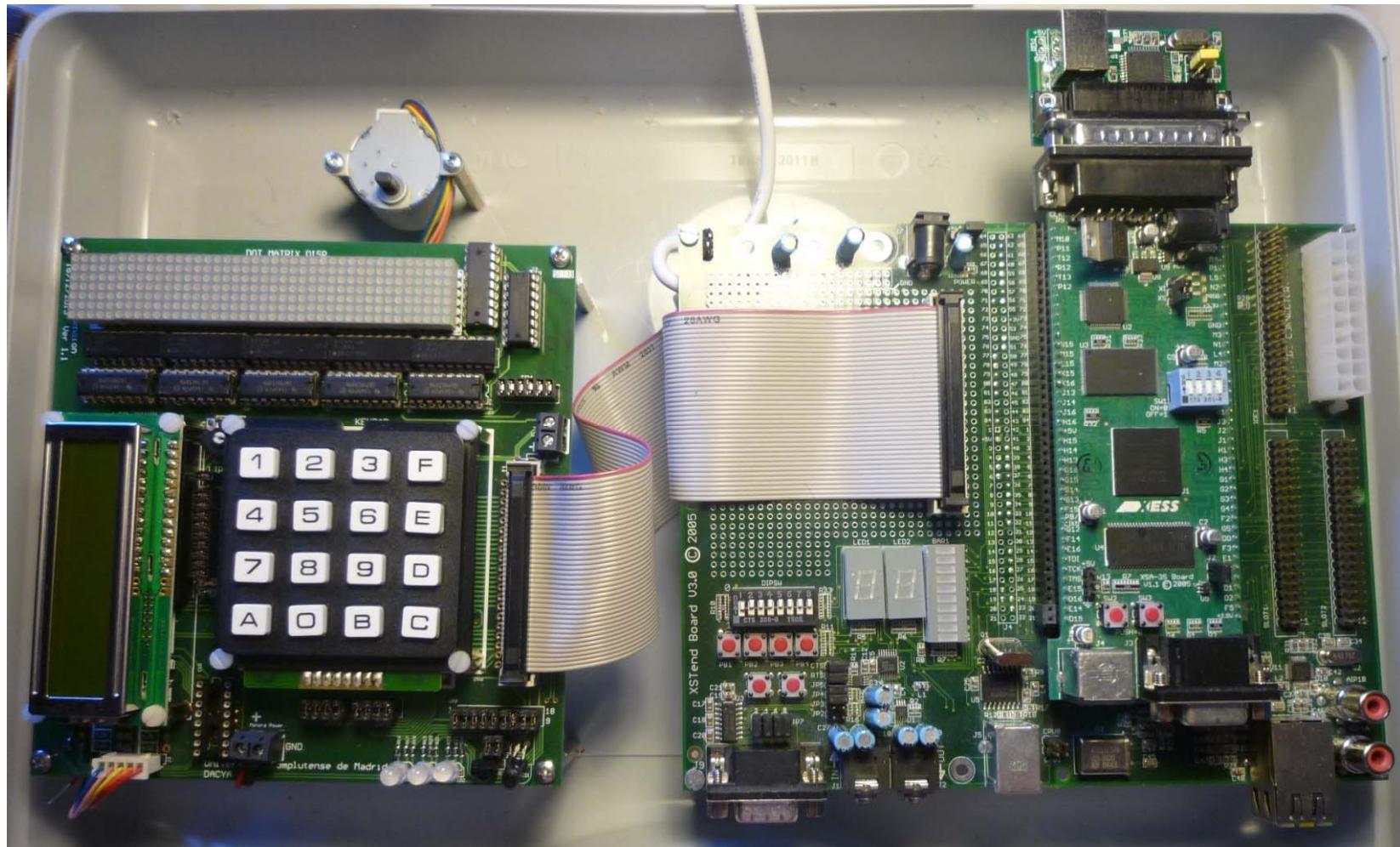
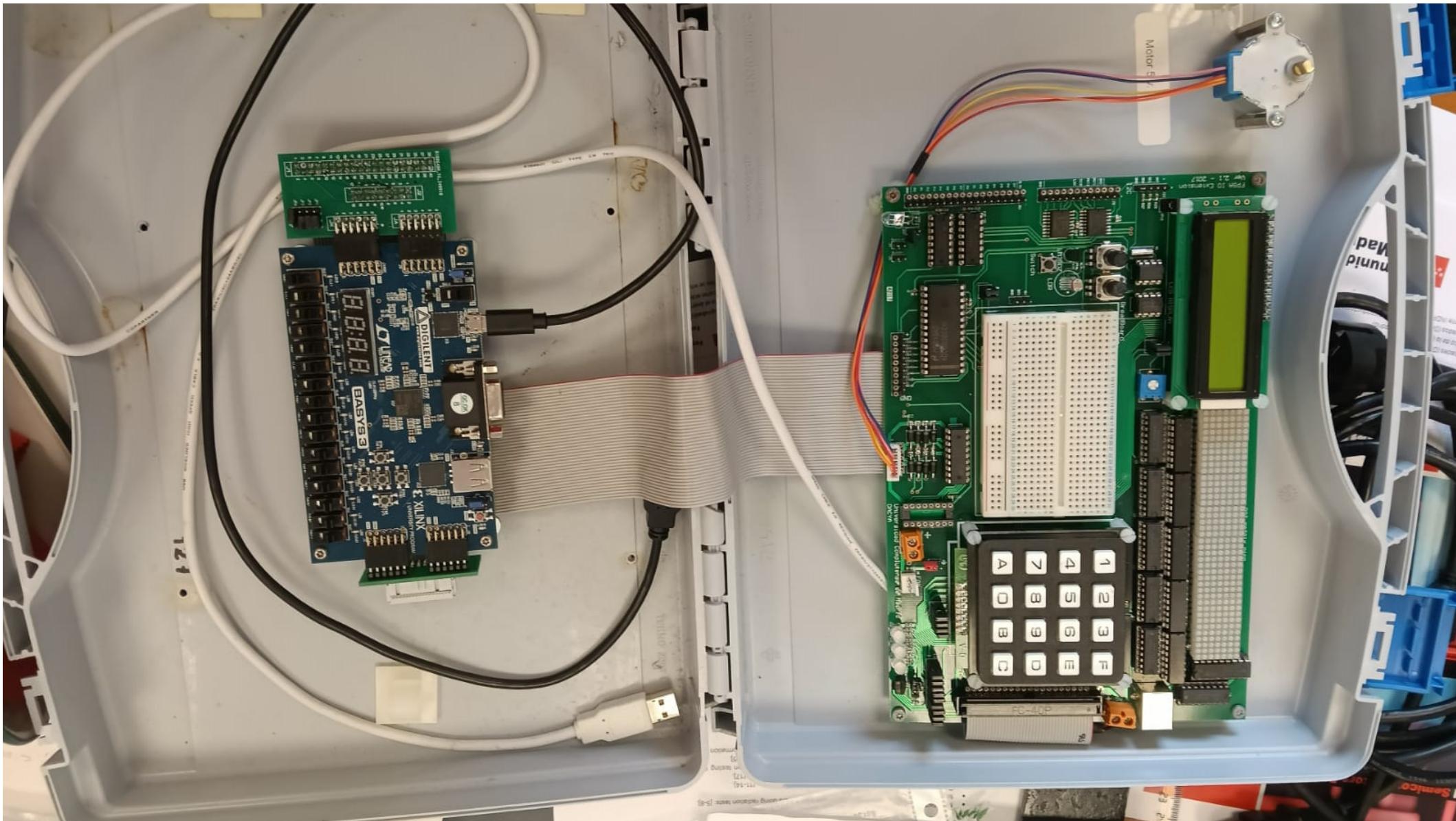




