Smart Helmet

by

Mohammed Syahid Nafi' bin Mohd Yusof

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Business Information Systems)

JANUARY 2009

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

SMART HELMET

by

Mohammed Syahid Nafi' bin Mohd Yusof

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Business Information Systems)

Approved:

Dr. Halabi Hasbullah
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

JANUARY 2009

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

Mohammed Syahid Nafi' bin Mohd Yusof

ABSTRACT

This report is written for the purpose of highlighting the background of the research project for the SMART HELMET, the scope of the study, research method and the literature review.

SMART HELMET is a system which aims to make all motorcyclists in Malaysia aware and compulsory to wear helmet whether the travel distance is in 100 meter radius or long distance. The system will use XBee technology which will connect from the transmitter at helmet to the receiver at motorcycle. Many type of switches being used such as temperature heat switch, clipped switch, and signal as a switch to make sure the motorcyclist not cheating to their self. If the system identified that the riders or user not wearing their helmet properly (clipped), the signal won't be send to the receiver at motorcycle which will cause the motorcycle can not start and being ride by motorcyclist.

The scope of the study will be using others studies and statistics from Malaysia government agencies in term of Road Safety; fatal motorcyclist accident causes and focus on the helmet wearing attitude and behavior. The scope also will cover on research of overview XBee Technology applications, behavior, characteristics as well as advantages and disadvantages of XBee Technology. Then will cover on implementation of the system in real daily life.

ACKNOWLEDGEMENTS

The author wishes to take the opportunity to express his utmost gratitude to the individual that have taken the time and effort to assist the author in completing the project. Without the cooperation of these individuals, no doubt the author would have faced some minor complications through out the course.

First and foremost the author's utmost gratitude goes to almighty Allah SWT without his blessing this project cannot be done fluent and successfully. To the author's parent, Mr Mohd Yusof bin Abd Jalal and Mrs. Zahrah Mohd Radzi who is able to heard and listen all the problems that occur and also supporting author in various type of support. Without them, who will be the author. To the author's supervisor, Dr. Halabi Hasbullah. Without his guidance and patience, the author would not be succeeded to complete the project. To the Final Year Project Coordinator, Ms. Savitha Sugathan and other CIS Department Final Year Project committees for providing his with all the initial information required to begin the project.

To all CIS lecturers who have been willing to give appropriate help, support and guidance in completing his research.

Last but not least, to all individuals that has helped the author in any way, but whose name is not mentioned here, the author thank you all.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

V	Volt
RF	Radio Frequency
PDRM	Polis DiRaja Malaysia
Hz	Hertz
DIY	Do It Yourself
LCD	Liquid Crystal Display
SK	Start-Up Kit
m	meter

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

Smart Helmet Project is a project is a project on how to make the proper helmet wearing can be compulsory implemented to all Malaysia citizen by developing a connection between the motorcycle helmet and the motorcycle. This project consist of study of fatal road accident and causes which involving motorcyclist, study on creating connection between motorcycle and helmet by using RF module, and developing a prototype on Smart Helmet itself. Helmet been created for the daily use when riding motorcycle in order to protect human or user head from seriously injured when accident happen. A quality helmets are made of a combination of fibers, glass and carbon on the outer shell and special polystyrene foam with fire-retardant lining on the inside to help absorb impact and prevent burns [1]. Before a quality helmet being sold to the motorcyclist, this helmet will be going for various types of inspections to make sure it is very safe and comfort to be used by the end user.

This Smart Helmet project was initiated especially to focus on connecting the helmet with the motorcycle by using a connection device or in specifically by using an RF module. This project is design to uncompromising the proper helmet usage among the motorcyclist. As we know, the improper helmet usage will cause a serious bad injury to the head of motorcyclist when an accident happens to them. It show that the motorcyclists as a focus group have the highest fatality and serious injury when the proper helmet wearing not implemented because their head will directly expose to the hazard when the helmet pull out from their head. With this Smart Helmet, the user will not be able to start their motorcycle if they are not wearing helmet tidily. The system of the smart helmet itself been computerized to make sure that the helmet must be neatly clipped before the user can start the ignition of their motorcycle.

The Smart Helmet also works as a medium or tool to make the motorcycle safe from being stolen by motorcycle thieves. This is because, by having the uniquely connection between motorcycle and helmet, the helmet will be created specifically unique to the owner of the motorcycle and specific motorcycle only can be use by user who have the specifically connected helmet. By having this function, the motorcycle owner can happily do their other job by reducing their mind on thinking of their motorcycle security.

This Smart Helmet is being connected to motorcycle by using RF module which is the Xbee RF module. This type of RF module is created to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices [2]. The module can be connected uniquely by giving a unique 8 bits address and support over 65000 unique addresses. This module operates within the ISM 2.4 GHz frequency band and is pin-for-pin compatible with each other [2]. This RF module was controlled by a PIC18 microcontroller which can configure the figuration of the XBee address, controlling the data in and out from the RF module as well as controlling the entire circuit being attached to the both helmet and motorcycle.

1.2 PROBLEM STATEMENT

The main problem in Malaysia and especially in rural areas is motorcyclists are most vulnerable to fatal road accident. One of the main reasons is because of failure to wear helmet in a proper way. This is actually caused by the attitude of the motorcyclists themselves who has the opinion that wearing helmet is unnecessary if the destination is just a stone-throw away or wearing helmet is enough as long as they just wear it without clipping the helmet's lock. When an accident happen, the improper helmet will push out from rider's head and the risk to have bad or serious head injury also increase which may cause to fatality.

1.2.1 Problem Identification

The problems identified are based on the common problems that happen among the motorcyclist while riding motorcycle and wearing helmet. The problems that the motorcyclist usually faces are:

i. **Motorcyclist helmet wearing attitude**

It is an attitude issues involve about the usage of the motorcycle helmet. The main attitude problem of motorcyclist nowadays is attitude that wearing helmet is not necessary when riding motorcycle. In Malaysia, we can see most of the rural area motorcyclist not wearing the helmet while riding their motorcycle at the open road. This kind of attitude leads the increment on the fatal road accident in the rural site of Malaysia caused by seriously injury of the motorcyclist head. In the statistic provided by the Polis DiRaja Malaysia, PDRM below (table 1.0); 49.2% motorcyclist fatality is caused by the badly injured of the motorcyclist head for the year 1997 in Malaysia [3]. this statistic show that, it is slightly 50% of the rider or motorcyclist have the attitude that wearing a helmet is not one of the important element in riding motorcycle.

Other attitude that occurs to the motorcyclist is about not wearing helmet in the appropriate way or improperly. This kind of attitude can be described by wearing a helmet without the strip or the strip not being fastened correctly. When this kind of attitude happen among Malaysia citizen, when there is accident occur involving them, the helmet which not properly tied will pull out from the rider head and will lead the high exposure of the hazard directly to their head. The consequences of this situation, the motorcyclist head can be hit and can cause the serious damage for his or her head as well as lead to the motorcyclist fatality.

Motorcyclist Fatalities by Part of Body Injured

Part of Body	Fatalities	Percentage (%)
Head	1766	49.2
Neck	95	2.6
Chest	256	7.1
Arms	17	0.5
Back	14	0.4
Hips	23	0.6
Legs	50	1.4
Multiple	1371	38.2
Total	3592	100

(Source: PDRM, 1997)

Table 1.0: Motorcyclist Fatality by Part of Body Injured [3]

ii. Fatal motorcyclist accident

Road Fatality Statistics for 2007 and 2008 (Jan-Dec)

CONSUMER CATEGORY	2007	%	2007 JAN-DEC	2008 JAN-DEC	DIFFERENCE	%
(TOP RISK CATEGORY)						
Passenger/Driver Car	1228	19.3	1228	1335	107	8.7
Motorcycle Pillion/Rider	3646	58.7	3646	3898	252	6.9
Pedestrian	636	9.5	636	598	-38	-6.0
(OTHERS CONSUMER)						
Pillion/Rider Bicycle	190	3.8	190	203	13	6.8
Bus Passenger/Driver	75	0.6	75	48	-27	-36.0
Driver / Lorry Attendant	204	3.6	204	195	-9	-4.4
Driver / Attendant Van	133	1.6	133	96	-37	-27.8
Driver / Attendant Race 4 Wheel	99	1.7	99	106	7	7.1
Others Vehicle	71	1.0	71	48	-23	-32.4
TOTAL	6282	100	6282	6527	245	3.9

Table 2.0: Road Fatality Statistics for 2007 and 2008 (Jan-Dec) [4]

For this problem, the problem being referred to the statistic which have been produced by the Polis DiRaja Malaysia above (table 2.0) [4]. The table shows the road fatality statistic in Malaysia for year 2007 and 2008 from month of January to the December in both year. Based on the table, the motorcycle pillion and rider contribute 58.7% of fatal road accident in 2007 and the value increase by 6.9% for the next year, 2008. it show that, eventhough the government taking the serious action among the motorcyclist policies in term of campaigns and summon distribution, still the number of motorcyclist fatality increase year by year. What we can evaluate here is, it come back to the attitude problem of the motorcyclist again.

For this time, it is about attitude problem among the motorcyclist which not aware to the campaigns that have been produced by Malaysia government. Sometime, the motorcyclist see this kind of programs and campaign not necessary for them and take it as the simple thing and some of them even forgot what is the campaigns all about after attending it. Other attitude that occurs here is about the attitude that the police summon can be pay easily. It is not about the summons been produce not in the large quantity or the police not take it seriously, but it is about the money value nowadays. The motorcyclist think that if they being summon because not wearing helmet or wearing helmet improperly, they still have ability to pay that summon even their summon already cumulated over a year. This both kind of attitude should be solve immediately and one of the way by producing some enforcement in wearing helmet or make the helmet compulsory to be wear while riding motorcycle.

1.3 OBJECTIVES OF PROJECT

1.3.1 To utilize the advantages of Xbee technology related to safety

By using Xbee technology, it will be possible to connect the helmet to cooperate with the motorcycle which makes them dependent to each other. This project also will be able to configure the RF module by using a microcontroller which can control the addresses, data send and receive and also the flow of the circuit which attached to both helmet and motorcycle. It can be related to the safety because of the XBee RF module is used in order make the connection between helmet and motorcycle to increase safety for motorcyclist.

1.3.2 To ensure maximum adherence of motorcyclist towards roads safety regulation in the aspect of helmet-wearing

Other objective if this project is about ensuring the maximum adherence of motorcyclist toward road safety. By creating dependencies between both helmet and motorcycle, motorcyclist will need helmet to ride the motorcycle in any circumstances. It will directly decrease their risk of head injury when accident happens while riding on road while ensuring that they adhere to the roads' safety regulations.

1.3.3 To give more focus while riding motorcycle.

The last but not least objective of this project is about increasing motorcyclist focus while riding motorcycle. The focus can be related with the motorcyclist way of thinking and what appear in their mind while riding motorcycle. When this project already developed and implemented, the motorcyclist mind will be clear from the unnecessary thing of thinking such as thinking about proper helmet wearing related things, way of avoiding policeman in any circumstances and also problem which will appear if the policeman summon them caused by not wearing helmet. All these thinking matter can be solve when the motorcyclist only and can only used motorcycle after wearing helmet properly.

1.4 SCOPE OF STUDY

This study will include research on the latest Xbee technology, function, security and also threat. It also will include other futures on microcontroller in handling the circuit, flow of data transmitted and received from RF modules and also in producing unique entity which can differentiate this project with others same project. After researches have been made, blueprint of the entire system will be developed and also the compatibility of the system with the hardware and tools required. After the system generation level complete, the study will continue with the application placement which will be suitable to use and also some modification of XBee RF module in to helmet and motorcycle. After modification is made, it is time to compiling both software and hardware and also test the entire system on the real-life to make sure its work properly and efficiently implemented.

CHAPTER 2

LITERATURE REVIEW

To gain a better understanding and to study future on what is Xbee, additional readings are essential in order to make a research in the Xbee technology area. From this reading, it enables and gives more clarification of the Xbee RF module concept, structure and also to gather more ideas on hoe to enhance this Xbee module in order to meet the user requirement especially to the motorcyclist. Other than that, a research on the area of road safety also has been done in order to get an overview of how to relate the theory of road safety and regulation with this Smart Helmet. The research also will cover the study on the Microcontroller technology which will be used through out this project and also get some clear review on what is microcontroller, microcontroller function and how its works. Throughout the research and overview among the journal and related medium of research, there are some new knowledge, idea as well as technologies that have been found which really help and can be used in the process of Smart Helmet development.

2.1 WHAT IS XBee (also be known as ZigBee)

According to the Xbee developer company, Maxstream,Inc. Xbee RF module is a 20 pins receive devices which can be mounted in the interface device. For more specific, Xbee RF module is the module that been engineered to meet IEEE 802.15.4 standards and support the unique needs of low-cost, low-power wireless sensor networks. The modules require minimal power and provide reliable delivery of data between devices. The modules operate within the ISM 2.4 GHz frequency band and are pin-for-pin compatible with each other [1]. This Xbee module supporting many kind of network topology such as single peer, multi-peer and also broadcast topology as shown below [5](Figure1.0).

