UNIVERSITI MALAYSIA PAHANG

DECLARATION OF THESIS AND COPYRIGHT

Author's full name : **RAJA SARAVANA KUMAR A/L SELVAKUMAR**

Date of birth : **17TH NOVEMBER 1985**

Title : **DESIGN OF MULTIPLE MICROCONTROLLERS PLATFORM FOR USE IN ACADEMIC ENVIRONMENT**

Academic Session : **2011 / 2012**

I declare that this thesis is classified as:

☐	**CONFIDENTIAL**	**(**Contains confidential information under the Official Secret Act 1972)*
☐	**RESTRICTED**	**(**Contains restricted information as specified by the organization where research was done)*
√	**OPEN ACCESS**	I agree that my thesis to be published as online open access (Full text)

I acknowledge that Universiti Malaysia Pahang reserve the right as follows:

1. The Thesis is the Property of Universiti Malaysia Pahang.
2. The Library of Universiti Malaysia Pahang has the right to make copies for the purpose of research only.
3. The Library has the right to make the copies of the thesis for academic exchange.

Certified By:

(Signature of Student)	(Signature of Supervisor)
851117-10-5945	**DR. KAMARUL HAWARI GHAZALI**
New IC / Passport Number	Name of Supervisor
Date : **9TH JULY 2012**	Date: **9TH JULY 2012**

NOTES : *If the thesis is CONFIDENTIAL or RESTRICTED, please attach with the letter from the organization with period and reason for confidentiality or restriction.

DESIGN OF MULTIPLE MICROCONTROLLERS PLATFORM FOR USE IN ACADEMIC ENVIRONMENT

RAJA SARAVANA KUMAR A/L SELVAKUMAR

Thesis submitted in fulfilment of the requirements
For the award of the degree of
Master of Engineering in Electronics

Faculty of Electrical & Electronics Engineering
UNIVERSITI MALAYSIA PAHANG

JULY 2012

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Engineering in Electronics.

Signature	:	______________________________
Name of Supervisor	:	DR. KAMARUL HAWARI BIN GHAZALI
Position	:	SENIOR LECTURER, FACULTY OF ELECTRICAL & ELECTRONICS ENGINEERING, UNIVERSITI MALAYSIA PAHANG
Date	:	9TH JULY 2012

Signature	:	______________________________
Name of Supervisor	:	MR. NIK MOHD KAMIL BIN NIK YUSOFF
Position	:	SENIOR LECTURER, FACULTY OF ELECTRICAL & ELECTRONICS ENGINEERING, UNIVERSITI MALAYSIA PAHANG
Date	:	9TH JULY 2012

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotation and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature	:	__________________________________
Name of Student	:	RAJA SARAVANA KUMAR A/L SELVAKUMAR
ID Number	:	MEL 09002
Date	:	9TH JULY 2012

DEDICATION

This thesis is dedicated to:

My family for their patience and support during my study,
My friends for brightening my life with their friendship and showing me that life have no greater reward to offer than a true friend.

ACKNOWLEDGEMENTS

I believe that I am truly privileged to participate in this fascinating and challenging project as a research member since 2009. I would like to give my deepest gratitude to my supervisor Dr. Kamarul Hawari, who has guided and encouraged my work with such passion and sincerity for knowledge, teaching and care. I would like also to thank Mr. Nik Mohd Kamil bin Nik Yusoff for his guidance, insight and vision on this project.

I am pleased to acknowledge the financial support of Research Management Centre (RMC), Universiti Malaysia Pahang under Fundamental Grant Research Scheme (FGRS) vote 090334 and vote 090302, and from Ministry of Higher Education Malaysia under Pembiayaan Pengajian Pasca Ijazah (PPPI) for scholarship awarded.

I would like to thank all the technicians in Faculty of Electrical & Electronics Engineering laboratories of Universiti Malaysia Pahang for all the help given during the development of this research project for encouraging and supporting my research efforts in microcontroller programming, software design and hardware integrations.

My special thanks go to my family. I would like to thank my parents, who taught me to take chances for better things in my life and also my gratitude to my grandmother, who has offered me unconditional love and care. I thank my sister for their care and encouragement during my study and I would like to wish them all the best in their studies and careers.

In retrospect, this project did start humble but has grown to be a success as now. There are many happy times and many disappointing moments, but now I am very happy because all the hardship I had to go through mostly alone finally paid off.

ABSTRACT

Embedded systems have an everyday presence and direct impact in every day's lives. Therefore, learning institutions are continuously improving their courses in microcontroller and embedded system programming. Although the diversity of curriculums, the availability of learning tools, where the student can practice and improve their skills, is a key factor to the success of the learning process. The platform developed and presented in this thesis results from author's experience in teaching and learning embedded systems. From the analysis of teaching and learning needs, a learning environment based on the Freescale HC11, MCS51, and Microchip PIC 18 Series families was designed, which led to the Multiple Microcontrollers Evaluation Tool (MicroEVAT). This tool can be expanded by modules and adjusted at a specific time to student's real needs. All modules can be interconnected by an IDE bus, allowing expanding the capabilities of the platform. Series of test-run are conducted to verify MicroEVAT system performance. The developed modules allow the practice of subjects related with digital Input and Output interface, analogue interface, user interface, wireless communications, and energy management and conservation. The hardware, software and system architecture used to develop the MicroEVAT are described in detailed in this thesis. In this respect, the MicroEVAT were applicable for education and expose the electrical engineering students to the understanding of microcontroller in electronic design field and embedded systems. The work done for this research project gives solid bases and chances for fast evolution in embedded control technology research.

ABSTRAK

Sistem kawalan terbenam mempunyai peranan khas dan kesan langsung didalam kehidupan seharian. Oleh itu, institusi pengajian sentiasa memperbaiki kursus-kursus yang berkaitan dengan mikropengawal dan pengaturcaraan sistem terbenam dari masa ke semasa. Dengan kepelbagaian kurikulum dan alat pembelajaran, dimana pelajar boleh berlatih dan meningkatkan kemahiran mereka, merupakan faktor utama kepada kejayaan proses pembelajaran mereka. Platform yang dibangunkan dan dibentangkan di dalam tesis ini adalah hasil daripada pengalaman penulis dalam pengajaran dan pembelajaran sistem kawalan terbenam. Daripada analisis keperluan pengajaran dan pembelajaran, persekitaran pembelajaran yang berdasarkan mikropengawal Freescale HC11, MCS51, dan Mikrochip siri keluarga PIC 18 telah direka (MicroEVAT). Alat ini dapat dikembangkan dengan modul dan diselaraskan pada masa ke semasa mengikut keperluan pelajar. Semua modul boleh disaling dengan bas IDE untuk mempertingkatkan keupayaan platform ini. Beberapa siri ujian telah dijalankan untuk mengesahkan prestasi sistem MicroEVAT. Modul-modul yang dibangunkan, membenarkan dalam penggunaan dalam subjek yang berkaitan dengan digit, analog, komunikasi tanpa wayar, pengurusan dan pemuliharaan tenaga. Seni bina perkakasan, perisian dan sistem yang digunakan untuk membangunkan MicroEVAT telah dinyatakan dengan terperincinya didalam tesis ini. Dalam hal ini, MicroEVAT dapat diamalkan melalui pendidikan dan mendedahkan pelajar kejuruteraan elektrik kepada pemahaman mikropengawal dalam bidang reka bentuk elektronik dan sistem terbenam. Kerja yang dilakukan untuk projek penyelidikan ini telah memberikan asas yang kukuh dan peluang untuk evolusi yang cepat didalam penyelidikan teknologi kawalan terbenam.

TABLE OF CONTENTS

CHAPTER 4 SOFTWARE DESIGN

CHAPTER 5 RESULTS AND TESTING

CHAPTER 6 CONCLUSION

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

R_X	Resistor
C_X	Capacitor
S_X	Switch
SW_X	Push Button, Selector Switch
D_X	Diode
J_X	Connector Jack
Y_X	Crystal
U_X	Integrated Circuit, Chip
P_X	Data Connecter
Q_X	Transistor
JP_X	Jumper Connector
MG_X	Motor
a	Angle per step
s	Number of steps per rotation
θ	Prescaler Phase
$Fosc$	Oscillator Frequency, Crystal Value
β	Desired Baud rate
C	Clock Mode

LIST OF ABBREVIATIONS

MicroEVAT	Multi-microcontroller Evaluation Tool
CPU	Central Processing Unit
CCU	Central Controlling Unit
MCS	Microcontroller systems
I/O	Input and Output
ALU	Arithmetic Logic Unit
LCD	Liquid Crystal Display
GLCD	Graphic LCD
DC	Direct Current
DIP	Dual-In-Line Package
ROM	Read Only Memory
RAM	Random Access Memory
IR	Instruction Register
SP	Stack Pointer
EEPROM	Electrically Erasable Programmable Read Only Memory
RxD	Receiving Signal
TxD	Transmitting Signal
GND	Ground

BCD	Binary Coded Decimal
DA	Data Available
CE	Chip Enable
ADC	Analog to Digital Converter
EPROM	Erasable Programmable Read Only Memory
ASCII	American Standard Code for Information Interchange
CR	Carriage Return
LF	Linefeed
GUI	Graphic User Interface
IDE	Integrated Development Environment
VS	Visual Studio
LED	Light Emitted Diode
FMEA	Failure-Mode Effect Analysis
WP11	Windows Programming for HC11
SREC	Motorola S-Record Binary File
I8HEX	Intel 8-bit Binary File
PC	Program Counter
USB	Universal Serial Bus
X-CTU	XBEE Configuration Tool Unit
UART	Universal Asynchronous Receiver Transmitter
USART	Universal Synchronous Asynchronous Receiver Transmitter

CHAPTER 1

INTRODUCTION

This chapter will briefly discuss on the research project overview. The background of the research, research problem description, research objective, scope of the research project, research design and implementation, project contribution, and outline of the thesis will be presented in this chapter.

1.1 BACKGROUND OF THE RESEARCH

The pervasive nature of microprocessors and microcontrollers in nearly every industry segment underscores the necessity for electrical and computer engineering, and science curriculums to include a fundamental of microcontroller course which covers assembly language programming and peripheral interfacing (Cady, 2008). Ni and Luo (2010) stated that hands-on laboratory experience is the key to student knowledge retention and the ability to apply that knowledge in practical endeavours. In addition, it is believed that debugging software and hardware in the lab environment enhances student understanding beyond the basic knowledge of computer operation.

Inadvertently, a course in embedded microcontroller technology is becoming one of the compulsory courses in most electrical and electronic engineering curriculum (Seong, 2007). This course is taught in either the third or the fourth year of a four-year engineering curriculum. It is the syllabus usually incorporated a substantial amount of hardware and software synthesis which requires students taking the course to develop and design an application based on embedded microcontroller system. In the process of helping students with their design projects, various development boards for various platforms are being developed and made commercially available (Stoltz et al., 2005).

These development boards usually come with their own firmware with aim to facilitate learning and usability of the board.

Research and education activities that involving embedded systems design and development requires different microcontroller types, each with its own development board, but structure of such boards is very similar, which only the microcontroller and its associated circuitry being different (Ling, 2008). However, based on the online survey that has been done by the same author indicates that the users have difficulties not only in developing an application system due to limited expendability but also in understanding the hardware structure and configuration of the microcontroller-based application to be developed. Most of the development boards come with limited expandability of the system that prevents the user to develop and to integrate their own application design systems (Ferreira et al., 2005).

A significant challenge to the academic community in embedded system research group is the obsolescence rate of computer laboratory hardware. Researchers began developing a new microcontroller hardware evaluation board with the following goals:

i. To specify a hardware platform based on the newest available technology with a clear roadmap for future product support. The evaluation board shall be self-contained and provide sufficient experimental flexibility,

ii. To design a hardware platform robust enough to withstand student handling and simple enough to operate without excessively compromising experimental flexibility, and

iii. To develop an introductory microcontroller laboratory course requiring minimal infrastructure changes. The laboratory must be self-contained to the extent that lecture content and flow is relatively independent of the laboratory experiments, and general enough to adapt too many different microcontroller texts and lecture topical organizations.

Therefore, a flexible and versatile prototype system of MicroEVAT has been developed. This system can be used extensively in experiment or project for diploma, undergraduate and short courses. This MicroEVAT has also been boosted with a simple application board that is suitable for the students to test their capabilities and to improve their knowledge in this course. In addition, a monitor program has been developed to integrate the basic software such as communication software, text editor, cross assembler, and compiler. Needless to say, it will tremendously create a user friendly environment.

1.2 RESEARCH PROBLEM DESCRIPTION

The students have experience the difficulty in understanding the architecture of the microcontroller system and its application in embedded controller courses. The development board that available seem do not fit to the requirement of the courses due to the limited access of microcontroller and its applications. Most of the embedded controller course has introduced the MCS51, Freescale HC11 or Microchip PIC microcontroller as the core controller. Some will use these microcontrollers for the class assessments and projects, hence will lead to use of development board to aid the student familiarising the microcontroller architectures. The usage of different microcontroller for different course will require different development board. Therefore, the cost will be rise as the teaching material for each course are different. Some development boards might not include user-friendly software for the students.

1.3 RESEARCH OBJECTIVE

The objective of this research study is to design and develop the Freescale HC11, MCS51, and Microchip PIC 18 series all together in one design system and application board with user-friendly environment. The aim is to allow the system can be used for experiment or project for diploma and undergraduate student. The developed design system must be multi-purpose, cost-effective, easy to operate and offer various input and output devices including some example of wireless devices. For this reason, this project is designed with Graphical-Oriented monitor program to provide a user-friendly environment with the MicroEVAT system.

1.4 SCOPE OF THE PROJECT

The scopes set forth for the research work as follows:

i. Designing a hardware design system using three (3) types of microcontroller families; Freescale MC68HC11, Intel MCS51, and Microchip PIC 18,

ii. Establishing an extensive applications board with various form of input and output devices,

iii. Providing a communication between the hardware and computer,

iv. Developing a program using assembly language as based of interface and execute the program in the system,

v. Providing samples program to execute each module of applications, and

vi. The system is designed based on open module for future expansion.

1.5 RESEARCH METHODOLOGY

In order to design a multiple microcontroller hardware platform, the project will be divided into two parts; hardware design and software design. The hardware consists of two major modules; system module and application module. The system module provides the microcontroller, storage units and interfacing units that can be easily interfaced with various I/O devices. The system is developed based on modular design where user can configure the system to be interfaced with any application modules or any industrial systems. Likewise, the application will be designed based on the most current devices.

In similar manner, the design of software will be divided into two categories; the monitor program and the Integrated Development Environment (IDE) parts. The monitor program is developed by using the assembly language to allow communication between the PC and the system module. From the PC, the user can communicate with the system board to issue commands to upload or download, execute and read or write to memory location. The IDE, on the other hand, integrates various basic software such as cross assembler, communication software and text editor to create user-friendly environment. This new software tool is developed to allow the user to perform all development activities without needing to exit any program.

Both the hardware and software will be integrated into the system and various testing will be conducted to test its interfacing capabilities with external I/O devices. A thorough free run test on the system will also be performed to ensure system reliability. The implementation of the project research is shown in Figure 1.1

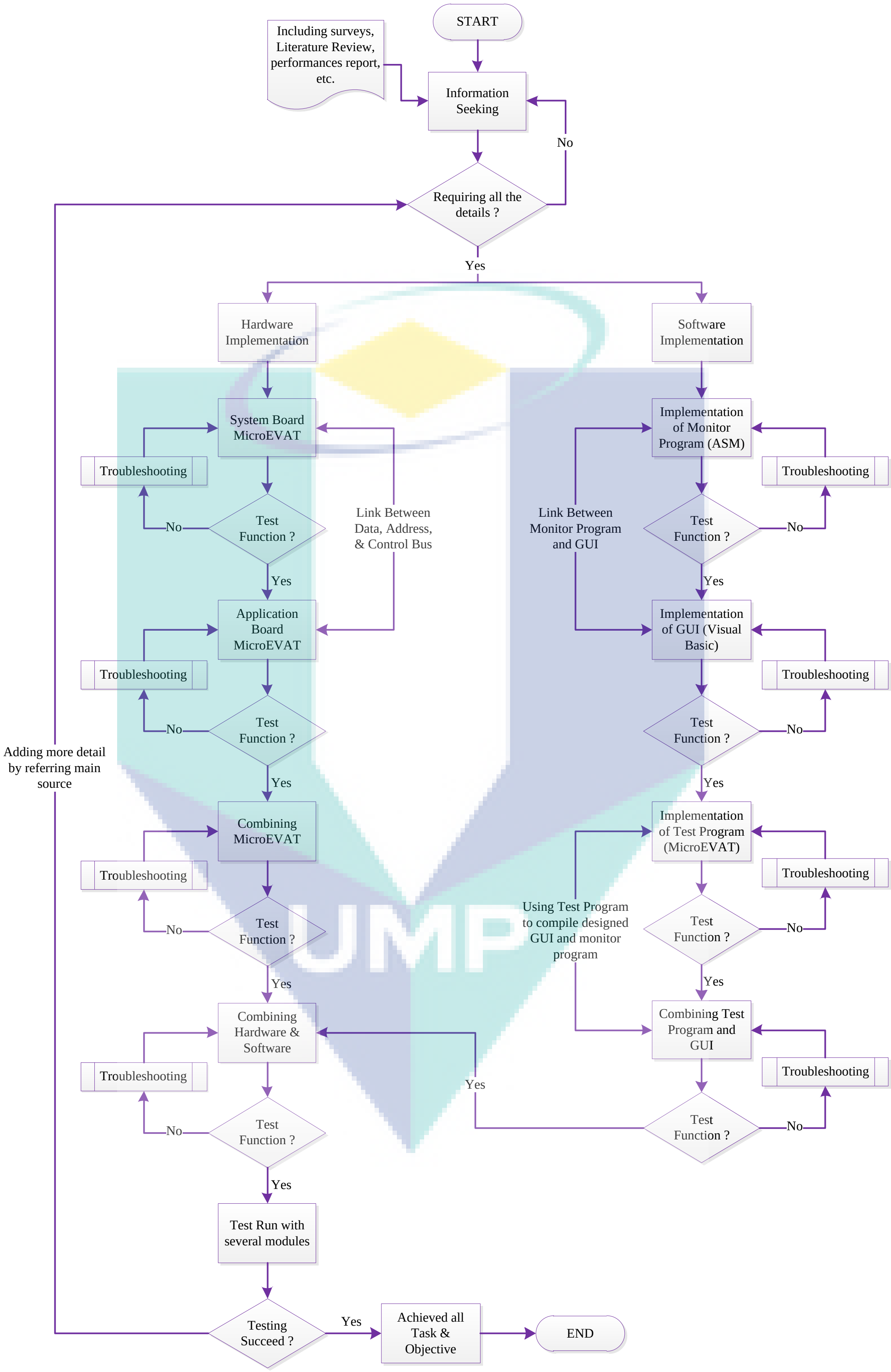


Figure 1.1: The research project implementation flow chart

1.6 PROJECT CONTRIBUTIONS

The project contributions are as follows:

i. A prototype of evaluation tool that offers three types of microcontroller families which are consists of Freescale HC11, MCS51, and Microchip PIC 18 has been developed,

ii. Developing an extended memory for more programming space using external RAM and ROM by using monitor program,

iii. The monitor program that has been created able to configure configuration registers in a microcontroller without using any other external software,

iv. Multiple application module without limitation of the port register of the microcontroller,

v. Designed using diode 1N4001 and serial driver DS275 for single communication port for all the three types of microcontroller using RS232, USB or ZigBEE protocol,

vi. Mode Selectivity for Freescale HC11, and MCS51 that can be used for accessing the external and internal memory, and

vii. A prototype of MicroEVAT has been developed. This is the major breakthrough in the effort of developing an educational multiple microcontroller hardware platforms.

1.7 THESIS ORGANIZATION

This thesis is organized into five chapters. The first chapter introduces the motivation, research objective, scopes of work and contribution of this project.

Chapter 2 reviews the microcontroller development board history, concept of embedded systems, related works on microcontroller hardware platforms and embedded systems design flow. System course and guidelines of choosing the microcontroller are also explained in this chapter.

Chapter 3 presents the methodology for hardware design for MicroEVAT system. The explanation is given separately according to the functionality of the system and is elaborated in a more technical way and specific terms.

The software design of the MicroEVAT system is discussed in Chapter 4. Detail explanation will be given by using subroutine function to optimize the algorithms.

Chapter 5 presents the control design methodology and result for each system and application module for the MicroEVAT system. The test-run of the system has been conducted in order to test the functionality of the MicroEVAT system. The preliminary tests are also conducted to ensure that the system developed functions properly.

In the final chapter, Chapter 6, the research work is summarized and the potential future works are outlined. The contents include the experience and the knowledge gained during accomplishing this project. Furthermore, several recommendations are also been suggested.

CHAPTER 2

LITERATURE REVIEW

This chapter reviews the previous works that are related to this research study, and presents the components that have been used in this project.

2.1 INTRODUCTION

In general, people consider the microcontroller as unseen workhorse embedded in automobiles, microwave ovens, guided missiles, talking dolls, and a myriad of other applications. Unlike microprocessors, which are designed for a broad range of applications, microcontrollers are generally designed with a specific application in mind. For the designer of an embedded system the choice of a microcontroller is the earliest and probably the most important to be made. Delivering the final product on schedule, within budget and specification will depend heavily on making the decision.

With components being ever more complex, a design methodology and the ability to easily test and verify the designed system on a real object are becoming increasingly important. Building of application prototypes, either hardware or software therefore plays an important role in the design process. Since microcontroller based application design can be described as a combination of computer programming and digital circuits constructing as published by a number of researcher such as Predko (2000), Wolf and Madsen (2000), and Al-Dhaher (2001), it is important to excel both fields at the same time. With the technology advancing very rapidly, the progress in this field also has to reflect in the pedagogical process (Caspi et al., 2005). Educational institutions, responsible to provide students with appropriate skills and knowledge, thus need to be aware of the industry needs and practices. On the other hand, according to

Jackson and Caspi (2005), industry is the one responsible to make its specifics and requirements well known.

Recognizing and acting upon the paradigm shifts in the design of contemporary systems thus calls for novel pedagogical approaches that should be supported by suitable hardware and software tools (Al-Dhaher, 2001). For that reason, researchers from Universiti Malaysia Pahang have designed a framework aiming to ease and quicken this process. It is hardware part in the MicroEVAT system with modular approach structure proposed by Smolnikar and Mohorcic (2008). This means that it is recomposed for each application separately. The main hardware parts are the main board with the attached microcontroller and the memory used to download the program code from a personal computer (PC) to a microcontroller.

In addition to these two major hardware parts, several typical plug-in test modules have been designed, intended for the verification of developed programs. The designed software tools can be divided into two; program code design, and programming the microcontrollers. In the latter group only freely available programs are considered, while in the case of development tools the author focusing on those offering high-level design, with the algorithms described in the form of different graphical representation (e.g. block diagrams, flowcharts) rather than written in C or assembly code.

The author postulates that proficient fundamental knowledge of any instruction set architecture and embedded microcontroller peripheral interface enables students to quickly adapt to any processor or platform. Assuming skill portability, one may argue that using dated hardware such as the Freescale HC11, in no way detracts from the educational efficiency of a microcontroller laboratory (Stoltz et al., 2005). In practice there are a few limitations to old hardware:

i. Hardware degrades and replacements become difficult to procure,

ii. Textbooks are removed from print as texts supporting new processors become available,

iii. Development tools become obsolete (e.g., MS-DOS based applications),

iv. Antiquated hardware fails to take advantage of higher processor speeds, larger memory spaces, and newer peripherals such as CAN networking, and

v. Latest personal computers do not have serial communication such as RS232 protocol.

Many learning institution around the world of engineering are still using an obsolete microcontroller platform such as the Freescale HC11, MCS51, and Microchip PIC in their laboratory. A few schools have started to upgrade their microprocessor or microcontroller courses to more recent platforms. For example, Iowa State University has faced a similar hardware upgrade challenge and chose the PowerPC based Freescale MPC555 processor for a new laboratory environment (Striegel and Rover, 2002).

Given the plethora of embedded processor options on the market today including Freescale HC11, MCS51, and Microchip PIC derivatives, selecting a processor is a key decision. Iowa State University chose a high-end processor with a C language development environment for a new lab platform. A trade-off exists between teaching processor architecture and programming fundamentals and using more powerful processors and tools. The Universiti Malaysia Pahang has selected these three microcontroller families with basic tools and an exclusively assembly code based introductory lecture and laboratories assessment (Sapein, 2009).

The MicroEVAT is very cost effective to implement in other student projects with the minor modification on system board. The MicroEVAT is unique when compared with other commercially available products. This system has been specifically designed as a complete, robust, stand-alone development system, for teaching an introduction to microprocessors and microcontrollers undergraduate course. Many currently available development tools include switch or LED I/O, but few include a mix of peripheral devices specifically chosen to support a 14-week university course based on Universiti Malaysia Pahang’s student performances report by Nik Yusoff

(2006). The MicroEVAT includes three types of microcontroller families on one evaluation system, allowing pod mode debugging without separate hardware. For designated particular program and constituencies, this choice has proven very effective. In summary, the characteristics of MicroEVAT with comparison of the conventional system that commercially available are listed as follows in Table 2.1.

Table 2.1: Comparison of MicroEVAT with Conventional System

Conventional System	Proposed System (MicroEVAT)
The existing training system only offers one type or family of microcontroller. [1] [2] [4] [5] [6]	The new innovation offers three type of microcontroller families, Freescale HC11, MCS51, and Microchip PIC.
Limit space of programming memory since only using internal memory. [1] [2] [3] [4]	User can choose either to use internal or external memory – limitless programming space.
Using 3rd party software to configure the system, come with bundle package or buy it separately. [1] [2] [3]	Monitor program are pre-installed in memory (ROM) for easy access of the system.
Only RS232 communication available on some EVBs. [3] [4] [5] [6]	RS232, USB and Wireless ZIGBEE protocol.
Some EVBs could not operate many application devices at one time. [1] [2] [4]	Overcome by using Multiplexing Mode and Address Decoding to provide various application devices operation at one time.

[1] MC Pros STK200 Atmel AVR Starter Kit
[2] Kanda PIC Training Kit
[3] BiPOM MicroTRAK
[4] Mini40 PIC Training Kit
[5] Vinytics 8051 Microcontroller Training Kit
[6] Acumen Microcontroller & Microprocessor Kits

2.2 CRITERIA OF CHOOSING A MICROCONTROLLER

For the designer of an embedded system, the first task is the choice of a microcontroller is among the earliest and probably the most important to be made. One of the most important criteria is the requirement from the FKEE faculty syllabus where the use of proposed microcontrollers in academic environment. According to Figure 2.1 that proposed by Vaglica and Gilmour (1990), this set of guidelines is for the neophyte users to use as a checklist for evaluating candidate devices. Even experienced users may discover in it selection criteria they have overlooked. A microcontroller is generally dedicated to a single application and embedded in it. Unlike a microprocessor, it tends to include all of the peripheral features needed to implement the computer portion of an embedded application (Ma et al., 2010). For this reason, the designing of MicroEVAT is based on this selection method.

2.2.1 Determine the Performances of Microcontroller

The most important criteria are the level of performance required, that is the size of the controller. A group of researcher leaded by Alves and Martinas Ferreira (1999) collects several data of available microcontrollers and can be grouped broadly as 4, 8, 16 and 32-bit products. Standard benchmarks of the type used to describe microprocessors are inappropriate for this evaluation. They must be replaced by critical real-time calculations and physical constraints peculiar to the system being designed. Critical calculations are those that, if not completed within a specified period of time, causing the system to lose synchronization, create a dangerous situation, or generate invalid results.

Physical constraints imposed by the system, such as precision and dynamic range, determine CPU class. For instance, a numerically controlled machine might require a relative accuracy of 1 degree within a single revolution of an axis. For this accuracy, an 8-bit data value, which can resolve no more than 1 part in 256, is inadequate, whereas a 16-bit data value will represent 1 part in 65 536, or about 20 seconds of arc (Vaglica and Gilmour, 1990).

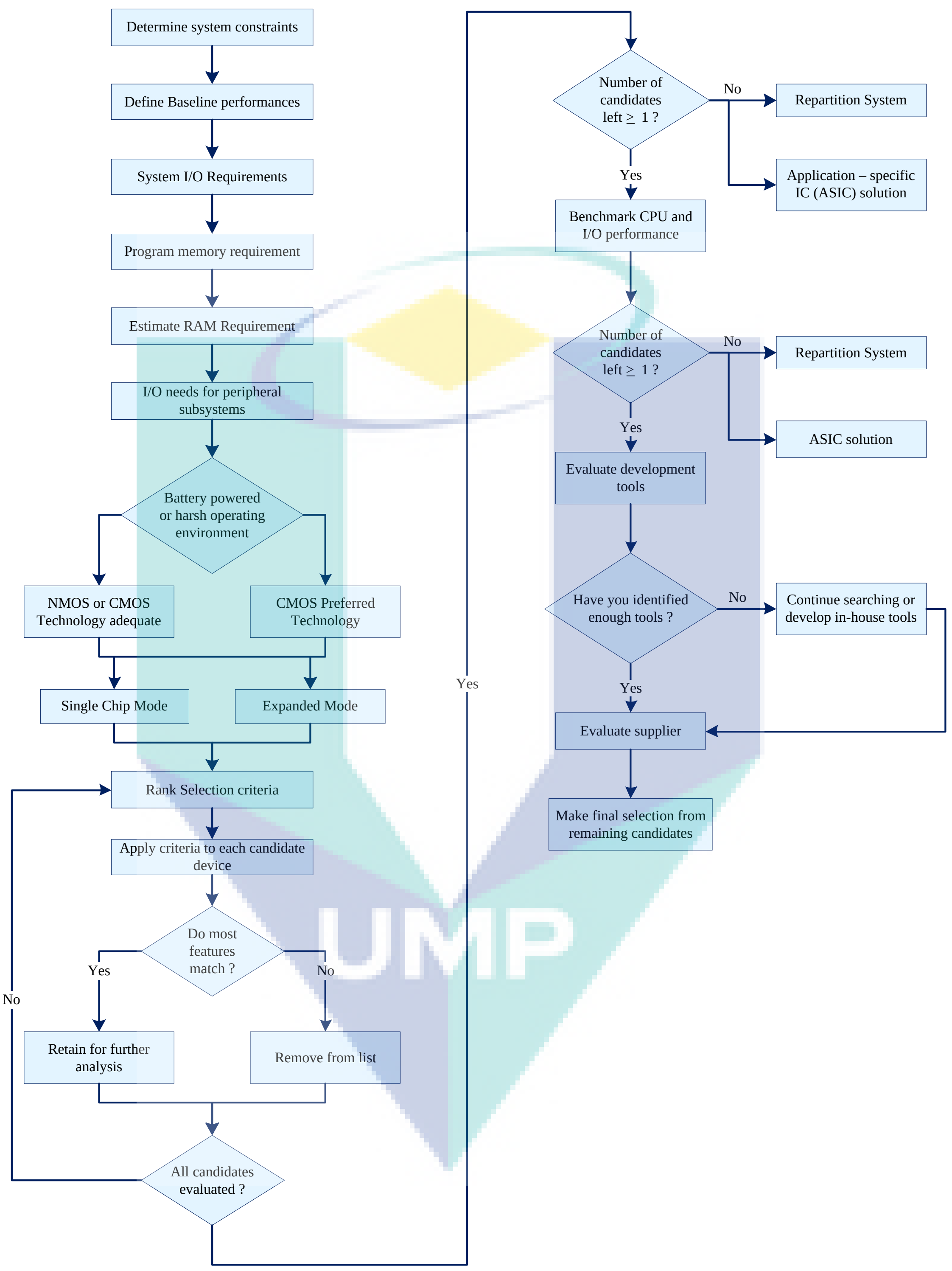


Figure 2.1: Guideline of choosing the suitable microcontroller

Source: Vaglica and Gilmour, 1990

Dynamic range further constrains the system. If the same machine must also be able to provide an absolute precision of l degree over 1000 revolutions of a lead screw, even 16 bits cannot represent the full range. In such a case, the designer could choose a 32-bit solution or, if extremely wide dynamic range is necessary, resort to a floating-point number system. The designer is cautioned, however, that choosing a CPU on the basis of a single calculation could be a mistake. If the application generally requires lower-resolution calculations, it may be better to select a processor that matches just the predominant operations, while performing the higher-resolution calculations in subroutines (Peatman, 1998). In most applications, for example, a good floating-point math package is superior to floating- point hardware.

The second step-the most often overlooked-is to identify the quantity, frequency, and type (analog, digital, serial, and so on) of all input/output (I/O) signals, as well as any other special requirements, including those imposed by mechanical aspects of the system. According to Lawrence and Mauch (1987), key to this part of the selection process is a complete top-level block diagram of the overall system. Although it is not important to assign microcontroller peripheral functions at this stage, a preliminary partitioning might be helpful in identifying system requirements.

The third consideration is the application's memory requirements, which should be further broken down into program memory and data memory. Microcontroller memory may be categorized broadly as volatile and non-volatile. Volatile random-access memory (RAM) retains data only while power is applied; it is generally used for data memory. Non-volatile read-only memory (ROM) retains its contents permanently; it typically contains the application program of the embedded system (Peatman, 1998).

Non-volatile memory comes in four variants: electrically programmable ROM (EPROM); electrically erasable/programmable ROM (EEPROM); one-time-programmable EPROM; and mask-programmed ROM, which is programmed permanently at the time of manufacture. EPROM and EEPROM devices, being erasable, are intended primary for development purposes. They are also the most expensive variants because of package cost, process complexity, and die size. An EPROM is erased using strong ultraviolet light and therefore requires a package with a quartz window. One-time-programmable memories are standard EPROM parts packaged in low-cost plastic packages without quartz windows. Because they can be programmed by the user, they eliminate the cycle time required to produce mask-programmed ROMs. According to Valvano and Jonathan (2003), an emerging memory technology is flash EEPROM, which offers denser storage than conventional EEPROM. EEPROM is a microcontroller feature that has yet to reach its full potential (Stoltz, 2003). Many designers think of it merely in terms of program storage, not as a strategic peripheral.

Sufficient RAM must be allocated in the system to store variables, the system stack frame, a scratchpad for intermediate calculations, and any data arrays or buffers. These values will be affected by the programming language used to develop the application (system stack), the selected partitioning of the problem (variables), and even the hardware configuration (data buffers). Preliminary estimates for variable stack and scratchpad storage are often done by taking a percentage of the ROM estimate. The rule of thumb used by the authors Vaglica and Gilmour (1990), is a ROM to RAM ratio in the range of 12-20 to 1. Applications written in assembly language will tend toward the lower number, while compiled code will require the higher amounts. Large data buffers should be added to this estimation.

2.2.2 Mapping I/O to Peripherals in Microcontroller

Peripheral functions commonly integrated on microcontrollers include timers, serial and parallel communications ports, and analog-to-digital (A/D) converters. Timers range from simple counters to complex subsystems with dedicated micro engines incorporating reduced-instruction-set computer (RISC) architecture. Timer systems are commonly called upon to generate periodic interrupts, capture the time an input event occurs, or generate output events at specified times. The more complex timers can produce the pulse trains required of multiphase stepper motors or even of sequencing the fuel injectors in an automobile engine without CPU intervention (Smith and Nair, 2005). The frequencies of the input and output signals generally dictate how timer complexity should be traded off against the CPU overhead required for servicing the peripheral.

External peripherals added to provide functions not usually found on microcontrollers may be interfaced through either a parallel (address, data and control) bus or a serial port. The parallel bus is conceptually simple, but fraught with many practical problems because of its rapid switching of many data lines. Based on researches that have been done by Lawrence and Mauch (1987), it generates a lot of radio frequency interference (RFI), and it consumes a lot of power. For applications that would be handled by a single chip except for a single special function, it is suitable to add the peripheral via a serial bus. Only two or three pins are required, RFI is generated only during serial transmissions, and power consumption rises only slightly. A variety of chips that perform peripheral functions are marketed with serial interfaces. Among them are A/D converters, phase-locked loop (PLL) building blocks, real-time clocks, display drivers, and EEPROMs.

Parallel I/O can be found on virtually all microcontrollers, but the ports are not all equivalent. The more versatile permit pins to be defined as input or output on a per-bit basis. This is significantly important if system parameters are subject to change before the design is complete (as is usually the case). Otherwise, less flexible designs offer fixed direction, input-only, or output-only pins. A/D converters are found on many microcontrollers. A converter with 8 or 10-bit ranges are most common, but not all manufacturers offer the same resolution for a given range. The specsmanship issues has been addressed by Hennessy and Patterson (1996), which apply to stand-alone A/D converters, are an even bigger problem with converter ICs. If the intended purpose is much more demanding than checking battery voltage, care must be taken to examine all the specifications of this subsystem.

An application that seems to require a unique peripheral may be designed using available peripherals in innovative ways to minimise cost. Digital-to-analog (D/A) converters, for example, are seldom integrated into microcontrollers. If it is required in the design, the resourceful engineer will provide one economically by integrating a pulse-width modulated waveform generated by an on-chip timer. The integrator can be as simple as a passive RC low-pass filter (Hennessy and Patterson, 1996). Peripherals requiring frequent interrupt service consume valuable CPU bandwidth. Several methods employed by microcontroller designers (Prasad et al., 2003) reduce interrupt overhead.

Unique vectors for each interrupt source, multiple priority levels, and hardware priority resolution circuits eliminate software polling of interrupt sources. Interrupt-driven direct-memory-access (DMA) peripherals transfer data with minimal CPU service overhead, although they still require considerable bus bandwidth. Other designs get rid of service requirements altogether by distributing enough intelligence to the peripheral to eliminate CPU servicing completely during normal operation.

Hadzic (2004) from University of Oslo stated that the bus bandwidth is affected by two components: bus width and transfer rate. Boosting either causes a corresponding increase in bandwidth. Unfortunately, this bandwidth increase normally implies a cost increase as well. Microcontroller architects have designed features into high-end products that minimize the effect of greater bandwidth on system cost. Seldom will all memory and peripherals in a system need to be accessed at the highest possible rate. Microcontrollers with dynamically sized buses permit memories of different widths to coexist on the same bus (Cady, 1997). Resources requiring high bandwidth, such as the stack RAM, can occupy the full bus width. Locations accessed less often, such as the boot ROM, can be configured for the width of a single memory part. Reducing the number of devices by means of dynamic sizing minimizes use of printed-circuit board area, increases reliability, and decreases RFI.

Cost, packaging, operating environment, and other physical conditions further constrain the designer. These constraints must be spelled out clearly early on, for they bear directly on the microcontroller selected. Power consumption and temperature range dictate the processing technologies suitable for the application. If the product is battery powered or will operate over an extended temperature range, a CMOS version would be the better choice.

2.2.3 Mode of Operation in Microcontroller

Microcontrollers often support multiple modes of operation for a better match to the application at hand. The two most common modes are single-chip and expanded. The address, data, and control buses required for memory or peripheral expansion are not brought out to pins. In the expanded mode, these buses are made available.

Microcontrollers capable of both single-chip and expanded modes fit well in applications where future upgrades or cost reductions are likely and a consequent complete rewrite of the application software undesirable (Kaeli and Platcow, 2006).

Choosing between single and multi-chip solutions depends heavily on memory size. Technology currently limits on-chip memories to 32K bytes for program (non-volatile) storage and 16K bytes for data RAM (Microchip, 2005). Otherwise, an expanded system using external memory ICs will be needed. The distribution of on-chip versus external memory can affect total IC cost significantly. Since memories are only available in a limited range of sizes and on-chip memories are more expensive (on a per-bit basis) than external memory, it may be advantageous to move all program and/or data memory off chip in exchange for a less expensive microcontroller chip.

In addition to memory size, other obvious factors that affect the single-chip vs. multi-chip decision are power consumption, peripheral mix, and cost. But there are also less obvious factors. For instance, microcontroller-generated RFI is a major concern in RF communication applications. High-speed digital outputs contribute significant energy to the radio spectrum. Confining these signals on chip, where capacitive loads and signal line lengths are reduced up to 100 times, significantly reduces emission levels. The researcher Vaglica and Gilmour (1990), realizing the problems that high-frequency signals cause, some microcontroller designers have gone so far as to provide software-programmable disables on potentially unused high-speed outputs.

2.2.4 The Selection Process of a Microcontroller

During the initial search, an absolute match between requirements and features is not necessary. Minor alterations to the requirements or the addition of a peripheral chip could create the most cost-effective solution from a less-than-perfect pairing.

The feature evaluation process should begin with a ranking of the requirements in order of descending priority, as determined by the application. If any requirements are not mapped to peripheral functions during the partitioning for the block diagram, they should be mapped at this point. Using this mapping will not restrict the selection to a particular device or manufacturer, but will identify the microcontroller features sufficiently for the selection process to progress (Al-Dhaher, 2001). The user is cautioned that devices may surface during the selection process that would be a perfect fit if only the system had been partitioned differently. An open mind will permit these possibilities to emerge.

The procedure at this stage is to compare each candidate microcontroller with the peripheral mapping and other requirements, retaining for further study any devices that match most of the requirements. Several iterations may be necessary at this stage before a suitable chip or chips are found by applying more stringent criteria at each stage. For instance, after the field has been narrowed to a handful of candidates, actual application code sequences will be written and used to benchmark CPU performance. If undertaken before the list is narrowed, this step would drag out the evaluation inordinately.

An established growth path could be an important evaluation point if a family of products is planned. Many vendors offer microcontroller families built around a common CPU core where different family members feature different I/O and memory mixes (Rupp et al., 2003). CPU performance improvements are made through clock frequency increases and upgrades to the core.

The final selection criteria are nontechnical. Potential vendors should be qualified after considering the following criteria: product line breadth, manufacturing excellence, financial status, second-sourcing strategy, and delivery performance. Development tools must also be taken into account. If inadequate, they could place the entire project in jeopardy; therefore, it is advisable to spend some time understanding what is available for the microcontroller chosen. The entry cost of development tools can be high (Ferreira et al., 2005).

Microcontroller selection is an arduous task entailing an inordinate number of decisions. Experience, in fact, is the second most useful tool the designer can possess; the most useful is a well-defined set of system requirements. Without a list, an already difficult task becomes almost impossible.

2.3 MICROCONTROLLER CONCEPT

Due to the fact that microcontrollers constitute a complex technology (Brockman, 1998), the MicroEVAT system has been designed using modular approach method to obtain the descriptive model of a complex technology developed by the Institute for Applied Electronics of the University of Vigo (Valdés et al., 1999). This method is represented in Figure 2.2 and comprises four principal stages:

i. Firstly, many different representative systems or devices are chosen,

ii. In the second stage the selected systems are analysed in detail to define the concepts associated to the technology. This task is carried out in two different phases:

   a. All the common characteristics are determined and classified to define the general characteristics or basic concepts of the complex technology, and

   b. In the second phase the basic concepts are characterized (including functionality, implementation, architecture, etc.) taking into account the specific characteristics of each particular system in such a way that the sub concepts of the descriptive model are obtained as well as its dependence relations. The same sub concept can be present in different systems but the set of sub concepts associated to each system can be different.

iii. In the third stage all the basic concepts and sub concepts are structured to obtain the descriptive model, and

iv. Finally, the descriptive model must be tested to verify its ability to describe not only the systems chosen to obtain the model but, all the commercial systems known.

Developing descriptive models is a tedious task requiring a lot of time and effort. Nevertheless, once the final result is obtained we achieve a very useful tool for the analysis of complex technologies as well as the particular systems included in them. Besides, if new systems are developed, updating the model with the inclusion of new characteristics is very easy. Figure 2.2 shows the complex technologies characterizing methodology.

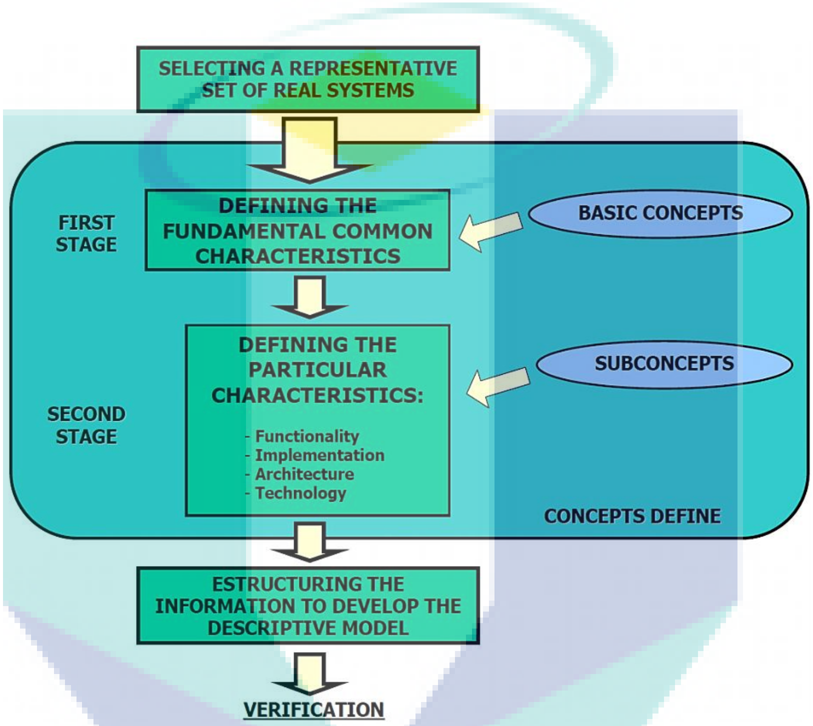


Figure 2.2: Complex technologies characterizing methodology

Source: Ferreira et al. (2005)

Microcontrollers are integrated computers composed of a memory unit, arithmetic and logic unit and a control unit. These three components are combined by means of the most important concepts of the microcontroller technology: internal architecture, external architecture and I/O interfaces (Ontoría et al., 2000). The most relevant characteristics of the MicroEVAT contains peripheral devices for simple digital system implementation and verification, expandable to connect external peripheral elements for complex digital system implementation that gives constitutes a rapid prototyping development system because all the microcontroller pins are available for peripheral devices and finally can be used as an 'In circuit debugger' connecting the microcontroller to a prototype board. The conceptual elements have been used to implement the systems and different elements can also be described by means of conceptual maps. The concepts which are characterize the different types of particular systems. It is a self-learning oriented system due to its switching on the fly capability from programming to running operation mode. Figure 2.3 shows the microcontroller conceptual map.

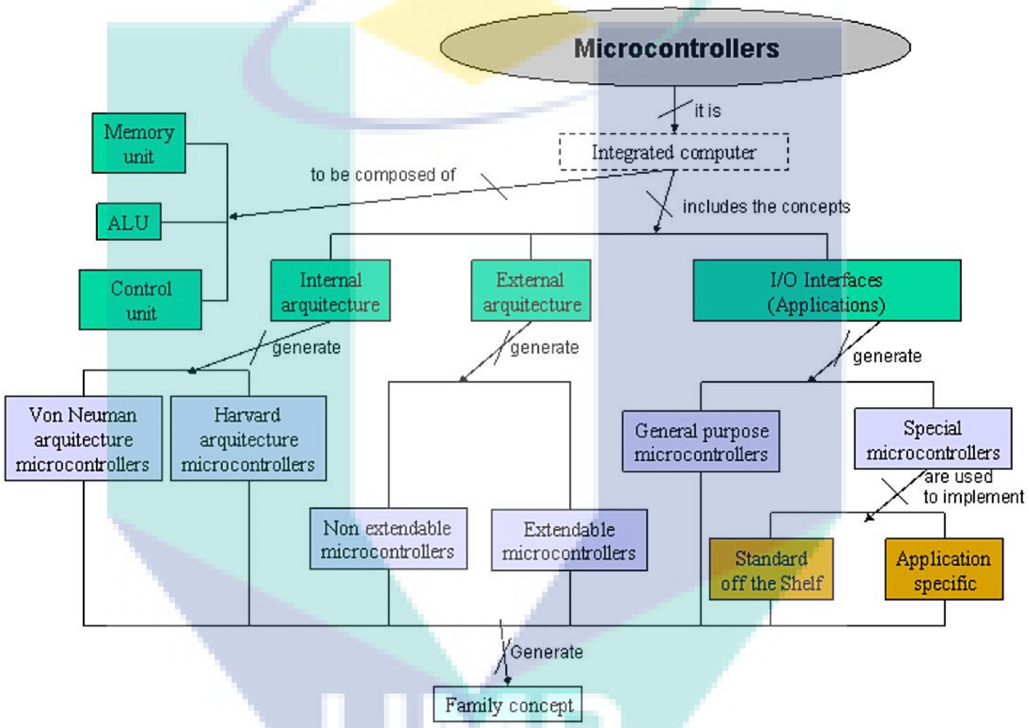


Figure 2.3: Microcontroller conceptual map

Source: Novak Joseph (1990)

2.4 IMPLEMENTATION OF DEVELOPMENT TOOL

Due to dramatically increasing complexity of development tool over the past decades, developers are facing ever-increasing challenges for their products to stay market competitive (Mosterman et al., 2004). In this context the utilization of systematic design methods is essential in order to aggregate rather than trade-off the technical, cost, and time-to-market feasibility factors (Rupp et al., 2003).

2.4.1 V-Diagram of Development Tool Design Flow

Typically, the implementation of development tool can be illustrated using a V-diagram representation. Several versions from different industry fields can be found in the literature, but here the author refer to the one applied to the design of embedded systems (Caspi et al., 2005). As depicted in Figure 2.4, the general progression of the design steps in time is indicated from left to right. Hence the horizontal axis of the diagram can be thought of as time, but since the design is often an iterative process, the actual development rather cycles between left and right leg of the diagram than proceeds linearly through the steps (National Instruments Corporation, 2007). The vertical axis represents the level of system components' abstraction, with the top steps representing high-level system view and the bottom steps representing very low-level processes.

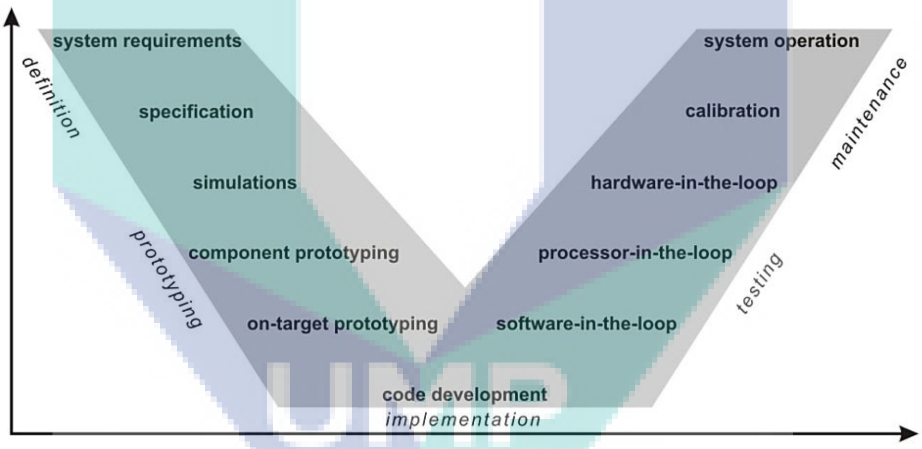


Figure 2.4: V-diagram of development tool design flow

Source: Smolnikar and Mohorcic (2008)

The design starts with specifying the system requirements. These are than processed, and the system is along the left, the decomposition leg of the V-diagram, split into smaller pieces. At the bottom of the V-diagram is the implementation stage, representing the transition from decomposition back to re-composition proceeding up along the right leg of the V-diagram. Here the system's entities are combined back into larger pieces, resulting in a finally assembled system. The goal of each design process is a system working according to the specifications, with the V-diagram as narrow as possible, thereby reducing the time and consequently the cost of design (National Instruments Corporation, 2007).

The implementation of MicroEVAT which is presented in this thesis aids narrowing the development tool design V-diagram by providing a common set of tools that are used through the prototyping implementation and testing stages. In prototyping stage first a simulation model is designed that behaves according to the requirements. Then for the validation of feasibility, the design components are prototyped on general prototyping hardware. The timing characteristics are not modelled in this step, and may change in the actual design. Once the components behave appropriately, the on-target prototyping is employed on the envisaged processing device, in our case a microcontroller.

