

# TEMA 8

## PROTOCOLOS DE COMUNICACIÓN HABITUALES EN INSTRUMENTACIÓN ELECTRÓNICA

La mayor parte de los sistemas de instrumentación están formados por distintos módulos que necesitan comunicarse entre sí. A la hora de definir la comunicación que se puede establecer entre los elementos de un sistema, hay que prestar atención a diversos aspectos que se enumeran a continuación:

1. **Distancia:** La comunicación puede darse entre dos elementos dentro de la misma placa, situados a pocos centímetros, o a decenas de metros de distancia. Podría plantearse, incluso, que el observador y el sistema estuvieran en dos países distintos.
2. **Cableada o inalámbrica:** En la mayor parte de los casos, la transmisión se realizará por cables aunque, algunas veces, bien por la presencia de obstáculos bien por la propia elección del diseñador, se realizará por medios inalámbricos. En este caso, la comunicación podría realizarse por infrarrojos o por radiofrecuencia. En este tema, sólo se estudiarán en detalle los protocolos de transmisión estrictamente ligados a la instrumentación electrónica con el objeto de no invadir el ámbito de otras asignaturas de la titulación.
3. **Carácter analógico o digital:** Dependiendo de las circunstancias, puede ser necesario transmitir la información por medio de una señal analógica (tensión o corriente) o una señal digital. En este último caso, también hay que fijarse en la presencia o ausencia de buses de datos.
4. **Destinatario:** En algunos casos, el receptor de la comunicación es un microcontrolador o FPGA que toma las decisiones necesarias sin necesidad de intervención externa. En otros, el destinatario es el propio usuario.

En conclusión, hay una gran cantidad de variables que han llevado a la aparición de protocolos de mayor o menor éxito en el mundo de la instrumentación. A la hora de elegir un protocolo apropiado para nuestro diseño, es conveniente realizar un balance de los siguientes aspectos y finalmente inclinarnos por una opción u otra:

1. **Complejidad del hardware:** Algunos protocolos de transmisión exigen numerosos cables y elementos de interfaz en tanto que otros utilizan buses de datos que simplifican el diseño. Esto nos lleva al siguiente punto:
2. Posibilidad de **bus de datos** y tamaño de éste.
3. **Complejidad del software:** Ésta es la contrapartida de los dos puntos anteriores. En general, cuanto mayor sea el número de cables más fácil de programar es el microcontrolador y menos sofisticados son los dispositivos esclavos.
4. **Velocidad** de transmisión de datos.
5. **Distancia máxima** para la transmisión de datos.
6. **Inmunidad** del protocolo **al ruido**.

7. Niveles de tensión o corriente implicados y, por tanto, **consumo de potencia**.
8. **Experiencia y tradición:** Cuando se diseña un sistema, nunca se parte de cero ya que se suele trabajar en un grupo con dilatada experiencia en algún protocolo. Esto hará que éste sea elegido con mayor frecuencia reforzando la condición de protocolo favorito. Asimismo, es posible que se disponga ya de material desarrollado de proyectos anteriores y que sea posible reutilizarlo en el nuevo diseño y así ahorrar tiempo.

Con estos datos, vamos a empezar a estudiar algunos de los protocolos más habituales en instrumentación electrónica. En primer lugar, abordaremos los protocolos cableados y, a continuación, los inalámbricos. En este grupo, distinguiremos los medios infrarrojos y los de radiofrecuencia.

## 1 Protocolos de transmisión con cable

### 1.1 El protocolo 4-20 mA

Este protocolo es clásico en la instrumentación electrónica que trabajará en ambientes industriales ruidosos, con muchas interferencias, y donde haya que transmitir señales a larga distancia. Su origen último se remontaría incluso a las líneas de telégrafo que recorrían continentes de punta a punta durante el siglo XIX. En este protocolo, una tensión  $V_X$ , definida dentro de un determinado rango  $[V_1, V_2]$  se transforma en una corriente en el intervalo  $[4, 20]$  mA. Esta transformación es lineal mediante la regla:

$$I_O(V_X) = 4 + \frac{16}{V_2 - V_1} \cdot (V_X - V_1).$$

Con lo que  $V_1 \equiv 4$  mA y  $V_2 \equiv 20$  mA. Este protocolo cuenta con muchas ventajas que lo han hecho muy popular en ambientes industriales:

#### Ventajas

- **Transmisión a larga distancia:** Por el principio de conservación de la carga, toda la corriente que entra por un cable tiene que salir íntegramente por el otro independientemente de la distancia. En la práctica, se puede enviar información a varios kilómetros de distancia aunque hay que tener en cuenta la caída de tensión en los cables. Si  $2 \cdot R_C$  es la resistencia parásita de los cables, en éstos se puede producir una caída de tensión de hasta  $40 \cdot R_C$ , con  $R_C$  expresada en  $k\Omega$ , valor que podría ser comparable a la tensión de alimentación del sistema.
- **Inmunidad a los efectos de impedancia del cable:** Otra vez por el principio de conservación de la carga, la información enviada como carga sale por el otro lado exactamente como entró. Esto diferencia a este protocolo de los basados en tensión, que se ven afectados por la caídas de tensión en los cables.
- **Inmunidad a interferencias:** A la luz de experimentos realizados con este protocolo, se constató que basta con trenzar los cables para eliminar por completo las interferencias y no es necesaria ninguna protección adicional.
- **Detección de fallos de línea:** En caso de corte accidental del cable o de interrupción de la señal de alimentación, la corriente transmitida sería 0 mA, fuera del rango admitido de valores, que es fácilmente detectable por el sistema. Ésta es la gran ventaja de este protocolo sobre otros parecidos como el 0-20 mA, que no permite detectar este tipo de problemas.

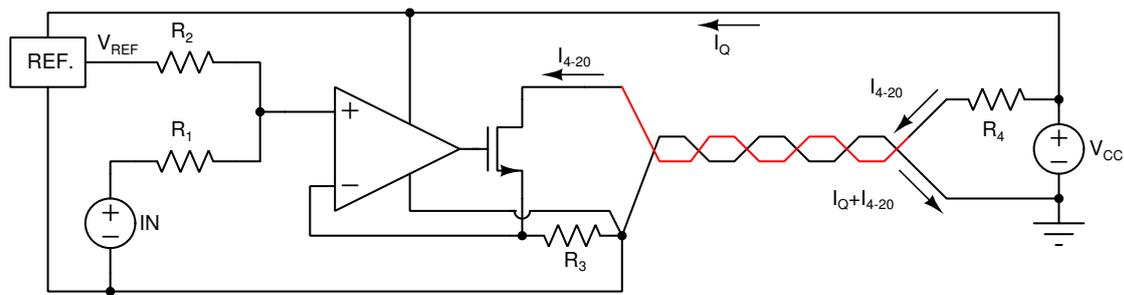


Figura 1: Montaje sencillo para realizar la transmisión de la señal con el protocolo 4-20 mA.  $I_Q$  es la corriente de alimentación de los distintos circuitos integrados que permiten la transmisión de la señal. En este diseño, se requieren TRES cables para realizar la conexión.

### Inconvenientes

- **Consumo** de corriente muy elevado.
- **Imposibilidad de creación de buses.** Cada conexión sólo admite un transmisor y un receptor cuyos papeles no pueden intercambiarse.

¿Qué circuitos permiten realizar la transmisión de la información con este protocolo? En principio, cualquier fuente de corriente controlada por tensión puede utilizarse para esta función. Por ejemplo, la clásica fuente que consta de un amplificador diferencial y un seguidor de tensión. Sin embargo, es más simple utilizar una fuente basada en un cascode activo como la que se muestra en Fig. 1.1. Es fácil ver que, en esta estructura, la corriente de salida  $I_{4-20}$  es:

$$I_{4-20} = \frac{1}{R_3} \cdot \left( \frac{R_1}{R_1 + R_2} \cdot V_{IN} + \frac{R_2}{R_1 + R_2} \cdot V_{REF} \right)$$

Eligiendo adecuadamente los valores, se puede ajustar la conversión de tensión a corriente a las directrices del protocolo. Las resistencias deben ser de precisión,  $V_{REF}$  debe provenir de una referencia de tensión y el transistor debería ser un NMOS o un JFET, no un bipolar pues aparecería un error de orden  $h_{FE}^{-1}$  ya que la corriente de colector no es exactamente la de emisor. Como receptor, bastaría una simple resistencia ( $R_4$ ) conectada en serie con la fuente de alimentación. La caída de tensión en la resistencia se puede medir con un amplificador de instrumentación.

