DEVELOPMENT OF GRAPHICAL USER INTERFACE (GUI) OF CAN BUS DASHBOARD

Nur Sakinah Binti Mohd Ismail

Bachelor of Electrical Engineering
(Control, Instrumentation & Automation)

June 2014

" I hereby declare that I have read through this report entitle "Development Of A Graphical User Interface (GUI) Of Controller Area Network (CAN) Bus Dashboard" and found that it has comply the partial fulfilment for awarding the degree of Bachelor of Electrical Engineering (Control, Instrumentation and Automation)"

Signature : ....

Supervisor's Name : Ahmad Fairuz Bin Muhammad Amin

Due : ....

DEVELOPMENT OF A GRAPHICAL USER INTERFACE (GUI) OF CAR DASHBOARD FROM CONTROLLER AREA NETWORK (CAN) BUS

NUR SAKINAH BINTI MOHD ISMAIL

A Report submitted in partial fulfilment of the requirements for the degree of Electrical Engineering (Control, Instrumentation and Automation)

Faculty of Electrical Engineering

UNIVERSITI TEKNIKAL MALAYSIA MELAKA

2014

I declare that this report entitle "Development Of A Graphical User Interface (GUI) Of Controller Area Network (CAN) Bus Dashboard" 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 : Nur Sakinah Binti Mohd Ismail

Date : ....

To My beloved Mother, Mrs. Salha Binti Sabar and My beloved father, Mr. Mohd Ismail Bin Roslan

ACKNOWLEDGEMENT

In the name of Allah, the Most Gracious and the Most Merciful. Alhamdulillah, all praises to Allah for the strengths and His blessing for me completing this Final Year Project II report.

I would like to take this golden opportunity to express my deepest appreciation to my final year project supervisor, Mr. Ahmad Fairuz Bin Muhammad Amin for being a dedicated lecturer guiding me through this project to run smoothly. This project cannot be completed and perfected if without the valuable suggestion and useful information from my supervisor. His patience and enthusiastic in guided me through this final year project have gave me a great courage in completing this project.

Furthermore, I would like to thank my panels Mr.Mohd Safirin Bin Karis and Dr. Lim Kim Chuan for their guidance and provide new idea so that this project can be done perfectly. I would like to express my deepest love and gratitude to my beloved parents, Mr. Mohd Ismail Bin Hj. Roslan and Mrs. Salha Binti Sabar giving me unlimited encouragement especially in moral and investment.

Lastly, thanks to all my friends, and my beloved Universiti Teknikal Malaysia Melaka, UTeM, and to those who indirectly contributed during my Final Year Project II. Your kindness will never be forgotten.

ABSTRACT

Nowadays, most of display system had been using digital and even Graphical User Interface (GUI) to provide the best application system. Purpose of this project is to develop a GUI display for Controller Area Network (CAN) dashboard. In this project, Controller Area Network (CAN) dashboard is chosen to improve their display system application. The CAN was used to control the performance of dashboard. Meanwhile, GUI was designed to display the performance of parameter. Therefore, CAN Bus is specialized to connect interaction between the dashboard and the input of circuit.

CAN is a serial network that was originally designed for the automotive industry. It was primarily used in embedded system and consists of two wires, half duplex, high speed network system. The CAN systems are well suited for high speed application using short wire. GUI is type of user interface allow interaction with electronic devices through graphical icons and visual indicator. This application is designed to make the program interactive and user friendly. By using GUI, user can be free from learning complex command languages.

Therefore, in this project required coding to display GUI Car Dashboard and CAN Bus network to connect GUI with micro board. In Chapter 1 explains on the introduction, objectives, and scope regarding this project. Then Chapter 2 is literature review to explain some theory involved and previous papers according to these theories. There are more than 10 previous research paper had been referred to complete this project. In Chapter 3 will explain about methodology of this project divide into 2 parts which is software part and hardware parts. Chapter 4 shows the results and analysis based on the results. Lastly, Chapter 5 explains about conclusion and recommendation in future towards this project.

ABSTRAK

Pada masa kini, kebanyakan system paparan telahmenggunakansistem digital dan juga grafik (GUI) untuk menyediakan system aplikasi yang terbaik. Tujuan projek ini adalah untuk membangunkan satupaparan GUI bagi Kawasan Pengawal Rangkaian (CAN) papanpemuka . Dalamprojekini , Kawasan Pengawal Rangkaian (CAN) papan pemuka makhluk memilih untuk memperbaiki aplikasi system paparanmereka. CAN ini digunakan untuk mengawal prestasi papan pemuka . Sementara itu, GUI telah direka untuk memaparkan prestasi parameter. Oleh itu, CAN Bus adalah khusus untuk menyambung interaksi antara papan pemuka dan input litar.

CAN adalah rangkaian siri yang pada asalnya direka untuk industriautomotif. Iadi gunakan terutamanya dalam system terbenam dan terdiri daripada dua wayar , separuh dupleks , system rangkaian kelajuan tinggi. Sistem CAN baik sesuai untuk aplikasi berkelajuan tinggi menggunakan wayar pendek. GUI adalah jenis antara muka pengguna membolehkan interaksi dengan peranti elektronik melalui ikon grafik dan penunjuk visual . Permohonan ini direka untuk membuat program interaktif dan mesra pengguna. Dengan menggunakan GUI, pengguna boleh bebas daripada mempelajari bahasa arahan kompleks.

Oleh itu, dalam projek ini diperlukan pengekodan untuk memaparkan GUI Dashboard Kereta dan CAN rangkaian Bas menyambung GUI dengan papan mikro. Dalam bab 1 akan menerangkan tentang pengenalan , objektif, dan skop mengenai projekini. Kemudian bab 2 adalah kajian Sastera untuk menjelaskan beberapa teori yang terlibat dan kertas sebelumnya mengikut teori-teori ini . Terdapat lebih daripada 10 kertas penyelidikan sebelumnya telah pun dirujuk kepada menyiapkanprojekini. Dalam bab 3 akan menerangkan mengenai metodologi jurang projek ini kepada 2 bahagian yang merupakan sebahagian perisian dan bahagian-bahagian perkakasan. Bab 4 akan menunjukkan hasil dan beberapa analisis mengenai keputusan. Akhir sekali , bab 5 menjelaskan tentang kesimpulan dan cadangan pada masa akan dating kearah projek ini.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GUI	-	Graphical User Interface
CAN	-	Controller Area Network
VB	-	Microsoft Visual Basic 2010 Express
ECU	-	Electronics Control Units
RPM	-	Revolutions per Minute
PC	-	Personal Computer
Mbps	-	Megabits per Second
CCS	-	Centrally Controlled System
FFT	-	Fast Fourier Transform
LCD	-	Liquid Crystal Display
UPS	-	Uninterrupted Power Supply
API	-	Microsoft Windows Application
LAN	-	Local Area Network
PIC	-	Pharmaceutical Inspection Convention
Arb.ID	-	Arbitration ID
SMS	-	Short Message Service
GPS	-	Global Positioning System

LIST OF APPENDICS

CHAPTER 1

INTRODUCTION

1.0 Overview

CAN was developed by Robert Bosch GmbH, Germany in 1986 for in vehicle network. In the past, automotive industry connected electronic control unit (ECU) by using wiring system. Then, bulky wire harnesses that were heavy and expensive are used in vehicles. Therefore, they are requested to develop a communication system between three electronic control units (ECU) in vehicle by Mercedes.

In 1987, first CAN silicon was developed. It is a high integrity- serial bus system for networking of intelligent devices as the standard vehicle in the network. In 1993, CAN become the international standard known as ISO 11898. Next, in 1994 higher level protocols have been standardized on CAN such as CAN open and Device net [28].

GUI was designed by Xerox Corporation's Palo Alto Research Centre in 1970. But due to CPU power and good quality monitor, GUI became expensive product. It makes GUI had slow acceptance from user. Therefore, Steve Jobs, co-founder of Apple decide to incorporate GUI system in his company computer.

Every year, GUI applications has increased and required a high demand from the market because of its criteria which is make the system easy to learn. Besides that, it had more convenient communication in human computer interaction and allows the developer to create graphics and audible display method. Moreover, it allows user take full advantages of the powerful multitasking capabilities of modern operating system. In this project, GUI is used to display the monitoring system in dashboard such as speedometer, temperature, RPM and engine oil which capture the data from CAN Bus. For this project, PC was used to display dashboard by using programming language in Visual Basic 2010.

1.1 Research motivation

Nowadays, cars had been developed in variety size, colour, and function from Variety Company. It shows world automotive industry was evolving from time to time. Malaysia does not miss the trend of automotive industry development. All cars competing companies market products that can fulfil the needs of the world community.

Furthermore, due to competition in automotive marketing over the globe, many automotive companies are vying to get the best marketability products. Moreover, by using GUI dashboard it can create more modern dashboard with a variety of themes can be displayed. So, the convenient vehicle with less oil usage can be produced to meet the modern lifestyle nowadays.

1.2 Problem Statement

Nowadays many automotive industries have been using analogue dashboard in vehicles. This analogue dashboard had been chosen because it easier to develop. Even though it had been widely used, but it has limitations during its operation. For example, fixed pattern and costly to replace when the dashboard damage.

Besides that, the analogue dashboard required wiring system with the bulk of heavy and expensive wires. Consequently, the GUI vehicle dashboards are introduced to overcome this problem by reducing wires and capture the data from CAN Bus network system.

1.3 Objectives

- To analyze CAN Bus data by using (National Instrument) NI-CAN USB 8473s
- To design a GUI dashboard

1.4 Scope

- To study about the capability of GUI in capturing real data.
- To develop the GUI dashboard by using Visual Basic 2010

CHAPTER 2

LITERATURE REVIEW

The objective of this chapter is to explain overall subjects related to this project. Several references had been used such as academic book and technical journals. This chapter cover types of topology used, controlled area network (CAN), dashboard and graphical user interface (GUI) in this project.

2.1 Type Of Bus

In every network, had two types of typology which is physical topology and logical topology. In logical topology describes more to communication of component across physical topology. However, there are various types of physical topology network such as star topology, mesh topology, ring topology and bus topology. In this project bus topology are used for networking system .Bus topology is a single cable network or “trunk” or “backbone” and linked to all computers. The illustration of this topology was shown in Figure 2.1.

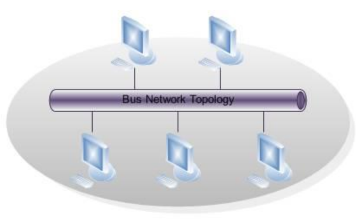


Figure 2.1: Bus network topology [21].

All the computer will connected to one single cable and affected the performance each computer. In this network, every computer cannot send data simultaneously. Only one computer can transmit data at a time. Others computers will be waiting to transport data. In addition, the data can't send in computer-computer interaction. Data will only commit to trunk cable not other computers in this network.

This network had been chosen because of it is easy to install, inexpensive, flexible and it's required less wires used. However, bus topology has their disadvantages such as it had a limitation of cable length and number of stations, when one node fails, the entire network will shut down and the executions of the network will fall due to increasing of node or computer.

2.2 Controlled Area Network (CAN)

Controlled Area Network (CAN) is a serial bus communication protocol that provides reliable, economic, and efficient link between devices to provide the distributed real time applications by using bit-wise deterministic collision- resolution mechanism [1]. Besides that, it also had high integrity serial bus system for networking intelligent devices as in standard vehicle in the network.

