**Pulse Width Modulation with External interrupt**

EX1 - External Interrupt 1  
IT1 - Interupt control bit (1 for the falling edge and 0 for rising edge Triggered - P3.3 is the pin for IT1)  
P3.2 for IT0  
  
Generate a PWM Waveform on P1.3 for a duty cycle of 75% that will be triggered by the External Interrupt 1 on its falling edge.  
  
Main function   
\* Enable the External Interrupt   
\* Enable the interrupt control bit   
Inside the External Interrupt 1  
\* Initialize the [timer](http://moodlecc.vit.ac.in/mod/assign/view.php?id=3335) information like TMOD, EA, ET1,TR1  
  
To see the output  
In the debug mode, open the Peripheral -> Io Ports and Port3.  
Untick the P3.3 pin, and you can see the PWM generated.

**Serial Communication using 8051**

Baud Rates: 57600, 28800, 19200, 14400, 9600, 7200, 4800, 3600, 2400, 1200, 600, 300,150  
Baud Rate Calculation:  
  
Baud Rate = Osc. Frequency / 12 \*32 \* (256-TH1)  
  
TH1= 256 - (OSc. Freq/384\*Baud Rate)  
For 9600, the value that has to be loaded to the TH1 is 253   
TH1=0xFD; Baud Rate 9600  
TH1=0xFF; for 28800 baud rate  
I need 57600 baud RATE. To double the baud rate, there is a control pin PCON.7, make it to 1 for doubling the baud rate.  
but the PCON is not a bit addressable register, So to double the baud rate,. PCON=0x80;  
  
Serial Communication using UART for 8051   
  
Steps:  
\* [Timer](http://moodlecc.vit.ac.in/mod/assign/view.php?id=3335) 1 is used for Serial Communication  
\* [Timer](http://moodlecc.vit.ac.in/mod/assign/view.php?id=3335) 1 to run with 8 bit auto reload and TH1 register is loaded for Baud Rate  
\* Serial Control Register to be loaded   
\* As long as TI and RI is clear, data will be transmitted and received...   
  
TMOD=0x20; 0b0010 0000  
TH1= ; (This register will be loaded with the baud rate)  
TR1=1  
SCON=0x50;    0      1    0      1    0      0     0    0  
    SCON -> SM0 | SM1 | SM2 | REN | TB8 | RB8 | TI | RI  
SM0    SM1  
0    0 - 8bit Shift register   
0    1 - 8 bit UART - [Timer](http://moodlecc.vit.ac.in/mod/assign/view.php?id=3335) 1 is used  
1    0 - 9 bit UART - OSC. Freq/64 - Multiprocessor  
1    1 - 9 bit UART - [Timer](http://moodlecc.vit.ac.in/mod/assign/view.php?id=3335) 1 - Multi processor   
  
SM2 - Multiprocessor mode  
REN- Receiver Enabled   
TB8, RB8 - For Multiprocessor  
TI - Transmit flag, will SET once the transmit completed  
RI - Receive Flag, Will SET once receive completed.  
  
For Transmission   
SBUF - Serial Buffer.  
  
SBUF=a;  
  
For Reception  
a=SBUF;  
  
  
Transmit a letter 'Y' serially and continously at a baud rate of 4800.  
  
Logic  
\* Load, TMOD, TH1, TR1, SCON  
\* while(1)  
{  
    SBUF='Y';  
    while(TI==0);  
    TI=0;       
}  
  
Transmit a String "HELLO" continously at a baud rate of 9600.  
  
Two strings is controlled by the pin P1.5, When P1.5 is high, send the first string else send the second string at a speed of 9600 baud rate.

**LCD Display using 8051**

Today we are going to do the programming on LCD Display   
  
RW and Data pin Selection (SELRWD=0xffc3;)  
LCD Selection (LCDSEL =0xffc4)  
  
SELRWD can be either 0x00 or 0x01  
0x00 - Writing Mode  
0x01 - Display Mode.   
  
16 x 2 LCD in integrated into 8051 Board.  
16 characters in 2 lines.   
Each character displays in a 5\*7 matrix.  
  
