" I admit that I had read this report and in my opinion, this report had fulfilled all scope and quality for the Bachelor Degree of Electronic Engineering (Industrial Electronic)

Signature : ....

Supervisor : SOO YEW GUAN

Date : 8/5/06

MICROMOUSE: AN AUTONOMOUS MAZE SOLVING ROBOT

JHONATHAN MARK DANKER

This Report is Submitted in Partial Fulfillment of Requirements for The Bachelor Degree of Electronic Engineering (Industrial Electronic)

Fakulti Kejuruteraan Elektronik dan Kejuruteraan Komputer
Kolej Universiti Teknikal Kebangsaan Malaysia

APRIL 2006

"I hereby declared that this report is a result of my own work except for the works that have been cited clearly in the references."

Signature : ....

Student : Shaudthan Danker ....

Date : 8-5-2006 ....

For Diana my love

ACKNOWLEDGEMENT

First of all I would like to thank Mr. Soo Yew Guan, my supervisor, for giving me the opportunity to undertake this project and also for helping me out throughout the entire span of the project either by advice or referrals. Secondly would like to thank the Faculty of Electronic and Computer Engineering for making this subject available and also providing the facilities that have help me complete this project. I would also like to thank the technician at the PSM lab that has assisted me in fabricating my PCB. Not forgetting, I would like to thank also my parent s for some financial backing regarding the project. And lastly I would like thank all my friends that have provided some input from time to time.

ABSTRACT

A micromouse is a small microprocessor-controlled robot vehicle, which is able to navigate its way through an unknown maze and to negotiate this path in the shortest time. The main challenge for the micromouse handler is to impart to the micromouse an adaptive intelligence, to explore different maze configuration and to work out the optimum route for the shortest run time from start to destination; and to design a reliable micromouse which will negotiate the maze at very high speed without crashing into the walls of the maze. The micromouse will consist if four section, the drive section, the sensor section, the control section and also the power source. The power source will have to be capable to provide power for the drive system, sensors and the controller. The drive section will consist of motor to move the micromouse, and feedback to sense the distance the micromouse has travel. The sensors will employ the use of infrared sensor for line detection, and ultrasonic sensor for obstacle avoiding. Lastly, microcontroller (s) will be used as the controller for the micromouse, preferably the MC9S12C32 microcontroller from Freescale Semiconductor.

ABSTRAK

Micromouse adalah robot mobil yang dikawal oleh pemproses mikro, yang berkemampuan menyelesaikan *maze* yang tidak dikenali dalam masa yang paling singkat. Cabaran utama micromouse adalah untuk mengadaptasikan micromouse dengan kepintaran untuk menjelajahi konfigurasi *maze* yang berlainan dan mendapatkan laluan yang paling optimum untuk sampai ke destinasi; dan juga untuk mereka bentuk micromouse yang dapat menyelesaikan *maze* pada kelajuan yang tinggi tanpa melanggar dinding *maze*. Micromouse ini terdiri daripada empat bahagian, bahagian pemacu, bahagian pengesan, bahagian pengawal dan bahagian bekalan kuasa. Bekalan kuasa mesti mempunyai kemampuan untuk membekalkan kuasa untuk sistem pemacu, pengesan dan pengawal. Bahagian pemacu terdiri daripada motor untuk mengerakan micromouse tersebut dan suap balik untuk mengesan jarak yang telah dilalui. Micromouse ini akan mengunakan pengesan inframerah untuk mengesan garisan dan pengesan ultrasonic untuk mengelak dari halangan. Akhirnya pengawal micro yang akan digunakan sebagai pengawal dalam micromouse adalah pengawal MC9S12C32 daripada Freescale Semiconductor.

