

Tras haber analizado el formato de las señales de los principales telemandos, hemos diseñado un pequeño decodificador de uso genérico utilizando un microcontrolador PIC.

# DECODIFICADOR RF CON USB

ALESSANDRO SOTTOCORNOLA



**D**espués de conseguir interpretar los códigos de algunos de los principales telemandos presentes en el mercado, hemos pensado aplicar estos conocimientos para realizar un decodificador, basado en PIC, capaz de deco-

dificar todas las señales analizadas. Las posibles aplicaciones van, por ejemplo, desde aprovechar la señal identificativa del control remoto para pilotar salidas normalmente gestionadas por un ordenador, al control de accesos para

para puertas o más en general para cerraduras electrónicas. En nuestro caso hemos querido presentar un proyecto más simple pero de uso didáctico y general: realizar un visualizador de código, o un sistema que permite visuali-

zar en el ordenador el código aplicado en el dip-switch del control remoto y/o sensor. Este sistema podría ser muy útil por ejemplo para negocios que



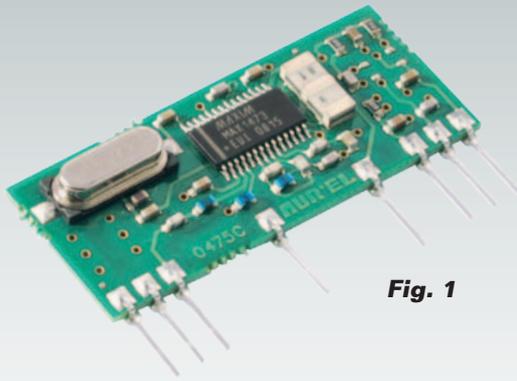
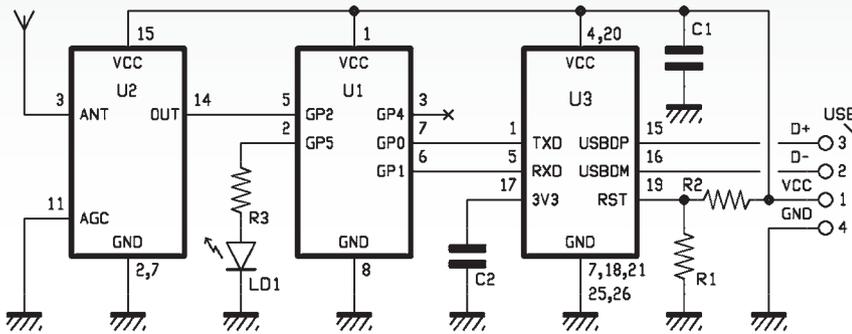


Fig. 1

Tabla 1

Pin 1	N.C.	-
Pin 2 - 7	Ground	Conexiones a GND
Pin 3	Antenna	Conexión para la antena, impedancia 50 ohm.
Pin 11	AGC On-Off	Habilita el control automático de la ganancia. Nivel lógico bajo: control activado Nivel lógico alto: control desactivado. Siempre máxima sensibilidad
Pin 13	RSSI OUT - Test Point	Salida RSSI proporcional a la potencia de la señal en entrada
Pin 14	Uscita dati	Salida de datos receptor. Normalmente bajo en ausencia de RF en entrada
Pin 15	+V <sub>s</sub>	Conexión a la alimentación

## [esquema ELÉCTRICO]



efectúan duplicados de control remoto porque permite identificar en pocos segundos si el control remoto original puede entrar en codificación conocida o no. También debe tenerse en cuenta que el sistema presentado funciona en la banda de frecuencia de 433 MHz

pero nada impide sustituir el módulo RF por uno de diferente frecuencia, por ejemplo si el propio control remoto o sensor utiliza la banda de 868 MHz es suficiente sustituir el módulo RF por uno que tenga la misma distribución de pines que el utilizado en este

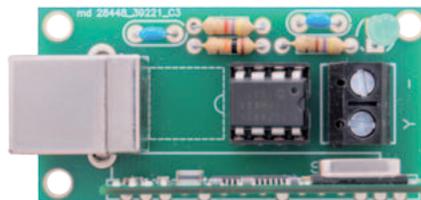
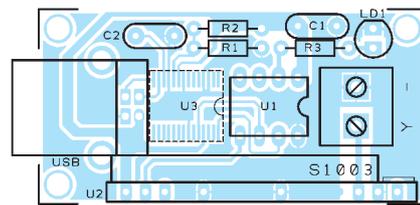
## [plano de MONTAJE]

### Lista de materiales:

- R1: 10 kohm
- R2: 4,7 kohm
- R3: 470 ohm
- C1, C2: 100 nF multicapa
- U1: PIC12F683-I/P (MF1003)
- U2: RX-4MM5
- U3: FT232RL
- USB: Conector USB-B
- LD1: LED 3 mm verde

### Varios:

- Clema de 2 terminales, 5,08 mm
- Antena tipo lápiz 433 MHz
- Circuito impreso



proyecto, ya que el código es siempre el mismo aun cambiando la frecuencia.

