MENU DEVELOPMENT FOR PORTABLE QIBLA FINDER USING GPS

NURULHUDA BINTI KAMARUDIN
Faculty of Electrical Engineering,
Universiti Teknologi MARA (UiTM),
40450 Shah Alam,
Selangor.
E – Mail: nurulhuda08@yahoo.com

ABSTRACT

Normally, people use a compass to determine the Qibla direction but in today developing era, there are so many thinkable solutions such as to design a portable Qibla finder using GPS. The scope of this project is to develop menu using assembly language to be appended into the PIC16F877A microcontroller that acts as the brain of the whole systems which consists of 5 modules which are the power supply module (5V DC power supply for operation), GPS module, controller unit module (PIC16F877A to store command), GLCD module (displaying text and graphics) and also the input module (tack switch for the navigation purposes). MPLAB IDE and PROTEUS VSM is used for assembling and debugging purposes.

Keywords
Graphic Liquid Crystal Display (GLCD), Global Positioning System (GPS), Integrated Circuit (IC), Liquid Crystal Display (LCD), Peripheral Interface Controller (PIC)

1.0 INTRODUCTION

Qibla (قبلة, translated as Qiblah, Kibla or Kiblah) is an Arabic word for the direction that should be faced when a Muslim prays but it has importance to more than just prayers [1]. Due to the development, this device came to picture as part of needs. Figure 1.1 shows the five main modules in this project which are the power supply module, controller unit module, GLCD display module, GPS module and input (keypads) module.

The input module is consist of a tack switch. The switch is for the navigation purposes. A 5V DC power supply is needed in operating all the components of the project. The power supply module comprises of direct power supply from the socket outlet and an adapter to convert the AC power supply to a DC power supply of 9V, which later will pass through a voltage regulator of 78L05 that produced a 5V of DC power supply as its output. The controller unit module is the heart of this project and it consist of a PIC16F877A microcontroller. Assembly language is used in writing the command. The microcontroller will stored the command in which will drive the GLCD module that consist of MGLS12864T graphic LCD to display the text and graphical output. MPLAB IDE and PROTEUS VSM is used for assembling and debugging purposes.

The GPS module will detect the user and automatically notify the user of the direction of Qibla by displaying it via the GLCD module. The microcontroller task is to interpret the data received from GPS module into output that can be displayed on the GLCD but the main intention of this project is the generation of the menu by which it involve the PIC16F877A microcontroller module, MGLS12864T GLCD module and input module (tack switch).

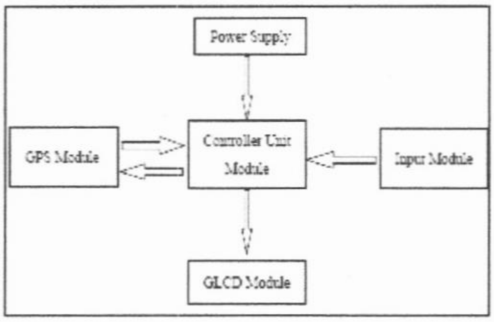

Figure 1.1: Block diagram of Qibla finder.

1.1 Properties of PIC 16F877A

Microcontroller contains program command code memory, data storage memory, bi-directional (input/output) ports and a clock oscillator. PICs are manufactured "empty", that is they are without program codes (commands) and cannot control circuit unless provided with program that tells them what to do. It is an EEPROM (electrically erasable programmable read only memory) device, which means it can be rapidly reprogrammed frequently without the need for ultra–violet erasing [2].

PIC has wide operating voltage range that is 2.0V to 5.5V, high sink/source current (25mA) and low power consumption.

1.2 Properties of GLCD

The GLCD used is the MGLS12864T which use Toshiba T6963C LCD controller. Figure 1.2 below shows the block diagram of the GLCD used.

