LABORATORY 4

INTRODUCTION TO 8051 MICROCONTROLLER SIMULATION KITS AND EXPERIMENT THE LED BLINKING PROGRAM

Objectives:

- To get familiar with 8051 Microcontroller and its simulation tools including Keil C51 Evaluation Kit and Proteus Kit.
- To simulate a simple example program- LED Blinking with 8051 Microcontroller.
- To understand the timer system (crystal oscillator circuit) and the relationship between clock frequencies and microprocessor speed.

Introduction: Microcontroller (MC) is called computer on chip since it includes microprocessor with internal ROM, RAM, parallel and serial ports within single chip. MC is broadly used in washing machines, vcd player, microwave oven, robotics and etc. 8051 is an 8 bit microcontroller, means 8 bit data bus, means able to read, write and process 8 bit data. Architecture of 8051 is presented in figure 1. 8051 executes code from an embedded masked ROM.

Intel's original MCS-51 family was developed using N-type metal-oxide-semiconductor (NMOS) technology like its predecessor Intel MCS-48, but later versions, identified by a letter C in their name (e.g., 89C51) used complementary metal–oxide–semiconductor (CMOS) technology and consume less power than their NMOS predecessors. This made them more suitable for battery-powered devices. In this particular experiment you are going to use **AT89C52, which** is an 8-bit microcontroller and belongs to Atmel's 8051 family. AT89C52 has 8KB of Flash programmable and erasable read only memory (PEROM) and 256 bytes of RAM. AT89C52 has an endurance of 1000 Write/Erase cycles which means that it can be erased and programmed to a maximum of 1000 times.

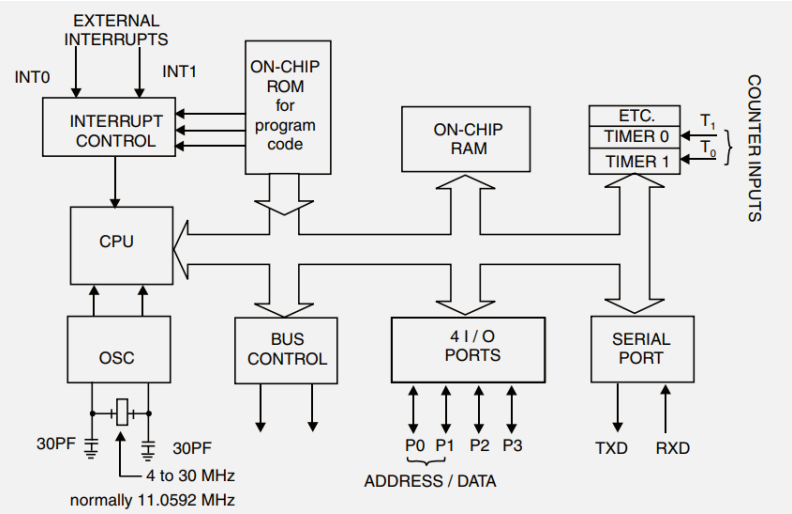

Figure 1: 8051 Architecture

You need to understand three (3) parts of 8051including –Oscillator and I/O ports to understand this experiment.

Oscillator: It is used for providing the clock to 8051 MC (using to input pins XTAL2, XTAL1) and decides the speed of MC. In this experiment, you are going to use crystal oscillator and its frequency vary from 4MHz to 30 MHz. but normally it formulates11.0592 MHz frequency.

Input Output Ports: There are four input output ports available P0, P1, P2, P3. Each port is 8 bit wide and has special function registers P0, P1, P2, P3 which are bit addressable, means each bit can be set or reset by the Bit instructions (SETB for high, CLR for low) independently. The data at any port which is transmitting or receiving is in these registers. The port 0 can perform dual works. It is used as Lower order address bus (A0 to A7) multiplexed with 8 bit data bus P0.0 to P0.7 is AD0 to AD7 respectively the address bus and data bus is demultiplex by the ALE signal and latch which is further discussed in details. P1 is a true I/O port (P1.0 to P1.7), because it doesn't have any alternative functions as is the case with P0, but can be configured as general I/O only. Port 2 can be used as I/O port as well as higher order address bus A8 to A15. Port 3 also have dual functions it can be worked as I/O as well as each pin of P3 has specific function and you will learn in details of each port later.

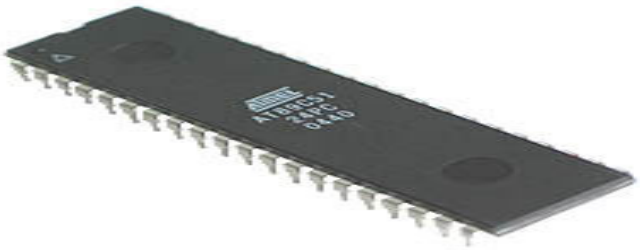

Figure 2: AT89c51 Architecture

Tools:

1. **KeilµVission5 -** Keil Microcontroller Tool includes C/C++ compilers, integrated development environments, RTOS, middleware, as well as debug adapters and evaluation boards for Arm Cortex®-M based devices.

2. **Proteus 8 Professionals-** Proteus is a complete software solution for circuit simulation and PCB design. It comprises several modules for schematic capture, firmware IDE and PCB layout that appear as tabs inside a single, integrated application. Proteus virtual system modeling (VSM) bridges the gap in the design life cycle between schematic capture and PCB layout. It enables you to write and apply your firmware to a microcontroller component on the schematic (PIC, AVR, ARM, 8051, etc.) and then co-simulate the program within a mixed-mode SPICE circuit simulation.

