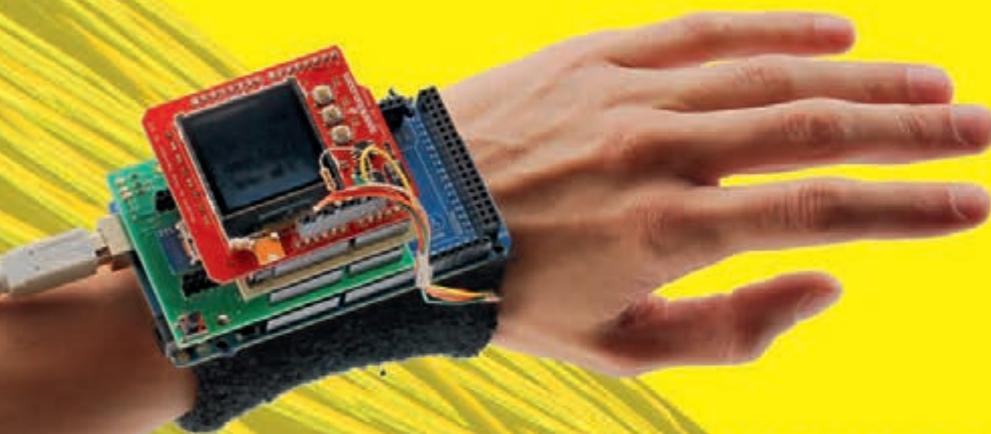


ELECTRÓNICA

NUEVA

Nº318
5,25 Euros.



RECEPTOR DE 1 Y 2 CANALES

PULSERA BIOMÉTRICA LEWE

WI-FI E INTERNET PARA ARDUINO

LA 3DRAG SE CONVIERTE EN CNC PARA CREAR PCBs

HISTORIA II



Lo mejor de la tecnología para tu hobby preferido



IMPRESORA 3D en kit
cod. K8200

¡Haz autónoma la impresora!

Controlador autónomo para impresora 3D.

cod. VM8201



¡Imprime su contenedor con la 3D!



Shield por Arduino:

RGB SHIELD



kit
cod. KA01

montado
cod. VMA01

AUDIO SHIELD



kit
cod. KA02

montado
cod. VMA02

MOTOR SHIELD



kit
cod. KA03

montado
cod. VMA03

ETHERNET SHIELD



kit
cod. KA04

montado
cod. VMA04

IN/OUT SHIELD



kit
cod. KA05

montado
cod. VMA05

PRODUCTOS DISPONIBLES EN TODOS LOS DISTRIBUIDORES
VELLEMAN DE ESPAÑA • WWW.VELLEMAN.EU

Director Editorial:

Eugenio Páez Martín

(revista@nuevaelectronica.com)

Director Técnico:

Felipe Saavedra

(tecnico@nuevaelectronica.com)

Diseño Gráfico:

Susana Domínguez

(dmcgraficas@gmail.com)

Webmaster:

Natalia García Benavent

Traducción en Lengua:

Susana Domínguez

(dmcgraficas@gmail.com)

DEPÓSITO LEGAL:

M-18437-1983

PAGINA WEB:

www.nuevaelectronica.com

SUSCRIPCIONES Y PEDIDOS

pedidos@nuevaelectronica.com

suscripciones@nuevaelectronica.com

Nº 318

2,50 Euros. Edición Digital (Incluido I.V.A.)

5,25 Euros. (Incluido I.V.A.)

Imprime:

Cromagraf

DIRECCIÓN

C/ Arboleda, 2

Oficina 405

28031 (MADRID)

Teléf: 902 009 419

Fax: 911 012 586

revista@nuevaelectronica.com

EN COLABORACIÓN CON

Elettronica In

Sumario

2014
Nº. 318

ARTICULOS

07**RECEPTOR DE 1 Y 2 CANALES**

Receptores telemando de simple y doble canal a 433,92 MHz capaces de emparejarse con un máximo de 10 transmisores cada uno. Monoestable o biestable.

13**PULSERA BIOMÉTRICA LEWE**

Cuando se lleva en la muñeca, detecta la temperatura y sudoración y transmite esa información mediante bluetooth a un smartphone, que prepara los datos recibidos y los publica en un servicio especial en la nube a través de su conexión a Internet.

21**WI-FI E INTERNET PARA ARDUINO**

Shield para Arduino que le conecta al mundo exterior mediante una red WiFi. Libera al micro de Arduino de las tareas de control del protocolo TCP/IP. Viene acompañado de su propia librería que facilita la labor de programación.

30**LA 3DRAG SE CONVIERTE EN CNC PARA CREAR PCBs**

Como transformar nuestra impresora 3D en una fresadora de control numérico CNC que podemos emplear para crear nuestros propios circuitos impresos sin insolar, revelar ni atacar con ácido, basta con enviar el archivo de diseño para tener nuestra placa lista.

40**HISTORIA II**

módulos y balizas

energía solar autónoma

www.ariston.es



JH001
Señalización para la construcción
Decoración de plazas, parques y patios



JH006
Decora al tiempo que ilumina plazas,
parques, patios y embellece veredas.



JH016
Especialmente para laterales o
márgenes de autopistas, autovías,
señalización de aceras y senderos
(plana)



JH002
Colocación en cualquier superficie
Circunvalaciones, intersecciones,
autopistas y autovías



JH007
Para iluminar y realzar en colores,
jardines, parques, patios, muros,
veredas.



JH018
Señalización para la construcción
y señalización del mar (faros)



JH003
Especialmente para laterales o
márgenes de autopistas, autovías,
señalización de aceras y senderos



JH008
Diseñado especialmente para la
demarcación y señalización de
cualquier espacio fluvial y marítimo,
puertos deportivos, lagos, canales,
piscinas.



JH019
Decora y señala rutas de plazas,
parques, muros y senderos
(forma de trébol)



JH004
Por sus características puede ser
colocado en columnas de parquings
o muros.



JH009
Decora y señala rutas de plazas, muros
y senderos



JH722
Luz para la señalización de peligro



JH005
Señalización de medianas y arcones
de autopistas, intersecciones y stops,
carreteras secundarias.

El final de una etapa . . .

Nueva Electrónica , 18 de Junio de 2014

Querido y fiel lector,

Hemos de empezar pidiéndote disculpas por todos los inconvenientes que te hayamos podido causar en estos últimos dos años de retrasos e incertidumbres y, al mismo tiempo, dándote las gracias por tu paciencia y por la comprensión que nos has demostrado.

Todos sabemos los terribles efectos que esta profunda crisis económica que estamos soportando en Europa, y España en particular, ha tenido sobre muchas compañías que llevaban ofreciendo productos de calidad desde los años 70 del siglo pasado y, desgraciadamente, Nueva Electrónica no ha sido una excepción y ha sufrido muy de cerca sus devastadoras consecuencias. Nuestra matriz italiana terminó por desaparecer el pasado año tras lo que podríamos calificar como una larga agonía de casi tres años. Poco a poco nos fuimos quedando desabastecidos, tanto de contenidos como de productos, y como consecuencia no podíamos cumplir nuestros compromisos con nuestros lectores y anunciantes.

No podíamos dejar que, tras más de treinta años en el mercado de habla hispana, Nueva Electrónica desapareciera sin más. No se trata solo un negocio; para algunos de nosotros es parte de nuestra vida y no podíamos dejarla morir sin presentar batalla. Durante meses hemos luchado contra viento y marea para intentar salir adelante. Pero las cosas no son fáciles, y menos en un entorno de crisis como el actual. Buscamos y buscamos, pero parece que nadie está dispuesto a invertir o arriesgar nada, y menos aún si no tienes detrás el respaldo de una marca solvente, un buen producto y tu proyecto está en un sector que, por definirlo con cariño, no tiene claro cuál es su futuro.

Con todo en contra, parece que ha llegado el momento de rendirse, de abandonar la pelea, de tirar la toalla . . .

.../...

... es el principio de la siguiente

... o por el contrario, aprovechar esta situación para plantearse un cambio radical de estrategia, de forma de hacer las cosas, buscando nuevas metas acorde con los nuevos tiempos, con las nuevas tecnologías y, sobre todo, con las nuevas necesidades que ahora tienen los lectores, y todo ello sin perder nuestra esencia, conservando nuestra identidad, siendo Nueva Electrónica.

Este es el camino que hemos elegido, y en él estamos invirtiendo ya todos nuestros esfuerzos. Pero no es un camino fácil de recorrer en solitario, y por ello hemos buscado compañeros de viaje que caminen a nuestro lado, que nos ayuden en los tramos difíciles y que nos animen y empujen para alcanzar la meta.

Por un lado tenemos a Board and Book, una nueva compañía editorial española especializada en publicaciones técnicas, electrónica y programación sobre todo, que nos dará cobertura en todas las cuestiones tecnológicas y de producción editorial, tanto impresa como online, y por otro tenemos a la compañía italiana Futura Elettronica, que edita entre otras la popular revista *Elettronica In* y que produce módulos electrónicos desde hace décadas, que será quien nos facilite contenidos y productos que ofrecer a nuestros lectores.

Todo este proceso requiere su tiempo y, aunque nos gustaría correr más y ofrecerte ya nuestra nueva imagen, queremos hacerlo bien., así que te rogamos un poco más de paciencia. A la vuelta del verano te presentaremos la renovada Nueva Electrónica, que estamos seguros estará a la altura de tus expectativas.

Como adelanto, en esta edición encontrarás cuatro proyectos en los que podrás apreciar la calidad de los contenidos y de los productos que te vamos a ofrecer en breve.

Permanece atento a nuestra web, www.nuevaelectronica.com, y a tu correo. Pronto tendrás noticias nuestras.

Gracias,

El Equipo de Nueva Electrónica

RECEPTOR 1 Y 2 CANALES

ALESSANDRO SOTTOCORNOLA

Receptores de 433,92 MHz mono y bicanal capaces de emparejarse con un máximo de 10 TX cada uno. Monoestable o biestable.



Basado en el integrado MM53200, emplea una de las codificaciones más antiguas y probadas que se utiliza en los radiocontroles generales y en los de apertura de puertas, tanto es así, que la mayoría de los transmisores disponibles en el mercado para tales aplicaciones la utilizan. El estar en el comercio durante mucho tiempo, tiene el problema de que los codificadores integrados quedaron obsoletos y fueron sustituidos por otros nuevos, por ejemplo por el UM3750 y el UM86409 (ambos fabricados por

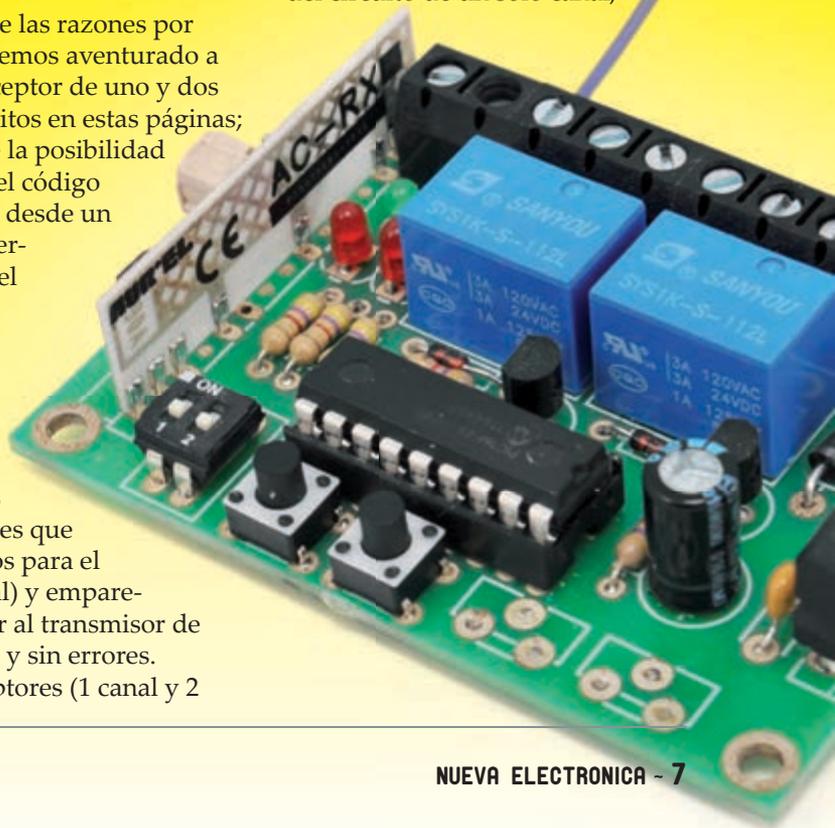
UMC). Por esta razón, es bueno tener un receptor con microcontrolador capaz de adaptarse a ésta codificación y que la aprenda en modo automático; esto aleja el problema de la obsolescencia de los chips y permite también replicar los receptores incluso sin disponer del decodificador.

Ésta es una de las razones por las que nos hemos aventurado a diseñar el receptor de uno y dos canales descritos en estas páginas; la otra es que la posibilidad de aprender el código directamente desde un transmisor permite reducir el tamaño del circuito del receptor (porque no se necesitan los interruptores dip de 12 pines que son necesarios para el ajuste manual) y emparejar el receptor al transmisor de forma rápida y sin errores. Los dos receptores (1 canal y 2

canales) descritos en este artículo son compatibles, tanto con los codificadores MM53200, UM86409 y UM3759, como con el más reciente Holtek HT-12, que tienen una configuración binaria de codificación compatible.

ESQUEMA ELÉCTRICO 1 CANAL

Empezaremos con el análisis del circuito de un solo canal,



basado en un microcontrolador PIC16F683 y un módulo híbrido receptor AC-RX2, sintonizado a 433,92 MHz. Este módulo, U2, contiene la parte de radio del circuito y es un receptor de Aurel modelo AC-RX2 que está provisto de un amplificador de la señal de antena (que le confiere una sensibilidad de -106 dB), un etapa de sintonía superregenerativa sintonizado a 433,92 MHz calibrada en fábrica y dotada de un filtro de RF (el filtro se encarga de mejorar la selectividad, que no es elevada en los súperregenerativos) y un demodulador de amplitud. Completan el módulo un comparador en cuadratura de la señal digital (nivel TTL) disponible en el pin 14 y un amplificador de BF para la señal de salida del demodulador AM. Inmediatamente después del encendido, el micro inicializa sus propias E/S estableciendo GP0 y GP1 como salidas para controlar respectivamente el LED de señalización LD2 (que indica tanto el modo de funcionamiento, como el procedimiento de auto aprendizaje) y el transistor NPN T1 que con-

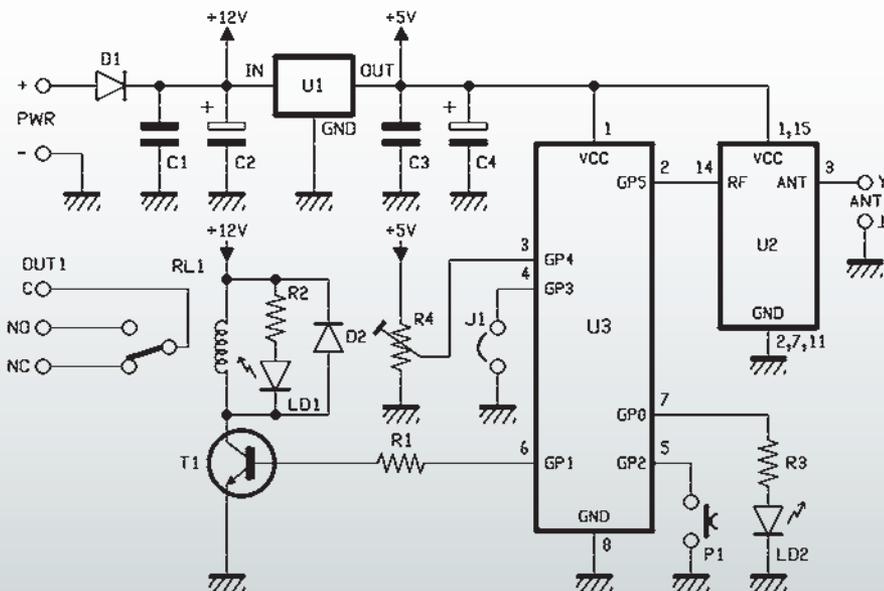
trola el relé. Éste último se excita cuando el transistor entra en saturación, lo que ocurre cuando GP1, que controla la polarización de la base, se halla a un nivel lógico alto (R1 limita la corriente de base del T1 para evitar que la unión base-emisor se dañe); cada vez que el transistor conduce, además de a la bobina del relé, alimenta R2 y LD1, que se encarga de la señalización de relé activo. En paralelo a la bobina del relé está también el diodo D2, cuya función es provocar un cortocircuito a la extracorrente inversa que genera la bobina del cuando el transistor pasa al corte y se desactiva el relé; si no existiera el diodo, la tensión se cerraría sobre la unión base-colector de T1, dañándolo en muy poco tiempo. Continuando con la inicialización, el PIC configura GP2, GP3 y GP5 como entradas, dedicados respectivamente a la lectura del botón P1, del puente J1 y de los datos de salida del AC-RX2; los dos primeros tienen activo el pull-up interno. La línea GP4 se inicializa como entrada asignada al con-

CARACTERÍSTICAS TÉCNICAS

- Número de salidas: 1-2
- Modalidad salida: Monostable, Bistable
- Alimentación: 12 Vcc
- Absorción: 40 mA max.
- Memoria: 10 códigos por canal
- Codificación: MM53200/HT-12

vertidor A/D y está destinada a futuros desarrollos del firmware. Por el momento es irrelevante en el funcionamiento del circuito. El circuito se completa el bloque de la fuente de alimentación, que parte de los puntos + y - PWR (bornes de alimentación) y, pasa por el diodo D1 (que protege contra el daño que podría causar la inversión de la polaridad) para llegar al regulador integrado de tres terminales U1 cuya entrada se filtra con C1 y C2, que reducen el rizado de la tensión de alimentación (C2) y de los interferencias de RF e impulsos captados de los cables de alimentación (C1). Función similar la desarrollan C3 y C4 para la salida del regulador. Dicho esto, podemos explicar cómo funciona el receptor de 1 canal: aplicando alimentación al circuito, el firmware, después de la inicialización de las E/S, hace que LD2 parpadee 5 veces para indicar que se ha puesto en marcha correctamente y funciona con normalidad, que se corresponde con la ejecución del comando recibido; a continuación, se ejecuta el bucle principal en el que se comprueba las variaciones del nivel en el pin 4 (es decir, el cambio en el estado del puente J1) así como el posible accionamiento de P1. Una rutina especial comprueba periódicamente el estado del pin 2, o lo que es lo mismo, la llegada de datos de AC-RX2; cuando presionamos uno de los botones en el transmisor, la señal de RF transmitida por éste último alcanza la antena receptora del AC-RX2, que se encarga de demodular los datos y

[esquema del RECEPTOR DE 1 CANAL



enviarlos al terminal 14 para que estén disponible para el microcontrolador, que los lee en el pin 2. El micro funciona como decodificador del código correspondiente a la tecla pulsada en el control remoto (pero el firmware también ofrece una rutina para aprender de forma automática el mismo código de base sin necesidad de ajustarlo mediante microinterruptores) y funciona de esta manera: el micro controlador toma los impulsos TTL, los pone en la RAM y los analiza con un firmware apropiado que primero distingue, de entre las muchas señales captadas en el éter, aquellas que son compatibles con el formato la codificación UM3750, para después, en caso afirmativo, comprobar si el código recibido es uno de aquellos que se memorizaron durante la fase de auto aprendizaje; en caso contrario elimina los datos de la RAM y se prepara para un nuevo análisis. Veremos más adelante cómo emparejar el transmisor al circuito, mediante auto aprendizaje; por ahora basta decir

Tabla 1 - Funcionamiento de los LED en el receptor de 1 canal.

Diodo	Funcionamiento Normal	Programación
LD1	actividad OUT1: on = RL1 activo off = RL1 en reposo	-
LD2	entrada o salida en modo de aprendizaje, borrado de los códigos, indicación de inicialización correcta con 5 parpadeos en modo normal.	se encenderá durante 2 segundos al entrar en modo programación, parpadea cuando el circuito ha aprendido el código transmitido; queda fijo si el aprendizaje ha fallado.

que el firmware puede aprender un máximo de 10 códigos, que incluso pueden ser de transmisores diferentes. De hecho cada código es memorizado completo, es decir, todos los 12 bits que lo componen. Ahora vamos a ver lo que sucede si la señal recibida contiene uno de los códigos aprendidos y almacenados en la EEPROM de trabajo del PIC: en este caso, pone en marcha la correspondiente rutina de gestión del relé, que determina diferentes acciones dependiendo de si el modo de funcionamiento es monoestable o biestable; el ajuste de modo se realiza con J1 (el puente abierto quiere decir funcionamiento mo-

noestable, mientras que cerrado lo hará en modo biestable). En el modo biestable el relé cambia de estado cada vez que el microcontrolador detecta un código válido, mientras que en el monoestable (o impulsivo) el relé pasa a un nivel alto tras la recepción de un código válido y vuelve a descansar cuando el botón correspondiente al código se suelta, es decir, al poner fin a la señal que causó la activación: en definitiva, el relé sigue al botón. Si simultáneamente se detiene la transmisión al tiempo que se recibe la de otro mando a distancia que haya sido emparejado (cuyo código sea uno del 10 aprendidos) dentro del tiempo de

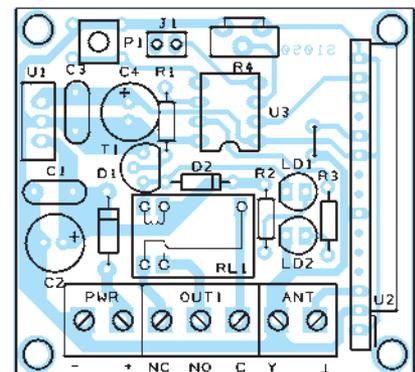
[plano de montaje del RECEPTOR DE 1 CANAL]

Lista de materiales:

- R1: 4,7 kohm
- R2: 1 Kohm
- R3: 470 ohmios
- R4: sin montar
- C1: 100 nF multicapa
- C2: 100 uF electrolítico VL 35
- C3: 100 nF multicapa
- C4: 10uF electrolítico VL 35
- U1: 7805
- U2: AC-RX2
- U3: PIC12F683 (MF1050)
- D1: 1N4007
- D2: 1N4148
- LD1: LED rojo 5 mm
- LD2: verde 5mm LED
- T1: BC547
- P1: Micropusador
- RL1: relé 12V

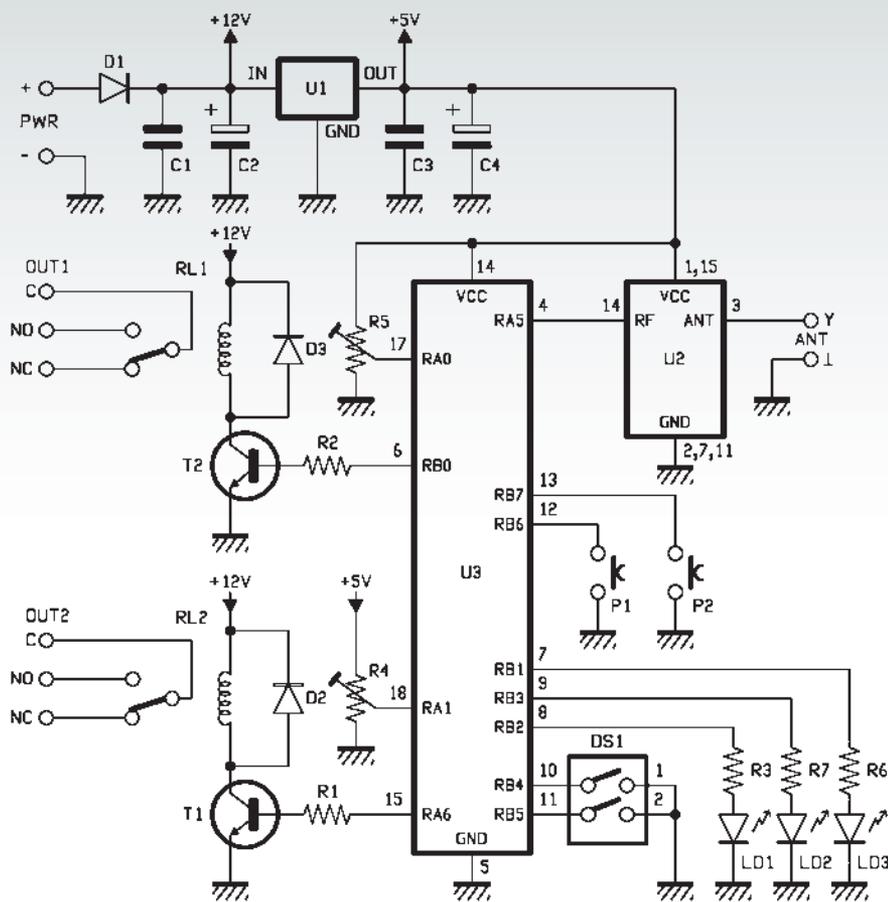


- Varios:
- Terminal 2 polos (2 uds.)
 - Terminal 3 polos
 - Zócalo 4+4



- Tira de pines macho 2 vías
- Puente
- Circuito impreso

[Esquema del **RECEPTOR DE 2 CANALES**



espera establecido en el firmware, el microcontrolador considera que la transmisión no se ha interrumpido. Si se coinciden el código de otro transmisor (TX) cuando el relé está activo con la transmisión de otro transmisor válido, no es aplicable lo anterior porque la señal del segundo interfiere con del primero y el receptor de radio demodula el batido entre los dos flujos de datos; como resultado, el microcontrolador interpreta que ya no es la señal de activación y desactiva el relé. Ten en cuenta que la modalidad de funcionamiento de la salida también se puede cambiar durante el funcionamiento del receptor, es decir, no es necesario apagar y reiniciar el circuito, ya que el estado del jumper J1 se lee continuamente.

