ABSTRACT

Street light monitoring system is meant to monitor the status of each street light in various geographical areas where data transmission in these areas is limited by geographical terrain and human population. This project is to build a working prototype of Street Light Monitoring System which utilized the GSM/GPRS network to one centre monitoring system. GSM/GPRS network is chosen because of its wide coverage. This project will cover all stage involved in building the system from building the local receiving, designing and implementing the transmission unit that based on GSM/GPRS network, setting up a web client to upload system status to monitoring system and creating an efficient graphical user interface to ease operator in monitoring the street light. In implementing this project, the whole stage is divided into two different time line. First stage will be more on research to identify the most effective and feasible method to achieve the transmission from transmission unit located on site to centre monitoring system in office. From the discussion the most suitable transmission network is GPRS as it more cost effective. Future works of this project will be based on GPRS network.

Formatted: Left: 1.38", Right: 1.18", Top: 1.18", Bottom: 0.98", Header distance from edge: 0.49", Footer distance from edge:

Formatted: Justified, Line spacing: 1.5 lines

~~TABLE OF CONTENTS~~

Formatted
Formatted
Formatted
Formatted
Formatted Table
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted

CHAPTER 1

INTRODUCTION

1.1 Background of study

In Malaysia almost every main street is fixed with lights. These~~is~~ street lights ha~~s~~ve increased the safety of the road user. Street lights are~~is~~ very important because they~~it~~ play~~s~~ big role in enhancing road users' vision at night. With the help of street lights road users can have better view of the surrounding areas. This will help the road users to asse~~ce~~ess the situation for any possible hazard~~s~~. ~~Usually s~~Street lights are usually fixed to increase the visibility of areas ~~enlighten an area around the~~along a -street~~. The visibility around the area is also increased~~. This reduces the possibility ~~of~~ of accidents and crimes in the area. ~~As the result~~Therefore the public safety ~~of public is also increased in general~~is increased. A better street light monitoring system will increase the public safety as written in one paper review by Malcolm Ramsay [1]. The ~~G~~general public realized the importance of street lights and failure to maintain street ~~the~~ lights in good condition will affect the public safety and tarnish the good name of street light operator.

As more streets will be built in~~ding~~ the future, more street lights will be located along the streets. Currently there are thousands ~~numerous~~ units of street lights operating throughout Malaysia including in the villages, housing estates and main roads that interconnecting~~the different~~ neighbouring states. These~~is~~ numbers

Formatted: Left, Line spacing: Multiple 1.15
Formatted: Line spacing: Multiple 1.15 li
Formatted: Font: 14 pt
Formatted: Font: 14 pt
Formatted: Left
Formatted: Justified

includes the functioning and faulty street lights. There is no limitation on where to place street lamplights. As long there is athe need for street lights and power supplies arey is available, street lights can be placed at anywhere. Therefore GSM/GPRS network is suitable as the data transmission medium as it has wide coverage. Because of GSM/GPRS has of these featurewider coverage, many telecommunication companyies have been providing Malaysian with wireless broadband which offer great internet mobility.

Formatted: Justified, Indent: First line: 0"

Formatted: Left

Formatted: Justified, Indent: First line: 0"

Formatted: Left

1.2 Problem Statement

It is undoubtedly increasing number of street light will give so many benefits to the publicLocating more street lights will reduce the risks of road accidents and increase the public safety. However another problem arises. When there are so many street lights, monitoring of all the lights willhas become a major challenges. A method is required to identify malfunctioning street lights. With current practices, street lights monitoring is done manually. Workers need to roam from one street to another street just to check condition of street if all the lights are functioning. Workers identify the faulty street lights then they will log reports. Only then the maintenance works can be carried out.

This The current monitoring practice is not efficient especially when the numberss of street lights are large rising. The areas that workers can covered by workers are limited. It is almost impossible for workers to check all the street lights in one night. In some cases, it takes few months before the light are being restored to its normal condition. This problem can be avoidedwill not occur if an effective street light monitoring system is implemented because the system will help the operator to monitor all street lights.

Formatted: Indent: First line: 0"

1.3 Objective

Formatted: Left

The objective of this project is to design a prototype of a monitoring system to monitor ~~the entire~~ street lights effectively. This system aims to eliminate human involvement in identifying faulty lights. ~~It is hoped w~~With this system, maintenance strategies ~~of~~ for street lights could be perfected. Th~~is~~e system will automatically notify the operator if any ~~lamp~~ street light becomes faulty. The system will provide the location of the street lights based on the identification number assigned to them ~~all light~~. The system is a real time monitoring system. ~~As soon as~~ When faulty street lights are detected ~~occurs~~, ~~the~~ operators are ~~is~~ notified to take necessary measures to restore the street light. The proposed system is shown in figure 1.

Comment [a1]: make sure pastikan tajuk tele

Formatted: Line spacing: Multiple 1.15 li

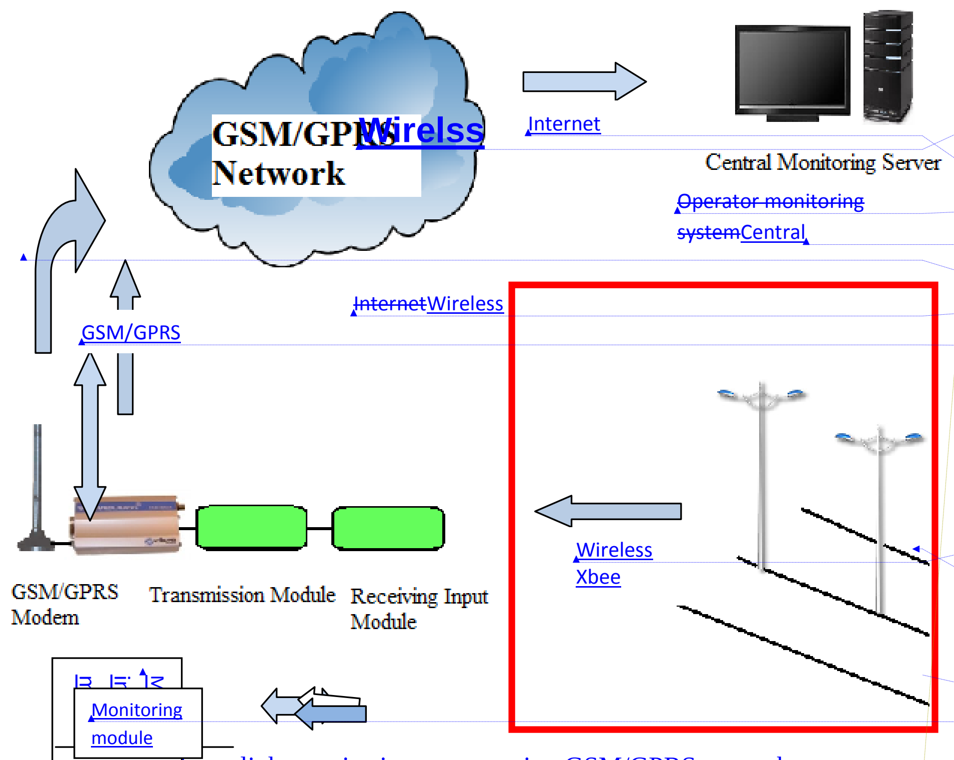


Figure 1: Street light monitoring system using GSM/GPRS network

The scope of this project is shown above. The section in the red rectangular is outside of this project's scope. The receiving input module will be interfaced to local street light monitoring system. The receiving input module will always be in standby mode to receive any data sent by transmitting modules of faulty street lights. The messages are then sent to transmission module so that operators in monitoring centre can be notified. Transmission module will establish and maintain GSM/GPRS connection. Since GPRS network can be connected to the internet, the data is passed to central monitoring server which is connected to the internet.

Formatted: Font: (Default) Times New Roman, 12 pt, Bold
Formatted: Font: (Default) Arial, 16 pt, Bold, Font color: Accent 2
Formatted: Font: 10 pt
Formatted: Font: 10 pt
Formatted: Font: 10 pt
Formatted: Font: 10 pt
Formatted: Font: 10 pt
Formatted: Font: 10 pt
Formatted: Font: 10 pt
Formatted: Space After: 0 pt, Line spacing: single
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: (Default) Times New Roman, 12 pt
Formatted: Font: (Default) Times New Roman, 12 pt
Formatted: Font: 10 pt
Formatted: Line spacing: Multiple 1.15 li
Formatted: Font: 10 pt
Formatted: Indent: First line: 0.5"
Formatted: Line spacing: Multiple 1.15 li

1.4 Scope of study

This project aims to build produce a prototype of street light monitoring system. To realise the system few factors need to be addressed properly. The first issue is to define how the status of each street light will be sent to the central node. This issue arises because there are so many street lights group together and only has one common node as the main receiver. The problem is made more complicated as we are using the wireless module to transmit the signal to the central node. Central node must be able to receive all the signals and have the capability to differentiate different signals from different street light. Firstly, the designing and implementation of transmission module will be emphasized in this project since it is the backbone of this project. Possible transmission network will be analyzed before the best option is chosen. GSM and GPRS are two different systems but they have same coverage as the GPRS is an enhancement to the GSM network. Once the suitable network is identified, hardware implementation stage is initiated with the designing process.

To help the operator to monitor large number of street lights, a graphical user interface (GUI) will be developed. The GUI will provide limited yet useful information to help the operator in identifying the status as well as the location of the faulty street lights. Ideally, the GUI will be updated with new information whenever faulty street lights are detected.

To build the central node which can receive transmitted signals, Xbee module needs to be integrated with a microcontroller. The microcontroller will provide the control signal for the Xbee module. Procedure to program the microcontroller to operate with Xbee module will be experimented in the future. This is necessary to make sure all data is received successfully.

Formatted: Justified

A considerably large number of incoming data need a proper server to store all the information before it can be accessed by GUI in the Street Light Monitoring Centre. The strategy to store the data in server will depend on what transmission network will be used. In the case of the GSM network the data is in the form of Short Message System (SMS) while in the GPRS network the data will arrive in the form of packets. Therefore both networks will require different strategies to store the data. Upon receiving the data, the data need to be forwarded to a server. The method used to forward the data can be selected from two different networks which are GSM or using GPRS. All the data must be forwarded to a server so that the operator could interpret the data correctly. Research on the two networks is necessary to determine which network is more suitable with our specification.

Formatted: Line spacing: single

Formatted: Line spacing: Multiple 1.15 li

Formatted: Centered, Line spacing: Multiple 1.15 li

Formatted: Centered, Line spacing: single

To help the operator to monitor the condition of thousand street lights, Graphical User Interface (GUI) would become very useful. With the help of GUI, the operator is not burdened by the task of analyzing thousand streams of input data. The design of the GUI will be made available once the system manages to send the status to a server.

Formatted: Font: 14 pt

CHAPTER 2

~~2.1~~ LITERATURE REVIEW

Formatted: Font: 14 pt

Formatted: Centered

Formatted: Font: 14 pt

2.1 Theory

Short Messaging system (SMS) and General Packet Radio Service (GPRS) are two options for the data transmission between ~~centre node~~monitoring modules and the central monitoring system on the operator side because they provide mobility. With these two networks, monitoring modules can transmit data ~~they can be operated~~ at any location as long as there is network coverage. Compared to telephone line, ~~SMS~~ GSM and GPRS provide wider coverage.

Formatted: Justified

Previously several monitoring systems based on the GSM network have been developed. One of such systems is Remote Monitoring in Agricultural Greenhouse Using Wireless Sensor and Short Message Services [2]. In the system, transmission of temperature readings~~raw data~~ from field is achieved with SMS which is a service provided by GSM network for data transmission. The data is sent to the mobile phone. When dealing with SMS and mobile phone, there ~~is~~ are certain limitations. Even though transmission using SMS is easier in the sense that data~~it~~ can be sent~~d~~ ~~the data~~ directly to ~~to~~ operator, the number of data received by the operator is limited. If too many information reach the operator's mobile phone at the same time, with limited mobile phone display and memory capabilities some data will be lost~~might be wasted~~.

This issue~~s~~ can be resolved if the data ~~are~~ is sent to operator's computer with the help of GSM modem. In this case, the data can be managed properly and analysis of the data is easier~~y to execute~~. However, GSM modem would introduce high overhead cost because each unit of GSM modem is expensive. Other limitation is the cost associated with every message sent to the~~he~~ operator. Every SMS sent to the operator will be charged accordingly. The maximum amount of data is 140 byte per message and each message would cost as minimum as 5 cent depending on the mobile carrier provider rate.

If SMS system is chosen to realize street light monitoring system, there is a trade off between system efficiency and cost. To minimize the cost, the system will have to send the information to the central monitoring system~~e~~ at much longer interval to reduce the number of SMS sent. ~~This s~~Situation becomes more severe when there are ~~is~~ maximum numbers of street lights. Because of every SMS is limited to 140 bytes, only limited amount of data can be fitted into one SMS. As more street lights are giving inputs to central~~e~~ monitoring system~~node~~, more SMSs are ~~is~~ required to ~~send~~ carry the street light statuses ~~the data~~ to the operator. As a result, the system will have higher ~~cost~~ operational cost.

GPRS is another alternative to ~~SMS~~ GSM ~~system~~network. This service is provided on top of GSM network. ~~It~~ GPRS is integrated to GSM network with ~~different~~ addition of hardware to the existing GSM~~hardware implementation~~ system. This service is also available where GSM network is available. GPRS provides a cheaper means of data transmission. Celcom Malaysia Sdn. Bhd. charge 10 cent for every 10 kb of data. The rate is much cheaper than SMS system. A ~~The~~ user will be charged on the basis of amount of data transmitted~~transmission data unit~~ instead of the number of transmission. This service provides much higher data rate. Another mobile network provider, Maxis offers connection speed which range from ~~56kbps~~ 20kbps and can reach up to ~~114kbps~~ 30kbps.~~depending on the service provider~~ Comparison between GSM and GPRS is provided in table 1. From table 1, a system which requires high degree of reliability and lower latency will prefer SMS system while GPRS system is more suitable for system that deals with greater volume of data.

Table 1: Comparison between SMS (GSM) and GPRS

	SMS (GSM)	GPRS
Cost depends on	Number of message	Data usage

Formatted: Space After: 0 pt

Formatted: Indent: First line: 0.02", Space After: 0 pt

Formatted: Left, Indent: First line: 0", Space After: 0 pt

Formatted Table

Formatted: Left, Indent: First line: 0.02", Space After: 0 pt

Rates	5 cent/message*	10 cent/10kb*
Reliability	99%	90%
Signals strength	Weak signal tolerance	Require good signals
Latency	Low	Unpredictable

*note: Rates varies with different service provider. The rates stated above are offered by Celcom Malaysia Sdn. Bhd.

GPRS can provide always on communication channel without incurring additional charges because as stated previously GPRS charges ~~is on the~~the users on the basis of amount of data ~~unit~~ transmitted. This feature can be manipulated so that data can be transmitted at any time. GPRS provides other useful features. With GPRS data can be uploaded to a ~~file~~ server. This~~It~~ is possible because GPRS network ~~provide connection based on the TCP/IP protocol~~sends data in the form of packets. These packets can be sent to a file server using internet connection. The internet connectivity is a great advantage because we have several options to capture the data.

~~The~~ Monitoring system based on GPRS will require GPRS modem for the monitoring module to upload the data~~system is not restricted by any hardware requirements such as the SMS system which require a GSM modem in order to work~~. However, monitoring system based on GSM will require two GSM modems to transmit data. When ~~the~~ condition~~status~~ of all street lights are uploaded to the internet, the data can be accessed anywhere~~everywhere~~ ~~where~~ as long ~~internet~~as internet connectivity is available. Unlike SMS system which requires two GSM modems, GPRS system will only need one modem for the monitoring module and accessing data will not require specific hardware like GSM modem. Even though GPRS seems to be very promising, implementation of GPRS in the street light monitoring system would require extra efforts. Monitoring based on GPRS is possible as it has been shown by previous works [3][4][5].

Formatted: Left, Indent: First line: 0", Space After: 0 pt
Formatted: Left, Indent: First line: 0.02", Space After: 0 pt
Formatted: Left, Indent: First line: 0", Space After: 0 pt
Formatted: Left, Indent: First line: 0.02", Space After: 0 pt
Formatted: Left, Indent: First line: 0", Space After: 0 pt
Formatted: Left, Indent: First line: 0.02", Space After: 0 pt
Formatted: Left, Indent: First line: 0", Space After: 0 pt
Formatted: Left, Indent: First line: 0.02", Space After: 0 pt
Formatted: Font: 10 pt
Formatted: Indent: Hanging: 0.1", Space After: 0 pt
Formatted: Justified
Formatted: Justified, Indent: First line: 0.5"
Formatted: Justified
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto

2.2 Performance of GSM and GPRS

GSM network is more reliable than GPRS network. Reliability for both networks is measured by calculating the differences between total number messages sent and the number of delivery reports received. Then it is divided by total number of messages sent. The reliability of GPRS network is lower than GSM network. GPRS network would require good signal strength in order to send data. Unlike GPRS, GSM network can still send data even when signal strength is low[6].

The latency is just the estimation of time between each message is sent and a delivery report is received. GSM system has higher reliability as well as low latency because before the message is transmitted to a recipient, a link is established between the sender and receiver. As a result the data is relayed directly to the recipient.

GSM network has low latency because of it is based on circuit switch network unlike GPRS which is based on packets switched networks. The packets switched networks latency depends on the routes each packet takes to reach the end devices. Each packet can take different routes and reaches the end devices out of order. Latency of GPRS network is unpredictable. At peak hours when many users are using GPRS services, there are limited numbers of free timeslots to serve large number of user requests. Therefore the latency of the system increased significantly.

. The main attraction for GPRS network as transmission medium is GPRS has low overhead and operating costs. The operational cost for GPRS is lower compared to GSM. One SMS(GSM) can only support 140 bytes of data and each message would cost 5 cent. Since the number of street lights in our monitoring system could reach thousands, the data streams are considerably large and GPRS services will have significantly low operating cost. Thus, GPRS is more preferable than GSM. To implement monitoring system using GSM, two GSM modems are required. If the author chose GPRS network, the monitoring system will only need one GPRS modem.

Formatted: Font: Bold
Formatted: Indent: First line: 0"
Formatted: Font: Bold
Formatted: Font color: Auto
Formatted: Font color: Auto
Formatted: Font color: Auto

GPRS reliability can be tolerated to a certain extents. Data transmitted contains the condition of street lights. The size of data is small and since the cost is cheaper than GSM network, data transmission can be done more frequently compared to SMS. Thus after certain tolerable delays, the data will finally reach the central monitoring system. Since there is no significance in receiving the faulty street lights conditions within few minutes or an hour, GPRS is very promising solution to be implemented.

Formatted: Space After: 0 pt

2.3 Socket Communication

Formatted: Font: Bold

Formatted: Indent: First line: 0"

Socket is a concept that represents one endpoint of two way communication link running on the internet network. In socket communication a server is compulsory. A server is computer which store resources required by clients. A server can be connected to many clients if the server run multi-threaded program. To connect to a socket server, a client must know the IP address of the server and port number that server is listening.

Formatted: Justified

Formatted: No underline, Font color: Auto

IP address is a unique four byte numbers assigned to each computer when connected to a network while port number is a way to identify a specific process to which an Internet or other network message is to be forwarded when it arrives at a server. Wrong port number will direct the data to wrong application and incorrect IP address will cause the data to reach false destination. As a result, socket connection cannot be established. Once connection is successful, a server will create a new thread to the client. This thread is used for data exchange between the server and the client.

There are several type of socket connections, two of them are UDP(User Datagram Protocol) and TCP(Transmission Control Protocol) sockets. TCP socket is more reliable as it confirms connection to a server before a client can send data. A communication link is established using handshaking method. A client will request a connection. If the server can serve the client at that moment, server will transmit acknowledge signals. Prior receiving the signals, then client can start sending data.

Socket communication can offer flexibility in receiving the data on the server side. Other than that, the purchased GPRS modem can only support TCP or UDP sockets. Latest version of GPRS modem may provide better application to ease transmission of data.

2.4 Active Server Page

ASP(Active Server Page) is used to create and run dynamic web applications. With ASP web page ones can combine HTML pages, script commands and other components to create interactive web pages and powerful web-base applications[7]. Creating an interactive web page is very promising idea since it will allow a microcontroller to send data at any time and the server can response to a microcontroller. Other advantage of ASP is it runs in IIS(Internet Information Services). IIS is a component of Microsoft operating systems. Using IIS, any personal computer can be configured as a server. To serve high traffic of request, it is advisable to get a decent server. However, IIS program provides a testing ground for web page developers. Lastly, ASP pages are known to work well in different web browsers.

