Building Energy Managment Internet of Things

by

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

Electrical and Computer Engineering Department
Caterpillar College of Engineering and Technology
Bradley University

Abstract

The purpose of this project is to present a framework for controlling operations of Internet of Things (IoT) devices operating in a building environment, for instance. Motivated by the open source software architecture developed under the Department of Energy called the Building Energy Management Open Source Software (BEMOSS), the current project aims at controlling IoT devices using the proposed BEMOSS framework. The existing BEMOSS framework has the ability to automate building operations using a particular set of supported IoT devices, such as heating, ventilating, and air conditioning (HVAC) controllers, lighting, load devices, and sensors. Here, we propose to generalize this idea by controlling the operation of a new IoT device that the existing BEMOSS framework does not recognize. The new IoT device considered in this work is a brushed permanent magnet DC motor that is expected to be operated using the proposed BEMOSS framework. BEMOSS is installed on server machine running Linux operating system. A hardware interfacing circuit has been designed and implemented for the DC motor to be recognized by the BEMOSS. A prototype of the operating principle of the proposed BEMOSS framework running a new IoT device (i.e., DC motor) will be presented to demonstrate its performance. Furthermore, an HVAC control algorithm developed to be integrated into the proposed BEMOSS framework.

Acknowledgements

I would like to thank all the little people who made this possible.

Dedication

This is dedicated to the one I love.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

As smart power grids become more prevalent and data on smart building energy usage becomes more available, the opportunity for energy efficiency increases drastically. Data collected from smart buildings allows a neural network to control the workings of a building in an efficient manner. We propose to expand on the loads that can be controlled through BEMOSS by enabling control of a DC motor. Building energy management open source software (BEMOSS) is an internet of things (IoT) software that was developed at Virginia Tech. This proposed control of a DC motor through an IoT software presents the opportunity to close blinds, open barriers, and move objects throughout the building. The ability to open and close blinds building wide permits the ability to regulate the interior temperature of a building with a considerably lower power consumption. For instance, closing blinds during a hot day will naturally cool a room while opening blinds during a cold day will naturally heat the room. This added ability to control the interior environment will reduce the energy consumed by a heating ventilation air conditioning (HVAC) system.

Add networking info and controlling python scripting on BEMOSS

1.1 Background Study

Authors in [?] proposed a energy management system based around Zigbees and PLCs. Han's system would place a energy measurement and communication unit in each outlet and light in the consumer's home. The energy usage will be measured and the data collected will be sent to a home server run on a Zigbee. These Zigbees will analyze the data and give feedback to the user on how to better manage their energy usage. Renewable energy will be connected to a PLC to allow the use of renewables as they need to converted to AC. The home server on the Zigbee will predict the amount of energy that will be obtained by renewables by accessing weather data and automatically adjust the user's device schedule.

In [?] the authors once again proposed sensors in all outlets and lights to measure the energy usage in homes. Renewable energy sources, like solar panels and wind energy, are connected to an inverter and battery system, to allow the storage of excess power. Collotta connects the internet of things devices using Bluetooth rather than WiFi, due to the lower power consumption. The system checks the current time and cross checks it with the peak time for energy consumption by the user's energy provider. If it is during peak times, the system checks the battery system for excess stored energy. When the user is trying to use energy during a peak time and without a battery charge the system warns the user, but allows the user to ignore these warnings.

In [?] the energy system uses two model predictive controllers (MPC), one for the building's HVAC system and one for the system's battery power flow. The building's energy management system predicts the temperature of each room separately using sensors and a Kalman filter for robustness. The battery system put in place by Mantovani can run on two modes, one to minimize energy cost and another to minimize power flow. The building is also c-onnected to wind turbines and photo-voltaic cells and predicts the energy produced using a simplified model.

The authors in [?] propose a Internet of Energy network rather than an IoT network. An IoE combines IoT and smart grid technology using four key components: Energy Router, Storage System, renewable sources, and plug-and-play appliances. The IoE allows for an easier way to produce a net zero energy building, that produces as much or more energy than it consumes. The energy router consists of a solid-state-transformer, grid control, and communication meant for data management. The storage system like batteries, reduce the stress on the power grid and lower voltage fluctuation. Renewable sources reduce carbon emmisions, however they reduce harmonics that need to be handled with additional hardware. Lastly the plug-and-play appliances are the devices that the end-user uses in a home.

