

# Sistemas Digitales

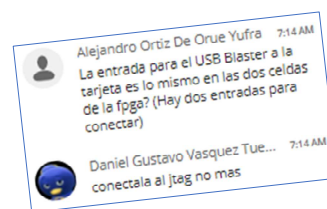
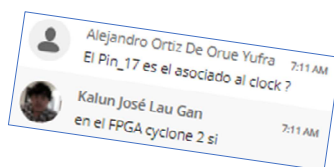
Semana 13 Laboratorio

Semestre 2020-2

Profesor Kalun José Lau Gan

1

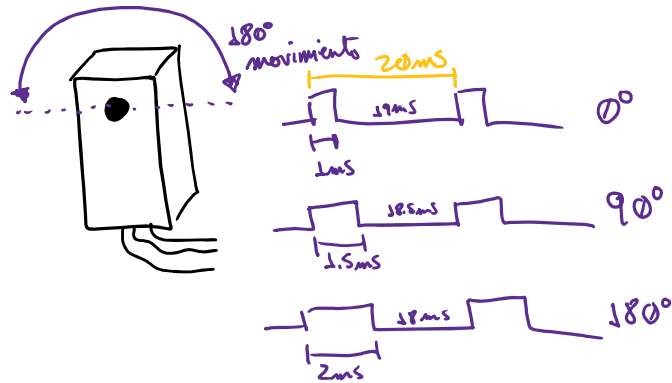
## Preguntas previas



- ¿Cómo mapeo las direcciones en el NIOS? Hay alguna recomendación?
  - La recomendación es que guarden un cierto orden y que NO HAYA CONFLICTO

2

## Sobre servomecanismos:



3

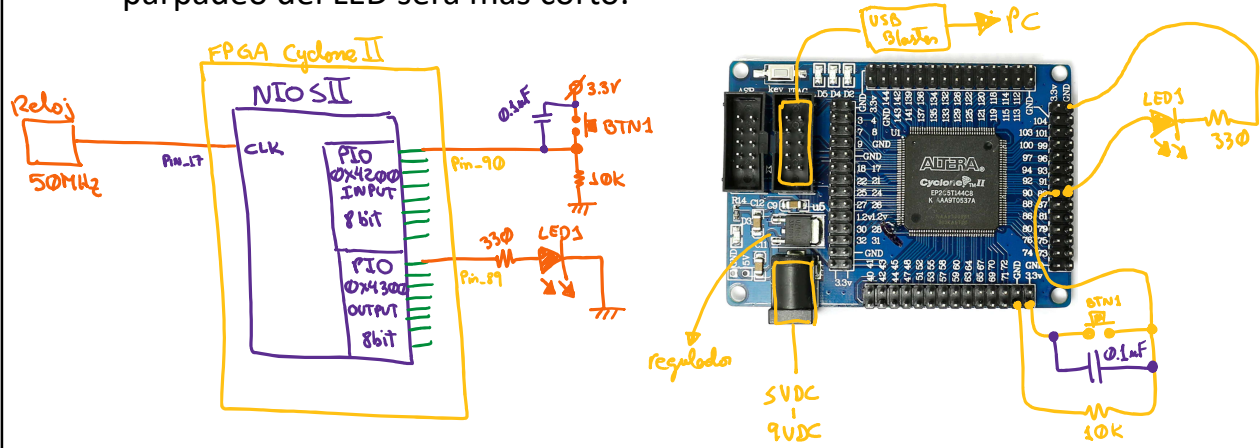
## Agenda:

- Ejemplos de aplicaciones con el NIOS II
  - Manejo de entradas digitales con el PIO
  - Interface con display LCD alfanumérico

4

## Ejemplo1: Lectura de PIO de entradas

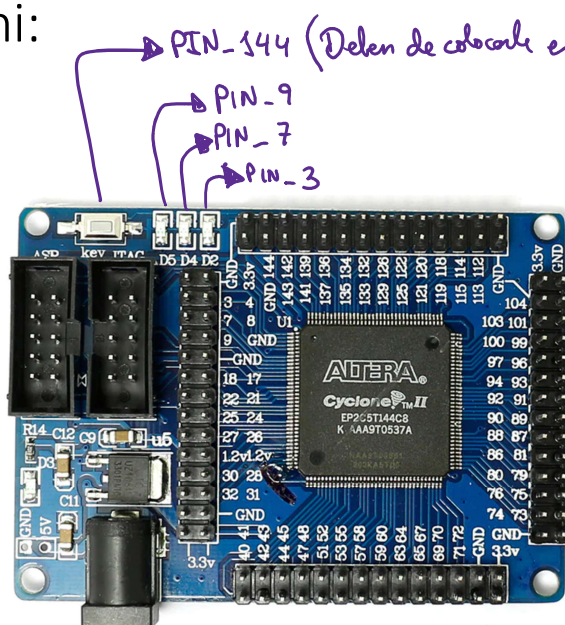
- En el siguiente circuito, cuando se presione el botón el tiempo de parpadeo del LED será mas corto.



