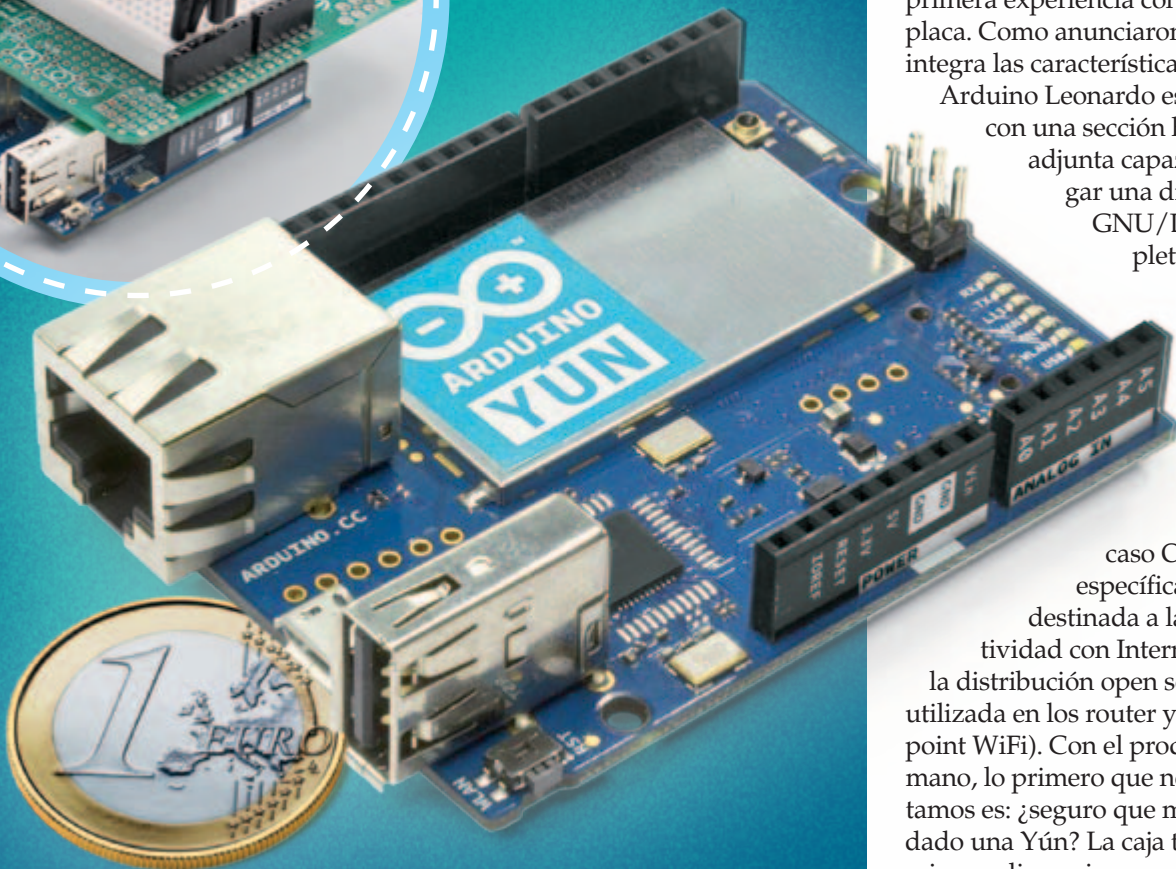
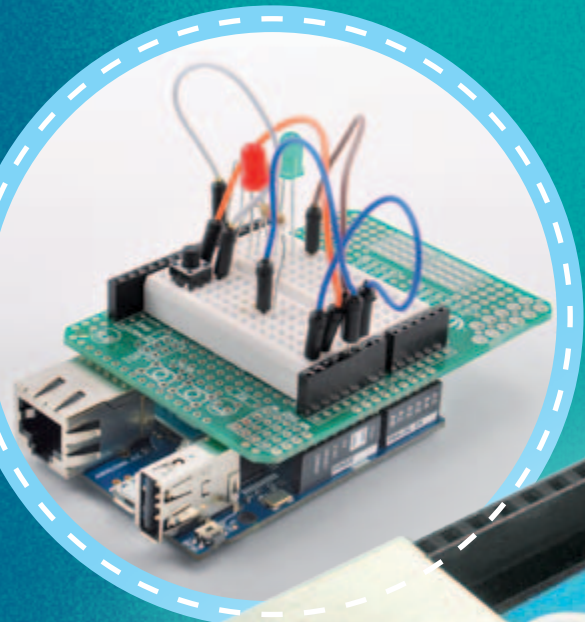


# ARDUINO YÚN: LA VIDA ES MÁS FACIL

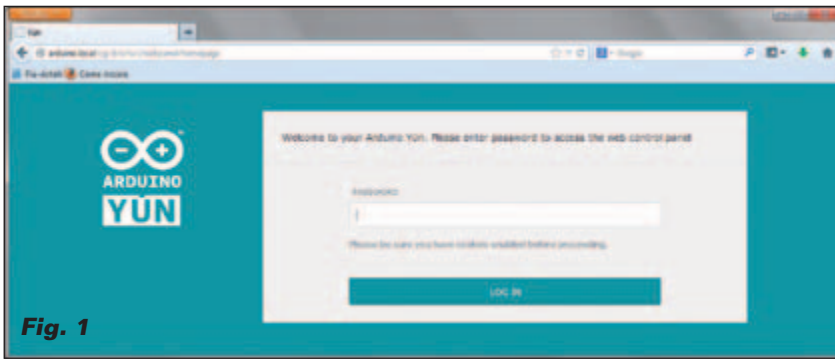
MARCO MAGAGNIN



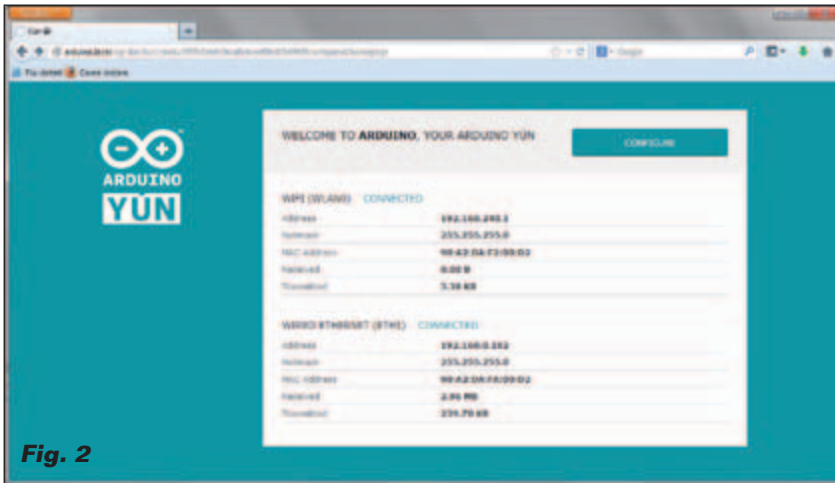
Aquí está la penúltima tarjeta de Arduino. El ADN es el de la familia, pero la evolución la ha abierto a las nuevas exigencias: Linux embebido, conectividad ethernet y WiFi. Para probarla, hemos desarrollado una aplicación que envía Tweets al producirse eventos locales.