Characteristics of code programming, fixed-point precision effects and interactions among the system's components are validated in this step (Mosterman et al., 2004). When all simulations and prototyping shows the correct operation of a design, a highly optimized code is produced in the implementation stage. Finally, in the testing stage the design is evaluated while moving up along the right leg of the V-diagram. Configurations such as software-in-the-loop, processor-in-the-loop, and hardware-in-the-loop are used in the proceeding steps.

2.5 SURVEYS

The neediness of this system to the education syllabus in our faculty is determined by providing the respondents with a set of questionnaire on the system features in MicroEVAT. In addition, feedbacks of the current development system that available in our faculty are also being questioned. A survey on student's understanding in embedded systems is conducted to detect the level of understanding in this course and the feedback from the students if there are develop in-house microcontroller educational development system based on the MCS51, HC11, and PIC 18 series microcontroller are developed. Figure 2.5 shows the pie chart of number of students who participant in survey.

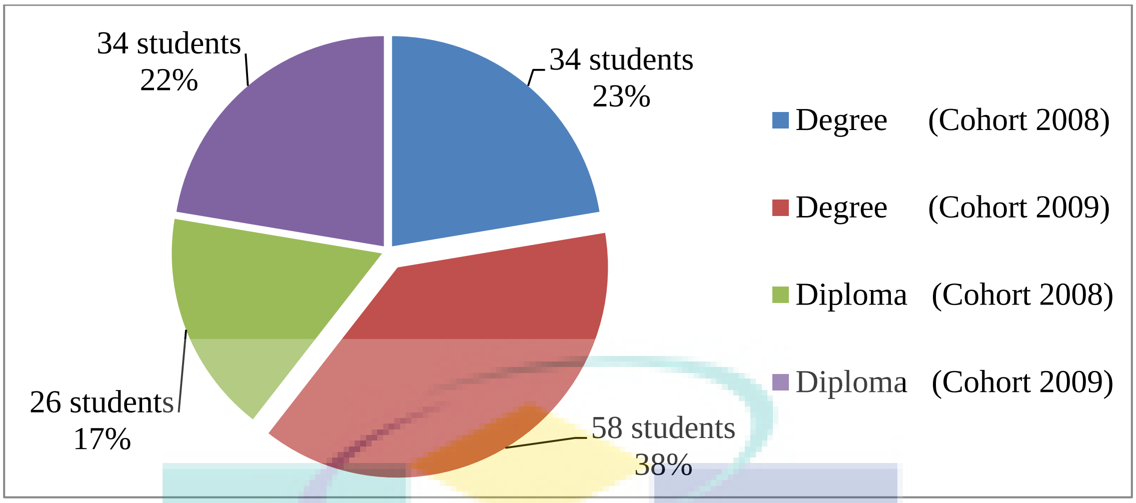


Figure 2.5: Pie chart on number of students by faculty program

A number of 152 responses from the undergraduate students in FKEE who replied the survey and 61% of the respondents are from degree program students. In several, the students learn at least with two types of microcontrollers in each of diploma or degree program. The MicroEVAT system must meet these requirements in order to fulfil the needs of education syllabus that standardise by FKEE. Figure 2.6 shows the bar graph on number of students with type of microcontrollers.

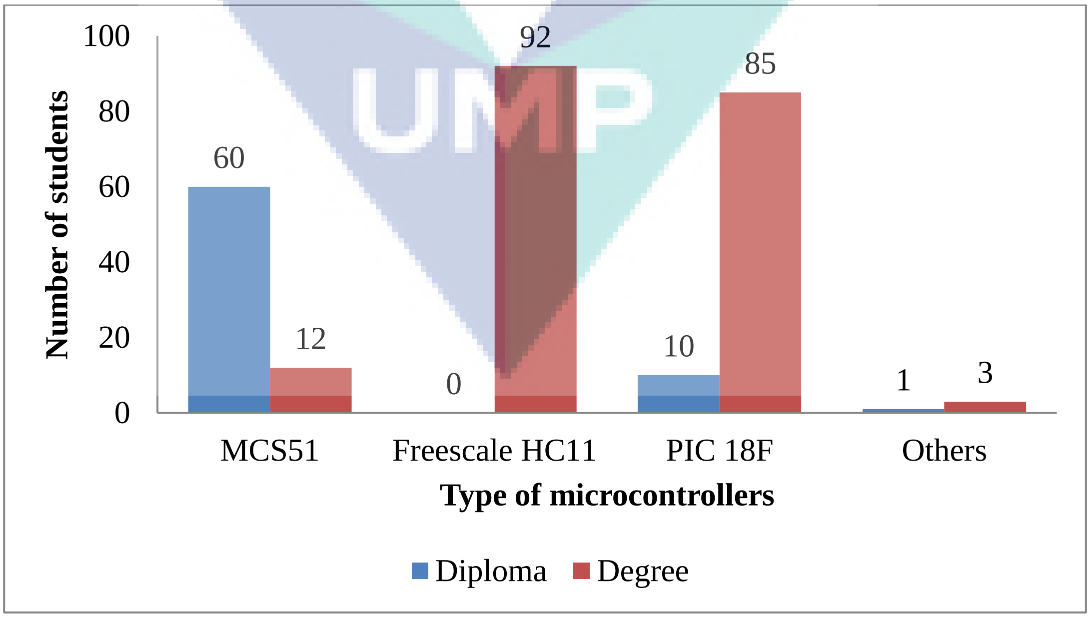


Figure 2.6: Bar graph on number of students with type of microcontrollers

From the Figure 2.6, the majority of the students use the microcontroller that they have learned in diploma or degree program, and the total of 2.6% uses other than the proposed microcontrollers, such as ATMEGA microcontrollers from the final year project. But, mostly of the students use the proposed microcontrollers for their class projects or final year projects. Students from degree program use at least all the proposed three types of microcontrollers in their assessments. From the result above, the MicroEVAT system can be applied in educational syllabus since the majority of students uses the same type of microcontrollers.

The survey also conducts the most preferable programming language on designing the program structure in MicroEVAT system. The result is shown in Figure 2.7 on number of students with the type of programming languages.

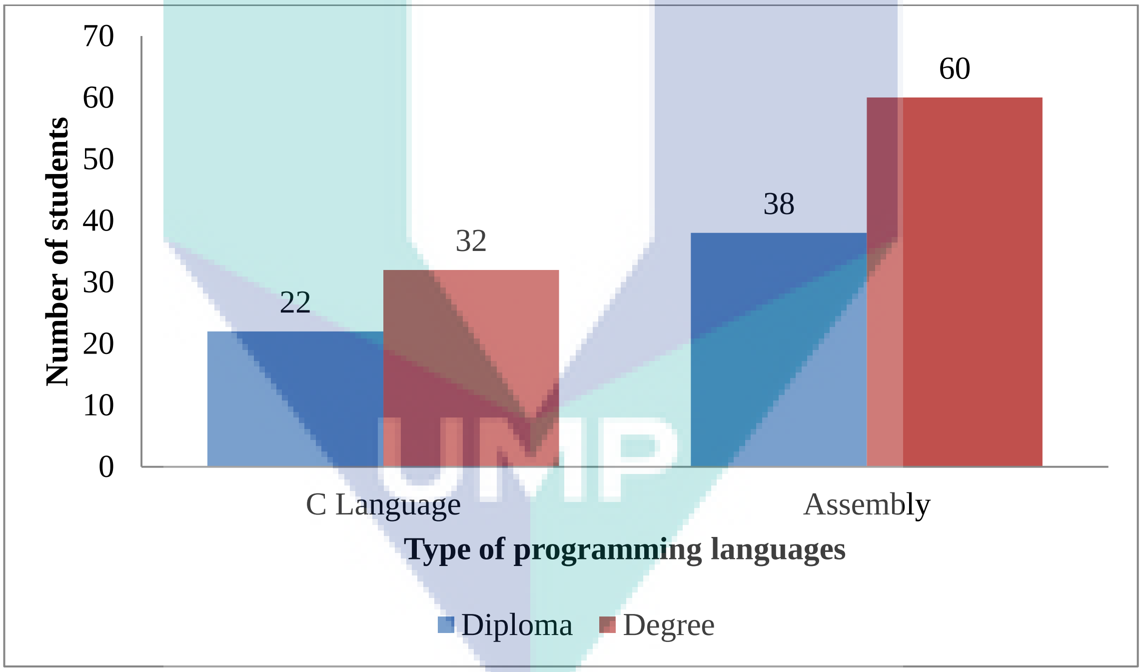


Figure 2.7: Bar graph on number of students with type of programming language

From the Figure 2.7, students in FKEE mostly prefer to develop a program code by using assembly language, with most probably with MCS51 and Freescale HC11 microcontrollers. This is because; our department has so far retained our assembly language programming requirement in class project. On the other hand, C language as programming language, where students use to implement the program structure in PIC microcontroller. Although C is a popular language for development in many of these microcontrollers, assembly language is still an important tool. Because a typical embedded processor runs without an operating system, programmers are free to use the raw capabilities in any way they choose. Assembly language provides the most natural and direct access to the processor's capabilities. Some program like MikroElektronika and MPLab uses IDE platform in programming microcontrollers. This is the easiest programming tool where extensive of reference notes available in C language. From this result, the MicroEVAT system are proposed to offer with these two types of programming language where the MCS51 and Freescale HC11 use assembly language, and PIC Microcontroller uses C language.

Figure 2.8 shows the bar graph on the elements of microcontroller studies with number of students.

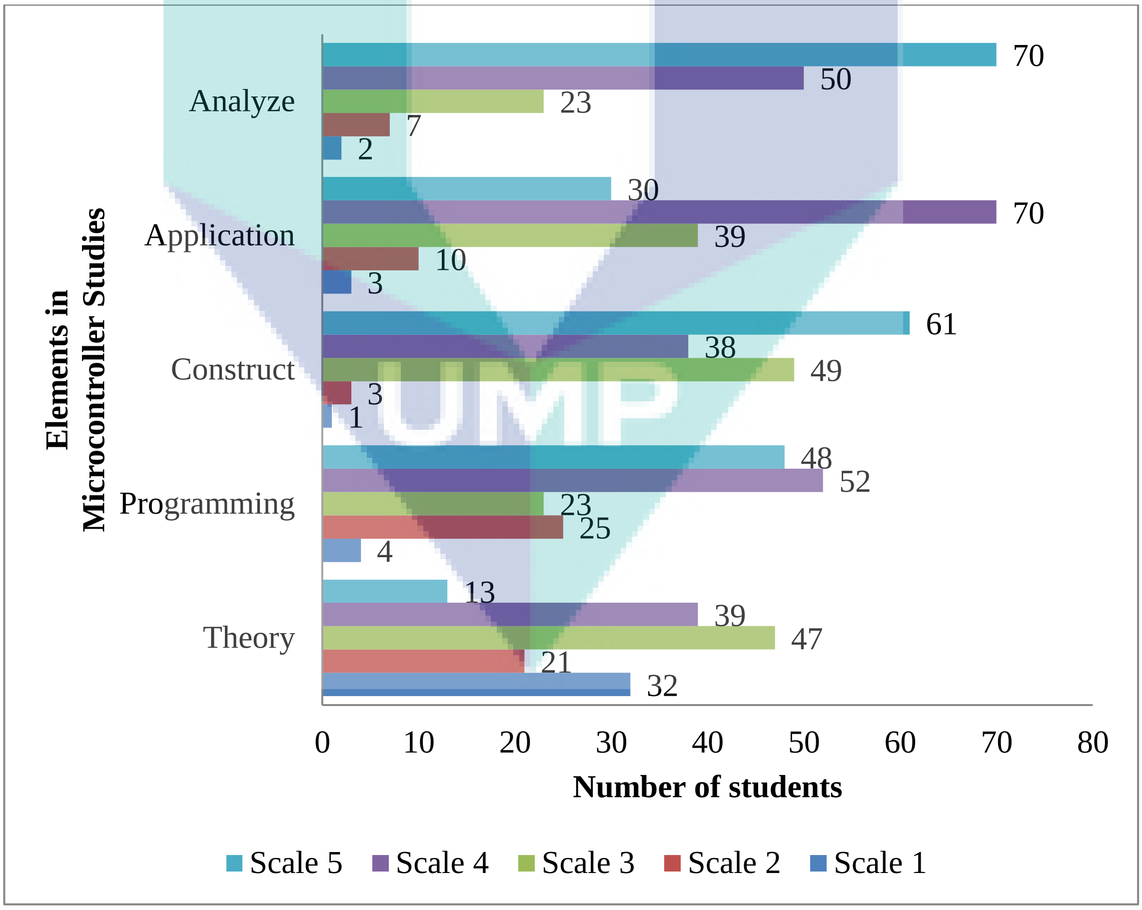


Figure 2.8: Bar graph on elements of microcontroller studies with number of students

Based on Figure 2.8, the elements of microcontroller studies are scaled from Scale 1 to Scale 5, where Scale 5 is considered very difficult. Out of 152 respondents, 65.8% found that theoretical elements are easy to understand since the assessments are done during course hours. However, the rest of elements studies, where 80.9% sees the programming are tough since requires programming skills with and without the hardware module. The students could not see the relevance between the programming and the application module since do not have a proper medium of teaching this core subject. Constructing a basic system using microcontroller, 97.4% found that the task is very difficult, where students have to build up a simple microcontroller system by using the provided electronics components by the faculty. This could take a larger cost and consumes a lot of time. Students found that applying a simple application module and analysing the output seems to be difficult where 91.5% of total students rate the application module as a tough elements. In a class project, students will be evaluate based on application module where they have to come up with a simple pattern program but in the end some of them could not come up with the project outcome due to the error of constructing the application module. Student failed to analyse and troubleshoot the system as they rate analyse elements with 94.1% of them found this element to be difficult.

As conclusion based on Figure 2.8, the MicroEVAT system can be develop with attaching the system board together with memory and application module. This can help the students by understanding the microcontroller's architecture and constructing the modules. Students also can do programming based on application module that attached with the MicroEVAT system without any limitation and could come up with more input and output devices. The questionnaire that been conducted is attached in Appendix D with the result of total 152 respondents.

2.6 RELATED WORK

Since the 1990s, a few research results on small-size development in embedded system evaluation tool have begun to appear in publications (Chandrakasan et al., 1992). During this time, experiments on low power in CMOS evaluation systems were conducted but were severely limited by the lack of software interface and limited memory programming. As an alternative approach, they often used an expanded mode system approach attached to the system board to allow a free-form of expandability but limited address range due to the size of external memory for programming. This system consists of simple construction of system board with Intel 80C51 as base core. The interfacing software using MS-DOS, which that time the most advanced available user-interface medium between personal computer and microcontroller. This made the problem easier when Mukherjea and Stasko (1993), comes up with a research that the user able to use the development board by using terminal software to establish a link between personal computer and microcontroller. However it is impossible to create a simple firmware program because there is no microcontroller in year 1993 that does have internal EEPROM for storing purposes. Until Microchip (1994), developed a first ever microcontroller that have EEPROM that can store firmware and hence a break-through in new achievement of embedded research.

According to Wild et al. (1997), a research on low power consumption of microcontroller design system was designed using MC68HC08 chipset. The system initially develop without using IDE that can be integrates with the microcontroller because using unilateral transistor, this design can be operated between 0.9V until 1.6V at 1 MHz of clock speed. For that a low-voltage graded-channel MOSFET (LV-GCMOS) has been developed for this type of application by using channel engineering both in lateral and in vertical direction. Although the system can provided as the requirement stated as above, it does have limited I/O application that enables to the user to upgrade in future and also does not provided external memory space for programming. This requires another study that developed by Hedley and Barrie (1998) and also researchers from University of Puerto Rico by Perez-Quinones and Cruz-Rivera (1999), with development system using the same family of microcontroller. A laboratory system using MC68HC11 that can be used using either Windows Programming 11 (WP11) or BUFFALO monitor software becomes easier to learn as problem based learning approach has been implement in class environment. However, these systems face the same fate as previous work that limited programming space and some of the devices focusing more on simulation and debugging at program code level. This can let the users to preventing from learning in the practical part of embedded researches.

At the beginning of millennium, many studies on 16 and 32-bits were introduced in academic syllabus. Systems like, TLCS900, ARM-7, and Power PC 555 microcontroller has been implements in education tool as a break-through in new level in embedded researches. In National Sun Yat-Sen University in Taiwan, Huang et al. (2002) proposed a stand-alone ARM-7 microcontroller system using In-Circuit Emulation Module can be a new way on teaching in class. The system offers extra debugging and testing mechanisms such as single stepping, breakpoint settings and detection, and internal resource monitoring and modification to support users in developing and maintaining the target system's hardware and software. They have successfully demonstrated the module's functionality and retarget ability by integrating it with 32-bit ARM-7 microcontroller. The user at the beginner level stated that in their conducted survey, feels that the introduction of this type of microcontroller seems to way advanced since they did not familiar with a simple 8-bits system.

A new trend of teaching has been introduced by Ferreira et al. (2005) using the system called MILES – Microcontroller Learning System, which is a multifunctional module for microcontroller based system design and applications learning. MILES combines a hypermedia software tool running on a personal computer with a hardware design board based on a flash PIC microcontroller. Likewise, as an educational tool like MILES and UDM-EVB (Stoltz, 2005), the system is not meant for any particular curriculum, but can be readily adapted to any secondary school or university course. Because of these systems, the modular concept it is also very appropriate for self-learning as it allows construction of actually required components, thus reducing the cost of the system. But still, using an old technology such as RS232 communication led to current years of researches till today. Table 2.2 shows the previous works that has been done by researchers all over the world in developing a better educational tool.

Table 2.2: Summary of previous work

Year	Description	Microcontroller	Development Environment	Summary
1997	A 0.9V Portable Microcontroller Kit by Wild et al. (University Politehnica Bucharest, Romania	MC68HC08	None	• Low power consumption • Limited I/O with wireless application • No external memory
1998	Microcontroller System Laboratory by Hedley and Barrie (University of Sydney, Australia)	MC68HC11A1	WP11	• Using PBL approach with ICSP method • Limited expandability and programming space • Robustness
1999	Microcontroller System Laboratory using IDE by Perez-Quinones and Cruz-Rivera (University of Puerto Rico)	MC68HC11	BUFFALO Monitor	• Simulator with less I/O module • Simulation based project
2001	Microcontroller based Stepper Motor Application by Flammini et al. (University of Brescia, Italy)	TLCS900 (16 bits)	Sanyo Data Acquisition IDE	• Real time analysis • Stepper motor application
2002	Embedded In-Circuit Emulation Module by Huang et al. (National Sun Yat-Sen University, Taiwan)	ARM-7 (32 bits)	None	• Break-through in 32 bits learning system • Stand-alone and a few number of application module
2002	Motorola 68HC11 Platform by Striegel and Rover (Iowa State University, Ames)	MC68HC11F1	BUFFALO Monitor	• Longevity and limited debugging • Designed to include in academic syllabus
2003	Microcontroller based System for Bio-Medical Engineering Student by Cho et al. (Inje University, Kimhae)	Intel 80C51	Action Script	• Complete IDE • Without application module
2005	Electronics Instrumentation Project by Ferrero Martin et al. (University of Oviedo, Spain)	PIC16F877	None	• A complete system with major use of sensor • Wireless topology • Designed as testing devices, not a development system
2005	MILES by Ferreira et al. (University of Vigo, Spain)	PIC16 Series	MPLAB IDE	• Expandable as a complete system of PIC16 for beginners • Hypermedia content • Using RS232 for communication and ICSP method

Table 2.2: Continued

Year	Description	Microcontroller	Development Environment	Summary
2005	The UDM-EVB by Stoltz et al. (University of Detroit Mercy, USA)	MC68HC12	BUFFALO Monitor	• Stand alone and portable • Limited expansion of teaching module
2006	Microcontroller Learning Kit by Khan and Ninad (Concordia University, Montreal)	Intel 8051	CodeBuffer	• Low cost using LPT port • Without external memory
2006	Linux Laboratory System by Wild (University Politehnica Bucharest, Romania)	MC68HC11A8	ICC11 v. 50	• Limited debugging • Easy to learn, large number of textbook • Already operational
2007	E-Blocks by Matrix Multimedia Corporations	PIC, Atmel AVR, and ARM	FlowCode	• Flexible, expandable and cost saving • Operate only one microcontroller at the time • Only I/O module constructed by the company can be used
2008	Microchip PIC Microcontroller (MPICds) based Application by Smolnikar and Mohorcic (Jozef Stefan Institute, Slovenia)	PIC 16F877	MPLAB IDE	• Starter kits for PIC beginner • Software available free from the Internet • Using ICSP mode for programming • No serial interface needed • Focusing only one type of microcontroller
2008	Metrowerks Educational Package Striegel and Rover (Iowa State University, Ames)	Power PC 555 (32 bits)	Code Warrior 6.0	• Using more powerful microcontroller • Low cost with excellent IDE • No custom I/O board
2009	MINI 11 by Sapein et, al. (University Malaysia Pahang, Malaysia)	MC68HC11E1	Monitor Program using VB GUI	• A complete system of HC11 for class environment • Using SCL approach • Portable and expandable • No external memory space and using only serial interface
2010	Project based Laboratory by Lee et al. (Lunghwa University of Science and Technology, Taiwan)	ARM-7 (32 bits)	None	• Real time application • Use of 32 bits system to only operate line following robot • Not a development system

2.7 SUMMARY

Educational tool has been an important element in the years of academic world and always the result of experience transformations in terms of which they are to be applied. One of defining characteristics of engineering education is the use of models for problem solving. A variety of graphical representations are thus used to illustrate the behaviour of object of interest. In a sense of microcontroller application design, where mastering of digital circuit construction and programming skills at the same time are essential, circuit schematics are used to represent the hardware, while different forms of block diagrams are used to illustrate the software execution flow. Comparing the product design cycle to the educational process, where a specific topic should be covered from its theoretical and practical point of view within a limited time frame, similar challenges can be identified, and efficient development tools are playing a crucial role in overcoming them (Wolf and Madsen, 2000).

In order to develop a reliable multiple microcontroller hardware platform, the MicroEVAT should follows the guidelines of choosing a suitable devices that fits to academic syllabus in Universiti Malaysia Pahang. Each criterion that be used in MicroEVAT must follows a designing assumptions which are the application's software design is usually a cyclic process of writing and testing the program code, its downloading to a microcontroller as well as verification should be supported. The system must be structured in a modular fashion to assure maximum flexibility and rapid prototyping. The choice of microcontrollers should be limited to those where the transfer of the firmware can be performed using monitor program module. The selection of microcontrollers is further restricted to those available in the PDIP and PLCC package types. The programmer must offer a possibility to download a firmware from PC to microcontroller using serial port or using FTDI protocol. The programmer must be designed in such way that it can be permanently connected to the main board, i.e. when programming or running the application. All microcontrollers' I/O pins must be accessible on the main board plug-in module connectors.

Finally, the research in this thesis finds it's significant in the establishment of a systematic methodology in design of multiple microcontroller hardware platforms that optimized for academic environment by using commercially available components such as microcontroller system, ICs for logical design, computers and communication devices. The microcontroller that has been chosen as MicroEVAT system because of its usage in educational syllabus of Universiti Malaysia Pahang compared to other microcontroller families in the market. At the university level, the MicroEVAT can be used as educational equipment in courses dealing with microprocessor, microcontrollers or embedded systems, covering both the programming and the hardware peripheral design support. Due to the growing need of interdisciplinary education the system is also an attractive solution for students of other engineering sciences, e.g. computer, mechanical, chemical or industrial engineering education. Some subjects that the system can aid covering include computer architectures, digital signal processing, mechatronics, automated measuring and etc.

This research aimed to achieve full design system that come with complete GUI based IDE software. These involved several steps such as hardware and software implementation, stability and control analysis hardware and software integration and test-run the MicroEVAT system. The research work presented in this thesis can also be used as solid basis for further design of microcontroller development operations. A multiple microcontroller hardware platform can be regarded as major breakthrough in the effort of designing more complicated fault-tolerant microcontroller design system which in case of failed synchronisation of application system that can ensure the continuation of the task given or switch to an reset position where the initial state of microcontroller previous setting before.

CHAPTER 3

HARDWARE DESIGN

This chapter discusses the architecture of MicroEVAT with implementation of the hardware design. Each of the modules will be described in detail and the complexity is elaborated clearly.

3.1 INTRODUCTION

Hardware implementation can be divided into three function sections. They are system board, memory board and application board. Hardware design for the system and memory board are based on two modes; expandable interfacing and non-expandable interfacing. These modes enable the user to use either to access external RAM, ROM, and I/O port or to use the internal architecture system. However, some ports are unavailable if the microcontroller operates on expandable mode since they are used for address bus and data bus. Since the limited I/O port available in the microcontrollers, address decoder such as SN74LS138, latch SN74LS373/4, and buffer SN74LS244 are used for connection to various I/O devices.

Some of the I/O devices are LED, Bar Graph, Seven and Multi-Segment Displays, Traffic Light Module, 8 x 8 Dot Matrix and Dual-In-Line Package (DIP) switches. The remaining application which includes Keypad, Stepper Motor, Liquid Crystal Display (LCD), Graphic LCD (GLCD) and DC motor are connected the available I/O port in the microcontroller. Figure 3.1 shows the proposed flow diagram of the MicroEVAT system.

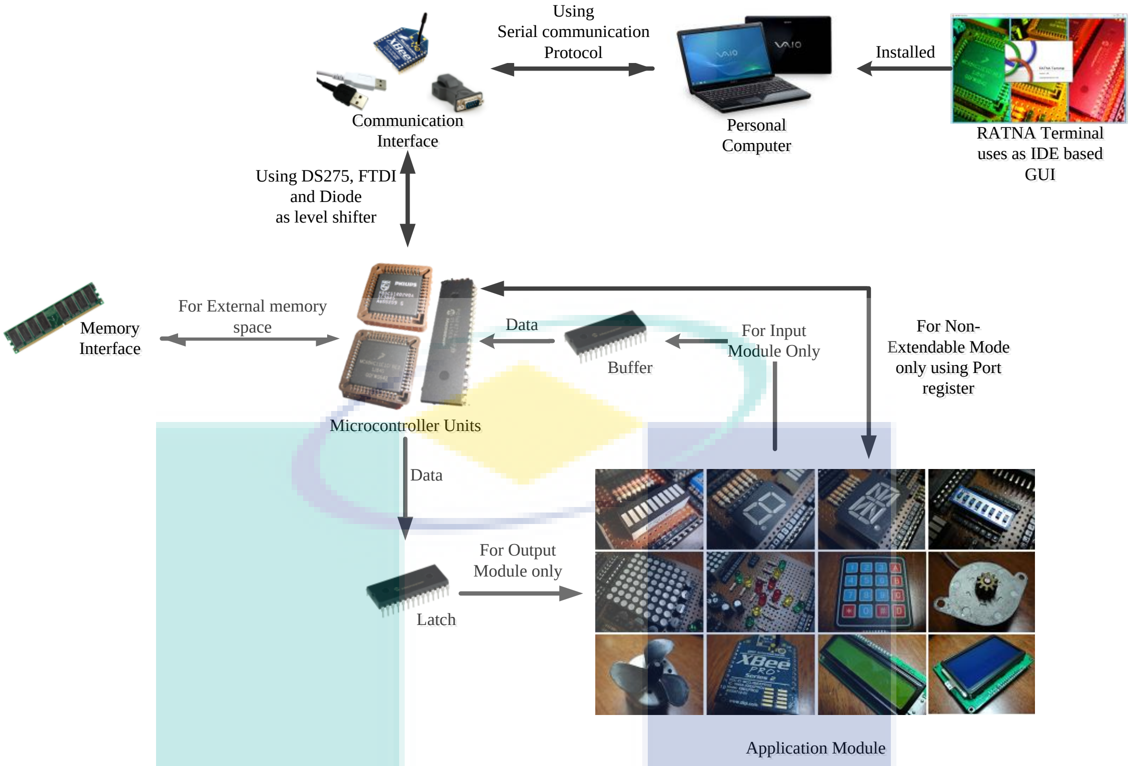


Figure 3.1: The Proposed Flow Diagram of the MicroEVAT system

3.2 SYSTEM BOARD OF MICROEVAT

The system board consists of microcontrollers, power units, and communication interfacing units. The power units refer to supply voltage of direct current (DC) from voltage regulators while the communication interfacing units refer to serial communication consist of RS232, USB and ZigBEE protocol. The interfacing to I/O port requires the use of data bus and address bus of designated of microcontroller. Each of the microcontrollers has separate power supply to avoid unnecessary power usage when currently only one of microcontroller is being use in the system. Although the systems consume separate power supply for each microcontroller, the communication interface of each microcontroller is combined into one serial communication link to the personal computer. This is one of the advantages of the design as the user does not require to plug out the serial or USB cable whenever the user operates on the different microcontroller. Figure 3.2 shows the block diagram of the MicroEVAT system board with power and communication interface.

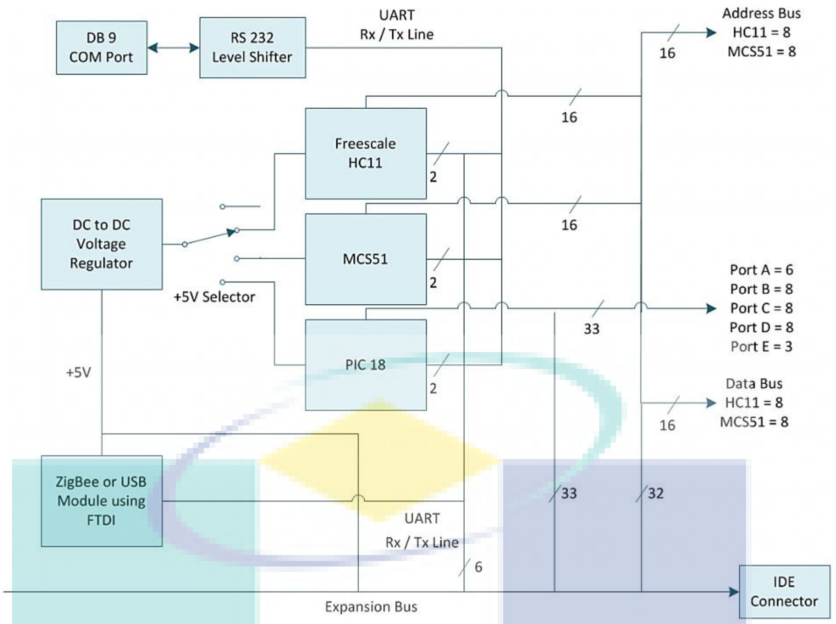


Figure 3.2: The block diagram of the MicroEVAT system board

3.2.1 RESET Circuit Module

Figure 3.3 shows the schematic diagram of reset circuit modules for MicroEVAT system. A bidirectional control signal acts as an input to initialize the microcontroller to a known start-up state. It also acts as an open-drain output to indicate that an internal failure is detected due to either the clock monitor or computer operating properly watchdog timer circuit. However, the reset pin for MCS51 is an active high logic while for Freescale HC11 and Microchip PIC is active low logic.

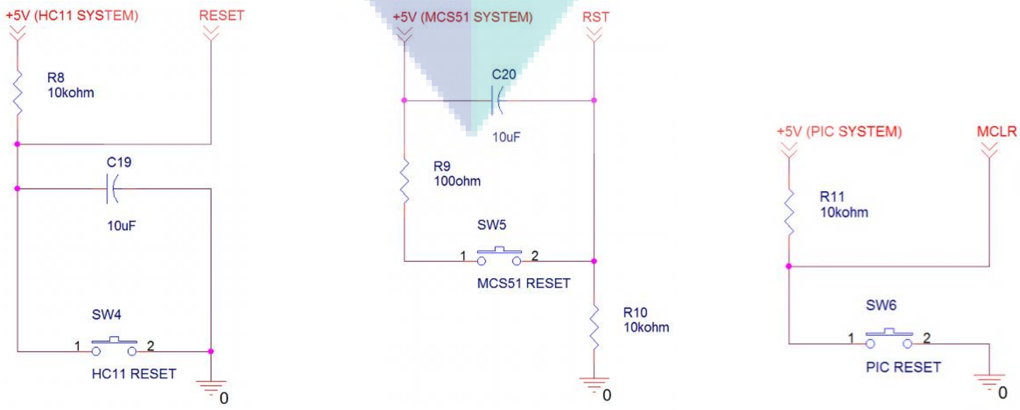


Figure 3.3: The schematic diagram of reset circuit modules for MicroEVAT system

To reset the MCS51 microcontroller, it is mandatory to drive the reset pin high at a desired time. Based on the Figure 3.3, if the push button is pressed, the current flows directly to ground through the reset button or is considered as short circuit. Thus, it is predicted there is no current flow through the reset pin. Since no current flowing through the reset pin, the capacitor will discharge and giving the reset pin in high logic. A high on this pin for two machine cycles will automatically reset the device. Once the push button is released, the current flows through the capacitor and drives it high in a short period. This is because the capacitor is charging to $5V_{DC}$ in a very short time. The logic of reset pin will turn to low. A resistor R10 to ground permits a power-on reset using only a capacitor C20 to logic V_{CC}.

To reset the Freescale HC11 and Microchip PIC microcontrollers, the logic of RESET and MCLR pin requires a logic low to perform an external reset status. The microcontroller distinguishes between internal and external reset conditions by sensing whether the reset pin rises to logic 1 in less than two E-clock cycles after an internal device releases reset. When a reset condition is sensed, the RESET pin is driven low by an internal device for four E-clock cycles, and then released. Two E-clock cycles later it is sampled. If the pin is still held low, the CPU assumes that an external reset has occurred. If the pin is high, it indicates that the reset is initiated internally by either the Watch-Dog Timer system (COP for Freescale HC11) or the clock monitor. The Microchip PIC devices, on the other hand, provide a MCLR noise filter in the MCLR Reset path. The filter will detect and ignore small pulses. Hence capacitor does not require in Microchip PIC reset circuit module.

3.2.2 Serial Communication Module

Asynchronous serial communication uses either RS232 or FTDI level standard to communicate between the microcontroller and computer. This is because the signal sends from the microcontroller propagates along the line and exposes to various form of noises. As a result, it may cause a voltage drop in the connections. Therefore this level shifter devices are used to increase the voltage level in order to ensure data can be transmitted and received by the microcontroller. Figure 3.4 shows the schematics diagram of Serial Communication module using DS275 as level shifter.

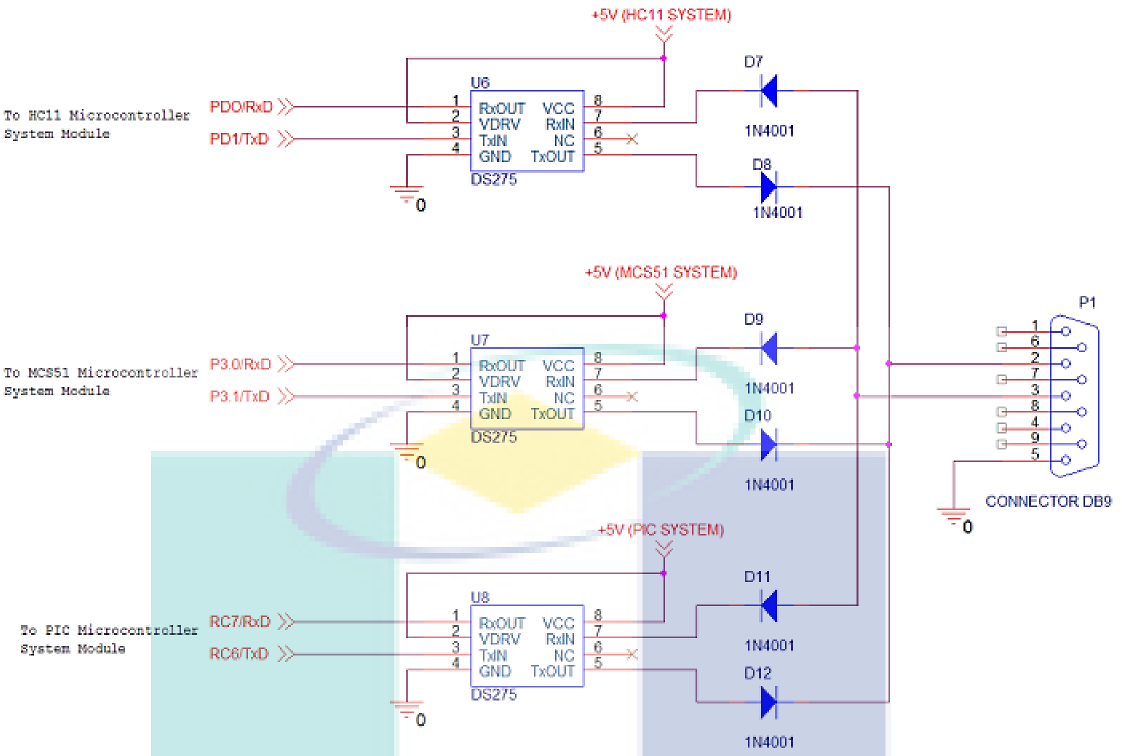


Figure 3.4: The schematics diagram of Serial Communication module using DS275

Diode 1N4001 D7, D8, D9, D10, D11 and D12 are used to provide a single way of flow current to the MicroEVAT system. Using more than one device in a single port may cause de-synchronisation at terminal window. Thus, by applying the diode, the MicroEVAT system can power-up the devices without plug-it out the serial cable from the PC. The voltage potential differential of DS275 is in optimum level since the voltage drop of every single diode is 0.7V and the minimum voltage required in DS275 is 3.3V.

Alternatively, in MicroEVAT system offers Universal Serial Bus (USB) and ZigBEE protocol as communication interface with the PC. Recently, serial port of computer has been replaced with USB. Alternatively, most developer chooses USB to serial converter to obtain virtual serial port. The level shifter is still necessary for UART interface. Using this FTDI level shifter may have an option to convert the RS232 to USB support, since no external software required in order for this device to function. Figure 3.5 shows the schematics diagram of USB module using FT232RL.

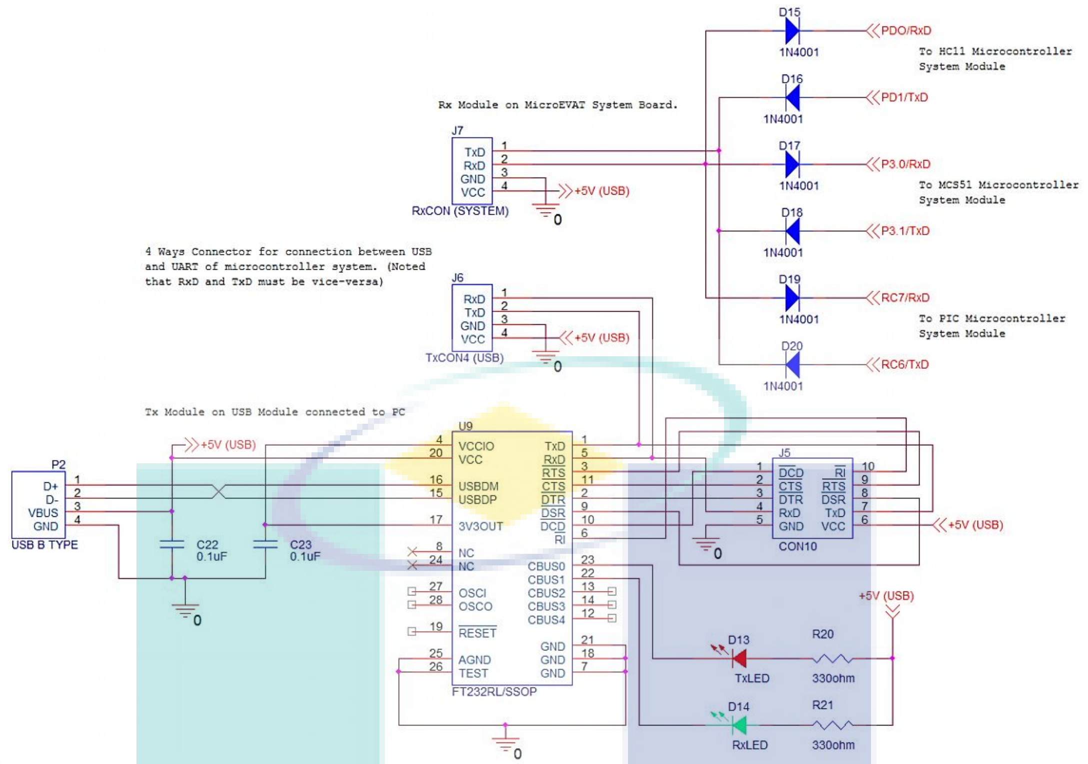


Figure 3.5: The schematics diagram of USB module using FT232RL

LED D13 and D14 are used to indicate the progress of transmitting and receiving data from the PC to the MicroEVAT system and vice versa. The application uses diode as similar function as the serial communication using DS275. Since the availability of this FTDI device is only in shrink small-outline package (SSOP), there is impossible to the author to implement this device in the MicroEVAT prototype system. A USB to UART converter is used in this MicroEVAT system. This USB to UART converter which offers USB plug and play, direct interface with microcontroller and it provide low current 5V supply from USB port.

ZigBEE protocol is another choice to communicate to the PC to issue the command from the host. This device provides a common footprint shared by multiple platforms, including Multi-point and Mesh-point topologies, and both 2.4Gz and 900 MHz solutions for long range deployment. The ZigBEE can be replaced for another, depending upon dynamic application needs, with minimal development, reduced risk and shorter time. ZigBEE multipoint RF modules are ideals for applications requiring low latency and predictable communication timing with providing quick, robust communication in point-to-point, peer-to-peer, and multi-point or mesh-point configuration. These wireless multipoint devices enable robust end-point connectivity with ease. It offers high performance for wireless communication. Figure 3.6 shows the schematic diagram of ZigBEE module.

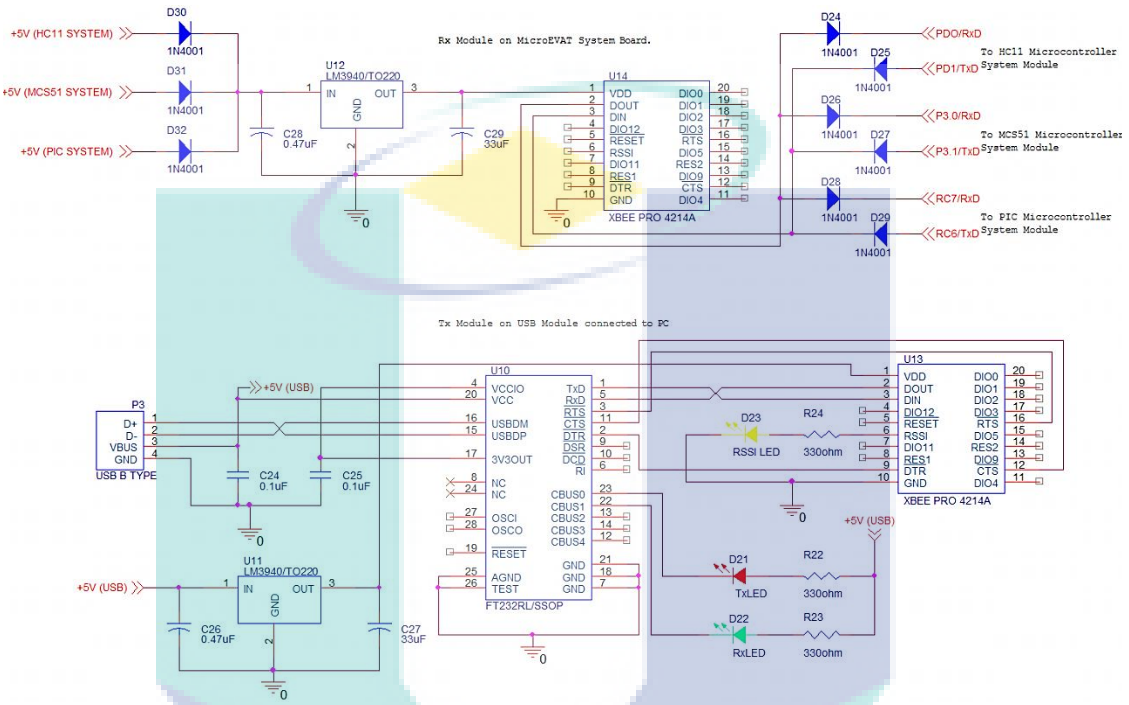


Figure 3.6: The schematics diagram of ZigBEE module using XBEE PRO 4214A

3.2.3 Mode Selector Module

The MicroEVAT system is designed to allow the user to select mode of operation either to access internal or external memory of the microcontroller. However, only Freescale HC11 and MCS51 offer this feature. Table 3.1 shows the mode of operation for Freescale HC11 microcontroller while Table 3.2 shows the mode of operation for MCS51 microcontroller.

Each of modes of operation's function is briefly explained in later section in this chapter. Figure 3.7 shows the schematic diagram of mode selector module for Freescale HC11 and MCS51 in MicroEVAT system.

Table 3.1: Mode Selector for Freescale HC11

MODA	MODB	Mode
0	0	Bootstrap Mode
0	1	Special Test (BUFFALO 3.4)
1	0	Single Chip
1	1	Expanded Mode

Table 3.2: Mode Selector for MCS51

EA/VPP	Mode
0	Single Chip
1	Expanded Mode

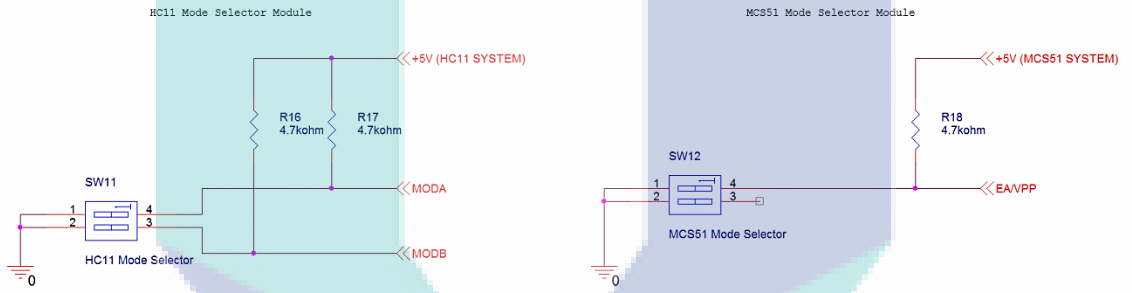


Figure 3.7: The schematic diagram of mode selector module

3.3 MEMORY BOARD OF MICROEVAT

The main purpose of the implementation of external memory storage is to offer a microcontroller expansion capability beyond the on-chip resource to avoid a potential design bottleneck. The resources can be either memory or I/O. The memory storage unit consists of RAM and EEPROM. As mention earlier, RAM is used as a temporary storage for system board and is volatile memory that requires power supply to keep the system operates. EEPROM, on the other hand, is used as a permanent storage for monitor program. In order to extend the memory storage for MicroEVAT system, an address decoder module is designed to enable the microcontroller devices to access the memory content at designated address. It depends on the type of microcontroller either the Freescale HC11 or MCS51. Six address decoder are used in order to create the addresses for memory storage and as well as the application module. The block diagram of the MicroEVAT memory board is shown in Figure 3.8

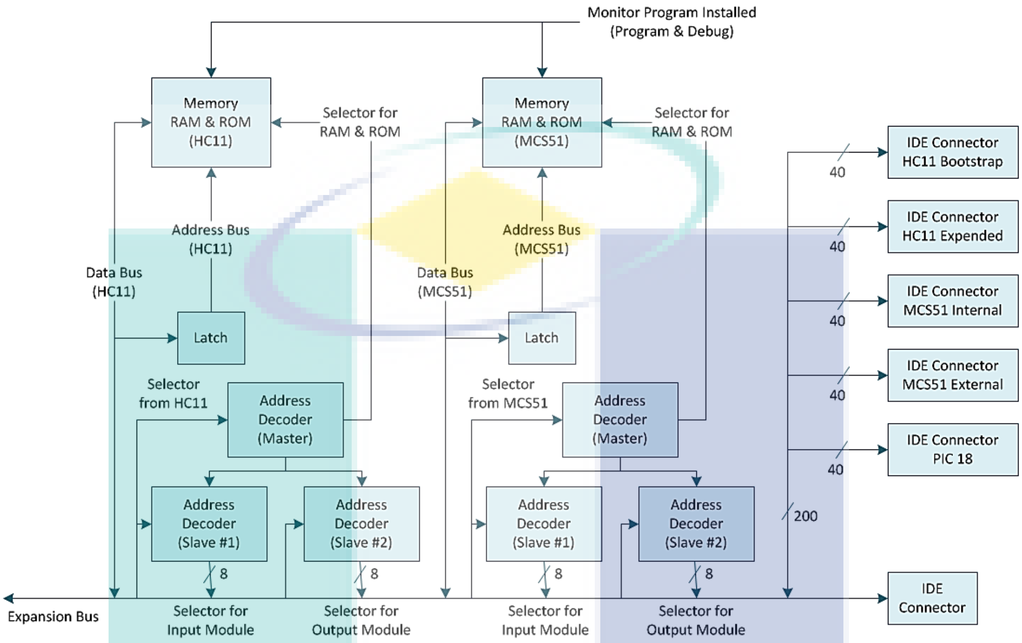


Figure 3.8: The block diagram of the MicroEVAT memory board

3.3.1 Address Decoder Module

Figure 3.9 is an address decoder for MicroEVAT system. As can be seen that two address decoder is used to nest another four address decoder to provide extra I/O device. To enable the nested SN74LS138 decoder for MCS51 device, the output Y1* and Y2* from the Master's address decoder must be connected to the enable pin in the Slave's address decoder. For Freescale HC11, the output Y3* and Y4* from the Master's address decoder must be connected to the enable pin in the Slave's address decoder. Address bus from A8 to A0 are not connected to the address decoder module, therefore it will be ignored. The address range for each device in Master's address decoder is based on the address bus of A13, A14, and A15. Likewise, the address range for each device in Slave's address decoder is depends on the address bus of A9, A10 and A11. The Master and Slave's address decoder can be mapped out based on the logic provided by address bus of Freescale HC11 and MCS51 device in MicroEVAT system.

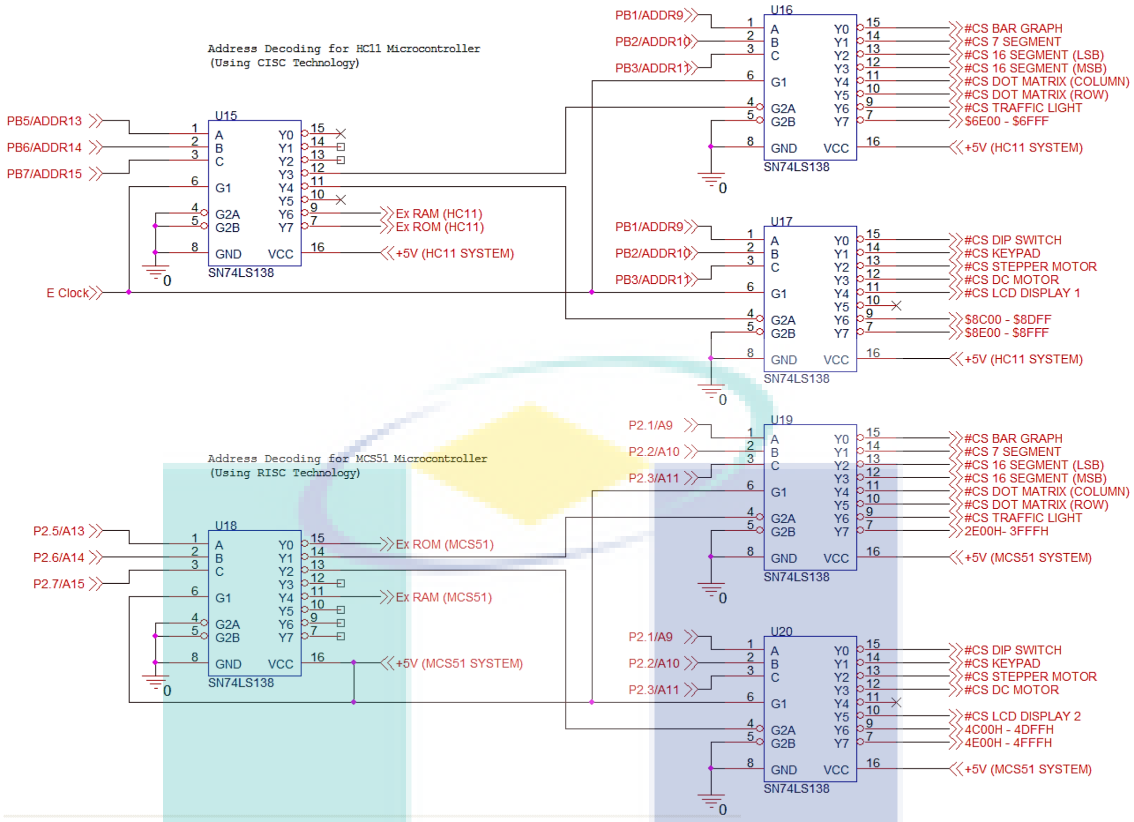


Figure 3.9: The schematic diagram of addressing module using SN74LS138 3-to-8 address decoder

For Freescale HC11, Port B is used for upper address bus and Port C is used for lower address bus and data bus. A latch SN74LS373 is used to de-multiplex the address information and data information of Port C. This is shown in Figure 3.10. The pin Address Strobe (AS) which is, an active-high latch enable signal for an external address latch, connected to Latch Enable pin (G) of SN74LS373 latch. Address information is allowed through the transparent latch while AS is high and is latched when AS drives low. When G is high, port C becomes address bus. On the other hand, if G is low, Port C is data bus. In similar manner, a latch SN74LS373 is also used in MCS51 microcontroller system to de-multiplex between data and address bus. Port 2 is used for upper address bus and Port 0 is used for lower address bus and data bus. The pin Address Latch Enable (ALE) from the MCS51 microcontroller is connected to pin G of latch. This is shown in Figure 3.11. ALE will give output pulse for latching the low byte of the address during an access to external memory.

