ENERGY EFFICIENT SMART LAMPPOSTS FOR SMART CITIES

By

Ajay Kumar Doshi

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

In partial fulfilment of the requirements

for the degree of

BACHELOR OF INFORMATION TECHNOLOGY (HONS)

COMMUNICATIONS AND NETWORKING

Faculty of Information and Communication Technology

(Perak Campus)

JANUARY 2016

DECLARATION OF ORIGINALITY

I declare that this report entitled "**ENERGY EFFICIENT SMART LAMPPOSTS FOR SMART CITIES**" 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 : ___________________________

Date : ___________________________

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Dr. Goh Hock Guan for his guidance, advice and opinions throughout the entire course of my FYP1 and FYP2. Without his knowledge and insight into wireless sensor networks, this project would not have been possible.

I would also like to thank my family for their encouragement and motivation while I was preparing for this report. Their backing certainly made me work harder to produce a good system.

Last but not least, my gratitude goes to my course mates who shared their knowledge and helped review my report.

ABSTRACT

The current street lighting systems in Malaysia rely on timer switches to turn on and off streetlamps. Besides that, they mostly use sodium-vapour lamps as the light source. This leads to significant energy wastage because streetlamps are turned on when not required, and the usage of sodium-vapour lamps require frequent replacement; leading to higher maintenance costs.

The proposed system will allow streetlamps to rely on sensors to turn on or off their lamps. This way, the system can provide its basic functionality while at the same time conserving energy by either dimming or turning off the lights completely when there are no users interacting with the street lighting system. The system will use an array of LEDs to replicate a streetlamp, and also incorporate the use of light sensors and motion sensors. A GUI is also designed to provide an accurate visual representation of the system. Besides that, a routing algorithm is designed and incorporated into the system which allows nodes to automatically find their parent and reroute in case the current one is down.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

3G	Third Generation
4G	Fourth Generation
ADSL	Asymmetric Digital Subscriber Line
CdSe	Cadmium selenide
EMI	Electromagnetic Interference
FHSS	Frequency Hopping Spread Spectrum
GPS	Global Positioning System
GUI	Graphical User Interface
HB	High Brightness
I/O	Input/Output
ISM	Industrial, Scientific and Medical
LED	Light Emitting Diode
LOS	Line-of-Sight
LTE	Long Term Evolution
NLOS	Non-Line-of-Sight
PIR	Passive Infrared
PWM	Pulse Width Modulation
RFI	Radio Frequency Interference
SSL	Solid-State Lighting
TCP/IP	Transmission Control Protocol/Internet Protocol
USB	Universal Serial Bus
Wi-Fi	Wireless Fidelity
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network

Chapter 1 - Introduction

1.1 Motivation and Problem Statement

The implementation of timer-based lampposts leads to energy wastage and are also inefficient. Turning on and off lampposts on fixed times each day can not only incur huge costs but also fail to serve the purpose of lampposts. For example, a street lamp is programmed to turn on at 7pm in the evening but the environment is still relatively bright. Besides that, pedestrians and road users are not able to rely on lampposts for navigation and safety during certain times of the day. The timing of sunset and sunrise is different throughout the year and therefore a fixed approach for a dynamic problem is simply inefficient. A lamppost which is turned on for the entire night, at maximum brightness for a quiet road with an average of 5 cars passing through it is a tremendous waste of resources. In efforts to reduce energy consumption and light pollution, lampposts should be energy efficient and dynamic in the sense that they can incorporate power saving features, and are able to turn on or off based on different scenarios. An effective solution to this problem should not only provide lighting based on the brightness of the environment but also reduce energy consumption by only being used when needed.

The motivation for this project arises from the vision to implement an effective and energy-efficient street lighting system. The lighting system is efficient in the sense that it can relay information such as a malfunction to an administrator rather than relying on feedback from users, which are typically reported a few hours after a malfunction. In this case, the few hours of downtime affect the other users as well, in terms of pedestrian safety and road traffic. Drivers tend to slow down in darker areas due to low visibility, significantly delaying their journey, and pedestrians on the other hand would hesitate to walk through a dark road.

In addition to that, the main purpose of this project is also to reduce the electrical power requirements and maintenance cost of lighting systems. Current street lighting systems incorporate photocells, which are susceptible to “dust or dirt build-up, improper installation or calibration” (Lavric et al., 2012) which causes malfunction and compromises the system. The implementation of LED lamps in favor of sodium-vapor lamps will give rise to tremendous energy savings. Energy consumption alone will be reduced by 80%, not ignoring the fact that LED lamps last 50 times longer than traditional light bulbs (Müllner and Riener, 2011) which will lead to lower maintenance costs. The implementation of this system will be seen as a step taken to a more effective lighting system. It will also encourage other parties to adopt a greener approach towards the environment, and raise awareness regarding the disadvantages brought by sodium light bulbs currently deployed in most parts of Malaysia. This will either directly or indirectly give rise to smart cities installed with energy efficient streetlamps.

1.2 Project Objectives

- **Employ the use of sensors in a lighting system to allow lampposts to be turned on only when required, based on sensor readings.**

  The system works by receiving input from 2 sensors, namely the light sensor and motion sensor. The light sensor returns its value to the nodes periodically and once the value falls below the threshold value, the lampposts are turned on using average brightness. When the motion sensor detects motion, the lampposts turns on with maximum brightness, as long as the motion sensor continues to detect motion. Otherwise, after a specified amount of time, the lampposts return to their previous state.

- **Provide a GUI which simplifies the management and administration of the street lighting system**

  The GUI used for this project will illustrate the connectivity between the nodes in real-time as well as allow administrators to remotely control the lampposts. The GUI also incorporates a security feature whereby any new node attempting to join the ad-hoc network would require explicit authorization from the administrator.

- **To demonstrate a reliable solution for an energy-efficient lighting system which can be used in Smart Cities**

  The existing systems are not attention-free as they require user feedback to determine the status of the lampposts. Through this system, however, the status of the lampposts as well as other raw data such as sensor readings are continuously sent to administrators in real-time. Administrators then can control the lampposts manually and turn them on or off, if required. Besides that, the system is designed with the motorists' and pedestrians' needs in mind. Once detected by motion sensors, the path which is travelled by them is lit up instead of the entire street or neighbourhood, saving valuable resources. The motion sensors could also be installed in desired safety zones which are prone to snatch thieves or are frequently associated with crime.

1.3 Project Scope

The ultimate deliverable of this project is a smart street lighting system which uses a WSN to control street lamps. Each node is placed on top of existing street lamps and is directly connected to the light source. The light source used in this project is a group of LEDs, which allow the brightness of a street lamp to be varied by changing the number of active LEDs. Each node would be connected to either a set of streetlights or a motion sensor. The light sensor, on the other hand, transmits periodically to the base station and is used to detect the luminosity of the environment. After the luminosity value falls below a specified threshold value, the node can instruct all other nodes to switch on their respective light sources with average brightness. The second variable that triggers the nodes to switch on their lights is motion. For example, on a straight road, two nodes are retrofitted with motion sensors with one on each side of the road. When motion is detected, regardless of pedestrian or vehicular movement, the node instructs all other nodes in the same series to turn on their lights. These lights are left on for a predefined amount of time as long as motion is continuously detected, and then are either dimmed or turned off completely.

In this project, all nodes report to a common base station. They send data such as luminosity value, battery voltage, LED status, temperature and motion status. The base station, usually connected to a desktop computer, displays the data from all the nodes and also stores these values into a database for historical or accounting purposes. Through this, the nodes in the system can be monitored and faults can be detected without physically observing the street lighting system. For example, a node which already records a low luminosity value but does not turn on its lights regardless of motion is clearly malfunctioning. This will allow the administrators to rectify it easily by identifying the faulty node, based on its ID. Other data such as battery voltage allow the administrators to change the power source of nodes before they are completely depleted, thereby aiding in minimizing the downtime of the system.

Each node requires a 51-pin interface board to be connected to the array of LEDs and the motion sensor. The compatible light sensor board is directly connected to the MICAz node via the 51-pin I/O connector located on the node, hence an interface board is not needed. Therefore, this project consists of 3 sets of nodes, the first being nodes connected to a motion sensor, secondly the nodes connected to a light sensor and thirdly nodes connected to LEDs. All 3 sets are used in this project to form an array of nodes. The first set is usually placed on the edge of roads to detect approaching vehicles, but can also be placed towards the centre to detect pedestrians. Nodes from the first set instruct all nodes in an array to switch on their light source upon detecting motion. The second set of nodes will only be required for each area as its sole function is to detect the luminosity. The third set comprises of the most nodes used in this system, where they will be acting as slaves to the master nodes from the first set.

On top of that, a GUI which displays the parent/child relationship between the nodes will be designed. This GUI also allows the toggling of nodes manually, for testing the connectivity from the base station to the target node. The GUI also incorporates a security feature, whereby whenever a new node attempts to join the network, it would require explicit authorization from the administrator first. Other features include graphical icons representing battery levels and different coloured lines to depict RSSI.

1.4 Main Contributions from the Project

Upon the conception of this project, an infrastructure of a street lighting system will be developed. This infrastructure can be used as a base to connect and configure multiple nodes in a single WSN. The main target audience of this project would be state governments as they would be in charge of implementing lighting systems throughout the country. At the end of the day, a wide scale implementation of this project will help make the realization of a Smart City possible.

1.5 Organization of the Report

This report consists of 7 chapters, in addition to the references and appendix. The first chapter is the Introduction, where the motivation and scope of the project is detailed. The main contribution is also explained.

The second chapter consists of a literature review of 4 different systems which are similar to this project. The systems were analyzed and their advantages, disadvantages and critical comments were summarized into a table. Besides that, the technologies which could be used for this system were also reviewed and the justification of the chosen technology was provided.

The third chapter entails the methodology used for the development of the system. Four different development models were considered and the chosen model was justified in terms of suitability with the system. The hardware, software and functional requirements were identified and the project milestone for both FYP1 and FYP2 were shown. The estimated costs for development and commercialization were tabulated to provide a rough idea on how much the system would cost.

The fourth chapter illustrates the architecture of the system and also identifies its functional modules. Flowcharts of these modules as well as the system flow are provided. The design of the database and GUI are shown in addition to a pseudocode of the routing algorithm.

The fifth chapter describes the hardware and software setups that are needed for the system as well as the steps required to set up and configure it. This chapter also includes screenshots of the system in operation.

The sixth chapter shows the tests carried out in order to measure the performance of the system. The results are tabulated in addition to several screenshots which show the results of the testing. Besides that, the project challenges were identified and the objectives from chapter 1 were evaluated.

The final chapter provides a conclusion of the system in terms of what was produced and how it can be used to increase energy savings compared to traditional systems. Besides that, several weak points of the system were identified and mentioned for future recommendation of improvement.

Chapter 2 – Literature Review

2.1 Review of Technologies

The idea behind the implementation of a Smart Lamppost can be achieved by a variety of existing technologies. Some have significant advantages over others and hence each should be considered carefully to meet the scope of the project. Several technologies are compared such as ZigBee, WiMAX, Wi-Fi and Bluetooth.

ZigBee is an open wireless technology global standard which was developed to address the needs of low-cost and low-power networks. It operates on the IEEE 802.15.4 standard and operates on unlicensed bands which include 2.4 GHz, 900 MHz and 868 MHz (Digi.com, 2015). Multiple topologies are supported such as point-to-point or mesh networks which allow for wider ranges, depending on the application needs. With a theoretical limit of 65,536 nodes per network (Telecomabc.com, 2005), ZigBee is the selected wireless technology that will be used in this project.

Bluetooth is a type of WPAN technology used mainly for short distance communication. Based on the IEEE 802.15.1 standard, it uses the unlicensed ISM band at 2.4 to 2.485 GHz (Bluetooth.com, 2015). Depending on the three classes of radio, Bluetooth devices can support ranges from 1 meter to 100 meters. It avoids collisions on the congested 2.4 GHz frequency by using FHSS which allows communications between Bluetooth devices to occur with minimal interference. However, a Bluetooth device can only connect to a maximum of 8 devices, which makes it infeasible for this project as it significantly limits the number of streetlamps within a system.

Wi-Fi is one of the most common wireless technology used in today's standards. Based on the IEEE 802.11 standards, it is the most popular solution deployed in home networks. One key advantage of Wi-Fi is that it supports frequencies of 2.4 GHz and 5 GHz (using the 802.11ac standard) with bandwidths up to 54 Mbps and 1300 Mbps respectively (SpeedGuide, 2014). This will allow the smooth transmission of heavy content such as high-definition video and games. However, it does not meet the scope of this project as the control messages used will only require a fraction of the supported bandwidth of Wi-Fi. Hence the use of Wi-Fi in this project will be unnecessary as the high speeds will require high power usage as well.

WiMAX is a wireless communications standard designed to provide connectivity for Wireless Metropolitan Area Networks based on the IEEE 802.16 standard. It supports both LOS and NLOS channel conditions which greatly improves the supported range for communication. WiMAX has been considered as one of the leading technologies for 4G networks, apart from LTE. An ideal WiMAX network is capable of reaching ranges from 80 to 90 kilometres (Engineersgarage.com, 2012), which will be very useful for deployment in cities. However, through this the power consumption is also generally higher, and this would result in nodes operating using WiMAX to require frequent replacement for their power source.

Technology	ZigBee	Bluetooth	Wi-Fi	WiMAX
Underlying Standard	802.15.4	802.15.1	802.11	802.16
Application	Monitoring & Controlling	Device to device interaction	Web, Email, Video, Games	Broadband access, Voice, Video
No. of supported nodes	65535	8	32	512
Power usage	Lowest	Low	High	Highest
Bandwidth	250 kbps	720 kbps	11 – 1300 Mbps	10 Mbps
Range (m)	1 – 100	1 – 100	35 – 250	50 000
Network Architecture	Star, Tree, Mesh	Star	Star	Point-to-Multipoint, Mesh
Optimized for	Reliability, Scalability, Low cost and power	Convenience, Low cost	Speed	Speed, Scalability

Table 2-1-T1 Comparison between ZigBee, Bluetooth, Wi-Fi and WiMAX

Based on the table above, it is clearly justifiable that the ZigBee radio suits the scope of this project perfectly. The different types of supported architectures as well as the large number of supported nodes make it ideal for installing this system on streetlamps within a city. The low power usage of the radio coupled with the maximum bandwidth supported goes in line with the main operations of the system, which is to transmit control messages to nodes while requiring minimum maintenance. Therefore, the usage of MICAz which use ZigBee radios provide a significant advantage, apart from the fact that the interface board on the node will directly allow the use of the sensors in the system, namely the light sensor, as well as the motion sensor via a 51-pin breakout board.

2.2 Review of Existing Systems/Applications

2.2.1 Alexandru Lavric, Valentin Popa, Ilie Finis, Daniel Simion

The system proposed by Lavric et. al. uses nodes based on the JenNet 5148 transceiver coupled with light sensors attached to ZigBee nodes to form a WSN local area network. Each local area network consists of a master node. It controls the slave nodes and determines whether their lights should be turned on or off, as well as allows communication between other networks. Each master node is connected to an e-Box Ethernet/WSN gateway, to allow remote access from a Command Center PC. The e-Box is connected to the master node via a USB to RS232 communication interface, and to the command centre via Internet using Ethernet cables. The lamps used in this system are 60 Electromax lamps with 20 HB LED light bulbs each. The dimming mechanism used in Pulse Width Modulation which allows the light intensity to be controlled in a range from 50% to 100%.

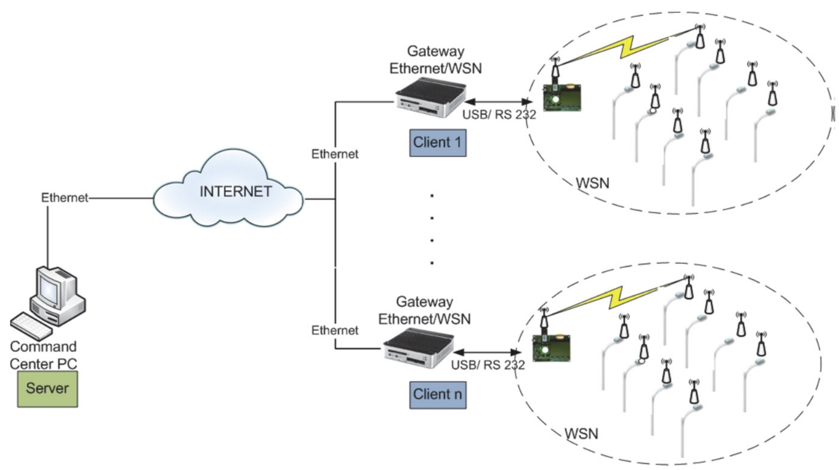

Figure 2-2-1-F1 System Architecture of Lavric et. al.

