PROBLEMS and INTENDED SOLUTIONS in TEACHING PIC MICROCONTROLLER in EET PROGRAM

Muhammad M. Baig
mailto: baigm@nsula.edu
Assistant Professor
Dept. of Engineering Technology
Northwestern State University
Natchitoches, LA 71497

Dr. Rafiqul Islam
mailto: islamr@nsula.edu
Associate Professor
Dept. of Engineering Technology
Northwestern State University
Natchitoches, LA 71497

Abstract

Microcontrollers have become an integral part in the areas of engineering and technology. These are being used in almost every application that requires a certain amount of intelligence. The application areas are expanded to industrial, medical, automotive, battery powered small devices and consumer electronics. This reinforces the essence of making sure that all engineering and technology students receive a basic understanding of microcontrollers. The students need to view it as a crucial hardware component. The industries are increasingly seeking graduates with appropriate background and training in microcontroller technology. Knowledge in this vital field will broaden the undergraduate experience of technology and significantly enhance the students' employment opportunities.

Viewing the fact that there is and will be an ever-increasing demand for engineers and technologists in this field, Northwestern State University, at Natchitoches, Louisiana, introduced a one semester (3 credit-hour class and 1 credit-hour lab) course on Microcontrollers. In both our instruction and lab experiments we have selected the Microchip's PIC 16 & PIC18 microcontroller family devices, namely 16F84A, 16F54, 16F870 and 18F452 for obvious reasons. We have attempted to provide our graduates adequate knowledge to design, test and analyze basic microcontroller-based circuits using both assembly and high-level language programming. But it is neither sufficient nor in pace with our industry's latest state of development. Most notable reasons are the: non-availability of (a) Text Book, (b) Experiment Book, and (c) Development Tools for both educators and students. Therefore, an educator must search here and there to complete and deliver each of his lessons adequately; thus, leaving the student feeling frustrated and unfulfilled in this course of study.

This paper will intend to highlight difficulties and problems being faced in producing engineers & technologists who can be readily and respectfully absorbed in the careers as well as the plans and suggestions to overcome those problems. The course outline along with laboratory experiments will also be discussed.

Keywords: Problems and Solutions for PIC Microcontroller

Introduction

At Northwestern State University of Louisiana, Natchitoches, we inducted a one-semester course on microcontrollers, obviously, for a multiple of reasons, for our undergraduate students doing majors in 'Electronic Engineering Technology (EET)'. At the very start of this venture, we found out that we were on a head-on-collision course to a number of challenging and multi-faced problems. This was quite disappointing. But we pursued on to develop a reasonably good course and our consistent efforts in doing-so ultimately became fruitful. As they say there is always a silver lining in a dark cloud. The problems, which appeared to be insurmountable at the first instant, turned out to be blessings in disguise - these problems became guidelines for developing a good course. At last, we have been able to develop a nice course which can be easily adopted by other institutions teaching EET courses. We will extend all sort of assistance to anyone, whenever & wherever required. The aim of writing all this is to inform those schools who have not yet included microcontrollers as a subject and who may, now, after listening / reading this paper, decide to make it a part of their syllabi.

Problems

These are:-

1) **Selection of the suitable microcontroller-family / families to be included in the course.**
2) **Determining Pre-requisite for the course**.
3) **Selection of Text Book and Lab Book**.
4) **Choice of Programming and Developing Tools.**
5) **Choice of a development / demo board.**

Solution to the Problems

6) **Selection of the suitable microcontroller family / families to be included in the course:** This is not a simple step and appeared to be a major hurdle since the modern industry / market is full of many types of microcontroller-families. Over the past many years, there have been many changes occurring in the microcontroller market. There have been taking place designs and developments for new device-families and improvements and changes in features of the existing device families. For example, the 16-bit microcontrollers were introduced and at the same time several families of 8-bit microcontrollers were designed and manufactured.

