Building Energy Management Internet of Things

Reece Bachman Robert O'Malley Jordan Ingram Advisor: Dr. Suruz Miah

Monday, November 5, 2018

Outline

Introduction

- Building Energy Management Open Source Software (BEMOSS) is an open source Internet of Thing (IoT) software developed under the Department of Energy
- We are tasked with creating a new device that can be controlled through this software over the internet
- We are creating an interface to control a DC motor through BEMOSS
  - Control curtains to regulate interior temperatures
  - Close/ open barriers
- We will also create a machine learning algorithm that can be supported within BEMOSS to reduce power consumption

Outline

Background Study

Literature

- The previous literature that has been released on BEMOSS has been describing what it can do instead of how to do it
- Currently BEMOSS is being used by Virginia Tech and McNeese University to monitor a microgrid

Outline

- The BEMOSS team has thus far successfully incorporated control of the following devices into BEMOSS:

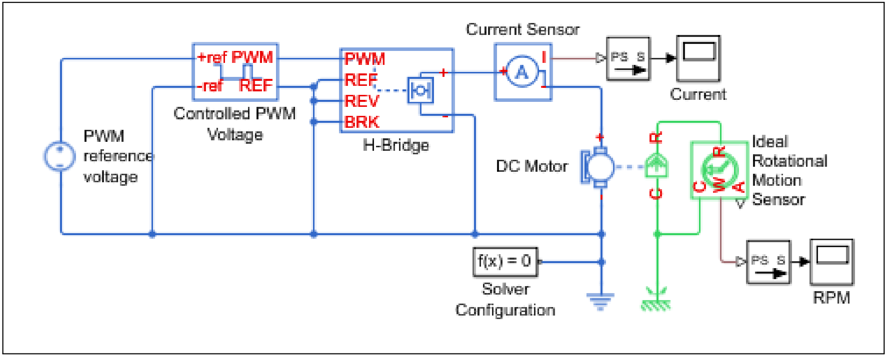
Device Model	Vendor	Protocol
HVAC controller		
CT30 w/ Wi-Fi USNAP module	Radio Thermostat	Wi-Fi
CT50 w/ Wi-Fi USNAP module	Radio Thermostat	Wi-Fi
PL-M1000RTU/M2000RTU	Prolon	Modbus RTU
VC1000/VC2000	Prolon	Modbus RTU
Lighting load controller		
WeMo light switch	Belkin	Wi-Fi
Philips Hue	Philips	Wi-Fi/Ethernet
LMRC-212-U	Wattstopper	BACnet MS/TP
Plug load controller		
WeMo switch	Belkin	Wi-Fi
WeMo insight switch	Belkin	Wi-Fi
LMPL-201	Wattstopper	BACnet MS/TP
Sensor		
LMLS-400	Wattstopper	BACnet MS/TP

Outline

Background Study
Challenges

- Create a new BEMOSS supported device
- Implement BEMOSS on a new device, such as a raspberry pi
- Develop and implement an algorithm for BEMOSS to use to efficiently utilize power consumption in a HVAC system

Outline


+ref PWM
-ref REF
Controlled PWM Voltage
PWM
REF
REV
BRK
H-Bridge
Current Sensor
PS S
Current
PWM reference voltage
DC Motor
Ideal Rotational Motion Sensor
f(x) = 0
Solver Configuration
PS S
RPM

Outline

BRADLEY University

Subsystem Level Functional Requirements

Motor Requirements

- The simulation must be modeled based on the real system in the lab
- The model must be run by a PWM signal and H-bridge to allow for feedback control later
- The model must produce a power calculation to be used later for the machine learning algorithm

Outline

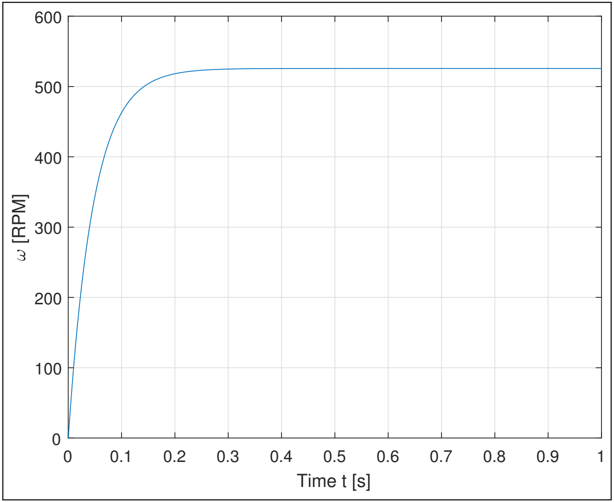
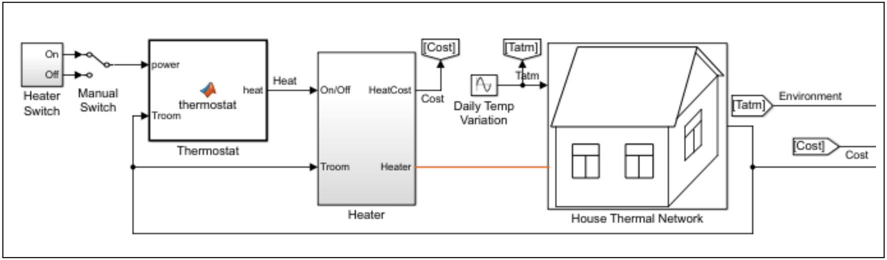