5

## Usando los periféricos integrados del EP2C5T Mini:

Mini:



Dos opciones:

- 1) Soldar la resistencia
- 2) Activar la resisten interna de Pull-up

En el Pin Planner:

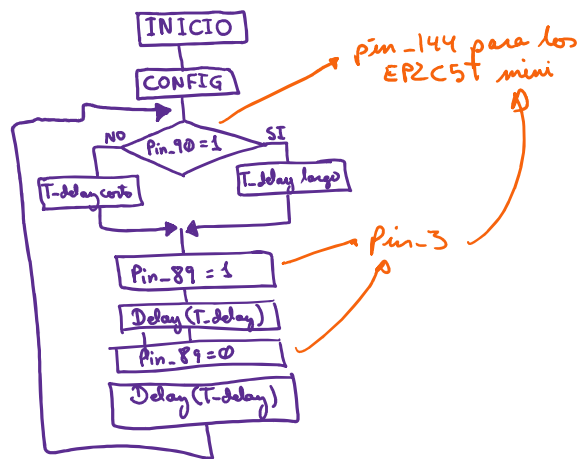
Node Name	Direction	Location	I/O Bank	Filter Location	I/O Standard	Current Strength	Weak Pull-Up Resistor
boton	Input	PIN_144	2		3.3-V...fault	24mA (default)	On

*Nota: Tanto el botón como los LED's son activos en bajo*

6

## Ejemplo1: Lectura de PIO de entradas

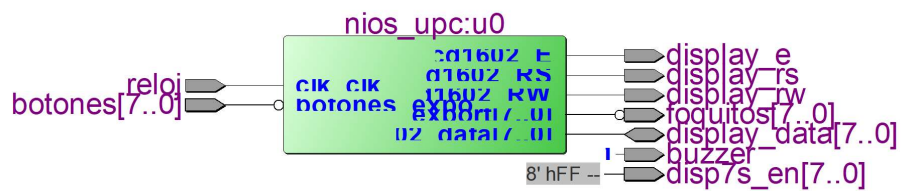
- Algoritmo en diagrama de flujo



7

## NIOS II implementado en vista RTL:

- Nota: Orientado a la tarjeta A-C2FB



8

## Ejemplo1: Lectura de PIO de entradas

- Código en C para NIOS II

```
#include "sys/alt_stdio.h"
#include "altera_avalon_pio_regs.h"
#include "system.h"

volatile unsigned char dato_in = 0;
volatile long tiempo = 0;

void main(void)
{
    alt_putstr("Hello from Nios II!\n");
    while(1){
        dato_in = (IORD_ALTERA_AVALON_PIO_DATA(0x4200)) & 0x01;
        if(dato_in == 1){
            tiempo = 200000;
        }
        else{
            tiempo = 100000;
        }
        IOWR_ALTERA_AVALON_PIO_DATA(0x4300, 0x01);
        usleep(tiempo);
        IOWR_ALTERA_AVALON_PIO_DATA(0x4300, 0x00);
        usleep(tiempo);
    }
    return 0;
}
```

