8051 Development Board Kit

An alternative to Arduino Uno, for Studying Embedded Systems hands-on

Reference Manual

AMOTECH LABS

Embedded systems | Industrial Automation | Robotics

Scan to download Soft Copy of Manual & Online Purchase

Scan for Video Tutorials

Contents

Overview of 8051 MCU Development Kit.....3

Introduction to AT89S52 Microcontroller.....6

Schematic 7

Steps of Keil uvision for Programming.....8

Lab1. LED Blinking .....:.....15

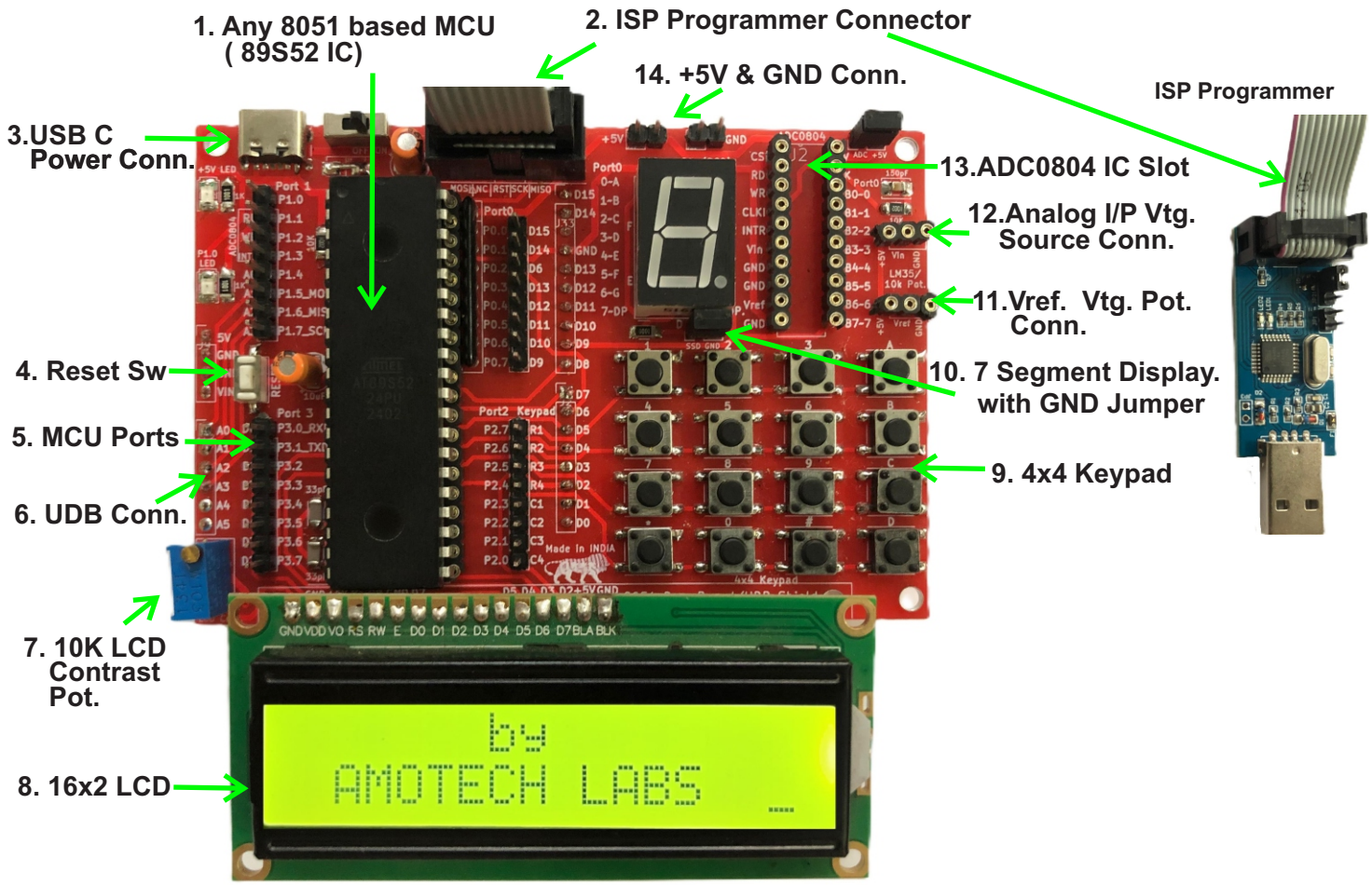
Lab5. SSD Interfacing.....17

Lab2. LCD_Interfacing.....19

Lab3. Keypad_Interfacing.....23

Lab4. ADC interfacing.....27

Overview of 8051 Development Board Kit:



Most of above labels are self explanatory. Below is the description needed for some of the labels. Also check brief Introduction of 8051(AT89S52) & Schematic of the Kit in following pages for detail understanding.

1. **Microcontroller (AT89S52) –** This Kit comes with 89S52 IC, but it can support any 8051 based 40 pin IC.
2. **ISP Programmer Connector –** It is used to connect USB ISP Programmer to program the MCU.
3. **USB-C Power Connector –** It is a USB-C connector. This board can be powered by USB C +5V adapter.
6. **UDB Conn:** These connectors are used to attach this kit to **'Universal Development Board(UDB)'** by Amotech Labs. Check UDB details on our website.
9. **4x4 Keypad –** The 4x4 Keypad is connected to Port 2 of 8051 MCU.
10. **7 Segment Display –** It is optionally connected to Port 0 of MCU. Port 0 values will reflect on it, when it's jumper is inserted. The jumper connects 7 Seg. Displays Common Cathode to GND.
11. **Vref Vtg. Pot. –** This Connector can be used to insert 10k pot. to adjust the ADC ref.(Vref) voltage for ADC0804. When this Pot. is not inserted, by default Vref is 5V, if external Vref is not given.
12. **Analog I/P Vtg. Pot. –** In this connector we can insert 10K pot. or Lm35 sensor to give variable analog Input voltage input to ADC0804.

'8051 Development Board' can be mounted on below UDB Kit:

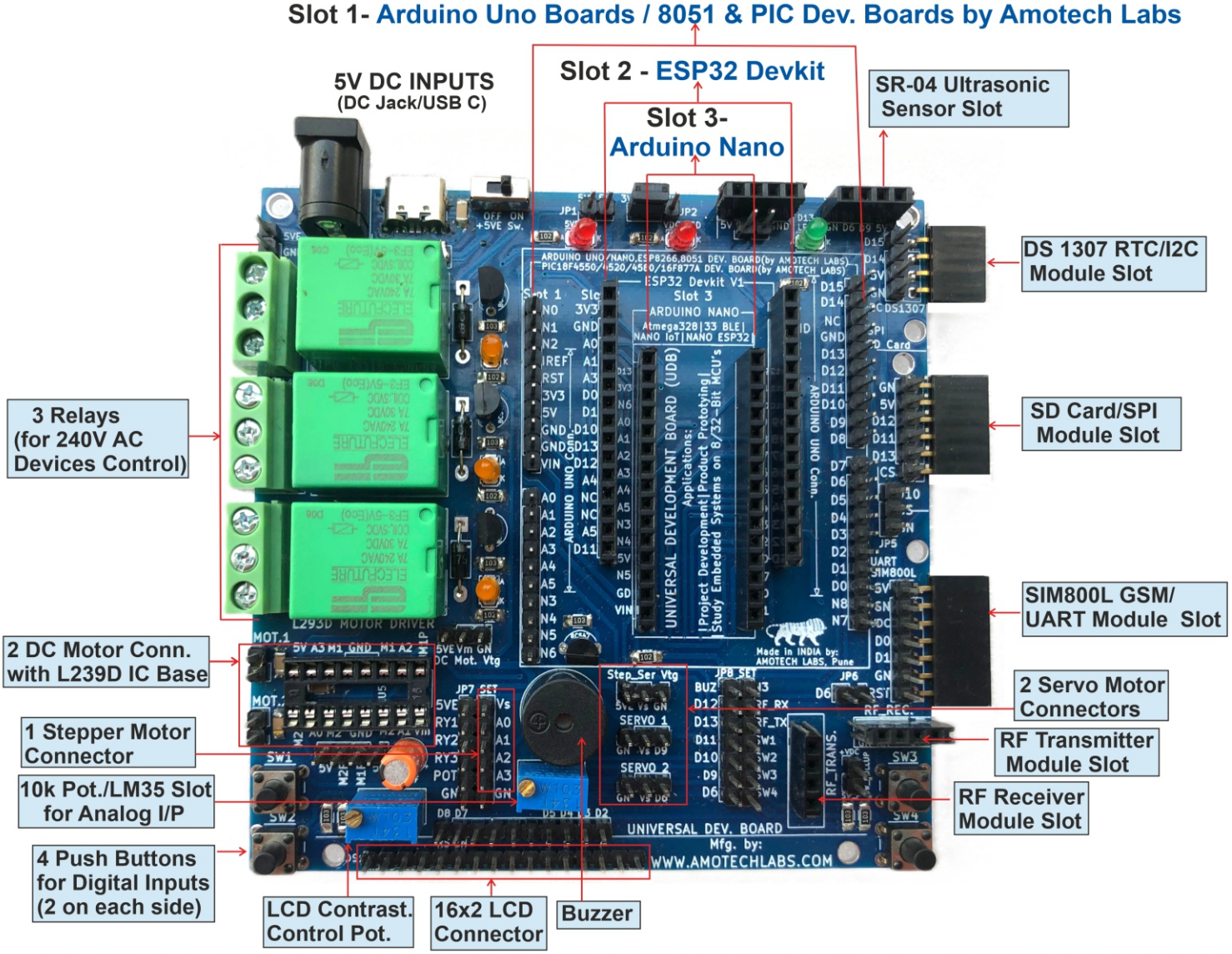
Universal Development Board(UDB) kit

One Development for

Studying Embedded Systems | Project Development | Product Prototyping

on below multiple 8 & 32-Bit Microcontrollers

8051 Shield & PIC18F/16F Dev. Boards | Arduino Uno | ESP32 Devkit V1 | Arduino/STM32 Nano Boards
(Made by Amotech Labs)



Below Micro-controller Shields/Boards can be inserted into UDB and interfaced with all above Peripherals on it

