HUMAN MACHINE INTERFACE DEVELOPMENT OF A DISTRIBUTION AUTOMATION SYSTEM

Muhammad Azlan Arif bin Sa'adon

Bachelor of Electrical Engineering
(Control, Instrumentation and Automation)

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

FINAL YEAR PROJECT II

BEKU 4792

TITLE : HUMAN MACHINE INTERFACE (HMI) DEVELOPMENT OF A DISTRIBUTION AUTOMATION SYSTEM

NAME	MUHAMMAD AZLAN ARIF BIN SA'ADON
IC NO.	910903-06-5175
MATRIX NO.	B011010067
COURSE	BACHELOR OF ELECTRICAL ENGINEERING (CONTROL, INSTRUMENTATION & AUTOMATION) - BEKC
SUPERVISOR NAME	PROFESSOR MADYA MOHD ARIFF BIN MAT HANAFIAH

" I hereby declare that i have read through this report entitle "Human Machine Interface Development of a Distribution Automation System" and found that it has comply the partial fulfillment for awarding the degree of Bachelor of Electrical Engineering (Control, Instrumentation and Automation)"

Signature :

Supervisor's Name:

Date :

HUMAN MACHINE INTERFACE (HMI) DEVELOPMENT

OF A DISTRIBUTION AUTOMATION SYSTEM

MUHAMMAD AZLAN ARIF BIN SA'ADON

A report submitted in partial fulfillment of the requirements for the degree of Bachelor of Electrical Engineering (Control, Instrumentation and Automation)

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2014

I declare that this reports entitle “Human Machine Interface Development of a Distribution Automation System” is the result of my own research except as cited in the references. The report has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Signature:

Name :

Date :

To my beloved mother and father

ACKNOWLEDGEMENT

First of all, i would like to express my sincere appreciation to my project supervisor, Professor Madya Mohd Ariff bin Mat Hanafiah for encouragement, guidance, motivate and toleration while completing this report. I am also would like to thankful to my co-supervisor Siti Hajar binti Raman for the advice and idea. This project would not have been same as presented here without their continue support and motivation. In preparing this report, i was in contact with many people, academicians, technicians and practitioners. They have contributed to my understanding in this project. I am also indebted to Universiti Teknikal Malaysia Melaka (UTeM) for funding my project and provide good facilities for me while completing this project. I also recognized my fellow postgraduate friend for their support. I am also extending sincere appreciation to all my colleagues for their views and useful tips. I am grateful to all my family members that give me moral support and motivation while completing this project.

ABSTRACT

A distribution automation system (DAS) is a monitoring and controlling power supply grid network system. The most important of this system is the operation reliability to supply the power to the customer efficiently. The outages happen because of an overload in the region supply area. It will affect the critical 24-hours none stop customer such as hospital, financial institutions and many more. The power supply restoration will take longer time while one or second of outages cannot be accepted. This research is to develop a Human Machine Interface (HMI) of a Distribution Automation System (DAS) as to increase the reliability of the system. In this research, the HMI is capable to monitor and control the system efficiently when there is fault occurring. A data logger is needed to store the information and data for further analysis.

ABSTRACT

Sistem automasi pembahagian ialah sebuah sistem pemerhatian dan pengawalan rangkaian bekalan kuasa elektrik. Kepercayaan terhadap operasi pembekalan kuasa kepada pelanggan secara cekap ialah faktor yang terpenting dalam sistem. Gangguan bekalan elektrik terjadi disebabkan oleh sistem menangung lebihan bebanan. Kesan utama gangguan bekalan elektrik ialah pengguna kritikal yang mengunakan secara seharian seperti hospital, institusi kewangan, dan lapangan terbang. Pemulihan bekalan elektrik juga memakan masa yang lama. Kajian ini ialah untuk membangunkan Human Machine Interface (HMI) untuk sistem automasi pembahagian untuk meningkatkan lagi kebolehpercayaan terhadap sistem. Di dalam kajian ini HMI berkebolehan untuk digunakan sebagai pemerhatian dan pengawalan sistem secara efektif sekiranya berlaku sebarang gangguan sistem. Sebuah data logger diperlukan untuk menyimpan maklumat dan data untuk penggunaan di masa hadapan.

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENT

CHAPTER 1

INTRODUCTION

1.1 Background

Electricity supply has become one of the most important utilities. However, the increasing of customer and the demand whether in domestic or commercial cause lots of challenge to the electrical supplier to deliver a better service. Distribution networks are major parts of the total electrical supply system, as it provides the final link between the customer and the bulk transmission system [1].In order to improve the reliability of the system, distribution automation is applied to the system.

Distribution Automation System (DAS) is a utility's operation of the distribution power system. A multifunction system to monitor and control the scattered feeder remote terminal unit (FRTU) is the definition of DAS [2]. DAS is based on a centralized system, where the central server receives all information from the field and has full authority of monitoring and control Remote Terminal Unit (RTU) [2].

To improve the reliability of the system, there are many ways and one of it is Supervisory Control and Data Acquisition (SCADA) by using Human Machine Interface (HMI). HMI is a computerized system that between human and machine occur. It is used for maximum supervisory on system, control and receives the feedback of the machine in order to achieve effectiveness of operation [3].

In this project, a small scale of load will be used as *emulation* of the large scale of DAS. It is because the real scale of DAS is way of scope to bachelor level. Besides that, there is a lot of aspects and protocols need to be considered before applying in real DAS. Tenaga Nasional Berhad (TNB), one of Malaysian company will not let their system to be tested or simulate to avoid and prevent the interruption to their system that give risk them cost.

The small scale of load system will be implemented by using a DC Motor as load, a current sensor, SK40C. This hardware will represent the real DAS. Five different speed

of DC motor will be used to varying the current. The higher the speed of DC motor the higher current uses. When the current is high, it will represent the over current in real DAS that will risky the equipment to damage. When the current is low, it will represent the overload in DAS, which the system cannot afford to supply enough power to consume and will cause the outages.

The five different speeds will be framed by using a current sensor. This algorithm will be sent to computer and will be analysed by Visual Basic software in order to represent the data in a graphical way which is Human Machine Interface (HMI). Form HMI, it will monitor the current thoroughly from the system. When there are over current or overload occur, the HIMI will cut off the supply to protect the system before damaged whether in automatically or by manual.

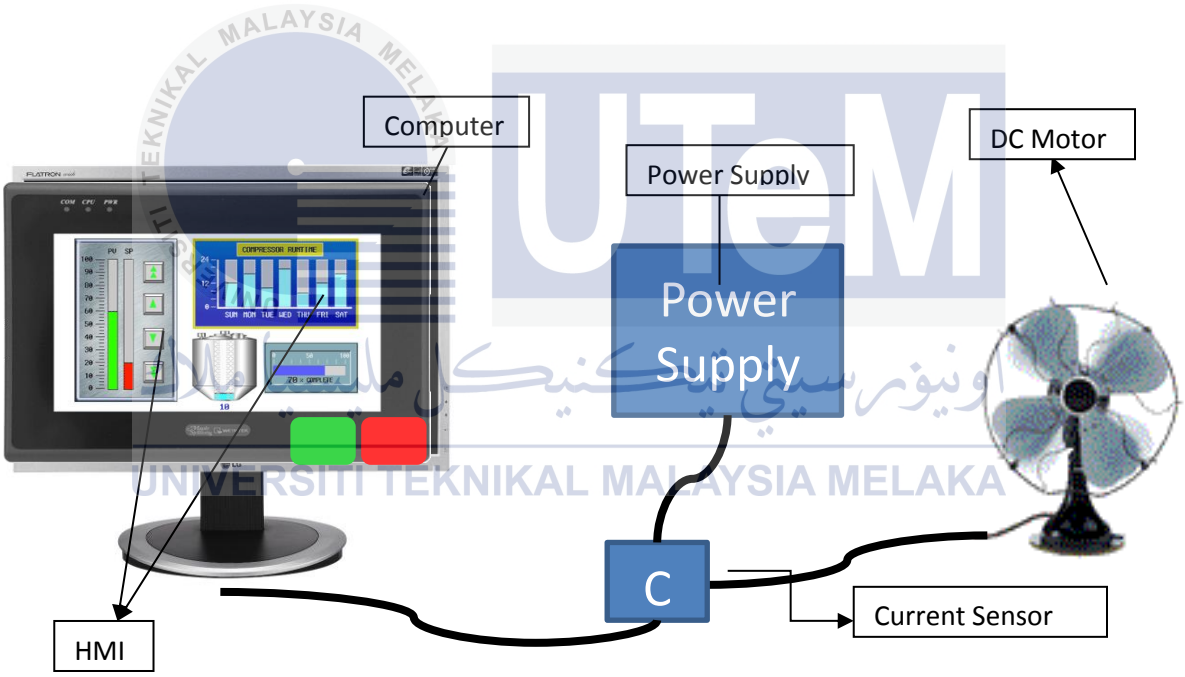


Figure 1.1 The overview of small scale of DAS

1.2 Problem Statement

The electrical supply is affecting a lot in human daily life as to run in domestic or commercial purpose. The reliability of power supply is important for country for developing country and economic growth [2][4].

The outage of electricity would be unacceptable even for one second after fault[2].When the outage occurs it will take so much time to restore back the power supply as the engineer or technician need to troubleshoot the fault location by manually. Outages can give impact the system failures seriously especially in critical those industries that require 24 hour service such as:

- Medical informatics system
- Airlines system
- Financial institutions

To overcome these outages, Distribution Automation System (DAS) through the Human Machine Interface (HMI) is applied. Distribution Automation System the most important function is to improve the power supply reliability. The DAS will monitor and collecting the data needed for HMI application that will be developed using Visual Basic (VB) software.

1.3 Objective

The objectives of this project are:

i) To implemented load current sensor in order to get algorithm for current sensor by using microcontroller and for data logging.

ii) To develop Human Machine Interface (HMI) that can monitor and control Distribution Automation System (DAS) by using Visual Basic (VB) software

1.4 Scope of Research

DAS is a big scale system and there are a lot of aspects and protocols need to be considered. There is no way to Tenaga Nasional Berhad (TNB), the Malaysian utility company to let their DAS to be test by some experiment. It is to avoid the interruption. This project will be implemented in a small size of scale of load system as *emulation* in order to show that my theory on DAS is capable to increase the reliability of the system.

This project will focus on development of Human Machine Interface (HMI) for Distribution Automation System (DAS) by using Visual Basic Software. The hardware uses a current sensor to collect the data. The hardware will be used a direct current Motor as a load and will be implemented in five different speed in order to vary the current. The algorithm produced from varying current will be used to set the limitation before the system is outages or over current.

1.5 Report Outlines

In chapter 1, it's an overview of the research is discussed along with objective and scope. This chapter also will go deeper into the problem statement that referring current. In chapter 2,it will be a detailed about hardware and software details and the reason of selecting that equipment. This chapter also will cover the related previous works. In chapter 3, the procedure of this research is discussed with the aids of the flow chart. The preliminary project also covered in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Theory

In order to develop monitoring and controlling system, the criteria of software and hardware also play role to increase the reliability of the system. Supervisory Controlling and Data Acquisition (SCADA) is a system that monitor and control industrial process by computer controlled [5]. There are four types of SCADA which is Human Machine Interface (HMI), communication network, field device interface and field device [6].

2.2 Power Distribution Substation

Power substation is a station that received supply either from Main Intake Substation (PMU), Primary Distribution Substation (PPU) or Primary Switch Station (SSU). Distribution substation is the final station providing electricity to consumer in Power Grid System. In power substation building, it contains of three major electrical equipment which is switchgear, transformer and low voltage distribution board (LVDB). However, the use of transformers is depending on the maximum demands of that area. The larger the demands of power supply, the higher the quantities of transformer.

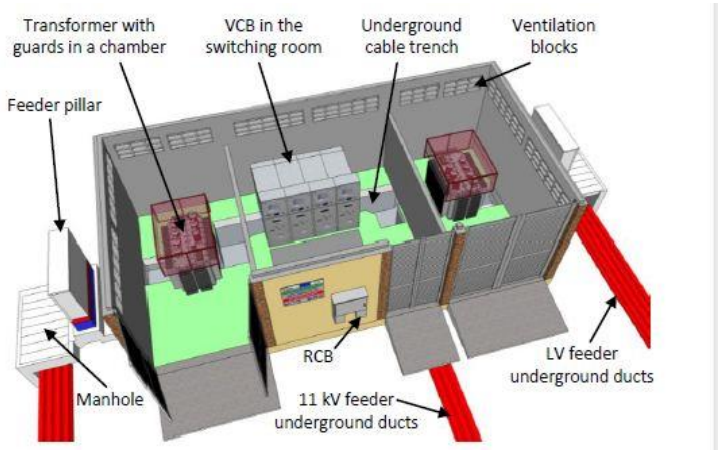


Figure 2.1 : Overview of double chamber substation

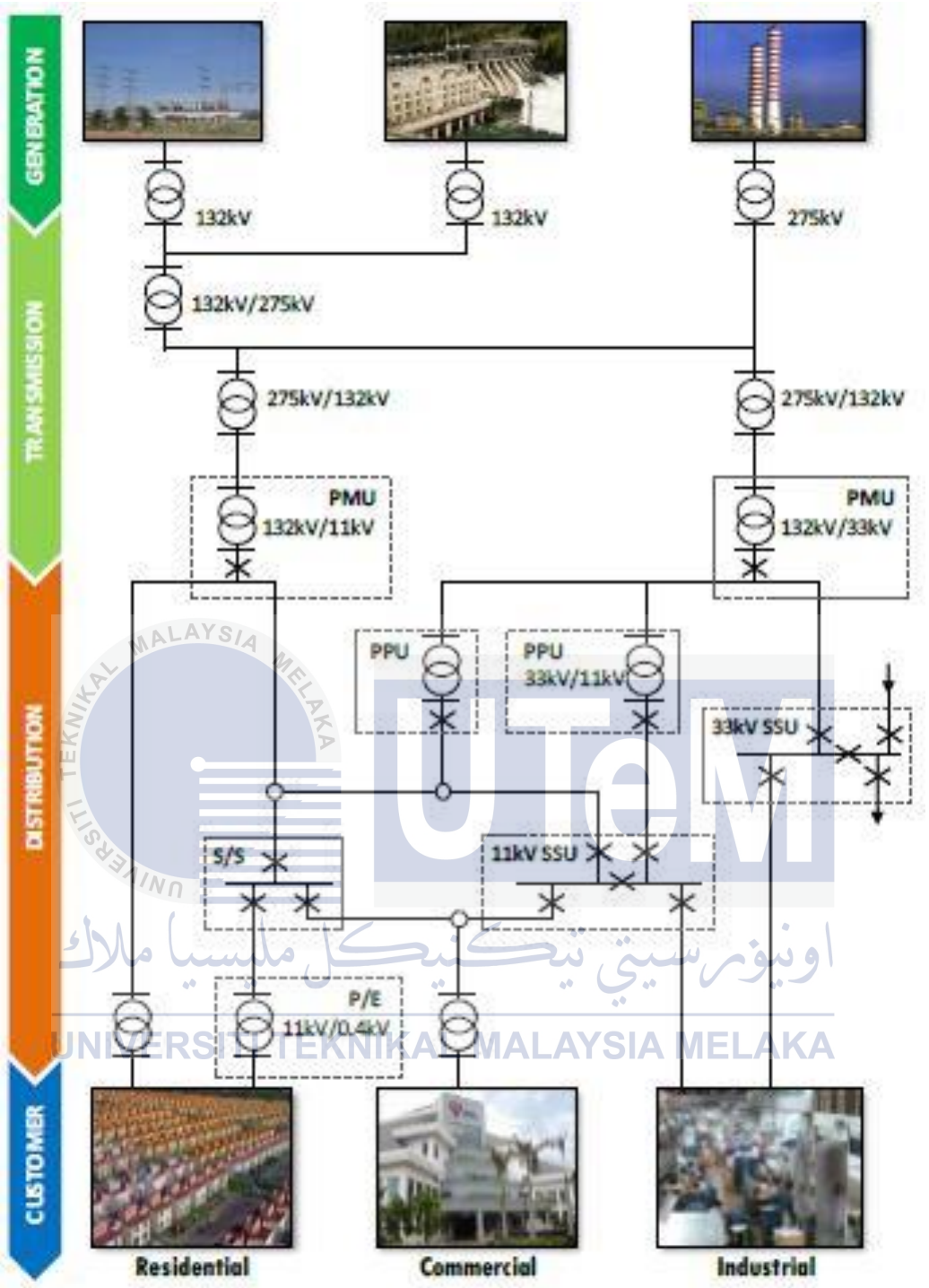


Figure 2.2: Typical flow of electricity

2.3 Distribution Automation System (DAS)

DAS is a multi-function utility's system to monitor and control the Remote Terminal Unit (RTU). By using a computer, DAS is an automatic controller does not need human force to monitor the system. It be able to increase the efficiency in the context of automation in power supply system. This DAS are equipped with protection system so when there is fault happening, the system will trip to avoid any risks.