Formatted: Justified, Indent: First line: 0"
Formatted: Justified
Formatted: Justified, Indent: First line: 0"
Formatted: Font: Bold, No underline, Font color: Auto
Formatted: Font: Bold
Formatted: Justified
Formatted: No underline, Font color: Auto
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: No underline, Font color: Auto
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: No underline, Font color: Auto
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: No underline, Font color: Auto
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: Justified, Indent: First line: 0"
Formatted: Justified

Formatted: Centered, Line spacing: single

Formatted: Justified

Formatted: Centered, Line spacing: single

CHAPTER 3

~~3.1~~ METHODOLOGY

3.1 Procedure Identification

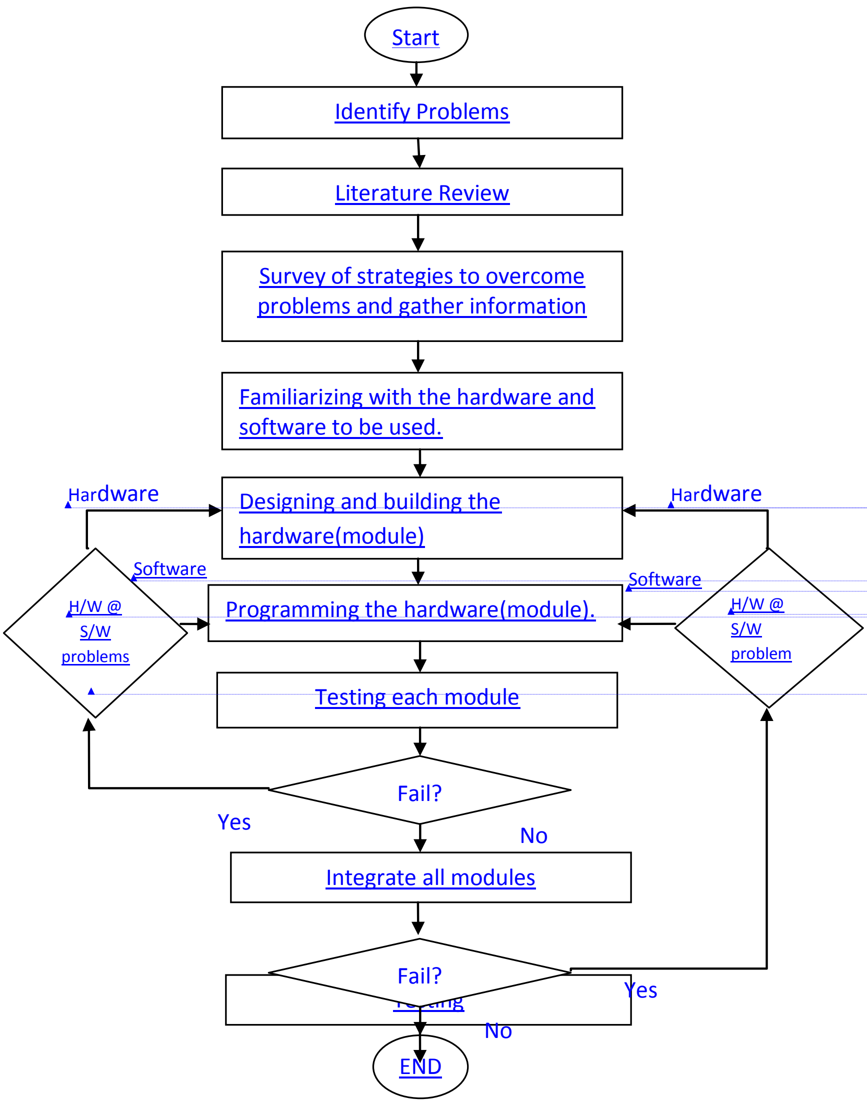


Figure 2: Project Flow

Formatted: Centered, Line spacing: single
Formatted: Font: 14 pt, No underline, Font color: Auto
Formatted: Centered
Formatted: Centered
Formatted: Centered
Formatted: Centered
Formatted: Centered, Line spacing: single
Formatted: Line spacing: single
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: 8 pt
Formatted: Font: 8 pt
Formatted: Centered
Formatted: Font: 9 pt
Formatted: Centered
Formatted: Centered, Line spacing: single
Formatted: Centered
Formatted: Centered, Line spacing: single
Formatted: Centered
Formatted: Font: (Default) Times New Roman, 12 pt
Formatted: Centered

3.2 Planning

The whole FYP project is divided into two separate timelines. FYP I focuses more on finding the correct strategies to solve the problems. The available options to achieve targets are surveyed first to minimize the possibility of making wrong decisions when considering difficulty level, available resources and effectiveness of the solutions. Useful knowledge about the hardware and software are acquired. The knowledge is necessary as it will be needed in realizing the prototype. Briefly described, the scopes of this project in FYP I are to gather information and get familiarize with the software and hardware. Initial design of the hardware and software will be useful for FYP II. FYP I is also a time for hardware acquisition. The author identified the hardware to be used and select the best hardware suitable for the project. All the major hardware is already received.

FYP II will focus on building the hardware and programming the system so that all hardware and software could be integrated successfully. By the end of FYP II, the system will have one data receiving unit, one transmission unit, a server and graphical user interface (GUI) installed on centre monitoring system station.

Formatted: Justified

Formatted: Font: Not Italic, English

3.3 Tools and Equipment

Formatted: Font: Bold, No underline, Font color: Auto

a) CCS Compiler

This software provides the user the capabilities to embed an application into a microcontroller. It enables application designer to write the source codes before compiling into HEX file. Then the code is ready to be embedded in a microcontroller. The advantage of using CCS C is it enables a microcontroller to be programmed using C language.

b) PICkit 2 v2.61

This software is used to download HEX file which is the output of CCS Compiler into a microcontroller. This software is provided for free. In order for HEX file to be downloaded, this program need to be used with a hardware called PIC programmer.

c) Visual Basic 6

Visual Basic 6 is event-driven programming language and Integrated Development Environment from Microsoft. VB enable ones to develop Graphical User Interface (GUI) applications, toolbox to work with database and simple yet comprehensive programming language. The application developed in this project will be based on Visual Basic 6. This will include socket application for the server and database access to update the status of faulty street lights.

d) Internet Information Services(IIS)

IIS is a free component of windows operating system. This program will be used to provide a testing environment to develop a web page and setting up a server. This server is aimed to receive updates from monitoring module at remote location. Besides, a web page will be created to display the latest updates of faulty street lights. This web page can be accessed from anywhere using internet.

e) Q24 GPRS modem from Mobitek

Q24 is a GPRS modem with a serial port. Using the serial port, it can be connected to a microcontroller using USART module of a microcontroller. The modem complies with GPRS class 10 and is embedded with TCP/IP stack of WIPSoft version 2.01. The TCP/IP stack is necessary to reduce the efforts of writing a TCP protocol into the microcontroller. With embedded TCP/IP stack in the

Formatted: Justified

Formatted: Font: Italic

Formatted: Font: Italic, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, Not Bold, No underline, Font color: Auto

Formatted: Font: Not Italic, No underline, Font color: Auto

Formatted: Font: Not Italic, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: Font: Not Italic, No underline, Font color: Auto

Formatted: Font: Not Italic, No underline, Font color: Auto

Formatted: Font: Italic, No underline, Font color: Auto

Formatted: Indent: First line: 0.5"

modem, more instructions can be stored in microcontroller since more spaces are available.

3.4 Project work

a) Interfacing GPRS modem

In order to utilize GSM/GPRS network structure to transfer data, the data is modulated using a GSM/GPRS modem. The information to be sent from transmission module will go to modem. A Modem and a microcontroller implement different voltage system. Thus, an interface circuit is required to convert the voltage signals between a modem and a microcontroller. The interfacing is achieved using Max232 IC. The datasheet for Max232 IC is provided in appendix I.

b) Setting up a server

A server requires an identity so that any computer on the network can connect to the server. A server can get this identity by using a hostname. Log on to http://www.no-ip.com/ and register a hostname. For this project, afiqfyp.no-ip.org is the hostname registered for the server.

After registering with www.no-ip.com, download and install Dynamic DNS Updater Clients on the server. Upon successful installation, run the updater. Figure 3 shows the updater program. The program will continuously update the IP address of the server. From figure 3, the IP address of the server is 121.120.230.71. This IP address is then saved by www.no-ip.com server.

Formatted: Font: Bold, No underline, Font color: Auto
Formatted: Font: Not Bold
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: Justified, Indent: First line: 0.5"
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: No underline, Font color: Auto
Formatted: Justified
Formatted: Line spacing: Multiple 1.15 li
Formatted: Font: Italic, No underline, Font color: Auto
Formatted: Font: Italic
Formatted: Justified
Formatted: No underline, Font color: Auto
Formatted: Indent: First line: 0.5"
Formatted: No underline, Font color: Auto
Formatted: No underline, Font color: Auto
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: No underline, Font color: Auto

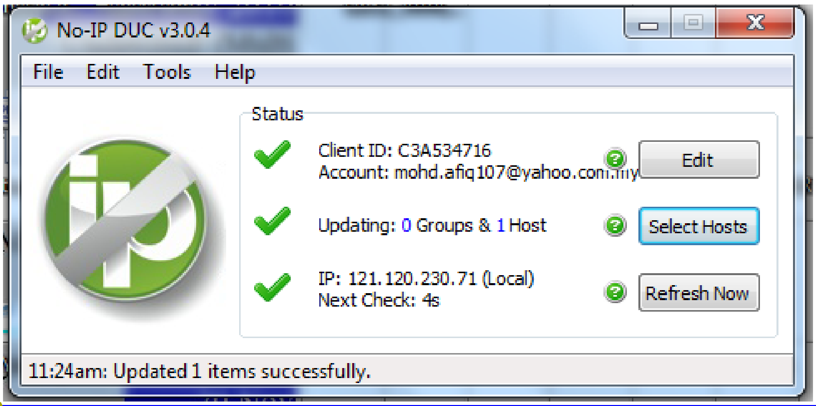


Figure 3: Dynamic DNS Updater client

If the updater works successfully, the server is now visible to the internet. To test connection to the server, open a command prompt in another computer which is connected to internet. Then type ping afiqfyp.no-ip.org. If no request timeout message is shown in the command prompt, the server is accessible. In figure 4, the server can be accessed from the internet. Note that address of afiqfyp.no-ip.org has the same IP address as in the updater program which is 121.120.230.71.

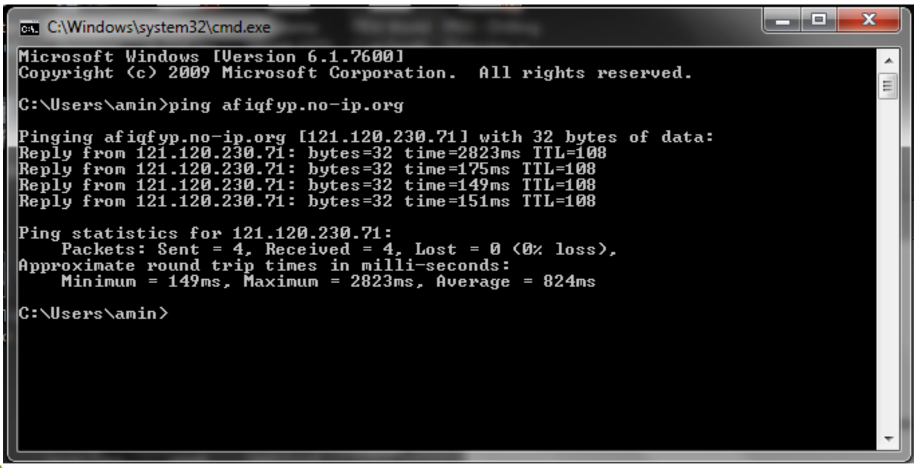


Figure 4: Command prompt windows

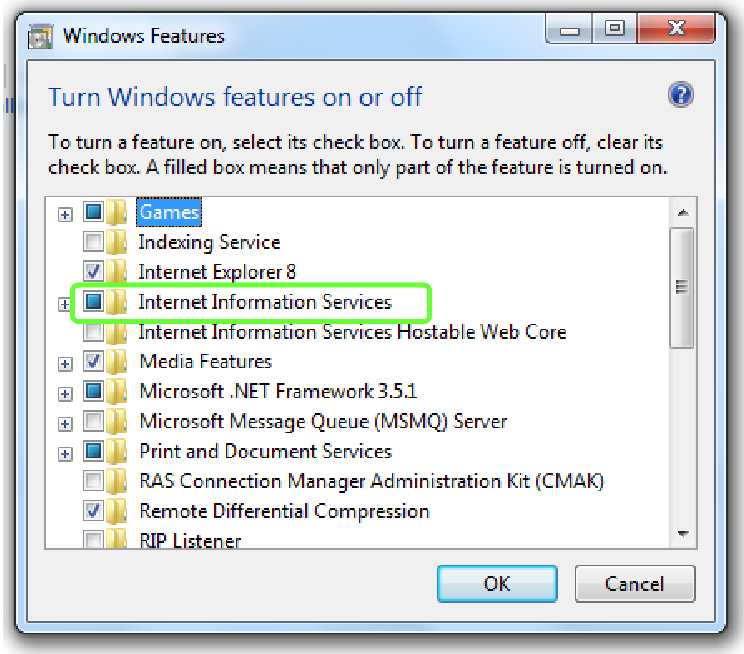
Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Indent: First line: 0.5"

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Now proceed with setting up the server. Actually any computer can be configured as a server. It is possible with the application called Internet Information Services (IIS). IIS is a component of Microsoft Windows operating system. By default IIS feature is turned off. To turn on IIS, go to Control Panel > Programs >Turn Windows Feature On or Off. The instruction provided is meant for Windows 7 of Microsoft operating system. The commands might differ slightly from different version of Windows.

Formatted: Indent: First line: 0.5"



Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Figure 5: Window features

Figure 5 shows the windows that will pop up after clicking Turn Windows Feature On or Off. Browse the list and find Internet Information Services and then select the options. Then click OK. To open IIS, go to run and type inetmgr. The IIS manager Window is shown in figure 6.

Formatted: Indent: First line: 0.5"

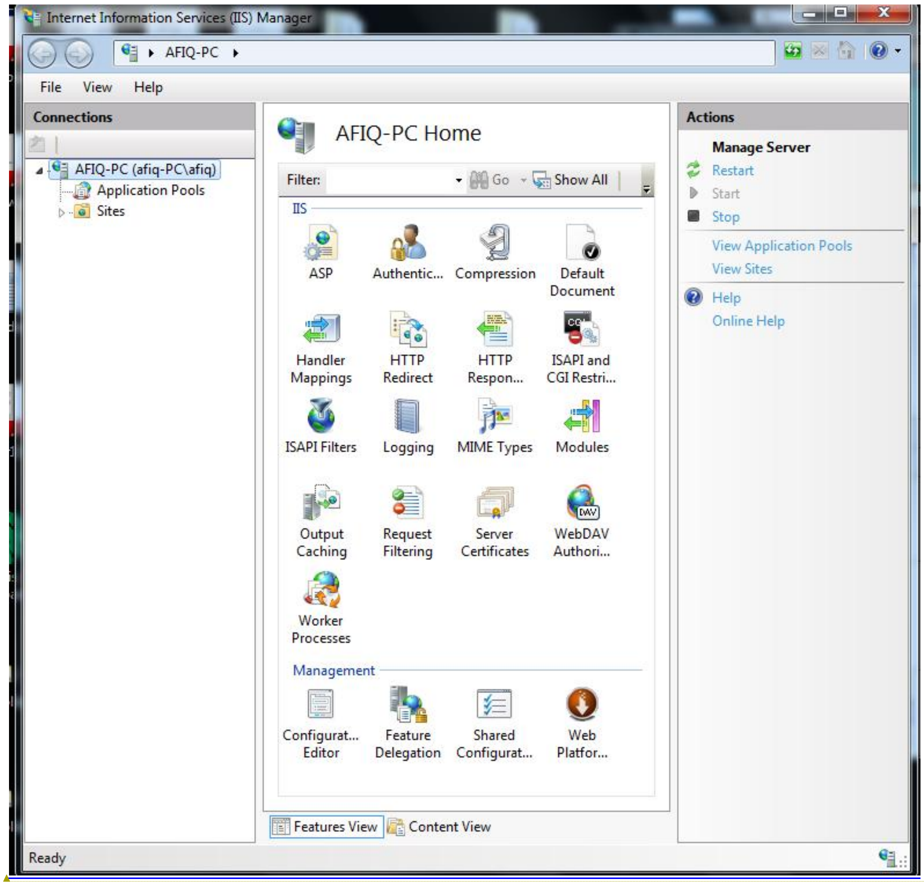


Figure 6: IIS Manager window

Then ones need to type in the hostname registered earlier so the server can bind the information it receives in the request from clients and which web page to return to client. Click Sites and then select Default Web Site. On the right column, select Binding.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Justified, Indent: First line: 0.5"

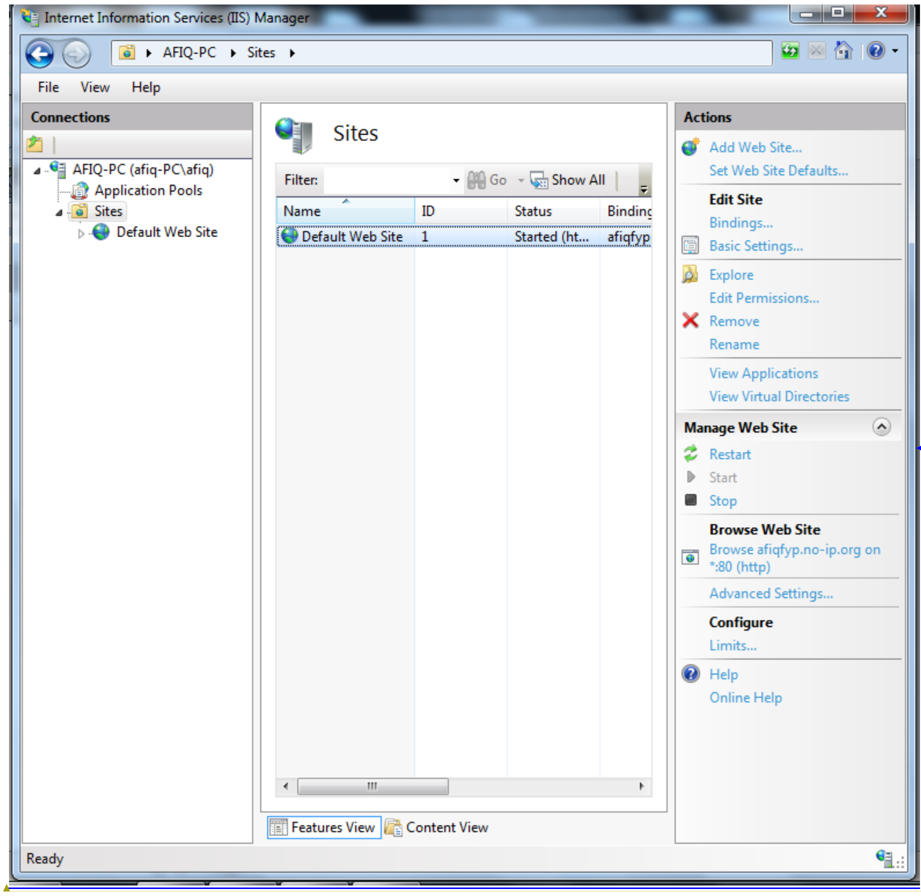


Figure 7: Binding setting

After clicking that option, new windows will come out. Click Add New. Then Fill the afiqfyp.no-ip.org in the Host Name, Port select as 80 and Type as HTTP. Then click OK. The complete binding step is shown in figure 8.

Now the server is ready to serve client request. All the web page created can be located in the default directory of IIS. The location of all web pages is in C:\inetpub\wwwroot.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified

Formatted: Justified, Indent: First line: 0.5"

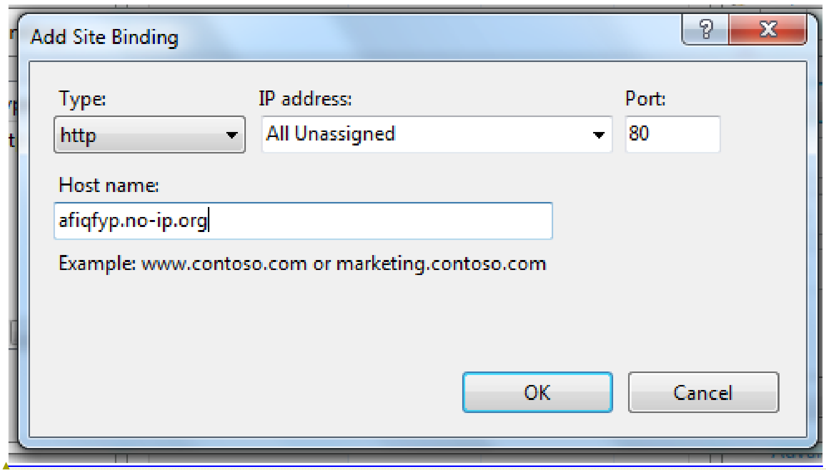


Figure 8: Add site binding

c) Developing Web page

Web page is an essential part of this project. The data of street lights condition is first stored by the server. These updates are then displayed to operator using a web page. Viewing using a web page is chosen as it will provide ease of access to operators as operators can make use of vast availability of internet connection to view the updates.

Web page will display very basic information about the faulty street lights such as identification number and the time when faulty signals are received by the server. The information is available on default page when browsing to afiqfyp.no-ip.org. The main page of street light monitoring system is shown in figure 9.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

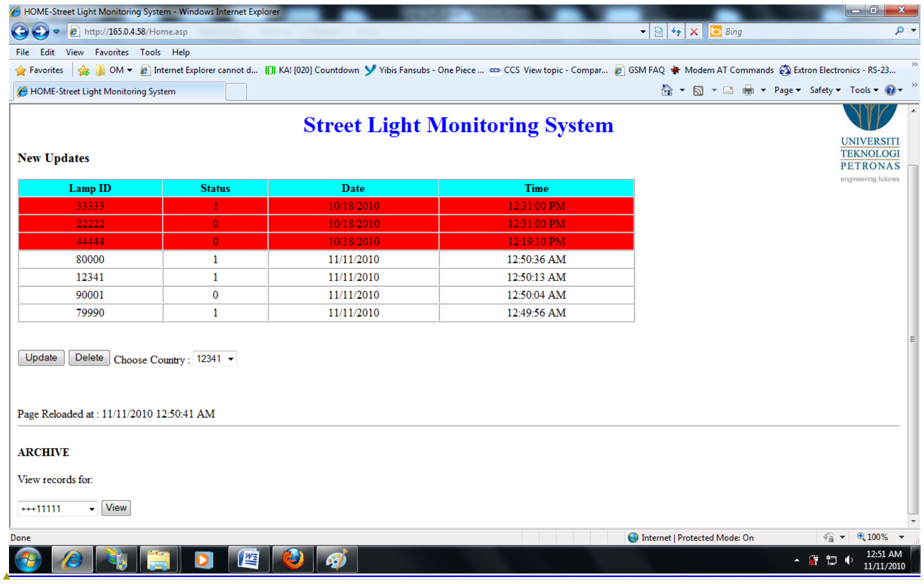


Figure 9: Main page of street light monitoring system

The first column provides the ID of street lights and in status column values of 1 indicates a street light is functioning normally. To extend the functionality, operators can delete any entry using the Delete button. If there is a need to input data manually into the system, operator can click Update button. Figure 10 shows the update page. Operators can input ID number of the street lights and the report the conditions whether on or off.