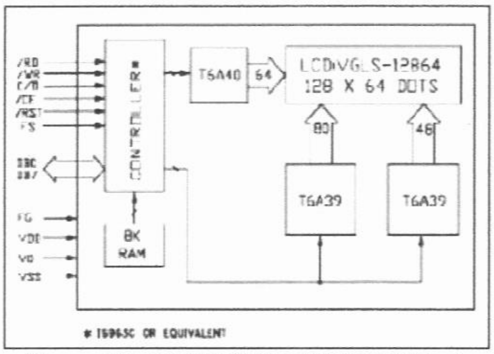

Figure 1.2: GLCD (MGLS12864T) block diagram

It is a 128 X 64 dot matrix LCD graphic module. There are 8 data lines (DB0–DB7) and 6 control lines (WR, RD, CE, C/D, RST and FS). The display has single positive supply line, VDD (recommended at 5V), two 0Vconnections. The entire pin out functions is shown in Table 1.1 while Table 1.2 shows syntaxes used in the assembler.

Table 1.1: GLCD pin out functions

Graphic LCD Pin Out Functions		
Pin	Symbol	Description
1	FG	Frame ground
2	VSS	Ground (0V).
3	VDD	Power supply for logic (+5V)
4	V0	Power supply for LCD drive.
5	WR	Data write. Write data into T6963C when WR="Low".
6	RD	Data read. Read data from T6963C when RD="Low".
7	CE	Chip enable for T6963C. CE must be "Low" when CPU communicates with T6963C.
8	C/D	WR=0: C/D=1: Command write, C/D=0: Data write RD=0: C/D=1: Status read, C/D=0: Data read
9	RST	"High": Normal (T6963C has an internal pull–up resistor). "Low": Initialize T6963C Text & graphic have addresses and text & graphic area settings are retained.
10–17	DB0–DB7	Data input / output (DB0=LSB, DB7=MSB)
18	FS	Font select. "High": 6X8 font. "Low": 8X8 font

Note 1: This pin is electrically connected to the metal bezel (frame). User can choose to connect this pin to VSS or leave it open.

Table 1.2: Command codes for PIC16F877

Command/Syntax		*Flags affected*	*Cycles*	*Description*
BYTE-ORIENTATED FILE REGISTER OPERATIONS				
ADDWF	f,d	C, DC, Z	1	Add W and f
ANDWF	f,d	Z	1	AND W with f
CLRF	f	Z	1	Clear f
CLRW	–	Z	1	Clear W
COMF	f,d	Z	1	Complement f
DECF	f,d	Z	1	Decrement f
DECFSZ	f,d	–	1 (2)	Decrement f, skip if 0
INCF	f,d	Z	1	Increment f
INCFSZ	f,d	–	1 (2)	Increment f, skip if 0
IORWF	f,d	Z	1	Inclusive OR W with f
MOVF	f,d	Z	1	Move f
MOVWF	f	–	1	Move W to f
NOP	–	–	1	No operation
RLF	f,d	C	1	Rotate left f through Carry
RRF	f,d	C	1	Rotate right f through Carry
SUBWF	f,d	C, DC, Z	1	Subtract W from f
SWAPF	f,d	–	1	Swap nibbles in f
XORWF	f,d	Z	1	Exclusive OR W with f
BIT-ORIENTATED REGISTER OPERATIONS				
BCF	f,b	–	1	Bit clear f
BSF	f,b	–	1	Bit set f
BTFSC	f,b	–	1 (2)	Bit test f, skip if 0
BTFSS	f,b	–	1 (2)	Bit test f, skip if 1
LITERAL AND CONTROL OPERATIONS				
ADDLW	k	C, DC, Z	1	Add literal and W
ANDLW	k	Z	1	AND literal with W
CALL	k	–	2	Call subroutine
CLRWDT	–	TO, PD	1	Clear Watchdog Timer
GOTO	k	–	2	Go to address
IORLW	k	Z	1	Inclusive OR literal with W
MOVLW	k	–	1	Move literal to W
RETFIE	–	–	2	Return from interrupt
RETLW	k	–	2	Return with literal in W
RETURN	–	–	2	Return from subroutine
SLEEP	–	TO, PD	1	Go into standby mode
SUBLW	k	C, DC, Z	1	Subtract W from literal
XORLW	k	Z	1	Exclusive OR literal with W

SCOPE OF WORK

2.1 Objectives

There are several objectives in order to complete this project, which are as follow to construct the program codes (commands) for the PIC 16F877A to display the menu for the Qibla finder, run simulation of the circuit to check the functionality and to observe the output and analyze the output from the simulation.

