Chapter-1

INTRODUCTION

Objective of the project

- To build a circuit to facilitate the movement of traffic in a 4-way lane system.
- To reduce the waiting time for commuter in a lane before he can pass the junction without risking the chances of accident during the lane change.

1.1 Brief description

- The project uses simple electronic components such as LED as TRAFFIC LIGHT indicator, a seven segment display and a MICROCONTROLLER for auto change of signal after a pre-specified time interval.

Figure shows the drawing of the 4-way junction, where each way has its traffic light and counter

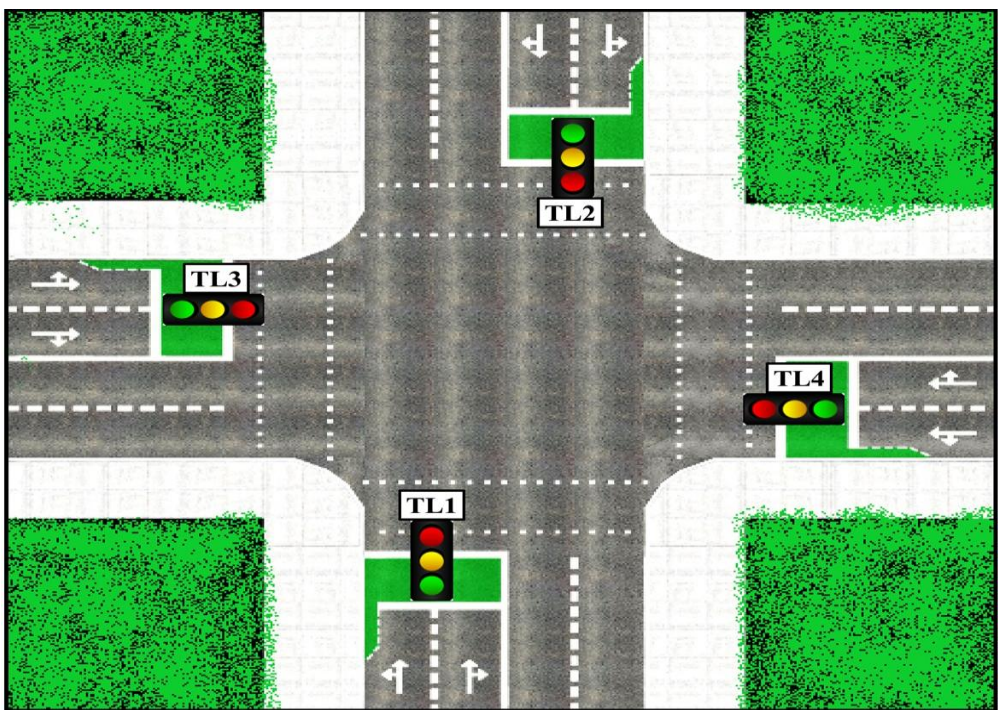
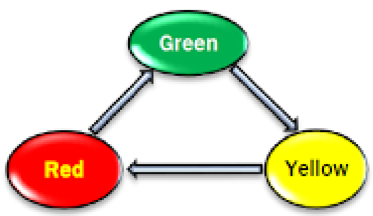


FIGURE-1 GENERAL 4-WAY TRAFFIC LANE

- Microcontroller PIC 16F877A is the brain of the project which initiates the traffic signal at a junction.

- LEDs used are red, yellow and green.
  Red LED indicates **"stop driving"**
  Yellow LED indicates **"start stopping"**
  Green LED indicates **"drive".**

The sequence of altering the LEDs according to their color is as shown in the figure below: Green-Yellow-Red-Green. Twelve LEDs are used; three to each traffic light.



The LED's are automatically on and off by making the corresponding port pin of the micro controller high. Furthermore associated is the right turn green lights which are on for the first 10 seconds of the total green light time.

- 7-segment LED displays are used to show the current count value. Since all of the traffic lights are working simultaneously, each one is to display a different digit than the other. When a traffic light is tuned green, its corresponding 7-segment displays start counting down from a specific value and decrements until zero is reached. After this the counter starts by a new count value at the moment the yellow light turns on.

- When the red light turns on after the yellow took its time, the count continues to decrement until reaching zero. This means that the same 7-segments, on each traffic light, are used to display the count when cars are allowed and not allowed to pass. In terms of counting, the yellow and red are considered one set while the green is another set. The circuit board designed supports in-circuit serial programming (ICSP) for the PIC. This support eases the way to the designer to program the microcontroller without the need to plug the microcontroller in and out repeatedly.

