

Todo sobre el puerto I2C, su protocolo, tramas, conexiones y ejemplos

Dr. Rubén E-Marmolejo, Profesor de la Universidad de Guadalajara.

ruben.estrada@hetpro.com.mx

ruben.estrada@academicos.udg.mx

Guadalajara, Jalisco, México.

28 de Octubre de 2017

I2C es un estándar de comunicación serial, define la trama de datos y las conexiones físicas para transferir información entre dos dispositivos digitales. El puerto incluye dos cables de comunicación, **SDA** y **SCL**. Además el protocolo permite conectar hasta 127 dispositivos esclavos con esas dos líneas, con hasta velocidades de 100, 400 y 1000 kbits/s. También es conocido como IIC ó TWI - Two Wire Interface.

El protocolo I2C es uno de los más utilizados para comunicarse con sensores digitales, ya que a diferencia del [puerto Serial](#), su arquitectura permite tener una confirmación de los datos recibidos, dentro de la misma trama, entre otras ventajas.

La conexión de tantos dispositivos al mismo bus, es una de las principales ventajas. Además si comparamos a I2C con otro protocolo serial, como **Serial TTL**, este incluye más bits en su trama de comunicación que permite enviar mensajes más completos y detallados.

Los mensajes que se envían mediante un puerto I2C, incluye además del byte de información, una dirección tanto del registro como del sensor. Para la información

que se envía siempre existe una confirmación de recepción por parte del dispositivo. Por esta razón es bueno diferenciar a los distintos elementos involucrados en este tipo de comunicación.

Elementos en una transferencia por I2C

Siempre que hablamos de una comunicación oral, se entiende que es entre dos o más personas. Como consecuencia podemos también indicar que en una comunicación digital existen distintos dispositivos o elementos. En el caso de puerto I2C se diferencian dos elementos básicos, un **MAESTRO** y un **ESCLAVO**. La **Figura-1**, muestra una conexión típica de tres dispositivos, el bus consiste de dos líneas llamadas, **Serial DAta - SDA** y **Serial CLock - SCL**. Es decir, Datos Seriales y Reloj Serial. En particular al bus se le conectan dos resistencias en arreglo **pull-up**, de entre 2.2K y 10K.

