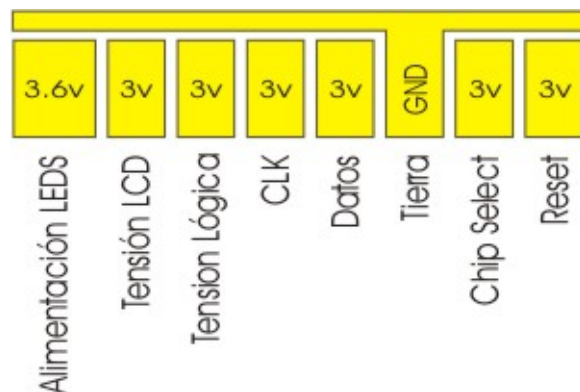


**Conexión, Inicialización  
Y  
Transmisión de datos  
Para  
LCD Nokia 1100**

## **Conexión.**

La conexión del LCD es simple y se puede realizar directamente al PIC siempre que este esté trabajando a 3v, de lo contrario deberemos agregarle un divisor o un adaptador de nivel para que no se quemé.

**La máxima tensión que soporta es 3.6.**



### **Alimentación LEDS**

Normalmente no la utilizo por el gran consumo que implica (20 mA).

### **Tensión LCD**

La tensión de alimentación de LCD la he probado hasta 4.3V y anda pero no es necesario llegar a ese nivel.

### **Tensión Lógica**

Es la tensión con la que se alimenta el chip del LCD, esta normalmente tiene un nivel proporcional al de la tensión de LCD para regular el contraste de forma manual, esto no es necesario ya que el chip cuenta con un generador de alta tensión interno que nos permite regular el contraste por software. (Lo explico en la parte de inicialización).

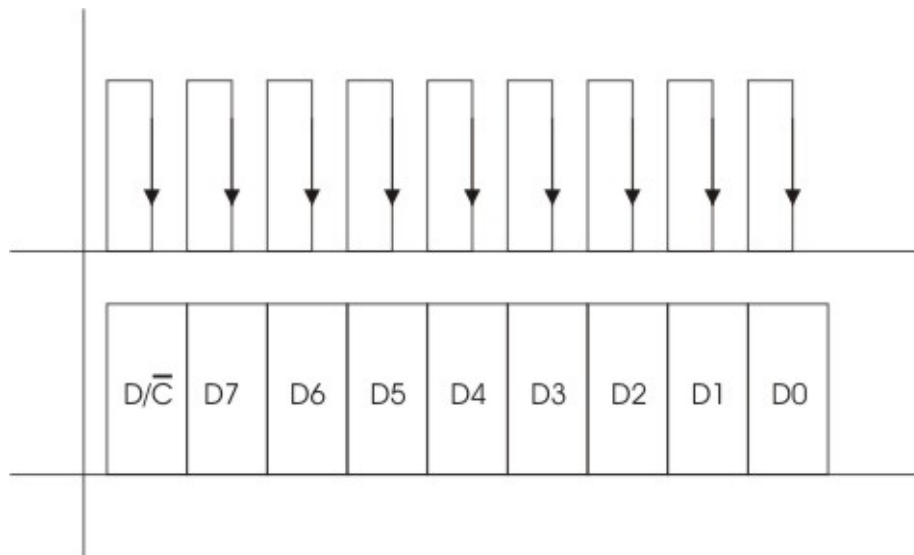
### **CLK**

El CLK del display puede llegar a 20ns entre pulsos por lo que no nos tenemos que preocupar por los retardos entre bits, en mi caso utilizando un PIC a 20MHz, la transmisión la realizo sin retardo alguno.

### **Datos**

El chip puede realizar transmisión en varios modos, pero por la disposición del hardware, solo podemos utilizar uno de ellos (transmisión de 8bits + 1 de control).

El bit de control se envía primero y luego el byte de datos. El byte de datos se debe enviar primero el más alto y en último lugar el más bajo.



En el grafico vemos que al bit lo toma el chip cuando el flanco es descendente.

Código para transmitir un byte al LCD

```
#define      sclk      RC4
#define      sda      RC5
#define      cs       RC6
#define      rst      RC7

void Lcd_Write(char cd,unsigned char c){
    char i;

    sclk = 0;
    sda = cd;
    sclk = 1;

    // enviar byte de control/dato
    for(i=0;i<8;i++){
        sclk = 0;
        if(c & 0x80)
            sda = 1;
        else
            sda = 0;
        sclk = 1;
        c <<= 1;
    }
}
```

cd: bit de control, indica si es un comando o un dato  
c: byte de dato a enviar

### **Chip Select (en cero queda seleccionado)**

El CS se utiliza para indicarle al LCD que le estamos por enviar datos, esta indicación se realiza mediante un cero lógico. Como en la mayoría de las aplicaciones no necesitamos más que un LCD, se puede conectar esta pata directamente a tierra.