CAN Bus can be classed as a broadcast communication mechanism because of it node operating system. This is because all the nodes can 'hear' the transmission, without exception. Furthermore, CAN is message oriented transmission protocol. It means, every message has its own message identifier and the whole network will define the priority of the message. [4]. Therefore, if two nodes send messages at a time, the one who higher priority will transmitted first followed by lower priority's message.

This transmission protocol will cause collision among the messages, and it'll be resolved by using bit-wise arbitration then allow the message to remain inviolate. In past, CAN Bus system are aim for automotive network system but nowadays, the CAN Bus is widely used in other sector such as automation.

European countries such as Swiss, German and Czech CAN Bus network had been enforced in public transport to control and monitoring door, light and others automation system. Thus, CAN Bus network system have a high demand in the world market place. In addition, there are several reasons why CAN Bus system had been chosen to replace the old wiring system.

In the past, most industries used point to point wiring system to connect electronic devices in vehicles. Due to improve on technology times by times, wiring system became more complicated, bulky, heavy and expensive. Then vehicle's weight will increase due to improved of the wiring system. By using CAN Bus, the wiring system can be scaled down.It can be represented as shown in Figure 2.2..

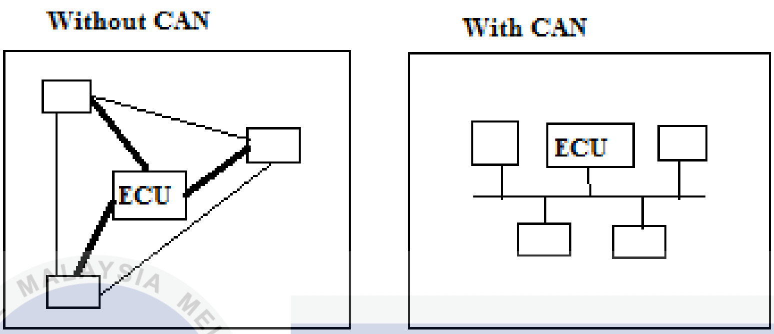


Figure 2.2: Network with CAN and without CAN

From Figure 1.1, it's shown that CAN Bus system only utilize a twisted cable to communicate with each other. Besides that, CAN Bus is contain several characters of large amounts of information, intuitive, high speed of responding, more accuracy and more stable display will had high demand in market [7]. Moreover, CAN Bus also had low cost due to lightweight network, wide broadcast communication, and priority system and had error checking capabilities.

Other features of CAN Bus had serial asynchronous Multi Master bus, transfer rate up to 1 Mbps, maximum bus length up to 40 m, unlimited number of nodes, 2032 different types of messages, 0 to 8 bytes information per data frame and have high safety level due to presence of mechanism for detection, notification and recovery faults.

Due to CAN Bus protocol, when the collision occurs, the arbitration will solved and allow message remain intact. Then, to main the arbitration process, the logic bit needs to define as either dominant or recessive [1]. For example in Figure 2.3, there will be 3 nodes in transmitting process. When all nodes start sending bit frames at the same time, node 1 and 2 will stop at identifier 6. This is because, at identifier 6, node 1 and 2 had contained bit 1 (recessive) while node 3 proceeds because it had bit 0 (dominant). So, node 1 and 2 will be in receiving mode.

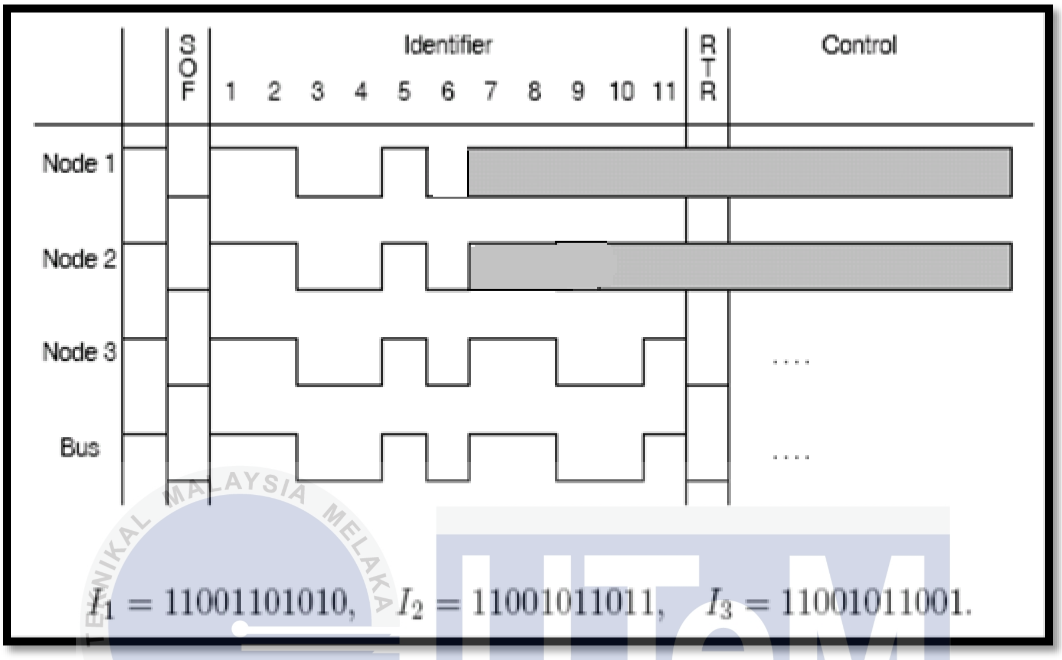


Figure 2.3: CAN arbitration [1]

2.3 Graphical User Interface (GUI)

Graphical User Interface is a communication between user and computer through certain software [6]. In other words, the GUI can describe as a graphical display that contain component to allow user perform without creating script [14]. There are several software can be used for GUI such as Microsoft Visual Basic, LAB View and Mat Lab.

This software is a platform to represent the GUI. The GUI was generating to make thing simple especially for the end-users program [13]. Therefore, the application can be easy to use for everybody. For example in medical monitoring equipment, there will be some graphic visual measurement such as type of blood, temperature, and heart beat rate. Thus, by using GUI the physician or technician can simply understand the measurement.

Without a GUI application, people had to work from the command line interface and make the things more hard and difficult [13]. Moreover, the GUI will save time and budget for user and system analyst. This is because, for user it is easier to use than it can save time to learn command line to operate some application. But for system analyst, when the GUI application shuts down or error, they just need to check the programming code to solve it. It can save budget because it doesn't need any tools to fix it.

For this project, the GUI was applied in PC. Microsoft Visual Basic will be used GUI in PC to display the simulation of CAN Bus system. For GUI visual, the component in the dashboard will be designed such as speedometer, RPM, Engine Oil and Temperature. The visual dashboard will be designed using programming code in Microsoft Visual Basic 2010 Express.

2.4 Related Research Work

From the previous research work, there are many techniques to develop a GUI of CAN Bus Dashboard. But, most of it represents a monitoring system of CAN Bus system.

2.4.1 Home Appliances Management System uses Controller Area Network [1]

In this paper describes more detailed about CAN is the main communication protocol for home management system for example security alarm. All the parts of the CAN were included such as CAN protocol, node and bus. Besides that, this system use three main components included CAN which is relayed module and centrally controlled system (CCS) and use GUI to obtain user input. The relay module is the switch of this management system. This paper will explained CAN as the main communication protocol in the system. The objective using CAN as main communication protocol because its immune from electrical interferences besides, low cost in its simple design.

2.4.2 A Design for Automotive CAN Bus Monitoring System [2]

The purpose of this paper is to increase CAN Bus network debugging sufficiency and to develop processes of vehicular CAN network system. To achieve this objective, this paper introduces a new CAN Bus monitoring system. Monitor can control more than a single control unit. In other word, this paper increase the functionality of CAN bus. For example, CAN not only can receive and sent data carefully but it also can save data in log file for off-line evaluation. Besides that, this paper use USB technology for CAN bus communication with PC in high speed especially when the experiment doesn't occur in laboratories.

2.4.3 Graphical User Interface Testing Optimization For Water Monitoring Application [5].

This paper chose global climate problems as their motivation to pursue their techniques. By using an application called Water-Evo GuiTest and Cyber water as experimental tools the results from research are obtained. Moreover, this paper was chosen because it use GUI for monitoring the application. It also described advantages of GUI for user which is it more friendly user and more intuitive. Otherwise, this paper presents an original automates testing method for GUI’s using genetic algorithms. From this paper, the ability and possibility of GUI was tested independently by using the application of source code itself and include all possible events.

2.4.4 Design Of Electrical Air-Condition Control System For Fuel Cell Vehicle Based On CAN Bus [8].

The principle of CAN and the structure of fuel vehicle control system based on CAN were described in this paper. It highlights in air condition control node and compressor drive system includes hardware and software design. The hardware design may include P89C51Rx2 and microcontroller ST72141 for brushless DC motor and independent CAN controller. ST72141 can effectively reduce hardware cost and improve control effect.

2.4.5 Research On Fault Diagnosis And Forecast System Forest Harvester Based On CAN- Bus Information [11].

The objective of this paper is to utilize on the current forest harvester, the complexity of faulty information, the difficult diagnosis on CAN technology. For the purpose, a USB-CAN intelligent interface card was designed in this paper to build the fault diagnosis system with neural network and Kalman Filter by USING Microsoft Visual

C++6. 0. The USB-CAN intelligent card is a data conversation card with which to connect the network card to PC through USB. From the experimental result, the USB-CAN could receive high speed signal from each CAN bus port. It's effective to diagnose faults of forest harvester with a BP neural network.

2.4.6 Graphical User Interface (GUI) For Thumbprint Image Enhancement and Minutiae Extraction

The purpose of this paper is to identify the individual identity based on thumbprints. As we know, every person has a different type of thumbprint. By using GUIs, thumbprint will be scanned and real end point, and real branch point will be identified. Both of the points give different values to matching processing of between thumbprint and person identity. To get results for this project several procedures should be done such ad histogram equalization process, enhanced by Fast Fourier Transform (FFT) factor and image linearization, minutiae extraction, and region of interest extraction. The software used in this project is MAT LAB R2009b and use GUI for identify input, thumbprint.

2.4.7 Graphical User Interface for Enhanced Retinal Image Analysis for Diagnosing Diabetic Retinopathy

The purpose of this paper is to detecting the diabetic Retinopathy problem using the GUI environment. Diabetic Retinopathy is an eye disease due to the critical level of diabetes. Funds camera will capture retina to analyze the retina condition. Then, image will process and optic disc in the retina will detect features found in writing and display by using GUI. Use the software to develop GUI is MATLAB 7.8.The algorithm from optic disc and MATLAB will improve by using available database. The GUI develops to make an automated environment for diagnosing diabetic retinopathy.

2.4.8 Graphical User Interface Foe Next Generation Power System

The purpose of this paper is to facilitate use and comprehension of power equipment by producing a simple tool that convenient for user to use. Therefore, to make the information for equipment easier to use, human machine interaction must be designed. There are some basic rules to follow in the design of graphical user interface such as position, orientation, environmental conditions, viewing distance, ambient lighting levels, and physical attributes of display, device design and human observer engaged in performing tasks.

