

Periféricos Básicos

Visualizador LCD

Es frecuente la necesidad de mostrar mensajes que tienen que ver con el estado de algo o el valor de un instrumento de medida electrónico. Para estos casos la utilización de una pantalla de cristal liquido LCD ofrece como ventaja con respecto a los displays de 7 segmentos, su bajo consumo de corriente y la no necesidad de multiplexar además de soportar caracteres alfanuméricos en el estándar ASCII.

Visualizador LCD LM016L

Esta pantalla de cristal liquido es una de las mas comunes consta de dos líneas de visualización de 16 caracteres cada una, donde cada caracteres esta conformado por una matriz de caracteres de 5x7 puntos. El circuito integrado controlador de este LCD es el Hitachi 44780.

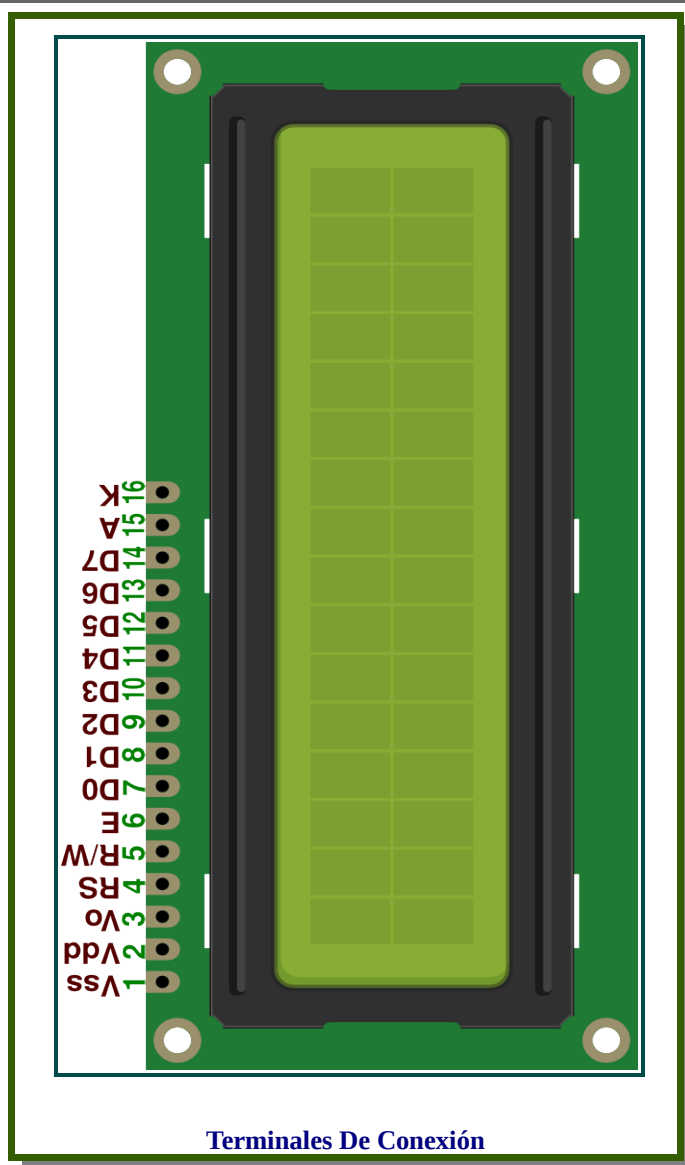
Tiene como características generales:

- Consumo de potencia muy reducido de aproximadamente 7,5 mW (sin uso de luz posterior BackLight).
- Soporte de caracteres en el estándar ASCII, caracteres japoneses, griegos y símbolos matemáticos.
- Desplazamiento de caracteres en sus líneas, ya sea a la izquierda o la derecha.
- Memoria que soporta 40 caracteres por linea.
- Movimiento del cursor y cambio de aspecto.
- Permite que el usuario pueda programar 8 caracteres propios.
- Permite dos configuraciones de conexión para 8 y 4 bits en el manejo de datos e instrucciones de control.

Terminales del LCD LM016L

Este consta de 14 o 16 terminales en caso de contar con iluminación de pantalla backlight.

La siguiente tabla y gráfico muestra la descripción y disposición de los terminales.