   After making a comparative reading and analysis of a number of manufacturer s' web sites and a few books [1, 2 & 3] we chose Microchip's 8-bit microcontrollers-families as our device-families. Microchip is one of the new market leaders in the today's competitive 8-bit microcontroller market. They are designing & manufacturing several families of microcontrollers. The PIC16 family & PIC 18 family are the two most popular families.

   We chose the PIC18-family microcontrollers to be taught in the class-course. The PIC18 family microcontrollers are high-end core devices. Besides having most of the features of the PIC17 family and better features than those of the PIC16 family, the PIC18 microcontrollers have an additional number of new features like: (a) the PIC 18 can access its own program memory, (b) it has 12-bit FSR registers to execute indirect addressing, (c) it has a 31-levels deep stack to facilitate subroutine and interrupt processes and (d) a number of conditional branch instructions.

For experiments and lab exercises, we chose **Microchip's PIC16F84A, PIC16F870, PIC18F452 and PIC18F8720** microcontrollers. The PIC16F84A and PIC16F870 are amongst the current microcontrollers of the choice for lab purpose / general purpose PIC microcontroller development using BASIC language programming and PICBasic PRO Compiler for compiling. The PIC18F452 and PIC18F8720 microcontrollers are chosen to illustrate and demonstrate lessons which are learnt in the PIC18 devices' class course.

7) **Determining Pre-requisite for the course**: The question, that what should be, if any, pre-requisites for the course, did not seem to have a simple direct answer. To establish it, we asked ourselves a few questions. One of these was a very obvious question of what **should be the objectives & outcomes of the course**. We decided the same objectives & outcomes for this course as the one we have for other subjects being taught in our EET program and that is that after completing this course, the students would have adequate subject-knowledge and hands-on skills necessary to develop hardware (both schematic circuits and physical circuits) and software programming and development for several types of microcontroller-based applications.

In doing so we kept in view the main objectives to be that after having successfully completed the course our EET students would be able:-

a) To confidently & effectively select and use microcontrollers' hardware in any of its stipulated roles and applications;
b) To write source codes for the microcontroller-based projects / applications using both assembly language and high-level languages;
c) To know how to use an assembler software to assemble (convert) an assembly code to an object code;
d) To know how to use a compiler software to compile (convert) a source code to a machine code and
e) To upload the assembled / compiled object codes (in the hexadecimal format) onto the microcontroller.

These objectives necessitates that the prospective students must have a reasonable background of electronics, especially that of **digital electronics** and that of a **high-level language such as BASIC, C / C^{++}.**

In light of the objectives as discussed above, it becomes very obvious that pre-requisites for the course should be at least a credit in a two-semester digital electronic engineering technology course and a credit in a one-semester high-level programming language.

8) **Selection of Text Book and Lab Book**. This was another pain-in-the-neck type difficulty. There was hardly any **text-book-type** text book and **lab-type-book** laboratory book available in the market. There was a number of books available (some of them enlisted in the 'Reference'), which more or less belong to a **reference-type** of books but none of them was found to be very helpful and usable for the educators and almost of little help to an undergraduate student.

However we have to select one out of the available books as a subject-text book and another one as a lab book. For the class course we select the following book as the text book:-

a) **Han-Way Huang; 'PIC Microcontroller: An Introduction to Software & Hardware Interfacing'** [1] .

And for the experiment & lab part of the course, we select this book:-

b) John Iovine: 'PIC Microcontroller Project Book[3]

Additionally we chose the following user's guides for learning how to use compiler software**:-**

c) **Microchip's MPLAB C18 C Compiler User's Guide**.
d) PICBasic / PIC BasicPRO Compiler User's Guide.

