

Pantalla LCD

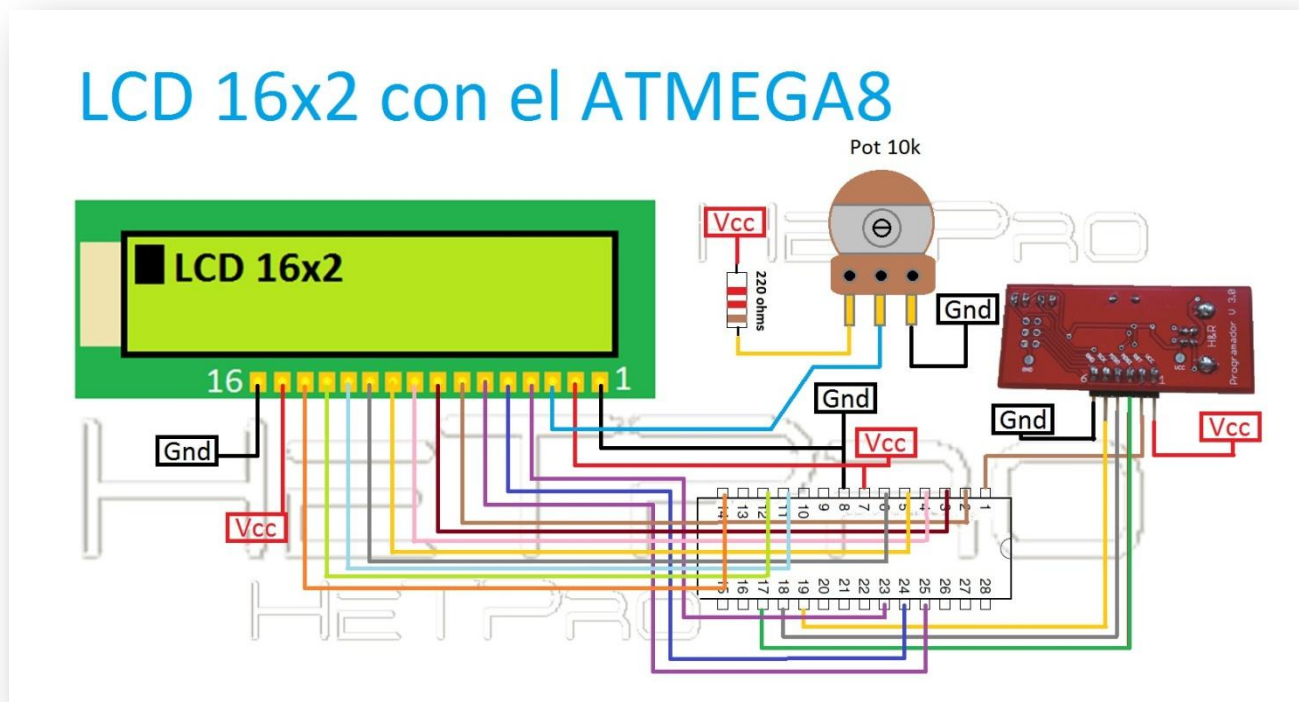


Pantalla LCD

Descripción

En este ejemplo se trabajara con una pantalla LCD de 16x2, se verán un par de funciones de la misma que se realizaran como escribir texto o limpiar la pantalla. En este caso para evitar un grado de complejidad mayor se trabajara la pantalla LCD con una comunicación de 8 bits, la cual es más sencilla que la de 4 bits. También se hará uso de un potenciómetro como divisor de voltaje para regular el PIN Vee el cual controla el contraste de la pantalla.

Diagrama Esquemático



Materiales

- 1 Pantalla LCD 16x2 JHD-162A
- 1 Resistencia de 220 Ohms
- 1 Potenciómetro de 10 kOhms
- 1 Microcontrolador ATmega8
- Programador USBasp V3.0

