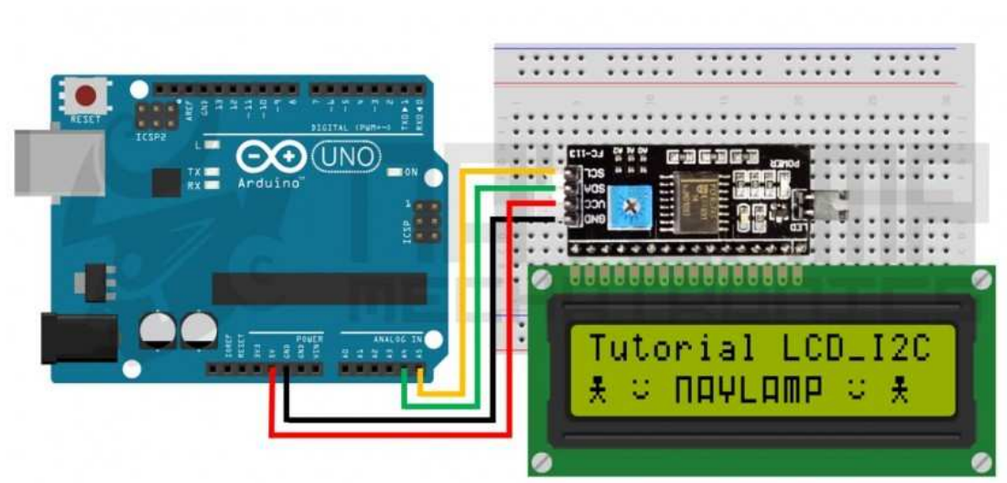
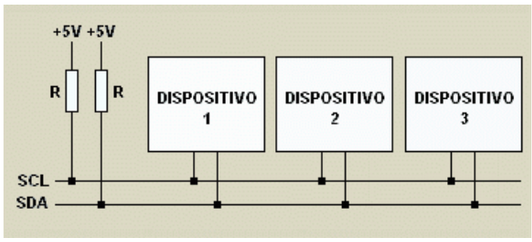
Tutorial LCD con I2C *(versión 19-7-17)*

Controla un LCD con solo dos pines



En este tutorial usaremos un módulo adaptador de LCD a I2C para poder controlar nuestro LCD con solo dos pines de nuestro Arduino.

El bus I2C, un estándar que facilita la comunicación entre microcontroladores, memorias y otros dispositivos con cierto nivel de "inteligencia", sólo requiere de dos líneas de señal y un común o masa. Fue diseñado a este efecto por Philips y permite el intercambio de información entre muchos dispositivos



Al igual que en un tutorial anterior, el objetivo es manejar un LCD 16X2, con la diferencia que aqui usaremos un módulo adaptador LCD a I2C, plantearemos ejercicios similares y veremos que el modulo es fácil de usar como si trabajáramos directamente con el LCD.

EL Módulo adaptador LCD a I2C que usaremos está basado en el controlador I2C PCF8574, el cual es un expansor de entrada y salidas digitales controlado por I2C, que en este módulo se usa para controlar un LCD.

Vamos a suponer que ya conoce los conceptos de I2C, de lo contrario debería ver el material preparado para este tema, que podrá encontrar en este mismo tutor.