1.2 BLOCK DIAGRAM

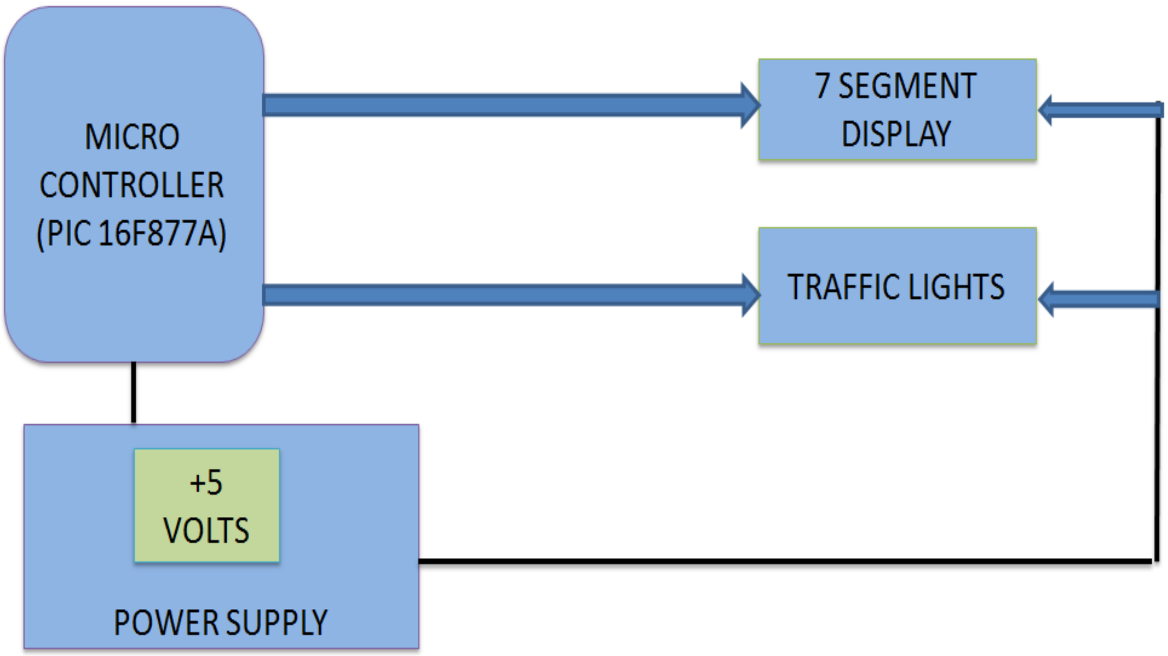


FIGURE-2 CIRCUIT BLOCK DIAGRAM

1.3 FLOW DIAGRAM

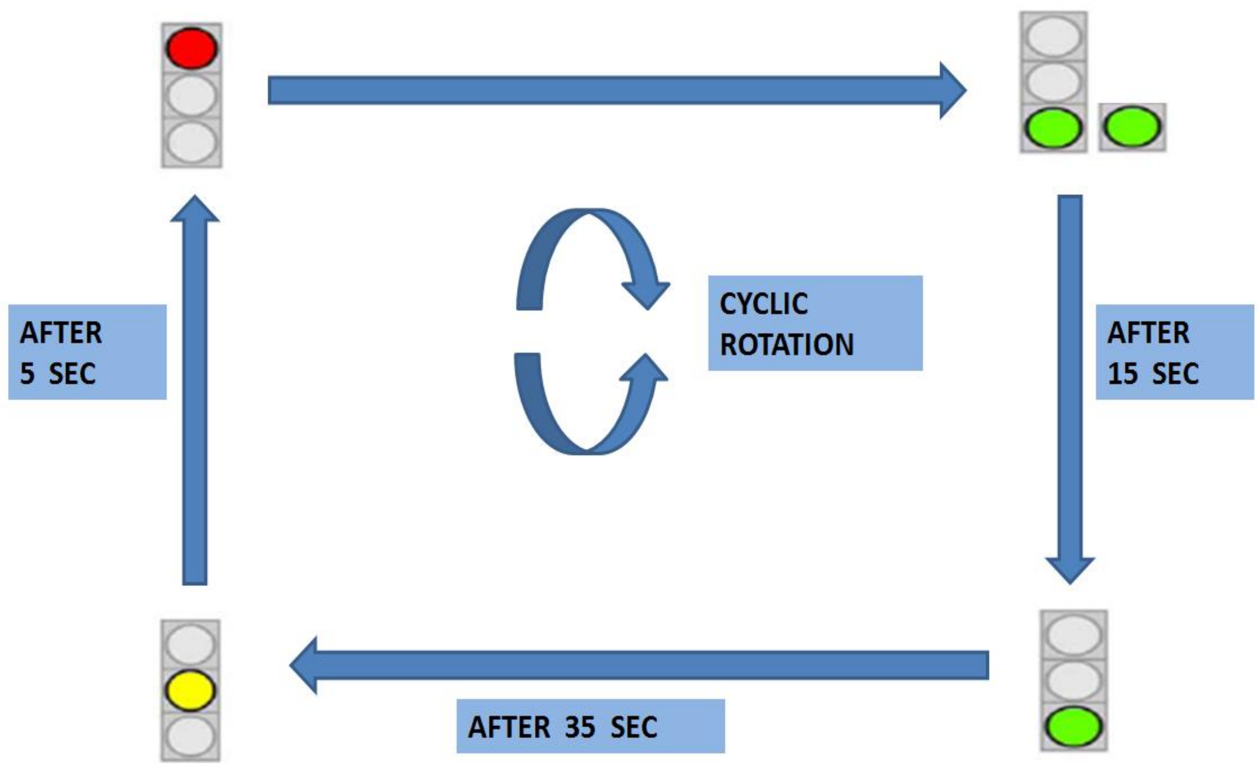


FIGURE-3 PROGRAM FLOW DIAGRAM

1.4 FLOW CHART

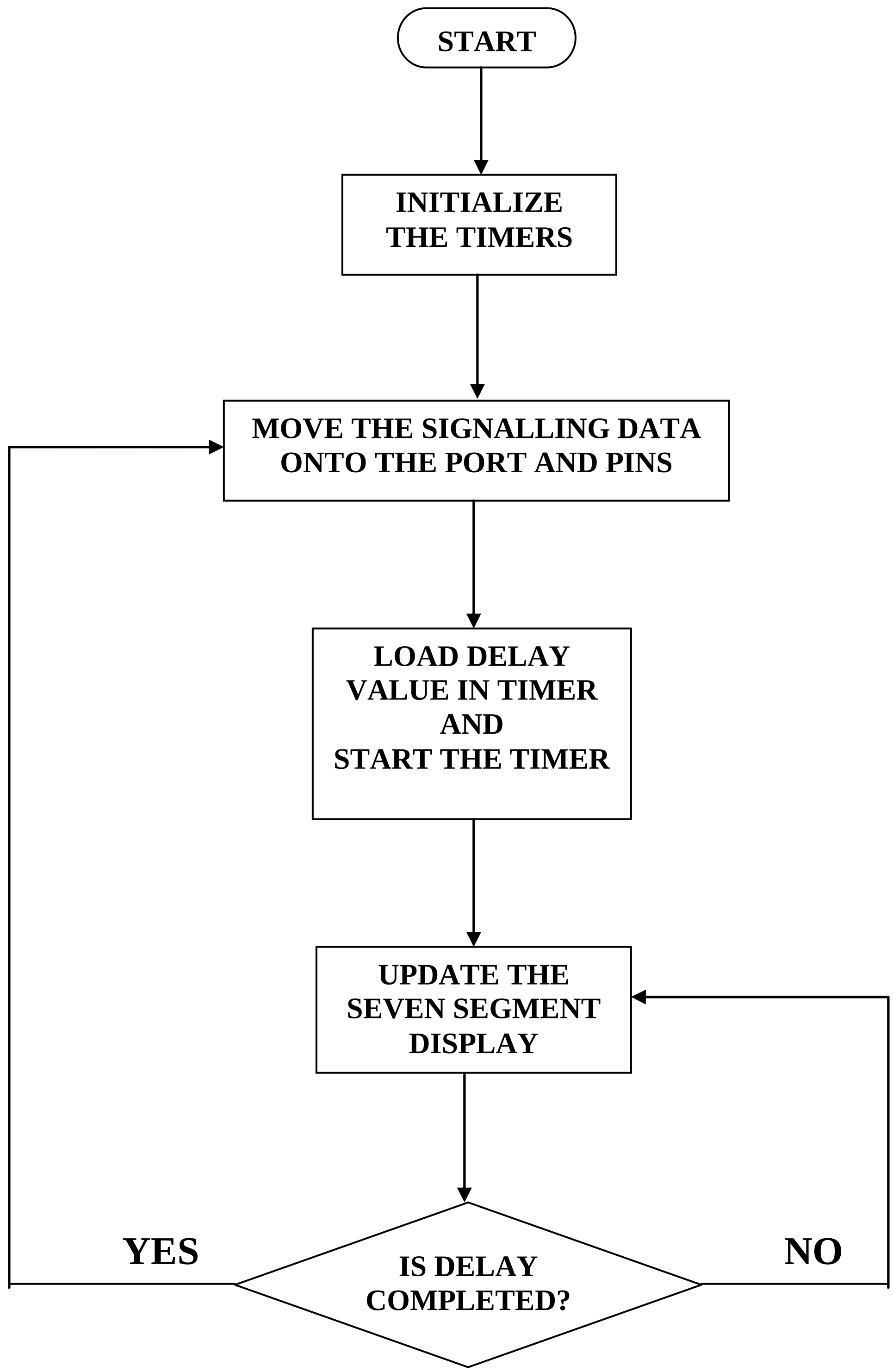


FIGURE-4 PROGRAM FLOW CHART

1.5 CONNECTION DIAGRAM

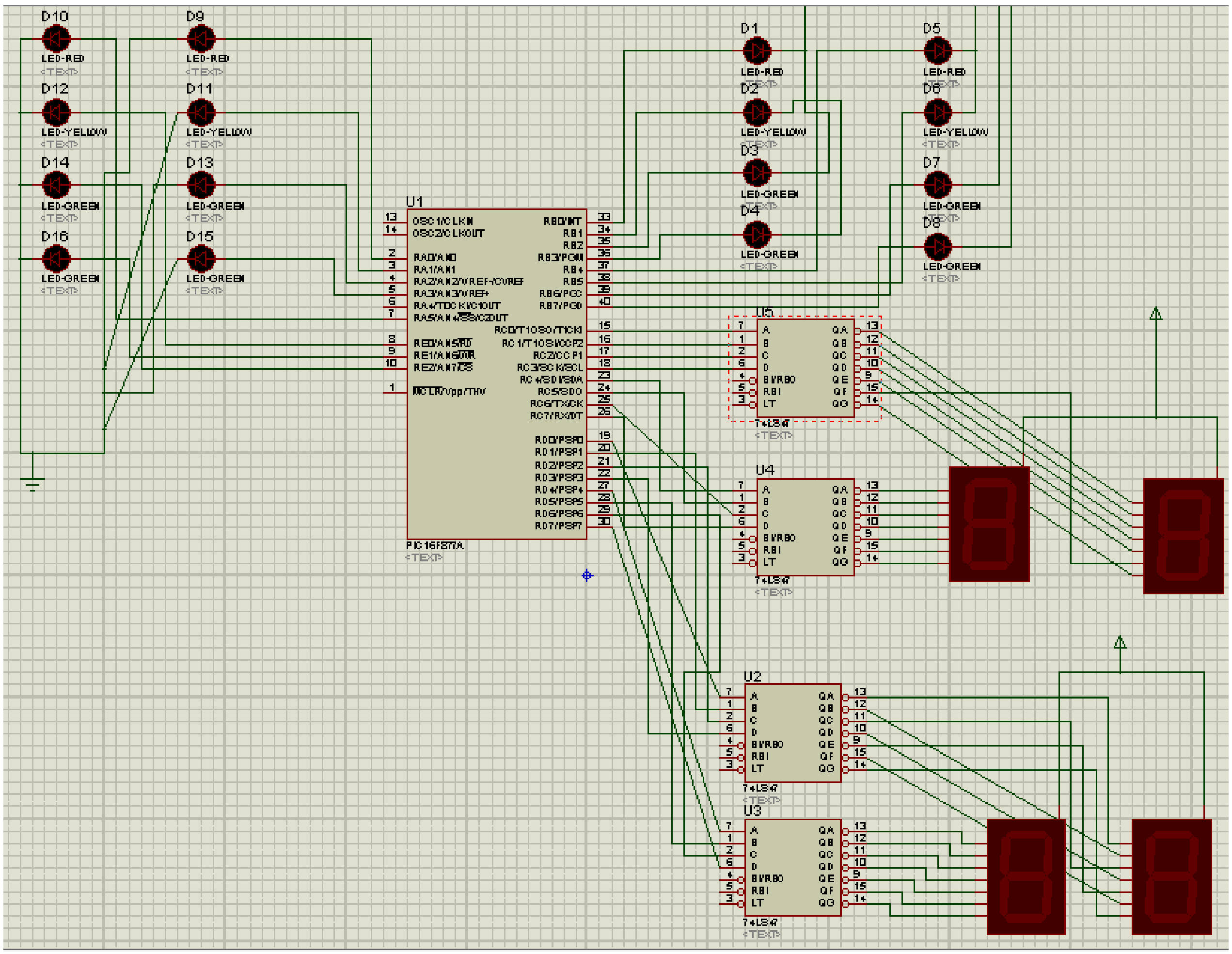


FIGURE-5 CIRCUIT CONNECTION DIAGRAM

Chapter-2
PIC MICROCONTROLLERS

2.1 INTRODUCTION

The term PIC stands for Peripheral Interface Controller .It is the brain child of Microchip Technology, USA. Originally this was developed as a supporting device for PDP computers to control its peripheral devices, and therefore named as PIC, Peripheral Interface Controller. They have coined this name to identify their single chip micro controllers. These 8-bit micro controllers have become very important now -a -days in industrial automation and embedded applications etc.

2.1.1 Overview and Features

- The PIC 16F8XX Microcontrollers are basically RISC microcontrollers with very small instruction set of only 35 instructions and a two-stage pipeline concept fetch and execution of instructions. As a result, all instructions execute in a single cycle except for program branches.
- There are four devices in 16F8xx family, PIC16F873, PIC16F874, PIC16F876 and PIC16F877.The PIC16F876/873 devices come in 28-pin packages and the PIC16F877/874 devices come in 40-pin packages. The Parallel Slave Port is not implemented on the 28-pin devices.
- PIC 16F877 is a 40-pin 8-Bit CMOS FLASH Microcontroller. The core architecture is high-performance RISC CPU. Since it follows the RISC architecture, all single cycle instructions take only one instruction cycle except for program branches which take two cycles.
- 16F877 comes with 3 operating speeds with 4, 8, or 20 MHz clock input. Since each instruction cycle takes four operating clock cycles, each instruction takes 0.2 µs when 20MHz oscillator is used.
- It has two types of internal memories .One is program memory and the other is data memory. Program memory is provided by 8K words (or 8K*14 bits) of FLASH Memory, and data memory has two sources. One type of data memory is a 368-byte RAM (random access memory) and the other is256-byte EEPROM (Electrically erasable programmable ROM).
- The core features include interrupt up to 14 sources,
- power saving SLEEP mode,
- a single 5V supply and
- In-Circuit Serial Programming (ICSP) capability.

2.1.2 SALIENT FEATURES

- **Speed :**

When operated at its maximum clock rate a PIC executes most of its instructions in 0.2 μs or five instructions per microsecond.

- **Instruction set Simplicity :**

The instruction set is so simple that it consists of only just 35 instructions.

- **Integration of operational features:**

Power-on-reset (POR) and brown-out protection ensure that the chip operates only when the supply voltage is within specifications. A watch dog timer resets the PIC if the chip malfunctions or deviates from its normal operation at any time.

- **Programmable timer options:**

Three timers can characterize inputs, control outputs and provide internal timing for the program execution.

- **Interrupt control:**

Up to 12 independent interrupt sources can control when the CPU deal with each sources.

- **Powerful output pin control:**

A single instruction can select and drive a single output pin high or low in its 0.2 μs instruction execution time. The PIC can drive a load of up to 25μA.

- **I/O port expansion:**

With the help of built in serial peripheral interface the number of I/O ports can be expanded. EPROM/DIP/ROM options are provided.

- High performance RISC CPU
- Operating speed: DC – 20 MHz clock input DC – 200 ns instruction cycle
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Three Timers Timer0,Timer 1 and Timer 2.
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code-protection
- Power saving SLEEP mode
- 10-bit multi-channel Analog-to-Digital converter
- Selectable oscillator options
- One USART /SCI port with 9-bit address detection.
- Low-power, high-speed CMOS EPROM/ROM technology
- Fully static design
- Wide operating voltage range: 2.5V to 6.0V
- Commercial, Industrial and Extended temperature ranges