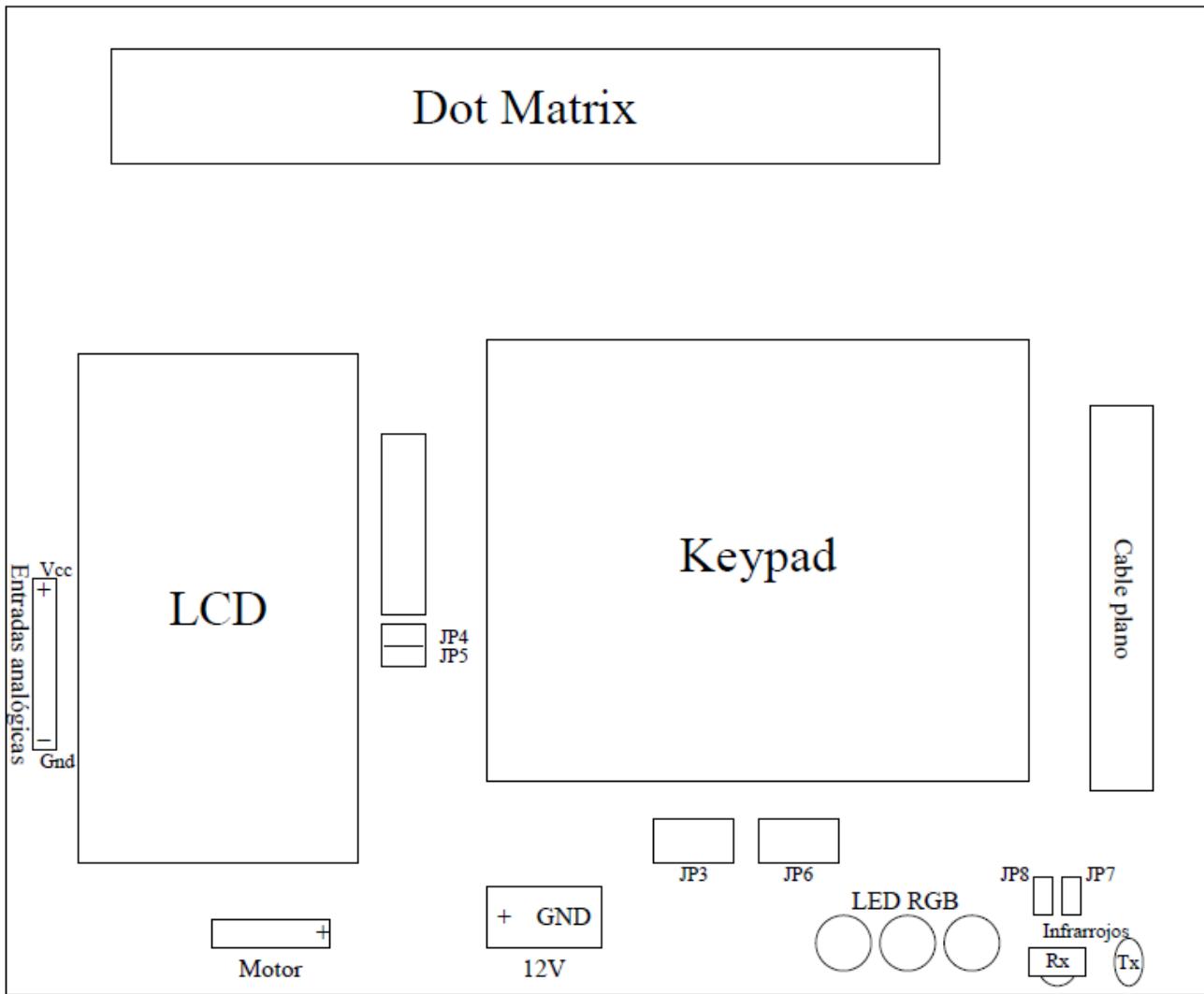
# Placa de expansión







# Placa de expansión





# LCD

- Un **LCD** (Liquid Crystal Display) es un dispositivo de bajo coste capaz de mostrar texto y/o gráficos. Incorpora:
  - el propio **display** de cristal líquido.
  - una **memoria de patrones** que almacena el mapa de bits de los caracteres imprimibles.
  - una **memoria de refresco**.
  - un **microcontrolador** que simplifica al máximo la circuitería de interfaz y que genera las señales eléctricas necesarias para el refresco de la información.
- Un **LCD de matriz de puntos alfanumérico** típico:
  - está formado por 2 filas de 8 caracteres cada una.
  - cada carácter se muestra sobre una matriz de  $5 \times 8$  puntos.
  - internamente almacena el patrón de 208 caracteres diferentes (seleccionables por su código ASCII).
  - permite la definición por el usuario del patrón de hasta 8 caracteres.
  - tiene facilidades para el manejo de cursor y la inicialización del LCD.
  - utiliza para comunicarse un protocolo tipo strobe con señales:
    - de selección operación (lectura/escritura)
    - de selección de registro (instrucción-estado/datos)
    - paralela de datos (8 bits).



# LCD

## ciclos de acceso

### ■ Memorias de caracteres y refresco

- *CGROM* (Character Generation ROM): almacena los patrones de 5x8 puntos de los 204 caracteres fijos.
- *CGRAM* (Character Generation RAM): almacena los patrones de 5x8 puntos de los 8 caracteres programables.
- *DDRAM* (Display Data RAM): almacena los códigos de los caracteres que se muestran en el LCD (2 filas de 40).

### ■ Registros arquitectónicos:

- *IR* (Instruction Register) - 8 bits: almacena órdenes para el LCD.
- *BS* (Busy Flag) - 1 bit: indica que una orden se está procesando.
- *AR* (Address Counter) - 7 bits: registro que se autoincrementa/decrementa por cada acceso a *DR*, y que contiene direcciones de la DD/CGRAM.
- *DR* (Data Register) - 8 bits: registro intermedio para la transmisión con la DD/CGRAM.

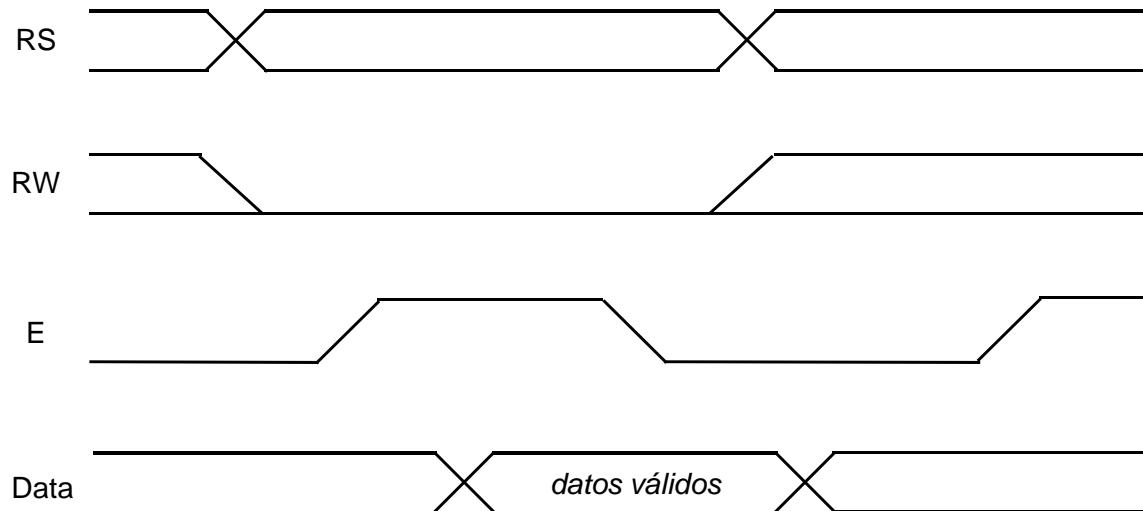
### ■ Líneas de comunicación:

- *Líneas de datos*: D0-D7, para la transferencia bidireccional de datos/instrucciones entre el sistema y el LCD.
- *Línea de control*: RS (Register Select) y RW (Read/Write).
  - RS=0, RW=0: escritura de IR.
  - RS=0, RW=1: lectura de estado (BS & AR).
  - RS=1, RW=X: escritura (RW=0) o lectura (RW=1) de DR.
- *Línea de protocolo*: E (Enable), para indicar la validez de las anteriores líneas (protocolo tipo strobe asíncrono).

