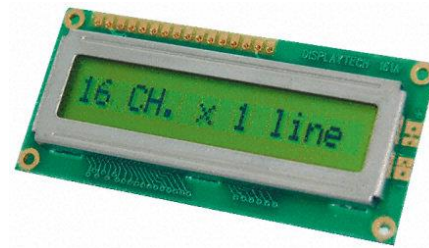
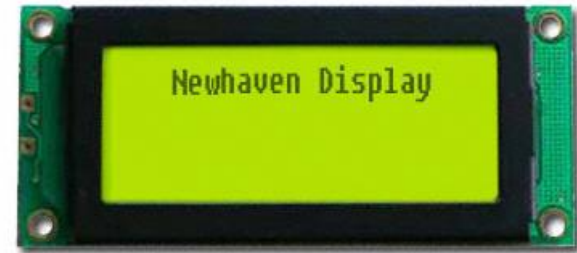
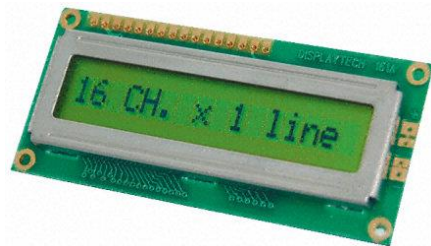


LCD (liquid crystal display)



