**Department of Electrical Engineering**

# Faculty Member:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Lab Engineer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Semester/Section:\_\_\_\_\_\_\_\_\_\_\_\_\_ Dated: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**EE-222 Microprocessor Systems**

Lab9: Write a Program That Turns on an LED

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Reg. no.** | **Report**  **Marks / 10** | **Viva Marks**  **/ 5** | **Total/15** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**Objective:**  Program the microcontroller AT89c52 to turn ON/OFF an LED (connected to Pin P1.1) on the Trainer, based on whether a switch (connected to Pin P1.2) is ON or OFF, and patch the circuit on breadboard and test it.

**Software Used:** Keil, Proteus

### What is a Microcontroller?

The microcontroller is an IC that has its own CPU, RAM and non-volatile memory (ROM, EPROM, FLASH EPROM) on the same chip.

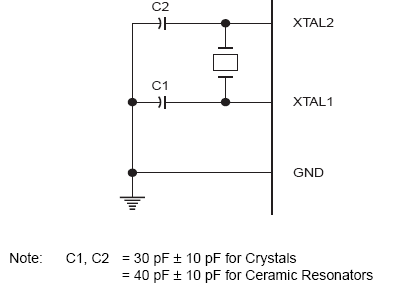
### 8051 ARCHITECTURE:

The AT89C52 (AT – ATMEL) is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel’s high density nonvolatile memory technology and is compatible with the industry standard MCS- 51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in- system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications. Its features are as follows:-

* 4K Bytes of In-System Reprogrammable Flash Memory
* Endurance: 1,000 Write/Erase Cycles
* Fully Static Operation: 0 Hz to 24 MHz
* Three-Level Program Memory Lock
* 128 x 8-Bit Internal RAM
* 32 Programmable I/O Lines
* Two 16-Bit Timer/Counters
* Six Interrupt Sources
* Programmable Serial Channel
* Low Power Idle and Power Down Modes

### OSCILLATOR CHARACTERISTICS:

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL1 and XTAL2 should be is driven as shown in Figure below:



**Exercise:**

**Step 1:** Create a new project on Keil using the assembly code given below and create the hex file.

ORG 0H ; Starting address of the ROM

SJMP START ; Jump to the beginning of the program

ORG 40H ; First address above the Interrupt Vector

START: ; Start Label

CLR P1.1 ; Turn off the LED to start with

HERE:

JNB P1.2, HERE ; If the Switch is off then wait here

SETB P1.1 ; Turn on the LED

WAIT:

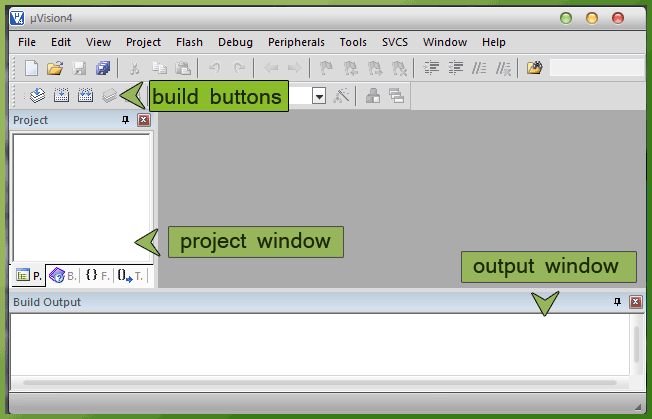
JB p1.2,WAIT

SJMP START

END

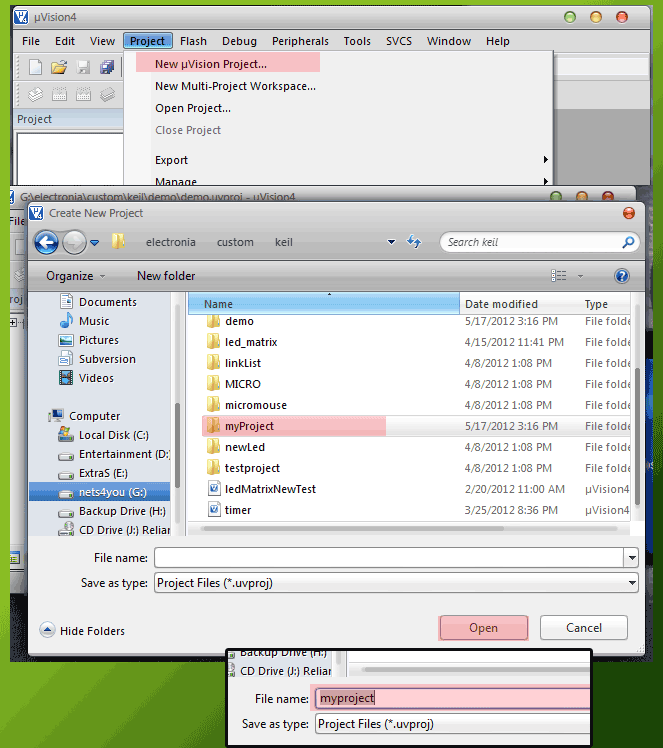
**Step 2: Keil Setup and Detail**

Your IDE looks like this and if not then you can select following option in view tab as shown in second image

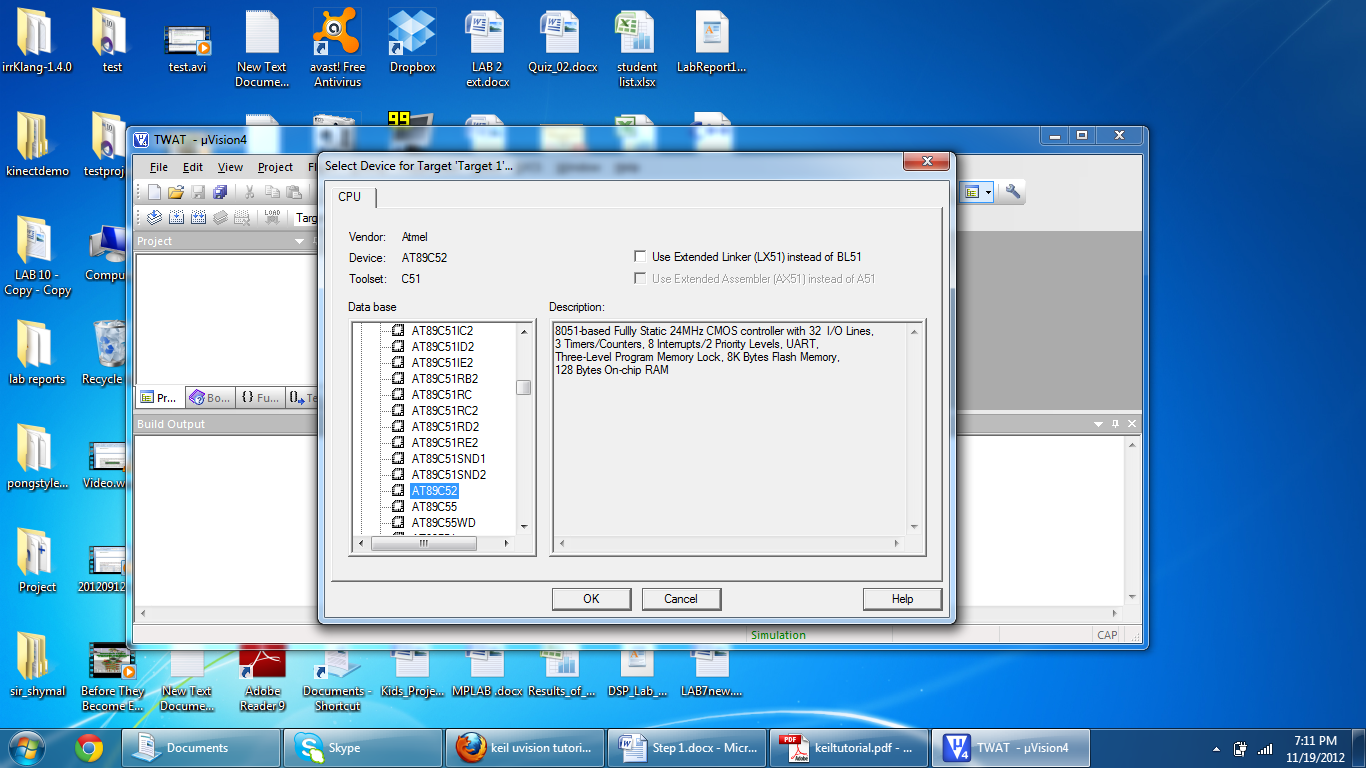
[](http://3.bp.blogspot.com/-4VHG_d4sZ0Y/T7UJP39vQAI/AAAAAAAAAX8/1AOuxb7rXK8/s1600/img1.gif)