Experiment-1: Complete the following task according to given instruction

1. Open: Keil**µ**Vission 5

2. Project (From Menu Bar)
3. New Vission Project
4. Create a project folder in Desktop (e. g. CSE442lab) and
5. Open a file with a name (e.g. cse442)
6. Save
7. A window appears: Select device for Target 1 'Target 1'
8. Click on ATML (+)
9. Select AT89C51 (8051 based Fully Static 24 MHz CMOS controller with 32 I/O lines)
10. OK
11. Yes
12. Click File (From menu bar)
13. Select new
14. File (from menu bar)
15. Click on Save as
16. Give a file name with extension .c (e.g. cse442.c)
17. Write the following code on text file.

```
#include<AT89X51.h>
        void delay(unsigned int);
        void main(void)
        {
                P1_1=0;
                P1_2=0;
                delay(300);
                P1_1=1;
                P1_2=1;
                delay(300);
        }
        void delay(unsigned int itime)
        {
                int i,j;
                for(i=0;i<itime;i++);
                for(j=0;j<5000;j++);
        }
```

18. Click on +Target 1 (From left side of the window)
19. Click on Rt. Mouse button on Source Group 1
20. Select Add existing files to Group 'Source Group 1'
21. Select : file cse442.c
22. Click on Add
23. Click on Close
24. Create an environment for creating a Hex file by clicking Target 1 (Rt. Mouse button)
25. Select Option for Target 'Target 1'
26. Click on Target

27. Set frequency Xtal (MHz): 12 MHz
28. Select Output and
29. Click on Create Hex File (Hex: 80)
30. OK
31. Click on Translate (or Press CTRL + F7: To see the errors on the code) (Two steps below the Menu bar)
32. Click on Build (or F7)
33. Go to Project folder (CSE442lab)
34. Go to Objects folder in the Project folder (CSE442lab)
35. Find Hex file (cse442.hex)

Go to software PROTEUS Professionals:

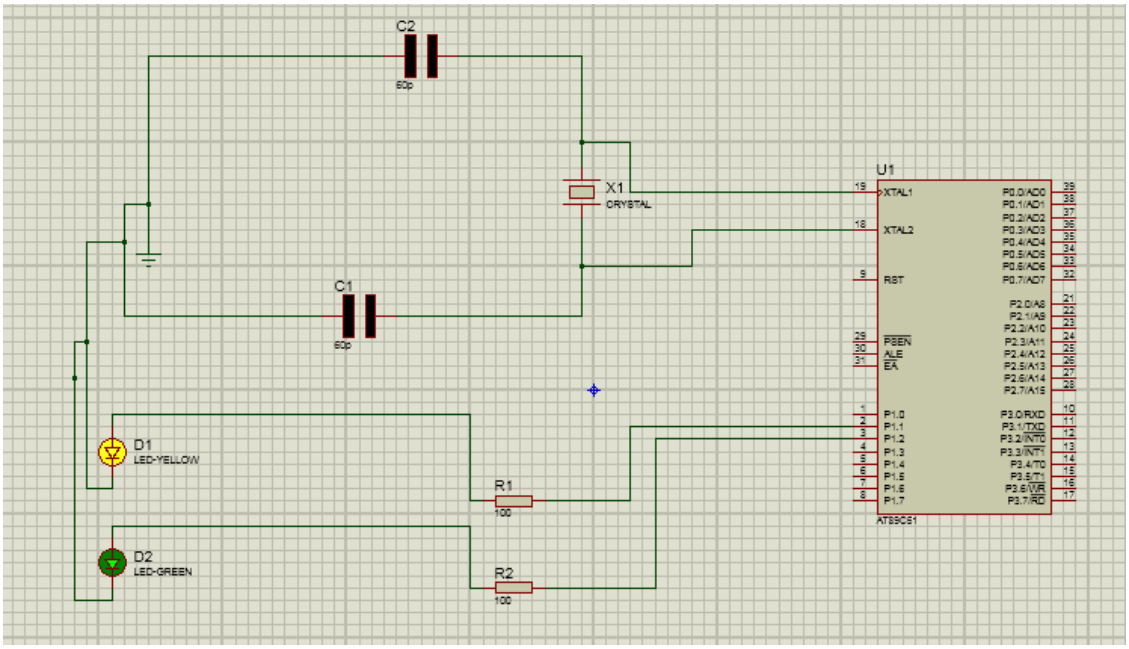

Figure 3: Circuit Diagram of the Experiment

Draw a circuit diagram by selecting the appropriate tools from the software:

1. Open: Proteus 8 Professionals
2. Select: ISIS
3. Select: **P** (Pick Devices) for Searching components to design a circuit diagram
4. Components: AT89C51, Resistor, LED, Capacitors, Crystal to design above diagram,(by typing in the keyword area) and select terminal mood from right side for ground.
5. Set value of all resistors 100 ohm(right mouse click on resistor)
6. Set value of all capacitor 60 pico
7. Set value crystal 12MHz
8. Right click on AT89C51(circuit) and select program file from project folder->object folder-> cse442.hex file->open-> ok

9. Click Run (On the bottom left corner) of the screen.

Experiment-2: Complete each of the following tasks and write the effect in your system.

- Change the code to increase delay (3000)
- Change the clock speed from 12MHz to 24MHz