

CAPÍTULO 3. PROTOCOLOS Y COMANDOS.

3.1. INTRODUCCIÓN.

Para poder desarrollar el programa será imprescindible tener un perfecto conocimiento de los distintos protocolos de comunicación, tanto de los sensores de temperatura como de la pantalla LCD. El microcontrolador debe controlar dichas comunicaciones atendiendo a las distintas ranuras de tiempo que tengamos en cada caso, asegurando así la integridad de los datos tanto enviados y como recibidos. En cuanto a los comandos de control de cada componente nos servirán para dar las ordenes pertinentes a cada periférico para realizar las funciones deseadas. Será importante que los comandos se usen de forma conveniente para no crear ningún conflicto en el funcionamiento de los dispositivos, especialmente en cuanto a los sensores de temperatura se refiere, ya que, como veremos a continuación, no se puede enviar cualquier comando en cualquier orden.

3.2. SENSORES DE TEMPERATURA.

Los sensores de temperatura poseen una gran variedad de comandos de control, si bien solo detallaremos los que se vayan a usar en el proyecto, pudiéndose encontrar información sobre el resto de los comandos en el datasheet correspondiente, que se puede descargar gratuitamente de www.maxim-ic.com.

3.2.1. Protocolos.

Todo acceso que se realice a los sensores a través de la línea de datos debe seguir un estricto protocolo que comprende los siguientes pasos:

1. Inicialización.
2. Comando de función de ROM.
3. Comando de función de RAM.
4. Operación/Envío o recepción de datos.

Debido a que la interfaz de los sensores es de un único hilo tendremos 4 protocolos de comunicación principales, escritura de 0, escritura de 1, lectura de 0 y lectura de uno. A estos debemos incluir el reset de los sensores y el pulso de presencia de los mismos. Será muy importante cumplir con las temporizaciones fijadas en cada caso, ya que debido a que los sensores se alimentan por la misma línea de datos a través de una resistencia de pull-up la línea deberá estar en alto el tiempo necesario para que se entregue a los sensores la alimentación necesaria.

- **Protocolos de escritura.-** El comienzo de una ventana de escritura comienza cuando el maestro (en nuestro caso el microcontrolador) conmuta la línea de nivel lógico alto dado por un pull-up, a valor lógico bajo. Todas las ventanas de escritura deben tener un mínimo de 60 μ s de duración (T_x), con un mínimo de 1 μ s de tiempo de recuperación (t_{REC}) entre dos ciclos de escritura individuales.

El DS18S20 muestrea la línea de entrada/salida en una ventana que va desde 15 μ s hasta 60 μ s después de que la línea haya caído a cero. Si la línea está a nivel alto cuando el sensor muestrea se escribe un "1", y si está a nivel bajo se escribe un "0".

Cuando el maestro genera una ventana de escritura de "1" la línea de datos debe ser puesta a nivel bajo y después liberada, permitiendo que el pull-up ponga la línea a nivel alto antes de 15 μ s después del comienzo de la ranura.

Cuando el maestro genera una ventana de escritura de "0" la línea de datos debe ponerse a nivel bajo y ser mantenida así durante 60 μ s.

Todo este protocolo se puede ver detalladamente en la siguiente figura:

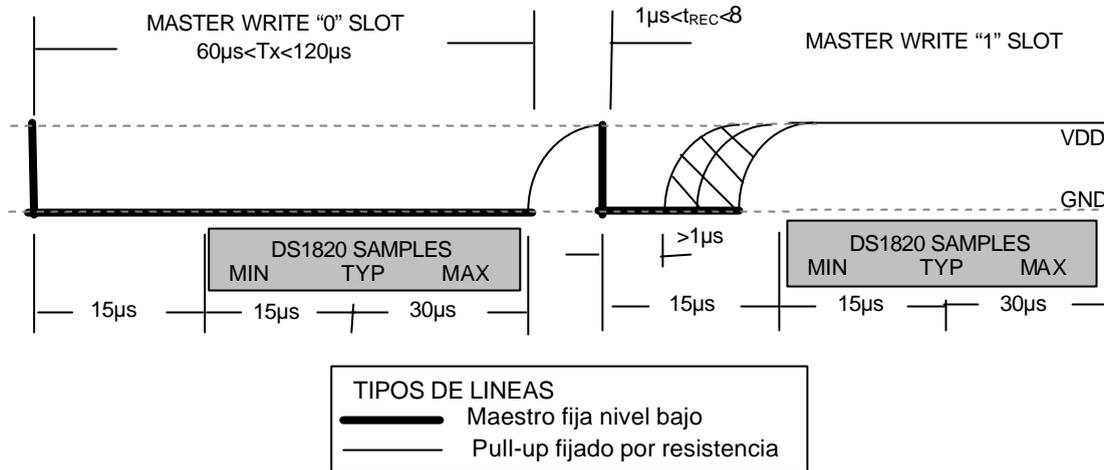


Fig 2.-Ventanas de escritura

- **Protocolos de lectura.-** Una ventana de lectura se inicia cuando el maestro conmuta la línea de datos de nivel alto a nivel bajo. La línea de datos debe permanecer a nivel bajo por al menos 1 μ s, y el dato procedente del DS18S20 es válido durante 15 μ s después del comienzo de la ventana. El maestro por tanto tendrá que liberar la línea antes de 15 μ s para que el DS18S20 tome el control de la misma y ponga el dato correspondiente. Al final de la ventana de lectura la línea de datos será puesta en alta por una resistencia de pull-up externa. Todas las ventanas de lectura deben tener una duración mínima de 60 μ s, con un mínimo de 1 μ s de tiempo de recuperación entre ventanas de lectura individuales. Este protocolo puede verse con más detalle en el siguiente diagrama:

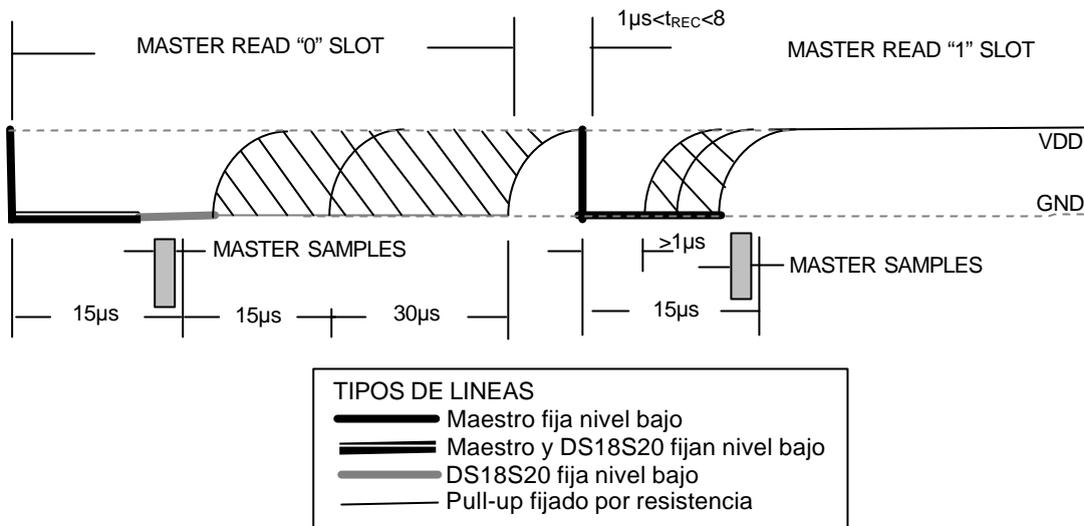


Fig 3.-Ventanas de lectura

Al leer un “1” es recomendable para minimizar cualquier error en la lectura que el tiempo inicio (T_{INIT}) de la ventana de lectura y el tiempo de recuperación (T_{REC}) sean lo menor posible, y que el maestro muestree la línea lo más cerca posible a los 15 μ s desde que se inició la ventana. Esto puede verse de forma detallada en el siguiente diagrama:

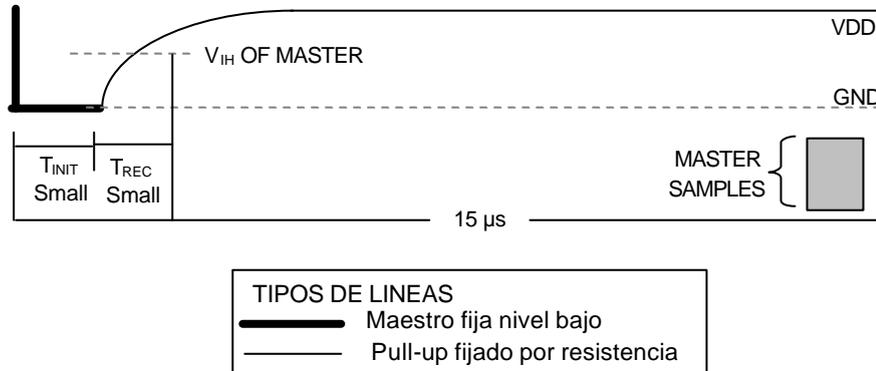


Fig 4.-Lectura de “1” recomendada

- **Reset y Presencia.-** Toda operación que se realice con los sensores deberá estar precedida de una secuencia de inicialización, consistente en un pulso de reset enviado por el microcontrolador seguido de un pulso de presencia enviado por los sensores. Las ventanas de tiempo para el reset y el pulso de presencia se muestran en la siguiente figura:

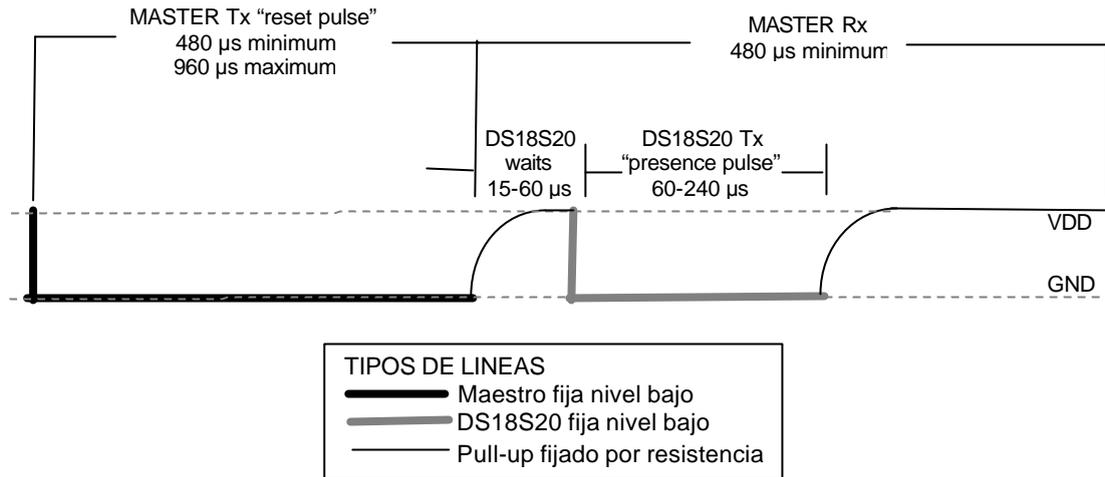


Fig 5.-Pulsos de reset y presencia

Un pulso de reset seguido por un pulso de presencia indica al maestro que el DS18S20 está preparado para recibir datos, pudiendo enviar a continuación el comando de control correspondiente.

Como vemos en la figura el proceso comienza cuando el maestro envía un pulso de reset poniendo la línea a nivel bajo durante un mínimo de 480 μs , liberando después la línea y entrando en modo de recepción de datos. Tras una ventana de 15 a 60 μs en la que se fija nivel alto a través del pull-up el DS18S20 transmite un pulso de presencia fijando la línea a nivel bajo durante un mínimo de 60 μs y un máximo de 240 μs . Esta secuencia de reset seguido por pulso de presencia es especialmente útil para comprobar si se han desconectado los sensores del sistema.

3.2.2. Comandos.

Los sensores de temperatura poseen un variado número de comandos que permiten realizar distintas operaciones, como son la identificación de cuantos sensores hay conectados y cuales son sus números de serie internos, ordenar que se mida la temperatura, leer la temperatura medida, escribir o leer la memoria interna de los sensores, etc. Estos comandos poseen todos una longitud de 8 bits, y se pueden dividir en dos grupos, los comandos de función de ROM, y los comandos de función de RAM. Veamos cada grupo de comandos por separado.

- **Comandos de función de ROM.-** Reciben este nombre debido a que todos implican al número de serie almacenado en la ROM de los sensores, cuya configuración se muestra a continuación:



Fig 6.-Distribución de la ROM de los sensores

Como vemos los 8 primeros bits de la ROM son un identificador de la familia de dispositivos, por lo que coincide para todos los sensores. Los siguientes 48 bits son el número de serie de los sensores propiamente dichos, y los 8 bits más significativos son un código de redundancia cíclica (CRC) del resto de los bits que se puede usar como código detector de errores por el maestro en la identificación de los sensores.

