BRADLEY University

Laboratory Notebook

Development of an Intelligent Building Energy Management System

Brian Lauer and Elliot Watkins

blauer@mail.bradley.edu, ejwatkins@mail.bradley.edu

Beginning September 14, 2020

Contents

r.

Monday, September 14, 2020

1 Meeting Minutes

BL
Today, in the meeting with Dr. Miah, we kicked off the project involving development of a building enery management platform. A Github repository was created for the project which will also be paired with a corresponding Google Drive titled "seniorProject1-2020-21". On Tuesdays and Thursdays, we will have lab time from 8am to 11am and weekly meetings with the advisor from 4pm to 5pm on Mondays this semester. The lab times are devoted specifically for work rather than research or documentation. Before coming to lab, work must be cut out for both of us.

2 Homework

BL
This document was created and pushed to the Github repository for both of us to use. For better portability, one of the laptops previously used for research on this project was wiped clean and LUbuntu was installed. As of now, most of the code developed for the platform was done on a desktop PC dual-booted with Ubuntu Linux and Windows 10. To be able to move between lab and home easily, a more ideal situation is to use a dedicated laptop with Linux installed. Because the USB drive used to install Linux was previously formatted with Ubuntu 18.04, the diskpart utility was used with the following commands:

```
select disk 1
clean
create partition primary
format quick
```

The Startup Disk Creator GUI program on Linux was utilized to flash the LUbuntu 64-bit desktop iso file on to the USB-thumb drive. The commands `sudo apt-get update` and `sudo apt-get upgrade` were run on the laptop to perform necessary upgrades and updates. Lastly, `sudo apt-get install git` was used to install git. Another package installed was `textlive-full` to compile latex with all the necessary packages.

Monday, September 21, 2020

1 Meeting Minutes

BL
In the meeting with Dr. Miah, Dr. Miah pointed out that templates are available in the Google Drive for the lab notebook and other deliverables which we did not realize. In the System Level Requirements document (the first deliverable) we must replace the current block diagram with the drawing of the house connected to a microgrid. Inside, the home will be a block of the BEMS core which will connect to various IoT devices in the building. The controllable loads in the building including devices like lights, appliances (including dishwashers, washing machines), and smart plugs. Examples of uncontrollable loads include desktop PCs, laptops, thermistors, and microwaves. In a potential future research project, connectivity with a microgrid may be possible, so powerlines representing the main grid will be connecting the house to a microgrid through a PCC (Point of Common Coupling). In the current version of the diagram, the file is simply too large as it contains many images from the Internet. This will be redrawn in inkscape with custom figures to reduce the file size.
The final outcome of the project will be

- Laptop running in the IoT lab or office
- 2-3 devices in office/lab can be controlled succesfully
- System should recognize the devices immediately upon connection to the network
- Algorithm implemented to help improve energy efficiency

As a part of the first deliverable, three modes of operation will be declared including

- Device search mode
- Device operation mode
- Third mode to be determined

The sections of the deliverable will be

- Problem Statement
- System Architecture

- Block Diagram
- Modes of Operation

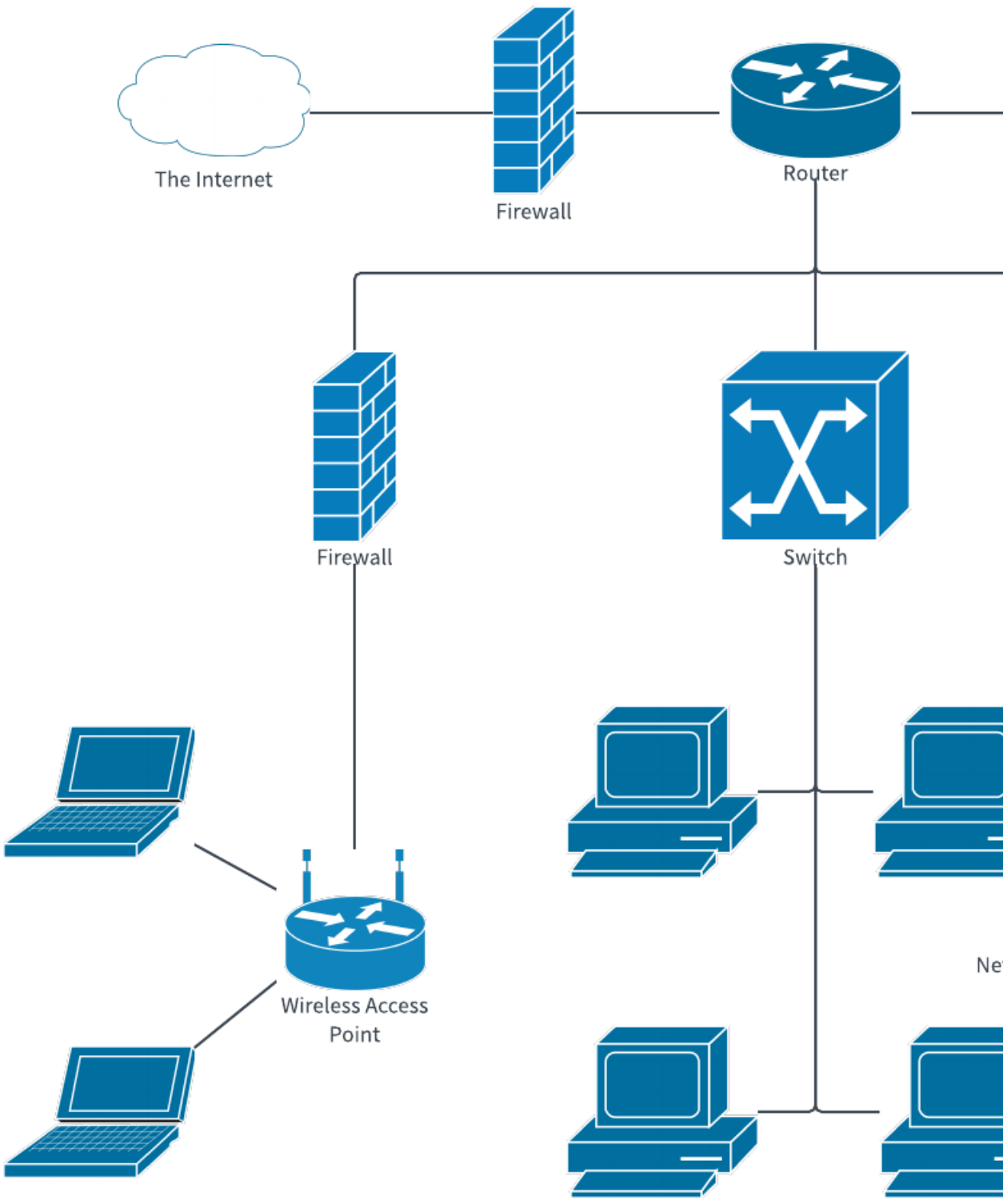
Another point brought up is whether an energy efficient algorithm should be included in the software. Dr. Miah said yes. This will possibly help to boost the complexity of the project and improve the usefuleness of the platform. This algorithm will have to be determined.

