TEMPERATURE CONTROL SYSYTEM

Electronics and Communication Engineering

Under the supervision of
Dr. GHANSHYAM SINGH

BY
111040- PALASH GUPTA
111060- CHANDERKANT SAPAIA
111085 -AKSHAY GAUTAM

Certificate

This is to certify that project report entitled "Temperature Control System" submitted by "Palash, Chanderkant and Akshay" in partial fulfillment for the award of degree of Bachelor of Technology in Electronics & Communication Engineering to Jaypee University Of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

22 May 2015 **Dr. Ghanshyam Singh**

Professor

Acknowledgements

We take this opportunity to express our profound gratitude and deep regards to our guide Dr. Ghanshyam Singh for his exemplary guidance, monitoring and constant encouragement throughout the course of this project. The blessing, help and guidance given by his time to time shall carry us a long way in the journey of life on which we are about to embark.

The in-time facilities provided by the Electronics department throughout the project development are also equally acknowledgeable.

At the end we would like to express our sincere thanks to all my friends and others who helped us directly or indirectly during this project work.

22 May 2015

Palash Gupta- 111040
Chanderkant Sapaia- 111060
Akshay Gautam- 111085

TABLE OF CONTENTS:

INTRODUCTION

Temperature control is a process in which change of temperature of a space (and objects collectively there within) is measured or otherwise detected, and the passage of heat energy into or out of the space is adjusted to achieve a desired average temperature.

As the name implies, temperature controller is an instrument used to control temperature. The temperature controller takes an input from a temperature sensor and has an output that is connected to a control element such as a heater or fan.

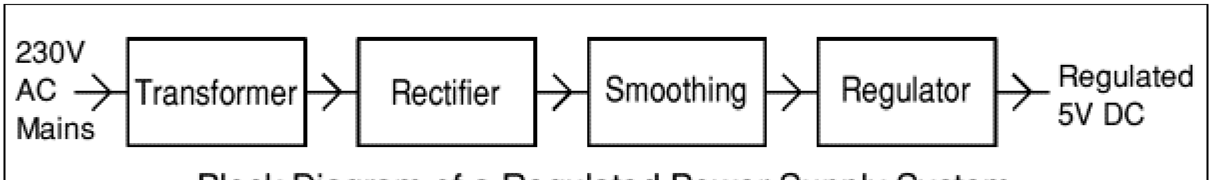
To accurately control process temperature without extensive operator involvement, a temperature control system relies upon a controller, which accepts a temperature sensor such as a thermocouple or RTD as input. It compares the actual temperature to the desired control temperature, or setpoint, and provides an output to a control element. The controller is one part of the entire control system, and the whole system should be analyzed in selecting the proper controller. The following items should be considered when selecting a controller:

1. Type of input sensor (thermocouple, RTD) and temperature range
2. Type of output required (electromechanical relay, SSR, analog output)
3. Control algorithm needed (on/off, proportional, PID)
4. Number and type of outputs (heat, cool, alarm, limit)

Regulated Power Supply

Power supplies are designed to convert high voltage AC mains to a suitable low voltage supply for electronic circuits and other devices. A power supply can be broken down into a series of blocks, each of which performs a particular function.

For example a 5V regulated supply:



Block Diagram of a Regulated Power Supply System

Each of the blocks has its own function as described below

1. Transformer – steps down high voltage AC mains to low voltage AC.
2. Rectifier – converts AC to DC, but the DC output is varying.
3. Smoothing – smoothes the DC from varying greatly to a small ripple.
4. Regulator – eliminates ripple by setting DC output to a fixed voltage.

`Transformer

Transformers convert AC electricity from one voltage to another with little loss of power. Transformers work only with AC and this is one of the reasons why mains electricity is AC. The two types of transformers

- Step-up transformers increase voltage.
- Step-down transformers reduce voltage.

Most power supplies use a step-down transformer to reduce the dangerously high mains voltage (230V in UK) to a safer low voltage. The input coil is called the primary and the output coil is called the secondary.There is no electrical connection between the two coils, instead they are linked by an alternating magnetic field created in the soft-iron core of the transformer. The two lines in the middle of the circuit symbol represent the core. Transformers waste very little power so the power out is (almost) equal to the power in. Note that as voltage is stepped down current is stepped up. The ratio of the number of turns on each coil, called the turn ratio, determines the ratio of the voltages. A step-down transformer has a large number of turns on its primary (input) coil which is connected to the high voltage mains supply, and a small number of turns on its secondary (output) coil to give a low output voltage.

Turns Ratio = $\boldsymbol{Vp/Vs}$ = $\boldsymbol{Np/Ns}$

And Power Out = Power In

Vs *□Is = Vp *IpWhere

Vp = primary (input) voltage

Np = number of turns on primary coil

Ip = primary (input) current

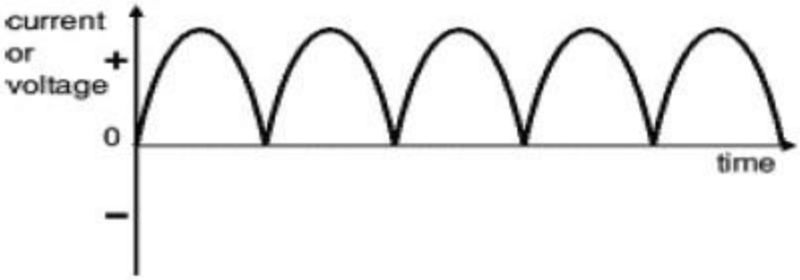
Ns = number of turns on secondary coil

Is = secondary (output) current

Vs = secondary (output) voltage

Bridge Rectifier

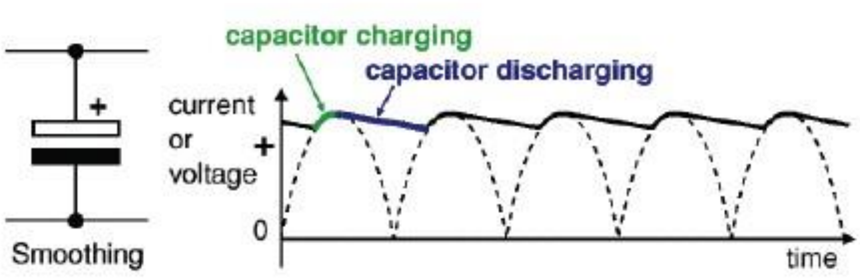
A bridge rectifier can be made using four individual diodes, but it is also available in special packages containing the four diodes required. It is called a full-wave rectifier because it uses all AC wave (both positive and negative sections). 1.4V is used up in the bridge rectifier because each diode uses 0.7V when conducting and there are always two diodes conducting, as shown in the diagram below. Bridge rectifiers are rated by the maximum current they can pass and the maximum reverse voltage they can withstand (this must be at least three times the supply RMS voltage so the rectifier can withstand the peak voltages). In this alternate pairs of diodes conduct, changing over the connections so the alternating directions of AC are converted to the one direction of DC.



OUTPUT − Full-wave Varying DC

SMOOTHING

Smoothing is performed by a large value electrolytic capacitor connected across the DC supply to act as a reservoir, supplying current to the output when the varying DC voltage from the rectifier is falling. The diagram shows the unsmoothed varying DC (dotted line) and the smoothed DC (solid line). The capacitor charges quickly near the peak of the varying DC, and then discharges as it supplies current to the output.



Note that smoothing significantly increases the average DC voltage to almost the peak value (1.4 × RMS value). For example 6V RMS AC is rectified to full wave DC of about 4.6V RMS (1.4V is lost in the bridge rectifier), with smoothing this increases to almost the peak value giving 1.4 × 4.6 = 6.4V smooth DC. Smoothing is not perfect due to the capacitor voltage falling a little as it discharges, giving a small ripple voltage. For many circuits a ripple which is 10% of the supply voltage is satisfactory and the equation below gives the required value for the smoothing capacitor. A larger capacitor will give fewer ripples. The capacitor value must be doubled when smoothing half-wave DC.