Figure: Motor Simulation Results

Outline

Subsystem Level Functional Requirements

Block Diagram



Outline

BRADLEY University

House Requirements

- The model must give a realistic thermal behavior of a room
- Modeled after work done by a previous work, the system is modeled after the system of equations:

$$\frac{\mathrm{d}\mathbf{T}^e}{\mathrm{dt}} = \begin{bmatrix} \dot{T}_1^e \\ \dot{T}_2^e \end{bmatrix} = \begin{bmatrix} -\frac{u_{cc}A_{cc}}{M_{cc}C_v} & \frac{Q_w \rho_w C_{\rho w}}{M_{cc}C_v} \\ 0 & -\frac{Q_w \rho_w C_{pw} + U_t A_t}{V_t \rho_w C_{pw}} \end{bmatrix} \begin{bmatrix} T_1^e \\ T_2^e \end{bmatrix} +$$

$$\begin{bmatrix} \frac{U_{cc}A_{cc}}{M_{cc}C_v} T_{amb} - \frac{Q_w \rho_w C_{pw}}{M_{cc}C_v} T_{wo} \\ \frac{U_t A_t}{V_t \rho_w C_{pw}} T_{amb} + \frac{Q_w \rho_w C_{pw}}{V_t \rho_w C_{pw}} T_{wo} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{15000}{V_t \rho_w C_{pw}} \end{bmatrix} \chi + \begin{bmatrix} \left(\frac{\rho_a C_{pa}}{M_{cc}C_v} Q_{a1} - \frac{U_{cc}A_{cc}}{M_{cc}C_v}\right) \\ 0 \end{bmatrix}$$