As of today, we should both be writing code on Tuesdays and Thursdays or possibly on our own, and we can both use code written for the research project. The split should be 50/50 so a Gantt chart will need to be made.

Tuesday, September 22, 2020

1 Lab work

BL
Elliot and I met over Zoom to work on getting a new Github repo setup for the senior project code specifically. Although Dr. Miah mentioned in the meeting that the code should be on Github, we simply will periodically upload the zipped Github project to the Drive to the implementation/ folder. I copied over the files from the research to the new Github repo cloned locally on my Linux laptop. After running the `install.sh` script, the installation failed as the Linux laptop I am using is running Python3.5 rather than Python3.6 which is required for f-string support. An example of a line that failed is shown in the figure below:


The Internet
Firewall
Router
Firewall
Switch
Wireless Access Point

A simple solution to this problem was simply using string concatenation:

```
strTStamp = str(tstamp.month) + "/" + str(tstamp.day) + "/" + str(tstamp.year) + " "
strTStamp += str(tstamp.hour) + ":" + str(tstamp.minute) + ":" + str(tstamp.second)
```

Further changes were made in the `ControlAgent.py` and `DiscoveryAgent.py` files. In particular, the line in `subscribe` to connect to the server acting as the central exchange to process publish/subscribe messages between the web server backend and agents listening for subscriptions was altered to support string concatenation. After these alterations to the source code were made, the platform was succesfully started. However, it is clear that no messages published to the server were received by both either the `ControlAgent` or `DiscoveryAgent` which will need to be debugged in the next lab or possilby before. This is evident as the terminals showing the debug output of each agent not indicating any messages received. Messages should be published via the publish function in the file `PROJECT_DIR/WebServer/pubsub.py`. All these problems must be fixed to allow the ActiveDevices page to discover and connect to devices.

Thursday, September 24, 2020

BL
Today, I debugged some of the problems with the `ControlAgent` agent not receiving messages. The agent was run in isolation and tested with the publish function in the `pubsub.py` file. The agent was able to successfully receive messages. However, this was not tested while the platform was running which will need to be done in the next lab session.
I (Elliot) started to create a device driver for the BeagleBone Blue based off of the WeMo Switch driver. In doing so, I also started researching ways to control individual pins via wifi, starting with the GPIO.

Monday, September 28, 2020

1 Meeting Minutes

EW

Agenda:
How to auto-detect devices on the network (interrupt or polling)? We were thinking of constantly pinging the network for supported devices in an infinite loop. Is there a way for a device to send a request to the server once it has joined the network?
We would like to add support for a smart power meter that can connect to WiFi, so power data for the entire building/house can be shown in the UI. What would you recommend in terms of a cost-effective solution? I found a device called the Sense Energy Monitor that can send data over wifi to an mobile or web app that comes with the device. The device itself connects to the electrical panel of a house. However, it is quite expensive at a price of 299 dollars.
Does the platform still need to be agent-based like BEMOSS?
Discussion: For auto-detecting, we need to research how to create an interrupt. IF that turns out to be impossible, we will try to replicate Windows trying to find bluetooth devices. It will be a poling process that is explicitly demanded by the user.
MATLAB can be used to measure power usage. Simscape is an addon that includes a power sensor module. After checking online, we also discovered it is possible to connect MATLAB to a website and transfer data. With this, we should be able to plot power usage on the UI.
FInally, Dr. Miah informed us that the project will indeed be agent-based.
Helpful Links:
https://drive.google.com/file/d/1jx7SldW3zNLdA5WBNl89LUBIK0KQuEMl/view?usp=sharing
https://www.codeguru.com/ https://smile.amazon.com/Bluetooth-Voltmeter-Multimeter-Resistance-Impedance/dp/B07PZRSYXD/

