Control de Pantallas de Lcd y Oled Utilizando Arduino

Ing. Esther Viridiana Vázquez Carmona
ev.vazquezc@gmail.com

Ing. Rodrigo Vázquez López
rodrigo_em2@hotmail.com

Ing. Nilda Fabiola Encarnación Avalos
fabi-nil_34@hotmail.com

Dr. Juan Carlos Herrera Lozada
jlozada@ipn.mx
Instituto Politécnico Nacional
Centro de Innovación y Desarrollo Tecnológico en Cómputo

Resumen

En los sistemas embebidos, es importante el uso de dispositivos para desplegar texto y gráficos que faciliten la interacción entre el usuario y el sistema. Algunos de los dispositivos más utilizados en la actualidad, por su simplicidad y fácil integración en dichos sistemas, son las pantallas de LCD y las pantallas OLED. El objetivo de este trabajo es realizar una serie de diseños para mostrar el funcionamiento de dichos dispositivos, utilizando las bibliotecas LiquidCrystal y Adafruit SSD1306 que proporciona la plataforma Arduino. Los circuitos realizados consistieron en la implementación de un contador que trabaja con una pantalla OLED de 128x64, utilizando el protocolo SPI para la comunicación con la plataforma Arduino. Adicionalmente se llevó a cabo una rutina para desplegar cadenas de texto, cuyo tamaño excede el número de columnas de una LCD de 16x2 de tal forma que los caracteres se vayan desplazando hacia la izquierda creando un efecto de marquesina.

Introducción

Una pantalla LCD (Liquid Crystal Dysplay) es un dispositivo que se utiliza para la visualización de información (caracteres y símbolos) de una forma gráfica. Los displays LCD más comunes son de 16x2, es decir, está compuesto de 2 filas de 16 caracteres cada una. Los píxeles de cada símbolo o carácter varían en función de cada modelo [1]. Existen modelos de última generación que son de gran tamaño, a todo color y cuentan con luz de fondo.

Un display OLED a diferencia de una pantalla LCD, está construido con un tipo de LED en el que la capa emisiva está formada por un compuesto orgánico que

emite luz en respuesta a la electricidad [2]. Las pantallas OLED más comunes son de 0.96", tienen un tamaño reducido y una resolución de 128x64 pixeles. Estas pantallas incorporan el controlador SDD1306 y son monocromas [3]. La ventaja de este tipo de pantallas es su bajo consumo eléctrico (alrededor de 20mA) y mejor visibilidad en ambientes luminosos por lo que no requieren de back light.

Para mostrar información en los displays o pantallas, es necesario conectarlos a un microcontrolador que envíe las señales de control específicas. En el caso de la pantalla OLED la comunicación puede realizarse utilizando el bus SPI o I2C (dependiendo el modelo), mientras que la comunicación entre la pantalla LCD hacía el microcontrolador se hace a través de un bus de 8 bits o 4 bits [4], así como las señales de control RS (chip select) RW (lectura/escritura) y E (enable).

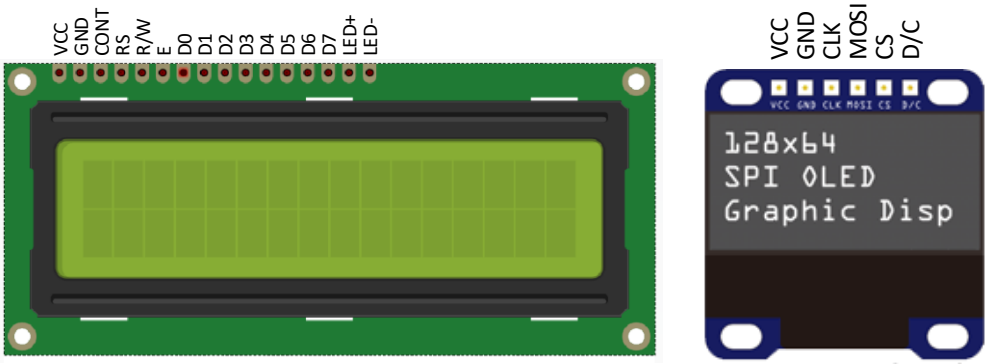

Figura 1. LCD y OLED.

La plataforma Arduino tiene a su disposición bibliotecas que facilitan el uso de estos dispositivos, una de ellas es LiquidCrystal, la cual incluye todas las operaciones necesarias para la manipulación del display LCD. Algunas de las funciones de la librería se muestran en la tabla 1.

