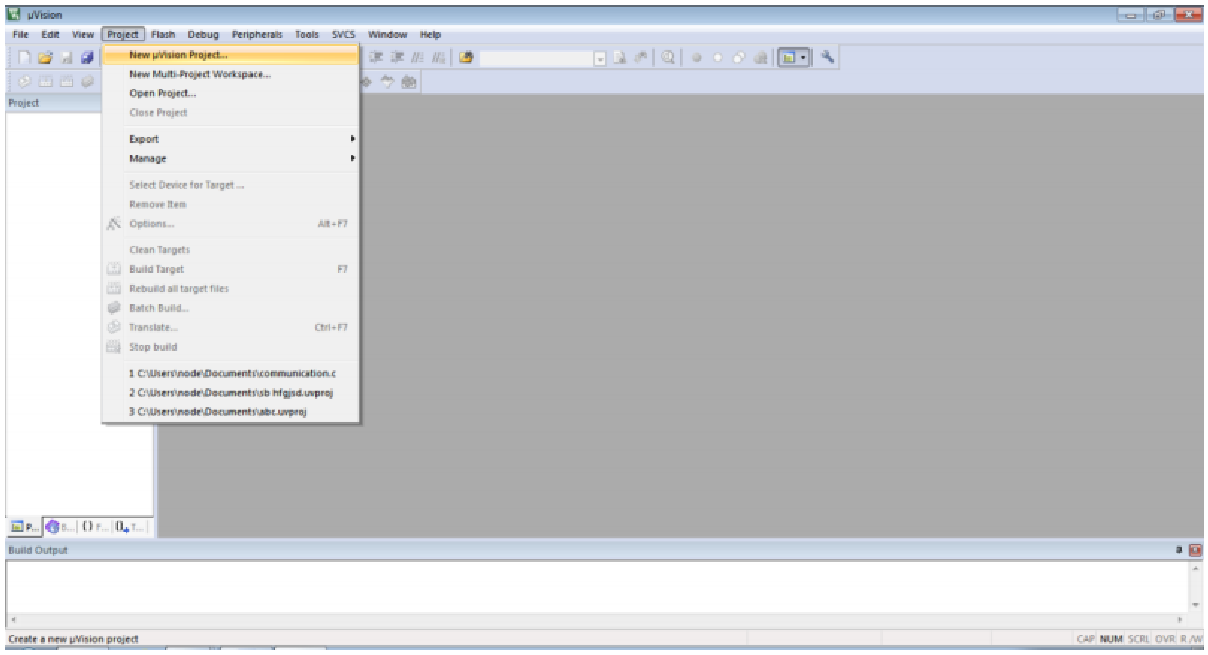
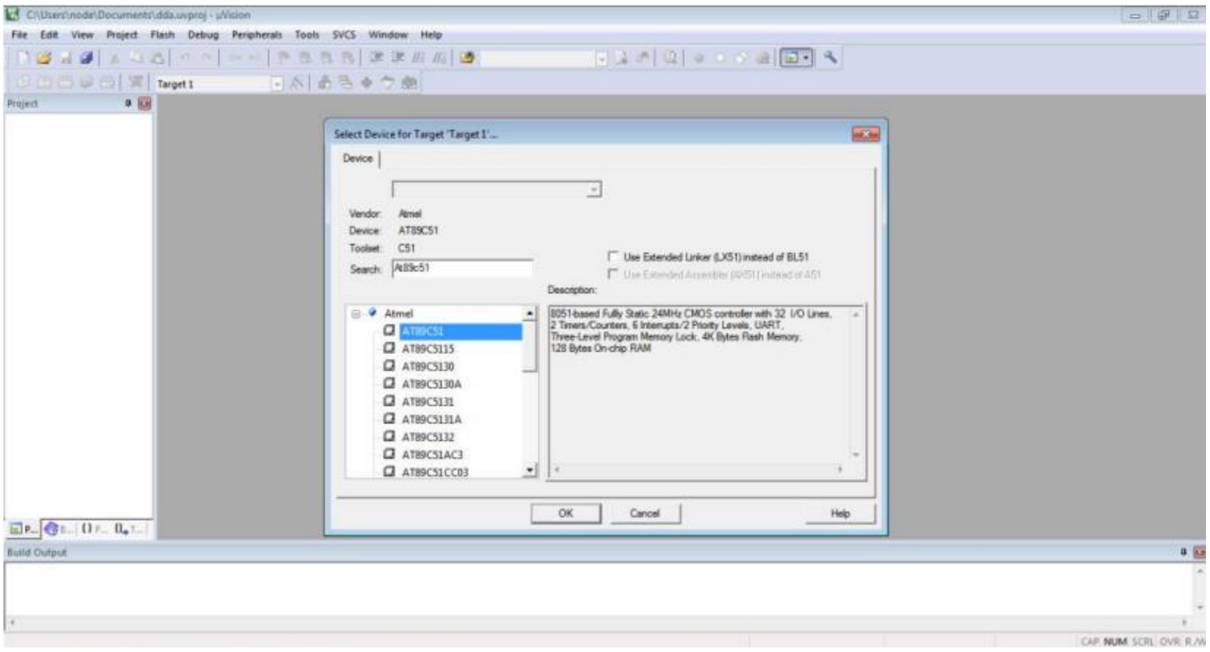
STEPS TO CREATE A NEW PROJECT IN KIEL SOFTWARE AND PROTEUS

1. OPEN Kiel software, after opening it from top left you will see a project option. CLICK on it then CLICK on new µ vision project. After selecting on µ vision project and SAVE it using PROJECT NAME


µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch Build...
Translate... Ctrl+F7
Stop build
1 C:\Users\node\Documents\communication.c
2 C:\Users\node\Documents\sb hfgjsd.uvproj
3 C:\Users\node\Documents\abc.uvproj
Project
Build Output
Create a new µVision project
CAP NUM SCRL OVR R/W

2. After saving choose processor (microcontroller)-------> AT89C51 --------> Click OK.

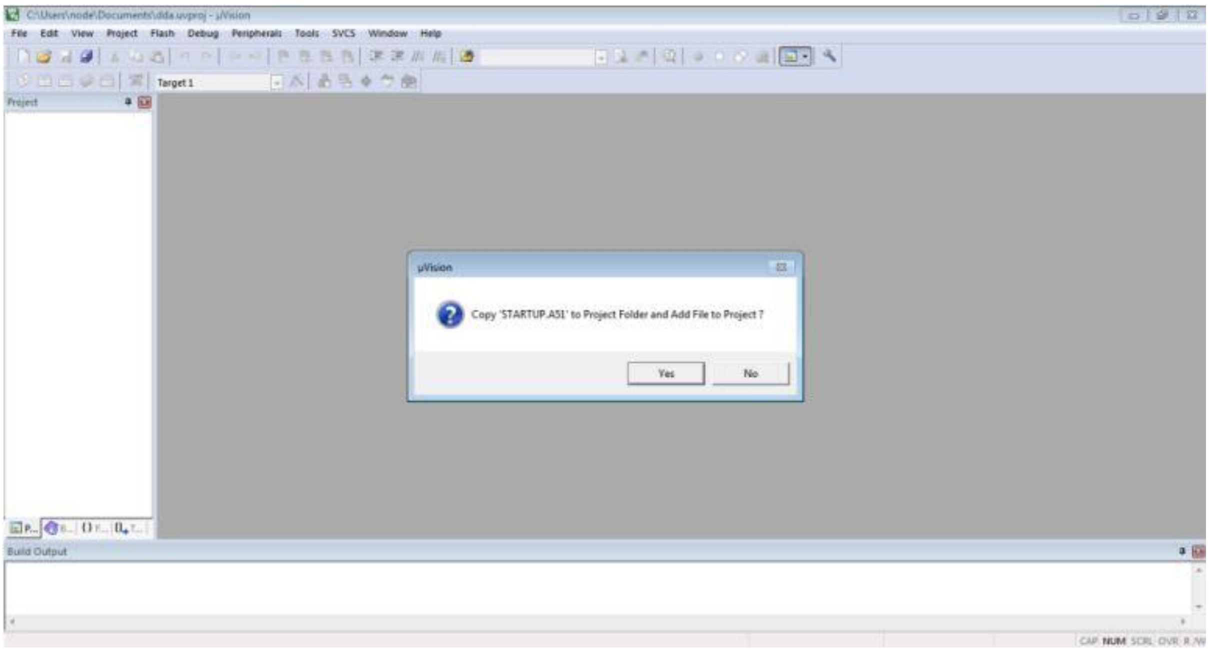


Vendor: Atmel
Device: AT89C51
Toolset: C51
Search: At89c51

Use Extended Linker (LX51) instead of BL51
Use Extended Assembler (AX51) instead of A51

Description:
8051-based Fully Static 24MHz CMOS controller with 32 I/O Lines, 2 Timers/Counters, 6 Interrupts/2 Priority Levels, UART, Three-Level Program Memory Lock, 4K Bytes Flash Memory, 128 Bytes On-chip RAM

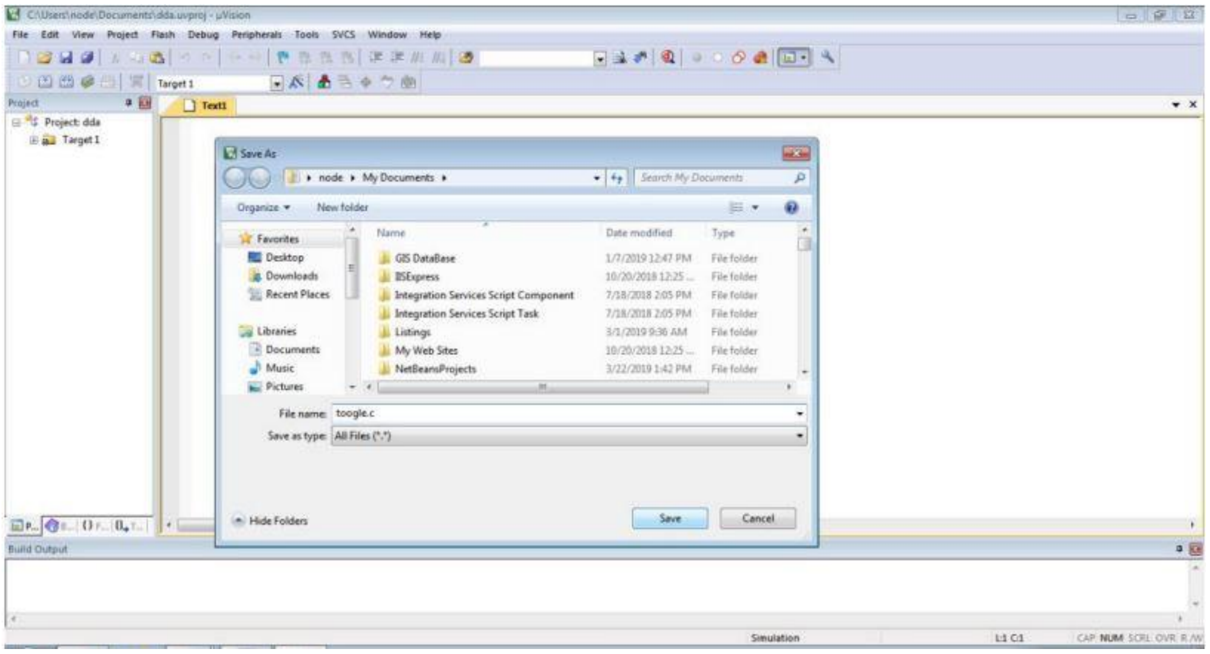
Then a window will open, select 'NO'.


C:\Users\node\Documents\dda.uvproj - µVision

µVision

Copy 'STARTUP.A51' to Project Folder and Add File to Project ?

3. Then Press CTRL-N (NEW FILE) and save the program by using Project name with ".C" .


C:\Users\node\Documents\dda.uvproj - µVision

Save As

node ▸ My Documents ▸

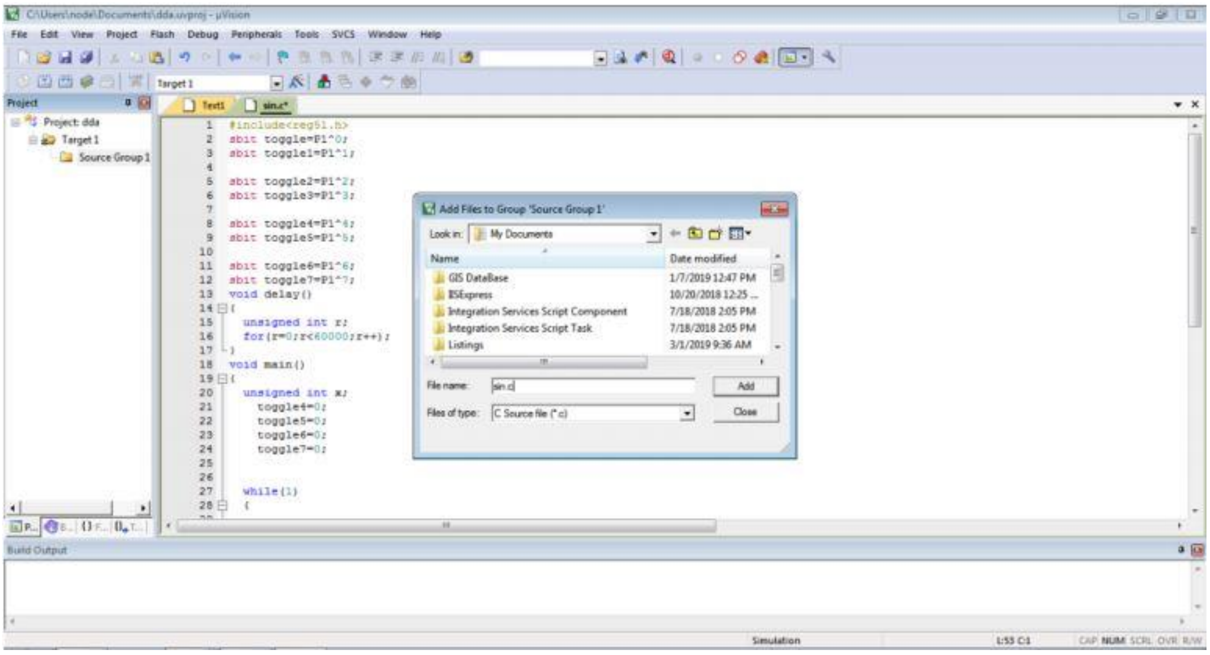
Name	Date modified	Type
GIS DataBase	1/7/2019 12:47 PM	File folder
IISExpress	10/20/2018 12:25 ...	File folder
Integration Services Script Component	7/18/2018 2:05 PM	File folder
Integration Services Script Task	7/18/2018 2:05 PM	File folder
Listings	3/1/2019 9:36 AM	File folder
My Web Sites	10/20/2018 12:25 ...	File folder
NetBeansProjects	3/22/2019 1:42 PM	File folder

File name: toogle.c

Save as type: All Files (*.*)

4. Write the appropriate code need for the PROGRAM and save the file with CTRL--S.

5. Click on Target -----> click on Source Group 1-------->Click add file to Source Group 1 click on the file that has been created and click add and close it.