**A**unque YUN ya lleva algún tiempo en el mercado, somos muchos los que no habíamos tenido oportunidad de cacharrear con él, así que hemos decidido compartir con vosotros nuestra primera experiencia con esta placa. Como anunciaron, la tarjeta integra las características de un Arduino Leonardo estándar, con una sección hardware adjunta capaz de albergar una distribución GNU/Linux completa, en este

caso OpenWRT, específicamente destinada a la conectividad con Internet (es la distribución open source utilizada en los router y access point WiFi). Con el producto en la mano, lo primero que nos preguntamos es: ¿seguro que me habéis dado una Yún? La caja tiene las mismas dimensiones que aquella



**Fig. 1**



**Fig. 2**

un cargador a 5V con toma micro USB (para entendernos, la misma que se usas para recargar los teléfonos) al conector micro USB en el centro de la tarjeta. El LED ON se enciende en verde y poco después el LED WAN cambia a amarillo. ¿Qué quiere decir? Abrimos un buscador y escribimos la dirección:

*http://arduino.local*

(en realidad es necesario leer un poco en el "getting started" que hay en la página web *www.arduino.cc* dedicada a Yún). Sorpresa: obtenemos la página que se en la Fig. 1. Ponemos la contraseña "arduino" (es la contraseña del usuario por defecto ("root")) y vemos que sucede. Obtenemos la página que vemos en la Fig. 2, que nos muestra la configuración de las interfaces de red activas. Esta página ya nos dice muchas cosas. Yún nace con un sistema operativo GNU/Linux

de Arduino Uno. Abierta la caja nos hemos asegurado: sí, es una Yún, aunque tenga las mismas dimensiones de una Arduino Uno (obvio, porque tiene que ser compatible con los mismos shield). Teniéndola entre las manos se ven dos entradas USB, una Ethernet, una entrada para micro SD-Card y los conectores estándar de un Arduino. Además están presentes tres pulsadores y un conector para una antena externa a la sección WiFi. Todo rodeado por una buena cantidad de electrónica SMD situada en ambas caras del circuito impreso. Queriendo tener un enfoque típico de experimentadores provenientes del mundo GNU/Linux, antes de profundizar más en las características de la tarjeta probemos a ver qué sucede, en todo caso, ¡después leeremos el manual! Entonces, conectamos un cable de red a la toma Ethernet de Arduino Yún y lo alimentamos mediante

**Fig. 3**



instalado, con los interfaces de red preconfigurados y un servidor web activo con la aplicación LuCI ya instalada para configurar el funcionamiento en GNU/Linux. A propósito: ¿habéis notado que en el primer arranque hemos usado un nombre (*arduino.local*) y no una dirección IP?

Esto significa que no es necesario averiguar la dirección IP asignada a Yún por el servidor DHCP de nuestra red, y si podemos acceder directamente con el nombre host de la tarjeta, debido a que la distribución GNU/Linux de Yún, orientada a la conectividad, está dotada de un servidor DNS interno.

En la Fig. 2 luego veremos cuál es la dirección IP asignada a la tarjeta WAN y la configuración de la sección WiFi, ya activada y preconfigurada como punto de acceso con una IP estática. Esto significa que podemos conectarnos a la Yún

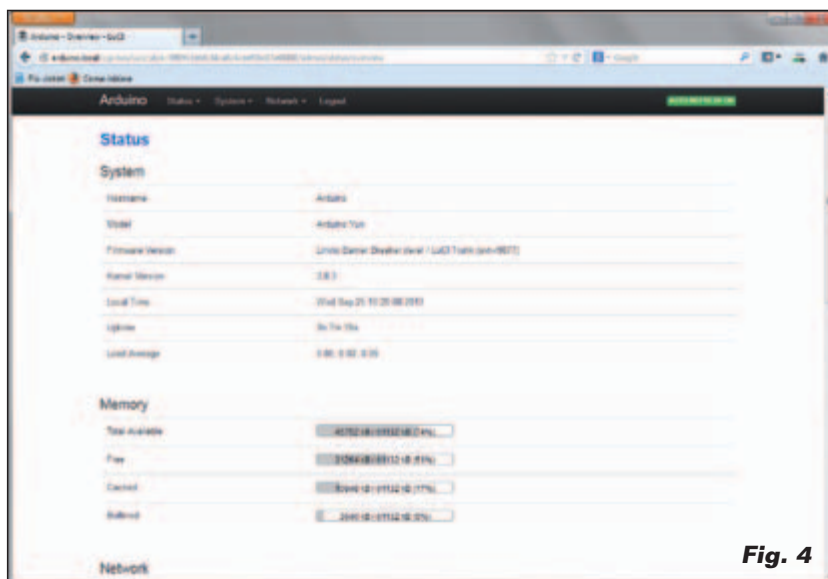


Fig. 4

Continuamos curioseando y hacemos clic en el pulsador grande “Configure” arriba a la derecha de la página: obtenemos otra página de configuración (Fig. 3) que muestra, arriba a la derecha, un link que atrae nuestra atención. Pero primero aprovechamos para configurar la zona horaria exacta y seleccionamos el check box

secciones de administración del sistema GNU/Linux (Fig. 5). A propósito, la distribución personalizada para Arduino Yún se llama “Linino” y, como ya hemos dicho y cómo podemos ver en el menú de configuración, ha derivado de Open WRT. Si por curiosidad vamos a la sección “Network” y “Wifi”, en el inte-



Fig. 5

también por WiFi, simplemente conectándonos a la red con SSID “Arduino Yun-<mac-address-de-la-tarjeta>”. Sin monitor, sin teclado ni ratón, ninguna conexión USB con el PC y, si queremos, ningún cable de red. Todo esto con el objetivo de facilitar las cosas para aquellos que son principiantes.

“Open” en la sección “Rest API Access”; luego hablaremos sobre esto. Bien, pinchamos sobre el link “advanced configuration panel” y entramos en un verdadero y propio ambiente de administración, se puede ver en la Fig. 4. Desde el menú arriba, al estilo de Google, podemos acceder a las diferentes

rrior de la página, presionamos el pulsador “Edit” y desplazamos la página un poco hacia abajo, encontramos la sección que nos permite configurar el módulo WiFi de la tarjeta en las diferentes modalidades (Access Point, Client, Ad Hoc, ecc). Las otras secciones nos permiten aplicar reglas de

accesos de seguridad (Firewall), reglas de Routing, DHCP y DNS. En definitiva, nos encontramos frente a un verdadero y propio dispositivo de conectividad que nos permitirá resolver una vasta gama de aplicaciones, de los requisitos de procesamiento y de configuración de red.

Una última curiosidad: probamos a conectarnos a la tarjeta utilizando los clientes SSH Putty y WinSCP. Nos conectamos con la dirección asignada por el DHCP de nuestra red (lo vemos en la página de la Fig. 2), con usuario "root" y contraseña "arduino". Obtenemos los resultados que vemos en la Fig. 6 y Fig. 7, que nos demuestran que también el servidor SSH está activo y funcionando "out of the box".

Con lo que hemos visto ya, sería material suficiente para escribir un libro sobre las posibilidades de esta tarjeta, que contiene una verdadera y propia arquitectura integrada que reúne en pocos centímetros cuadrados un punto de acceso WiFi, un sistema GNU/Linux con aplicaciones web ya disponibles y un microcontrolador Arduino capaz de interactuar con el mundo exterior de las maneras que ya conocemos. Casi nos

olvidamos de que se trata de una tarjeta Arduino...

En este momento no podemos dejar de subrayar el valor didáctico de este dispositivo, que, al enfoque simplificado de la programación de los microcontroladores, añade la posibilidad de acercarse de manera muy simplificada al mundo GNU/Linux en su configuración embebida y al mundo de la conectividad inalámbrica y de red en general, en sus diferentes configuraciones. Profundicemos estos argumentos, aunque no son de corte específicamente electrónico como en artículos específicos; somos conscientes que el mundo de la electrónica, de la informática y de las arquitecturas de red se están fundiendo en un todo fuertemente integrado.

Desde este punto de vista, Arduino Yún representa el punto de partida óptimo para empezar a experimentar la integración de las distintas disciplinas electrónicas, informáticas y de red, además de para realizar aplicaciones que unen la adquisición y elaboración de datos provenientes de sensores con la publicación y gestión de los mismos por el mundo exterior. En estos momentos, la máxima expresión de esta tendencia es el



La necesidad de la conectividad de red cableada e inalámbrica para las tarjetas embebidas se siente no solo por el equipo Arduino, sino también por cualquiera que desarrolle prototipos de sistemas embebidos. Veamos que se mueve en este sector.

