

CONTROLLING LAB PCs USING WI-FI

BY

ANGELINE TAI WEI JING

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2015

DECLARATION OF ORIGINALITY

I declare that this report entitled “**Controlling Lab PCs using Wi-Fi**” is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : ANGELINE TAI WEI JING

Date : 6 APRIL 2015

ACKNOWLEDGEMENT

I would like to take this opportunity to thank my supervisor, Mr. Goh Hock Guan who has always given me guidance and advices throughout the duration of this project.

Next, I would like to thank my family members for the words of encouragement that they've given me whenever I faced a problem in the undertaking of this FYP1. Due to their endless encouragement, I was motivated to keep going in finding solutions to successfully finish this project.

Last but not least, my friends, especially Then Vadivalagi and Loo Xin Yi who have helped me understand the project better and where they had also spent time in researching and thus coming up with a solution to problems faced in this project. Not forgetting my coursemates and seniors who had also provided me with useful tips.

ABSTRACT

Wireless communication using embedded system is on the rise. It is usable in various situations where mobility is important and wires are impractical. Often, the lab officers have to manually switch the computers on and off one by one which consumes time. Occasionally, few computers will be missed and it will be left on throughout the night. This project is carried out to prevent such errors and reduce the time taken by the officers to switch on all the computers. A solution will be proposed in this project to switch on and off the computers simultaneously without the conventional push of a button. This project will be using the Spark Core as the embedded device to control the computers through the network via the Spark Cloud. By mounting this device onto the computer, the efficiency of it in powering up all 40 computers in the lab will also be tested. An Android application will also be built as another alternative to control the Core wirelessly. On top of that, iterative and incremental development method is adopted in this project. The system implementation of the hardware, setup and configuration is included to show the simple architecture of this system.

TABLE OF CONTENTS

Title Page	i
Declaration of Originality	ii
Acknowledgement	iii
Abstract	iv
Table of Content	v
List of Figures	viii
List of Tables	x
List of Abbreviations	xi
CHAPTER 1 - PROJECT BACKGROUND	1
1.1 Problem Statement	1
1.2 Project Objective	2
1.3 Project Scope	3
1.4 Expected Contributions	3
CHAPTER 2 - LITERATURE REVIEW	5
2.1 Review of Technologies	5
2.2 Review of Existing Technologies	8
2.2.1 TeamViewer	8
2.2.2 Out of Band Management	10
2.2.3 Wake on Wireless LAN	10
2.2.4 Summary of Existing Systems	11
2.3 Concluding Remark	12

CHAPTER 3 - METHODOLOGY	13
3.1 System Development Models	13
3.1.1 Waterfall Model	13
3.1.2 Spiral Model	14
3.1.3 Prototype Model	15
3.1.4 Iterative and Incremental Model	15
3.1.5 Selected Model	16
3.2 Technologies Involved	19
3.2.1 Hardware	19
3.2.2 Software	21
3.3 Functional Requirement	21
3.4 Project Timeline	22
3.5 Estimated Cost	23
3.6 Concluding Remark	24
CHAPTER 4 - SYSTEM DESIGN	25
4.1 System Architecture	25
4.2 Functional Modules in the System	26
4.3 System Flow	27
4.4 GUI Design	28
4.5 Concluding Remark	28
CHAPTER 5 - SYSTEM IMPLEMENTATION	29
5.1 Hardware Setup	29
5.2 Software Setup	29

5.3 Setting and Configuration	30
5.4 System Operation	37
5.5 Concluding Remark	39
CHAPTER 6 - SYSTEM IMPLEMENTATION AND DISCUSSION	40
6.1 System Testing and Performance Metrics	40
6.2 Testing Setup and Result	41
6.3 Project Challenges	42
6.4 Objectives Evaluation	42
6.5 Concluding Remark	43
CHAPTER 7 – CONCLUSION AND RECOMMENDATION	44
7.1 Conclusion	44
7.2 Recommendation	44
REFERENCES	45
APPENDIX A	A-1
A-1 Final Year Project Biweekly Report	A-1
A-2 Final Year Project Biweekly Report	A-2
A-3 Final Year Project Biweekly Report	A-3
A-4 Final Year Project Biweekly Report	A-4
A-5 Final Year Project Biweekly Report	A-5

LIST OF FIGURES

Figure Number	Figure Title	Page
Figure 2-2-1	Waking up a computer through another computer within a LAN	8
Figure 2-2-2	Waking up a computer through its public address	9
Figure 3-1-1	Waterfall Model	13
Figure 3-1-2	Spiral Model	14
Figure 3-1-3	Prototype Model	15
Figure 3-1-4	Iterative and Incremental Development Model	17
Figure 3-2-1	Spark Core Wi-Fi Development Board	20
Figure 3-2-2	Spark Core Main Components	20
Figure 3-6-1	Project I Timeline	22
Figure 3-6-2	Project II Timeline	23
Figure 4-1-1	System Architecture	25
Figure 4-1-2	System Flow	25
Figure 4-3-1	System Flowchart	27
Figure 4-4-1	UI of Local Browser	28
Figure 4-4-2	UI of Android Application	28
Figure 5-1-1	Hardware Setup	29
Figure 5-3-1	Main Page of Spark Website	30
Figure 5-3-2	Sign-up Screen	30
Figure 5-3-3	Web IDE	31
Figure 5-3-4	Device Manager	32
Figure 5-3-5	COM Port Number	32
Figure 5-3-6	Putty Installer Download	33
Figure 5-3-7	Putty Configuration	

Figure 5-3-8	Open the Putty Configuration	34
Figure 5-3-9	Claim a Core	35
Figure 5-3-10	Connect to Local Wi-Fi Network	35
Figure 5-3-11	myMain Function	36
Figure 5-3-12	Setup Function	37
Figure 5-4-1	LED Breathing Cyan	37
Figure 5-4-2	Local Browser Interface (Off)	38
Figure 5-4-3	When Switch is on	38
Figure 5-4-4	Hardware Setup with One Computer	38

LIST OF TABLES

Table Number	Table Title	Page
Table 2-1-1	Comparison between Wi-Fi, Zigbee and Bluetooth	6
Table 2-2-1	Comparison of Existing System	11
Table 3-1-1	Comparison between Different Models	16
Table 3-7-1	Estimated cost	23
Table 6-2-1	Result of System Reliable Testing	41

LIST OF ABBREVIATIONS

<i>Wi-Fi</i>	Wireless Fidelity
<i>RF</i>	Radio frequency
<i>PC</i>	Personal computer
<i>LAN</i>	Local Area Network
<i>WPAN</i>	Wireless Personal Area Network
<i>UTAR</i>	University Tunku Abdul Rahman
<i>IP</i>	Internet Protocol
<i>BIOS</i>	Basic Input/Output System
<i>WOL</i>	Wake-on-LAN
<i>IT</i>	Information Technology
<i>UDP</i>	User Datagram Protocol
<i>USB</i>	Universal serial bus
<i>MAC</i>	Media Access Control
<i>DTIM</i>	Delivery traffic indication message
<i>BSS</i>	Basic service set
<i>NIC</i>	Network interface card
<i>FICT</i>	Faculty of Information and Communication Technology
<i>FYP</i>	Final year project
<i>I/O</i>	Input/Output
<i>DNS</i>	Domain Name System
<i>PWM</i>	Power on
<i>UART</i>	Universal Asynchronous Receiver-Transmitter
<i>SPI</i>	Serial Peripheral Interface
<i>API</i>	Application Programming Interface
<i>Kbps</i>	Kilobyte per second
<i>MHZ</i>	Megahertz
<i>GHz</i>	Gigahertz
<i>ARM</i>	Advanced RISC Machines

<i>RAM</i>	Random Access Memory
<i>KB</i>	Kilobytes
<i>IDE</i>	Integrated Development Environment
<i>ISM</i>	Industrial, scientific and medical
<i>OFDM</i>	Orthogonal Frequency-Division Multiplexing
<i>CSMA/CA</i> avoidance	Carrier sense multiple access with collision
<i>CRC</i>	Cyclic redundancy check
<i>AES</i>	Advanced Encryption Standard
<i>CBC-MAC</i>	Cipher block chaining message authentication code
<i>WEP</i>	Wired equivalent privacy
<i>WEP2</i>	Wired equivalent privacy 2
<i>CCM</i>	CTR with CBC-MAC
<i>REST</i>	Representational State Transfer
<i>SPDT</i>	Single Pole Double Throw

CHAPTER 1: INTRODUCTION

Wi-Fi stands for Wireless Fidelity and covers the IEEE 802.11 technologies. Wi-Fi or WLAN operates in the form of wireless Ethernet, an open-standard technology which allows wireless connectivity between equipment and local area networks. Similar with mobile phones, Wi-Fi network utilizes the radio waves to convey information through the network. A wireless adapter is needed to translate data sent into radio signal which will be transmitted to the router via the antenna (Gatwart, 2014).

For joining network nodes, computer network uses wireless data connections known as wireless network. This is a method by which home, telecommunication networks and enterprise establishment use to bypass the expensive operation of bringing in cables into a building. By using RF technology, which is a frequency within the electromagnetic spectrum involved with radio wave propagation, wireless network can be implemented (Moran 2008). Wireless is a more contemporary substitute to the conventional wired networking that depends on cables to link networkable devices simultaneously.

1.1 Motivation and Problem Statement

It has always been a problem incorporating both hardware and software together in order to build an internet-linked product. Various methods can be used to overcome the problem of manually switching on and off the PCs in the lab. The lab assistants have to manually on and off all the PCs. Occasionally, the lab officer unintentionally left out a few computers when closing the lab and as a result these computers will be left on a standby mode the whole night. Also, at times the assistants may be busy with other work any may not have sufficient time to turn on or off the PCs in the lab. With this project, all the PCs can just be switched on and off wirelessly with a master PC or on an Android mobile device which will save time and human error can be reduced.

Although the implementation of the Spark Core to control multiple PCs can be widely used in many different areas, however, the focus of this project is mainly to help

lessen the workload of UTAR's computer lab officers. Indirectly, productivity will increase and the lab officer will have more time on hand for other tasks.

The Spark Core, which is an embedded system, does not have an operating system. Instead, it runs on firmware. The Core is a development platform powered by cloud which is also Arduino-compatible and Wi-Fi enabled (Kickstarter, 2013). Spark Core is used in this project as the Spark Core is quite flexible in a way that Spark's web development environment allows the creation of a connected product in the cloud, without installing a toolchain on the system. One can flash a code written in its web IDE to the Core wirelessly. Setting up of this device is also simple and relatively fast either from a desktop via USB or from an Android device by using the Tinker app.

1.2 Project Objectives

This project is carried out to help in cutting down the time taken by the lab officer to switch on and off every PCs in the lab. A few objectives need to be achieved for this project to be successful.

The first objective of this project is to develop a solution to switch on and off PCs simultaneously without the conventional pushing of the button for every PC. Without utilizing the typical way of switching on a PC, the main goal in this project is on/off all the PCs at one go.

The second objective is to develop a wireless control system by mounting on a device to a main PC. This PC will act as the main switch to control all the PCs in the lab by flashing the code to turn on and off into the Core.

The third objective is to develop the mobile application as an interface to control the Core. A mobile application will be used similar like a remote control to control the Core to on and off the PCs in the lab. This is a secondary method to control the Core instead of using the main PC in the lab.

The last objective of this project is to test the reliability of this system in powering up all the computers at the same time. This final objective is to check whether all the above objectives have been achieved successfully by managing to power on and off all

the computers in the FICT lab by controlling the Core with a main PC and a mobile application.