C:\Users\node\Documents\dda.uvproj - µVision

```
#include<reg51.h>
sbit toggle=P1^0;
sbit toggle1=P1^1;

sbit toggle2=P1^2;
sbit toggle3=P1^3;

sbit toggle4=P1^4;
sbit toggle5=P1^5;

sbit toggle6=P1^6;
sbit toggle7=P1^7;
void delay()
{
  unsigned int r;
  for(r=0;r<60000;r++);
}
void main()
{
  unsigned int x;
    toggle4=0;
    toggle5=0;
    toggle6=0;
    toggle7=0;


  while(1)
  {
```

Add Files to Group 'Source Group 1'

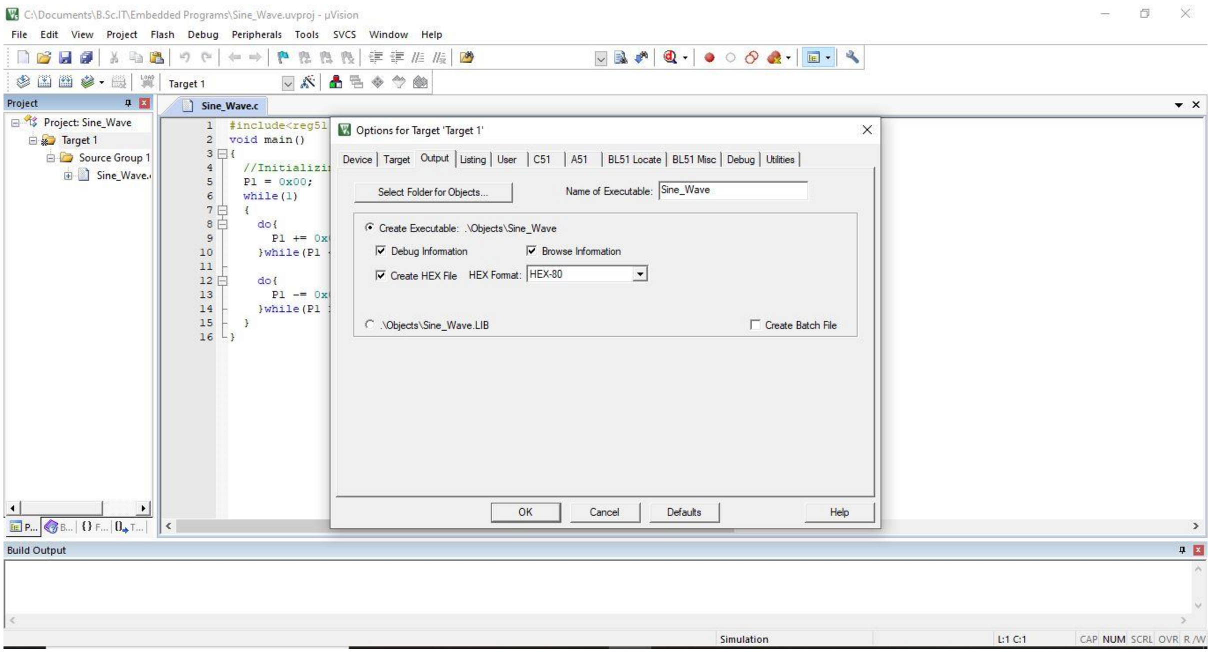
Look in: My Documents

Name	Date modified
GIS DataBase	1/7/2019 12:47 PM
ISExpress	10/20/2018 12:25 ...
Integration Services Script Component	7/18/2018 2:05 PM
Integration Services Script Task	7/18/2018 2:05 PM
Listings	3/1/2019 9:36 AM

File name: sin.c

Files of type: C Source file (*.c)

6. Right click on Target or (ALT---F7) --------> option to target a window SET frequency Xtal (MHZ)----->11.0592, CLICK ON the check box create a HEX file and press OK.



C:\Documents\B.Sc.IT\Embedded Programs\Sine_Wave.uvproj - µVision

```
#include<reg51
void main()
{
  //Initializi
  P1 = 0x00;
  while(1)
  {
    do{
      P1 += 0x
    }while(P1
    
    do{
      P1 -= 0x
    }while(P1
  }
}
```

Options for Target 'Target 1' — Output

Name of Executable: Sine_Wave

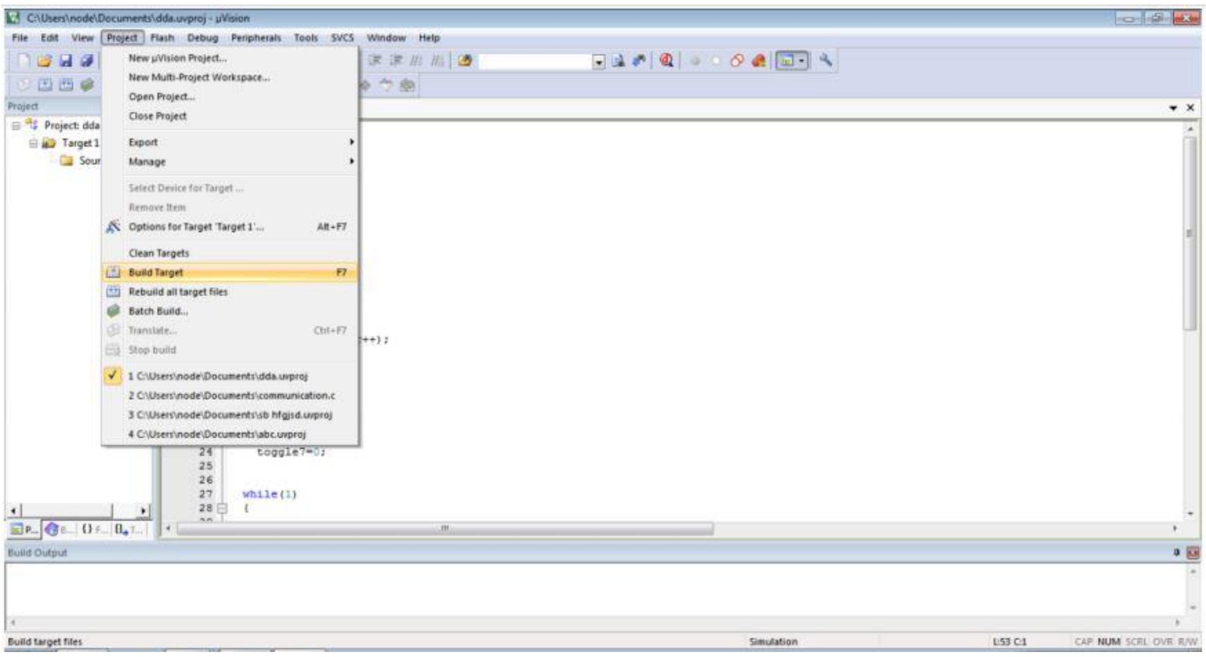
(•) Create Executable: .\Objects\Sine_Wave

[✓] Debug Information [✓] Browse Information

[✓] Create HEX File HEX Format: HEX-80

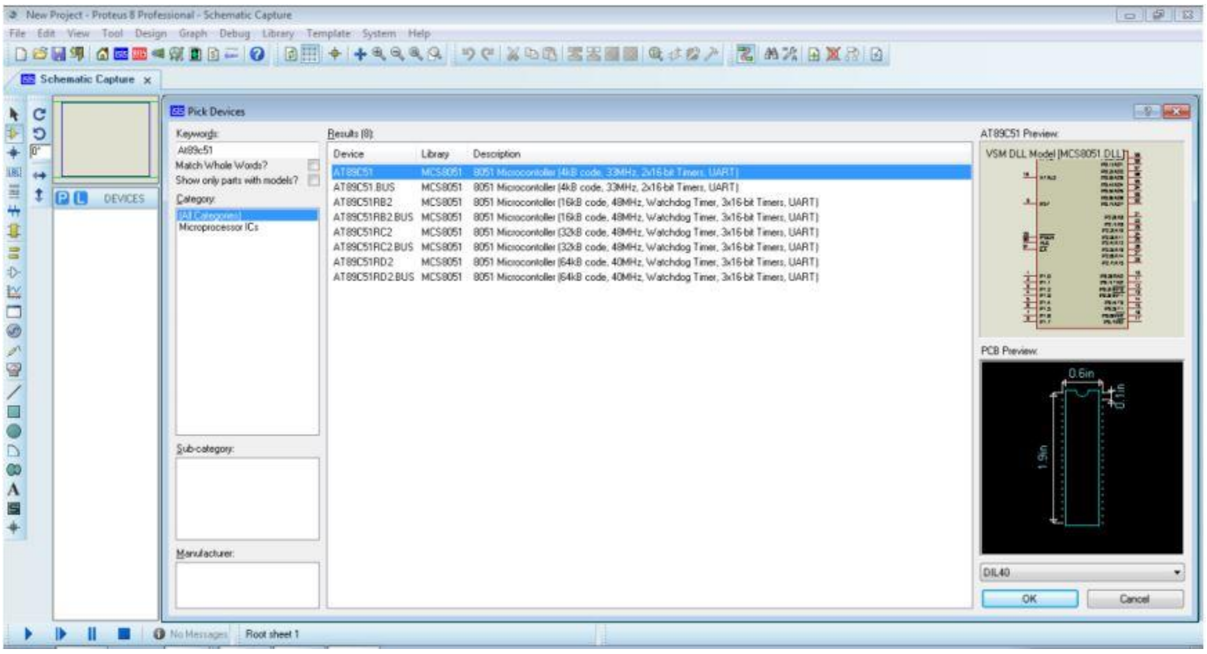
( ) .\Objects\Sine_Wave.LIB [ ] Create Batch File

7. Then build Target the build target (F7) will show you the error in the code.


C:\Users\node\Documents\dda.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
New µVision Project...
New Multi-Project Workspace...
Open Project...
Close Project
Export
Manage
Select Device for Target ...
Remove Item
Options for Target 'Target 1'... Alt+F7
Clean Targets
Build Target F7
Rebuild all target files
Batch Build...
Translate... Ctrl+F7
Stop build
1 C:\Users\node\Documents\dda.uvproj
2 C:\Users\node\Documents\communication.c
3 C:\Users\node\Documents\sb hfgjsd.uvproj
4 C:\Users\node\Documents\abc.uvproj
Project
Build Output
Build target files
Simulation
L:53 C:1
CAP NUM SCRL OVR R/W

```
        toggle7=0;


    while(1)
    {
```

8. OPEN PROTEUS SOFTWARE, Click on P Search for Atmel 89C51 on keywords and Other appropriate things needed for the simulation.


New Project - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
Pick Devices
DEVICES
Match Whole Words?
Show only parts with models?
Sub-category:
Manufacturer:
AT89C51 Preview:
VSM DLL Model [MCS8051.DLL]
PCB Preview:
0.6in
1.9in
DIL40
OK
Cancel
No Messages
Root sheet 1

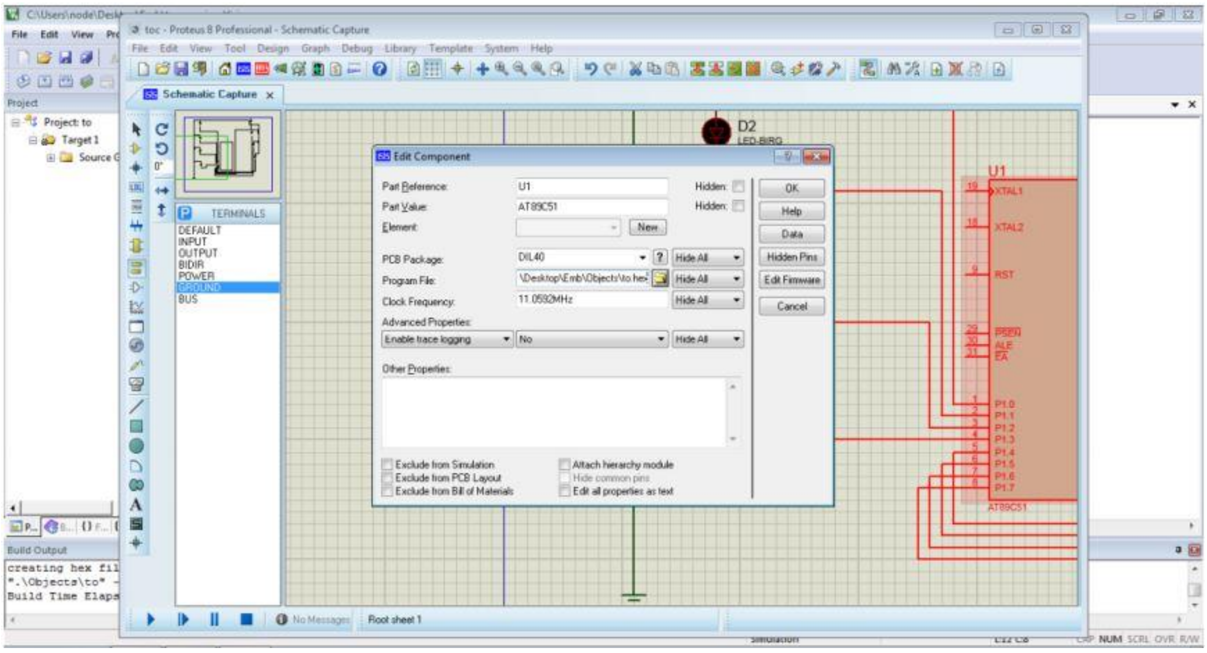
Keywords: At89c51

Category: (All Categories) Microprocessor ICs

Results (8):

Device	Library	Description
AT89C51	MCS8051	8051 Microcontoller (4kB code, 33MHz, 2x16-bit Timers, UART)
AT89C51.BUS	MCS8051	8051 Microcontoller (4kB code, 33MHz, 2x16-bit Timers, UART)
AT89C51RB2	MCS8051	8051 Microcontoller (16kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)
AT89C51RB2.BUS	MCS8051	8051 Microcontoller (16kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)
AT89C51RC2	MCS8051	8051 Microcontoller (32kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)
AT89C51RC2.BUS	MCS8051	8051 Microcontoller (32kB code, 48MHz, Watchdog Timer, 3x16-bit Timers, UART)
AT89C51RD2	MCS8051	8051 Microcontoller (64kB code, 40MHz, Watchdog Timer, 3x16-bit Timers, UART)
AT89C51RD2.BUS	MCS8051	8051 Microcontoller (64kB code, 40MHz, Watchdog Timer, 3x16-bit Timers, UART)

9. Then after selecting microcontroller chip and other thing use drag and select, to select the object after all the connection is done, right double click on the Atmel 89C51 set frequency 11.0592 and add HEX file to it and click “ok”.


Edit Component

Part Reference: U1 Hidden: ☐
Part Value: AT89C51 Hidden: ☐
Element:
PCB Package: DIL40 Hide All
Program File: \Desktop\Emb\Objects\to.hex Hide All
Clock Frequency: 11.0592MHz Hide All
Advanced Properties: Enable trace logging No Hide All
Other Properties:

☐ Exclude from Simulation
☐ Exclude from PCB Layout
☐ Exclude from Bill of Materials
☐ Attach hierarchy module
☐ Hide common pins
☐ Edit all properties as text

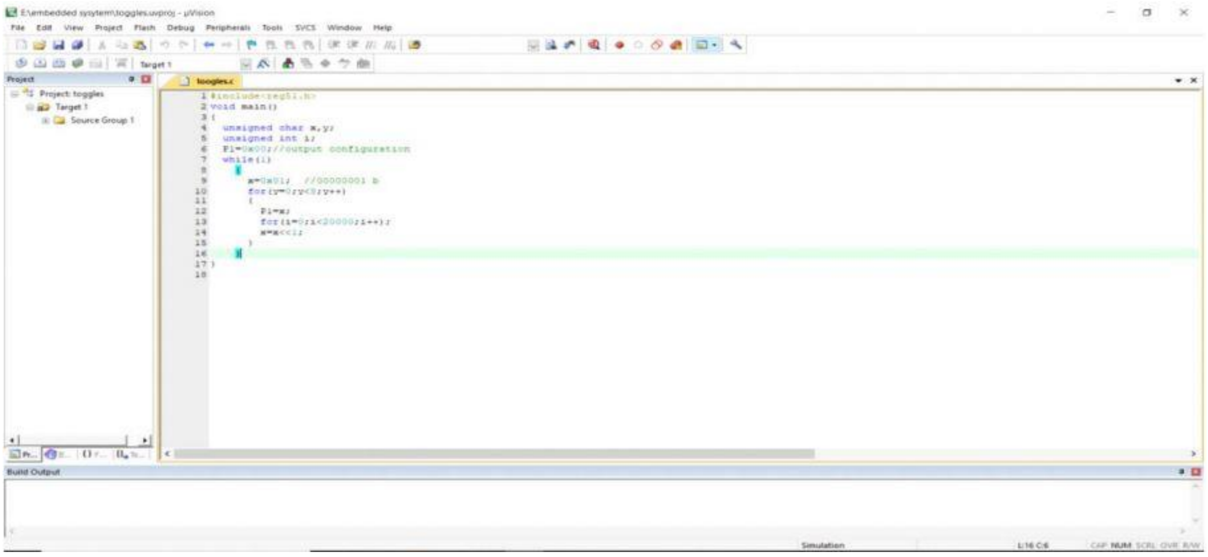
Build Output

```
creating hex fil
".\Objects\to" -
Build Time Elaps
```

10. And start SIMULATION.

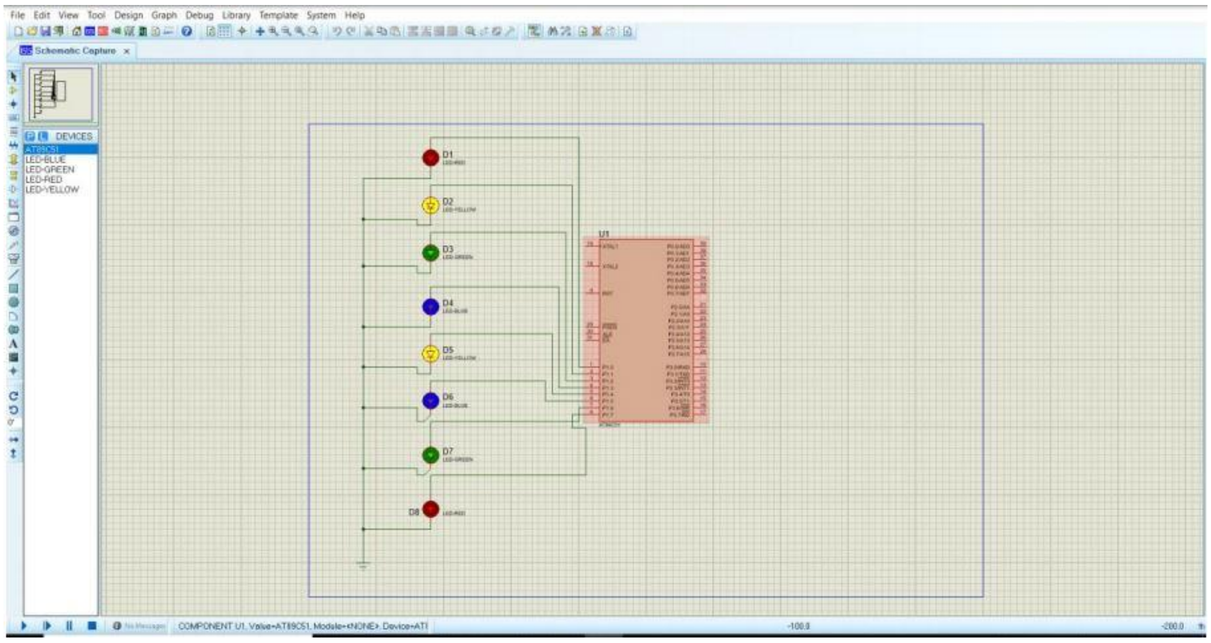
Practical: 1 TOGGLED LED

Step 1: Write a C program.



```
#include<reg51.h>
void main()
{
  unsigned char x,y;
  unsigned int i;
  P1=0x00;//output configuration
  while(1)
    {
      x=0x01;   //00000001 b
      for(y=0;y<8;y++)
      {
        P1=x;
        for(i=0;i<20000;i++);
        x=x<<1;
      }
    }
}
```

Step 2: Make the following setup.

