SSG EMBEDDED SOLUTIONS

SSG

PROJECTS AT A GLANCE

ABHAY DESHPANDE
Mrch 2, 2022

INDEX

8051 EXPERIMENTAL BOARD

	https://www.nuvoton.com/resource-download.jsp?tp_GUID=SW0320120105135349	
7	**Program files in c code in 8051 folder**	
8	**XCTU Download for serial communication program checking**	**8 to 11**
9	**Refer 8051 Development board folder (in PEN DRIVE)for** **Datasheet – 7447,DS1307,24C08,MAX232,** **Manual –QUICK START** **New – LESSON-C CODE** **8051 board image** **Software**	

I]DRIVER INSTALLATION:

Click on CH34x_Instal_Window_v3_4Driver

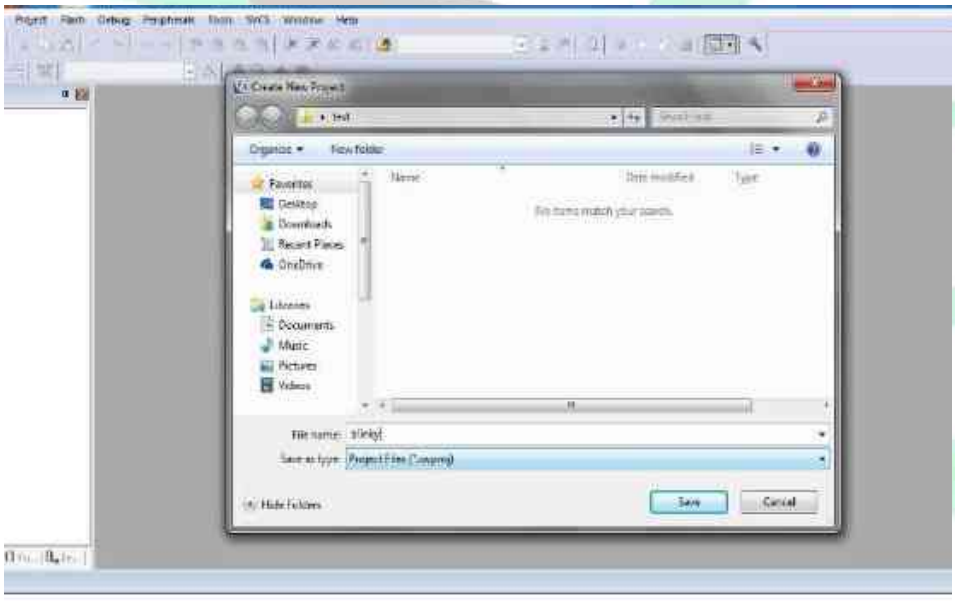
II]Keil software installation for changes in program

https://www.keil.com/download/product/

III]How to create file in KEIL--Using C:

Open the Keil IDE, under main menu goto "Project->New uVision Project…" and a window prompt will open asking to save the new project. Type your desired project name and save.

Step 2. After that, a new window will appear as shown below



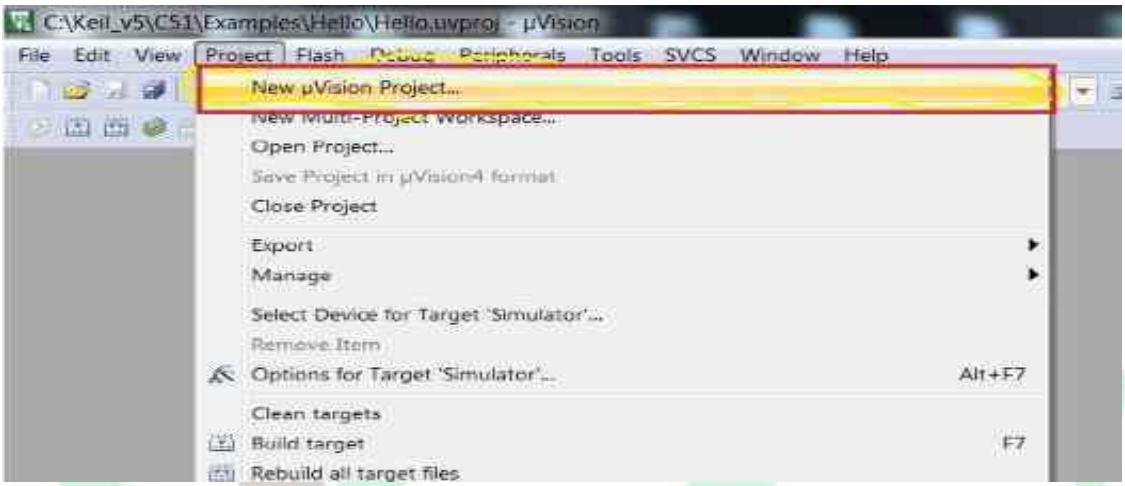
Step By Step: Create Keil Project for 8051

STEP1

Before we proceed any further, first launch Keil uVision5 application from computers program menu.

STEP2

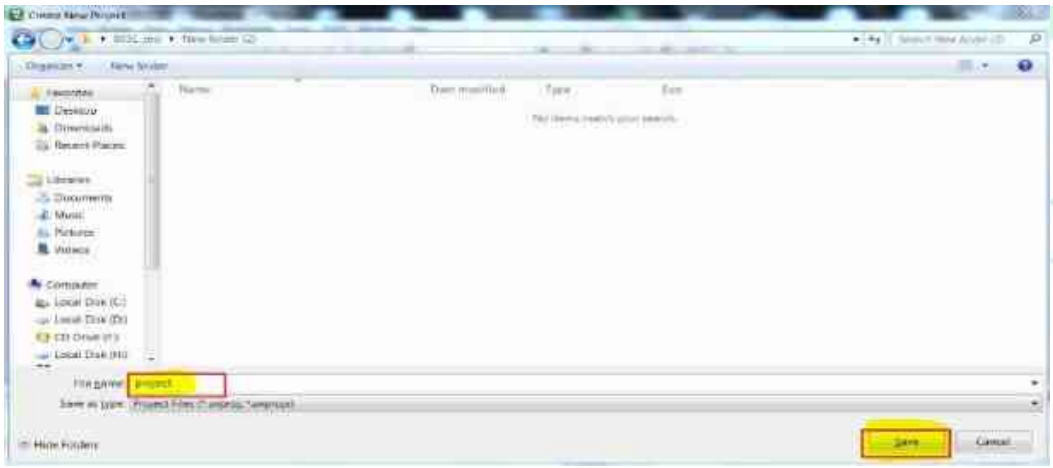
Now click on **Project** tab and select **New uVision Project…**



Create new Keil Project

STEP3

Give any name to project (Here we have given a name as **project**).
And **Save** project in new folder.

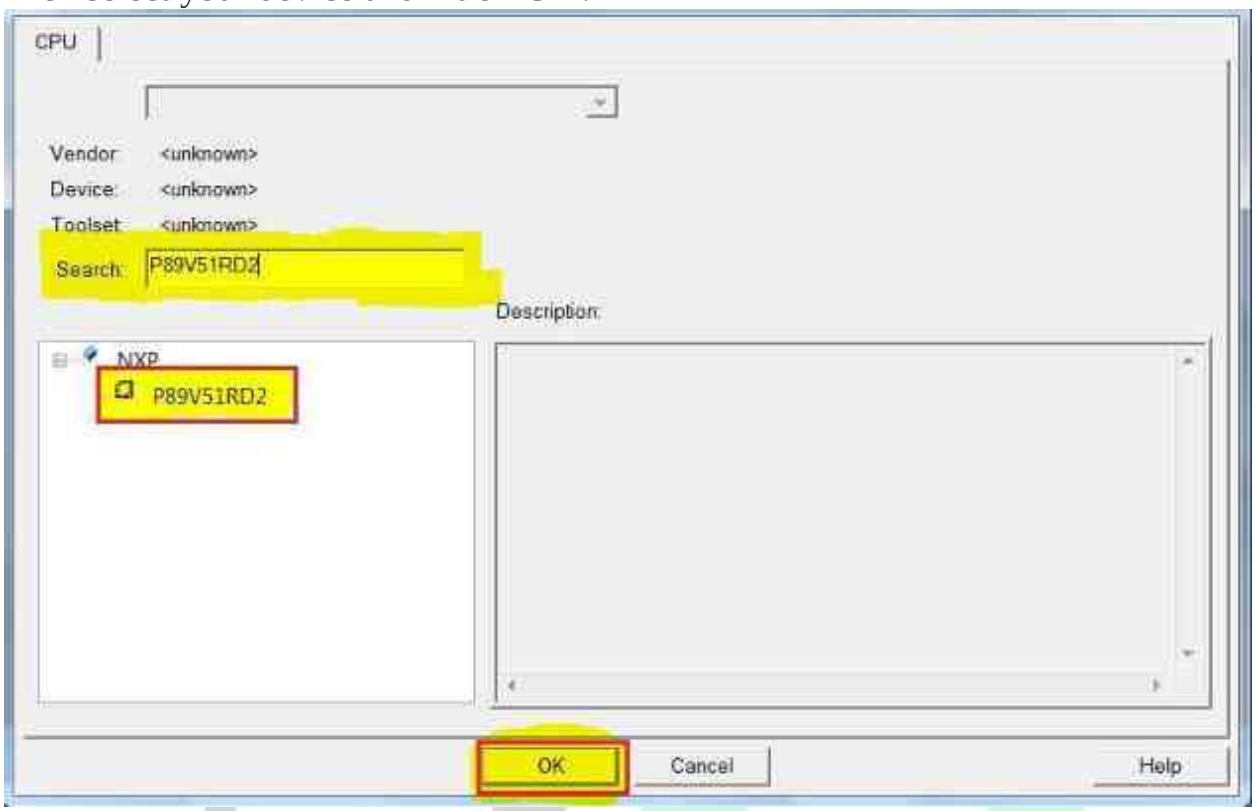


Name and save Project

STEP4

- Now in select **Device for Target 'Target 1'…** Search for appropriate derivative from family of 8051 microcontroller (for example: P89V51RD2, AT89C51, AT89S51 etc). Here in this case we have chosen **Nuvoton W78E052D**. As we'll be experimenting in future with W78E052D

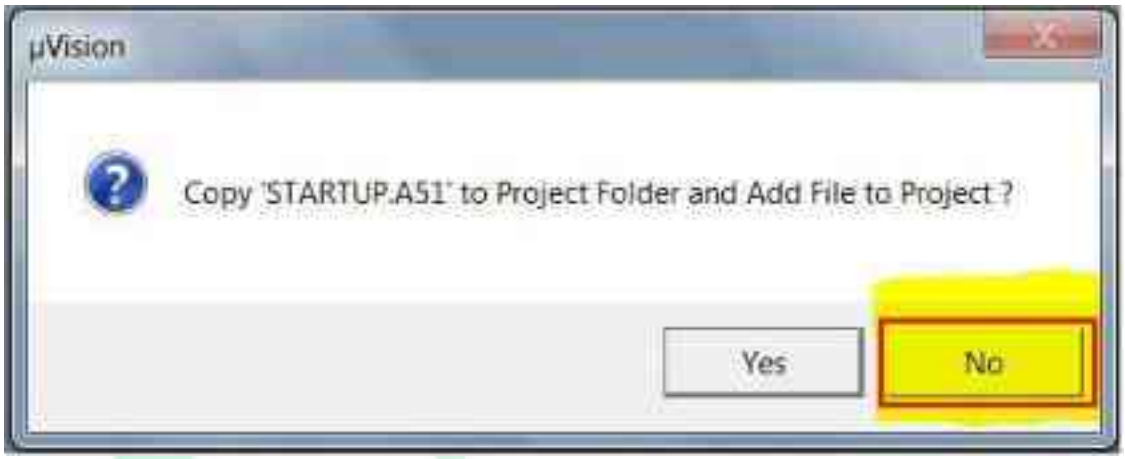
Then select your device and hit on **OK**.



Select W78E052D Microcontroller

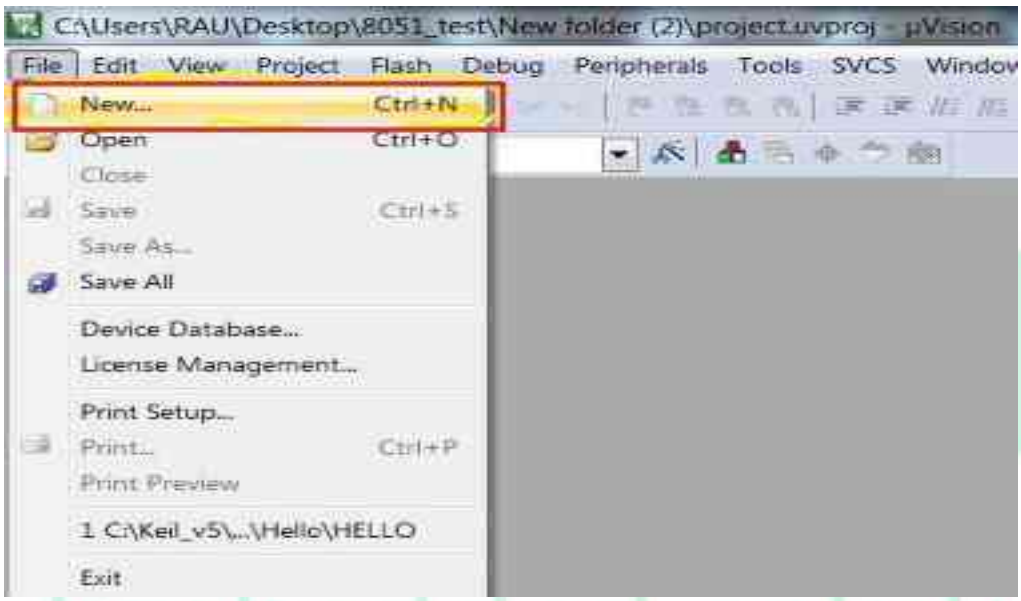
STEP5

When ask for **Copy 'STARTUP.A51' to Project Folder and Add File to Project?** Click **NO**



STEP6

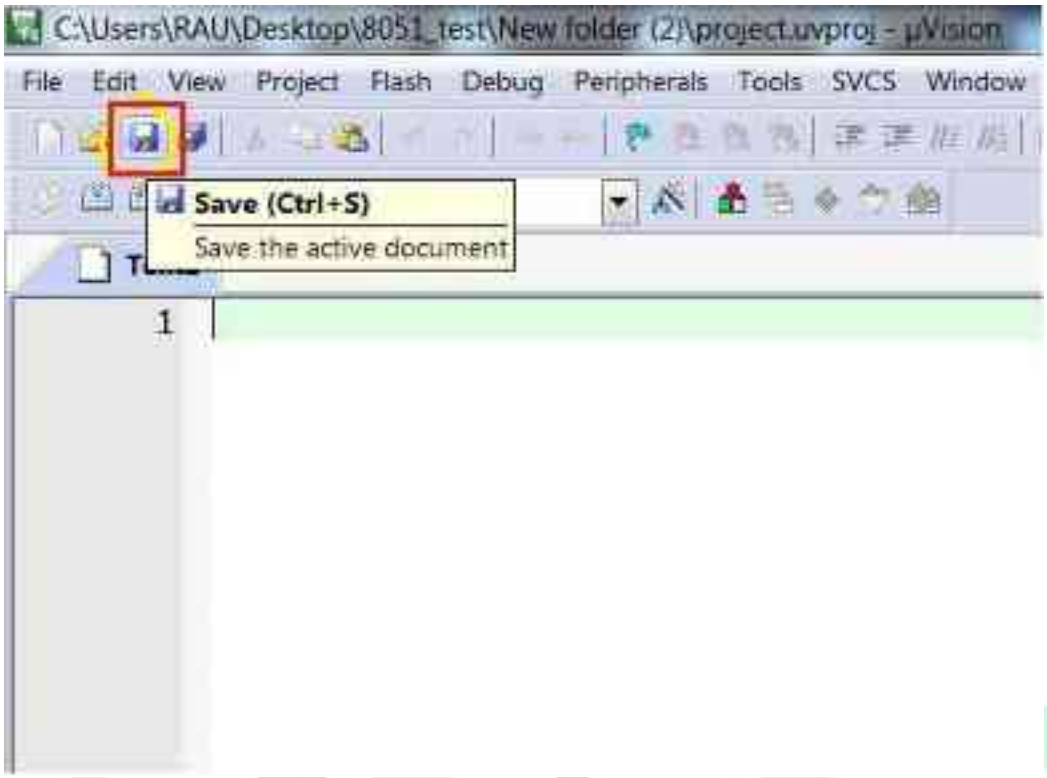
Now to create source file go to **File** tab and click on **New…**



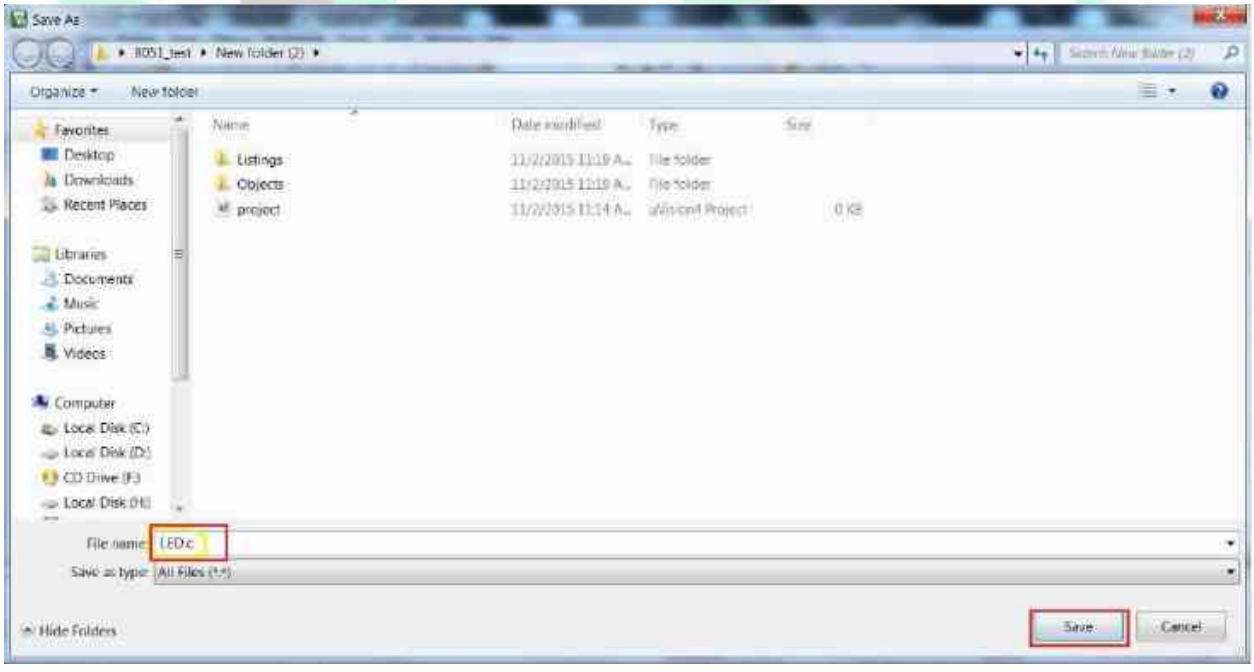
Create new source file

STEP7

Here **save** this file with **.c** (e.g. LED.c) extension in **project** folder. which we have created earlier.



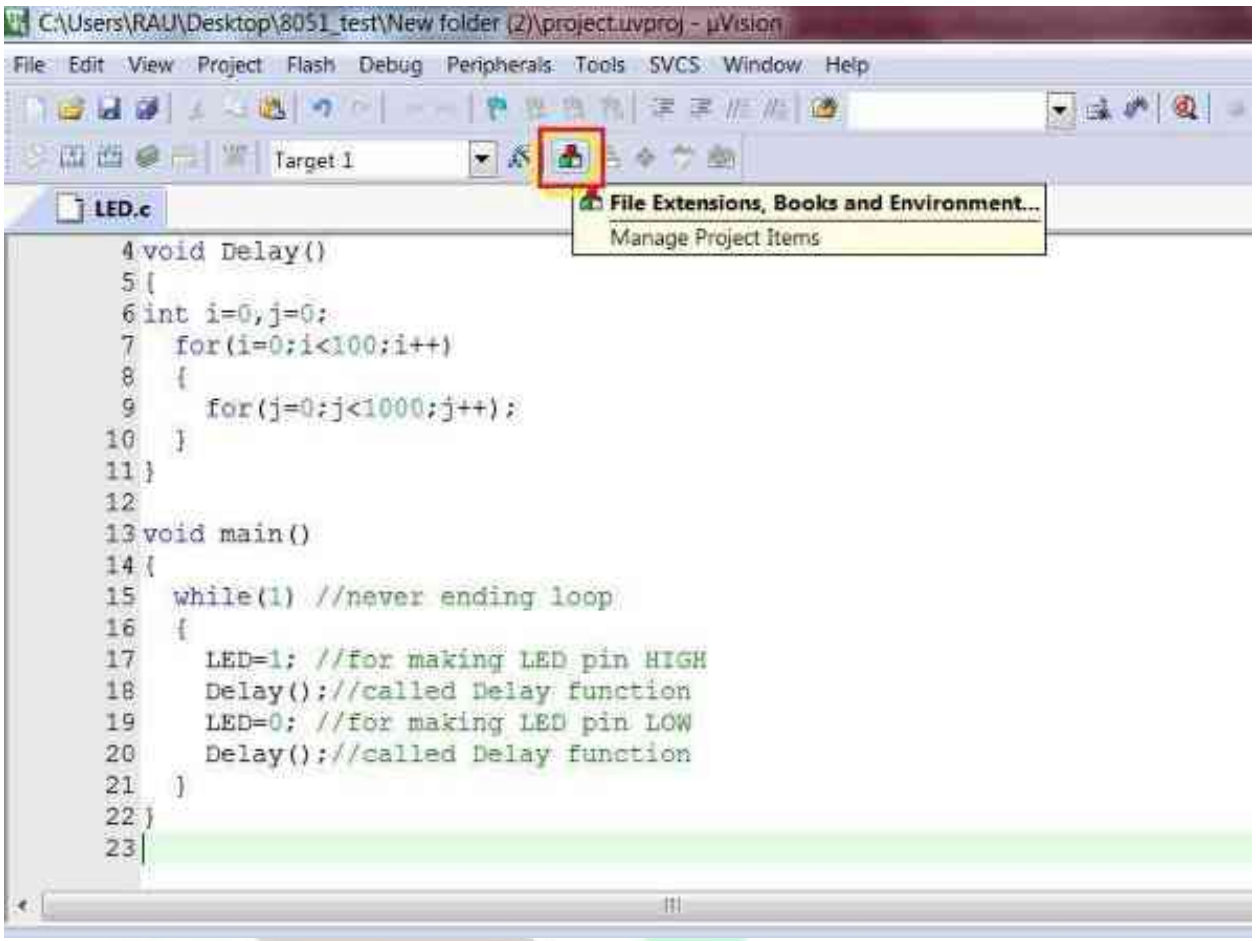
Save file as c extension



Save as led.c file

STEP8

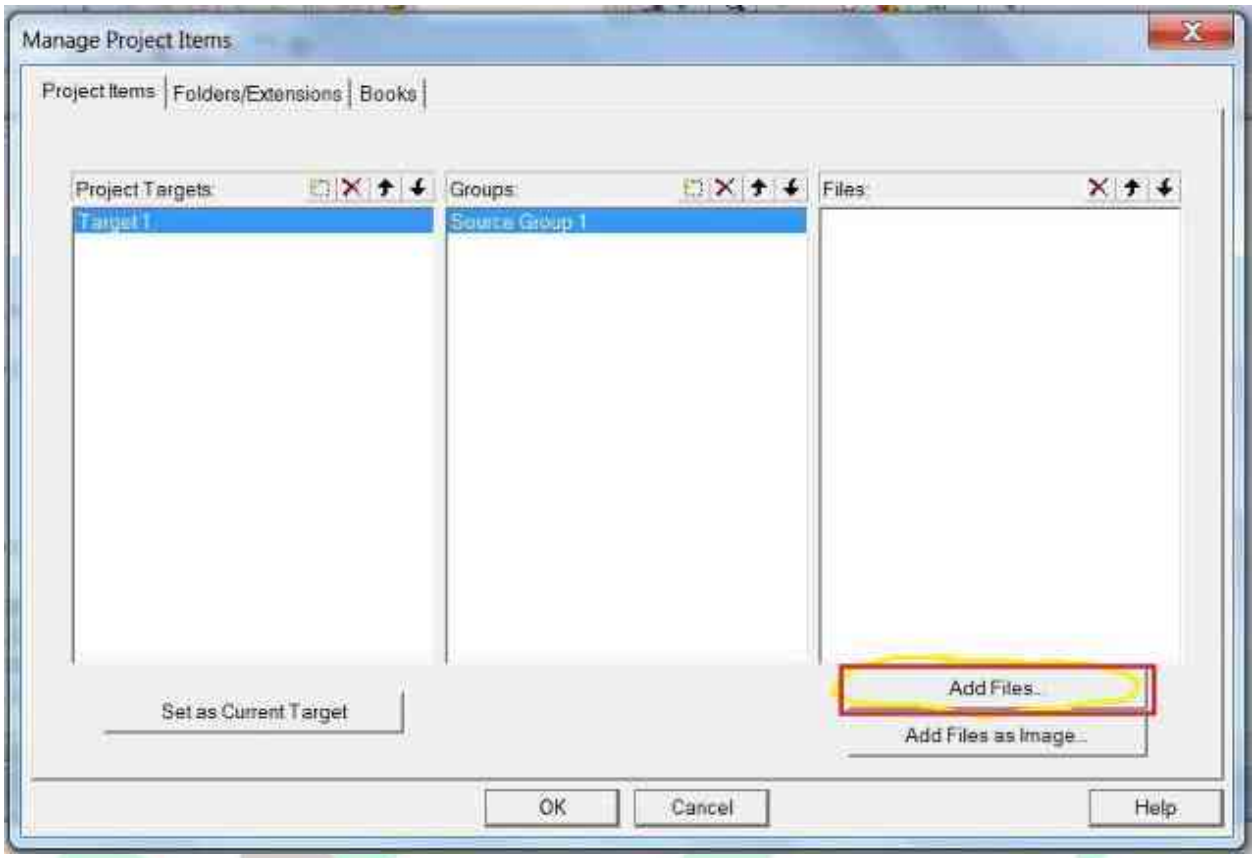
At this point, we're ready to write code/program. Now click on **File Extension, Books and Environment…** tab



Setup file extension

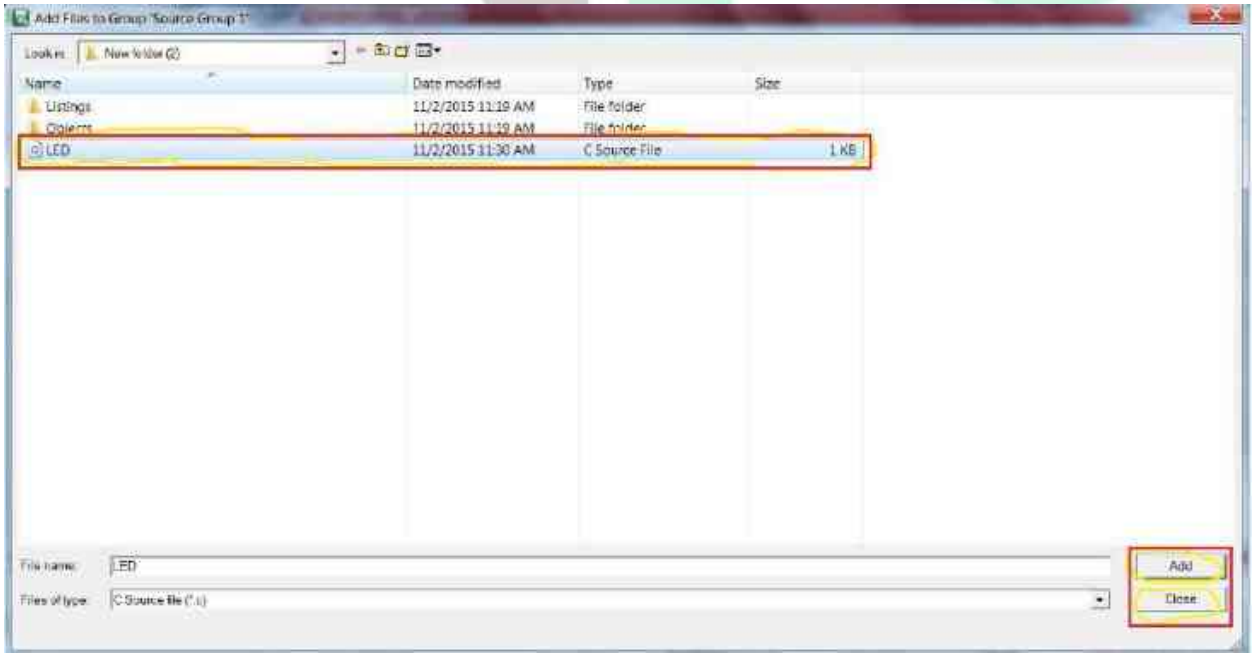
STEP9

Then click on **Add Files**...