Slot 1

Arduino Uno

STM32F0/F4 Nucleo Boards

8051 & PIC18F/16 Mini Dev. Boards by Amotech Labs

8051 & PIC18F/16 Dev. Boards by Amotech Labs

Slot 2

ESP32 Devkit V1

Slot 3

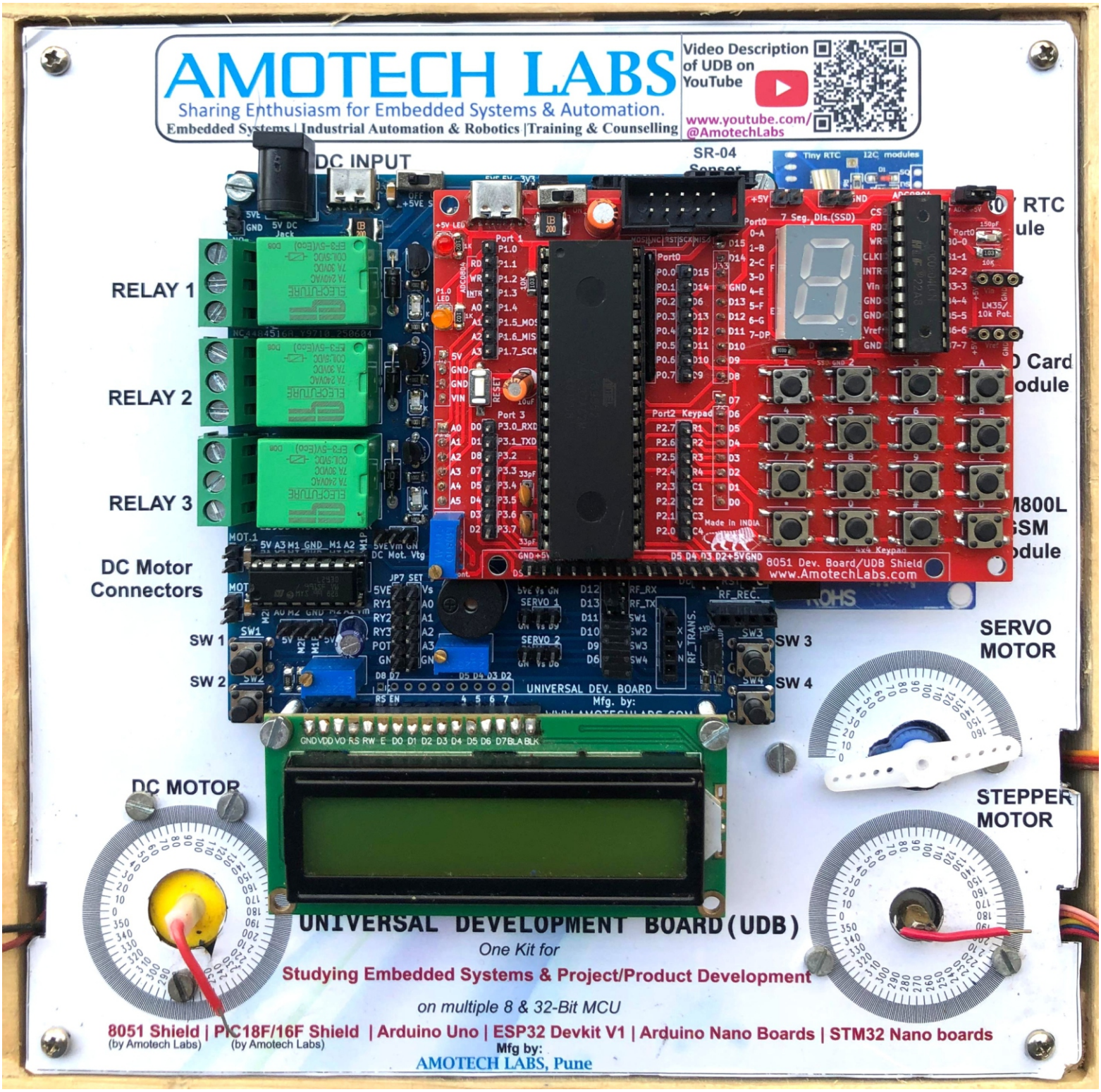
Arduino Nano Boards

STM32 Nucleo Nano Boards

Arduino Nano Rp2040

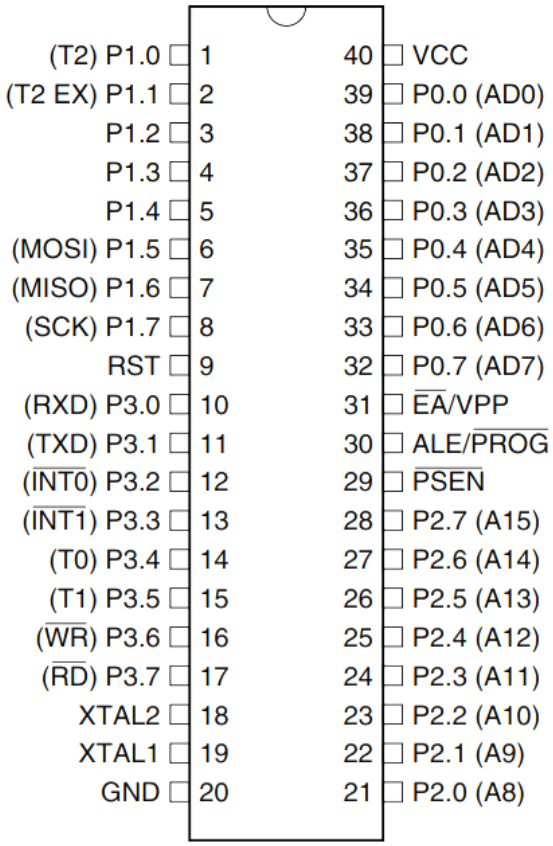
All Development Boards/Shields & in Arduino Nano Footprint

'8051 Development Board' mounted on UDB Kit's Slot 1 can be seen in the below UDB Kit which has DC, Servo and Stepper Motors mounted on-board.



Note: This Kit was assembled for College Lab use in Wooden Box Enclosure. That's why it is fitted on a Acrylic Sheet with Motors mounted on it for studying their interfacing using 8051 MCU & UDB. For Students, only UDB is also available to buy with 'UDB 2 Wheel Robot Acrylic Sheet'. Students can configure the UDB by adding the peripherals as per their application needs.

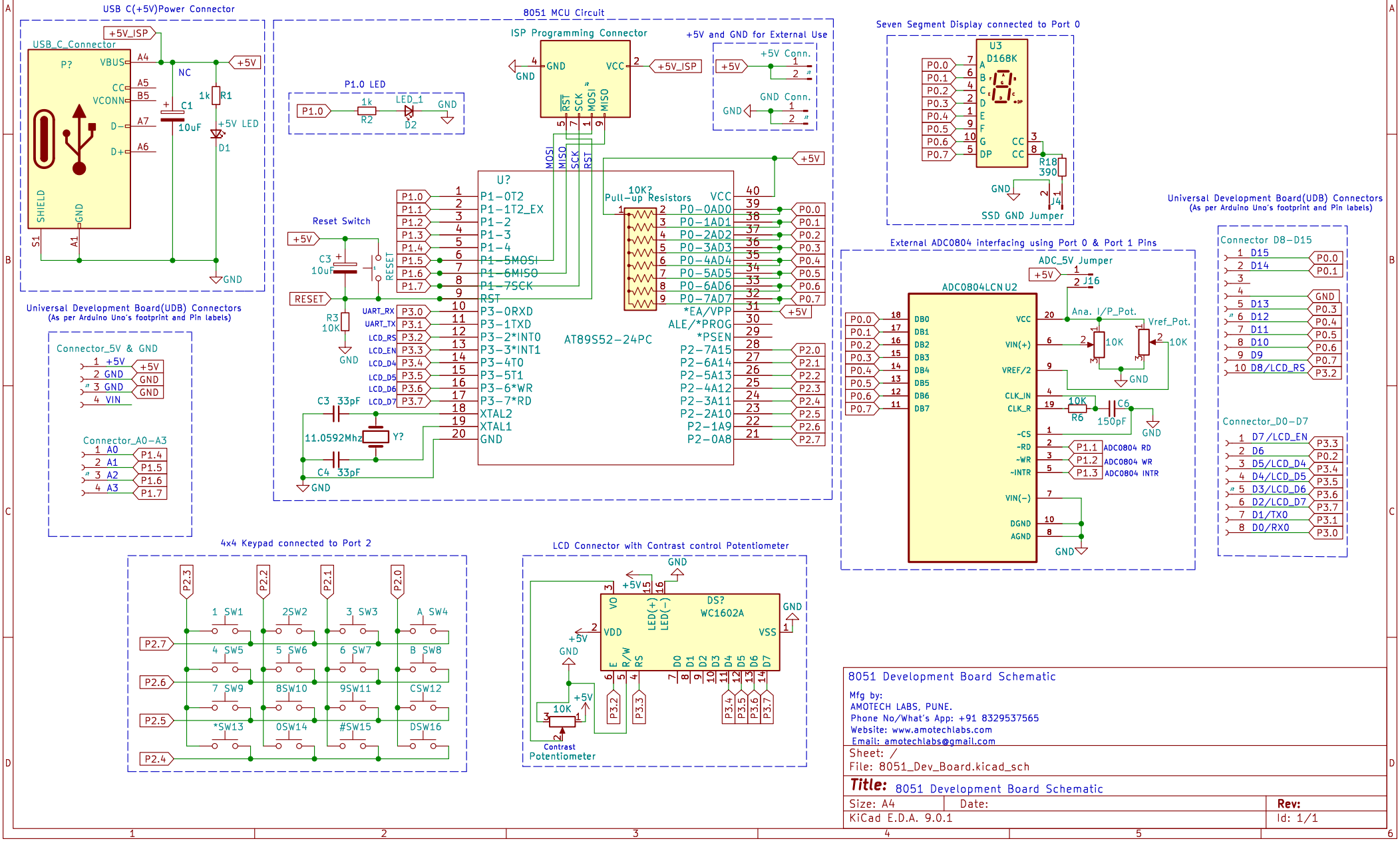
Introduction to AT89S52 Microcontroller



Features of AT89S52:

- ✓ Operating Frequency : DC 33 MHZ
- ✓ Program Memory (Bytes): 8K
- ✓ Program Memory (Instructions) : 8192
- ✓ Data Memory (Bytes): 256
- ✓ Data EEPROM Memory (Bytes) : Not available
- ✓ Interrupt Sources : 6
- ✓ I/O Ports : 4(PORT 1 ,PORT 2 ,PORT 3 , PORT 4)
- ✓ Timers : 3 (16 Bit each)
- ✓ Capture/Compare/PWM Modules : No
- ✓ Enhanced Capture/ Compare/PWM Modules : No
- ✓ Serial Communications : UART
- ✓ Universal Serial Bus (USB) Module : No
- ✓ Streaming Parallel Port (SPP) : No
- ✓ 10-Bit Analog-to-Digital Module : No
- ✓ Comparators : No
- ✓ Programmable Low-Voltage Detect : No
- ✓ Instruction Set : 255
- ✓ Packages : 40-pin DIP, 44-pin PLCC , 44-pin TQFP

8051 Development Board Schematic

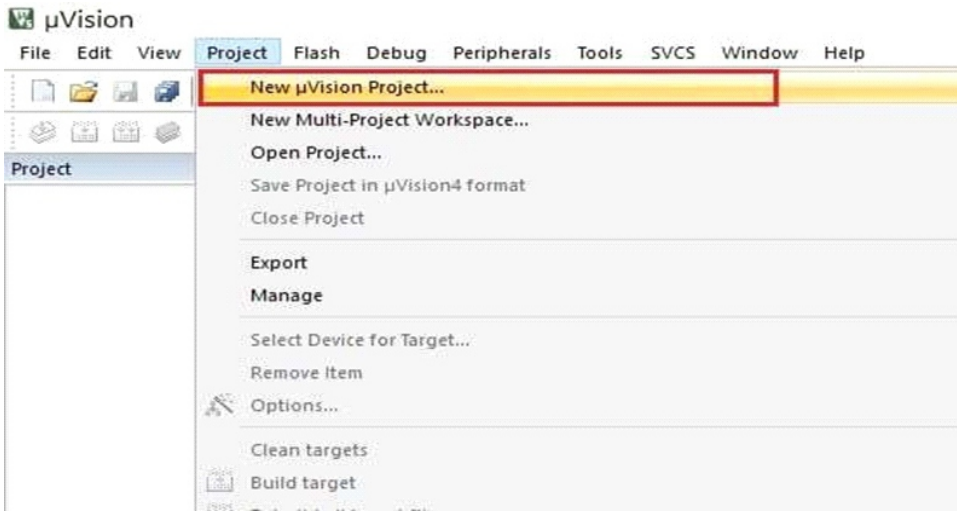


8051 Development Board Schematic	
Mfg by: AMOTECH LABS, PUNE. Phone No/What's App: +91 8329537565 Website: www.amotechlabs.com Email: amotechlabs@gmail.com	
Sheet: / File: 8051_Dev_Board.kicad_sch	
Title: 8051 Development Board Schematic	
Size: A4	Date:
KiCad E.D.A. 9.0.1	Id: 1/1

