

In omaggio  
il nuovo numero online di  
**MICROCHIP**

mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

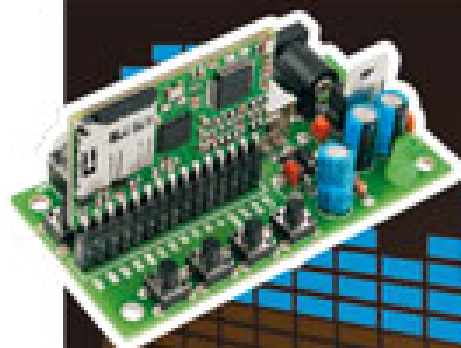
# Elettronica In

www.elettronicaonline.it

oltre l'elettronica

€ 6,00

Anno XX - n. 189  
Ottobre 2014



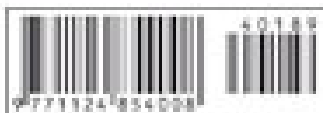
## Modulo mp3 Embedded

- **dynDNS:**  
localizza l'IP dinamico
- **Arduino MRF24J40**  
wireless shield
- **Teensy 3.1, microcontrollore**  
tuttofare compatibile con Arduino

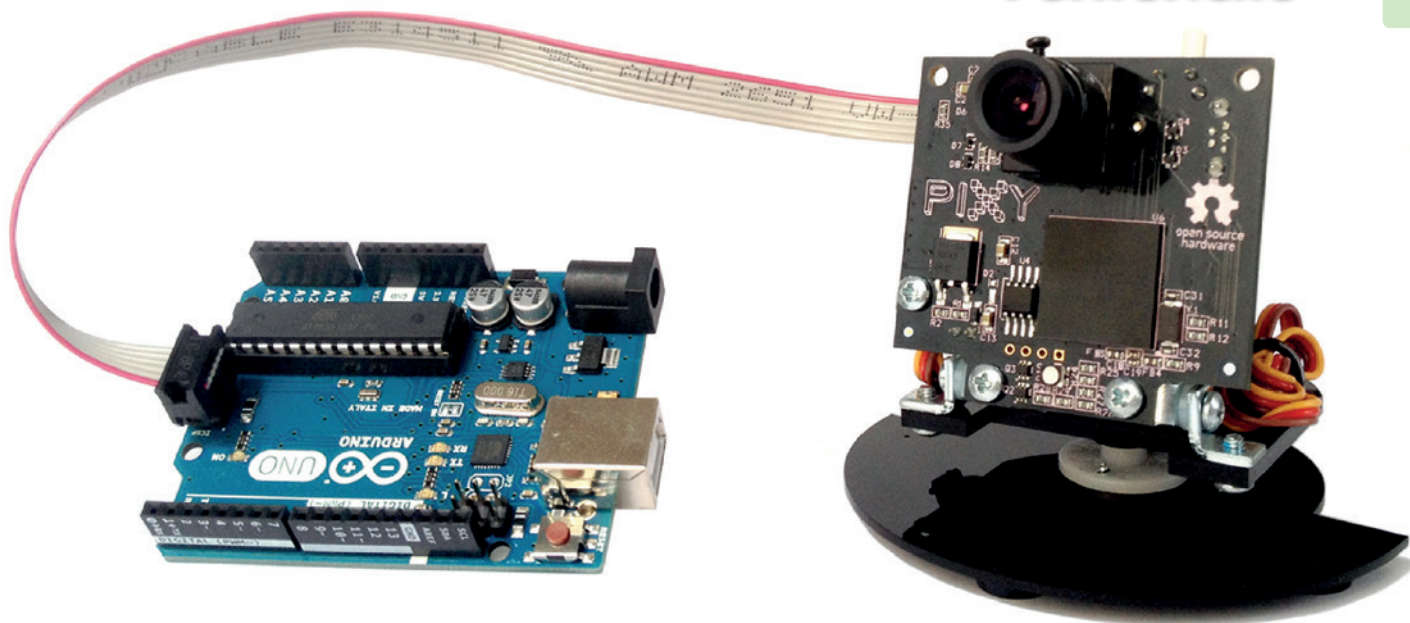
# 3D PRAG

## stampa anche il cioccolato

- **Pixy Camera**
- **myKIT, generatore**  
di onde triangolari
- **Arietta G25,**  
l'ultimo pinguino  
nato da Acme Systems







# PIXY: LA CAMERA CHE DISTINGUE

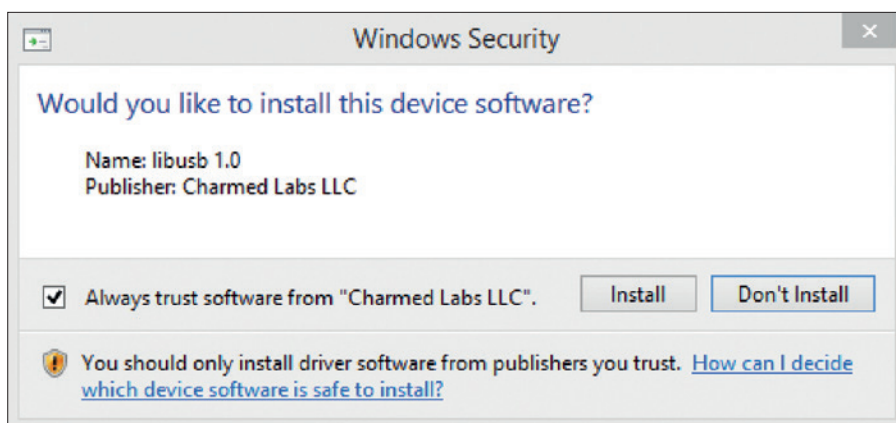


Primo incontro  
con la telecamera  
capace di  
riconoscere il  
colore degli  
oggetti e  
localizzarne la  
posizione.

..... di ALAN PIPITONE

**F**ino a poco tempo fa era davvero difficile sviluppare applicazioni in grado di sfruttare algoritmi di visione artificiale, soprattutto utilizzando sistemi con limitate risorse hardware. Fortunatamente, però, le cose sono cambiate: oltre ai vari framework open source, che aiutano gli sviluppatori ad implementare facilmente anche gli algoritmi più complessi, si stanno diffondendo

periferiche complete sia della parte hardware, che si occupa di acquisire le immagini provenienti dal mondo esterno, sia della parte software, capace di analizzare ed interpretare tali immagini. Con tali periferiche è possibile creare applicazioni dotate di visione artificiale in maniera davvero semplice. Basta utilizzare questi dispositivi in coppia con un microcontrollore o con una scheda di prototipazio-



**Fig. 1** - Installazione del software PixyMon.

ne, come ad esempio Arduino. Queste periferiche si occupano di tutta la parte di calcolo, lasciando libere le risorse hardware del dispositivo cui le si collega, scambiando con esso solo le informazioni essenziali. Così chiunque, in maniera estremamente semplice, potrà realizzare applicazioni in grado di osservare ed analizzare l'ambiente circostante.

In questo articolo ci occuperemo di esaminare il funzionamento della telecamera Pixy, dispositivo open source in grado di localizzare la posizione degli oggetti e comunicarne le coordinate ad Arduino o ai microcontrollori.

## INSTALLARE I COMPONENTI NECESSARI

Per prima cosa dobbiamo possedere una telecamera Pixy; è possibile acquistarla presso Futura Elettronica all'indirizzo [www.futurashop.it](http://www.futurashop.it) (il codice del prodotto è 7300- PIXYCAM). In questo articolo utilizzeremo anche il modulo pan/tilt, accessorio che permette alla telecamera Pixy di seguire gli spostamenti degli oggetti; esso non è obbligatorio, in quanto potrebbe essere autocostruito, tuttavia consigliamo di acquistarlo già pronto. Anche il modulo pan/

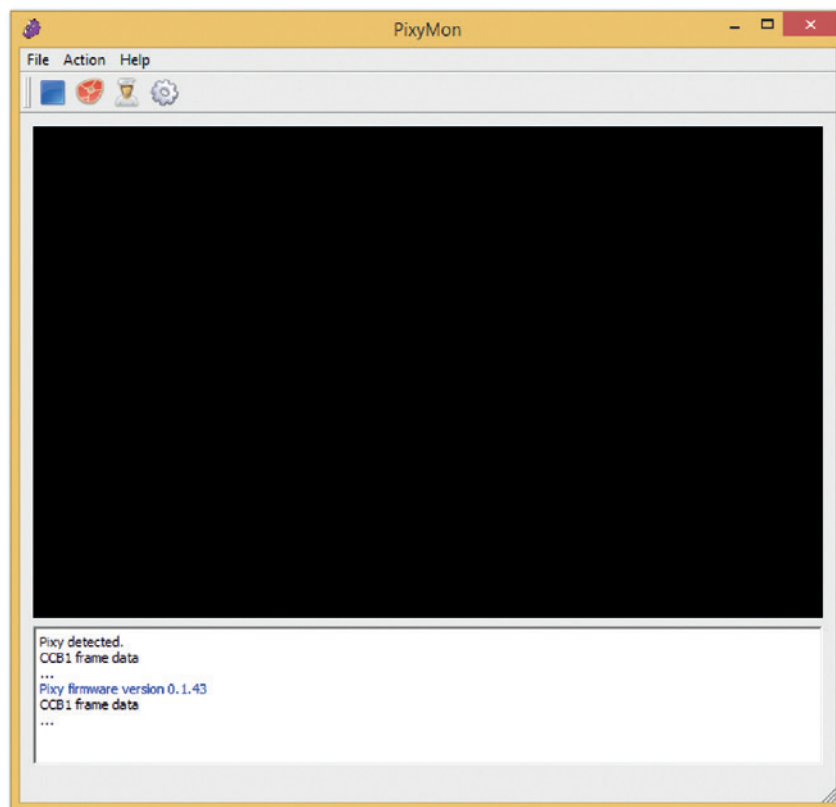
tilt è acquistabile presso Futura Elettronica.

Una volta recuperato tutto il materiale hardware è possibile scaricare il software di gestione, necessario sia ad impostare facilmente gli oggetti che la telecamera Pixy deve riconoscere, sia ad effettuare il debug della teleca-

mera stessa. Il software si chiama PixyMon ed è possibile scaricarlo dalla pagina web <http://cmucam.org/projects/cmucam5/files>. Se possedete un MAC dovreste scaricare il file *PixyMon\_mac-0.1.49.dmg*, aprirlo e trascinare l'applicazione PixyMon nella cartella Applications. Se invece possedete un PC Windows, dovreste scaricare il file *pixymon-windows-0.1.49.exe*, aprire l'eseguibile, iniziare la procedura di installazione ed infine attendere che compaia la schermata riportata in **Fig. 1**.

Una volta comparsa tale schermata, dovreste proseguire l'installazione cliccando sul pulsante "Install".

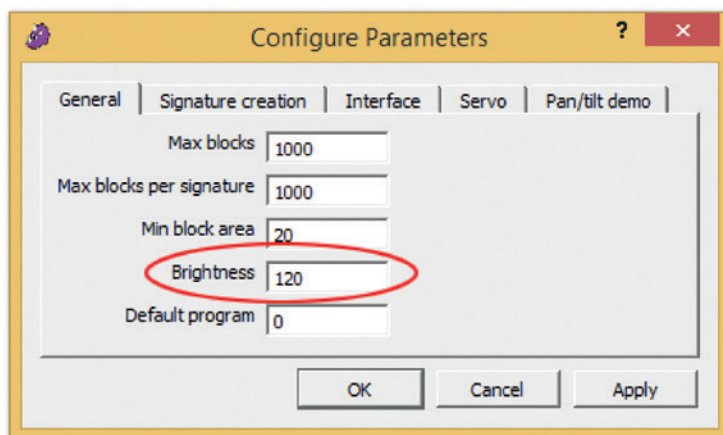
Se possedete un PC con sistema operativo Linux dovreste invece scaricare i sorgenti e compilarli; l'intera procedura è illustrata nella guida ufficiale disponibile



**Fig. 2** - Schermata principale del programma PixyMon.



**Fig. 3**  
Configurazione  
della luminosità.



su [http://cmucam.org/projects/cmucam5/wiki/Installing\\_PixyMon\\_on\\_Linux](http://cmucam.org/projects/cmucam5/wiki/Installing_PixyMon_on_Linux).

#### **PIXYMON: IL SOFTWARE DI GESTIONE**

Terminata la procedura di installazione del software, possiamo collegare la telecamera Pixy al nostro PC. Sulla parte posteriore di Pixy è presente un connettore di tipo miniUSB; dotiamoci quindi del cavo opportuno e colleghiamo la telecamera al nostro computer.

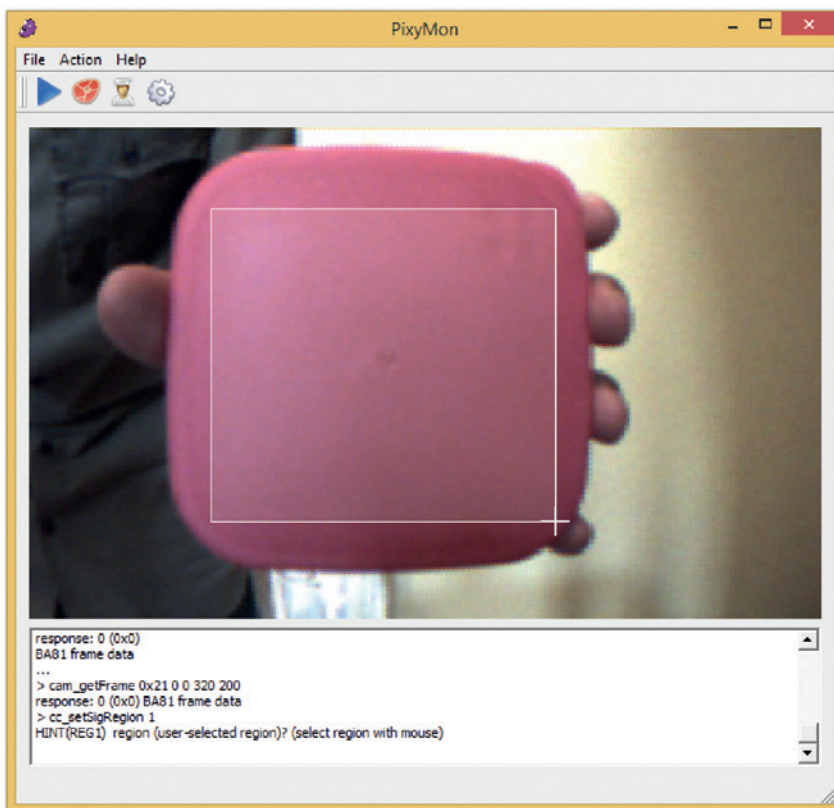
Se nel computer è installato un sistema operativo Windows, dobbiamo aspettare che esso rilevi la connessione della telecamera; una volta rilevato, il dispositivo verrà installato in maniera automatica. Con un MAC, invece, basterà semplicemente collegare la telecamera alla porta USB, senza dover attendere alcun tempo. Terminata la procedura di collegamento è possibile aprire il programma PixyMon, del quale in Fig. 2 è visibile la schermata principale.

Per far sì che l'area nera venga sostituita dalle immagini del flusso video, basterà cliccare sull'icona a forma di bistecca, posta in alto a sinistra. Se le immagini risultano troppo scure

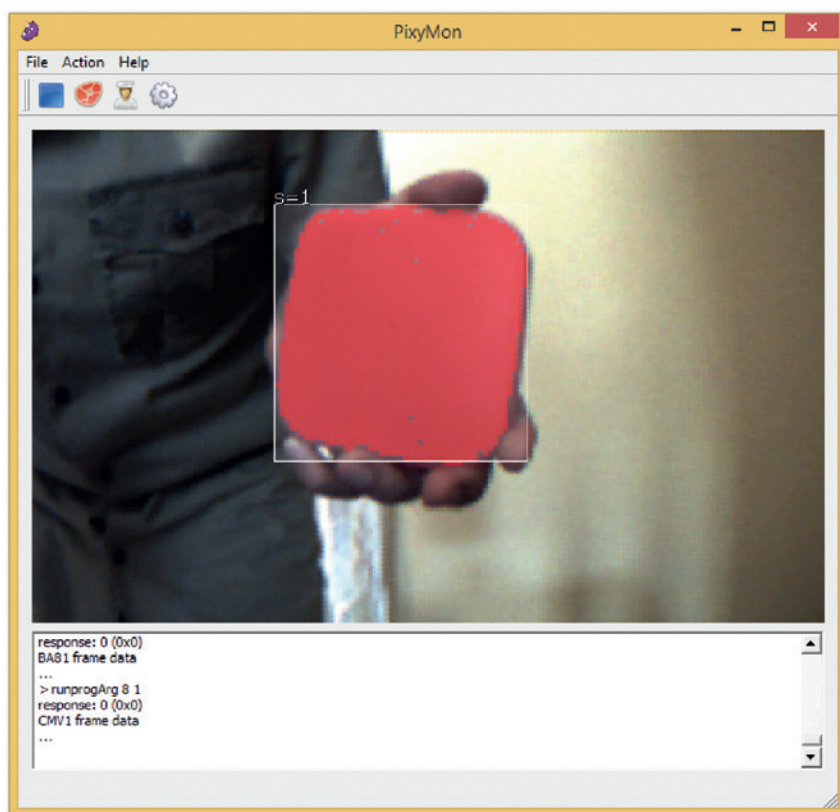
è possibile modificare la luminosità di Pixy. Dal menu *Configure Parameters*, accessibile cliccando sull'icona a forma di ingranaggio, possiamo modificare il parametro *Brightness*, come visibile in Fig. 3. Aumentando questo parametro le immagini risulteranno più chiare; viceversa saranno più

scure. Pixy è in grado di aumentare o diminuire la luminosità in maniera automatica, tuttavia se l'oggetto da riconoscere è troppo scuro o troppo chiaro bisognerà aggiustare manualmente il valore della luminosità.

A questo punto possiamo far apprendere a Pixy l'oggetto da riconoscere; il dispositivo riconoscerà l'oggetto in base al suo colore, quindi è importante che esso sia il più omogeneo possibile. Fatta questa premessa, possiamo iniziare la fase di apprendimento. Dobbiamo porre l'oggetto da riconoscere davanti all'obiettivo della telecamera, dopodiché, dal menu *Action*, dobbiamo selezionare "Set Signature 1...". Una volta fatto ciò, tenendo premuto il tasto sinistro del mouse, dobbiamo evidenziare il corpo dell'oggetto, come visibile in Fig. 4. Rilasciando il pulsante del



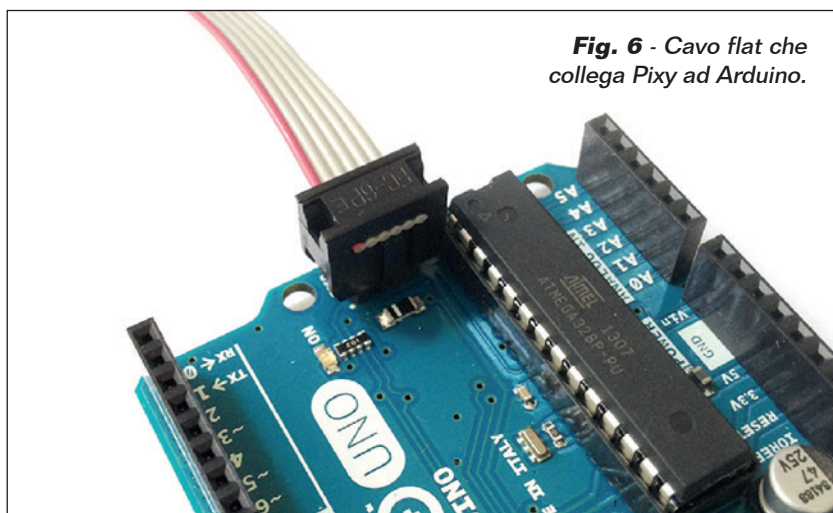
**Fig. 4** - Fase di apprendimento del colore di un oggetto.



**Fig. 5** - Riconoscimento dell'oggetto.

mouse, la telecamera memorizzerà il colore dell'oggetto, come visibile in **Fig. 5**. È anche possibile memorizzare un oggetto senza utilizzare PixyMon: premendo il pulsante posto sulla parte superiore

della telecamera, rispettando un'opportuna sequenza, è infatti possibile memorizzare l'oggetto posto davanti l'obiettivo. Questo metodo però è più complesso e lo vedremo quando collegheremo la telecamera ad Arduino.



**Fig. 6** - Cavo flat che collega Pixy ad Arduino.

Indipendentemente dalla modalità utilizzata, Pixy è in grado di memorizzare fino a sette oggetti differenti.

La telecamera li manterrà in memoria finché essi non verranno eliminati o sovrascritti, perciò scollegare Pixy dal PC o dalla sua sorgente di alimentazione non comporta l'eliminazione degli oggetti appresi.

Il software di gestione PixyMon ci consente anche di testare il sistema pan/tilt; per fare ciò, bisogna memorizzare almeno un oggetto, dopodiché, dal menu *Action*, basterà cliccare su "Run pan/tilt demo". Vedremo quindi i servo del modulo pan/tilt muovere la telecamera al fine di inquadrare l'oggetto memorizzato in precedenza.

È possibile che durante la demo del sistema pan/tilt la telecamera venga disconnessa dal PC diverse volte. Questo accade se si possiede un cavo USB di scarsa qualità oppure se il PC in nostro possesso ha una gestione delle porte USB non proprio ottimale.

## COLLEGARE ARDUINO

Connettere Pixy ad Arduino è abbastanza semplice: basta prendere il cavo flat contenuto nella confezione della telecamera, collegare il connettore più grosso alla parte posteriore di Pixy (il connettore sul retro impedisce di sbagliare verso) ed infine inserire il connettore più piccolo nella connessione ICSP di Arduino. Fate attenzione all'orientamento del cavo rosso, visibile in **Fig. 6**. A questo punto possiamo collegare Arduino al nostro PC e caricare i nostri primi sketch. È possibile scaricare le librerie e gli sketch di esempio dal link [http://cmucam.org/attachments/download/1065/arduino\\_pixy-0.1.5.zip](http://cmucam.org/attachments/download/1065/arduino_pixy-0.1.5.zip). Per importare le librerie è necessario aprire l'IDE di Arduino,

## Listato 1

```
//
// begin license header
//
// This file is part of Pixy CMUcam5 or "Pixy" for short
//
// All Pixy source code is provided under the terms of the
// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).
// Those wishing to use Pixy source code, software and/or
// technologies under different licensing terms should contact us at
// cmucam@cs.cmu.edu. Such licensing terms are available for
// all portions of the Pixy codebase presented here.
//
// end license header
//

/*
 06.04.2014 v0.1.3 John Leimon
 + Now using pixy.init() to initialize Pixy in setup().
 */

#include <SPI.h>
#include <Pixy.h>

Pixy pixy;

void setup()
{
    Serial.begin(9600);
    Serial.print("Starting...\n");

    pixy.init();
}

void loop()
{
    static int i = 0;
    int j;
    uint16_t blocks;
    char buf[32];

    blocks = pixy.getBlocks();

    if (blocks)
    {
        i++;

        if (i%50==0)
        {
            sprintf(buf, "Detected %d:\n", blocks);
            Serial.print(buf);
            for (j=0; j<blocks; j++)
            {
                sprintf(buf, " block %d: ", j);
                Serial.print(buf);
                pixy.blocks[j].print();
            }
        }
    }
}
```

dopodiché, dal menu *Sketch*, impartire il comando "Import Library...", poi "Add Library..." ed infine selezionare la cartella in cui sono contenuti i file dello zip appena scaricato. Questa operazione caricherà

anche gli sketch di esempio. Per vederne il codice è sufficiente cliccare sul menu *File*, poi su *Examples* ed infine sul sottomenu *Pixy*.

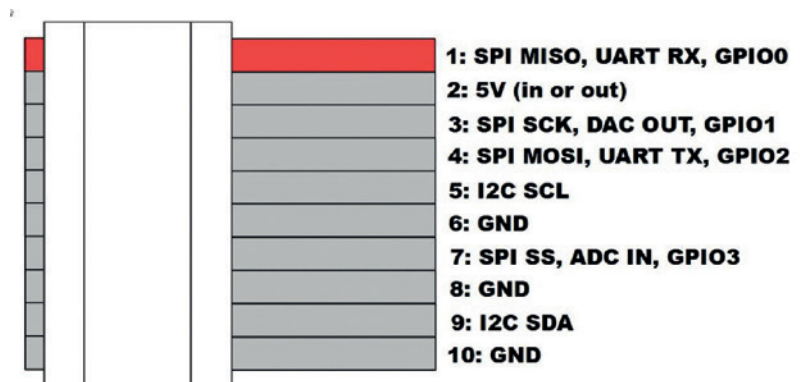
### PRIMO SKETCH: HELLO WORLD

In totale gli sketch di esempio sono quattro. Il primo si chiama *hello\_world*, ed il sorgente è visibile nel **Listato 1**.

In esso, come prima cosa viene fatto l'import delle librerie necessarie ad instaurare la corretta comunicazione tra Arduino e la telecamera. Successivamente viene creato un oggetto di tipo *Pixy*, che poi è inizializzato nella funzione *setup*.

In questa funzione viene imposta anche la comunicazione seriale con il PC, così potremo vedere i vari messaggi di debug. Il metodo **getBlocks()**, richiamato nel corpo della funzione *loop*, serve a contare quanti oggetti memorizzati si trovano davanti all'obiettivo della telecamera. Il metodo ritorna un numero intero, il cui valore è dato dal numero totale di oggetti riconosciuti. Per ogni oggetto identificato è possibile recuperare una serie di informazioni, come ad esempio le sue coordinate e le dimensioni in pixel dell'immagine memorizzata relativa ad esso.

**Fig. 7 - Schema del connettore flat di Pixy.**



## Listato 2

```
//
// begin license header
//
// This file is part of Pixy CMUcam5 or "Pixy" for short
//
// All Pixy source code is provided under the terms of the
// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).
// Those wishing to use Pixy source code, software and/or
// technologies under different licensing terms should contact us at
// cmucam@cs.cmu.edu. Such licensing terms are available for
// all portions of the Pixy codebase presented here.
//
// end license header
//

#include <SPI.h>
#include <Pixy.h>

#define X_CENTER 160L
#define Y_CENTER 100L
#define RCS_MIN_POS 0L
#define RCS_MAX_POS 1000L
#define RCS_CENTER_POS ((RCS_MAX_POS-RCS_MIN_POS)/2)

class ServoLoop
{
public:
    ServoLoop(int32_t pgain, int32_t dgain);

    void update(int32_t error);

    int32_t m_pos;
    int32_t m_prevError;
    int32_t m_pgain;
    int32_t m_dgain;
};

ServoLoop panLoop(500, 800);
ServoLoop tiltLoop(700, 900);

ServoLoop::ServoLoop(int32_t pgain, int32_t dgain)
{
    m_pos = RCS_CENTER_POS;
    m_pgain = pgain;
    m_dgain = dgain;
    m_prevError = 0x80000000L;
}

void ServoLoop::update(int32_t error)
{
    long int vel;
    char buf[32];
    if (m_prevError!=0x80000000)
    {
        vel = (error*m_pgain + (error - m_prevError)*m_dgain)>>10;
        //sprintf(buf, "%ld\n", vel);
        //Serial.print(buf);
        m_pos += vel;
        if (m_pos>RCS_MAX_POS)
            m_pos = RCS_MAX_POS;
        else if (m_pos<RCS_MIN_POS)
            m_pos = RCS_MIN_POS;

        //cprintf("%d %d %d\n", m_axis, m_pos, vel);
    }
    m_prevError = error;
}

Pixy pixy;

void setup()
```

(Continua)

Il ciclo "for" successivo serve proprio a questo.

Ogni oggetto individuato viene automaticamente salvato nell'array blocks dell'oggetto pixy; l'istruzione **pixy.blocks[j].print()** ci consente dunque di accedere all'oggetto identificato e stamparne le informazioni corrispondenti.

Oltre al metodo "print" ogni oggetto ha anche i seguenti attributi: **x**, **y**, **width**, **height** e **signature**. Quest'ultimo contiene il numero dell'oggetto identificato, che varia da 1 a 7 in base all'ordine con cui l'oggetto è stato memorizzato nella telecamera.

### SECONDO SKETCH: PAN/TILT

Lo sketch chiamato "pantilt" consente di comandare il modulo pan/tilt direttamente da Arduino.

Il sorgente dello sketch è visibile nel **Listato 2**. Dopo l'import delle librerie e la definizione di alcune costanti, troviamo il codice della classe **ServoLoop**. Per capire lo scopo di tale classe dobbiamo analizzare il codice del metodo **update**.

Questo metodo serve a mantenere l'oggetto individuato sempre al centro dell'obiettivo. Nel codice della funzione possiamo notare un algoritmo di calcolo simile a quello di un sistema PID. In particolare viene utilizzato l'argomento "error" per calcolare il valore da assegnare al servo, valore che viene poi salvato nella variabile **m\_pos**. L'argomento **error** contiene la differenza tra la posizione dell'oggetto individuato ed il centro della telecamera. Nel corpo della funzione "loop" possiamo vedere che tale differenza viene calcolata in due parti, una delle quali è contenuta nella variabile **panError**, mentre l'altra è salvata nella variabile **tiltError**. Questi valori vengono poi



utilizzati da due istanze della classe ServoLoop, **panLoop** e **tiltLoop**, che tramite il proprio metodo **update** si occupano di aggiornare il valore da assegnare ai due servo. Successivamente la funzione **pixy.setServos** imposta fisicamente la posizione dei due servo.

Questo sketch funziona alla perfezione solamente se si è in possesso del sistema pan/tilt ufficiale, cioè quello acquistabile presso Futura Elettronica.

Se utilizzate altri tipi di servo, dovete fare qualche prova pratica per verificarne la compatibilità e molto probabilmente cambiare alcuni parametri, come ad esempio il valore dei vari guadagni del sistema PID o la posizione massima che i servo possono assumere.

#### GLI ULTIMI SKETCH

Negli sketch che abbiamo appena visto, la comunicazione tra Arduino e Pixy avviene tramite il bus SPI. Tuttavia è possibile utilizzare anche bus differenti, in particolare I<sup>2</sup>C e l'UART: gli ul-

#### Listato 2 (segue)

```
{
  Serial.begin(9600);
  pixy.init();
}

void loop()
{
  static int i = 0;
  int j;
  uint16_t blocks;
  char buf[32];
  int32_t panError, tiltError;

  blocks = pixy.getBlocks();

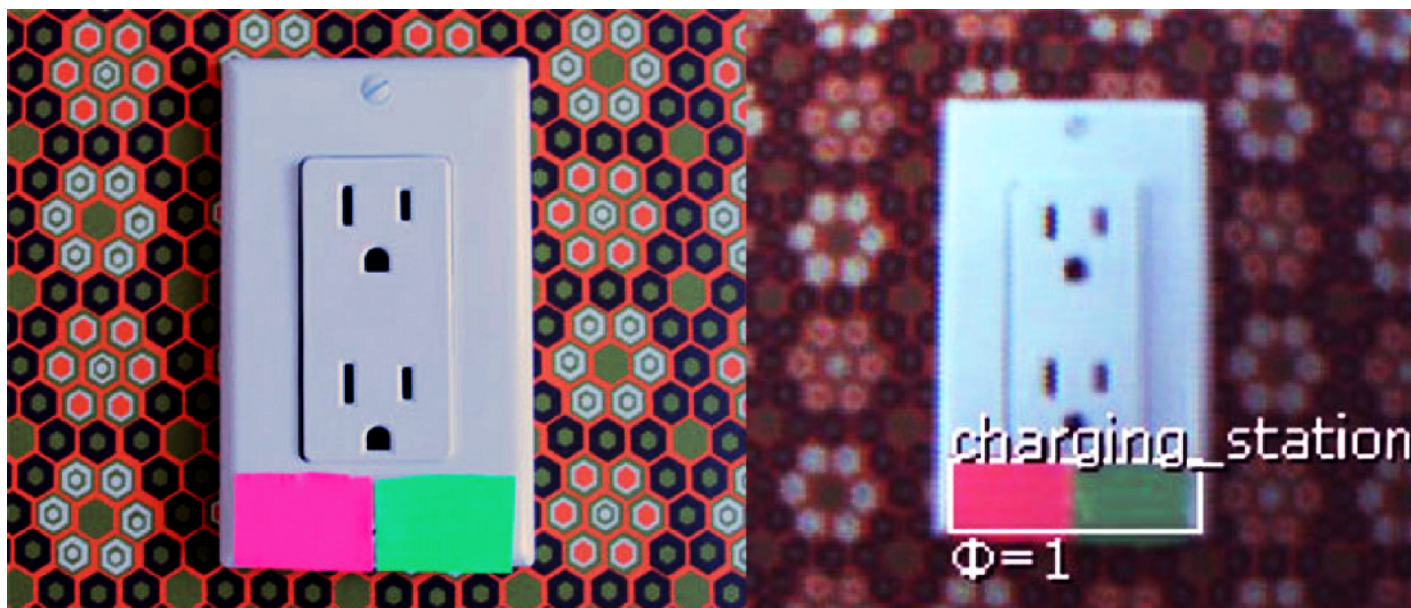
  if (blocks)
  {
    panError = X_CENTER-pixy.blocks[0].x;
    tiltError = pixy.blocks[0].y-Y_CENTER;

    panLoop.update(panError);
    tiltLoop.update(tiltError);

    pixy.setServos(panLoop.m_pos, tiltLoop.m_pos);

    i++;

    if (i%50==0)
    {
      sprintf(buf, "Detected %d:\n", blocks);
      Serial.print(buf);
      for (j=0; j<blocks; j++)
      {
        sprintf(buf, " block %d: ", j);
        Serial.print(buf);
        pixy.blocks[j].print();
      }
    }
  }
}
```



**Fig. 8** - Color Code formato da due colori.

## Listato 3

```
// begin license header
//
// This file is part of Pixy CMUcam5 or "Pixy" for short
//
// All Pixy source code is provided under the terms of the
// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).
// Those wishing to use Pixy source code, software and/or
// technologies under different licensing terms should contact us at
// cmucam@cs.cmu.edu. Such licensing terms are available for
// all portions of the Pixy codebase presented here.
//
// end license header
//

#include <Wire.h>
#include <PixyI2C.h>

PixyI2C pixy;

void setup()
{
  Serial.begin(9600);
  Serial.print("Starting...\n");

  pixy.init();
}

void loop()
{
  static int i = 0;
  int j;
  uint16_t blocks;
  char buf[32];

  blocks = pixy.getBlocks();

  if (blocks)
  {
    i++;

    if (i%50==0)
    {
      sprintf(buf, "Detected %d:\n", blocks);
      Serial.print(buf);
      for (j=0; j<blocks; j++)
      {
        sprintf(buf, " block %d: ", j);
        Serial.print(buf);
        pixy.blocks[j].print();
      }
    }
  }
}
```

timi due sketch di esempio mostrano proprio come utilizzare questi. Però, prima di utilizzare i due bus, dobbiamo modificare il collegamento tra arduino e Pixy, perché il cavo flat in dotazione con la telecamera non collega tutti i pin di Pixy, ma solo quelli necessari per la comunicazione via SPI e pochi altri. In Fig. 7

possiamo osservare lo schema del connettore flat posto sul retro della telecamera. Per utilizzare il bus I<sup>2</sup>C bisogna collegare il pin SDA di Arduino al pin 9 di Pixy, il pin SCL al pin 5 di Pixy ed infine il pin GND di Arduino al pin 6, 8 o 10 della telecamera. Il dispositivo ha a bordo le due resistenze di pull-up

necessarie al corretto funzionamento del bus I<sup>2</sup>Cc, quindi non dobbiamo preoccuparcene. Per quanto riguarda la connessione UART, è sufficiente collegare il pin TX di Arduino al pin 1 di Pixy, il pin RX al pin 4 di Pixy ed infine il pin GND di Arduino al pin 6, 8 o 10 della telecamera. Terminata la descrizione del collegamento tra arduino e Pixy, possiamo analizzare i due sketch rimanenti. Nel **Listato 3** è visibile il codice sorgente dello sketch chiamato i2c, mentre nel **Listato 4** è visibile il codice dello sketch chiamato uart. Il sorgente di entrambi i listati è molto simile a quello del **Listato 1**. Quello che cambia è l'import delle librerie, che variano a seconda del bus utilizzato, e la definizione dell'oggetto Pixy, il quale nel **Listato 3** viene definito con l'istruzione **PixyI2C pixy**, mentre nel **Listato 4** viene definito tramite **PixyUART pixy**.

## APPENDIMENTO

A questo punto possiamo vedere come apprendere nuovi oggetti senza l'ausilio di un PC. Quando la telecamera viene alimentata, il LED RGB montato sulla parte frontale inizia a lampeggiare: appena il LED si spegne bisogna premere per almeno un secondo il pulsante posto sulla parte superiore di Pixy. Il LED a questo punto si accenderà nuovamente ed assumerà colore bianco, poi diventerà rosso e successivamente cambierà in altri colori. Quando diventerà rosso dovremo rilasciare il pulsante. A questo punto il LED assumerà il colore del pixel centrale dell'immagine che l'obiettivo sta inquadrando. Il diodo RGB può essere utilizzato come feedback per capire se l'oggetto da apprendere è al centro della telecamera o meno.



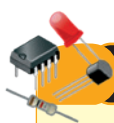
Quando il feedback ci soddisfa, possiamo salvare l'oggetto nella memoria di Pixy. Per farlo è sufficiente premere e rilasciare il pulsante della telecamera. Se Pixy determinerà che le caratteristiche dell'oggetto sono buone (in particolare la saturazione del colore) il LED farà un breve lampeggio e la procedura potrà essere considerata conclusa. Se invece l'oggetto non è idoneo ad essere salvato, allora il diodo verrà semplicemente spento e la procedura sarà da rifare.

### CONCLUSIONI

Come abbiamo avuto modo di osservare è possibile creare, in maniera abbastanza semplice, sistemi embedded in grado di riconoscere gli oggetti e localizzarne la posizione.

Tuttavia la forza di Pixy non si limita solo a quanto abbiamo visto fino ad ora.

Il nuovo firmware di Pixy e la nuova versione di PixyMon offrono altre funzioni molto interessanti. Un esempio è dato dalla possibilità di memorizzare i "Color Codes", cioè diverse



### per il MATERIALE

La telecamera Pixy (cod. PIXYCAM) è disponibile al prezzo di Euro 65,00. Il modulo Pan/Tilt per questa telecamera (cod. PIXYPANTILT) è anch'esso disponibile al prezzo di Euro 36,00. La scheda Arduino Rev.3 (cod. ARDUINOUNOREV3) è disponibile ad Euro 24,50. Tutti i prezzi sono da intendere IVA compresa.

Il materiale va richiesto a: Futura Elettronica, Via Adige 11, 21013 Gallarate (VA) Tel: 0331-799775 • Fax: 0331-792287 <http://www.futurashop.it>

## Listato 4

```
// begin license header
//
// This file is part of Pixy CMUcam5 or "Pixy" for short
//
// All Pixy source code is provided under the terms of the
// GNU General Public License v2 (http://www.gnu.org/licenses/gpl-2.0.html).
// Those wishing to use Pixy source code, software and/or
// technologies under different licensing terms should contact us at
// cmucam@cs.cmu.edu. Such licensing terms are available for
// all portions of the Pixy codebase presented here.
//
// end license header
//

#include "PixyUART.h"

PixyUART pixy;

void setup()
{
    Serial.begin(9600);
    Serial.print("Starting...\n");

    pixy.init();
}

void loop()
{
    static int i = 0;
    int j;
    uint16_t blocks;
    char buf[32];

    blocks = pixy.getBlocks();

    if (blocks)
    {
        i++;

        if (i%50==0)
        {
            sprintf(buf, "Detected %d:\n", blocks);
            Serial.print(buf);
            for (j=0; j<blocks; j++)
            {
                sprintf(buf, " block %d: ", j);
                Serial.print(buf);
                pixy.blocks[j].print();
            }
        }
    }
}
```

combinazioni di colori che vanno a formare un'unica signature. In questo modo è possibile creare diversi marker univoci, come visibile in Fig. 8.

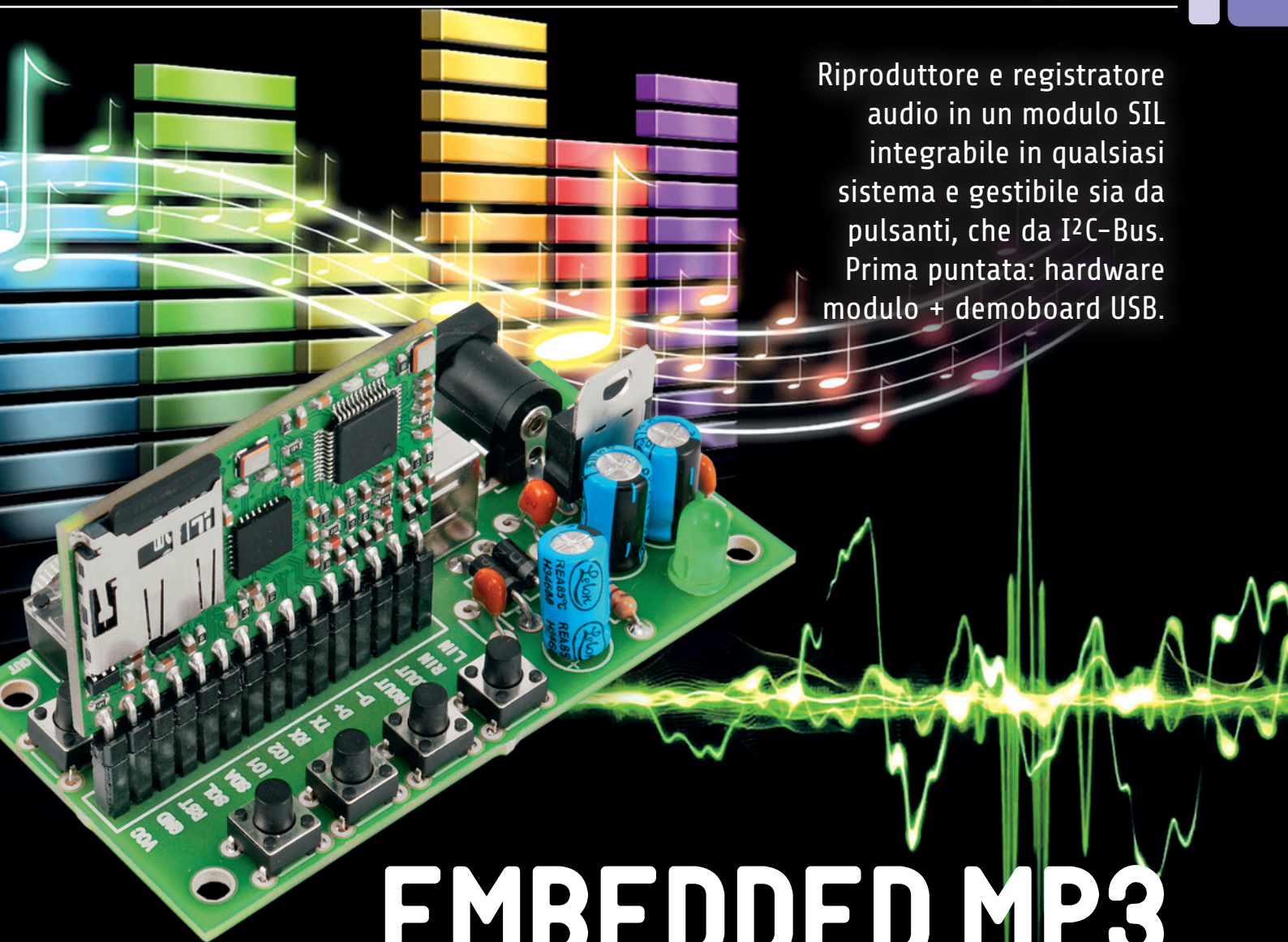
Ciò aumenta notevolmente le potenzialità offerte dal dispositivo. Non abbiamo trattato l'argomento in questo articolo poiché sia il nuovo firmware, sia la nuova

versione di Pixymon, sono ancora in fase beta.

Nei prossimi articoli affronteremo diversi progetti interessanti, andando ad approfondire le varie funzioni avanzate che la telecamera mette a disposizione. Non perdetevi quindi i prossimi appuntamenti dedicati a questo fantastico dispositivo.



Riproduttore e registratore audio in un modulo SIL integrabile in qualsiasi sistema e gestibile sia da pulsanti, che da I<sup>2</sup>C-Bus. Prima puntata: hardware modulo + demoboard USB.

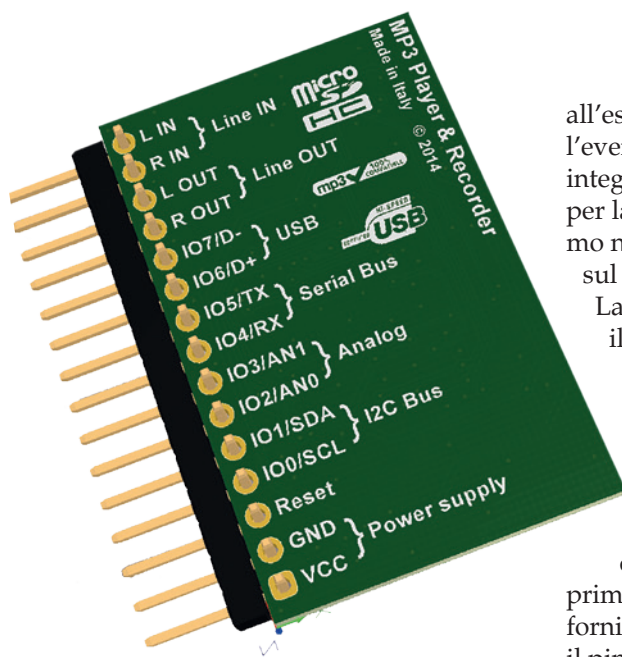


# EMBEDDED MP3 PLAYER & RECORDER

dell'ing LUCA PASCARELLA

**D**isporre di un modulo facilmente adattabile per riprodurre brani musicali è sempre comodo: permette di implementare facilmente sintesi vocali da inserire in promobox, oggetti nelle vetrine o nei punti di passaggio delle persone, in prossimità di porte automatiche o tornelli, oppure in sportelli automatici per varie pratiche. Se poi il modulo è in grado di riprodurre anche brani musicali di una certa durata e se supporta i file MP3, ormai divenuti uno standard per svariati dispositivi (come i lettori musicali tascabili),

abbiamo raggiunto il top. In quest'articolo ci siamo concentrati sulla realizzazione di un nuovo dispositivo elettronico che integra al suo interno sia un riproduttore che un registratore audio. Gli algoritmi di codifica o decodifica supportati sono: l'ormai diffusissimo formato MP3 ed il suo diretto concorrente royalty-free che è Ogg Vorbis. Il tutto è stato pensato in maniera tale da alloggiare ogni componente necessario al funzionamento su un circuito stampato di ridotte dimensioni (circa 25 x 40 mm). Quest'ultimo è dotato di un connettore



**Fig. 1** - La disposizione dei contatti è serigrafata sul retro del PCB del modulo.

(Break Away Header) angolare a strip da 15 contatti con passo standard 2,54 mm, il che lo rende idoneo per l'inserimento in verticale in una breadboard e in generale in qualsiasi circuito stampato. Il montaggio può essere sia a saldare, sia a innesto in un connettore femmina SIL connettore che ne consente l'inserimento e la rimozione dal circuito ospitante. Il circuito al suo interno prevede uno slot *Push-Pull* per microSD che funge da memoria di massa dedicata all'archiviazione dei flussi audio registrati o da supporto alla riproduzione dei brani musicali. Inoltre, il circuito al suo interno ha tutti i componenti necessari a funzionare, pertanto è sufficiente fornire una tensione di alimentazione e scegliere la modalità di interazione per poter in poco tempo ascoltare il proprio brano preferito oppure avviare una registrazione a breve o lungo termine.