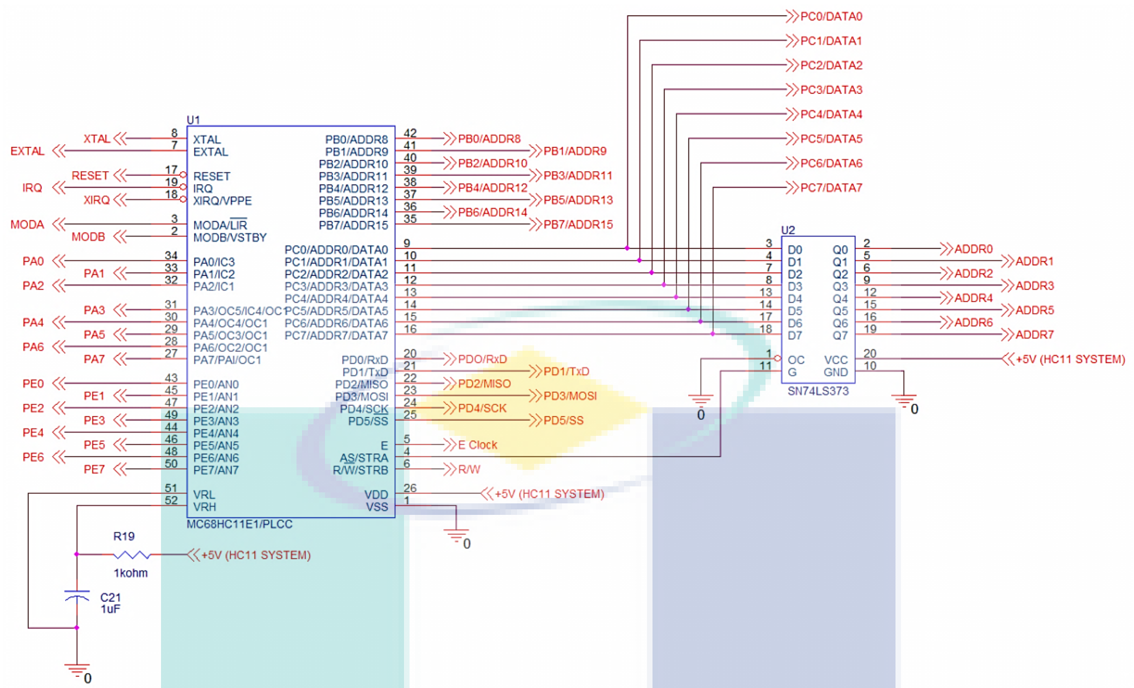


Figure 3.10: The Address / Data De-multiplexing module for Freescale HC11 microcontroller

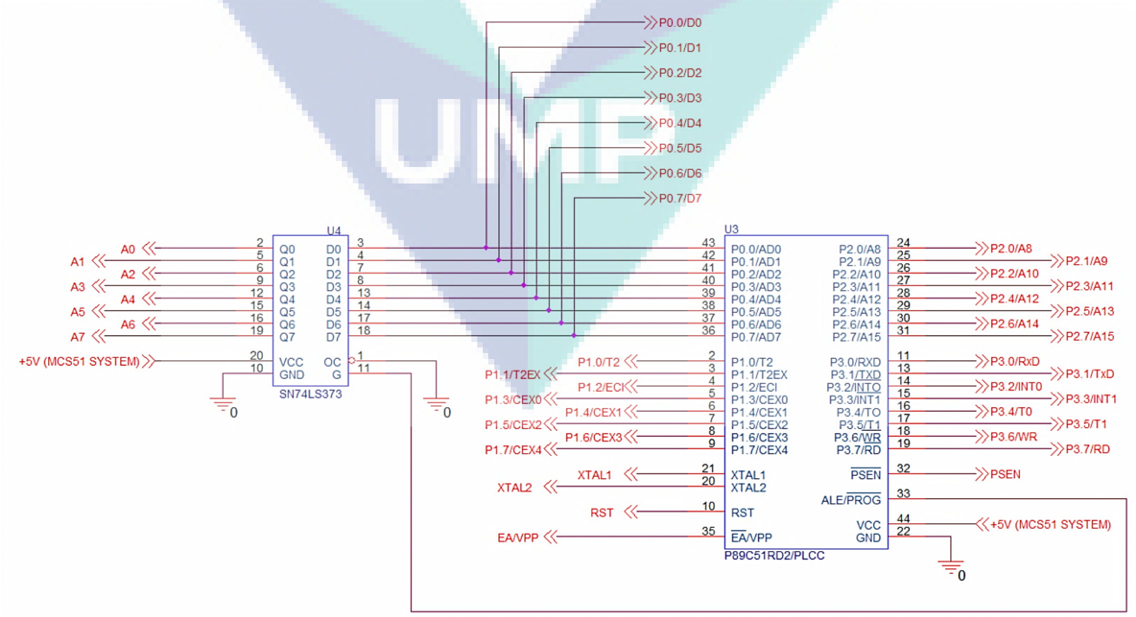


Figure 3.11: The Address / Data De-multiplexing module for MCS51 microcontroller

In normal operation, ALE is emitted twice every machine cycle, and can be used for external timing or clocking. One ALE pulse is skipped during each access to external data memory. ALE can be disabled by setting the Special Function Register (SFR) auxiliary with the value of ‘0’. With this bit set, ALE will be active high logic only during a MOVX instruction.

3.3.2 External Memory Module

If the system is operating in Expended Memory Mode, external ROM and RAM are normally used. Normally, ROM is used as a permanent storage for monitor program. As mention earlier, the type of ROM that being used is Electrically Erasable Programmable Read Only Memory (EEPROM). Normally, this type of memory could erase its content by supplying appropriate voltage using a chip programmer or universal writer. Figure 3.12 shows the schematics diagram of external memory interface using AT28C64B as EEPROM and HY6264 as RAM.

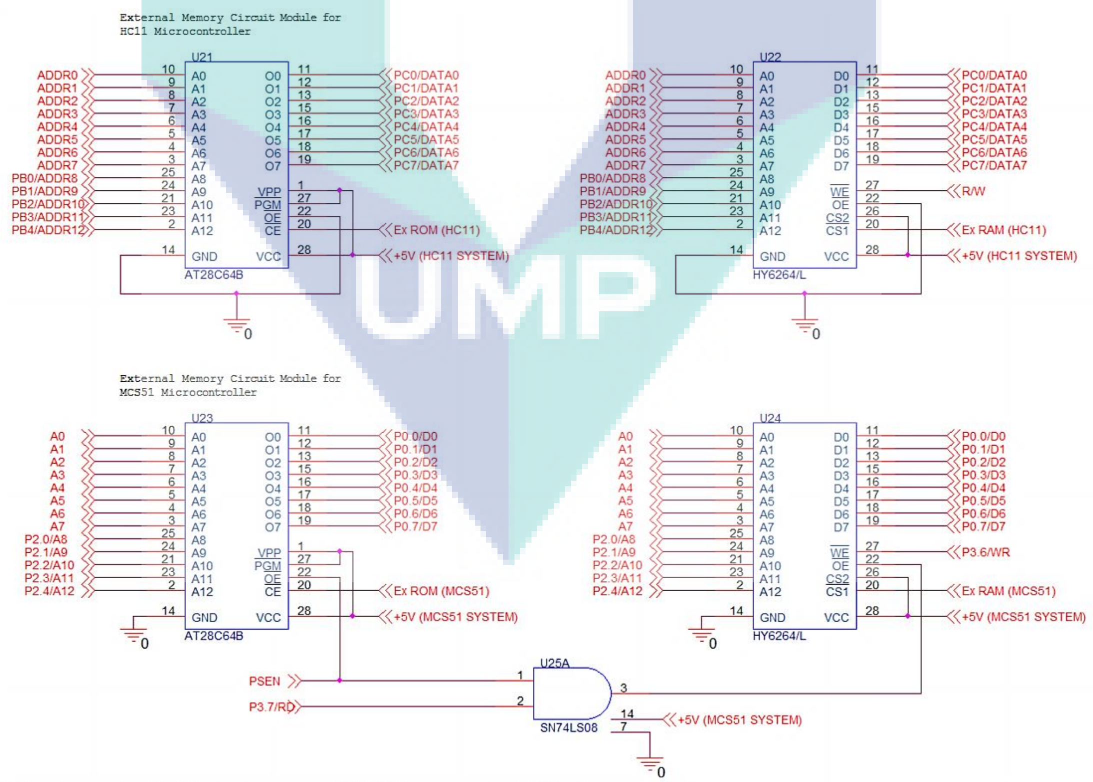


Figure 3.12: The schematics diagram of external memory interface using AT28C64B as EEPROM and HY6264 as RAM

Both the ADDR0 - ADDR7 connections (for Freescale HC11) and A0 - A7 connections (for MCS51) are the output from 74HC373 resulting from de-multiplexing technique of Port C (Freescale HC11) and Port 0 (MCS51), while DATA0 - DATA7 from Port C (Freescale HC11) and D0 – D7 from Port 0 (MCS51) are the input as shown in Figure 3.12. An AND gate SN74LS08 is used in MCS51 external memory interface for enabling the read strobe to the external program memory from PSEN* pin of MCS51 microcontroller. Logic between PSEN* and RD* is explained detail in Table 3.3.

Table 3.3: Logic between PSEN* and RD* for MCS51 microcontroller

PSEN*	RD*	OE* of RAM (Status)	
0	**0**	**0**	(Output Enable for HY6264 RAM)
0	**1**	**0**	(Read from AT28C64B program memory)
1	**0**	**0**	(Write program into HY6264 RAM - where WR* give logic '0' to WE* of RAM)
1	**1**	**1**	(Output Disable for HY6264 RAM)

3.4 APPLICATION BOARD OF MICROEVAT

Figure 3.13 is a block diagram of application modules in MicroEVAT prototype system. The module consists of three inputs and ten outputs device. The former include input modules such as Temperature Sensor, 8-ways DIP switches, and Keypad and the latter consists of output modules such as LED, Bar Graph, 7-Segment Display, 16-Segment Display, Dot Matrix, Traffic Light, LCD, GLCD, DC motor, and Stepper Motor.

Several of the modules are directly connected to the I/O port of microcontroller for non-expandable interface. Due to the limited ports that available in the microcontroller, six address decoders are used to select additional ten I/O devices by using latch and buffer. The latch is an eight-bit edge-triggered register coupled to eight tri-state outputs for enabling the devices. Buffer, on the other hand, is an eight tri-state buffers or line drivers to enabling two input modules.

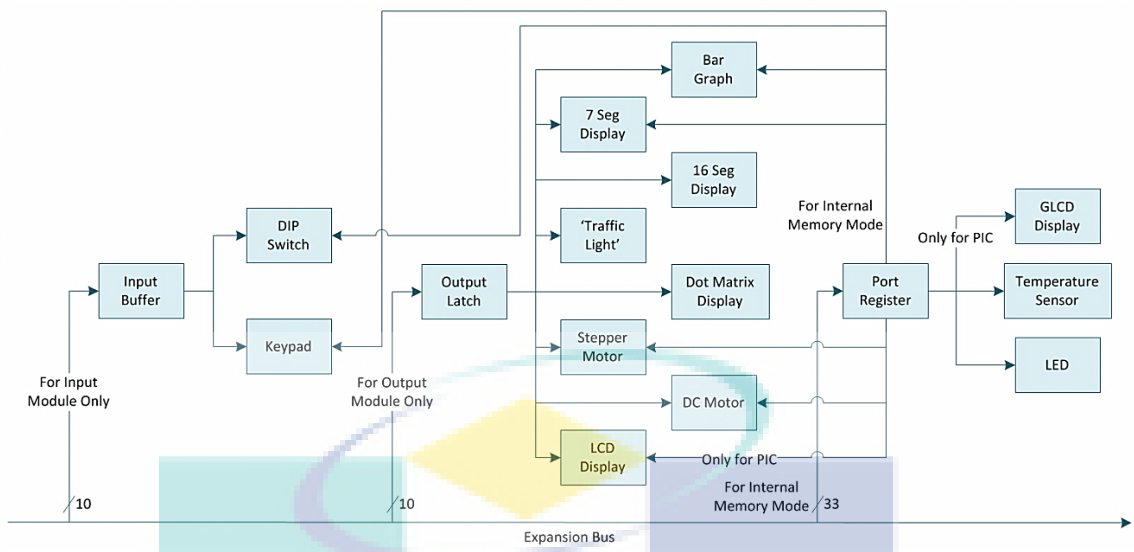


Figure 3.13: The block diagram of MicroEVAT application board

3.4.1 Seven Segment Display Module

A Seven-Segment display is a form of electronic display device for displaying decimal numerals that is an alternative to the more complex Dot-Matrix displays. This module can be operated by using a Binary coded Decimal (BCD) decoder as decoding device to convert the data byte in binary form into decimal display on Seven-Segment display. In the MicroEVAT system, BCD decoder is not used to derive the display from the binary code.

The unavailability of BCD decoder in the system cause the display of segments is deriving from the data byte that connected to the microcontroller. Figure 3.14 shows the schematic diagram of Seven-Segment displays circuit in expandable interface. This Seven-Segment display is operating in common cathode mode.

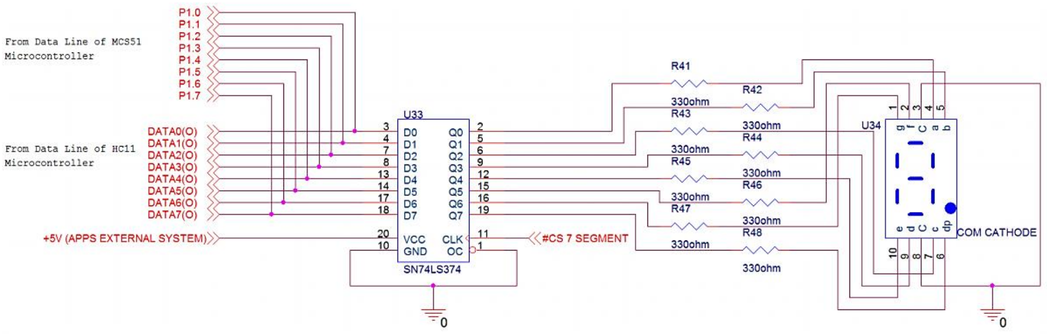


Figure 3.14: The schematics diagram of Seven - Segment display in expandable interface

Figure 3.15 shows the schematics diagram of Seven-Segment display circuit connection in non-expandable interface. The Seven-Segment display is connected without using the latch and the jumper has been used to enabling the module by using active high logic.

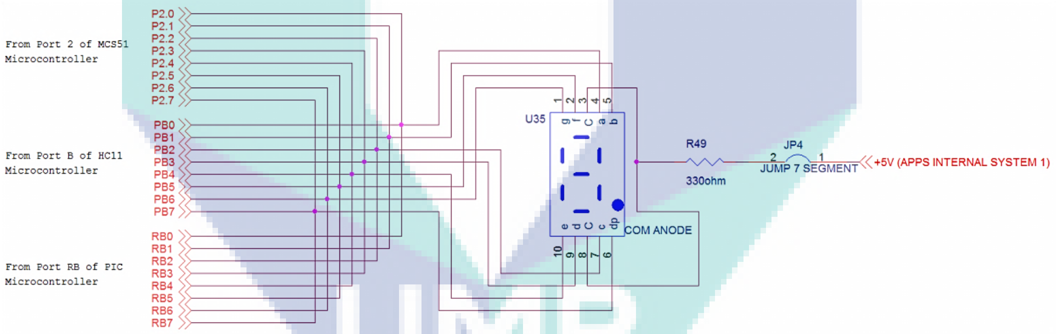


Figure 3.15: The schematics diagram of Seven - Segment display in non-expandable interface

Since the module did not use the BCD decoder, the binary value is manually decoded to turn the Seven-segment display. The detailed is summarized in Appendix E.

3.4.2 16-Segment Display Module

This device can be used to display Arabic numerals and letters of the basic modern Latin alphabet. 16-Segment display devices use fewer elements than a full dot-matrix display, and may produce a better character appearance where the segments are shaped appropriately. This can reduce the number of driver components and power consumption. Figure 3.16 shows the schematic diagram of 16-Segment displays circuit in expanded mode. This 16-Segment display is operating in common cathode mode. Only the expandable interface has been designed for this output module.

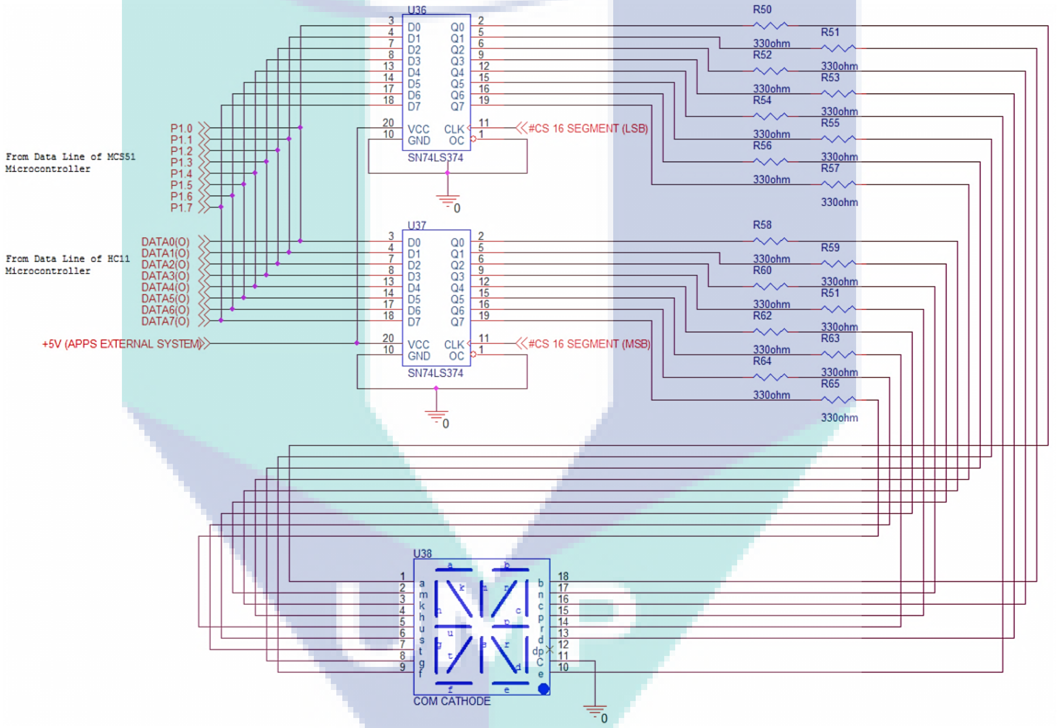


Figure 3.16: The schematics diagram of 16 - Segment display in expandable interface

Based on Figure 3.16, in order to display a single Arabic numeral or a character, the program code must consist of two byte of data, most significant byte (MSB) and low significant byte (LSB). The LSB represents the segment ‘a’ until segment ‘h’ and MSB consists of segment ‘k’, ‘m’, ‘n’, ‘p’, ‘r’, ‘s’, ‘t’ and ‘u’. The encoding performed in program code using lookup table technique in order to achieve better synchronisation display on this device. The details on encoding the 16-segment display are summarized in Appendix E.

3.4.3 Dot Matrix Display Module

The position of the LSB and MSB in the Dot Matrix display must be identified before character or numerals can be display on the module. A single colour Dot Matrix requires two latches to control the colour and display mechanism. Transistors Q1 until Q8 are used for each column to drive the current flow through the Dot Matrix. The scrolling mechanism of the lit on LED from the first column to the last column is applied in this board so as to display fixed or scroll characters. It is controlled by microcontroller system through decoder by enabling and disabling certain latches. In real situation, only one column for each LED dot matrix will lit up at one time. Figure 3.17 is the schematics diagram of Dot Matrix in expandable interface. The data bus is connected to latch input while the latch output is connected to Dot Matrix display.

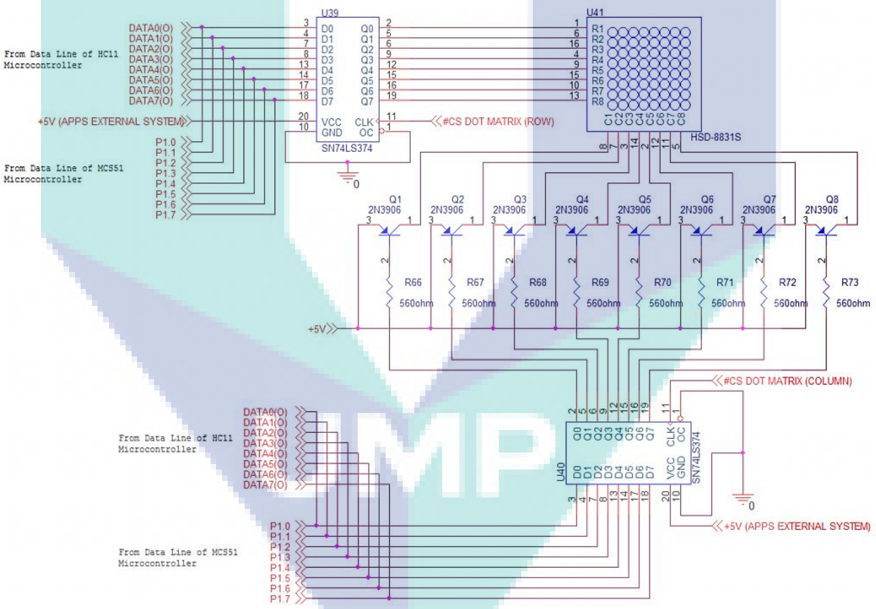


Figure 3.17: The schematics diagram of Dot Matrix in expandable interface

Initially, the first column for LED display blocks is lit up. It is followed by the second column. The process will continue until the last column. Then the process is repeated itself to produce the character that has been assigned. The process is so fast, so that all of the LED looks like lit up simultaneously. The latch plays a vital role in controlling the data into rows and columns for the LED. Two latches are needed for each dot matrix where one of them is used for column. For example, to display B in the

Dot Matrix, a set of hex numbers instructions must be provided to the microcontroller for rows and columns. The numbers are 81, B5, B5, and CB for the row and C3 for column. Logic '0' will light up the LEDs on the Dot Matrix. Figure 3.18 shows character 'B' on the Dot Matrix.

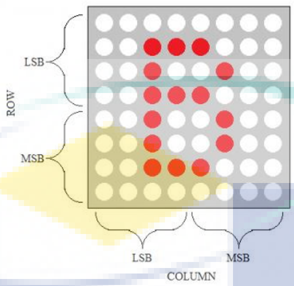


Figure 3.18: Dot Matrix represents character 'B'

3.4.4 Application Examples

To provide the user with experience of implementation in basic digital system in a microcontroller, a simple 4-way junction traffic light experiment is introduced in the application module of MicroEVAT system. The traffic light system uses I/O data bus to control the green lights and the red lights sequence for the 4-way junction. Figure 3.19 shows the schematics diagram of 'Traffic Light' in expended mode.

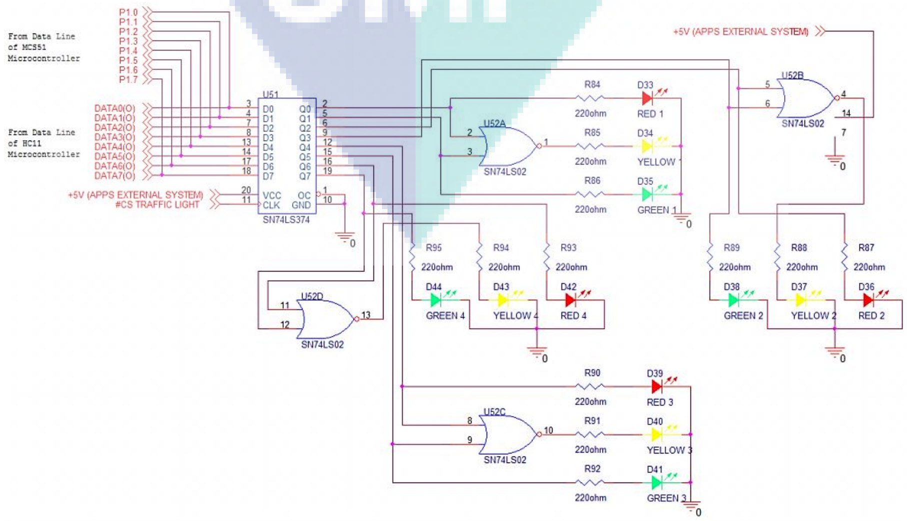


Figure 3.19: The schematics diagram of 'Traffic Light' in expandable interface

By using SN74LS02 NOR gate, the amber LEDs can be controlled by obtaining the logics from green and red LEDs. The sequence of the 'Traffic Light' module is explained in Appendix E.

3.4.5 Keypad Module

A Keypad is a set of buttons arranged in a block or "pad" which usually bear digits, symbols and usually a complete set of alphabetical letters. If it mostly contains numbers then it can also be called a numeric keypad. Figure 3.20 shows the schematics diagram of Keypad circuit connection in expandable interface. The type of keypad used is 4 X 4 Keypad, total of 16 inputs combining of hexadecimal and symbols. The total eight pins from keypad are connected to 5 X 4 Keypad Encoder MM74C923. Four output pins (OA to OD) from Keypad Encoder are connected to buffer of microcontroller. The Data Available (DAVBL) pin from keypad encoder is connected to input buffer bit 7. DAVBL is used to indicate a valid data are available on output OA until OD. Alternatively, 4 X 4 Keypad Encoder MM74C922 also can be used in this MicroEVAT system.

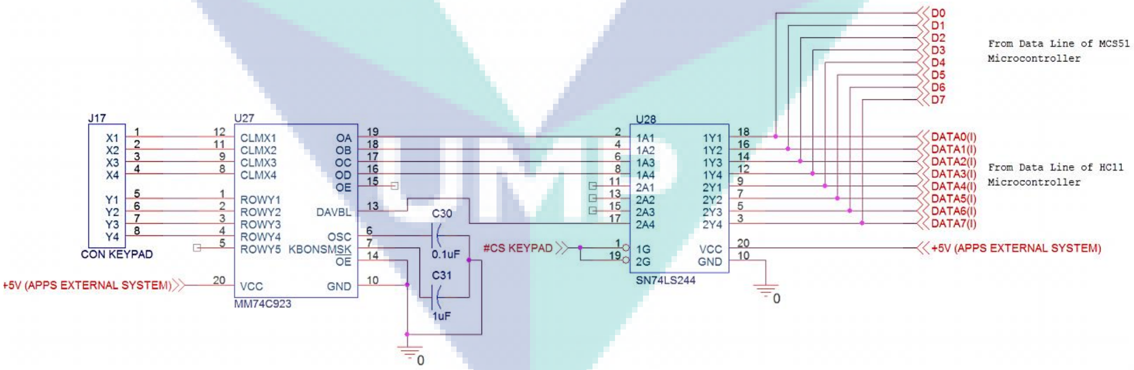


Figure 3.20: The schematics diagram of Keypad in expandable interface

The Keypad also can be constructed in bootstrap mode, so that the Freescale HC11, MCS51 and Microchip PIC microcontroller can have this device connected using the available port register without using buffer circuit. Figure 3.21 shows the schematics diagram of Keypad circuit connection in non-expandable interface.

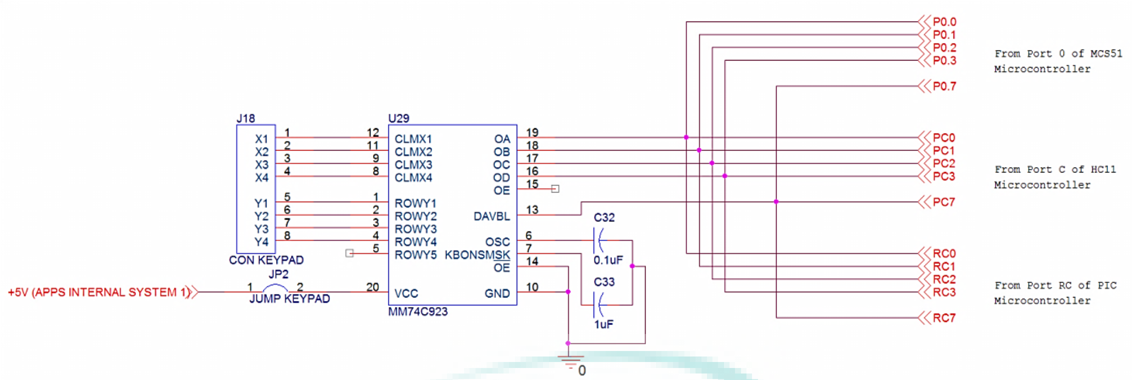


Figure 3.21: The schematics diagram of Keypad in non-expandable interface

3.4.6 Stepper Motor Module

A Stepper Motor (or step motor) is a brushless, electric motor that can divide a full rotation into a large number of steps. The motor's position can be controlled precisely without any feedback mechanism, as long as the motor is optimally sized to the application. Figure 3.22 shows the schematics diagram of Stepper Motor circuit connection in expandable interface.

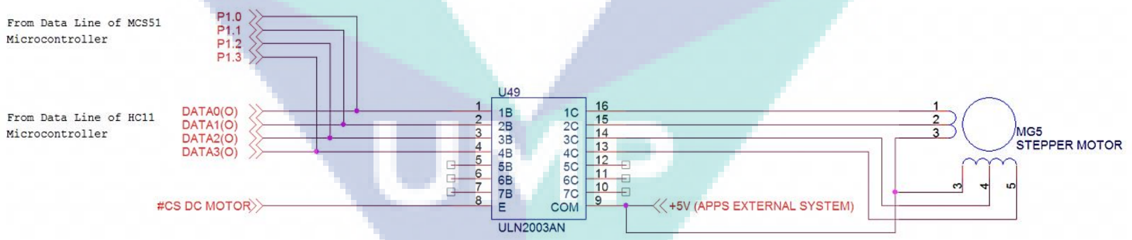


Figure 3.22: The schematics diagram of Stepper Motor in expandable interface

The Stepper Motor also can be constructed in bootstrap mode, connected using the available port register. Figure 3.23 shows the schematics diagram of Stepper Motor circuit connection in non-expandable interface.

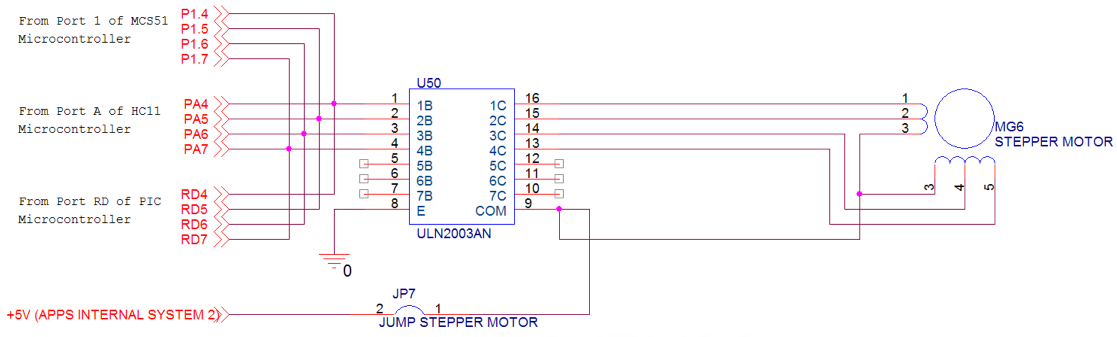


Figure 3.23: The schematics diagram of Stepper Motor in non-expandable interface

A typical Stepper Motor circuit connection is shown in Figure 3.22 and Figure 3.23. Four inputs from data bus or port register are connected to input pins of Stepper Motor driver. Thus, four output pins from Stepper Motor driver are connected to Stepper Motor. The Enable pin (E) pin is an enable pin to active the motor driver. Therefore logic low is supplied to this pin to active the motor. Based on Figure 3.21 and Figure 3.22, this bipolar Stepper Motor gives 48 steps in one rotation. Hence by using formula given by the equation 3.1:

$$Angle\ per\ step, a = \frac{360^{\circ}}{Number\ of\ steps\ per\ rotation, s} \qquad (3.1)$$

where angle per step a can be found from the total number of steps s of 48 to be divided with 360˚. From there, the PM35-048L Stepper Motor has angle per step of 7.5˚.

3.4.7 LCD Module

A liquid crystal display (LCD) is an electro-optical amplitude modulator realized as a thin, flat display device made up of any number of colour or monochrome pixels arrayed in front of a light source or reflector. Figure 3.24 shows a schematics diagram of LCD circuit connection in expandable interface. The output Y4* from second slave's address decoder is connected to NOT gate SN74LS04 before connected to enable pin of the LCD. On the other hand for the MCS51 microcontroller configuration, the output Y5* from second slave's address decoder is connected to inverter and flip flop based of NAND gate SN74LS00 before connected to enable pin in

LCD. The flip flop is required because the LCD provides a multiplex pin for Read and Write operation. On the contrary, the MCS51 microcontroller has separate Read and Write pin for this operation. In addition, there is address bus A0 and ADDR0 connected together with flip flop for MCS51 and pin R/W* of Freescale HC11to the pin R/W* of LCD. Thus, the addresses for LCD are \$8800 and \$8801 for Freescale HC11 microcontroller and 4800H and 4801H for MCS51 microcontroller. Both addresses of \$8800 and 4800H are the address for display setting in LCD and both addresses of \$8801 and 4801H are the address for display character in LCD. Data bus is connected directly to LCD. The potential meter connected to pin V_{EE} in LCD is used to adjust contrast of LCD.

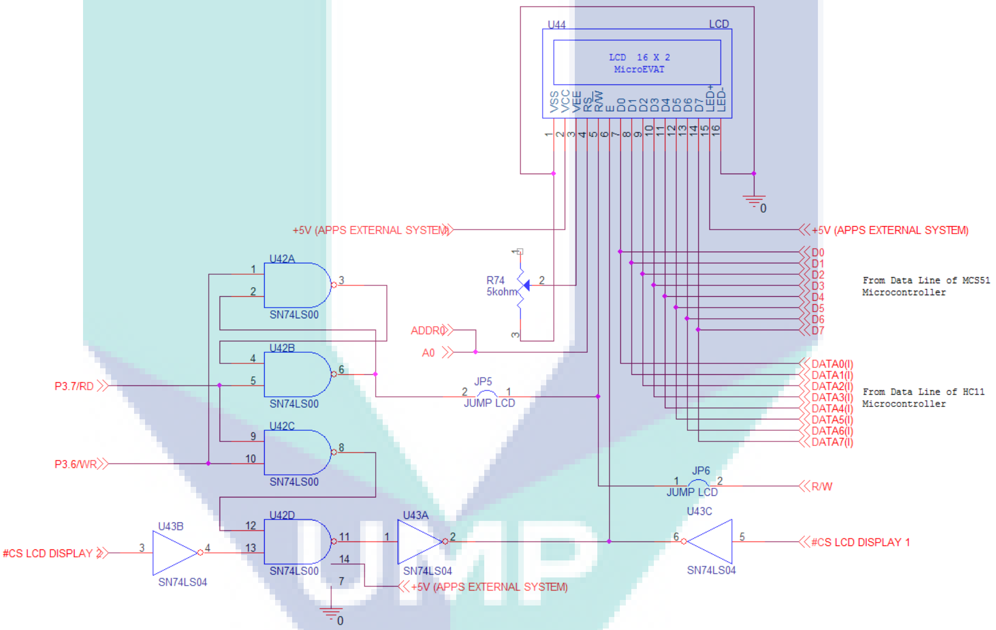


Figure 3.24: The schematics diagram of LCD in expandable interface

The connection of LCD can be done in non-expandable interface. However, this module only been designed using Microchip PIC microcontroller only. The reason for not implemented in either Freescale HC11 or MCS51 microcontroller is the limited memory space using internal memory of both devices. Figure 3.25 shows the schematics diagram of LCD in bootstrap mode using Microchip PIC.

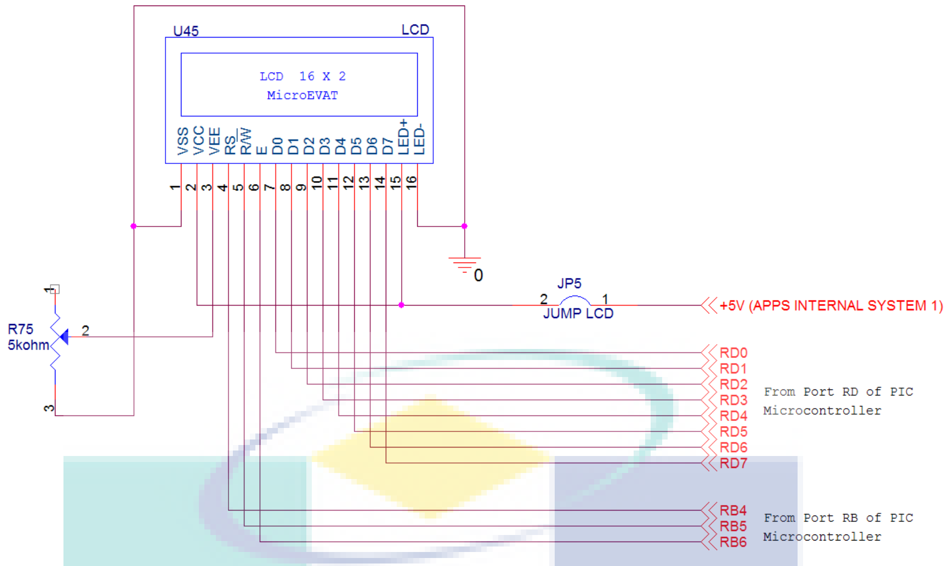


Figure 3.25: The schematics diagram of LCD using Microchip PIC

3.4.8 Graphic LCD Module

The GLCD has a built-in Samsung's KS0108 controller which performs all of the refreshing and data storage tasks of the GLCD display. The display is split logically in half. It contains two controllers with controller 1 (Chip select 1, CS1) controlling the left half of the display and controller 2 (Chip select 2, CS2) controlling the right half. Each controller must be addressed independently. The page addresses, 0-7, specify one of the 8 horizontal pages which are 8 bits (1 byte) high. A drawing of the display and how it is mapped to the refresh memory is shown in Figure 3.26.

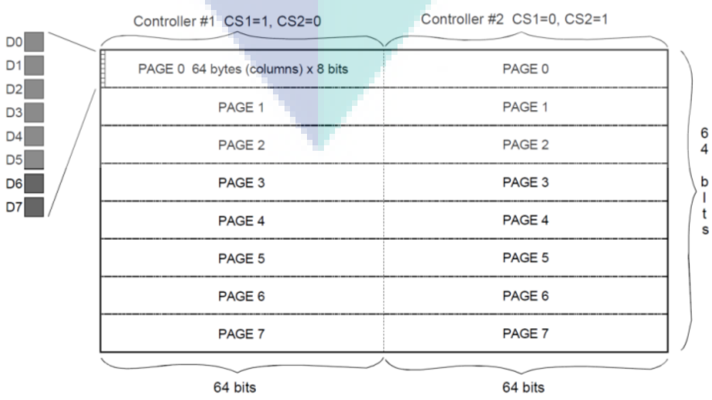


Figure 3.26: Memory map of GLCD 128 X 64

The schematics diagram of circuit connection for interface the GLCD with the Microchip PIC is depicted in Figure 3.27. Like any other LCDs, initialization process is required to set up the GLCD module. Data or instruction register selection (RS) pin is connected to port RB4. The role of this pin is similar to the command mode of the 16 X 2 LCD. In similar manner, pin R/W* is connected to port RB5. The output CS1 and CS2 are for the drivers and are connected directly to port RB2 and RB3 of Microchip PIC. Both enable, E and reset, RST pin are connected to port RB6 and RB7 respectively. The former is required to be activated when a new character or image need to be displayed.

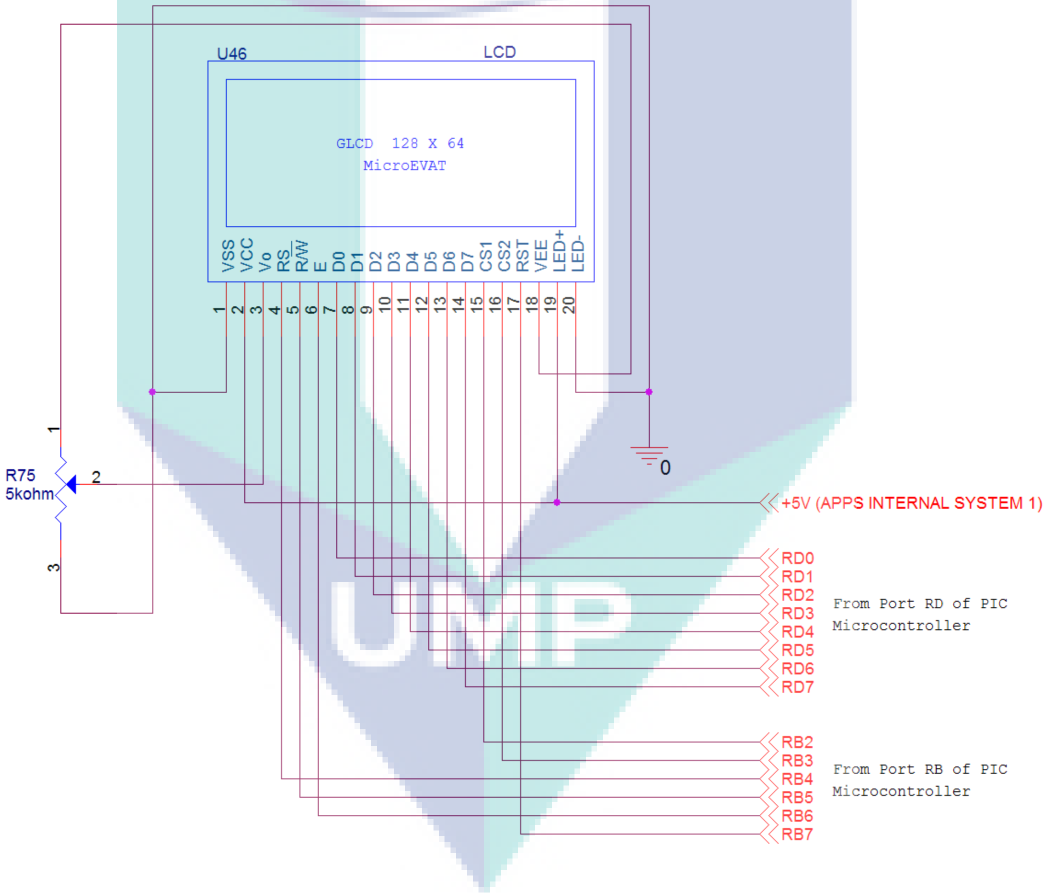


Figure 3.27: The schematics diagram of GLCD using Microchip PIC

3.4.9 Temperature Sensor Module

The LM35 series are precision integrated-circuit temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. Figure 3.28 shows the schematics of LM35 Temperature Sensor circuit connection in non-expandable interface. This module can only implement in Freescale HC11 and Microchip PIC microcontroller because they have the internal Analog-to-Digital Converter (ADC). On the other hand, MCS51 requires external ADC of ADC0804 in MicroEVAT application module to perform the conversion. The output voltage from temperature sensor varies proportionally to the change of temperature. It is connected directly to Port E (PE3) of Freescale HC11 microcontroller that provides ADC function.

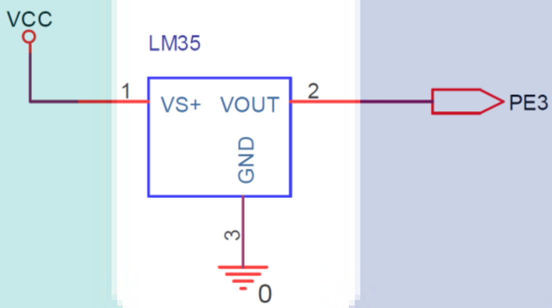


Figure 3.28: The schematics of LM35 Temperature Sensor using Freescale HC11

3.5 MEMORY MAP OF MICROEVAT

The memory map for the MicroEVAT system is simplified in Table 3.4, 3.5, and 3.6 that represent for HC11, Intel MCS51 and PIC18F microcontroller respectively. It consists of several application modules with memory allocation in this system. The memory map tabulates the contents of each memory in each microcontroller. Some of them have their own internal memory that can be used to store temporary data or even a program. There are a few locations reserved for special purpose. One of them is used to store vector address for interrupt routine. This will be explained in later chapter. Another one is reserved for internal register.

Table 3.4: Memory Map of MicroEVAT system (MC68HC11E1)

Memory Range or Port Register	Component Allocation
$0000 - $01FF	Internal Ram 512 Bytes
$1000 - $103F	Register Block 64 Bytes
Port A	Stepper Motor, DC Motor
Port B	Bar Graph, 7 Segment Display
Port C	DIP Switch, Keypad
Port D	Serial Communication Interface (SCI)
Port E	Temperature Sensor
$6000 - $61FF	Bar Graph
$6200 - $63FF	7 Segment Display
$6400 - $65FF	16 Segment Display (LSB)
$6600 - $67FF	16 Segment Display (MSB)
$6800 - $69FF	Dot Matrix (Column)
$6A00 - $6BFF	Dot Matrix (Row)
$6C00 - $6DFF	Traffic Light
$8000 - $81FF	DIP Switch
$8200 - $83FF	Keypad
$8400 - $85FF	Stepper Motor
$8600 - $87FF	DC Motor
$8800	LCD Display Setting
$8801	LCD Display Character
$B600 - $B7FF	Internal EPROM 512 Bytes Bootstrap Mode
$BF00 - $BFBF	Boot ROM BUFFALO Monitor Program
$BFC0 - $BFFF	Special Mode Interrupt Vectors
$C000 - $DFFF	External RAM 8k Bytes
$E000 - $FFFF	External EEPROM 8k Bytes Expended Mode
$FFC0 - $FFFF	Normal Mode Interrupt Vectors

Table 3.5: Memory Map of MicroEVAT system (P89C51RD2)

Memory Range or Port Register	**Component Allocation**
0000H - 1FFFH	External EEPROM 8k Bytes Expended Mode
0000H - 007FH	Lower 128 Bytes Internal RAM
0080H - 00FFH	Upper 128 Bytes Internal RAM Special Function Register
Port 0	DIP Switch, Keypad
Port 1	Stepper Motor, DC Motor
Port 2	Bar Graph, 7 Segment Display
Port 3	Serial Control (SCON)
0100H - 02FFH	Expanded RAM 512 Bytes
2000H - 21FFH	Bar Graph
2200H - 23FFH	7 Segment Display
2400H - 25FFH	16 Segment Display (LSB)
2600H - 27FFH	16 Segment Display (MSB)
2800H - 29FFH	Dot Matrix (Column)
2A00H - 2BFFH	Dot Matrix (Row)
2C00H - 2DFFH	Traffic Light
4000H - 41FFH	DIP Switch
4200H - 43FFH	Keypad
4400H - 45FFH	Stepper Motor
4600H - 47FFH	DC Motor
4A00H	LCD Display Setting
4A01H	LCD Display Character
8000H - 9FFFH	External RAM 8k Bytes
FC00H - FFFFH	Boot ROM

Table 3.6: Memory Map of MicroEVAT system (PIC18F452)

Memory Range or Port Register	Component Allocation
000000H - 00007FH	RAM Access 128 Bytes
000080H - 0005FFH	General Purpose Register 1408 Bytes
000F80H - 000FFFH	Special Function Register 128 Bytes
Port RA	Temperature Sensor
Port RB	Bar Graph, 7 Segment Display, LCD Control Bus, GLCD Control Bus
Port RC	DIP Switch, Keypad, USART
Port RD	Stepper Motor, DC Motor, LCD Data Bus, GLCD Data Bus
Port RE	Parallel Slave Port
001000H - 007FFFH	On-Chip Program Memory 28k Bytes
008000H - 1FFFFFH	Program Memory Space 'Read 0'

3.6 SUMMARY

This chapter has presented the design and implementation of multiple microcontroller hardware platforms in terms of hardware design. The hardware construction of MicroEVAT system has been described in detail in this chapter. In order to allow the usage of internal and external memory storage, the mode selector switching circuit is designed to enable the user to select the mode of application in MicroEVAT system. The serial communication interface is designed and explained in detail in this chapter. In this case, the user could use the MicroEVAT system in various form of serial interface. The memory interface and addressing module implementation allows the limitless memory and application integration for user without any limitation while exploring this system. The implementation of proposed hardware has been designed and the tests for each system module will be explained in Chapter 5.

CHAPTER 4

SOFTWARE DESIGN

This chapter discusses the architecture of MicroEVAT with emphasize on implementation of the software design. The use of this software design is to create a user-friendly environment and to enable user to issue several command such as to upload and execute the program code through the serial communication to microcontrollers.

4.1 INTRODUCTION

The software implementation in MicroEVAT system can be divided into 4 major categories; the software tools, the program code for MicroEVAT module, the operating system for MicroEVAT and Integrated Development Environment (IDE). The software tools that is used to program in the program code into the microcontroller or external memory devices. The implementation of program code for MicroEVAT module to ensure the microcontroller can interface with the application devices based on testing program that has been constructed.

The development of operating system is vital in order to ensure the MicroEVAT system can communicate with the PC without the use of external programming or development tools. On the other hand, the IDE module integrates several interfacing, which will be used in a computer terminal based GUI development. These entire four element plays the important role in order to implement the RATNA Terminal, which the created computer terminal GUI based software for the MicroEVAT prototype system.

4.2 SOFTWARE TOOLS

Software tools are used to implement the task between hardware and software. These tools of software are selected based on the microcontroller products and application devices that are used during this project. During the implementation of MicroEVAT's microcontrollers based application development, several different software tools are used. Fortunately, one-time programming tool is used to program the mini operating system (machine code) and bootloader in such a way so that the users are no longer require these tools. However these software tools are only be used once for programming the monitor programs and bootloaders that has been created by the author into microcontrollers or external memories. As mention in Chapter 1 in contribution section, the main goal of this MicroEVAT system is to be used without of using these external software tools.

4.2.1 X-CTU

The X-CTU software is a Windows-based application provided by Digi. This program is designed to interact with the firmware files found on Digi's RF products and to provide a simple-to-use graphical user interface to them. The screen shot of X-CTU software with XBEE PRO 4214A modem communication test is shown in Figure 4.1.