TABLE OF CONTENTS

CHAPTER	CONTENT	PAGE
	PROJECT TITLE	**i**
	CONFESSION	**ii**
	DEDICATION	**iii**
	ACKNOWLEDGEMENT	**iv**
	ABSTRACT	**v**
	ABSTRAK	**vi**
	CONTENTS	**vii**
	LIST OF TABLES	**xi**
	LIST OF FIGURES	**xii**
I	**INTRODUCTION**	**1**
	1.1 ORIGIN OF MICROMOUSE	1
	1.2 PROJECT OBJECTIVES	3
	1.3 SCOPE OF WORK	3
	1.4 THESIS ORGANIZATION	4
II	**LITERATURE REVIEW**	
	2.1 THE MOUSE	5
	2.2 THE CONTROLLER	6
	2.2.1 The MCU	6
	2.2.2 Metrowerk Codewarrior IDE	7

2.3	THE DRIVE SYSTEM	11
	2.3.1 Displacement	11
	2.3.2 Velocity	12
	2.3.3 Acceleration	12
	2.3.4 The motor	13
	2.3.4.1 Stepper Motor	13
2.4	SENSOR	16
	2.4.1 Purpose	16
	2.4.2 Sensor Types	17
	2.4.2.1 Range Finding	17
	2.4.2.2 IR Reflective Intensity	17
	2.4.2.3 IR Triangulation	18
	2.4.2.4 Ultrasound Time-of-Flight	19
	2.4.2.5 Existence	20
	2.4.2.6 IR Reflective	20
	2.4.2.7 IR Interrupt	21
2.5	POWER SUPPLY	22
	2.5.1 Chemistries	22
	2.5.2 Voltage Regulation	23
	2.5.3 Logic Voltage Regulation	24
	2.5.4 Energy Calculation	24
	2.5.5 Charging and Discharging	25
2.6	CHASIS	26
	2.6.1 Materials	26
	2.6.1.1 Metal	27
	2.6.1.2 Plastic	27
	2.6.1.3 Wood	28
	2.6.2 Physics	28
2.7	OTHER MICROMICE	29
	2.7.1 Enterprise	29
	2.7.2 Voyager	32

III	**PROJECT METHODOLOGY**		**35**
	3.1	METHODOLOGY: GENERAL APPROACH	35
	3.2	LITERATURE REVIEW	37
	3.3	HARDWARE DESIGN	37
	3.4	SOFTWARE DESIGN	37
	3.5	FINALIZING	38
IV	**HARDWARE SETUP**		**39**
	4.1	CHASIS	39
	4.2	DRIVE SYSTEM	40
		4.2.1 The Stepper Motor	40
		4.2.2 Stepper Motor Driver	41
		4.2.3 Electrical Setup Of Drive System	45
	4.3	SENSOR SYSTEM	45
		4.3.1 Infrared Sensors	46
		4.3.1.1 Electrical Setup for IR sensors	48
		4.3.2 Ultrasonic Sensors	49
		4.3.3 Sensor Layout on the Mouse	49
	4.4	THE CONTROLLER	50
		4.4.1 Microcontroller Setup	51
		4.4.2 Electrical Setup of the controller	52
	4.5	POWER SUPPLY	54
		4.5.1 Signal Flow for micromouse	54
V	**SOFTWARE DESIGN**		**55**
	5.1	INTRODUCTION TO THE PROGRAMMING	55
	5.2	THE SUBROUTINES	56
		5.2.1 Pulse Motor	56
		5.2.2 Delay	56
		5.2.3 ATD sample and Store	57

	5.2.4	Turn Left 90°	59
	5.2.5	Turn Right 90°	59
	5.2.6	Turn 180°	60
	5.2.7	Alignment	60
	5.2.8	Resume	62
	5.3	THE MAIN PROGRAMM	62
	5.3.1	The Wall Following Algorithm	63
	5.3.2	The Flood Fill Algorithm	66
		5.3.2.1 Mapping The Maze	67
		5.2.3.2 The Algorithm	68
		5.2.3.2.1 Flags	68
		5.2.3.2.2 Programming Flow Chart	69
VI		**CONCLUSION**	**74**
	6.1	CONCLUSION	74
	6.2	FUTURE WORK	75
		REFERENCES	**76**

