CSYE 7374

Connected Devices

Semester Project

Smart Home Lighting System

(SHoLS)

Project Report

Andrews Tito

Table of Contents

Chapter 1: Project Overview

1.1 Introduction

Home automation is "The Internet of Things" - *an ecosystem of embedded computing devices that have ambient knowledge of their surroundings (via internal or external sensors) and can communicate with each other and remote services via the Internet* [1]. Also, in other words, the way that all our devices and appliances will be networked together to provide us with a seamless control over all aspects of our home and more. Home automation has been around for many years, and only recently has technology caught up for the idea of the interconnected world, allowing full control of your home from anywhere, to become a reality. With home automation, you dictate how a device should react, when it should react, and why it should react. You set the schedule and the rest is automated and based off your personal preferences thus providing convenience, control, money savings, and an overall smarter home. Home automation can also alert you to events that you might want to know about right-away while you are gone like water leaks and unexpected access to your home, or any part of it. At any time, you can grab your iPhone, Android device or other remote control and change the settings in your house as desired [2].

1.2 Project Category

Area: Home Automation

Team Members: Andrews Tito, Yogesh Suresh, Aneesh Kulkarni

1.3 Problem Statement

In a modern household, typically, conventional wall switches controls lighting in different parts of a house which necessitates manual operation of switching on or off these switches. Over time, it gets virtually impossible to keep track of lights that are running and to monitor their usage. These lights when left on without any use, increases the energy usage which in turn leads to higher electricity bills. In order to tackle the wastage of this energy and assist people with easy management of switching operation, an automated home solution is necessary.

1.4 Solution

A simple automated light system is designed and developed that controls switching mechanism of light fixture devices at home automatically. This is achieved by incorporating usage of a motion sensor and light sensor (team integration). The recorded motion sensor data and light sensor data is collectively used to produce the desired actuation of switching on/off the lights. Based on the natural light level and presence of a human in the room, decisions are made by the prototype automation system to actuate the switching operation. Furthermore, to track the overall usage of lights in the house over time and understand the usage pattern and perform various analytics on data, collected sensor data is saved to a secure cloud on real time basis.

1.5 Requirements

Software:

- IDLE: IDE for Python programming (CoAP – Server & Client, MQTT – Client)
- IBM Bluemix©: for recording sensor/machine generated values and storage of values for further data analytics and operations
- CoAPthon[5]: a python library to the CoAP protocol compliant with the RFC
- Paho[4]: library that implements a client class that can be used to add MQTT support to Python program

*** For future implementation***

- *Raspbian OS – Linux based OS for Raspberry Pi*
- *RPI-GPIO library – GPIO interface library for Raspberry Pi*

Hardware:

