

MICROCONTROLADORES

Práctica No. 11. Control de una Pantalla LCD Alfanumérica.

1. Objetivo

- Display LCD alfanumérico 2x16.
- Modulo PCF8574

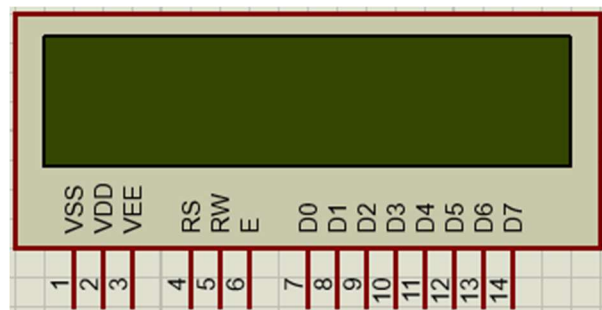
2. Material y Equipo.

- Computador o laptop con el STM32CubeIDE.
- Un Display LCD alfanumérico.
- Un módulo PC8574.

3. Marco de Referencia.

Las pantallas LCD son muy utilizadas existen diferentes tipos de pantallas pueden ser alfanuméricas y graficas incluso unas con la capacidad de tener un panel touch. La pantalla más sencilla de utilizar son las alfanuméricas, su inicialización y su manipulación es bastante sencilla.

El modulo LCD usado en esta practica es una pantalla estándar LM016L, se puede configurar en modo de 8 bits o 4 bits, la siguiente imagen muestra el pinout de la pantalla.



La descripción de sus pines es la siguiente.

SEÑAL	DESCRIPCION	PINES	FUNCIONAMIENTO
V _{SS}	Tierra	1	Tierra (0 V)
V _{DD}	Voltaje de Alimentacion	2	Voltaje de Alimentacion (+5 V).
V _{LC}	Voltaje de Contraste	3	Voltaje de Contraste Ajustable.
RS	Register Select	4	RS = 0, Modo Comando. RS = 1, Modo Carácter.
R/W	Read / Write	5	R/W = 0, Escribir en LCD. R/W = 1, Leer del LCD.
E	Enable	6	E = 0, LCD Deshabilitado. E = 1, LCD Habilitado.
DB0 - DB7	Data Bus	7 - 12	Bus de Datos.

La pantalla se alimenta con +5V a través del modulo del PCF8574, este voltaje debe ser proporcionado a través del programador ST-Link o conectando la tarjeta a través del puerto micro usb. El voltaje de contraste se ajusta a través de un potenciómetro de 10KΩ.

La pantalla tiene una memoria llamada DDRAM (Data Display RAM) en donde se almacenan los códigos ASCII de los caracteres que aparecen en pantalla. Son 80 bytes 40 por cada línea, de los cuales 32 se pueden visualizar a la vez (16 bytes por línea).

Las direcciones mas importantes de esta memoria es la 0x00 que corresponde al comienzo de la primera línea y la dirección 0x40 que corresponde al comienzo de la segunda línea.

La siguiente memoria es la CGROM es una sección de memoria de 192 caracteres que se pueden visualizar en la pantalla, cada carácter tiene una representación de 8 bits. El dato que se quiere visualizar es recibido por el bus de datos. Por ejemplo, el carácter "A" el código que se debe recibir por el bus de datos es el 0b01000001. La siguiente tabla muestra la tabla guardada en la CGROM.

Lower 4 Bits	Upper 4 Bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	CG RAM (1)				0	0	P	`	P				-	9	3	α	p
xxxx0001	(2)				!	1	A	Q	a	9			▯	7	4	ä	q
xxxx0010	(3)				"	2	B	R	b	r			"	イ	ツ	×	ρ
xxxx0011	(4)				#	3	C	S	c	s			」	ウ	テ	ε	∞
xxxx0100	(5)				\$	4	D	T	d	t			\	イ	ト	μ	Ω
xxxx0101	(6)				%	5	E	U	e	u			•	オ	ナ	1	ü
xxxx0110	(7)				&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ
xxxx0111	(8)				'	7	G	W	g	w			フ	キ	ヌ	ラ	π
xxxx1000	(1)				(8	H	X	h	x			イ	ク	ネ	リ	⌘
xxxx1001	(2))	9	I	Y	i	y			ウ	ク	ル	リ	γ
xxxx1010	(3)				*	:	J	Z	j	z			エ	コ	ハ	レ	j
xxxx1011	(4)				+	;	K	L	k	l			オ	サ	ヒ	ロ	⌘
xxxx1100	(5)				,	<	L	¥	l	l			ハ	シ	フ	ワ	⌘
xxxx1101	(6)				-	=	M	J	m	}			ユ	ズ	ハ	ン	÷
xxxx1110	(7)				.	>	N	^	n	→			ヨ	セ	ホ	ン	ñ
xxxx1111	(8)				/	?	O	_	o	+			ッ	ソ	マ	ン	ö

También se pueden definir ocho nuevos caracteres, que no estén incluidos en la tabla anterior, estos caracteres se guardan en una zona de la RAM llamada CGRAM (Character Generator RAM).