9) **Choice of Programming and Development Tools.** Based on our choice of the PIC16 and the PIC18 families, the following programming and development tools are required to be used for teaching both the lecture and the lab courses:
   a) **Microchip's MPLAB Integrated Development Environment (MPLAB IDE).** Microchip's MPLAB Integrated Development Environment (MPLAB IDE**)** is necessarily required and **to be** used to create, debug, build, test / troubleshoot new projects on PCs, irrespective of the programming language, before loading them onto a microcontroller.
   b) **Microchip's MPLAB C18 C Compiler.** Microchip's MPLAB C18 C Compiler is essentially required and is to be used for compiling (converting) all C-programs.
   c) **PICBasic PRO Compiler**. PICBasic PRO Compiler is essential required and is to be used to compile all BASIC programs.
   d) **Micro-Engineering Lab's Programming Carrier Board**. Micro-Engineering Lab's Programming Carrier Board is required to upload the object code onto the microcontrollers.

10) **Choice of a development / demo board**: To us there were two choices, either (a) to choose **a development / demo board** (also known as an **experiments board**), available from the manufacturers and the third parties from the market which has a specific number of experiments built on them and supposedly recommended to be quite suitable for course teaching and / hand-on training or (b) to choose **a solderless-breadboard** which is used to build each experiment circuit from resistors, capacitors, pieces of wires , etc., before inserting a programmed-microcontroller in its designated place. We chose a solderless-breadboard to be used as a development / demo board for the course. In addition to cost-saving, this choice was based on the fact that having used a solderless-breadboard, our students gain hand-on experience and better understanding of the microcontrollers and its operation.

Teaching Requirements

In order to determine teaching requirements for the course, first of all, we have to know how a microcontroller-based project is developed. For this purpose a brief description of the development process is appended here.

Development of a microcontroller-based application (project) is a two-part development process: (a) **a hardware development** process and (b) **a software development process**. The two development processes are closely related to each other and must progress side-by-side but are distinct from each other.

On the hardware side, one needs to convert a given problem into a schematic circuit, to collect components and to connect them on a solderless breadboard and to check the connected circuit for its correctness. Then one needs to choose / collect a microcontroller, if not already chosen / collected, which can meet the hardware requirements of the project. If the microcontroller is not loaded with the object code, then one needs to load the assembled / compiled program onto the microcontroller. This is the software development process.

The software development, itself, is basically a three-step processes as:

(a) **To write a program / code:** To write a program / code, (also known as a source code), is to write a sequence of instructions, that a microcontroller must execute to do a job;

(b) **To assemble / to compile a source code**: To assemble / to compile a source code is to convert a source code into an object code, i.e., a code which is executable by the microcontroller. (.HEX code);

(c) **To program the microcontroller:** This is the final step in which the object code is loaded onto the microcontroller.

There is some other very important aspects in the software development, like: (a) If a source code / program is written in the assembly language, the source code needs an **assembler**-software to assemble (convert) it to the object code and (b) If a source code is written in a high-level language, the source code needs a **compiler**-software to compile (convert) the source code into the object code.

It is also to be noticed that Microchip's MPLAB IDE is a complete development tool for assembly- codes. In addition to this, Micro-Engineering Lab's Programming Carrier Board is used to upload the assembled object code onto a microcontroller. On the other hand, PICBasic PRO-based programs and C-language-based programs are to be compiled before loading them onto the microcontrollers.

Class Course

In light of 'teaching requirements' it is evident that teaching of the class course requires teaching of (a) PIC18 hardware, (b) PIC18 assembly language programming (c) the C-language and use of Microchip's C18 C Compiler (d) Microchip's PIC 18 development tool: MPLAB IDE.

This means the class course will include lessons on the PIC18 family microcontrollers' organization, architecture and features, data memory and program memory, addressing modes, instruction sets and assembler directives, assembly language programming and C language programming.

The course should also cover on how to use Microchip's MPLAB IDE software development tool to create and build a project, and on how to use MPLAB SIM to simulate and run the project and view output variables using 'Watch Windows'.

The course also includes lesson on as how to use MPLAB C18 C Compiler to compile C -programs. **Table 1 enlists the course topics**.