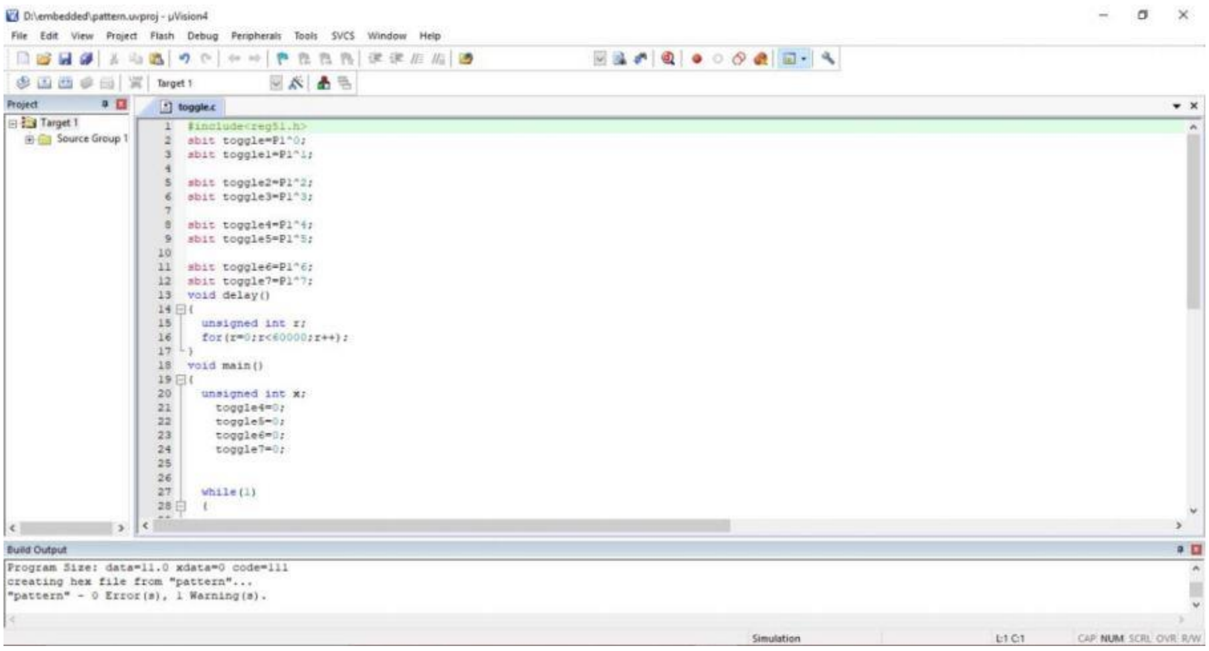


Source Code:

```
#include<reg51.h>
void main()
{
                    unsigned char x, y;
          unsigned int i;
          P1 = 0x00; //Output configuration
          while(1)
          {
                    x = 0x01; //000000001 b
                    for( y = 0 ; y < 0 ; y++ )
                    {
                              P1 = x;
                              for( i = 0 ; i < 20000 ; i++ )
                              x = x << i;
                    }
          }
}
```

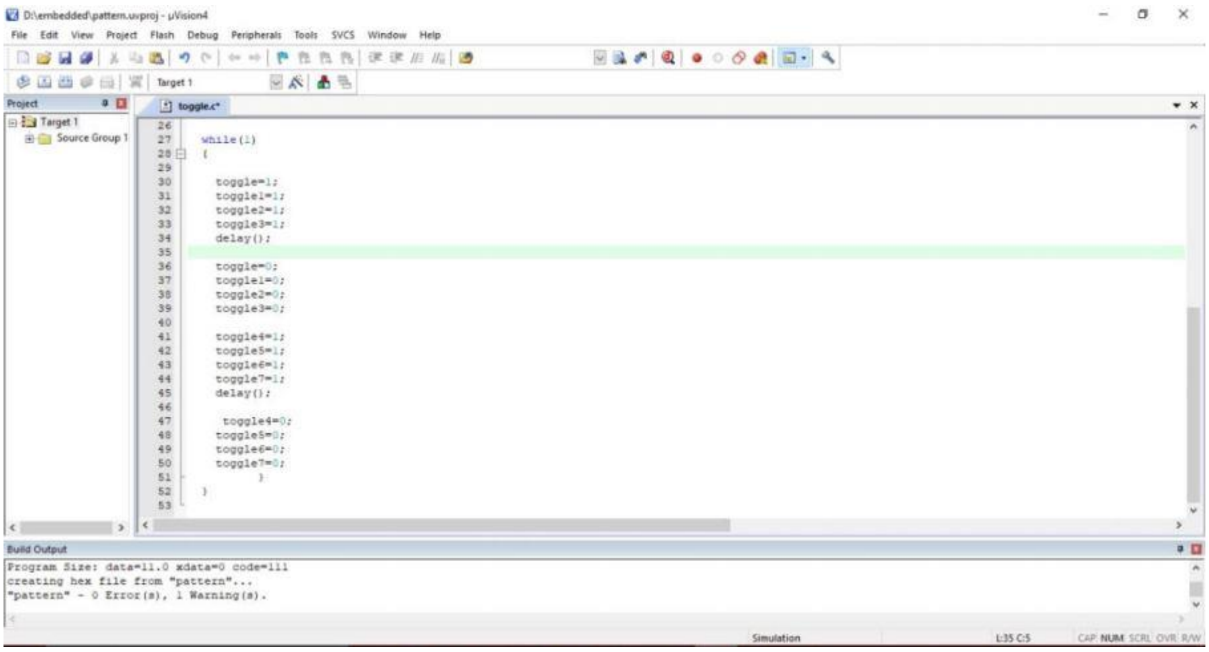
Toggled LED Pattern

Step 1: Write the C program.


D:\embedded\pattern.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
Source Group 1
toggle.c
Build Output
Simulation
L:1 C:1
CAP NUM SCRL OVR R/W

```
#include<reg51.h>
sbit toggle=P1^0;
sbit toggle1=P1^1;

sbit toggle2=P1^2;
sbit toggle3=P1^3;

sbit toggle4=P1^4;
sbit toggle5=P1^5;

sbit toggle6=P1^6;
sbit toggle7=P1^7;
void delay()
{
  unsigned int r;
  for(r=0;r<60000;r++);
}
void main()
{
  unsigned int x;
    toggle4=0;
    toggle5=0;
    toggle6=0;
    toggle7=0;


  while(1)
  {
```

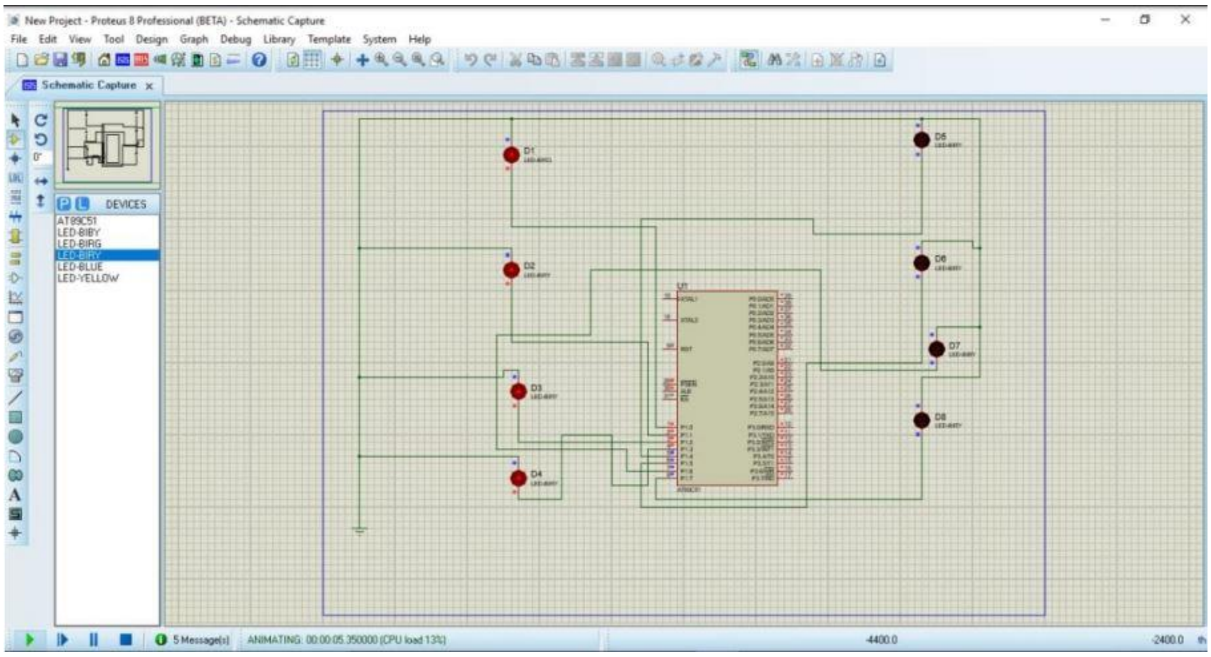
```
Program Size: data=11.0 xdata=0 code=111
creating hex file from "pattern"...
"pattern" - 0 Error(s), 1 Warning(s).
```

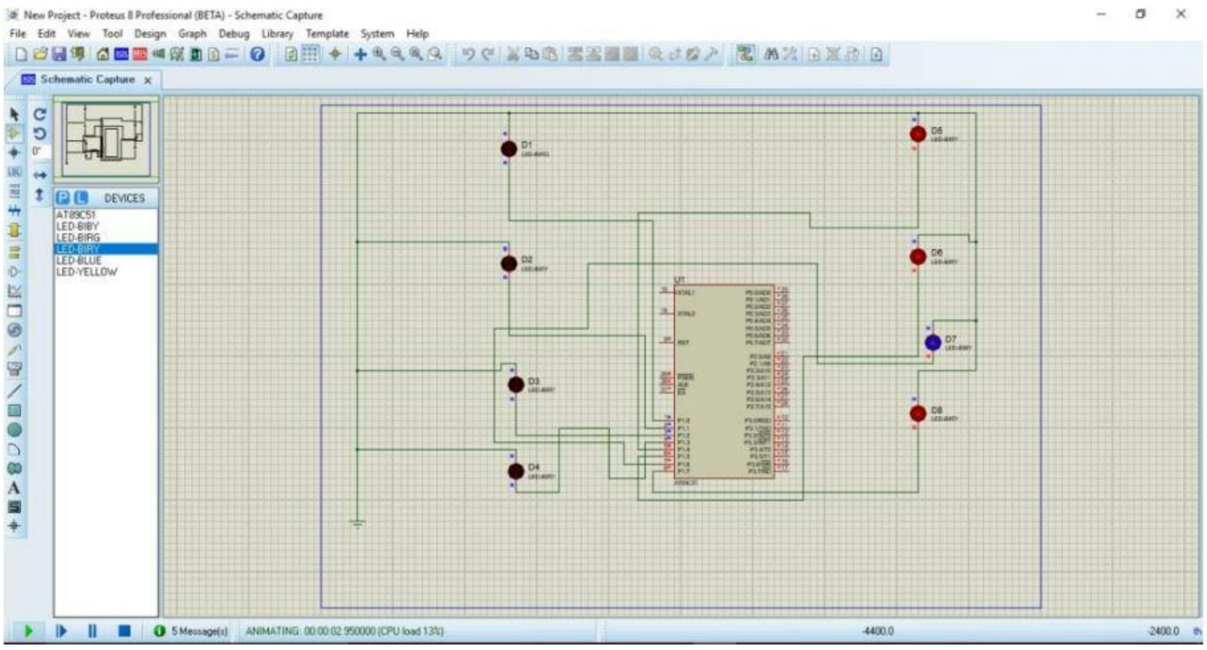

D:\embedded\pattern.uvproj - µVision4
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Target 1
Source Group 1
toggle.c*
Build Output
Simulation
L:35 C:5
CAP NUM SCRL OVR R/W

```

while(1)
{

  toggle=1;
  toggle1=1;
  toggle2=1;
  toggle3=1;
  delay();

  toggle=0;
  toggle1=0;
  toggle2=0;
  toggle3=0;

  toggle4=1;
  toggle5=1;
  toggle6=1;
  toggle7=1;
  delay();

   toggle4=0;
  toggle5=0;
  toggle6=0;
  toggle7=0;
        }
}

```

```
Program Size: data=11.0 xdata=0 code=111
creating hex file from "pattern"...
"pattern" - 0 Error(s), 1 Warning(s).
```

Step 2: Make the following setup.


New Project - Proteus 8 Professional (BETA) - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
AT89C51
LED-BIBY
LED-BIRG
LED-BIRY
LED-BLUE
LED-YELLOW
D1
D2
D3
D4
D5
D6
D7
D8
U1
5 Message(s)
ANIMATING: 00:00:05.350000 (CPU load 13%)
-4400.0
-2400.0


New Project - Proteus 8 Professional (BETA) - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
AT89C51
LED-BIBY
LED-BIRG
LED-BIRY
LED-BLUE
LED-YELLOW
D1
D2
D3
D4
D5
D6
D7
D8
U1
5 Message(s)
ANIMATING: 00:00:02.950000 (CPU load 13%)
-4400.0
-2400.0

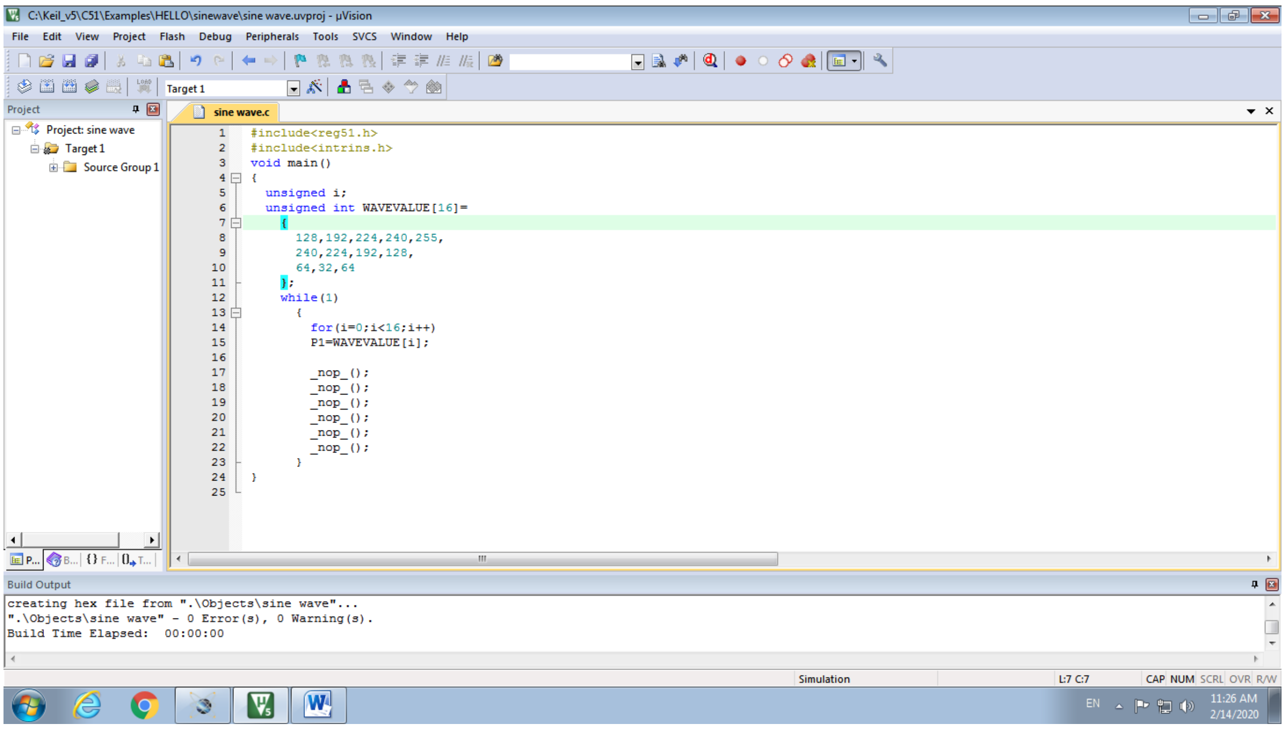
Source Code:

```
#include<reg51.h>
abit toggle = P1^0;
abit toggle = P1^1;
abit toggle = P1^2;
abit toggle = P1^3;
abit toggle = P1^4;
abit toggle = P1^5;
abit toggle = P1^6;
abit toggle = P1^7;
void delay()
{
        unsigned int r;
        for ( r = 0 ; r < 60000 ; r++)
}
void main()
{
        unsigned itn x;
        toggle4 = 0;
        toggle5 = 0;
        toggle6 = 0;
        toggle7 = 0;
        while(1)
        {
                toggle = 1;
                toggle1 = 1;
                toggle2 = 1;
                toggle3 = 1;
                delay();
                toggle = 0;
```

```
			toggle1 = 0;
			toggle2 = 0;
			toggle3 = 0;
			toggle4 = 1;
			toggle5 = 1;
			toggle6 = 1;
			toggle7 = 1;
			delay();
			toggle4 = 0
			toggle5 = 0;
			toggle6 = 0;
			toggle7 = 0;
		}
	}
```

