3 Hardware

4 Software Implementation

5 Conclusion

# Embedded Computer Circuit for Power Usage Data



- The embedded computer does not have the ability to measure power, but it does have ADC
- Here, the power from the motor driver is being run through a resistor. The ADC reads the voltage across the resistor and since resistance is known, current can be calculated for 1 motor

# Embedded/Robotics Computer

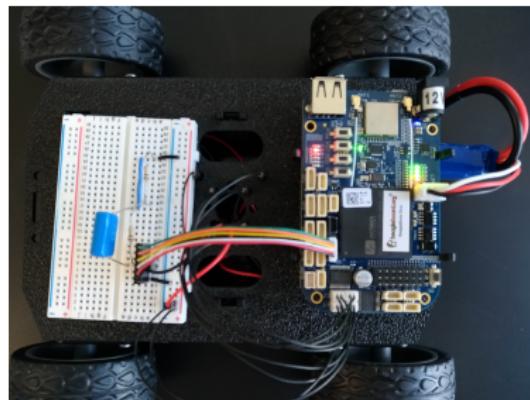


Figure 6: Embedded Computer Not Connected

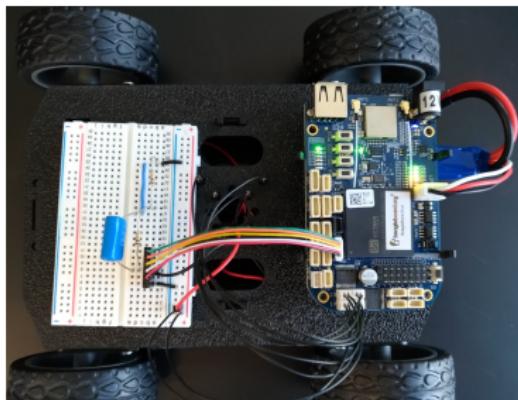


Figure 7: Embedded Computer Connected

- Embedded computer accepts control commands and sends data to the server
- Red LED indicates the embedded computer is not connected and green LED indicates the embedded computer is connected

# WeMo Insight Switch



- Appliance is plugged into the WeMo Switch
- WeMo Switch can internally turn power on and off
- iBEMS will control the WeMo Switch remotely

Figure 8: WeMo Insight Smart Plug

# Overall Experimental Setup



Figure 9: Overall View

Tested the WeMo Insight Switch with a bank of lights and the embedded computer with a robot chassis