Tuesday, September 29, 2020

1 Lab work

BL
Today, work was done on getting the linux laptop connected to the "AERO" network in the senior project lab, so that experiments can be done properly with the WeMo Switch and eventually Beaglebone Blue. Because this was not working, Elliot set up a hotspot on his laptop which was tethered through BUSecure. Secondly, Elliot paired the WeMo Switch with the WiFi network in the lab with his phone as I did not remember the password to my WeMo account, so a new one was made. A couple of bugs were still present in the current version of the software. One of them was causing the `ControlAgent` to crash everytime a publish message was attempted to be sent through interprocess communication. We found the problem was due to the following error

```
can't convert type 'dict' to 'str' implicitly
```

This was fixed by using `json.dumps` in the `pubsub.publish` function. At the end of the lab session, we were able to succesfully toggle the WeMo Switch on and off through our platform. One of the things Elliot pointed out was the fact that the checkbox toggle button for turning the device on and off is not set to the proper position when the page is loaded as it is always in the off position. We decided this feature will need to be added soon as it had not been added.

Thursday, October 1, 2020

1 Lab work

BL
We decided to write down the trace of commands/function calls that occur from the top layer (UI layer) all the way down to the bottom layer (device API layer) to turn on/off the Switch:

- Main active devices page: `PROJECT_DIR/WebServer/templates/active_devices.html`
- JS code running on `active_device.js` calls `$.ajax()` which sends an AJAX request to the url '/active_devices/ajax/setDeviceStatus'
- Sending a POST request to URL '/active_devices/ajax/setDeviceStatus' calls sendDeviceStatusToControlAgent()
- In subscribe of `PROJECT_DIR/AgentPlatform/ControlAgent.py` a call to processMethod() occurs which calls setDeviceStatus()
- setDeviceStatus() calls `WeMoAPI.setState`
- `PROJECT_DIR/DeviceDrivers/WeMoAPI.py` setState() function sends an XML SOAP request over HTTP to the WeMo Switch

In order to display the correct device status on the web server via the `getDeviceStatus()` method in the `ControlAgent` class:

- `getState` will retrieve the status using XML SOAP requests
- `renderActiveDevices()` will be called in `active_devices.py`. We would like to eventually poll for a state change (Not necessarily this function). Also, we want to use the most recent timeseries data from a table.
- First we will need to use `getState()`, but eventually we want to use a data table. Also, we need to research interrupts.

I suggested installing the Cassandra database in the software as a first step towards getting the status updated on the web page as it can store time series data easily. The instructions for install Cassandra are listed below taken from [1]

[1] https://cassandra.apache.org/download/

1. `echo "deb https://downloads.apache.org/cassandra/debian 311x main" | sudo tee -a /etc/apt/sources.list.d/cassandra`
2. `curl https://downloads.apache.org/cassandra/KEYS | sudo apt-key add -`
3. `sudo apt-get update`

We encountered the following error which was documented on the site

```
GPG error: http://www.apache.org 311x InRelease:
The following signatures couldn't be verified
because the public key is not available: NO\_PUBKEY: E91...
```

The PUBKEY is extracted from the error message itself and must be added with `sudo apt-key adv -keyserver pool.sks-keyservers.net -recv-key E91..`

Lastly the software was installed with `sudo apt-get install cassandra`. To interface with the software with Python, we decided to use the DataStax Python driver for Apache Cassandra. It was installed with the following commands

1. source venv/bin/activate
2. pip3 install cassandra-driver

Monday, October 5, 2020

1 Meeting

BL
Agenda:

- What is a reliable and efficient way to obtain the status of the WeMo and update the checkbox on the web page? This is needed in case someone turns the WeMo on and off manually.
- Is a Gantt chart required for the first deliverable?
- Could you review what we have written so far for the first deliverable?
- We are considering integrating the DC motor with the software through the BeagleBone Blue. As an extension from the project in 2019 would we be able to add a speed control algorithm to use PWM to control motor speed? If so, should we develop our own or use the one from your mechatronics textbook?