There are 3 registers and a data block in LCD   
RW - Read Write , 1 for reading and 0 for writing  
  
RS - Register Select   
  
EN - Enable a Latch, First send a 1 and later send a 0 with a   
  
delay for refreshing. busycheck()  
  
void busycheck()  
{  
delay();  
SELRWD=0x00;  
}  
  
Data Block   
D7    D6    D5    D4    D3    D2    D1    D0  
0    0    1    1    1    0    0    0  
0X38 - ITS 5X7 DISPLAY WITH 2 LINES  
0X01 - cLEAR dISPLAY  
0X80 - cursor at the first position  
0xC0 - cursor at the first position in the second line.  
  
busycheck();  
\*LCDSEL=0x38;  
busycheck();  
\*LCDSEL=0x01;  
  
etc. 0x38,0x06,0x01,0x0c,0x80  
Write a 8051 C program to print your name in First line of the   
  
16x2 LCD and your registration number in the second line of   
  
the LCD.  
  
Declare the pointers to the devices, \*SELRWD, \*LCDSEL  
Declear two arrays that have your name in one array and your   
  
reg number is another array.  
unsigned char arr1[]={"Charan"};  
unsigned char arr2[]={"14BCE1104"};  
\*LCDSEL=arr1[i];  
  
define a delay() function  
define the busycheck() function. delay(); SELRWD=0x00;  
  
main  
select the addresses, 0xffc3 ,0xffc4  
set the control words (command words) 0x38,0x06,0x01,0x0c,0x80  
using a for loop display the first array to LCDSEL.  
  
the 0xc0   
and display the second array to LCDSEL.  
(for every display, make the SELRWD to 0x01)  
  
Control Words

LCD - Liquid Crystal Display  
  
16x2 means  
  
Has three control signals and one databus  
  
R/W - 1 for Read from the LCD RAM and 0 for Write to the LCD RAM  
EN - Latch Enable - have to send 1 and then a 0 signal with a particular delay to latch the data...  
RS - Register Select - the value is 1, then accepts data to be displayed. if 0, it accepts instructions for the LCD like setting font, cursor position etc.  
  
D0 to D7 - Data bus  
  
D7    D6    D5    D4    D3    D2    D1    D0  
RS    RW    EN    DL    N    F    0    0   
0    0    1    1    1    0    0    0 =0X38 - 2 LINES, 8 BIT  
  
  
1    Function Set: 8-bit, 1 Line, 5x7 Dots    0x30  
2    Function Set: 8-bit, 2 Line, 5x7 Dots    0x38\*  
3    Function Set: 4-bit, 1 Line, 5x7 Dots    0x20  
4    Function Set: 4-bit, 2 Line, 5x7 Dots    0x28  
5    Entry Mode                0x06\*  
6    Display off Cursor off            0x08      
7    Display on Cursor on            0x0E  
8    Display on Cursor off            0x0C\*      
9    Display on Cursor blinking        0x0F  
10    Shift entire display left        0x18  
11    Shift entire display right        0x1C  
12    Move cursor left by one character    0x10      
13    Move cursor right by one character    0x14      
14    Clear Display                 0x01\*  
15    cursor position on display        0X80\*  
  
one Example  
0x38,0x06,0x01,0x0c,0x80

**Board LED and Switch**

Raisonance IDE (RIDE)  
Flash Magic 3.6.x  
  
Start Windows -> Ride -> Run as administrator  
  
Go to  
Project-> New -> Create a project in My Documents (dont Create anywhere else)..  
  
Create new file and named at <filename>.c   
  
Project-> Add Node/Source Application  
  
Write the code  
  
Project -> Make All and then Build All  
  
Open the Flash Magic Software (There are two versions)  
  
Open the Version 3.68.241  
  
Go to the Device manager and Check thge COM Port, and make a node of the port number (COM1 is my case)  
  
Name of the Controller is 89V51RD2

**Seven Segment LED using 8051 Board**

Raisonance IDE (RIDE)  
Flash Magic 3.6.x  
  
Start Windows -> Ride -> Run as administrator  
  
Go to  
Project-> New -> Create a project in My Documents (dont Create anywhere else)..  
  
Create new file and named at <filename>.c   
  
Project-> Add Node/Source Application  
  
Write the code  
  
Project -> Make All and then Build All  
  
Open the Flash Magic Software (There are two versions)  
  
Open the Version 3.68.241  
  
Go to the Device manager and Check thge COM Port, and make a node of the port number (COM1 is my case)  
  
Name of the Controller is 89V51RD2

**LCD in Board**

Today we are going to do the programming on LCD Display   
  
RW and Data pin Selection (SELRWD=0xffc3;)  
LCD Selection (LCDSEL =0xffc4)  
  
SELRWD can be either 0x00 or 0x01  
0x00 - Writing Mode  
0x01 - Display Mode.   
  