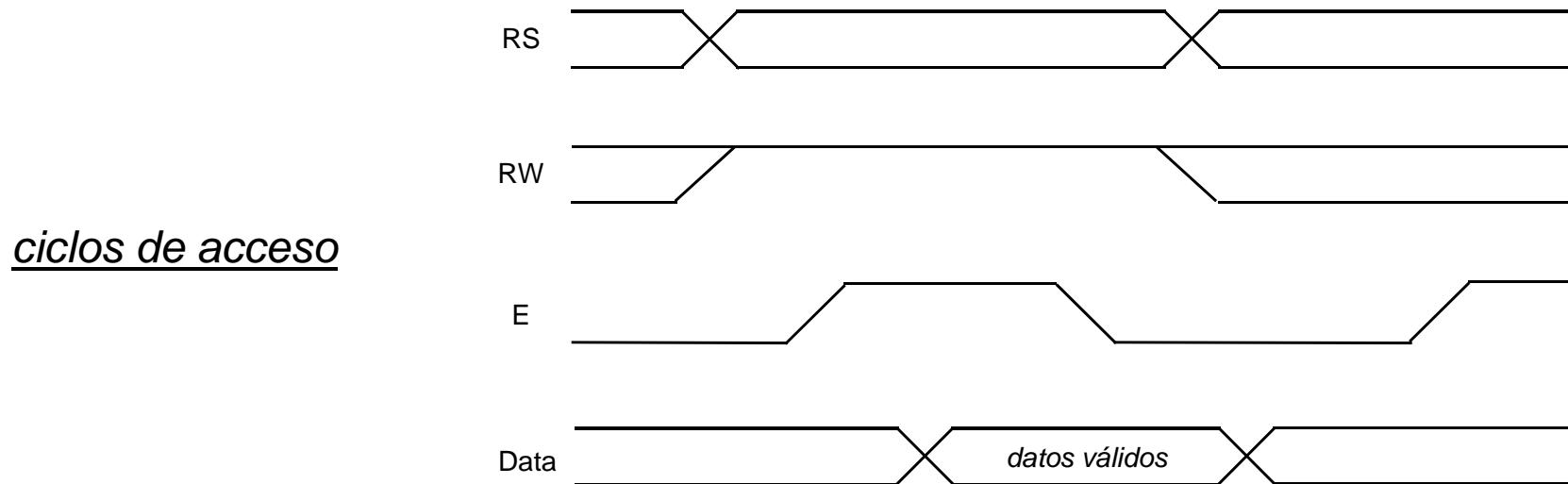


# LCD

## ciclos de acceso



Ciclo de escritura

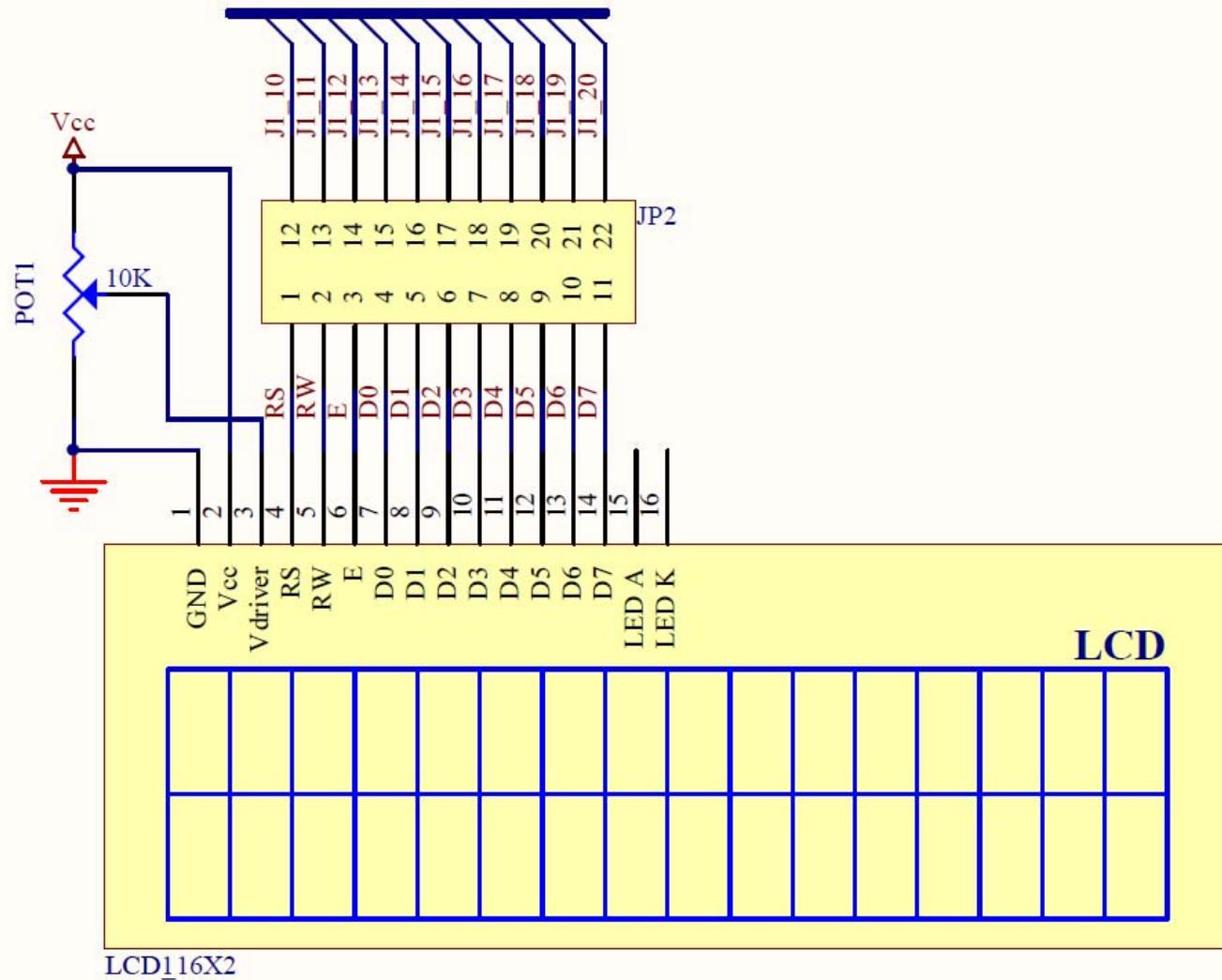


Ciclo de lectura

ciclos de acceso



# LCD

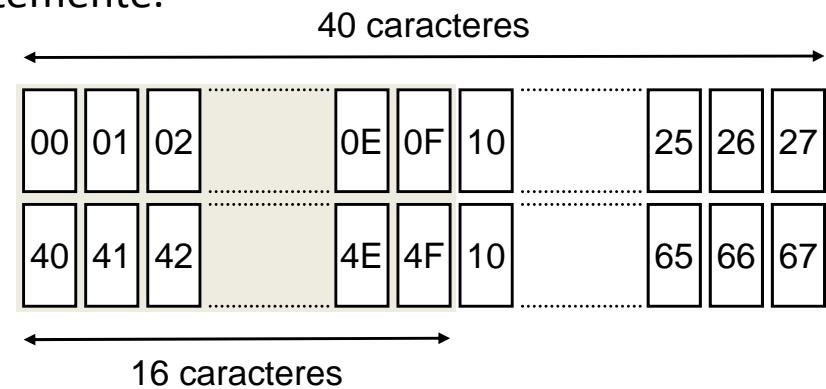




# LCD

## instrucciones

- **Establecimiento del modo de funcionamiento (001dnfxx)**
  - $d=1$ : transmisión de 8 bits,  $d=0$ : transmisión 4 bits
  - $n=1$ : display de 2 líneas,  $n=0$ : display de 1 línea.
  - $f=1$ : caracteres de  $5\times 11$  puntos ,  $f=0$ : caracteres de  $5\times 8$  puntos.
- **Establecimiento del modo de entrada (000001cs)**
  - $s=0$ : a cada escritura del DR, el cursor se mueve hacia la derecha ( $c=1$ ) o a la izquierda ( $c=0$ ). AR se actualiza convenientemente.
  - $s=1$ : a cada escritura del DR, la ventana visible se desplaza a la derecha ( $c=1$ ) o a la izquierda ( $c=0$ ). AR se actualiza convenientemente.



- **Inicialización del display (00000001)**
  - borra el contenido de DDRAM, inicializa AC a 0, retorna el cursor a la esquina superior izquierda, y entra en modo de funcionamiento con cursor y autocincremento de AR.



# LCD

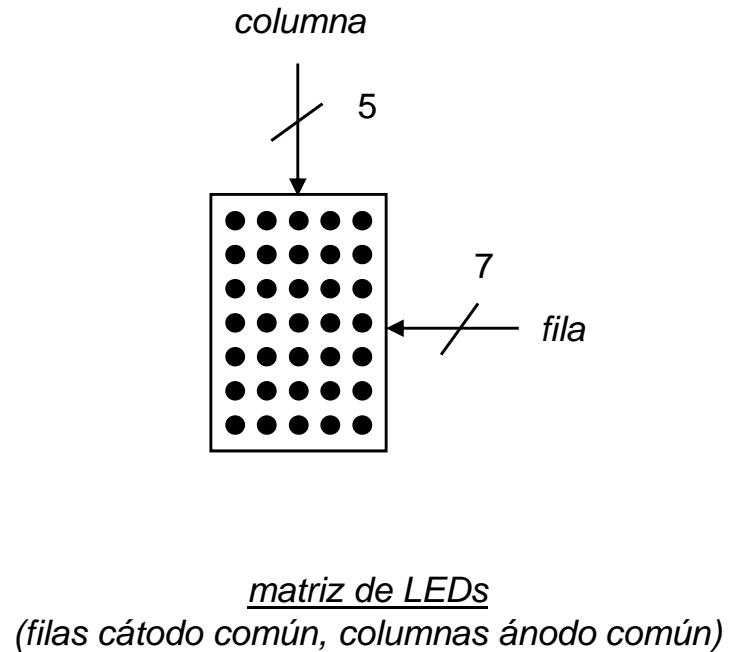
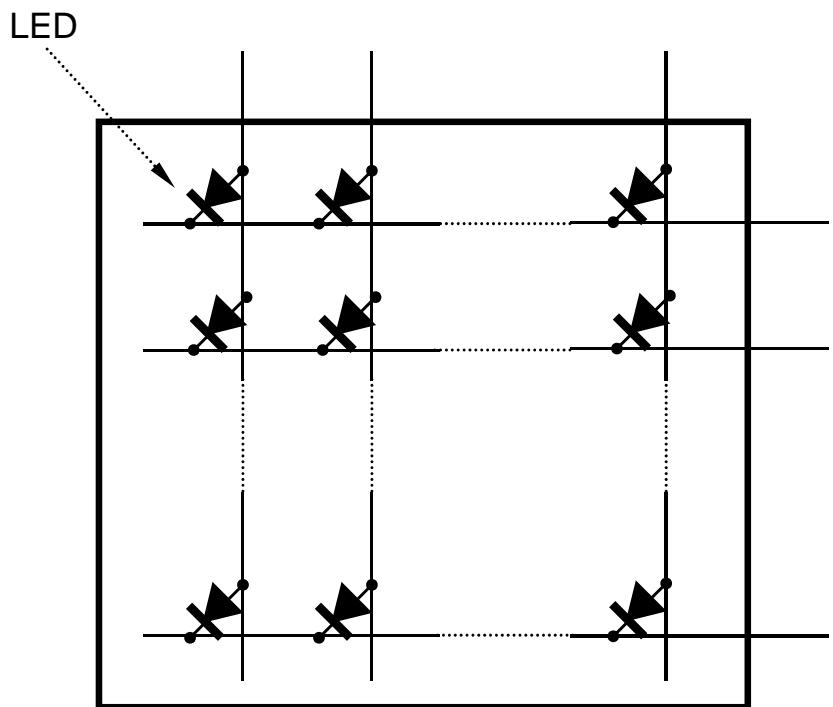
## instrucciones

- **Control de ON/OFF (00001dcb)**
  - $d=1$ : enciende el display,  $d=0$ : apaga el display.
  - $c=1$ : enciende el cursor,  $c=0$ : apaga el cursor.
  - $b=1$ : el cursor parpadea,  $b=0$ : el cursor no parpadea.
- **Fijar dirección de CGRAM (01xxxxxx)**
  - Carga AR con una dirección, e indica que direccione la CGRAM
- **Fijar dirección de DDRAM (1xxxxxxxx)**
  - Carga AR con una dirección, e indica que direccione la DDRAM
- **Desplazamiento (0001csxx)**
  - $s=0$ : desplaza el cursor a la derecha y AR se incrementa ( $c=1$ ) o desplaza el cursor a la izquierda y AR se decrementa ( $c=0$ ).
  - $s=1$ : desplaza la ventana visible a la derecha ( $c=1$ ) o a la izquierda ( $c=0$ ).
- **Vuelta a casa (0000001x)**
  - Manteniendo el contenido de la DDRAM, inicializa AC a 0, retorna el cursor a la esquina superior izquierda, y desplaza la ventana visible a la izquierda.



# Matriz de puntos

- Una **matriz de LEDs** es una colección de LEDs dispuestos en filas y en columnas con acceso multiplexado.
  - todo LED comparte un terminal con todos sus vecinos de fila y el otro terminal con todos sus vecinos de columna.



matriz de LEDs  
(filas cátodo común, columnas ánodo común)