Since the server receive all inputs and store them in a database, operators can view history of any street light in the achieve section. In figure 11, history of street light of ID 1111 is shown. From the page, information about the light condition, date and time of arrival of particular data is also available.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

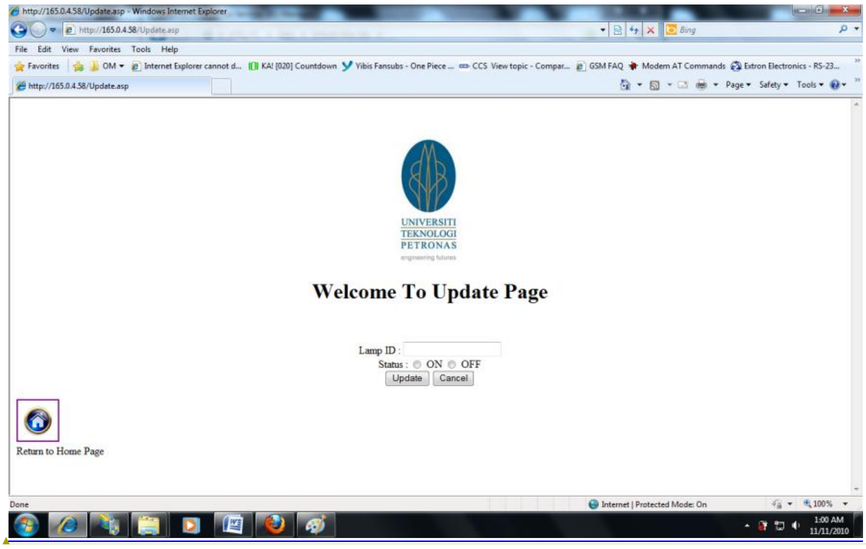
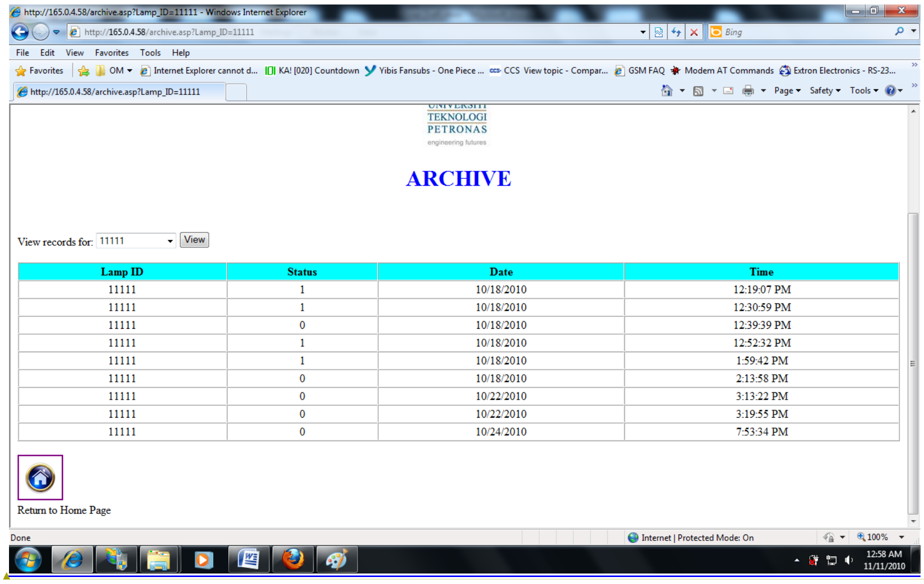


Figure 10: Update page of street light monitoring system



Lamp ID	Status	Date	Time
11111	1	10/18/2010	12:19:07 PM
11111	1	10/18/2010	12:30:59 PM
11111	0	10/18/2010	12:39:39 PM
11111	1	10/18/2010	12:52:32 PM
11111	1	10/18/2010	1:59:42 PM
11111	0	10/18/2010	2:13:58 PM
11111	0	10/22/2010	3:13:22 PM
11111	0	10/22/2010	3:19:55 PM
11111	0	10/24/2010	7:53:34 PM

Figure 11: Achieve page

Example of viewing history of street light with ID 11111. The complete web page file is provided in the appendix IV.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

d) Writing socket application

Sending data to central monitoring server cannot be achieved using web page request because the modem TCP/IP stack does not support HTTP client application. The modem can only support TCP client application which equivalent to TCP socket application. Thus in preparing the server to be able to receive input from monitoring module which sends data via modem, the server must listen to certain port and run an application to cater for socket connection.

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

The application is designed and developed using visual basic 6. Visual basic 6 has Winsock tools which made the process developing socket application easier. With Winsock control toolbox, ones can create socket application whether TCP or UDP type. In this project TCP is selected over UDP because it offers better reliability. Connection will be established once the server acknowledges request from client.

The socket application will listen to any port selected by the programmer. Port 9000 is selected as it is not assigned to any specific purpose according to the standards. The modem needs to be told to establish connection using the same port. The program will display the IP address of client connected to the server as connection is established. Data received by the application will be manipulated before being displayed. Raw data is received in segments so socket application need to re-arrange the data before can be displayed and updated in database. The database provides the information to be displayed in the web page.

Figure 12 shows the socket application before starting to listen for incoming connections. To start listening, click start button and to stop listening, click stop. The complete code for the socket application is provided in appendix II.

Formatted: No underline, Font color: Auto

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

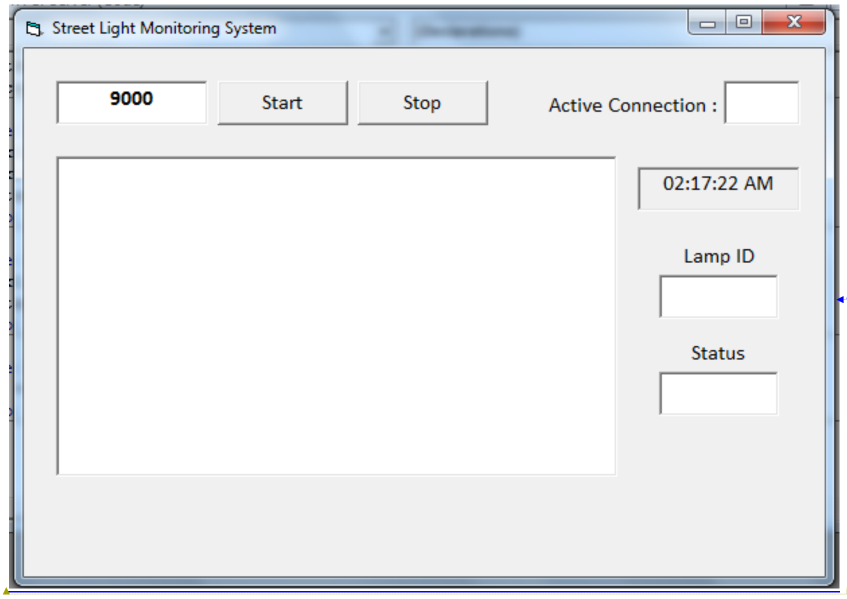


Figure 12: Socket application

e) Connecting to server and sending data

Connecting and establishing connection to the server is the task of transmission module. It connects to a server by issuing AT commands to a GSM/GPRS modem which in this project Q24 GSM/GPRS modem. Instead of writing own TCP/IP stack, it is embedded in the modem firmware. Thus, making the job of connecting to the server is easier. The modem has its own AT set to implement GPRS connection. These proprietary instruction sets can only be used with modem based on Wavecom chip. However, different modem will have different version of AT instruction sets. Modem Q24 GSM/GPRS modem is implementing WIPSOFT version 2.01.

In order to establish GPRS connection, some information is essential. Ones need to have details of Access Point Name(APN) of service provider, IP address of the server and port number which the server listen. The testing was done using a GPRS enabled Celcom SIM card. A SIM card or *Subscriber Identity Module* is a portable memory chip used in some models of cellular telephones. The SIM card

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto
Formatted: Indent: First line: 0.5"
Formatted: Font: Not Italic
Formatted: Font: Italic, No underline, Font color: Auto
Formatted: Justified, Indent: First line: 0.5"

holds information about cellular account of subscriber. APN identifies an IP packet data network (PDN), that a mobile data user wants to communicate with. Each GPRS service provider will have its own APN.

The details required are:

IP Address: 010.128.001.242

Port Number: 8080

APN Name: celcom

The author tried to connect to a proxy server of GPRS provider. In the actual implementation, the connection is established to another specific server with different IP address and the port number is 9000. The APN name remains the same as it is the bearer or provider of GPRS connectivity instead of the target destination of a connection. Proxy server is used by the GPRS provider as a gateway to the internet. To access web page using GPRS, a client will send a request for a web page and the proxy server will fetch the page then forward it to the client.

The setup for the test is shown in figure 13. The SIM card is inserted into the modem and the modem is connected to a laptop which will issue the commands. Sending instruction to the modem is done using HyperTerminal application which comes as standard component of older version of Windows operating system. The snapshot of HyperTerminal is provided in figure 14. Instruction must be typed in the HyperTerminal window and press ENTER to send the instruction to the modem.

Formatted: No underline, Font color: Auto

Formatted: Justified

Formatted: Justified, Indent: First line: 0.5"

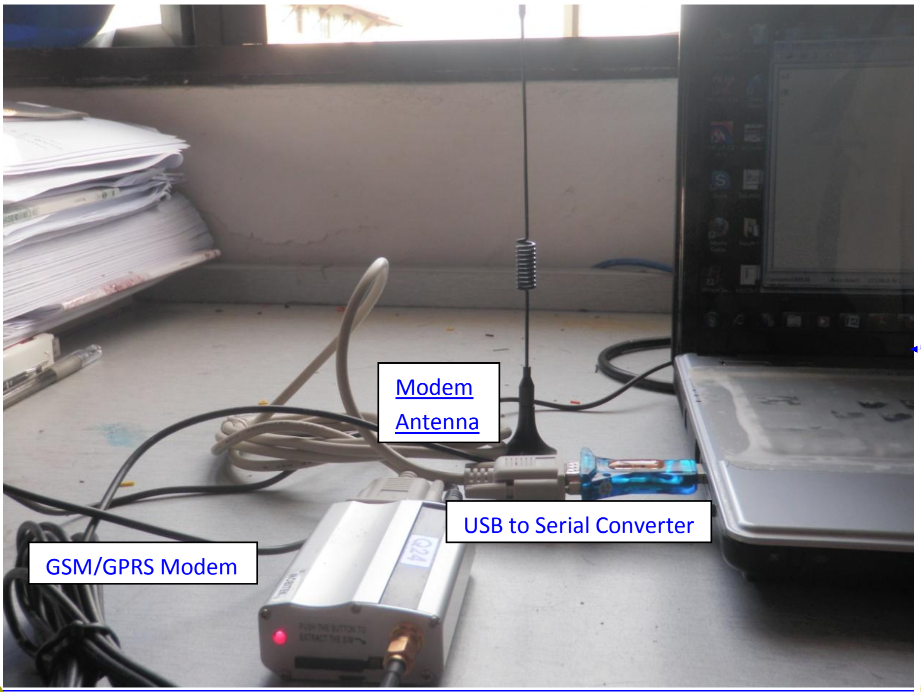


Figure 13: Test setup to connect to Celcom GPRS proxy server

Socket
Application

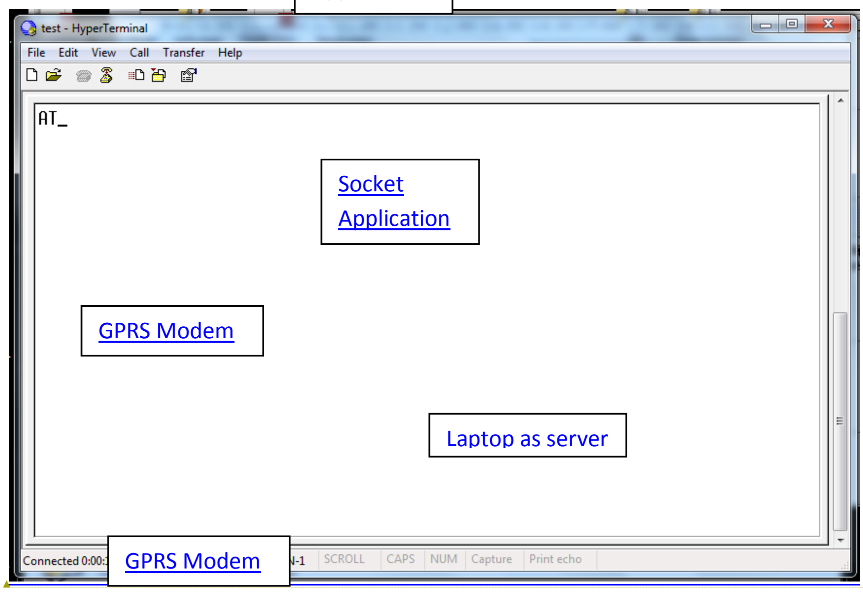


Figure 14: HyperTerminal window

Laptop as server

Formatted: Justified

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

The first step to establish connection is to start TCP/IP stack of the modem. To verify whether the modem is connected and is ready to receive command type AT in HyperTerminal as in Figure 12. The modem should response OK when it is ready. Then type in AT+WIPCFG=1 in HyperTerminal. This command will start TCP/IP stack of the modem. TCP/IP stack is required to establish connection and it facilitate transmission of data using TCP socket. Modem should response OK.

Next ones need to open GPRS bearer. The command is AT+WIPBR=1,6. 1 is assigned to Open command and 6 is the identification of bearer which is GPRS. The modem will response OK. The full reference of commands is provided in user manual of Wavecom modem with WIPSoft Version 2.01.

Formatted: Justified, Indent: First line: 0.5"

Now enter the APN Name before attaching to GPRS network. The command is AT+WIPBR=2,6,11,"celcom". Then issue command of AT+WIPBR=4,6,0 to start GPRS bearer. Modem will take a while to connect to GPRS network. Then modem should response OK. At this moment GPRS network has assigned one unique IP address to the modem. Use AT+WIPBR=3,6,15 to read the IP address assigned to modem.

The modem is ready to create socket connection to the server. Use AT+WIPCREATE=2,1,"010.128.001.242","8080". The command requires IP address of the server and port number. When modem response OK followed by +WIPREADY 1,1 , a socket connection to the server is successfully created. Proceed to next instruction to start sending data.

Type AT+WIPDATA=2,1,1 to start sending data. When CONNECT is receive one can type any data and the data is sent to the server. After sending each data, press ENTER to insert new data. When all data is sent to server, modem is required to be back in AT mode before attempting to close the socket connection. Type +++ and wait for a while before the modem is ready in AT mode.

Closing the socket connection is straight forward. First need to close IP service, then stop GPRS bearer and lastly close GPRS bearer. Following commands are issued sequentially.

AT+WIPCLOE=2,1

AT+WIPBR=5,6

AT+WIPBR=0,6

All these command are the backbone of transmission module. The complete program for the transmission module is written in CCS and written to microcontroller using PICKit 2 ver2.61. The complete command is provided in appendix III.

f) Establishing GPRS connection and sending data using prototype transmission module

In this progress, author has integrated all components and module developed so far to achieve data transmission using GPRS network. In order to achieved data transmission as well as displaying receive data to the operator, author has came up with own server connected to GPRS network. A laptop is connected to GPRS network by inserting GPRS enabled SIM card in a modem as shown in the figure 15. A standard laptop is configured as a server by running a socket application. After connecting to the network and run the socket application, the laptop can now be able to receive data from transmission module. The address of the server is obtained by typing ipconfig in the command prompt in the server. The IP address is needed by transmission module to connect to server.

Formatted: Justified

Formatted: No underline, Font color: Auto

Formatted: Font: Italic, No underline, Font color: Auto

Formatted: Font: Italic

Formatted: Justified, Indent: First line: 0.5"

Formatted: No underline, Font color: Auto

Formatted: Font: Not Bold, No underline, Font color: Auto

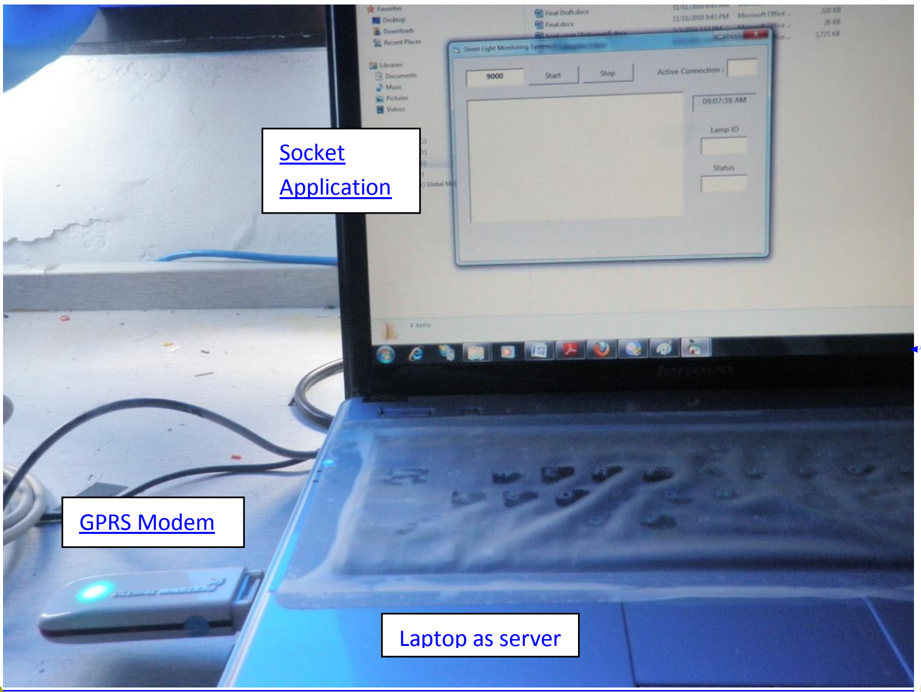


Figure 15: Testing server setup

Transmission module has primary objective to transmit data to server. In this section, for testing purpose only, the transmission module starts to connect to the testing server when the switches change states either on to off or vice versa. The complete transmission module setup is shown in figure 16.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified

Formatted: Justified, Indent: First line: 0.5"

Formatted: No underline, Font color: Auto

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: No underline, Font color: Auto

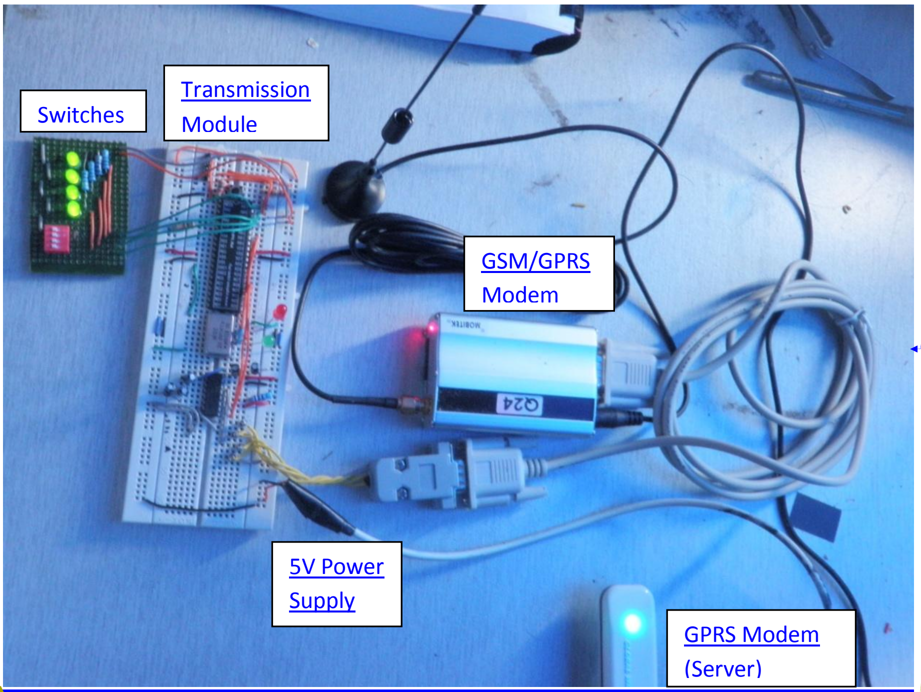


Figure 16: Test setup for prototype of transmission module

Each switch is already assigned with one identification number. After connection is established, transmission module sends the status or condition of each switch. Then transmission module closes the socket connection and will be able to send the data again when another switch change states which triggers connection establishment.

Upon receiving data from transmission module, the socket application updates the database. The updates can be viewed using any web browser. The web page is refreshed continuously to make sure the latest updates are available to operators. Once again the purpose of this test is to test ability of the transmission module prototype to establish connection, send data and then close connection.

Formatted: Justified, Tab stops: 5.71", Right

Formatted: Font: (Default) Times New Roman, 12 pt, Bold, No underline, Font color: Red

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: Justified

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold, Font color: Auto

Formatted: Justified, Indent: First line: 0.5"

In real implementation whereby the transmission module is required to send updates from hundreds of street lights, it will receive the data serially from receiving input module. The actual circuit is not being tested but the concept is transmission module will only manage the connection to server which means it will establish socket connection and then close connection after receiving unit finish sending all data.

Formatted: No underline, Font color: Auto

The task of managing incoming data from street light transmitters and when to send the data is the responsibility of receiving input module. It will provide signals for transmission module to start establishing connection. When link is established to server, transmission module will signal receiving input module to start transmitting data. Coordination between the transmission module and receiving input module is crucial.

The block diagram of the integration receiving input module into transmission module is shown in figure 17. In figure 17, transmission module and receiving input module will share one GPRS modem. Both of modules will take turns to issue instruction or data to modem. Receiving input module will be the master by providing synchronizing signals transmission modules.