Otra opción consiste en utilizar circuitos integrados especialmente preparados para transmitir señales de tensión con el protocolo 4-20 mA. Ejemplos de ellos son el XTR117 de Texas Instruments, AD5749 de Analog Devices, etc. En algunos casos, el transmisor es básico aunque, en otros, el fabricante incorpora más características para facilitar el diseño.

El protocolo 4-20 mA es, en general, un protocolo cuasiestático ya que no se pensó para transmitir señales muy rápidas. Podía ser utilizado, por ejemplo, para transmitir la lectura de una Pt100 a gran distancia y, lógicamente, ésta variará con bastante lentitud. Por ese motivo, se pensó que era factible superponer una señal digital a la transmitida en el protocolo 4-20 mA y transmitir ambas de manera simultánea a través del cable. Las señales serían fácilmente separables en el destino por medio de filtros u otras técnicas. Todo esto dio lugar a la aparición de un nuevo protocolo, llamado **HART** (*Highway Addressable Remote Transducer*), que permite la transmisión de señales digitales utilizando la instalación del 4-20. De tipo abierto, antiguamente era gestionado por la Fundación HART, ahora Field Comm Group. Este protocolo es, sin embargo, relativamente lento para los estándares actuales (1200 bps) aunque, por su robustez, se ha convertido en uno de los más habituales para transmisión digital en el ámbito industrial. Muchos de los conversores 4-20 mA actuales permiten agregar la señal HART con facilidad.

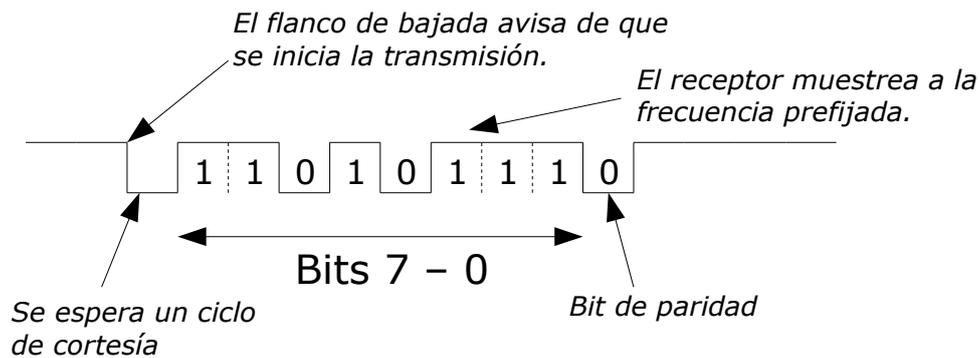


Figura 2: Tren de bits definido para la transmisión de señal RS-232.

## 1.2 El protocolo RS-232

Es este un protocolo de transmisión de datos digitales relativamente antiguo que ha pervivido en el tiempo gracias a su facilidad de uso y escasa demanda de recursos. Así, sigue siendo uno de los más populares en sistemas de instrumentación que no requieran una gran velocidad de transmisión y la distancia no sea excesiva. Puesto que para poder utilizarlo todo dispositivo tiene que disponer de una UART o USART (*Universal (Serial) Asynchronous Receiver Transmitter*), los términos UART y USART se utilizan como sinónimos de RS-232.

### Características generales

- **Número de dispositivos:** Dos. Uno será el emisor y otro el receptor. Los roles pueden intercambiarse según las necesidades de la transmisión.
- **Número de cables:** Este protocolo requiere tres cables. Uno para transmitir información del dispositivo A al B, otro cable para transmitir del B al A y, finalmente, un tercer cable para unir las tierras.
- **Niveles lógicos de tensión:** Este protocolo se caracteriza por definir el «0» lógico como cualquier tensión superior a +3 V e inferior a +12 V y el «1» como cualquier tensión entre -12 V y -3 V. Las tensiones intermedias no pueden ser interpretadas correctamente. Al elegir tensiones tan elevadas se evitan malinterpretaciones de valores en caso de que exista una caída de tensión significativa en el cable de tierra en la transmisión a larga distancia. Para transformar los niveles lógicos TTL en niveles RS-232 y viceversa, se pueden utilizar dispositivos como el MAX232 en los que se utilizan capacidades conmutadas y una fuente de alimentación de +5 V. Hay momentos, sin embargo, en los que esta transformación no es necesaria. Por ejemplo, si se pretende realizar la comunicación entre dos microcontroladores dentro de la misma placa. En este caso, la distancia es muy pequeña y no es necesaria la conversión de niveles.
- **Velocidad de transmisión y alcance:** La velocidad de transmisión es, por defecto, 9600 bps aunque es posible aumentarla hasta 115000 bps. Cuanto mayor sea la velocidad menor es la distancia aunque es posible alcanzar sin dificultad una distancia de 15 m.
- **Definición del tren de bits:** Ésta es una de las características más estrictas del protocolo (Fig. 2). En primer lugar, el protocolo es asíncrono y la recepción correcta sólo está

garantizada si el receptor conoce de antemano la frecuencia a la que está transmitiendo el emisor. Por otra parte, el protocolo sólo permite la transmisión de bytes, palabras de 8 bits.

En ausencia de transmisión, el emisor fija la señal del cable a un valor ALTO («1») que permanece indefinidamente hasta que se debe enviar un byte. En este caso, se pone el cable a «0» y se espera un ciclo de reloj. A 9600 bps, esta espera es de 1,04 ms.

El flanco de bajada avisa al receptor que debe prepararse para recibir datos y que tiene un ciclo de reloj para comenzar a leer. Cuando ha transcurrido ese ciclo de reloj, el emisor empieza a enviar los bits de menor a mayor significado con la frecuencia de transmisión prefijada. En el otro extremo, el receptor recoge los datos muestreando la señal a la misma frecuencia. Tras enviar los datos, en algunos casos, se puede mandar el bit de paridad calculado a partir del byte para certificar la corrección de la transmisión. Finalmente, el sistema vuelve a poner la señal a «1» a la espera de volver a enviar la información.

Es evidente que las frecuencias de trabajo de receptor y emisor deben ser prácticamente iguales. Sin embargo, el corto número de bits enviados por mensaje (8) hacen la transmisión factible incluso presentándose leves variaciones entre las frecuencias de trabajo. Así, la transmisión sería exitosa incluso con variaciones de la frecuencia del 12,5%. Sin embargo, la buena calidad de los osciladores actuales hace que este margen de desviación sea mucho más que suficiente para una buena transmisión.

Otra gran ventaja de este protocolo es que puede ser utilizado sin mayor dificultad en los ordenadores personales. Así, en sistemas Microsoft Windows® se pueden usar programas como Hyperlink<sup>1</sup>, o PuTTY para escribir y leer desde el PC del usuario. En sistemas GNU Linux, el puerto RS-232 (también llamado «serie») corresponde al fichero `/dev/tty0` o similar, etc., y existen interfaces gráficas que ayudan al control (p.e., PuTTY o GTKTerm). Por supuesto, entornos como LabView® de National Instruments permiten controlar este puerto.

El mayor problema es, sin embargo, que la mayor parte de los PCs carecen de puerto RS232. Por ello, es necesario utilizar adaptadores RS232 a USB para poder utilizar estos puertos (Fig. 3). Sin embargo, este hecho no tienen mayor dificultad ya que, una vez reconocidos por el sistema operativo, no hay diferencia entre un puerto RS232 real y otro emulado. Todos estos detalles han facilitado que un protocolo relativamente obsoleto aún se siga utilizando en sistemas de instrumentación.

En muchos dialectos del lenguaje C destinado a la programación de microcontroladores, carece de sentido suponer que va a haber una pantalla y un teclado. Por ese motivo, algunas funciones convencionales como `printf()`, destinada a escribir por pantalla; `getc()`, destinada a recoger un carácter del teclado, etc. se modifican para tener el puerto serie como origen/destino. Así, en un microcontrolador PIC programado con CCS la instrucción

```
printf("Hola mundo\n")
```

enviará esta frase directamente al puerto serie y se mostrará tal cual en el monitor del puerto serie (p.e., HyperTerminal).