Smoothing capacitor for 10% ripple, $C = \frac{5 \times Io}{Vs \times f}$

Where

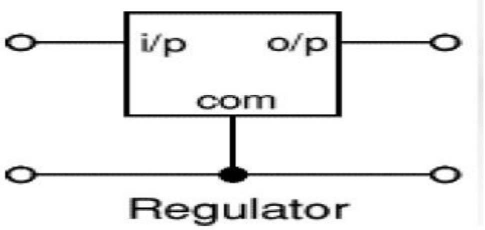
C = smoothing capacitance in farads (F)

Io = output current from the supply in amps (A)

Vs = supply voltage in volts (V), this is the peak value of the unsmoothed DC

f = frequency of the AC supply in hertz (Hz), 50Hz in the UK

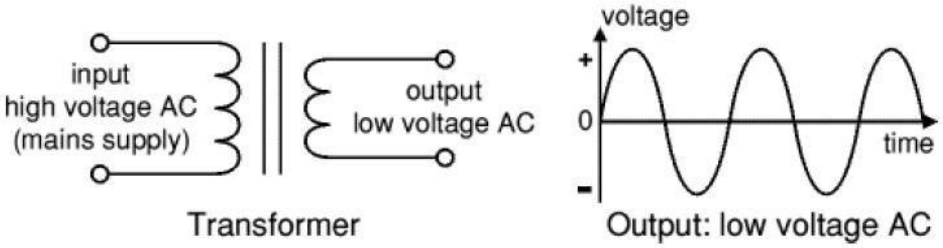
REGULATOR



Voltage regulator ICs are available with fixed (typically 5, 12 and 15V) or variable output voltages. They are also rated by the maximum current they can pass. Negative voltage regulators are available, mainly for use in dual supplies. Most regulators include some automatic protection from excessive current (‘overload protection') and overheating (‘thermal protection'). Many of the fixed voltage regulator ICs has 3 leads and look like power transistors, such as the 7805 +5V 1A regulator shown on the right. They include a hole for attaching a heat sink if necessary.

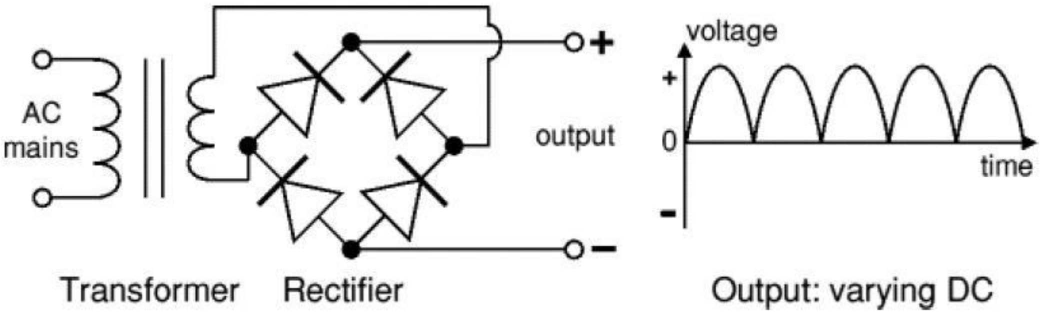
Working of Power Supply

- **Transformer**



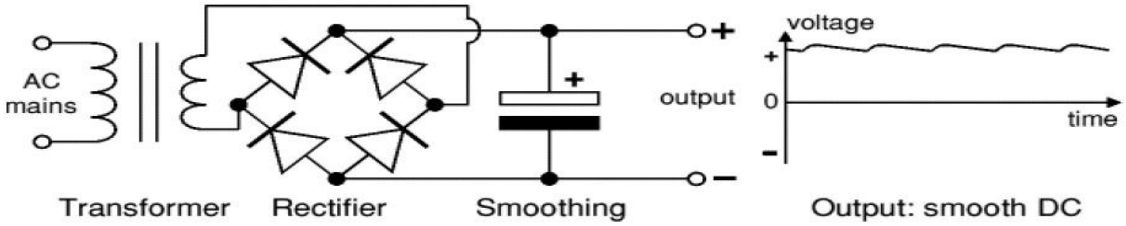
The low voltage AC output is suitable for lamps, heaters and special AC motors. It is not suitable for electronic circuits unless they include a rectifier and a smoothing capacitor.

- **Transformer + Rectifier**



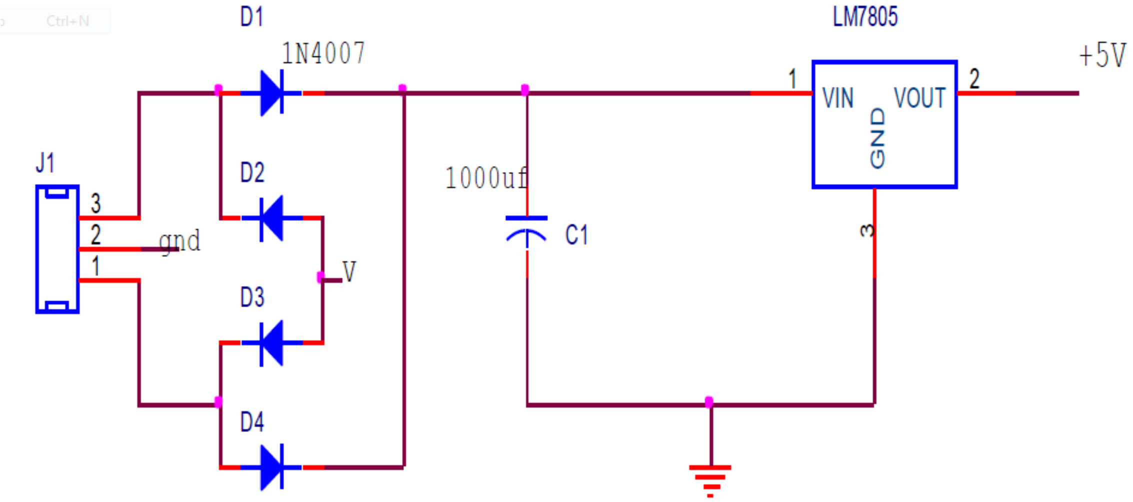
The varying DC output is suitable for lamps, heaters and standard motors. It is not suitable for electronic circuits unless they include a smoothing capacitor.

- **Transformer + Rectifier + Smoothing**



The smooth DC output has a small ripple. It is suitable for most electronic circuits.

- **Transformer + Rectifier + Smoothing + Regulator**



The regulated DC output is very smooth with no ripple. It is suitable for all electronic circuits.

Microcontroller PIC 16FXXX Series

In our day to day life the role of micro-controllers has been immense. They are used in a variety of applications ranging from home appliances, FAX machines, Video games, Camera, Exercise equipment, Cellular phones musical Instruments to Computers, engine control, aeronautics, security systems and the list goes on.

Microcontroller versus Microprocessors

What is the difference between a microprocessor and microcontroller? The microprocessors (such as 8086, 80286, 68000 etc.) contain no RAM, no ROM and no I/O ports on the chip itself. For this reason they are referred as general- purpose microprocessors. A system designer using general- purpose microprocessor must add external RAM, ROM, I/O ports and timers to make them functional. Although the addition of external RAM, ROM, and I/O ports make the system bulkier and much more expensive, they have the advantage of versatility such that the designer can decide on the amount of RAM, ROM and I/o ports needed to fit the task at hand. This is the not the case with microcontrollers. A microcontroller has a CPU (a microprocessor) in addition to the fixed amount of RAM, ROM, I/O ports, and timers are all embedded together on the chip: therefore, the designer cannot add any external memory, I/O, or timer to it. The fixed amount of on chip RAM, ROM, and number of I/O ports in microcontrollers make them ideal for many applications in which cost and space are critical. In many applications, for example a TV remote control, there is no need for the computing power of a 486 or even a 8086 microprocessor. In many applications, the space it takes, the power it consumes, and the price per unit are much more critical considerations than the computing power.

These applications most often require some I/O operations to read signals and turn on and off certain bits. It is interesting to know that some microcontroller‘s manufactures have gone as far as integrating an ADC and other peripherals into the microcontrollers.

Microcontrollers for Embedded Systems

In the literature discussing microprocessors, we often see a term embedded system. Microprocessors and microcontrollers are widely used in embedded system products. An embedded product uses a microprocessor (or microcontroller) to do one task and one task only. A printer is an example of embedded system since the processor inside it performs one task only: namely, get data and print it. Contrasting this with a IBM PC which can be used for a number of applications such as word processor, print server, network server, video game player, or internet terminal. Software for a variety of applications can be loaded and run. Of course the reason a PC can perform myriad tasks is that it has RAM memory and an operating system that loads the application software into RAM and lets the CPU run it. In an embedded system, there is only one application software that is burned into ROM. A PC contains or is connected to various embedded products such as the keyboard, printer, modem, disk controller, sound card, CD-ROM drive, mouse and so on. Each one of these peripherals has a microcontroller inside it that performs only one task. For example, inside every mouse there is a microcontroller to perform the task of finding the mouse position and sending it to the PC. Although microcontrollers are the preferred choice for many embedded systems, there are times that a microcontroller is inadequate for the task. For this reason, in many years the manufacturers for general-purpose microprocessors have targeted their microprocessor for the high end of the embedded market.

PIC Microcontroller

The PIC microcontroller family of microcontrollers is manufactured by *Microchip Technology Inc*. Currently they are one of the most popular microcontrollers used in many commercial and industrial applications. The PIC microcontroller architecture is based on a modified Harvard RISC (Reduced Instruction Set Computer) instruction set with dual-bus architecture, providing fast and flexible design with an easy migration path from only 6 pins to 80 pins, and from 384 bytes to 128 Kbytes of program memory.

