

Microcontroladores

Laboratorio Sesión 9

Semestre: 2021-2

Profesor: Kalun José Lau Gan

1

Preguntas previas:

- ¿Cómo soldar bien?

- Tener buenas herramientas: cautín con control de temperatura, pinzas, pelacables, desarmadores, **pasta para soldar (flux)**, soldadura de buena calidad, esponja humedecida con agua, alcohol isopropílico.
- Practicar y desarrollar la habilidad (mirar videos instructionales)
- Verificar que lo soldado tenga una forma cónica, si esta en circunferencia significa que es un soldado frío.

- ¿Empresas que desarrollan PCBs?

- Jlcpcb.com
- Pcbway.com

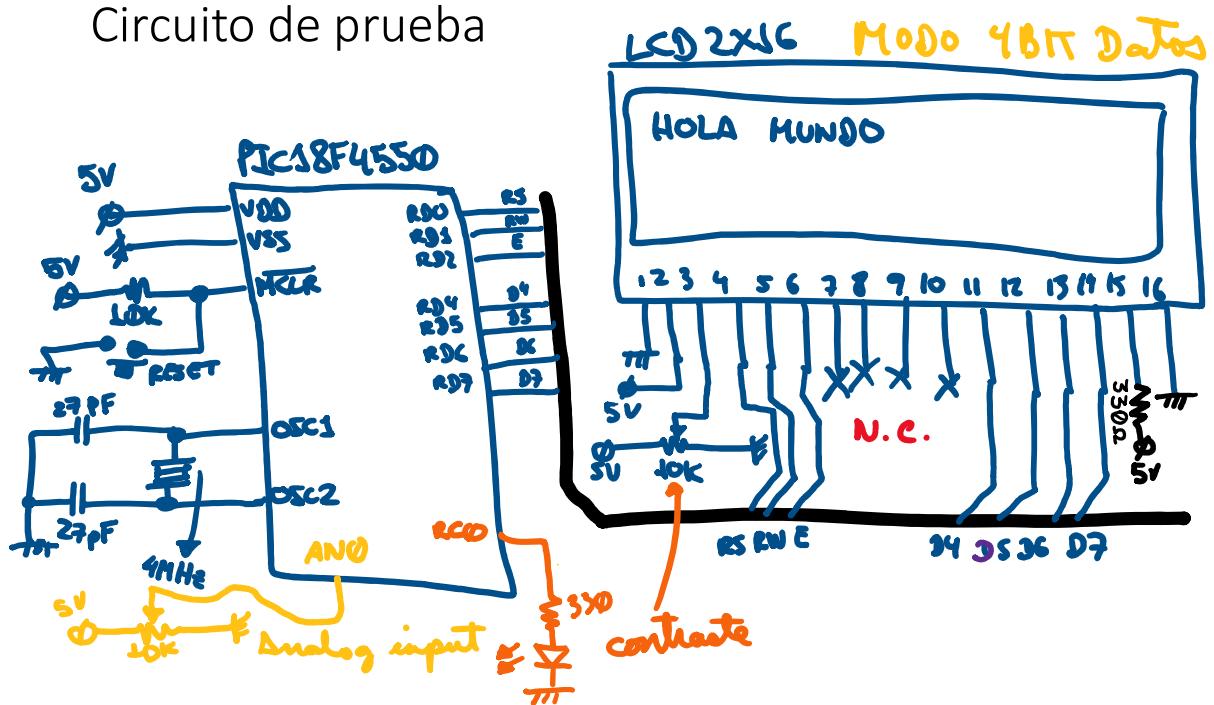
2

Agenda:

- Aspectos introductorios del XC8
- El display LCD alfanumérico HD44780
- Librería LCD para XC8
- El módulo conversor A/D en XC8