### **Reset (en cero se resetea)**

El reset tiene dos puntos importantes a tener en cuenta para el correcto funcionamiento del LCD, el primero de ellos es que necesita una resistencia de por lo menos 10k. el segundo es el tiempo que se lo mantiene en bajo, según las especificaciones es necesario tener el LCD en Reset por lo menos 20ms antes de continuar con la inicialización.

## **Inicialización del LCD**

El siguiente código es el que inicializa el LCD

```
void Lcd_Init(void){

    cs = 0;
    rst = 0;
    Delay_ms(30);
    rst = 1;

    Lcd_WriteC(0x20); // VOR
    Lcd_WriteC(0x90); // EV
    Lcd_WriteC(0xA4); // Display normal
    Lcd_WriteC(0x2F); // Power control set
    Lcd_WriteC(0x40); // Y = 0
    Lcd_WriteC(0xb0);
    Lcd_WriteC(0x10); // X = 0
    Lcd_WriteC(0x0);
    Lcd_WriteC(0xc8); // invertir pantalla en vertical
    Lcd_WriteC(0xa1); // invertir pantalla en horizontal
    Lcd_WriteC(0xac); // linea inicial de pantalla
    Lcd_WriteC(0x07);
    Lcd_WriteC(0xF9); // modo icono
    Lcd_WriteC(0xaf); // prender display;
    Lcd_Clear(); // limpiar pantalla

}
```

Lo mas importante del código, son las dos líneas siguientes

```
Lcd_WriteC(0x20); // VOR  
Lcd_WriteC(0x90); // EV
```

Con el control del VOR y EV se controla el contraste del LCD, en la practica, solo variando el VOR en el rango de 0x20 a 0x29 se cambia el contraste ya que estos valores le indican al chip que nivel de tensión utilizar (ver tabla 31 de la hoja de datos).

En mi caso, puesto que utilizo 3V, lo establecí en 0x20.

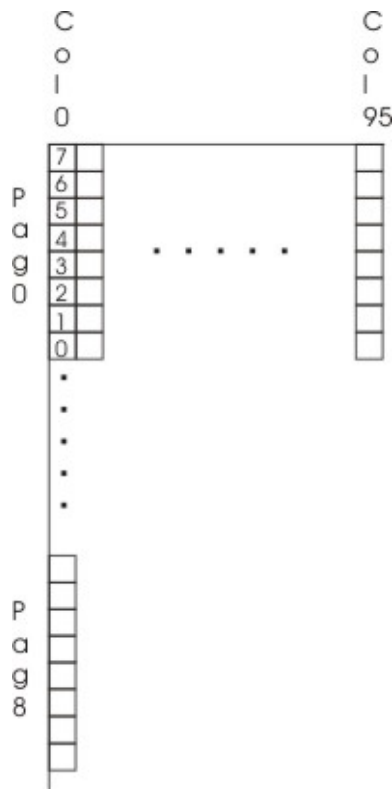
Las demás líneas de código están comentadas y en los apartados correspondientes de la hoja de datos se explican bastante bien.

Un detalle importante es la activación del modo icono, con este logramos mayor velocidad de inicialización y refresco además de menor nivel de tensión.

## **RAM del LCD**

La RAM del LCD esta organizada en 9 páginas, donde la ultima de ellas solo nos sirve para utilizar un bit.

Cada página esta dividida en 96 columnas de 8 bits, donde los bits mas alto es el píxel que primero se muestra (el de más arriba)



```
Lcd_WriteC(0x40); // Y = 0
Lcd_WriteC(0xb0);
```

El comando 0x40 seguido del comando 0xBX selecciona la pagina. Donde X es el número de página. Las páginas se numeran de cero a ocho.

```
Lcd_WriteC(0x10); // X = 0
Lcd_WriteC(0x00);
```

El comando 0x10 indica que numero de columna se esta por acceder, el segundo byte es dicho numero de columna

Vale aclarar que el numero de columna y pagina se incrementan automáticamente cada vez que enviamos un dato, por lo que si seleccionamos la primera posición de pantalla, bastará con enviar 8\*96 bytes para pintar toda la pantalla.

El siguiente código limpia la pantalla

```
void Lcd_Clear(void){
    unsigned int i;

    Lcd_WriteC(0x10);
    Lcd_WriteC(0x00);
    Lcd_WriteC(0xB0);

    for(i=0;i<864;i++)
        Lcd_WriteD(0);

}
```

Espero que esta información le aclare algunas dudas.

Atte.

Alberto