# Outline

1 Introduction

2 Architecture

3 Hardware

4 Software Implementation

5 Conclusion

# Control Agent Functionality

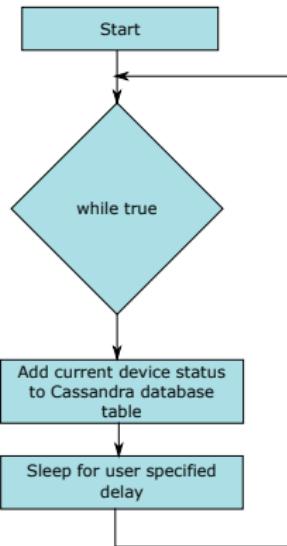


Figure 10: periodicQueryBehavior Flow Chart

# Discovery Agent Functionality

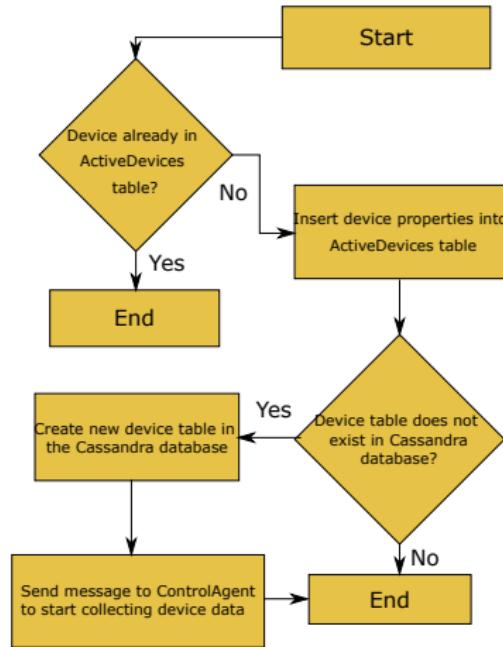


Figure 11: setDeviceToActive Flow Chart

# Discovery Agent Functionality

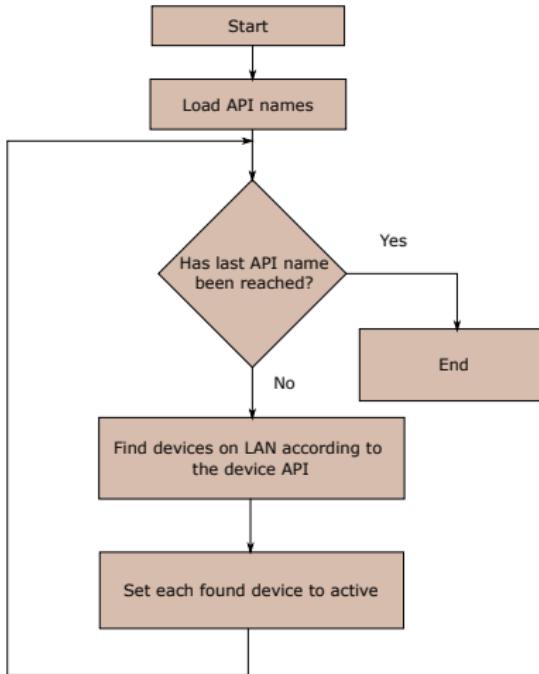


Figure 12: `searchForDevices` Flow Chart

# Embedded Computer Receiver Program

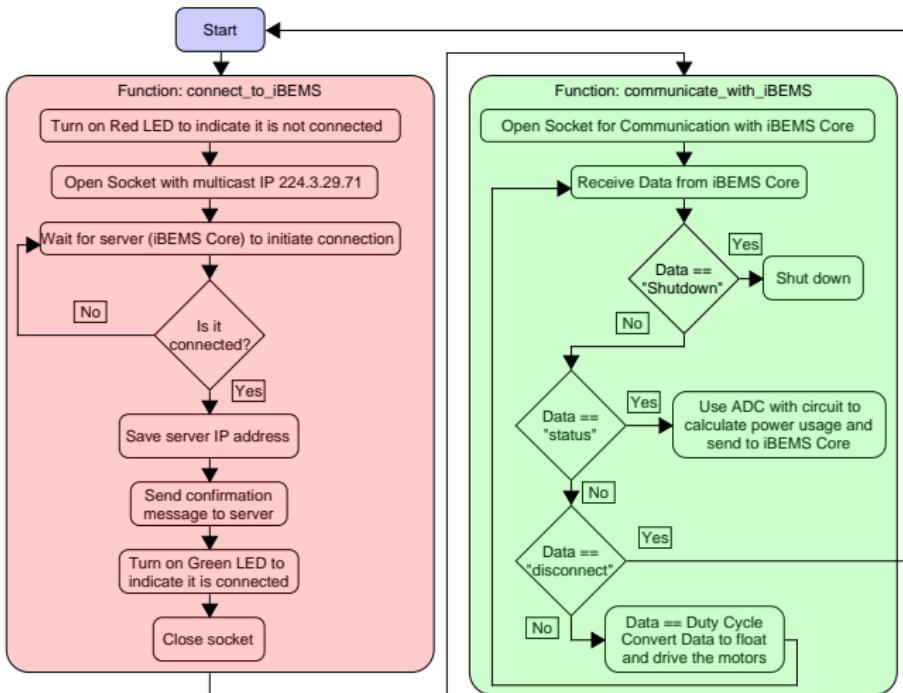


Figure 13: Receiver Running on Embedded Computer

# Startup Menu



Figure 14: Desktop GUI



Figure 15: Popup Error

- Clicking "Start iBEMS" will launch the web server and all 3 agents as well as launch the computer's web browser and load the login page automatically
- If the user accidentally clicks "Start iBEMS" while it already running, a warning message will appear to prevent errors
- Clicking "Stop iBEMS" will stop the web server and agents and close the web browser