Terminales De Conexión

Ter	Sim	Descripcion
1	Vss	Tierra del circuito
2	Vdd	Alimentación a 5v
3	Vo	Ajuste del contraste de la pantalla vara de 0 - 5v
4	RS	Selección del registro control - datos. RS = 0 - Control RS = 1 - Datos
5	R/W	Lectura o escritura en el LCD R/W = 0 - Escritura R/W = 1 - Lectura
6	E	Habilitación del LCD E = 0 - Desconectado E = 1 - Conectado
7	D0	Bit de dato 0
8	D1	Bit de dato 1
9	D2	Bit de dato 2
10	D3	Bit de dato 3
11	D4	Bit de dato 4
12	D5	Bit de dato 5
13	D6	Bit de dato 6
14	D7	Bit de dato 7
15	A	Alimentación de luz de fondo backlight puede ser 3.5v o 5v
16	K	Tierra del backlight

Tabla De Terminales De Conexión

Modos de funcionamiento

La pantalla de cristal liquido tiene 3 modos de funcionamiento principales

Modo comando: Se envía por el bus de datos (D0 – D7) instrucciones de operación como “Borrar Display”, “Mover cursor”, “Desplazar a la izquierda”, etc. Para trabajar en el modo de operación comando, el terminal RS debe estar a “0”. El terminal R/W debe

también estar en “0” para indicar que se esta realizando una operación de escritura. Las operaciones en este modo tardan aproximadamente 1,64 ms.

Modo Carácter o Dato: Se envía por el bus de datos el carácter a escribir en la DDRAM. Para trabajar en este modo de operación, el terminal RS debe ser “1” y el terminal R/W debe ser “0” por que se esta realizando una operación de escritura. Las operaciones en este modo tardan aproximadamente 40 us.

Modo LCD Ocupada: En el bit 7 “D7” del bus de datos del LCD se informa al microcontrolador el estado de la pantalla, para ello se lee el bus de datos con RS en “0” y R/W en “1”, si el D7 es “1” se indica que en ese momento el LCD esta ocupado realizando operaciones y que no puede aceptar instrucciones o datos nuevos. Se espera hasta que D7 sea “0” para poder enviar datos o instrucciones nuevamente. El bit de LCD ocupada se llama busy flag (BF). Este modo de operación se ha ideado para evitar posibles problemas con los tiempos de espera entre operaciones del LCD.

Comandos de control.

Los comandos de control son una serie especial de instrucciones que gobiernan el LCD, estos comandos son enviados por el bus de datos. La *tabla de comandos* muestra los comandos que admite el modulo LCD.

Los comandos en detalle de la tabla son:

Clear Display (00000001) : Borra la pantalla y devuelve el cursor a la posición inicial (dirección 0 de la DDRAM).

Return Home (0000001x) : Cursor a casa. Coloca el cursor en la posición inicial de la pantalla fila 1 carácter 1 (dirección 00h de la DDRAM).

Entry Mode Set (000001 I/D S) : Modo Entrada. Estable las características de la forma de escribir los datos en pantalla Increment/Decrement y Shift

- **S=0** La información visualizada en pantalla no se desplaza al escribir un nuevo carácter.
- **S=1** La información se desplaza al escribir un nuevo carácter. El sentido de desplazamiento en la pantalla se indica con el bit I/D.
- **I/D = 1** Incremento automático de la posición del cursor. La posición de la DDRAM se incrementa automáticamente tras cada lectura o escritura de la misma.
- **I/D = 0** Decremento de la posición del cursor. Se decrementa los posición de la DDRAM.

Display Control (00001 D C B) Control de pantalla:

- **B=0 Blink OFF**, no hay efecto de parpadeo en el cursor.
- **B=1 Blink ON**, efecto de parpadeo con el cursor rectangular.
- **C=0 Cursor OFF**, el cursor no se visualiza.
- **C=1 Cursor ON**, el cursor se visualiza.
- **D=0 Display OFF**, el display se apaga.
- **D=1 Display ON**, el display se enciende.

Cursor and Display Shift (0001 S/C R/L x x) Control de los desplazamientos del cursor y de la pantalla:

- **R/L = 0** A la izquierda
- **R/L = 1** A la derecha
- **S/C = 0** El efecto de desplazamiento se aplica solo al cursor sin afectar el contenido de la **DDRAM**.
- **S/C = 1** El efecto de desplazamiento se aplica en todo el display.