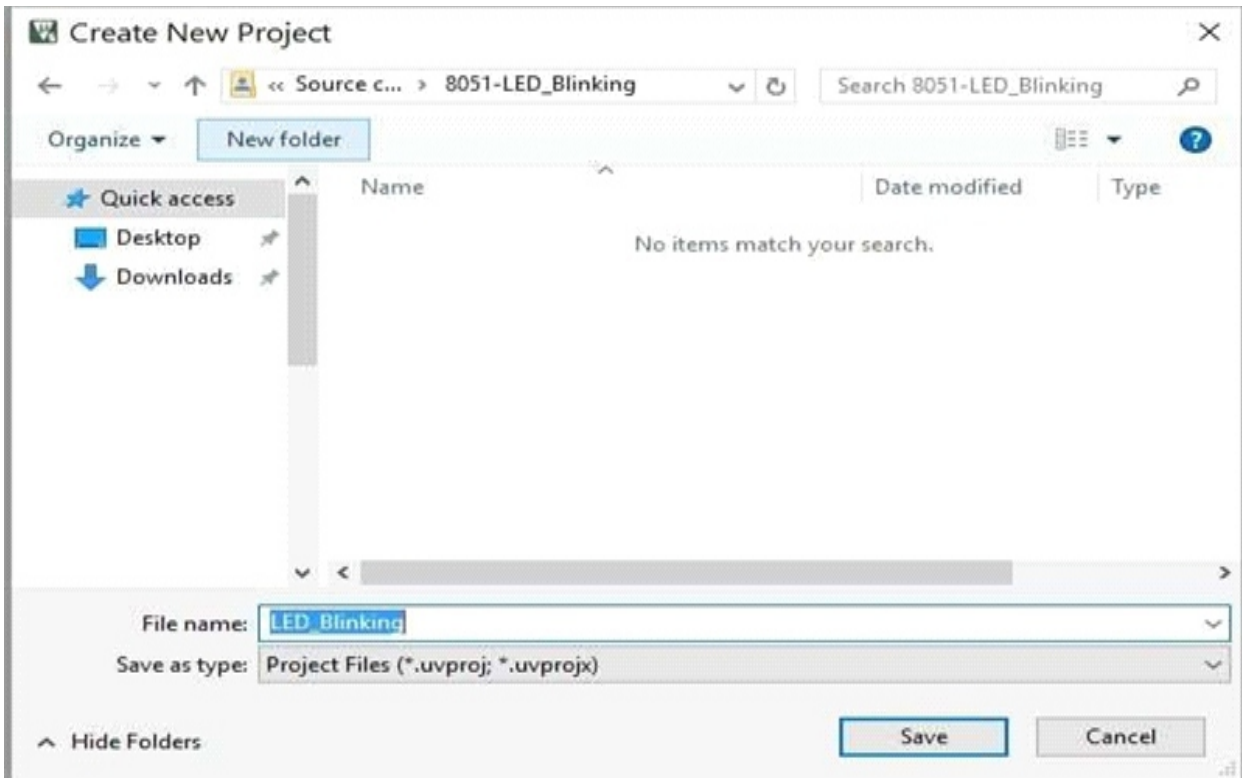
Keil IDE for Programming of 8051 (AT98S52) MC

Below are the steps to program any AT89S52 code

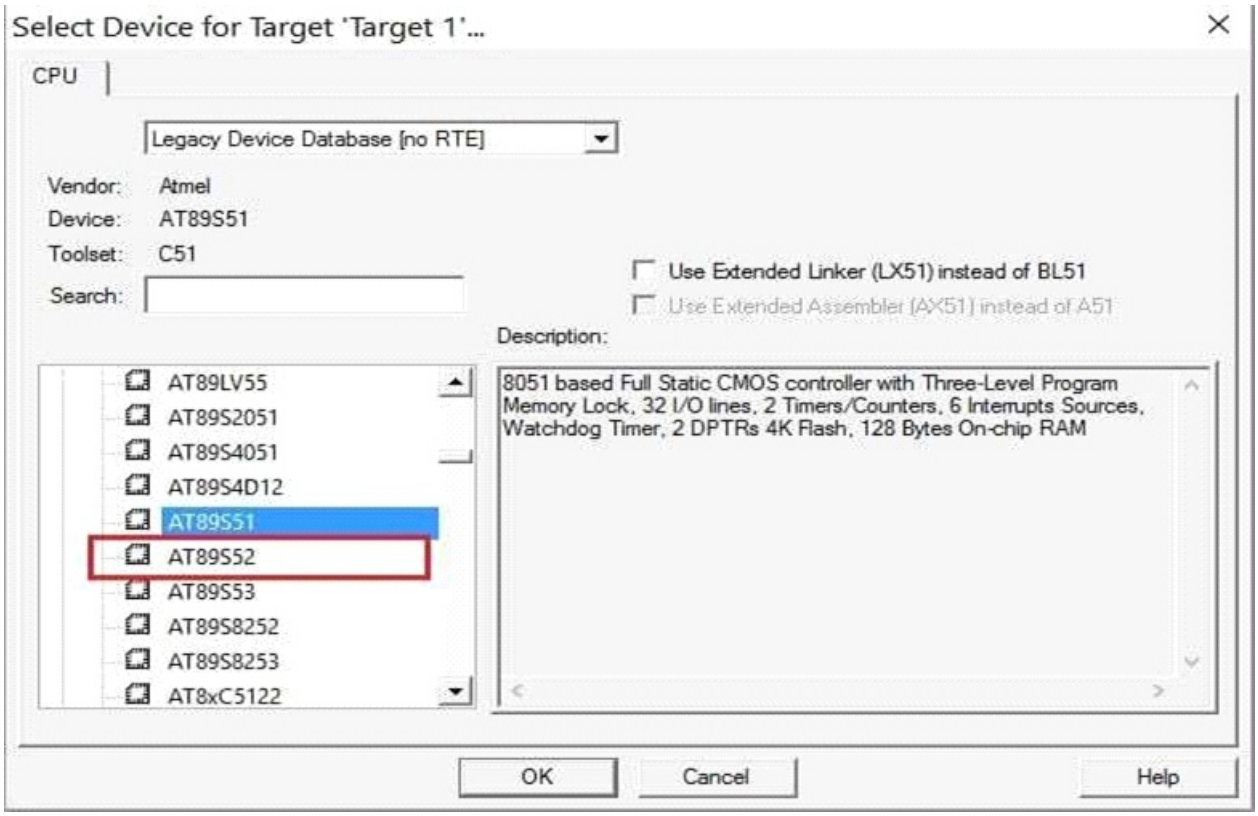
1.Select New µVision Project from the Project Menu.



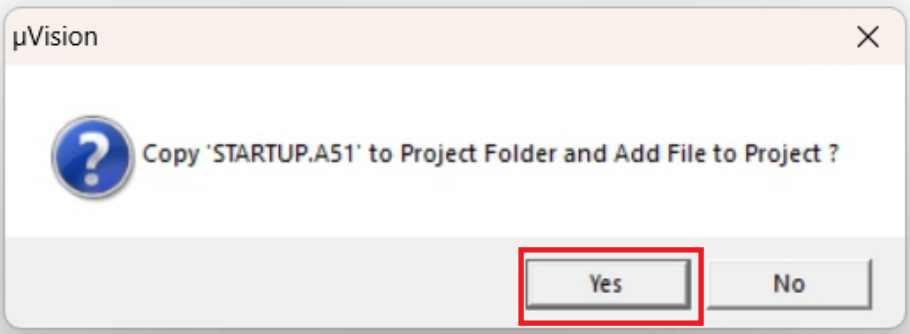
"2. Create New Project" window will pop up, type a project name and location for the project and save.



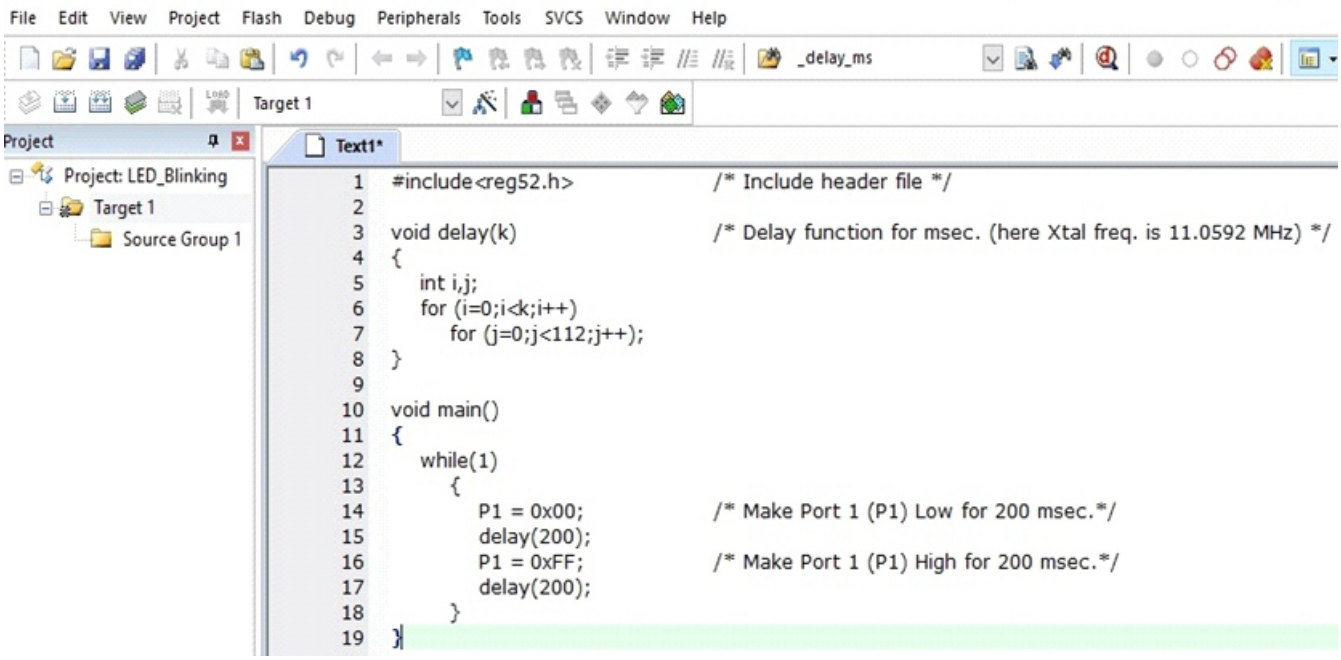
3.“Select Device for Target” window will pop up, select your device (here we selecting AT89S52)



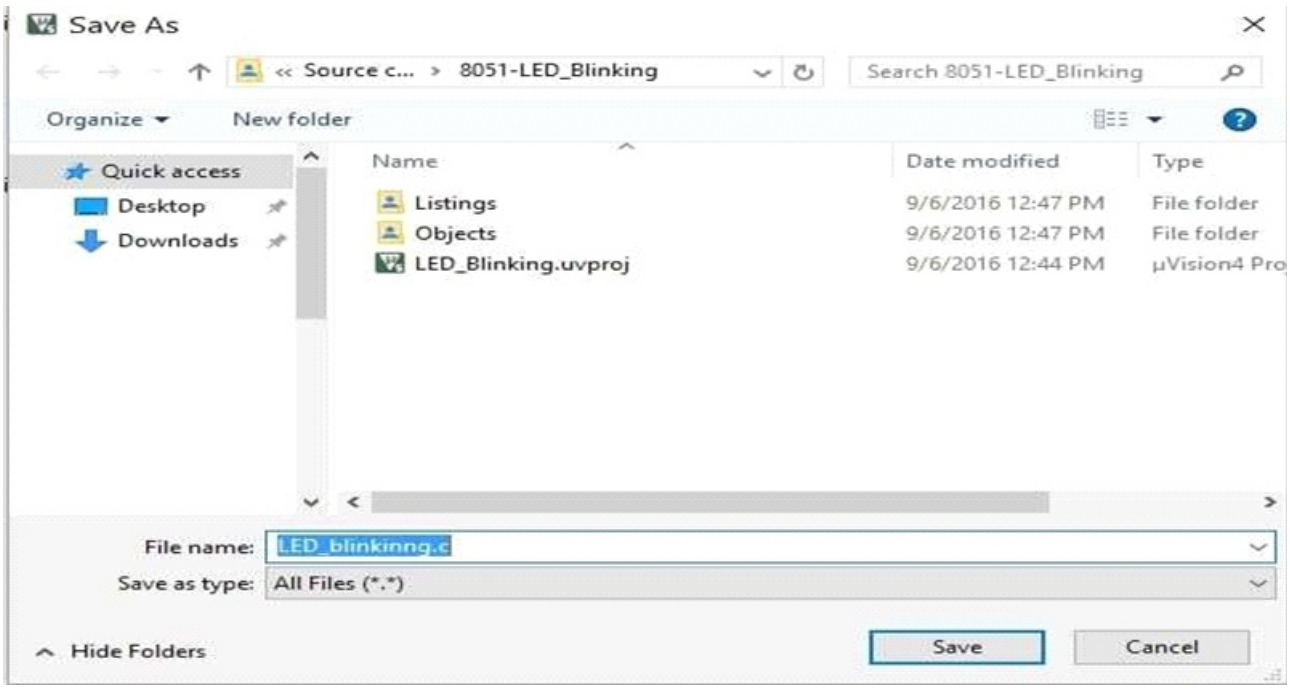
4.“µVision” window will ask for copy STARTUP.A51 to the project folder and add a file to the project (If necessary you can select “No” but we recommended you to select “Yes” because it will help to run the program smoothly.)