Each node in the system reads the luminosity level in the surrounding environment whereas the master node detects the presence of vehicles using an additional sensor. Apart from this, the master node also selects the radio channel used for communication, initialize and allow nodes to join the network, enables the transfer of data from the WSN network to the client application stored in the e-Box and transfers commands initiated by the server. The implementation of the Ethernet/WSN gateway allows for the support of a client-type application. The operations of the nodes involve establishing a network connection to the master node, followed by ensuring network connectivity. If either of these were to fail, then the nodes enter a standalone mode where they turn their lamps on or off depending on the luminosity of the environment. Otherwise, the nodes monitor messages received from the master node and periodically read sensor values, check lamp status and forward information to the master node.

2.2.2 Pedestrian aware Smart Street Lighting (Reinhard Muellner and Andreas Riener, 2011)

Müllner and Riener designed a Smart Street Lighting system in 2011 mainly for rural residential areas which directly involves the users into the system. The system proposed entails lampposts embedded with ZigBee motes, a central SSL server, and a base station acting as an intermediary between the server and the motes and finally end user devices such as smartphones. Smartphones are used with their GPS capabilities to periodically transmit the location of the users to the system. Users can set the size of the area to be lit via an application. The larger the size, the longer the lights will be kept on.

The system addresses privacy concerns raised with the continuous polling of the user's location to the server. All data exchanged between the user and the server is fully anonymized by using a randomly generated ID for user identification. Data integrity check mechanism is also enforced to prevent unauthenticated access. This is also reinforced by using an authentication key for communication between the user and the server. Furthermore, the system does not save a user's history of usage but rather overwrites it in the next cycle. Negative acknowledgment is incorporated as well, primarily to reduce bandwidth consumption.

The implementation of an application to control the areas to be lit would first require a large number of users using the application before it gains traction. However, the implementation of such a system would least likely be suitable in Malaysia as it requires a constant Internet connection. According to Tradingeconomics.com the percentage of population covered by mobile cellular network in Malaysia is 92% in 2010. In Germany, however, the same statistic in 2010 stood at 99% (Tradingeconomics.com, 2015). This means that if the system were to be deployed in Malaysia, it would be unavailable in certain areas. Not only that, the idea of using an application before illuminating a particular street is time consuming. The far more effective solution would be to use sensors as it does not require initiation from users. Moreover, the paper states that the LED lamps will be dimmed when there is no pedestrian activity but fails to explain what mechanism will be used to achieve this. By allowing the user to control the duration for the lights to be on, energy efficiency cannot be guaranteed as the user can specify a longer time than actually required. In the event that the user has already walked through a street within the specified time, the lights are still powered on which fails to satisfy the provisions of an energy saving lighting system in the first place.

2.2.3 A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX (Fabio Leccese, Marco Cagnetti and Daniele Trinca, 2014)

In 2014, Leccese, Cagnetti and Trinca propose a street lighting isle based on ZigBee Sensor Network, Raspberry-Pi Card and WiMAX. The main idea behind this system was to implement an isle of lampposts in an area which could not be reached by typical ADSL lines and 3G signals. The lampposts are connected to a ZigBee mote in a mesh topology which transmit to a Raspberry-Pi control unit. A WiMAX connection is used in the grid to control and manage the entire network from a remote station.

One of the advantages of this system is that it is very organized in the sense that it employs a hierarchical structure layer made up of sensors, actuators and local control cards (Leccese, Cagnetti and Trinca, 2014). In the first layer, called the “apparata layer”, the monitoring stations consisting of the nodes transmit to the Local Control Unit Layer via ZigBee communication.

The radio module used in this system is the XBee radio module by MaxStream. The authors claim that apart from having a high radio sensitivity which will result in a lower probability to receive corrupted packets, the module used also has a very low power consumption, with 50mA consumed when connected to a 3V DC source. In sleep mode, the current draw is less than 10µA. However, the MICAz mote to be implemented in this project can draw far less power than this in the same condition. It typically consumes 19.7mA in normal operating mode and during sleep, the power consumption is reduced to a mere 1µA (Datasheet of Mica-Z Wireless Measurement System, 2005). Thus implementing the XBee radio module will not only be less energy efficient but also will require frequent maintenance to replace the power source.

2.2.4 Street Lighting Control System Based On Large-Scale WSN (Alexandru Lavric, Valentin Popa, Stefan Sfichi, 2014)

Another system proposed by Lavric, Popa and Sfichi in 2014 describes a street lighting monitoring and control system which consists of a control centre, gateway modules and control modules. The JenNet protocol is used for local communication between the nodes and gateway, whereas for long-distance communication via a control centre the TCP/IP protocol is used. The proposed system also incorporates video surveillance systems already installed in certain municipalities into the monitoring system. Since the cameras send information through a TCP/IP network, the gateway module can be added to this network, allowing simultaneous control of the lamps as well as a live feed of the system in effect. The control centre allows the street lights to be turned on and off remotely, or can be configured to follow a timetable to allow a more dynamic approach in controlling the lights.

The usage of the JenNet protocol allows the system to meet the criteria of a high performance lighting system, such as large number of supported nodes, reduced complexity for development and lack of additional licence costs (Lavric, Popa and Sfichi, 2014). The user interface proposed in this system also allows for a more convenient way of remote control and monitoring. The same also applies for maintenance as the user interface displays the lamp ID and GPS coordinates if a malfunctioning lamp is detected. The dimming mechanism leads to a reduced energy consumption and the periodic reports entailing energy consumption details aid in maintenance.

However, the usage of PWM as the dimming mechanism brings about a lot of disadvantages, such as vulnerability to EMI and RFI (Maxwell, Cacan, Haile, 2013). The inclusion of a video feed in the TCP/IP network will also require a connection with a high bandwidth. Apart from that, the system uses a lot of additional hardware components, such as Doppler sensors for the detection of vehicles, a switch to allow connectivity of the TCP/IP network and an e-Box 4300 coupled with a USB/RS232 coordinator for the gateway module. This, together with the cost of integrated circuit packages for PWM increases the cost of the system significantly. This would create a negative perception that street lighting systems are costly and deter organizations or municipalities from installing smart street lighting systems.

2.2.5 Summary of Existing Systems

	Alexandru Lavric, Valentin Popa, Ilie Finis, Daniel Simion	Pedestrian aware Smart Street Lighting	A Smart City Application: A Fully Controlled Street Lighting Isle	Street Lighting Control System Based On Large-Scale WSN	Proposed System
Vehicle Detection Feature	Yes	No	Yes	Yes	Yes
User Interaction	Indirect, sensor based	Direct, mobile application	Indirect, sensor based	Indirect, sensor based	Indirect, sensor based
Radio Module	JN 5148	MICAz	XBee	JN 5148	MICAz
Lamp Brightness Control	PWM	N/A	None	PWM	Reduce number of active LEDs in array

Table 2-2-5 T1 Comparison of Existing Systems

2.3 Concluding Remark

The MICAz mote is used worldwide in a variety of applications such as indoor building monitoring and security, high speed sensor data such as acoustic, video and video as well as large scale sensor networks, supporting more than 1000 nodes per network (Datasheet of Mica-Z Wireless Measurement System, 2005). Its key features such as support for large networks, low power consumption and easy implementation for applications makes it a better choice for implementation in wireless sensor networks.

Most of the compared systems use an indirect approach with the user by relying on sensors to determine the functionality of the system. If users were to be interact with the system directly, it would defeat the purpose of having a Smart Street Lighting system. The feature of being able to detect vehicles is certainly one of the key functionalities of the system. The radio module used in this system certainly proves to be the best choice among others such as JN 5148 or XBee, due to a number of better features such as power consumption, cost-effectiveness and so on. The lamp control mechanism used in this system is distinct from the others, as it varies the number of LEDs which are on to vary the brightness emitted. This mechanism makes use of the processing capabilities of the MICAz mote, by programming the nodes to allow them to balance between power consumption and effectiveness of the system. This system instils a high reliability on the motes to determine the brightness of the lamp according to the sensor readings. Overall, this reduces the possibilities of errors to occur in the system because the brightness is not controlled by any intermediate device.

Therefore, the proposed system would be best implemented in Smart Cities as opposed to others due to its low cost, ease of configuration and low maintenance cost. It will not require users of the system to manually turn on the lights like the Pedestrian Aware Smart Street Lighting, and will not be susceptible to EMI and RFI as it does not use PWM unlike the system proposed by Alexandru Lavric, Valentin Popa, Ilie Finis, and Daniel Simion, or the Street Lighting Control System Based On Large-Scale WSN system. The interoperability of MICAz with other sensor boards or interfaces via the 51-pin interface gives it a valuable advantage over other systems and allows for future growth by increasing the functionalities of the system.

Chapter 3 – System Methodology

3.1 System Development Models

There are various types of system development models which can be used in this project. The Prototype model, Waterfall model, Agile model and V-model were evaluated based on their strengths and weaknesses as well as their suitability for this project.

3.1.1 Prototype Model

The Prototype model involves building, testing and reworking an early approximation of a final system or product until an acceptable prototype is achieved (Rouse, 2015). This model is well suited for projects whose requirements cannot be known in detail ahead of time.

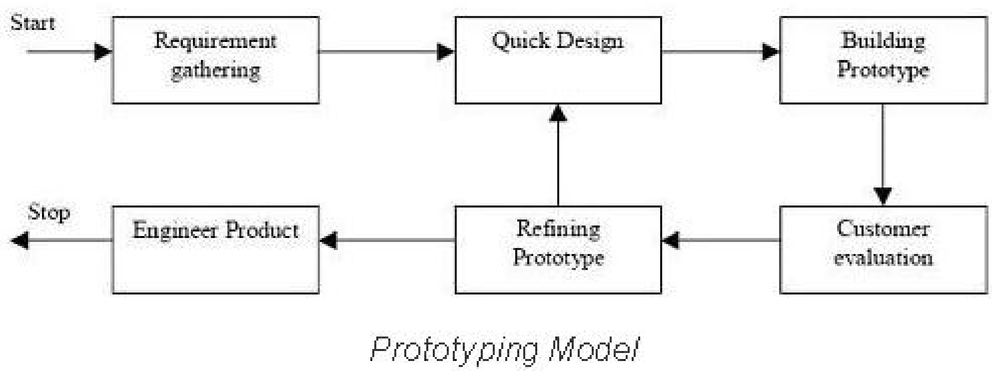

Figure 3-1-1 F1 Prototyping Model (ISTQB Exam Certification, 2015)

One of the key advantage of this model is that since a working model of the system is being provided, users will be able to obtain a better understanding of the system. This will also allow any possible errors to be detected much earlier, before the system is actually finalized. This minimizes the chances of discovering an error much later on, thus saving time from fixing errors at the final stages. However, this model might also be impractical in some cases where the complexity of the system may increase if the scope of the system is expanded beyond the original plans (Istqbexamcertification.com, 2015). This might lead to the end product not completely satisfying the requirements of the system.

3.1.2 Waterfall model

The Waterfall model is one of the most frequently used system development methodology. Each phase in the waterfall model requires the previous to be completed before the next can begin, and there can be no overlapping in the phases

(Tutorialspoint.com, 2015). At the end of each phase, the progress is reviewed to determine if the project is on track and whether or not to carry on with development or discard it completely (Istqbexamcertification.com, 2015).

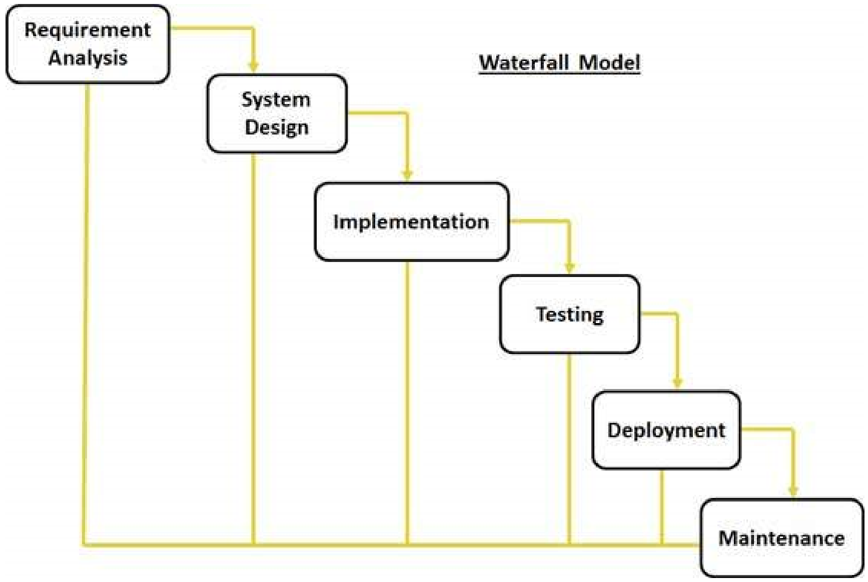

Figure 3-1-2-F1 Waterfall Model (tutorialspoint, 2015)

This model is best suited for systems which have their requirements clear, fixed and well-defined, such as small projects. One of the advantage of the waterfall model is that it is simple to use and easy to understand. It also aids in building a system systematically, step-by-step. The use of the waterfall model comes with several disadvantages, such as the fact that it involves a high amount of risk and uncertainty. It is also not suitable for projects which are ongoing, or have a long duration to complete. Also, any project in the testing phase which requires review will result in difficulties when reverting to the previous stage.

3.1.3 Agile model

The Agile model involves a combination between the "iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product" (tutorialspoint.com, 2015). In this model, the tasks are separated into an iterative approach, with each iteration having its own Testing, Planning, Requirements Analysis, Designing and Building phase. An iterative would typically last between 2 to 3 months, and at the end of each iterative a working software build is produced.

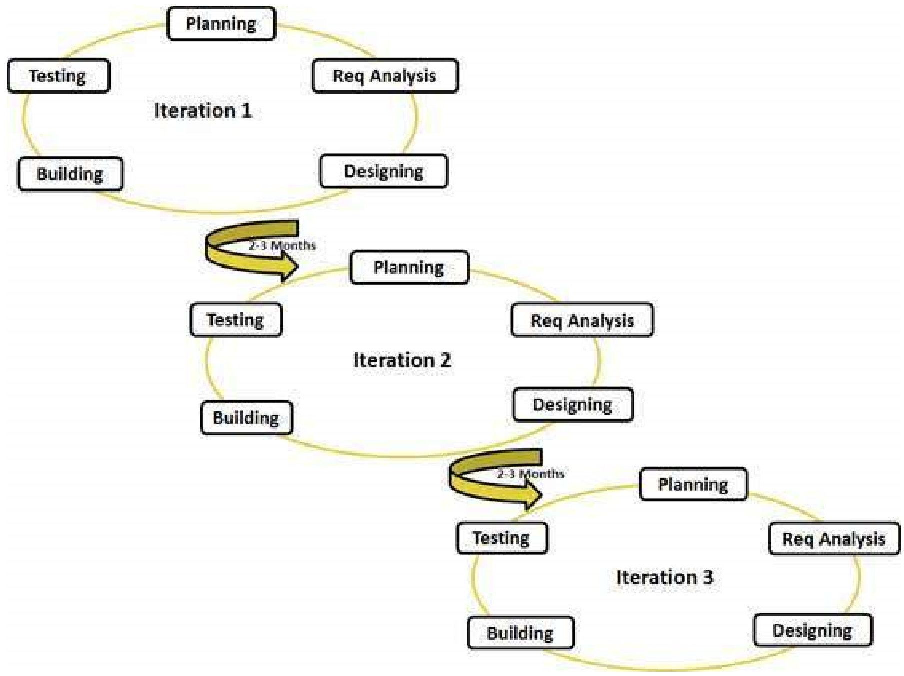

Figure 3-1-3-F1 Agile Model (tutorialspoint, 2015)

Each working build is released with an increasing number of features, with the final build holding all the required features according to the requirements analysis. The agile model is advantageous as it is accorded as a realistic approach in software development. Besides, the delivery of a working build after each iteration allows partially working solutions to be released early. However, the usage of the agile model introduces the risk of sustainability, maintainability and extensibility (tutorialspoint.com, 2015). The model is also not apt for complex projects, as it would be difficult to produce early working models after each iteration. Also, the model lacks any support for documentation, hence if the details of the system were to be transferred to different individuals, such as business partners or new team members, the task might be challenging.

3.1.4 V-Model

The V-Model, also known as the Verification and Validation model, is a model where the execution of the processes happens in a sequential order, following the V-shape (tutorialspoint.com, 2015). Like the Waterfall model, the V-Model also requires each phase to be completed prior to the next one being started.

