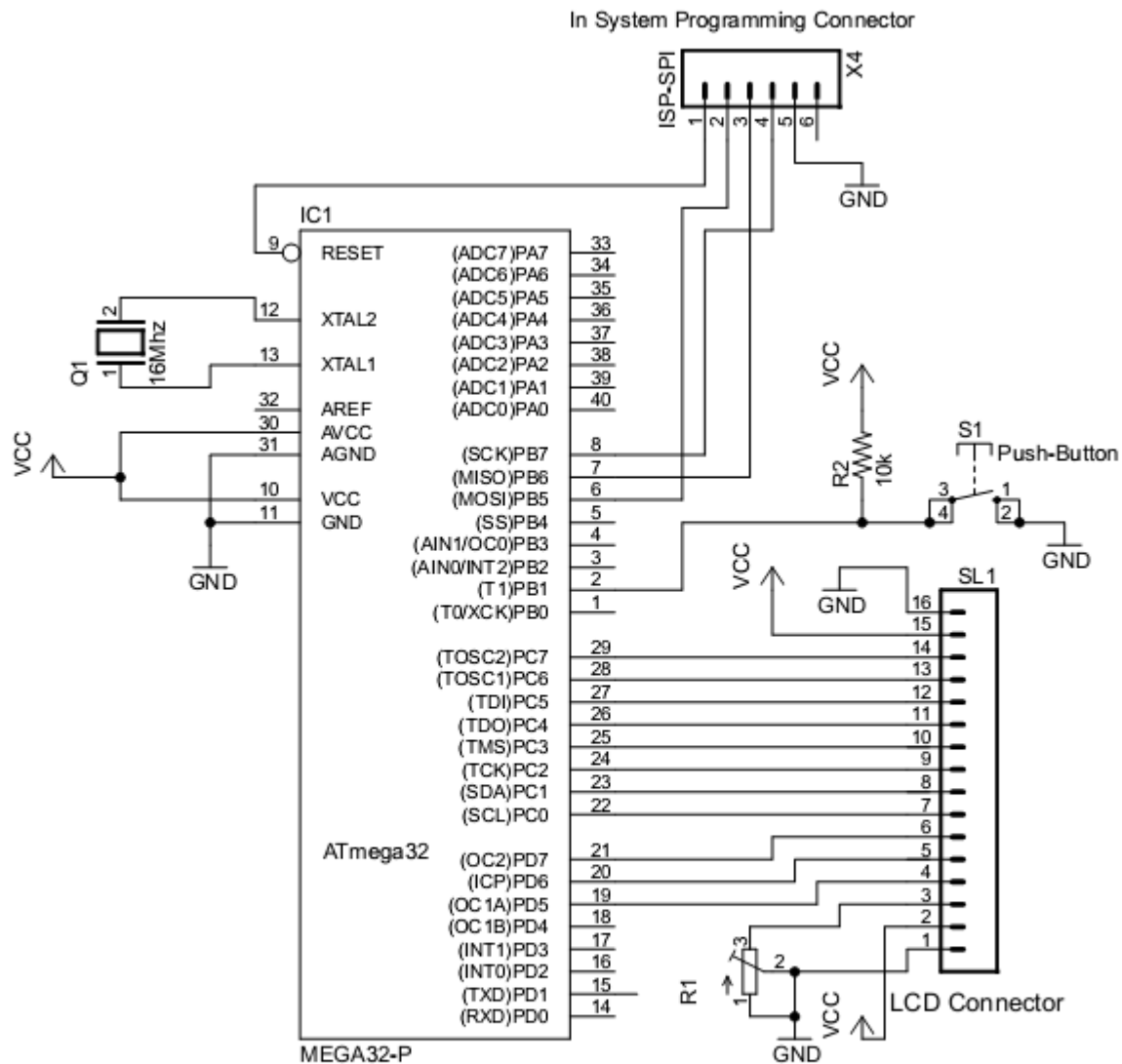


## LCD interfacing with ATmega32

This document discusses interfacing of an 16x2 intelligent LCD with ATmega32. The schematic is shown here:



Schematic for LCD Testing

CC Dharmani

**Code:**

The code is given here. Please note that if you have bought a new ATmega32 IC then you have to disable the JTAG interface which is multiplexed with the PORTC pins. JTAG can be disabled by changing the fuse values while programming, refer to datasheet for further details. If JTAG is not disabled, the LCD will remain blank. Alternatively, you can connect LCD with PORTA, and change the code accordingly to replace PORTC with PORTA.