2.2 Programming Using MPLAB IDE

Program codes (assembly language) is a language that man and microcontroller use to communicate. The process is shown in Figure 2.1. An assembly language, consists of alphabet signs and words, is written in a text processing (editor) program, according to the assembler rules. The text processing program which capable of producing an ASCII file, such as MPLAB, will then translated the assembly language into machine language, that consists "0" and "1", so that the PIC can understand it. The execution file (often noted as the HEX file) is the place where this translated assembly language is found.

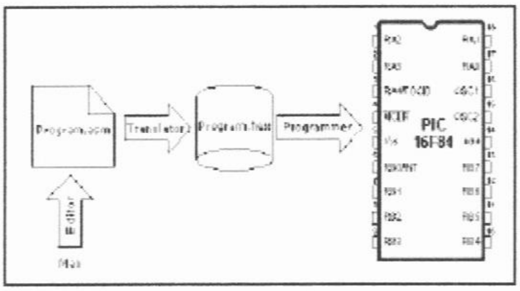

Figure 2.1: MGLS12864T GLCD block diagram

2.3 Debugging Using PROTEUS VSM

The circuit which connects the PIC 16F877A to the MGLS12864T GLCD, shown in Figure 2.2, was constructed and the commands wrote before was implemented into the PIC in the simulator program. Running the simulator will prove whether the circuit, as well as the implemented commands, are giving the right output as desired.

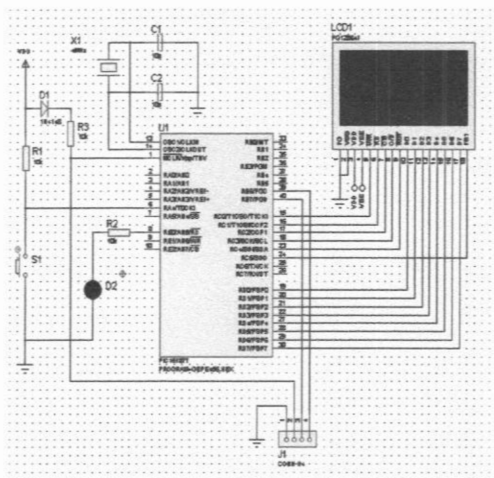

Figure 2.2: Schematic diagram of the connection between the PIC and the GLCD

In order to have a clear explanation about the development of this project, it is important to plan the arrangement and organization of the thesis. This project consists of 5 chapters by which each chapters begins with the introduction, followed by interpretation that associate with each chapter's scope. Chapter 1 exhibits the overview of the project that includes the problem argument, significance of the project, the objectives of the project and a glimpse of the scope of work. Chapter 2 cover the literature review of the project. Chapter 3 is focusing on the methodology, by means the software design in constructing the source code for the menu. Chapter 4 emphasis the results, together with discussion for each results obtained throughout the completion of this project. The final chapter, chapter 5 will consists the conclusion and the future development of the project.

3.0 METHODOLOGY

The software development is done using MPLAB and PROTEUS softwares. The processed of the software generation is shown in Figure 3.1 below.

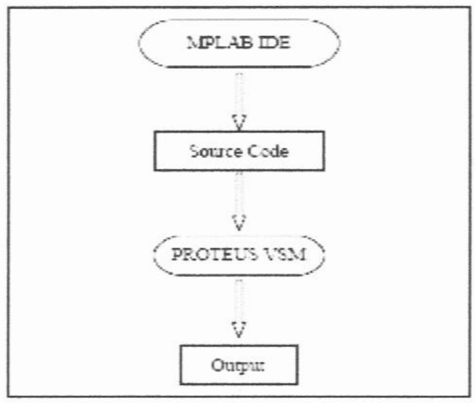

Figure 3.1: Flow diagram of software generation

3.1 Source Code Generation Using MPLAB IDE

Before constructing the program code, it is important to understand the flow of the displaying the menu.