### 1.3 Los protocolos RS422 y RS485

El protocolo RS422 surgió por la necesidad de aumentar el alcance y la velocidad de transmisión del protocolo RS232. El mayor problema de éste era que los niveles lógicos se definían valores de tensión absolutos de tal modo que en el caso de que hubiera una tensión del modo común muy alta iba a ser muy difícil distinguir los niveles lógicos entre sí. Así, desde la tierra del emisor se podrían emitir señales de  $\pm 3$  V para definir los niveles lógicos pero, si el modo común era, por ejemplo, +6 V, al receptor llegarían valores de tensión de 3 y 9 V con lo que todos los datos se tomarían como «0».

<sup>1</sup>Presente en versiones antiguas de este sistema operativo.



Figura 3: Adaptador de puerto USB a RS232 (a). Detalle del conector macho DB9 para RS232 (b). Los pines están numerados aunque, en la práctica, basta con usar los pines 2 y 3 (TX y RX).

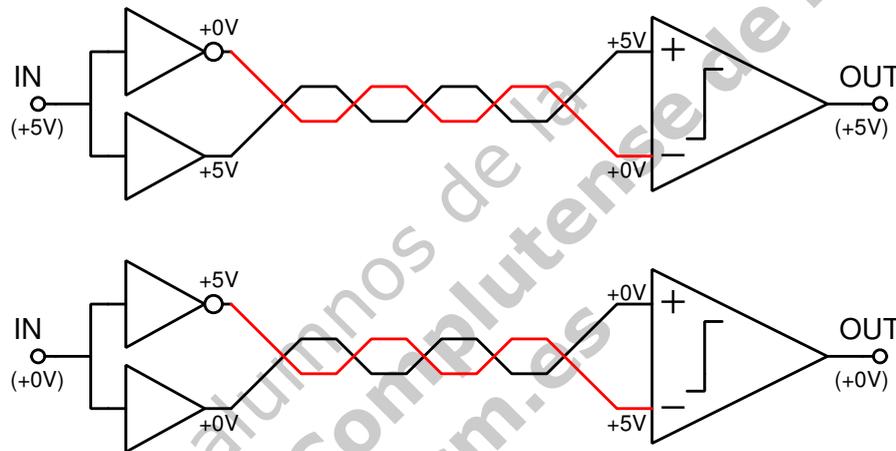


Figura 4: Transmisión RS422 de «1» y «0».

En el protocolo RS422, este problema se resuelve enviando las señales en modo diferencial (Fig. 4). La transmisión de señales en un determinado requiere dos cables cuya diferencia de tensión es la que indica si se ha enviado un «0» o un «1». Si hay que enviar respuesta, se requiere otro par de cables más dirigidos en sentido opuesto.

A diferencia del protocolo RS232, no existe ninguna convención para construir el tren de bits. Cada diseñador de circuitos puede definir el protocolo que le convenga aunque, en la práctica, nada impide utilizar el tren de bits definido en el RS232. Por otro lado, dado que la tensión se transmite en modo diferencial, no tiene sentido cambiar las tensiones de los niveles lógicos. No hay ningún problema en transmitir tensiones de 0 V y +5 V o en cualquier otro nivel que se esté utilizando.

Contando con un simple par de cables trenzados, este protocolo permite alcanzar distancias de 1200 m y velocidades de transmisión de 10 Mbit/s. El problema es que es imposible alcanzar ambos logros al mismo tiempo. En la práctica, las transmisiones a máxima velocidad son impensables más allá de 10 o 15 m. *Grosso modo*, se puede tomar como regla de trabajo que el producto de la velocidad máxima en Mbits/s por la distancia en metros es del orden de 100. Así, transmitir a 1200 m sólo es factible si la velocidad se reduce a 50-100 kbit/s.

El protocolo RS485 introduce una mejora sustancial al protocolo RS422 pues permite crear buses de datos en los que hasta 32 (o 256) dispositivos pueden transmitir señal a lo largo de un único par de cables. Se tienen que terminar las líneas con resistencias de  $120 \Omega$  para realizar acoplamientos correctos y evitar reflexiones a alta frecuencia. Este protocolo es

lo suficientemente robusto como para ser utilizado en áreas como la aviónica donde se exige robustez, larga distancia y una velocidad de envío de datos no muy alta.

## 1.4 El protocolo SPI

Este protocolo, *Serial Peripheral Interface*, nació para superar al protocolo RS232. A semejanza de éste, se concibió para trabajar a distancias cortas y con facilidad de uso. El objetivo era mejorar la baja velocidad de transmisión y que un único maestro pudiera controlar varios dispositivos esclavos.

### Características

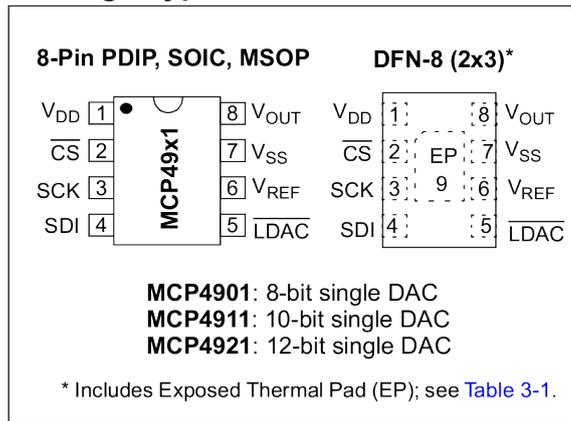
- **Sincronía:** La señal de reloj proviene directamente del maestro. Éste elige la frecuencia de transmisión que requiera
- **Cableado:** En este protocolo, se definen cuatro líneas:
  - SCK: Señal de reloj proveniente del maestro. Cada vez que esta señal se activa, se indica al esclavo que debe leer un nuevo bit, o bien escribirlo si está enviando datos.
  - CS: *Chip Select*. También se puede denominar «SS» o «*Slave Select*». Ésta es la entrada de habilitación propia de cada dispositivo. Es posible que sea una entrada negada y que sólo habilite al dispositivo esclavo al pasar a estado de salida bajo.
  - MOSI: Acrónimo de «*Master-OUT-Slave-IN*». Es la línea de datos que utiliza el maestro para enviar datos al esclavo. En muchos casos, el pin del esclavo destinado a recibir datos es el SDI (*Serial Data Input*). Un ejemplo clásico de receptor de datos es un DAC.
  - MISO: Acrónimo de «*Master-IN-Slave-OUT*». El maestro recibe datos de los esclavos a través de esta línea. En los esclavos, este pin se suele llamar SDO «*Serial Data Output*».

En todo sistema debe haber al menos tres líneas: SCK, CS y al menos una de las dos líneas de datos. Si hay un maestro y  $N$  esclavos, hay que usar bien  $N + 2$  líneas, bien  $N + 3$ : SCK,  $N$  CS y MISO y/o MOSI. Lógicamente, no se deben habilitar varios esclavos a la vez para evitar problemas en las líneas. Sólo responde el dispositivo que se haya activado y los demás permanecen en latencia mostrando salidas de alta impedancia.

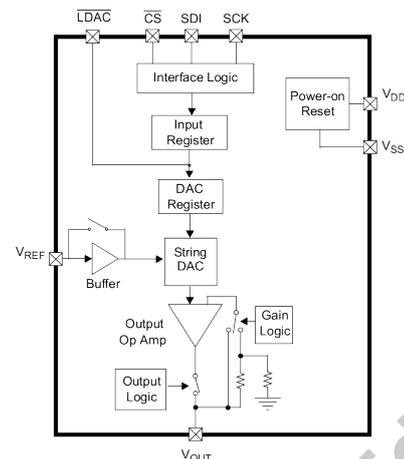
- **Tamaño de palabra:** A diferencia de otros protocolos, las palabras que se envíen no deben ser forzosamente de 8 bits. Así, es posible enviar palabras de 10, 12, 16 bits, etc. Otra cosa es que el receptor, que puede ser un microcontrolador, tenga que añadir bits adicionales de relleno para poder trabajar con los datos. En todo microcontrolador, las palabras tienen 8, 16 o 32 bits.
- **Bidireccionalidad:** El maestro y el esclavo establecen una comunicación cruzada aunque siempre gobernada por la señal de reloj del maestro.
- **Niveles de tensión:** Se usarán los niveles naturales, 0 V para «0»,  $V_L$  para «1».
- **Distancia:** Este protocolo no permite transmisiones fiables a más de un metro de distancia. Cuanta mayor distancia, menor velocidad.