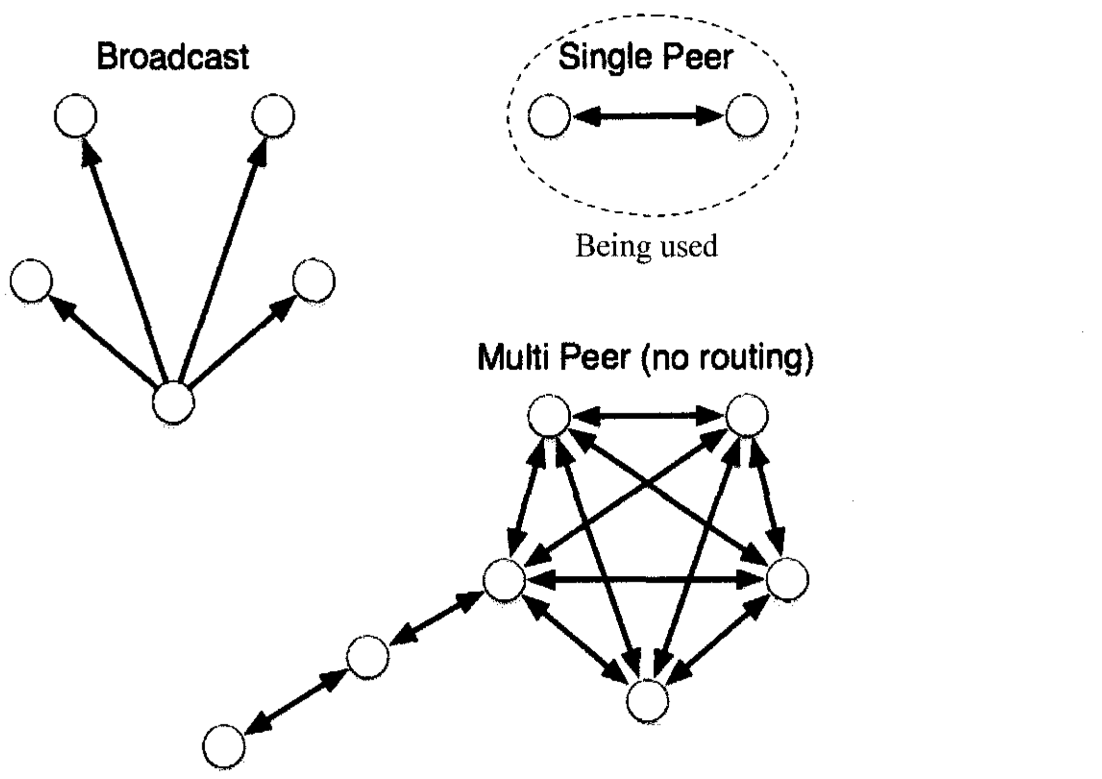


Figure 1.0: 802.15.4 Devices Network Topologies [5]

2.2 XBee ADVANTAGES

The advantages of using Xbee RF module as the main data receiver and transmitter are it is the latest wireless technology in Malaysia nowadays, it can be differentiate with another Xbee by producing own unique serial number, it is low cost, low power consuming in wireless sensor network and most of it, it is ease to be used by the developer.

In term of ease to be used, the Xbee module not needs any configuration for out-of box radio frequency communication. It is because all the configuration can be set up by using the X-XTU Software which applicable in testing and configure the XBee RF module. (Figure 2.0)

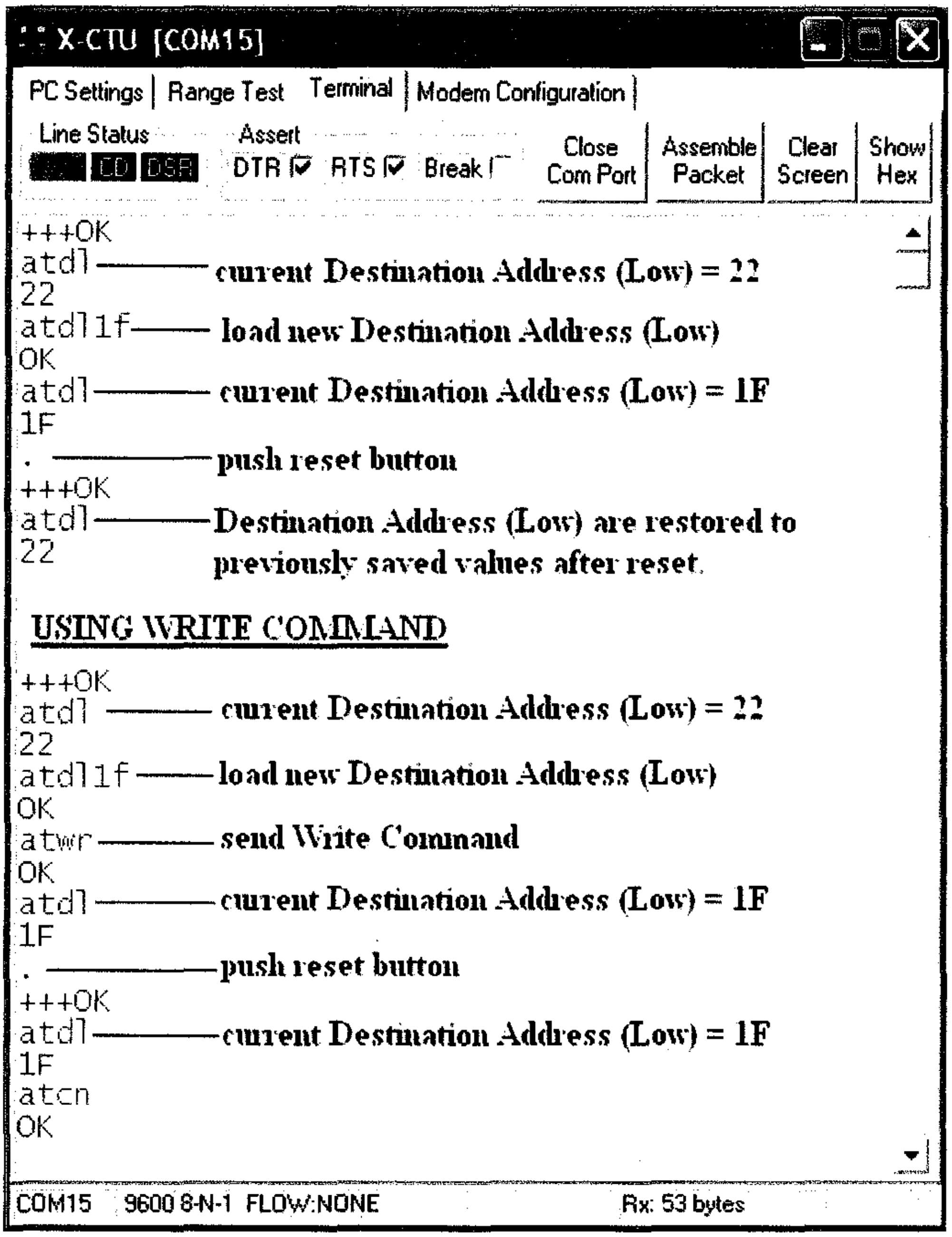


Figure 2.0: example of X-CTU interface. [8]

It is also easy when it com for command mode, because it is a wireless communication module, this Xbee RF module supporting both AT and API Command modes for the module parameters configuration as shown below(Figure 3.0)[5].

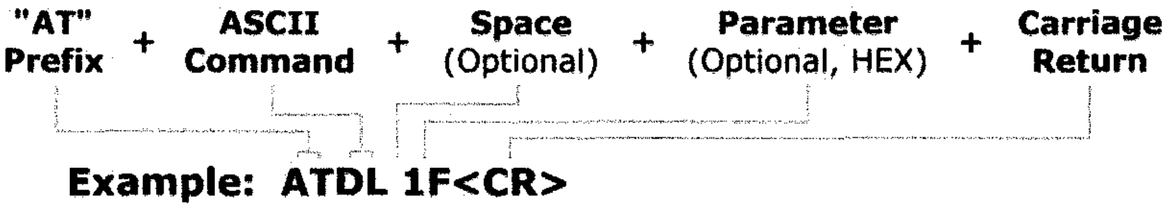


Figure 3.0: example of AT Command mode parameter [5]

When it come for the latest wireless technology advantage in Malaysia, choosing Xbee RF module for this project will create great advantages in term of promoting this technology and also differentiate this wireless technology with others matured technology such as Bluetooth technology. When it is different, it may lead for competitive advantages of this project or product when it entering the market.

In term of low cost and low power advantage, operation the Xbee module only using 3.3V of power [1] and it cost only RM200 for this kind of modules. When the power consuming only 3.3V, it is possible to run it by using 9V battery which is small, relevant and compatible to be attach inside helmet and also motorcycle. For more specification detail, refer to table 3.0 below:

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	up to 100 ft. (30 m)	Up to 300 ft. (100 m)
Outdoor RF line-of-sight Range	up to 300 ft. (100 m)	Up to 1 mile (1500 m)
Transmit Power Output (software selectable)	1mW (0 dBm)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP*
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 115200 bps (non-standard baud rates also supported)	1200 - 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 – 3.4 V	2.8 – 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	If PL=0 (10dBm): 137mA(@3.3V), 139mA(@3.0V) PL=1 (12dBm): 155mA (@3.3V), 153mA(@3.0V) PL=2 (14dBm): 170mA (@3.3V), 171mA(@3.0V) PL=3 (16dBm): 188mA (@3.3V), 195mA(@3.0V) PL=4 (18dBm): 215mA (@3.3V), 227mA(@3.0V)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 µA	< 10 µA
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85º C (industrial)	-40 to 85º C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector	Integrated Whip, Chip or U.FL Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	n/a	005NYCA0378 (Max. 10 dBm transmit power output)**

Table 3.0: Xbee and XBee Pro RF Module specification [1]

2.3 INTERGRATING ROAD SAFETY WITH Xbee RF MODULE

Because this Smart Helmet project is a project involving helmet and motorcycle and also targeting the motorcyclist as the targeted end user, it is automatically related to the road safety towards the helmet wearing policies. The main idea of this project also related to the road safety which the helmet need to be warn properly [4] before the user can run the motorcycle on the road. The study on the road safety has been focused on the helmet wearing policies in Malaysia for motorcyclist. The studies also includes the statistic that have been produced by PDRM (Table 4.0)[4] and also the campaign research on motorcycle safety in term of proper usage of the safety helmet advertisement as shown below(Figure 4.0).

Fatality Comparison of Motocycle Rider and Pillion between State 2007 and 2008 (Jan - July)

STATE	2007 JAN-JULY	2008 JAN-JULY	DIFFERENCE	%
PERLIS	22	37	15	68.2
KEDAH	191	193	2	1.0
P.PINANG	159	154	-5	-3.1
PERAK	237	278	41	17.3
SELANGOR	351	410	59	16.8
K.LUMPUR	78	82	4	5.1
N.SEMBILAN	115	146	31	27.0
MELAKA	83	82	-1	-1.2
JOHOR	333	361	28	8.4
PAHANG	137	132	-5	-3.6
KELANTAN	129	127	-2	-1.6
TERENGGANU	97	93	-4	-4.1
SABAH	42	60	18	42.9
SARAWAK	76	84	8	10.5
TOTAL	2050	2239	189	9.2

Source : PDRM

Table 4.0: Fatality Comparison of Motorcycle Rider and Pillion between State 2007 and 2008 (Jan-July) [4]

Figure 4.0: example of Safety Helmet advertisement on billboard [6]

On the proper usage of the safety helmet advertisement on television and billboard campaign, there is quite impressive result have been found which are over 78% of the 750 respondents were able to recall the advertisement slogan, 97% respondent agree with the message and 90% respondent claims do follow the campaign proposition[6]. But, when this project already developed, from 78% who recall the slogan can be change to 100% not only recall but applying the slogan and from 90% do follow the campaign proposition changes to 100% do follow the rules due to compulsory procedure of wearing helmet properly before riding motorcycle.

2.4 WHAT IS PIC MICROCONTROLLER

According to Tim Wilmshurst [7] PIC microcontroller is a family that consists of various types of microcontrollers and the architecture is base on a modified Harvard RISC (Reduced Instruction Set Computer) instruction set with dual-bus architecture, providing fast and flexible design with an easy migration path from only 6 pins to 80 pins, and from 384 bytes to 128 kilobytes of program memory. PIC microcontroller also available with many different specification depending on memory type, input-output (I/O) pin count, memory size and other special features including CAN, USB, LCD, motor control, and radio frequency. For this Smart Helmet project, the PIC 16F877A microcontroller will be used which is a 40 pins PIC consist of program memory, data memory, I/O ports, and timers for developing this project.

2.5 ADVANTAGES OF PIC MICROCONTROLLER

The advantages of PIC microcontroller for the developer are varies and one of the key issue is compatible. According to Tim Wilmsmhurst [7], although there are many models of microcontrollers in the PIC family, they all share some common features, such as program memory, data memory, I/O ports, and timers. Some devices have additional features such as A/D converters, USARTs and so on. Because of these common features, we can look at these attributes and cover the operation of most devices in the PIC family. For this Smart Helmet Project, the advantage of using PIC microcontroller is easy to developing the project and it is ease to configure the microcontroller for the short time basis project. Because PIC16F877A microcontroller also shares the common feature with other microcontrollers, it is also one of the advantages in using this kind of microcontroller for this Smart Helmet project.

2.6 EXAMPLE OF XBEE PROJECT

In order to make further study of Xbee RF module, study on the example of Xbee program is needed. For this time, one of the Cytron Technology [8] DIY projects has been choose which the title of the project is "Multifunction Mobile Robot". Even though it is a robotic project, but the usage of the Xbee RF module is available and it is one of the main program in make the robot functional. For instance, this project is about a robot which can be operating in 3 different ways such as line following robot with optional add on gadget and capable of line following, distance measure, and control wirelessly [8]. But in evaluating and analyzing this robot, not all the function will be measured. Because one of the function is using Xbee RF module and it controlled by PIC16F877A, so the author choose to analyze only for this two functions and the major study of this product is base on it microcontroller codification.

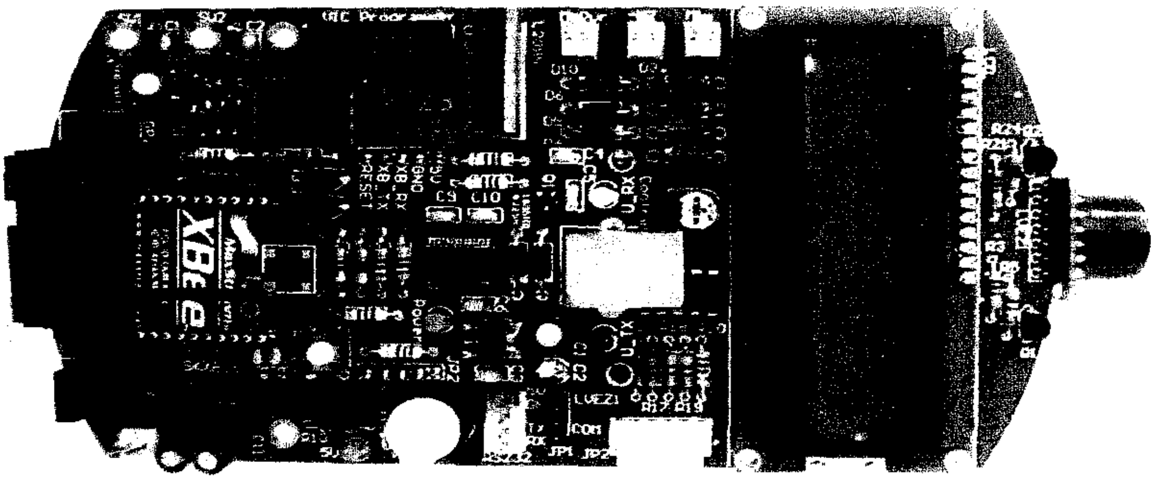


Figure 5.0: XBee RF Module attached to the robot circuit [8]

As shown at the figure 5.0 above, the robot can be control wirelessly by using Xbee RF Module. To control this robot, the keyboard Num pad 8 is used move the robot forward, Num pad 4 for move to the left, Num pad 6 for move to the right and Num pad 2 for moving backward[8]. In order to make this Num pad can control the robot, some data will transmit from the Xbee module on the computer to be received by another Xbee on the Robot and controlled by a PIC microcontroller on the robot. For make it functionally, some codification on the microcontroller have taken place which all the information will be shown on the **Appendix A**.