**Step 3: Project setup**

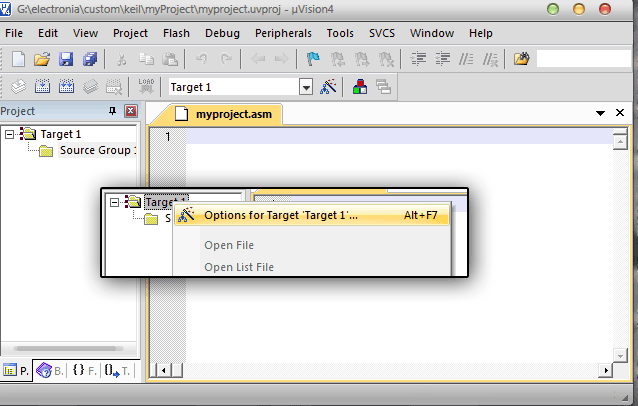
To setup your project click on Project tab >> New uVision Project >> a dialog box is open  
Then make a new folder (myproject) so your all files of this project not mix with other project files  >> open that folder  
Choose a Project name (myproject)

[](http://1.bp.blogspot.com/-AQqMQZ-dwqI/T7Thn2lctfI/AAAAAAAAAWY/daVyYaXmeHg/s1600/img3.gif)

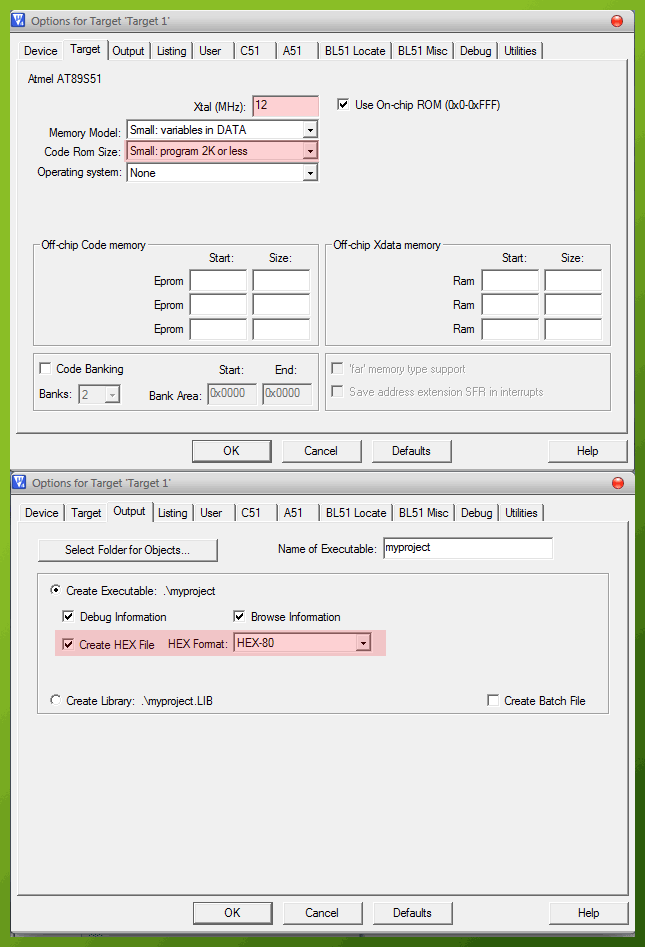
**Step 4: Select Target Device**  
After saving your file this new window open to select you CPU Now select your microcontroller (here use  AT89c52) expand Atmel >> and select AT89c52 (or which you use) >> click OK