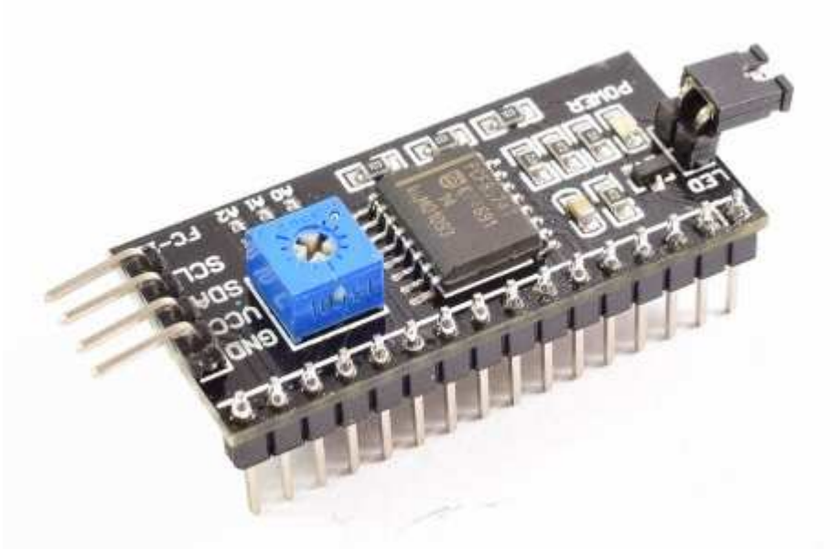
La dirección del módulo I2C que viene por defecto es 0x27, pero podemos cambiarlo soldando los puentes A0,A1y A2; quedando la dirección en binario de la siguiente forma :

0|0|1|0|0|A2|A1|A0.

Por defecto A0,A2,A1 valen 1 pero si soldamos los puentes, estos se conectan a tierra teniendo un valor 0, por ejemplo si soldamos los tres puentes la dirección seria 0|0|1|0|0|0|0|0 (0x20).

**También existen módulos con dirección por defecto 0x3F cuyos bits de configuración son:*

0|0|1|1|1|A2|A1|A0



Aclaración:

Direccionamiento del modulo 8574

Este módulo dispone de dos modelos (PCF8574 y PCF8574A) cuya única diferencia es su dirección.

PCF8574

0	1	0	0	A2	A1	A0

PCF8574A

0	1	1	1	A2	A1	A0

Los primeros cuatro bits vienen configurados de fábrica y los tres últimos son configurables por hardware, (por eso se menciona que se debe soldar para cambiarlo o puentearlos si fuera el caso) , por lo que podemos tener 8 módulos conectados al mismo bus I2C (y si combinamos ambas versiones hasta 16).

Por ello anteriormente cuando dice que el modulo I2C viene por defecto con dirección 0x27, quiere decir que hablamos del modelo PCF8574:

0	1	0	0	A2	A1	A0
0	1	0	0	1	1	1

2	**7**

Los módulos que vienen con la dirección por defecto 0x3F son los PCF8574A:

0	1	1	1	A2	A1	A0
0	1	1	1	1	1	1

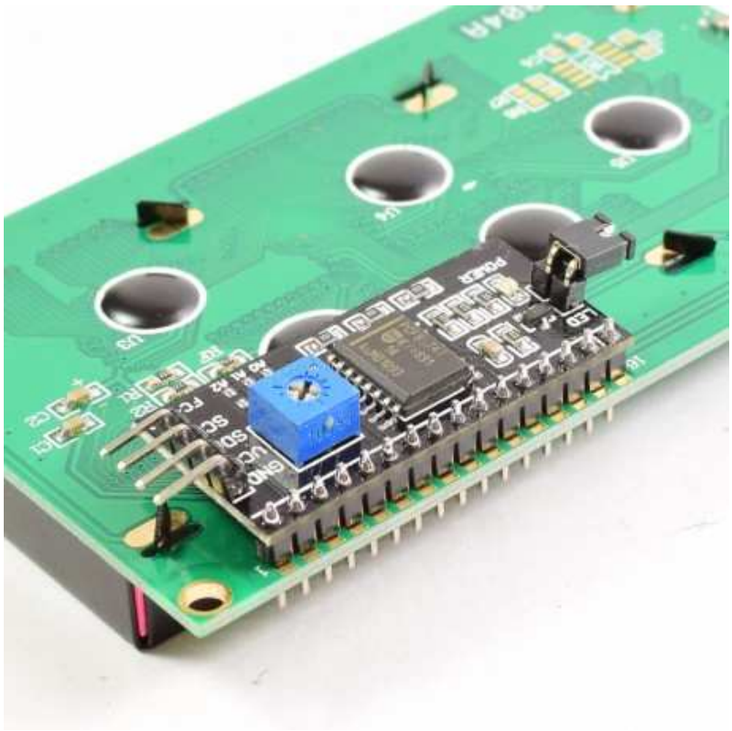
3	**F**

Para controlar el contraste solo necesitamos variar el potenciómetro que se encuentra en el módulo.