Function Set (001 DL N F x x) Características de control de hardware:

- **F = 0** Caracteres de 5 x 7 puntos
- **F = 1** Caracteres de 5 x 10 puntos
- **N = 0** Pantalla de una línea
- **N = 1** Pantalla de dos líneas
- **DL = 0** Modo de comunicación a **4 bits**. Se le indica al display que se utilizarán las líneas **DB4 – DB7** para el envío de datos y que se hará enviando primero el nibble alto.
- **DL = 1** Modo de comunicación a **8 bits**.

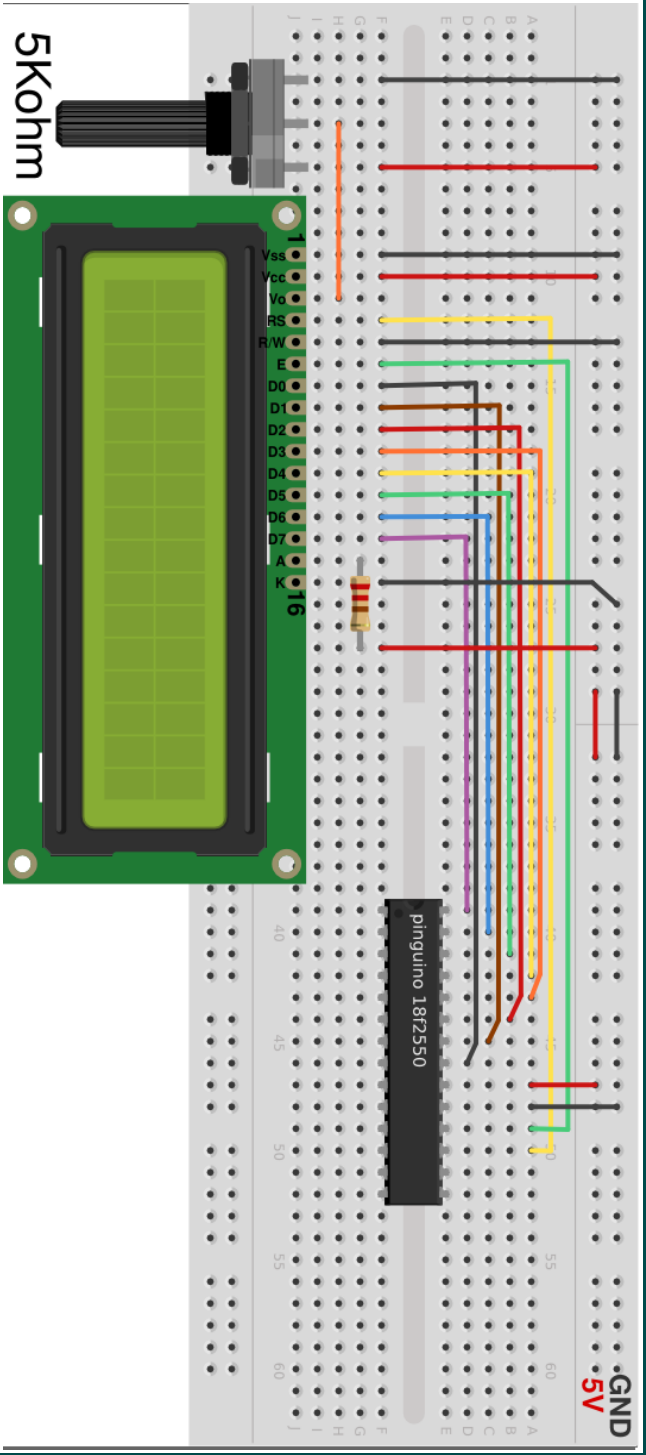
Set CGRAM Address (01dddddd): se escribe sobre la dirección **CGRAM** señalada.

Set DDRAM Address (1ddddddd) : Esta instrucción modifica la dirección en la que se apunta en la **DDRAM**.

Read Busy Flag. Lee el **BF** indicando si hay una operación interna en proceso, además lee el contenido de la **DDRAM** apuntada.