2.4 Fault Current

Fault current is a situation in which the flow of current that is travelling through an electrical circuit is not within a normal range. There are a lot of causes of fault current and it's divided into two which is internal and external cause. For the external cause or to be exact is transient fault, is a physical contact that interrupt to the system such as momentary tree contact, animal contact and lightning strike. For internal cause, it happens because of overload that carrying by the substation. As an example if the substations carry maximum demands is 232 kW.

$$P = IV\cos\theta$$

$$I = \frac{P}{V\cos\theta}$$

$$= \frac{232 \times 1000}{415\sqrt{3}\,(0.8)}$$

$$= 403.45\ Amp$$

P = Power

I = Current

V = Voltage

$\cos\theta$ = Power factor

By using this formula, the substation needs to produce 403.45 Amp in order to supply maximum demands which is 232kW. In order to select the transformer for the substation, the transformers must work under 70 percent to 80 percent capacity. If the transformer works more than requirement it will trip the system for protection. According to the Table 2.1 transformer with power of 500 kVA is selected as it can produce 665 ampere which is more than current maximum demands which is 403.45 ampere and it makes the transformer works in the range 60 percent which is meeting the requirement of transformer selection. However, after years, the maximum demands are larger due to increasing of power supply usage. After years, new maximum demand now is 400 kW. By using the same formula, the new current maximum demand is 695 ampere which had made the transformer working over capacity with 104 percent. This situation is called overload.

Table 2.1 : Transformer rating

Transformer Power	Ampere
100 kVA	133 A
300 kVA	399 A
500 kVA	665 A
750 kVA	997.5 A
1000 kVA	1330 A

2.5.1 Human Machine Interface (HMI)

Human machine interface (HMI) that will be developed by using Visual Basic (VB) software is device systems that build for presenting processed data to a human operator for monitoring and controlling the process system. It is part of a supervisory system used as a safety tool and widely used in SCADA.

There are several reasons why VB software is chosen for Human HMI application compare to SCADA package software such as Indusoft software. For VB there is no runtime license which costs cheaper as it does not need to buy and renew the license [9].

With SCADA packages, it is expensive in general that the customer will incur the additional runtime cost for each application which must be licensed and renew. Besides that, VB is more easy to build the interface as its more flexible than SCADA packages [9].

2.5.2 Communication Network

All data that will be collected by current sensor in Distribution Automation System (DAS) will be sent through wired system. Soon as the computer receives the signal, the Visual Basic (VB) will visualize the data in the simple understanding interface. By then human operator will monitor and controlling the system.

There are few types of wired that can be used which is coaxial metallic cable, Fibre optic cable and Power Line Carrier (PLC). Fibre optic cable may cost more than other two but the speed of transferring data is the best among three. This will be a plus point to the system [10].

The reason why the data will transfer from the current sensor to computer by wired because the wired network offer faster than wireless networks. Besides that, wired network, its capabilities to work up to maximum potential more than a wireless network. This feature will increase the reliability of the Distribution Automation System as to increase the speed of system [10].

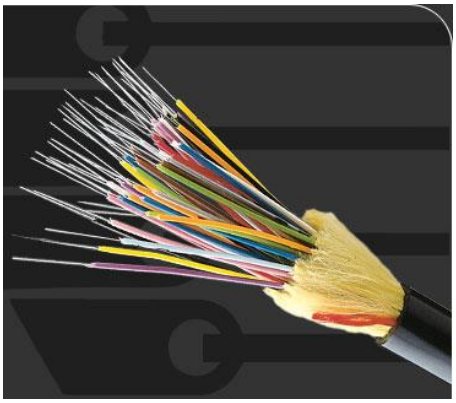


Figure 2.3: The picture of fiber optic cable

The other reason why wired is better than wireless is because of security guarantee. With wired, all the activity of transmission is safe compare to wireless that may occur some interruption from surrounding. Besides that, wired networks are cheaper and easy to setup compare to the wireless network. Here are some comparison between the wired network and wireless network [10].

Table 2.2: The comparison between Wired and Wireless Network [10]

Network	Pros	Cons
Wired	• Ethernet cable, switches and hubs is very cheap and reliable • Wired LANs offer superior performance	• Need to run cables in difficult environment
Wireless	• Neater working environment as its less cabling	• May have problem with the speed of transferring data • Expensive adapter

2.5.3 Field Device Interface

For device interface, programmable integrated circuit (PIC) is used in this research. The PIC is a small computer in a single integrated circuit that contain programmable peripheral, memory and processor core.

Figure 2.4 : The picture of programmable integrated circuit (PIC)

Microcontroller software is use for the controller. It is because PIC is built in microcontroller languages. Here are some features for microcontroller software

- Free product lifetime technical support.
- Over 350 PIC microcontrollers supported.
- Many hardware and software libraries.
- Numerous ready-to-use practical examples.

2.5.4 Field Device

A simple hardware is built using PIC 16F877A, SK40C, push button, and motor to prototype load. A current sensor will be used to measure the current needed for different speed of the motor in order to get the algorithm data. The difference of speed will reflect in the amount of current.

2.5.4.1Current Sensor.

Figure 2.5: The picture of Cytron BB-ACS current sensor

The type of current sensor is BB-ACS 756 from Cytron. Cytron current sensor has a better feature compare to others brand as an example Panucatt current sensor. Besides that, Cytron offer an inexpensive price that suitable for this research compare to Panucatt. The table shows the comparison features of both current sensors.

Table 2.3: The comparison between Cytron and Panucatt current sensor

Types of current sensor	Features
Cytron	• X050 (50 Amp) version (model: ACS756KCA-050B-PFF-T) • Total output error 0.8% at TA = 25°C • Monolithic Hall IC for high reliability • Ultra-low power loss: 130u Ohm internal conductor resistance • 3 kV RMS minimum isolation voltage from pins 1-3 to pins 4-5 • 5.0 V single supply operation • 3 µs output rise time in response to step input current • 40 mV/A output sensitivity • Output voltage proportional to AC or DC currents • Factory-trimmed for accuracy • Extremely stable output offset voltage

Panucatt	• Fast and accurate analog output • High isolation from measured circuit • May be used as High-side or Low-Side sensing • 3V to 5V single supply operation • Measured circuit up to 300V AC/DC • High capacity stitched copper pads • Analog output proportional to measured current • Flexible connection and mounting options

2.5.4.2 SK40C

SK40C from Cytron brand is a 40pin PIC microcontroller starter kit. It's designed to offer an easy to start solution for user for (PIC MCU).

Figure 2.6: The picture of SK40C

The features of SK40C:

- ICSP connector for UIC00B -simple and fast method to load program.
- Perfect fit for 40 pins PIC16F and PIC18F.
- Operating Voltage Range: 7V - 15VDC
- 2 x programmable switch.
- 2 x LED indicator.
- Existing pad for 16 x 2 characters LCD display.
- UART connection to interface with other controller or even computer.
- USB on board for certain PIC18F.
- Users are able to utilize the function of PIC by directly plugging in the I/O components in whatever way that is convenient to user.

2.6 Related Previous Works

2.6.1 Development of a Novel fault Indicator for Distribution Automation

This paper is discussed the significant and immediate improvement in reliability and hence service to the electricity customer by developing and implementing a novel fault for distribution automation system

The proposed fault indicator is designed based on Zigbee communication as it has been designed to possess general purpose protocol with low-cost, low power consumption and self-networking and suitable for communication network in the distribution automation system.

In order to improve the system, the fault detection and identification system will be designed with extra a feature which is capabilities to find out the fault location effectively and efficiently after a fault occurred.

2.6.2 Design and Implementation of Multi agent-Based Distributed Restoration System in Distribution Automation System

The author discussed the efficient operation of power supply through remote control and monitoring. This paper highlight that the most important function of Distribution Automation System (DAS) is the restoration of stable power supply to a customer after a fault occurs.

This paper proposes to shorten the restoration time by using a concept of Multi agent System (MAS) for distributed control system as one second of outages cannot be accepted in this digital age. The restoration of power supply will take about five minutes as it needs to go through several processes

- Fault section detection
- Fault section isolation
- The restoration by connecting to other health feeder

CHAPTER 3

DESIGN METHODOLOGY

3.0 Overview

This chapter will explain the details about the methodology that will be implemented in this project. There is some part that needs to be considered before planning the methodology to avoid redundant works. In this methodology, its divided into two parts which is the first part is the research and the second part is the implementation of the project.

For this research, it starts with briefing the project and brainstorms the idea for this project in objective to achieve of understanding and analyse the objective of the project. Next, by doing a literature review of DAS, HMI, and MikroC, the project is ready to take another level which is implementation part.

For the implementation part, its divided into two parts which is hardware and software. The designed circuit is tested in the simulation first before implement in hardware to make sure the circuit is working and to avoid the redundant works.