La luz de fondo se controla por software, desde el Arduino; pero el modulo tiene un Jumper para desconectar el Led de la luz de fondo.

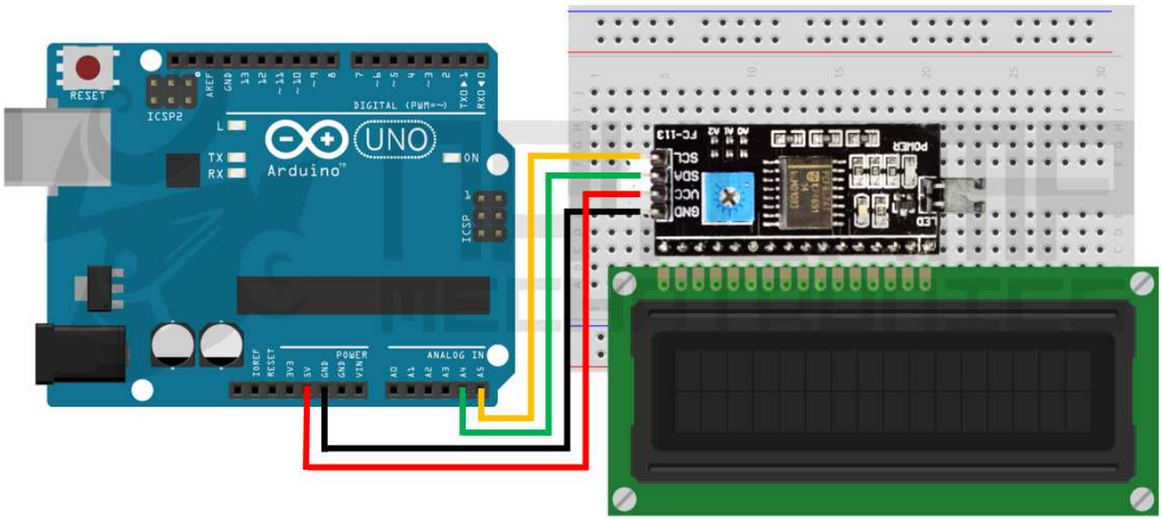
Conexiones entre Arduino y Módulo adaptador LCD a I2C

El adaptador LCD a i2C tiene los pines ordenados para solo conectar al LCD, esto lo podemos hacer a través de un protoboard o soldando directamente al LCD



Para conectar con el Arduino solo utilizamos los pines I2C del Arduino y alimentación

Adaptador LCD a I2C	Arduino Uno, Nano, Mini.	Arduino Mega , DUE	Arduino Leonardo
GND	GND	GND	GND
VCC	5V	5V	5V
SDA	A4	20	2
SCL	A5	21	3



***** Las conexiones para el LCD de 20 x 4 son las mismas.

Librería LiquidCrystal_I2C para Arduino

Existen diferentes tipos y versiones de la librería para LCD I2C, información más completa pueden encontrar en: http://playground.arduino.cc/Code/LCDi2c , nosotros usaremos la librería **LiquidCrystal_I2C**

Las funciones que utiliza esta librería son similares a la librería **LiquidCrystal** de Arduino, explicaremos las funciones principales.