In [?] the system heavily interfaces with the users' smartphones as a way to monitor the building occupants. Since smartphones almost all have a way to track GPS, the system tracks the users' location and send it to the building's server. The building's server is broken up into a number of subservers to handle an individual room's needs. By tracking location the building can do such things like pre-heating or pre-cooling a room before the user is even in the building. All the subservers are connected to the main server which is connected to cloud storage which is used for hosting the large amount of data and handling the intense computations.

1.2 Project Statement

Our project had three main goals. The first goal was to implement a new device not currently supported by BEMOSS on BEMOSS. The second goal was to run BEMOSS on a single board computer, such as a raspberry pi. The last goal is to develop a control algorithm to reduce energy cost and implement the algorithm on BEMOSS.

1.3 Report Structure

Chapter 2

Observations

This would be a good place for some figures and tables.

Some notes on figures and photographs…

- A well-prepared PDF should be

  1. Of reasonable size, *i.e.* photos cropped and compressed.
  2. Scalable, to allow enlargment of text and drawings.

- Photos must be bit maps, and so are not scaleable by definition. TIFF and BMP are uncompressed formats, while JPEG is compressed. Most photos can be compressed without losing their illustrative value.

- Drawings that you make should be scalable vector graphics, *not* bit maps. Some scalable vector file formats are: EPS, SVG, PNG, WMF. These can all be converted into PNG or PDF, that pdflatex recognizes. Your drawing package probably can export to one of these formats directly. Otherwise, a common procedure is to print-to-file through a Postscript printer driver to create a PS file, then convert that to EPS (encapsulated PS, which has a bounding box to describe its exact size rather than a whole page). Programs such as GSView (a Ghostscript GUI) can create both EPS and PDF from PS files. Appendix **??** shows how to generate properly sized Matlab plots and save them as PDF.

- It's important to crop your photos and draw your figures to the size that you want to appear in your thesis. Scaling photos with the includegraphics command will cause loss of resolution. And scaling down drawings may cause any text annotations to become too small.

For more information on LaTeX see the uWaterloo Skills for the Academic Workplace course notes at saw.uwaterloo.ca/latex. [1]

[1] Note that while it is possible to include hyperlinks to external documents, it is not wise to do

The classic book by Leslie Lamport [?], author of LaTeX, is worth a look too, and the many available add-on packages are described by Goossens *et al.* [?]. Some on-line documentation is linked to from saw.uwaterloo.ca/latex.

Here is an example of how to include figures in LaTeX. Figure **??** shows a cantilever beam of circular cross-section subjected to a point load and a uniformly distributed load, both of which are uncertain. Note that it is better not to include the extension of the figure's source file.

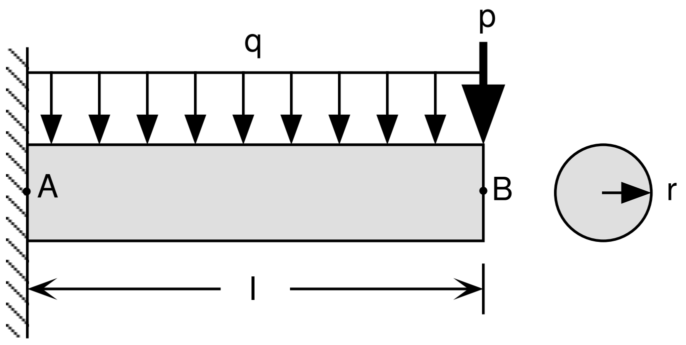

Figure 2.1: Cantilever Beam

2.1 Adding Nomenclature

The following example is part of the "nomentbl" package. Refer to the package's documentation for more details.

Let's start with equations to show how to use greek and mathematical symbols within Nomenclature.

Here is an equation

$$\dot{Q} = k \cdot A \cdot \Delta T \tag{2.1}$$

Don't forget to run:

```
makeindex -s nomentbl.ist -o uottawa-thesis.nls uottawa-thesis.nlo
```

so, since anything you can't control may change over time. It *would* be appropriate and necessary to provide external links to additional resources for a multimedia "enhanced" thesis. But also note that if the **hyperref** package is not included, as for the print-optimized option in this thesis template, any `\href` commands in your logical document are no longer defined. A work-around employed by this thesis template is to define a dummy `\href` command (which does nothing) in the preamble of the document, before the **hyperref** package is included. The dummy definition is then redifined by the **hyperref** package when it is included.

Chapter 3

Control Algorithm

To start the group wanted to create an algorithm that would monitor all devices on BEMOSS and use machine learning to optimize the consumer's behavior. To begin however we developed models for an HVAC system that could currently be controlled on BEMOSS but has no built in energy management support and a model of a motor that we developed to be controlled on BEMOSS.