2.4.9 Speed Trap Detection with Doppler Effect[22]

The purpose of this paper is to auto capture and auto save an image in directory if vehicle speed is over speed limit. Therefore, authors use Doppler radar to detect the speed of vehicle. Doppler radar being used as a sensor because it apply principles same as Gun Radar to detect signal for certain moving object. Doppler will send signal to microcontroller and the speed of vehicle will calculate, and display on LCD. Next, data will send to PC through RS 232port via Serial Communication Interface. On the PC, GUI is developed with Visual Basic 6 to receive data from micro controller and capture data. Microsoft Access database program is used to save and record data.

2.4.10 GUI for PC Auto-Shutdown [23]

The purpose of this paper is to develop of GUI for auto-shutdown of personal computer by using Microsoft Visual Basic. This application capable to save all programs and shut down PC safely when power goes out. It will save software and PC from damage. When power goes out, Uninterrupted Power Supply (UPS) will supply power to PC and shut down safely before reach its limits.UPS will send signal to PC via RS232 serial port. Microsoft Windows Application (API) will applied in Microsoft Visual Basic to shut down down the computer system.

2.4.11 An Introduction to Data Capturing [24]

The purpose of this paper is to implement data capturing in different network situations, including wired Local Area Network (LAN) and wireless LAN, to introduces the data capturing engines in UNIX and Windows system, to compares the differences between them. To presents the critical technique of data capturing behaviour in network and to point out the trend of future a development. Data capturing for software and hardware had different method. This paper explains that hardware has better computing capability than software, therefore capturing hardware data is a hot research fields.

2.4.12 Application Design of Data Packet Capturing Based on Sharpcap [25]

The purpose of this paper is to capture data using new method which is Sharpcap on dot. Net. Sharpcap is a development package of capturing the first floor network data, data that flows through data linker layer. By applied Sharpcap on dot.Net, it has solved long standing problem of capturing packets. It is powerful complement and has to be integral part of network monitoring when someone will develop on dot Net problem.

2.4.13 Study on Log-Based Change Data Capture and Handling Mechanism in Real-Time Data Warehouse [26]

This paper proposes a framework of change data capture and data extraction, and then loaded to real time to improve the quality of data and have best system performance. Firstly, changed data will be captured on log analysis and data will be processes data to push for data queue. Then queue's data will process based scheduling algorithm and loaded to real time warehouse. This paper had elaborate technique to capture changed data by using online log and researching data scheduling strategy of captured data.

2.4.14 Efficient Data Capturing for Network Forensics in Cognitive Radio Networks [27]

This paper proposes a systematic method to capture data in cognitive radio networks with a small numbers of monitors. Besides that, this paper also propose a protocol that schedules multiple monitors to perform channel scanning and packet capturing in an efficient manner. A monitor scheduling algorithm and a comprehensive protocol to coordinate monitors among channels. By using this method, capturing data in cognitive radio network will improve and became easier.

2.4.15 The Design of Communication Converter Based on CAN Bus [28]

This paper proposes a new design of a communication converter and a method of conversion is between RS 232 and CAN Bus. Besides that, in the design have some advantages which are high communication velocity, simple configuration and low cost design. The field CAN Bus system have a nicer application foreground in control system domain for a good many excellences of CAN.

2.5 Summary

This chapter describes basically about previous research article related to this project. CAN bus system is a communication protocol for system to transmit data and signal. Graphical User Interface is a technique to display in monitoring system with easy and user friendly environment. Both CAN and GUI are important element in this project. Besides that, there has method to capturing data to analyze data transfer to GUI. There had 15 research papers with different method and application of GUI and CAN. Most of research paper use software such as Mat Lab, Lab View and Visual Basic as a tool to obtain analysis result.

CHAPTER 3

METHODOLOGY

In this chapter, every method to achieve the objectives will be described. It includes method for software and hardware to obtain results. A component used for system modelling will explain.

3.1 Introduction

In this project, software and hardware will be used to analyze results and achieve the objective. Figure 3.0 shows CAN Bus network system in this project. 5 software had been used which are Microsoft Visual Basic 2010 Express, Proteus 8, PIC Compiler, PIC Kit, and Virtual Serial Port Emulator. Firstly, software part had to be completed before proceed to hardware parts. Mostly this project used coding to display and transmits data from serial port to GUI. Flow Chart for the whole system can be shown in Figure 3.1

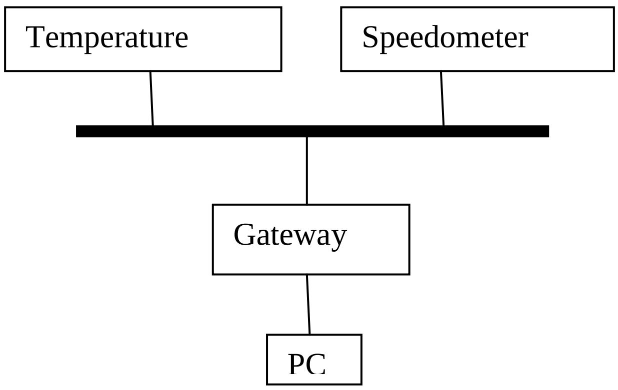
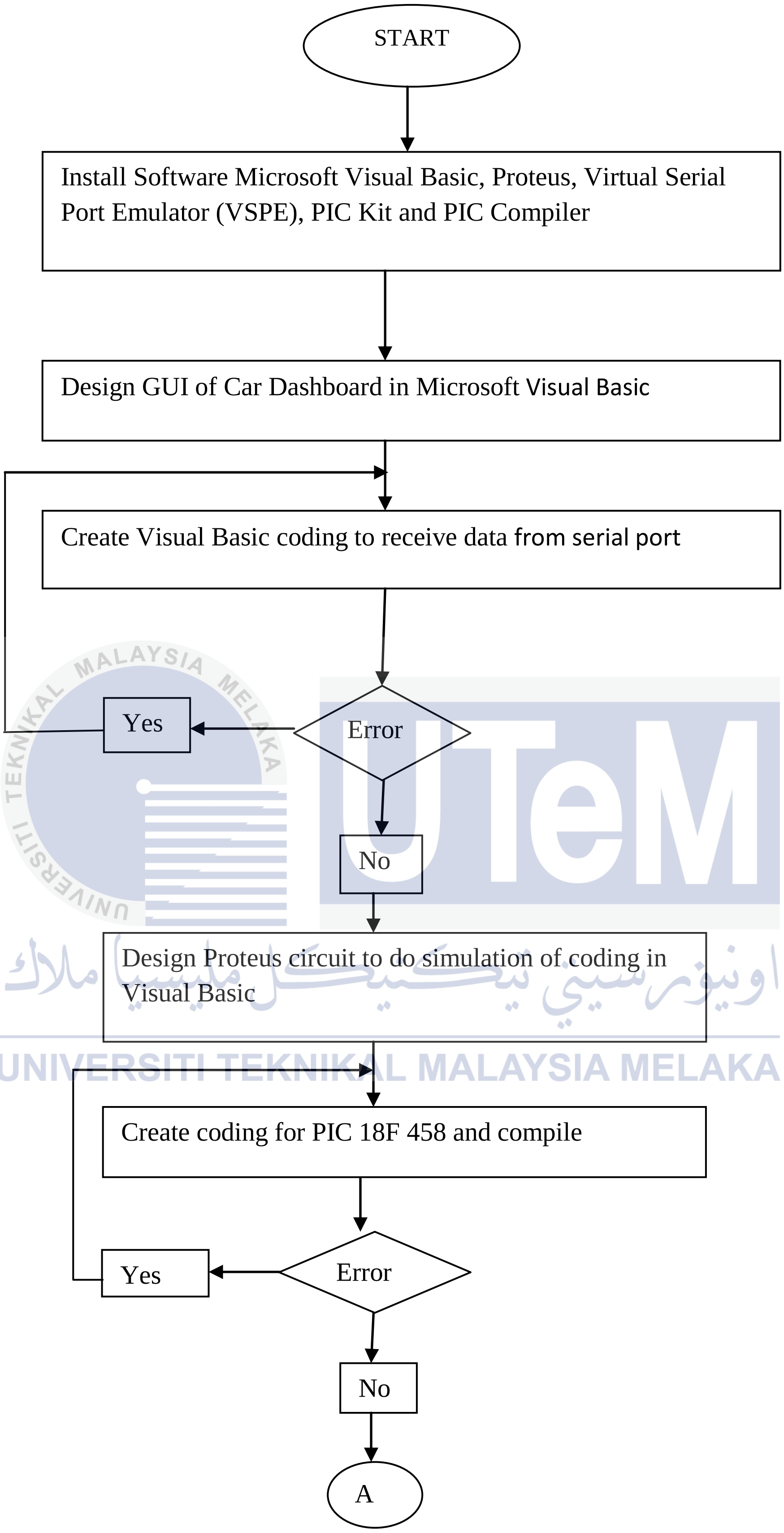
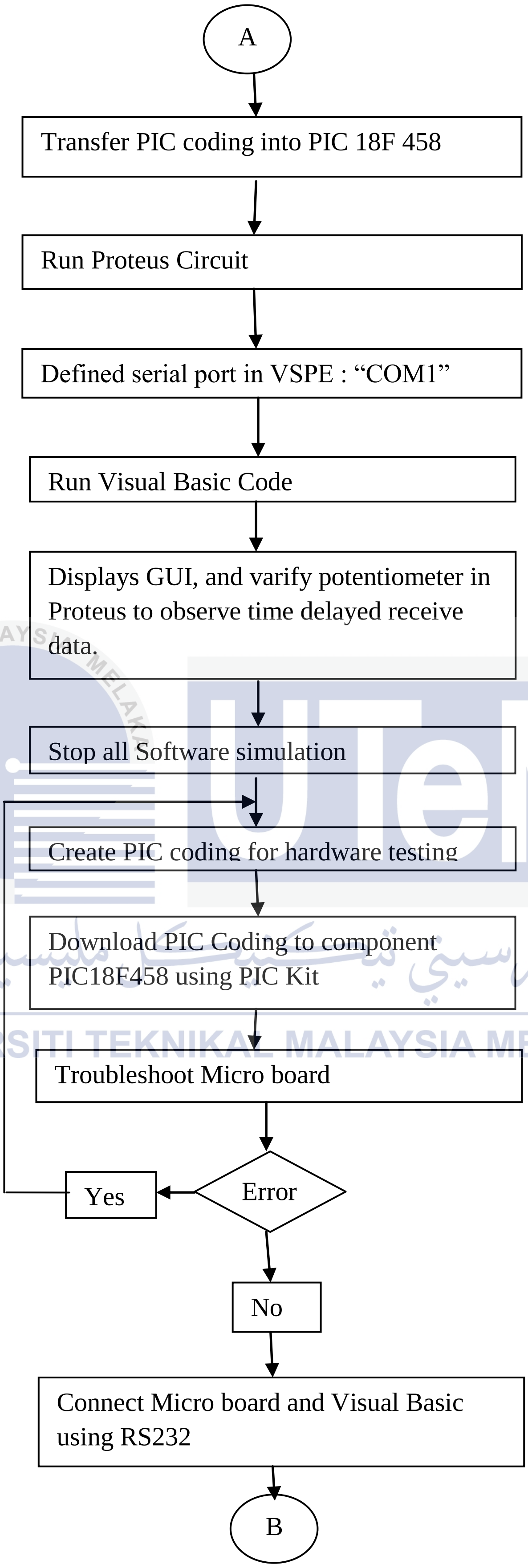