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: No underline, Font color: Auto

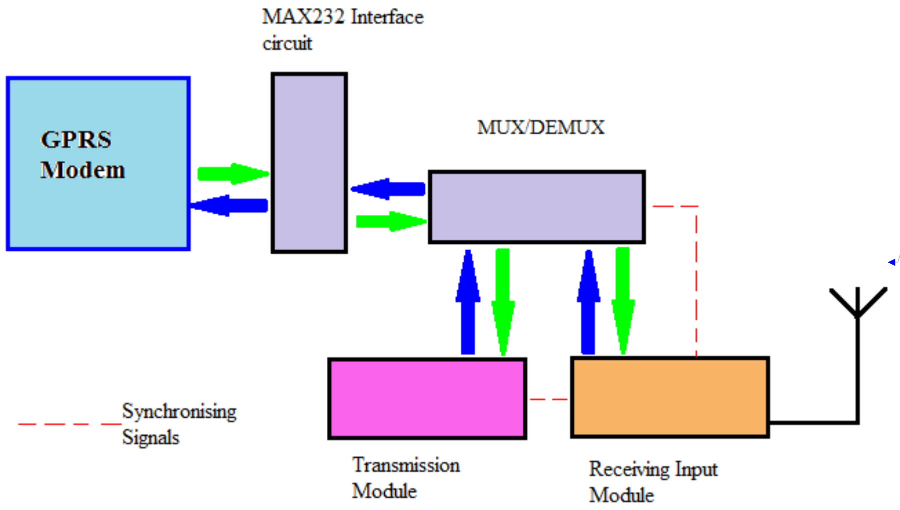


Figure 17: Integration of transmission module and receiving input module

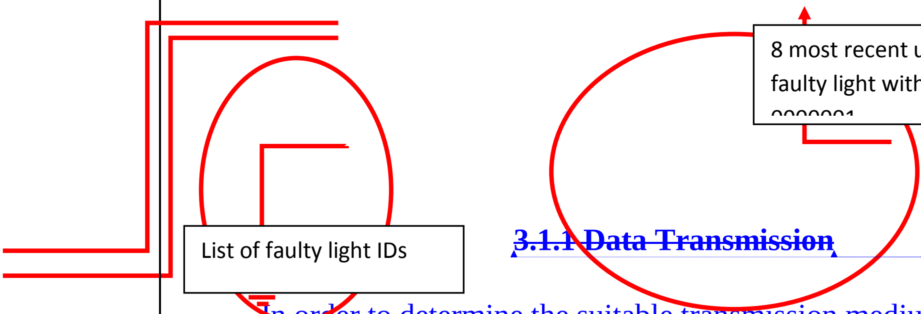
Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Centered, Line spacing: Multiple 1.15 li

Formatted: Line spacing: Multiple 1.15 li

Formatted: Centered, Line spacing: Multiple 1.15 li

3



3.1.1 Data Transmission

In order to determine the suitable transmission medium whether to choose SMS or GPRS, the author has reviewed some of technical papers and journals from the IEEE organization. This will help the author to be aware of possibilities and limitation of both systems. To be exact, up until the completion of this report, four journals are reviewed and the number will be increased in the future.

With the sufficient background about SMS (GSM) and GPRS services, the suitable transmission services will be chosen. After that, the data transmission section in the system will be specified to be integrated with the chosen service. To help the operator to monitor and analyze the status of all street lights, a suitable graphical user interface will be developed.

3.1.2 Central node and transmission unit

Central node refers to a unit of Xbee module fixed to a microcontroller in order to capture the status of all street light in the wireless network. Transmission unit in this project refer to GSM/GPRS

Formatted: Centered, Line spacing: Multiple 1.15 li

Formatted: Line spacing: Multiple 1.15 li

Formatted: Centered

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto application board from Rabbit. The stage of hardware implementation will commence once the system specification procedure is completed.

Formatted: Font: 14 pt

To build the hardware for the system, system specifications are required. Then list of possible components and equipments to be used in the project are specified. While waiting for the components and equipments to be available, designing the prototype will be done using simulation software. By the time the all the components are available, the design of the hardware are completed, the source code of the programs are available and the specification of the system must be possible to be implanted with the components. If the programs or the hardware designs cannot work properly, modifications are required. Xbee module and GSM/GPRS application board provide complete resources on their website. This manuals and datasheets are very important in the designing stage.

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

The strategy used to build the hardware is to test every completed section separately. After one section is built, some tests are carried out to make sure the section works as the specification. This is a standard procedure for all hardware section. In the end when all sections are working successfully, all the sections are merged together and the final test will determine the functionality of the hardware.

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

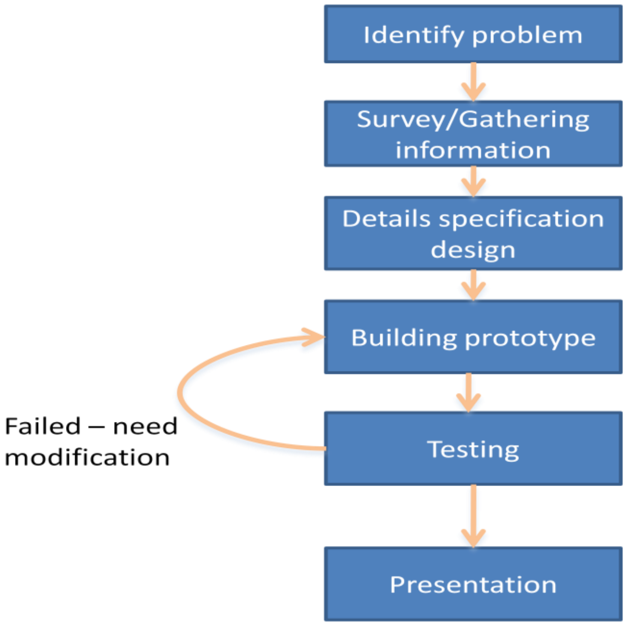


Figure 1 : Project general process flow

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

CHAPTER 4

4.1 RESULT AND DISCUSSION

4.1.1 GSM and GPRS evaluation

After doing paper reviews on the suitable service to relay data to the system, it is decided that GPRS is preferable. The comparison between SMS (GSM) and GPRS are provided in the table 1 .

	SMS (GSM)	GPRS
Cost depends on	Number of message	Data usage
Rates	5 cent/message*	10 cent/10kb*
Reliability	99%	80% - 90%
Signals strength	Weak signal tolerance	Require good signals
Latency	Low	Unpredictable

*note: Rates varies with different service provider. The rates stated above are offered by Celcom Malaysia Sdn. Bhd.

Table 1: Comparison between SMS (GSM) and GPRS

SMS services has low latency because of it is based on circuit switch network unlike GPRS which is based on packets switched networks. The packets switched networks latency depends on how many users served by one node of the GPRS system. The effect of latency is observed whenever data is transmitted it takes unpredicted times to reach the destination. The reliability of GPRS is lower than SMS system. This can be improved by specifying a procedure to be implemented by our hardware to make sure all the data transmitted can reach the system server on time and with no loss.

The main attraction for GPRS is the cost factor. The operational cost for GPRS is lower compared to SMS. One SMS can only support 140 bytes of data and each

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Centered, Indent: First line: 0"

Formatted: Tab stops: 0.85", Left

Formatted: Indent: First line: 0"

Formatted: Centered

Formatted: Centered, Indent: First line: 0" message would cost 5 cent. Since the number of street lights in our monitoring system could reach thousands, the data streams are very huge and GPRS services will have significantly low operating cost. It is why GPRS is more preferable than SMS. The hardware cost associated with SMS implementation is high. The price of one unit GSM modem is approximately RM 380. Compared to GPRS, no specific hardware is required. A general desktop with internet connectivity is required to access the data.

To overcome the problem of weak signals, the transmission unit will be located at suitable location. A site survey will be conducted and the location of strongest signal strength is the perfect spot to place the transmission unit.

4.1

4.1.2 System DescriptionStreet light monitoring system using GPRS network

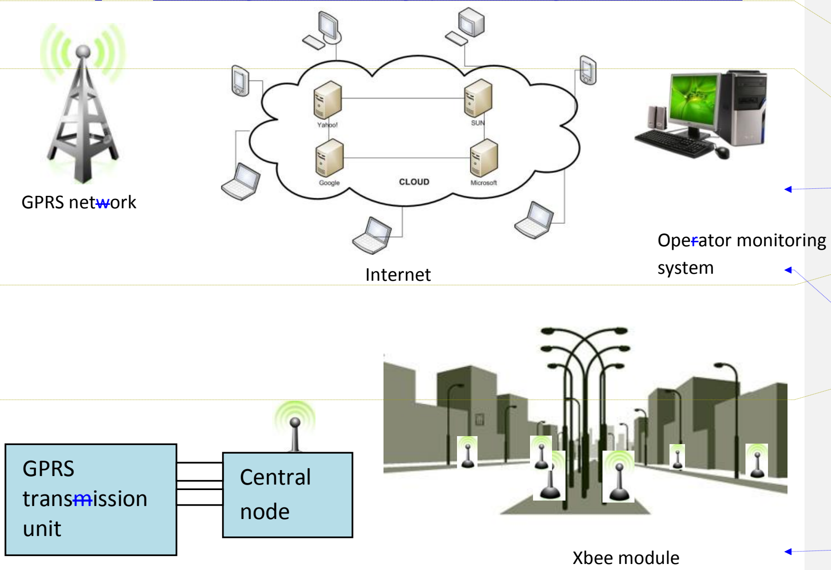


Figure 2: Integration of Zigbee and GPRS

Previously we have decided to use the GPRS services. Data is sent to the operator monitoring system across the internet connection. There are few ways the operator can accesses the data using the internet.

Formatted: Font: Bold, No underline, Font color: Auto

Formatted: Indent: First line: 0"

Formatted: Font: Bold

Formatted: Font: (Default) Times New Roman, 12 pt, Bold, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Line spacing: 1.5 lines

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified, Tab stops: Not at 1.14"

Formatted: Justified, Line spacing: Multiple 1.15 li, Tab stops: Not at 0.49"

Using GPRS network, there are several options on how to store the data in a server.

The first method will ~~is by~~ accessing~~ing~~ the database using a web browser. To achieve this, the monitoring system needs one dedicated ~~file~~ server so that all the data feeds from GPRS transmission unit can be stored. The server will store all the incoming data and when requested by ~~the~~ operators, the server will display all the data to the operators. The street light operator company needs to register with internet service provider and pay certain amount of fees to host web-based database server. -The server will have fixed IP address and visible by all computers connected to the internet.

~~Operator can have mobility as he can check the status of street lamp in the system as long as internet connectivity is available.~~

Another option is the street light operator company will host the web server. To achieve this, the company need to subscribe an internet connection and setup a small network for a database server. If the subscription does not provide fixed IP address, the operator can use the service from no-ip.com which offers web hosting services.

~~In order to achieve this, the company need to invest in establishing a file server. This file server will be given fixed IP address by the internet provider. During transmission, the packets from the field will be directed to this server. In this case, all the information is kept in the server.~~

4.2 Interfacing GPRS modem

Formatted: Justified

Formatted: Font: Bold, No underline, Font color: Auto

Formatted: Font: Bold

GPRS modem can be connected to a microcontroller to provide the internet connectivity. Suitable modem designed for Machine-to-Machine(M2M) application can provide this connectivity. This feature is available because the GPRS modem itself is embedded with TCP/IP stack. With the provided stack, communication via GPRS network can be achieved using AT commands. One of modem capable of delivering the connectivity is Q24 GSM/GPRS Modem from Mobitek.

Communication between the modem and the microcontroller can be realised using the UART interface in the microcontroller. The serial communication can be configured from 300 baud to 921600 baud. Both microcontroller and modem will need to be configured to this common baud rate. One point to note, the serial communication to the modem is using RS 232 standard. It is different from analogue input/output of microcontroller that use different the voltage level. For microcontroller the voltage used is 0V to 5V maximum. However, in RS 232 binary 0 is represented by voltage between 3V to 15V and binary 1 is represented by -15V to 3V.

To enable communication between modem and microcontroller, a MAX 232 IC will be used. The IC will convert 0V to 5V into +10V and -10V to represent two binary values used in RS 232 communication. To achieve this, the IC needs to be supplied with 5V voltage source. The datasheet for MAX 232 is attached as Appendix II. The connection from microcontroller to standard RS 232 connector is shown in figure 18. The pin configuration for RS 232 in given in the appendix V.

Formatted: Justified

Formatted: No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: No underline, Font color: Auto

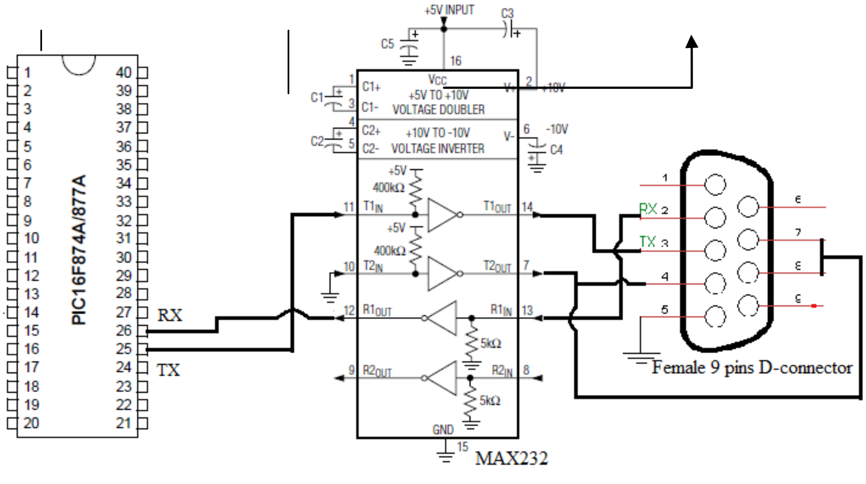
Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: Font color: Auto



CAPACITANCE (µF)					
DEVICE	C1	C2	C3	C4	C5
MAX232	1.0	1.0	1.0	1.0	1.0

Another option to manage the data is to direct all the data to one operator computer system. In this case, the operator computer system has become of a server. The data can only be accessed using the operator computer. One advantage of this configuration is increased in security level.

The design of the Graphical User Interface for this system will be discussed in the future. The main objective of this project is to be able to make sure successful transmission of data from the field to remote systems where the operator can analyze the data.

In the initial work to setup the web server, a fixed IP address is required. This not a problem because the fixed IP address be requested from internet service provider. For the purpose of demonstration, our FTP server will have dynamic IP addressed assigned by network provider. This IP address will change after few hours or days depending on the service provider. Before the transmission, the IP address will be determined by visiting website whatismyipaddress.com. The result displayed is the IP address for the FTP. If the FTP is connected to internet via a proxy server, the address returned is not the address of FTP server. Instead it is the value of the

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto gateway or the router itself. To solve this problem, further studies might give another option.

4.2 Alternative to GSM/GPRS application board

Formatted: Line spacing: 1.5 lines, Tab stops: 0.49", Left

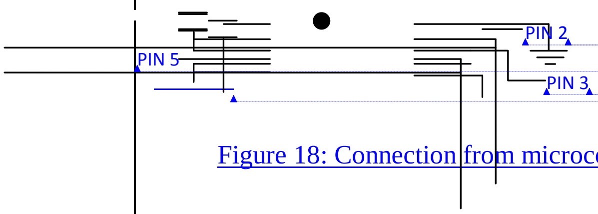


Figure 18: Connection from microcontroller pin to standard RS 232 9-pin

Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: Font: 9 pt
Formatted: English (U.S.)
Formatted: Centered

In RS 232 signals, logic 0 is represented by 3V to 15V and logic 1 is represented by -3V to -15V. Signals from the microcontroller will range from 0V to 5V. This signal is referred as TTL signals. MAX 232 has two RS 232 input pins and two TTL input pins. Referring to figure 18, a Since the suitable GSM/GPRS application board is not available, a simple solution to this problem is to develop a microcontroller based system that will have connection to internet. The system will be build with the reference from Application Note A731 from Microchip. The system will required additional component different from the detail described in the note.

Formatted: Justified

The reference provided in the note is to have a vending machine as an internet client. In that system the microcontroller will keep track of price for each item, number of item left, amount of deposited money and amount of money available for change. It will have a preset interval when the vending machine will establish connection to the internet and connect to server before uploading information. Connection established is a full duplex. Thus the client can receive instruction from server at the same time. The block diagram of system is shown in figure 3.

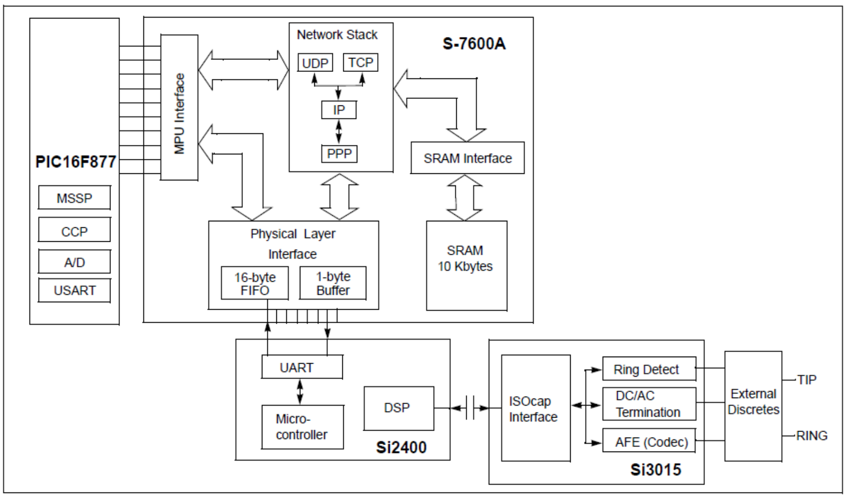


Figure 3 : Block diagram of reference system.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

4.2.1 Modification

The given reference system will read input from vending machine. To adapt this concept to Street Light Monitoring System, the PIC 16F877 microcontroller will be configured to receive input from Xbee module. This module will send the information to the microcontroller using serial communication function available in the microcontroller. Street light monitoring system will used the same S-700A chip to offload the task of establishing internet connection from microcontroller. Since GSM/GPRS network is decided to be communication medium between the transmission unit and operator monitoring system, GSM/GPRS modem will replace the telephone line modem. Telephone line modem is circuit with SI2400 and SI3015 chip.

As the conclusion, some modifications are required to incorporate the reference system into Street Light Monitoring system using GSM/GPRS network. The block that require modifications are shown in red triangle in figure 3.

4.2.2 Description of major part

i. Microcontroller

Microcontroller to be used is PIC 16F877 is the centre of operation for transmission unit. It will control the whole operation of transmission unit. However the task connecting and sending data to the internet is offloaded to separate chip. This is necessary so that microcontroller can focus more on the application.

One of the applications is retrieving the data from centre node. Centre node will receive stream of data from a groups of Xbee transmitter in that area. To cater for this, microcontroller will retrieve the information when data is available on the centre node. Data retrieval is base on interrupt routine. Centre node will signal an interrupt vector to microcontroller before microcontroller begins to retrieve the data serially. A raw source code to retrieve and retransmit data is provided below. This code requires modification before can be integrated in the full system.

Sample code

```
#include <16F877.h> //register definition file for
#fuses HS, NOWDT            //pic 168F77
#use delay(clock=10000000)
#use rs232(baud=9600,parity=N,xmit=PIN_C6,rcv=PIN_C7,bits=8) //initialize usart

#define RX_BUFFER_SIZE 24
char Rx_Buffer[RX_BUFFER_SIZE+1]; //character array (buffer)
char Rx_Wr_Index =0;                    //index of next char to put into buffer
char Rx_Rd_Index= 0;                    //index of next char to fetch from buffer
char Rx_Counter=0;                            //total count of char in buffer
int1 Rx_Buffer_Overflow =0;         //flag set on Uart

//UART Transmit Buffer
#define TX_BUFFER_SIZE 24
char Tx_Buffer [TX_BUFFER_SIZE+1];   //char array (buffer)
char Tx_Rd_Index =0;                //index of next char to put into buffer
char Tx_Wr_Index=0;                 //index of next char to fetch from char in buffer
char Tx_Counter=0;                          //total count of character in buffer


//usart receiver interrupt service rouine
#int_rda              /*preprocessor directive identifying the following
                      routine as interrupt: rs232 receive data available  */
```


Formatted: Justified, Space After: 10 pt, Line spacing: 1.5 lines

```
void serial_rx_isr ()                               //receive interrupt routine
{
        //put char in buffer
        Rx_Buffer[Rx_Wr_Index]= getc();                  //get char from I/O,PIN_C7
        if(++Rx_Wr_Index>RX_BUFFER_SIZE)Rx_Wr_Index =0; //reset the pointer to 0
        if(++Rx_counter > RX_BUFFER_SIZE)               //check overflow
        {
Rx_Counter = RX_BUFFER_SIZE;         /*in overflow receive counter equal buffer size*/
                Rx_Buffer_Overflow =1;                  //indicate overflow
        }
}

//We want to get character from UART receiver buffer
char bgetc (void)
{
        char c;
        while (Rx_Counter == 0);                    /*when no character in
buffer program will wait for
                                                    character;if no character program stuck here*/

c= Rx_Buffer[Rx_Rd_Index];      /* there is data in receive buffer,thus get the data and set to variabler c*/
        //read index greater than buffer,reset it to 0
        if(++Rx_Rd_Index > RX_BUFFER_SIZE)Rx_Rd_Index =0;
```

```
//as long as Rx_counter not zero,reduce by one because one already read
        if (Rx_Counter)Rx_Counter--;
        return c;
}

//Usart transmitter interrupt service routine
#int_tbe        //prepocessor directive indentifying following subroutine as interrupt
                        //trigger when RS232 transmit buffer empty
void serial_tx_isr()
{
        //if there are char to be transmitted
        if (Tx_Counter !=0)             //as long as there is data in transmit buffer....
        {
                //send the character out the port
                //put char to default usart PIN_C6 as define above
                putc(Tx_Buffer[Tx_Rd_Index]);
                //test and wrap pointer
                if(++Tx_Rd_Index > Tx_Buffer_size)Tx_Rd_index=0;

                Tx_Counter--;           //one char is sent out so char not is left
                //if no more char,disable the interrupt
                if(Tx_Counter == 0)disable_interrupts(int_tbe);
                //disable_interrupt provided by the compiler;not user define

        }
```

```
}
// writing a character to serial transmit buffer
void bputc (char c)                  //define function of bptc which expect a char as input
{
        char restart = 0;
        //just wait here until tx buffer is not full
        while(Tx_Counter>(TX_BUFFER_SIZE-1));
        if(Tx_Counter==0)restart=1; //if buffer is empty, setup for interrupt
        Tx_Buffer[Tx_wr_Index++]=c; //jam the char into tx buffer

        if(Tx_Wr_Index>TX_BUFFER_SIZE)Tx_Wr_Index=0;          //wrap the pointer

        Tx_Counter++;    // since char is added to tx buffer, sount of char also increase

        //if there were not char, but have now
        //re-enable the interrupt
        if(restart==1)enable_interrupts(int_tbe);
}
void main(void)
{
        unsigned char k;
        enable_interrupts(global); //enable interrupt in global
        enable_interrupts(int_rda); //enable the receive interrupt

        //print message that we are up and running
```

```
//sending char to function bputc which actually sent data using USART
	printf(bputc,"\r\n\Running...\r\n");
	while(1)
	{
		//buffer up 5 characters before transmitting them
		if(Rx_Counter>4)
		{
			// send the character untill there are none left
			while(Rx_Counter>0)
			{
				//get the char;actually calling the function which return char
				k=bgetc();
bputc(k);			// and echo it back
			}
		}
		//since there is no waiting on getchar ot putchar,we can do other things
	}
}
```

Formatted: Justified

ii. TCP/IP

The embedded system will use dial up connection. Thus the network access layer is point-to-point(PPP). This protocol is used when data transmission using serial link as modem. PPP has the function to IP and TCP/UDP packet with header, configure link between two peer and configure network connection. PPP also handles the username and password authentication. Basically PPP support two protocol which are PAP(Password Authentication Protocol) and CHAP (Challenge

Handshake Authentication Protocol). Once data link parameters are negotiated, PPP is used to transfer data packets between two hosts and terminate connection.

iii. Transport Layer

Transport layer in this system will provide basic communication between a server and a client. One type of transport protocol is User Datagram Protocol(UDP). In this protocol, a host will receive a packet from another host but it will not send an acknowledge frame. UDP is proven to work best when transmitting small packets. Compare with UDP, Transmission Control Protocol (TCP) is more reliable because it implements handshaking. Even though it is more reliable, more work is required to configure the handshaking. To choose which work best, the importance of data to be transmitted must be taken into consideration. In our case, the status of street light is the information. The status is not very crucial in term of, operator monitoring system can still get the status during next data transmission. Thus UDP is more feasible in the Street Light Monitoring System.

iv. TCP/IP stack device

This device takes over the task of implementing network access layer, internet layer and transport layer. The device proposed is manufactured by Seiko Instruments. The datasheet of the device is available. One single chip contain 10kb of RAM, microcontroller interface, TCP/IP stack engines and UART for serial data transmission.