A number of projects to perform simple computations, to read / write (interface I/O) from / onto PORTs and to use of Timers, interrupt etc., were developed and run to demonstrate how a project are built and how a microcontroller fulfills its features.

Table 2 gives a list of projects for lesson demos.

Lab Course

For the lab part of the course, experiments (applications) were grouped according to type of programming language used.

A group of experiments are done using BASIC language. The PIC16F84A & 16F870 microcontrollers are the selected devices. The source codes are written in BASIC and are compiled to obtain the executable code using the **PICBasic / PIC BasicPRO Compiler.**

A second group of experiments are based on the PIC18 family microcontrollers, namely, PIC18F452 and PIC18F8720. Programs are written using both assembly language and c-language. Microchip's MPLAB C18 C Compiler is used to convert the c-programs and MPLAB ASM used to convert assembly programs into object codes.

In doing each of the experiment, Micro Engineering Lab's Programming Board is used to upload the object code on to the microcontroller.

Table 3 enlists the experiments performed in the lab course.

Conclusion

We have almost highlighted all aspect of a subject on microcontroller. Our students, after completing the course, are supposed to know how to use a microcontroller to do a project. They showed keen interest in developing and building new projects. They are equally interested and dedicated to complete the projects on PCs and on breadboards.

An example program in assembly language to multiply two 16-bits signed integers is shown in Figure 1. The program was run using MPLAB IDE.

The Watch window in Figure 2 shows the result of the program. The two 16-bit integers are M2M1 and N2N1. The product variable is P4P3P2P1.

So the answer is: **(0x67FF) *(0x7F9A) = 0x33D61066.**

Figure 3 shows a breadboard project, namely, a PIC16F84A microcontroller-based seven-segment display.

Another broadband project, namely, a PIC18F452 microcontroller-based two-way traffic light signal controller built on a solderless breadboard is shown in Figure 4.

These projects are developed and built by students during their class course (Figure 1) and during their lab classes (Figure 3 & Figure 4).

Table 1: Course Topics

A. Analyze the basic concepts & application of PIC 18 microcontroller's memory, CPU registers, pipelining, instruction format, addressing modes and a sample of instruction set.
B. An introduction to and use of the PIC 18 family microcontrollers' assembly language.
C. An introduction to and use of the PIC18 Development tools, namely, MPLAB IDE, MPLAB SIM
D. An introduction to and use of the C language and how to use the MPLAB C 18 C Compiler.
E. Analyze the basic concepts & application of PIC 18 microcontroller's interrupts, interrupt operation & programming and resets.
F. An introduction to and application of parallel (I / O) ports, Timers and CCP Modules.

Table 2: Projects for Lesson Demos

G. PIC18F452-based: Binary counting displayed on PORTD. Program is written in c-language.
H. PIC18F452-based: LED pattern displayed on PORTD. Program is written in c-language.
I. PIC18F452-based: Binary counting displayed on PORTD, with a time delay created using TIMER and interrupt features. Program is written in c-language.
J. PIC18F452-based: A time delay created using program loop. Program is written in assembly-language.
K. PIC18F8720-based: Compute 0x68 ÷ 0x10 (unsigned). Program is written in assembly language.
L. PIC18F8720-based: Compute 0x5329 * 0x8437 (unsigned). Program is written in assembly language.
M. PIC18F8720-based: Determine maximum element of an array. Program is written in assembly language.
N. PIC18F452-based: Compute sum of even / odd numbers of an array. Program is written in assembly-language
O. PIC18F452-based: Swap elements of an array store in program memory. Program is written in assembly-language
P. PIC18F8720-based:Compute 0xFF7F * 0XFF7F (signed). Program is written in assembly language.
Q. PIC18F452-based: Compute average of elements of an array. Program is written in assembly-language.