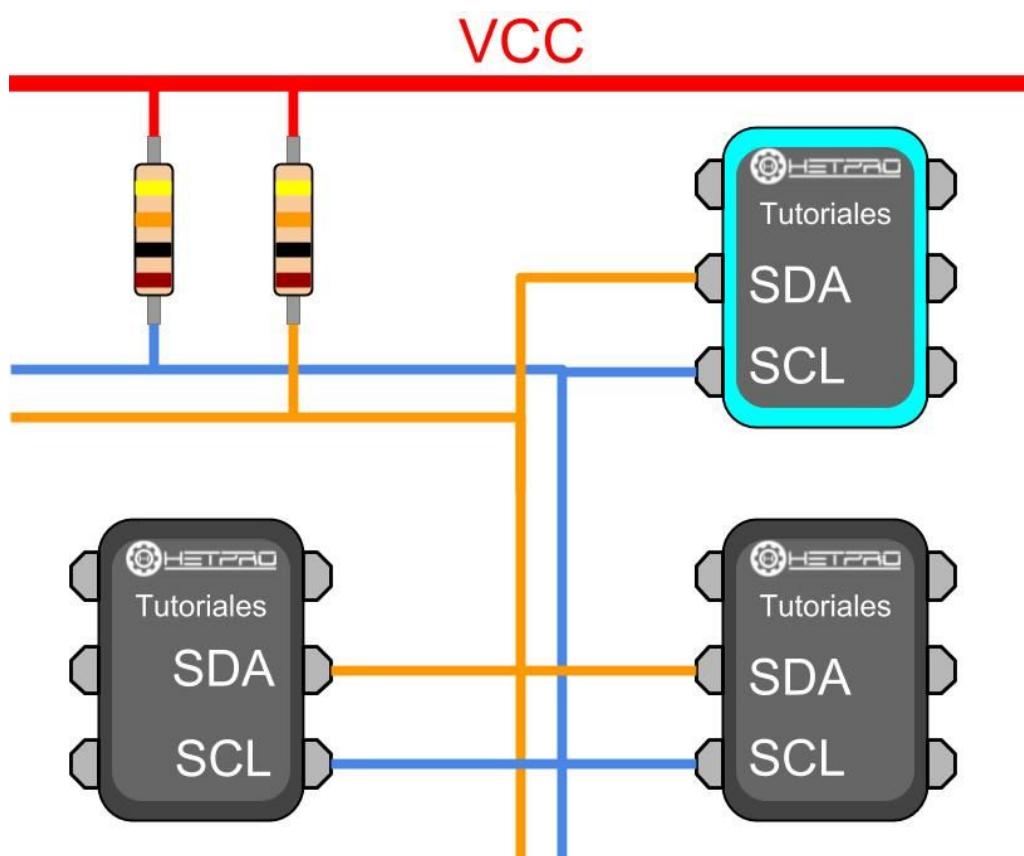


Figura-1. Conexión de tres dispositivos a un bus de comunicación I2C.

El **MAESTRO** I2C se encarga de controlar al cable de reloj, por sus siglas en inglés llamada SCL - Serial CLock. Además el MAESTRO se encarga de iniciar y parar la comunicación. La información binaria serial se envía sólo por la línea o cable de datos seriales, en inglés se llama SDA - Serial DAta. Dos Maestros no pueden hacer uso de un mismo protocolo I2C. Puede funcionar de dos maneras, como maestro-transmisor o maestro-receptor. Sus funciones principales son:

- Iniciar la comunicación - S
- Enviar 7 bits de dirección - ADDR
- Generar 1 bit de Lectura ó Escritura - R/W
- Enviar 8 bits de dirección de memoria
- Transmitir 8 bits de datos -
- Confirmar la recepción de datos - ACK - ACKnowledged
- Generar confirmación de No-recepción, NACK - No-ACKnowledged
- Finalizar la comunicación
-

El **ESCLAVO** I2C, generalmente suele ser un sensor. Este elemento suministra de la información de interés al MAESTRO. Puede actuar de dos formas: esclavo-transmisor ó esclavo-receptor. Un dispositivo I2C esclavo, no puede generar a la señal I2C. Sus funciones principales son:

- Enviar información en paquetes de 8 bits.
- Enviar confirmaciones de recepción, llamadas ACK

Protocolo de comunicación I2C

El protocolo de comunicación I2C se refiere al conjunto de bits que son necesarios para enviar uno o varios bytes de información. En lo particular, para este protocolo existen los siguientes bits importantes:

- Inicio ó Start - S
- Parada - P
- Confirmación - ACK
- NoConfirmación - NACK
- Lectura-/Escritura - L/W

- 7 bits para la dirección del dispositivo esclavo/maestro
- 8 bits de dirección (para algunos sensores pueden ser 16 bits)
- 8 bits de datos

El conjunto de estos bits y su orden va formando distintas tramas de comunicación.

Existen distintos modos de comunicación dependiendo del arreglo de estos bits. Tanto el maestro como el esclavo pueden o no generar los bits anteriores, según los modos de comunicación.

El bus I2C está disponible si las dos líneas, SDA y SCL están en un nivel lógico alto.

Modos de comunicación I2C

Los modos de comunicación en I2C se refieren a las distintas tramas que pueden formarse en el bus. Estas tramas o modos dependen de por ejemplo, si queremos leer al sensor esclavo, o si lo queremos configurar. Existen principalmente dos modos de comunicación:

- Maestro-Transmisor y Esclavo-Receptor. Este modo se usa cuando se desea configurar un registro del esclavo I2C.
- Maestro-Receptor Y Esclavo-Transmisor. Se usa cuando queremos leer información del sensor I2C.