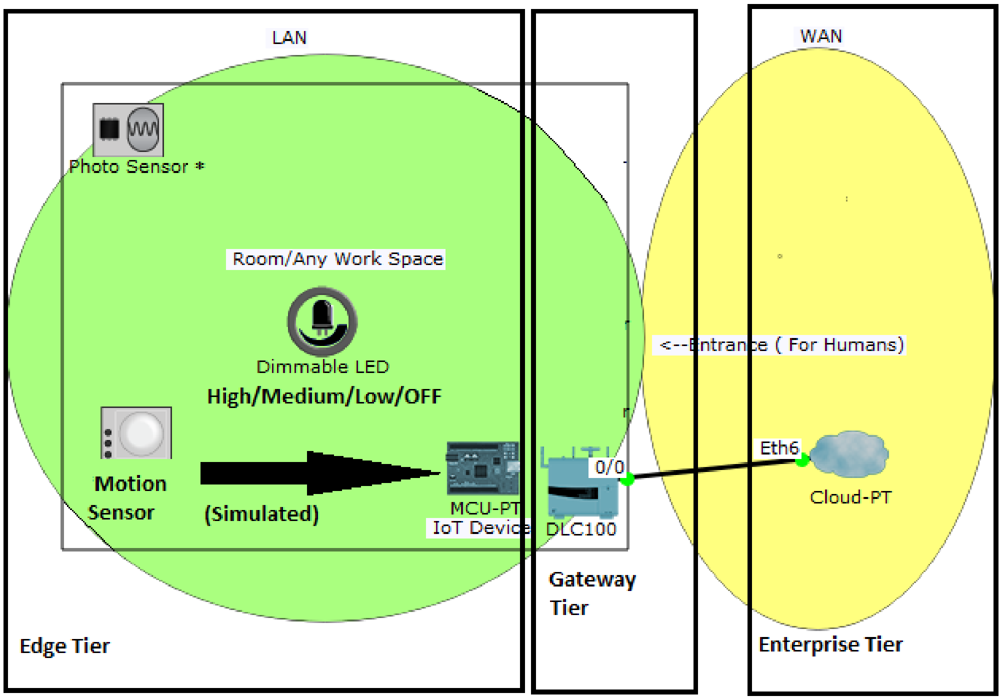
- Windows / Linux based machine

*** For future implementation***

- *Raspberry Pi 3B* **-** *a low cost, credit-card sized computer, which uses a standard keyboard and mouse*
- *Adafruit TSL2561 luminosity sensor[P1] - an advanced digital light sensor allowing for exact lux calculations*
- *PIR Motion Sensor[P2]* **-** *a simple to use motion sensor*
- *Power Monitor[P3] – a energy sensor to track and store energy consumption levels*

Chapter 2: Methodology

2.1 Design Overview



** denotes sensor of team member (integration)*

Fig 1: Gateway-Mediated Edge Connectivity and Management Pattern[6]

- Topology: Hub & Spoke
- Edge Tier: Actuation (MQTT Subscriber) & Sensing (CoAP Client)
- Gateway Tier: CoAP server & MQTT publisher (Cloud, Actuator)
- Enterprise Tier: IBM Bluemix Cloud service

2.2 Use Case

Use Cases are based on full integration with SLS project because with motion sensor data alone, project implementation cannot achieve fully efficient solution.

Trigger	Action	Effect
Human Presence	1. Record Motion sensor value 1 (default action)	1. Update cloud with motion sensor value
	2. Accept payload (lumens value) via subscribe "ACT" (default action)	2. If lumens value > 3 (high intensity natural light), turn OFF light else turn ON light
	3. Check lumens value	
No Human Presence	1. Record Motion sensor value 0 (default action)	1. Update cloud with motion sensor value
	2. Accept payload (lumens value) via subscribe "ACT" (default action)	2. turn OFF light