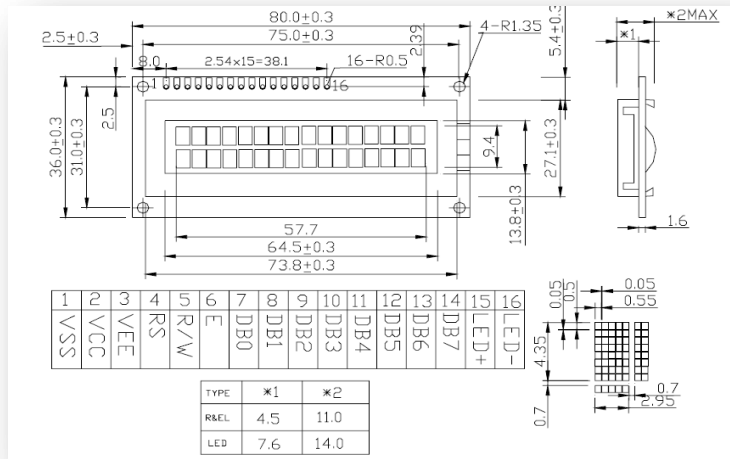
Introducción

Pantalla LCD

Una pantalla de cristal liquido o LCD es una pantalla delgada y plana formada por un determinado número de pixeles monocromos (para este caso), colocados delante de una fuente luminosa. Una de las principales características de las pantallas LCD es su bajo consumo de energía eléctrica.

Pantalla JHD-162A

Configuración de los pines, tamaño de la pantalla y de los caracteres.



Controlador de la pantalla JHD-162A, HD44780

El controlador HD44780 es el circuito que se encuentra en la pantalla JHD-162A y es el que controla el manejo del LCD, en este es recomendable revisar la hoja de datos del mismo ya que explica los modos de manejo del LCD y las instrucciones de este. En la siguiente imagen podemos ver una de las páginas de la hoja de datos del controlador en donde se pueden ver varias de las instrucciones que el LCD maneja.

Instruction	Code										Description	Execution Time (max) (when f_{op} or f_{osc} is 270 kHz)
	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Clears entire display and sets DDRAM address 0 in address counter.	
Return home	0	0	0	0	0	0	0	0	1	—	Sets DDRAM address 0 in address counter. Also returns display from being shifted to original position. DDRAM contents remain unchanged.	1.52 ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	S	Sets cursor move direction and specifies display shift. These operations are performed during data write and read.	37 μ s
Display on/off control	0	0	0	0	0	0	1	D	C	B	Sets entire display (D) on/off, cursor on/off (C), and blinking of cursor position character (B).	37 μ s
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	—	—	Moves cursor and shifts display without changing DDRAM contents.	37 μ s
Function set	0	0	0	0	1	DL	N	F	—	—	Sets interface data length (DL), number of display lines (N), and character font (F).	37 μ s
Set CGRAM address	0	0	0	1	ACG	ACG	ACG	ACG	ACG	ACG	Sets CGRAM address. CGRAM data is sent and received after this setting.	37 μ s
Set DDRAM address	0	0	1	ADD	ADD	ADD	ADD	ADD	ADD	ADD	Sets DDRAM address. DDRAM data is sent and received after this setting.	37 μ s
Read busy flag & address	0	1	BF	AC	AC	AC	AC	AC	AC	AC	Reads busy flag (BF) indicating internal operation is being performed and reads address counter contents.	0 μ s

Programa en C

```
#include <avr/io.h>
#include <util/delay.h>

#define Enable_On          PORTC|=_BV(PC2)    //Definiciones
#define Enable_Off        PORTC&=~_BV(PC2)
#define RS_On             PORTC|=_BV(PC0)
#define RS_Off            PORTC&=~_BV(PC0)
#define RW_On             PORTC|=_BV(PC1)
#define RW_Off            PORTC&=~_BV(PC1)

#define Data               PORTD
#define DelayL             _delay_ms(5);
int i=0;

void PORT_init (void){                                //Función para inicializar puertos
    DDRC=0x07;
    DDRD=0xFF;
    PORTC=0x00;
    PORTD=0x00;
}

void LCD_init (void){                                //Función para inicializar el LCD
    Data=0x0F;
    Enable_On;
    DelayL;
    Enable_Off;
    Data=0x00;
    RS_On;
    DelayL;
}

void WriteLCD(char text[15]){                          //Función para escribir en el LCD
    RS_On;
    for (i=0;i<16; i++){
        Data=text[i];
        Enable_On;
        DelayL;
        Enable_Off;
        DelayL;
    }
    i=0;
    Data=0x00;
    RS_Off;
}
```

```

void ClearLCD(void){                                     //Función para limpiar la pantalla del LCD
    Data=0x01;
    Enable_On;
    DelayL;
    Enable_Off;
}

int main (void){                                       //Inicio del programa

    PORT_init();
    LCD_init();

while(1){
    WriteLCD("Hola mundo");                             //Escribir en el LCD
    _delay_ms(1000);

    ClearLCD();                                         //Limpiar la pantalla del LCD

    _delay_ms(500);

}
}

```

Detalles del programa

```

#define Enable_On      PORTC|=_BV(PC2)
#define Enable_Off    PORTC&=~_BV(PC2)
#define RS_On         PORTC|=_BV(PC0)
#define RS_Off        PORTC&=~_BV(PC0)
#define RW_On         PORTC|=_BV(PC1)
#define RW_Off        PORTC&=~_BV(PC1)

#define Data          PORTD
#define DelayL        _delay_ms(5);