2.2 ARCHITECTURE

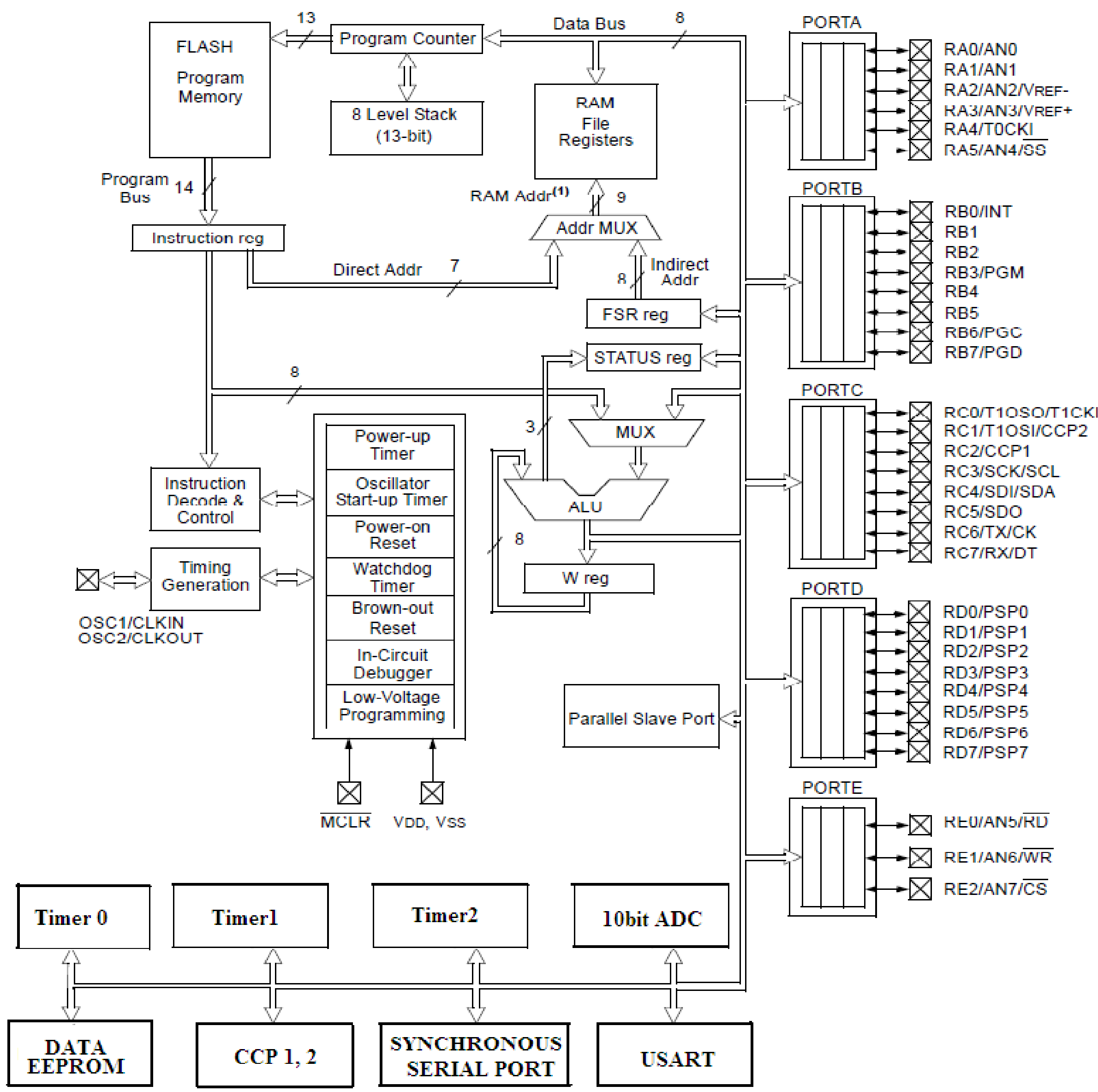


FIGURE-6 BLOCK DIAGRAM OF PIC 16F877A MICROCONTROLLER

2.3 PIN DIAGRAM

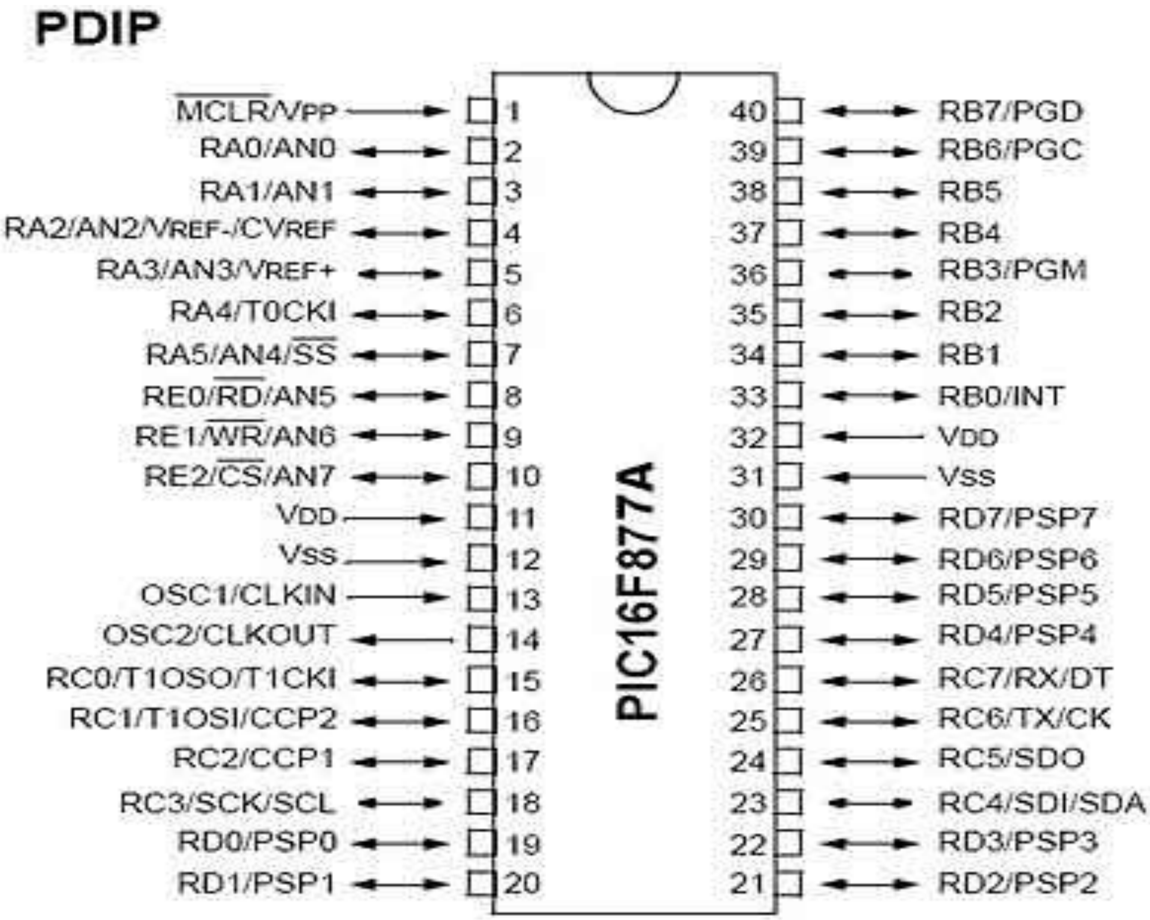


FIGURE-7 PIC16F877A PIN DESCRIPTION

2.4 PIC FEATURES

2.4.1 MEMORY ORGANIZATION

The memory module of the PIC controller has three memory blocks.

a) Program memory
b) Data memory and
c) Stack

a) Program Memory

The PIC 16F8XX has 4k x14 program memory space (0000H-0FFFH).It has a 13 bit Program counter(PC) to access any address (2^{13}=4k). This PIC family uses 13-bit program counter allowing the controllers to an 8k-program memory without changing the CPU structure.

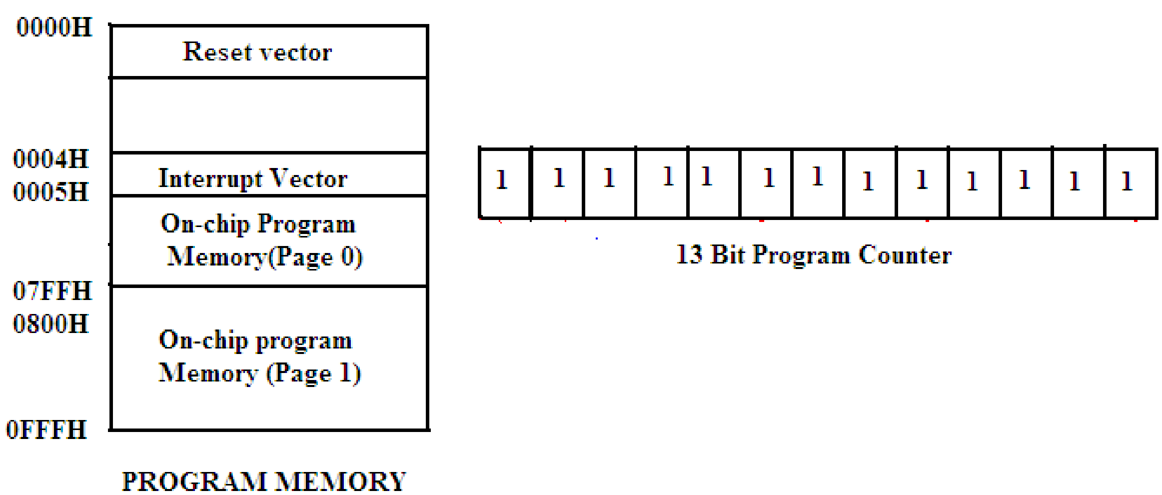


FIGURE-8 PROGRAM MEMORY

b) Data memory

The data memory of PIC 16F8XX is partitioned into multiple banks which contain the general purpose registers and the Special function Registers.(SFRs).The bits RP1 and RP0 bits of the status register are used to select these banks. Each bank extends upto 7FH(128 Bytes).The lower bytes of the each bank are reserved for the Special Function Registers. Above the SFRs are general purpose registers implemented as static RAM.

2.4.2 REGISTER FILE STRUCTURE

In PIC Microcontrollers the Register File consists of two parts namely

a) General Purpose Register File
b) Special Purpose Register

2.4.3 PARALLEL I/O PORTS

Most of the PIC16cx/7x family controllers have 33 I/O lines and five I/O ports They are PORT A, PORT B, PORT C , PORT D and PORT E.

- <u>PORT A</u>:

Port A is a 6-bit wide bi-directional port. Its data direction register is TRISA setting TRISA bit to 1 will make the corresponding PORT A Pin an input. Clearing a TRIS a bit will make the corresponding pin as an output.

- <u>PORT B</u>:

Port B is an 8-bit wide, bi-directional port. Four of the PORT B pins RB_7 – RB_4 have an interrupt-on- change feature. Only the pins configured as inputs can cause this interrupt to occur.

- <u>PORT C</u>:

Port C is an 8-bit wide, bidirectional port. Bits of the TRISC Register determine the function of its pins. Similar to other ports, a logic one 1 in the TRISC Register configures the appropriate port pin as an input.

- <u>PORT D</u>:

Port D is an 8-bit wide bi-directional port. In addition to I/O port, Port D also works as 8-bit parallel slave port or microprocessor port. When control bit PSPMODE (TRISE:4) is set.

- <u>PORT E</u>:

It is a 3-bit bi-directional port. Port E bits are multiplexed with analog inputs of ADC and they serve as control signals (RD, WR, CS) for parallel slave port mode of operation.

2.4.4 TIMER MODULES

There are three completely independent Timers available in PIC 16F8XX Microcontrollers. They are

- Timer 0
- Timer1 and
- Timer2

2.4.5 ADDRESSING MODES

The PIC microcontrollers support only TWO addressing modes .They are
(i) Direct Addressing Mode
(ii) Indirect Addressing mode

Direct Addressing Mode
In direct addressing mode 7 bits (0-6) of the instruction identify the register file address and the 8 th bit of the register file address register bank select bit (RP0).

Indirect Addressing Mode
In the indirect addressing mode the 8-bit register file address is first written into a Special Function Register (SFR) which acts as a pointer to any address location in the register file. A subsequent direct access of INDF will actually access the register file using the content of FSR as a pointer to the desired location of the operand.

2.4.6 INSTRUCTION SET

The instruction set of PIC is divided into three basic categories. They are:
(a) Byte oriented Instructions
(b) Bit oriented Instructions
(c) Literal and Control Instructions

Byte Oriented Instructions
In a byte oriented Instructions **f** represents a file register and **d** represents destination register. The destination specifies where the result of operation is to be placed. If D= 0 the result is placed in W register (Accumulator) and if d = 1, the result is placed in the file register specified in the instruction.

ADDWF f, d	: Add W and f
CLRF f	: Clear f
MOVWF f, d	: Move f
NOP	: No operation
SUBWF f, d	: Subtract W from f

Bit Oriented Instruction
In bit oriented instructions, b represents a bit field designator which selects the number of the bit affected by the operation and f represents the number of the file in which the bit is located**.**

BCF f, b	: Bit clear f
BSF f, b	: Bit set f
BTFSC f, b	: Bit test f, skip if set

Literal and Control Instructions

In literal and control instructions K represents an 8 or 11 bit constant or literal value.

ADDLW k : Add literal and W
ANDLW k : AND literal with W
CALL k : Call subroutine
MOVLW k : Move literal to W

2.4.7 CLASSIFICATION OF INSTRUCTIONS

(i) Arithmetic Operations
(ii) Logical Instructions
(iii) Increment/Decrement Instructions
(iv) Data Transfer instructions
(v) Clear Instructions
(vi) Rotate Instructions
(vii) Branch Instructions

2.4.8 PIC I/O PROGRAMMING (PROGRAMMING THE PORTS)

The PIC 16F family of microcontrollers has a total of 33 pins arranged into 5 ports. PortA, Port B, Port C, Port D and Port E. In order to use them as I/O ports, they must be properly programmed. In addition to acting as I/O ports, they also have certain additional functions like ADC, Timers, Interrupts and serial communication pins etc.

- **PORT A and the TRIS A Registers**

PORTA is a 6-bit wide, bidirectional port. The corresponding data direction register is TRISA. Setting a TRISA bit (= 1) will make the corresponding PORTA pin an input. Clearing a TRISA bit
(= 0) will make the corresponding PORTA pin an output.

Other PORTA pins are multiplexed with analog inputs and the analog VREF input for both the A/D converters and the comparators. The operation of each pin is selected by clearing/setting the appropriate control bits in the ADCON1 and/or CMCON registers.

- **PORT B and the TRIS B Registers**

PORTB is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISB. Setting a TRISB bit (= 1) will make the corresponding PORTB pin an input. Clearing a TRISB bit (= 0) will make the corresponding PORTB pin an output.

- **PORT C and the TRIS C Registers**

PORTC is an 8-bit wide, bidirectional port. The corresponding data direction register is TRISC. Setting a TRISC bit (= 1) will make the corresponding PORTC pin an input. Clearing a TRISC bit (= 0) will make the corresponding PORTC pin an output.

.

- **PORT D and the TRIS D Registers**

PORTD is an 8-bit port with Schmitt Trigger input buffers. Each pin is individually configurable as an input or output. PORTD can be configured as an 8-bit wide microprocessor port (Parallel Slave Port) by setting control bit, PSPMODE (TRISE<4>). In this mode, the input buffers are TTL.

- **PORT E and TRIS E Registers**

PORTE has three pins (RE0/RD/AN5, RE1/WR/AN6 and RE2/CS/AN7) which are individually configurable as inputs or outputs. These pins have Schmitt Trigger input buffers. The PORTE pins become the I/O control inputs for the microprocessor port when bit PSPMODE (TRISE<4>) is set. In this mode, the user must make certain that the

TRISE<2:0> bits are set and that the pins are configured as digital inputs. Also, ensure that ADCON1 is configured for digital I/O. In this mode, the input buffers are TTL.