Basically, the microcontroller need to control all the data which been transmitted from the computer Xbee module, interpret it and give some instruction to the robot by turn on and off the motor inside the robot. For example, if the Num pad 8 been push, the microcontroller will execute this following code:

```
lcd_goto(20);
```

```
          if (RCREG == '8')
// if character '8' is detected, the robot move forward
          {
                forward();
                send_string("FORWARD          ");
          }
```

Then, the microcontroller will find the `forward()` function and execute this following code:

```
     void forward ()
     {
          motor_ra = 0;
          motor_rb = 1;
          motor_la = 0;
          motor_lb = 1;
     }
```

After execute this code, the microcontroller will send a string "FORWARD" to the LCD of the robot to display it until the Num pad 8 being release. For Smart Helmet, the study of these codification can help as a reference in codify the microcontroller code. But before microcontroller can execute all the data transmitted and received. The microcontroller needs to be configuring its input and output pins and also UART setup for XBee RF module in the beginning of the codification.

CHAPTER 3

METHODOLOGY

3.1 METHODOLOGY OF RESEARCH

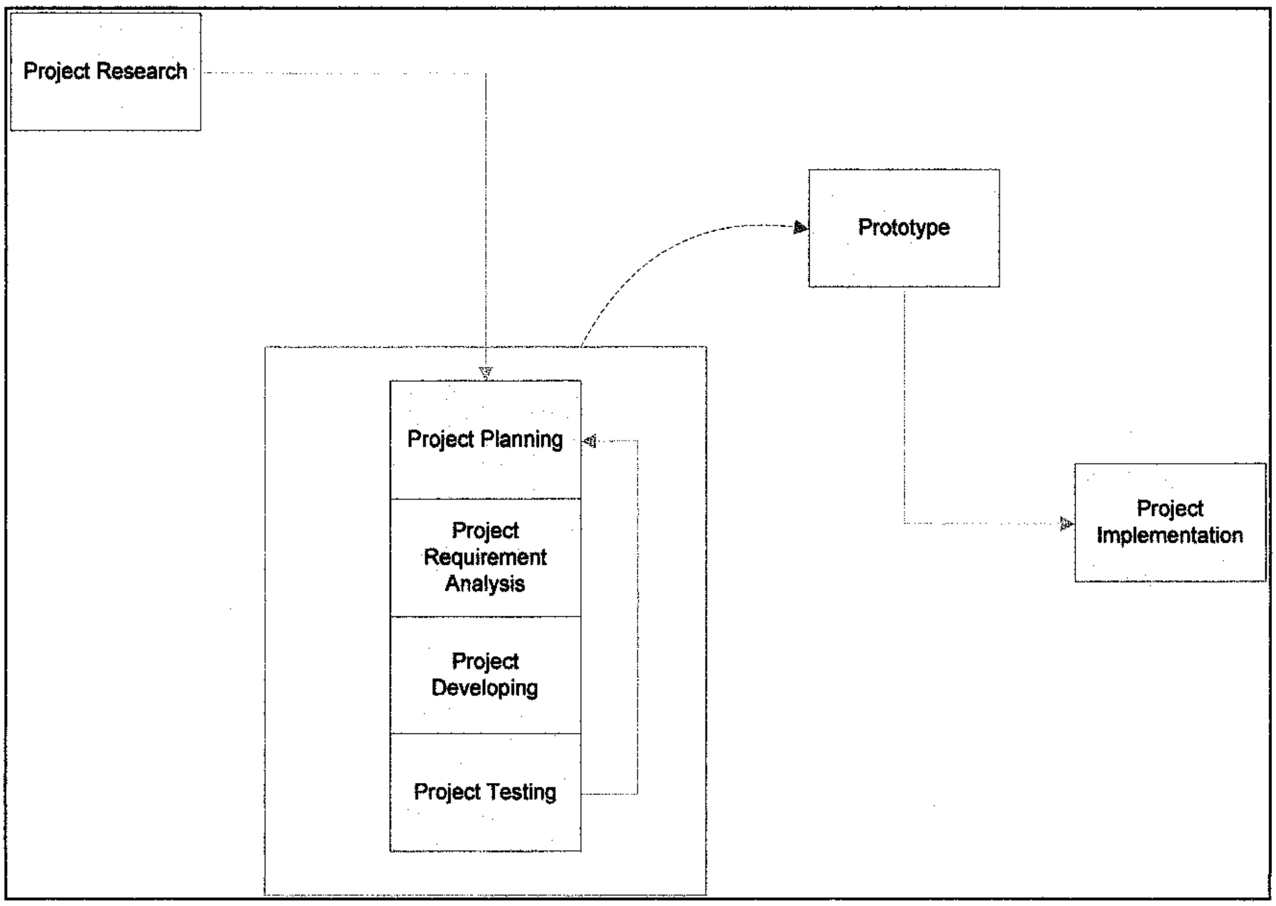


Figure 6.0: The Throwaway Prototyping Model

In developing Smart Helmet Project, a Throwaway prototyping Model of the Project Development Life Cycle is been used. This methodology model is based on four major stage which are research stage, developing stage, prototype stage and the project implementation stage. The graphical view of the throwaway prototyping model which been refered during the development phases of this project is shown in figure 6.0 above.

In general, the development phase of smart helmet project followed by seven sub phases shown in the model. The flow will be start at the research phases, followed by project planning phase, then through the project planning phases and next been followed by project developing and project testing phase. When finish this five phases, a prototype of the project been produced and project implementation been take part.

Specifically, the project life cycle of the Smart Helmet followed the Throwaway Prototyping Model which is explained below:

3.1.1 Information gathering on Connection Technology (Researching)

In gather information on connection technology, the very large library in the world which is internet been used. All the articles and also manuals in applying this technology in Smart Helmet project been studied in this phase (as shown as in the literature review). Some discussion also being made here with the appropriate personal through the web and online discussion and also some interview with the data communication and networking lecturers also happen at this stage. In this stage also, initial research with the RF modules and devices been done for the planning stage.

3.1.2 Identifying proper device for the connection technology (Planning)

The planning stage objective is to make sure the technology compatible with the system to be developed, or in the other word, to identify the devices to be used and the need in designing the blueprint of the entire Smart Helmet development. In order to do that, further discussion been done with the project supervisor time by time. Other than discussion, opinions and idea also has been gathered from they personal who have knowledge in repairing and also modifying motorcycles especially the fabricators for the Smart Helmet blueprint on the motorcycle.

3.1.3 Application programming – research on the suitable language and developing blueprint of Smart Helmet (Analyzing)

For this analyzing stage, after designing the blueprint of the project, the appropriate research and choose of the language will be analyzed. To do so, further discussion with supervisor and research been done on the language available in programming the RF module have been selected. Because the Smart Helmet is using a microcontroller to run the module, the languages that available and been analyzed are C language and Assembly language. Some example of the developed RF module codes also been analyzed and tested here at this stage. All the example codes have been referred and been get from the others RF project which available to download from the internet.

3.1.4 Prototype Development (Developing)

On prototype development, the research must be practically done. Which mean, some of the practical work have been done. Some modification on the RF module and also motorcycles will occur in this developing stage. By referring to the project blueprint, the modification and attachment of the RF module have been done smoothly. This stage also has been done by some help from the experts in both field RF module and motorcycle modification. Codification on the microcontroller also been done in this stage which been refer with the other related example project have been gather before.

3.1.5 Prototyping testing (Testing)

In testing the prototype, it is done by testing the circuit and codes with the appropriate simulation software before attach the device to the helmet and motorcycle. At this stage, MPLAB software and PIC Simulation software been used supervised by supervisor and with help from some experts. As brief explanation, the bad result and outcome have been throwing away here until the final prototype been produced and all the result and outcome of the testing will be deeply discuss later on the result and discussion in chapter four.

3.1.6 Producing End Product (Prototyping)

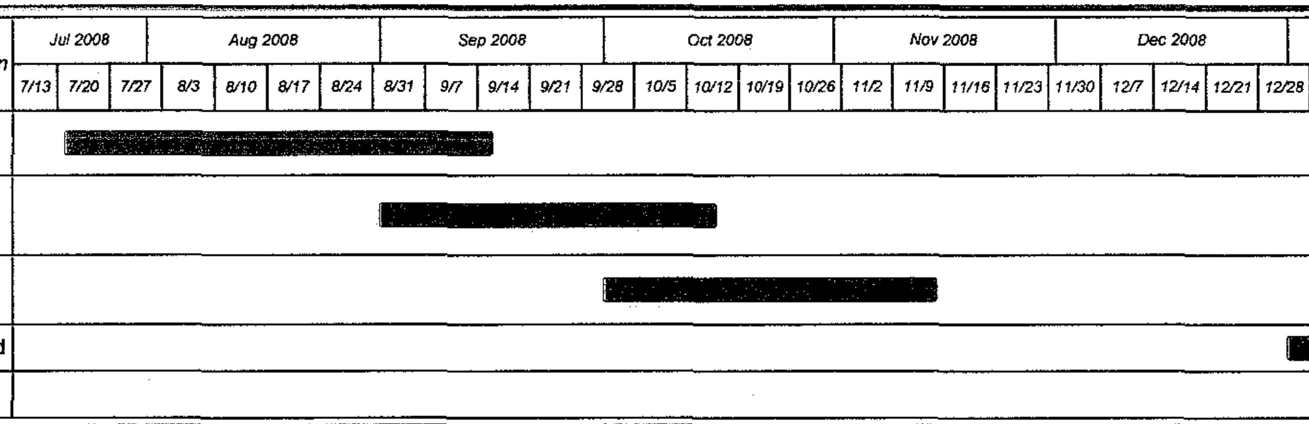
At this stage, the tested prototyping system which been produced before will be attached to the both helmet and motorcycle. In this stage also, the suitable hardware modification have been done to the both and being tested again and again until the final product being produced. The test include the appropriate places that possible for the circuit being placed in helmet based on the project blueprint which been developed in the planning stage before and also same goes to the motorcycle. Outcome of this stage is the final product that will be used for the next stage (implementation).

3.1.7 Smart Helmet Implementation on real-life (Project implementation)

For this implementation stage, the final product which is attached RF module Helmet and motorcycle tested in real-life helmet wearing. The test sets will be produced which are testing the product without wearing helmet, testing the product with wearing but not clipped helmet and also testing product with proper wearing helmet. These tests been made to make sure that this ended product can be implemented in daily used and also applicable to be consume and implemented by the motorcyclist at the first place.

3.2 GANTT CHART OF THE SMART HELMET DEVELOPMENT

ID	Task Name	Start	Finish	Duration
1	Information gathering on the study (Researching)	7/21/2008	9/15/2008	8w 1d
2	Identifying proper device for the bluetooth technology (Planning)	9/1/2008	10/15/2008	6w 3d
3	Application programming -- research on the suitable language (Analyzing)	10/1/2008	11/14/2008	6w 3d
4	Prototype Development (Developing)	1/1/2009	3/13/2009	10w 2d
5	Prototyping testing (implementation)	3/10/2009	6/1/2009	12w



ID	Task Name	Start	Finish	Duration
1	Prototype Development (Developing)	1/1/2009	3/13/2009	10w 2d
2	Prototyping testing (implementation)	3/10/2009	6/1/2009	12w

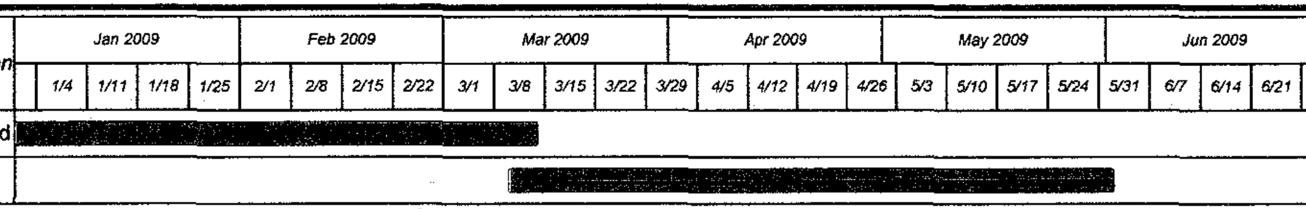


Figure 7.0: Smart Helmet Project Gantt chart.

3.3 TOOLS AND EQUIPMENT REQUIRED

3.3.1 Helmet and Motorcycle

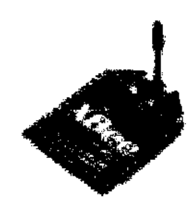
Base of this project, these two equipments are the basic equipment that must be prepare. The helmet will be connecting to the motorcycle by using connection device (XBee).

3.3.2 Xbee Starter Kit (master & slave)

By using Xbee technology, this device is use to connect both motorcycle and helmet as already mention in chapter 2. Some specification detail provided by supplier as shown in table 5.0 below:

XBee and XBee Pro

Description: Finally, user may enjoy another simple yet reliable wireless communication for wireless control and monitoring. XBee OEM RF module has been used in many robotics applications world wide to offer wireless communication, point to point and also mesh network. Save the time searching for surrounding device and request for connection, it can send data wireless after powering up without any extra configuration. Besides, XBee module can be used as standalone wireless transceiver for control and monitoring.



: Circuit power and busy indicator LED
: ISM 2.4 GHz operating frequency
: Industrial temperature rating (-40° C to 85° C)
: Interface data rate: Up to 115.2 Kbps
: Bidirectional wireless communication (transmit and receive)
: Operating frequency: 2.4 GHz
: Supply voltage: 2.8 - 3.4 V
: Power-down sleep current
: <10 uA : Wire Antenna

Specification	XBEE	XBEE-PRO
Communication Range	up to 300 ft (100m)	up to 1 mile (1500n
Indoor / Urban Range	up to 100 ft (30 m)	up to 300 ft (100 n
Transmit current	45mA (@3.3V)	270 mA (@ 3.3 V
Receive current	50 mA (@ 3.3 V)	55 mA (@ 3.3 V)
Power Output	1 mW (0 dBm)	60 mW (+18 dBm
Dimension	2.438cm x 2.761cm	2.438 cm x 3.294 c

Table 5.0: Xbee and Xbee Pro details

3.3.3 Wires

Wires used in order to make the circuit and modification on the device that being used in this project.