### UDOO

El mismo día del anuncio por parte del equipo de Arduino de la disponibilidad de la Yún, sobre el blog de Udo se anunciaba el inicio de la producción de la tan deseada placa Udo, un proyecto financiado gracias a Kickstarter, con la intención de superar la pobre propensión de un sistema de microcontrolador como Arduino a la

dispositivo myRIO de National Instruments, que integra en un único chip la potencia en tiempo real de un FPGA y un procesador que aloja un GNU/Linux con la conectividad entre los dos, basada en 16 canales DMA ultra veloces. Es ahora el momento de pararnos y profundizar las características generales de la Yún. Tomándola de nuevo entre las manos notamos

Fig. 6

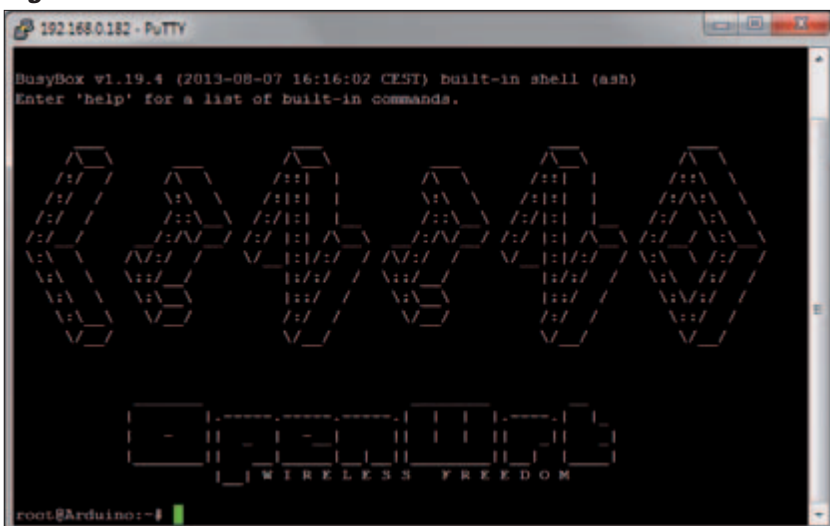
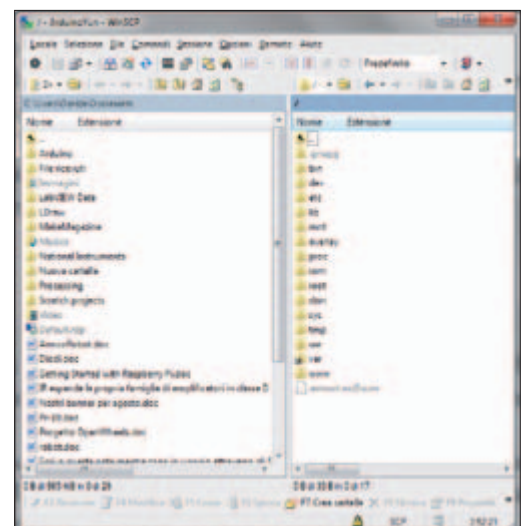


Fig. 7



conectividad (cosa usual para sistemas basados en Linux, que pueden proporcionar el acceso a Internet de manera natural). De hecho la tarjeta se acompaña de dos procesadores: un Freescale i.MX6 ARM quad o dual core (depende de la versión que elijáis) con 1 GB di RAM, para la sección GNU/Linux, y un Atmel SAM3X8E ARM Cortex-M3 para la sección de Arduino (el mismo que Arduino Due). La comunicación entre las dos secciones debería hacerse a través de un enlace serie dedicado. La tarjeta se acompaña de una versión personalizada de Ubuntu, pero gracias al apropiado slot para la tarjeta SD es posible instalar otra distribución Linux o Android. La tarjeta está dotada de módulo

WiFi y expone también un conector Ethernet, una única salida HDMI, un puerto SATA, un controlador USB, un jack para los altavoces y uno para el micrófono.

## NI myRIO

Un sistema embebido compacto y dotado de gran conectividad es myRIO de National Instruments, creado para ofrecer a los estudiantes la posibilidad de desarrollar rápidamente sistemas de ingeniería reales y completos. Basado en la misma tecnología de la plataforma NI CompactRIO, NI myRIO utiliza un SoC Xilinx Zynq-7020 que combina un procesador dual-core ARM @Cortex™-A9 y un FPGA 7-Series de Xilinx con 28000 celdas lógicas programables. myRIO se

programa en LabVIEW, con lo cual es rápido e intuitivo. NI myRIO dispone de 10 entradas analógicas, seis salidas analógicas, canales I/O audio y 40 líneas de I/O digitales. Además está dotado de WiFi en la tarjeta, de acelerómetro a tres ejes y diferentes LED programables.

La introducción de NI myRIO en la arquitectura LabVIEW RIO (Reconfigurable I/O) permite a National Instruments proporcionar instrumentos a todos los niveles de experiencia, desde el aprendizaje base de los estudiantes, hasta las soluciones más avanzadas para los diseñadores. Ideal para lecciones en clase y en laboratorio, NI myRIO incluye recursos didácticos gratuitos, es compatible con todos los NI miniSystems y

con una vasta gama de actuadores y sensores de terceras partes. NI myRIO es programable en LabVIEW, C, C++ y otros entornos de desarrollo, de manera que es fácil de integrar en cursos didácticos sobre controles, la robótica, la mecatrónica y los sistemas embebidos.



otra característica muy útil: sobre los lados exteriores de los conectores para los shield están serigrafadas las referencias principales de cada pin, que permiten utilizar la tarjeta sin necesitar el esquema de las conexiones. En la Fig. 8 se puede ver la imagen de la tarjeta con las conexiones externas y los pulsadores de reset.

Pero procedamos en orden: del lado "microcontrolador", Yún utiliza el chip Atmega32U4, el mismo de Arduino Leonardo, del cual reproduce las características (16 MHz de reloj y 32 kB de RAM, de los cuales bootloader ocupa cuatro). En cuanto a la alimentación, no hay regulador a 5V, por lo que es necesario prestar atención de no alimentar la tarjeta con una tensión superior (puede causar daños irreparables), si se tiene la intención suministrar la alimentación a través del contacto Vin. En referencia a esto, la toma de alimentación en formato micro-USB es una ayuda: por su naturaleza, las alimentaciones de

este tipo de salida suministran 5V. Esta micro-USB sirve también para conectar la tarjeta al PC para la programación tradicional por parte de Arduino. Pero es también posible ejecutar la carga de los sketch a través de la red, física o WiFi directamente desde el IDE. A propósito: el IDE a utilizar, es descargable libremente (como siempre) desde la dirección web

<http://arduino.cc/en/Main/Software>, es la versión 1.5.4 r2 ó superior, que resuelve un bug de funcionamiento bajo Windows 8. Existen también versiones para GNU/Linux y Mac OS X. Otra diferencia de comportamiento respecto a Leonardo, está en el hecho que el microcontrolador no se resetea cuando se abre el monitor serie del IDE. Además

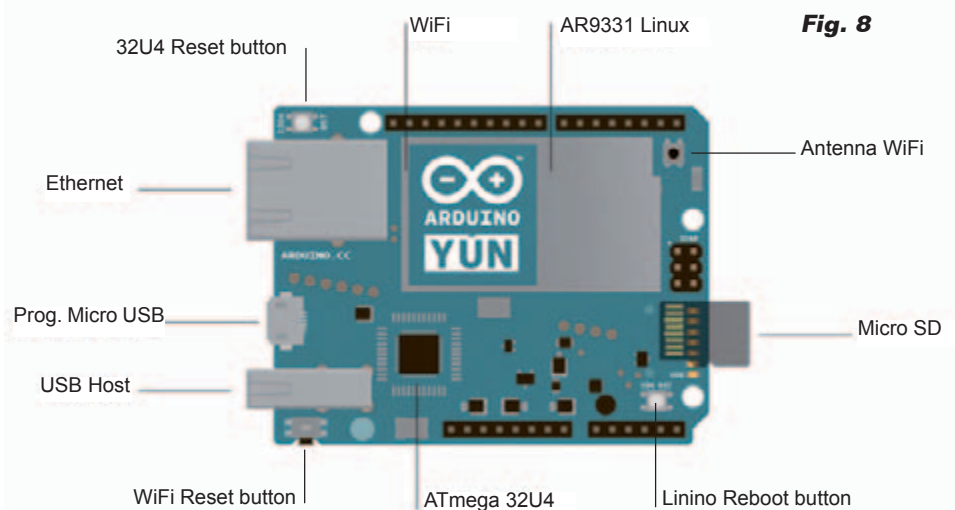


Fig. 8

no es posible utilizar la interfaz Serial1, reservada a la comunicación con la sección GNU/Linux. Para resetear el microcontrolador está el pulsador específico "32U4 Reset Button" al lado del conector Ethernet (ver Fig. 8).