1.3 Project Scope

The final deliverable of this project is a prototype to connect and control multiple PCs in a lab at the same time through a main PC or mobile application by making use of the Spark Core. There are 40 PCs in a UTAR lab. So, these PCs will be linked together by using the Spark Core via Wi-Fi to control it. In the context of this project, the scope of 'control' here is to switch on and off the PCs at the same time. In other words, the Core, via UTAR Wi-Fi will connect to the Cloud and holds open a persistent connection (Spark, n.d.). This will make it accessible from wherever in UTAR where it will be very convenient to the lab officer as they need not open the lab just to switch on all the PCs as they can do it through the application.

Two wire cables that are colour coded are the system panel cables, which helped to identify where the cables connect to the motherboard system panel connector. A pin will be connected from the Core to the front panel header on the motherboard to power up the computers. Before all this can take place, the hardware with Wi-Fi connectivity has to be connected to an access point first.

1.4 Main Contributions

This project contributed a solution to automatically switch on and off a group of networked PCs in UTAR lab without the hassle of having to do it manually PC by PC. Not only can it be used in UTAR lab, but other similar environment where the need to switch on and off PCs at the same time is high and this prototype can be used to cut human error.

It also cuts the workload of the lab officers and the time consumed to carry out this task manually every day. With this, the officer has more time for other lab-related matters.

1.5 Organization of the Report

Introduction of the report is in Chapter 1 where it will discuss briefly on the overall aim, objectives and scope of the project. Chapter 2 is on literature review where existing technologies are being highlight, compared and contrast with the proposed project. Next, system methodology will discussed on the comparison of various system development models and subsequently, the best model will be chosen to be used in this project. Chapter 3 will also consist of hardware, software and functional requirement of the project besides also on the expected challenges and estimated cost.

System design is in Chapter 4 where it will describe on how the project is developed along with its system flow and GUI design. In Chapter 5, System Implementation, it will consist of hardware and software setup of this project and also its setting and configuration and system operation. The next chapter which is Chapter 6 is about system evaluation and discussion where the system will be tested and evaluated. The final chapter is on the conclusion and recommendation of the project.

CHAPTER 2: LITERATURE REVIEW

2.1 Review of Technologies

In a project, various technologies can be used to create an application or software in order to produce a workable end product. It is typically a norm to compare several technologies before finally deciding on which technology to utilize in a project. For this project, comparison of Wi-Fi, Bluetooth and ZigBee technologies has been carried out before finally deciding on which technology to use to accomplish the objectives of this project.

Bluetooth is designed for short-range wireless connectivity. It is low in terms of cost, range and power supply with IEEE 802.15.1 standard. The unlicensed 2.4GHz ISM Band which ranges from 2400 – 2483.5 MHz is used by Bluetooth as it also shares this range with Wi-Fi and IEEE 802.15.4. Bluetooth uses frequency-hopping spread spectrum (FHSS) which means that it jumps from one channel to another in order to evade collision with other wireless communications using the same ISM band. Bluetooth is structured in a master-slave mode. Bluetooth radios are defined in 3 classes, namely Class 1, 2 and 3. Class 3 has the shortest range with 1m, followed by 10m and 100m for Class 2 and 1 respectively (Fornazier, Martin and Messner, 2012). However, its topology can support 1 master and up to 7 slaves only.

Zigbee, like Bluetooth has the same range that is in the frequency band of 2.4GHz and both are also a WPAN standard. Zigbee is based on IEEE 802.15.4 standard with communication reaching up to 500m (Fornazier, Martin and Messner, 2012). Direct sequence spread spectrum (DSSS) is the modulation technique utilized by this technology where its bandwidth of the signal will be spread into a wider frequency range. Zigbee supports 20kbps, 40kbps and 250kbps data rates depends on the frequency band adopted for transmission. According to Kanwar and Khazanchi (2012, pp.67-74), this technology is low cost and low power consumption besides being able to support large number of nodes up to 65,645 nodes.

On the other hand, Wi-Fi technology for wireless personal area network(WLAN) includes IEEE 802.11a/b/g/n standards. In order to implement full-duplex communication in one channel, this technology utilizes OFDM along with CSMA/CA to evade collision

among the overlapping channels and also different technologies that uses the same frequency (Fornazier, Martin and Messner, 2012). The range of Wi-Fi enabled devices is approximately 100m. The latest version of Wi-Fi can reach up to 250mbps of data transfer rate.

Technology	Wi-Fi				Bluetooth	Zigbee
IEEE Specifications	802.11a	802.11b	802.11g	802.11n	802.15.1	802.15.4
Maximum data rate (Mbps)	54	11	54	~600	0.8 – 1.0	0.02 – 0.2
RF band (GHz)	5.0	2.4	2.4	2.4 or 5.0	2.4	2.4
Maximum Range (m)	120	140	140	250	1 – 100	10 - 100
Network size	32				8	>65000
Topology	Ad-hoc/star				Ad-hoc/ piconet, scatternet	Ad-hoc, point-to- point/ star or mesh
Security	WPA2, WPA, WEP, TKIP				EO stream cipher	AES block cipher, WEP
Cost	High				Low	Low

Table 2-1-1 Comparison between Wi-Fi, Bluetooth and Zigbee (Lee, Su and Shen, 2007; Fornazier, Martin and Messner, 2012)

Wi-Fi technology is chosen in this project to control the PCs in the lab by utilizing the Spark Core. One obvious advantage that Wi-Fi has over the other short-range technologies is that it operates at a faster rate, 11Mbps compared to Bluetooth of around 720Kbps. Besides that, Wi-Fi is designed to link the entire network up which makes it easier to link multiple PCs in the computer lab and it is easier when the PCs in the lab will all be in the same network. Aside from the 2.4GHz frequency, Wi-Fi also operates on 5GHz, thus, having additional choices to form several networks which will not interfere with other technology that uses the same 2.4GHz frequency.

Moreover, both Bluetooth and Zigbee have a shorter range up to 100m only whereas the range of Wi-Fi can reach up to 250m. This will be ideal as the Spark Core will be able to connect and control more computers. Wi-Fi technology can support up to 32 nodes as compared to Bluetooth of only 7 slave nodes. As the purpose of this project is to control multiples PCs in the lab, thus for Zigbee technology which can support more

than 65000 nodes is unnecessary as there only have 40 PCs in a lab. Star topology is also easy to set up, expand and troubleshoot.

In terms of security, EO stream cipher and shared secret with 16-bit CRC is adopted by Bluetooth technology while Zigbee uses AES block cipher with counter mode (CTR) and CBC-MAC. As there were a few flaws with the original encryption method of WEP, it is then succeeded by WEP2 where CCM and AES block cipher are also used (Lee, Su and Shen, 2007). Overall, Wi-Fi technology is used here in this project to connect and control in terms of switching on and off the PCs in the lab. Wi-Fi has the upper hand in terms of range, data rate, frequency, network size and security though with all the added advantages, the cost is slightly higher than the other technologies.

2.2 Review of the Existing Technologies

2.2.1 TeamViewer (software)

TeamViewer is an instinctive, fast and secure application for collaboration on projects over the Internet, with support for web conferencing, remote login, control and sharing of files (Secor, n.d). In TeamViewer, a function called Wake-on-LAN (WOL) will enable the turning on of an offline computer remotely. Wake-on-LAN on TeamViewer can be used through two different methods:

- Waking up a computer through another computer within a LAN

This method can be used when there is a computer in the network that is always on. The particular PC will be used to wake up the target computer.

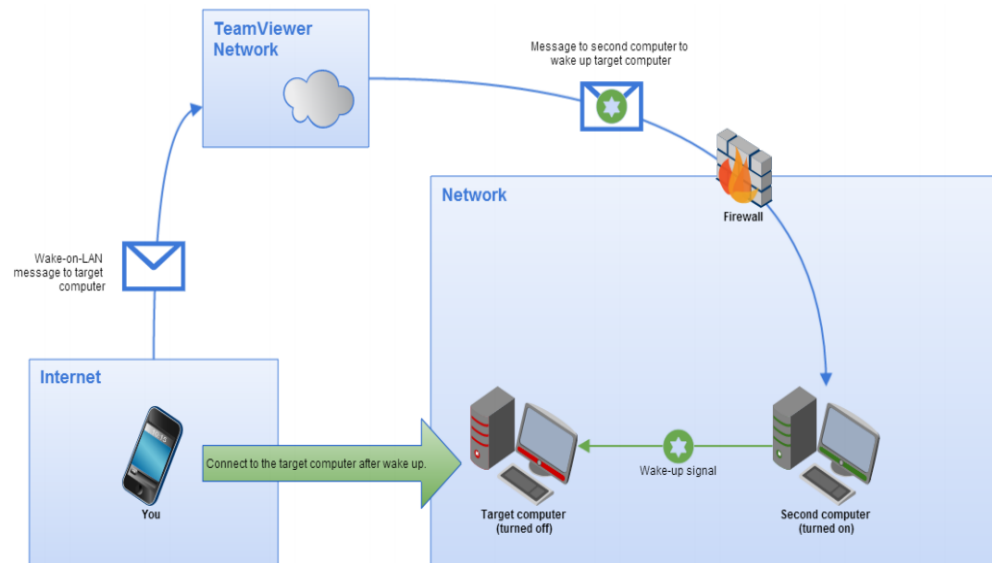


Figure 2-2-1 Waking up a computer through another computer within a LAN

(TeamViewer, 2013)

- Waking up a computer through its public address

This method is used when a fixed static IP is available through the internet provider or the target computer's router has a public domain address

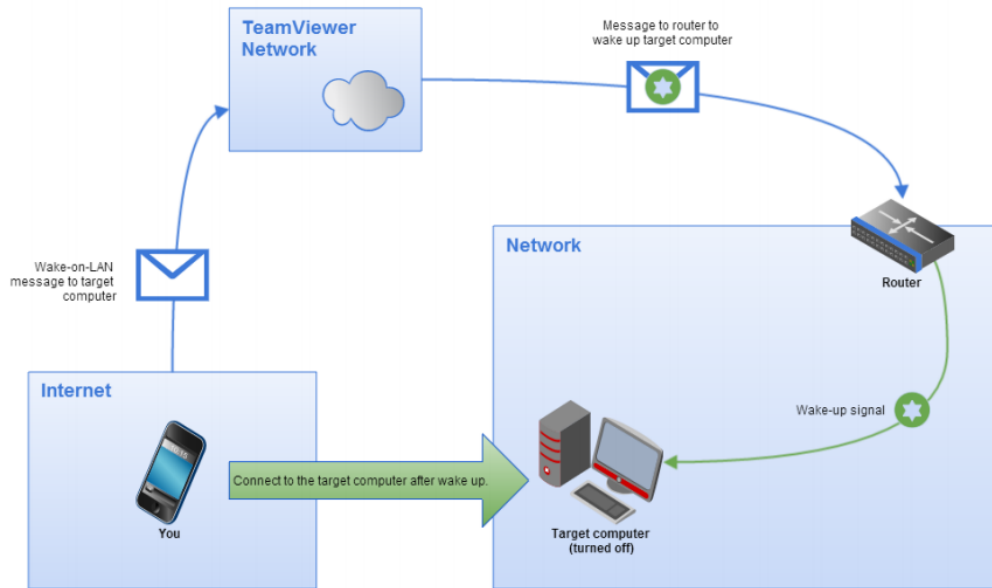


Figure 2-2-2 Waking up a computer through its public address (TeamViewer, 2013)

For the computer to perform WOL, first and foremost it has to be on a wired connection and properly set up on the machine. First of all, WOL has to be activated on the machine's BIOS and this procedure differs from PC to PC. Besides that, the network card of the computer has to be configured so that it is being supplied with power at all times (TeamViewer, 2013). Though it is relatively simple to configure and enable WOL on the target PC, however, not every computer supports WOL function.