3.3.4 Analog Temperature Sensor

To be function as second switch at helmet and this will detect human body heat (approximately $\pm$ 33'C). Then, as switch, it will make the current of the circuit flow to on the XBee device at helmet.

3.3.5 SK40 Microcontroller Programmable Device.

It is device to program the PIC Microcontroller in order to make the Xbee device being used and functioning well. The SK40 device also can be used as a part of the circuit board as well without separated the microcontroller from its main circuit. Some specification detail provided by supplier as shown in table 6.0 below:

Description: SK40B is designed to offer an easy-to-start platform for PIC MCU user. This board comes with basic components for the user to begin the project development. User is able to utilize the function of PIC by directly plugging in the components to the SK40B in whatever way that is convenient. With the on-board UIC00A connector, user can load the program faster and easier using the UIC00A programmer. The PIC MCU is not included in this kit to provide the freedom for the user to choose PIC MCU.	
: Alternative power supply using AC-DC adapter : Compact, powerful, flexible and robust start-up platform : Connector for UIC00A - simple and fast method to load program : No extra components is required for the PIC to function : All 33 I/O pins are clearly labeled : RS-232 port allows serial communication with the computer : Support bootloader for program loading : No more frustration to plug the PIC in and out for reprogramming : Perfect fit for 40-pin PIC16F and PIC18F family PIC MCU : 20MHz crystal	

Table 6.0: Enhances 40 pins PIC Start-Up kit details

3.3.6 PR11-Temperature detector

This board is a built in board which consist of thermostat, 40pins microcontroller, output devices and some resistors. This board is programmable and also eases to be programmed. Some modification is needed. Some specification detail provided by supplier as shown in table 7.0 below:

Temperature Control System Using LM35

Description: This project utilizes the PIC16F876A to read the temperature feedback from the LM35, display it on the 16x2 Character LCD and control 2 brushless fans. It illustrates the process to configure the ADC on PIC to measure the analog voltage from the temperature sensor. It can be further modified into a temperature monitoring system.

: Other necessary components
: PR11 PCB : PIC16F876A
: LM35DZ Temperature Sensor
: 16x2 Character LCD
: Buzzer
: 12V DC Brushless Cooling Fan

1. UIC00A Programmer
2. AC-DC Adaptor TMC-500PM

Table 7.0: Temperature Control System details

CHAPTER 4

RESULT AND DISSCUSSION

As the overview of this chapter, this particular chapter will discuss more deeply on the developing stage of the Smart Helmet. It will include the main flow of the Smart Helmet System, the blueprint design or frame work for the Smart Helmet, the out come of the testing, the suggested place to attach the circuit module on both helmet and motorcycle, and other matter or advancement on the Smart Helmet Project.

In the short and simple word or description, this chapter will touch on the Smart Helmet project from the planning phase of the methodology model till the implementation phase of that model for this project.

4.1 MAIN FLOW OF SMART HELMET

In this sub topic, the main flow of the Smart Helmet will be shown (Figure 8.0) which already developed in the Planning Stage of the project cycle Methodology in the chapter 3. Basically, the main idea or flow of the Smart Helmet will be start with the user or motorcyclist put their head inside the helmet. When it happens, the microcontroller on the helmet will execute the thermometer circuit on the helmet to get the human temperature data. When the data interpreted as human temperature, the microcontroller will get some input from the helmet clip circuit whether the user clipped their helmet or not. If it is yes, then the microcontroller will turn off the thermometer circuit and turn on the Xbee RF module which happen after confirming that user already wear their helmet properly. When the Xbee RF module turned on, the data will be transmitted to the other Xbee module which attached to the motorcycle and microcontroller there will execute the connection between these two Xbee modules and allow the user to start up their motorcycle. For information, the Xbee module at motorcycle will turned on when the user switch on their motorcycle key switch. All the flow can be simplify as being shown in the figure 8.0 below:

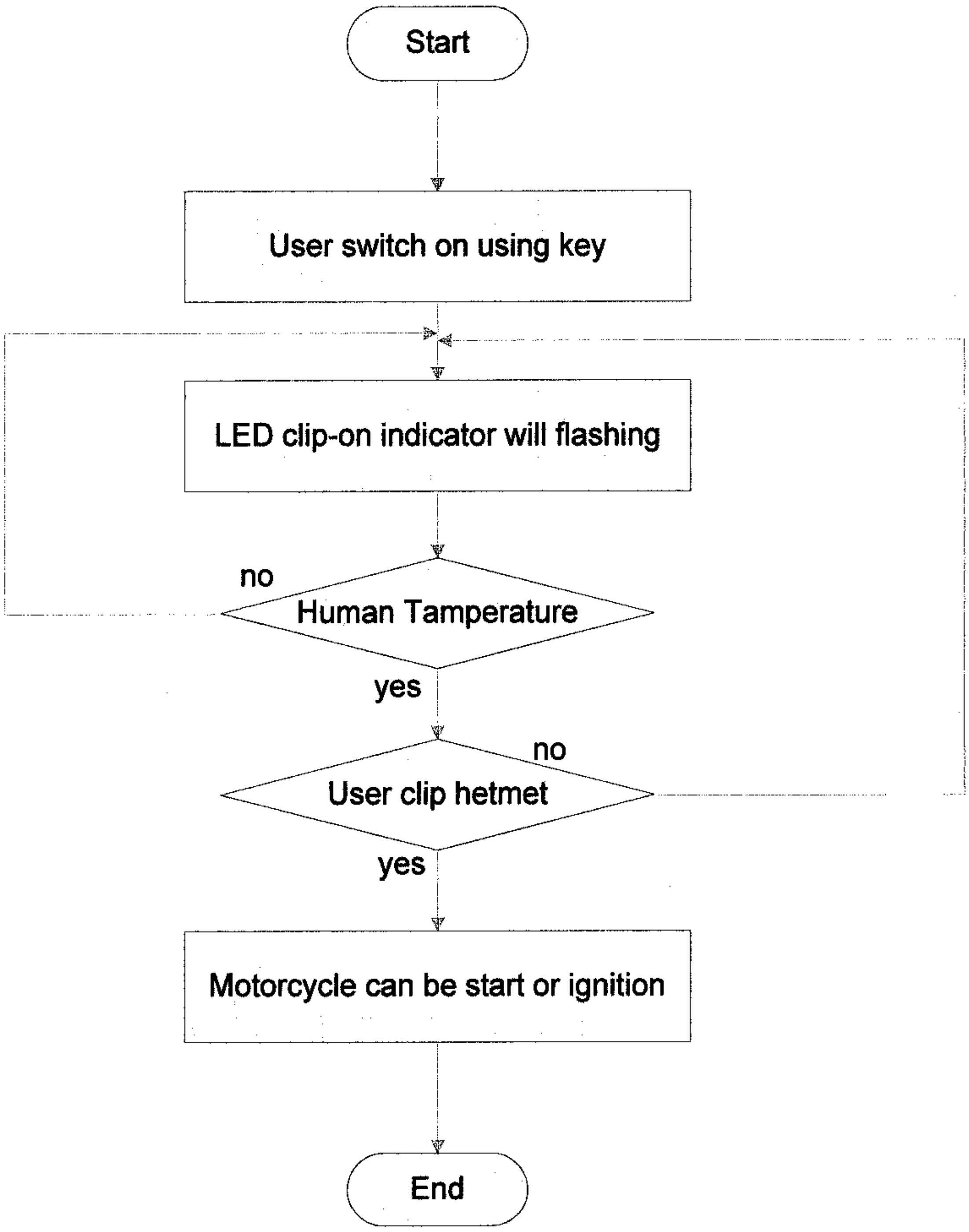


Figure 8.0: Main Flow of Smart Helmet.

4.2 FRAME WORK OF SMART HELMET (HELMET CIRCUIT)

Base on the main flow of the Smart Helmet project, the frame work of the circuit can be simplified and shown as below (Figure 9.0):

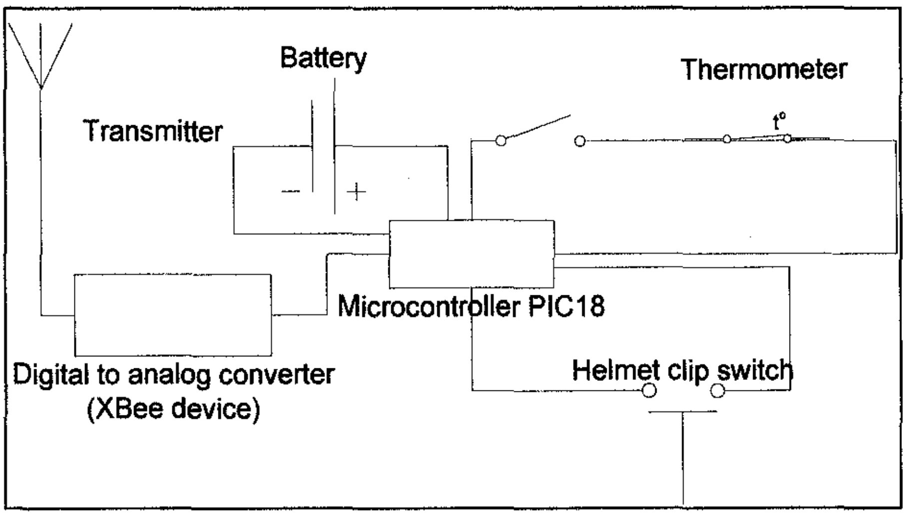


Figure 9.0: Frame work of the electronic circuit

And it can be shown in microcontroller circuit design as shown in figure 10.0 below:

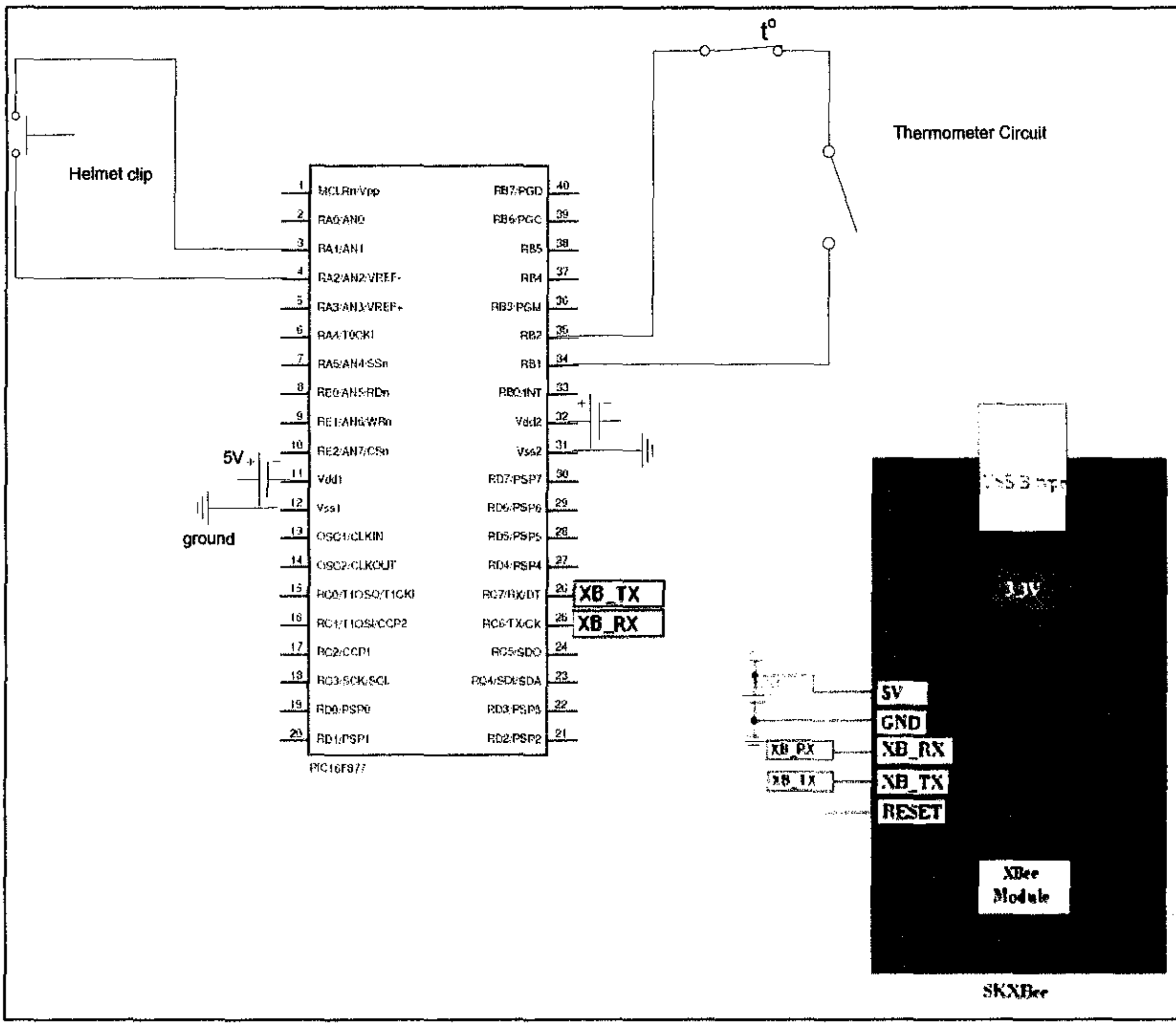


Figure 10.0: Microcontroller circuit design

Because the microcontroller that has been used is PIC16F877A, the figure 10.0 above shows how the thermometer circuit, helmet clip circuit and Xbee module circuit being connected to the microcontroller. For instance, PORTRB1 and PORTRB2 will be execute as the input and output pin for the thermometer circuit with the microcontroller which will allow microcontroller to send and receive the data from the temperature sensor (analog sensor) and run next function. When the helmet clip circuit turn on, thermometer circuit will be turned off and turn on the Xbee module by the microcontroller which connected to PORTRC6 and PORTRC7 as transmitter and receiver pin for microcontroller. Here, microcontrollers will also being executed as Analog to Digital Converter for both thermometer and Xbee module circuit.