When packets are received, TCP/IP engines determine the IP and port number. Then it will calculate the checksums and then transfer the data to a packet buffer. It will use the interrupt line to indicate there is data available for microcontroller.

The chip will act as data buffer. It stores data to be transmitted up to 1024 bytes in its internal RAM buffer. This chip will append various header and checksums to data packets. Then it will transmit these packets from UART.

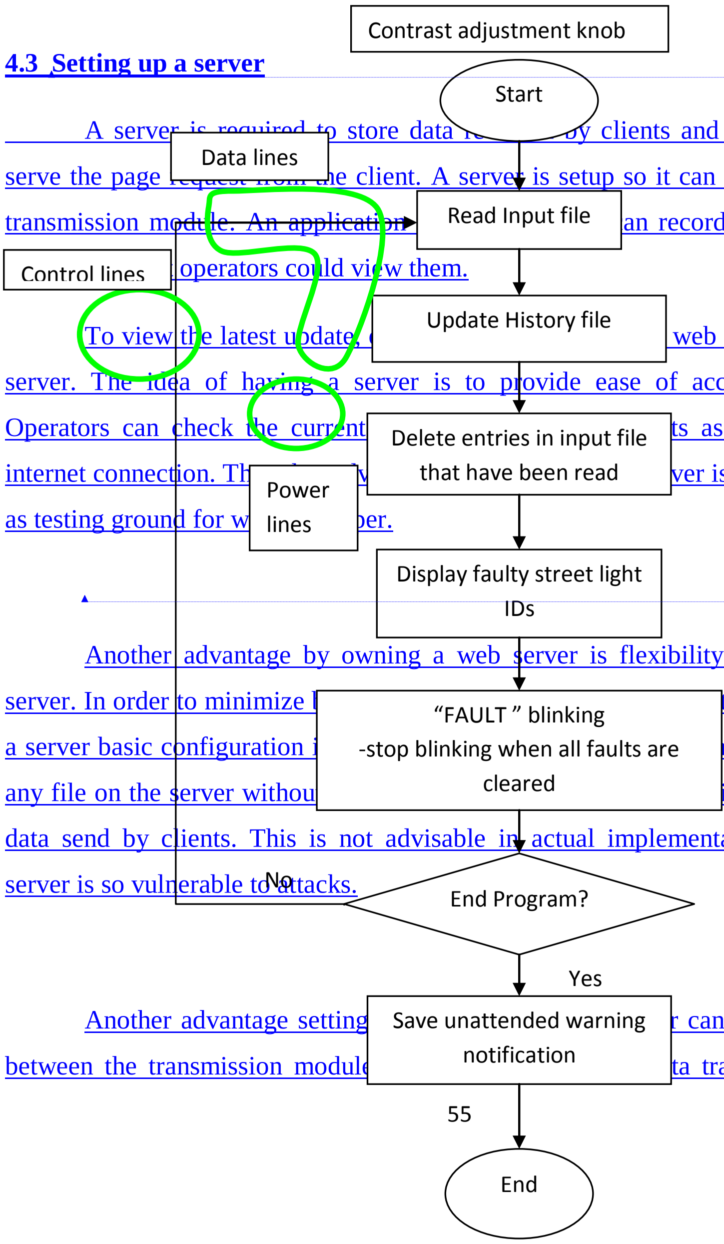
Formatted: Justified, Line spacing: 1.5 lines microcontroller will send data serially using pin 25. Pin 25 is connected to TTL input of MAX 232. RS 232 signals are generated at pin 14. This pin is connected to pin 3 of 9 pin D-connector. If modem is sending data, pin 2 of 9 pin D-connector will be used. The signals are sent to pin 13 on MAX 232 and the output is at pin 12. The outputs at pin 12 are TTL signals thus data transmitted can be captured by pin 26 of microcontroller.

4.3 Setting up a server

A server is required to store data re... by clients and at the same time serve the page request from the client. A server is setup so it can receive data from transmission module. An application ... an record the errors into a ... operators could view them.

To view the latest update, ... web page stored in the server. The idea of having a server is to provide ease of access to operators. Operators can check the current ... ts as long as there is internet connection. Th... ver is because it serves as testing ground for w... er.

Another advantage by owning a web server is flexibility to configure the server. In order to minimize ... tween a client and a server basic configuration i... client can access any file on the server withou... it will receive any data send by clients. This is not advisable in actual implementation because the server is so vulnerable to attacks.



Another advantage setting ... r can see any progress between the transmission module ... ta transmission failed,

Formatted: Line spacing: single

Formatted: Centered, Line spacing: single

Formatted: Font: Not Italic, No underline, Font color: Auto

Formatted: Indent: First line: 0.5"

Formatted: Font: Not Bold

troubleshooting is easier as the owner can trace the problems along the data transmission as he has 100% control over the system. With the information, modification to the program in transmission module or the application on the server could be done to improve efficiency and reliability of street light monitoring system.

4.3.1 Server accessibility

There are many network configurations as designed and being implemented by network administrator. From the IP address assigned to a node (a computer) connected to the internet or any network, it gives some information on the network architecture. There are 5 type of IP address which is A, B, C, D and E. Class D and E are more for research while class A, B and C are widely use. The address range of address is shown table 2. Class A, B and C IP addresses are called Private IP address.

Table 2: IP address class address range

Class	**Private Start Address**	**Private Finish Address**
A	10.0.0.0	10.255.255.255
B	128.0.0.0	191.255.255.255
C	192.0.0.0	223.255.255.255

If one is trying to setup a server and its server has the IP address in class A, B or C, other computers outside the private network could not reach the server because incoming connections are being block either by firewalls, router or proxy which use NAT (Network Address Translation). To make the server is accessible from outside private network; ones could have the network administrator to allow incoming connection to reach the server which resides in private network. This is called port forwarding. Other option is to connect the server to a network which assigned public IP address.

Formatted: Font: Bold, No underline, Font color: Auto
Formatted: Font: Bold
Formatted: Indent: First line: 0.5"
Formatted: Font: Not Bold, No underline, Font color: Auto
Formatted: No underline, Font color: Auto
Formatted: Font: Not Bold, Font color: Auto
Formatted: Font: Bold, No underline, Font color: Auto
Formatted: Centered, Space After: 0 pt
Formatted Table
Formatted: Font: Bold
Formatted: Font: Bold, No underline, Font color: Auto
Formatted: Font: Bold
Formatted: Font: Bold, No underline, Font color: Auto
Formatted: Centered, Space After: 0 pt
Formatted: Centered, Space After: 0 pt
Formatted: Centered, Space After: 0 pt
Formatted: Indent: First line: 0.5"

Because of Q24 GSM/GPRS Modem has limited functionality (which will be discussed in next section), the server for receiving updates from transmission module is connected to a private network by using GPRS enabled SIM card from Celcom. Transmission module is also connected to the same private network. Thus the module can connect and transmit data to server. Figure 19 illustrated the network connection between the server and transmission module.

One advantage of this configuration is data transmission from transmission module to server will incur no cost at all. Usually GPRS service is charged for the amount of data received from outside the private network which pass through Celcom proxy server. The disadvantage is the updates of street lights status cannot be viewed from a web browser with internet connectivity. If at the same time the server is attached to another network and the network configuration allows the server being access from internet, then the updates can be viewed using web browser. This is only possible when a server is allowed to have two IP address.

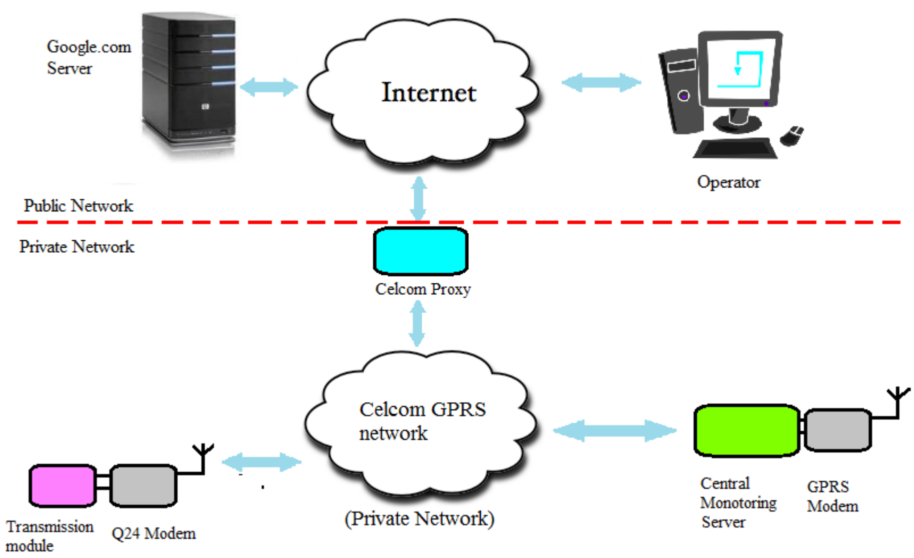


Figure 19: Transmission module and server network connection

From figure 19, operator computer who is connected to public network (internet) cannot access central monitoring server because the reside in private network. Celcom proxy server will block operator's desktop from accessing the server. However transmission module can access the server easily because both reside in same private network.

4.4 Connecting to server and sending data

From previous section, the author has clearly stated that with Q24 GPRS modem Street Light Monitoring System with internet connectivity is not feasible given limited amount of time. The transmission module via the modem can setup up socket connection to a server outside the Celcom private network. Current version modem firmware does not support socket connection through server.

GPRS enabled mobile phone and smart phone can accessed internet using GPRS services because the phones have software version of TCP/IP stack incorporated into their phone firmware. Microcontroller can be connected to internet provided ones must write the TCP/IP stack program from scratches and load them into microcontroller. Since time is limited and the author is not familiar writing his own stack program, the objective of providing internet accessibility to the status page cannot be achieved.

4.5 Writing socket application

Socket connection can be view as single communication thread between a server and a client. This may give inaccurate view of how socket connection handles data to be delivered to server. Ones may think as data being written to socket application, immediately the server will receive the data.

Actually, the socket control will hold the data first. The reason is socket connection is still based on packet. Each packet will contain one segment of data as well as few IP headers so that the data can reach the specified destination safely. According to WIPSoft V2.01, data sent to TCP socket will be buffered until one of these 3 conditions occurs:

i. Buffered data is twice or more than preferred segmentation size
ii. Internal time expires
iii. Sockets is shutting down or closing

After conducting few trials to send data to listening socket application on the server, data is received in segments. This segmentation issue raise a challenge to capture the incoming data. The method to solve the problem is by capturing each character received and store it into a temporary array. Upon receiving terminating character which is chosen to be character x, the socket application know one complete data is received. The data is then displayed on screen and inserted into database.

4.6 Establishing GPRS connection and sending data using prototype transmission module

Transmission module is programmed to start negotiating socket connection when any switch changes state. The figure 18 shows initial states of switches.

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: i, ii, iii, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.75"

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: Font: Bold, Not Italic, No underline, Font color: Auto

Formatted: Font: Bold, Not Italic, No underline, Font color: Auto

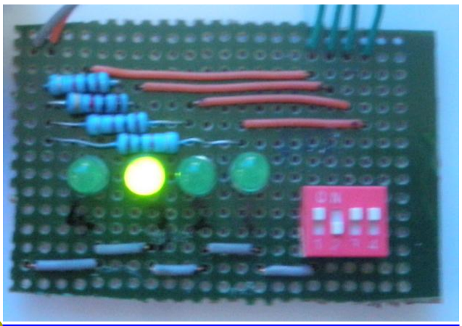


Figure 20: Initial switch conditions

Figure 21 shows the socket application is already listening for incoming connection request.

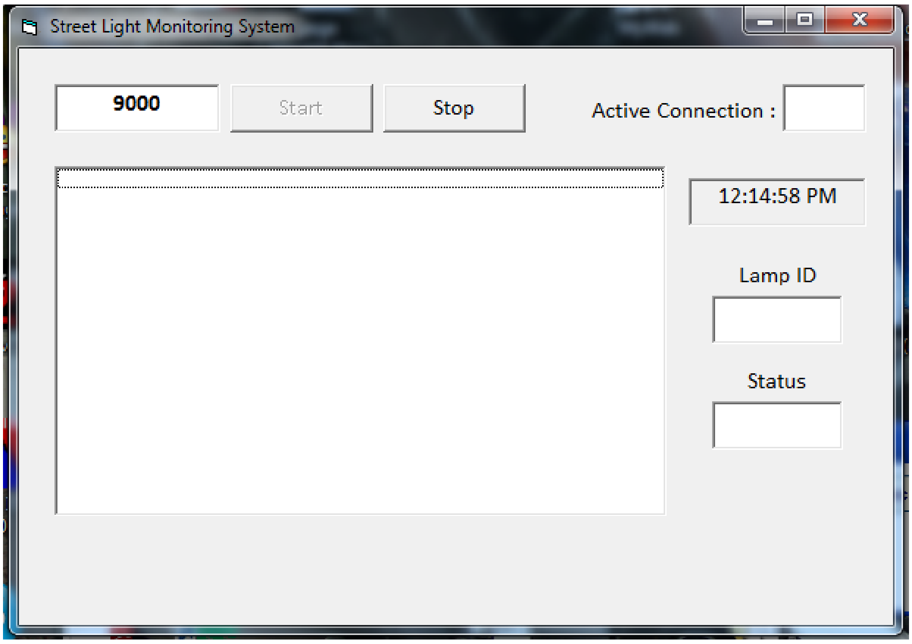


Figure 21: Socket application listening for incoming connection

Figure 22 shows the web page intended to show current status to operator. The server has not received any input so the page is blank.

Formatted: Font: (Default) Times New Roman, 14 pt, Bold, No underline, Font color: Auto

Formatted: Font: 12 pt, Not Bold

Formatted: No underline, Font color: Auto

Formatted: Font: 12 pt, Not Bold, No underline, Font color: Auto

Formatted: Font: 12 pt, Not Bold

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Line spacing: 1.5 lines

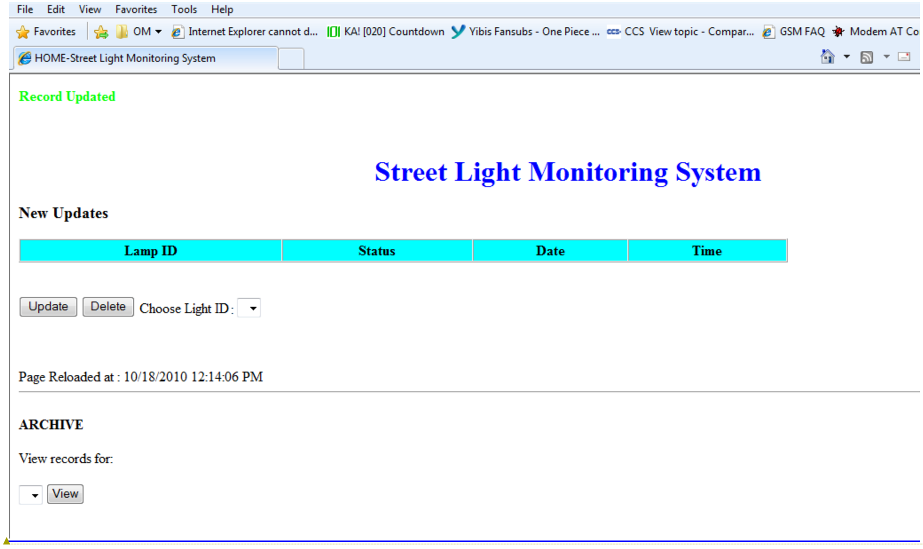


Figure 22: Main page of Street light monitoring system

When the first switch changes its state from low to high the microcontroller senses the changes on its pins. Based on the signals, microcontroller and modem start to negotiate connection and read each switch. Since each pin is already assigned an ID number in the program, the ID numbers and the conditions are sent to server. The data received by the server is displayed on screen. Figure 22, shows the information received by server.

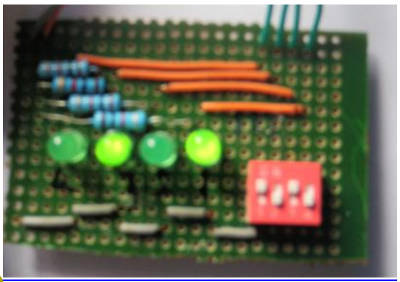


Figure 23: First switch change from low to high

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified, Line spacing: 1.5 lines

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: 0.5", Left + Not at 1.14"

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

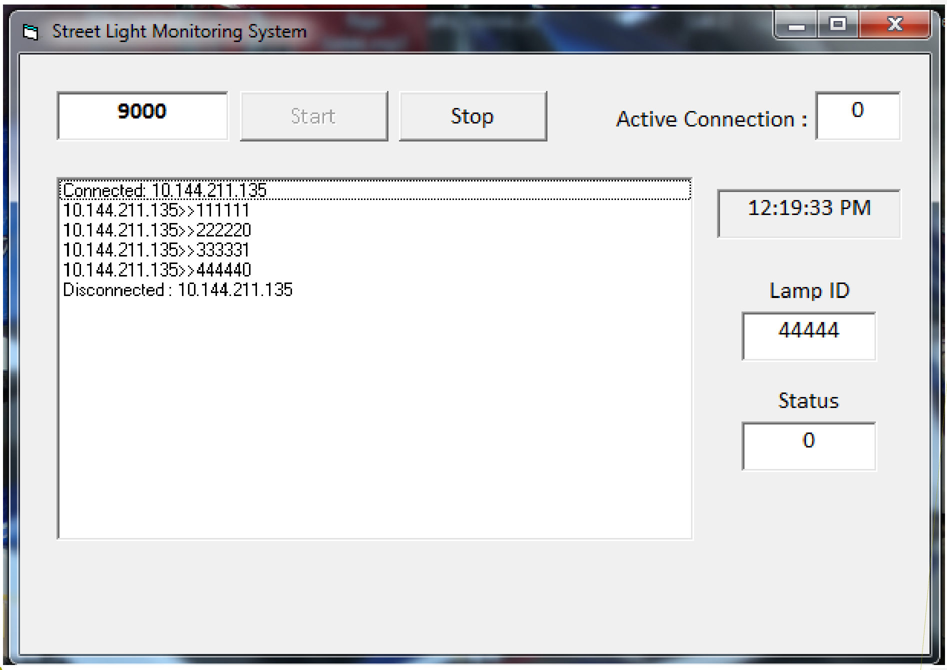


Figure 24: Information received displayed on socket application

The data received is used to update the database. Operators who wish to view the status can browse the page using the same IP address the transmission module used. The main page is shown in figure 25.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: 0.5", Left + Not at 1.14"

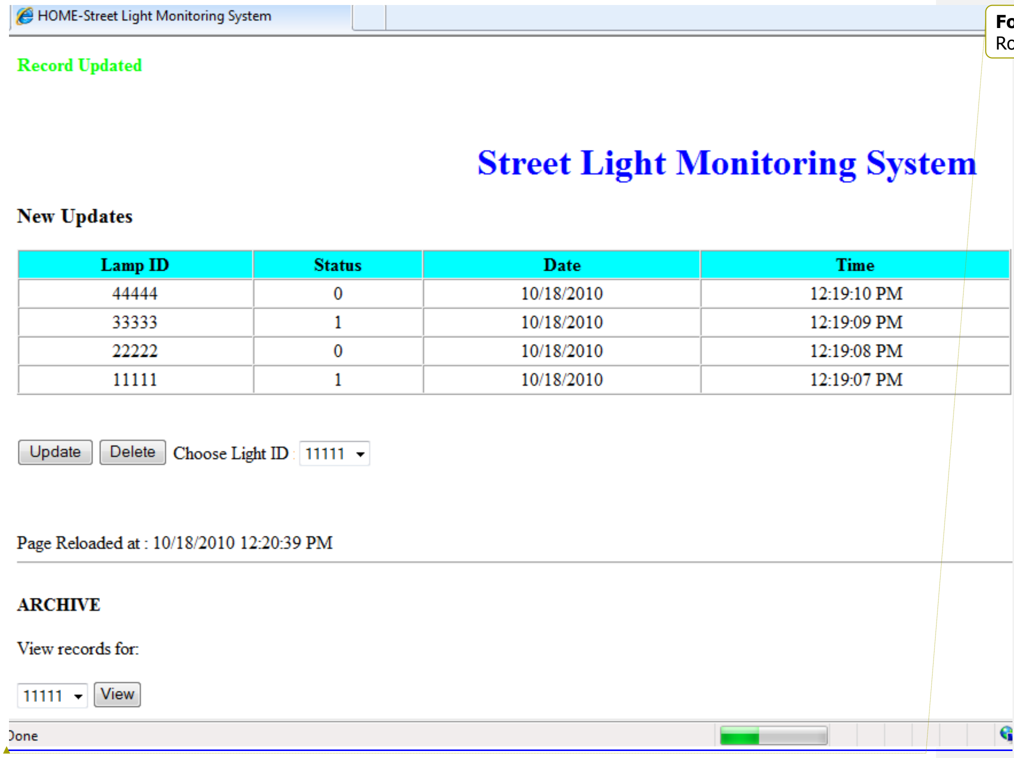


Figure 25: Main page after the server received the updates

4.6.1 Discussion

This test is meant to test whether the transmission module can pass data to a server and the server can be accessed to view the updates. Up to this stage, the transmission module and the server manage to be integrated.