Por último, tenemos los modos de trabajo de la pantalla LCD.

MODOS DE FUNCIONAMIENTO	
MODO COMANDO	El display recibe por el bus de datos un dato que se interpretará como un comando. Para interpretar este dato como un comando el pin RS y R/W deben estar a nivel bajo (0v). Una operación en este modo tarda un máximo de 1.64 ms.
MODO CARÁCTER	El display recibe por el bus de datos un dato que se interpretará como un carácter que se desplegará en el LCD. Para interpretar este dato como un carácter el pin RS debe estar en alto (+5V) y R/W debe estar a nivel bajo (0v). Una operación en este modo tarda un máximo de 40 us.
MODO LECTURA	En este modo el bit 7 del bus de datos del LCD se usa para señalar que el display está ocupado. Para leer este bit el pin RS debe estar a cero mientras R/W debe estar en uno. Si el bit 7 del bus está en 1 eso indica que la pantalla está ocupada realizando alguna operación interna y no aceptará nuevos comandos o datos. Hay que esperar a que el bit 7 sea cero para poder enviarle un nuevo comando o dato a la pantalla. Este modo no se usará en las prácticas y se sustituirá por un retardo entre comando o dato que se envíe a la pantalla.

Los comandos de control se definen en el archivo "lcd.h". Las definiciones son las siguientes.

```

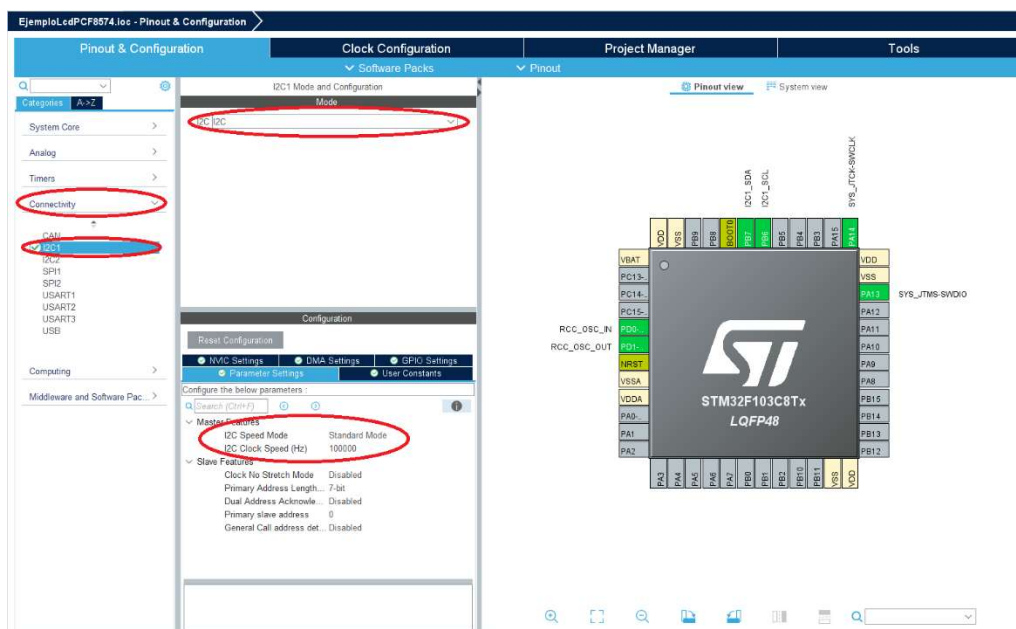
10 // comando para configurar el LCD
11 #define LIMPIARLCD          0x01
12 #define INICIOLINEA        0x02
13 #define RECORRECURRIZQ     0x04
14 #define RECORRECURRDER     0x06
15 #define RECORRERPANTDER    0x05
16 #define RECORRERPANTIZQ    0x07
17 #define DISPOFFCUROFF     0x08
18 #define DISPOFFCURON      0x0A
19 #define DISPONCUROFF      0x0C
20 #define DISPONCURBLNK1     0x0E
21 #define DISPONCURBLNK2     0x0F
22 #define DESPCURIZQ        0x10
23 #define DESPCURDER        0x14
24 #define DESPDISPIZQ       0x18
25 #define DESPDISPDER       0x1C
26 #define FORZARCURINCLINEA1 0x80
27 #define FORZARCURINCLINEA2 0xC0
28 #define DOSLINEAS5X74B    0x28
29 #define DOSLINEAS5X78B    0x38

```

4. Desarrollo y Procedimiento.

Se creará un proyecto en el STM32CubeIDE como se indicó anteriormente.