Ejemplos-1 configuración de 1-registro

Ejemplo-1. Para diseñar registrador de datos, es necesario en algunas circunstancias usar un reloj de tiempo Real, RTC. Un **RTC** común es el **DS1307** que tiene un puerto **I2C** para poder leer y configurar el tiempo. La Figura-2, muestra la trama de datos necesaria para configurar la hora para que sean las 08 hrs. Para poder encontrar el sensor se requiere conocer su dirección. Para determinar la dirección se tienen 7 bits ($2^7 = 128$), de aquí que se puedan comunicar con dos líneas hasta 127 sensores, la dirección 0 es una llamada general.

Después se requiere un bit de lectura o escritura. Entonces, este bit acompaña a los 7-bits de dirección. Si el bit de lectura/escritura vale = 0, significa que se escribirá al esclavo. Si este bit vale = 1, significa entonces que se leerá. Cuando el Esclavo I2C recibe su dirección, es cuando este, responde con una confirmación (ACK). El

esclavo queda en espera de 8 bits de la memoria que se quiere escribir y responde con un ACK. Posteriormente el esclavo espera a que el maestro le envíe los 8-bits de datos correspondientes a la configuración del registro anterior y responde con un ACK. Finalmente el Maestro-I2C responde con un bit de fin de comunicación.

Configurar la hora 0x08 del registro 0x02 del RTC DS1307 que tiene la dirección 0x68



Figura-2. Trama de bits que se colocan en el puerto I2C para configurar el registro 0x02 del RTC-DS1307, con el valor de 0x08.



Ejemplo-2. Lectura del día RTC-DS1307 I2C

La Figura-3, muestra la trama que se requiere escribir en el puerto I2C para leer un byte del registro cuya dirección es 0x03. Este registro guarda el día del RTC-DS1307 cuya dirección para el puerto es 0x68. Para poder leer el byte, primero es necesario indicar al dispositivo esclavo que se quiere LEER la dirección 0x03. Una vez que el esclavo responde con el ACK, el Maestro entonces enviará un bit de Inicio-repetido seguido de la dirección pero con bit de Lectura. Después, el esclavo responde con el byte que contiene la memoria 0x03 y seguirá respondiendo con la información de la siguiente memoria (0x04) hasta que el MAESTRO dé por terminado la comunicación, esta vez con un NACK antes del bit de Paro.

Lectura de la memoria 0x03 del RTC DS1307 con la dirección 0x68, el valor regresado es el día 7 (Domingo).