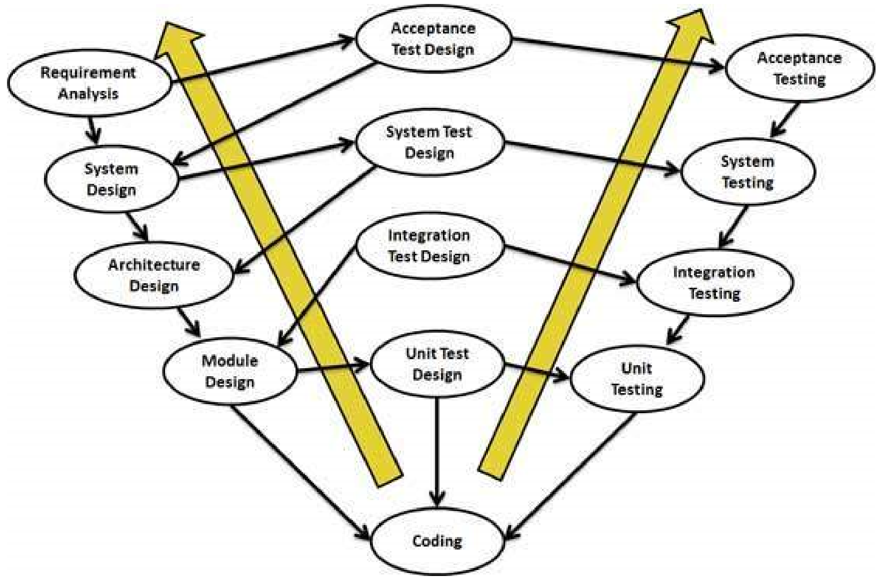

Figure 3-1-4-F1 V-Model (tutorialspoint.com, 2015)

Based on Figure X, it can be seen that the testing phase (right) is planned in parallel with a corresponding development phase (left). The testing phase is known as the validation phase, whereas the development phase is known as the verification phase. The coding phase is where the actual coding is carried out after the system modules have been designed at the design phase. V-Model provides several advantages, for example it is suited for smaller projects with well-defined requirements. It is also a highly disciplined model which requires each phase to be completed one at a time. One of its clear disadvantages is that the software is only developed during the implementation phase, hence no early prototypes of the software is produced (Istqbexamcertification.com, 2015). This might not be suitable under certain situations where the project has to produce a prototype to keep the clients updated. Furthermore, any updates that occur midway through the development will require all test documents as well as requirement documents to be updated.

3.1.5 Selected Model

Upon much consideration and comparison, the Prototype model was selected for this project. Unlike the V-Model, the Prototype model is suitable for smaller projects. Besides, with a prototype, a proper clarity and feeling of the general functionality of the software can be obtained. This will allow any changes or modifications after that to be implemented before fully developing the final system. This model also reduces risk of failure as the risks are identified early on, hence mitigation steps can be taken. Although it is a slow process in general, the advantages certainly outweigh the disadvantages in terms of this system as a fully functional system can be developed, after improving on the prototypes, with minimal errors and maximum efficiency.

3.2 System Requirements

This project will employ the use of MICAz nodes attached with either a sensor board or a 51-pin breakout board to allow the node to control the motion sensor or LEDs. A separate node attached to a desktop computer will act as a base station. The software components include TinyOS and Java Development Kit.

3.2.1 Hardware

MICAz MPR2400 Mote

The node uses the Chipcon CC2420, which is a radio frequency transceiver compliant with IEEE 802.15.4 standards and ZigBee-ready. It is coupled with an Atmega128L micro-controller, serial flash memory and also a 51 pin I/O connector, which will be used to connect to the LED.

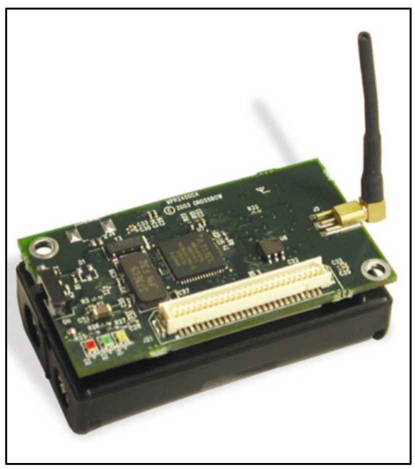

Figure 3-2-1-F1 MICAz Mote (Crossbow, 2007)

MTS101CA Sensor Board

This sensor board consist of a precision thermistor, a CdSe photocell and a general prototyping area. However, for this project, only the photocell will be used to record the luminosity of the surroundings.

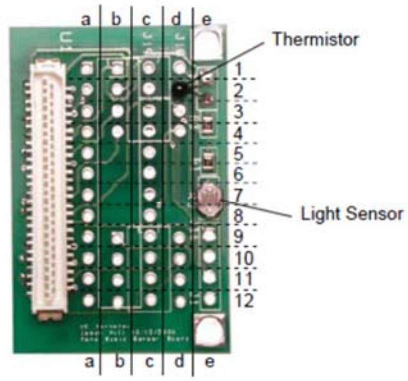

Figure 3-2-1-F2 Top view of sensor board (Crossbow, 2007)

MIB520 Programming Board

The programming board is used to program code into the node using the Atmega16L in-system processor. Mainly used for communication and in-system programming, it can also supply power to attached nodes so that they can function without batteries. For the purpose of this project, the node attached to the board will be used as a base station, which will be receiving packets transmitted from the other nodes.

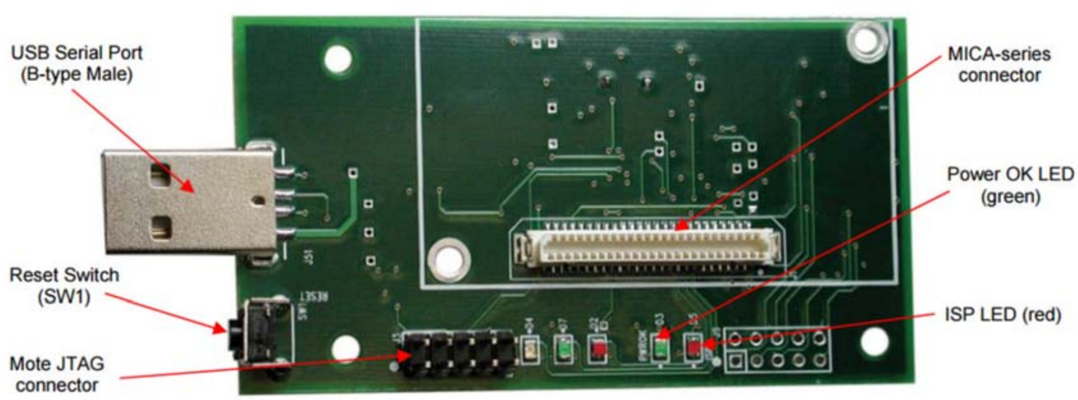

Figure 3-2-1-F3 Top view of MIB520 programming board (Crossbow, 2007)

51-Pin Breakout Board

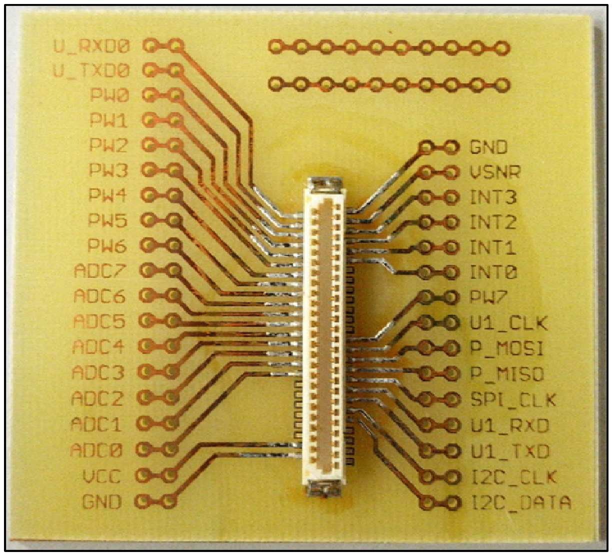

Figure 3-2-1-F4 – Overhead view of breakout board

The breakout board will be used to connect either the motion sensor or LEDs, depending on the type of node. It allows the output to be directly manipulated by the node by specifying which pins are connected to the attached module.

PIR Motion Sensor

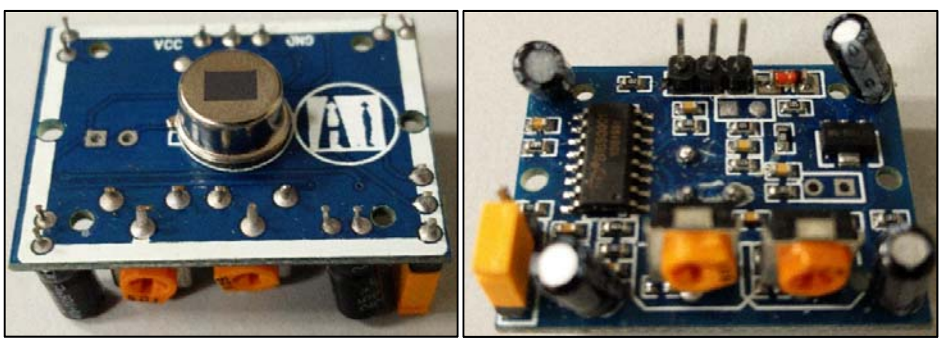

Figure 3-2-1-F5 – Top and bottom view of motion sensor

A PIR sensor can be used to detect movement such as movement by either humans or vehicles. It can be powered by any voltage between 5V to 16V and has a sensing angle of less than 100 degrees. The motion sensor's sensitivity for distance or output delay can be adjusted via the on-board potentiometers.

4x AA Battery Holder

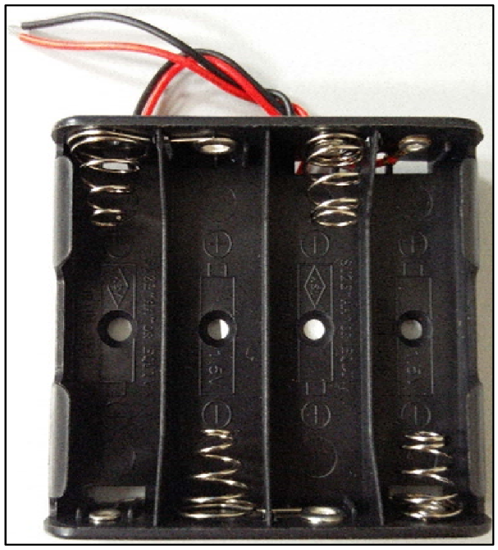

Figure 3-2-1-F6 – Battery holder with positive and negative wires

Since that the motion sensor requires a higher voltage than what can be provided by the MICAz alone, an external battery holder is to be connected to the motion sensor and node to provide sufficient power for the motion sensor to operate. Four AA batteries provides a steady voltage of 6V which is within the acceptable range of supplied voltage for the motion sensor.

LEDs

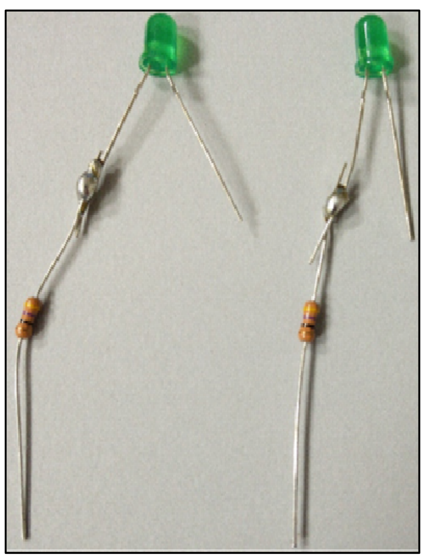

Figure 3-2-1-F7 LED diodes soldered with 50 Ohms resistor

A simple diode with its anode and cathode will be connected to the node via the breakout board. This will allow the node to directly toggle the LED. The LEDs are soldered with a resistor at the positive terminal to prevent a high voltage from damaging the MICAz node.

Desktop Computer

A desktop computer will be used to connect to the programming board so that the nodes can be programmed, and the data obtained from the base station can be stored in the database located in the computer.

3.2.2 Software

TinyOS 2.0.2 running on Oracle VirtualBox

TinyOS is an operating system designed mainly for low-power wireless embedded systems. It has a collection of drivers for microcontrollers and Integrated Circuits.

nesC language

A dialect of the C language, nesC is used to write application in TinyOS.

JDK Version 1.8.0_73

JDK will be required to display output received by the base station. The development of the GUI will be based on the Java language, which includes the process of inserting data into the database. The GUI will include several complex graphical components which cannot be displayed by the default JDK installed in the TinyOS, which is of an older version.

Database (Wamp Server)

A Wamp server will be formed in the system where the MySQL database is installed on the Windows 7 operating system, and an Apache server will be used to display data on a webpage using PHP. The base station attempts to insert data to the database each time it receives output, by specifying the IP address of the host PC running the VirtualBox.

Eclipse Java Mars

Although the development of the project is done in a Linux environment, the implementation of the GUI and database connectivity in carried out separately on a Windows environment using an IDE. This is because the XubunTOS used in the project does not come installed with any IDE, hence the task of writing Java codes might prove to be difficult. The codes are developed using Eclipse and then copied into the XubunTOS to be executed.

3.3 Functional Requirement

Wireless Network (ZigBee)

A WSN implementing a master-slave architecture is one of the functional requirements of this system. The base station node will act as the first master node of the system, and its slaves will act as the master nodes for the subsequent nodes. This applies to all nodes and hence a multihop wireless network is formed. The successful implementation of this multihop network will ensure that all nodes are connected and information can be transmitted from the base station to all the other nodes in the network, and vice-versa. The transmission of packets between nodes should also implement some sort of acknowledgment protocol. This is so that any packets unsuccessfully received by the receiving node can be retransmitted again by the sender. It is a basic yet pivotal characteristic of wireless sensor networks.

Routing Protocol

With the exception of the child nodes of the base station, all the other nodes in the network do not have a fixed parent. In the event of a connection loss, the nodes are able to re-route to the nearest node which has a parent. This connection is maintained until the keep-alive connection, which is implemented by the use of an acknowledgement timer, disconnects and the node proceeds to find a new parent.

Light Sensor Node

The proposed system requires at least one node per area to detect the luminosity of the environment. This light sensor node relays the raw light sensor values to the base station. The base station then periodically disseminates the raw value to all nodes in the network. The light sensor node is not connected to a light source as the interface pin of the node is already connected to the light sensor. For each area, the threshold value for the light sensor must be set. This is because certain areas might require a higher or lower threshold, so that the lampposts are able to turn on and off depending on the environment they are installed in. For example, roads and highways will have a lower threshold value so that the lamppost can be turned on to provide better clarity to road users, compared to lampposts in residential areas which are not as sensitive to changes in luminosity.

Motion Sensor Node

Behaving as the master node in the system, the motion sensor node must be able to instruct all its slave nodes in the area to toggle their LEDs based on the received light sensor value and the presence of motion. If the received value is low, then all nodes will be lighted up with average brightness only; it is only when the value is low and motion is detected where all the lamps will be lit up with maximum brightness. The motion sensor node also transmits information to the base station.

3.4 Project Milestone

Task	**Project Week**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Data collection														
Define project objective and scope														
Analysis for literature review														
Determine system and functional requirements														
Define technologies involved														
Determine system development model														
Outline system architecture														
Outline system flow														
Create Database and Algorithm design														
Presentation														
Documentation														

Table 3-4-T1 Gantt chart showing the project milestones (FYP1)

Task	**Project Week**													
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Implement base station and sensor node source codes, begin development of GUI														
Consolidating the use of flash memory and buffer for all nodes as well as development of routing algorithm														
Test system scalability by increasing number of sensor nodes														
Finalizing design and implementation of database as well as Java GUI														
Finalize system for presentation														
System testing and performance														
Presentation														
Documentation														

Table 3-4-T2 Gantt chart showing the project milestones (FYP2)

3.5 Estimated Cost

Equipment	Quantity	Price Per Unit (RM)	Price (RM)
MICAz Mote (MPR2400)	14	Supplied by UTAR	-
Programming Board (MIB520)	1	Supplied by UTAR	-
Light Sensor Board (MTS101CA)	1	Supplied by UTAR	-
PIR Motion Sensor	6	Supplied by UTAR	-
4x AA Battery Holder	6	Supplied by UTAR	-
51-pin Breakout Board	12	Supplied by UTAR	-
LED	36	Supplied by UTAR	-
AA Batteries	26	Supplied by UTAR	-
Computer	1	Supplied by UTAR	-
TinyOS, Java, nesC	1	Supplied by UTAR	-
Total			0.00

Table 3-5-T1 Estimated Cost for FYP Development

Equipment	Quantity	Price Per Unit (RM)	Price (RM)
MICAz Mote (MPR2400)	14	600.00	8400.00
Programming Board (MIB520)	1	590.00	590.00
Light Sensor Board (MTS101CA)	1	157.00	157.00
PIR Motion Sensor	6	10.00	60.00
4x AA Battery Holder	6	2.00	12.00
51-pin Breakout Board	12	50.00	600.00
LED	36	0.50	18.00
AA Batteries	26	2.00	52.00

Computer	1	1500.00	1500.00
TinyOS, Java, nesC	1	Freeware	-
Total			11,389.00

Table 3-5-T2 Estimated Cost for Commercialization

The estimated cost for commercialization only serves as an approximation and should not be considered as the actual cost for implementation.