2.3 Detailed Workflow

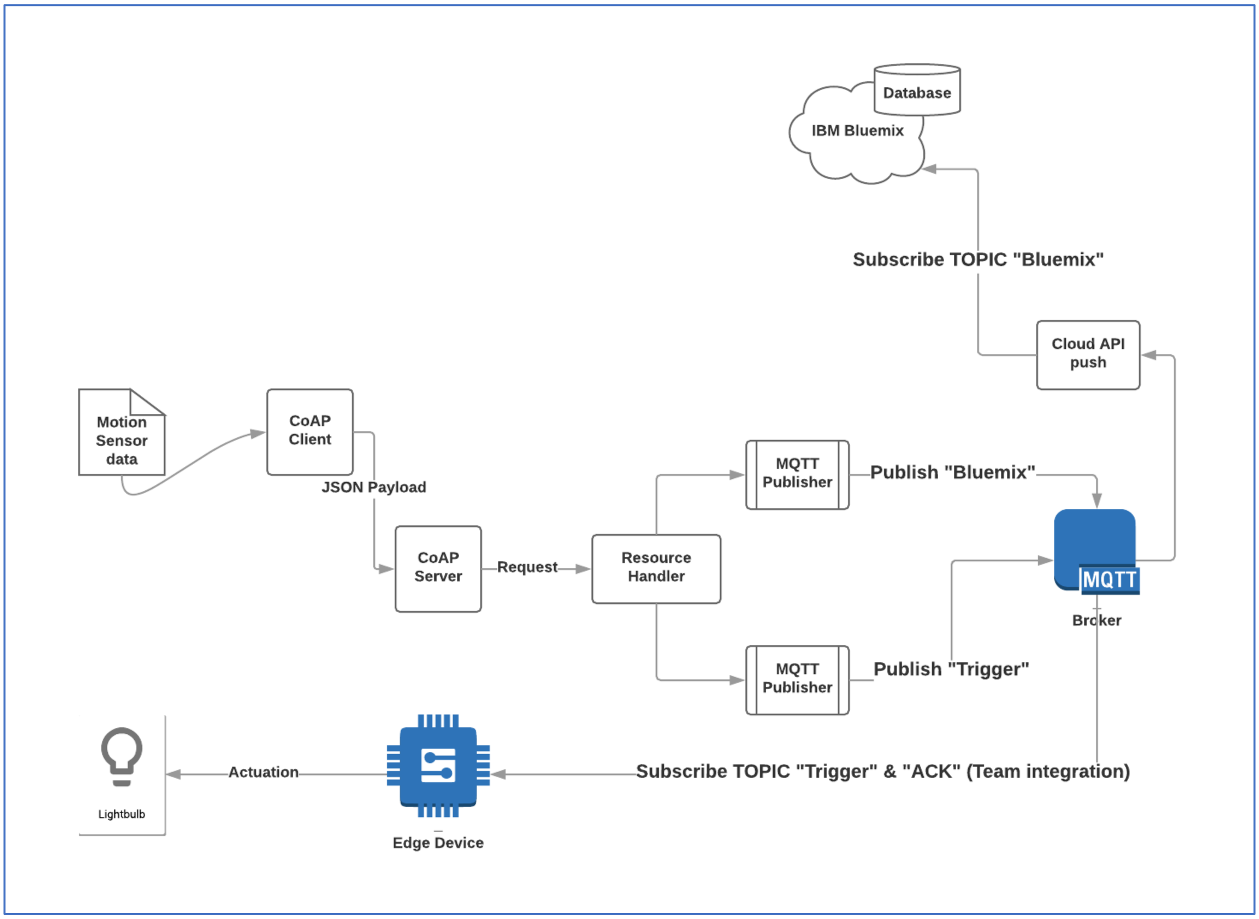


Fig 2: Detailed workflow

Chapter 3: Results and Conclusion

3.1 Results

Output:

1. **Edge device**

```
Connected with result code 0
Presence data from (SHoLS):  1
Turn ON the light
Presence data from (SHoLS):  0
Turn OFF the light
Presence data from (SHoLS):  0
Turn OFF the light
Presence data from (SHoLS):  1
Turn ON the light
Presence data from (SHoLS):  1
Turn ON the light
```

2. **CoAP Server**

```
2018-04-20 11:28:08,077 - MainThread - CoapServer - INFO - CoapServer Resource initiation
2018-04-20 11:28:08,078 - MainThread - CloudResource - INFO - CloudCoapResource initiation successful
2018-04-20 11:28:08,079 - MainThread - ActResource - INFO - ActCoapResource initiation successful
CoAP server initialized. Binding: 127.0.0.1:5683
['/', '/Bluemix', '/Trigger']
```

3. **Cloud (IBM Bluemix)**

21ed3a40-448e-11e8-81...	b'{"Group Name": "Home ...	MS111	MotionSensor	Presence data
23131ca0-448e-11e8-93...	b'{"Group Name": "Home ...	MS111	MotionSensor	Presence data
24871f00-448e-11e8-a4...	b'{"Group Name": "Home ...	MS111	MotionSensor	Presence data
26049871-44a8-11e8-b...	b'{"Group Name": "Home ...	MS111	MotionSensor	Presence data
26103000-448e-11e8-8...	b'{"Group Name": "Home ...	MS111	MotionSensor	Presence data

```
OK. Connected with result code 0
OK. New data received
OK. Data sent to Cloud
OK. New data received
OK. Data sent to Cloud
OK. New data received
OK. Data sent to Cloud
OK. New data received
OK. Data sent to Cloud
OK. New data received
OK. Data sent to Cloud
```

4. CoAP Client

```
Parsing URL: coap://localhost:5683/test
Executing all CoAP comm tests...
Testing discover...
Created CoAP client ref: <coapthon.client.helperclient.HelperClient object at 0x05223BD0>
 coap://127.0.0.1:5683/test
2018-04-20 11:39:06,013 - MainThread - coapthon.layers.messagelayer - DEBUG - send_request - Fro
nown, Uri-Path: core, ] No payload
2018-04-20 11:39:06,020 - MainThread - coapthon.client.coap - DEBUG - send_datagram - From None,
ri-Path: core, ] No payload
2018-04-20 11:39:06,031 - Thread-1   - coapthon.client.coap - DEBUG - Start receiver Thread
2018-04-20 11:39:06,034 - MainThread-Retry-42994 - coapthon.client.coap - DEBUG - retransmit loo
2018-04-20 11:39:06,042 - Thread-1   - coapthon.client.coap - DEBUG - receive_datagram - From ('
] </Bluemix>;ct=0;if="...76 bytes
2018-04-20 11:39:06,058 - Thread-1   - coapthon.layers.messagelayer - DEBUG - receive_response -
ype: 40, ] </Bluemix>;ct=0;if="...76 bytes
2018-04-20 11:39:06,064 - Thread-1   - coapthon.client.coap - DEBUG - Waiting for retransmit thr
2018-04-20 11:39:06,066 - MainThread-Retry-42994 - coapthon.client.coap - DEBUG - retransmit loo
Source: ('127.0.0.1', 5683)
Destination: None
Type: ACK
MID: 42994
Code: CONTENT
Token: None
Content-Type: 40
Payload:
</Bluemix>;ct=0;if="if1";obs;rt="rt1",</Trigger>;ct=0;if="if1";obs;rt="rt1",

 Path entry [0]:Bluemix
 Path entry [1]:Trigger
 Path entry [2]:
```