## LA PIEDINATURA

Tutte le connessioni da portare

all'esterno per l'interazione con l'eventuale circuito in cui volete integrare il modulo (ma anche per la demoboard di cui parleremo nelle prossime pagine) sono sul connettore single-in-line.

La disposizione delle linee è illustrata dall'apposita serigrafia prevista sulla parte posteriore del circuito (**Fig. 1**); i dettagli sono invece descritti nella **Tabella 1**. Proseguendo per ordine, analizziamo di seguito ognuno di essi. I primi due pin servono per poter fornire alimentazione al circuito: il pin 1 corrisponde all'alimentazione positiva, che può essere 5 V oppure 3,3 V; in ogni caso i GPIO rimarranno sempre a 3,3 V (non sono 5 V tolerant) pertanto se il modulo dovrà essere interfacciato con un dispositivo con I/O a 5 V

sarà necessario adattare i segnali in ingresso al riproduttore al fine di evitare un *overvoltage* sulle linee corrispondenti.

Il pin 2 è il riferimento di massa sia per l'alimentazione che per i segnali analogici di *Line-In* e *Line-Out*. Per evitare interferenze da parte dei segnali digitali, nel circuito in cui monterete il modulo si consiglia di impiegare una linea di GND molto grossa in modo tale da diminuire il più possibile l'effetto delle cadute di tensione o gli effetti dovuti alla condivisione dello stesso riferimento tra GND digitale e GND analogico.

Il pin 3 interviene direttamente sul *reset* del microcontrollore U2: in condizioni di funzionamento normale è sufficiente lasciare non collegato questo ingresso. Per inviare un impulso di *hard reset* è

**Tabella 1** - Descrizione del connettore angolare da 15 contatti a passo 2,54mm.

| #  | Nome      | Descrizione primaria                                | Descrizione alternativa      |
|----|-----------|-----------------------------------------------------|------------------------------|
| 1  | Vdd       | Alimentazione, 5 V o 3,3 V. Il GPIO è sempre 3,3 V. |                              |
| 2  | Gnd       | Riferimento di massa                                |                              |
| 3  | MCLR      | Hard reset                                          |                              |
| 4  | SCL/I00   | I <sup>2</sup> C clock                              | General purpose I/O numero 0 |
| 5  | SDA/I01   | I <sup>2</sup> C data                               | General purpose I/O numero 1 |
| 6  | AN0/I02   | Input analogico numero 0                            | General purpose I/O numero 2 |
| 7  | AN1/I03   | Input analogico numero 1                            | General purpose I/O numero 3 |
| 8  | RX/I04    | UART nativa ricevitore                              | General purpose I/O numero 4 |
| 9  | TX/I05    | UART nativa trasmettitore                           | General purpose I/O numero 5 |
| 10 | USB_P/I06 | USB Data +                                          | General purpose I/O numero 6 |
| 11 | USN_N/I07 | USB Data -                                          | General purpose I/O numero 7 |
| 12 | Rout      | Line-Out, canale destro in uscita                   |                              |
| 13 | Lout      | Line-Out, canale sinistro in uscita                 |                              |
| 14 | Rin       | Line-In, canale destro in ingresso                  |                              |
| 15 | Lin       | Line-In, canale sinistro in ingresso                |                              |

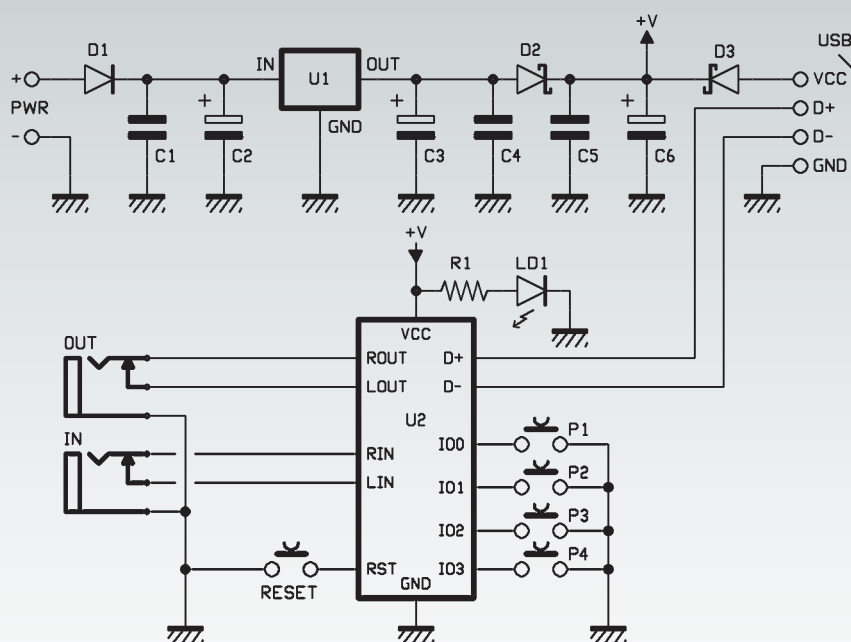
**Tabella 2** - Descrizione delle funzionalità associate al file "conf.ini" di esempio per l'impiego dei pulsanti presenti sulla demoboard.

| Ingresso | Pulsante | Comando | Funzionalità                                                             |
|----------|----------|---------|--------------------------------------------------------------------------|
| I00      | P1       | 1       | Avvia l'esecuzione dei file contenuti nella play list                    |
| I01      | P2       | 2       | Ferma l'esecuzione corrente                                              |
| I02      | P3       | 4       | Attiva o disattiva la pausa                                              |
| I03      | P4       | 5       | Avvia la registrazione nominando il file secondo i parametri di conf.ini |





## [schema elettrico della DEMOBOARD]



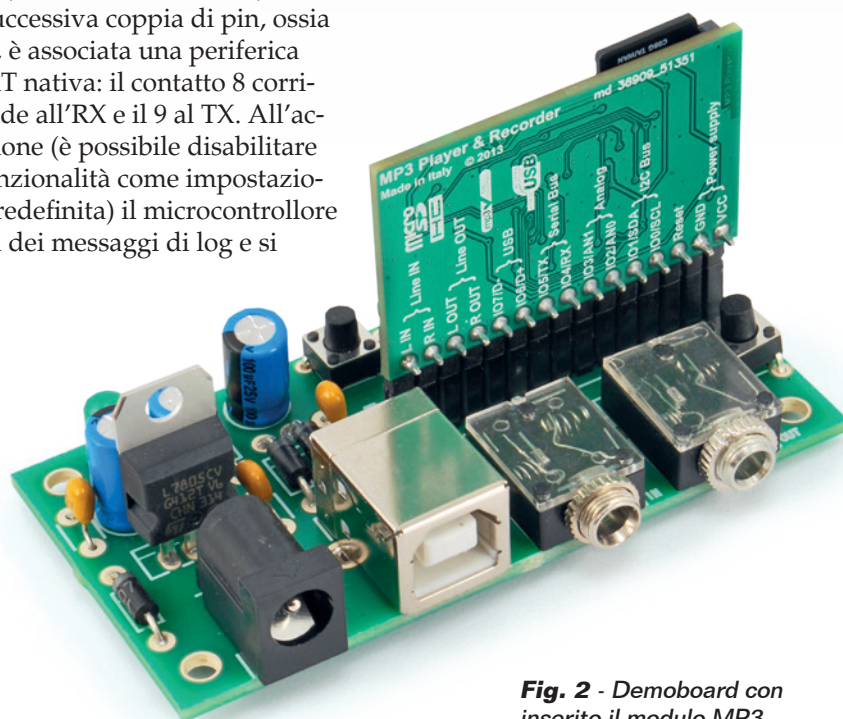
sufficiente forzare questa linea ad un livello logico basso per qualche millisecondo; al rilascio, la linea tornerà nuovamente a 3,3 V grazie al resistore R2, che provvede al pull-up.

I pin 4 e 5 sono la prima coppia di I/O programmabili a piacere dall'utente; su questi pin è possibile abilitare l'uso del bus I<sup>2</sup>C (il riproduttore funziona da slave) pertanto le funzioni dei pin diventano rispettivamente SCL ed SDA. Per il corretto funzionamento del bus è necessario inserire due resistori di pull-up, come da specifica del protocollo I<sup>2</sup>C, in generale è sufficiente un valore compreso tra 2,2 kohm e 10 kohm. Tramite protocollo I<sup>2</sup>C è possibile dialogare con il riproduttore leggendo e scrivendo i registri interni dedicati alla funzionalità I<sup>2</sup>C. Viceversa, la stessa coppia di pin può funzionare da I/O general purpose, identificati rispettivamente con GPIO0 e GPIO1; in questo caso è possibile associare ad ognuno di essi una funzione di input oppure una funzione di output (riferirsi alla Tabella 5).

I contatti numero 6 e 7 corrispondono alla funzione di ingresso analogico (non ancora implementata nel software v0.1) e fungono rispettivamente da AN0 ed AN1; in alternativa funzionano da seconda coppia di IO general purpose (sono GPIO2 e GPIO3). La successiva coppia di pin, ossia 8 e 9, è associata a una periferica UART nativa: il contatto 8 corrisponde all'RX e il 9 al TX. All'accensione (è possibile disabilitare la funzionalità come impostazione predefinita) il microcontrollore invia dei messaggi di log e si

pone in ascolto dei comandi impartiti dall'utente. Disabilitando la funzionalità seriale è possibile usare questi pin come terza coppia di I/O general purpose (GPIO4 e GPIO5).

Infine, l'ultima coppia di pin programmabili è quella formata dai contatti 10 e 11, ai quali è associata la funzionalità di D+ e D- del bus USB. Il software, quando il circuito viene collegato ad un computer, crea contemporaneamente una periferica *Mass Storage* (consentendo l'esplorazione del contenuto della microSD inserita nel modulo) ed una *Serial Port Emulator*, che può essere impiegata per dialogare con il dispositivo al posto della UART nativa. Se il driver di Windows USB Composite device non dovesse essere normalmente disponibile, tramite il sistema di recupero dei driver di Windows è possibile aggiungerlo manualmente; il driver è reperibile insieme ai sorgenti del progetto. Nel caso in cui la funzionalità USB non fosse desi-



**Fig. 2** - Demoboard con inserito il modulo MP3.

derata, è possibile configurare il software in modo tale da usare la funzione I/O general purpose e quindi i piedini 10 e 11 diventano rispettivamente GPIO6 e GPIO7. Nel paragrafo *IO general purpose* sarà spiegato come associare una funzionalità della **Tabella 5** ad uno degli otto pin di GPIO. Continuando con la descrizione delle funzionalità associate al connettore, troviamo i pin 12 e 13, i quali funzionano da *Line-Out*, ed i pin 14 e 15, che invece funzionano da *Line-In* compatibilmente con i livelli standard (*Consumer audio* a -10 dBV - livello nominale di 316 mV rms) per i segnali audio.

#### SCHEMA ELETTRICO

Il riproduttore e registratore MP3/ Ogg Vorbis è animato dal com-

ponente U2 (PIC32MX270F256B): questo è un nuovo microcontrollore *Microchip*, diretta evoluzione pin-to-pin compatibile del microcontrollore PIC32MX250F128B. Entrambi i dispositivi sono dotati di una CPU RISC a 32 bit capace di lavorare con una frequenza di clock di 40 MHz (50 MHz, se l'intervallo di temperatura di funzionamento è compreso tra i -40 °C ed i +85 °C) raggiungendo in condizioni ideali circa 80 DMIPS. Inoltre entrambi i dispositivi sono dotati di alcune implementazioni hardware di bus di comunicazione standard utili per il nostro progetto, tra cui ricordiamo: la *Serial Peripheral Interface* (SPI) usata per accedere alla microSD, l'*Inter Integrated Circuit* (I<sup>2</sup>C) per dialogare con l'esterno ed una periferica *Universal Serial Bus* (USB) Client

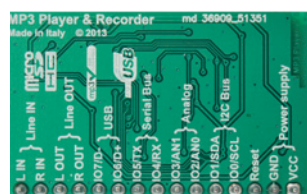
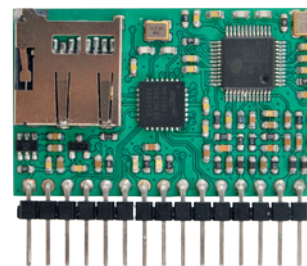
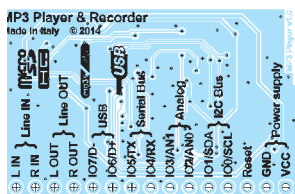
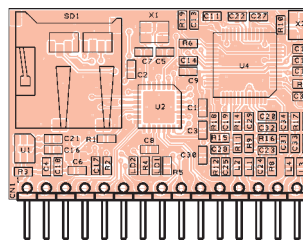
*Full Speed*. I due microcontrollori differiscono per il quantitativo di memoria disponibile: il secondo è dotato di 128 kB di memoria programma e 32 kB di RAM, mentre il PIC32MX270F256B è stato ideato per supportare il doppio sia di memoria di programma che di RAM. Siccome per volontà di *Microchip* i due microcontrollori sono pin-to-pin compatibili, nel modulo potete montare entrambi, tuttavia per ottenere tutte le funzionalità software senza limitazioni sarà necessario scegliere il PIC32MX270F256B.

Continuando la descrizione del circuito stampato, il secondo componente di maggiore importanza è U4 (VS1063): si tratta di un MP3 *slave processor* realizzato dalla *VLSI Solution*; questo componente si presenta come stretta evoluzione

## [piano di montaggio del **MODULO MP3**]

### Elenco Componenti:

C1÷C4: 100 nF ceramico (0603)  
C6, C9÷C15, C19, C20: 100 nF ceramico (0603)  
C5, C7: 33 pF ceramico (0603)  
C8, C16÷C18, C21, C24, C25: 10 µF ceramico (0603)  
C22, C27: 33 pF ceramico (0603)  
C23, C28: 3,3 nF ceramico (0603)  
C26, C32, C33: 1 µF ceramico (0603)  
C29: 47 nF ceramico (0603)  
C30, C31, C34, C35: 10 nF ceramico (0603)  
R1÷R3, R6÷R8, R12, R18, R19: 10 kohm (0603)  
R4, R5, R9: 470 ohm (0603)  
R10: 1 Mohm (0603)  
R11, R13, R17: 470 ohm (0603)  
R14: 10 ohm (0603)  
R15, R16: 20 ohm (0603)  
LD1: LED blu (0603)  
LD2: LED rosso (0603)  
L1÷L4: Ferrite bead 600 ohm 100 Hz (Farnell: 2336290)  
U1: AP7312 (Farnell: 2138324)  
U2: PIC32MX250F128B-I/ML (MF1129)  
U3: AP7333 (DigiKey: AP7333-33SRG-7DICT-ND)  
U4: VS1063 (Sparkfun: COM-11025)



X1: Quarzo 20 MHz per U2  
X2: Quarzo 12,288 MHz per U4

Varie:

- Connettore micro-SD
- Strip maschio 15 vie 90°
- Circuito stampato

del fratello minore VS1053. Le differenze principali risiedono nella capacità di registrare in formato MP3 oltre che in Ogg Vorbis. Inoltre il formato Ogg Vorbis non necessita dell'upload del plugin dedicato alla codifica, il che rende l'uso del nuovo dispositivo molto più flessibile.

Per quanto riguarda la sezione d'alimentazione, il circuito è dotato di ben tre regolatori LDO (*Low-dropout regulator*) contenuti in due integrati: U1 ed U3. Il primo è un AP7312 e contiene, in un package dalle dimensioni molto ridotte (è un SOT236) due regolatori, LDO: uno fornisce 1,8 V stabilizzati e l'altro 2,8 V stabilizzati. La corrente massima è di 150 mA per ciascuna uscita. I due LDO sono impiegati per alimentare la sezione DSP e la sezione analogica del VS1063. Invece U3 (AP7333) è un LDO singolo con uscita 3,3 V capace di

erogare una corrente massima di 300 mA; è impiegato per fornire alimentazione al microcontrollore principale ed alla microSD.

L'integrato VS1063 riceve il clock dal proprio oscillatore interno, quarzato mediante X1 (quarzo a 20 MHz) e abbinato ad un PLL (*Phase-locked loop*) interno per ottenere un clock di 50 MHz necessario per il corretto funzionamento del software. Il microcontrollore ha anch'esso un proprio quarzo (X2, da 12,288 MHz) che scandisce il funzionamento dell'oscillatore di clock integrato.

Il circuito è completato da reti in serie agli ingressi e alle uscite, inserite per filtrare gli spikes dovuti alla conversione digitale/analogico. Le reti sono di due tipi: quelle di uscita constano di due celle RC in cascata, una passa basso ed una passa-alto (costituiscono un filtro passa-banda) ed una bobina in

serie all'uscita (L1 per LOUT ed L2 per ROUT) mentre quelle di ingresso hanno un filtro passa-banda LC ed un passa-basso RC. Inoltre sono presenti le serie R15/C30 ed R16/C31 per stabilizzare l'impedenza dei piedini di uscita del VS1063.

## LA DEMOBOARD

Per consentire l'impiego sperimentale del riproduttore senza bisogno di realizzare circuiti elettronici dedicati, abbiamo realizzato una demoboard che ospita il modulo in un'apposita fila di 15 contatti femmina. Come vedete nel relativo schema elettrico (dove il modulo MP3 è siglato U2), la scheda fornisce al modulo l'alimentazione necessaria, prelevata dall'apposito connettore PWR o alternativamente dall'ingresso USB; nel primo caso i 5 volt provengono dal regolatore integrato U1, il cui ingresso è protetto contro l'inversione di polarità dal diodo D1 e filtrato da C1 e C2; l'uscita, filtrata da C3 e C4, applica i propri 5 V allo Schottky D2, che, insieme a D3, permette di applicare al modulo la tensione di valore maggiore senza che U1 venga sovraccaricato dall'USB e viceversa.

Sulla demoboard sono presenti sia i connettori di ingresso che di uscita per il *Line-In* ed il *Line-Out* (nominati IN ed OUT), sia 5 pulsanti (di cui 4 programmabili a piacimento). P5 è dedicato al hard-reset del riproduttore, invece P1, P2, P3 e P4 possono essere liberamente programmati, interagendo con il file di configurazione, come verrà spiegato più avanti. Per una rapida interazione suggeriamo di impiegare il file di configurazione che risulta in allegato al progetto che risulta già tarato per offrire le seguenti funzionalità, comunicazione seriale in modalità CLI tramite

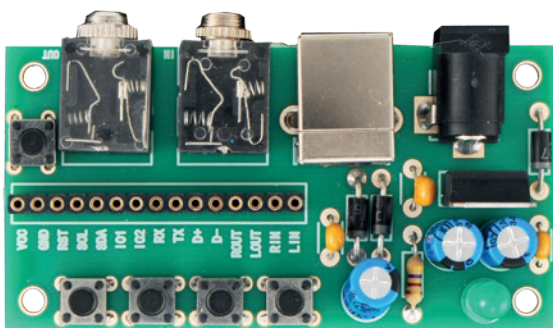
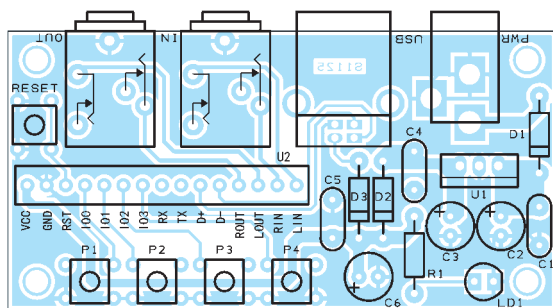
## [piano di montaggio della DEMOBOARD]

### Elenco Componenti:

R1: 470 ohm  
C1: 100 nF multistrato  
C2, C3, C5: 100  $\mu$ F 25 VL elettrolitico  
C4, C6: 100 nF multistrato  
U1: 7805  
U2: Modulo MP3 (FT1129)  
D1: 1N4007  
D2, D3: 1N5819  
P1 ÷ P4: Microswitch  
RESET: Microswitch  
LD1: LED 5 mm verde

#### Varie:

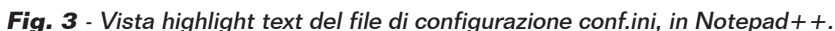
- Plug alimentazione
- Strip femmina 15 vie
- Connettore USB-B
- Presa jack 3.5mm da CS (2 pz.)
- Circuito stampato





## COME FUNZIONA

La modalità con cui si può interagire è configurabile e prevede tre opzioni: invio dei comandi tramite collegamento seriale (UART nativa o USB), interazione con segnali logici programmabili ed interazione con lettura e scrittura di opportuni registri tramite I<sup>2</sup>C. Nella **Fig. 3** è riportato un frammento di esempio del file di configurazione auto-generato dal software (il file di inizializzazione



Il formato adottato sin da *Windows 95* è tipicamente diviso in sezioni individuate da etichette inserite tra parentesi quadre (ad esempio *[Volume]*); al seguito della sezione sono presenti le coppie chiave-valore. Genericamente, i parametri sono separati da un segno di '=' dagli argomenti. Le

righe che iniziano per '#' oppure per ';' sono dei commenti e vengono semplicemente ignorate dal parser. Per editare il file di configurazione di nostro interesse è sufficiente usare un qualsiasi editor di testo, ma noi consigliamo *Notepad++* per Windows, in quanto ha la funzione di *highlight text* che torna utile nell'individuazione a colpo d'occhio delle sezioni, dei parametri o dei commenti. *Notepad++* è reperibile

**Tabella 3 - Rappresentazione tabellare delle sezioni e dei parametri di conf.ini.**

| Sezione Console |           |                                                                                         |
|-----------------|-----------|-----------------------------------------------------------------------------------------|
| Parametro       | Default   | Descrizione argomento                                                                   |
| baudrate        | 115200    | baudrate UART: 2400, 4800, 9600, 19200, 38400, 56000, 57600 e 115200                    |
| console         | 0         | Modalità d'interazione: 0 = CLI, 1 = SCC e 2 = SCC con eco del comando                  |
| port            | 0         | Porta console: 0 = UART e 1 = USB serial emulator                                       |
| verbose         | 1         | Livello dei messaggi informativi: 0 = disattivo, 1 = minimo e 2 = debug                 |
| boot            | 2         | Informazioni del bootloader: 0 = disattivo, 1 = minimo e 2 = esteso                     |
| Sezione Play    |           |                                                                                         |
| Parametro       | Default   | Descrizione argomento                                                                   |
| playlist        | play.pls  | Nome di default del file contenente la lista dei brani da riprodurre in sequenza        |
| repeat          | 0         | Modalità playlist: 0 = ripete per sempre e 1 = lo stop interrompe l'esecuzione          |
| Sezione Record  |           |                                                                                         |
| Parametro       | Default   | Descrizione argomento                                                                   |
| r_name          | track.mp3 | Nome di default del file in registrazione                                               |
| prog_over       | 1         | Scrittura progressiva: 0 = sovrascrive e 1 = incrementa progressivamente                |
| bitrate         | 128       | bitrate in kbit/s: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 e 320       |
| samplerate      | 44100     | samplerate in Hz: 8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100 e 48000         |
| bitrate_mode    | 1         | Modalità del bitrate: 0 = Quality Mode, 1 = VBR, 2 = ABR e 3 = CBR                      |
| gain            | 1         | Guadagno in fase di registrazione: da 1x a 63x e 0 abilita il guadagno automatico       |
| max_gain        | 10        | Massimo valore del guadagno automatico: da 1x a 63x                                     |
| format          | 0         | Formato di codifica: 0 = MP3 e 1 = Ogg Vorbis                                           |
| Sezione Volume  |           |                                                                                         |
| Parametro       | Default   | Descrizione argomento                                                                   |
| volume          | 10        | Attenuazione del volume all'accensione in passi di 0,5 dB                               |
| balance         | 0         | Bilanciamento dei canali: -100 = 100% a destra, 0 disabilitato e +100 = 100% a sinistra |
| treble_db       | 0         | Guadagno toni alti in passi di 1,5 dB (-8 ... 7, 0 = off)                               |
| treble_freq     | 1         | Frequenza di taglio inferiore per i toni alti in passi di 1000 Hz (1 ... 15)            |
| bass_db         | 0         | Guadagno toni bassi in passi di 1 dB (0 ... 15, 0 = off)                                |
| bass_freq       | 2         | Frequenza di taglio superiore per i toni bassi in passi di 10 Hz (2 ... 15)             |
| Sezione GPIO    |           |                                                                                         |
| Parametro       | Default   | Descrizione argomento                                                                   |
| mode<X>         |           | Modalità di funzionamento associabile a ciascun pin di IO, 0 = disabilitato             |
| duration<X>     |           | Durata temporale del cambiamento di stato: 0 = toggle                                   |
| idle<X>         |           | Stato logico in condizioni di riposo: 0 = basso o 1 = alto                              |
| pull<X>         |           | Pull-up/down interno: 0 = off, 1 = pull-up e 2 = pull-down                              |

bile gratuitamente all'indirizzo web: <http://notepad-plus-plus.org/>. Nella **Tabella 3** sono riportate, con relativa descrizione, tutte le voci presenti nel file *conf.ini* che l'utente può impostare secondo le proprie esigenze.

Nello sviluppo del software del riproduttore e registratore MP3 ci siamo concentrati sugli aspetti funzionali, cercando di implementare il maggior numero di casi d'uso che abbiamo immaginato. Nella realizzazione dei singoli scenari abbiamo pensato di facilitare l'uso del circuito consentendo di personalizzare le funzionalità previste tramite l'impiego di un file testuale di configurazione, oltre ad im-

plementare una *Shell Unix like*, nominata *Command Line Interpreter* (CLI), per interagire dinamicamente con le funzionalità del dispositivo.

L'impiego di una riga di comando per l'inserimento dei parametri o delle operazioni è sicuramente agevole per un essere umano, ma può essere inappropriata nel caso in cui si voglia usare il circuito in maniera automatizzata rispetto ad un'altra CPU, per questo abbiamo previsto una terza modalità d'interazione nominata *Single Character Commands* (SCC) che consente di impartire le sequenze di comandi desiderati in

maniera molto più compatta. In alternativa a queste due forme di interazione seriale, si può usare un approccio ancora più elementare, nominato *IO general purpose*, che si basa sull'applicazione di segnali logici tra 0 V e 3,3 V. Per poter selezionare una modalità di interazione e renderla permanente, bisogna modificare il file di configurazione nominato *conf.ini* come descritto nella sezione *IO general purpose* di questo articolo.

Il file *conf.ini* contiene tutte le coppie parametro-argomento, modificabili dall'utente e delle brevi descrizioni associate ad ognuno dei precedenti parametri modificabili. Inoltre, per preve-

nire la perdita di una configurazione valida, il software provvede automaticamente a creare una copia di backup dell'ultima configurazione valida creata; il file di *dump* prenderà il nome di *conf.dmp*. Nel caso in cui il file *conf.ini* risultasse corrotto per qualche motivo accidentale, il software impiegherà la copia di backup per il ripristino dei parametri persi; se per qualche motivo anche la copia di backup risultasse corrotta, allora il software rigenererà in maniera automatica una nuova configurazione predefinita.

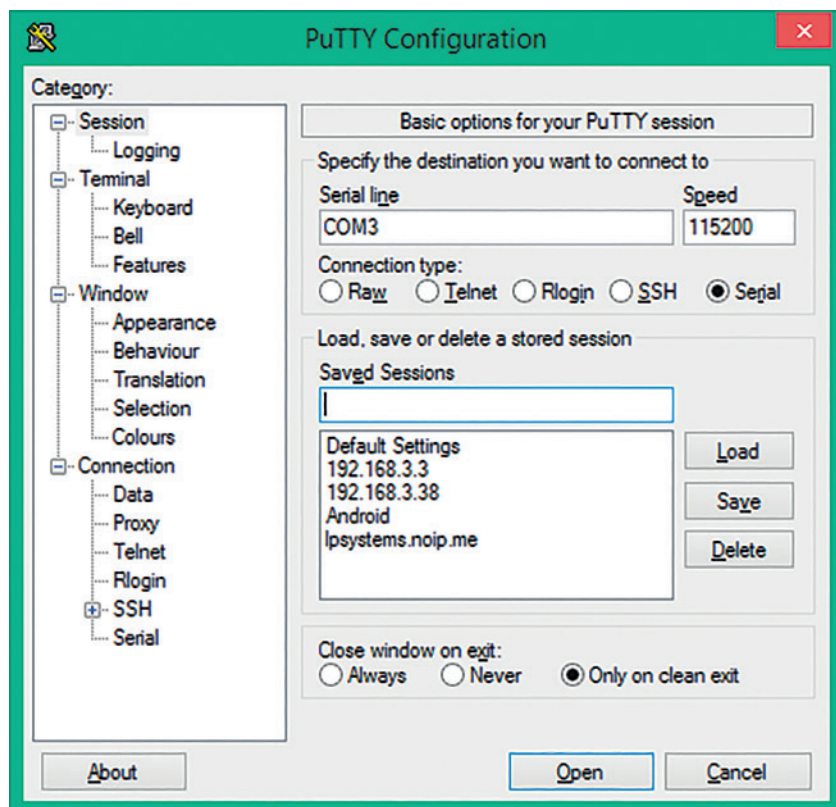
Avendo nominato la microSD come sistema di archiviazione principale, è bene precisare che il software è in grado di gestire correttamente memorie formattate in FAT12, FAT16 e FAT32; per ovvi motivi di dimensioni si consiglia di usare il File System

FAT32. La massima dimensione da noi provata è di 32 GB formattata in FAT32. Il software non ha limitazioni per quanto riguarda i nomi e le gerarchie dei file, infatti può gestire file con nomi lunghi *Long filename* (LFN) disposti anche in sotto cartelle. Infine, per ottenere delle buone prestazioni in fase di registrazione è consigliabile usare delle microSD molto

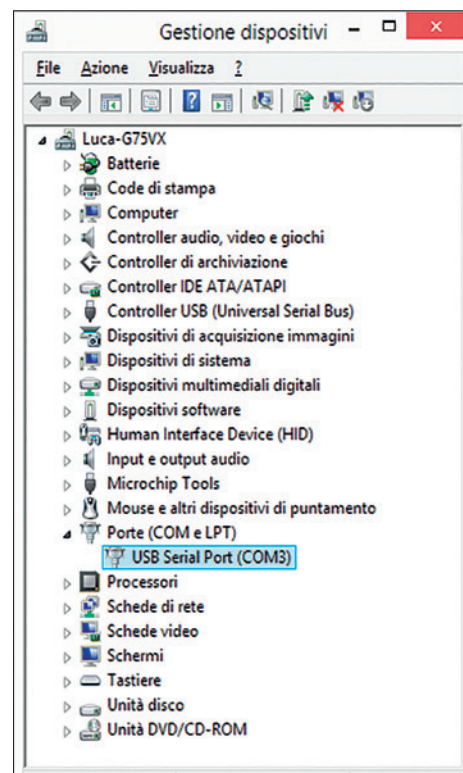
veloci, di classe 4 oppure 10. In caso contrario, risulta impossibile registrare un flusso MP3 con un elevato bit-rate e samplerate senza la perdita di frammenti del flusso audio. L'unica limitazione software, presente nella versione *v0.1*, riguarda l'impiego di spazi bianchi all'interno dei nomi dei file o delle cartelle; per consentire un corretto funzionamento bisogna evitare l'impiego



**Fig. 6** - Messaggio generato dal bootloader e dalla routine di inizializzazione.



**Fig. 4** - Vista delle impostazioni dell'emulatore di terminale PuTTY.



**Fig. 5** - La COM3 in Gestione dispositivi di Windows 7.

|                |                                                                                       |                                                                             |
|----------------|---------------------------------------------------------------------------------------|-----------------------------------------------------------------------------|
| <b>Comando</b> | <b>bass [att, freq]</b>                                                               |                                                                             |
| Esempio        | bass 7 10                                                                             |                                                                             |
| Descrizione    | Imposta o ritorna il valore dei toni bassi                                            |                                                                             |
| Parametro 1    | att                                                                                   | guadagno toni bassi in passi di 1 dB (0 ... 15, 0 = off)                    |
| Parametro 2    | freq                                                                                  | frequenza di taglio superiore per i toni bassi in passi di 10 Hz (2 ... 15) |
| <b>Comando</b> | <b>cat file</b>                                                                       |                                                                             |
| Esempio        | cat conf.ini                                                                          |                                                                             |
| Descrizione    | Mostra il contenuto di un file testuale                                               |                                                                             |
| Parametro 1    | file                                                                                  | nome del file testuale                                                      |
| <b>Comando</b> | <b>cd dir</b>                                                                         |                                                                             |
| Esempio        | cd songs                                                                              |                                                                             |
| Descrizione    | Cambia la cartella di lavoro corrente                                                 |                                                                             |
| Parametro 1    | dir                                                                                   | nome della nuova cartella di lavoro                                         |
|                | ..                                                                                    | se usata la stringa '..' si torna alla cartella genitore                    |
|                | /                                                                                     | se usato il carattere '/' si torna alla cartella root                       |
| <b>Comando</b> | <b>cfg [opt]</b>                                                                      |                                                                             |
| Esempio        | cfg 1                                                                                 |                                                                             |
| Descrizione    | Imposta o ritorna il valore associato al terminale in ascolto                         |                                                                             |
| Parametro 1    | opt                                                                                   | 0 imposta l'ascolto sull'UART nativa, 1 imposta l'ascolto sull'USB          |
| <b>Comando</b> | <b>cp src cpy</b>                                                                     |                                                                             |
| Esempio        | cp old.mp3 new.mp3                                                                    |                                                                             |
| Descrizione    | Crea una copia di un file                                                             |                                                                             |
| Parametro 1    | src                                                                                   | nome del file di origine                                                    |
| Parametro 2    | cpy                                                                                   | nome del file di destinazione                                               |
| <b>Comando</b> | <b>ear [n]</b>                                                                        |                                                                             |
| Esempio        | ear 12000                                                                             |                                                                             |
| Descrizione    | Imposta o ritorna il valore associato al parametro EarSpeaker Spatial Processing      |                                                                             |
| Parametro 1    | n                                                                                     | miglior impostazione quando si impiegano gli altoparlanti                   |
|                |                                                                                       | 12000: Adatto per l'ascolto di normali partiture musicali con le cuffie     |
|                |                                                                                       | 38000: Adatto per ottenere un effetto 'lontano'                             |
|                |                                                                                       | 50000: Adatto per ottenere un effetto 'invecchiato'                         |
| <b>Comando</b> | <b>flash magic</b>                                                                    |                                                                             |
| Esempio        | flash 0xAA996655                                                                      |                                                                             |
| Descrizione    | Consente di caricare un nuovo firmware all'interno della memoria del microcontrollore |                                                                             |
| Parametro 1    | magic                                                                                 | numero magico usato per prevenire scritture accidentali del firmware        |
| <b>Comando</b> | <b>fplay [spd]</b>                                                                    |                                                                             |
| Esempio        | fplay 1                                                                               |                                                                             |
| Descrizione    | Imposta o ritorna il valore associato alla velocità di esecuzione                     |                                                                             |
| Parametro 1    | spd                                                                                   | moltiplicatore (1 = 1x ... 10 = 10x)                                        |
| <b>Comando</b> | <b>gpio [n, m, d, i, p]</b>                                                           |                                                                             |
| Esempio        | gpio 2 1 100 1 1                                                                      |                                                                             |
| Descrizione    | Imposta o ritorna il valore associato ad un GPIO                                      |                                                                             |
| Parametro 1    | n                                                                                     | numero dell'IO da 0 a 7                                                     |
| Parametro 2    | m                                                                                     | modalità (si veda il paragrafo IO general purpose)                          |
| Parametro 3    | d                                                                                     | durata in millisecondi (10 ... 1000, 0 = toggle)                            |
| Parametro 4    | i                                                                                     | stato in attesa: 0 = basso or 1 = alto                                      |
| Parametro 5    | p                                                                                     | Pull-up/down interno. 0 = off, 1 = pull-up or 2 = pull-down                 |
| <b>Comando</b> | <b>help [cmd]</b>                                                                     |                                                                             |
| Esempio        | help play                                                                             |                                                                             |
| Descrizione    | Ritorna la lista dei comandi o la descrizione dettagliata di un comando               |                                                                             |
| Parametro 1    | cmd                                                                                   | comando di cui ottenere la descrizione dettagliata                          |
| <b>Comando</b> | <b>info</b>                                                                           |                                                                             |
| Esempio        | info                                                                                  |                                                                             |
| Descrizione    | Ritorna le informazioni sulla traccia in esecuzione o in registrazione                |                                                                             |
| <b>Comando</b> | <b>ls [-l[a]]</b>                                                                     |                                                                             |
| Esempio        | ls -la                                                                                |                                                                             |
| Descrizione    | Ritorna il contenuto della corrente cartella di lavoro                                |                                                                             |
| Parametro 1    | l                                                                                     | formato dei risultati estesi                                                |
| Parametro 2    | a                                                                                     | include i file nascosti                                                     |
| <b>Comando</b> | <b>mkdir dir</b>                                                                      |                                                                             |
| Esempio        | mkdir songs                                                                           |                                                                             |
| Descrizione    | Crea una nuova cartella                                                               |                                                                             |
| Parametro 1    | dir                                                                                   | nome della nuova cartella                                                   |
| <b>Comando</b> | <b>mv file path</b>                                                                   |                                                                             |
| Esempio        | mv 1.mp3 folder/1.mp3                                                                 |                                                                             |
| Descrizione    | Sposta o rinomina un file o una cartella                                              |                                                                             |
| Parametro 1    | file                                                                                  | file da spostare o rinominare                                               |
| Parametro 2    | path                                                                                  | percorso di destinazione                                                    |

**Tabella 4** - Rappresentazione schematizzata dei comandi accettati in modalità CLI.

del carattere ' ' (ovvero spazio), magari sostituendolo con ' \_ ' (*underscore*).

Per poter impiegare rapidamente il nostro circuito proponiamo una rapida panoramica su tutti i comandi impartibili in modalità CLI, ovvero con un'interazione da riga di comando. Per questo consigliamo di usare come simulatore di terminale il famoso programma *PuTTY*, reperibile gratuitamente all'indirizzo web: <http://www.putty.org/>. Quest'ultimo dovrà essere configurato per accedere in maniera seriale al dispositivo. Il software del riproduttore è stato realizzato per consentire in maniera mutua esclusiva l'impiego della modalità CLI o SCC tramite UART nativa oppure tramite USB *Serial Emulator*. Quindi per poter interagire con il riproduttore tramite *PuTTY* è necessario collegare il circuito al PC; le alternative sono due: la prima consiste nell'impiegare un convertitore USB seriale oppure un convertitore RS232 TTL collegandolo alla seriale del riproduttore. In questo caso, da *PuTTY* bisogna impostare il programma come in **Fig. 4**. Se il file *conf.ini* non ha subito modifiche, il firmware abilita l'ascolto dei comandi in modalità CLI sulla UART nativa (pin 7 RX e pin 8 TX con baud-rate di 115.200 bps).

Come visibile nella **Fig. 4**, bisogna selezionare la *Connection type: Serial*, impostare il parametro *Speed* a 115200 e selezionare la *Serial line* associata alla porta seriale del PC desiderata (RS232 oppure convertitore USB seriale), per gli utenti Windows è possibile conoscere l'elenco delle periferiche seriali, nominate COMx, agendo da pannello di controllo, gestione dispositivi, voce *Porte (COM e LPT)* come da **Fig. 5**. Una volta instaurata la connes-



|                |                                                                                             |                                                                                   |
|----------------|---------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------|
| <b>Comando</b> | <b>pause [t]</b>                                                                            |                                                                                   |
| Esempio        | pause 100                                                                                   |                                                                                   |
| Descrizione    | Inserisce o disinserisce la pausa durante la riproduzione o la registrazione                |                                                                                   |
| Parametro 1    | t                                                                                           | inserisce la pausa per t millisecondi                                             |
| <b>Comando</b> | <b>play file</b>                                                                            |                                                                                   |
| Esempio        | play song.mp3                                                                               |                                                                                   |
| Descrizione    | Consente la riproduzione di una traccia di formato MP3 oppure Ogg Vorbis                    |                                                                                   |
| Parametro 1    | file                                                                                        | nome della traccia da eseguire                                                    |
| <b>Comando</b> | <b>playlist list</b>                                                                        |                                                                                   |
| Esempio        | playlist play.lst                                                                           |                                                                                   |
| Descrizione    | Consente di riprodurre le tracce di formato MP3 oppure Ogg Vorbis presenti in una play list |                                                                                   |
| Parametro 1    | list                                                                                        | nome della play list da eseguire                                                  |
| <b>Comando</b> | <b>reboot</b>                                                                               |                                                                                   |
| Esempio        | reboot                                                                                      |                                                                                   |
| Descrizione    | Consente di effettuare un soft reset dell'intero circuito                                   |                                                                                   |
| <b>Comando</b> | <b>rec file [s, b, g, m, v, f]</b>                                                          |                                                                                   |
| Esempio        | rec rec.mp3 44100 160 0 5 1 0                                                               |                                                                                   |
| Descrizione    | Sposta o rinomina un file o una cartella                                                    |                                                                                   |
| Parametro 1    | file                                                                                        | nome del file della traccia                                                       |
| Parametro 2    | s                                                                                           | sample rate in Hz: 8000, 11025, 12000, 16000, 22050, 24000, 32000, 44100 e 48000  |
| Parametro 3    | b                                                                                           | bitrate in kbit/s: 32, 40, 48, 56, 64, 80, 96, 112, 128, 160, 192, 224, 256 e 320 |
| Parametro 4    | g                                                                                           | Guadagno in fase di registrazione: da 1x a 63x e 0 abilita il guadagno automatico |
| Parametro 5    | m                                                                                           | Massimo valore del guadagno automatico: da 1x a 63x                               |
| Parametro 6    | v                                                                                           | Modalità del bitrate: 0 = Quality Mode, 1 = VBR, 2 = ABR e 3 = CBR                |
| Parametro 7    | f                                                                                           | Formato di codifica: 0 = MP3 e 1 = Ogg Vorbis                                     |
| <b>Comando</b> | <b>rtcc [DDMMYYYY hh mm ss]</b>                                                             |                                                                                   |
| Esempio        | rtcc 18 06 2014 14 45 00                                                                    |                                                                                   |
| Descrizione    | Imposta o ritorna il valore del Real Time Clock Calendar                                    |                                                                                   |
| Parametro 1    | DD                                                                                          | giorno del mese: da 1 a 31                                                        |
| Parametro 2    | MM                                                                                          | mese dell'anno: da 1 a 12                                                         |
| Parametro 3    | YYYY                                                                                        | anno solare esteso a 4 cifre                                                      |
| Parametro 4    | hh                                                                                          | ora: da 0 a 23                                                                    |
| Parametro 5    | mm                                                                                          | minuti: da 0 a 59                                                                 |
| Parametro 6    | ss                                                                                          | secondi: da 0 a 59                                                                |
| <b>Comando</b> | <b>rm file</b>                                                                              |                                                                                   |
| Esempio        | rm 123.ogg                                                                                  |                                                                                   |
| Descrizione    | Elimina un file                                                                             |                                                                                   |
| Parametro 1    | file                                                                                        | nome del file da eliminare                                                        |
| <b>Comando</b> | <b>rmdir dir [-not-empty]</b>                                                               |                                                                                   |
| Esempio        | rmdir songs                                                                                 |                                                                                   |
| Descrizione    | Elimina una cartella                                                                        |                                                                                   |
| Parametro 1    | dir                                                                                         | nome della cartella da eliminare                                                  |
| Parametro 2    | -not-empty                                                                                  | forza la rimozione di una cartella non vuota                                      |
| <b>Comando</b> | <b>speed [spd]</b>                                                                          |                                                                                   |
| Esempio        | speed 1000                                                                                  |                                                                                   |
| Descrizione    | Imposta o ritorna il valore dello speed shifter                                             |                                                                                   |
| Parametro 1    | spd                                                                                         | fattore moltiplicativo x (680 = 0.68x ... 1000 = 1.00x ... 1640 = 1.64x)          |
| <b>Comando</b> | <b>stop [-all]</b>                                                                          |                                                                                   |
| Esempio        | stop                                                                                        |                                                                                   |
| Descrizione    | Termina la riproduzione o la registrazione                                                  |                                                                                   |
| Parametro 1    | -all                                                                                        | termina l'esecuzione di tutte i brani della play list                             |
| <b>Comando</b> | <b>treble [att, freq]</b>                                                                   |                                                                                   |
| Esempio        | treble 7 10                                                                                 |                                                                                   |
| Descrizione    | Imposta o ritorna il valore del guadagno delle frequenze alte                               |                                                                                   |
| Parametro 1    | att                                                                                         | guadagno toni alti in passi di 1,5 dB (-8 ... 7, 0 = off)                         |
| Parametro 2    | freq                                                                                        | frequenza di taglio inferiore per i toni alti in passi di 1.000 Hz (1 ... 15)     |
| <b>Comando</b> | <b>verbose [lev]</b>                                                                        |                                                                                   |
| Esempio        | verbose 2                                                                                   |                                                                                   |
| Descrizione    | Imposta o ritorna il valore dei messaggi informativi                                        |                                                                                   |
| Parametro 1    | lev                                                                                         | livello informativo: 0 = disattivati, 1 = minimo e 2 = debug                      |
| <b>Comando</b> | <b>version</b>                                                                              |                                                                                   |
| Esempio        | version                                                                                     |                                                                                   |
| Descrizione    | Ritorna il valore le informazioni circa la versione del firmware                            |                                                                                   |
| <b>Comando</b> | <b>vol [att, [bal]]</b>                                                                     |                                                                                   |
| Esempio        | treble 7 10                                                                                 |                                                                                   |
| Descrizione    | Imposta o ritorna il valore del volume                                                      |                                                                                   |
| Parametro 1    | att                                                                                         | attenuazione del volume in passi di 0,5 dB (255 = off ... 0 = max vol)            |
| Parametro 2    | bal                                                                                         | attenuazione del canale in percentuale (L = -100% ... 0% ... 100% = R)            |

**Tabella 4 - ... continua**  
da pagina precedente.

sione con il dispositivo, sulla finestra di dialogo dell'emulatore *PuTTY* all'accensione apparirà una lista di stringhe come in **Fig. 6**, queste rappresentano il risultato del processo di boot e possono essere disabilitate impostandone il valore della proprietà *boot* nel file *conf.ini* a '0'. A questo punto il simbolo '>' indica che il dispositivo è pronto per ricevere un comando. Ad esempio, digitando 'ls' seguito dalla pressione del tasto *enter* si otterrà l'elenco dei file presenti nella directory corrente. Per conoscere l'elenco dei comandi disponibili basta digitare 'help', per conoscere i dettagli di un singolo comando basta digitare 'help ' seguito dal nome del comando, così ad esempio 'help ls' stamperà i dettagli relativi all'uso del comando *list*. Per poter inserire facilmente un comando o un nome di un file lasciando l'onere del completamento al software basta ad esempio digitare i primi caratteri del comando e premere il tasto *tab*; ad esempio 'pl' seguito dal tasto *tab* inserirà automaticamente il comando, oppure, nel caso di più comandi con lo stesso prefisso, stamperà l'elenco dei comandi disponibili.