2 Meeting

BL
Minutes:

- We should use an Interrupt Service Routine to get device status from Wemo Switch. This will be used to update the status being displayed on the web server.
- We will not be using a Gantt chart.
- If we want to implement the DC motor, we have to come up with a useful application. The shaft must be connected to a load. We are thinking of driving a fan which could realistically be used to control a fan possibly a CPU heatsink fan. It does not matter if we use Dr. Miah's control algorithm or our own.
- Our current issue is: the Wemo switch fails to turn on/off from the application if someone turns it on/off manually. For

Tuesday, October 6, 2020

1 Lab work

BL
Today, Elliot and I worked on ways of possibly implementing an interrupt whenever the WeMo Switch is turned on or off. Various blogs on interrupts in Python3 were found regarding KeyboardInterrupts and GPIO with the Raspberry Pi. It may not be possible to cause the Switch to generate an interrupt signal on its own as this may not be built into the firmware of the device. Initially some problems were encountered as a problem was encountered in the file `PROJECT_DIR/WebServer/static/js/apps/active_device.js` as the Javascript object named `evt` was not passed as a parameter to the callback function for the input checkbox event listener. This was then used to call

```
evt.preventDefault();
```

which according to the Jquery documentation cancels the default navigation of the click. This is not necessarily needed as it runs properly without the method call. Work was done on implementing the feature to update the checkbox position whenever the DOM is ready. One of the problems that needs to be solved is obtaining the data from the web server after sending a AJAX GET request. This will be done by receiving a JSON object in the response to the AJAX request. According the jQuery documentation[1], one of the optional arguments to be passed is the `success` parameter which can hold the arguments

- data
- status
- settings

[1] https://api.jquery.com/jquery.ajax/

Thursday, October 8, 2020

1 Lab work

BL and EW
One of the problems we faced into today's lab session was the Switch's port number incrementing by 1. Yesterday, the port number was 49154 and today it is 49155. We'll have to determine why this is happening. We are also trying to improve the getDeviceStatus() function to get both the id of the WeMo and the powerstate. A solution to this problem is using a list nested in a dictionary along with a key value pair for the id of the device. One of the feaures of javascript that we do not know to implement is how to set the checkbox position with javascript.

Monday, October 12, 2020

1 Meeting

BL
Agenda

- Go over current presentation
- Schedule an appointment for presentation
- Additional ECE faculty member for presentation

2 Meeting

EW
Minutes:

- We should put the Deliverables into a folder for Deliverables in the Docs repo.
- For the introduction to the presentation, we should have photos/animations in addition to text (Columns).
- Start with diagrams, then have text to explain more details.
- The diagram with the BeagleBone Blue - it should just be called an embedded computer. Also, no black blocks; add more color.
- Text and Diagrams should be directly related - Have 1 slide for each major part of each diagram.
- When making a presentation, keep in mind that we need to sell this to a third party who will not understand everything.
- In Figure 2, label the signals going to and from the BEMS Core.
- Start with minimum requirements, and then talk about exceeding that.

Tuesday, October 13, 2020

EW

1 Lab work

Brian and I tried to get the refresh button on the "active devices" page to reload the check box for the WeMo switch. This task is proving to be more challenging than we thought. Neither of us has much experience with JavaScript. We need to learn how to access each button and their properties in active_device.js if we want this feature to work.

Thursday, October 15, 2020

EW

1 Lab work

Today, the Aero router was missing from the lab. We instead used my Windows laptop as a hotspot to connect to the Wemo. Once we got connected, we continued with the task of updating the check box on the active devices page with the current Wemo status. At the beginning of this session, refreshing the check box would turn the Wemo on or off instead of just changing the check box. By 9:22 AM, we were able to refresh the status of the check box on the active devices page with the actual state of the Wemo switch. I took a short video of our progress with the Wemo Switch; that needs to be uploaded to the Deliverables folder of the Docs repo. Next, we decided to start implementing the Apache Cassandra database. This will lead us into the next feature for the Wemo: recording power usage. Our plan:

- Set up the database
  - The Cassandra database needs to be set up outside of python, then it can be accessed
  - We believe we can automate the set up process with a shell script that can be run from python
- Implement a function to write data to it
- Implement a function to read data back from the database
- Implement a function to display data on the active devices page
- Figure out a way to plot a graph of power usage with matplotlib in python

Monday, October 19, 2020

1 Meeting minutes

BL
Agenda:

- System level requirements pres
- Changes to be made to the sys level requirements draft
- Should Elliot and I work together on the same features (pair programming) or work separately on different tasks?
- Should we have a well defined schedule of what features we would like to work for the rest of senior project?

Minutes:

- We should list all the features found in BEMOSS or another BEMS platforms and explain how ours is different. We need to gather some standard features that people have used in similar projects. Find at least 5 other platforms. Use library resources and contact Dr. Miah if we can't get access to a source.
- I (Elliot) will start implementing support for the beaglebone blue and Brian will work on implementing additional features (right now that is the Apache Cassandra Database and power recording).

Tuesday, October 20, 2020

1 Lab work

EW
Today, I started looking into running python files on the BeagleBone Blue. I installed the Adafruit library: https://pypi.org/project/Adafruit-BBIO/ and was able to run a simple program to flash the on-board LEDs. However, Dr. Miah mentioned that I should look into the rcpy library as an alternative. After settling on a control library, I need to figure out how to connect to the BeagleBone Blue from the platform. Dr. Miah also reminded us both to start looking into standard features found in other energy management systems similar to BEMS.