Add led.c file

Now **select file** which we have saved with .c extension. Then click on **Add** button. And then hit on **Close**.

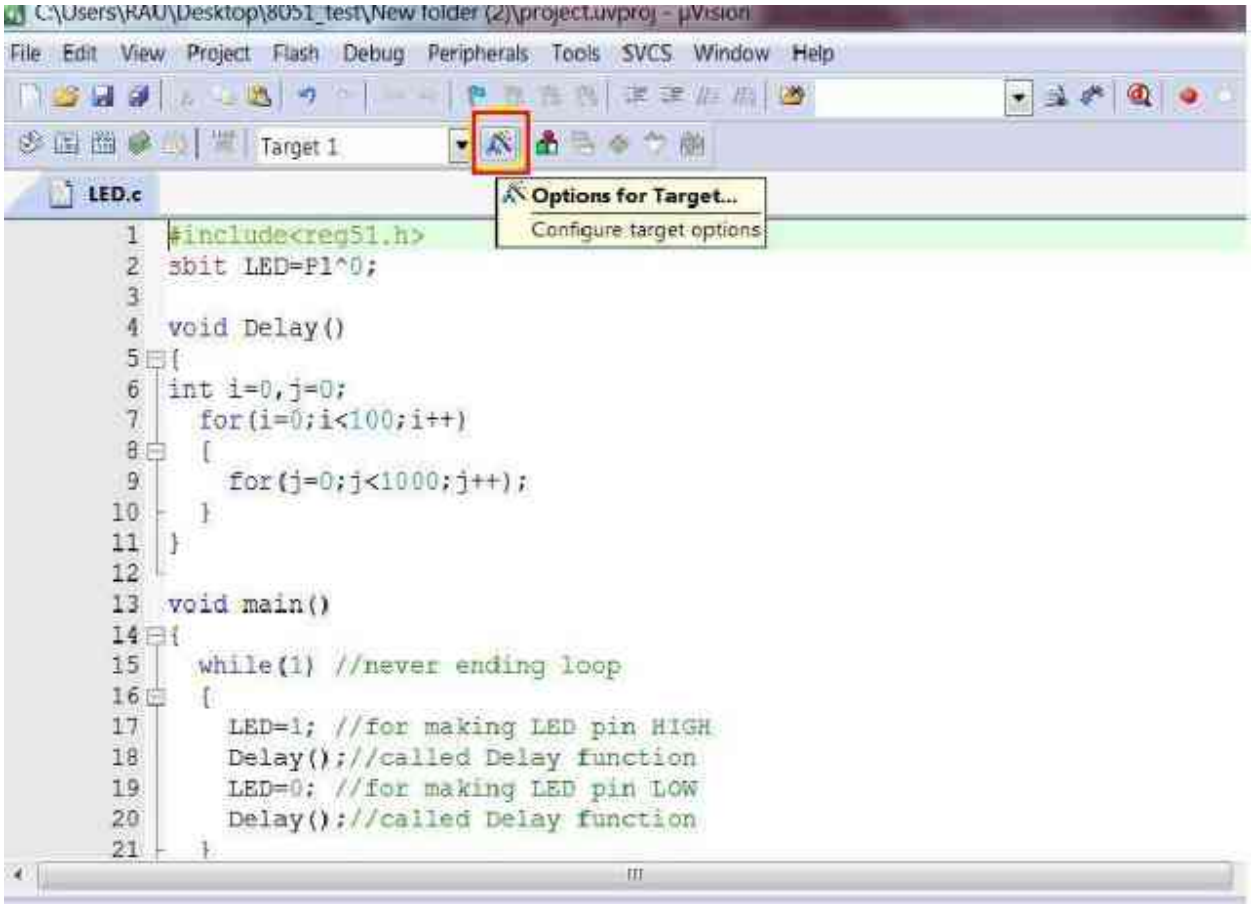


Add file to project

Here you can see your file gets added in Files: window. Now click **OK**

STEP10

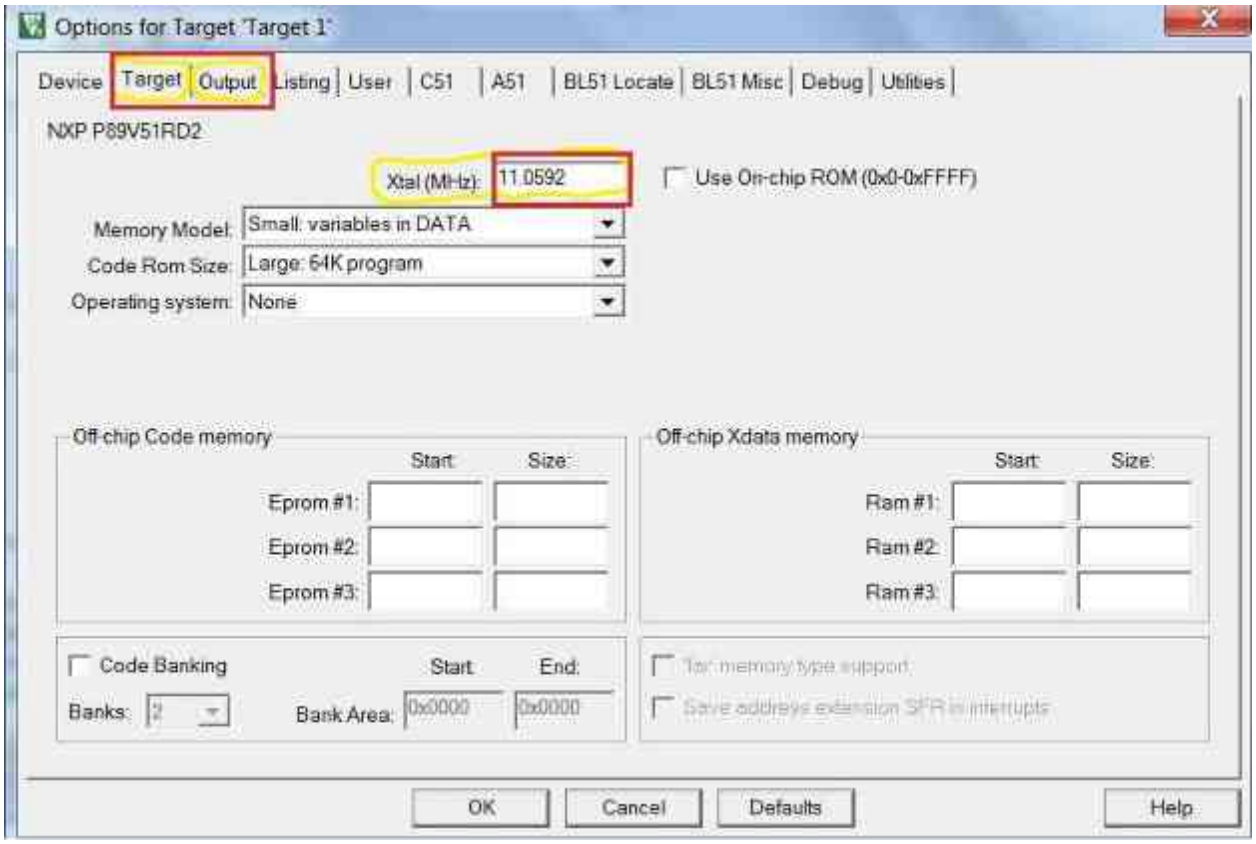
Now click on **Options for Target**… tab



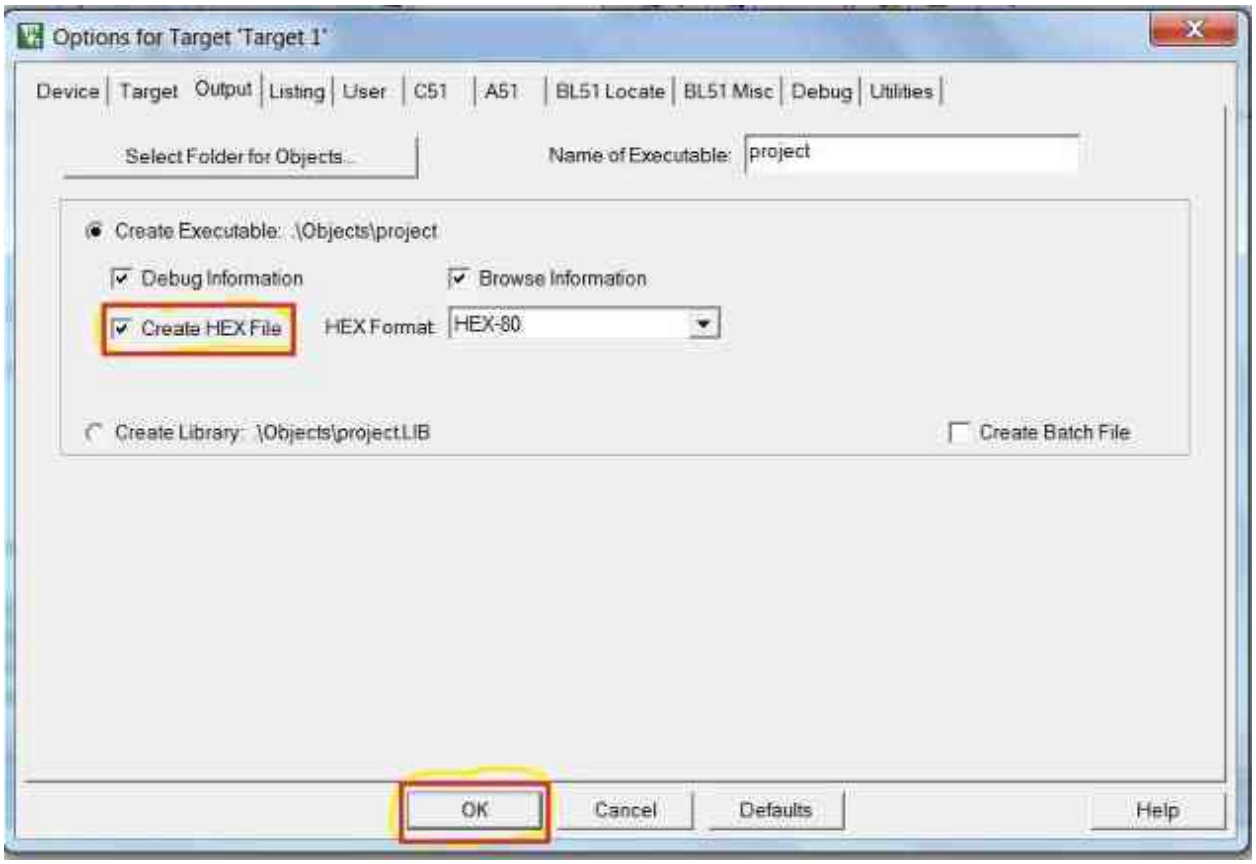
Option for Target

Option for Target

In Target window type clock frequency which we're using for 8051 microcontroller. Throughout this tutorial series we are going to use **11.0592MHz crystal** clock.



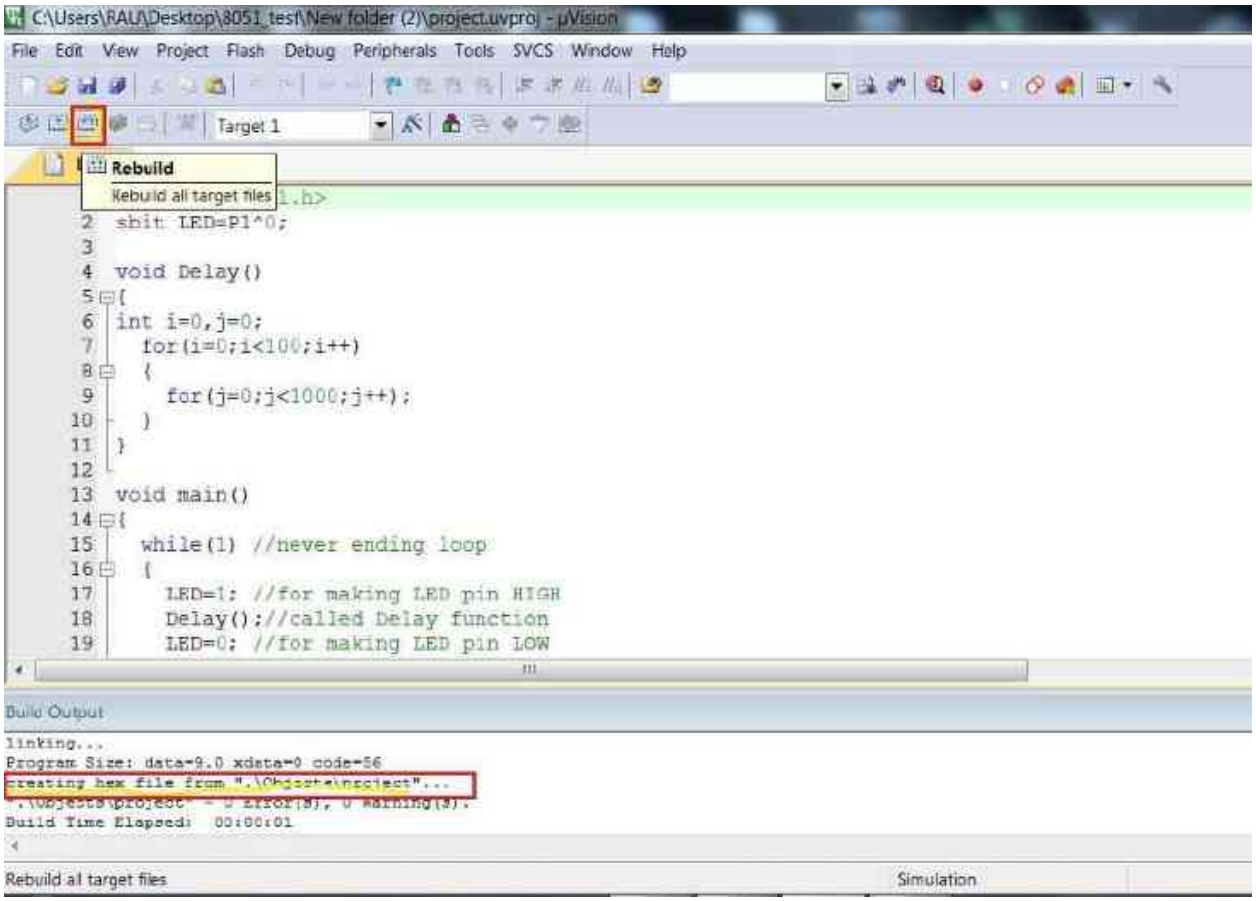
Now in **Output** tab click on check box of **Create HEX File**. Then hit on **OK**



Create hex file

STEP11

Now click on "**Build**" tab for building project file. Then click on **Rebuild**

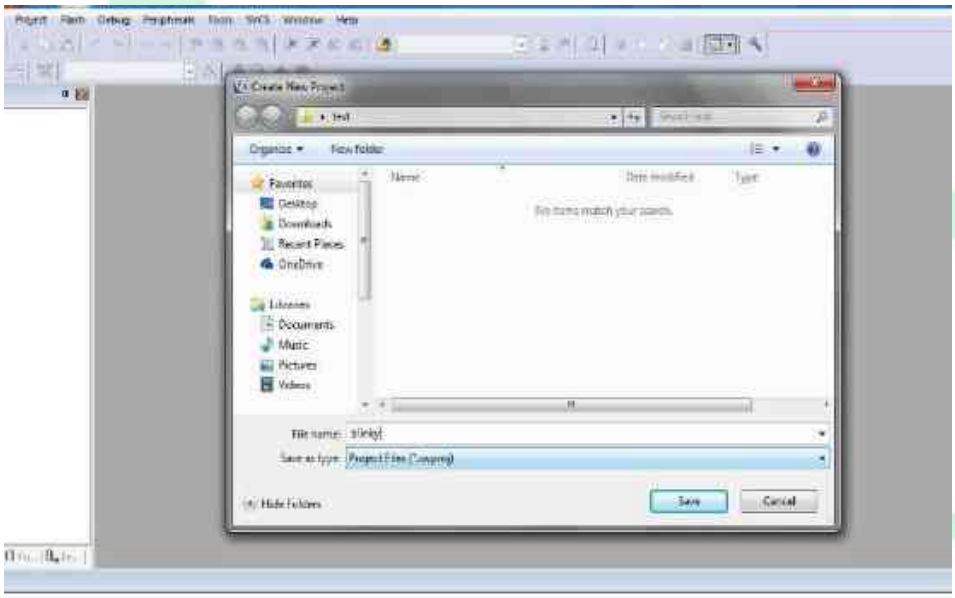


At this point, if there isn't be any error then we'll get HEX file into "**Objects folder**" with .HEX extension. This is how we'll Create Keil project for 8051 Microcontroller using uVision5.

IV]How to create file in KEIL—For ASSEMBLY LANGUAGE:

Open the Keil IDE, under main menu goto "Project->New uVision Project…" and a window prompt will open asking to save the new project. Type your desired project name and save.

Step 2. After that, a new window will appear as shown below



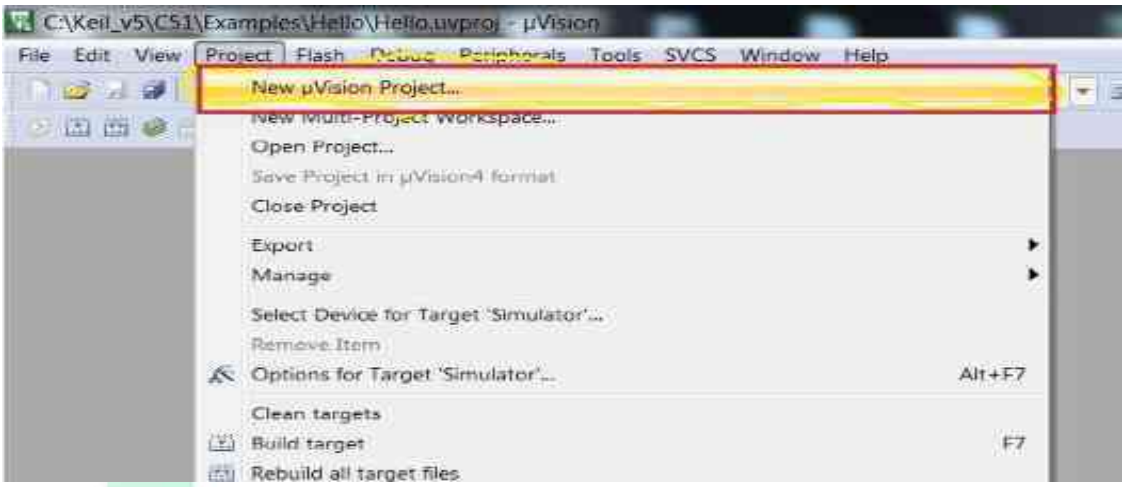
Step By Step: Create Keil Project for 8051

STEP1

Before we proceed any further, first launch Keil uVision5 application from computers program menu.

STEP2

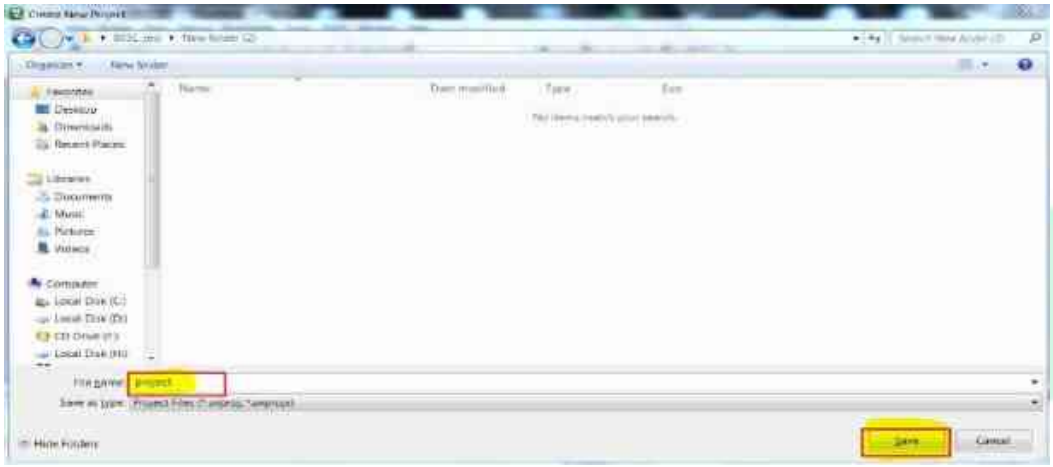
Now click on **Project** tab and select **New uVision Project…**



Create new Keil Project

STEP3

Give any name to project (Here we have given a name as **project**).
And **Save** project in new folder.