3.6 Concluding Remark

The prototype model selected for the development of this system will help ensure that the final deliverable is a working, error-free solution thanks to the continuous improvement over the created prototypes. The identified system and functional requirements will help ensure that the project is kept on track. The execution of the system testing and performance will help reduce the risk of the system failing under unforeseen circumstances. The testing will also help verifying the correctness and efficiency of the project. The project milestones aid in the estimation of time taken to complete the project report in addition to ensuring the system testing is carried out within the specified time period. The costs for development and commercialization were also estimated to give a rough idea on how the system would be implemented in actual Smart Cities.

Chapter 4 – System Design

4.1 System Architecture

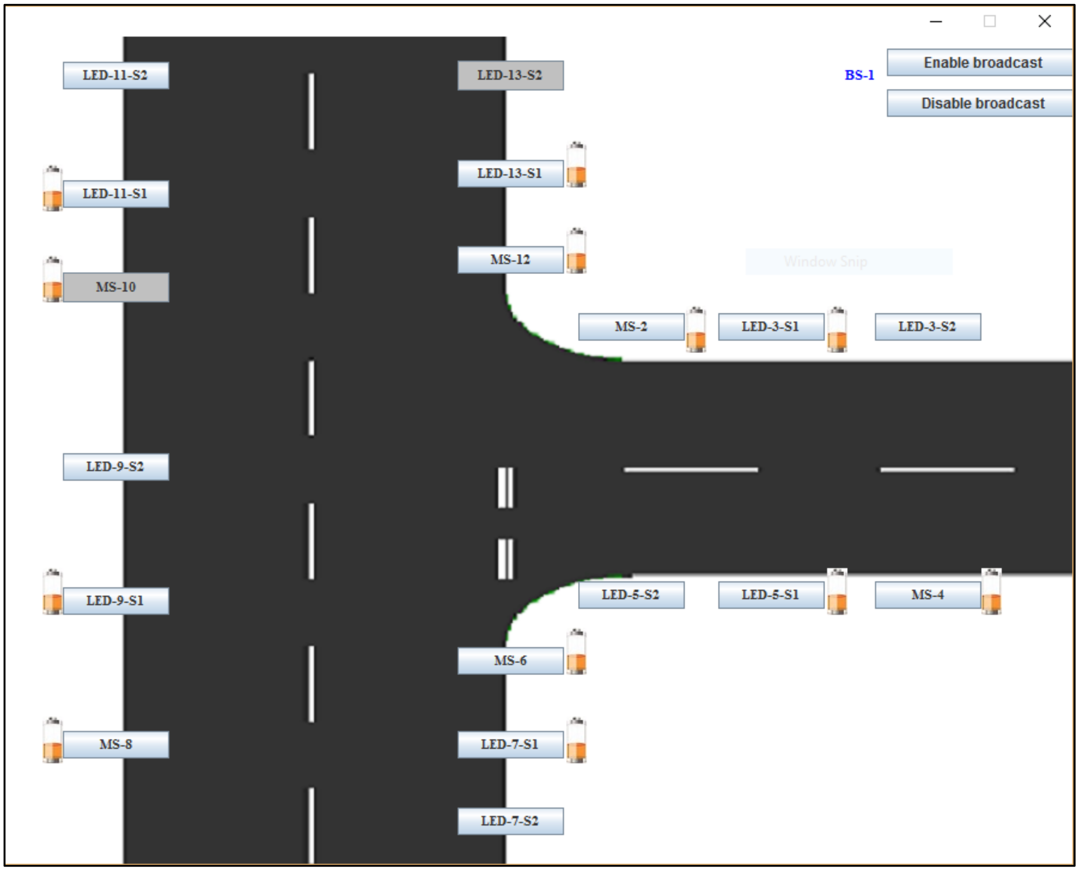

Figure 4-1-F1 – Overview of system architecture

The system architecture depicts a T-junction. The base station is connected to a computer and acts as an intermediary between the administrator and the system. Connected to the programming board via USB, it will receive the output from all other nodes via the attached node on the programming board. A Java GUI will be displayed on the desktop computer to show the relationship between all the nodes as well as the battery level of each node.

The motion sensor nodes are installed strategically such that motion is able to be detected on junctions rather than only straight roads. Whenever motion is detected, a motion icon is displayed on the GUI indicating which lane the motion was detected. The light sensor node, required only for each area, will read and transmit the light sensor value to the other nodes. The motion sensor nodes will be delegated with the task of making decisions whether to turn on or off the lamps. This decision will be carried forward to slave nodes of the motion sensor. Each motion sensor has a fixed LED slave node which controls 2 sections of the road. The reason for this implementation is to remove the need of an extra node that merely controls the LED lights. By consolidating the control of 2 sections of the road to one node, the number of nodes required for the system can be reduced. If motion is detected by the motion sensor, only its child LED nodes will turn on. Doing so helps reduce energy wastage by cutting down energy consumption where not required.

The light sensor node shown in the figure are not connected to any light source and as such, its only purposes are to share the light sensor values with other nodes. The motion sensors and LED nodes on the other hand are retrofitted with a 51-pin interface board where the LEDs or the PIR motion sensor are connected.

4.2 Functional Modules in the System

4.2.1 Base Station Module

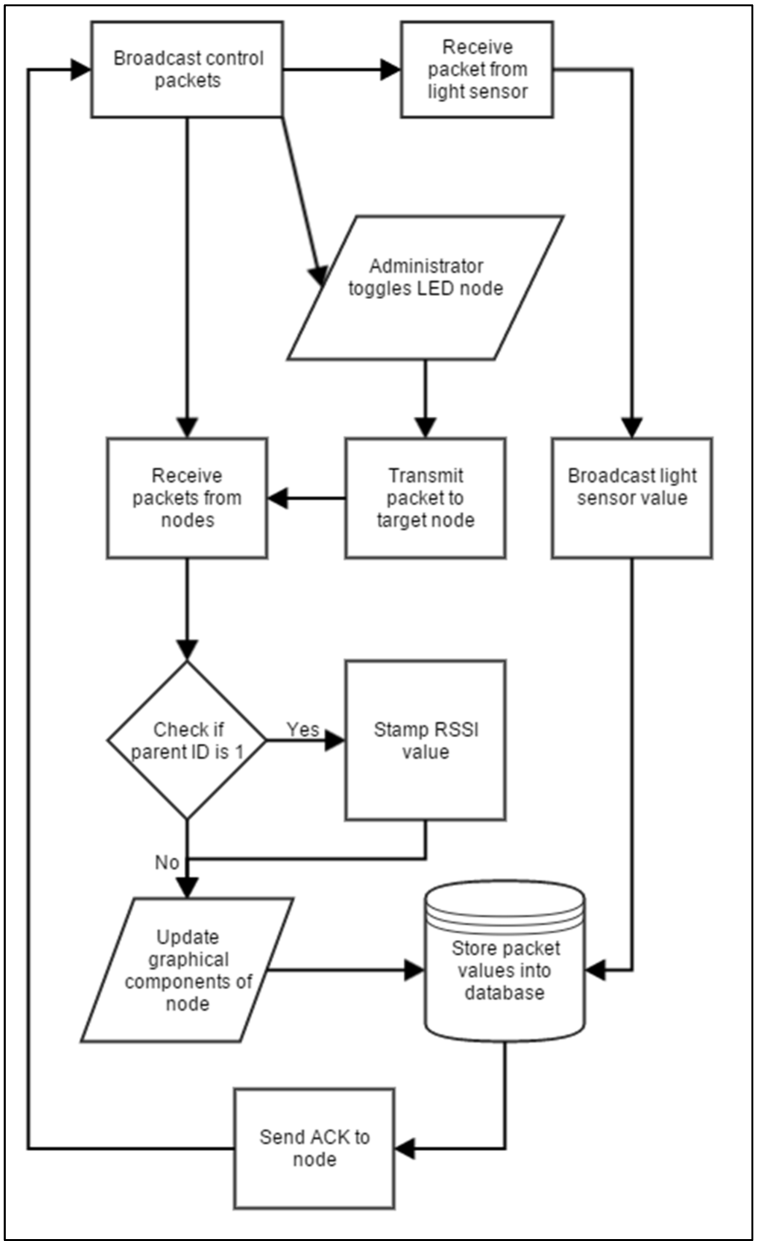

Figure 4-2-1-F1 – Flowchart of base station

The base station continuously listens for packets transmitted by the nodes. The packets include details regarding specific nodes such as node ID, sensor values and battery voltage. As the system is started, each output received from the nodes are attempted to be inserted into a local database. In the event that this fails, an error message is displayed on the GUI. The data stored on the database can be retrieved for accounting and maintenance purposes. If the base station were to fail, then the output from the nodes cannot be retrieved. The base station also periodically broadcasts control packets to connect to master motion sensor nodes. Without these master nodes, the packets from all other nodes cannot be received by the base station. The transmitting power of the base station is reduced so that only the nearest nodes are able to pair with the base station to form a parent-child relationship, thereby reducing the workload on the base station. Besides that, it also receives the luminosity value from the light sensor and transmits it to all other nodes in the network. The base station is also used to remotely toggle LED nodes from the GUI. Once a user action is received, the base station transmits a packet to the target LED node, and updates the colour of the icon of the target node.

4.2.2 Motion Sensor Module

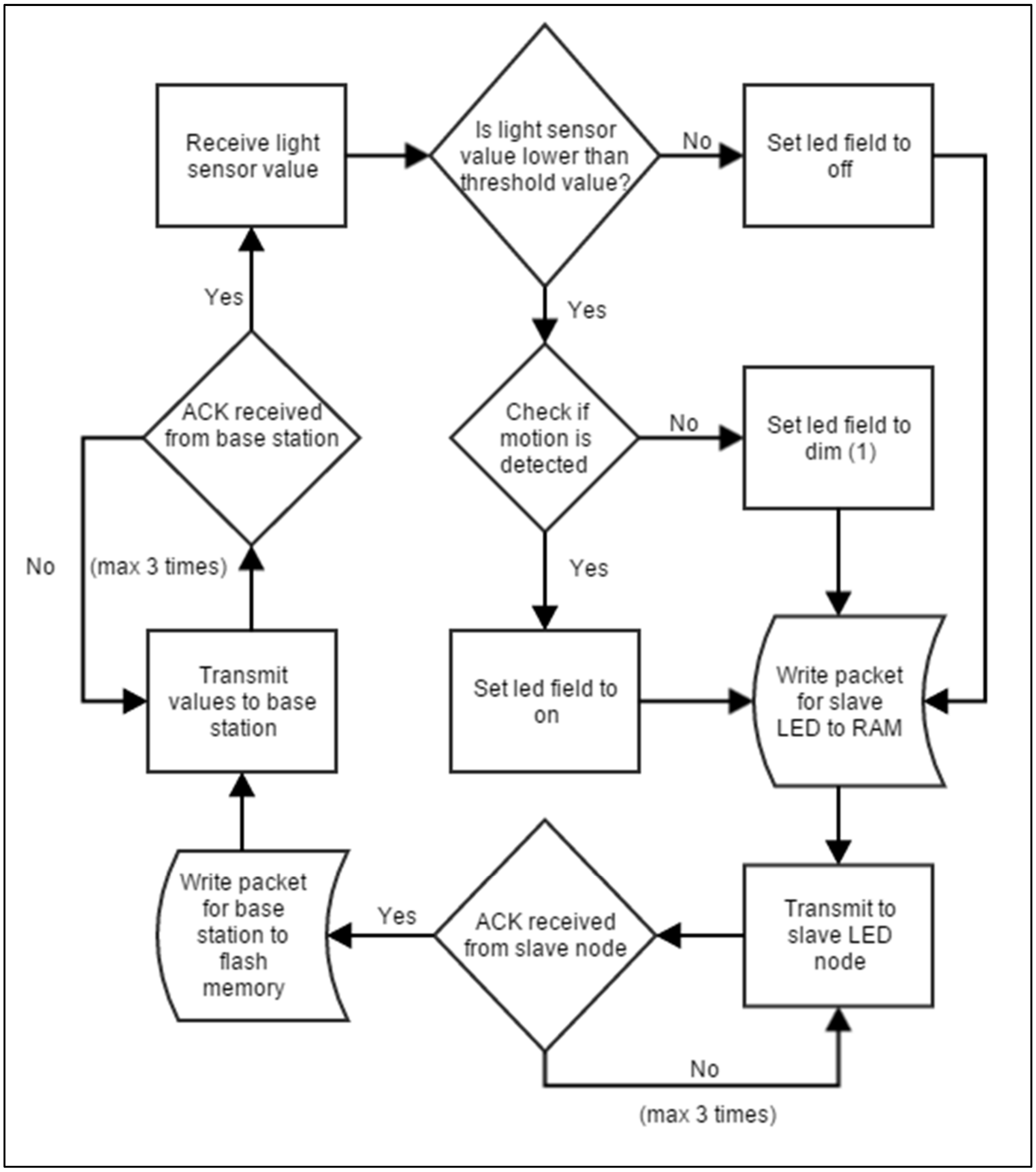

Figure 4-2-2-F1 – Flowchart of motion sensor

The motion sensor node is one of the most important functional modules of the system. It receives the light sensor value and controls the on/off function of all lamps within its section. Once the light sensor value is known, the node checks for motion at a specified time interval. By doing this, the processing overhead is reduced by completely skipping this step if the light sensor value is high. Once motion is detected, the slave LEDs are turned on. Only in the event that motion is not being detected will the LEDs return to their average brightness. The node transmits its raw values to the base station by storing the packets in the flash memory first and the waiting for an ACK.

If an ACK is received, the next packet in the buffer in transmitted. Otherwise, if the ACK is not received within a predefined amount of time, the packet will be retransmitted twice. For packets meant for the LED node, they are stored in the flash memory as it is faster to write to compared to the flash memory. The toggling of the LED should be as fast as possible and hence the RAM is used as a buffer instead of the flash memory.

4.2.3 Light Sensor Module

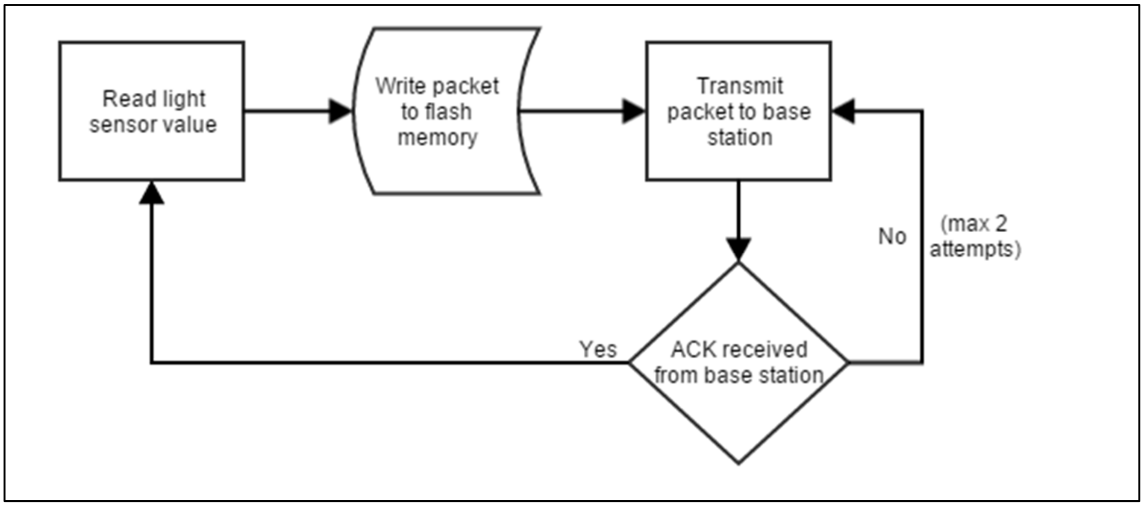

Figure 4-2-3-F1 – Flowchart of light sensor

The functional module of the light sensor node shows the simplicity of its functionality, yet the light sensor is one of the most important modules of the system. The system does not turn on the LEDs in the event of motion alone, as the luminosity value must be lower than the threshold value as well. The light sensor periodically transmits to the base station by writing the packet to the flash memory first, and then retransmitting twice if an ACK is not received.