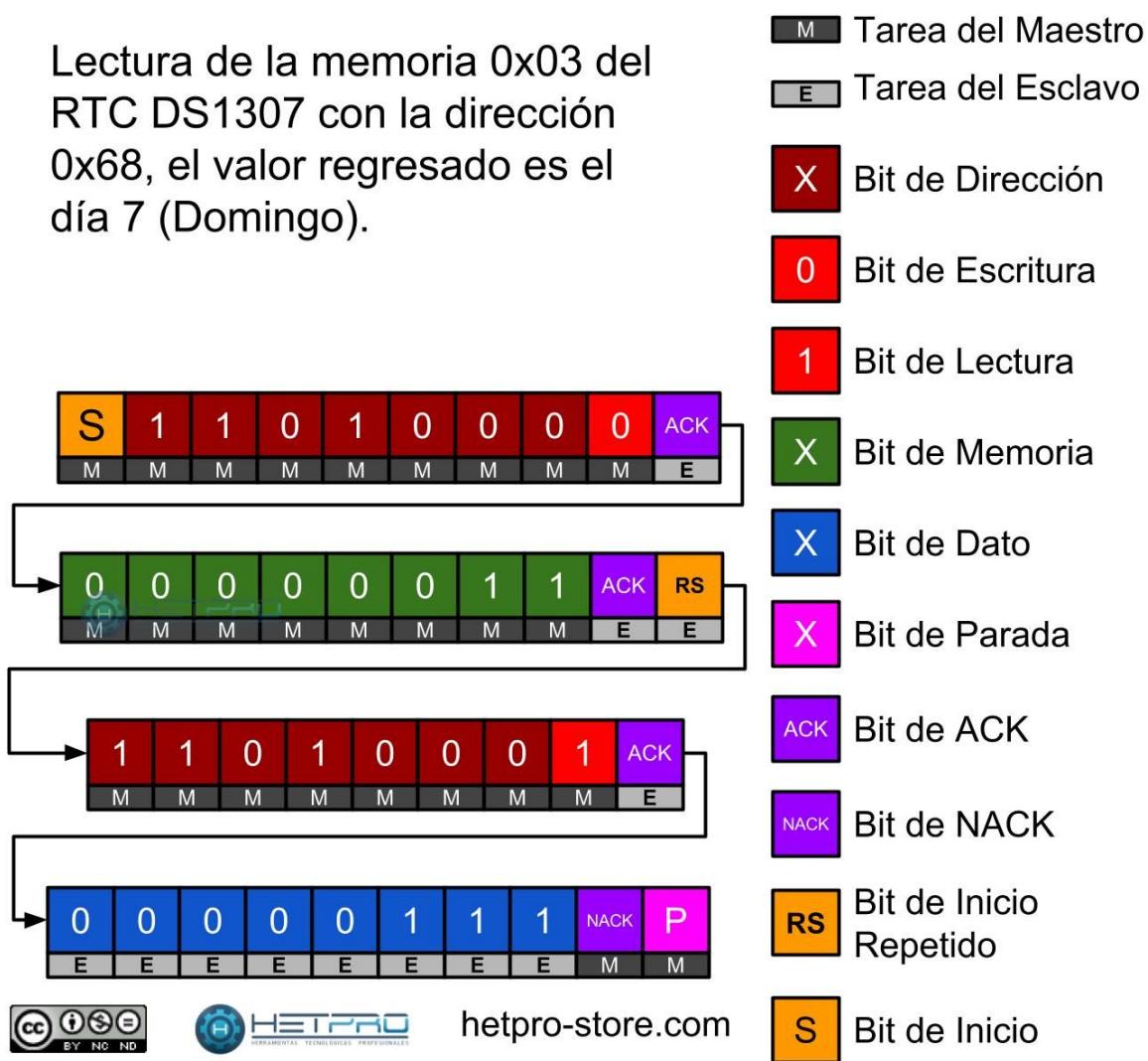


Figura-3. Trama de datos para el protocolo I2C para leer el registro 0x03 del DS1307. El RTC retorna el valor 0x07 que corresponde al día 7, es decir, Domingo.

Velocidad del puerto I2C

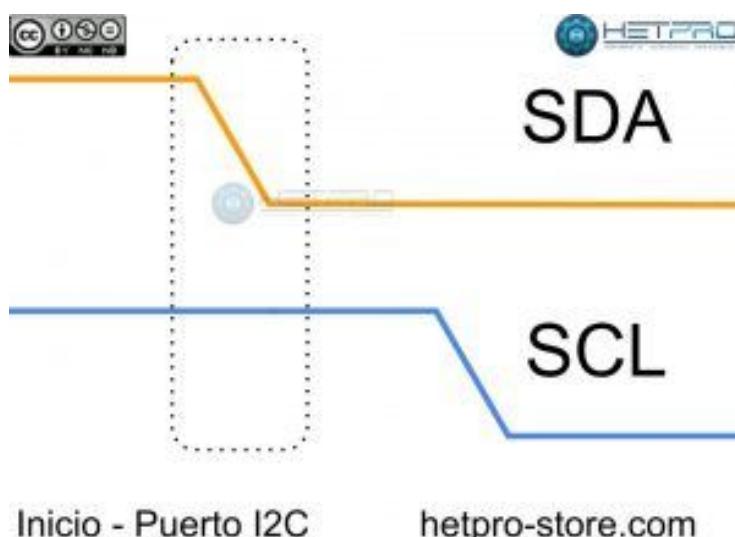
La velocidad del puerto I2C se refiere al tiempo que le toma al puerto transferir un bit de información. Entonces, este valor se mide en bits/segundo. Típicamente vamos a encontrar la referencia en ciclos/segundo o Herz. Existen tres velocidades estándar, 100Khz, 400Khz y 1Mhz, es decir, 100kbits/s, 400kbits/s y 1000kbits/s. Por ejemplo, para la trama de la Figura-2, ese paquete de datos tiene 29 bits, por lo que a una velocidad de 100kbits/s le tomaría al puerto 0.29 ms enviar la información.