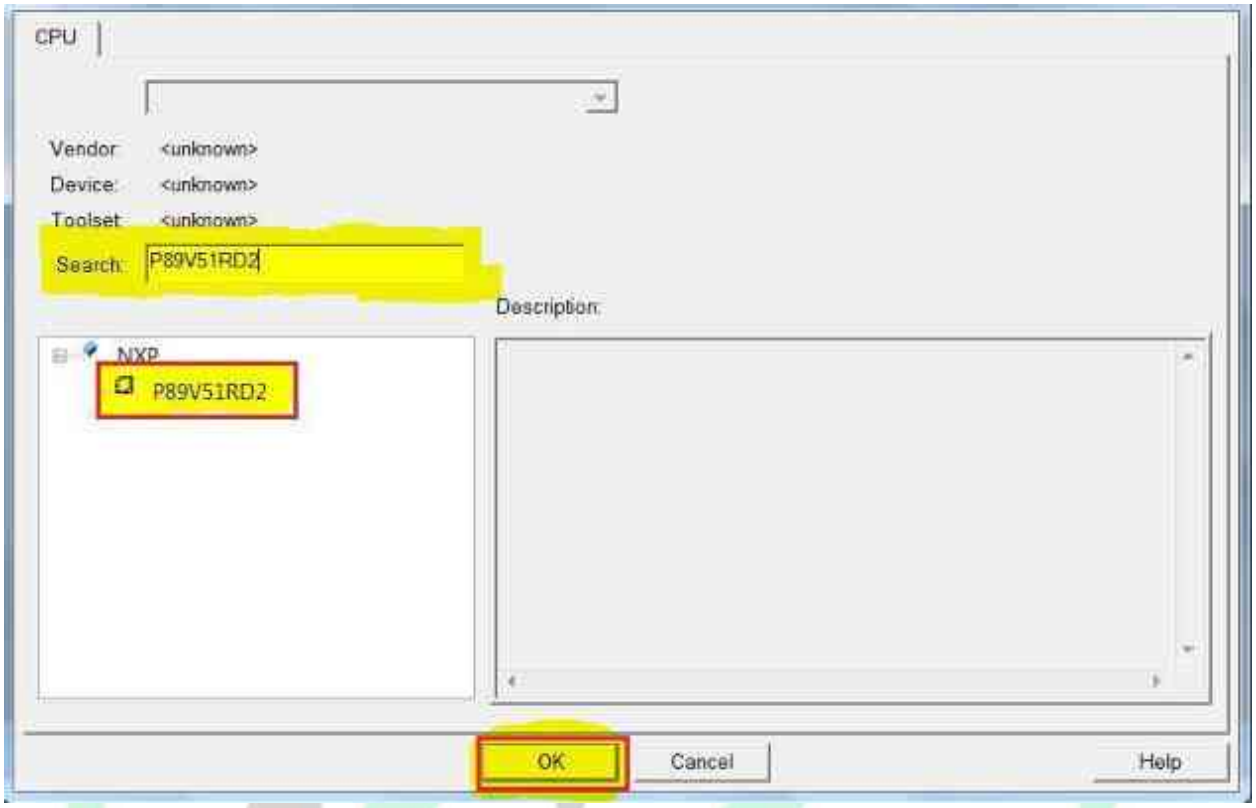


Name and save Project

STEP4

- Now in select **Device for Target 'Target 1'…** Search for appropriate derivative from family of 8051 microcontroller (for example: P89V51RD2, AT89C51, AT89S51 etc). Here in this case we have chosen **Nuvoton W78E052D**. As we'll be experimenting in future with W78E052D

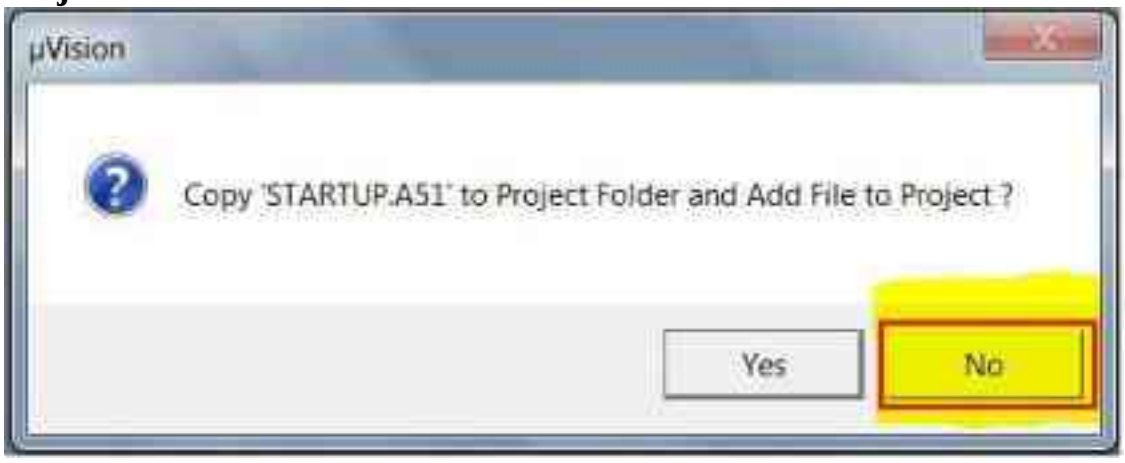
Then select your device and hit on **OK**.



Select W78E052D Microcontroller

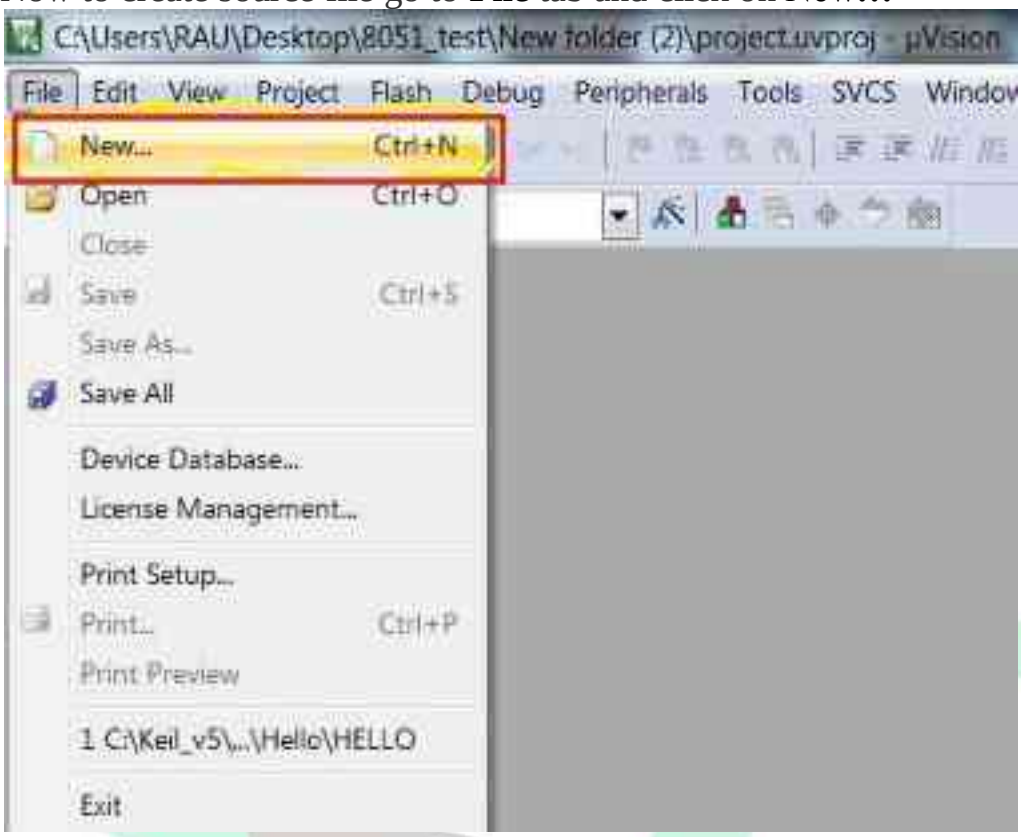
STEP5

When ask for **Copy 'STARTUP.A51' to Project Folder and Add File to Project?** Click **NO**



STEP6

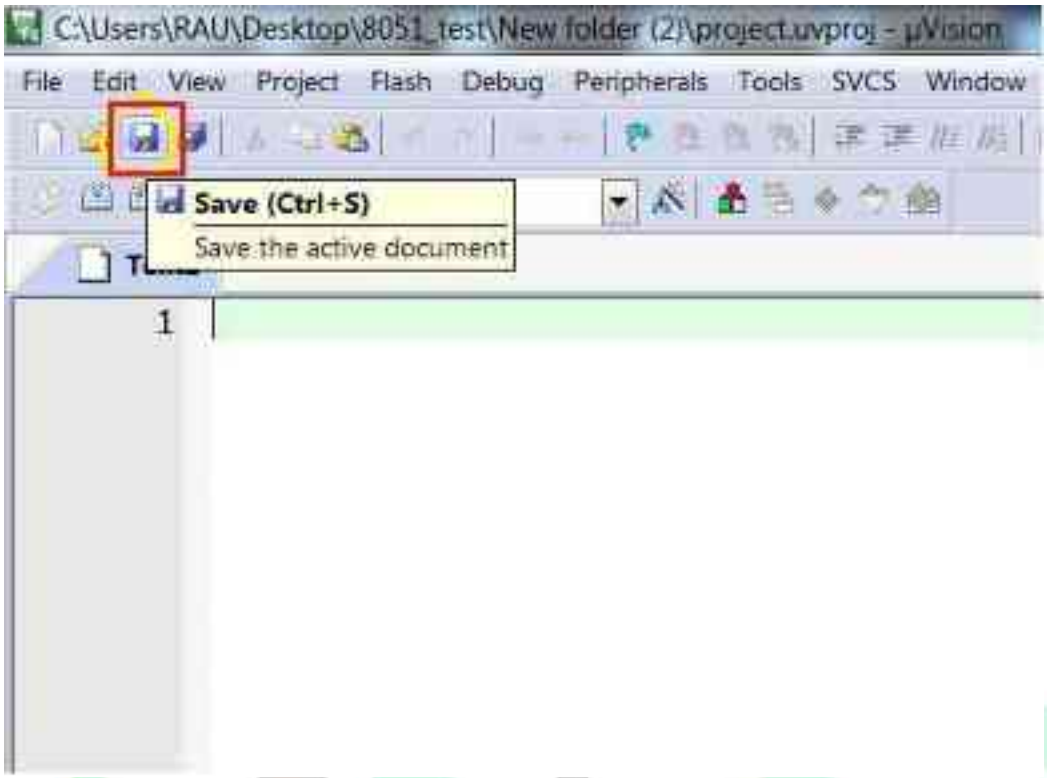
Now to create source file go to **File** tab and click on **New…**



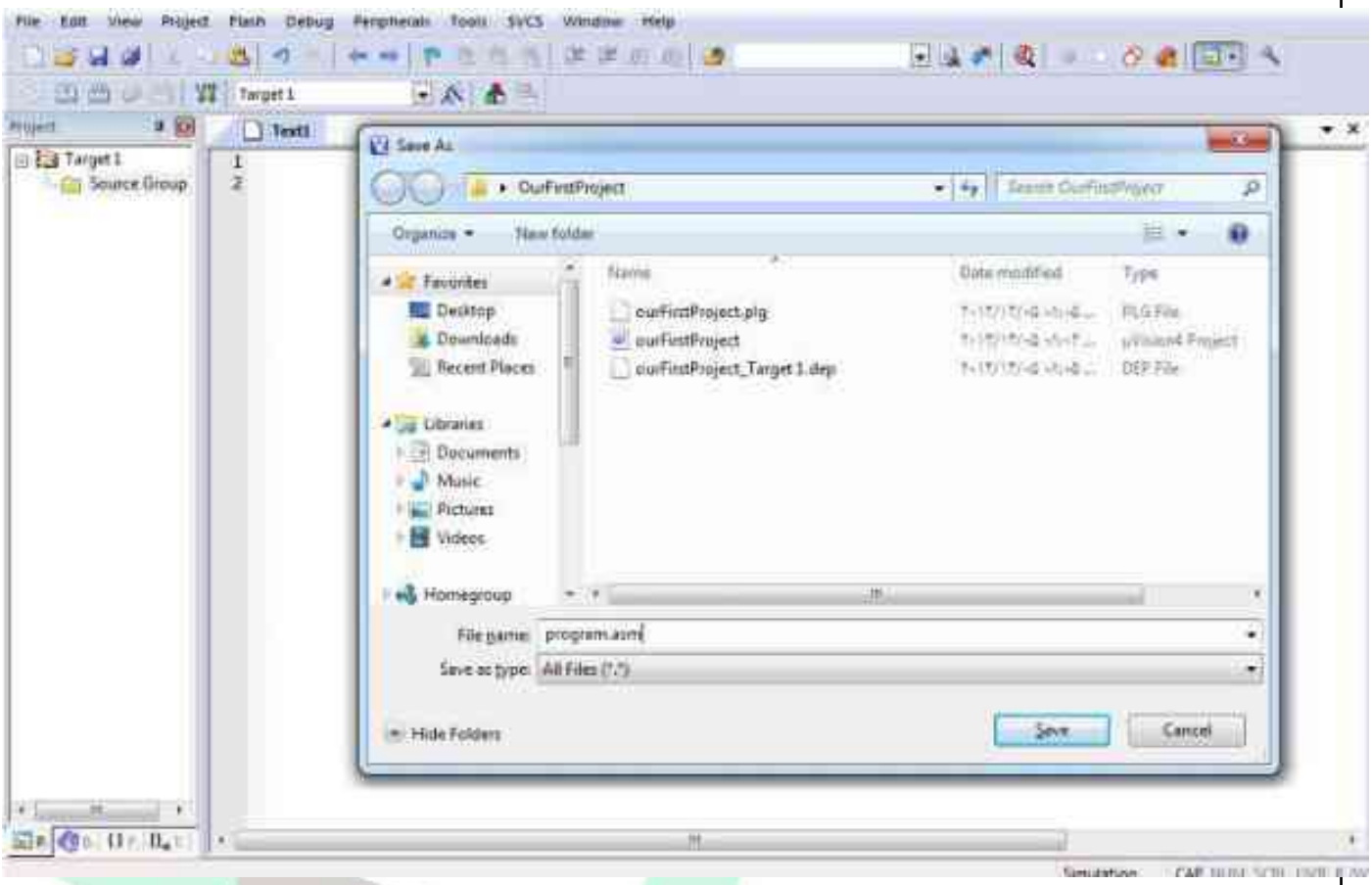
Create new source file

STEP7

Here **save** this file with **.asm** (e.g. program.asm) extension in **project** folder. which we have created earlier.



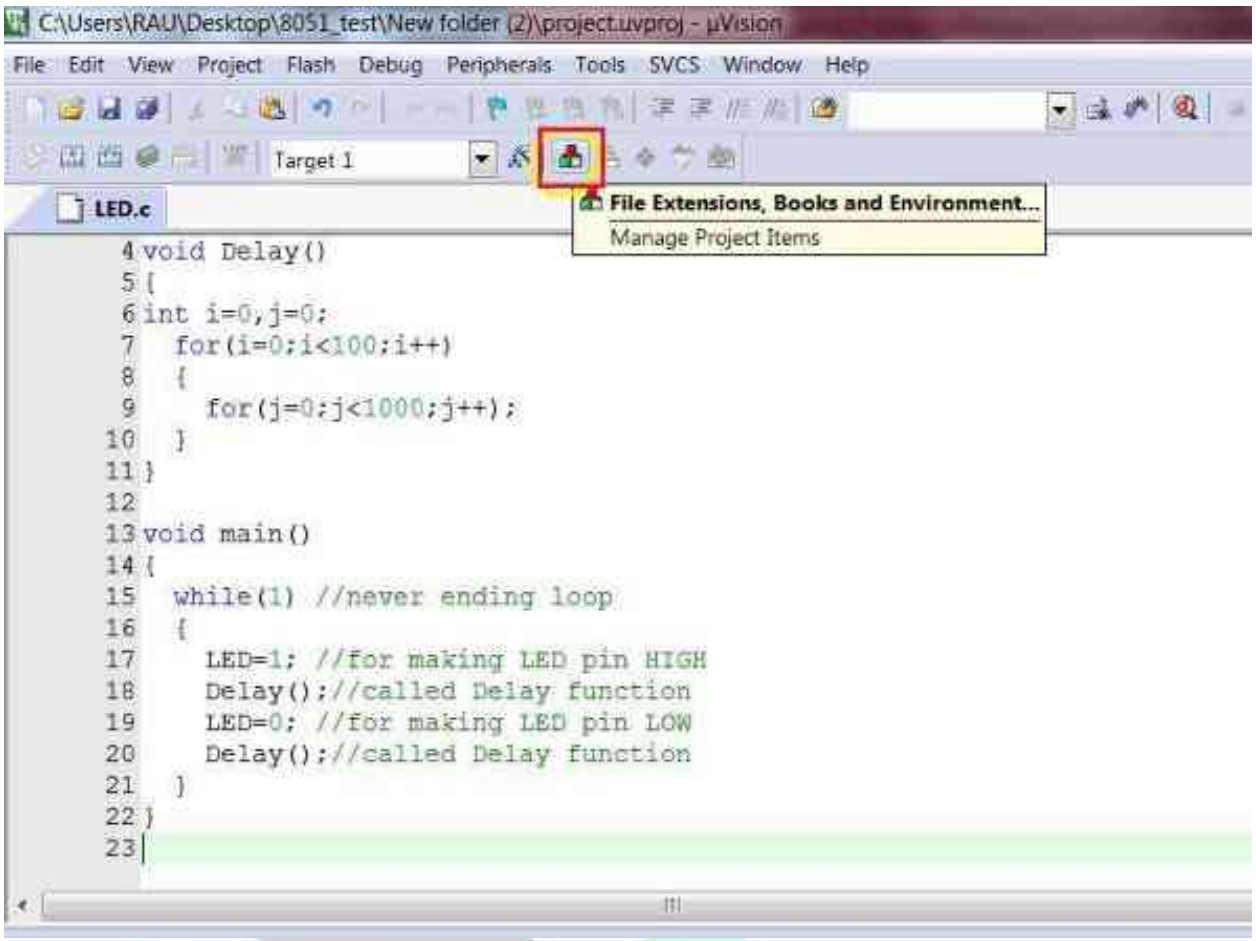
Save file as c extension



Save as led.c file

STEP8

At this point, we're ready to write code/program. Now click on **File Extension, Books and Environment…** tab



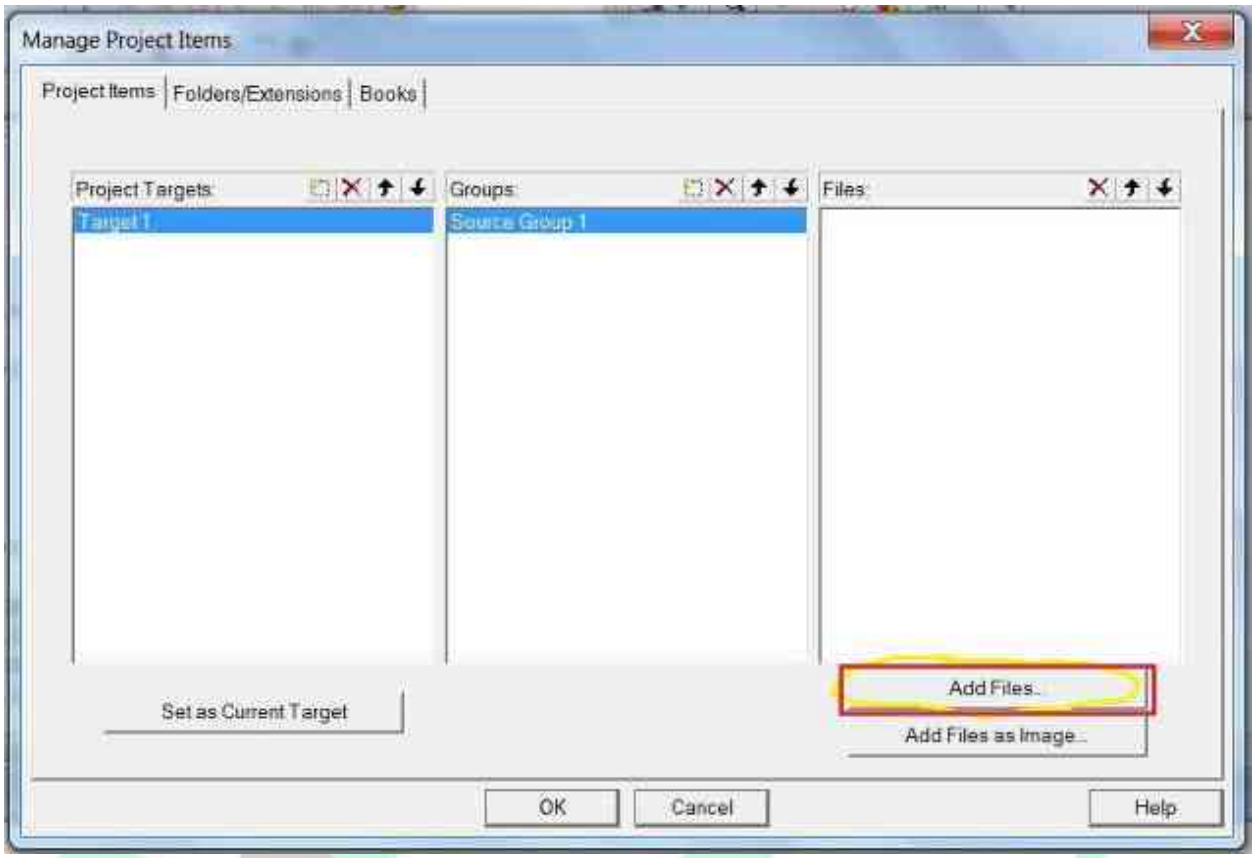
Setup file extension

code.asm

```
1   org 00H
2
3   ;MAIN PROGRAM
4
5   toggle:  MOV P1, #01H        ; move 00000001 to PORT1
6            CALL delay          ; execute delay
7            MOV A, P1           ; move PORT1 value to accumulator
8            CPL A               ; complement PORT1 value
9            MOV P1, A           ; move 11111110 to PORT1
10           CALL delay          ; execute delay
11
12           sjmp toggle
13
14
15  ;DELAY SUB-ROUTINE
16
17  delay:   MOV R5, #10         ; load register R5 with 10
18  third:   MOV R6, #200        ; load register R6 with 200
19  second:  MOV R7, #200        ; load register R7 with 200
20
21           DJNZ R7, $          ; decrement R7 till it is zero
22           DJNZ R6, second     ; decrement R6 till it is zero
23           DJNZ R5, third      ; decrement R5 till it is zero
24
25           ret                 ; go back to main program
26  END
27
```

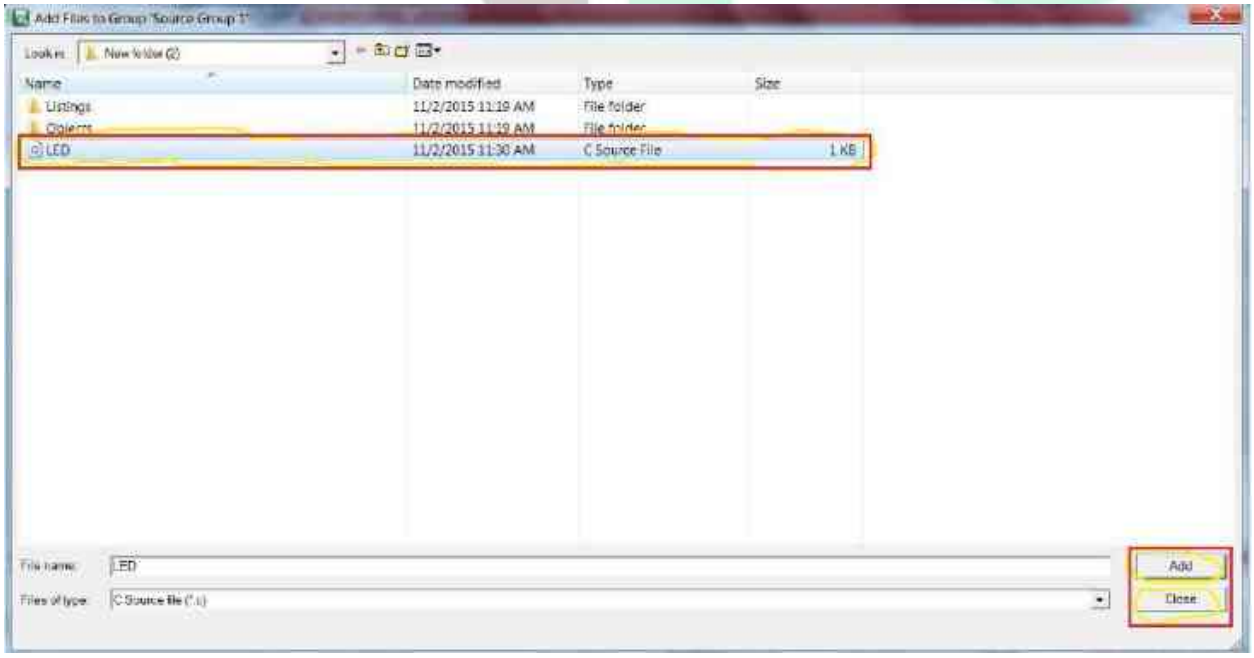
STEP9

Then click on **Add Files…**



Add led.c file

Now **select file** which we have saved with .c extension. Then click on **Add** button. And then hit on **Close**.