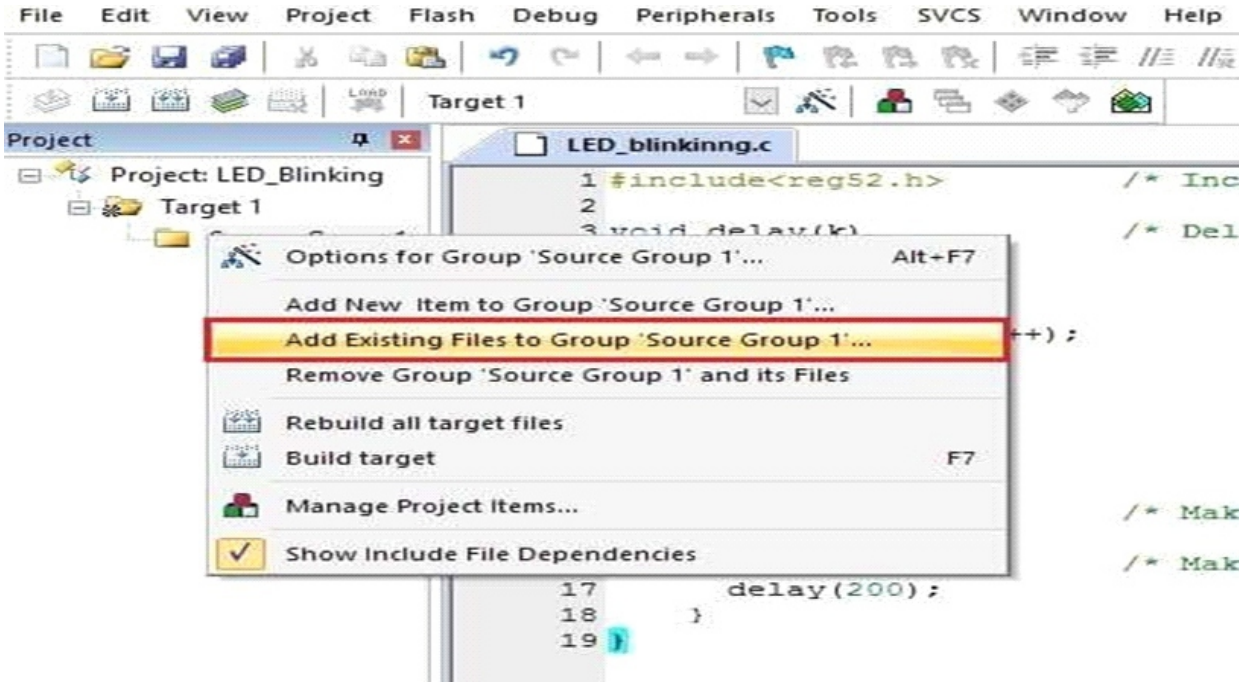
5.Now select New file from the File menu and type your program code (here we have typed LED blinkingprogram)



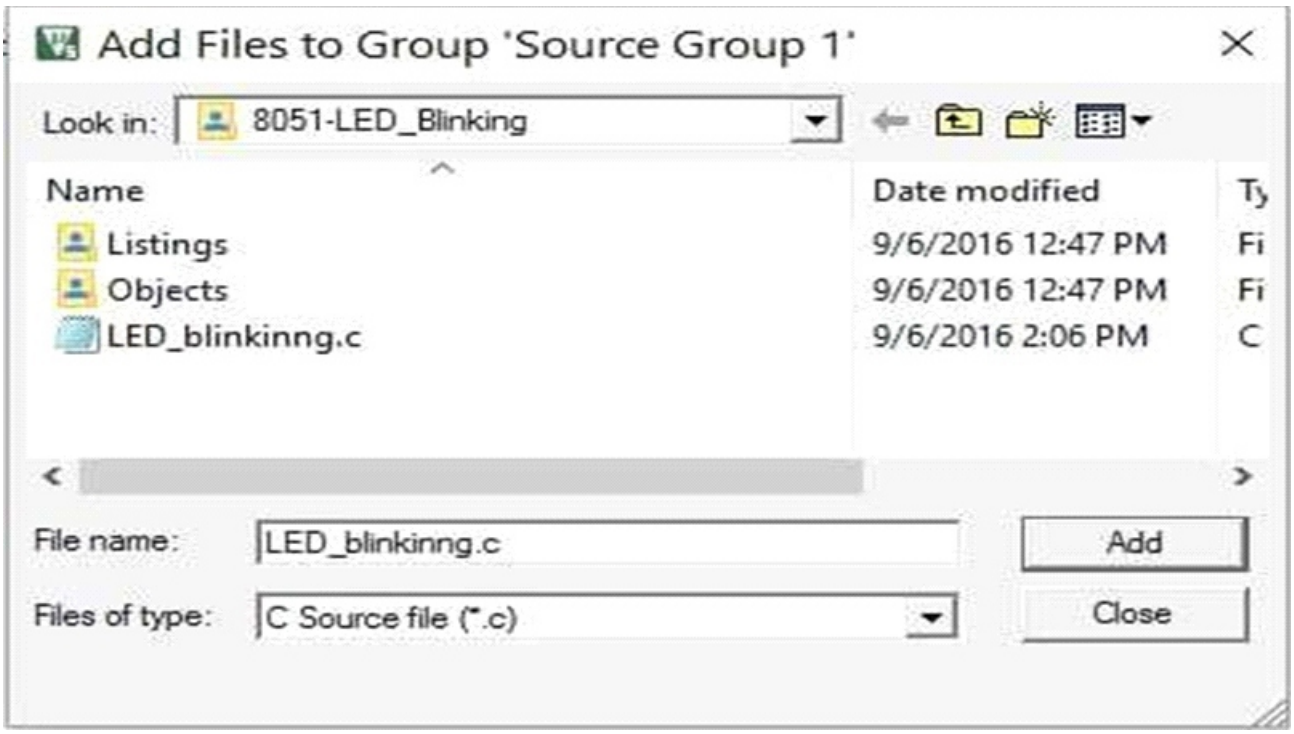
6.Save program code with ".c" extension (In case if you are using assembly language then save



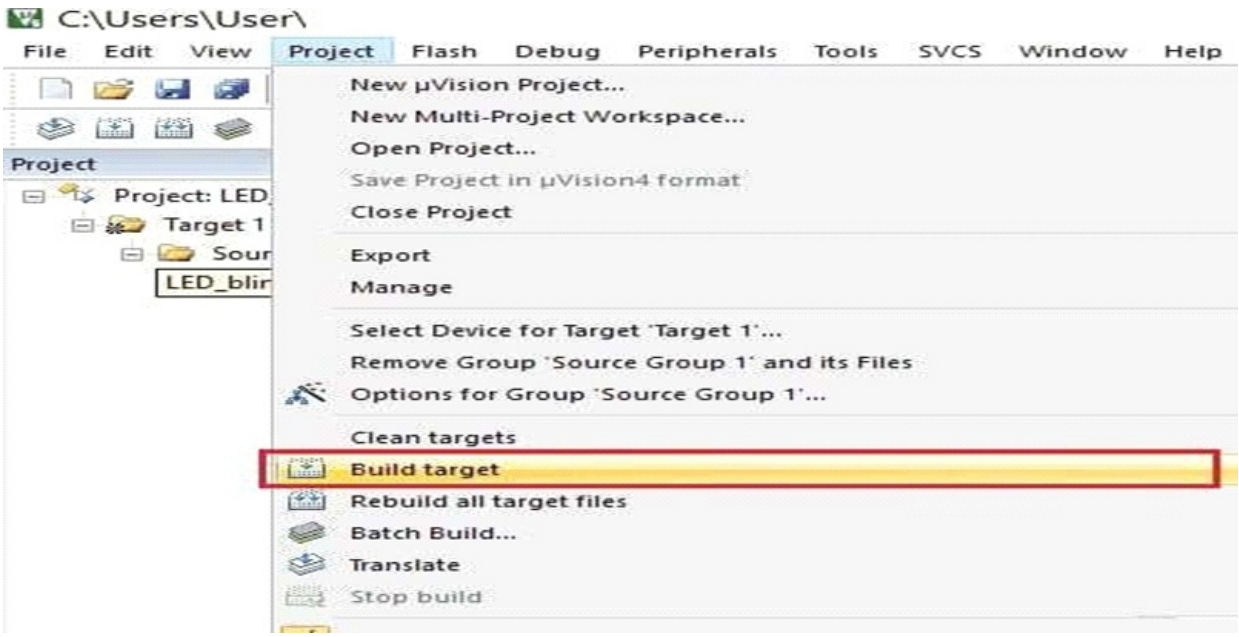
7 .Right-click on Source Group 1 folder from Target 1 and select “Add existing files to Group ‘Source Group 1’”



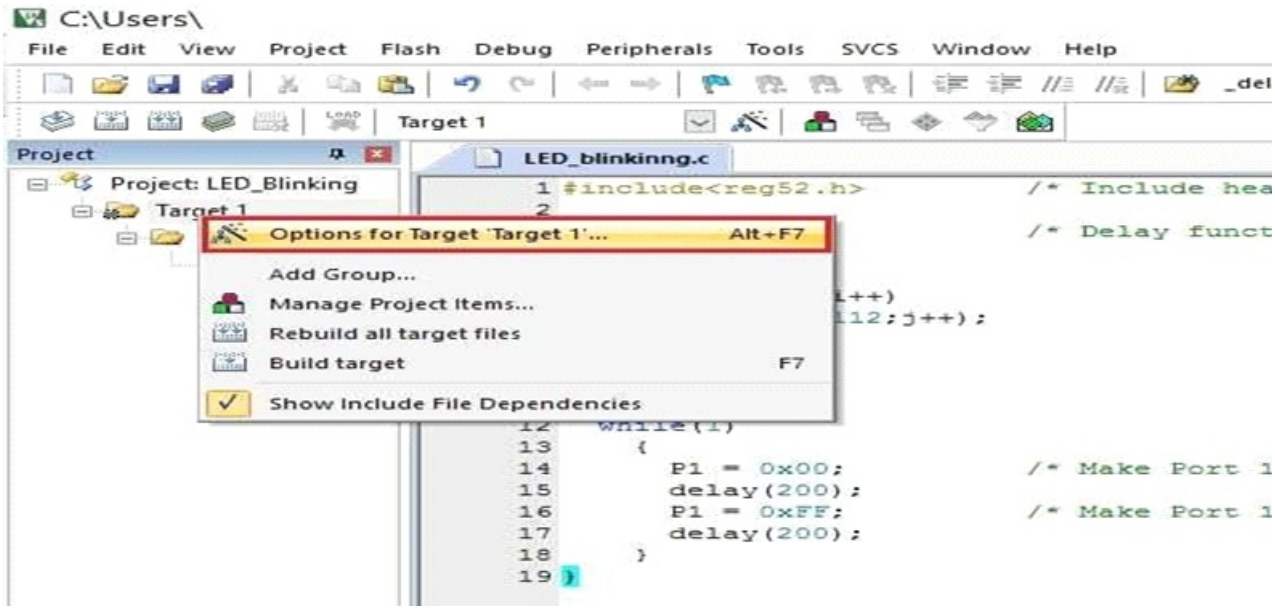
8.Select the program file saved with “.c “or “.asm” (in case of assembly language) and add it. Then close that window. You can see the added file in the “Source Group 1” folder in the left project window.