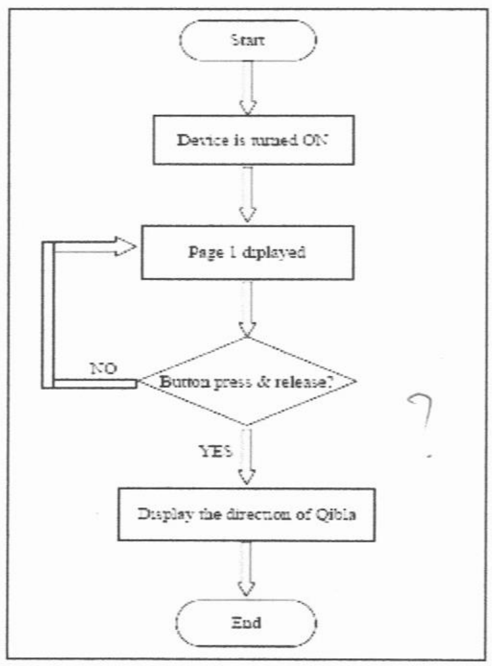

Figure 3.2: Flow diagram of how the menu will be displayed

The flow diagram as in Figure 3.2 shows the overall view on how the menu will be displayed. When the device is turned on, the welcoming page which is the page 1 is displayed. When the button was press and then release, page 2 will appeared. Page 2 will displayed the Qibla direction.

MPLAB is a text processing program which will translate the assembly language into machine language so that the microcontroller can understand it. Executing file (*.HEX), Program errors file (*.ERR) and List file (*.LST) are the results of the process of translating a program written in assembler language. [4]

In order to prepare the program for loading into microcontroller, few basic steps have to be considered which are designing a project, writing the program and converting to zero-one code comprehensible by microcontroller (i.e. compiling). Figure 3.3 shows the flow of building the source code.

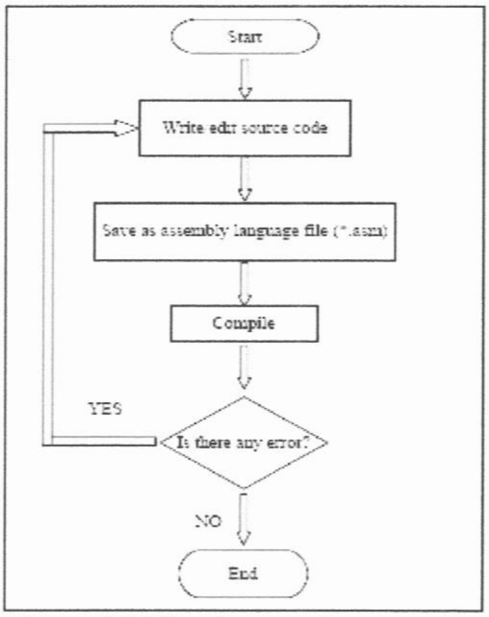

Figure 3.3: Flow diagram of source code generation using MPLAB IDE

Flow diagram in Figure 3.4 show the flow of main routine for the project. Firstly, all the STATUS registers and ports should be defined. Then, a set of data held in table form (CGTABLE) is read and stored in user–defined CG RAM. The characters created are stored in consecutive CG RAM locations. Data will be written in several subroutines to ensure that it is placed at the exact screen addresses required. Next, the program will pass through the GRAPHIC routine which consist of SETUP subroutine. Last but not least, the program will continue and call the TRY routine which consist the subroutines that build the pages of menu displayed. Finally, when all the subroutines in the TRY routine were completely gone through, the process will go to GRAPHIC routine again to end the flow before the process starts all over again.