La sección GNU/Linux se gestiona por el segundo procesador, un Atheros AR9331 a 400 MHz. Probablemente en la elección ha sido determinante la particular predisposición del SoC para albergar la distribución GNU/Linux OpenWRT, de la cual se ha derivado Linino, y por los consumos particularmente reducidos. La sección GNU/Linux está dotada de un conector Ethernet y de un conector USB-A Host. Hay un slot para alojar una micro SD-Card, sobre la cual direccionar la parte "personal" del sistema de archivos

de la sección GNU/Linux, de manera que se ahorre la poca RAM NAND asignada al procesador Atheros (por otra parte caracterizada por un número de posibles reescrituras bastante limitado: cerca de 100.000). Para hacer esto es suficiente insertar una micro SD en el slot y crear dos carpetas que se reconozcan automáticamente y se utilizan por el sistema operativo:

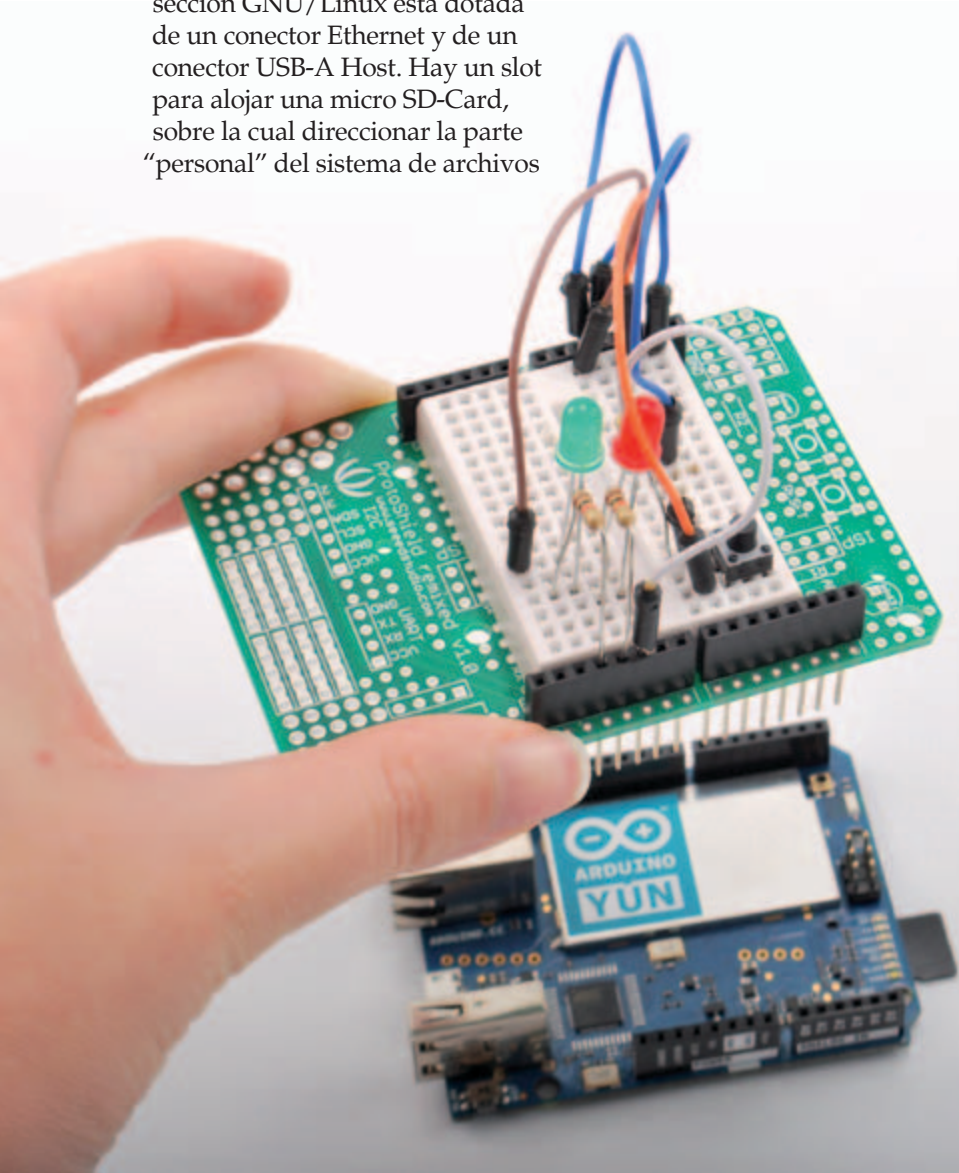
```
/arduino e  
/data
```

En realidad, en el momento del boot el sistema operativo monta

automáticamente todos los dispositivos de almacenamiento masivo externos denominados "sda", más alguna otra cosa. Veremos en profundidad el detalle el uso de estas carpetas.

En la sección GNU/Linux está presente también un módulo WiFi, conectado también a un procesador Atheros AR9331, que proporciona accesibilidad inalámbrica al sistema completo, con la posibilidad adicional de conectar una antena externa.

Vayamos ahora a los pulsadores de reset: el primero lo hemos visto ya y permite el clásico reset del microcontrolador Atmega32U4. Los otros dos se encargan del reset del procesador Atheros AR9331 y del módulo WiFi. La presión del pulsador "Linino reset button" permite ejecutar el reboot del sistema operativo Linino. Claramente todos los datos en memoria se perderán y todos los procesos se terminarán. El pulsador "WiFi reset button" tiene una doble función: pulsado brevemente resetea el módulo WiFi a la configuración de fábrica, ósea al funcionamiento como Access Point con IP estática igual a 192.168.240.1. Sin embargo, pulsando por un tiempo (5 segundos) el pulsador, se resetea el módulo WiFi. El reset del módulo WiFi hace también el reboot de Linino. Es importante darse cuenta que no debe presionarse la tecla más de 5 segundos, porque en cuanto se presiona más de 30 segundos, provoca la restauración del sistema operativo Linino con la versión original, o lo que es lo mismo la configuración de fábrica. Se ha elegido este funcionamiento para ayudarnos en el caso en que queramos recuperar el sistema después de algunas "torpes operaciones". La función de restauración nos da la libertad de experimentar sin temor de encontrarnos en situaciones sin salida: más



simple que así...

Restaurando el sistema claramente se pierden todas las configuraciones hechas después del primer encendido y otras actualizaciones ejecutadas sobre Linino.

Ahora digamos algunas palabras sobre los LED de la tarjeta Arduino Yún dispone de los siguientes LED:

- USB - azul, indica actividad sobre el conector USB;
- WLAN (WiFi); WiFi activo;
- POWER; indica la presencia de alimentación;
- WAN; indica la presencia de una conexión de red Ethernet activa;
- Pin 13; reporta el estado del pin 13 (con tarjeta nueva parpadea por el programa Blink cargado en origen);
- Serial TX; señala actividad en el puerto serie en transmisión;
- Serial RX señala actividad en el puerto serie en recepción.

En la Fig. 9 puedes ver el esquema de la arquitectura general de la tarjeta. Son claramente visibles las dos secciones "Arduino" y "Linino", con los periféricos asignados a cada uno y el "puente" de conexión indicado como "Bridge". Esta característica es la que permite a los dos ambientes trabajar conjuntamente y constituye una innovadora solución Linux embebido capaz de trasladar al mundo exterior todas las funcionalidades del microcontrolador Arduino, hasta ahora confinadas solo al ámbito local. A decir verdad, en el pasado se han presentado soluciones de conectividad basadas en shield Ethernet, WiFi, Bluetooth, etc., pero todas tenían que hacer frente a la vocación "microcontrolador" de los dispositivos y con los limitados recursos de memoria disponibles. Era casi imposible gestionar distintos usuarios y páginas HTML complejas, por no

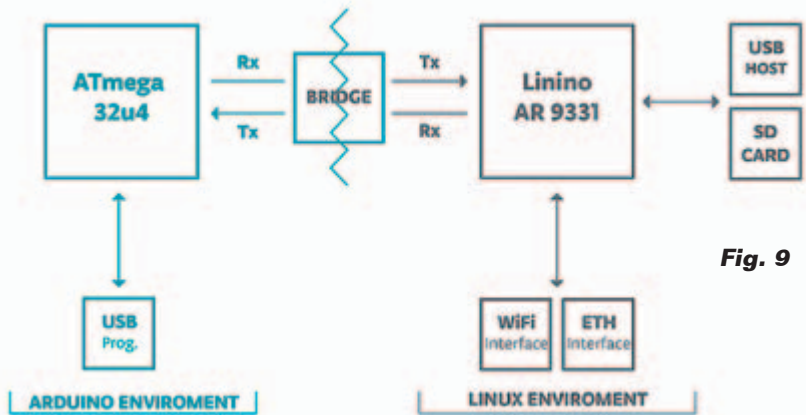


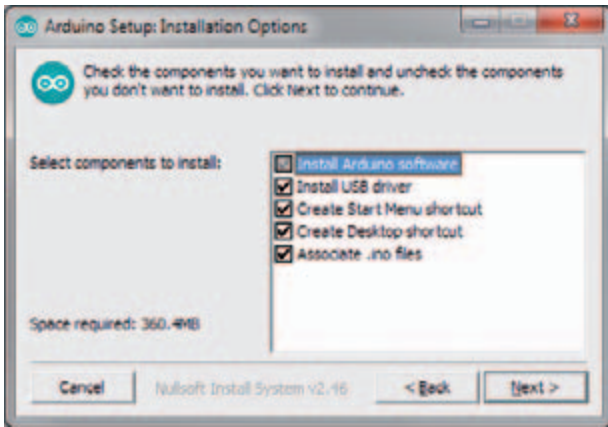
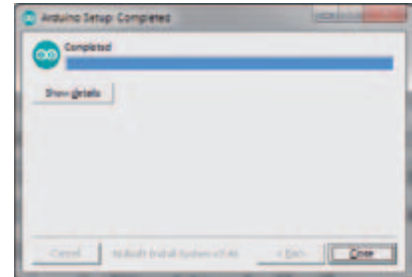
Fig. 9