LiquidCrystal_I2C(lcd_Addr, lcd_cols, lcd_rows)

Función constructor, crea una variable de la clase LiquidCrystal_I2C, con dirección, columnas y filas indicadas.

init()

Inicializa el modulo adaptador LCD a I2C, esta función internamente configura e inicializa el I2C y el LCD.

clear()

Borra la pantalla LCD y posiciona el cursor en la esquina superior izquierda (posición (0,0)).

setCursor(col, row)

Posiciona el cursor del LCD en la posición indicada por col y row(x,y); es decir, establecer la ubicación en la que se mostrará posteriormente texto escrito para la pantalla LCD.

print()

Escribe un texto o mensaje en el LCD, su uso es similar a un Serial.print

scrollDisplayLeft()

Se desplaza el contenido de la pantalla (texto y el cursor) un espacio hacia la izquierda.

scrollDisplayRight()

Se desplaza el contenido de la pantalla (texto y el cursor) un espacio a la derecha.

backlight();

Enciende la Luz del Fondo del LCD

noBacklight();

Apaga la Luz del Fondo del LCD

createChar (num, datos)

Crea un carácter personalizado para su uso en la pantalla LCD. Se admiten hasta ocho caracteres de 5x8 píxeles (numeradas del 0 al 7). Dónde: **num** es el número de carácter y**datos** es una matriz que contienen los pixeles del carácter. Se verá un ejemplo de esto más adelante.

Explicado la librería veamos unos ejemplos:

---

1. Un Hola mundo con Arduino y LCD

Mostraremos texto y un valor numérico en el LCD, para esto cargamos el siguiente sketch:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//Es probable que deba usar una variante de esta libreria
//Se suministra el archivo LiquidCrystal_I2CDJB.ZIP
//para tal variante- lea como instalar librerias

//Crear el objeto lcd  dirección  0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2);   //
//Es probable que deba cambiar la direccion 0x27 por 0x3F
//Dependera de su LCD- Si tiene el PCF8574 o el  PCF8574A
void setup() {
  // Inicializar el LCD
  lcd.init();

  //Encender la luz de fondo.
  lcd.backlight();

