

# Sistemas Digitales Laboratorio Semana 11

2022-1  
Por Kalun Lau

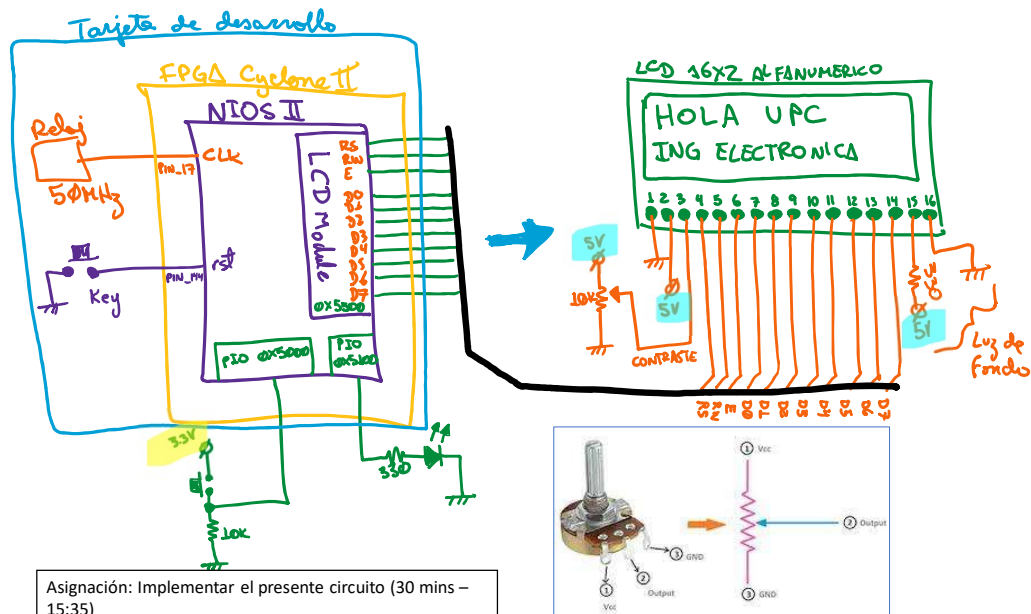
1

## Agenda

- Manipulación de datos en Eclipse
- Uso del LCD

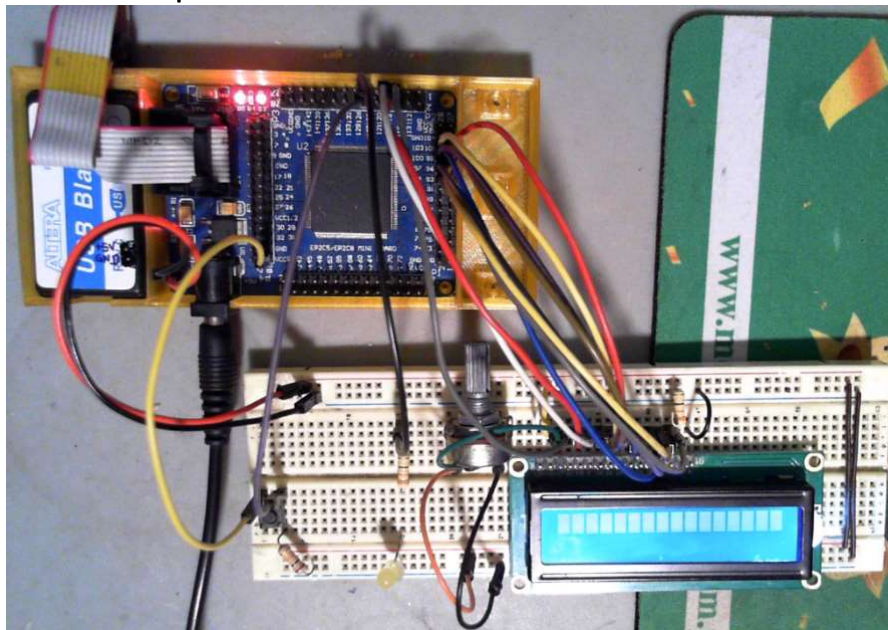