# Startup Menu Functions

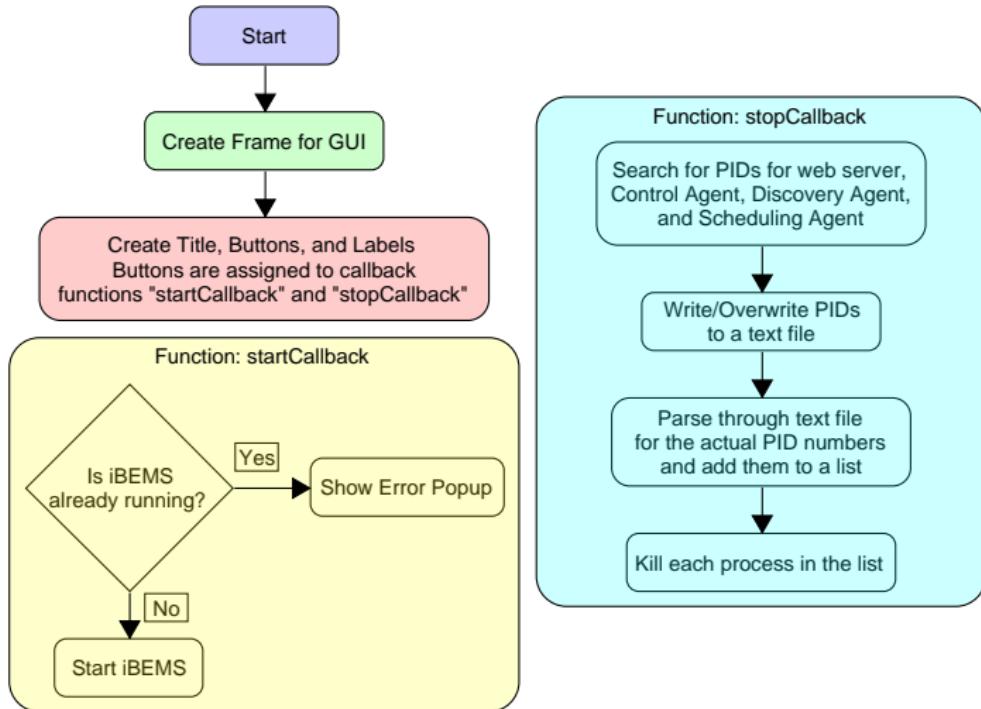


Figure 16: Diagram for Startup Menu Functionality

# Home Page



Figure 17: Home Screen

- This will be loaded when iBEMS is first booted and it simply shows some current data about the weather
- The Home Page is also accessible by clicking the Home Icon in the upper left corner

# Active Devices Page

Intelligent Building Energy Management System

Active Devices Device Scheduling

Belkin International Inc. Lamp

Smart plug for monitoring and controlling receptical loads remotely in the home.

Off

BeagleBoard Embedded Computer

Single board computer capable of collecting sensor data and controlling actuators.

0

Figure 18: Active Devices Page

- Page allows users to directly control available devices in real time
- WeMo Insight Smart Plug can be turned on and off
- PWM duty cycle of the embedded computer can be adjusted in increments of 0.1

# WeMo Power Plot

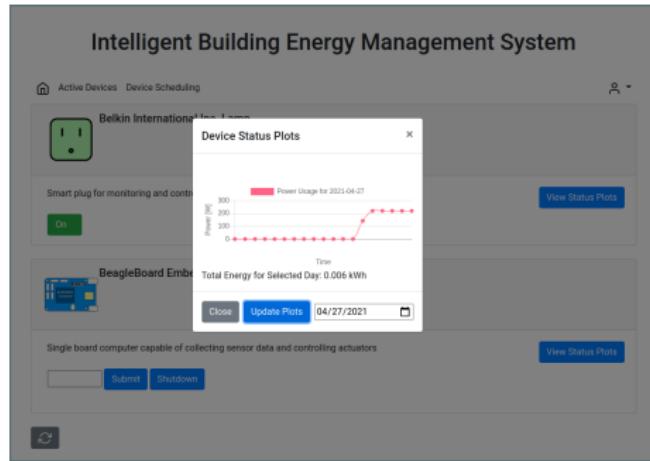


Figure 19: Active Devices Plot

- Clicking "View Status Plots" launches modal
- User can select any day and view the power usage

# Scheduling/Applications Page

Intelligent Building Energy Management System

Active Devices Device Scheduling

Weekly Scheduling

Sunday Monday Tuesday Wednesday Thursday Friday Saturday

Belkin International Inc. Lamp

ID	API	Status	Starting at	Ending at	Confirm	Remove
1	WeMo	0.2	4:30 PM	5:00 PM		
1	WeMo	-0.3	6:00 PM	6:15 PM		

Add New Period

Update Schedule

BeagleBoard Embedded Computer

ID	API	Status	Starting at	Ending at	Confirm	Remove
2	Beaglebone	Duty Cycle: 0.2	4:30 PM	5:00 PM		

Figure 20: Scheduling Page

- Clicking on "Device Scheduling" on the nav bar loads the weekly scheduling feature
- User can input whatever times they want with devices to be in a specified state (on/off for WeMo plug and duty cycle for the embedded computer)

# Outline

1 Introduction

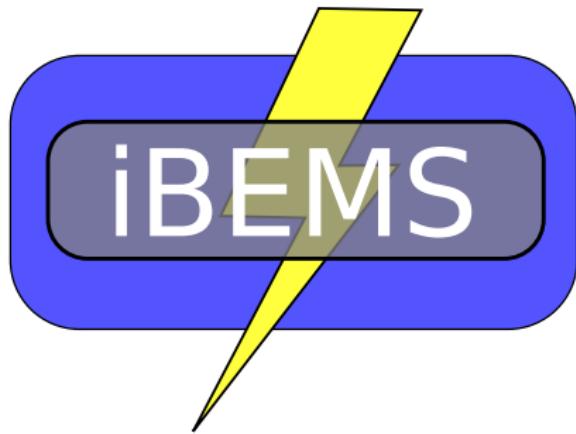
2 Architecture

3 Hardware

4 Software Implementation

5 Conclusion

# Conclusion



- Advantages: open source, simple interface, modular agent-based code base
- Disadvantages: can only operate on a LAN, lacks features of a commercial BEMS system
- Challenges: adapting the platform for an enterprise network in a commercial building, security features

# Conclusion

## Further Improvements

- Refactor the user interface
- Granular user access control
- Refactor agent communication
- Intelligent algorithms
- Mobile app
- Persistent device information storage