APRENDIZAJE DE CÓDIGOS PARA EL MÓDULO DE 1 CANAL

Una vez visto cómo se comporta el receptor cuando llega un comando, veamos ahora como introducir los códigos, es decir, cómo se desarrolla el aprendizaje; esto se puede hacer en cualquier momento pulsando y manteniendo apretado el botón P1 hasta que el LED verde (LD2) se encienda: entonces comienza la fase de auto aprendizaje. En este punto, hay que transmitir al receptor el código a aprender presionando el botón del transmisor del control remoto que se desea aprender, y esperar hasta que el LED parpadee para indicar que el aprendizaje ha sido un éxito; si el TX dispone de más pulsadores, y por

tanto de más canales, se pueden aprender los códigos de todos. Si el LED permanece con la luz fija, significa que la memoria está llena o el código transmitido no es válido (no tiene el formato requerido para un TX con codificador MM53200, UM3750, UM86409 o HT-12. Presta atención a los detalles: por cómo funciona el firmware y por cómo se gestiona el EEPROM del microcontrolador, el circuito puede almacenar todos los códigos de transmisión sin limitación, excepto para el que proporciona todos los bits a 1 (todos los interruptores del transmisor en ON); por tanto, el transmisor podrá usar 4.095 combinaciones en lugar de 4.096. Si se desea borrar los códigos, hay que decir que el procedimiento se ha previsto, pero no es posible hacerlo de forma individual, ya que se puede borrar la memoria entera pero no parcialmente. Para borrar la EEPROM y así eliminar los códigos aprendidos, es necesario quitar alimentación al circuito y volver a encenderlo después de pulsar el botón P1, que sólo se soltará cuando el LED verde permanezca encendido de forma continua durante 2 segundos, lo que indica que el borrado de la memoria se ha completado. En este punto, se puede soltar el botón, con lo que el LED verde emitirá 5 parpadeos para indicar la salida del procedimiento de borrado y el arranque normal del receptor (Tabla 1).

ESQUEMA ELÉCTRICO 2 CANALES

Ahora echemos un vistazo al circuito de 2 canales sin repetir lo que ya se ha explicado para el receptor de un solo canal. Observando el esquema, se puede comprobar que la etapa de alimentación es la misma que

para el esquema de un solo canal. Lo mismo para la etapa receptora. El microcontrolador es diferente, es un PIC16F688, cuya elección se debe al mayor número de E/S que dispone con respecto al PIC16F683 utilizado en la versión de un solo canal.

Obviamente tenemos dos etapas de relé, que funciona como ya se ha descrito, y otros tantos diodos de protección en paralelo a las correspondientes bobinas.

Encontramos también un doble interruptor dip, que sirve para establecer la modalidad de activación de la salida de cada uno de los dos canales, y dos potenciómetros de ajuste que, como se menciona en la descripción del circuito de un solo canal, están reservados para desarrollos futuros. Inmediatamente después del encendido (señalado por una secuencia de 5 parpadeos del LED verde que indican una correcta puesta en marcha) el micro inicia sus E/S configurando RA4 como entrada para la adquisición de los datos procedentes del receptor AC-RX2, RB4 y RB5 como entradas (con pull-up interno activado en ambas líneas) para leer el estado de los conmutadores y RB6, RB7 para leer el estado de los botones P1 (inicio de aprendizaje de códigos para el canal 1) y P2 (inicio de aprendizaje de códigos para el canal 2); también estas últimas líneas de E/S tienen activo el pull-up interno. Continuando con la inicialización, el microcontrolador configura RB1, RB2, RB3 como salidas dedicadas, respectivamente, al control del LED de señalización LD3 (verde), que indica los pasos del procedimiento de auto aprendizaje y el funcionamiento normal, LD1, que indica la actividad del canal 1, y LD2 hace lo mismo para el canal 2. Para los dos relés el procedimiento es similar al realizado para RL1 del circuito de un solo canal, dado que las dos

etapas son iguales a aquella: T1 controla el relé RL2 mediante RA6, mientras que T2 controla RL1 mediante RB0. D2 y D3 protegen los transistores.

Continuando con la inicialización, el PIC configura GP2, GP3 y GP5 como entradas, dedicadas respectivamente a la lectura del botón P1, del puente J1 y de la salida de datos del AC-RX2; para los dos primeros está activo el pull-up interno.

El conmutador del DS1 se utiliza para ajustar la modalidad de funcionamiento deseado para cada canal; más exactamente, el Dip1 permite el ajuste de salida del OUT1 y el Dip2 hace lo propio para la salida OUT2. Cerrados significa funcionamiento biestable, mientras que abiertos significa funcionamiento monoestable. Como en el circuito de un solo canal, en la modalidad monoestable la salida no está temporizada, por tanto se activa pulsando el botón en el transmisor y se desactiva cuando se libera el mismo botón. Los potenciómetro R4 y R5 no están montados, pero están previstos para futuros desarrollos. Los dos LED rojos siguen el estado de las salidas durante el funcionamiento. LD1 se enciende si el relé del OUT1 se activa y LD2 se enciende si el relé del OUT2 se activa.

APRENDIZAJE DE CÓDIGOS EN EL MÓDULO DE 2 CANALES

Para emparejar los transmisores

con el circuito receptor es necesario proceder de la siguiente manera: presione y mantenga presionado el botón del canal que desea memorizar (P1 para el canal OUT1 y P2 para el canal OUT2). El LED rojo relativo a la salida correspondiente (LD1 para OUT1 y LD2 para OUT2) se enciende para indicar que se está en modo de aprendizaje. Una vez hecho esto, se debe pulsar el botón del mando a distancia que se desea memorizar y esperar hasta que el LED parpadee, lo que indica que el aprendizaje se ha realizado correctamente. Si el LED permanece fijo, quiere decir que la memoria está llena o el código transmitido no es válido. Como en el caso del receptor de un canal, debido a la forma en que se gestiona la EEPROM, no se permite el aprendizaje de los códigos que corresponden a todos los interruptores dip del transmisor en la posición ON, por lo que tenemos disponibles 4.095 combinaciones en vez de 4.096.

En cuanto al borrado de los códigos, se puede borrar la memoria en un canal de salida, pero no un código particular del mando a distancia. Para borrar la memoria de un canal hay que encender el circuito mientras se mantiene pulsado el botón asociado al canal que se desea borrar: P1 para el canal 1 (OUT1) o P2 para el canal 2 (OUT2). Cuando el LED verde (LD3) permanece encendido fijo durante 2 segundos significa que

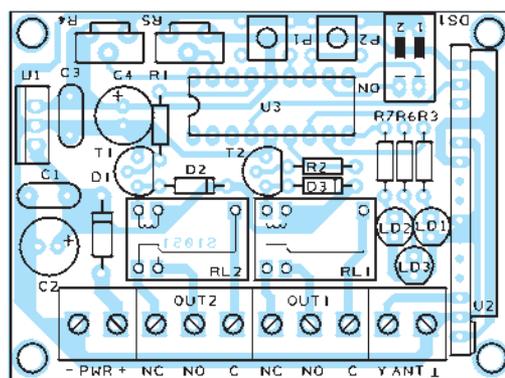
Tabla 2 - Funcionamiento de los LED en el receptor de 2 canales.

Diodo	Funcionamiento Normal	Programmazione
LD1	actividad OUT1: on = RL1 activo off = RL1 en reposo	-
LD2	actividad OUT2: on = RL2 activo off = RL2 en reposo	-
LD3	entrada o salida en modo de aprendizaje, borrado de los códigos, indicación de inicialización correcta con 5 parpadeos en modo normal.	se encenderá durante 2 segundos al entrar en modo programación, parpadea cuando el circuito ha aprendido el código transmitido; queda fijo si el aprendizaje ha fallado.

[plano de montaje del RECEPTOR DE 2 CANALES

Lista de materiales:

R1: 4,7 kohm
R2: 4,7 kohm
R3: 470 ohm
R4: no montado
R5: no montado
R6, R7: 470 ohm
C1, C3: 100 nF multicapa
C2, C4: 100 μ F 35 V
electrolítico
U1: 7805
U2: AC-RX2
U3: PIC16F88 (MF1051)
D1: 1N4007
D2, D3: 1N4148
LD1: LED 3 mm rojo
LD2: LED 3 mm rojo
LD3: LED 3 mm verde



T1-T2: BC547
P1: Micropulsador
P2: Micropulsador
RL1: Relé 12V
RL2: Relé 12V
DS1: Microinterruptor DIP 2 vías

Varie:
- Terminal 2 polos (2 pz.)
- Terminal 3 polos (2 pz.)
- Zócalo 9+9
- Circuito impreso

se ha completado el borrado del banco de memoria correspondiente al canal elegido y se puede soltar el botón. El LED verde parpadeará 5 veces para indicar la salida del procedimiento de borrado y el arranque normal del receptor (Tabla 2).

REALIZACIÓN PRÁCTICA

Para ambos circuitos hemos diseñado un circuito impreso, del cual está disponible para su descarga (en la web www.elettronica.in.it) el diseño de la cara de cobre que se puede utilizar para crear la película necesaria para el fotograbado y construcción del circuito impreso. Una vez grabada y perforada la placa, podemos montar los pocos componentes necesarios, empezando por las resistencias y los diodos, continuando con los zócalos, el interruptor dip (sólo en la versión de 2 canales), los condensadores (primero los no polarizados y luego los electrolíticos), transistores, LEDs, relés, regulador de tensión, bornes de conexión, etc. Para la correcta orientación de los elementos polarizados (transistor, diodos, regulador, electrolíticos y micro-

controlador) debemos guiarnos por el correspondiente plano de montaje reproducido en estas páginas. Una vez completado en montaje, sólo queda insertar el microcontrolador (con el firmware correspondiente programado) en su zócalo, prestando atención a la orientación. Hay que dotar al receptor de una antena adecuada para 434 MHz: por ejemplo un plano de tierra o un sencillo lápiz; la conexión se hace preferiblemente con un cable coaxial RG59, cuyo conductor central (alma) tiene que unir el lápiz con el terminal Y del conector ANT, mientras que un extremo de la malla de pantalla tiene que ir a masa del mismo conector en el lado del circuito y el otro extremo al eventual plano de masa de la antena (si es un simple lápiz, la masa del lado de la antena debe aislarse, es decir, sólo irá conectada al circuito). La antena puede ser también un simple trozo de alambre de cobre de 17 cm de largo (para tener una antena a $\frac{1}{4}$ de longitud de onda) o 35cm (para tener una antena a media onda) unido al contacto Y del conector ANT. Todo esto vale para ambos receptores (1 ó

2 canales). Para la alimentación de los circuitos, se puede utilizar una fuente corriente continua de 12÷16 Vcc y capaz de suministrar una corriente de 70 mA para el receptor de un solo canal y 100 mA para el de dos canales.



EL MATERIAL

Todos los componentes utilizados en estos proyectos son fáciles de encontrar. El master de las placas de circuito impreso y el firmware se puede descargar desde la web www.elettronica.in.it.

Ambos receptores están también disponibles en kit: cod. FT1050K (mono canal, 17,00 Euros); cod. FT1051K (bicanal, 19,00 Euros). Con estos receptores es posible utilizar transmisores de 2 canales (cod.8110-TX3750-2CS 19,00 Euros) y 4 canales (cod.8110-TX3750-4CS 22,00 Euros).

Los precios incluyen IVA.
Los gastos de envío no van incluidos.
Puede hacer su pedido en:
pedidos@nuevaelectronica.com

LEWE PULSERA BIOMÉTRICA

Cuando se lleva en la muñeca, detecta la temperatura y la sudoración y transmite la información a través de bluetooth a un smartphone, conectado a Internet y preparado para publicarlos en un servicio especial en la nube.

MIRCO SEGATELLO
y ALESSANDRO PASQUALINI

El desarrollo de los medios de comunicación ha experimentado un impulso increíble en los últimos años; en particular, la telemática ha alcanzado niveles cada vez más sofisticados, dando lugar a una verdadera revolución en la difusión de la

información, rompiendo barreras económicas, geográficas y culturales. El uso de las nuevas tecnologías garantiza hoy la oportunidad de hacer circular las ideas y proyectos, dando vida a un flujo de comunicación donde los argumentos y contenidos digitales son compartidos por millones de usuarios en todas las partes, son al



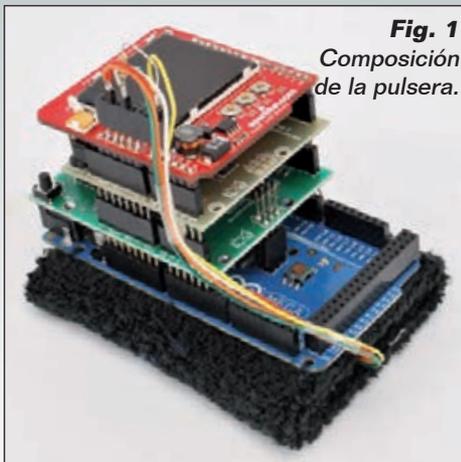


Fig. 1
Composición
de la pulsera.

mismo tiempo usuarios, creadores y coautores de un trabajo global. El proyecto Lewe parte de este ámbito y los objetivos de recogida y procesamiento de las tecnologías disponibles para lograr una idea a

bajo coste aplicable a todos los sectores que utilizan sensores para la recolección de datos. El proyecto tiene como objetivo la realización de una portal en la nube, llamado Lewe, el protocolo de comunicación, denominado jack (capaz de transportar los datos de una manera fiable) y de una serie de dispositivos de diferente naturaleza que explotan esta estructura para publicar automáticamente los datos en la web. Ejemplos de la aplicación de este protocolo pueden ser la puesta en marcha de una alarma de coche, la administración remota de un invernadero,

o bien el brazalete biométrico Lewe descrito en estas páginas. La pulsera biométrica nace de la idea de los autores: Alexander, que ama inmensamente la escalada y el paracaidismo, y Mirco, su entrenador personal, que quería conocer en tiempo real los datos biométricos de Alexander simplemente accediendo a la nube online de Lewe. Todo esto queda plasmado en una pulsera para la lectura de los datos biométricos que se enviarán a través de bluetooth

nio público...) e integrar todo en una sola tarjeta o a lo sumo dos, muy compacta y aplicable en la muñeca mediante un recipiente adecuado con pulsara (tipo medidor de presión sanguínea). En el proyecto que hemos desarrollado se incluyeron las siguientes funciones:

- detección de la temperatura y sudoración del cuerpo;
- visualización local de los parámetros adquiridos;
- envío de los valores a una

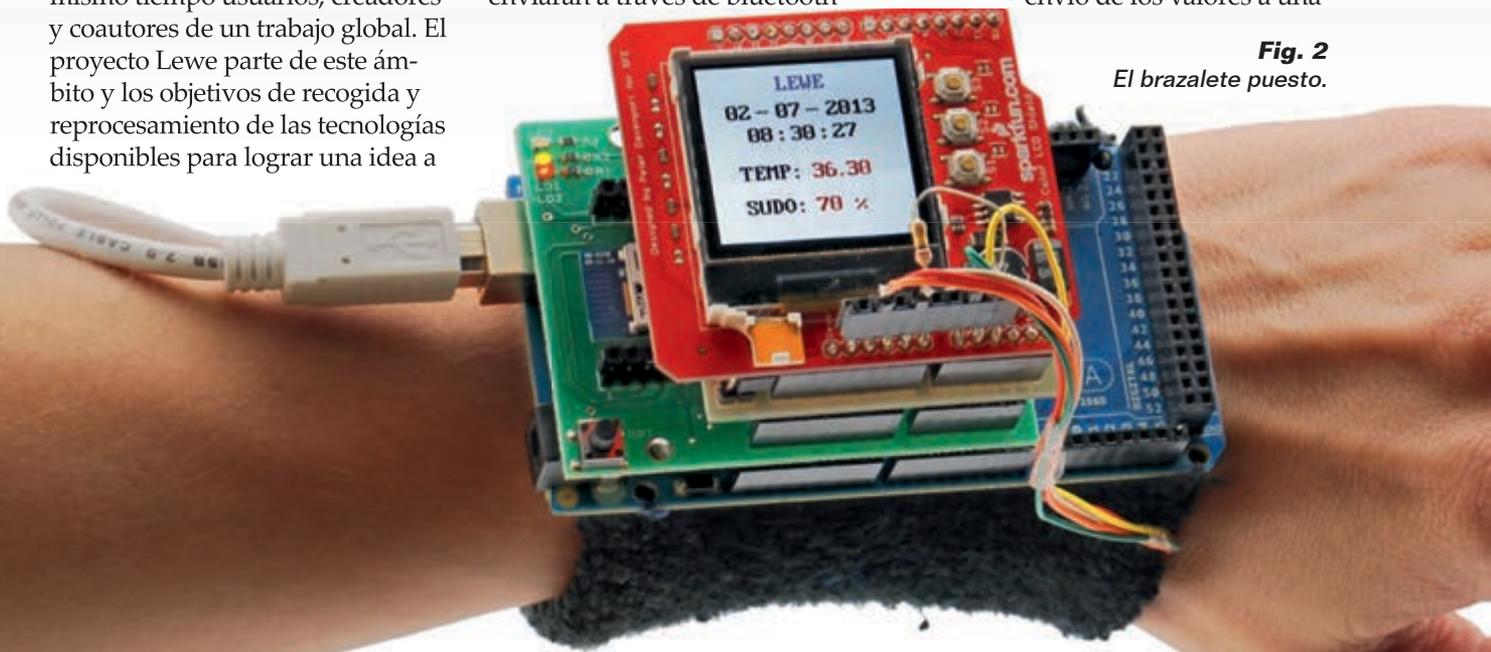


Fig. 2
El brazalete puesto.

a un smartphone Android y, a través de él, a un servicio cloud para el archivo y la organización algunos datos. En la versión experimental, éste proyecto se llevó a cabo (ver Fig. 4) mediante una tarjeta Arduino Mega que alberga un Bluetooth de Futura Electrónica, un shield RTC (siempre de Futura Electrónica) y un LCD a color de pantalla Sparkfun; claramente todo tiene un tamaño nada despreciable, pero nuestra intención es explicar cómo realizar la aplicación, sin perjuicio de que si crees que puedes conseguir tu propia arquitectura de hardware Arduino Mega (los esquemas de las Placas Arduino son de domi-

aplicación instalada en un smartphone;

- carga gratuita de los datos en una nube online;
- organización de los datos en gráficos simples que ayudan a comprender la evolución temporal de los parámetros biométricos y compararlos con los de otros usuarios.

HARDWARE

El brazalete se ha desarrollado en torno a la plataforma Arduino Mega (Fig.3). Para la comunicación con el smartphone se ha utilizado un módulo Bluetooth RN-42 interconectado con Arduino a través de un shield Bluetooth

de Futura Electrónica, mediante el hardware serie (pin 0 y 1). En este proyecto hemos preferido utilizar un interfaz serie software en los pines 6 y 10, para permitir que el interfaz serie hardware quede libre para la comunicación con el PC, incluso durante la comunicación Bluetooth, los que resulta útil para la depuración del software. Para ello, debemos interrumpir las pistas que salen de los pines 0 y 1 y conectarlos a los pines 10 y 6 del módulo respectivamente; por comodidad, se pueden usar los pines del conector denominado RS232 ya presente en el shield.

Para evitar que los pines

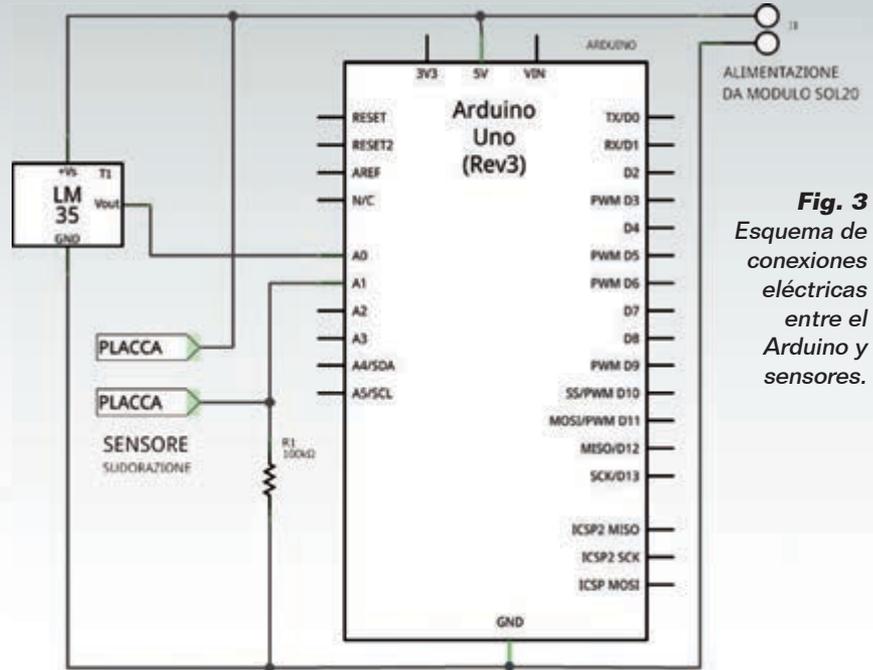


Fig. 3
Esquema de conexiones eléctricas entre el Arduino y sensores.



0 y 1 del módulo Bluetooth entren en comunicación con los contactos inferiores de la placa Arduino, puede cortar o bien doblar hacia el exterior (Fig.5), a fin de no introducirlos en los conectores de la tarjeta. Para la lectura de la temperatura hemos utilizado un sensor común LM35 con encapsulado TO-92, cuya salida está conectada a la entrada analógica A0 de Arduino. En cambio, para la detección de la sudoración se utilizaron dos electrodos conectados

a un divisor de tensión unido a la entrada analógica A1 del Arduino. El inicio del funcionamiento del sistema de detección de la transpiración se basa en la conductividad del sudor, el cual, por ser una solución salina, permite que la corriente fluya de un electrodo al otro; cuanto mayor sea el nivel de la transpiración, menor es la resistencia medida entre los electrodos.