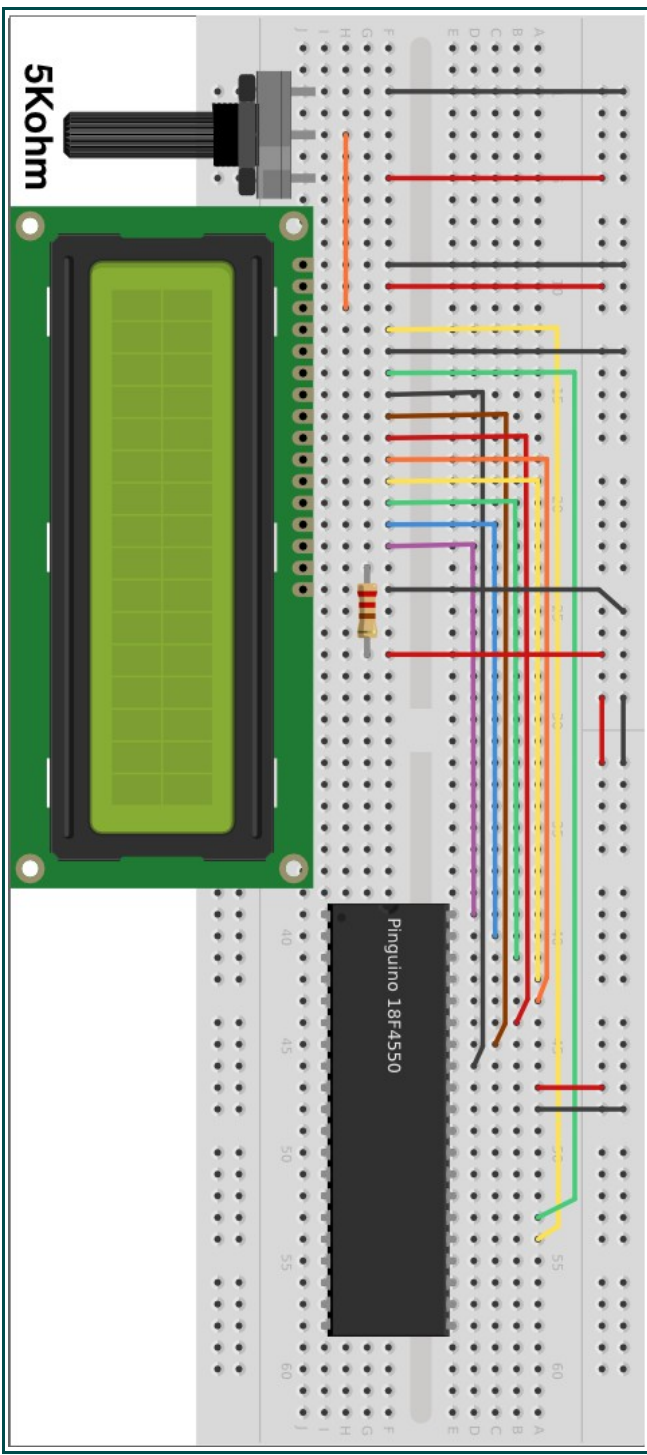
Comando	RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Clear Display	0	0	0	0	0	0	0	0	0	1
Return Home	0	0	0	0	0	0	0	0	1	*
Entry Mode Set	0	0	0	0	0	0	0	1	I/D	S
Display Control	0	0	0	0	0	0	1	D	C	B
Cursor and Display Shift	0	0	0	0	0	1	S/C	R/L	*	*
Function Set	0	0	0	0	1	DL	N	F	*	*
Set CGRAM Address	0	0	0	1	CGRAM Address					
Set DDRAM Address	0	0	1	DDRAM Address						
Read Busy Flag	0	1	BF	DDRAM Address						
Write RAM	1	0	Write Data							
Read RAM	1	1	Read Data							

Tabla De Comandos



This diagram illustrates the breadboard wiring for a PIC18F2550 microcontroller connected to an LCD display. The PIC18F2550 is positioned vertically on the right side of the breadboard, with its pins labeled A through K. The LCD is placed horizontally on the left. The wiring includes: a 5kΩ potentiometer at the top left; power connections (Vss, Vcc, Vo, RS, R/W) to the LCD; data bus connections (D0-D7) between the PIC and LCD; and control signal connections (A, K) for the PIC. A 5V GND connection is shown at the bottom right. The breadboard grid is labeled with letters A-J and numbers 40-60.

Montaje LCD con pinguino PIC18F2550



This diagram illustrates the breadboard wiring for a PIC18F4550 microcontroller connected to an LCD display. The PIC18F4550 is positioned vertically on the right side of the breadboard, with its pins labeled A through J. The LCD is placed horizontally on the left. The wiring includes: a 5kΩ potentiometer at the top left; power connections (Vss, Vcc, Vo, RS, R/W) to the LCD; data bus connections (D0-D7) between the PIC and LCD; and control signal connections (A, K) for the PIC. A 5V GND connection is shown at the bottom right. The breadboard grid is labeled with letters A-J and numbers 40-60.