**Step 6: Target setup**  
Right click on Target 1 Group >> and select Option for Target

[](http://4.bp.blogspot.com/-y_see9xyzgo/T7TxdM4aaBI/AAAAAAAAAW8/aqvm97GaWq4/s1600/img11.gif)

A dialog box opens as shown in the image below.  
Change the setting as shown in Red Area (for Target and Output)

[](http://1.bp.blogspot.com/-UZbas8qQw38/T7TzdbjiJSI/AAAAAAAAAXE/tzhw-S_snI8/s1600/img13.gif)

**Step 6: Writing Code**

Now make a new file and save it with .a extension (myproject.a),  
and put this test code in file and save.

ORG 0H ; Starting address of the ROM

SJMP START ; Jump to the beginning of the program

ORG 40H ; First address above the Interrupt Vector

START: ; Start Label

CLR P1.1 ; Turn off the LED to start with

HERE:

JNB P1.2, HERE ; If the Switch is off then wait here

SETB P1.1 ; Turn on the LED

WAIT:

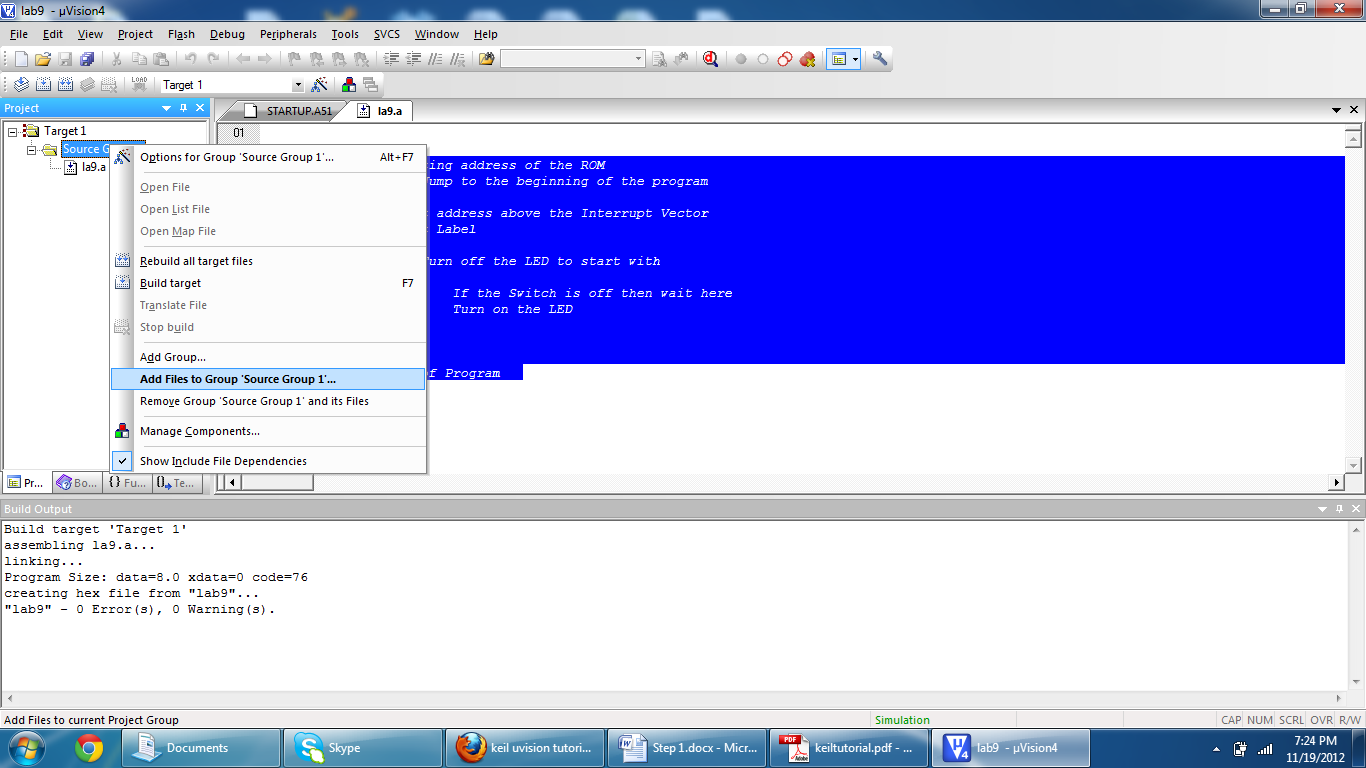
JB p1.2,WAIT

SJMP start

END ; End of Program

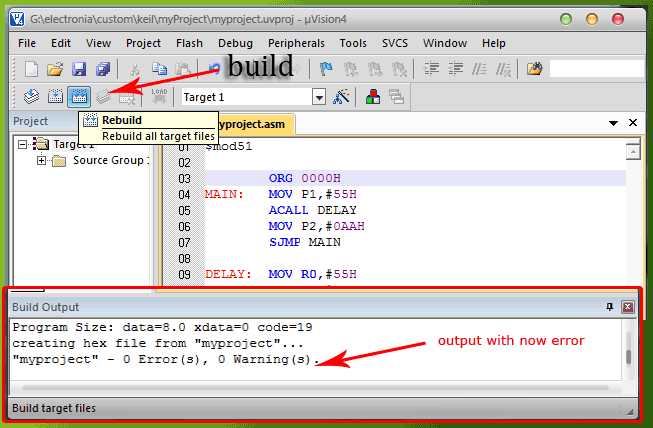
**Step 7: Add file to project**

Now to add saved asm file to your project, right click Source Group 1 and select add files to the group as shown below .change file type asm (.a) to select the myproject.a

****

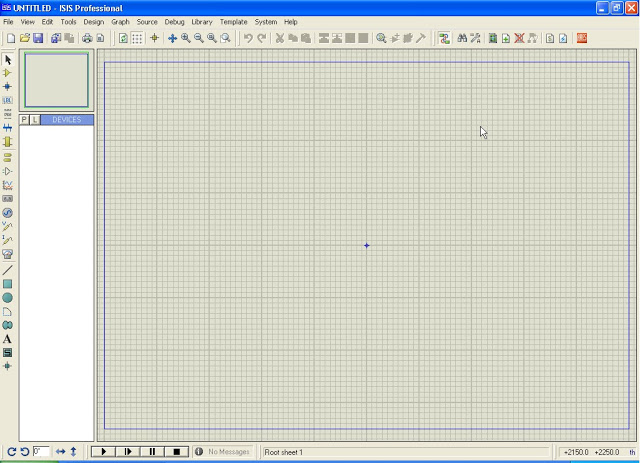
|  |  |
| --- | --- |
| **Step 8: build and create hex file** |  |

Now build your code by pressing F7 or Go to project menu and select build target. This will also create your hex file that you can burn into the microcontroller EEPROM.

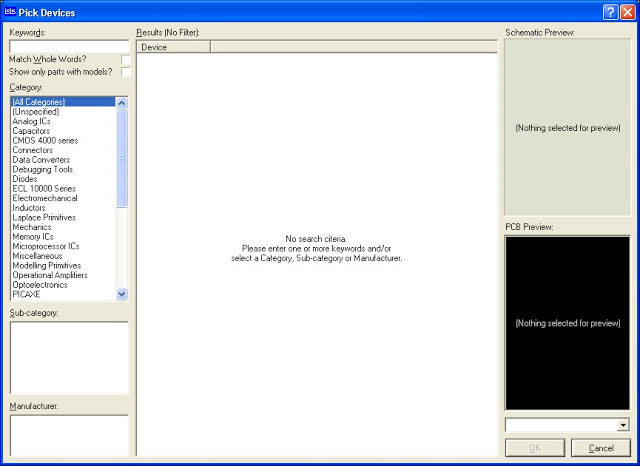
[](http://2.bp.blogspot.com/-G37Bt2psBRI/T7T4eE_6VFI/AAAAAAAAAXY/2lC9RDrz1yM/s1600/IMG17.gif)