PIC microcontrollers are available with many different **specifications** depending on:

- **Memory Type**
  - Flash
  - OTP (One-time-programmable)
  - ROM (Read-only-memory)
  - ROMless
- **Input–Output (I/O) Pin Count**
  - 4–18 pins
  - 20–28 pins
  - 32–44 pins
  - 45 and above pins
- **Memory Size**
  - 0.5–1K
  - 2–4K
  - 8–16K
  - 24–32K
  - 48–64K

– 96–128K

- **Special Features**

– CAN
– USB
– LCD
– Motor Control
– Radio Frequency

Although there are many models of PIC microcontrollers, the nice thing is that they are upward compatible with each other and a program developed for one model can very easily, and in many cases with no modifications, be run on other models of the family. The basic assembler instruction set of PIC microcontrollers consists of only 33 instructions and most of the family members (except the newly developed devices) use the same instruction set. This is why a program developed for one model can run on another model with similar architecture without any changes.

All PIC microcontrollers offer the following features:

- RISC instruction set with only a handful of instructions to learn
- Digital I/O ports
- On-chip timer with 8-bit prescaler
- Power-on reset
- Watchdog timer
- Power saving SLEEP mode
- High source and sink current
- Direct, indirect, and relative addressing modes
- External clock interface
- RAM data memory
- EPROM or Flash program memory

Some devices offer the following additional features:

- Analogue input channels
- Analogue comparators

- Additional timer circuits
- EEPROM data memory
- External and internal interrupts
- Internal oscillator
- Pulse-width modulated (PWM) output
- USART serial interface

Some even more complex devices in the family offer the following additional features:

- CAN bus interface
- I2C bus interface
- SPI bus interface
- Direct LCD interface
- USB interface
- Motor control

Although there are several hundred models of PIC microcontrollers, choosing a microcontroller for an application is not a difficult task and requires taking into account these factors:

- Number of I/O pins required
- Required peripherals (e.g. USART, USB)
- The minimum size of program memory
- The minimum size of RAM
- Whether or not EEPROM non-volatile data memory is required
- Speed
- Physical size
- Cost

The important point to remember is that there could be many models which satisfy all of the above requirements. You should always try to find the model which satisfies your minimum requirements and the one which does not offer more than you may need. For example, if you require a microcontroller with only 8 I/O pins and if there are two identical microcontrollers, one with 8 and the other one with 16 I/O pins, you should select the one with 8 I/O pins. Although there are several hundred models of PIC microcontrollers, the family can be broken down into three main groups, which are:

- 12-bit instruction word (e.g. 12C5XX, 16C5X)
- 14-bit instruction word (e.g. 16F8X, 16F87X)
- 16-bit instruction word (e.g. 17C7XX, 18C2XX)

All three groups share the same RISC architecture and the same instruction set,
with a few additional instructions available for the 14-bit, and many more instructions available for the 16-bit models. Instructions occupy only one word in memory, thus increasing the code efficiency and reducing the required program memory. Instructions and data are transferred on separate buses, thus the overall system performance is increased.

The features of some microcontrollers in each group are given in the following
sections.

16-bit instruction word

16-bit microcontrollers are at the high-end of the PIC microcontroller family.
These microcontrollers cannot be used with the PicBasic compiler, but the
PicBasic Pro can be used to program them. Most of the devices in this group can
operate at up to 40 MHz, have 33 I/O pins, and 3 timers. They have 23 instructions
in addition to the 35 instructions found on the 14-bit microcontrollers.

Microcontroller Program

Memory:

Data

RAM

Max speed

(MHz)

I/O

Ports

A/D

Converter

17C43 4096 x16 454 33 33 –
17C752 8192 x16 678 33 50 12
18C242 8192 x16 512 40 23 5
18C252 16384x16 1536 40 23 5
18C452 16384x16 1536 40 34 8

All memory for the PIC microcontroller family is internal and it is usually not very easy to extend this memory externally. No special hardware or software features are provided for extending either the program memory or the data memory. The program memory is usually sufficient for small to medium size projects. But the data memory is generally small and may not be enough for medium to large projects unless a bigger and more expensive member of the family is chosen. For some large projects even this may not be enough and the designer may have to sacrifice the I/O ports to interface an external data memory, or to choose a microcontroller from a different manufacturer.

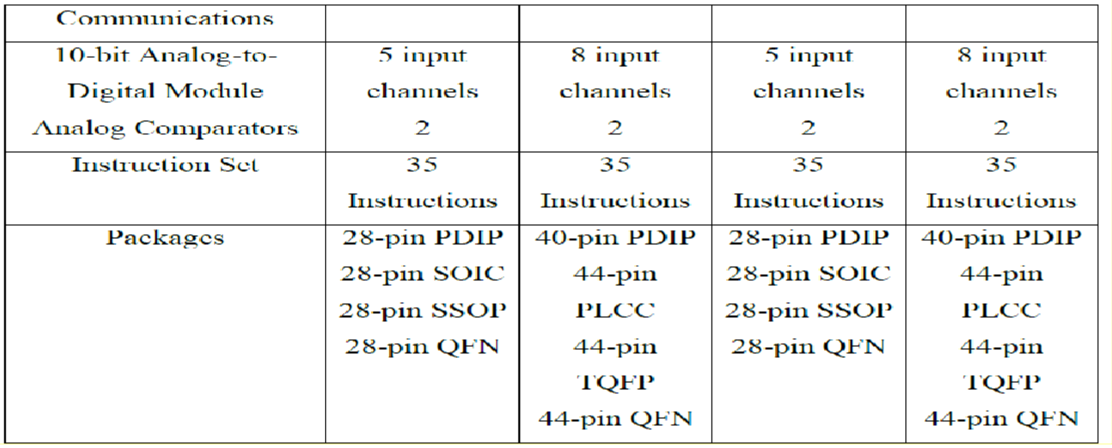
PIC16F877A:

PIC16F877A 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
- The 28-pin devices have three I/O ports, while the 40/44-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.

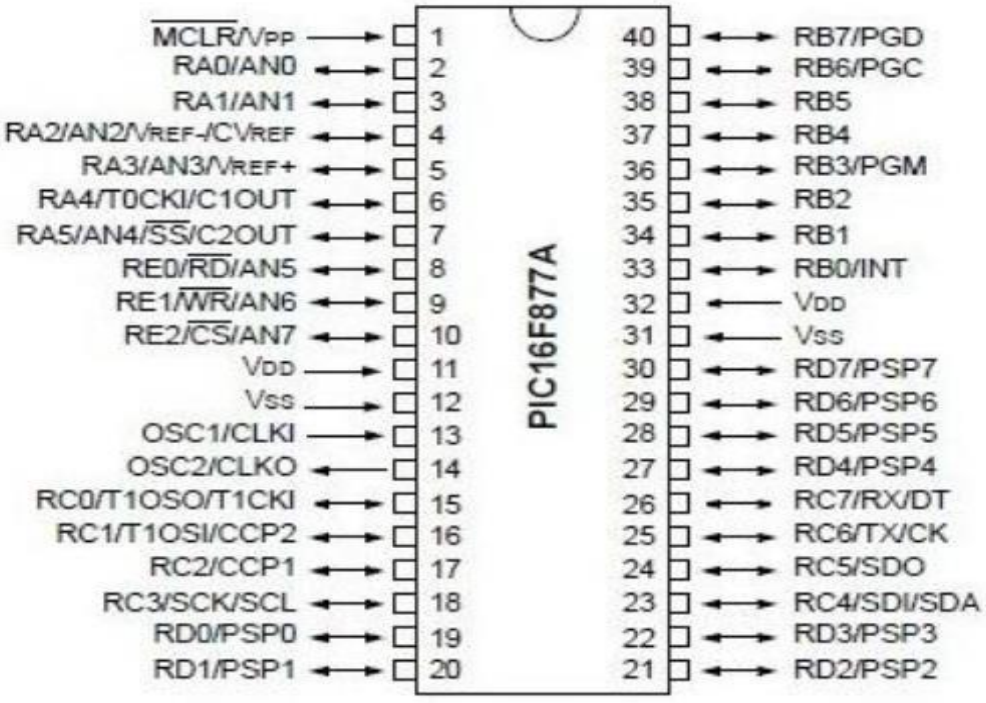
The available features are summarized

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	8K
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	—	PSP	—	PSP



Communications				
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

PIN DIAGRAM



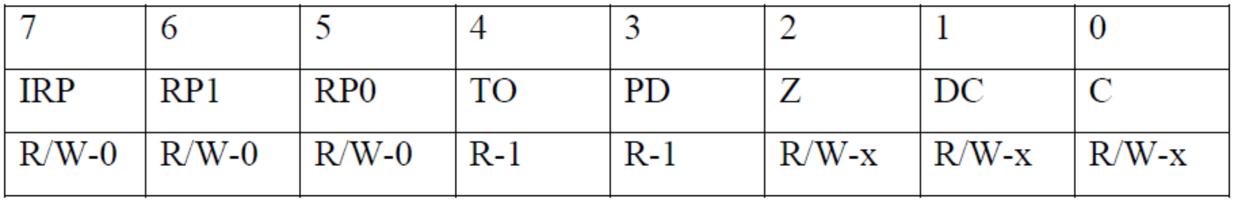
Register file map and special function registers

Register File Map (RFM) is a layout of all the registers available in a microcontroller and this is extremely useful when programming the device, especially when using an assembler language.