Montaje LCD con pinguino PIC18F4550

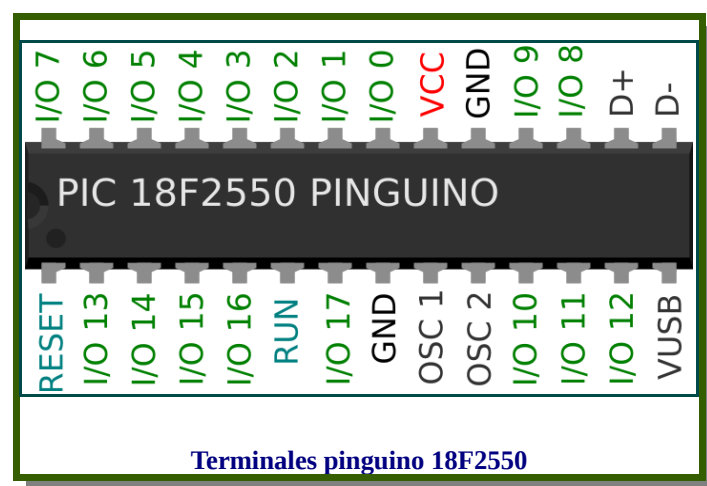
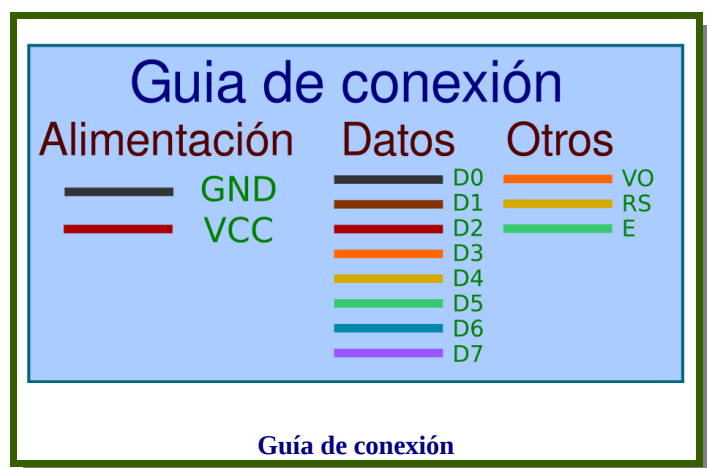
Guías de montaje

Los montajes corresponden al display LCD conectado a pinguino con el microcontrolador **18F2550** y al **18F4550**. En este esquema no se tiene en cuenta los otros componentes que acompañan a pinguino como son el cristal, el pulsador de reset entre otros.

La [guía de conexión](#) indica los colores utilizados en las los cables empleados en el protoboard, la secuencia empleada en los cables de datos es equivalente al empleado en los cables ribbon o la que sigue el código de colores en las resistencias.

La gráfica [terminales pinguino 18F2550](#) indica como se referencia estos dentro del software, de manera que se tengan en cuenta cuando se este programando.

La gráfica [terminales pinguino 18F4550](#) indica como se referencia estos dentro del software, de manera que se tengan en cuenta cuando se este programando.



Programación en Pinguino.

Las versiones mas recientes del IDE de pinguino ya traen consigo la librería para el manejo del LCD y facilitan la elaboración de programas con este dispositivo.

Las instrucciones mas relevantes para el manejo del

LCD en pinguino son las siguientes:

Modos de 8 y 4 bits

Permite configura el LCD en el modo de transmisión de datos o instrucciones ya sea en el modo de **4** o **8 bits**. Los primeros valores corresponden a la conexión de los terminales de pinguino con los terminales **RS** y **E** del LCD, los siguientes corresponde a los terminales de datos **D0 – D7** en el modo de 8 bits y **D4 – D8** en el modo de 4 bits.

//Modo de 8 bits

```
lcd(8, 9, 0, 1, 2, 3, 4, 5, 6, 7); // RS, E, D0 ~ D7
```

//Modo de 4 bits

```
lcd(8, 9, 0, 1, 2, 3, 0, 0, 0, 0); // RS, E, D4 ~ D8
```

Inicializar LCD

Se indica el numero de lineas y tamaño del los puntos en el LCD. Para este caso tenemos un LCD de dos filas y el mínimo tamaño de puntos en el LCD.

