

# Informe tercera práctica: Display LCD

Grupo SDMI 22E

David Guerra

Roman Valls

Daniel Clemente

---

Se pide hacer la conversión analógico/digital, como en la práctica anterior, pero mostrando el resultado en el display LCD de 16x2 caracteres y en forma de porcentaje en relación al nivel máximo de corriente.

## Descripción del conexionado:

Utilizamos el puerto P1 para conectar el display LCD, éste se encuentra en el conector JP2. Al conectarlo, las patillas 4, 6, 8 y 10 del JP2 (P1.4 a P1.7) se conectan a las patillas 11, 12, 13 y 14 del conector del LCD (D4 a D7); el pin 7 del JP2 (P1.2) se conecta al pin 6 del conector LCD (entrada E) y el pin 3 (P1.0) va al pin 4 del LCD (C/D). El resto de patillas del P1 van al aire (P1.1 y P1.3).

En el conector del LCD conectamos las patillas restantes de la siguiente manera: las patillas 1, 5, 7, 8, 9 y 10 (GND, R/W, D0, D1, D2 y D3) se conectan a GND (patilla 1 del JP2); la patilla 2 (Vcc) se conecta a Vcc (patilla 2 del JP2); la patilla 3 (VEE) a la salida variable del potenciómetro que regula el contraste del LCD, este potenciómetro además, también va conectado a Vcc y GND en el JP2.

## Parte software

### mainlcd.c:

En esta sesión, hemos reusado bastante código de la práctica anterior (convertor AD). Las inicializaciones del convertor AD son las mismas que las de la práctica anterior, añadiendo previamente una llamada a la función INICIALITZA\_LCD() (implementada en el fichero Lcd.c) que prepara el display para recibir datos.

```
wConversio=0xAA;
bADEnCurs=1; // Empezamos convirtiendo
(...)
INICIALITZA_LCD();
asm{ di; }
(...)
```

A continuación, borramos el contenido presente en el LCD (con CLRSCR\_LCD), nos desplazamos mediante GOTOXY a la posición (2,1). Mediante un bucle, enviamos un mensaje definido anteriormente (missatge[10]="SDMI jarl") carácter a carácter hacia el display.

```
CLRSCR_LCD();
GOTOXY(2,1);
i=0;
while (i<strlen(missatge)) {
    DADA_LCD(missatge[i]);
    i++;
}
```

Finalmente, la parte más significativa de la práctica consiste en escribir en el LCD el valor obtenido en la lectura del convertor AD y expresarlo con un porcentaje. Para ello hemos usado divisiones y módulos base 10 para convertir el valor de la variable wConversio a %.

```
if (bADEnCurs) {
    //ioport1=wConversio;

    // wConversio; // 0 a 255

    wConversio=wConversio&0xFF;
    percent=(wConversio*100)/255;
    percent=percent&0xFF;

    missatge[0]= (percent/100) %10 +'0';
    missatge[1]= (percent/10) %10 +'0';
    missatge[2]= (percent/1) %10 +'0';
    missatge[3]='%';
}
```

```

GOTOXY(6,2);    // Segunda línea
i=0;
while (i<=3) {
    DADA_LCD(missatge[i]);
    ++i;
}

```

### **stdisp.a96:**

Las modificaciones en el fichero ensamblador se reducen a añadir la dirección del VI a modificar (del conversor AD) y a incluir la rutina correspondiente (RSIADDone()) en mainlcd.c.

```

cseg at 0d020h
    br AD_CONVERSION

```

### **RSI en mainlcd.c:**

```

void RSIADDone()
{
    asm {di;}

    wsr=0x0; regtemp=ad_result;
    wConversio=regtemp>>8;
    regtemp = regtemp & 0xF7; // Desactivar el bit de aviso
    wsr=0xf; ad_result=regtemp;

    asm {ei;}
}

```