For the first method, the PC has to be constantly on in order to send a signal to the target PC. Power consumption will be a major problem here. Moreover, the router has to be configured to enable port forwarding for the second method to work. Port forwarding has to be set up depending on whether all the computers in the network is to be awoken or only selected computers are to be awoken (TeamViewer, 2013). In worst cases, the router's configuration menu is unable to save a static ARP entry to allow the router to

access to the appropriate computer, and thus it has to be done through Telnet or SSH (TeamViewer, 2013).

2.2.2 Out-of-Band Management

In Configuration Manager 2007 SP1, out of band management offers great management control for computers that has Intel vPro chipset and Intel Active Management Technology (Intel AMT) firmware versions 3.2 or later (Bailey, 2008). The computer can be managed below the operating system layer in situations where the Configuration Manager client is not installed, the computer either has stopped responding or it does not have an operating system installed or it is switched off (Microsoft System Center, 2013). When the computer is switched off, in sleep or is hibernating, or in an unresponsive state, out-of-band management permits an administrator to link to a computer's management controller via the operating system (Microsoft System Center, 2013).

Out-of-band management tools can be used for remote IT management to offer secure, alternate paths to remote access, monitor and manage IT assets throughout the network infrastructure. The conventional WOL technology uses UDP datagrams, and this out-of-band management method differs in terms of it uses an established transport session for a more consistent and controlled communication. As a result of this, this method takes up more processing and is time consuming. Out-of-band system has to function as an integrated system for it to be effective. The capability of the out-of-band tools also need to support all industry-standard security protocols and specifications to be incorporated into the current security infrastructure of an organization or network (Koh, 2011).

2.2.3 Wake on Wireless LAN (WoWLAN)

Wake on wireless LAN (WoWLAN) is a technology that enables remote waking up of workstations from a standby mode. It is similar with Wake-on-LAN (WOL). According to Nagy (2010), to wake up a machine, one need to log into the management suite and enters the workstation DNS name. To find out its IP and MAC address, the device in DNS and within its inventory is queried by the management workstation. Next,

transmission will begin for the magic packet for a pre-determined interval when the master workstation in the LAN is contacted. During the next DTIM period, the packet is sent out where the wireless computer awakes from its standby mode when queued broadcast traffic shown in the DTIM beacon and the magic packet instructions for the workstation to be fully awake from standby is sent out.

The capability to place workstations into a low-power mode saves energy and has a more efficient power management. However, integrated adapters are required to carry out WoWLAN. The motherboard will be integrated with the wireless adapter to control the workstation power state. Wireless clients must remain connected to the BSS at all times and thus, workstation has to be on a standby mode. Power has to be supplied to the wireless NIC to maintain its network association and therefore hibernation or full power off states will not work. Moreover, this method also has a negative performance impact. This is because permitting broadcast wireless frames across the wireless network will decrease network performance.

2.2.4 Summary of Existing Systems

Solutions	TeamViewer	Out of Band Management	WowLAN	Requirement of this Project
Switching on all computers	Yes	Yes	Yes	Yes
Switching off all computers	No	Yes	No	Yes
Switching on certain computers	Yes	Yes	Yes	No
Switching off certain computers	No	Yes	No	No
Cost-effectiveness	Yes	Yes	Yes	Yes
Router configuration	No	No	Yes	No
Internet connection	Yes	No	Yes	Yes
Specific wireless adapters needed	No	No	Yes	No

Table 2-2-1 Comparison of Existing Systems

2.3 Concluding Remark

Wi-Fi has come to be an essential part of our daily lives. These days, Wi-Fi is utilized in home and office network besides also being used in multiple electronic devices. In this project, Wi-Fi technology is chosen to control the computers in the lab by utilizing the Spark Core. To my best knowledge it has yet to be done. Wi-Fi is used as it has high data rate and longest maximum range compared to Zigbee and Bluetooth. Besides that, Wi-Fi also operates in an additional 5GHz and this frequency can help prevent interference from other technology which uses the 2.4GHz frequency. That is why this technology is chosen in this project.

From the existing technology, TeamViewer and WoWLAN do not control the computers by switching it off. It only works in powering it on while in this project, both the function of switching on and off all the computers in the lab will be implemented. It also does not require any router configuration and special adapter to use Spark Core.

CHAPTER 3: METHODOLOGY

3.1 System Development Models

The development of a software product needs a software development process in order for the project to meet its requirements and deadlines. Choosing the right system development model is important in a project as each model has their pros and cons and may offer better advantages in one situation over the other.

3.1.1 Waterfall Model

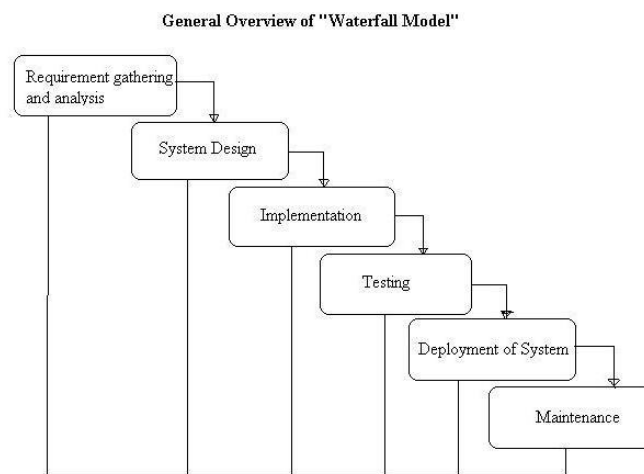


Figure 3-1-1 Waterfall Model (ISTQB Exam Certification, 2012)

Waterfall model is also referred as a linear-sequential model where one phase has to be completed before moving on to the next. The waterfall model consists of 6 stages, namely requirement gathering and analysis, system design, implementation, testing, deployment and maintenance. This model is typically use for small projects with clear requirements and shorter timeline. Clients often does not have a clear idea of what they want in a system and tend to change their mind so using this model may cause confusion for changes to be made in between phases (ISTQB Exam Certification, 2012). Each of the phases also does not overlap with one another.

Waterfall model is also rigid as it is not possible to go back to the previous stage and there will certainly be a major problem during the implementation stage. The amount of risk and uncertainty is high with this model and the client is only involved in the

beginning stage of this model. The cost to fix any failures is high with this waterfall model as every stage need to be updated.

3.1.2 Spiral Model

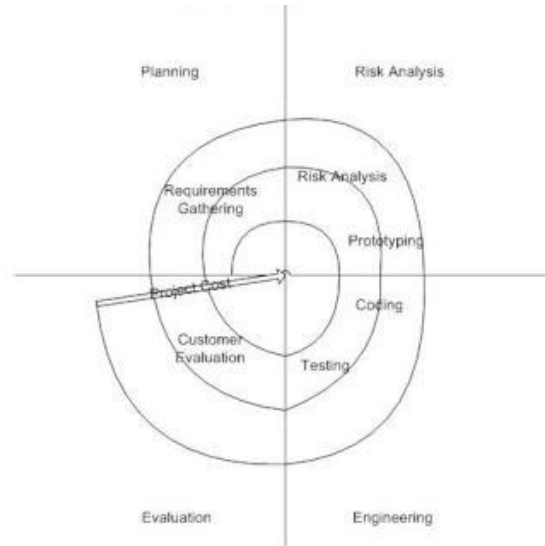


Figure 3-1-2 Spiral Model (ISTQB Exam Certification, 2012)

There are four phases in this model which are planning, risk analysis, engineering and evaluation. A project will repeatedly pass through all these phases several times in iterations in their development stage. In the planning phase, requirements of the project are gathered. The second phase will produce a prototype at the end of the phase. However, before that can happen, the risk of this project will be identified and an alternative solution will be recommended and implemented. In the engineering phase, the development of the software is done along with the testing after the software is developed. In the last phase which is the evaluation phase, the software will be evaluated by the client if it is the expected output before the development of the project proceeds to the next spiral (ISTQB Exam Certification, 2012).

Spiral model can be implemented when client are unsure of their needs and also the assessment of costs and risk is significant. One of the advantages of this model is that unlike the waterfall mode, additional functionalities can be added at any stage of the development process and this model emphasize on risk analysis thus risk avoidance is enhanced. On the down side, this model is expensive to be used and an expert is needed

to come up with very specific risk analysis. Smaller projects are discouraged from using the spiral model (Jain & Maheshwari, 2012).

3.1.3 Prototype Model

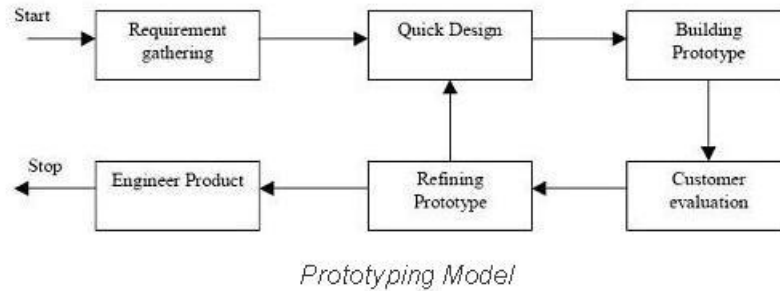


Figure 3-1-3 Prototype Model (ISTQB Exam Certification, 2012)

According to Jain & Maheshwari (2012), for this model, with the requirements of the software known, a prototype will be constructed hence the client will have an idea of the completed product. The client will also understand the requirements better by interacting with the prototype. Large system usually uses this model for their development.

Client involvement is high in this model, as a result, any errors or confusion can be sorted out and fixed at an earlier stage. Missing functions can also be easily adapted into the model. On the other hand, using this model may lead to software expansion way beyond what was originally intended as the prototype model implements a prototype then repairs it repetitively until the client is satisfied with the software or output that the project produced. Prototype model is typically used where there are heavy user involvement so that feedbacks from client can be used to better the system constantly.

3.1.4 Iterative and Incremental Model

This model is the combination of iterative method and incremental model. The idea of this model is to develop a system iteratively and in a smaller portion at one time. A part of the requirements of the project will be implemented and iteratively improve the system until the full expected end system is done. Modifications to the design will be made and additional functionalities will be implemented at each iteration. Phases will

undergo repetition so that the system can meet with the constant changing requirements of the users. As the important work is done first in this model, risk of delay will be reduced.

3.1.5 Selected Model

The table below shows which model to use under various conditions.

Model/Features	Waterfall	Spiral	Prototype	Iterative and Incremental
Requirement specifications	Beginning	Beginning	Frequently changed	Beginning
Cost	Low	High	High	Low
Resource control	Yes	Yes	No	Yes
User involvement	Only at beginning	High	High	Intermediate
Maintenance	Less	Typical	Routine maintenance	Promotes maintainability
Documentation & training required	Vital	Yes	Weak	Yes
Time frame	Long	Long	Short	Long
Guarantee of Success	Less	High	Good	High

Table 3-1-1 Comparison between Different Models (Jain & Maheshwari, 2012)

The model chosen to be implemented in this project is the incremental and iterative method. According to Technopedia (2014), this model identifies and implements a part of the software where it will be revised and additional requirements will be added and carries out in stages. The requirement specification is at the beginning of the project, where in this project it is stated in the objectives. Besides, the cost of this method is low compared to the spiral and prototype model. User involvement is intermediate and this will be helpful in terms of satisfying user requirements in between development by being able to add in additional functions for a better end product. User involvement of the waterfall model is only at the beginning where the user will not have much say later on in

this project. This model also promotes maintainability which will make the product more relevant over times and the success rate is also higher compared to the other models.