The RFM is divided into two parts: the *Special Function Registers* (SFR), and the *General Purpose Registers* (GPR). SFR is a collection of registers used by the microcontroller to control the internal operations of the device. Depending upon the complexity of the devices the number of registers in the SFR varies. Depending on the model of PIC microcontroller used there could be other registers. You need not know the operation of some of the registers since PicBasic and PicBasic Pro compiler loads these registers automatically. For example, writing and reading from the EEPROM are controlled by SFR registers EECON1, EECON2, EEADR, and EEDATA. But fortunately, PicBasic and PicBasic Pro compilers provide simple high-level instructions for writing to and reading from the EEPROM and thus you do not need to know how to load these registers.

Status Register (ADDRESS 03h, 83h, 103h, 183h)

The Status register contains the arithmetic status of the ALU, the Reset status and the bank select bits for data memory. The Status register can be the destination for any instruction, as with any other register. If the Status register is the destination for an instruction that affects the Z, DC or C bits, then the write to these three bits is disabled. These bits are set or cleared according to the device logic. Furthermore, the TO and PD bits are not writable, therefore, the result of an instruction with the Status register as destination may be different than intended.



7	6	5	4	3	2	1	0
IRP	RP1	RP0	TO	PD	Z	DC	C
R/W-0	R/W-0	R/W-0	R-1	R-1	R/W-x	R/W-x	R/W-x

Legend:

R = Readable bit W = Writable bit U = Unimplemented bit, read as _0‘ - n =

Value at POR _1‘ = Bit is set _0‘ = Bit is cleared x = Bit is unknown

bit 7 **IRP:** Register Bank Select bit (used for indirect addressing)

1 = Bank 2, 3 (100h-1FFh)

0 = Bank 0, 1 (00h-FFh)

bit 6-5 **RP1:RP0**: Register Bank Select bits (used for direct addressing)

11 = Bank 3 (180h-1FFh)

10 = Bank 2 (100h-17Fh)

01 = Bank 1 (80h-FFh)

00 = Bank 0 (00h-7Fh)

Each bank is 128 bytes.

bit 4 **TO**: Time-out bit

1 = After power-up, CLRWDT instruction or SLEEP instruction

0 = A WDT time-out occurred

bit 3 **PD**: Power-down bit

1 = After power-up or by the CLRWDT instruction

0 = By execution of the SLEEP instruction

bit 2 **Z**: Zero bit

1 = The result of an arithmetic or logic operation is zero

0 = The result of an arithmetic or logic operation is not zero

bit 1 **DC**: Digit carry/borrow bit

(for borrow, the polarity is reversed)

1 = A carry-out from the 4th low order bit of the result occurred

0 = No carry-out from the 4th low order bit of the result

bit 0 **C**: Carry/borrow bit

1 = A carry-out from the Most Significant bit of the result occurred

0 = No carry-out from the Most Significant bit of the result occurred

OPTION register(ADDRESS 81h, 181h)

This register is used to setup various internal features of the microcontroller and is named as OPTION_REG. This is a readable and writable register which contains various control bits to configure the on-chip timer and the watchdog timer. This register is at address 81 (hexadecimal) of the microcontroller . The OPTION REG register is also used to control the external interrupt pin RB0. This pin can be setup to generate an interrupt, for example, when it changes from logic 0 to logic 1. The microcontroller then suspends the main program execution and jumps to the interrupt service routine (ISR) to service the interrupt. Upon return from the interrupt, normal processing resumes.

7	6	5	4	3	2	1	0
RBPU	INTEDG	T0CS	T0SE	PSA	PS2	PS1	PS0

OPTION_REG bit definitions:
Bit 7: PORTB Pull-up Enable
1: PORTB pull-ups disabled
0: PORTB pull-ups enabled
Bit 6: INT Interrupt Edge Detect
1: Interrupt on rising edge of INT input
0: Interrupt on falling edge of INT input
Bit 5: TMR0 Clock Source
1: T0CK1 pulse
0: Internal oscillator
Bit 4: TMR0 Source Edge Select
1: Increment on HIGH to LOW of T0CK1
0: Increment on LOW to HIGH of T0CK1
Bit 3: Prescaler Assignment
1: Prescaler assigned to Watchdog Timer
0: Prescaler assigned to TMR0
Bit 2-0: Prescaler Rate

I/O registers

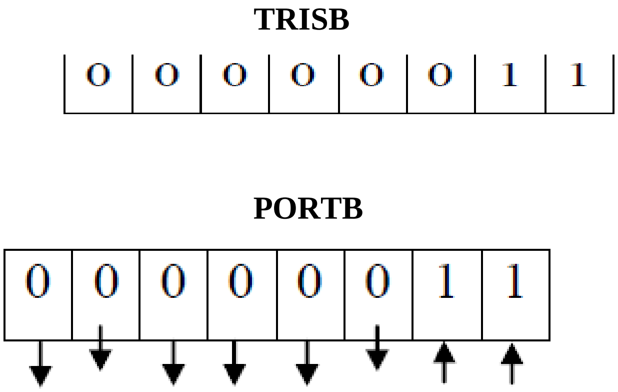
These registers are used for the I/O control. Every I/O port in the PIC microcontroller has two registers: *port data register* and *port direction control register*. Port data register has the same name as the port it controls. For example, PIC16F84 microcontroller has two port data registers PORTA and PORTB. A PIC16F877 microcontroller has 5 port data registers PORTA, PORTB, PORTC, PORTD, and PORTE.

TRIS Registers

An 8-bit data can be sent to any port, or an 8-bit data can be read from the ports. It is also possible to read or write to individual port pins. For example, any bit of a given port can be set or cleared, or data can be read from one or more port pins at the same time. Ports in a PIC microcontroller are bi-directional. Thus, each pin of a port can be used as an input or an output pin. Port direction control register configures the port pins as either inputs or outputs. This register is called the TRIS register and every port has a TRIS register named after its port name. For example, TRISA is the direction control register for PORTA. Similarly, TRISB is the direction control register for PORTB and so on. Setting bit in the TRIS register makes the corresponding port register pin an input. Clearing a bit in the TRIS register makes the corresponding port pin an output. For example, to make bits 0 and 1 of PORTB input and the other bits output, we have to load the TRISB register with the bit pattern.

00000011

Figure shows the TRISB register and the direction of PORTB pins.



TRISB and PORTB direction

PORT Registers

Data on an I/O pin is accessed via a PORTx register. A read of the PORTx register reads the value of the I/O pin, while a write to the PORTx register writes the value to the port data latch.

LAT Registers

The LATx register associated with an I/O pin eliminates the problems that could occur with read-modify-write instructions. A read of the LATx register returns the values held in the port output latches, instead of the values on the I/O pins. A read modify- write operation on the LAT register, associated with an I/O port, avoids the possibility of writing the input pin values into the port latches. A write to the LATx register has the same effect as a write to the PORTx register.

The differences between the PORT and LAT registers can be summarized as follows:

- A write to the PORTx register writes the data value to the port latch.
- A write to the LATx register writes the data value to the port latch.
- A read of the PORTx register reads the data value on the I/O pin.
- A read of the LATx register reads the data value held in the port latch.

CAPTURE/COMPARE/PWM MODULES

Each Capture/Compare/PWM (CCP) module contains a 16-bit register which can operate as a:

- 16-bit Capture register
- 16-bit Compare register
- PWM Master/Slave Duty Cycle register

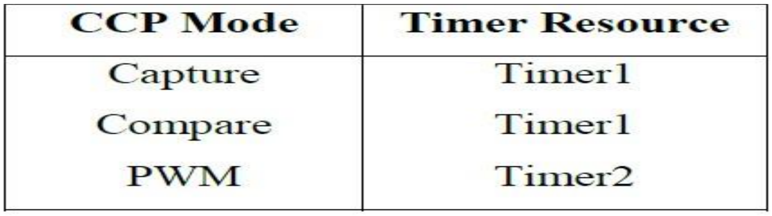
Both the CCP1 and CCP2 modules are identical in operation, with the exception being the operation of the special event trigger. Tables show the resources and interactions of the CCP module(s). In the following sections, the operation of a CCP module is described with respect to CCP1. CCP2 operates the same as CCP1 except where noted.

CCP1 Module:

Capture/Compare/PWM Register 1 (CCPR1) is comprised of **two 8-bit registers**: **CCPR1L (low byte)** and **CCPR1H (high byte)**.The CCP1CON register controls the operation of CCP1. The special event trigger is generated by a compare match and will reset Timer1.