4.3 FRAME WORK OF SMART HELMET (MOTORCYCLE CIRCUIT)

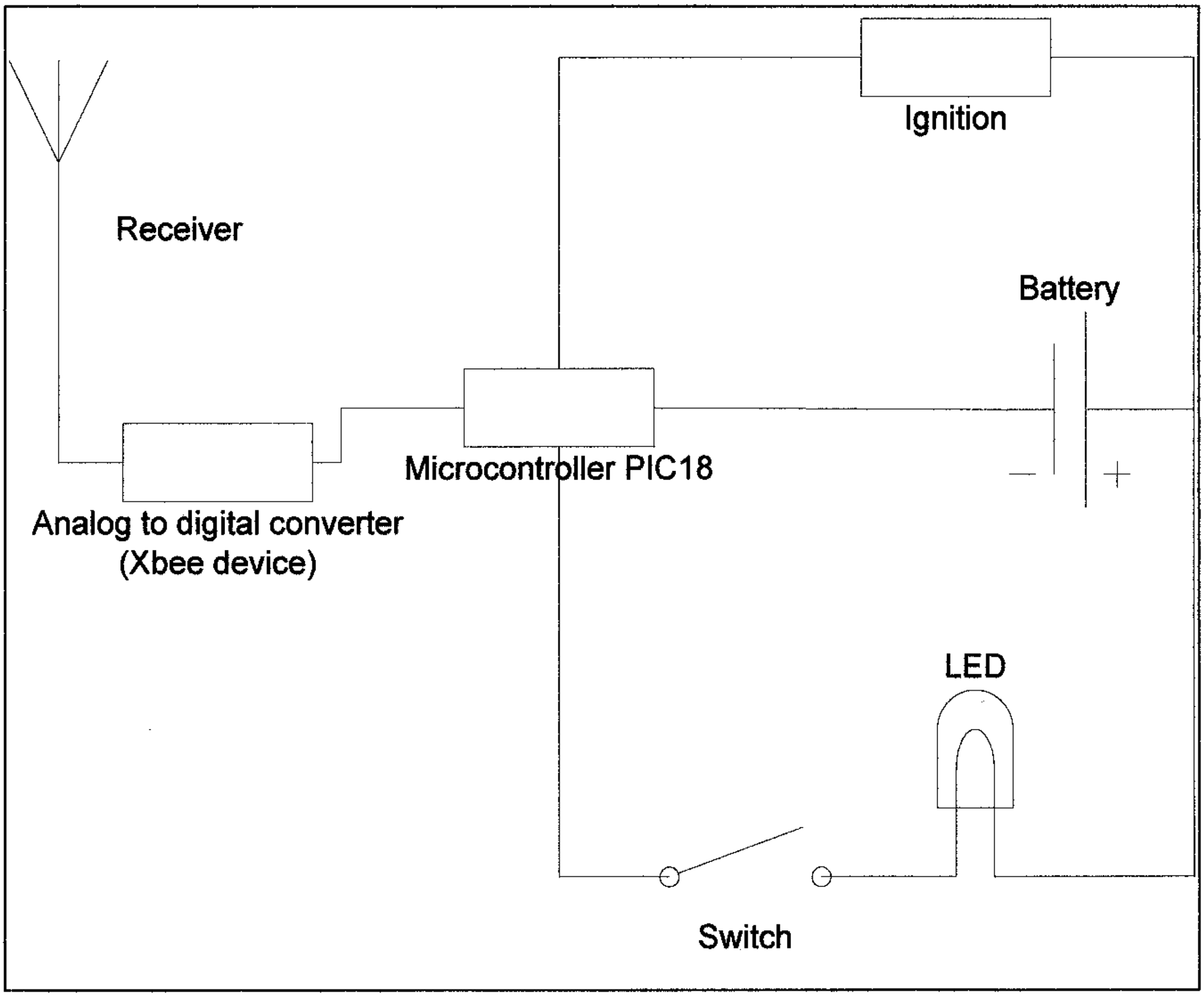


Figure 11.0: Frame work of Motorcycle circuit

4.4 OUTCOME FROM THE TESTING

Testing phase is the phase when the devices being test to find some problem and also to solving that particular problem. As have been mention in project cycle methodology, this Smart Helmet project will go through several test before come out with the prototype and also being implement in real-life environment. For the first test, is about testing the Xbee module by using 2 computers. The objective of this test is to make sure the module can be connected and the unique address for both modules can be generated.

In this test, the software and tools required are 2 unit computers which can be in Microsoft Window XP or Vista operating system, 2 unit of SKXBee RF module [9], and also the X-CTU software which already installed in both computers. First of all, both SKXBee RF modules must be connected to both computer (1 unit SKXBee module for each computer) as been shown in figure12.0. Then, run the X-CTU software and detect the module attach for each computer.

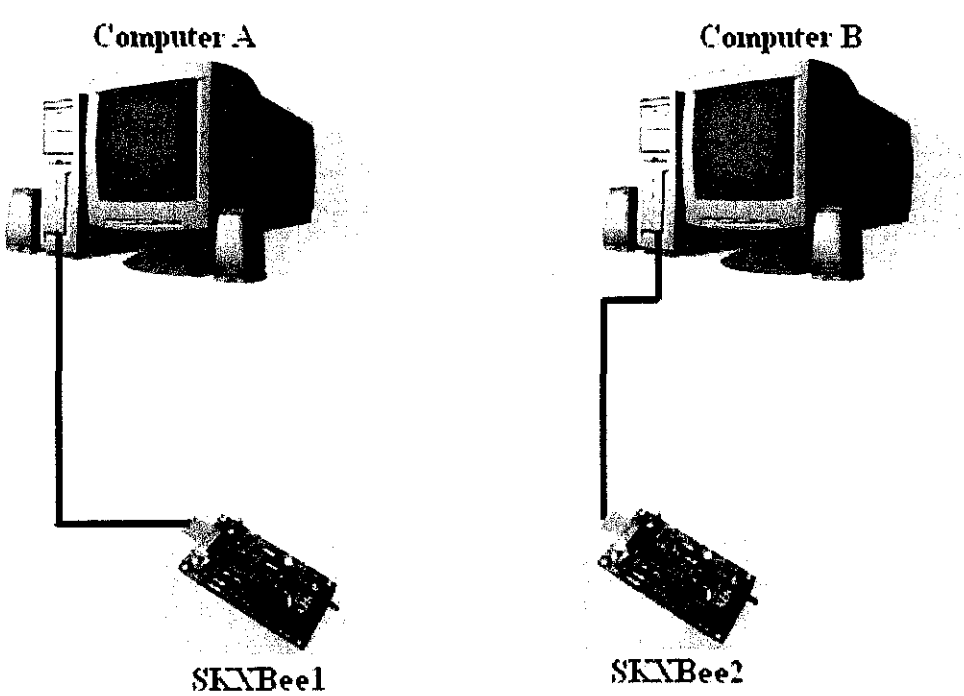


Figure 12.0: Tools required for testing

After detecting both modules for each computer by using X-CTU software, the output will be shown as in the figure 13.0. Then, it is time to configure the address for both Xbee and tried to make connect between these two modules. For set up the ordinary address, the AT commands can be used in the terminal tab on the X-CTU software. For this test, SKXBee 1 will be addressed as 1111 and SKXBee2 as 2222. But, before implement the command, the baud rate for both modules must be set same as each other in detecting module stage. All of these processes will be graphically explain in figure14.0 below:

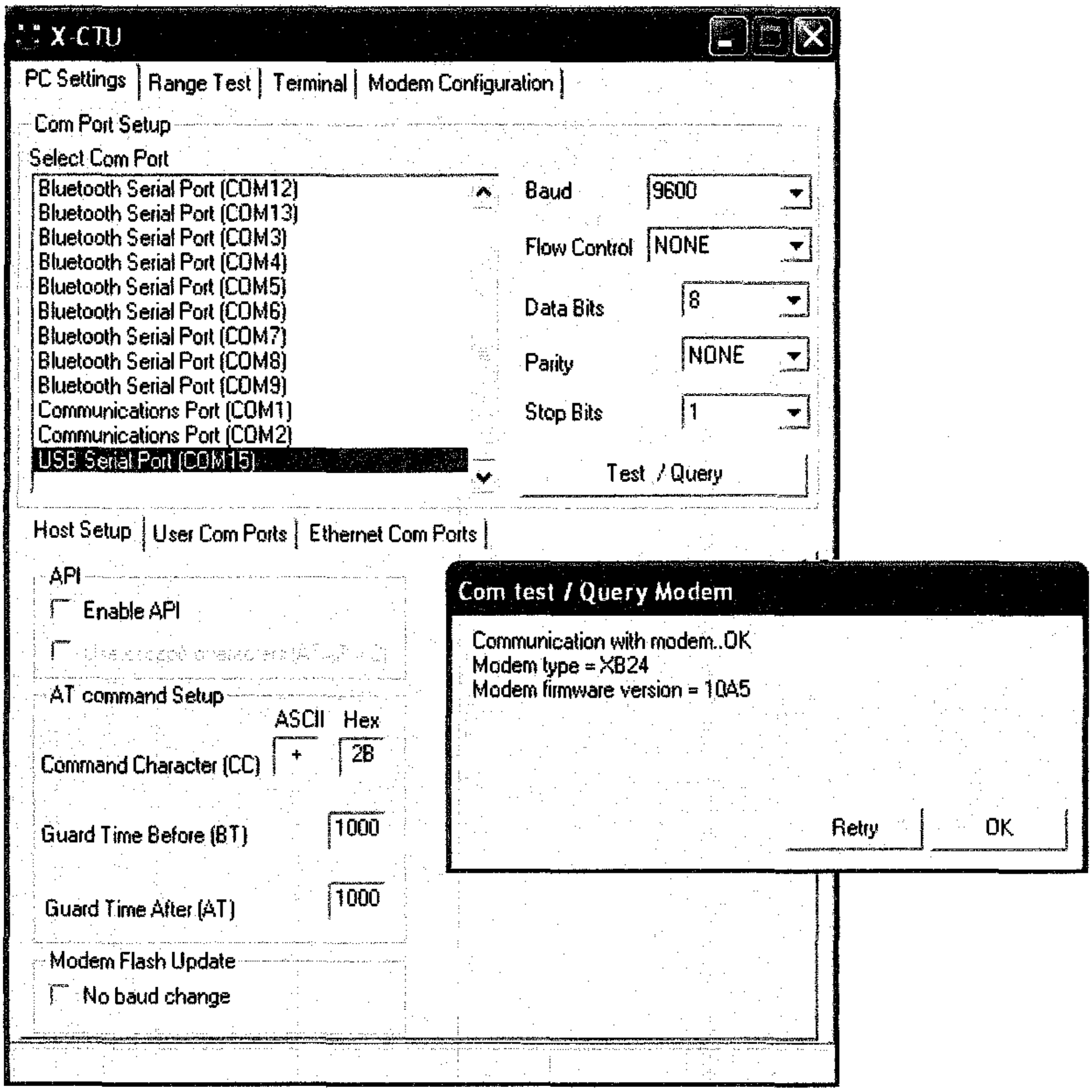


Figure 13.0: Print screen on detecting Xbee module using X-CTU software

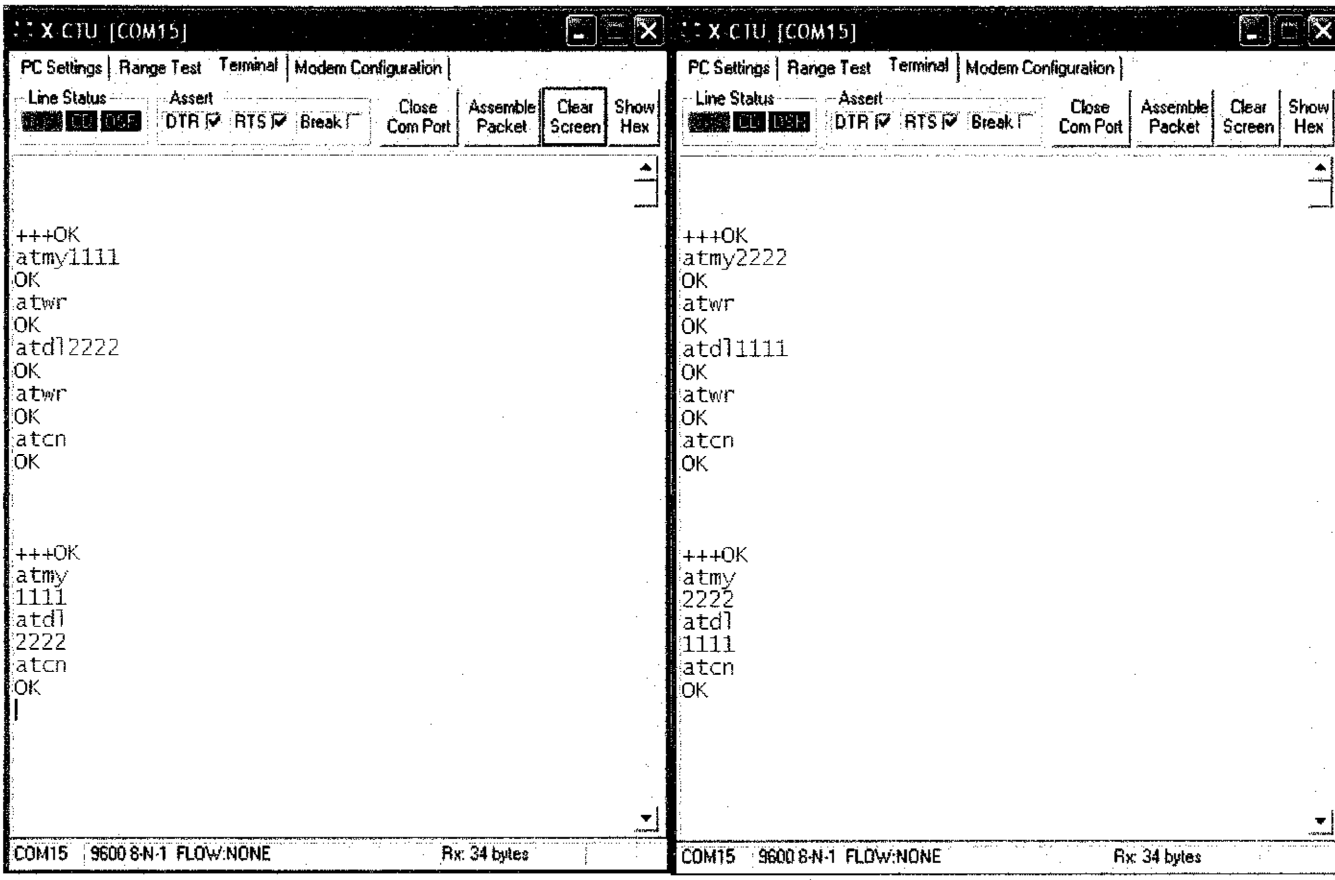


Figure 14.0: Print screen of writing the address for Xbee modules using X_CTU software

Note that, the red OK word been generated automatically to make sure the command is ok and no error occur when execute the command. After this stage, both of the modules, already connected automatically after the address configured. As can be see in figure 14.0 above, the current address (`atmy`) for SKXBee 1 is 1111 and its destination low (`atdl`) address is 2222 and it is vice versa with the SKXBee 2.