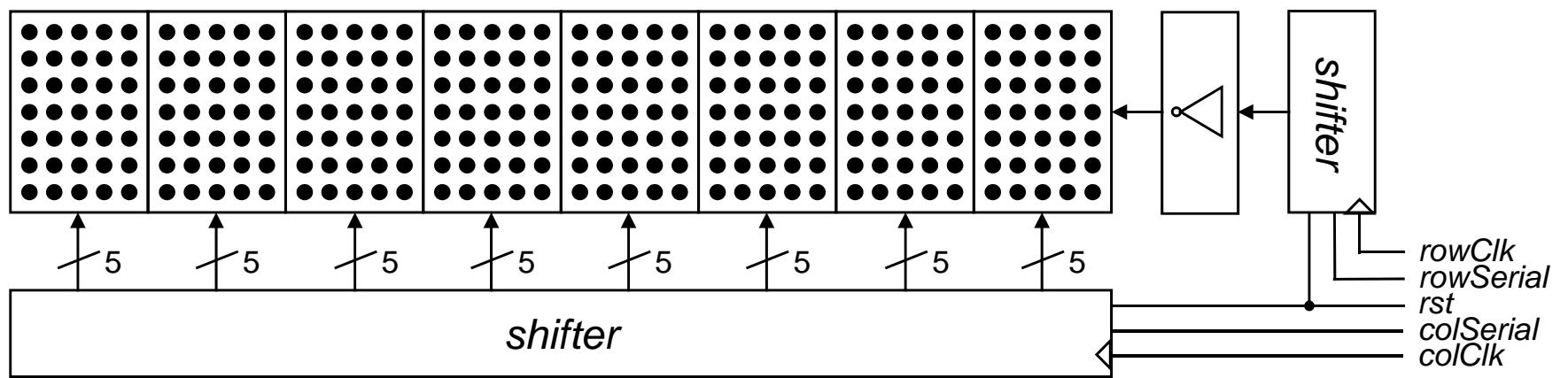
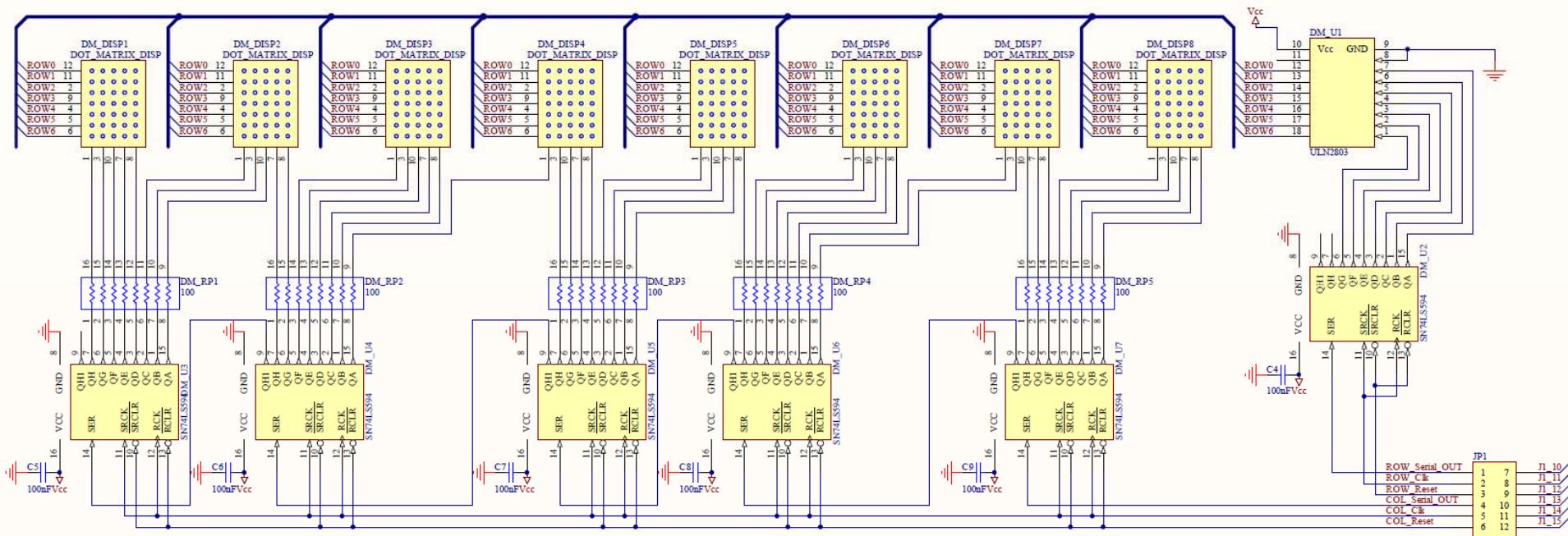


# Matriz de puntos

- Cuando el número de LEDs es elevado, el acceso a los mismos suele multiplexarse para reducir el pinout y el consumo.
- Cuando un banco de LEDs se multiplexa, en todo momento sólo un LED está encendido por lo que la información que muestra debe refrescarse periódicamente.
  - **Frecuencia de refresco:** frecuencia a la que un LED se selecciona
    - el ojo humano detecta como parpadeo frecuencias inferiores a 50 Hz (periodo 20 ms)
  - **Tiempo de persistencia:** tiempo durante el cual un LED está seleccionado
    - conforme disminuye la persistencia, disminuye la luminosidad.
  - En diseño HW, la frecuencia de refresco y la persistencia están relacionadas según el grado de multiplexación.
    - $(\text{tiempo persistencia}) \times (\text{num. elementos a refrescar}) = (\text{periodo de refresco})$



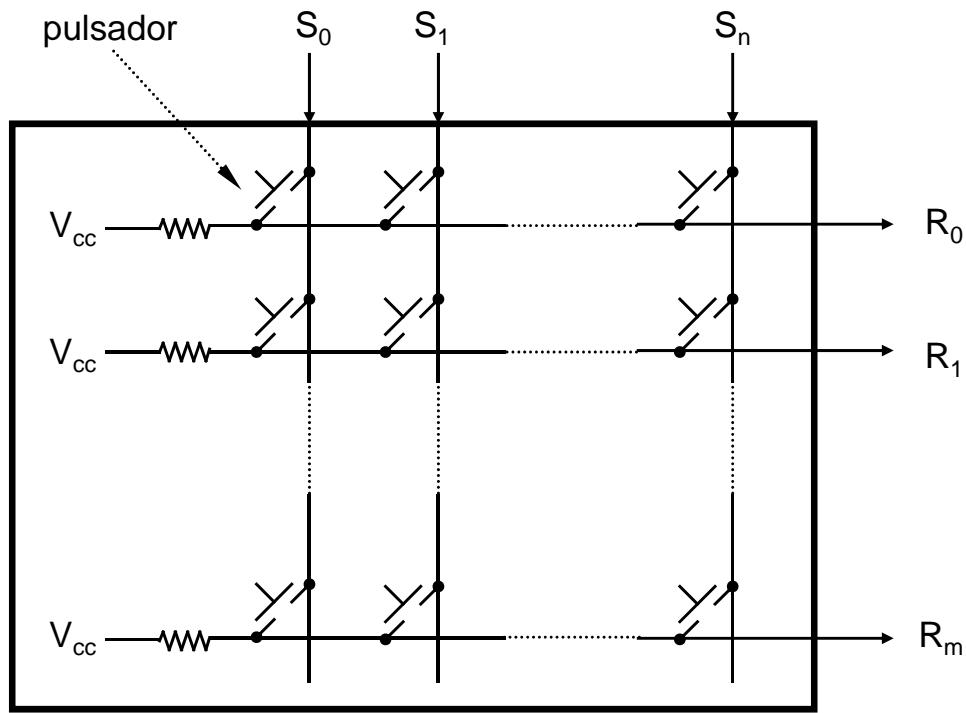
# Matriz de puntos



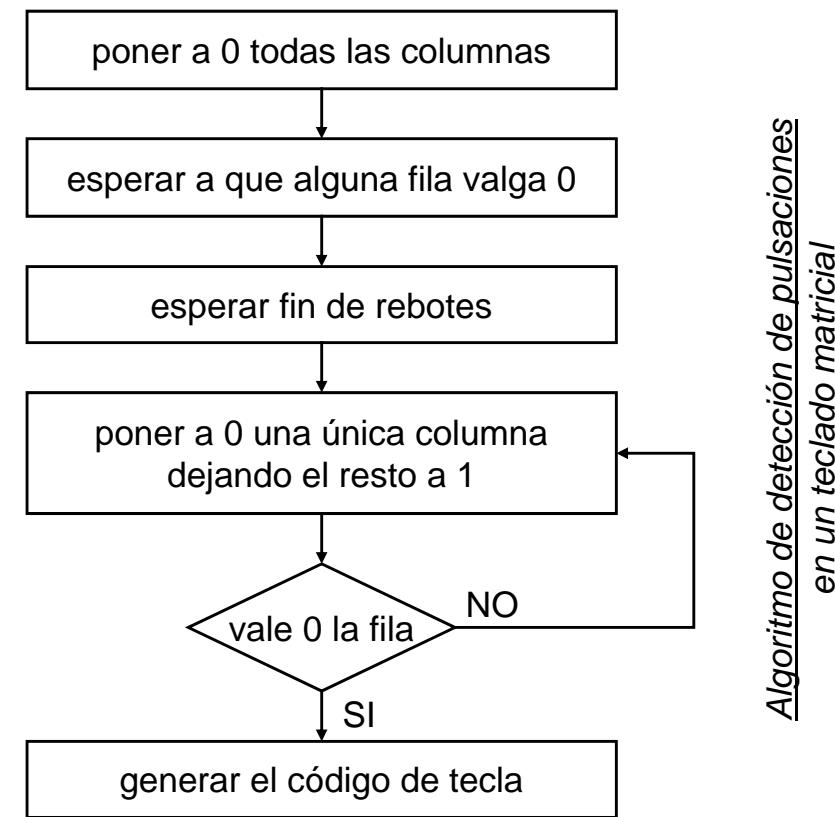


# Teclado matricial

- Un **teclado** es una colección de pulsadores cuyo estado puede ser conocido individualmente
- Las configuraciones más comunes son:
  - **teclados lineales**: cada pulsador tiene una línea dedicada para conocer su estado
  - **teclados matriciales**: los pulsadores se disponen en filas y columnas, de manera que los pulsadores de la misma fila comparten la línea que permite conocer su estado



