Practical: 14

Aim: To learn IO interfacing with controllers, and implement LED and switch interfacing with 8-bit microcontrollers.

Input Interface Devices: Push button switch, Keypad, Infrared sensor, Temperature sensor, gas Sensor etc. These devices provide some information to the Microcontroller, and this is called input data.

Output Interface Devices: LED, LCD, Buzzer, Relay driver, DC Motor Driver, 7-Segment Display etc.

Storage Interface Devices : Used to store/ retain the data, example, SD card, EEPROM, DataFlash, Real Time Clock , etc.

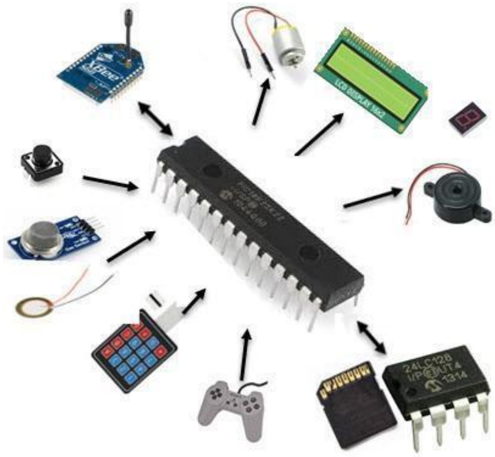

figure 14.1

Interfacing of an LED

Interfacing comprises hardware (Interface device) and Software (source code to communicate, also called as the Driver). Simply, to use an LED as the output device, the LED should be connected to the Microcontroller port and the MC has to be programmed inside to make the LED ON or OFF or blink or dim. This program is called the driver/firmware. The driver software can be developed using any programming language like Assembly , C etc.

Microcontroller

The Microcontroller was invented in the 1980's by Intel. Its foundation is based on Harvard architecture and this Microcontroller was developed principally for bringing it to be used in Embedded Systems. We have discussed Microcontroller History and Basics. It is a 40 Pin PDIP (Plastic Dual Inline Package).

Microcontroller has an on-chip oscillator, but it requires an external clock to run it. A quartz crystal is connected in between the XTAL pins of the MC. This crystal needs two same value capacitors (33pF) for generating a clock signal of the desired frequency. Features of Microcontroller have been explained in our previous article.

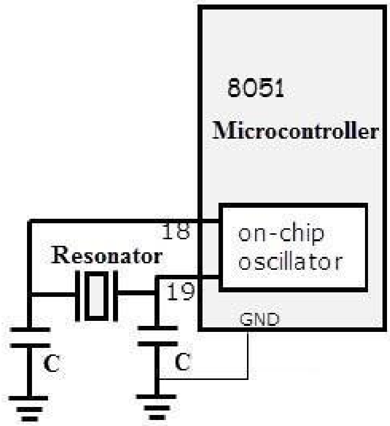

figure 14.2

LED (Light Emitting Diode)

LED is a semiconductor device used in many electronic devices, mostly used for signal transmission /power indication purposes. It is very cheaply and easily available in a variety of shapes, color, and size. The LEDs are also used for design message display boards and traffic control signal lights etc.

It has two terminals positive and negative as shown in the figure.

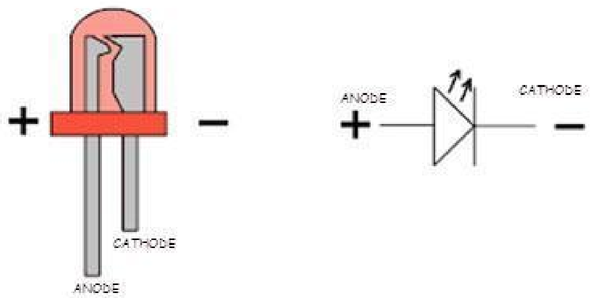

figure 14.3

The only way to know polarity is either to test it with a multimeter or by carefully observing inside the LED. The larger end inside the led is -ve (cathode) and the shorter one is +ve (anode), that is how we find out the polarity of the LED. Another way to recognize the polarity is, connecting leads, POSITIVE terminal has more length than NEGATIVE terminal.

LED Interfacing

There are two ways in which we can interface LEDs to the Microcontroller . But the connections and programming techniques will be different. This article provides information on LED interfacing with an LED blinking code for AT89C52/ AT89C51 Microcontroller.