Tabla 1. Algunas funciones de la librería LiquidCrystal

Función	Descripción
LiquidCrystal ()	Crea una variable
begin ()	Inicializa la interfaz a la pantalla LCD
clear ()	Limpia la pantalla de la LCD
setCursor ()	Posiciona el cursor en la LCD
print ()	Imprime el texto en la LCD

En el caso de la pantalla OLED, Adafruit tiene a su disposición la biblioteca SSD1306 que contiene la lógica necesaria para el manejo de dicha pantalla, así como la biblioteca GFX, la cual contiene las funciones necesarias para desplegar texto o primitivas graficas como son puntos, líneas, rectángulos, círculos, etc. [5]. Adicionalmente requiere de las bibliotecas Wire y SPI para la comunicación por medio del bus I2C o SPI respectivamente.

El protocolo SPI es un estándar de comunicaciones, para la transferencia de información entre dispositivos digitales, como pantallas LCD, sensores, microcontroladores, entre otros [6], es muy utilizado, ya que cuenta con una buena velocidad de transmisión, para cuestiones de implementación resulta ser muy sencillo.

Despliegue de un contador en una pantalla de LCD

El primer diseño consistió en un contador ascendente del 0-99 que muestra en pantalla dicha cuenta, además incluye la cadena “IPN-CIDETEC”, la cual se ubica en el primer renglón del display LCD. El sistema cuenta con un botón para mantener el valor de la cuenta, así como un botón que restablece el conteo.
El diagrama de flujo de la figura 2 muestra el funcionamiento del programa, el cual lee el estado de los botones reset y hold. En caso de que se haya presionado el botón reset el valor de la cuenta se regresa a 0, en caso contrario se procede a preguntar por el estado del botón hold. Si se presionó el botón hold, el programa mantiene el valor de la cuenta, en caso de que no se haya presionado el botón del hold, el contador se incrementa una unidad, siempre y cuando el valor de la cuenta no exceda el número 99.

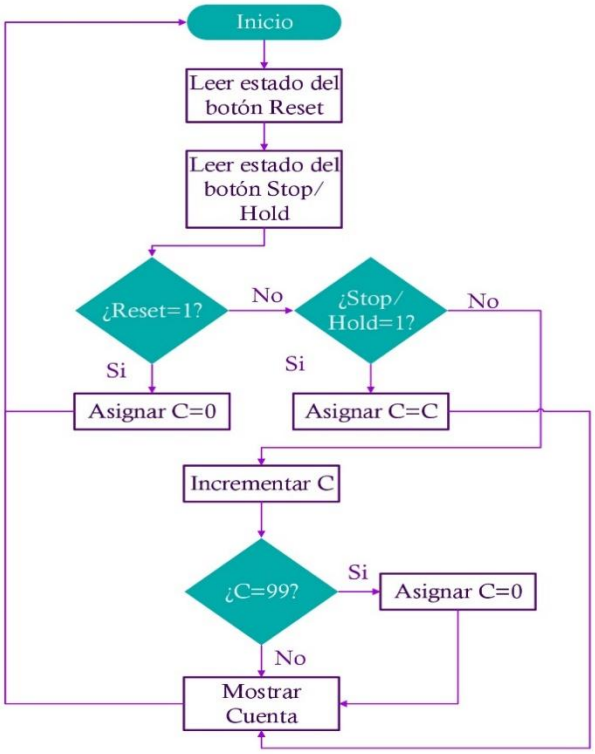

Figura 2. Contador LCD.

Finalmente, el valor de la cuenta se muestra en el segundo renglón de la LCD. El diagrama de conexión del circuito se muestra en la figura 3. El resultado final se puede observar en la figura 7.

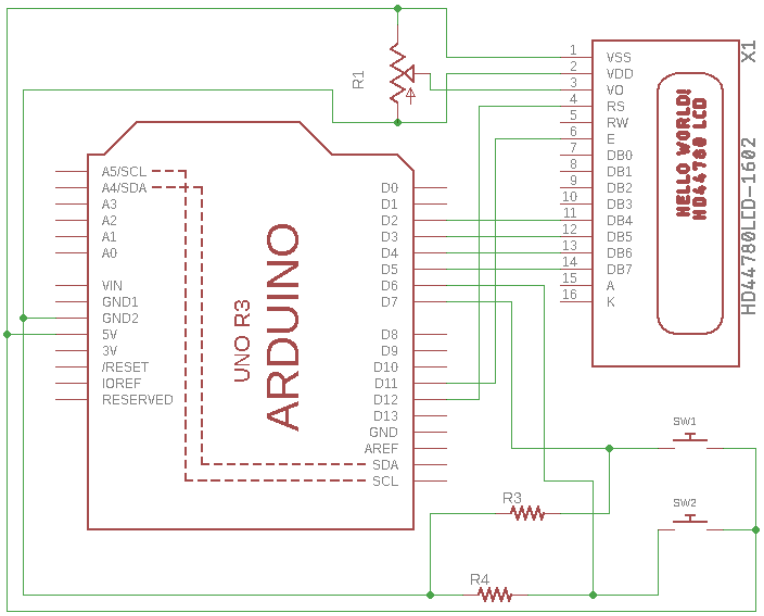

Figura 3. Diagrama de conexiones del contador LCD.