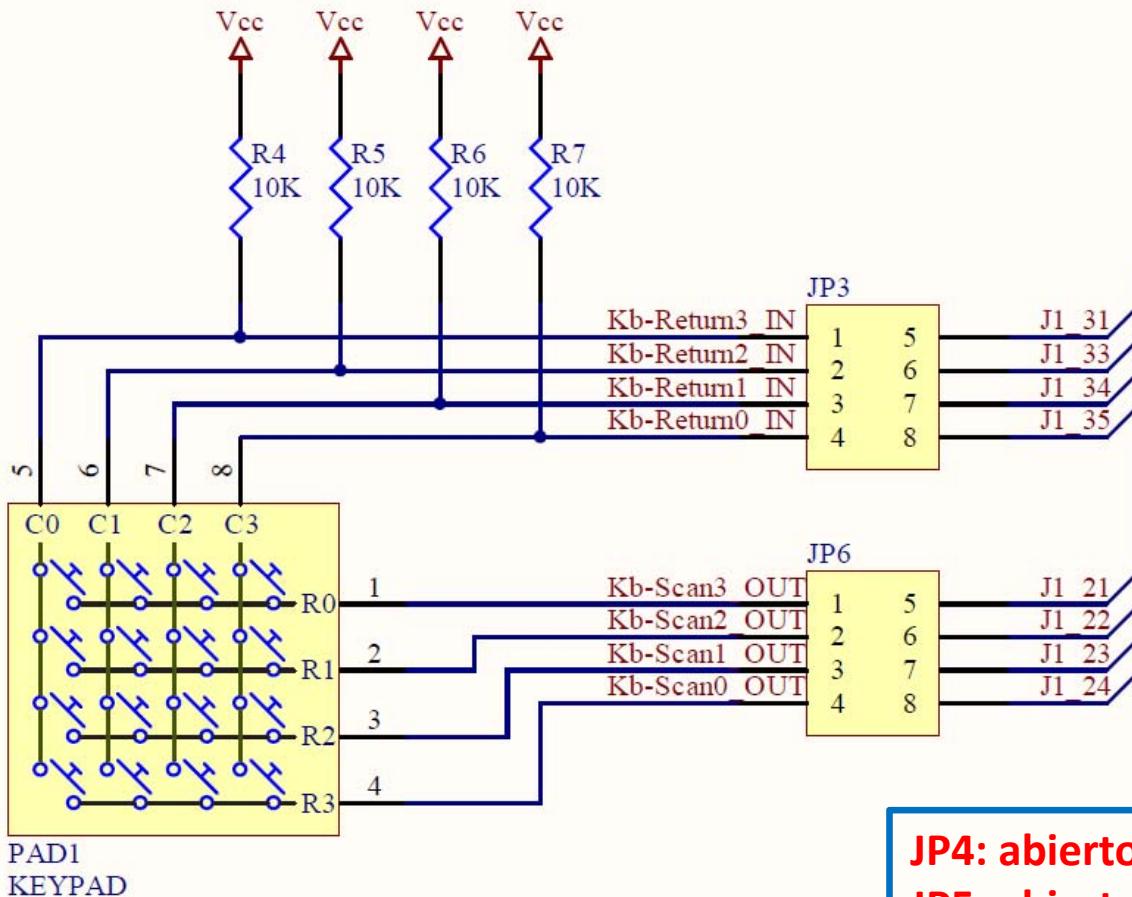
teclado matricial



Algoritmo de detección de pulsaciones en un teclado matricial



# Teclado matricial



JP4: abierto  
JP5: abierto  
JP3 (4 jumpers): cerrado



# Altavoz / zumbador

- Un **altavoz**, es un transductor capaz de generar una onda sonora análoga (en frecuencia y amplitud) a una señal eléctrica dada.
- Se compone de una membrana elástica unida a una bobina móvil que se monta dentro del campo magnético de un imán permanente.
  - La fuerza de atracción entre la bobina y el imán es función de la intensidad y el sentido de la corriente que circule por la bobina.
  - Cambios en la corriente, provocan movimientos en la bobina que se traducen en vibraciones de la membrana. Si estas vibraciones tienen la frecuencia adecuada, se escucha un sonido.
- Dado que un sistema digital puede generar una señal digital periódica a través de uno de sus pines, puede producir sonidos si dicho pin se conecta un altavoz.
  - Si se desea generar un sonido complejo, se debe generar por separado cada uno de sus armónicos y sumarlos para generar una única señal.
- Un **zumbador** es un pequeño altavoz conectado a un oscilador de frecuencia fija.
  - desde un sistema digital solo puede controlarse si suena o no



# Altavoz / zumbador

- Un **sonido** es una onda longitudinal mecánica con una frecuencia comprendida entre 20 y 20000 Hz.
  - **Tono**: percepción humana de la frecuencia de una onda
  - **Volumen**: percepción humana de la intensidad de una onda que depende principalmente de su amplitud.
  - **Timbre**: percepción humana de la forma de la onda.
- Un **altavoz**, es un transductor capaz de generar una onda sonora análoga (en frecuencia y amplitud) a una señal eléctrica dada.
  - Se compone de una membrana elástica unida a una bobina móvil que se monta dentro del campo magnético de un imán permanente.
  - Dado que un sistema digital puede generar una señal digital periódica a través de uno de sus pines, puede producir sonidos si dicho pin se conecta un altavoz.
- Un **zumbador** es un pequeño altavoz conectado a un oscilador de frecuencia fija.
  - desde un sistema digital solo puede controlarse si suena o no

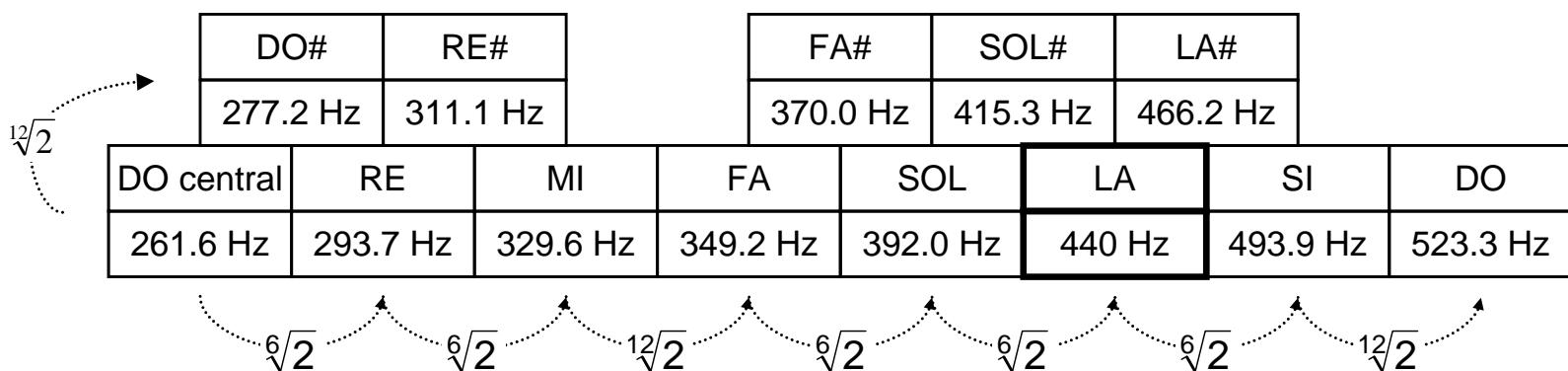


# Altavoz /zumbador

- Una **escala uniformemente temperada** incluye 13 notas por octava cuyas frecuencias están uniformemente espaciadas
  - cualesquiera 2 notas consecutivas difieren entre sí en un semitono, y sus frecuencias siempre guardan una relación de  $\sqrt[12]{2}$
  - Este hecho permite calcular la frecuencia absoluta de cualquier nota a partir de la frecuencia de una de ellas según la fórmula

$$f_{nota} = f_{LA} \cdot \sqrt[12]{2^n} = 440 \cdot \sqrt[12]{2^n} \approx 440 \cdot 1.06^n$$

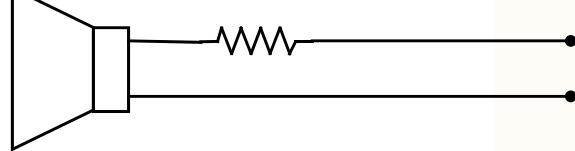
donde  $n$  es el número de notas de separación respecto del LA (negativo hacia la izquierda).



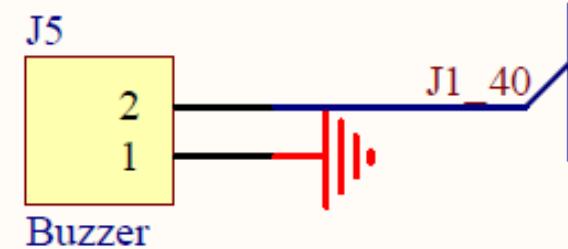
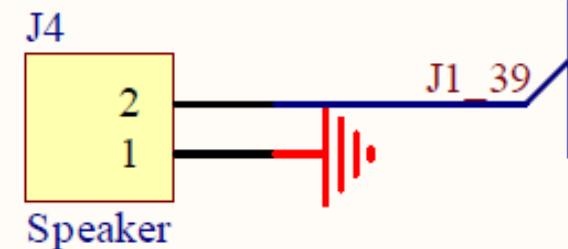
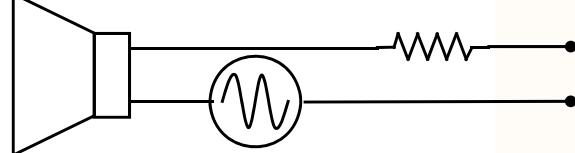