ADVANTAGES

The PIC architecture have these advantages:

- Small instruction set to learn
- RISC architecture
- Built in oscillator with selectable speeds
- Easy entry level, in circuit programming plus in circuit debugging PIC Kit units available for less than $50
- Inexpensive microcontrollers
- Wide range of interfaces including I^{2}C, SPI, USB, USART, A/D, programmable comparators, PWM, LIN, CAN, PSP, and Ethernet.

LIMITATIONS

The PIC architectures have these limitations:

- One accumulator
- Register-bank switching is required to access the entire RAM of many devices
- Operations and registers are not orthogonal; some instructions can address RAM and/or immediate constants, while others can only use the accumulator

Stack limitations:

- The hardware call stack is not addressable, so preemptive task, switching cannot be implemented.
- Software-implemented stacks are not efficient, so it is difficult to generate reentrant code and support local variables.

Chapter-3
HARDWARE DEVELOPMENT TOOLS

Numerous hardware development tools are available for the PIC18 microcontrollers. Some of these products are manufactured by Microchip Inc., and some by third-party companies. The most ones are:

- Development boards
- Device programmers
- In-circuit debuggers
- In-circuit emulators
- Breadboards

3.1 HARDWARE COMPONENTS USED

3.1.1 PICKIT 2 USB PROGRAMMER

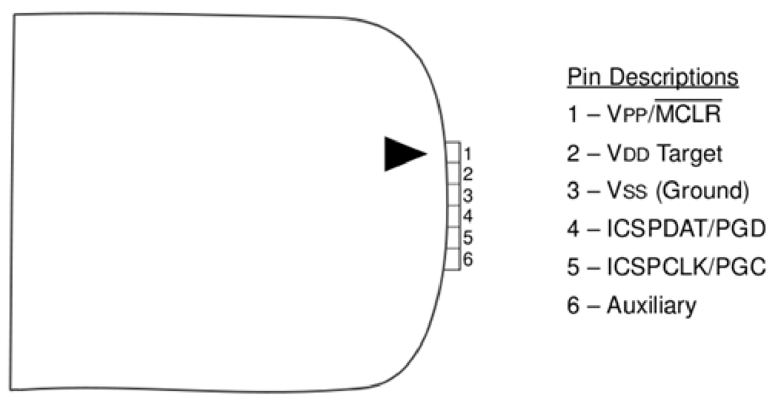
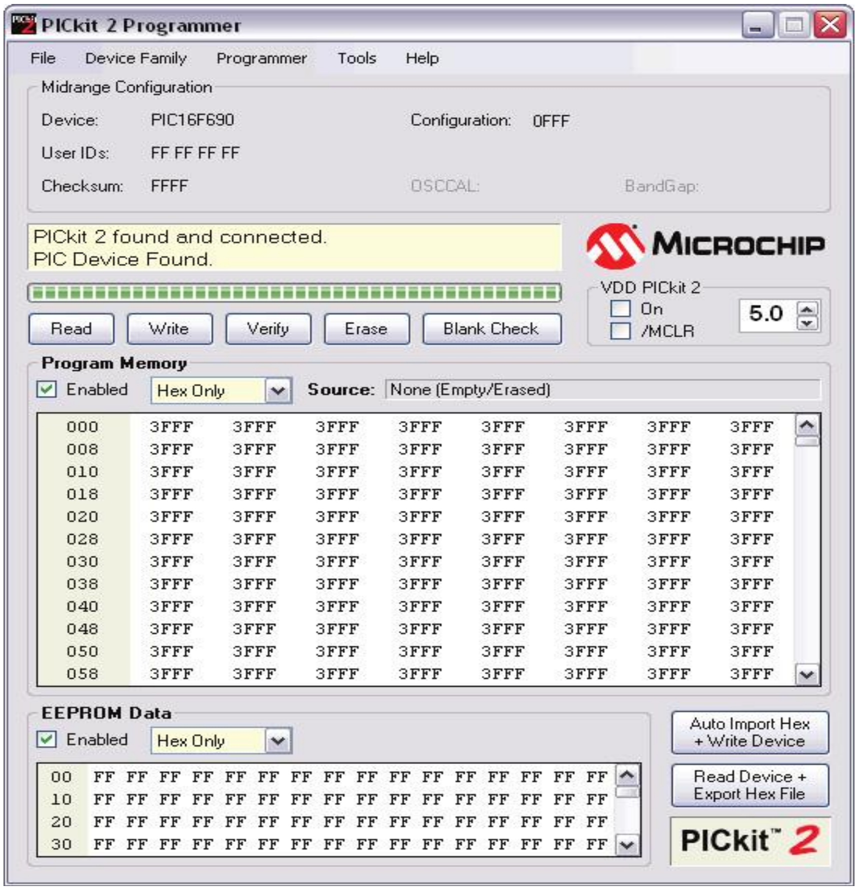


FIGURE-9 PICKIT 2 PIN DIAGRAM

Features:

- Separate programmer/debugger unit which plugs into the board carrying the chip to be programmed.
- The PICkit 2 is open to the public, including its hardware schematic, firmware source code and application programs.
- Programmer-To-Go: Set up a PICkit 2 to program a device without the need for a PC.
- 128K byte memory.
- Easy to use with MIKROC® IDE and other development environments.
- Includes the UART Tool and Logic Tool microcontroller development utilities.

The PICkit 2 Programmer application allows you to program all supported devices listed in the PICkit 2 Readme file.



MENU BAR

The menu bar selects various functions of the PICkit 2 Programmer application. A summary of the functions are:

File

Import Hex – Import a hex file for programming. The hex file format INHX32 is supported.

Export Hex – Export a hex file read from a device. The hex file is created in the INHX32 format.

File History – Up to the last four hex files opened are displayed with their filepath. These recent hex files may be selected to quickly import them. Note that the file history will initially be blank on a new install action until a hex file is imported.

Exit – Exit the program.

Device Family

Select a device family to search for a connected device in that family. Selecting the device family of the current part will clear all device data. Some families which cannot be auto-detected (such as Baseline) will bring up a drop down box from which supported devices may be selected.

Programmer

Read Device- Reads program memory, data EEPROM memory, ID locations and Configuration bits.

Write Device- Writes program memory, data EEPROM memory, ID locations and Configuration bits.

Verify- Verifies program memory, data EEPROM memory, ID locations and Configuration bits read from the target MCU against the code stored in the programming application.

Erase- Performs a Bulk Erase of the target MCU. OSCCAL and band gap values are preserved on parts with these features.

Blank Check- Performs a Blank Check of program memory, data EEPROM memory, ID locations and Configuration bits.

Verify on Write- When checked, the device will be immediately verified after programming on a Write (recommended). When unchecked, the device will be programmed but not verified on a Write.

Hold Device in Reset- When checked, the MCLR (VPP) pin is held low (asserted). When unchecked, the pin is released (tri-stated), allowing an external pull-up to bring the device out of Reset.

Write on PICkit Button- When checked, a Write operation will be initiated by pressing the PICkit 2 push button.

Tools

Enable Code Protect – Enables code protection features of the microcontroller on future Write operations.

Set OSCCAL- Allows the OSCCAL value to be changed for devices where it is stored in the last location of Program Memory.

Target VDD Source

- Auto-Detect- The PICkit 2 will automatically detect whether the target devicehas its own power supply or needs to be powered by the programmer on each operation.
- Force PICkit 2- The PICkit 2 will always attempt to supply VDD to the target device.
- Force Target- The PICkit 2 will always assume the target has its own power supply.

Calibrate VDD & Set Unit ID- Opens a wizard that steps the user through calibrating the PICkit 2 VDD supplied voltage so it is more accurate, and optionally assigning a Unit ID to identify between multiple PICkit 2 devices.

Use VPP First Program Entry- When checked, it allows the PICkit 2 to connect to and program devices with configurations and code that interferes with the ICSP signal pins, preventing PICkit 2 from detecting them. Using this feature requires that the PICkit 2 supplies VDD to the target.

Fast Programming- When checked, the PICkit 2 will attempt to program the device as fast as possible. When unchecked, the PICkit 2 will slow down ICSP communication. This may be helpful for targets with loaded ICSP lines.

Check Communication- Verifies USB communication with the PICkit 2 and ICSP communication with a target device by attempting to identify the connected device by its device ID.

UART Tool- Puts the PICkit 2 in UART Mode and opens a terminal-like interface for communicating with a PIC MCU device program through the USART pins.

Troubleshoot- Opens a wizard to help with troubleshooting connectivity from the PICkit 2 to the target device. This is most useful where the programmer is unable to detect the target device at all.

Download PICkit 2 Programmer Operating System-Performs a download of the PICkit 2 operating system (firmware).

3.1.2 VOLTAGE REGULATOR (IC 7805)

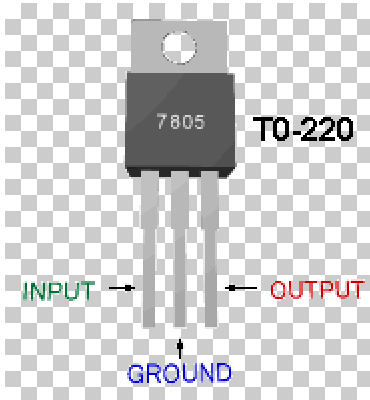


FIGURE-10 IC 7805

Features:

- **IC 7805** is a **5V Voltage Regulator** that restricts the voltage output to 5V and draws 5V regulated power supply.
- It comes with provision to add heat sink. The maximum value for input to the voltage regulator is 35V.
- It can provide a constant steady voltage flow of 5V for higher voltage input till the threshold limit of 35V.
- If the voltage is near to 7.5V then it does not produce any heat and hence no need for heat sink. If the voltage input is more, then excess electricity is liberated as heat from 7805.
- It regulates a steady output of 5V if the input voltage is in rage of 7.2V to 35V. Hence to avoid power loss try to maintain the input to 7.2V.

3.1.3 SEVEN SEGMENT DISPLAY

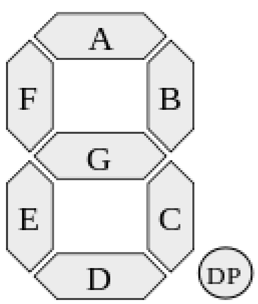


FIGURE-11 SEVEN SEGMENT DISPLAY

- Four common cathode 7-segment displays are used in this 4-way traffic light control system. Figure shows a segment LED display and its pin description.

- It is a 10-pin display device. The pins are: a, b, c, d, e, f, g, DP (dot) and com. The pins labeled as "com" are internally connected to each other.
- The digits displayed in every 7-segment range from 0 to 9. Since the required numbers to display are more than 9, two 7-segment LED displays (left and right) were used for every traffic light and this is why 4 displays were used in this project (a pair for each opposite lanes). The data pins (a, b, c, d, e, f and g) for the left and right displays in every strip board are connected to each other because they are multiplexed.
- Multiplexing is made by using a 3 to 8 decoder. The IC (74LS47) is used to select which 7 segment to show the digit. This chip has 4 input pins labeled A, B and C and 8 output pins labeled. The input pins are connected to the microcontroller for it to select which 7 segment to light up.

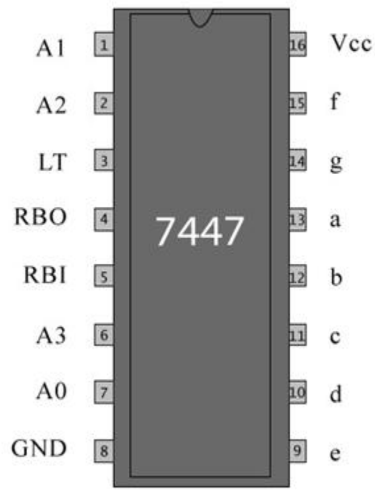


FIGURE-12 7447 PIN DIAGRAM

TABLE-1 HEX CODES TO DISPLAY VARIOUS DIGITS

Digit	gfedcba	abcdefg	a	b	c	d	e	f	g
0	0x3F	0x7E	**on**	**on**	**on**	**on**	**on**	**on**	off
1	0x06	0x30	off	**on**	**on**	off	off	off	off
2	0x5B	0x6D	**on**	**on**	off	**on**	**on**	off	**on**
3	0x4F	0x79	**on**	**on**	**on**	**on**	off	off	**on**
4	0x66	0x33	off	**on**	**on**	off	off	**on**	**on**
5	0x6D	0x5B	**on**	off	**on**	**on**	off	**on**	**on**
6	0x7D	0x5F	**on**	off	**on**	**on**	**on**	**on**	**on**
7	0x07	0x70	**on**	**on**	**on**	off	off	off	off
8	0x7F	0x7F	**on**	**on**	**on**	**on**	**on**	**on**	**on**
9	0x6F	0x7B	**on**	**on**	**on**	**on**	off	**on**	**on**

3.1.4 RESISTOR

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In this circuit 330 ohm and 10 kohm resisters are used.