Estos dos electrodos deben ser del tamaño de una moneda de un céntimo y, obviamente, tener buenas características de conductividad y resistencia a la oxidación. Con el divisor de tensión obtenemos un valor de tensión dependiente del nivel de sudoración, que a su vez está relacionado con los estímulos emocionales de quién se lo pone y también es un buen indicador del esfuerzo físico del atleta. Evidentemente las tensiones y corrientes implicadas son tan pequeñas como para no suponer ningún peligro para la persona que lleva la pulsera. En el esquema eléctrico se puede ver que los sensores se alimentan a través del pin 5V de Arduino, pero como

la pulsera estará alimentada por batería, hemos previsto la posibilidad de controlar la alimentación de los sensores, activándola solo cuando sea necesario para realizar la medida. La alimentación del sensor LM35 se puede tomar del pin 12, mientras que el sensor de la transpiración se puede alimentar desde el pin 7. Estos pines se mantienen normalmente a nivel bajo y el software los activa sólo cuando sea necesario llevar a cabo la medida (esto permite limitar el consumo eléctrico, y alargar la vida de la batería); la corriente de salida de 40mA indicada en las hojas de características del micro controlador Atmel, es suficiente para alimentar los pines afectados por la medición. Para correlacionar temporalmente los datos detectados, hemos utilizado un módulo RTC puesto en comunicación con el Arduino a través del bus I²C, mediante los pines analógicos A4 y A5. Hemos también incluido un shield LCD Color para mostrar los datos biométricos en tiempo real y el estado de funcionamiento, además de fecha y hora; las mediciones adquiridas

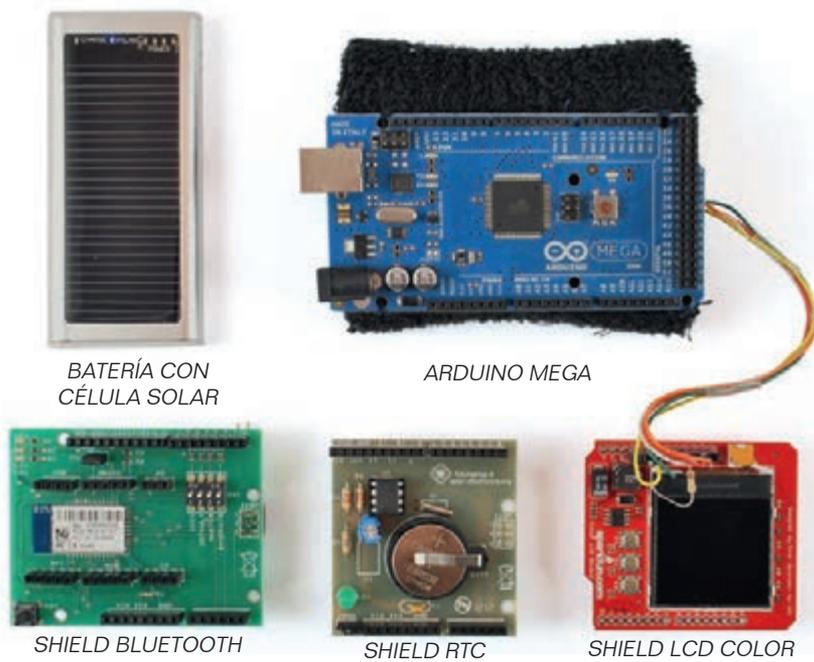


Fig. 4 - Componentes del sistema Lewe.

a intervalos regulares de tiempo ajustable, se envían en tiempo real a un smartphone para su posterior almacenamiento. Los pines 8, 9, 11 y 13 (transferencia de datos) junto con los pines 3, 4 y 5 (botones) son utilizados por el shield LCD color. El brazalete, en ausencia de conexión Bluetooth con el smartphone, puede almacenar hasta 12 mediciones; tan pronto como se conecta,

envía las 12 mediciones. Todo el sistema se alimenta mediante un módulo de batería, distribuido por Futura Electrónica (código SOL20), que le permite recargarse a través de un puerto USB, y con el pequeño panel solar incorporado, se optimiza para la utilización al aire libre. Si es necesario, el mismo módulo puede servir para recargar el smartphone. El módu-

lo SOL20 proporciona una tensión de salida de 5V estabilizada, que se puede utilizar para alimentar el Arduino utilizando los pines +5V y GND; para hacer esto, sin embargo, es necesario cortar un cable USB, de modo que una extremo se conserva el conector USB para conectar al módulo de batería, mientras que en el otro extremo se conectan los cables directamente a los pines de alimentación +5V y GND de Arduino. Para evitar la manipulación del cable es posible usar el mismo cable utilizado para la programación del Arduino, pero en lugar de conectarlo al puerto USB del PC será insertado en el módulo SOL20: algo un poco engorroso, pero muy funcional. El mismo sistema se puede utilizar para alimentar con batería cualquier otro proyecto, con la ventaja de disponer de doble modalidad de carga: a través de USB o con un panel solar integrado.

FIRMWARE

El firmware se ha desarrollado sobre el IDE de Arduino utilizando diferentes librerías estándar y alguna externa. La librería color LCD se utiliza para controlar el shield LCD color de Sparkfun, la librería HashMap se usa para gestionar las matrices de cadenas (strings) y, finalmente, la librería Wrapper se usa para facilitar la extracción de valores de cadena encapsulados en la matriz. Para la utilización del shield Bluetooth se ha creado la librería Software SerialJack, basada en SoftwareSerial del SDK de Arduino, para implementar una comunicación más segura gracias a la adición del protocolo de comunicación Jack que garantiza una conexión fiable entre el emisor y el receptor, por la inclusión un sistema de acuse de recibo de los mensajes. Este protocolo se ha desarrollado para ser compatible con

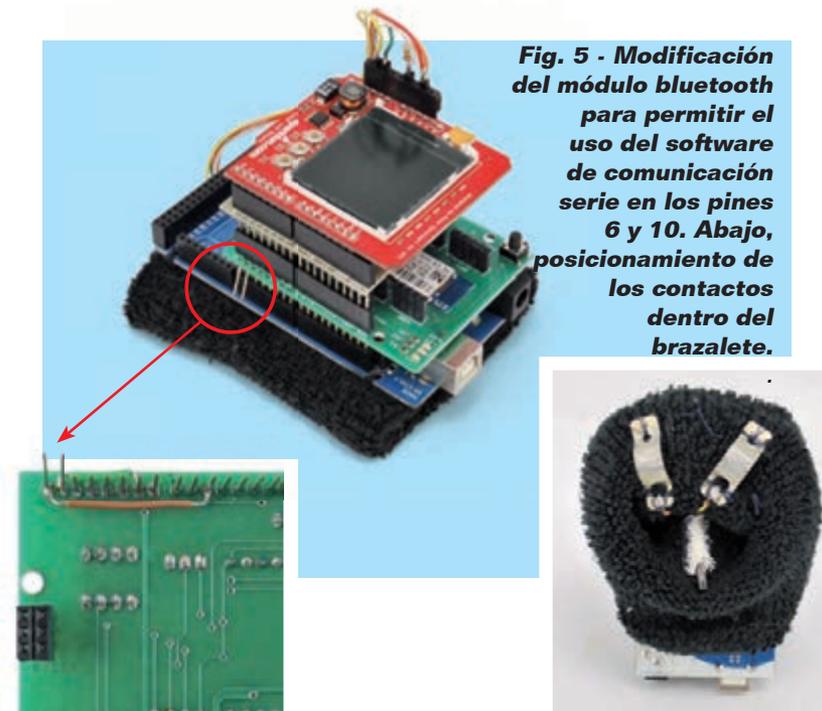


Fig. 5 - Modificación del módulo bluetooth para permitir el uso del software de comunicación serie en los pines 6 y 10. Abajo, posicionamiento de los contactos dentro del brazalete.

Fig. 6
LCD
mostrando
el reloj..

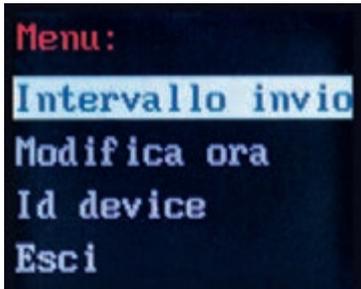


Fig. 8 - Mostrando el menú

cualquier medio de transmisión e implementa las siguientes funciones **boolean available (void)**, **String getMessage (void)** y void **SendMessage (String)**. La función boolean available (void) devuelve un valor booleano verdadero si hay mensajes disponibles para su descarga durante la transmisión, de lo contrario es falso. String getMessage (void) devuelve en formato cadena, una cada vez, los mensajes recibidos desde el medio de transmisión. Por último, SendMessage (String) se ocupa de enviar por el medio de transmisión los mensajes, como cadenas, los datos tratados por el protocolo Jack.

Éste último proporciona una clase que actúa como un contenedor para los datos llamado JData y utiliza las clases de contenedor para almacenar los datos y conservar el tipo. El método **send(JData)** del protocolo se ocupa de crear un mensaje de tipo cadena con el mensaje real para ser enviado con un identificador único del mensaje, que está compuesto por la fecha y hora de adquisición, el valor de la temperatura y el de la transpiración.

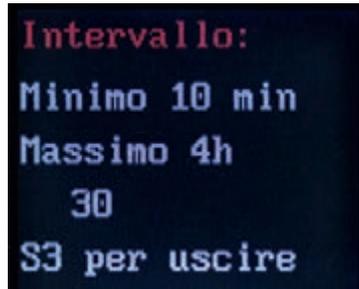


Fig. 9 - Configuración

El mensaje se envía entonces al receptor para ser procesado. Una vez recibido, se selecciona la función **available()** la cual pasará el mensaje al función **getMessage()** que salvará el mensaje recibido. Para procesar el mensaje recibido se utiliza el método **execute(String)**, que comprobará si el mensaje es conforme al estándar,

Fig. 7 - LCD
mostrando los datos.

después extraerá el código único y los datos contenidos en el mensaje; para confirmar la recepción del mensaje, se envía al remitente el identificador único. Si el mensaje no llega o es corrupto, no habrá ningún acuse de recibo y el transmisor entenderá que el mensaje se ha perdido y lo tendrá que retransmitir. De este modo, incluso una caída momentánea de la conexión Bluetooth no comportará la pérdida de información, que será devuelta tan pronto como se restablezca la conexión. El código de tiempo incluido en el mensaje permitirá la reconstrucción de la secuencia temporal correcta. Para compilar correctamente el sketch de Arduino hay instalar las librerías adicionales ColorLCDShield, HashMap, Jack, RTCLib, SoftwareSerialJack y Wrapper, todas ellas proporcionadas con los archivos de este proyecto; el número considerable de líneas código implica que el sketch ocupa una gran cantidad de espacio que sólo puede funcionar con una tarjeta Arduino mega 2560

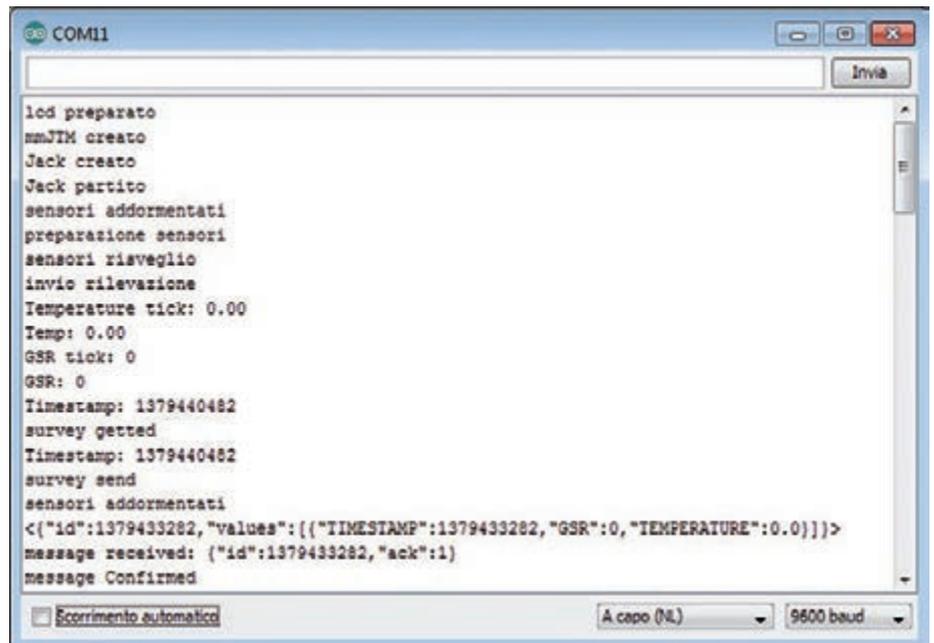


Fig. 10 - Tráfico de datos recibidos en el monitor serie.

que dispone de 256 KB de espacio para el programa; un Arduino UNO no dispone de suficiente capacidad para el sketch. Durante el funcionamiento, la pulsera controla constantemente a través de USB las cadenas enviadas y recibidas por Bluetooth, de modo que pueda verificar la corrección del protocolo utilizado; el campo "id" representa el identificador del mensaje, valor clave utilizada para verificar la calidad de los mensajes recibidos. El módulo Bluetooth utiliza todos los ajustes por defecto y los puentes del shield están posicionados de la siguiente manera: AUTO=off, DEFAULT=off, MASTER=off y BAUDRATE=on. El módulo se ve obligado a trabajar como esclavo con velocidad de comunicación de 9600bps.

Si el módulo se ha programado previamente, es necesario reiniciarlo a valores de fábrica, función de la que se encarga el botón RESET situado en el módulo. Sin embargo es importante asignarle

un nombre que sea compatible con "lewe" o sea del tipo LEWE_idDevice, donde IdDevice es un número de identificación del dispositivo. Para este procedimiento hemos escrito un sketch adecuado con el que es posible asignar un nombre, distinto del predeterminado por defecto, al módulo Bluetooth.

El sketch se llama RN_42_serial-monitor.ino y permite la interconexión con el módulo RN-42 a través del monitor de serie de la placa Arduino. Después de cargar el sketch de Arduino, basta con enviar los siguientes comandos AT:

- establecer "sin salto de línea" en el monitor de serie y enviar \$\$\$ para entrar en modo comando (respuesta: CMD);
- establecer "NL es a la vez CR" en el monitor de serie;
- enviar D para tener la configuración como una respuesta (respuesta: AOK);
- enviar SN, LEWE_01 para establecer el nombre del módulo (respuesta: AOK);
- establecer "NL es a la vez CR"

sobre monitor de serie;

- enviar --- para salir del modo de comando (respuesta: END). Dejando en ejecución éste sketch es posible interconectar con un dispositivo Android y probar un intercambio de datos simplemente instalando un APP con función de Bluetooth SPP, es decir, capaz de manejar las comunicaciones en serie a través de BT. Hemos hecho algunas pruebas con la aplicación "Bluetooth spp pro" gratuita, con la cual es posible enviar y recibir cadenas. Ésta APP también permite la modalidad "Button" con la que se pueden programar los botones para enviar cadenas predefinidas, útil para gestionar las salidas de Arduino u órdenes más avanzadas.

APLICACIÓN PARA SMARTPHONE

La aplicación para Smartphone Android constituye el verdadero corazón de Lewe, porque se ocupa de recibir los datos de la pulsera y de almacenarlos localmente en el dispositivo y, si se configura correctamente, subirlos a la nube. Se trata de una actividad principal en la que se muestran los últimos datos recibidos de la pulsera y una actividad secundaria que muestra el gráfico que contiene todos los datos almacenados. Al hacer clic sobre el icono del engranaje, se puede entrar en la configuración de la aplicación donde se selecciona el brazalete al que conectarse y se configuran los datos de acceso a la nube. La aplicación ha sido realizada en Java con el apoyo de Eclipse provisto de ADT Plugin, que permite el desarrollo de aplicaciones de Android; para los gráficos es preferible utilizar la aplicación AchartEngine, disponible bajo licencia Apache 2.0. La aplicación hace uso de cuatro servicios que trabajan en segundo plano:

- LeweService;

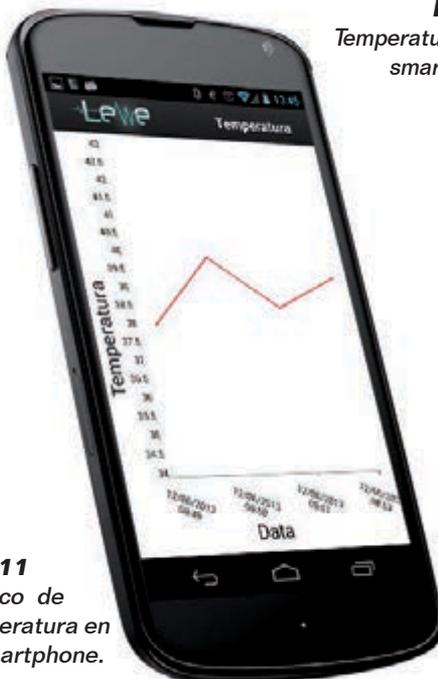


Fig. 11
Gráfico de temperatura en el smartphone.

Fig. 12
Temperatura en el smartphone.



- LewebluetoothService;
- LeweDatabaseService;
- LeweWebCloudService.

LeweService es el servicio principal de la aplicación que se inicia en el primer arranque y permanece en segundo plano hasta su cierre, pone en marcha los restantes servicios y se ocupa de finalizarlos una vez recibida la orden para apagar. El servicio se comunica directamente con los demás servicios LeweWeb-CloudService y Lewebluetooth-Service; además de esto, envía las ordenes de conexión y desconexión a los servicios antes mencionados, que se comunican con el mundo exterior.

LewebluetoothService explota la clase BluetoothChatService para la recepción de datos y ha sido modificado para añadirle la posibilidad de recuperar la conexión con el dispositivo Bluetooth en el caso de que falle, y además se ha hecho compatible para operar con el protocolo Jack. LeweDatabaseService se ocupa de gestionar la base de datos SQLite de los datos adquiridos. LeweWebCloudService gestiona la comunicación entre la aplicación y la nube, en particular se ocupa de enviar los datos a la nube a través de HTML post y verificar la lectura correcta a través de la confirmación de recepción de los datos, tal como se prevé en el protocolo Jack. La nube permite el acceso en línea a sus datos desde cualquier lugar del mundo con la utilización de credenciales de acceso a tu propio espacio personal disponible de forma gratuita; si no dispones de una cuenta, hay un servicio de registro. Los datos se representan gráficamente y en forma de tabla clásica (Fig.13); también hay un módulo para hacer una búsqueda por fecha. Si no se utiliza ningún parámetro, se muestra

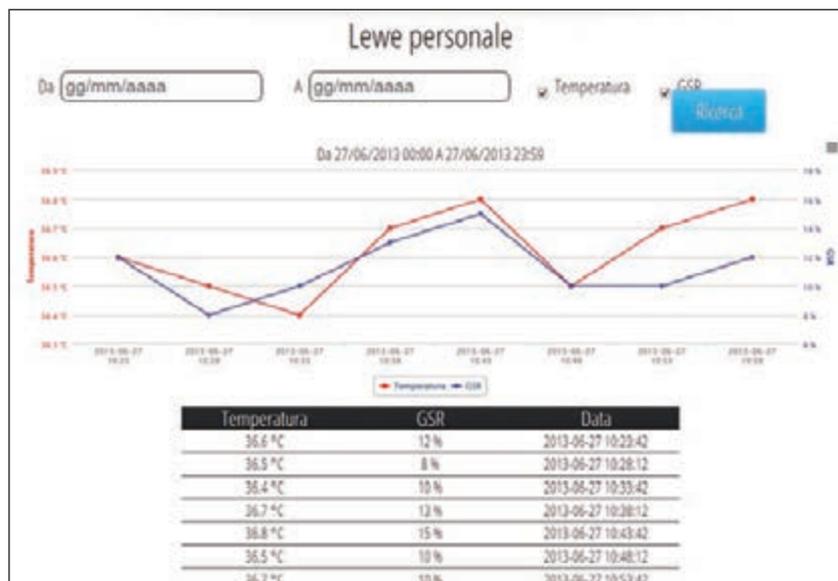


Fig. 13 - Captura de la nube.

automáticamente los datos relativos al último día de utilización de la pulsera. Mediante la configuración de la aplicación con los datos de acceso a tu espacio personal, puedes comenzar a cargar los datos biométricos recogidos por el brazalete. También es posible facilitar el acceso a sus datos a otras personas compilando con su username (nombre de usuario) el módulo presente en el página "Publico mi Lewe". Al acceder a la página "Lewe de un amigo" es posible, ver los nombres de los usuarios que nos han autorizado y, haciendo clic sobre uno de ellos, veremos los datos organizados de una manera similar a la página "Mi Lewe". La carga de los datos ha sido realizada por una sub-nube a la cual no se puede acceder directamente, sino que se lleva a cabo después de la autenticación efectuada por la aplicación Android para smartphone.

Tal sub-nube ha sido realizada con la ayuda del framework Azzurro y posee una implementación del protocolo de comunicación Jack para la recepción y la confirmación de los datos a

almacenar.

Para activar el brazalete es suficiente alimentarlo mientras la aplicación Lewe.apk está instalada en un dispositivo Android. En primer lugar es necesario vincular el brazalete al smartphone accediendo a la pantalla de configuración, símbolo de engranaje en la esquina superior derecha (Fig.14). Desde el menú ajustes, es posible configurar la comunicación con el brazalete y la nube de internet (Fig. 15). Al hacer clic sobre el botón Lewe y sucesivamente sobre el botón ON, se inicia la búsqueda de dispositivos Lewe: en la casilla inferior aparecen los nombres de las pulseras que la aplicación detecta (Fig.16); gracias a un control del nombre de dispositivo Bluetooth, sólo se muestran los que responden al patron LEWE_idDevice, donde IdDevice es un número de identificación del dispositivo. Seleccionando la pulsera a la que se quiere conectar, se establece la conexión y la aplicación se ejecuta. Este paso sólo es necesario para el primer arranque de la aplicación, o cuando se cambia de dispositivo,



Fig. 14 - Aplicación Lewe.



Fig. 15 - Configuración de la app.



Fig. 16 - Configuración de la app.



Fig. 17 - Configuración de la nube.

en cuanto la aplicación memoriza el dispositivo Bluetooth al cual se conecta, intentará conectarse automáticamente cada vez que se abra la aplicación. Para la nube, es necesario hacer clic sobre "Web Cloud" y escribir la dirección de correo electrónico y contraseña en los campos adecuados (los que dan acceso a la nube web online) y marcar "Usar Web Cloud" (Fig. 17). La aplicación también puede funcionar sin la nube y por tanto su configuración es opcional. El sistema ha sido ampliamente probado con smartphones Samsung Galaxy Nexus, Galaxy Nexus 4 e Galaxy S2 con sistema operativo Android 4.0. Es posible ver la nube online accediendo a la página web <http://www.lewe.tk>. Para el acceso utilice el formulario haciendo clic sobre el botón "login", usando las credenciales: email prova@prova.it y contraseña Prova123.

CONCLUSIONES

No siempre es fácil desarrollar nuevas cosas, y mucho menos cuando pueden afectar a un gran número de personas y a una amplia gama de aplicaciones. Las nuevas tecnologías, especialmente los teléfonos móviles, ofrecen todavía un margen considerable para la innovación, más aún en los casos en que el software se encuentra con el hardware, y en esto Arduino nos ha enseñado mucho..



EL MATERIAL

Todos los componentes utilizados en estos proyectos son fáciles de encontrar. El cargador solar de baterías de 1200 mAh cod. SOL20 cuesta 22,00 euros, el módulo Bluetooth cod. FT1032M cuesta 34,00 Euros, el Arduino Mega2560 REV3 (Cod. ARDUINO - MEGAREV3, sale por 51,00 Euros y el RTC cod. RTCSHIELDKIT por 11,00 Euros.

Los precios incluyen IVA.
Los gastos de envío no van incluidos.
Puede hacer su pedido en:
pedidos@nuevaelectronica.com

Dotar al Arduino de una placa y una librería de datos para conseguir que se comunique con el mundo exterior a través de WiFi.