hablar de la gestión de las concurrencias entre accesos web y gestión de sensores y periféricos de E/S. Con esta arquitectura se ha superado todo esto y cada sección se ocupa mejor de su respectiva funcionalidad. La disponibilidad de librerías y ambientes de desarrollo adecuados a gestionar el "puente" de comunicación integrado en el IDE y desarrollo en la lógica de simplificación propia de la "filosofía" Arduino, hacen el resto, posibilitando el desarrollo de soluciones, incluso complejas, con la habitual sencillez y linealidad. Paremos un momento en la solución Bridge: los procesadores ATmega 32u4 y Atheros AR9331 se conectan mediante puerto serie; del lado de Arduino, la librería "Bridge" permite realizar el "puente" hacia el ambiente Linino, utilizando el puerto serie Serial1, que no puede utilizarse para otros propósitos. Simplificando mucho, la librería "Bridge" envía sobre el puerto serie "comandos" de "consola" a Linino. Los "comandos" permiten activar del lado Linino los procesos capaces de hacer de puente con los servicios y aplicaciones "GNU/Linux". Del lado de Linino, el puerto serie es asignado, en el momento del boot, a la interfaz CLI (Command Line Interface). De esta manera todos los paquetes de datos de comandos enviados desde la librería "Bridge" se interpretan por Linino

como comandos de shell, que activan programas en Python que se encargan de realizar las diferentes modalidades de interfaz disponibles. De esta manera es posible instaurar la comunicación entre la sección Arduino y Linino, simulando una emulación de terminal remoto (Consolas), como shell para la gestión de comandos y procesos remotos, como puente para leer y escribir archivos sobre la SD-Card de Linino desde los sketch Arduino, como puente de comunicaciones de URL HTTP para gestionar desde el sketch las peticiones pasadas del web server uHTTPd de Linino (micro Hypertext Transfer Protocol Daemon) y en otras modalidades. Siempre del lado de Linino, están disponibles distintos instrumentos para utilizar esta tipología de comunicación. Citamos el framework REST (REpresentational State Transfer) que permite la interfaz entre los pines de E/S del lado Arduino mediante URL HTTP del tipo:

<http://arduino.local/arduino/digital/13/1>

que significa: lleva a nivel alto el pin de E/S digital 13. En próximos artículos presentaremos ejemplos de aplicaciones basadas en este framework. Otro framework gestionable con la librería "Bridge" es *Spacebrew*, que permite a diferentes dispositivos Client comuni-

**Fig. 10****Fig. 11****Fig. 12**

carse entre ellos en modo Publisher o Subscriber por medio de un server “coordinador” público o privado. Por fin, existe la posibilidad de la interfaz de las API de Temboo que permiten comunicar con una gran cantidad de servicios “web” como las redes sociales, las educativas, el Khan Academy, los servicios de transporte, las médicas e infinitas otras. En este artículo, como primera muestra de uso, os proponemos un ejemplo de uso de las API Temboo con la librería “Bridge” con el objetivo de enviar Tweets en función de los eventos procedentes de las E/S de Arduino. Para realizar esto, debemos preparar un sketch para compilar y transferir sobre la memoria flash de la sección Arduino de Yún. Aprovechamos para introducir el uso del IDE con la Yún conectada a través red Ethernet.

### EL IDE DE ARDUINO

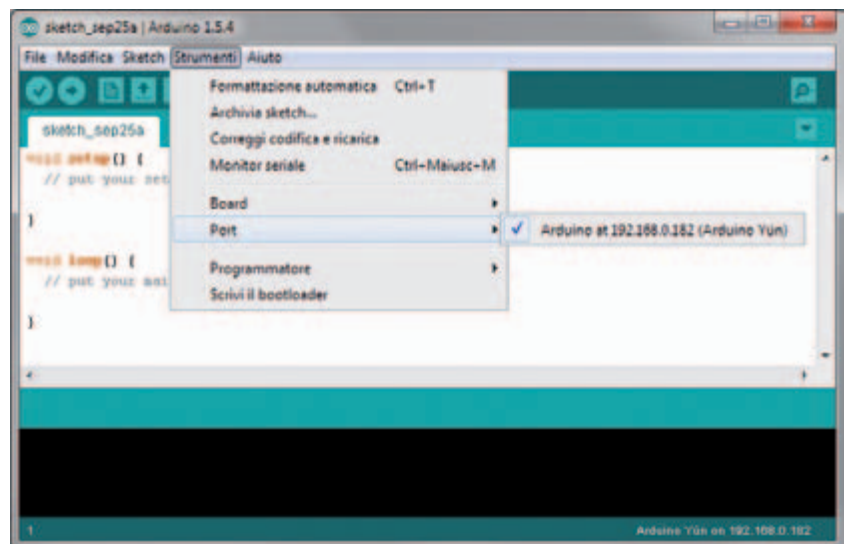
Hemos anticipado ya que podemos descargar el IDE de desarrollo para Arduino Yún del sitio web de Arduino. Es necesario descargar los dos archivos *arduino-1.5.4-r2-windows.exe* y *arduino-1.5.4-r2-windows.zip*, o, si existen, las sucesivas versiones. El primer archivo permite la carga de los drivers USB necesarios para reconocer la tarjeta Yún e instalar el segundo paquete. Ejecutamos el primer archivo, aceptamos los términos de licencia y seleccionamos todas las opciones como en la

**Fig. 10.** Respondemos con “Install” a las peticiones de las pantallas sucesivas (**Fig. 11**) y al final confirmamos con “Close” (**Fig. 12**). Terminada la instalación lanzamos el IDE desde el icono del desktop o desde el menú “Start”, con lo que se abrirá la interfaz del IDE, que contiene algunas novedades. En la **Fig. 13** vemos la configuración del puerto de comunicación entre el IDE y la tarjeta Yún; en realidad se presenta la dirección IP de la tarjeta: lo seleccionamos y activamos el canal de comunicación, sin necesidad de conectar cables USB e instalar driver. Tenemos presente que este método funciona, obviamente, también si la tarjeta está conectada en modo inalámbrico, lo que constituye una gran novedad respecto al pasado, permitiendo programar y configurar

la Yún aunque esta se encuentre en un lugar recóndito o montada sobre un dispositivo móvil, como ejemplo un robot. Hecha la elección del puerto de comunicación, nos queda configurar el tipo de dispositivo - obviamente Arduino Yún -, a través de la ruta “Instrumentos”>>“Board”. El resto corresponde a lo que hemos hecho siempre con las otras tarjetas Arduino.

### LA INTERFAZ TWITTER

La aplicación que os proponemos aquí está basada en uno de los ejemplos proporcionados con el IDE y permite, utilizando las librerías y el servicio Temboo,

**Fig. 13**



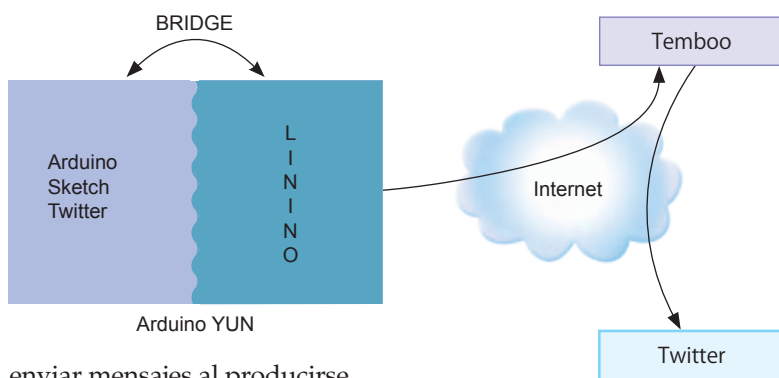


Fig. 14

enviar mensajes al producirse un evento local, que en nuestro caso es la presión de un pulsador conectado al pin 5 de Arduino Yún. Para obtener este resultado debemos realizar una cadena de comunicación como la que podéis ver en la Fig. 14. El servicio Temboo ofrece la posibilidad de conectarse para enviar y recibir mensajes de y para distintos tipos de servicios disponibles en red, entre los cuales las redes sociales, servicios de pago como PayPal, servicios meteorológicos y médicos, de formación y muchos otros, para un total de más de cien clases de API web disponibles. Simplificando mucho se conecta a Temboo, se instancian objetos utilizando las clases puestas a disposición por el mismo Temboo y, mediante este mecanismo, preguntamos a Temboo para enviar o recuperar informaciones de o hacia un cierto servicio.