El dato digital ó lógica que leerá cada uno de los dispositivos, corresponde el voltaje en los flancos de subida de la señal de reloj - SCL.

Condiciones eléctricas

Cada uno de los bits antes descritos (**Inicio-S**, **ACK**, **NACK**, Parada, **RS**) significan condiciones eléctricas en el bus I2C. Las condiciones de voltaje son las siguientes:

Inicio. La condición de inicio se genera cuando el bus está disponible, cuando mientras la línea de SCL está en alto (1), existe un flanco de bajada (un cambio de estado lógico de alto a bajo), en la línea de SDA). Este bit sólo lo puede generar el MAESTRO.



hetpro-store.com

Figura-4. Condición de INICIO-I2C. Estando el bus disponible, hay un flanco de bajada en la línea SDA.

Paro. La señal o bit de PARO se genera cuando hay un flanco de subida en la línea de datos, mientras que la línea de reloj se encuentra en alto. Esta condición sólo es posible generarla desde el MAESTRO.

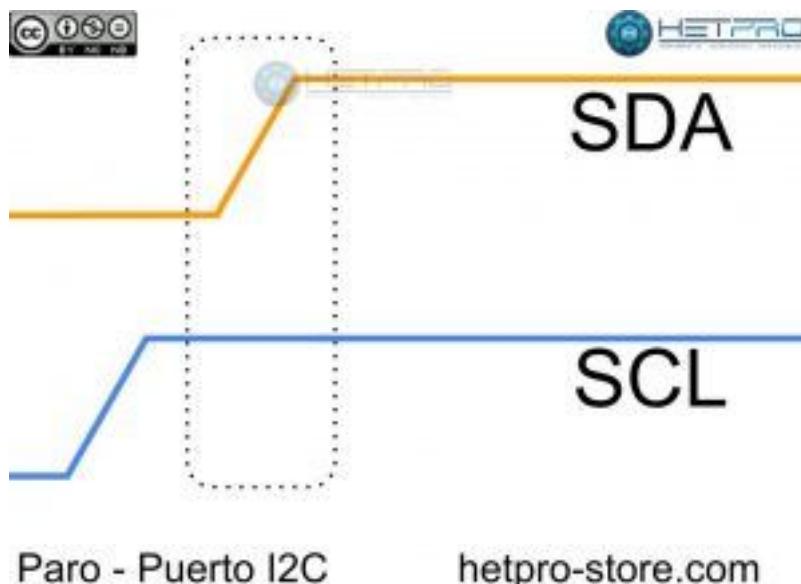


Figura-5. Condición de paro en el puerto I2C.

ACK. Confirmación de recepción. Esta condición se crea cuando estando la señal SCL en alto, SDA está en bajo. Esta señal la puede generar tanto el MAESTRO como el ESCLAVO.