3.1 System Modeling

To develop our models we used Matlab and Simulink and created block diagram models for a HVAC system as well as a motor controlled using a PWM signal and H-bridge. To create realistic models we used a Simulink library, Simscape, that has built in electromechanical blocks that we can adjust the parameters of to better model our real world system.

3.1.1 Motor

The model of the motor was based on the initial design constraints of our hardware. As can be noted the model is controlled witha PWM signal and H-bridge that can be used to run the motor in clockwise and counterclockwise directions. A current sensor is used to read the current output so that the power calculation can be done later in the model. A rotational sensor reads the rotation of the motor in rotations per minute. The DC motor is based on the Pittman 24V DC motor we are currently controlling through BEMOSS in the lab. To do this a DC motor block from simscape is used and the specifications are defined as they appear in the Pittman datasheet. This allows an accurate reading, however it should be noted that the blocks used in Simulink are ideal blocks with no losses to friction or transmission losses.

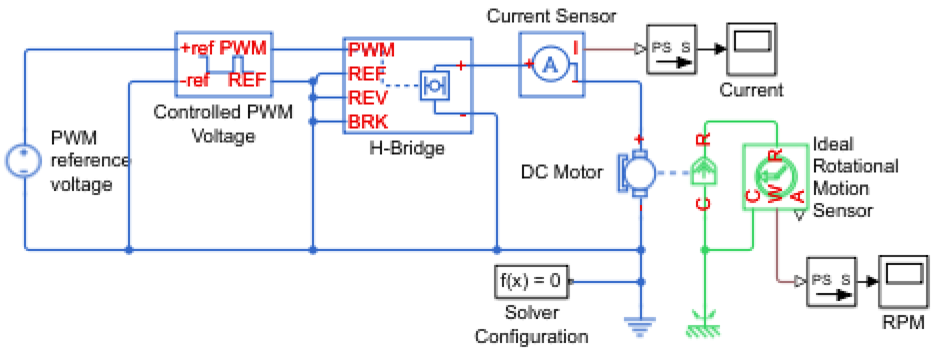

Figure 3.1: Motor Model in Simulink

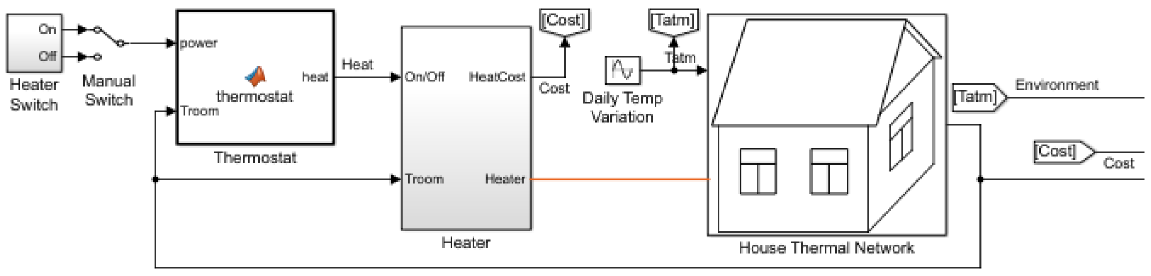

Figure 3.2: HVAC House Model

3.1.2 HVAC Model

In this HVAC House Model in Figure 3.2, the thermal properties of a one room home is modeled. An on-off heating control is used to control the temperature of the house. A thermostat block is used to determine if the current temperature in the house model is below the reference signal and using this basic check switches the heating system on or off. The house model itself exchanges heat between three components: air to window, air to wall, and air to roof. This is a simple modeling that is used to show the typical behavior of a house, even if the house itself is not a real world object. The outside temperature is a fluctuating sine wave to determine if the simple on off control will handle changes in temperature.

3.1.3 State-Space Representation of HVAC System

To better design a more realistic model the team looked at the works of [?]. The design put forth by the team in [?] is designed using a State-Space Representation. The typical control problem is defined as

$$\dot{x} = Ax + Bu \tag{3.1}$$

The system used in [?] is a conventional HVAC system used in many buildings that controls temperature and humidity. In addition the system also measures and controls CO_2.
To begin a conventional HVAC system that controls only temperature and humidity will be defined and later in the report, built upon to include CO_2 monitoring.

3.2 Control of HVAC Model

Chapter 4

Implementing BEMOSS

Our first task wen learning about

4.1 Deploying BEMOSS

We were able to run Linux fine but had issues with ssh