Despliegue de un contador en una pantalla OLED

En este diseño se reemplazó la pantalla LCD del circuito anterior por una pantalla OLED de 128x64. La comunicación entre la pantalla OLED y Arduino se realizó utilizando el protocolo de comunicación SPI. Nótese que el algoritmo es exactamente el mismo a excepción del protocolo de comunicación. El diagrama de conexiones se muestra en la figura 4. El resultado final se observa en la figura 7.

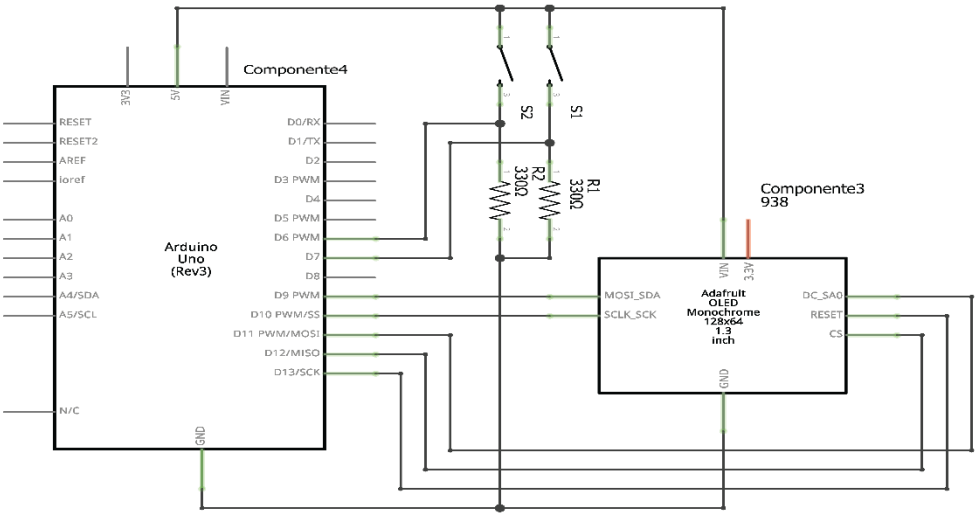

Figura 4. Diagrama de conexiones del contador en OLED.

Marquesina

En algunos casos se requiere mostrar cadenas de texto cuyo tamaño rebasa el número de columnas de la pantalla LCD, por lo que es necesario realizar un efecto de" marquesina "que desplace el texto para que se despliegue por completo. La figura 5 muestra el diagrama de flujo que representa el algoritmo propuesto, la cadena principal se segmenta en su cadena del tamaño de columnas de la LCD y estas de despliegan en pantalla cada 200ms, representando así el efecto de una marquesina. El diagrama de conexiones se observa en la figura 6. La implementación del circuito se puede observar en la figura 7.

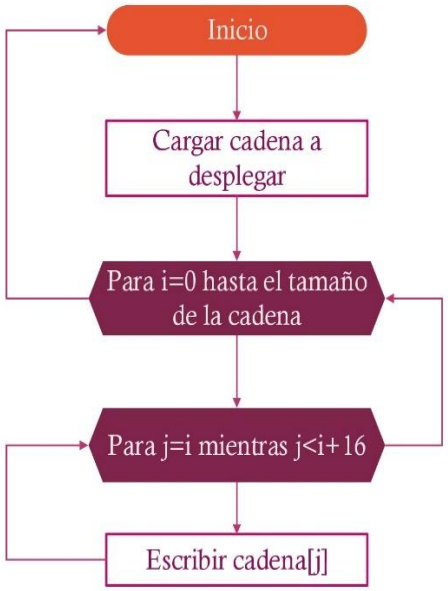

Figura 5. Diagrama de flujo de Marquesina.