3.1 Methodology Flow Chart

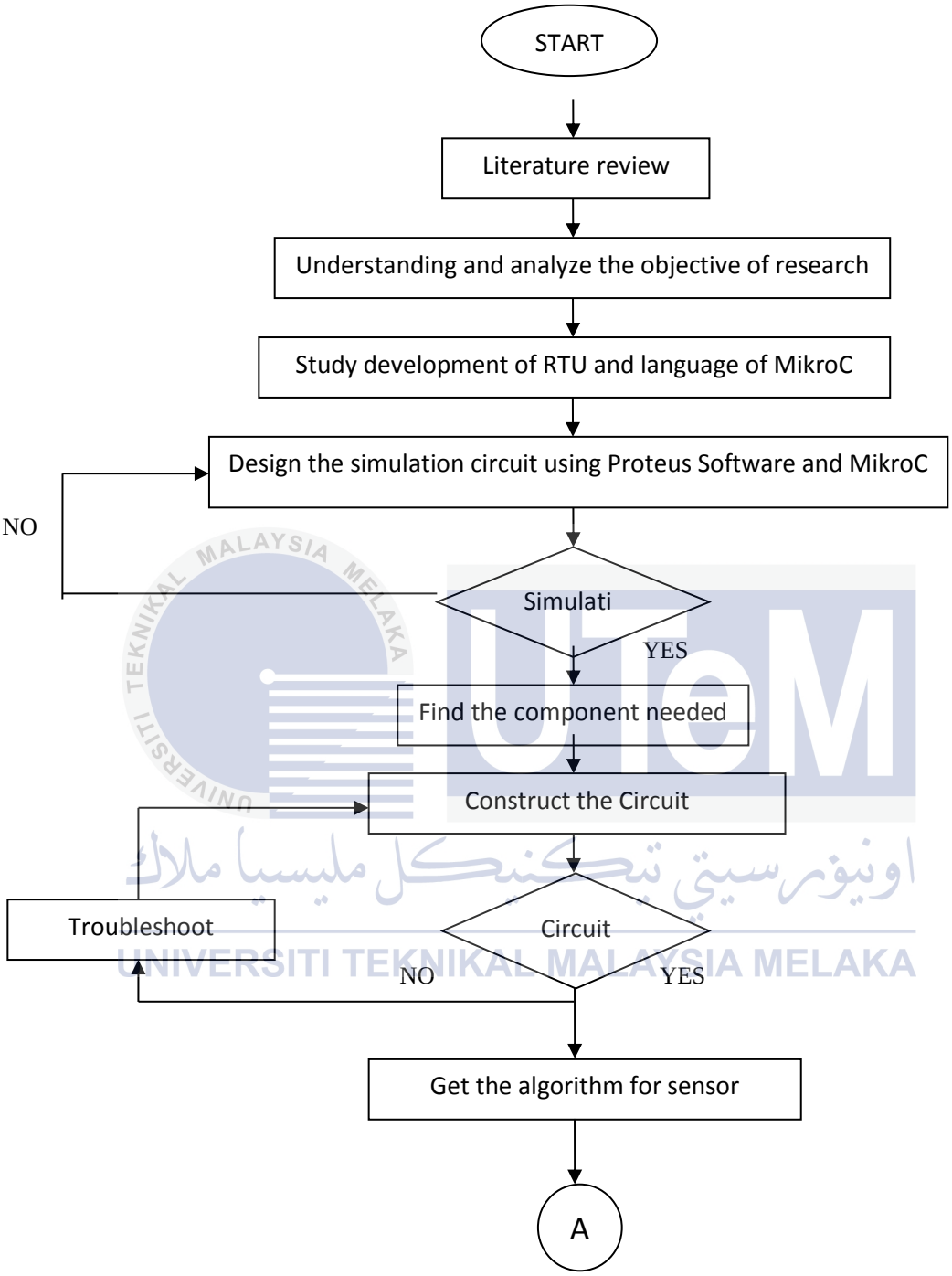


Figure 3.1: The flow chart of research

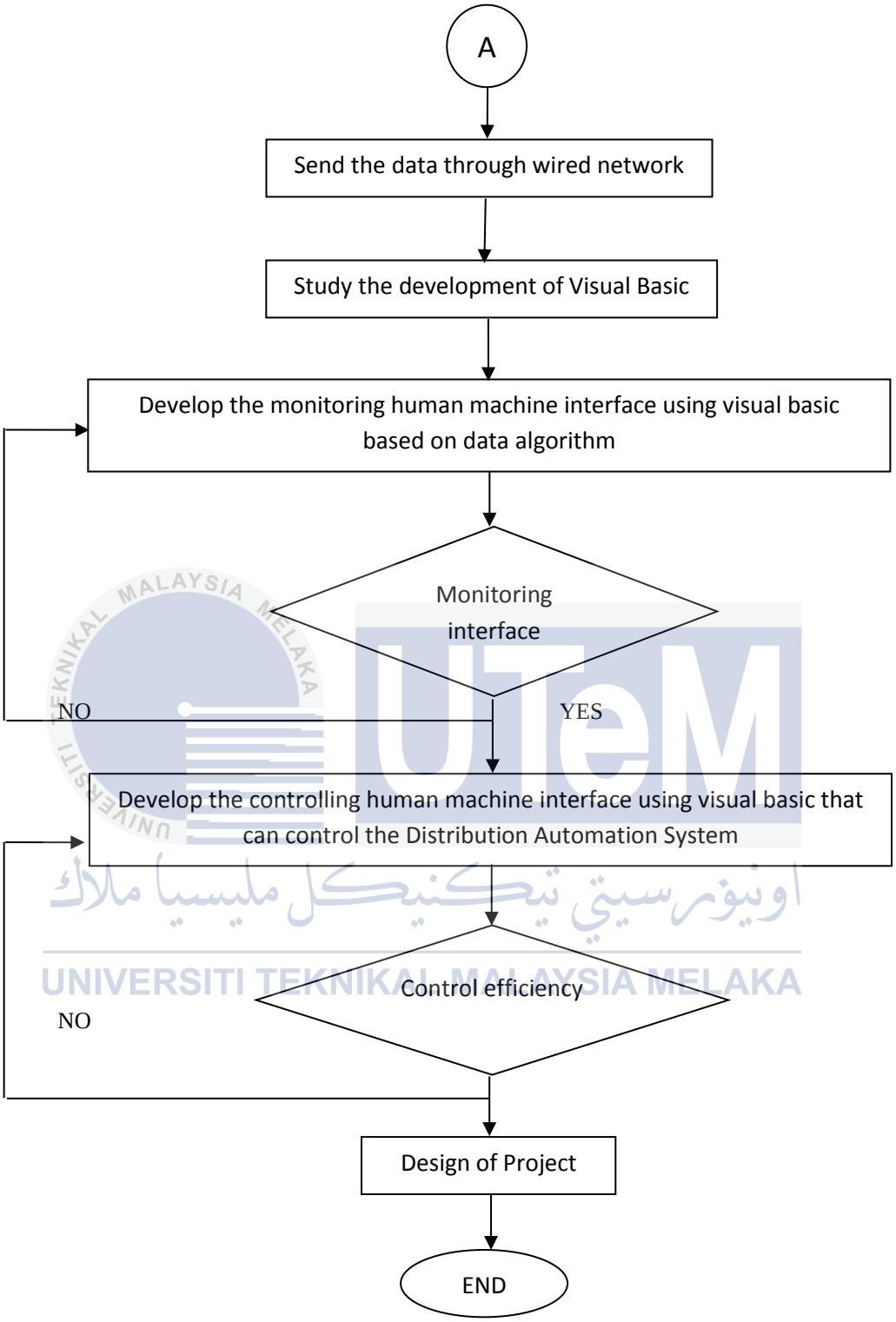


Figure 3.2: The flow chart of research

3.2 Research Methodology

3.2.1 Project Design

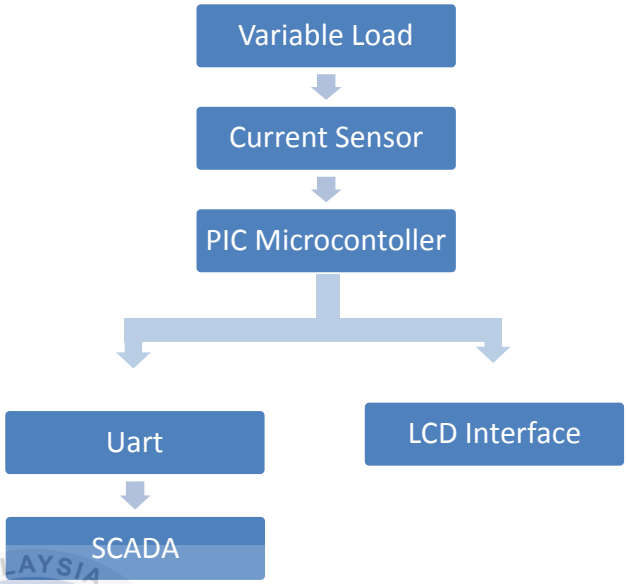


Figure 3.3: The Flow Chart of Project Design

To develop HMI using VB, firstly the data algorithm for sensor must be collected as to achieve the first objective which is to create an algorithm for the current sensor by using microcontroller. In order to get the data algorithm, a DC motor will be used as the load that received a DC power supply (12V). A PIC microcontroller which is the controller of DC motor will vary the speed. As the speed of DC motor is variable, it produces a varying current that will be detected by current sensor. The variances of the current DC motor will represent the resident area power supply usage. As the output of current sensor in analog data, it will feed into the Analog Digital Converter (ADC) pin of the microcontroller for converting into digital data.

After the microcontroller process the output data, it will show the value of the current that in the digital value to the LCD display. In the mean time, microcontroller also sends the data to computer by using UART. The UART is a medium device that uses to connect between microcontroller and computer. After the data were transmitted, VB software will analyse the data and will turn into a Graphical User Interface (GUI). VB software also will create a data logger to store information for further purposes.

A few criteria will be considered in order to build HMI that friendly user, easy to understand and effective, so that the system will not have the flaw. Besides that the HMI that will be developed will have ability to control the microcontroller so that it will whether to cut off the supply to DC motor or not.

Table 3.1 : The Comparison the Actual Scale and Project Scale

Aspects	Actual Scale	Project Scale
Load	 Resident Area	 DC Motor
Power Supply	 TNB substation	 DC adapter
Controller	 Switch Gear	 Microcontroller

3.3 Component Hardware Details

3.3.1 DC Motor (Load)

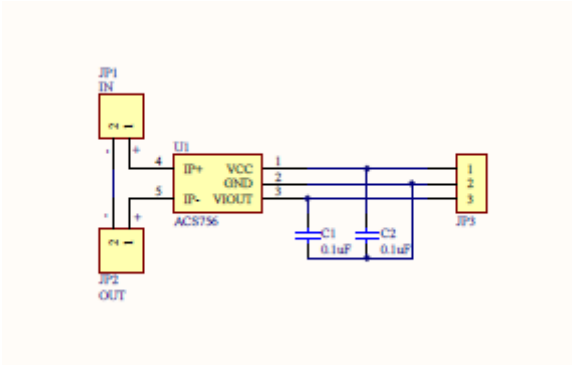
DC motor is an electric motor that converts electrical energy into mechanical energy will be used as a load to represent the consumer usage. It is for varying the current so that the different data algorithm will be read by current sensor. The DC motor will be injected 12volts power supply and the speed of rotation will be controlled by a microcontroller.

Figure 3.4: DC Motor

3.3.2 Current Sensor

Current sensor will be used to detect electrical current and generate signals in analog. The analog signal output will pin to the ADC pins at microcontroller for analyse and converting the data into digital output. In this project, current sensor from Cytron will be used as it has qualities more than others. The current sensor will solder in series with DC motor.

Table 3.2: Current sensor

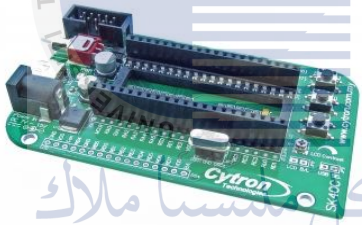
Current Sensor	Schematic Diagram
	

3.3.3 Microcontroller

Microcontroller is a small computer on a single integrated circuit which is containing a processor core, programmable input/output peripherals and memory. Microcontroller is often used as remote controls, power tools and any related product that used automatically controlled. In this project, PIC16F877A is used as it has 40 pins and makes it easier to use the peripherals as the functions are spread out over the pins.

To simplify the implemented circuit design, SK40C is also used as motherboard of PIC16877A. SK40C is a PIC microcontroller starter kit designed to offer a plug and use feature to easy start solution for PIC16F877A. This board comes with 40pins, LCD pad, UART connection, and an operating voltage range for PIC16877A.

Table 3.4: Microcontroller

SK40C	PIC16F8877A
	

3.3.4 LCD Interface

Liquid Crystal Display (LCD) is a flat panel display that uses the light modulating properties of liquid crystals. The LCD is widely used for computer monitors, instrument panels, calculators and others type of interface. In this project LCD is used to show the digital output data that already processed by the microcontroller. The digital output data which is the current reading of DC motor. This LCD has 16 characters by 2 lines display.

Figure 3.5 : LCD panel

3.3.5 UART

Universal Asynchronous Receiver/Transmitter (UART) is a device that translates data between parallel communication and serial forms. The UART is used to convert the data from microcontroller which is in this project is SK40C to the computer by using Universal Serial Bus (USB).

Figure 3.6: UART

3.3.6 Other Components

Table 3.5 Other Components

No	Components	Quantities	Details
1	Resistor	5 2	200ohm 10kohm
2	LED	5	Various colour
3	Push button	2	-
4	H-bridge	1	-
5	Adapter	1	12V

3.4 Software Programming

3.4.1 MikroC

The microcontroller needs to be programmed before capable of performing anything useful. The programme must be in the C language and compile into hex file. The hex file program will burn into PIC and implemented into the board. In this project, MikroC is chosen for the C language as it's easy to understand the sequence of routine compare to the others. This is the most important part in this project, because of microcontroller that will be programmed will acts as the brain and execution of the whole process subroutine. Any false command will lead this project into failed and cause time to troubleshoot.

3.4.1.1 Flow Chart of Programming

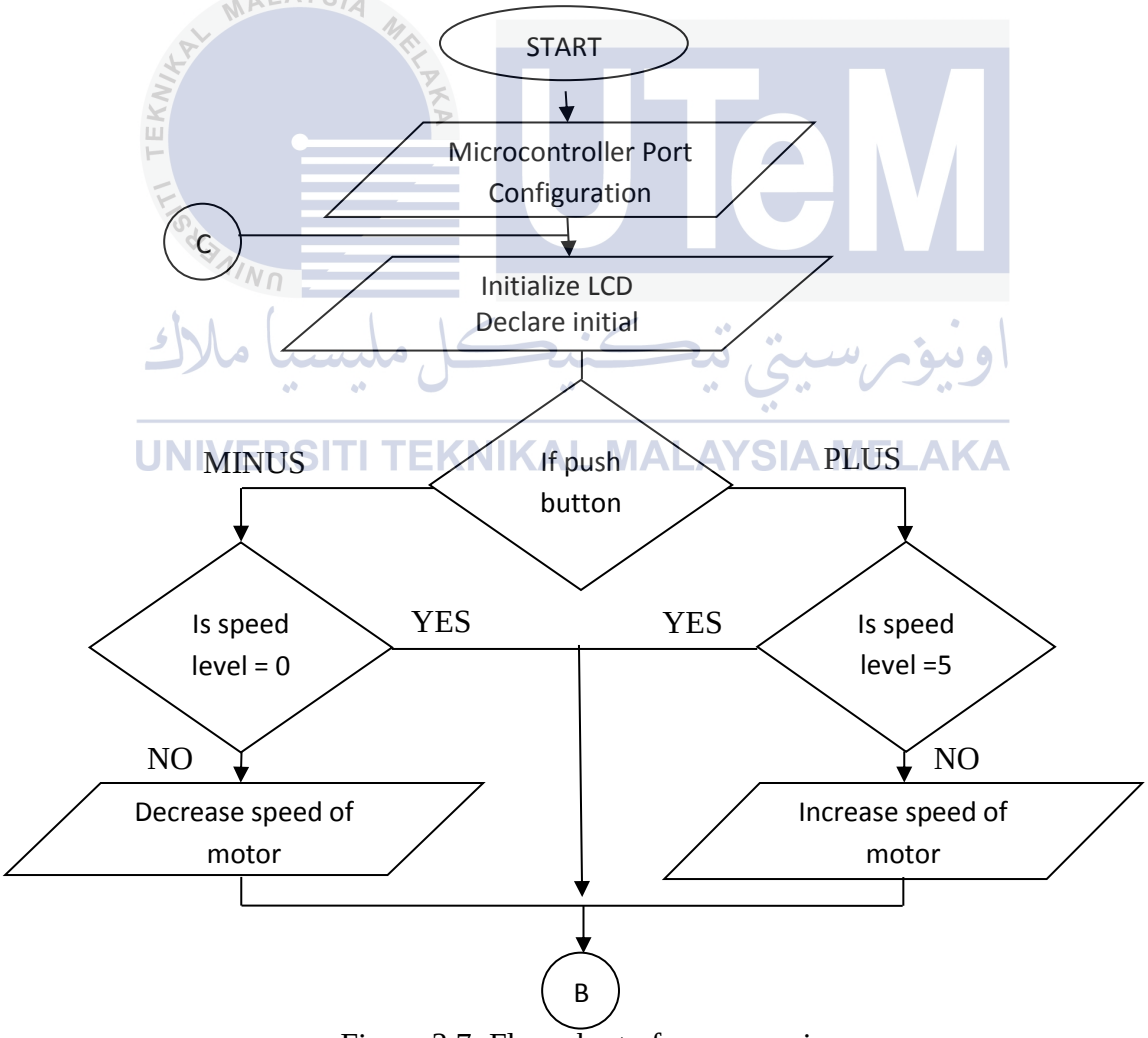


Figure 3.7: Flow chart of programming

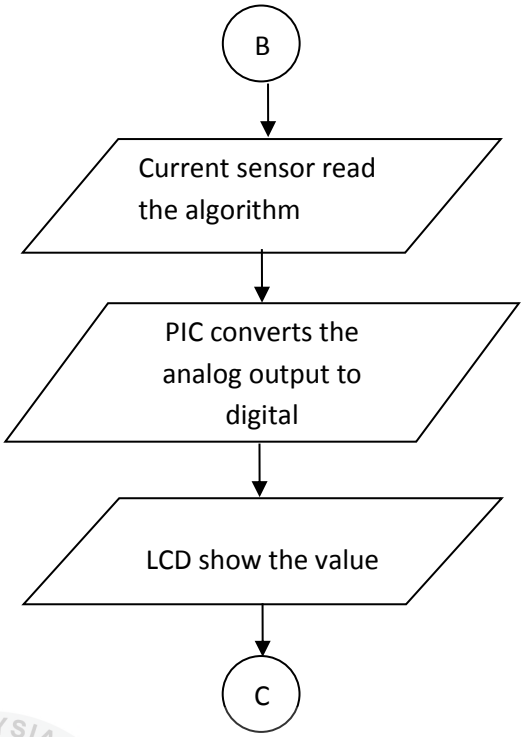


Figure 3.8: Flow chart of programming

3.4.1.2 Coding of Programming

Table 3.6 MikroC source code and explanation

NO	Coding	Explanation

1	`#include "Global.h"`
	`#include "GSM2.0.c"`
	
	`#define GPO8 PORTD.F2`
	
	`// LCD module connections`
	`sbit LCD_RS at RB2_bit;`
	`sbit LCD_EN at RB3_bit;`
	`sbit LCD_D4 at RB4_bit;`
	`sbit LCD_D5 at RB5_bit;`
	`sbit LCD_D6 at RB6_bit;`
	`sbit LCD_D7 at RB7_bit;`
	
	`sbit LCD_RS_Direction at TRISB2_bit;`
	`sbit LCD_EN_Direction at TRISB3_bit;`
	`sbit LCD_D4_Direction at TRISB4_bit;`
	`sbit LCD_D5_Direction at TRISB5_bit;`
	`sbit LCD_D6_Direction at TRISB6_bit;`
	`sbit LCD_D7_Direction at TRISB7_bit;`
	`// End LCD module connections`
	
	`unsigned int count_delay;`
	`unsigned short spike_delay=0;`
	`unsigned long value;`
	`float amp, ampScale;`
	`char txt[13];`
	`char txt2[4];`
	
	`int j;`
	`void initmain(){`
	
	`trisd=0b00000011;portd=0;`
	`trisb=0b00000011;portb=0;`
	`trisc=0;portc=0;j=0;`
	`PWM1_Init(5000);PWM1_Start();`
	
	`}`
		Calling Global and GSM2.0 file
		
		Define GP08 at portd.f2
		
		Declare LCD pins to the port
		
		Declare variable initial count_delay, short spike delay, long value, and single character with label txt and txt2.
		