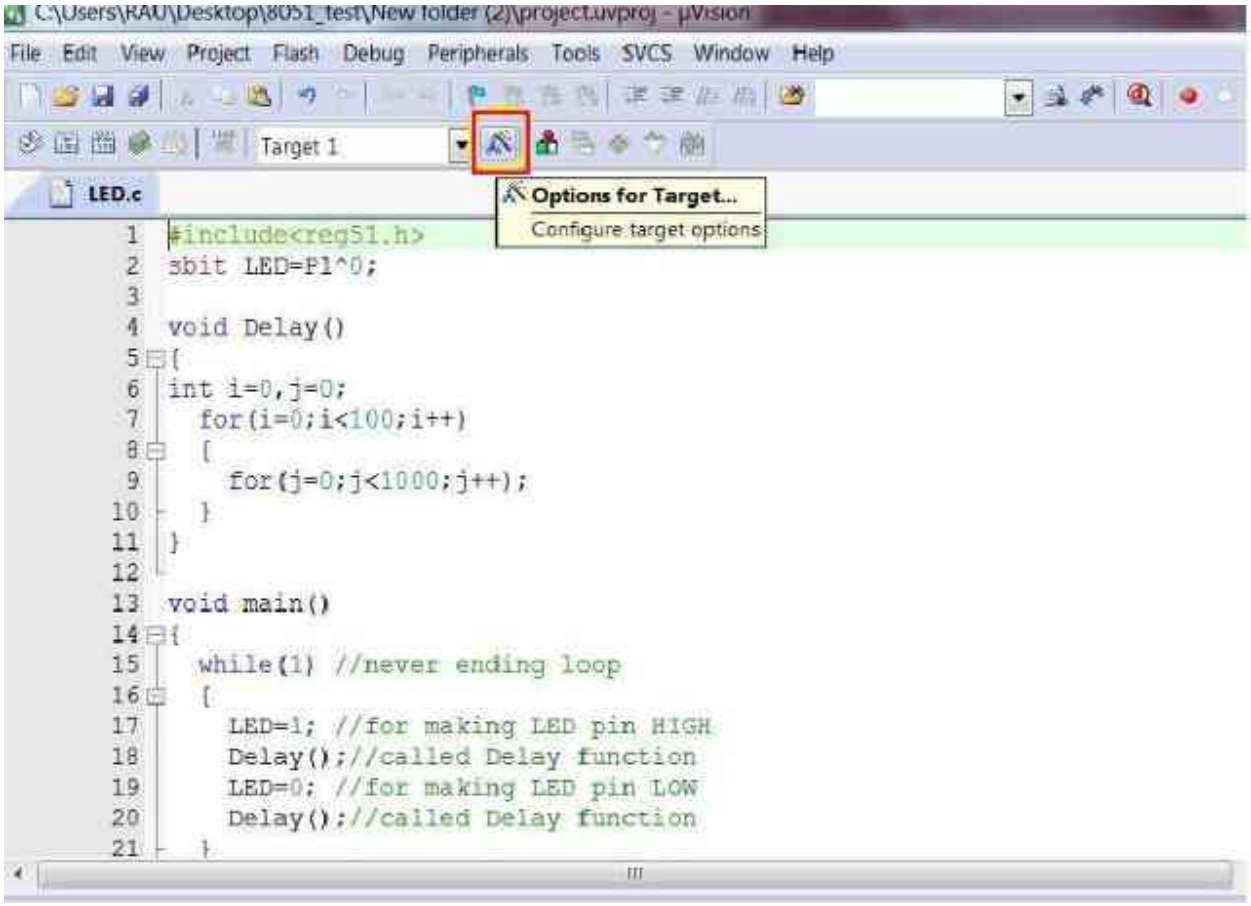


Add file to project

Here you can see your file gets added in Files: window.ie program.asm
Now click **OK**

STEP10

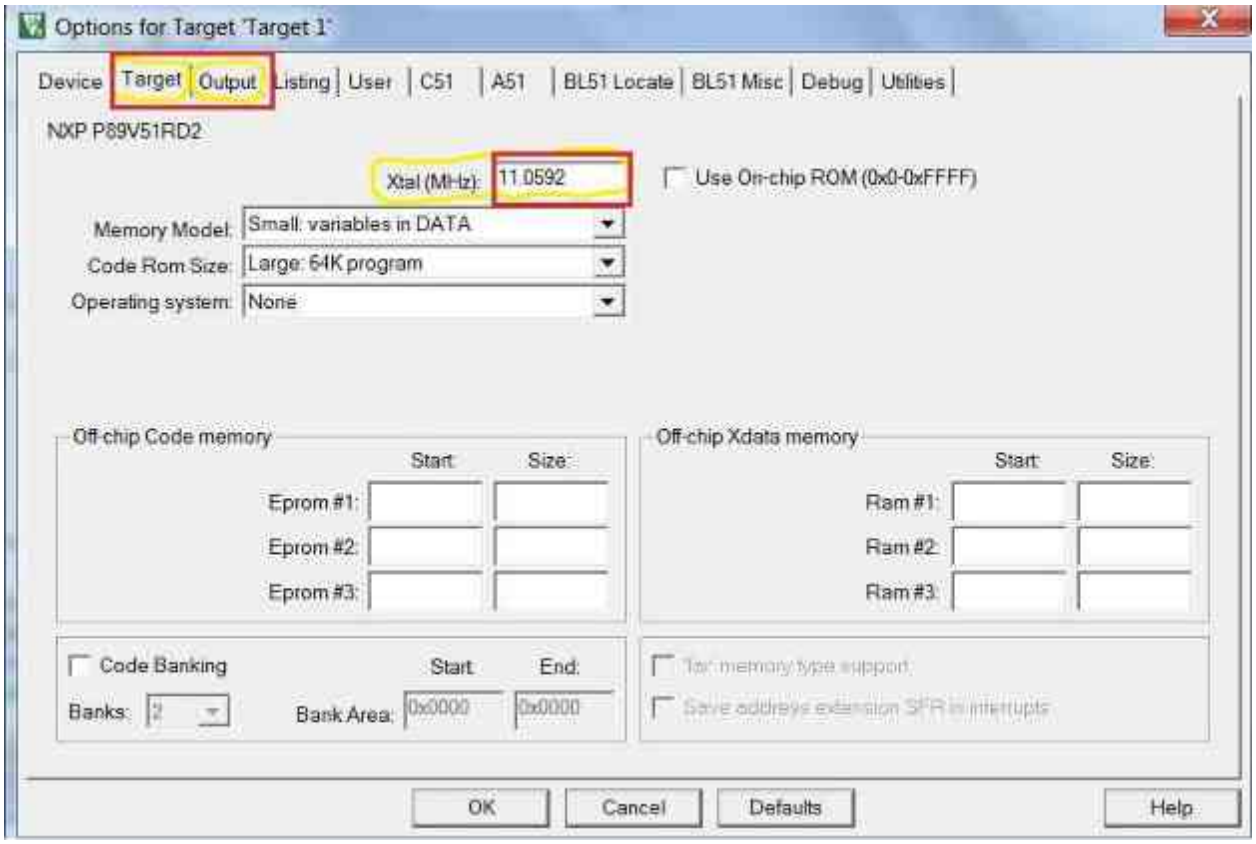
Now click on **Options for Target**… tab



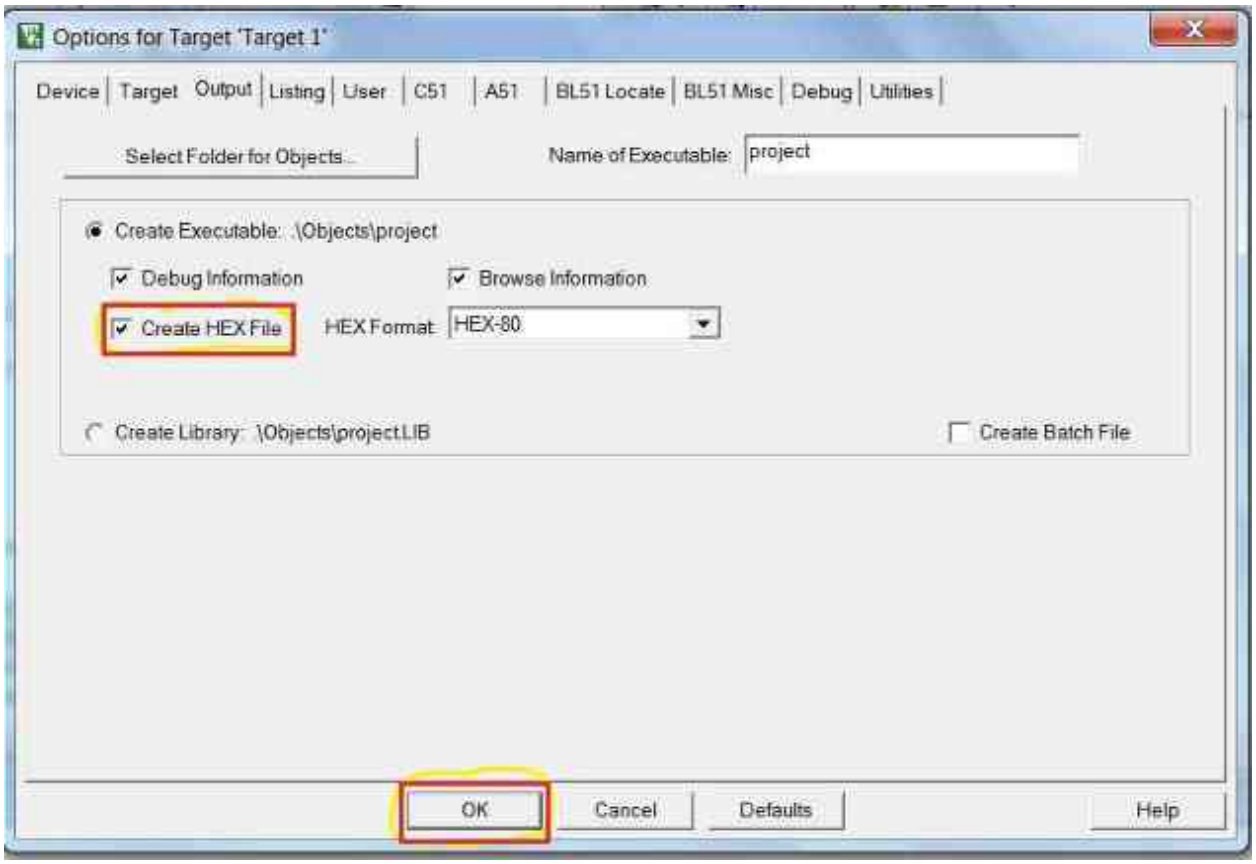
Option for Target

Option for Target

In Target window type clock frequency which we're using for 8051 microcontroller. Throughout this tutorial series we are going to use **11.0592MHz crystal** clock.



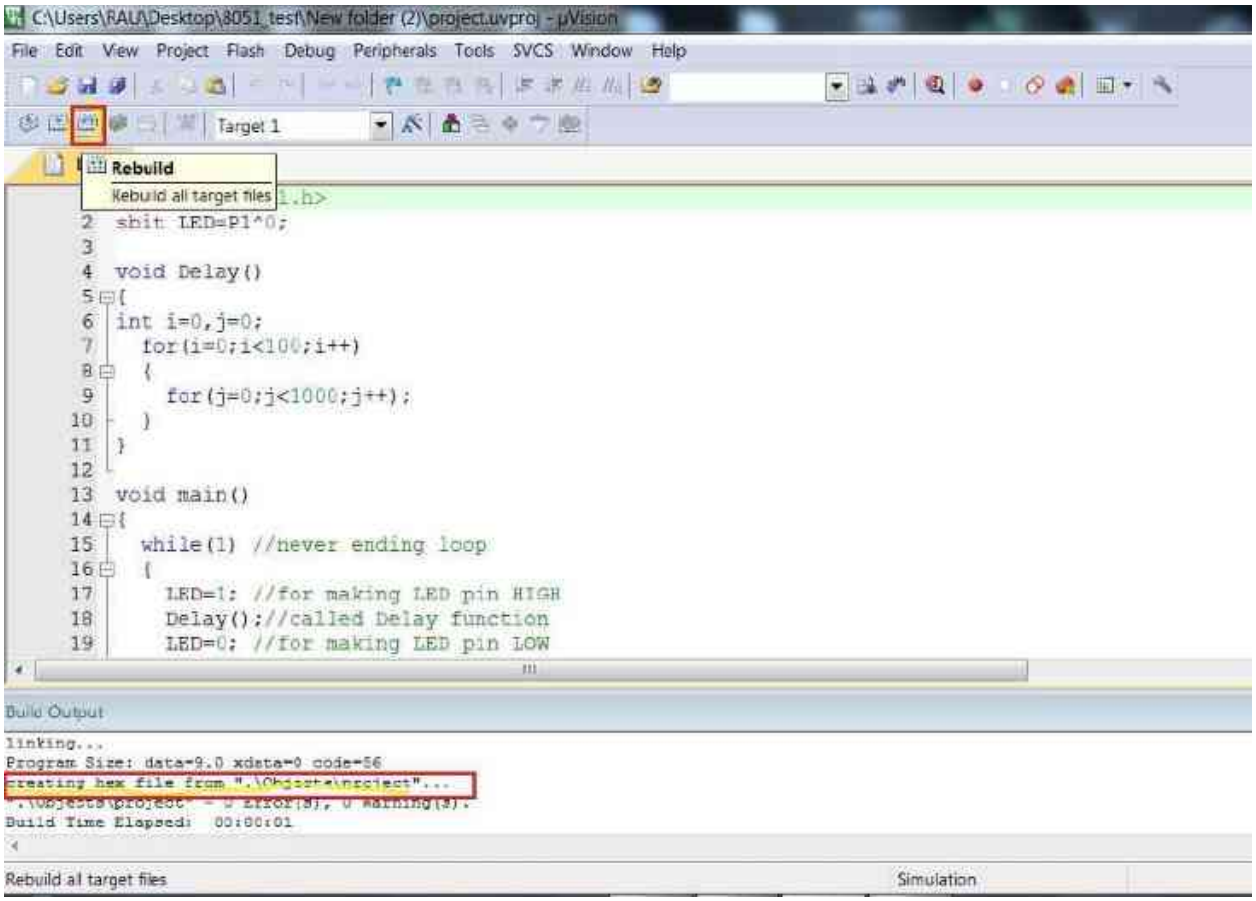
Now in **Output** tab click on check box of **Create HEX File**. Then hit on **OK**



Create hex file

STEP11

Now click on "**Build**" tab for building project file. Then click on **Rebuild**



At this point, if there isn't be any error then we'll get HEX file into "**Objects folder**" with .HEX extension. This is how we'll Create Keil project for 8051 Microcontroller using uVision5.

code.asm

```
1   org 00H
2
3   ;MAIN PROGRAM
4
5   toggle:  MOV P1, #01H        ; move 00000001 to PORT1
6            CALL delay          ; execute delay
7            MOV A, P1           ; move PORT1 value to accumulator
8            CPL A               ; complement PORT1 value
9            MOV P1, A           ; move 11111110 to PORT1
10           CALL delay          ; execute delay
11
12           sjmp toggle
13
14
15  ;DELAY SUB-ROUTINE
16
17  delay:   MOV R5, #10         ; load register R5 with 10
18  third:   MOV R6, #200        ; load register R6 with 200
19  second:  MOV R7, #200        ; load register R7 with 200
20
21           DJNZ R7, $          ; decrement R7 till it is zero
22           DJNZ R6, second     ; decrement R6 till it is zero
23           DJNZ R5, third      ; decrement R5 till it is zero
24
25           ret                 ; go back to main program
26  END
27
```

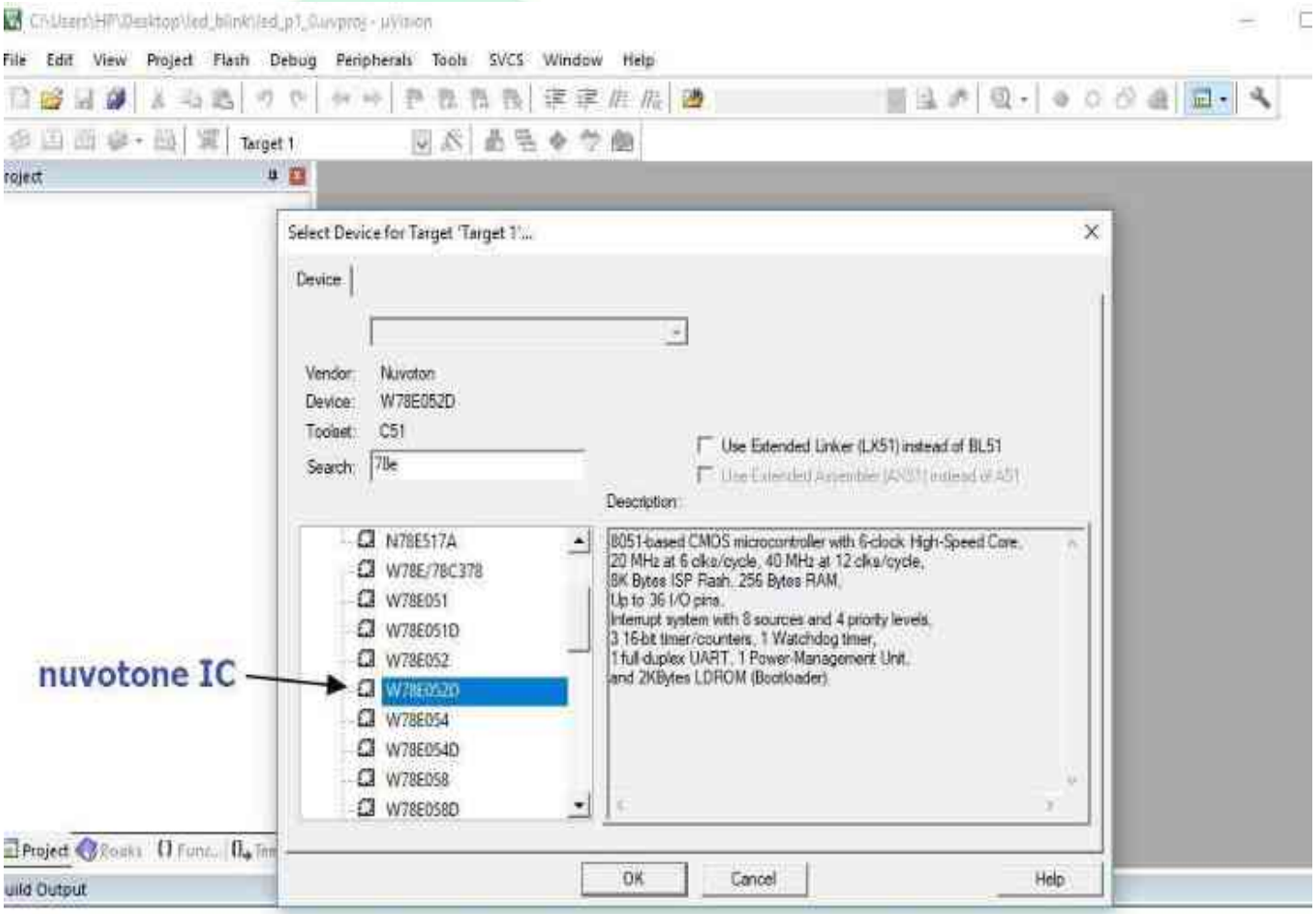

V]KEIL IDE

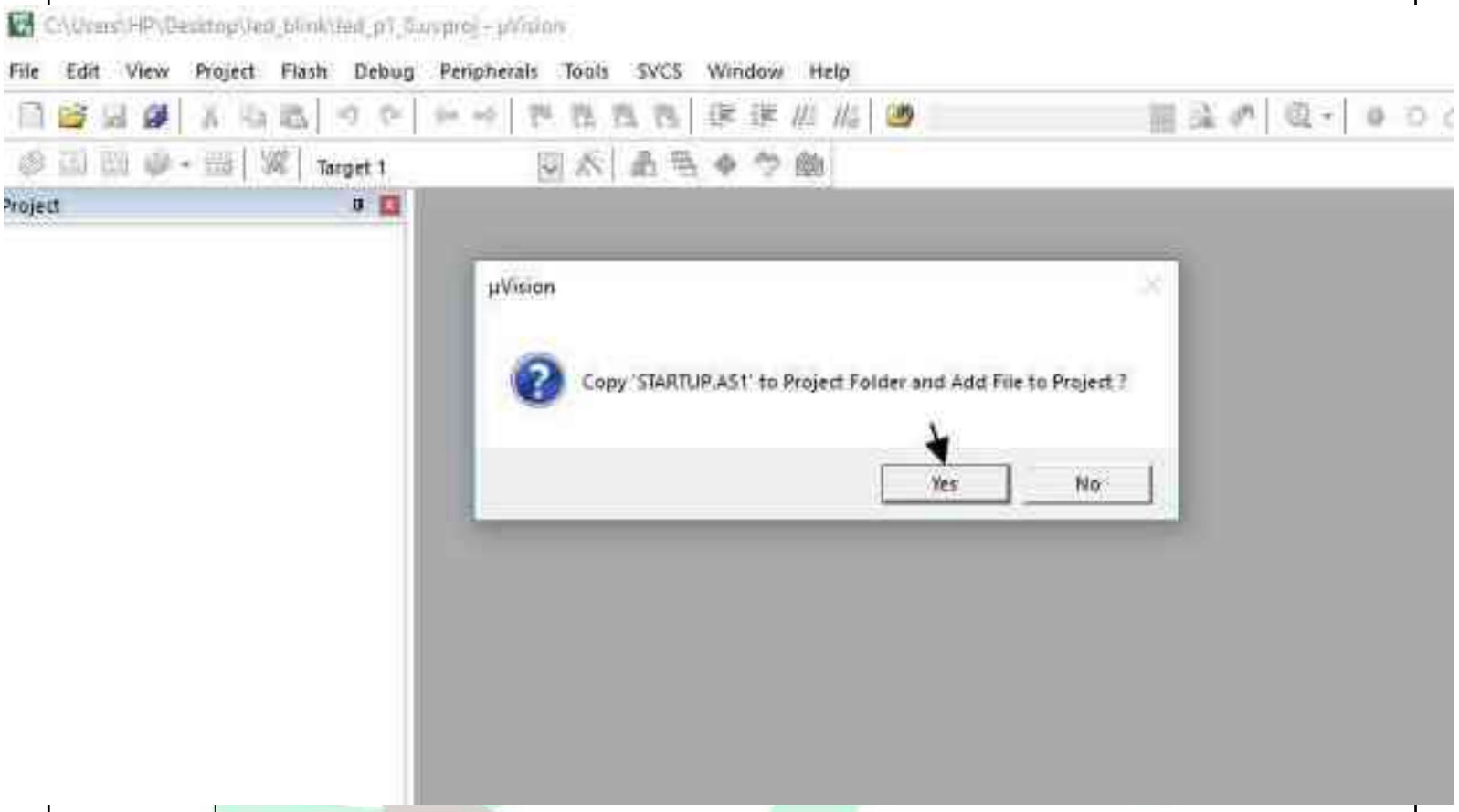
KEIL IMAGES

The creation of HEX file is similar to that of 89S52 using KEIL U-VISION software.

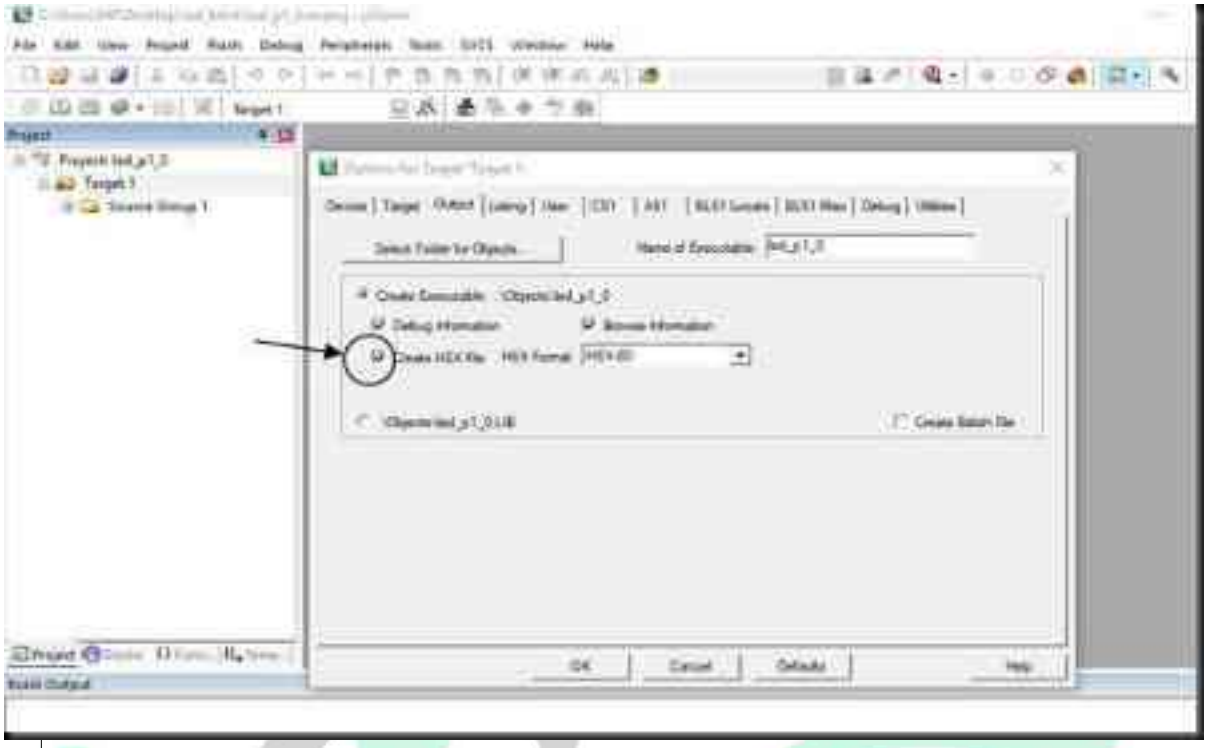
Open KEIL & create NEW Project.