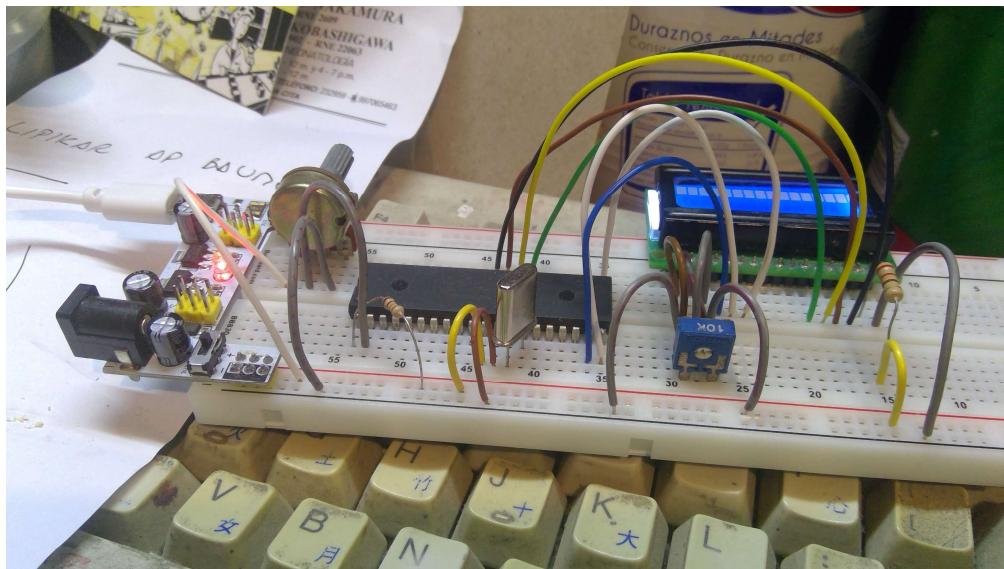
3

Circuito de prueba



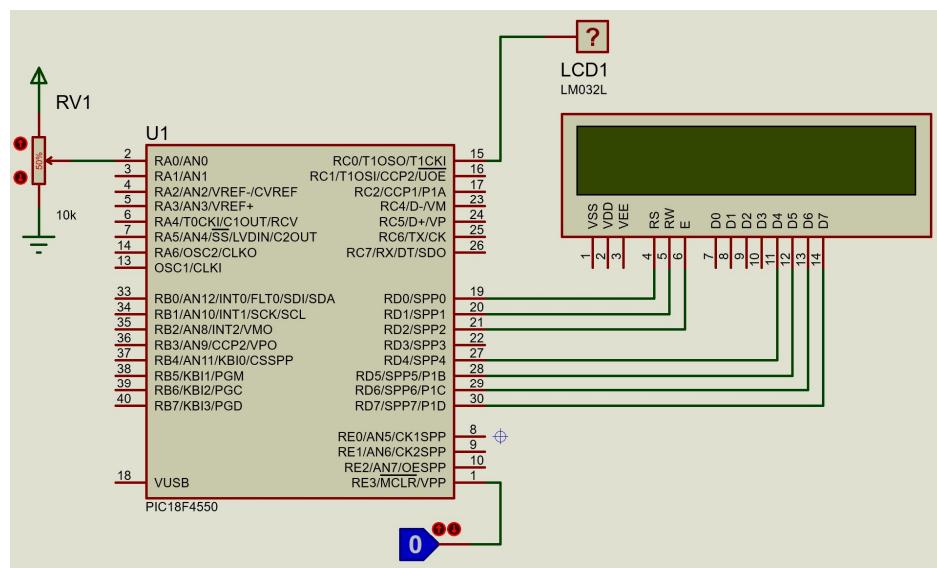
4

Implementación física



5

Circuito simulado en Proteus



6

Referencia de plantilla:

- La plantilla para los códigos en C se ha basado en la plantilla de programas en Arduino IDE:
 - Uso de función `setup()` donde se coloca los aspectos iniciales de configuración antes de correr el programa de la aplicación. En XC8 crearemos una función similar. Por ejemplo `configuración()`
 - Uso de función `loop()` donde se detalla el programa de la aplicación. En XC8 usaremos la función `main()` donde dentro llamaremos a la función `configuración` antes de detallar el programa de usuario.

```

/*
 *Title: My Awesome Arduino Program
 *Author: Liz Miller
 *Date: 02/15/2018
 *Version: v1.0
 *Purpose: This code shows you how to write an Arduino Program!
 */
void setup() {
  // put your setup code here, to run once:
}

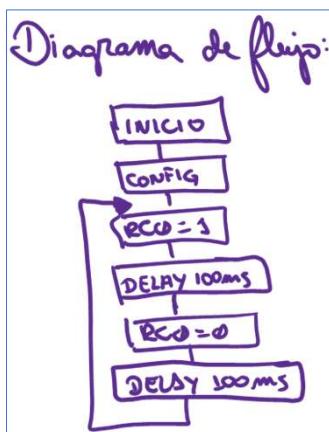
void loop() {
  // put your main code here, to run repeatedly:
}