### EL CIRCUITO

En base a lo descrito en una edición anterior referente al desarrollo de un circuito decodificador, se ha realizado una tarjeta con interfaz USB que permite el envío al ordenador de dato identificativo del control remoto. Con el objetivo de realizar un circuito con interfaz con el ordenador sencillo, no se ha previsto ninguna alimentación adicional y el circuito se alimenta con la tensión de 5Vdc presente en el puerto USB. Además para disminuir los componentes, y también para reducir el precio del producto final se ha utilizado un módulo RF modelo RX-4MM5 alimentado también a 5Vdc y de discreta sensibilidad, cuyo coste es entorno a diez euros, pero nadie prohíbe utilizar otros módulos RF que tengan la misma distribución de pines (Fig. 1 / Tabla 1).

Para la conexión USB, se ha utilizado un integrado de FTDI, el modelo FT232RL que garantiza una conexión estable y no utiliza componentes externos aparte de un condensador cerámico conectado al pin 17.

Entrando más en detalle sobre el circuito, como se puede ver en el esquema eléctrico, hay 3 integrados, U1 que es nuestro PIC12F683 programado como decodificador, U2 que es el módulo RF y finalmente U3 que es el convertidor TTL/USB. Precisamente este últi-

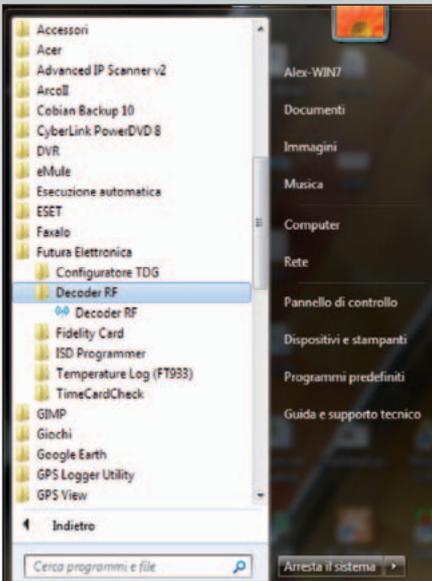


Fig. 2

mo ha sido conectado al condensador C2 sobre la salida 3V3 y dos resistencias (R1 e R2) que permiten efectuar un reset del FTDI y por tanto inicializar correctamente el puerto USB en el momento en el que su cable se conecta al ordenador. Estas dos resistencias no son obligatorias, pero si se utilizan como en este caso, proporcionar una mayor funcionalidad. El LED LD1 conectado al pin 2 de U1, permite controlar de forma visible (LED encendido), y aun no teniendo un software corriendo sobre el propio ordenador, si hay una transmisión válida para nuestros propósitos, o por un sistema de transmisión reconocida por nuestro decodificador.

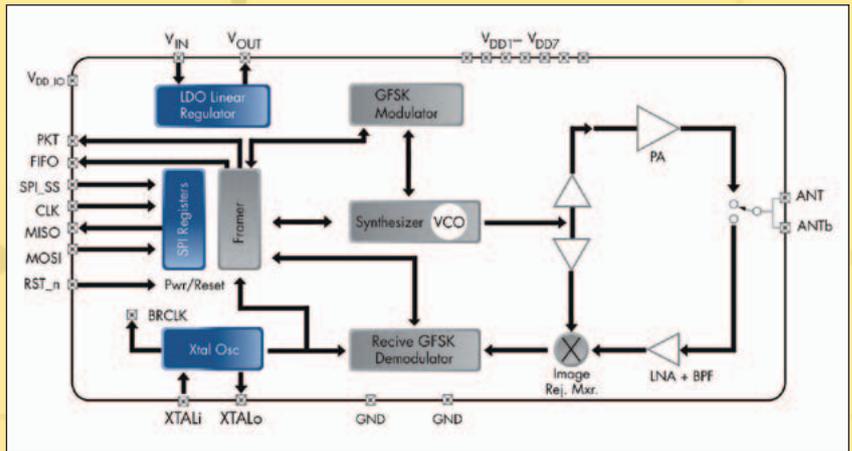
## EL SOFTWARE Y EL PROTOCOLO

Para completar nuestro proyecto hemos preparado un programa específico para visualizar de manera muy clara el código recibido e interpretado por nuestro circuito.