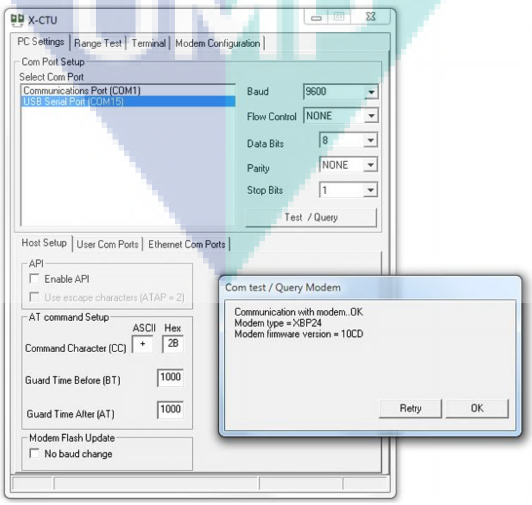


Figure 4.1: The screen shot of X-CTU software with XBEE PRO 4214A modem communication test

The X-CTU is designed with all Windows-based computers running Microsoft Windows 98 SE and compatible. The purpose of using this X-CTU software is to configure the XBEE modem for communication setup. Before using this program, an update on selected modems are required to avoid an error on firmware devices. Two XBEE modems are used for communicating the MicroEVAT system with the PC. One modem will be used on MicroEVAT board and the other one will be used with the PC through the USB port. Figure 4.2 shows the MicroEVAT prototype system with XBEE PRO 4214A modems.

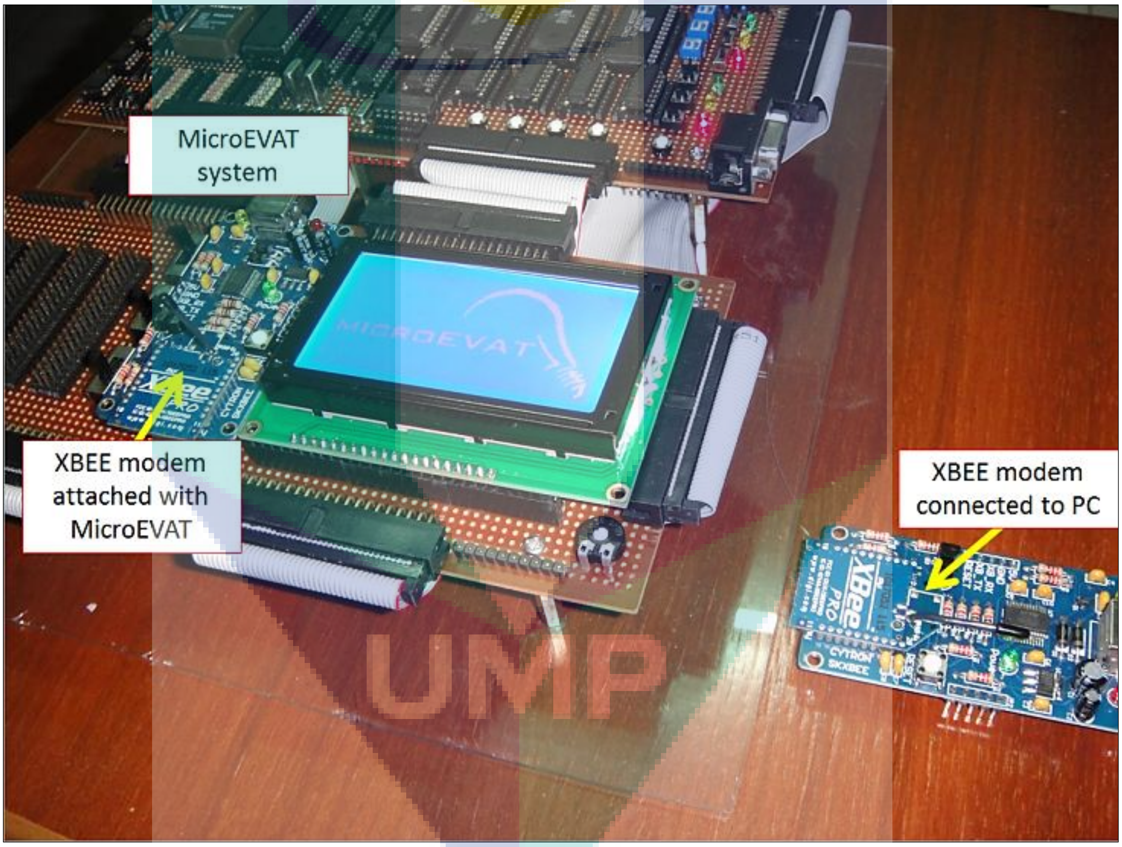


Figure 4.2: The MicroEVAT prototype system with XBEE PRO 4214A modems

Based on Figure 4.2, the XBEE modems must be configured its Destination Address (DL) and Source Address in order to form the communication link between the two modems. This procedure is to avoid the communication error and to ensure the data sent operation secured as the encryption and decryption are performed initially. The screen shot of Modem Configuration is shown in Figure 4.3.

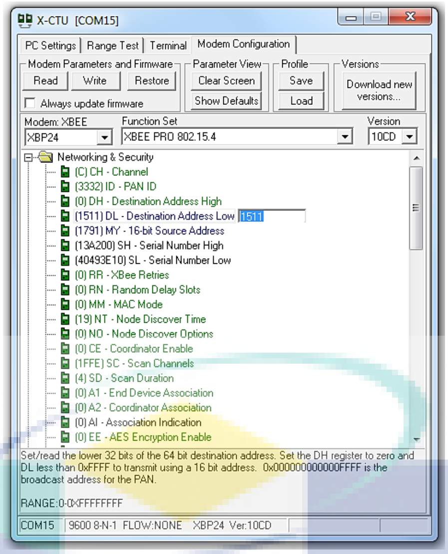


Figure 4.3: The screen shot of Modem Configuration

The configuration between two XBEE modems is performed by setting up the similar address for both Source Address (MY) of XBEE modem on MicroEVAT system and Destination Address (DL) of XBEE modem connected to PC. Likewise, the similar process is conducted on Destination Address (DL) of XBEE modem on MicroEVAT system and Source Address (MY) of XBEE modem. These addresses are assigned to the MicroEVAT system in communication interface as shown in Table 4.1.

Table 4.1: Source and Destination Address of MicroEVAT's XBEE modems

XBEE Modem	Source Address (MY)	Destination Address (DL)
MicroEVAT	1511	1791
PC	1791	1511
2nd PC or more*	0000	1511

* If more than a PC is used, this method has to be taken for creating a Mesh Network in order to have access from many PCs. A default Source Address (MY) of 0000 is used to indicate the selected XBEE modem have capabilities of receiving the data from MicroEVAT system and any other XBEE modems. However, the Destination Address (DL) must be set to 1511 in order to transmit the data to MicroEVAT system only, not to any other XBEE modems.

4.3 PROGRAM CODE FOR MICROEVAT MODULE

In the development of program code plays on important role in the MicroEVAT system as the user can request several commands to perform its job. In addition, some program codes are provided in testing the functionality of several I/O devices.

In order to produce on effective and high performance system, two types of program language are written; assembly language and high-level language. For all microcontrollers in MicroEVAT, the assembly language is chosen to develop the mini-operating system. Evidently, the Universiti Malaysia Pahang, particularly the FKEE department has retained the assembly language programming requirement in ECT courses.

Assembly language provides the most natural and direct access to the processor's capabilities. By introducing assembly language programming in such an environment, the user can see the relevance of this language while gaining an appreciation for the details of program execution at microcontroller architecture level. On the other hand, the high-level language is preferred for Motorola PIC microcontroller as the language widely used in short courses and final year project.

In several basic assembler and compiler, we required to convert the source code into binary file or most commonly known as the machine code. These machine codes will be programmed into the memory storage. Several procedures must be performed in converting the assembly language or the high level language into binary file or the machine code. Figure 4.4 shows the flow chart of the source code implementation in application module.

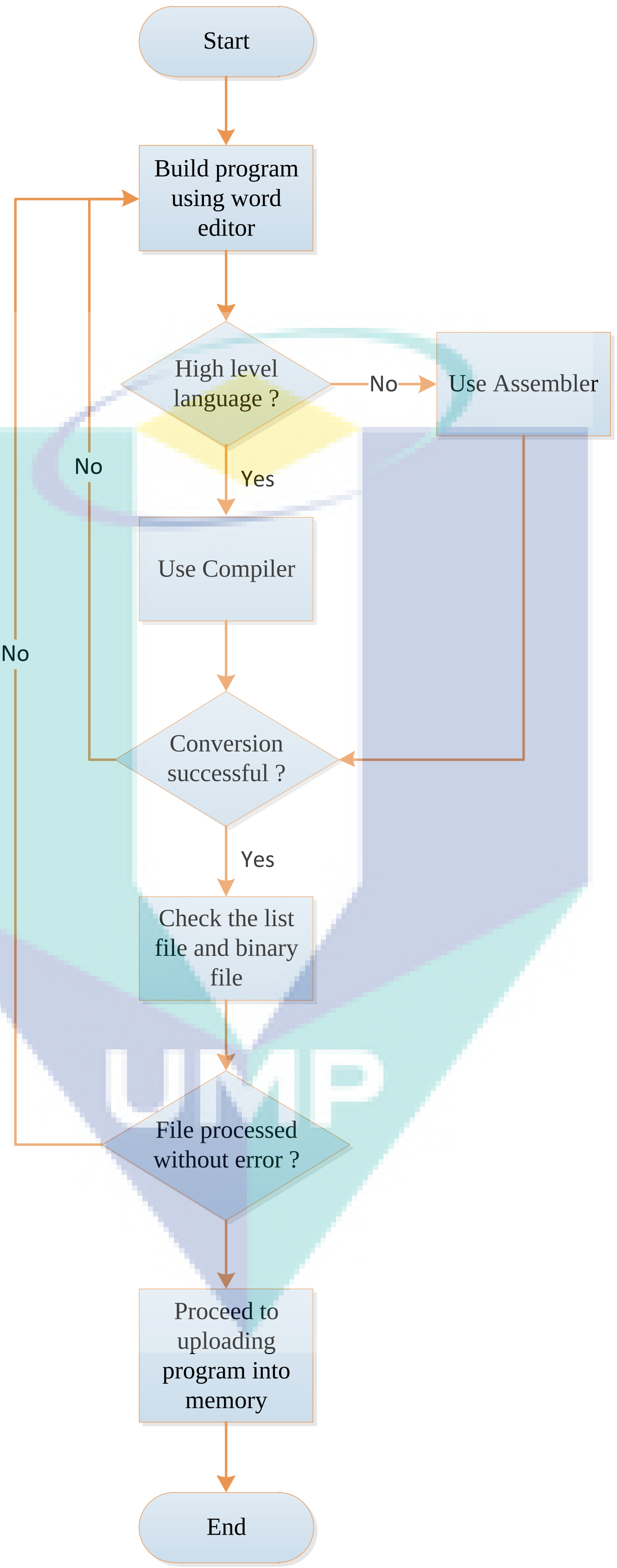


Figure 4.4: The flow chart of the source code implementation

4.3.1 Serial Interface

The communication between the PC and MicroEVAT can be develop using Universal Asynchronous Receiver Transmitter (UART), one of two independent serial input/output (I/O) subsystems in the series of microcontrollers. It has a standard non-return-to-zero (NRZ) format (one start bit, eight or nine data bits, and one stop bit). Several baud rates are available, however, only one default baud rate is chosen in this system; which is 9600.

Typically, the serial data format requires an idle line in the high state before transmission or reception of a message. A start bit, logic '0', transmitted or received, that indicates the start of each character and the data generally is transmitted and received at least significant bit (LSB) first. A stop bit, logic '1', used to indicate the end of a frame. A frame consists of a start bit, a character of eight or nine data bits, and a stop bit. A break, defined as the transmission or reception of logic '0' for some multiple number of frames.

In the Freescale HC11 microcontroller, initializing of the Serial Communication Interface (SCI) register involves five addressable registers. Four of them are control and status registers which are the Serial communications control register 1 (SCCR1), Serial communications control register 2 (SCCR2), Baud rate register (BAUD), and Serial communications status register (SCSR). One data register which is the Serial communications data register (SCDR). The SCCR1 register provides the control bits that determine word length and select the method used for the wakeup feature and the SCCR2 register provides the control bits that enable or disable individual SCI functions. The SCSR provides inputs to the interrupt logic circuits for generation of the SCI system interrupt. The BAUD allows the user to select different baud rates for the SCI system. The SCDR is a parallel register that performs two functions of the receive data register when it is read and the transmit data register when it is written. Normally, this registers is written once during initialization.

Figure 4.5 shows the steps of initializing the SCI in Freescale HC11 microcontroller as well as the calculation of the BAUD register value in this system.

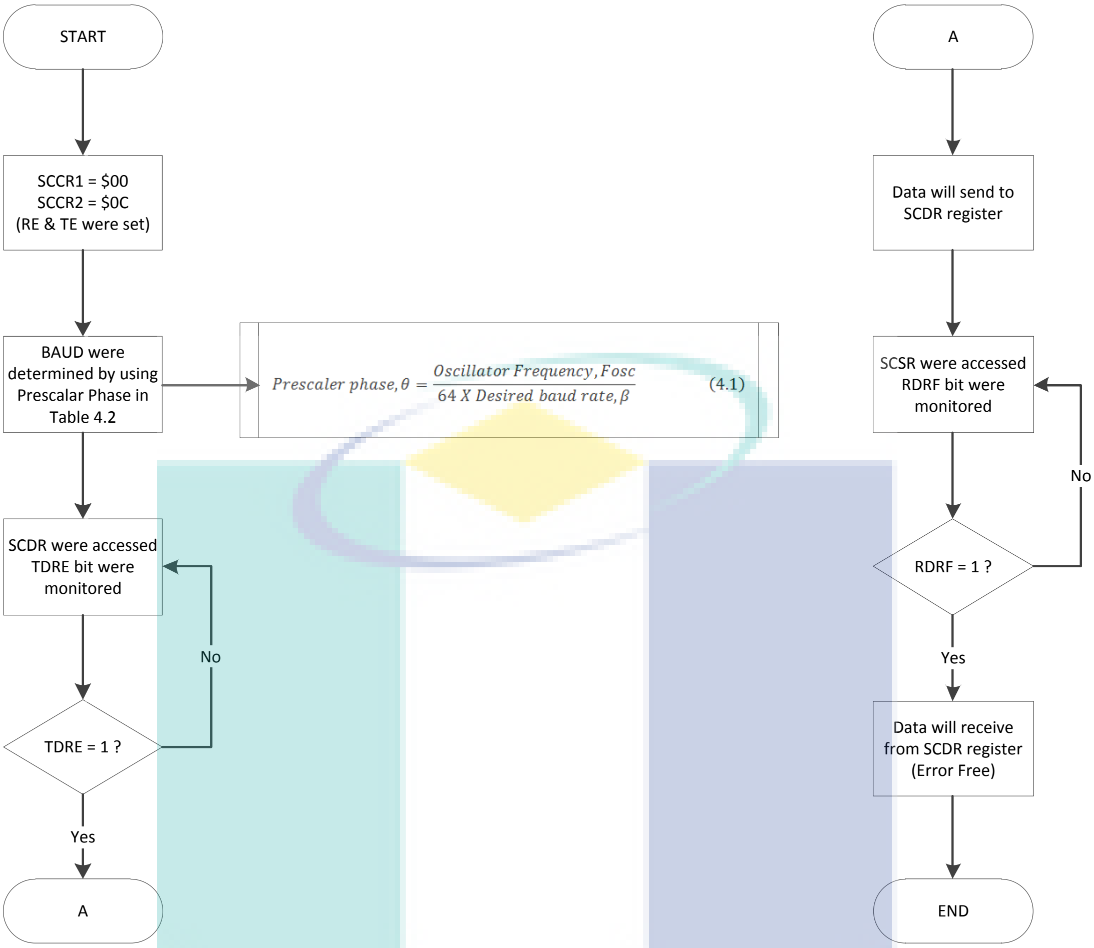


Figure 4.5: The flow chart of SCI Initialization in Freescale HC11

Table 4.2: Baud Rate Divide Value for Freescale HC11

Prescaler Divide	**Prescaler Selects**			**Baud Rate Divide**	**Prescaler Selects**		
	SCP2	SCP1	SCP0		SCR2	SCR1	SCR0
1	0	0	0	**1**	0	0	0
				2	0	0	1
3	0	0	1	**4**	0	1	0
				8	0	1	1
4	0	1	0	**16**	1	0	0
				32	1	0	1
13	0	1	1	**64**	1	1	0
				128	1	1	1

For MCS 51 microcontroller, the serial communication receive and transmit registers are both accessed at Special Function Register, Serial Data Buffer (SBUF). Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register. The serial communication of MCS51 microcontroller can operate in four modes, Serial Mode 0, Serial Mode 1, Serial Mode 2, and Serial Mode 3. Table 4.3 shows the relation between SCON register and Serial Mode function.

Table 4.3: The relation between SCON register and Serial Mode function

SCON Register		**Function**
SM0 (bit 7)	**SM1 (bit6)**	
0	0	Serial Mode 0; 8-bit data, non-variable baud rate
0	1	Serial Mode 1; 8-bit data, 1 stop and start bit, variable baud rate
1	0	Serial Mode 2; 8-bit data, 1 stop and start bit, programmable 9th data bit, programmable baud rate
1	1	Serial Mode 3; 8-bit data, 1 stop and start bit, programmable 9th data bit, variable baud rate

In all the modes that available in this MCS51 microcontroller, Mode 1 is chosen as it is compatible with IBM PC and allow the baud rate to be variable. Figure 4.6 shows the steps for initializing the serial communication in MCS51 microcontroller as well as the calculation of the baud rate in Timer High 1 (TH1) register value in this system.

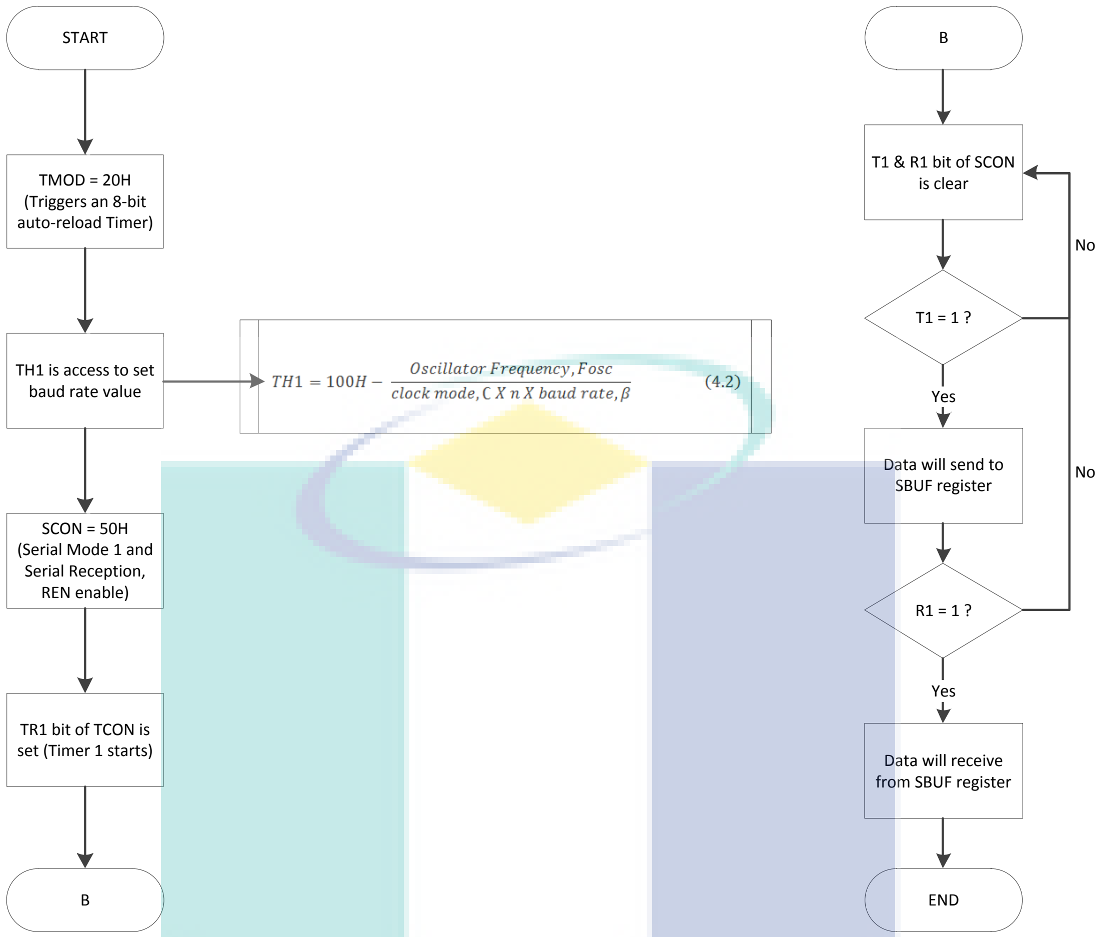


Figure 4.6: The flow chart of Serial communication Initialization in MCS51 microcontroller

For Microchip PIC microcontroller, the Universal Synchronous Asynchronous Receiver Transmitter (USART) can be configured as a full duplex asynchronous system that can communicate with peripheral devices, such as CRT terminals and personal computers, or it can be configured as a half-duplex synchronous system that can communicate with peripheral devices, such as A/D or D/A integrated circuits, and serial EEPROMs. The USART Asynchronous Mode is used and in this mode, where the USART is used standard non-return-to-zero (NRZ) format.

The most common data format is 8-bits. An on-chip dedicated 8-bit baud rate generator can be used to derive standard baud rate frequencies from the oscillator. The USART transmits and receives the LSB bit first. The USART's transmitter and receiver are functionally independent, but use the same data format and baud rate. The baud rate generator produces a clock, either 16 or 64 of the bit shift rate, depending on BRGH bit in Transmit Status and Control Register (TXSTA).

Parity is not supported by the hardware, but can be implemented in software. Asynchronous mode is paused during SLEEP mode. Asynchronous mode is selected by clearing SYNC bit in TXSTA. The USART Asynchronous module consists of these elements; Baud Rate Generator, Sampling Circuit, Asynchronous Transmitter and Asynchronous Receiver. For an Asynchronous Transmission and Reception initializing process begins as shown in Figure 4.7.

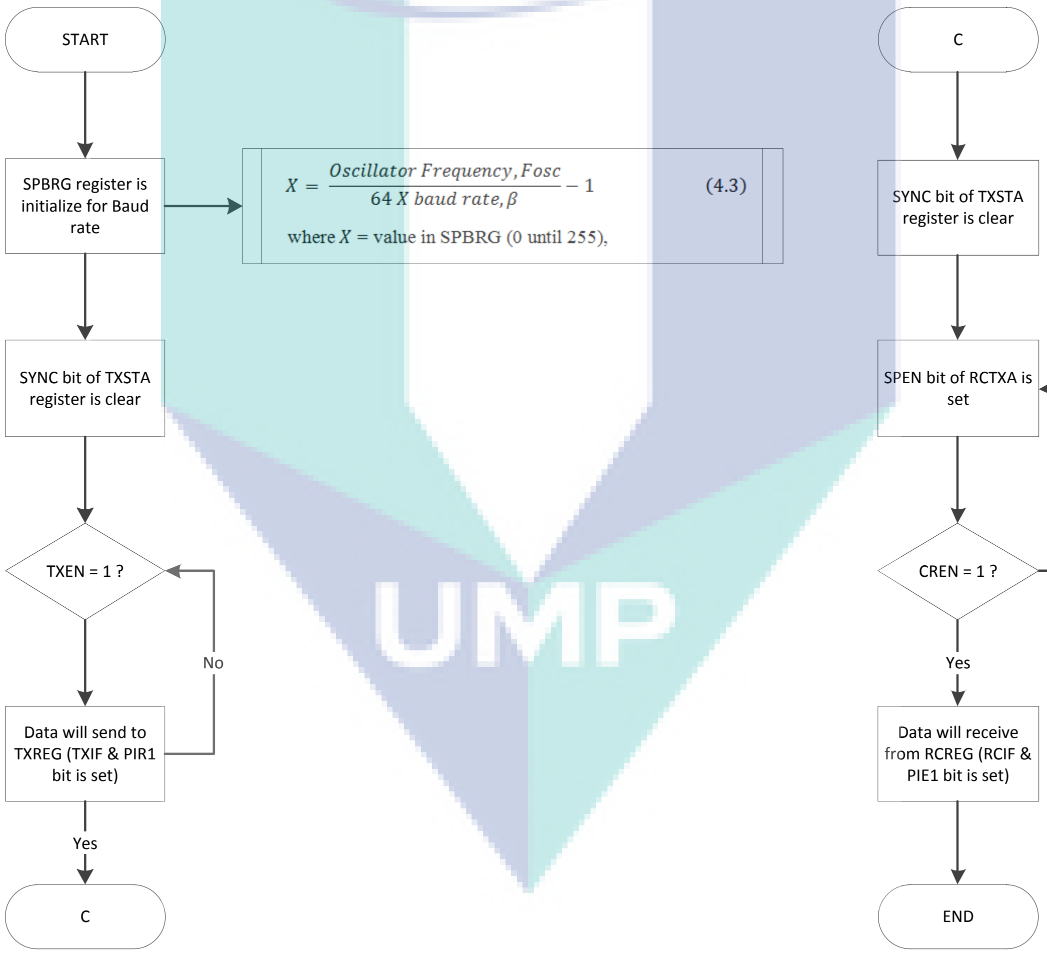


Figure 4.7: The flow chart of Asynchronous Serial Transmission and Reception initialization in Microchip PIC microcontroller

4.3.2 Memory Interface

As mention previously, the MicroEVAT system used external and internal memory space to store these codes that consists of several modules for microcontroller's operation. In the system, a method of uploading the program into RAM and executing the program must be developed along the designing of the operating system. This must be included as uploading and executing the program inside the microcontroller's operation. This section will be briefly explained on the development of program subroutine of uploading and executing program code for MicroEVAT system.

In one of the important routine is uploading program module. However, to develop this module, each microcontroller has its own operating code format. For example, Freescale HC11 microcontroller uses Motorola S Record. The SREC format file consists of a series of ASCII records. All hexadecimal numbers are Big Endian formats. The records have the following structure:

- **Start code**, one character, an S.
- **Record type**, one digit, 0 to 9, defining the type of the data field.
- **Byte count**, two hex digits, indicates the number of bytes (hex digit pairs) that follow in the rest of the record (in the address, data and checksum fields).
- **Address**, four, six, or eight hex digits as determined by the record type for the memory location of the first data byte. The address bytes are arranged in big endian format.
- **Data**, a sequence of 2n hex digits, for n bytes of the data.
- **Checksum**, two hex digits - the LSB of ones' complement of the sum of the values represented by the two hex digit pairs for the byte count, address and data fields. For example:

S1137AF00A0A0D0000000000000000000000000000061

13+7A+F0+0A+0A+0D+00+00+00+00+00+00+00+00+00+00+00+00+00
= 19E,

Checksum can be obtained by using the one's complement of least significant byte which in this case byte (9E) equals with byte 61. Figure 4.8 shows the example of SREC file after assembling the program code for Freescale HC11 microcontroller.

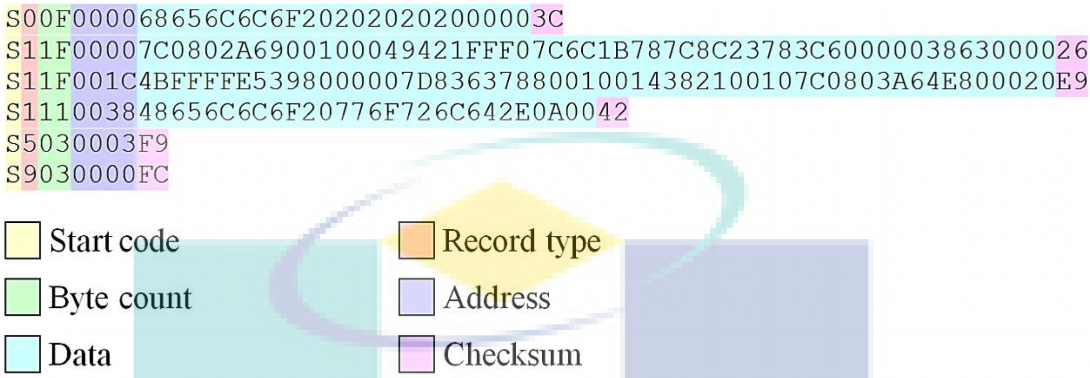


Figure 4.8: The example of SREC file formats

Based on Figure 4.8, the uploading process is designed and developed by using all the elements that available on SREC file. Each of the elements could shows the characteristic of SREC file and these advantages can be implemented in the next task of this research project methodology. Figure 4.9 shows the flow chart of program subroutine for uploading S19 file in Freescale HC11 microcontroller.

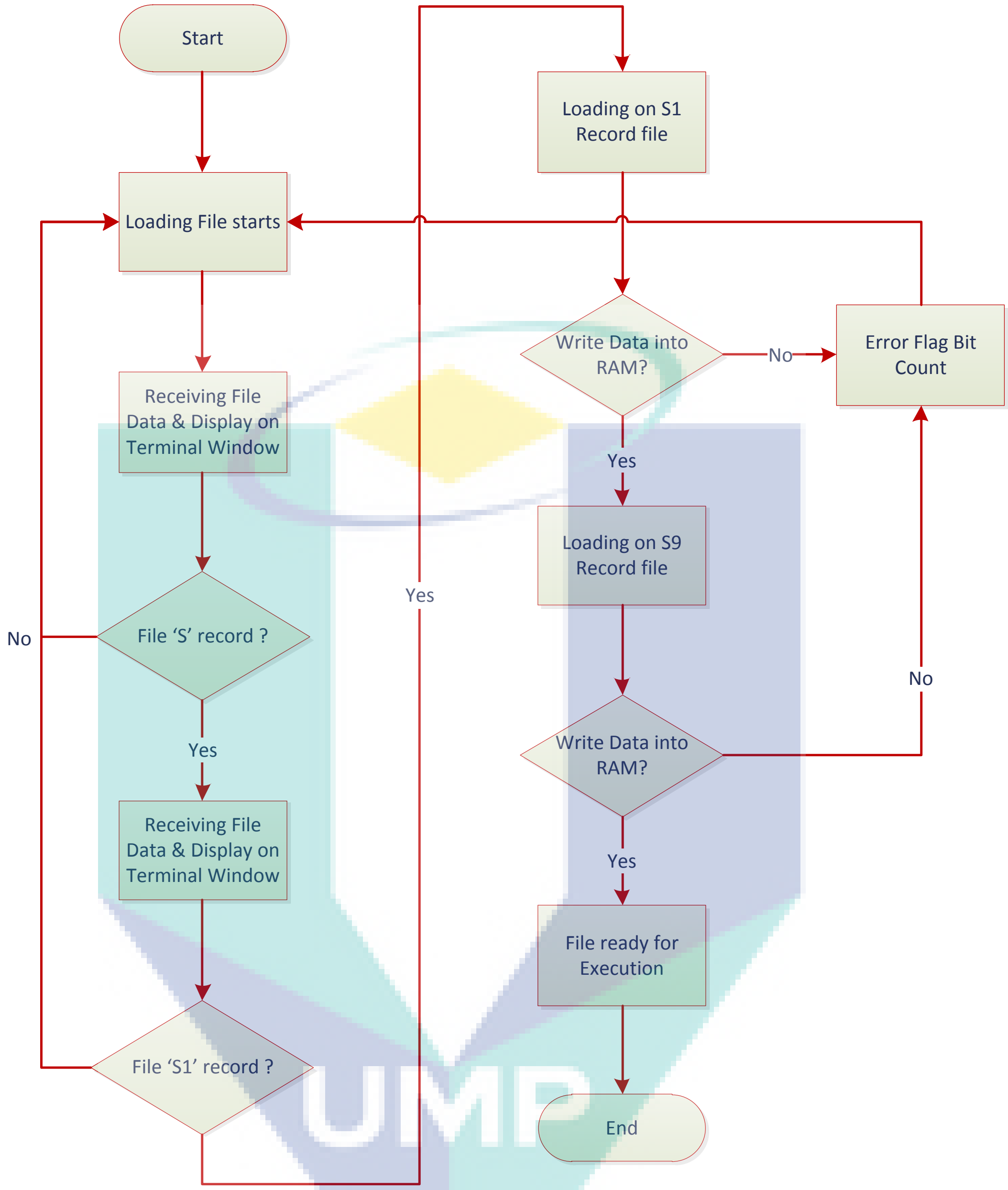


Figure 4.9: The flow chart of uploading program subroutine for Freescale HC11

According to the uploading steps using the program subroutine that has been designed as the flow chart, the file of SREC will be checked for character 'S', indicates the 'S Record' file will be uploaded. The second step will be the checking of 'S1 Record file' that indicates the 8-bit microcontroller associates. This will lead to the next techniques where the SREC file will be uploaded into RAM. The file that has been uploaded into RAM considered an error free SREC file, the file is ready for execution process.

In the process of program execution, the program that has been uploaded will be executed by request the starting address of the program uploaded inside RAM. Based on Figure 4.9, the program will started from the location of address $C000 of RAM. Figure 4.10 shows the flow chart of the executing program subroutine in Freescale HC11 microcontroller.

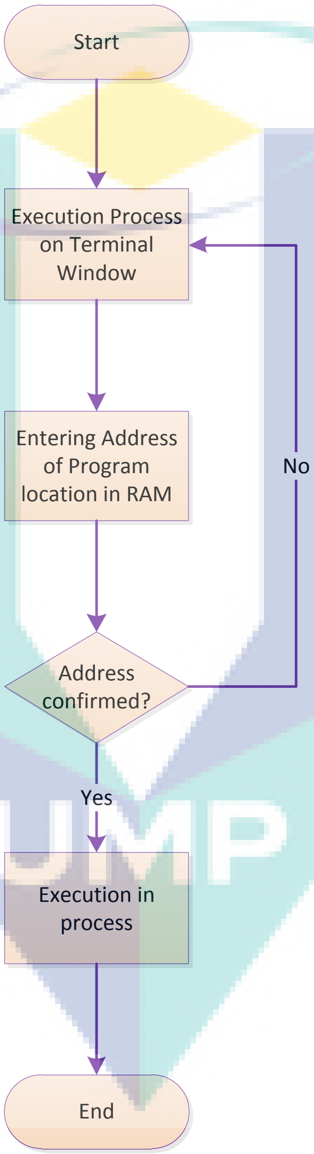


Figure 4.10: The flow chart of executing subroutine for Freescale HC11

Based on Figure 4.10, the program file that stored inside RAM will be addressed by using Address Indexing Mode in Freescale HC11 register. The address entered will be stored inside Index X register, for executing the file that referred by this register.

Unlike Freescale HC11, for MCS51 and Microchip PIC microcontrollers does not applied the same principal as the assembled file is in format of Intel Hex or known as I8HEX file. The extension file in .hex and will be uploaded into memory space of the microcontroller. The format is a text file, with each line containing hexadecimal values encoding a sequence of data and their starting offset or absolute address. Each line of Intel HEX file consists of six parts:

- **Start code**, one character, an ASCII colon ':'.
- **Byte count**, two hex digits, a number of bytes (hex digit pairs) in the data field. 16 (10H) or 32 (20H) bytes of data are the usual compromise values between line length and address overhead.
- **Address**, four hex digits, a 16-bit address of the beginning of the memory position for the data. Limited to 64 kB, the limit is worked around by specifying higher bits via additional record types. This address is big endian.
- **Record type**, two hex digits, 00H to 05H, defining the type of the data field.
- **Data**, a sequence of n bytes of the data themselves, represented by 2n hex digits.
- **Checksum**, two hex digits - the LSB of the two's complement of the sum of the values of all fields except fields 1 and 6 (Start code ':' byte and two hex digits of the Checksum).

It is calculated by adding together the hex-encoded bytes (hex digit pairs), then leaving only the least significant byte of the result, and making a two's complement (either by subtracting the byte from 0x100, or inverting it by implementation of XOR gating concept with 0FFH and adding 01H). If with 16-bit and more variables, the techniques must suppress the overflow by AND gating concept the result with 0FFH. The overflow may occur since both 100H - 0H and (00H $\oplus$ 0FFH) + 1H equal 100H. If the checksum is correctly calculated, adding all the bytes (the Byte count, both bytes in Address, the Record type, each Data byte and the Checksum) together will always result in a value wherein the LSB is zero (00H). For example,

:10010000214601360121470136007EFE09D2190140

10+01+00+00+21+46+01+36+01+21+47+01+36+00+7E+FE+09+D2+19+01 = 3C0, Checksum can be obtained by using the two's complement of least significant byte which in this case byte (C0) equals with byte 40. Figure 4.11 shows the example of I8HEX file for either MCS51 or Microchip PIC microcontroller.

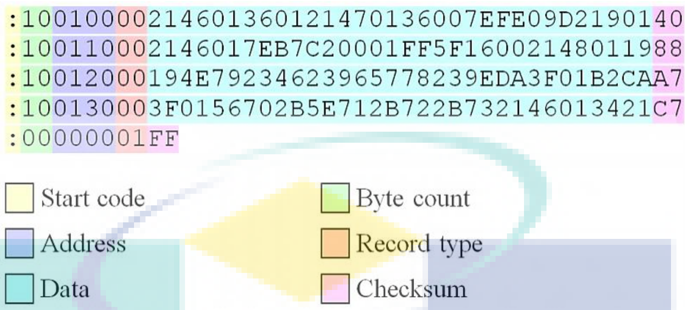


Figure 4.11: The example of I8HEX file formats

By referring the I8HEX file, the program file uploading process of this file into RAM address location can be done according to the concept of validation in I8HEX file's elements. The arrangement of the I8HEX file is different than the SREC file in terms of the location for Byte count, Address, and Record type. The Byte count and Address will appear before the Record type. Figure 4.12 shows the flow chart of uploading subroutine for both MCS51 and Microchip PIC microcontroller.

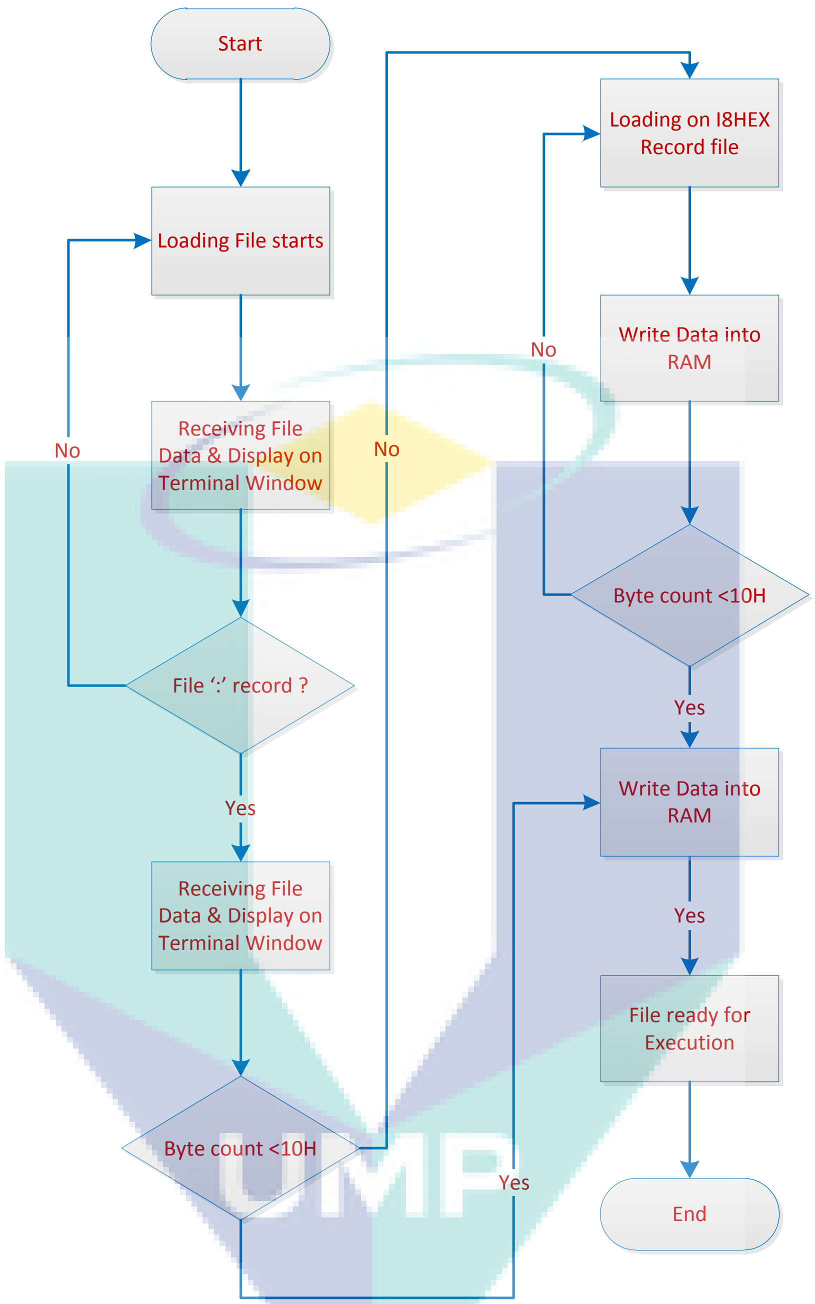


Figure 4.12: The flow chart of uploading subroutine for both MCS51 and Microchip PIC microcontroller

In this case I8HEX does not reflect of having error flag bit count. This is because the byte count of I8HEX file does not exist more than the 16 bytes, so that ignoring the error occurs since the serial communication has synchronization glitch when the more bytes sent in a period of time, this program subroutine will be designed without this module of programming.

Like program subroutine for execution in Freescale HC11, MCS51 and Microchip PIC microcontroller undergo the similar process, except for the address of execution code is placed in Data Pointer register. Based on Figure 4.10, the address for RAM will be loaded in Data Pointer (DPTR) for two bytes, High (DPH) and Low (DPL). The DPTR will show the address location and index addressing mode will be implemented. However, this technique of program code execution process does not applied for Microchip PIC microcontroller since the loaded program will be stored in FLASH memory of the microcontroller. The program will be executed once the power is consumed and will be erased once the new programs are loaded into the same location. Since that, the necessary of creating this executing program subroutine for Microchip PIC microcontroller is not necessary.

4.4 IDE INTERFACE – RATNA TERMINAL

An integrated development environment (IDE) also known as integrated development environment, integrated debugging environment or interactive development environment. This is an approach for software application that provides comprehensive facilities to computer programmers for software design. An IDE normally consists of:

i. a source code editor,
ii. a compiler or an interpreter,
iii. build automation tools, and
iv. a debugger.

Sometimes a version control system and various tools are integrated to simplify the construction of a Graphical User Interface (GUI). Many modern IDEs also have a class browser, an object inspector, and a class hierarchy diagram, for use with object-oriented software design. However the implementation of IDE interface in MicroEVAT, the elements that have been mention above does not fully involved in this architecture of designed RATNA Terminal, a designed GUI based IDE approach software that developed for MicroEVAT system.

RATNA Terminal is the developed software for MicroEVAT system where, the user can communicate these systems using the proposed serial communication interfaces and used for uploading and execute the desired program code with the provided application module or any other user-designed application interface. This system is developed by using Microsoft Visual Studio 2010 (VS2010) as the platform for the entire recent developed GUI using Windows Presentation Foundation (WPF) or known as the Microsoft Silverlight.

The reason of creating the RATNA Terminal using VS2010 platform is because the previous version of Visual Studio does not have the latest version of .Net Framework. The use of the current version of .Net Framework 4 (.NF4) helps to improve support for parallel computing, which target multi-core or distributed systems, which will be used in current computer system like Intel Core system processor. If the RATNA Terminal integrates with the current version of windows OS, these supports of .NF4 must be implemented in order to install the RATNA Terminal. Figure 4.13 shows the main screen of RATNA Terminal software.

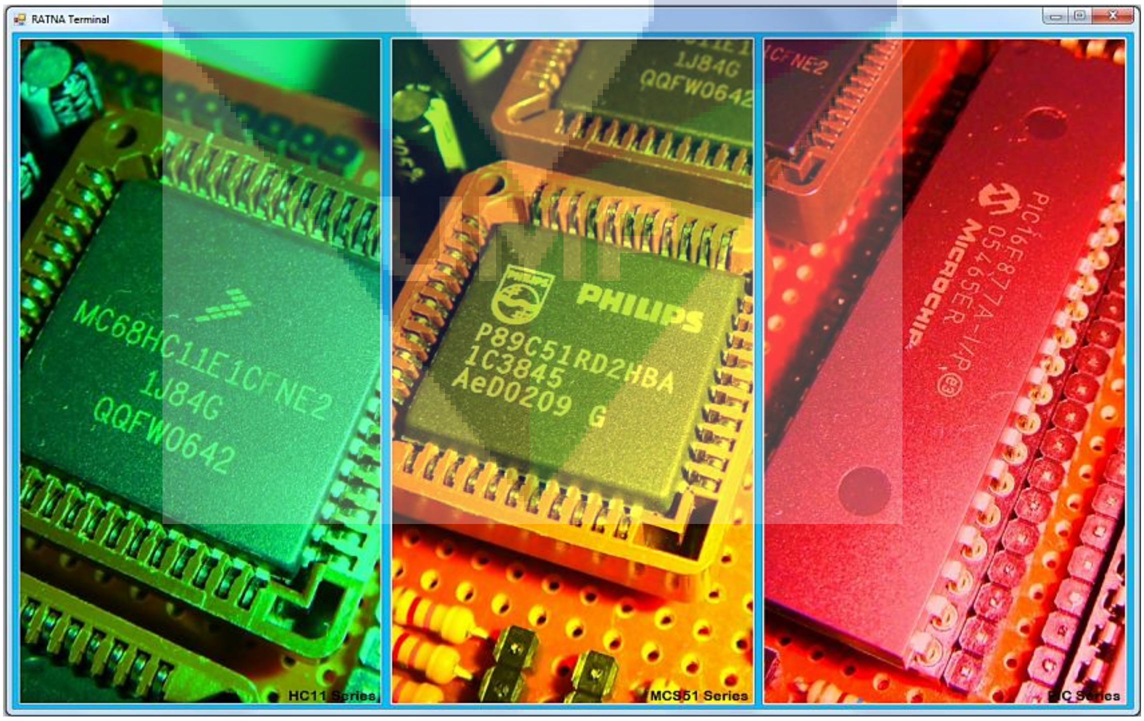


Figure 4.13: The screen shot of RATNA Terminal main screen

4.4.1 Main Screen

Based on Figure 4.13, the RATNA Terminal has divided between three subroutines of IDE programs. This include for Freescale HC11, MCS51, and Microchip PIC software subroutine. For each subroutine, the Microsoft Serial Communication (MSComm) subroutine program is developed in order for the MicroEVAT system to use this RATNA Terminal as window terminal software. According to Figure 4.1, the configuration settings of the X-CTU software are used to implement this subroutine program and the rest of the microcontroller software subroutines have this MSComm configurations. However, this MicroEVAT system has only one baud rate configuration since the crystal value and the value of serial communication interface are not varied. The MSComm is configured with the number of data bits of 8, stop bits of one, and the parity and handshaking ports are configured with none. This is written in this subroutine program of MSComm. The selection of serial port can be selected from the available active ports. The MSComm subroutine program is shown in Figure 4.14.