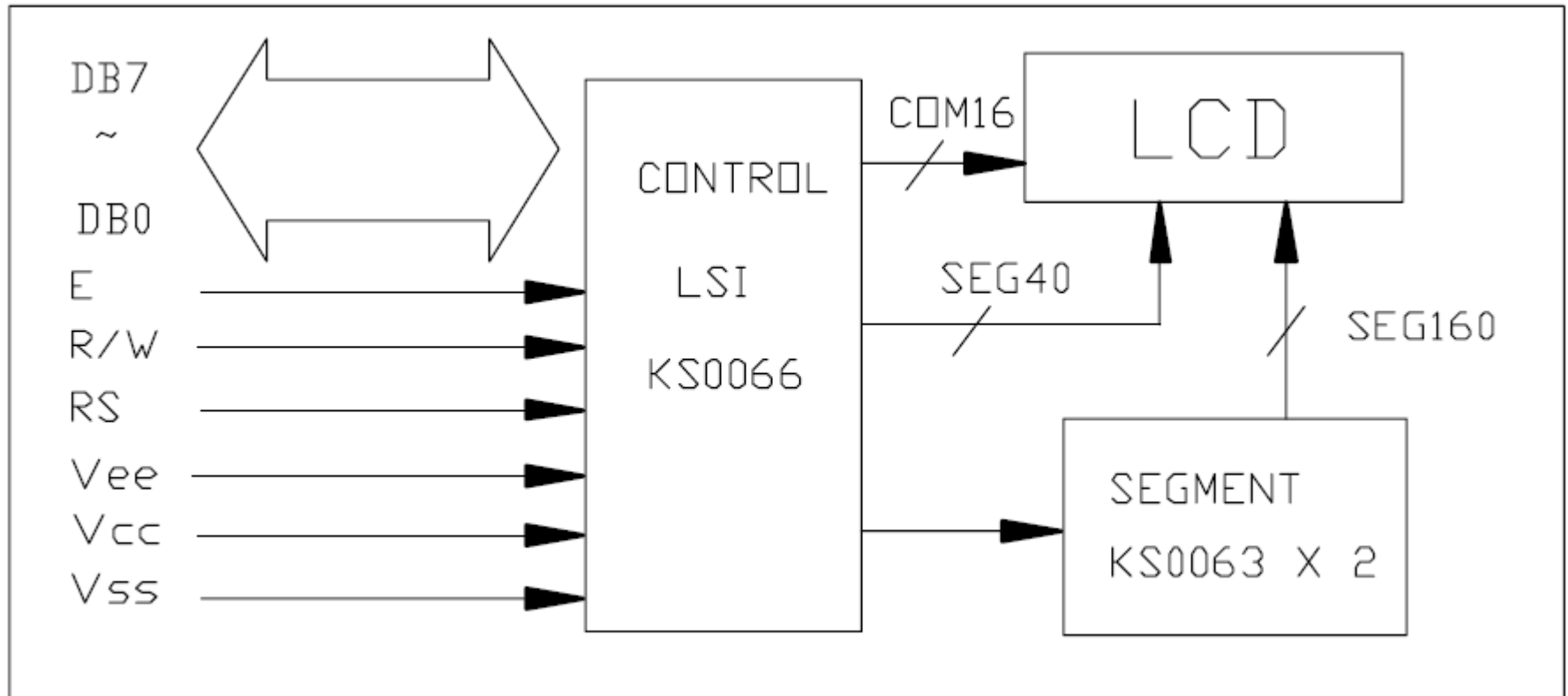
LCD

- LCD quiere decir “liquid crystal display”
- La mayoría de los LCDs alfanuméricos tienen una luz de fondo (backlight) basada en LED
- Los displays tienen un número de renglones y columnas. Algunos ejemplos: 8x1, 8x2, 16x1, 16x2, 16x4, 20x1, 20x2, 20x4, 40x1, 40x2, y 40x4.



LCD

- Comúnmente los LCDs utilizan un controlador Hitachi (HD44780 o KS0066)



Señales del LCD

Interface Signals

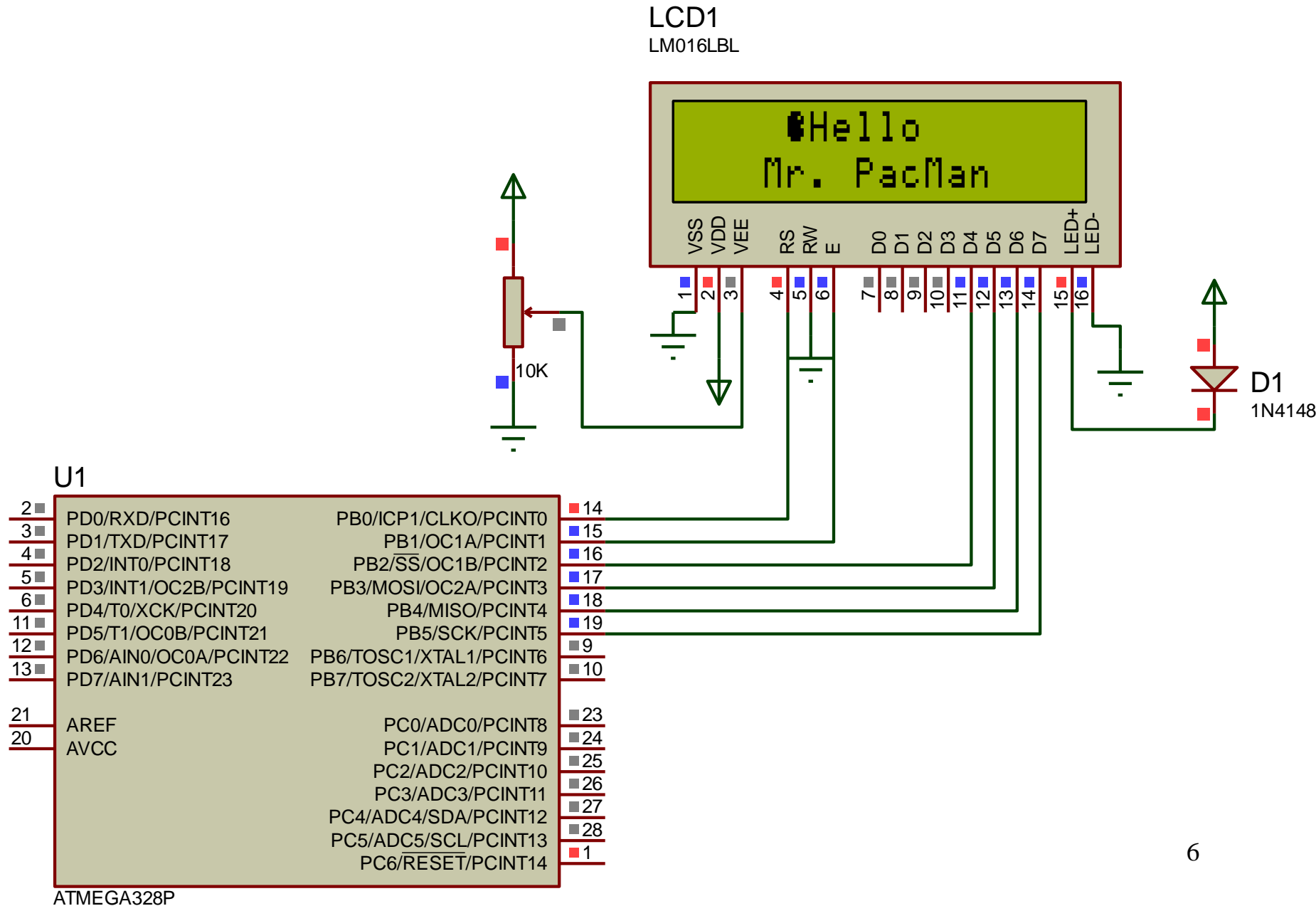
Pin No.	Symbol	Level	Description
1	Vss	0V	Ground
2	Vcc	5.0V	Power supply voltage for logic and LCD(+)
3	Vo	0.3V	Power supply voltage for LCD(-)
4	RS	H/L	Selects registers
5	R/W	H/L	Read/Write Signal
6	E	H/L	Chip enable Signal
7	DB0	H/L	Data bit0
8	DB1	H/L	Data bit1
9	DB2	H/L	Data bit2
10	DB3	H/L	Data bit3
11	DB4	H/L	Data bit4
12	DB5	H/L	Data bit5
13	DB6	H/L	Data bit6
14	DB7	H/L	Data bit7
15	LED+		+4.3V Anode
16	LED-		GND Cathode