Practical No.: 2 Sine Wave

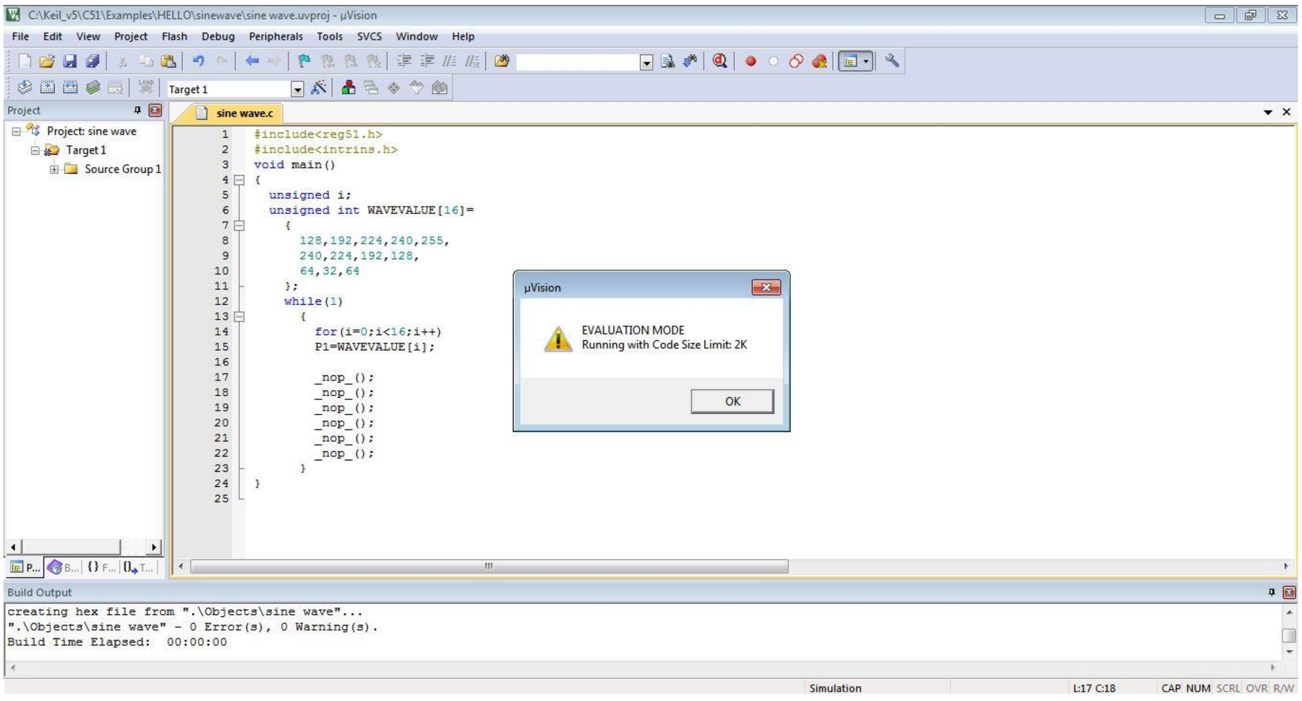
Step 1: Write the C program.


C:\Keil_v5\C51\Examples\HELLO\sinewave\sine wave.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: sine wave
Target 1
Source Group 1
sine wave.c
Build Output
Simulation
L:7 C:7
CAP NUM SCRL OVR R/W

```c
#include<reg51.h>
#include<intrins.h>
void main()
{
  unsigned i;
  unsigned int WAVEVALUE[16]=
    {
      128,192,224,240,255,
      240,224,192,128,
      64,32,64
    };
    while(1)
      {
        for(i=0;i<16;i++)
        P1=WAVEVALUE[i];

        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
      }
}
```

```
creating hex file from ".\Objects\sine wave"...
".\Objects\sine wave" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

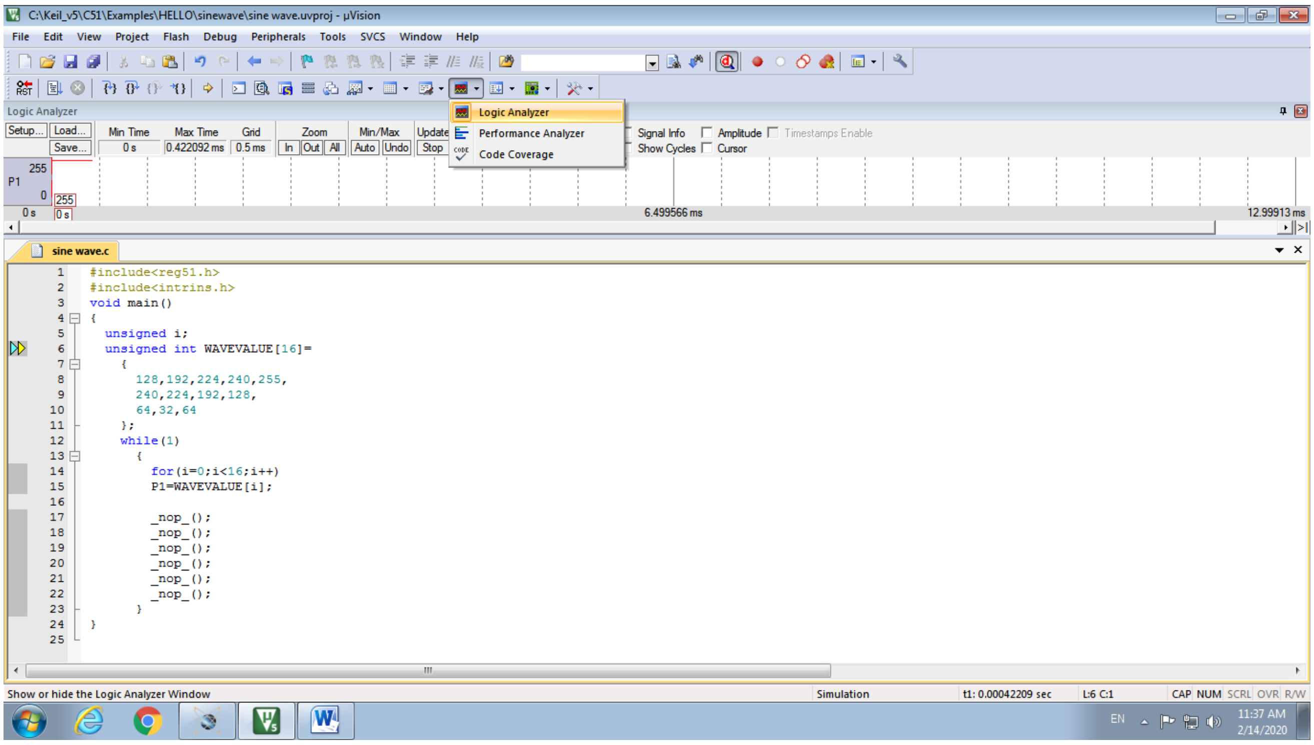
Step 2: Click on Debug.


C:\Keil_v5\C51\Examples\HELLO\sinewave\sine wave.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: sine wave
Target 1
Source Group 1
sine wave.c
µVision
OK
Build Output
Simulation
L:17 C:18
CAP NUM SCRL OVR R/W

EVALUATION MODE
Running with Code Size Limit: 2K

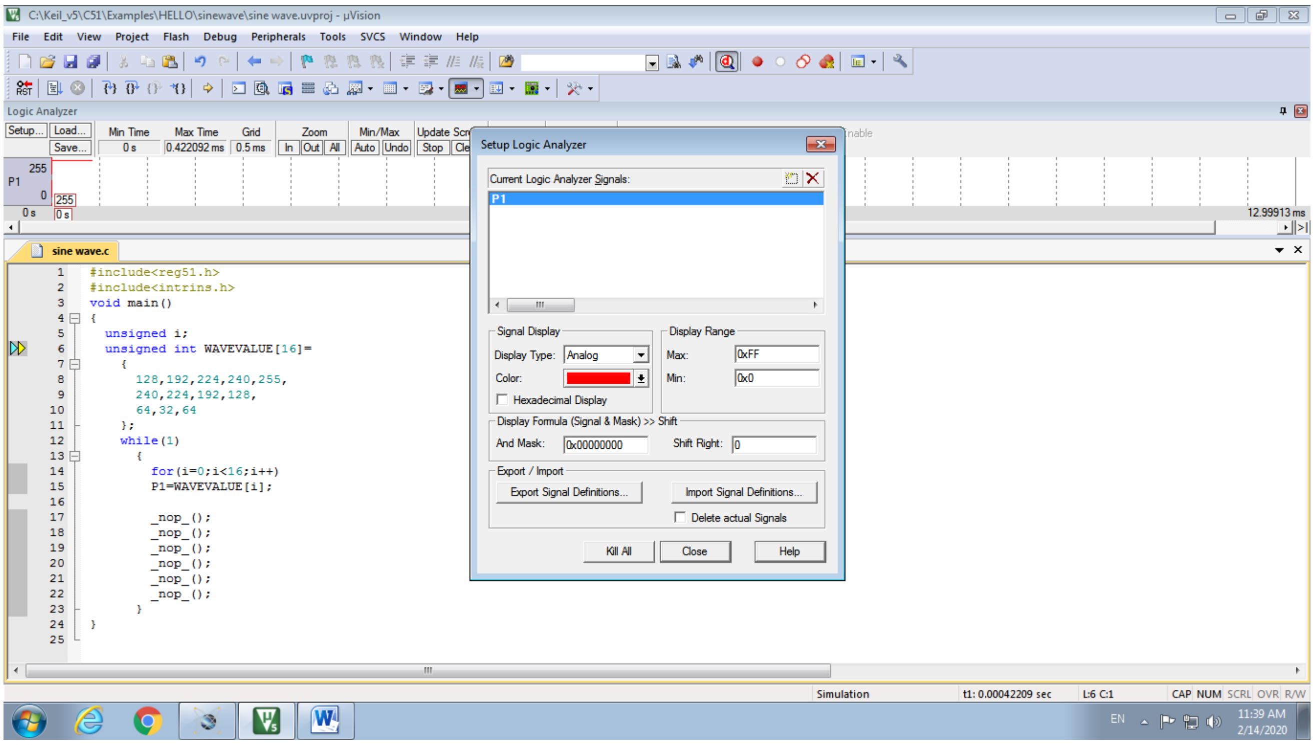
```
creating hex file from ".\Objects\sine wave"...
".\Objects\sine wave" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Step 3: Click on Logic Analyzer:


C:\Keil_v5\C51\Examples\HELLO\sinewave\sine wave.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Logic Analyzer
Performance Analyzer
Code Coverage
Setup... Load... Save... Min Time 0 s Max Time 0.422092 ms Grid 0.5 ms Zoom In Out All Min/Max Auto Undo Update Stop
Signal Info Show Cycles Amplitude Cursor Timestamps Enable
P1 255 0
6.499566 ms
12.99913 ms
sine wave.c
Show or hide the Logic Analyzer Window
Simulation t1: 0.00042209 sec L:6 C:1 CAP NUM SCRL OVR R/W
11:37 AM 2/14/2020

```c
#include<reg51.h>
#include<intrins.h>
void main()
{
  unsigned i;
  unsigned int WAVEVALUE[16]=
    {
      128,192,224,240,255,
      240,224,192,128,
      64,32,64
    };
    while(1)
      {
        for(i=0;i<16;i++)
        P1=WAVEVALUE[i];

        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
        _nop_();
      }
}
```

Step 4: Click on Set up, write P1 and Press Enter.


C:\Keil_v5\C51\Examples\HELLO\sinewave\sine wave.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Logic Analyzer
Setup Logic Analyzer
Export Signal Definitions...
Import Signal Definitions...
Kill All
Close
Help
11:39 AM 2/14/2020

Current Logic Analyzer Signals: P1

Signal Display — Display Type: Analog; Color: 🟥; Hexadecimal Display ☐

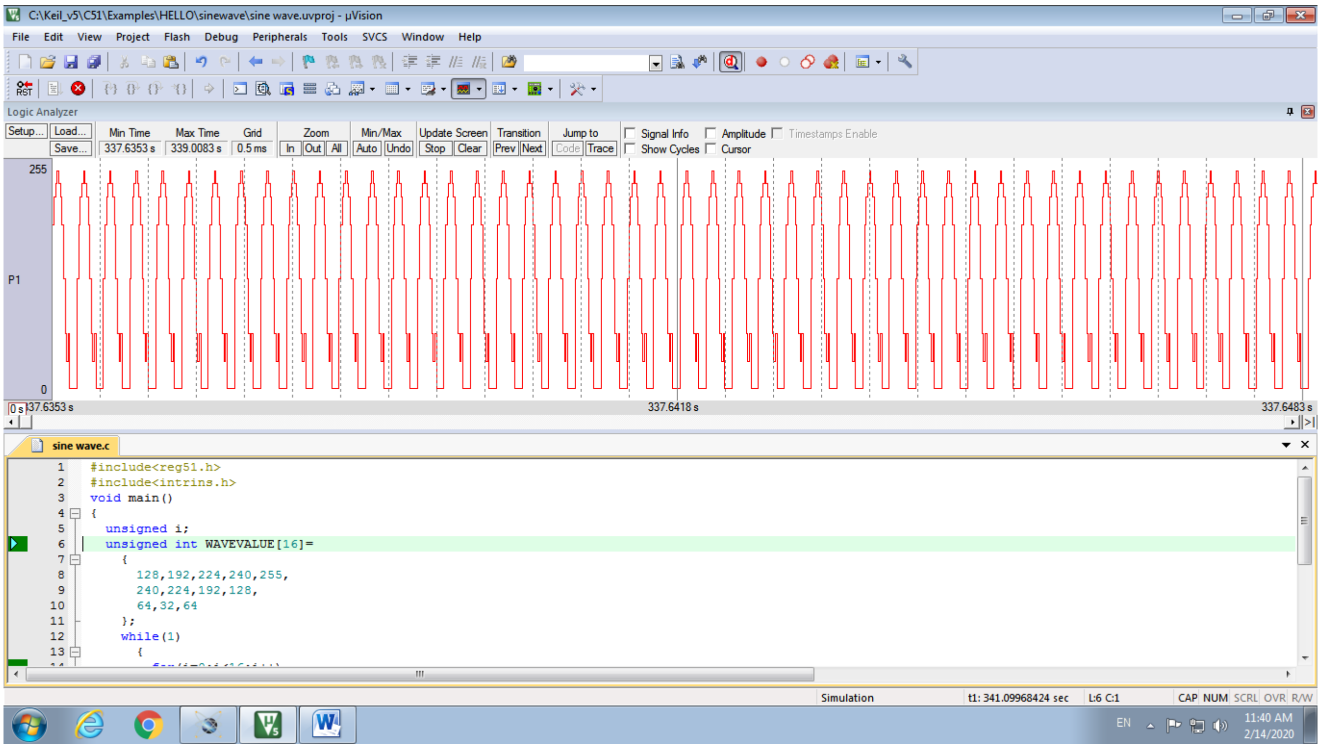
Display Range — Max: 0xFF; Min: 0x0

Display Formula (Signal & Mask) >> Shift — And Mask: 0x00000000; Shift Right: 0

Export / Import — Delete actual Signals ☐

Simulation t1: 0.00042209 sec L:6 C:1

Sine Wave is generated:



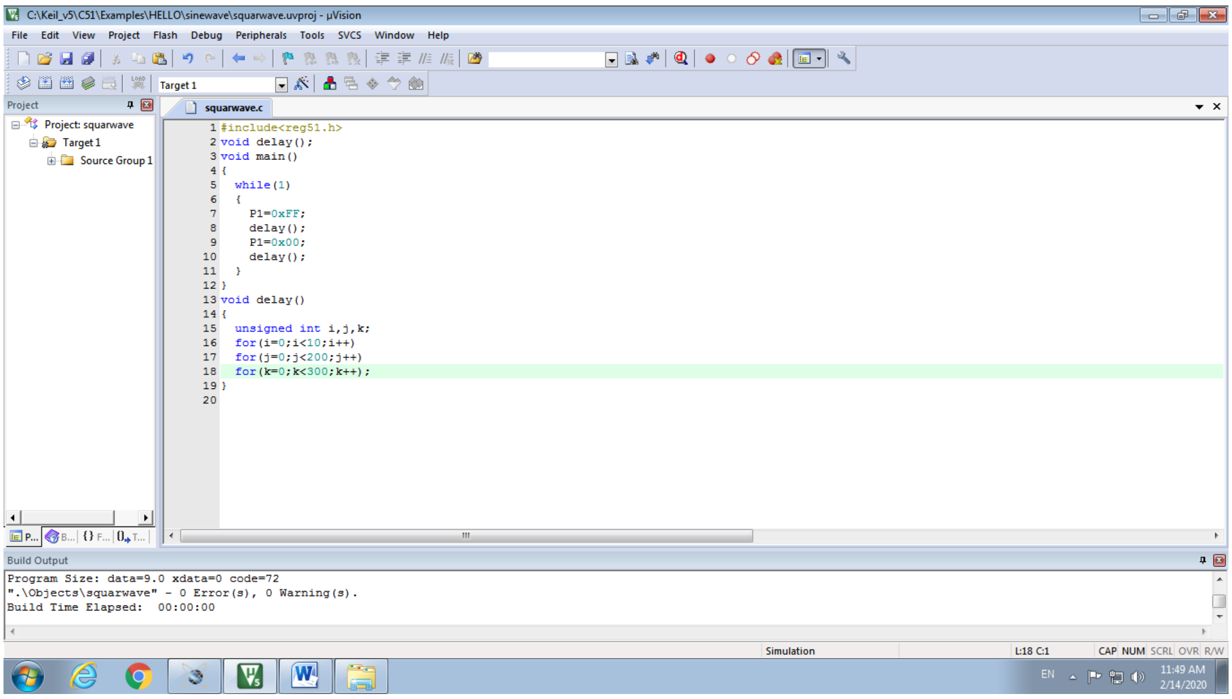
```
#include<reg51.h>
#include<intrins.h>
void main()
{
  unsigned i;
  unsigned int WAVEVALUE[16]=
    {
      128,192,224,240,255,
      240,224,192,128,
      64,32,64
    };
    while(1)
      {
```

Source Code:

```
#include<reg51.h>
 #include<intrins.h>
 void main()
 {
            unsigned i;
            unsigned int WAVEVALUE[16] =
                        {
                                    128,192,224,240,255,
                                    240,224,192,128,
                                    64,32,64
                        };
                        while(1)
                                    {
                                                for(I = 0 ; I < 16 ; i++)
                                                P1 = WAVEVALUE[i];
                                                _nop_();
                                                _nop_();
                                                _nop_();
                                                _nop_();
                                                _nop_();
                                                _nop_();
                                    }
 }
```

Practical No.: 3 Square Wave

Step 1: Write the C program.

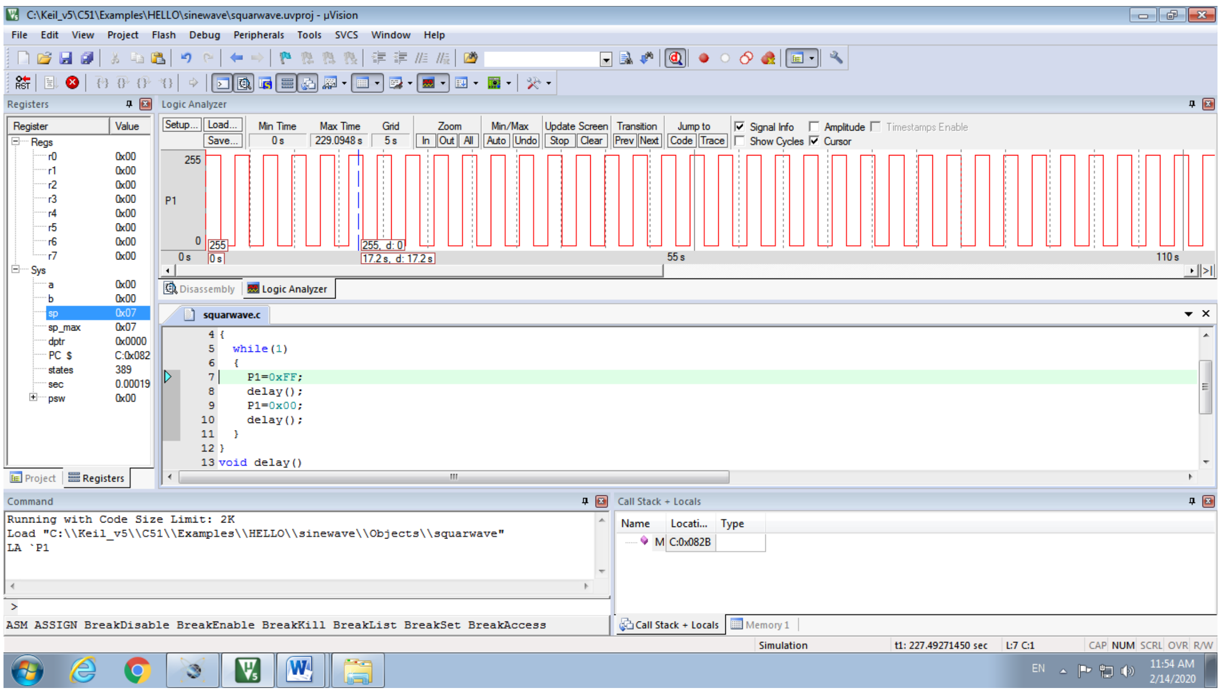


```
#include<reg51.h>
void delay();
void main()
{
  while(1)
  {
    P1=0xFF;
    delay();
    P1=0x00;
    delay();
  }
}
void delay()
{
  unsigned int i,j,k;
  for(i=0;i<10;i++)
  for(j=0;j<200;j++)
  for(k=0;k<300;k++);
}
```

```
Program Size: data=9.0 xdata=0 code=72
".\Objects\squarwave" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

FOLLOW SAME STEP AS DONE IN PRACTICAL 2

Step 2: Use the Debug and Logic Analyzer for Output.



```
Register    Value
Regs
  r0        0x00
  r1        0x00
  r2        0x00
  r3        0x00
  r4        0x00
  r5        0x00
  r6        0x00
  r7        0x00
Sys
  a         0x00
  b         0x00
  sp        0x07
  sp_max    0x07
  dptr      0x0000
  PC $      C:0x082
  states    389
  sec       0.00019
  psw       0x00
```

```
{
  while(1)
  {
    P1=0xFF;
    delay();
    P1=0x00;
    delay();
  }
}
void delay()
```

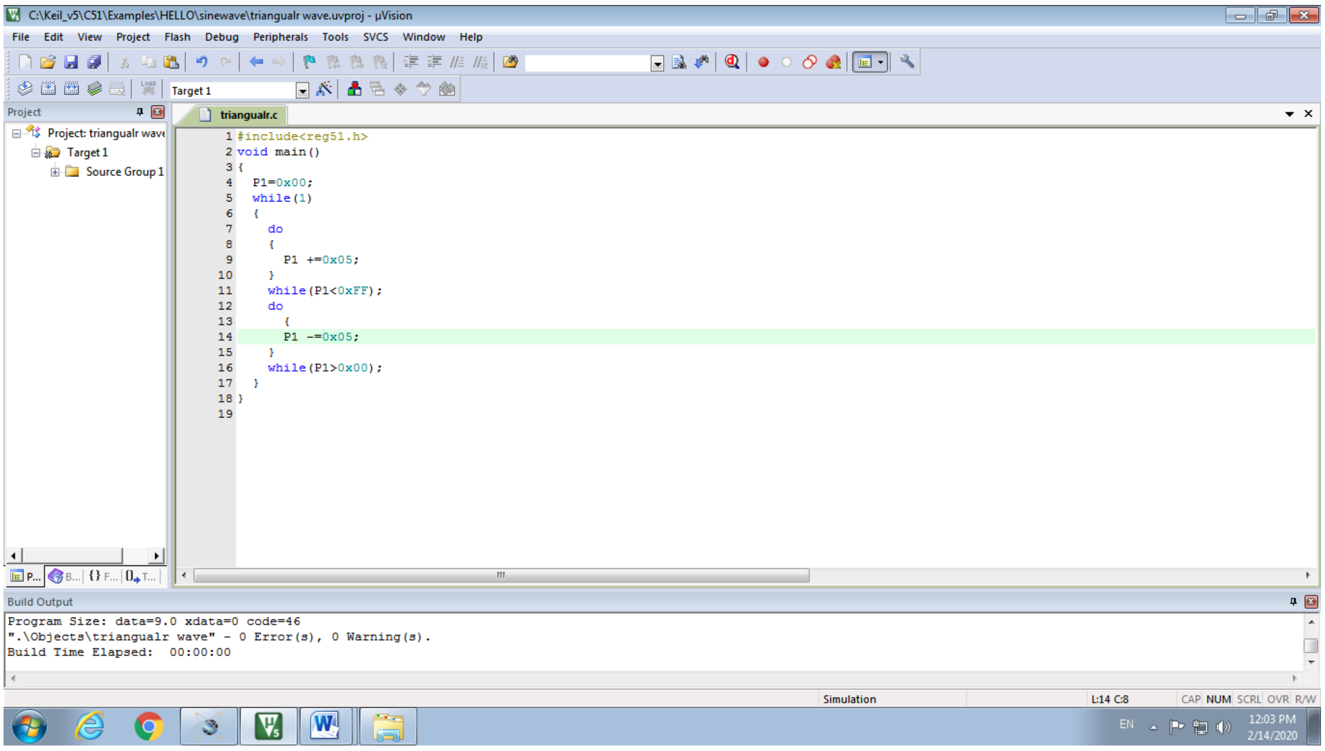
```
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\sinewave\\Objects\\squarwave"
LA `P1
```

Source Code:

```
#include<reg51.h>
void delay();
void main()
{
        while(1)
        {
                P1 = 0xFF;
                delay();
                P1 = 0x00;
                delay();
        }
}
void delay()
{
        unsigned int i, j, k;
        for(i = 0  ; i < 10 ; i++)
        for(j = 0 ; j < 200 ; j++)
        for(k = 0 ; k < 300 ; k++);
}
```

Practical No.: 4 Triangular Wave

Step 1: Write the C program.



C:\Keil_v5\C51\Examples\HELLO\sinewave\triangualr wave.uvproj - µVision

Project: triangualr wave / Target 1 / Source Group 1

triangualr.c

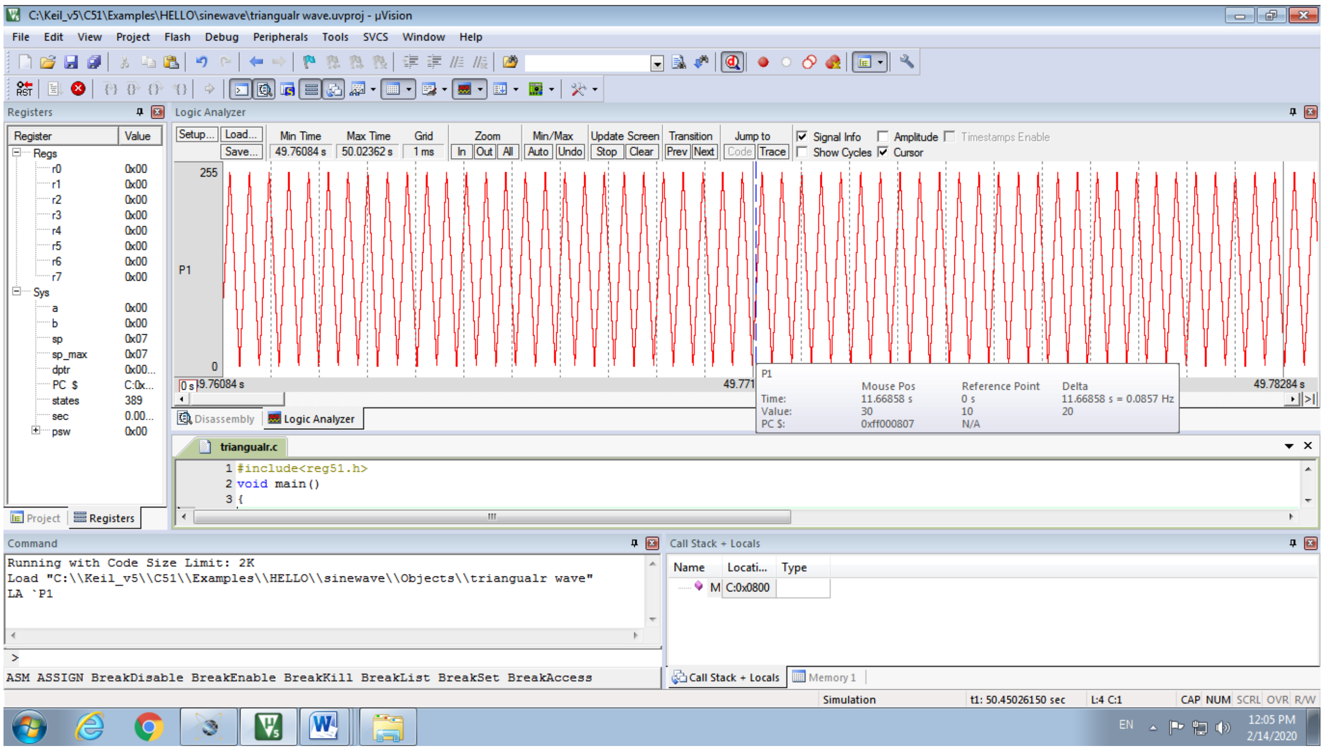
```
#include<reg51.h>
void main()
{
  P1=0x00;
  while(1)
  {
    do
    {
      P1 +=0x05;
    }
    while(P1<0xFF);
    do
      {
      P1 -=0x05;
    }
    while(P1>0x00);
  }
}
```

Build Output

```
Program Size: data=9.0 xdata=0 code=46
".\Objects\triangualr wave" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Simulation   L:14 C:8

Step 2: Use the Debug and Logic Analyzer for Output.