**How to work with Proteus**

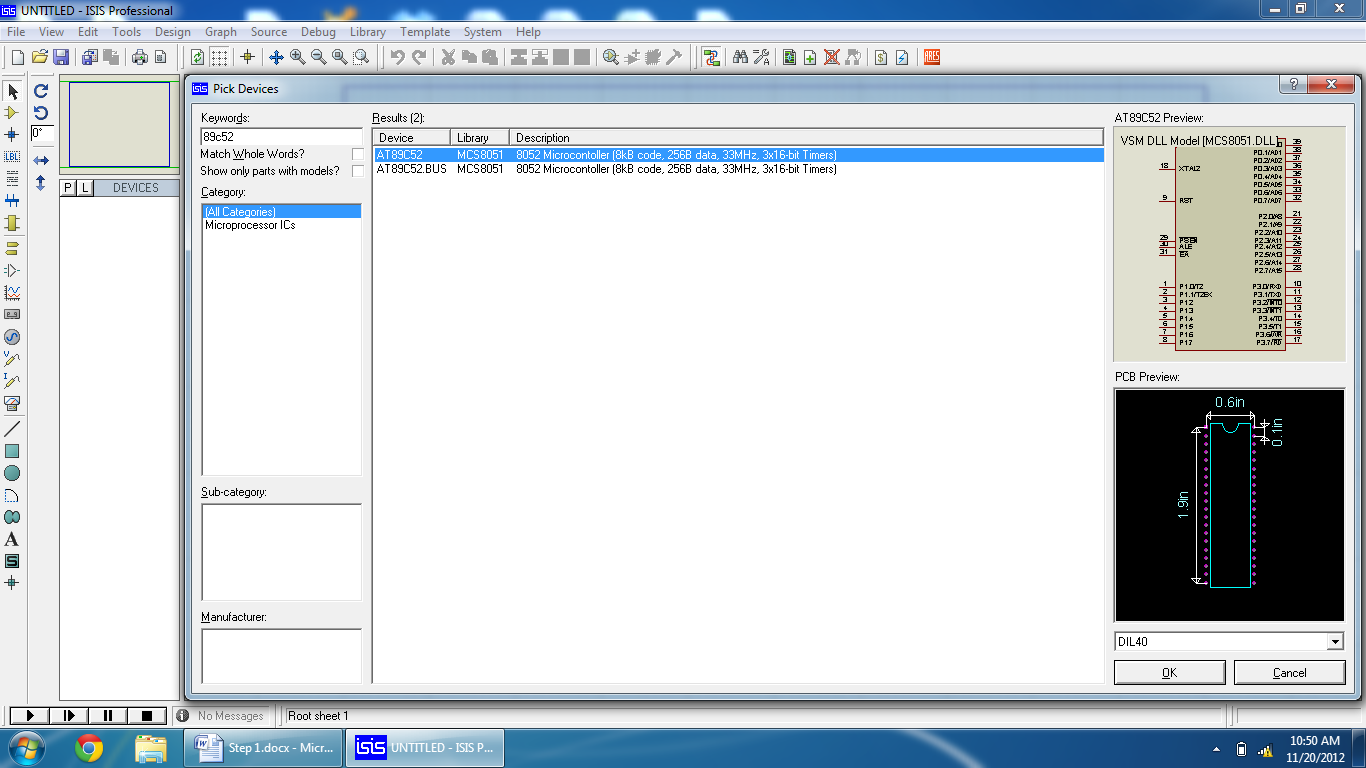
*Here you can study about Proteus ISIS and ARES in step by step with images*

[](http://3.bp.blogspot.com/-3g2UBazlIWI/UDI9XPP3zCI/AAAAAAAAA-o/VqNjXEoMTiM/s1600/First+window.jpg)