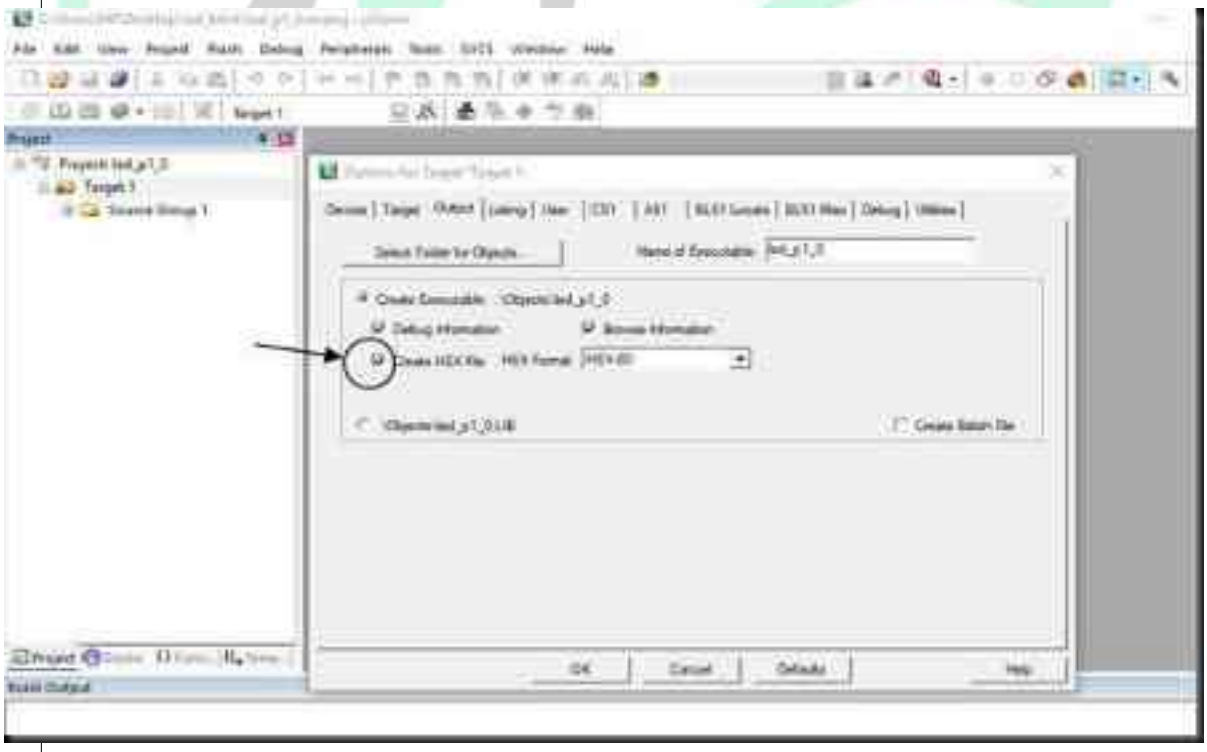
Select Device for target as W79E052D (selection of 89S52 also works).





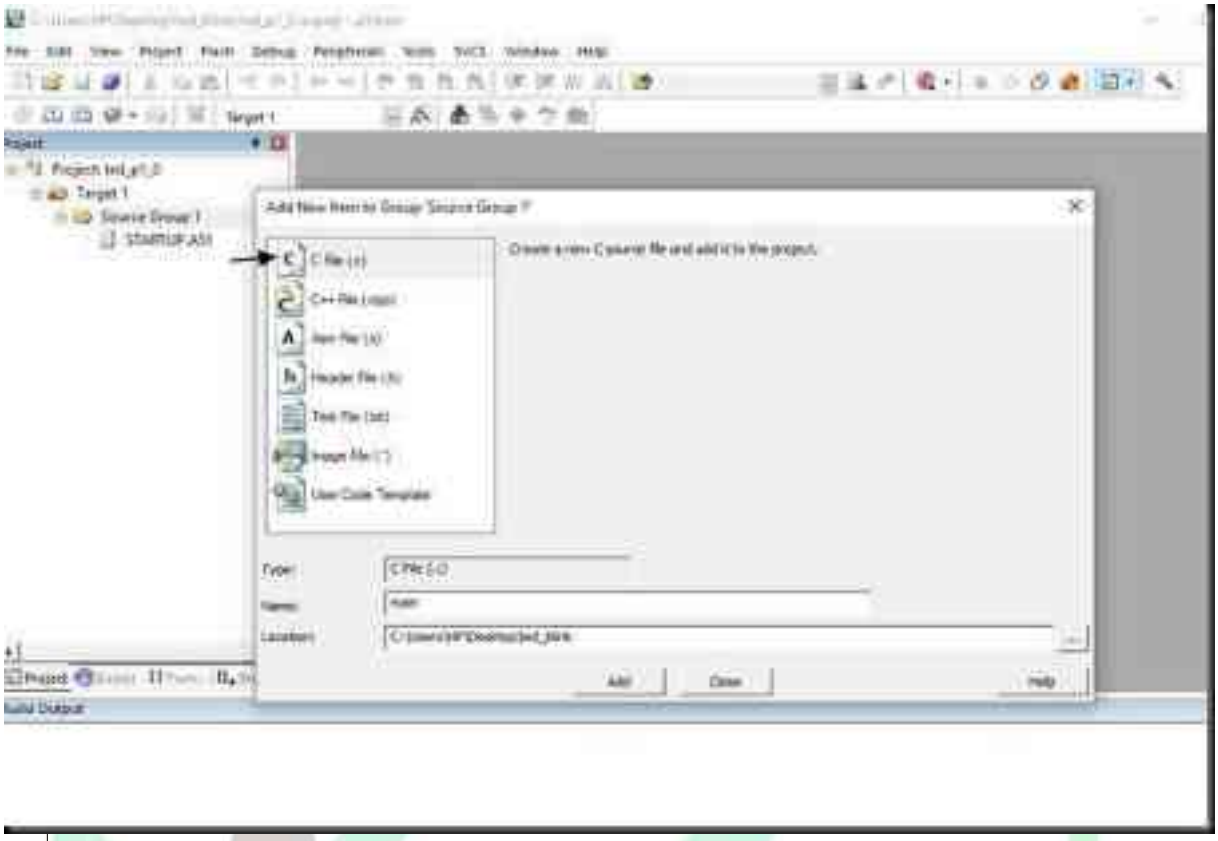
Right click Target & under options for Target click on OUTPUT tab & put a tick mark against CREATE HEX File.



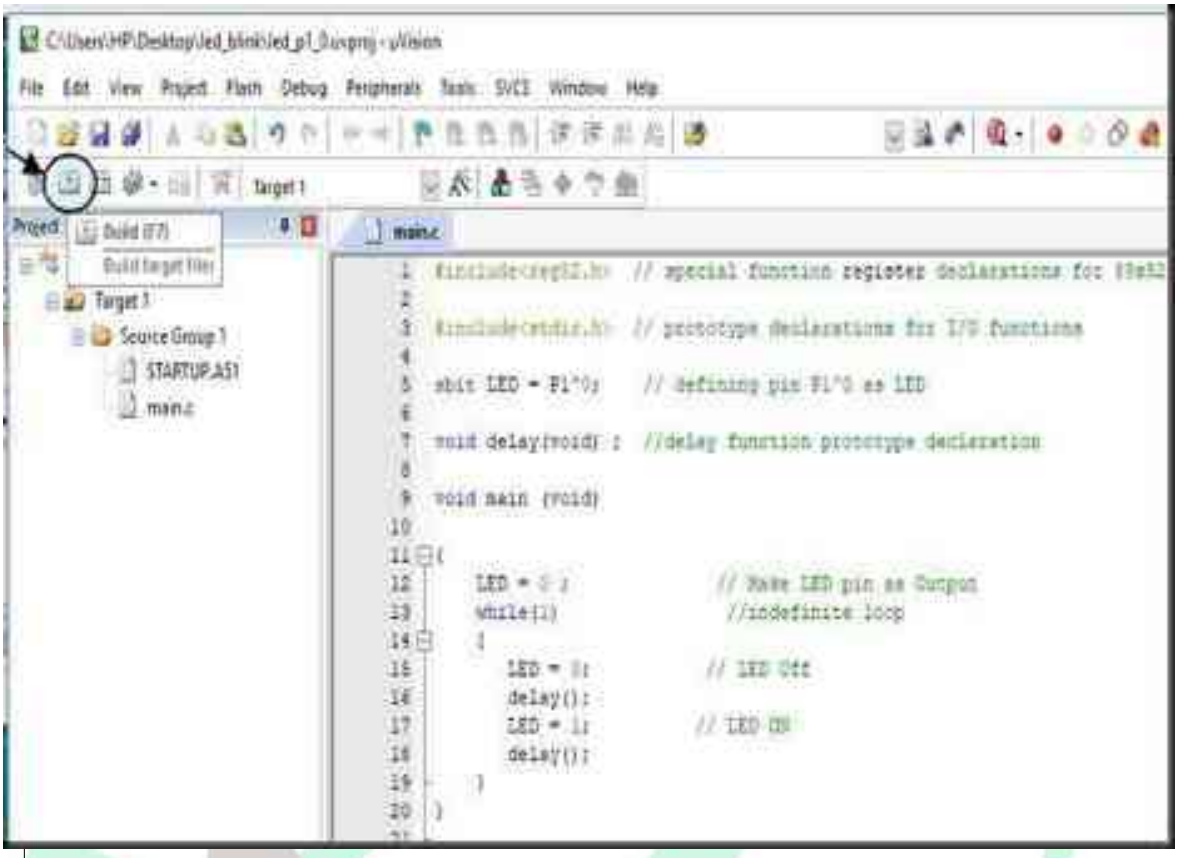


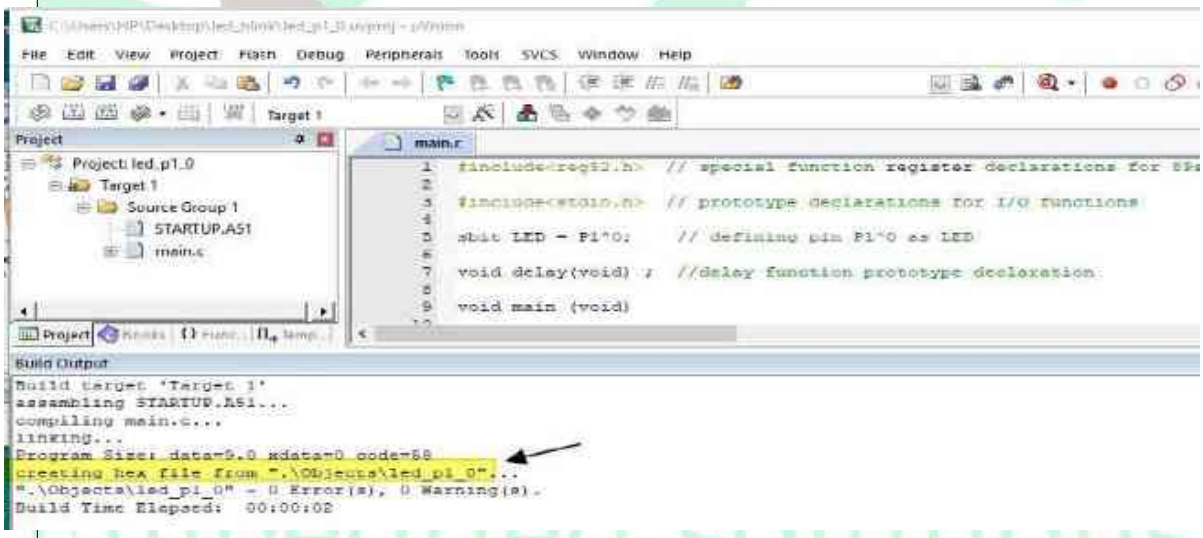
Right click SOURCE GROUP & click Add New item to Group.

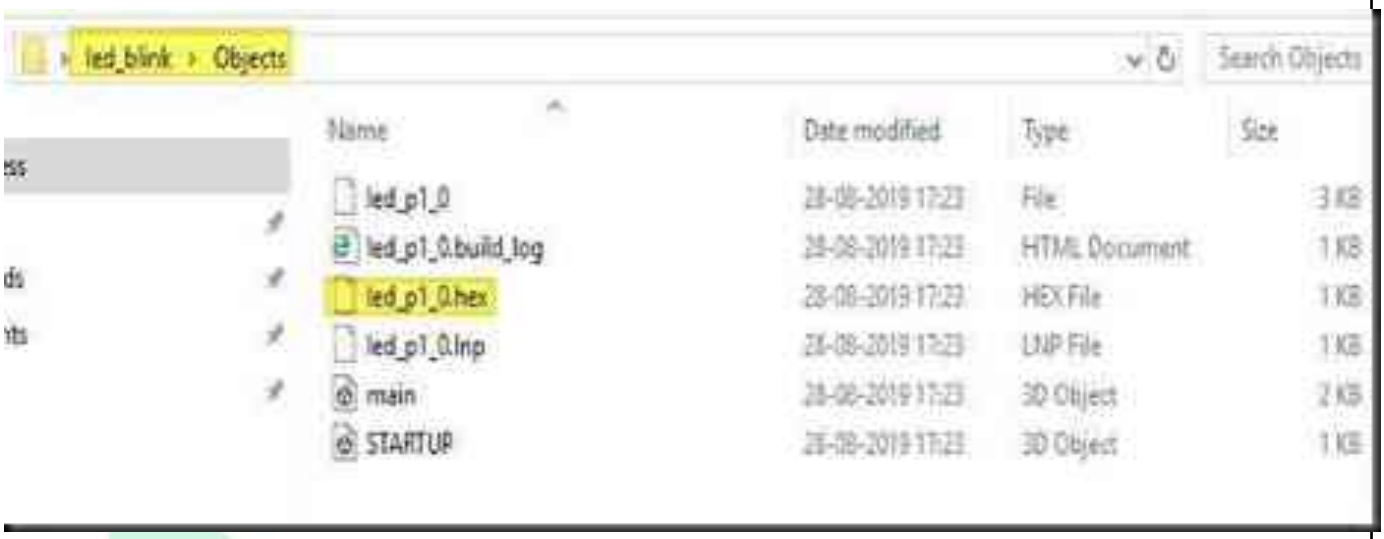
Select C File & provide a name for the file.



Now write your C code & click on BUILD or press F7 key

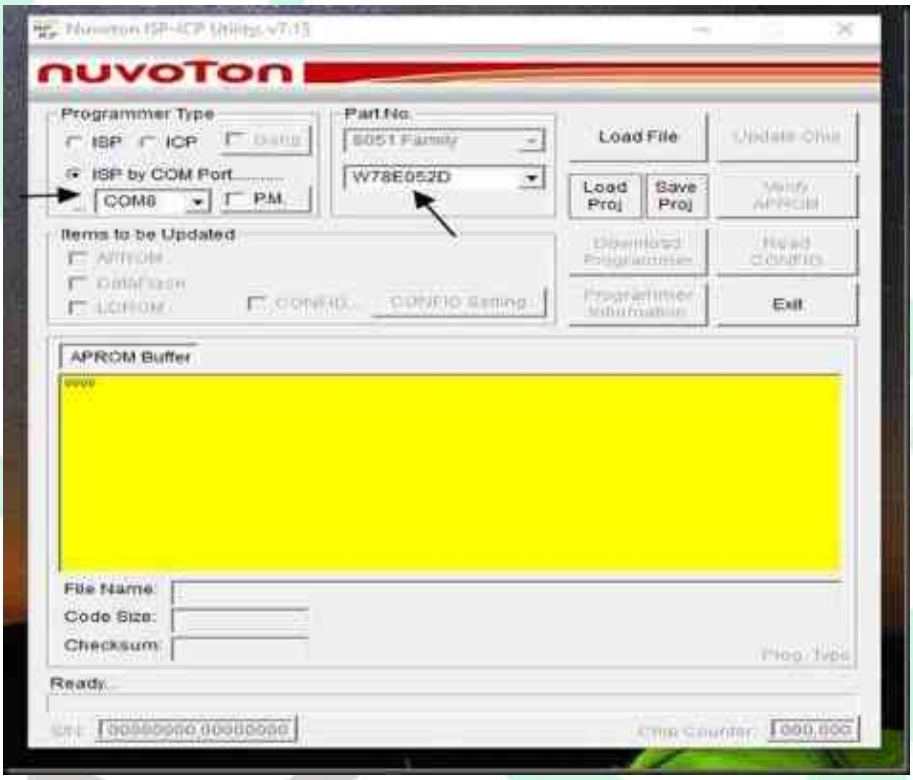




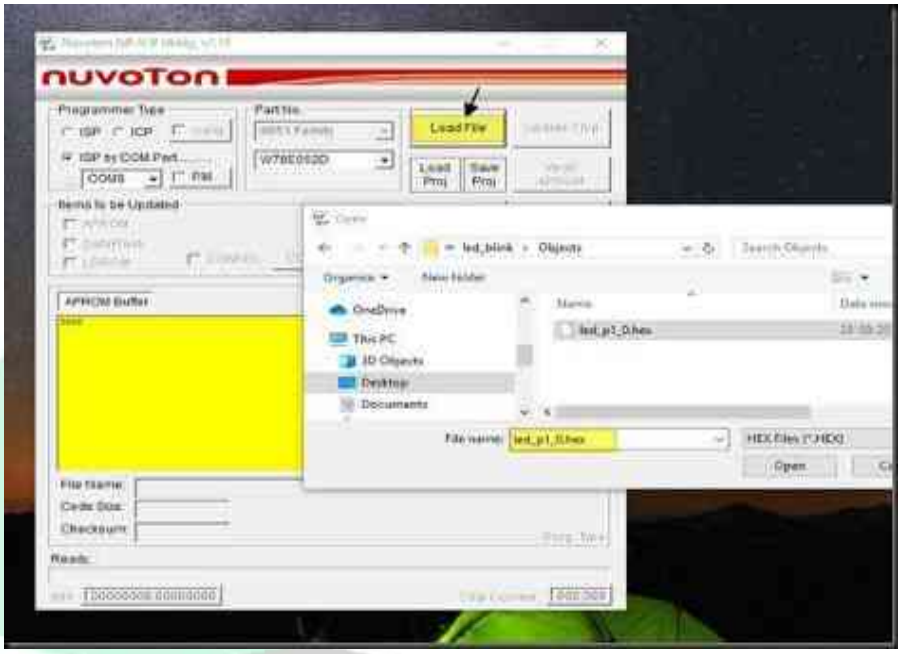


VI]NUVOTON ISP ICP PROGRAMMER

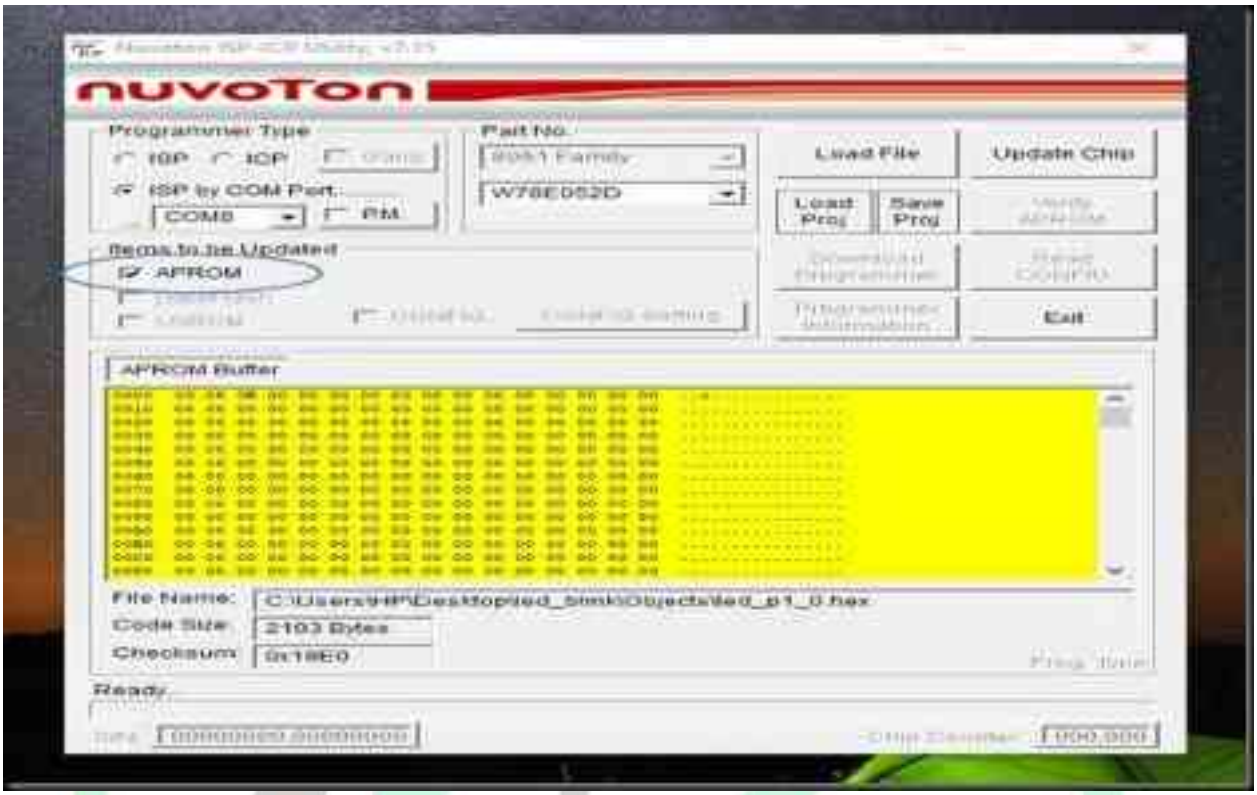
1) Select IC and Port

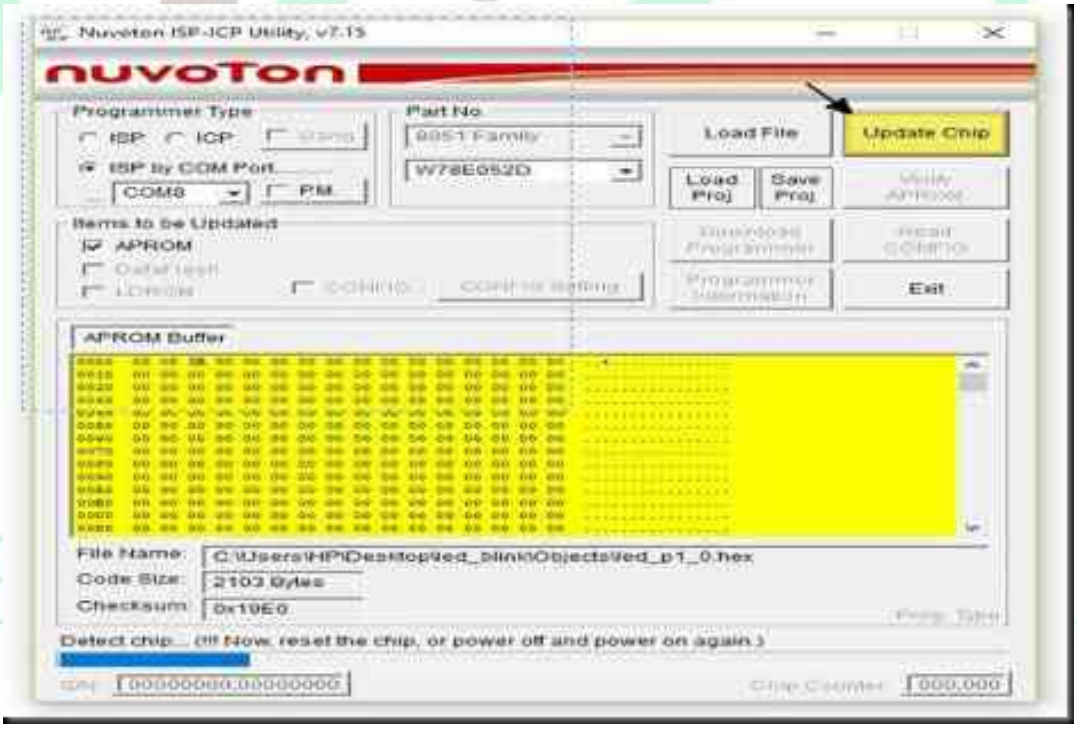


2)Load File



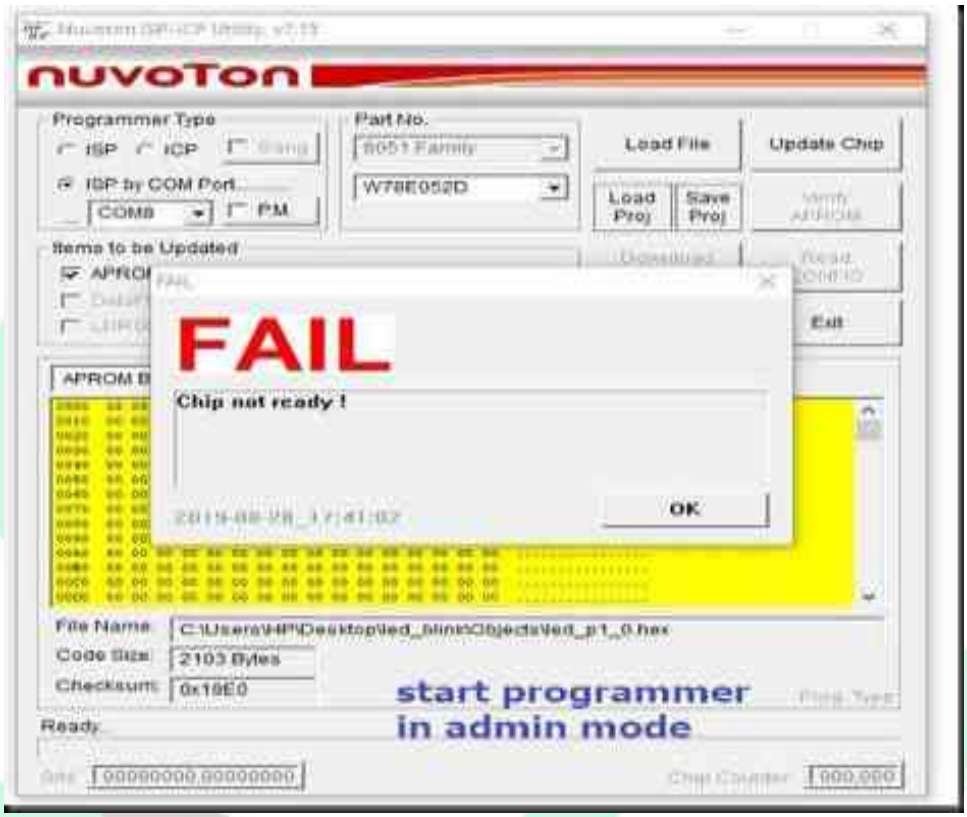
3)Select APROM



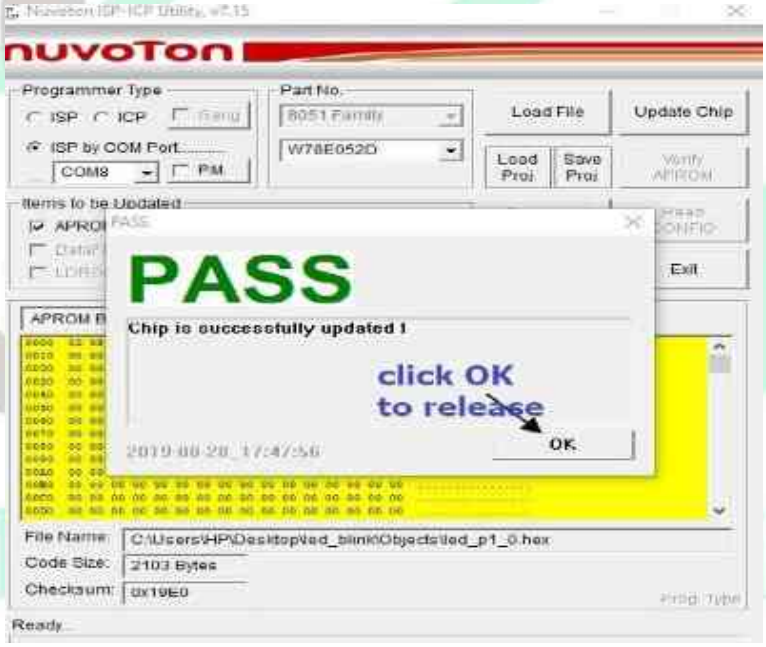


If RESET button is not pressed , you get FAIL report.