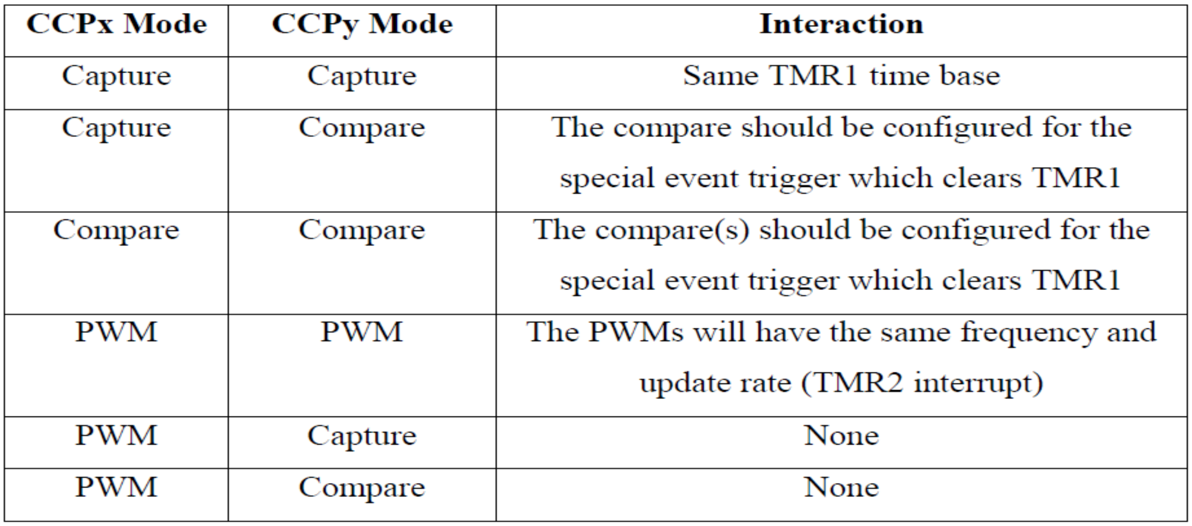
CCP2 Module:

Capture/Compare/PWM Register 2 (CCPR2) is comprised of **two 8-bit registers**: **CCPR2L (low byte)** and **CCPR2H (high byte)**. The CCP2CON register controls the operation of CCP2. The special event trigger is generated by a compare match and will reset Timer1 and start an A/D conversion (if the A/D module is enabled).



CCP Mode	Timer Resource
Capture	Timer1
Compare	Timer1
PWM	Timer2

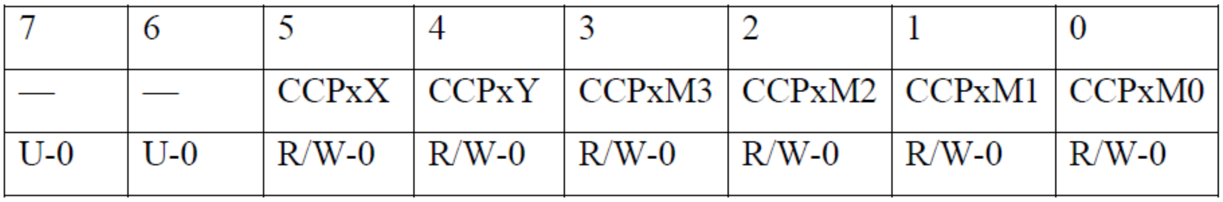
CCP MODE – TIMER RESOURCES REQUIRED



CCPx Mode	CCPy Mode	Interaction
Capture	Capture	Same TMR1 time base
Capture	Compare	The compare should be configured for the special event trigger which clears TMR1
Compare	Compare	The compare(s) should be configured for the special event trigger which clears TMR1
PWM	PWM	The PWMs will have the same frequency and update rate (TMR2 interrupt)
PWM	Capture	None
PWM	Compare	None

INTERACTION OF TWO CCP MODULES

CCP1CON REGISTER/CCP2CON REGISTER



7	6	5	4	3	2	1	0
—	—	CCPxX	CCPxY	CCPxM3	CCPxM2	CCPxM1	CCPxM0
U-0	U-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

bit 7-6 **Unimplemented:** Read as _0‘

bit 5-4 **CCPxX:CCPxY**: PWM Least Significant bits

Capture mode:

Unused.

Compare mode:

Unused.

PWM mode:

These bits are the two LSbs of the PWM duty cycle. The eight MSbs are found in CCPRxL.

bit 3-0 **CCPxM3:CCPxM0**: CCPx Mode Select bits

0000 = Capture/Compare/PWM disabled (resets CCPx module)

0100 = Capture mode, every falling edge

0101 = Capture mode, every rising edge

0110 = Capture mode, every 4th rising edge

0111 = Capture mode, every 16th rising edge

1000 = Compare mode, set output on match (CCPxIF bit is set)

1001 = Compare mode, clear output on match (CCPxIF bit is set)

1010 = Compare mode, generate software interrupt on match (CCPxIF bit is set, CCPx pin is unaffected)

1011 = Compare mode, trigger special event (CCPxIF bit is set, CCPx pin is unaffected); CCP1 resets TMR1; CCP2 resets TMR1 and starts an A/D conversion (if A/D module is enabled)

11xx = PWM mode

ANALOG-TO-DIGITAL CONVERTER (A/D) MODULE

The Analog-to-Digital (A/D) Converter module has five inputs for the 28-pin devices and eight for the 40/44-pin devices. The conversion of an analog input signal results in a corresponding 10-bit digital number. The A/D module has high and low-voltage reference input that is software selectable to some combination of VDD, VSS, RA2 or RA3.The A/D converter has a unique feature of being able to operate while the device is in Sleep mode. To operate in Sleep, the A/D clock must be derived from the A/D‘s internal RC oscillator. The A/D module has four registers. These registers are:

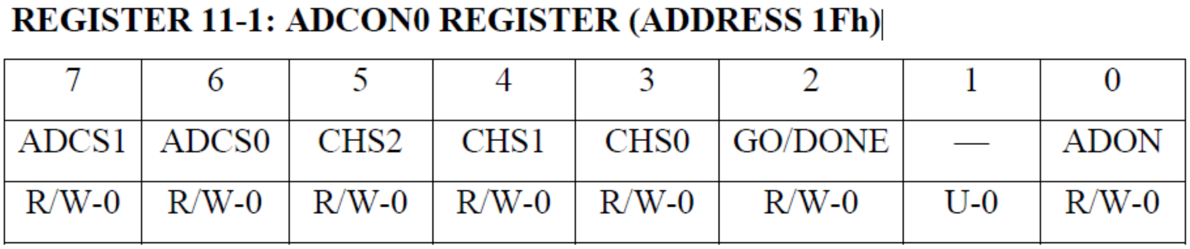
• A/D Result High Register (ADRESH)

• A/D Result Low Register (ADRESL)

• A/D Control Register 0 (ADCON0)

• A/D Control Register 1 (ADCON1)

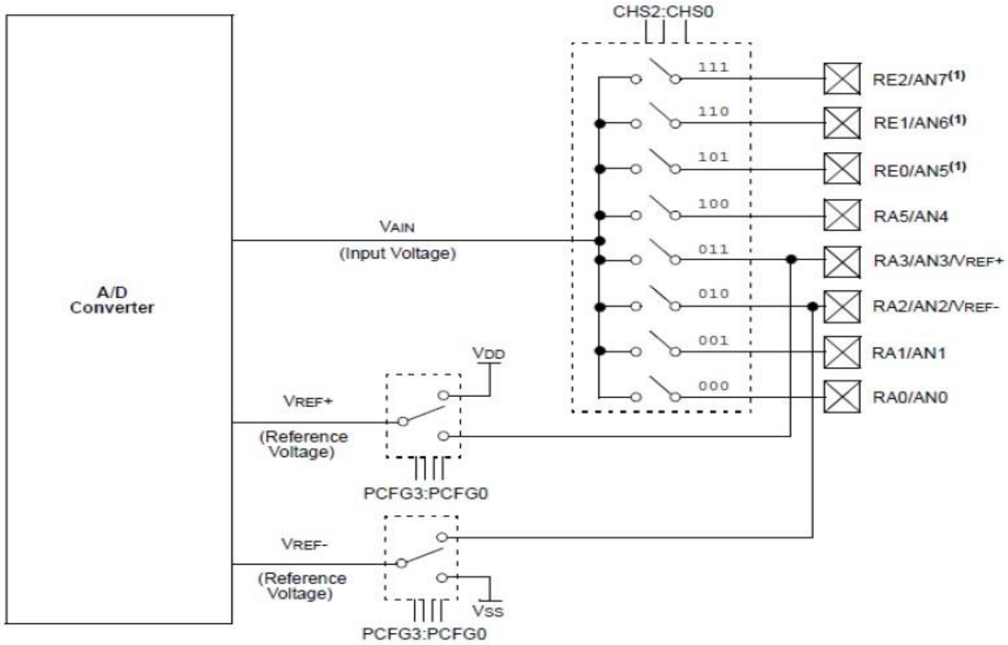
The ADCON0 register, shown in Register 11-1, controls the operation of the A/D module. The ADCON1 register, shown in Register 11-2, configures the functions of the port pins. The port pins can be configured as analog inputs (RA3 can also be the voltage reference) or as digital I/O