Next, the output from this testing phase will be shown on the figure15.0 below:

SKXBee 1 SKXBee 2

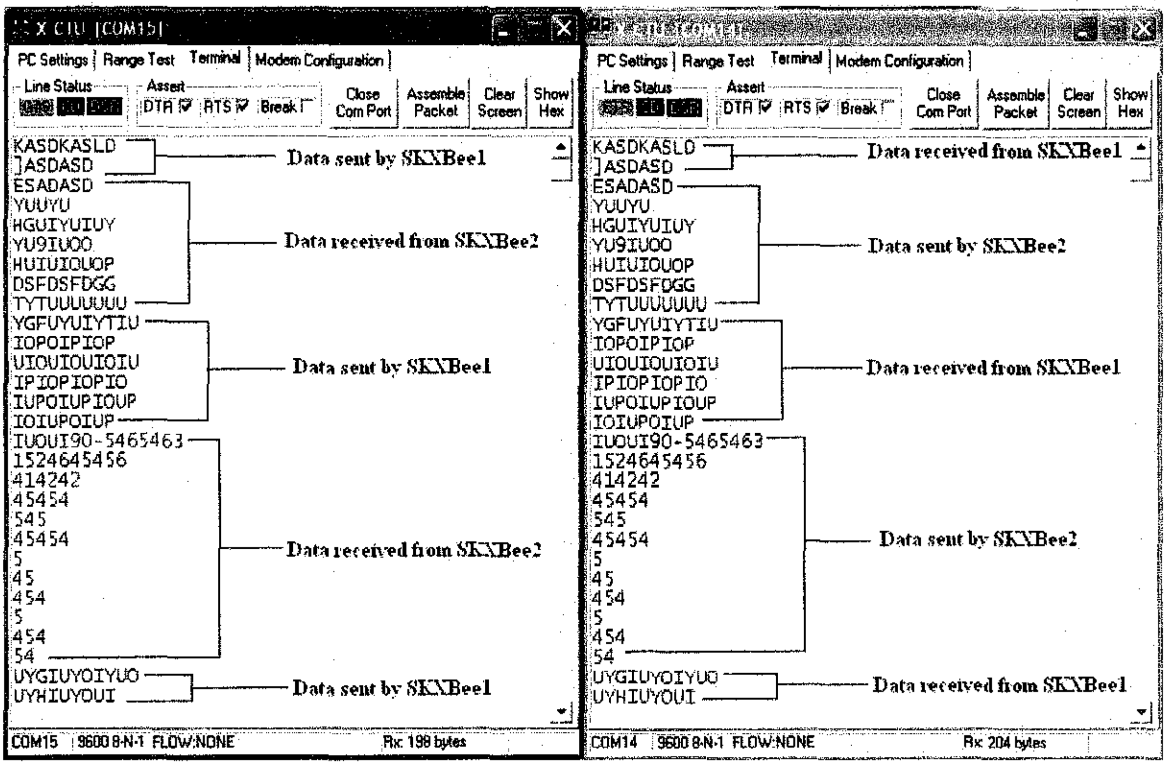


Figure 15.0: output of the connection testing phase

4.5 CONTINUOUS CONNECTION TEST

Another testing that occurs here is testing the continuous connection among these 2 Xbee RF modules. The objective for this testing is to make sure that this connection can last longer for several hours. When the connection established for several hour, it means that when the module implemented in to the helmet and motorcycle, the end user can travel used this smart helmet for a long journey which take several hours to reach the destination. The devices requirement for this testing is same as the testing that have been tested before and the connection setting also same as the previous test. The only thing that different here for this test is the output of the test which will show as the figure 16.0 to 18.0 below:

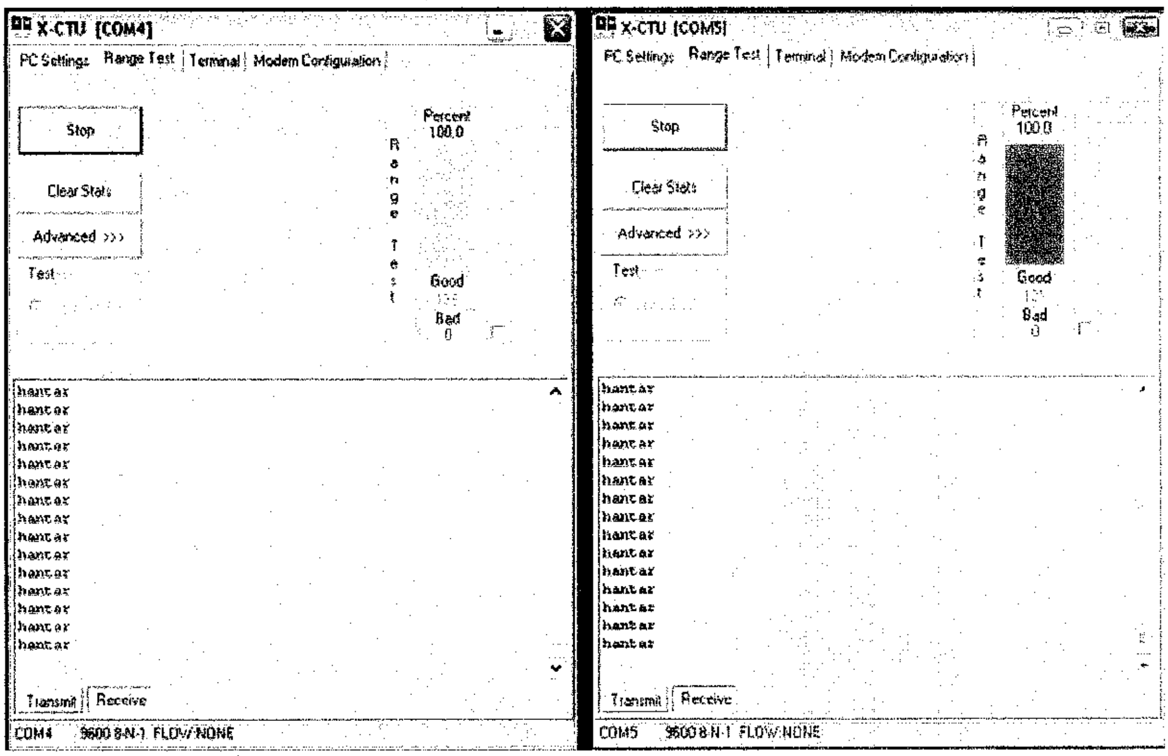


Figure 16.0: output of the early stage of the continuously connection test (11:39pm)

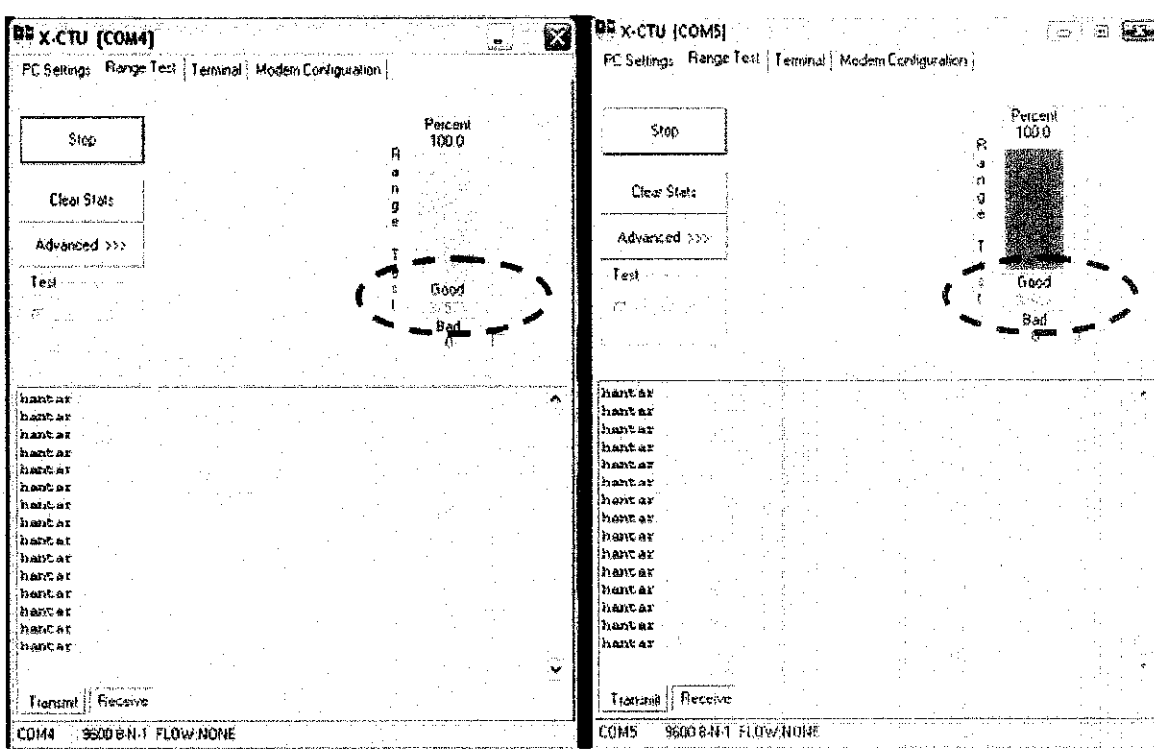


Figure 17.0: output after 49 minutes running the connection (12:28am)

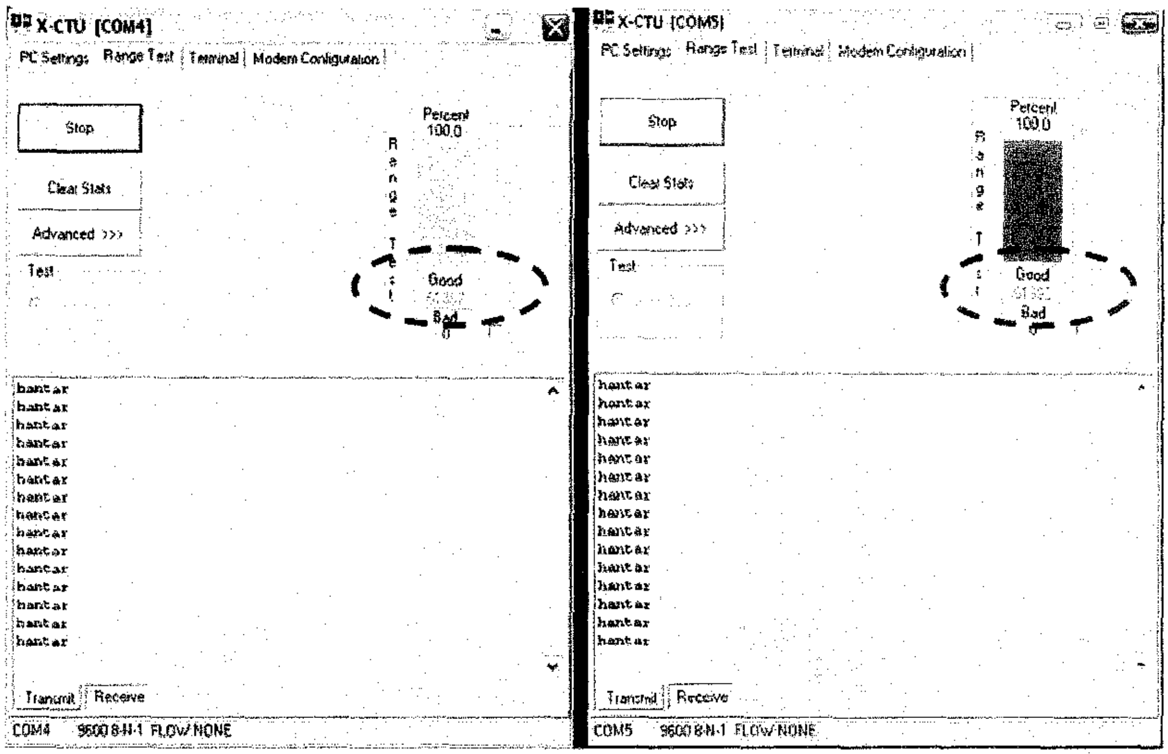


Figure 18.0: output after 1 hour 30 minutes of the testing (12:59am)

From these 3 figures above, the connection already established over than 1 hour 30 minutes and the signal strength base on package transmitted and received still good after over than 61000 packages received and transmitted between this 2 Xbee modules. So, the result supports the objective of this continuous connection test before the test take place.

4.6 DEVICES ATTACHED ON HELMET

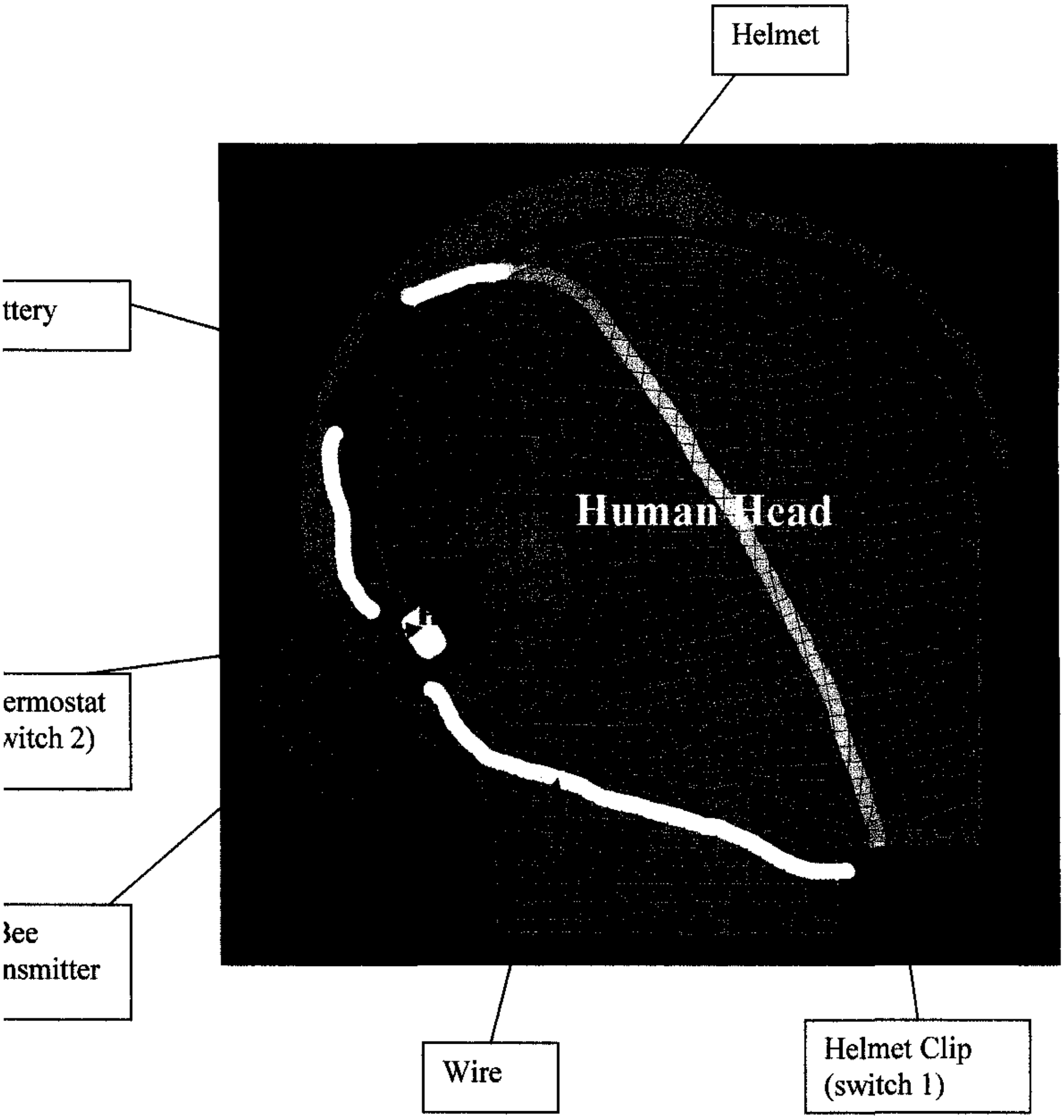


Figure 19.0: Suggested device attached to the helmet

4.6.1 Description and procedure:

Base on figure 19.0 above, the entire circuit will be turn on when helmet clip tight and clipped properly and also the thermostat detecting human heat from the rider head which basically between 33'C to 37'C. When both switches turn on, the Xbee device or transmitter which powered by battery will turn on too and transmit signal from helmet to motorcycle receiver as the main switch for the motorcycle's circuit and ignition.