Figure 3.0: CAN Bus network system


START
Install Software Microsoft Visual Basic, Proteus, Virtual Serial Port Emulator (VSPE), PIC Kit and PIC Compiler
Design GUI of Car Dashboard in Microsoft Visual Basic
Create Visual Basic coding to receive data from serial port
Error
Yes
No
Design Proteus circuit to do simulation of coding in Visual Basic
Create coding for PIC 18F 458 and compile
Error
Yes
No
A


A
Transfer PIC coding into PIC 18F 458
Run Proteus Circuit
Defined serial port in VSPE : “COM1”
Run Visual Basic Code
Displays GUI, and varify potentiometer in Proteus to observe time delayed receive data.
Stop all Software simulation
Create PIC coding for hardware testing
Download PIC Coding to component PIC18F458 using PIC Kit
Troubleshoot Micro board
Error
Yes
No
Connect Micro board and Visual Basic using RS232
B

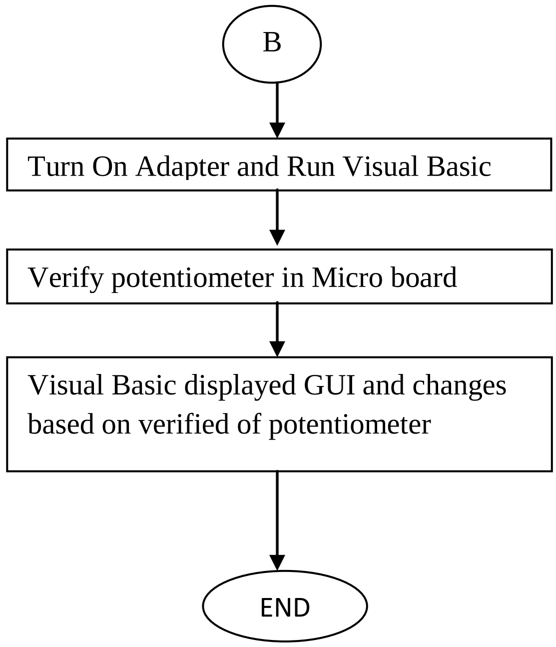


Figure 3.1: Flow Chart for System Flow

The Figure 3.1 explains the operation of this project. Firstly, all software are installed. GUI for car dashboard was designed using Microsoft Visual Basic (VB).In fact, coding for receiving data transmitted from serial port was also developed in VB. Then, compiled to identify any error occurs. If there any error, coding in VB's rechecked and compile again. After debugging VB successfully, proceed with design circuit in Proteus Software. Figure 3.2 shows circuit had been designed using Proteus Software.

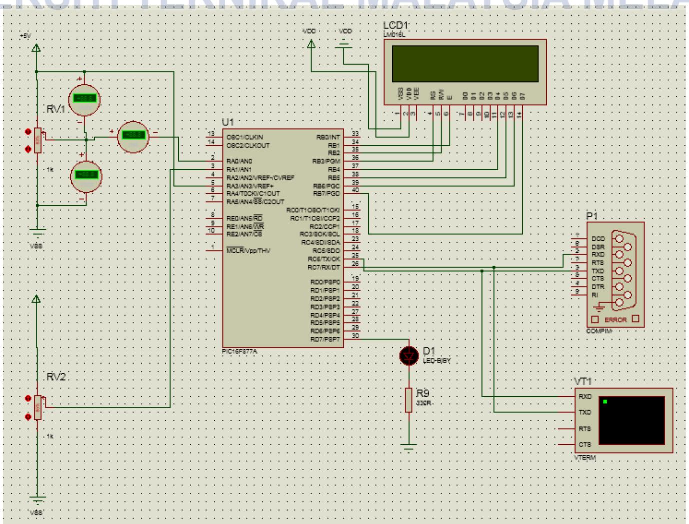


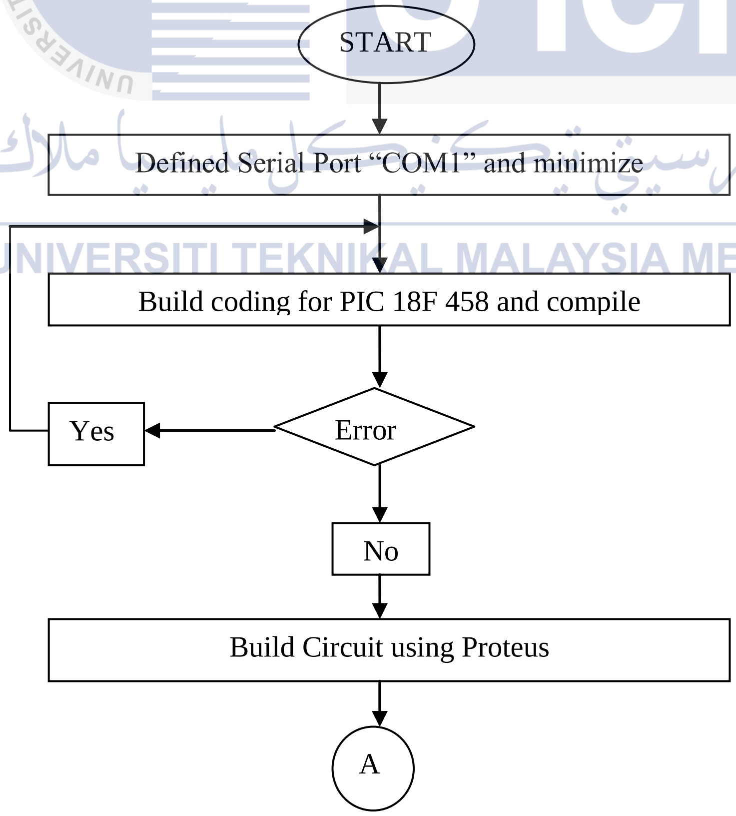
Figure 3.2: Proteus circuit had been design for simulate coding software

Next, create coding for PIC 18F458 using PIC Compiler. Coding PIC will be transferred to component PIC 18F458 in Proteus as programmer’s component. Run VB and GUI Car dashboard will be pop out. Hardware part is the connection between CAN Bus and GUI to verify the changes of potentiometer and GUI in real time.

For hardware part, microcontroller board will be analyzed by using (National Instrument) NI-CAN USN 8473s to see data transfer, data speed and data in hexadecimal form. Next, proceed with simulation of GUI display of CAN Bus dashboard. Several components such as potentiometer, wire 1 core, multimeter, RS 232 cable was used during hardware simulation.

3.2 Software Design

Software design was used to stimulate coding in Visual Basic towards input data. Several software was such as Proteus, PIC Compiler, Virtual Serial Port Emulator, and Visual Basic 2010 Express. Figure 3.3 shows flow chart for software design.



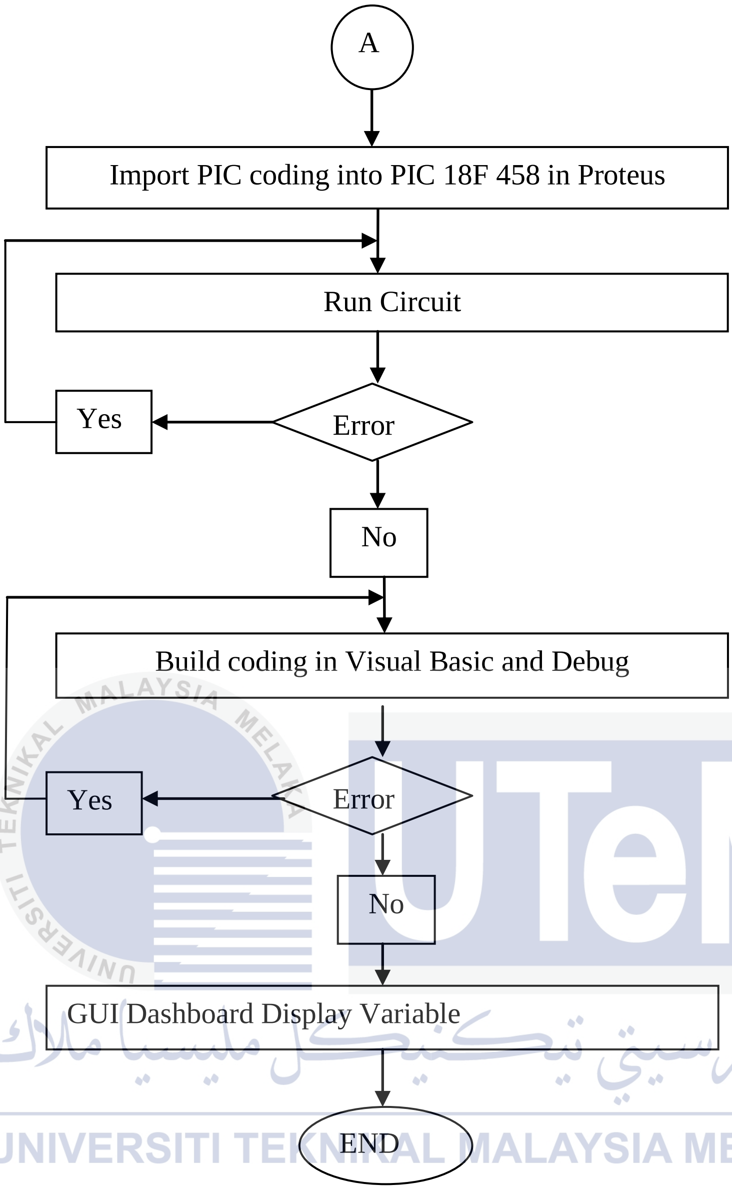


Figure 3.3: Flow chart for Software Design

Firstly, serial port was defined using VSPE, "COM1" and "COM2" was choosing as serial port. Figure 3.4 shows "COM1" and "COM2" defined as serial port by using VSPE.