4.2.4 LED-connected Node Module

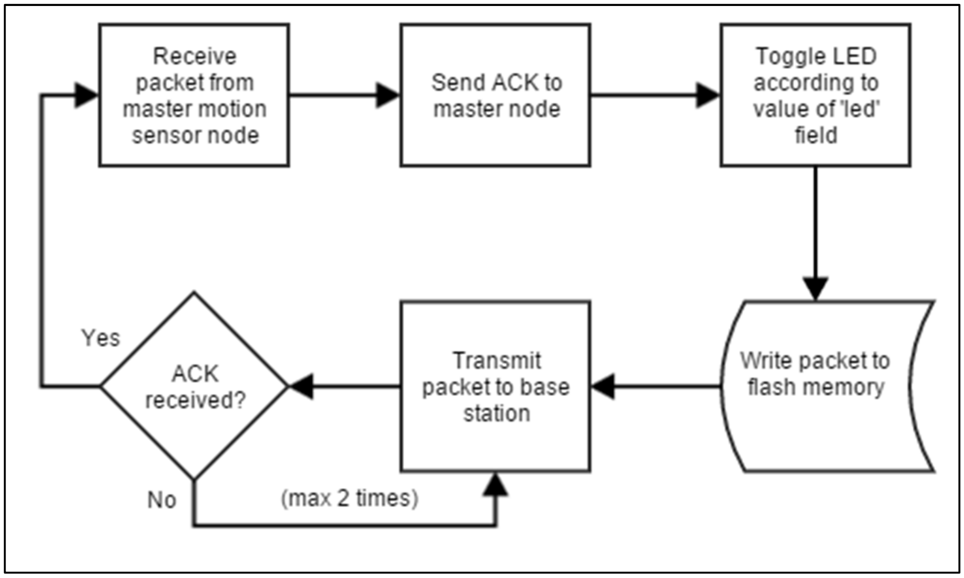

Figure 4-2-4-F1 – Flowchart of LED node

The nodes which are not connected to any sensors are only connected to the LEDs. Like the other nodes, the LED nodes also transmit their raw values to the base station by using the flash memory as buffer. The comparison of the light sensor value and threshold value is done by the master motion sensor node. Therefore, the master node can automatically determine whether the LEDs should be turned off, dimmed or turn on. The function of the LED node is only to toggle its LEDs based on the value received by the master node.

4.2.5 DAT-ACK Protocol

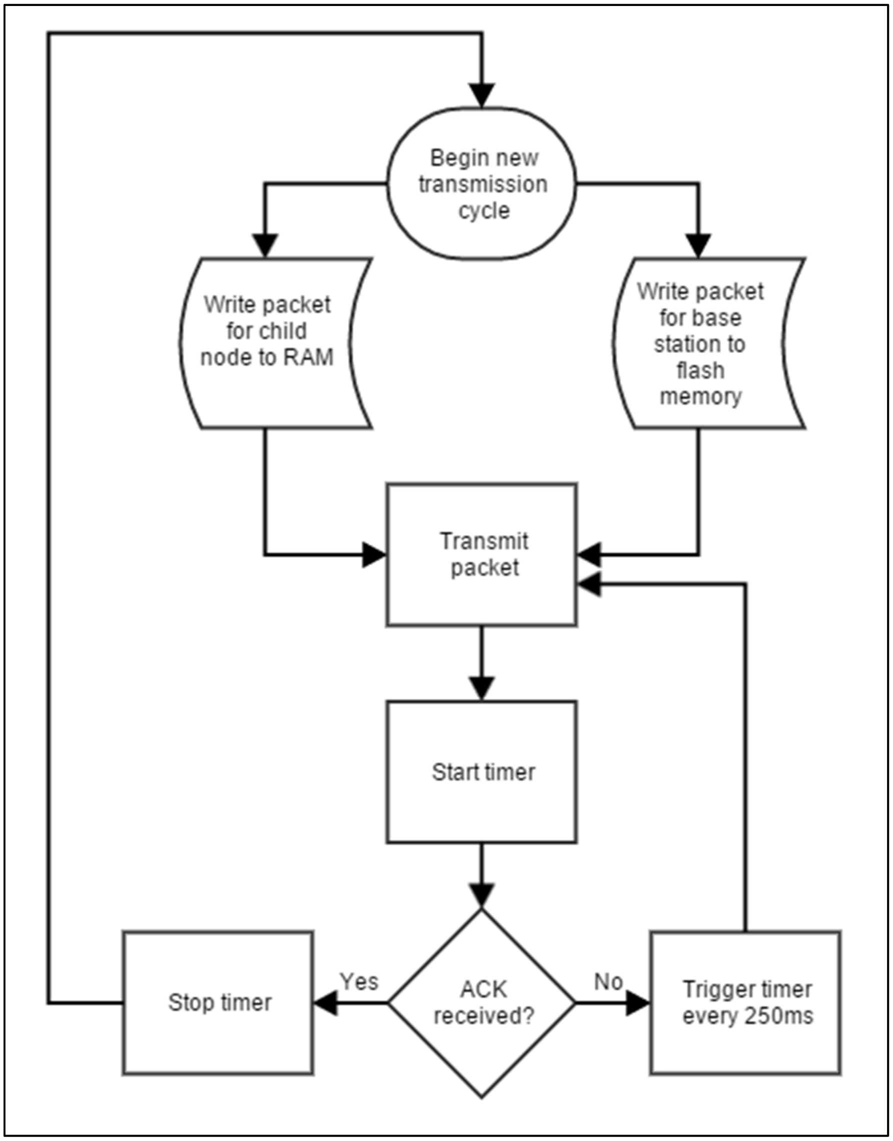

Figure 4-2-5-F1 – Flowchart of DAT-ACK protocol

The DAT-ACK protocol is used when a packet is transmitted to a base station or from a parent node to a child node. In the former, the packet to be sent is first stored in the flash memory, read from the flash memory and transmitted over the radio. A timer is triggered every 250ms to retransmit the packet if the ACK is not received from the base station. Retransmission occurs for a total of 2 times before the timer is stopped and the next transmission cycle begins. In the latter, the packets to be sent to the child nodes are stored in the flash memory. This is because these packets are more time-sensitive than the packets sent to the base station. For example, to toggle a LED child node would require a longer amount of time if the packet were to be written to the flash memory compared to the RAM. Therefore, the packet is stored in the RAM first, and then the timer runs every 250ms to check for ACK from the child node. If no ACK is received, the parent node retransmits for a total of 2 times before moving on to the next transmission cycle for the next packet in the buffer.

4.2.6 Routing Algorithm

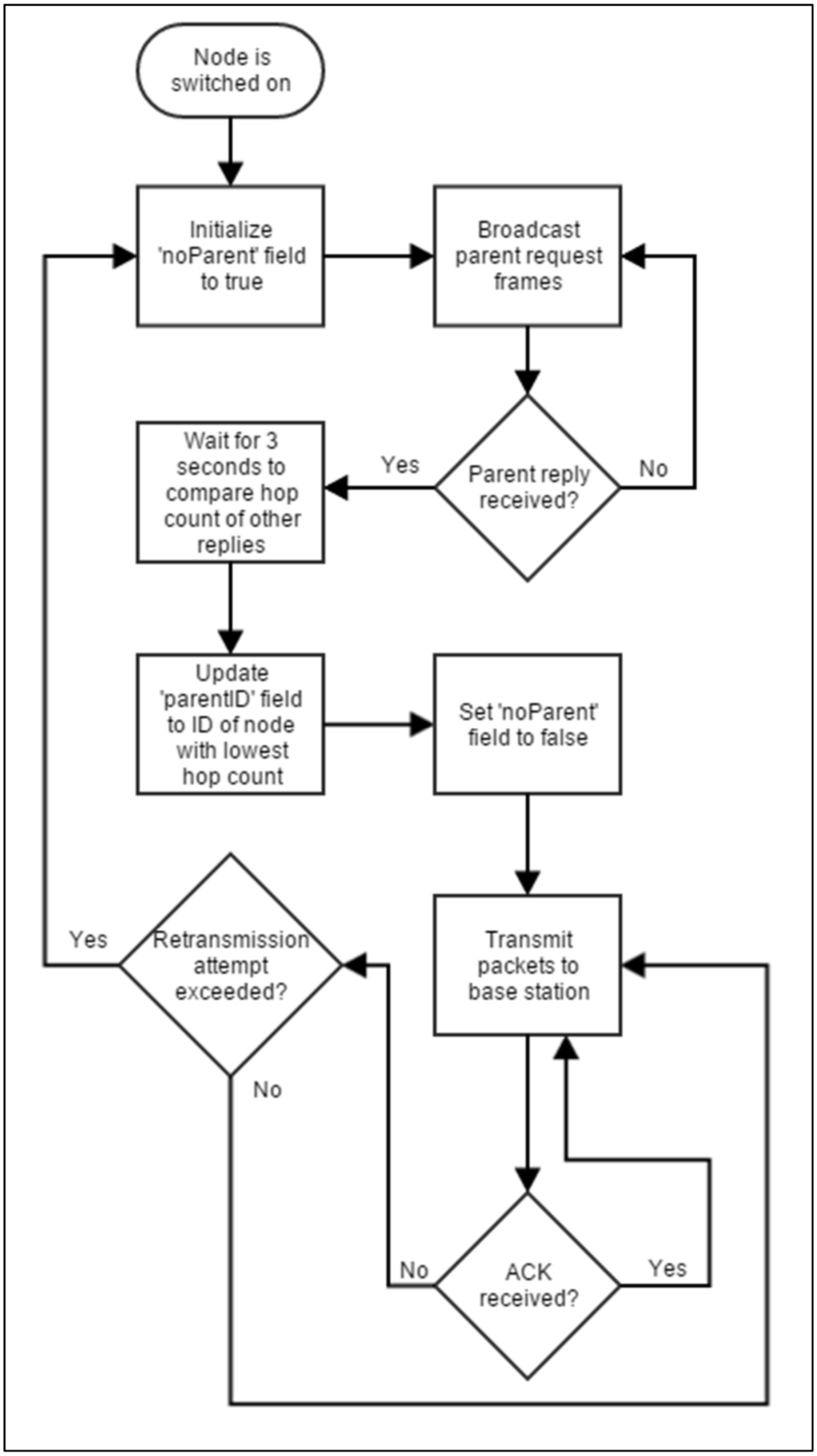

Figure 4-2-6-F1 – Flowchart of routing algorithm

Although the codes used for the 3 sets of nodes are slightly different, their behaviour in finding a parent node and acting as a parent node for child nodes is the same. During boot, the nodes attempt to find a parent first before transmitting any packets to the base station. A parent request broadcast is sent out every 1 second, and when the first reply to this response is received, the node will store the hop count of the node that replied and wait for 3 seconds to receive reply from other nodes. This is because the algorithm uses hop count as its sole metric, hence it waits to check if there are any nodes in the system with a lower hop count. After 3 seconds, the node with the lowest hop count is chosen as the parent. It is only after this step where the nodes begin their operation as light sensor nodes, motion sensor nodes or LED nodes. In the event that ACK is not received to the packets sent to the base station, this would indicate that the parent is down because the ACK was not forwarded back to the node. After retransmitting the last packet twice and not receiving any ACK, the node reverts back to the initial stage during boot and does not transmit any packets to the base station until a new parent is found. This demonstrates the self-healing characteristic of wireless sensor networks and their ability to form ad-hoc networks.

4.3 System Flow

The system flow of this project will be represented in a flow chart and emulates a situation where a vehicle passes by the road at night and the light sensor value is below the threshold value.

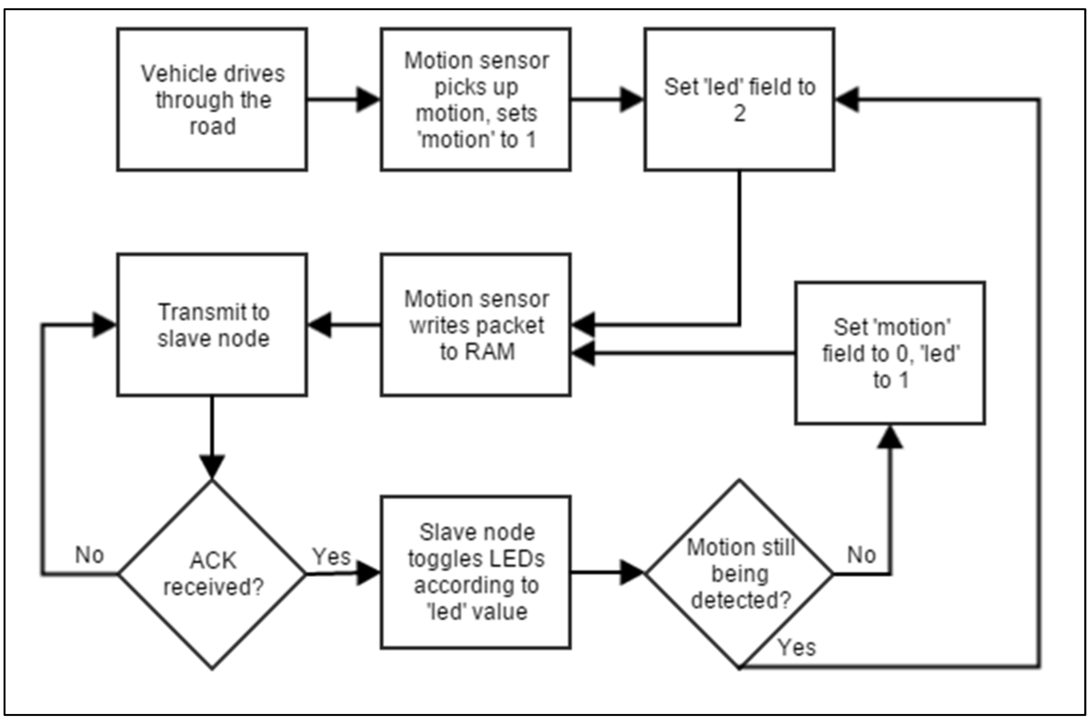

Figure 4-3-F1 – Flowchart of system flow

4.4 Database Design

The system will also store the output received from the nodes after it is displayed on the desktop computer. This is handled by the base station and the information stored could be accessed at a later time. The information can be used to fine tune the system as well.

Column Name	Data Type	Description
startTime	DATE	Timestamp of PC when packet was received
nodeid	INTEGER	ID of each node
pktid	INTEGER	Sequence number of each node
adc1_mv	FLOAT	Light sensor value
led	INTEGER	Status of LEDs, whether off, dimmed, or on represented by 0,1,2 respectively
motion	INTEGER	Set to 1 if motion is detected, otherwise 0
batt_val	FLOAT	Node battery voltage

Table 4-4-T1 – Database design of system

4.5 Algorithm Design

The network incorporates a routing algorithm to take into account situations whereby one of the nodes in the system would fail. As a result, all child nodes of the failed node are disconnected from the network and as such, the system's effectiveness is greatly hampered. To overcome this, the child nodes must be able to detect once their parent node is down and proceed to send out broadcast packets to request for a new parent. If a reply is received, the node waits for a predefined amount of time for replies from other nodes. This is so that the child node can connect to the newer parent with the lowest hop count. The routing protocol is implemented using a new packet structure for control packets, PacketCMsg. The fields in the control packet include:

Size (bytes)	Field Name	Description
2	Node ID	Uniquely identifies all nodes in the network
1	Hop Count	Stores the hop count, which is the number of nodes away from the Base Station. A node count of 1 indicates the node has the Base Station as its parent
1	Request	A broadcast request for a parent has a value of 1, whereas a reply to this request is 0.

