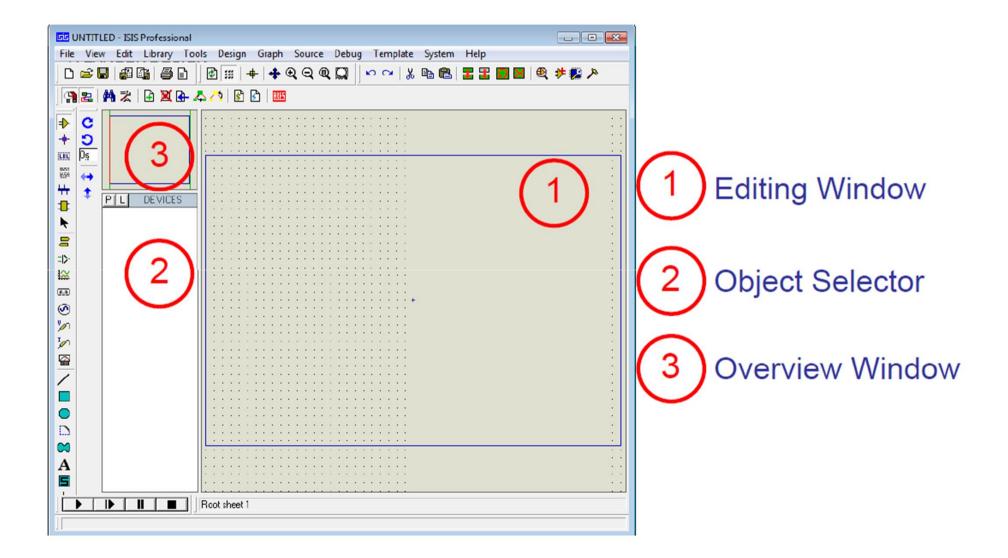
Microprocessors and Microcontrollers (EE-231)

Lab-2

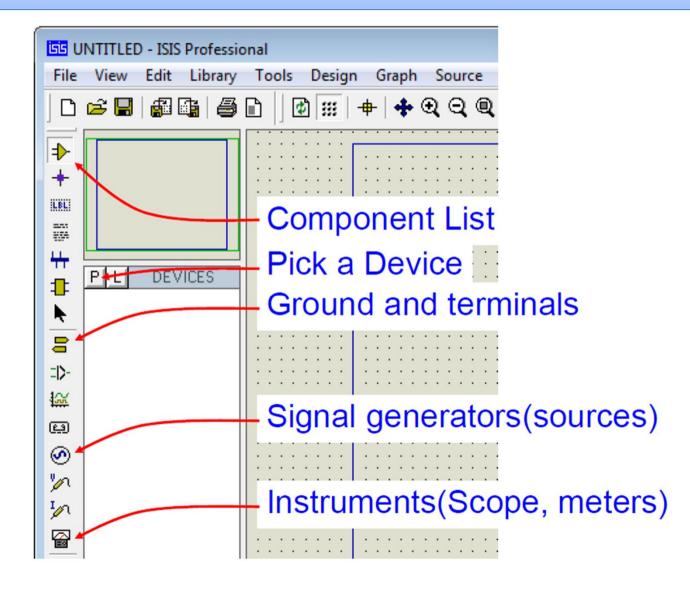
# Main Objectives

- Introduction to Proteus ISIS and its use in 8051 simulation.
- Writing an Assembly program in KEIL and Its debugging.
- Generating Delay using a subroutine in Assembly.
- Generating Delay using For Loop in "C".