3.1.5 CAPACITOR

A capacitor (originally known as a condenser) is a passive two-terminal electrical component used to store energy electrostatically in an electric field. This circuit uses a 2 uF capacitor.

3.1.6 BERG CONNECTOR

A Berg connector is a brand of electrical connector used in computer hardware.

3.1.7 OSCILLATOR

An electronic oscillator is an electronic circuit that produces a repetitive, oscillating electronic signal, often a sine wave or a square wave. Oscillators convert direct current (DC) from a power supply to an alternating current signal. This circuit uses a 2 MHz oscillator.

3.1.9 LED

Light emitting diodes **(**LEDs**)** are semiconductor light sources. The light emitted from LED**s** varies from visible to infrared and ultraviolet regions. They operate on low voltage and power. LEDs are one of the most common electronic components and are mostly used as indicators in circuits. They are also used for luminance and optoelectronic applications.

Three colors of LED are used in this project. They are:

- RED
- YELLOW
- GREEN

Chapter-4 SOFTWARE DEVELOPMENT TOOLS

The tools for developing software and hardware for microcontroller-based systems include editors, assemblers, compilers, debuggers, simulators, emulators, and device programmers. A typical development cycle starts with writing the application program using a text editor. The program is then translated into an executable code with the help of an assembler or compiler. If the program has several modules, a linker is used to combine them into a single application. Any syntax errors are detected by the assembler or compiler and must be corrected before the executable code can be generated. Next, a simulator is used to test the application program without the target hardware. Simulators are helpful in checking the correctness of an algorithm or a program with limited or no input-outputs, and most errors can be removed during simulation. Once the program seems to be working and the programmer is happy with it, the executable code is loaded to the target microcontroller chip using a device programmer, and the system logic is tested. Software and hardware tools such as in-circuit debuggers and in-circuit emulators can analyze the program’s operation and display the variables and registers in real time with the help of breakpoints set in the program. Software development tools are computer programs, usually run on personal computers, that allow the programmer (or system developer) to create, modify, and test applications programs. Some common software development tools are:

- Text editors
- Assemblers/compilers
- Simulators
- High-level language simulators
- Integrated development environments (IDEs)

4.1 SOFTWARES USED

4.1.1 MIKROC PRO IDE

mikroC is a powerful , feature rich development tool for PIC micros. It is designed to provide the customer with the easiest possible solution for developing applications for embedded systems, without compromising performance or control.

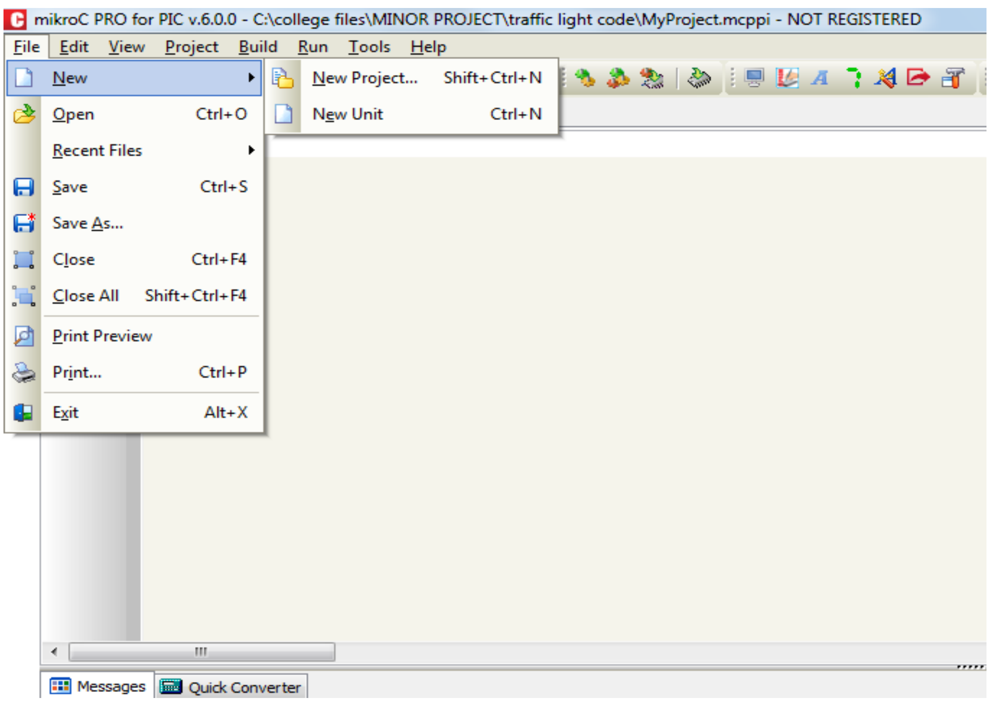
PIC and C fit together well: PIC is the most popular 8-bit chip in the world, used in a wide variety of applications, and C, prized for its efficiency, is the natural choice for developing embedded systems. mikroC provides a successful match featuring highly advanced IDE, ANSI complaint compiler, broad set of hardware libraries, comprehensive documentation, and plenty of ready-to-run applications.

mikroC allows us to quickly develop and deploy complex applications:

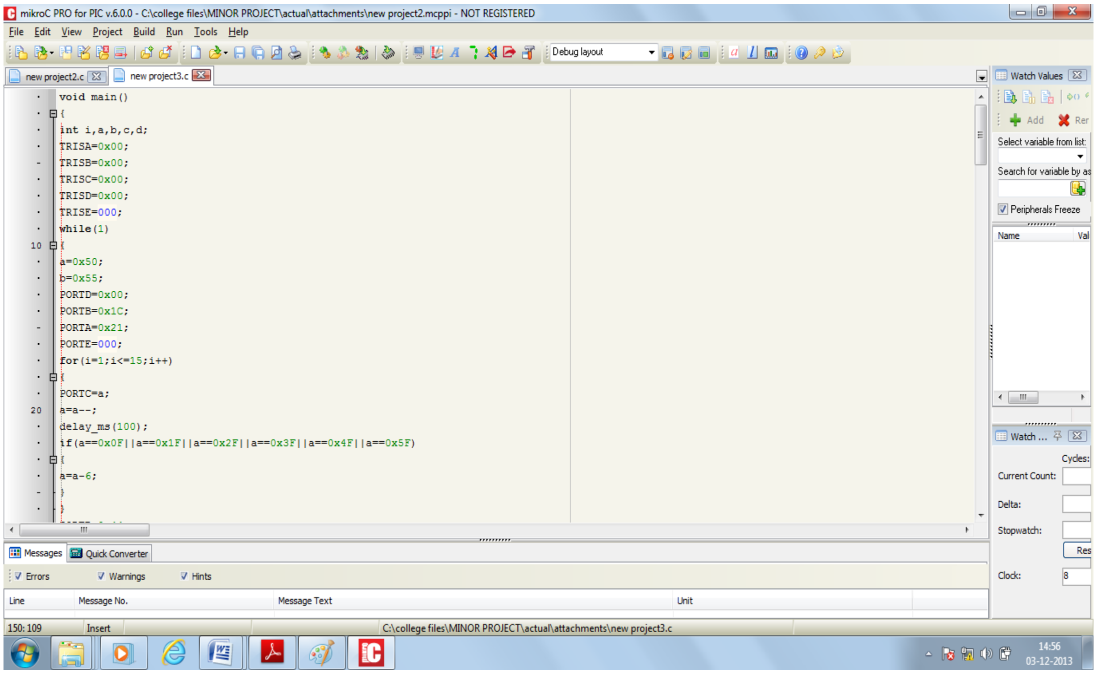
- Writing C code using highly advanced Code Editor
- It includes design libraries to dramatically speed up the development: data acquisition, memory, displays, conversions, communications.
- Monitoring program structures, variables and functions in the code explorer. Generates commented, human-readable assembly, and standard HEX compatible with all programmers.
- Inspecting program flow and debugging executable logic with the integrated debugger. Getting detailed reports and graphs on code statistics, assembly listings, calling tree.

HOW TO USE mikroC PRO

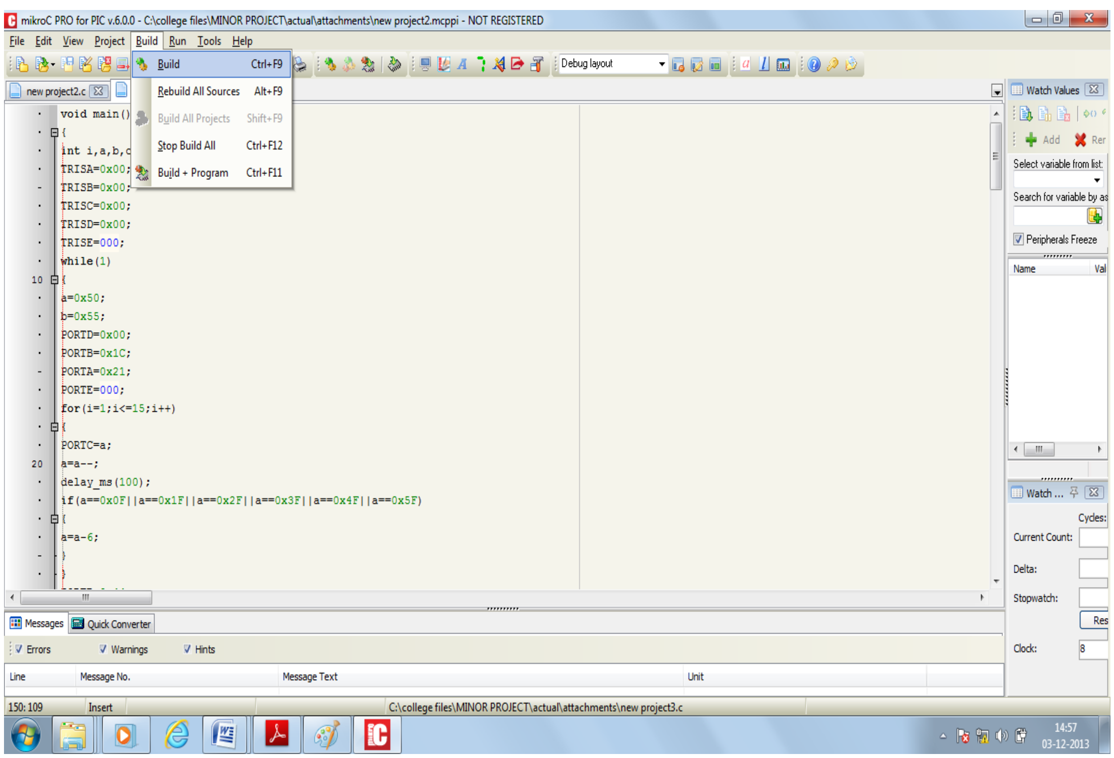
- Open mikroC, go to "File", then "New" for new project or "Open" for old project.



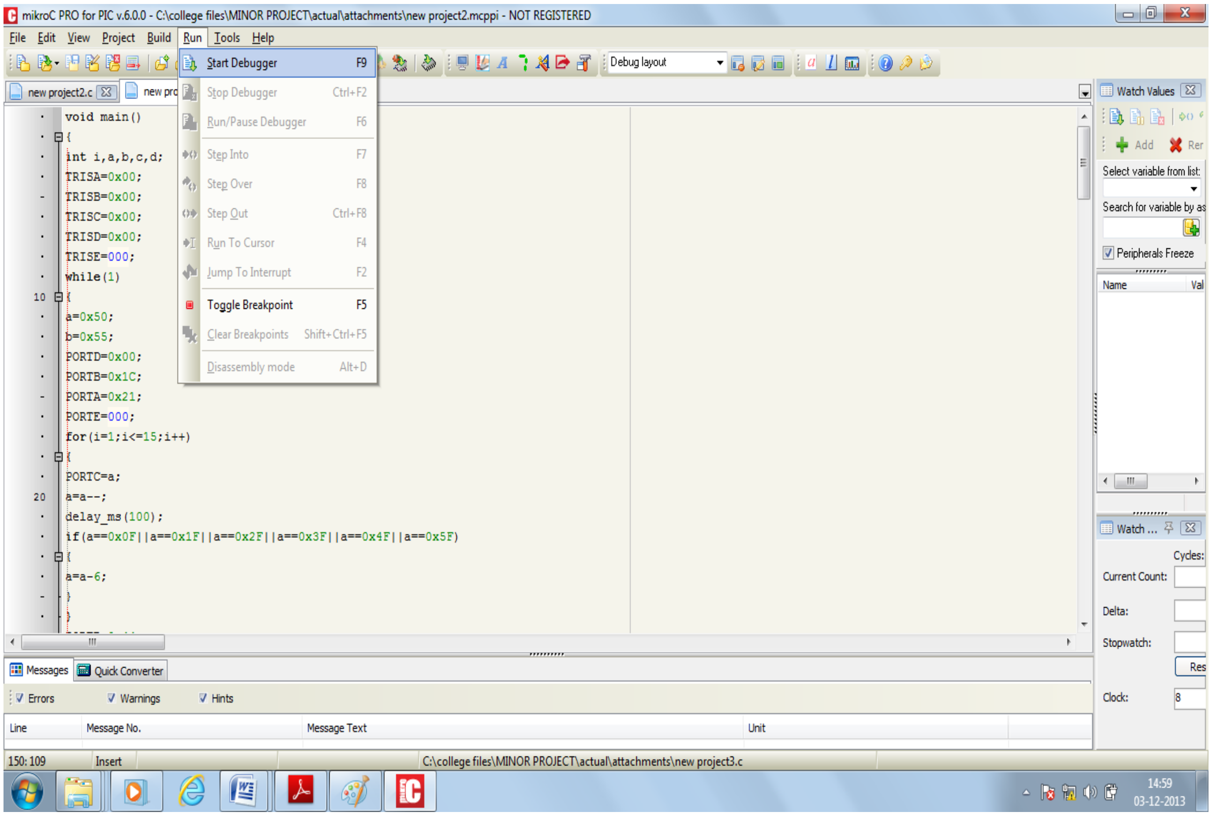
- Then write code for the required application.