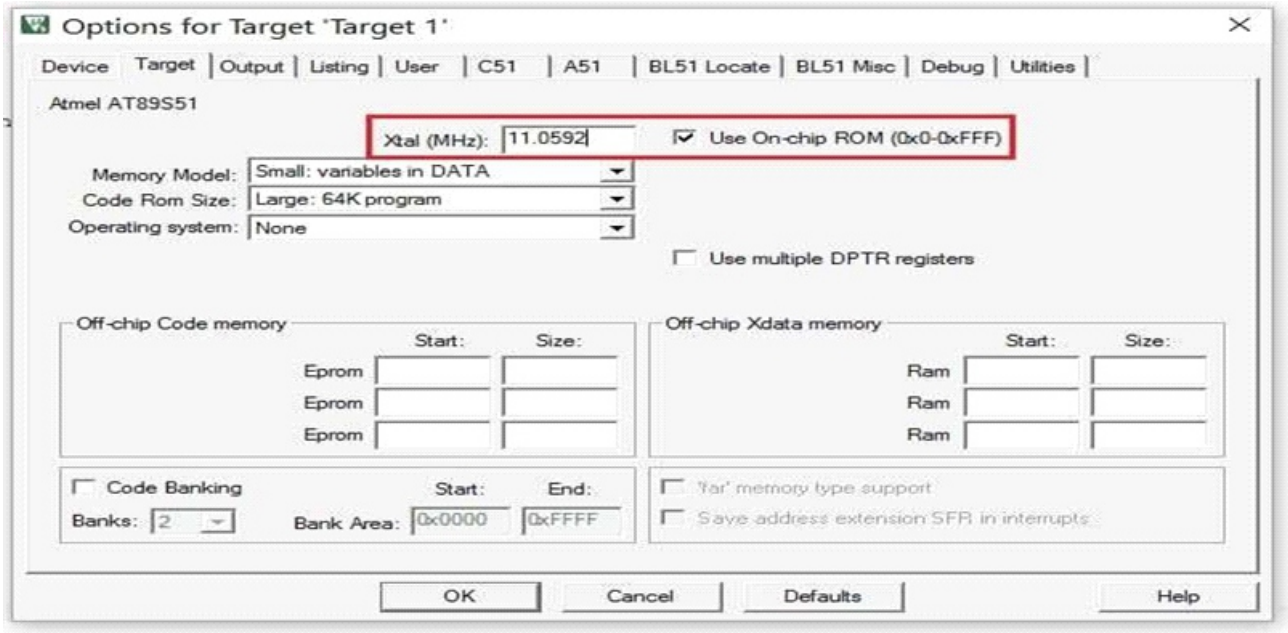
9..Now select the Project menu and click on “Build target”, it will build a project and give status in the Buildoutput window with Error and Warning count if any.



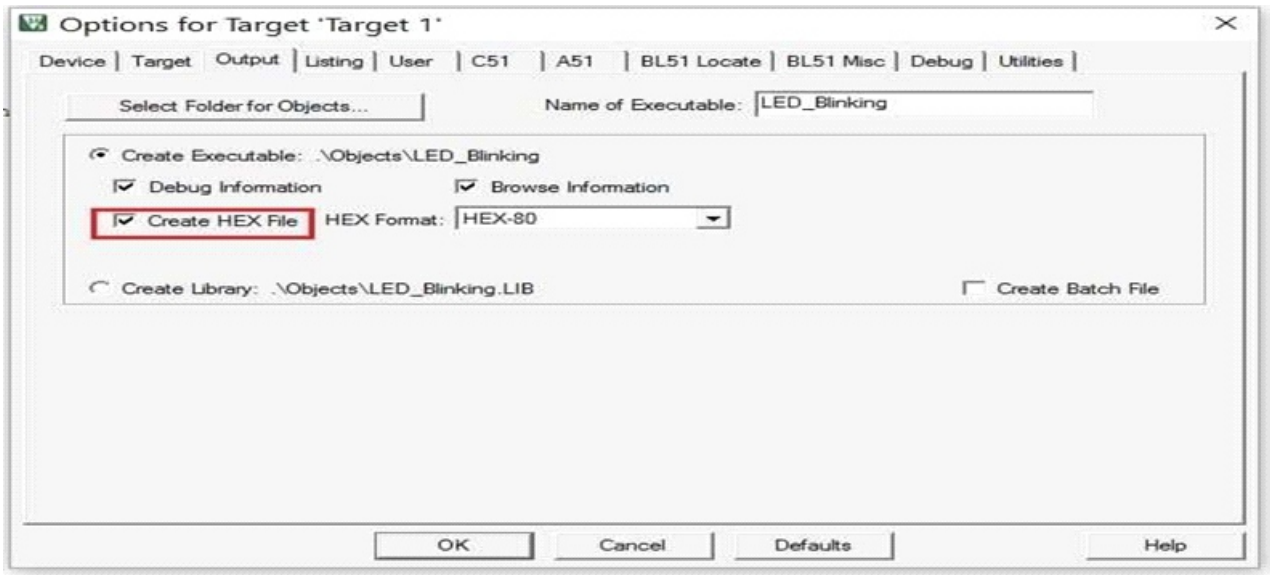
10.To create a Hex file right click on Target 1 and select Option for Target ‘Tar



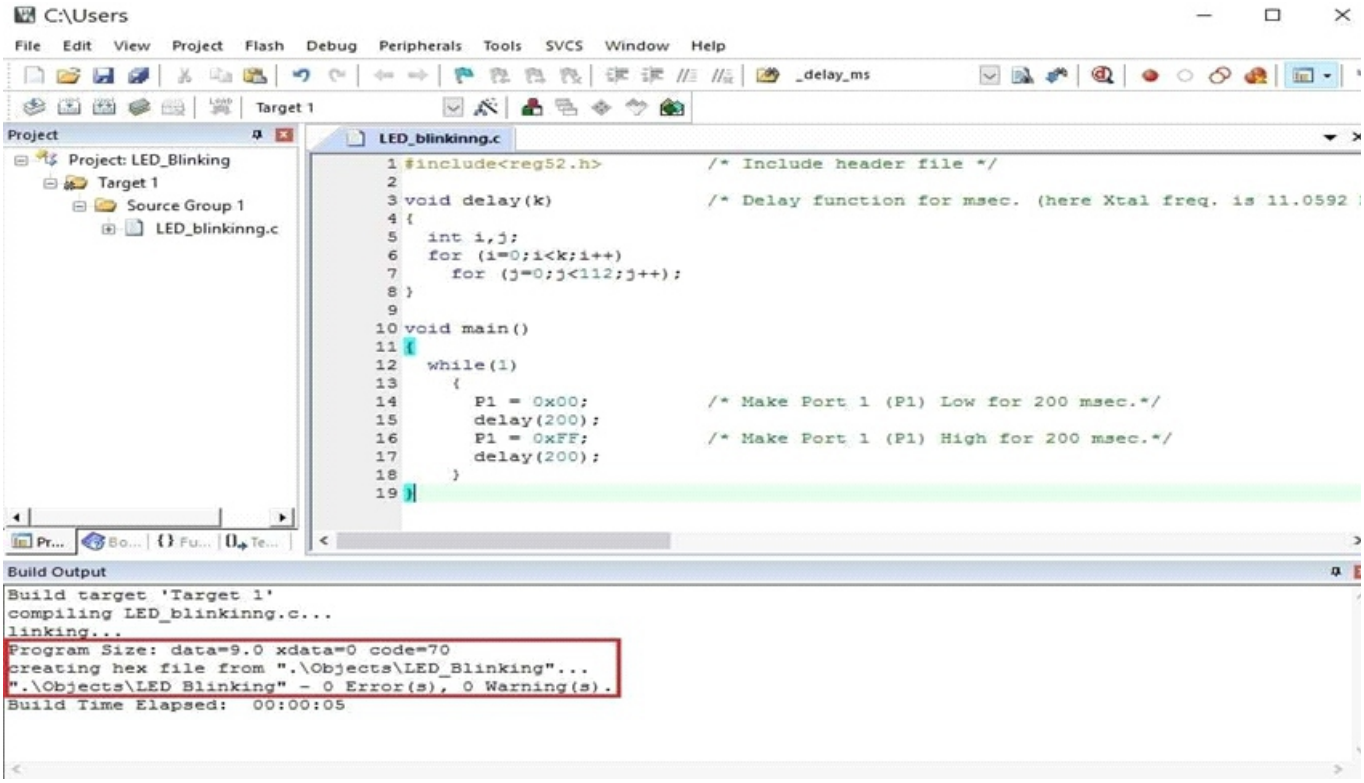
11. The target window will pop up, enter Xtal (MHz) frequency (here we used 11.0592 MHz), and tick mark in front of the “Use On-chip ROM” tag.



12. Within the same window select the “Output” option and tick mark in front of the “Create Hex File” tag and click on OK.



13 .Now again select build target from the Project menu or simply hit the F7 shortcut key for the same, it will build target and also create a Hex file. You can see creating a Hex file in the Build output window