BL
I worked on adding calls to the various cassandra library methods and class constructors to interact with the Cassandra database. In the Discovery agent, I added a query to create a new table for each device when it is discovered by the discovery agent in the `setDeviceToActive` method. Each table will have fields

- id text
- power double
- status text
- tstamp timestamp

for now. However, this will likely need to be changed when new devices are added. Secondly, another query execution call was added to the method `periodicQueryBehavior` in the `ControlAgent` which will add a new entry to a given device table every 10 seconds inside a separate thread from the main program. In order to initialize the database, a keyspace was created titled "bemstimeseries" which will need to be added to the database when the platform is installed. This likely be added to the `install.sh` script. The last task I worked on was installing matplotlib which gave me trouble as python3.5 is installed on the LUbuntu laptop running the server. However, this python package requires Python3.6 and later. I will need to work on figuring out how to update python on the laptop if possible.

Thursday, October 22, 2020

1 Lab work

BL
I decided to try just installing matplotlib and all the necessary dependencies like numpy with older versions supported by python3.5. The following dependencies and their corresponding versions are required for python3.5

- matplotlib==1.5.1
- numpy==1.11.0
- dateutil==2.4.2
- pytz==2014.10

Another problem was encountered when attempting to install the dateutil python3 package as a messaged was displayed that python3.5 reached the end of its life on September 13, 2020. A solution to this problem I will try to implement is installing the latest version of Lubuntu 20.04 LTS. Dr. Miah approved this.

The next task is to determine where to create a new keyspace in the cassandra table. I added this in the `run.sh` file for now but may make more sense to place in the `install.sh` file. Next I worked on inspecting the keyspace for new tables when a new device is discovered. Unfortunately, no new tables were showing up after the `DiscoveryAgent` did its work in the `setDeviceToActive` method. The CQL query:

```
SELECT * FROM system_schema.tables WHERE keyspace_name = 'bemstimeseries';
```

should list all the table information in the 'bemstimeseries' keyspace. The next logical step is to run the query in the `setDeviceToActive` method which will create a table for the device. In the `cqlsh` command line program, I ran into the exception

```
InvalidRequest: Error from server: code=2200 [Invalid query]
message="No PRIMARY KEY specifed (exactly one required)"
```

This error makes sense as no primary key was supplied in the CREATE TABLE command. A second error was encountered after adding in a primary key:

```
SyntaxException: line 1:59 no viable alternative at input
'PRIMARY_KEY' (... EXISTS bemstimeseries.device1 (id [UUID] PRIMARY_KEY...)
```

After executing the query

```
CREATE TABLE IF NOT EXISTS bemstimeseries.wemo
(id UUID PRIMARY_KEY, power double, status text, tstamp timestamp);
```

EW
I have made some progress on the Beaglebone API. It is possible to connect to the beaglebone via python socket programming and send commands. This will be the basis for controlling the beaglebone from the platform. At the end of lab today, I was having trouble running the program to turn on the on-board LEDs. I need to figure out why that is not working and then add more functions to better control the LEDs. Eventually, this will be integrated into the platform and the discovery/control agents will be used to access and control beaglebones. We will also need wires to connect to the motor drivers.

Monday, October 26, 2020

1 Meeting

EW
Agenda

- Links to IEEE articles on BEMS we don't have access to:
  - https://ieeexplore-ieee-org.ezproxy.bradley.edu/document/6852647
  - https://ieeexplore-ieee-org.ezproxy.bradley.edu/document/8494012
  - https://ieeexplore.ieee.org/document/8246800

  Other sources with features for BEMS:
  - https://greencoast.org/building-energy-management-systems/
- Go over progress on BeagleBone Blue API (Elliot)

BL
Other links

- https://ieeexplore.ieee.org/document/8403212
- https://www.mdpi.com/2071-1050/9/2/264/pdf-vor
- https://www.mdpi.com/2079-9292/8/7/763 - LoBEMS - LoRa Building and Energy Management System

From third link (LoRa BEMS), some of the features added to their platform are

- LoRa support
- A/C automation
- Plot of A/C usage
- air quality monitoring
- If This Then That support - BEMS was tested at a school, parents were notified when the air quality was too low

- Sensors with SOC's, powered by batteries, using the LoRa communication interface monitored temperature, humidity, luminosity, and air quality

Minutes

- Beaglebone API connection is valid. Need to continue building functionality.
- Cassandra Database is a valid option for storing data. We will continue working with that. Brian should try Setting the primary key to "device ID."
- The final paper that Brian found is an excellent source to reference, we will be reading this in the near future.
- Dr. Miah will look into the papers that Elliot found to see if they are good sources to use.

Thursday, October 22, 2020

1 Lab work

EW
Today, I made some more progress with the socket connection to the BBB (beaglebone blue). I wrote a couple a functions in a file called LED.py to control the LEDs. Then, I ran a socket server on my laptop and a client on the BBB that could call those functions in LED.py. This was a major milestone because it proved that programs can be run on the BBB from a laptop connected over wifi. My next task will be altering the server/client relationship to be more robust. Right now, it has to go back and forth between the server and client, I would like to be able to send multiple messages to the client and vice versa.