Similmente, per l'inserimento di un nome di un file, sarà sufficiente premere *tab* dopo il nome del comando seguito da uno spazio bianco. Ad esempio digitando

## Listato 1

```
>play my_track.mp3<enter>
File opened successful
>vol<enter>
Volume: -5.0dB balance: 0
Left: 10pt
Right: 10pt
>vol 15<enter>
>stop<enter>
Finish playing
Try to close file
File closed successful
```

## Listato 2

```
>playlist playlist.pls<enter>
Try to open playlist: playlist.pls
Playlist opened successful
Playlist execute: test1.mp3
Try to open: test1.mp3
File opened successful
>bass<enter>
Bass enhancement: 0dB @ 20Hz
>bass 7 10<enter>
>stop
Finish playing
Try to close file
File closed successful
Playlist ended
```

do 'play <tab>' si otterrà l'elenco dei file presenti nella directory corrente, oppure se vi è presente un unico file con quel particolare prefisso, il software inserirà automaticamente tutto il nome del file. Ad esempio, se nella directory corrente è presente un file del tipo *my\_favourite\_track.mp3* e si digita il comando 'play my' e poi si preme *tab*, si otterrà come risultato la trascrizione automatica del nome del file sulla riga di comando. A questo punto possiamo presentare una panoramica dei comandi disponibili in modalità CLI; ogni comando può essere usato singolarmente o con dei parametri opzionali: questi ultimi saranno indicati tra i simboli '[' e ']' per mostrare l'opzionalità dell'argomento. Infine ricordiamo che, come in una vera *shell linux*, la pressione degli *arrow keys* (freccette) genera i relativi comandi. Ad esempio con la freccia sinistra e destra si può spostare il cursore, invece con l'impiego della freccia superiore si richiama l'ultimo comando inserito.

### COMANDO IN MODALITÀ CLI

Sinora abbiamo visto l'elenco dei comandi disponibili; ecco ora alcuni esempi d'uso comuni. Il primo (**Listato 1**) consiste nell'avviare la riproduzione di un certo brano musicale, regolare il volume e stoppare l'esecu-

zione. In ognuno degli esempi seguenti immagineremo che il valore *verbose* sia impostato a 2 (modalità debug) nel *conf.ini*. Il secondo esempio consiste nell'esecuzione di una playlist, nella regolazione dei toni bassi e nella terminazione della playlist (**Listato 2**). Il terzo mostrerà: l'uso dell'*help*, l'uso della pausa, e la creazione di cartella con rispettiva navigazione (il relativo codice è nel **Listato 3**).

L'ultimo esempio (**Listato 4**) mostra come poter conoscere la versione di firmware corrente e poter in seguito effettuare l'upgrade ad una versione successiva caricando all'interno della microSD il nuovo firmware. Va precisato che il firmware si compone di due progetti separati: il *bootloader* che è installato ad hoc nella sezione della Flash dedicata al *boot* e può avvenire esclusivamente con un programmatore ad esempio l'*In-Circuit Debugger (MPLAB ICD 3)* interagendo con gli opportuni pin di IO presenti sul connettore a pettine, e dal progetto vero e proprio. Quest'ultimo può essere caricato in un sol colpo insieme al *bootloader* dall'*ICD3* oppure essere caricato/aggiornato in un secondo momento dallo stesso *bootloader*. Per poter effettuare un aggiornamento è necessario inserire il firmware nella microSD ed inviare il comando *flash*, il file contenente

il nuovo software dovrà essere inserito esclusivamente nella root della microSD e rinominato esattamente: *image.hex*.

### I/O GENERAL PURPOSE

Ognuno degli otto pin di I/O può avere una particolare funzionalità. Di seguito vedremo come configurare il file *conf.ini* per rendere le modifiche permanenti, oppure provarle tramite comando 'gpio'. La **Tabella 5** riassume le funzionalità che è possibile associare a ciascun pin. Per testare una funzionalità in maniera momentanea è possibile usare il comando 'gpio'; quest'ultimo, se lanciato senza parametri restituisce la configurazione corrente, mentre con gli opportuni parametri consente

## Listato 3

```
>help<enter>
bass
[...]
verbose
version
vol
>pause<enter>
Pause: ON
>pause<enter>
Pause: OFF
>pause 1000<enter>
Pause for 1000 ms
>mkdir testDir
>ls -la<enter>
-drw 0.0 Byte 9/34/2062 0:7 .
-drw 0.0 Byte 9/34/2062 0:7 ..
-drw 0.0 Byte 9/34/2062 0:7 testDir
Total: 3
>cd testDir<enter>
>cd /<enter>
```

## Listato 4

```
>version<enter>
LP Systems Bootloader v0.2
MCU Type: PIC32MX250F128B @ 50MHz
Embedded MP3 Player v0.1
>flash 0xAA996655<enter>
Entering in bootloader mode to flash
new image.
Send trigger signal to start firmware
upgrade..
```

**Tabella 5** - Descrizione delle funzioni associabili all'I/O programmabile del microcontrollore.

| Comando          | Funzionalità                                                                                           |
|------------------|--------------------------------------------------------------------------------------------------------|
| 0                | Al pin non è associata alcuna funzionalità.                                                            |
| <b>Ingresso:</b> |                                                                                                        |
| 1                | Avvia l'esecuzione dei file contenuti nella play list.                                                 |
| 2                | Ferma l'esecuzione corrente.                                                                           |
| 3                | Avvia l'esecuzione della play list o termina l'esecuzione corrente.                                    |
| 4                | Attiva o disattiva la pausa.                                                                           |
| 5                | Avvia la registrazione nominando il file secondo i parametri di conf.ini.                              |
| 6                | Avvia la registrazione o termina la registrazione.                                                     |
| 7                | Soft reset.                                                                                            |
| <b>Uscita</b>    |                                                                                                        |
| 20               | Riproduzione avviata.                                                                                  |
| 21               | Riproduzione terminata.                                                                                |
| 22               | Riproduzione in pausa.                                                                                 |
| 23               | Riproduzione avviata o terminata.                                                                      |
| 24               | Riproduzione avviata, in pausa o terminata.                                                            |
| 25               | File non trovato.                                                                                      |
| 26               | Registrazione avviata.                                                                                 |
| 27               | Registrazione terminata.                                                                               |
| 28               | Registrazione in pausa.                                                                                |
| 29               | Registrazione avviata o terminata.                                                                     |
| 30               | Registrazione avviata, in pausa o terminata.                                                           |
| 31               | Riproduzione o registrazione avviata.                                                                  |
| 32               | Riproduzione o registrazione terminata.                                                                |
| 33               | Riproduzione o registrazione in pausa.                                                                 |
| 34               | Riproduzione o registrazione avviata o terminata.                                                      |
| 35               | Riproduzione o registrazione avviata, in pausa o terminata.                                            |
| 36               | Un comando è stato correttamente interpretato.                                                         |
| 37               | Un comando non è stato correttamente interpretato.                                                     |
| 38               | La microSD è stata correttamente inizializzata.                                                        |
| <b>Speciale:</b> |                                                                                                        |
| 60               | Abilita I <sup>2</sup> C in modalità slave, disponibile solo per i pin 4 (SCL) e 5 (SDA).              |
| 70               | Abilita l'input analogico, disponibile solo per i pin 6 (ANO) e 7 (AN1).                               |
| 80               | Abilita il bus UART nativa, disponibile solo per i pin 8 (RX) e 9 (TX).                                |
| 90               | Abilita il bus USB Composite device: Serial emulator + Mass Storage, solo pin 10 (USB_P) e 11 (USB_N). |

di modificare la funzionalità. Il primo parametro del comando è 'n' e si riferisce al numero di I/O da modificare (il valore è compreso tra 0 e 7). Il parametro successivo è 'm' e indica la funzionalità; è un valore numerico da scegliere come da **Tabella 5**: '0' corrisponde a nessuna funzionalità, tutti gli altri valori consentono di associare una specifica funzionalità all'I/O in questione. Il terzo parametro è 'd' e corrisponde alla durata del segnale digitale: nel caso

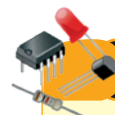
di funzionalità di input indica il tempo minimo necessario per validare una variazione di segnale, mentre se l'I/O è impostato come uscita indica l'intervallo temporale in cui il segnale cambierà stato. Nel caso in cui il valore sia impostato a '0' lo stato logico corrente verrà modificato in maniera permanente invertendo l'ultima condizione logica. Il quarto parametro è 'i' e serve a settare lo stato di *idle*, ovvero il valore che sia in caso di input che di output è conside-

rato come stato di riposo. Infine l'ultimo parametro è 'p' e serve per poter abilitare il pull-up o il pull-down interno. In questo caso '0' disabilita la funzionalità, '1' abilita il pull-up e '2' abilita il pull-down.

Analogamente a quanto descritto per la variazione momentanea (tramite il comando 'gpio'), è possibile ottenere il duale permanente interagendo con i rispettivi parametri del file *conf.ini*; in particolare, impostando i seguenti parametri: *mode<X>*, *duration<X>*, *idle<X>* e *pull<X>*, dove anche in questo caso la X rappresenta il numero di I/O interessato e sarà un valore compreso tra 0 e 7.

## CONCLUSIONI

Concludiamo quest'articolo anticipandovi che nella prossima puntata descriveremo la modalità di funzionamento SCC ed I<sup>2</sup>C, nonché il software sviluppato da noi.



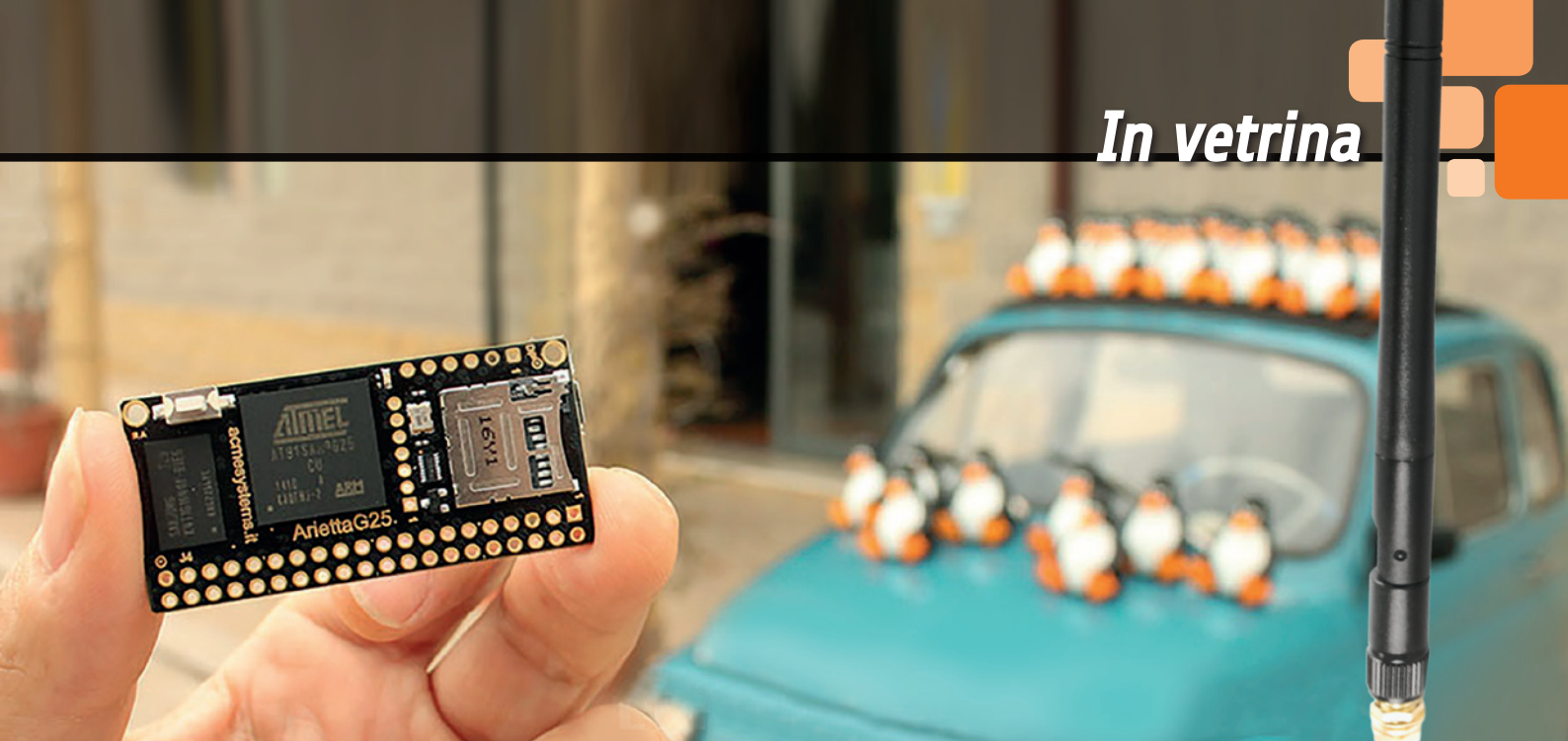
## per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono facilmente reperibili sul mercato. I master dei C.S., il software ed il firmware sono scaricabili gratuitamente dal sito della rivista ([www.elettronica-in.it](http://www.elettronica-in.it)). Il modulo MP3 è disponibile anche montato e collaudato al prezzo di Euro 42,00 (cod. FT1129M) mentre la demoboard relativa è disponibile in scatola di montaggio al prezzo di 6,00 Euro (cod. FT1125K). I prezzi si intendono IVA compresa.

Il materiale va richiesto a:  
Futura Elettronica, Via Adige 11,  
21013 Gallarate (VA)  
Tel: 0331-799775 • Fax: 0331-792287  
<http://www.futurashop.it>







# ARIETTA G25

## L'ULTIMO PINGUINO NATO DA ACME SYSTEMS

di MARCO MAGAGNIN

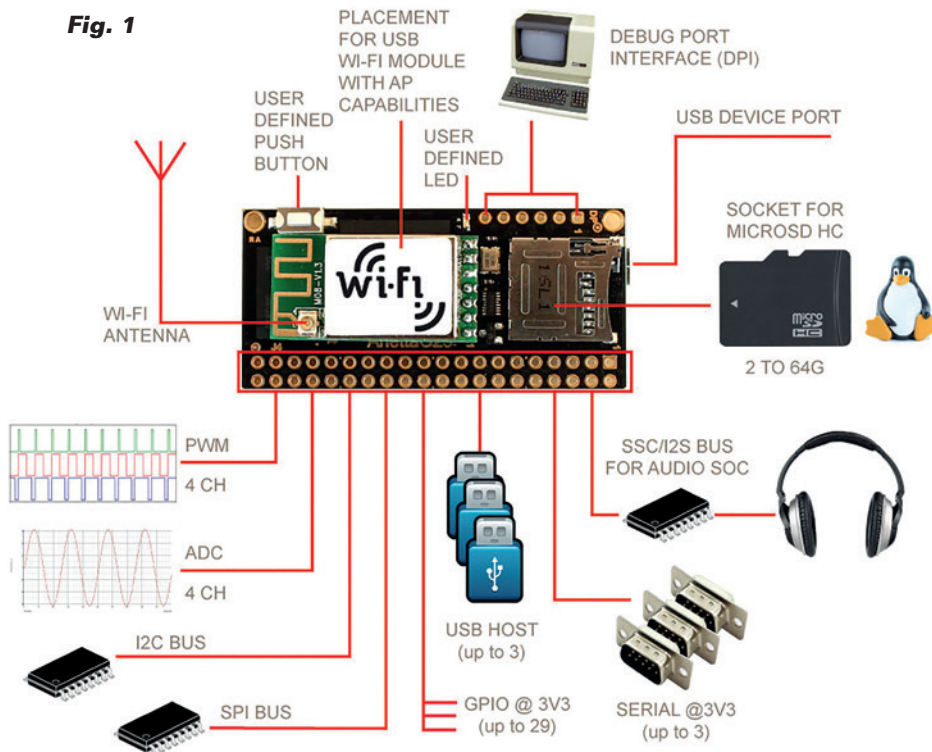


Se avete “rotto il ghiaccio” con GNU/Linux grazie a Raspberry Pi, con questa scheda spiccate il salto verso il mondo professionale, in modo guidato e con strumenti totalmente “Open”. Arietta G25 è la “mascotte” di una serie di schede professionali progettate e realizzate in Italia da Acme Systems.

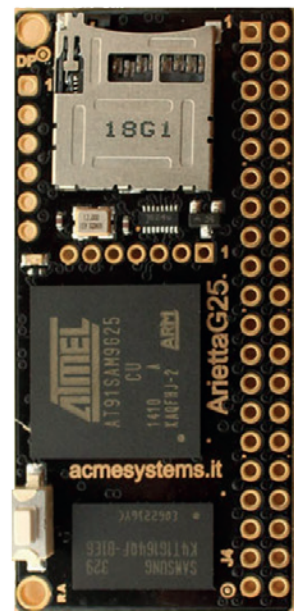
**P**erché occuparci di un altro microcomputer embedded animato da GNU/Linux? Ebbene, i motivi sono tanti: il primo è che la scheda Arietta G25 condensa in pochi centimetri quadrati di basso costo lunghi anni di esperienza nel campo GNU/Linux embedded, concretizzati nella produzione di sistemi professionali distribuiti in migliaia

di esemplari. Il secondo è che si tratta di un prodotto totalmente progettato e realizzato in Italia da Acme Systems (<http://www.acmesystems.it>, azienda che garantisce il supporto alle proprie schede per oltre 5 anni), cosa che di questi tempi non può che farci piacere. Il terzo è che le caratteristiche tecniche sono di tutto rispetto per una scheda di queste

**Fig. 1**



**Fig. 2**



dimensioni. Insomma, Arietta G25 è ad un livello tecnico che la rende un tassello importante in un percorso di formazione e di utilizzo professionale dei sistemi GNU/Linux embedded, che, in pratica, comporta la capacità di personalizzare ed ottimizzare il kernel di GNU/Linux in base alle esigenze dei nostri progetti. Kernel "tuttofare" come quello di Raspberry Pi, zeppi di driver e moduli "plug and play" permettono una gestione semplice per la maggior parte delle periferiche, ma appesantiscono la distribuzione, in termini sia di occupazione di memoria, sia di prestazioni. Nel nostro caso possiamo mantenere il kernel il più "magro" possibile, risparmiando memoria e cicli di CPU, godendo di prestazioni di tutto rispetto. Ciò, anche a vantaggio di un limitato consumo di energia, il che rende possibile l'utilizzo del sistema a batteria. Vedremo come affrontare questo aspetto con delle guide passo-passo che vi presenteremo in seguito.

La scheda è dotata di un GPIO con 40 pin di I/O configurabili. Oltre agli I/O digitali "classici" sono presenti dei pin che possono essere configurati sia come I/O digitali generici, sia come ingressi/uscite analogiche ADC e PWM, oltre che come bus I<sup>2</sup>C, SPI, I<sup>2</sup>S e UART. I progettisti di ACME Systems hanno voluto dare un valore aggiunto alla loro offerta con un sito ricco di informazioni e diversi "tutorial" per gli utilizzatori delle loro schede. In particolare, è presente un'applicazione per configurare i diversi pin e generare in automatico il file .dtb (Device Tree Blob) con l'associazione delle funzioni desiderate ai pin fisici. Il file .dtb è letto dal kernel in fase di boot per leggere ed applicare la configurazione iniziale del sistema, associando i pin del SoC ai pin fisici del GPIO e caricando i relativi driver. In Raspberry Pi la configurazione è pressoché predefinita e dove è possibile si interviene con lo strumento *raspi-config* o al più con i file di configurazione in formato

testo nella partizione di boot e nella cartella */etc*. Nell'articolo su Banana Pi, in questo stesso numero della rivista, è richiesto un processo che prevede la decompilazione e la ricompilazione del file script.bin, oltre che di un'accorta configurazione di driver. La scheda Arietta G25 è una via di mezzo: è sì necessario configurare un file di descrizione dell'I/O, ma i progettisti hanno messo a disposizione sul loro sito uno strumento grafico provvisto di interfaccia web che permette di configurare i pin di I/O della scheda nel modo desiderato e poi ottenere e scaricare il file di configurazione da inserire nella partizione di boot della scheda. Nel sito sono presenti istruzioni su come ricompilare il kernel per includervi driver specifici, ma anche immagini precompilate da utilizzare direttamente. In un percorso formativo ideale, per chi vuole iniziare a conoscere e poi approfondire il mondo GNU/Linux embedded, consigliamo di iniziare con Raspberry Pi e poi, acquisite le conoscenze

di base, proseguire con la scheda Arietta G25 per entrare più in profondità nelle "segrete cose" di GNU/Linux e del mondo embedded.

Come vedremo, per utilizzare Arietta G25 è praticamente necessario utilizzare il saldatore, le breadboard e progettare "schede di espansione" a seconda dei progetti che si vogliono realizzare. Cosa che stiamo pensando di realizzare in un prossimo futuro. A seguire, ed a seconda delle esigenze, si può passare ad utilizzare la famiglia professionale delle schede ACME Systems, oppure dirigersi verso schede più "application oriented" come la Banana Pi e le schede di potenza superiore come la UDOO.

Il vantaggio di tutte queste schede è di utilizzare praticamente lo stesso sistema operativo, e di conseguenza, pressoché tutto ciò che si impara e si realizza, viene capitalizzato e si può trasferire da un sistema all'altro con poca o nessuna fatica.

### CARATTERISTICHE DI ARIETTA G25

In Fig. 1 vediamo lo schema d'insieme della scheda e l'indicazione dei principali componenti, connettori e componenti aggiuntivi. In Fig. 2, la vista della scheda così come esce dalla sua confezione.

Elenchiamo le caratteristiche principali della scheda:

- CPU Atmel AT91SAM9G25 SoC (ARM9 a 400 MHz);
- RAM DDR2 disponibile in due versioni (128 o 256 MByte);
- alimentazione: singola a 3,3 Vcc;
- livelli digitali: TTL 3,3V (non tollera i livelli a 5V);
- PCB a 8 layer;
- LED e pulsante configurabili dall'utente;
- disponibilità garantita: > 5 anni;

- progettata e realizzata in Italia.

### Connettore J4

Si tratta di una fila di piazzole dorate predisposte per accogliere un connettore THT da due file di 20 pin di passo 2,54 mm (100 mil). In alcuni casi più funzioni insistono sugli stessi pin che devono essere configurati con il processo guidato via web, nel file "acme-arietta.dtb", nella partizione di root. Di conseguenza, le funzioni descritte di seguito non sono tutte disponibili contemporaneamente:

- 1 USB host;
- 1 USB host high-speed (condivide le linee della porta microUSB);
- 1 USB host hi-speed (condivide le linee dell'adattatore WiFi);
- 3 linee UART;
- 2 bus;
- bus SPI: 1 bus con la possibilità di selezionare fino a 3 chip (da 5 a 50 MHz);
- 4 uscite PWM;
- convertitore ADC a 4 canali con 10 bit di risoluzione (1024 "passi");
- interfaccia audio SSC/I2S;
- non sono presenti uscite video o per display TFT ma è possibile utilizzare il bus SPI per piccoli display TFT e il bus I2C per piccoli display LCD alfanumerici;
- non è presente un'interfaccia LAN diretta ma le porte USB

possono essere usate per collegare un adattatore USB/LAN.

### Connettore DP

È una fila di piazzole predisposta per accogliere un connettore maschio THT da 6 pin di passo 2,54 mm (100 mil) per collegare il convertitore USB/Seriale DPI (Debug Port Interface).

### Connettore WiFi

Piazzole predisposte per accogliere un connettore femmina da 7 pin di passo 2 mm per collegare l'adattatore WiFi IEEE802.11 b/g/n.

### Alloggiamento per micro SD

Accetta SD-Card fino a 64GB di capacità con la configurazione GNU/Linux (bootable).

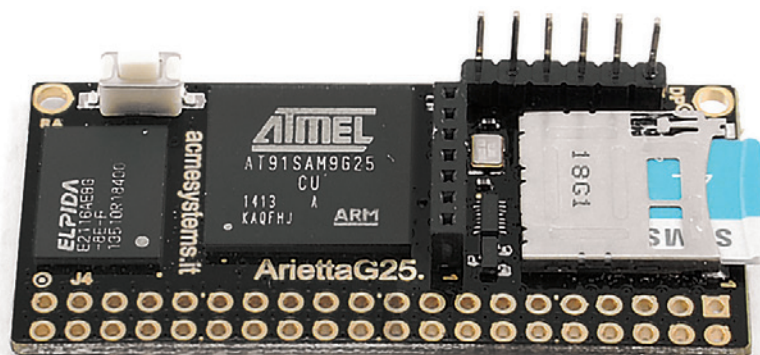
### Connector microUSB

Permette di collegare la scheda Arietta G25 al PC ed accedere al sistema. La porta si presenta come un emulatore LAN su USB (USB gadget).

### Caratteristiche software

- versione Kernel Linux 3.14.7;
- distribuzione: EmDebian Grip Linux "Wheezy" 7.4;
- programmabile utilizzando pressoché ogni linguaggio: C, C++, Python, Java, PHP, Perl, Node JS, Lua, Basic, Shell;
- GNU C, Python, Perl and PHP preinstallati nella distribuzione su microSD per un utilizzo immediato;

Fig. 3





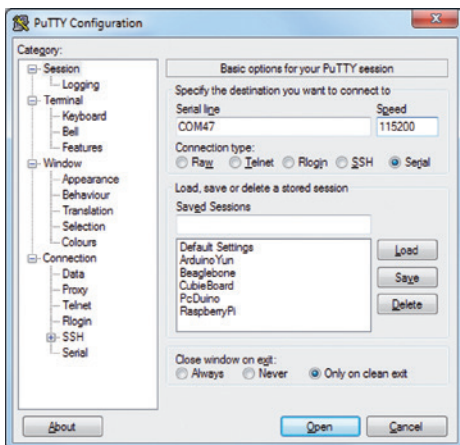


Fig. 4

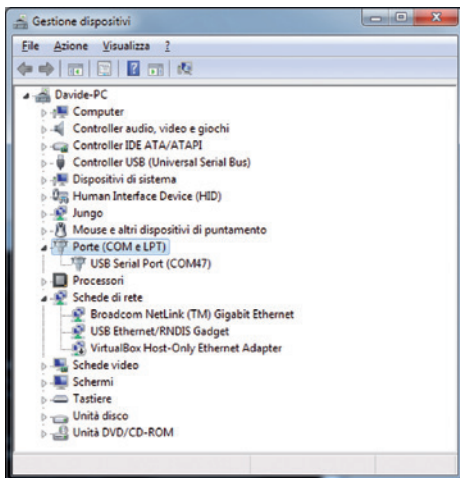


Fig. 5

- IDE Codiad con interfaccia web per editare i progetti ed i programmi direttamente via Browser;
- emulatore di terminale con interfaccia web "Shell-in-a-box" per la gestione remota del sistema da linea di comando

### Documentazione e Strumenti

- micro SD Card con GNU/Linux preinstallato disponibili in vendita presso Acme Systems eShop ed immagini scaricabili liberamente dal sito Acme binary repository per realizzare le proprie micro SD Card in autonomia;
- guide complete per compilare la propria distribuzione a partire dai sorgenti;
- firmware totalmente Open

Source, ambienti software e toolchain di sviluppo scaricabili liberamente;

- tutorial ed esempi di codice scaricabili liberamente;
- migliaia di pacchetti Open Source scaricabili liberamente dal repository Debian Wheezy for Armel.

### PREPARIAMO ARIETTA G25

Abbiamo anticipato che per lavorare con la scheda serve il saldatore. Per il primo utilizzo raccomandiamo di usare la scheda con il suo adattatore WiFi ed il convertitore FTDI per accedere alla console di sistema. I due piccoli adattatori possono essere saldati direttamente sulla scheda oppure montati su uno strip a 7 poli femmina passo 2 mm per l'adattatore WiFi ed uno strip a 6 poli maschi, passo 2,54 mm, per l'interfaccia seriale.

Il lavoro finito è visibile in Fig. 3. Noi abbiamo scelto l'adattatore WiFi con l'attacco per l'antenna esterna. E quindi montiamo il cavetto pigtail e l'antenna sul modulo. Poi alloggiare entrambi i moduli sui loro connettori. Se non pensiamo di utilizzare alcun pin del GPIO la scheda è pronta. Altrimenti saldiamo un connettore a 40 pin per il GPIO. Sono possibili diverse alternative, utilizzando connettori dritti, a 90 gradi maschi o femmine a seconda delle esigenze. Infatti la scheda può essere utilizzata stand alone, oppure posizionata su un'ulteriore scheda di supporto in posizione orizzontale o verticale. Ora dobbiamo alimentare la scheda e connetterci alla stessa per le personalizzazioni. Anche in questo caso vi sono diverse possibilità, in quanto la scheda è molto versatile. Si può utilizzare

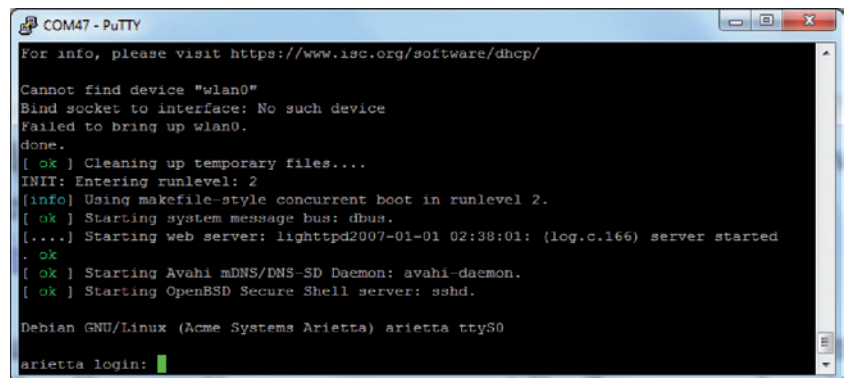


Fig. 6

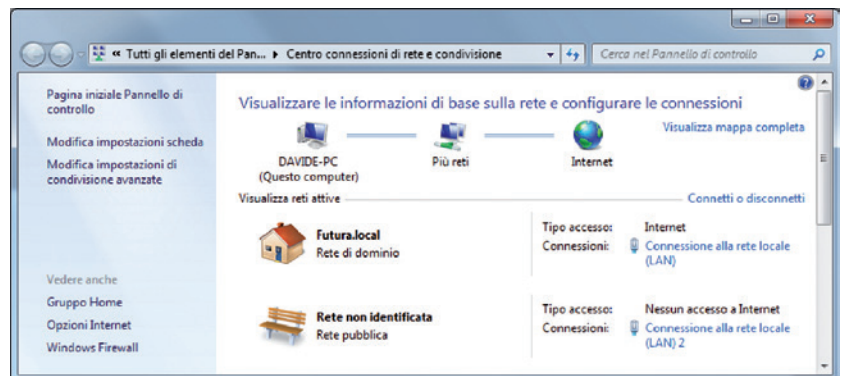


Fig. 7



la connessione via "console seriale", oppure collegarsi via porta micro USB "Device Port", nel qual caso la scheda viene "vista" come un adattatore LAN. Possiamo collegarci in WiFi utilizzando il modulo di estensione che abbiamo citato od infine utilizzando una connessione via cavo ethernet cablando una porta USB al GPIO della scheda ed utilizzando un adattatore LAN/USB. Quest'ultima soluzione richiede la verifica di compatibilità dell'eventuale adattatore con i driver previsti. In caso di incompatibilità è necessario attrezzarsi per ricompilare il kernel. Ma questa è un'altra storia, che affronteremo in seguito. Per un primo utilizzo "esplorativo" abbiamo comunque a disposizione diverse possibilità. Per iniziare colleghiamo il cavo con la presa miniUSB al modulo seriale DPI. Apriamo un

terminale PuTTY sul nostro PC e lo configuriamo come visibile in Fig. 4. Configuriamo il campo "Connection type" su "Serial". In *Serial line* inseriamo il numero di porta COM assegnata al convertitore DPI. Impostiamo "Speed" al valore "115200". Per individuare quale porta COM è stata assegnata alla scheda apriamo (Windows 7) "Pannello di controllo>>Sistema>>Gestione Dispositivi" ed espandiamo la voce "Porte (COM e LPT) come visibile in Fig. 5. Nel nostro caso la porta assegnata è la COM47. Torniamo alla finestra di configurazione di PuTTY e clicchiamo sul pulsante "Open". Si apre la finestra di comunicazione seriale, al momento inattiva. Premiamo due volte Invio e vedremo lampeggiare il LED rosso sul modulo DPI. Ora colleghiamo il cavo con la presa microUSB al con-



Fig. 10

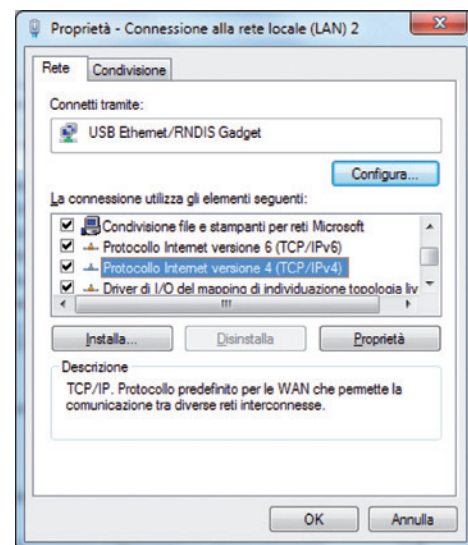


Fig. 8

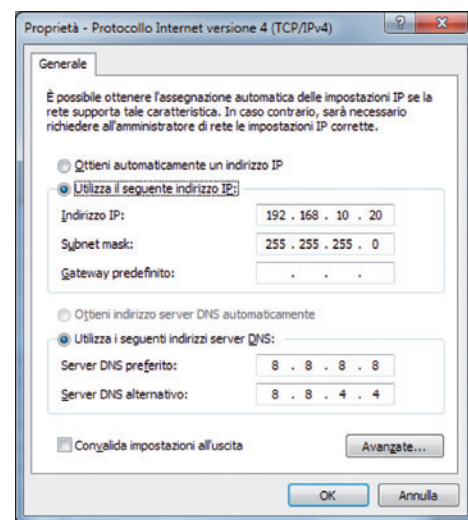
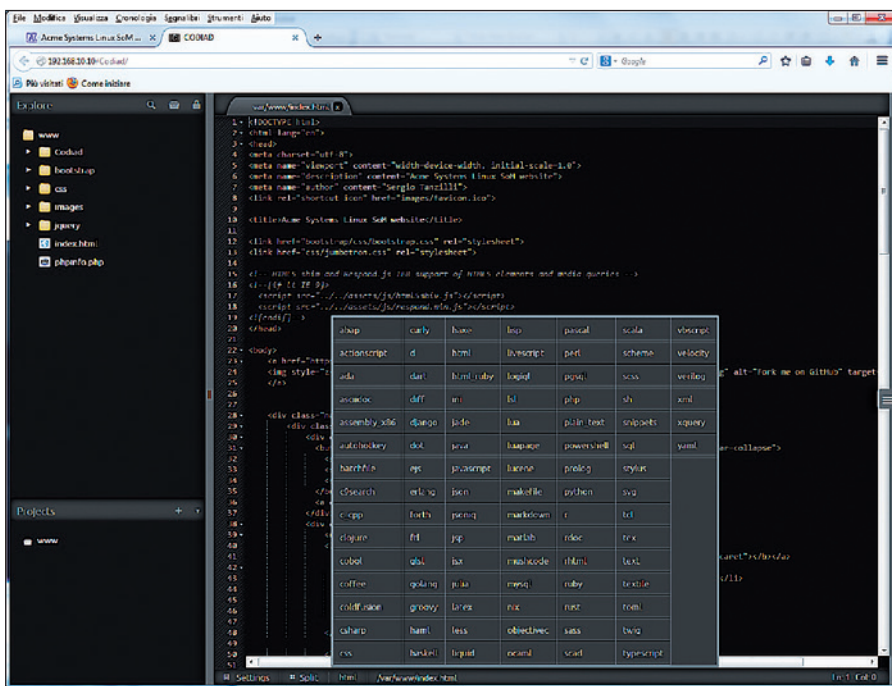


Fig. 9

nettore microUSB "Device Port" e dall'altro lato ad un'ulteriore porta USB del PC. Arietta G25 inizierà il boot e potremo vedere i messaggi a console, fino alla richiesta di login come visibile in Fig. 6.

Nel frattempo, essendo la prima volta che colleghiamo Arietta G25 via USB, verrà installato sul PC il driver per l'adattatore RNDIS/Ethernet Gadget. Se utilizzate un PC con XP o comunque il driver non viene reperito automaticamente, lo potete scaricare dall'indirizzo <http://www.driverscape.com/download/rndis-ethernet-gadget> ed installarlo manualmente.



Scegliete la versione adatta al vostro sistema. Scompattatela in una cartella creata appositamente ed eseguite l'aggiornamento del driver puntando alla cartella appena creata. Alla fine, se andate in "Impostazioni>>Connessioni di rete", vedrete apparire una nuova connessione LAN (**Fig. 6**). Cliccate col tasto destro del mouse sulla nuova scheda e selezionate "Proprietà" dal menu a tendina (**Fig. 7**). Qui impostate l'indirizzo IP della scheda e la subnet mask con i seguenti valori (**Fig. 7**):

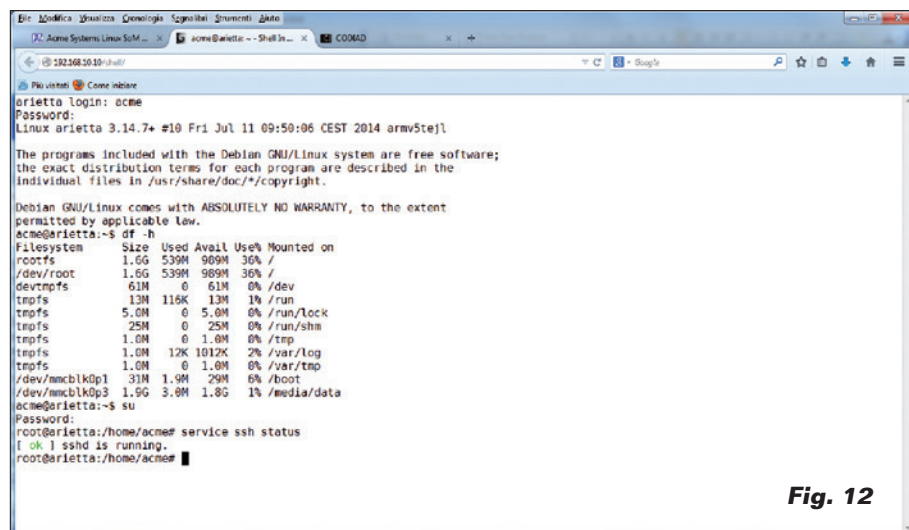
IP 192.168.10.20  
Subnet Mask: 255.255.255.0

Date tutte le conferme necessarie e richiudete. Aprite un browser e digitate l'indirizzo 192.168.20.10, che è l'indirizzo predefinito della scheda. Vi apparirà la pagina di **Fig. 10** a conferma che tutto funziona a dovere. Già questa pagina ci conferma che sulla scheda è attivo un server web. Si tratta di "lighttpd", una versione "leggera" di web server e sono presenti un paio di applicazioni

interessanti: “Codiad Web IDE” e “Linux Command Prompt”. Le applicazioni sono accessibili dalla pagina iniziale appena aperta. Le userid e le password da utilizzare sono indicate nella pagina stessa. Più facile di così. La prima è un IDE di sviluppo e gestione di progetti multi linguaggio che sarà molto utile nella realizzazione dei nostri progetti. Per accedervi clicchiamo sul pulsante blu “Codiad Web IDE”. Nella pagina successiva

inseriamo la *userid* e la *password* indicate in modo da ottenere la pagina visibile in **Fig. 11**. Qui vediamo che la struttura è quella tipica di un IDE: in alto a sinistra c'è la finestra di *Explore* che permette di navigare tra le sotto-cartelle e i file che appartengono alla cartella *www*. La finestra in basso a sinistra ci permette di gestire i progetti mentre la finestra grande a destra permette di editare i singoli programmi. È possibile evidenziare la sintassi delle istruzioni impostando il corretto linguaggio scegliendolo nella ricca lista che si apre cliccando sul menu del linguaggio nella barra in basso. Inoltre è possibile dividere la finestra in più parti ed eseguire il merge dei contenuti tra le due finestre.

La seconda applicazione è un terminale remoto via web; ad essa è necessario accedere con l'utente generico: "acme" nel nostro caso. Una volta entrati, abbiamo a disposizione una finestra di terminale utilizzabile da remoto, come visibile in **Fig. 12**. Come esempio abbiamo impartito il comando che ci permette di verificare l'occupazione delle partizioni:

 $df-h$ 

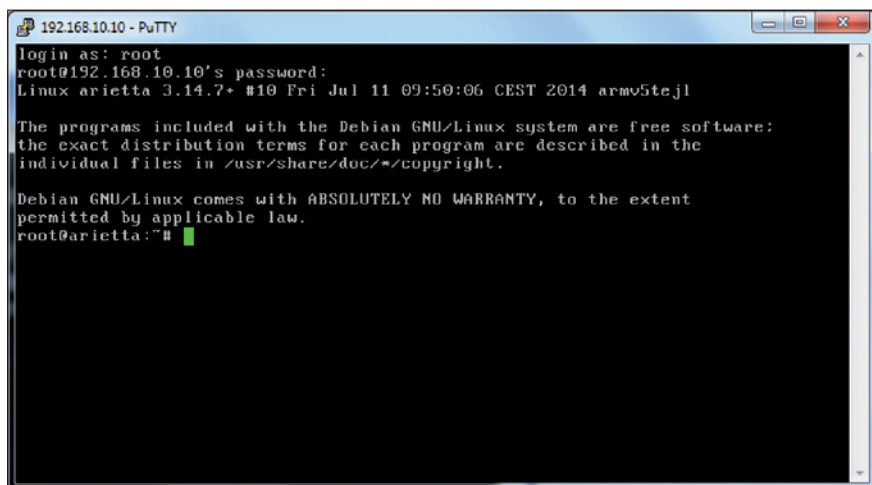


Fig. 13

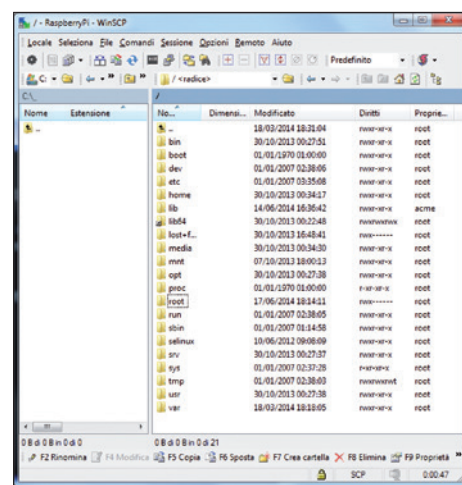


Fig. 14

I caratteri “-” e “+” non devono essere digitati utilizzando la tastiera normale, bensì con il tastierino numerico, oppure, se utilizzate un portatile come è nel nostro caso, ottenuti con le combinazioni di tasti che emulano il tastierino numerico; nel nostro caso, premendo in contemporanea i tasti “Fn” ed il “-” associato alla matrice “Fn”. Per accedere come utente “root” è necessario digitare il comando “superuser”:

su

e poi fornire la password di root (“acmesystems”). Queste due applicazioni ci permettono di gestire praticamente l’intera scheda da remoto. Continuiamo ad analizzare le diverse possibilità di collegamento alla scheda, via server SSH e via console seriale con il modulo DPI. Controlliamo la presenza e il funzionamento del server SSH con il comando (Fig. 12):

*service ssh status*

La risposta ci conferma che nella configurazione predefinita il server è già installato e viene attivato al boot. Ci colleghiamo utilizzando il terminale remoto “PuTTY” e l’applicazione di File Manager remota “WinSCP”. Per la descrizione, l’installazione e

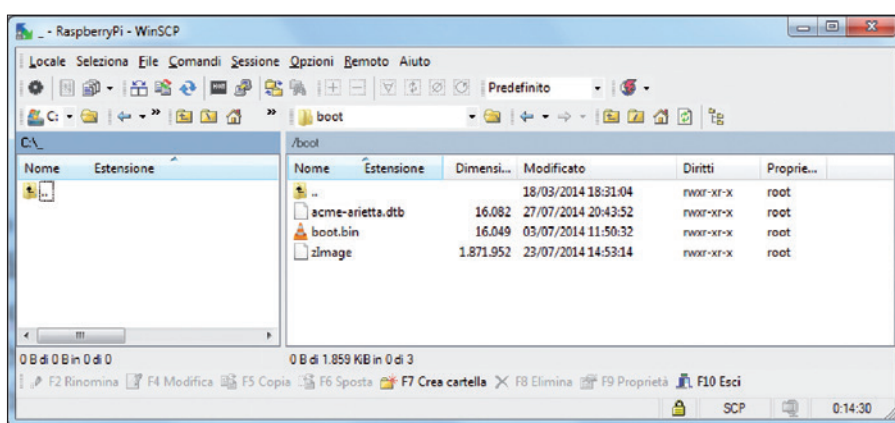


Fig. 15

l’utilizzo di questi strumenti vi rimandiamo al libro “Raspberry Pi-II mio primo Linux embedded” in quanto il sistema operativo di Arietta G25 è praticamente il medesimo utilizzato con Raspberry Pi. Per connetterci con entrambi gli applicativi utilizziamo l’indirizzo della scheda 192.168.10.10. Eseguiamo i login con userid e password ed otteniamo i risultati visibili nelle Fig. 13 (PuTTY) e Fig. 14 (WinSCP). Approfittiamo del file manager per anticipare un argomento che ci sarà indispensabile per le configurazioni e personalizzazioni successive. Andiamo alla cartella/boot che “punta” alla partizione di boot e viene montata automaticamente all’avvio della scheda e verifichiamo il contenuto (Fig. 15). Due file sono particolarmente importanti. Il file zImage è il file

immagine del kernel di Linux. È il risultato degli eventuali processi di compilazione per personalizzare la configurazione del sistema operativo in base alle nostre esigenze. Nel sito di Acme Systems sono disponibili dei kernel già compilati per particolari esigenze, come nel caso volessimo attivare il convertitore analogico digitale. Il file “acme-arietta.dtb” contiene la configurazione dei pin ottenuta dal configuratore “web” messo a disposizione sul sito di Acme Systems. Di questo argomento parleremo nel seguito dell’articolo. Ora torniamo a un momento alla finestra dell’emulatore di terminale seriale collegato alla “console” di sistema. Possiamo utilizzare questa finestra come una finestra terminale, con qualche limitazione che vedremo caso



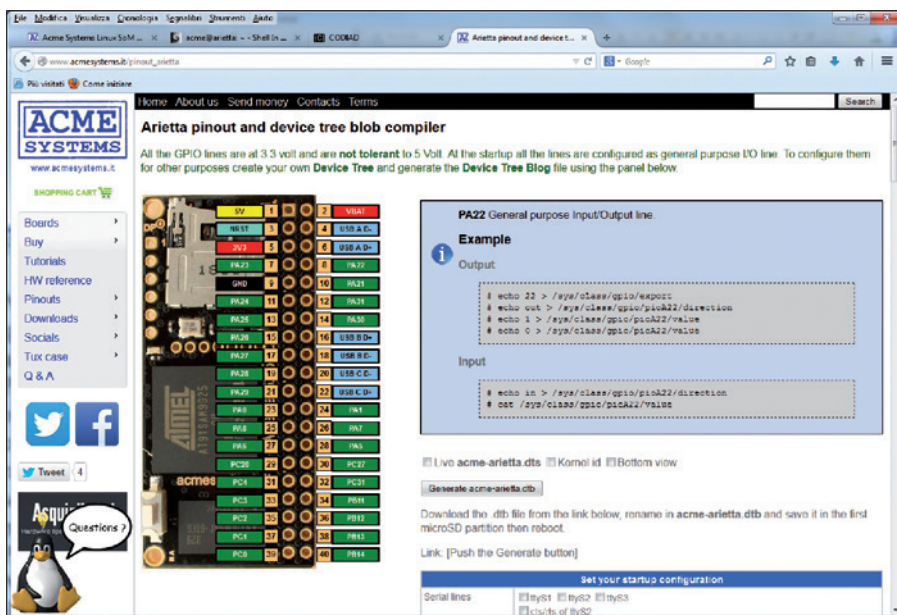


Fig. 16

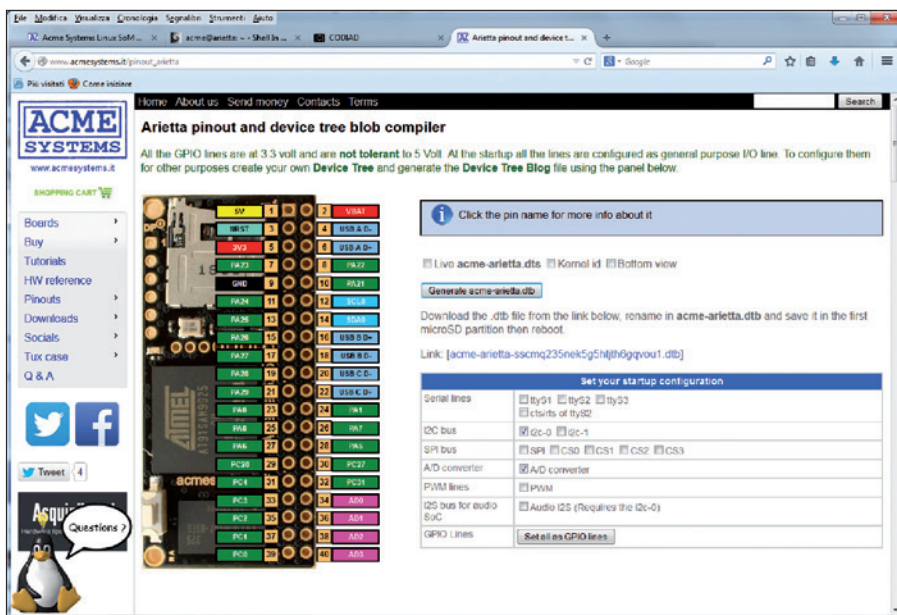


Fig. 17

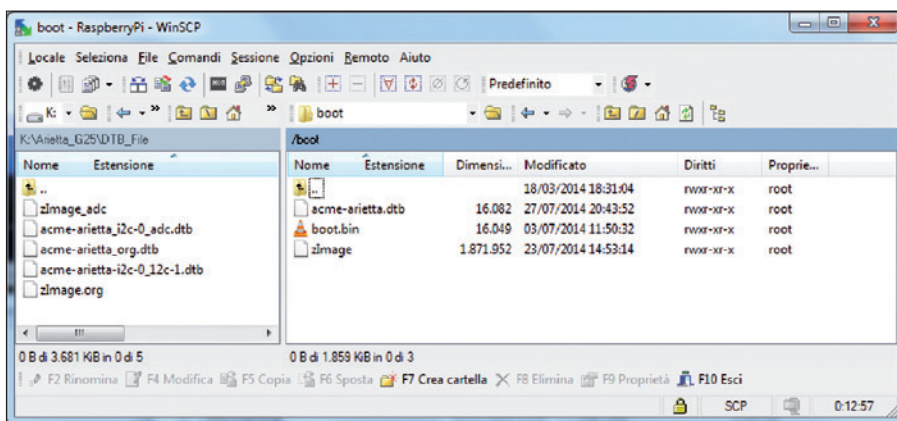


Fig. 18

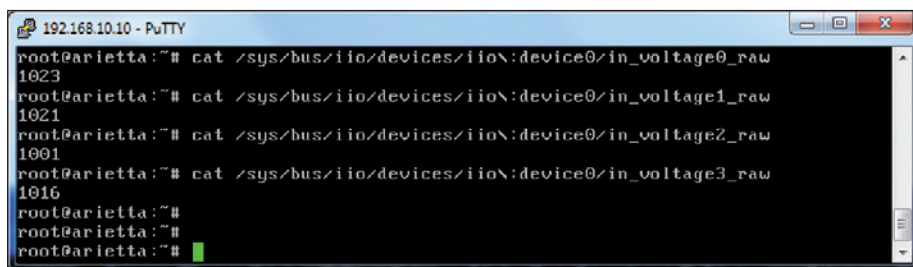
per caso. L' utilità principale della console è di presentarci tutti i messaggi emessi in fase di boot e di shutdown del sistema. Dopo il login, come userid "root" e password "acmesystems" provate a dare un comando di:

*reboot*

Vedrete che la console non si disconnette, come fanno le connessioni via server SSH. Per vedere i messaggi di boot dall'inizio, basta stabilire la connessione prima di alimentare la scheda dal connettore micro USB. Ora approfondiamo la gestione del GPIO. Per conoscere il significato dei pin GPIO di Arietta G525 lo strumento migliore è il configuratore web visibile in Fig. 16.