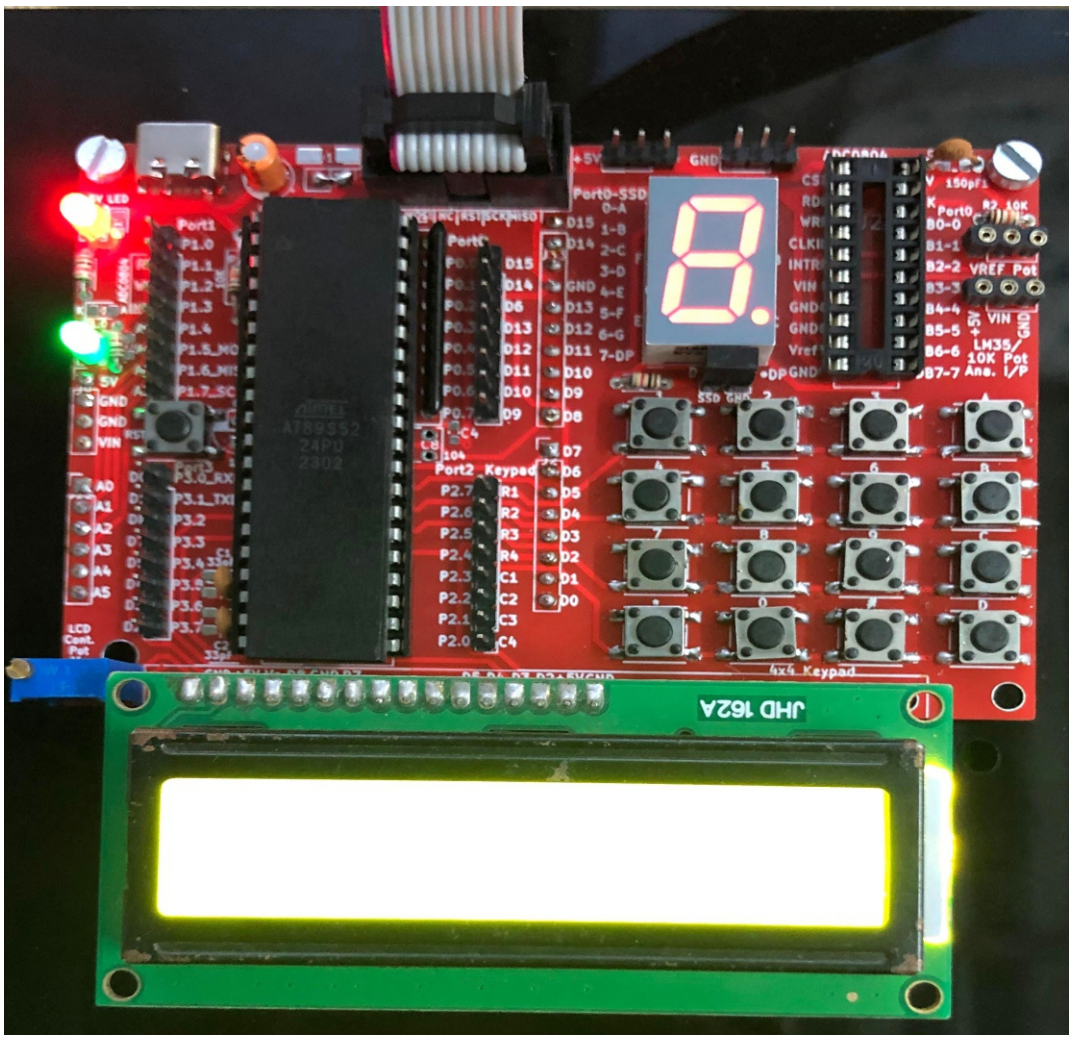


14. Check Video Tutorials on our YouTube Channel, for Instructions of Installing ISP Programmer Drivers, Compiling the program and Downloading the Hex File into our 8051 Development Boards. In Videos for '8051 Development Boards', under their description you will also find a Google Drive Link to download ISP Programmer, Drivers, Programmer Software Files, Sample Programs and Schematic.

YouTube Channel: https://www.youtube.com/@AmotechLabs

Lab 1 - LED Blinking

Aim: To study the LED's Interfacing with AT89S52 and blink them in different patterns with delay.



Procedure:

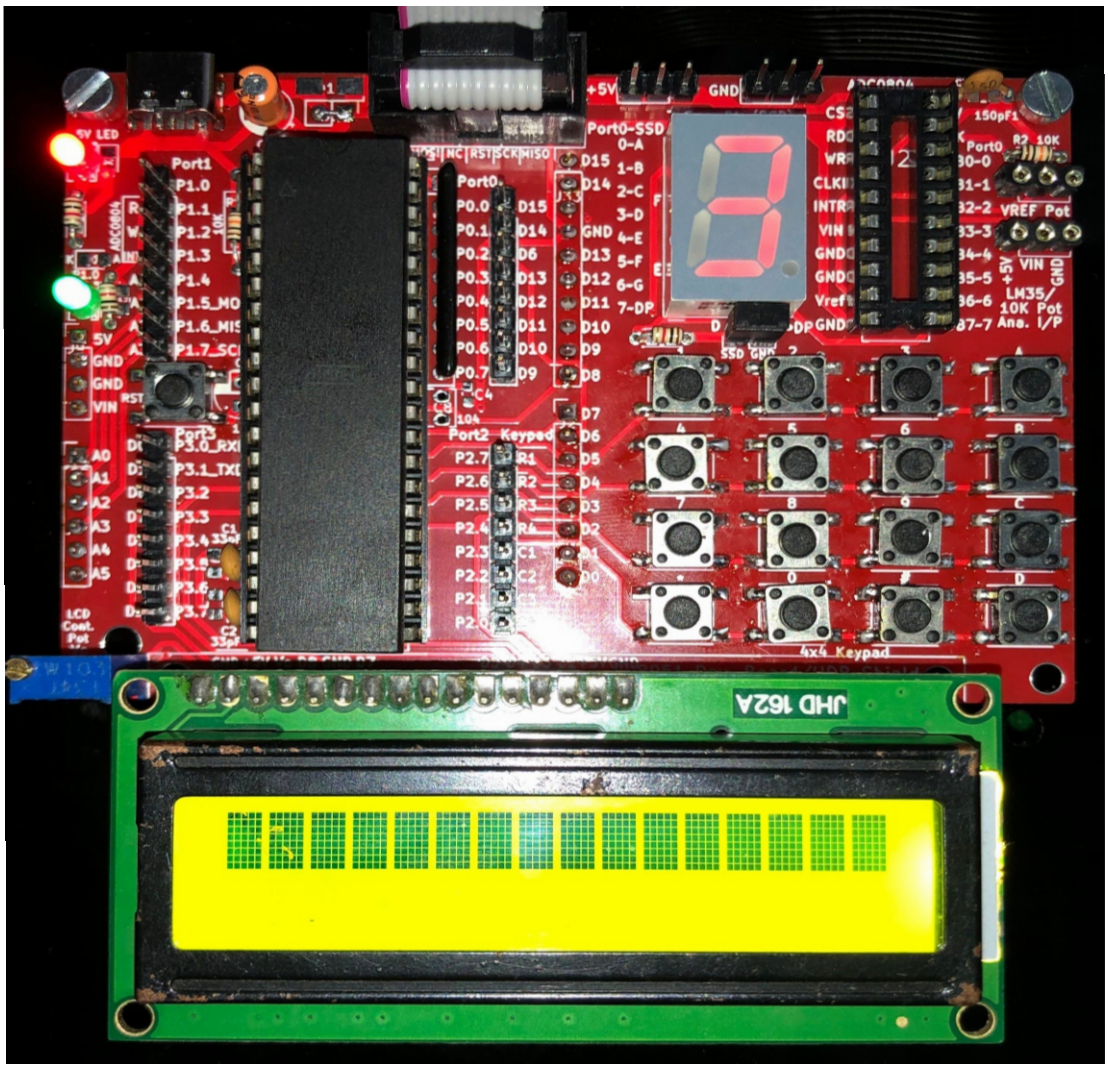
1. Read the LED Blink Program on the next page with all the comments explaining the code.
2. Open the Project folder in Keil µVision IDE or create a new project following the Keil manual.
3. Build the project and generate the HEX file.Use Keil Programmer or another suitable tool to download the HEX file to the 8051 microcontroller.
4. Ensure that the LED Ground jumper is properly inserted to provide ground to all the LED's cathode pins.

Program:

```
1. #include<reg52.h>     // special function register declarations
2.
3. sbit LED1 = P1^0;      //LED pin
4. void delay(unsigned int count);
5. void main (void)
6. {
7.      while(1) // infinite loop
8.      {
9.              LED1 = 0;
10.             P0 = 0x00;
11.             delay(1000);
12.             LED1 = 1;
13.             P0 = 0xFF;
14.             delay(1000);
15.      }
16. }
17.//delay function generates delay in millisecond.
18.void delay(unsigned int count)
19.{
20.     int i,j;
21.     for(i=0;i<count;i++)
22.     for(j=0;j<112;j++);
23.}
```

Lab 2 - SSD Counter

Aim: To study the Interfacing SSD(Seven Seg. Displays) with AT89S52 by doing SSD Counter Program.



Procedure:

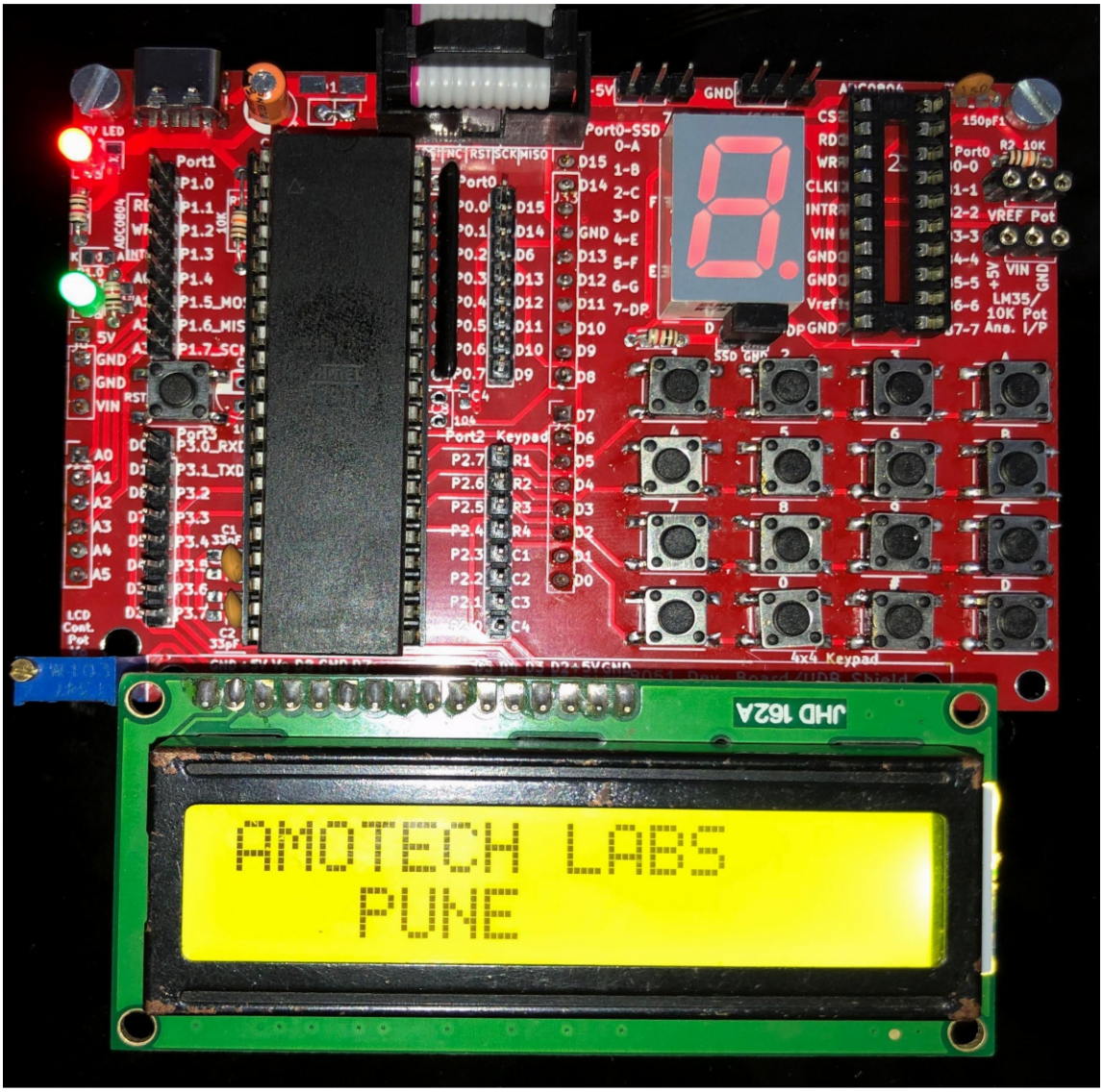
1. Read the SSD Counter Program with comments explaining the code.
2. Open the Project folder in Keil µVision IDE or create a new project following the Keil manual.
3. Build the project and generate the HEX file.Use Keil Programmer or another tool to download the HEX file to the 8051 microcontroller.
4. Insert the SSD Ground jumper to provide ground to both SSDs.

