

```
//Original Author : Manpreet Singh Minhas
//Modified version for EEE students in Cukurova Univ. 2019
//Made In Embedded systems LAB |Cukorova univ.
// This file is for 4 bit mode LCD interfacing with msp430g2553 chip
```

```
// 16x2 LCD is used
```

```
/*
```

```
 *Default Connections
```

```
P1.0 - D4 Pin11
```

```
P1.1 - D5 Pin12
```

```
P1.2 - D6 Pin13
```

```
P1.3 - D7 Pin14
```

```
P1.4 - RS Pin4
```

```
P1.5 - R/W Pin5
```

```
P1.6 - E Pin6
```

```
*/
```

```
#include <msp430g2553.h>
```

```
#define DR P1OUT = P1OUT | BIT4 // define RS high
```

```
#define CWR P1OUT = P1OUT & (~BIT4) // define RS low
```

```
#define READ P1OUT = P1OUT | BIT5 // define Read signal R/W = 1 for reading
```

```
#define WRITE P1OUT = P1OUT & (~BIT5) // define Write signal R/W = 0 for writing
```

```
#define ENABLE_HIGH P1OUT = P1OUT | BIT6 // define Enable high signal
```

```
#define ENABLE_LOW P1OUT = P1OUT & (~BIT6) // define Enable Low signal
```

```
unsigned int i;
```

```
unsigned int j;
```

```
char positionString[6]; //to be used in itoa function
```

```
void delay(unsigned int k) //watch you freq.
```

```
{  
for(j=0;j<=k;j++)  
{  
for(i=0;i<100;i++);  
}  
}
```

```
void data_write(void)
```

```
{  
ENABLE_HIGH;  
delay(2);  
ENABLE_LOW;  
}
```

```
void data_read(void)
```

```
{  
ENABLE_LOW;  
delay(2);  
ENABLE_HIGH;  
}
```

```
void check_busy(void)
```

```
{  
P1DIR &= ~(BIT3); // make P1.3 as input  
while((P1IN&BIT3)==1)
```

```
{  
data_read();  
}  
P1DIR |= BIT3; // make P1.3 as output  
}
```

```
void send_command(unsigned char cmd)  
{  
check_busy();  
WRITE;  
CWR;  
P1OUT = (P1OUT & 0xF0) | ((cmd >> 4) & 0x0F); // send higher nibble  
data_write(); // give enable trigger  
P1OUT = (P1OUT & 0xF0) | (cmd & 0x0F); // send lower nibble  
data_write(); // give enable trigger  
}
```

```
void send_data(unsigned char data)  
{  
check_busy();  
WRITE;  
DR;  
P1OUT = (P1OUT & 0xF0) | ((data >> 4) & 0x0F); // send higher nibble  
data_write(); // give enable trigger  
P1OUT = (P1OUT & 0xF0) | (data & 0x0F); // send lower nibble  
data_write(); // give enable trigger  
}
```

```
void send_string(char *s)
```

```
{  
while(*s)  
{  
send_data(*s);  
s++;  
}  
}
```

```
char *itoa(int num, char *str, int radix) {  
    char sign = 0;  
    char temp[17]; //an int can only be 16 bits long  
    //at radix 2 (binary) the string  
    //is at most 16 + 1 null long.  
    int temp_loc = 0;  
    int digit;  
    int str_loc = 0;  
    //save sign for radix 10 conversion  
    if (radix == 10 && num < 0) {  
        sign = 1;  
        num = -num;  
    }  
    //construct a backward string of the number.  
    do {  
        digit = (unsigned int)num % radix;  
        if (digit < 10)  
            temp[temp_loc++] = digit + '0';  
        else  
            temp[temp_loc++] = digit - 10 + 'A';  
        num = (((unsigned int)num) / radix);  
    } while (num > 0);  
    if (sign) temp[str_loc++] = '-';  
    temp[str_loc] = '\0';  
    return str;
```

```

} while ((unsigned int)num > 0);
//now add the sign for radix 10
if (radix == 10 && sign) {
    temp[temp_loc] = '-';
} else {
    temp_loc--;
}
//now reverse the string.
while ( temp_loc >=0 ) { // while there are still chars
    str[str_loc++] = temp[temp_loc--];
}
str[str_loc] = 0; // add null termination.
return str;
}

```

```

void send_int(int num)
{
    itoa(num, positionString, 10);
    send_string(positionString);
}

```

```

void lcd_init(void)
{
    P1DIR |= 0xFF;
    P1OUT &= 0x00;
    send_command(0x33);
    send_command(0x32);
    send_command(0x28); // 4 bit mode
    send_command(0x0E); // clear the screen
}

```

```
send_command(0x01); // display on cursor on  
send_command(0x06); // increment cursor  
send_command(0x80); // row 1 column 1  
}
```