LIST OF TABLES

NO	TITLE	PAGE
2.1	Types of Batteries	22
4.1	Sensor Test Results	47
4.2	Microcontroller I/O assignment	51
4.3	BDM Multilink I/O assignment	52
5.1	Direction Flag conversion	68

LIST OF FIGURES

NO	TITLE	PAGE
2.1	MC9S12C32 Block Diagram	7
2.2	MC9S12C32 BDM Multilink Setup	8
2.3	Code Warrior user interface	9
2.4	Simulator Mode user interface	10
2.5	Stepper Motor Schematic	13
2.6	Top View Of Enterprise	30
2.7	Back View Of Enterprise	31
2.8	Side View Of Enterprise	31
2.9	Top View Of Voyager	33
2.10	Side View Of Voyager	33
2.11	Back View Of Voyager	34
3.1	Project Methodology	36
4.1	Stepper Motor Wire Configuration	40
4.2	Drive System Electrical Setup	45
4.3	Difference between Conventional Maze and the Maze of Line	46
4.4	Line Detection	47
4.5	Electrical Setup of IR sensors	48
4.6	Sensor layout under the micromouse	49
4.7	Actual Layout of the sensors	50
4.8	MCU wiring Diagram	52

4.9	BDM header schematic	53
4.10	Actual MCU Board	53
4.11	Signal Flow for the micromouse	54
5.1	Wall Following Algorithm Flow Chart	64 – 65
5.2	Cell values of the Maze store in the Memory in the Beginning	67
5.3	Pointer to the cell values of the Maze	67
5.4	The Flood Fill Algorithm	70 - 73

CHAPTER I

INTRODUCTION

1.1 ORIGIN OF MICROMOUSE

The micromouse competition has been running since the late 1970s around the world. In 1977, the IEEE Spectrum magazine introduced the concept of the micromouse. In May 1977, Spectrum announced the 'Amazing Micromouse Competition' which would be held in 1979 in New York. There were 15 competitors running out of around 6000 initial entries. This competition involved mice finding their way out of a 10" by 10" maze. When the competition was held, the winner was a high-speed, dumb wall follower.

In 1980, Professor John Billinsley, of Portsmouth Polytechnic, modified the rules and introduced the first European competition - held in London at Euromicro. The rule changes required the mice to find a goal in the centre of the maze and wall followers could be prevented from finding the goal. There were 200 enquiries and 100 entries, but only 9 mice at the finals. Nick Smith's Sterling Mouse became the first ever (and that year the only) micromouse to find the centre and know it had done so. Although performance was less than stunning at about 0.18m/s, it was and still is a remarkable feat.

Essentially, you have a wooden maze made up of a 16 by 16 grid of cells. Mice must find their way from a predetermined starting position to the central area of the maze unaided. The mouse will need to keep track of where it is, discover walls as it explores, map out the maze and detect when it has reached the goal. The winner is the mouse that manages this the fastest. There are many versions of the full rules on-line and there are a number of minor variations on how the score of the mouse is determined.

Although modern micromice are relatively sophisticated, this is an extremely challenging undertaking. One of the earliest mice now about 20 years old is still regularly entered in competitions and puts up a very respectable show.

The micromouse competition appears to have waned a little in popularity in recent years. The UK events haven't attracted very large entries. There still seems to be plenty of interest in the US and Japan.

Since the event is very competitive, there seems to be something of a lack of useful information available on the Internet. While there are many sites, and search engines produce plenty of hits, real details beyond general descriptions and patchy progress reports are few and far between.

1.2 PROJECT OBJECTIVES

The objectives of this project is to study the development of an efficient drive system that will be able to travel in a straight line and make a 90° turn without offsetting, to study the development of a tracking and navigating system that is able to record and remember the journey of the mouse, to develop the ability to sense its surrounding and map its location, to investigate potential algorithms able to solve the maze in the shortest time, to investigate the ability of the micromouse to memorize the track that it already traveled/explored and calculated the shortest path to take, also it should be able to retrace its steps and to develop an efficient power system so that the micromouse can start and finish the maze with a single set of batteries.