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    If serialPort.IsOpen Then
        serialPort.Close()

    End If
    Try
        With serialPort
            .PortName = ComboBox1.Text
            .BaudRate = 9600
            .Parity = IO.Ports.Parity.None
            .DataBits = 8
            .StopBits = IO.Ports.StopBits.One
            .Handshake = IO.Ports.Handshake.None
            .Encoding = System.Text.Encoding.ASCII
            .Encoding = System.Text.Encoding.UTF32
            .Encoding = System.Text.Encoding.UTF8
            .Encoding = System.Text.Encoding.BigEndianUnicode
            .Encoding = System.Text.Encoding.Default

        End With
        serialPort.Open()

        TextBox4.Text = ComboBox1.Text & " Connected."
        Button1.Enabled = False
        Button2.Enabled = True
        ConnectToolStripMenuItem.Enabled = False
        DisconnectToolStripMenuItem.Enabled = True
        Button8.Enabled = True
        Button9.Enabled = True
        AddressingLayoutToolStripMenuItem.Enabled = True
        OperationToolStripMenuItem.Enabled = True
        ApplicationToolStripMenuItem.Enabled = True
        TextBox1.Enabled = True

    Catch ex As Exception
        MsgBox("Please select COM Port before connecting", MsgBoxStyle.OkOnly, _
            Me.Text)
    End Try
End Sub
```

Figure 4.14: The subroutine program of MSComm

4.4.2 Microcontroller Selection Subroutine

RATNA Terminal has been divided into three program subroutine. This method is to ensure the synchronism of all three microcontrollers whenever they communicate with this GUI based IDE approach software. For Freescale HC11 and MCS51 microcontroller, the designed GUI is same for these microcontrollers interface. For Microchip PIC microcontroller, software called Tiny PIC Bootloader is used to communicate with SCI of Microchip PIC microcontroller.

After the developed monitor program installed in microcontroller memory space, it can receives a user program from the PC and writes it in the RAM, then launches this program in execution. Same as the monitor program, the bootloader itself must be written into the flash memory with an external burner. Figure 4.15, 4.16 and 4.17 shows the main screen of each microcontroller interfaces. More screen shot on this terminal is included in Appendix B.

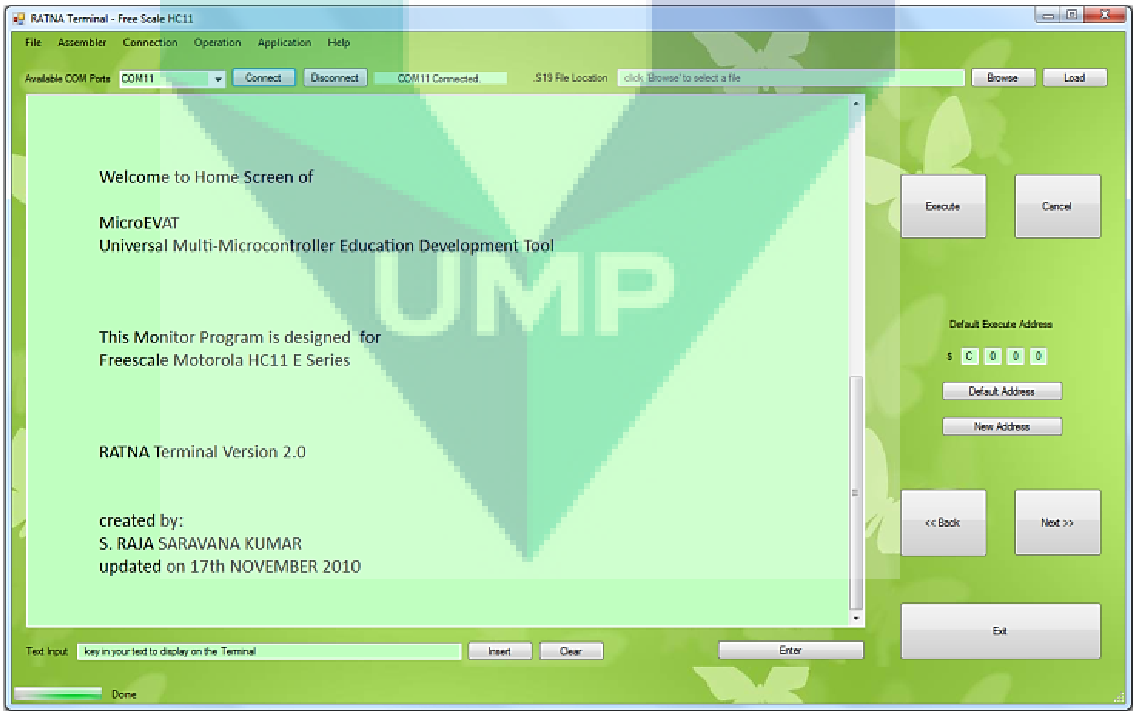


Figure 4.15: The screen shot of RATNA Terminal for Freescale HC11

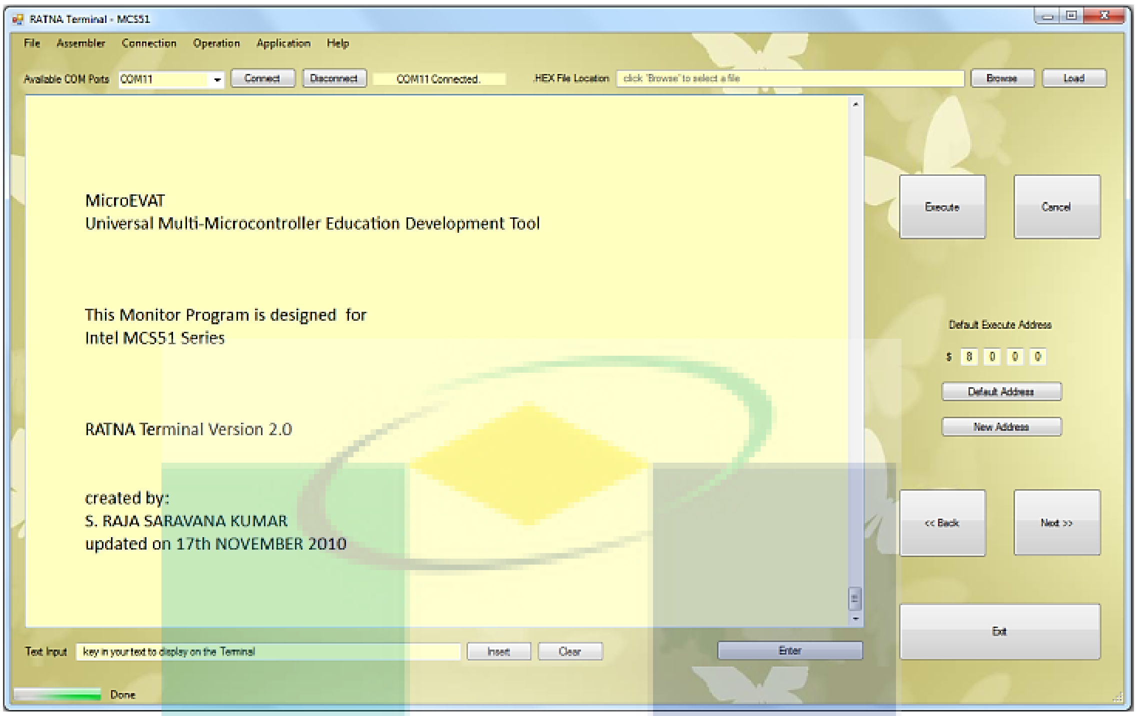


Figure 4.16: The screen shot of RATNA Terminal for MCS51

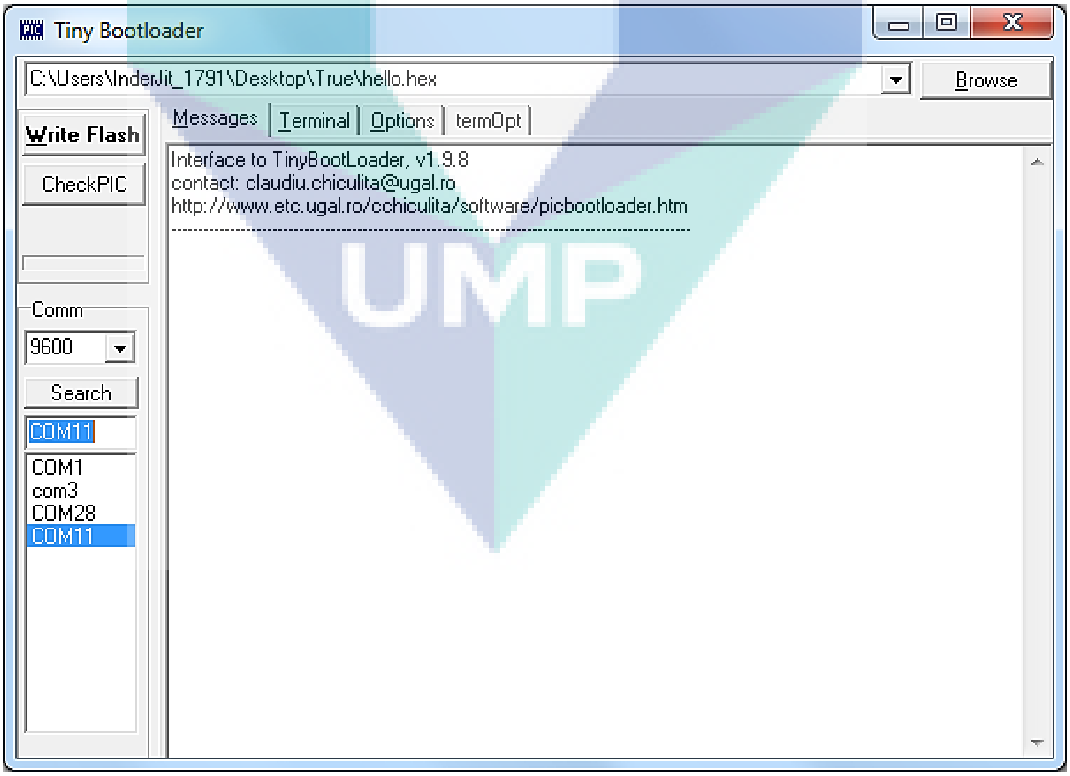
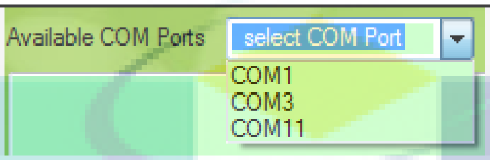
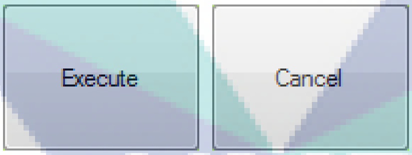
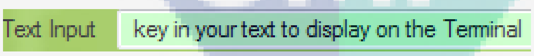
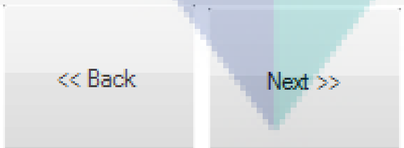
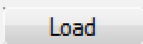


Figure 4.17: The screen shot of RATNA Terminal using Tiny PIC Bootloader for Microchip PIC

The software as shown in Figure 4.15 and 4.16 is divided to various subroutines with each performs specific task. Some of the main subroutines and their functions are shown in the Tables 4.4.

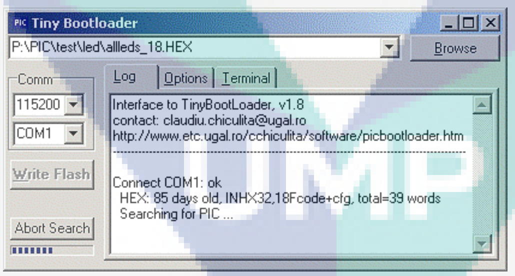
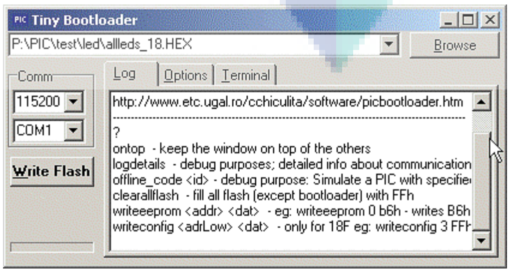
Table 4.4: The function of each subroutine for Freescale HC11 and MCS51

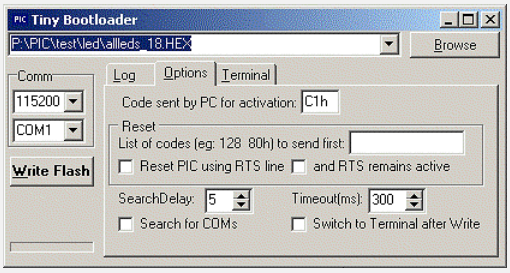
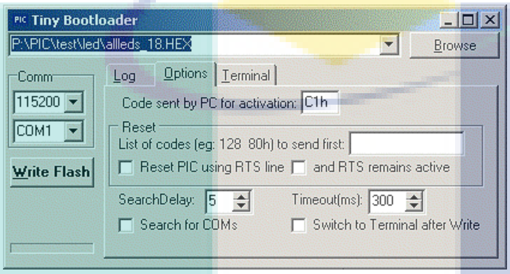
Subroutine	Icon	Functionality
Serial Port selection		To select the comm port for MicroEVAT. Available port only listed here.
Browse section		To load the binary file into MicroEVAT's RAM. Browse button to enable the file to be uploaded
Execution address		By default, the system will be showing the address location RAM. The user can choose the desired location that the program to be executed.
Execution manual button		User can choose the previous link that able to execute the program manually by clicking this button. 'CANCEL' button to disable the action
Text Input		Key-in the desired instruction or address for viewing. For memory modification, the address range can be entered.
Navigation buttons		Used for viewing the current and previous of address range for memory modification.
Load button		Used to load the binary file by manually.
Enter button		Used to load the instruction from the text input navigation.
Exit button		Exit the system.

4.4.3 Tiny PIC Bootloader

Tiny PIC Bootloader is a firmware program that designed for Microchip PIC microcontroller. Basically, a bootloader is a program that stays in the microcontroller and communicates with the PC (usually through the serial interface). The bootloader receives a user program from the PC and writes it in the flash memory, then launches this program in execution. Bootloader can only be used with those microcontrollers that can write their EEPROM through software. The bootloader itself must be written into the flash memory with an external burner. In order for the bootloader to be launched after each reset, a "goto bootloader" instruction must exist somewhere in the first 4 instructions; it is the job of the bootloader to reallocate the first 4 instructions of the user program to another location and execute them when the bootloader exits. The following Table 4.5 shows the function that available in Tiny PIC Bootloader.

Table 4.5: The function of subroutine for Tiny PIC Bootloader

Subroutine	Icon	Functionality
Write Flash		Connects to the specified port with the chosen baud rate Reads and analyses the hex file, determines if it's a 16F or 18F code, determine the size, EEPROM and configuration data.
Commands		Command line parameter: hex filename to be written. When using a command line parameter writing flash will start automatically.

<table>
<tr>
<td>Options</td>
<td>
</td>
<td>"SearchDelay" is the amount of time the application will try to contact the pic.
"Timeout" is the ping interval; also it acts as a timeout for all serial operations. If the user expects large communication delays, should be increasing this value.</td>
</tr>
<tr>
<td>Terminal</td>
<td>
</td>
<td>A rudimentary Terminal, with the following possibilities of:

Display received data as text, hex dump or character codes.
Automatically saves raw data to the specified file.
It can send:

ASCII string
one by one characters as the user type
decimal or hex codes

It has a limited (adjustable) display buffer (default 10kB)
</td>
</tr>
</table>

4.5 SUMMARY

This chapter has presented the design and implementation of multiple microcontroller hardware platforms in terms of software design. The implementation of software part that integrates with the MicroEVAT system, a GUI based IDE approach RATNA Terminal is done using all the elements that suite in this design. This chapter covers the architecture of software design where the designing begins in the development of program subroutine of each function that will be inserted in the MicroEVAT's operating system. After all the required elements are designed including the construction of application devices program testing, serial interface subroutine, and the memory modification subroutine, the OS are developed for each microcontroller in MicroEVAT system. This will lead to the graphical integration software for user-friendly purposes. Finally, as mention above, the designing of RATNA Terminal using IDE approach had been done using Visual Studio 2010 and complete environment software has been designed. The implementation of proposed software method for MicroEVAT system has been designed and the testing of the system for each module will be explained in Chapter 5.

CHAPTER 5

RESULTS AND TESTING

The chapter presents the description of various testing that have been performed on this research. Each module will be tested individually before they are integrated into the system and the results of this testing reveals a substantial achievement.

5.1 INTRODUCTION

Various tests are conducted in this project. The results of the testing reveal the system have achieved of substantial goal. The system that consists of main module and application module will be tested independently. Later, the modules are integrated and tested for its functionality. The MicroEVAT system has integrated with three independent microcontroller devices that connected into one serial interface for communication purposes.

As shown in Figure 5.1, the MicroEVAT system has been setup with the communication interface with the PC using all the available links for experimental purpose which will be used for reveal the testing results. The system board is tested as the core of the MicroEVAT system. The preliminary testing consists of power supply and the communication interfaces. The subsequent testing covers on the application module interface and the mini operating system with GUI testing flow. Various testing are performed in order to ensure the objective and the scope of the research project that are fully succeeded.

Figure 5.1: The experimental setup for MicroEVAT prototype system

5.2 SYSTEM BOARD TESTING

The core of the MicroEVAT system project is the system board module. Thus, various tests are conducted in the system to ensure its functionality. The initial test involves a free running test where sample circuit in Figure 5.2 and 5.3 are used. Then the program of Bar Graph free running test is written for these entire three microcontrollers as shown in Figure 5.4, 5.5, and 5.6. When the program is executed, a free running pattern is provided. It concludes that the microcontroller system is fully functioning where the communication link between microcontroller and the application device are executes properly.

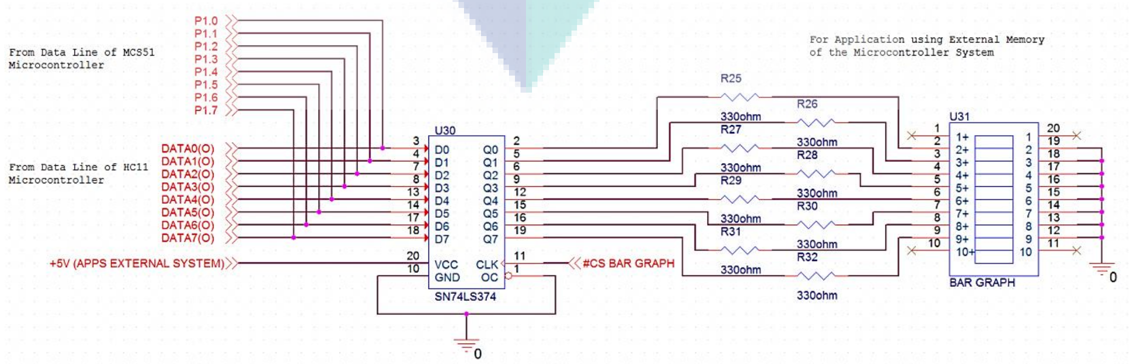


Figure 5.2: The schematics diagram of Bar Graph free running test in expended mode

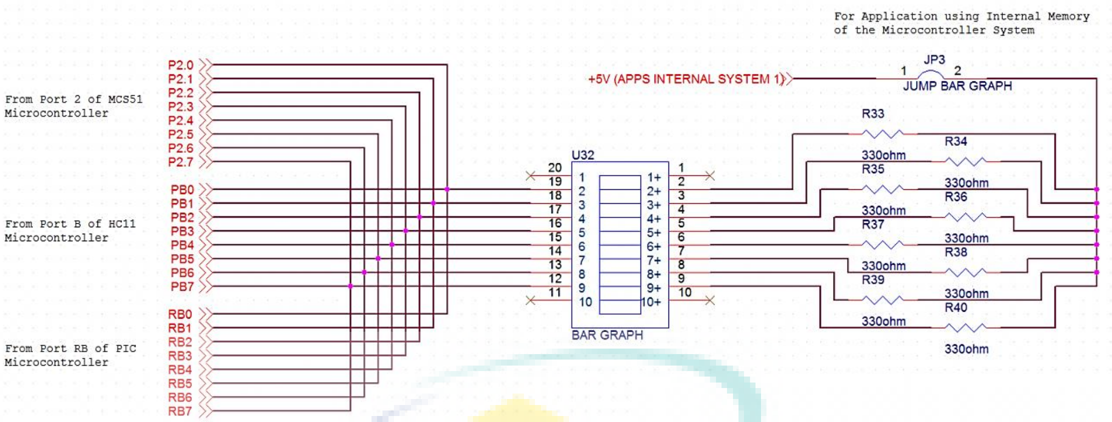


Figure 5.3: The schematics diagram of Bar Graph free running test in bootstrap mode

```
         ORG     $C000
BGRAPH   EQU     $6000
         LDS     #$FF

COLLECT  LDX     #DATA1
         LDAA    #0
         STAA    BGRAPH
         BSR     DELAY

LOOP_CL  LDAA    0,X
         STAA    BGRAPH
         BSR     DELAY
         INX
         CPX     #DATA1+23
         BNE     LOOP_CL
         BRA     COLLECT

DELAY    PSHX
         LDAB    #2
LOOP2    LDX     #$FFFF
LOOP1    DEX
         BNE     LOOP1
         DECB
         BNE     LOOP2
         PULX
         RTS

DATA1    FCB     $81,$42,$24,$18,$24,$42,$81,$80,$40,$20,$10
         FCB     $08,$04,$02,$01,$02,$04,$08,$10,$20,$40,$80,0

         END
```

Figure 5.4: The program testing for Freescale HC11 microcontroller

```
        $MOD51
        ORG     8000H
BGRAPH  EQU     2000H
        MOV     SP,#5FH

AGAIN:  CLR     A
        MOV     DPTR,#TABLE
NEXT:   MOVX    A,@DPTR
        PUSH    ACC
        JZ      AGAIN
        INC     DPTR
        MOV     24H,DPH
        MOV     25H,DPL
        MOV     P1,A
        MOV     DPTR,#BGRAPH
        MOVX    @DPTR,A

DELAY:  MOV     R2,#10
INLP1:  MOV     R1,#30
INLP2:  MOV     R0,#255
INLP3:  DJNZ    R0,INLP3
        DJNZ    R1,INLP2
        DJNZ    R2,INLP1
        MOV     DPH,24H
        MOV     DPL,25H
        POP     ACC
        LJMP    NEXT


        ORG     9000H

TABLE:  DB      18H,3CH,7EH,0FFH,7EH,3CH,18H,24H,
        DB      42H,81H,42H,24H,18H,10H,08H,20H,
        DB      04H,40H,02H,80H,01H,80H,02H,40H,
        DB      04H,20H,08H,10H,0

        END
```

Figure 5.5: The program testing for MCS51 microcontroller

```
        ORG     0X0000
                GOTO    INIT
                NOP
                NOP
                NOP

INIT            MOVLW   6
                MOVWF   ADCON1
                CLRF    TRISB
                CLRF    PORTB

START           MOVLW   D'10'
                RCALL   DELAY
                COMF    PORTB
                BRA     START

DELAY           MOVWF   CNT1
DL1             MOVLW   D'255'
                MOVWF   CNT2
DL2             MOVLW   D'255'
                MOVWF   CNT3
DL3             NOP
                DECFSZ  CNT3
                BRA     DL3
                DECFSZ  CNT2
                BRA     DL2
                DECFSZ  CNT1
                BRA     DL1
                RETURN