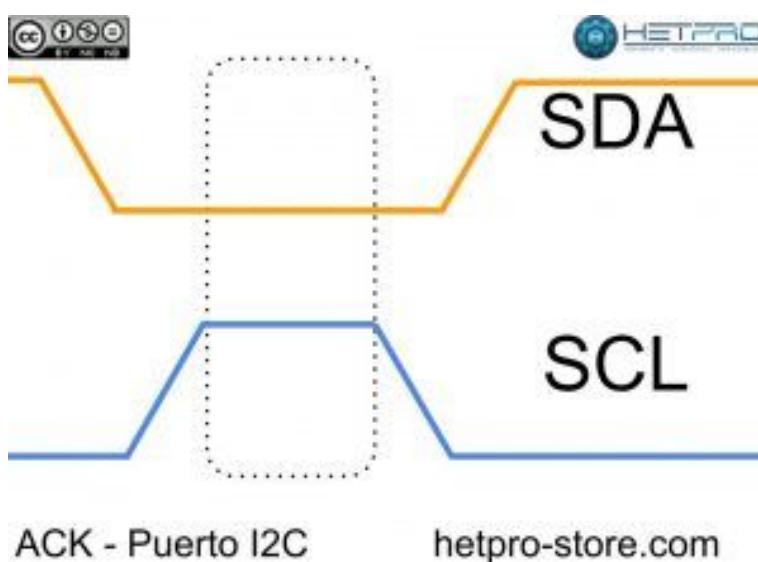


Figura-6. Condición o bit de ACK. Este bit se usa para confirmar la recepción.

NACK. Este bit es usado en el esquema de comunicación donde se leen varios bytes de un ESCLAVO en una sola transmisión. Entonces, el bit NACK se usa cuando ya no se quieren recibir más bytes. La condición sólo la puede generar el MAESTRO.

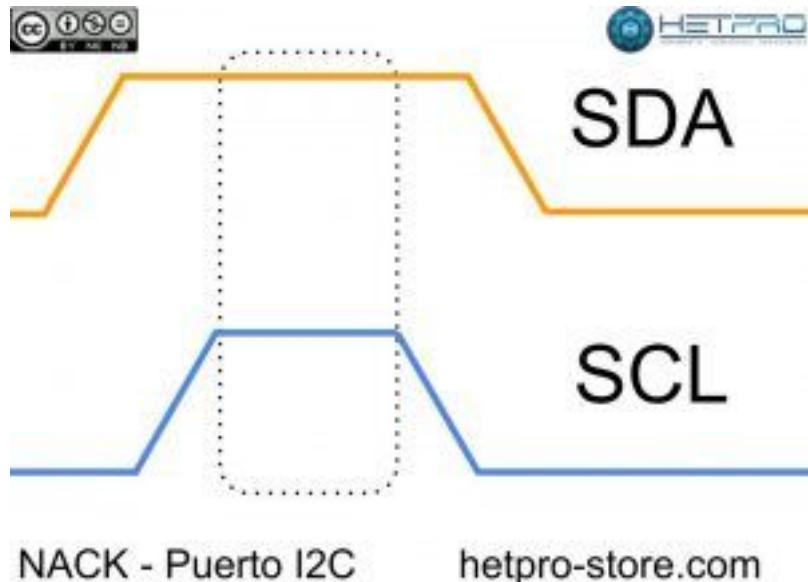


Figura-7. Condición NACK, sólo es generada por el MAESTRO-I2C.

Resumen

El puerto y protocolo I2C es un tema de mucha importancia para los ingenieros que desarrollan sistemas embebidos. Además de poder conectar hasta 127 sensores, el protocolo I2C permite manejar una comunicación segura entre dos dispositivos digitales. Como desventaja podemos mencionar el esquema de comunicación. Debido a que el esquema usado es UN-Maestro MUCHOS-Eslavos, esto puede generar una latencia en el envío y recepción de datos de cada uno de los esclavos. Esto sucede al que sólo podemos usar el bus I2C para comunicar un esclavo a la vez. Sin embargo, el puerto I2C es uno de los más utilizados en aplicaciones embebidas. Podemos encontrar sensores I2C en la mayoría de los dispositivos electrónicos digitales, desde televisores, celulares o laptops. En otro tutorial mencionaré ejemplos del uso de I2C con distintos lenguajes de programación, como Arduino, Mbed, ensamblador y C/C++.

Fuentes.

- [1] - <https://datasheets.maximintegrated.com/en/ds/DS1307.pdf>
- [2] - <https://www.nxp.com/docs/en/application-note/AN10216.pdf>