To implement street light monitoring system the transmission module is required to prepare a link to a server for receiving input module. The Receiving Input module is still under progress. From design, the receiving module will receive updates from transmitter and put the data into buffers. If updates are sent to receiving module using Xbee module, the baud rate is 9600 while modem and transmission module communicate at baud rate of 115200. The receiving unit will receive the updates at 9600 and retransmit to server at higher baud rate 115200.

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Font: Bold, No underline, Font color: Auto

Formatted: Font: Bold

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Some observations were made and briefly described in following three sections.

a. Data was transmitted at low cost.

During the test, about thirty data sets were sent to the central monitoring server. These thirty data sets were equal to 180 bytes. 180 bytes were in form of data and if included with the IP header the size were larger. Before the test was carried out, airtime credits of the SIM card was recorded. After completing the test, the balance of airtime credits was checked. There was no difference between the value of the credits before and after the test.

It can be concluded that data was sent at no cost at all. This situation could be explained with the understanding how the GSM structure of the network provider. Usually data usage is charged by the network provider when data goes through provider proxy server. In this test both transmission unit and server were setup in private network own by the service provider(Celcom). Since no data passed through the proxy server, no charged was incurred.

b. Time taken to connect to the server was unpredictable.

Before the transmission and monitoring unit be able to start transmitting data, a communication link between the server and the transmission unit was necessary. The transmission module started to initiate connection as it received signals from monitoring module. From observation it was noticeable that time taken to establish connection varied each time the transmission unit tried to send data.

The variations in time to establish connection was due to the limitation of the GSM network structure. In GSM network, one Base Station Subsytem (BSS), could only serve limited number of GPRS request. In time of high network traffic, some requests may take awhile before being granted access to use the GPRS services. Sometimes requests expire and the transmission unit needs to send another request.

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5", Tab stops: Not at 1.14"

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: List Paragraph, Justified, Indent: First line: 0.5", Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5", Tab stops: Not at 1.14"

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: List Paragraph, Justified, Indent: First line: 0.5", Line spacing: 1.5 lines, Tab stops: Not at 1.14"

c. Transmission and Monitoring units halt.

Both transmission and monitoring units issued commands and waited for the responses from the GSM/GPRS modem. Responses from the modem were required before next instructions were issued. If instructions were issued at incorrect moment or wrong instructions were issued, the modem would not response. As the consequences, the whole system will halt. It could not receive or send any data.

This system flaws were the result of insufficient features of the transmission and monitoring unit. The unit need more features to deal with this problem. New algorithm could be embedded into the system to solve the problems. After certain period of times expired, the unit can re-initiated the connection again. In the case of serious errors when the modem refused to response, extra circuit and new algorithm could turn the modem off and on again. This could make the modem responses again.

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.25" + Indent at: 0.5", Tab stops: Not at 1.14"

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Tab stops: Not at 1.14"

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: Line spacing: 1.5 lines

CHAPTER 5

~~5.1~~ CONCLUSION AND RECOMENDATION

5.1 Conclusion

Street light monitoring system using GSM/GPRS network is aimed to improve the efficiency of existing street light monitoring. With the implementation of the system, human involvement in identifying faulty street lights may be reduced to minimal level which may enable more human involvement in the maintenance works. Street light is an important entity to the public as it plays an important role in the public safety and ensuring the safety of road users.

The proposed system improved the existing street light monitoring system by collecting conditions of street lights from transmitters located at each street light. The collected data is retransmitted to a central monitoring server for viewing by the operator. The updates can be viewed using the web browser thus providing easy access for the operator to view and monitor the street lights.

The system is composed of two separate sections. One section is focusing on receiving inputs from the local transmitter and re-transmitting the received data to the central monitoring server. Another section is for receiving data and managing that data in a server so that operators can assess the condition of the street lights. It is important to notice that the strategy and method used to enable all street lights in one area to transmit data to receiving input module is beyond the scope of this project.

Formatted: Centered

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Centered, Line spacing: 1.5 lines

Formatted: Justified, Line spacing: 1.5 lines, Tab stops: 0.49", Left

This project is more concerned about the transmission of data using GSM/GPRS network to the server.

The prototype managed to send data to the server. However the prototype is not capable to provide reliable solution to monitor street lights. The problems with the prototype are inconsistency to connect to the server and the prototype tends to stop responding when it has problems with the GSM/GPRS modem. In order to improve the prototype, more features should be added to the prototype. The future improvements on the prototype are described in recommendations.

5.2 Recommendations

Before the prototype could achieve the objective of providing a reliable street light monitoring system, the prototype requires some modifications. The modifications are necessary to overcome with the problem discussed in previous chapter.

Following are the features to be embedded into the prototype to produce a more reliable system.

a. Interrupts based on timer to bring out the system from system halt.

   Every time the prototype issues a command and waits for the response from the modem, the timer will start to ticks. The modem should response before the timer expires. If the modem fails to response, the prototype will issue command to solve the problem.

b. Additional circuit and algorithm to release the modem from errors.

   In

   the event of serious errors, simply restarting the modem could bring back the modem to work properly. To achieve this additional circuits are required to actually turn the modem off and turn it on again. Additional algorithm is also required to detect the condition that required modem reset.

c. Write an efficient and better code for socket application.

   On the server side, one socket application is waiting for incoming connection from the prototype. The application is very crucial as it is the one which will acknowledge or refuse an incoming connection. From the test, the application still has unsolved bugs. Other problem with the socket application is it takes time to update and response to the prototype.

Formatted: Font: Bold, No underline, Font color: Auto

Formatted: Justified, Line spacing: 1.5 lines

Formatted: No underline, Font color: Auto

Formatted: Justified, Indent: First line: 0.5", Line spacing: 1.5 lines

Formatted: List Paragraph, Justified, Space Before: 12 pt, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.5" + Indent at: 0.75"

Formatted: Font: (Default) Times New Roman, 12 pt

Formatted: List Paragraph, Justified, Indent: Left: 0.75", Line spacing: 1.5 lines

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.5" + Indent at: 0.75"

Formatted: List Paragraph, Justified, Indent: Left: 0.75", First line: 0.25", Line spacing: 1.5 lines

Formatted: List Paragraph, Justified, Indent: Left: 0.75", Line spacing: 1.5 lines

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.5" + Indent at: 0.75"

Formatted: List Paragraph, Justified, Indent: Left: 0.75", First line: 0.25", Line spacing: 1.5 lines

Better socket application code will reduce time spend on transmitting data.

d. Send data in blocks to reduce number of negotiations with the server.

Waiting for responses from the server causes the prototype to be less efficient. More time waiting means less time is spend on sending data. The efficiency could be improved by sending data in blocks. Whenever possible, arrange the data into blocks and send many set of updates in one session. However, ones need to consider of providing real times updates to central monitoring station. Thus, before adding this feature ones need to consider trade off between efficiency and real time monitoring requirement.

Formatted: List Paragraph, Justified, Indent: Left: 0.75", Line spacing: 1.5 lines

Formatted: List Paragraph, Justified, Line spacing: 1.5 lines, Numbered + Level: 1 + Numbering Style: a, b, c, … + Start at: 1 + Alignment: Left + Aligned at: 0.5" + Indent at: 0.75"

Formatted: List Paragraph, Justified, Indent: Left: 0.75", First line: 0.25", Line spacing: 1.5 lines

Formatted: Font: Not Bold

Formatted: Font: Bold, No underline, Font color: Auto

Formatted: Font: Bold

~~To summarize this report, the progress of this project focuses more on the transmission of data from the field. It is already decided to use GPRS because it is economically feasible. However GPRS implementation requires more knowledge on networking as FTP server need to be setup before any transmission. Setup the server before transmission is a challenging task. Enhancing the system with good Graphical User Interface might be possible when the task complete within the specified timeline.~~

~~Some hardware will be developed to automate the process of retrieving the data from the wireless network. The data retrieved will then be stored in the FTP server at remote location using the internet. Some programming is required to program the GSM/GPRS application board, the microcontroller and the Xbee module itself. All the components need to be integrated properly to make sure it will function as it is intended.~~

~~More progress is expected after the submission of the report. The future research will be more the communication because it is the main objective of this project. The system should receive raw data from field, transmit to the operator system and view in GUI.~~

Formatted: Centered

Formatted: Centered, Tab stops: 0.49", Left + Not at 1.14"

Formatted: Centered, Line spacing: 1.5 lines, Tab stops: 0.49", Left + Not at 1.14"

Formatted: Centered

Formatted: Centered, Tab stops: 1.04", Left

REFERENCESREFERENCES

1. R.G. Prabhudesai, Antony Joseph, Yogesh Agarvadekar, Nitin Dabholkar, Prakash Mehra, Albert.Gouveia, Suryakant Tengali, Vijaykumar, and Abijith Parab, "GPRS based Real-Time Reporting and Internet Accessible Sea Level Gauge for Monitoring Storm Surge and Tsunami", National Institute of Oceanography, Dona Paula, Goa- 403 004, India

2. Cao Jian, Qian Suxiang, Hu Hongsheng, Yan Gongbiao," Wireless Monitoring and Assessment System of Water Quality Based on GPRS", Mechanical & Electrical Engineering, Jiaxing University, Jiaxing

3. Timo Ruohonen, Leena Ukkonen, Mikael Soini, Lauri Sydänheimo, Markku Kivikoski, "Quality and Reliability of GPRS Connections", Tampere University of Technology, Finland

4. Guang Shao, Jialin Hou, Bing Zou and Zhen Wang," Research and Development of Non-magnetic Heat meters Based on ZigBee and GPRS", The College of Mechanical and Electronic Engineering, Shandong Agricultural University Taiwan 271018, China

5. Geoff Sanders, Lionel Theorens, Manfred Reisky, Oliver Rulik, Stefan Deylitz, GPRS NETWORKS, WILEY

6. Embedding PICmicro® Microcontrollers in the Internet, *Rodger Richey/Steve Humberd*
Microchip Technology Inc.Chandler, AZ

[1] M. Ramsay, R. Newton," The Effect Of Better Street Lighting,On Crime And Fear: A Review", Home Off., Queen Anne's Gate, London, Crime Prevention Unit paper 29,1991.

Formatted: Centered

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Font: 14 pt, No underline, Font color: Auto

Formatted: Font: 14 pt

Formatted: Justified, Indent: Left: 0", Hanging: 0.3", Line spacing: 1.5 lines, Tab stops: 0.3", Left + Not at 1.04"

[2] Izz. Abd. Aziz, M. H. Hassan, M. J. Ismail, M. Mehat, N. S. Haron," Remote Monitoring in Agricultural Greenhouse Using Wireless Sensor and Short Message System(SMS)", IJET, Vol 9, No 9,pp 1-12.

[3] Jian, Cao; Suxiang, Qian; Hongsheng, Hu; Gongbiao, Yan, "Wireless Monitoring and Assessment System of Water Quality Based on GPRS," Electronic Measurement and Instruments, 2007. ICEMI '07. 8th International Conference on , vol., no., pp.2-124-2-127, Aug. 16 2007-July 18 2007.

[4] Prabhudesai, R.G.; Joseph, A.; Agarvadekar, Y.; Dabholkar, N.; Mehra, P.; Gouveia, A.; Tengali, S.; Vijaykumar; Parab, A.; , "GPRS based Real-Time Reporting and Internet Accessible Sea Level Gauge for Monitoring Storm Surge and Tsunami," *OCEANS 2006 - Asia Pacific* , vol., no., pp.1-4, 16-19 May 2007.

[5] Lee, P.K.; Lai, L.L.; , "A Practical Approach to Wireless GPRS On-Line Power Quality Monitoring System," *Power Engineering Society General Meeting, 2007. IEEE* , vol., no., pp.1-7, 24-28 June 2007

[6] Ruohonen, T.; Ukkonen, L.; Soini, M.; Sydanheimo, L.; Kivikoski, M.; , "Quality and reliability of GPRS connections," *Consumer Communications and Networking Conference, 2004. CCNC 2004. First IEEE* , vol., no., pp. 268- 272, 5-8 Jan. 2004.

[7] Active Server Pages, http://msdn.microsoft.com/en-us/library/aa286483.aspx

Formatted: No underline, Font color: Auto

Formatted: Font: Not Italic, No underline, Font color: Auto

Formatted: No underline, Font color: Auto

Formatted: Indent: Left: 0", Hanging: 0.3"

Formatted: Font: Not Bold, No underline, Font color: Auto

Formatted: Font: Not Bold

Formatted: Justified, Line spacing: 1.5 lines

Formatted: Justified, Line spacing: 1.5 lines

APPENDICES

Formatted: Font: 14 pt, Bold, No underline, Font color: Auto

Formatted: Space After: 0 pt, Line spacing: 1.5 lines

APPENDIX I

MAX 232 DATASHEET

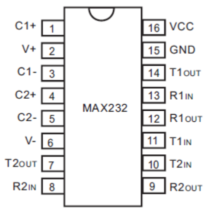
Single Supply Voltage, RS-232 Transceiver

DESCRIPTION

MAX232 is compatible with RS-232 standard, and consists of dual transceiver. Each receiver converts TIA/EIA-232-E levels into 5V TTL/CMOS levels. Each driver converts TT-L/COMS levels into TIA/EIA-232-E levels. The MAX232 is characterized for operation from -40°C to +85°C for all packages.

MAX232 is purposed for application in high-performance information processing systems and control devices of wide application.

PIN CONFIGURATION



(Top View)

FEATURES

- Input voltage levels are compatible with standard CMOS levels
- Output voltage levels are compatible with EIA/TIA-232-E levels
- Single Supply voltage: 5V
- Low input current: 0.1μA at T_A= 25 °C
- Output current: 24mA
- Latching current not less than 450mA at T_A= 25°C
- The transmitter outputs and receiver inputs are protected to ±15kV Air ESD

APPLICATION

- Battery-Powered RS232 Systems
- Terminals
- Modems
- Computers

ORDERING INFORMATION

Temperature Range	Package		Orderable Device	Package Qty
-40°C to +85°C	SOP16L	Pb-Free	MAX232D	50Units/Tube
			MAX232DR	3000Units/R&T
	DIP16L		MAX232N	25Units/Tube

Formatted: Font: 14 pt, Bold, No underline, Font color: Auto

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

PIN DESCRIPTION

No.	Name	Function
1	C1+	External capacitance of positive voltage multiplier unit
2	V+	Output of positive voltage of multiplier unit
3	C1-	External capacitance of positive voltage multiplier unit
4	C2+	External capacitance of negative voltage multiplier unit
5	C2-	External capacitance of negative voltage multiplier unit
6	V-	Output of negative voltage of multiplier unit
7	$T2_{OUT}$	Output of transmitter data (levels RS – 232)
8	$R2_{IN}$	Input of receiver data (levels RS – 232)
9	$R2_{OUT}$	Output of receiver data (levels TTL/CMOS)
10	$T2_{IN}$	Input of transmitter data (levels TTL/CMOS)
11	$T1_{IN}$	Input of transmitter data (levels TTL/CMOS)
12	$R1_{OUT}$	Output of receiver data (levels TTL/CMOS)
13	$R1_{IN}$	Input of receiver data (levels RS – 232)
14	$T1_{OUT}$	Output of transmitter data (levels RS – 232)
15	GND	Ground
16	V_{CC}	Supply voltage

ABSOLUTE MAXIMUM RATINGS

Parameter		Symbol	Min	Max	Unit
Supply voltage		V_{CC}	-0.3	6.0	V
Transmitter high output voltage		V+	V_{CC} -0.3	14	V
Transmitter low output voltage		V-	-0.3	-14	V
Transmitter input voltage		V_{TIN}	-0.3	(V+) + 0.3	V
Receiver input voltage		V_{RIN}	-30	30	V
Dissipated power	DIP package	P_D		842	mW
	SOP package			762	
Output current of transmitter short circuit		I_{SC}		Continuously	mA
Storage temperature ranges		T_{stg}	-60	150	°C

RECOMMENDED OPERATING CONDITIONS

Parameter	Symbol	Min	Max	Unit
Supply voltage	V_{CC}	4.5	5.5	V
Transmitter output high voltage	V+	5.0		V
Transmitter output low voltage	V-	-5.0		V
Transmitter input voltage	V_{TIN}	0	V_{CC}	V
Receiver input voltage	V_{RIN}	-30	30	V
Transmitter short circuit output current	I_{SC}		±60	mA

DC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Test Conditions		25°C		-40°C to +85°C		Unit
				Min	Max	Min	Max	
Supply Current	I_{CC}	V_{CC} =5.0V, V_{IL} = 0V			10.0		14.0	mA
Receiver								
Hysteresis voltage	V_h	V_{CC} =5.0V		0.2	0.9	0.2	1.0	V
On (operation) voltage	V_{on}	$V_O \le$ 0.1V, $I_{OL} \le$ 20 mA			2.4		2.3	V
Off (dropout) voltage	V_{off}	$V_O \ge V_{CC}$ -0.1 V, $I_{OH} \le$ -20 mA		0.8		0.9		V
Output low voltage	V_{OL}	I_{OL} = 3.2mA, V_{CC} = 4.5V, V_{IH} = 2.4V			0.3		0.4	V
Output high voltage	V_{OH}	I_{OH} = -1.0mA, V_{CC} = 4.5V, V_{IL} = 0.8V		3.6		3.5		V
Input resistance	R_I	V_{CC} =5.0V		3.0	7.0	3.0	7.0	kΩ
Transmitter								
Output low voltage	V_{OL}	V_{CC} = 4.5V, V_{IH} = 2.0V, R_L = 3.0kΩ			-5.2		-5.0	V
Output high voltage	V_{OH}	V_{CC} = 4.5V, V_{IL} = 2.0V, R_L = 3.0kΩ		5.2		5.0		V
Input low current	I_{IL}	V_{CC} =5.5V, V_{IL} = 0V			-1.0		-10.0	μA
Input high current	I_{IH}	V_{CC} =5.5V, V_{IH} = V_{CC}			1.0		10.0	μA
Speed of output front change	SR	V_{CC} =5.0V, C_L =50 ~ 1000pF R_L = 3.0 ~ 7.0kΩ		3.0	30	2.7	27	V/μs
Output resistance	R_O	V_{CC} = V+ = V- = 0 V, V_O = ±2V		350		300		Ω
Short circuit output current	I_{SC}	V_{CC} =5.5V V_O = 0V	V_I = V_{CC}		-50		-60	mA
			V_I = 0 V		50		60	
Speed of information transmission	ST	V_{CC} =4.5V, C_L = 1000pF R_L = 3.0kΩ, t_W = 7ms (for extreme t_W = 8ms)		140		120		kbits/s

AC ELECTRICAL CHARACTERISTICS

Parameter	Symbol	Test Conditions	25°C		-40°C to +85°C		Unit
			Min	Max	Min	Max	
Signal propagation delay time when switching on (off)	t_{PHLR} (t_{PLHR})	V_{CC} = 4.5V, C_L = 150pF V_{IL} = 0V, V_{IH} = 3.0V t_{LH} = $t_{HL} \le$ 10ns		9.7		10	ms
Signal propagation delay time when switching on (off)	t_{PHLT} (t_{PLHT})	V_{CC} = 4.5V, C_L = 150pF V_{IL} = 0V, V_{IH} = 3.0V t_{LH} = $t_{HL} \le$ 10 ns		5.0		6.0	ms

CAPACITANCE

Parameter	Symbol	Test Conditions	Value	Unit
Input capacitance	C_{IN}	V_{cc}=5.0V	9.0	pF
Dynamic capacitance	C_{PD}		90	pF