9

## El LCD alfanumérico



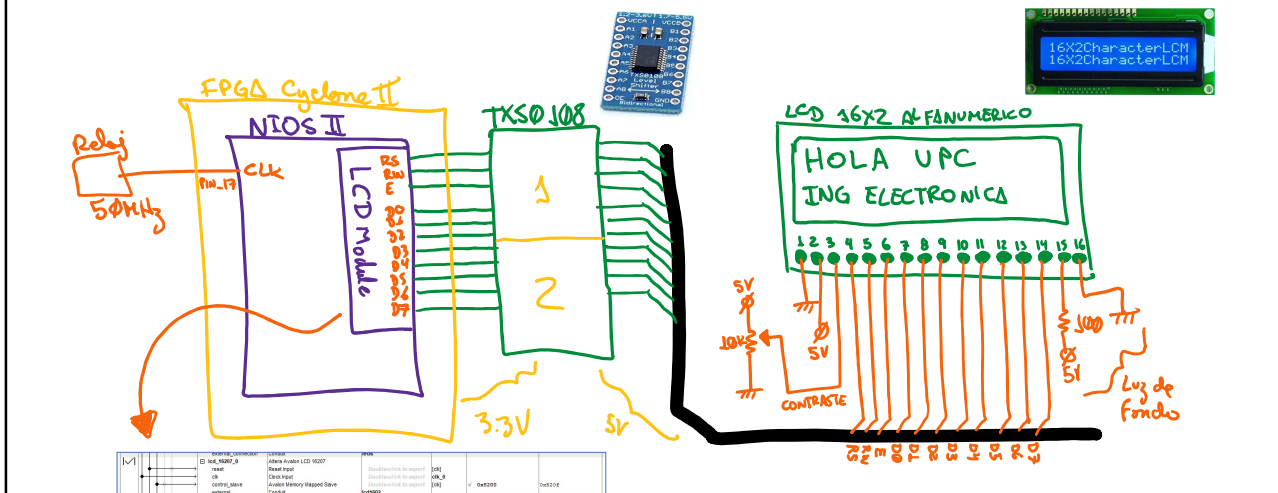
- Se pueden visualizar mensajes tanto en letras como en números.
- Su ROM de caracteres se asemeja a la tabla ASCII de 7bits
- Posee hasta ocho caracteres personalizados (CGRAM)
- Interface de 8 bits de datos + 3 líneas de control

	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
xxxx0001	(2)	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
xxxx0010	(3)	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>
xxxx0011	(4)	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N
xxxx0100	(5)	O	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]
xxxx0101	(6)	^	_	`	a	b	c	d	e	f	g	h	i	j	k	l
xxxx0110	(7)	m	n	o	p	q	r	s	t	u	v	w	x	y	z	{
xxxx0111	(8)		~													
xxxx1000	(1)															
xxxx1001	(2)															
xxxx1010	(3)															
xxxx1011	(4)															
xxxx1100	(5)															
xxxx1101	(6)															
xxxx1110	(7)															
xxxx1111	(8)															

10

## Ejemplo2: Interface a display LCD alfanumérico

- Implementar el siguiente circuito y generar la visualización mensajes



11

## Inicialización del LCD en el C para NIOS II:

- El LCD estará funcionando en modo 8 bits de datos (total 11 líneas que se conectan al NIOS II)
- Esta función se ejecuta por única al inicio de operación de la aplicación.
- Tener en cuenta que al inicio del código se debe incluir la librería y definiciones siguientes:

```
#include "altera_avalon_lcd_16207_regs.h"
```

```
#define LCD_WR_COMMAND_REG 0
#define LCD_RD_STATUS_REG 1
#define LCD_WR_DATA_REG 2
#define LCD_RD_DATA_REG 3
#define LCD_0_BASE 0x5200
```

↑  
dirección asignada  
en Qsys

```
void lcd_init(void){
    usleep(15000); /* Wait for more than 15 ms before init */

    /* Set function code four times - 8-bit, 2 line, 5x7 mode */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(4100); /* Wait for more than 4.1 ms */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(100); /* Wait for more than 100 us */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(5000); /* Wait for more than 100 us */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
    usleep(100); /* Wait for more than 100 us */

    /* Set Display to OFF */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
    usleep(100);

    /* Set Display to ON */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
    usleep(100);

    /* Set Entry Mode - Cursor increment, display doesn't shift */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
    usleep(100);

    /* Set the Cursor to the home position */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);
    /* Display clear */
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
    usleep(2000);
}
```

12

## Comandos en el LCD alfanumérico

- 0x01 – Limpia el display
  - 0x02 – Mueve el cursor al inicio de la primera línea
  - 0x0C – Apaga el cursor
  - 0x0E – Cursor visible estático
  - 0x0F – Cursor visible parpadeando
  - 0x10 – Mueve cursor una posición a la izquierda
  - 0x14 – Mueve cursor una posición a la derecha
  - 0xC0 – Mueve cursor al inicio de la segunda línea
  - 0x94 – Mueve el cursor al inicio de la terca línea
  - 0xD4 – Mueve el cursor al inicio de la cuarta línea
- } Para LCDs de cuatro líneas*

13

## Visualización de mensajes en el display:

- Visualizar “hola” en la primera linea

```
void main(void){
    //Funcion para inicializar el LCD
    lcd_init();

    //Comando para ubicarnos al inicio de la primera línea
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);

    //Visualización del mensaje "hola"
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'h');
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'o');
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'l');
    usleep(100);
    IOWR(LCD_0_BASE, LCD_WR_DATA_REG, 'a');
    usleep(100);

    while(1);

    return(0);
}
```