16 x 2 LCD in integrated into 8051 Board.  
16 characters in 2 lines.   
Each character displays in a 5\*7 matrix.  
  
There are 3 registers and a data block in LCD   
RW - Read Write , 1 for reading and 0 for writing  
  
RS - Register Select   
  
EN - Enable a Latch, First send a 1 and later send a 0 with a delay for refreshing. busycheck()  
  
void busycheck()  
{  
delay();  
SELRWD=0x00;  
}  
  
Data Block   
D7    D6    D5    D4    D3    D2    D1    D0  
0    0    1    1    1    0    0    0  
0X38 - ITS 5X7 DISPLAY WITH 2 LINES  
0X01 - cLEAR dISPLAY  
0X80 - cursor at the first position  
0xC0 - cursor at the first position in the second line.  
  
busycheck();  
\*LCDSEL=0x38;  
busycheck();  
\*LCDSEL=0x01;  
  
etc. 0x38,0x06,0x01,0x0c,0x80  
Write a 8051 C program to print your name in First line of the 16x2 LCD and your registration number in the second line of the LCD.  
  
Declare the pointers to the devices, \*SELRWD, \*LCDSEL  
Declear two arrays that have your name in one array and your reg number is another array.  
unsigned char arr1[]={"Charan"};  
unsigned char arr2[]={"14BCE1104"};  
\*LCDSEL=arr1[i];  
  
define a delay() function  
define the busycheck() function. delay(); SELRWD=0x00;  
  
main  
select the addresses, 0xffc3 ,0xffc4  
set the control words (command words) 0x38,0x06,0x01,0x0c,0x80  
using a for loop display the first array to LCDSEL.  
  
the 0xc0   
and display the second array to LCDSEL.  
(for every display, make the SELRWD to 0x01)  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
LCD - Liquid Crystal Display  
  
16x2 means  
  
Has three control signals and one databus  
  
R/W - 1 for Read from the LCD RAM and 0 for Write to the LCD RAM  
EN - Latch Enable - have to send 1 and then a 0 signal with a particular delay to latch the data...  
RS - Register Select - the value is 1, then accepts data to be displayed. if 0, it accepts instructions for the LCD like setting font, cursor position etc.  
  
D0 to D7 - Data bus  
  
D7    D6    D5    D4    D3    D2    D1    D0  
RS    RW    EN    DL    N    F    0    0   
0    0    1    1    1    0    0    0 =0X38 - 2 LINES, 8 BIT  
  
  
1    Function Set: 8-bit, 1 Line, 5x7 Dots    0x30  
2    Function Set: 8-bit, 2 Line, 5x7 Dots    0x38\*  
3    Function Set: 4-bit, 1 Line, 5x7 Dots    0x20  
4    Function Set: 4-bit, 2 Line, 5x7 Dots    0x28  
5    Entry Mode                0x06\*  
6    Display off Cursor off            0x08      
7    Display on Cursor on            0x0E  
8    Display on Cursor off            0x0C\*      
9    Display on Cursor blinking        0x0F  
10    Shift entire display left        0x18  
11    Shift entire display right        0x1C  
12    Move cursor left by one character    0x10      
13    Move cursor right by one character    0x14      
14    Clear Display                 0x01\*  
15    cursor position on display        0X80\*  
  
one Example  
0x38,0x06,0x01,0x0c,0x80  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
#include<reg51.h>  
#include<stdio.h>  
xdata  char \*SELRWD;  
xdata  char \*LCDSEL;   
unsigned char array[]  ={"PRADEEPKUMAR"};     
unsigned char array1[] =  {"VIT Chennai"};    
unsigned char i;        
  
void delay()  
{    
int i,j;  
for(i=0;i<0xf;i++)  
for(j=0;j<0xff;j++);  
}   
        
void busycheck()  
{  
    delay();  
    \*SELRWD = 0x00;      
}  
  
  
void main()  
{    
SELRWD = 0xffc3;  
LCDSEL = 0xffc4;  
  