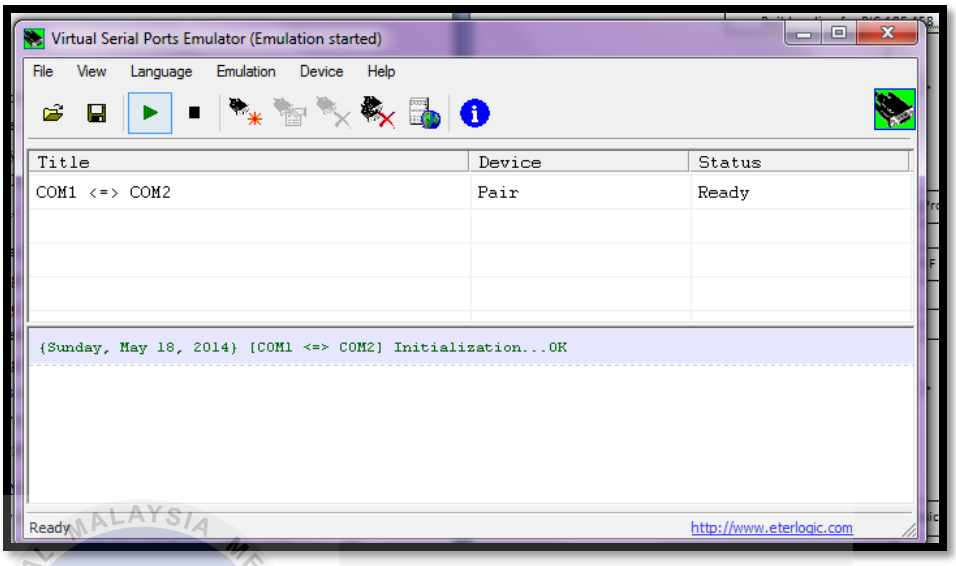


Figure 3.4: COM1 and COM2 was defined

From Figure 3.4, VSPE wills acts as RS 232 in simulation part, connect from Proteus to visual basic. Next, coding in PIC Compiler was compiled and transfer into PIC18F458 in Proteus. Coding for PIC component will define data in analogue and convert to digital for display on LCD and transfer to serial port. Figure 3.5 shows input data was defined into PIC Compiler coding.


```
#include<16f877a.h>                                    //Device
//#include <string.h>
#fuses xt,nowdt,protect,noput,nolvp                    //Fuses
#device adc=10                                         //ADC 10-bit
#use delay (clock=4000000)                             //4.00 MHz
#use rs232 (Baud=9600, xmit=PIN_C6, rcv=PIN_C7, parity=N,bits=8,stream=PC)
#define use_portb_lcd TRUE                             //(only for printf function)
//#include"llcd_new.h"                                   //LCD driver
```

Figure 3.5: Coding to defined input analogue data

From Figure 3.5, time baud, LCD to display data and number pin receive and send data was mention. Proteus circuit's simulation was run to get result. Input Data was analogue bit but PIC will convert to digital and display using LCD. Figure 3.6 shows Proteus Circuit during simulation.

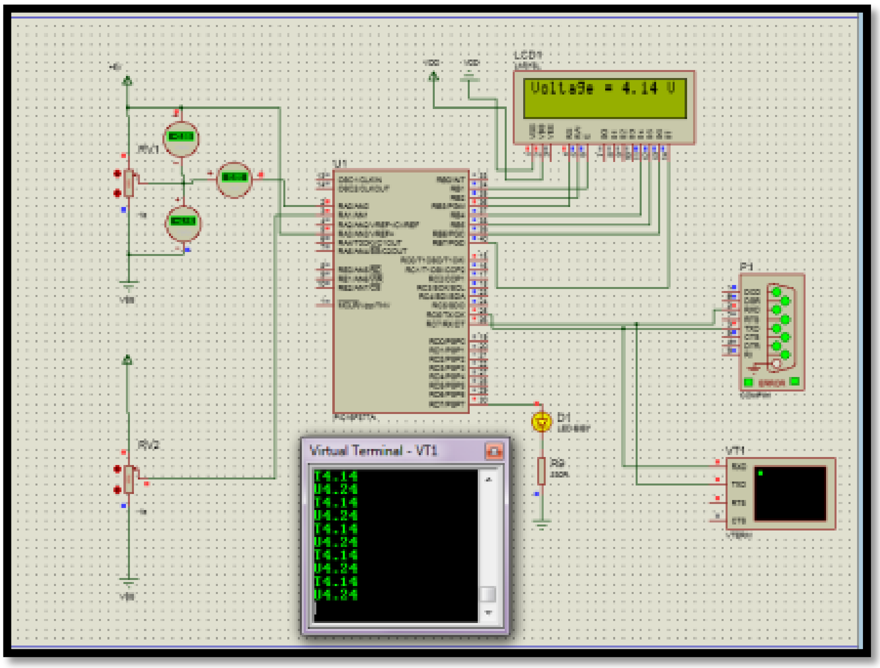


Figure 3.6: Proteus circuit during Simulation

Next, the application is executed and during this simulation, all continuous data will display in speedometer. Data displays in speedometer are continuous data transfer from Proteus simulation circuit Speedometer's reading will changed due to verification of potentiometer in Proteus circuit. Figure 3.7 shows GUI display during debugging of VB coding.

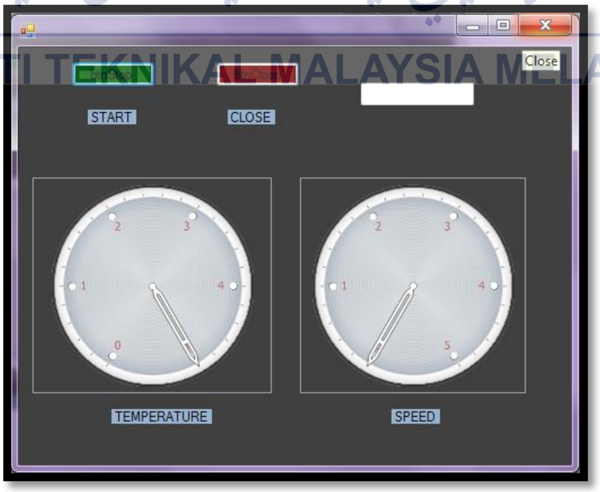
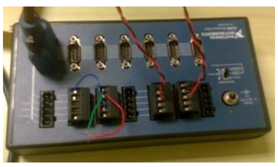


Figure 3.7: GUI display during Software simulation

3.3 Hardware Design

In previous sub chapter was explained about software simulation. In this sub chapter will discussed about hardware simulation. This simulation been done to analyze time changes between potientometer verify and GUI display in VB. Several component needed to run the hardware simulation Each component had theirs part to make this simulation works. List of component can be seen in Table 3.0.

Table 3.0 : Components used during Hardware simulation

Type of components	Figure	Function
Potentiometer		Know as variable resistor or rheostat. It Its resistance can be adjusted to control the current flow to circuit. In this project, its acts as adjustable input or as pedal in real time to measure speed.
CAN Bus		Controller Area Network (CAN) Bus mostly applied in automotive and automation field. In this project, it acts as connction betwwen miccrocontroller and GUI display. It will send and receive data from potentiometer and GUI respectively.
RS 232 cable		RS 232 cable atcs as conector between serial port in microcontroller boards and GUI display.
Multimeter		To check cintinuity tse between each components before simulation and during error occur.
Wire 1 core 15 mm		As jumper for between component such as between potentiometer and microcontroller board.
Microcontroler 18F458		Electronic components that can be programmed to carry out a vast range of tasks. In this project, it restore coding write in PIC compiler to simulate input signal.
NI-CAN USB 8437s		Acts as CAN Bus in microcontroller board. It will connect hardware part with PC. From this instrument, data transfer (in hexadecimal) can be seen

From Table 3.0, each component had their function during simulation. Next,to make this simulation succesfull there are flow chart to guide this simulation. Firstly, Flow chart for hardware simulation an be seen in Figure 3.8.

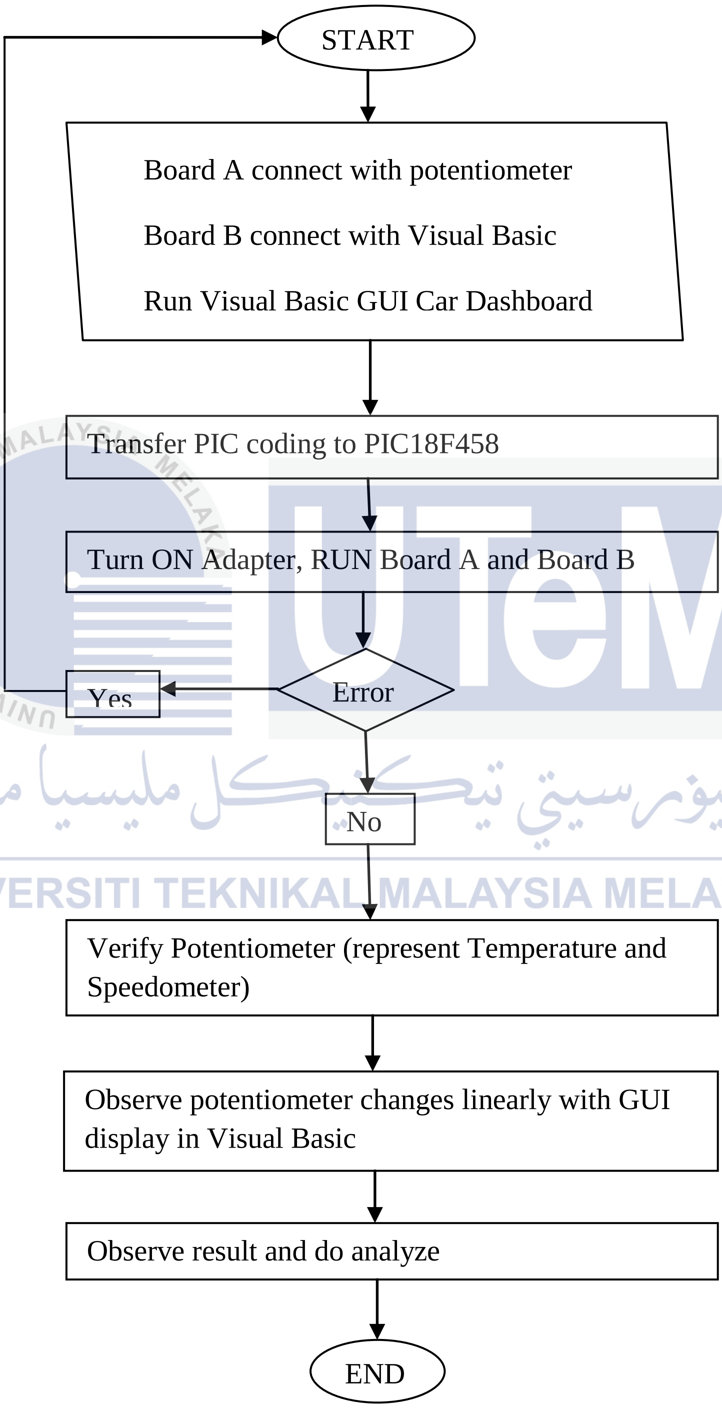


Figure 3.8: Flow chart for Hardware design

Before hardware simulation operating, Board A and Board B was troubleshooting to identify any error. Board A and Board B can be seen in Figure 3.9 and Figure 3.10 respectively.