En la siguiente tabla se recoge una descripción de cada comando de ROM junto con su palabra digital correspondiente:

INSTRUCCION	DESCRIPCION	PROTOCOLO
Read ROM	Lee el número de serie de un sensor cuando es el único conectado	33h
Match ROM	Hace que al siguiente comando solo responda un sensor determinado	55h
Skip ROM	Hace que todos los sensores conectados atiendan al siguiente comando	CCh
Search ROM	Comienza con el algoritmo de identificación de sensores conectados	F0h
Alarm Search	Comprueba si ha ocurrido una alarma de temperatura en algún sensor	ECh

Tabla 1.- Comandos de ROM

Puesto que no necesitaremos usar todos los comandos disponibles explicaremos en detalle solo los que nos sean útiles, pudiéndose consultar el resto de comandos en el datasheet correspondiente.

Match ROM.- Este comando seguido por los 64 bits del número de serie de un sensor permita acceder a un DS18S20 determinado. Solo el DS18S20 que posea el número de serie enviado responderá al siguiente comando de función de memoria, mientras que el resto de sensores cuyo número de serie no coincida permanecerán inactivos hasta el siguiente pulso de reset. Este comando es especialmente útil cuando hay varios dispositivos conectados.

Skip ROM.- Este comando está especialmente indicado para ahorrar tiempo en una configuración con un solo sensor, permitiendo al maestro acceder al DS18S20 sin necesidad de enviar el número de serie del sensor. Si existen varios sensores conectados y se envía este comando seguido de un comando de lectura habrá colisión de

datos cuando los distintos esclavos transmitan simultáneamente, por lo que esta situación debe evitarse (La colisión en la línea tendrá como resultado la operación lógica AND de los datos enviados).

Search ROM.- Cuando el sistema es inicialmente encendido el maestro puede no conocer el número de serie de los sensores conectados, ni la cantidad de los mismos. Este comando permite al maestro usar un proceso de eliminación para identificar los 64 bits del número de serie de cada sensor. Debido a la importancia del proceso de identificación de sensores se explicará el algoritmo usado para ello en un apartado por separado.

- **Comandos de función de RAM.-** Reciben este nombre debido a que son comandos para realizar funciones en la memoria RAM o en la E²RAM de los sensores, cuya configuración se muestra a continuación:

RAM	BYTE	E ² RAM
TEMPERARURE LSB	0	
TEMPERATURE MSB	1	
TH/USER BYTE 1	2	TH/USER BYTE 1
TL/USER BYTE 2	3	TL/USER BYTE 2
RESERVED	4	
RESERVED	5	
COUNT REMAIN	6	
COUNT PER °C	7	
CRC	8	

Tabla 2.- Distribución de las memorias RAM y E²RAM

Los 2 primeros bytes se usan para almacenar la temperatura medida, el primer byte almacena la temperatura propiamente dicha y el segundo byte almacena el signo. Los bytes 2 y 3 se usan para que el usuario defina ciertos valores de temperatura que funcionan como alarma, de forma que si la temperatura medida supera estos valores el sensor avisará de ello. Los bytes 6 y 7 se usan para ampliar la resolución de la temperatura medida en el caso de que los 0.5°C de resolución por defecto no sean necesarios, y el último byte es el CRC de los bytes anteriores que puede usar el maestro como código detector de errores cuando realice la lectura de la RAM.

En la siguiente tabla se recoge una descripción de cada comando de RAM junto con su palabra digital correspondiente:

INSTRUCCION	DESCRIPCION	PROTOCOLO
COMANDOS PARA LA CONVERSION DE TEMPERATURA		
Convert T	Inicia la conversión de temperatura	44h
COMANDOS DE MEMORIA		
Read RAM	Lee la memoria RAM	BEh
Write RAM	Escribe en las bytes 2 y 3 de la RAM	4Eh
Copy RAM	Copia la RAM en la E ² RAM (solo los bytes 2 y 3)	48h
Recall E²	Devuelve los valores de la E ² RAM a la RAM	B8h

Tabla 3.- Comandos de RAM

Al igual que en el caso de los comandos de ROM no necesitaremos usarlos todos, por lo que explicaremos en detalle solo los que nos sean útiles, pudiéndose consultar el resto de comandos en el datasheet correspondiente.