```

Done Saving.
Arduino/Cenuine Uno on /dev/cu.usbmodem1421

7

Primer ejemplo en XC8: titilador de un bit por RCO



```

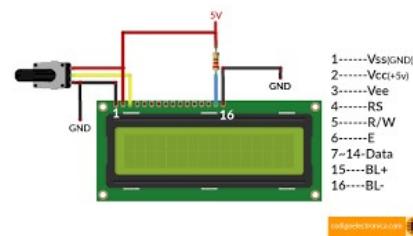
4 #pragma config PLLDIV = 1           // PLL Prescaler Selection
5 #pragma config CPUDIV = OSC1_PLL2 // System Clock Post
6 #pragma config FOSC = XTPPLL_XT // Oscillator Selection
7 #pragma config PWRT = ON          // Power-up Timer Enable
8 #pragma config BOR = OFF          // Brown-out Reset Enable
9 #pragma config WDT = OFF          // Watchdog Timer Enable
10 #pragma config CCP2MX = ON         // CCP2 MUX bit (CCP2M)
11 #pragma config PBADEN = OFF        // PORTB A/D Enable bit
12 #pragma config MCLRE = ON          // MCLR Pin Enable bit
13 #pragma config LVP = OFF          // Single-Supply ICSP
14
15 #include <xc.h>
16 #define _XTAL_FREQ 48000000UL
17 /*El microcontrolador tiene funcionando el PLL con 48MHz*/
18
19 void main(void) {
20     TRISCbts.RCO = 0;           //RC0 salida
21     while(1){
22         LATCbits.LC0 = 1;        //Encendemos el LED
23         __delay_ms(100);        //retardo de 100ms
24         LATCbits.LC0 = 0;        //Apagamos el LED
25         __delay_ms(100);        //retardo de 100ms
26     }
27     return;
28 }

```

8

El LCD alfanumérico HD44780

- Basado en el controlador Hitachi HD44780A
- Diferentes tamaños, desde 1x8 hasta 4x40
- Interface paralela de datos (4 ó 8 bits)
- Tiene control de contraste y luz de fondo
- Posee un ROM de caracteres predefinidos



9

El LCD alfanumérico HD44780

- ROM de caracteres:
 - Muy similar al código ASCII en 7 bits
 - El símbolo de grado (°) en ASCII es Alt+0167, en el ROM de caracteres del HD44780 es 0xDF
 - El símbolo “ñ” en ASCII es Alt+164, en el ROM de caracteres del HD44780 es 0xEE
 - Capacidad de ocho caracteres personalizados (CGRAM 0x00-0x07)

Upper	Lower	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
xxxx0000	(1)	0	Q	P	^	P				-	タ	ミ	ア	ぱ			
xxxx0001	(2)	!	1	A	Q	a	q			.	ア	フ	ミ	ä	q		
xxxx0010	(3)	"	2	B	R	b	r			「	イ	ツ	×	ø			
xxxx0011	(4)	#	3	C	S	c	s			」	ウ	テ	€	≈			
xxxx0100	(5)	\$	4	D	T	d	t			、	エ	ト	ト	μ	ø		
xxxx0101	(6)	%	5	E	U	e	u			・	オ	ナ	ニ	σ	ü		
xxxx0110	(7)	&	6	F	V	f	v			ヲ	カ	ニ	ヨ	ρ	Σ		
xxxx0111	(8)	'	7	G	W	g	w			ア	キ	フ	ラ	g	π		
xxxx1000	(1)	(8	H	X	h	x			イ	ク	ネ	リ	χ			
xxxx1001	(2))	9	I	Y	i	y			タ	テ	ル	・	γ			
xxxx1010	(3)	*	:	J	Z	j	z			エ	コ	ハ	レ	j	‡		
xxxx1011	(4)	+	;	K	L	k	l			オ	サ	ヒ	ロ	*	δ		
xxxx1100	(5)	,	<	L	¥	1	l			タ	シ	フ	ワ	¶	¶		
xxxx1101	(6)	-	=	M]	m	}			ュ	ス	ヘ	ン	±			
xxxx1110	(7)	.	>	N	^	n	+			ヨ	セ	ホ	・	ñ			
xxxx1111	(8)	/	?	O	_	o	€			弾	ウ	ソ	¶	ö	¶		