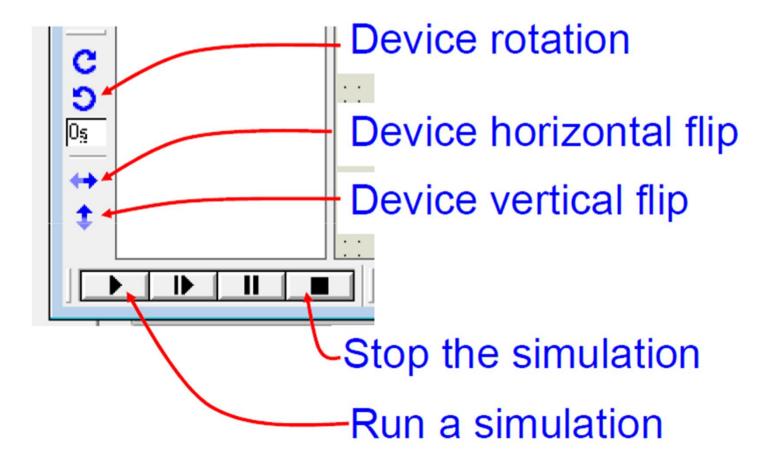
#### **Proteus Environment**



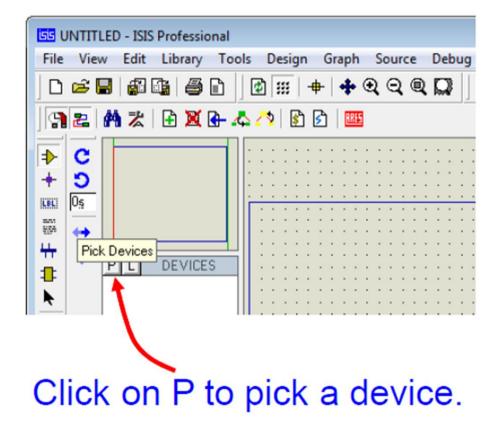
#### **Tools and Buttons**



#### **Tools and Buttons**



#### Picking a Device



## Picking a Device

- In order to pick a microcontroller
- Now either go in "Microprocessors category" or in keyword space write "AT89C51"

isis Pick Devices				
Keywor <u>d</u> s:		<u>R</u> esults (8):		
At89c51		Device	Library	Description
Match Whole Words?		AT89C51	MCS8051	8051 Microcontoller (4kB code, 33MHz, 2x16-bit Timers, UART)
Show only parts with models?	~	AT89C51.BUS	MCS8051	8051 Microcontoller (4kB code, 33MHz, 2x16-bit Timers, UART)
Category:	_	AT SC51RB2 AT SC51RB2.BUS	MCS8051 MCS8051	8051 Microcontoller (16kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART) 8051 Microcontoller (16kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)
(All Categories)		AT8 C51RC2 AT85 C51RC2.BUS	MCS8051	8051 Microcontoller (32kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)
Microprocessor ICs		AT89051RD2	MCS8051 MCS8051	8051 Microcontoller (32kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART) 8051 Microcontoller (64kB code, 40MHz, Watchdog Timer, 3x16-bit Timers, UART)
		AT89051RD2.BUS		8051 Microcontoller (64kB code, 40MHz, Watchdog Timer, 3x16-bit Timers, UART)

Double click on this highlighted line.

## Picking a Device

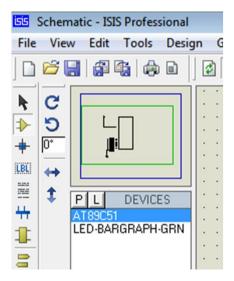
- Now in order to pick a " BAR LED"
- write BAR LED in keyword space or choose 'Optoelectronics'

55 Pick Devices			
Keywor <u>d</u> s:	<u>R</u> esults (2):		
BAR LED	Device	Library	Description
Match Whole Words?	LED-BARGRAPH-GRN	DISPLAY	Green LED Bargraph Display
Show only parts with models?	LED-BARGRAPH-RED	DISPLAY	Red LED Bargraph Display
Category: (All Categories) Analog ICs Optoelectronics			

Double click on whichever color you want

## **Drawing Schematic**

• Now choose device from the list and just left click in the editing window/schematic window.



• An outline of the device will appear. Place it where you want to.

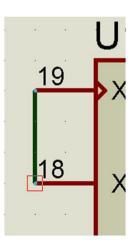
# **Connecting Wires**

• Just move your cursor over to the pin of the device, a small square box will appear. Click on that pin.



- Now a wire will follow the mouse cursor.
- Take the cursor to the pin you want the wire to connect to.

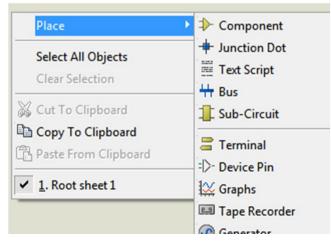
• Click on that pin and connection will be made.



## Placing Vcc and GND

- In order to place a Vcc or GND terminal in your schematic, just select terminal mode from toolbar.
- A list of terminals will appear.
- Choose whichever terminal you want to place and then click on the schematic to place the selected terminal.