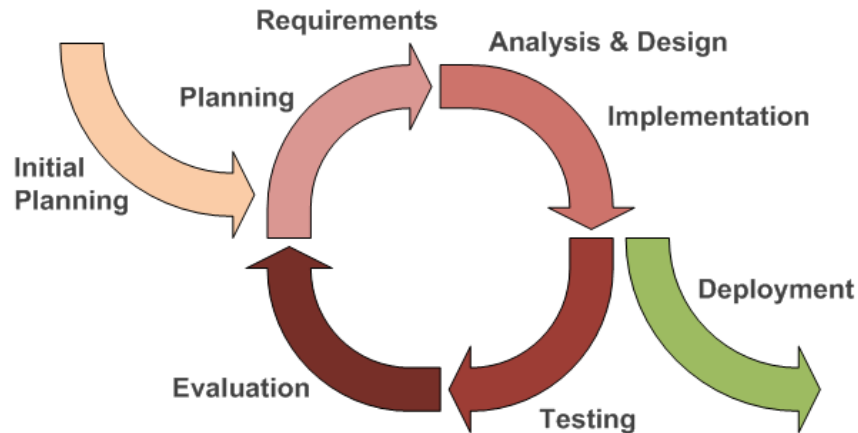


Figure 3-1-4 Iterative and Incremental Development Model
(Agile Development Tools, 2013)

In this project, there is an estimation to have 2 iteration phases within the duration given to develop this project. A prototype should be produced in the first iteration while the final product should be produced in the second. The first and second iteration would take approximately around 4 months. Requirements analysis and design will be the focus of the first iteration while implementation and testing will be the focus of the second iteration. Iterative and incremental method is also more flexible compared to waterfall model where a backtrack is not possible as it is a linear sequential model.

Planning

This is the stage where the proposal is drafted along with the project timeline and milestone. An initial survey will be carried out with a few of the lab assistants in UTAR to find out about the time taken to switch on all the PCs in the computer lab. Before the beginning of requirement gathering, identification of project constraints needs to be done. Planning stage is a vital stage for any project as it ascertains the system requirements that will be developed.

Requirement

In this stage, by researching on the Internet, the requirement of solution is gathered. Literature review as preliminary analysis helps to identify current technology being used and various other methods to turn on and off PCs simultaneously will be reviewed and compared.

Analysis and Design

In analysis and design stage, this is where the problem statement, project objectives and project scopes are formulated. Functional module to be used in this project is identified in this step, such as the Spark Core where it will be used to control the PCs. The system will then be designed where the Spark Core will be connected to the PCs to control them via Wi-Fi through the cloud. The existing prototype will be improved further through the feedbacks in the first iteration in the same stage of the second iteration.

Implementation

Before implementation, the pin that is responsible to power on and off the computer is identified. This pin will then connect to the Spark Core and the code will be flash into the Core to power on or off the respective computers. An Android application will be developed using Java to serve as an interface to control the Core. Output of the first iteration should be a simple prototype that controls only one computer.

Testing

The prototype will be tested to ensure that it has achieved its objectives. The first testing is to test the effectiveness of the Spark Core to switch on all the computers in the lab. The second testing will be done to test whether the mobile application created to control the Core will be able to control it in power on and off the computers in the lab. Milestone in this stage is the test report.

Evaluation

The report from the previous stage will be evaluated in this stage in order to identify any problems found from the previous testing stage. The result from the first iteration will be used to improve the prototype developed for the second iteration and the result of the second iteration will be used to conclude this project.

3.2 Technologies Involved

Technology is an indispensable part of most project. In this project, the help of technology is needed to successfully achieve the project objectives. Both hardware and software plays an important role in this project.

3.2.1 Hardware

Spark Core is a development platform which is Arduino-compatible, Wi-Fi enabled and powered by cloud. Below are a few hardware specifications of the Spark Core (Spark, n.d.):

- 32-bit STM32 72MHz ARM Cortex M3
- 128KB of Flash memory, 20KB of RAM
- Texas Instruments SimpleLink CC3000 Wi-Fi chip
- 8 digital I/O pins, 4 PWM pins, 8 analog I/O pins
- UART (Serial), SPI, and I2C peripherals
- Wi-Fi range of 100-300ft



Figure 3-2-1 Spark Core Wi-Fi Development Board (Spark, n.d.)

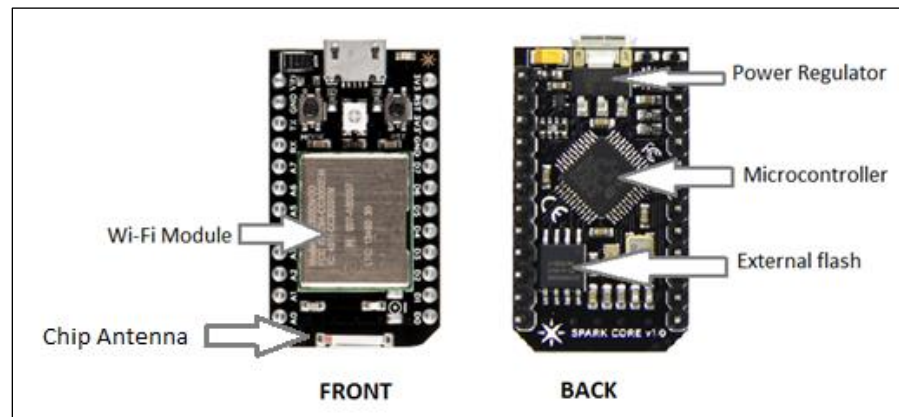


Figure 3-2-2 Spark Core Main Components (Spark, n.d.)

A breadboard is also needed to make it convenient to wire the components to the Core without soldering. The USB cable is used to connect the Core to the computer, a USB power brick or to a USB battery pack to power up the Core and to program it. A relay shield, along with the Spark Core allows the control of the PC over the internet.

A desktop with Windows 7 installed as the OS is also needed as an interface to program the cloud. A smart device has to be equipped with an Android operating system be it a smart phone or a tablet. This will be used as an alternative to control the Core. Lastly, an access point is needed to enable the Core and the smart device to connect to the Wi-Fi.

3.2.2 Software

Besides hardware, software is also another important element in this project to ensure that the hardware is working smoothly. The language used to write the program for the Spark Core is the Arduino Programming Language and the REST API is used to write the software for the smart device that connects to the Core. Windows 7 is used as the platform for the computers.

Arduino Programming Language is a cross-platform software and is compatible with other platform as well, like Linux, and thus why this language is used. It is also easy to use for beginners and flexible enough to accommodate advanced users. Being an open source and extensible software are also among the reasons why this language is chosen as it can also be expanded via the C++ library.

Spark build is used to create and flash customized applications to the Spark Core from any web browser and also to equip the Core with the full capabilities of the internet. The Spark Cloud, along with the REST API enable the user to access internet from anywhere besides also creating a secure environment where the web browser and the Core does not need to speak the same language. Spark Android application helps in connecting the Core to the Wi-Fi network and in controlling the core.

As for the control by using Android part, Android Software Development Kit (SDK) is used to create the interface to control the Core via Android mobile application. Besides that, PhoneGap is also used to create the mobile application. It is an open-source framework for building cross-platform mobile apps using HTML5, Javascript and CSS. Eclipse Luna which is an open-source integrated development environment (IDE) is also used as it supports Java, which is the native language for Android applications. PhoneGap is created within Eclipse.

3.3 Functional Requirements

- User interface (UI)

This will act as an intermediate between the user and the system. An Android UI will be created to serve as the interface for the user to use to control the Spark in switching on and off the computers in the lab

➤ Hardware connectivity

The relay shield has 4 relays controlled by pin D0, D1, D2 and D3 on the Core. NPN transistor will triggered each relay and it will take a control signal from the core and switches the relay coil ON and OFF which subsequently makes or breaks the electrical contact on the output. The relays are SPDT type, thus having 3 terminals at the output: COMMON (COMM), Normally Open (NO) and Normally Closed (NC) (Spark, n.d.). Here, the ground pin of the front panel header of a PC will be connected to the COMM terminal of the relay shield and the power switch pin will be connected to the NO terminal of the relay shield to control the PC. One relay shield is able to control up to 4 PCs at a time.

➤ Network connectivity

Network connectivity is needed in order for the Core to be connected to its Spark Cloud. A function will be defined and exposed in the Core firmware in order to control it. These functions will be called remotely using the Spark Cloud API.

3.4 Project Timeline

This is the Gantt representation of project timeline during the first semester of project development.

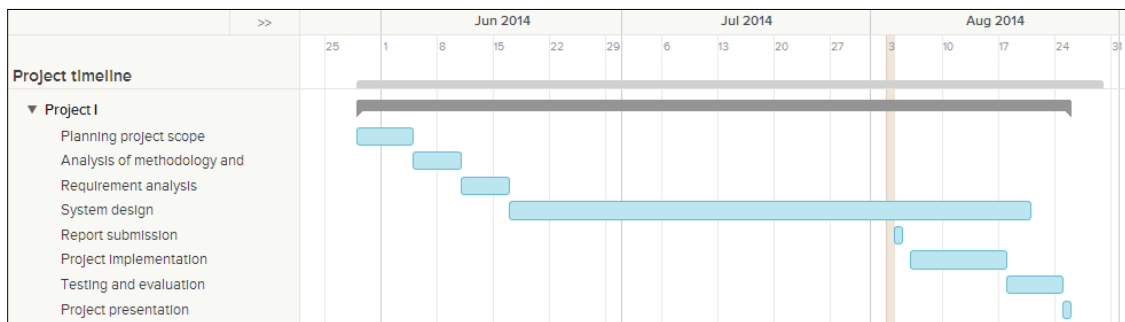


Figure 3-6-1 Project I timeline

Below is the Gantt representation of project timeline during the second semester of project development.

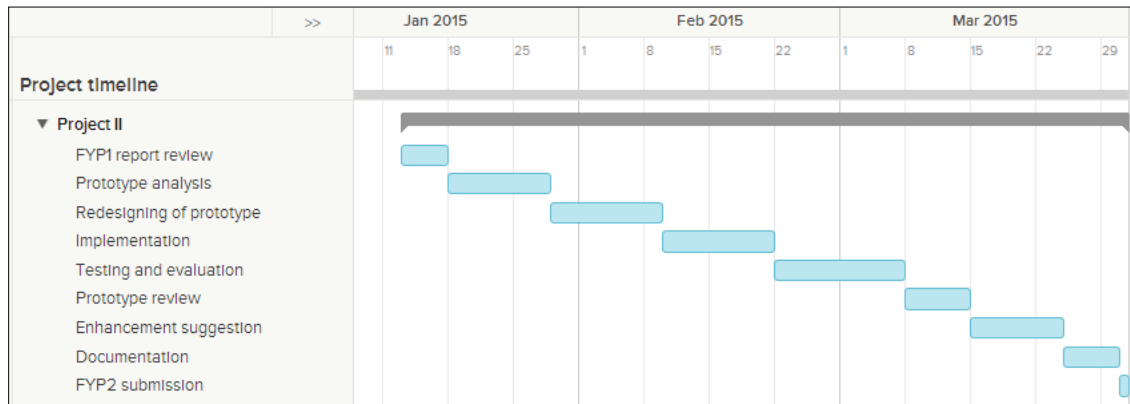


Figure 3-6-2 Project II timeline

3.5 Estimated Cost

Items	Cost per Unit	Cost per Lab (40 PCs)
Spark Core	RM120	RM120 x 10 = RM1200
Relay shield	RM95	RM95 x 10 = RM950
Wire	RM1	RM1 x 40 = RM40
TOTAL	RM216	RM2,190

Table 3-7-1 Estimated Cost

In this project, the estimated cost is not too expensive as only a Spark Core module is purchased for the price of RM120. The other Spark module is from UTAR computer lab while all the computers and relay that are used for testing and implementation purposes are from the computer lab as well. A relay shield can connect to 4 computers and therefore only 10 relay shield is needed with each Core mounted on a shield.