Table 3: List of Experiments Done in the Lab Course

A. PIC 16F84A-based: Winking of 2 LEDs connected to RB0 & RB1 pins of PORTB; Program in BASIC & compiled using PIC Basic / PICBasicPRO Compiler.
B. PIC 16F84A-based: Binary counting from Hex 00 to HexFF using 8 LEDs connected to PORTB; Program in BASIC & compiled using PIC Basic / PICBasicPRO Compiler.
C. PIC 18F452-based: Traffic light controller using PORTD: Program written in c-language.
D. PIC 16F84A-based: Binary progression on 8 LEDs connected to PORTB; Program in BASIC & compiled using PIC Basic / PICBasicPRO Compiler.
E. PIC 16F84A-based: Binary counting, at changing counting rate and holding counting; Program in BASIC & compiled using PIC Basic / PICBasicPRO Compiler.
F. PIC 16F84A-based: Binary counting from hex 00 to FF using 8 LEDs connected to PORTB: Program in BASIC & compiled using PIC Basic / PICBasicPRO Compiler.
G. PIC18F452-based: Drive seven segment displays: Program written in c-language & compiled using MPLAB C18 C Compiler.

H. PIC16F84A-based: Drive seven segment display: Program written in BASIC language & compiled using PIC Basic / PICBasicPRO Compiler.
G. PIC16F84A-based: Control of a dc motor: Program written in BASIC language & compiled using PIC Basic / PICBasicPRO Compiler

Figure 1: A Program (In PIC18 Assembly Language) To Multiply Two 16-Bits Signed Integers

(Page 1)

```
Title "A program to multiply two 16-bits"
      #include <p18F8720.inc>
        clock 0x10
        M1,M2
      ;multiplicand
        endc
        clock 0x20
        N1,N2
      ;multipliers
        endc
        clock 0x30
        P1,P2,P3,P4
        endc
  m1 equ     0xFF
  m2 equ     0x7F
  n1 equ     0xFF
  n2 equ     0x7F
     org     0x00
     go to   start
     org     0x08
     retfie
     org     0x18
     retfie
  start clrf P1,A
        clrf P2,A
        clrf P3,A
        clrf P4,A
        movlw     m1
        movwf     M1,A
        movlw     m2
        movwf     M2,A
        movlw     n1
        movwf     N1,A
        movlw     n2
        movwf     N2,A
        movf      M1,W,A
        mulwf     N1,A
```

(Page 2)

```
        movff     PRODL,P1
        movff     PRODH,P2
        mulwf     N2,A
        movf      PRODL,W,A
        addwf     P2,F,A
        movff     PRODH,WREG
        addwfc    P3,F,A
        movf      M2,W,A
        mulwf     N1,A
        movf      PRODL,W,A
        addwf     P2,F,A
        movf      PRODH,W,A
        addwfc    P3,F,A
        movf      M2,W,A
        mulwf     N2,A
        movf      PRODL,W,A
        addwf     P3,F,A
        movf      PRODH,W,A
        addwfc    P4,F,A
        btfss     M2,7,A
        goto      next
        movf      M1,W,A
        subwf     P3,F,A
        movf      M2,W,A
        subwfb    P4,F,A
next    btfss     N2,7,A
        goto      done
        movf      N1,W,A
        subwf     P3,F,A
        movf      N2,W,A
        subwfb    P4,F,A
done    nop
        end
```