Nella versione di base vediamo la configurazione predefinita dei pin GPIO. Cliccando su uno qualsiasi dei pin, in alto a destra nella pagina vediamo la descrizione della funzione del pin. Per alcuni pin sono anche indicate le istruzioni per leggere e modificare il livello del pin. Per i pin di I/O il metodo di accesso è quello valido anche per Raspberry Pi, descritto nel libro citato, ovviamente rispettando i numeri di GPIO indicati nello schema. Se vogliamo personalizzare la configurazione dei pin possiamo usare la sezione di configurazione presente a destra della rappresentazione del GPIO, sotto la configurazione dei pin. La configurazione si esegue selezionando con un segno di spunta le funzioni che si vogliono attivare. Se fate qualche prova potete verificare che alcune funzioni condividono gli stessi pin come il convertitore ADC ed il PWM, che condividono i pin 34, 36, 38 e 40. Pertanto possiamo scegliere una sola delle opzioni. Non è ammessa la configurazione mista di questi pin, alcuni come ADC ed





**Fig. 19**

altri come PWM. Un altro caso è il bus I<sup>2</sup>S, che richiede la attivazione contestuale del bus I<sup>2</sup>C. Nel nostro test abbiamo optato per attivare il bus I<sup>2</sup>C-0 e il convertitore ADC (Fig. 17).

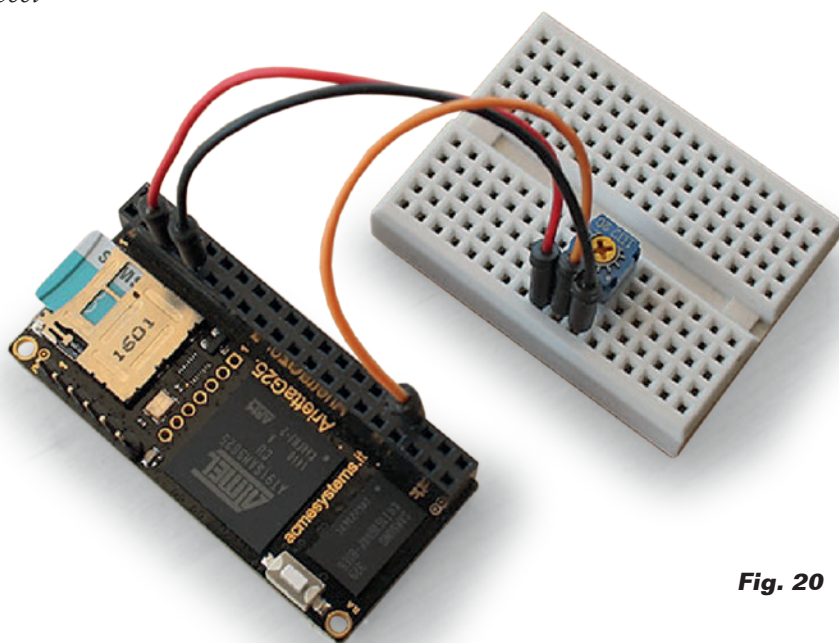
Ottenuta la configurazione che desideriamo, chiediamo la preparazione del file .dtb personalizzato, cliccando sul pulsante "Generate acme-arietta.dtb". Dopo la creazione del file, ci viene proposto un link per scaricare il file sul nostro PC. Vi invitiamo a controllare la sezione "Tutorial" del sito per verificare se sono necessarie operazioni aggiuntive. Nella sottosezione "Using the A/D converter lines" troviamo una sorpresa. Nella immagine Linux distribuita con Arietta G25 non è presente il driver Atmel ADC così è necessario abilitarlo utilizzando la funzione "menu-config" (vedere l'appendice al libro citato) e ricompilare il Kernel Linux. Nel tutorial è presente un link dove scaricare un kernel già compilato, che comprende il driver richiesto: lo scarichiamo in una cartella del nostro PC. Ora dobbiamo aggiornare la partizione di boot della scheda con i due file che abbiamo scaricato. Noi abbiamo deciso di utilizzare lo strumento WinSCP che ci permette di spostare e sostituire i file tra il file system della scheda e quello del PC, come visibile in Fig. 18. Prima creiamo una cartella di backup sul PC, spostiamo i file originali zImage e acme-arietta.dtb nella cartella di backup. Rinominiamo il file di configurazione, che abbiamo scaricato dal confi-

guratore web, in "acme-arietta.dtb". Spostiamo i file "zImage" e "acme-arietta.dtb" nella cartella boot. Controlliamo bene i nomi dei file e la loro presenza nella cartella. Ora andiamo alla finestra con la console e digitiamo il comando:

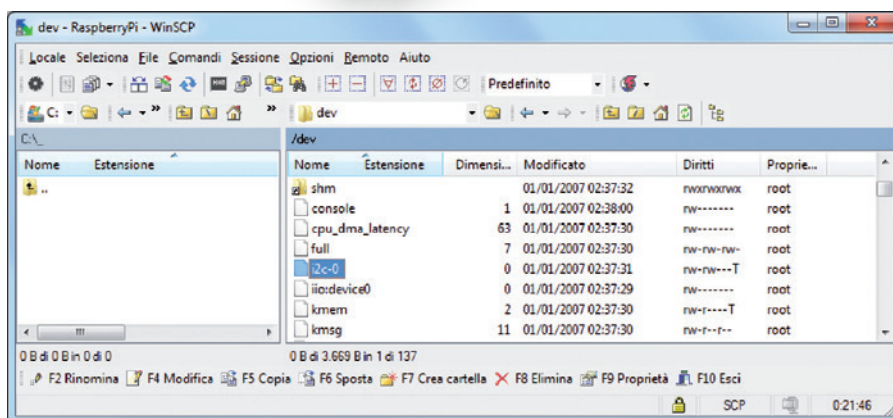
*reboot*

Una volta terminato il processo di boot, riconnettiamo la finestra di terminale aperta con PuTTY e verificiamo il funzionamento del convertitore ADC impartendo i seguenti comandi (Fig. 19):

```
root@arietta:~# cat /sys/bus/iio/devices/iio:device0/in_voltage0_raw
1023
root@arietta:~# cat /sys/bus/iio/devices/iio:device0/in_voltage1_raw
1022
root@arietta:~# cat /sys/bus/iio/devices/iio:device0/in_voltage2_raw
1001
root@arietta:~# cat /sys/bus/iio/devices/iio:device0/in_voltage3_raw
```



**Fig. 20**



**Fig. 21**

```

/etc/network/interfaces - RaspberryPi
# interfaces(5) file used by ifup(8) and ifdown(8)
auto lo
iface lo inet loopback
pre-up modprobe g_ether
auto usb0
iface usb0 inet static
address 192.168.10.10
netmask 255.255.255.0
gateway 192.168.10.20

auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
# wpa-scan-ssid 1
# wpa-ap-scan 1
# wpa-key-mgmt WPA-PSK
# wpa-proto RSN WPA
# wpa-pairwise CCMP TKIP
# wpa-group CCMP TKIP
wpa-ssid "<vostro_SSID>"
wpa-psk "<vostro_password>"

```

Fig. 22

```

1017
root@arietta:~#

```

Possiamo anche collegare un trimmer da 4,7 kohm nel modo indicato in Fig. 20 e provare ad acquisire i valori digitali in corrispondenza di diverse posizioni del cursore del trimmer, ovviamente utilizzando il comando relativo al canale ADC al quale abbiamo collegato il trimmer, il primo nel nostro caso. Ricordiamo sempre di non superare i 3,3V su ciascun pin, pena la distruzione della scheda (gli ingressi del SoC AT91 non tollerano tensioni superiori ai 3,3V). Vediamo ora il bus I<sup>2</sup>C che abbiamo attivato. Una prima prova della sua presenza l'abbiamo andando a verificare la presenza del file di device in /dev. Effettivamente troviamo il file i2c-0, ad indicare che l'operazione di configurazione è andata a buon fine (Fig. 21). Per gestire e controllare al meglio il bus I<sup>2</sup>C è opportuno installare il pacchetto i2c-tools, che fa parte del repository della distribuzione. Per far questo dobbiamo collegare la scheda Arietta G25 alla nostra rete internet.

```

192.168.10.10 - PuTTY
root@arietta:~# apt-get install i2c-tools
Reading package lists... 0%
Reading package lists... 0%

Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 22 not upgraded.
root@arietta:~#
root@arietta:~#
root@arietta:~#
root@arietta:~#

```

Fig. 23

```

192.168.10.10 - PuTTY
root@arietta:~# i2cdetect -y 0
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
10: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
20: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
30: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
40: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
50: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
60: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
70: -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
root@arietta:~#
root@arietta:~#
root@arietta:~#

```

Fig. 24

Al momento possiamo utilizzare la connessione WiFi. Abbiamo già collegato il modulo alla scheda, quindi per farlo funzionare con la nostra rete non ci resta che configurare il file /etc/network/interfaces. In Fig. 22 è visibile un fac-simile del file. In realtà i parametri indispensabili da configurare sono quelli relativi all'SSID, al tipo di protezione ed alla password. Per una connessione con protezione WPA potete utilizzare l'esempio indicato, ovviamente inserendo i parametri relativi alla vostra rete.

```

auto wlan0
allow-hotplug wlan0
iface wlan0 inet dhcp
# wpa-scan-ssid 1
# wpa-ap-scan 1
# wpa-key-mgmt WPA-PSK
# wpa-proto RSN WPA
# wpa-pairwise CCMP TKIP
# wpa-group CCMP TKIP
wpa-ssid "<Vostro_SSID>"
wpa-psk "<Vostar_PSK>"

```

Questa configurazione è valida per una rete protetta con chiave WPA. Per una rete protetta con chiave WPA2 potete fare riferimento all'articolo "Internet radio con Raspberry Pi e shield LCD" nel numero 184 di Elettronica In ed agli esempi di configurazione

reperibili all'indirizzo [http://www.lsi.upc.edu/lcsi/Manuales/wireless/files/wpa\\_supplicant.conf](http://www.lsi.upc.edu/lcsi/Manuales/wireless/files/wpa_supplicant.conf).

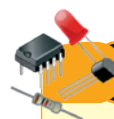
Una volta collegati alla rete locale, possiamo installare il pacchetto i2c-tools, o tutti i pacchetti che desideriamo.

Aggiorniamo il data base dei pacchetti, e la distribuzione stessa, con i comandi:

```

apt-get update
apt-get upgrade

```



## per il MATERIALE

Il modulo Arietta-G25 è disponibile sia con memoria da 128 MB che da 256 MB. Entrambe le versioni utilizzano una CPU Atmel AT91SAM9G25 con core ARM9 da 400 MHz. La versione a 128 MB (cod. Arietta-G25) costa 24,40 Euro mentre la versione a 256 MB (cod. Arietta-G25-256) costa 36,60 Euro. Tutti i prezzi si intendono IVA compresa.

Il materiale va richiesto a:  
 Futura Elettronica, Via Adige 11,  
 21013 Gallarate (VA)  
 Tel: 0331-799775 • Fax: 0331-792287  
<http://www.futurashop.it>



Poi installiamo il pacchetto con il comando (Fig. 23):

```
apt-get install i2c-tools
```

Infine verifichiamo il buon funzionamento del driver I<sup>2</sup>C con il comando:

```
i2cdetect -y 0
```

che ci darà il risultato visibile in Fig. 24. Solo un avvertimento: dovete dare il comando da una finestra di terminale aperta con PuTTY, oppure dalla finestra di terminale web "Shell in a box". Se date il comando dalla finestra di terminale seriale che fa capo alla console otterrete dei messaggi di errore. Non allarmatevi e ... cambiate finestra. Arrivati

a questo punto possiamo dire di aver fatto conoscenza con la scheda Arietta G25 e le sue principali modalità di accesso e di gestione. Ricordiamo che Arietta G25 è la "mascotte" di una famiglia di schede più potenti e destinate all'utilizzo in campo professionale, come la storica G20 e la più recente Aria G25. Hanno in comune lo stesso sistema operativo e le stesse modalità di configurazione ed utilizzo.

Il pregio di questa scheda è di permettere di accrescere le proprie competenze nel mondo GNU/Linux embedded in modo, tutto sommato, abbastanza facilitato dalle guide e dagli strumenti messi a disposizione del produttore Acme Systems, che rendono il percorso di apprendimento anche

divertente, senza le "frustrazioni" che spesso accompagnano chi si addentra nei meandri di GNU/Linux senza una guida adeguata. A maggior valore, l'esperienza fatta con la "piccola" Arietta G25 non andrà persa, e non andrà persa nemmeno se passiamo a sistemi più potenti dalla vocazione più "gestionale". Un'altra scheda che si è guadagnata a pieno titolo un posto fisso sulla nostra scrivania. Non mancheremo di utilizzare questa scheda, o le sorelle maggiori, in progetti dedicati e stiamo pensando anche ad un'altra sorpresa sul piano didattico. Per ora, se avete intenzione di basare il vostro futuro, anche solo in parte, puntando sull'evoluzione "digitale" del nostro mondo, non fatevi sfuggire questa occasione. ■

# PROTECH

## CONTROLLO TOTALE!

**PROTECH** è il centro di comando che offre molteplici soluzioni di automazione per ambienti domestici e piccole imprese, ma allo stesso tempo è in grado integrarsi perfettamente a sistemi complessi di antintrusione, videosorveglianza e domotica per uffici e fabbriche. **PROTECH** è versatile, funzionale e facile da installare; l'unità di controllo può essere collegata alla rete LAN/WiFi permettendo di gestire facilmente gli impianti anche da Smartphone e Tablet.

**€95,00**  
IVA inclusa

Accedi a tutte le informazioni tecniche sul prodotto scansionando questo QR-Code con il tuo Smartphone o collegandoti a [www.protechsystem.it](http://www.protechsystem.it)  
email: [info@protechsystem.it](mailto:info@protechsystem.it)



**PROTECH SYSTEM** Studio di progettazione elettronica, consulenza e sviluppo applicativi mobile.

# 3DRAG

## STAMPA CON IL CIOCCOLATO

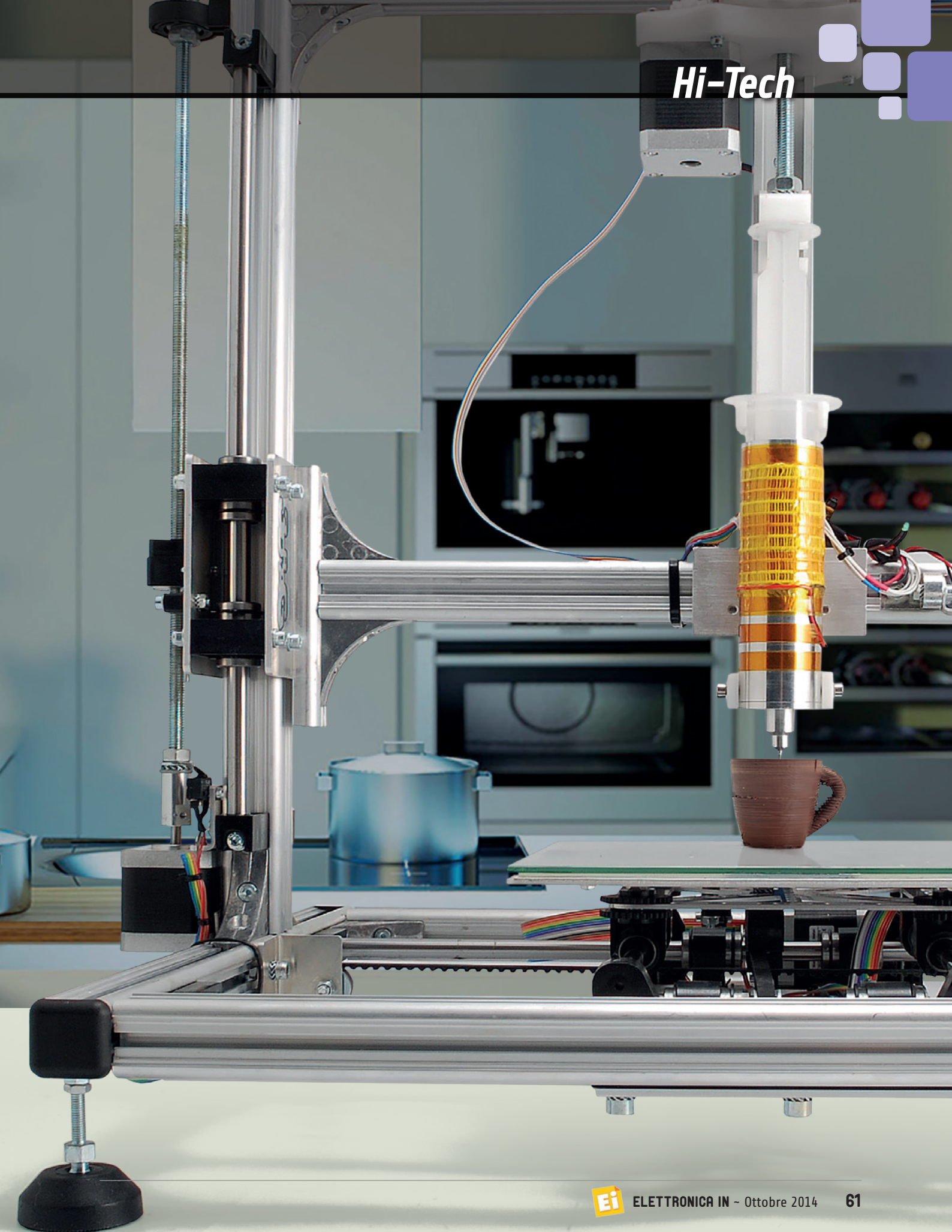
Sostituiamo la testa di stampa con una siringa motorizzata e un riscaldatore, e creiamo oggetti di ogni genere e dolci di qualsiasi forma: dalle uova di Pasqua a cioccolatini personalizzati, a scritte per torte e ogni altra cosa che la fantasia ci suggerisce.

a cura della REDAZIONE

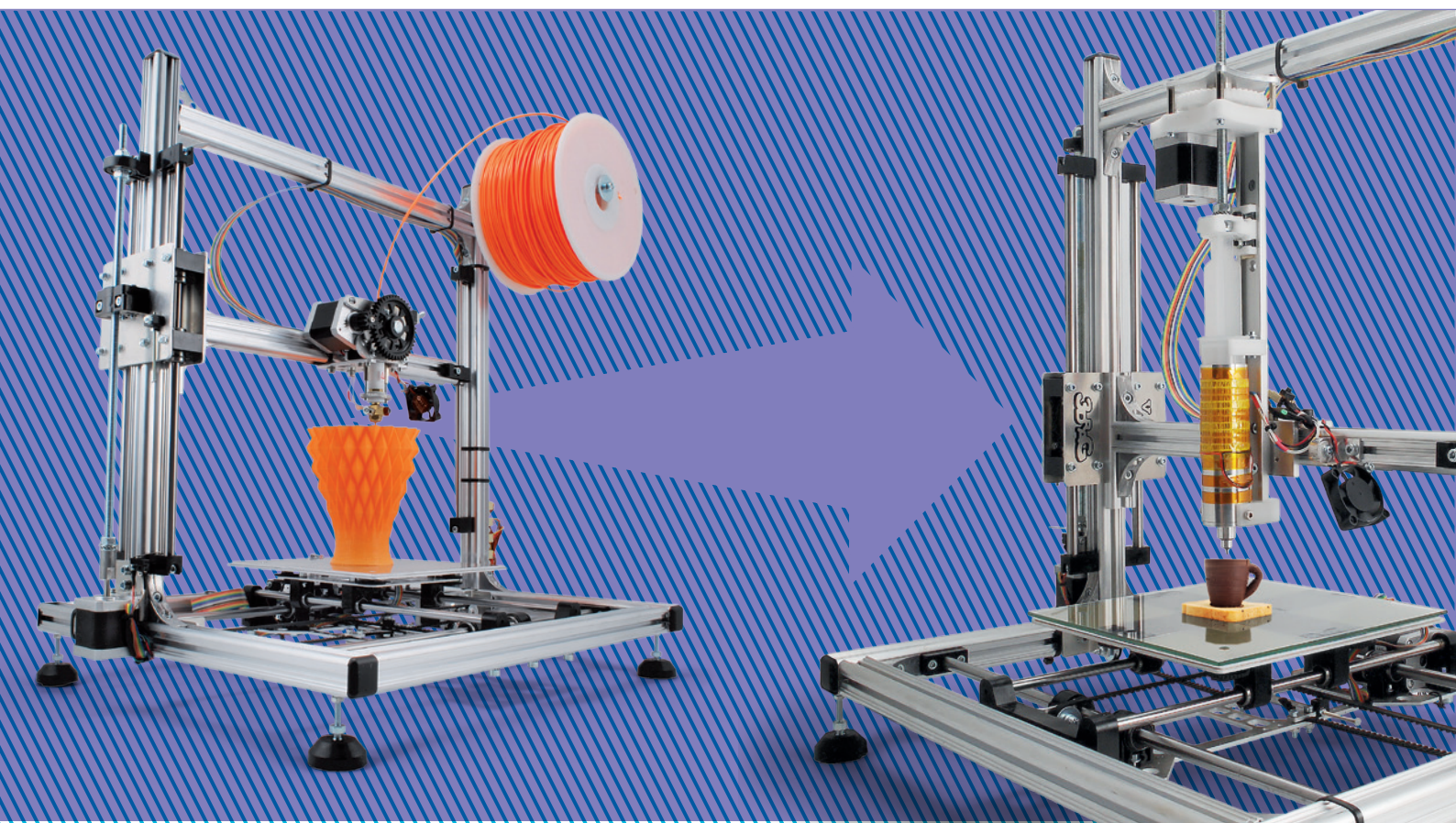
**M**aitre Chocolatier si nasce o si diventa? Probabilmente, se si tratta di creare originali miscele di cioccolato, ci si nasce, ma se quel che bisogna raggiungere è l'arte nella pasticceria, allora un po' tutti possiamo diventarlo: basta avere le macchine e gli attrezzi giusti. Uno di questi è, strano ma vero, la nostra stampante 3Drag, che opportunamente modificata permette di costituire un vero e proprio "sac a poche" tecnologico per lavori di precisione o una siringa riscaldata con cui tracciare scritte e linee col cioccolato di qualsiasi tipo: al latte, bianco o fondente. Il tutto, con il vantaggio di poter caricare in un computer il disegno o l'oggetto da realizzare e farlo preparare dalla stampante 3D.











In questo articolo spiegheremo come modificare la 3Drag per farle stampare oggetti con il cioccolato fuso; prossimamente, inoltre, pubblicheremo i progetti di due semplici accessori: lo scaldasiringhe (per mantenere calde le “ricariche di cioccolato” per averle pronte al momento di montarle nella macchina, così da lavorare più rapidamente) e il sistema di raffreddamento del cioccolato, di cui tra breve spiegheremo l'utilità. Sappiamo che qualcuno, leggendo queste prime righe, si starà chiedendo se stiamo parlando sul serio o meno, quindi è opportuno precisare qualcosa su questa “deriva culinaria” di un prodotto che di fatto nasce per la prototipazione rapida e la realizzazione tra le mura domestiche di oggetti di uso comune, plastici e simili. La 3Drag è una macchina di stampa tridimensionale basata sulla tecnologia a filamento fuso, ma la sua meccanica è tale da

prestarsi a depositare non solo plastica, ma qualsiasi altro materiale fuso purché plasmabile a temperature relativamente basse. L'idea di usare la stampa 3D per costruire alimenti non è nuova (tanto per intendersi, ci lavora la NASA allo scopo di avere in orbita macchine capaci di preparare cibi partendo da materie prime selezionate e aggiungendo eccipienti per conferire gusto e aspetto voluti) e nasce da un concetto alternativo di cucina e di gastronomia. Sostituendo la testa di stampa con un qualcosa in grado di iniettare in un estrusore un alimento plasmabile, come panna, cioccolato, gelatine, ma anche impasti, consente di realizzare dolci e prodotti salati con decorazioni e forme anche particolarmente impegnativi. Da qui l'idea di utilizzare il cioccolato, che è facilmente plasmabile, permette di realizzare oggetti abbastanza resistenti e simpatici,

ma anche di decorare con scritte e forme, senza comportare sollecitazioni tali da deformare o rendere imprecisa la meccanica della 3Drag. Più difficile sarebbe deporre materiali come gli impasti a base di farina o cose del genere, perché normalmente sono piuttosto densi e la relativa lavorazione richiede una certa potenza.

### LAVORARE COL CIOCCOLATO

Per poter utilizzare il cioccolato allo scopo di preparare decori, salse con cui cospargere le torte o per fare dei cioccolatini di forma varia, occorre fonderlo e a volte temperarlo. Fondere il cioccolato significa scioglierlo, farlo diventare plasmabile, mentre tempe-



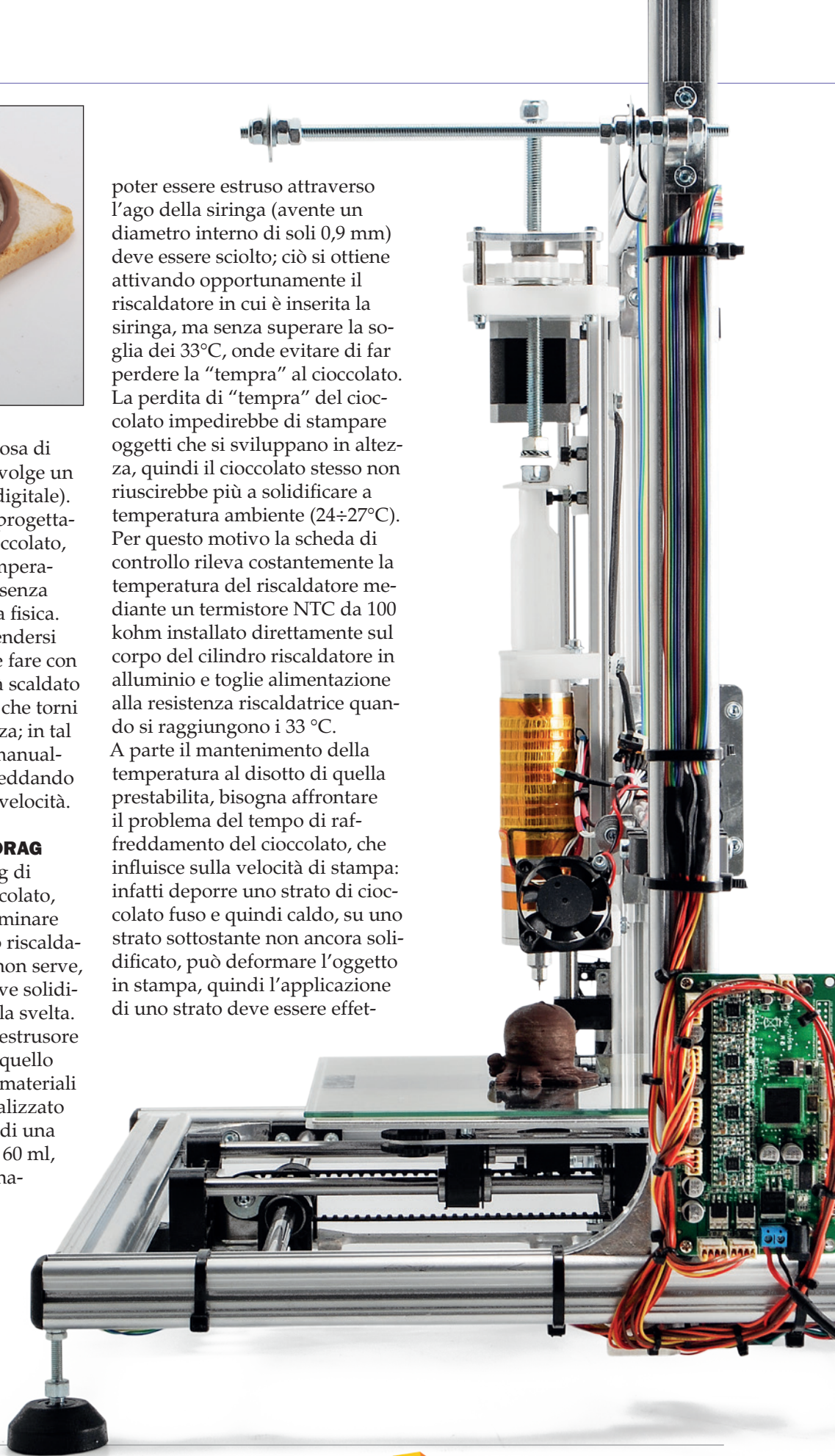
rare il cioccolato è qualcosa di più complesso, che coinvolge un termometro (meglio se digitale). La nostra 3Drag è stata progettata solo per fondere il cioccolato, mantenendolo a una temperatura tale da fondere ma senza alterarne la sua struttura fisica. La tempra può invece rendersi necessaria se avete a che fare con cioccolato che è stato già scaldato eccessivamente e volete che torni con la propria consistenza; in tal caso dovete procedere manualmente, fondendo e raffreddando il cioccolato a una certa velocità.

#### LE MODIFICHE ALLA 3DRAG

Per consentire alla 3Drag di stampare oggetti in cioccolato, bisogna innanzitutto eliminare dalla stampante il piatto riscaldato, che ovviamente qui non serve, dato che il cioccolato deve solidificare e possibilmente alla svelta. Poi occorre montare un estrusore speciale (da sostituire a quello utilizzato per estrudere materiali plastici) che abbiamo realizzato e appositamente dotato di una comunissima siringa da 60 ml, il cui pistone viene azionato mediante un motore passo NEMA17, oltre che di un riscaldatore cilindrico per mantenere ad una temperatura di circa  $32\div 33\text{ }^{\circ}\text{C}$  il cioccolato contenuto nella siringa. Il cioccolato, per

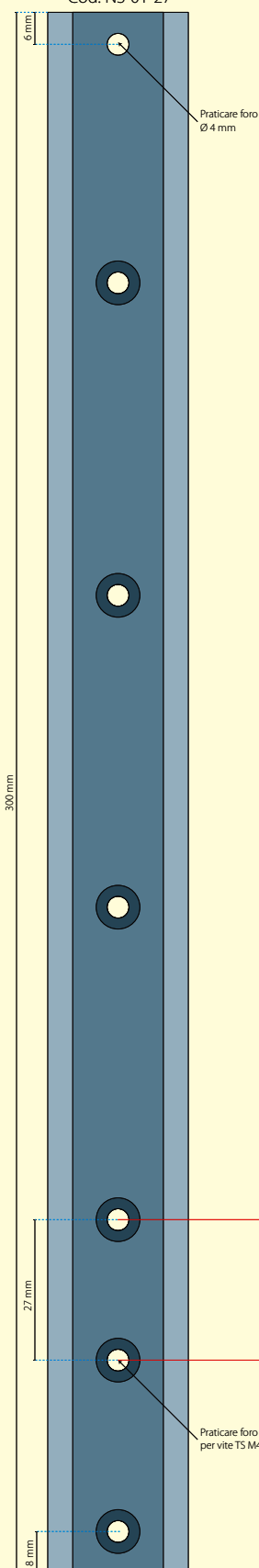
poter essere estruso attraverso l'ago della siringa (avente un diametro interno di soli 0,9 mm) deve essere sciolto; ciò si ottiene attivando opportunamente il riscaldatore in cui è inserita la siringa, ma senza superare la soglia dei  $33^{\circ}\text{C}$ , onde evitare di far perdere la "tempra" al cioccolato. La perdita di "tempra" del cioccolato impedirebbe di stampare oggetti che si sviluppano in altezza, quindi il cioccolato stesso non riuscirebbe più a solidificare a temperatura ambiente ( $24\div 27^{\circ}\text{C}$ ). Per questo motivo la scheda di controllo rileva costantemente la temperatura del riscaldatore mediante un termistore NTC da 100 kohm installato direttamente sul corpo del cilindro riscaldatore in alluminio e toglie alimentazione alla resistenza riscaldatrice quando si raggiungono i  $33\text{ }^{\circ}\text{C}$ .

A parte il mantenimento della temperatura al disotto di quella prestabilita, bisogna affrontare il problema del tempo di raffreddamento del cioccolato, che influisce sulla velocità di stampa: infatti deporre uno strato di cioccolato fuso e quindi caldo, su uno strato sottostante non ancora solidificato, può deformare l'oggetto in stampa, quindi l'applicazione di uno strato deve essere effet-

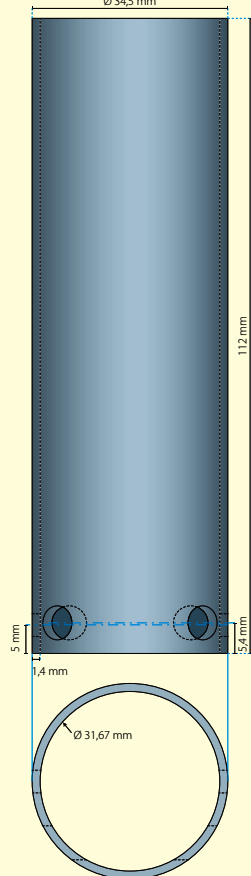




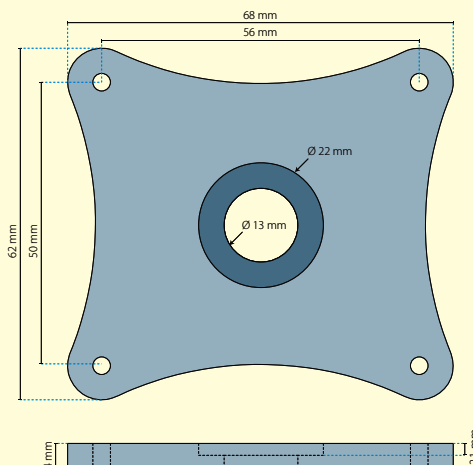
Guida IGUS  
Cod. NS-01-27



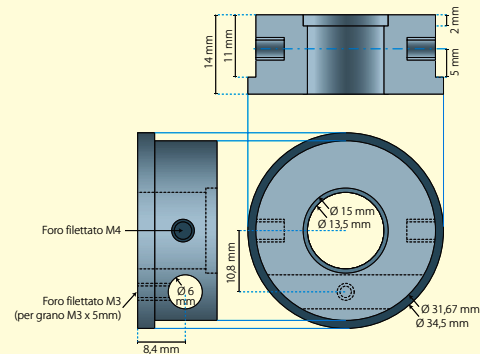
Cilindro  
Ø 34,5 mm



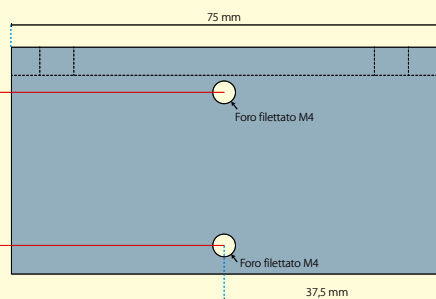
Piastra alluminio4



Bloccetto

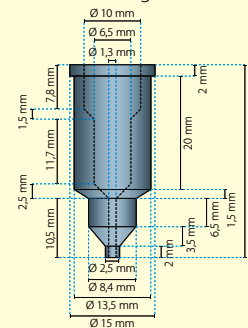


*Disegni meccanici quotati delle parti metalliche componenti il kit di modifica; il pattino e la guida sono prodotti IGUS che si acquistano già pronti (nella guida vanno realizzati dei fori aggiuntivi, come indicato qui a sinistra).*

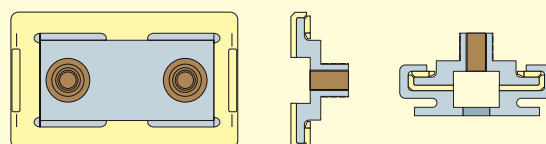


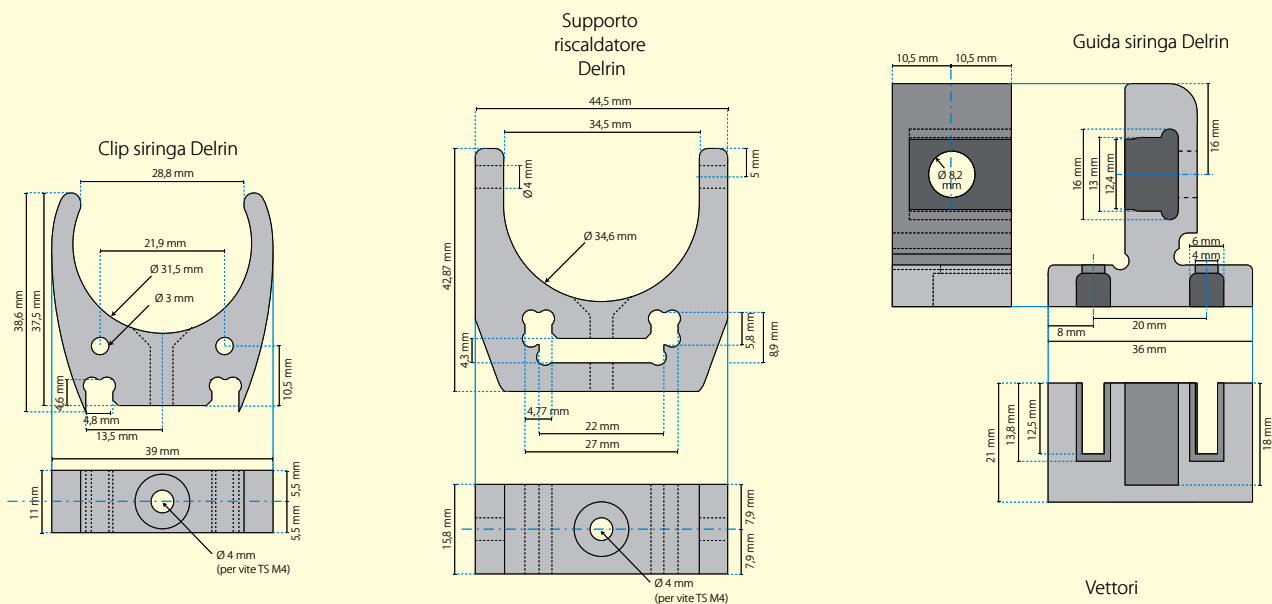
Staffa  
angolare

Camicia ago



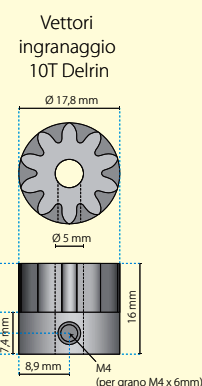
Pattino IGUS Cod. NW-02-27



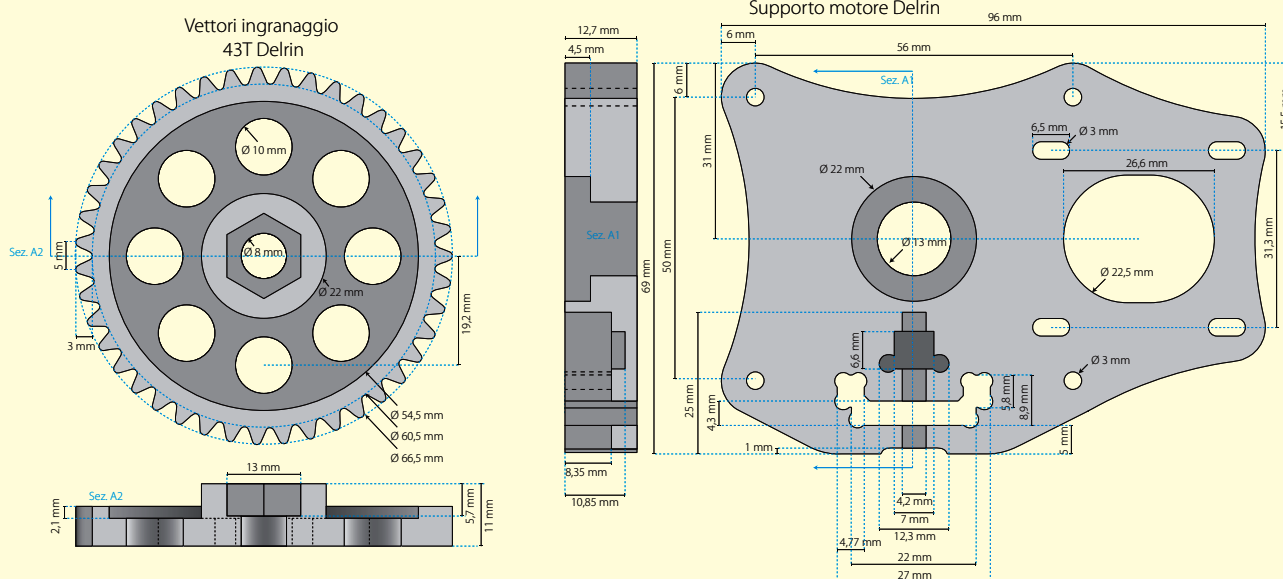


tuata quando quello sottostante è sufficientemente freddo da resistere sia al peso del cioccolato soprastante, sia al calore apportato dal cioccolato fuso deposto. Diciamo che lavorando con una quantità relativamente limitata come quella che può uscire da un ago con diametro interno di 0,9 mm (quale quello adottato nelle nostre prove) si può andare abbastanza veloce, ma ognuno dovrà, in base alla propria esperienza, scegliere la velocità di stampa appropriata, sfruttando adeguatamente le funzioni e le impostazioni del client di stampa

(Repetier-Host, Cura ecc.) come riterrà opportuno. Per raffreddare più in fretta il cioccolato (a tutto vantaggio della velocità di stampa) abbiamo tentato varie soluzioni, tutte provate in laboratorio: la prima è stata applicare al posto del piatto riscaldato una piastra in alluminio, di pari dimensioni, raffreddata mediante celle di Peltier, a loro volta ventilate sul lato caldo mediante dissipatori di calore. Questa soluzione si è dimostrata però adatta solo a realizzazioni poco spese, come le scritte in cioccolato, perché non



*Disegni meccanici quotati  
dei particolari in plastica  
necessari al kit di modifica.*



## Qualcosa sul cioccolato...

Il cioccolato è una sospensione di particelle solide finissime di zucchero e cacao disperse in una fase continua semisolida di grassi, meglio nota come "burro di cacao". Lo zucchero (il comune saccarosio) è una molecola che si scioglie bene in acqua, mentre non fa altrettanto nei grassi, quindi di suo non si scioglie nel burro di cacao. Per legarlo a quest'ultimo, bisogna ricoprirlo con molecole che lo aiutino a mescolarsi ai grassi; parliamo dei cosiddetti emulsionanti. E' questa la funzione della lecitina di soia che viene aggiunta al cioccolato: aiutare lo zucchero a veni-

re ricoperto e disperso nel burro di cacao.