//formato del LCD

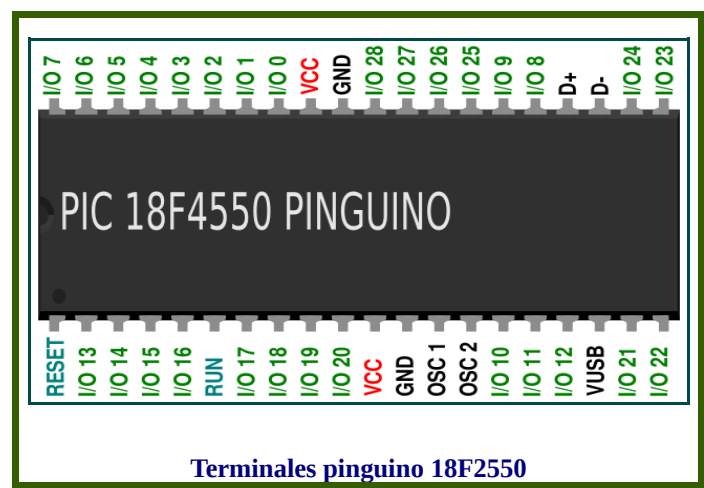
```
lcd.begin(2, 0);
```

Colocar el cursor en la posición inicial

Colocar el cursor en la posición inicial del LCD, esto seria en la primer fila y primer carácter.

//Cursor a posicion inicial

```
lcd.home();
```



Imprime texto en pantalla

Coloca un texto desde la posición en la que este el cursor.

//Imprimir texto en pantalla

```
lcd.print("Hola Pinguino");
```

Ubicar cursor en una posición dada

Coloca el cursor en la una posición que se indique, en este caso el primer parámetro corresponde a la columna o numero de carácter y el segundo parámetro corresponde a la fila.

```
//Posicion del cursor fila columna
```

```
lcd.setCursor(0, 1);
```

Imprime un valor entero en pantalla

Esta instrucción permite visualizar un valor entero en pantalla desde la posición en la que este el cursor. Sus parámetros son el valor a imprimir y la base numérica a utilizar, soporta base 2 o binaria, base 10 o decimal, y base 16 o hexadecimal. El ejemplo muestra el valor en forma decimal

```
//Imprime un valor entero en pantalla.
```

```
lcd.printNumber(133, 10);
```

Imprime un valor decimal en pantalla.

Esta instrucción permite visualizar un valor decimal en pantalla desde la posición en que este el cursor. Sus parámetros son el valor a imprimir y el numero de cifras significativas deseadas.

```
//Imprime un valor decimal en pantalla.
```

```
lcd.printFloat(15.85, 2);
```

Limpiar pantalla.

Esta instrucción limpia la pantalla LCD.

```
//Limpia pantalla.
```

```
lcd.clear();
```

No mostrar datos en pantalla.

Esta instrucción desactiva la visualización en pantalla.

```
//Desactivar visualizacion
```

```
lcd.noDisplay();
```

Mostrar datos en pantalla.

Esta instrucción activa la visualización en pantalla.

```
//Activar visualizacion
```

```
lcd.display();
```

Ocultar cursor.

Esta instrucción desactiva la visualización del cursor en pantalla.

```
//Desactivar visualizacion del cursor
```

```
lcd.noCursor();
```

Ver cursor.

Esta instrucción activa la visualización del cursor en pantalla.

```
//Activar visualizacion
```

```
lcd.cursor();
```

Parpadeo de cursor.

Esta instrucción activa la opción de parpadeo del cursor en pantalla.

```
//Parpadeo en el cursor
```

```
lcd.blink();
```

Desactivar parpadeo del cursor.

Esta instrucción desactiva la opción de parpadeo del cursor en pantalla.

```
//Desactiva parpadeo en el cursor
```

```
lcd.noBlink();
```

Desplazamiento de caracteres hacia la izquierda

Esta instrucción desplaza hacia la izquierda los caracteres que estén escritos en el LCD.

```
//Desplazamiento hacia la izquierda
```

```
lcd.scrollDisplayLeft();
```

Desplazamiento de caracteres hacia la derecha

Esta instrucción desplaza hacia la derecha los caracteres que estén escritos en el LCD.

```
//Desplazamiento hacia la derecha
```

```
lcd.scrollDisplayRight();
```

Desplazamiento de caracteres hacia la derecha

Esta instrucción desplaza hacia la derecha los caracteres que estén escritos en el LCD.

```
//Desplazamiento hacia la derecha
```


`lcd.scrollDisplayRight();`

Desplazamiento de texto de derecha a izquierda

Esta instrucción desplaza el texto de derecha a izquierda desde la posición del cursor.