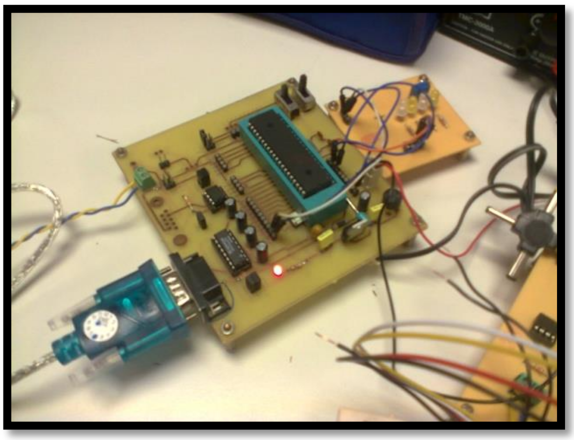


Figure 3.9: Board A during simulation

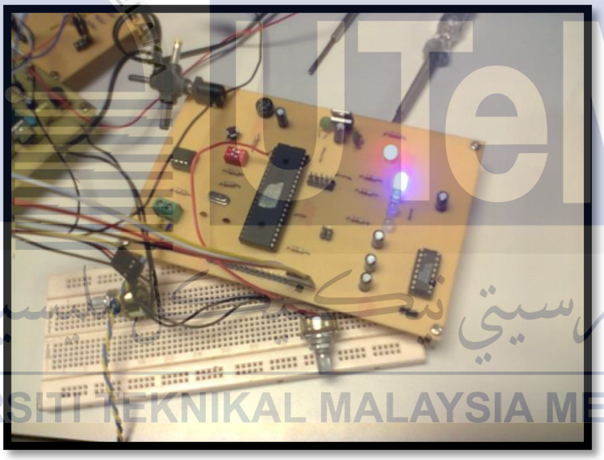


Figure 3.10: Board B during simulation

Both boards contain component PIC18F458 and its used CAN Bus to connect to each others. PIC coding was transfer to both PIC18F458 using PIC Kit.. Both boards receive voltage supply from adapter and potentiometers only at Board B to verify voltage by adjust its switch. Board A will connect to GUI Car Dashboard by using RS232 cable.

Turn On adapter and Run VB to display GUI Car Dashboard. If there any error, rechecked coding and transfer again to PIC component. If there are no error, observe the changes of potentiometer adjust are linearly to gauge changes. Does gauge movement have any delayed based on potentiometer tuning. Figure 3.11 shown connection between Board A, Board B and VB.

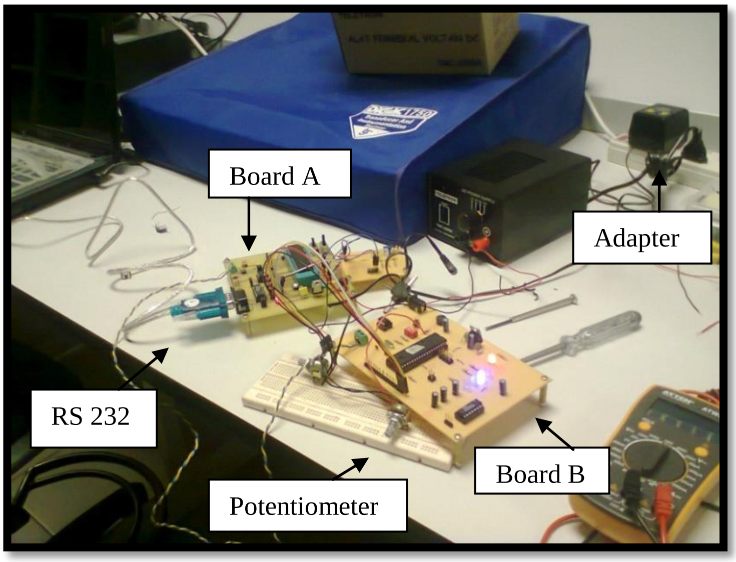


Figure 3.11: Connection during Hardware simulation

Next, microcontroller board will be analyzed by using NI-CAN USB 8437s to identify several parameters such as:

- Numbers of bytes in every capture frame
- Data bytes contained in the capture frames
- The transmission rate of the frame
- Total numbers of transmitted frames
- Minimum and maximum delta time between two frames

The new hardware connection was built up to analyze microcontroller board using NI-CAN USB 8437s. During this simulation, CAN Bus connection was replaced with NI-CAN USB 8437s to transfer data to PC. The hardware connection can be seen in Figure 3.12.

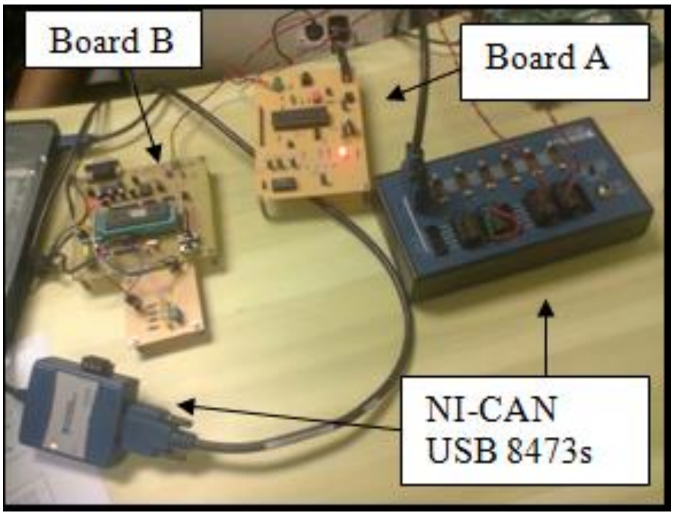


Figure 3.12: Hardware connection during analyze system by using NI-CAN USB 8437s

3.4 Summary

This chapter will describe method use for this project. These methods include software and hardware simulation. There are also types of component used in micro controller board during hardware simulation. Micro controller board will transmit, convert signal to digit, and transfer to GUI. Coding used during software simulation and hardware simulation different due to requirements of simulation type.

CHAPTER 4

RESULT AND DISCUSSION

This chapter describes the result for this project. From the objective, there are two results should explained which are the analyze system using NI-CAN USB 8473s and develop Graphical User Interface of dashboard. This chapter will present results to fulfil project objectives. Moreover, in this chapter will discuss analysis about the result.

4.1 Software Result

Software was used to test whether VB coding valid for hardware simulation. From software simulation, the maximum value for gauge's readings is 5. Figure 4.0 shows data readings in Proteus circuit.

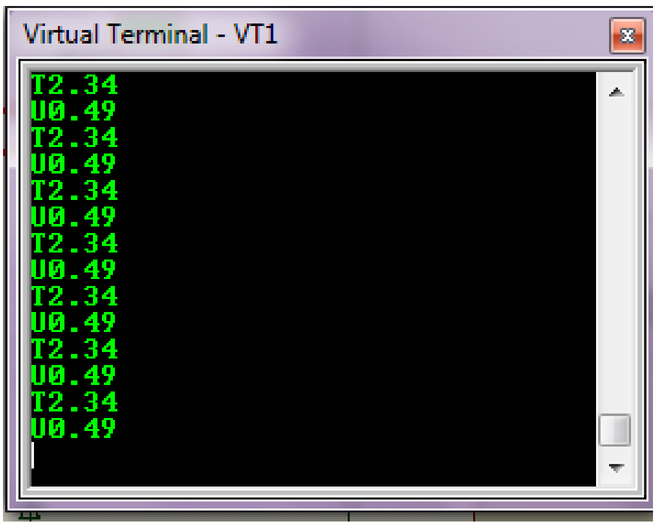


Figure 4.0: Data reading in Proteus Circuit during software simulation

Data display in Figure 4.0 shows two parameters was reading continuously which are “T” and “U”. This is because, there are 2 potentiometers (input) in Proteus circuit. Since there are 2 inputs, PIC coding had to define 2 parameters to differentiate between 2 inputs. Figure 4.1 shows coding PIC to define parameter “T” and “U”.

```
//fprintf(PC, (c*5/1024));
//putc(adcvalue >> 8);
//putc(adcvalue & 0xff);
printf("T%f\n\r", adcvalue);
printf("U%f\n\r", adcvalue1);
delay_ms(100);
}
```

Figure 4.1: PIC coding to define 2 inputs in microcontroller

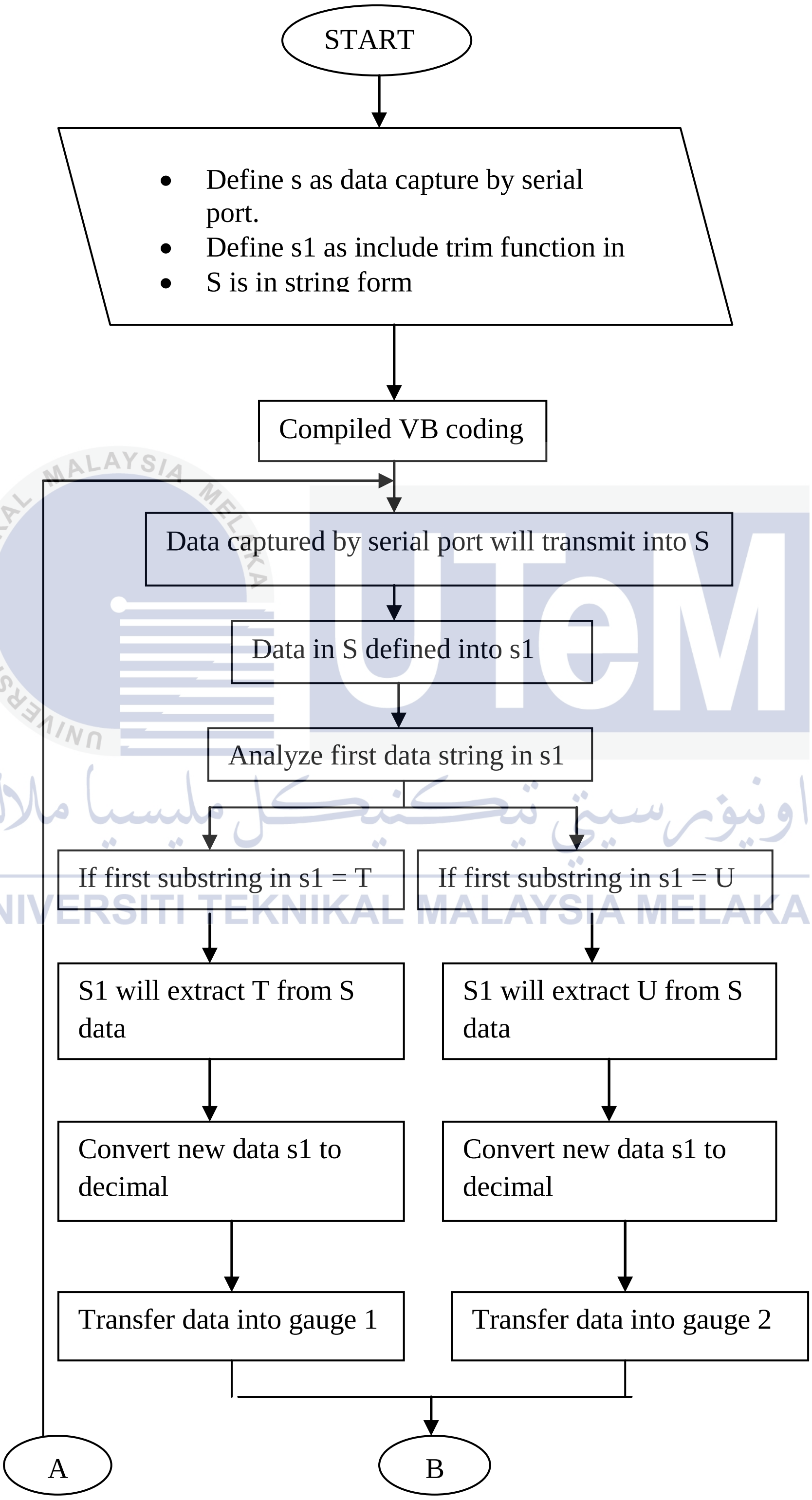
Figure 4.1 shows PIC coding to define 2 inputs in microcontroller. Parameter “T” was defined from adc value and parameter “U” was defined from adcvalue1. Then serial port will capture 2 parameters data into VB. In VB, all data was import as String. Since there are 2 inputs, therefore there are 2 types of data starting with ‘T’ and ‘U’. Data T and Data U import into visual basic at the same time. Thus it should separate to take data and change it decimal for 2 parameters. Command to separate data t and data u can be seen in Figure 4.2


```
'SerialPort1.Write("<hello>")
s = SerialPort1.ReadLine()
s1 = s.Trim
'If s = "T" Then
'If stringComparer__1.Equals("T", s) Then ' The following statement returns True (does "F" satisfy "F"?)
If s1.Substring(0, 1) = "T" Then

    s1 = s.Trim
    s1 = s1.Substring(1, 4)
    'TextBox1.Text = TextBox1.Text + s1 + vbCrLf
    number = Convert.ToDecimal(s1)
    ArcScaleComponent1.Value = s1
    ' number =
End If
```

Figure 4.2: Coding in VB to separate data from Proteus

Data T will command to gauge1 and Data U will command to gauge2. From Figure 4.2, s will read data from serial port which is Tx.xx and Ux.xx. ” x” will represent numeric during capturing data then s1 will command to trim function toward variable s. Then ‘If and End If’ condition was applied. If string start will display at gauge 1.Before that, data t will separate between two categorized which is numeric and non-numeric by using command (s1= s1.Substring (1, 4)). For example, when data T reads = “T4.44”, only 4 substring from back will consider. Therefore, it will only consider 4.44. This process was applied same method towards data D but different type of display. Data D use gauge2 to display its readings. Figure 4.3 shows flow chart for previous expaination.