2

## Ejercicio: Implementar el siguiente circuito



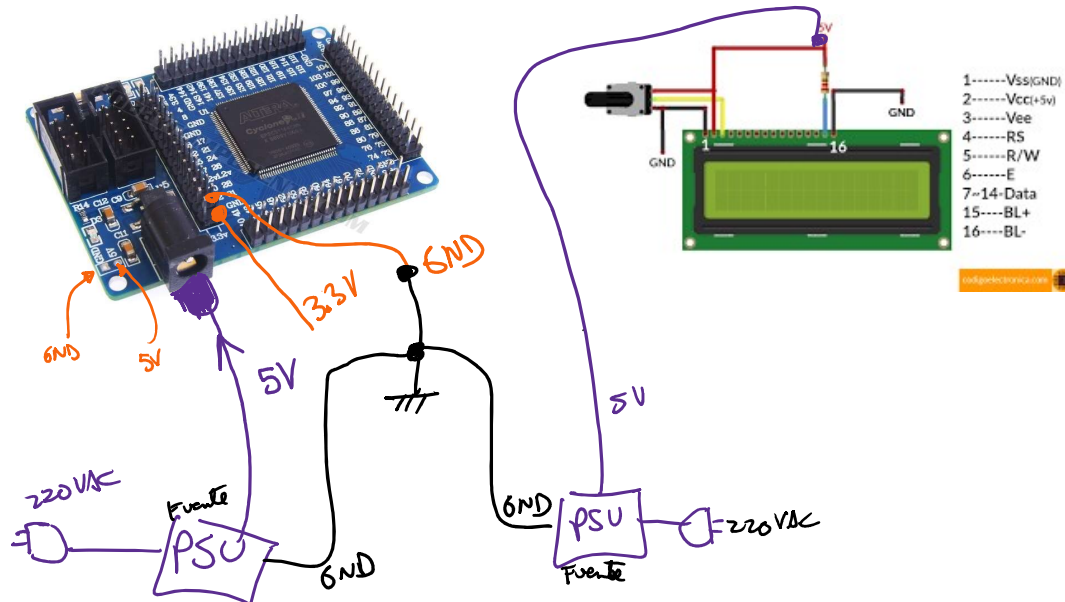
3

## Circuito implementado



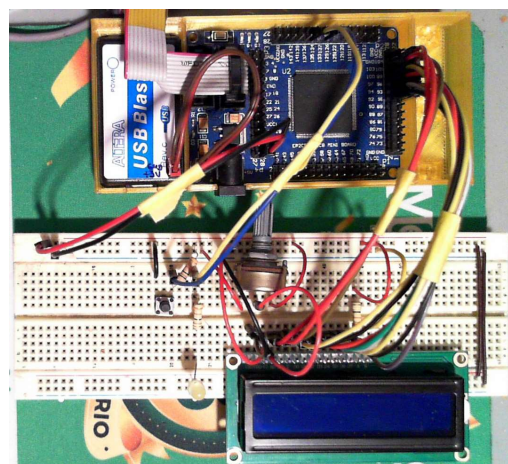
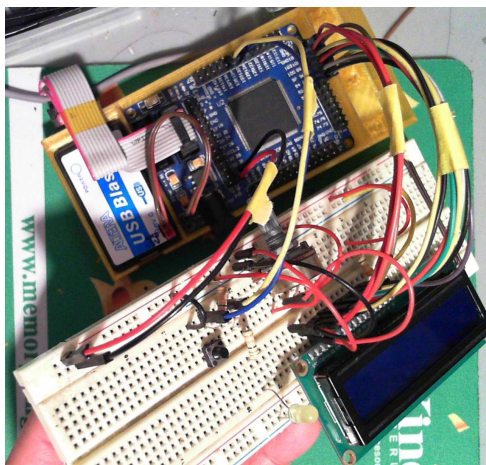
4