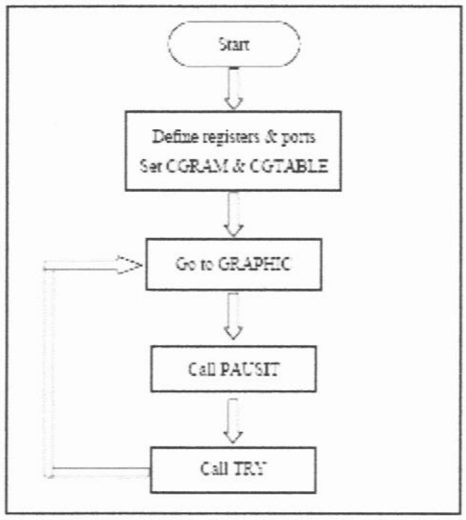

Figure 3.4: Flow diagram of main routine

Figure 3.5 until Figure 3.11 shows parts of the flow diagram of the subroutines.

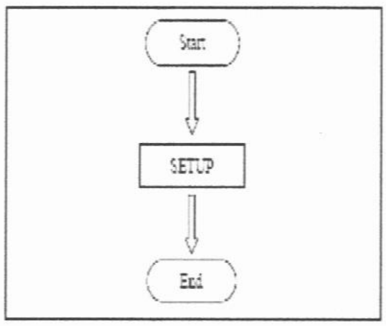

Figure 3.5: Flow diagram for GRAPHIC routine

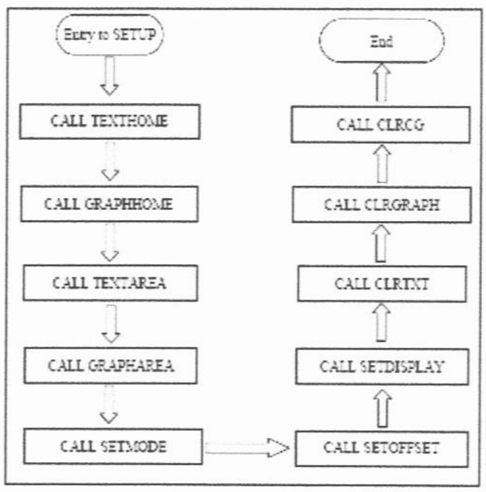

Figure 3.6: Flow diagram of SETUP routine

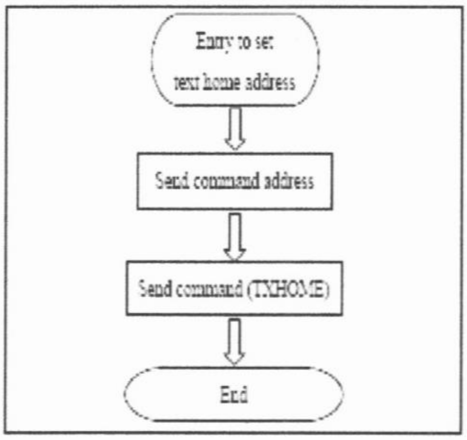

Figure 3.7: Flow diagram of TEXTHOME routine

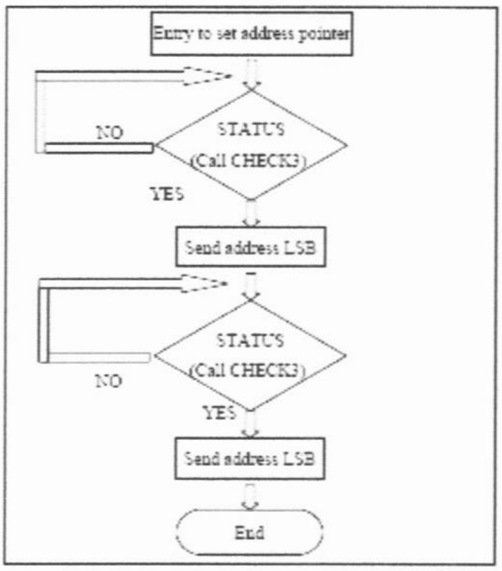

Figure 3.8: Flow diagram of CMDADR routine

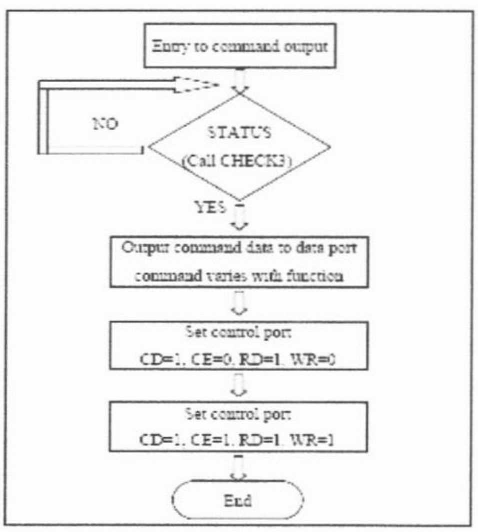

Figure 3.9: Flow diagram of SENDCMD routine

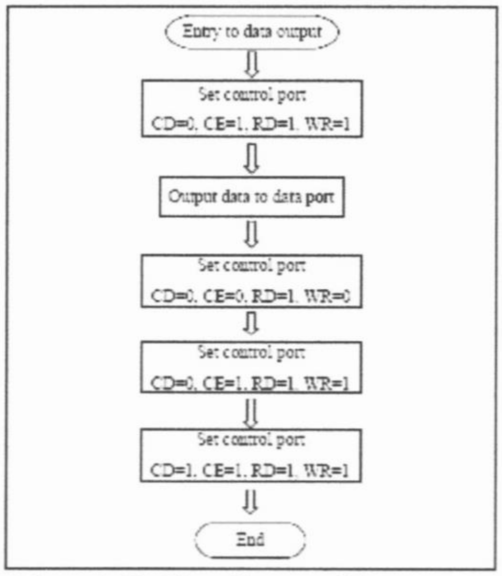

Figure 3.10: Flow diagram of OUTDATA

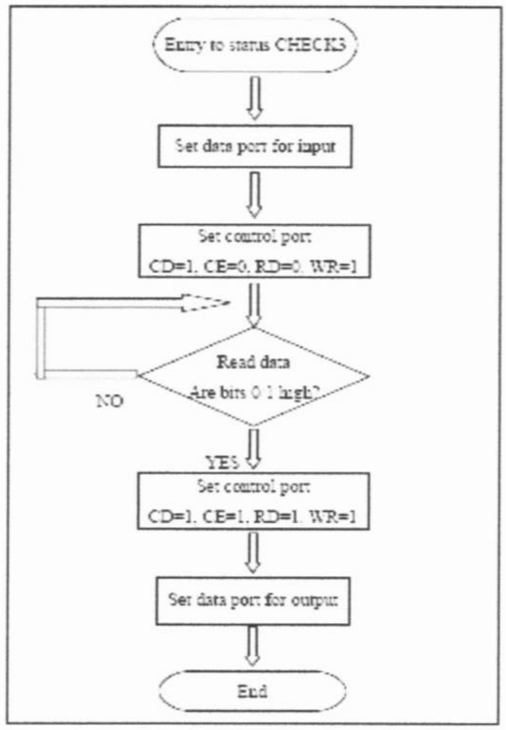

Figure 3.11: Flow diagram of CHECK3

3.2 Debugging Using PROTEUS VSM

To test the functionality of the command which was successfully build using MPLAB IDE, PROTEUS VSM is used. The circuit used in the program is based on the Everyday Practical Electronics (E. P. E.) development kits which is written by John Becker on "How to Use Graphic Liquid Crystal Displays with PICs". It is the free supplement provided by E. P. E. in February 2001 Special Supplement. The source code is embedded in the "*.asm" file as the command for the PIC16F877A in the circuit. Then, the instruction set will be built. After the output window appear with the word "Build completed", it means that the next step can be proceeded. The next step is to run or debug the circuit, in which the source code had been embedded before. The final step is to observe the displayed output on the GLCD. If it is displaying as desired, then the source code generation of the menu development the portable Qibla finder menu is a success.