LED Backlight will work on 5VDC if through a 1N4001 Diode.
This gives approximately 4.3VDC

Mapa de Caracteres

CHARACTER FONT TABLE														
LOWER 4 BITS UPPER 4 BITS	0000	0010	0011	0100	0101	0110	0111	1010	1011	1100	1101	1110	1111	
0000	CG RAM (1)		0	@	P	`	F		-	タ	ミ	α	p	
0001	(2)	!	1	A	Q	a	q	。	ア	チ	厶	ä	q	
0010	(3)	"	2	B	R	b	r	「	イ	ツ	×	β	θ	
0011	(4)	#	3	C	S	c	s	」	ウ	テ	ε	ε	∞	
0100	(5)	\$	4	D	T	d	t	、	エ	ト	ト	μ	Ω	
0101	(6)	%	5	E	U	e	u	・	オ	ナ	1	℃	ü	
0110	(7)	&	6	F	V	f	v	ヲ	カ	ニ	ヨ	ρ	Σ	
0111	(8)	'	7	G	W	g	w	ア	キ	ヌ	ラ	g	π	
1000	(1)	<	8	H	X	h	x	イ	ク	ネ	リ	フ	̄	
1001	(2)	>	9	I	Y	i	y	ウ	ケ	ル	ル	、	y	
1010	(3)	*	:	J	Z	j	z	エ	コ	ハ	レ	j	チ	
1011	(4)	+	;	K	[k	[オ	サ	ヒ	ロ	*	斤	
1100	(5)	、	<	L	¥	l	l	ヤ	シ	フ	ワ	¢	円	
1101	(6)	-	=	M]	m]	ユ	ズ	ハ	ン	も	÷	
1110	(7)	.	>	N	^	n	→	ヨ	セ	ホ	ッ	ñ		
1111	(8)	/	?	O	_	o	←	ッ	リ	マ	°	ö	■	

Conexión LCD



Uso de librería display.h

- Se debe bajar la librería de siquiros y copiar en el directorio /inc de Codevision
- Al inicio del programa antes de los #include poner:

```
#asm
.equ __lcd_port=0x05
.equ __lcd_EN=1
.equ __lcd_RS=0
.equ __lcd_D4=2
.equ __lcd_D5=3
.equ __lcd_D6=4
.equ __lcd_D7=5
#endasm

#include <io.h>
#include <delay.h>
#include <display.h>
```

Puerto (ATmega328P)	lcd_port
PORTB	0x05
PORTC	0x08
PORTD	0x0B

Todos los pines del display se tienen que conectar al mismo puerto (B,C o D)

Procedimientos de display.h

- Antes de cualquier acceso a algún procedimiento del LCD se tiene que iniciar con la siguiente instrucción:

```
SetupLCD ( ) ;
```

- Se pueden crear hasta 8 caracteres especiales. Para crear uno se debe poner un arreglo con la información del dibujo:

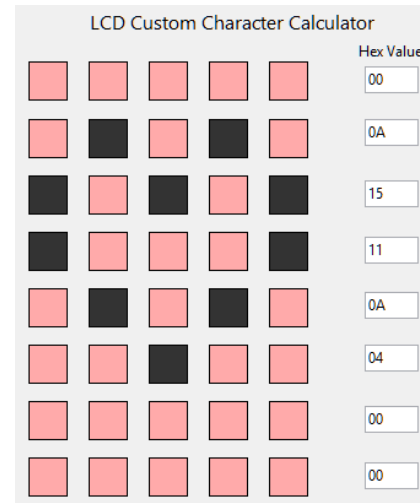
```
unsigned char corazon[8]={0x00,0x0A,0x15,0x11,0x0A,0x04,  
0x00,0x00} ;
```

Al inicio del programa se debe programar en la RAM del display el caracter propio:

```
CreateChar (0 , corazon) ;
```

Para desplegar en el LCD se pondría:

```
CharLCD (0) ;
```



Procedimientos de display.h

- El display siempre escribirá caracteres en donde se encuentre el cursor (aunque sea invisible). Para mover el cursor se utiliza el procedimiento:

MoveCursor (x, y) ;

Donde para un LCD de 16 caracteres por 2 los rangos serían:

x: 0 a 15

y: 0 o 1

- Para escribir una letra en el LCD se puede usar:

CharLCD ('H') ; //Escribe una letra H en el LCD

CharLCD (0) ; //Escribe el caracter especial 0

CharLCD (1) ; //Escribe el caracter especial 1

Procedimientos de display.h

- Para escribir un string fijo (que no cambia durante la ejecución) se usa:

StringLCD ("HOLA") ;

El string se escribirá donde esté el cursor.

- Para escribir un string fijo con un retardo entre letra y letra (como máquina de escribir) se usa:

StringLCD2 ("MUNDO", 200) ;

Donde el primer parámetro es el string y el segundo el tiempo entre letras en milisegundos.

- Para escribir un string variable (que cambia durante la ejecución) se usa:

StringLCDVar (Cadena) ;

Donde el argumento es un string variable que se puede cambiar durante la ejecución.

Procedimientos de display.h

- Para borrar el LCD se usa:

EraseLCD () ;

Se borrará el LCD y el cursor se moverá a la posición inicial (0,0)

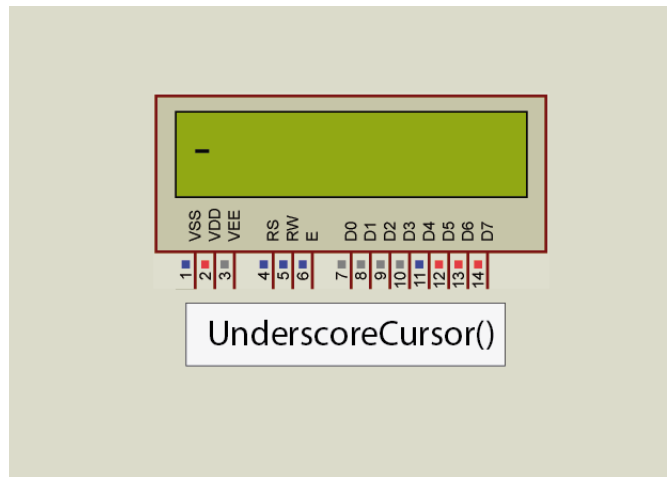
- Por default el cursor es invisible. Sin embargo se puede cambiar con las siguientes opciones:

UnderscoreCursor () .- pone el cursor “underscore”

NoUnderscoreCursor () .- quita el cursor “underscore”

BlinkCursor () .- pone el cursor intermitente

NoBlinkCursor () .- quita el cursor intermitente



1. Desarrollar un programa que permita tener una animación de 2 pantallas en el LCD. El tiempo entre pantallas deberá ser de 2 segundos y se utilizará al menos la generación de un carácter especial (utilizar el proyecto demo)