Table 4-5-T1 – Packet structure and description of PacketCMsg

The pseudocodes below explain the operation when a new node is switched on.

```
Initialize noParent to true
Initialize wait to false
Intiialize lowest to 1
Initialize timer to 0
if noParent is equals to true
    while a broadcast reply is not received
        broadcast parent request frames every 1 second
    if a parent reply frame is received
        set wait to true
        if hopcount <= lowest
            set the value of lowest as hopcount of received frame
            set the value of parentid with the nodeid of received frame
        else
            set the value of lowest as hopcount of received frame
            set the value of parentid with the nodeid of received frame
        wait for 2 seconds by increasing value of timer every 1 second
    if timer == 2
        reset value of timer to 0
        set value of noParent and wait to false
else
    begin transmission cycle
    if a broadcast parent request frame is received
        transmit a parent reply frame
```

4.6 GUI Design

The GUI is a Java application which displays graphical components, raw packet values from nodes and inserts these values to the database. By reading the values in each received packet, the GUI is able to display the connectivity between the nodes by drawing lines. Although the positions of the nodes are fixed, their relationship however is dynamic and can change at any time. By using lines, each node's parent and child can be seen as well as their RSSI, which is represented by different colours. For example, if the RSSI value is below -80dBm, then a red line is used as the signal strength is considered to be weak. If the value is below -60dBm then the yellow colour is used. Values greater that -60dBm are considered to be strong and hence the green colour is used. The GUI employs the use of multiple threads for certain tasks as running it on a single thread would not make many things possible, for example updating the text area as soon as a packet as received. To overcome this, a separate thread runs this function when each packet is received. Doing this allows the main thread to update the graphical components at the same time.

4.7 Concluding Remark

The system proposed will consist of 6 modules, and together they form an automatic, smart lighting system without the need for user input. It will also ease the task of maintenance and quality assurance based on the output received from all the nodes. The system proposed will be attention-free, as it will not require the users to report any malfunction. The usage of LED lights which last much longer than traditional light bulbs lowers the maintenance cost as they will not be required to be replaced frequently. The nodes in the system are self-healing as they can reroute in the event that the connection to their parent is lost. A basic component of wireless sensor networks, ad-hoc functionality increases the resiliency and reliability of these nodes in a real-world implementation. It also increases the effective range or size of a system as the higher the number of nodes, the greater the distance that can be covered.

Chapter 5 – System Implementation

5.1 Hardware Setup

The hardware implementation for the MICAz node is different for the motion sensor node and LED node. They both use the 51-pin breakout board to connect to their respective components but their wiring to the pins are different. Besides that, the motion sensor has to be soldered to a power source before being soldered to the breakout board. The figure below shows the hardware design for the MICAz nodes used in this project. Besides that, a diagram of the 51 supported pins is also shown.

5.1.1 Motion sensor breakout board pin wiring

Pin	Function
PW4	The pin which reads the output of the motion sensor. A value of 1 indicates motion.
GND	Connected to the same ground as the motion sensor and battery holder

Table 5-1-1-T1 – Wiring of motion sensor breakout board

5.1.2 Motion sensor wiring

Pin	Function
VCC	Connected to the positive terminal of the battery holder
Negative	Connected to the GND pin on the breakout board, which also connects to the negative terminal of the battery holder.
OUT	Connected to the PW4 pin on the breakout board. Returns a high voltage of 3.3V (1) when motion is detected and low voltage of 0V (0) when there is no motion.

Table 5-1-2-T1 – Wiring of motion sensor

5.1.3 LED node breakout board pin wiring

Pin	Function
PW0	First LED of segment 1
PW1	Second LED of segment 1
PW2	Third LED of segment 1

INT1	First LED of segment 2
INT2	Second LED of segment 2
INT3	Third LED of segment 2
GND	Connected to the same ground as all the other LEDs

Table 5-1-3-T1 – Wiring of LED node breakout board

5.2 Software Setup

5.2.1 Installation of XubunTOS on Oracle VM VirtualBox

The process of configuring the nodes and receiving packets from the base station requires a Linux environment, namely XubunTOS. However, rather than installing this OS directly into a machine, it can be installed in VirtualBox to work in a Windows environment.

1) Download and install VirtualBox from https://www.virtualbox.org/wiki/Downloads

2) Obtain a Virtual Machine Image Disk (.vmdk) of the XubunTOS from the UTAR FICT laboratories or from the following link http://tinyos.stanford.edu/tinyos-wiki/index.php/Installing_XubunTOS_(with_TinyOS_from_tp-freeforall/prod_repository)_in_VirtualBox and follow the instructions stated in the webpage. The remaining 3 steps may be skipped if XubunTOS is downloaded from the above link.

3) In VirtualBox, click on 'New' and set the values of the Name, Type and Version to 'Xubuntos', 'Linux' and 'Ubuntu (32 bit)' respectively. All other settings can be left as default. Continue clicking 'Next'.

4) When prompted for hard drive settings, choose 'Use an existing virtual hard drive file' and browse and select the .vmdk file from step 2.

5) Click on 'Create'. The VM can now be started by clicking on 'Start'.

The default username and password for the XubunTOS is 'xubuntos2' and 'tinyos' respectively.

5.2.2 Updating Java

1) Download the latest Oracle Java binaries, jdk-8u73-linux-i586.tar.gz and jre-8u73-linux-i586.tar.gz from http://www.oracle.com/technetwork/java/javase/downloads/index.html

2) Using a terminal, navigate to the folder where the files are downloaded. Copy the files to the Java path using the commands below:

```
sudo cp -r jdk-8u73-linux-i586.tar.gz /usr/lib/jvm
sudo cp -r jre-8u73-linux-i586.tar.gz /usr/lib/jvm
cd /usr/lib/jvm
```

3) Extract the files from the archive using the following commands:

```
sudo tar xvzf jdk-8u73-linux-i586.tar.gz
sudo tar xvzf jre-8u73-linux-i586.tar.gz
```

4) Modify the Linux system path file to update the Java system variables. To do this, run the command ‘sudo gedit /etc/profile’ and append the following lines:

```
JAVA_HOME=/usr/lib/jvm/jdk1.8.0_73
PATH=$PATH:$HOME/bin:$JAVA_HOME/bin
JRE_HOME=/usr/lib/jvm/jre1.8.0_73
PATH=$PATH:$HOME/bin:$JRE_HOME/bin
export JAVA_HOME
export JRE_HOME
export PATH
```

Save the file and exit.

5) Next, the system has to be informed of the new location for the JRE and JDK folders, and the new default Java version has to be updated. To do this, run the following commands:

```
sudo update-alternatives --install "/usr/bin/java" "java" "/usr/lib/jvm/jre1.8.0_73/bin/java" 1
sudo update-alternatives --install "/usr/bin/javac" "javac" "/usr/lib/jvm/jdk1.8.0_73/bin/javac" 1
sudo update-alternatives --install "/usr/bin/javaws" "javaws" "/usr/lib/jvm/jre1.8.0_73/bin/javaws" 1
sudo update-alternatives --set java /usr/lib/jvm/jre1.8.0_73/bin/java
sudo update-alternatives --set javac /usr/lib/jvm/jdk1.8.0_73/bin/javac
sudo update-alternatives --set javaws /usr/lib/jvm/jre1.8.0_73/bin/javaws
```

6) Lastly, reload /etc/profile and verify the Java version by executing the following commands:

```
. /etc/profile
java -version
```

The message should display that the installed Java version is 1.8.0_73

5.2.3 Preparing source codes for node configuration

In order to successfully configure any node in this project, 5 files are required. They are: FileM.nc, FileC.nc, PacketMsg.h, mig.txt, Makefile and volumes-at45db.xml. The importance of these files can be summarized into the table below:

File	Description
FileC.nc	The configuration file which includes the components that are to be used by the node. These include timers, LEDs, flash memory operations and so on.
FileM.nc	The actual source code of the node, contains implementation of the components. Codes which specify what to do during boot, when a packet is received, etc. is defined here.
PacketMsg.h	Stores the structure of the different packets used by the nodes. Includes the size and name of each field for each packet type. A key component which bridges the Java implementation with the node functionality.
mig.txt	Contains the command to generate Java class files for each packet structure. This will allow the GUI to read and generate the different types of packets. The content of this file should be renamed depending on which packet type the class file is to be generated for. This file can be executed using the command './mig.txt'
Makefile	Contains definition of the AM group and channel, which should be the same for all nodes in the system. Otherwise, packets transmitted will not be received by other nodes.
volumes-at45db.xml	Defines the name of the volume used in the flash memory as well as its size. The size declared must be a numerical value of bytes.

Table 5-2-3-T1 – Description of project files

5.2.4 Files used by GUI

The java_app folder contains the source code, dependencies and resources of the GUI. The .jar files are used by the compiler to access the definitions and usage of the libraries of TinyOS and JDBC. Meanwhile, the resources folder stores the images used in the GUI such as the battery icon, the picture of the road, and so on. In case any of the .java files are edited, they need to be recompiled for the changes to take effect. To do this, simply run the 'javac *.java' before running the program using the command 'java PingPong'.

5.2.5 Configuring and Installing WampServer

1) Download and install WampServer from http://www.wampserver.com/en/

2) Once the installation is complete, select 'Run WampServer' from the start menu.

3) In a web browser, navigate to the loopback address, 127.0.0.1 and click on phpmyadmin.

4) Proceed to the 'mysql' database which was created during the installation.

5) Create a table called 'packets' according to the database design in section 4.4.

5.3 Setting and Configuration

The CD contains 5 folders, namely BaseStation, MotionSensor, LightSensor, LEDNode and java_app. These folders contain the source code required to program different components of the system as well as a GUI to display and control the nodes in the system. To set up the demonstration, 14 nodes will have to be programmed, based on the instructions below:

1) Connect the MIB520 programming board to the PC via USB and attach a node to it. Ensure that the programming board is attached to the VM by selecting it under USB devices from VirtualBox.

2) Using a terminal, navigate to the BaseStation folder and execute the command ‘make micaz install,1 mib510,/dev/ttyUSB0’. The value ‘1’ represents the node ID that will be assigned to the node. **Ensure that no other node in the network is programmed with the ID ‘1’.**

3) Navigate to the LightSensor folder and execute the command ‘make micaz install,99 mib510,/dev/ttyUSB0’. Attach the light sensor board to the node.

4) Navigate to the MotionSensor folder and execute the command ‘make micaz install,2 mib510,/dev/ttyUSB0’. Modify the value of led_node in FileM.nc based on the system architecture in section 4.1. Repeat this for 5 more nodes with the IDs 4,6,8,10 and 12.

5) Navigate to the LEDNode folder and execute the command ‘make micaz install,3 mib510,/dev/ttyUSB0’. Repeat this for 5 more nodes with the IDs 5,7,9,11 and 13.

6) Reattach the base station node to the programming board. Navigate to the java_app folder and run the following commands:

```
export MOTECOM=serial@/dev/ttyUSB1:57600
javac *.java
java PingPong
```

7) Turn on nodes 2, 3, 12 and 13 sequentially (refer to section 5.4 for explanation). Then proceed to turn on all other nodes.

5.4 System Operation

Once the configuration steps from section 5.3 have been executed, the GUI is presented on the screen. As long as the GUI is displayed, the base station continuously transmits beacon replies so that surrounding nodes are able to pair up with the base station as their parent node. Based on the system architecture, nodes with ID 2, 3, 12 and 13 will have their parent as the base station as they are the nearest nodes. As the base station receives packets, the text area on the left displays the raw values of the packet. The colour and position of the lines are also updated for each node when a new packet is received, along with the battery level indicator icon. In order to demonstrate the routing algorithm, node 2 will be switched off. As a result, all the child nodes do not have a parent to allow communication to the base station. To overcome this, the node enters a state of 'no parent' and withholds from transmitting raw values to the base station. Instead, it continuously transmits and listens to beacon replies from potential parent nodes. Once the nodes find a parent with the lowest hop count, they proceed to transmit their raw values to the base station.

Another method of demonstrating the functionalities of the system is to toggle one of the LEDs manually. To do this, simply right click on any LED node and choose between turning it off, dim, or turning it on. If this action is successful, the button of the node will change to represent the current state of the node.

5.5 Concluding Remark

To summarize, this section explained the steps required to create the system, from both a hardware and software perspective. Given that all the required equipment are available, one simply has to program the nodes according to the system architecture to observe the action of a motion sensor node controlling the behaviour of LED nodes and the self-healing feature of a wireless sensor network whereby child nodes are able to adapt to losing a parent node. The steps listed to update the Java on XubunTOS is pivotal because if the default installed Java version is used, then some elements on the GUI may not display correctly.

Chapter 6 – System Evaluation and Discussion

6.1 System Testing and Performance Metrics

A form of testing should be carried out to determine the effectiveness of the DAT-ACK protocol. To achieve this, 50 packets will be sent from a node to the base station. By observing the packet ID on the console, any missing packets as well as duplicate packets can be identified. This testing will be carried out twice, first with only 2 nodes and secondly with 3 nodes, with the extra node acting as an intermediary between the 2 nodes. The results are repeated 10 times to obtain an average.

In order to toggle a LED node manually from the GUI, the packet would have to be transmitted from the base station to the target node. Depending on the number of hops away the target node is, this may or may not be successful and hence a testing is to be carried out to obtain the success rate. This testing will involve toggling a LED node for a total of 10 times. The packet from the base station will only be transmitted once so if it is lost or discarded on the way, it will not be retransmitted. By calculating the average of the success rate of each LED node, a percentage is obtained and hence the overall reliability can be obtained.

To demonstrate the routing algorithm of this project, one of the nodes directly connected to the base station as a parent, node 13, will be switched off. This will force all its child nodes to reroute. A screenshot of the system before and after this node is switched off is shown to better understand the behaviour of these nodes on how they reroute. If all the child nodes are able to reroute and continue transmission normally, this would prove the effectiveness of the routing algorithm.

To determine how the system handles additional nodes, a node with ID 67 is added to the system to check if it can find a parent and then proceed to transmit and receive packets from the base station. These nodes are not represented in the system architecture. Therefore, when a packet is received from these nodes, the GUI first shows a dialog box to ask the administrator if the node should be allowed to be a part of the system. If the administrator rejects, no ACK packets are sent to this node and all subsequent packets are ignored. If the administrator approves, however, the node is a part of the system but does not appear in the GUI.

6.2 Testing Setup and Result

6.2.1 DAT-ACK Protocol Packet Reliability

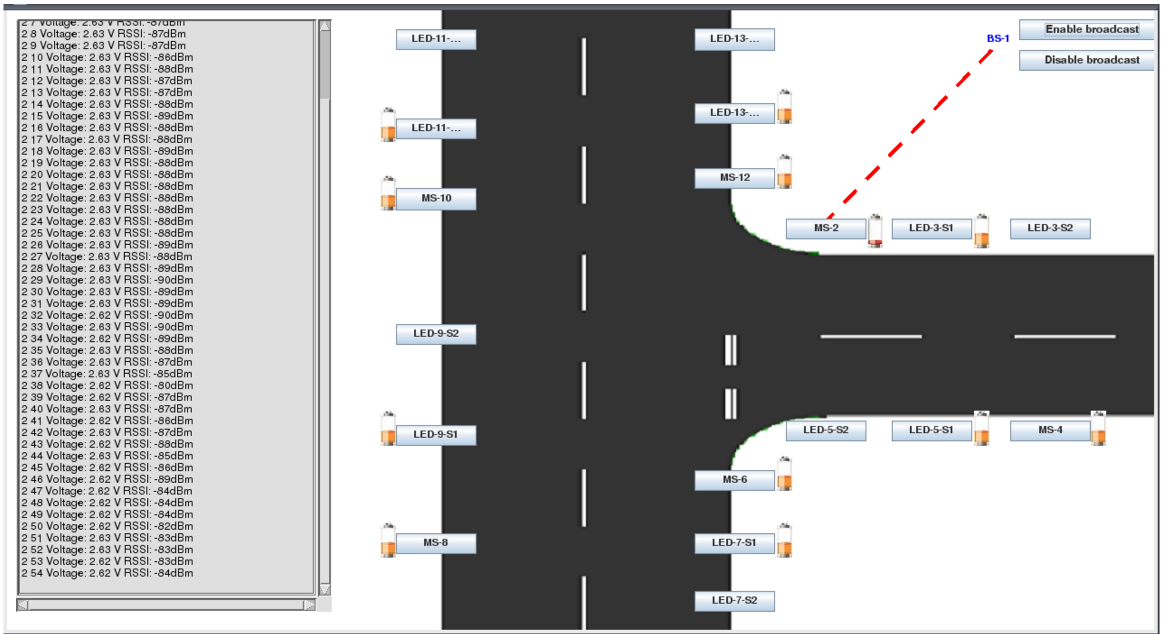

Figure 6-2-1-F1 DAT-ACK protocol testing with 2 nodes

The figure above shows node 2 transmitting packets to the base station. The raw packet values can be seen in the text area on the left of the GUI. The results for all 10 attempts are shown below.