El registro gratuito permite enviar/recibir 1.000 mensajes al mes; para gestionar más es necesario una suscripción de pago.

El programa que "gira" sobre la sección Arduino predispone la conexión hacia el servicio Temboo, que incluye tanto las credenciales de acceso al mismo servicio, como a la aplicación Twitter a la que se quiere enviar el mensaje. En cuanto a Temboo, en la sección "setup" es instanciado el objeto necesario para la conexión con Twitter. Al término del setup, el programa entra en el bucle principal, donde

en cada ciclo se verifica si el pulsador ha sido pulsado o no. Si no ha sido presionado, el programa se prepara para un nuevo ciclo; en caso contrario, se inicia el proceso de envío del mensaje Twitter. Esto conlleva la compilación de los datos del objeto con el ajuste del mensaje mismo, de las credenciales de acceso a Temboo y Twitter, y finalmente con la invocación del método "run" del objeto que procede a la ejecución del proceso verdadero y propio de envío del mensaje. En este punto, el programa se suspende hasta recibir el estado de retorno de la ejecución, con el informe de cualquier anomalía.

Para tener bajo control el comportamiento del programa hemos añadido dos LED: uno verde que queda encendido después de presionar el pulsador durante todo el tiempo necesario para enviar el Tweet, y uno rojo que se enciende parpadeando de manera intermitente en caso que se verifique algún error, continuando en este estado hasta que el error se resuelve. Para realizar el circuito simple de nuestra aplicación, hemos utilizado el shield experimental para Arduino con la breadboard de fácil inserción. Ambas tarjetas están disponibles en Futura Elettronica, con los códigos

7300-PCBPROTOV10 (más 1 7300-STRIP6 y 3 7300-STRIP8) y 7300-BBMINIW.

En la Fig. 15 veis el prototipo, y su esquema de cableado está dibujado en la Fig. 16.

### CONFIGURACIÓN TWITTER Y TEMBOO

Para realizar este ejemplo es necesario disponer de cuenta Twitter a la cual enviar los Tweets y configurar una aplicación específica de la cual utilizaremos las credenciales de acceso al interior de nuestro sketch. Además es necesario registrarse en Temboo para obtener las credenciales necesarias para utilizar el servicio y configurar nuestro sketch también con estas. Empecemos con Twitter: si no tenemos cuenta Twitter creamos una, siguiendo las instrucciones indicadas en el proceso de registro. Una vez obtenido el password (vía e-mail) vamos a la dirección <https://dev.twitter.com/apps/new>, donde, si no estamos logeados ya, nos pedirán el userid y password de Twitter (Fig. 17).

En la página que nos aparece (Fig. 18) creamos una aplicación rellenando los campos obligatorios. En cuanto al campo "website" podemos proporcionar una dirección web inventada, mientras sintá-

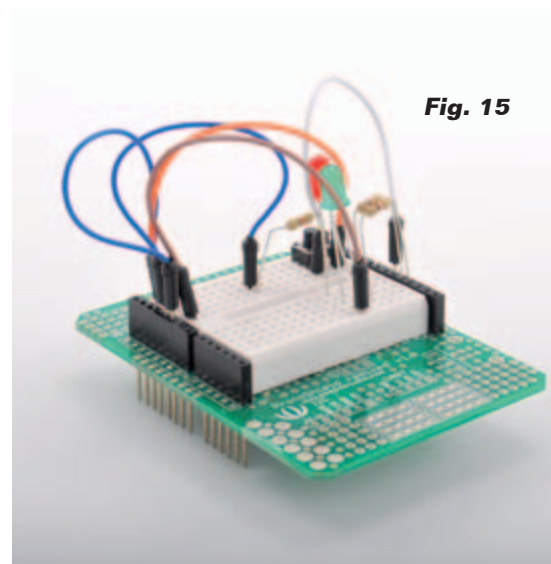


Fig. 15

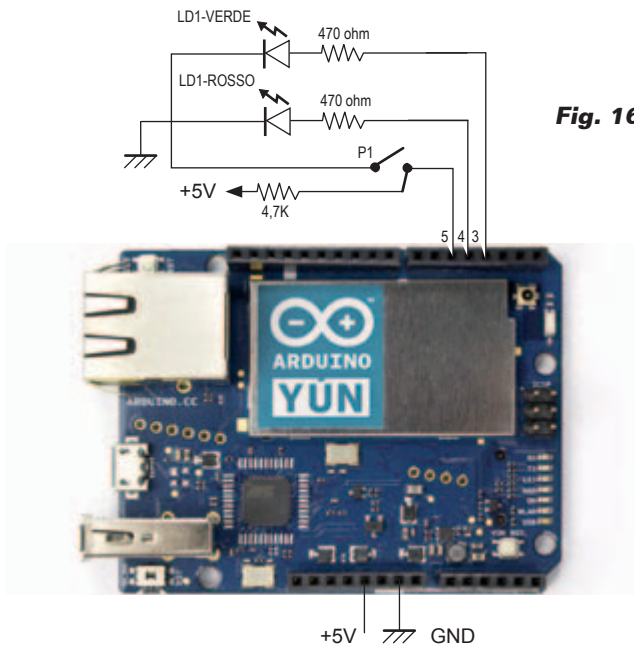


Fig. 16

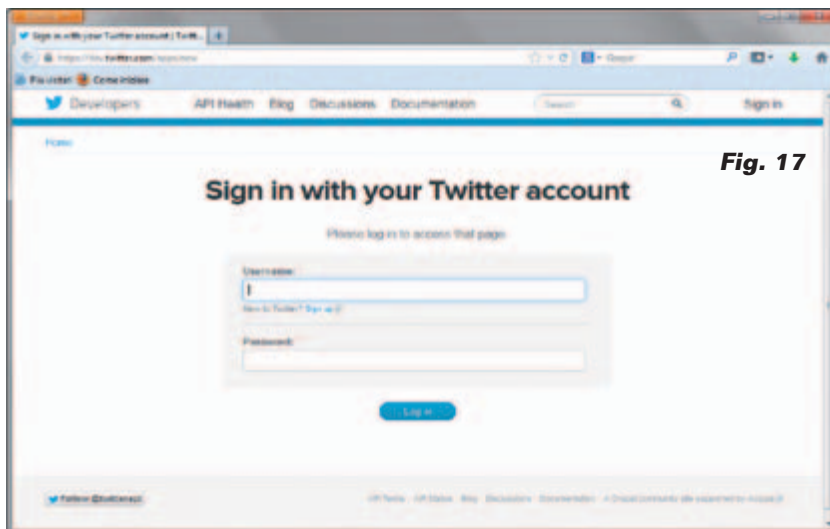


Fig. 17

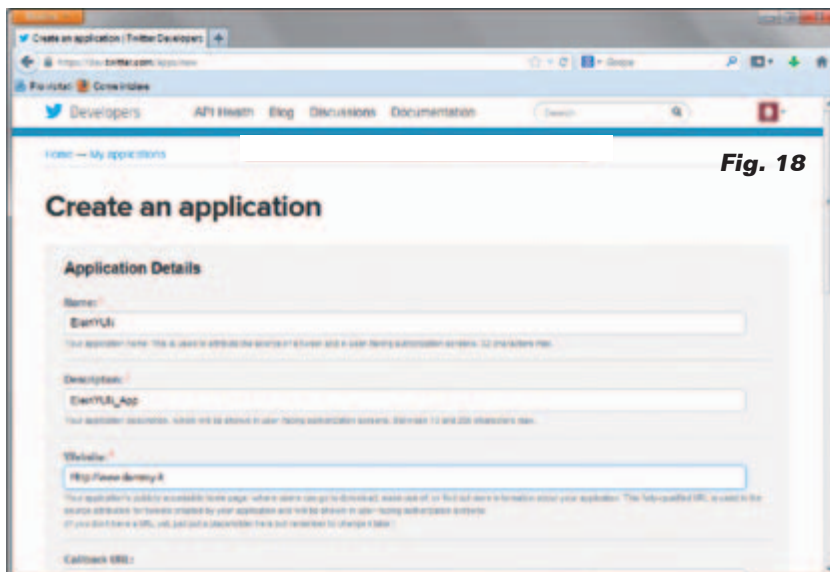


Fig. 18

ticamente sea correcta: tipo *www.noesunsitio.com*.