Este último dato puede plantear una seria duda acerca de la utilidad de este método. Sin embargo, el protocolo SPI se va a utilizar en general en el interior de las placas. En este caso, las distancias son muy cortas y es aquí donde las ventajas superan con creces a los

## Package Types



(a)



(b)

Figura 5: Patillaje (a) y estructura de bloques (b) del DAC MCP4911. Imagen tomada de la hoja de características del fabricante.

inconvenientes. Más aún, junto con algunos de los protocolos que veremos a continuación, es el gran dominador en la comunicación dentro de las placas.

Veamos un par de ejemplos para comprender mejor el protocolo. En primer lugar, el DAC MCP4911, fabricado por Microchip® y con 10 bits de resolución (Fig. 5). Este chip será gobernado por un microcontrolador que destinará tres pines para la transmisión SPI: CS, SCK y SDI. Los tres pines estarán configurados como salida del microcontrolador.

Los datos son transmitidos según se muestra en Fig. 6. Los pasos que se van a dar son los siguientes:

1. En primer lugar, se habilita el dispositivo poniendo CS a nivel bajo. Estará así durante toda la transmisión.
2. A partir de ahí, el microcontrolador colocará valores de bit en SDI e indicará al DAC que debe leerlo llevando la señal de reloj a «1». En este instante, el DAC guardará el dato y esperará al siguiente ciclo de reloj.
3. Los cuatro primeros bits que se hayan enviado tienen funciones especiales de configuración y así lo entenderá el DAC. Sólo a partir del quinto el dispositivo comenzará a grabar el dato que debe mostrar en la salida. En este caso, los bits que llegan antes serán los más significativos.
4. En teoría, sólo sería necesario enviar 14 bits. Sin embargo, lo más normal es que el microcontrolador haya enviado una variable de 16 bits o dos de 8. Por este motivo, al dispositivo llegan dos bits adicionales cuyo valor es irrelevante pues no son tenidos en cuenta por el convertor.

Hay que decir, finalmente, que el protocolo SPI puede estar sujeto a algunas modificaciones sobre la definición elemental. Pongamos, por ejemplo, el convertor analógico-digital MAX1132 de Maxim Integrated® (Fig. 7). Para controlar este dispositivo, se necesitan cinco conexiones. Así, además de las señales CS y SCK, el convertor necesita una línea adicional para recibir los bits de configuración (DIN, equivalente a SDI), otra para enviar los datos de la conversión digital (DOOUT, equivalente a SDO) y otra línea adicional (SSTRB) para indicarle al maestro que, tras ser configurado el ADC, la conversión ha finalizado y que procede a enviar los datos.

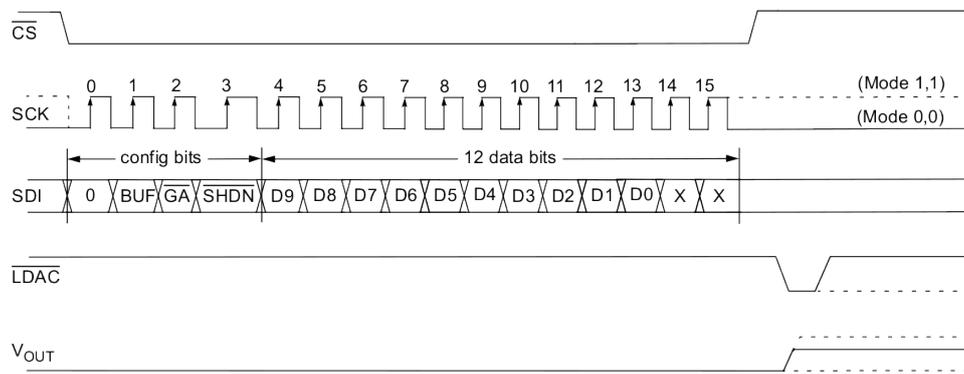


Figura 6: Secuencia de señales SPI para el control del DAC MCP4911. Imagen tomada de la hoja de características del fabricante.

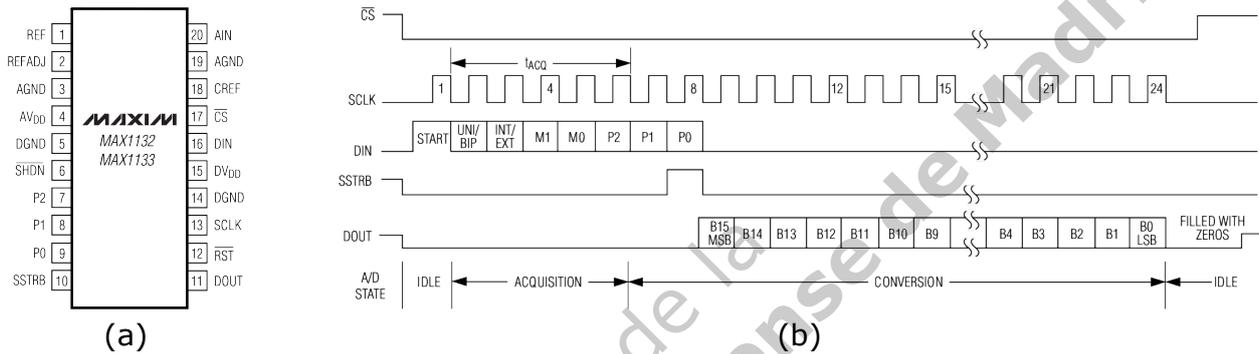


Figura 7: Patillaje (a) y secuencia de señales SPI para el control (b) del ADC MAX1132. Imágenes tomada de la hoja de características del fabricante.

El uso del protocolo SPI en microcontroladores es relativamente sencillo. Veamos el caso de las placas Arduino y similares (la familia MSP430, de Texas Instruments, programables con Energía) cuentan con la librería SPI.h para programar las transmisiones de datos. Hay que tener en cuenta algunos detalles técnicos muy simples pero que pueden impedir una transmisión correcta si no se cuidan. Un fallo muy común tiene que ver con la estructura de datos pues para recibir datos de un ADC de 14 bits hay que destinar una variable de 16 bits, con dos bits de relleno. Luego, estos tendrían que ser descartados mediante, por ejemplo, un desplazamiento de bits de dos posiciones hacia la derecha.

### 1.5 El protocolo I<sup>2</sup>C

A pesar de las grandes ventajas del protocolo SPI atendiendo a la velocidad de transmisión de los datos, puede ser inviable en caso de que existan muchos esclavos por el gran número de conexiones requeridas. En este caso, una alternativa para la transmisión de datos entre componentes es el protocolo I<sup>2</sup>C (*Inter-Integrated Circuits*), que fue presentado por Philips Semiconductors® (hoy NXP Semiconductors®). Este protocolo también se denomina TWI (*Two-Wire Interface*) en algunos textos para evitar conflictos legales con los desarrolladores por el uso del término. Todos los detalles sobre este protocolo están recogidos en <http://www.i2c-bus.org/>.