## Detalle de fuentes en el ejemplo:



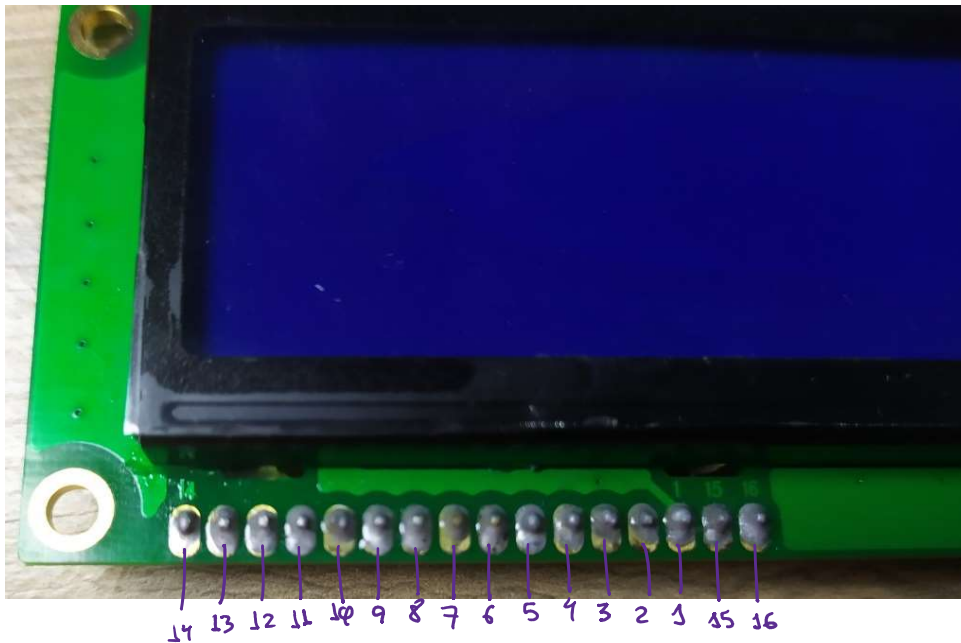
5

## Implementaciones



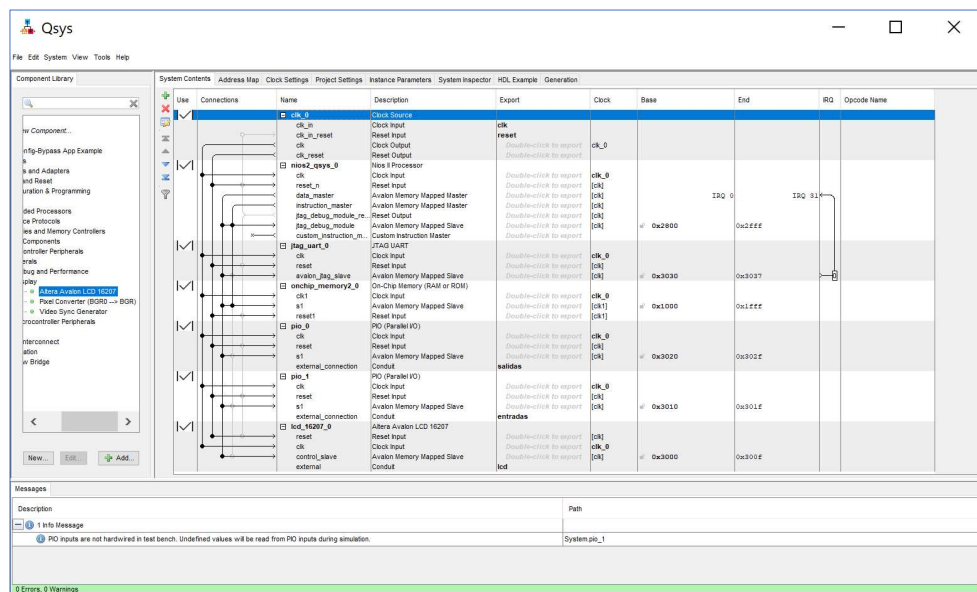
6

## Caso: LCD desconocido



7

Qsys:



8

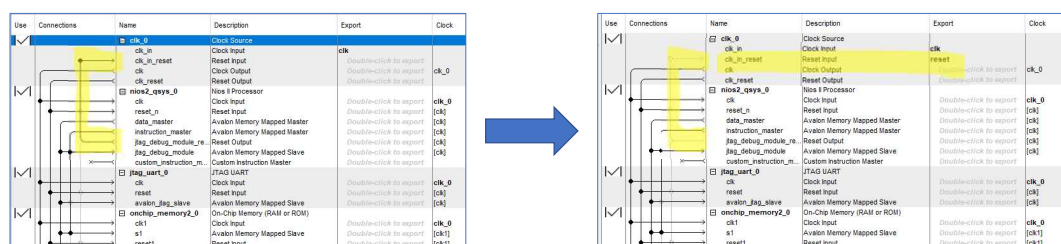
## Comentarios:

- Tener en cuenta el mapeo de direcciones, no debe de haber conflictos
- El módulo LCD tiene dirección 0x3000
- El PIO entrada de 8 bits tiene dirección 0x3010
- El PIO salida de 8 bits tiene dirección 0x3020

9

## Aclaración sobre uso de la entrada de reset al NIOSII

- Para tener una entrada de reset al NIOSII es necesario hacer una modificación en el Qsys:



- Considerar que la entrada habilitada para el reset del NIOS II **es activo en bajo** por lo que se Debera de tener esto en cuenta en el hardware utilizado en la implementación así como en la instanciación del NIOS en el VHDL

10

## VHDL:

```

1 library IEEE;
2 use IEEE.std_logic_1164.all;
3 use IEEE.std_logic_arith.all;
4 use IEEE.std_logic_unsigned.all;
5
6 entity mensajeador is
7   port( clk, reset      : in std_logic;
8         botones         : in std_logic_vector(7 downto 0);
9         leds            : out std_logic_vector(7 downto 0);
10        display_e, display_rs, display_rw: out std_logic;
11        display_data     : inout std_logic_vector(7 downto 0));
12 end mensajeador;
13
14 architecture estructura of mensajeador is
15   .
16   .
17   .
18   component procesador_niosii is
19     port (
20       clk_clk      : in  std_logic      := 'X';           -- clk
21       salidas_export : out std_logic_vector(7 downto 0); -- export
22       entradas_export : in  std_logic_vector(7 downto 0) := (others => 'X'); -- export
23       reset_reset_n : in  std_logic      := 'X';           -- reset_n
24       lcd_RS        : out std_logic;      -- RS
25       lcd_RW        : out std_logic;      -- RW
26       lcd_data      : inout std_logic_vector(7 downto 0) := (others => 'X'); -- data
27       lcd_E         : out std_logic;      -- E
28     );
29   end component procesador_niosii;
30
31 begin
32   .
33   .
34   .
35   u0 : component procesador_niosii
36     port map (
37       clk_clk      => clk,           -- clk.clk
38       salidas_export => leds,        -- salidas.export
39       entradas_export => botones,    -- entradas.export
40       reset_reset_n => reset,        -- reset.reset_n
41       lcd_RS        => display_rs,   -- lcd.RS
42       lcd_RW        => display_rw,   -- lcd.RW
43       lcd_data      => display_data, -- lcd.data
44       lcd_E         => display_e,    -- lcd.E
45     );
46
47 end estructura;

```

11

## Nota:

- Durante el desarrollo del VHDL de la instanciación se colocó una señal interna para negar el reset. Esto es para pulsadores que tengan la configuración activo en alto. Se deberá de prescindir de dicha señal ya que la entrada de reset del NIOSII es activo en bajo y el pulsador integrado en la tarjeta de FPGA (botón 'key' en PIN\_144) es activo en bajo.
- Tener en cuenta que deberá de activarse el Weak Pullup para el PIN\_144 en el PinPlanner ya que el botón 'key' no tiene implementado una resistencia de pullup.