Figure 2: Viewing Variables on the Watch Window

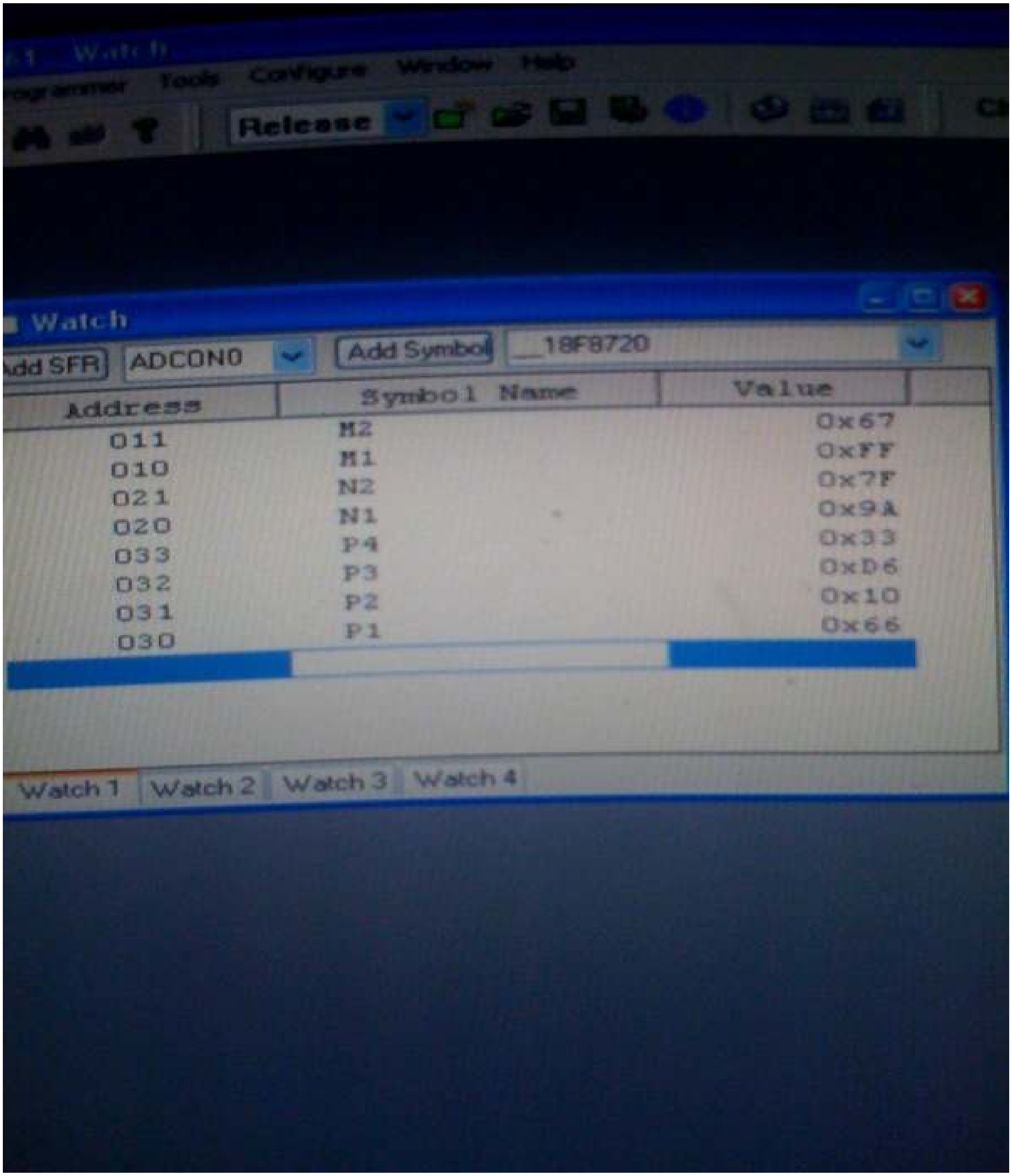


Figure 3: 16F84A-Based Seven-Segment Decoder /Driver Circuit.