3.6 Concluding Remark

In conclusion, iterative and incremental development model is used in the process of developing this prototype. The technologies involved are listed here with the functional requirement so as to not get sidetracked in the next phase of development. Project timeline is also listed so as every project planning stages will be on track and not left behind. Along with Spark Core and its necessary hardware and software, this project is carried out in order to achieve its objectives of controlling the computers in the lab.

CHAPTER 4: SYSTEM DESIGN

4.1 System Architecture

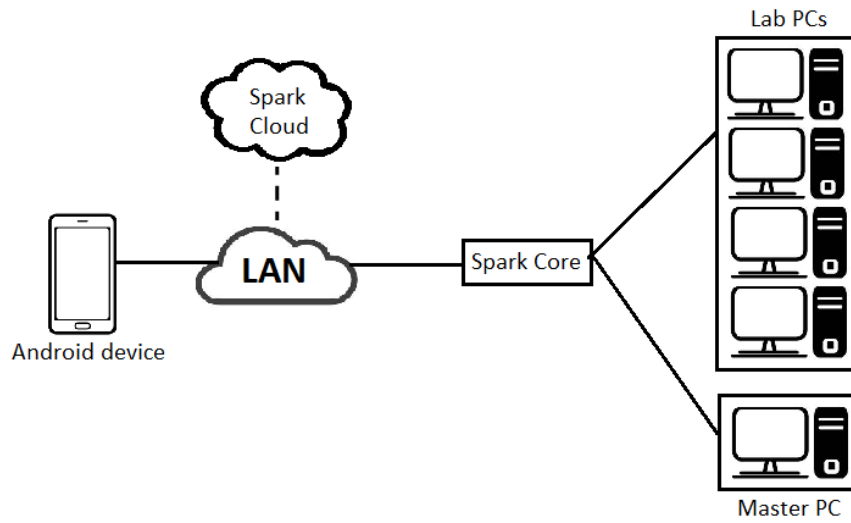


Figure 4-1-1 System Architecture

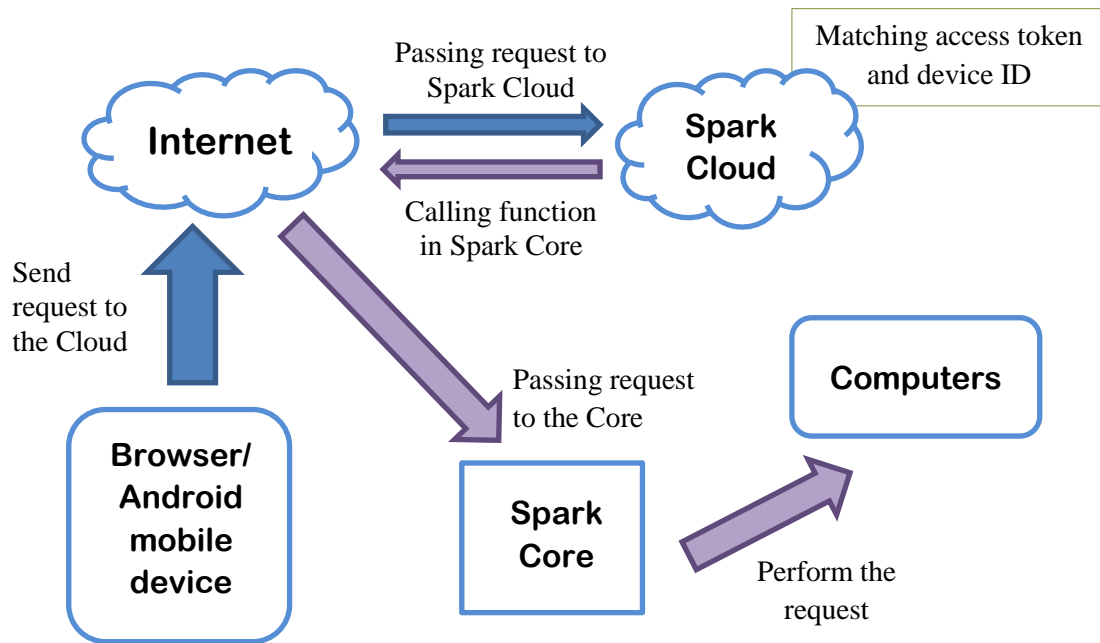


Figure 4-1-2 System Flow

Figure 4-1-1 above depicts the proposed system architecture for this project. The 40 computers and the lab assistant's computer will be wired to the Spark Core which is mounted on a relay. This will make the core accessible from any place through the REST API. An Android mobile device will be connected to the same access point and it establishes a connection to the Spark Cloud. The Spark Cloud which acts as the middle man exists for simplicity, global availability and security. The Core will connect with the cloud and hold open a persistent connection when it connects with the Wi-Fi network. This will enable the controlling of the computers in the lab from the mobile device through the cloud over the internet.

Figure 4-1-2 depict the system flow of the system. By using a local browser or an Android application, an ON/OFF button will be toggled and request will be send to the Spark Cloud through the Internet. At the cloud, it will match the access token and device ID of the Spark Core. If it matches, it will send the call function that is requested by the browser or the application to the Spark Core. When the Spark Core received the call function, the Core will check the request that contained the function name whether it matches with the function name that is in the firmware. When it matches, the Core will perform the function accordingly and the computers will be switched on.

4.2 Functional Modules in the System

A module is standardized parts or a stand-alone unit that can be used to build a more complex structure, in this case, a complete system. One of the functional module of this project is the Internet. The Internet provides the connectivity to connect to the Spark Cloud in order to control the Core. It also acts as a medium for the transfer of request to and from the Cloud through the Internet.

Besides that, another functional module is the hardware in this project, especially the Spark Core as it is the main module of this project. The Core, together with the relay shield is the components that allow the computers to be controlled from the browser and mobile application.

On top of that, user interface is another functional module in this project. It allows the user to interact with the Core. An Android application with a simple interface is created to be a secondary method to control the computers in the lab.

4.3 System Flow

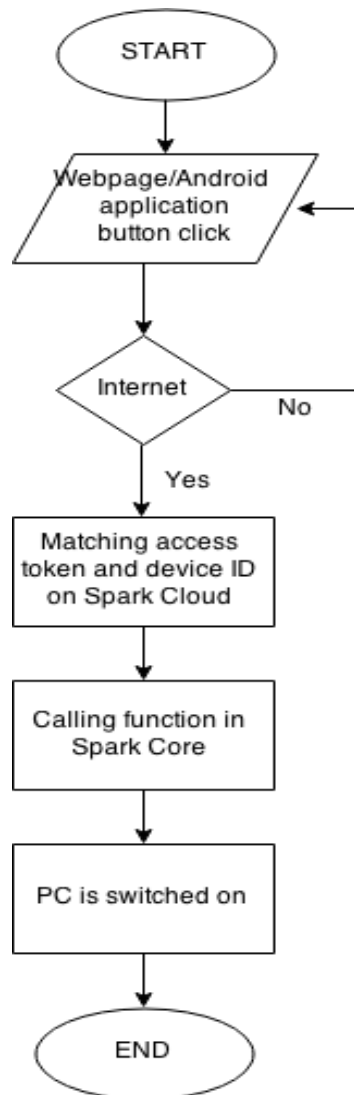


Figure 4-3-1 System Flowchart

Figure above shows the flowchart of this system. When the button on the browser or Android application is clicked, it will access the Spark Cloud through the Internet by using Wi-Fi. At the cloud, it will match the access token and device ID of the Core. Next, it will then call the function in the Spark Core and subsequently the computer is switched on.

4.4 UI Design

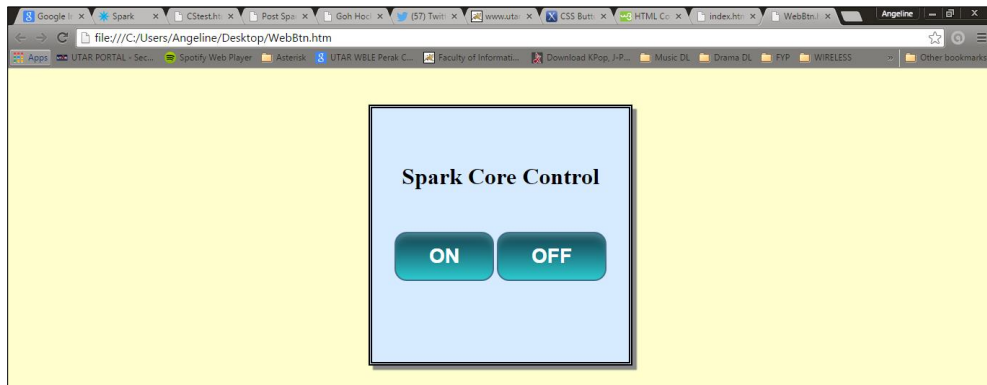


Figure 4-4-1 UI of Local Browser

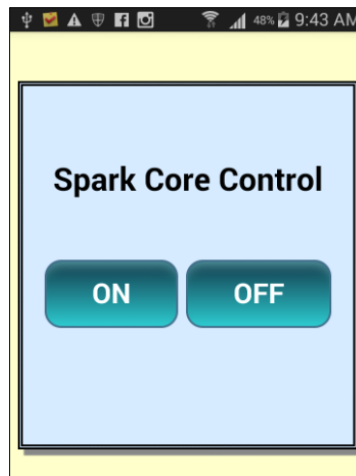


Figure 4-4-2 UI of Android Application

The design of the UI that is used as a middleman to connect the user to the core is relatively simple. It will have a toggle switch to control the Core to on or off the computers.

4.5 Concluding Remark

In summary, the Spark Core will connect to the Spark Cloud before the Core can control the computers in the lab. The computers will be controlled wirelessly via the Core once it holds open a connection to the cloud. A UI is created to act as an intermediate between the user and also the system which will make the system more user-friendly.

CHAPTER 5: SYSTEM IMPLEMENTATION

5.1 Hardware Setup

In order to carry out this project, an access point without a proxy is required. The Spark Core is mounted onto a relay shield that has 4 relays. Each relay will be connected to the panel header on the motherboard and the Core will then be powered on via a USB cable to a master computer. Figure 5-1-1 is a demo connecting just 1 computer to the relay. The core in the figure below can be considered as the master core and the other core that connects to the other group of 4 computers is the slave core. The other slave core can be powered on by using a power adapter.

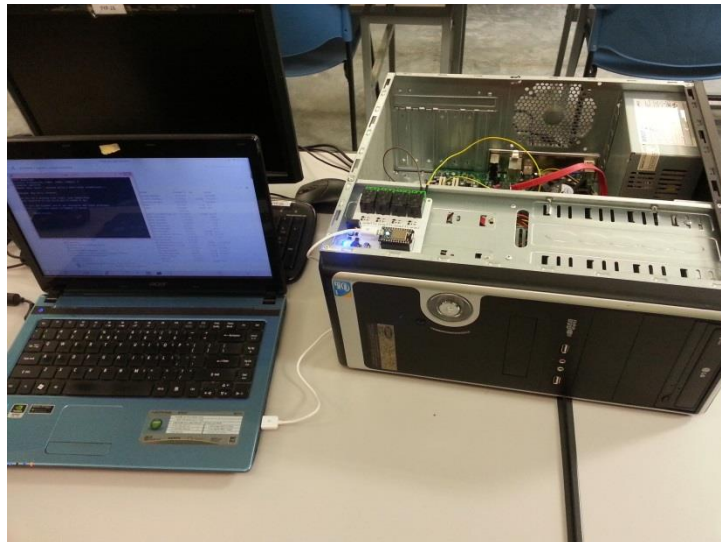


Figure 5-1-1 Hardware Setup

5.2 Software Setup

The Spark Core is connected to the Wi-Fi network over USB by communicating through Serial. A serial terminal, PuTTY is downloaded to be used in this case. PuTTY is a free, open-source terminal emulator, serial console and network file transfer application. It supports network protocol such as SCP, SSH, Telnet, rlogin and raw socket connection.