# Altavoz / zumbador

altavoz

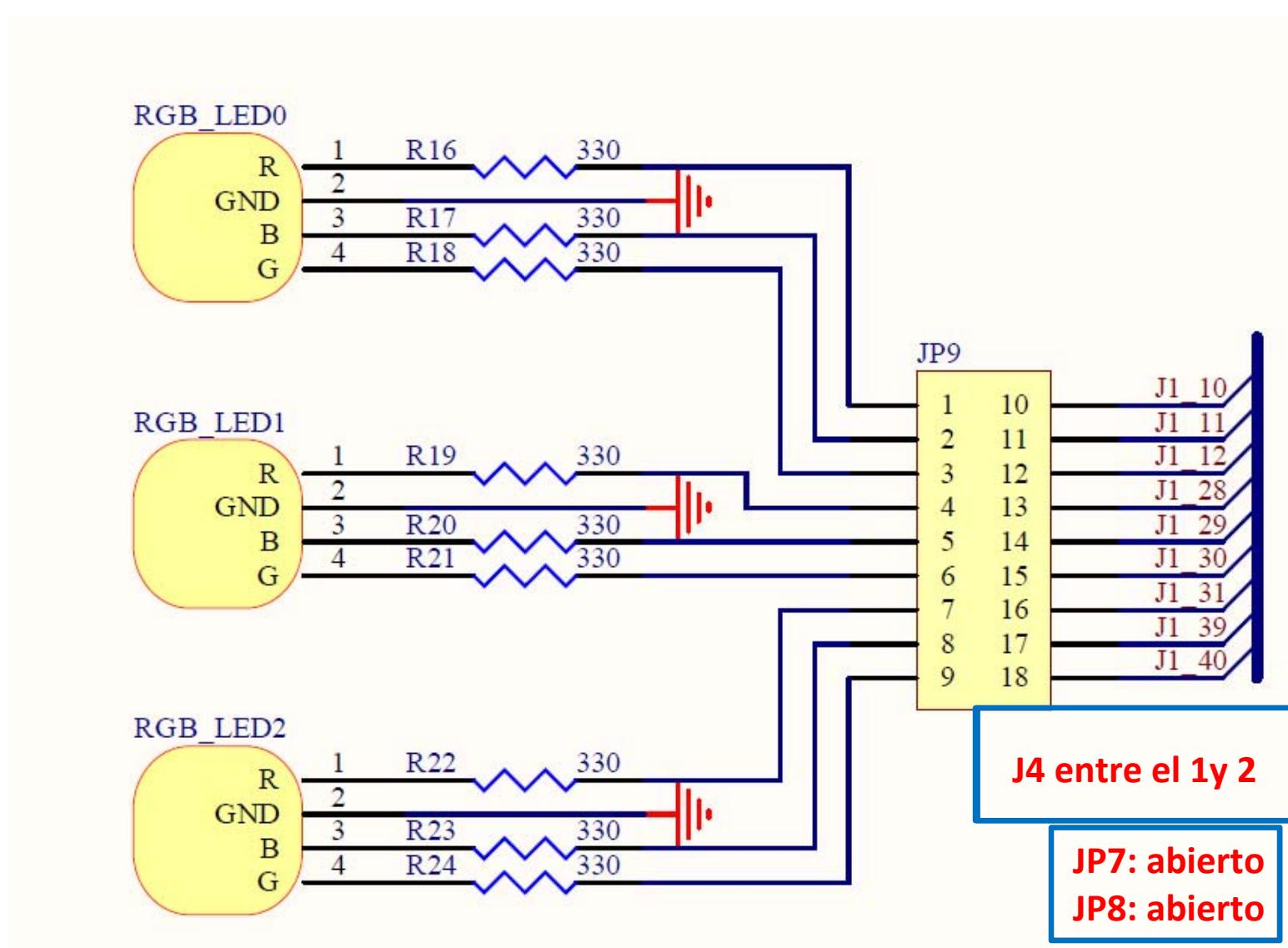


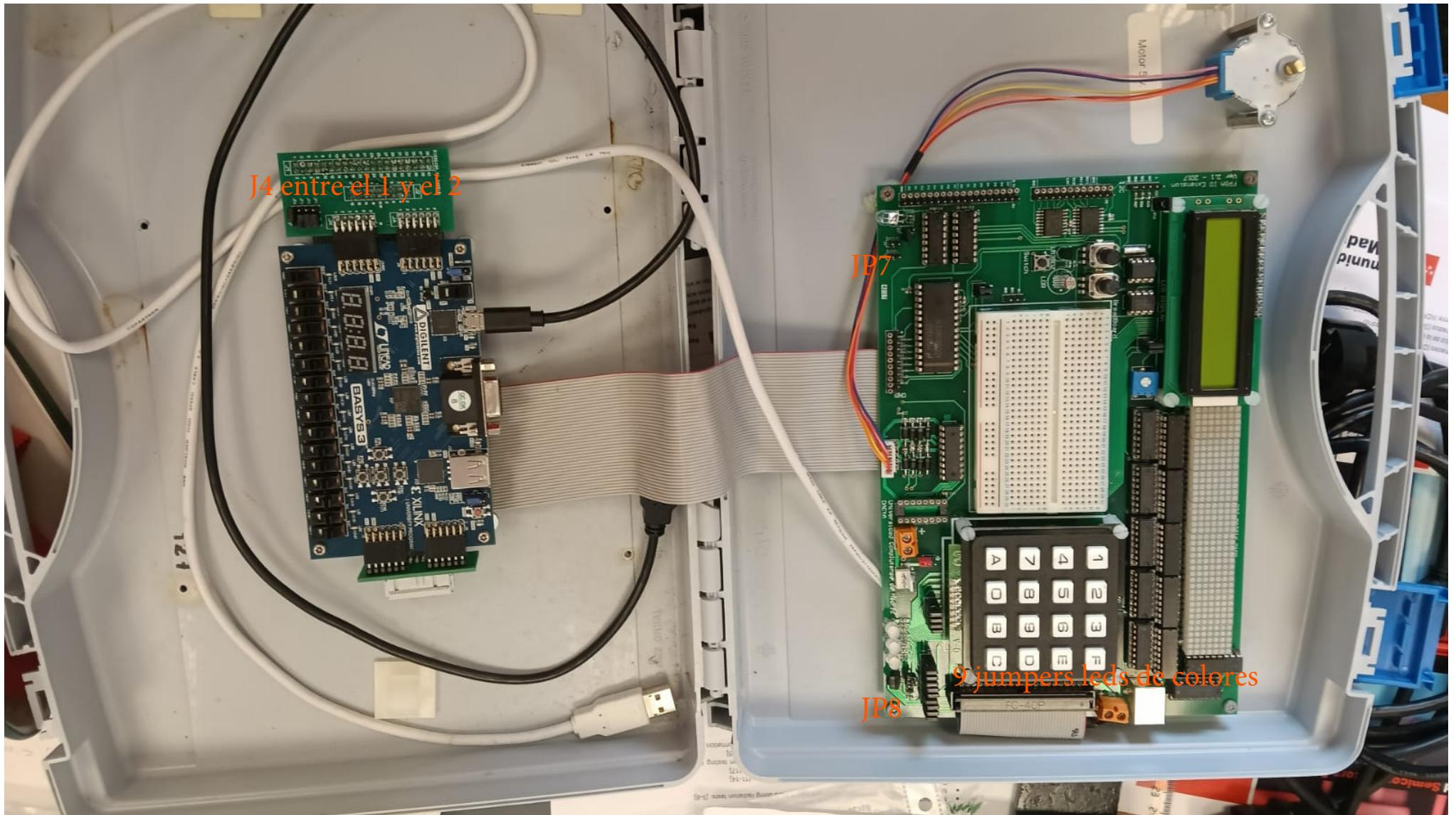
zumbador





# Leds RGB



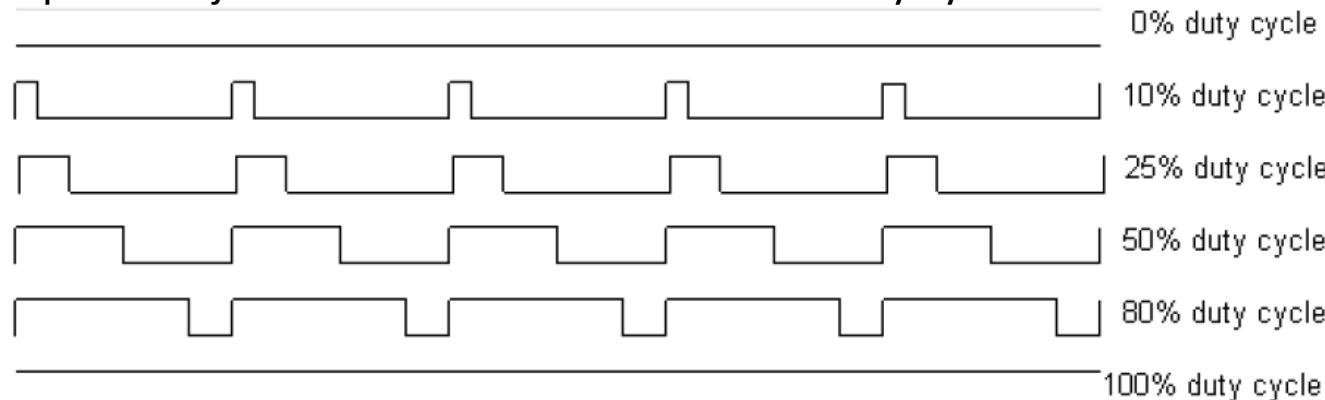




# Leds RGB

- **Pulse Width Modulation** es una técnica de modulación de ondas cuadradas que determina la anchura de los pulsos transmitidos.

- Determina qué porcentaje del período de oscilación que la señal vale 1.
  - A este porcentaje se le denomina comúnmente duty cycle



- Una señal PWM se puede utilizar para controlar la intensidad de encendido de un LED.
  - Si la frecuencia de esta señal es lo suficientemente alta, el ojo no detectará que el LED realmente está parpadeando, con lo que únicamente apreciará que éste se enciende con más o menos intensidad.
  - Si se controla por separado las intensidades relativas de un led RGB, puede visualizarse cualquier color.



# Emisor / transmisor infrarrojos

- Un **emisor de infrarrojos** es un LED que emite luz infrarroja.
- Un **receptor de infrarrojos** es un dispositivo capaz de capturar códigos enviados a través de luz infrarroja.
  - Estos dispositivos solo detectan señales infrarrojas emitidas dentro de una cierta frecuencia portadora, en nuestro caso 38 KHz
  - La frecuencia portadora debe ser mucho mayor que la frecuencia de transmisión de datos y sirve para modular los datos de modo que ocupen un cierto ancho de banda alrededor de la frecuencia portadora.