- Then go to “build” to compile the file and create the hex file to be loaded into the microcontroller.



- **Select the "run" option to start the debugging process and run the program.**



4.1.2 PROTEUS EDA

Proteus v7.6 is software for microprocessor simulation, schematic capture, and printed circuit board (PCB) design. It is developed by Labcenter Electronics.

System components:

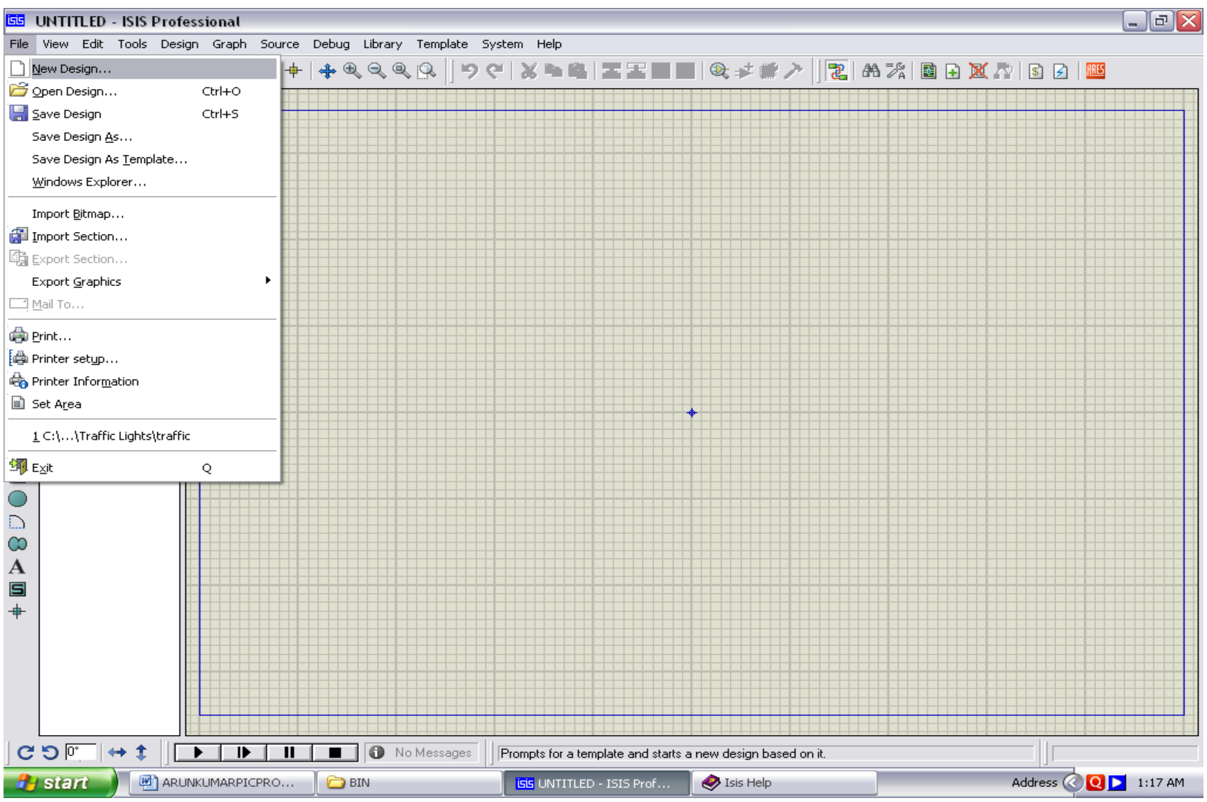
- **ISIS Schematic Capture** - a tool for entering designs.
- **PROSPICE Mixed mode SPICE simulation** - industry standard SPICE3F5 simulator combined with a digital simulator.
- **ARES PCB Layout** - PCB design system with automatic component placer, rip-up and retry auto-router and interactive design rule checking.
- **VSM** - Virtual System Modelling lets co-simulate embedded software for popular micro-controllers alongside hardware design.
- **System Benefits** Integrated package with common user interface and fully context sensitive help.

Other general features include:

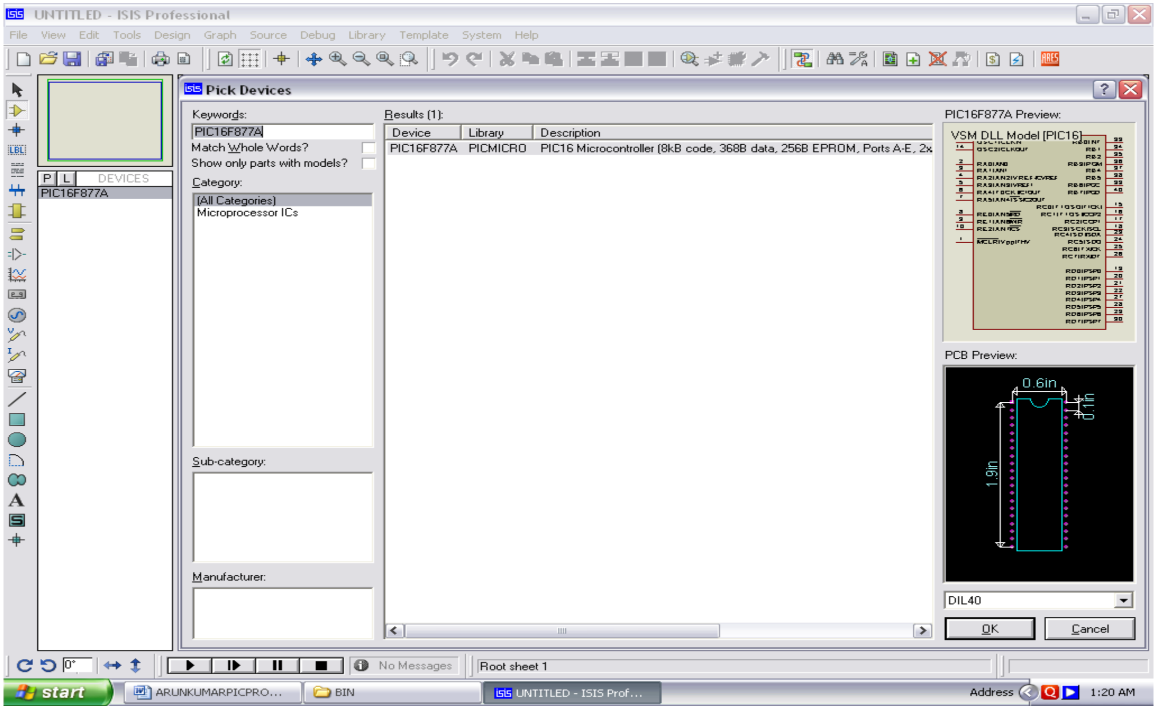
- Runs on Windows 2k and XP.
- Automatic wire routing and dot placement/removal.
- Powerful tools for selecting objects and assigning their properties.
- Total support for buses including component pins, inter-sheet terminals, module ports and wires.
- Bill of Materials and Electrical Rules Check reports.
- Netlist outputs to suit all popular PCB layout tools.

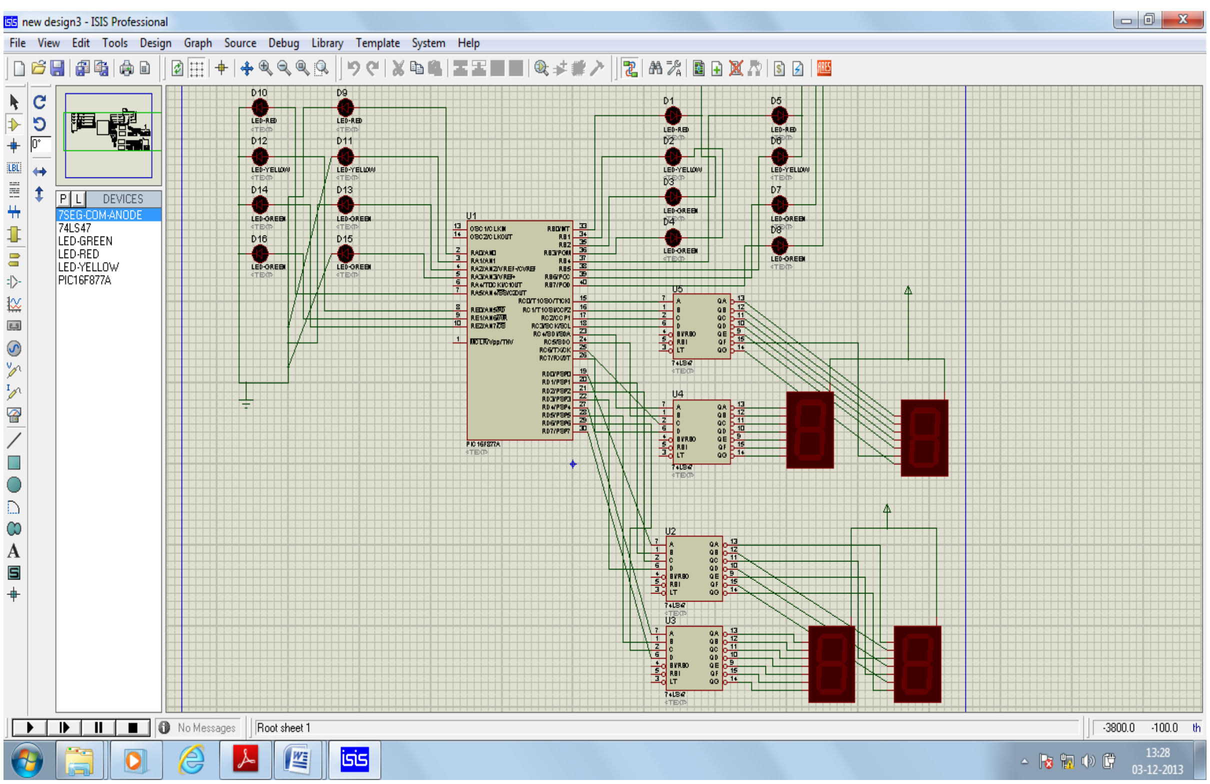
How to create & simulate design in proteus isis?

- Firstly prepare a code having required function implementation & prepare its “Hex File”.
- Open proteus, go to “File” then “New Design” then select template.