Browser is also needed to control the Core via the internet through the cloud. Android application will also be used to control the Core to power on and off the computers.

5.3 Setting and Configuration

Spark Cloud Account Registration

A Spark account is needed to access the web IDE where one can write code and upload it to the core.

1. Go to <https://www.spark.io/> and click on “Launch” as shown below



Figure 5-3-1 Main Page of Spark Website

2. A pop-up will appear that requires user to enter their username and password in order to sign-up. Enter the details accordingly and click on “Sign-up”.

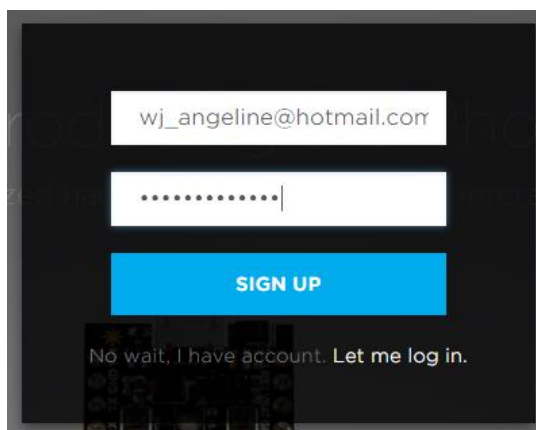


Figure 5-3-2 Sign-Up Screen

3. Once log in, the following screen can be seen with the menu pane on the left. This is the Web IDE or the Spark Build.



Figure 5-3-3 Web IDE

Connect the Spark Core Device

Once registration is done, next is to connect the Spark Core device. Make sure the Spark Core is plugged into the computer via USB cable.

1. Go to <https://s3.amazonaws.com/spark-website/Spark.zip> and download the Windows driver for Spark Core.
2. Put the Core in listening mode by holding the Mode button until the RGB LED starts blinking Blue. The Mode button is the one on the side with the analog pins.
3. Go to device manager, click on Action on the top menu bar, then scan for hardware changes.

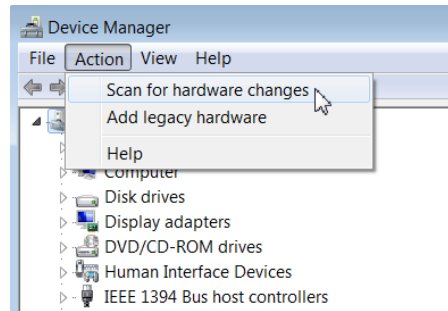


Figure 5-3-4 Device Manager

4. When the 'Other Devices' appears, choose Spark Core with Wi-Fi, right click and update driver software. Then, choose Browse for the driver software.
5. Select driver from the directory that the Spark.zip is stored, then click Next and install. A Windows success driver update will appear which indicates that it has successfully installed the driver. Take note of the number of the COM port.

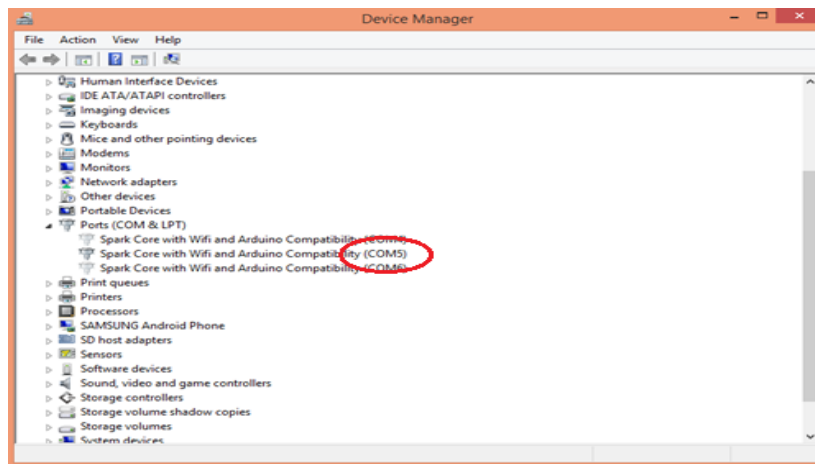


Figure 5-3-5 COM port number

Installation of Putty and Connect Spark Core to Local Network

Before running Putty, the Spark Core has to be plugged into the computer over USB. In order for this to work, the Spark Core need to be in Listening Mode, where the RGB led is blinking blue.

1. Go to <http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html> to download the Putty installer.

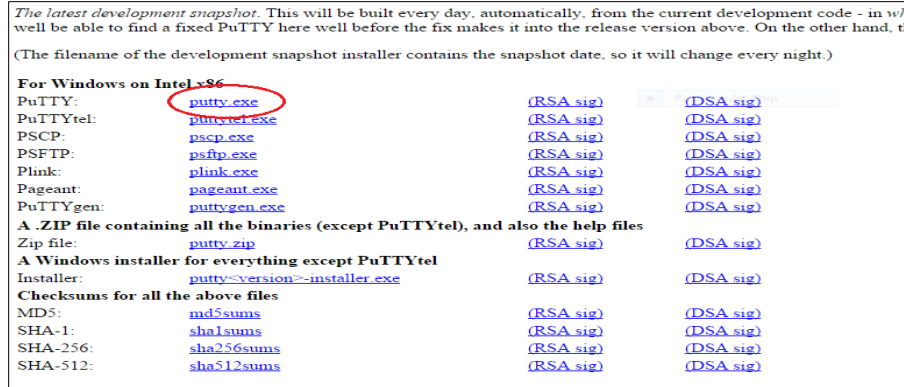


Figure 5-3-6 Putty Installer Download

2. Double click on putty.exe and run to install.
3. When the Core is in Listening Mode, go to serial to open a serial port and set the following standard settings.

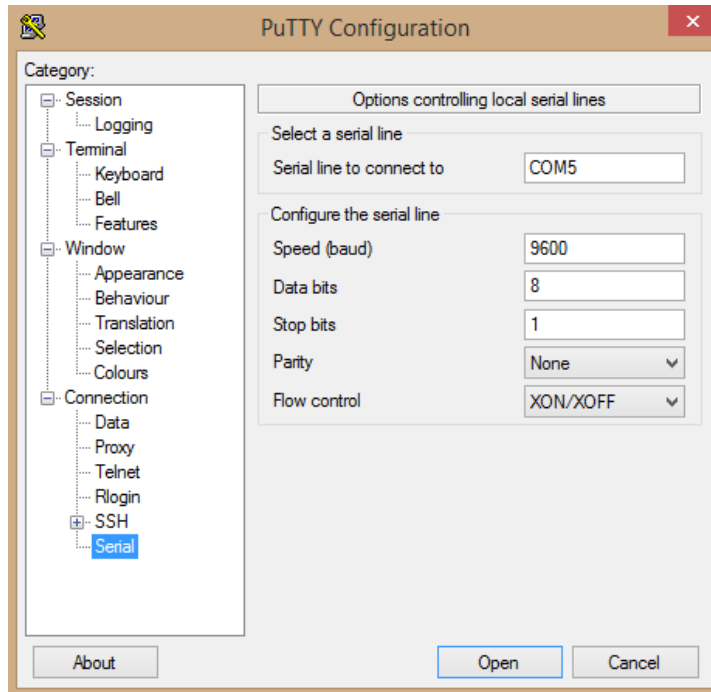


Figure 5-3-7 Putty Configuration

4. Go back to Session. Make sure that the COM port is set correctly and click “Open”.

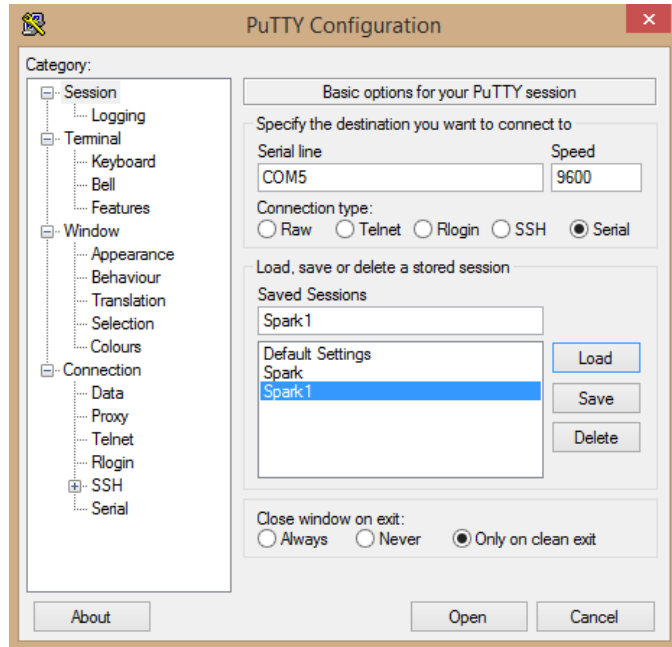


Figure 5-3-8 Open the Putty Configuration

5. Press “i” to display the Core ID. Save this ID as it will be used to claim the Core later.
6. Press “w” to set up the Wi-Fi SSID and password.
7. After SSID has been entered, it will then prompt for security type and password.
8. Once all of the above has been entered, the Core will connect to the cloud where the RGB led will blink green, to cyan and when it is connected to the cloud, it will then be breathing cyan.

Claiming the Core

1. In the web IDE, go to the Core icon on the left pane and click “Add New Core”.
2. A pop-up will appear requesting for the Device ID as shown in the figure below.

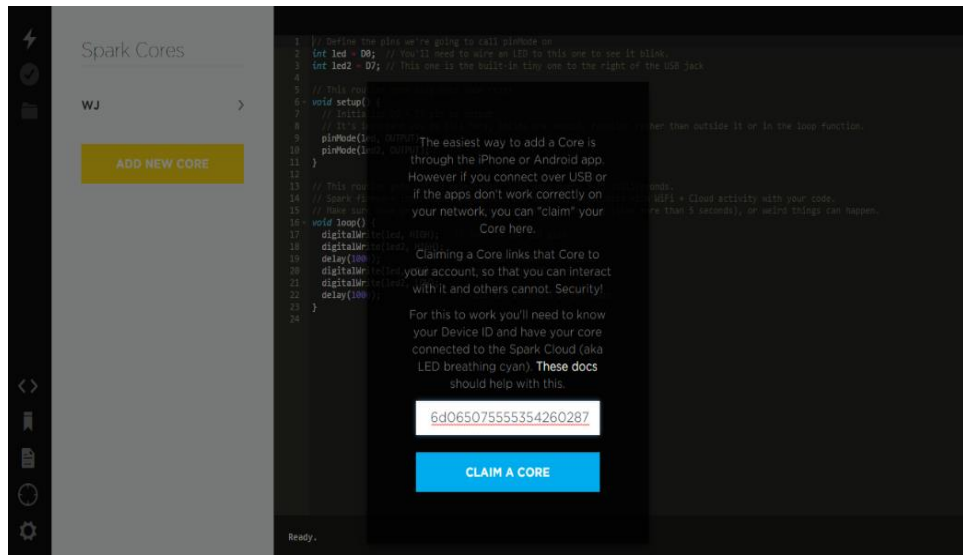


Figure 5-3-9 Claim a Core

3. Key in the device ID that has been acquired earlier and click on “Claim a Core”.
4. A pop-up box will prompt for a name for the newly added Core. Enter a name for the Core.