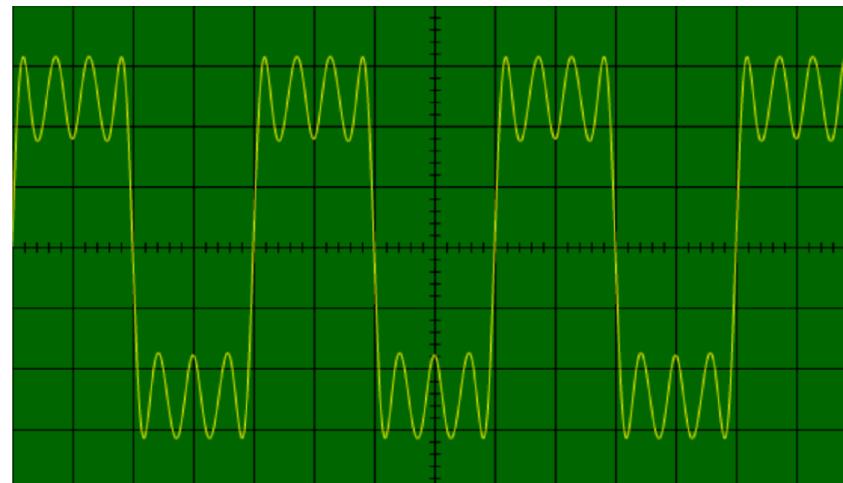
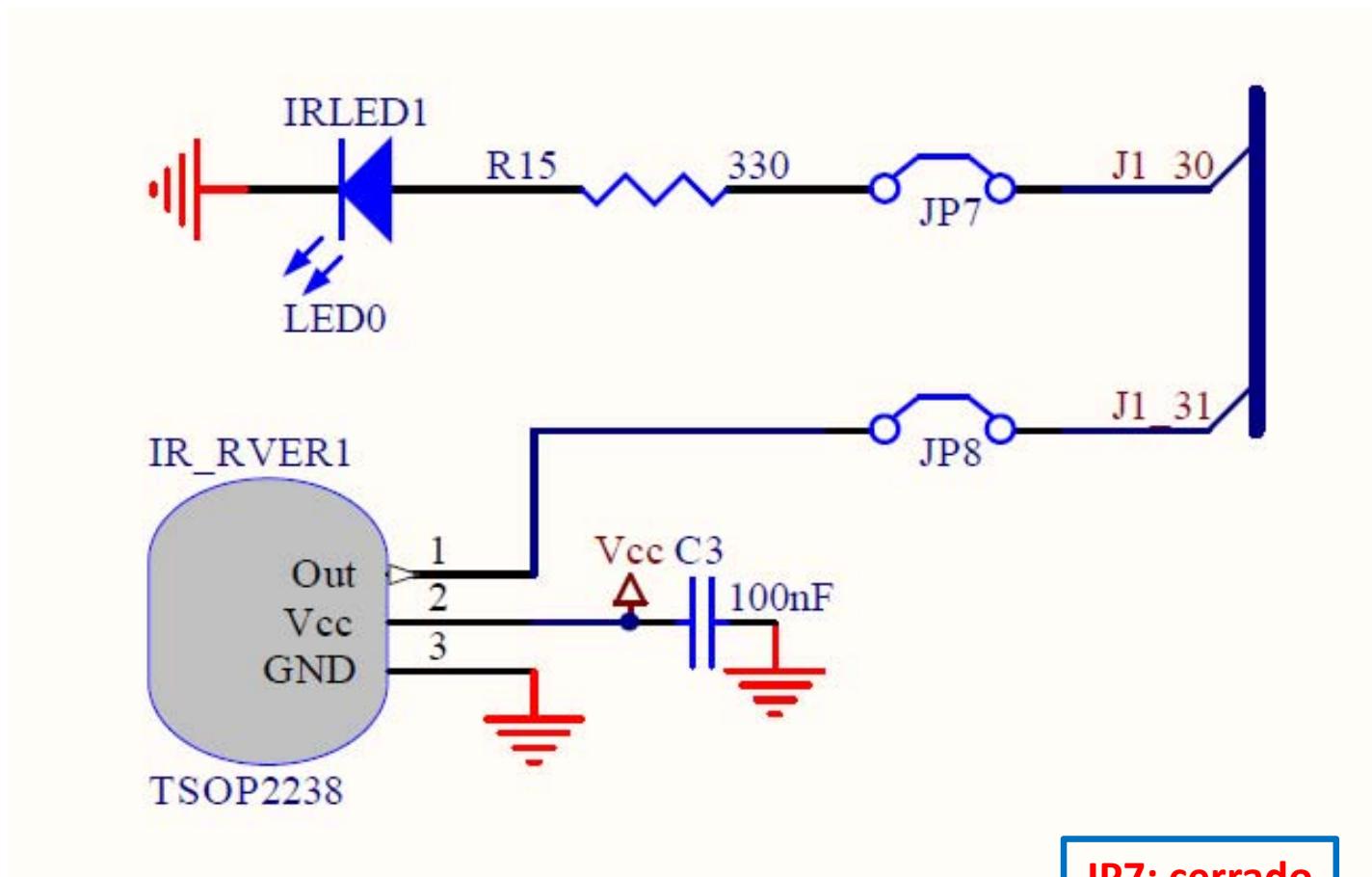


FIGURA 10.3: Ejemplo de una portadora de 38 KHz modulada con una onda de 9 KHz



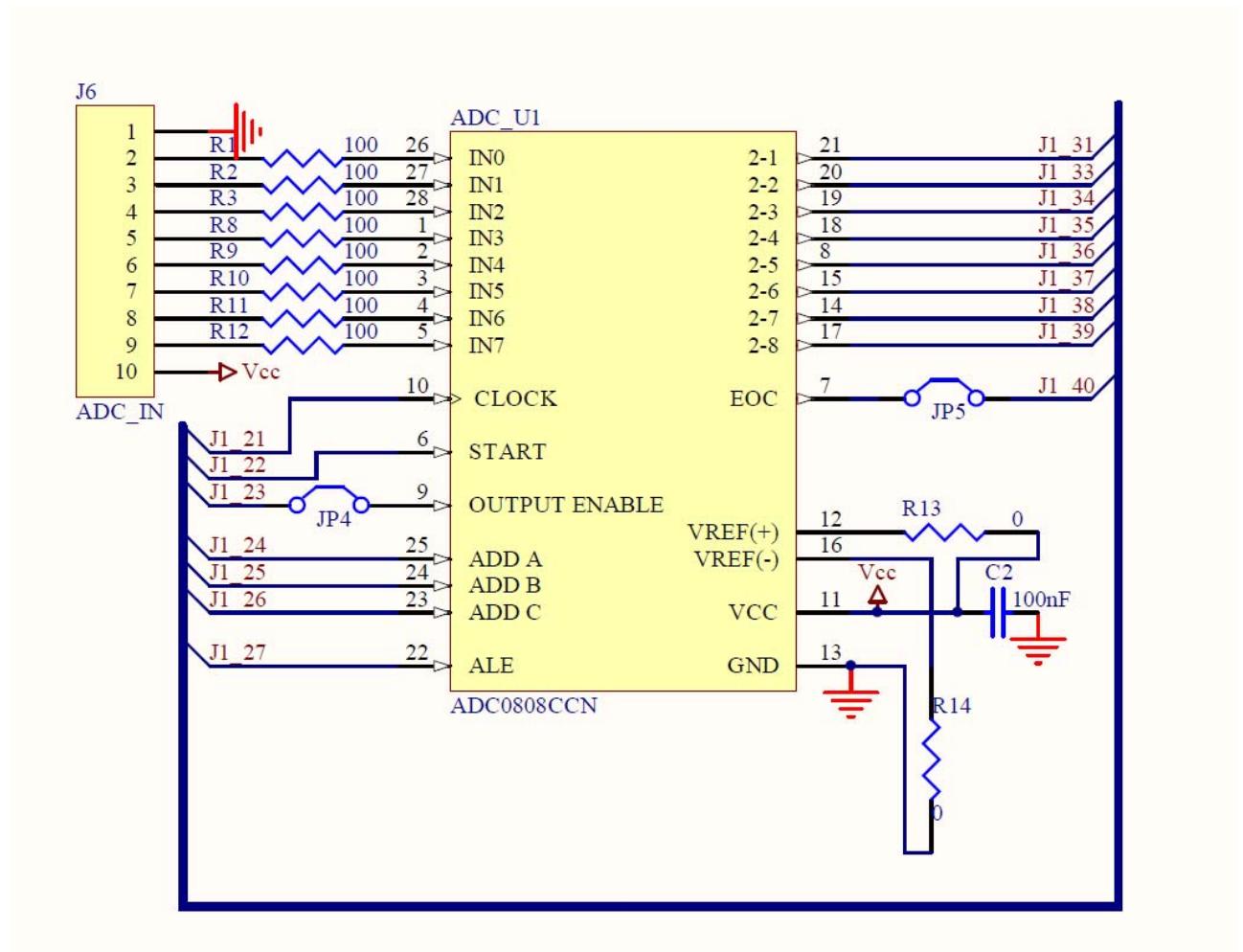
# Emisor / transmisor infrarrojos



JP7: cerrado  
JP8: cerrado



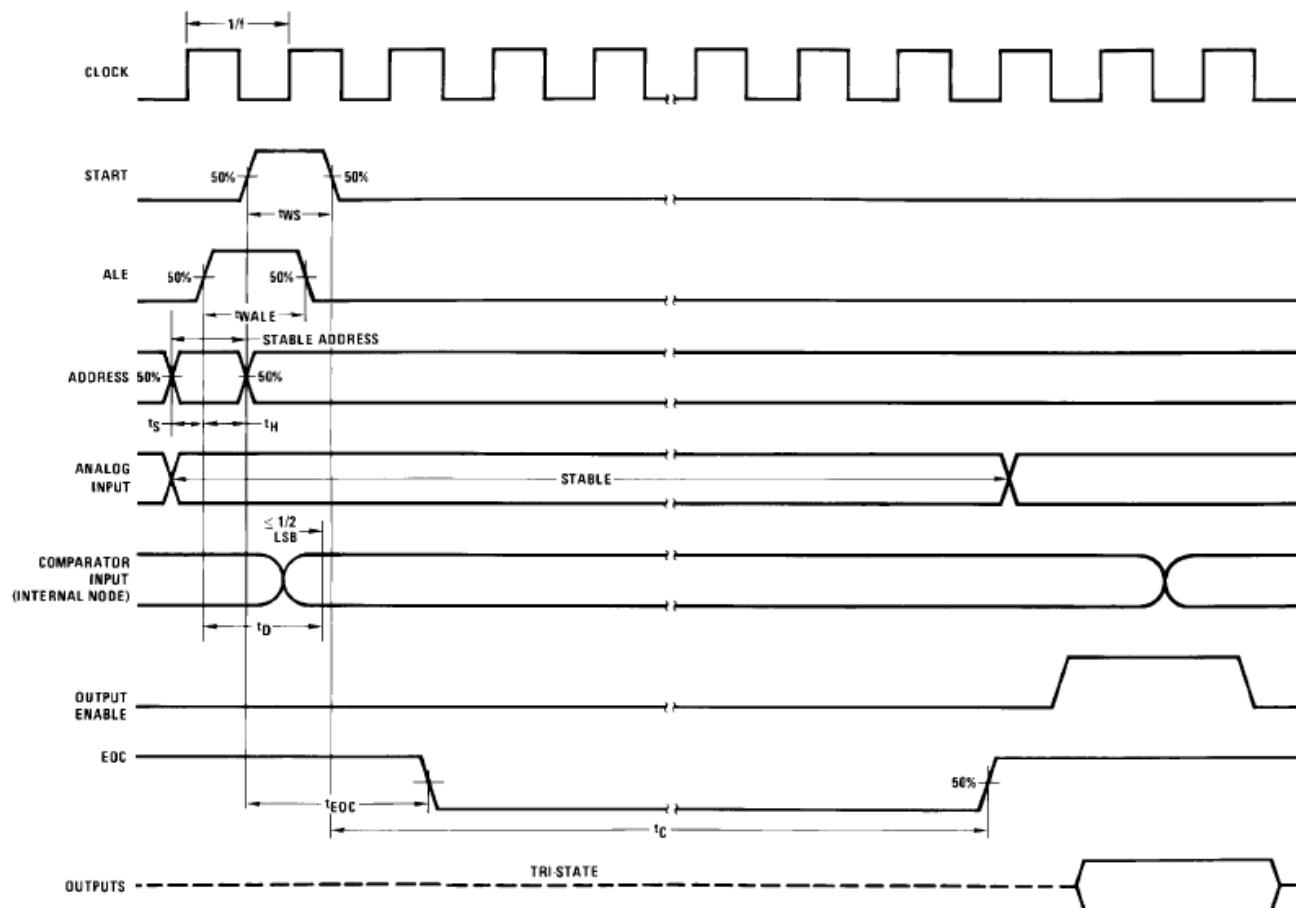
# Conversor ADC



JP4: cerrado  
JP5: cerrado  
JP3 (4 jumpers): abierto



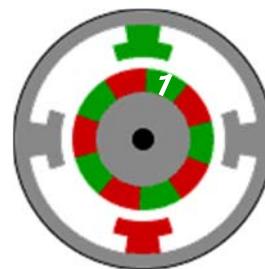
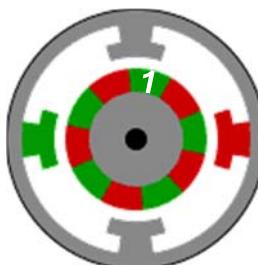
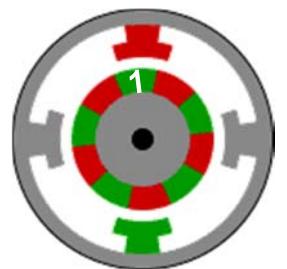
# Conversor ADC