                END
```

Figure 5.6: The program testing for Microchip PIC microcontroller

5.3 MICROEVAT – PRELIMINARY TESTING

Before the MicroEVAT system tests are performed on the RATNA Terminal, several preliminary tests are conducted to ensure the stability of the MicroEVAT system board that will operate as desired during the test. These are the power Supply test and the serial communication interfaces test.

5.3.1 Power Supply Testing

This test is done to ensure the stability and the durability of power scheme module for MicroEVAT. There are two types of parameter need to consider for building a power supply module, the current load and the heat temperature. For typical LM 7805 voltage regulator devices, the current load must be not exceed 1.0 A at $5V_{DC.}$ If the system or the application module uses more current load than the specified load will resulting increase of heat temperature of the voltage regulator.

A heatsink is attached on voltage regulator to reduce the thermal load. The size of heatsink is determined after the actual thermal resistance produced by the voltage regulator. According to the LM 7805's datasheet, 5˚C/W to 65˚C/W will be produced based on the load resistance that applied for both system and application module. Figure 5.7 and 5.8 shows the schematics diagram of measuring the current load for power supply module using Digital Clamp Multimeter indicated '**A**' in both circuits.

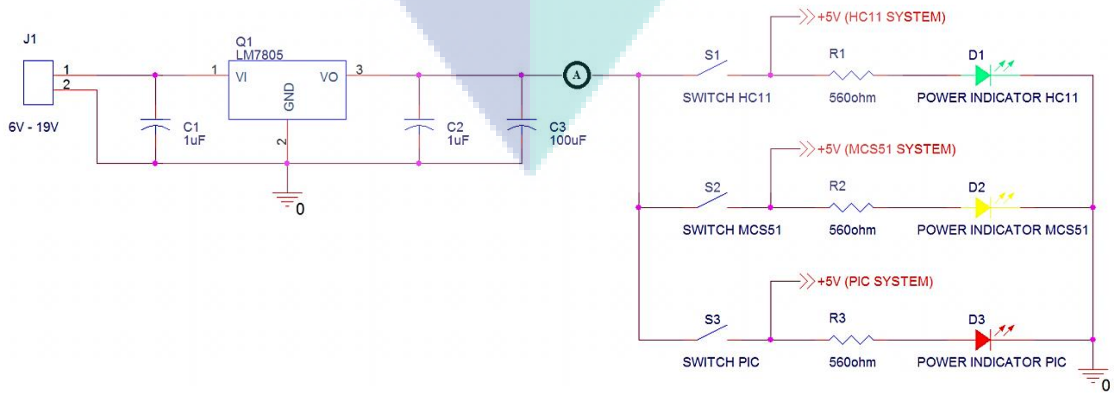


Figure 5.7: The schematics diagram of current load measuring for System board power supply module

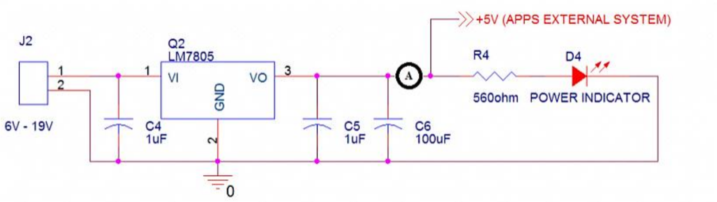


Figure 5.8: The schematics diagram of current load measuring for Application board power supply module

Based on figure 5.7 and 5.8, the experimental setup is conducted as shown on Figure 5.1 to measure the current load and the thermal load for MicroEVAT system. Figure 5.9 shows the diagram of how the current load is measured using UT204 Digital Clamp Multimeter and Figure 5.10 shows the diagram of how the thermal load is measured at room temperature using PK-600T Digital Thermal Multimeter. Resulting from the testing procedures is simplified in Table 5.1.

Figure 5.9: The current load measurements

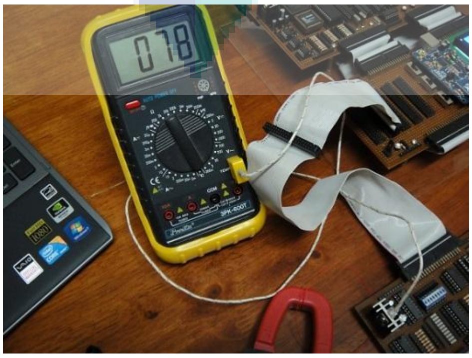


Figure 5.10: The thermal load measurements

Table 5.1: The testing results for current and thermal load for MicroEVAT system

Board	No of module	Load current, mA	Temperature, ˚C*
Offline System	All offline	0.0	29
	Power Supply	6.2	30
	One microcontroller	68.5	36
	2 microcontrollers	117	39
	3 microcontrollers	259	47
Application	One module	98	38
	2 modules	169	42
	3 modules	257	46
	4 modules	419	63
	More than 4 modules	557 - 912	72 - 100

Based on Table 5.1, the MicroEVAT system can operate multiple microcontroller system in one main board since the current and the thermal load does not exceed the nominal value. Likewise, the application module can operates more than 4 modules in one board executed at the same period of time, which will give the value of 912 mA of current. However, the heat temperature rises until 100˚C, and will give the application board a heat transfusion to the application devices. Although it is considered safe to have more than 4 modules in one application board, but if this system produces in a large scale, it is considered as a risk for the user when handling this development board and may get injured from the heat produced by the board. Figure 5.11 shows the line graph on output current with junction temperature at constant output voltage of 5V.

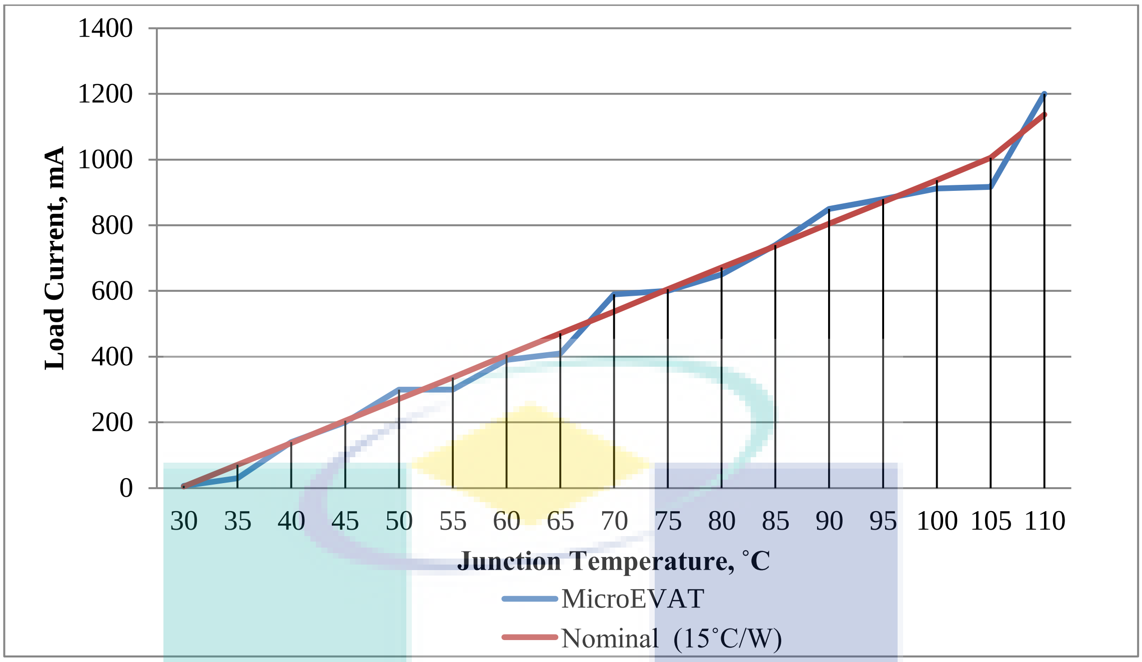


Figure 5.11: Line graph on Load current with Junction Temperature of Voltage regulator at constant output voltage of $5V_{DC}$

Based on Figure 5.11, the thermal resistance on the voltage regulator LM7805 is to be found approximately 15˚C/W based on results that obtain in Table 5.1. The actual nominal value of LM7805 with thermal resistance of 15˚C/W is included in Figure 5.11. The quiescent current of the MicroEVAT system is value at 6.2 mA where closest with the nominal value of 5.0 mA of the voltage regulator. The results are not obtained below the quiescent junction temperature since the MicroEVAT system will only works above the room temperature. Changes in output voltage, Vo due to heating effect can be neglected since the MicroEVAT system can be operated below under temperature of 100˚C. As from the line graph shown in Figure 5.11, the recommendation for the MicroEVAT system is to have the combination of 4 modules in one application board and could have up to three microcontroller in one main system board as the objective to produce multiple microcontroller hardware platform are achieved.

5.3.2 Serial Communication Interfaces Test

For the communication interfaces testing procedure, the monitor programs are developed and the program testing is conducted in previous sections. To test whether the serial communication could hold all three microcontroller devices communicate with the terminal window on PC, the process of uploading and executing a program are conducted. A flowchart of system board serial interfacing with PC is shown in Figure 5.12. After the program is executed, the output of program such as welcoming message and menu selection are displayed on the computer screen. Therefore it can be concluded that the monitor program is able to communicate with the computer.

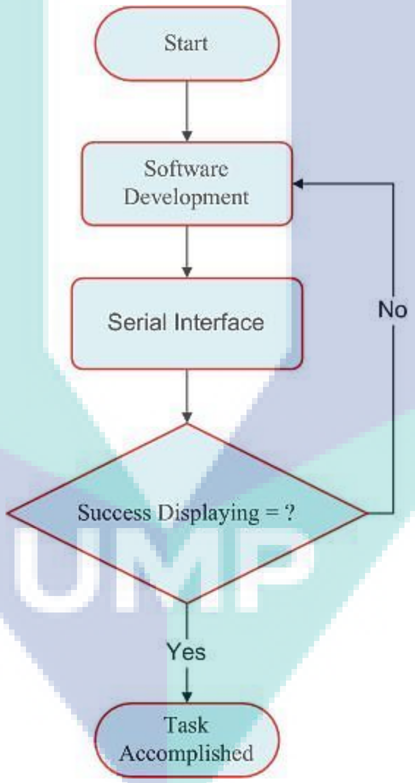


Figure 5.12: The flow chart of serial communication interface for MicroEVAT system

The outcome of the process is shown in Appendix B which been tested using RATNA Terminal and Tiny PIC Bootloader. This conclude that the system board is successful designed as all these three microcontrollers could communicate with the PC. Users can choose the function from the menu provided such as loading, execute, modify or even conduct various testing on the application board.

5.4 MICROEVAT – SUBSEQUENT TESTING

After the preliminary testing has been conducted to ensure the stability of MicroEVAT system board, the subsequent test will be conducted in terms of the software and the application module integrations. This testing procedure is to ensure the connected application devices works well with the designed GUI based IDE approach software for MicroEVAT system.

5.4.1 Application Module Test

Since the application board consists of extensive devices, each of them must be tested individually to ensure its functionality. Figure 5.13 shows some of the example in application module testing. Flowchart of Figure 5.14 is used as a guide line for the testing module. A complete testing program for each module is available in mini-operating system design. The result of the testing reveals that application board is ready for interface purpose.

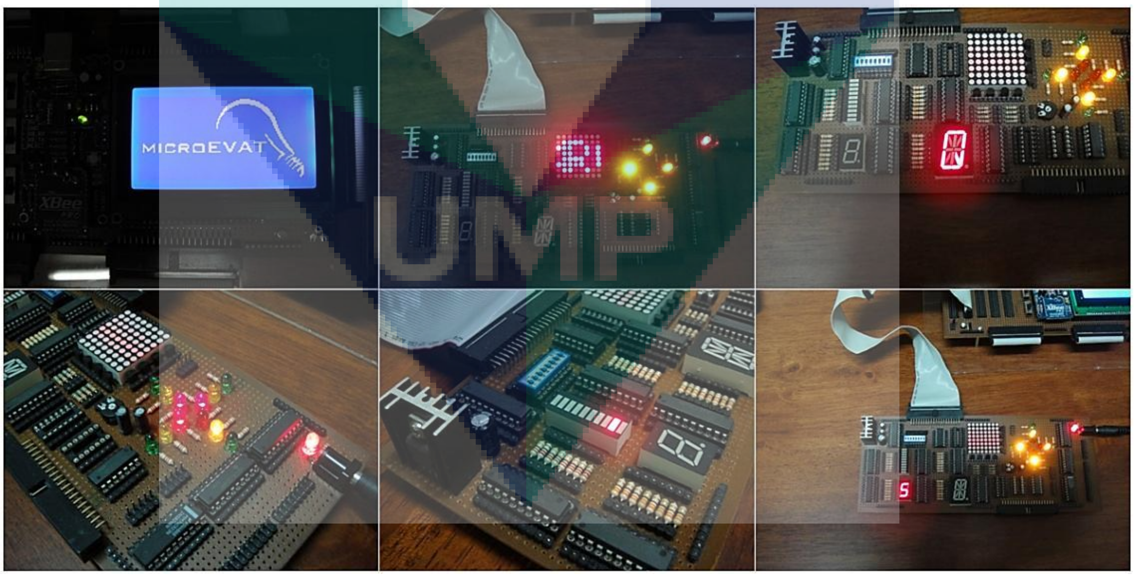


Figure 5.13: The example of testing result for some application module

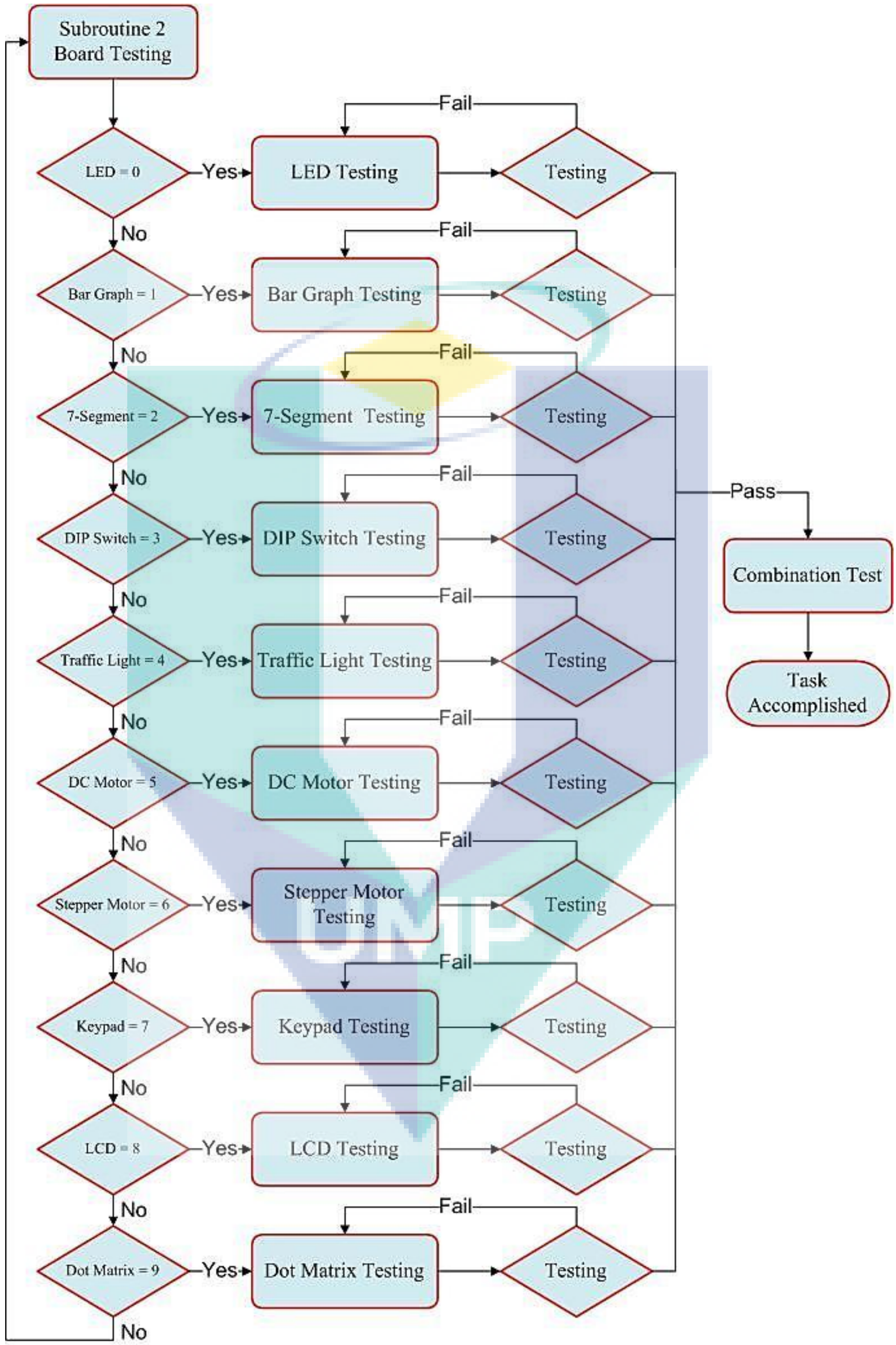


Figure 5.14: The flow chart of application board testing

5.4.2 IDE Interfaces Test

As mentioned in chapter 4, there is GUI based IDE approach type of display mechanism that are provided in the project. The graphical-oriented is developed by using Visual Studio 2010. The VS 2010 is the most powerful programming language that provided standard windows object and graphic user interface that will make the program become user friendly. Figure 5.15 is a flowchart for the graphical-oriented testing. It requires initialization process before it can fully communicate with the development board.

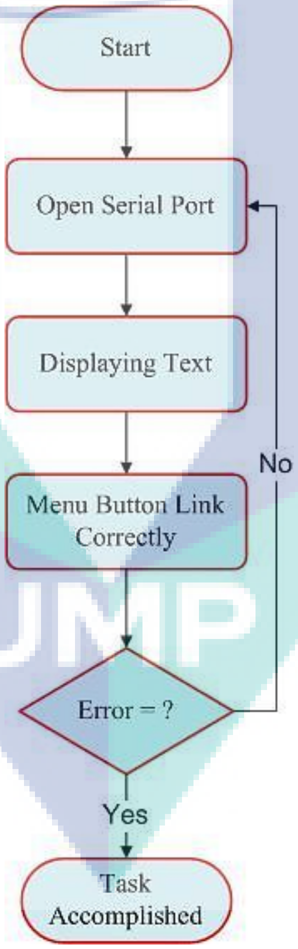


Figure 5.15: The flowchart of graphical-oriented testing

Figure 5.16 shows the plan view of the MicroEVAT system after the various testing procedure are conducted to reveal the contribution of multiple microcontroller hardware platform that optimized for use in academic environment.

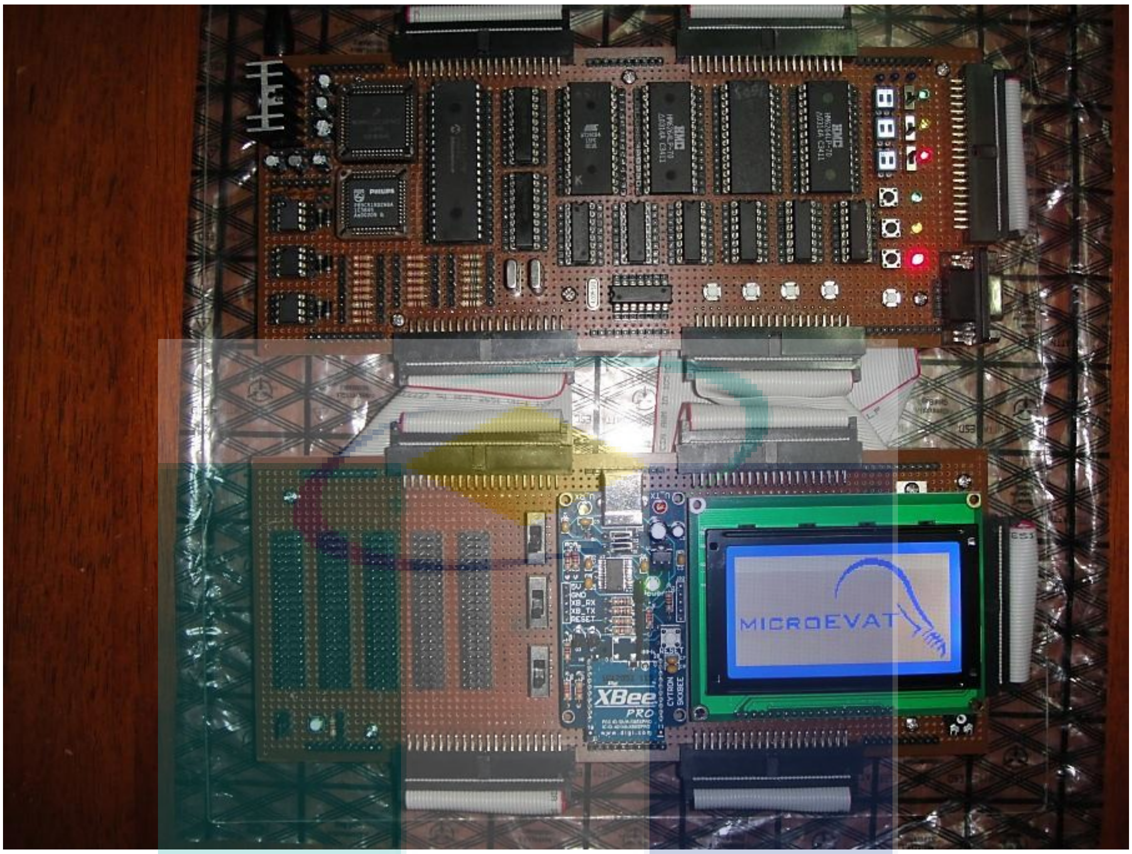


Figure 5.16: Plan View of the MicroEVAT prototype system

5.5 SUMMARY

This chapter describes the performance evaluation of a multiple microcontroller hardware platform that developed through the tests conducted in system board testing, preliminary tests and subsequent tests to reveal the contribution of multiple microcontroller hardware platforms that optimized for use in university environment. The proposed design system has shown the capability of multiple microcontroller modules as proposed in one evaluation platform. The feedback from the user surveys for MicroEVAT evaluations sheet could not be implemented in the tests due to the limitation of resources in this project.

CHAPTER 6

CONCLUSION

As a conclusion, this chapter discusses the outcome of the project and provides solution or suggestion of further work.

6.1 CONCLUDING REMARKS

The main focus of this project is to develop a multiple microcontroller hardware platform optimized for a use in academic environment. In order to achieve the objective, a prototype of evaluation tool that consists of the proposed three types of microcontroller families is designed. The importance of these microcontrollers in academic environment is determined by conducting a survey to students and also the neediness of multi-microcontroller hardware platform as development tools in the faculty. From the survey conducted, Freescale HC11, MCS51 and Microchip PIC18 microcontroller is identified as the most used in student's faculty academic program in Universiti Malaysia Pahang. The interface of these three microcontrollers are designed by providing separate power supply to avoid unnecessary power usage and combined into one serial communication link to personal computer. The separated power supply in the system board has produce less heat on voltage regulator compare to all microcontrollers powered at once. The combined serial communication interface requires limitless user operation on different microcontrollers.

The usage of unlimited number of input and output devices and memory can be overcome by using multiplexing mode and address decoding module. The use of address decoder in the system provides more programming space and application module that can be operated at one time. Extended memory that consist of external

RAM and ROM in the system are designed to implement mini-operating system that to aid user to upload and execute programs without any use of external software. The designed graphical user interface, RATNA Terminal is designed to work with the mini-operating system as to provide more user-friendly environment.

The mode selectivity of Freescale HC11 and MCS51 microcontroller is designed using mode selector module for accessing the internal and external memory of the system. This is to achieve goal for developing extensive application module with various form of input and output devices. The MicroEVAT system also is designed based on open module where expansion for input and output port for each microcontroller had been developed. The complete MicroEVAT integration is done after all the required system module are built and installed considering several factors such as power supply regulation, serial communication interface, address decoding module, and memory interface. In similar manner, application board offers several I/O devices such as LED, Bar Graph, 7 and 16 Segment Display, Traffic Light, Dot Matrix, LCD and GLCD, 8-ways DIP switches, Keypad, DC motor, and Stepper Motor. These various form of I/O devices are suitable to be used for educational purpose.

The serial communication interface for the MicroEVAT system is designed using diode IN4001 and serial driver DS275 for single communication port. The proposed serial communication interface has shown capable of stabilizing the output data from the microcontroller to the personal computer. The data that received shows the error free on the RATNA Terminal where the developed mini-operating system displayed successfully. This concludes that the system board is successfully communicated with the PC as the serial communication interface could hold all three microcontrollers in the MicroEVAT system.

Through the prototype testing, the MicroEVAT system had satisfactorily designed, built and tested. Hence it is concluded that the proposed multiple microcontroller hardware platform was successfully developed for the academic use. The list of author's publication paper can be viewed in Appendix G.

6.2 ACHIEVEMENTS

This research study has achieved all the objectives and scopes in designing a multiple microcontroller hardware platform for academic environment. The achieved goals for this research work as follows:

i. A single platform consisting three type of microcontroller families has been designed,

ii. Multi-varied of applications module that tested together with main system board has been tested successfully,

iii. The designed MicroEVAT system offers multi-purpose solution where students could select any of the microcontrollers provided using mode selector module,

iv. Reduced cost of the development system, where the MicroEVAT system could hold up to three type of microcontroller families,

v. The MicroEVAT system were developed with RATNA Terminal, the designed GUI for better user-friendly environment and single communication port for serial link between the microcontrollers,

vi. Serial communication with three types of interface, USB, Serial RS232 and wireless ZigBEE were designed, and

vii. The monitor program that designed with Graphical-oriented operating system, RATNA Terminal provides user interface with the MicroEVAT system and test programs for application modules.

6.3 RECOMMENDATIONS OF FUTURE WORK

The work in this thesis suggests that future enhancement can be carried out to further improve the design for achieving better performance or a more complete operation. Below are some of the proposed future works:

The first recommendation is the system should be constructed using printed circuit board (PCB) to produce reliable system. PCB of multi layers is suggested to be used to produce a compact and portable development board. On the other hand, small outline IC package (SOP) are better choice for the PCB to replace plastic lead carrier chip (PLCC) and dual-in line package (DIP) package if the size is considered.

The second recommendation is the program of the system should be improved to provide more user friendly environment. A complete IDE environment using Visual Studio must be implemented, this include a graphical simulator and a debugger for troubleshooting purpose.

The final recommendation is the improvement of more application module can be added and I/O devices can be included in the application board (expanded mode) for the needs of user as there are modern technologies, better components and reliable sensors are being produced today. Internal memory or Bootstrap mode system should be enhanced where users only need to change the switches without removing the pin assignment. By doing this, the operation done by the user can be reduced as the proposed serial communication interface in the MicroEVAT system.

REFERENCES

Abdul Aziz, H., Nik Yusoff, N.M.K. and Sapien, M.Z.B.M. 2010. MINI11: Microcontroller Development Board for SCL Approach. *Proceedings of 2010 IEEE Student Conferences on Research and Development*, pp. 178-182

Al-Dhaher, A.H.G. 2001. Integrating Hardware and Software for the Development of Microcontroller Based Systems. *Elsevier Journal of Microprocessors and Microsystems*. **25**(7): 317-328

Alves, G.R. and Martinas Ferreira, J.M. 1999. From Design-for-Test to Design-for-Debug-and-Test: Analysis of Requirements and Limitations for 1149.1. *17th IEEE VLSI Test Symposium*, pp. 479-480

Arefin, A.S., Masum Habib, K.M., Sultana, R. and Lutful Kabir, S.M. 2007. Designing a Low Cost Microcontroller-based Device for Multipurpose Learning. *IEEE Transactions on Education*. **44**(7): 88-94

Azad, A.K.M. and Lakkaraju, V.K. 2003. Development of a Microcontroller Laboratory Facility for Directing Students towards Application Oriented Projects. *Proceedings for American Society for Engineering Education IUIN sectional conference*, pp. 145-151

Brawicz, A. and Morawski, R. 1999. Teaching Measuring systems beyond the year 2000. *IEEE Instrumentation Measurement Magazine*. **2**(1): 20-27

Brockman, J. 1998. Process Multi-Circuit Optimization. *Proceedings for Design Automation Conference*, pp. 382-387

Cady, F. 1997. *Software and Hardware Engineering – Motorola M68HC11*. London: Oxford University Press

Cady, F. 2008. M68HC11 Microcontroller Board and Prototyping, MC68HC11 and MC68HC12. http://www.coe.montana.eduJee/cady/ee361/hcll1nks.htm (23 July 2010)

Caspi, P., Sangiovanni-Vincentelli, A. and Almeida, L. 2005. Guidelines for a Graduate Curriculum on Embedded Software and Systems. *ACM Transactions on Embedded Computing System*. **4**(3):587-611

Chandrakasan, A.P., Sheng, S. and Brodersan, R.W. 1992. Low Power CMOS Digital Design. *IEEE Journal of Solid-State Circuit.* **27**: 473-477

Cheng, B.H.C., Rover, D.T. and Mutka, M.W. 1998. A Multi-pronged Approach to Bringing Embedded System into Undergraduate Education. *ASEE Annual Conferences*, pp. 117-121

Ching, P.C., Cheng, Y.H. and Ko, M.H. 1994. An In Circuit Emulator for TMS320C25. *IEEE Transactions on Education.* **37**(1): 51-56