WI-FI E INTERNET PARA ARDUINO

DANIELE DENARO

La placa Arduino ha supuesto una verdadera revolución en el campo de la electrónica, demostrándose particularmente adecuada para los no expertos y al mismo tiempo lo suficientemente versátil como para ser capaz de manejar una amplia gama de aplicaciones en diversos campos. La hemos visto, por ejemplo, afrontar tareas complicadas, como la conexión con el mundo exterior, bien usando las simples interfaces básicas (serie, I²C-Bus), bien con lo que llamamos conectividad real, es decir, interfaces inalámbricas y red Ethernet. En ambos casos, los límites de Arduino se manifiestan con claridad, porque la placa del equipo Arduino, como sa-

bes, se basa en un hardware mínimo, mientras las conexiones con las redes y aún más la conectividad a Internet, requieren una gestión asincrónica y protocolos de comunicación esencialmente multitarea. Por esta razón, la conexión de Arduino con Internet siempre ha sido un poco difícil, tanto como para empujar a sus creadores a implementar un sistema híbrido como el YÚN, donde el procesador Atheros tiene la función principal de administrar las comunicaciones. YÚN sin embargo, no es la única solución. De hecho como demostraremos en estas páginas, Arduino puede disponer de conectividad a Internet a través de una placa dotada con un

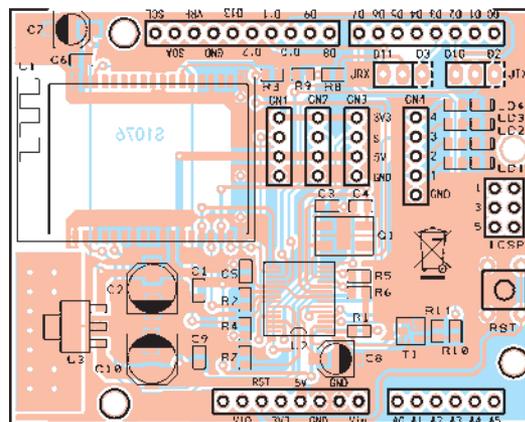
[Plano de **MONTAJE**]

Lista de materiales

- R1, R4, R11: 10 kohm (0805)
- R2, R10: 4,7 kohm (0805)
- R3: 100 kohm (0805)
- R5: 1 Mohm (0805)
- R6: -
- R7: -
- R8: 1 kohm (0805)
- R9: 1,5 kohm (0805)
- R12÷R15: 330 ohm (0805)
- C1, C5, C6, C9: 100 nF multicapa (0805)
- C2, C10: 220 µF 6,3 VL electrolítico (D)
- C3, C4: 22 pF cerámico (0805)
- C7, C8: 10 µF 35 VL electrolítico (B)
- U1: MRF24WB0MA/RM
- U2: MCW1001A
- U3: TC1262-3.3 (SOT-223)
- T1: BC817
- Q1: Cuarzo 8 MHz (HCX-7SB) > RS: 675-4703
- RST: Micro interruptor
- LD1÷LD4: LED verde (0805)

Varios:

- Tira de pines macho 3 vías (2 pzas.)
- Tira de pines macho 4 vías (3 pzas.)
- Tira de pines macho 5 vías (1 pzas.)
- Tira de pines macho/hembra 6 vías (1 pzas.)
- Tira de pines macho/hembra 8 vías (2 pzas.)
- Tira de pines macho/hembra 10 vías (1 pzas.)
- Tira de pines macho/hembra 3 vías (2 pzas.)
- Puente (2 uds.)
- Circuito impreso



WiFi; el que se describe aquí es una versión mejorada del anterior, que se basaba en un módulo Microchip que ahora está fuera de producción, y dispone de hardware que le permite gestionar por sí mismo el protocolo TCP/IP. Más exactamente, la diferencia sustancial entre la placa antigua y esta nueva está en que en la primera era el Arduino (mediante una librería adecuada) quien controlaba el módulo WiFi y manejaba el protocolo TCP/IP, mientras que en este nuevo circuito, hemos insertado un procesador del interfaz que se ocupa de la gestión del protocolo TCP/IP, descargando así al Arduino de una tarea que, por la experiencia del primer módulo, llegó a ser

demasiado exigente. Sin embargo, una tarjeta de red llega a ser realmente útil si está controlada por un software que simplifique tanto como sea posible su uso; por este motivo, hemos dotado a la tarjeta de una librería que teníamos intención de utilizar en Arduino, con un grado de simplificación mayor que la de un hardware más complejo (aunque más flexible) como la de YÚN. Llegando a la descripción del hardware, podemos ver que el shield (que tiene el mismo tamaño que una palca de Arduino) se basa esencialmente en un módulo WiFi MRF24WB0MA y un procesador MCW1001A, ambos Microchip. La fuente de alimentación es

de 3,3V (obtenida mediante un regulador de tensión alimentado desde la línea 5V de Arduino) pero el procesador soporta también niveles TTL. El módulo de radio tiene la antena integrada. El procesador gobierna el módulo WiFi a través del bus SPI (líneas SDO, SDI, SCK) y otras líneas de control como CS, INT, RST, HIBERNATE, WP; de lado del host (es decir, Arduino) el procesador está conectado a una interfaz en serie de dos hilos en lógica TTL (RX,TX). Esto nos obliga a utilizar un puerto serie simulado por software con una versión personalizada de la librería SoftwareSerial. La personalización es necesaria a causa de la

WiFi 5 GHz + Bluetooth

(después de descomprimirlo) en el directorio "libraries" del IDE de Arduino, al igual que las otras librerías; se debe utilizar incluyendo en el sketch el archivo <MWiFi.h >.

Se llama como un objeto y se puede utilizar; la primera función llamada es begin (), que sirve para inicializar la tarjeta:

```
#include <MWiFi.h>
MWiFi WIFI;
void setup()
{
  WIFI.begin();
  :
```

El arranque de la tarjeta determina el encendido del primer LED. Probamos, ahora, a conectar Arduino en red utilizando un punto de acceso, que puede ser el router WiFi de casa, por ejemplo asignando el nombre (SSID) "D-Link-casa":

```
WIFI.ConnSetOpen("D-Link-casa");
```

válido en el caso de que la red no esté protegida, o bien:

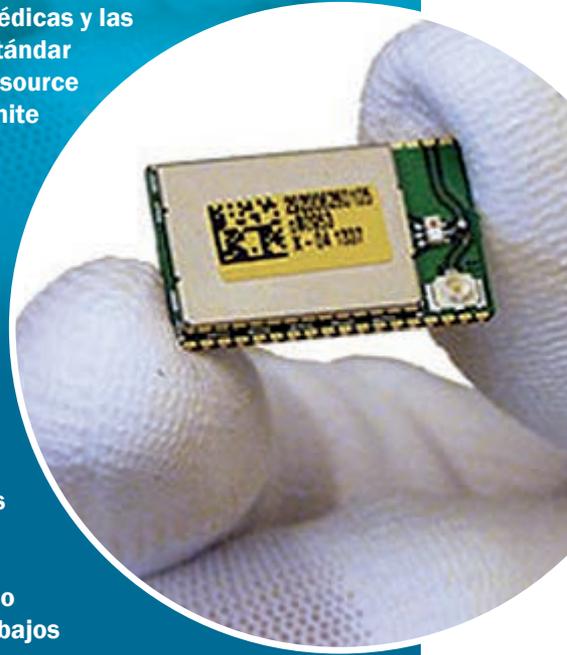
```
WIFI.ConnSetWPA("D-Link-casa","mipassword");
```

en el caso de que la red tenga protección WPA con contraseña "mipassword". Las funciones anteriores se utilizan para preparar la conexión de la tarjeta; pero la verdadera y real conexión se obtiene llamando a la función:

```
WIFI.Connect();
```

Si la conexión se realiza correctamente, se enciende el segundo LED. Hay que tener en cuenta que en el caso de que la red esté protegida, la conexión puede tardar más de medio minuto, ya que el controlador de la tarjeta debe codificar la clave con la contraseña. La versión actual de la librería proporciona un reset automático de Arduino en caso de que la conexión se pierda;

ConnectBlue presenta OWL355, un módulo WLAN capaz de funcionar tanto en la banda clásica a 2,4 GHz, como en la nueva a 5 GHz, así como interconectar con cualquier estándar Bluetooth. Diseñado para el Internet de las cosas, las aplicaciones biomédicas y las industriales, supera el actual estándar IPC que es AQL. Gracias al open source Linux host driver, el módulo permite implementar con facilidad una conexión wireless en cualquier dispositivo electrónico, proporcionando hardware modular homologado, con certificación EMC, calificación Bluetooth, una amplia gama de antenas. Entre otras características, señalar la tecnología Castellation Package que utiliza los relieves metálicos a los lados del módulo para hacer mucho más fácil la soldadura (también manualmente) permitiendo el uso del módulo en aplicaciones con bajos volúmenes de producción.



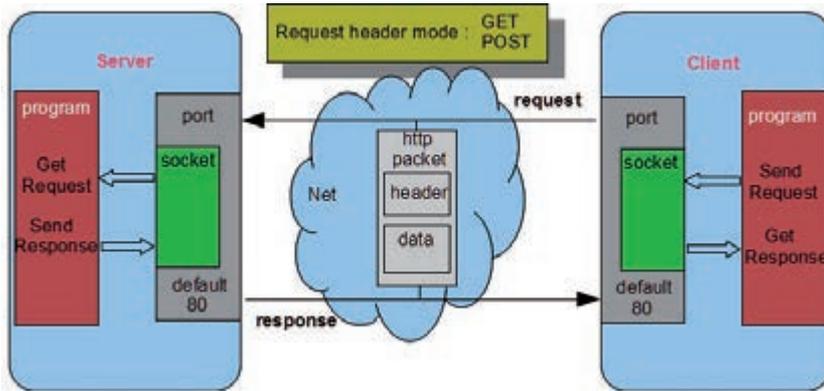
la restauración automática se produce incluso si los errores son detectados por el controlador. De este modo, se evita un posible bloqueo y el sistema puede ser "unattended" (funcionar sin supervisión humana). También se han añadido funciones de conexión directa, que preparan la conexión y la ejecutan en un único paso. Además se ha añadido la posibilidad de generar la clave de contraseña numérica, para utilizar sucesivamente ésta en lugar de la contraseña; de hecho el acceso con contraseña, como necesita cada vez la elaboración de la clave numérica, puede tardar hasta un minuto, mientras que el acceso con clave es muy rápido. El router asigna al Arduino una dirección IP dinámica, porque ese es el comportamiento por defecto de la tarjeta;

pero si se desea se puede imponer una dirección IP fija. La dirección asignada puede ser necesaria para una función de la librería. En este punto elegimos hacer funcionar Arduino como servidor: primero llamamos a la función **openServerTCP()**, que crea el oyente sobre un puerto determinado (por ejemplo 5000), y luego ponemos en bucle la recepción de una posible solicitud de link (enlace):

```
int ssocket=WIFI.openServerTCP(5000);
void loop()
{
  int csocket=WIFI.
  pollingAccept(ssocket);
  :
```

Las variables enteras ssocket y csocket son de las referencias (handle), respectivamente, para el servidor y la conexión del socket con cualquier equipo que quie-

Fig. 2 - Comunicación HTTP.



ra efectuar el link. La función **pollingAccept()** devuelve un número menor de 255 si se ha solicitado el enlace, o 255 si no hay ninguna solicitud de conexión entrante. En el caso de que la conexión haya sido establecida podremos enviar o recibir mensajes haciendo referencia a este csocket. Por ejemplo, para recibir un registro, que es una cadena que termina con un salto de línea, puede utilizar la función:

```
char *line=WIFI.  
readDataLn(csocket);
```

Será restituído una "cadena terminada null" pero sin el line-feed. En este caso, la librería utiliza un buffer predefinido de 81 caracteres (pero su longitud puede ser modificada en su definición) por eso no es necesario que nos proporcione dicha información. Sin embargo, hay otras posibilidades. Si queremos responder podemos utilizar la función:

```
WIFI.writeDataLn(csocket, answer);
```

La variable que llamamos answer (respuesta) corresponde a un buffer de char, pero tiene que ser una "cadena terminada en null". Utilizar un "cadena terminada en null" significa que nosotros no proporcionamos la longitud de los caracteres útiles en la matriz, porque la función la calcula automáticamente, pudiendo basarse sobre el carácter de ter-

minación null. La mayor parte de las funciones en C administran y producen este tipo de cadena, que no debe confundirse con el objeto String presente también en el lenguaje de referencia de Arduino. De este simple modo hemos establecido y utilizado una conexión WiFi con un equipo remoto. En la librería, entre los ejemplos indicados, hay un ejemplo de servidor llamado *CommandServer*, que permite controlar a Arduino utilizando un programa del tipo *telnet* en el equipo remoto. Para simplificar las pruebas se añadió un programa de Java que funciona como *telnet*. Si, en cambio, quisiéramos hacer actuar a Arduino como un "cliente" que se conecta a un servidor, la situación sería aún más simple,

porque tenemos que crear solo un socket de enlace, así:

```
int csocket=WIFI.openSoc --  
kTCP("192.168.1.2",5000);
```

Y si csocket es válido (menor de 255), quiere decir que se ha establecido la conexión con el ordenador a la dirección 192.168.1.2 sobre el puerto 5000. En este punto podemos utilizar las funciones de lectura y escritura vistas anteriormente. Entre los ejemplos, hay uno (*SendData*) que se enlaza a un servidor remoto para enviar a intervalos regulares las lecturas de los sensores; en este caso es necesario un programa de servidor instalado en el equipo remoto. Para facilitar las pruebas, junto con la librería se proporciona un programa Java que recibe los datos y los descarga en un archivo añadiendo una marca de tiempo (para obviar la falta de un RTC, Real Time Clock, en Arduino). Además de estas características básicas, la librería cuenta con todas las funciones necesarias para definir diversos parámetros como la dirección de enmascaramiento de la red (255.255.0.0 es la predeterminada), la dirección de Gateway, la lectura del código MAC de la



tarjeta, etc. En particular, están presentes las funciones para detectar puntos de acceso presentes y visibles en el entorno.

Por ejemplo, para detectar todas las redes presentes, se usa la instrucción:

```
int nn=WIFI.scanNets();
```

La variable entera nn contendrá el número de redes detectadas. En cambio la función:

```
char net=WIFI.getNetScanned(i);
```

restituirá las características (bajo la forma de registros) del “i-esima” red detectada. Por último, por conveniencia, se proporciona una función que restituye el nombre de la red no protegida que proporciona la señal más potente. Nos referimos a la documentación de la librería para la descripción de todas las funcionalidades disponibles. La librería contiene un *help* (ayuda) y está documentada en los archivos de código (en particular los archivos *.h*). Pero la librería no se limita a la simple gestión de conexión y socket; de hecho, en ella se incluye una clase derivada (y por lo tanto especializada) que administra el protocolo HTTP. Éste último es un protocolo que proporciona siempre una solicitud y una respuesta; tanto la solicitud

Listado 1

```
WEBRES rs[8]=
{
  {"/index",pindex},
  {"/Analog",panalog},
  {"/RDigital",rdigital},
  {"/Wdigital",wdigital},
  {"/wdig",wdig},
  {"/Pwm",pwmpage},
  {"/PwmSet",pwmsset},
  {"/End",sessend}
};

void loop()
{
  WIFI.getRequest(csocket,8,rs);
  :
```

Listado 2

```
prog_char pagerdigital[] PROGMEM=
:
"<tr>"
"<td><div align='center'>@</div></td>"
"<td><div align='center'>@</div></td>"
"<td><div align='center'>@</div></td></tr>"
:

void rdigital(char *query)
{
  char *val[3];
  if(digitalRead(4)) val[0]=ON;else val[0]=OFF;
  if(digitalRead(5)) val[1]=ON;else val[1]=OFF;
  if(digitalRead(12)) val[2]=ON;else val[2]=OFF;
  WIFI.sendDynResponse(csocket,pagerdigital,3,val);
}
```

como la respuesta hacen viajar por la red paquetes formados de algunos encabezamientos (header) y datos reales (tales como las páginas html, imágenes, video o también simple texto). Para descargar al usuario de todos estos problemas, la librería de HTTP se molestó en dar forma a estos paquetes utilizando, además, la modalidad PROGMEM, es decir, la posibilidad de colocar en el área Flash constantes de programa y en particular los textos. Hay que tener en cuenta que el protocolo HTTP es un protocolo textual: utiliza únicamente caracteres. El uso de la memoria Flash para los textos permite ahorrar espacio en la memoria RAM de Arduino.

LA LIBRERÍA HTTP

Siendo una clase derivada de *MwiFi*, la librería HTTP hereda todas las funciones de *MwiFi*; sin embargo, si desea utilizar las nuevas funciones es necesario incluir el archivo *HTTPLib.h* (en lugar de *MwiFi.h*) y crear una instancia de un objeto HTTP:

```
#include HTTPLib.h HTTP WIFI;
```

Respecto a la conexión con un punto de acceso y a la gestión del socket, todo permanece como antes (tal vez hora escogeremos el puerto 80); pero también esta vez hay que decidir si hacer funcionar al Arduino como servidor (es vez del servidor Web) o como

cliente que se accede a un servidor de aplicaciones Web (como Tomcat, GlassFish, Jboss, PHP, etc.). Supongamos que desea crear un servidor Web para ser consultado por cualquier browser (navegador): para hacer funcionar a Arduino como un servidor Web tenemos que preparar los recursos que él puede poner a disposición, es decir, las páginas html de respuesta. Estas páginas se almacenan en áreas PROGMEM por los motivos mencionados anteriormente. Por ejemplo:

```
prog_char pageindex[] PROGMEM=
"<html><head>"
"<title>Index</title>"
```

En este punto, tenemos que conectar estos buffer de memoria con los nombres de los recursos invocados a través del browser. Los nombres de los recursos son la parte de la ruta de acceso local de la dirección URL (o URI), que es, en esencia, el nombre del archivo en la URL completa (por ejemplo: `http://www.mio.es/mipag.html`). En este contexto, los recursos mínimos para ser invocados se identifican sólo por su nombre sin extensión. Así que se trata de asociar una página memorizada con su nombre de Internet correspondiente (por ejemplo `"/indice"`). En realidad esta página no lo enviará sólo puesto que necesita enlazar el nombre del recurso con una función que se ocupará de enviarlo. Para hacer el mecanismo

Listado 3

```
void pwmset(char *query)
{
    char *pwmval;
    pwmval=WIFI.getParameter(query, strlen(query), "PWM10");
    if (pwmval!=NULL)
    {int pv; sscanf(pwmval, "%d", &pv);
      analogWrite(10, pv); d10=pv;}
    pwmval=WIFI.getParameter(query, strlen(query), "PWM11");
    if (pwmval!=NULL)
    {int pv; sscanf(pwmval, "%d", &pv);
      analogWrite(11, pv); d11=pv;}
    pwmpage(query);
}
```

lo más automático posible, se ha preparado una estructura o más exactamente un *typedef* llamado WEBRES; esta estructura se forma de la unión de dos campos: el nombre del recurso y el nombre de la función (que en C corresponde a una dirección). Se trata de formar tantas parejas nombre-función para pasar a la función **getRequest()**, que se ocupará de poner en marcha la función correcta (función call-back), o enviar un mensaje estándar "Not Found" de respuesta, en caso de que el nombre no coincida con ninguno de los predispuestos. En el Listado 1 se muestra un ejemplo de la construcción de un conjunto de 8 estructuras WEBRES y su inclusión en la llamada a la función **getRequest()**. Cada segundo campo de la estructura corresponde a la función call-back que **getRequest()** lanzará. La función de call-back (devolución de llamada) se tendrá que preocupar de enviar el buffer correspondiente a la página elegida y su prototipo se espera que sea de tipo void (es decir, de retorno nulo) y tiene un solo argumento; en la práctica, un puntero de una cadena terminada en cero suministrada por la persona que llama (véase a continuación):

```
void pindex(char *query)
{
    WIFI.sendResponse(csocet, pagei
ndex);
}
```

En resumen: **getRequest()** puesto en el bucle, se hará cargo de

toda la gestión de la solicitud y detectará la modalidad utilizada GET o POST, comportándose en consecuencia (los datos están de diferente modo) y lanzará la función correspondiente a la solicitud (o el mensaje "Not Found"). La función de call-back **pindex()** descrita anteriormente, sin embargo no hace más que enviar en respuesta una página html estática, que se define de una manera fija.

El servidor web Arduino, definido así, no es muy útil, porque se supone que se puede utilizar para leer los valores proporcionados por los sensores (Fig. 4) o activar salidas (Fig. 5); para lograr esto, la página html de respuesta tiene que construirse en el momento, insertando los valores que se quieren leer. Tiene que, en definitiva, ser una página dinámica. Pero es demasiado costoso de construir, dentro de la función call-back, toda la página entera; para simplificar la tarea se prevé definir la página "única" como una página estática, pero puede ser capaz de entrar en el interior de las etiquetas (etiquetas marcadoras de posición) en la posición que desea completar en el momento. Para este fin existe una función alternativa a **sendResponse()**, la función **sendDynResponse()** que se encarga de encontrar y sustituir las etiquetas y enviar la página. La sustitución pasa secuencialmente recorriendo una matriz de cadenas dispuestas

sobre el momento; a continuación, la primera etiqueta encontrada se sustituye con la primera cadena de la matriz, y así sucesivamente. La etiqueta utiliza el carácter @; se debe utilizar sólo una, independientemente de la longitud de la cadena que será reemplazada. En el ejemplo que se encuentra en el listado 2, se sustituyen tres etiquetas por el mismo número de cadenas que representan los valores de tres entradas digitales. A la función **SendDynResponse()** se debe pasar una matriz de cadenas

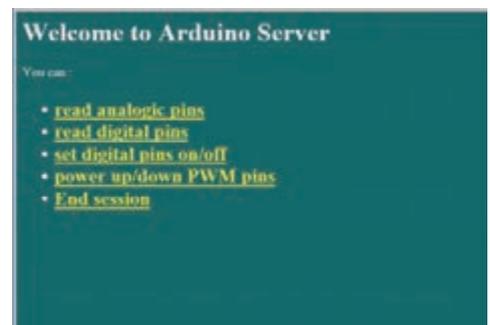


Fig. 3 - Página de inicio del servidor Web.

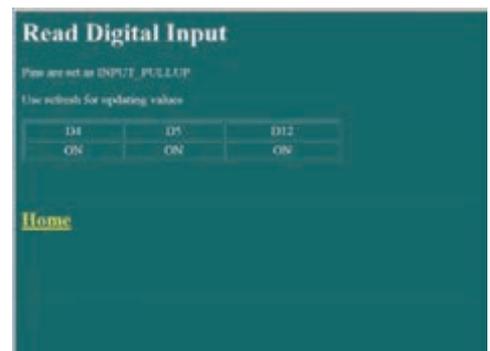


Fig. 4



Fig. 5

Listado 4

```
#include <WiFi.h>
WiFi WiFi;
setup()
{
    WiFi.begin();
    WiFi.ConnectWPAwithPSW("MioAcp","pippo");
    server=WiFi.openServerTCP(5000);
}
loop()
{
    if (!OpenCom)
        socket=WiFi.pollingAccept(server);
    if(socket<255) OpenCom=true;
    if (OpenCom)
        record=WiFi.readDataLn(socket);
    WiFi.writeDataLn(socket,".....");
}
```

y de su tamaño. Para asegurarse de que Arduino actúa en consecuencia, los siguientes comandos son lanzados desde el navegador (por ejemplo mediante botones de formulario), es necesario leer los datos enviados por la solicitud junto al nombre del recurso. La situación es diferente si la petición llega en forma de GET en lugar de POST en un formulario (los dos métodos fundamentales del protocolo HTTP). Es, en todo caso, para utilizar finalmente ese argumento pasado a la función de call-back de **getRequest()**. En el primer caso, los datos están representados por parejas nombre-valor que identifican un parámetro. Los parámetros se añaden al nombre del recurso en un formato que codifica espacios y caracteres especiales y que podemos llamar cadena de consulta. La cadena de consulta se proporciona siempre (aunque tenga longitud cero) a la función call-back (en efecto es parte del prototipo). Entonces podemos utilizar la función **getParameter()** para recuperar el valor (siempre en forma de cadena) del parámetro con un cierto nombre (ver Listado 3). En el segundo caso, en cambio, la cadena de consulta contendrá valores que pueden estar en formato cadena de consulta (como en general hacen los formularios) o bien en un formato cualquiera.