  // Escribimos el Mensaje en el LCD.
  lcd.print("Hola Mundo");
}

void loop() {
    // Ubicamos el cursor en la primera posición(columna:0) de la segunda línea(fila:1)
  lcd.setCursor(0, 1);
    // Escribimos el número de segundos trascurridos
  lcd.print(millis()/1000);
  lcd.print(" Segundos");
  delay(100);
}
```

Después de cargar, en su LCD deben obtener el siguiente resultado

La programación es similar para un LCD de 20x4, solo hay que modificar en:

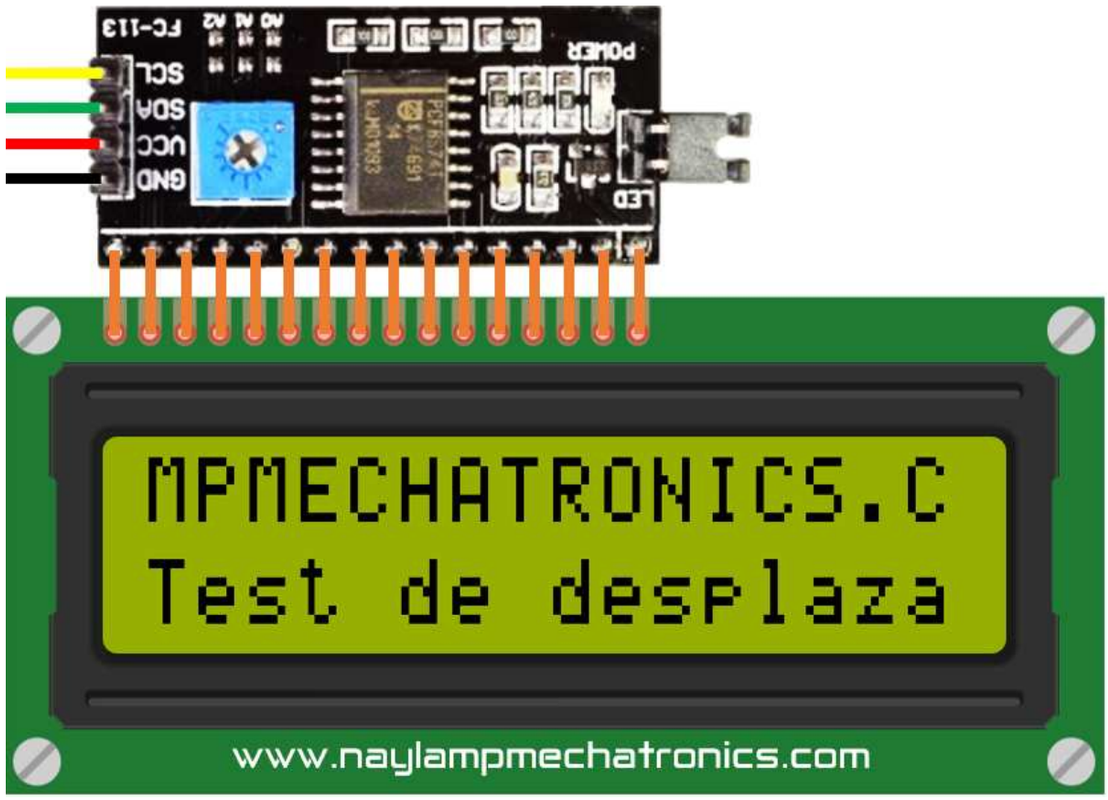
LiquidCrystal_I2C lcd(0x27,20,4);

2. Desplazando el texto en el LCD

En este ejemplo para desplazar el texto usaremos la función scrollDisplayLeft(). El codigo es el siguiente:

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//Crear el objeto lcd  dirección  0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2);   //

void setup() {
  // Inicializar el LCD
  lcd.init();

  //Encender la luz de fondo.
  lcd.backlight();

  // Escribimos el Mensaje en el LCD en una posición  central.
  lcd.setCursor(10, 0);
  lcd.print("WWW.NAYLAMPMECHATRONICS.COM");
  lcd.setCursor(4, 1);
  lcd.print("Tutorial LCD, Test de desplazamiento  ");
}

void loop() {
  //desplazamos una posición a la izquierda
  lcd.scrollDisplayLeft();
  delay(500);
}
```

EL resultado sera el texto desplazándose de derecha a izquierda.



Otro ejemplo de dezplamiento de texto:

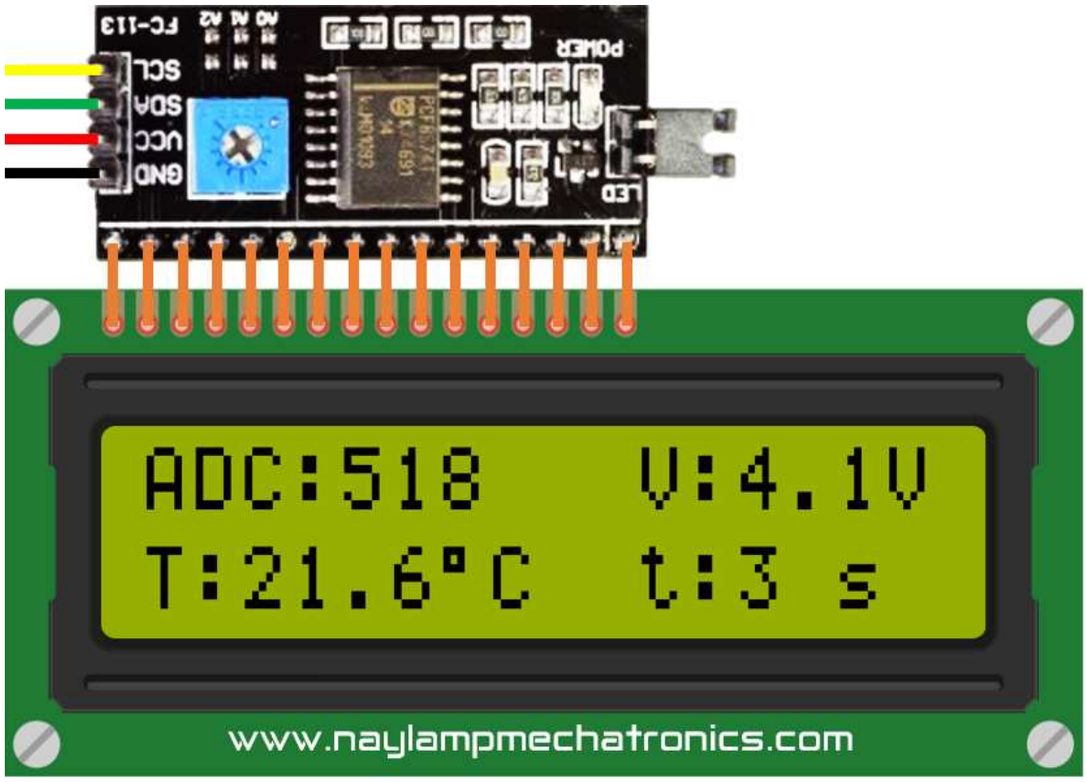
```
 // Prueba de Modulo I2C para LCD 2x16 By: http://dinastiatecnologica.com

#include <LiquidCrystal_I2C.h> // Debe descargar la Libreria que controla el
I2C
#include<Wire.h>

LiquidCrystal_I2C lcd(0x3F,16,2);

//Es probable que deba cambiar la direccion 0x27 por 0x3F
//Dependera de su LCD- Si tiene el PCF8574 o el  PCF8574A


void setup() {
  lcd.init();
  lcd.backlight();
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("Dinastia Tecnologica...."); // Mensaje a despegar
  delay(3000);
}

void loop() {

  for(int c=0;c<12;c++){
  lcd.scrollDisplayLeft();
  delay(400);
  }
  for(int c=0; c<12;c++){
  lcd.scrollDisplayRight();
  delay(400);
  }
}
```

3. Mostrando datos de sensores o variables en el LCD

En este ejemplo mostramos en el LCD variables, que pueden representar valores de sensores u otros datos. Para simular los sensores usaremos **potenciómetros conectados a los puertos analógicos**.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//Crear el objeto lcd  dirección  0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2);   //

void setup() {
  // Inicializar el LCD
  lcd.init();

  //Encender la luz de fondo.
  lcd.backlight();

}

void loop() {
  int sen1=analogRead(A0);
  float sen2=analogRead(A1)*(5.0 / 1023.0);
  float sen3=analogRead(A2)*(100.0 / 1023.0);
  int tiempo=millis()/1000;
  // Cursor en la primera posición de la primera fila
  lcd.setCursor(0,0);
  lcd.print("ADC:");
  lcd.print(sen1);
  lcd.print("  ");
  // Cursor en la 11° posición de la primera fila
  lcd.setCursor(10,0);
  lcd.print("V:");
  lcd.print(sen2,1);//1 decimal
  lcd.print("V  ");
  // Cursor en la primera posición de la 2° fila
  lcd.setCursor(0,1);
  lcd.print("T:");
  lcd.print(sen3,1);  //1 decimal
  lcd.print("337C  ");  // "337" -> "°"
  // Cursor en la 11° posición de la 2° fila
  lcd.setCursor(10,1);
  lcd.print("t:");
  lcd.print(tiempo);
  lcd.print(" s  ");

  delay(200);