### Características

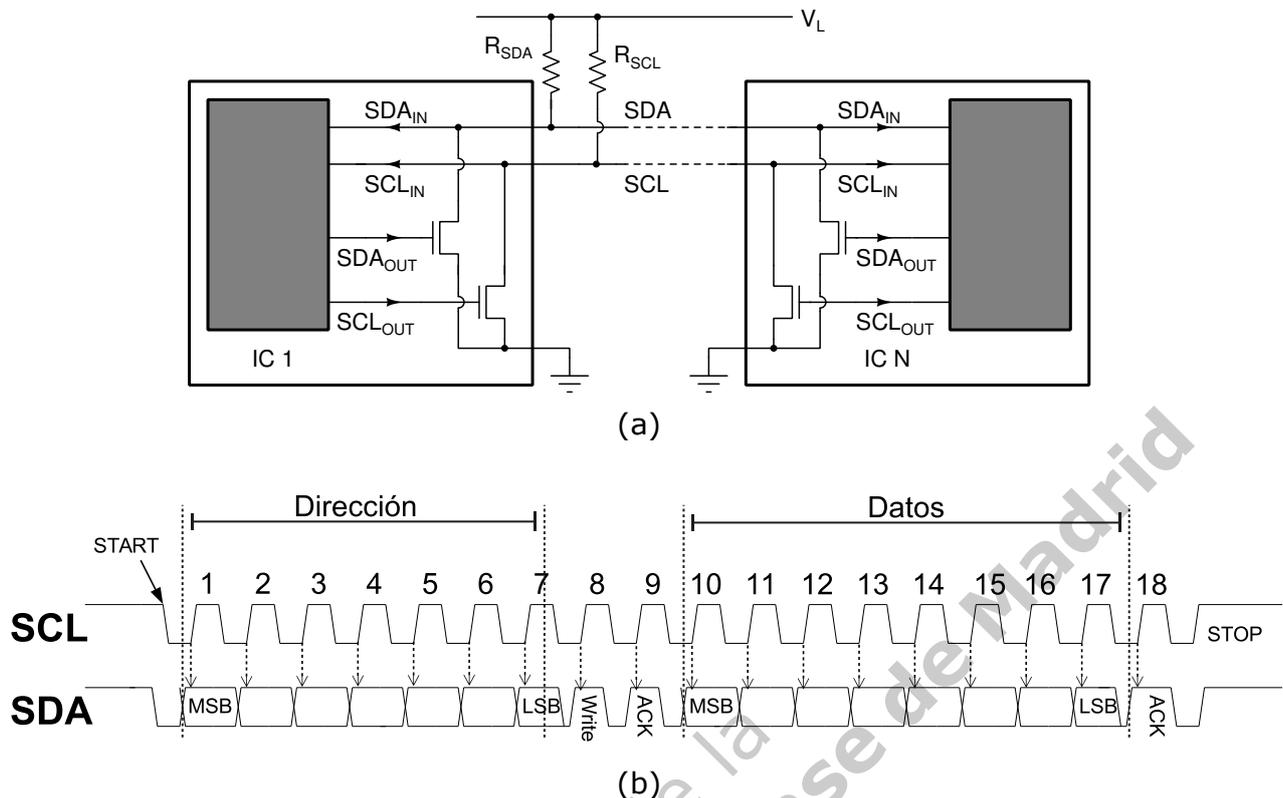


Figura 8: Disposición de líneas (a) y secuencia de control (b) del bus I<sup>2</sup>C. No se han incluido las capacidades parásitas de los cables, de gran importancia para el máximo alcance de la señal.

- Líneas de transmisión:** Sólo dos (Fig. 8a). La primera, SCL (*Serial Clock*), será la señal de reloj proveniente del maestro. La otra, SDA (*Serial Data*), será la que se utilice para las transmisiones. Ambas señales se deben conectar a la tensión de alimentación mediante resistencias de *pull-up*.  
 En este protocolo, los dispositivos, sean el maestro o los esclavos, deben conectarse a las líneas SCL y SDA por un terminal que esté en estado de alta impedancia. Así, la tensión lógica en reposo de ambos buses es  $+V_L$  y será interpretada como un «1» lógico.
- Frecuencia de trabajo:** La marcada por SCL. Depende de la distancia de trabajo y, sobre todo, de la velocidad de respuesta de los dispositivos. No hay que confundirla con la velocidad (efectiva) de transmisión de datos pues un solo byte requiere el envío de hasta 18 bits.
- Alcance:** El protocolo I<sup>2</sup>C se concibió para transmisión de señales dentro una única placa. Sin embargo, es posible alcanzar distancias de varios metros de longitud (se habla en algunos textos de casi un centenar) usando cables con baja capacidad parásita y disminuyendo la frecuencia de transmisión.
- Direccionamiento:** Cada elemento esclavo tiene asignada una dirección única de 7 bits que lo identifica dentro del bus. Idealmente, se podrían direccionar  $2^7 = 128$  elementos distintos pero, al reservarse 16 direcciones para funciones especiales, sólo se pueden direccionar 112 dispositivos en el bus. Esta dirección va a estar generalmente grabada en el interior del sensor. En algunos casos puede cambiarse exteriormente (p. e., el sensor inteligente LM73 elige una de tres posibles direcciones grabadas según se conecte a tierra, alimentación o se deje al aire un determinado pin). Si lo desea, el alumno puede consultar la web <https://learn.adafruit.com/i2c-addresses/the-list> para conocer algunos sensores con salida I<sup>2</sup>C y cuál es la dirección que se les asigna en la fábrica.
- Tren de bits:** Fig. 8b muestra el tren de bits característico del bus. Corresponde al caso

más sencillo, que es el de un esclavo que envía información al maestro cuando éste lo demanda. En reposo, todos los elementos, cuyas salidas son de drenador abierto, se encuentran en estado de alta impedancia y la tensión de los buses es  $V_L \equiv "1"$ .

- La transmisión comienza a requerimientos del maestro poniendo SDA a 0 V y haciendo que el reloj SCL baje a 0.
- En cada uno de los 7 ciclos de reloj siguiente, el maestro indicará la dirección del esclavo elegido y, tras enviarlo, irá a estado de reposo tras indicar en el último momento con un bit alto que «escriba» la lectura en el bus. Se transmitirá primero el bit más significativo.
- A continuación, el esclavo dará a entender que ha recibido la instrucción («ACKnowledgment») y procederá a enviar el dato de 8 bits, controlando él mismo el reloj ya que el maestro se encuentra en estado de alta impedancia. Como en el caso de la dirección, el primer bit es el más significativo.
- Una vez terminado, el maestro indicará con la señal ACK que ha recibido la información y todos los elementos vuelven a estado de reposo.

Como vemos, se ha necesitado el envío de 18 bits para transmitir un único byte, además de la pausa inicial correspondiente al arranque de transmisión. El diálogo entre elementos puede ser incluso más complicado. Por ejemplo, una memoria FRAM como la FM24V10 de Cypress Semiconductor con entrada I<sup>2</sup>C necesita una comunicación más compleja cuyos detalles aparecen en la hoja de características del dispositivo.

Por otro lado, aunque todos los ciclos de reloj de Fig. 8b tienen el mismo periodo, en la práctica los ciclos pueden diferir ya que esta señal puede provenir de distintos dispositivos.

- **Tamaño de palabra:** El byte es la unidad básica de envío de información de este protocolo. Esto lo hace rígido como el RS-232 y con peores resultados que el SPI, que es perfectamente configurable.

No es difícil crear un maestro capaz de controlar los dispositivos esclavos con este protocolo. Así, la máquina de estados del bloque que controlaría este protocolo es fácil de implementar en lenguajes como VHDL o Verilog y exportarlo a FPGAs. Asimismo, la mayoría de los microcontroladores disponen de pines específicos para ser utilizados en este protocolo y los dialectos de C u otros lenguajes disponen de librerías que permiten una programación sencilla: Librería Wire en Arduinos, python-smbus e i2c-tools en Raspberry PI, etc. Es posible utilizar el PC como maestro adaptando el puerto RS-232.

## 1.6 El protocolo 1-Wire

Este protocolo es menos popular que cualquiera de los otros dos anteriores aunque su uso está bastante extendido. Fue desarrollado por Dallas Semiconductors®, actualmente parte de Maxim Integrated®<sup>2</sup>. En este protocolo, se consigue establecer la comunicación utilizando un único cable (además de tierra y, aunque no siempre, de la alimentación positiva). Este protocolo se caracteriza por usar lógicas de colector/drenador abierto como el I<sup>2</sup>C combinadas con el asincronismo del RS-232. El bus de transmisión está normalmente a «1» al estar todos los elementos en estado de alta impedancia y la transmisión se inicia con un flanco de bajada proveniente del maestro con una duración de 480  $\mu$ s. Que se transmita un «0» o un «1» depende de la duración del pulso: inferior a 15  $\mu$ s equivale a «1», superior a 60  $\mu$ s equivale a «0».

Este protocolo exige direccionamiento como el I<sup>2</sup>C. Sin embargo, es de 64 bits lo que permite no sólo la numeración de elementos sino la presentación al maestro comunicando la

<sup>2</sup><https://www.maximintegrated.com/en/app-notes/index.mvp/id/1796>

función que realizan y otros parámetros adicionales. Este protocolo es el que permite menores velocidades de transmisión dada su baja frecuencia de trabajo y la longitud de su tren de bits aunque permite transmitir a mayores distancias que los otros dos competidores. Además, cuenta con la ventaja de que los dispositivos pueden identificarse según se vayan conectando al bus. En algunos casos, no es necesario alimentar el dispositivo esclavo pues puede almacenar energía en condensadores a partir de alguna de las líneas de bus.

Como en el caso del I<sup>2</sup>C, puede usarse el puerto RS-232 (y, en general, cualquier dispositivo con UART) para que un PC reciba datos desde una red de dispositivos. Asimismo, existen librerías específicas en C para implementar este protocolo en microcontroladores.