Outline

BRADLEY University

Engineering Efforts

Simulation

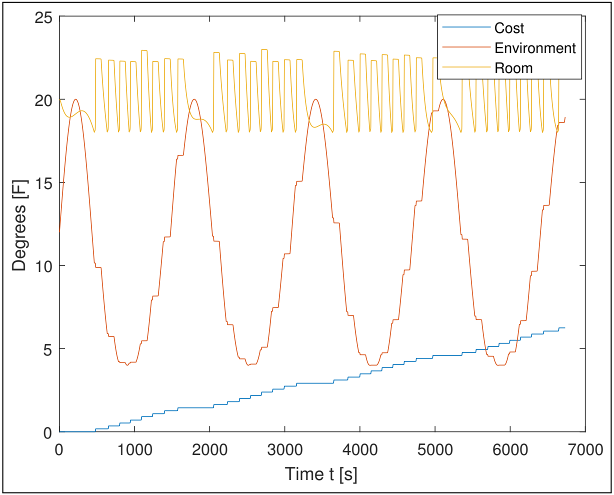


Figure: House Simulation Results using on/off control

Outline

BRADLEY University

Engineering Efforts

Design

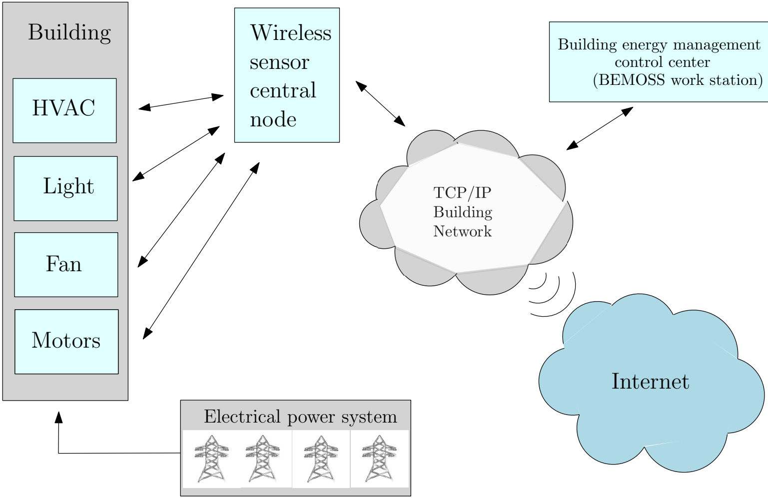


Figure: BEMOSS Network

Hardware video demonstration

Engineering Efforts

Design

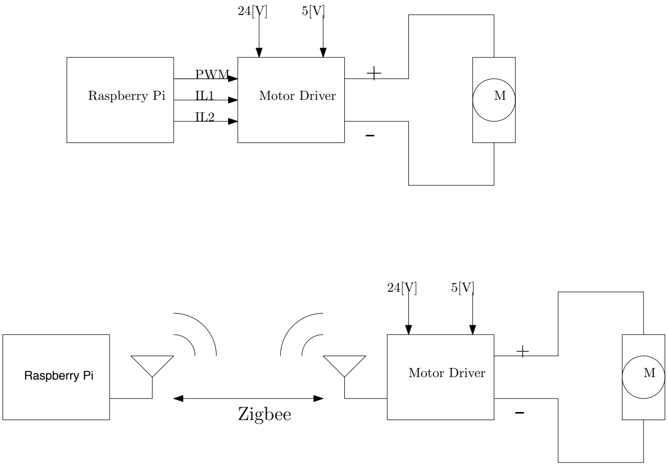


Figure: Raspberry Pi Motor Driver Interface

Outline

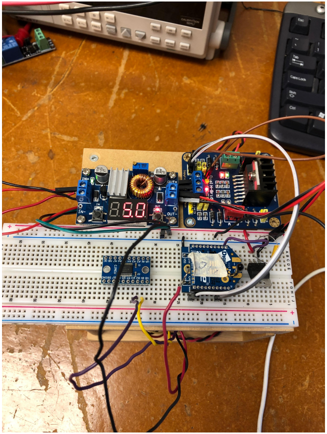


Figure: Remote Circuit 10/27/18

- XBee S2C radio module
  - Receive motor direction commands
  - Toggle H-Bridge pins
  - Relay encoder positional data to central node
- L298 H-bridge
  - Supply motor power
  - control motor rotation direction
- Pittman 24v DC motor
  - Close/ open curtains
- Optical encoder
  - Measure and relay positional data
- Buck/Boost Converter
  - Allow for a single power source to remote node
  - Bucks incoming 20V line down to 5V for motor driver, XBee module, and encoder logic

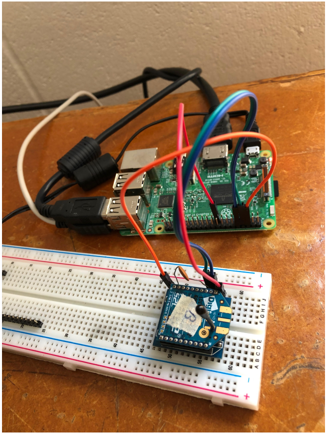


Figure: Raspberry Pi Central Node

Engineering Efforts

Experimental Activities

- Raspberry Pi Model 3B
  - Connect to BEMOSS
  - TX pin to transmit commands
  - RX pin to recieve encoder data
- Python Script:
  - Transmit remote AT commands to XBee Module
  - Process encoder data and send stop start command
- XBee S2C radio module
  - Coordinator XBee

BEMOSS video presentation

- Future work includes:
    - Utilize logic converters to allow data exchange between the 5V logic H-bridge and encoder to the 3.3v logic XBee module (Pictured in "Remote Circuit 10/27/18)
    - Employ I2C protocol in place of XBee radio communication between nodes (cheaper communication method)
    - Implement feedback control using the motor encoder to control the position and speed of the motor
    - Further energy efficiency by only powering the encoder during times of motor rotation (through use of relays)

Parts List

Part	Price	Quantity
XBee Interface Board	$5.31	1
8 Channel Logic Converter	$5.99	1
3.3 [V] Relay Board	$11.90	1
Buck/ Boost Converter	$9.99	1
WeMo Smart Plug	$ 27.90	2

Table: List of ordered parts

Outline

- Jordan
  - Develop accurate models using Simscape
  - Algorithm for BEMOSS
  - Machine learning research and implementation
- Reece
  - Interfacing a new device
  - Motor driver configuration
- Robert
  - BEMOSS
  - Implement Reece and Jordan's work

Outline

Fall Semester 2018

- Working BEMOSS - September 2018
- Transmit AT command via XCTU - October 2018
- Transmit AT command via python - November 2018
- Adaptive HVAC Control Algorithm - December 2018

Spring Semester 2019

- Interface new device - March 2019
- Implement Machine learning - April 2019
- Have device supported by BEMOSS - April 2019

Summary

- Introduce a new supported device within BEMOSS
- Configure BEMOSS to work on a single board computer like raspberry pi
- Develop an algorithm for BEMOSS to determine energy consumption and energy saved

Discussion

Questions?

For Further Reading I

P. M. Ferreira
Neural networks based predictive control for thermal comfort and energy savings in public buildings/
Energy and Buildings, 2012

V. Reppa
A Distributed Architecture for HVAC Sensor Fault Detection and Isolation.
IEEE Transactions on Control Systems Technology, 2015.

X. Zhang
Deploying IoT devices to make buildings smart: Performance evaluation and deployment experience.
2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)

BRADLEY University