Cho, J.M., Choi, S.I. and Lee, D.K. 2003. A Flash based Multimedia Interactive Tutoring System for Distance Education of Biomedical Engineering Student: New Approach to Teaching Microcontroller based System. *Proceedings of the 25TH Annual International Conference of the IEEE EMBS*. pp. 3540-3543

Chu, R.H., Lu, D.D. and Sathiakumar, S. 2008. Project Based Lab Teaching for Power Electronics and Drives. *IEEE Transactions on Education.* **51**(1): 108-113

Correa, C.R. and Awad, S.S. 2003. Embedded Controller Software and Algorithm Development Tool. *IEEE Transactions on Instrumentation and Measurement.* **52** (3): 885-890

Databeans. 2009. Microcontroller Market Share, Electronics.ca Publications, Mar 2010. http://www.electronics.ca/publications/products/2009-Microcontrollers-Market-Share.html (2 August 2010)

Dutson, A.J., Todd, R.H., Magleby, S.P. and Sorensen, C.D. 1997. A Review of Literature on Teaching Engineering Design Through Project-oriented Capstone Courses. *Journal on Engineering Education.* **32**: 17-28

Ferreira, L.F., Matos, E.L., Menendez, L.M. and Mandado, E. 2005. MILES: A microcontroller Learning System combining Hardware and Software Tools. *35th ASEE / IEEE Frontiers in Education Conferences*, pp. 77-81

Ferrero Martin, F.J., Campro Rodriquez, J.C. and Alvarez Anton, J.C. 2005. An Electronics Instrumentation Design Project for Computer Engineering Students. *IEEE Transactions on Education.* **48**(3): 472-481

Flammini, A., Marioli, D. and Taroni, A. 2001. A Low Cost Diagnostic Tool for Stepping Motors. *IEEE Transactions on Instrumentation and Measurement.* **50**(1): 157-162

Hadzic, D. 2004. *Application Layer - Communication Protocol for Home Automation.* M.Eng. Thesis. University of Oslo, Norway

Hedley, M. and Barrie, S. 1998. An Undergraduate Microcontroller System Laboratory. *IEEE Transactions on Education.* **41**(4): 345-349

Hennessy, J.L. and Patterson, D.A. 1996. *Computer Architecture, A Quantitative Approach, Second Edition*. USA: Morgan Kaufmann

Hilbor, R.B. 1994. Team Learning for Engineering Students. *IEEE Transactions on Education.* **37**(2): 207-211

Huang, I.J., Kao, C.H., Chen, H.M. and Juan, C.N. 2002. A Retargetable Embedded In-Circuit Emulation Module for Microprocessor. *IEEE Transactions on Design and Test of Computers.* **40**(2): 28-38

Huber, N., Hromalik-Pouchet, M.S., Carozzi, T.D., Gough, M.P. and Buckley, A.M. 2011. Parallel processing speed increase of the one-bit auto-correlation function in hardware. *Elsevier Journal on Microprocessors and Microsystems*. **35**: 297–307

Hussmann, S. and Jensen, D. 2007. Crazy Car Race Contest: Multicourse Design Curriculum in Embedded System Design. *IEEE Transactions on Education.* **50**(1): 61-67

Jackson, D.J. and Caspi, P. 2005. Embedded Systems Education: Future Directions, Initiatives, and Cooperation. *SIGBED Review (Special Issue on the First Workshop on Embedded System Education).* **2**(4): 113-121

Kaeli, D. and Platcow, R. 2006. Experiences with the Blackfin Architecture in an Embedded System Lab. *Proceedings for 2006 Workshop on Computer Architecture Education*. pp. 301-307

Kaufman, D.B., Felder, R.M. and Fuller, H. 2000. Accounting for Individual Learning Effort in Cooperative Learning Teams. *Journal on Engineering Education.* **38**: 237-241

Kavvadias, N., Neofotistos, P., Nikolaidis, S., Kosmatopoulos, K. and Laopoulos, T. 2004. Measurement Analysis of the Software-Related Power Consumption in Microprocessors. *IEEE Transactions on Instrumentation and Measurement*. **53**(4): 1106–1112

Khan, T.H. and Ninad, N.A. 2006. Design of a Low Cost 8051 Architecture Based Microcontroller Learning Kit. *Asian Journal of Information Technology*. **5**(2): 213-218

Konstantakos, V., Chatziqeorgiou, A., Nikolaidis, S. and Laopoulos, T. 2008. Energy Consumption Estimation in Embedded Systems. *IEEE Transactions on Instrumentation and Measurement.* **57**(4): 797-804

Kroupis, N. and Soudris, D. 2011. FILESPPA: Fast Instruction Level Embedded System Power and Performance Analyzer. *Elsevier Journal on Microprocessors and Microsystems*. **35**: 329-342

Laopoulos, T. 1999. Teaching Instrumentation and Measurement in the Complex Systems Era. *IEEE Instrumentation Measurement Magazines.* **2**(1): 28-30

Lawrence, P.D. and Mauch, K. 1987. *An Introduction Real-time Microcontroller System Design*. USA: McGraw-Hill

Lee, C.S., Su, J.H., Lin, K.E. and Chang, J.H. 2010. A Project Based Laboratory for Learning Embedded System Design with Industry Support. *IEEE Transactions on Education.* **50**(2): 173-181

Lee, C.S., Wang, Y.N., Su, J.H., Chang, C. and Chen, H.S. 2006. Work in Progress: A Joint Effort of Lunghwa University and Holtek Semiconductor Inc. on Improving Microcontroller Education. *36th ASEE / IEEE Frontiers in Education Conference*, pp. 128-129

Ling, L.C. 2008. *The MC68HC11 Development Board with Integrated Development Environment (IDE)*. B.Eng. Thesis. Universiti Malaysia Pahang, Malaysia

Ma, C., Li, Q., Liu, Z. and Jin,Y. 2010. Low Cost AVR Microcontroller Development Kit for Undergraduate Laboratory and Take-home Pedagogies. *2010 2nd International Conference on Education Technology and Computer*, pp. 35-38

MacNamee, C. and Heffernan, D. 2000. Emerging On-Chip Debugging Techniques for Real-Time Embedded Systems. *IEEE Journal Computing and Control Engineering.* **11**(6): 295-303

Magana, M.E. and Holzapfel, F. 1998. Fuzzy-logic Control of an Inverted Pendulum with Vision Feedback. *IEEE Transactions on Education.* **41**(2): 165-190

Mallalieu, K., Ariestas, R. and So'Brien, D. 1994. An Inexpensive PC Based Laboratory Configuration for Teaching Electronics Instrumentation. *IEEE Transactions on Education.* **37**(1): 91-96

Maskell, D.L. 1999. Student-based Assessment in a Multi-disciplinary Problem based Learning Environment. *Journal on Engineering Education.* **35**: 237-241

Maskell, D.L. and Grabau, P.J. 1998. A Multi-disciplinary Cooperative Problem-based Learning Approach to Embedded System Design. *IEEE Transactions on Education.* **41**(3): 101-103

Matrix Multimedia Limited. 2007. Flowcode and E-blocks for Education purposes. http://www.matrixmultimedia.com (16 December 2009)

Microchip. 2005. 25LC160 Page. http://www.microchip.com/download/en/devicedoc (18 January 2010)

Mitescu, S.M. 2005. *Microcontrollers in Practice*. Berlin: Springer

Moallem, M. 2004. A Laboratory Test bed for Embedded Computer Control. *IEEE Transactions on Education.* **47**(3): 340-347

Mohd Sapein, M.H. 2009. *Mini 11 Microcontroller Laboratory Hardware Platform with IDE Approach*. B.Eng. Thesis. Universiti Malaysia Pahang, Malaysia

Morawski, R., Pallas Areny, R., Petriu, E. and Siegal, M. 2000. Current Trends on Teaching Instrumentation and Measurement. *16th IEEE Instrumentation Measurements Technology Conference*, pp. 1715-1726

Mosterman, P.J. 2006. Automatic Code Generation: Facilitating New Teaching Opportunities in Engineering Education. *36th ASEE/IEEE Frontier in Education Conference*, pp. 165-171

Mosterman, P.J., Prabhu, S. and Erkkinen, T. 2004. An Industrial Embedded Control System Design Process. *The Inaugural CDEN Design Conference*, pp. 17-21

Motorola Inc. 2001. M68HC11E Series Family Technical Datasheet. http://www.alldatasheet.com/motorolahc11tec.pdf (15 September 2010)

Muhammad Ali, M. 2000. *The 8051 Microcontroller and Embedded Systems*. Upper Saddle River, New Jersey: Prentice Hall

Mukherjea, S. and Stasko, J.T. 1993. Applying Algorithm Animation Techniques for Program Tracing, Debugging and Understading. *15TH International Conference on Software Engineering*. pp. 456-465

Murray, M.R. and Garbini, J.L. 1997. Embedded Computing in the Mechanical Engineering Curriculum: A Course featuring Structured Laboratory Exercises. *Journal on Engineering Education*. **32**: 285-290

Muttreja, A. and Raghunathan, A. 2007. Automated Energy / Performance Macro-Modeling of Embedded Software. *IEEE Transactions on Computer - Aided Design Integrations Circuits System*. **26**(3): 542–552

Ni, J. and Luo, J. 2010. Microcontroller-based Engineering Education Innovation. *2010 International Conference on Educational and Information Technology*, pp. 109-112

Nik Yusoff, N.M.K. 2006. DEE3263 Embedded Controller Technology. *Annual Student Performance Report*. Universiti Malaysia Pahang, Malaysia

Nikolaidis, S., Kavvadias, N., Laopoulos, T., Bisdounis, L. and Blionas, S. 2005. Instruction Level Energy Modeling for Pipelined Processors. *Elsevier Journal on Embedded Computing*. **1**(3): 317–324

Novak, J.D. 1990. Concept Mapping: A useful Tool for Science Education. *Journal of Research in Teaching*. **27**(10): 937-949

Ontoria, A. 2000. *Concept Maps: A Technique for Learning*. Spain: Narcea S.A. Ed

Palop, J.G.M. and Teruel, M.A. 2000. Virtual Work Bench for Electronic Instrumentation Teaching. *IEEE Transactions on Education.* **43** (1): 15-18

Perez Quinones, M.A. and Cruz-Rivera, J.L. 1999. Integrated Development Environment for a Microcontroller System Laboratory. *29th ASEE / IEEE Frontiers in Education Conferences*, pp. 111-116

Perez, M. 2003. *A Methodology of Develop Multimedia / Hypermedia Software for Education*. Ph.D. Thesis. University of North Carolina, USA

Piguet, C., Masgonty, J.M., Arm, C. and Durand, S. 1997. Low-Power Design of 8-b Embedded CoolRisc Microcontroller Cores. *IEEE Journal of Solid-State Circuits*. **32**(7): 1067-1078

Peatman, J.B. 1998. *Design with PIC Microcontroller*. USA: Prentice Hall

Prasad, C.P.W., Dominic, M.M. and Singh, A.K. 2003. Variable Order Verification Use of Logic Representation. *6TH International Conference on Asian Digital Libraries*. pp. 689-695

Predko, M. 2000. *Programming and customizing PICmicro microcontrollers*. USA: McGraw-Hill

Qiu, M., Guo, M., Liu, M., Xue, C., Yang, L.T. and Loop, S. 2009. Scheduling and bank type assignment for heterogeneous multi-bank memory. *Journal of Parallel and Distributed Computing (JPDC)*. **69**(6): 546–558.

Raghavan, M. 2005. *Testing of a New Wireless Embedded System*. M.Eng. Thesis. University of North Carolina, USA

Risks, K.G. and Jackson, D.J. 2006. A Case for the VME bus Architecture in Embedded Systems Education. *IEEE Transactions on Education*. **49**(3): 332-345

Rupp, M., Burg, A. and Beck, A. 2003. Rapid prototyping for Wireless Designs: The Five-ones Approach. *Elsevier Journal of Signal Processing*. **83**(7): 1427-1444

Schmalzel, J.L. 1999. I&M Education for the New Millennium. *IEEE Instrumentation Measurement Magazine*. **2**(1): 31-36

Scott, M.I. 1999. *The 8051 Microcontroller*. Upper Saddle River, New Jersey: Prentice Hall, 1999.

Seong, K.C. 2007. *The MC68HC11 Development Board*. B.Eng. Thesis. Universiti Malaysia Pahang, Malaysia

Sheu, S.S. and Cheng, K.H. 2011. Fast-write Resistive RAM (RRAM) for Embedded Applications. *IEEE Transaction on Design and Test of Computers*. **56**(2): 64-71

Smith, J.E. and Nair, R. 2005. The Architecture of Virtual Machine. *IEEE Journal of Computer*.**38**(5): 32-38

Smolnikar, M. and Mohorcic, M. 2008. A Framework for Developing a Microchip PIC Microcontroller Based Applications. WSEAS Transactions on Advances in Engineering Education. **5**(2): 83-91

Spasov, P. 2004. *Microcontroller Technology - The 68HC11 and 68HC12*. USA: Pearson Education International

Spasow, P. 1996. *Microcontroller Technology, The 68HC11*. Eaglewood Cliffs, New Jersey: Prentice Hall

Stapleton, W.A. 2006. Microcomputer Fundamentals for Embedded Systems Education. *36th ASEE / IEEE Frontiers in Education Conference*, pp. 56-61

Stoltz, T. 2003. *The Development of Microcontroller Hardware Platform with MC68HC11E2*. M.Eng. Thesis. University of Detroit Mercy, USA

Stoltz, T., Paulik, M. and Al-Holou, N. 2005. A Microcontroller Laboratory Hardware Platform for Academic Environment: The UDM-EVB. *35th ASEE / IEEE Frontiers in Education Conferences*, pp. 35-40

Striegel, A. and Rover, D. 2002. Enhanced Student Learning in an Introductory Embedded System Laboratory. *32nd ASEE / IEEE Frontiers in Education Conferences*, pp. 138-143

Vaglica, J.J. and Gilmour, P.S. 1990. How to select a Microcontroller. *IEEE SPECTRUM*. **18**(4): 106-109

Valdes, M.D., Moure, M.J. and Mandado, E. 1999. Educational Application of Virtual Instruments Based on Reconfigurable Logic. *IEEE MSE '99 International Conferences on Microelectronics System Education*. pp. 24-25

Valvano, A. and Jonathan, B. 2003. *Embedded Microcomputer Systems: Motorola 6811 and 6812 Simulation*. Pacific Grove, USA: Brooks / Cole-Thomson Learning

Vun, N. and Goh, W.B. 2007. The design of effective low cost embedded processor development kits for supporting take-home self-practice pedagogies. *2007 International Conference on Parallel and Distributed Systems*, pp. 1-7

Watanabe, K. 1999. Instrumentation Educations in Japan: Despair and Expectation. *IEEE Instrumentation Magazines*. **2**(1): 14-19

Wild, A., Quigley, J., Feddler, J. and Ledford, S. 1997. A 0.9V Microcontroller for Portable Applications. *IEEE Journal of Solid-State Circuit*. **32**(7): 1049-1055

William, K. 1998. *Microprocessor and Microcontroller Fundamental*. Upper Saddle River, New Jersey: Prentice Hall

Witte, R.A. 1999. Engineering Education for Test and Measurement Designers. *IEEE Instrumentation Measurements Magazine*. **2**(1): 11-13

Wolf, W. and Madsen, J. 2000. Embedded Systems Education for the Future. *IEEE Transactions on Education*. **88**(1): 23-30

Zhang, L., Qiu, M., Edwin, H.M. and Zhuge, Q. 2011. Variable assignment and instruction scheduling for processor with multi-module memory. *Elsevier Journal on Microprocessors and Microsystems*. **35**: 308-317

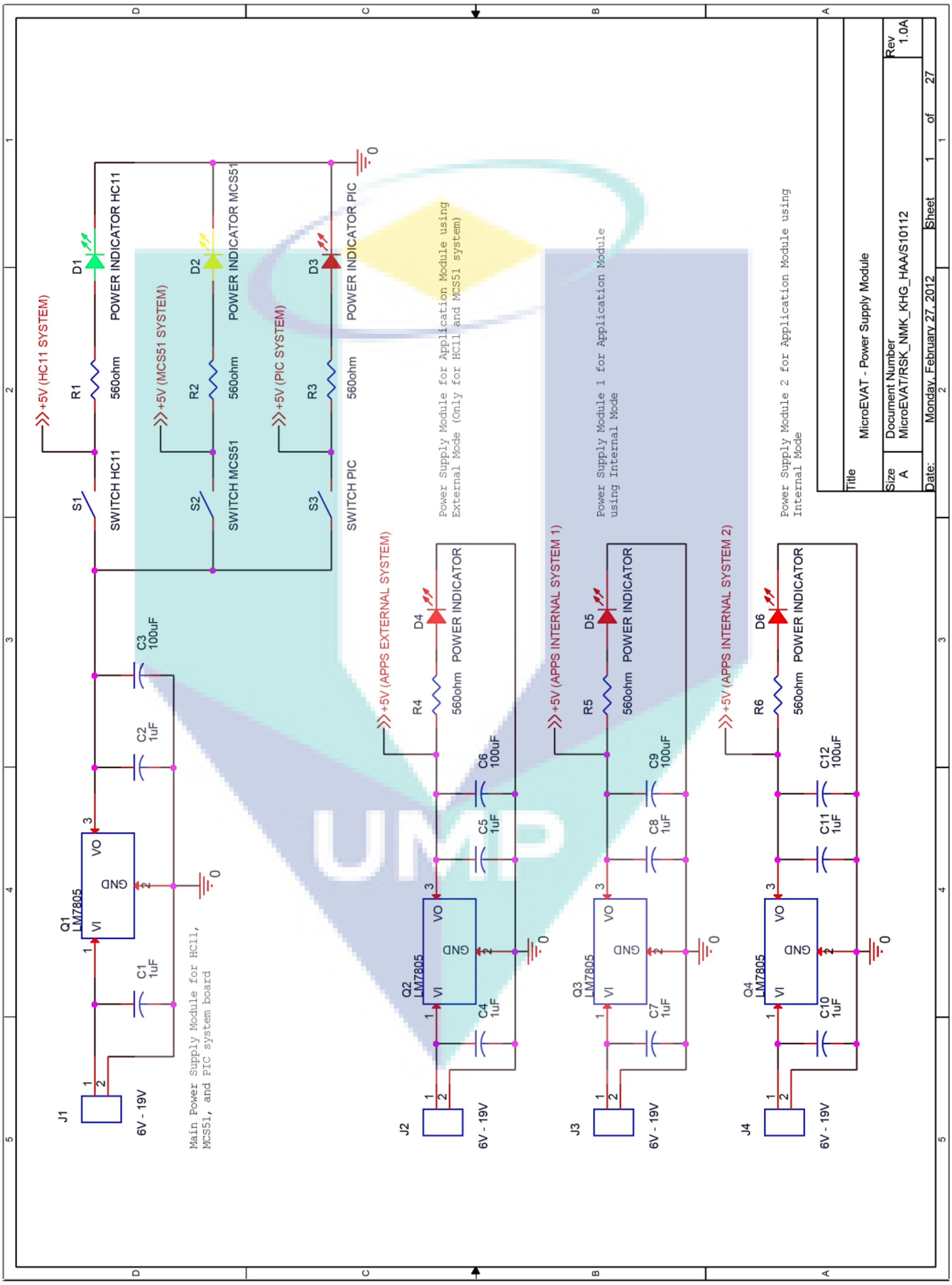
Zimmermann, R. and Gupta, R. 1996. Low-Power Logic Styles: CMOS vs. CPL. *22nd European Solid-State Circuit Conferences*. pp. 112-115

Zotos, K., Litke, A., Chatzigeorgiou, A., Nikolaidis, S. and Stephanides, G. 2005. Energy Complexity of Software in Embedded Systems. *IASTED International Conferences in Automation and Control Applications (ACIT-ACA)*, pp. 67-71

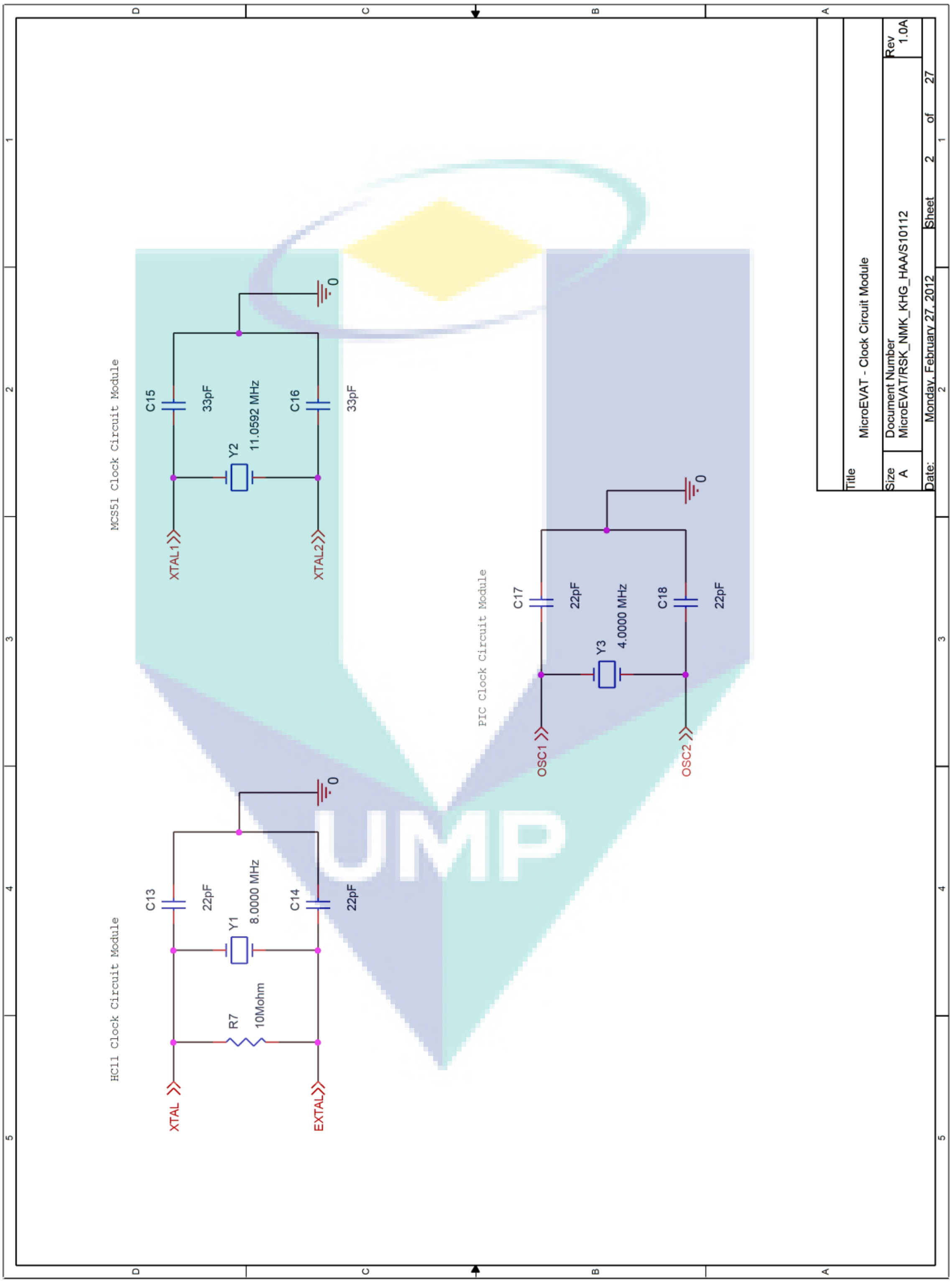
APPENDIX A

SCHEMATICS DIAGRAM OF MICROEVAT

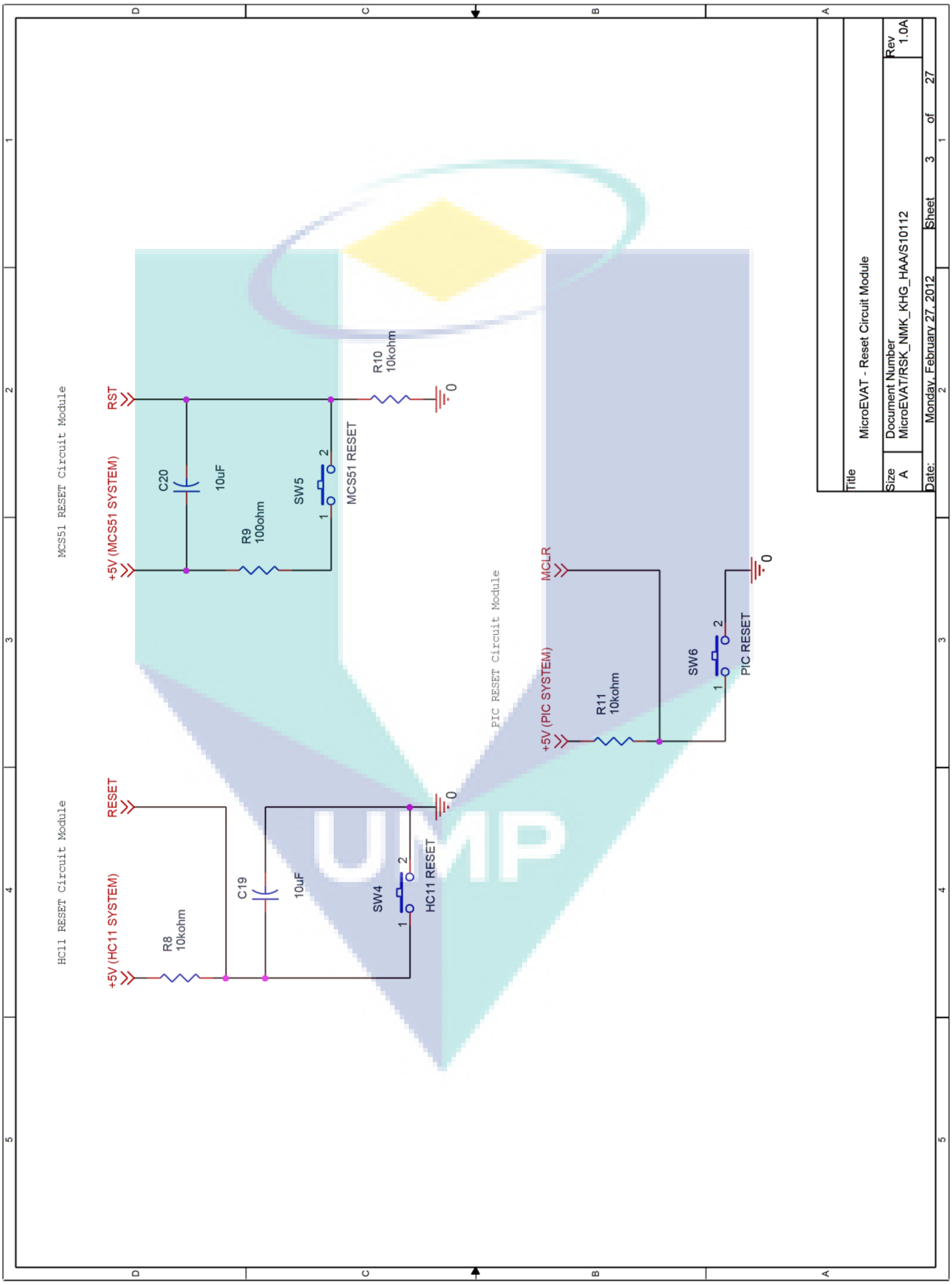
A1 Power Supply Module



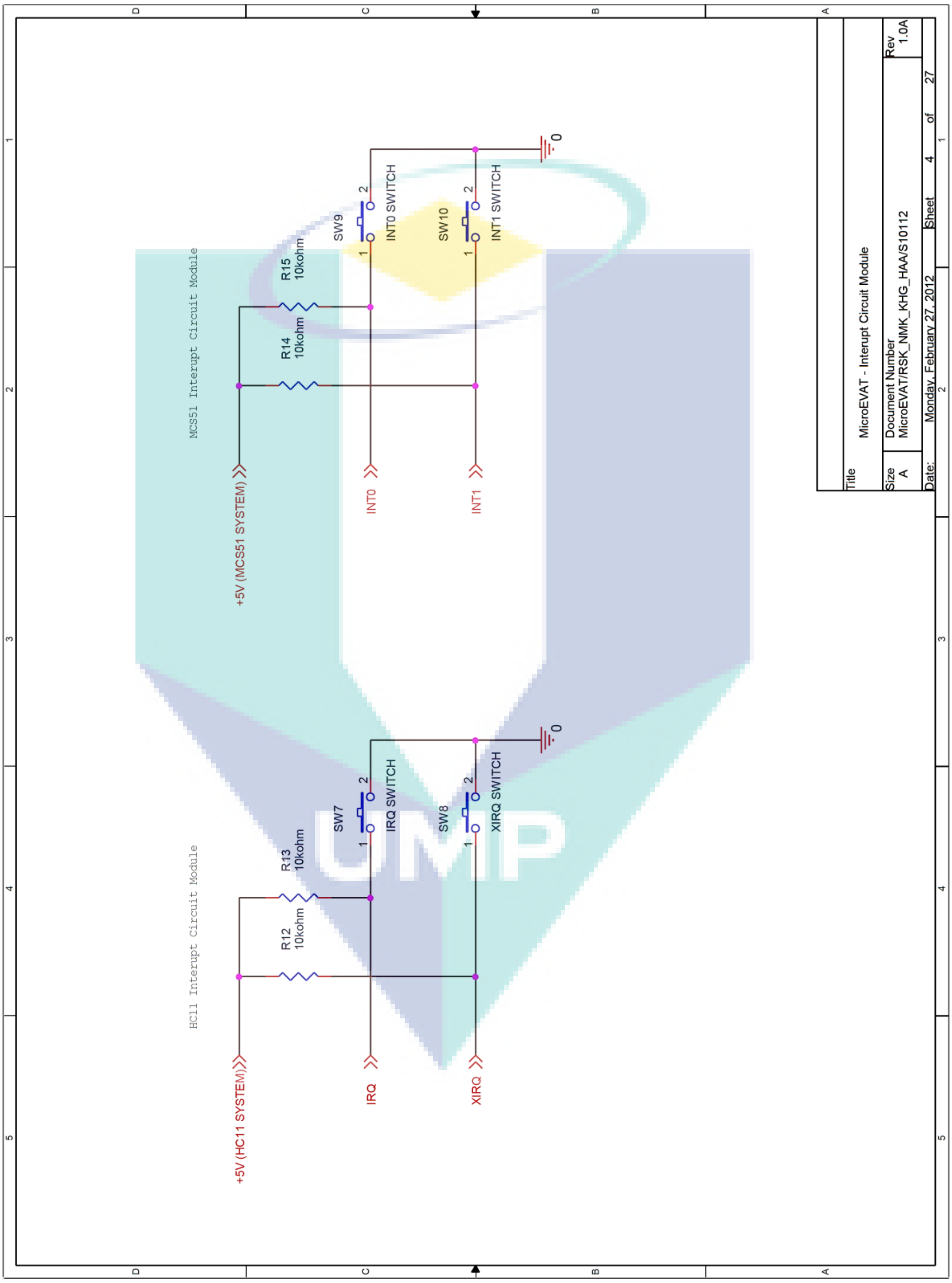
A2 Clock Circuit Module



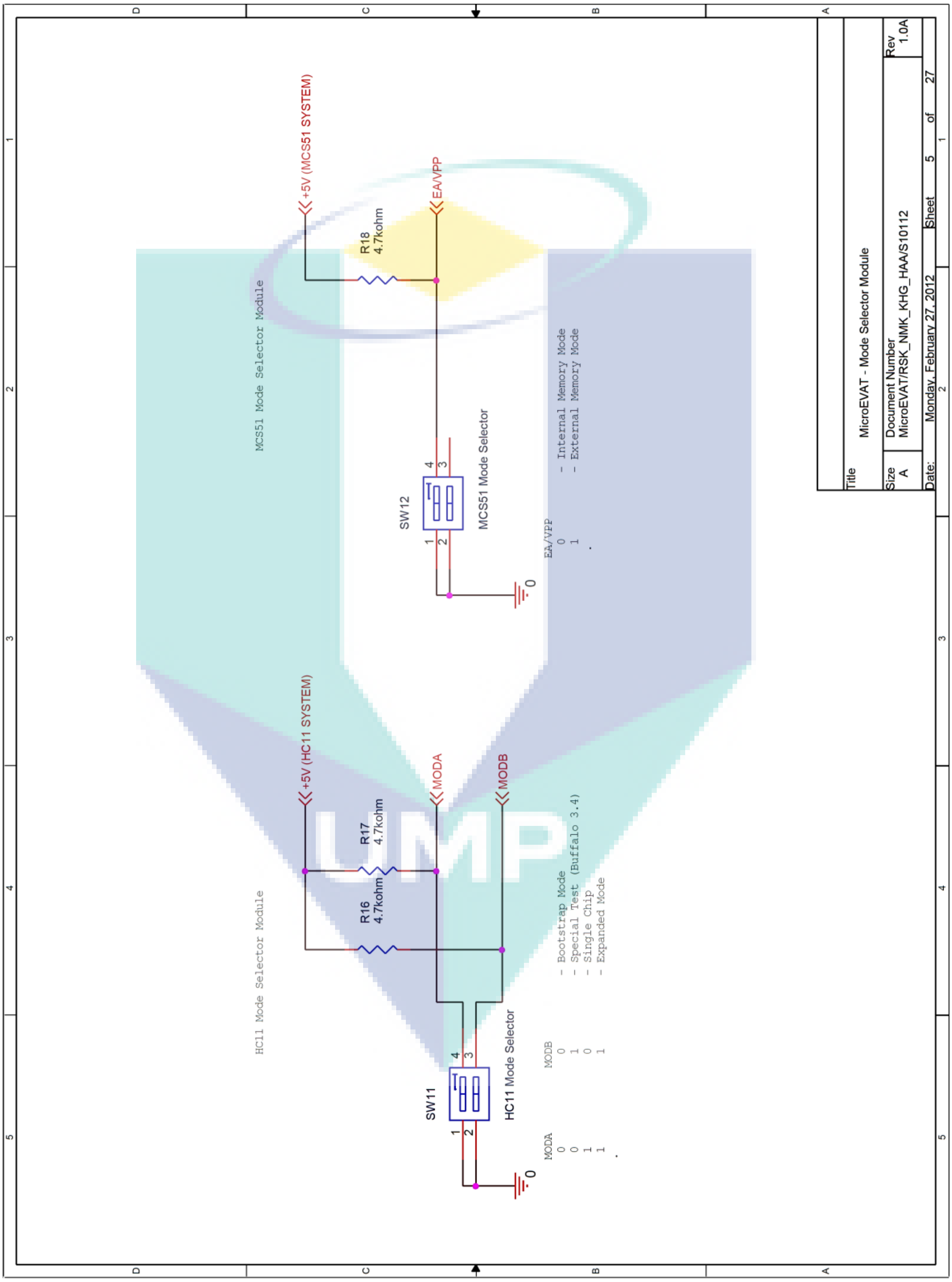
A3 RESET Circuit Module



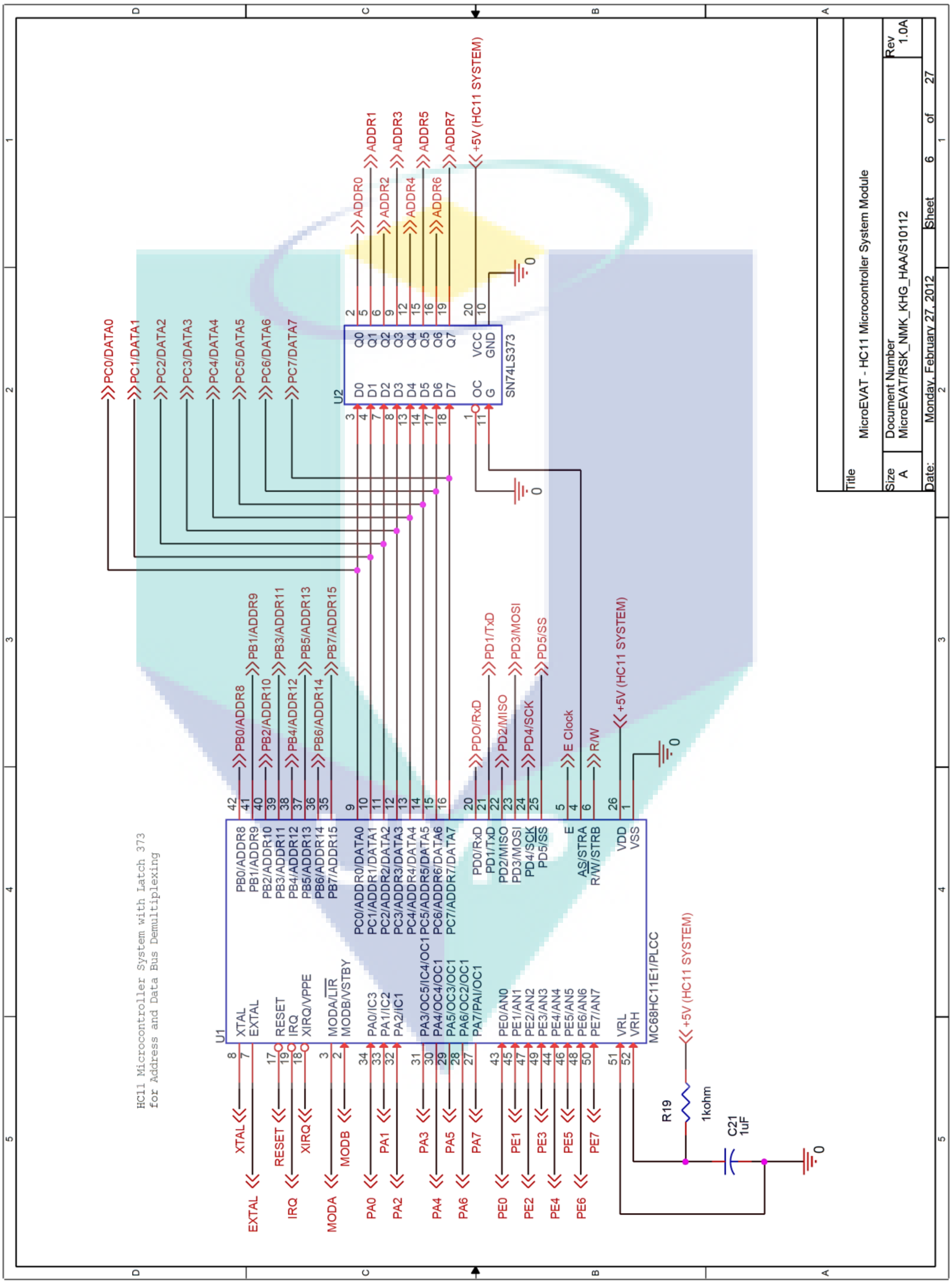
A4 Interrupt Circuit Module



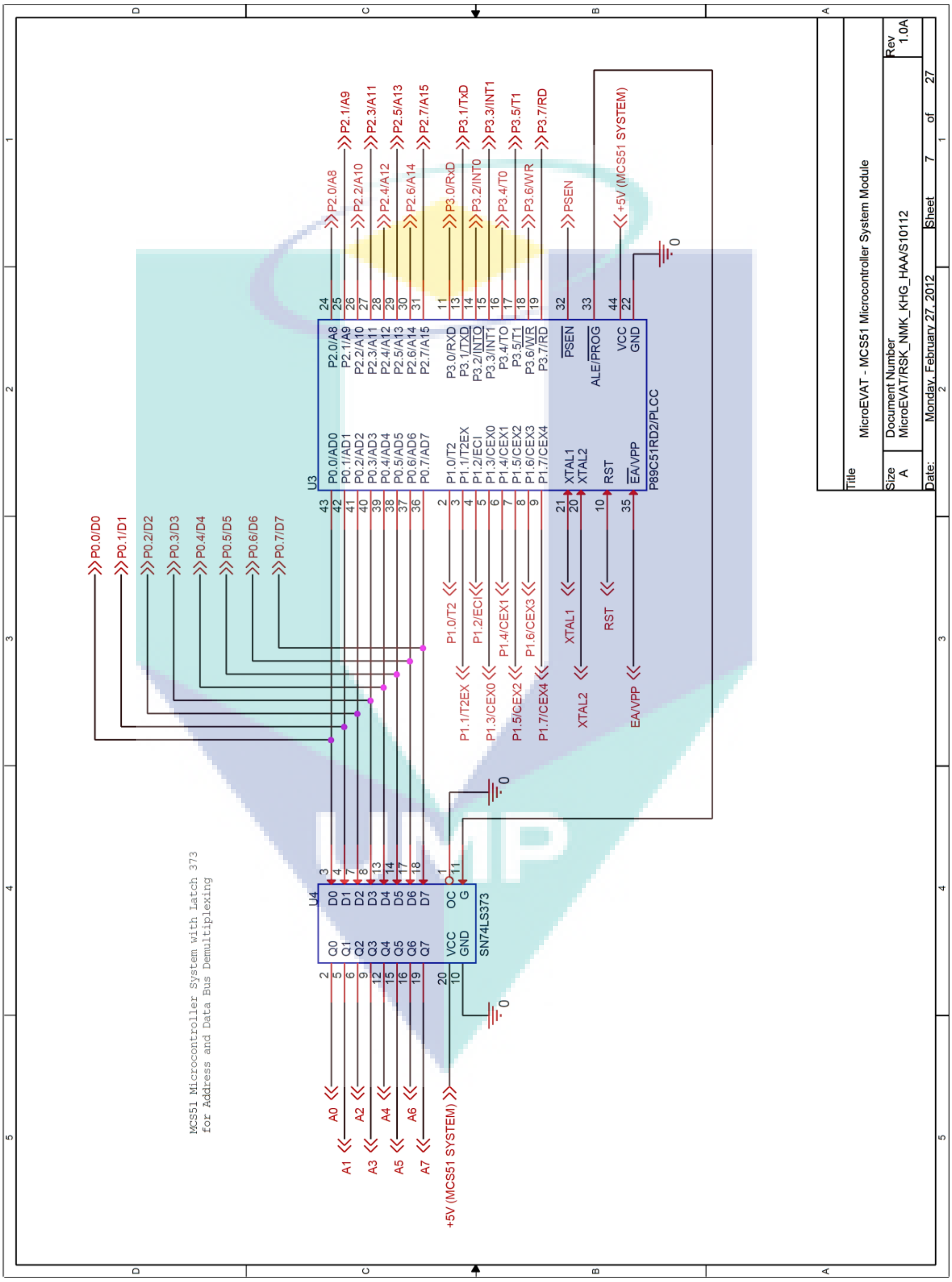
A5 Mode Selector Module



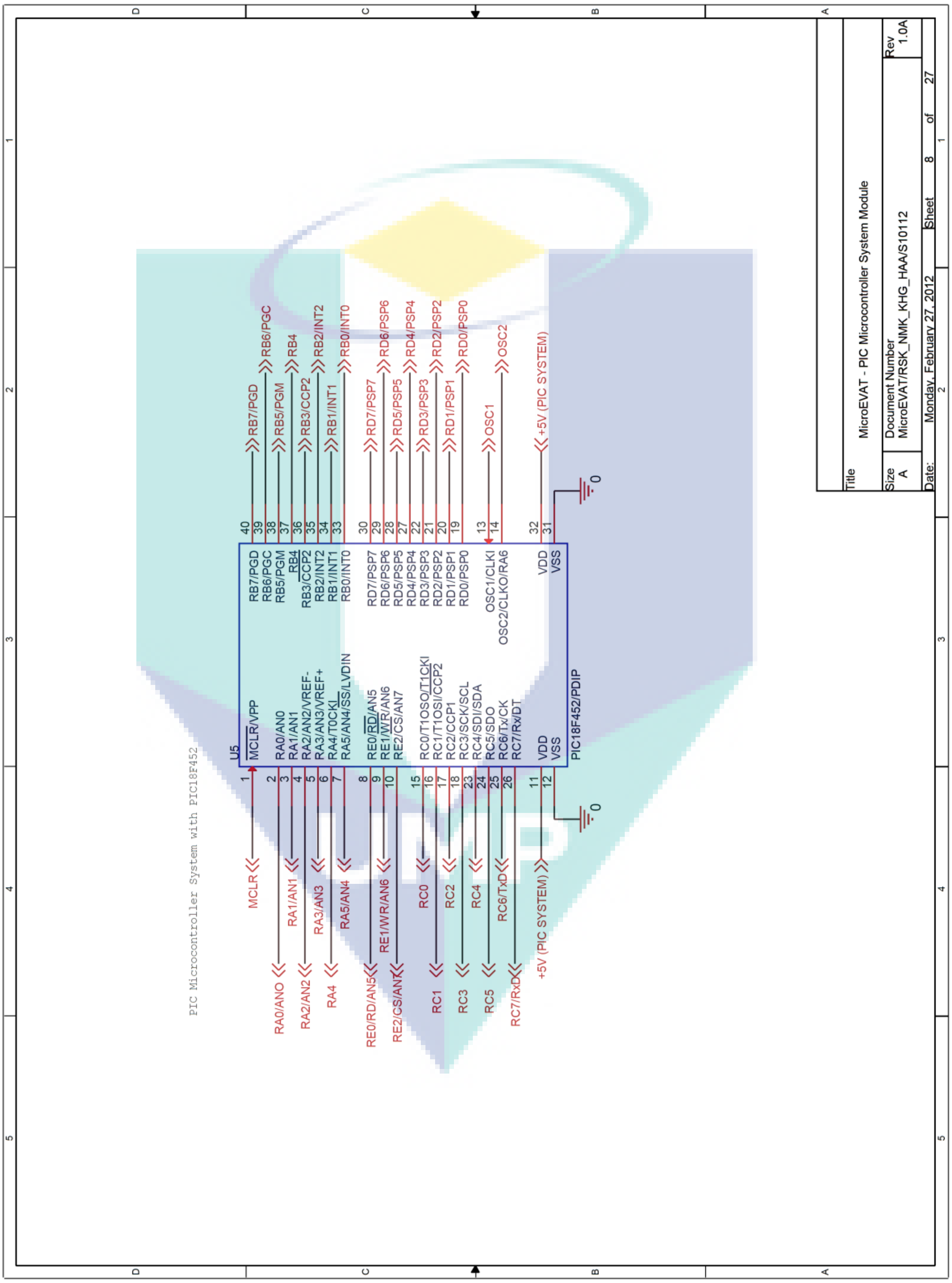
A6 HC11 Microcontroller System Module



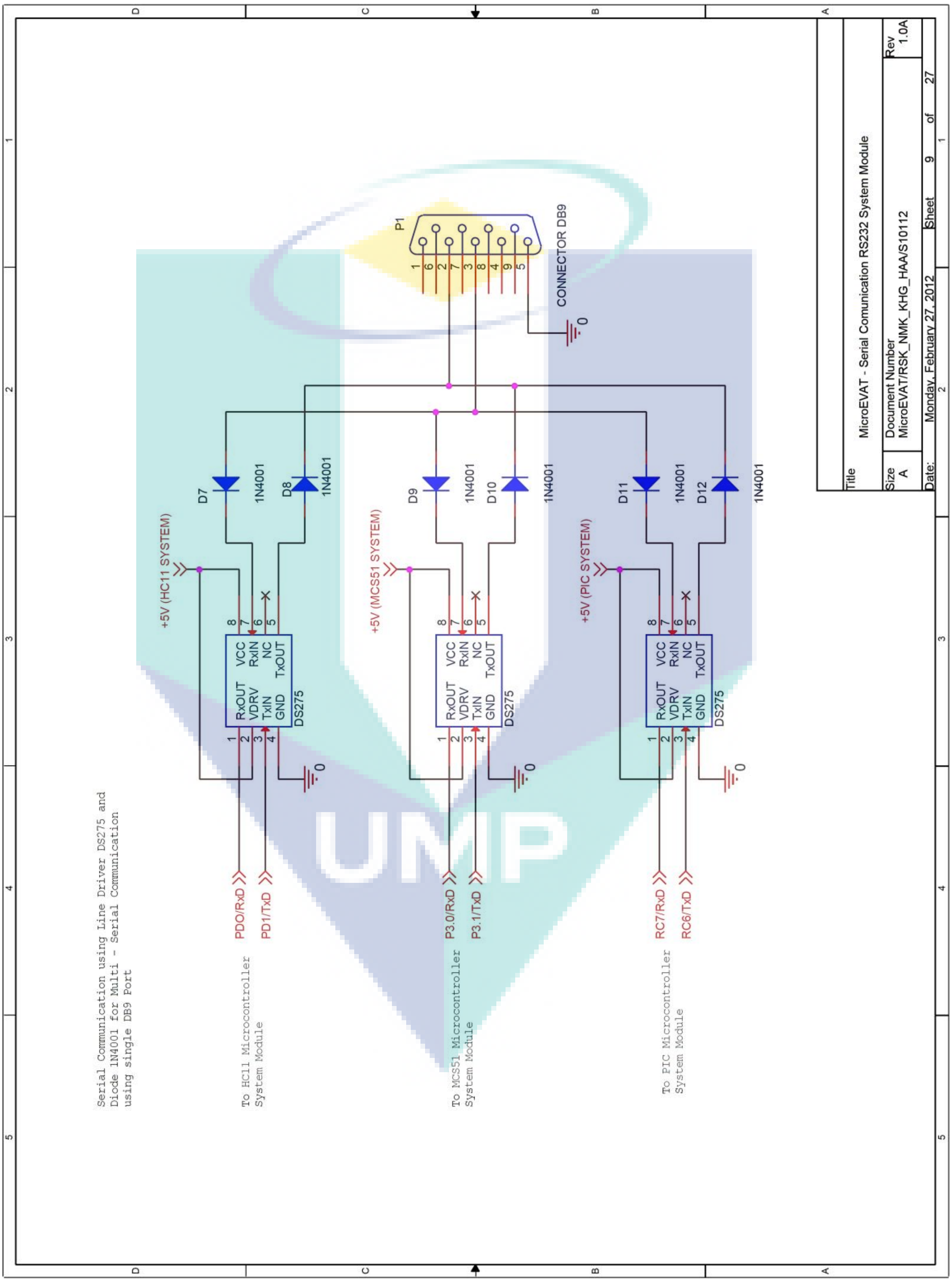
A7 MCS51 Microcontroller System Module



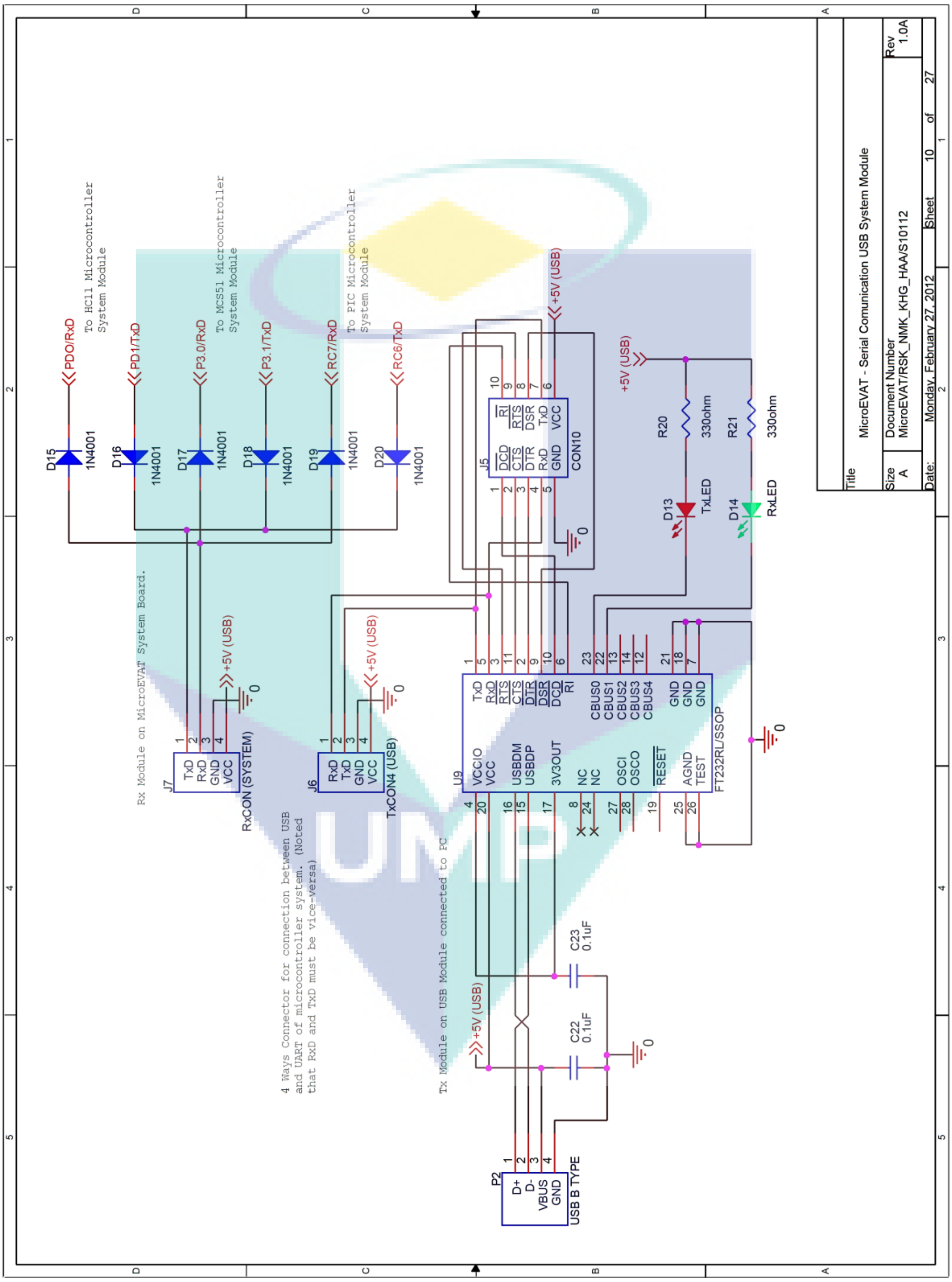
A8 PIC Microcontroller System Module



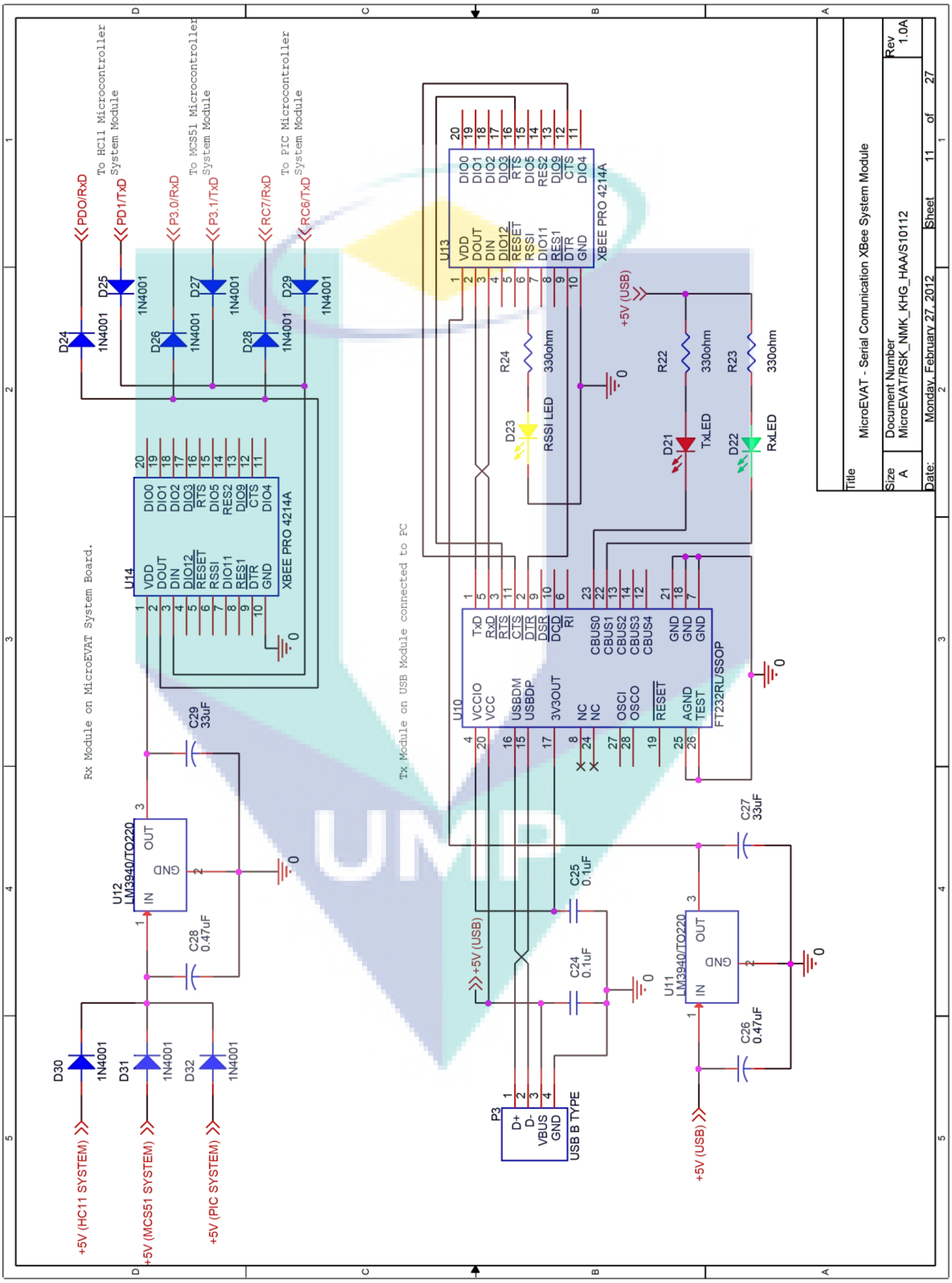
A9 Serial Communication RS232 System Module



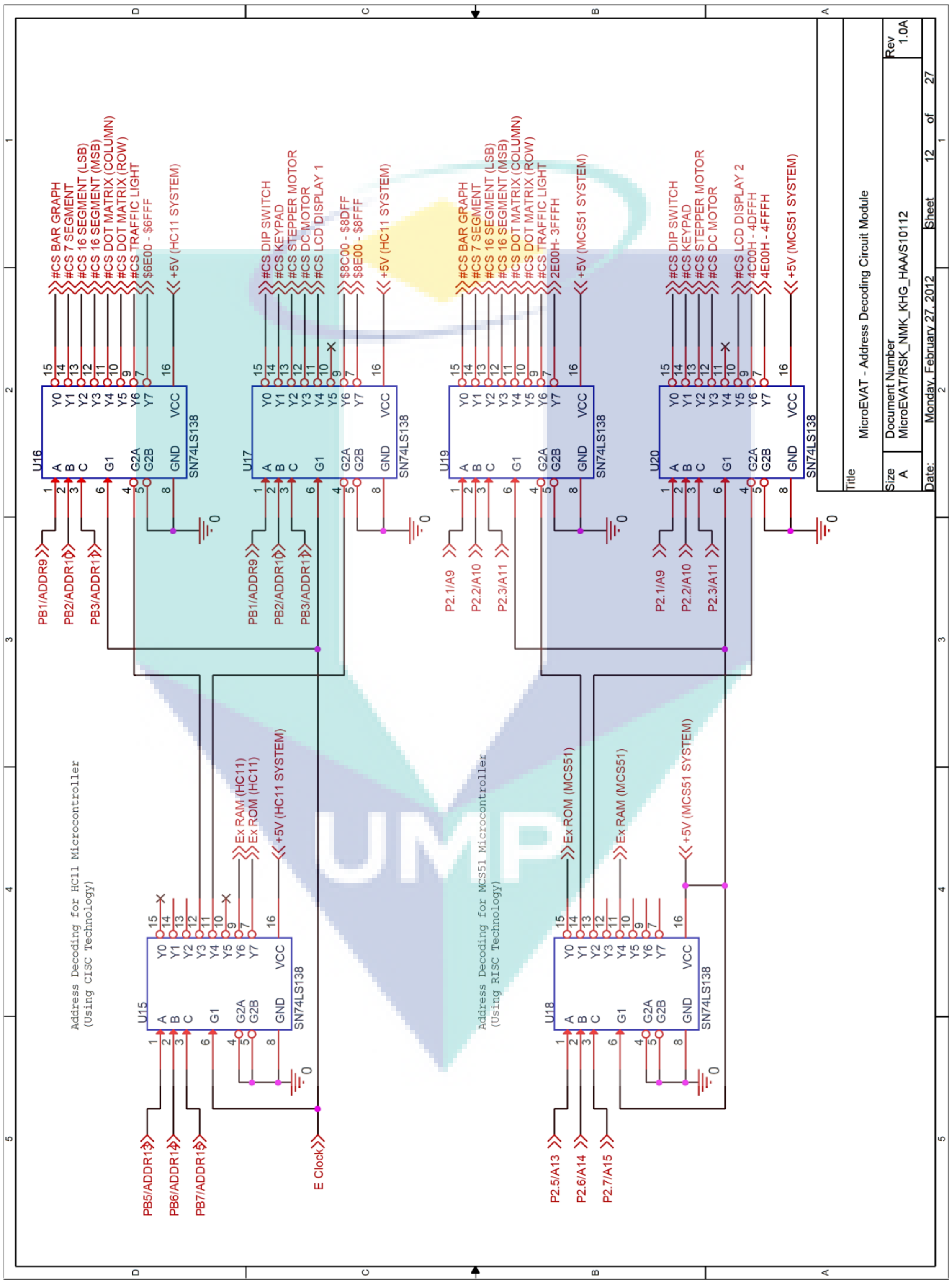
A10 Serial Communication USB System Module



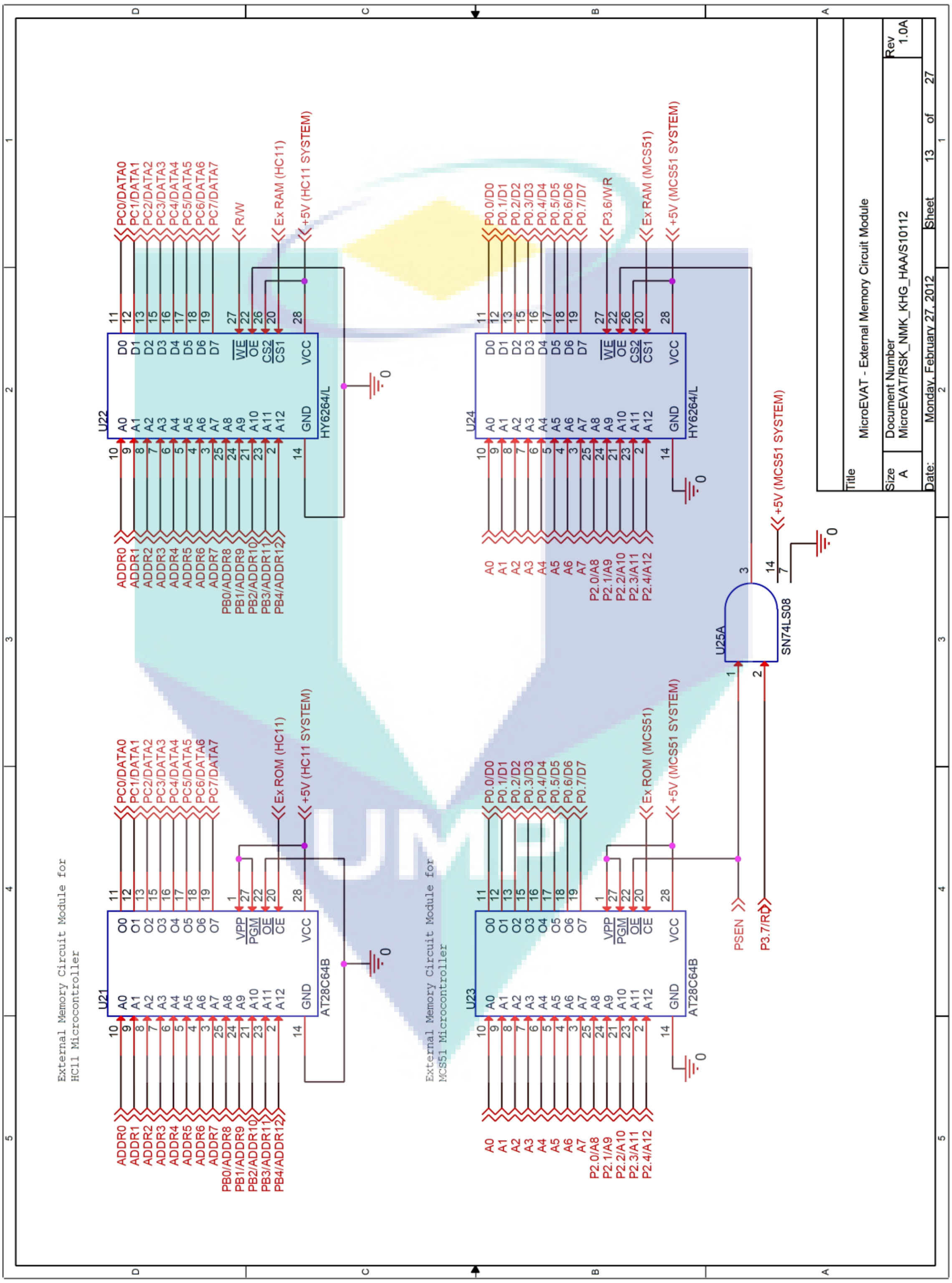
A11 Serial Communication ZigBEE System Module



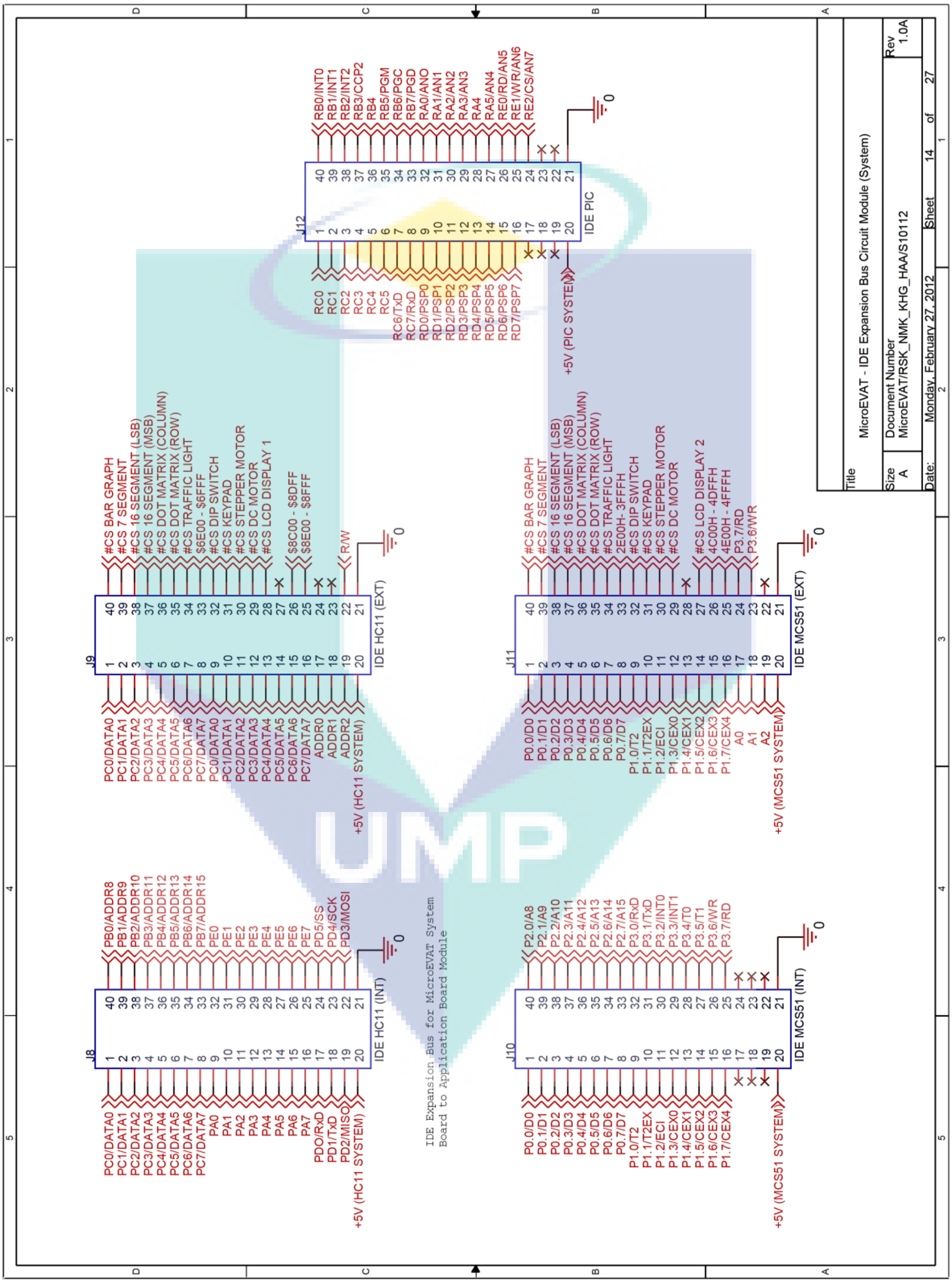
A12 Address Decoding Circuit Module



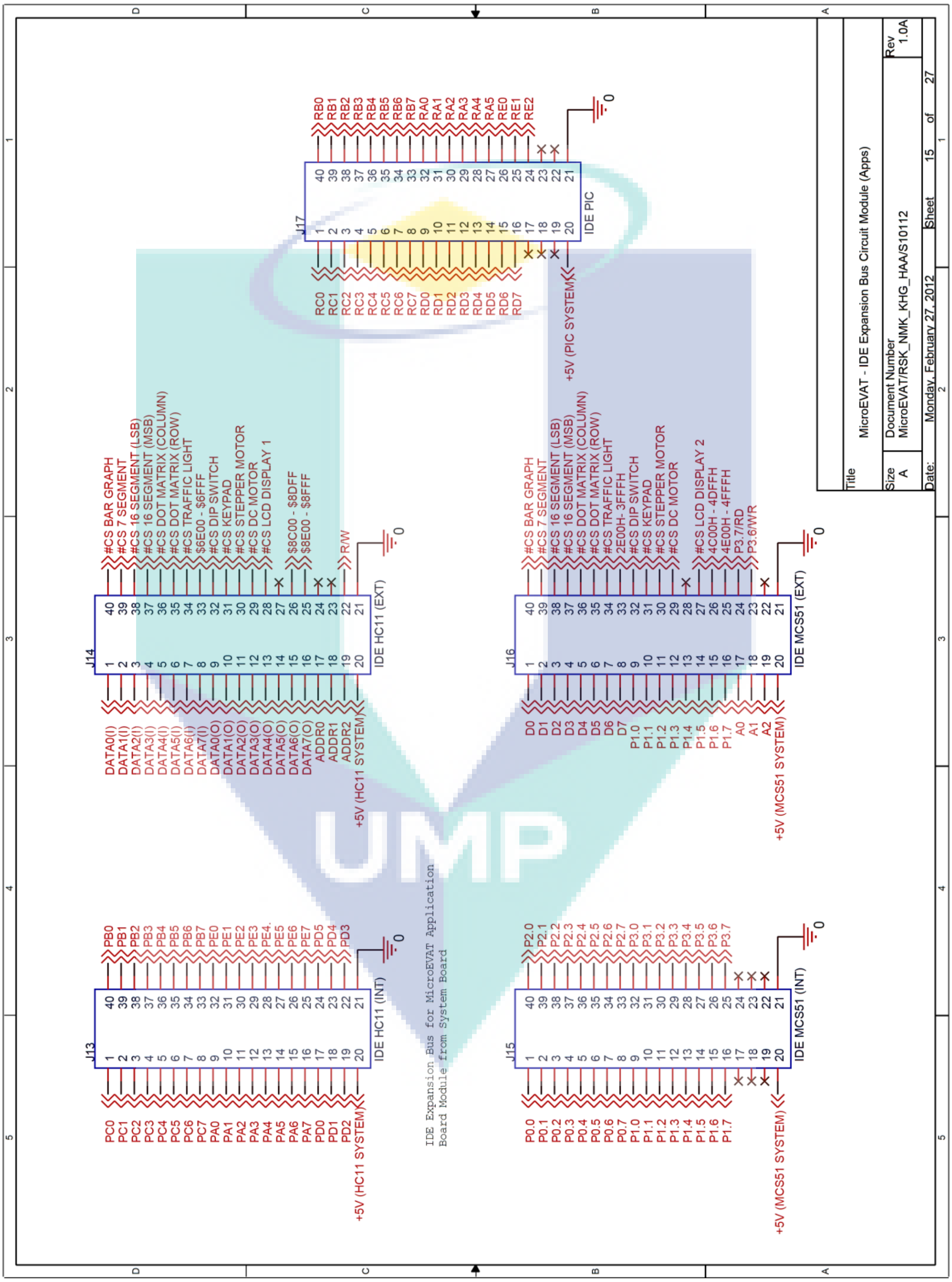
A13 External Memory Circuit Module



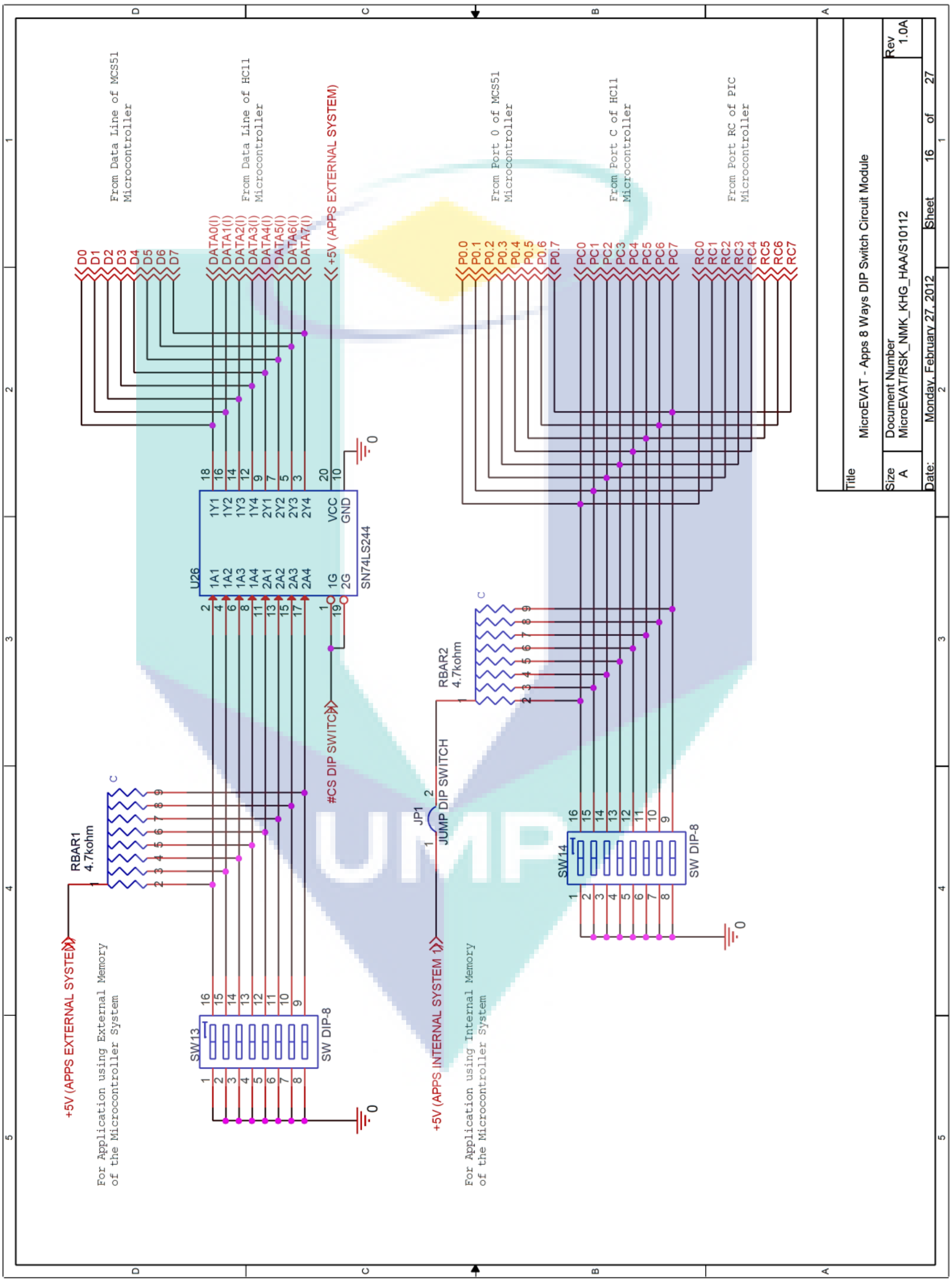
A14 IDE Expansion Bus Circuit Module (System)



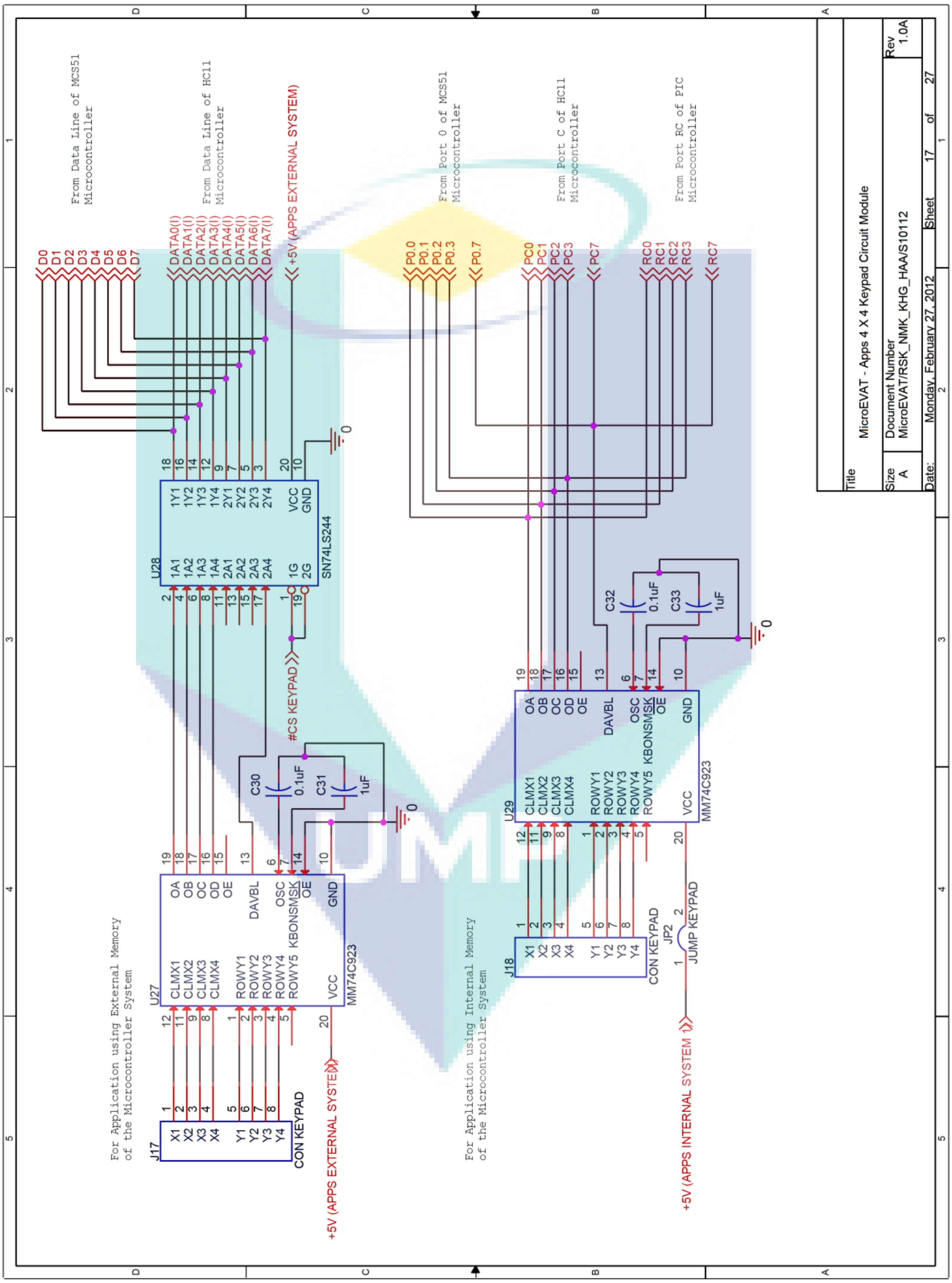
A15 IDE Expansion Bus Circuit Module (Application)



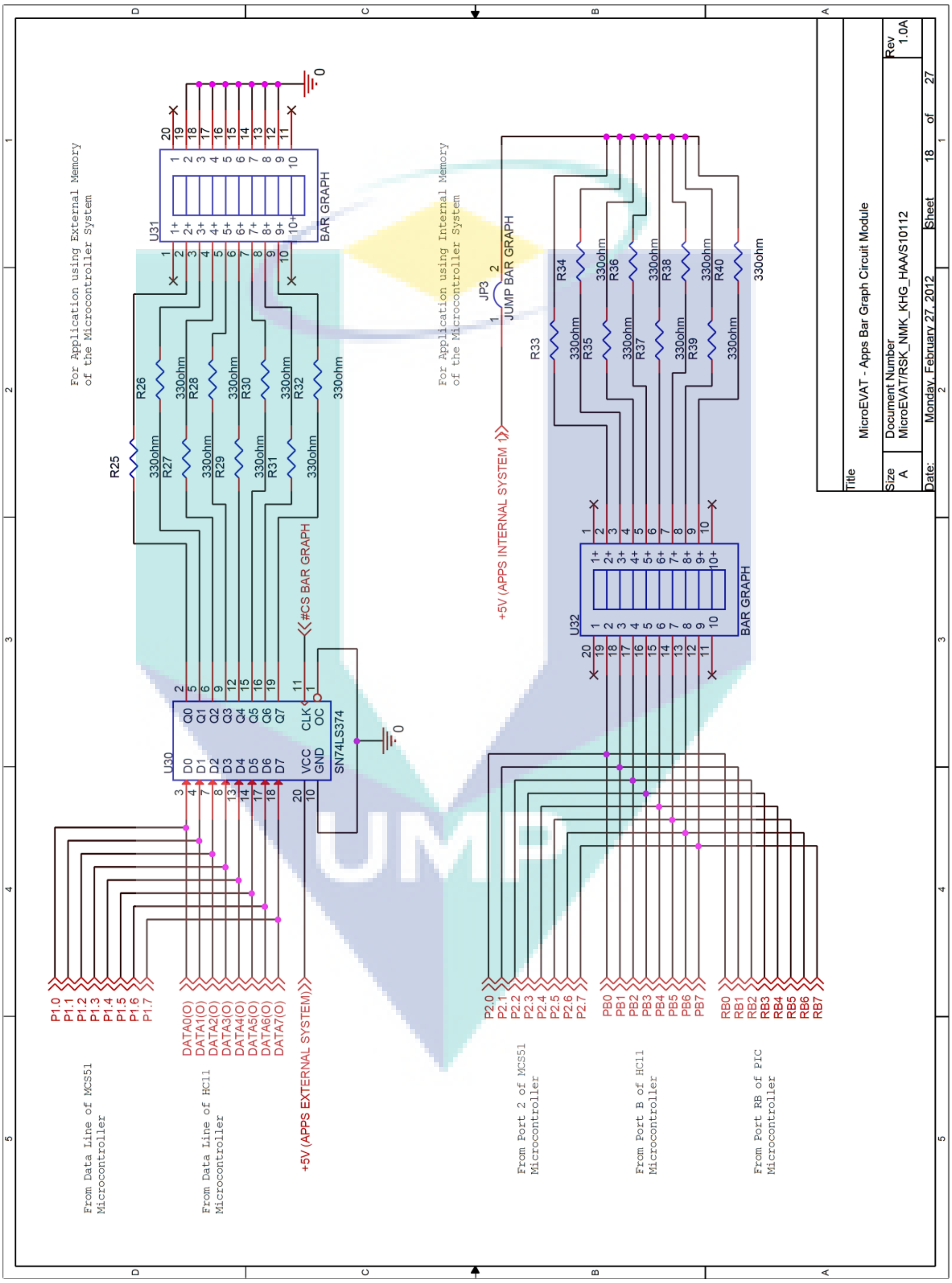
A16 8-Ways DIP Switch Circuit Module



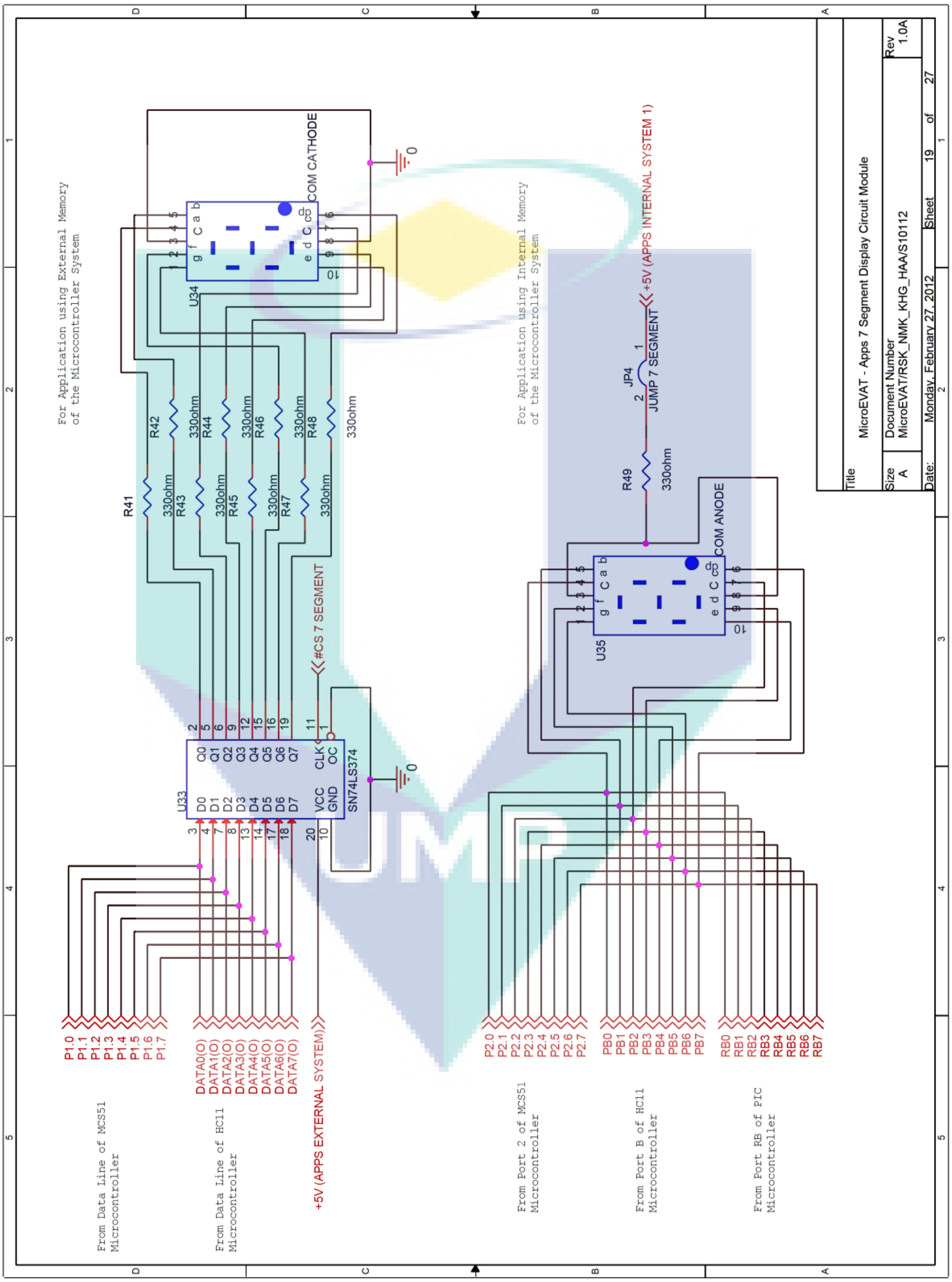
A17 4 X 4 Keypad Circuit Module



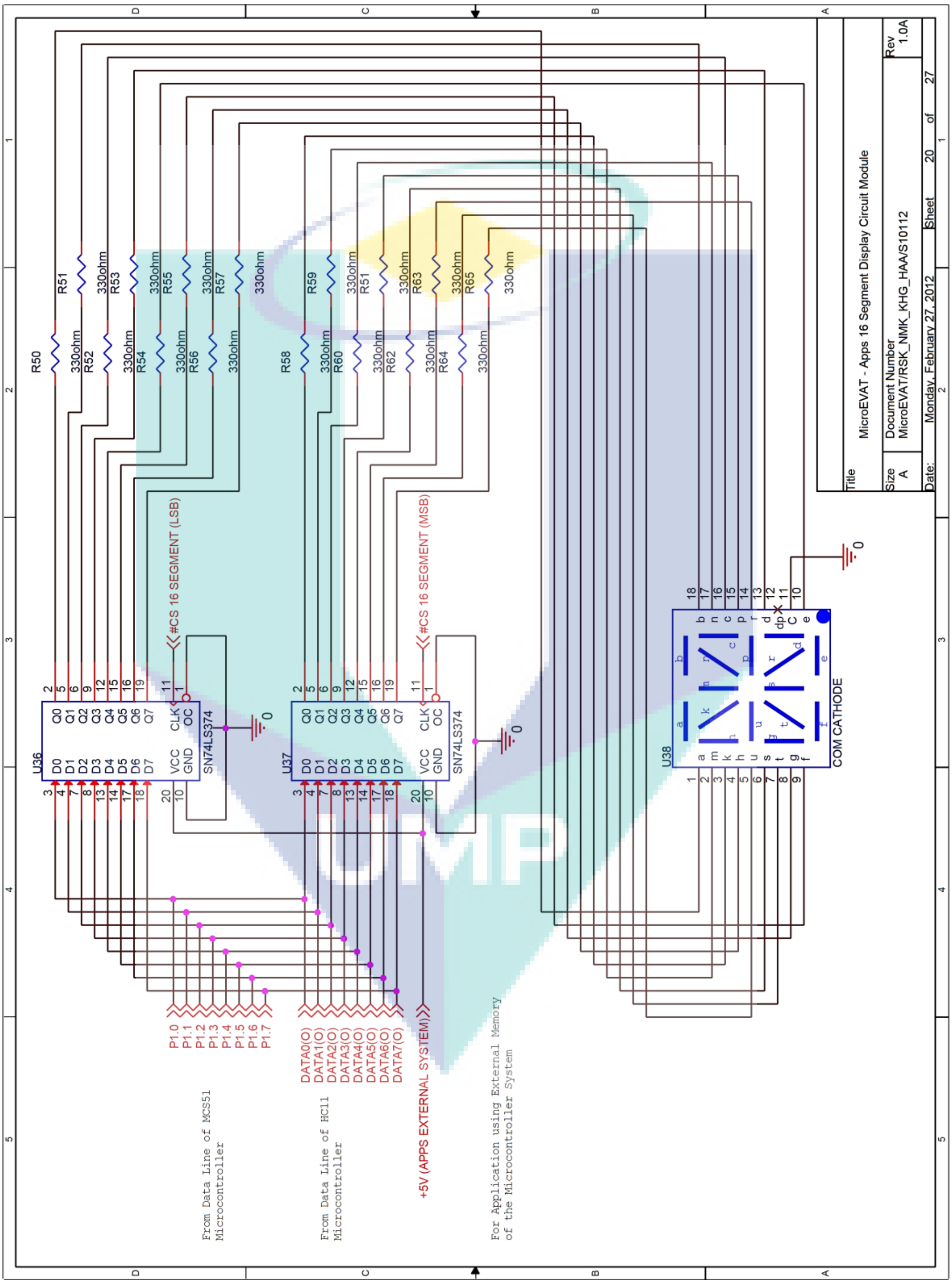
A18 Bar Graph Circuit Module



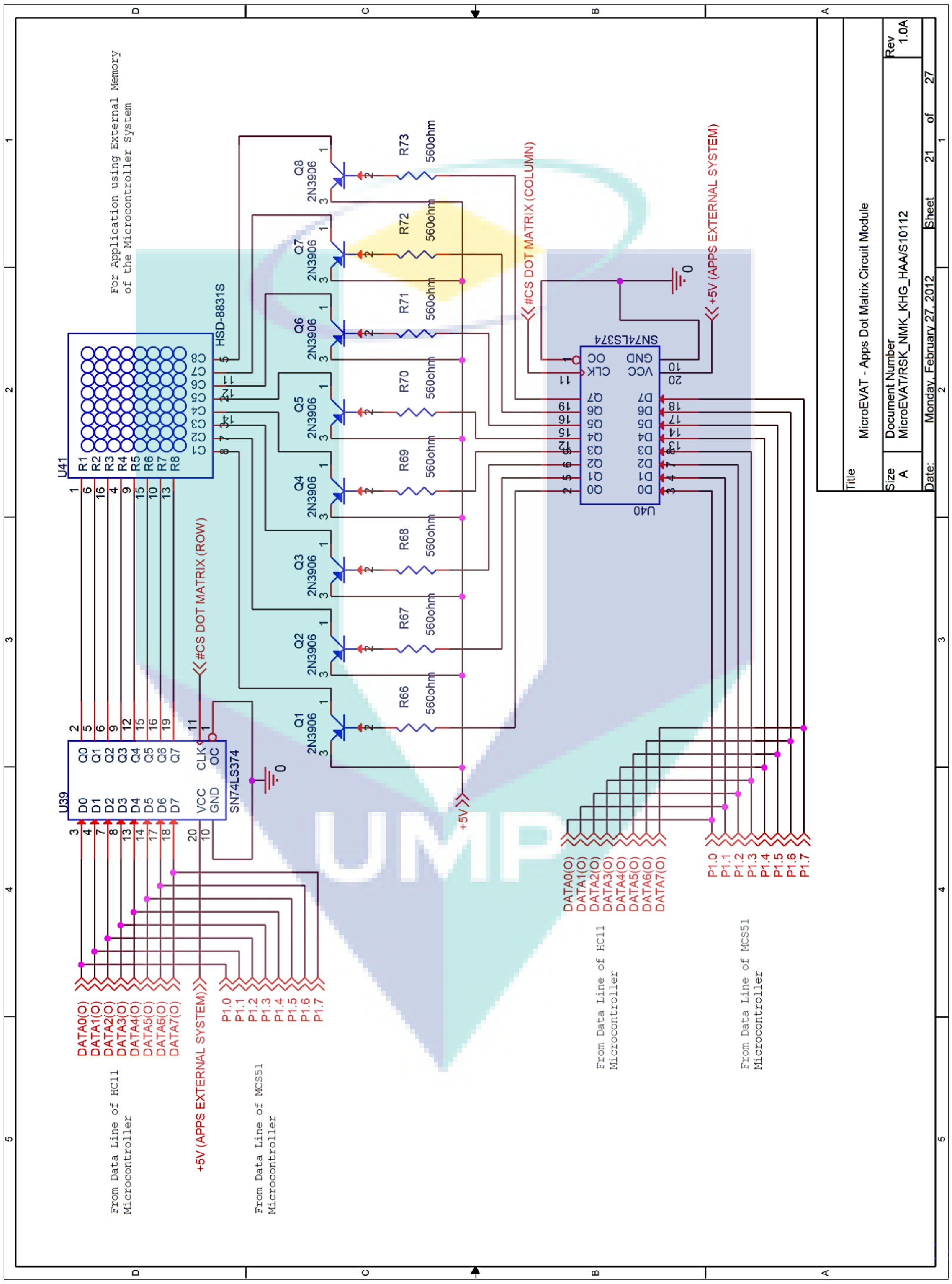
A19 7-Segment Display Circuit Module



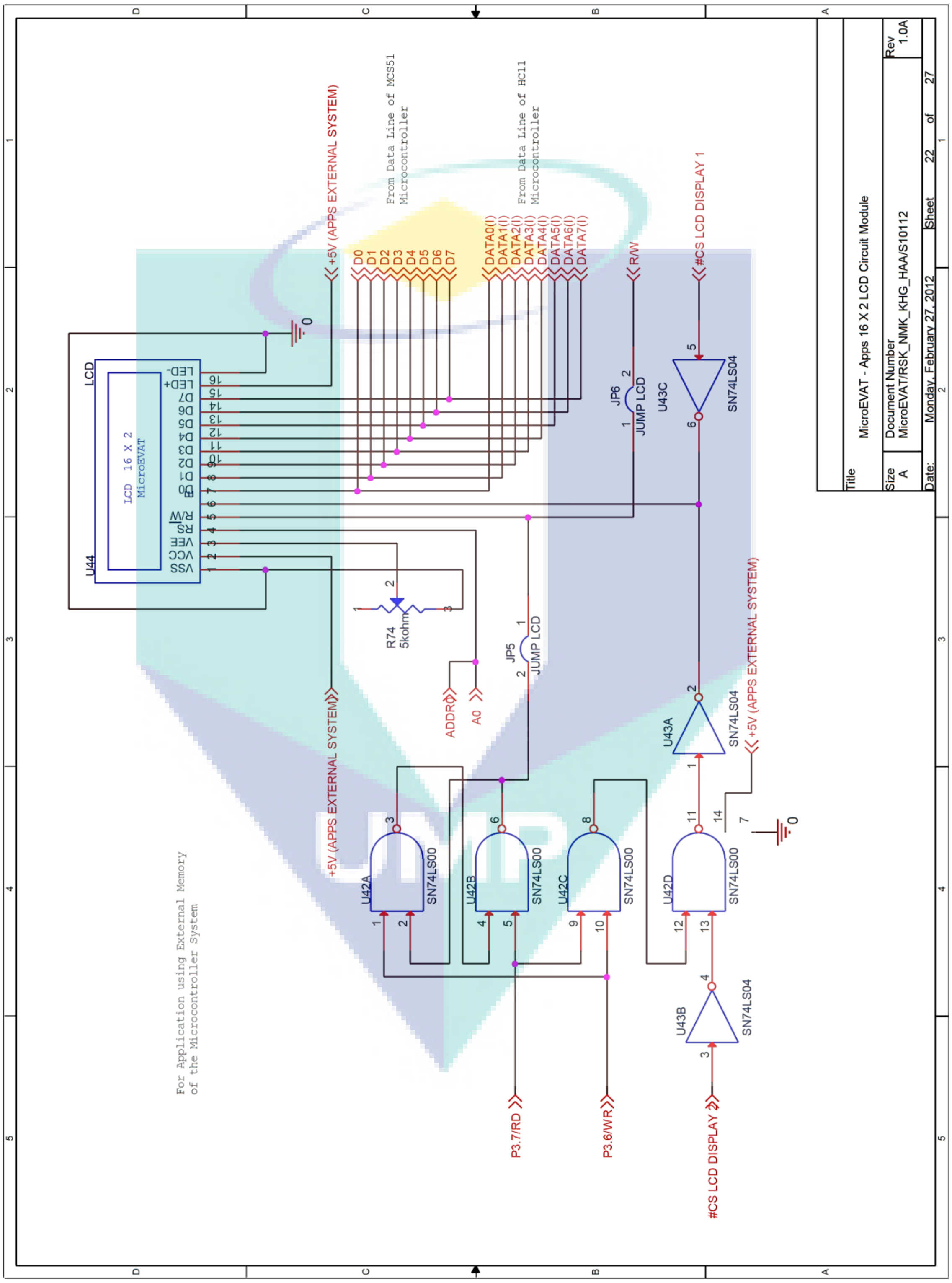
A20 16-Segment Display Circuit Module



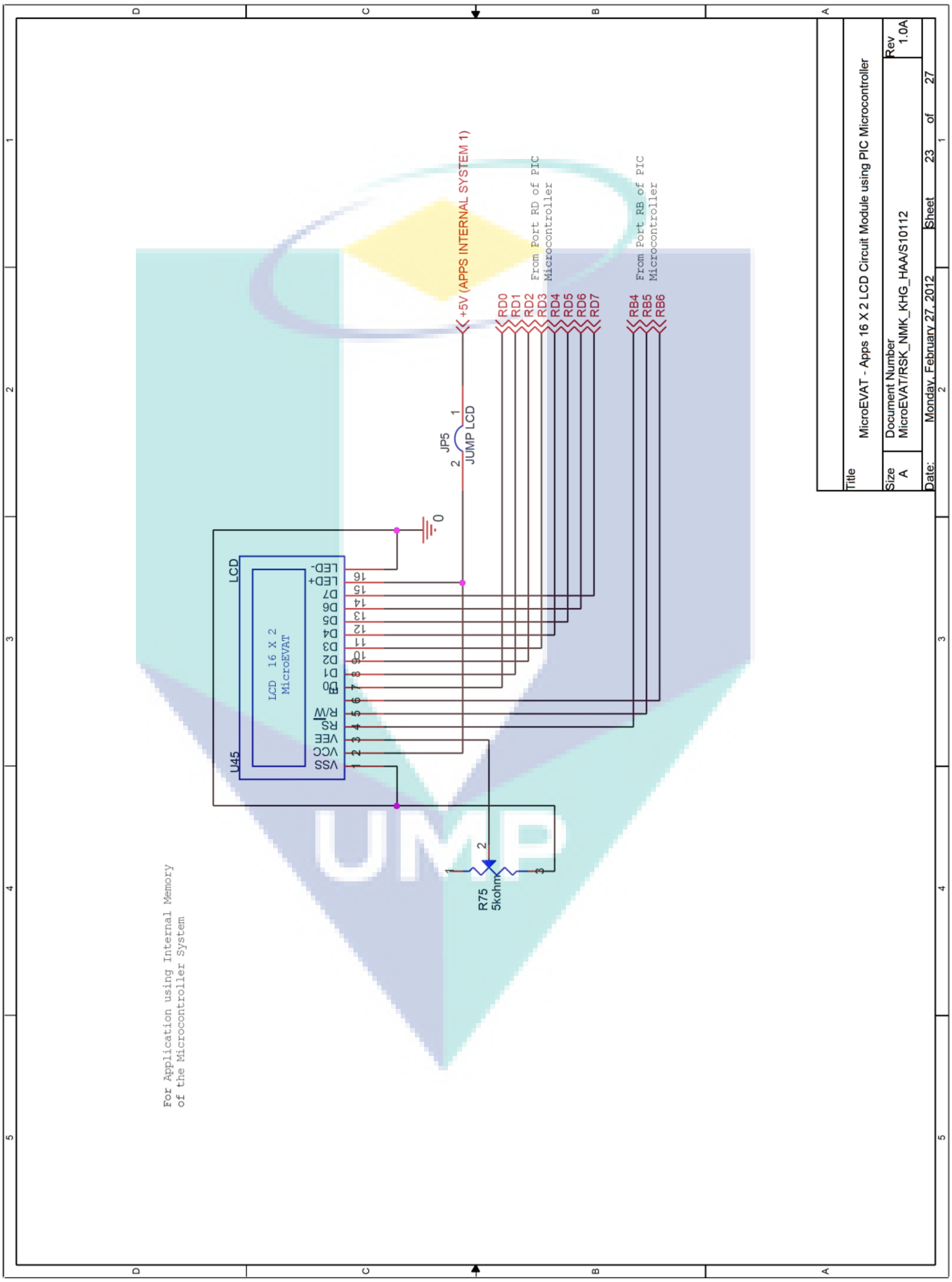
A21 8 X 8 Dot Matrix Display Circuit Module



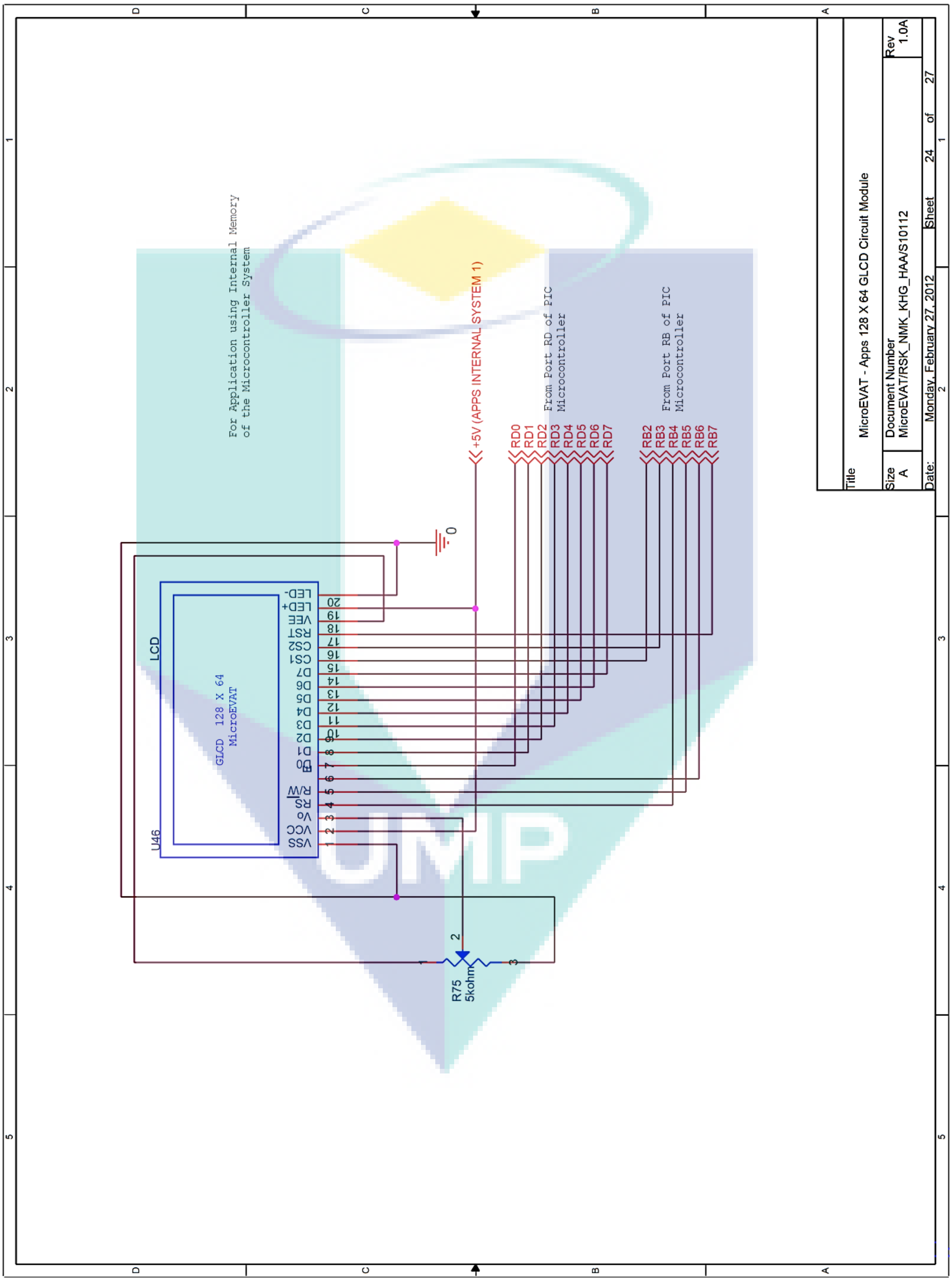
A22 16 X 2 LCD Circuit Module



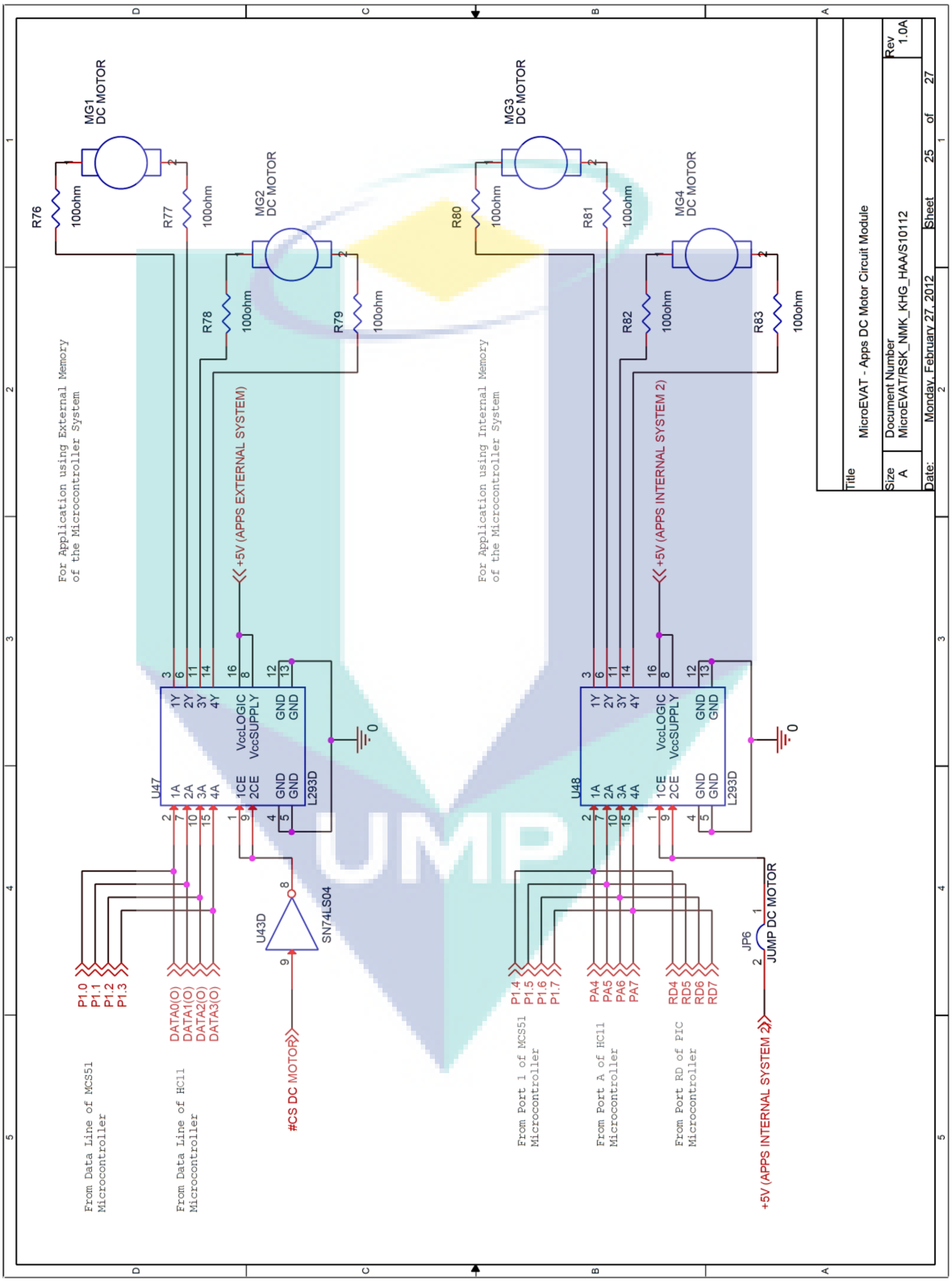
A23 16 X 2 LCD Circuit Module using PIC Microcontroller



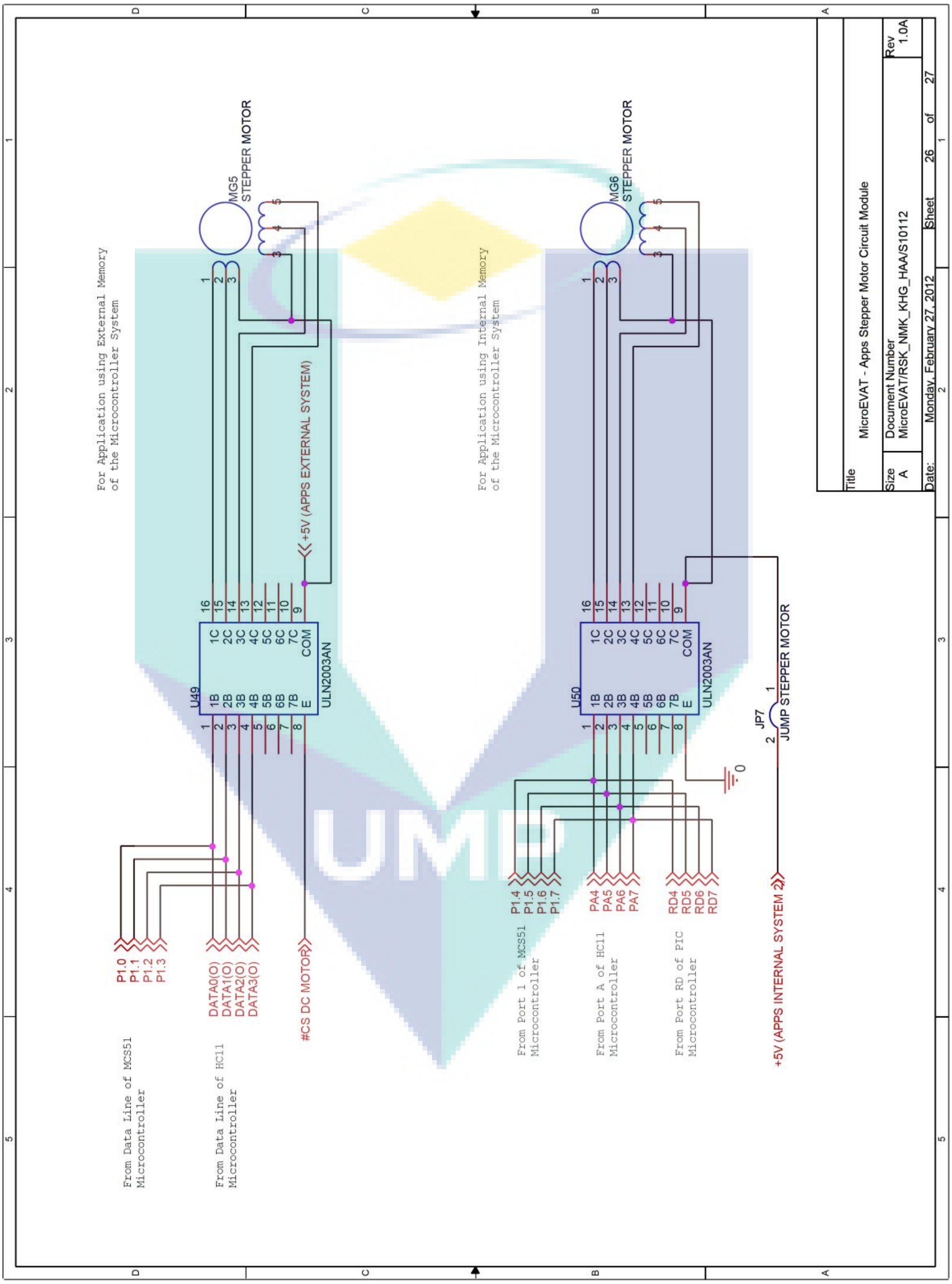
A24 128 X 64 Graphical LCD Circuit Module



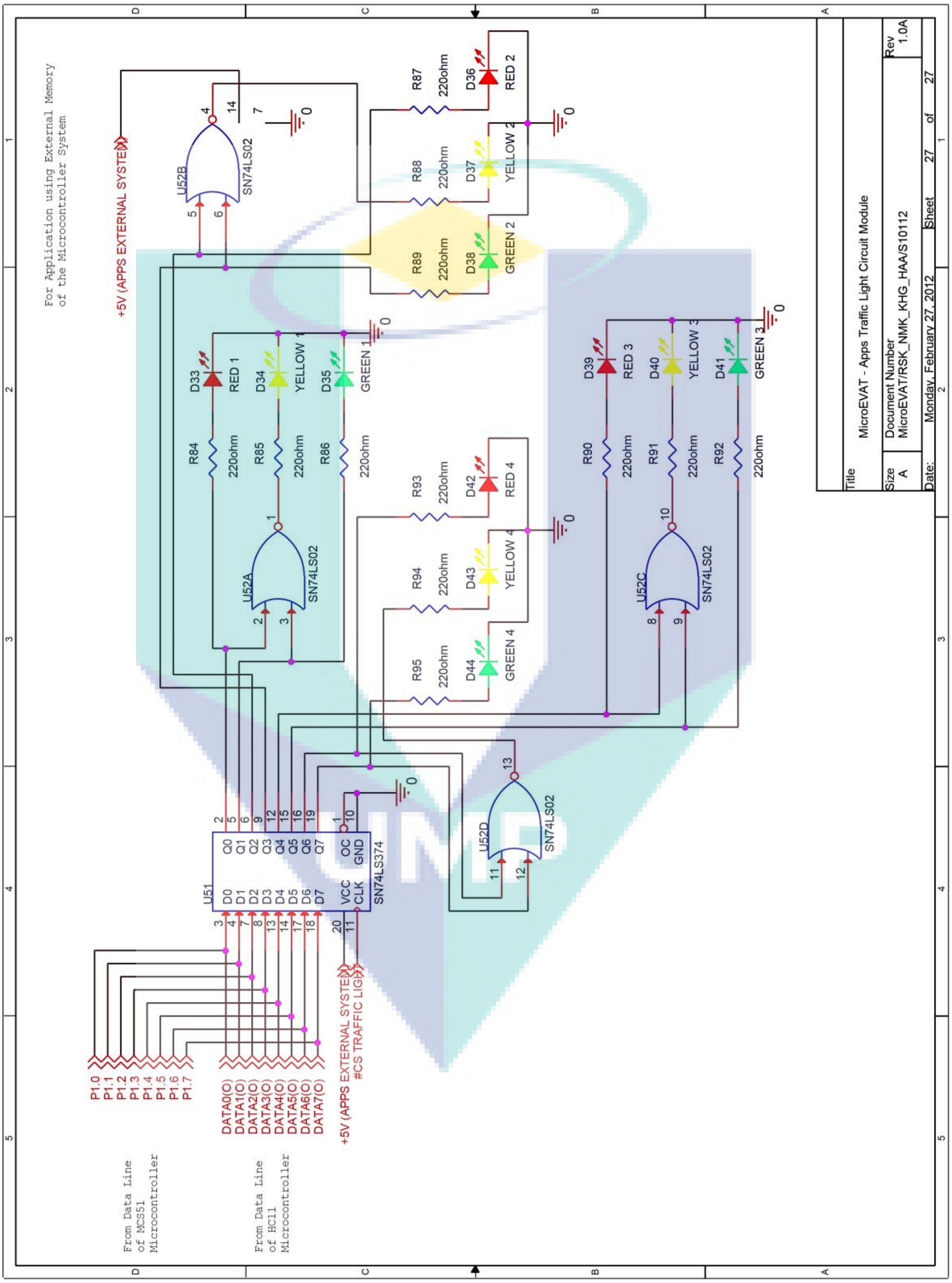
A25 DC Motor Circuit Module



A26 Stepper Motor Circuit Module



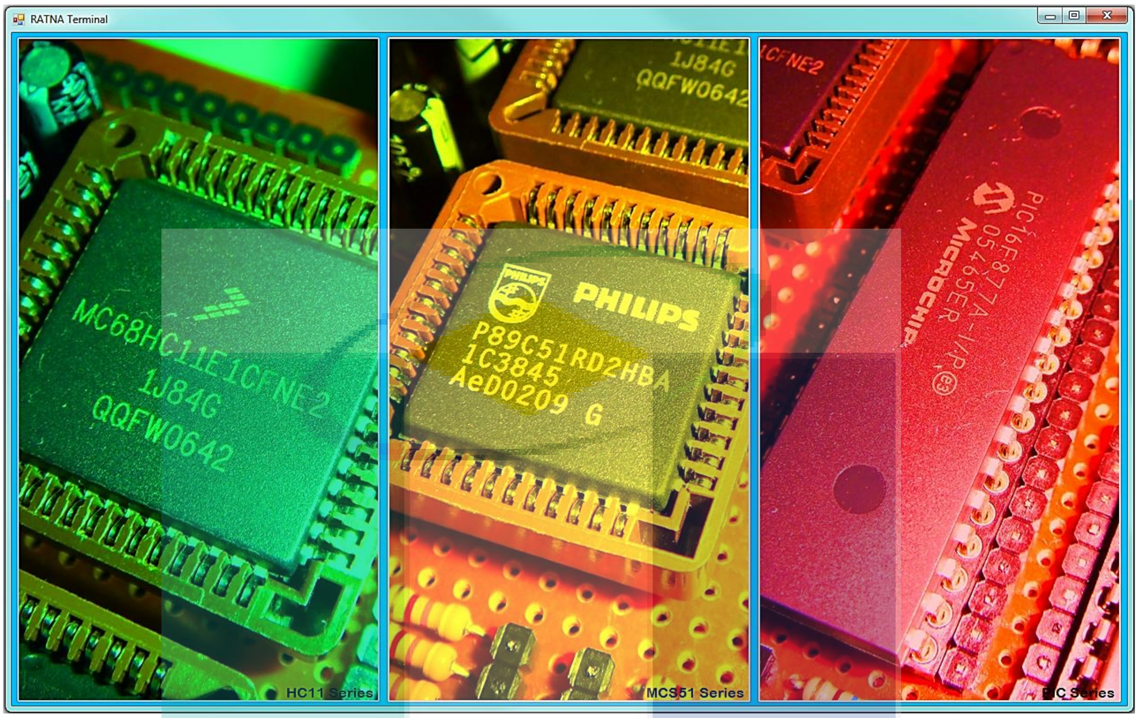
A27 'Traffic Light' Circuit Module



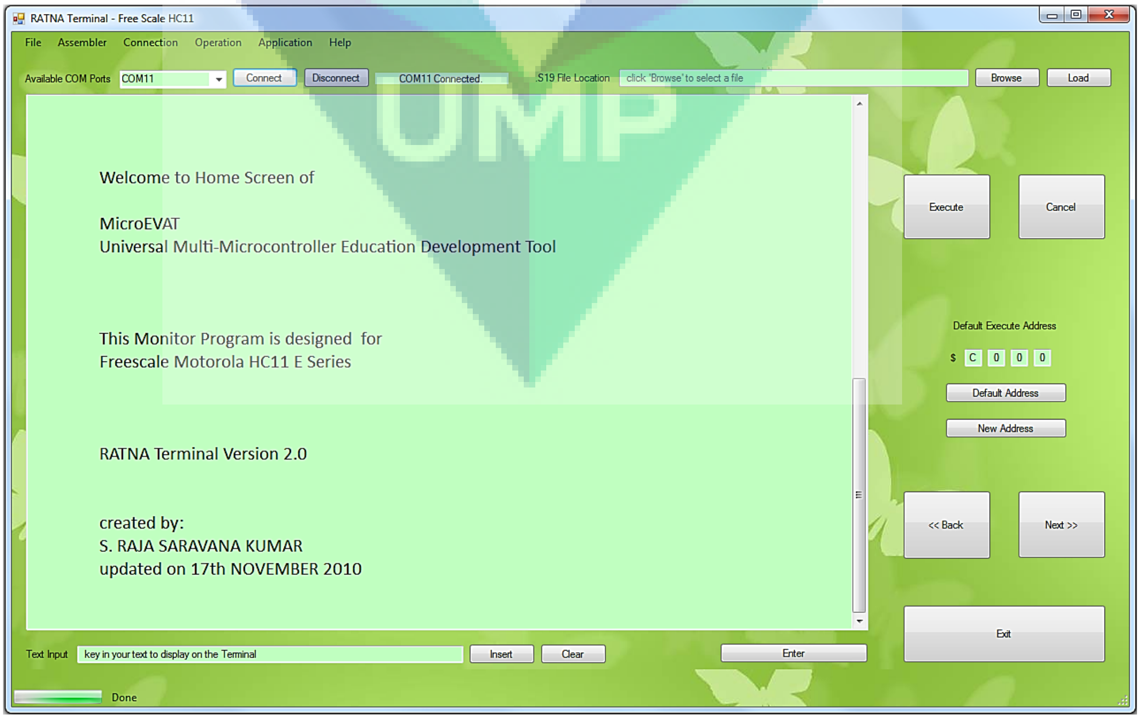
APPENDIX B

SCREEN SHOTS OF RATNA TERMINAL

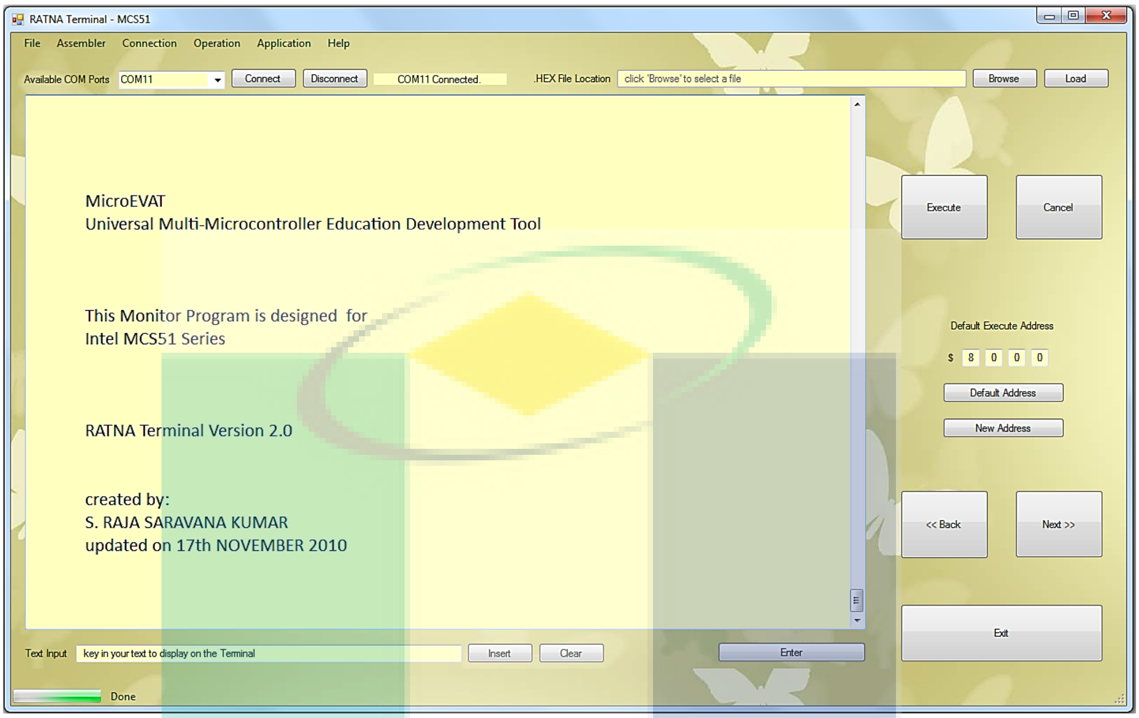
B1 Main Screen of RATNA Terminal



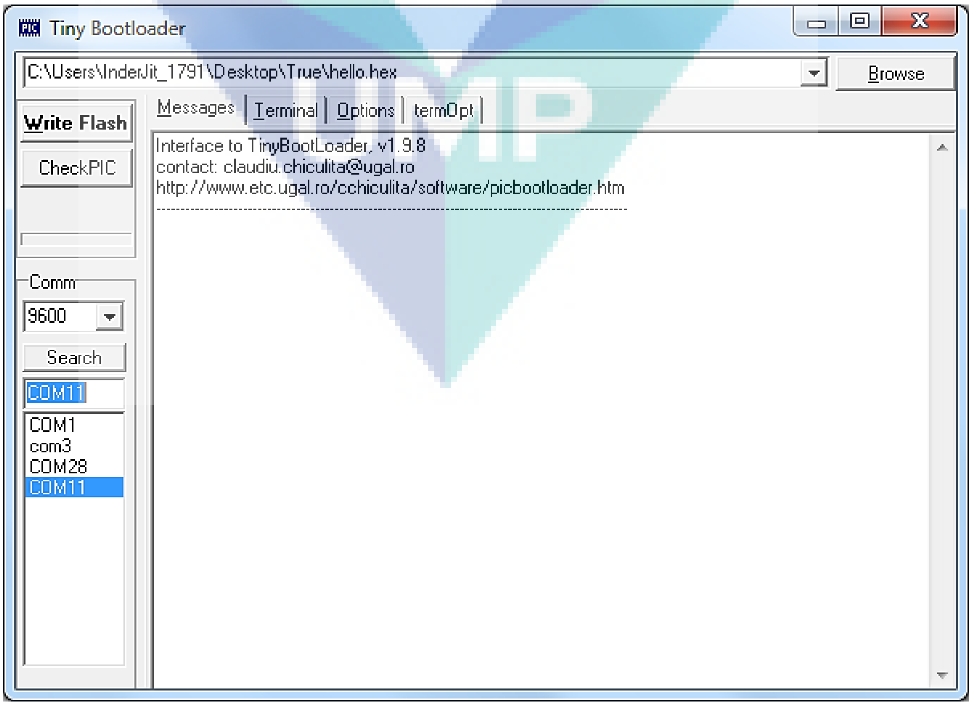
B2 RATNA Terminal for Freescale HC11 Microcontroller



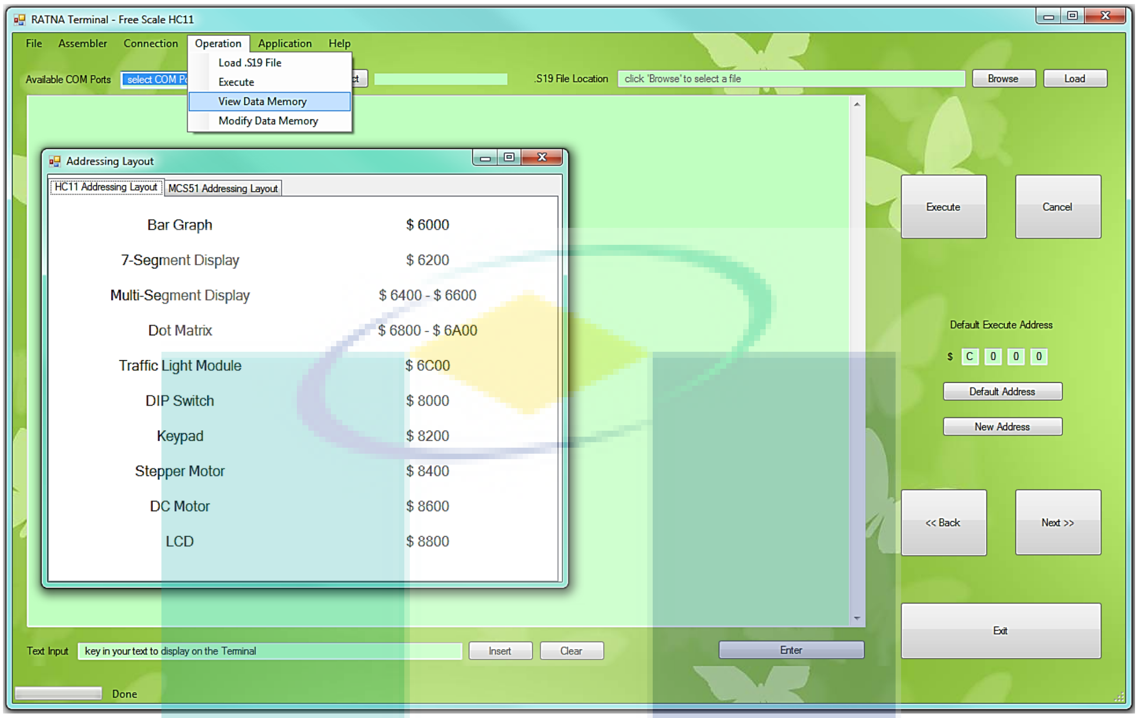
B3 RATNA Terminal for MCS51 Microcontroller



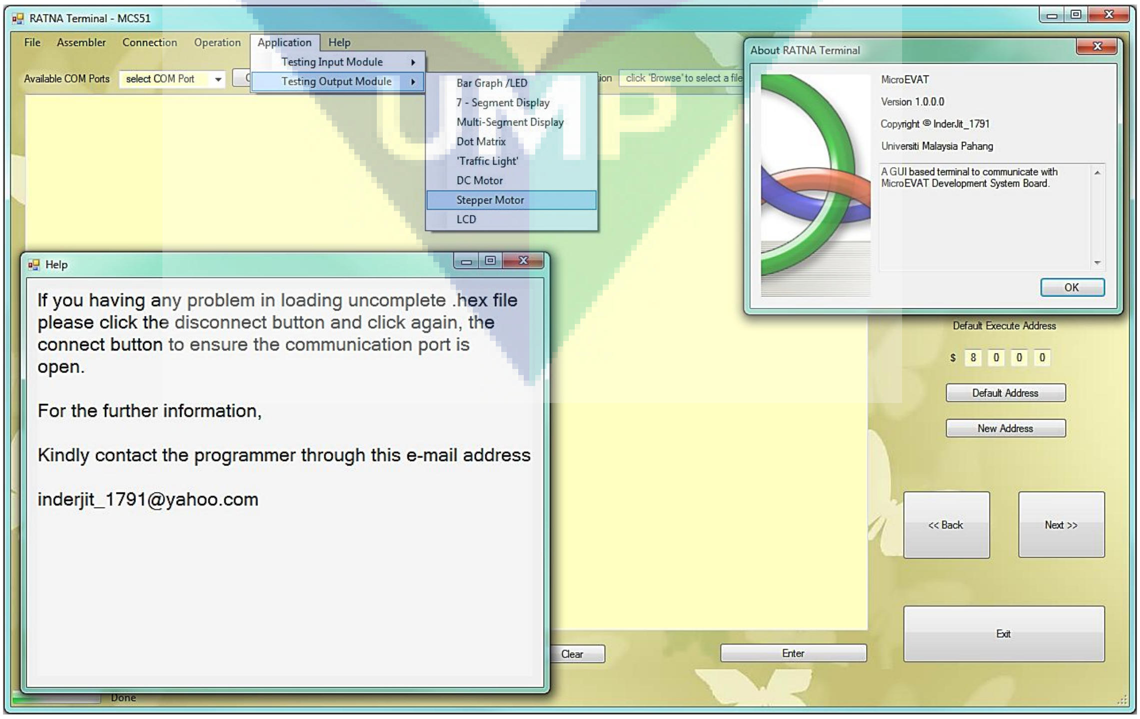
B4 RATNA Terminal with Tiny PIC Bootloader for PIC18 Microcontroller



B5 Basic Operation with List of Addressing Layout



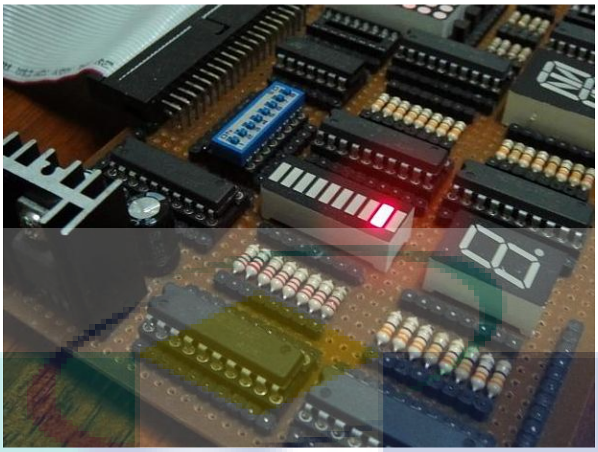
B6 Module Testing with Help Menu



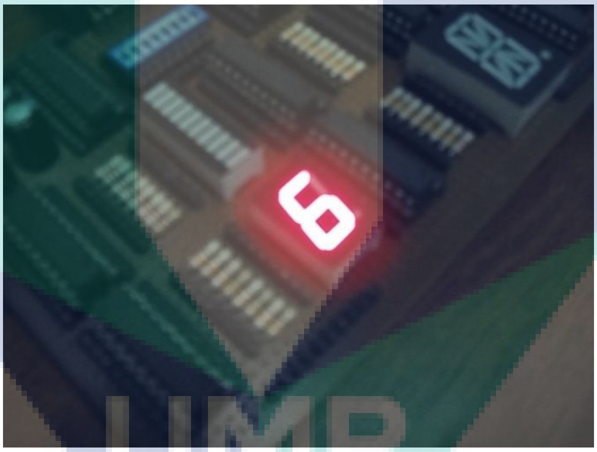
APPENDIX C

MICROEVAT'S APPLICATION MODULE TESTING

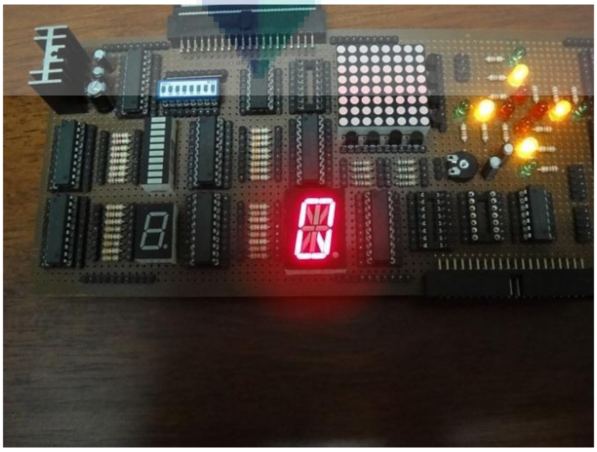
C1 Bar Graph Testing



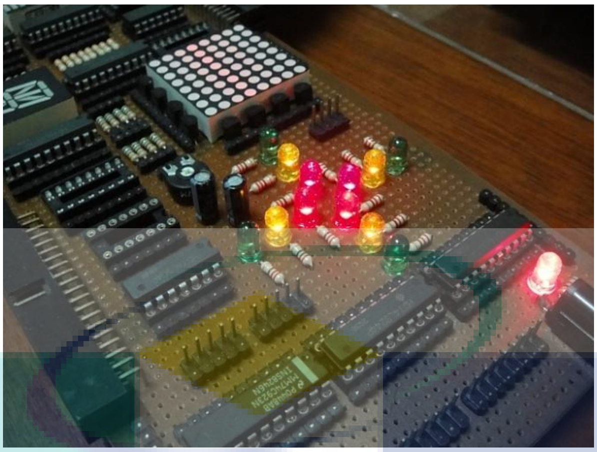
C2 7-Segment Display Testing



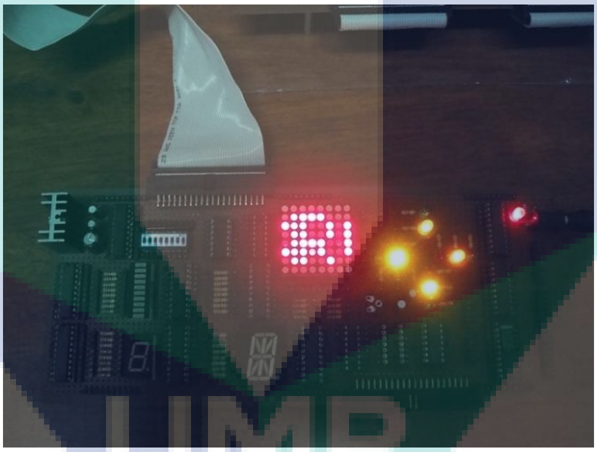
C3 16-Segment Display Testing



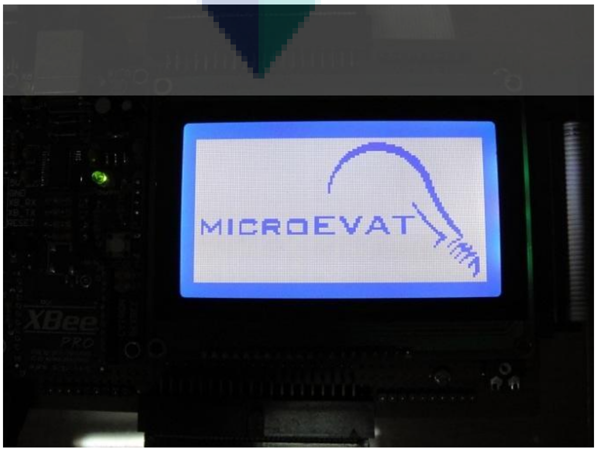
C4 ‘Traffic Light’ Module Testing



C5 Dot Matrix Display Testing



C6 Graphical LCD Module Testing



APPENDIX D

SURVEY

D1 Results for Survey – Question 1 until Question 3

Results for survey: Survey on student's understanding in Embedded systems

Page: 1/4