Para esta practica nos adelantaremos un poco y usaremos el LCD 2X16 y el modulo PCF8574 que es un expensor de I/Os por I2C (este tema lo cubriremos de lleno en una practica posterior por el momento solo trabajaremos con la librería para el PCF8574). Para poder utilizar el PC8574 vamos a la opción “Connectivity” y seleccionamos ahora “I2C1” los pines PB6 y PB7 se habilitarán de color verde, después en la columna central en la opción “I2C” seleccionamos “I2C” después abajo en “Configuration” seleccionamos “Parameter Settings” y nos aseguramos que la primera opción “I2C Speed Mode” este “Standard Mode” y en la segunda opción “I2C Clock Speed (Hz)” este 100 KHz (100000 Hz) la configuración final se muestra en la siguiente figura.

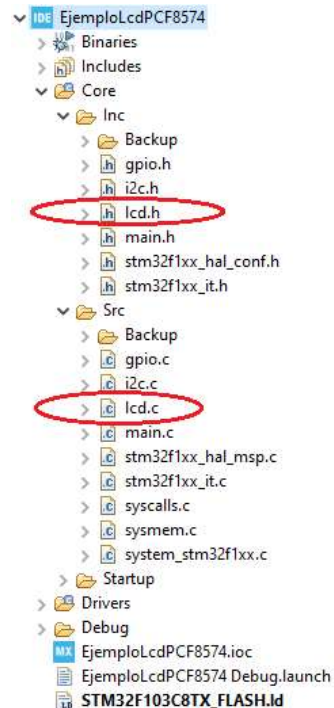


El código de la práctica es el siguiente. Recuerde que el siguiente código debe estar entre los comentarios “USER CODE BEGIN” y “USER CODE END”.

```
1 #include "main.h"
2 #include "i2c.h"
3 #include "gpio.h"
4
5 #include "lcd.h"
6
7 char str1[] = "Prueba LCD";
8 char str2[] = "LCD 2X16";
9 char str3[] = "Universidad";
10 char str4[] = "Tecnologica";
11
12 void SystemClock_Config(void);
13
14 int main(void)
15 {
16     HAL_Init();
17
18     SystemClock_Config();
19
20     MX_GPIO_Init();
21     MX_I2C1_Init();
22     Lcd_Init();
23
24     while (1)
25     {
26         Lcd_Gotoxy(1, 1);           // posicionamos el cursor en 1,1 (línea 1)
27         Lcd_Print(str1);           // escribimos el primer mensaje
28         Lcd_Gotoxy(1, 2);           // posicionamos el cursor en 1,2 (línea 2)
29         Lcd_Print(str2);           // escribimos el segundo mensaje
30         HAL_Delay(2000);           // esperamos 2 segundos
31         Lcd_Command(LIMPIARLCD);   // limpiamos la pantalla LCD
32         HAL_Delay(10);             // esperamos 10ms a que termine de ejecutar el comando
33         Lcd_Gotoxy(1, 1);           // posicionamos el cursor en 1,1 (línea 1)
34         Lcd_Print(str3);           // escribimos el tercer mensaje
35         Lcd_Gotoxy(1, 2);           // posicionamos el cursor en 1,2 (línea 2)
36         Lcd_Print(str4);           // escribimos el cuarto mensaje
37         HAL_Delay(2000);           // esperamos 2 segundos
38         Lcd_Command(LIMPIARLCD);   // limpiamos la pantalla LCD
39         HAL_Delay(10);             // esperamos 10ms a que termine de ejecutar el comando
40     }
41 }
```