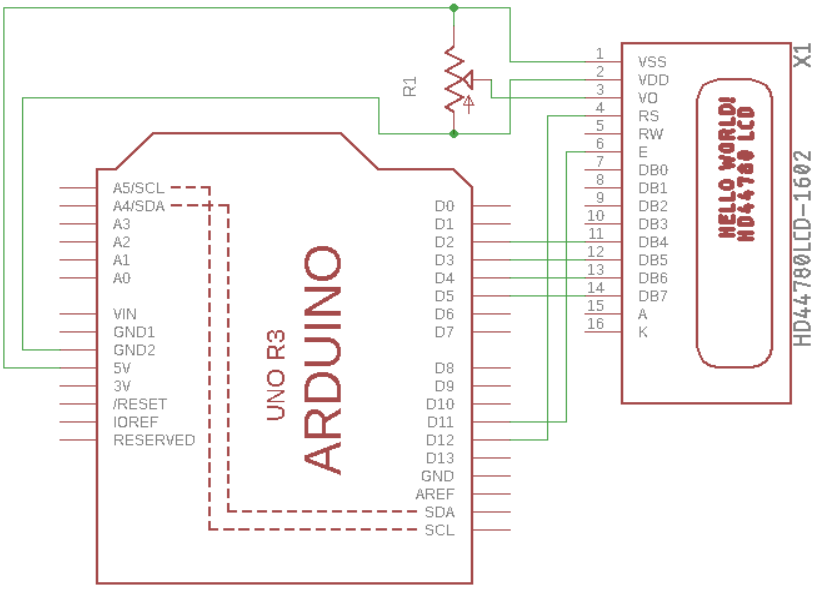

Figura 6. Diagrama de conexiones de Marquesina.

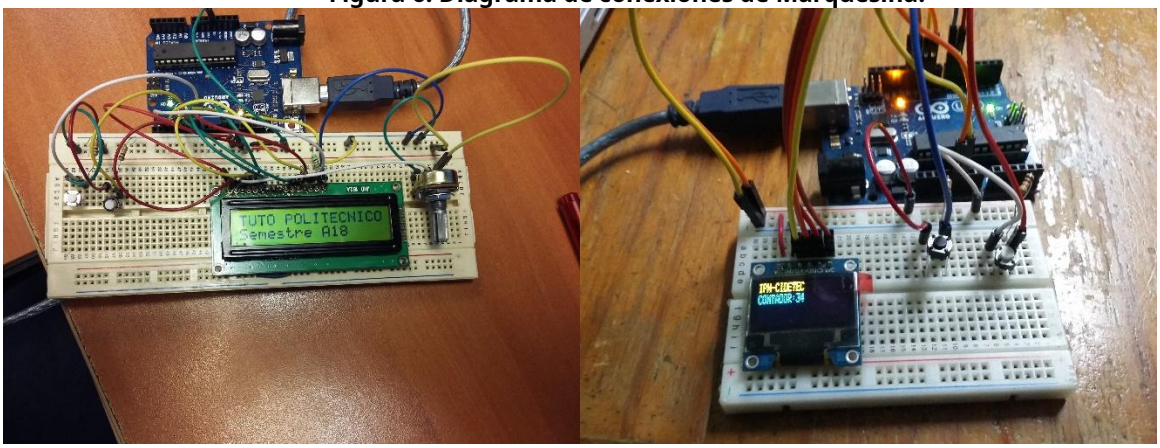

Figura 7. Implementación de los diseños en LCD y OLED.

Conclusiones

La programación de interfaces de despliegue en la plataforma Arduino es fácil de implementar gracias a las librerías que encapsulan la comunicación a bajo nivel.

A diferencia del manejo del display LCD con un microcontrolador clásico, el uso de la librería LiquidCrystal en Arduino simplifica el trabajo de comunicación, por lo que inicializar una pantalla LCD y mostrar texto se puede hacer con pocas instrucciones.

En cuanto al manejo de las pantallas OLED, este se simplifica gracias a las librerías de Adafruit, que encapsulan la comunicación del protocolo SPI o I2C y permiten dibujar primitivas graficas de forma rápida y sencilla, presentando una mejor calidad de despliegue debido a la resolución en este tipo de dispositivos.

Referencias

[1] J. Illiet," How to Use Intelligent LCDs-1.Everyday with Practical Electronics, Num 26.2, 1997),pp 84-89.
[2] OLED-info. An introduction to OLED displays. Accesed april 2018. [Online]. Available: https://www.oled-info.com/introduction
[3] Adafruit. Monochrome 1.3"128x64 OLED graphic display. Accesed april 2018. [Online]. Available: https://www.adafruit.com/product/938
[4] Naylamp Mechatronics SAC. Tutorial LCD, conectando tu arduino a un LCD1602 y LCD2004. Accesed August 2018. [Online]. Available: https://www.naylampmechatronics.com/blog/34_Tutorial-LCD-conectando-tu-arduino-a-un-LCD1.html
[5] P. Burgess. Adafruit GFX Graphics Library. Accesed april 2018. [Online]. Available: https://learn.adafruit.com/adafruit-gfx-graphics-library?view=all
[6] K. Navarro. Como funciona el protocolo SPI. Accesed april 2018. [Online]. Available: http://panamahitek.com/como-funciona-el-protocolospi/
[7] P. Marwedel, Embedded system design, 3rd ed. Berlin, Germany: Springer, 2008.
[8] TodoElectrodo. Lcd 16x2. Accesed april 2018. [Online]. Available: http://todoelectrodo.blogspot.mx/2013/02/lcd-16x2.html