Attempt	Number of packets sent	Number of packets lost	Number of duplicate packets
1	50	2	1
2	50	1	0
3	50	0	0
4	50	1	1
5	50	0	0
6	50	0	1
7	50	2	3
8	50	1	1
9	50	2	1
10	50	2	2
Average	50	1.1	1.0

Table 6-2-1-T1 – Results of DAT-ACK protocol testing with 2 nodes

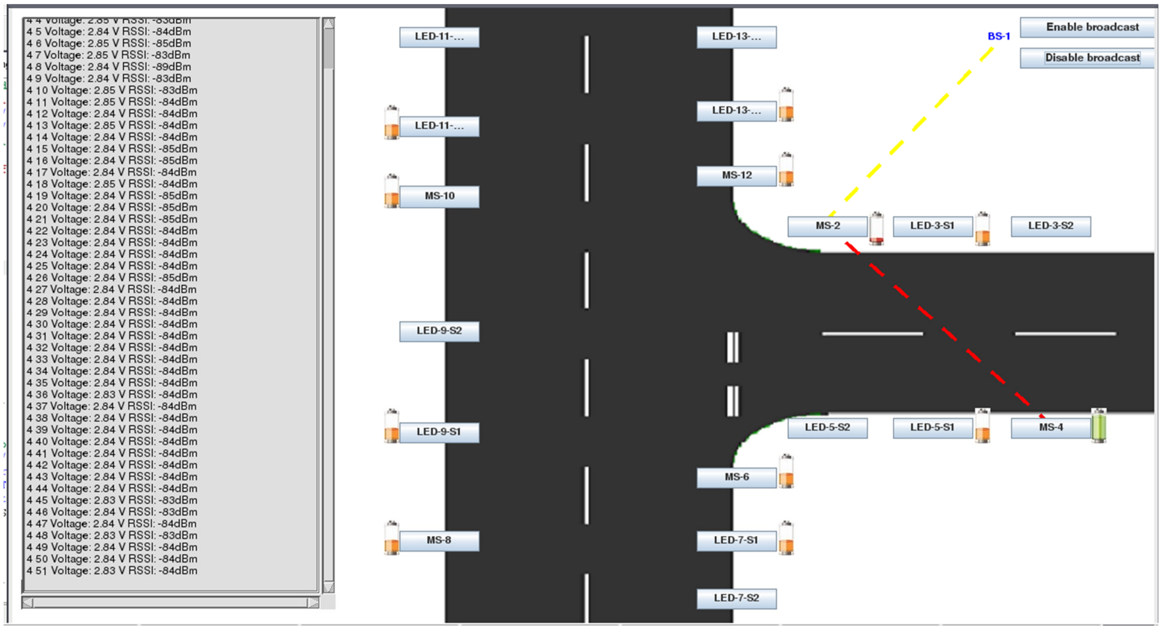

Figure 6-2-1-F2 – DAT-ACK protocol testing using 3 nodes

For the second attempt, the target node, node 4 transmits 50 packets to the base station by hopping through node 2. The output on the left omits the packets transmitted by node 2 for the purpose of this test.

Attempt	Number of packets sent	Number of packets lost	Number of duplicate packets
1	50	0	1
2	50	0	0
3	50	3	5
4	50	1	3
5	50	0	2
6	50	1	3
7	50	2	1
8	50	3	2
9	50	3	3
10	50	1	1
Average	50	1.4	2.2

Table 6-2-1-T2 – Results of DAT-ACK protocol testing with 3 nodes

The presence of duplicate packets would indicate that the node had to retransmit because an acknowledgement was not received within 250ms. A possible explanation for this is that the ACK packet was lost during transit to the target node, or that the radio was busy and hence could not transmit the packet. Based on the 2 tests, the average number of packets lost or duplicated is between 1 to 2.2. Both results prove that the DAT-ACK protocol is reliable. The highest number of packets lost was 3 and this could have been influenced by a number of factors, such as interference from other signals such as Wi-Fi signals or environmental issues such as moving objects in the testing area.

6.2.2 Base Station to LED Node Success Rate

To simulate the actual environment where the system will be implemented, the target LED node is 1 hop away from the base station. This is to test if the intermediary node is efficient in forwarding packets to other nodes in the system while at the same time transmitting its own raw values to the base station. To test this, node 11 will be toggled manually and if the background colour of the button changes according to the toggled value, it is considered a success. Note that the background colour does not change according to which option is clicked on while toggling the node but rather the value received in the packet from the target node.

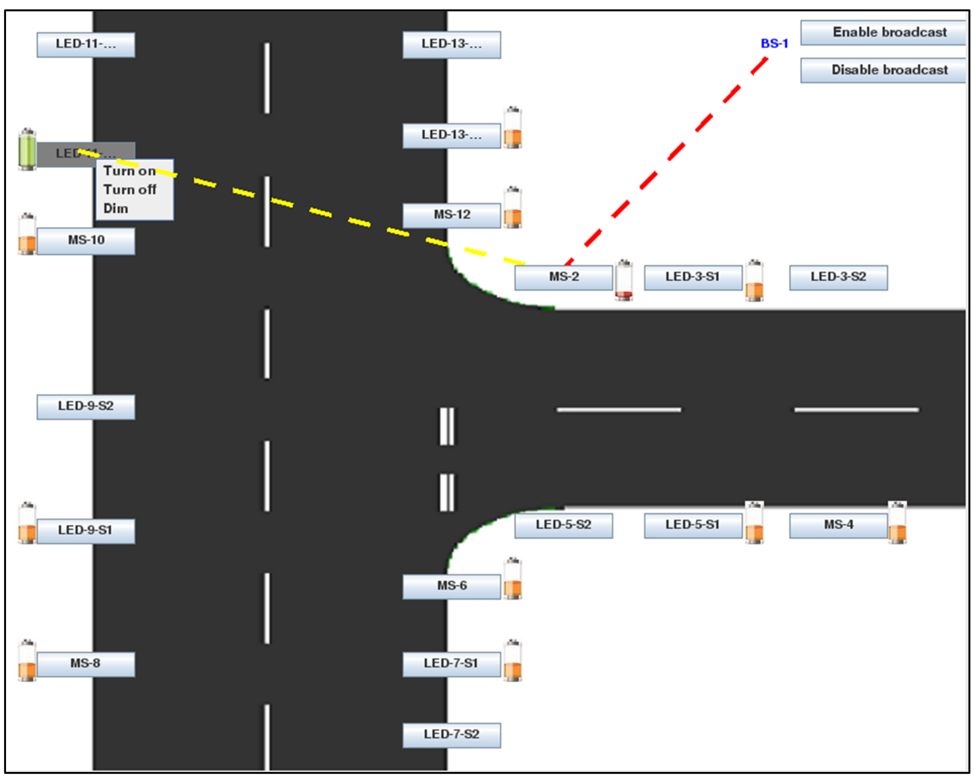

Figure 6-2-2-F1 – Screenshot of system before toggling LED node

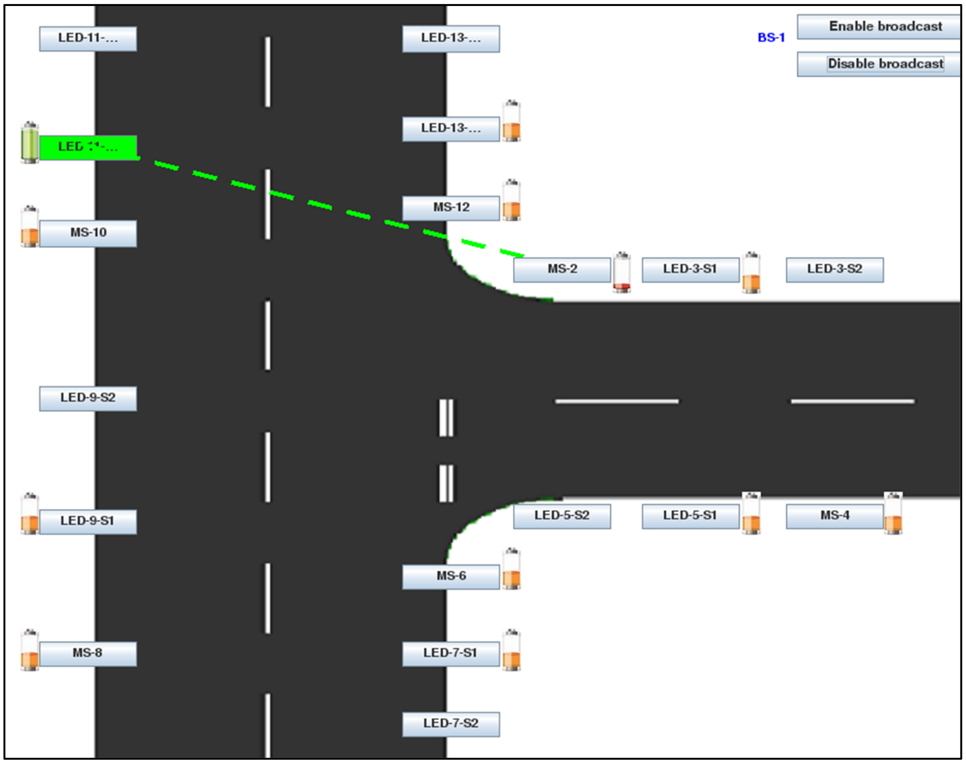

Figure 6-2-2-F2 – Screenshot of system after toggling LED node

6.2.3 Node Failure Testing

To test the reaction of nodes once their parent is down, node 11, which is directly connected to the base station, has 2 child nodes relying on it for connection to the base station. If node 11 were to be switched off, then its child nodes would stop receiving ACK packets from the base station and subsequently would have to search for a new parent. This testing aims to analyse this behaviour and determine the new node which is selected as the parent.

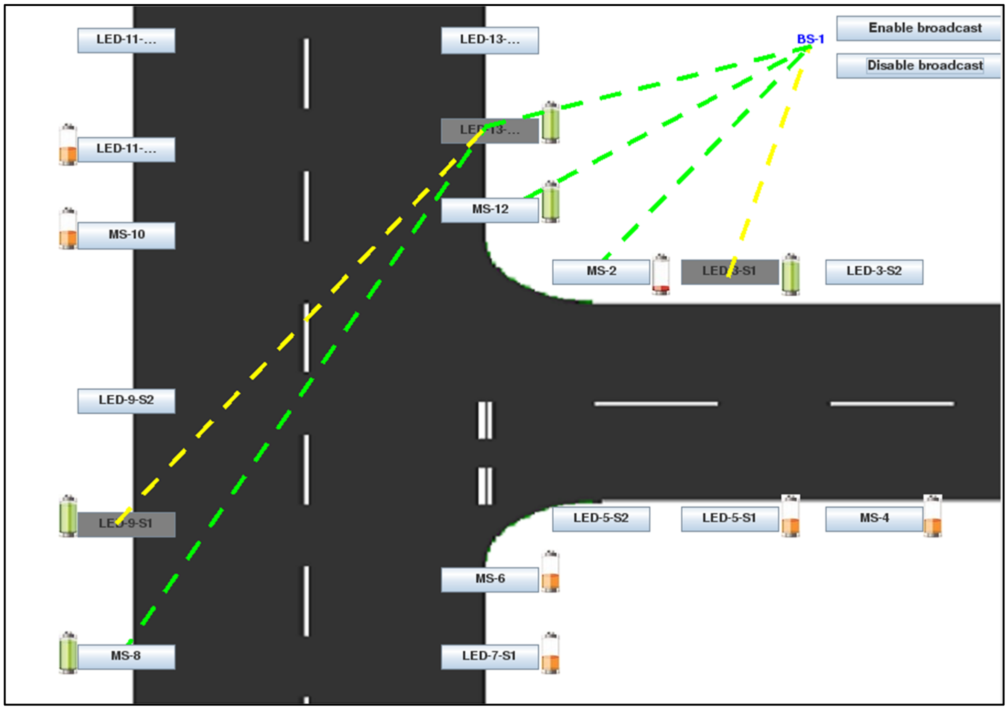

Figure 6-2-3-F1 – Screenshot of system before node 11 is turned off

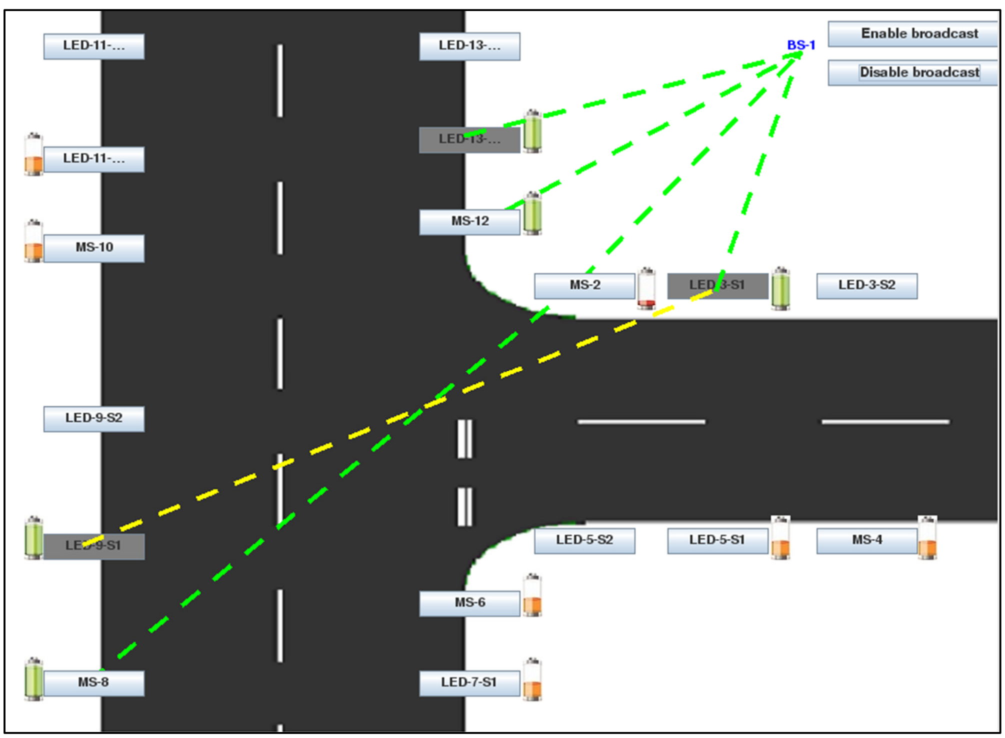

Figure 6-2-3-F2 – Screenshot of system after node 11 is turned off

It is observable that once a node fails, the child nodes automatically reroute to other nodes which have a parent. This shows that the algorithm is written to not only find its own parent but also to find a new one in case the current one is disabled.

6.2.4 Scalability Testing

Figure 6-2-4-F1 – Screenshot of alert box requiring user confirmation to add node 67

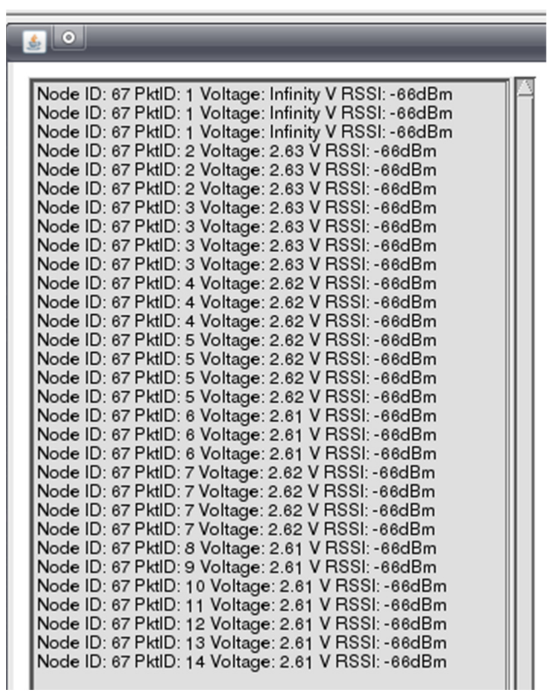

Figure 6-2-4-F2 – Raw packet values of node 67

If the administrator decides to not allow the node to join the system, then no ACK packet is sent to the node and any new packets received are ignored by the base station.

The blocked nodes' ID are stored in an array and hence the only way to allow a rejected node to enter the system is to restart the GUI. Note that once a node is accepted, its output is displayed in the text area. However, the GUI does not support representing the new node in the form of a button like the other nodes.