10

El LCD alfanumérico HD44780

- Tabla de caracteres ASCII de 7 bits

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0 000	NUL	(null)	32	20 040	0#32;	Space	64	40 100	0#64;	Ø	96	60 140	0#96;	~			
1	1 001	SOH	(start of heading)	33	21 041	0#33;	!	65	41 101	0#65;	A	97	61 141	0#97;	a			
2	2 002	STX	(start of text)	34	22 042	0#34;	"	66	42 102	0#66;	B	98	62 142	0#98;	b			
3	3 003	ETX	(end of text)	35	23 043	0#35;	#	67	43 103	0#67;	C	99	63 143	0#99;	c			
4	4 004	EOT	(end of transmission)	36	24 044	0#36;	\$	68	44 104	0#68;	D	100	64 144	0#100;	d			
5	5 005	ENQ	(enquiry)	37	25 045	0#37;	%	69	45 105	0#69;	E	101	65 145	0#101;	e			
6	6 006	ACK	(acknowledge)	38	26 046	0#38;	&	70	46 106	0#70;	F	102	66 146	0#102;	f			
7	7 007	BEL	(bell)	39	27 047	0#39;	'	71	47 107	0#71;	G	103	67 147	0#103;	g			
8	8 010	BS	(backspace)	40	28 050	0#40;	(72	48 110	0#72;	H	104	68 150	0#104;	h			
9	9 011	TAB	(horizontal tab)	41	29 051	0#41;)	73	49 111	0#73;	I	105	69 151	0#105;	i			
10	A 012	LF	(NL line feed, new line)	42	2A 052	0#42;	*	74	4A 112	0#74;	J	106	6A 152	0#106;	j			
11	B 013	VT	(vertical tab)	43	2B 053	0#43;	+	75	4B 113	0#75;	K	107	6B 153	0#107;	k			
12	C 014	FF	(NP form feed, new page)	44	2C 054	0#44;	,	76	4C 114	0#76;	L	108	6C 154	0#108;	l			
13	D 015	CR	(carriage return)	45	2D 055	0#45;	-	77	4D 115	0#77;	M	109	6D 155	0#109;	m			
14	E 016	SO	(shift out)	46	2E 056	0#46;	.	78	4E 116	0#78;	N	110	6E 156	0#110;	n			
15	F 017	SI	(shift in)	47	2F 057	0#47;	/	79	4F 117	0#79;	O	111	6F 157	0#111;	o			
16	10 020	DLE	(data link escape)	48	30 060	0#48;	0	80	50 120	0#80;	P	112	70 160	0#112;	p			
17	11 021	DC1	(device control 1)	49	31 061	0#49;	1	81	51 121	0#81;	Q	113	71 161	0#113;	q			
18	12 022	DC2	(device control 2)	50	32 062	0#50;	2	82	52 122	0#82;	R	114	72 162	0#114;	r			
19	13 023	DC3	(device control 3)	51	33 063	0#51;	3	83	53 123	0#83;	S	115	73 163	0#115;	s			
20	14 024	DC4	(device control 4)	52	34 064	0#52;	4	84	54 124	0#84;	T	116	74 164	0#116;	t			
21	15 025	NAK	(negative acknowledge)	53	35 065	0#53;	5	85	55 125	0#85;	U	117	75 165	0#117;	u			
22	16 026	SYN	(synchronous idle)	54	36 066	0#54;	6	86	56 126	0#86;	V	118	76 166	0#118;	v			
23	17 027	ETB	(end of trans. block)	55	37 067	0#55;	7	87	57 127	0#87;	W	119	77 167	0#119;	w			
24	18 030	CAN	(cancel)	56	38 070	0#56;	8	88	58 130	0#88;	X	120	78 170	0#120;	x			
25	19 031	EM	(end of medium)	57	39 071	0#57;	9	89	59 131	0#89;	Y	121	79 171	0#121;	y			
26	1A 032	SUB	(substitute)	58	3A 072	0#58;	:	90	5A 132	0#90;	Z	122	7A 172	0#122;	z			
27	1B 033	ESC	(escape)	59	3B 073	0#59;	:	91	5B 133	0#91;	[123	7B 173	0#123;	{			
28	1C 034	FS	(file separator)	60	3C 074	0#60;	<	92	5C 134	0#92;	\	124	7C 174	0#124;				
29	1D 035	GS	(group separator)	61	3D 075	0#61;	=	93	5D 135	0#93;]	125	7D 175	0#125;)			
30	1E 036	RS	(record separator)	62	3E 076	0#62;	>	94	5E 136	0#94;	^	126	7E 176	0#126;	~			
31	1F 037	US	(unit separator)	63	3F 077	0#63;	?	95	5F 137	0#95;	_	127	7F 177	0#127;	DEL			