Thursday, October 29, 2020

BL
Work was continued on getting the data logging feature implemented for the WeMo Switch. I came across this problem:

```
Traceback (most recent call last):
  File "/usr/lib/python3.8/threading.py", line 932, in _bootstrap_inner
    self.run()
  File "/usr/lib/python3.8/threading.py", line 870, in run
    self._target(*self._args, **self._kwargs)
TypeError: periodicQueryBehavior() argument after * must be an iterable, not int
```

I decided to try and run the ControlAgent to debug the problem with an int getting sent to the `periodicQueryBehavior` rather than an iterable object like a list or string.

Further problems were encountered when multiple WeMo switches were appearing on the active devices page. Some of the problems were still being reported in the terminals. The next step is to determine how to properly add a timestamp to a database entry. Some of the problems were fixed with adding entries to the device table for the Switch. However, when adding a timestamp, the following was added by using the cql call `toTimeStamp(now())`

```
2020-10-29 16:42:28.166000+0000
```

The current time in the lab was 11:42 am meaning the timezone must be configured. Research will have to be done on this topic. First, a problem with the current columns in the table is the presence of an id column which is actually defined in the name of the table. The id column will be removed. However, a primary key is required. Originally this was set as the id column but will be changed to the timestamp column.
EW
Continued working on the server and client files for connecting to the BBB. I created 2 new files called new_server.py and new_client.py. new_server.py uses object-oriented programming to create a server and then run specific functions. The goal here is to produce several functions for controlling the BBB that the BEMS core will be able to call instead of having the user manually type commands in a terminal.

November 2, 2020

1 Meeting

BL
Agenda

- May we use a process control algorithm like Model Predictive Control or Linear Quadratic Regulator to simulate control of building temperature? I was thinking we could have some sort of HVAC model that is able to communicate with our BEMS. We could try to predict the appropriate amount of heat to apply to the building depending on the outside temperature. I found an article listed in our proposal regarding this method.
- If we should use MPC, could you possibly explain Model Predictive Control?
- If we should use LQR, could you possibly explain LQR?
- Should we plan on adding any microgrid algorithm or connectivity to our proposal in the tasks/planning section with the Gantt chart?
- Should we come up with our own ideas for this microgrid connectivity or should we come up with our own?
- Go over current proposal

EW
Minutes

- We need to read Dr. Miah's presentation on LQR algorithm. LQR Documentation and implementation will be on Google Drive. We will not be implementing MPC at all.
- Project Implementation will suspend on November 17th 2020. After that we will focus on the presentation and proposal.
- Right now we need to focus on BBB API and Data Logging.
- We need to finish setting up the platform before Spring Semester.
- In the Spring Semester, we will add features (that are not totally necessary, but add the the value of project) like simscape models for microgrid and HVAC systems.

November 3, 2020

1 Lab work

BL
I determined how to set the timezone of a CQL timestamp. It uses the RFC 822 4-digit time zone format (+/- HHmm) which is the offset from GMT (Greenwich Mean Time). When testing with the `cqlsh` command line program, I found that after adding an entry with an offset -0600 for the first time, the timestamps were each set to CST (Illinois time). So after a timestamp is added once with -0600, the further timestamps can be added with -0000. The data was properly added to the device[] table which corresponds to the device id. Another feature that needs to be added is adding seconds data into each timestamp. The strftime method is used to format the time in the form year-month-day hours:minutes:seconds. Data is succesfully being pushed to the database every 20 seconds as configured in the software agent. A simple command to clear out the table is

```
TRUNCATE bemstimeseries.device1;
```

for instance for the table for device with id 1. There is a bug in the current version that causes three rows to be saved to the table rather than 1 intended row every 20 seconds. I believe the source of the problem is the pubsub.publish function that publishes three messages rather than 1 to ensure the message is sent.
EW
I started looking into using a multicast signal to control the BBB. I wrote a receiver for the BBB and a server program for my laptop, but haven't gotten it to work yet. Debugging will continue on Thursday.

November 3, 2020

1 Lab work

EW
Today I helped Brian access the data in csv file he downloaded from the database. We were able to use pandas to access the data and plot a graph of the WeMo power usage vs. time using MatPlotLib. Our plan is to demonstrate this feature by just collecting data on a per minute basis as opposed to a per day basis (we don't have that much time for testing). This will still be proof that data can be plotted for any length of time the user may want. Brian also helped me finish debugging the multicast issues I have been having. At this point, it is only my Windows laptop that is unable to send a multicast signal to the BBB. Brian achieved this with both the Linux laptop and his own Windows laptop. The next step will be to send a signal from the BEMS core instead of running specific functions in a terminal.

November 9, 2020

1 Meeting Minutes

BL
Agenda

- Go over requirements for proposal presentation
- Go over current proposal draft

EW
Minutes

- We need a more detailed version of the system architecture figure. This should go under the Experimental Activities (new subsection called Experimental Setup. This will include the laptop as the BEMS Core, the router, and the devices. Arrows should be utilized with comments to explain the connection. Overall, the final document should show our current work on the project.
- The presentation draft does not have to be finalized. We do not have to have the setup yet, we just need to copy stuff over from the proposal draft and the previous presentation
- We will take a video on Thursday of our current work. Thursday's work will take place from 8-11:00 AM
- A question came up as to whether or not to go into technical detail in the proposal. We can add some code and explain what it does. It should be more high-level than technical though. We should stick with inputs and outputs of functions.