		Declare integer ‘j’
		Subroutine for ‘initialize’ PIC input/output pins
		Declare port as input
		Initialize PWM channel

NO	Coding	Explanation

2	`void run(int p){`
	`portc.f0=1;portc.f3=0;`
	`switch(p){`
	`case(0):portd=0b00000111,PWM1_Set_Duty(0);break;`
	`case(1):portd=0b00001111,PWM1_Set_Duty(115);break;`
	`case(2):portd=0b00011111,PWM1_Set_Duty(135);break;`
	`case(3):portd=0b00111111,PWM1_Set_Duty(170);break;`
	`case(4):portd=0b01111111,PWM1_Set_Duty(220);break;`
	`case(5):portd=0b11111111,PWM1_Set_Duty(255);break;`
	
	`}`
	`}`
	
	`void main(){`
	`initmain();`
	
	`TRISC=0;PORTC=0;`
	`TRISA = 0xFF;`
	`TRISE = 0b00001111;`
	`ADCON1 = 0b10000000;`
	
	`Lcd_Init();`
	`Lcd_Cmd(_LCD_CLEAR);`
	`Lcd_Cmd(_LCD_CURSOR_OFF);`
		Subroutine for run
		
		Case of subroutine
		
		Main program
		
		Declare port C,A,E
		
		Initialize LCD

NO	Coding	Explanation

3	{
	current_adc_peak = 0;
	while(1){
	for(count=0;count<2000;count++)
	{
	current_adc = adc_read(0);
	if(current_adc>current_adc_peak)
	{current_adc_peak=current_adc;}
	
	if(current_adc_peak>0)
	{current = (current_adc_peak) ;
	
	currentrms = current;
	}
	else
	{current=0;currentrms=0;}
	
	floattostr(currentrms,buffer);
	Lcd_Out(2,1,codetxt_to_ramtxt(Line_Clear));
	Lcd_out(2,1,buffer);
	Lcd_Out(2,16,"A");
		declare adc port
		
		equation for adc change the output from analog to digital
		
		Print out the output on LCD

NO	Coding	Explanation

4	`if(currentrms > 250)`
	`{`
	
	`if(spike_delay > 5)`
	`{`
	
	`GPO8 = 1;`
	`PORTC.F5 = 1;`
	`delay_ms(2000);`
	
	`PORTC.F5 = 0;`
	
	`Lcd_Out(1,1,codetxt_to_ramtxt("OVERcurrent"));`
	`spike_delay=0;`
	`}`
	`spike_delay++;`
	`}`
	`else`
	`{spike_delay=0;}`
	
	`if(currentrms < 99999)`
	`{`
	
	`if(spike_delay > 5)`
	`{`
	
	`GPO8 = 1;`
	`PORTC.F5 = 1;`
	`delay_ms(2000);`
	
	`PORTC.F5 = 0;`
	
	`Lcd_Out(1,1,codetxt_to_ramtxt("UNDERcurrent"));`
	`spike_delay=0;`
	`}`
	`spike_delay++;`
		Condition if current more than 250
		LCD show overcurrent
		
		If current less than 99999 LCD show under current

NO	Coding	Explanation

5	
	if(portd.f0==1&&portd.f1==0){
	j++;
	if(j>5){
	j=5;}
	run(j),delay_ms(100);
	}
	
	else if (portd.f0==0&&portd.f1==1){
	j--;
	if(j<0){
	j=0;}
	run(j),delay_ms(100); }
	
	} } }
	}}	

3.4.2 Proteus Virtual System Modeling

Before proceeding to the hardware implementation, the code of programming must be simulate first to avoid any fault command. In this project, Isis Proteus is selected as it is software for microprocessor simulation and schematic design. It combines mixed mode SPICE circuit simulation, animated components and microprocessor model to facilitate co-simulation of complete microcontroller based on design.

It is possible to develop and simulate the interaction between software running on a microcontroller and any analog or digital electronics connected to it. In this project, the part that must be simulated is the sequences of working DC motor and the ability of adc pins to collect analog output from variable resistor, convert the data into digital and show it on the LCD.

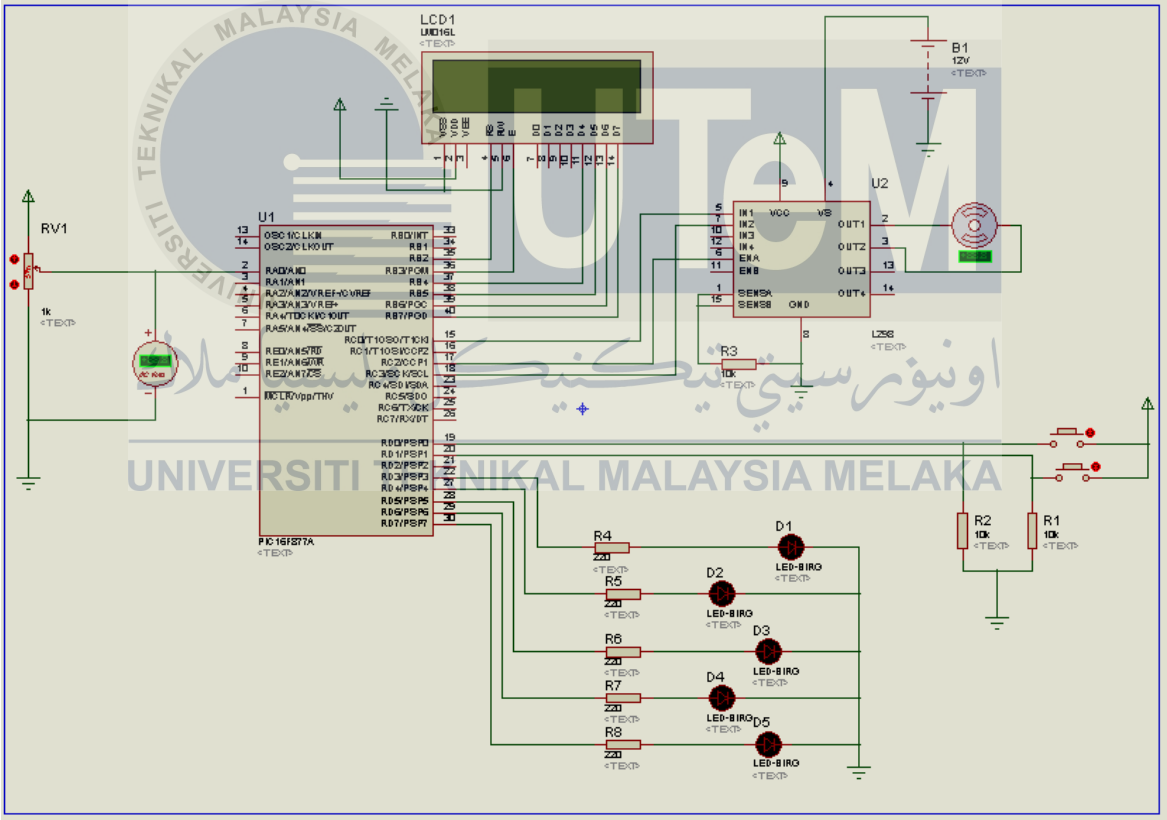


Figure 3.9: Designing Circuit using Proteus

3.4.3 Visual Basic (VB)

As this project required to develop HMI for a DAS, a few programming software will be reviewed to select the most practical, effective and efficient software. Visual Basic (VB) which is owned by Microsoft is an event-driven programming language and integrated development environment. VB is included software tools to automatically create the detailed programming required by windows. VB software just not only create Windows programs, but it also takes full authority to programmer to design graphical way that Windows works by letting programmers to draw the system on his own. That is why VB is chosen to develop HMI for SCADA system as it easier to write programs for the windows computer operating system.

Table 3.7: SCADA system Features

No	Functionality	Explanation
1	Event	• To provide information about machine operation and status to the operator • Easy interpretation and determination of need for action
2	Trending	• To provide a means for visual analysis of data on current or past machine operation
3	Alarm	• To provide notification to the operator of abnormal operating conditions and events
4	User Input	• To facilitate inputs from the operator to adjust machine operation • To perform machine setups, and respond to events
5	Data Logging Storage	• To provide for the storage of historical machine operating data for part traceability and analysis • To store and retrieve machine setup data needed

3.5 Simulation of the system

3.5.1 Overview

This section will explain about the simulation. It will divide by two which is the DC motor and the current sensor. This simulation must be done to avoid redundant works. Besides that, to check the simulation whether is follows the sequences of the flow chart programming. However, the data cannot be taken as the simulation is not in real time.

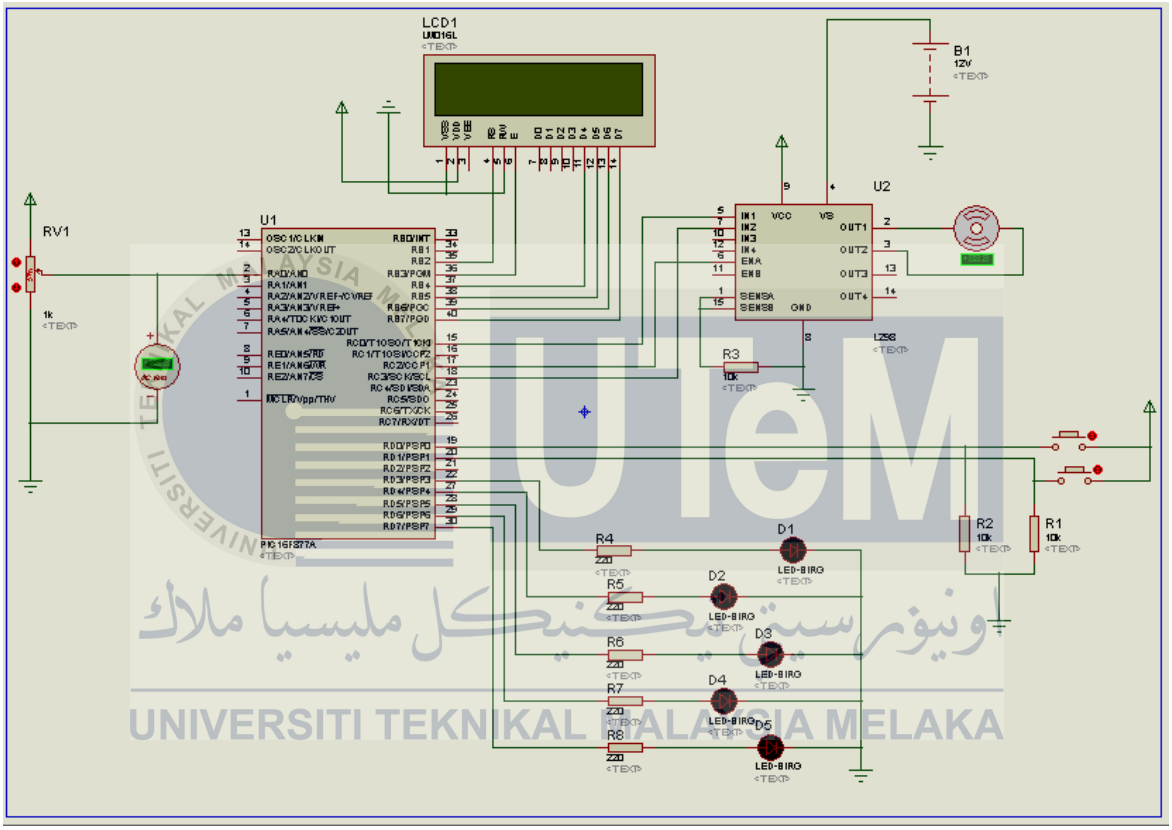


Figure 3.10 : Designing Circuit using Proteus

3.5.2 DC Motor

For the DC motor simulation, the switch plus one has been pushed and it triggers to the PIC16F877a so that the LED one light up to show the speed number one and DC motor start to energize. When the switch plus one has been pushed again and the LED indicator light up one by one until five LED to show that the DC motor has been speeded up. This simulation has proof that the program coding is following the flow chart progress.

Table 3.8: Simulation of DC motor

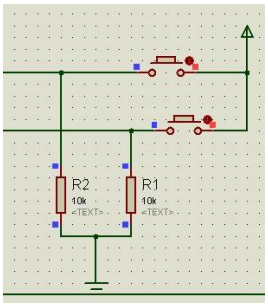
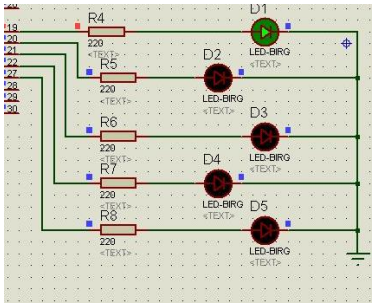
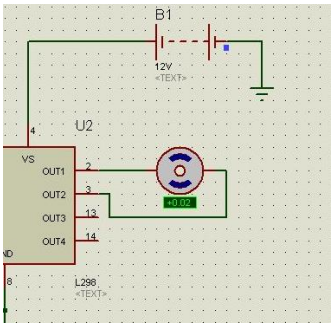
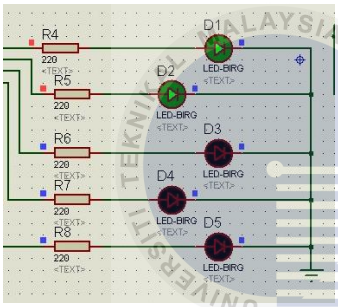
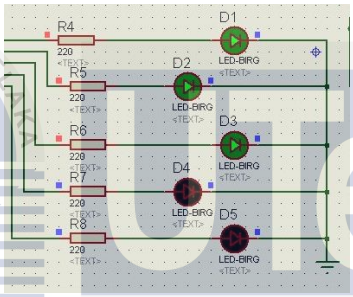
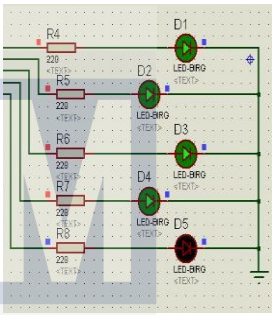
Switch	LED indicator 1	DC Motor
		

Table 3.9 : The sequences of LED indicator

LED indicator 2	LED indicator 3	LED indicator 4
		

3.5.3 Current sensor

For the current sensor simulation, variable resistor is used to make a situation of over current as in Proteus there is no element of current sensor. When varying resistor, the PIC16F877a will receive the analog output from the variable resistor and convert it into digital and display it to the LCD panel.

In this simulation, the variable resistor is set to 0% and the LCD show the value of current that absolute zero. When the variable resistor is up to 60%, the LCD show that the increasing of the amount of current. It proves that the program coding is following the flow chart progress.

Table 3.10 : Simulation of current sensor (before)

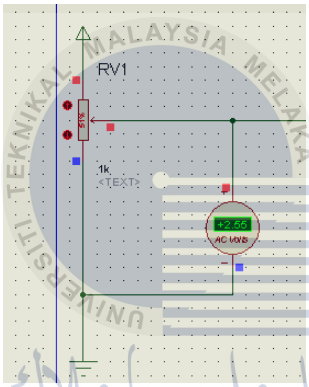
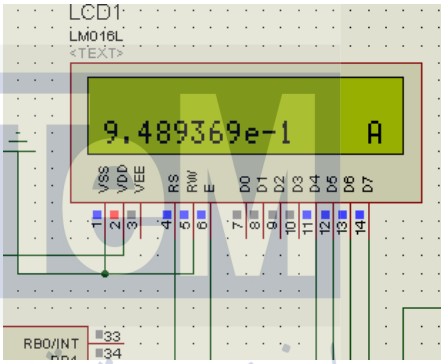
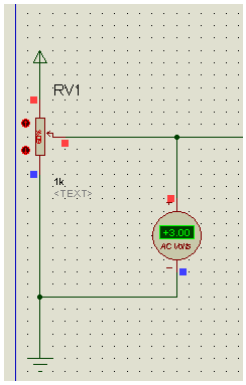
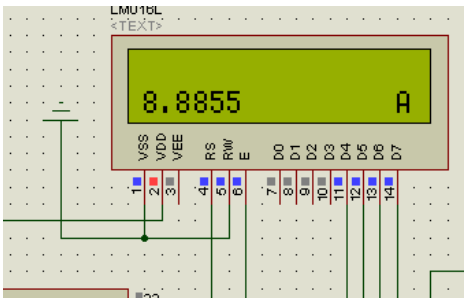
Variable resistor	LCD panel
	

Table 3.11 : Simulation of current sensor (after)

Variable resistor	LCD panel
	

CHAPTER 4

RESULT & DISCUSSIONS

4.0 Introduction

In this chapter, it will describe the finding result and discussion of this project which is the project objective that had been set. This chapter begins with the hardware implementation. The first part of hardware will explain the first hardware before it has been re-develop due to some problem occur

For the second part of this chapter, it will describe the development of the HMI which SCADA system for this RTU. Starts from the 5 principle of SCADA system which is trending, data logger, user interface, alert, event and alarm. There are also extra features that have been added up to the system as a plus point for the HMI to enhance efficiency.