2. Hacer un programa que despliegue el número de segundos que han transcurrido desde que se inició la aplicación.

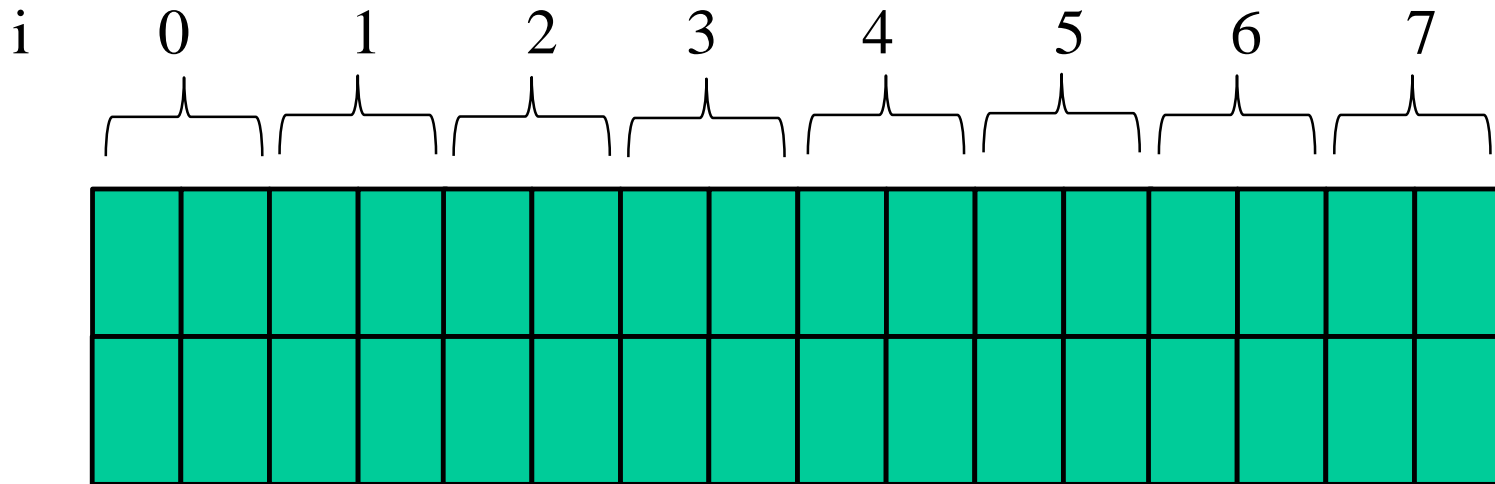
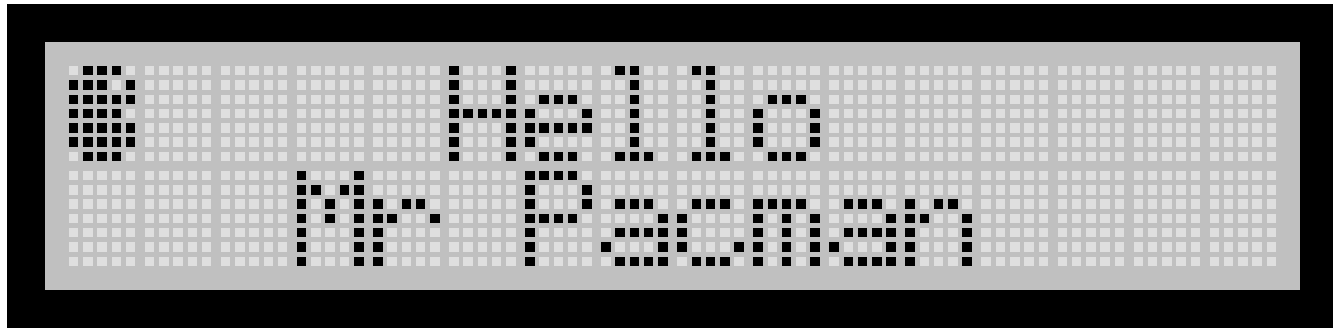
Si la variable seg es:

- unsigned char seg (rango 0 a 255seg) ~4 min reiniciaría
- unsigned short seg (rango 0 a 65535seg) ~18h reiniciaría
- unsigned long seg (rango 0 a $2^{32}-1$) ~136 años se reiniciaría

3. Se deberá desarrollar un cronómetro por medio del LCD y el microcontrolador ATMega328P. El formato del cronómetro será “MM:SS:D” donde MM, SS y D denotan minutos, segundos y décimas de segundo respectivamente. Se deberán tener dos botones para control del cronómetro, uno de “start/stop” y el otro de “reset”.
4. Agregar al problema 3 una animación con un carácter especial que cuando el cronómetro este andando se mueva y permanezca quieto cuando este en “stop” (Tarea Individual) .