Il cioccolato fondente, quello al latte e quello bianco contengono percentuali diverse di zucchero, cacao, burro di cacao ed eventualmente latte. Il cioccolato bianco in realtà non contiene cacao ma solamente burro di cacao, oltre a zucchero, aromi e latte. La massa di cacao contiene mediamente il 55% di burro di cacao. La percentuale totale di grassi presenti in una tavoletta di cioccolato dipende da quanto burro di cacao è presente. Solitamente, ma non sempre, più è amaro il cioccolato e più grasso

contiene rispetto allo zucchero. Ciò si comprende osservando le etichette delle tavolette di cioccolato: una al 70%, su 100 grammi di cioccolato contiene 30 g di zucchero, mentre una all'80% per lo stesso peso ha solo 20 g di zucchero. È quindi evidente che quest'ultima è più amara.

Le proprietà fisiche del cioccolato dipendono principalmente dalla composizione chimica dei suoi grassi: una miscela di trigliceridi composti quasi esclusivamente da acido palmitico (27%), acido stearico (34%), entrambi grassi saturi, e acido oleico (34%), monoinsatu-

consente di raffreddare gli strati alti. Inoltre l'insieme delle celle di Peltier e dei dissipatori pesa e costituisce una certa inerzia che ostacolerebbe in qualche modo il movimento del piatto, quando si stampa ad alta velocità. Abbiamo quindi ipotizzato il raffreddamento diretto del cioccolato deposto, inizialmente con un sistema della Vortex ([www.vortexitalia.com](http://www.vortexitalia.com)) a "tubi raffreddatori" basato su aria compressa, che consente la refrigerazione rapida localizzata, tuttavia richiede l'abbinamento a un compressore d'aria, che è ingombrante e rumoroso e che non rappresenta certo il migliore abbinamento a una macchina da cucina e da pasticceria. A questo punto abbiamo deciso

di orientarci nuovamente verso le celle di Peltier, per realizzare un sistema fisso e compatto di generazione del freddo per raffreddare il flusso d'aria diretto da una ventola direttamente contro la zona in cui viene deposto il cioccolato fuso; questa soluzione si è dimostrata la migliore, sebbene, malgrado le dimensioni ridotte della bocca che soffia l'aria fredda, in qualche modo il flusso investe l'ugello contenente l'ago, raffreddandolo e ostacolando il riscaldamento del cioccolato alla temperatura necessaria a fondere quest'ultimo. Per ovviare a questo inconveniente è stato necessario rivestire l'ugello con un isolante in grado di ripararlo dall'aria fredda. Descriveremo il sistema di raf-

freddamento, che è un optional da utilizzare solo quando si vuol stampare molto veloce, in un articolo futuro; ora spieghiamo come funziona il set di base per stampare col cioccolato.

### COME ESTRUDERE IL CIOCCOLATO

Il sistema che usiamo consiste nel riscaldare il corpo della siringa ed ancor più l'ago, mediante un riscaldatore elettrico a resistenza: l'ago è inserito in un cilindro (camicia) forato per farcelo passare e per ospitare anche l'innesco, capace di portare il tutto alla temperatura richiesta e mantenercelo.



Fig. 1



Fig. 2



Fig. 3



ro. Un trigliceride è costituito da una molecola di glicerina (pensate alla forma della lettera E) a cui sono legati tre acidi grassi ai tre segmenti della E. Nel cacao, il trigliceride POP (glicerina+acido Palmitico+acido Oleico+acido Palmítico) è presente per il 15%-20%. Il POS (Palmitico+Oleico+Stearico) al 35%-40%, mentre il SOS (Stearico+Oleico+Stearico) al 23%-25%. L'esatta composizione di questa miscela di grassi dipende dal luogo di origine del cacao e da come è stato coltivato: più il cacao cresce vicino all'equatore, più i suoi grassi fondono a temperature minori e quindi è più morbido.

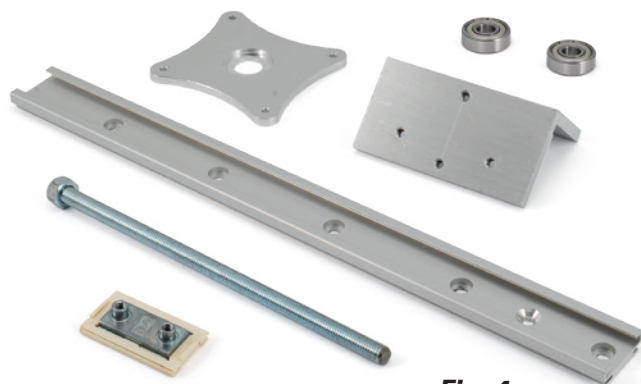
I grassi del cacao sono poco miscelabili con altri tipi di grasso, ad eccezione dei trigliceridi del latte, che invece si mescolano perfettamente con essi: non a caso esiste il cioccolato al latte. Chiudiamo questa disamina sul cioccolato parlando della tempra: vi è mai capitato di lasciare del cioccolato al caldo e di ritrovarlo tutto molliccio? Avrete notato che anche mettendolo in frigorifero, quando solidifica diventa opaco e pastoso e prendendolo in mano si ammolisce. Il motivo di questo comportamento risiede nel polimorfismo del burro di cacao contenuto nel cioccolato, cioè nella sua capacità di formare dei cristalli

di tipo diverso con proprietà diverse. Il burro di cacao può cristallizzare in sei strutture differenti, ma solo una, chiamata *forma V*, o fase  $\beta(V)$ , è quella che fornisce al cioccolato le proprietà che conosciamo. Per ripristinarla occorre fondere il cioccolato a circa 50 gradi e farlo raffreddare con una specifica variazione di temperatura nell'unità di tempo, finalizzata a consentire ai cristalli disomogenei di ricrescere secondo l'orientamento dell'unico cristallo stabile tra tutti quelli esistenti. L'operazione più casalinga consiste nel versare il cioccolato fuso su una lastra di marmo e mescolarlo con cura con una spatola (senza

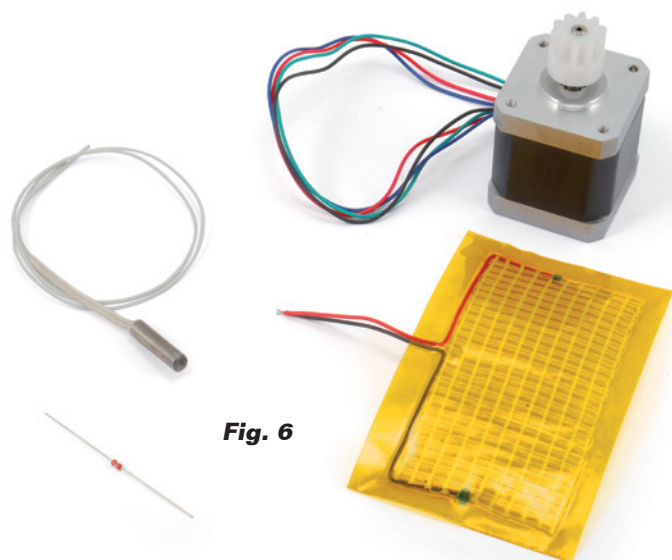
incorporare troppa aria) fino al raggiungimento di una temperatura di 27-28 gradi. Si rimette poi il cioccolato sul fuoco fino a quando la massa non abbia raggiunto una temperatura di 31 °C per il fondente e 29 °C per quello al latte e bianco (non sopra i 33 °C). In alternativa allo spatolamento su marmo, si può aggiungere cioccolato in pezzi a quello fuso (per abbassarne la temperatura) aggiungendo successivamente altro cioccolato fuso. I pasticciere esperti ricorrono anche alla "prova del foglio": il cioccolato è ben temperato se una goccia lasciata cadere su della carta da forno dà segni di indurimento entro 30÷60 secondi.

In questo modo il cioccolato che si trova nella testa (zona di uscita dell'iniettato) della siringa si fonde e può colare. Perché ciò avvenga, bisogna spingere lo stantuffo della siringa stessa e farlo con regolarità: qui entra in gioco l'azionamento dell'alimentatore, che consiste in una vite senza fine che spinge verso il basso lo stantuffo stesso. La cosa riesce meglio se tutto

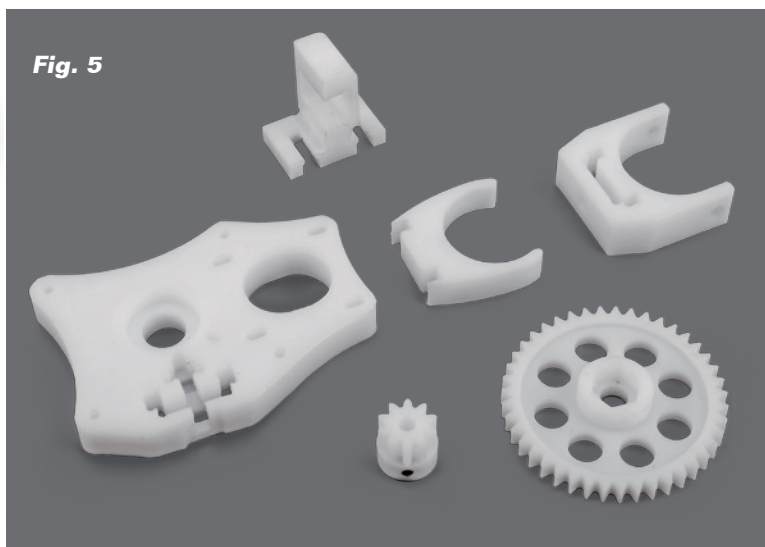
il cioccolato nella siringa è "ammollato", ecco perché l'intero corpo della siringa è introdotto in un cilindro in alluminio il cui duplice scopo è sostenere la siringa stessa e portarle il calore prodotto dal riscaldatore, il che è reso possibile dal fatto che il cilindro è fissato a quello che riscalda l'ago. Essendo l'alluminio un buon conduttore di calore,



**Fig. 4**  
Guida in alluminio, pattino, piastra, staffa angolare, cuscinetti, barra filettata.



**Fig. 6**



**Fig. 5**

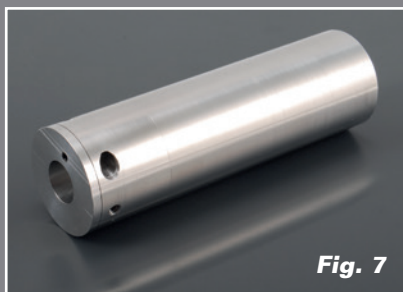


Fig. 7

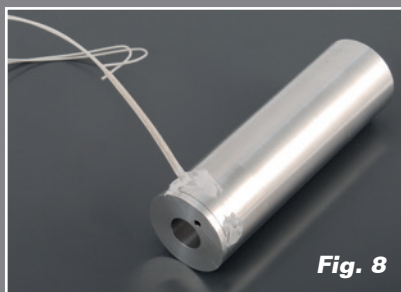


Fig. 8



Fig. 9

se è vero che trasmette assai bene il caldo alla parete esterna della siringa, è vero altrettanto che la sua parete esterna tende a disperdere e quindi a raffreddare; per questa ragione abbiamo pensato di avvolgere esternamente il cilindro con un riscaldatore in pellicola di Kapton.

La siringa può quindi essere caricata in due modi: colando all'interno della siringa cioccolato già fuso a bassa temperatura, quindi inserendola nell'apposito cilindro; inserendo nella siringa scaglie di cioccolato che poi fonderanno grazie all'azione del riscaldatore, una volta inserita la siringa nel cilindro.

#### MODIFICARE LA 3DRAG: LA NUOVA TESTA DI STAMPA

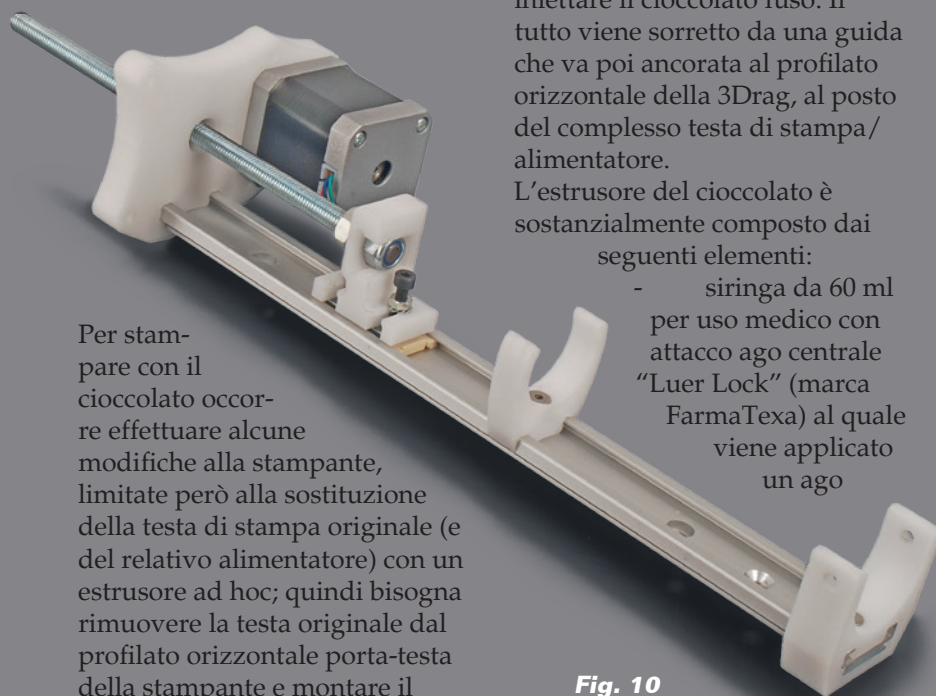


Fig. 10

Per stampare con il cioccolato occorre effettuare alcune modifiche alla stampante, limitate però alla sostituzione della testa di stampa originale (e del relativo alimentatore) con un estrusore ad hoc; quindi bisogna rimuovere la testa originale dal profilato orizzontale porta-testa della stampante e montare il

nuovo corpo, che consta di una siringa sostenuta da un apposito alloggiamento che ha il duplice compito di sostenerla e riscaldarla, oltre che da un meccanismo a vite senza fine che preme sul pistone della siringa per farle iniettare il cioccolato fuso. Il tutto viene sorretto da una guida che va poi ancorata al profilato orizzontale della 3Drag, al posto del complesso testa di stampa/alimentatore.

L'estrusore del cioccolato è sostanzialmente composto dai seguenti elementi:

- siringa da 60 ml per uso medico con attacco ago centrale "Luer Lock" (marca FarmaTexa) al quale viene applicato un ago

standard da 1,2 mm di diametro esterno (0,9 mm interni) opportunamente accorciato a 33,5 mm;

- corpo cilindrico in alluminio con diametro esterno di 35mm ed interno di 31,6 mm, opportunamente forato su cui viene avvolta una cartuccia riscaldante di tipo "flat" di SparkFun con tensione di alimentazione di 5 Vcc (Fig. 1);
- blocchetto cilindrico in alluminio (in cui viene alloggiata una cartuccia riscaldante cilindrica

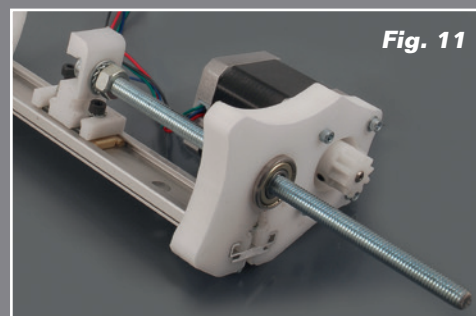


Fig. 11

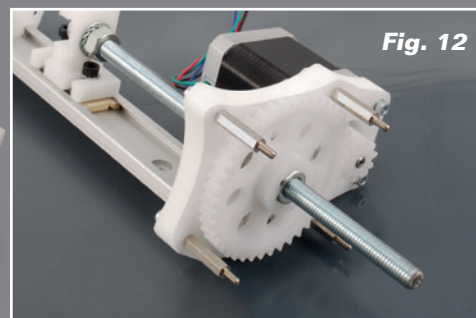
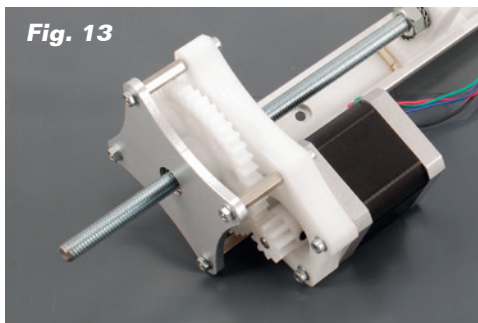


Fig. 12





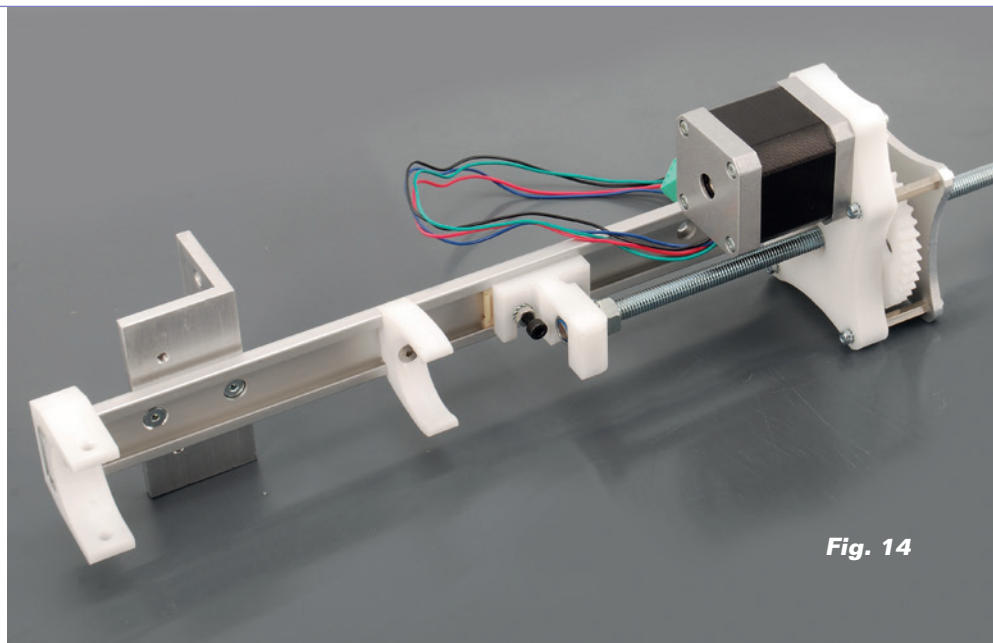
**Fig. 13**

15V/33W) opportunamente forato (Fig. 2);

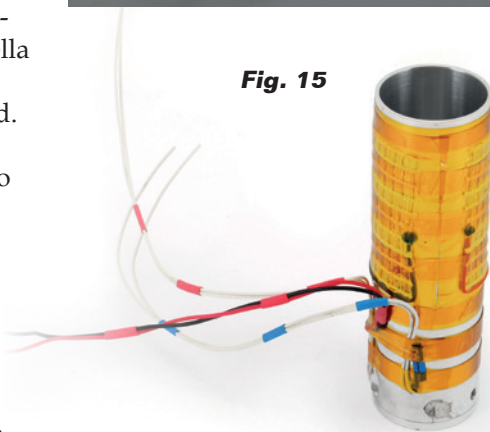
- camicia copri ago (ricavata da un tondino di alluminio) che permette di "tenere caldo" l'ago della siringa durante la stampa (Fig. 3);

Il sistema estrusore così composto si aggancia alla struttura della stampante e si completa con:

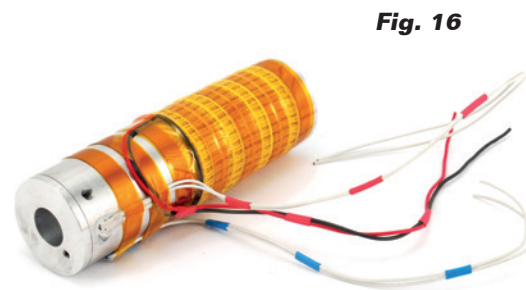
- guida in alluminio IGUS (cod. NS-01-27) da 27x300mm che rappresenta anche l'elemento portante di tutta l'unità di estrusione;
- pattino IGUS (serie 27 cod. NW-02-27) a cui viene ancorato il pistone della siringa mediante un'apposita staffa realizzata in Delrin;
- piastra in alluminio da 4 mm di spessore per il fissaggio della ruota dentata;
- staffa angolare in alluminio per fissaggio dell'estrusore al braccio della stampante 3Drag;
- barra filettata M8x190mm per azionamento pistone siringa;



**Fig. 14**



**Fig. 15**

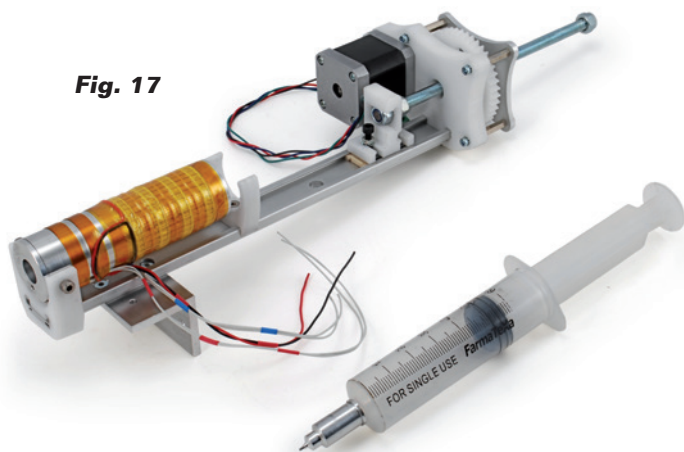


**Fig. 16**

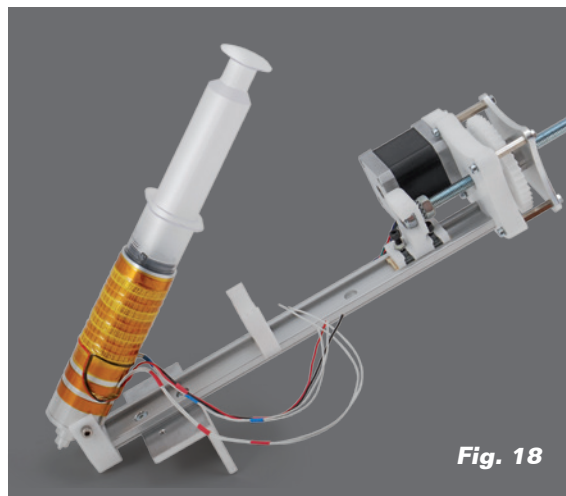
- 2 cuscinetti a sfere 8x22x7 mm con parapolvere;
- Supporti per il fissaggio del riscaldatore, della siringa e del motore, ruota dentata a 43 denti (Ø66,2mm) e pignone a

10 denti (Ø 17,6mm) realizzati in Delrin (Fig. 5).

L'alimentatore che spinge sul pistone della siringa è azionato mediante un motore passo-passo

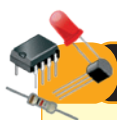


**Fig. 17**



**Fig. 18**





## per il MATERIALE

Il set di modifica della 3Drag per la stampa a cioccolato, completo di blocco estrusore e di una siringa, è disponibile (cod. 3DCHOCO) al prezzo di Euro 178,00. Il solo motore passo-passo Nema17 (cod. STEPMOT3DRAG) costa Euro 24,00; è comunque possibile recuperarlo dall'estrusore originale. La stampante 3Drag di serie (cod. 3Drag/K) costa Euro 580,00. Tutti i prezzi si intendono IVA compresa.

Il materiale va richiesto a:  
Futura Elettronica, Via Adige 11,  
21013 Gallarate (VA)  
Tel: 0331-799775 • Fax: 0331-792287  
<http://www.futurashop.it>

NEMA17 da 2,5A (potete recuperare questo motore dalla testa della 3Drag di serie); la cartuccia riscaldante da infilare nel blocchetto cilindrico in alluminio è da 15V/33W, mentre per rivestire e riscaldare il cilindro bisogna applicare la cartuccia riscaldante tipo "Flat" accoppiata ad un termistore NTC assiale da 100 kohm. Queste parti sono visibili nella Fig. 6.

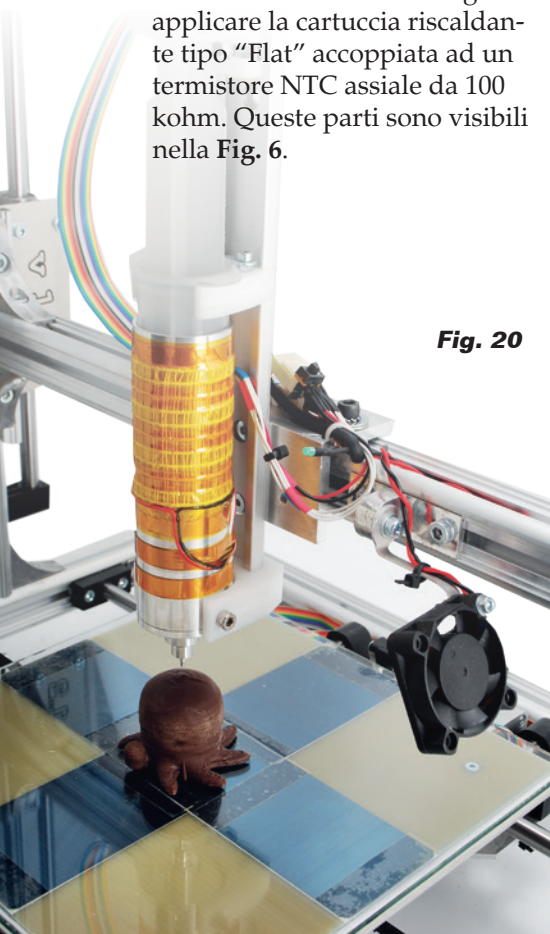


Fig. 20

## MONTAGGIO DEL CORPO ESTRUSORE

Una volta in possesso di tutti gli elementi necessari (il kit per la trasformazione può essere acquistato presso Futura Elettronica, [www.futurashop.it](http://www.futurashop.it)) si può procedere all'assemblaggio dei vari elementi dell'estrusore, procedendo in questo ordine:

1. inserire il blocchetto cilindrico nell'estremità forata del cilindro in alluminio (Fig. 7);
2. inserire la cartuccia riscaldante nel blocchetto dopo averla cosparsa con pasta termica, quindi bloccarla con un grano M3 (Fig. 8);
3. infilare il pattino nella guida in alluminio, quindi fissare a quest'ultima, mediante viti M4, il supporto per siringa, quello per il riscaldatore e per ultimo quello per il motore come visibile in Fig. 9;
4. inserire nelle relative sedi i due cuscinetti;
5. montare sul pattino mediante due viti M4 l'apposita staffa in Delrin, cui andrà fissata la barra filettata M8 mediante dado autobloccante da un lato e dado classico con rondella dentata dall'altro come visibile in Fig. 10;
6. inserire sulla barra filettata (lato supporto motore) una rondella 8x16 quindi la ruota dentata in Delrin ed il dado M8; infilare sul perno del motore il pignone in Delrin quindi fissare il motore stesso al relativo supporto dopo aver individuato il corretto accoppiamento tra le ruote dentate.
7. fissare il pignone col relativo grano (Fig. 11);
8. montare quindi sul supporto motore quattro distanziali esagonali da 20 mm (Fig. 12);
9. fissare ai distanziali la piastra in alluminio utilizzando quattro dadi (Fig. 13);

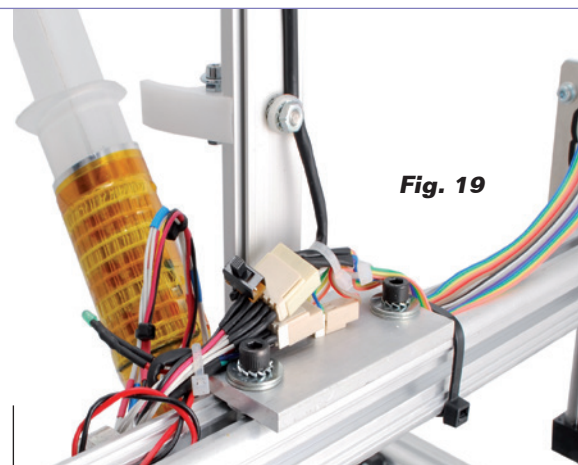


Fig. 19

10. fissare l'angolare in alluminio alla guida mediante due viti M4 a testa svasata (Fig. 14).
11. adesso dovete riprendere in mano il cilindro in alluminio porta-siringa e applicare il termistore NTC sulla sua superficie esterna, quindi fissarlo con del nastro in Kapton (Fig. 15);
12. avvolgere sul cilindro anche la cartuccia riscaldante tipo "flat" fissandola con nastro in Kapton (Fig. 16);
13. fissare il cilindro completo al relativo supporto mediante due viti M4 e poi infilare sull'ago della siringa l'apposita "camicia in alluminio" quindi inserire la siringa nel cilindro, fino ad ottenere la situazione visibile in Fig. 17; nel compiere questa operazione, accertatevi che la siringa sia libera di oscillare frontalmente per circa 45° (Fig. 18).

Avete dunque preparato il corpo che permette la stampa con il cioccolato; ora dovete montarlo sulla 3Drag. Allo scopo, dovette rimuovere dalla stampante l'estrusore per materiale plastico installato "di serie", quindi procedere così:

1. infilare due dadi quadri M5 nella cava superiore del braccio della stampante quindi fissare ad esso l'estrusore per cioccolato mediante due viti M5 (Fig. 19);

2.collegare in serie le due cartucce riscaldanti ed effettuare i collegamenti elettrici alla macchina della cartuccia riscaldante così ottenuta e dell'NTC dell'estrusore (Fig. 20).

Le modifiche meccaniche alla stampante terminano qui. Per poter utilizzare la 3Drag come stampante per il cioccolato è necessario effettuare anche l'aggiornamento firmware della scheda elettronica. Il firmware "Marlin versione V1 (LCD ON, Choco ON)" può essere scaricato direttamente dalla sezione Software (pagina web <http://3dprint.elettronica.in.it/2012/09/06/software/> in cui sono raccolte tutte le versioni disponibili) dove troverete anche la relativa Procedura di aggiornamento. Ovviamente il firmware della scheda può essere cambiato a proprio piacimento tutte le volte che lo si desidera, in funzione dell'estrusore montato sulla stampante, quindi se asportate la testa di stampa a cioccolato per rimettere quella di serie, potete ripristinare il firmware originale, perché la trasformazione è reversibile.

### MODIFICARE IL FIRMWARE

Spendiamo qualche parola per spiegare che cosa è stato cambiato nel firmware Marlin: per operare col cioccolato è necessario mantenere il riscaldatore a temperature molto più basse rispetto a quelle richieste nell'uso con i materiali plastici. A tal proposito ricordiamo che il firmware Marlin nasce per il controllo di stampanti 3D a fusione di filamento plastico e include tutta una serie di protezioni tra le quali quella che impedisce di estrudere il materiale al di sotto dei 170°C. Dovendo sciogliere il cioccolato ad una temperatura di

circa 33°C, dobbiamo modificare l'istruzione:

```
#define EXTRUDE_MINTEMP 170
```

trasformandola in:

```
#define EXTRUDE_MINTEMP 10.
```

Questa modifica sposta la soglia di protezione da 170°C a 10°C e consente di scendere a 33 °C.

Un'altra modifica firmware necessaria riguarda gli step/mm del motore dell'estrusore, imposta dalla differente conformazione meccanica di quest'ultimo. L'istruzione da ritoccare è:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT {64.25,64.25,2560,600};
```

ciò, allo scopo di permettere un'estrusione fluida del cioccolato. L'ultimo parametro (600) indica la velocità di rotazione del motore; per ottenere un'estrusione adeguata è necessario ridurre di 10 volte questo parametro, pertanto l'istruzione corretta diventerà:

```
#define DEFAULT_AXIS_STEPS_PER_UNIT {64.25,64.25,2560,60}
```

Un'ulteriore modifica riguarda il PID per il controllo della temperatura: per evitare che questa oscilli eccessivamente, modifichiamo il parametro P, ovvero l'istruzione:

```
#define DEFAULT_Kp 22.2
```

trasformandola in:

```
#define DEFAULT_Kp 32.2.
```

Questo valore è stato calcolato empiricamente e durante le stampe da noi effettuate ha dato risultati soddisfacenti.

### ACCORGIMENTI PER LA STAMPA

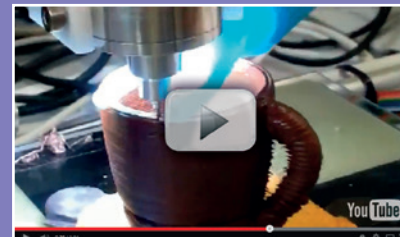
Come già accennato all'inizio di quest'articolo, per stampare un oggetto in cioccolato bisogna tener presente che i tempi di raffreddamento dei vari strati sono molto più lunghi rispetto quelli della stampa con le plastiche. Dall'esperienza pratica fatta in laboratorio con le nostre macchine modificate, consigliamo pertanto una velocità di stampa non supe-

riore ai 20 mm/s per consentire ai singoli strati di cioccolato di raffreddarsi quanto basta. Inoltre, nei parametri del software slicer bisogna impostare un diametro dell'ugello di 0,9 mm (è possibile anche impostare un 0,8 mm per "imbrogliare" lo slicer e quindi ottenere stampe più rifinite), un'altezza di layer di 0,7 mm e una temperatura dell'estrusore non superiore ai 35°C.

Queste impostazioni sono generiche: potreste avere bisogno di fare diverse prove per ottenere i risultati ottimali.

I parametri di configurazione dipendono anche dalla forma dell'oggetto, dal tipo e dalla qualità del cioccolato utilizzato e dalla temperatura ambiente in cui opera la stampante, pertanto trovare quelli giusti dipenderà dall'esperienza personale di ognuno.

**Guarda su YouTube  
come il cioccolato  
prende forma  
con la 3Drag!**



La 3Drag diventa pasticceria. Vuoi vedere di cosa è capace? Collegati alla pagina <http://youtu.be/SX-lsxacr1Y>.

Da questa pagina potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.



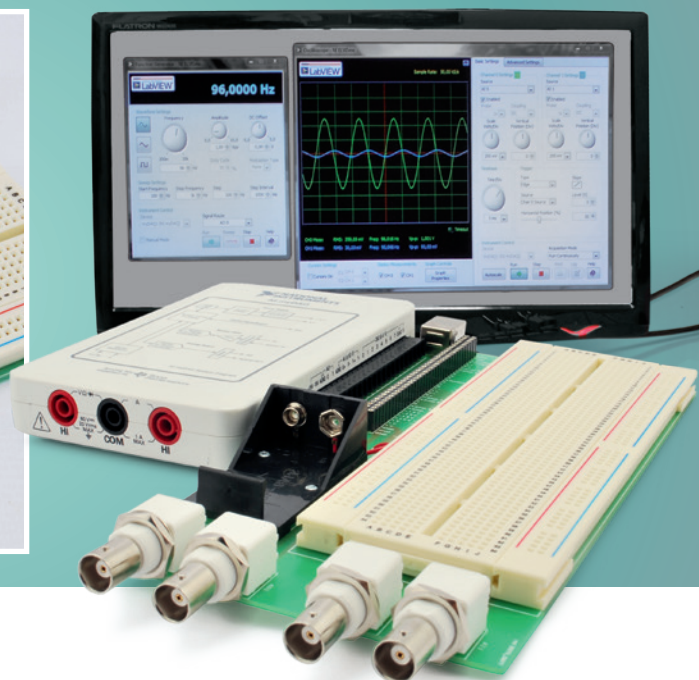
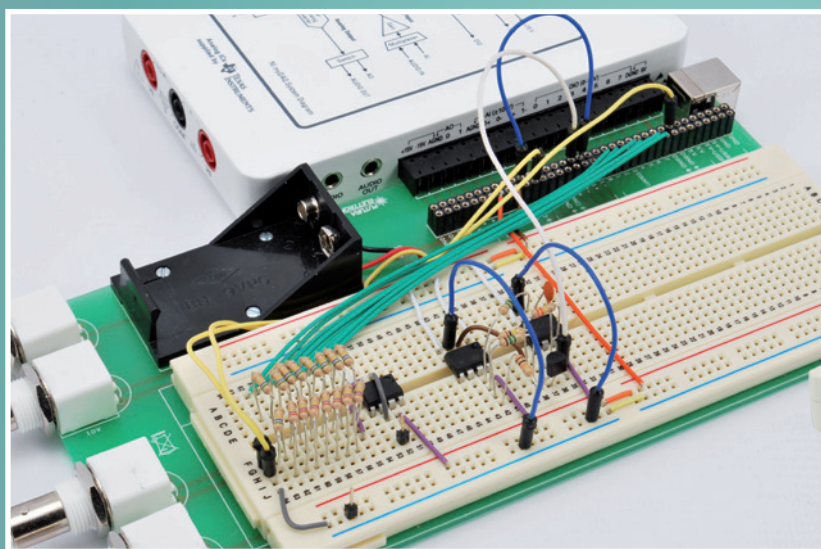




# myKIT

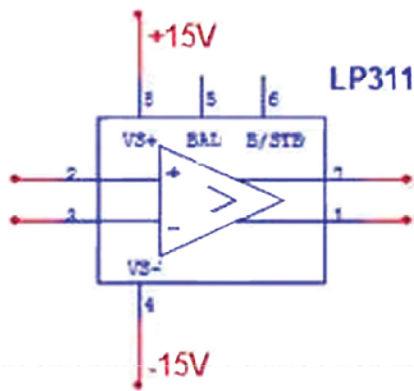
## GENERATORE DI ONDE TRIANGOLARI

di MARCO MAGAGNIN

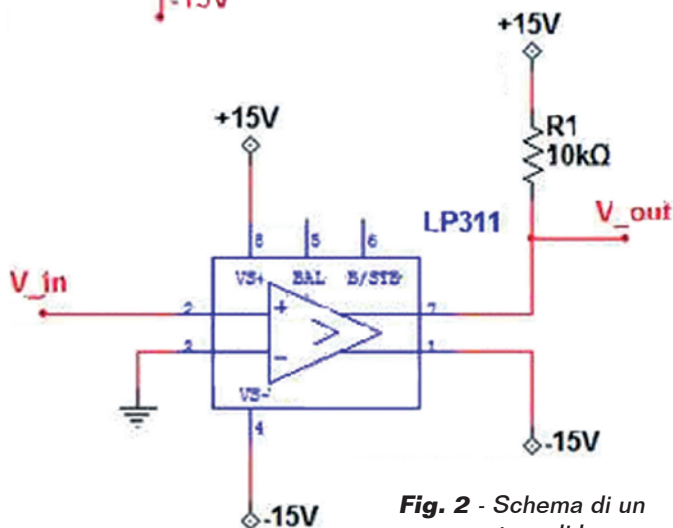


Il comparatore è un dispositivo che mette a confronto due tensioni e solitamente è un amplificatore operazionale (cioè un amplificatore con due ingressi, *invertente* e *non-invertente*, che amplifica la differenza tra le tensioni applicate ad essi) non retroazionato. Se la tensione presente all'ingresso non invertente è maggiore di quella applicata all'ingresso invertente, il terminale di uscita del comparatore assume un potenziale vicino a quello dell'alimentazione positiva, mentre se la tensione presente all'ingresso non-invertente è inferiore a quella applicata all'invertente, l'uscita si porta a livello basso. L'integrato LP311 è un comparatore a collettore aperto, cioè il suo capo di uscita è il collettore di un BJT; quando la tensione sull'ingresso non-invertente supera quella sull'invertente, l'uscita si comporta come un circuito aperto, pertanto per portarla a livello alto bisogna collegarla al positivo di

Realizziamo e studiamo le caratteristiche di un semplice generatore di onde triangolari utilizzando la funzione Scope di myDAQ per scoprire tutti i segreti di questo tipo di circuito.



**Fig. 1**  
Schema  
a blocchi  
dell'integrato  
LP311



**Fig. 2** - Schema di un  
comparatore di base.

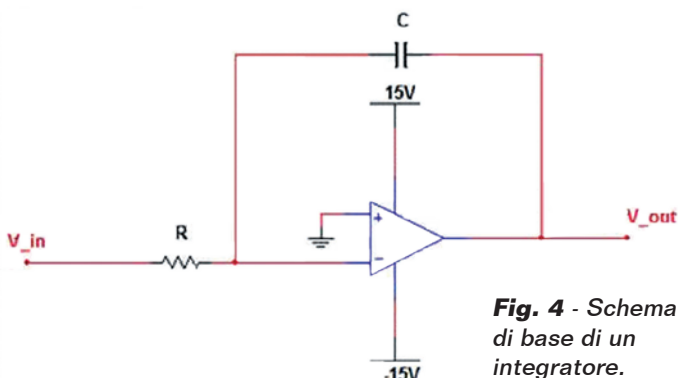
alimentazione con una resistenza (di pull-up). Quando la tensione sull'ingresso invertente supera quella sul non-invertente, l'uscita ha il potenziale  $-V_{CC}$ . L'uscita del comparatore in funzione delle tensioni di ingresso è visibile nella Fig. 3.

$$V_{OUT} = \begin{cases} +V_{CC} & \text{if } (V_{IN+} - V_{IN-}) > 0 \\ -V_{CC} & \text{if } (V_{IN+} - V_{IN-}) < 0 \end{cases}$$

**Fig. 3** - Funzione di trasferimento di un comparatore.

### CIRCUITO INTEGRATORE DI BASE

Un amplificatore operazionale può essere configurato



**Fig. 4** - Schema  
di base di un  
integratore.

come integratore semplicemente sostituendo la resistenza di retroazione con un condensatore. In questo modo si realizza un circuito che integra la tensione di ingresso.

Per sperimentare un circuito integratore si può realizzare lo schema di Fig. 4, utilizzando una resistenza da 1 Mohm ed un condensatore da 10 nF ceramico, riconoscibile dalla scritta 103 sul corpo. L'integrato è un TL081.

Idealmente, quando l'operazionale è configurato come amplificatore i suoi ingressi sono allo stesso potenziale; di conseguenza, se l'ingresso non-invertente è collegato a massa, anche l'ingresso invertente si trova al potenziale di massa. Pertanto, la corrente che scorre attraverso R è data dalla formula di Fig. 5:

$$i_R(t) = \frac{V_{IN}(t)}{R}$$

La corrente che attraversa il condensatore durante il transitorio di carica è data dalla formula di Fig. 6:

$$i_C(t) = -C \frac{dV_C(t)}{dt}$$

Nella formula di Fig. 6, la corrente è considerata negativa perché va dalla R all'uscita e  $V_C$  è considerata tra  $V_{OUT}$  e l'ingresso non-invertente dell'amplificatore operazionale. Semplificando, integrando e calcolando in funzione della tensione iniziale, si ottiene la formula di Fig. 7:

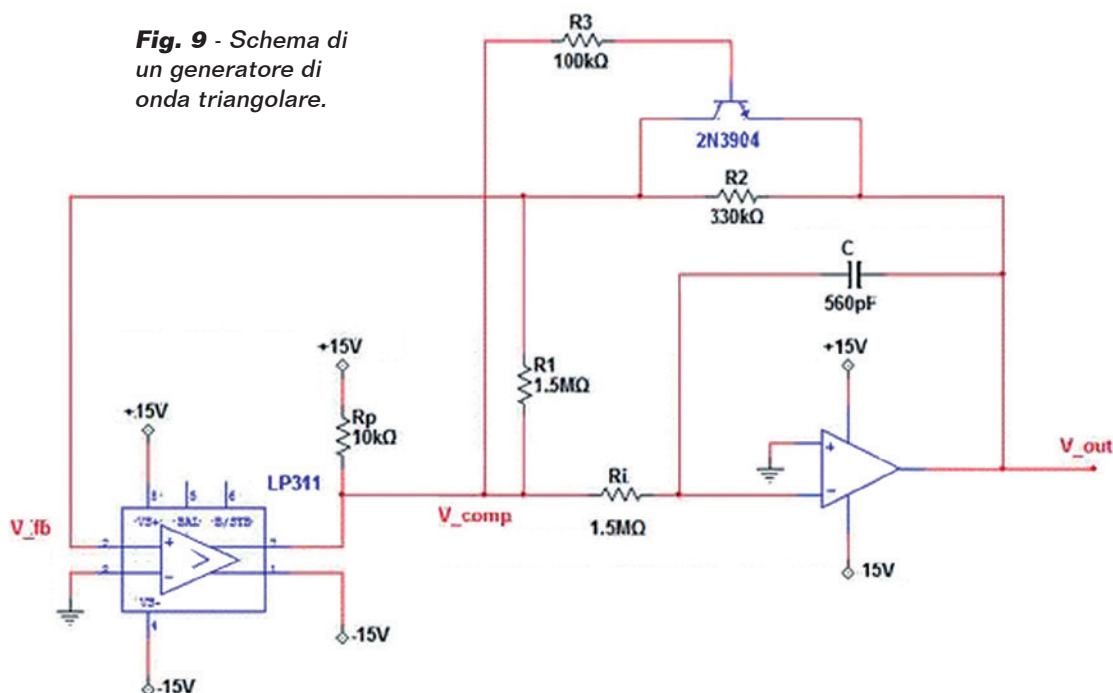
$$V_C = V_0 - \frac{1}{C} \int_{t_1}^{t_2} i_C(t) dt$$

Siccome l'operazionale ha resistenza d'ingresso infinita, la corrente negli ingressi è nulla, quindi tutta la corrente nella resistenza attraversa il condensatore e si può scrivere  $I_C = -I_R$ .

La tensione in uscita dall'operazionale è uguale a quella ai capi del condensatore. L'equazione di Fig. 8 mostra che il circuito integra la tensione di ingresso ad un tasso di  $1/RC$ . Se  $V_{IN}$  è una tensione continua, il potenziale dell'uscita varierà linearmente nel tempo con una pendenza pari a  $-V_{IN}/(RC)$ . Se invece in ingresso all'integratore è presente un segnale ad onda quadra, il livello di uscita cresce quando l'ingresso è a livello basso e decresce quando l'ingresso è a livello alto. Questo comportamento è alla base del funzionamento del generatore di onda triangolare (Fig. 8):

$$V_{OUT} = V_0 - \frac{1}{RC} \int_{t_1}^{t_2} V_{IN}(t) dt$$

**Fig. 9** - Schema di un generatore di onda triangolare.



### GENERATORE DI ONDA TRIANGOLARE

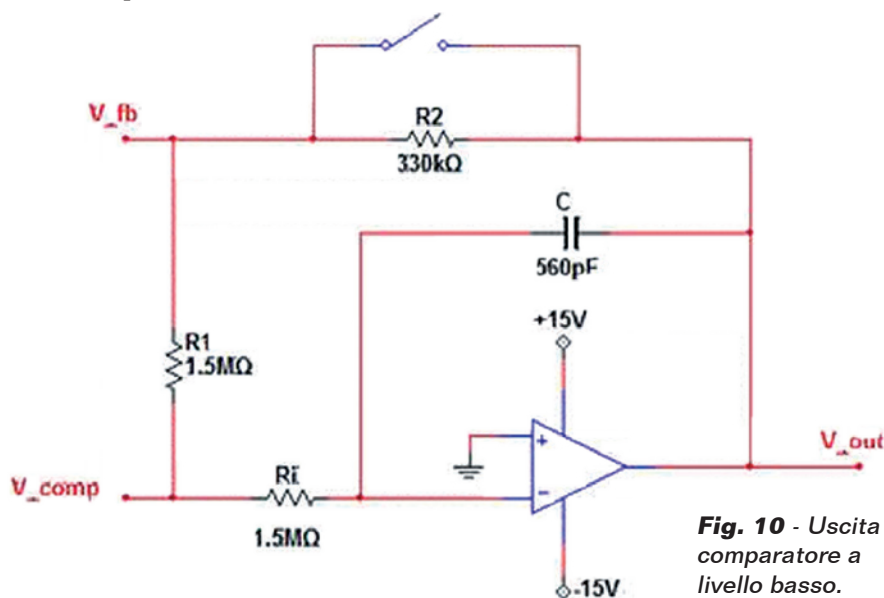
Si può realizzare un generatore di onda triangolare combinando un comparatore con un integratore. Siccome il comparatore presenta unicamente due livelli di uscita, può essere utilizzato come generatore di onda quadra. La rampa di salita dell'onda triangolare può essere utilizzata per innescare il comparatore al raggiungimento di un certo valore; a questo punto il segnale d'uscita dell'integratore comincia a scendere ed il comparatore verrà innescato di nuovo quando il segnale passerà dal livello di soglia. Per sperimentare un generatore di onda triangolare si può realizzare lo schema di Fig. 9.

L'operazionale è un TL081. Il condensatore da 560pF è di tipo ceramico, riconoscibile dalla sigla 561. Il terminale  $V_{OUT}$  deve essere collegato all'ingresso analogico di myDAQ per visualizzare il segnale con lo strumento Scope. Per comprendere il funzionamento del circuito bisogna analizzarne il comportamento in corrispondenza dei due livelli (alto e basso) assunti dall'uscita del comparatore.

Per semplificare, la Fig. 10 mostra il circuito equivalente alla sezione dell'integratore quando il transistor (usato come interruttore statico pilotato dal comparatore) è interdetto. Sempre per semplificare i calcoli, non si tiene in considerazione la tensione di saturazione dell'integrato LP311; si assume, inoltre, che

la  $V_{OUT}$  sia a 0 V. All'inizio si impone che l'uscita del comparatore commuti da livello alto a basso; essendo l'ingresso invertente dell'LP311 a massa, ciò significa che sia la tensione d'uscita dell'integratore, sia il potenziale sull'ingresso non-invertente dell'LP311 sono negativi.

A causa di ciò, all'ingresso dell'integratore giunge il potenziale dell'alimentazione negativa dell'LP311; finché dura questa condizione, la tensione all'uscita dell'operazionale cresce linearmente in funzione del tempo e  $V_{fb}$  diventa sempre meno negativa. Quando  $V_{fb}$  raggiunge zero volt, l'uscita del comparatore cambia di livello. La tensione di uscita corrispon-



**Fig. 10** - Uscita comparatore a livello basso.



dente ad una  $V_{fb}$  uguale a zero può essere calcolata con le formule di Fig. 11:

$$I_{R2} = \frac{V_{out} - V_{comp}}{R_1 + R_2}$$

$$V_{fb} = V_{out} - R_2 - I_{R2}$$

$$\text{Ponendo } V_{fb} = 0V$$

$$V_{out} = R_2 - I_{R2} = 330k\Omega \left( \frac{V_{out} - (-15V)}{1,5M\Omega + 330k\Omega} \right)$$

$$V_{out} = 0,1803(V_{out} + 15V)$$

$$V_{out} = \frac{0,1803 - 15V}{0,8197} = 3,3V$$

Parimenti, la tensione di uscita dall'integratore continuerà a salire fino a raggiungere il valore di 3,3 V, in corrispondenza dei quali l'uscita del comparatore commuterà dal livello basso a quello alto.

Ora cerchiamo il punto di innesco inferiore. Lo schema ridisegnato e semplificato per ottenere un livello alto in uscita dal comparatore è mostrato in Fig. 12.

Si può notare che il BJT del circuito originale viene considerato in saturazione e di conseguenza la R2 cortocircuitata. In questa configurazione diventa facile determinare il punto di commutazione inferiore in quanto la  $V_{fb}$  è uguale a  $V_{OUT}$ .

Di conseguenza, il punto di commutazione inferiore corrisponde ad una tensione di 0 V. Tutto questo porta a concludere che l'intero processo produce in uscita

un'onda triangolare che inizia a 0 V, sale fino al valore di 3,3 V, per poi ridiscendere a 0 V. Questo ciclo si ripete all'infinito.

Detto ciò, si possono calcolare i tempi di salita e di discesa delle rampe dell'onda triangolare. Come visto in precedenza, la tensione in uscita dal comparatore può essere calcolata con la formula di Fig. 13:

$$V_{OUT} = V_0 - \frac{1}{RC} \int_{t_1}^{t_2} V_{IN}(t) dt$$

Inizialmente si può adattare la formula per calcolare la durata del fronte di salita (Fig. 14):

$$V_{OUT} = - \frac{1}{(1,5M\Omega)(560pF)} \int_0^{t_r} (-15V) dt$$

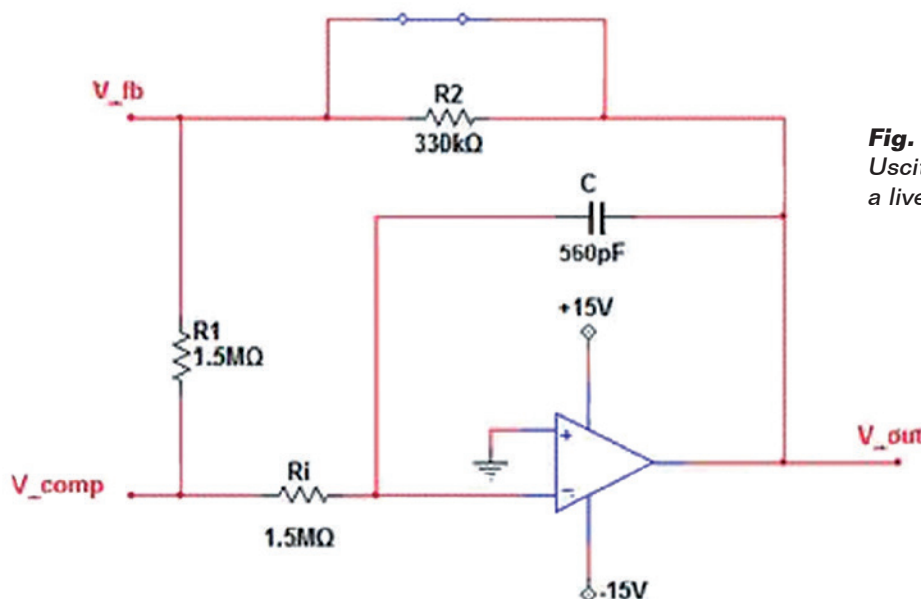
$$V_{OUT} = \frac{15}{(1,5M\Omega)(560pF)} (t_r - 0)$$

$$t_r = V_{OUT} \left( \frac{(1,5M\Omega)(560pF)}{15} \right)$$

Siccome il comparatore è stato calcolato per commutare quando la tensione  $V_{OUT}$  raggiunge il valore di 3,3 V, inserendo questo valore nella formula di Fig. 14 si ottiene il risultato di Fig. 15:

$$t_r = (3,3V) \left( \frac{(1,5M\Omega)(560pF)}{15} \right) = 184,8\mu s$$

In modo simile, si può svolgere la formula di Fig. 13 introducendo i valori tipici di quando l'uscita del com-



**Fig. 12**  
Uscita comparatore  
a livello alto.

paratore è a livello alto; la rampa di tensione all'uscita dell'integratore è discendente e la tensione  $V_{OUT}$  è uguale a 3,3 V (**Fig. 16**):

$$V_{OUT} = (3,3V) - \frac{1}{(1,5M\Omega)(560pF)} \int_0^{t_f} (15V) dt$$

$$V_{OUT} = (3,3V) - \frac{15}{(1,5M\Omega)(560pF)} (t_f - 0)$$

$$t_f = (3,3V - V_{out}) \frac{(1,5M\Omega)(560pF)}{15}$$

Nel caso di fronte di uscita discendente, il comparatore commuta il livello della propria uscita in corrispondenza di una tensione  $V_{OUT}$  uguale a 0, portando alla formula di **Fig. 17**:

$$t_f = (3,3V) \frac{(1,5M\Omega)(560pF)}{15} = 184,8\mu s$$

A questo punto si può notare che il tempo  $t_r$  è uguale al tempo  $t_f$ : ciò dice che le forme d'onda generate dal circuito sono simmetriche, ossia che le semionde negative eguagliano quelle positive. La frequenza delle forme d'onda si può calcolare con la formula di **Fig. 18**:

$$f_{\Delta} = \frac{1}{t_r + t_f} = 2,7kHz$$

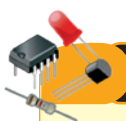
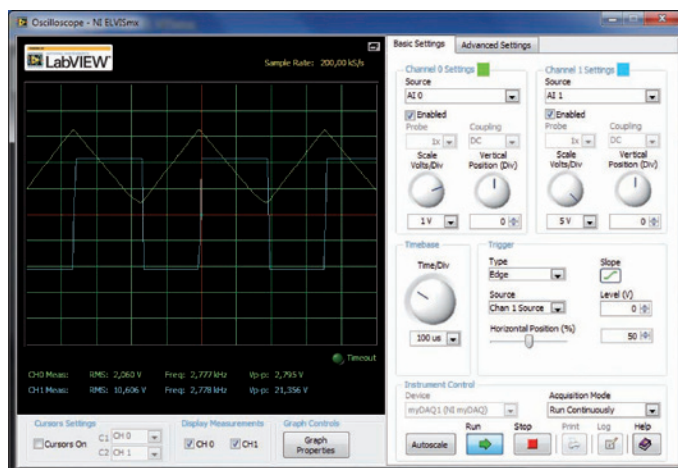
La schermata di **Fig. 19** mostra la forma d'onda in uscita dal generatore: si può notare che essa non è perfettamente simmetrica e che la frequenza è leggermente diversa da quella calcolata. Questo fenomeno è dovuto, come specificato all'inizio della descrizione

del processo di calcolo, alle semplificazioni introdotte nei calcoli e al fatto che nella spiegazione si è considerato il comportamento dell'amplificatore operazionale ideale, non di quello reale.

Per dare una visione più immediata del comportamento dell'intero circuito, nella **Fig. 20** è mostrata la forma d'onda triangolare in uscita dal generatore con l'onda quadra in uscita dal comparatore.

Anche in questo caso la frequenza e le tensioni sono leggermente differenti rispetto a quelle calcolate, a causa degli effetti del carico presente all'uscita del comparatore.

**Fig. 20**



## per il MATERIALE

Questo progetto fa parte del set di esperimenti didattici realizzato da IRS-Elettronica per la board NI myDAQ con il supporto del software LabVIEW. Il set (cod. SETmyKIT, Euro 232,00) comprende tutti i componenti per realizzare i 22 esperimenti, i VI relativi ed una protoboard.

È possibile richiedere protoboard aggiuntive (cod. PROTOBOARDmyDAQ) al prezzo di 52,00 Euro.

Tutti i prezzi sono comprensivi di IVA e non comprendono la board NI myDAQ né il software LabVIEW.

Futura Elettronica distribuisce questi prodotti presso gli utenti privati mentre la distribuzione presso gli Istituti Scolastici è a cura di IRS srl (Via Vigonovese 81 - 35127 PADOVA - Tel: +39 049 8705156 - info@irsweb.it).

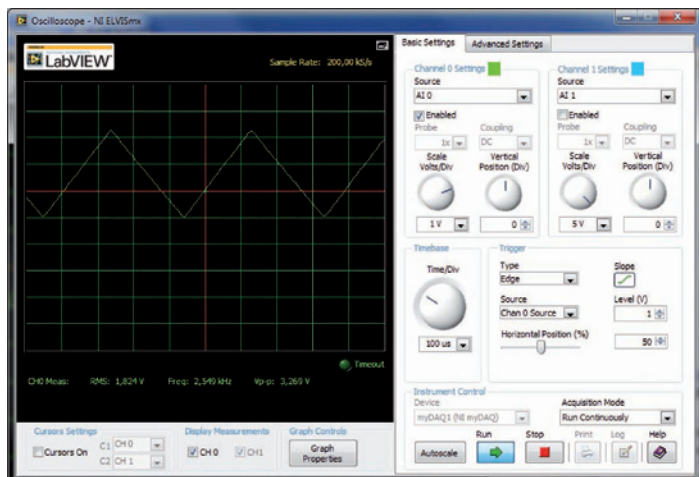
Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)

Tel: 0331-799775 • Fax: 0331-792287

www.futurashop.it

**Fig. 19**









Products

Solutions

Content

Resources

Partners

Contact Us

Search Dyn

SIGN IN

EN

Register/transfer a domain:

Enter domain to search

.com

CHECK AVAILABILITY

## Devices



## REMOTE ACCESS

*DynDNS Remote Access*

Access your computer, DVR, or camera system remotely with a memorable hostname. [LEARN MORE](#)

Features

## Personal



## BASIC WEBSITES

*Standard DNS*

Complete DNS hosting for a single domain in one easy-to-use interface. [LEARN MORE](#)

Features

## Business



## GROWING

*Managed DNS Express*

For companies and startups that need maximum reliability, speed, and scalability. [LEARN MORE](#)

Features

## ENTERPRISE

*Managed DNS*

For the biggest and best brands who can't afford a minute of downtime. [LEARN MORE](#)

Advanced Features

Connettiamoci a un DVR e a una telecamera IP affacciati sul web tramite una connessione a IP dinamico, mediante il servizio dynDNS.

# DYNDNS: LOCALIZZA L'IP DINAMICO

**L**a diffusione dei sistemi di videosorveglianza cui abbiamo assistito negli ultimi anni, e più recentemente dell'Internet of Things, hanno reso la connessione da remoto un'esigenza generalizzata. Fortunatamente, funzionalità che solo pochi anni fa erano riservate alle grandi realtà, oggi sono alla portata di tutti, dato che oggi giorno l'implementazione dell'accesso web ai sistemi, richiede un investimento

ridotto ed è abbastanza semplice. Poter vedere da remoto cosa accade nella propria abitazione e visionare le registrazioni della casa di villeggiatura mentre si è in città, ad esempio, è qualcosa che oggi è davvero alla portata di tutti. In questo articolo spiegheremo come farlo

utilizzando due dei dispositivi dedicati a tali funzioni e in che modo farli interagire con il servizio di DNS Dinamico (dynDNS, per semplicità).

di GIANLUCA STELLA

Cominceremo con un po' di teoria sulle reti e i servizi di dynDNS, passando poi al primo esempio pratico che prevede la configurazione del DVR FR517

(acquistabile su *www.futurashop.it*) con quattro ingressi analogici, registrazione su HDD e un web server integrato. Vedremo poi un secondo esempio pratico utilizzando la telecamera PTZ wireless CP526W.

Negli esempi pratici illustreremo i cinque passi necessari alla configurazione, che sono:

- 1) registrazione di un account su dynDNS.it;
- 2) creazione di un host web dinamico;
- 3) configurazione delle impostazioni di rete del dispositivo;
- 4) configurazione del servizio dynDNS.it;
- 5) configurazione del port forward sul router.

Infine vedremo poche semplici regole per diagnosticare e

correggere eventuali problemi di connessione da remoto.

### IP PUBBLICI E PRIVATI

Gli indirizzi IP servono per identificare un nodo (computer, server, PLC, DVR, ecc) connesso a una rete informatica; Internet, la rete delle reti, utilizza gli indirizzi IP per indirizzare, appunto, i pacchetti di dati scambiati dai dispositivi di tutto il mondo.

Gli indirizzi IP pubblici sono quelli indirizzi utilizzati nella rete pubblica (Internet), mentre gli IP privati sono utilizzati all'interno delle case e delle aziende per le reti locali (LAN). Per scoprire il vostro indirizzo IP privato, controllate le impostazioni di rete del PC con cui siete connessi alla rete: sarà del tipo 192.168.X.Y, oppure del tipo 10.X.Y.Z o del

tipo 172.X.Y.Z (per i più pignoli, nell'ultimo esempio X è compreso tra 16 e 21).

Per scoprire invece l'indirizzo IP pubblico del vostro modem o router (i dispositivi che vi permettono di connettervi a Internet), collegatevi a <http://checkip.dynDNS.it>.

Il router vi permette di navigare in Internet con il vostro PC (che ha un indirizzo IP privato sulla rete locale) attraverso l'indirizzo IP pubblico assegnatogli dal provider di connessione. Sulla vostra rete locale ci possono essere anche altri dispositivi connessi con indirizzo IP privato e tutti potranno navigare, sempre utilizzando lo stesso IP pubblico. Questo processo, in termini tecnici, si chiama Network Address Translation (NAT). Avrete sicuramente sentito parlare o letto sui forum degli IP "nattati"; ecco, ciò significa che la vostra rete, in cui ogni elemento ha un proprio indirizzo IP, si affaccia ad Internet tramite un unico indirizzo IP pubblico.

Non essendoci più indirizzi IP pubblici disponibili, molti provider utilizzano la stessa tecnica del vostro router domestico anche sulle connessioni internet, soprattutto quelle che avvengono in "mobile" tramite chiavetta: vi assegnano un indirizzo IP privato e un loro router utilizza un solo IP pubblico per tutti i router domestici connessi. Quando il vostro provider vi assegna un indirizzo IP privato "nattato" il vostro router e la vostra rete domestica non saranno accessibili dall'esterno; è quindi fondamentale verificare con il proprio provider di connettività la disponibilità dell'indirizzo IP pubblico.

Negli esempi pratici che seguiranno, analizziamo l'ipotesi di una connessione ADSL con IP pubblico dinamico, che rappre-

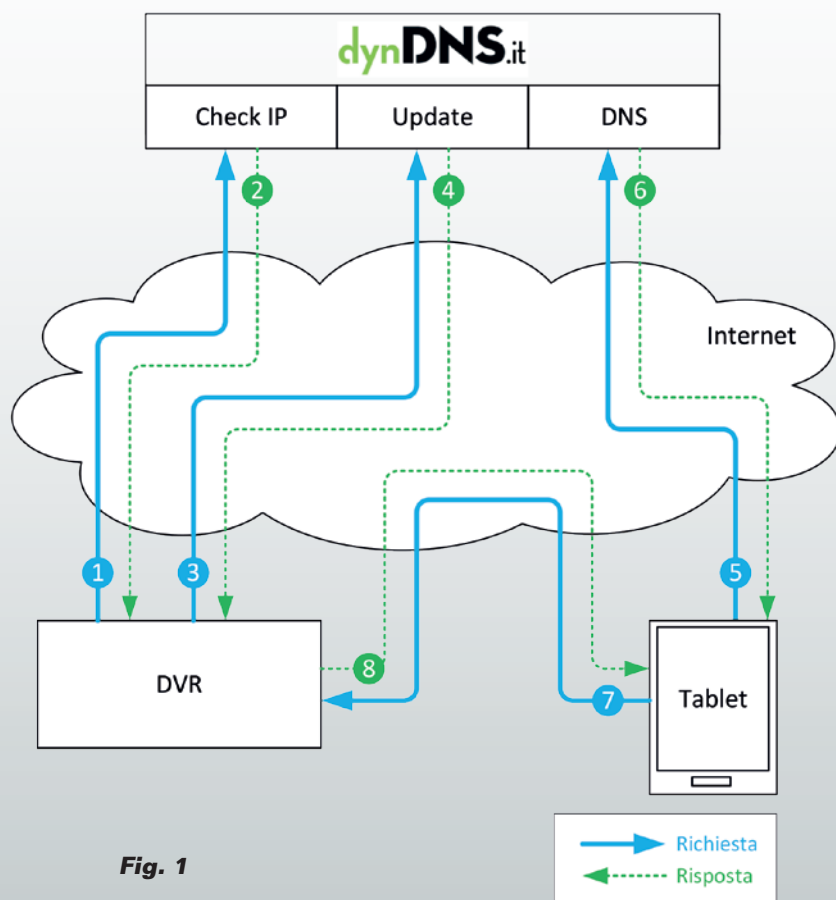
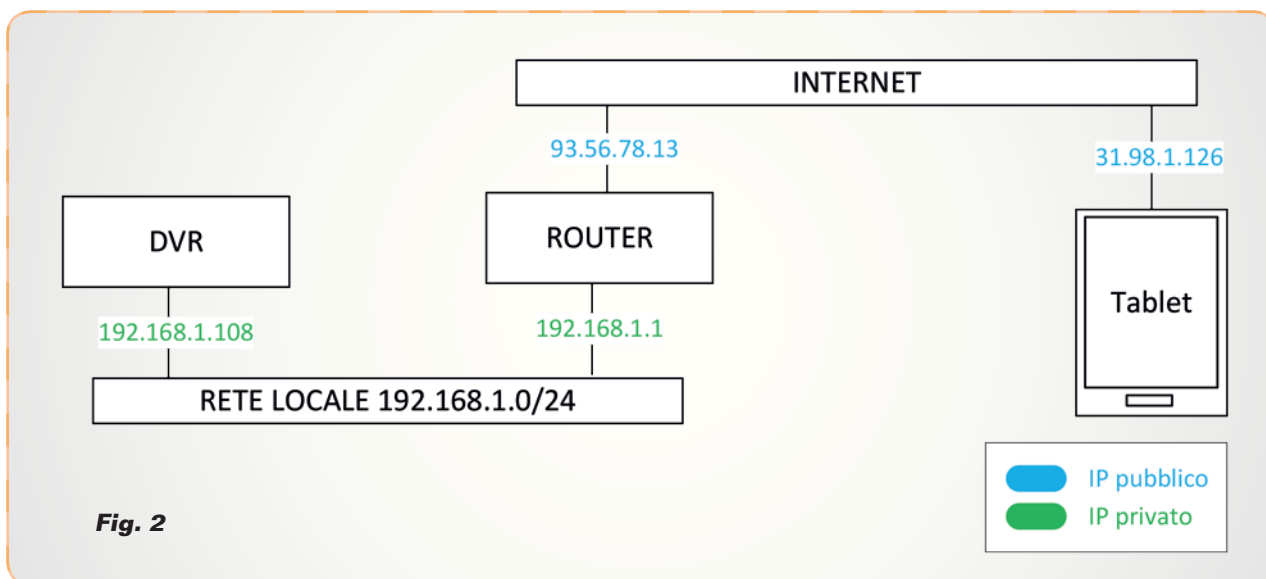


Fig. 1



senta il tipico contratto ADSL ad uso privato; in questo tipo di connessione, l'indirizzo è definito "dinamico" perché non è sempre uguale: tutte le volte che il vostro router effettuerà la connessione gli verrà assegnato un nuovo IP, diverso dal precedente.

Alcuni di voi si staranno chiedendo: "se il mio indirizzo IP cambia tutte le volte, come faccio a connettermi alla mia rete di casa? Come posso sapere quando l'IP è cambiato?"

La risposta la vedremo tra poco.

### LA SOLUZIONE PER GLI IP DINAMICI: IL DYNDNS

In Internet viene usualmente impiegato il Domain Name Service (DNS), che è un servizio di rete creato per semplificare l'accesso ai siti mediante l'attribuzione di nomi facilmente ricordabili; grazie ad esso, quando nella barra degli indirizzi del browser web inserite l'indirizzo espresso come nome, il DNS identifica l'indirizzo IP corrispondente e vi collega al sito da esso identificato. Per esempio, se digitate *http://www.elettronica.in.it*, il DNS comunica al vostro computer (client web) a quale indirizzo IP corrisponde il nome a dominio *www.elettronica.in.it*. Nel browser web

potete comunque scrivere direttamente l'indirizzo IP: in questo caso il DNS non interviene. Se l'indirizzo IP è dinamico, il DNS da usare deve essere anch'esso dinamico: quando l'indirizzo IP cambia, anche il nome a dominio deve essere associato al nuovo indirizzo IP. Questo sistema si chiama, appunto, DNS dinamico o dynDNS.

Il DNS dinamico è un servizio che registra l'attuale IP di un sito o di un utente Internet e lo associa al nome con cui viene ricercato nel browser. Negli esempi pratici che vedremo in seguito utilizzeremo dynDNS.it (*http://dynDNS.it*), un DNS dinamico italiano che permette una registrazione gratuita e che è compatibile con tutti i dispositivi che supportano *dyndns.org*. La base di un DNS dinamico è l'*host*, un indirizzo internet che potrete scegliere a vostro piacere per accedere al vostro dispositivo di videosorveglianza. Negli esempi pratici che vedremo tra poco utilizzeremo *dvr0001.homepc.it* e *camera0001.controlliamo.com*.

### Schema di funzionamento

Il funzionamento di un sistema dynDNS è rappresentato in Fig. 1, in cui sono illustrati un DVR

connesso tramite un router ADSL ad internet, il sistema *dynDNS.it* e un utente remoto connesso ad Internet tramite il proprio tablet con connessione 3G.

Tutto comincia dal DVR, che invia una richiesta per verificare l'indirizzo IP pubblico del router (1); il servizio *checkip* di *dynDNS.it* risponde indicando l'IP attuale (2). Il DVR, se l'IP ricevuto è diverso da quello dell'*host*, invia una richiesta di *update* a *dynDNS.it* (3) che risponde indicando il risultato dell'operazione (4). Ad esempio, se tutto è andato a buon fine risponderà *good*.

Quando l'utente remoto digita il nome dell'*host* sul proprio tablet, come abbiamo già spiegato poco fa, il tablet contatta il DNS per sapere l'indirizzo IP a cui collegarsi (5); DynDNS fornisce in risposta l'indirizzo IP pubblico (6) che il DVR ha inviato precedentemente con la richiesta di *update* (3). A questo punto il tablet può collegarsi all'indirizzo IP pubblico cui il DVR è connesso (7), che risponderà fornendo l'accesso alla propria interfaccia web (8). I più attenti tra voi avranno notato che non è spiegato come la richiesta (7) possa raggiungere il DVR; ebbene, è tutto merito del router e del Port Forward.



**Fig. 3**

### Schema della rete

Nei paragrafi precedenti abbiamo illustrato i flussi delle informazioni e spiegato cosa avviene quando da remoto si accede al proprio dispositivo di videosorveglianza. Utilizzeremo ora lo schema in Fig. 2 per capire a livello di rete come sia possibile raggiungere il DVR, collegato in rete locale, dalla rete Internet. Il DVR e il router sono connessi tra di loro tramite la rete locale (Local Area Network, LAN) con indirizzamento 192.168.1.0/24; abbiamo già visto come questa classe di indirizzi sia definita "privata".

Il router, oltre essere collegato in LAN, è collegato anche ad Internet tramite il proprio modem ADSL interno; la rete Internet utilizza indirizzamenti pubblici, e il vostro provider di connessione ADSL assegna al router l'indirizzo pubblico 93.56.78.13. Per consentire ai dispositivi connessi in LAN di navigare in Internet, il vostro router mette in atto il NAT, che abbiamo già illustrato in precedenza.

Ma come può un PC o un tablet, anch'esso connesso ad Internet all'esterno della rete locale, dialogare con il DVR connesso in LAN?

Ebbene, tutti i router in commercio supportano una funzionalità che viene chiamata Port Forward (letteralmente, inoltramento della porta); attraverso il Port Forward diremo al nostro router: "tutte le richieste che arrivano da Internet sulla porta 80 (la porta predefinita del web) devi mandarle al DVR che si trova in rete locale all'indirizzo 198.168.1.108, sempre sulla porta 80".

In linea generale, quindi, per configurare il Port Forward occorre inserire una regola basata su:

- 1) porta pubblica: la porta sull'indirizzo internet pubblico che resterà in ascolto;
- 2) indirizzo privato: l'indirizzo IP del vostro dispositivo di videosorveglianza a cui inoltrare le richieste;
- 3) porta privata: la porta sulla quale verranno inoltrate le richieste;
- 4) protocollo: si può scegliere tra TCP, UDP o entrambi, in questo articolo useremo TCP.

Passiamo ora alla pratica, configurando il DVR FR517 e la IP camera CP526W.

### PRIMO ESEMPIO PRATICO: DVR FR517

Il DVR FR517 è un dispositivo con quattro ingressi analogici per collegare altrettante telecamere, che permette la registrazione video su HDD, il collegamento di un monitor esterno, l'accesso dalla rete locale e da remoto grazie alla interfaccia web. In pochi semplici passi vedremo come creare un host dynDNS.it, come configurare il DVR e il Port Forward sul vostro router.

### Creazione di un host DynDNS

L'elemento base di ogni dynDNS è l'host, cioè il nome che daremo al nostro dispositivo e che verrà aggiornato con l'IP pubblico

del router. In questo modo, per connettervi al vostro impianto di videosorveglianza vi basterà ricordare il nome che avete scelto: il sistema dynDNS lo tradurrà per voi nell'indirizzo IP pubblico che il vostro router ha in quel momento.

La prima cosa da fare è collegarsi al sito <http://dynDNS.it> e registrarsi cliccando sul link *Registrati visibile* in alto a destra (Fig. 3); tra le soluzioni proposte (DynDNS\2 - DynDNS - solutions), scegliete il pacchetto FREE e compilate i campi necessari alla registrazione (DynDNS\3 - dynDNS - registrazione). Al termine della compilazione riceverete nella vostra casella di posta elettronica un messaggio di conferma contenente la vostra password per l'accesso. Utilizzando il link contenuto nel messaggio, inserite il vostro nome utente e la vostra password e fate clic su *Collegati*.

Il vostro account dynDNS.it è ora operativo e, come vedremo tra poco, potete creare il vostro host. Il prossimo passo è quello di registrare un nuovo host, che, come abbiamo visto, è l'indirizzo internet che useremo per accedere al DVR.

Per creare un nuovo host è necessario cliccare su *Gestione Host* nel menu di navigazione in alto e successivamente sul pulsante *Nuovo Host* (DynDNS\5 - dynDNS - gestione host).

A questo punto (DynDNS\6 - dynDNS - nuovo host) dovete scegliere e inserire il vostro nome host e scegliere il dominio tra quelli disponibili. Nel caso illustrato abbiamo scelto il nome *dvr0001* appartenente al dominio *homepc.it*. Senza modificare il campo Indirizzo IP, che riporta l'IP pubblico dal quale vi state connettendo, premete *Crea Host* in fondo alla pagina: una nuova pagina (DynDNS\7 - host creato)

vi darà conferma dell'avvenuta registrazione dell'host. Ora non ci resta che configurare il DVR per mantenere aggiornato l'host appena creato.

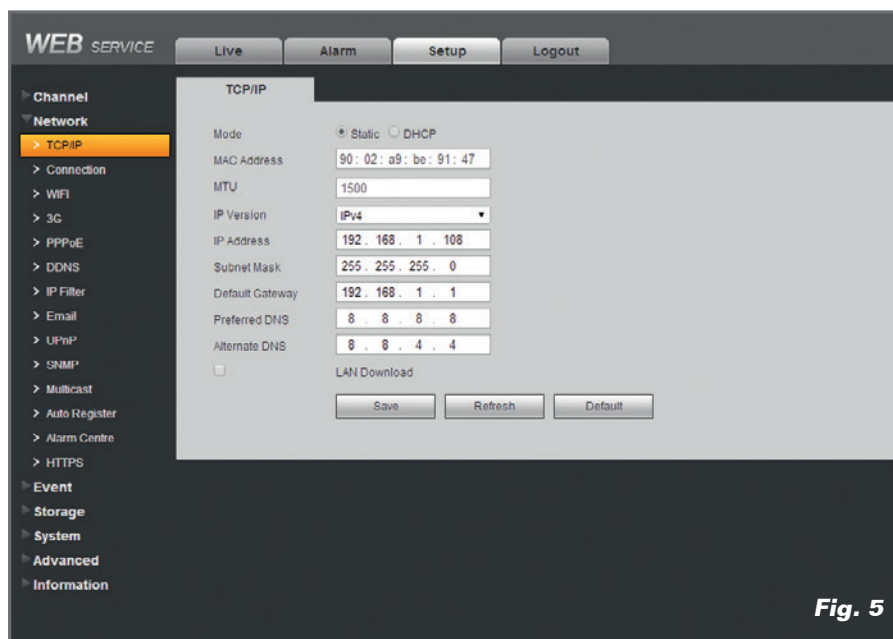
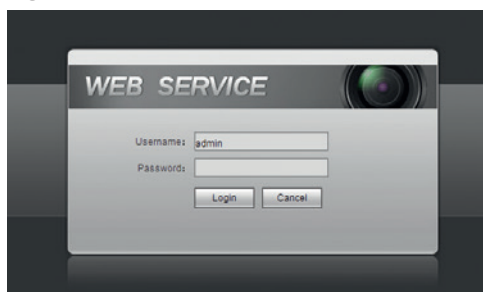
Come abbiamo visto, è fondamentale mantenere allineato il vostro host con l'indirizzo IP pubblico che il provider assegna al vostro router; in due semplici passi vedremo come il DVR FR517 può collegarsi a dynDNS.it sfruttando una modalità compatibile con dyndns.org.

Collegiamoci all'interfaccia web del nostro DVR, aprendo il nostro browser e digitando l'indirizzo IP del dispositivo. Le impostazioni di fabbrica prevedono l'indirizzo 192.168.1.108 con sottomaschera di rete 255.255.255.0 e gateway 192.168.1.1. Queste impostazioni sono adatte alla maggior parte delle reti domestiche e per accedere sarà sufficiente digitare *http://192.168.1.108*.

Apparirà la schermata di login (Fig. 4) e dovremo accedere con l'utente admin; la password di fabbrica è *admin*, ma vi consigliamo di cambiarla, visto che tra poco il vostro DVR sarà accessibile da tutto il mondo.

Eseguito il login, viene visualizzata la schermata principale; in essa, cliccate su *Setup* e accedete alla sezione *Network > TCP/IP* dal menu laterale e verranno visualizzate le impostazioni di rete (Fig. 5). Per attivare la modalità compatibile dovete impostare il campo *Preferred DNS* a

**Fig. 4**



**Fig. 5**

109.232.88.3 e il campo *Alternative DNS* a 109.232.88.4, come illustrato in Fig. 6; premete *Save* per confermare le modifiche.

La modalità compatibile è ora attiva, vediamo ora come attivare l'aggiornamento automatico dell'IP tramite il dynDNS.

Dal menu laterale, cliccate su *Network > DDNS* e selezionate nel campo *DDNS Type* il valore *Dyndns DDNS* (Fig. 7). Mettete il segno di spunta su *Enabled* e inserite i dati del vostro host: nel campo *Domain name* inserite il vostro nome host (*dvr0001.homepc.it*, in questo esempio); nel campo *Username* inserite il nome utente dynDNS.it che avete scelto in fase di registrazione e nel campo *Password* scrivete la password di accesso a dynDNS.it che avete ricevuto via e-mail.

Il campo *Interval* indica ogni quanti secondi il DVR controllerà se c'è stato un cambio di indirizzo IP e, nel caso, invierà l'aggiornamento a dynDNS.it; il valore di 300 secondi (5 minuti) predefinito è un valore tipico e per questa applicazione non serve modificarlo. Ora il DVR è pronto per rilevare il vostro IP pubblico dinamico e comunicarlo al vostro host; per poter accedere serve un ultimo

passaggio, la configurazione del router per il Port Forward.

### Configurazione del router

Ora dobbiamo configurare il router in modo che, quando riceve una richiesta web sul suo indirizzo IP pubblico, deve inoltrarla al vostro DVR.

Vediamo ora come configurare il Port Forward per il DVR FR517 sulla tipica interfaccia dei dispositivi D-Link; sarebbe impossibile in queste pagine mostrare la configurazione su tutti i router in commercio, ma quanto mostrato è facilmente applicabile anche al vostro router.

Per configurare il Port Forward per il vostro DVR su un router D-Link, collegatevi all'interfaccia di amministrazione del router, digitando sul browser *http://192.168.1.1*. Dopo aver inserito i dati di accesso, entrate nella sezione *Advanced* e selezionate dal menu laterale *Virtual Servers*. Potrete configurare più di una regola di inoltro, ma per il nostro scopo è sufficiente impostare una regola che chiameremo DVR0001, impostando *IP Address* a 192.168.1.108; *Public Port* e *Private Port* vanno entrambe impostate al valore 80.

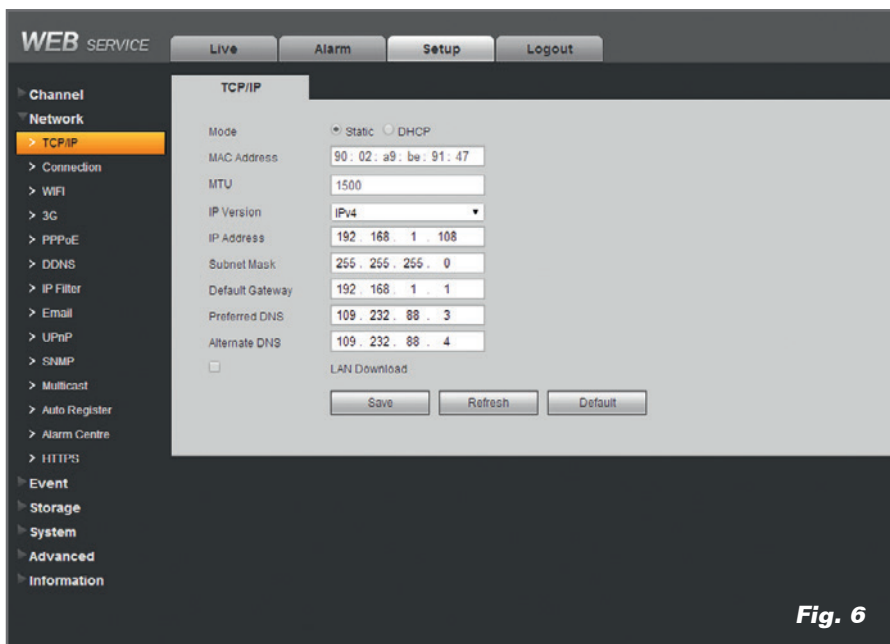


Fig. 6

Per il campo *Protocol*, selezioniamo il valore TCP e salviamo i cambiamenti premendo *Save Settings*. (dlink porta 80)  
Così abbiamo terminato la configurazione e tra poco potrete visualizzare l'interfaccia del DVR.

#### Test e verifica del sistema

Finora abbiamo visto come registrarsi su dynDNS.it, come creare un host, come configurare dynDNS.it sul DVR FR517 e come istruire il vostro router per accedere dall'esterno. È giunto il momento di testare il sistema.

Utilizzando uno smartphone connesso alla rete mobile (o un altro dispositivo esterno alla vostra rete locale) aprite il browser e digitate il vostro nome host <http://dvr0001.homepc.it>. Apparirà la schermata di login del DVR FR517: il sistema è perfettamente funzionante e potete ora controllare il vostro dispositivo di videosorveglianza da ogni parte del mondo tramite Internet. Se qualcosa è andato storto e non è possibile accedere al vostro DVR vedremo, dopo il secondo esempio pratico con la IP camera

CP526W, come diagnosticare il problema.

#### SECONDO ESEMPIO: IP CAMERA CP526W

L'IP camera CP526W è una telecamera Pan-Tilt-Zoom (PTZ) con interfaccia LAN e wireless che vi permette di ispezionare il vostro ambiente anche al buio, grazie all'illuminatore infrarosso di cui dispone. Vedremo ora come configurarla, analogamente a quanto fatto per il DVR FR517, per l'accesso da remoto: dopo aver creato l'host su dynDNS.it procederemo all'impostazione della IP camera e del router.

#### Creazione di un host DynDNS

Per creare un host per la vostra IP camera è sufficiente eseguire il passo 1 e il passo 2 già affrontati per il DVR FR517. In questo esempio, abbiamo creato l'host [camera0001.homepc.it](http://camera0001.homepc.it) e vediamo ora come impostare la IP Camera per l'aggiornamento automatico dell'indirizzo IP pubblico dinamico sul vostro host.

#### Configurazione dell'host sulla IP camera

Per collegarsi all'interfaccia WEB

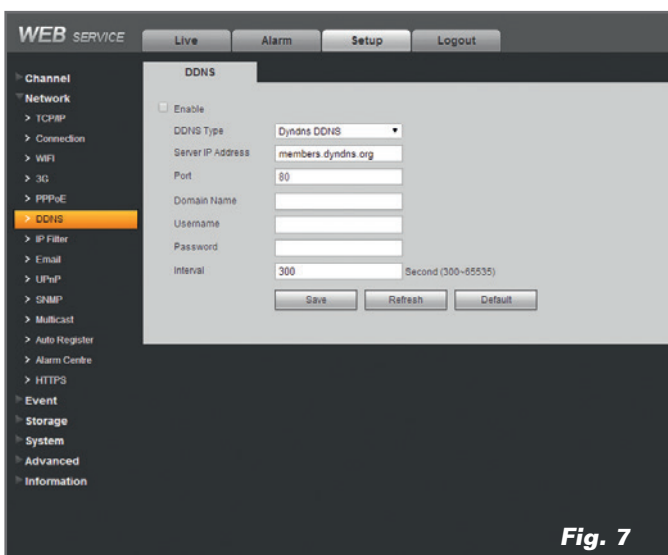


Fig. 7

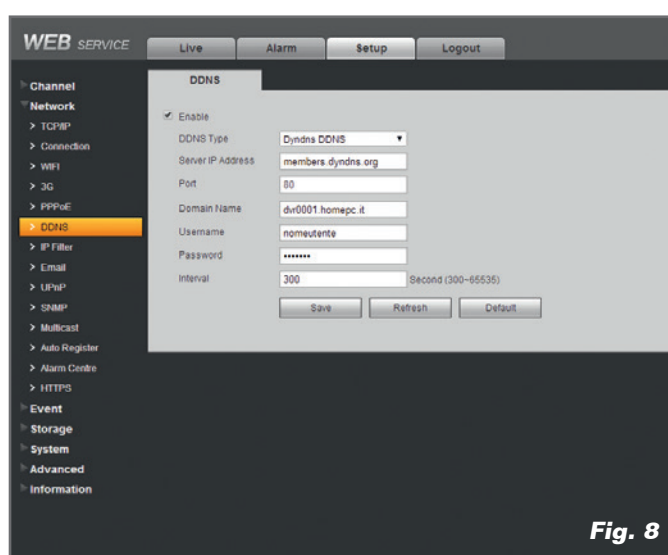


Fig. 8



della vostra IP camera CP526W dobbiamo rilevare il suo indirizzo IP, che viene assegnato tramite DHCP. Se conoscete già l'indirizzo IP potete passare direttamente alle impostazioni di rete per la modalità compatibile.

Inserite nel vostro PC il CD fornito in dotazione con la telecamera e avviate il programma *D:\English\DeviceClient\search tool 3.6.exe* (dove *D:* identifica il vostro lettore CD). Questo utilissimo programma permette di visualizzare le telecamere collegate alla rete, siano esse collegate via WiFi o tramite cavo (Fig. 9). Nella barra laterale, fate doppio click sulla vostra telecamera *IPCAM(192.168.1.245)[81]* e si aprirà il browser visualizzando la maschera di login all'interfaccia web. Vediamo ora come procedere alla configurazione della modalità compatibile e del servizio *dynDNS.it*.

Dopo essersi autenticati come *admin*, senza digitare una password (come da impostazioni di fabbrica) cliccate su *Videostream Mode (for firefox)* visualizzerete la schermata principale di gestione. Qui, cliccate sull'ultima icona sotto lo schermo video, che raffigura

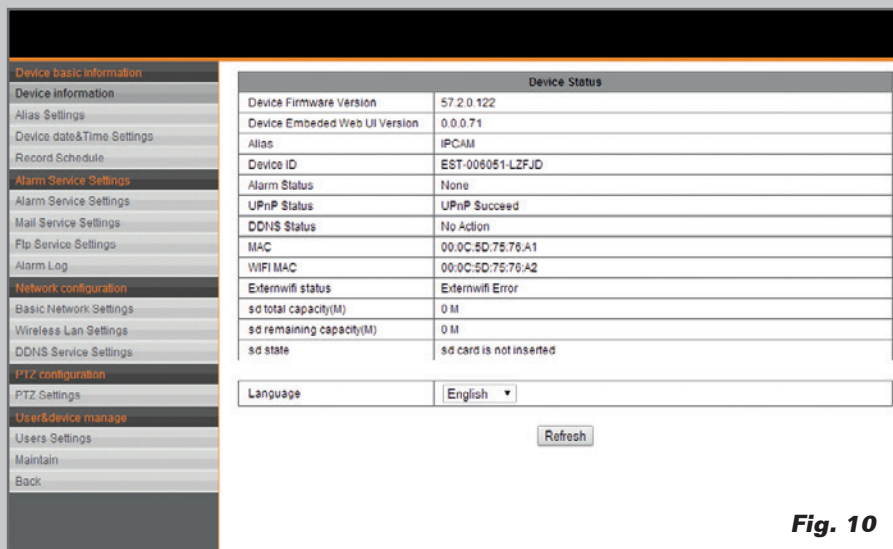


Fig. 10

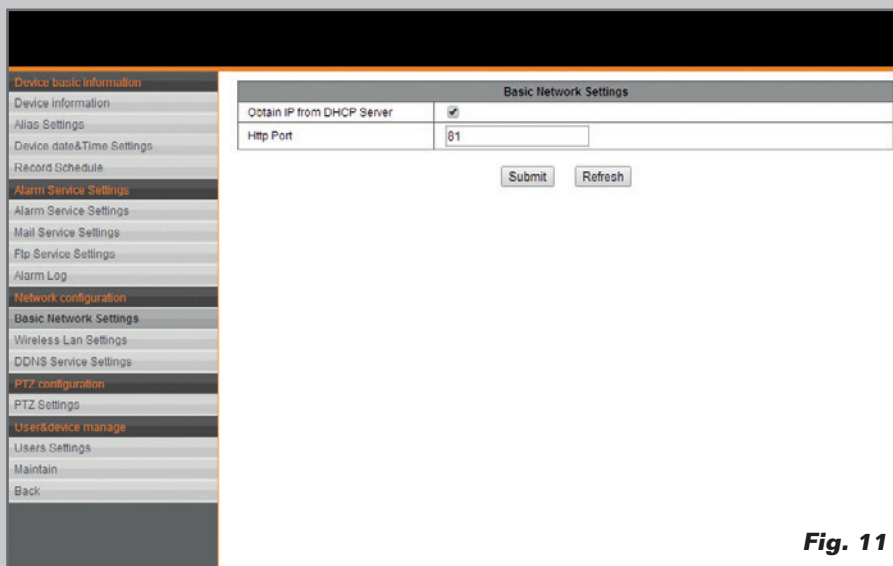


Fig. 11

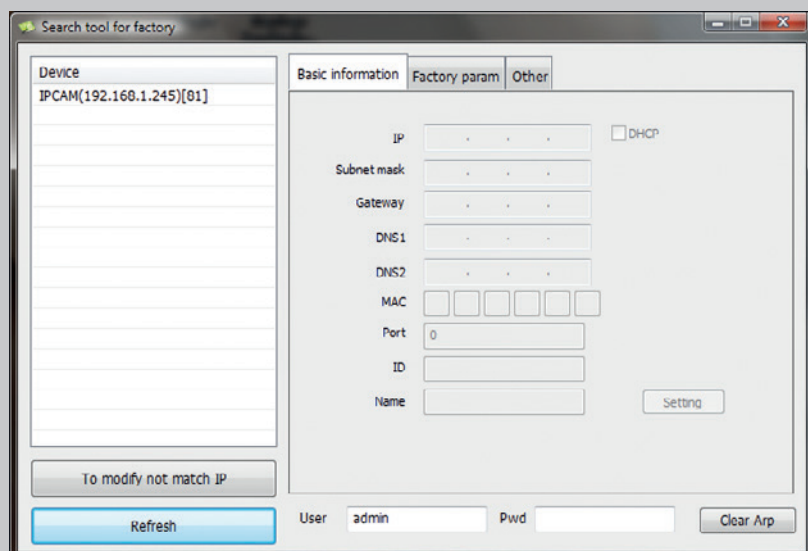


Fig. 9

gli attrezzi, e verrà visualizzata la pagina delle impostazioni (Fig. 10); entrate in *Basic Network Settings*, togliete il segno di spunta da *Obtain IP from DHCP Server* e impostate i DNS della modalità compatibile (Fig. 11); nel campo *DNS Server1* inserite 109.232.88.3, nel campo *DNS Server2* inserite 109.232.88.4 come illustrato in Fig. 12. Ora fate clic sul pulsante *Submit* e attendete il riavvio del dispositivo, avvenuto il quale esso è pronto per essere configurato con *dynDNS.it*. Dopo il riavvio della telecamera, sarà necessario rientrare nella pagina

**Fig. 13**

delle impostazioni, cliccando su *Videostream Mode (for firefox)* e poi sull'ultima icona sotto lo schermo video, che raffigura gli attrezzi; entrate quindi nella sezione *DDNS Service Settings* e nel campo *DDNS Service* selezionate *DynDns.org*: verranno visualizzate le impostazioni.

Come illustrato in **Fig. 13**, inserite nel campo *DDNS User* il vostro nome utente *dynDNS.it* che avete scelto in fase di registrazione al passo 1; inserite nel campo *DDNS Password* la password di accesso che avete ricevuto via email dopo la registrazione. Infine scrivete nel campo *DDNS Host* il nome del vostro host: in questo esempio, *camera0001.homepc.it*. Ora fate clic sul pulsante *Submit*, e, dopo qualche secondo, sul pulsante *Refresh*: verrà visualizzato lo stato *DynDns Succeed http://camera0001.homepc.it:81*.

La vostra IP camera è ora pronta per essere visualizzata dall'esterno. Non resta che configurare il Port Forward sul router.

### Configurazione del router

Abbiamo già visto, nell'esempio pratico del DVR FR517, come impostare il Port Forward su router D-Link. Seguite la medesima procedura, impostando una regola che chiameremo CAME-

RA0001, impostando *IP Address* a 192.168.1.245; *Public Port* e *Private Port* vanno entrambe impostate al valore 81. Per il campo *Protocol*, selezioniamo il valore TCP e salviamo i cambiamenti premendo *Save Settings*. Ora potrete accedere al vostro host.

### Test e verifica

Come per il DVR, utilizzando uno smartphone connesso alla rete mobile (o un altro dispositivo esterno alla vostra rete locale), aprite il browser e digitate il vostro nome host *http://camera0001.homepc.it:81*; è fondamentale aggiungere il suffisso *:81* per indicare al vostro browser che deve connettersi alla porta 81; normalmente, quando non viene specificato nulla viene utilizzata la porta 80, come nel caso del DVR FR 517.

A questo punto vi verranno richiesti il nome utente e la password della IP Camera e subito dopo verrà visualizzata l'interfaccia web.

Tra poco vedremo come diagnosticare eventuali problemi di connessione.

### SE QUALCOSA VA STORTO...

Se non riuscite a collegarvi all'host *dynDNS.it* che avete creato, di seguito troverete alcuni

consigli che vi aiuteranno a diagnosticare la causa del problema.

### Test 1: Verificare

#### *l'accesso tramite indirizzo IP*

Il primo passo per diagnosticare un problema di accesso è provare a collegarsi usando direttamente l'indirizzo IP pubblico, in modo da verificare la configurazione di rete e del Port Forward.

Collegatevi con il vostro browser a *http://checkip.dynDNS.it* e copiate l'indirizzo IP che viene mostrato.

Assumiamo che il vostro IP pubblico sia 93.2.7.65: aprite una nuova finestra del browser e digitate *http://93.2.7.65* (utilizzando il vostro IP pubblico). Se state usando la IP camera e la porta 81, digitate invece *http://93.2.7.65:81*. Se riuscite ad accedere al vostro dispositivo, passate al test successivo; in caso contrario dovete verificare a) di avere un indirizzo IP pubblico, contattando il vostro provider; b) che la configurazione di rete del dispositivo di videosorveglianza sia corretta e che si possa connettere ad Internet; c) che il Port Forward sia correttamente configurato.

### Test 2: Verificare

#### *l'aggiornamento dell'host*

Se il test 1 ha dato esito positivo occorre verificare che l'host *dyndns* sia correttamente aggiornato all'indirizzo IP pubblico attuale. Effettuate una ricerca DNS aprendo il vostro Terminale o prompt dei comandi e digitate *nslookup dvr0001.homepc.it* (utilizzando il vostro nome host). Il comando risponderà indicandovi l'indirizzo del server DNS usato e l'indirizzo IP associato al vostro nome host.

Verificate quindi che nella sezione *Non-authoritative answer (nslookup)* l'indirizzo IP *Address* corrisponda a quello che avete rilevato

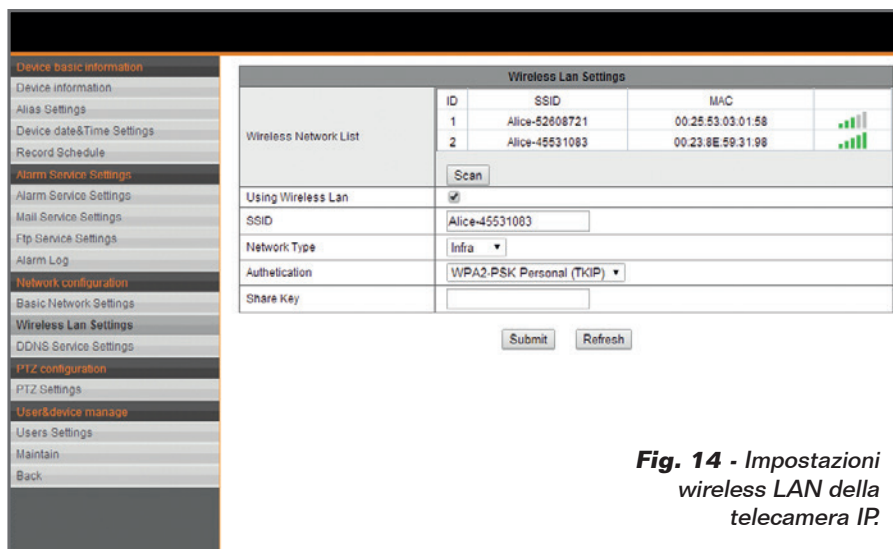
durante il test 1. Se così non fosse, il vostro DVR non sta inviando gli aggiornamenti automatici a dynDNS.it; verificate quindi a) di aver impostato correttamente gli indirizzi dei server DNS della modalità compatibile; b) di aver digitato correttamente il nome host e i vostri dati di accesso a dynDNS.it nel vostro DVR; c) che il vostro host sia correttamente attivo e, ad esempio, non sia stato bloccato per aggiornamenti illeciti.

Per verificare lo stato del vostro host e verificare tutti gli aggiornamenti automatici ricevuti, collegatevi al sito di dynDNS.it, come vedremo tra poco.

#### Controllo degli aggiornamenti e notifiche email

Sull'interfaccia di *Gestione Host* di dynDNS.it, che avete già utilizzato negli esempi pratici al passo 2, potrete facilmente visualizzare tutti gli aggiornamenti inviati dal vostro dispositivo.

Cliccate sull'icona azzurra in fianco al vostro host e visualizzerete la lista degli aggiornamenti automatici inviati dal vostro dispositivo e correttamente ricevuti da dynDNS.it. Troverete anche indicato se il vostro aggiornamento è stato considerato illecito;



**Fig. 14 - Impostazioni wireless LAN della telecamera IP.**

per maggiori informazioni sugli aggiornamenti illeciti visitate la pagina <http://dynDNS.it/support/illegal-update/>

Sempre dalla pagina di Gestione Host, potrete accedere al dettaglio del vostro host, cliccando sul nome *dvr0001.homepc.it* o *camera0001.homepc.it*, dove troverete alcune interessanti impostazioni aggiuntive: la notifica via email del cambio IP, l'impostazione del record MX (per ricevere le email sul vostro host) e tanto altro. Alcune funzionalità potrebbero essere disponibili solo agli utenti con abbonamento; per aggiornare e gestire il vostro abbonamento cliccate sul vostro nome utente in alto a destra.

#### Conclusioni

In questo articolo abbiamo visto come accedere da remoto al vostro DVR FR517 e alla vostra IP camera CP526W; in particolare, abbiamo registrato un account su dynDNS.it e creato un host, l'indirizzo web da usare per collegarvi al vostro device. Abbiamo poi modificato la configurazione di rete per attivare la modalità compatibile e impostato dynDNS.it sul vostro dispositivo di videosorveglianza, in modo che comunichi il vostro indirizzo IP pubblico in automatico. Infine abbiamo configurato il vostro router per permettere l'accesso dell'esterno e verificato che il sistema fosse funzionante. ■

## Elettronica In

**Sei un appassionato di elettronica? Ti piace progettare, costruire, modificare qualsiasi cosa purché abbia a che fare con transistor, display o microcontrollori? Vorresti condividere le tue conoscenze con gli altri lettori?**

**Cosa aspetti? ... Entra subito a far parte dei COLLABORATORI HARDWARE della nostra Rivista**

Realizzare un progetto per Elettronica In è molto semplice!

Dopo aver concordato con la redazione le linee guida del progetto, ti invieremo il materiale necessario con quale farai le prove e realizzerai un primo prototipo, anche su breadboard. Successivamente ci manderai il disegno dello stampato: alla sua realizzazione pratica provvederemo noi che ti rimanderemo la basetta ed i componenti per realizzare il prototipo finale, quello che poi andremo a fotografare e che presenteremo sulle pagine della rivista. Agli schemi elettrici definitivi provvederemo sempre noi mentre a te non rimarrà altro che descrivere il progetto nello stile della rivista.

Manda una mail a: [redazione.hardware@elettronica.in](mailto:redazione.hardware@elettronica.in)

Non perdere questa opportunità!  
**Contattaci oggi stesso!**





# ARDUINO MRF24J40 WIRELESS SHIELD

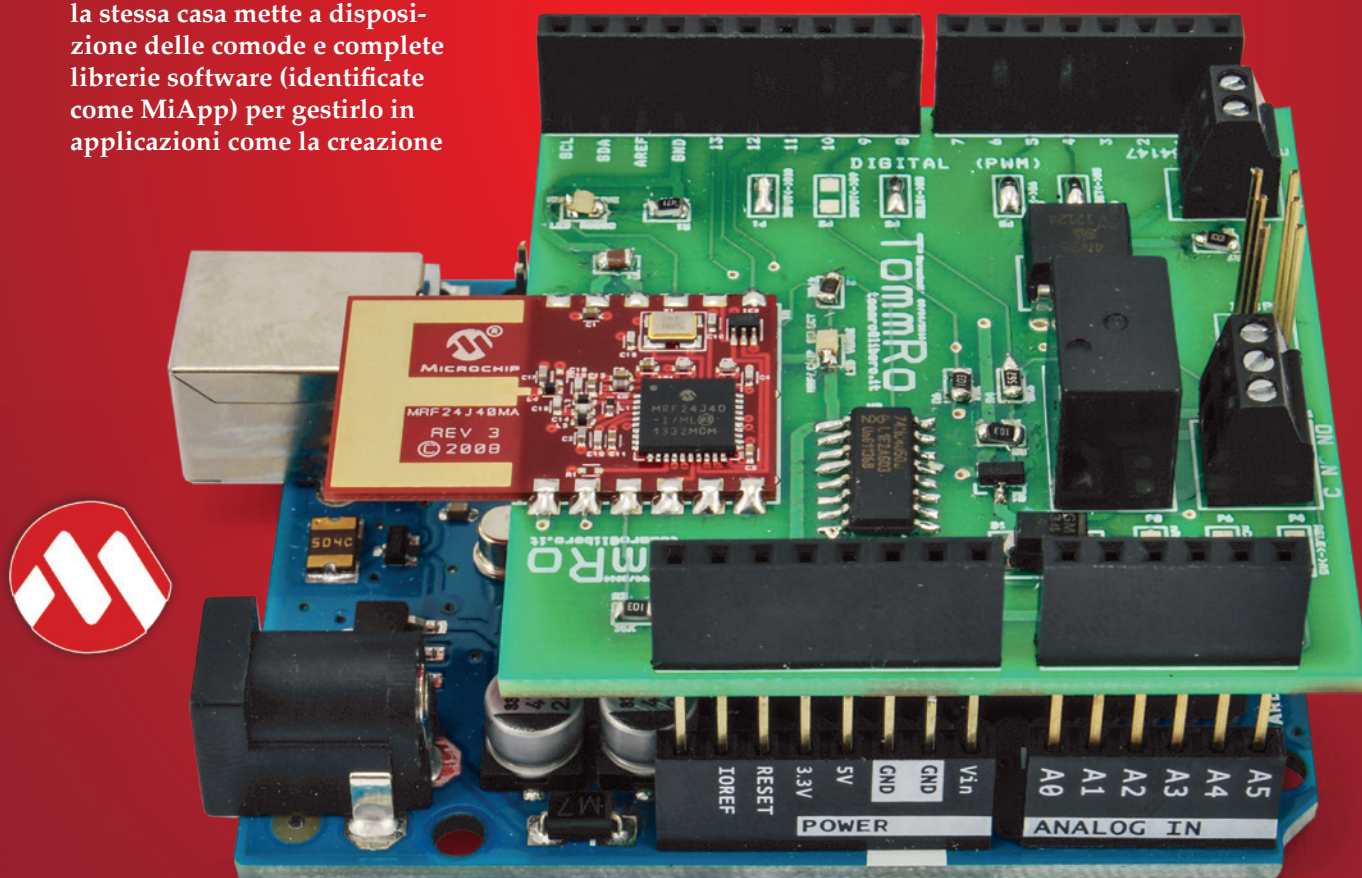
Dotiamo Arduino della connettività wireless e realizziamo un'applicazione dimostrativa dove remotizziamo un relé comandato a distanza da un ingresso mobile e rendiamo virtuale la porta USB seriale di Arduino.

degli Ingg. TOMMASO e ALESSANDRO GIUSTO

**D**i recente vi abbiamo presentato il modulo radio MRF24J40MA, basato sulla tecnologia wireless MiWi e sulle relative librerie software MiApp; come spiegato in quell'occasione, il chip radio è prodotto dalla Microchip e la stessa casa mette a disposizione delle comode e complete librerie software (identificate come MiApp) per gestirlo in applicazioni come la creazione

di una complessa rete wireless, l'identificazione del network, l'indirizzamento dei dispositivi, l'invio e la ricezione dei dati, ecc. Per questo, la prima applicazione che ci è venuta in mente studiando il componente è stata

pensata e sviluppata avendo come target implementativo una MCU Microchip. Chiaramente Microchip fornisce le proprie librerie scritte per l'utilizzo nei propri microcontrollori (PIC) e per il relativo



ambiente di sviluppo (MPLAB con compilatore C). Inoltre rende gratuito l'utilizzo delle stesse a patto che queste siano compilate attraverso MPLAB e per essere eseguite all'interno di una CPU Microchip.

Essendo MiApp scritta in C, tecnicamente sarebbe possibile adattarla e compilarla anche per microcontrollori di altri produttori, tuttavia si violerebbe l'accordo (l'agreement sopra accennato) secondo cui Microchip la concede.

Per quanto riguarda l'hardware, il discorso è differente: la Microchip mette a disposizione il data-sheet di tutti i propri dispositivi (MRF24J40MA incluso) e quindi fornisce tutte le conoscenze di base per la relativa gestione. A questo punto nulla vieta di studiarla completamente il data-sheet e, partendo da zero, scrivervi delle proprie librerie di gestione e comunicazione specifiche per la propria CPU e relativo ambiente di sviluppo.

Da queste considerazioni è nato il progetto che vi presentiamo in queste pagine, il cui scopo finale è riuscire ad utilizzare il modulo wireless MRF24J40MA usando come piattaforma hardware una scheda Arduino (in particolare, la Arduino Uno Rev 3). Il progetto porterà a realizzare uno shield Arduino che consenta di collegare il modulo radio alla CPU di Arduino; definire delle librerie software basilari di gestione del chip radio e fornire un esempio applicativo (basato sulle librerie) che mostri come utilizzare lo shield per realizzare una applicazione reale.

Lo shield sviluppato da noi, oltre ad avere il modulo radio e tutti i relativi componenti per il corretto funzionamento, dispone anche di un'uscita a relé e di un ingresso fotoaccoppiato. Motivo

di questa scelta è dotare lo shield non solo della radio, ma almeno anche di una coppia input/output per interagire con l'esterno. Per quanto riguarda le librerie software per Arduino, quelle finora sviluppate non riescono a coprire appieno tutte le funzionalità offerte da quelle Microchip (consentono di creare diversi tipi di rete, comunicazioni broadcast, ecc...). Al momento le operazioni disponibili sono solo quelle basilari di gestione di una rete WiFi: inizializzazione del modulo, impostazione dell'ID PAN di rete, configurazione dell'indirizzo di rete del nodo, trasmissione e ricezione di dati.

A causa di queste differenze tra sistema Microchip e librerie Arduino, alcune caratteristiche di utilizzo sono cambiate. In particolare, per Arduino non è più presente la distinzione tra nodi Coordinator ed End Device ma gerarchicamente tutti i nodi hanno lo stesso livello ed offrono le stesse funzionalità. Forse si perde qualche performance applicativa, ma sicuramente si conseguono enormi vantaggi (che, a nostro parere, coprono ampiamente le mancanze) dal punto di vista della semplicità e praticità di utilizzo.

#### CHIP HARDWARE WIFI

Il modulo MRF24J40 è stato ampiamente descritto nel fascicolo n° 188 (articolo Demoboard wireless MiWi), quindi in queste pagine ci limitiamo ad esporre alcuni concetti base. Il chip è un transceiver RF compatibile con lo standard IEEE 802.15.4 a 2,4 GHz; nello shield utilizziamo la versione "modulo completo" (MRF24J40MA), che oltre ad integrare il transceiver è già provvisto di antenna accordata, quarzo ed altri componenti esterni necessari per il funzionamento

del WiFi. Inoltre ha un formato meccanico a strip passo 2,54 mm ed ha già superato tutte le certificazioni wireless necessarie.

Per interfacciarsi ad una MCU esterna, dispone di una porta SPI, pin di interrupt, wake e reset; sono inoltre disponibili dei pin per l'alimentazione.

L'accesso alle diverse funzionalità della radio avviene tramite lettura/scrittura di una lunga serie di registri interni del chip (per i dettagli vi rimandiamo al datasheet completo). Le nostre librerie si fanno compito di gestire questi registri e a livello applicativo Arduino nascondono i dettagli implementativi, ma offrono le chiamate generali di configurazione/comunicazione.

#### LIBRERIA ARDUINO MRF24J40

Le librerie MRF24J40 per Arduino che abbiamo scelto, offrono alcune funzioni base di configurazione della rete wireless e relativa comunicazione. Come anticipato, non sono presenti le stesse funzionalità di MiApp Microchip, ma almeno ci sono tutte quelle necessarie per mettere in piedi una rete WiFi, indirizzare i nodi e permettere lo scambio di dati tra gli stessi. La libreria fornisce un oggetto denominato **Mrf24j** da utilizzare per la gestione completa dello stack WiFi. Il costruttore dell'oggetto prevede come input il numero dei pin Arduino da usare per il reset, chip select e interrupt del modulo radio (la porta SPI utilizzata è quella predefinita della scheda Arduino, localizzata ai pin 11, 12 e 13, almeno per la Arduino Uno). È presente una funzione **Reset** che, attraverso il pin di reset definito in precedenza, resetta il chip (buona cosa sarebbe resettare sempre l'hardware all'avvio del software) ed una **Init** (da richiamare dopo la **Reset**) che inizializza



za la porta SPI di comunicazione e i registri base del modulo radio. Per poter funzionare e per potersi scambiare dei messaggi, i nodi necessitano di specificare a quale rete WiFi appartengono (ID PAN, chiaramente lo stesso per tutti) e il proprio indirizzo di rete univoco. Per la configurazione dell'ID PAN della rete wireless, sono presenti due funzioni (**void set\_pan(word panid)** e **word get\_pan(void)**) che rispettivamente scrivono e leggono l'ID PAN da utilizzare; invece per la configurazione dell'indirizzo vanno usate **void address16\_write(word address16)** e **word address16\_read(void)** (dal formato **word** dell'indirizzo si capisce che è una variabile a 16 bit e quindi possono esistere più di 256 nodi).

Altre funzioni di configurazione che possono essere utili sono:

- la **void set\_channel(byte channel)** che permette di specificare quale numero di canale radio usare;
- la **void set\_promiscuous(boolean enabled)** che permette di configurare se utilizzare la ricezione normale oppure promiscua (cioè se si desidera ricevere qualsiasi pacchetto dal canale);
- le **void rx\_enable(void)** ed **void rx\_disable(void)** che rispettivamente abilitano o disabilitano la sezione di ricezione;
- la **void set\_palna(boolean enabled)** che permette di abilitare il controller esterno PA/LNA;
- la **void set\_bufferPHY(boolean bp)** che permette abilitare la ricezione non solo dei dati utente ma anche delle informazioni del payload fisico di comunicazione.

Passiamo ora ad analizzare come viene gestita la trasmissione e ricezione e quali sono le funzioni interessate. La prima cosa da dire è che tutto il meccanismo viene

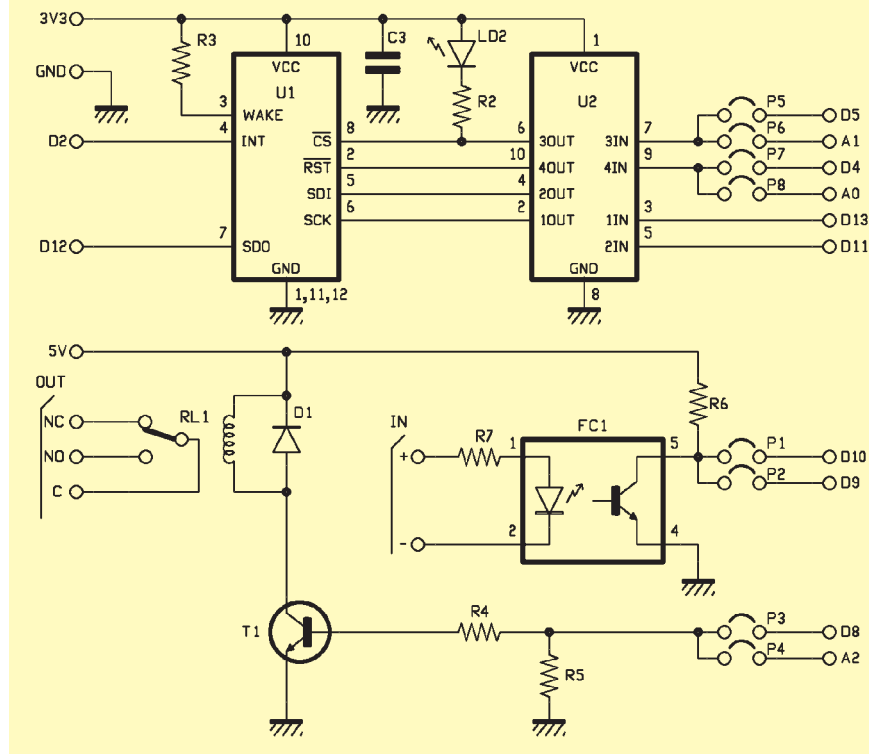
realizzato in un modo un po' particolare utilizzando gli interrupt della CPU di Arduino. Questa implementazione forse è un po' più difficile rispetto al solito ma permette di gestire il tutto in modo più efficiente impegnando la CPU di Arduino solo quando necessario.

Il chip radio dispone di un pin di Interrupt che utilizza per indicare alla CPU che ha dei dati pronti; il software Arduino deve essere predisposto per gestire l'interrupt indicando cosa deve fare in questi casi. Come vedremo meglio più avanti analizzando il sorgente, si utilizza la funzione **attachInterrupt** per indicare ad Arduino quale funzione eseguire quando capita un "change interrupt" del pin radio MRF interrupt.

La libreria software contiene la funzione **void interrupt\_handler(void)** che deve essere richiamata direttamente dal gestore dell'interrupt specificato. La funzione gestisce la lettura di tutti i dati dal modulo radio

e rende "disponibile" quest'ultimo alla ricezione di nuove informazioni. Altra cosa da definire sono due funzioni che rispettivamente si fanno carico di gestire la ricezione e la trasmissione dei dati. Una volta definite queste funzioni, la libreria MRF24J40 offre il metodo **void check\_flags(void (\*rx\_handler)(void), void (\*tx\_handler)(void))** che deve essere richiamato ciclicamente (tipicamente nel metodo **loop()**) e si preoccupa di eseguire correttamente gli handler definiti, evitando sovrapposizione tra le funzioni di ricezione e quelle di trasmissione.

Infine, le funzioni vere e proprie per la trasmissione e ricezione sono rispettivamente **void send16(word dest16, char \* data)** che prevede come parametri l'indirizzo di rete del nodo destinazione e il puntatore al buffer dei dati da trasmettere e **rx\_info\_t \* get\_rxinfo(void)** che ritorna un puntatore ad una struttura di tipo **rx\_info\_t** appositamente definita per incapsulare tutte



le informazioni utili in fase di ricezione.

### SCHEMA ELETTRICO

Iniziamo l'analisi dello schema elettrico dello shield wireless; l'hardware del dispositivo è basato sul modulo WiFi MRF24J40MA (integrato U1). Il chip radio richiede di essere gestito da una CPU principale e per farlo mette a disposizione una porta SPI hardware attraverso cui riceve i comandi e trasmette le risposte. La porta SPI si trova sui pin 5, 6 e 7 che dovranno essere collegati ai pin della porta SPI di Arduino (SCK, MISO e MOSI). Il modulo radio richiede una tensione di alimentazione di 3,3V, mentre le porte digitali di Arduino sono compatibili con livelli TTL (a 5V), pertanto non è possibile eseguire un collegamento diretto ma è necessario utilizzare un meccanismo di traslazione dei livelli. La nostra scelta è caduta su un integrato ideato per questo tipo di lavoro,

cioè il chip 74HC4050D (integrato U2 nello schema). Come vedremo più avanti, tutti i pin di collegamento tra Arduino e MRF24J40MA hanno queste caratteristiche e pertanto tutti vengono "fatti passare" attraverso il 74HC4050D.

Oltre alla porta SPI, il chip radio prevede dei pin di Wake, Interrupt, Reset e Chip Select. Il pin di Wake è collegato fisso al livello di tensione positivo, in quanto abbiamo scelto di tenere sempre attivo il modulo radio.

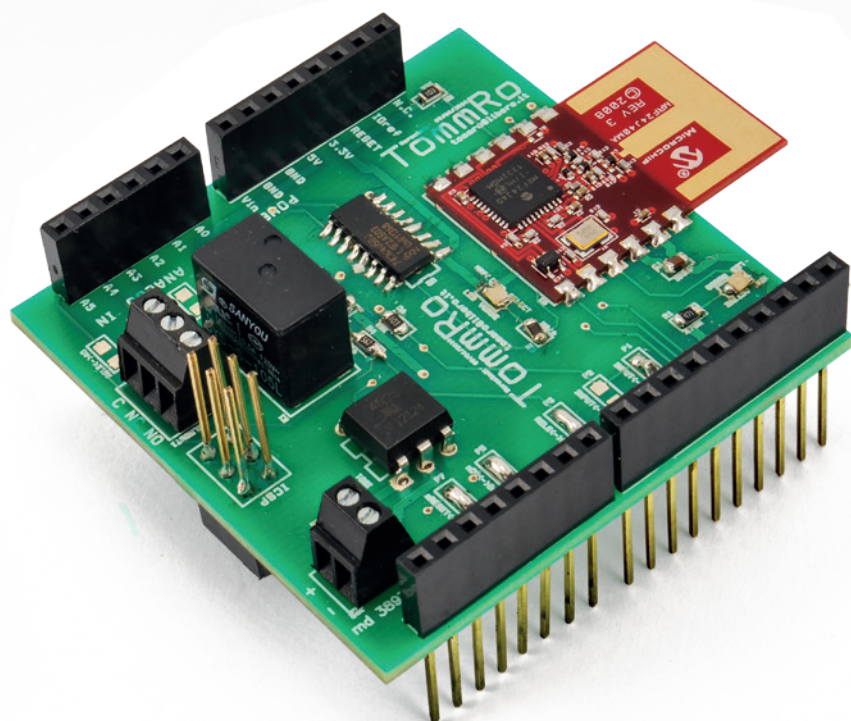
Il pin di Interrupt è collegato alla porta digitale D2 di Arduino; questo perché il pin D2 può essere gestito da Arduino tramite interrupt e, come vedremo meglio analizzando il sorgente dello sketch, questo meccanismo permette una gestione più efficiente delle risorse. Gli altri due pin (Reset e Chip Select) possono invece essere gestiti da qualsiasi porta digitale di Arduino; per lasciare una certa dose di libertà di scelta (in modo da adattarsi ad

eventuali ulteriori shield), abbiamo predisposto due ponticelli a saldare per ogni pin; chiudendo con una goccia di stagno uno di essi, è possibile effettuare la propria scelta.

Terminando l'analisi relativa al modulo radio facciamo notare il condensatore C3 da 100nF collegato all'alimentazione del chip WiFi (come richiesto dal data-sheet) e la presenza di un LED (LED verde nello schema) collegato al pin chip select dell'MRF24J40MA stesso (questo per avere la segnalazione visiva della presenza di comunicazione). Questo è quanto riguarda la parte radio; passiamo adesso alla circuiteria riguardante gli input ed output di cui lo shield è dotato. Per quanto riguarda l'ingresso (IN) è fotoaccoppiato mediante l'optoisolatore 4N25, il cui fototransistor di uscita risulta interdetto (quindi il collettore si trova a livello logico alto) quando all'IN manca tensione o c'è una differenza di potenziale minore di 2 V, mentre va in saturazione (collettore a 0 logico) quando l'IN è alimentato con più di 2 Vcc. L'uscita dello shield è invece costituita da un relé (RL1, pilotato da uno stadio BJT T1). Anche per quanto riguarda questi I/O, abbiamo deciso di lasciare una certa dose di libertà su quale porta di Arduino usare: l'assegnazione della linea che legge l'uscita del 4N25 si effettua con i jumper P1 e P2, mentre quella della linea di Arduino che comanda T1 si effettua con i ponticelli P3 e P4.

### SKETCH ARDUINO

Avendo analizzato in precedenza le librerie Arduino per il chip MRF24J40, risulta ora più semplice analizzare il sorgente dello sketch di esempio. Scopo dell'esempio è mostrare come



## Listato 1 – SKETCH DI ESEMPIO

realizzare una rete wireless in cui “convivono” più nodi WiFi (ciascuno identificato da un proprio indirizzo univoco) e permetterne lo scambio dei messaggi. Per il sorgente dell’articolo abbiamo deciso di “abbinare” due nodi specifici ed inviare da una scheda all’altra tutti i caratteri ricevuti sulla porta seriale e viceversa. In più esiste una sorta di “remotizzazione” di I/O, nel senso che lo stato dell’input di una scheda viene replicato sull’output dell’altra e viceversa.

Pur avendo due schede distinte, il sorgente da caricare su entrambe è fondamentalmente lo stesso: l’unica cosa da tenere presente è che le schede devono avere indirizzi di rete distinti e che l’indirizzo di rete di un nodo dovrà essere il destinatario dell’altro e viceversa.

Analizziamo ora il codice (**Listato 1**): partendo dall’inizio, includiamo le librerie necessarie (**SPI** e **mrf24j** rispettivamente per porta SPI e modulo radio); poi definiamo i collegamenti tra pin di Arduino e MRF24J40MA e i pin usati per gli I/O hardware: per il reset abbiamo deciso di utilizzare D4, per il chip select il pin D5, per l’interrupt (come descritto in precedenza) siamo “obbligati” ad usare D2. Per comandare l’uscita a relé utilizziamo il pin D8 e per leggere l’ingresso fotoaccoppiato, il pin D10. Nelle righe seguenti definiamo il codice PAN identificativo della rete wireless (istruzione **const word PANWireless = 0x1235**), l’indirizzo di rete del nodo (istruzione **const word thisNodeAddress = 0x6001**) ed infine l’indirizzo di rete del nodo destinatario dei nostri messaggi (istruzione **const word destNodeAddress = 0x6002**). In seguito definiamo l’oggetto per la gestione delle funzionalità del modulo radio (istruzione