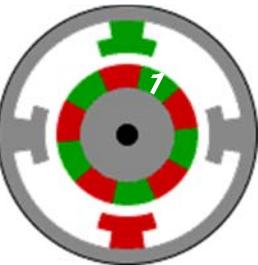
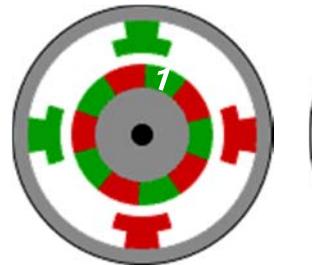
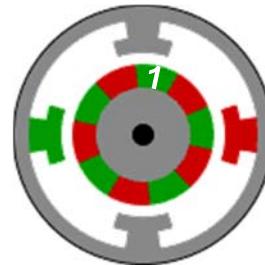
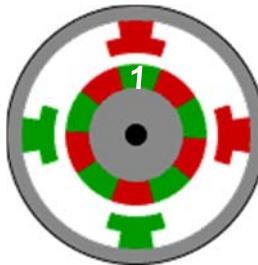
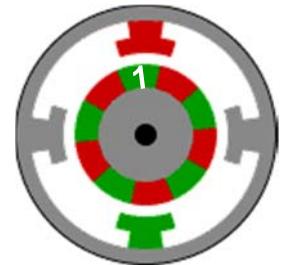


# Motores paso a paso

- Un **motor paso a paso** está formado por:
  - Un rotor, que generalmente es un imán permanente.
  - Un estator, formado por varias bobinas.
- Cuando la corriente circula por una de las bobinas, se crea un campo magnético que hace girar el rotor hasta alinearse con aquél.
  - Activando las bobinas del estator en la secuencia adecuada, puede conseguirse que el rotor gire en cada momento un ángulo determinado.



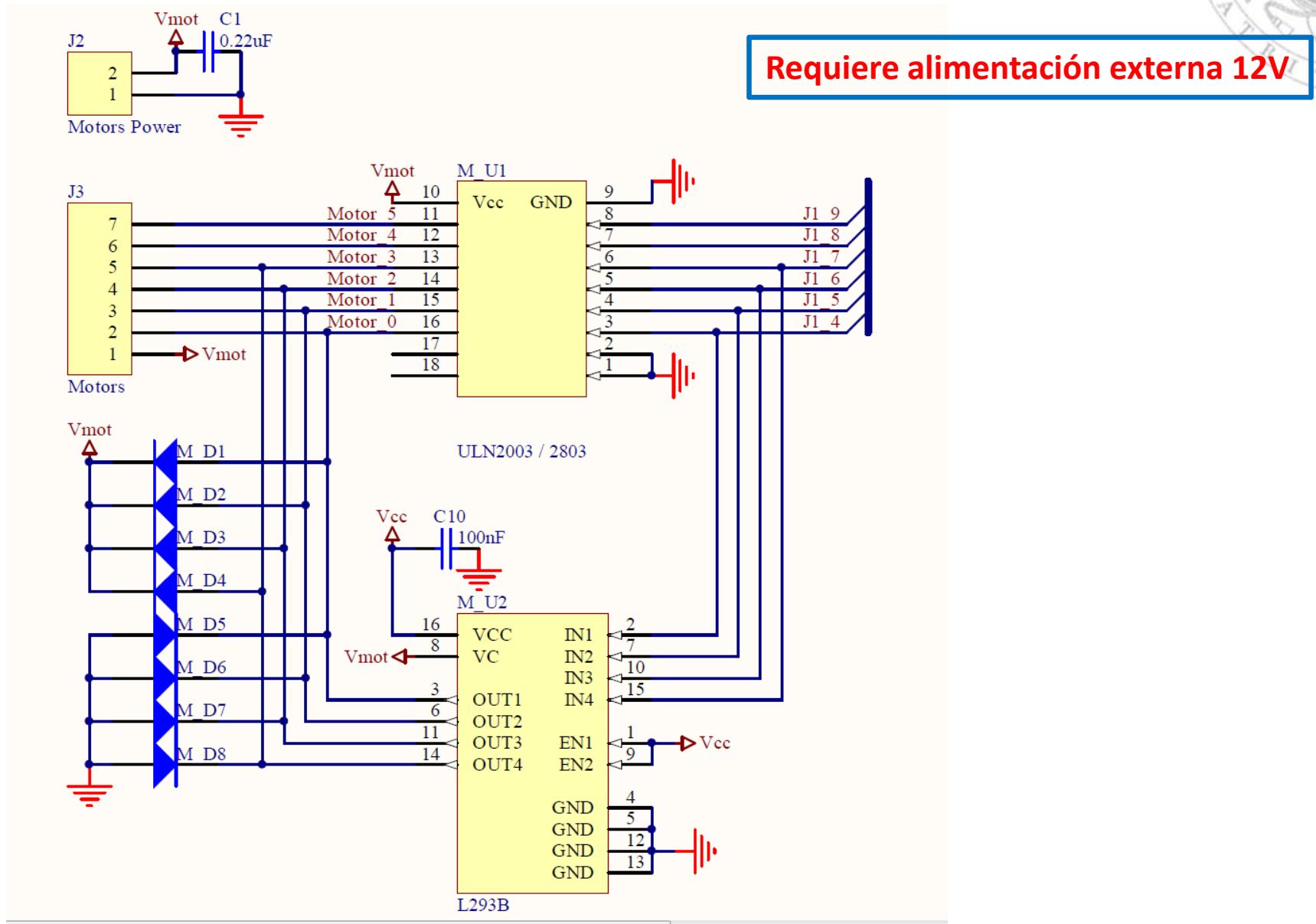
Full step



Half step



# Motor paso a paso





# Pineado y conflictos

XStend Board					XSA-3S1000	JP3,JP6	JP4, JP5	JP2	JP1	JP7, JP8	JP9				
Proto. Pin	Net Name	Direction	LEDs	Switches	FPGA Pin	J1	Teclado	Adc	LCD	DotMatrix	IR	Leds	Altavoz	Zumbador	Motor
52	GND				GND	1									
22	+2.5V				+2.5V										
54	+3.3V				+3.3V										
2	+5V				+5V										
23	I2C-SCL	OUT			F5										
24	I2C-SDA	IN/OUT			D2										
79	ETHER-IRQ	IN	BAR8		L15	4								Out-0	
57	PB-A0	OUT	BAR1	PB2	L5	5								Out-1	
56	PB-A1	OUT	BAR2	PB3	N2	6								Out-2	
51	PB-A2	OUT	BAR3	PB4	M3	7								Out-3	
50	PB-A3	OUT	BAR4		N1	8								Out-4	
70	PB-A4	OUT	BAR5		T13	9								Out-5	
71	PB-D0	IN/OUT		DIPSW1	P12	10	RS	Row_Serial_OUT			R-0				
40	PB-D1	IN/OUT		DIPSW2	J1	11	RW	Row_Clk			B-0				
39	PB-D2	IN/OUT		DIPSW3	H1	12	E	Row_Reset			G-0				
38	PB-D3	IN/OUT		DIPSW4	H3	13	D0	Col_Serial_OUT							
35	PB-D4	IN/OUT		DIPSW5	G2	14	D1	Col_Clk							
80	PB-D5	IN/OUT		DIPSW6	K15	15	D2	Col_Reset							
81	PB-D6	IN/OUT		DIPSW7	K18	16	D3								
10	PB-D7	IN/OUT		DIPSW8	F15	17	D4								
27	PB-D8	IN/OUT	LED2-A		E2	18	D5								
28	PB-D9	IN/OUT	LED2-B		E1	19	D6								
29	PB-D10	IN/OUT	LED2-C		F3	20	D7								
32	PB-D11	IN/OUT	LED2-D		F2	21	Scan-3	Clock							
33	PB-D12	IN/OUT	LED2-E		G4	22	Scan-2	Start							
34	PB-D13	IN/OUT	LED2-F		G3	23	Scan-1	OE							
36	PB-D14	IN/OUT	LED2-G		G1	24	Scan-0	Address A							
37	PB-D15	IN/OUT	LED2-DP	PB1	H4	25	Address B								
61	PB-RD#	OUT			P2	26	Address C								
62	PB-WR#	OUT			R1	27	ALE								
18	SLOT1-CS#	OUT			E15	28	Out-7				R-1				
84	SLOT1-IRQ	IN	BAR9		J16	29	Out-8				B-1				
19	SLOT2-CS#	OUT			D18	30					Out-01	G-1			
83	SLOT2-IRQ	IN	BAR10		J14	31	Return-3	In-0			In-01	R-2			
1	VIDIN-CLK	IN			H18										
4	VIDIN-Y0	IN	LED1-A		H14	33	Return-2	In-1							
59	VIDIN-Y1	IN	LED1-B		M4	34	Return-1	In-2							
58	VIDIN-Y2	IN	LED1-C		P1	35	Return-0	In-3							
60	VIDIN-Y3	IN	LED1-D		N3	36		In-4							
78	VIDIN-Y4	IN	LED1-E		M15	37		In-5							
5	VIDIN-Y5	IN	LED1-F		H13	38		In-6							
6	VIDIN-Y6	IN	LED1-G		G18	39		In-7			B-2	Out-0			
77	VIDIN-Y7	IN	LED1-DP		N15	40		EOC			G-2	Out-0			



# Prueba

- Conectar cable RS-232 y arrancar un emulador de terminal (Termite):
  - Velocidad: 9600 bps
  - Bits de datos: 8
  - Bits de stop: 1
  - Paridad: none
  - Control de flujo: none
- Ajustar los jumpers cuando corresponda
- Descargar el test correspondiente