4.0 RESULTS AND DISCUSSIONS

4.1 Source Code Generation Using MPLAB IDE

The first step taken was to grab-hold the functionality of the assembler. So, a simple program consisting words only, was written. The aim is to get a page that displaying the words of " Testing For My Final Year Project2 Portable Qibla Finder". The source code was written and assembled.

Figure 4.1 shows the result which consists of the project window, a glimpse of the command, and the output window. After successfully write the command, it will later being assemble. The output window that will pop-up, is then observe. If the result is as appeared in Figure 4.1 which stated that "BUILD SUCCEEDED", then the second step can be done.

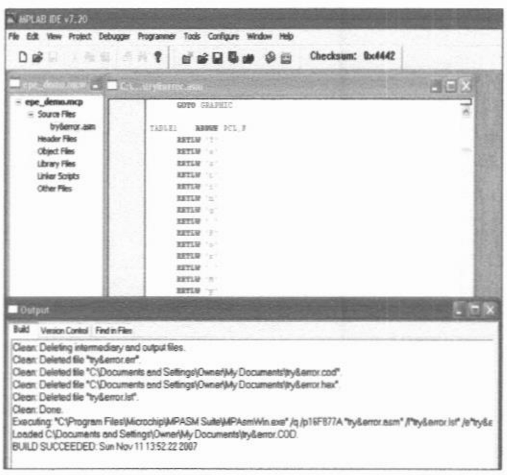

Figure 4.1: Result for first trial (using PROTEUS VSM)

Since the first try was a success, the building of the real source code for the menu display of the portable Qibla finder is carried on. The aim this time is to display both words and graphics by the GLCD. This source code is target for page one of the menu which is the greeting page. The concept is the same, which are writing the command and later, assemble it. Finally, observed the output window. So, same goes for this time. The command wrote will be assemble, and the output window is to be observed. The result is as in Figure 4.2. As the previous one, the program code is also successfully build.

Figure 4.2: Real result of first page (using MPLAB IDE)

4.2 Source Code Debugging Using PROTEUS VSM

Each source codes which were successfully build using MPLAB IDE have had to be ran using PROTEUS VSM in order to ensure that the functionality of each commands by observing the output displayed on the virtual GLCD. To debug the command written, it will be inserted into the PIC in the circuit. The successfully builded source code is append into the circuit. Then, the circuit will be assembled. A pop-up output window will appear to notify whether or not it is build completely.

The output of the first trial which aim the display of " Testing For My Final Year Project2 Portable Qibla Finder" is shown in Figure 4.5. As will be seen, the output window stated that the circuit which have been appended with the command is successfully build and the debug data is extracted. From the GLCD in the circuit, it is obvious that the source code is functioning very well and appeared is the desired words.

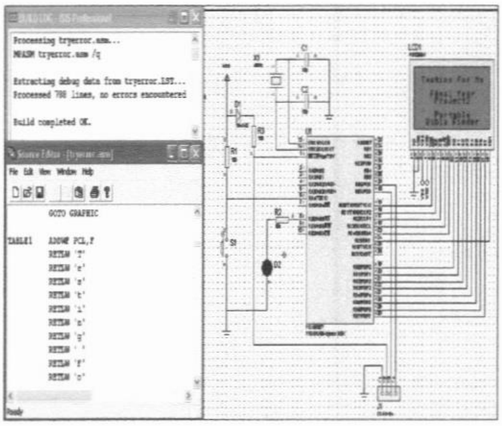

Figure 4.5: Result for first trial (using PROTEUS VSM)

Since the first try was a real success, the debugging of the real source code for the menu display of the portable Qibla finder which also been successfully build before is carried on. The aim is to display both words and graphics by the GLCD and this source code is target for page one of the menu which is the greeting page. Same as the previous one, the command needs to be append into the circuit. So, the command is to be add as the new source code. Figure 4.6 shows the output of the command and it is definitely confirmed that this source code is also functioning very well, just like before. The output window stated that the debug data is extracted, no errors encountered and most importantly is, the building is completed. The GLCD displaying as desired.