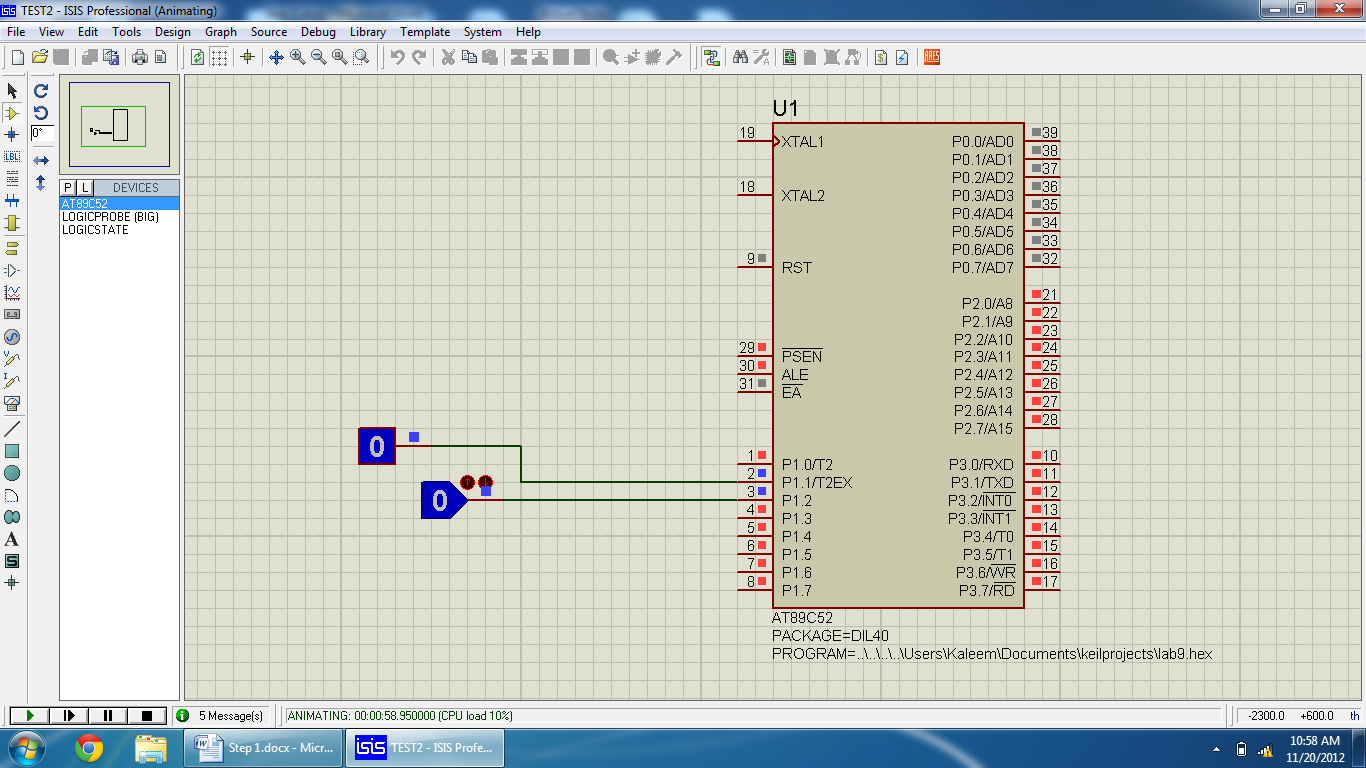
**Go To Start >> All Programs >> Proteus 7 Professional >>ISIS 7 Professional**  
Now you can see the above window then click the P on left side

[](http://1.bp.blogspot.com/-Q5ScWSHrG8k/UDI-ADW-lqI/AAAAAAAAA-w/MmkLa4SiHfA/s1600/Second.jpg)