4.7 DEVICES ATTACHED ON MOTORCYCLE

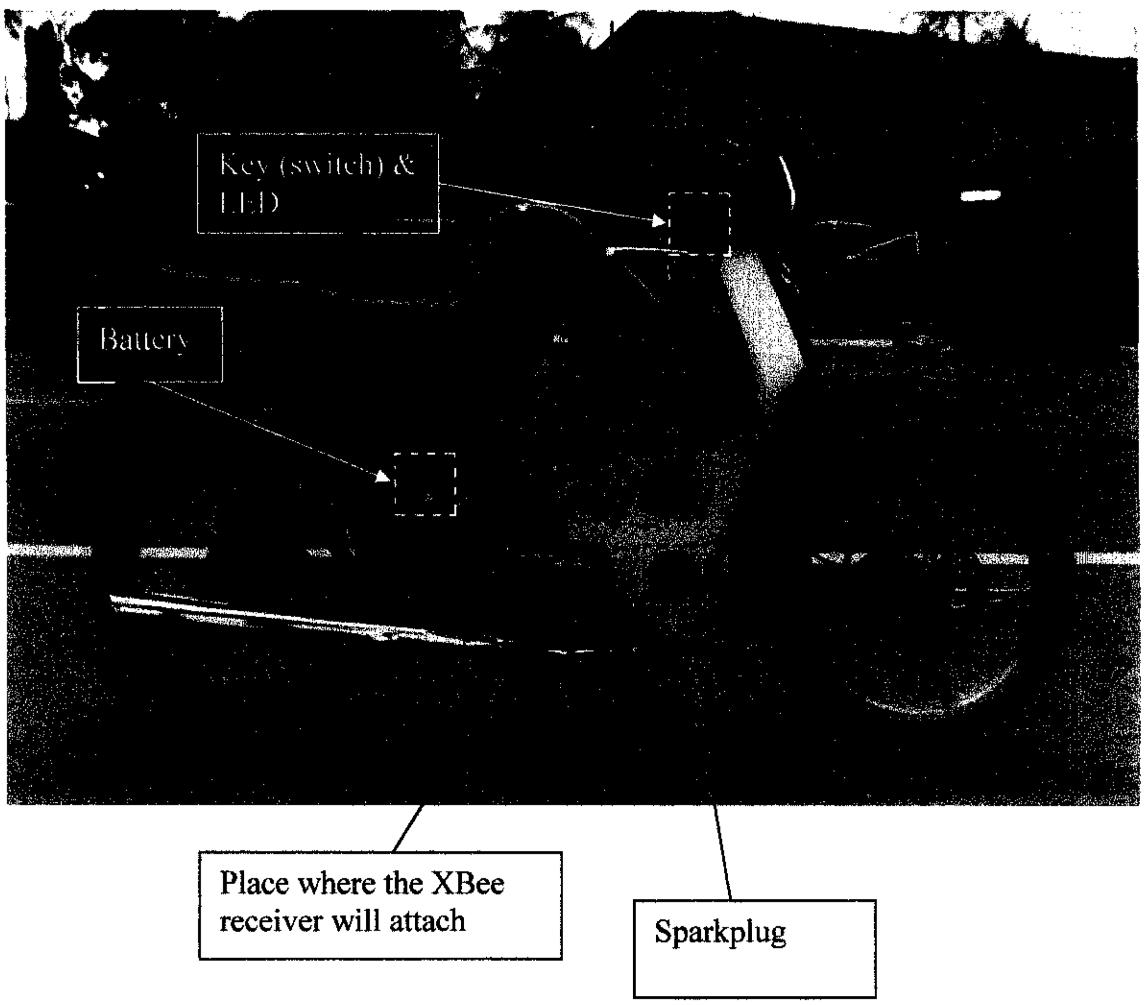


Figure 20.0: Device attached to the motorcycle

4.7.1 Description and procedure

Base on the motorcycle picture (figure20.0) above, the spotted place is where the equipment and device located and will be located. When the key is on, at sparkplug, there is wire from battery to power the spark plug to sparking a small fire spark to burn the gasoline at the block of the motorcycle and then power up the motorcycle. But, when the project being install, if the key is on but the rider not clip the helmet (no signal), the circuit still close but the current flow only go to the LED circuit, turn on the LED but not allow the rider to start the motorcycle (no spark at sparkplug because the sparkplug circuit still open and not be complete by the signal which come from transmitter).

In other side, when the helmet clip closes properly, the transmitter will transmit Xbee signal from the helmet to the motorcycle receiver. When there is a signal, automatically the LED circuit will turn off (done by three-state) and turn on the ignition circuit which allow sparkplug to produce fire spark in order to start motorcycle.

CHAPTER 5

CONCLUSION

XBee technology is a short-range wireless specification aimed at simplifying communications among Internet devices and between devices and the final user. In conclusion it can be said that XBee refers not only to a technology but also to a standard and a specification since it presents freedom of mobility. As being the engineers of the future, we are responsible for not only improving such technologies, but also spread them around the world by using them in our daily products. And for this product, Smart Helmet project is possible to be implemented and also being improve in future because it is very useful to increase protection on the road for motorcyclist, decrease fatal motorcycle road accident, prevention for motorcycle stealing crime and also can utilize the value and benefit of the XBee technology and other newcomer technologies.

5.1 RECOMMENDATION

There are many other side problems if this project implemented in future such as related with the rigidity of the project. Rigidity will occur when rider want to go to shop which situated 50m from their house and riding their motorcycle within their housing area which police will not summon them. For this future problem, the best solution is to have some equipment that can limit their distance traveled without wearing helmet. The basic idea is, the system can include some device which can limit their motorcycle traveling more than 100m radius. The device can be attaching at motorcycle tire and be manipulating by an application base on motion of motorcycle wheel. The formula will be like this:

For detecting 100m distance or limitation to the motorcycle,

Limitation	= limitation distance / circumference of tire	= 100m / 1.5m
	= times of wheel rotating	= 66.67 x

Base on the theory, the limitation will base on the rotation of wheel. Example, if the circumference is 1.5m, its mean 100m divide by 1.5m equal to 66.67 times or slightly to 67 times wheel rotated. When it reach to that number, the motorcycle engine will be program to stop function and stop the motorcycle until the rider wearing helmet properly again. So base on this calculation, Smart Helmet can limit the distance of traveling without helmet.

CHAPTER 6
REFERENCES

[1] PanduCermat Website. http://www.panducermat.org.my/fasten_helmet.html

[2] Maxstream Inc. : (2006) **"XBee™/XBee-PRO™ OEM RF Modules",** Xbee manual from manufacturer.
http://www.cytron.com.my/datasheet/WirelessDevice/manual_xb_oemrfmodules_802.15.4.pdf

[3] S Kulamthayan, Radin Umar, Ahmad Hariza,Mohd Nasir,S Harwan : (March 2000) **"Compliance of Proper Safety Helmet Usage in Motorcyclist"**
http://www.miros.gov.my/publications/ComplianceOfProperSafetyHelmetUsageInMotorcyclists.pdf

[4] Jabatan Keselamatan Jalan Raya Malaysia Website.
http://www.jkjr.gov.my/statistics.html

[5] Rob Faludi: (2006) "XBee and More", Class presentation.

[6] R.S Radin Umar: (2006) "Motorcycle safety programmes in Malaysia: how effective are they?" International Journal of Injury Control and Safety Promotion, Vol. 13, No. 2.

[7] Tim Wilmsmhurst: (2007) **"The PIC Microcontroller Family",** Newnes Know It All Series, Chapter 1.
www.newnespress.com

[8] Cytron Technology Website: www.cytron.com.my

[9] Cytron Technology : (2008) "XBee Starter Kit – SKXBee" , Xbee RF Module manual. www.cyctron.com.my