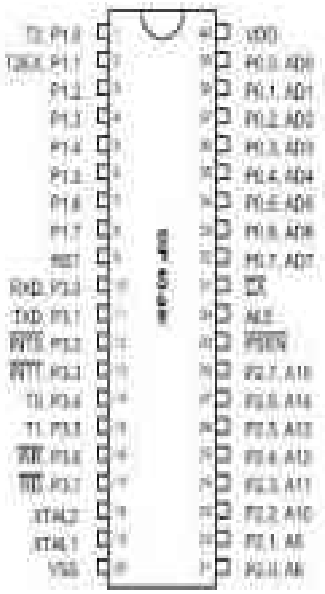
Also note that the NUVOTONE Utility must be started in ADMIN mode.

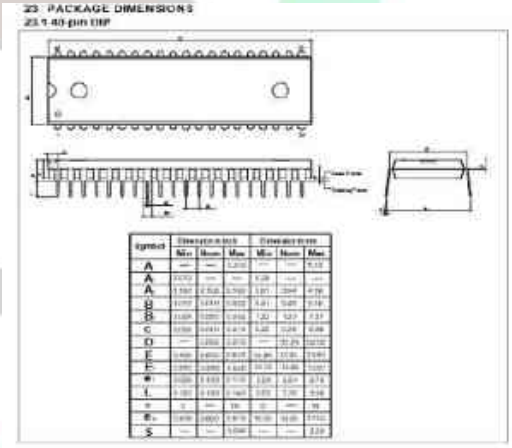



nuvoTon
Programmer Type
ISP
ICP
ISP by COM Port
COM8
Part No
8051 Family
W78E052D
Load File
Update Chip
Load Proj
Save Proj
Items to be Updated
Exit
PASS
Chip is successfully updated !
2019-08-28_17:47:56
OK
File Name:
C:\Users\HP\Desktop\led_blink\Objects\led_p1_0.hex
Code Size:
2103 Bytes
Checksum:
0x19E0
press RESET button
on board while uploading
Ready...


nuvoTon
Programmer Type
ISP
ICP
ISP by COM Port
COM8
Part No.
8051 Family
W78E052D
Load File
Update Chip
Load Proj
Save Proj
Items to be Updated
Exit
PASS
Chip is successfully updated !
click OK
to release
2019-08-28_17:47:56
OK
File Name:
C:\Users\HP\Desktop\led_blink\Objects\led_p1_0.hex
Code Size:
2103 Bytes
Checksum:
0x19E0
Ready...

Pin Configuration





The NuvoTon W78E052D controller Code downloading Steps

Download Nuvoton ISP-ICP Utility software.

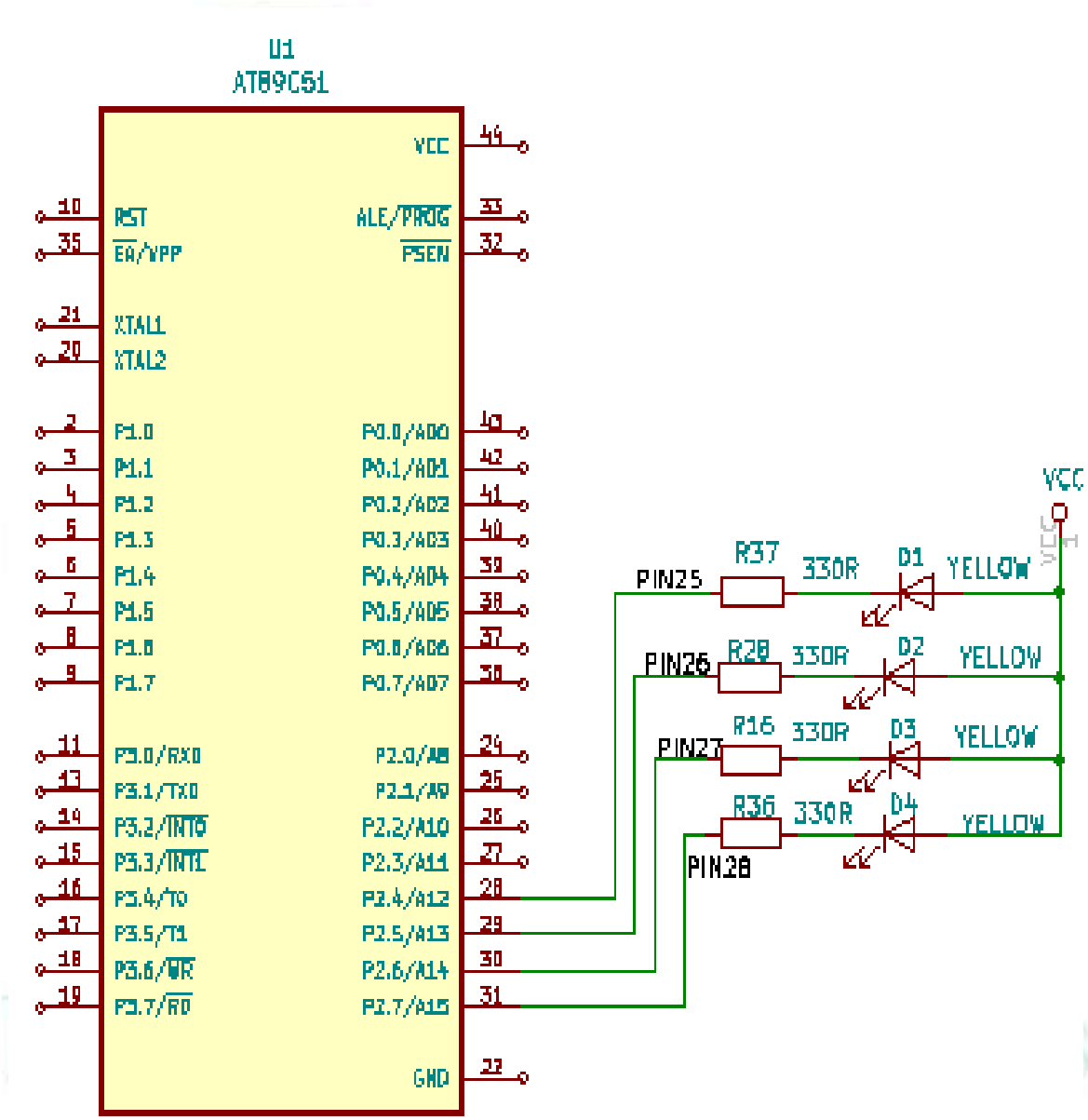
1. Now open the Nuvoton ISP-ICP Utility software and follow the below steps.
2. Select the ISP by COM port option for flashing the .hex file through a COM port.

3. Select the COM port from the drop-down. Check the device manager for the com port number.
4. Choose the required controller. In this case, it is W78E052D.
5. Browse and select by clicking on Load File.
6. Finally, click on Update chip to flash the .hex file then press the reset button from the board.

VII]EXAMPLES

EXPERIMENT NO.1

LED BLINKING



Lesson : 1

Neme : Led Blink

Details : 8 leds are connection to Port2 of the 8051 the Annode are common to VCC while the cathode are connected to each pin of the PORT2 the rate of blinking can be adjusted bychangin the delay rate

**********/

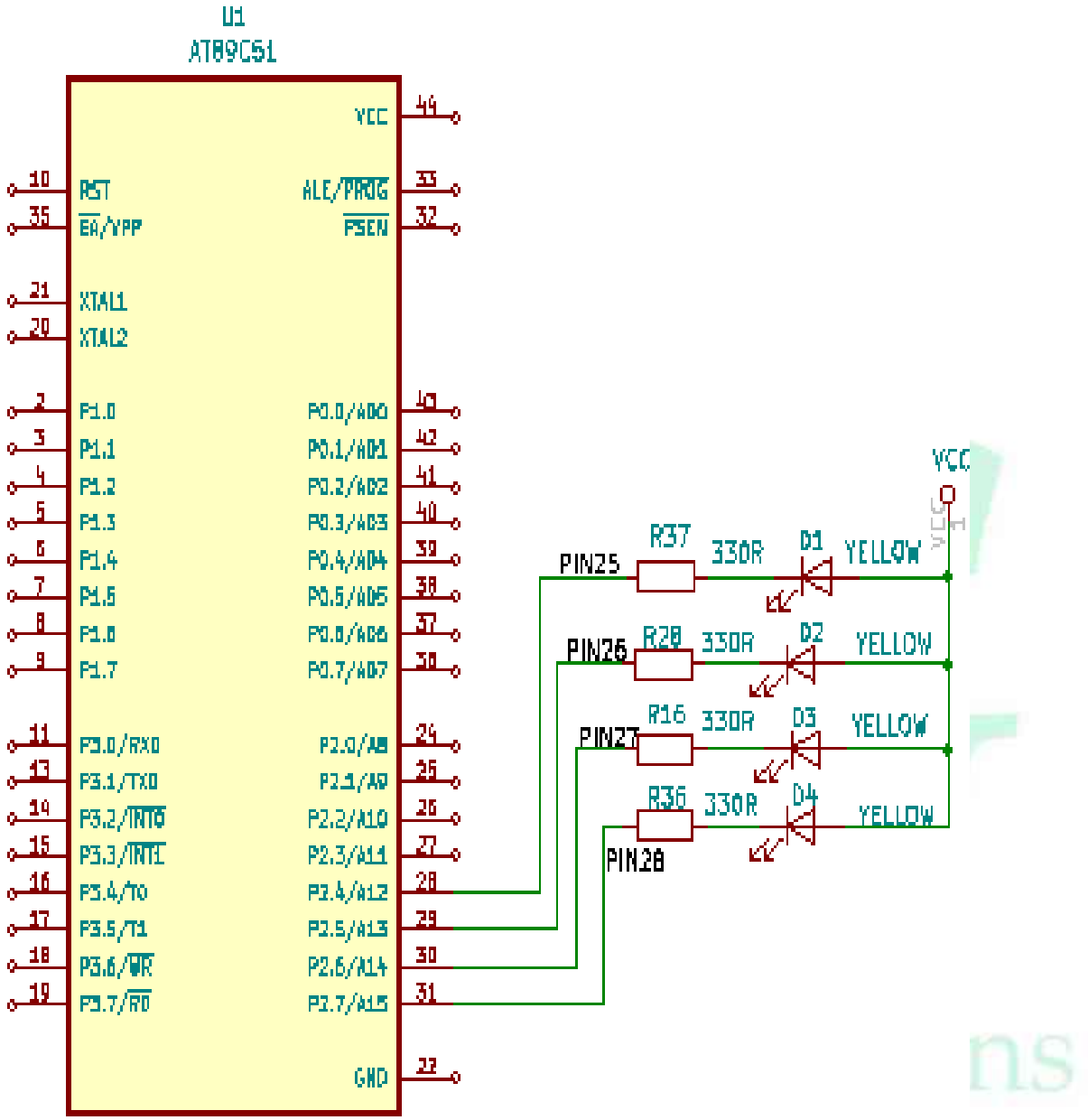
```
#include<REG51.H>
//Fuction Prototypes
void delay(void);

// Program Starts Here
void main()
{
        P2 = 0x00; // Leds are ON
        delay();
        P2 = 0xff; // Leds are OFF
        delay();
}

void delay(void)
{
        unsigned inta,b;
        for(a=0;a<1000;a++)
                {
                        for(b=0;b<120;b++)
```

```
                ;
            }
        }
```


Embedded solutions

EXPERIMENT NO.2

LED CHASER



Lesson : 2

Neme : Led chaser

Details : 8 leds are connection to Port2 of the 8051

the Annode are common to VCC while the cathode

are connected to each pin of the PORT2

the rate of chasing can be adjusted by

45hanging the delay rate

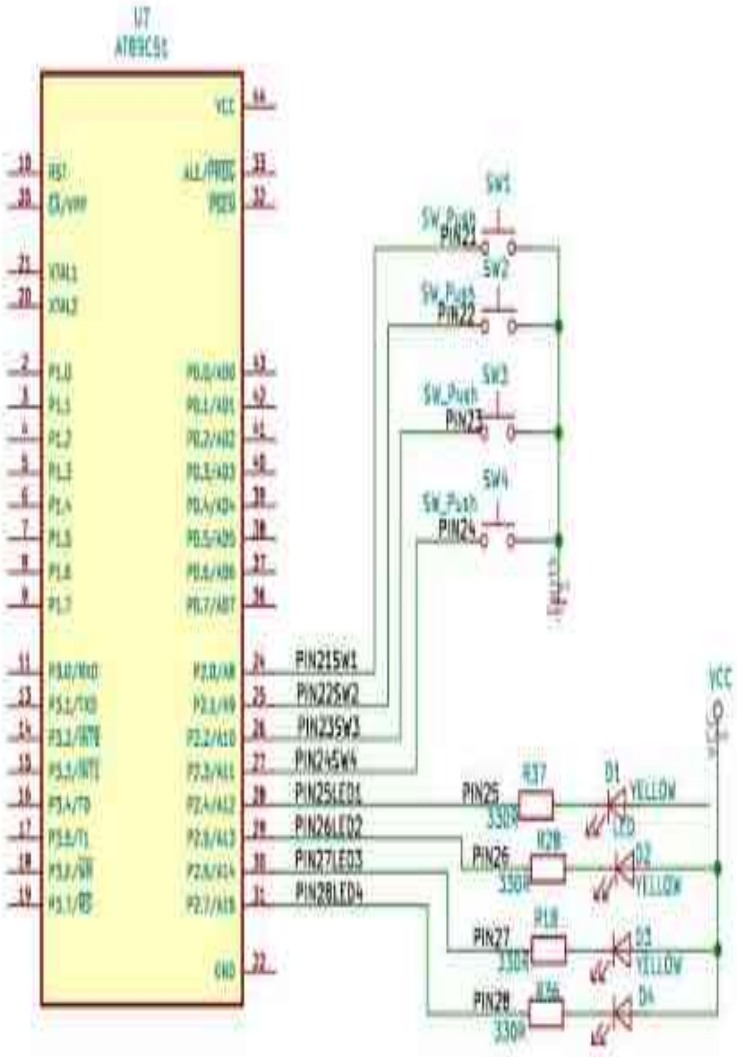
**********/

```
#include<REG51.H>
//Fuction Prototypes
void delay(void);

// Program Starts Here
void main()
{
        unsigned char a;
        a=0xef;

        while(1)
        {
         P2= a;
         a = a<<1;
         if (a==0xf0)
         a=0xef;
```

```
        delay();
        }
}

void delay(void)
{
        unsigned inta,b;
        for(a=0;a<1000;a++)
                {
                        for(b=0;b<50;b++)
                        ;
                }
```

EXPERIMENT NO.3

BUTTON TEST



Lesson : 3

Neme : Button Test

Details : 8 buttons are connected to port1,

the initial value of the port is HIGH(1)

when the button is prssed the value is LOW(0)

8 leds are connection to Port2 of the 8051

the Annode are common to VCC while the cathode are connected to each pin of the PORT2 when each button is pressed the same currospond led at port2 is glow.

**********/

```
#include<REG51.H>
//Fuction Prototypes
void delay(void);

// Program Starts Here
void main()
{
        P2 = 0Xff; // makes the Port 1 as input port
        while(1)
        {
           if((P2 & 0x0f) == 0x0e)
                {
                P2= (P2 & 0x0f)| 0xe0;
                delay();
                }
                else if ((P2 & 0x0f) == 0x0d)
                {
                P2=(P2 & 0x0f)|0xd0;
                delay();
                }
                else if ((P2 & 0x0f) == 0x0b)
                {
                P2=(P2 & 0x0f)|0xb0;
                delay();
                }
                else if ((P2 & 0x0f) == 0x07)
                {
                P2=(P2 & 0x0f)|0x70;
                delay();
```

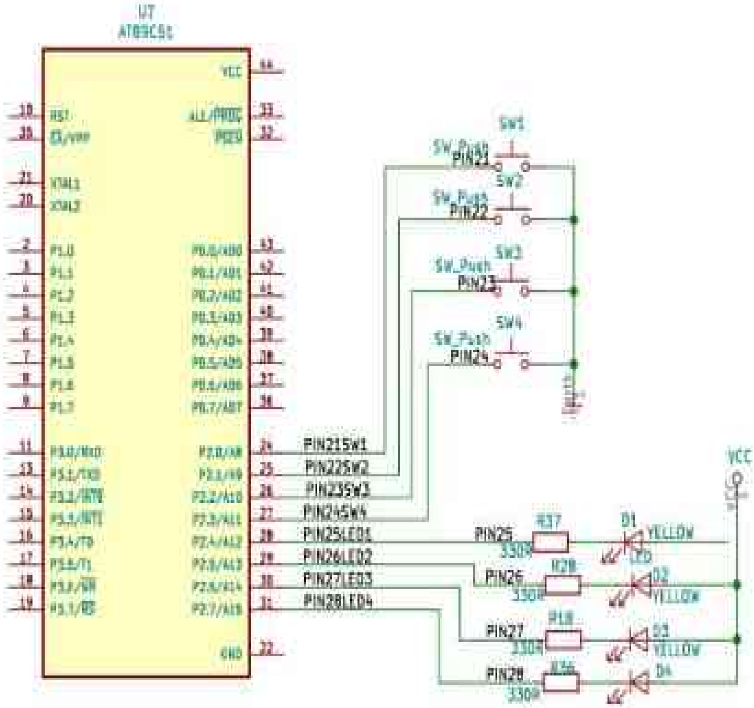
```
                }

                P2=0xff;
        }
}

void delay(void)
{
        unsigned inta,b;
        for(a=0;a<10;a++)
                {
                        for(b=0;b<120;b++)
                        ;
                }
}
```

Then click on OK

1) **OUPPUT show on kitAfter pressing 4 switch sequensely glow 4 LED sequensely**

EXPERIMENT NO.4

BUTTON MORE



Lesson : 4
Neme : Button More
Details : 8 buttons are connected to port1,
the initial value of the port is HIGH(1)
when the button is prssed the value is LOW(0)
8 leds are connection to Port2 of the 8051
the Annode are common to VCC while the cathode
are connected to each pin of the PORT2
when the switch 1 is pressed the each led is
inverted

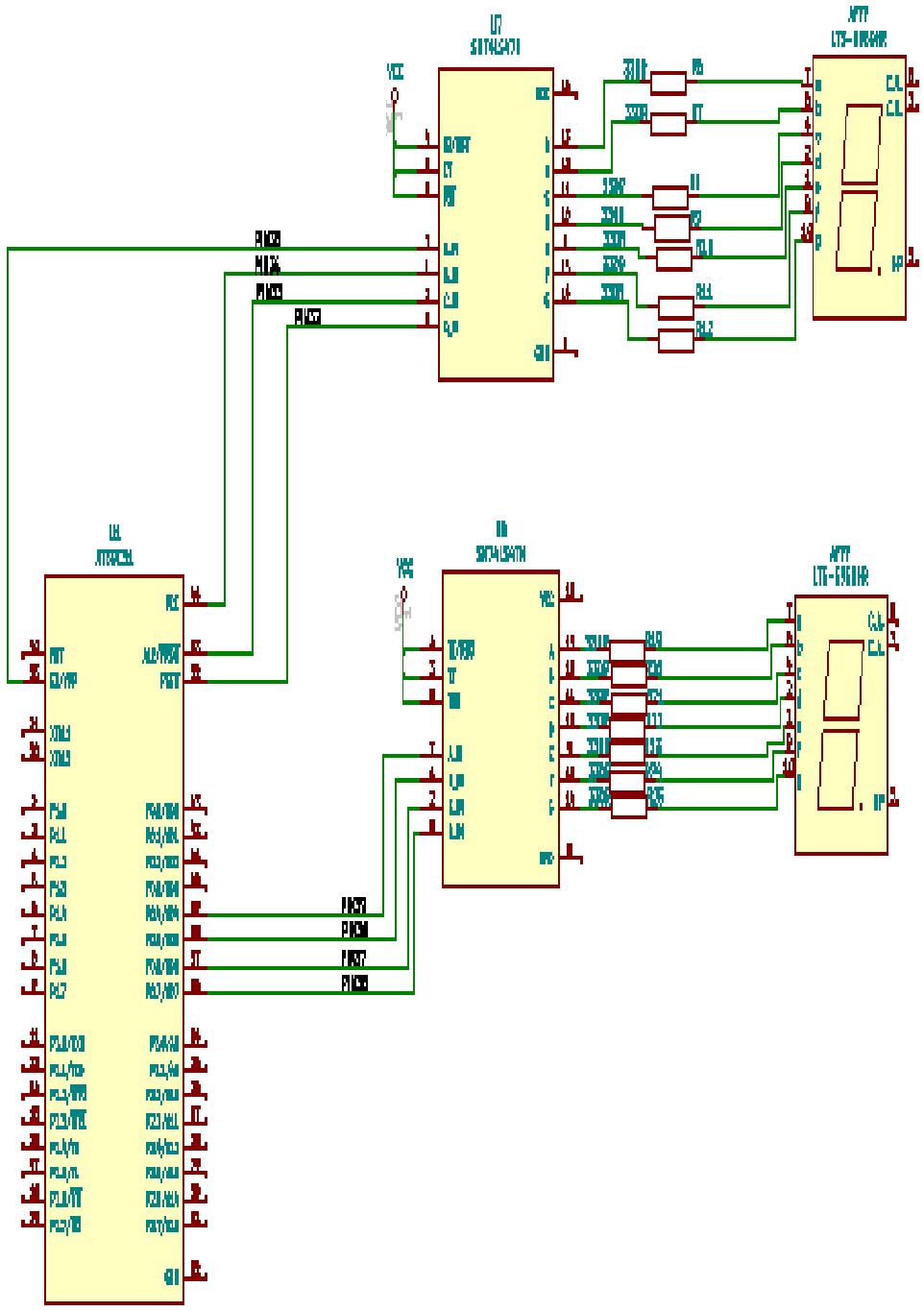
```
**************************************************************************/
#include<REG51.H>
//Fuction Prototypes
```

```
void delay(void);
sbit sw1=P2^0;

// Program Starts Here
void main()
{
    P2 = 0Xff; // makes the Port 1 as input port
    while(1)
    {
     if(sw1 == 0)        //checked the button is pressed or not
0=pressed, 1= not pressed
            {

                    P2=(P2 & 0x0f)|0x50;
                    delay();
            }

                    P2=(P2 & 0x0f)|0xaf;
    }
}

void delay(void)
{
    unsigned inta,b;
    for(a=0;a<100;a++)
            {
                    for(b=0;b<120;b++)
                    ;
            }
}
```