1.3 SCOPE OF WORK

The scope of work for this project is divided into 2 parts, the hardware and the software. The hardware involves designing the mechanical parts and the electrical parts of the mouse. The mechanical part involves the chassis, the drive system, the sensors layout and the electrical part involves the microcontroller, driver circuit and sensor interfaces.

The software involve designing and writing a program for the mouse to move forward, backward, turn left and right. Also to develop an algorithm for tracking, navigating, mapping and solving the maze.

1.4 THESIS ORGANIZATION

This thesis comprises of five chapters. Chapter II discusses the literature background of a Micromouse. Explanation of subsystems that make up the Micromouse are discusses here.

Chapter III discusses the methodology used in realizing this project. Chapter IV discusses the hardware setup for this project. Explanation of each subsystem used in this project will be explained in detailed here.

Software design is elaborated in Chapter V. The program for running the mouse and the algorithm use for solving the maze are explained here.

Finally Chapter VI gives the overall conclusion and the possibility for future works regarding the project.

CHAPTER II

LITERATURE REVIEW

2.1 THE MOUSE

The micromouse will basically made up of four main systems. The controller which is the brains of the mouse makes all the decision in moving the mouse and solving the maze. The drive system which is the legs of the mouse moves the mouse anywhere the mouse wants to go. The sensors system, the eyes of the mouse help the mouse to navigate in the maze. Lastly the power supply which is the heart of the mouse, pumps power to all the other systems.

2.2 THE CONTROLLER

2.2.1 The MCU

In this project, I will use the MC9S12C32 microcontroller from Freescale Semiconductors to control the mouse. The MC9S12C32 microcontroller unit is a 16-bit device composed of standard on-chip peripherals including a 16-bit central processing unit (HCS12 CPU), 32K bytes of Flash EEPROM, 2K bytes of RAM, one asynchronous serial communications interfaces (SCI), one serial peripheral interfaces (SPI), an 8-channel IC/OC enhanced capture timer, one 8-channel, 10-bit analog-to-digital converters (ADC), an 6-channel pulse-width modulator (PWM), 35 discrete digital I/O channels, 5 discrete digital I/O lines with interrupt capability. The MC9S12C32 has full 16-bit data paths throughout. However, the external bus can operate in an 8-bit narrow mode so single 8-bit wide memory can be interfaced for lower cost systems. The inclusion of a PLL circuit allows power consumption and performance to be adjusted to suit operational requirements. Figure 1 shows the Block Diagram form the MC9S12C32 device. The device comes in 52, 48 pin low profile quad flat package (LQFP) or 80-pin quad flat package (QFP).