Tuesday, November 10, 2020

1 Lab work

EW
I got the BeagleboneAPI.py file up and running. It has the same functionality as the test files did (I removed those from the repo). The next step is to alter the discovery agent so that it can run the BeaglboneAPI.py. After that, we should be able to create a button in the Web Server to control the LEDs from the BEMS Core.

Monday, November 16, 2020

1 Lab work

BL
I worked on fixing one of the issues stemming from the HTML plot generation. One of the issues is the x and y values have different lengths which cause issues when plotting. This problem was fixed by constructing a tuple from the power level associated with a timestamp and the corresponding timestamp object. A problem I am currently facing is the xticks not rotating to 90 degrees although specified in the `matplotlib.pyplot.xticks` call. I would like to create a drop down menu that will allow the plots for a specific day to be selected.

2 Meeting Minutes

Agenda

- Schedule for proposal presentation practice
- Should we discuss previous work from IEEE papers in the presentation?
- Progress on Beaglebone
- Progress on power plotting
- Should the power and on/off status plots be placed on the home page or active devices page?
- How many references should we have in the proposal?

Minutes

- We will have a practice session for the presentation on Monday, November 30th (after Thanksgiving Break). It will 10 minutes long with 5 minutes for Q/A.
- We should add to Background Study section using the IEEE papers that Dr. Miah provided in the references folder and BEMOSS-Literature. We should use 4-5 exiting BEMS examples in the proposal and only explain 1-2 in the presentation (use pictures for visual aid).

- We need to push the final draft of the presentation Latex code by the end of the day on Wednesday, November 25. Dr. Miah will review and make necessary changes at this point.
- In the proposal, we should write only "empbedded computer" when referring to the Beagle Bone. However, in the presentation, we should write "embedded computer (beagle bone blue). Also, we should say "embedded computer" or "single board computer."
- We should get a lot of screenshots and put them in the presetation and proposal. Everything should be explained in the proposal, but we should only talk about screenshots in the presentation.
- Home page should be decorative and attractive to customers. It should be all visual aids and high-level explanation. The power plots and other technical data should go on the active devices page.
- We will use existing motors in the department rather than buying a new one. However, we will not be using the Pittman motors, we will instead use a more expensive, better motor.
- If time allows, Brian will start working on user login. I will offer as much help as possible in this so as not to offset the workload.
- We will also 2 more videos. One of the power plot being generated and one of the beagle bone in action. These videos will be part of the presentation.

Tuesday, November 17, 2020

1 Lab work

EW
Today, Brian and I worked on adding sources to the Background Study Section I added information on this article: https://ieeexplore.ieee.org/document/8246800 I also made plans to update the System Architecture diagram according to the Nov. 9th meeting minutes.

Monday, November 30, 2020

1 Lab work

BL
Agenda:

- Practice presentation
- Go over presentation/make suggestions or changes
- What day are we to present?
- Are we supposed to work on lab work Tuesday and Thursday or just the proposal?

Monday, December 7, 2020

1 Meeting minutes

BL
Agenda:

- Go over proposal again potentially
- Go over further presentation comments
- Should we still be meeting in the lab tomorrow?
- I found an alternative to the matplotlib plot library for generating interactive plots on webpages called chart.js. Could we keep this open as an option even though we stated in the proposal that we will be using Matplotlib?
- Would it be more ideal to use an email for login or simply a custom user name?
- What is the benefit of a using a PID controller (or any type of controller) to regulate the speed of a motor? Is it absolutely necessary for speed control or can proper control be done without it.
- Will we still be meeting over Christmas break?

======= EW
Minutes:

- We are meeting tomorrow in the lab for the last time this semester.
- There is an alternative to using Matplotlib for plotting power usage on the active devices page.
- The user login feature can use either a person's email for username or a custom username
- We will use a speed control algorithm. There are existing simulink files on the Google Drive that we will use for control algorithms. The input will be the user-defined duty cycle and the algorithm will maintain a constant speed. An encoder will be used to measure the speed and send this information back to the algorithm. A variable can be simulated with Simscape.

- We will meet sporadically and unofficially over Winter Break. Dr. Miah will primarily be focused on his research project.

Thursday, January 14, 2021

1 Meeting minutes

EW
Agenda:

- Go over my (Elliot) work over break
- Go over Brian's work over break

======= Minutes:

- I have been trying to find a python library that will allow me to use the motor drivers on the beaglebone blue. I want it to be in python because that is what we are already using for communicating with the core. Brian found this: https://guitar.ucsd.edu/rcpy/rcpy.pdf which is much more promising that anything I have found. If it works, I will use it for controlling the LEDs and anything else on the beaglebone as well.
- Brian has been looking into ways to embed the power plots in the active devices web page. He found another library that makes it very convenient to achieve this (I forgot what the name of the library is).