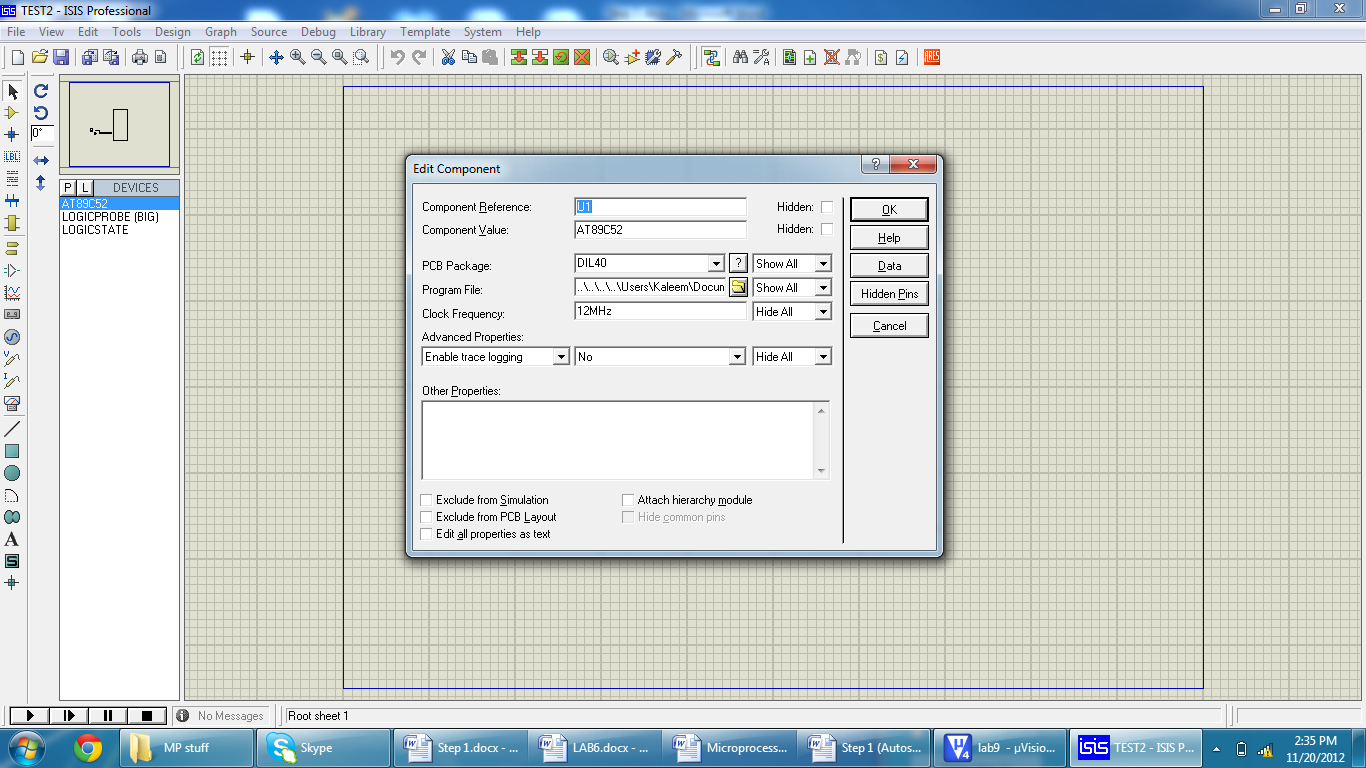
* Now you can see this window. Then enter your keyword in keyword input. If you want 89c52 microcontroller, enter 89c52 in keyword input.



* Now the related items are showing in the 'Pick Devices' window. Then select microprocessor >> Select which types of IC you want in the sub category section. Now select the IC ( you can see the PCB preview in the right sidebar )
* Now you can see the Microcontroller IC in devices’ section. Select the components. Now place where you want these to appear. Now you can go to make wires, pencil mark used to making wire. Then select which pin you want to connect to making wire then drag the pencil mark to end point pin.



* Now you can see the finished wire in this window
* once your circuit is complete double click the microcontroller IC and you will see “Edit component ” dialog box select the hex file from Program file tab and set the Clock frequency to 12Mhz



* Now click the play button Present on the left bottom of the screen to simulate your circuit.

**Step 9: Burn the tested program into the microcontroller.**

**Step 10: Patch the circuit on breadboard and test its function**

## Lab Tasks 2:

**Write a Program that change LEDs state on port 1, based on a switch (connected to Pin P2.1), when the switch changes from 0 to 1 and remain the same state for all others.**

*Show your codes and proteus circuit in lab report.*