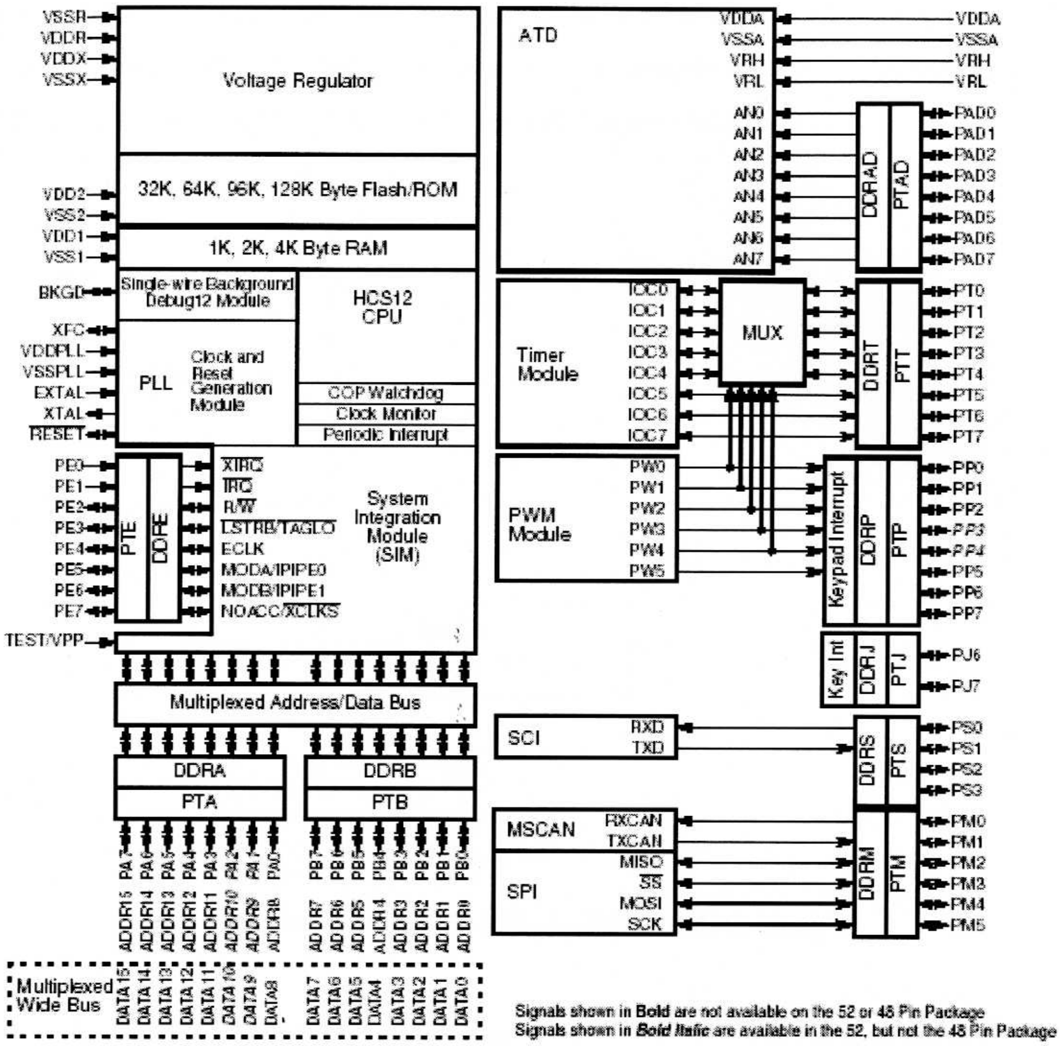

Figure 2.1: MC9S12C32 Block Diagram

2.2.2 Metrowerks CodeWarrior IDE

The CodeWarrior IDE is software developed by Metrowerks for programming purposes with microcontrollers. It is compatible with the MC9S12C32 and with the use of this software, it is possible to write the assembly code, compile it and then download it to the MCU through the USB port of a computer which is connected to the MC9S12C32 through the BDM link. See block diagram below:

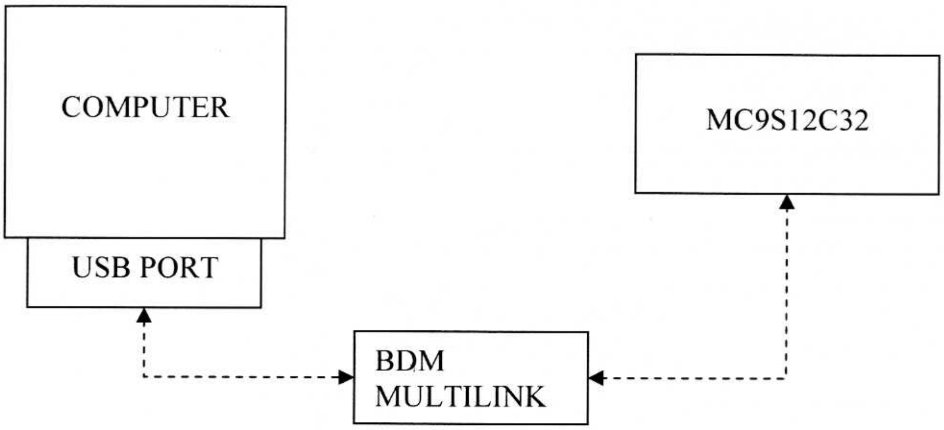

Figure 2.2: MC9S12C32 BDM Multilink set up

The CodeWarrior provides debugging functionality when running the code on the 9S12C32 and also simulator mode to be used when running the code on the computer. The diagram below depicts the CodeWarrior graphical user interface:

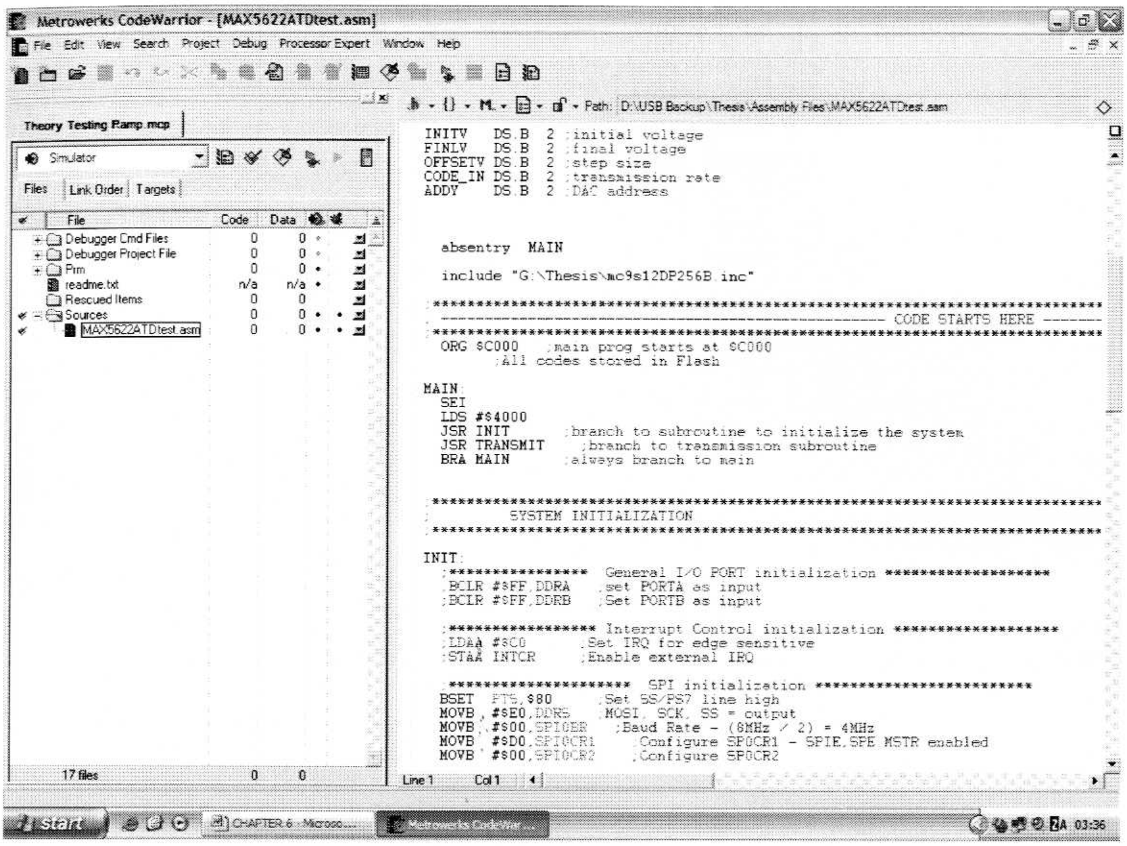

Figure 2.3: CodeWarrior user interface

In the absence of the 9S12C32, it is possible to debug the assembly code in the simulator mode of the CodeWarrior. The software builds a simulation of the microcontroller with the compiled assembly code and the user may debug the code on the computer with windows showing the current state of the device memory locations and registers. The diagram below shows the CodeWarrior user interface when debugging in simulator mode:

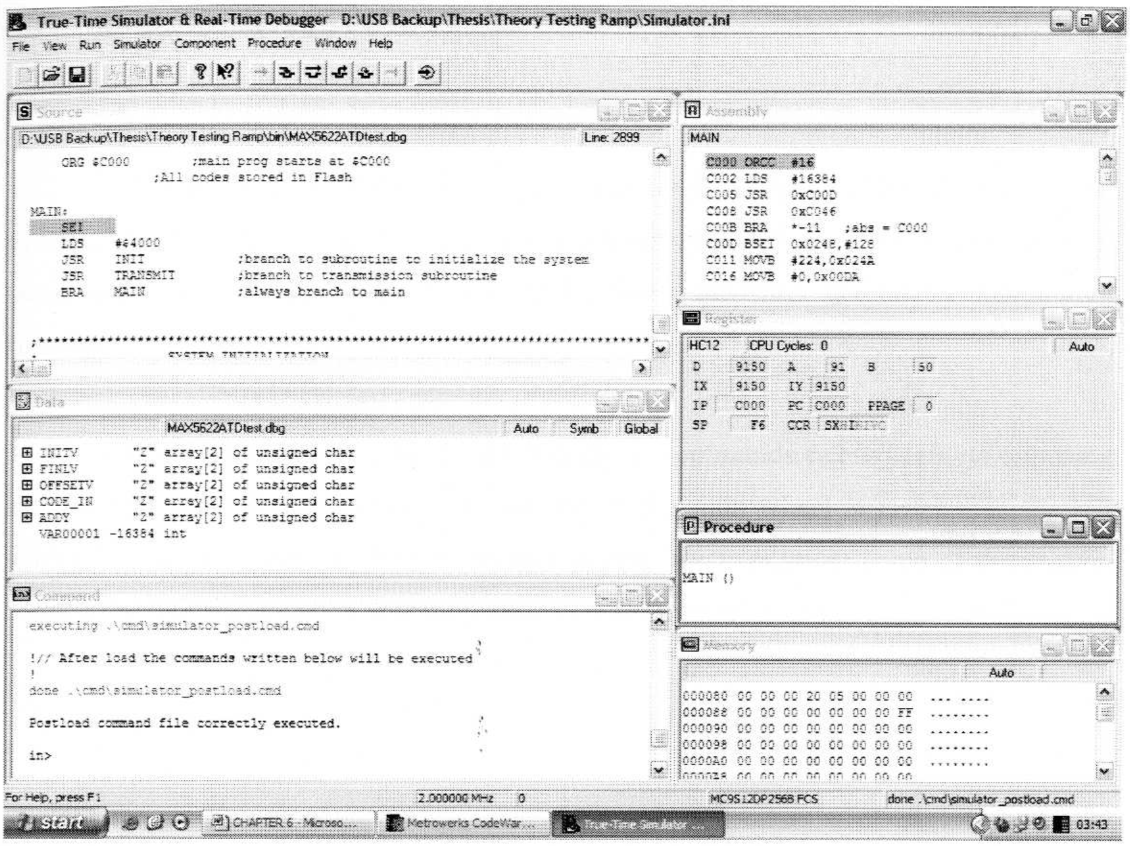

Figure 2.4: Simulator mode user interface

When debugging the assembly code that has been downloaded onto the 9S12C32, the interface also provides the user with information of the MCU memory locations and registers as well similar to the simulator mode. The difference in this mode is that the code is executed on the MCU itself and is being debugged on the computer. The user interface of the real-time debugger is also the same as the simulator mode.