## 1.7 Protocolos avanzados

En los apartados anteriores, se han descrito diversos protocolos que se utilizan, o se han utilizado, para la construcción de sistemas de instrumentación. Todos se caracterizan por la posibilidad de transmisión a larga distancia, robustez, fácil implementación en el sistema, etc. Sin embargo, en algunos casos estos protocolos de transmisión no pueden satisfacer completamente las necesidades del desarrollador. Por ejemplo, no existe posibilidad de realizar una comunicación sencilla con el PC pues, aunque nominalmente algunos protocolos pueden hacerlo, todos se basan en el RS-232, que es bastante lento. Por este motivo, es necesario utilizar otros protocolos avanzados y, en general, más generalistas.

### 1.7.1 Protocolo USB

Este protocolo serie (*Universal Serial Bus*) permite transmisión a gran velocidad con el PC aunque tanto la velocidad como la distancia depende de la versión específica del protocolo (actualmente, 3.0). Es bidireccional aunque sólo permite un esclavo por puerto. Este puerto está disponible en la mayoría de los PCs y en muchos microcontroladores. El problema es que la programación de microcontroladores con este dispositivo no es tan sencilla como las de los otros con lo que, en muchos casos, se prefiere simplemente utilizar un adaptador USB/RS-232, USB-SPI, etc. En cualquier caso, nada impide su uso si el diseñador del sistema lo considera oportuno.

### 1.7.2 Protocolo CAN

Las siglas corresponden a *Controller Area Network* (Bus). Este protocolo permite la creación de un bus con muchos maestros y esclavos con sólo dos cables de conexión. Este protocolo es muy robusto, tolerante a fallos y permite velocidades altas de transmisión lo que le ha hecho muy interesante en el mundo de la automoción. Es un protocolo que utiliza dos cables cuya diferencia de tensión marca el valor del bit lógico: «0», si está entre +1,5 y +3 V, «1» si está entre -120 y +12 mV. La gran ventaja de este código es que no sólo permite direccionamiento de esclavos sino que se puede dar mayor o menor importancia a los mensajes de los esclavos con los llamados «bits de arbitrio», que resuelven conflictos producidos si dos dispositivos inician la comunicación de manera simultánea.

Su implementación es también relativamente avanzada y muchos microcontroladores carecen de terminales adecuados. En su lugar, es necesario utilizar circuitos integrados de interfaz como el MCP2515 y el MCP2561, que permiten la creación de un nudo de la red CAN tras la conversión de datos SPI a este formato.

### 1.7.3 Protocolo GPIB

Éste es el acrónimo de *General Purpose Instrumentation Bus* y también se lo conoce como IEEE-488. A diferencia de los protocolos anteriores, está pensado para conectar dispositivos de instrumentación complejos (voltímetros, osciloscopios, fuentes controladas, etc.) a un PC



(a)



(b)

Figura 9: Detalles de un cable GPIB: Cable completo (a) y detalles de la parte externa del conector (b).

maestro. Es un protocolo bastante antiguo cuyas primeras versiones se remontan a los años 60. Sin embargo, ha sobrevivido y sigue gozando de relativamente buena salud por:

- **Facilidad de programación:** Cada dispositivo tiene una dirección grabada y las suites de instrumentación electrónica como LabView permiten reconocerlos con seguridad, transmitir órdenes y recibir datos.
- **Mantenimiento de dispositivos de calidad:** Los dispositivos electrónicos para cuyo control se concibió el GPIB son muy caros y por eso deben durar en el tiempo. No es extraño, por tanto, que los laboratorios sigan utilizando dispositivos de hace 10-20 años que aún son perfectamente funcionales. Como estos dispositivos se controlaban con GPIB al no haber nacido aún otros protocolos, el protocolo se debe mantener en tanto que los dispositivos no hayan quedado obsoletos.
- **Cables endurecidos:** Una de las características del protocolo es el uso de cables muy robustos, con conectores atornillados, que los hacen muy adecuados para ambientes de trabajo relativamente hostiles (Fig. 9).

Desde el punto de vista técnico, el bus GPIB es de tipo paralelo, permite la conexión de hasta 15 dispositivos a una distancia no superior a 20 m y a una velocidad de 8 MB/s. Como la mayor parte de los elementos actuales, tanto los PCs como los esclavos, carecen de tarjeta de adquisición de datos por motivos de espacio, se comercializan adaptadores USB/GPIB que permiten la emulación de este puerto.

#### 1.7.4 Ethernet

Muchos microcontroladores actuales son suficientemente potentes como para que pueda implementarse en ellos todos los protocolos basados en Ethernet: TCP/IP, UDP, SSH, etc. Al disponer de direcciones IP y MAC, se puede interactuar directamente con el microcontrolador.

Dado que la frontera entre microcontroladores y microprocesadores se está diluyendo con el paso de los años, algunos microprocesadores en los que se está ejecutando un sistema operativo pueden realizar funciones propias de los microcontroladores. Pensemos, por ejemplo, en los ARM Cortex dentro de pequeños ordenadores como las *Raspberries Pi* y con entradas GPIO compatibles con SPI e I<sup>2</sup>C aunque sin ADCs integrados.

Además, muchos aparatos de instrumentación modernos como voltímetros u osciloscopios disponen de tarjeta de red que permite su control a distancia (incluso, a centenares o miles de kilómetros). La llegada de este protocolo ha sido la causa del retroceso del GPIB.

## 2 Protocolos de transmisión inalámbricos

En muchos casos, el sistema no puede utilizar cables para poner en comunicación sus componentes. Los motivos pueden ser variados: Pudiera ser que existan barreras físicas que impidieran lanzar cables de conexión, porque se buscara una movilidad que no esté limitada por los cables, por ahorro de material, etc. En estas circunstancias, hay que establecer conexiones de tipo inalámbrico. Este tipo de transmisión tiene problemas como la atenuación con la distancia, la posibilidad de interceptación, etc.

Dentro de esta familia de protocolos, hay que distinguir dos grandes grupos: Las transmisiones por infrarrojos y las de radiofrecuencia. Sus ámbitos de aplicación se solapan en algunos casos.

### 2.1 Transmisión por infrarrojos

Éste es el protocolo que se utiliza en muchos mandos a distancia utilizados en televisiones, sistemas de climatización, control de luces, etc. No es casualidad que su uso esté restringido al interior de recintos cerrados.

En general, se deben tener en cuenta varios puntos de interés:

- **Longitud de onda:** La emisión suele estar en torno a los 950 nm o, en general, en alguna de las longitudes de onda en las que el vapor de agua presente un máximo de absorción y así eliminar gran parte del ruido procedente de la luz solar.
- **Modulación a 38 kHz:** El tren de bits que se emita debe estar modulado a una frecuencia entre 30 y 56 kHz (generalmente, 38 kHz) tal y como se mostró en el Tema 4. El emisor será un LED infrarrojo y el receptor debe tener ajuste de ganancia. Esto hace que la velocidad de transmisión de bits esté limitada a no más de 1 bit/ms. Esto es, sin embargo, más que suficiente para los sistemas habituales.
- **Distancia:** Por motivos de potencia, no es habitual que se transmita la información a más de 10 m de distancia.
- **Legislación:** Al tener un alcance tan limitado, su uso se restringe a espacios reducidos donde el sistema no va a interferir con los de la vecindad y, por tanto, nos encontramos en un ambiente totalmente privado donde no tiene sentido establecer una legislación que organice su uso. El diseñador puede utilizar este protocolo sabiendo que no violará ninguna ley.

Por definición, dado que la transmisión inalámbrica equivale formalmente a una única línea cableada y, en consecuencia, no pueden coexistir una señal de reloj y otra de datos. Por tanto, todas las comunicaciones por infrarrojo son formalmente asíncronas. Dentro de esta transmisión, hay que distinguir asimismo dos aspectos: Cómo se indica que el valor de cada bit y cómo se constituye el tren de bits.

### 2.1.1 Indicación del valor del bit

En principio, nada impide realizar la transmisión de los bits tal y como salen del microcontrolador modulando directamente el tren de bits y transmitiéndolos. Así, por ejemplo, podríamos sacar un dato por el puerto RS232 de un microcontrolador con el aspecto mostrado en Fig. 2, modularla a 38 kHz, transmitirla, demodularla y llevar la señal recibida al receptor de la UART de un microcontrolador. Sin embargo, en la práctica se suelen usar otros sistemas de codificación que han demostrado mayor robustez y fiabilidad que la transmisión propiamente dicha.