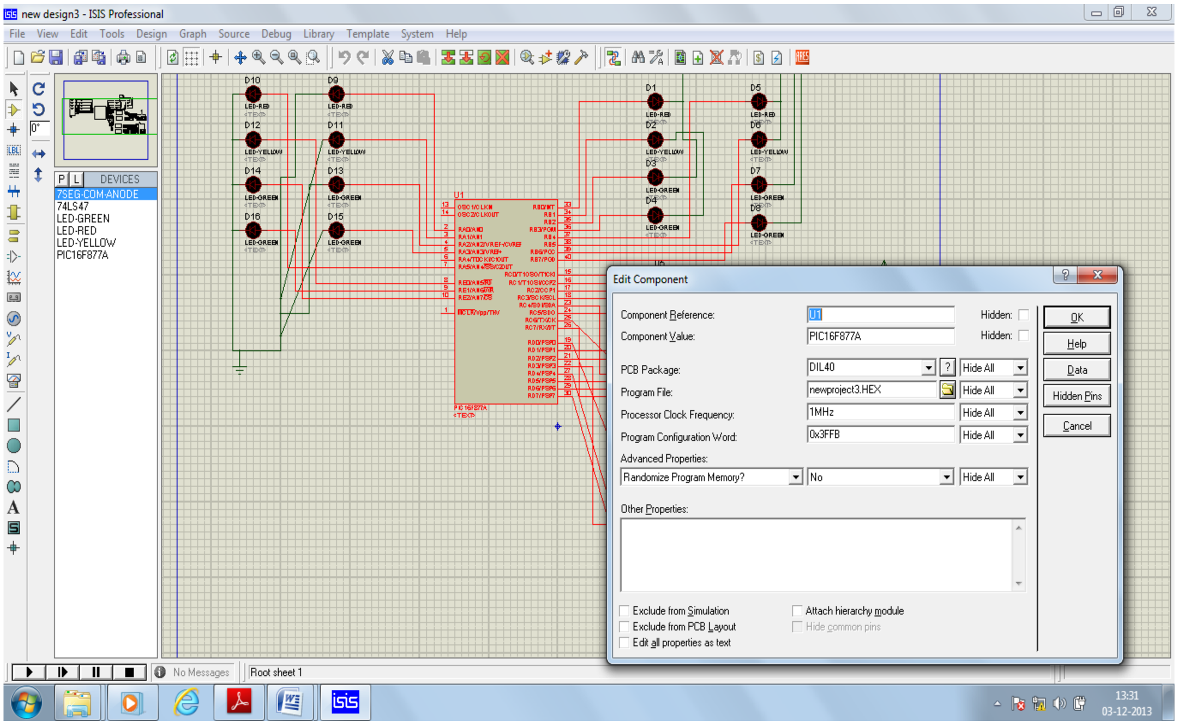


- Then place components (microcontroller and other equipments)& make connections between them.

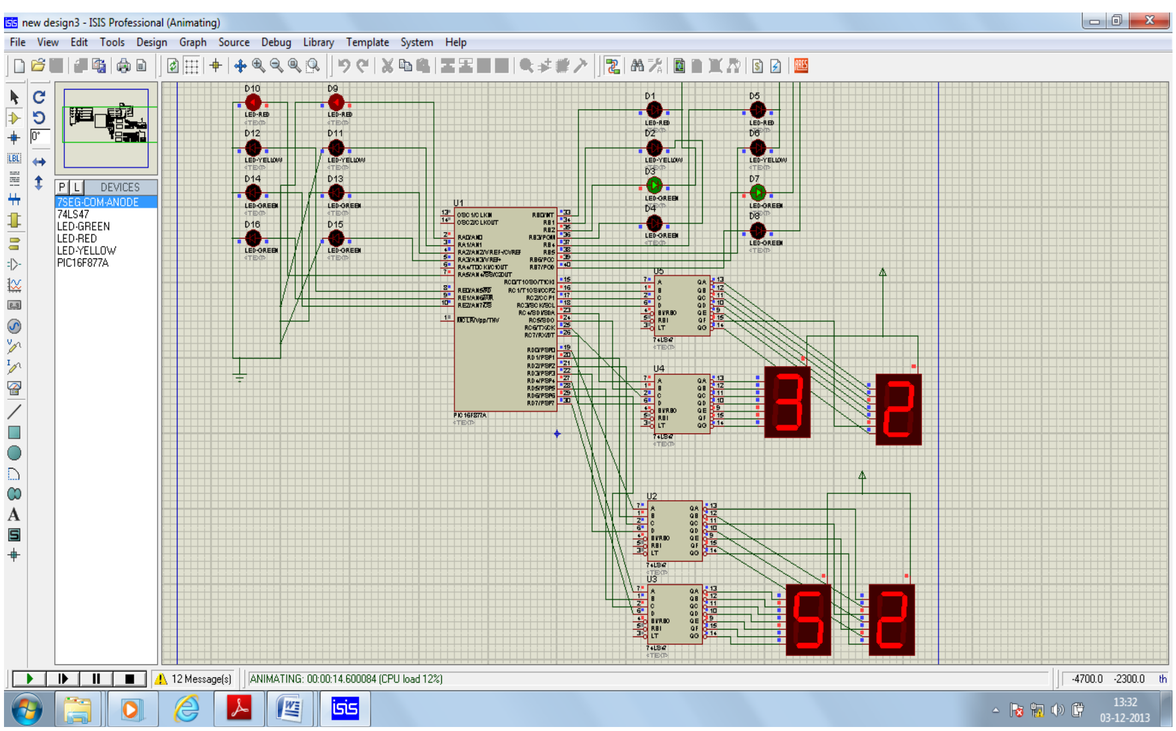




- Now, click on microcontroller & load the hex file in its memory.



- Now run your project by using the controls in left lower side of window.



CONCLUSION

- During our minor project we developed a **PIC microcontroller based TRAFFIC LIGHT CONTROLLER** to tackle the vehicular traffic on roads.

- This project led us to the following outcomes:
  1. A traffic light controller eases the congestion on roads.
  2. Concurrently operating 4-way traffic lane reduces the waiting time for commuters.
  3. Microcontroller based traffic control is much more reliable than a traffic police.
  4. The operational cost of this device is reasonable.

FUTURE SCOPE

This project can be enhanced in such a way as to automatically control the signals depending on the traffic density on the roads using sensors like IR detector/receiver module extended with automatic turn off when no vehicles are running on any side of the road which helps in power consumption saving.

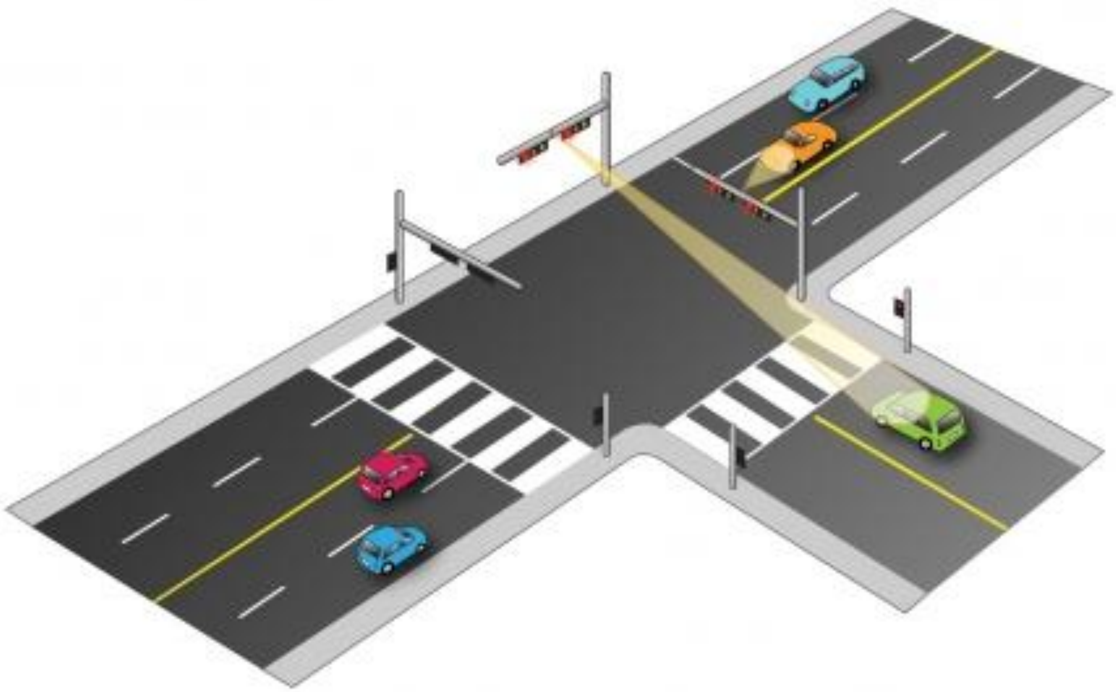


FIGURE-13 TRAFFIC LIGHT CONTROLLER WITH SENSORS

REFRENCES

[1] John B.Peatman- PIC Microcontroller design, 1st edition: Pearsons Education; 1997

[2] Muhammad Ali Mazidi- PIC Microcontroller and Embedded systems, 3rd edition: Pearsons Education; 2007

[3] Proteus All-in-One Manual

[4] Iovine John- PIC Microcontroller Project Book, 2nd Edition, Singapore: McGraw Hill 121-123; 2000.

[5] Lawrence A. Duarte- The Microcontroller Beginner's Handbook, 2nd Edition, United States of America: Prompt Publication. 3-5; 1998.

[6] MPLAB IDE, Simulator, Editor User's Guide

[7] http://www.seattlerobotics.org/encoder

[8] http://www.microchip.com

[9] http://robotaaly.blogspot.in/2013/02/traffic-light-control-system.html

[10] http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en010242

[11] hhttp://www.slideshare.net/deepu671/demo-traffic-light-in-pic16f87

[12] http://ww1.microchip.com/downloads/en/devicedoc/39582b.pdf

APPENDIX A

SOURCE CODE

```
void main()
{
int i,a,b,c,d;
TRISA=0x00;
TRISB=0x00;
TRISC=0x00;
TRISD=0x00;
TRISE=000;
while(1)
{
a=0x50;
b=0x55;
PORTD=0x00;
PORTB=0x1C;
PORTA=0x21;
PORTE=000;
for(i=1;i<=15;i++)
{
PORTC=a;
a=a--;
delay_ms(100);
if(a==0x0F||a==0x1F||a==0x2F||a==0x3F||a==0x4F||a==0x5F)
{
a=a-6;
}
}
PORTB=0x44;
for(i=1;i<=35;i++)
{
PORTC=a;
PORTD=b;
a=a--;
b=b--;
delay_ms(100);
if(a==0x0F||a==0x1F||a==0x2F||a==0x3F||a==0x4F||a==0x5F)
{
a=a-6;
}
if(b==0x0F||b==0x1F||b==0x2F||b==0x3F||b==0x4F||b==0x5F)
{
```

```
b=b-6;
}
}
a=0x05;
PORTB=0x42;
for(i=1;i<=5;i++)
{
PORTC=a;
PORTD=b;
a=a--;
b=b--;
delay_ms(100);
if(a==0x0F||a==0x1F||a==0x2F||a==0x3F||a==0x4F||a==0x5F)
{
a=a-6;
}
if(b==0x0F||b==0x1F||b==0x2F||b==0x3F||b==0x4F||b==0x5F)
{
b=b-6;
}
}
PORTC=0x00;
PORTB=0xC1;
for(i=1;i<=15;i++)
{
PORTD=b;
b=b--;
delay_ms(100);
if(b==0x0F||b==0x1F||b==0x2F||b==0x3F||b==0x4F||b==0x5F)
{
b=b-6;
}
}
PORTB=0x21;
b=0x05;
for(i=1;i<=5;i++)
{
PORTD=b;
b=b--;
delay_ms(100);
if(b==0x0F||b==0x1F||b==0x2F||b==0x3F||b==0x4F||b==0x5F)
{
b=b-6;
}
}
PORTD=0x00;
```

```
c=0x50;
d=0x55;
PORTD=0x00;
PORTB=0x11;
PORTA=0x2c;
PORTE=000;
for(i=1;i<=15;i++)
{
PORTC=c;
c=c--;
delay_ms(100);
if(c==0x0F||c==0x1F||c==0x2F||c==0x3F||c==0x4F||c==0x5F)
{
c=c-6;
}
}
PORTA=0x04;
PORTE=100;
for(i=1;i<=35;i++)
{
PORTC=c;
PORTD=d;
c=c--;
d=d--;
delay_ms(100);
if(c==0x0F||c==0x1F||c==0x2F||c==0x3F||c==0x4F||c==0x5F)
{
c=c-6;
}
if(d==0x0F||d==0x1F||d==0x2F||d==0x3F||d==0x4F||d==0x5F)
{
d=d-6;
}
}
c=0x05;
PORTA=0x02;
PORTE=100;
for(i=1;i<=5;i++)
{
PORTC=c;
PORTD=d;
c=c--;
d=d--;
delay_ms(100);
if(c==0x0F||c==0x1F||c==0x2F||c==0x3F||c==0x4F||c==0x5F)
{
```

```
c=c-6;
}
if(d==0x0F||d==0x1F||d==0x2F||d==0x3F||d==0x4F||d==0x5F)
{
d=d-6;
}
}
PORTC=0x00;
PORTA=0x01;
PORTE=110;
for(i=1;i<=15;i++)
{
PORTD=d;
d=d--;
delay_ms(100);
if(d==0x0F||d==0x1F||d==0x2F||d==0x3F||d==0x4F||d==0x5F)
{
d=d-6;
}
}
PORTA=0x01;
PORTE=001;
d=0x05;
for(i=1;i<=5;i++)
{
PORTD=d;
d=d--;
delay_ms(100);
if(d==0x0F||d==0x1F||d==0x2F||d==0x3F||d==0x4F||d==0x5F)
{
d=d-6;
}
}
PORTD=0x00;
}
}
```

APPENDIX B

TABLE-2 LIST OF HARDWARE COMPONENTS

COMPONENT	QUANTITY
LED • RED • YELLOW • GREEN	 4 4 8
SEVEN SEGMENT DISPLAY	4
IC 7447	4
IC 7805	1
OSCILLATOR(20MHz)	1
REGISTERS • 330 ohm • 10 Kilo ohm	 10 1
CAPACITOR(22 pf)	2
DC JACK	1

APPENDIX C

PIC DATASHEET

DEVICE OVERVIEW

This document contains device specific information about the following devices:

- PIC16F873A
- PIC16F874A
- PIC16F876A
- PIC16F877A

PIC16F873A/876A devices are available only in 28-pin packages, while PIC16F874A/877A devices are available in 40-pin and 44-pin packages.