APPENDIX A – Example codification of Multifunctional Mobile Robot

```
//==========================================================================
//    Author                          :CYTRON
//    Project                         :PR23
//    Project description        :Simple line follow
//  Version                      :v1.4
//==========================================================================

//    include
//==========================================================================
#include <pic.h>

//    configuration
//==========================================================================
__CONFIG ( 0x3F32 );

//    define
//==========================================================================
#define sw1               RE0
#define sw2               RE1
#define motor_ra   RC0
#define motor_rb   RC3
#define motor_la   RC4
#define motor_lb   RC5

#define s_left            RB0
#define s_mleft           RB1
#define s_mright   RB2
#define s_right           RB3

#define buzzer            RE2

#define     rs                  RB7
#define     e                   RB6
#define     lcd_data      PORTD
#define b_light     RB5
#define SPEEDL            CCPR1L
#define SPEEDR            CCPR2L

#define     CHANNEL0      0b10000001
#define     CHANNEL1      0b10001001
#define RX_PIN            RC7
#define TX_PIN            RA2
#define BOT_ADD           100


//    global variable
//==========================================================================

unsigned char data[5] = {0};

unsigned int result;
unsigned int To=0,T=0,TH=0;
unsigned char REC;
```

```
unsigned char i=0,raw;

unsigned int us_value (unsigned char mode);



//    function prototype
//===========================================================================
void init(void);
void delay(unsigned long data);
void send_config(unsigned char data);
void send_char(unsigned char data);
void e_pulse(void);
void lcd_goto(unsigned char data);
void lcd_clr(void);
void send_string(const char *s);
void dis_num(unsigned long data);

void line_follow(void);
void ultrasonic(void);

//Xbee function prototype
void wireless_xbee (void);

void analog_sen(void);
void read_adc(char config);

void forward(void);
void stop (void);
void backward (void);
void reverse (void);
void left(void);
void right(void);

//    interrupt prototype
//===========================================================================
static void interrupt isr(void)
{
      if (TMR0IF)
      {
            TMR0IF = 0;
            To +=0x100;
      }

      if(RBIF)
      {
            RBIF = 0;
            if (RB4)
            {
                  TMR0 = 0;
                  To = 0;
            }

            else TH = TMR0 + To;

      if(RCIF)
      {
```

```
            RCIF = 0;
            if (RCREG == 'R') data[i=0]= RCREG;
            else if (RCREG == 100) data[i=0]= RCREG;
            if ((data[0] == 'R'))data [i++] = RCREG;
            if (i>4) i = 4;
      }
}


//    main function
//==================================================================
void main(void)
{

      unsigned char m =0 ,i =0;
      buzzer = 1;
      init();
      delay(20000);

      lcd_clr();
      send_string("Select mode");
      buzzer = 0;
      while(1)
      {
            if( !sw1)
            {
                  while(!sw1);
                  m++;
                  if ( m > 3) m = 0;
            }

            if (!sw2)
            {
                  while(!sw2);
                        switch(m)
                        {

                        case 0 :
                                          line_follow();
                                          break;
                        case 1 :
                                          ultrasonic();
                                          break;
                        case 2 :
                                          analog_sen();
                                          break;
                        case 3 :
                                          wireless_xbee();
      // mode 4 : wireless xbee mode
                                          break;
                        default :    ;
                  }
            }

                        switch(m)
            {
```

```
                    case 0 :    lcd_goto(20);

                          send_string("1.LINE FOLLOW    ");
                                break;
                    case 1 :    lcd_goto(20);

                          send_string("2.Ultrasonic     ");
                                break;
                    case 2 :    lcd_goto(20);

                          send_string("3.Analog Sensor ");
                                break;
                    case 3 :    lcd_goto(20);

                          send_string("4.SKXBEE         ");
                                break;
                    default :   ;

            }
      }
}

// Initailization
// Description : Initialize the microcontroller
//==================================================================
void init()
{
      // ADC configuration
      ADCON1 = 0b10000100;
      RBIE = 1;
      PR2 = 255;
      T2CON =     0b00000100;
      CCP1CON =   0b00001100;
PWM   ( for more detail refer datasheet section 'capture/compare/pwm')
      CCP2CON =   0b00001100;
PWM


      TRISA = 0b00000011;
      TRISB = 0b00011111;
      TRISC = 0b10000000;
      TRISD = 0b00000000;
      TRISE = 0b00000011;


      T0CS = 0;
      PSA = 0;
      PS2 = 1;
      PS1 = 1;
      PS0 = 1;
      TMR0IE = 1;
      TMR0 = 0;

      //setup UART
      SPBRG = 0x81;                 //set baud rate to 9600 for 20Mhz
      BRGH = 1;                     //baud rate high speed option
      TXEN = 1;                     //enable transmission
```

```
	TX9 = 0;
	CREN = 1;			//enable reception
	SPEN = 1;			//enable serial port
	RX9 = 0;
	RCIE = 1;			//enable interrupt on eachdata
received
// enable all unmasked interrupt
	GIE = 1;
	PEIE = 1;

	// LCD configuration
	send_config(0b00000001);		//clear display at lcd
	send_config(0b00000010);		//Lcd Return to home
	send_config(0b00000110);		//entry mode-cursor increase
1
	send_config(0b00001100);		//diplay on, cursor off and
cursor blink off
	send_config(0b00111000);		//function

	TX_PIN = 1;
	b_light = 0;
	buzzer = 0;
	stop();

}


// Mode subroutine
//===================================================================

// Mode 1 : line follow subroutine
// Description: Program for the robot to follow line
//===================================================================
void line_follow()
{
	unsigned char memory;

	lcd_clr();
	send_string("Position");

	while(1)
	{
		if ((s_left==1)&&(s_mleft==0)&&(s_mright==0)&&(s_right==0))
				// if only sensor left detected black line
		{	forward();

			SPEEDL = 0;

			SPEEDR = 255;

			memory = PORTB&0b00001111;

			lcd_goto(20);

			send_string ("right  ");

		}
```

```
            else if
((s_left==1)&&(s_mleft==1)&&(s_mright==0)&&(s_right==0))
            {       forward();

                    SPEEDL = 180;

                    SPEEDR = 255;

                    memory = PORTB&0b00001111;
                    lcd_goto(20);
                    send_string ("m_right2");
            }
            else if
((s_left==0)&&(s_mleft==1)&&(s_mright==0)&&(s_right==0))
            {       forward();

                    SPEEDL = 200;

                    SPEEDR = 255;

                    memory = PORTB&0b00001111;
                    lcd_goto(20);
                    send_string ("m_right1  ");
            }
            else if
((s_left==1)&&(s_mleft==1)&&(s_mright==1)&&(s_right==0))
            {       forward();

                    SPEEDL = 200;

                    SPEEDR = 255;

                    memory = PORTB&0b00001111;
                    lcd_goto(20);
                    send_string ("m_right1  ");
            }
            else if
((s_left==0)&&(s_mleft==1)&&(s_mright==1)&&(s_right==0))
            {       forward();

                    SPEEDL = 255;

                    SPEEDR = 255;

                    memory = PORTB&0b00001111;
                    lcd_goto(20);
                    send_string ("middle ");
            }
            else if
((s_left==0)&&(s_mleft==0)&&(s_mright==1)&&(s_right==0))
            {       forward();

                    SPEEDL = 255;

                    SPEEDR = 200;

                    memory = PORTB&0b00001111;
```

```
                lcd_goto(20);
                send_string ("m_left1    ");
            }
            else if
((s_left==0)&&(s_mleft==1)&&(s_mright==1)&&(s_right==1))
            {   forward();

                SPEEDL = 255;

                SPEEDR = 200;

                memory = PORTB&0b00001111;
                lcd_goto(20);
                send_string ("m_left1    ");
            }
            else if
((s_left==0)&&(s_mleft==0)&&(s_mright==1)&&(s_right==1))
            {   forward();

                SPEEDL = 255;

                SPEEDR = 180;

                memory = PORTB&0b00001111;
                lcd_goto(20);
                send_string ("m_left2 ");
            }
            else if
((s_left==0)&&(s_mleft==0)&&(s_mright==0)&&(s_right==1))
            {   forward();

                SPEEDL = 255;

                SPEEDR = 0;

                memory = PORTB&0b00001111;
                lcd_goto(20);
                send_string ("left    ");
            }
            else if
((s_left==0)&&(s_mleft==0)&&(s_mright==0)&&(s_right==0))
            {   forward();

                if ((memory == 0b00000001)||(memory ==
0b00000011)||(memory == 0b0000010)||(memory == 0b0000111))
                {
                    SPEEDL = 0;

                    SPEEDR = 255;

                }
                else if ((memory == 0b00001000)||(memory ==
0b0000100)||(memory == 0b00001100)||(memory == 0b0001110))
                {
                    SPEEDL = 255;
```

```
                        SPEEDR = 0;

                  }
            }
            else if
((s_left==1)&&(s_mleft==1)&&(s_mright==1)&&(s_right==1))
            {
                  forward();

            }
      }
}

// Mode 2 : Ultrasonic
// Description: Maintain distance measure using ultrasonic between
obsacle and robot
// Can choose data acquiring methode between ADC, PWM and UART
//=======================================================================
void ultrasonic(void)
{
      int distance;
                                                // variable for distance
measuring
      char n=1;
                                                // index for indicat mode
      lcd_clr();
                                                // clear lcd
      send_string("Measure mode");
                                                // display string
      lcd_goto(20);
                                                // lcd goto 2nd line
      send_string("ADC");
                                                // Display string "ADC"

      while(1)
                                                // loop forever unless sw2 is
pressed
      {
            if( !sw1)
                                                // if button s1 is pressed
            {
                  while(!sw1);
                                                // wait until sw1 is release
                  n++;
                                                // increment n
                  lcd_goto(20);
                                                // goto 2nd line
                  switch (n)
                                                // check current value of n
                  {
                        case 2 :    send_string("PWM ");

                                                break;
                                                // break out from switch
                        case 3 :    send_string("UART");
```

```
                                        break;
                                        // break out from switch
                        default:    send_string("ADC ");
                                    // if not 2 or 3, set ti back to 1
and display "ADC"
                                        n =1;
                    }

            }

            if (!sw2)
                                        // if button sw2 is pressed
            {
                    while(!sw2);
                                        // wait until sw2 is release
                    break;
                                        // break out form looping
            }
    }


    lcd_clr();
                                        // clear the lcd
    send_string("Distance");
                                        // display string "Distance"

    while(1)
                                        // loop forever
    {
            lcd_goto(20);
                                        // lcd goto 2nd line
            distance = us_value(n);
                                        // disance variable are equal
to value return from subroutine us_value
            dis_num(distance);
                                        // display the value of
distance


            if (distance> 40)
                                        // check if distance more
than 40
            {
                    forward();
                                        // then forward with full
speed
                    SPEEDL = 255;
                    SPEEDR = 255;
                    buzzer = 0;
            }
            else if (distance> 30)
                                        // check if distance more
than 40
            {
                    forward();
                                        // forward with medium speed
                    SPEEDL = 230;
```

```
			SPEEDR = 230;
			buzzer = 0;
		}
		else if( distance >20)
						// check if distance more
than 40
		{
			stop();
						// then stop
			buzzer = 0;
		}
		else
						// else, distance less than
20
		{
			backward();
						// then backward with medium
speed and on the buzzer
			SPEEDL = 230;
			SPEEDR = 230;
			buzzer = 1;
		}
	}

}


// Mode 3 : Analog Distance Sensor
// Description : Maintain distance between robot and obstacle
//==================================================================
void analog_sen(void)
{
	int distance;
						// variable for distance
measuring
	lcd_clr();
						// clear lcd
	send_string("Distance");
						// display string

	while(1)
	{
		lcd_goto(20);
						// lcd goto 2nd line
		read_adc(CHANNEL1);
						// read adc channel 1 (
analog distance sensor input)
		distance = result;
						// assign distance as the
result oft he reading
		dis_num(result);
						// display the value

		if (distance< 200)
						// check if distance less
than 200
		{
```

```
                backward();
                                        // backward with full speed
                SPEEDL = 255;
                SPEEDR = 255;
                buzzer = 0;
            }
            else if (distance< 250)
                                        // check if distance less than 250
            {
                backward();
                                        // backward with medium speed
                SPEEDL = 230;
                SPEEDR = 230;
                buzzer = 0;
            }
            else if( distance < 300)
                                        // check if distance less than 300
            {
                stop();
                                        // stop
                buzzer = 0;
            }
            else
                                        // else, distance more than 300
            {
                forward();
                                        //forward with medium speed and on
buzzer
                SPEEDL = 230;
                SPEEDR = 230;
                buzzer = 1;
            }
        }
}


//Mode 4 :  Xbee
// Description : Control the robot using UART ( XBEE or an UART
wireless module.
//=======================================================================

void wireless_xbee (void)
{
        lcd_clr();
                                        // clear the lcd
        while(1)
                                        // looping forever
        {
            lcd_goto (0);

            if (data[0] == 100)
                                        // check if UART start byte is met
            {
                send_string("  XBEE CONTROL  ");
                                        // display string
                SPEEDL = 200;
                                        // set the motor speed
```

```
                SPEEDR = 200;
                while(1)
                {
                      lcd_goto(20);
                      if (RCREG == '8')
                                  // if character '8' is detected,
the robot move forward
                      {
                            forward();
                            send_string("FORWARD        ");
                      }

                      else if (RCREG == '2')
                                  // if character '2' is detected,
the robot move backward
                      {
                            backward();
                            send_string("BACKWARD       ");
                      }
                      else if (RCREG == '6')
                                  // if character '6' is detected,
the robot turn right
                      {
                            right();
                            send_string("TURN RIGHT     ");
                      }

                      else if (RCREG == '4')
                                  // if character '4' is detected,
the robot turn left
                      {
                            left();
                            send_string("TURN LEFT      ");
                      }

                      else if (RCREG == '5')
                                  // if character '5' is detected,
then stop the robot
                      {
                            stop();
                            send_string("INVALID COMMAND  ");
                      }

                      else
                                        // else then stop the robot
                      {
                            stop();
                            send_string("INVALID COMMAND  ");
                      }
                }
          }
          else  send_string("COMMAND");
    }
}


// Ultrasonic value
```

```
// Description : Retrive data from Ultrsonic. Can choose methode
between ADC, PWM and UART
// Parameter : mode             1) using analog
//                              2) using pwm
//                              3) using uart
//==============================================
unsigned int us_value (unsigned char mode)
                                        // subroutine for ultrasonic
measurement
{
      unsigned int value;
      switch (mode)
                                                    // retrive value of
measured distane based on the methode selected
      {
            case 1:     read_adc(CHANNEL0);
                        value = result;
                                                    // max vslue 2.55v =
2.55/5 *1024 - 1 =  522, resolution = 10mV/ inch, 10m/5*1024 =~ 2
                        break;
            case 2:     value = TH;
                                                    // each value =
256*4/20mhz = 51.2us, i inch = 147us
                        break;
                                                    // can change using
smaller timer prescale, but resulation fixed 147us / inch
            case 3:     if ( data [0]=='R')      value = (data[1] -
0x30)*100+ (data[2] - 0x30)*10+ (data[3] - 0x30); // 1 = 1 inch
                        else
                        {
                              lcd_goto(20);
                                                    // if stater byte is
not 'R', Display 'not connected'
                              send_string("not connected");
                              while(1);
                                                    // loop forever
                        }
            default: ;
      }
      return value;
}


// read adc
// Description: subroutine for converting analog value to digital with
average 200 samples
// Parameter : config  ( select the channel )
//============================================================

void read_adc(char config)
{
      unsigned short i;
      unsigned long result_temp=0;

      ADCON0 = config;
      delay(10000);                             // delay after changing
configuration
```

```
	for(i=200;i>0;i-=1)				//looping 200 times for
getting average value
	{
		ADGO = 1;				//ADGO is the bit 2 of
the ADCON0 register
		while(ADGO==1);				//ADC start, ADGO=0
after finish ADC progress
		result=ADRESH;
		result=result<<8;		//shift to left for 8 bit
		result=result|ADRESL;		//10 bit result from ADC

		result_temp+=result;
	}
	result = result_temp/200;		//getting average value
	ADON = 0;				//adc module is shut off
}


// Motor control function
// Description : subroutine to set the robot moving direction
//==========================================================================
void forward ()
{
	motor_ra = 0;
	motor_rb = 1;
	motor_la = 0;
	motor_lb = 1;
}

void backward ()
{
	motor_ra = 1;
	motor_rb = 0;
	motor_la = 1;
	motor_lb = 0;
}

void left()
{
	motor_la = 1;
	motor_lb = 0;
	motor_ra = 0;
	motor_rb = 1;
}
void right()
{
	motor_la = 0;
	motor_lb = 1;
	motor_ra = 1;
	motor_rb = 0;
}

void stop()
{
	motor_la = 1;
	motor_lb = 1;
```

```
        motor_ra = 1;
        motor_rb = 1;
}


//      LCD    functions
//==========================================================================
void delay(unsigned long data)              //delay function, the delay
time
{
        for( ;data>0;data-=1);              //depend on the given value
}

void send_config(unsigned char data)        //send lcd configuration
{
        rs=0;                               //set lcd to config mode
        lcd_data=data;                      //lcd data port = data
        delay(400);
        e_pulse();
        //pulse e to confirm the data
}

void send_char(unsigned char data)          //send lcd character
{
        rs=1;                               //set lcd to display mode
        lcd_data=data;                      //lcd data port = data

        delay(400);
        e_pulse();
        //pulse e to confirm the data
}

void e_pulse(void)
        //pulse e to confirm the data
{
        e=1;
        delay(300);
        e=0;
        delay(300);
}

void lcd_goto(unsigned char data)           //set the location of the lcd
cursor
{
        if(data<16)                         //if the given value is (0-
15) the
        {
        //cursor will be at the upper line
                send_config(0x80+data);
        }
        else                                //if the given value is (20-
35) the
        {
        //cursor will be at the lower line
                data=data-20;
        //location of the lcd cursor(2X16):
```

```
            send_config(0xc0+data);                              // --------
---------------------------------------------
      }
      // |  |00|01|02|03|04|05|06|07|08|09|10|11|12|13|14|15|  |
}
      // |  |20|21|22|23|24|25|26|27|28|29|30|31|32|33|34|35|  |

      // -----------------------------------------------------

void lcd_clr(void)
      //clear the lcd
{
      send_config(0x01);
      delay(350);
}

void send_string(const char *s)                          //send a string
to display in the lcd
{
      while (s && *s)send_char (*s++);
}

void dis_num(unsigned long data)
{
      unsigned char hundred_thousand;
      unsigned char ten_thousand;
      unsigned char thousand;
      unsigned char hundred;
      unsigned char tenth;

      hundred_thousand = data/100000;
      // devide to get the numerator                eg: 5234/1000 = 5
      data = data % 100000;
      // modulas to get the remainder          eg: 5234%1000 = 234
      ten_thousand = data/10000;
      data = data % 10000;
      thousand = data / 1000;
      data = data % 1000;
      hundred = data / 100;
      data = data % 100;
      tenth = data / 10;
      data = data % 10;

      send_char(hundred_thousand + 0x30);
      //0x30 added to become ASCII code char '0' to '9'
      send_char(ten_thousand + 0x30);
      send_char(thousand + 0x30);
      send_char(hundred + 0x30);
      send_char(tenth + 0x30);
      send_char(data + 0x30);
}
```