- **Codificación Bifase o Manchester:** Esta codificación indica que el bit es un «0» o un «1» en función de que aparezca un flanco descendente o ascendente en el punto medio de cada ciclo (Fig. 10a). En caso de que haya que transmitir dos veces seguidas el mismo bit, hay que realizar una transición previa al inicio del ciclo. En la práctica, esto es equivalente a convertir cada «0» en el par de bits «10» y cada «1» en «01», doblando la frecuencia de transmisión de bits. Hay que decir que este convenio no es universal pues en algunos sistemas la interpretación es exactamente la contraria. Su uso no está restringido a la transmisión por infrarrojos y goza de bastante popularidad para realizar la transmisión de datos en internet. Este sistema de codificación goza de varias ventajas:
  - Es posible determinar de manera simple la frecuencia de transmisión de bits, pues el periodo es la separación mínima entre dos flancos de subida inmediatos. Esto permite una sincronización automática entre emisor y receptor.
  - Al codificarse los datos como flancos de subida y de bajada, no importan los niveles absolutos de tensión. Esto permite eliminar el acoplamiento DC ya que la tensión del modo común no interviene. Sin embargo, esta ventaja carece de interés en sistemas de transmisión en infrarrojo pues los sistemas están normalmente aislados entre sí.

Lamentablemente, este sistema tiene diversas desventajas que deben indicarse:

- Al realizarse la detección a mitad de periodo y poder existir pulsos de medio periodo de transmisión de duración, si un sistema puede transmitir pulsos a una determinada velocidad, la velocidad de transmisión de bits es, al menos, la mitad. En otras palabras, se desperdicia ancho de banda.
  - Desde el punto de vista práctico, la máquina de estados o el código de programación para la decodificación de este protocolo es uno de los más difíciles de implementar.
- **Codificación por anchura de pulsos:** En esta codificación, los dos valores se implementan como pulsos de distinta duración (Fig. 10b). Es, por tanto, uno de los sistemas más sencillos de implementar ya que basta con un contador tanto para transmitirlo como para recibirlo. No se puede hablar de una velocidad fija de transmisión ya que todo depende del número de «0» y «1» que haya en el mensaje.
- **Codificación por distancia de pulsos:** En este caso, los pulsos tienen una duración fija y lo que distingue cada valor es la distancia entre pulsos (Fig. 10c). Igualmente, no se puede hablar de velocidad de transmisión y también es fácilmente implementable con un simple contador.

En general, se pueden elegir los protocolos que se deseen al haber total libertad de uso. En muchos casos, un protocolo ha adquirido popularidad simplemente porque los primeros desarrolladores lo utilizaron y el resto de competidores se limitaron a seguir las directrices para hacer sus dispositivos compatibles con los del primer desarrollo.

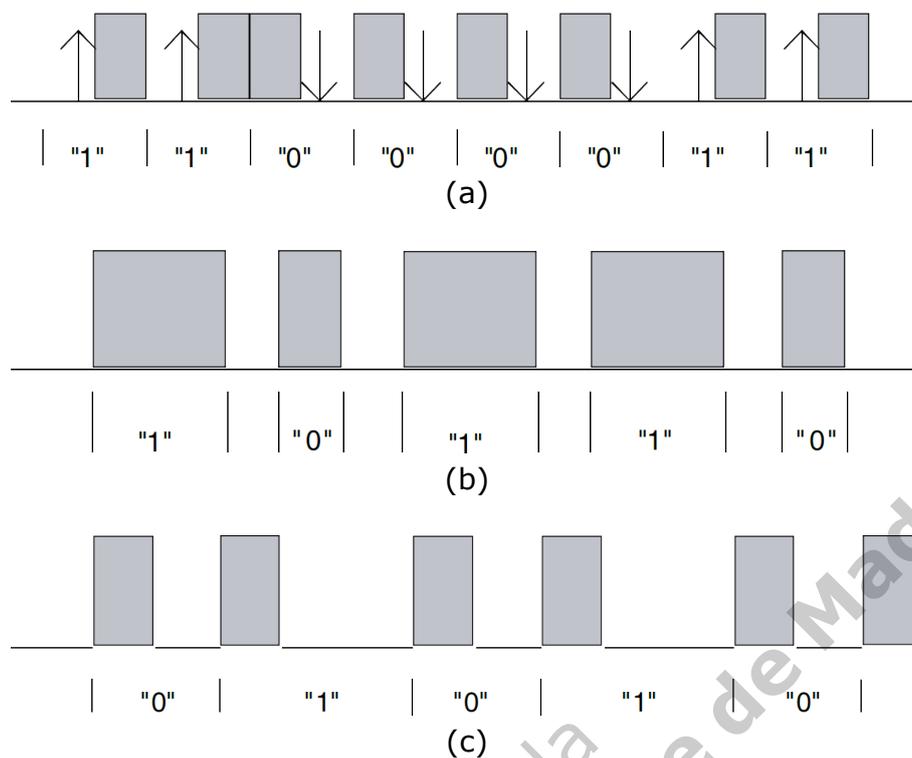


Figura 10: Codificación de bits en comunicación infrarroja: Manchester o bifase (a), por longitud de pulso (b) y por distancia entre pulsos (c). Imágenes tomadas de "Data Formats for IR Control", disponible en <http://www.vishay.com/>.

### 2.1.2 Construcción del tren de bits

En general, todos los protocolos de transmisión deben indicar en su tren de bits tres cosas: En primer lugar, que se ha iniciado la transmisión. A continuación, hay que indicar a qué dispositivo se está enviando el mensaje y, finalmente, el mensaje propiamente dicho.

Hay muchísimos protocolos y variaciones de los mismos, que pueden consultarse en la web SB Projects<sup>3</sup>. Además, cada diseñador puede establecer el que desee atendiendo a sus necesidades particulares. Describiremos, a continuación, un par de protocolos de amplio uso que se pueden tomar como ejemplos didácticos.

- **RC-5:** Desarrollado por Philips, factor que le ha dado gran popularidad en Europa (Fig. 11). Se caracteriza por utilizar codificación tipo Manchester y una portadora a 36 kHz que, por error, algunos desarrolladores no vinculados a Philips han cambiado a la frecuencia convencional de 38 kHz. Los bits se transmiten cada 1,778 ms, tiempo que corresponde a 64 ciclos a 36 kHz. En cada mensaje, se deben transmitir los siguientes segmentos:
  - *Bits de inicio:* Se envían dos bits a «1» al inicio de cada transmisión. De este modo, se avisa al receptor de la transmisión y, además, puede realizar una sincronización de datos si fuera necesario.
  - *Toggle bit:* Éste es un bit cuyo valor es el opuesto del que se haya enviado en la transmisión anterior. El cambio se produce cada vez que se pulsa una tecla. De este modo, el receptor puede saber si se está enviando un nuevo mensaje o si se está repitiendo el mismo.
  - *Dirección:* Cada receptor se identifica con un código único de 5 bits, que se inician con el bit más significativo. Esto implica que se pueden identificar hasta 32 dispositivos diferentes. Normalmente, no se realiza una distinción individual sino que cada

<sup>3</sup><http://www.sbprojects.com/knowledge/ir/index.php>

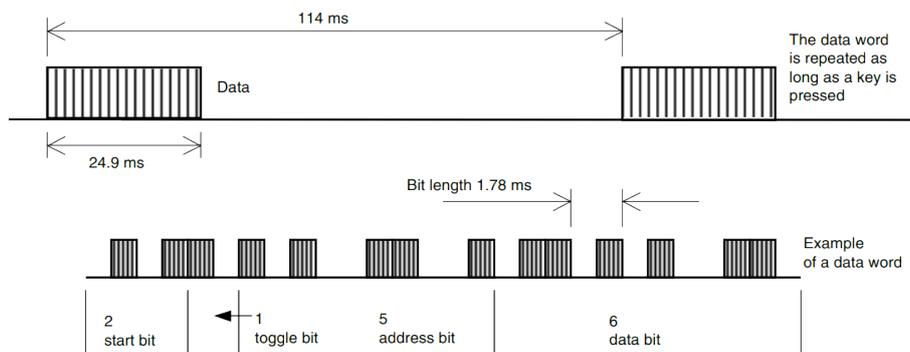


Figura 11: Tren de bits en el protocolo RC-5. Imágenes tomadas de "Data Formats for IR Control", disponible en <http://www.vishay.com/>.

modelo, y no cada dispositivo, tiene una identificación diferente. Así, todas las televisiones tienen la misma dirección, los vídeos otra, los DVD otras, etc. De este modo se consigue que los mandos de dos televisores distintos del mismo fabricante sean intercambiables pero que no valgan para controlar un DVD.