12



## Pin Planner:

**Top View Wire Bond**

**Cyclone II EP2C5T144C8**

Node Name	Direction	Location	I/O Bank	Fitter Location	I/O Standard	Current Strength	Weak Pull-Up Resistor
botones[7]	Input			PIN_22	3.3-V LVTTTL (default)	24mA (default)	
botones[6]	Input			PIN_21	3.3-V LVTTTL (default)	24mA (default)	
botones[5]	Input			PIN_18	3.3-V LVTTTL (default)	24mA (default)	
botones[4]	Input			PIN_120	3.3-V LVTTTL (default)	24mA (default)	
botones[3]	Input			PIN_119	3.3-V LVTTTL (default)	24mA (default)	
botones[2]	Input			PIN_118	3.3-V LVTTTL (default)	24mA (default)	
botones[1]	Input			PIN_121	3.3-V LVTTTL (default)	24mA (default)	
botones[0]	Input	PIN_125	2	PIN_125	3.3-V LVTTTL (default)	24mA (default)	
clk	Input	PIN_17	1	PIN_17	3.3-V LVTTTL (default)	24mA (default)	
display_data[7]	Bidir	PIN_94	3	PIN_94	3.3-V LVTTTL (default)	24mA (default)	
display_data[6]	Bidir	PIN_96	3	PIN_96	3.3-V LVTTTL (default)	24mA (default)	
display_data[5]	Bidir	PIN_97	3	PIN_97	3.3-V LVTTTL (default)	24mA (default)	
display_data[4]	Bidir	PIN_99	3	PIN_99	3.3-V LVTTTL (default)	24mA (default)	
display_data[3]	Bidir	PIN_100	3	PIN_100	3.3-V LVTTTL (default)	24mA (default)	
display_data[2]	Bidir	PIN_101	3	PIN_101	3.3-V LVTTTL (default)	24mA (default)	
display_data[1]	Bidir	PIN_103	3	PIN_103	3.3-V LVTTTL (default)	24mA (default)	
display_data[0]	Bidir	PIN_104	3	PIN_104	3.3-V LVTTTL (default)	24mA (default)	
display_rs	Output	PIN_113	2	PIN_113	3.3-V LVTTTL (default)	24mA (default)	
display_rv	Output	PIN_114	2	PIN_114	3.3-V LVTTTL (default)	24mA (default)	
leds[7]	Output	PIN_115	2	PIN_115	3.3-V LVTTTL (default)	24mA (default)	
leds[6]	Output			PIN_132	3.3-V LVTTTL (default)	24mA (default)	
leds[5]	Output			PIN_57	3.3-V LVTTTL (default)	24mA (default)	
leds[4]	Output			PIN_58	3.3-V LVTTTL (default)	24mA (default)	
leds[3]	Output			PIN_59	3.3-V LVTTTL (default)	24mA (default)	
leds[2]	Output			PIN_126	3.3-V LVTTTL (default)	24mA (default)	
leds[1]	Output			PIN_60	3.3-V LVTTTL (default)	24mA (default)	
leds[0]	Output			PIN_122	3.3-V LVTTTL (default)	24mA (default)	
reset	Input	PIN_122	2	PIN_122	3.3-V LVTTTL (default)	24mA (default)	On
<new node>							

13

## Programmer:

**Programmer - D:/Sisdig20221/el42\_niosii\_sem11\_7am/mensajeador - mensajeador - [Chain1.cdf]\***

File Edit View Processing Tools Window Help

Hardware Setup... USB-Blaster [USB-0] Mode: JTAG Progress: 100% (Successful)

☐ Enable real-time ISP to allow background programming (for MAX II and MAX V devices)

File	Device	Checksum	Usercode	Program/Configure	Verify	Bank	Camir	acurir	ras	ISP	LAM
output_files/mensajeador.sof	EP2C5T144	001D7534	001D7534	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Start Stop Auto Del Delete Add File Range F Save F Add Devi Up Down