3.2 Integrations

```
Connected with result code 0
Presence data from (SHoLS):  1
Turn ON the light
Presence data from (SHoLS):  0
Turn OFF the light
Presence data from (SHoLS):  0
Turn OFF the light
Presence data from (SHoLS):  1
Turn ON the light
Presence data from (SHoLS):  1
Turn ON the light
Lumen value from (SLS): 5
Turn ON the light
Lumen value from (SLS): 1
Turn ON the light
```

3.3 Message format

Payload type: JSON StringByte

Payload size: 128 bytes

Payload format:

```
{ 'Group Name':'Home Automation',
'Name':'SHoLS',
 'Location':'Home',
'TimeStamp':<time>,
'SensorData':<sensorVal>
}
```

3.4 Problems faced

This project, by no means, completed without any problems and difficulties along the way. Following are few problems we faced during the completion of this project:

- Due to the diverse nature of the project, the hardware components (sensors) matching our requirements were difficult to source.
- As it was our first take on protocol intensive software and hardware design, we faced complexities and overall process was a challenging one.
- On the software part, CoAP handler development produced some critical scenarios where it took some handsome amount of effort.

3.5 Limitations

This project is our take on trying to create a prototype of a fully working Smart Home Light System. Although every effort was made to make it closer to an ideal solution that we wanted to achieve, our project has following limitations:

- Due to the difficulty in procuring a compatible sensor, real time sensor data was replaced with random machine generated value.
- Since we utilized the trial plan of IBM Bluemix as cloud service provider, there is a delay of around 7 – 10 seconds in storing the payload data in the cloud DB, which stopped us from creating live graphs and charts for the monitoring of sensor data.

3.6 Future Enhancement

By interfacing PIR Motion Sensor *and Adafruit TSL2561 luminosity sensor*[P1] *with Power Monitor*[P2]*,* we can program an automated light control system with real time sensor data which will enable us to execute an ideal solution and monitor power usage. Furthermore, having a premium subscription to the cloud service will help us in generating flawless statistical data for data analysis.

3.7 Conclusion

It can be observed from the project that, with concurrent usage of CoAP and MQTT protocols, a prototype solution to the problem statement was achieved. Although there were few limitations in building the project, we were able to produce a medium to high fidelity prototype which serves the purpose of energy conservation and automation of operation of light switching.

References

[p1] - SHoLS project (Andrews)

[p2] - SLS project (Yogesh)

[p3] - Project Buzz (Aneesh)

[1] Connected Devices - Overview and Intro - Week 1.pdf

[2] https://www.smarthome.com/sc-what-is-home-automation

[3] IBM Bluemix - IBM Bluemix is a cloud platform as a service (PaaS) developed by IBM.

[4] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," The Journal of Open Source Software, vol. 2, no. 13, May 2017

[5] G.Tanganelli, C. Vallati, E.Mingozzi, “CoAPthon: Easy Development of CoAP-based IoT Applications with Python”, IEEE World Forum on Internet of Things (WF-IoT 2015)

[6] https://www.iiconsortium.org/IIC_PUB_G1_V1.80_2017-01-31.pd Ibid. Page 43 - 44.

Other references

http://ijarcet.org/wp-content/uploads/IJARCET-VOL-5-ISSUE-4-1220-1223.pdf

https://www.slideshare.net/RaizMaharjan/home-automation-with-android

https://www.slideshare.net/syedmuhammadhammad/home-automation-13430874