For the third part, will explain the three experiments that have been carried out which is measuring current using multimeter, reading current using current sensor at LCD and reading current using current sensor at HMI. These three experiments will be compare and analysis.

4.1 Project Result

As for the result presentation, it will divide in three parts which is the hardware implementation, HMI development and experiment result for easy explanation in order of the project flow.

4.2.1 Hardware Implementation

In this project, the hardware implementation is the one of the important part. The hardware which is the load and the current sensor has been built in one microcontroller which is in one board. However, when the hardware implementation is done, there is a problem with current sensor when the LCD shows the value that not logical and unstable. Besides that the DC motor cannot rotate smoothly. From this problem, it might happen because of the microcontroller was in excess duty as it processes two unrelated main function which is the speed of the motor and the process of current sensor.

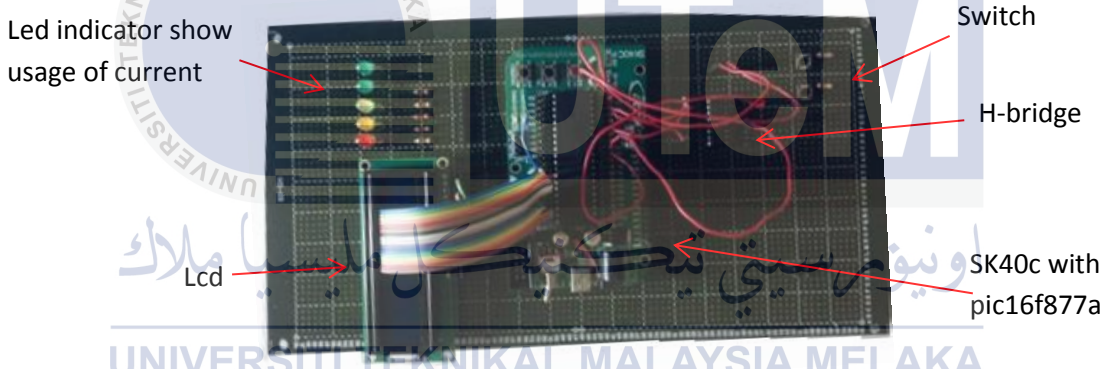


Figure 4.1: The overview of hardware

4.2.2 Re-development of hardware

Due to the problem occur, the best things that could is separate the two unrelated functions to the each microcontroller. It will decrease the microcontroller load to process the subroutine. As the hardware is divided by two microcontrollers, of course it will change the coding program.

Table 4.1 : Source code for DC motor

Source code for DC motor (Load)	Explanation

int i;
void initmain(){
trisd=0b00000011;portd=0;trisb=0b00000011;
portb=0;trisc=0;portc=0;i=0;
PWM1_Init(5000);PWM1_Start();
}
void run(int p){
portc.f0=1;portc.f3=0;
switch(p){
case(0):portd=0b00000000,PWM1_Set_Duty(0);break;
case(1):portd=0b00001111,PWM1_Set_Duty(115);break;
case(2):portd [0x0008] portd = 0PWM1_Set_Duty(135);break;
case(3):portd=0b00111111,PWM1_Set_Duty(170);break;
case(4):portd=0b01111111,PWM1_Set_Duty(220);break;
case(5):portd=0b11111111,PWM1_Set_Duty(255);break;
}
}
void main() {
initmain();
while(1){
if(portb.f0==1&&portd.f1==0){
i++;
if(i>5){
i=5;}
run(i),delay_ms(100);
}
else if(portb.f0==0&&portd.f1==1){
i--;
if(i<0){
i=0;}
run(i),delay_ms(100);
}
}
	Declare integer, subroutine and port
	Initialize PWM channel
	Conditional switch p,duty ratio of dc motor, and LED indicator.
	Condition that will change duty ratio b

Table 4.2 : Coding program for LCD and Current Sensor

Source code for LCD and Current sensor	Explanation

#include "Global.h"
#include "GSM2.0.c"
 	
#define GP08 PORTD.F2
 	
// LCD module connections
sbit LCD_RS at RB2_bit;
sbit LCD_EN at RB3_bit;
sbit LCD_D4 at RB4_bit;
sbit LCD_D5 at RB5_bit;
sbit LCD_D6 at RB6_bit;
sbit LCD_D7 at RB7_bit;
 	
sbit LCD_RS_Direction at TRISB2_bit;
sbit LCD_EN_Direction at TRISB3_bit;
sbit LCD_D4_Direction at TRISB4_bit;
sbit LCD_D5_Direction at TRISB5_bit;
sbit LCD_D6_Direction at TRISB6_bit;
sbit LCD_D7_Direction at TRISB7_bit;
// End LCD module connections
 	
unsigned int count_delay;
unsigned short spike_delay=0;
unsigned long value;
float amp, ampScale;
char txt[13];
char txt2[4];
 	
int j;
void initmain(){
 	
Lcd_Init(); // Initialize LCD
Lcd_Cmd(_LCD_CLEAR); // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF); // Cursor off
PWM1_Init(5000);PWM1_Start();
 	
}
	Calling Global and GSM2.0 file
	Define GP08 at portd.f2
	Declare LCD pins to the port
	Declare variable initial count_delay, short spike delay, long value, and single character with label txt and txt2.
	initialize LCD

```
void main(){
initmain();
TRISC=0;PORTC=0;
TRISA = 0xFF;
TRISE = 0b00001111;
ADCON1 = 0b10000000; //all pin analog


Lcd_Init();                        // Initialize LCD
Lcd_Cmd(_LCD_CLEAR);               // Clear display
Lcd_Cmd(_LCD_CURSOR_OFF);          // Cursor off



current_adc_peak = 0;
while(1){
for(count=0;count<2000;count++)
{
current_adc = adc_read(0);   // 99:1 voltage divider
if(current_adc>current_adc_peak)
{current_adc_peak=current_adc;}

if(current_adc_peak>0)
{current = (current_adc_peak ) ;
 currentrms = current;
}
else
{current=0;currentrms=0;}

floattostr(currentrms,buffer);
Lcd_Out(2,1,codetxt_to_ramtxt(Line_Clear));
Lcd_out(2,1,buffer);
Lcd_Out(2,16,"A");
```

declare adc port

equation for adc change the output from analog to digital

Print out the output on LCD

```
if(currentrms > 250)
{

 if(spike_delay > 5)
 {

 GPO8 = 1;
  PORTC.F5 = 1;
  delay_ms(2000);

  PORTC.F5 = 0;

 Lcd_Out(1,1,codetxt_to_ramtxt("OVERcurrent"));
 spike_delay=0;
 }
 spike_delay++;
}
else
{spike_delay=0;}

  if(currentrms < 99999)
{
 if(spike_delay > 5)
 {

 GPO8 = 1;
  PORTC.F5 = 1;
  delay_ms(2000);

  PORTC.F5 = 0;

 Lcd_Out(1,1,codetxt_to_ramtxt("UNDERcurrent"));
 spike_delay=0;
 }
 spike_delay++;
```

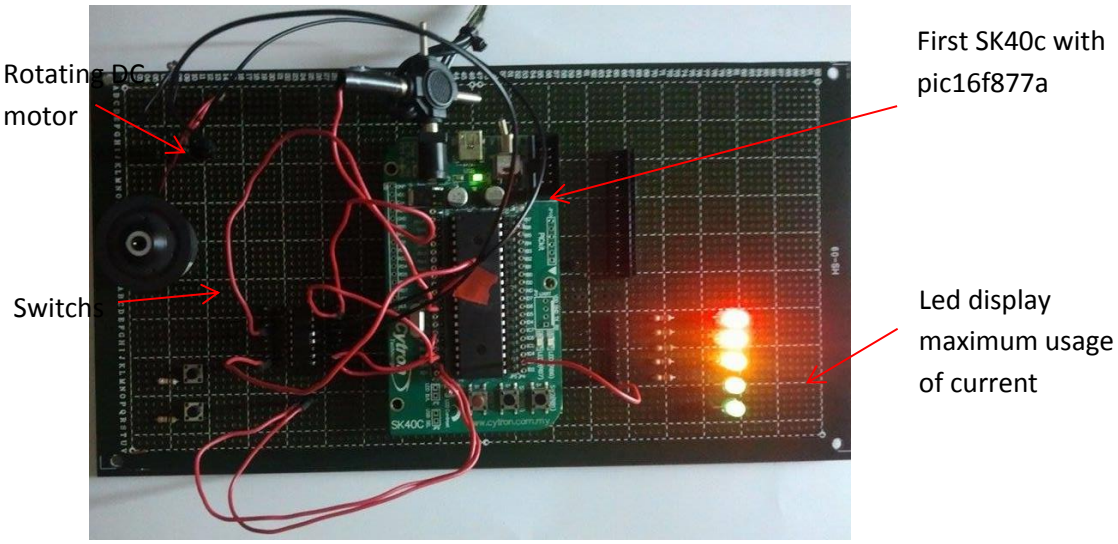


Figure 4.2 : The load board (DC Motor)

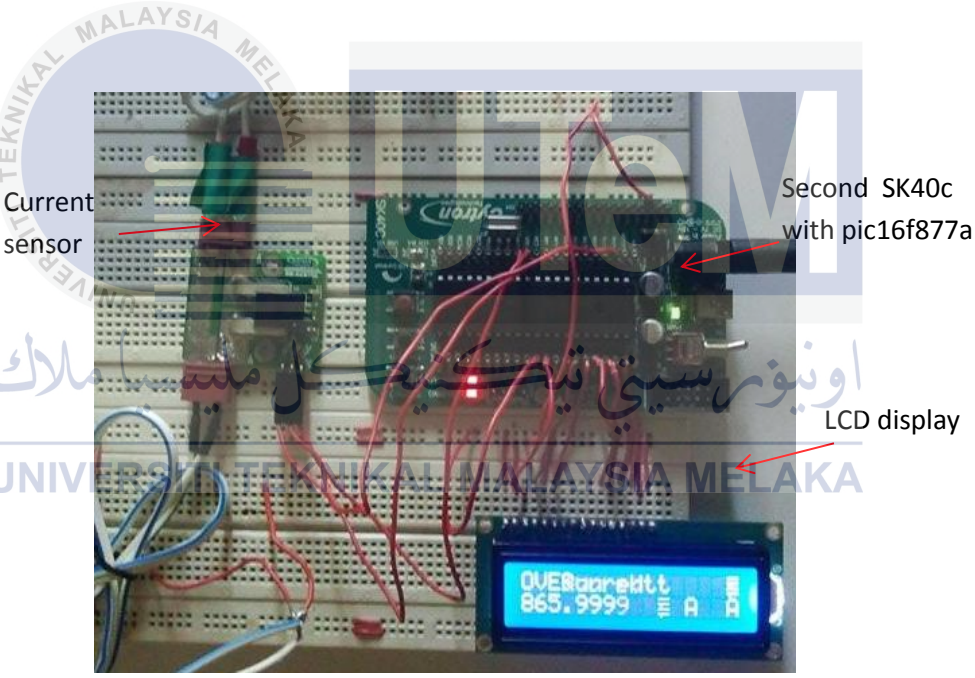


Figure 4.3 : The monitoring board (Current sensor)

4.3 HMI Developing

4.3.1 Main Form

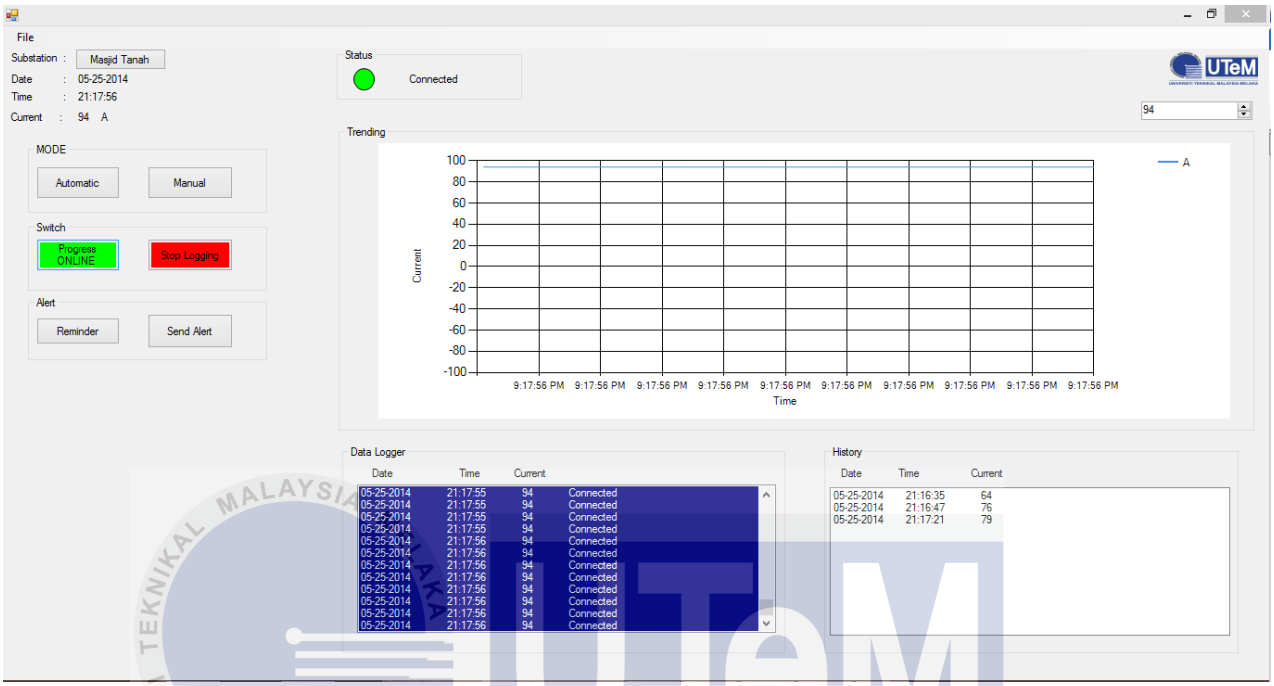


Figure 4.4 : The overview of HMI

This is the full interface view of SCADA system that has been developed. In this form, contain of substation name, map of substation, date, time, trending graph of current, data logger of current, history, controller mode, calendar reminder and alert system. This form has been decorated in a simple way for easy monitoring and friendly user.

The trending graph presents the current usage of current in every second. This trending graph gets input from the UART, it shows the current trending of DC motor.

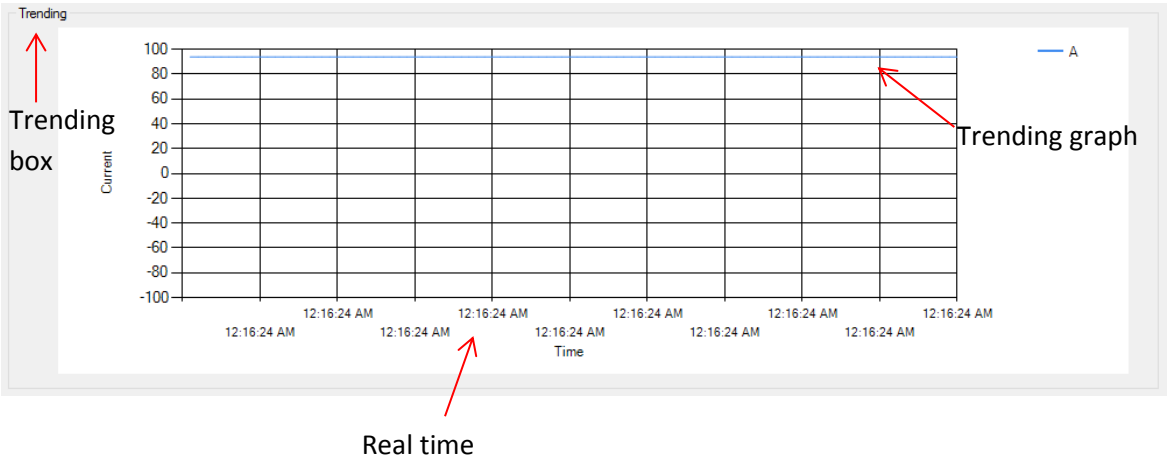


Figure 4.5 : Trending Graph

Besides trending graph, this main form also contains list box of data loggers. The purpose of a data logger is to store the information and data algorithm for further purpose inquiry. In data logger panel, it contains information of date, time, . The data logger read and writes nine times for every second. The second list box is the history of the substation. The history list box will list any fault or over current occur. It contains the date, time, the current value when fault happen.