TDI TD0

EP2C5T144

14

## Código ejemplo:

- Se esta enviando Mensajes tanto a la consola como al LCD
- Cuando se presiona el pulsador del PIO de entrada empezará a parpadear el LED ubicado en el PIO de salida

```

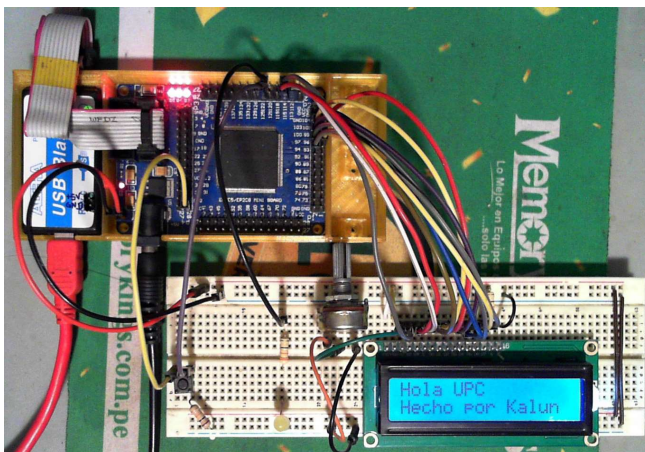
hello_world_small.c
1#include "sys/alt_stdio.h"
2#include "system.h"
3#include "unistd.h"
4#include "altera_avalon_pio_regs.h"
5#include "altera_avalon_lcd_16207_regs.h"
6
7#define LCD_WR_COMMAND_REG 0
8#define LCD_RD_STATUS_REG 1
9#define LCD_WR_DATA_REG 2
10#define LCD_RD_DATA_REG 3
11#define LCD_0_BASE 0x3000
12
13void lcd_init(void) {
14    usleep(15000);
15    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
16    usleep(4100);
17    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
18    usleep(100);
19    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
20    usleep(5000);
21    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
22    usleep(100);
23    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
24    usleep(100);
25    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
26    usleep(100);
27    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
28    usleep(100);
29    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
30    usleep(2000);
31    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
32    usleep(2000);
33}

35void ESCRIBE_MENSAJE(const char *cadena, unsigned char tam) {
36    unsigned char i=0;
37    for(i=0;i<tam;i++) {
38        IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
39        usleep(100);
40    }
41}
42
43int main()
44{
45    lcd_init();
46    alt_putstr("Hello from Nios II!\n");
47    alt_putstr("Hola UPCino, soy el NIOS II\n");
48    alt_putstr("Hecho por Kalun Lau\n");
49    alt_putstr("Prueba Semana 11\n");
50    alt_putstr("Presiona el boton para que parpadee el LED\n");
51    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
52    usleep(2000);
53    ESCRIBE_MENSAJE("Hola UPC", 8);
54    usleep(2000);
55    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC0);
56    usleep(2000);
57    ESCRIBE_MENSAJE("Hecho por Kalun", 15);
58    usleep(2000);
59    while (1) {
60        if ((IORD ALTERA_AVALON_PIO_DATA(0x3010) & 0x01) == 1) {
61            IOWR ALTERA_AVALON_PIO_DATA(0x3020, 0x01);
62            usleep(100000);
63            IOWR ALTERA_AVALON_PIO_DATA(0x3020, 0x00);
64            usleep(100000);
65        }
66        IOWR ALTERA_AVALON_PIO_DATA(0x3020, 0x00);
67    }
68}

```

15

## Pruebas de implementación



Problems Tasks Console Properties Nios II Console

hola mundo Nios II Hardware configuration - cable: USB-Blaster on localhost [USB-0] device ID: 1 instance ID: 0 name: jaguar\_0

```

Hello from Nios II!
Hola UPCino, soy el NIOS II
Hecho por Kalun Lau
Prueba Semana 11
Presiona el boton para que parpadee el LED
Hello from Nios II!
Hola UPCino, soy el NIOS II
Hecho por Kalun Lau
Prueba Semana 11
Presiona el boton para que parpadee el LED
Hello from Nios II!
Hola UPCino, soy el NIOS II
Hecho por Kalun Lau
Prueba Semana 11
Presiona el boton para que parpadee el LED

```

16



## Fin de la sesión

- Revisar documentación:

- <https://github.com/tocache/Altera-Cyclone-II-FPGA/blob/master/NIOS%20II/03%20NIOSII%20LCD.pdf>
- [https://people.ece.cornell.edu/land/courses/ece5760/DE1\\_SOC/embedded\\_ip\\_users\\_guide.pdf](https://people.ece.cornell.edu/land/courses/ece5760/DE1_SOC/embedded_ip_users_guide.pdf)
- <https://www.sparkfun.com/datasheets/LCD/HD44780.pdf>
- <https://www.engineersgarage.com/making-custom-characters-on-lcd-using-arduino/>