C:\Keil_v5\C51\Examples\HELLO\sinewave\triangualr wave.uvproj - µVision

Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x07
sp_max	0x07
dptr	0x00...
PC $	C:0x...
states	389
sec	0.00...
psw	0x00

Logic Analyzer

Min Time 49.76084 s   Max Time 50.02362 s   Grid 1 ms

✓ Signal Info   Amplitude   Timestamps Enable   Show Cycles   ✓ Cursor

P1: 255 / 0

0 s 49.76084 s   49.771   49.78284 s

P1	Mouse Pos	Reference Point	Delta
Time:	11.66858 s	0 s	11.66858 s = 0.0857 Hz
Value:	30	10	20
PC $:	0xff000807	N/A	

Command

```
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\HELLO\\sinewave\\Objects\\triangualr wave"
LA `P1
```

```
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Call Stack + Locals

Name	Locati...	Type
M	C:0x0800	

Simulation   t1: 50.45026150 sec   L:4 C:1

Source Code:

```
#include<reg51.h>
void main()
{
	P1 = 0x00;
	while(1)
	{
		do
		{
			P1 += 0x05;
		}
		while(P1 < 0xFF);
		do
			{
			P1 -= 0x05;
		}
		while(P1 > 0x00);
	}
}
```

Practical No.: 5 Delay Timer Using 8051 Microcontroller

Delay using 8051 timer. The 8051 microcontroller has two independents 16-bit up counting timers named Timer 0 and Timer 1 and this article is about generating time delays using the 8051 timers. Generating delay using pure software loops have been already discussed here but such delays are poor in accuracy and cannot be used in sensitive applications. Delay using timer is the most accurate and surely the best method.

A timer can be generalized as a multi-bit counter which increments/decrements itself on receiving a clock signal and produces an interrupt signal up on roll over. When the counter is running on the processor's clock, it is called a "Timer", which counts a predefined number of processor clock pulses and generates a programmable delay. When the counter is running on an external clock source (may be a periodic or a periodic external signal) it is called a "Counter" itself and it can be used for counting external events.

In 8051, the oscillator output is divided by 12 using a divide by 12 network and then fed to the Timer as the clock signal. That means for an 8051 running at 12MHz, the timer clock input will be 1MHz. That means the timer advances once in every 1uS and the maximum time delay possible using a single 8051 timer is (2^16) x (1µS) = 65536µS. Delays longer than this can be implemented by writing up a basic delay program using timer and then looping it for a required number of times.

Designing a delay program using 8051 timers While designing delay programs in 8051, calculating the initial value that has to be loaded into TH and TL registers forms a very important thing. Let us see how it is done.

• Assume the processor is clocked by a 12MHz crystal.

• That means, the timer clock input will be 12MHz/12 = 1MHz

• That means, the time taken for the timer to make one increment = 1/1MHz = 1uS

• For a time delay of “X” the timer has to make “X” increments.

• 2^16 = 65536 is the maximum number of counts possible for a 16-bit timer.

• Let TH be the value value that has to be loaded to TH register and TL be the value that has to be loaded to TL register.

• Then, THTL = Hexadecimal equivalent of (65536-X) where (65536-X) is considered in decimal. While designing delay programs in 8051, calculating the initial value that has to be loaded into TH and TL registers forms a very important thing. Let us see how it is done.

Example:

Let the required delay be 1000uS (i.e.; 1mS).

That means X = 1000.

65536 – X = 65536 – 1000 = 64536.

64536 is considered in decimal and converting it to hexadecimal gives FC18.

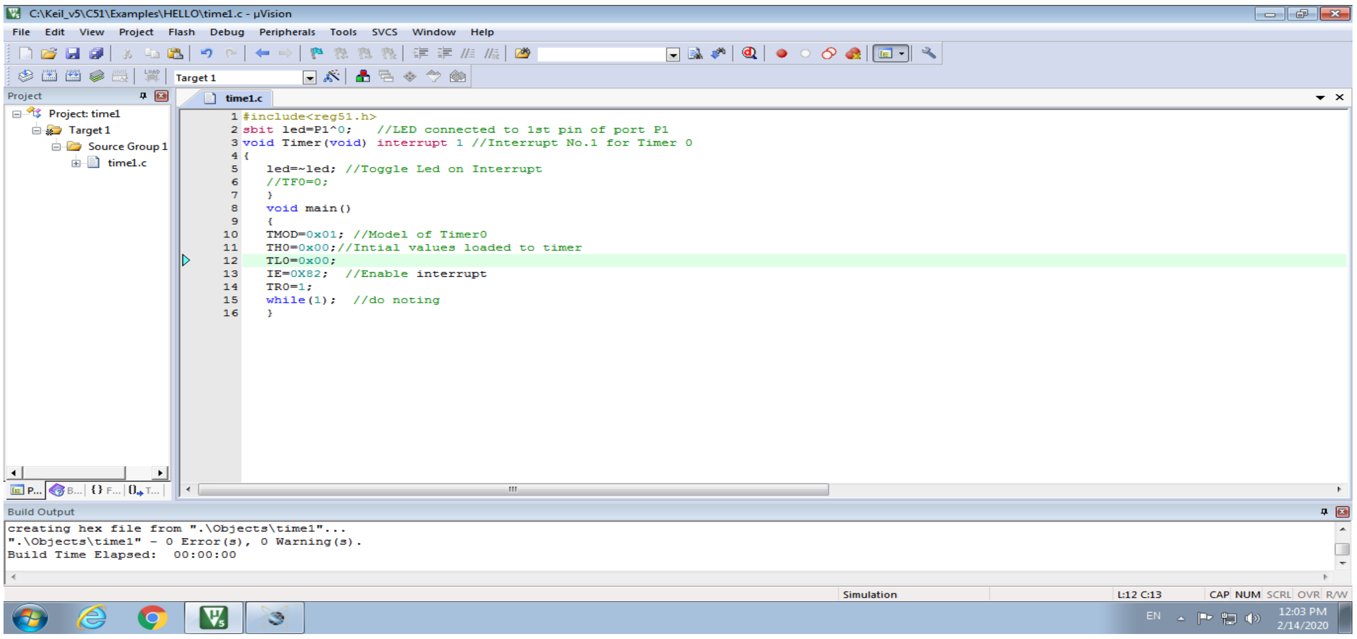
That means THTL = FC18.

Therefore TH=FC and TL=18.

Program for generating 1mS delay using 8051 timer.

The program shown below can be used for generating 1mS delay and it is written as a subroutine so that you can call it anywhere in the program. Also, you can put this in a loop for creating longer time delays (multiples of 1mS). Here Timer 0 of 8051 is used and it is operating in MODE1 (16-bit timer).

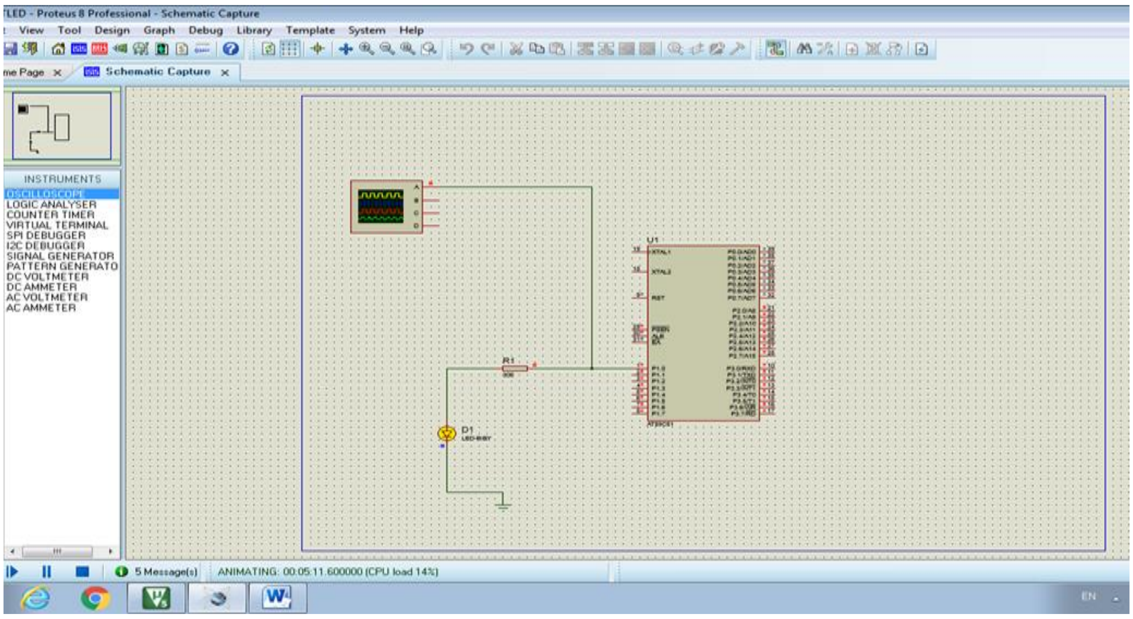
Step 1: Write the C program.

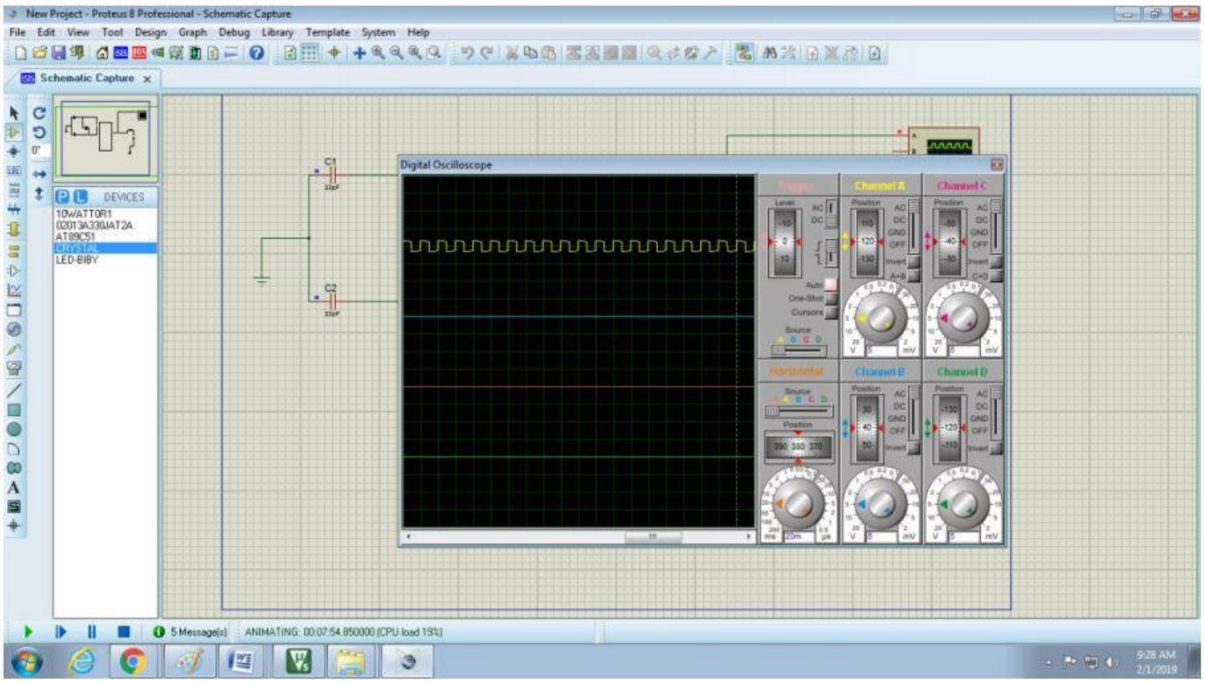

C:\Keil_v5\C51\Examples\HELLO\time1.c - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: time1
Target 1
Source Group 1
time1.c
Build Output
Simulation
L:12 C:13
CAP NUM SCRL OVR R/W

```
#include<reg51.h>
sbit led=P1^0;   //LED connected to 1st pin of port P1
void Timer(void) interrupt 1 //Interrupt No.1 for Timer 0
{
   led=~led; //Toggle Led on Interrupt
   //TF0=0;
   }
   void main()
   {
   TMOD=0x01; //Model of Timer0
   TH0=0x00;//Intial values loaded to timer
   TL0=0x00;
   IE=0X82;  //Enable interrupt
   TR0=1;
   while(1);  //do noting
   }
```

```
creating hex file from ".\Objects\time1"...
".\Objects\time1" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:   00:00:00
```

Step 2: Make the following setup.



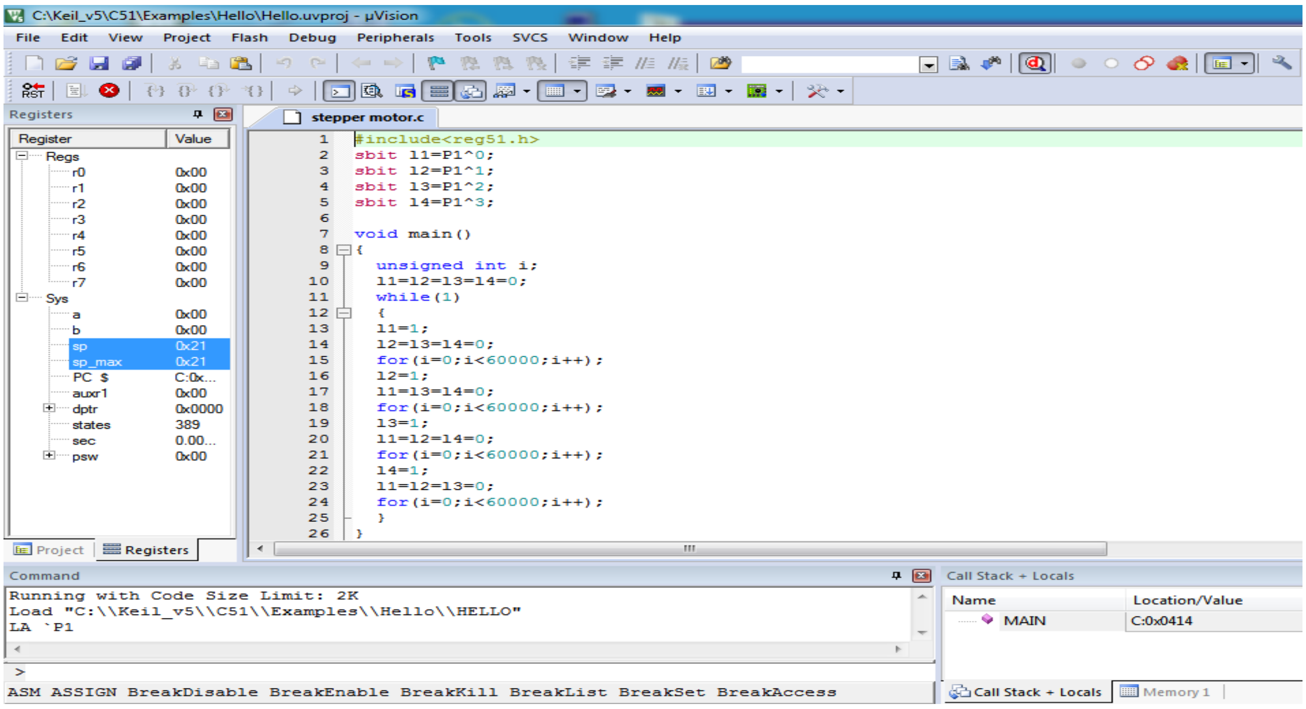

New Project - Proteus 8 Professional - Schematic Capture
Digital Oscilloscope
DEVICES
10WATT0R1
02013A330JAT2A
AT89C51
CRYSTAL
LED-BIBY
5 Message(s)
ANIMATING: 00:07:54.850000 (CPU load 19%)

Source CODE:

```
#include<reg51.h>
sbit led = P1^0;   //LED connected to 1st pin of port P1
void Timer(void) interrupt 1 //Interrupt No.1 for Timer 0
{
        led =~ led;  //Toggle Led on Interrupt
         //TF0 = 0;
}
void main()
{
        TMOD = 0x01;  //Model of Timer0
        TH0 = 0x00; //Initial values loaded to timer
        TL0 = 0x00;
        IE = 0X82;  //Enable interrupt
        TR0 = 1;
        while(1);  //do noting
}
```

Practical No.: 6 Stepper Motor

STEP 1: Write the C Program.


C:\Keil_v5\C51\Examples\Hello\Hello.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
stepper motor.c
Project
Registers
Command
Call Stack + Locals
Memory 1

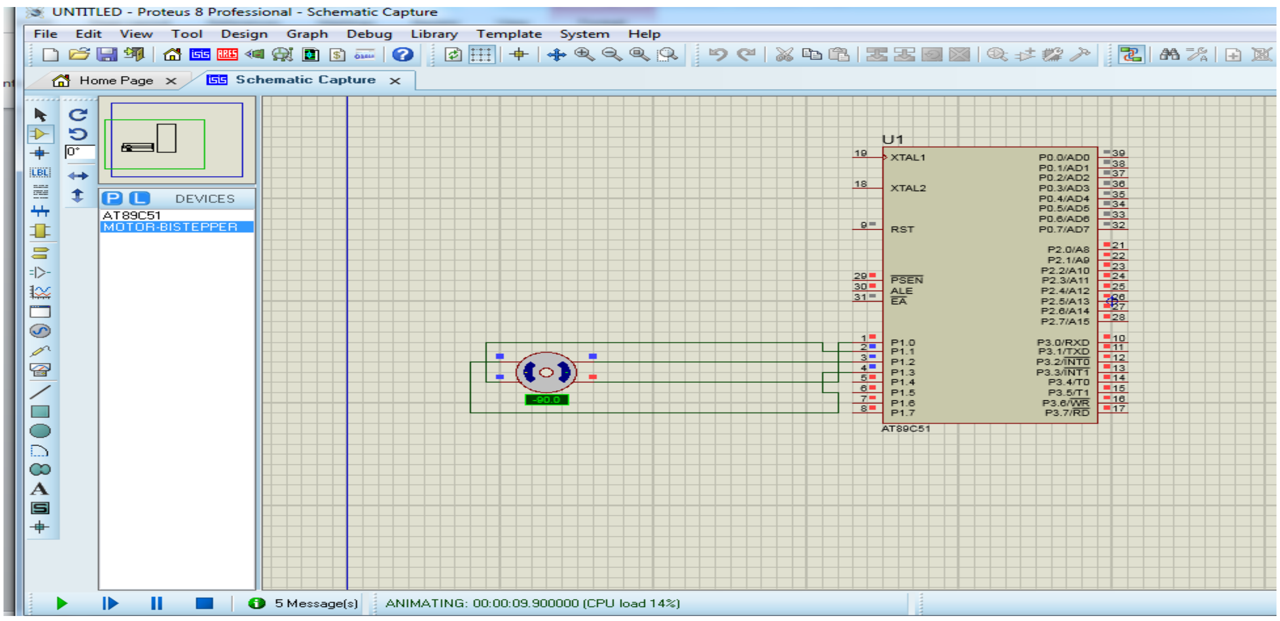
Register	Value
Regs	
r0	0x00
r1	0x00
r2	0x00
r3	0x00
r4	0x00
r5	0x00
r6	0x00
r7	0x00
Sys	
a	0x00
b	0x00
sp	0x21
sp_max	0x21
PC $	C:0x...
auxr1	0x00
dptr	0x0000
states	389
sec	0.00...
psw	0x00