SSG
Embedded solutions

EXPERIMENT NO.5

SEVEN SEGMENT ROLLING



/***

Lesson : 5

Neme : seven segment

Details : A seven segment display is connected through a

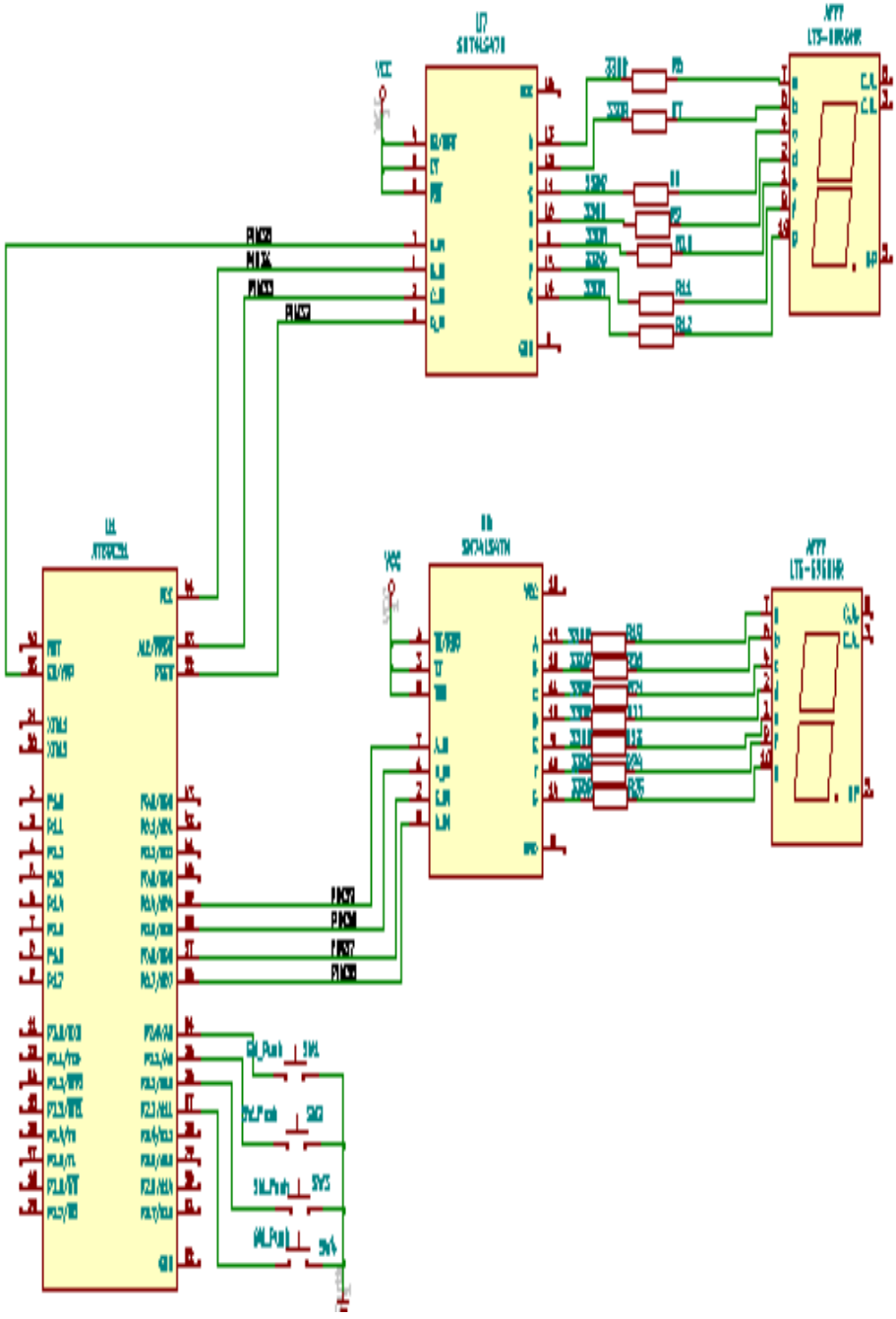
7447 ic which is bcd to 7 segment driver IC. The BCD inputs are connected to port3.4 to port3.7 of the 8051. The table for 1-9 is

```
-----------------------------
BCD  |      7 SEGMENT OUTPUT
-----------------------------
0000  |      0
0001  |      1
0010  |      2
0011  |      3
0100  |      4
0101  |      5
0110  |      6
0111  |      7
1000  |      8
1001  |      9
*************************************************/
#include<REG51.H>
#pragma
//Fuction Prototypes
void delay(void);

// Program Starts Here
void main()
{
        unsigned char a,b;
        while(1)
        {
                for(a=0;a<99;a++)
                {
                 b=a>>4;
                 b |= a<<4;
                 P0 =b ;
                 delay();
                }
```

```
        }
}

void delay(void)
{
        unsigned inta,b;
        for(a=0;a<1000;a++)
                {
                        for(b=0;b<120;b++)
                        ;
                }
}
```

EXPERIMENT NO. 6

SEVEN SEGMENT WITH BUTTON



/**

Lesson : 6

Neme : seven segment

Details : A seven segment display is connected through a 7447 ic which is bcd to 7 segment driver IC. The BCD inputs are connected to port3.4 to port3.7 of the 8051. The table for 1-9 is

BCD	7 SEGMENT OUTPUT
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9

When each button is pressed the display shows the no of pressed button

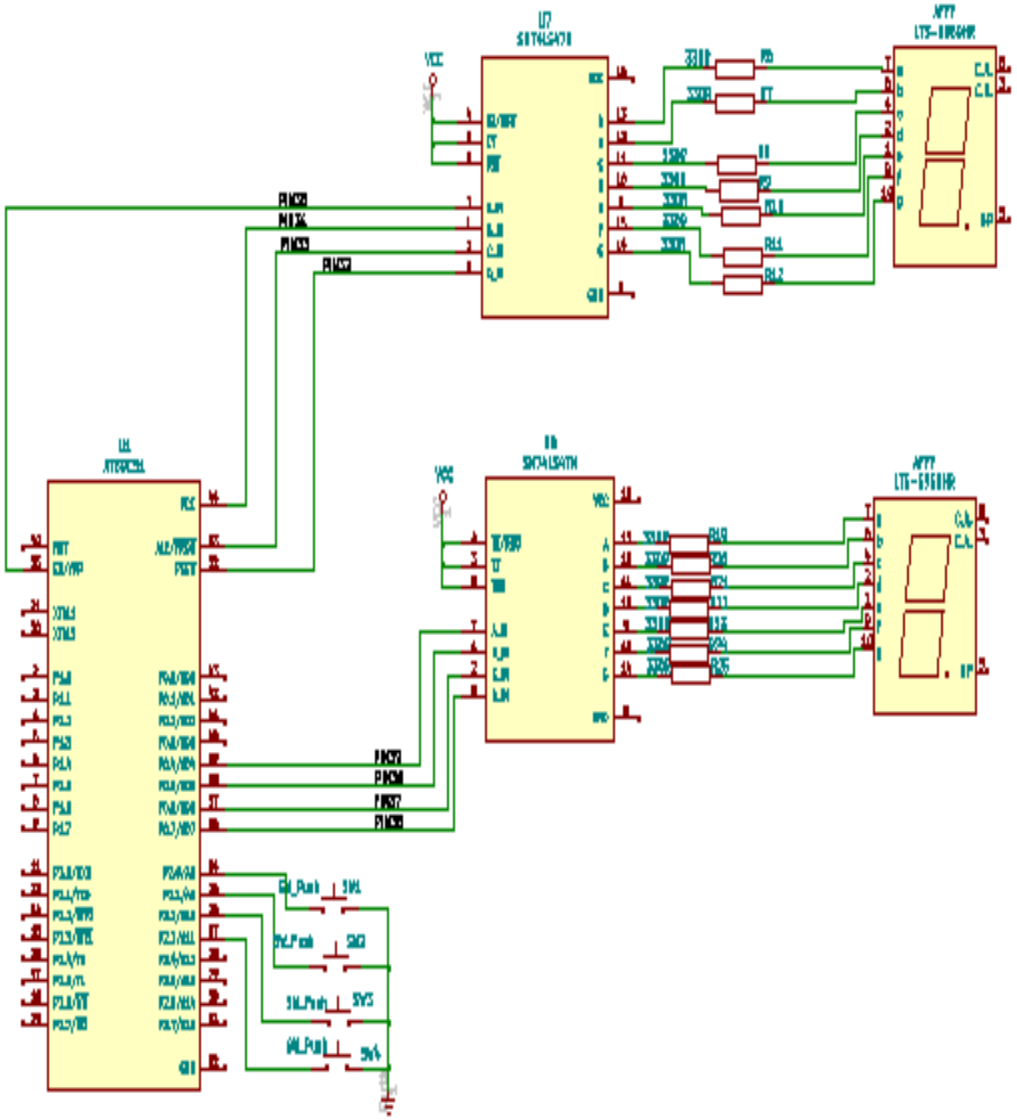
```
**************************************************/
#include<REG51.H>
//Fuction Prototypes
void delay(void);

// Program Starts Here
void main()
{
   P2=0xff;
```

```
P0=0X00;
while(1)
{
        if((P2 & 0x0f) == 0x0e)
        {
        P0= 0x10;
        delay();
        }
        else if ((P2 & 0x0f) == 0x0d)
        {
        P0=0x20;
        delay();
        }
        else if ((P2 & 0x0f) == 0x0b)
        {
        P0=0x30;
        delay();
        }
        else if ((P2 & 0x0f) == 0x07)
        {
        P0=0x40;
        delay();
        }
        else
        P0=0x00;
    }
}

void delay(void)
{
```

```
unsigned inta,b;
for(a=0;a<10;a++)
        {
                for(b=0;b<120;b++)
                ;
}
}
```

EXPERIMENT NO. 7

SEVEN SEGMENT UP AND DOWN



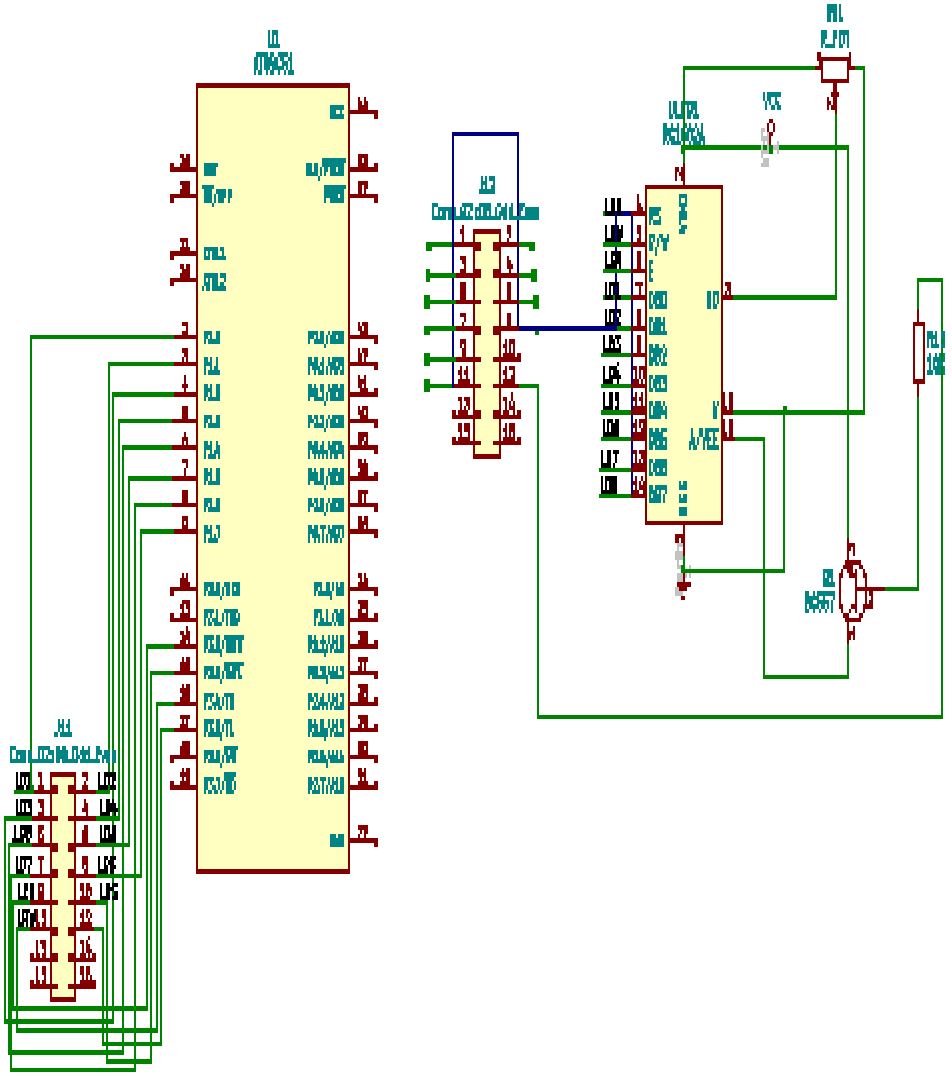
```
/*************************************************
Lesson  : 7
Neme    : seven segment
Details : A seven segment display is connected through a
          7447 ic which is bcd to 7 segment driver IC.
          The BCD inputs are connected to port3.4 to
          port3.7 of the 8051. The table for 1-9 is
```

```
-----------------------------
BCD  |      7 SEGMENT OUTPUT
-----------------------------
0000  |      0
0001  |      1
0010  |      2
0011  |      3
0100  |      4
0101  |      5
0110  |      6
0111  |      7
1000  |      8
1001  |      9
            WHEN SW1 IS PRESSED THE COUNTER IN
INCREMENTED
            WHEN SW2 IS PRESED THE COUNTER IS
DECREMENTED
**********************************************************/
#include<REG51.H>
//Fuction Prototypes
void delay(void);
sbit sw1=P2^0;
sbit sw2=P2^1;
void delayms(unsigned inttt);
// Program Starts Here
void main()
{
    unsigned char a,b;
    P2=0xff;
    a=0;
    while(1)
    {
    if(sw1 == 0)
            {
            delayms(20);
```

```
                while(sw1==0);
                a++;
                b=a>>4;
                b |= a<<4;

        }
        else if (sw2 == 0)
        {
                delayms(20);
                while(sw2==0);
                if(a>0)
                a--;
                b=a>>4;
                b |= a<<4;

        }
        P0 = b ;
        }
}
void delayms(unsigned inttt)
{
    unsigned inta,b;
    for(a=0;a<tt;a++)
                {
                for(b=0;b<1275;b++)
                ;
}
}
```

EXPERIMENT NO. 8

LCD SIMPLE



/***

Lesson : 8

Neme : LCD Interface

Details : The LCD has two line and each line has 16 character

the lcd works here 4 bit mode

the display show “welcome” on first line

and “have a nice day” on second line

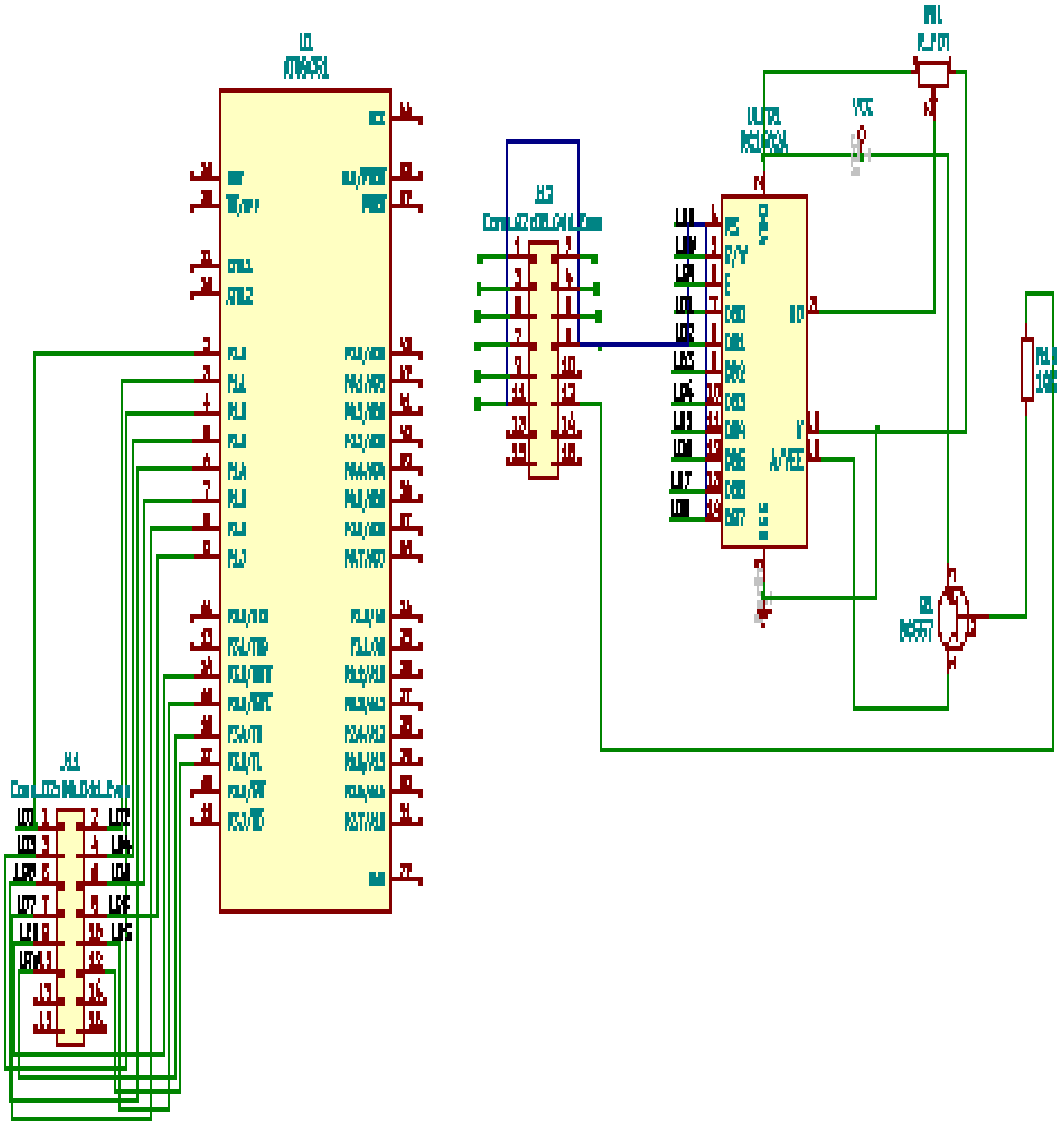
```
**************************************************/
#include<REG51.H>
//Fuction Prototypes
void delay(unsigned int);
void lcd_string(unsigned char *p);
void lcd_data(unsigned char );
void lcd_cmd(unsigned char );


sbit EN = P3^2;
sbit RS = P3^3;
sbit RW = P3^4;
sbit light = P3^5;

#define lcddata P1;
// Program Starts Here
void main()
{

    light=0; //ON the lcd light
    lcd_cmd(0x38);
    delay(1);
    lcd_cmd(0x06);      // display move cursor to right
    delay(1);
    lcd_cmd(0x0E);      // LCD On, & cursor        on
    delay(1);
    lcd_cmd(0x01);      // LCD Clear
    delay(1);
    lcd_cmd(0x80);      // LCD start 1st line
    delay(1);
    lcd_string("   WELCOME    ");
    delay(25);
    lcd_cmd(0xc0);      // LCD start 2nd line
    delay(1);
```

```
lcd_string("HAVE A NICE DAY ");
      while(1);

}

void lcd_string(unsigned char *p)
{
    while(*p != '\0')
    {
    lcd_data(*p);
    p++;
    delay(10);
    }
}

void lcd_data(unsigned char x)
{
 RW=0;
 RS=1;
 P1 = x;
 EN=1;
 delay(1);
 EN=0;
 }

void lcd_cmd(unsigned char z)
{
 RW=0;
 RS=0;
 P1 = z;
 EN =1;
 delay(1);
 EN=0;
 }
```

```
void delay(unsigned inttt)
{
    unsigned inta,b;
    for(a=0;a<tt;a++)
            {
                    for(b=0;b<1275;b++)
                    ;
            }
}
```

EXPERIMENT NO. 9

LCD FUNCTION



```
/*************************************************************
  Lesson  : 9
  Neme    : LCD FUCTIONS
```

Details : The LCD has two line and each line has 16 character
the lcd works here 4 bit mode
the display is show various fuction of lcd

```
**************************************************/
#include<REG51.H>
//Fuction Prototypes
void delay(unsigned int);
void lcd_string(unsigned char *p);
void lcd_data(unsigned char );
void lcd_cmd(unsigned char );


sbit EN = P3^2;
sbit RS = P3^3;
sbit RW = P3^4;
sbit light = P3^5;

#define lcddata P1;

// Program Starts Here
void main()
{
     light=0;
    lcd_cmd(0x38);       //LCD 4 bit Mode
     delay(1);
    lcd_cmd(0x06);       // display move cursor to right
     delay(1);
    lcd_cmd(0x0C);       // LCD On, & cursor       OFF
     delay(1);
    lcd_cmd(0x01);        // LCD Clear
     delay(1);
    lcd_cmd(0x80);       // LCD start 1st line
     delay(1);
    lcd_string("    HELLO    ");
     delay(100);
```

```
        lcd_cmd(0xc0);        // LCD start 2nd line
         delay(1);
    lcd_string(" AND WELCOME ");
         delay(400);
         while(1)
         {
        lcd_cmd(0x01);        // LCD Clear
         delay(1);
        lcd_string("LCD FUCTIONS   ");
         delay(400);
        lcd_cmd(0x01);        // LCD Clear
         delay(1);
        lcd_string("FIRST LINE   ");
         delay(1);
         delay(400);
        lcd_cmd(0x01);        // LCD Clear
         delay(1);
        lcd_cmd(0xc0);        // LCD start 2nd line
         delay(1);
        lcd_string("SECOND LINE   ");
          delay(400);
        lcd_cmd(0x01);        // LCD Clear
         delay(1);
        lcd_string("CURSOR ON   ");
         delay(1);
        lcd_cmd(0xc0);        // LCD start 2nd line
        lcd_cmd(0x0E);        // CURSOR ON
         delay(500);
        lcd_cmd(0x01);        // LCD Clear
         delay(1);
        lcd_string("CURSOR OFF  ");
         delay(1);
        lcd_cmd(0xc0);        // LCD start 2nd line
        lcd_cmd(0x0C);        // CURSOR OFF
         delay(500);
```

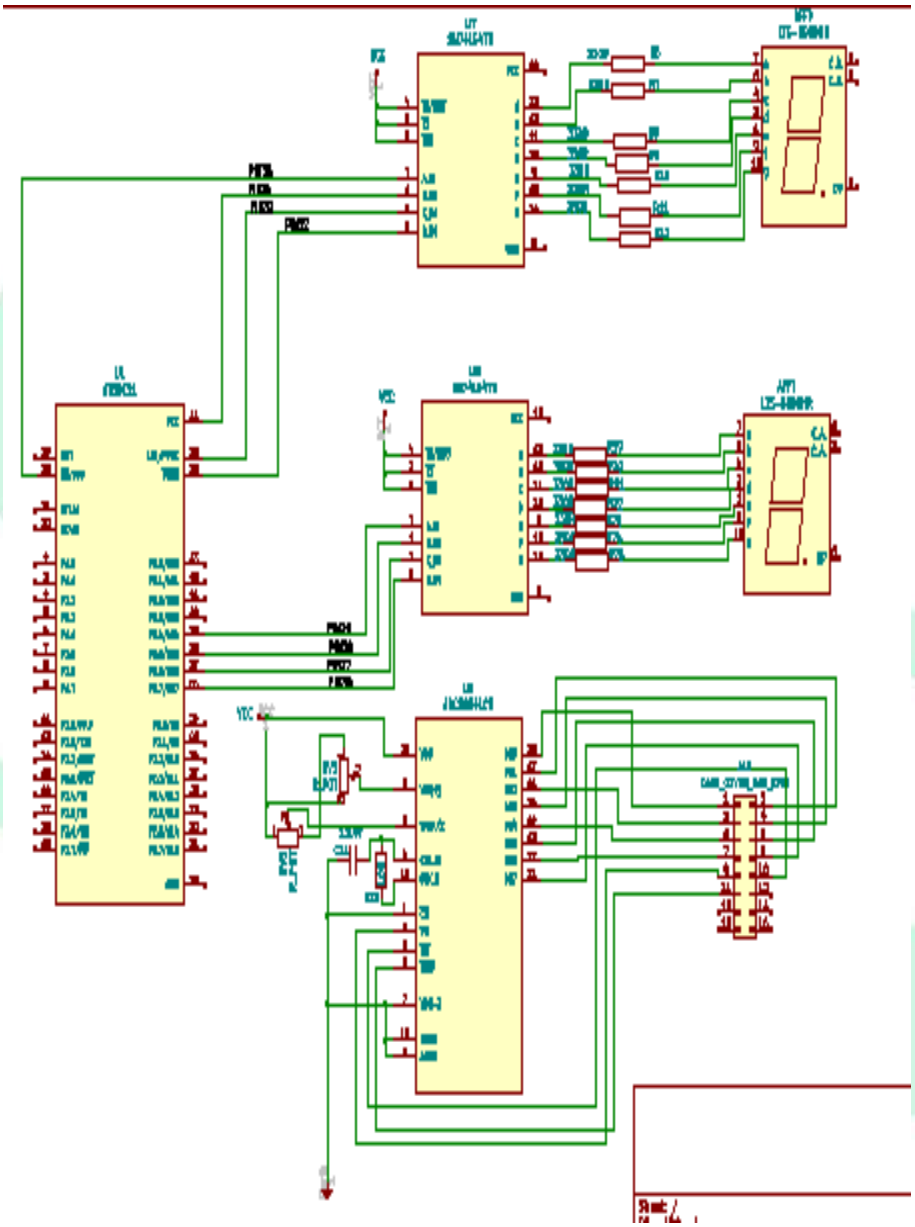
```
lcd_cmd(0x01);   // LCD Clear
     delay(1);
    lcd_string("CURSOR BLINK");
     delay(1);
    lcd_cmd(0xc0);        // LCD start 2nd line
    lcd_cmd(0x0F);        // CURSOR BLINK
     delay(500);
    lcd_cmd(0x01);        // LCD Clear
     delay(1);
    lcd_string("CURSOR  RIGHT");
     delay(1);
    lcd_cmd(0xc0);        // LCD start 2nd line
    lcd_cmd(0x14);        // CURSOR RIGHT
     delay(100);
    lcd_cmd(0x14);        // CURSOR RIGHT
     delay(100);
    lcd_cmd(0x14);        // CURSOR RIGHT
     delay(100);
    lcd_cmd(0x14);        // CURSOR RIGHT
     delay(100);
    lcd_cmd(0x14);        // CURSOR RIGHT
     delay(100);
     delay(500);
    lcd_cmd(0x01);        // LCD Clear
     delay(1);
    lcd_string("CURSOR  LEFT");
     delay(1);
    lcd_cmd(0xc5);        // LCD start 2nd line
    lcd_cmd(0x10);        // CURSOR LEFT
     delay(100);
    lcd_cmd(0x10);        // CURSOR LEFT
     delay(100);
    lcd_cmd(0x10);        // CURSOR LEFT
     delay(100);
    lcd_cmd(0x10);        // CURSOR LEFT
```

```
        delay(100);
        lcd_cmd(0x10);        // CURSOR LEFT
        delay(100);
        delay(500);
        lcd_cmd(0x01);        // LCD Clear
        lcd_cmd(0x0C);        // CURSOR OFF
        delay(1);
        lcd_string("LCD OFF");
        delay(200);
        light=1;
        lcd_cmd(0x0A);        // LCD OFF
        delay(500);
        lcd_cmd(0x01);        // LCD Clear
        delay(1);
        lcd_string("LCD ON");
        delay(1);
        light=0;
        lcd_cmd(0x0C);        // LCD ON
        delay(500);
      }

}

void lcd_string(unsigned char *p)
{
    while(*p != '\0')
    {
    lcd_data(*p);
    p++;
    }
}

void lcd_data(unsigned char x)
{
 RW=0;
```

```
 RS=1;
 P1 = x;
 EN=1;
 delay(1);
 EN=0;
}

void lcd_cmd(unsigned char z)
{
 RW=0;
 RS=0;
 P1 = z;
 EN =1;
 delay(1);
 EN=0;
}


void delay(unsigned inttt)
{
    unsigned inta,b;
    for(a=0;a<tt;a++)
            {
                    for(b=0;b<1000;b++)
                    ;
            }
}
```