14

## Visualización de mensajes en el display:

- Visualizar "hola" (cadena) en la primera línea

```
const unsigned char cadena[] = {"hola"};
int x_var = 0;

void main(void){
    //Funcion para inicializar el LCD
    lcd_init();

    //Comando para ubicarnos al inicio de la primera línea
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);

    //Visualización del mensaje "hola"
    for(x_var=0;x_var<4;x_var++){
        IOWR(LCD_0_BASE, LCD_WR_DATA_REG,cadena[x_var]);
        usleep(100);
    }
    while(1);
    return(0);
}
```

15

## Visualización de mensajes en el display:

- Parametrizando en una función la visualización de una cadena

```
void ESCRIBE_MENSAJE(const char *cadena,unsigned char tam)
{
    unsigned char i = 0;
    for(i = 0; i<tam; i++)
    {
        IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
        usleep(100);
    }
}
```

*tam: Tamaño de la  
cadena*

- Ejemplo

```
void main(void){
    //Funcion para inicializar el LCD
    lcd_init();

    //Comando para ubicarnos al inicio de la primera línea
    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
    usleep(2000);

    //Visualización del mensaje "hola"
    ENVIA_MENSAJE("hola", 4);
    while(1);
    return(0);
}
```

16



## Programa ejemplo completo:

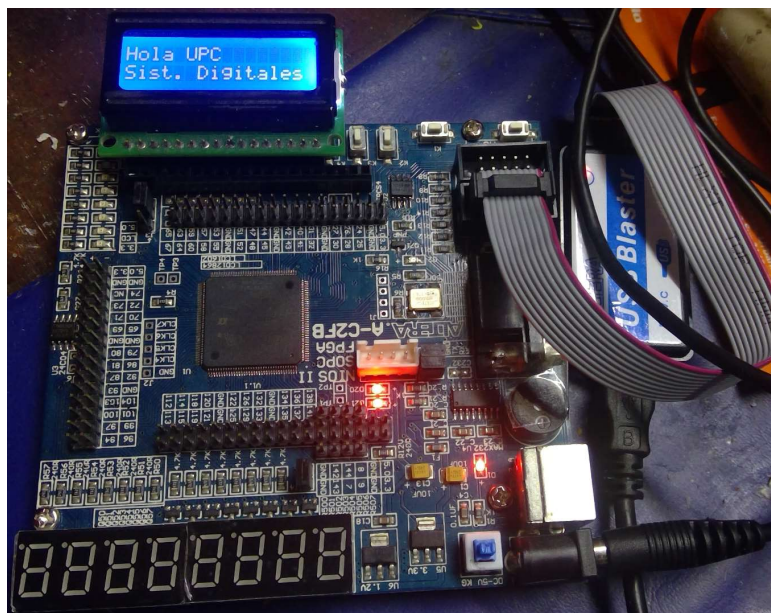
```

1#include "sys/alt_stdio.h"
2#include "system.h"
3#include "altera_avalon_lcd_16207_regs.h"
4
5#define LCD_WR_COMMAND_REG 0
6#define LCD_RD_STATUS_REG 1
7#define LCD_WR_DATA_REG 2
8#define LCD_RD_DATA_REG 3
9#define LCD_0_BASE 0x5200
10
11void lcd_init(void){
12    usleep(15000);
13    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
14    usleep(4100);
15    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
16    usleep(100);
17    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
18    usleep(5000);
19    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x38);
20    usleep(100);
21    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x08);
22    usleep(100);
23    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x0C);
24    usleep(100);
25    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x06);
26    usleep(100);
27    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02);
28    usleep(2000);
29    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x01);
30    usleep(2000);
31}
32
33void ESCRIBE_MENSAJE(const char *cadena,unsigned char tam)
34{
35    unsigned char i = 0;
36    for(i = 0; i<tam; i++)
37    {
38        IOWR(LCD_0_BASE, LCD_WR_DATA_REG, cadena[i]);
39        usleep(100);
40    }
41}
42
43void main(void)
44{
45    lcd_init();
46    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0x02); //Primera linea
47    usleep(2000);
48    ESCRIBE_MENSAJE("Hola UPC", 8);
49    IOWR(LCD_0_BASE, LCD_WR_COMMAND_REG, 0xC0); //Segunda linea
50    usleep(2000);
51    ESCRIBE_MENSAJE("Sist. Digitales", 15);
52    while (1);
53}

```

17

## Implementación en la tarjeta A-C2FB



18

Fin de la sesión