We can place any item by right clicking And Selecting "Place"



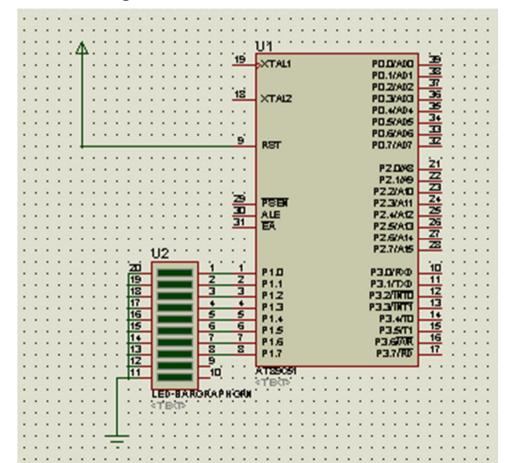


GROUND

**Terminals Mode** 

### Simulating 8051

• Draw the following schematic



#### Downloading Hex File in 8051

• Just double click on the 8051 IC schematic.

button.

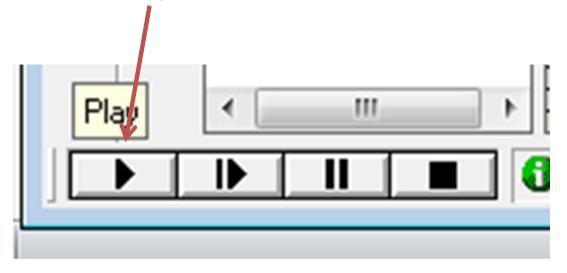
• Now browse for "Program file by clicking on the browse

Edit Component Component <u>R</u> eference:	U1	Hidde		? <b>X</b>	
				<u>0</u> K	
Component <u>V</u> alue:	AT89C51	Hidde	en:	<u>H</u> elp	
PCB Package:	DIL40	▼ ? Hide All	-	<u>D</u> ata	
Program File:	\\Test\Assembly.hex	🔄 Hide All	-	Hidden <u>P</u> ins	
Clock Frequency:	11.0592MHz	Hide.All			
Advanced Properties:			l	<u>C</u> ancel	
Enable trace logging	▼ No	▼ Hide All	•		
Other Properties:					
			A		
			~		
Exclude from <u>Simulation</u> Exclude from PCB <u>Layou</u>		archy <u>m</u> odule on pins			

• Pick your desired hex file and click 'open'

# Running Simulation.

• Click on the "Play" button to run the simulation



## Creating Assembly file in KEIL

- Create the new file.
- Save it as .asm file.
- Now when you add this file to the Project it will be an assembly file.
- You can write your assembly code in this file.

# Creating Assembly file in KEIL

- Write the following code and run it in Proteus.
- ORG 0
- MOV A, #055
- MOV P1, A
- END

#### LAB Task

- 1. Write a code that toggles the bits of P1 port Continuously.
- 2. Write a code that toggles the bits of P1 port after some **delay**. Use delay routine.

ORG 0
Start:
MOV A, #55H
MOV P1,A
MOV A, #OAAH
MOV P1,A
SJMP Start

01	ORG 0
02	Start:
03	MOV A, #55H
04	MOV P1,A
05	ACALL Delay
06	MOV A, #OAAH
07	MOV P1,A
80	ACALL Delay
09	SJMP Start
10	
11	ORG 100
12	Delay:
13	MOV R1,#255
14	Label1:
15	MOV R2,#255
16	Label2:
17	DJNZ R2, Label2
18	DJNZ R1, Label1
19	RET

# Delay in C

• To Generate delay in C simply use the for loop.

```
for(i=0;i<5000;i++);
```

• Or to multiply delay, use nested for loop.

```
for(i=0;i<500;i++)
for(i=0;i<10;i++);
```

## Task

 Write your name on easy 8051 Kit LEDs using delay in C.

#### Task

- void chr\_led(char);
- void delay(int);
- void main(void)
- {
- while(1)
- {
- chr\_led('B');
- delay(10);
- chr\_led('A');
- delay(10);
- chr\_led('B');
- delay(10);
- chr\_led('A');
- delay(10);
- chr\_led('R');
- delay(36);
- •
- }
- //Function Definitions
- void chr\_led(char mychar)
- {
- switch(mychar)
- •
- case 'A':// To write A
- P0=~0xFE;
- P1=~0x11;
- P2=~0x11;

- P3=~0xFE;
- break;
- case 'B':// To write B
- P0=~0xFF;
- P1=~0x99;
- P2=~0x99;
- P3=~0x66;
- break;
- case 'R':// To Write R
- P0=~0xFF;
- P1=~0x19;
- P2=~0x19;
- P3=~0xE6;
- break;
- }
- }
- void delay(int delay)
- {
- int i,j;
- for(i=0;i<10000;i++)
- for(j=0;j<delay;j++);</li>
- •