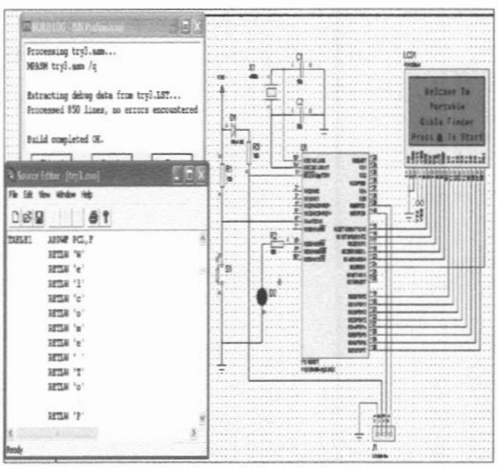

Figure 4.6: Real result of first page (using PROTEUS VSM)

5.0 CONCLUSIONS

The main aim is to construct the program codes (commands) for the PIC16F877A to display the menu for the Qibla finder, run simulation of the circuit to check the functionality, observe the output and analyze the output from the simulation. The source codes are successfully build with no errors. The debugging process is also completed with no errors and the output displayed on the GLCD is also as desired. The objectives of this project are achieved but the completion level of this project is around 85%. This is because the generation of command for the most important part, that is the page to display the direction of the Qibla still fail to run as coveted due to time constraint and have not yet had a full grab on understanding how the other complex syntaxes works.

It can be concluded that the menu development of the portable Qibla finder using GPS depends on the source code appended into the PIC16F877A microcontroller.

Upon the accomplishment of this project, the knowledge and functions on PIC16F877A microcontroller, MGLS12864T GLCD, softwares such as MPLAB IDE and PROTEUS VSM as well as the skill in writing program code using assembly language is gained. The hardship in completing this project also encourage the experience of not to give-up easily and how to work under pressure.

6.0 FUTURE DEVELOPMENT

There are a few recommendations that can be done as for future developments. First, this project should be continued with the hardware development. The circuit should consider to build the device with a built-in, rechargeable battery as it power supply, a ON/OFF button since this circuit was connected directly to the source and there is no need of turning the device on for all day long.

As a second recommendation, the circuit can make the displayed output to speed up by taking a larger value of the crystal clock frequency. This can enhance the performance of moving display graphics such as waveforms or makes objects to move or even movement of words.

Thirdly, the assembler software used (i.e. MPLAB IDE) should have its own library of source codes for the PIC microcontrollers to drive other electronic devices such as the LCD, GPS receiver or maybe a micro SD card for instance. This is to save the user's time in writing the command if the library is already there.

This device can also be combined with a prayer time so that the user not only knows where to head for their prayers but also knows the exact time of prayers as advance development.

7.0 ACKNOWLEDGEMENT

First of all, praise to **Allah s.w.t** for His blessing upon the author who for the strength and ability to complete this project and report. The author would like to extend her appreciation to Mr. Abdul Karimi Halim for his judgment, guidance and precious advice upon completing this project. Finally, the author would like to thank her family and friends for their support and encouragement. This project would not be possible without their helps.

8.0 REFERENCES

[1] "Qibla".
Internet: http://en.wikipedia.org/wiki/Qibla

[2] John Baker. "EPE PIC Tutorial V2 Part 1". Everyday Practical Electronics, April 2003.

[3] John Baker. "How To Use Graphics Liquid Crystal Displays With PICs". Everyday Practical Electronics, February 2001.

[4]"MikroElektronika:Books:Microcontroller".
Internet:
http://www.mikroe.com/en/books/picbook/picbook.htm