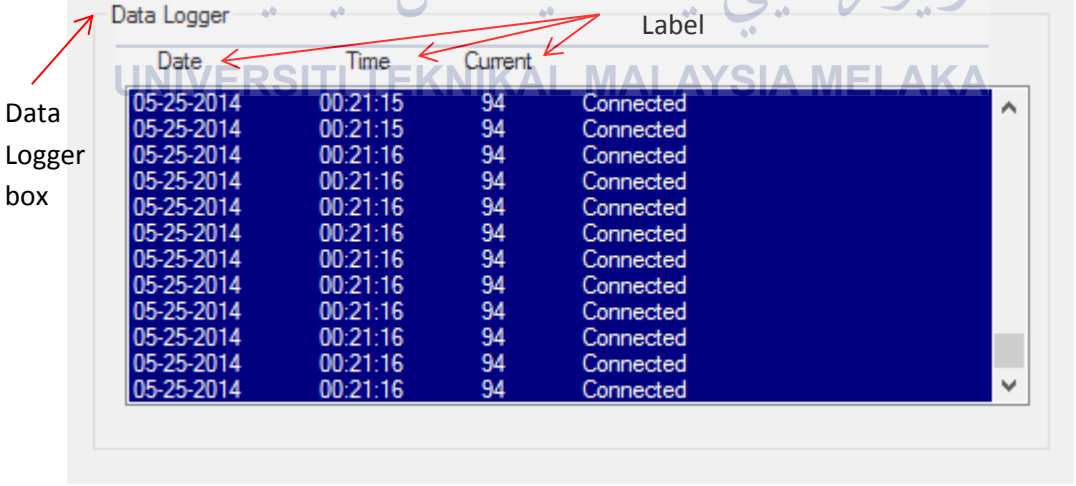


Figure 4.6 : Data Logger for current

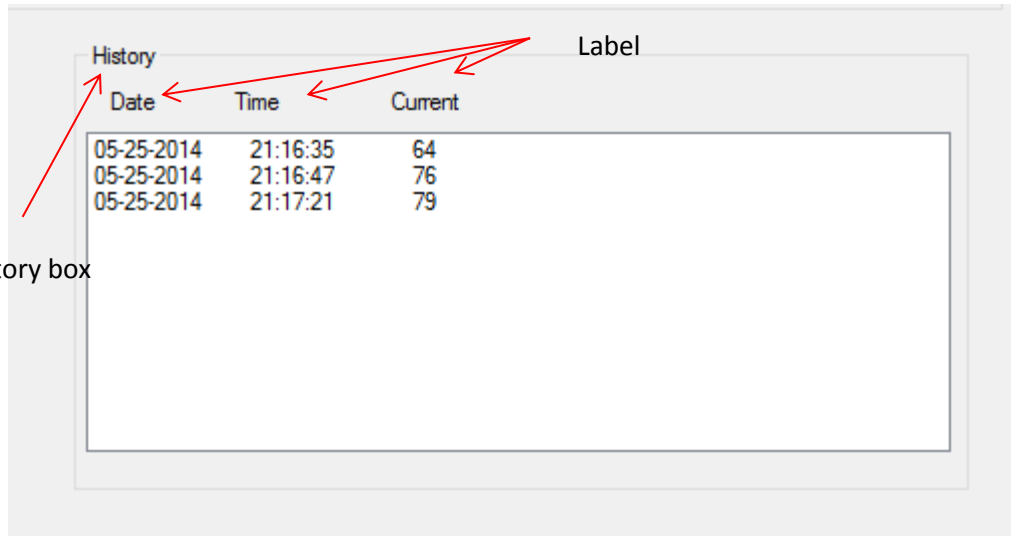


Figure 4.7: History for over current

For this HMI system, it will two modes to run the SCADA progress which is Automatic and Manual. For the Automatic mode, the system will run by itself which is trending the graph, run the data logger and when there is faulty or over current occur the system will cut the supply and stop logging. The purpose of switch button deactivate in Automatic mode is for the safety precaution to avoid any chance of careless operator that might spontaneously click the button and effect to the system.

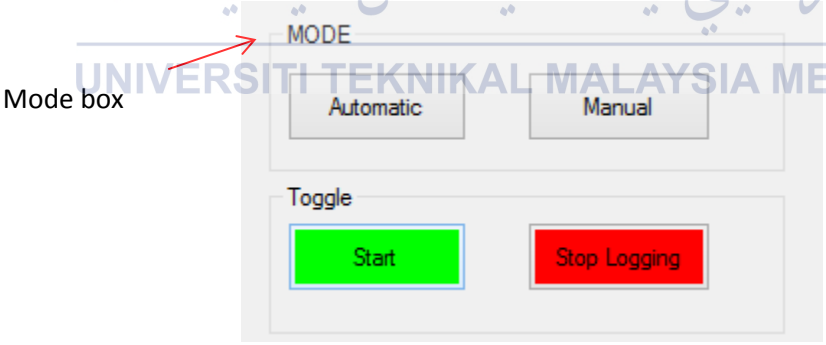


Figure 4.8 : Control Panel (Automatic Mode)

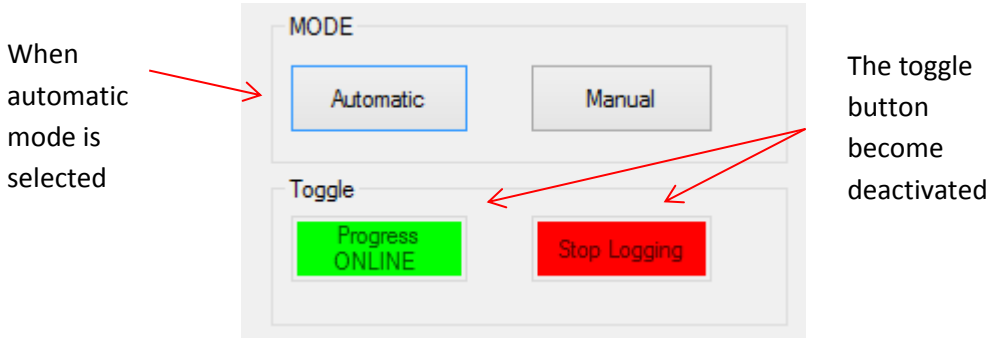


Figure 4.9 : Control Panel (Switch Deactivated)

For the manual mode, the system is still same with the automatic mode but the switch button is activated. As the switch button is activated, the system can cut off the supply and stop data logging even there is no fault or over current happen. The purpose of a manual mode is when there is an inspection or some routine cable test at the substation, it can cut off the supply for the safety.

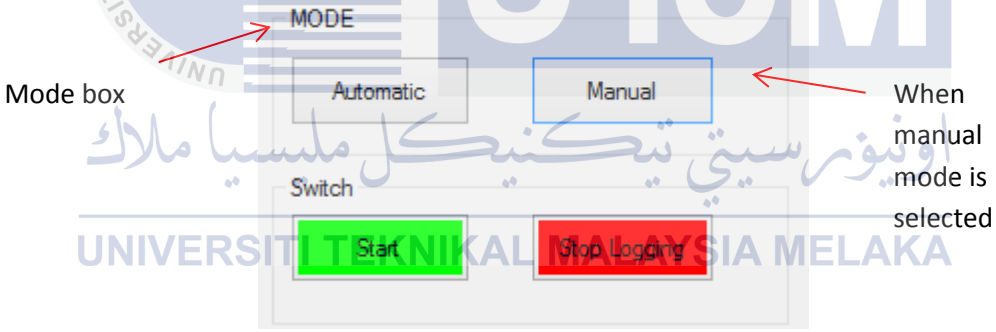


Figure 4.10 : Control Panel (Manual Mode)

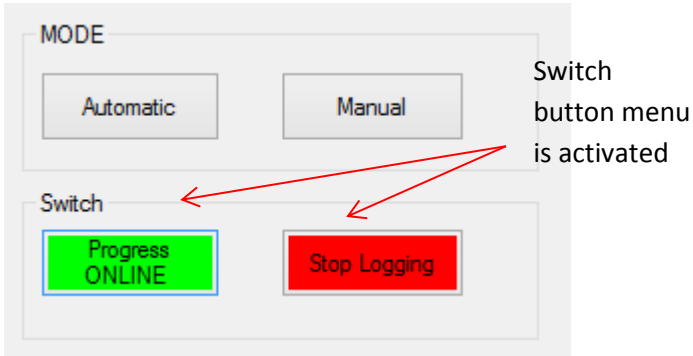
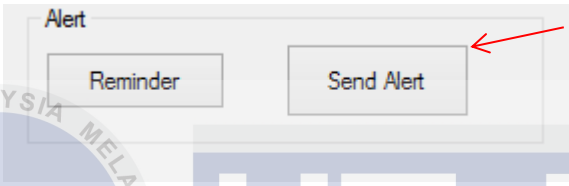


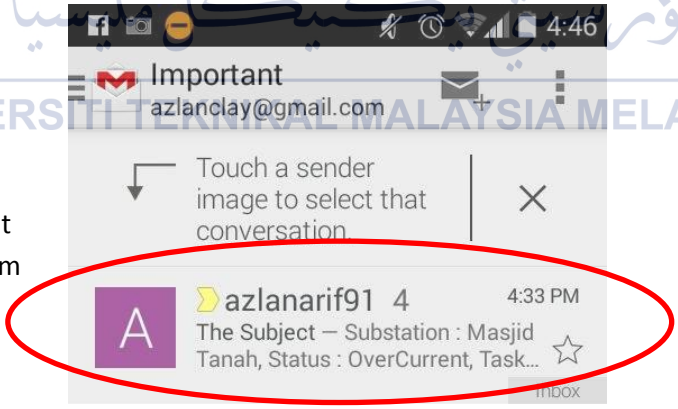
Figure 4.11 : Control Panel (Switch Activated)

As for the extra features, this HMI is equipped with alert system. As an example, when there is fault happening, the system will cut off the supply automatically, and the operator can alert to the engineer/technician by sending an information message through e-mail system. In the e-mail information message, contain of name of substation, status of substation, a task that need to be done and direction to the substation by using google maps. This feature will help to decrease time searching of which substation that need to undergo inspection. Compare to conventional way which is the engineer/technician need to search one by one which of the substation that need to be inspected, this system will increase the reliability of the system.



By clicking send alert button, the system will automatically send the e-mail message

Figure 4.12 : Alert Panel



e-mail message that sent by control room

Figure 4.13 : Receive e-mail from control room

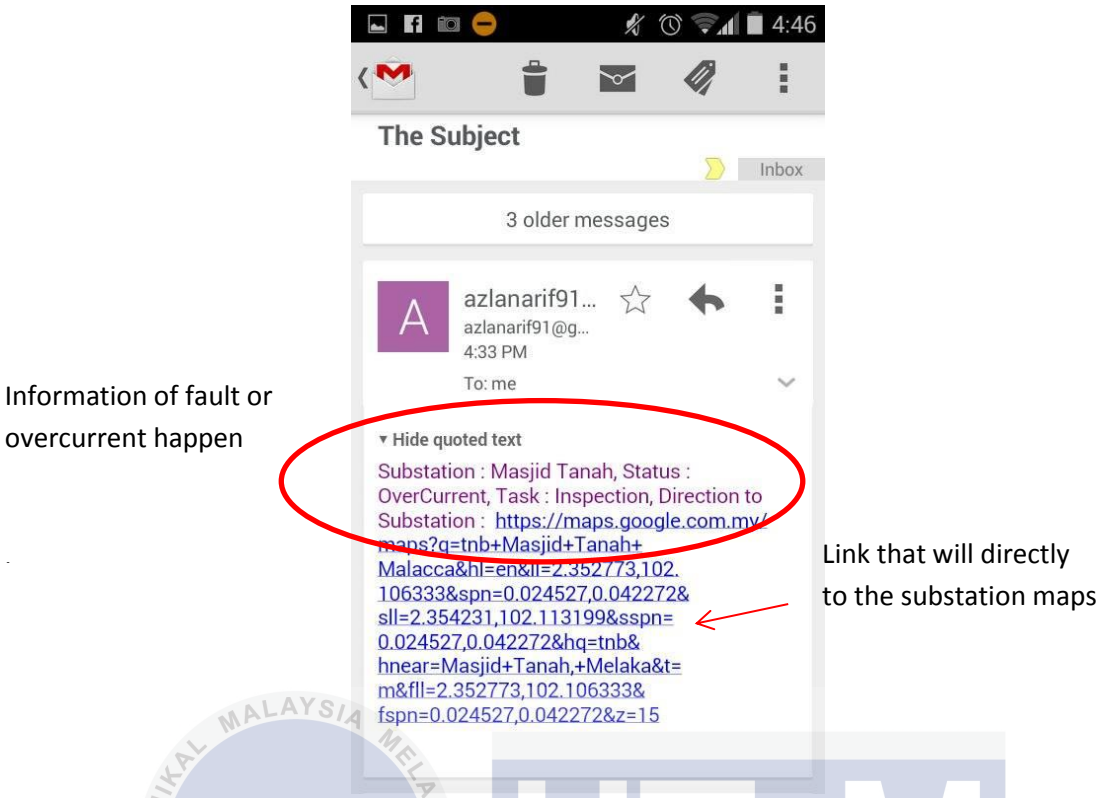


Figure 4.14 : The details about the fault and link to direction of substation

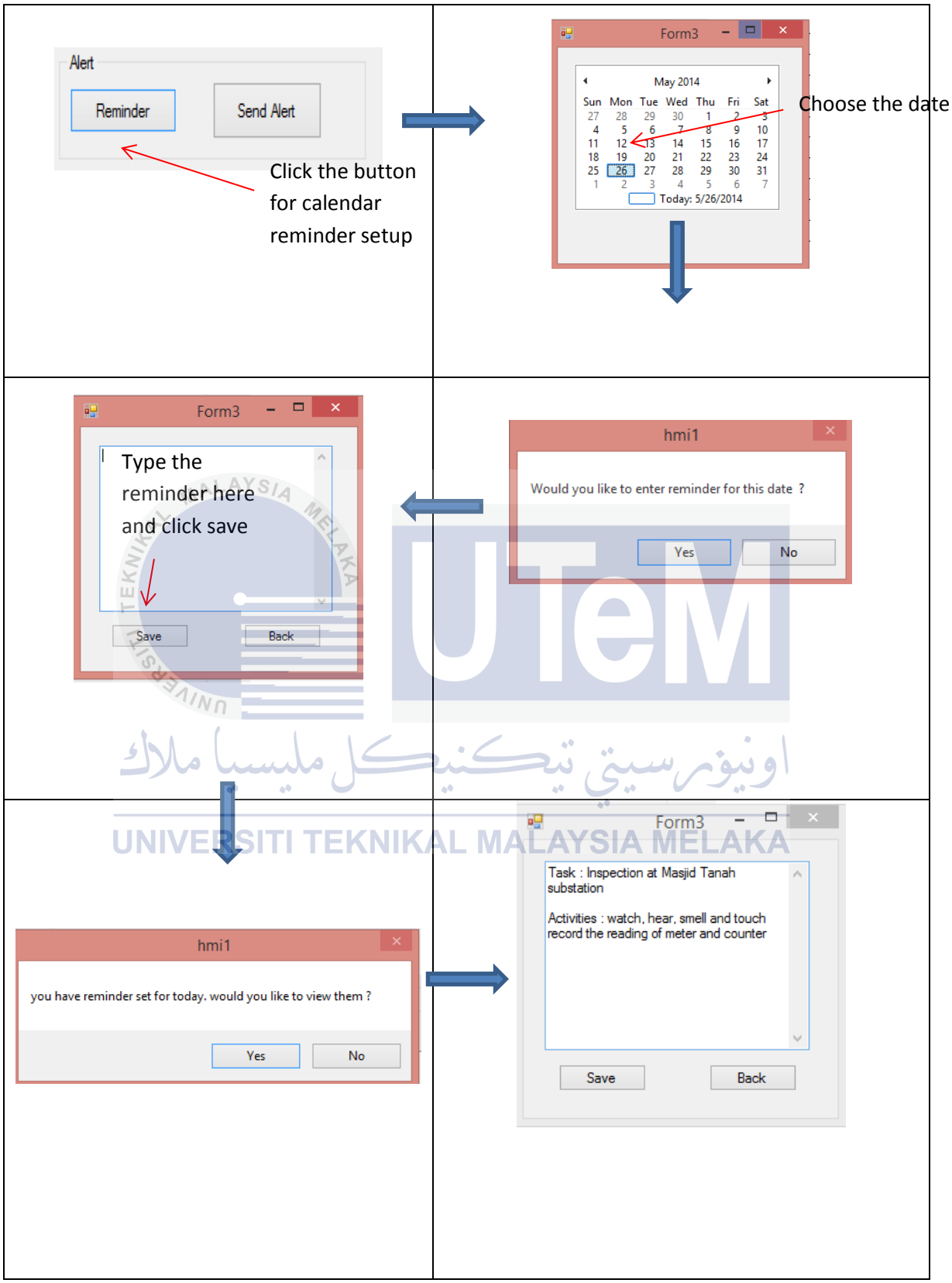
Figure 4.15: The direction to the substation

Besides that, the operator can also alert the engineer and technician to undergo some routine works at the substation to make sure the environment of the substation in a good condition. The calendar reminder feature will help the operator to remind some routine maintenance by setting the date manually. Based on Table 3.2, it shows that the maintenance type of substation and the frequency of the routine.