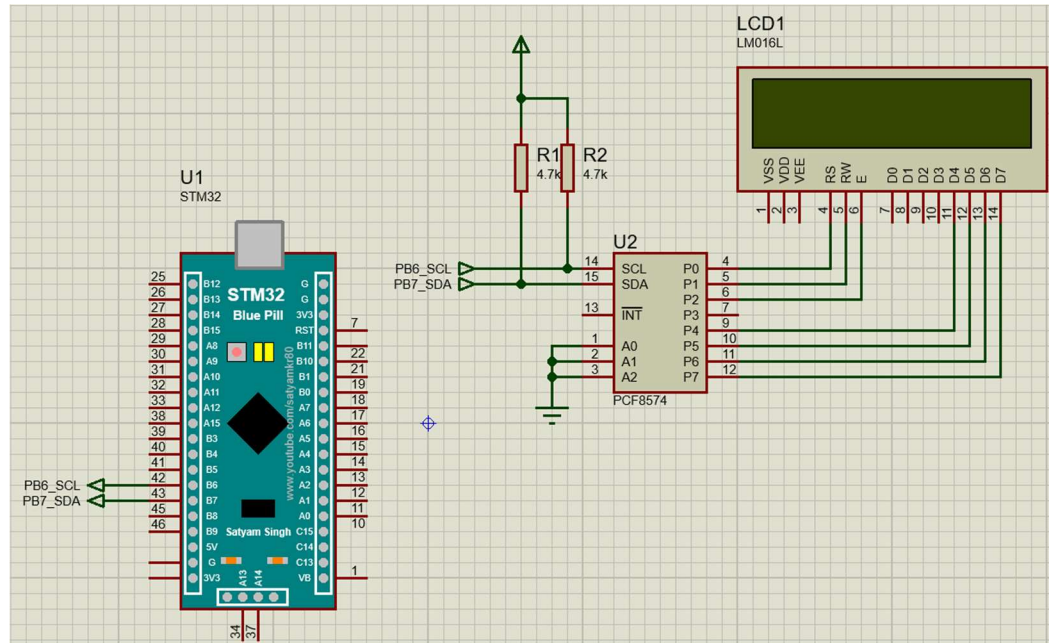
Para esta práctica se usará la librería “lcd.h” incluida en la línea número 5 de la imagen anterior. Esta librería está compuesta por dos archivos el “lcd.c” y el “lcd.h” estos archivos deben guardarse en ciertas carpetas del proyecto. Para agregar estos archivos al proyecto primero descargamos los archivos del repositorio de GitHub después abrimos la carpeta del “workspace” donde estamos guardando los programas y vamos a la carpeta del programa para este ejemplo el proyecto es “EjemploLcdPCF8574” después abrimos la carpeta “Core/Src” y en esta carpeta guardamos el archivo

“lcd.h” después regresamos un nivel arriba y ahora seleccionamos “Core/Inc” y en esta carpeta agregamos el archivo “lcd.h” después en el STM32CubeIDE seleccionamos el proyecto “EjemploLcdPCF8574” y damos click derecho y nos despliega un menú y buscamos la opción “Refresh” y los archivos nos aparecerán de la siguiente manera.

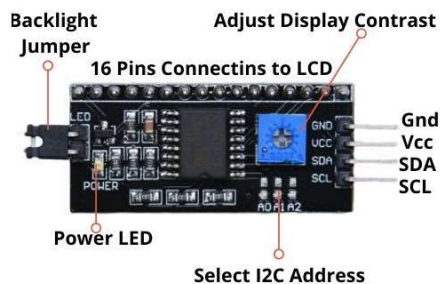


5. Esquemático del circuito.

La simulación de la práctica se muestra a continuación.



En el circuito anterior tenemos que considerar lo siguiente, el LCD se alimenta con 5V si se alimenta con 3.3V no va a funcionar correctamente la siguiente imagen muestra en detalle el modulo PCF8574.



1. Tenemos un Jumper para el Backlight si quitamos ese jumper la luz de fondo del LCD se apagará.
2. Tenemos un potenciómetro para ajustar el contraste del LCD en caso de tener un contraste muy alto ajustamos lo ajustamos con este potenciómetro hasta que solo resalte el mensaje que se va a mostrar.
3. Por último, tenemos cuatro pines que son la interfaz de comunicación y alimentación en las líneas SDA y SCL se conectan a los pines mostrados en el circuito esquemático. En los pines de alimentación para el correcto funcionamiento de la pantalla tenemos que alimentar la tarjeta con 5V y tenemos dos opciones la primera conectamos los 5V directo del programador ST-Link y la segunda es conectar el módulo PCF8574 al pin de 5V de la tarjeta y después tenemos que conectar la tarjeta “bluepill” a la computadora a través de un cable micro usb.

Nota: Si se usa esta última opción no conecte el programador ST-Link mientras alimenta la tarjeta con el cable de micro usb por que podría causar un daño en los puertos USB, si programa el microcontrolador desconecte el cable micro usb y una vez programado desconecte el ST-Link y conecte el cable micro usb.

6. Mejora

Modifique el programa para que aparezca su nombre(s) y después sus apellidos los nombre(s) uno en cada linea y los apellidos de igual forma.

7. Observaciones.

Esta sección es para que el alumno anote sus observaciones.

8. Conclusiones.

Esta sección es para que el alumno anote sus conclusiones.



9. Importante.

La práctica deberá ser validada en el salón de clases antes de anexar el reporte al manual de prácticas. Una vez validada realizar el reporte de práctica como se anteriormente y anexar al manual de prácticas que se entregará a final del curso.