REGISTER 11-1: ADCON0 REGISTER (ADDRESS 1Fh)

7	6	5	4	3	2	1	0
ADCS1	ADCS0	CHS2	CHS1	CHS0	GO/DONE	—	ADON
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	U-0	R/W-0

A/D BLOCK DIAGRAM



To do an A/D Conversion, follow these steps:

1. Configure the A/D module:

- Configure analog pins/voltage reference and digital I/O (ADCON1)
- Select A/D input channel (ADCON0)
- Select A/D conversion clock (ADCON0)
- Turn on A/D module (ADCON0)

2. Configure A/D interrupt (if desired):

- Clear ADIF bit
- Set ADIE bit

- Set PEIE bit
- Set GIE bit

3. Wait the required acquisition time.

4. Start conversion:

- Set GO/DONE bit (ADCON0)

5. Wait for A/D conversion to complete by either:

- Polling for the GO/DONE bit to be cleared (interrupts disabled); OR
- Waiting for the A/D interrupt

6. Read A/D Result register pair
(ADRESH:ADRESL), clear bit ADIF if required.

7. For the next conversion, go to step 1 or step 2 as required. The A/D conversion
time per bit is defined as TAD.

Selecting the A/D Conversion Clock :

The A/D conversion time per bit is defined as TAD. The A/D conversion requires a minimum 12 TAD per 10-bit conversion. The source of the A/D conversion clock is software selected. The seven possible options for TAD are:

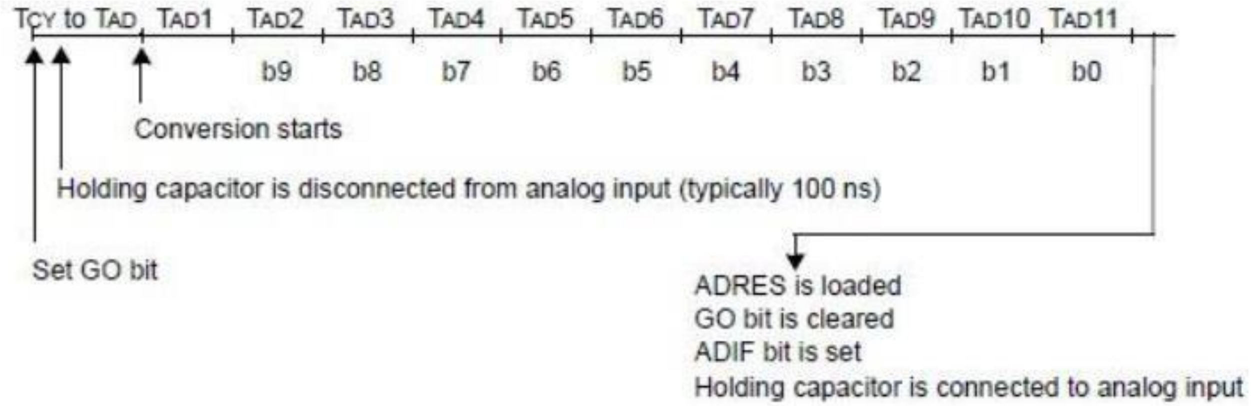
- 2 TOSC
- 4 TOSC
- 8 TOSC
- 16 TOSC
- 32 TOSC
- 64 TOSC
- Internal A/D module RC oscillator (2-6 μs)

For correct A/D conversions, the A/D conversion clock (TAD) must be selected to ensure a minimum TAD time of 1.6 μs.

Table shows the resultant TAD times derived from the device operating frequencies and the A/D clock source selected.

A/D Conversions:

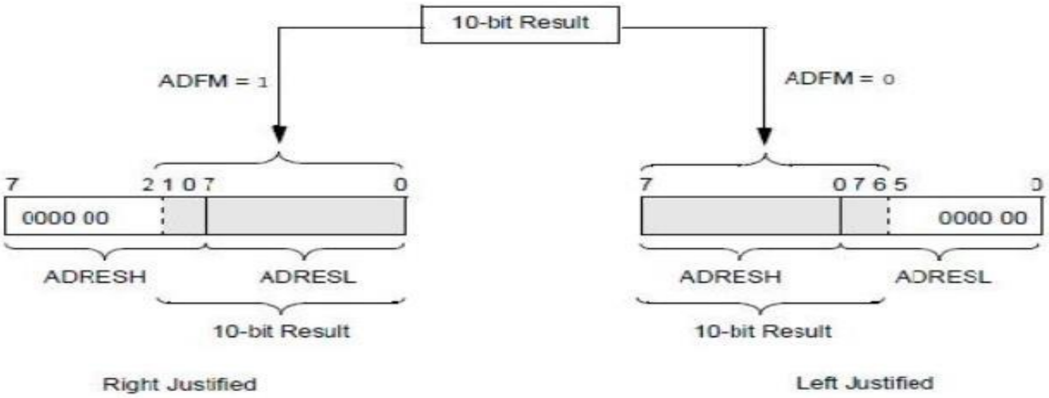
Clearing the GO/DONE bit during a conversion will abort the current conversion. The A/D Result register pair will NOT be updated with the partially completed A/D conversion sample. That is, the ADRESH: ADRESL registers will continue to contain the value of the last completed conversion (or the last value written to the ADRESH: ADRESL registers). After the A/D conversion is aborted, the next acquisition on the selected channel is automatically started. The GO/DONE bit can then be set to start the conversion. In Figure after the GO bit is set, the first time segment has a minimum of TCY and a maximum of TAD.



A/D conversion TAD cycles

A/D RESULT REGISTERS:

The ADRESH: ADRESL register pair is the location where the 10-bit A/D result is loaded at the completion of the A/D conversion. This register pair is 16 bits wide. The A/D module gives the flexibility to left or right justify the 10-bit result in the 16-bit result register. The A/D Format Select bit (ADFM) controls this justification. Figure shows the operation of the A/D result justification. The extra bits are loaded with _0‘s. When an A/D result will not overwrite these locations (A/D disable), these registers may be used as two general purpose 8-bit registers.



A/D Operation during Sleep

The A/D module can operate during Sleep mode. This requires that the A/D clock source be set to RC (ADCS1:ADCS0 = 11). When the RC clock source is selected, the A/D module waits one instruction cycle before starting the conversion. This allows the SLEEP instruction to be executed which eliminates all digital switching noise from the conversion. When the conversion is completed, the GO/DONE bit will be cleared and the result loaded into the ADRES register. If the A/D interrupt is enabled, the device will wake-up from Sleep. If the A/D interrupt is not enabled, the A/D module will then be turned off, although the ADON bit will remain set. When the A/D clock source is another clock option (not RC), a SLEEP instruction will cause the present conversion to be aborted and the A/D module to be turned off, though the ADON bit will remain set. Turning off the A/D places the A/D module in its lowest current consumption state.

Effects of a Reset

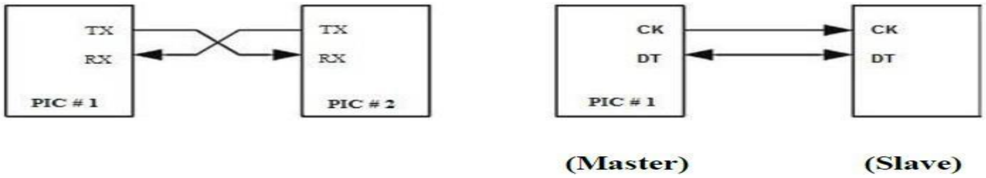
A device reset forces all registers to their Reset state. This forces the A/D module to be turned off and any conversion is aborted. All A/D input pins are configured as analog inputs. The value that is in the ADRESH: ADRESL registers is not modified for a Power-on Reset. The ADRESH: ADRESL registers will contain unknown data after a Power-on Reset.

The watchdog timer

A *watchdog timer* is an internal timer running independently of the system clock. It resets the device in the event of a program or circuit malfunction or if an unknown logical state is encountered. For example, if the program hangs, the watchdog timer will time out and reset the processor. The PIC16F84 has a watchdog timer with a timeout period of approximately 18msec, with no prescaler, determined by a separate internal *RC* oscillator.

USART Serial Port

Medium-end PIC microcontrollers have a serial communications port called the universal synchronous asynchronous receiver transmitter (USART) or serial communication interface (SCI). This port can be configured to establish a simultaneous asynchronous bidirectional communication (full duplex) or nonsimultaneous synchronous (transmitting the clock signal) bidirectional communication (half duplex).