De todos modos, hay que tener presente que el buffer que contiene la cadena de consulta lo proporciona la librería y tiene una longitud de 64 caracteres (pero se puede definir mediante la definición de `HTTPLib.h`). Los datos en exceso se pierden. Entre los ejemplos hay un servidor Web completo que le permite leer valores analógicos y valores digitales, activar y desactivar una salida digital y regular dos salidas PWM. El sketch es particularmente compacto (la mitad está constituida de las páginas html en `PROGMEM`) gracias a la automatización producida de las funciones **getRequest()** y **sendDynResponse()**. Si, en cambio, se quiere utilizar Arduino como cliente de un servidor de aplicaciones Web (o de una CGI más sencilla), utilizaremos las funciones **sendRequest()** y **getResponse()**. **SendRequest()** está efectivamente constituido de dos funciones separadas en base a la modalidad que se quiere utilizar: GET o POST. Si tuviéramos que utilizar **sendRequestGET()** proporcionaríamos tanto el nombre del recurso como los parámetros en una única cadena de consulta. En el caso de que utilizáramos **sendRequestPOST()** proporcionaríamos separadamente el nombre del recurso y los datos situados en un buffer de tipo cadena terminada en null, de este modo:

```
WiFi.addParameter(query,128,"/
TestClient",NULL); WiFi.
addParameter(query,128,"A1", sa1);
:
WiFi.sendRequestGET (csocket,query);
```

O bien así:

```
sprintf(rec,"%d %d %d
%d",an1,an2,d1,d2); WiFi.sendReques-
tPOST (csocket,"/ TestClient",rec);
```

En el caso de utilizar **sendRequestGET()**, la cadena de consulta se formaría con la ayuda de la función **addParameter()**. La primera vez que se inicia la cadena de consulta sería con el nombre del recurso (valor nulo), y luego con las parejas de nombre-valor para los parámetros individuales. La función **getResponse()**, es aquella que se utiliza para recuperar la respuesta desde el servidor. Ésta puede estar también formada por numerosos datos en varios formatos: desde la página HTML, a los datos en formato XML, JSON o csv (valores separados por comas). Si los datos no pueden ser contenidos en un único buffer, es posible volver a llamar a la función **getNextResponseBuffer()** en un bucle hasta que no retorne 0. En el Listado 4 hay un ejemplo elemental de conexión para recibir y enviar registros.



EL MATERIAL

Con la información contenida en este artículo, cualquier persona podría poner en práctica este proyecto. Sin embargo, hemos decidido ofrecer éste módulo WiFi ya montado y probado para dar a todos la oportunidad de dotar de conexión WiFi la propia placa de Arduino. El módulo ya montado, código FT1076M, tiene un precio de **56,00 Euros**.

Los precios incluyen IVA.
Los gastos de envío no van incluidos.
Puede hacer su pedido en:
pedidos@nuevaelectronica.com

LA 3DRAG SE CONVIERTE EN CNC PARA CREAR PCBs

Editado por REDACCIÓN

Como transformar nuestra impresora 3D en una fresadora CNC que podemos utilizar para crear circuitos impresos sin insoladoras, ni revelador ni, ni ácido, simplemente enviando al controlador archivo de diseño obtenido a partir del Gerber o de un plugin de Eagle.



Vista y comprobada la estabilidad y la precisión de nuestra impresora 3Drag, muchos nos habéis preguntado si es posible montar sobre el cabezal una fresadora, en lugar del sistema de deposición calentador/extrusor, a fin de obtener una máquina CNC. Pues bien, esto es posible, y no creo que sorprenda a nadie el que hayamos estado trabajando en ello para presentaros ahora el resultado con esta primera aplicación que seguro despertará un cierto interés en muchos de vosotros. De hecho, después de montar la fresadora y comprobar la calidad del trabajo, hemos tratado de hacer una aplicación un poco inusual pero indudablemente sugerente: fabricar los circuitos impresos por vía mecánica, en lugar de químicamente, como se nos ha enseñado en la escuela y como se hace normalmente en la industria. Esta técnica, alternativa a la canónica, puede sorprender por la calidad del resultado y resulta, si no más limpia (debido a que la incisión por el taladro implica la producción de una gran cantidad de virutas) al menos más respetuosa con el medio ambiente, ya que no implica la necesidad de utilizar, y por tanto de reciclar, soluciones de ácido (tales como el cloruro férrico), bases (el revelador a base de hidróxido de sodio), etc. El único defecto, si quiere verse así, es que el grabado de las pistas de cobre, debido a la forma del vértice de la fresa, produce una erosión mínima en el sustrato de fibra de vidrio o de baquelita del circuito impreso, lo que produce una

cierta disminución de la resistencia mecánica y, cuando se trata de circuitos de doble cara, se produce una leve disminución en la rigidez dieléctrica entre capas. En este artículo se describe cómo modificar mecánicamente la impresora 3Drag para transformarla en una máquina fresadora CNC y utilizarla para la producción de PCBs y los procedimientos para obtener el archivo G-Code, que necesita la máquina contornear las pistas y el taladrado de la placa, a partir de los archivos Gerber (y el uso del software de conversión PCBMill), o desde el proyecto de la placa realizado con el software de Eagle y convertido con un plugin especial.

PRIMER PASO: LA MECÁNICA

La operación sustancial consiste en la sustitución del cabezal de impresión por el soporte que soporta la fresadora; la herramienta eléctrica que hemos decidido utilizar es el PROXXON IBS/E. Para utilizar el 3Drag como fresadora CNC es necesario conocer el funcionamiento de la impresora y su software de gestión (Repetier-Host); puedes consultarlo en numerosos artículos dedicados a la impresión en 3D que se encuentran en números anteriores de *Electrónica In*. Para montar la fresadora PROXXON IBS/E es necesario imprimir primero el soporte con la 3Drag, cuyo archivo se puede descargar desde la web en <http://3dprint.electronica.in>.



Lo necesario para la modificación

Para transformar la impresora en una fresadora CNC es necesario en primer lugar tener la estructura, piezas metálicas y diversos elementos (visibles en la foto) de los que indicamos la lista completa.

Herramientas:

- Calibre centesimal.
- Llave fija de medida 10 - 11 mm.
- Llave Allen de medida 4 e 5 mm.
- Escuadra con lado de al menos 150 mm.



Minuterie metálica:

- 2 tornillos de cabeza cilíndrica M5x40.
- 1 arandela plana M5.
- 1 tornillo de cabeza cilíndrica M6x35.
- 2 arandelas planas M6.
- 2 arandelas dentadas M5.
- 1 tuerca auto bloqueante M6.
- 2 arandelas planas M5x20.

Varios:

- Frasora/taladro eléctrico PROXXON IBS/E (u otro modelo similar).
- Soporte para la fresadora impreso con el 3Drag (con collarín de la herramienta eléctrica adecuada).
- Adaptador para el comparador de montaje, impreso con el 3Drag (opcional).
- Iluminación Led (opcional; para iluminar el plano de trabajo es posible utilizar una lámpara común).
- Placa de vidrio de forma cuadrada de 200mm de lado y aproximadamente 3mm de espesor (también se puede usar un espejo estándar de forma cuadrada que se puede comprar en IKEA – modelo SÖRLI).
- Placa de forma cuadrada, de 200mm de lado y 10mm de espesor, en madera (MDF, contrachapado o similar) o material plástico (forex, espuma de poliuretano, etc.) sin deformaciones ni relieves.
- Cinta adhesiva de doble cara.

it/wp-content/uploads/2013/12/Supporto-mandrino-G-. El soporte, dotado de un collarín con un talador de 20mm, permite fijar a la estructura cualquier taladradora o fresadora que tenga mango cilíndrico del mismo diámetro. Para utilizar la 3Drag modificada como fresadora CNC, es necesario que el plano de trabajo esté exactamente a la misma distancia de la herramienta con el fin de obtener una incisión de igual profundidad en todo el circuito impreso; hay que realizar un ajuste particularmente preciso, como describiremos a continuación. Para este ajuste también se puede utilizar un comparador que montaremos sobre el mismo soporte de la herramienta eléctrica mediante un adaptador imprimible con la misma 3Drag y que, evidentemente, se hará antes de realizar cualquiera de

los cambios mecánicos descritos a continuación. El archivo de este adaptador está disponible en la página web <http://3dprint.elettronica.in.it/wp-content/uploads/2013/12/Adattatore-Comparatore-G-code.zip>.

La fresadora se alimenta directamente de la red a 220 V CA, por lo que no lo controla la 3Drag; tienes que encenderlo y apagarlo manualmente con el interruptor adecuado. Como ayuda, montaremos un anillo de LED blancos que iluminará la zona que se está trabajando.

La iluminación puede alimentarse con su propia fuente de alimentación de red capaz de proporcionar una tensión de salida continua de 15 Vcc, o bien puede utilizar la misma fuente de alimentación de la tarjeta del controlador del 3Drag.

Empezamos con el procedimien-

to para modificar la mecánica.

1. Inserta una tuerca galvanizada M5 en la parte superior del brazo hueco aprovechando la apertura existente entre el perfil de aluminio y el soporte de plástico frontal.
2. Desplaza el cuerpo del extrusor desatornillando los dos tornillos M5 que sujetan el soporte, pero primero haga una marca con el lápiz sobre el brazo de la impresora en la línea central del soporte que sostiene la extrusora, de modo que en el futuro puedas volver a montar la máquina de extrusión.

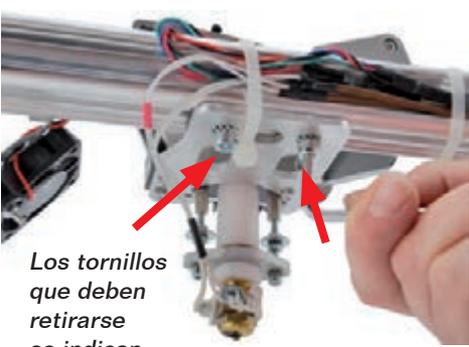


Iluminación LED.

3. Desatornilla con una llave allen los dos tornillos que unen el soporte en el brazo, luego afloje los tornillos que sujetan el ventilador.
4. Desplaza el cuerpo del extrusor sobre el brazo totalmente a la izquierda y lleve hacia el mismo lado también el ventilador: de este modo la boquilla de la extrusora estará lo suficiente alejada de la superficie de trabajo permaneciendo bloqueada en la estructura.



Insertión de la tuerca M5 en la parte superior del brazo.



Los tornillos que deben retirarse se indican mediante flechas rojas.



El soporte para la fresadora

5. Introduce en un agujero del soporte un tornillo M5, que debe ser atornillado en la tuerca M5 previamente introducida en el brazo.
6. Aprieta los tornillos para asegurar los elementos al brazo, y nuevamente los cables con bridas. Se debe cubrir con una película transparente el cuerpo entero de la extrusora para protegerla del polvo que se genera durante las operaciones de fresado de la PCB.
7. Toma el soporte de la fresadora e inserta un tornillo de cabeza cilíndrica M6x35 (completa con dos arandelas planas y una tuerca auto bloqueante M6) que servirá para apretar el collarín para sujetar la fresadora.
8. Inserta en el orificio de la iluminación un tornillo M5x40 con arandela dentada y, por la otra parte, la arandela plana. Inserta el tornillo de montaje de la iluminación en uno de los agujeros del soporte como se muestra en la foto; en el otro agujero, introduce el otro tornillo M5x40 con la arandela dentada y dos arandelas planas.
9. Monta bajo el soporte del brazo de la impresora el soporte que sostendrá la fresadora mediante los dos tornillos de cabeza cilíndrica M5x40 previamente insertados y atornillados a las dos tuercas cuadradas ya presentes en el brazo y previstas para fijar la máquina de extrusión; antes

de apretar los tornillos, posiciona el soporte de la fresadora con la línea central desplazada aproximadamente 1cm hacia la izquierda en comparación con la señal de referencia, marcada a lápiz sobre el brazo, de modo que durante el HOME, la posición de la fresadora estará fuera de la placa (esto evitará colisiones con el herramienta).

10. Introduce la herramienta eléctrica en el collarín y, después de haber fijado el soporte de plástico, y fija cualquier fresa en su mandril y asegúrate de que, en la posición de HOME, no hay contacto entre la herramienta y la superficie de trabajo, después aprieta los tornillos del soporte del eje.
11. Aplica sobre el plato de impresión, con un poco de cinta adhesiva de doble cara, una placa de vidrio de forma cuadrada de 200mm de ancho y aproximadamente 3 mm de espesor. Como alternativa,

también puede utilizar un espejo común como el modelo SÖRLI de IKEA.

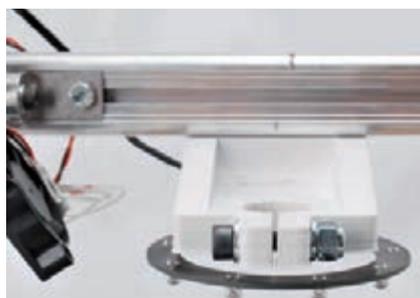
12. Une a la base del vidrio/espejo una placa "sacrificable" gruesa (aproximadamente de 1cm), que se utilizará durante la perforación de la PCB para permitir el descenso de la herramienta sin tocar el vidrio; puede ser de MDF, forex, madera contrachapada, etc... El espesor debe ser suficiente para mantener levantada la PCB y realizar el trabajo sin que se accione el interruptor de fin de carrera Z.

Para poder utilizar el 3Drag como fresadora CNC es también necesario actualizar el firmware de la placa electrónica, por lo que el eje Z debe asumir valores negativos con respecto a la posición cero; finalmente desde la sección Software de la web descarga firmware "Marlin versión V1 (Piatto riscaldato ON, LCD ON, Mill ON)" y sigue el procedimiento para su actualización.

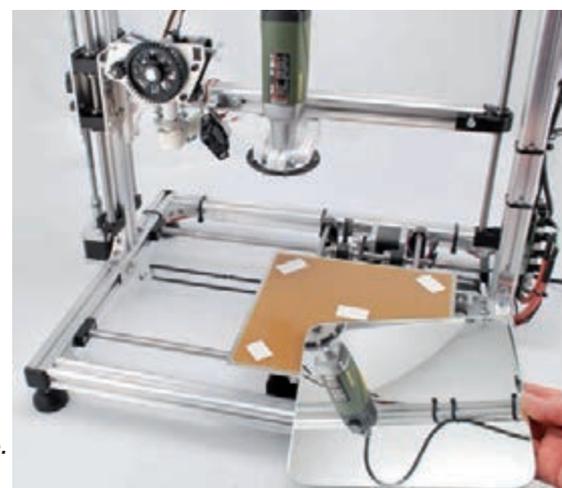
G-CODE APARTIR DEL ARCHIVO GERBER

Para crear una PCB con el 3Drag es necesario tener el archivo G-Code para realizar el contorno de las pistas y la las perforaciones del circuito impreso. En esta

Posición de la placa de vidrio.



La iluminación montado en el soporte.



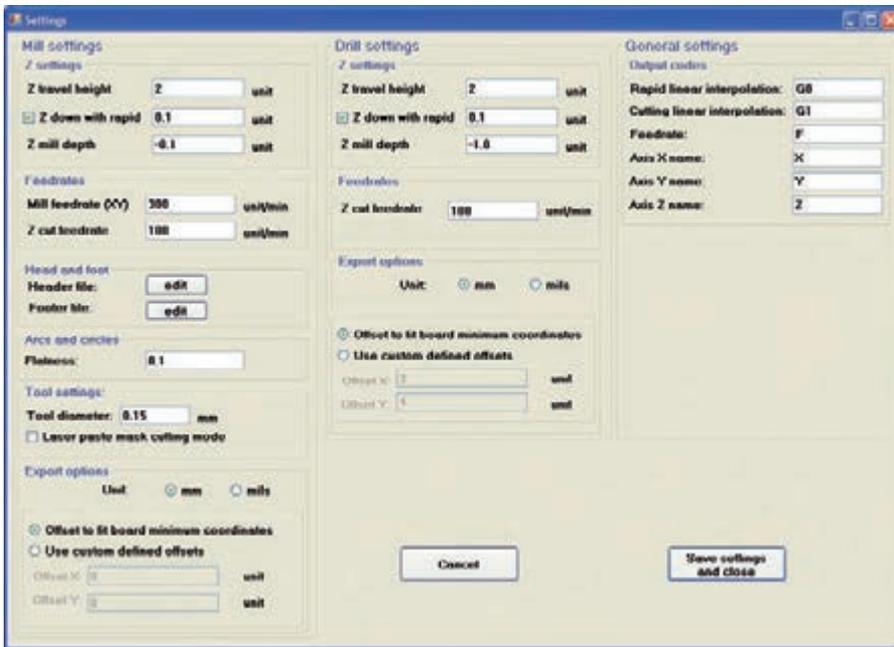


Fig. 1

sección se describe cómo obtener este archivo G-Code partiendo de un archivo Gerber (este es el formato utilizado por el software de diseño PCB); para la conversión se utiliza el software gratuito PCBMill, proporcionado por CNCdrive (se encuentra en la sección Softwaretools freeware, en la página <http://cncdrive.com/downloads.html>), que le permitirá

obtener un fichero en formato G-Code que se pueden importar en Repetier Host. La realización del circuito impreso requiere el empleo de una fresa para llevar a cabo el grabado de las pistas y de una broca de diámetro adecuado para la perforación, primero montaremos una y después la otra. Una vez descargado el archivo, instálalo haciendo doble

clic sobre Setup y luego ejecútalo mediante un doble clic sobre su icono; primero es necesario efectuar los ajustes para obtener el archivo de G-Code correcto. Haz clic sobre PCBMill-> Settings y abra la

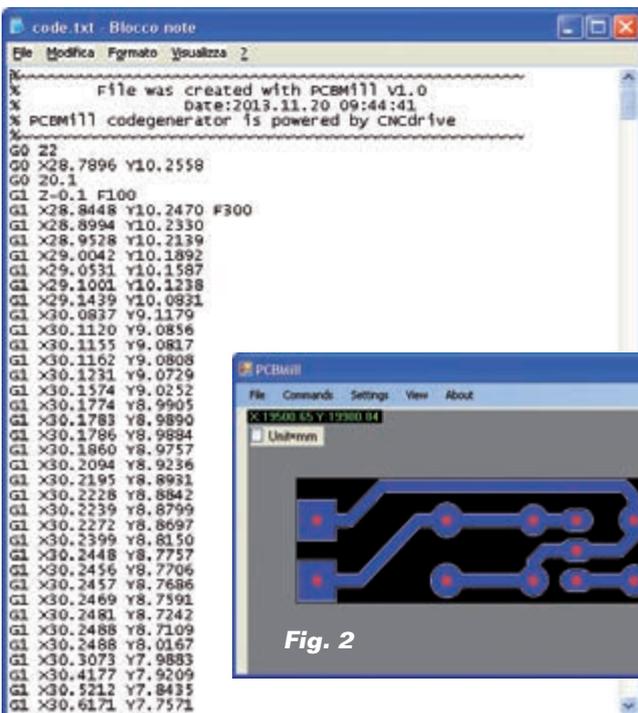


Fig. 2

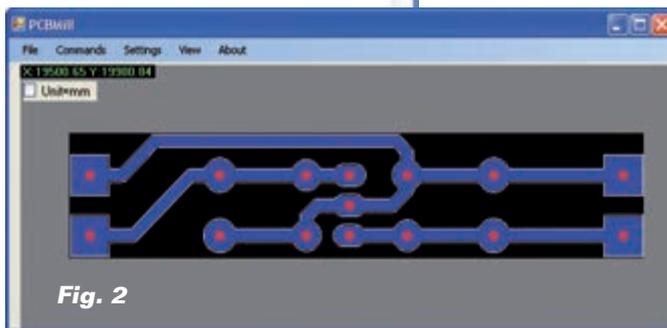


Fig. 3

ventana de la Fig. 1.

El menú *Settings* se divide en tres partes: la primera (*mill settings*) le permite configurar algunos parámetros necesarios para la creación de la incisión G-Code para las pistas, como por ejemplo la velocidad de avance, la profundidad de grabado, etc. Aconsejamos que se mantengan los parámetros que indicamos en la Fig.1. Diámetro de herramienta depende de la fresa (cincel) que sea utilizado (en nuestro caso 0,15mm). La segunda parte, es decir, la configuración de taladro, se refiere a la configuración de la perforación del PCB; la tercera se relaciona con la generación del código G (aquí hay que dejar los parámetros por defecto).

Creación del archivo G-Code

Para que el procesado sea correcto, se debe generar un archivo de Gerber en relación con el “espejo” del circuito impreso, es decir, un negativo, de modo que, el ancho del cobre (normalmente definido como Bottom Copper), permanece hacia arriba. En el menú de PCBMill da la orden de Gerber y abre el archivo relativo a las pistas (por ejemplo Bottom Copper.gbr); de forma automática, el software lleva a cabo el G-Code en el contorno de las pistas (Fig.2). El código se puede ver mediante la selección de la opción Export-> Milling G-Code (Fig.3). En este punto es necesario repetir Host y preparar el G-Code: abrir Repetier Host, seleccionar “CNC Router” (de “Configuración de la impresora”, indicado por la flecha roja en Fig.4) y ajuste los parámetros como ha visto en la Fig.5. Para visualizar correctamente tanto el grabado como la trayectoria de la herramienta, selecciona *Show filament* y *Show Travel*. Ahora importa en el Editor G-Code

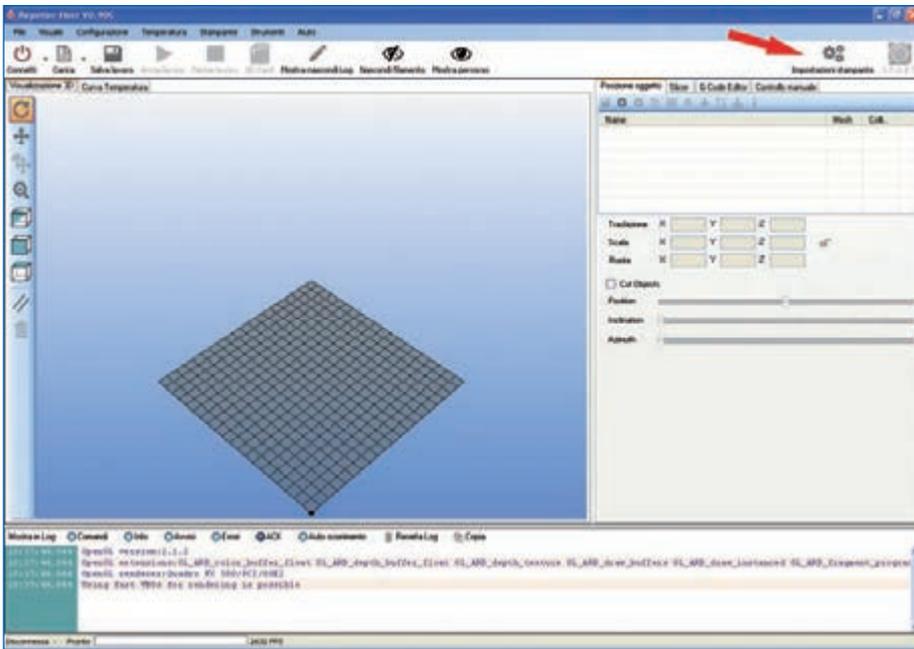


Fig. 4

Host de Repe (simplemente con un copia pega) el fresado de las pistas, generado previamente con PCBMill, que se muestra en la Fig.3. La única precaución es eliminar las primeras 5 líneas de G-Code, o aquellas que empiezan con %:

```
%-----
% File was created with PCBMill V1.0
% Fecha:2013.11.19 04:36:26
% PCBMill codegenerator is powered by CNCdrive
%-----
```

Repetier-Host es capaz de interpretar sin ningún problema el G-Code creado por PCBMill (se han eliminado manualmente las primeras 5 líneas), sin embargo, antes de comenzar con el grabado, es bueno para definir el inicio y el final del G-Code imponer a la máquina la posición de partida y fin del trabajo. Esta operación es muy simple de realizar en Repetier, ya que se prevé un Start Code (código inicial) que Repetier lanzará antes de ejecutar el G-Code principal (tratamiento de la palca) y un End Code (código final) que se ejecutará una vez acabado el trabajo. A la derecha, selecciona la tecla Editor G-Code, abre el menú desplegable y selecciona Start Code (Fig.6).

Introduce las siguientes líneas de comandos:

```
G92 X0 Y0 Z0 ; offset del sistema de
coordenadas (impuesto el punto de
partida como home)
G01 F500; movimiento lineal
```

(seleccione una velocidad de movimiento de 500 mm / min).