```c
#include<reg51.h>
sbit l1=P1^0;
sbit l2=P1^1;
sbit l3=P1^2;
sbit l4=P1^3;

void main()
{
  unsigned int i;
  l1=l2=l3=l4=0;
  while(1)
  {
  l1=1;
  l2=l3=l4=0;
  for(i=0;i<60000;i++);
  l2=1;
  l1=l3=l4=0;
  for(i=0;i<60000;i++);
  l3=1;
  l1=l2=l4=0;
  for(i=0;i<60000;i++);
  l4=1;
  l1=l2=l3=0;
  for(i=0;i<60000;i++);
  }
}
```

```
Running with Code Size Limit: 2K
Load "C:\\Keil_v5\\C51\\Examples\\Hello\\HELLO"
LA `P1
>
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet BreakAccess
```

Name	Location/Value
MAIN	C:0x0414

Step 2: Make the following setup.

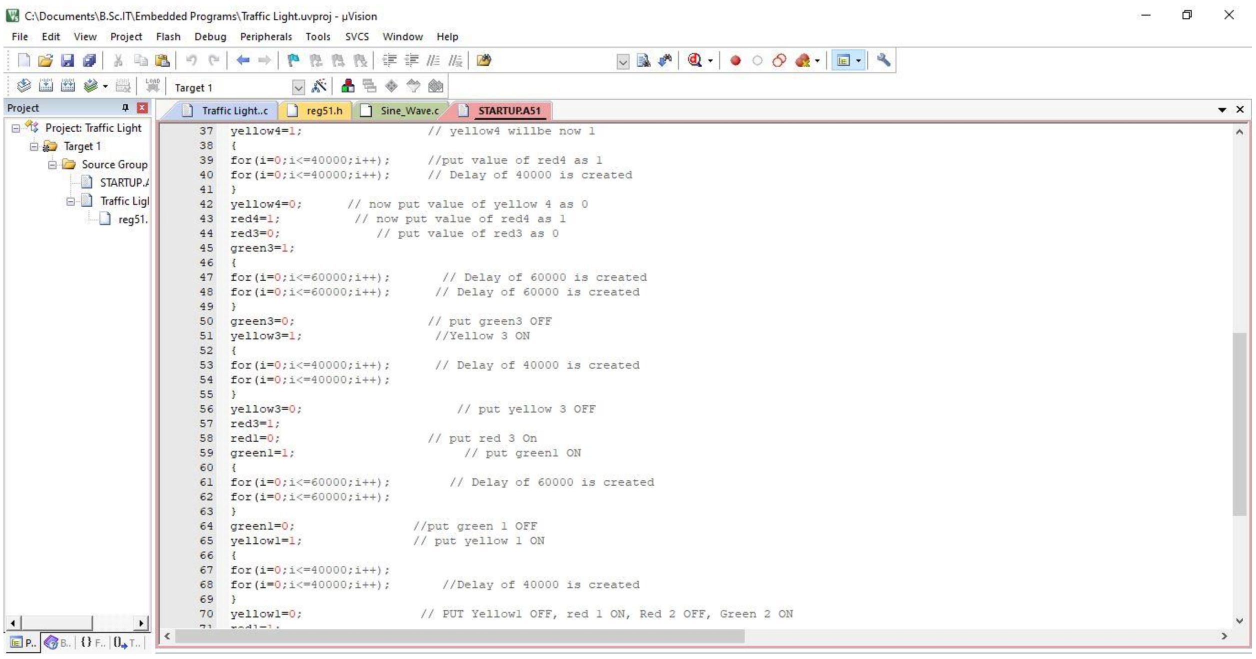


Source Code:

```
#include<reg51.h>
abit l1 = P1^0;
abit l2 = P1^1;
abit l3 = P1^2;
abit l4 = P1^3;
void main()
{
        unsigned int i;
        l1 = l2 = l3 = l4 = 0;
        while(1)
        {
                l1 = 1;
                l2 = l3 = l4 = 0;
                for(i = 0 ; i < 30000 ; i++)
                l2 = 0;
                l1 = l3 = l4 = 1;
                for(i = 0 ; i < 30000 ; i++)
                l3 = 1;
                l1 = l2 = l4 = 0;
                for(i = 0 ; i < 30000 ; i++)
                l4 = 0;
                l1 = l2 = l3 = 1;
                for(i = 0 ; i < 30000 ; i++)
        }
}
```

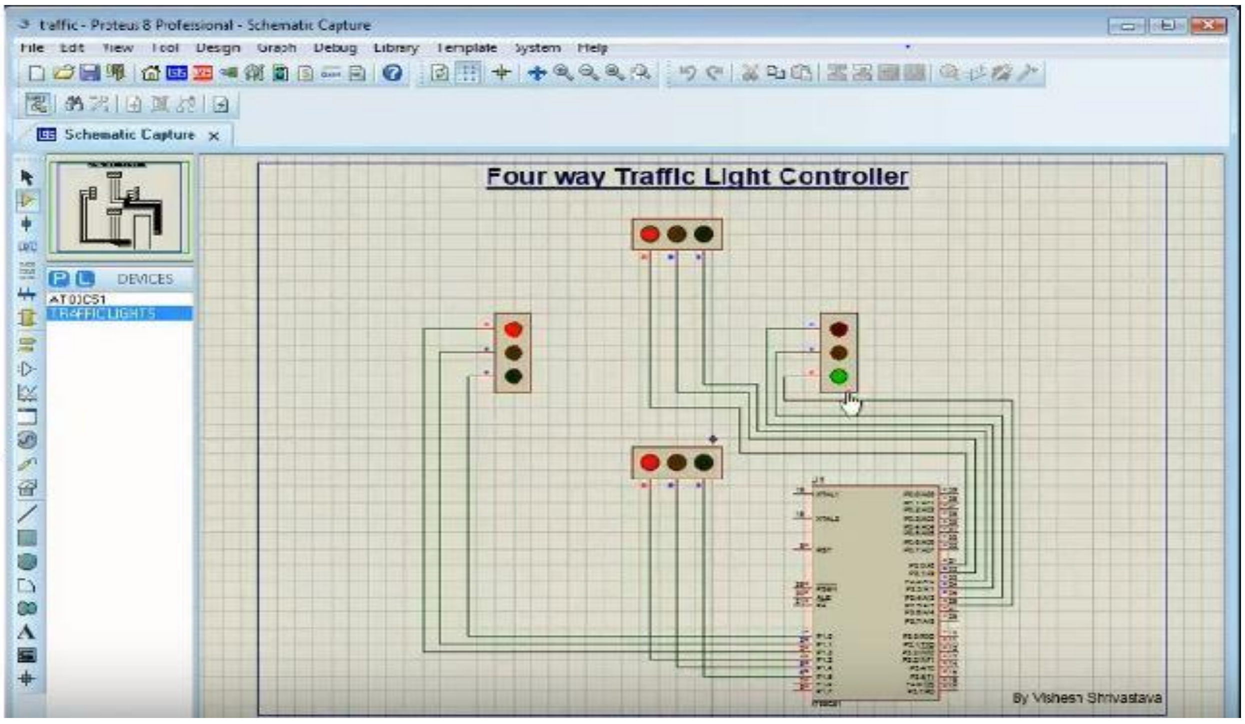
Practical No.: 7 4 Way Traffic LED

Step 1: Write a C program.


C:\Documents\B.Sc.IT\Embedded Programs\Traffic Light.uvproj - µVision

```
yellow4=1;                    // yellow4 willbe now 1
{
for(i=0;i<=40000;i++);        //put value of red4 as 1
for(i=0;i<=40000;i++);        // Delay of 40000 is created
}
yellow4=0;          // now put value of yellow 4 as 0
red4=1;              // now put value of red4 as 1
red3=0;                 // put value of red3 as 0
green3=1;
{
for(i=0;i<=60000;i++);          // Delay of 60000 is created
for(i=0;i<=60000;i++);         // Delay of 60000 is created
}
green3=0;                     // put green3 OFF
yellow3=1;                     //Yellow 3 ON
{
for(i=0;i<=40000;i++);         // Delay of 40000 is created
for(i=0;i<=40000;i++);
}
yellow3=0;                        // put yellow 3 OFF
red3=1;
red1=0;                       // put red 3 On
green1=1;                          // put green1 ON
{
for(i=0;i<=60000;i++);           // Delay of 60000 is created
for(i=0;i<=60000;i++);
}
green1=0;                   //put green 1 OFF
yellow1=1;                  // put yellow 1 ON
{
for(i=0;i<=40000;i++);
for(i=0;i<=40000;i++);          //Delay of 40000 is created
}
yellow1=0;                   // PUT Yellow1 OFF, red 1 ON, Red 2 OFF, Green 2 ON
```

Step 2: Make the following setup.

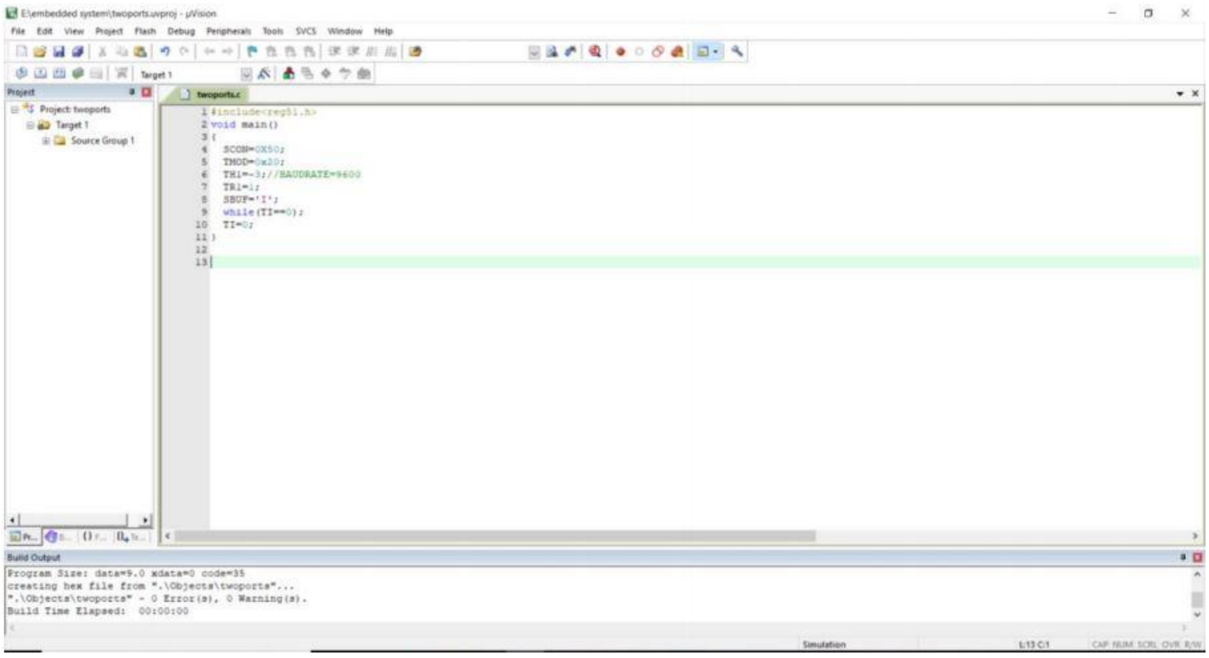

traffic - Proteus 8 Professional - Schematic Capture
Schematic Capture
DEVICES
AT89C51
TRAFFIC LIGHTS
Four way Traffic Light Controller
By Vishesh Shrivastava

Source Code:

```
#include<reg51.h>
sbit red1 = P1^2;
sbit yellow1 = P1^1;
sbit green1 = P1^0;
sbit red2 = P1^3;
sbit yellow2 = P1^4;
sbit green2 = P1^5;
sbit red3 = P2^0;
sbit yellow3 = P2^1;
sbit green3 = P2^2;
sbit red4 = P2^3;
sbit yellow4 = P2^4;
sbit green4 = P2^5;

void main(void)
{

        unsigned int i;
        red1 = yellow1 = green1 = 0;
        red2 = yellow2 = green2 = 0;
        red3 = yellow3 = green3 = 0;
        red4 = yellow4 = green4 = 0;
        while(1)
        {
                red1 = 1;
                red2 = 1;
                red3 = 1;
                green4 = 1;
                yellow4 = 0;
                red4 = 0;
        {
                for(i = 0 ; i <= 60000 ; i++);
                for(i = 0 ; i <= 60000 ; i++);
                unsigned int;
        }
        green4 = 0;
        yellow4 = 1;
        {
                for(i = 0 ; i <= 40000 ; i++);
                for(i = 0 ; i <= 40000 ; i++);
        }
        yellow4 = 0;
        red4 = 1;
        red3 = 0;
        green3 = 1;
        {
                for(i = 0 ; i <= 60000 ; i++);
                for(i = 0 ; i <= 60000 ; i++);
        }
green3 = 0;
yellow3 = 1;
{
            for(I = 0 ; i <= 40000 ; i++);
            for(I = 0 ; i <= 40000 ; i++);
}
yellow3 = 0;
red3 = 1;
red1 = 0;
green1 = 1;
{
            for(I = 0 ; i <= 60000 ; i++);
            for(I = 0 ; i <= 60000 ; i++);
}
green1 = 0;
yellow1 = 1;
{
            for(I = 0 ; i <= 40000 ; i++);
            for(I = 0 ; i <= 40000 ; i++);
}
yellow1 = 0;
red1 = 1;
red2 = 0;
green2 = 1;
{
            for(I = 0 ; i <= 60000 ; i++);
            for(I = 0 ; i <= 60000 ; i++);
}
green2 = 0;
yellow2 = 1;
{
            for(I = 0 ; i <= 40000 ; i++);
            for(I = 0 ; i <= 40000 ; i++);
}
yellow2 = 0;
red2 = 1;

}
}
```

PRACTICAL 8: Serial communication between 8051

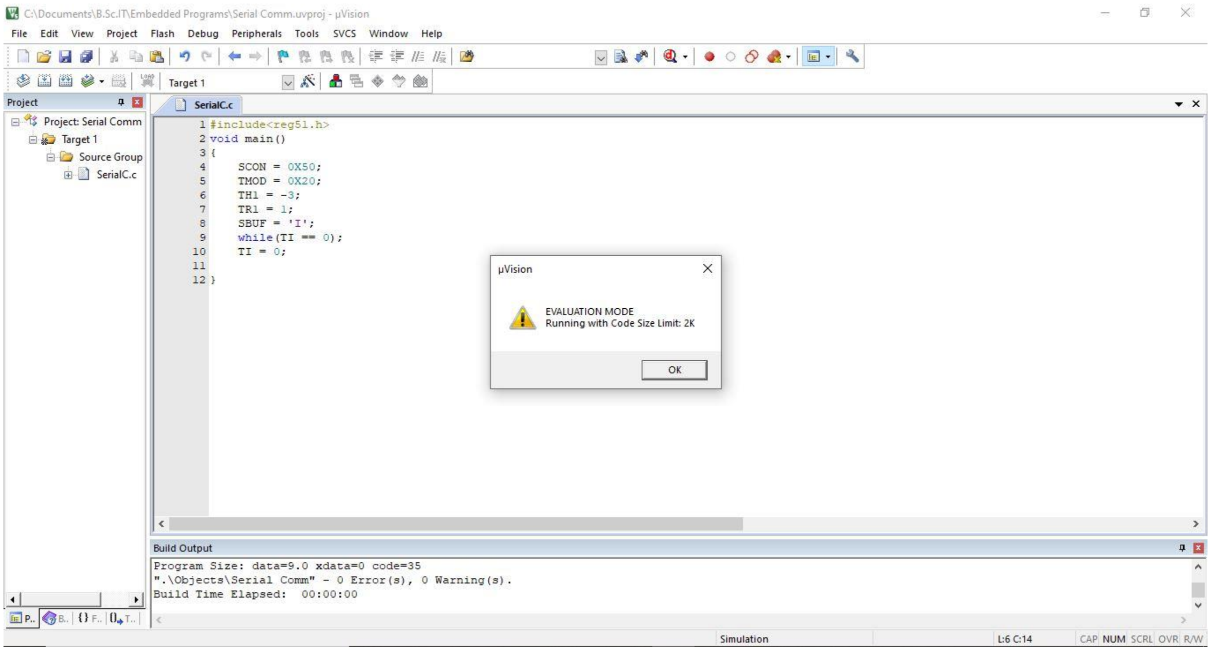
Step 1: Write the C program.



```c
#include<reg51.h>
void main()
{
  SCON=0X50;
  TMOD=0x20;
  TH1=-3;//BAUDRATE=9600
  TR1=1;
  SBUF='I';
  while(TI==0);
  TI=0;
}
```

```
Program Size: data=9.0 xdata=0 code=35
creating hex file from ".\Objects\twoports"...
".\Objects\twoports" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Step 2: Click on Debug.