Antes de entrar en los puntos fuertes del software, sería conveniente explicar cómo se presenta el protocolo de comunicación, o mejor como se presenta el paquete de datos identificativos del código de control remoto para permitir a quien quiera realizar

Es el más avanzado en términos de consumo, fiabilidad, inmunidad a ruidos y costes que existe en el mercado. La solución radio on-chip WirelessUSB™-NL a 2.4 GHz de Cypress Semiconductor es ideal para realizar dispositivos HID de próxima generación y aplicaciones de control remoto como el ratón o el teclado inalámbrico. Los reducidos consumos de los dispositivos basados en esta tecnología permiten, por ejemplo, alimentar el ratón y el teclado inalámbrico por un periodo de hasta tres años utilizando solamente dos baterías AA, garantizando al mismo tiempo un alcance de hasta 30 m. La sensibilidad en recepción de -87 dBm de WirelessUSB-NL garantiza

un intervalo de funcionamiento aún más amplio. El dispositivo está caracterizado por valores bajos de corriente tanto en modo activo como en standby (inferior a 1 uA), mientras la arquitectura de radio en lazo cerrado minimiza la desviación en frecuencia de señal con el objetivo de asegurar una mayor fiabilidad de las prestaciones radio. Al igual que los componentes de las otras familias WirelessUSB, el dispositivo WirelessUSB-NL asegura mejores prestaciones en presencia de interferencias producidas por fuentes de 2,4 GHz como por ejemplo sistemas WiFi, Bluetooth, teléfonos inalámbricos y aparatos microondas.



su propio software hacerlo sin grandes dificultades. Dada la sencillez del paquete de datos, y no teniendo implementada ninguna función en particular, no ha sido necesario realizar una librería DLL para permitir la comunicación en los programas ad hoc, ya que el programador tendrá una única tarea, monitorizar el puerto serie virtual y extrapolar el único paquete de datos posible que se recibirá.

El paquete de datos informativo del ID del sistema de transmisión inicia con el carácter "<", seguido inmediatamente de un número (2 byte) que identifica el tipo de có-

digo, seguido a su vez del símbolo ":" y de tantos byte como aquellos que componen el código o, si es un código de 12 bit, esperaremos 12 byte. Para concluir esta cadena, estará el carácter ">". Respecto a los 2 byte que representan el tipo de código, se podrá esperar no más de tres alternativas, esto porque en nuestro decodificador PIC se han implementado solo 3 códigos, por tanto nos podremos esperar los siguientes casos:

- 01: Código sensoresVelleman
- 02: Código MM53200/HT12
- 03: Código Motorola

En el caso que no haya ninguna transmisión en curso, el sistema

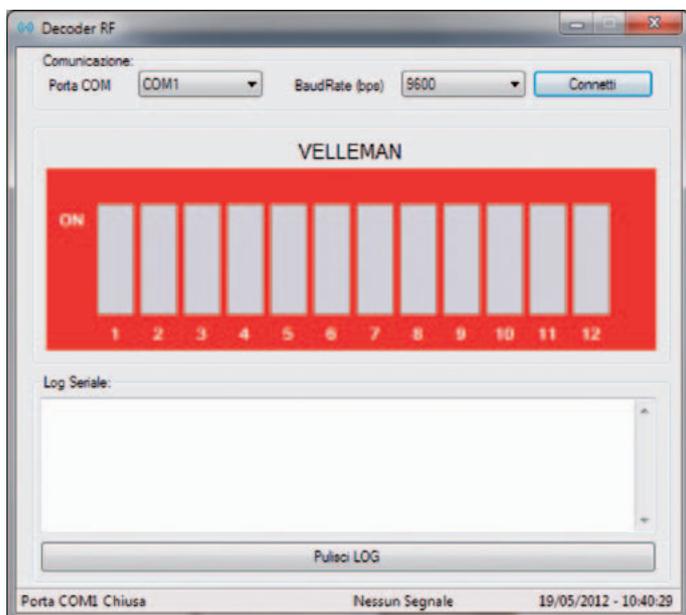


Fig. 3

no enviara ningún dato. Tomando ejemplo real, en el caso se utilice un sensor de alarmas (como los proporcionados por Velleman suministrados con la unidad HAM06WS) nos podemos esperar un paquete de datos tipo:

<01:111100101100>

No hace falta decir que la combinación "111100101100" representa propiamente el ID del sensor. Sin embargo en el caso en que se disponga de un control remoto basado en el código conocido MM53200 o HT12 nos podemos esperar un código del tipo:

<02:101011101011>

También en este hay 12 byte representados por el ID del control remoto utilizado. Mientras para el último código examinado, o el de Motorola, deberemos esperar algo ligeramente distinto siendo solo 9 byte representativos del código en cuestión, por tanto deberemos esperar cualquier cosa del tipo:

<03:--+0--+0+>

En este último caso no se tendrán mas valores "1" que indican la

posición "ON" del dip-switch del control remoto, o "0" que representa la posición "OFF" del dip-switch; sino que siendo un código three-state, los estados lógicos posibles serán 3, "+" representará la posición "+" sobre el control remoto, "-" para la correspondiente posición "-" y finalmente "0" para indicar la posición central del el dip-switch.

Entendida la cadena de referencia, veamos ahora el software, que en nuestro caso no deberá hacer otra cosa que visualizar el código en pantalla, después de haber analizado el paquete de datos deberá reconocer y visualizar el código y obviamente el código asociado. El software ha sido realizado mediante el entorno de desarrollo VB .NET 2010, que nos ha permitido realizar el paquete de instalación compatible también con los actuales sistemas operativos Microsoft, por tanto está garantizada la compatibilidad en Windows XP/ Vista/7 tanto para sistemas de 32 bit como 64 bit.

Para poder utilizar el software, es necesario instalarlo primero en el propio ordenador ejecutando el archivo "setup.exe"; al término de la instalación en el menú de "Inicio" estará presente una

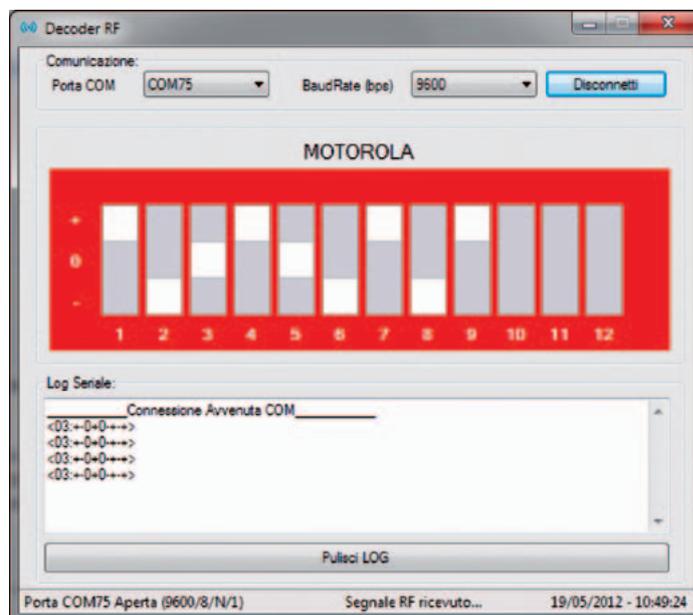


Fig. 4

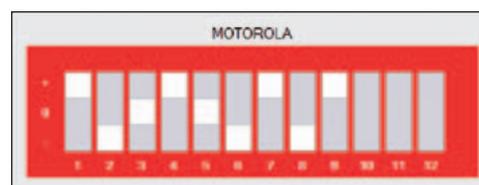


Fig. 5

carpeta desde la cual se podrá ejecutar el software, específicamente encontraremos:

"Inicio>Todos los programas > Futura Elettronica > Decodificador RF" (Fig. 2).

Arrancado el programa se mostrará una ventana desde la cual es posible elegir el puerto COM asociada a nuestro receptor, una barra de estado en la cual se puede conocer el estado del periférico, o si está conectado y si está en curso la recepción de una señal y por lo tanto el estado de los dip-switch y el correspondiente log del código (Fig. 3).