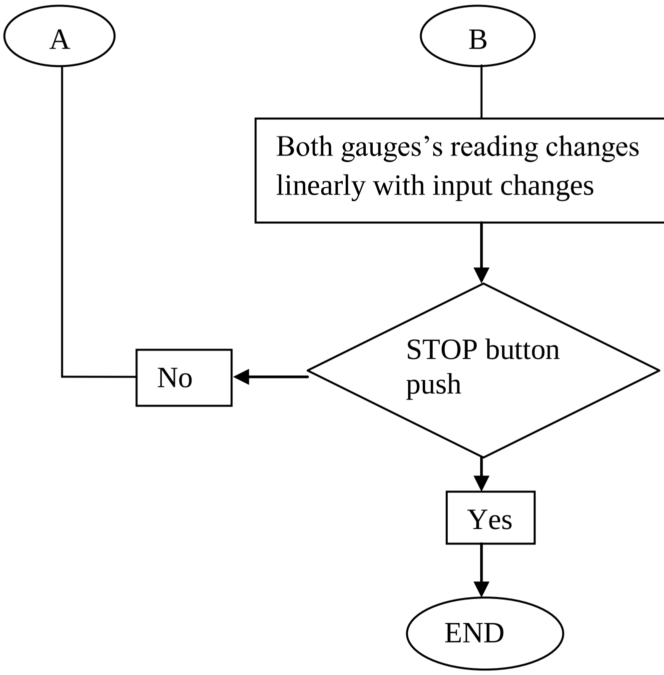


Figure 4.3: Flow chart for extracted data from serial port

Figure 4.3 shows process flow during extracted data from serial port. Extracted data convert to decimal and display on speedometer (gauge 1 and gauge 2). Potentiometer will verify and give different type of input. It makes gauge readings changes linearly toward changes of potentiometer value. Figure 4.4 shows GUI display during Software simulation which is the result of simulation.

Figure 4.4: Results from software simulation

Data transfer will linearly to potentiometer in Proteus. When potentiometers adjust increasing, the gauge's reading will also increasing. It shows gauge was interact linearly with the PIC coding and Proteus circuit.

4.1 Hardware Result

From hardware simulation, data transfer from microcontroller board to GUI. All data was displayed at speedometer. All data readings can be seen by using NI-CAN BUS 8473s. By using this instrument, it will capture data transfer and show it in PC. Figure 4.4 shows data capture during hardware simulation.

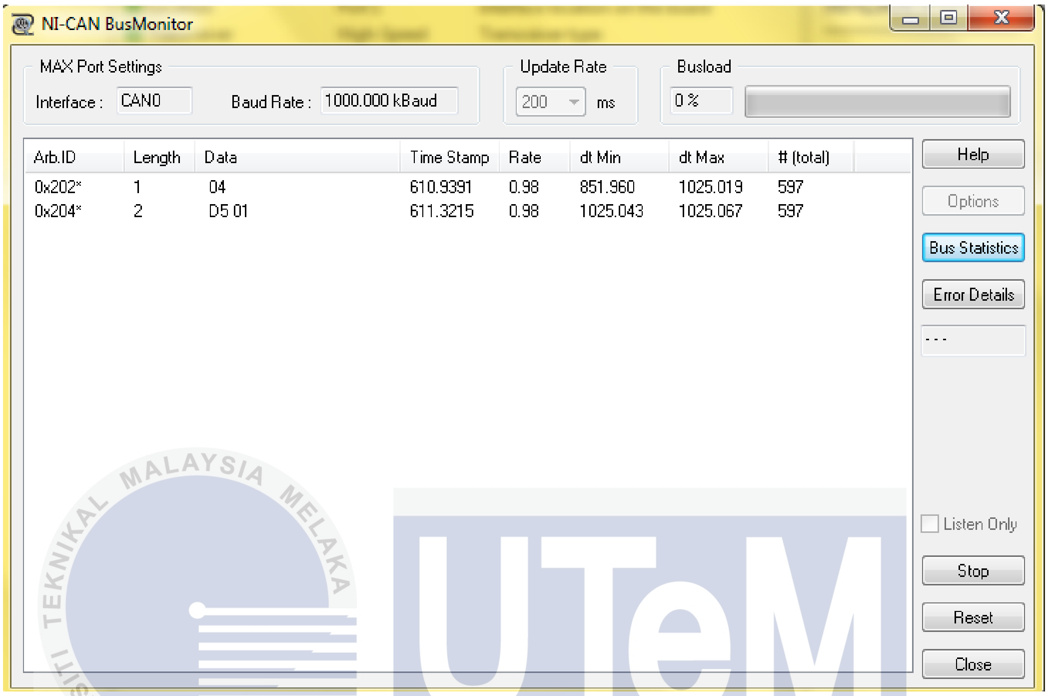


Figure 4.5: Data transfer during hardware simulation

Figure 4.5 shows data transfer during hardware simulation. Parameter “Arb.ID” is arbitration ID of the CAN frame usually came in 2 forms which are standard and extended. Standard contain 11 bits, while extended contain of 29 bits. In this project, extended frame is denoted with symbol “*”. Two arb.ID in Figure 4.4 are extended frames since it had “*” symbols in end of its ID. Next, parameter “Data” is bytes data types in hexadecimal formats in the frame. Both of them had 0.98 Hertz (Hz) of transmission rate of frame. Parameter “dt Min” and “dt Max” are minimum and maximum respectively of delta time between two frames. Lastly, parameter “# (total)” is total number of transmitted frames since its start transferring.

In hardware simulation, gauge readings were increasing due to increasing of potentiometer adjusted. Figure 4.6 shows data display in speedometer during hardware simulation.

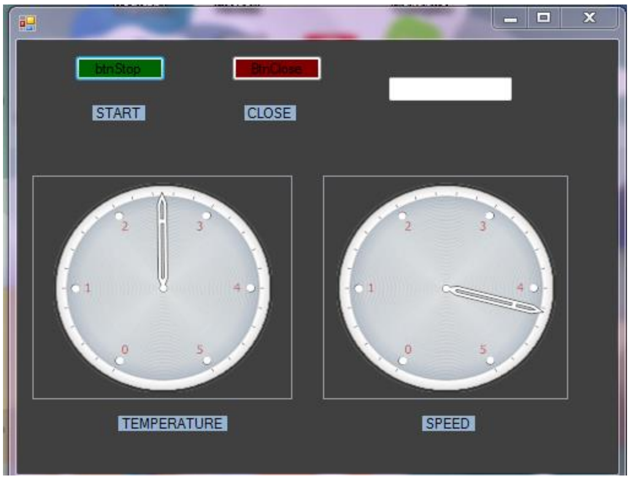
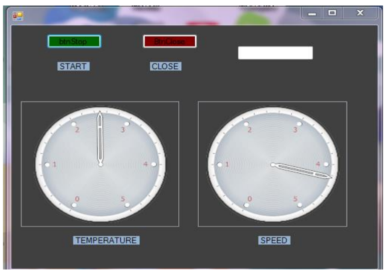
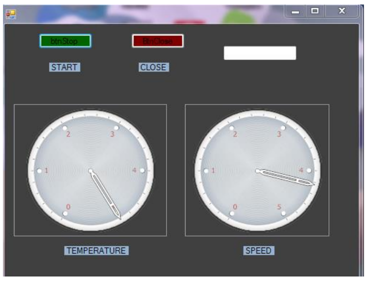


Figure 4.6: GUI display during hardware simulation

Unfortunately, gauge reading during software simulation was different with hardware simulation. Simulation part and hardware part had its own maximum readings which are 5 and 1023 respectively. All value display from hardware and software simulation can be seen in Table 4.0.

Table 4.0: Result for this project

Potentiometer Adjust,%	Value during Software simulation (focused on Temperature gauge)	Value during Hardware simulation
0	 =0	0
50	 =2.50	615

75	 =3.75	923
100	 =5.00	1023

From table 4.0, can be seen value display for each simulation was different. During hardware and software simulation it's display 4 decimal digits. Unfortunately, in software simulation it had "." but in hardware simulation only digits. It's had different display numbers due to different coding in PIC programmer.

In software simulation, there had a formula to change display data in decimal number. Figure 4.7 shows formula in PIC coding.


```
adcvalue = (c*5)/1024;
adcvalue1 = (d*5)/1024;
printf(lcd_putc,"Voltage = %1.2f V        ",(c*5)/1024);
```

Figure 4.7: Formula in PIC Coding

From figure 4.7 shows parameter "c" and "d" was included in formula. Parameter "c" and "d" contain data after extracted non numeric and numeric data. After extracted non numeric in input data, all numeric numbers will transmit into this formula. However, this formula does not applied during hardware simulation. Table 4.1 shows gauge readings before include in calculation.

Table 4.1: Real data reading

Potentiometer Adjust,%	Value during Software simulation	Value during Hardware simulation
0	0	0
50	512	615
75	768	923
100	1024	1023

From Table 4.1 can be seen value data during software and hardware simulation are different. This different due to microcontroller board and PIC coding had its delayed time frames. In addition, during experimental 10 meters wires had been used. The length of wire was affected to value captured. By applied hardware simulation, delayed time will increase. Therefore, data captured during hardware simulation was increased.

However, software simulation had lower delayed time because does not include with microcontroller board. Then, data readings between hardware and software simulation should had slightly different. Even though it had different reading data, gauge still shows its reading during potentiometer verify.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

This project was designed to display GUI of CAN Bus dashboard. It is practically applied on car dashboard to replace manual dashboard using needle speedometer. By using this application, consumers can have change theme of its interface. It's differently with previous dashboards which is permanent display.