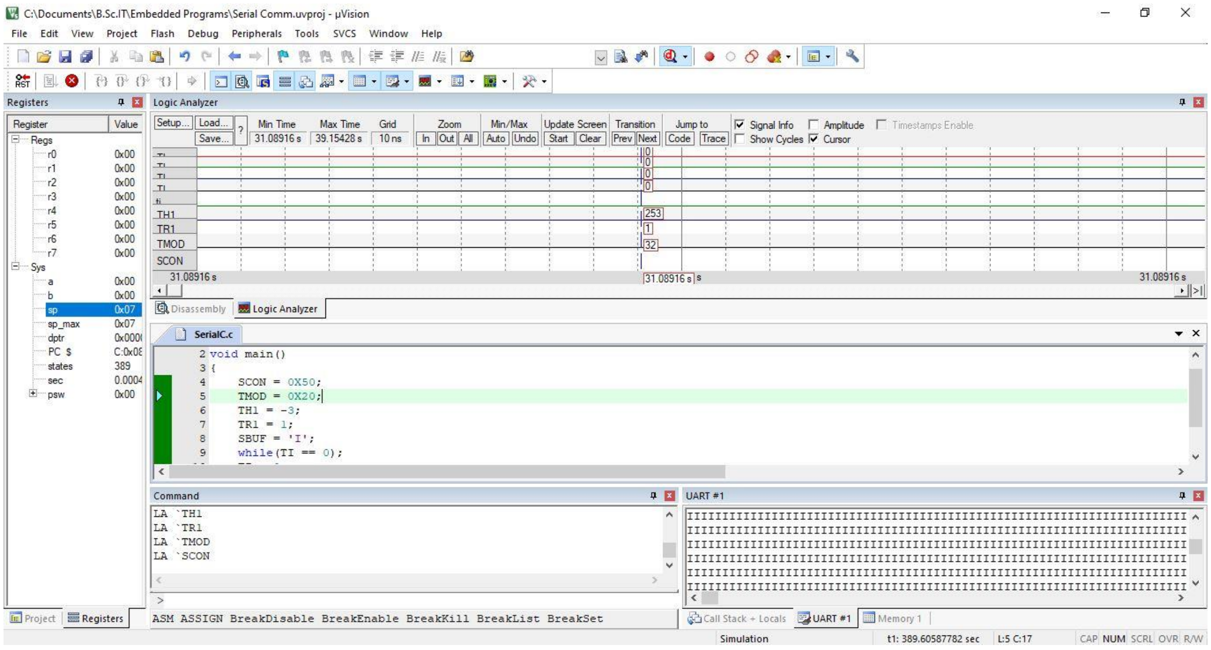
C:\Documents\B.Sc.IT\Embedded Programs\Serial Comm.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Target 1
Project
Project: Serial Comm
Target 1
Source Group
SerialC.c
SerialC.c
µVision
OK
Build Output
Simulation
L:6 C:14
CAP NUM SCRL OVR R/W

```c
#include<reg51.h>
void main()
{
    SCON = 0X50;
    TMOD = 0X20;
    TH1 = -3;
    TR1 = 1;
    SBUF = 'I';
    while(TI == 0);
    TI = 0;

}
```

EVALUATION MODE
Running with Code Size Limit: 2K

```
Program Size: data=9.0 xdata=0 code=35
".\Objects\Serial Comm" - 0 Error(s), 0 Warning(s).
Build Time Elapsed:  00:00:00
```

Step 3: Click on Serial Windows icon and run the program.


C:\Documents\B.Sc.IT\Embedded Programs\Serial Comm.uvproj - µVision
File Edit View Project Flash Debug Peripherals Tools SVCS Window Help
Registers
Logic Analyzer
SerialC.c
Command
UART #1

```
void main()
{
    SCON = 0X50;
    TMOD = 0X20;
    TH1 = -3;
    TR1 = 1;
    SBUF = 'I';
    while(TI == 0);
```

```
LA `TH1
LA `TR1
LA `TMOD
LA `SCON
```

```
ASM ASSIGN BreakDisable BreakEnable BreakKill BreakList BreakSet
```

```
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII
```

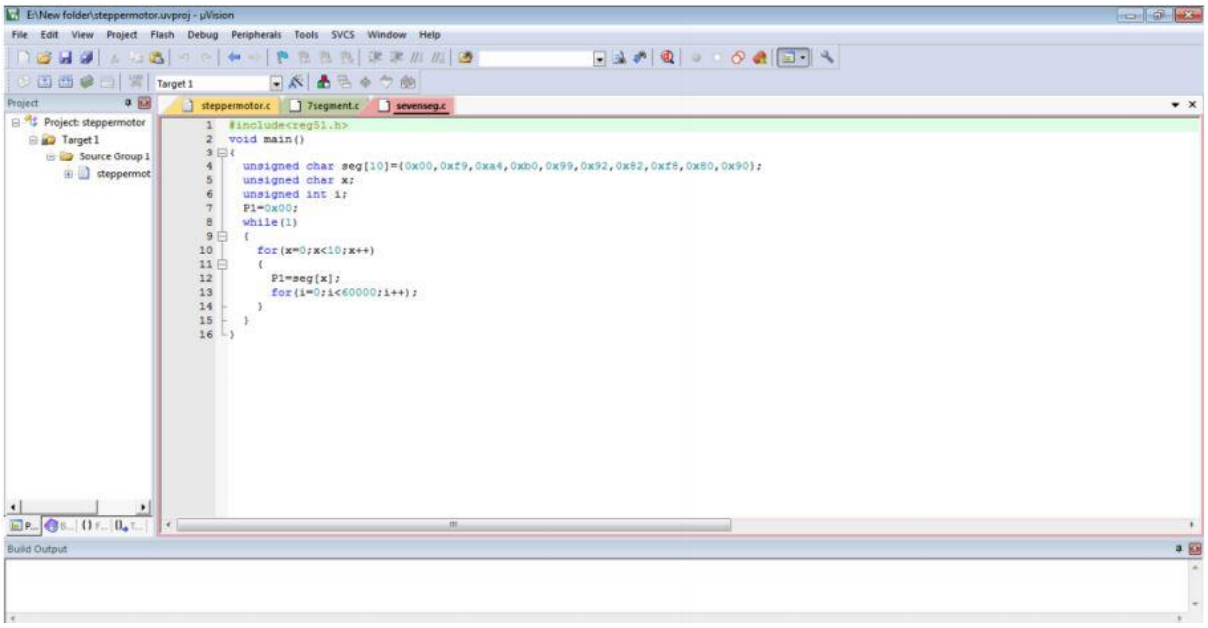
Simulation t1: 389.60587782 sec L:5 C:17

Source Code:

```
#include<reg51.h>
void main()
{
            SCON = 0X50;
            TMOD = 0X20;
            TH1 = -3;
            TR1 = 1;
            SBUF = 'I';
            while(TI == 0);
            TI = 0;

}
```

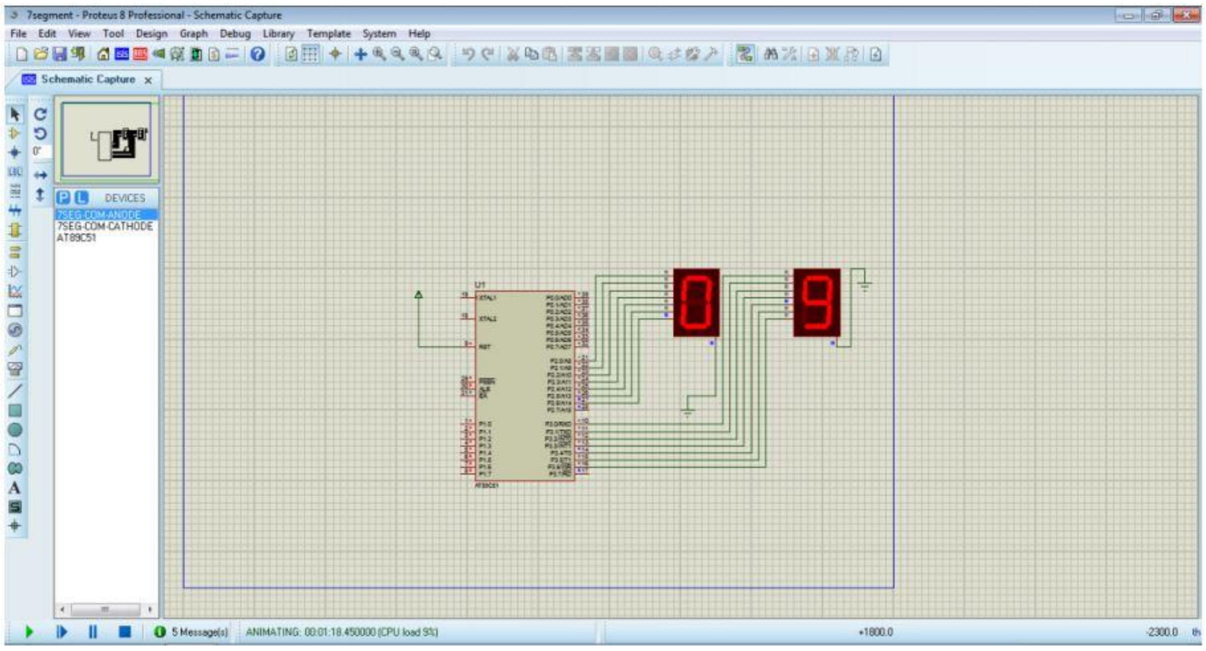
Practical No.: 9 Seven Segment LED

Step 1: Write the C program for One LED Project.

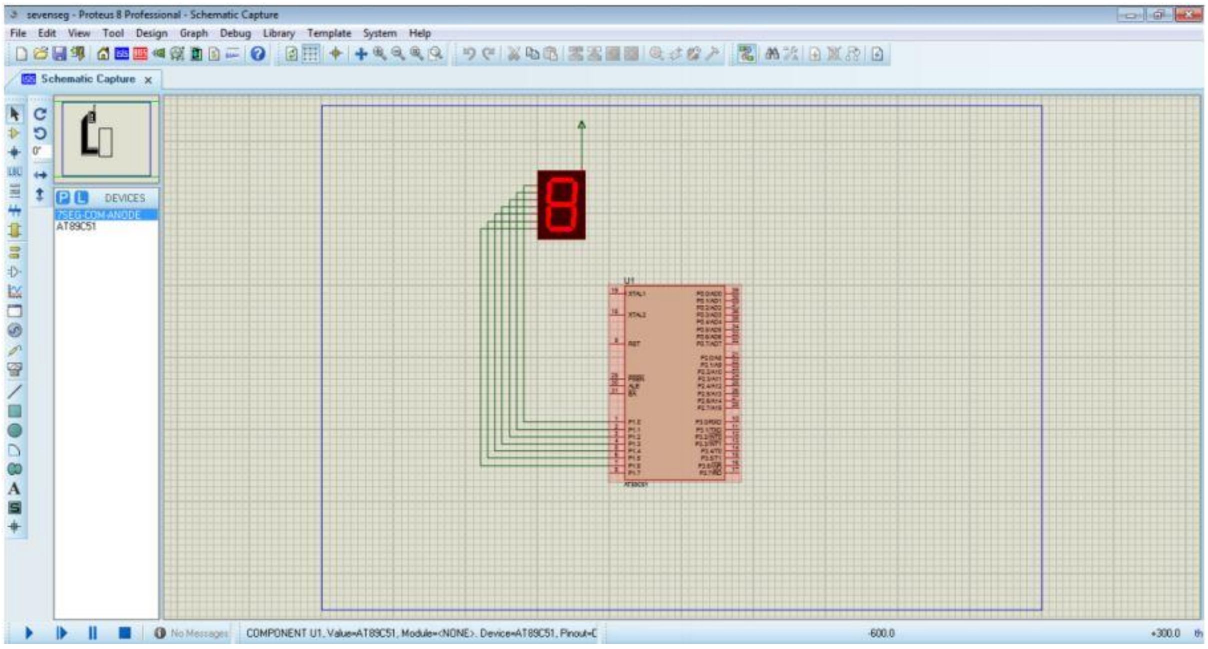


```
#include<reg51.h>
void main()
{
  unsigned char seg[10]={0x00,0xf9,0xa4,0xb0,0x99,0x92,0x82,0xf8,0x80,0x90};
  unsigned char x;
  unsigned int i;
  P1=0x00;
  while(1)
  {
    for(x=0;x<10;x++)
    {
      P1=seg[x];
      for(i=0;i<60000;i++);
    }
  }
}
```

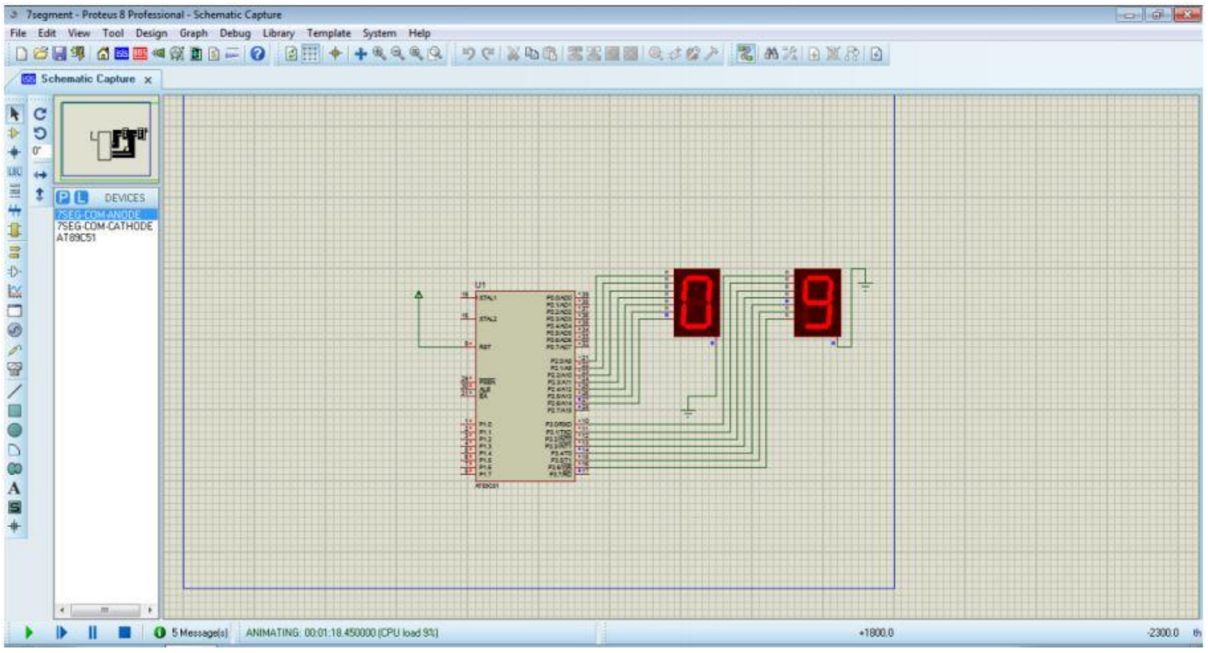
C program for Two LED Project:


7segment - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
7SEG-COM-ANODE
7SEG-COM-CATHODE
AT89C51
U1
5 Message(s)
ANIMATING: 00:01:18.450000 (CPU load 9%)
+1800.0
-2300.0

Step 2: Make the following setup for One LED Project


sevenseg - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
7SEG-COM-ANODE
AT89C51
U1
No Messages
COMPONENT U1, Value=AT89C51, Module=<NONE>, Device=AT89C51, Pinout=C
-600.0
+300.0

Setup for Two LED project for Two LED Project:


7segment - Proteus 8 Professional - Schematic Capture
File Edit View Tool Design Graph Debug Library Template System Help
Schematic Capture
DEVICES
7SEG-COM-ANODE
7SEG-COM-CATHODE
AT89C51
U1
5 Message(s)
ANIMATING: 00:01:18.450000 (CPU load 9%)
+1800.0
-2300.0

Source Code for One LED Project:

```
#include<reg51.h>
void main()
{
	unsigned char seg[10] = {0x00, 0xf9, 0xa4, 0x99, 0x92, 0xf8, 0x80, 0x90}:
	unsigned char x;
	unsigned int i;
	P1 = 0x00;
	while(1)
	{
		for(x = 0 ; x < 10 ; x++)
		{
			P1 = seg[x];
			for(i = 0 ; i < 60000 ; i++);
		}
	}
}
```

Source Code for Two LED Project:

```
#include<reg51.h>
void delay(unsigned int ms)
{
	unsigned int i, j;
	for(i = 0 ; i < ms ; i++);
	for(j = 0 ; j < 1275 ; j++);
}
void main(void)
{
	char number[10] = {0x3F, 0x06, 0x5B, 0x4F, 0x66, 0x6D, 0x7D, 0x07, 0x7F, 0x6F};
	int i;
	P2 = 0x00;
	P3 = 0x00;
	while(1)
	{
		for(i = 0 ; i <= 9 ; i++)
		{
			P2 = number[i];
			for(i = 0 ; i <= 9 ; i++)
			{
				P3 = number[j];
				delay(50);
			}
		}
	}
}
```