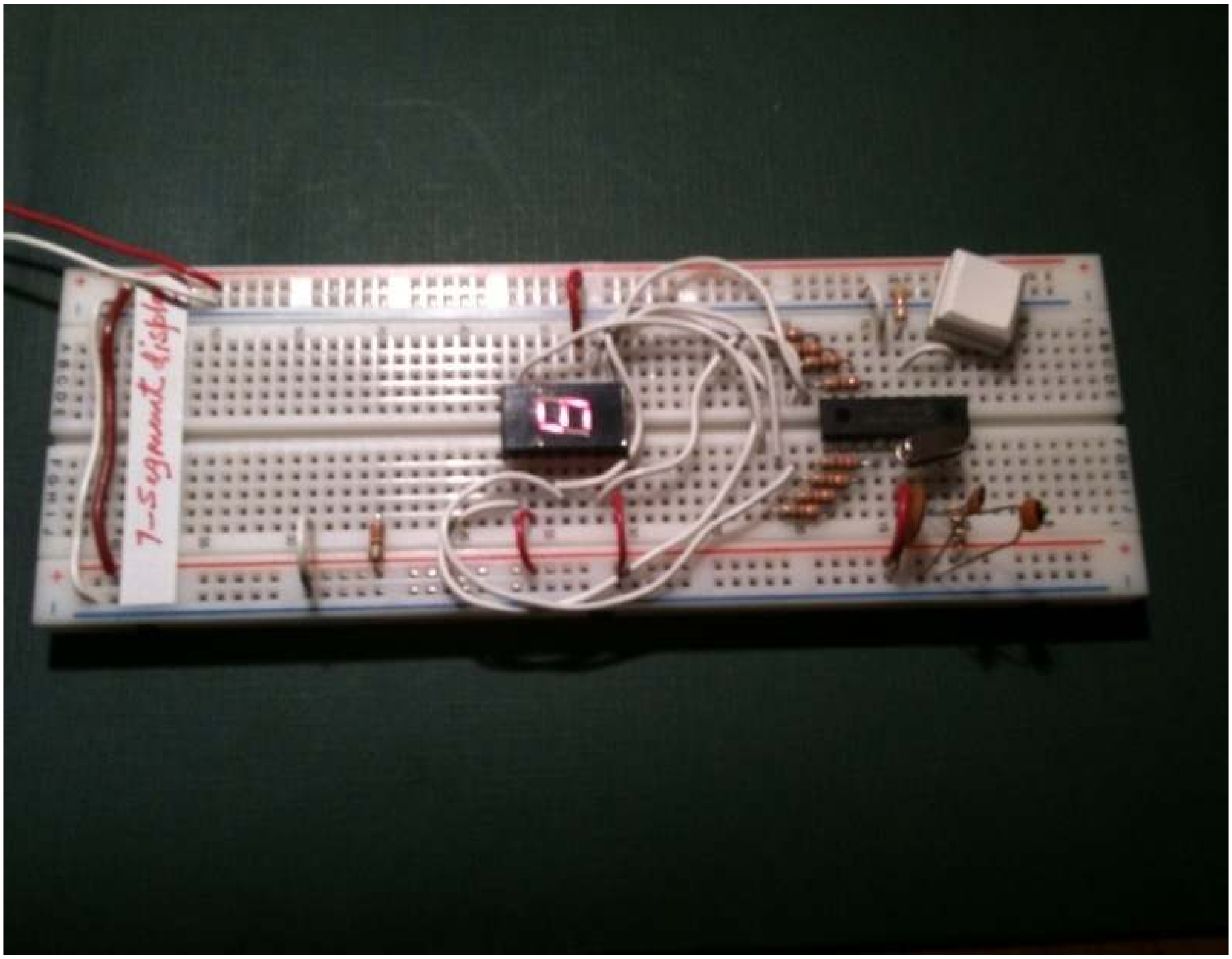
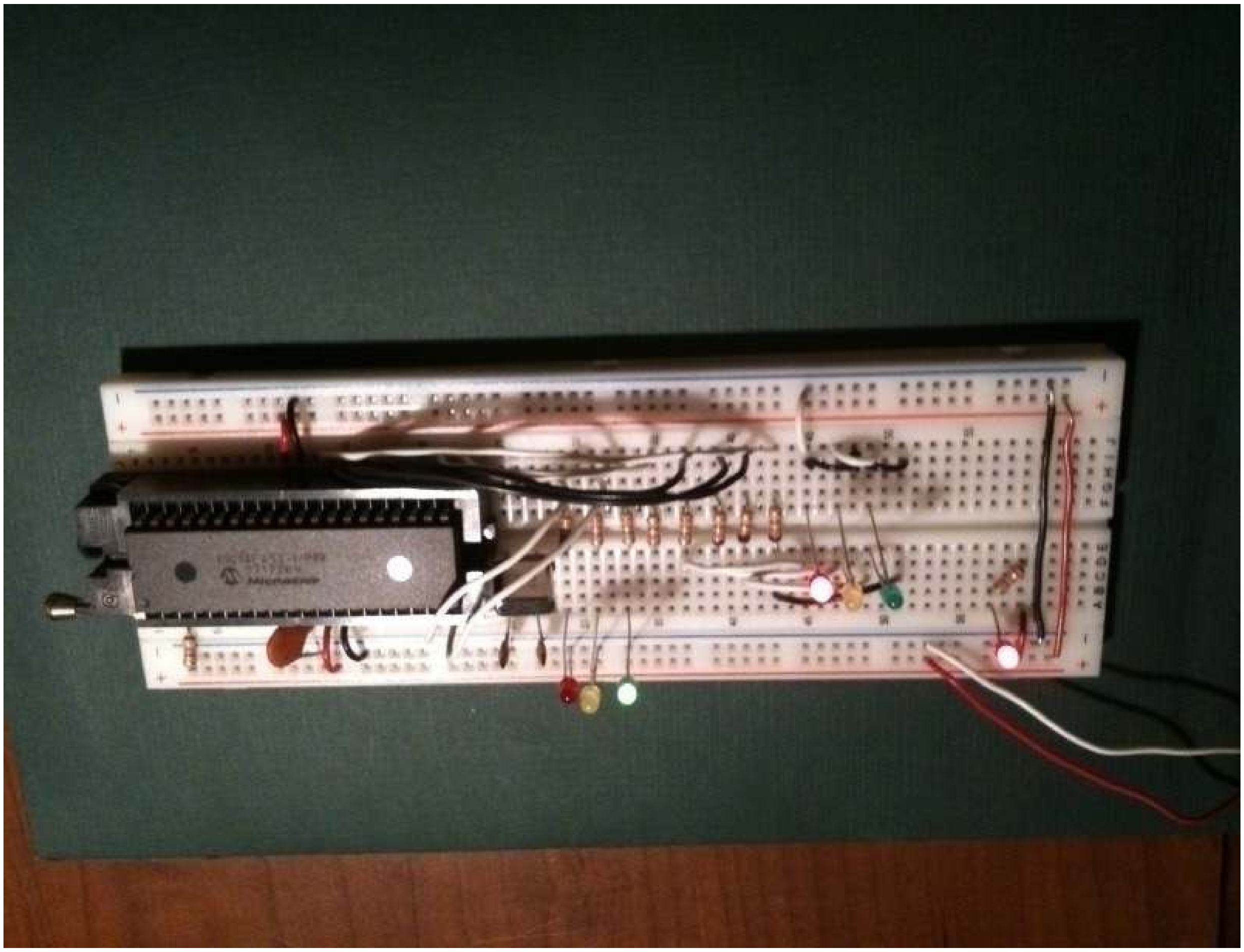


Figure # 4: A PIC18F452-Based Two-Way Traffic Light Controller



References

1) Han-Way Huang, *PIC Microcontroller: An Introduction to Software & Hardware Interfacing,* Publisher: Thomson Delmar Learning

2) Myke Predko, *Programming And Customizing The Pic Microcontroller*, publisher: McGraw-Hill

3) John Iovine, *PIC Microcontroller Project Book* , publisher: McGraw-Hill/TAB Electronics

4) Microchip's web site: *www:microchip.com*

5) Myke Predko, *Handbook of Microcontrollers*, publisher: McGraw-Hill

6) Microchip, *MPLALB C18 C Compiler User's Guide*

7) MicroEng, *PICBasic PRO Compiler User's Book*