```

La directiva *#define* define una macro que proporciona un mecanismo de reemplazo de código por una secuencia de caracteres, en este caso la sintaxis del *#define* es:

```
#define macro_identificador <secuencia-de-tokens>
```

tomando como referencia uno de las definiciones previamente usadas podemos ver que lo que se está haciendo con:

```
#define Enable_On PORTC|=_BV(PC2)
```

es asignándole a `Enable_On` la sección de código `PORTC|=_BV(PC2)` la cual activa el bit 2 del registro `PORTC`.

```
void PORT_init (void){  
    DDRC=0x07;  
    DDRD=0xFF;  
    PORTC=0x00;  
    PORTD=0x00;  
}
```

Esta es una función en la que dentro de ella se configuran los registros de los puertos

```
void LCD_init (void){  
    Data=0x0F;  
    Enable_On;  
    DelayL;  
    Enable_Off;  
    Data=0x00;  
    RS_On;  
    DelayL;  
}
```

Para inicializar la pantalla LCD tenemos que seguir una serie de instrucciones las cuales como se puede apreciar en el código son las siguientes que se encuentran representadas en la tabla.

Instruc \ Pin	E	R/W	RS	Data DB0:7
1	0	0	0	0x0F
2	1	0	0	0
3	0	0	0	0
4	0	0	1	0

Después de inicializar la pantalla un puntero debe aparecer en la misma, y nos indica que esta lista para recibir los caracteres.

```

void WriteLCD(char text[15]){
    RS_On;
    for (i=0;i<16; i++){
        Data=text[i];
        Enable_On;
        DelayL;
        Enable_Off;
        DelayL;
    }
    i=0;
    Data=0x00;
    RS_Off;
}

```

Para escribir en el LCD es necesario seguir una serie de instrucciones, las cuales nos llevaran a escribir el texto, que queramos ver en el LCD, carácter por carácter, por lo tanto tenemos que separar nuestro texto en caracteres, esto lo logramos haciendo la función con una entrada de tipo char (carácter), pero en este caso va a ser un vector, esto es una serie de caracteres o de chars, por lo tanto cuando llamemos la función y a la entrada de la misma pongamos algún texto este se guardara en un vector de caracteres, por ejemplo en texto "Hola Mundo":

Posicion	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Caracter	H	o	l	a		M	u	n	d	o						

Para habilitar la entrada de texto, se activa el pin de RS, a continuación el código inicia un ciclo for de 0 a 15, para lograr desplazarnos por las 16 posiciones que tiene la pantalla. Al iniciar el ciclo se le asigna a la variable Data el carácter que está en el vector text en la posición j, por ejemplo en el primer caso inicia con j=0 en caso de haber escrito el texto "Hola Mundo", el vector quedaría como en la tabla superior y cómo podemos ver la posición 0 tendría el carácter "H", después se le da un pulso al Enable un retardo y el ciclo for continua. Al terminar se limpian las variables y el RS se desactiva

Instruc \ Pin	E	R/W	RS	Data DB0:7
1	0	0	1	0
2	0	0	0	Caracter
3	1	0	0	0
4	0	0	0	0

```

void ClearLCD(void){
    Data=0x01;
    Enable_On;
    DelayL;
    Enable_Off;
}

```

Para limpiar la pantalla, se escribió la función ClearLCD la cual inicializa el puntero de la pantalla y borra todos los caracteres escritos en ella. Para limpiar la pantalla se realizaron las siguientes instrucciones.

Instruc \ Pin	E	R/W	RS	Data DB0:7
1	0	0	0	0x01
2	1	0	0	0
3	0	0	0	0

```

int main (void){

```

```

    PORT_init();
    LCD_init();

```

La función main o principal, a la cual al correr el micro es la primera en ejecutar, en este caso llama a las funciones inicializadoras de los puertos y del LCD.

```

while(1){
    WriteLCD("Hola mundo");           //Escribir en el LCD
    _delay_ms(1000);

    ClearLCD();                       //Limpiar la pantalla del LCD
    _delay_ms(500);
}

```

Dentro del ciclo while infinito, se llama a la función WriteLCD la cual escribe el texto "Hola Mundo" en el LCD de la manera que se explico previamente, a continuación damos un retardo de 1000 ms y posteriormente se limpia la pantalla llamando a la función ClearLCD a lo que esperamos 500 ms, después de esto el ciclo inicia de nuevo y el texto vuelve a aparecer.