6.3 Project Challenges

The completion of this system would not have been possible if the many challenges encountered along the way were not solved. These challenges include:

- **Stamping RSSI at receiver node:** The base station configuration file was modified to stamp the RSSI onto the packet for its own child nodes. This was accomplished by checking the value of the parentid field of the packets. For other nodes, the RSSI was to be stamped into the packet first before being forwarded to the base station.
- **Writing packets to flash memory:** The flash memory is mainly used to store data variables such as integer arrays or other variables. However, the storage of packets required several tweaks to the code for it to work. The packets were to be written to the flash buffer first, and then read from the buffer before transmission.
- **Consolidating the use of flash memory and RAM as buffers:** Before packets are transmitted, with the exception of packets used for routing, they are stored in a buffer to allow for retransmission. Packets between nodes are time-sensitive hence they are stored in the RAM as it is faster to write to compared to flash memory. The combined approach of 2 buffers in different locations increased the complexity of the programming as there were 4 buffer pointers in total to manage.
- **Writing a routing algorithm:** The implementation of the routing algorithm required the use of a separate packet header format. Not only were the nodes supposed to manually find a parent, they would also have to deal with losing a parent node. The algorithm compares the hop count of different nodes before choosing the ideal parent node. This allowed the system to be resilient.
- **Designing a GUI from scratch:** The GUI for this system was designed from the bottom up, first by drawing buttons to represent nodes along the road as illustrated in the system architecture. The GUI was also responsible to reflect the signal strength between nodes as well as their battery levels. The parent and

child nodes could also be easily determined as the GUI draws line to and from parent nodes. All this, including the use of a text area, required multiple threads as a single thread alone could not display all these elements in a timely manner.

6.4 Objectives Evaluation

The first objective of this project was to employ the use of sensors to control the lampposts. This was successfully achieved as the nodes were programmed to rely on the light sensor and motion sensor values to control the LEDs, as shown in section 4.2.

The second objective was to provide a GUI that would ease the management and administration of the system. This has been accomplished as the GUI allows a full view of the system as well as the status of each individual nodes. The text area allows the administrator to view raw packet values as they are received by the base station, whereas the GUI is able to display the RSSI between nodes as well as the presence of motion for each motion sensor node. This allows the administrator to visualize what is happening on the road without physically being there. The administrative tasks are also enhanced by allowing LEDs to be toggled manually and controlling the entry of new nodes into the system.

The final objective of the system was to demonstrate a solution for an energy-efficient street lighting system which is reliable. Based on the results of the tests carried out in section 6.2, it can be said that the system has achieved a high degree of reliability and is ready to be implemented to control actual lampposts.

6.5 Concluding Remark

The tests carried out and performance metrics obtained prove the efficiency and reliability of the system. The challenges faced were key aspects which improved and enhanced the final system. It can be said that all objectives were achieved by ensuring the hardware and software setup satisfied the requirements of the project and that the testing setup and results further proved the viability of this project, should it be implemented in a real-world situation.

Chapter 7 – Conclusion and Recommendation

7.1 Conclusion

Through the development of this project, it is hoped that the prospects of Smart Street Lighting systems are brought to light. They are capable of significantly reducing costs in terms of energy consumption. By implementing this system, the reliance on sodium vapour lamps will slowly diminish and replaced by cheap, long-lasting LED lamps. It will be a move that will spread to other countries and eventually, it is hoped that the concept of using sensors to control street lamps will be an adopted practice.

This project also successfully converted the project objectives into deliverables such as a GUI, a routing algorithm, a DAT-ACK protocol and a system which forms an ad-hoc network which uses sensors as a means to toggle street lights. Through this project, the idea of reporting a faulty street light or even governments spending large amounts of money on the maintenance and cost of sodium-vapour lampposts may potentially be a thing of the past.

7.2 Recommendation

There are several aspects of the system which can be improved upon. Firstly, the routing algorithm does not implement any sort of load balancing. It is possible for a node to have more than 10 child nodes while other surrounding nodes do not have any child nodes. To overcome this, the algorithm should use the number of current child nodes as the second metric in selecting a parent node.

Besides that, the system could also leverage on the use of location sensors to further enhance the efficiency of the system. Location sensors allow for more complex deployment scenarios compared to the T-junction used in this project. The use of location sensors may also add more functionalities to the system, such as automatically displaying new nodes in the GUI.

Since that the system will only be active during the night, there is no need for the nodes to transmit packets to the base station or check the sensor values during the day. Rather, the nodes should be programmed to be active during a specified time period so that the communications and sensor values checking will only occur when the system is active. This can be implemented by the use of timers and the sleep/wake functionality of the radio.

Furthermore, the use of a single base station results in the possibility of a single point of failure. A failed base station node will result in the entire system failing as all nodes will lose their parent node. Although the motion sensor nodes will continue to pick up motion and toggle their respective LED nodes, the manageability and administrative control of the system would be greatly hampered. Rather, there should be a backup base station or 2 base stations in the system to provide for redundancy.

References

Bluetooth.com, (2015). *Basics | Bluetooth Technology Website*. [online] Available at: http://www.bluetooth.com/Pages/Basics.aspx [Accessed 11 Jul. 2015].

Datasheet of Mica-Z Wireless Measurement System. (2005). 1st ed. [ebook] San Jose: Crossbow Technologies Inc. Available at: http://www.cmt-gmbh.de/MICAz.pdf [Accessed 8 Aug. 2015].

Digi.com, (2015). *ZigBee® Wireless Standard - Technology - Digi International*. [online] Available at: http://www.digi.com/technology/rf-articles/wireless-zigbee [Accessed 11 Jul. 2015].

Engineersgarage.com, (2012). *WiMax vs WiFi - Difference between WiFi and WiMax Technologies*. [online] Available at: http://www.engineersgarage.com/contribution/wimax-vs-wifi [Accessed 11 Jul. 2015].

ISTQB Exam Certification, (2015). *Prototyping Model*. [image] Available at: http://istqbexamcertification.com/wp-content/uploads/2012/01/Prototype-model.jpg [Accessed 11 Jul. 2015].

Istqbexamcertification.com, (2015). *What is V-model- advantages, disadvantages and when to use it?*. [online] Available at: http://istqbexamcertification.com/what-is-v-model-advantages-disadvantages-and-when-to-use-it/ [Accessed 11 Jul. 2015].

Istqbexamcertification.com, (2015). *What is Waterfall model- advantages, disadvantages and when to use it?*. [online] Available at: http://istqbexamcertification.com/what-is-waterfall-model-advantages-disadvantages-and-when-to-use-it/ [Accessed 11 Jul. 2015].

Lavric, A., Popa, V. and Sfichi, S. (2014). Street lighting control system based on large-scale WSN: A step towards a smart city. *2014 International Conference and Exposition on Electrical and Power Engineering (EPE)*.

Lavric, A., Popa, V., Finis, I. and Simion, D. (2012). The design of a street lighting monitoring and control system. *2012 International Conference and Exposition on Electrical and Power Engineering*. [online] Available at: http://red.pe.org.pl/articles/2012/11a/70.pdf [Accessed 3 Apr. 2015].

Leccese, F., Cagnetti, M. and Trinca, D. (2014). A Smart City Application: A Fully Controlled Street Lighting Isle Based on Raspberry-Pi Card, a ZigBee Sensor Network and WiMAX. *Sensors*, 14(12), pp.24408-24424.

Maxwell, W., Cacan, M. and Haile, C. (2013). *Pulse Width Modulation*.

MPR-MIB Users Manual. (2007). 1st ed. [ebook] San Jose: Crossbow Technology, pp.5, 19, 44. Available at: http://staff.utar.edu.my/gohhg/research/files2/xbow_MPR-MIB_Series_Users_Manual.pdf [Accessed 5 Apr. 2015].

MTS/MDA Sensor Board Users Manual. (2007). 1st ed. [ebook] San Jose: Crossbow Technology, Inc, p.4. Available at: http://staff.utar.edu.my/gohhg/research/files2/xbow_MTS-MDA_Series_Users_Manual.pdf [Accessed 5 Apr. 2015].

Müllner, R. and Riener, A. (2011). An energy efficient pedestrian aware Smart Street Lighting system. *Int J of Pervasive Comp & Comm*, 7(2), pp.147-161.

Rouse, M. (2015). *What is Prototyping Model? - Definition from WhatIs.com*. [online] SearchCIO. Available at: http://searchcio.techtarget.com/definition/Prototyping-Model [Accessed 11 Aug. 2015].

SpeedGuide, (2015). *What is the actual real-life speed of wireless networks ?*. [online] Available at: http://www.speedguide.net/faq/what-is-the-actual-real-life-speed-of-wireless-374 [Accessed 11 Jul. 2015].

Telecomabc.com, (2005). *Zigbee - Telecom ABC*. [online] Available at: http://www.telecomabc.com/z/zigbee.html [Accessed 11 Jul. 2015].

Tradingeconomics.com, (2015). *Population covered by mobile cellular network (%) in Malaysia*. [online] Available at: http://www.tradingeconomics.com/malaysia/population-covered-by-mobile-cellular-network-percent-wb-data.html [Accessed 5 Apr. 2015].

Tradingeconomics.com, (2015). *Population covered by mobile cellular network (%) in Germany*. [online] Available at: http://www.tradingeconomics.com/germany/population-covered-by-mobile-cellular-network-percent-wb-data.html [Accessed 5 Apr. 2015].

tutorialspoint, (2015). *Agile Model*. [image] Available at: http://www.tutorialspoint.com/sdlc/images/sdlc_agile_model.jpg [Accessed 11 Jul. 2015].

tutorialspoint, (2015). *V-Model*. [image] Available at: http://www.tutorialspoint.com/sdlc/images/sdlc_v_model.jpg [Accessed 11 Jul. 2015].

tutorialspoint, (2015). *Waterfall Model*. [image] Available at: http://www.tutorialspoint.com/sdlc/images/sdlc_waterfall_model.jpg [Accessed 11 Jul. 2015].

tutorialspoint.com, (2015). *SDLC - Agile Model*. [online] Available at: http://www.tutorialspoint.com/sdlc/sdlc_agile_model.htm [Accessed 11 Jul. 2015].

tutorialspoint.com, (2015). *SDLC - V-Model*. [online] Available at: http://www.tutorialspoint.com/sdlc/sdlc_v_model.htm [Accessed 11 Jul. 2015].

tutorialspoint.com, (2015). *SDLC - Waterfall Model*. [online] Available at: http://www.tutorialspoint.com/sdlc/sdlc_waterfall_model.htm [Accessed 11 Jul. 2015].

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 2
Student Name & ID: Ajay Kumar Doshi – 1102867	
Supervisor: Dr. Goh Hock Guan	
Project Title: Energy Efficient Smart Lampposts for Smart Cities	

1. WORK DONE - Discussed design and architecture of system - Obtained breakout boards and motion sensor
2. WORK TO BE DONE - Redefine project scope and project objectives - Purchase additional battery holders and motion sensors
3. PROBLEMS ENCOUNTERED - None
4. SELF EVALUATION OF THE PROGRESS - Reevaluated project scope and objectives

Supervisor's signature

Student's Signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 4
Student Name & ID: Ajay Kumar Doshi – 1102867	
Supervisor: Dr. Goh Hock Guan	
Project Title: Energy Efficient Smart Lampposts for Smart Cities	

1. WORK DONE - Identified components of GUI - Discussed and determined system architecture in detail
2. WORK TO BE DONE - Begin development of GUI - Test the functionality of LED node controlling 2 different arrays of LEDs
3. PROBLEMS ENCOUNTERED - Cannot write packets to flash memory - RSSI values were inaccurate
4. SELF EVALUATION OF THE PROGRESS - Report-writing skills improving, feel more capable in explaining any aspect of the system more thoroughly

Supervisor's signature

Student's Signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 6
Student Name & ID: Ajay Kumar Doshi – 1102867	
Supervisor: Dr. Goh Hock Guan	
Project Title: Energy Efficient Smart Lampposts for Smart Cities	

1. WORK DONE
- Successfully discovered how to write packets to flash memory
- Designed simple GUI with a simple T-junction as background image

2. WORK TO BE DONE
- Insert additional components to GUI, and try to display RSSI by changing color of lines between nodes
- Redefine hardware and software requirements of system
- Begin development of routing algorithm

3. PROBLEMS ENCOUNTERED
- Difficulty in understanding concept of routing algorithm and what it should do when parent node is down

4. SELF EVALUATION OF THE PROGRESS
- Report progressing as scheduled, confident of finishing before deadline

Supervisor's signature

Student's Signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 8
Student Name & ID: Ajay Kumar Doshi – 1102867	
Supervisor: Dr. Goh Hock Guan	
Project Title: Energy Efficient Smart Lampposts for Smart Cities	

1. WORK DONE - Developed prototype GUI which displays connectivity between nodes, RSSI values by using different coloured lines, battery icons and motion icon when motion is detected. - Prototype routing algorithm which could find and pair with parent node was developed
2. WORK TO BE DONE - Determine scenarios for system testing purposes - Redefine GUI to handle introduction of new nodes to the system - Estimate cost for development and commercialization - Consolidate usage of flash memory and RAM for each node
3. PROBLEMS ENCOUNTERED - difficultly in developing routing algorithm as it did not work most of the time - difficulty in developing GUI as there were many bugs which resulted in NullPointerException and IndexOutOfBoundsException - Text area in GUI does not refresh as new packets are received
4. SELF EVALUATION OF THE PROGRESS - Managed to learn more regarding routing algorithm and Java GUI programming and looking forward to applying them to the development of the system

Supervisor's signature   Student's Signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 10
Student Name & ID: Ajay Kumar Doshi – 1102867	
Supervisor: Dr. Goh Hock Guan	
Project Title: Energy Efficient Smart Lampposts for Smart Cities	

1. WORK DONE
- Ideal scenario to test system identified
- Cost for development and commercialization determined
- GUI employed the use of multiple threads via SwingUtilities to overcome problem of text area not refreshing as packet is received
- GUI shows alert box when new node attempts to join network
- DAT-ACK protocol redefined

2. WORK TO BE DONE
- Draw out system architecture
- Redefine flow charts for all functional modules of system
- Redefine routing algorithm to make choose parent based on hop count

3. PROBLEMS ENCOUNTERED
- Bug in routing algorithm where nodes would attempt to pair up with other nodes without parent. Also, the hop count values were too high and unrealistic.
- The usage of both flash memory and RAM proved challenging as there were a total of 4 pointers to manage

4. SELF EVALUATION OF THE PROGRESS
- Report nearly complete, a larger and the clearer picture of the system is obtained
- Improved understanding of multithreaded Java programming

Supervisor's signature

Student's Signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	Study week no.: 12
Student Name & ID: Ajay Kumar Doshi – 1102867	
Supervisor: Dr. Goh Hock Guan	
Project Title: Energy Efficient Smart Lampposts for Smart Cities	

1. WORK DONE - Depicted the system flow in the form of a flowchart - System testing was carried out - GUI was finalized by adding functionality of inserting packet values into database - Routing algorithm was finalized and can now choose parent based on hop count
2. WORK TO BE DONE - Configure all 14 nodes to prepare for demonstration - Perform soldering of components - Test GUI for few hours to ensure that no bugs are present - Prepare presentation slides - Finalize report
3. PROBLEMS ENCOUNTERED - Difficulty in soldering motion sensor and LEDs to breakout board
4. SELF EVALUATION OF THE PROGRESS - Confident in preparing a demonstration showing all functionalities of the system

Supervisor's signature

Student's Signature

Turnitin Originality Report

Energy Efficient Smart Lampposts for Smart Cities by Ajay Kumar Doshi

From FYP I & II reports submission (IIPSPW, FYP1, FYP2 - Jan 2016)

Processed on 13-Apr-2016 02:21 MYT
ID: 658413262
Word Count: 12116

Similarity Index	Similarity by Source	
6%	Internet Sources:	4%
	Publications:	2%
	Student Papers:	3%

sources:

1 — 1% match (publications)
Lavric, Alexandru, Valentin Popa, and Stefan Sfichi. "Street lighting control system based on large-scale WSN: A step towards a smart city", 2014 International Conference and Exposition on Electrical and Power Engineering (EPE), 2014.

2 — < 1% match (Internet from 12-Apr-2016)
http://blog.hemantthorat.com/

3 — < 1% match (Internet from 10-Aug-2013)
http://www.wikihow.com/Upgrade-Oracle-Java-on-Ubuntu-Linux

4 — < 1% match (Internet from 02-Mar-2016)
http://www.mdpi.com/1424-8220/14/12/24408/htm

5 — < 1% match (Internet from 16-Jul-2015)
http://www.ci.seaside.ca.us/Modules/ShowDocument.aspx?documentid=10514

6 — < 1% match (student papers from 07-Jul-2015)
Submitted to Regis University on 2015-07-07

7 — < 1% match (Internet from 23-Mar-2016)
http://www.superrade.info/author/admin/

8 — < 1% match (Internet from 02-Mar-2016)
http://www.mdpi.com/1424-8220/14/12/24408/pdf

9 — < 1% match (student papers from 06-Jul-2011)
Submitted to University of Westminster on 2011-07-06

10 — < 1% match (Internet from 21-Jun-2015)
http://www.slideshare.net/sureshca505/sdlc-tutorial