Lo mismo se debe hacer con End Code, insertando los comandos: *G01 X0 Y0 F1200; movimiento lineal (vuelve al punto X0 Y0 con una velocidad de 1.200mm/min).*

PREPARACIÓN DEL CNC Y GRABADO DEL PCB

Monta en la fresadora la fresa para el grabado, con una dimensión para la punta idéntica a la que se define en la configuración de PCBMill (recomendamos un cincel de metal duro, debido a que el vidrio es altamente abrasivo), dejando que sobresalga el mínimo indispensable para evitar vibraciones excesivas. Acerca tanto como sea posible la herramienta a la placa "sacrificable" hasta tocarlo; mueve el plano en todas las direcciones para comprobar que la distancia la herramienta

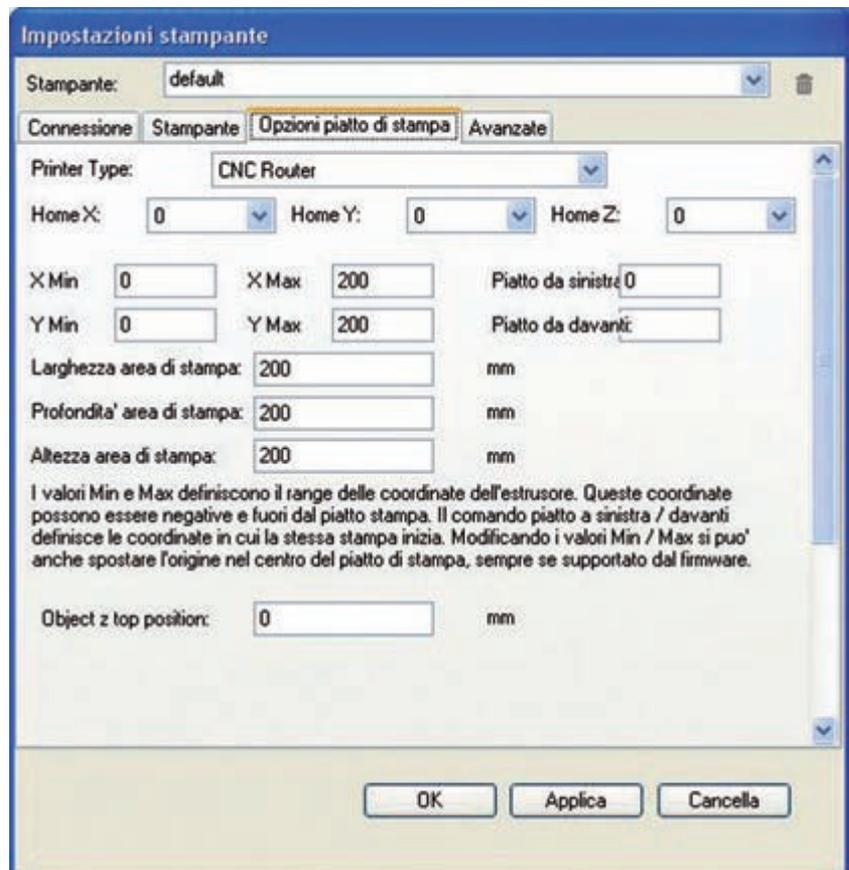


Fig. 5

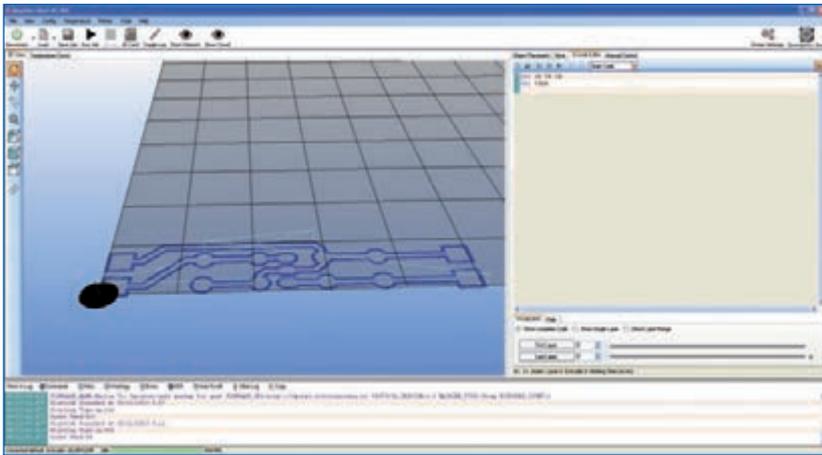


Fig. 6

sea constante. Haz un pre-ajuste de la placa de la impresora, girando los mandos, de modo que quede tanto como sea posible al ras con la herramienta (Fig 7). Ahora puedes fijar con la cinta adhesiva de doble cara la palca de cobre en el plano de trabajo. Después de fijarla, es necesario efectuar un ajuste girando los mandos adecuados, de modo que la base esté a nivel con la herramienta (para realizar la operación procede como anteriormente, acercando la punta de la herramienta a la base, a continuación mueve lentamente la placa en dirección x e y de modo que la distancia permanezca lo más constante posible sobre toda la superficie de la base. Como alternativa, puedes volver a utilizar un comparador con adaptador – como el impreso previamente – insertado en el cuellos previsto para la fresadora.

Empezamos por lo tanto el grabado de las pistas: baje el eje Z (utilizando el control manual de Repetier-Host), suficiente para acercar la fresa a la base (aproximadamente de 2 a 3 mm) y entonces ajusta los ejes X e Y para posicionar la herramienta en el origen del PCB que se quiere grabar. Ahora, enciende la fresadora, ajusta la velocidad máxima, y después gira manualmente (muy despacio) el eje del motor Z para bajar la fresadora hasta que la punta de la fresa toque el PCB. El reajuste correcto se obtiene cuando el utensilio empieza a “pulir” el cobre sin llegar a grabarlo. En la pantalla del Repetier-Host, haz clic sobre el botón de puesta en marcha: la máquina realizará el contorno del PCB como está programado. Terminada la incisión, la herramienta eléctrica volverá exactamente al punto de partida.

Ahora, mediante el comando *Export-> Drill G-code* (Fig.9) exporta el fichero. El archivo generado por PCBMill (Fig. 10) y abierto en el Bloc de notas, u otro editor de texto por defecto en el sistema operativo, se debe a importar al G-Code Editor del Repetier-Host, simplemente copiando y pegando. En este caso, el archivo importado debe ser modificado. En la ventana de visualización 3D aparecerán los recorridos de la herramienta de taladro (*Show filament* y *Show Travel* deben estar activados). Eleva unos diez centímetros el eje Z mediante las teclas del control manual (tenga cuidado de no mover X e Y) sustituye la fresa cincel por una broca de acero, de diámetro adecuado (por ejemplo 0,8mm). Mueve el eje Z para colocar la punta en aproximadamente 1 a 2 mm de la placa de cobre. Ahora enciende la fresadora y gira manualmente el brazo Z para bajar la máquina hasta que toque con la punta el PCB: El punto de contacto debe coincidir con el punto de contacto previamente definido con la herramienta para el grabado. Ahora, en Repetier-Host, haz clic sobre el *botón inicio* del trabajo; la máquina perforará según lo previsto para el PCB (Fig.11). Una vez acabado el trabajo se repositionará en las coordenadas X0 Y0.

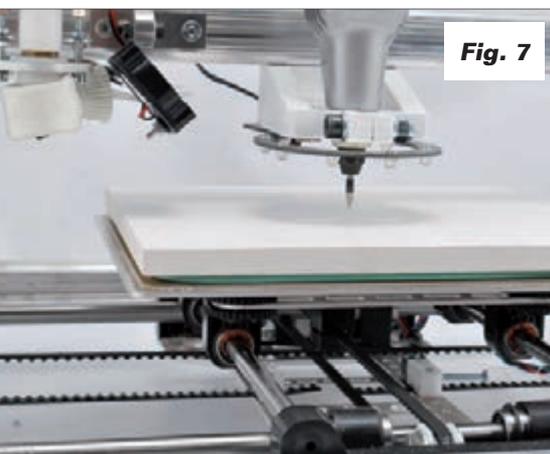


Fig. 7

Creación del archivo G-Code relativo a las incisiones

En el menú fichero PCBMill selecciona *Open drill*: se abrirá un menú en el cual deberá seleccionar la opción *Read drill file* (Fig.8). Carga el fichero de taladrado del PCB al que se han contorneado las pistas mediante fresado (por ejemplo *Drill fecha.drl*); el software creará automáticamente el G-Code correspondiente, mostrando como puntos rojos los agujeros que van a realizar.



Fig. 8

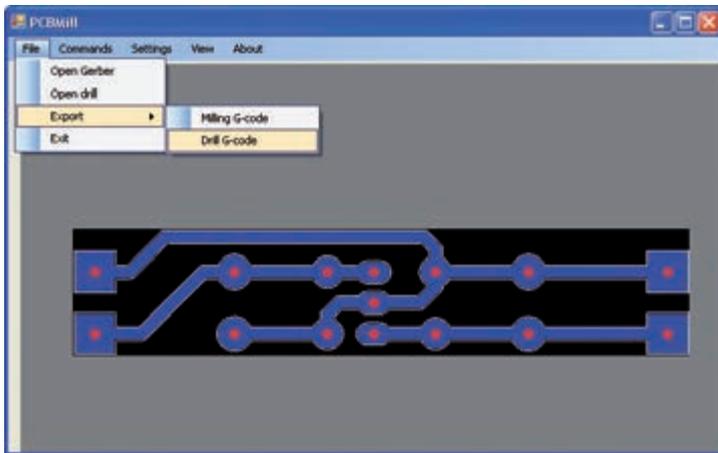


Fig. 9

IMPRIMIR DESDE EAGLE

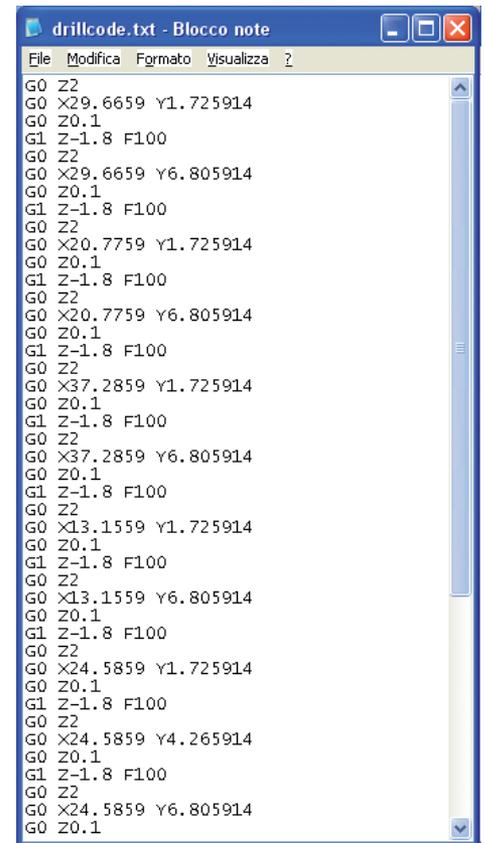
El G-Code también se puede obtener utilizando el plugin de Eagle adecuado, que el conocido programa CAD tiene disponible también en versión freeware. El plugin se llama PCB-gcode y permite definir la extensión del archivo de exportación (.nc) de modo que sea compatible con Repetier-Host. Si no tienes instalado EAGLE, ve a la página web www.cadsoftusa.com/download--eagle/?language=en y descarga la versión para tu sistema operativo; después ejecuta el archivo .exe e instala la versión freeware. A continuación, descarga PCB-gcode (nosotros utilizamos la versión 3.6.0.4) de la <http://groups.yahoo.com/neo/groups/pcb-gcode/?les!!%20Software> (es necesario tener una cuenta con Yahoo) o descargar de los foros dedicados (por ejemplo <http://pcb-gcode.org/read.php?12,803,803>). El manual completo de PCB-gcode (pcb-gcode.pdf) está disponible en la carpeta "docs" del programa pcb-gcode-3.6.0.4 descargado. Descomprime la carpeta y, a continuación copia los archivos a la carpeta ULP de Eagle. Después, haz clic sobre File/Open/Board y selecciona el archivo del PCB que se desea realizar mediante fresado (Fig.12).

Haz el "espejo" de la placa de circuito impreso y alinea la esquina inferior izquierda de la placa con el punto de origen de la hoja

de trabajo; esta operación mantiene el PCB dentro del plano de trabajo y lo orienta de modo que el lado de cobre a trabajar (denominado "Bottom Copper") quede hacia arriba. Ahora vamos a configurar los parámetros PCB-gcode: empieza escribiendo el PCB-gcode, en la línea de comandos, *run pcb-gcode-setup* (Fig. 13). Después de presionar enter aparecerá la ventana donde encontrarás varias pestañas de configuración (*Generation Options, Machine, GCode Style, GCode Options...*); en la pestaña *Generation Options* marca los dos elementos *Top Side* y el elemento *Single pass de Isolation*. Esta última permite hacer el contorno de las huellas con una única pasada de la herramienta.

Establece un valor para el aislamiento mínimo (Minimum, es decir, la mínima distancia entre

Fig. 10



la pista y el pad) y el tamaño de la punta de la herramienta de fresado utilizada (*Etching Tool Size*): por ejemplo 0,2mm. En la pestaña *Machine* es necesario definir, además de la unidad de medida (mm, Mils, etc...), los parámetros relacionados con el avance de la máquina, o sea el *Feed Rates* (300mm/min para X e



Fig. 11

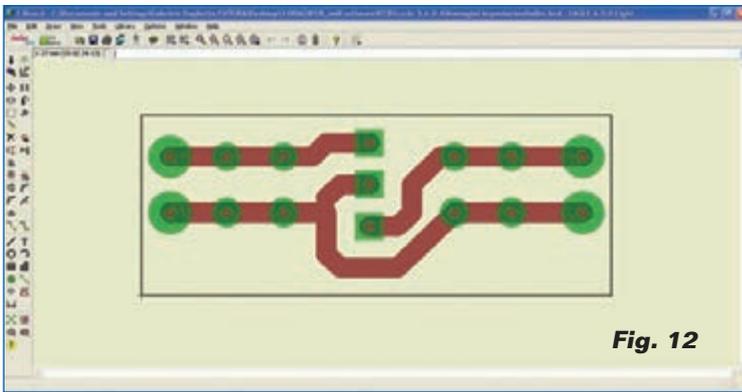


Fig. 12

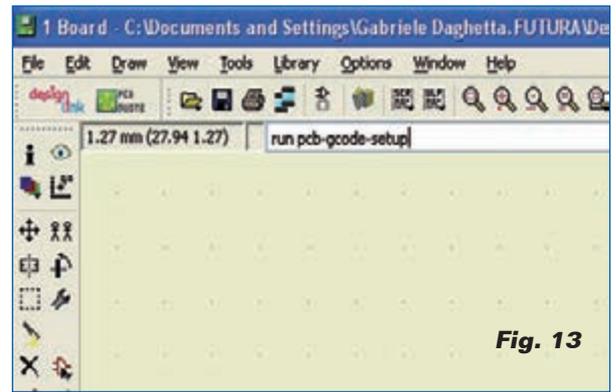


Fig. 13

Y y 100mm/min para Z), la colocación de la Z (Fig.14):

- Z High: elevación máxima de la Z (por ejemplo 5mm);
 - Z Up: levantamiento de Z durante el cambio de un punto al otro (por ejemplo 0,8 mm);
 - Z Down: profundidad de la incisión (por ejemplo 0,1mm);
- Drill Depth: profundidad de perforación (por ejemplo 1,8mm; depende del grosor de la base);
 - Drill Dwell: tiempo de espera antes de la aparición de la broca (por ejemplo 0,5 s).

En la pestaña GCode Style se esta-

blece el estilo del G-Code generado más adecuado para nuestra máquina (hemos escogido C:/Program Files (x86)/EAGLE-6.5.0/ulp/profiles/generic.pp). La pestaña GCode Options (Fig. 15) permite elegir entre insertar comentarios en el G-Code generado (en nuestro caso no tiene que estar marcado ningún elemento), si utiliza un G-Code personalizado y además de otro. En la realización de nuestra PCB, nosotros hemos medido sólo el elemento Use simple drill code (que permite obtener un G-Code compatible con la perforación Repetier-Host) y definir "nc" como Extension (siempre que sea compatible con los archivos generados por Repetier-Host). Las últimas dos pestañas (Plugins y Other) no han sido tomadas en cuenta. Después de ajustar los parámetros de PCB-gcode, haz clic sobre Accept para confirmar las opciones (la ventana de setup se cierra y reaparecerá en la pantalla del monitor Board editor con su PCB). Para crear el archivo G-Code útil para el grabado de las pistas y la perforación de la placa, teclea en campo adecuado run pcb-gcode y pulsa intro.

Fig. 14

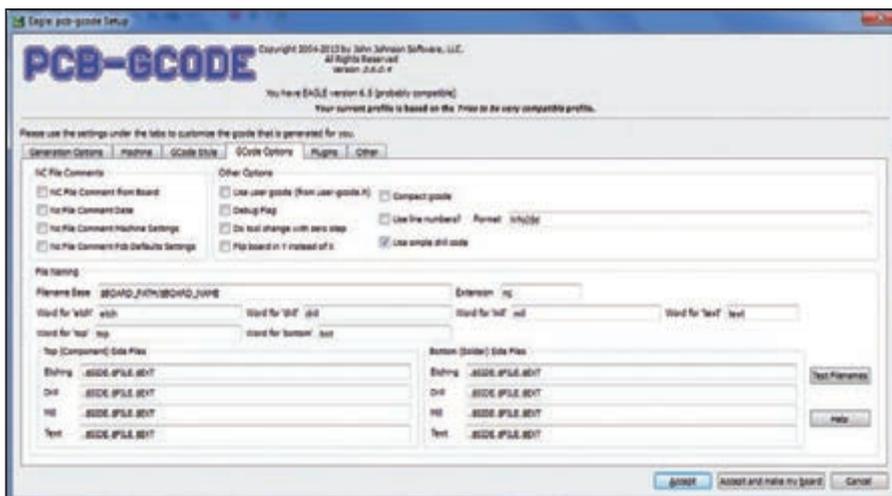
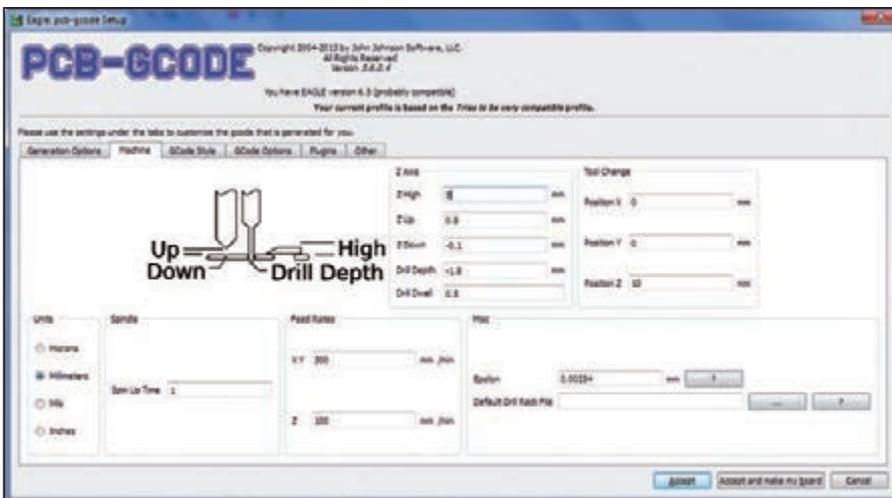


Fig. 15

AJUSTAR REPETIER-HOST E IMPORTAR/MODIFICAR EL G-CODE

En Repetier Host, selecciona CNC Router (se accede desde los ajustes de la impresora en la parte superior derecha, indicado por la flecha roja en la Fig.16) y ajusta los parámetros como aparece en

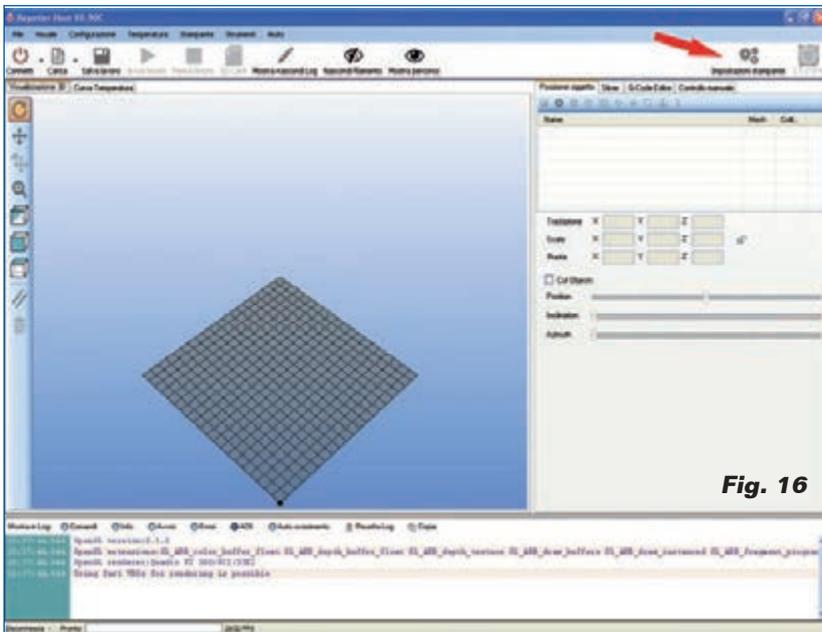


Fig. 16

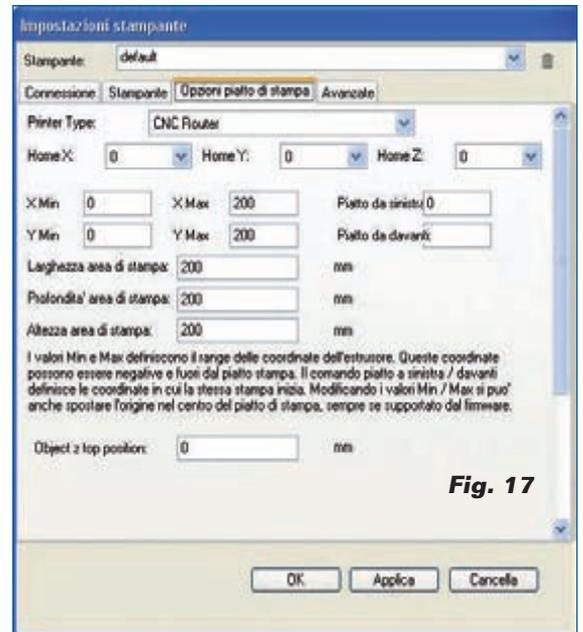


Fig. 17

la Fig. 17. Carga en Repetier-Host el Gcode *xxx.top.etch.nc*, o sea el archivo correspondiente al recorrido de la herramienta del PCB que se desea realizar (xxx representa el nombre asignado al PCB, que en este caso es "alim"). En la ventana de visualización 3D aparece el contorno de las pistas del PCB que se realizarán mediante la fresadora (*Show filament* y *Show Travel* deben estar activos). Desde el editor de G-Code, tendrás que eliminar las líneas iniciales en las que aparecen paréntesis (Fig.18): en este caso, la segunda línea de comandos. Ahora tendrás que hacer clic sobre el menú

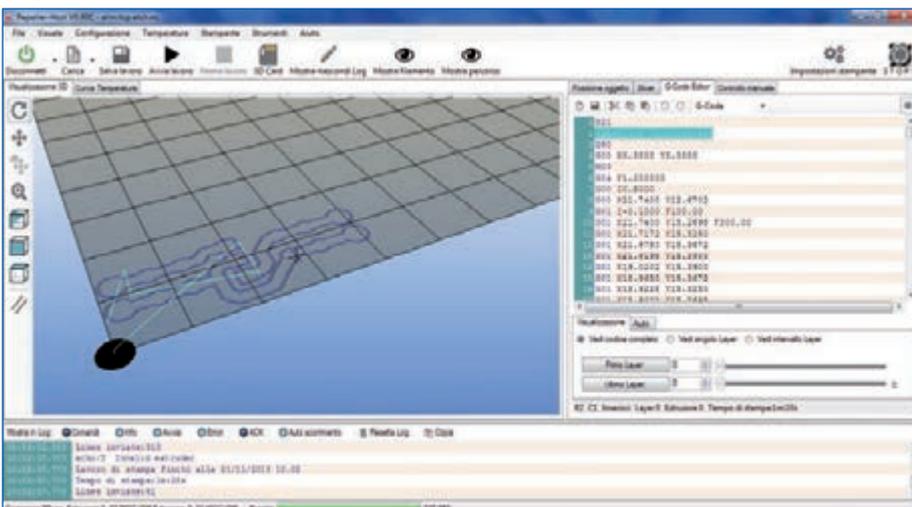
desplegable del G-Code editor y seleccionar el código original, a continuación introduce el GCode inicial:

- 1ª fila = G92 X0 Y0 Z0; desplazamiento del sistema de coordenadas, ajuste X0 Y0 Z0 como punto de partida;
- 2ª fila = G01 F500; movimiento lineal con una velocidad de 500mm/min.