As conclusion, both objectives had been achieved. Firstly, to analyze CAN Bus network by using NI-CAN USB 4873s. CAN Bus was analyzed their network and being explain in chapter 4. Secondly, to develop GUI of CAN Bus dashboard. GUI dashboard was developed in VB and being verified using software simulation before applied in hardware simulation. Both simulations give real time data to gauge readings when potentiometer adjusted.

Lastly, this project had fulfilled its objective and the GUI coding was verified with software and hardware simulation.

5.2 Recommendation

As recommendation for improvement in this project, there should add more gauges in GUI display. By add more gauge, there should some addition towards coding in PIC programmer, coding in VB and Proteus circuit. Besides that, during simulation it also requires more potentiometer to provide more input to display in GUI of CAN Bus dashboards.

Furthermore, in future can add some interact with this GUI display such as wireless network, SMS system and GPS navigator. By add this application; consumer can communicate to others during driving, without look at their phones. It will reduce percentage of car accidents. Besides that, by using GPS consumer can save time to find their destination.

REFERENCES

1. Kent How Teh, Wei Lun Ng, CheeKyun Ng, Nor KamariahNordin, " Home Appliance Management System Using Controller Araea Network", 2011
2. RenjunLi, Chu Liu, FengLuo, "A Design For Automotive CAN Bus Monitoring System", 2008.
3. RozehaA. Rashid, RohaizaYusoff, "BluetoothPerformance Analysis InPersonelAraeaNetwot (PAN)", 2006.
4. HUAQUN Guo, Jun JieAng, Yongdong Wu, " Extracting Controller Area Network Data For Reliable Car Communication".2009
5. GentianaLatiu, Octavian Cret, Lucia Vacariu, " Graphical User Interface Testing Optimization For Water Monitoring Applications".2013
6. Imran SarwarBajwa, Muhammad Abbas Chaudray." Alanguage Engineering System For GUI Desigb (Lesgui): A Rdule Based Approach"2006.
7. JiejieDai, HuiSong, " Design And Realization Of CAN Bus Vehicle Instrument Cluster Based On Microcontroller/OS-II".2011
8. Hui Chen, LinZhuan, "DesignOf Electrical Air- Condition Control System For Fuel Cell Vehicle Based On CAN Bus", 2010.
9. HongyanZhuo, Yingsong Song, Jiaru Zhang, "Research On Reliabi;ItyDesign Of DistrubutedMeasyrementAnd Control System Based On CAN Bus", 2011.
10. Liangxue Ding, Yi Shen, Yang Zhou, "Real Time Performance Analysis And Evaluation Of Can Bus With An Extended Petri Net Module",2009.
11. Wand Dian, Liu Jinhao, Zhang Bo," Research On Fault Diagnosis And Forecast System Of Forest Harvester Based On CAN Bus Information",2010.
12. A.Molina-Garcia, R. Torres, J.L. Munoz, N.Encinas, " Application Of Controller Area Network To Direct Load Control In Residential Areas",2007.

13. NaniFadzlinaNaim, Abdul Rahman Bin Daud, Ahmad IhsanMohdYassin, “; Graphical User Interface (GU) For Thumbprint Image Enhanceent And Minutiae Extraction”,2011.
14. Prof B. Venkatalakshmi, V. Saravan, G. Jenny Niveditha, “ Graphical User Interface For Enhanced Retinal Image Analysis For Diagnosing Diabetic Retinopathy”, 2011.
15. Marc Lavergne, “ Graphical User Interface For Next Generation Power System”, 2000.
16. Robert. A. Mayer , “GUS: A Graphical System For Capturing Structural Knowledge About Communications Networks”, 1988.
17. WaMingxia, Huang Qichun, Chen Qi, “ Graphical User Interface Based On Multidatabase Query Optimization”,2004
18. Chris Holt, TohnFumo, “ Visual Basic Environment Yields Advanced Navigation Simulation”
19. Elliot B. Sloane, “Strategic Graphic Dashboards For Improved Technology Management Decisions”, 2003.
20. Marvin S. Seppanen, “ Developing Industrial Strength Simulations Models Using Visual Basic For Applications (VBA)”, 2000.
21. Network topology diagram. Edward Diagrams Solutions.12 nov 2013.
22. Rahman, Z. A., Hasan, W. S., Noor, S. B., & Hassan, M. K. (2003). Speed trap detection with Doppler effect. doi:10.1109/SCORED.2003.1459693
23. Ali, M. L., Islam, R., Anom, A. S., & Amziah, M. Y. (2008). GUI for PC auto-shutdown.*Computers & Electrical Engineering*. doi:10.1016/j.compeleceng.2007.02.009
24. Zhang, L., & Zhang, H. (2008). An Introduction to Data Capturing. doi:10.1109/ISECS.2008.152
25. Guohui, Y., & Wei, G. (2011). Application Design of Data Packet Capturing Based on Sharpcap. doi:10.1109/CSO.2011.90
26. Shi, J., Bao, Y., Leng, F., & Yu, G. (2008). Study on Log-Based Change Data Capture and Handling Mechanism in Real-Time Data Warehouse. doi:10.1109/CSSE.2008.926
27. Chen, S., Zeng, K., & Mohapatra, P. (2011). Efficient data capturing for network forensics in cognitive radio networks. doi:10.1109/ICNP.2011.608904

28. Ran, P., Wang, B., & Wang, W. (2008). The design of communication convertor based on CAN bus. doi:10.1109/ICIT.2008.4608607
29. Retrieved from http://emanager.srmuniv.ac.in/elibrary/temp/CAN and CAN open.pdf

APPENDIX A

PIC COMPILER CODING

```
#include<16f877a.h>                                //Device
//#include <string.h>
#fuses xt,nowdt,protect,noput,nolvp               //Fuses
#device adc=10                                    //ADC 10-bit
#use delay (clock=4000000)                        //4.00 MHz
#use rs232 (Baud=9600, xmit=PIN_C6, rcv=PIN_C7, parity=N,bits=8,stream=PC)
//RS-232 parametres
#define use_portb_lcd TRUE                        //(only for printf function)
//#include"llcd_new.h"                            //LCD driver
void main()
{
float adcvalue, adcvalue1;
float c,d;
lcd_init();
set_tris_a(0b11111111);                           //RA0 analog
set_tris_c(0b00000000);
//set_tris_d(0b00000000);
setup_adc(adc_clock_internal);                    //internal clock
setup_adc_ports(all_analog);                      //ra0     //RA2 analog input
set_adc_channel(0);                               //  (0)        //set channel
set_adc_channel(1);
```

```
//set_adc_channel(0);
//delay_ms(10);
//value1=read_adc();
//set_adc_channel(1);
//delay_ms(10);
//value2=read_adc();
while(TRUE)
{
  set_adc_channel(0);                                    //  (0)          //set channel
  c=read_adc();                                          //voltage conversion
  delay_ms(100);
  set_adc_channel(1);
  d=read_adc();
  delay_ms(150);                                         //10-bit = 1024
  lcd_gotoxy(1,1);                                       //0-5 V and 1024 levels
  if (c < 922)
    output_high(PIN_D7);
  else
    output_low(PIN_D7);
  adcvalue = (c*5)/1024;
  adcvalue1 = (d*5)/1024;
  printf(lcd_putc,"Voltage = %1.2f V      ",(c*5)/1024);
  //fprintf(PC, (c*5/1024));
  //putc(adcvalue >> 8);
  //putc(adcvalue & 0xff);
  printf("T%f\n\r", adcvalue);
  printf("U%f\n\r", adcvalue1);
  delay_ms(100);
}
```

APPENDIX B

VISUAL BASIC CODING

```
Public Class Form1

    Dim stopclick As Boolean = False

    Private Sub Form1_Disposed(ByVal sender As Object, ByVal e As System.EventArgs)
        If SerialPort1.IsOpen() Then
            SerialPort1.Close()
        End If

    End Sub

    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SerialPort1.PortName = "COM1"
        SerialPort1.Open()
    End Sub

    Private Sub BtnClose_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles BtnClose.Click
        Dim response As MsgBoxResult
        response = MsgBox("Do you want to close form?", MsgBoxStyle.Question +
MsgBoxStyle.YesNo, "Confirm")
        If response = MsgBoxResult.Yes Then
            If SerialPort1.IsOpen() Then
                SerialPort1.Close()
            End If
            stopclick = True
            Me.Dispose()
        ElseIf response = MsgBoxResult.No Then
            Exit Sub
        End If
    End Sub

    Private Sub Form1_FormClosing(ByVal sender As Object, ByVal e As
System.Windows.Forms.FormClosingEventArgs) Handles Me.FormClosing

        Dim result As DialogResult
```

```
        result = MessageBox.Show("Are you sure you wish to close the program?", "Close
program?", MessageBoxButtons.YesNo)

        If SerialPort1.IsOpen() Then
            SerialPort1.Close()
        End If

        If result = Windows.Forms.DialogResult.Yes Then
            e.Cancel = False
        Else
            e.Cancel = True
        End If

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click

        'BtnForward.Enabled = False
        btnStop.Enabled = True
        BackgroundWorker1.RunWorkerAsync()

    End Sub


    Private Sub btnStop_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnStop.Click
        stopclick = True
    End Sub

    Private Sub BackgroundWorker1_DoWork(ByVal sender As System.Object, ByVal e
As System.ComponentModel.DoWorkEventArgs) Handles BackgroundWorker1.DoWork
        'Dim i As Long
        Dim s As String
        Dim s1 As String
        Dim stringComparer__1 As StringComparer = StringComparer.OrdinalIgnoreCase
        'Dim testCheck As Boolean
        Dim number As Decimal

        stopclick = False

        While Not stopclick

            ' If Not SerialPort1.IsOpen() Then
            ' SerialPort1.Open()
            'End If

            TextBox1.Text = ""
            'TextBox1.Refresh()
            'Do While i < 50
```

```
            'stopclick = True


            'SerialPort1.Write("<hello>")
            s = SerialPort1.ReadLine()
            s1 = s.Trim
            'If s = "T" Then
            'If stringComparer__1.Equals("T", s) Then ' The following statement returns True
(does "F" satisfy "F"?)
            If s1.Substring(0, 1) = "T" Then

                s1 = s.Trim
                s1 = s1.Substring(1, 4)
                'TextBox1.Text = TextBox1.Text + s1 + vbCrLf
                number = Convert.ToDecimal(s1)
                ArcScaleComponent1.Value = s1
                ' number =
            End If

            If s1.Substring(0, 1) = "U" Then

                s1 = s.Trim
                s1 = s1.Substring(1, 4)
                'TextBox1.Text = TextBox1.Text + s1 + vbCrLf
                number = Convert.ToDecimal(s1)
                ArcScaleComponent2.Value = s1
                ' number =
            End If
            'TextBox1.Text = TextBox1.Text + s + vbCrLf
            ' TextBox1.Refresh()
            'i = i + 1

        End While
    End Sub


    Private Sub BackgroundWorker1_RunWorkerCompleted(ByVal sender As Object,
ByVal e As System.ComponentModel.RunWorkerCompletedEventArgs) Handles
BackgroundWorker1.RunWorkerCompleted
        'BtnForward.Enabled = True
        btnStop.Enabled = False
    End Sub
End Class
```

APPENDIX C

PROTEUS CIRCUIT