EXPERIMENT NO. 10

ADC

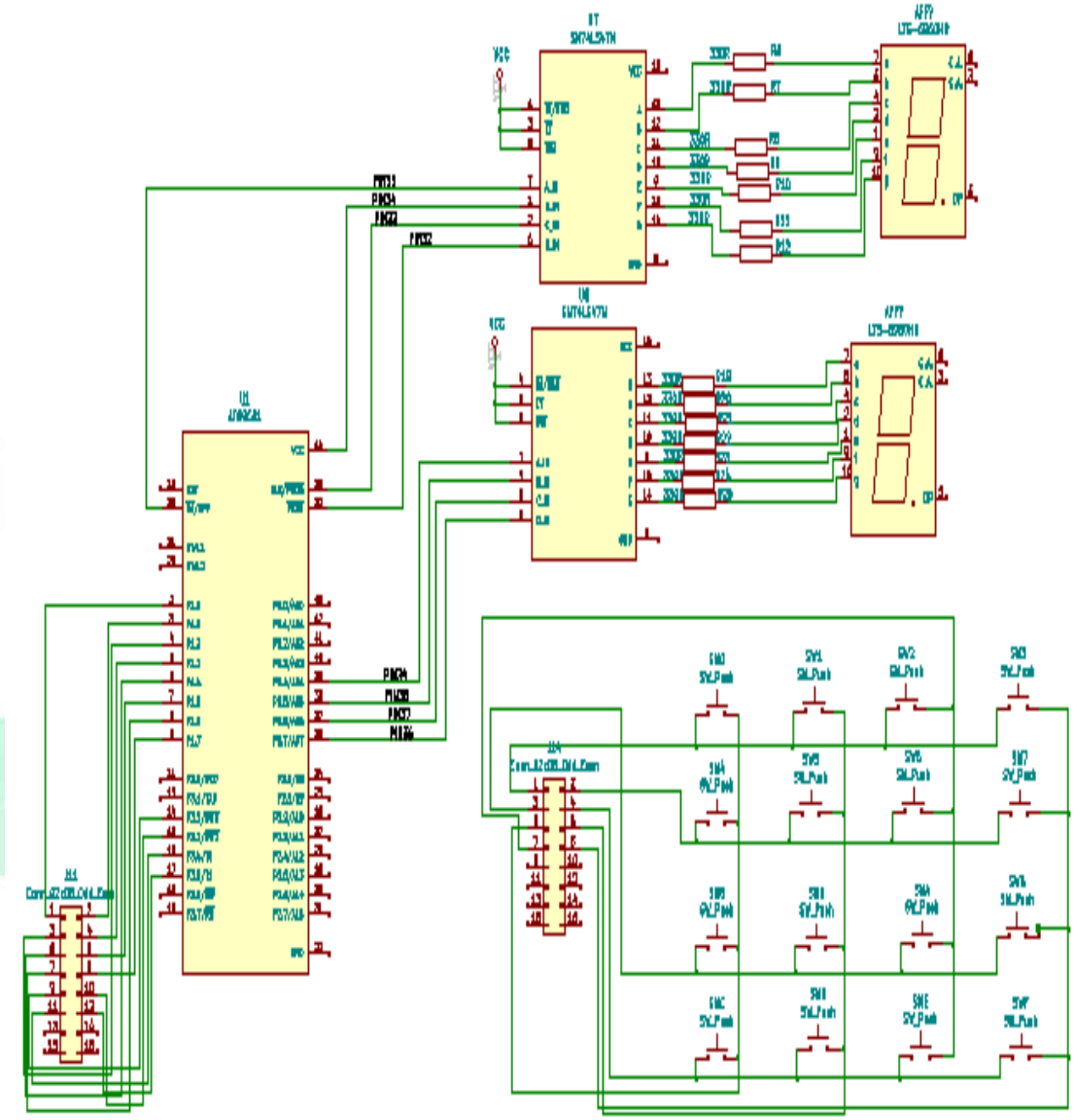


```
#include <at89c51xd2.h>
//ADC
sbitrd = P3^2;
sbitwr = P3^3;
sbitintr = P3^4;
// DELAY DEFINATIONS
void delay(void);
void show(unsigned char);
void main()
        {
                unsigned char a,adc;
                P1=0xff;
                intr=1;
                rd=1;
                wr=1;
                while(1)
                        {

                                adc=0;
                                for(a=0;a<5;a++)
                                {
                                wr=0;
                                wr=1;
                                while(intr);
                                rd=0;
                                adc =P1;
                                }
                                show(adc);
                                rd=1;
                                delay();

                        }
                }
```

```
void show(unsigned char key)
{
        unsigned char b;
        b= key >> 4;
        b |= key << 4;
        P0=b;
}

void delay(void)
{
        unsigned inta,b;
        for(a=0;a<1000;a++)
         for(b=0;b<120;b++);
}
```

EXPERIMENT NO. 11

MATRIX KEYBOARD



```
#include <at89c51xd2.h>
void show(unsigned char);
void delay(void);
sbit col1= P1^7;
sbit col2= P1^6;
sbit col3= P1^5;
sbit col4= P1^4;
sbit row1= P1^0;
sbit row2= P1^1;
```

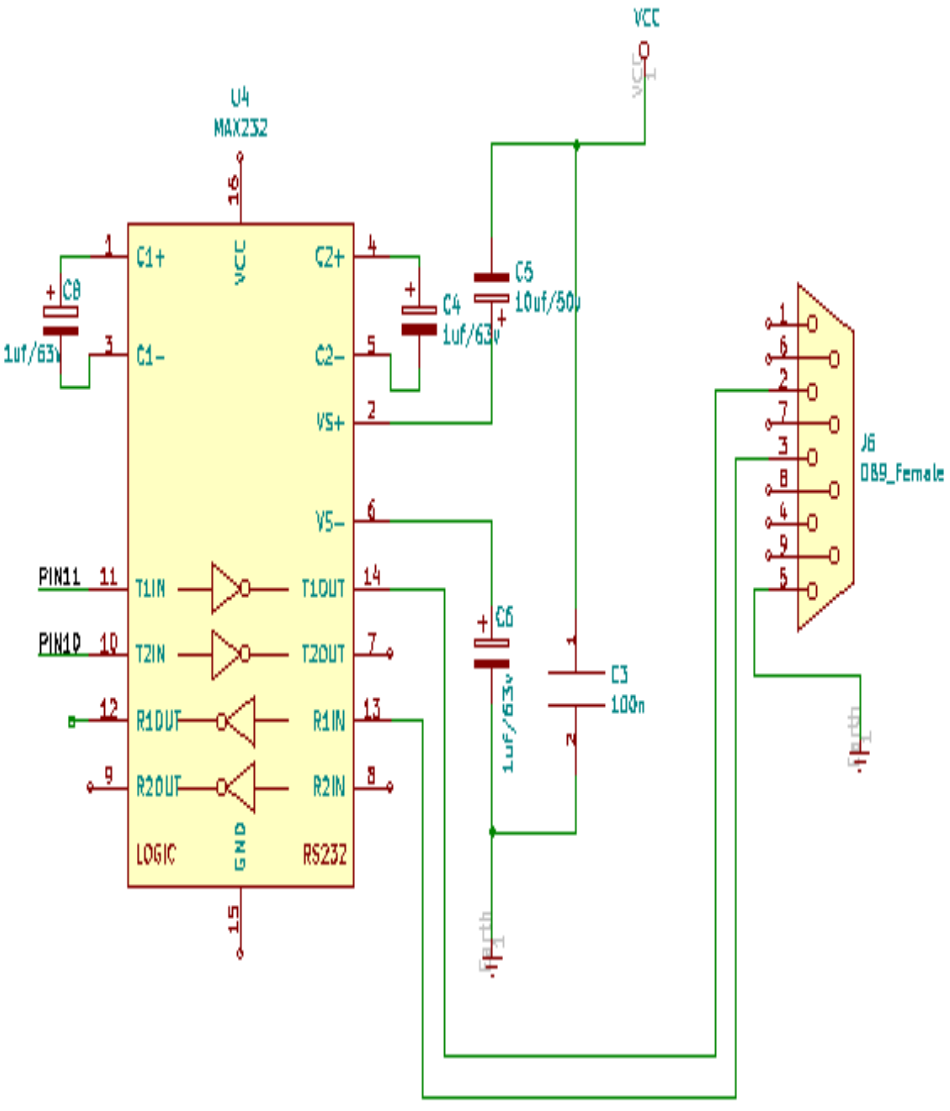
```
sbit row3= P1^2;
sbit row4= P1^3;
unsigned char code keytable[4][4]={0 ,1 ,2 ,3 ,
                                   4 ,5 ,6 ,7 ,
                                   8 ,9 ,0x10,0x11,
                                   0x12,0x13,0x14,0x15
                                   };


void main(void)
{
  unsigned char col,rowno;
  P1=0xf0;     //make cols high and rows low

  while(1)
  {
   do
        {
                P1=0xf0;
                col=P1;
                col &=0xf0;
        }while(col != 0xf0);
        do
        {
                do
                {
                 col=P1;
                 col &=0xf0;
                }while(col == 0xf0);
                delay();
                col=P1;
                col &=0xf0;
        }while(col == 0xf0);
        P1 |= 0xf0;
        while(1)
```

```
{
 row1=0;row2=1;row3=1;row4=1;
 col=P1;
 col &=0xf0;
 if(col != 0xf0)
 {
        rowno=0;
        break;
 }
 row1=1;row2=0;row3=1;row4=1;
 col=P1;
 col &=0xf0;
 if(col != 0xf0)
 {
        rowno=1;
        break;
 }
 row1=1;row2=1;row3=0;row4=1;
 col=P1;
 col &=0xf0;
 if(col != 0xf0)
 {
        rowno=2;
        break;
 }
 row1=1;row2=1;row3=1;row4=0;
 col=P1;
 col &=0xf0;
 if(col != 0xf0)
 {
        rowno=3;
        break;
 }
}
if(col == 0xe0)
```

```
            show(keytable[rowno][0]);
            else if(col == 0xd0)
            show(keytable[rowno][1]);
            else if(col == 0xb0)
            show(keytable[rowno][2]);
            else if(col == 0x70)
            show(keytable[rowno][3]);
      }

}
void show(unsigned char key)
{
            unsigned char b;
            b= key >> 4;
            b |= key << 4;
            P0=b;
}
void delay(void)
{
            unsigned char a,b;
            for(a=0;a<20;a++)
             for(b=0;b<120;b++);
}
```

EXPERIMENT NO. 12

SERIAL INERFACE



/***

Lesson : 12

Neme : SERIAL PORT

Details : THIS PROGRAM WILL SHOW HOW TO INTERFACE WITH PC

SERIAL PORT

**********/

```
#include<REG51.H>
#include<stdio.h>
//Fuction Prototypes
void delay(unsigned int);
void serial_init(void);


// Program Starts Here
void main()
{
 unsigned char a=0x0a;
serial_init();
 while(1)
 {
printf("This is Serial Test \n");
  delay(100);
 }
}

void serial_init()
{
        SCON = 0x50;        // Setup serial port control register Mode 1:
                                // 8-bit uartvar baud rate REN: enable receiver
        TMOD |= 0x20;   // Set M1 for 8-bit autoreload timer
        TH1 = 0Xfd;     // Set autoreload value for timer1 9600 baud
                                // with 11.0592 MHz xtal
        TR1 = 1;        // Start timer 1
        TI = 1;         // Set TI to indicate ready to xmit
}

void delay(unsigned inttt)
```

```
{
        unsigned inta,b;
        for(a=0;a<tt;a++)
                {
                        for(b=0;b<1000;b++)
                        ;
                }
}
```

VIII]XCTU

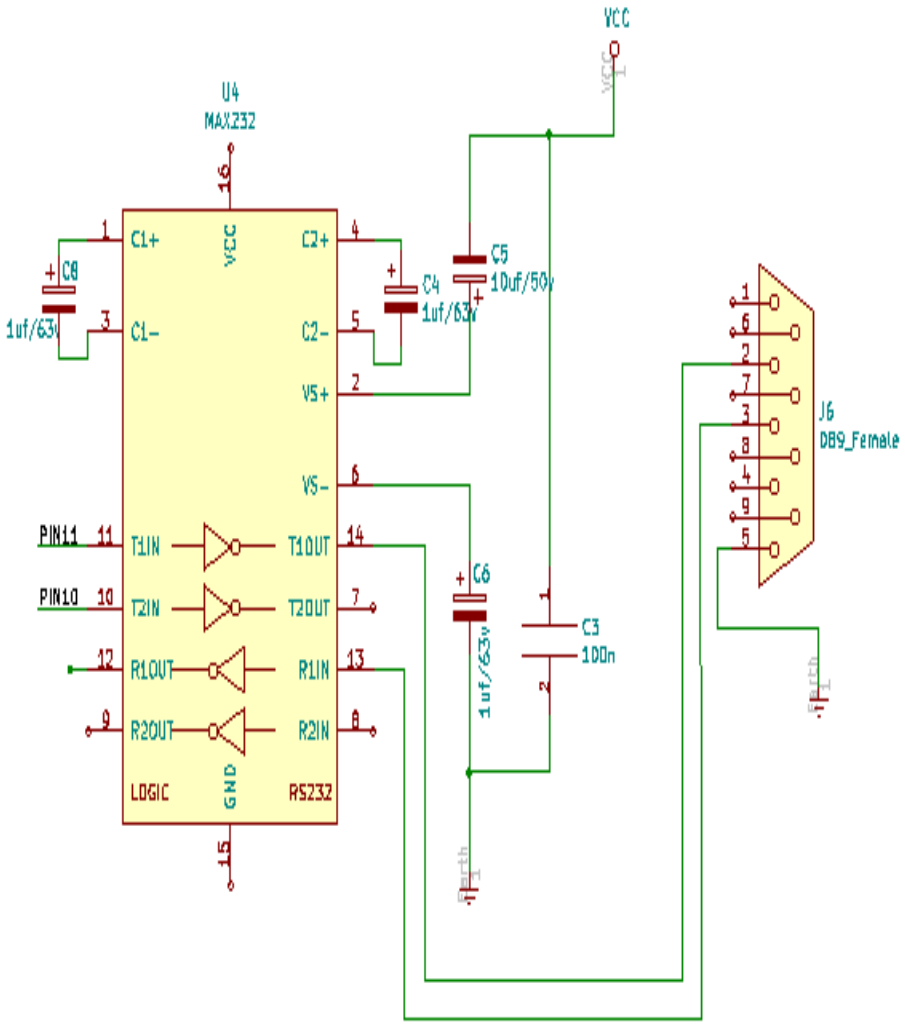
To see above output open XCTU,you will get following window.

- **In XCTU first select Communication Port (COM5).**
- **Then click on Terminal,you will get following window.**

- **Then click on Open Com Port.**
- **See the output as** "This is Serial Test".
- **Then Close Com Port.**
- **Click on Clear Screen.**

EXPERIMENT NO. 13

SERIAL INPUT OUTPUT



/***

Lesson : 13

Neme : SERIAL PORT

Details : THIS PROGRAM WILL SHOW HOW TO INTERFACE WITH PC

SERIAL PORT

```
**************************************************/
#include<REG51.H>
#include<stdio.h>
//Fuction Prototypes
void serial_init(void);


// Program Starts Here
void main()
{
 unsigned char a,b,c,tmp;
serial_init();
 while(1)
 {
printf("*************************** \n");
printf("FIND THE SUM OF TWO NUMBERS \n");
printf("*************************** \n");
printf("\n\n");
printf("Enter the First  No :- ");
  while(!(a=_getkey()));
putchar(a);
printf("\n");
printf("Enter the Second No :- ");
  while(!(b=_getkey()));
putchar(b);
printf("\n\n");
printf(" The Answer is     :- ");
  c=(a & 0x0f)+(b & 0x0f);
tmp = c/10;
tmp |= 0x30;
printf("%c",tmp);
```

```
tmp = c%10;
tmp |= 0x30;
printf("%c",tmp);
printf("\n\n\n");
  }

}
void serial_init()
{
    SCON = 0x50;        // Setup serial port control register Mode 1:
                                // 8-bit uartvar baud rate REN: enable receiver
    TMOD |= 0x20;   // Set M1 for 8-bit autoreload timer
    TH1 = 0xFD;     // Set autoreload value for timer1 9600 baud
                                // with 11.0592 MHz xtal
    TR1 = 1;        // Start timer 1
    TI = 1;         // Set TI to indicate ready to xmit
}
```

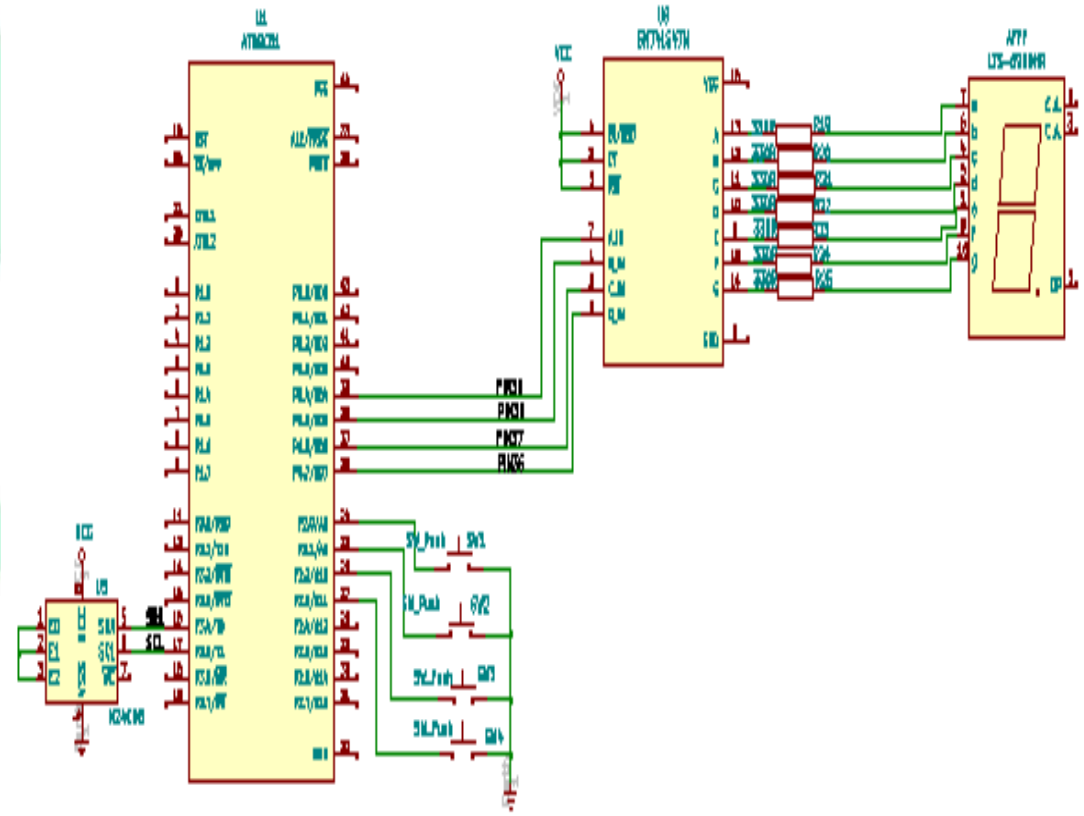
To see above output open XCTU,you will get following window.

- **In XCTU first select Communication Port (COM5).**
- **Then click on Terminal,you will get following window.**

- **Then click on Open Com Port.**
- **Type any two numbers,you will get sum.**

EXPERIMENT NO. 14

SERIAL EEPROM




```
/*****************************************************
  Lesson  : 14
  Neme    : SERIAL EEPROM
  Details : the serial eeprom works on I2C protocal
                in which there are two line
                SDA : serial data line
```

SCL : serial clock line
in this program u can store and
update a no in eeprom

```
**************************************************/
#include<REG51.H>
#include<intrins.h>
//Fuction Prototypes

void delay(unsigned int);
unsigned char ReadBYTE(unsigned char Addr);
void WriteBYTE(unsigned char Addr,unsigned char Data);
void ReadRTC(unsigned char * buff);
void WriteRTC(unsigned char * buff);

#define ACK                 1
#define NO_ACK              0
#define READ_I2C      0xA1  //I2C DEVICE ADDRESS FOR READ
OPERATION
#define WRITE_I2C   0xA0  //I2C DEVICE ADDRESS FOR WRITE
OPERATION


sbit SDA  =  P3^6;        // connect to SDA pin (Data)
sbit SCL  =  P3^7;        // connect to SCL pin (Clock)

// Program Starts Here
void main()
{
    unsigned char a,no,value;
    P2=0xFF;
    SDA=1;
    SCL=1;
    no=ReadBYTE(0x00);
    if(no == 0xff)
    {
```

```
            WriteBYTE(0x00,0x30);
        }
        while(1)
        {
        no=ReadBYTE(0x00);
        a=no>>4;
        a |= no<<4;
        P0=a;
        while(P2==0xff);
        a=P2;
        while(P2 != 0xff);
        switch(a)
        {
            case 0xfe:
                value=1;
                break;
            case 0xfd:
                value=2;
                break;
            case 0xfb:
                value=3;
                break;
            case 0xf7:
                value=4;
                break;

          }
        WriteBYTE(0x00,value);
        delay(100);

      }

    }
```

```
void delay(unsigned inttt)
{
    unsigned inta,b;
    for(a=0;a<tt;a++)
            {
                    for(b=0;b<500;b++)
                    ;
            }
}


//------------------------------
// start I2C
//------------------------------
void Start(void)
{
   SDA = 1;
    SCL = 1;
    _nop_();_nop_();_nop_();
    SDA = 0;
    _nop_();_nop_();_nop_();
    SCL = 0;
    _nop_();_nop_();_nop_();
}

//------------------------------
// stop I2C
//------------------------------
void Stop(void)
{
    SDA = 0;
    _nop_();_nop_();_nop_();
    SCL = 1;
    _nop_();_nop_();_nop_();
```

```
    SDA = 1;
}

//------------------------------
// Write I2C
//------------------------------
void WriteI2C(unsigned char Data)
{
  unsigned char i;
    for (i=0;i<8;i++)
    {
     SDA = (Data & 0x80) ? 1:0;
            SCL=1;
            SCL=0;
            Data<<=1;
    }
    SCL = 1;
    _nop_();_nop_();
    SCL = 0;


}

//------------------------------
// Read I2C
//------------------------------
unsigned char ReadI2C(bit ACK_Bit)
{
   unsigned char i,Data=0;

   SDA = 1;
    for (i=0;i<8;i++)
    {
            SCL   = 1;
```

```
            Data<<= 1;
            Data  = (Data | SDA);
            SCL   = 0;
            _nop_();
        }

        if (ACK_Bit == 1)
        SDA = 0; // Send ACK
        else
        SDA = 1; // Send NO ACK

        _nop_();_nop_();
        SCL = 1;
        _nop_();_nop_();
        SCL = 0;

        return Data;
    }

    //------------------------------
    // Read 1 byte form I2C
    //------------------------------
    unsigned char ReadBYTE(unsigned char Addr)
    {
        unsigned char Data;
        Start();
        WriteI2C(WRITE_I2C);
        WriteI2C(Addr);
        Start();
        WriteI2C(READ_I2C);
        Data = ReadI2C(NO_ACK);
        Stop();
        return(Data);
    }
```

```
//------------------------------
// Write 1 byte to I2C
//------------------------------
void WriteBYTE(unsigned char Addr,unsigned char Data)
{
    Start();
    WriteI2C(WRITE_I2C);
    WriteI2C(Addr);
    WriteI2C(Data);
    Stop();
}
```