Rellenamos el campo captcha y confirmamos haciendo clic sobre el pulsador azul del final de la página. En la página "settings" de la Fig. 19, en la sección "Application Type" seleccionamos la opción "Read and Write". Confirmamos también esta página y pasamos a la siguiente, que se puede ver en la Fig. 20, donde nos proporcionan la primera serie de códigos que tendremos que insertar en nuestro sketch. Se trata, en particular, de los campos "Consumer key" y "Consumer secret". En la sección "Your access token" pulsamos el botón azul para crear una copia de token de acceso.

De esta sección nos servirán los campos "Acces token" y "Access token secret". Mientras tenemos abierta esta página, abrimos una nueva pestaña en el browser y vamos a la dirección de registro del servicio Temboo (Fig. 21) <https://temboo.com/signup>; insertamos nuestro e-mail en el único campo disponible y atendemos a la confirmación por parte de Temboo. Una vez recibido el mail pulsamos sobre el link proporcionado y, en la página de confirmación del registro, seleccionamos el usuario y un password que utilizaremos para el login en la página (Fig. 22) en la dirección <https://temboo.com/login>.

Insertamos todos los campos requeridos y entramos en la página de nuestra cuenta, como se ve en la Fig. 23; aquí seleccionamos el elemento de menú "APPLICATION KEYS" y una vez dentro, clicamos sobre la opción "(show)" pegada a la fila de asteriscos bajo el epígrafe "KEY". De esta manera obtenemos los últimos datos necesarios para personalizar nuestro sketch: en la práctica, el equivalente de userid y password necesarios para usar las librerías de las

clases y las API disponibles por el mismo Temboo. De hecho, un tipo de servicio de “consumo” donde un cierto número de usos al mes es gratuito, mientras que después es necesario una suscripción de pago.

### EL SKETCH ARDUINO

Pasemos ahora a describir el funcionamiento del sketch para Arduino (**Listado 1**) cuya riqueza de comentarios no debería dejarnos dudas sobre el funcionamiento. Copiamos el código en el IDE, teniendo en cuenta que es siempre posible descargarlo de la página web [www.nuevaelectronica.com](http://www.nuevaelectronica.com). Ahora copiamos y pegamos los campos de las credenciales de la página web de Twitter, que hemos dejado abierta, y Temboo, en los campos relativos del sketch. Sugerimos el método de copiar y pegar porque debido a la complejidad de la clave, en caso de copia manual, puede llevar con certeza a cometer errores difícilmente identificables. Hemos elegido la modalidad upload del código sobre Arduino mediante ethernet, renunciando a usar el USB. Configuramos correctamente “Board” y “Port” y hacemos clic sobre el pulsador “Carga”: primero ejecutará la compilación del código y a continuación ejecutará el código compilado sobre la tarjeta Yún. Eso es todo. Cuando termine pulsamos el botón sobre la breadboard: el LED verde quedará iluminado el tiempo necesario para el envío del Tweet a nuestra cuenta; cuando termine se apagará y será posible pulsar el botón de nuevo. Si ocurre un error se enciende también el LED rojo y vuelve a intentar el envío. Para ver qué ocurre, podemos provocar el error desconectando el cable de red después de haber pulsado y después de un poco volvemos a conectar el cable y vemos que, al