```

/*****
*****
* Titolo:      Mrf24j40ShieldExample01
* Autori:      Ingg. Tommaso Giusto e Alessandro Giusto
* Data:        26/06/2014
* Email:       tommro@libero.it
*****
*****/

/*****
*****
* Inclusione librerie
*****
*****/
*/
#include <SPI.h>
#include <mrf24j.h>

/*****
*****
* Definizione pin I/O Arduino
*****
*****/
// Pin reset modulo radio
const int pinResetMRF = 4;
// Pin chip select modulo radio
const int pinCSMRF = 5;
// Pin interrupt modulo radio
const int pinIntMRF = 2;
// Pin rele MRF shield
const int pinShieldRele = 8;
// Pin input MRF shield
const int pinShieldInput = 10;

// PAN rete wireless
const word PANWireless = 0x1235;

// Indirizzo proprio nodo rete wireless
const word thisNodeAddress = 0x6001;
//const word thisNodeAddress = 0x6002;

// Indirizzo nodo destinazione rete wireless
const word destNodeAddress = 0x6002;
//const word destNodeAddress = 0x6001;

/*****
*****
* Variabili programma
*****
*****/

// Oggetto Mrf24j shield
Mrf24j mrfShield(pinResetMRF, pinCSMRF, pinIntMRF);

// Ultimo stato input
int lastStatoInput;

/*****
*****
* Codice programma
*****
*****/

// Inizializzazione Scheda
void setup() {
  // Inizializzo I/O
  pinMode(pinShieldRele, OUTPUT);
  pinMode(pinShieldInput, INPUT);
  // Disattivo rele
  digitalWrite(pinShieldRele, LOW);
  // Leggo valore corrente input
  lastStatoInput = LOW;

  // Avvio porta seriale
  Serial.begin(9600);
}

```

(Continua)



## Listato 1 – SKETCH DI ESEMPIO (segue)

```
// Resetto oggetto Mrf24j shield
mrfShield.reset();
// Inizializzo oggetto Mrf24j shield
mrfShield.init();

// Imposto PAN rete wireless
mrfShield.set_pan(PANWireless);
// Imposto indirizzo nodo rete wireless
mrfShield.address16_write(thisNodeAddress);

// uncomment if you want to receive any packet on this channel
//mrfShield.set_promiscuous(true);

// uncomment if you want to enable PA/LNA external control
//mrfShield.set_palna(true);

// uncomment if you want to buffer all PHY Payload
//mrfShield.set_bufferPHY(true);

// Collego la funzione da eseguire quando capita change interrupt
// del pin MRF interrupt
// 0 -> corrisponde ad Arduino pin 2 (INT0 per Arduino ATmega8/168/328)
attachInterrupt(0, MRFInterruptRoutine, CHANGE);
interrupts();
}

// Funzione esecuzione MRF interrupt
void MRFInterruptRoutine() {
    // Passo controllo routine interrupt oggetto Mrf24j shield
    mrfShield.interrupt_handler();
}

// Programma Principale
void loop() {
    int statoInput;
    unsigned long currentMillis;
    char serialBuffer[105];
    int index;
    int flag;

    // Richiamo funzione gestione handler ricezione/trasmmissione
    mrfShield.check_flags(&handleRx, &handleTx);

    // Leggo stato input
    statoInput = digitalRead(pinShieldInput);

    // Se variato stato input
    if (statoInput != lastStatoInput) {
        // Invio lo stato (protocollo di comunicazione)
        if (statoInput == HIGH)
            mrfShield.send16(destNodeAddress, "INPUT0");
        else
            mrfShield.send16(destNodeAddress, "INPUT1");

        // Aggiorno variabile ultimo stato
        lastStatoInput = statoInput;
    }

    // Se dati ricevuti sulla porta seriale
    if (Serial.available() > 0) {
        // Inizializzo il buffer comando
        serialBuffer[0] = 'S';
        serialBuffer[1] = 'E';
        serialBuffer[2] = 'R';
        serialBuffer[3] = 'I';
        serialBuffer[4] = 'A';
        index = 5;
        // Memorizzo i caratteri ricevuti
        for (;;) {
            // Se nuovo carattere disponibile
            if (Serial.available() > 0) {
                // Accodo il carattere
                serialBuffer[index] = (char) (Serial.read());
                index++;
            }
        }
    }
}
```

**Mrf24j mrfShield(pinResetMRF, pinCSMRF, pinIntMRF)**): come si vede dal listato, in fase di creazione dell'oggetto è necessario indicare quali pin utilizzare per gestire il MRF24J40MA.

Prima della funzione **setup** definiamo una variabile (**int lastStatoInput**) che ci servirà per rilevare i cambiamenti di stato dell'ingresso fotoaccoppiato; invece nella **setup** impostiamo che il pin per il relé è un'output, mentre quello per l'ingresso è un input, ed inizializziamo la porta seriale con baud rate a 9.600. In seguito resettiamo (istruzione **mrfShield.reset()**) e inizializziamo (istruzione **mrfShield.init()**) l'oggetto per la comunicazione wireless. Di seguito scriviamo nel modulo radio l'identificativo PAN della rete (istruzione **mrfShield.set\_pan(PANWireless)**) e l'indirizzo del nodo (istruzione **mrfShield.address16\_write(thisNodeAddress)**).

La cosa che manca per terminare la configurazione è impostare la gestione dell'interrupt: con l'istruzione **attachInterrupt(0, MRFInterruptRoutine, CHANGE)** colleghiamo la funzione da eseguire quando accade l'interrupt del pin MRF Int: in particolare, 0 corrisponde al pin 2 di Arduino (INT0 per Arduino ATmega8/168/328); **MRFInterruptRoutine** è la nostra routine di interrupt (la vedremo a breve) ed infine **CHANGE** indica il tipo di interrupt richiesto (cambiamento). Appena sotto, definiamo la nostra routine di interrupt (funzione **void MRFInterruptRoutine()**) dove semplicemente richiamiamo la routine di interrupt dell'oggetto **Mrf24j shield** (istruzione **mrfShield.interrupt\_handler()**).

A questo punto entriamo nel **main** (funzione **void loop()**)

## Listato 1 – SKETCH DI ESEMPIO (segue)

del programma: per prima cosa, come richiesto dalla libreria del chip radio, richiamiamo la funzione **mrfShield.check\_flags(&handleRx, &handleTx)** che si occupa di gestire gli handler di ricezione e trasmissione (definiremo le funzioni **handleRx** e **handleTx** più avanti nel codice). In seguito leggiamo lo stato corrente dell'ingresso e, se variato, inviamo lo stato corrente sulla radio (istruzione **mrfShield.send16(destNodeAddress, "INPUT0")** in caso di input basso). Poi verifichiamo se sono presenti dati sulla porta seriale e, in caso affermativo, li leggiamo e sempre tramite la stessa istruzione li inviamo al nodo destinatario. A questo punto la funzione loop è terminata; passiamo a vedere i due handler di ricezione e trasmissione: il primo corrisponde alla funzione **void handleRx()** che verrà richiamata solo in caso di avvenuta ricezione di dati dalla radio. Qui dentro, la prima cosa che verifichiamo è che i dati siano validi secondo il protocollo da noi definito (istruzione **if (mrfShield.rx\_datalength() > 5)**) la quale, appunto, verifica che siano stati ricevuti almeno 5 byte (il numero minimo che prevede il protocollo affinché un comando sia valido); in caso affermativo leggiamo i dati dalla sezione radio (tramite l'istruzione **mrfShield.get\_rxinfo()**) accediamo alla struttura che contiene i dati e tramite **mrfShield.get\_rxinfo()->rx\_data[]** accediamo ai singoli byte del buffer di ricezione; altri campi della struttura che possono essere utili sono **mrf.get\_rxinfo()->lqi** e **mrf.get\_rxinfo()->rssi** che contengono due indici sulla qualità del segnale radio), verifichiamo quale comando sia stato ricevuto e ci comportiamo di conseguen-

```
// Indico ricevuto carattere
flag = 0;
}
// Se nuovo carattere non disponibile
else {
    // Se già non ricevuto carattere
    if (flag == 1)
        // Termino ciclo for memorizzo i caratteri ricevuti
        break;

    // Indico non ricevuto carattere
    flag = 1;
    // Attesa ricezione prossimo carattere
    delay(5);
}
}
// Aggiungo fine stringa
serialBuffer[index] = (char) ('\0');
// Invio il comando
mrfShield.send16(destNodeAddress, serialBuffer);
}

// Handler eseguito alla ricezione di dati dalla radio
void handleRx() {
    int i;

    // Se ricevuto comando valido
    if (mrfShield.rx_datalength() > 5) {
        // Se ricevuto comando INPUT
        if ((mrfShield.get_rxinfo()->rx_data[0] == 'I') &&
            (mrfShield.get_rxinfo()->rx_data[1] == 'N') &&
            (mrfShield.get_rxinfo()->rx_data[2] == 'P') &&
            (mrfShield.get_rxinfo()->rx_data[3] == 'U') &&
            (mrfShield.get_rxinfo()->rx_data[4] == 'T')) {
            // Imposto stato rele
            if (mrfShield.get_rxinfo()->rx_data[5] == '0')
                digitalWrite(pinShieldRele, LOW);
            else
                digitalWrite(pinShieldRele, HIGH);
        }
        // Se ricevuto comando SERIA
        else if ((mrfShield.get_rxinfo()->rx_data[0] == 'S') &&
            (mrfShield.get_rxinfo()->rx_data[1] == 'E') &&
            (mrfShield.get_rxinfo()->rx_data[2] == 'R') &&
            (mrfShield.get_rxinfo()->rx_data[3] == 'I') &&
            (mrfShield.get_rxinfo()->rx_data[4] == 'A')) {
            // Invio i dati sulla seriale
            for (int i = 5; i < mrfShield.rx_datalength(); i++)
                Serial.write(mrfShield.get_rxinfo()->rx_data[i]);
        }
    }
}