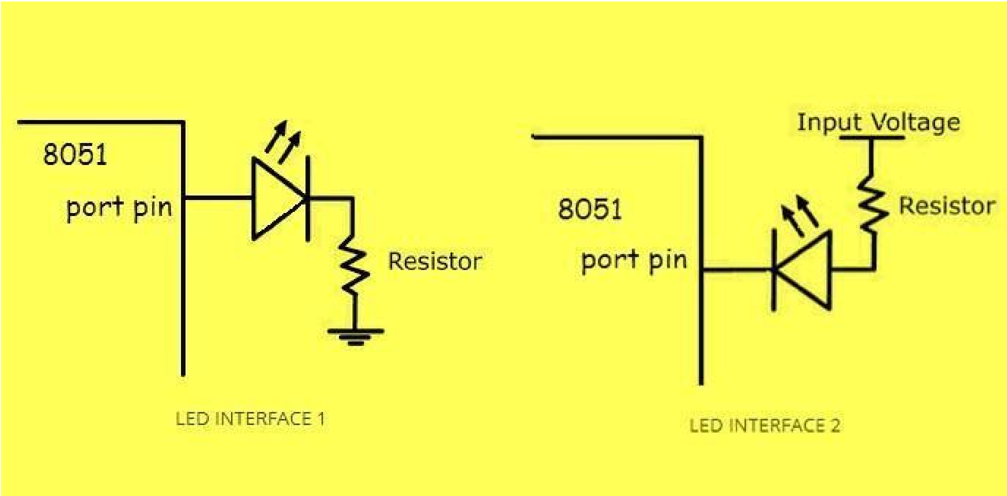

figure 14.4

Observe carefully the interface LED 2 is forward biased because the input voltage of 5v connected to the positive terminal of the LED, So here the Microcontroller pin should be at LOW level. And vice versa with the interface 1 connections.

The resistor is important in LED interfacing to limit the flowing current and avoid damaging the LED and/or MCU.

- Interface 1 will glow LED, only if the PIN value of the MC is HIGH as current flows towards the ground.
- Interface 2 will glow LED, only if the PIN value of the MC is LOW as current flows towards PIN due to its lower potential.

The circuit diagram is shown below. An LED is connected to the pin-0 of port-1.

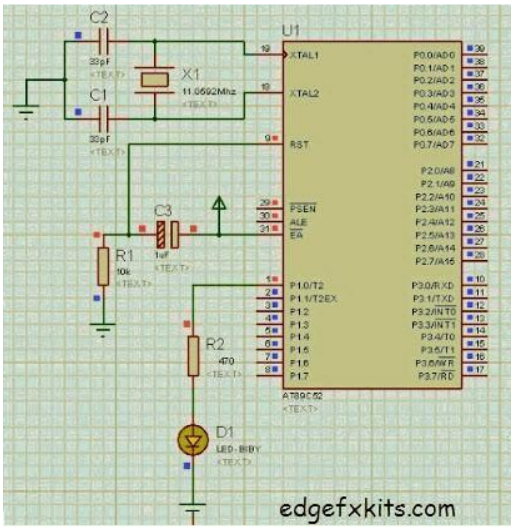

figure 14.5

A crystal of 11.0592 MHz is connected to generate the clock. As we know that Microcontroller executes an instruction in 12 CPU cycles [1], hence this 11.0592Mhz crystal makes this run at 0.92 MIPS (Million of instructions per second).

In the code below, the LED is defined as the pin 0 of the port 1. In the main function, the LED is toggled after every half second. The 'delay' function executes null statements every time when it executes.

A value of 60000 (compiled using Keil microvision 4 software) generates about 1 seconds (delay time) null statement execution time when 11.0592 MHz crystal is being used. In this way, the LED attached to the P1.0 pin is made to blink using the code given below.

CODE

```
#include<reg51.h>  sbit LED= P1^0; // pin0 of port1 is named as LED
//Function declarations
void cct_init(void);  void delay(int a);  int main(void)
{
cct_init();  while(1)
{
LED=0;  delay(60000);  LED=1;  delay(60000);
}
}
void cct_init(void)
{
P0= 0x00;
P1= 0x00;
P2= 0x00;
P3= 0x00;
}
void delay (int a)
{  int i;
for( i=0; i<a; i++) // Null statement execution i.e, delay time
}
```