Fig. 19

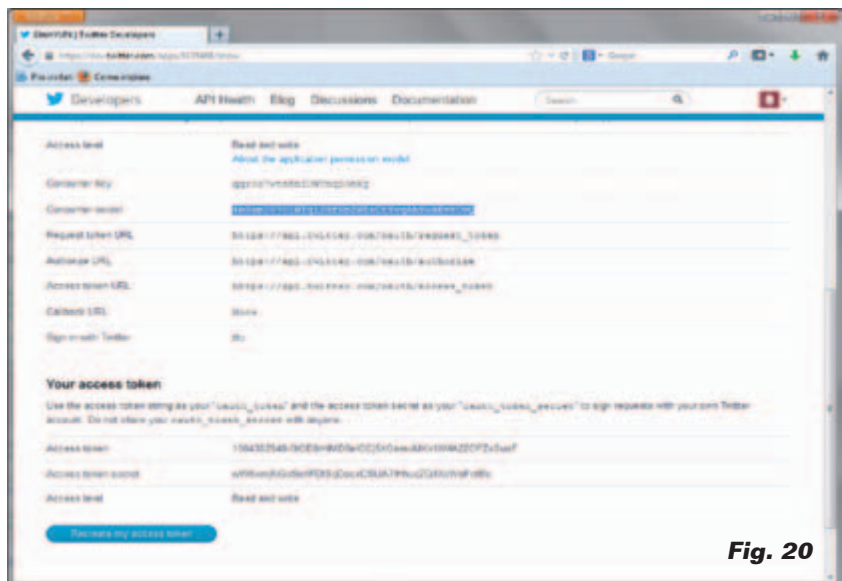


Fig. 20

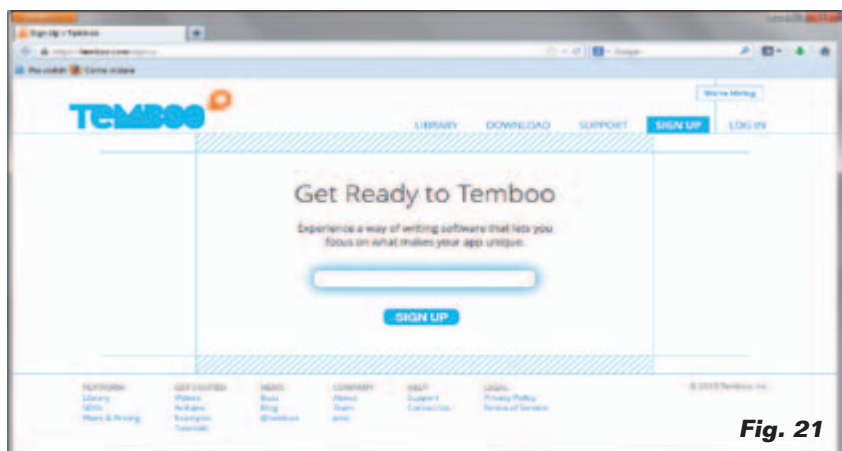


Fig. 21

## Listado 1

```
/*
  Envía un Tweet
*/

#include <Bridge.h>
#include <Temboo.h>

/** SUSTITUIR LAS CREDENCIALES */

const String TWITTER_ACCESS_TOKEN = "1904382948-0...v1XhfA2ZOFZxSuxF";
const String TWITTER_ACCESS_TOKEN_SECRET = "wW5xmjNGoSoX.....7IHhucZQ4XoWoFotBc";
const String TWITTER_CONSUMER_KEY = "qqr8c7wt.....qD0eXg";
const String TWITTER_CONSUMER_SECRET = "4bNamlV7U1.....eCtYvqAhSwsKvH0vQ";

const String TEMBOO_ACCOUNT = "dev....cain"; // your Temboo account name
const String TEMBOO_APP_KEY_NAME = "myF....pp"; // your Temboo app key name
const String TEMBOO_APP_KEY = "1b.....32-9"; // your Temboo app key

int numRuns = 0; // Contadores para gestionar el envío en caso de error
int maxRuns = 1;

int led_rojo = 4; // LED error en curso
int led_verde = 3; // LED envío TWEET en curso
int butTonPin = 5; // Pulsador de envío TWEET
int buttonState = 1;

void setup() {
  pinMode(led_rosso, OUTPUT);
  pinMode(led_verde, OUTPUT);
  pinMode(butTonPin, INPUT);
  Bridge.begin(); // Inicializa la comunicación con LINUX vía serie compartida
}

void loop()
{
  buttonState = digitalRead(buttonPin);
  if (buttonState == LOW) { // Se pulsa el botón

    while (numRuns < maxRuns) { // intenta los envíos hasta buen puerto

      // Texto del mensaje TWEET
      String tweetText("Alguien ha pulsado el botón - " + String(millis()) );

      digitalWrite(led_verde, HIGH);

      TembooChoreo StatusesUpdateChoreo; // Instancia el objeto para envío TWEET

      // Invoca el client Temboo
      StatusesUpdateChoreo.begin();

      // Pone las credenciales para acceder a Temboo
      StatusesUpdateChoreo.setAccountName(TEMBOO_ACCOUNT);
      StatusesUpdateChoreo.setAppKeyName(TEMBOO_APP_KEY_NAME);
      StatusesUpdateChoreo.setAppKey(TEMBOO_APP_KEY);

      // Pone la librería de API a utilizar (Twitter > Tweets > StatusesUpdate)
      StatusesUpdateChoreo.setChoreo("/Library/Twitter/Tweets/StatusesUpdate");

      // Pone las credenciales de acceso a Twitter
      StatusesUpdateChoreo.addInput("AccessToken", TWITTER_ACCESS_TOKEN);
      StatusesUpdateChoreo.addInput("AccessTokenSecret", TWITTER_ACCESS_TOKEN_SECRET);
      StatusesUpdateChoreo.addInput("ConsumerKey", TWITTER_CONSUMER_KEY);
      StatusesUpdateChoreo.addInput("ConsumerSecret", TWITTER_CONSUMER_SECRET);

      // referencia al mensaje a enviar
      StatusesUpdateChoreo.addInput("StatusUpdate", tweetText);

      // Requiere la ejecución del proceso de envío
      unsigned int returnCode = StatusesUpdateChoreo.run();

      // El código de retorno (0) significa proceso llegado a buen puerto
      if (returnCode == 0) {
        numRuns = numRuns + 1;
        digitalWrite(led_verde, LOW);
      } else {
        // En caso de error no se sale del loop while y se reintenta el envío
        while (StatusesUpdateChoreo.available()) {
          char c = StatusesUpdateChoreo.read();
        }
        digitalWrite(led_rosso, HIGH);
        delay(5000);
        digitalWrite(led_rosso, LOW);
        delay(5000);
      }
      StatusesUpdateChoreo.close();
    }
    numRuns = 0; // Todo ok en espera de una nueva pulsación del botón
  }
}
```

final, nuestro Tweet llegará a su destino (Fig. 24).

## CONCLUSIONES

En este primer acercamiento a la tarjeta Arduino Yún hemos tocado solo superficialmente su potencial, sobretodo en el plano didáctico. Yún se presta de hecho como banco de prueba, simple y poco costoso para profundizar los temas relativos al mundo GNU/Linux y a las redes, además de las propias del mundo Arduino, como ya era posible con sus predecesores. El sistema operativo Linino, con su corazón OpenWRT, se presta a experimentar, a bajo coste, la configuración y gestión de redes desde el punto de vista de la administración y de la seguridad, de manera cableada o wireless. Con un puñado de tarjetas Yún se pueden realizar configuraciones de red complejas; y esto, sobre todo en los tiempos de dificultades económicas en los que vivimos, debería interesar decididamente a las Escuelas Superiores y la Universidad. La misma cosa se puede decir para el estudio de aplicaciones embebidas, típicamente compuestas por secciones dirigidas al muestreo y elaboración de datos desde y hacia sensores y E/S analógicas y digitales y de secciones de coordinación e interfaz humano haciendo uso de soluciones de aplicación basadas en web. Todo ello llevado a un nivel simplificado, propio de la filosofía "Arduino", que permite también y sobre todo, a quien es principiante encontrar una satisfacción inmediata y un empujón para profundizar en las tecnologías que cada vez estarán más presentes en nuestro entorno. No dejaremos de profundizar las posibilidades de Yún y de proponérselas en próximos artículos, junto a las demás novedades que están surgiendo. (180043) ■

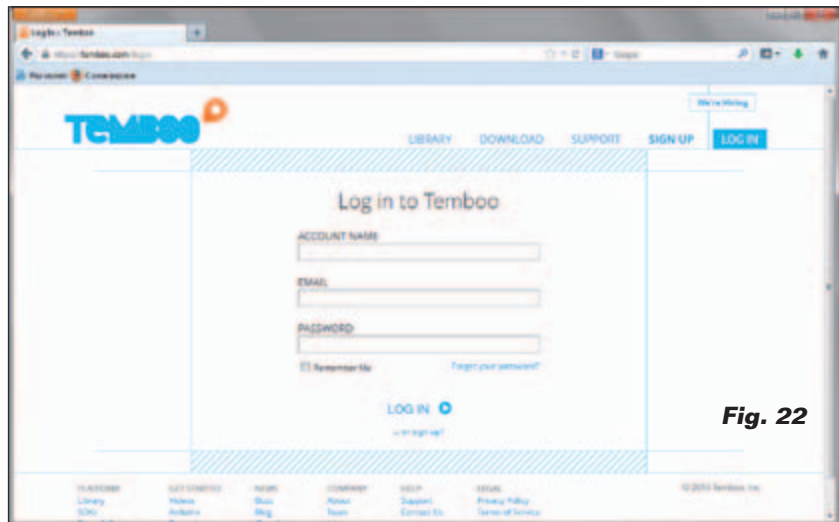


Fig. 22

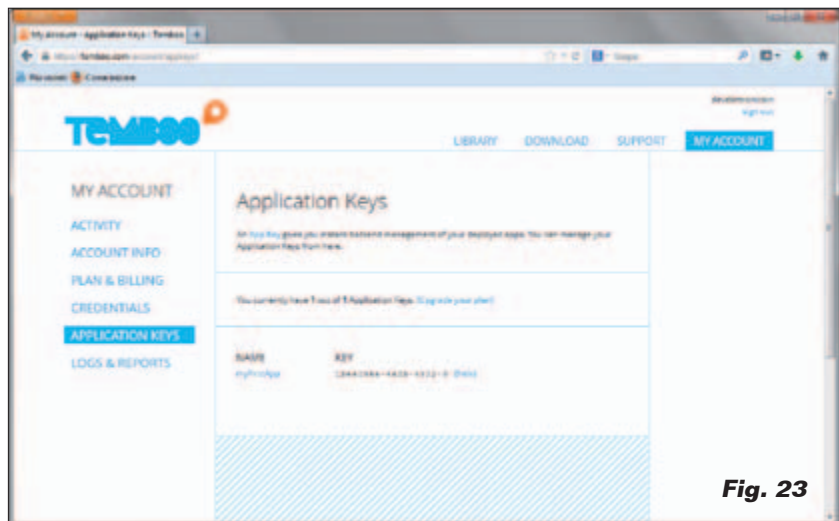


Fig. 23

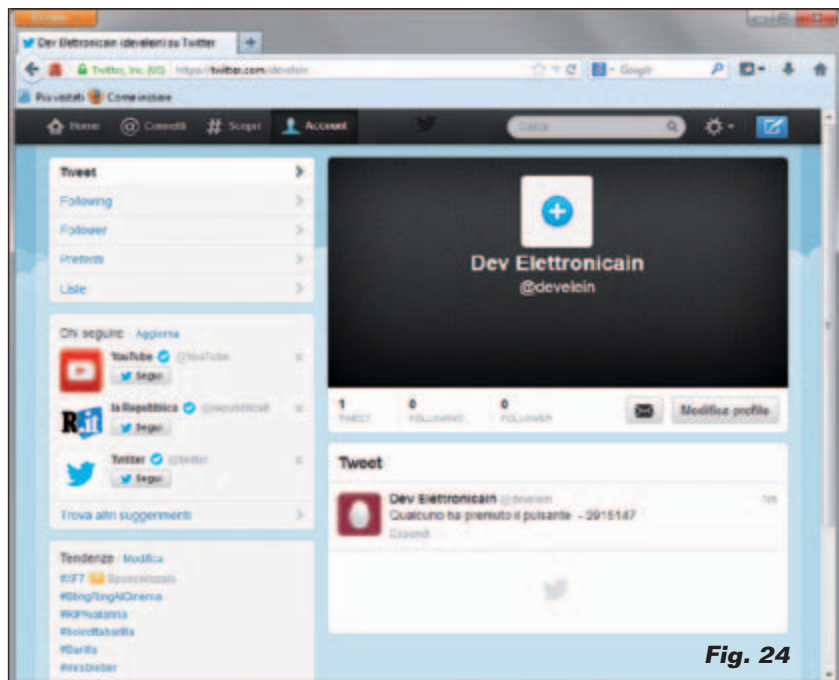


Fig. 24