Después de haber elegido el COM correcto y haber presionado la tecla "Conectar" justo después de haber conectado el peri-



férico mediante cable USB al ordenador, se tendrá una conexión estable. Tomamos un telemando, al accionar uno de sus pulsadores, se mostrara en la barra de estado un primer mensaje "Señal RF recibida...", y al mismo tiempo se mostrara el código en el panel rojo del software y el nombre del código encima del panel, pero es de señalar que con respecto a la codificación de los sensores Antirrobo, MM53200 y HT12, se mostraran todos 12 los dip-switch en la ventana software, mientras que para el código Motorola se mostraran solo los primeros 9, 8 identificativos del código y uno para identificar el pulsador accionado, dejando en gris los últimos 3 dip-switch del panel. Tendremos por ejemplo una pantalla como en la Fig. 4. Como podemos ver, se informa de la recepción de una señal RF cuya codificación es Motorola y la cadena recibida en serie <03:+-0+0-++>. El código es por tanto representado como en la

Fig. 5, dejando vacías las posiciones 10, 11 y 12 dado que no están presentes el tal codificación.

### EL CÓDIGO DEL CONTROL REMOTO

En el mercado existen muchos tipos de controles remotos con formas y dimensiones distintas, cada uno de los cuales puede tener características distintas, por ejemplo se puede encontrar controles remotos alimentados a 3V, a 12V, controles remotos compactos como aquellos de los automóviles, que a veces están integrados en la misma llave, o grandes como un paquete de cigarrillos, en conclusión controles remotos para todos los gustos y para todos los bolsillos. Probablemente muchos de estos podrían tener uno de los métodos descritos para la codificación y para el ajuste del ID del control remoto.

Para el ajuste del código de transmisión, por lo general es posible clasificar tres soluciones comúnmente adoptadas, la primera en la cual está presente un dip-switch donde se utilizan los micro interruptores para ajustar el código; una segunda en la cual hay un pulsador para cambiar el código, en este caso a cada pulsación se genera un código diferente; y una tercera solución, en la cual es el fabricante quien memoriza directamente en el interior del control remoto el código identificativo, sin dar por tanto la posibilidad de cambiarlo autónomamente, o si se trata de controles remotos de auto aprendizaje, en los cuales el usuario tiene la posibilidad de aprender en su interior el código de un control remoto existente porque aun funcionando permite duplicar rápidamente el propio control remoto. Para decirlo de manera breve, se puede afirmar que a día de hoy, por cada



Fig. 6



Fig. 7

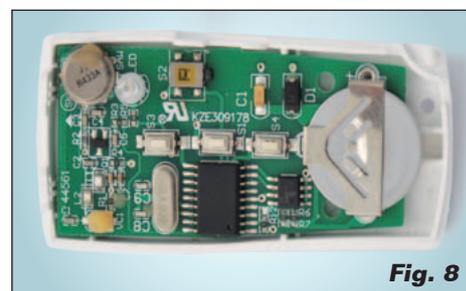


Fig. 8

exigencia es posible encontrar un control remoto adecuado. Para nuestro sistema hemos decidido examinar los tres tipos, por tanto una primera de dip-switch, utilizada para la prueba mostrada en la Fig. 5 donde cada micro interruptor es ajustado a voluntad (Fig. 6), y una segunda en la cual es el pulsador que permite generar un código diferente (Fig. 7). En tal caso hemos examinado un sensor PIR de una centralita antirrobo, donde la presión de la tecla en fase de alimentación permite generar un nuevo código, que en tal caso es posible visionar propiamente por el software presentado en estas páginas. Finalmente el tercer tipo, en el cual, como se muestra en la Fig. 8, no está presente ningún pulsador u otra cosa de fácil trazabilidad a un sistema utilizado para establecer el código identificativo del control remoto.

(170109)

## el MATERIAL

Todos los componentes utilizados en este proyecto son de fáciles de encontrar mientras el firmware del PIC y el master del PCB del receptor se pueden descargar gratuitamente de la página web de la revista ([www.nuevaelectronica.com](http://www.nuevaelectronica.com)). El receptor está disponible como kit de montaje (cod. FT1003K) al precio de 19,50 Euros. La antena tipo lápiz adicional (cod. AG433) cuesta 8,00 Euros. Están además disponibles de forma separada el módulo Aurel (cod. RX-4MM5, 9,80 Euros), el integrado FT232RL (7,00 Euros) y el microcontrolador ya programado (cod. MF1003, 6,00 Euros).

Precios IVA incluido sin gastos de envío.

Puede hacer su pedido en:

[www.nuevaelectronica.com](http://www.nuevaelectronica.com)

[pedidos@nuevaelectronica.com](mailto:pedidos@nuevaelectronica.com)