Students in Faculty of Electrical & Electronics Engineering, Universiti Malaysia Pahang are required to learn at least one of the architecture of any embedded system microcontroller either in class assessments or their final year projects (PSM). This survey were conducted to know the level of understanding in this course and the feedback from the students if there are develop in-house microcontroller educational development system based on the MCS51, HC11, and PIC 18 series microcontroller were developed

Please be feeling free to answer this questionaire

	Total	Percentage
Respondents :	152	100.00%

Question 1
Which course are you in?

	Total	Percentage
BEE Cohort 2008 / 2009	34	22.37%
BEE Cohort 2009 / 2010	58	38.16%
DEE Cohort 2008 / 2009	26	17.11%
DEE Cohort 2009 / 2010	34	22.37%

Question 2
Your current CGPA is?

	Total	Percentage
3.67 - 4.00	18	11.84%
3.00 - 3.66	61	40.13%
2.50 - 2.99	56	36.84%
2.00 - 2.49	17	11.18%
1.99 and lower	0	0.00%

Question 3
Have you taken Embedded Controller Technology subject?

	Total	Percentage
Yes	149	98.03%
No	3	1.97%

D2 Results for Survey – Question 4 until Question 7

Page: 2/4

Question 4
What type of microcontrollers that have you learned or currently use in any of your project?
(Please tick more than one if necessary)

	Total	Percentage
PIC 16 Series	1	0.38%
PIC 18 Series	95	36.54%
MCS51	72	27.69%
HC11	92	35.38%

Question 5
Which programming language that you good at?
(Please tick more than one if necessary)

	Total	Percentage
C language	54	35.53%
Basic Assembly Language	98	64.47%
Jscript	0	0.00%
VB Script	0	0.00%

Question 6
Which microcontroller that you learn or use for any of your class assessment?
(Please tick more than one if necessary)

	Total	Percentage
PIC 16 Series	0	0.00%
PIC 18 Series	0	0.00%
MCS51	72	43.90%
HC11	92	56.10%

Question 7
Have you been instructed to construct a project board based on microcontroller that you stated above?

	Total	Percentage
Yes	147	96.71%
No	5	3.29%

D3 Results for Survey – Question 8 until Question 11

Page: 3/4

Question 8
How you will describe if you learning any of these programming languages?
(Please rate the following matter, 1 star indicates the easiest, 5 star indicates the hardest)

	1	2	3	4	5
C language	5%	9%	27%	32%	27%
Basic Assembly Language	14%	5%	18%	23%	40%
Jscript	5%	0%	18%	0%	77%
VB Script	5%	5%	36%	14%	40%

Question 9
How you will describe if you learning any of these microcontrollers?
(Please rate the following matter, 1 star indicates the easiest, 5 star indicates the hardest)

	1	2	3	4	5
PIC 16 Series	9%	9%	41%	9%	32%
PIC 18 Series	5%	9%	27%	18%	41%
MCS51	0%	9%	9%	23%	59%
HC11	14%	5%	14%	36%	31%

Question 10
Did you construct any of the project using any microcontroller for your final year project?

	Total	Percentage
Yes	149	98.03%
No	3	1.97%

Question 11
If yes, which microcontroller that you use for your final year project?
(Please tick more than one if necessary)

	Total	Percentage
PIC 16 Series	4	2.44%
PIC 18 Series	95	57.93%
MCS51	11	6.71%
HC11	42	25.61%

D4 Results for Survey – Question 12 until Question 14

Page: 4/4

Question 12
Please rate the following matter, (1 star indicates the easiest, 5 star indicates the hardest)

	1	2	3	4	5
Understanding the architecture of microcontroller	5%	14%	63%	18%	0%
Understanding the programming of microcontroller	14%	9%	41%	32%	4%
Construct a basic system uding microcontroller	9%	14%	41%	27%	9%
Write a simple microcontroller program	9%	23%	14%	45%	9%
Using low end device as application in microcontroller system (Ex, LEDs, 7-Segment Display, and etc.)	18%	9%	23%	14%	36%
Write a microcontroller program using any of low end devices	14%	5%	27%	23%	31%
Using mid-range device as application in microcontroller system (Ex, LCDs, Motors, and etc.)	5%	5%	23%	32%	35%
Write a microcontroller program using any of mid-range devices	5%	9%	18%	23%	45%
Using wireless device as application in microcontroller system (Ex, ZigBee, Bluetooth, and etc.)	0%	0%	18%	18%	64%
Write a microcontroller program using any of wireless devices	5%	9%	9%	14%	63%

Question 13
What do you think if a training kit or educational system kit where develop to help student in understanding more about programming using any of these microcontroller?

	Total	Percentage
Yes, it really helpful	98	64.47%
Might overcome problem in programming	32	21.05%
Student will depends in this system	2	1.32%
Save time and money for class assessment	20	13.16%

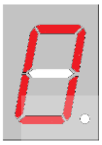
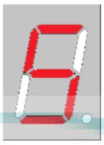
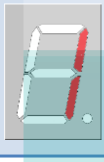
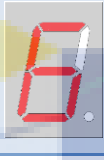
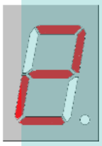
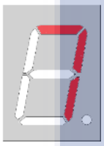
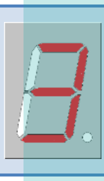
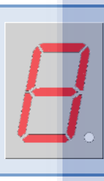
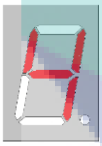
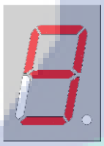
Question 14
If a traning kit has been developed, which are the important features that should have in this training kit?

	Total	Percentage
User-friendly, complete IDE	66	43.42%
Application module, extensive I/O	43	28.29%
Software module	42	27.63%
Other - a lot of assessment examples	1	0.66%

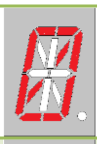
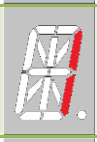
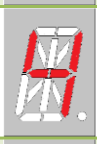
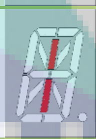
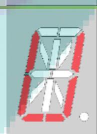
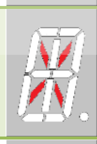
APPENDIX E

HEXADECIMAL TABLE FOR APPLICATION MODULE

E1 Binary coded Decimal (BCD) value for 7-Segment display

Display	BCD Value (Hex)		Display	BCD Value (Hex)	
	Cathode	Anode		Cathode	Anode
	**3F**	**C0**		**6D**	**92**
	**06**	**F9**		**7D**	**82**
	**5B**	**A4**		**07**	**F8**
	**4F**	**B0**		**7F**	**80**
	**66**	**99**		**67**	**98**

E2 Encoding Value for 16-Segment display

Display	Encoding		Display	Encoding		Display	Encoding	
	Upper	Lower		Upper	Lower		Upper	Lower
	44	FF		00	F3		00	FF
	00	0C		12	3F		88	C7
	88	77		88	F3		20	FF
	88	3F		80	C3		A8	C7
	88	8C		08	FB		88	BB
	A0	B3		88	CC		12	03
	88	FB		12	00		00	FC
	00	0F		00	7C		44	C0
	88	FF		A4	C0		60	CC
	88	8F		00	F0		65	00
	88	CF		05	CC		15	00
	1A	3F		21	CC		44	33

E3 The sequence of the 'Traffic Light' module (Top to Bottom)

R4	Y4	G4	R3	Y3	G3	R2	Y2	G2	R1	Y1	G1
1	0	0	1	0	0	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0	0	0	1
1	0	0	1	0	0	1	0	0	0	1	0
1	0	0	1	0	0	1	0	0	1	0	0
1	0	0	1	0	0	0	0	1	1	0	0
1	0	0	1	0	0	0	1	0	1	0	0
1	0	0	1	0	0	1	0	0	1	0	0
1	0	0	0	0	1	1	0	0	1	0	0
1	0	0	0	1	0	1	0	0	1	0	0
1	0	0	1	0	0	1	0	0	1	0	0
0	0	1	1	0	0	1	0	0	1	0	0
0	1	0	1	0	0	1	0	0	1	0	0

Note:

R4 - P1.6 / DATA6
Y4 - $\overline{P1.7 / DATA7 \oplus P1.6 / DATA6}$
G4 - P1.7 / DATA7
R3 - P1.4 / DATA4
Y3 - $\overline{P1.5 / DATA5 \oplus P1.4 / DATA4}$
G3 - P1.5 / DATA5
R2 - P1.2 / DATA2
Y2 - $\overline{P1.3 / DATA3 \oplus P1.2 / DATA2}$
G2 - P1.3 / DATA3
R1 - P1.0 / DATA0
Y1 - $\overline{P1.1 / DATA1 \oplus P1.0 / DATA0}$
G1 - P1.1 / DATA1

E4 The relation between data bits and Keypad

Key pressed	Data Byte (LSB to MSB)					
	OA	**OB**	**OC**	**OD**	**OE**	**DAVBL**
1	0	0	0	0	X	1
2	1	0	0	0	X	1
3	0	1	0	0	X	1
A	1	1	0	0	X	1
4	0	0	1	0	X	1
5	1	0	1	0	X	1
6	0	1	1	0	X	1
B	1	1	1	0	X	1
7	0	0	0	1	X	1
8	1	0	0	1	X	1
9	0	1	0	1	X	1
C	1	1	0	1	X	1
*****	0	0	1	1	X	1
0	1	0	1	1	X	1
#	0	1	1	1	X	1
D	1	1	1	1	X	1

E5 The relation between data bits and Stepper Motor input

Direction	ULN2003AN Input			
	1B	**2B**	**3B**	**4B**
1st step	1	1	0	0
2nd step	0	1	1	0
3rd step	0	0	1	1
4th step	1	0	0	1

APPENDIX F

ADDRESS ALLOCATION FOR MICROEVAT SYSTEM

F1 Address Allocation for Freescale HC11 and MCS51 microcontroller

<table>
<tr><th colspan="4">Master (Freescale HC11)</th><th colspan="4">Slave (Freescale HC11)</th><th rowspan="2">Address Range</th><th rowspan="2">Description</th></tr>
<tr><th>A15</th><th>A14</th><th>A13</th><th>Output</th><th>A11</th><th>A10</th><th>A9</th><th>Output</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>Y0*</td><td>X</td><td>X</td><td>X</td><td></td><td>$0000 - $1FFF</td><td>Internal Register & Control Bit for HC11</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Y1*</td><td>X</td><td>X</td><td>X</td><td></td><td>$2000 - $3FFF</td><td>Future Expansion</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Y2*</td><td>X</td><td>X</td><td>X</td><td></td><td>$4000 - $5FFF</td><td>Future Expansion</td></tr>
<tr><td rowspan="8">0</td><td rowspan="8">1</td><td rowspan="8">1</td><td rowspan="8">Y3*</td><td>0</td><td>0</td><td>0</td><td>Y0*</td><td>$6000 - $61FF</td><td>Chip Select
BAR GRAPH</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Y1*</td><td>$6200 - $63FF</td><td>Chip Select
7 SEG DISPLAY</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Y2*</td><td>$6400 - $65FF</td><td>Chip Select
16 SEG (LSB)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Y3*</td><td>$6600 - $67FF</td><td>Chip Select
16 SEG (MSB)</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Y4*</td><td>$6800 - $69FF</td><td>Chip Select
DOT MATRIX (C)</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Y5*</td><td>$6A00 - $6BFF</td><td>Chip Select
DOT MATRIX (R)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Y6*</td><td>$6C00 - $6DFF</td><td>Chip Select
TRAFFIC LIGHT</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Y7*</td><td>$6E00 - $6FFF</td><td>Future Expansion</td></tr>
<tr><td rowspan="8">1</td><td rowspan="8">0</td><td rowspan="8">0</td><td rowspan="8">Y4*</td><td>0</td><td>0</td><td>0</td><td>Y0*</td><td>$8000 - $81FF</td><td>Chip Select
DIP SWITCH</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>Y1*</td><td>$8200 - $83FF</td><td>Chip Select
KEYPAD</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Y2*</td><td>$8400 - $85FF</td><td>Chip Select
STEPPER MOTOR</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Y3*</td><td>$8600 - $87FF</td><td>Chip Select
DC MOTOR</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>Y4*</td><td>$8800 - $89FF</td><td>Chip Select
LCD</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Y5*</td><td>$8A00 - $8BFF</td><td>Reserved for LCD
(MCS51)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Y6*</td><td>$8C00 - $8DFF</td><td>Future Expansion</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Y7*</td><td>$8E00 - $8FFF</td><td>Future Expansion</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>Y5*</td><td>X</td><td>X</td><td>X</td><td></td><td>$A000 - $BFFF</td><td>Bootstrap Mode
($B600 - $B7FF)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>Y6*</td><td>X</td><td>X</td><td>X</td><td></td><td>$C000 - $DFFF</td><td>Chip Select RAM</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>Y7*</td><td>X</td><td>X</td><td>X</td><td></td><td>$E000 - $FFFF</td><td>Chip Select
EEPROM</td></tr>
</table>

F2 Address Allocation for Freescale HC11 and MCS51 microcontroller (Continued)

Master (MCS51)				Slave (MCS51)				Address	Description
A15	A14	A13	Output	A11	A10	A9	Output		
0	0	0	Y0*	X	X	X		0000H - 1FFFH	Chip Select EEPROM
0	0	1	Y1*	0	0	0	Y0*	2000H - 21FFH	Chip Select BAR GRAPH
				0	0	1	Y1*	2200H - 23FFH	Chip Select 7 SEG DISPLAY
				0	1	0	Y2*	2400H - 25FFH	Chip Select 16 SEG (LSB)
				0	1	1	Y3*	2600H - 27FFH	Chip Select 16 SEG (MSB)
				1	0	0	Y4*	2800H - 29FFH	Chip Select DOT MATRIX (C)
				1	0	1	Y5*	2A00H - 2BFFH	Chip Select DOT MATRIX (R)
				1	1	0	Y6*	2C00H - 2DFFH	Chip Select TRAFFIC LIGHT
				1	1	1	Y7*	2E00H - 2FFFH	Future Expansion
0	1	0	Y2*	0	0	0	Y0*	4000H - 41FFH	Chip Select DIP SWITCH
				0	0	1	Y1*	4200H - 43FFH	Chip Select KEYPAD
				0	1	0	Y2*	4400H - 45FFH	Chip Select STEPPER MOTOR
				0	1	1	Y3*	4600H - 47FFH	Chip Select DC MOTOR
				1	0	0	Y4*	4800H - 49FFH	Reserved for LCD (HC11)
				1	0	1	Y5*	4A00H - 4BFFH	Chip Select LCD
				1	1	0	Y6*	4C00H - 4DFFH	Future Expansion
				1	1	1	Y7*	4E00H - 4FFFH	Future Expansion
0	1	1	Y3*	X	X	X		6000H - 7FFFH	Future Expansion
1	0	0	Y4*	X	X	X		8000H - 9FFFH	Chip Select RAM
1	0	1	Y5*	X	X	X		A000H - BFFFH	Future Expansion
1	1	0	Y6*	X	X	X		C000H - DFFFH	Future Expansion
1	1	1	Y7*	X	X	X		E000H - FFFFH	Boot ROM (FC00H - FFFFH)

Note:

Legend	Descriptions
	Output from Slave's address decoder not valid
Y0* - Y7*	Output from address decoder
X	Don't care (Logic either '0' or '1')
Future Expansion	For use to upgrade the MicroEVAT system
LSB	Least Significant Bit
MSB	Most Significant Bit
C	Column Address
R	Row Address

APPENDIX G

LIST OF PUBLICATIONS

F1 LIST OF CONFERENCES

1. R.S.K. Selvakumar, N.M.K. Nik Yusoff, A.N. Abd Alla, A New Development of Multi-application of Micro-controller through educational Training Systems, *Malaysian Technical Universities Conference on Engineering and Technology 2009 (MUCEET'09)*, Kuantan, Pahang, 20-22 June 2009.

2. R.S.K. Selvakumar, N.M.K. Nik Yusoff, H. Abdul Aziz, MCS51® Educational Development System with IDE Approach, *International Conference on Software Engineering and Computer Science 2009 (ICSECS'09)*, Kuantan, Pahang, 21-23 Sept 2009.

3. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, Designing A Modular-Approach Of Multi-Microcontroller Based Educational Training System With IDE, *National Conference On Postgraduate Research (NCON-PGR) 2009*, Kuantan, Pahang, 1 Oct 2009.

4. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, Designing of Multi-Microcontroller Training System, *2nd International Conference on Electronic Computer Technology (ICECT 2010)*, Kuala Lumpur, Malaysia, 7 – 8 May 2010.

5. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, Designing of a Multi-Microcontroller Hardware Platform for Academic Environment: The UMP-EVT, *Malaysian Technical Universities Conference on Engineering and Technology 2010 (MUCET 2010)*, Malacca, Malaysia, 28- 30 June 2010.

6. N.M.K. Nik Yusoff, H. Abdul Aziz, M.Z.B.M. Sapein, R.S.K. Selvakumar, K.H. Ghazali, MINI 11 – Microcontroller Laboratory Hardware Platform with IDE Approach, *Malaysian Technical Universities Conference on Engineering and Technology 2010 (MUCET 2010)*, Malacca, Malaysia, 28 - 30 June 2010.

7. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, A New Development of Modular-Approach Multi-Microcontroller Based Educational Training System, *National Postgraduate Seminar 2010 (NaPaS'2010)*, Shah Alam, Selangor, 6 – 7 July 2010.

8. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, Designing of an Embedded Flexible Multi-Microcontroller Based Training System, *International Conference on Advanced Infocomm Technology 2010 (ICAIT'10)*, Hainan, China, 20 – 22 July 2010.

9. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, Designing of Flexible Multi-Microcontroller Training System, *3rd International Conference on Computer and Electrical Engineering (ICCEE 2010)*, Chengdu, China, 16 – 18 Nov 2010.

10. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, MCS51® Evaluation Tool With IDE Approach For Learning Environment – The UMP-MET, *2010 International Conference on Embedded System & Microprocessor (ICESM 2010)*, Manila, Philippines, 4-5 December 2010

11. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, Multi-Microcontroller Evaluation Tool For Education In Embedded Control Technology: *The UMP-EVT, 2010 International Conference on Embedded System & Microprocessor (ICESM 2010)*, Manila, Philippines, 4-5 December 2010

12. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, Learning Embedded Systems With Multi-Microcontroller Evaluation Tool, *International Conference on Electrical, Control and Computer Engineering 2011 (InECCE 2011)*, Kuantan, Pahang, 21 – 22 June 2011

13. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, Framework of Designing Multiple Microcontrollers based Applications, *United Kingdom-Malaysia-Ireland Engineering Science Conferences 2011 (UMIES 2011)*, Universiti Malaya, Kuala Lumpur, 12 - 14 July 2011

F2 LIST OF JOURNALS

1. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, Designing of Flexible Multi-Microcontroller Based Training System, *Global Conference in Science and Engineering 2011 (GCSE 2011)*, Dubai, U.A.E, 28-30 December 2011. - To be published in *Procedia Engineering Journal*, (ISSN: 1877-7058, *ELSEVIER*)

2. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, Framework of Multi-Microcontroller Evaluation Tool for a use of Academic Environment, *International Journal of Advanced Computer Science (IJACSci)*. - To be published in International Journal of Advanced Computer Science (IJASci), (ISSN: 2251-6379)

F3 LIST OF AWARDS

1. N.M.K. Nik Yusoff, A.N. Abd Alla, R.S.K. Selvakumar, Development of Multi-Application of Microcontroller through Educational Training System, 20th International Invention, Innovation and Technology Exhibition 2009 (ITEX '2009), Kuala Lumpur Convention Centre, Malaysia, 15th – 17th May 2009 - Silver Medal

2. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, MicroEVAT – Multi-Microcontroller Development Board for SCL Approach, POSTAC Postgraduate Poster Competition 2011, Universiti Malaysia Pahang, Kuantan, Pahang, 7 - 9th April 2011 - Winner Medal

3. R.S.K. Selvakumar, K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, µ-EVAT Universal Development / Evaluation Tool For Microcontrollers, Creation, Innovation, Technology & Research Exposition (CITRex 2011), Universiti Malaysia Pahang, Kuantan, Pahang, 13 - 14th April 2011 - Silver Medal

4. K.H. Ghazali, N.M.K. Nik Yusoff, H.A.Aziz, R.S.K. Selvakumar, MicroEVAT - Multi-Microcontroller Development Board for SCL Approach, *22nd International Invention, Innovation and Technology Exhibition 2011 (ITEX 2011)*, Kuala Lumpur Convention Centre, Malaysia, 20th – 22nd May 2011 - Bronze Medal