Tuesday, January 19, 2021

1 Meeting minutes

EW
Agenda:

- Go over my (Elliot) since last meeting
- Go over Brian's since last meeting

======= Minutes:

- Brian's suggestion worked. I had some trouble installing rcpy because their instructions for installing a couple of dependencies were outdated. However, I managed to find a way to install all the necessary software to run rcpy. My next step is to write a python file with several functions for running the motors and LEDs.
- Brian needs to write a function to separate the columns in the power data (date, time, power usage) and also find all entries for a specific date.

At this point, we have 1 week left before classes begin again. Without homework it is possible to finish our current tasks, but we need to stay focused.

Saturday, January 23, 2021

1 Meeting minutes

EW
Agenda:

- Go over my (Elliot) since last meeting
- Go over Brian's since last meeting

======= Minutes:

- I re-imaged the beaglebone and reinstalled rcpy with all its dependencies. It is now possible to start and stop the motors very easily without the beaglebone crashing (had some trouble with that earlier). I also updated README.md with instructions on how to install rcpy on a beaglebone.
- Brian has implemented the plot feature almost entirely. There is one last glitch that needs to be worked out. When hovering the cursor over the graph, the title of the graph changes, which is not desirable.
- We started working on creating a new button for inputting the PWM for the beaglebone motor drivers. This will have to be finished up Monday and Tuesday before school resumes on Wednesday.

Thursday, January 28, 2021

1 Meeting minutes

- Weekly meeting time - Thursday 3:00 pm - 3:30 pm
- Brian went over his progress
- On power plots - find a way to get the plot to update automatically
- We need to add scheduling - duty cycle for motor, power on or off for switch
- Elliot went over his progress
- Power meter device

Plan:

- Work on polishing these two devices (Switch and motor) for now
- Dr. Miah will work on the more research heavy side of the project
- Microgrid and HVAC with MATLAB and Simscape with simulation only
- Digital thermometer as a potential option later after the two devices

2 Lab work

BL
The problem I need to solve is how to compute the total energy consumption of the device plugged into the WeMo switch over a given day given:

$$E = \int_0^{t_f} p(t)dt$$

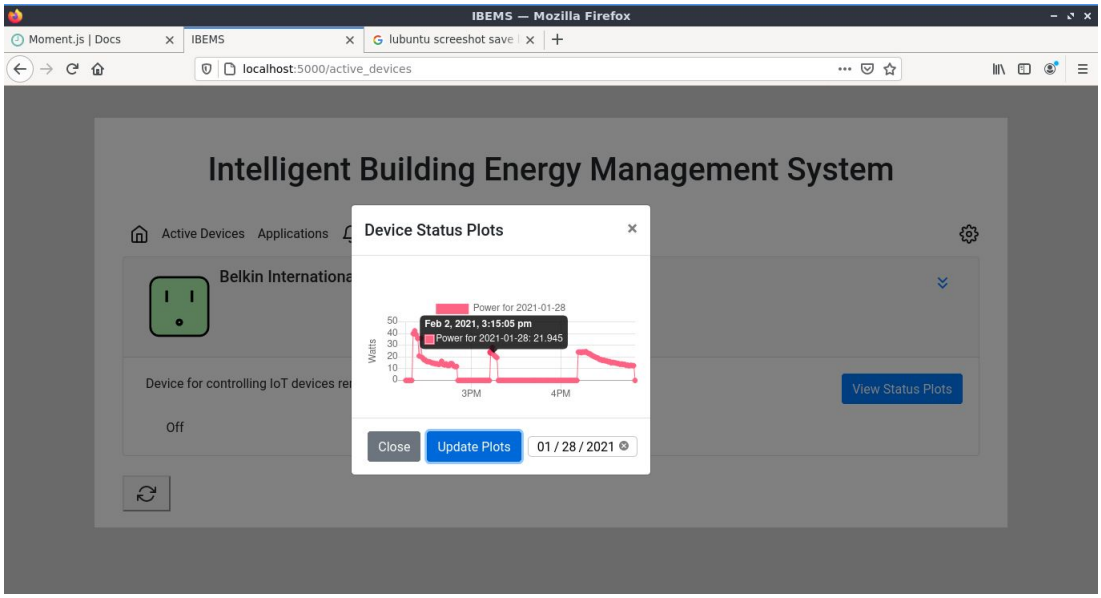
In discrete form:

$$E = \sum_{t=0}^{t_f} p(t)\Delta t$$

Tuesday, February 2, 2021

1 Lab work

BL
I decided to take a break from working on the energy display to work on fixing a bug regarding how the Moment.js objects are constructed to allow proper display of the dates in the power plots of each device [1]. When hovering over the plot values in the chart js plot the dates are incorrect as shown as the current date selected is February 2, 2021 and the date selected in the plot is January 28, 2021:



We solved the issue of the dates being incorrect by updating the javscript code. Next, we are working on calculating the total energy usage over a given day using trapezoidal rule for integration and displaying this value on the webpage. We succesfully developed the algorithm to sum up and calculate the total energy for a given day.

[1] https://momentjs.com/docs/