`//Desplazamiento de derecha a izquierda`

`lcd.leftToRight();`

Desplazamiento de texto de izquierda a derecha

Esta instrucción desplaza el texto de izquierda a derecha desde la posición del cursor.

`//Desplazamiento de izquierda a derecha`

`lcd.rightToLeft();`

Justificar texto a la derecha

Esta instrucción justifica el texto a la derecha desde la posición del cursor.

`//Justificar a la derecha`

`lcd.autoscroll();`

Justificar texto a la izquierda

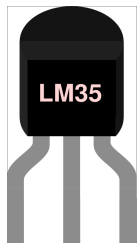
Esta instrucción justifica el texto a la izquierda desde la posición del cursor.

`//Justificar a la izquierda`

`lcd.noAutoscroll();`

```
1  /*****
2  Ejemplo basico para utilizacion de la pantalla LCD en pinguino
3  lcd00.pde|
4  *****/
5  void setup(){
6      //Modo de 8 bits
7      lcd(8, 9, 0, 1, 2, 3, 4, 5, 6, 7); // RS, E, D0 ~ D7
8      //Formato LCD
9      lcd.begin(2, 0);
10 }
11 void loop(){
12     //Colocar cursor en la posicion incial del LCD
13     lcd.home();
14     //Imprime texto en pantalla
15     lcd.print("Hola pinguino");    // Texto en pantalla
16     //colocar cursor en una posicion dada
17     lcd.setCursor(0, 1);
18     //Imprimir un numero entero en formato decimal
19     lcd.printNumber(2010, 10);
20
21 }
```

Ejemplo de código de LCD en pinguino



Periféricos Básicos

Sensor De Temperatura LM35

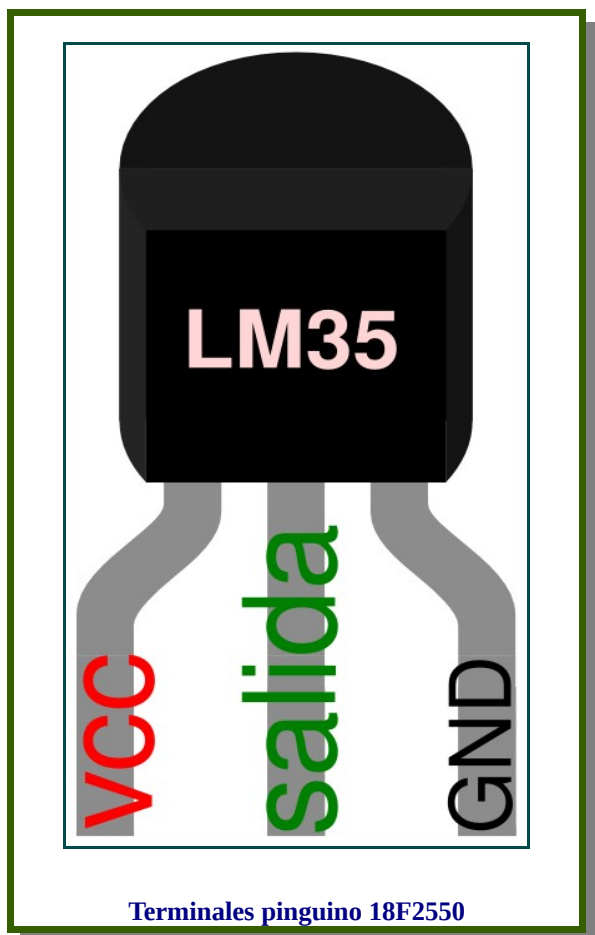
El LM35 es un sencillo sensor de temperatura que puede trabajar en un rango de temperatura de entre -55 a 150 grados centígrados. Este dispositivo viene en un encapsulado plástico TO-92 dándole la misma apariencia de un transistor de uso general.