- *Mensaje*: Codificado con 6 bits. Lógicamente, son propios de cada modelo.

Como el mensaje requiere 14 bits, la duración total será de 24,9 ms. Si el usuario mantiene pulsado el botón, el mensaje se sigue transmitiendo cada 114 ms sin cambiar, por supuesto, el *toggle bit*. Este periodo de silencio es el necesario para enviar 64 bits ( $64 \times 1,778 \approx 114$  ms).

- **NEC**: Fue desarrollado por NEC, actualmente Renesas, una empresa japonesa líder que ha hecho que este protocolo sea muy popular en este país. En él, la codificación de los bits se realiza por distancia entre pulsos (Fig. 12). Se requiere una portadora de 38 kHz y cada pulso tendrá una duración de  $560 \mu\text{s}$ . Si la distancia entre flancos de subida de dos pulsos distintos es de 1,125 ms, el bit se interpretará como un «0» lógico. Si es 2,25 ms, tendremos un «1». El tren de pulsos consta de los siguientes elementos:
  - *Pulso de inicio*: No es un bit propiamente dicho. Se transmite un pulso de 9 ms de duración seguido por un silencio de 4,5 ms. El objetivo es dar aviso del inicio de transmisión y ayudar al ajuste de ganancia del receptor.
  - *Dirección*: El dispositivo se identifica con 8 bits, comenzando por el bit menos significativo. Por tanto, pueden direccionarse hasta 256 dispositivos.
  - *Dirección negada*: Otros 8 bits, con los valores de dirección invertidos.
  - *Datos*: Otros 8 bits, cuya función depende por supuesto del dispositivo receptor y de las funciones que pueda realizar. Por tanto, pueden transmitirse hasta 256 órdenes distintas.
  - *Datos negados*: En este momentos, se envía el byte de datos con los valores invertidos.

¿Por qué se envían dos veces tanto la dirección como los datos? Simplemente, porque conseguimos un poco de redundancia y porque en el conjunto de 16 bits de dirección más dirección invertida (o de datos) garantizamos que haya siempre 8 bits a «1» y otros 8 a «0» con una duración constante de  $8 \cdot 1,125 + 8 \cdot 2,25 = 27$  ms. Por tanto, cada mensaje durará  $9 + 4,5 + 27 + 27 = 67,5$  ms. Si se mantiene el botón pulsado, se transmite cada 110 ms una señal de repetición, que consiste en un pulso de 9 ms, un silencio de 2,25 ms y un pulso rápido de  $560 \mu\text{s}$ .

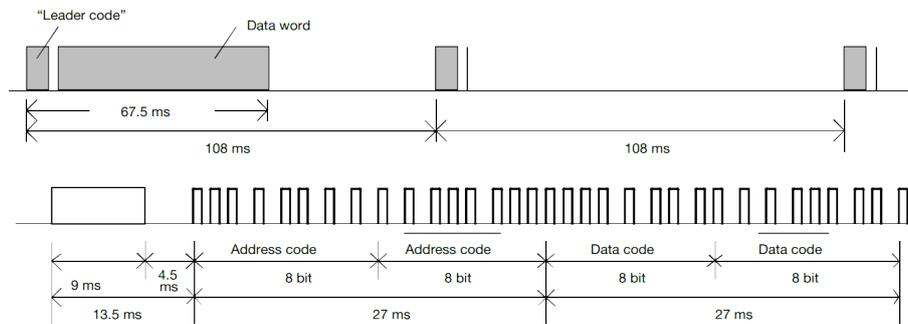


Figura 12: Tren de bits en el protocolo NEC. Imágenes tomadas de "Data Formats for IR Control", disponible en <http://www.vishay.com/>.

Los protocolos anteriores son básicos y han sufrido cambios a lo largo del tiempo para adaptarlos a nuevos escenarios. Existen, además, otros protocolos que no se han descrito aquí por falta de espacio. En algunos casos, las especificaciones son públicas pero en otros el desarrollador ha decidido mantener en secreto. Al hacer tecnología inversa, se ha podido descifrar el protocolo pero, por supuesto, nada garantiza que esta descripción sea del todo fidedigna.

## 2.2 Protocolos de transmisión basados en radiofrecuencia

A diferencia de los protocolos basados en infrarrojo, los protocolos basados en radiofrecuencia están sujetos a la legislación de los distintos países. Así, la transmisión debe hacerse en la zona de los 2,4 MHz, que suele reservarse para uso libre aunque respetando las reglas de organización. En general, estos protocolos distan de ser sencillos pues ya estaríamos hablando de redes de comunicación perfectamente organizadas, donde se deben realizar una identificación de elementos, protección de la información enviada, etc. Los protocolos más habituales son:

- **ZigBee:** O, más correctamente, IEEE 802.15.4. En realidad, se habla de un conjunto de protocolos sujetos a variaciones que pueden impedir la correcta comunicación entre dispositivos aparentemente similares. Se caracteriza por la presencia de un único coordinador que se comunica con un número indeterminado de *routers* siendo estos los que se comunican con los dispositivos finales. En definitiva, nos encontraríamos ante una red tipo árbol.

Este protocolo se caracteriza por alcanzar una distancia relativamente corta (100 m en las mejores circunstancias, normalmente mucho menos) y por una velocidad de transmisión de 250 kB/s. El material necesario es extremadamente sencillo. Así, es posible encontrar módulos como el XBee<sup>4</sup> que transforman en Zigbee datos que van a y vienen del puerto RS232 de un microcontrolador (Fig. 13). Por otra parte, el consumo de estos módulos es extremadamente bajos ya que cuentan con la opción de caer en latencia. Todo esto ha hecho que sea uno de los protocolos más utilizados y con mayor porvenir en el campo de la domótica.

- **Bluetooth:** Un protocolo que no requiere presentación al estar presente en la mayoría de los teléfonos móviles, ordenadores, tabletas, etc. Su nombre correcto es IEEE 802.15.1 y contempla una estructura de un maestro y varios esclavos. También utiliza la zona en torno a los 2,4 GHz y se caracteriza por realizar saltos en la frecuencia de transmisión. Permite una velocidad de transmisión más alta que el ZigBee (De 1 a 32 Mbit/s dependiendo de la versión del protocolo) y un alcance similar aunque depende enormemente de la potencia de la antena emisora. Su consumo es bastante elevado en comparación

<sup>4</sup><https://www.digi.com/xbee>

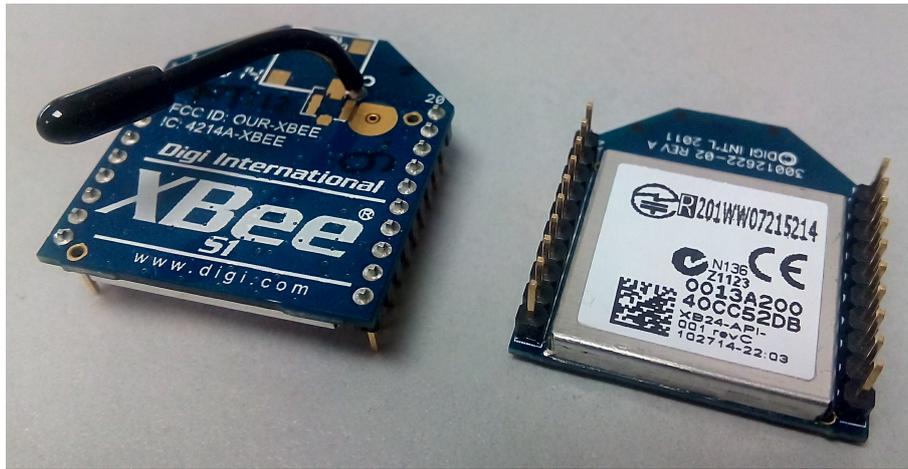


Figura 13: Ejemplos del modulo XBee para la conexión ZigBee.

con el ZigBee, hecho que junto con que la programación de dispositivos para trabajar en esta red sea complicada, ha restringido su uso práctico a sistemas donde se necesite una velocidad demasiado alta para el protocolo ZigBee.

- **WiFi:** Equivalente inalámbrico del protocolo Ethernet descrito en la sección 1.7.4, con las mismas ventajas e inconvenientes. Evidentemente, es el protocolo más avanzado y permitiría controlar el sistema a larguísimas distancias. Sin embargo, su configuración es difícil y no suele utilizarse más que en casos contados.

Como puede intuirse, este apartado hace que se solape la asignatura con otras de la titulación con lo que se remite al estudiante a éstas.