}
```

Su resultado debe ser el siguiente.



4. Agregando nuevos caracteres a nuestro LCD

Explicaremos nuevamente como esta característica en caso no hayan revisado el tutorial anterior.

En algunos casos el LCD no tiene los caracteres que deseamos, o necesitamos dibujar caracteres personalizados, en este caso usamos la función **createChar ()** pero antes expliquemos como está constituido un carácter:

Un carácter está formado por 5x8 pixeles los cuales se representan por 8 bytes, uno para cada fila, los 5 bits menos significativos representan los pixeles del carácter

	d7	d6	d5	d4	d3	d2	d1	d0
byte0	-	-	-	1	1	1	1	0
byte1	-	-	-	1	0	0	0	1
byte2	-	-	-	1	0	0	0	1
byte3	-	-	-	1	1	1	1	0
byte4	-	-	-	1	0	1	0	0
byte5	-	-	-	1	0	0	1	0
byte6	-	-	-	1	0	0	0	1
byte7	-	-	-	0	0	0	0	0

Como máximo podemos crear 8 caracteres nuevos.

Para este ejemplo crearemos los siguientes caracteres:

N naylamp

1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
0	0	0	0	0

A naylamp

1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
1	0	0	0	1
0	0	0	0	0

Y naylamp

1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0
0	0	0	0	0

L naylamp

1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	0	0	0	0
1	1	1	1	1
0	0	0	0	0

M naylamp

1	1	1	1	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
1	0	1	0	1
0	0	0	0	0

P naylamp

1	1	1	1	1
1	0	0	0	1
1	0	0	0	1
1	0	0	0	1
1	1	1	1	1
1	0	0	0	0
1	0	0	0	0
0	0	0	0	0

cara

0	0	0	0	0
1	0	0	0	1
0	0	0	0	0
0	0	0	0	0
1	0	0	0	1
0	1	1	1	0
0	0	0	0	0
	0	0	0	0

cuerpo

0	1	1	1	0
0	1	1	1	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	1	0	1	0
1	0	0	1	1
0	0	0	0	0

A continuación se muestra el código para implementar los nuevos caracteres.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

//Crear el objeto lcd  dirección  0x27 y 16 columnas x 2 filas
LiquidCrystal_I2C lcd(0x27,16,2);   //
byte N[8] = {
B11111,
B10001,
B10001,
B10001,
B10001,
B10001,
B10001,
B00000,
};
byte A[8] = {
B11111,
B10001,
B10001,
B10001,
B10001,
B11111,
B10001,
B00000,
};
byte Y[8] = {
B10001,
B10001,
B10001,
B10001,
B11111,
B00100,
B00100,
B00000,
};
byte L[8] = {
B10000,
B10000,
B10000,
B10000,
B10000,
B10000,
B11111,
B00000,
};
byte M[8] = {
B11111,
B10101,
B10101,
B10101,
B10101,
B10101,
B10101,
B00000,
};
byte P[8] = {
B11111,
B10001,
B10001,
B10001,
B11111,
B10000,
B10000,
B00000,
```

```
};
byte cara[8] = {
B00000,
B10001,
B00000,
B00000,
B10001,
B01110,
B00000,
};
byte cuerpo[8] = {
B01110,
B01110,
B00100,
B11111,
B00100,
B01010,
B10001,
B00000,
};

void setup() {
   // Inicializar el LCD
   lcd.init();

   //Encender la luz de fondo.
   lcd.backlight();

   //creamos los nuevos caracteres
   lcd.createChar (0,N);
   lcd.createChar (1,A);
   lcd.createChar (2,Y);
   lcd.createChar (3,L);
   lcd.createChar (4,M);
   lcd.createChar (5,P);
   lcd.createChar (6,cara);
   lcd.createChar (7,cuerpo);
  // Escribimos el texto en el LCD
   lcd.setCursor(0, 0);
   lcd.print("Tutorial LCD_I2C");
   lcd.setCursor(0, 1);
   lcd.write (byte (7));
   lcd.print(" ");
   lcd.write (byte (6));
   lcd.print(" ");
   lcd.write (byte (0));
   lcd.write (byte (1));
   lcd.write (byte (2));
   lcd.write (byte (3));
   lcd.write (byte (1));
   lcd.write (byte (4));
   lcd.write (byte (5));
   lcd.print(" ");
   lcd.write (byte (6));
   lcd.print(" ");
   lcd.write (byte (7));



}

void loop () {}
```

El resultado del ejemplo anterior es el siguiente:

FUENTE:	**http://www.naylampmechatronics.com**