Source: www.LookupTables.com

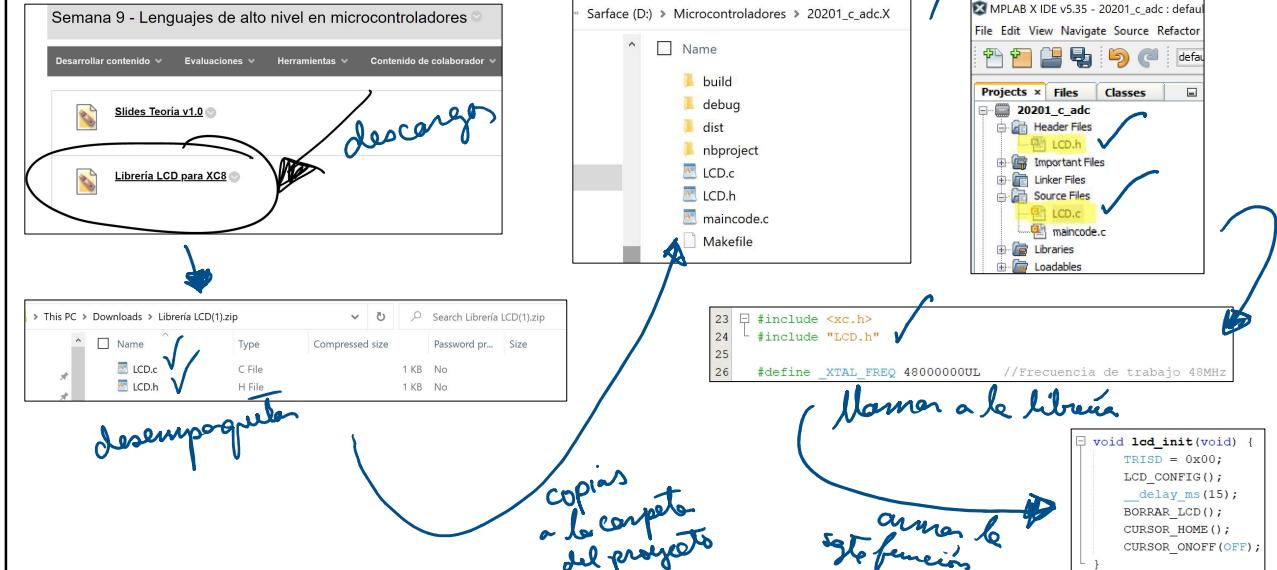
11

El LCD alfanumérico HD44780

- Referencia: Hoja técnica del HD44780
 - http://academy.cba.mit.edu/classes/output_devices/44780.pdf
- Para trabajar con el display se ha creado una librería de comandos (desarrollado por el profesor Sergio Salas) en la cual posee las siguientes características:
 - Interface de 4 bits
 - Comandos para: Limpiar pantalla, ocultar cursor, pasar de línea, caracteres personalizados, etc.
 - Puerto D empleado
 - Tener en cuenta FOSC especificado dentro de la librería (48MHz)

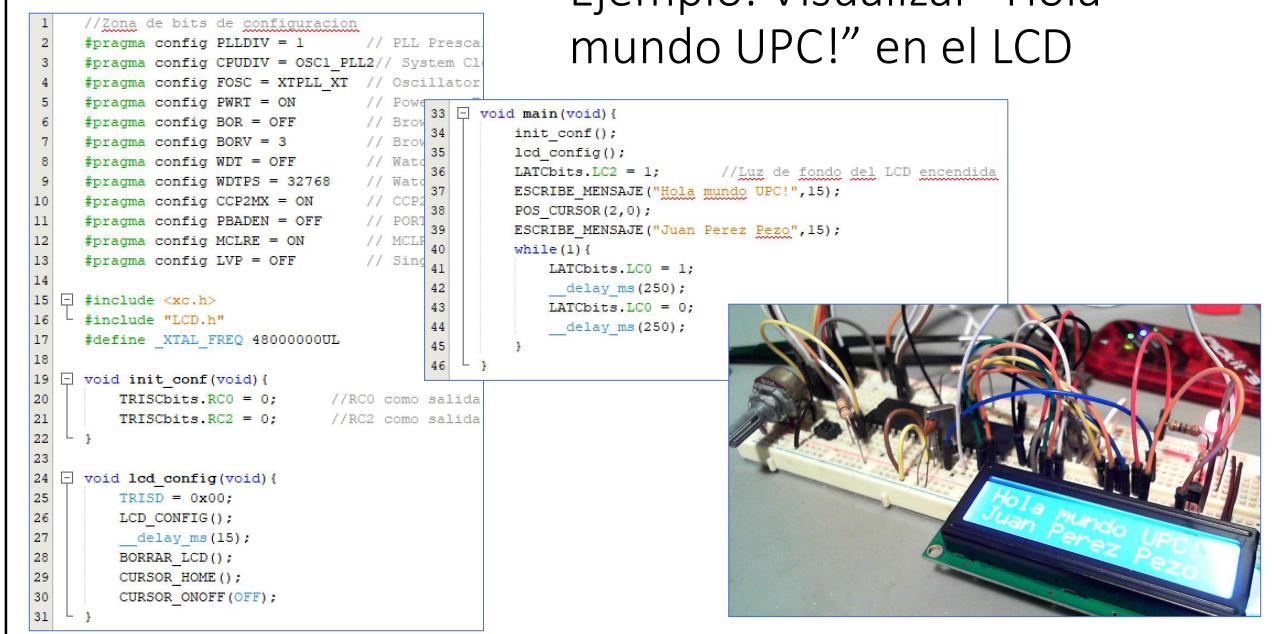
12

Uso del LCD en XC8 (librería S_SAL)