5. Hacer un programa donde se ponga un mensaje en 2 renglones y luego un PACMAN se los coma.



Pasos a seguir para cada par (i=0 a 7)

Cursor seleccionado: Underscore para visualizar el algoritmo

Paso 1



Poner cursor en primer elemento del par

Paso 2



Imprimir 1er Pacman

Retardo

Paso 3



Poner cursor en primer elemento del par

Paso 4



Imprimir espacio (borrar el 1er Pacman)

Paso 5



Imprimir 2o Pacman

Retardo

Paso 6



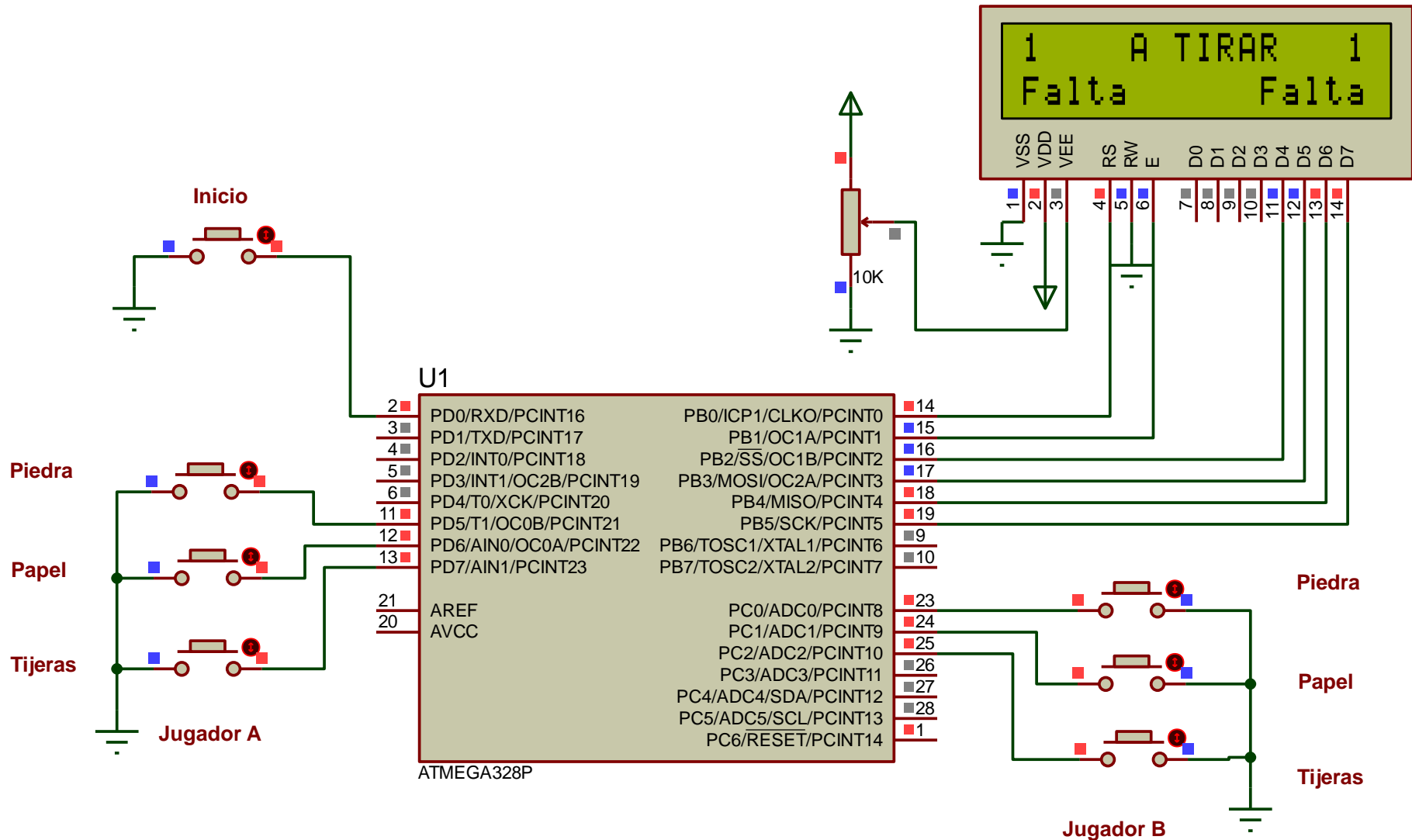
Poner cursor en segundo elemento del par

Paso 7



Imprimir espacio (borrar el 2o Pacman)

6. Hacer un programa que permita a usuarios A y B jugar *Piedra, Papel o Tijeras*



Bibliografía

Hoja de datos de HANDTRONIX LCD