Program:

```
1.  #include<reg52.h>        // special function register declarations
2.
3.  sbit LED1 = P1^0;          //LED pin
4.  void delay(unsigned int count);
5.  unsigned char I;
6.  // Segment patterns for digits 0-9 (Common Cathode)
7.  unsigned char segment_code[] = {0x3F, 0x06, 0x5B, 0x4F, 0x66,
8.                                  0x6D, 0x7D, 0x07, 0x7F, 0x6F};
9.  void main (void)
10.{
11.      while(1)     // infinite loop
12.     {
13.              for (i=0;i<10;i++)
14.               {
15.                       P0 = segment_code[i];
16.                      delay(1000);
17.               }
18.
19.              delay(1000);
20.     }
21.}
22.//delay function generates delay in millisecond.
23.void  delay(unsigned int count)
24.{
25.     int i,j;
26.     for(i=0;i<count;i++)
27.     for(j=0;j<112;j++);
28.}
```

Lab 3 - LCD Interfacing

Aim: To study the 16x2 LCD Interfacing with AT89S52.



.**Procedure:**

1. Read the LCD Interfacing Program with all comments explaining the code.
2. Open Keil µVision, create/open a project for 8051 (AT89C51), write the program, and build it to generate the Hex file.
3. Use Flash Magic (or another programmer) to download the Hex file to the 8051 microcontroller.
4. Adjust the LCD contrast using the 10k pot. Once programmed and disconnected, the expected content will appear on the LCD.

Program:

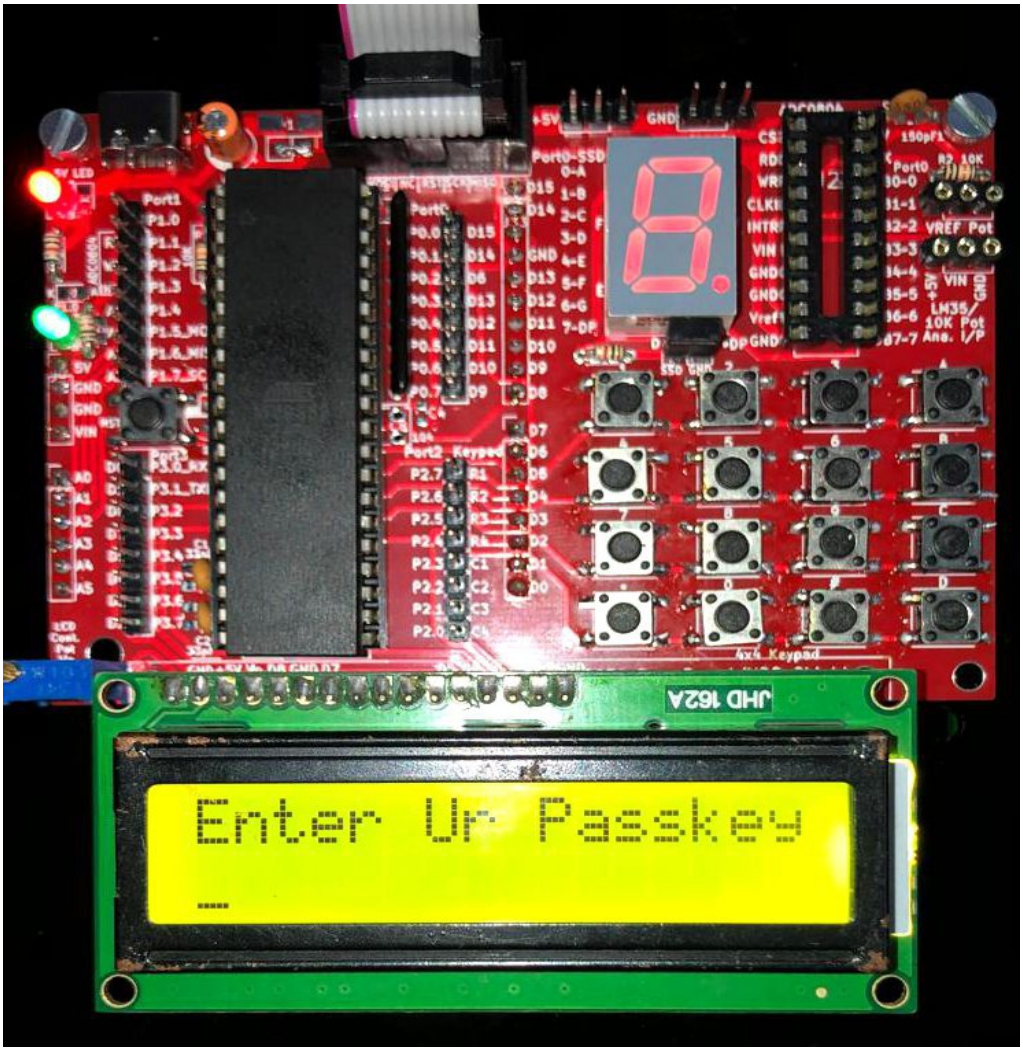
```
#include<reg51.h>

sbit D7=P3^7;  //Ard D2
sbit D6=P3^6;  //Ard D3
sbit D5=P3^5;  //Ard D4
sbit D4=P3^4;  //Ard D5

sbit rs=P3^2; /* Register select pin */  //Ard D8
sbit en=P3^3; /* Enable pin */          //Ard D7

//#define LCD_Port P1
int LCD_Port ;

/* Function to provide delay Approx 1ms with 11.0592 Mhz crystal*/
void delay(unsigned int count)
{
    int i,j;
    for(i=0;i<count;i++)
    for(j=0;j<112;j++);
}

void LCD_Command (char cmnd)  /* LCD16x2 command funtion */
{
LCD_Port = (cmnd & 0xF0)>>4;/* Send upper nibble */
D7=LCD_Port&0X08;
D6=LCD_Port&0X04;
D5=LCD_Port&0X02;
D4=LCD_Port&0X01;
rs=0; /* Command reg. */
// rw=0; /* Write operation */
en=1;
delay(1);
en=0;
delay(2);

LCD_Port = (cmnd & 0x0F);/* Send lower nibble */
D7=LCD_Port&0X08;
D6=LCD_Port&0X04;
D5=LCD_Port&0X02;
D4=LCD_Port&0X01;
rs=0;
en=1; /* Enable pulse */
delay(1);
en=0;
delay(5);
}
```

```
47. void LCD_Char (char char_data) /* LCD data write function */
48. 
49.     LCD_Port =(char_data & 0xF0)>>4;/* Send upper nibble */
50.     D7=LCD_Port&0X08;
51.     D6=LCD_Port&0X04;
52.     D5=LCD_Port&0X02;
53.     D4=LCD_Port&0X01;
54.     rs=1; /*Data reg.*/
55.     // rw=0; /*Write operation*/
56.     en=1;
57.     delay(1);
58.     en=0;
59.     delay(2);
60. 
61.     LCD_Port = (char_data & 0x0F);/* Send lower nibble */
62.     D7=LCD_Port&0X08;
63.     D6=LCD_Port&0X04;
64.     D5=LCD_Port&0X02;
65.     D4=LCD_Port&0X01;
66.     en=1; /* Enable pulse */
67.     delay(1);
68.     en=0;
69.     delay(5);
70. }
71. void LCD_String (char *str) /* Send string to LCD function */
72. 
73.     int I;
74.     for(i=0;str[i]!=0;i++) /* Send each char of string till the NULL */
75.     {
76.             LCD_Char (str[i]); /* Call LCD data write */
77.     }
78. }
79. void LCD_String_xy (char row, char pos, char *str) /* Send string to LCD function */
80. {
81.     if (row == 0)
82.     LCD_Command((pos & 0x0F)|0x80);
83.     else if (row == 1)
84.     LCD_Command((pos & 0x0F)|0xC0);
85.     LCD_String(str); /* Call LCD string function */
86. }
87. void LCD_Init (void) /* LCD Initialize function */
88. {
89.     delay(20); /* LCD Power ON Initialization time >15ms */
90.     LCD_Command (0x02); /* 4bit mode */
91.     LCD_Command (0x28); /* Initialization of 16X2 LCD in 4bit mode */
92.     LCD_Command (0x0C); /* Display ON Cursor OFF */
```

```
93.      LCD_Command (0x06); /* Auto Increment cursor */
94.      LCD_Command (0x01); /* clear display */
95.      LCD_Command (0x80); /* cursor at home position */
96. }
97.
98. void main()
99. {
100.    LCD_Init(); /* Initialization of LCD*/
101.
102.    LCD_String("AMOTECH LABS");/* write string on 1st line of LCD*/
103.
104.     LCD_Command(0xc0); /* Go to 2nd line*/
105.
106.     LCD_String("   PUNE     ");/*write string on 2nd line*/
107.
108.      delay(1000);
109.
110.
111. while(1); /* Infinite loop. */
112. }
```

Lab 4 - 4x4 Keypad Interfacing

Aim: To study the Interfacing of 4x4 Keypad with AT89S52.



Procedure:

1. Read the Keypad Interfacing Program on the next page with all the comments explaining the program.
2. Open the Project folder for the program in Keil µVision IDE or create a new project following the procedure explained in the Keil manual. Build the project and generate the HEX file.
3. Use Keil Programmer or another suitable tool to download the HEX file to the 8051 microcontroller.
4. If you entered the correct passkey then you will get acess otherwise you will get the message “Access denied”.

Program:

```
#include<reg51.h>
#include<string.h>
#define lcdport P3

sbit col1=P2^3;
sbit col2=P2^2;
sbit col3=P2^1;
sbit col4=P2^0;
sbit row1=P2^7;
sbit row2=P2^6;
sbit row3=P2^5;
sbit row4=P2^4;
sbitrs =P3^2;
sbit en=P3^3;
sbit buzzer=P1^0;
char pass[4],i=0;

void delay(int itime)
{
    int i,j;
    for(i=0;i<itime;i++)
    for(j=0;j<1275;j++);
}
void daten()
{
    rs=1;
    en=1;
    delay(5);
    en=0;
}
void lcddata(unsigned char ch)
{
    lcdport=ch & 0xf0;
    daten();
    lcdport=(ch<<4) & 0xf0;
    daten();
}
void cmden(void)
{
    rs=0;
    en=1;
    delay(5);
    en=0;
}
.. void lcdcmd(unsigned char ch)
{
    lcdport=ch &
    cmden();
    lcdport=(ch<<4) & 0xf0;
   cmden();
}
void lcdstring(char *str)
{
    while(*str)
    {
   lcddata(*str);
   str++;
    }
}
void lcd_init(void)
{
    lcdcmd(0x02);
   lcdcmd(0x28);
   lcdcmd(0x0e);
    lcdcmd(0x01);
}
 void keypad()
{
    int cursor=192,flag=0;
    lcdcmd(1);
    lcdstring("Enter Ur Passkey");
    lcdcmd(0xc0);
   i=0;
 while(i<4)
    {
    flag=cursor;
     col1=0;
     col2=col3=col4=1;
     if(!row1)
     {
       lcddata('1');
        pass[i++]='1';
        cursor++;
        while(!row1);
      }
      else if(!row2)
     {
        lcddata('4');
       pass[i++]='4';
        cursor++;
        while(!row2);
        }
      else if(!row3)
      {
        lcddata('7');
```