Anexos

Código Contador LCD

```
//Biblioteca para pantalla LCD
#include <LiquidCrystal.h>
```

```
//Declaración e inicialización de los pines conectados a la pantalla LCD
int C = 0, stateRst = 0, stateStop = 0;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
const int btnRst = 6, btnStop = 7;
LiquidCrystal lcd (rs, en, d4, d5, d6, d7);


void setup () {
//Inicializa los puertos de botón reset y botón stop
  pinMode (btnRst, INPUT);
  pinMode (btnStop, INPUT);
//Inicializa la posición en donde se va a comenzar a escribir en la LCD
  lcd. begin (16, 2);
  lcd.print("IPN-CIDETEC");
}
//Función principal

void loop () {
//Lectura de los estados de botón reset y botón stop

  stateRst = digitalRead(btnRst);
  stateStop = digitalRead(btnStop);

   if (stateRst == HIGH) {
    C=0;
    lcd. clear ();
    lcd.print("IPN-CIDETEC");
  }
  else {
    if(stateStop==HIGH) {
      C=C;
    }
    else {
      C=C+1;
```

```
      if (C == 99) {
        C=0;
        lcd.clear();
        lcd.print("IPN-CIDETEC");
      }
    }
  }
  lcd.setCursor(0, 1);
  lcd.print("CONTADOR:");
  lcd.print(C);
  delay(200);
}
```

Código Marquesina LCD

```
//Biblioteca para pantalla LCD
#include <LiquidCrystal.h>

//La cadena que se desplegará en la pantalla LCD se inicializa con el texto
char cad1[]="INSTITUTO POLITECNICO
NACIONAL-CENTRO DE INNOVACION Y
DESARROLLO TECNOLOGICO EN COMPUTO ";
const int rs = 12, en = 11, d4 = 5, d5 =4, d6 = 3, d7 = 2;
LiquidCrystal lcd (rs, en, d4, d5, d6, d7);

void setup () {
//Inicializa la posición en donde se va a comenzar a escribir en la LCD
lcd. begin (16, 2);
}

//Función principal
void loop () {
for (int i=0; i<90;i++) {
    lcd. setCursor (0,0);
```

```
    for (int j=i;j<i+16;j++) {
        lcd. write(cad1[j]);
 }

lcd. setCursor (0,1);
lcd.print ("Semestre A18");
delay (200);
 }
}
```

Código Contador OLED

```
//Bibliotecas para pantalla OLED
#include <SPI.h>
#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
//Se definen los pines de la pantalla OLED
#define OLED_MOSI   9
#define OLED_CLK   10
#define OLED_DC    11
#define OLED_CS    12
#define OLED_RESET 13
Adafruit_SSD1306 display (OLED_MOSI, OLED_CLK, OLED_DC, OLED_RESET,
OLED_CS);
#define SSD1306_LCDHEIGHT 64
#if (SSD1306_LCDHEIGHT! = 64)
#error ("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif
//Declaración e inicialización de los puertos que corresponden al estado reset
y estado stop
int C = 0, stateRst = 0, stateStop = 0;
const int btnRst = 6, btnStop = 7;
```

```
void setup () {
//Configuraciones adicionales para la pantalla OLED
  Serial.begin(9600);
  display. begin(SSD1306_SWITCHCAPVCC);
  display.display();
  delay (2000);
  display. clearDisplay ();
  display. setTextSize (1);
  display.setTextColor(WHITE);

  pinMode (btnRst, INPUT);
  pinMode (btnStop, INPUT);
}
//Función principal
void loop () {
  stateRst = digitalRead(btnRst);
  stateStop = digitalRead(btnStop);
  if (stateRst == HIGH) {
    C=0;
  }
  else {
    if(stateStop==HIGH) {
      C=C;
    }
    else {
      C=C+1;
      if (C == 100) {
        C=0;
      }
    }
```

```
  }
  display. clearDisplay ();
  display. setCursor (0,0);
  display. println("IPN-CIDETEC");
  display.print("CONTADOR:");
  display.print(C);
  display. display ();
  delay (200);
}
```