All devices in the PIC16F87XA family share common architecture with the following differences:

- The PIC16F873A and PIC16F874A have one-half of the total on-chip memory of the PIC16F876A and PIC16F877A/44-pin devices have five
- The 28-pin devices have fourteen interrupts, while the 40/44-pin devices have fifteen
- The 28-pin devices have five A/D input channels, while the 40/44-pin devices have eight
- The Parallel Slave Port is implemented only on the 40/44-pin devices

TABLE-3 PIC 16F87XA DEVICE FEATURES

Key Features	PIC16F873A	PIC16F874A	PIC16F876A	PIC16F877A
Operating Frequency	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz	DC – 20 MHz
Resets (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
Flash Program Memory (14- bit words)	4K	4K	8K	8 K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory (bytes)	128	128	256	256
Interrupts	14	15	14	15
I/O Ports	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C	Ports A, B, C, D, E
Timers	3	3	3	3
Capture/Compare/PWM modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Analog Comparators	2	2	2	2
Instruction Set	35 Instructions	35 Instructions	35 Instructions	35 Instructions
Packages	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40-pin PDIP 44-pin PLCC 44-pin TQFP 44-pin QFN	28-pin PDIP 28-pin SOIC 28-pin SSOP 28-pin QFN	40- pin PDIP 44- pin PLCC 44- pin TQFP 44- pin QFN

TABLE-4 PIC16F8777A PIN OUT DESCRIPTION

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKI	13	14	30	32		ST/CMOS	Oscillator crystal or external clock input.
OSC1					I		Oscillator crystal input or external clock source input. ST buffer when configured in RC mode; otherwise CMOS.
CLKI					I		External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO	14	15	31	33		—	Oscillator crystal or clock output.
OSC2					O		Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode.
CLKO					O		In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
MCLR/VPP	1	2	18	18		ST	Master Clear (input) or programming voltage (output).
$\overline{\text{MCLR}}$					I		Master Clear (Reset) input. This pin is an active low Reset to the device.
VPP					P		Programming voltage input.
RA0/AN0	2	3	19	19		TTL	PORTA is a bidirectional I/O port.
RA0					I/O		Digital I/O.
AN0					I		Analog input 0.
RA1/AN1	3	4	20	20		TTL	
RA1					I/O		Digital I/O.
AN1					I		Analog input 1.
RA2/AN2/VREF-/CVREF	4	5	21	21		TTL	
RA2					I/O		Digital I/O.
AN2					I		Analog input 2.
VREF-					I		A/D reference voltage (Low) input.
CVREF					O		Comparator VREF output.
RA3/AN3/VREF+	5	6	22	22		TTL	
RA3					I/O		Digital I/O.
AN3					I		Analog input 3.
VREF+					I		A/D reference voltage (High) input.
RA4/T0CKI/C1OUT	6	7	23	23		ST	
RA4					I/O		Digital I/O – Open-drain when configured as output.
T0CKI					I		Timer0 external clock input.
C1OUT					O		Comparator 1 output.
RA5/AN4/$\overline{\text{SS}}$/C2OUT	7	8	24	24		TTL	
RA5					I/O		Digital I/O.
AN4					I		Analog input 4.
$\overline{\text{SS}}$					I		SPI slave select input.
C2OUT					O		Comparator 2 output.

TABLE-4 PIC16F8777A PIN OUT DESCRIPTION(CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RB0/INT	33	36	8	9		TTL/ST	
RB0					I/O		Digital I/O.
INT					I		External interrupt.
RB1	34	37	9	10	I/O	TTL	Digital I/O.
RB2	35	38	10	11	I/O	TTL	Digital I/O.
RB3/PGM	36	39	11	12	I/O	TTL	Digital I/O.
RB3					I		Low-voltage ICSP programming
PGM							enable pin.
RB4	37	41	14	14	I/O	TTL	Digital I/O.
RB5	38	42	15	15	I/O	TTL	Digital I/O.
RB6/PGC	39	43	16	16	I/O	TTL/ST	Digital I/O.
RB6					I		In-circuit debugger and ICSP
PGC							programming clock.
RB7/PGD	40	44	17	17	I/O	TTL/ST	Digital I/O.
RB7					I/O		In-circuit debugger and ICSP
PGD							programming data.

TABLE-4 PIC16F8777A PIN OUT DESCRIPTION(CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RC0/T1OSO/T1CKI	15	16	32	34	I/O	ST	Digital I/O.
RC0					O		Timer1 oscillator output.
T1OSO					I		Timer1 external clock input.
T1CKI							
RC1/T1OSI/CCP2	16	18	35	35	I/O	ST	Digital I/O.
RC1					I		Timer1 oscillator input.
T1OSI					I/O		Capture2 input, Compare2 output, PWM2 output.
CCP2							
RC2/CCP1	17	19	36	36	I/O	ST	Digital I/O.
RC2					I/O		Capture1 input, Compare1 output, PWM1 output.
CCP1							
RC3/SCK/SCL	18	20	37	37	I/O	ST	Digital I/O.
RC3					I/O		Synchronous serial clock input/output for SPI mode.
SCK							
SCL					I/O		Synchronous serial clock input/output for I^2C mode.
RC4/SDI/SDA	23	25	42	42		ST	
RC4					I/O		Digital I/O.
SDI					I		SPI data in.
SDA					I/O		I^2C data I/O.
RC5/SDO	24	26	43	43		ST	
RC5					I/O		Digital I/O.
SDO					O		SPI data out.
RC6/TX/CK	25	27	44	44	I/O	ST	Digital I/O.
RC6					O		USART asynchronous transmit.
TX							
CK					I/O		USART1 synchronous clock.
RC7/RX/DT	26	29	1	1	I/O	ST	Digital I/O.
RC7					I		USART asynchronous receive.
RX							
DT					I/O		USART synchronous data.

TABLE-4 PIC16F8777A PIN OUT DESCRIPTION(CONTINUED)

Pin Name	PDIP Pin#	PLCC Pin#	TQFP Pin#	QFN Pin#	I/O/P Type	Buffer Type	Description
RD0/PSP0 RD0 PSP0	19	21	38	38	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD1/PSP1 RD1 PSP1	20	22	39	39	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD2/PSP2 RD2 PSP2	21	23	40	40	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD3/PSP3 RD3 PSP3	22	24	41	41	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD4/PSP4 RD4 PSP4	27	30	2	2	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD5/PSP5 RD5 PSP5	28	31	3	3	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD6/PSP6 RD6 PSP6	29	32	4	4	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RD7/PSP7 RD7 PSP7	30	33	5	5	I/O I/O	ST/TTL	Digital I/O. Parallel Slave Port data.
RE0/RD/AN5	8	9	25	25		ST/TTL	
RE0					I/O		Digital I/O.
RD					I		Read control for Parallel Slave Port. Analog
AN5	9	10	26	26	I	ST/TTL	input 5.
RE1/WR/AN6							
RE1					I/O		Digital I/O.
WR					I		Write control for Parallel Slave Port.
AN6	10	11	27	27	I	ST/TTL	Analog input 6.
RE2/CS/AN7							
RE2					I/O		Digital I/O.
CS					I		Chip select control for Parallel Slave Port.
AN7					I		Analog input 7.
VSS	12, 31	13, 34	6, 29	6, 30, 31	P	—	Ground reference for logic and I/O pins.
VDD	11, 32	12, 35	7, 28	7, 8, 28 , 29	P	—	Positive supply for logic and I/O pins.
NC	—	1, 17, 28, 40	12,13, 33, 34	13	—	—	These pins are not internally connected. These pins should be left unconnected.

TABLE-5 PIC 16F877A INSTRUCTION SET

Mnemonic, Operands		Description	Cycles	14-Bit Opcode				Status Affected	Notes
				MSb			LSb		
BYTE-ORIENTED FILE REGISTER OPERATIONS									
ADDWF	f, d	Add W and f	1	00	0111	dfff	ffff	C,DC,Z	1,2
ANDWF	f, d	AND W with f	1	00	0101	dfff	ffff	Z	1,2
CLRF	f	Clear f	1	00	0001	1fff	ffff	Z	2
CLRW	-	Clear W	1	00	0001	0xxx	xxxx	Z	
COMF	f, d	Complement f	1	00	1001	dfff	ffff	Z	1,2
DECF	f, d	Decrement f	1	00	0011	dfff	ffff	Z	1,2
DECFSZ	f, d	Decrement f, Skip if 0	1(2)	00	1011	dfff	ffff		1,2,3
INCF	f, d	Increment f	1	00	1010	dfff	ffff	Z	1,2
INCFSZ	f, d	Increment f, Skip if 0	1(2)	00	1111	dfff	ffff		1,2,3
IORWF	f, d	Inclusive OR W with f	1	00	0100	dfff	ffff	Z	1,2
MOVF	f, d	Move f	1	00	1000	dfff	ffff	Z	1,2
MOVWF	f	Move W to f	1	00	0000	1fff	ffff		
NOP	-	No Operation	1	00	0000	0xx0	0000		
RLF	f, d	Rotate Left f through Carry	1	00	1101	dfff	ffff	C	1,2
RRF	f, d	Rotate Right f through Carry	1	00	1100	dfff	ffff	C	1,2
SUBWF	f, d	Subtract W from f	1	00	0010	dfff	ffff	C,DC,Z	1,2
SWAPF	f, d	Swap nibbles in f	1	00	1110	dfff	ffff		1,2
XORWF	f, d	Exclusive OR W with f	1	00	0110	dfff	ffff	Z	1,2
BIT-ORIENTED FILE REGISTER OPERATIONS									
BCF	f, b	Bit Clear f	1	01	00bb	bfff	ffff		1,2
BSF	f, b	Bit Set f	1	01	01bb	bfff	ffff		1,2
BTFSC	f, b	Bit Test f, Skip if Clear	1 (2)	01	10bb	bfff	ffff		3
BTFSS	f, b	Bit Test f, Skip if Set	1 (2)	01	11bb	bfff	ffff		3
LITERAL AND CONTROL OPERATIONS									
ADDLW	k	Add Literal and W	1	11	111x	kkkk	kkkk	C,DC,Z	
ANDLW	k	AND Literal with W	1	11	1001	kkkk	kkkk	Z	
CALL	k	Call Subroutine	2	10	0kkk	kkkk	kkkk		
CLRWDT	-	Clear Watchdog Timer	1	00	0000	0110	0100	$\overline{TO}$,$\overline{PD}$	
GOTO	k	Go to Address	2	10	1kkk	kkkk	kkkk		
IORLW	k	Inclusive OR Literal with W	1	11	1000	kkkk	kkkk	Z	
MOVLW	k	Move Literal to W	1	11	00xx	kkkk	kkkk		
RETFIE	-	Return from Interrupt	2	00	0000	0000	1001		
RETLW	k	Return with Literal in W	2	11	01xx	kkkk	kkkk		
RETURN	-	Return from Subroutine	2	00	0000	0000	1000		
SLEEP	-	Go into Standby mode	1	00	0000	0110	0011	$\overline{TO}$,$\overline{PD}$	
SUBLW	k	Subtract W from Literal	1	11	110x	kkkk	kkkk	C,DC,Z	
XORLW	k	Exclusive OR Literal with W	1	11	1010	kkkk	kkkk	Z	