13

Ejemplo: Visualizar "Hola mundo UPC!" en el LCD



14

Código ejemplo para configurar y leer AN0 en el A/D:

```

28     unsigned int res_ad = 0;
29
30     void configuracion(void) {
31         //Aqui colocas las configuraciones iniciales
32         ADCON2 = 0xA4;           //ADFM=0 (just derecha), 8TAD, Fosc/4
33         ADCON1 = 0x0E;          //Canal AN0 habilitado
34         ADCON0 = 0x01;          //Canal AN0 seleccionado y encendemos el A/D
35
36     void main(void) {
37         configuracion();
38
39         while (1) {
40             //Tu programa de usuario
41             ADCON0bits.GODONE = 1;           //Inicio una captura de muestra en AN0
42             while(ADCON0bits.GODONE == 1);   //Espero a que termine de convertir
43             res_ad = (ADRESH << 8) + ADRESL; //Grabar el resultado en la variable res_ad
44             convierte(res_ad);            //Sacar los dígitos
45
46         }
47
48     }
49
50     void convierte(unsigned int numero) {
51         millar = numero /1000;
52         centena = (numero % 1000) / 100;
53         decena = (numero % 100) / 10;
54         unidad = numero % 10;
55
56     }
57
58     void lcd_init() {
59
60     }
61
62     void main(void) {
63         configuracion();
64         ESCRIBE_MENSAJE("VIRTUAlASO",10);
65         while (1) {
66             //Tu programa de usuario
67             ADCON0bits.GODONE = 1;
68             while(ADCON0bits.GODONE == 1);
69             res_ad = (ADRESH << 8) + ADRESL;
70             convierte(res_ad);
71             POS_CURSOR(2,0);
72             ENVIA_CHAR(millar+0x30);
73             ENVIA_CHAR(centena+0x30);
74             ENVIA_CHAR(decena+0x30);
75             ENVIA_CHAR(unidad+0x30);
76         }
77     }
78
79 
```

15

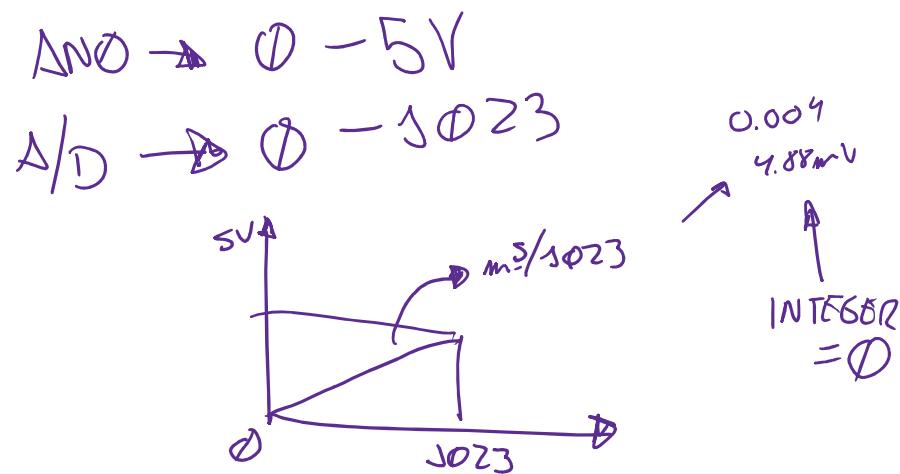
Código en XC8 del ejemplo inicial:

```

26 #define _XTAL_FREQ 48000000UL //Frecu
27
28 unsigned int res_ad = 0;
29 unsigned int millar = 0;
30 unsigned int centena = 0;
31 unsigned int decena = 0;
32 unsigned int unidad = 0;
33
34 void convierte(unsigned int numero){
35     millar = numero /1000;
36     centena = (numero % 1000) / 100;
37     decena = (numero % 100) / 10;
38     unidad = numero % 10;
39 }
40
41 void lcd_init(void) {
42     TRISD = 0x00;           //Puerto D
43     LCD_CONFIG();
44     __delay_ms(15);
45     BORRAR_LCD();
46     CURSOR_HOME();
47     CURSOR_ONOFF(OFF);
48 }
49
50 void configuracion(void) {
51     //Aqui colocas las configuraciones
52     ADCON2 = 0xA4;          //ADET
53     ADCON1 = 0x0E;          //Canal A
54     ADCON0 = 0x01;          //Canal A
55     lcd_init();
56 }
57
58 void main(void) {
59     configuracion();
60     ESCRIBE_MENSAJE("VIRTUAlASO",10);
61     while (1) {
62         //Tu programa de usuario
63         ADCON0bits.GODONE = 1;
64         while(ADCON0bits.GODONE == 1);
65         res_ad = (ADRESH << 8) + ADRESL;
66         convierte(res_ad);
67         POS_CURSOR(2,0);
68         ENVIA_CHAR(millar+0x30);
69         ENVIA_CHAR(centena+0x30);
70         ENVIA_CHAR(decena+0x30);
71         ENVIA_CHAR(unidad+0x30);
72     }
73 }
```