After loading the code into microcontroller, the LCD will display a message “Hi, welcome! Have a nice Day”. When the key is pressed the message will change to “Atmega32 Board Starter’s kit”. The messages will toggle between this two whenever the pushbutton is pressed.

```
//*****  
//***** PROGRAM FOR LCD INTERFACING *****  
//*****  
//Controller:  ATmega32 (Crystal: 16 Mhz)  
//Compiler:    ImageCraft ICCAVR  
//Author:      CC Dharmani, Chennai  
//*****  
  
// *****      LCD.h : Header file *****  
  
void LCD_init(void);  
void LCD_WriteCommand (unsigned char CMD);  
void LCD_WriteData (unsigned char Data);  
void LCD_DisplayString_F(char row, char column, const unsigned  
char *string);  
void LCD_Cursor(char row, char column);  
void delay_ms(int miliSec);  
  
#define ENABLE_LCD          PORTD |= 0x80  
#define DISABLE_LCD        PORTD &= ~0x80  
  
#define SET_LCD_DATA        PORTD |= 0x20  
#define SET_LCD_CMD         PORTD &= ~0x20  
  
#define pushButton1_PRESSED  !(PINB & 0x02)  
#define pushButton1_OPEN    (PINB & 0x02)
```

```
//***** lcd.c *****

#include <iom32v.h>
#include <macros.h>
#include "LCD.h"

void port_init(void)
{
    DDRA  = 0x00;
    PORTA = 0x00;
    DDRB  = 0x00;
    PORTB = 0x00;
    DDRC  = 0xFF;
    PORTC = 0x00;
    DDRD  = 0xF0;
    PORTD = 0x00;
}

//call this routine to initialize all peripherals
void init_devices(void)
{
    CLI(); //disable all interrupts
    port_init();
    LCD_init();

    MCUCR = 0x00;
    GICR  = 0x00;
    TIMSK = 0x00; //timer interrupt sources
}

//***** MAIN FUNCTION *****
void main(void)
{
    unsigned char PB_Status=0xff,Change_Display=0,
    Change_Display1=1;

    init_devices();

    while(1)
    {
        if(Change_Display != Change_Display1)
        {
            if(Change_Display==0)
            {
                LCD_DisplayString_F(1,1," Hi, Welcome! ");
                LCD_DisplayString_F(2,1,"Have a nice Day");
            }
        }
    }
}
```

```
    }
    else
    {
        LCD_DisplayString_F(1,1," ATmega32 Board ");
        LCD_DisplayString_F(2,1," Starter's Kit ");
    }

    Change_Display1 = Change_Display;
}

CHECK_PB:
while(pushButton1_OPEN); // wait here until push button1 is
                        //pressed
delay_ms(20);           // 20ms delay for key debouncing
                        // after key-pressed
if(pushButton1_OPEN) goto CHECK_PB;
while(pushButton1_PRESSED); //wait here till the
                        //pushbutton1 is kept pressed
delay_ms(50);           // 50ms delay for key debouncing after
                        // key released

Change_Display = ~Change_Display;

}

}

//*****
//*****      LCD Functions      *****
//*****

// *****
// *** Initialize the LCD driver ***
// *****
void LCD_init(void)
{
    delay_ms(100);           // wait for 100ms
    LCD_WriteCommand (0x38); // 8 data lines
    LCD_WriteCommand (0x06); // cursor setting
    LCD_WriteCommand (0x0f); // display ON
    LCD_WriteCommand (0x01); // clear LCD memory
    delay_ms (10);           // 10ms delay after clearing LCD
}
```

```
// *****
// *** Write a command instruction to the LCD ***
// *****
void LCD_WriteCommand (unsigned char Command)
{
    SET_LCD_CMD;                // Set LCD in command mode

    PORTC = Command;            // Load data to port

    ENABLE_LCD;                  // Write data to LCD

    asm("nop");
    asm("nop");

    DISABLE_LCD;                // Disable LCD

    delay_ms(1);                // wait for 1ms
}

// *****
// *** Write one byte of data to the LCD ***
// *****
void LCD_WriteData (unsigned char Data)
{
    SET_LCD_DATA;                // Set LCD in data mode

    PORTC = Data;                // Load data to port

    ENABLE_LCD;                  // Write data to LCD

    asm("nop");
    asm("nop");

    DISABLE_LCD;                // Disable LCD

    delay_ms(1);                // wait for 1ms
}

// *****
```

```
// Display a string at the specified row and column, from FLASH
//*****
void LCD_DisplayString_F (char row, char column ,const unsigned
char *string)
{
    LCD_Cursor (row, column);
    while (*string)
        LCD_WriteData(*string++);
}

// *****
// *** Position the LCD cursor at "row", "column". ***
// *****
void LCD_Cursor (char row, char column)
{
    switch (row)
    {
        case 1: LCD_WriteCommand (0x80 + column - 1); break;
        case 2: LCD_WriteCommand (0xc0 + column - 1); break;
        default: break;
    }
}

// *****
// **** Function for delay of 1 msec (appx.) at 16Mhz ****
// *****
void delay_ms(int miliSec) //for 16 Mhz crystal
{
    int i,j;

    for(i=0;i<miliSec;i++)
        for(j=0;j<1550;j++)
        {
            asm("nop");
            asm("nop");
        }
}

// ***** END *****
```