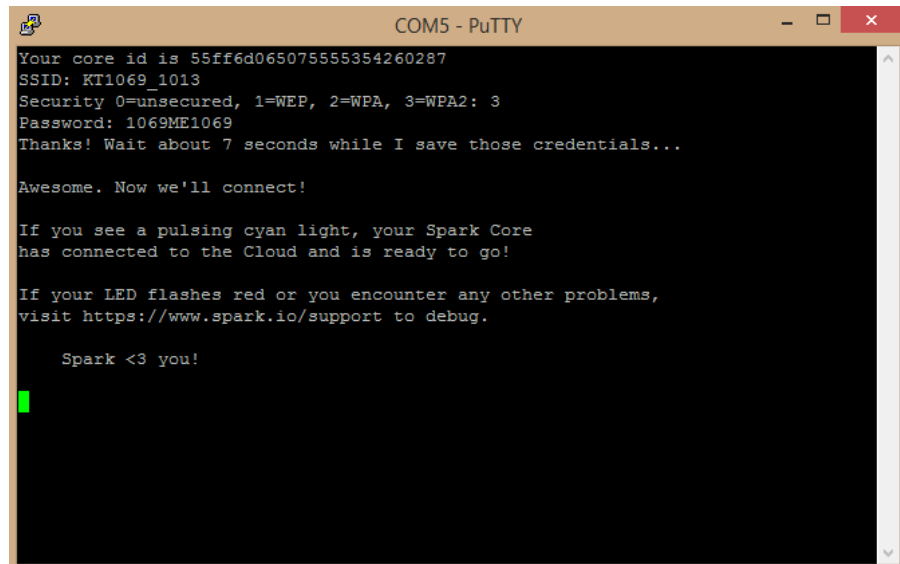


Figure 5-3-10 Connect to Local W-Fi Network

PhoneGap Installation

In order to create an Android application, PhoneGap which is an open source software is used. Eclipse Luna and Android Software Development Kit (SDK) also need to be installed in order to run PhoneGap.

1. Go to <http://www.adobe.com/devnet/archive/html5/articles/getting-started-with-phonegap-in-eclipse-for-android.html> for steps on installing PhoneGap.

Functions

Below are the Core codes required to be flash into the Core in order for it to control the computers.

The Spark.publish() function publishes an event, 'relay-on' through the Spark Cloud that will be forwarded to Cores listening via Spark.subscribe(). The pin number from the browser is converted to integer. When the browser button "ON" is clicked, the Core will set the output of D0, D1, D2 and D3 to high, wait for 2 second and set it to low again. The same goes to when the "OFF" button is clicked.

```
39 int myMain(String command)
40 {
41
42     int state = 0;
43     //find out the pin number and convert the ascii to integer
44     int pinNumber = (command.charAt(1) - '0') - 1;
45
46     // find out the state of the led
47     if(command.substring(3,7) == "HIGH")
48     {
49         state = 1;
50         Spark.publish("relay-on", "ON"); //publish an event to the Cloud
51
52         digitalWrite(led1, HIGH); //set led1-4 to HIGH
53         digitalWrite(led2, HIGH);
54         digitalWrite(led3, HIGH);
55         digitalWrite(led4, HIGH);
56         delay(2000); // wait for 2sec
57         digitalWrite(led1, LOW); //set led1-4 to LOW
58         digitalWrite(led2, LOW);
59         digitalWrite(led3, LOW);
60         digitalWrite(led4, LOW);
61     }
62
63     else if(command.substring(3,6) == "LOW")
64     {
65         state = 0;
66         Spark.publish("relay-on", "ON");
67
68         // write to the appropriate pin
69         digitalWrite(led1, HIGH); //set led1-4 to HIGH
70         digitalWrite(led2, HIGH);
71         digitalWrite(led3, HIGH);
72         digitalWrite(led4, HIGH);
73         delay(2000); // Wait for 2sec
74         digitalWrite(led1, state);
75         digitalWrite(led2, state);
76         digitalWrite(led3, state);
77         digitalWrite(led4, state);
78         return 1;
79     }
80     else return -1;
81 }
```

Figure 5-3-11 myMain Function

The code in Figure 5-3-12 is flashed into the other Spark Core and not the master Core. The Spark.subscribe() function will subscribe to the published event of 'relay-on' that is published by the master Core. Here, the Core will only subscribe to events from only the master Core as indicated by the master Core's device ID.

```
26 - void setup(){
27
28     //subscribe to relay-on function published by the master core
29     Spark.subscribe("relay-on", relayOn, "55ff6e065075555328171787");
30     pinMode(relay1, OUTPUT);
31     pinMode(relay2, OUTPUT);
32     pinMode(relay3, OUTPUT);
33     pinMode(relay4, OUTPUT);
34     delay(2000);
35     digitalWrite(relay1, LOW);
36     digitalWrite(relay2, LOW);
37     digitalWrite(relay3, LOW);
38     digitalWrite(relay4, LOW);
39 }
```

Figure 5-3-12 Setup Function

The events received will be passed to a handler function. A subscription handler takes two arguments, the first being the name of the published event and the second is data that came along with the event.

5.4 System Operation

Once the Core is connected to the cloud, the LED will be breathing cyan which indicates that it is already connected.

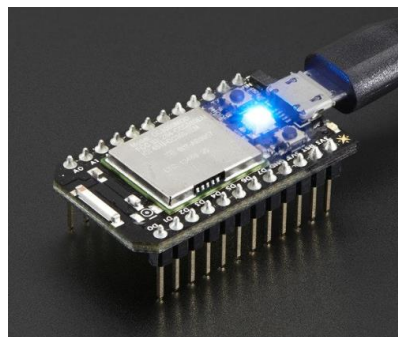


Figure 5-4-1 LED Breathing Cyan

After the firmware is loaded into the cores respectively, the local browser will be used to control the Core via the Internet.

Chapter 5: System Implementation

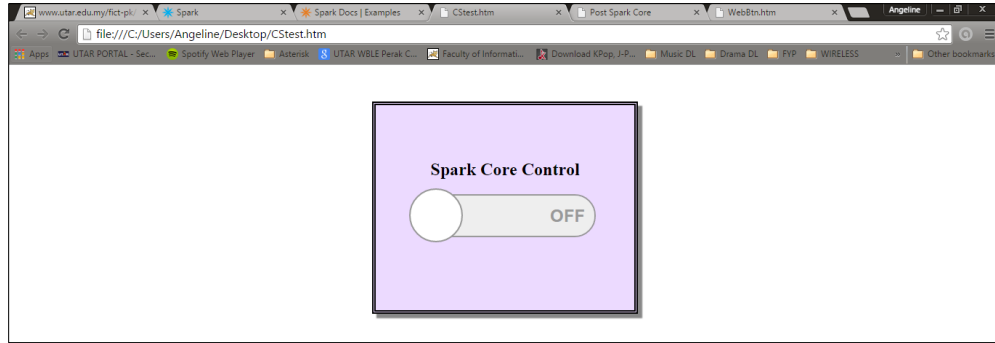


Figure 5-4-2 Local Browser Interface (Off)

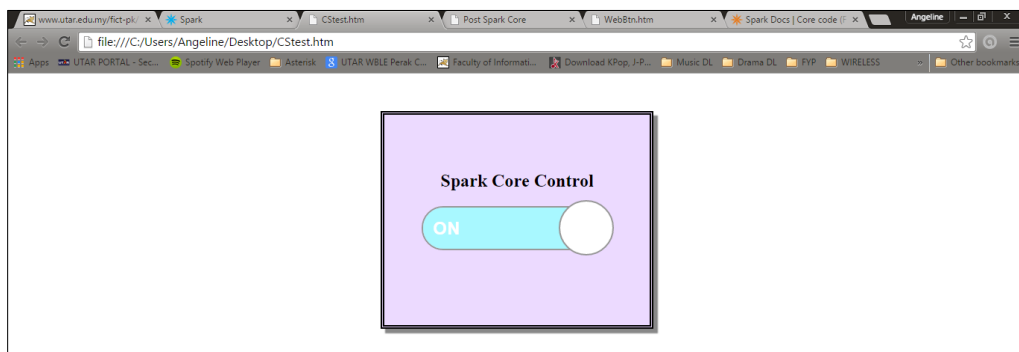


Figure 5-4-3 When Switch is on

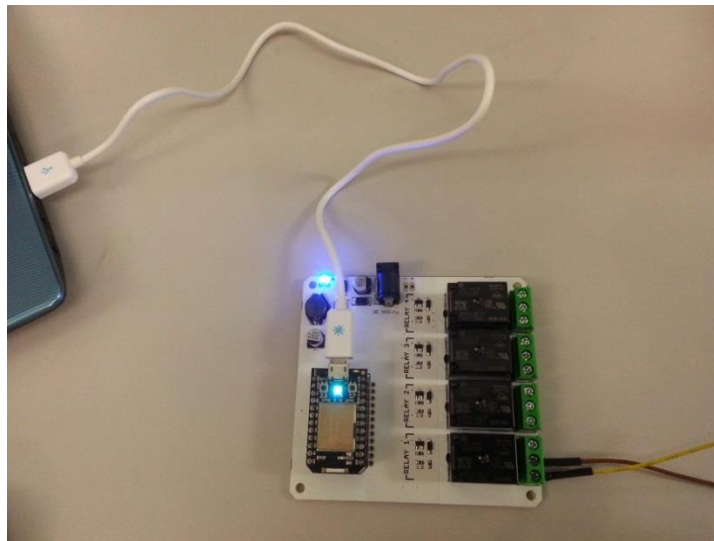


Figure 5-4-4 Hardware Setup with One Computer

The Core will be mounted on the relay shield and is powered on by USB cable to a computer. In Figure 5-4-2, only 1 relay is connected to the lab's computer for a demo purpose. The LED on the relay will light up which will indicate that voltage is flowing

through the relay that will subsequently power on or off the computers. The whole system will work in master-slave architecture where a master core will publish the request to power on the computers and all the other group of cores, which is the slave cores will have to subscribe to the master core's event. This means that if the master core failed to publish its event to the cloud, all the other cores will not be able to power on their computers as well.

5.5 Concluding Remark

The chapter consists of the hardware and software setup that is important in order for the system to operate properly. By setting up and configuring the necessary software and codes into the Core, along with the hardware and software setup, this system will be able to run as intended to perform the function of switching on and off the computers in the lab.

CHAPTER 6: SYSTEM EVALUATION AND DISCUSSION

6.1 System Testing and Performance Metrics

System testing is where the behavior of the system is tested to ensure that it is done within its project scope. It is also a testing carried out on a complete, integrated system to assess the system's compliance with its specified requirements. In this system testing, only reliability testing is being conducted with on and off considered as 1 trial. There will be 10 trials to test out the reliability of this system.

Method: Reliability Testing

Applicable Requirement: Able to switch on and off the computers

Purpose: To calculate the number of successful times the computers managed to switch on and off which is considered as one trial

Item Under Test: Number of successful trial

Precautions: Desktop are functioning well and the electric socket must be switched on

Limitation: This testing is conducted with 2 Cores mounted on relay with 2 computers connected to each relay

Equipment: Desktop, relay shield and Spark Core

Acceptance Criteria: The number of failure trials should not exceed the number of successful trials

6.2 Testing Setup and Result

The table below depicts the number of successful power on and off attempts out of 10 trials conducted. The setup of the hardware is as explained in Chapter 5 section 5.1.