Convert T.- Este comando indica a los sensores que comiencen con la medida de temperatura. La medida es realizada por los sensores a través de osciladores dependientes con la temperatura, contando el número de ciclos de los mismos en un determinado periodo de tiempo, realizando así una conversión de temperatura a palabra digital que se almacena en la RAM codificada en complemento 2 con signo. Mientras dure la conversión de temperatura se debe fijar en la línea un pull-up fuerte para que los sensores tengan la alimentación necesaria mientras dure el proceso. El pull-up debe estar activo durante 500 ms.

Read RAM.- Como su propio nombre indica este comando sirve para leer el contenido de la RAM. La lectura comenzará por el byte 0 y seguirá hasta que el byte 9 sea leído. Las posiciones reservadas se leerán con todos sus bits a 1. El maestro puede enviar un pulso de reset en cualquier momento para poner fin a la lectura, lo cual resulta muy útil en el caso de que solo se quieran leer los dos primeros bytes de la RAM correspondientes a la temperatura medida.

NOTA.- Es importante observar que todos los comandos vistos anteriormente, así como cualquier dato que se escriba o lea de los sensores, se envían siempre empezando por el bit menos significativo.

3.3. PANTALLA LCD.

La pantalla LCD es configurable para poder usar interfaz de 8 o de 4 líneas, pero como para reducir el diseño lo máximo posible elegiremos el interfaz de 4 líneas solo nos referiremos a sus protocolos, sin considerar los de interfaz de 8 líneas. Para una mejor comprensión de este apartado empezaremos por ver una descripción de los terminales de que dispone la pantalla LCD:

Señal	Input/ Output	Función
RS	Input	Señal para seleccionar modo de instrucción o de datos High: Modo datos (para lectura y escritura) Low: Modo instrucción o lectura de bus
R/W	Input	Señal usada para seleccionar modo lectura o escritura High: Modo lectura Low: Modo escritura
E	Input	Activa la lectura o escritura de los datos
DB4- DB7	Input/ Output	4 líneas más altas del bus de datos. Usadas para transferir datos entre el maestro y el LCD. DB7 se usa también como señal de busy
DB0- DB3	Input/ Output	4 líneas más bajas del bus de datos. Usadas para transferir datos entre el maestro y el LCD. Estas líneas no son usadas en el interfaz de 4 bits
Vdd Vss		Vdd = 5 V Vss = GND
Vo		Tensión para el ajuste del contraste Vdd - 11 ~ Vdd + 0.3

Tabla 4.- Terminales de la pantalla LCD

Como observamos en esta tabla, y puesto que usamos el interfaz de 4 bits, tendremos 4 líneas de datos y 3 líneas de control, que sirven para seleccionar el modo de comandos o de datos, la lectura y escritura, y para comenzar la lectura o escritura usando el bit de Enable. En concreto con RS y R/W se pueden seleccionar 4 operaciones básicas que se detallan en la siguiente tabla:

RS	R/W	Función
0	0	Escritura de instrucción
0	1	Lectura del bit de busy
1	0	Escritura de datos
1	1	Lectura de datos

Tabla 5.- Operaciones básicas del LCD

3.3.1. Protocolos.

Aunque podríamos detallar ampliamente los protocolos usados para lectura, escritura, inicialización a 4 u 8 bits, etc., simplificaremos tratando solo los que van a ser necesarios para nuestros propósitos, y que dividiremos en 2 grupos, protocolos de comunicación, y protocolo de inicialización para interfaz de 4 bits.

- **Protocolo de comunicación.-** Ya sea para enviar una instrucción, o para enviar un carácter, la comunicación con la pantalla se realizará siempre siguiendo el mismo protocolo, que puede observarse en la siguiente figura:

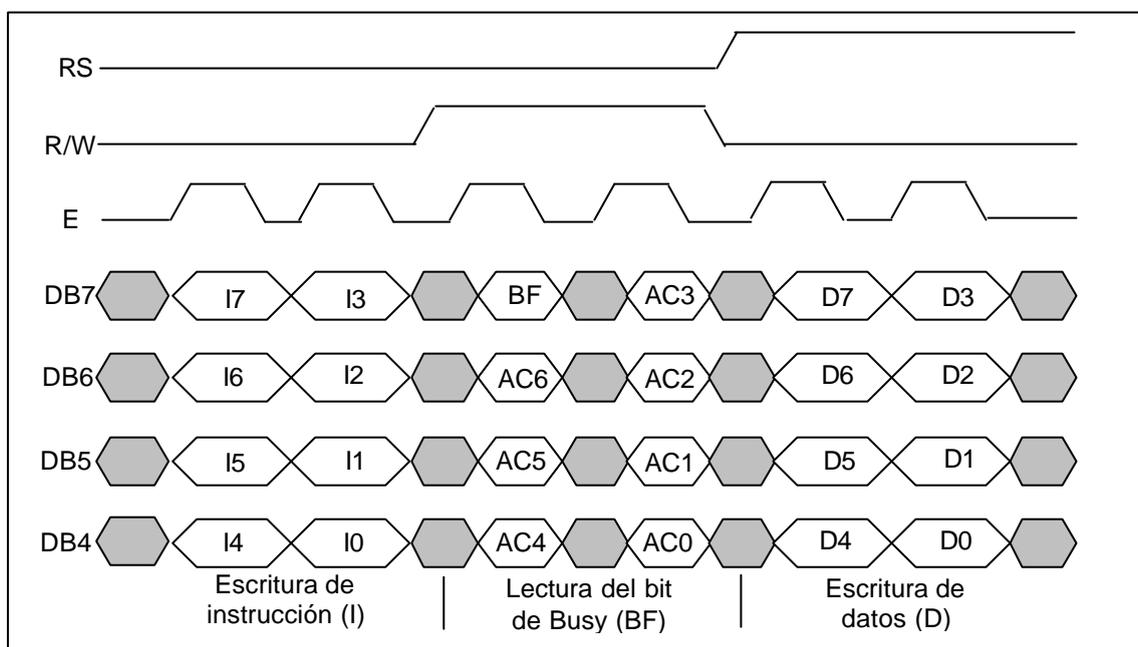


Fig 7.-Protocolo de comunicación con el LCD

Como vemos en el diagrama anterior la escritura de cualquier comando o dato en el LCD se realiza mediante la activación de la señal de Enable. Como usamos el interfaz de 4 bits cada instrucción y cada dato se envían dos pasos, primero los 4 bits más significativo, y después los 4 menos significativos.

Un detalle importante a tener en cuenta es la lectura del bit de Busy, ya que mientras esté activado nos indicará que la pantalla está realizando alguna operación y no podemos enviarle ningún comando, pero no basta con esperar a que se desactive el bit. Como se muestra en la figura cuando se chequea el bit de Busy la pantalla envía también su contador de direcciones interno, lo que hará que el proceso se realice en 2 ciclos, uno donde DB7 se corresponde al Busy, y otro en el que se corresponde al bit 3 del contador de

direcciones (AC3) de la pantalla. Una vez que se han leído los 2 ciclos debe comprobarse si el Busy está activo, y si es así debe volver a repetirse el proceso de 2 ciclos de lectura hasta que el bit de Busy esté inactivo. Si se manda una instrucción o un dato justo después de que el bit de Busy se desactive, sin esperar a leer AC3, producirá un error que generalmente conducirá a que la escritura de cualquier carácter en la pantalla sea incorrecta.

Otro detalle importante será tener en cuenta las temporizaciones necesarias para que la comunicación sea robusta. Dichas temporizaciones se recogen en la siguiente figura:

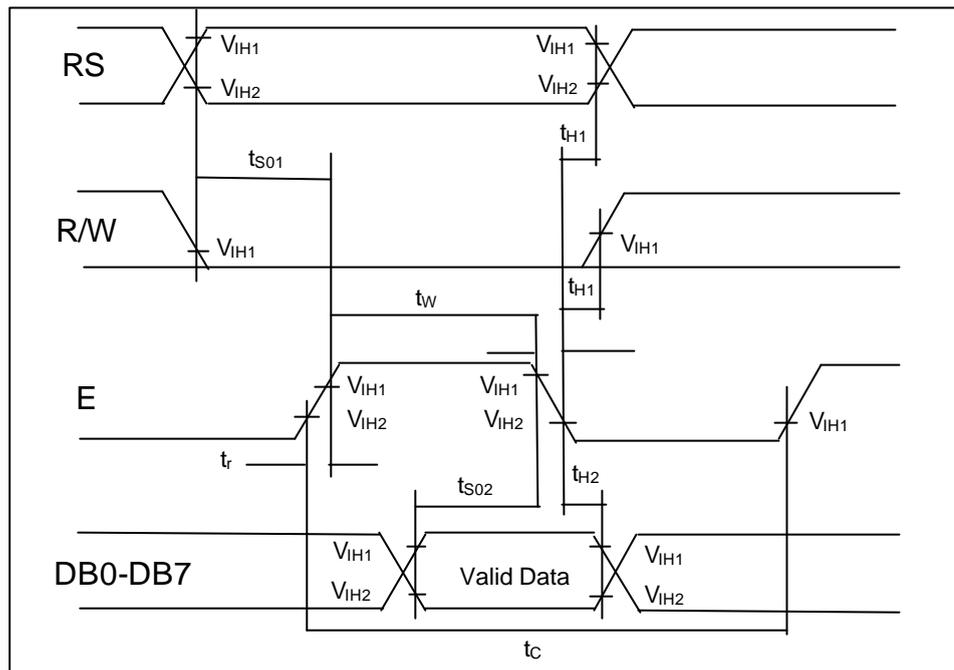


Fig 8.-Ranuras de tiempo del LCD

Los valores concretos de cada intervalo de tiempo se recogen en la siguiente tabla:

Parámetro	Símbolo	Min.	Max.	Unidad	PIN implicado
Enable cycle time	t_c	500	-	ns	E
Enable "H" level pulse width	t_w	220	-	ns	E
Enable rise/fall time	t_r, t_f	-	25	ns	E
RS, R/W setup time	t_{sU}	40	-	ns	R/W, RS
RS, R/W address hold time	t_{H1}	10	-	ns	R/W, RS
Read data output delay	t_d	-	120	ns	DB0-DB7
Read data hold time	t_{dh}	20	-	ns	DB0-DB7

Tabla 6.- Tiempos de operación del LCD

Puesto que el microcontrolador elegido funciona a 4 MHz no habrá ningún problema en asegurar el cumplimiento de estos tiempos, asegurando así la integridad de los datos enviados al LCD. Estas ranuras de tiempo se corresponden con el protocolo de escritura, que será el único que usemos. Para las ranuras de tiempo de lectura consultar el datasheet del LCD.

- **Protocolo de inicialización.-** Cuando se conecta la alimentación el LCD arranca por defecto en interfaz de 8 bits, por lo que será necesario como primer paso la inicialización de la pantalla para interfaz de 4 bits. El proceso para dicha inicialización se recoge en el siguiente diagrama:

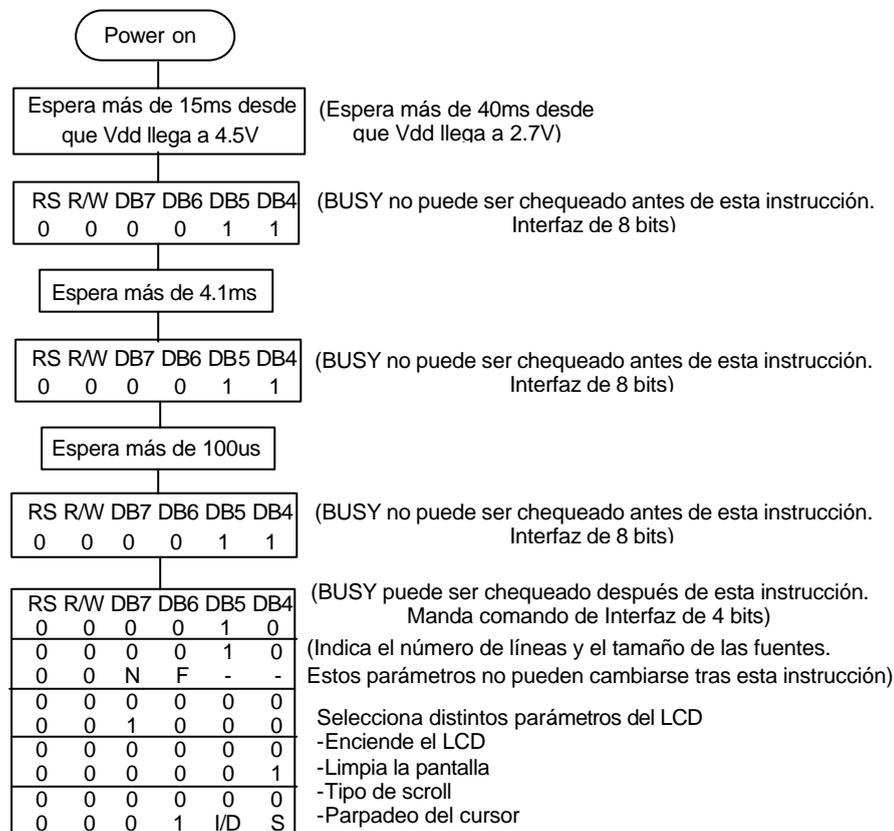


Fig 9.-Diagrama de inicialización para interfaz de 4 bits

Es muy importante que este protocolo se cumpla estrictamente, especialmente en cuanto a temporizaciones se refiere, ya que de lo contrario se corre el riesgo de que la pantalla permanezca en interfaz de 8 bits, no pudiendo controlar la misma.

3.3.2. Comandos.

Debido al elevado número de operaciones que se pueden realizar con el LCD la colección de comandos que presenta es muy amplia. La totalidad de instrucciones que se pueden usar se recogen en la siguiente tabla:

Comando	CODIGO										Descripción	Tiempo de ejecución
	RS	RW	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0		
Clear display	0	0	0	0	0	0	0	0	0	1	Limpia el display y lleva el cursor a la 1º posición	1.64 ms
Return home	0	0	0	0	0	0	0	0	1	-	Lleva el cursor a la 1º posición	1.64ms
Entry mode set	0	0	0	0	0	0	0	1	I/D	1	Selecciona el movimiento del cursor y especifica el desplazamiento o no del display	40µs
Display on-off control	0	0	0	0	0	0	1	D	C	B	Selecciona on/off del display (D), cursor on/off (C) y parpadeo del cursor (B)	40µs
Cursor or display shift	0	0	0	0	0	1	S/C	R/L	-	-	Mueve el cursor y desplaza el display sin cambiar la DDRAM	40µs
Function set	0	0	0	0	1	DL	N	F	-	-	Selecciona longitud de interfaz (DL), nº de líneas del display (N) y tamaño de fuente (F)	40µs
Set CGRAM address	0	0	0	1	AG	AG	AG	AG	AG	AG	Selecciona la dirección de CGRAM. El dato de CGRAM es enviado o recibido tras esta instrucción	40µs
Set DDRAM address	0	0	1	AD	Selecciona la dirección de DDRAM. El dato de DDRAM es enviado o recibido tras esta instrucción	40µs						
Read busy flag&address	1	1	BF	AC	AG	AG	AG	AG	AG	AG	Lee el bit de busy y el contador de direcciones	0µs
Write data	1	0	Dato a escribir								Escribe dato en DDRAM o CGRAM	46µs
Read data	1	1	Dato a leer								Lee dato de DDRAM o CGRAM	46µs
I/D=1:Incremento I/D=0:Decremento S=1:Acompaña desplazamiento del display S/C=1:Desplaza display S/C=0:Mueve el cursor R/L=1:Desplaza a la derecha R/L=0:Desplaza a la izquierda DL=1:8 bits, DL=0:4 bits N=1:2 líneas, N=0:1 línea F=1:5x10 puntos, F=0:5x8 puntos BF=1:Operación interna BF=0:Listo para instrucción											DDRAM: Display Data RAM CGRAM: Character Generator RAM AG: Dirección de CGRAM AD: Dirección de DDRAM (se corresponde con la posición del cursor) AC: Contador usado para direcciones de DDRAM y CGRAM	Los tiempos de ejecución son para 250KHz de frecuencia de operación

Tabla 7.- Lista de comandos del LCD

En esta tabla la DDRAM se corresponde con la memoria RAM que controla directamente los caracteres que salen en pantalla, correspondiéndose sus direcciones con las posiciones del display. La CGRAM en cambio es una memoria RAM para crear caracteres distintos a los que están predefinidos por defecto en el LCD.

Como vemos el abanico de instrucciones es muy amplio, permitiéndonos realizar un gran número de operaciones que simplifican el manejo del LCD. Es importante notar que tras la escritura de un carácter el cursor se incrementa automáticamente, por lo que no será necesario realizar manualmente los incrementos de DDRAM antes de enviar el nuevo carácter.