Sensor de temperatura LM35

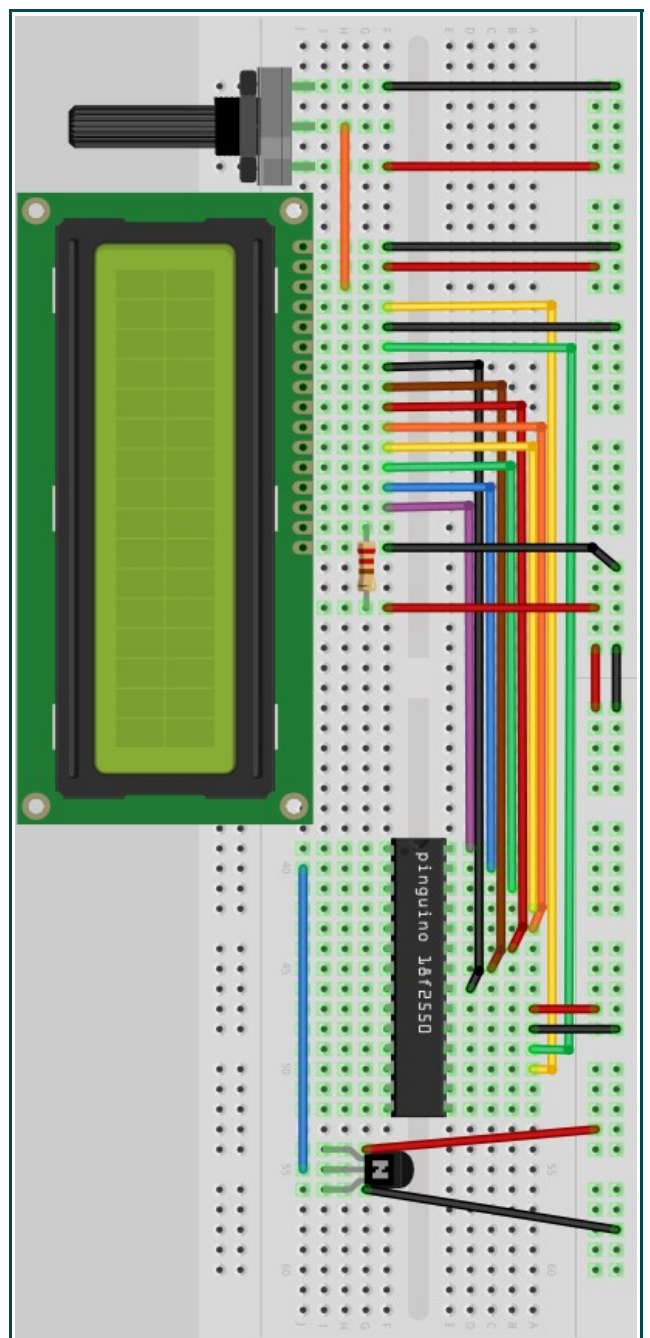
Este sensor cuenta con tres terminales uno de alimentación, uno a tierra y otro de salida análoga. Este dispositivo cuenta en su salida con una variación de $10\text{mV}/^{\circ}\text{C}$, por lo tanto ya sea el microcontrolador que empleamos que tenga un ADC de 8 bits o 10 bits podemos obtener directamente el valor de temperatura sin la necesidad de adecuación de señal.

Este sensor puede trabajar en un rango de temperatura de entre -55°C y 150°C , la fuente de alimentación puede estar entre 4 voltios y 30 voltios.

La disposición de terminales se muestra en el siguiente gráfico.



Guía de montaje



Montaje LM35 con pinguino PIC18F2550

El montaje con LM35 es igual al empleado con el LCD, solo que se conecta la salida de este dispositivo en una de las entradas análogas de la tarjeta pinguino en este caso el terminal indicado como 13, tal como se ilustra en el diagrama de terminales del microcontrolador configurado como pinguino.

En este ejemplo básico solo se lee el dato y se convierte a su equivalente en grados centígrados, ya luego se emplean las instrucciones vistas en el manejo de la pantalla LCD para lograr su visualización.

Programación en Pinguino.

En este caso emplearemos la instrucción de lectura análogo digital en pinguino para capturar el dato de salida del LM35.

Lectura análogo digital

Esta instrucción retorna un valor de entre 0 a 1023 según el valor análogo de entrada. Este ADC es de 10 bits por lo tanto tiene una resolución de 5mv. Los terminales que se pueden utilizar como entradas análogas son 13,14,15,16,17

//Almacena en la variable dato el valor leído en 13

dato=analogRead(13);

```
1  /*****
2  Ejemplo basico de lectura de un dato analogo
3  adc00.pde
4  *****/
5
6  unsigned int dato;
7  unsigned int temperatura;
8  void setup(){
9
10 }
11
12 void loop(){
13     dato=analogRead(13);
14     temperatura=dato/2;
15 }
```

Ejemplo de código de LCD en pinguino