Ahora haz clic sobre el icono del disco para guardarlo, luego selección *G-Code final*, introduce G01 X0 Y0 F500 (vuelve a la posición X0 Y0 con un movimiento lineal de 500 mm/ min) y guárdalo. Estos cambios se utilizan para

definir el desplazamiento del sistema de coordenadas de modo que la máquina funcione en el área deseada y vuelva al punto de partida una vez acabado el trabajo. En Repetier Host, ejecuta el trabajo como se describe en las páginas precedentes. Bien, con esto creemos haberle explicado todo; si deseas más detalles, los puedes hallar en el tutorial que hemos publicado en la página web <http://3dprint.elettronica.in.it/la-3drag-para-tuyo-circuiti-stampati/>.

Fig. 18



EL MATERIAL

La impresora 3Drag está disponible en el kit ya montado y comprobado. La versión actual (1.2) completa con placa calentada en el kit está disponible a un precio de **680,00 euros** (cod. 3DRAG/K), mientras que la versión editada cuesta **860,00 euros** (código 3DRAG/M).

Los precios incluyen IVA.
Los gastos de envío no van incluidos.
Puede hacer su pedido en:
pedidos@nuevaelectronica.com

PREHISTORIA II continuación

III.- El telégrafo de Cooke-Wheatstone.

Desde los ensayos de Salvá hasta la obtención de la primera patente de un telégrafo, en 1837, hubo varios sistemas de telegrafía eléctrica, pero no pasaron de ensayos de laboratorio. Se suelen citar el de Soemmerring, que utilizaba la electrólisis que había propuesto Salvá y necesitaba 35 conductores o el de Schilling, que, en 1832, fue el primero en utilizar el efecto del paso de una corriente eléctrica sobre una aguja imantada para detectar señales.

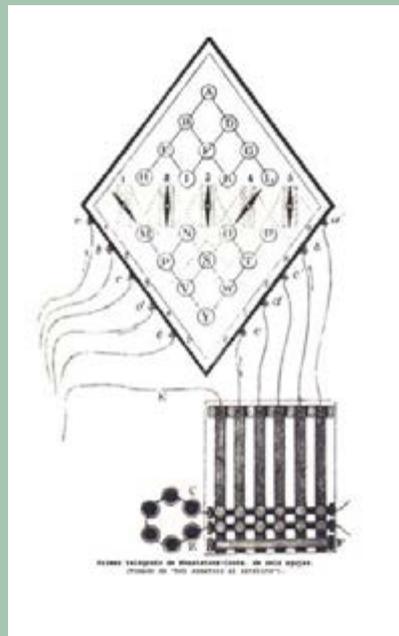


Figura 16.- Telégrafo de Soemmerring.- Tomada de "Del semáforo al satélite", publicación de la U.I.T. de 1965.

Lo más sobresaliente de este período es el descubrimiento de los efectos electromagnéticos, hecho por Oersted y la posibilidad de cerrar el circuito por tierra, aplicado por primera vez por Stenheil .

En 1837 William Fothergill Cooke, asociado con Charles Wheatstone, profesor de Física del Kings College de Londres, patentó un telégrafo que, utilizando cinco conductores y cinco agujas imantadas, podía transmitir cualquier mensaje, empleando veinte letras. El sistema estaba, todavía, muy cerca de los ensayos de gabinete, pero ya era operativo. La letra que se transmitía era la que señalaba la intersección de las líneas que generaban las dos agujas que se movían. No se pudo hablar de código para la transmisión de las señales sino de habilidad para hacer coincidir el movimiento de las agujas adecuadas.

Figura 17.- Aparato Wheatstone de cinco



agujas.- Tomada de "Del semáforo al satélite", publicación de la U.I.T. de 1965.

Este primer aparato fue ofrecido a la compañía de ferrocarriles Londres-Birmingham, pero no tuvo éxito. Por contra

otra compañía ferroviaria les encargó el establecimiento de un telégrafo entre la estación de Paddington, en Londres, y la de West Drayton, a 21 Kilómetros de distancia. La establecieron y el telégrafo empezó a funcionar el 9 de julio de 1839. Esta puede ser la fecha de la puesta en marcha, de manera formal, del telégrafo eléctrico.

Pero cambiaron de aparato. En realidad ofrecieron dos variantes de un mismo aparato: de una aguja y de dos agujas. El funcionamiento era idéntico. El receptor era un simple galvanómetro y su aguja registraba el paso de la corriente y su sentido, desplazándose a derecha o a izquierda. El transmisor era una manivela que podía enviar corriente positiva o negativa girando a derecha o a izquierda.

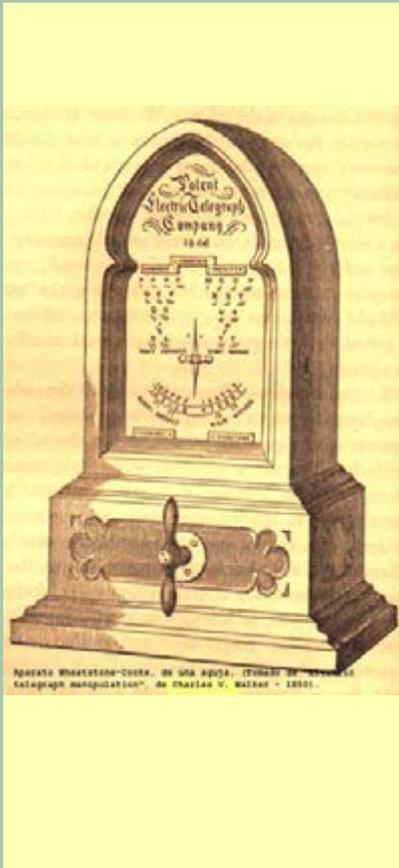
El aparato llevaba grabado el código para hacer más fácil la operación. Utilizaba un código abierto y necesitaba un máximo de cuatro impulsos para una letra.

Por ejemplo:

a: ++ ; b: --+ ; c: +-
d: -++ ; e: - etc.

que se traducían en aguja a la derecha para la corriente positiva y aguja a la izquierda para corriente negativa. Dos bandazos a la derecha para la a; dos a la izquierda, uno a la derecha y uno a la izquierda para la b, etc..

Figura 18.- Aparato Wheatstone, de una aguja.- Tomado de "Electric Telegraph



"Manipulation", de Charles V. Walker, de 1850.

El mismo aparato, con dos galvanómetros y dos manivelas, permitía aumentar algo la velocidad de transmisión, al emplear ambas agujas para una combinación. Pero obligaba a utilizar dos conductores y a que el operador transmitiera simultáneamente con las dos manos.

Este aparato de dos agujas fue el que se empleó en la primera línea del telégrafo eléctrico español.

El aparato de Cooke-Wheatstone sirvió para demostrar que el telégrafo era viable, no sólo para auxiliar al ferrocarril y facilitar las maniobras de los trenes, sino transmitiendo noticias. Los primeros folletos explicando el

telégrafo tiene algún apartado que dice la utilidad que puede obtenerse del envío de noticias con rapidez. Uno de los hechos que hizo llegar al gran público la utilidad del telégrafo fue la detención de un asesino que había huido montándose en el tren y el telégrafo permitió que los policías le esperaran al final del viaje.

Ya no había que usar diccionarios de claves para frases previamente convenidas sino que la transmisión era alfabética, puesto que la velocidad de transmisión era incomparablemente mayor que la de los telégrafos ópticos. Sin embargo este aparato no podía conservar memoria escrita de los signos que recibía y esto sería pronto un inconveniente.

Se puede decir, pues, que 1839 es la fecha de la puesta en funcionamiento del primer telégrafo. Los intentos anteriores se podrían considerar simples ensayos.

IV.- Otros telégrafos de la primera época.

El éxito de Cooke y Wheatstone, aunque al principio fue asociado al ferrocarril, animó a otros inventores, y en la mayor parte de los países occidentales aparecieron varios modelos de aparatos telegráficos.

En Francia, donde estaba funcionando una extensa red de telégrafos ópticos, con funcionarios entrenados, hubo una cierta resistencia a pasar a la telegrafía eléctrica. Se desconfiaba de la electricidad y los torreros temían por sus puestos de trabajo.

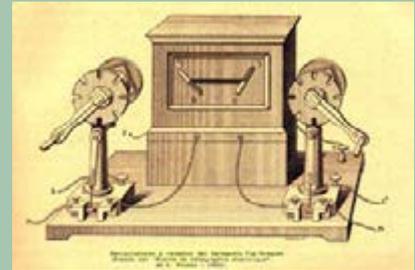


Figura 19.- Telégrafo de Foy-Breguet.- Tomado de "Tratado de telegrafía eléctrica", de H. Thomas, de 1903.

Seguramente por ello se produjo una especie de transición híbrida: se pasó a funcionar con un telégrafo eléctrico que reproducía las señales del telégrafo óptico. El telégrafo Foy-Breguet recreaba, eléctricamente, las señales del telégrafo de Chappe.

El receptor estaba formado por dos electroimanes que actuaban sobre un mecanismo que hacía avanzar por pasos a dos agujas, que correspondían a los indicadores del telégrafo óptico de Chappe. El regulador, que era la barra más larga del telégrafo óptico, permanecía inmóvil horizontal (pintada en el aparato).

Necesitaba dos manipuladores para la transmisión, uno para cada indicador, y, por lo tanto, dos conductores. Como ya no necesitaba funcionar mediante un código de frases hechas, sino que podían utilizar el lenguaje ordinario, su codificación era alfabética. Por ello los indicadores solamente tomaban seis posiciones. Giraban ángulos de 45, 90 y 135 grados por encima del regulador y 45, 90 y 135 grados por debajo. Disponían, con ello, de $6 \times 6 = 36$ signos. Los manipuladores estaban formados por discos sobre cuya circunferencia había seis hendiduras, enviaban tantos impulsos como fueran necesarios

para situar al indicador en la posición deseada.

Figura 20.- Código Foy-Breguet.- Tomado del

—	—	—	—	—	—	—	—
Ferme	Ouvart	Arrière	Drogaire	Facile	Remplum	Made	Euron
—	—	—	—	—	—	—	—
a	b	c	d	e	f	g	h
i	j	k	l	m	n	o	p
q	r	s	t	u	v	x	y
z	ch	c'	ou	ou'	ou''	ou'''	
ou	in	on	ou'	mon	ron	s	z
3	4	5	6	7	8	9	

"Nouveau Traité de Télégraphie électrique", de E.E.Blavier, sin fecha.

No había codificación de señales eléctricas, pero si de las figuras de los indicadores. Se enviaban los impulsos necesarios, con un máximo de seis para cada letra.

En el argot profesional, a los ángulos de 45, 90 y 135 grados les denominaban cinco, diez y quince. Y la voluntaria semejanza de este telégrafo eléctrico con su predecesor, el óptico de Chappe, se pone de relieve en la denominación de los signos. Se decía, por ejemplo, "quince cielo, diez tierra" para designar la combinación ----. Es decir, se indicaba cielo por arriba y tierra por abajo, sin duda términos heredados de los torreros ópticos.

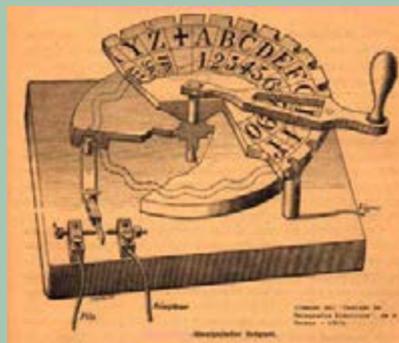
Este aparato fue utilizado por los franceses en sus primeras líneas eléctricas en 1845, y su utilización

sirvió para reciclar a los antiguos torreros. Su uso se abandonó al poco tiempo.

Otro aparato francés que utilizaba los mismos principios, pero sin sujetarse al recuerdo de Chappe, fue construido por Breguet. Curiosamente reproducía, eléctricamente, el telégrafo óptico de Betancourt y de su propio abuelo Breguet.

El funcionamiento era muy simple. El emisor era una manivela que podía girar recorriendo un disco con veintiséis muescas. A cada muesca le correspondía un signo. Al girar la manivela enviaba un impulso al pasar por cada muesca.

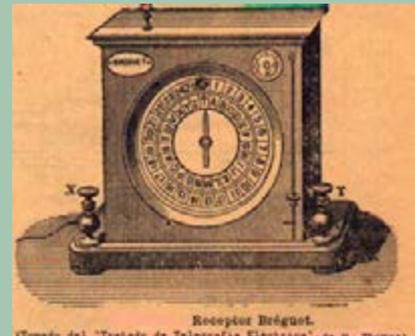
Figura 21.- Transmisor del aparato Breguet.-



Tomado del "Nouveau Traité de Télégraphie électrique", de E.E.Blavier, sin fecha.

El receptor era un disco idéntico, con las mismas 26 divisiones, cada una de las cuales tenía asignada una letra o signo. Un electroimán y un juego de relojería hacían avanzar a la aguja un paso por cada impulso recibido y, partiendo de una posición "cero" de partida, común al emisor y al receptor, éste seguía fielmente la posición que iba marcando la manivela del emisor.

Figura 22.- Receptor Breguet.- Tomado del



"Nouveau Traité de Télégraphie électrique", de E.E.Blavier, sin fecha.

Este aparato solamente necesitaba un conductor. Se utilizó en Francia para sustituir al Foy-Breguet y fue, a su vez, reemplazado por el morse. No obstante, por su extrema sencillez de manejo fue adoptado por muchas compañías de ferrocarriles y en ellas su uso perduró muchos años. En España estuvo en servicio, casi exclusivamente, en las estaciones de las líneas del ferrocarril, donde se mantuvo hasta la década de 1930.

Este tipo de aparatos se denominan de cuadrante y, por su elemental configuración, hubo varios modelos. Los más conocidos fueron el construido por Wheatstone y por Siemens-Halske, ambos utilizaban corrientes de doble polaridad.

V.- Morse.

Parece ser que Samuel Morse, que era un pintor, especializado en escenas históricas - alguna de las cuales adorna el Capitolio de Washington -, se interesó por los electroimanes en 1832 y estuvo trabajando en prototipos, asociado al mecánico Alfred Vail, durante varios años hasta conseguir que el Congreso de Estados Unidos le concediera un crédito para establecer una línea

entre Washington y Baltimore. La línea se inauguró en febrero de 1845.

Figura 23.- Circuito eléctrico de una comunicación "morse".- Tomado de



"Appareils et installations telegraphiques" de E. Montoriol, de 1921.

El circuito eléctrico es muy simple. El emisor, o manipulador, es una palanca que puede conmutar la línea al receptor (posición de reposo) ó a la pila, de acuerdo con el código, para emitir. En la figura se ha representado, muy esquemáticamente, el manipulador de la estación A unido directamente al electroimán E de la estación B, pero, en una instalación real, la línea pasaría antes por el manipulador de esta estación.

El principio de funcionamiento era sencillos. Se enviaban a la línea señales de diferente duración. Se distinguía entre dos señales: unas cortas y otras largas. Una señal de corta duración, que hoy podríamos denominar impulso elemental, excitaba al electroimán y, a través de una palanca adecuada, imprimía un punto sobre una cinta. Una señal de larga duración, equivalente a tres impulsos elementales, excitaba al electroimán e imprimía una raya. La combinación de puntos y rayas constituye el código de Morse.

Figura 24.- Código "morse".- Tomado de

Alfabeto.	
A ---	O ---
A ----	O ----
B ---	P ---
C ---	Q ---
D ---	R ---
E -	S ---
E ----	T ---
F ---	U ---
G ---	U ----
H ---	V ---
I -	W ---
J ---	X ---
K ---	Y ---
L ---	Z ---
M ---	CH ---
N ---	Telegrafo ---

Números.	
1 ---	6 ---
2 ---	7 ---
3 ---	8 ---
4 ---	9 ---
5 ---	0 ---

Puntuacion.	
Punto	-----
Dos puntos	-----
Punto de admiracion	-----
Punto de interrogacion	-----
Coma	-----
Punto y coma	-----

Tomado del "Tratado de la Redi Academia de Ciencias sobre telegrafia electrica", 1861.

"Tratado de telegrafia eléctrica", de H. Thomas, de 1903.

Probablemente el código es la aportación más importante que hubo en el campo telegráfico durante muchos años. Hoy día sigue siendo de utilidad. Se ha adaptado sin dificultad a diferentes etapas tecnológicas y ha sido usado con éxito por distintos medios.

El código morse se elaboró asignando las señales más cortas a las letras más usadas en inglés. El punto corresponde a la letra "e", que, al parecer, es la letra más común en dicho idioma. Al adoptarse internacionalmente hubo algunas modificaciones, sobre todo en lo referente a señales para avisos de servicio (por ejemplo hay una combinación que se usa para invitar al corresponsal a que empiece a emitir, otra para

indicarle que espere, etc.).

El manipulador morse es un simple conmutador de dos posiciones. Conmuta la línea, que está conectada a su parte central, con la pila (posición de trabajo) o con el receptor (posición de reposo).

Figura 25.- Manipulador "morse".- Tomado



de "Tratado de telegrafia eléctrica", de H. Thomas, de 1903

El manipulador original de Morse era muy diferente del que se utilizó después. Era un sistema automático para dar puntos y rayas, seguramente porque desconfiaba de la habilidad de la mano para dar la duración necesaria para distinguir claramente los puntos de las rayas. Muy pronto se adoptó el de la figura que se usó con muy escasas variantes constructivas.

El receptor es un electroimán de dos bobinas, que mueve una armadura, móvil entre dos topes. Cuando circula corriente, se atrae la armadura al tope de trabajo y, cuando deja de pasar, un resorte antagonista la lleva otra vez a la posición de reposo. Solidaria de la armadura va una palanca, llamada pluma, que conseguirá la impresión de la cinta.



Figura 26.- Receptor "morse".- Tomado de "La telegrafía actual en Francia y en el extranjero", de L. Montillot, de 1891.

En el aparato original de Morse la impresión se hacía con un punzón, sin emplear tinta. Se decía que era un receptor "de punta seca". Pero muy pronto se construyeron aparatos más fáciles de manejar, aunque respondiendo rigurosamente a su esquema de funcionamiento.

En realidad la parte más voluminosa y complicada del receptor morse es el aparato de relojería necesario para que la cinta, sobre la que se imprimen los signos, se mueva de manera uniforme.

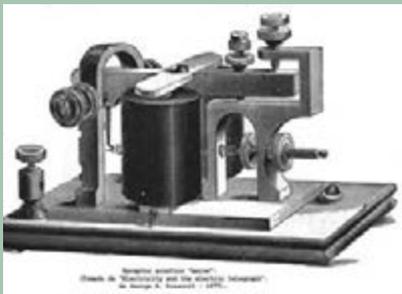


Figura 27.- Acústico "morse".- Tomado de "Tratado de telegrafía eléctrica", de H. Thomas, de 1903

Si no se desea tener constancia impresa de los signos que se reciben, se puede prescindir del aparato de relojería y utilizar como receptor el electroimán. En estos casos el receptor se denominaba acústico y se construía de modo que el movimiento de la armadura del electroimán pudiera oírse claramente, bien porque se le dotaba de una caja de resonancia, bien porque se construyera con material adecuado para ello.

VI.- Constitución de la red telegráfica mundial.

Las líneas de telegrafía óptica fueron todas nacionales y, aunque pasaban noticias e informaciones de un país a otro, no llegaron a constituir una red internacional. Cada país tenía su propio sistema que, además, consideraba como un secreto a guardar.

Al principio de la telegrafía eléctrica pasaba lo mismo y si, en algún caso, por ejemplo cuando servía a un ferrocarril, la línea tenía que atravesar una frontera, cada nación tenía su puesto terminal donde se intercambiaban el mensaje traducido al lenguaje corriente. Sin embargo pronto se dieron cuenta de que este tipo de intercambio retrasaba los mensajes y enseguida se establecieron acuerdos.

Se puede decir, por ello, que en la telegrafía se establecieron acuerdos internacionales existen desde su nacimiento. Pero, además, los acuerdos supusieron espíritu de colaboración. Desde el primer momento su constató que para poder comunicarse dos estaciones, ambos colaterales tenía que utilizar el mismo

aparato y el mismo código, tener las mismas reglas para establecer la llamada, etc.. No era posible mantener, cada uno, su sistema nacional, por ese, después de algunas pruebas, decidieron adoptar el sistema morse para las relaciones internacionales. Los acuerdos llegaron rápidamente y las comunicaciones telegráficas constituyeron, en pocos años, una verdadera red internacional.

En los primeros tiempos hubo que desarrollar todos los elementos para hacer posible enlazar primero las ciudades de un país, luego los países y, casi inmediatamente, los continentes.

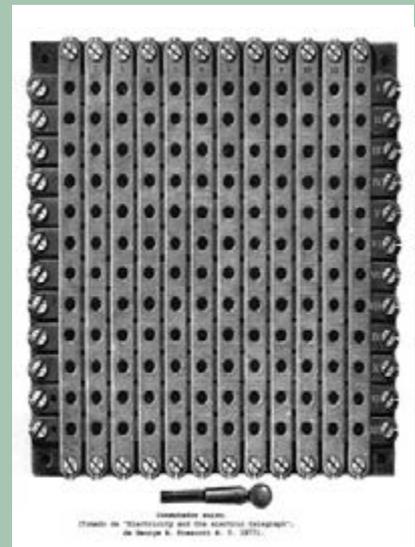


Figura 28.- Conmutador suizo.- Tomado de "Tratado de telegrafía eléctrica", de H. Thomas, de 1903

En el interior de cada nación, al crear la red, se tuvo que pasar de una comunicación punto a punto - estación A se comunica con estación B - a una comunicación todos contra todos. Al principio esa fue la pretensión: Madrid tenía un telegrama para Zaragoza, le llamaba y se lo daba; Guadalajara tenía un telegrama para Bilbao,

le llamaba y se lo daba, etc.. Esta pretensión se pudo mantener mientras sólo hubo una línea y pocos telegramas, pero se hizo inmediatamente inviable. Tuvieron que organizarse Centros colectores, conectados entre sí, que se encargaran de concentrar los mensajes.

Las estaciones tenían, en general, varios hilos que les unían a la red y disponían de varios aparatos. Para hacer posible una interconexión flexible entre ellos se diseñaron conmutadores más o menos complicados. Entre los más simples los había que utilizaban una manija que conmutaba entre dos hilos, otros utilizaban clavijas que podían conectar un hilo a diferentes aparatos. Entre estos últimos, el llamado "conmutador suizo", que era un conmutador de barras cruzadas, fue el más utilizado.

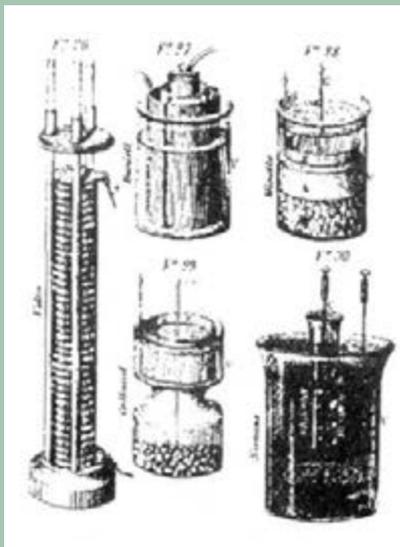


Figura 29.- Algunos tipos de pilas.- Tomado de "Tratado elemental de telegrafía práctica", de F. Pérez Blanca, de 1881.