while(1)  
      {  
   
      busycheck();  
        \*LCDSEL = 0x38;  
        busycheck();  
        \*LCDSEL= 0x06;  
        busycheck();  
        \*LCDSEL = 0x01;  
        busycheck();  
        \*LCDSEL = 0x0c;  
        busycheck();  
        \*LCDSEL = 0x80;    
      
        for (i=0;i<15;i++)  
        {  
       busycheck();  
       delay();  
        \*SELRWD = 0x01;  
        \*LCDSEL = array[i];  
        }  
          
        busycheck();  
        \*LCDSEL = 0xC3;    
      
        for (i=0;i<10;i++)  
        {  
       busycheck();  
       delay();  
        \*SELRWD = 0x01;  
        \*LCDSEL = array1[i];  
        }  
    while(1);  
   
        }     
    
}

**Stepper Motor**

Today we are going to do the programming on LCD Display   
  
RW and Data pin Selection (SELRWD=0xffc3;)  
LCD Selection (LCDSEL =0xffc4)  
  
SELRWD can be either 0x00 or 0x01  
0x00 - Writing Mode  
0x01 - Display Mode.   
  
16 x 2 LCD in integrated into 8051 Board.  
16 characters in 2 lines.   
Each character displays in a 5\*7 matrix.  
  
There are 3 registers and a data block in LCD   
RW - Read Write , 1 for reading and 0 for writing  
  
RS - Register Select   
  
EN - Enable a Latch, First send a 1 and later send a 0 with a delay for refreshing. busycheck()  
  
void busycheck()  
{  
delay();  
SELRWD=0x00;  
}  
  
Data Block   
D7    D6    D5    D4    D3    D2    D1    D0  
0    0    1    1    1    0    0    0  
0X38 - ITS 5X7 DISPLAY WITH 2 LINES  
0X01 - cLEAR dISPLAY  
0X80 - cursor at the first position  
0xC0 - cursor at the first position in the second line.  
  
busycheck();  
\*LCDSEL=0x38;  
busycheck();  
\*LCDSEL=0x01;  
  
etc. 0x38,0x06,0x01,0x0c,0x80  
Write a 8051 C program to print your name in First line of the 16x2 LCD and your registration number in the second line of the LCD.  
  
Declare the pointers to the devices, \*SELRWD, \*LCDSEL  
Declear two arrays that have your name in one array and your reg number is another array.  
unsigned char arr1[]={"Charan"};  
unsigned char arr2[]={"14BCE1104"};  
\*LCDSEL=arr1[i];  
  
define a delay() function  
define the busycheck() function. delay(); SELRWD=0x00;  
  
main  
select the addresses, 0xffc3 ,0xffc4  
set the control words (command words) 0x38,0x06,0x01,0x0c,0x80  
using a for loop display the first array to LCDSEL.  
  
the 0xc0   
and display the second array to LCDSEL.  
(for every display, make the SELRWD to 0x01)  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
LCD - Liquid Crystal Display  
  
16x2 means  
  
Has three control signals and one databus  
  
R/W - 1 for Read from the LCD RAM and 0 for Write to the LCD RAM  
EN - Latch Enable - have to send 1 and then a 0 signal with a particular delay to latch the data...  
RS - Register Select - the value is 1, then accepts data to be displayed. if 0, it accepts instructions for the LCD like setting font, cursor position etc.  
  
D0 to D7 - Data bus  
  
D7    D6    D5    D4    D3    D2    D1    D0  
RS    RW    EN    DL    N    F    0    0   
0    0    1    1    1    0    0    0 =0X38 - 2 LINES, 8 BIT  
  
  
1    Function Set: 8-bit, 1 Line, 5x7 Dots    0x30  
2    Function Set: 8-bit, 2 Line, 5x7 Dots    0x38\*  
3    Function Set: 4-bit, 1 Line, 5x7 Dots    0x20  
4    Function Set: 4-bit, 2 Line, 5x7 Dots    0x28  
5    Entry Mode                0x06\*  
6    Display off Cursor off            0x08      
7    Display on Cursor on            0x0E  
8    Display on Cursor off            0x0C\*      
9    Display on Cursor blinking        0x0F  
10    Shift entire display left        0x18  
11    Shift entire display right        0x1C  
12    Move cursor left by one character    0x10      
13    Move cursor right by one character    0x14      
14    Clear Display                 0x01\*  
15    cursor position on display        0X80\*  
  
one Example  
0x38,0x06,0x01,0x0c,0x80