TIMING DIAGRAM

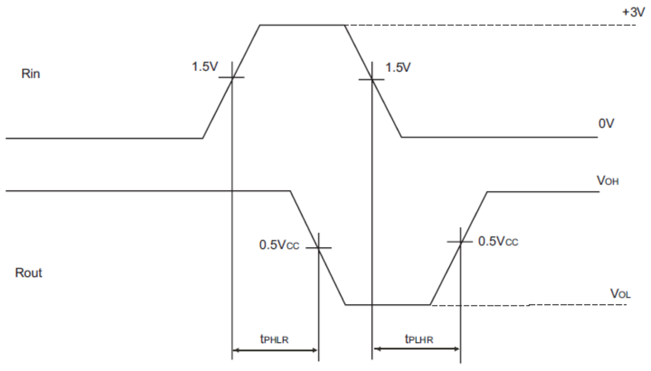


Figure 1. Waveforms for t_{PHLR} and t_{PLHR} Measurement

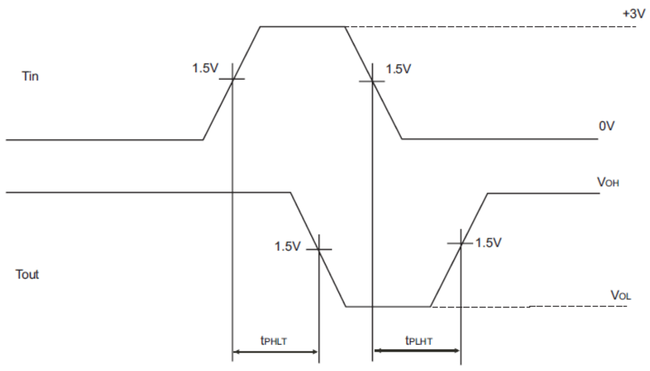


Figure 2. Waveforms for t_{PHLT} and t_{PLHT} Measurement

APPENDIX II

SOCKET APPLICATION SOURCE CODE

```
Dim intMax As Long
Dim ReceivedData As String

Private Sub cmdListen_Click()
Winsock1(0).LocalPort = txtPort.Text
Winsock1(0).Listen
cmdListen.Enabled = False
End Sub

Private Sub cmdStop_Click()
Winsock1(0).Close
cmdListen.Enabled = True
End Sub

Private Sub Form_Load()
intMax = 0
End Sub



Private Sub lstLog_DblClick()
Dim selected As Integer

lstLog.RemoveItem lstLog.ListIndex
lstLog.Refresh
End Sub

Private Sub Timer1_Timer()
```

```
lblTime.Caption = Format(Now, "hh:mm:ss ampm")
End Sub

Private Sub Winsock1_Close(Index As Integer)
Dim ServerMsg As String


intMax = intMax - 1
lblActiveConn.Caption = intMax
ServerMsg = "Disconnected : " & Winsock1(Index).RemoteHostIP
lstLog.AddItem ServerMsg
Unload Winsock1(Index)

End Sub

Private Sub Winsock1_ConnectionRequest(Index As Integer, ByVal requestID As Long)
Dim ServerMsg As String
If Index = 0 Then
intMax = intMax + 1
lblActiveConn.Caption = intMax
'creating new instance
Load Winsock1(intMax)
Winsock1(intMax).LocalPort = 0
Winsock1(intMax).Accept requestID
ServerMsg = "Connected: " & Winsock1(intMax).RemoteHostIP
lstLog.AddItem ServerMsg
End If

End Sub

Private Sub Winsock1_DataArrival(Index As Integer, ByVal bytesTotal As Long)
```

```
Dim Data As String
'Dim ID As String
Dim ServerMsg As String
Dim Stat, IntCount As Integer
Dim conConnection As New ADODB.Connection
Dim cmdCommand As New ADODB.Command
Dim rsRecordSet As New ADODB.Recordset
EndStr = String$(3, "+")

Winsock1(Index).GetData Data, vbString
'manipulate data
'+++ means other end is closing.so stop recording data
If Data = EndStr Then Exit Sub

ReceivedData = ReceivedData & Data
'lstLog.AddItem ReceivedData
If Right$(ReceivedData, 1) = "x" Then
'displaying text received
IntCount = Len(ReceivedData)
ServerMsg = Winsock1(Index).RemoteHostIP & ">>" & Left$(ReceivedData, IntCount - 1)
lstLog.AddItem ServerMsg
'ID = Left$(ReceivedData, IntCount - 2)
'Status = Right$(ReceivedData, 2) 'last position for stop char
lblLampID.Caption = Left$(ReceivedData, IntCount - 2)
lblStatus.Caption = Mid$(ReceivedData, IntCount - 1, 1)

conConnection.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\inetpub\wwwroot\FYP.mdb;Mode=read|write"
conConnection.Open
With cmdCommand
.ActiveConnection = conConnection
```

Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted
Formatted

```
    .CommandText = "SELECT * FROM NewUpdate;"
    .CommandType = adCmdText
    .Execute
End With

With rsRecordSet
    .CursorType = adOpenStatic
    .CursorLocation = adUseClient
    .LockType = adLockOptimistic
    .Open cmdCommand
    .AddNew
'need to change to suit with our socket connection
    .Fields(0) = Left$(ReceivedData, IntCount - 2)
    .Fields(1) = Mid$(ReceivedData, IntCount - 1, 1)
    .Fields(2) = Format(Now, "dd/mm/yyyy")
    .Fields(3) = Format(Now, "hh:mm:ss ampm")
    .Update
End With
rsRecordSet.Close

'open second database
With cmdCommand
    .ActiveConnection = conConnection
    .CommandText = "SELECT * FROM StatusUpdate;"
    .CommandType = adCmdText
    .Execute
End With
With rsRecordSet
    .CursorType = adOpenStatic
    .CursorLocation = adUseClient
    .LockType = adLockOptimistic
```

```
Open cmdCommand
AddNew
'need to change to suit with our socket connection
Fields(0) = Left$(ReceivedData, IntCount - 2)
Fields(1) = Mid$(ReceivedData, IntCount - 1, 1)
Fields(2) = Format(Now, "dd/mm/yyyy")
Fields(3) = Format(Now, "hh:mm:ss ampm")
Update
End With

rsRecordSet.Close
conConnection.Close
Set conConnection = Nothing
Set cmdCommand = Nothing
Set rsRecordSet = Nothing
Winsock1(Index).SendData vbCrLf & "OK" & vbCrLf
ReceivedData = ""
Else

Exit Sub

End If
End Sub
```

Formatted: Font: (Default) +Body, 10 pt, Not Bold, No underline, Font color: Auto

Formatted: Font: (Default) +Body, 10 pt, Not Bold

Formatted: Font: (Default) +Body, 10 pt, Not Bold, No underline, Font color: Auto

Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...

Formatted: Left, Line spacing: single

Formatted ...
Formatted ...

Formatted: Font: (Default) +Body, 10 pt

APPENDIX III

TRANSMISSION MODULE SOURCE CODE

```
#include "C:\Users\afiq\Documents\PIC\FYP_AFIQ\FYP_AFIQ.h"


//INITIALIZATION

int8 ptr,NO_LF;
char c,response1[30];
char OK[3]="OK",ERROR[6]="ERROR",WIPREADY[16]="+WIPREADY: 2, 1",CONNECT[8]="CONNECT";
int8 TrueVal=0,OverWrite=0,Update=0;
int8 InstNo;
#define ErrorLED PIN_B1
#define ConnOnLED PIN_B2

int1 Lamp1,Lamp2,Lamp3,Lamp4;
int1 Restart=0;
char IP_Addr[16]="10.139.163.33";

//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  ClearReceive  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
void ClearReceive(){
int8 i;
for(i=0;i<30;i++)response1[i]=0;//clear response array
ptr=0;
NO_LF=0;
c=0;
TrueVal=0;
}

void RecordResponse(){
do{
        c=getc();
        if(c=='\n' || c=='\r'){
        if(c=='\n')++NO_LF;
                        }
        else response1[ptr++]=c;
      }while (NO_LF<2);
      //append a null character
       response1[ptr]=0;
}
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  SendInst XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
void SendInst(int8 InstNo){
switch  (InstNo){

      case 1:
      //before send instruction, make sure everything is initialized
      ClearReceive();
      //send first instruction; TO Test Modem
      printf("AT\r");
      //record response
      RecordResponse();
      //check the response,VARY ON EVERY INSTRUCTION.recHEcK!!!!!
```

```
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 2:
ClearReceive();
//send 2nd instruction; TO START TCP/IP
printf("AT+WIPCFG=1\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 3:
ClearReceive();
//send 3ND instruction;OPEN GPRS BEARER
printf("AT+WIPBR=1,6\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 4:
ClearReceive();
//send 4RD instruction;START GPRS BEARER
printf("AT+WIPBR=4,6,0\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 5:
ClearReceive();
//send 5th instruction;OBTAINING IP ADDRESS
printf("AT+WIPBR=3,6,15\r");
RecordResponse();
if(strcmp(response1,ERROR)==0){
TrueVal=0;
break;
}
else {
ClearReceive();
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break;
}

case 6:
ClearReceive();
//send 6th instruction; SETUP IP SERVICE
printf("AT+WIPCREATE=2,1,\"%s\",\"9000\"\r",IP_Addr); //CREATING TCP SOCKET:change
here for destinnation ip and port
RecordResponse();
if(strcmp(response1,ERROR)==0){
TrueVal=0;
break;
}
else{
ClearReceive();
RecordResponse();
```

```
if(strcmp(response1,WIPREADY)==0)TrueVal=1;
}
break ;

//CLOSE CONNECTION
case 7:
ClearReceive();
// TO CLOSE IP SERVICES
printf("AT+WIPCLOSE=2,1\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 8:
ClearReceive();
// TO STOP GPRS BEARER
printf("AT+WIPBR=5,6\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 9:
ClearReceive();
// TO STOP GPRS BEARER
printf("AT+WIPBR=0,6\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 10:
ClearReceive();
// TO STOP TCPIP STACK
printf("AT+WIPCFG=0\r");
RecordResponse();
if(strcmp(response1,OK)==0)TrueVal=1;
break ;

case 11:
ClearReceive();
//Start Data service
printf("AT+WIPDATA=2,1,1\r");
RecordResponse();
if(strcmp(response1,CONNECT)==0){
TrueVal=1;
break;
}

case 12:
ClearReceive();
//send ETX.End of transmission
printf("+++");//this instruction for modem to be back to AT mode,not record by server
delay_ms(40000);//need dalay because modem sent many data
break;


}
```

```
}

void Blink (void){
while(OverWrite==0){//OverWrite can only be 1 when interrupted
output_low(ErrorLED);
delay_ms(100);
output_high(ErrorLED);
delay_ms(100);
}
OverWrite=0;
Restart=1;
}

#int_ext
void ExtInt(){
OverWrite=1;
}
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX InitConn XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

void InitConn(){
int8 i;
//TO INITIALISE MODEM,COULD BE WRITE IN SUBROUTINE
  //send first instruction; TO Test Modem
  for(i=0;i<3;i++){
  SendInst(1);//INST NO
  if(TrueVal==1)break;
  delay_ms(100);//
  }
  if(i==3)Blink();

  //send 2nd instruction; TO START TCP/IP
  SendInst(2);
  if(TrueVal==0)Blink();

  //send 3ND instruction;OPEN GPRS BEARER
  SendInst(3);
  if(TrueVal==0)Blink();

  //send 4RD instruction;START GPRS BEARER
  SendInst(4);
  if(TrueVal==0)Blink();

  //send 5th instruction;OBTAINING IP ADDRESS
  SendInst(5);
  if(TrueVal==0)Blink();

  //send 6th instruction; SETUP IP SERVICE
  SendInst(6);
  if(TrueVal==0)Blink();
  else {
  Restart=0;
  Output_high(ConnOnLED);
  }

}
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX CloseConn XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

```
void CloseConn(){

//7  TO CLOSE IP SERVICES
  SendInst(7);//INST NO
  if(TrueVal==0)Blink();

//8  TO STOP GPRS BEARER
  SendInst(8);//INST NO
  if(TrueVal==0)Blink();

//9  TO CLOSE GPRS BEARER
  SendInst(9);//INST NO
  if(TrueVal==0)Blink();

//10  TO CLOSE TCP/IP STACK
  SendInst(10);//INST NO
  if(TrueVal==0)Blink();
output_low(ConnOnLED);//indication data is sent properly

}
//XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX   SendData  XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
void SendData(){
int8 i=0;

//START SENDING DATA;IF CONNECT THEN BARU BLEY ANTAR
  SendInst(11);//INST NO
  if(TrueVal==0)Blink();

  //NOW SEND DATA
 // printf("103460x\r");//each data x as terminator
 for(i=0;i<3;i++){
 ClearReceive();
 if (Lamp1==1)printf("111111x");
 else printf("111110x");
 RecordResponse();
 if(strcmp(response1,OK)==0)break;
 delay_ms(100);
 }
 if(i==3)Blink();

 for(i=0;i<3;i++){
 ClearReceive();
 if(Lamp2==1)printf("222221x");
 else printf("222220x");
 RecordResponse();
 if(strcmp(response1,OK)==0)break;
 delay_ms(100);
 }
 if(i==3)Blink();

 for(i=0;i<3;i++){
 ClearReceive();
 if (Lamp3==1)printf("333331x");
 else printf("333330x");
 RecordResponse();
 if(strcmp(response1,OK)==0)break;
```

```
delay_ms(100);
}
if(i==3)Blink();

for(i=0;i<3;i++){
ClearReceive();
if(Lamp4==1)printf("444441x");
else printf("444440x");
RecordResponse();
if(strcmp(response1,OK)==0)break;
delay_ms(100);
}
if(i==3)Blink();
Update=0;
 //back to AT command
 SendInst(12);//INST NO
 if(TrueVal==0)Blink();
}

#INT_RB
void RB_isr(){

Lamp1=input(PIN_B4);
Lamp2=input(PIN_B5);
Lamp3=input(PIN_B6);
Lamp4=input(PIN_B7);

Update=1;
}
void main()
{
 setup_adc_ports(NO_ANALOGS);
 setup_adc(ADC_OFF);
 setup_psp(PSP_DISABLED);
 setup_spi(SPI_SS_DISABLED);
 setup_timer_0(RTCC_INTERNAL|RTCC_DIV_1);
 setup_timer_1(T1_DISABLED);
 setup_timer_2(T2_DISABLED,0,1);
 setup_comparator(NC_NC_NC_NC);
 setup_vref(FALSE);

 // TODO: USER CODE!!
 ext_int_edge(0,H_TO_L);
 enable_interrupts(int_EXT);
 //set_tris_b(0xF1);
 enable_interrupts(INT_RB);
 enable_interrupts(global);

 while(1){
 if(Update==1){
 InitConn();
 if (Restart==0)SendData();
 CloseConn();
 }
 }
 }
```

APPENDIX IV

WEB PAGE FILE

1. Street light monitoring main page

```
<html>

<head>
<title>HOME-Street Light Monitoring System</title>
<style type="text/css">

body {
background-image:url('utp.jpg');
background-repeat:no-repeat;
background-position:right top;}

table.center {margin-left:auto; margin-right:auto;}

</style>
<meta http-equiv="refresh" content="30" >
</head>

<body>
<%
set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open (Server.MapPath("FYP.mdb"))

dim ID_Lamp, Hari
ID_Lamp=request.querystring("Lamp_ID")
Hari=date()

IF ID_Lamp<>"" THEN
sql="DELETE FROM NewUpdate where LampID='"& ID_Lamp &"'"
on error resume next
conn.Execute sql
on error resume next
 conn.Execute sql
 if err<>0 then
  response.write("<H1 style=" & "color:red" & "> No update permissions!</H1>")
 else
  response.write("<H4 style=" & "color:red" & "> Record " & ID_Lamp & " was deleted!</H4>")
 end if

else
response.write("<H4 style=" & "color:#00FF00" & "> Record Updated</H4>")
END IF

set rs=Server.CreateObject("ADODB.recordset")
sql="SELECT * FROM NewUpdate ORDER BY Masa DESC"
rs.Open sql,conn

%>
```

```
</br></br>
<div style="text-align:center">
<H1 style="color: Blue">Street Light Monitoring System</H1>
</div>
<H3>New Updates</H3>


<table width="70%" cellspacing="0" cellpadding="2" border="1" class="center" style="text-align:center">
<tr>
<th bgcolor=#00FFFF>Lamp ID</th>
<th bgcolor=#00FFFF>Status</th>
<th bgcolor=#00FFFF>Date</th>
<th bgcolor=#00FFFF>Time</th>
</tr>
<%
do until rs.EOF
  if (Hari<>rs.fields("Tarikh")) then
  response.write("<tr bgcolor=#FF0000>")
  else
  response.write("<tr>")
  end if
  response.write("<td>" & rs.fields("LampID") & "</td>")
  response.write("<td>" & rs.fields("Status") & "</td>")
  response.write("<td>" & rs.fields("Tarikh") & "</td>")
  response.write("<td>" & rs.fields("Masa") & "</td>")
  response.write("</tr>")
  rs.MoveNext
loop
rs.close
set rs=Nothing

set rs=Server.CreateObject("ADODB.recordset")
sql="SELECT DISTINCT LampID FROM NewUpdate ORDER BY LampID"
rs.Open sql,conn

%>
</table>

</br></br>

<form method="get">
<INPUT type="button" value="Update" onClick="location.href='Update.asp'">
<input type="submit" value="Delete">
Choose Light ID :  <select name="Lamp_ID">
<%
do until rs.EOF
   response.write("<option")
   if rs.Fields("LampID") = ID_Lamp then
   response.write(" selected")
   end if
   response.write(">")
   response.write(rs.fields("LampID"))
   response.write("</option>")
   rs.MoveNext
```

```
loop
rs.close
set rs=Nothing
%>
</select>
</form>
<BR /><BR />
Page Reloaded at : <% response.write(Now())%>
<HR/>

<H4>ARCHIVE</H4>
<%
set rs=Server.CreateObject("ADODB.recordset")
sql="SELECT DISTINCT LampID FROM StatusUpdate ORDER BY LampID "
rs.Open sql,conn
%>
<form method="get" action="archive.asp">
<p>View records for: </p> <select name="Lamp_ID">
<%
do until rs.EOF
  response.write("<option")
  if rs.Fields("LampID") = ID_Lamp then
  response.write(" selected")
  end if
  response.write(">")
  response.write(rs.fields("LampID"))
  response.write("</option>")
  rs.MoveNext
loop
rs.close
set rs=Nothing
conn.Close
set conn=Nothing
%>
</select>
<input type="submit" value="View">
</form>

</body>
</html>
```

Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted ...
Formatted: Font: 10 pt, No underline, Font color: Auto
Formatted: Space After: 0 pt, Line spacing: single
Formatted: Font: (Default) +Body, 10 pt
Formatted: Font: (Default) +Body, No underline, Font color: Auto

2. Street Light Monitoring Update page

```
<html>
<body>
</br></br>
<div style="text-align:center;">
<img src="utp.jpg" width="100" height="195" />
<H1>Welcome To Update Page</H1>
<br/><br/>
<form name="LampStatus" action="UpdateStatus.asp" method="get">
Lamp ID : <input type="text" name="LampID"><br/>
Status : <input type="radio" name="Status" value="1"/> ON
<input type="radio" name="Status" value="0"/> OFF <br/>
<input type="submit" value="Update">
<input type="reset" value="Cancel">
</form>
</div>
<a href="Home.asp"><img src="Home.png" width="60" height="60"></a>
</br>Return to Home Page
</body>
</html>
```

3. Street Light Monitoring Archive page

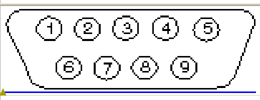
```
<html>
<body>
</br></br>
<div style="text-align:center; color:Blue">
<img src="utp.jpg" width="100" height="195" />
<H1>ARCHIVE</H1>
<br/><br/>
</div>
<%
dim ID_Lamp
ID_Lamp=request.querystring("Lamp_ID")

set conn=Server.CreateObject("ADODB.Connection")
conn.Provider="Microsoft.Jet.OLEDB.4.0"
conn.Open (Server.MapPath("FYP.mdb"))

set rs=Server.CreateObject("ADODB.recordset")
sql="SELECT DISTINCT LampID FROM StatusUpdate ORDER BY LampID "
rs.Open sql,conn
%>
<form method="get">
View records for: <select name="Lamp_ID">
<%
do until rs.EOF
   response.write("<option")
   if rs.Fields("LampID") = ID_Lamp then
```

```
  response.write(" selected")
  end if
  response.write(">")
  response.write(rs.fields("LampID"))
  response.write("</option>")
  rs.MoveNext
loop
rs.close
set rs=Nothing
%>
</select>
<input type="submit" value="View">
</form>

<%
set rs=Server.CreateObject("ADODB.recordset")
sql="SELECT * FROM StatusUpdate WHERE LampID='"& ID_Lamp &"' ORDER BY Masa"
rs.Open sql,conn
%>

<table width="100%" cellspacing="0" cellpadding="2" border="1" style="text-align:center">
<tr>
<th bgcolor=#00FFFF>Lamp ID</th>
<th bgcolor=#00FFFF>Status</th>
<th bgcolor=#00FFFF>Date</th>
<th bgcolor=#00FFFF>Time</th>
</tr>