Program:

```
97.  pass[i++]='7';
98.        cursor++;
99.        while(!row3);
100.   }
110 else if(!row4)
111.   {
112.       lcddata('*');
113.        pass[i++]='*';
114.        cursor++;
115.       while(!row4);
116.        }
117.     col2=0;
118.     col1=col3=col4=1;
119.     if(!row1)
120.     {
121.        lcddata('2');
122.        pass[i++]='2';
123.        cursor++;
124        while(!row1);
125.        }
126.    else if(!row2)
127.     {
128.        lcddata('5');
129.        pass[i++]='5';
130.        cursor++;
131.        while(!row2);
132.        }
133.      else if(!row3)
134.     {
135.        lcddata('8');
136.        pass[i++]='8';
137.        cursor++;
138.        while(!row3);
139.        }
140.      else if(!row4)
141.     {
142.        lcddata('0');
143        pass[i++]='0';
144.        cursor++;
145.        while(!row4);
146.        }
147.     col3=0;
148.     col1=col2=col4=1;
149. if(!row1)
150.     {
151.        lcddata('3');
151. pass[i++]='3';
152.        cursor++;
153        while(!row1);
154.        }
155.      else if(!row2)
156.     {
157.        lcddata('6');
158.      pass[i++]='6';
159.        cursor++;
160.        while(!row2);
161.        }
162.     else if(!row3)
163.     {
167.        lcddata('9');
168.        pass[i++]='9';
169.       cursor++;
170.      while(!row3);
171.        }
172     else if(!row4)
173.    {
174.       lcddata('#');
175.        pass[i++]='#';
176.        cursor++;
177.        while(!row4);
178.        }
179.    col4=0;
180.     col1=col3=col2=1;
181.     if(!row1)
182.    {
183.        lcddata('A');
184        pass[i++]='A';
185.        cursor++;
186.        while(!row1);
187.        }
188.      else if(!row2)
189.    {
190.        lcddata('B');
191.        pass[i++]='B';
192.        cursor++;
193.        while(!row2);
194.        }
195. else if(!row3)
196.     {
197.        lcddata('C');
198.        pass[i++]='C';
199.        cursor++;
200.        while(!row3);
201        }
```

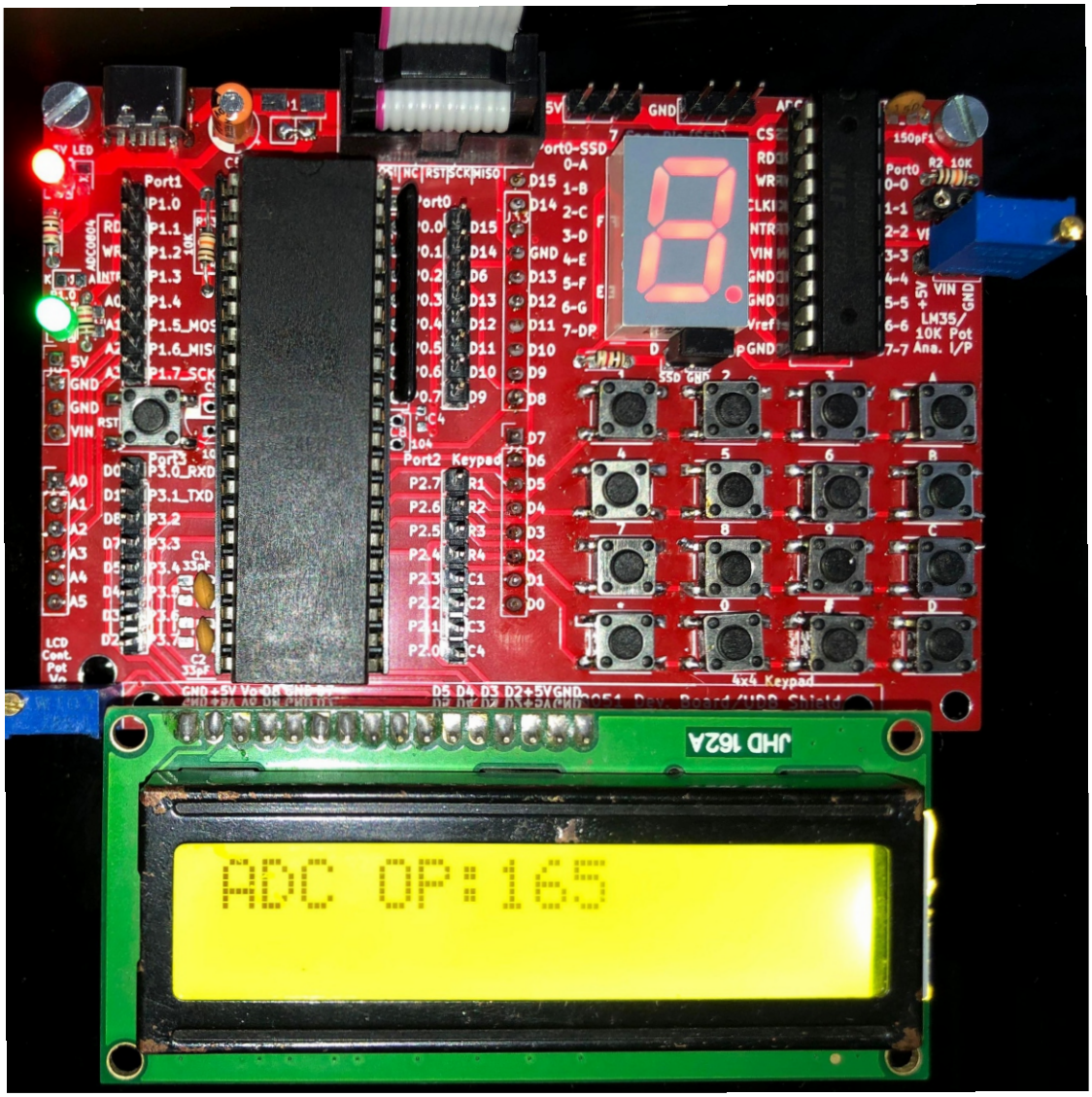
Program:

```
else if(!row4)
    {
       lcddata('D');
       pass[i++]='D';
       cursor++;
       while(!row4);
        }
    if(i>0)
    {
       if(flag!=cursor)
       delay(100);
       lcdcmd(cursor-1);
       lcddata('*');
    }
}
}
void accept()
{
   lcdcmd(1);
    lcdstring("Welcome");
    lcdcmd(192);
  lcdstring("Password Accept");
    delay(200);
}
void wrong()
{  buzzer=0;
          lcdcmd(1);
    lcdstring("Wrong Passkey");
   lcdcmd(192);
   lcdstring("PLZ Try Again");
   delay(200);
   buzzer=1;
}
void main()
{
   buzzer=1;
   lcd_init();
   lcdstring("AMOTECH LABS");
   lcdcmd(0xc0);
  lcdstring("   PUNE     ");
  delay(1000);
   lcdcmd(1);
   lcdstring("Electronic Code");
   lcdcmd(0xc0);
   lcdstring("  Lock System  ");
   delay(400);
while(1)
  {
        i=0;
        keypad();
        if(strncmp(pass,"4201",4)==0)
        {
        accept();
        lcdcmd(1);
        lcdstring("Access Granted ");
        delay(300);
           }
      else
      {
        lcdcmd(1);
        lcdstring("Access Denied");
        wrong();
        delay(300);
      }

   }
}
```

Lab 5 - ADC Interfacing

Aim: To study the ADC Interfacing with AT895S2

.



Procedure:

1. Read the ADC Interfacing Program on the next page with all the comments explaining the program
2. Open the Project folder for the program in Keil µVision IDE or create a new project following the procedure explained in the Keil manual. Build the project and generate the HEX file.
3. Use Keil Programmer or another suitable tool to download the HEX file to the 8051 microcontroller.
4. Make sure that the jumper is used for internal ADC or Lm35.

```
1.  #include<reg51.h>
2.  sbit  D7=P3^7;                          //Ard D2
3.  sbit  D6=P3^6;                          //Ard D3
4.  sbit  D5=P3^5;                          //Ard D4
5.  sbit  D4=P3^4;                          //Ard D5
6.  sbit  rs = P3^2;                        /* Register select pin */  //Ard D8
7.  sbit  en =P3^3;
                                            /* Enable pin */          //Ard D7
8. int  LCD_Port ;
                                            //#define LCD_Port P1
9.  sbit  rd=P1^1;                          //ADC Part
10. sbit  wr=P1^2;
11. sbit  intr=P1^3;

12. unsignedchar  adc(), get_value, conv;
13. int I;

14. void delay(unsignedint  count) /* Function to provide delay Approx 1ms with 11.0592 Mhz crystal*/
15. {
16.     int i,j;
17.     for(i=0;i<count;i++)
18.     for(j=0;j<112;j++);
19. }

20. void LCD_Command (char cmnd)                /* LCD16x2 command funtion */
21. {
22.     LCD_Port = (cmnd & 0xF0)>>4;                    /* Send upper nibble */
23.     D7=LCD_Port&0X08;
24.     D6=LCD_Port&0X04;
25.     D5=LCD_Port&0X02;
26.     D4=LCD_Port&0X01;
27.     rs=0;                                           /* Command reg. */
28.     en=1;                                           // rw=0; /* Write operation */
29.     delay(1);
30.     en=0;
31.     delay(2);

32.     LCD_Port = (cmnd & 0x0F);                       /* Send lower nibble */
33.     D7=LCD_Port&0X08;
34.     D6=LCD_Port&0X04;
35.     D5=LCD_Port&0X02;
36.     D4=LCD_Port&0X01;
37.     rs=0;
38      en=1;                                           /* Enable pulse */
39.     delay(1);
40.     en=0;
41.     delay(5);
42. }
```

```
43. void LCD_Char (char char_data)              /* LCD data write function */
44. {
45.     LCD_Port =(char_data & 0xF0)>>4;       /* Send upper nibble */
46.     D7=LCD_Port&0X08;
47.     D6=LCD_Port&0X04;
48.     D5=LCD_Port&0X02;
49.     D4=LCD_Port&0X01;
50.     rs=1;                                  /*Data reg.*/
51.     // rw=0; /*Write operation*/
52.     en=1;
53.     delay(1);
54.     en=0;
55.     delay(2);

56.     LCD_Port = (char_data & 0x0F);         /* Send lower nibble */
57.     D7=LCD_Port&0X08;
58.     D6=LCD_Port&0X04;
59.     D5=LCD_Port&0X02;
60.     D4=LCD_Port&0X01;
61.     en=1;                                  /* Enable pulse */
62.     delay(1);
63.     en=0;
64.     delay(5);
65.  }

66. void LCD_String (char *str)                 /* Send string to LCD function */
67. {
68.     int I;
69.     for(i=0;str[i]!=0;i++)                 /* Send each char of string till the NULL */
70.     {
71.             LCD_Char (str[i]);             /* Call LCD data write */
72.     }
73. }

74. void LCD_String_xy (char row, char pos, char *str)        /* Send string to LCD function */
75. {
76.     if (row == 0)
77.             LCD_Command((pos & 0x0F)|0x80);
78.     else if (row == 1)
79.             LCD_Command((pos & 0x0F)|0xC0);
80.             LCD_String(str);                        /* Call LCD string function */
81. }
82. void LCD_Init (void)                                 /* LCD Initialize function */
83. {
84.     delay(20);                      /* LCD Power ON Initialization time >15ms */
85.     LCD_Command (0x02);             /* 4bit mode */
86.     LCD_Command (0x28);             /* Initialization of 16X2 LCD in 4bit mode */
87.     LCD_Command (0x0C);             /* Display ON Cursor OFF */
88.     LCD_Command (0x06);             /* Auto Increment cursor */
89. }
```

```
90. unsigned char adc()
91. {
92.       wr= 0;
93.       for(i=0;i<1000;i++);
94.            rd=1;
95.       for(i=0;i<1000;i++);
96.       wr=1;
97.       for(i=0;i<1000;i++);
98.       while(intr==1);
99.       rd=0;
100.      conv=P0;
101.      return conv;
102. }
103. void main()
104.{
105.    LCD_Init();                    /* Initialization of LCD*/
106.    LCD_String("ADC OP:");
107.    while(  )
108.   {
109.                  get_value = adc();
110.                  LCD_Command(0x87);
111.                  LCD_Char ((get_value/100)+48);
112.                  LCD_Char (((get_value/10)%10)+48);
113.                  LCD_Char ((get_value%10)+48);
114.
115.                  //  LCD_Char (0x60);
116.                   // LCD_Char ('C');
117.            delay(200);
118    }
119. }
```