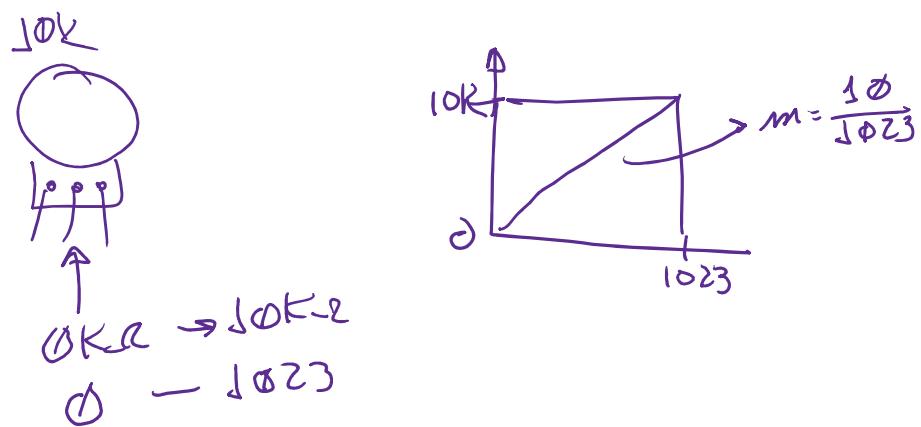
16

Proceso de escalamiento



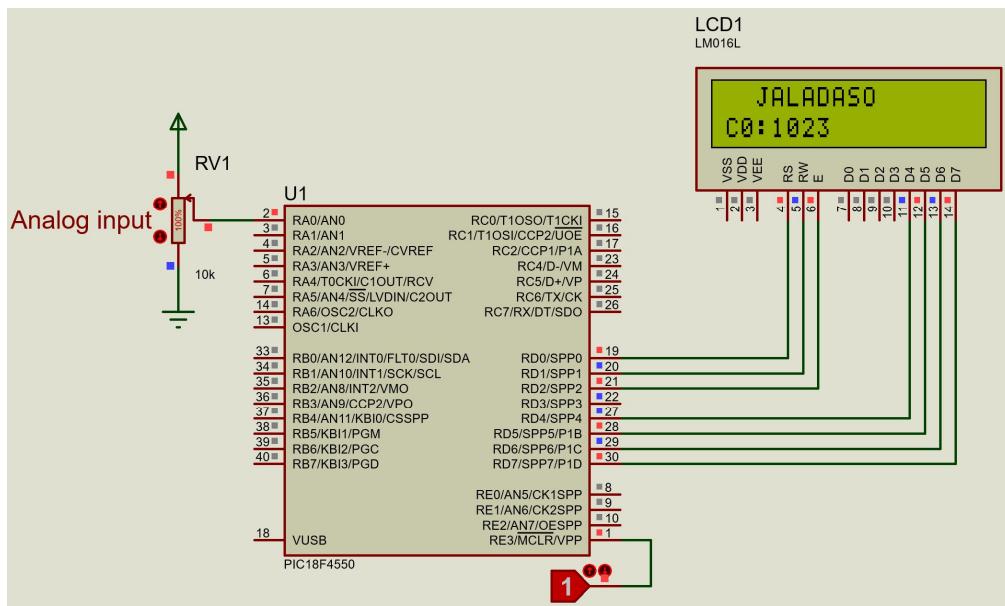
17

Proceso de escalamiento (valor del pot)



18

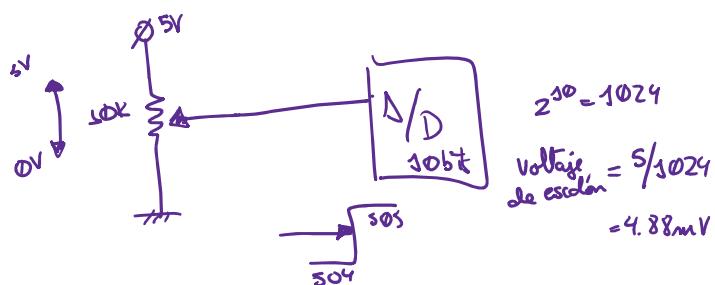
Simulación en Proteus:



19

Observaciones:

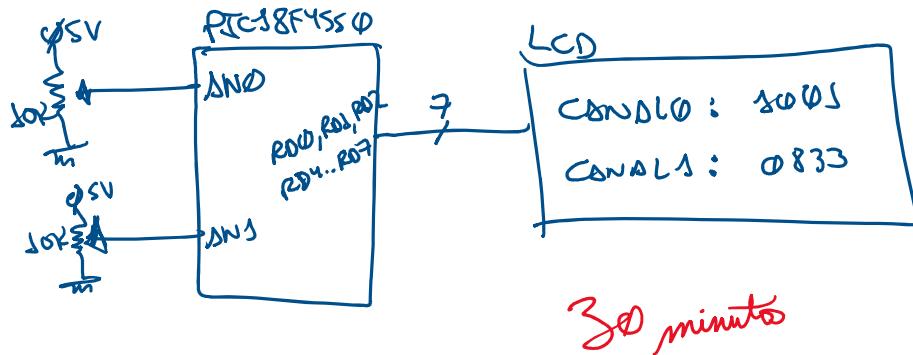
- El último dígito que se muestra (de las cuentas de 10 bits del resultado del A/D) varía bastante:



20

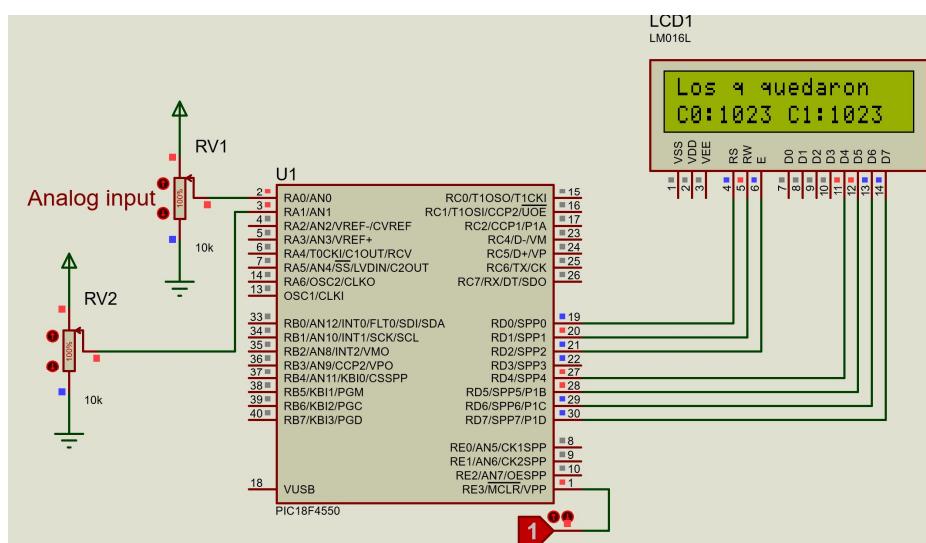
Asignación:

- Leer dos canales analógicos y mostrarlos en el LCD



21

Simulación:



22

Código en XC8:

```

27  void configuracion(void) {
28      //Aquí colocas las configuraciones inic
29      ADCON2 = 0xA4;           //ADFM=0 (
30      ADCON1 = 0x0D;          //Canal AN0
31      ADCONbits.ADON = 1;     //Encendemos
32      lcd_init();
33
34      unsigned int res_ad0 = 0;
35      unsigned int res_ad1 = 0;
36
37      unsigned int millar = 0;
38      unsigned int centena = 0;
39      void convierte(unsigned int numero){
40          millar = numero /1000;
41          centena = (numero % 1000) / 100;
42          decena = (numero % 100) / 10;
43          unidad = numero % 10;
44      }
45
46      void lcd_init(void) {
47          TRISD = 0x00;           //Pue
48          LCD_CONFIG();
49          __delay_ms(15);
50          BORRAR_LCD();
51          CURSOR_HOME();
52          CURSOR_ONOFF(OFF);
53      }
54
55  }
56
57  //Aquí colocas las configuraciones inic
58  ADCON2 = 0xA4;           //ADFM=0 (
59  ADCON1 = 0x0D;          //Canal AN0
60  ADCONbits.ADON = 1;     //Encendemos
61  lcd_init();
62
63  void main(void) {
64      configuracion();        //Llamada a
65      ESCRIBE_MENSAJE("Los q quedaron",14);
66      while(1){
67          ADCON0 = 0x03;
68          //ADCON0bits.GODONE = 1;
69          while(ADCON0bits.GODONE == 1);    //
70          res_ad0 = (ADRESH << 8) + ADRESL;
71          ADCON0 = 0x07;                  //
72          while(ADCON0bits.GODONE == 1);    //
73          res_ad1 = (ADRESH << 8) + ADRESL;
74          POS_CURSOR(2,0);              //S
75          ESCRIBE_MENSAJE("CO:",3);
76          convierte(res_ad0);          //
77          ENVIA_CHAR(millar+0x30);    //E
78          ENVIA_CHAR(centena+0x30);   //E
79          ENVIA_CHAR(decena+0x30);    //E
80          ENVIA_CHAR(unidad+0x30);    //E
81          ESCRIBE_MENSAJE(" Cl:",4);
82          convierte(res_ad1);          //
83          ENVIA_CHAR(millar+0x30);    //E
84          ENVIA_CHAR(centena+0x30);   //E
85          ENVIA_CHAR(decena+0x30);    //E
86          ENVIA_CHAR(unidad+0x30);    //E
87      }
88  }

```

23

Fin de la sesión!

24