Pero hubo que diseñar, también, los elementos auxiliares que eran imprescindibles para establecer las comunicaciones. Además del aparato encargado de enviar las

señales había que disponer de generadores. Se utilizaron pilas. La pila de Volta, en estado puro, se polarizaba rápidamente y no servía para dar una tensión constante.

Hubo muchos modelos que resolvieron el problema. Cada país escogió el suyo. Hubo algunos modelos que se extendieron más, pero se puede decir que coexistieron varios de ellos. Las más corrientes utilizaban dos electrodos de cobre y zinc y una disolución de sulfato de cobre. La f.e.m. de estas pilas variaba entre 1 y 1,25 voltios por vaso. Con muy pocas variantes se han utilizado durante más de cien años. Otros complementos necesarios que hubo que diseñar fueron las protecciones contra la electricidad de la atmósfera que, conducida por los hilos telegráficos, se cebaba en las primeras estaciones. Se pusieron como pararrayos los denominados "descargadores de puntas" que se ponían en paralelo con los hilos a la entrada de las estaciones.

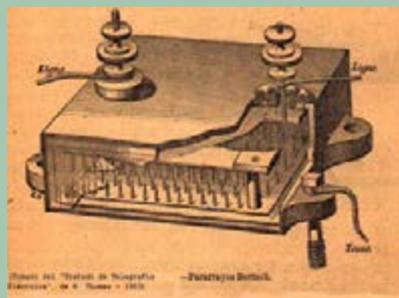


Figura 30.- Descargador de puntas.- Tomado de "Tratado de telegrafía eléctrica", de H. Thomas, de 1903.

Además de estos dispositivos complementarios, necesarios en todas las estaciones, también hubo que diseñar timbres para avisar, relevadores de corriente

ó relés para hacer posible comunicaciones a larga distancia, aisladores para soportar a los hilos, métodos para empalmar los conductores, etc..

La primera red nacional la tuvo Inglaterra, que en 1852 ya tenía 6.500 Kms. de líneas. En Estados Unidos no fue muy rápido el desarrollo de la red, pero recibió un gran impulso al fundarse la Western Union Telegraph Company, que en 1866 tenía ya 2.250 oficinas y 100.000 Kms. de líneas. En Francia la primera línea eléctrica se puso en funcionamiento en 1845, pero no se abrió al público en general hasta 1850. En este mismo año se tendió el primer cable submarino que enlazó Francia con Inglaterra. También a finales de 1850 se enlazaron París y Berlín.

En 1860 se consiguió un enlace entre Londres y Karachi. En 1865, cuando se fundó la Unión Telegráfica Internacional, la longitud de las líneas de los estados miembros - todos europeos - era ya de 500.000 Kms. .

Sin embargo faltaba un eslabón importante. Hasta 1866 no se consiguió tender con éxito un cable entre Europa y América, con lo que se conseguía una red mundial o, por lo menos, que enlazaba puntos de todas las partes del mundo.

VII.- Desarrollo en España.

En España los primeros ensayos del telégrafo eléctrico se hicieron en el puerto de Bilbao. En 1849 se enviaban avisos de servicio entre Bilbao y Portugalete, por una línea de 13 Kilómetros. Pero el primer telégrafo para el servicio convencional de telegramas al

público se autorizó al ferrocarril Madrid-Aranjuez, en 1852. Sin embargo era un telégrafo con pocas perspectivas de ampliación, puesto que no había líneas férreas donde apoyarse y no prosperó.

En 1852 el gobierno decidió que, bajo la dirección del Jefe de los telégrafos ópticos, el brigadier Mathé, los propios toreros construyeran una línea de telegrafía eléctrica entre Madrid e Irún, pasando por Guadalajara, Zaragoza, Pamplona y San Sebastián. La línea tenía 605 Kms., se terminó a finales de 1854 y en abril de 1855 se abrió al servicio público.

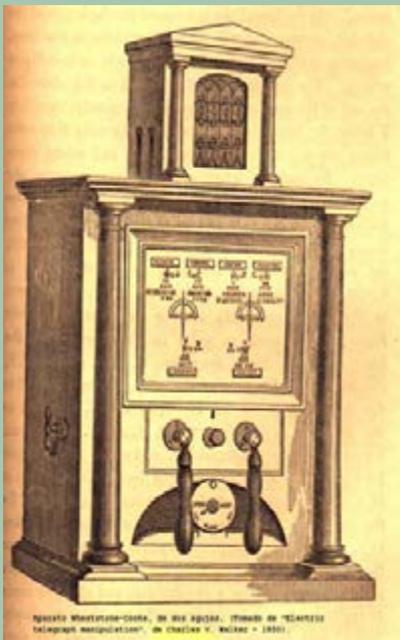


Figura 31.- Aparato de Wheatstone, de dos agujas.- Tomado de "Electric Telegraph Manipulation", de Charles V. Walker, de 1850.

En Irún la línea enlazaba con las líneas francesas, pero el intercambio de mensajes tenía que hacerse a mano en la frontera, porque la línea española utilizaba aparatos Wheatstone de dos agujas y las líneas francesas

empleaban aparatos Foy-Breguet, y ambos sistemas eran totalmente incompatibles.

En 1855, por Ley de 22 de Abril, el gobierno decidió la construcción de una red telegráfica que enlazara todas las capitales de provincia peninsulares con Madrid. Se pretendía que la red tuviera 6.280 Kms.. La decisión no debió ser fácil porque la idea dominante era que el telégrafo era un subproducto del ferrocarril y en España, en esas fechas, había muy pocos Kilómetros de vías férreas y, además, sin continuidad entre ellos.

La red se terminó a finales de 1857 y tenía 6.497 Kms., con un desarrollo de 17.214 Kms. y enlazaba 118 estaciones.

Cuando se empezó a construir la red ya se había llegado a un acuerdo entre los países europeos para adoptar el morse como aparato para las comunicaciones internacionales, de modo que, aunque en el pliego de condiciones inicial se decía que los aparatos serían de Wheatstone, se rectificó y se cambió en el curso de la tramitación.

En España, como en los demás países, iba improvisándose a la hora de organizar el servicio. El brigadier Mathé había organizado las líneas del telégrafo óptico y pretendía mantener una organización semejante para el telégrafo eléctrico. Pero la amplitud del nuevo servicio desbordaba, con mucho, la organización

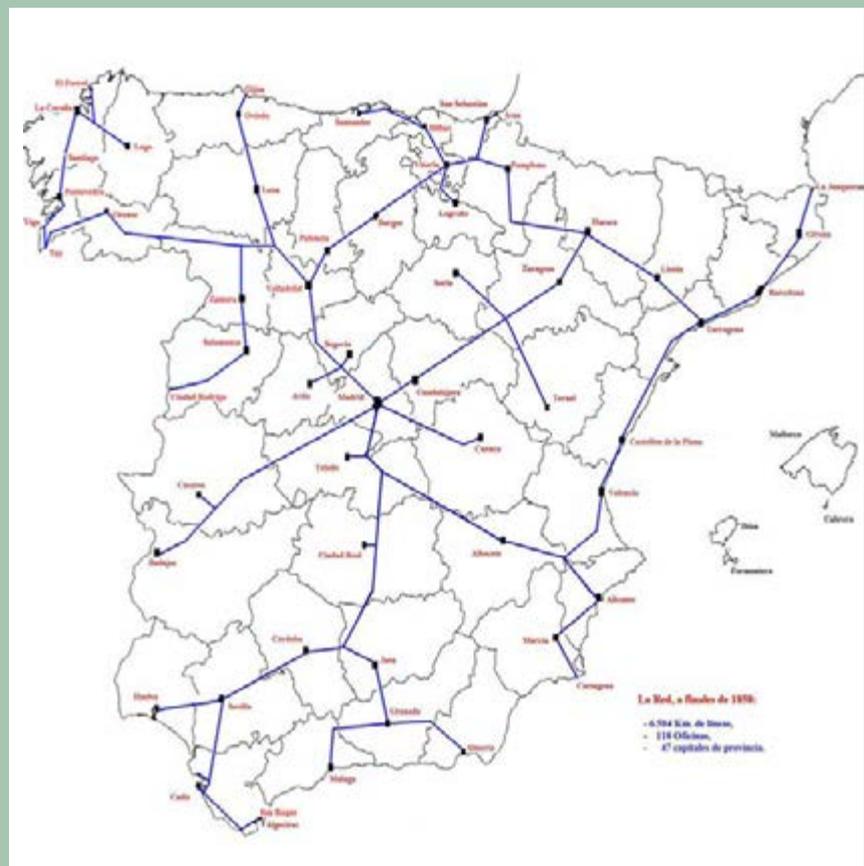


Figura 32.- La red española en 1858.- Dibujo propio.

paramilitar del primero. Ello ocasionó algún barullo administrativo que, quizá, ocasionó el retraso en la aparición de la profesión de Ingeniero de Telecomunicación (o Ingeniero de Telégrafos, como se hubiera llamado entonces).

La explotación del telégrafo se asignó al Ministerio de la Gobernación, seguramente porque se consideraba al telégrafo un instrumento del orden público. Pero la construcción de las líneas, que se establecieron al amparo de la Ley de 1855, se encargó al Ministerio de Fomento y fue dirigida por Ingenieros de Caminos, a pesar de que el Pliego de condiciones de Mathé y que en dicho Pliego se decía que el modelo a seguir sería la línea de Irún, construida por ellos.

Sin embargo al poco tiempo, a finales de 1857, se reintegró al Ministerio de la Gobernación, es decir, a los telegrafistas, la facultad de construir líneas telegráficas.

La red inicial era radial, desde Madrid a la periferia. Pero enseguida se empezaron a construir líneas para cerrar polígonos. La primera línea de este tipo enlazó Badajoz con Sevilla, se continuó con otra línea Santander-Ferrol, y así sucesivamente hasta disponer, diez años después, en 1868, de más de 11.000 Kilómetros de líneas, de los cuales casi 300 eran los cables submarinos que unían la península con Baleares. La red tenía en servicio 184 estaciones.

En la construcción de alguna de estas líneas se produjo la contratación del primer

Ingeniero de Telégrafos. La empresa alemana Siemens ganó el concurso para construir unas líneas, denominadas del bajo Aragón, de unos 450 Kilómetros de longitud, y contrató para dirigir la construcción al funcionario de Telégrafos, a título particular, como Ingeniero. Su nombre era Ignacio de Hácár.

Las instrucciones acaban con la nota: "Esta Instrucción se ha compilado, teniendo a la vista, entre otros datos, la circulada por la Administración francesa, sobre el mismo asunto", lo que deja claro la colaboración entre Administraciones.

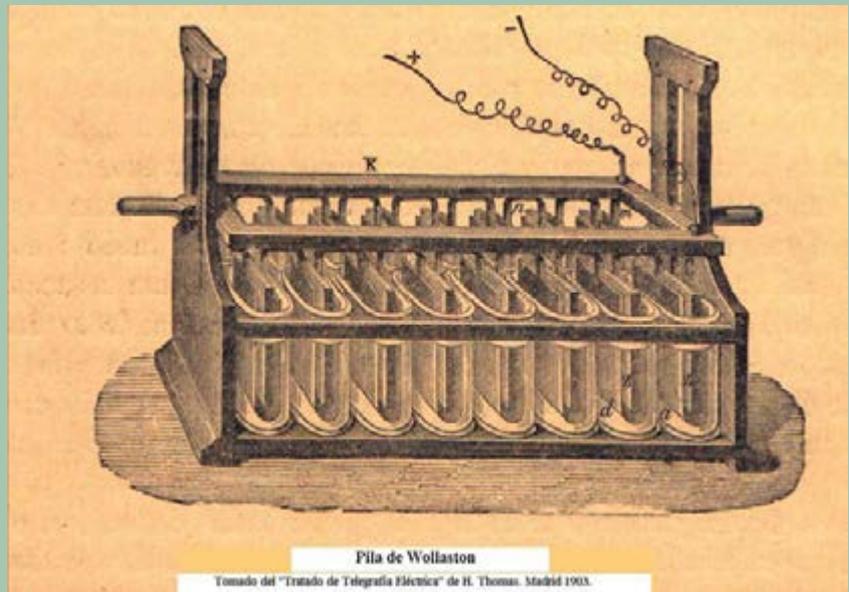


Figura 33.- Pilas Wollastron.- Tomado de "Tratado de telegrafía eléctrica", de 1903

Las pilas utilizadas en las estaciones de la primera línea de Irún eran del tipo Wollastron, que fue uno de los primeros modelos que trataban de evitar la polarización de los electrodos. Con las nuevas líneas se pasó al modelo de pilas de Daniel.

Existe un folleto de instrucciones sobre como utilizar la pila Daniel, firmado por el propio Director general de Telégrafos, Mathé, en enero de 1957. Es curioso ver que los únicos conceptos eléctricos que maneja son la intensidad y la resistencia. Todavía no aparece el concepto de fuerza electromotriz y las unidades de medida no existen.

Bibliografía para este Tema:

- "Electric Telegraph Manipulation".- Charles V. Walker.- Londres 1850.
- "Tratado de Telegrafía".- Antonino Suárez Saavedra.- Barcelona 1880.
- "Tratado elemental de telegrafía práctica".- F. Pérez Blanca.- Madrid 1881.
- "La telegrafía actual en Francia y en el extranjero".- L. Montillot.- Madrid 1891.
- "Tratado de telegrafía eléctrica".- H. Thomas.- Madrid 1903.
- "Del semáforo al satélite".- U.I.T..- Ginebra 1965.

TEMA Nº 3.- LOS CABLES SUBMARINOS.

- I.- Primeros cables.*
- II.- El entorno europeo.*
- III.- Rutas rentables: la India y América.*
- IV.- Los cables trasatlánticos.*
- V.- El proyecto de cable trasatlántico español.*
- VI.- España: los cables a Baleares, a Marruecos y a Canarias.*
- VII.- Aparatos para comunicar por cables submarinos.*

I.- Primeros cables.

La idea de atravesar el mar mediante cables debidamente aislados está ya expresada por Salvá en su Memoria de 1800. Cuando el telégrafo eléctrico era ya una realidad se hicieron múltiples ensayos, generalmente atravesando ríos y brazos de mar, incluso Wheatstone envió a la Cámara de los Comunes un proyecto de cable para cruzar el canal de la Mancha, pero el inconveniente elemental era la dificultad de mantener un buen aislamiento de los conductores. Se intentó utilizar el caucho y, entre otros, Samuel Morse hizo pruebas en el puerto de Nueva York, en 1842, utilizando un cable con una cubierta de algodón, asfalto y caucho, sin mucho éxito porque el cable se inutilizó rápidamente.

Hacia 1845 se empezó a ensayar el aislamiento con gutapercha - que es el látex de un árbol de Malasia. En Europa y en Estados Unidos se hicieron, casi simultáneamente, varios intentos: en Nueva York, en 1845 se sumergió en el Hudson un cable de dos millas que se mantuvo funcionando durante varios meses; en 1848, en Alemania, se

tendió otro cable en el puerto de Kiel, y en 1849, desde un barco situado cerca del puerto de Folkstone, en la costa inglesa del canal de la Mancha, se estableció una comunicación con un trayecto submarino de dos millas de longitud, enlazando, a través de líneas convencionales, con Londres.

El aislamiento con gutapercha dio buen resultado y en Inglaterra se constituyó una compañía para tender un cable hacia Francia.

Pero por otra parte había que vencer otros inconvenientes, además de los que presentaba el propio cable. En Inglaterra el telégrafo estaba en manos de compañías privadas, en Francia era monopolio estatal. En Inglaterra se utilizaba el aparato de Wheatstone, en Francia se empleaba el Foy-Breguet. Pero, confiados en las buenas perspectivas del negocio, fueron vencidas las dificultades administrativas y se decidió que se utilizaría, en ambos extremos del cable, un aparato patentado por los hermanos Brett, que, al mismo tiempo, eran los empresarios que pretendían tender el cable. El aparato en cuestión imprimía las letras directamente.

La compañía de los hermanos Brett hizo fabricar un cable de 25 millas, con dos conductores de cobre, de dos milímetros de diámetro, aislados por gutapercha. El cable fue tendido por el remolcador Goliath el 28 de agosto de 1850, pero no llegó a entrar en funcionamiento.

Para sumergir el cable se habían colgando pesos de 8 a 10 Kilogramos cada cien metros

para que se hundiera bien. Se partió de la costa inglesa y se llegó a la francesa manteniendo la comunicación eléctrica, pero, sin que advirtieran la causa, se cortó de repente. Parece ser que un pescador lo había pescado en sus redes, cortó un trozo y creyó que se había encontrado una desconocida especie de alga marina que tenía la parte central de oro.

El fracaso no desanimó a la Compañía y al año siguiente lo intentó de nuevo. El nuevo cable estaba formado por cuatro conductores de un diámetro de 1,65 milímetros, recubiertos por dos capas de gutapercha. Los cuatro conductores formaban un haz sobre el que se enrollaba un hilo de cáñamo embreado. Para rellenar los espacios vacíos se embutían otros hilos de cáñamo. Todo ello se recubría por diez hilos de hierro galvanizado de 7 milímetros de diámetro, que formaban una armadura protectora.

Esta armadura protectora se puso para evitar que el roce con las rocas cortara el cable. La profundidad en la zona del tendido no sobrepasaba los 75 metros. Con la armadura, el cable pesaba 7 Toneladas por milla, y ya no era necesario agregarle pesos suplementarios para hundirlo.

Este segundo cable se tendió con éxito el 25 de septiembre de 1851, entre Dover y el cabo Gris-Nez, cerca de Calais, con una longitud de 46 Kilómetros, aproximadamente, y se mantuvo funcionando perfectamente durante varios años. Este cable demostró que la conexión era posible y rentable económicamente, si tenía

suficiente tráfico - cosa que ya ocurría entre Inglaterra y Francia.

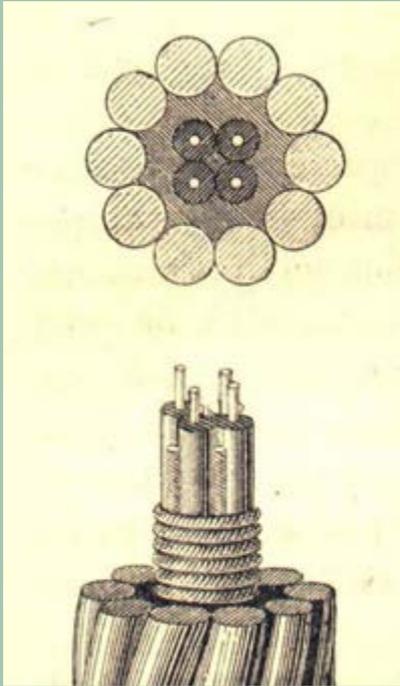


Figura 34.-Primer cable submarino operativo (1851).- Tomado de "Traité Général des Lignes et transmissions électriques", de Lazare Weiller, de 1892.

La comunicaciones entre Londres y París, sobre todo para las cotizaciones de la Bolsa y las noticias de los periódicos, se establecieron rápidamente y enseguida surgieron nuevas propuestas para tender cables.

II.- El entorno europeo.

El éxito del primer cable animó a las empresas a intentar el tendido de otros cables semejantes. Los primeros intentos se llevaron a cabo entre Inglaterra e Irlanda. En 1852 se tendió entre ambas costas un cable de 120 Kilómetros, de un solo conductor, recubierto de gutapercha y con una cubierta de 12 hilos de hierro galvanizado de 3 milímetros de diámetro. No tenía protección de cáñamo entre la gutapercha

y la cubierta de hilos de hierro. El cable funcionó solamente durante dos o tres días.

En 1853 se tendió un cable entre Ramsgate, en Inglaterra, y Ostende, en Bélgica, de 130 Kilómetros, con seis conductores recubiertos de gutapercha, cáñamo y una cubierta de doce hilos de hierro galvanizado de 7 milímetros de diámetro y funcionó perfectamente durante años.

La corta experiencia que se tenía indicaba que el peligro de rotura de los cables estaba cerca de las costas, sobre todo por la acción de las anclas de los barcos, y para evitarlo se ensayó el diversificar el arribo de los hilos. Se construyó un cable, entre Inglaterra y Holanda, que tenía cuatro conductores y, cuando llegaba a unos cinco Kilómetros de la costa, se separaban los cuatro hilos, constituyendo cuatro cables independientes hasta la costa con objeto de salvar la comunicación en caso de accidente de uno de ellos. El ensayo evitó alguna incomunicación pero no se adoptó como solución.

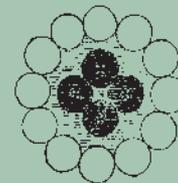
Entre 1853 y 1860, desde Inglaterra se tendieron varios cables hacia Francia, Holanda. También entre las islas danesas.

En el Mediterráneo se ensayó el tendido de los cables en la costa italiana, entre Córcega y Cerdeña, entre Cerdeña y Argelia, entre Malta y Sicilia, etc.. Los primeros cables que se tendieron entre la Spezia y Córcega y entre Córcega y Cerdeña eran idénticos, de 6 conductores, con cubierta de 12 hilos de hierro galvanizado de 8 milímetros de diámetro. La cubierta del cable ofrecía una

gran protección a los conductores, pero tenía un peso de unos 5.000 Kgs. por Kilómetro. Se pusieron, sin problemas, en 1854. Pero cuando se quiso colocar el mismo cable entre Cagliari y Bona, en Argelia, con una distancia de unos 200 Kilómetros y con profundidades de más de 3.000 metros, al ir tendiéndolo se fue al fondo sin que las máquinas del barco pudieran retenerlo a causa del excesivo peso.

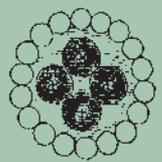
En 1857 se repitió la operación del tendido entre Cagliari y Bona, pero se cambió el cable, reduciendo su peso. Se utilizaron dos tipos de cable. Uno, para los primeros Kilómetros cercanos a las costas, compuesto por cuatro conductores (cada uno de ellos, a su vez, compuesto por cuatro hilos de cobre diferentes, pero enrollados para formar un solo conductor), recubiertos por dos capas de gutapercha y rodeados por una cuerda de cáñamo y una cubierta de 12 hilos de hierro galvanizado de 5 milímetros de diámetro.

CABLE DE COSTA.



OR 1 11

CABLE DE FONDO.



TOMADO DE "NOUVEAU TRAITÉ DE TÉLÉGRAPHIE ÉLECTRIQUE" DE E. BLAVIER.

Figura 35.- Cable de costa y cable de fondo.- Tomado del "Nouveau Traité de Télégraphie électrique", de E.E.Blavier, sin fecha.

Continuara...

DISPONIBLE EN PAPEL Y EBOOK

Disponible en:

www.jcmlanau.com

Amazon

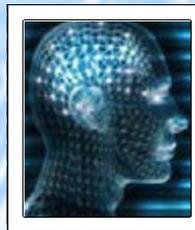
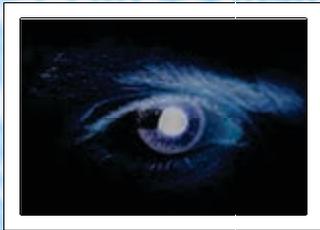
Y también en Apple, Casa del Libro y muchos más!

Tecnología de altas tensiones

J.C.M. Lanáu



"Tecnología de altas tensiones" de J.C.M. Lanáu. es un manual para la construcción de todo tipo de aparato que produce tensiones elevadas o simplemente eleva el voltaje. Cada capítulo describe con minuciosidad cómo **construir de maneras sencillas y caseras** las máquinas más conocidas que producen altas tensiones como pueden ser la bobina de Tesla, el generador de **Van Der Graaff**, el carrete de **Rummfortt**, el elevador de tensión de **Marx**, las máquinas multiplicadoras de voltaje y otros ingenios que apenas conocemos.



VISNOC TECHNOLOGY



INVESTIGACION E INNOVACION TECNOLOGICA



VISNOC TECHNOLOGY, S.L.

Polígono Industrial Las Salinas de Poniente c/ Alfred Nobel nº 22

11500 El Puerto de Santa María (Cádiz) - España

<http://www.visnoc.com> info@visnoc.com

Tlf. 956 144 424 - Fax. 956 548 241



Fondo Social Europeo
"El FSE invierte en tu futuro"