Pins TX/CK and RX/DT used by the USART serial port. (a) Connection between two PICs in asynchronous mode. TX and RX are data pins that transmit and receive, respectively. The clock signal is not transmitted. (b) Synchronous mode connection. CK is the clock pin: output in the master device and input in the slave
device. DT is the data pin: output in the transmitter and input in the receiver.

General Description

The USART serial port uses the TX/CK and RX/DT pins from the microcontroller, which normally share functions with two pins in the parallel port C. In asynchronous mode, TX/CK is the pin that the transmitter uses to transmit data and RX/DT is the receiver pin for data input. In synchronous mode, TX/CK is the pins to output the clock signal if the device has been configured as a master or the pin or the input clock signal if it has been configured as a slave. RX/DT is the bidirectional pin for data signal. Figure above shows the use of these terminals between two PIC microcontrollers. The USART serial port uses two special function registers, TXREG and RCREG, which are used to store the data to be transmitted or the data received. The USART serial port also uses the registers TXSTA and RCSTA to control the port, and the register SPBRG to establish the communication speed.

PIC CODE :-

```
 #include<lcd8out.h>

 char *str11=" Temprature  ";
 char *str12 = " Control System";
 char *str14 =   "Normal          ";
 char *str13 =   "Critical        " ;

#define buzz PORTC.F7

#define s1 PORTB.f0
#define s2 PORTB.f1
#define s3 PORTB.f2

 void main()
 {
   unsigned int x;
   unsigned char maxval=0;

   TRISB=0x07;
   TRISD=0x00;
   TRISC=0x00;
   TRISA = 0xff; // PORTA is input// Configure analog inputs and Vref
   ADCON1=0x80;

   Lcd8_Config(&PORTC,&PORTD,0,2,1,7,6,5,4,3,2,1,0); // Initialize LCD on PORTD
   Lcd8_Cmd(0x01);
   Lcd8_Cmd(0x0c); // Turn cursor off
   Lcd8_Cmd(0x80);
   Lcd8_Out_Cp(str11);
```

```
Lcd8_Cmd(0xc0);
Lcd8_Out_Cp(str12);
Delay_ms(3000);

s: lcd_cmd1(0x01);
buzz=1;
maxval=0 ;
lcd_cmd1(0x80);
lcd_puts("Choose Max. Temp");
while(s3==1)
{
 if(s1==0)
 {
   maxval++;
   if(maxval==100)
   maxval=0;
 }

 if(s2==0)
 {
  maxval--;
  if(maxval==255)
  maxval=100;
 }

 lcd_cmd1(0xc0);
 displaypval(maxval);
 Delay_ms(200);
}

while(s3==0);
```

```
   lcd_cmd1(0x01)      ;
  while(1)
  {
   x= Adc_Read(0); // Get results of AD conversion
     lcd_cmd1(0xc0);
   lcd_puts("TEMP:") ;
   displaypval(x);
   delay_ms(100);

   if (x>maxval)
   {
    lcd_cmd1(0x80);
    lcd_puts(str13);
    buzz=0;
   }

   else
   {
    lcd_cmd1(0x80);
    lcd_puts(str14);
    buzz=1;
   }

   if(s1==0)
   {
    while(s1==0);
    goto s;
   }
     delay_ms(3000);
}
}
```

Simple Switch Pad

A micro switch is a switch which can be operated using a very small force and also possibly using a small movement. This is a two terminal switch. Both terminals get shorted when it is pressed (i.e. in closed state) and both the terminals get disconnected when it is released (i.e. in open state). Any input signal is applied at one of its terminals and when it is pressed (i.e. in closed position) the same input signal is received at the other terminal.

The switches can be used in two ways:

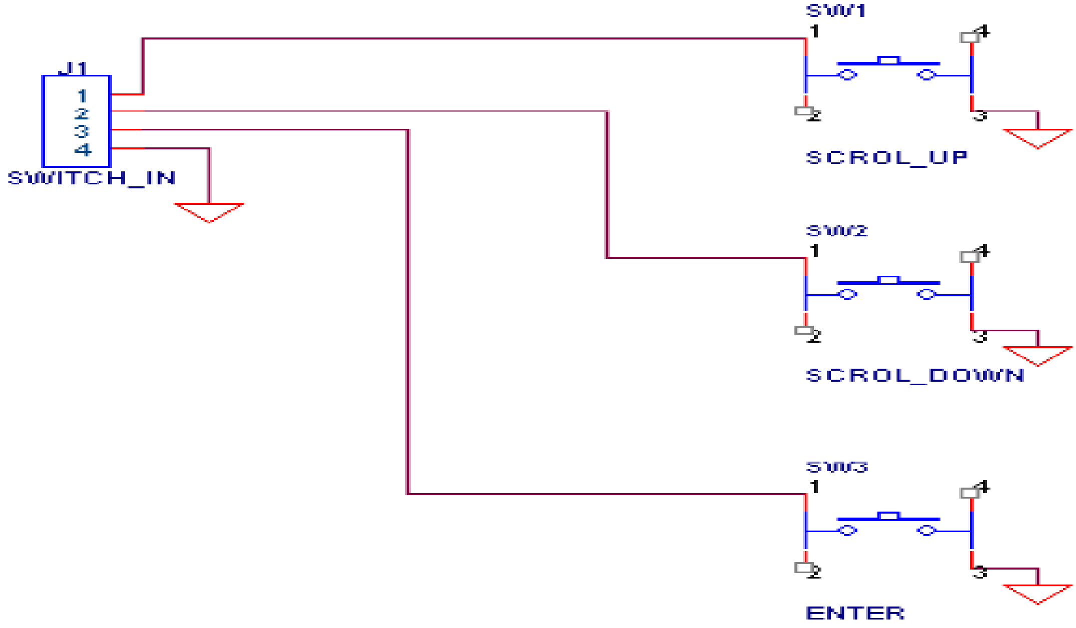
(a) Switch in Press Position when the MCU performs its task immediately after pressing the switch.

(b) Switch in release Position when MCU does not perform any task but waits for the release of switch and performs the task after releasing the switch.

In our switch pad, there are three micro switches connected to the microcontroller I/O pins. One end of each switch is grounded and other is connected to the microcontroller port, as in Figure. When any switch is pressed that particular port is grounded. The microcontroller always monitors these switches in real time (i.e. in continuous mode). The configuration of the micro switches is as follows:

- Switch1 (S1): Increment
- Switch2 (S2): Decrement
- Switch3 (S3): Set

Interfacing of Switches with µC



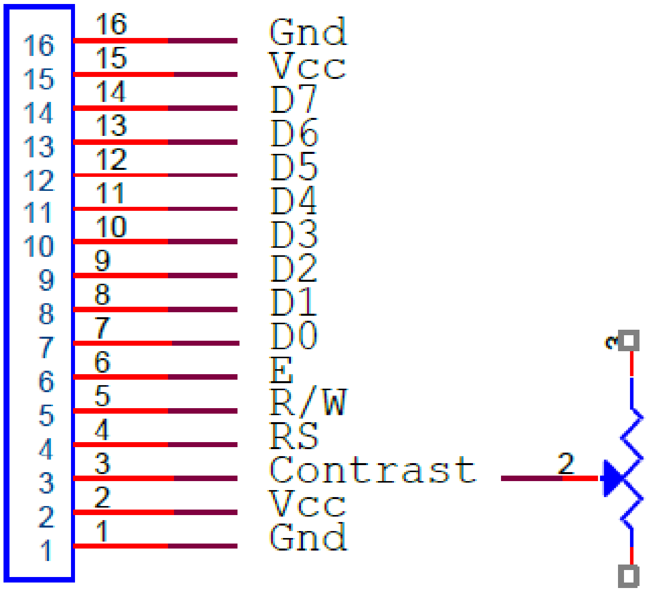
Display Unit

Liquid Crystal Display

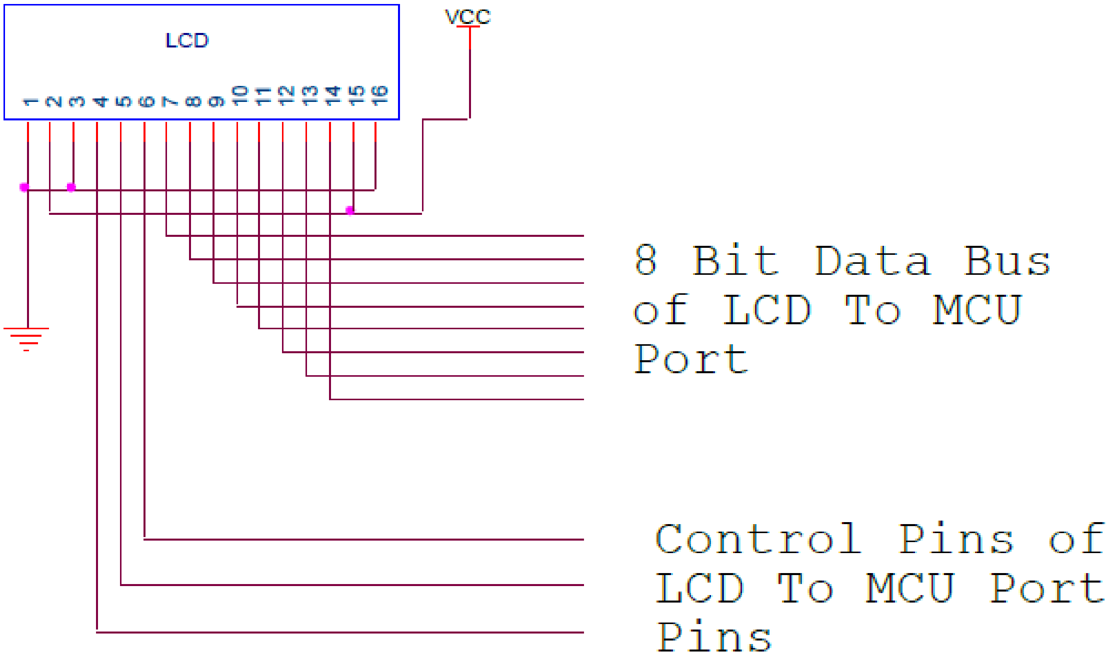
Liquid crystal displays (LCD) are widely used in recent years as compares to LEDs. This is due to the declining prices of LCD, the ability to display numbers, characters and graphics, incorporation of a refreshing controller into the LCD, their by relieving the CPU of the task of refreshing the LCD and also the ease of programming for characters and graphics. HD 44780 based LCDs are most commonly used. The LCD, which is used as a display in the system, is LMB162A. The main features of this LCD are: 16 X 2 display, intelligent LCD, used for alpha numeric characters & based on ASCII codes. This LCD contains 16 pins, in which 8 pins are used as 8-bit data I/O, which are extended ASCII. Three pins are used as control lines these are Read/Write pin, Enable pin and Register select pin. Two pins are used for Backlight and LCD voltage, another two pins are for Backlight &
LCD ground and one pin is used for contrast change.

LCD pin description

Pin	Symbol	I/O	Description
1	VSS	-	Ground
2	VCC	-	+5V power supply
3	VEE	-	Power supply to control contrast
4	RS	I	RS=0 to select command register, RS=1 to select data register.
5	R/W	I	R/W=0 for write, R/W=1 for read
6	E	I/O	Enable
7	DB0	I/O	The 8 bit data bus
8	DB1	I/O	The 8 bit data bus
9	DB2	I/O	The 8 bit data bus
10	DB3	I/O	The 8 bit data bus
11	DB4	I/O	The 8 bit data bus
12	DB5	I/O	The 8 bit data bus
13	DB6	I/O	The 8 bit data bus
14	DB7	I/O	The 8 bit data bus



LCD Pin Description Diagram

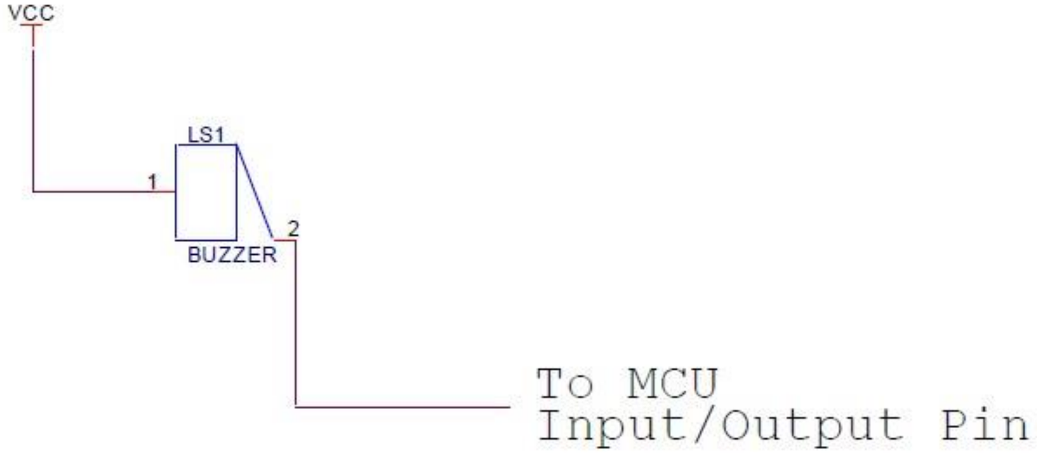


Interfacing of Microcontroller with LCD

Buzzer for Beep Source

A **buzzer** or **beeper** is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric. Typical uses of buzzers and beepers include alarm devices, timers and confirmation of user input such as a mouse click or keystroke.

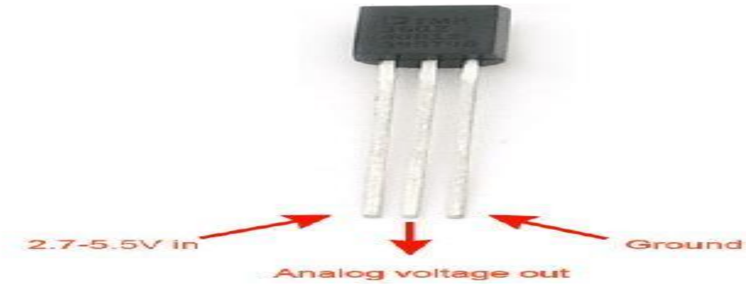
Interfacing Circuit to MCU



Temperature Sensor (LM35)

Most commonly-used electrical temperature sensors are difficult to apply. For example, thermocouples have low output levels and require cold junction compensation. Thermistor are nonlinear. In addition, the outputs of these sensors are not linearly proportional to any temperature scale. Early monolithic sensors, such as the LM3911, LM134 and LM135, overcame many of these difficulties, but their outputs are related to the Kelvin temperature scale rather than the more popular Celsius and Fahrenheit scales. Fortunately, in 1983 two I.C.'s, the LM34 Precision Fahrenheit Temperature Sensor and the LM35 Precision Celsius Temperature Sensor, were introduced. The LM35 is an integrated circuit sensor that can be used to measure temperature with an electrical output proportional to the temperature (in **o**C). The LM35 sensor thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling.

The LM35 series are precision integrated-circuit LM35 temperature sensors, whose output voltage is linearly proportional to the Celsius (Centigrade) temperature. The LM35 sensor thus has an advantage over linear temperature sensors calibrated in ° Kelvin, as the user is not required to subtract a large constant voltage from its output to obtain convenient Centigrade scaling. The LM35 sensor does not require any external calibration or trimming to provide typical accuracies of ±¼°C at room temperature and ±¾°C over a full -55 to +150°C temperature range. Low cost is assured by trimming and calibration at the wafer level. The LM35's low output impedance, linear output, and precise inherent calibration make interfacing to readout or control circuitry especially easy. It can be used with single power supplies, or with plus and minus supplies. As it draws only 60 µA from its supply, it has very low self-heating, less than 0.1°C in still air. The LM35 is rated to operate over a -55° to +150°C temperature range, while the LM35C sensor is rated for a -40° to +110°C range (-10° with improved accuracy). The LM35 series is available packaged in hermetic TO-46 transistor packages, while the LM35C, LM35CA, and LM35D are also available in the plastic TO-92 transistor package. The LM35D sensor is also available in an 8-lead surface mount small outline



Specifications

- Calibrated directly in ° Celsius (Centigrade)
- Linear + 10.0 mV/°C scale factor
- 0.5°C accuracy guarantee able (at +25°C)
- Rated for full −55° to +150°C range
- Suitable for remote applications
- Operates from 4 to 30 volts
- Less than 60 µA current drain
- Nonlinearity only ±0.25°C typical
- Low impedance output, 0.1 W for 1 mA load

There are several versions for operation over different temperature ranges:

LM35 -55°C to 150°C
LM35C -40°C to 110°C
LM35D 0°C to 100°C

Why Use LM35s to Measure Temperature?

- You can measure temperature more accurately than a using a thermistor.
- The sensor circuitry is sealed and not subject to oxidation, etc.
- The LM35 generates a higher output voltage than thermocouples and may not require that the output voltage be amplified.

How to Measure Temperature

- Using the TMP36 is easy, simply connect the left pin to power (2.7-5.5V) and the right pin to ground. Then the middle pin will have an analog voltage that is directly proportional (linear) to the temperature. The analog voltage is independant of the power supply.

To convert the voltage to temperature, simply use the basic formula:

Temp in °C = [(Vout in mV) - 500] / 10

So for example, if the voltage out is 1V that means that the temperature is ((1000 mV - 500) / 10) = 50 °C

REFERENCES :

- Electronic Devices and Circuits by Jacob Millman, Christos C. Halkias
- Electronic Devices and Circuits by Boylsted and Nashelsky PHI, 6e, 2001
- www.robot-electronics.com