Number of Trials	Success/Fail
1	Success
2	Success
3	Success
4	Fail
5	Fail
6	Success
7	Success
8	Success
9	Success
10	Success

Table 6-2-1 Result of System Reliability Testing

Number of trials: 10

Number of successful trials: 8

Number of failed trials: 2

$$\text{Success Rate: } \frac{8}{10} \times 100\% = 80\%$$

$$\text{Failure Rate: } \frac{2}{10} \times 100\% = 20\%$$

With the failure rate being 20%, it failed to switch on twice out of 10 trials, which is less than the success rate of 80%, hence the system is considered to be reliable. The system will fail to power on the computer when the device is disconnected from the Wi-Fi or when the Wi-Fi connection is unstable but the Core will re-establish its connection to the Wi-Fi once the Wi-Fi is stable again.

6.3 Project Challenges

There will never be a smooth-sailing project without some bumps and hiccups along the way. In this project, several challenges were encountered and eventually were solved or a workaround was found after much researching.

One of the main challenges faced was the relatively new hardware in the market that was being used. In the early stages of this project, little information was known on the workings of the Spark Core and its firmware. Many readings and researching has to be done in order to fully understand the Spark Core and all of its functions. All of the researching can only be done on the Spark forum itself only.

Besides that, for the Android module, initial problem faced was on how to submit the request of the user to the cloud. After much researching, PhoneGap was discovered as an alternative where it is a cross-platform mobile application that enable the HTML5 codes to be converted into an Android application by wrapping it with PhoneGap. This also helps in saving time to code a separate code to run on Android application.

6.4 Objectives Evaluation

The first objective of this project is to develop a solution to switch on and off PCs simultaneously without the conventional pushing of a button. This is accomplished by using the Spark Core connected to the Wi-Fi and being controlled via a web browser and an Android application to power on and off all the computers at the same time.

The second objective is to develop a wireless control system by mounting on a device to a main PC. This objective is achieved when the Core is mounted onto the relay shield which can either be powered on via USB connected to the Core or by using a power adapter to connect it directly to the power input of the relay shield. The core can be controlled to power on 4 computers with one relay shield at a time.

Developing a mobile application as a secondary means to control the core is also part of the objective in this project. As previously mentioned, an Android application was also created to control the core similar to how a remote control would work. The workings of the program flow are identical to that of the web browser where it had to go through the internet to the cloud and back to the core from the mobile device.

The final objective in this project is to test the efficiency of this system in powering up all the computers at the same time. It is also a way to ensure that all the previous objectives have been achieved. Though only reliability testing is conducted for system testing, but it is confirmed that the Spark Core is reliable enough to power on and off the computers in UTAR computer lab.

6.5 Concluding Remark

To conclude, after the system has undergone reliability testing, it is determined that with the success rate of 80%, this system is considered reliable with only twice out of 10 trials where it failed. During the progress of this system, there were several challenges faced which was eventually solved and the objectives stated in the beginning of the report were evaluated to determine if it has achieved all of its objectives. After evaluation, all 4 of the project objectives have been achieved.

CHAPTER 7: CONCLUSION AND RECOMMENDATION

7.1 Conclusion

In a nutshell, the main objective of this project is to lessen the time taken to switch on all the computers in the lab and also to minimize human error in case one or two computers is missed by the lab officer when closing the lab and it will be on standby the whole night by creating a switch to power on and off the computers. With this project, these errors can be minimized. In this project, Wi-Fi technology is used by using the Spark Core as the embedded device in this project. By utilizing the Spark Cloud that comes with the Spark Core, a function in the Core firmware will be exposed through the cloud and can be called remotely using the Spark Cloud API. With this, the Core is controlled by using a local browser and also an Android application to switch the computer on and off via Wi-Fi. Several problems encountered and solved during the project stages also helped in understanding this system and its hardware better. While this hardware may be new during the initial stages of this project, but once the project is nearing its completion, more knowledge and tricks had become increasingly clear.

7.2 Recommendation

While this system may not be perfect, further improvements can be done to develop it further. One of it is to create a selective power on and off of the computers in the lab. This means that the user has the option to select which computer to on and off instead of all the computers are powered on at once. In a way, this will help to save power of the labs in UTAR. Besides that, it is also better if it can be extended further to include the ability to restart the computer and also to put the computer to sleep.

REFERENCES

- Bailey, C. 2014. Introducing: Out of Band Management in Configuration Manager 2007 SP1. *Configuration Manager Writers*, [blog] 21 Jan 2008, Available at: http://blogs.technet.com/b/wemd_ua_-_sms_writing_team/archive/2008/01/21/introducing-out-of-band-management-in-configuration-manager-2007-sp1.aspx [Accessed: 5 Apr 2014].
- Fornazier, H., Martin, A. and Messner, S. (2012). *Wireless Communication : Wi-Fi, Bluetooth, IEEE 802.15.4, DASH7*. [online] Available at: <http://rose.eu.org/2012/wp-content/uploads/2012/03/Wireless-communication.pdf> [Accessed 22 Jul. 2014].
- Istqbexamcertification.com, (2014). *What is Prototype model- advantages, disadvantages and when to use it?*. [online] Available at: <http://istqbexamcertification.com/what-is-prototype-model-advantages-disadvantages-and-when-to-use-it/> [Accessed 17 Jul. 2014].
- ITProPortal, (2010). *Comparison of Various Software Development Life Cycle*. [online] Available at: <http://www.itproportal.com/2010/07/04/comparison-various-software-development-life-cycle/> [Accessed 22 Jul. 2014].
- Jain, D. and Maheshwari, S. (2012). A Comparative Analysis of Different types of Models in Software Development Life Cycle. [online] *International Journal of Advanced Research in Computer Science and Software Engineering*, 2(5), pp. 285-290. Available from: http://www.ijarcsse.com/docs/papers/May2012/Volum2_issue5/V2I500405.pdf [Accessed: 10 Apr 2014].
- Kickstarter. 2013. *Spark Core: Wi-Fi for Everything (Arduino Compatible)*. [online] Available at: <https://www.kickstarter.com/projects/sparkdevices/spark-core-wi-fi-for-everything-arduino-compatible> [Accessed: 10 Apr 2014].
- Koh, R. 2014. *Out Of Band Management*. [online] Available at: <http://www.slideshare.net/imgod2/out-of-band-management> [Accessed: 5 Apr 2014].

References

- Lee, J., Su, Y. and Shen, C. (2007). *A Comparative Study of Wireless Protocols: Bluetooth, UWB, ZigBee, and Wi-Fi*. [online] GUC: Faculty of Information Engineering & Technology. Available at: http://eee.guc.edu.eg/Announcements/Comparaitive_Wireless_Standards.pdf [Accessed 29 Aug. 2014].
- Maheshwari, S and Jain, D (2012) A Comparative Analysis of Different Types of Models in Software Development Life Cycle. 2(5): 285-290.
- Microsoft System Center. 2013. *Introduction to Out of Band Management in Configuration Manager*. [online] Available at: <http://technet.microsoft.com/en-us/library/gg682076.aspx> [Accessed: 20 Mar 2014].
- Moran, J. 2008. *Demystifying a Wireless Network for Small Business Owners - Page 1*. [online] Available at: http://www.smallbusinesscomputing.com/thebasics/networking/article.php/12167_3730281_3/Demystifying-a-Wireless-Network-for-Small-Business-Owners.htm [Accessed: 10 Apr 2014].
- Nagy, A. 2010. *Revolution Wi-Fi: Wake on Wireless LAN*. [online] Available at: <http://www.revolutionwifi.net/2010/11/wake-on-wireless-lan.html> [Accessed: 10 Apr 2014].
- Secor, D. n.d. *How to Turn on TeamViewer Remotely*. [online] Available at: <http://everydaylife.globalpost.com/turn-teamviewer-remotely-42950.html> [Accessed: 24 Mar 2014].
- Spark. n.d. *Spark / Wi-Fi for Everything*. [online] Available at: <https://www.spark.io/> [Accessed: 1 Apr 2014].
- TeamViewer 9 Manual. 2013. *Wake-on-LAN*. [online] Available at: <http://www.teamviewer.com/en/res/pdf/TeamViewer-Manual-Wake-on-LAN-en.pdf> [Accessed: 5 Apr 2014].

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 4
Student Name & ID: Angeline Tai Wei Jing 1102105	
Supervisor: Dr. Goh Hock Guan	
Project Title: Controlling Lab PCs Using WiFi	

1. WORK DONE <ul style="list-style-type: none">• Enhance Chapter 1 - 4
2. WORK TO BE DONE <ul style="list-style-type: none">• Adding flow chart in Chapter 4• Editing system flow
3. PROBLEMS ENCOUNTERED <ul style="list-style-type: none">• Wrong system flow diagram• Diagrams lacked references
4. SELF EVALUATION OF THE PROGRESS <ul style="list-style-type: none">• Understand the basis of the project better

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 6
Student Name & ID: Angeline Tai Wei Jing 1102105	
Supervisor: Dr. Goh Hock Guan	
Project Title: Controlling Lab PCs using WiFi	

1. WORK DONE

- Further polished Chapter 1- 4
- Creating a local HTML document to control the Core

2. WORK TO BE DONE

- Enhancing the web interface and connectivity

3. PROBLEMS ENCOUNTERED

- Connectivity issue connecting to the Cloud thus not able to control the Core

4. SELF EVALUATION OF THE PROGRESS

- Didn't understand well enough on the mechanism of the Spark Core

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 8
Student Name & ID: Angeline Tai Wei Jing 1102105	
Supervisor: Dr. Goh Hock Guan	
Project Title: Controlling Lab PCs Using WiFi	

1. WORK DONE <ul style="list-style-type: none">• Managed to get the Core to function in a master-slave manner• Control the Core via web browser
2. WORK TO BE DONE <ul style="list-style-type: none">• Android app• Report Chapter 5
3. PROBLEMS ENCOUNTERED <ul style="list-style-type: none">• Communication issues between 2 set of cores
4. SELF EVALUATION OF THE PROGRESS <ul style="list-style-type: none">• Consult and discuss with supervisor how to let 2 cores communicate with each other

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 10
Student Name & ID: Angeline Tai Wei Jing 1102105	
Supervisor: Dr. Goh Hock Guan	
Project Title: Controlling Lab PCs Using WiFi	

1. WORK DONE <ul style="list-style-type: none">• Core-to-core has a master-slave architecture• A part of Chapter 5
2. WORK TO BE DONE <ul style="list-style-type: none">• Android app
3. PROBLEMS ENCOUNTERED <ul style="list-style-type: none">• Unsure of which software to use for user-friendliness• Android app to communicate with Spark Cloud
4. SELF EVALUATION OF THE PROGRESS <ul style="list-style-type: none">• Understand Android better• Guidance from supervisor helps in clarifying the issue with the Android connectivity

Supervisor's signature

Student's signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project I / Project II)

Trimester, Year: Y3S3	Study week no.: 12
Student Name & ID: Angeline Tai Wei Jing 1102105	
Supervisor: Dr. Goh Hock Guan	
Project Title: Controlling Lab PCs Using WiFi	

1. WORK DONE <ul style="list-style-type: none">• Chapter 6 and 7• Browser interface and Android design
2. WORK TO BE DONE <ul style="list-style-type: none">• Finalize on the report• Finalize on the system interface
3. PROBLEMS ENCOUNTERED <ul style="list-style-type: none">• System design
4. SELF EVALUATION OF THE PROGRESS <ul style="list-style-type: none">• Understand PhoneGap• Guidance from supervisor helps in finalizing the report

Supervisor's signature

Student's signature