<%
do until rs.EOF
  response.write("<tr>")
  response.write("<td>" & rs.fields("LampID") & "</td>")
  response.write("<td>" & rs.fields("Status") & "</td>")
  response.write("<td>" & rs.fields("Tarikh") & "</td>")
  response.write("<td>" & rs.fields("Masa") & "</td>")
  response.write("</tr>")
  rs.MoveNext
loop

rs.close
set rs=Nothing

%>
</table>
<p></p>
<a href="Home.asp"><img src="Home.png" width="60" height="60"></a>
</br>Return to Home Page
</body>
</html>
```

APPENDIX V

9 PIN CONNECTOR

~~APPENDIX 1: TCP/IP CLIENT – SODA MACHINE (SHEET 1 OF 3)~~

	9 Pin Connector on a DTE device (PC connection)
Male RS232 DB9	
Pin Number	**Direction of signal:**
1	Carrier Detect (CD) (from DCE) Incoming signal from a modem
2	Received Data (RD) Incoming Data from a DCE
3	Transmitted Data (TD) Outgoing Data to a DCE
4	Data Terminal Ready (DTR) Outgoing handshaking signal
5	Signal Ground Common reference voltage
6	Data Set Ready (DSR) Incoming handshaking signal
7	Request To Send (RTS) Outgoing flow control signal
8	Clear To Send (CTS) Incoming flow control signal
9	Ring Indicator (RI) (from DCE) Incoming signal from a modem

Formatted: Font: (Default) +Body, 10 pt

Formatted: Space After: 0 pt

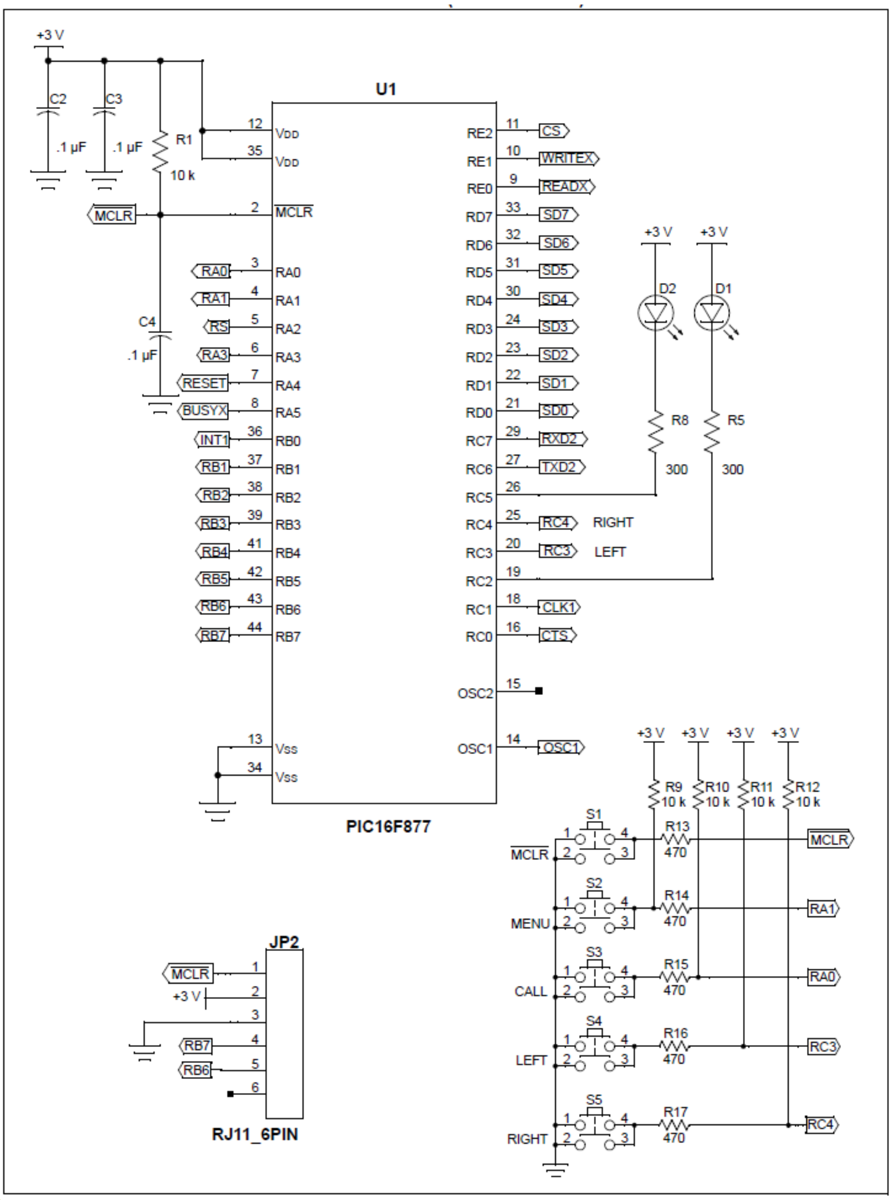
Formatted: Font: 14 pt

Formatted: Centered

Formatted: Font: 14 pt

Formatted Table

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto



Formatted: Font: (Default) Times New Roman, 14 pt, Bold, No underline, Font color: Auto

Formatted: Line spacing: single

Formatted: Font: 14 pt, Bold

Formatted: Font: 14 pt

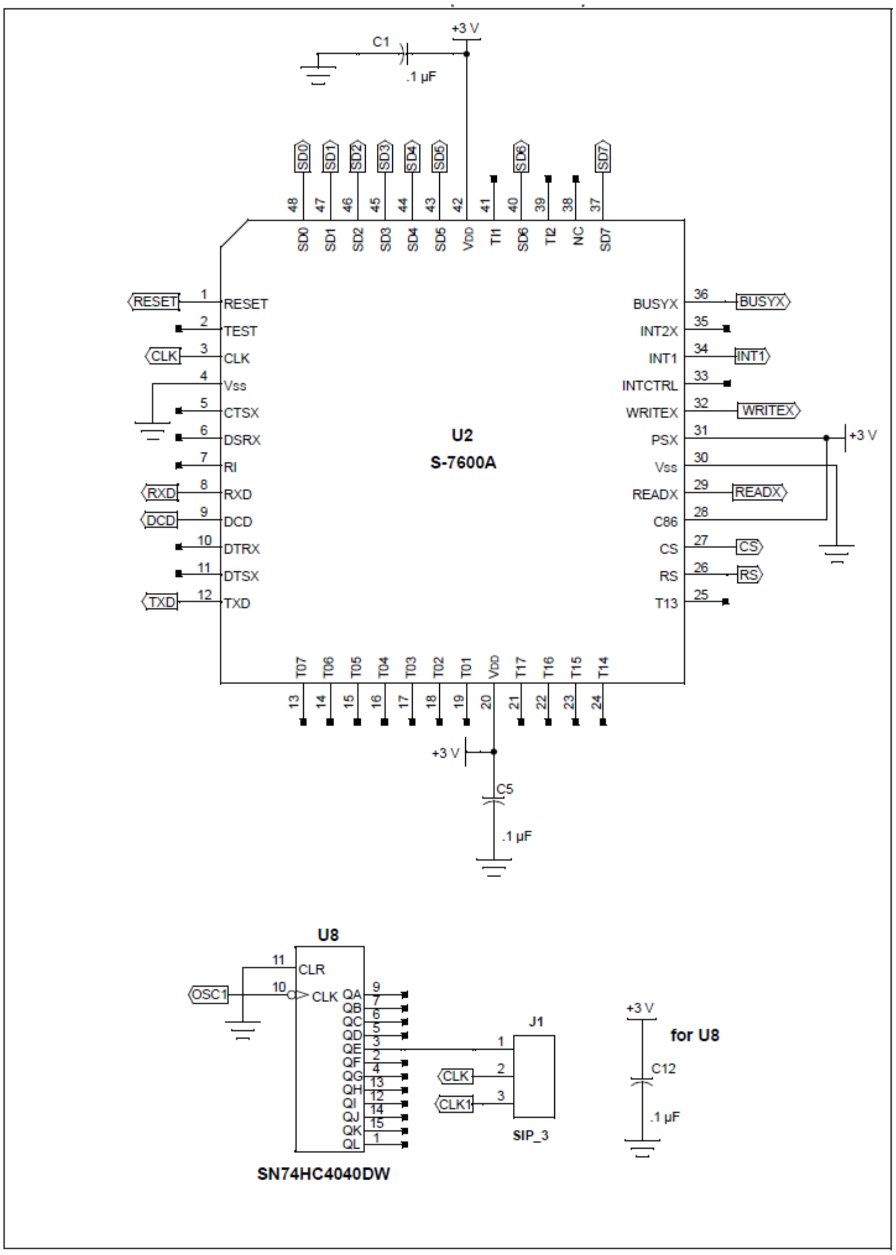
Formatted: Space After: 0 pt, Line spacing: single, Tab stops: Not at 1.04"

~~APPENDIX 1: TCP/IP CLIENT – SODA MACHINE (SHEET 2 OF 3)~~

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

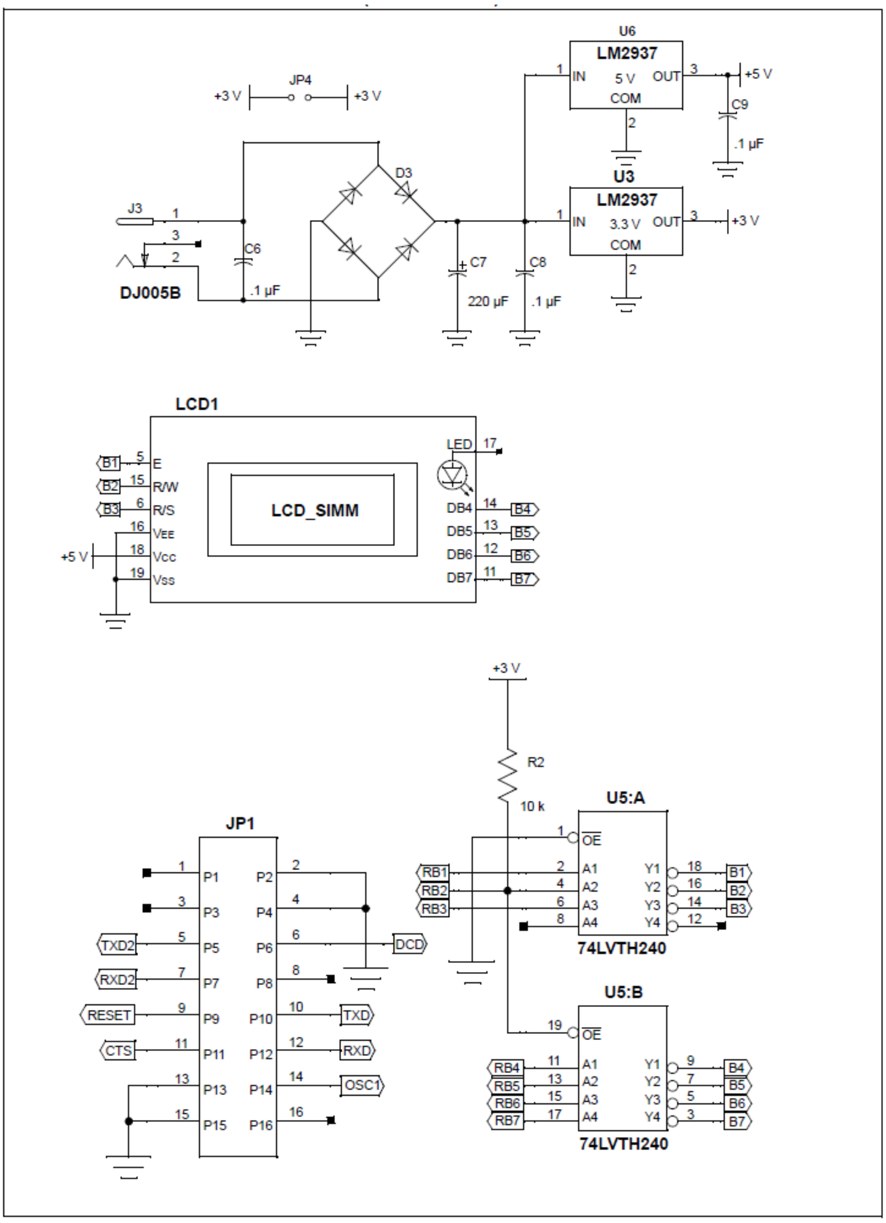
Formatted: Space After: 0 pt, Line spacing: single

Formatted: Space After: 0 pt, Line spacing: single, Tab stops: Not at 1.04"



~~APPENDIX 1: TCP/IP CLIENT – SODA MACHINE (SHEET 3 OF 3)~~

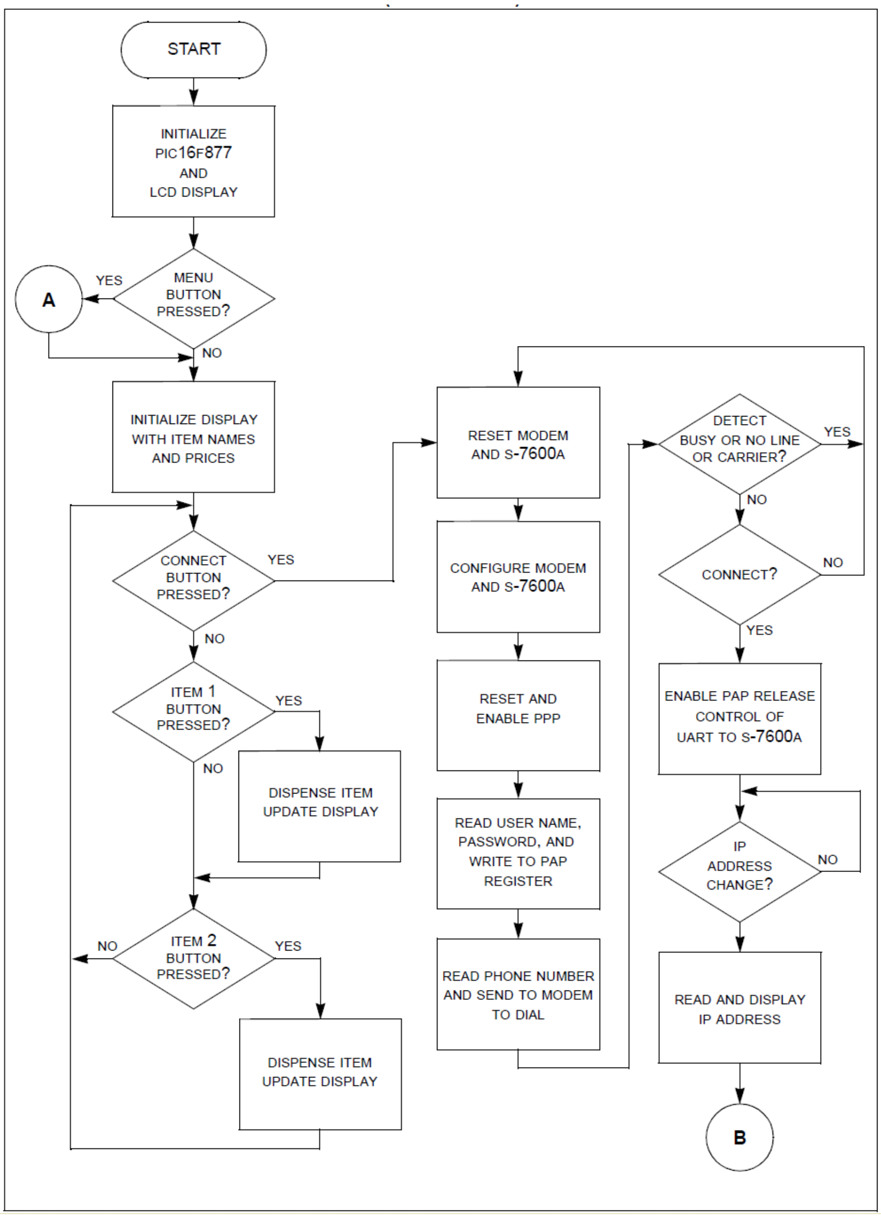
Formatted: Space After: 0 pt, Line spacing: single



~~APPENDIX 2: WEB CLIENT FLOWCHART (SHEET 1 OF 3)~~

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

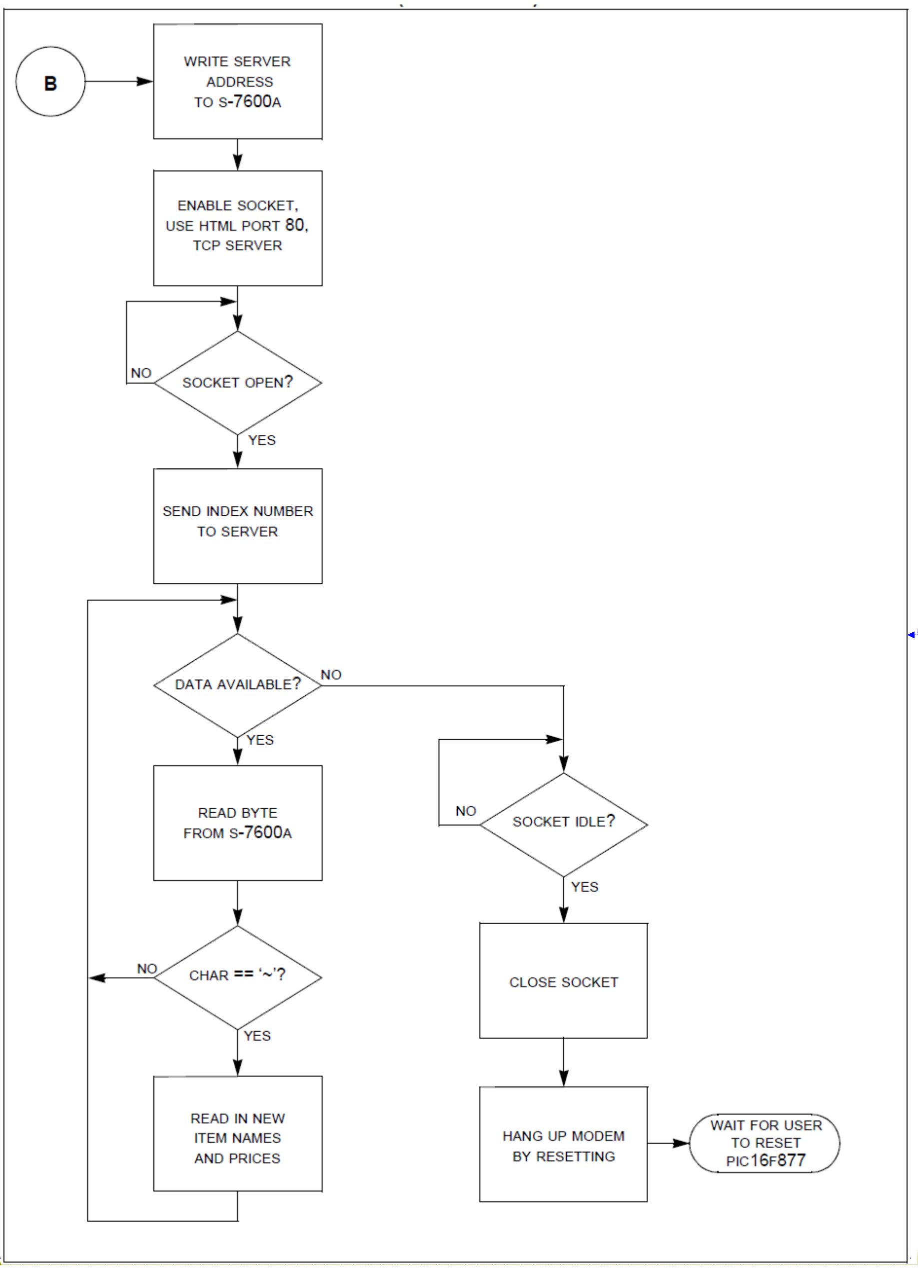
Formatted: Space After: 0 pt, Line spacing: single, Tab stops: Not at 1.04"



Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

~~APPENDIX 2: WEB CLIENT FLOWCHART (SHEET 2 OF 3)~~

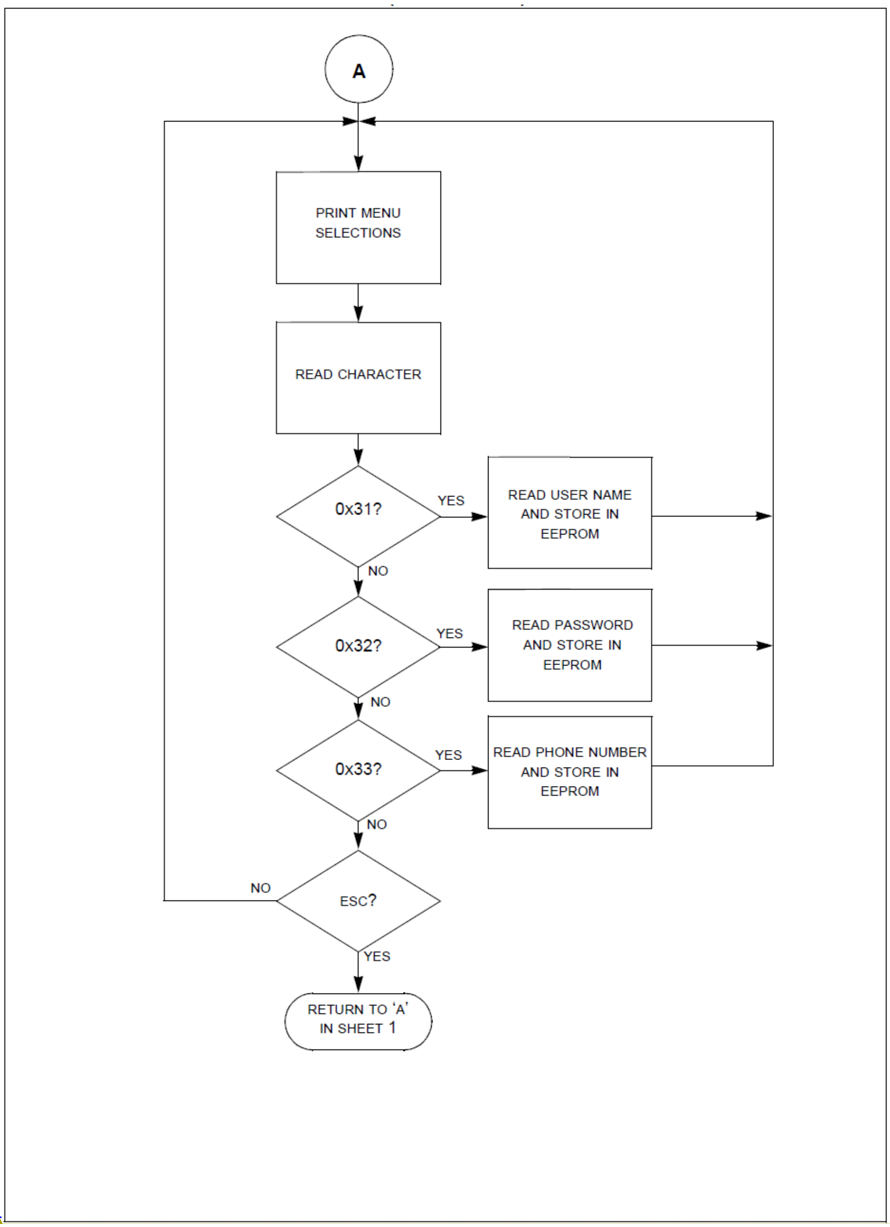
Formatted: Space After: 0 pt, Line spacing: single



~~APPENDIX 2: WEB CLIENT FLOWCHART (SHEET 3 OF 3)~~

Formatted: Font: (Default) Times New Roman, 12 pt, No underline, Font color: Auto

Formatted: Space After: 0 pt, Line spacing: single, Tab stops: Not at 1.04"


A
PRINT MENU SELECTIONS
READ CHARACTER
0x31?
YES
READ USER NAME AND STORE IN EEPROM
NO
0x32?
YES
READ PASSWORD AND STORE IN EEPROM
NO
0x33?
YES
READ PHONE NUMBER AND STORE IN EEPROM
NO
ESC?
NO
YES
RETURN TO 'A' IN SHEET 1