Table 4.3: Maintenance type

Maintenance Type	Frequency	Activities
Inspection	3 month	• Watch, hear smell and touch • Record the reading of meter and counter
Routine	6 month	• Watch, hear, smell and touch • Record the reading of meter and counter • Substation cleanliness • Change gel silica • Earth test
Condition Assessment	12 month	• Infrared thermo graphic scanning • Ultrasonic scanning
Preventive	18 month	• Watch, hear, smell and touch • Record the reading of meter and counter • Substation cleanliness • Change gel silica • Earth test • Shutdown the operation and run several test to substation tools • Cleaning and improvement

Table 4.4: How to set a calendar reminder



4.3.2 Simulation

In this part, the simulation of HMI is to prove that the system is working. By using virtual serial port driver, the port one and port two are pairing which is between Proteus and VB software. The data algorithm that produces at Proteus will be send to the VB software. When both software is connected, the proteus starts sending the data to the VB software. By using variable resistance, it will vary the output of the data and it will clear see as in Figure 4.18.

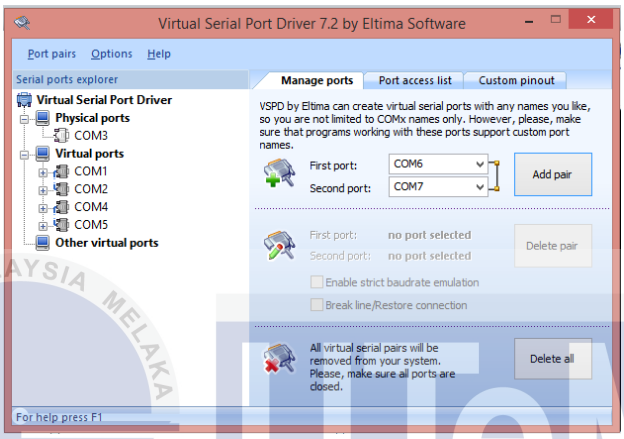


Figure 4.16 : Virtual serial port driver

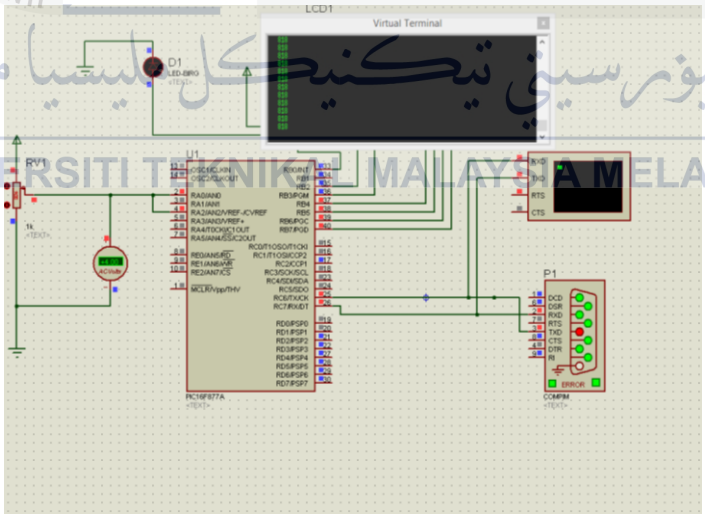


Figure 4.17: Variable data from Proteus

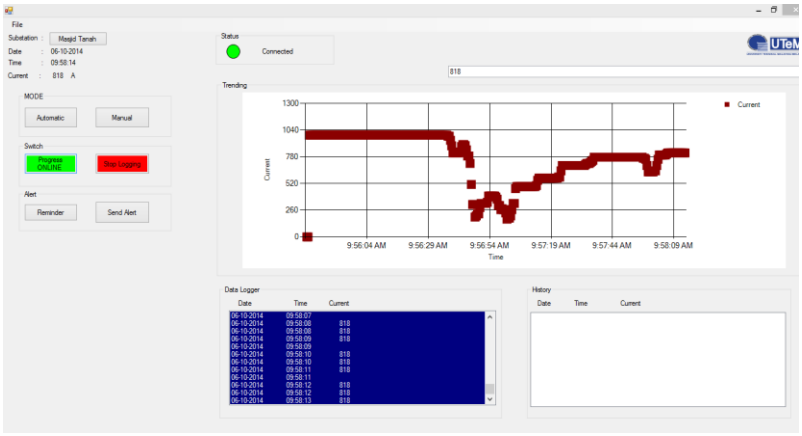
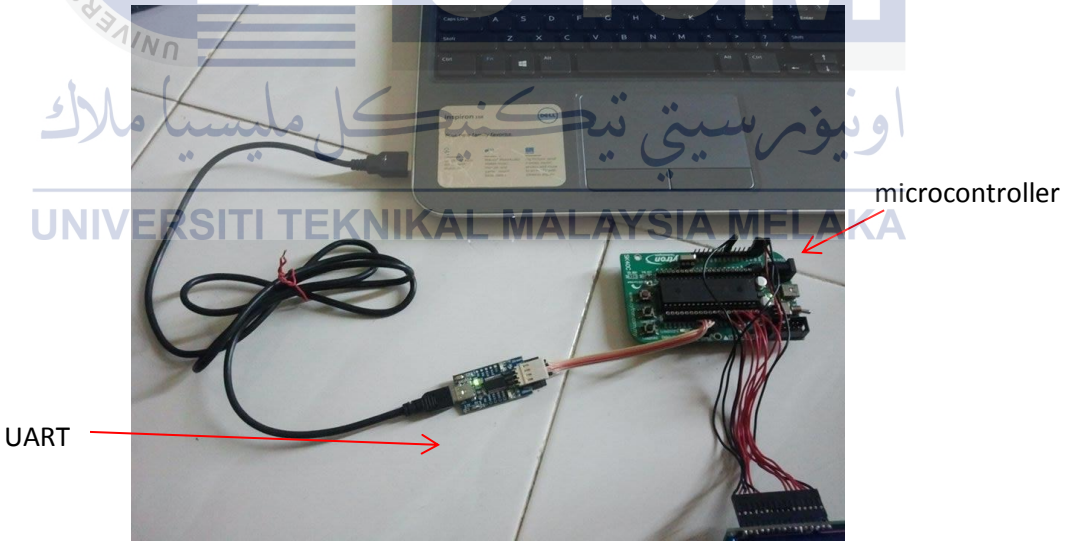


Figure 4.18: Trending graph drawing input from Proteus data

4.3.3 Sending data from hardware

This section will explain how to send the data to HMI from the microcontroller. As the current sensor reads the analog, microcontroller will convert into digital value and send to the HMI computer by using UART. The connection of UART is between microcontroller and HMI computer as in Figure 4.19 As the different speed of DC motor, it will create different algorithm data and it clearly can be seen in Figure 4.20.



re 4.19: Connection of Microcontroller to computer through UART

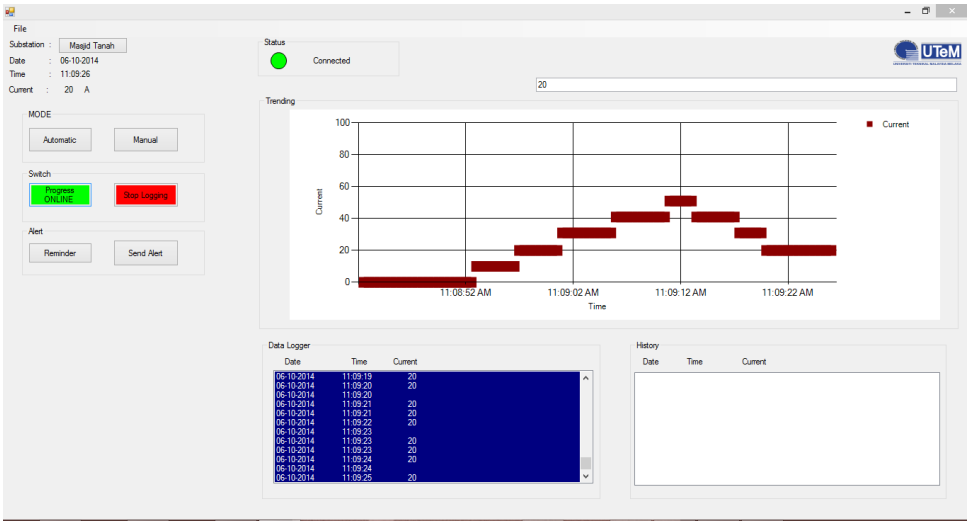


Figure 4.20: Trending graph drawing input from microcontroller data

4.4 Experiment Result and Discussion

From this project, it will be three experiments that will be needed to testify, which is measuring current by using multimeter, measuring current by using a current sensor that appear at LCD and measuring current that appear it in HMI. This test is to make sure the accuracy of the value.

Based on the Table 4.5 and Figure 4.21, it clearly sees that the increasing of DC motor, will reflect the amount current usage that had been measured using multimeter. However, the trending of the graph in Figure 4.21 is not in uniform increasing. This may happen because of the improper soldering method and due to the unstable performance of a DC motor that affect the reading..

Based on the Table 4.6 and 4.7, it clearly sees that the value that display at LCD and HMI is same. However, the both value is a bit different compare to current that measure by multimeter. It might happen because of the analog that read by current sensor less accurate than multimeter. Besides that, the converting analog to digital equation in microcontroller may affect the value of current.

From Figure 4.21, 4.22 and 4.23, the graph show that from speed zero and speed one the DC motor needs high current to rotate the motor but when changing speed one to two and so on, DC motor just need a little increasing of current. This is because DC motor needs high starting current for torque first turned on.

Table 4.5: Measuring current by using Multimeter

Speed of DC motor	Current value (mA)			Average current value (mA)
0	0	0	0	0
1	32.1	32.0	32.0	32.0333
2	35.8	35.6	35.6	35.6667
3	42.7	42.6	42.5	42.6
4	51.8	51.9	51.6	51.7667
5	56.1	56.0	56.2	56.1

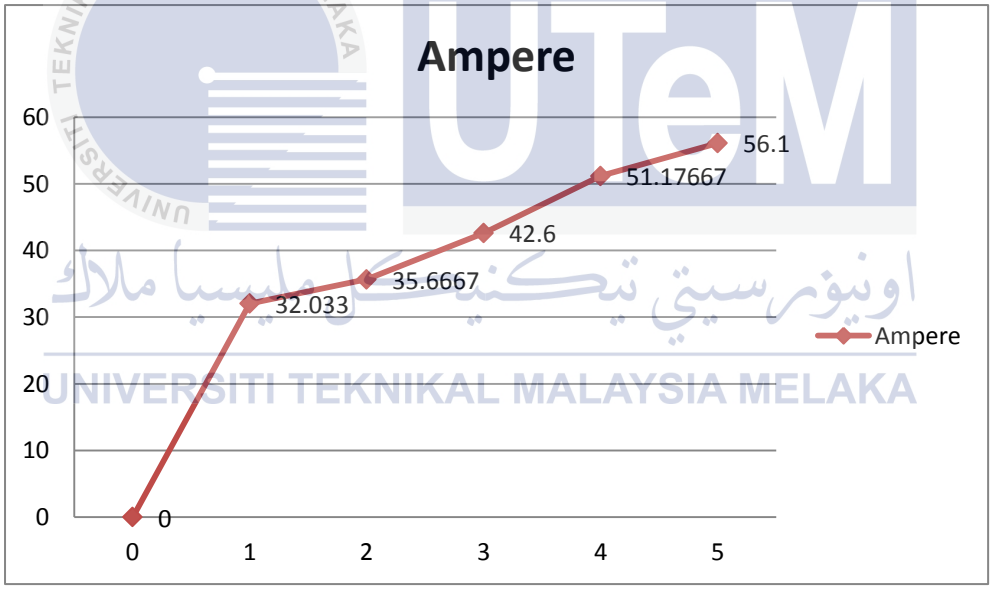


Figure 4.21: The graph of measuring current by using multimeter

Table 4.6: Measuring current by using current sensor that display at LCD

Speed of DC motor	Current value (mA)			Average current value (mA)
0	0	0	0	0
1	33	33	32	32.667
2	36	36	35	35.667
3	43	43	43	43
4	54	54	55	54.333
5	57	59	56	57.333

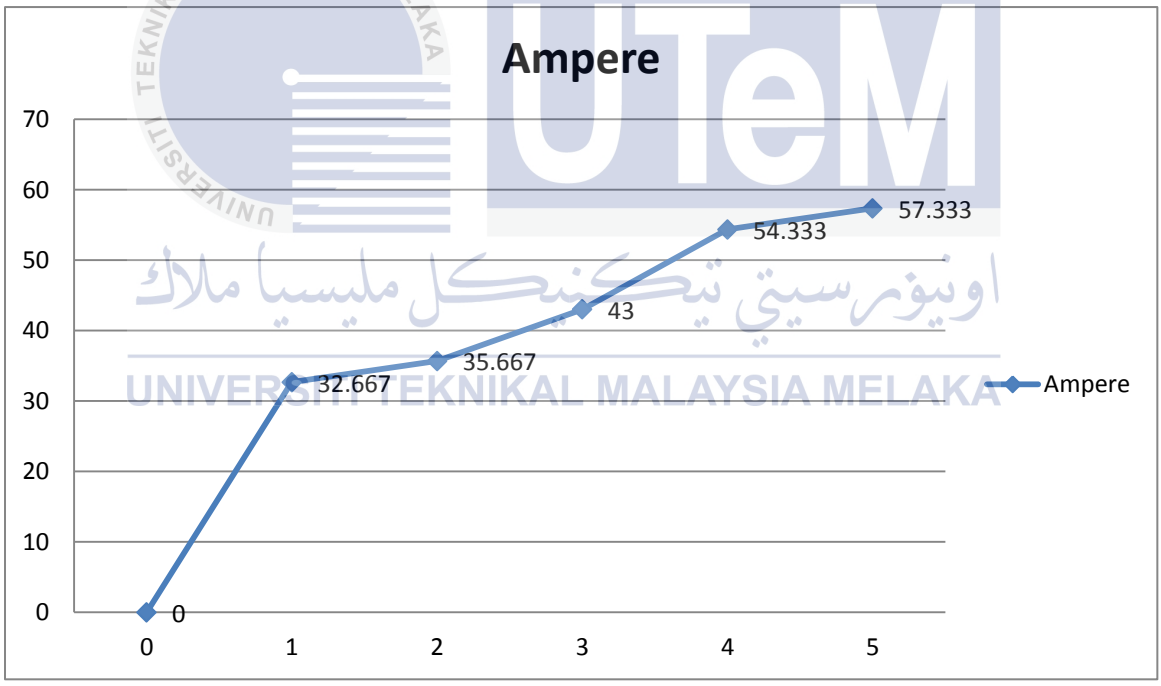


Figure 4.22: The graph of measuring current show at LCD

Table 4.7: Measuring current by using current sensor that display at HMI

Speed of DC motor	Current value (mA)			Average current value (mA)
0	0	0	0	0
1	33	33	32	32.667
2	36	36	35	35.667
3	43	43	43	43
4	54	54	55	54.333
5	57	59	56	57.333

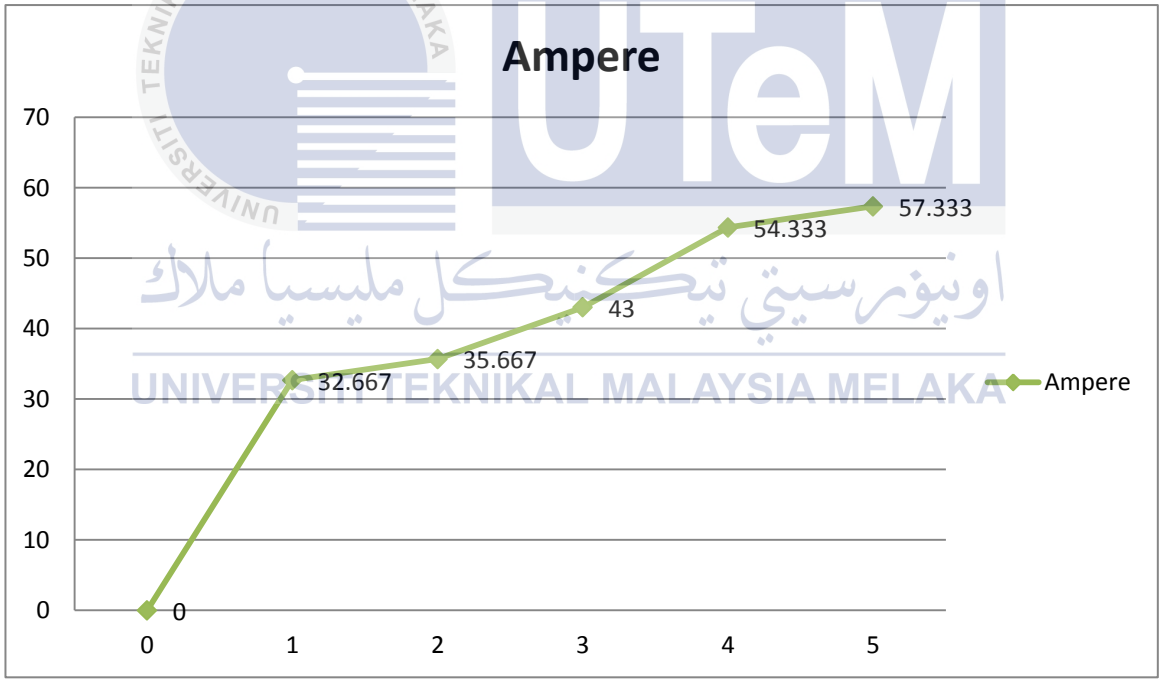


Figure 4.23 : The graph of measuring current show at HMI

Table 4.8: Comparison between three experiments

Speed of DC motor	Average Current value (mA)		
	Multimeter	LCD	HMI
0	0	0	0
1	32.0333	32.667	32.667
2	35.6667	35.667	35.667
3	42.6	43	43
4	51.7667	54.333	54.333
5	56.1	57.333	57.333

CHAPTER 5

CONCLUSIONS& RECOMMENDATIONS

As for the conclusions, this project which is Human Machine Interface (HMI) Development for Distribution Automation System (DAS) will help the power supply company to monitor and control the distribution system. The objective of this project is to implement data algorithm of current to give input to the HMI for monitoring purpose. Besides that, this project also implemented that the HMI can control the power supply to cut off if there is a fault or over current. As this system work fine and achieved the objective stated earlier, this project will make the automation of distribution system very efficient and effective. It may increase the reliability of the system and less cost of maintenance and operation For the future, this project will be implemented in iRTU which is project under supervisor Datuk Prof Dr Mohd Ruddin Abd Ghani. iRTU project has won gold medal in three different exhibition. This project will be a plus point to the system iRTU for the next exhibition.

REFFERENCES

[1] S. Luan, J. Teng, and C. Chen, "Development of a Novel Fault Indicator for Distribution Automation,"IJIEE., vol. 1, no. 2, pp. 105–109, 2011.

[2] I.-H. Lim, T. S. Sidhu, M.-S. Choi, S.-J. Lee, S. Hong, S.-I. Lim, and S.-W. Lee, "Design and Implementation of Multiagent-Based Distributed Restoration System in DAS," *IEEE Trans. Power Deliv.*, vol. 28, no. 2, pp. 585–593, Apr. 2013.

[3] N. Hashimoto, S. Kato, and S. Tsugawa, "HMI for elderly drivers in an off-vehicle parking assistance system," *2008 IEEE Intell. Veh. Symp.*, pp. 482–487, Jun. 2008.

[4] I. Lim, T. S. Sidhu, M. S. Choi, S. J. Lee, and B. N. Ha, "An Optimal Composition and Placement of Automatic Switches in DAS," *IEEE Trans. Power Deliv.*, vol. 28, no. 3, pp. 1474–1482, Jul. 2013.

[5] H. K. Targhi, "Accuracy Evaluation of Delivered Measurements to HMI in a Real SCADA Automation System,"Instrumentation Control and Automation (ICA), 2011 2nd International Conference. November, pp. 279–283, 2011.

[6] I. Control, "Distributed Power System Automation With,"IEEE Transactions., vol. 41, no.1, pp. 1–12, 2010.

[7] S. Lindsey, C. Raghavendra, F. Member, K. M. Sivalingam, and S. Member, "Data Gathering Algorithms in Sensor Networks Using Energy Metrics," vol. 13, no. 9, pp. 924–935, 2002.

[8] J. B. Weber, "Applying Visual Basic for Human Machine Interface Applications," Instrument Society of America .1999.

[9] W. N. S. E. W. Jusoh, "Remote Terminal Unit (RTU) Hardware Design and Implementation Efficient in Different Application," no. June, pp. 570–573, 2013.

Appendix A

Visual Basic Form 1

```
Imports System.Net.Mail


PublicClassForm1


Dim x AsInteger = 0
Dim array(100) AsInteger
PrivateSub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
HandlesMyBase.Load
        Label5.Text = TimeString
        Label6.Text = DateString
        Timer2.Start()

        Button1.Text = "Start"
        NumericUpDown1.Minimum = 1
        NumericUpDown1.Maximum = 10000
        Chart1.Series.Clear()
        Chart1.Series.Add("A")
        Chart1.ChartAreas("ChartArea1").AxisX.Minimum = 0
        Chart1.ChartAreas("ChartArea1").AxisX.Maximum = 100
        Chart1.ChartAreas("ChartArea1").AxisX.Interval = 10
        Chart1.ChartAreas("ChartArea1").AxisX.Title = "Time"
        Chart1.ChartAreas("ChartArea1").AxisY.Minimum = -100
        Chart1.ChartAreas("ChartArea1").AxisY.Maximum = 100
        Chart1.ChartAreas("ChartArea1").AxisY.Interval = 20
        Chart1.ChartAreas("ChartArea1").AxisY.Title = "Current"
        Chart1.Series("A").ChartType =
DataVisualization.Charting.SeriesChartType.Line
For i AsInteger = 0 To 100
            array(i) = 0
            Chart1.Series("A").Points.AddXY(i, array(i))
Next



EndSub

PrivateSub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        ListBox1.TopIndex = ListBox1.Items.Count - 1
        ListBox1.ScrollAlwaysVisible = False

If NumericUpDown1.Text <= 80 Then
            ListBox2.Items.Add(Label6.Text + "        " + Label5.Text + "
" + NumericUpDown1.Text)
EndIf


        ListBox1.Items.Add(Label6.Text + "             " + Label5.Text + "
" + NumericUpDown1.Text + "             " + Label1.Text)
        Label5.Text = TimeString
        Label9.Text = NumericUpDown1.Value
```

```
        x = x + 1
For i AsInteger = 0 To 99
            array(i) = array(i + 1)
Next
        array(100) = 100 * (Math.Sin(0.1 * x))
        Chart1.Series("A").Points.Clear()
For i AsInteger = 0 To 100
            Chart1.Series("A").Points.AddXY(DateTime.Now.ToLongTimeString(),
NumericUpDown1.Value)
Next
If NumericUpDown1.Value <= 80 Then Timer1.Stop()
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
        Button1.Text = "Progress ONLINE "
        OvalShape1.Visible = True
        OvalShape3.Visible = False
        Label1.Visible = True
        Label10.Visible = False
        Timer1.Interval = NumericUpDown1.Value
        Timer1.Start()



EndSub

PrivateSub NumericUpDown1_ValueChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles NumericUpDown1.ValueChanged
        Timer1.Interval = NumericUpDown1.Value
EndSub

PrivateSub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer2.Tick
        Label5.Text = TimeString

EndSub

PrivateSub Label5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label5.Click

EndSub

PrivateSub CloseToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles CloseToolStripMenuItem.Click
End
EndSub

PrivateSub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles PictureBox1.Click
        System.Diagnostics.Process.Start("http://www.utem.edu.my/web2012/")

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        Button2.Text = "Stop Logging"
        Button1.Text = "Continue Logging"
        OvalShape1.Visible = False
        OvalShape3.Visible = True
        Label1.Visible = False
```

```
        Label10.Visible = True
        Timer1.Stop()
EndSub

PrivateSub Label6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label6.Click

EndSub

PrivateSub Button3_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button3.Click
Dim EmailMessage AsNewMailMessage()

Try
            EmailMessage.From = NewMailAddress("azlanarif91@gmail.com")
            EmailMessage.To.Add("azlanclay@gmail.com")
            EmailMessage.Subject = "The Subject"
            EmailMessage.Body = "Substation : Masjid Tanah, Status : OverCurrent,
Task : Inspection, Direction to Substation :
https://maps.google.com.my/maps?q=tnb+Masjid+Tanah+Malacca&hl=en&ll=2.352773,102.10
6333&spn=0.024527,0.042272&sll=2.354231,102.113199&sspn=0.024527,0.042272&hq=tnb&hn
ear=Masjid+Tanah,+Melaka&t=m&fll=2.352773,102.106333&fspn=0.024527,0.042272&z=15"
Dim SMTP AsNewSmtpClient("smtp.gmail.com")
            SMTP.Port = 587
            SMTP.EnableSsl = True
            SMTP.Credentials = New
System.Net.NetworkCredential("azlanarif91@gmail.com", "azlanur13")
            SMTP.Send(EmailMessage)
Catch ex AsException

EndTry
EndSub

PrivateSub Form1_Activated(ByVal sender As System.Object, ByVal e As
System.EventArgs) HandlesMyBase.Activated
Form2.Hide()
Form3.Hide()
EndSub

PrivateSub Label8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label8.Click

EndSub



PrivateSub Button4_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button4.Click
Form2.Show()
EndSub

PrivateSub Button5_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button5.Click
Form3.Show()
EndSub

PrivateSub Button6_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button6.Click
        Timer1.Start()
        Button1.Enabled = False
        Button2.Enabled = False
```

```
        Button1.Text = "Progress ONLINE "
'   If NumericUpDown1.Value = 80 Then Timer1.Stop()
EndSub

PrivateSub Chart1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Chart1.Click

EndSub

PrivateSub ListBox2_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ListBox2.SelectedIndexChanged

EndSub

PrivateSub Button7_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button7.Click
        Button1.Enabled = True
        Button2.Enabled = True
EndSub

PrivateSub GroupBox6_Enter(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles GroupBox6.Enter

EndSub

PrivateSub Label1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label1.Click

EndSub

PrivateSub SplitContainer1_Panel1_Paint(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles SplitContainer1.Panel1.Paint

EndSub

PrivateSub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ListBox1.SelectedIndexChanged

EndSub

PrivateSub Label2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label2.Click

EndSub

PrivateSub Label9_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Label9.Click

EndSub


PrivateSub Label18_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)

EndSub
EndClass
```

Appendix B

Visual Basic Form 2

```
PublicClassForm2
Dim Loc AsString
PrivateSub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
HandlesMyBase.Load
        Loc = "http://maps.googleapis.com/maps/api/staticmap?center="&"Masjid Tanah
Melaka Malaysia"&"&zoom=15&size=560x560&sensor=false"
        WebBrowser1.Navigate(Loc)
EndSub
EndClass
```

Appendix C

Visual Basic 3

```
Imports System.IO
PublicClassForm3

Dim m AsMsgBoxResult
Dim t AsString


PrivateSub MonthCalendar1_DateSelected(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DateRangeEventArgs) Handles MonthCalendar1.DateSelected
        t = MonthCalendar1.SelectionRange.Start.Month.ToString &
MonthCalendar1.SelectionRange.Start.Day.ToString

Try
IfFile.Exists(t &".txt") = TrueThen
                MonthCalendar1.Enabled = False
                MonthCalendar1.Hide()
                TextBox1.Enabled = True
                TextBox1.Show()
                Button1.Enabled = True
                Button1.Show()
                Button2.Enabled = True
                Button2.Show()
                TextBox1.Text = File.ReadAllText(t &".txt")
Else
                m = MsgBox("Would you like to enter reminder for this date  ?",
MsgBoxStyle.YesNo)
If m = MsgBoxResult.Yes Then
                    MonthCalendar1.Enabled = False
                    MonthCalendar1.Hide()
                    TextBox1.Enabled = True
                    TextBox1.Show()
                    TextBox1.Text = ""
                    Button1.Enabled = True
                    Button1.Show()
                    Button2.Enabled = True
                    Button2.Show()

EndIf
EndIf
Catch ex AsException
            MsgBox(ex.Message)
EndTry

EndSub

PrivateSub Button2_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button2.Click
        TextBox1.Enabled = False
        TextBox1.Hide()
        Button1.Enabled = False
        Button1.Hide()
        Button2.Enabled = False
        Button2.Hide()
        MonthCalendar1.Enabled = True
        MonthCalendar1.Show()
```

```
EndSub

PrivateSub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Button1.Click
Try
If TextBox1.Text = ""Then
IfFile.Exists(t &".txt") = TrueThen
File.Delete(t &".txt")
EndIf
EndIf
If TextBox1.Text.Length > 0 Then
File.WriteAllText(t &".txt", TextBox1.Text)
EndIf
Catch ex AsException
            MsgBox(ex.Message)
EndTry
EndSub

PrivateSub Form3_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
HandlesMyBase.Load
Dim m1 AsMsgBoxResult
        t = MonthCalendar1.SelectionRange.Start.Month.ToString &
MonthCalendar1.SelectionRange.Start.Day.ToString

IfDate.Today = MonthCalendar1.TodayDate AndFile.Exists(t &".txt") = TrueThen
            m1 = MsgBox("you have reminder set for today. would you like to view
them ?", MsgBoxStyle.YesNo)
If m1 = MsgBoxResult.Yes Then
                MonthCalendar1.Enabled = False
                MonthCalendar1.Hide()
                TextBox1.Enabled = True
                TextBox1.Show()
                Button1.Enabled = True
                Button1.Show()
                Button2.Enabled = True
                Button2.Show()
                TextBox1.Text = File.ReadAllText(t &".txt")

EndIf

EndIf



EndSub
EndClass
```