// Handler eseguito per trasmissione di dati sulla radio
void handleTx() {
}
```

za (cambiando il pin di gestione del relé oppure inviando i dati sulla seriale). Per finire l'analisi ci manca da vedere l'handler di trasmissione (funzione **void handleTx()**) che verrà richiamato al termine di ogni trasmissione. Nel nostro specifico esempio l'handler è vuoto (perché non dobbiamo fare alcuna operazione

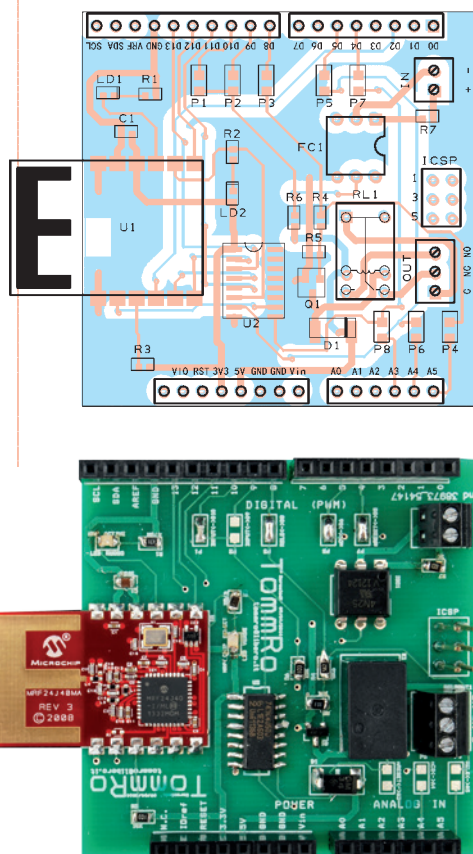
quando la trasmissione viene terminata) ma in questa funzione è possibile identificare se la trasmissione ha avuto esito positivo o negativo (ed eventualmente provare a ritrasmettere). Per fare questo, la libreria offre la struttura **mrf.get\_txinfo()** che contiene tutte le informazioni relative alla trasmissione; in particolare, il

## Elenco Componenti:

R1: 470 ohm (0805)  
 R2: 470 ohm (0805)  
 R3: 10 kohm (0805)  
 R4: 5,6 kohm (0805)  
 R5: 10 kohm (0805)  
 R6: 10 kohm (0805)  
 R7: 1 kohm (0805)  
 C1: 100 nF ceramico (0805)  
 LD1: LED verde (0805)  
 LD2: LED rosso (0805)  
 U1: MRF24J40MA  
 U2: 74HC4050D  
 FC1: 4N25  
 D1: GF1M  
 RL1: Relé miniatura 5V

### Varie:

- Zoccolo 3+3
- Morsetto 2 poli passo 2,54mm
- Morsetto 3 poli passo 2,54mm
- Strip Mascio/Femmina 3 vie (2 pz.)
- Strip Mascio/Femmina 6 vie
- Strip Mascio/Femmina 8 vie (2 pz.)
- Strip Mascio/Femmina 10 vie
- Circuito stampato



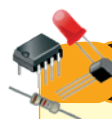
campo `mrf.get_txinfo()` -> `tx_ok` indica se la trasmissione è andata a buon fine, altrimenti significa che la trasmissione è fallita (non è stata ricevuta la risposta di conferma dal destinatario). Altri campi di `mrf.get_txinfo()` sono `retries`, che indica il numero di tentativi di trasmissione effettuati e `channel_busy`, che invece indica se la trasmissione è fallita causa canale occupato.

## REALIZZAZIONE PRATICA

Per testare l'esempio applicativo è necessario avere almeno due sistemi Arduino + MRF24J40 Shield. Come per la versione con PIC Microchip, a livello hardware i due sistemi sono identici, ma differisce il firmware di gestione (in particolare la definizione dell'indirizzo di rete del nodo e del destinatario). Le schede sono

state progettate in tecnologia SMD, ma prestando una certa attenzione è possibile montarle a mano (chi non fosse in grado, presso Futura Elettronica troverà gli shield già montati). Come è buona regola, partite dai componenti in SMD più piccoli (usando una pinzetta per tenere il componente mentre si salda) per terminare poi con quelli in formato tradizionale. Il modulo wireless va saldato direttamente alla scheda mantenendo la parte con l'antenna all'esterno del circuito stampato (riferitevi alle immagini presenti nell'articolo). Prima di terminare il montaggio, decidete quali pin di Arduino utilizzare per la gestione degli I/O e saldate i relativi ponticelli. Per la programmazione di Arduino si utilizza il solito IDE, ma prima è necessario modificare i

sorgenti in modo da adattarsi ai pin hardware decisi e modificare gli indirizzi di rete dei due nodi. In particolare, andranno modificate le variabili `pinResetMRF`, `pinCSMRF`, `pinShieldRele` e `pinShieldInput` per quanto riguarda l'hardware; `thisNodeAddress` e `destNodeAddress` servono per definire gli indirizzi dei nodi sorgente/destinazione. A questo punto siete pronti per accendere il sistema e provarlo. Provate ad alimentare IN di una scheda e verificate che scatti il relé dell'altra. Infine collegate con diversi cavi USB un PC alle schede; aprite diversi emulatori di terminale (HyperTerminal o simile) ognuno "collegato" a una diversa porta COM virtuale, per tutti selezionate la configurazione 9.600, 8, N, 1 e nessun controllo di flusso e verificate che "scrivendo" dei dati in una finestra (ricordatevi che se non avete l'echo locale abilitato non ne vedete la digitazione) questi vengano ricevuti e trasmessi sulla porta seriale da tutte le schede.



## per il MATERIALE

I componenti utilizzati in questo progetto sono di facile reperibilità. Il master del circuito stampato può essere scaricato dal sito della rivista così come il firmware per Arduino. Il modulo WiFi MRF24J40MA-I/RM è disponibile al prezzo di 12,00 Euro; lo shield (FT1150M) è disponibile in versione montata e collaudata a 38,00 Euro. I prezzi si intendono IVA compresa.

Il materiale va richiesto a:  
 Futura Elettronica, Via Adige 11,  
 21013 Gallarate (VA)  
 Tel: 0331-799775 - Fax: 0331-792287  
<http://www.futurashop.it>



Anteprima

# TEENSY 3.1

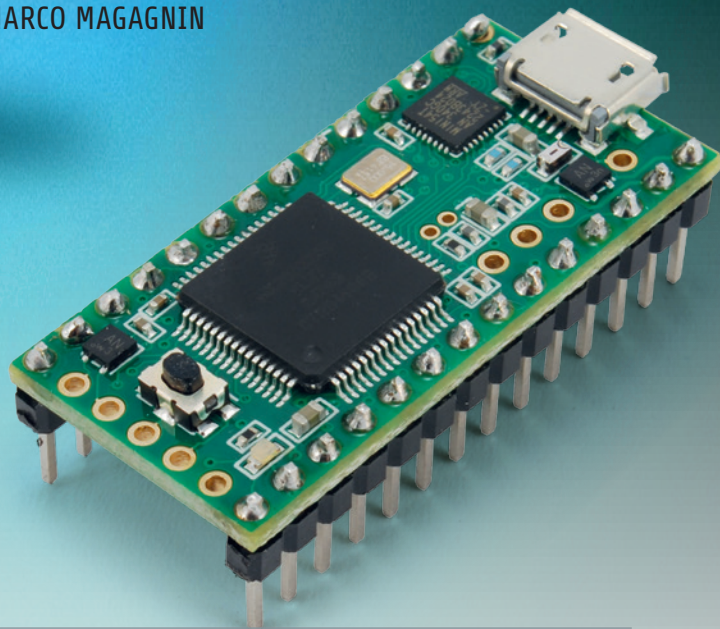
## MICROCONTROLLORE TUTTOFARE COMPATIBILE ARDUINO

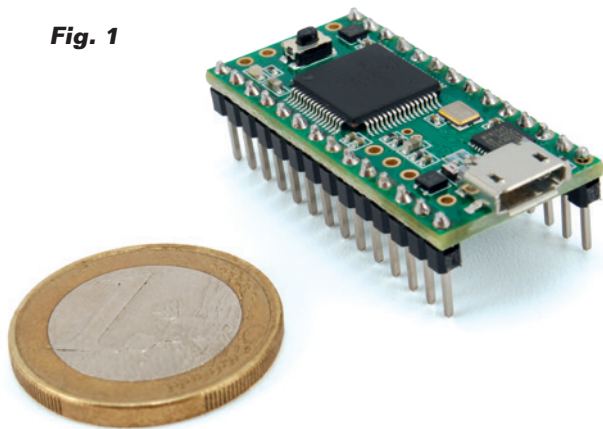
Potente e ricco di funzionalità, è nato per le applicazioni "affamate" di risorse, che necessitano di interfacciarsi a sistemi a 3,3V ed è dotato di un proprio RTC e di convertitori ADC e DAC di precisione.

di MARCO MAGAGNIN

**D**a tempo stiamo seguendo lo sviluppo della famiglia di moduli a microcontrollore Teensy, prodotti dalla Pjrc ([www.pjrc.com/teensy](http://www.pjrc.com/teensy)). Pur avendo sempre considerato interessanti le caratteristiche di tali dispositivi, non abbiamo pensato di presentarli

su Elettronica In fino a quando, alla ricerca di un microcontrollore da utilizzare in un progetto di integrazione con le schede GNU/Linux embedded, non ci è capitato tra le mani l'ultimo prodotto della famiglia: il Teensy 3.1. Le sue caratteristiche



**Fig. 1**

sono particolarmente interessanti, a nostro parere, proprio nel caso lo si voglia impiegare in progetti di integrazione con sistemi eterogenei, in particolare collegati a schede con GNU/Linux embedded come Raspberry Pi, Banana Pi, la Arietta G25 (che presentiamo in questo stesso numero) e pressoché tutte le altre schede disponibili sul mercato.

La nostra ricerca era finalizzata a individuare un modulo a microcontrollore con un chip più potente di quelli generalmente reperibili, sia in termini di "word" che di "clock", che fosse compatibile sia con i livelli a 5V che a 3,3V; questo per evitare problemi con le conversioni di livello. Doveva offrire il maggior numero possibile di pin di I/O, un buon convertitore ADC, il maggior numero di "bus" di comunicazione.

Vi sembra abbastanza? A noi no: serviva anche di piccole dimensioni. Se poi fosse stato anche "compatibile" con la logica di sviluppo di Arduino, avremmo risparmiato tempo e fatica riutilizzando codice e librerie già disponibili.

Cerca che ti cerca, il piccolo circuito stampato che ci siamo trovati tra le mani offre un processore Cortex M4 a 32 bit e 72 MHz di clock (overclock fino a 96 MHz), 34 pin di I/O a 3,3V che tollerano i 5V, compatibile con il linguaggio e l'IDE di sviluppo di Arduino. Le dimensioni? Le potete vedere nella foto di **Fig. 1**. Le caratteristiche? Le riepiloga la **Tabella 1**. Analizziamo alcune delle peculiarità della scheda.

### PROCESSORE

L'adozione del processore Cortex M4, a 32 bit con un clock fissato a 72 MHz, ma configurabile fino a 96 MHz, permette notevoli velocità di campionamento e il trattamento di segnali, rendendo il Teensy 3.1 utilizzabile anche in progetti dove sono necessarie prestazioni avanzate, come il trattamento di segnali audio, anche stereo.

### MEMORIA

La RAM è da 64 K, quindi nessun problema di limiti di utilizzo di librerie e dimensioni dei programmi.

**Tabella 1**

| Caratteristica         | Teensy 3.1          | Unità di misura |
|------------------------|---------------------|-----------------|
| Processore             | MK20DX256VLH7       |                 |
| CPU                    | Cortex-M4           |                 |
| Clock predefinito      | 72                  | MHz             |
| Overclockable          | 96                  | MHz             |
| Memoria Flash          | 256                 | kbyte           |
| Velocità               | 192                 | Mbyte/sec       |
| Cache                  | 256                 | Byte            |
| RAM                    | 64                  | kbyte           |
| EEPROM                 | 2                   | kbyte           |
| Direct Memory Access   | 16                  | Canali          |
| I/O Digitali           | 34                  | Pin             |
| Tensione in uscita     | 3.3V                | Volt            |
| ..Tensione in ingresso | 3.3V - 5V Tollerati | Volt            |
| Ingressi Analogici     | 21                  | Pin             |
| Convertitori           | 2                   |                 |
| Risoluzione            | 16                  | Bit             |
| Utilizzabili           | 13                  | Bit             |
| Ampl. Guadagno         | 2                   |                 |
| Touch Sensing          | 12                  | Pin             |
| Comparatori            | 3                   |                 |
| Uscite Analogiche      | 1                   | Pin             |
| Risoluzione DAC        | 12                  | Bit             |
| Timer                  | 12 in totale        |                 |
| FTM Type               | 3                   |                 |
| Uscite PWM             | 12                  | Pin             |
| PDB Type               | 1                   |                 |
| CMT (infrarossi) Type  | 1                   |                 |
| LPTMR Type             | 1                   |                 |
| PIT (interval) Type    | 4                   |                 |
| Systick                | 1                   |                 |
| RTC (data/ora)         | 1                   |                 |
| Comunicazione          |                     |                 |
| USB                    | 1                   |                 |
| Seriale                | 3                   |                 |
| With FIFOs             | 2                   |                 |
| High Res Baud          | 3                   |                 |
| Fast Clock             | 2                   |                 |
| SPI                    | 1                   |                 |
| With FIFOs             | 1                   |                 |
| I2C                    | 2                   |                 |
| CAN Bus                | 1                   |                 |
| I2S Audio              | 1                   |                 |
| Dimensione FIFO        | 8                   |                 |

Si possono affrontare anche applicazioni che richiedono la gestione di icone e grafica su piccoli schermi LCD a colori o la gestione di applicazioni con grandi tabelle o di effetti audio, o ancora, di linee di ritardo. Tutte queste possibilità sono rafforzate dalla dimensione della memoria Flash di 256Kb e dalla notevole velocità di trasferimento dati della memoria stessa.

### INGRESSI DIGITALI 5V-TOLERANT

Il panorama attuale delle schede digitali presenta la maggioranza dei microcontrollori con i livelli di I/O a 5V e la maggioranza delle schede microcomputer



## DOPPIO CONVERTITORE ANALOGICO DIGITALE

bit del convertitore A/D ne permette l'impiego nel trattamento di segnali che richiedono una notevole precisione.

### DAC A 12 BIT

## TOUCH SENSE

## BUS CAN

## PIN DI RESET

## REAL TIME CLOCK E BATTERIA TAMPONE

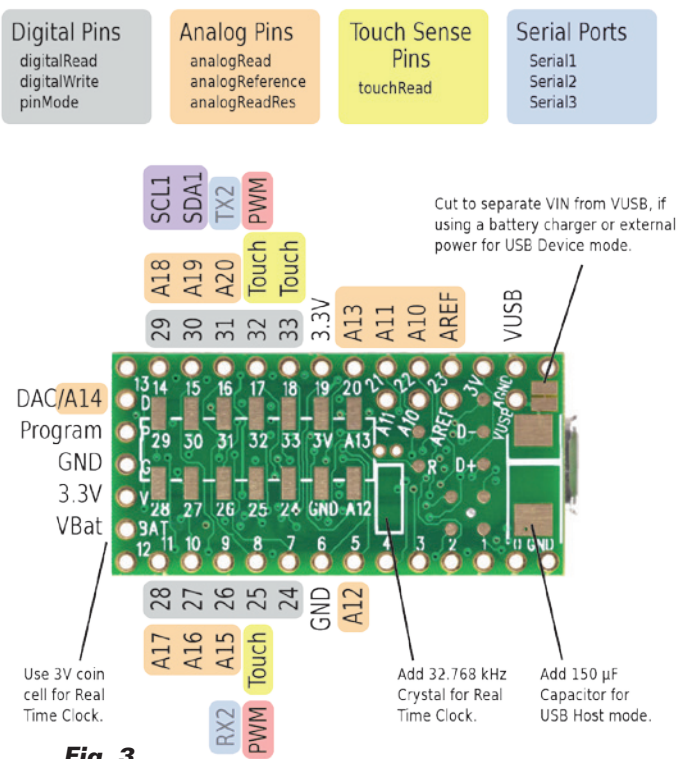
Un'altra funzionalità utile in un microcontrollore è il Real Time Clock, che normalmente richiede uno shield aggiuntivo. Sul Teensy 3.1 la funzionalità è già presente, per attivarla devono essere aggiunti un quarzo a 32.768 kHz ed una batteria tampone a 3V.

## PINOUT DEL TEENSY 3.1

I pin del Teensy 3.1 sono veramente tanti e per alloggiarli tutti in così poco spazio, il circuito stampato, nella sua parte inferiore, è letteralmente ricoperto di piazzole saldabili. Nelle **Fig. 2** e **Fig. 3** possiamo vedere le linee presenti sui connettori a passo 2,54mm e su quelli localizzati nella parte inferiore del circuito stampato. Le immagini nelle figure citate vengono fornite con ciascun modulo e riassumono in una forma grafica autoesplicativa le funzioni associabili a ciascun pin o piazzola.

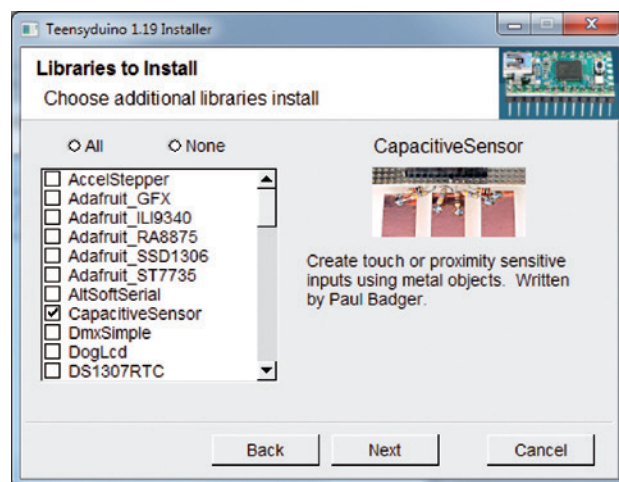
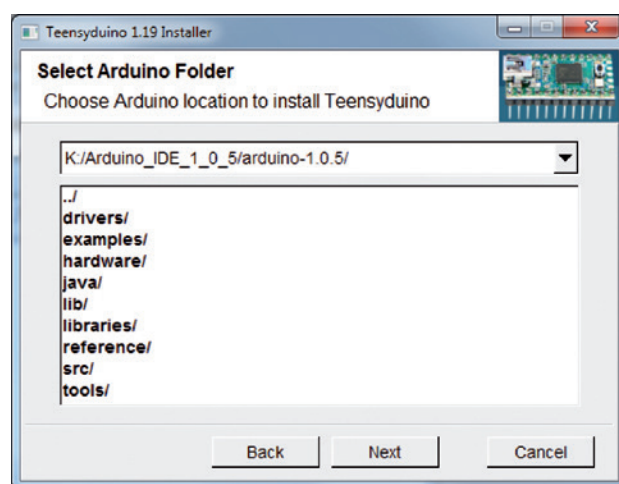
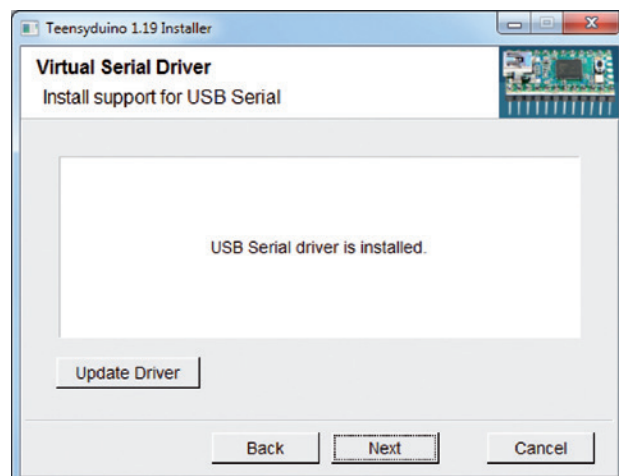
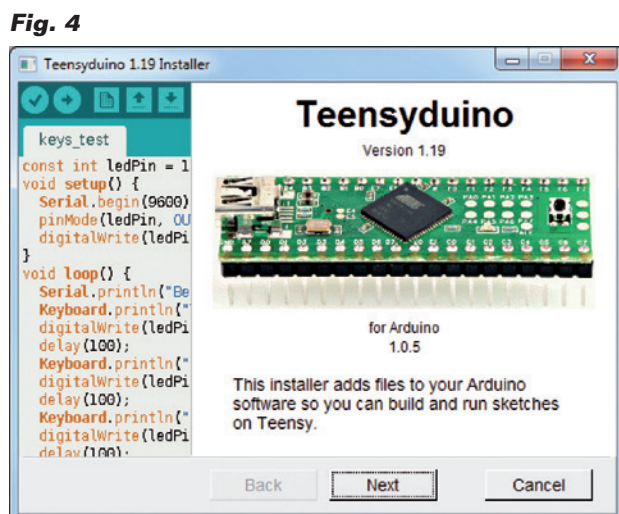






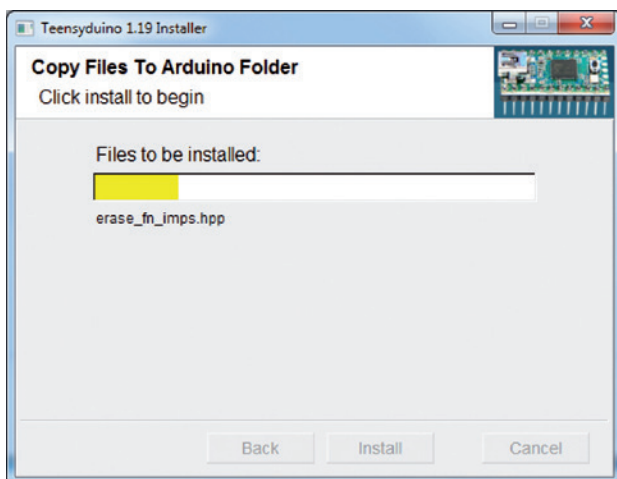
**SUBITO PRONTO ALL'USO**

Teensy 3.1 esce dalla fabbrica con due programmi preinstallati: il bootloader ed un programma di test che fa lampeggiare ciclicamente il LED presente sulla scheda, collegato al pin 13. Il bootloader permette di caricare i programmi utilizzando una versione modificata dell'IDE di Arduino. La modifica consiste nell'aggiungere all'IDE di Arduino il modulo add-on TeensyDuino, che permette di riconoscere e gestire la scheda Teensy 3.1, sia in termini di scrittura, compilazione e download dei programmi che di debug via console seriale.

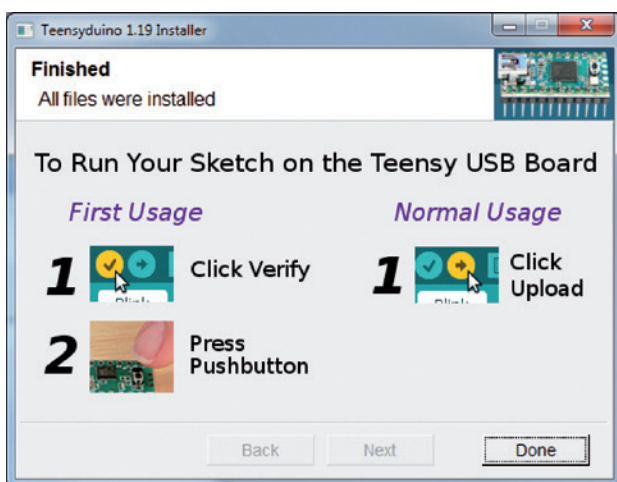


## TEENSYDUINO

TeensyDuino supporta unicamente la versione dell'IDE di Arduino 1.0.5; le versioni successive non sono supportate, ma ciò non costituisce un problema insormontabile: infatti è possibile installare sul proprio PC diverse versioni dell'IDE in contempo-



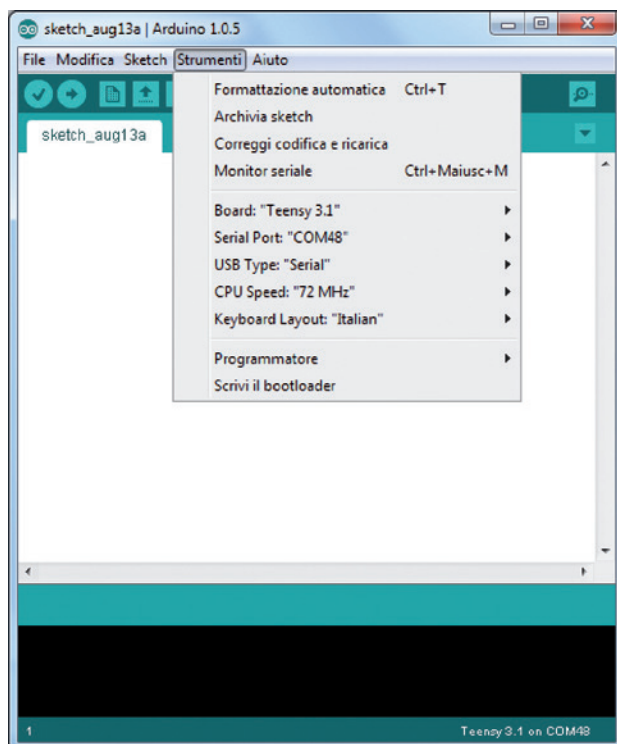
**Fig. 8**



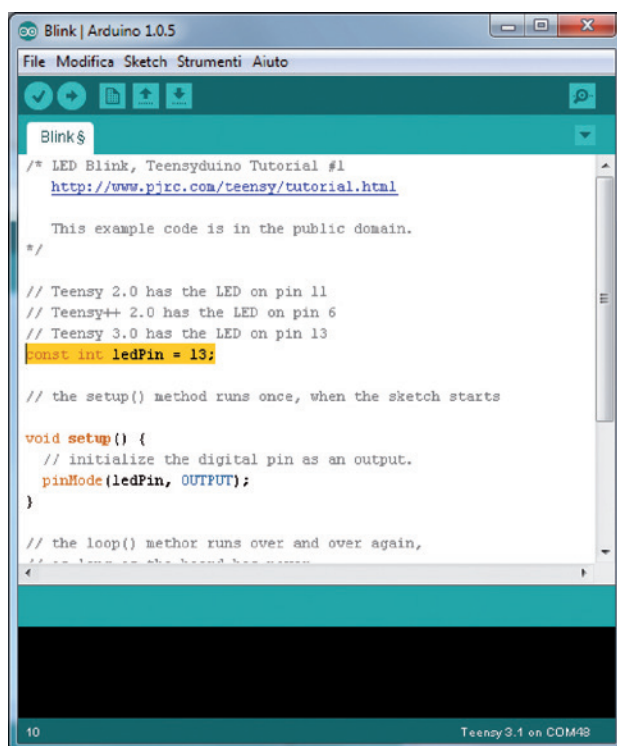
**Fig. 9**

reana, per poi scegliere quale usare. Dunque, se la versione 1.0.5 non è quella installata sul nostro PC la scarichiamo dall'indirizzo <http://www.arduino.cc/en/Main/Software> e la installiamo come di consueto; poi scarichiamo l'add-on TeensyDuino dall'indirizzo [https://www.pjrc.com/teensy/td\\_119/teensyduino.exe](https://www.pjrc.com/teensy/td_119/teensyduino.exe). Una volta terminato il download eseguiamo il programma "teensyduino.exe", che installa tutti i file necessari a gestire la scheda Teensy 3.1 con l'IDE di Arduino (Fig. 4). Quello che descriviamo è il processo che permette di installare l'add-on in ambiente Windows.

Sul sito PJRC sono disponibili le istruzioni per installare l'add-on in ambiente GNU/Linux e MAC OSX. Lo stesso programma permette di installare il driver per la porta seriale virtuale che permette di riconoscere la scheda ed assegnargli una porta seriale dopo averla collegata ad una presa USB del nostro PC. Quando viene presentata la finestra che richiede l'installazione del driver selezionate "Next". Nella finestra successiva impostate la cartella dove è contenuto l'IDE di Arduino. Nella cartella di destinazione



**Fig. 10**



**Fig. 11**

vengono installate diverse librerie che possono essere utilizzate con Teensy 3.1. Possono essere installate ulteriori librerie opzionali spuntandone i nomi nella finestra di selezione visibile in Fig. 7; man mano che selezioniamo una libreria, viene fornita una breve descrizione delle funzionalità della stessa in modo



Fig. 12

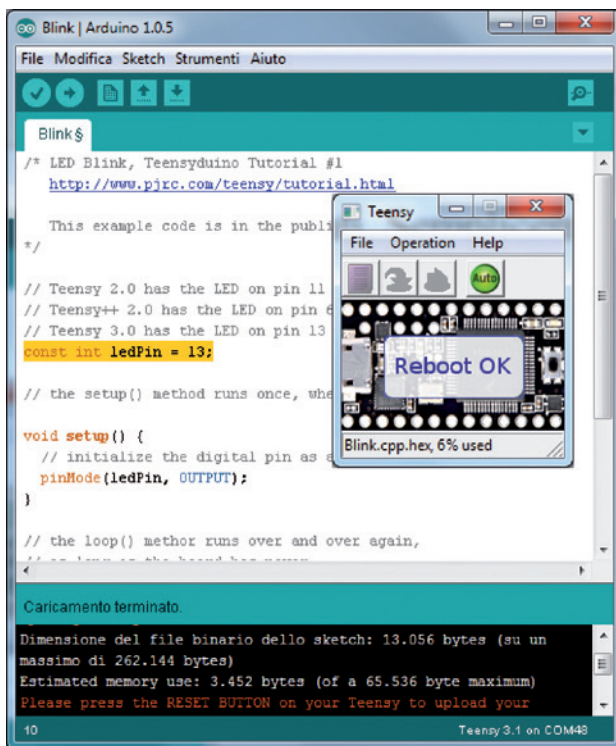


Fig. 13

da facilitare la scelta. Cliccando sul pulsante "Next" tutte le librerie selezionate vengono copiate nelle apposite cartelle dell'IDE di Arduino. Al termine dell'installazione ci verrà presentata la finestra finale. Per il momento clicchiamo sul pulsante "Done" per uscire dal programma di installazione.



Fig. 14

### COLLEGARE TEENSY 3.1 AL PC?

Nulla di più semplice: basta procurarsi un cavo USB-A da un lato e microUSB dall'altro, quindi inserire la spina microUSB nel connettore presente su Teensy 3.1 e il connettore USB-A in una presa USB del nostro PC. Il microcontrollore dovrebbe essere riconosciuto automaticamente come un dispositivo FTDI. In caso si verifichi un errore e il device non venga riconosciuto dal sistema operativo, possiamo forzare il riconoscimento del driver corrispondente nella finestra che viene aperta in caso di errore e farlo puntare alla cartella dei driver all'interno della cartella dove abbiamo scompattato l'IDE di Arduino. Appena collegato il Teensy 3.1 al PC dovremmo vedere lampeggiare il LED, collegato al pin 13: ciò significa che il programma di test precaricato funziona correttamente.

Da questo momento possiamo iniziare a programmare il microcontrollore utilizzando l'IDE di Ardui-

### Listato 1

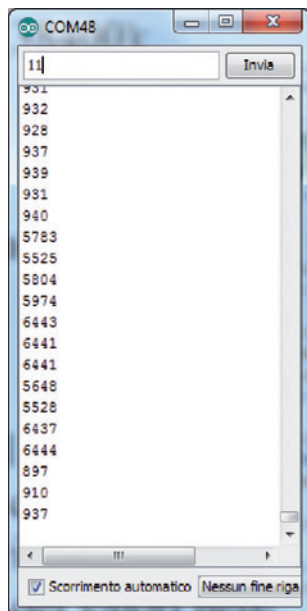
```
// Generatore onda sinusoidale per Teensy 3.1

float phase = 0.0;
float twopi = 3.14159 * 2;
elapsedMicros usec = 0;

void setup() {
  analogWriteResolution(12);
}

void loop() {
  float val = sin(phase) * 1000.0 + 1050.0;
  analogWrite(A14, (int)val);
  phase = phase + 0.02;
  if (phase >= twopi) phase = 0;
  while (usec < 500); // wait
  usec = usec - 500;
}
```



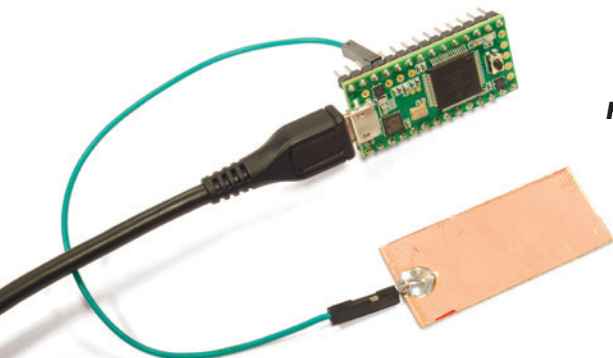
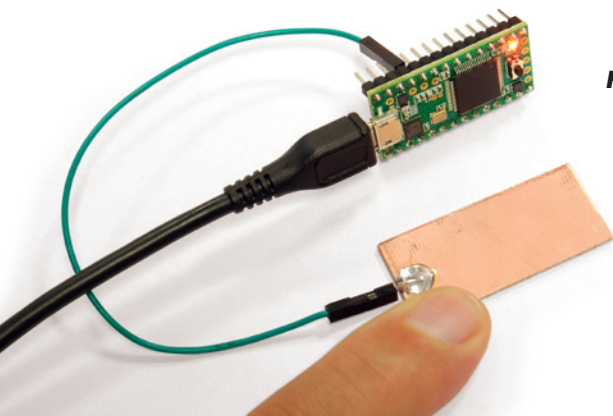
**Fig. 15**

no. Appena aperto l'IDE lo dobbiamo configurare per riconoscere la board Teensy 3.1: dal menu *Strumenti*, impartiamo il comando *Board* e selezioniamo "Teensy 3.1". In *Serial Port* selezioniamo la porta COM assegnata alla board. Infine, in *USB Type* selezioniamo "Serial" e poi impostiamo la "CPU Speed" a 72 MHz. Ora possiamo preparare uno sketch, compilarlo e caricarlo sul Teensy 3.1. Per fare una prima prova, carichiamo il programma "blink.pde": lo raggiungiamo aprendo il menu *File* e impartendo il comando *Apri*, quindi sfogliando la cartella degli esempi, dove troviamo una cartella "Teensy" inserita durante l'installazione di TeensyDuino. All'interno della cartella "Tutorial1" troviamo la cartella "blink" con lo sketch che cerchiamo. Lo apriamo, modifichiamo l'istruzione:

```
const int ledPin = 11;
```

con

```
const int ledPin = 13;
```

**Fig. 16****Fig. 17**

## Listato 2

```
// Riconoscimento TouchPad per Teensy 3.1

const int ledPin = 13;
int Valore = 0;

void setup() {
  Serial.begin(38400);
  pinMode(ledPin, OUTPUT);
}

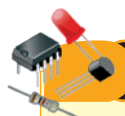
void loop() {
  Valore = touchRead(A9);
  Serial.println(Valore);

  if (Valore > 1100) {
    digitalWrite(ledPin, HIGH);
  }
  else {
    digitalWrite(ledPin, LOW);
  }
  delay(100);
}
```

e lanciamo l'upload dello sketch, che include anche la fase di compilazione. Se la compilazione va a buon fine vedremo aprirsi l'applicazione "Loader"; questa è stata anch'essa caricata durante l'installazione di TeensyDuino ed è l'applicazione finalizzata a caricare i programmi sul microcontrollore che ha a bordo il bootloader HalfKay.

La finestra di dialogo dell'applicazione "Loader" ci chiede di premere il pulsante presente sullo stampato del microcontrollore: la pressione del pulsante per un paio di secondi porta il microcontrollore in modalità bootloader, in modo da fargli accettare lo sketch appena compilato. Al termine del download viene eseguito il reboot del microcontrollore (**Fig. 13**) ed in pochissimo tempo dovremmo vedere il LED tornare a lampeggiare. Una chicca di Teensy 3.1 è il fatto che all'indirizzo [https://www.pjrc.com/teensy/loader\\_cli.html](https://www.pjrc.com/teensy/loader_cli.html) è disponibile una versione dell'applicazione Loader utilizzabile da linea di comando che funziona con i sistemi operativi Mac, Windows, GNU/Linux e BSD Unix.

Proviamo ad approfondire alcune delle funzionalità presenti in Teensy 3.1: la conversione digitale analo-



## per il MATERIALE

La scheda Arm Cortex M4 32 bit (cod. TEENSY3.1) di cui si parla nell'articolo è disponibile al prezzo di Euro 26,90. Il prezzo si intende IVA compresa.

Il materiale va richiesto a: Futura Elettronica,  
Via Adige 11, 21013 Gallarate (VA)  
Tel: 0331-799775 • Fax: 0331-792287 - [www.futurashop.it](http://www.futurashop.it)

gica e i sensori capacitivi. Per testare le funzionalità di conversione digitale/analogica utilizziamo lo sketch visibile nel **Listato 1**, che permette di generare un segnale sinusoidale. Il listato contiene un semplicissimo sketch alla "Arduino" che non richiede commenti. Semplicemente impostiamo la risoluzione del convertitore con l'istruzione:

```
analogWriteResolution(12);
```

prepariamo il valore digitale da trasformare in segnale analogico e lo veicoliamo al convertitore DAC sul pin A14, mediante l'istruzione "analogWrite()":

```
analogWrite(A14, (int)val);
```

Vediamo il risultato sul monitor dell'oscilloscopio di RedPitaya (**Fig. 14**).

Nel **Listato 2** vediamo invece uno sketch finalizzato a "leggere" i valori capacitivi che si presentano sui pin "Touch". Per eseguire il test colleghiamo al pin A9 un "Touch Pad" realizzato con un ritaglio di basetta per

circuiti stampati collegata al pin citato con un cavetto. Una volta compilato e caricato lo sketch, proviamo a toccare la basetta collegata al pin A9 con un dito e verificiamo il cambiamento dei valori nella finestra del "Monitor Seriale" (**Fig. 15**).

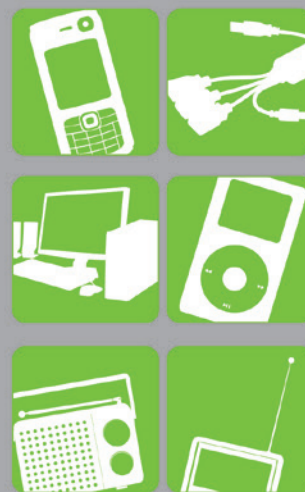
Quando avviciniamo il dito alla basetta stampata si accende il LED sulla basetta collegato al pin 13 (**Fig. 16 e Fig. 17**). Notate che, trattandosi di un ingresso capacitivo, la cosa funziona anche se inseriamo un dielettrico tra il pad ed il dito, come un foglio di carta o di plastica. Possiamo utilizzare questa funzionalità per realizzare un sistema di input per un'applicazione da operare attraverso un pannello trasparente, oppure un originale strumento musicale o di gestione luci. Restiamo in attesa dello sviluppo delle librerie da parte di PJRC per sfruttare appieno i due convertitori ADC (in contemporanea) ed il comparatore. Per ora lo impiegheremo come front end di acquisizione ed elaborazione dati di misura da interfacciare con i sistemi GNU/Linux embedded, e perché no, anche come sistema di controllo del funzionamento dei sistemi stessi. Vi terremo informati degli sviluppi. ■

# Verona 22 - 23 novembre 2014

## La fiera dell'elettronica



Elettronica  
Radiantismo  
Strumentazione  
Componentistica  
Informatica  
Videogiochi  
Giocattoli Elettronici  
Mercatino del privato



# www.elettroexpo.it

organized by  
 VERONAFIERE





# FIRST LEGO LEAGUE ITALIA 2014-2015



**Sono iniziate le iscrizioni alla competizione tra robot realizzati da ragazzi, che selezionerà i partecipanti al Campionato mondiale 2014-2015, che quest'anno prende il nome di World Class.**

a cura della REDAZIONE



**L**a competizione mondiale organizzata da FIRST (For Inspiration and Recognition of Science and Technology, ovvero Per l'ispirazione e il riconoscimento di Scienza e Tecnologia) e la multinazionale LEGO, torna in scena e chiama a sè giovani ragazzi che progettano, costruiscono e programmano robot autonomi. FLL aiuta i giovani a scoprire il divertimento nella scienza e nella tecnologia, costruendo nel contempo autostima, conoscenza, competenze. FLL si disputa in tutto il mondo e prevede manifestazioni nazionali, continentali e mondiali; per l'Italia, l'istituto referente è la Fondazione Museo

Civico di Rovereto. Attualmente FLL coinvolge 70 nazioni distribuite sui 5 continenti; le qualificazioni partono dalla fase regionale per proseguire in quella nazionale, che qualifica 4 team per eventi internazionali (World Festival, Open European Championship, Open Canadian Championship).

L'ultimo campionato italiano ha coinvolto 76 team per un totale di 681 ragazzi; alla finale nazionale svoltasi al Palasport Marchetti di Rovereto hanno assistito 2.000 persone e in campo si sono sfidati oltre 200 ragazzi di 26 team provenienti da tutta Italia.

FIRST LEGO League è un concor-

so mondiale di scienza e robotica tra squadre di ragazzi dai 9 ai 16 anni (dalla quarta elementare alla seconda superiore, non obbligatoriamente della stessa classe o istituto) che progettano, costruiscono e programmano robot autonomi, applicandoli a problemi reali di grande interesse generale, ecologico, economico, sociale, per cercare soluzioni innovative. FLL non è solo robotica, perché come dice il nome scelto per l'edizione 2014-15 -World Class- tratterà soluzioni per innovare e migliorare il sistema scolastico ed educativo in generale. Per mettere sullo stesso piano concorrenti di età differen-

ti, i promotori della competizione hanno pensato di proporre a tutti i partecipanti l'uso esclusivo, per la realizzazione dei robot, dei materiali LEGO Mindstorms.

### Come funziona la FLL

FLL coniuga una fase di ricerca ed esposizione ad una gara vera e propria di robotica. L'evento si divide in quattro prove distinte, ognuna delle quali ha uguale peso sul risultato finale.

La prima prova è *Gara di robotica*, la più spettacolare perché si svolge all'interno di un palazzetto o comunque di uno spazio adibito all'accoglienza del grande pubblico. In quest'area è possibile esporre banner degli sponsor e avere a disposizione uno stand espositivo. Onde permettere al grande pubblico di seguirla, vengono effettuate riprese in diretta delle fasi della gara trasmesse in tempo reale sia su un maxischermo che in streaming via internet. Le squadre dovranno realizzare e programmare, con materiale esclusivamente LEGO, un robot autonomo che realizzi il maggior numero delle missioni presenti sul tavolo di gara in 2,5 minuti. Il percorso di gara con le missioni viene fornito ai concorrenti almeno 2 mesi prima all'atto dell'iscrizione in modo che questi possano testare le soluzioni adottate con largo anticipo e scegliere

la strategia di gara adeguata. Il percorso rimane sempre uguale dalla fase regionale alla finale mondiale.

La seconda è *Progetto Tecnico*-Documentazione e presentazione delle soluzioni robotiche adottate a una giuria di esperti: si svolge in una saletta accessibile a un pubblico limitato, il team presenta con la modalità che preferisce le soluzioni robotiche adottate per risolvere alcune missioni, rispondendo ai quesiti dei giurati ed effettuando dimostrazioni del funzionamento del robot.

La terza è *Progetto Scientifico*-Realizzazione e presentazione di una ricerca scientifica sulla tematica assegnata: si svolge in una sala accessibile a un pubblico limitato, dove i concorrenti sono liberi di utilizzare tutti gli strumenti che preferiscono per effettuare una ricerca da documentare e presentare durante la gara a una giuria di esperti. Il tema quest'anno riguarda l'innovazione nell'istruzione dei futuri cittadini globali. I casi valutati dalla giuria "di alto valore" verranno proposti alla FIRST per concorrere al FLL Global Innovation Award, che darà, oltre a un riconoscimento in denaro, risonanza a livello accademico e istituzionale, ma anche la possibilità di veder brevettata la propria idea.

L'ultima prova è *Core Values*-

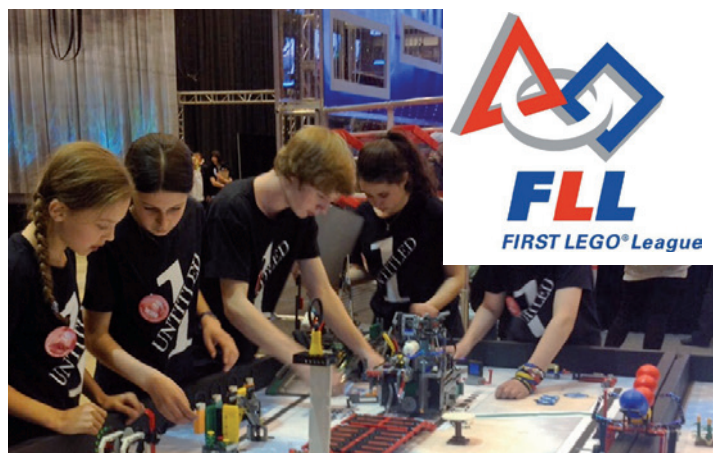
Valutazione del team e delle sue dinamiche: alcuni esperti nella dinamica di gruppo valutano i comportamenti del team nella soluzione di problemi e li intervistano per comprendere la gestione dei rapporti interni, con gli educatori e con i concorrenti. I giudizi acquisiti in questa fase vengono integrati dall'esame dei comportamenti sul campo di gara e negli stand.

### FLL Italia 2014-2015

Le iscrizioni a FLL 2014-2015 sono già aperte dal 15 settembre e chiuderanno il 15 dicembre 2014. Le squadre iscritte parteciperanno alla fase regionale più vicina alla loro collocazione geografica. Le qualificazioni sono organizzate secondo il seguente calendario:

- Nord-Ovest: 24 Gennaio 2015 - Cuneo (Scuola di Robotica);
- Nord-Est: 14 Febbraio 2015 - Bolzano (Fondazione MCR);
- Sud: 18 Febbraio 2015 - Catania (I.T.I.S. "Archimede");
- Centro: 21 Febbraio 2015 - Pistoia (I.T.T.S. "Fedi Fermi").

La finale Italiana è in programma nei giorni 20/21 Marzo 2015 a Milano, presso le strutture dell'EXPO 2015. La finale Mondiale si terrà invece dal 23 al 26 Aprile 2015 a St. Louis, negli Stati Uniti. Tutte le informazioni sono sul sito della competizione, <http://fll-italia.it>.





# Date un passaggio al **ROBOT....**

...è questo il messaggio lanciato dal progetto Hitchbot, il viaggio di un robot umanoide (che ha già compiuto oltre 6.000 chilometri) nato dall'idea di alcuni ricercatori canadesi con l'intenzione di realizzare sostanzialmente un esperimento sociale, basato sulla fiducia che gli umani possono nutrire nei

confronti dei robot. Questo esperimento vede protagonista un robot, chiamato HitchBot, alto quanto un bambino di cinque o sei anni, alimentato da celle solari che ne rivestono la pancia, che integra un ricevitore GPS ed una fotocamera ed è connesso ad internet tramite la rete 3G; questo robot "avventuriero" fa l'autostop da una costa all'altra del Canada, chiedendo passaggi e permettendo di valutare quanto l'uomo sarebbe propenso a dargliene uno, fidandosi di ospitarlo a bordo.

Hitchbot è stato costruito da un gruppo di ricercatori di Port Credit, in Ontario, tra i quali segnaliamo il dottor David Smith, della McMaster University e il dottor Frauke Zeller, della Ryerson University.

Il piccolo robot è partito il 27 luglio scorso da Halifax, per poi attraversare sempre e solo rigorosamente in autostop il Canada coast to coast, e la Nuova Scozia fino a Victoria, sulle coste dell'Oceano Pacifico, nella British Columbia.

"Sono un robot-spirito-libero che vuole esplorare il Canada e incontrare nuovi amici lungo la strada", ha dichiarato Hitchbot sul suo sito web. Ed è quello che ripete la sua voce elettronica quando qualche automobilista che lo trova sul ciglio della strada accosta per guardarlo meglio; chi, colto dalla curiosità e abbastanza intraprendente da voler scoprire di che si tratta, può prenderlo con sé e fare con esso un pezzo di strada, per poi farlo scendere dove ritie-



## IL ROBOT ASPIRAPOLVERE DEL FUTURO

Una gestazione lunga sedici anni, 36 milioni di euro di investimenti per svilupparlo e 420 brevetti; tanto è servito a James Dyson per realizzare la sua creatura più innovativa: il robot Dyson 360 Eye.

Questo aspirapolvere del futuro è dotato di una videocamera panoramica interna capace di catturare 30 immagini al secondo e di fornirgli una visione completa a 360 gradi, fondamentale per vedere le più piccole particelle di polvere (anche 600 volte più di quelle localizzabili dai diretti concorrenti sul mercato). Grazie ad un sistema a infrarossi, che può essere attivato anche da smartphone, il robot riesce a mappare l'ambiente calibrando l'azione di pulizia che gli consente di evitare

di scontrarsi con i mobili di casa. Dyson 360 Eye ha uno spessore più alto della media (circa 12 centimetri), ha un motore ad alta velocità con aspirazione ciclonica e un'autonomia di circa trenta minuti. Quando sta per scaricarsi, torna alla base, dove si ricarica in tempi molto brevi.

Dyson 360 Eye non ha ruote ma solo cingoli e dunque non necessita dell'intervento umano, ad esempio per sbrogliare eventuali accumuli di polvere o nel caso dovesse impantanarsi in un tappeto o nella moquette.

Durante la presentazione alla stampa, il robot ha ripulito un piano stracolmo di farina, messo a confronto con uno dei rivali di

maggior successo sul mercato e si è distinto per efficacia.

L'aspirapolvere robot di Dyson sarà sul mercato in primavera in Giappone e pochi mesi dopo anche in Inghilterra ad un costo di 750 sterline (circa mille euro).

<https://www.dyson360eye.com/>







ne. Durante il viaggio, Hitchbot comunicherà via web la propria posizione per aggiornare i dati sul viaggio...e magari evitare che qualcuno ci si affezioni tanto da tenerlo come souvenir di un viaggio nel Nordamerica...

Il robot è in grado di interagire con le persone, in quanto è dotato di un software di riconoscimento vocale, grazie al quale può amabilmente chiacchierare con i suoi accompagnatori, fornendo risposte, potendo accedere al web e ad esempio a Wikipedia.

<http://www.hitchbot.me/>



## CIBER-FORMICHE SU LUNA E MARTE

Un team di ingegneri della NASA sta lavorando a robot grandi come tavolini su ruote che potrebbero essere inviati nello Spazio alla ricerca di risorse preziose, come fossero un'operosa colonia di formiche. Basandosi sulla ricerca condotta dall'Università del New Mexico, gli ingegneri del Kennedy Space Center della NASA hanno sviluppato dei

software che guidano i piccoli robot a individuare in maniera casuale in una determinata area alcuni materiali ritenuti interessanti dai ricercatori. Lo scopo è mandare i robot su un asteroide o sulla Luna o su Marte, per scansionare le loro superfici alla ricerca di acqua o ghiaccio, oppure di risorse che potrebbero essere trasformate in combustibili per i razzi spaziali. Per ora i test sono stati realizzati con quattro prototipi denominati "swarmies", con l'intento di organizzare squadre di robot che si muovono in sincrono come sciami d'api, o come battaglioni di formiche. Ognuno di essi è dotato di una webcam, di un modulo WiFi e di un ricevitore GPS. Sono programmati per lavorare in autonomia e sorvegliare un'area: non appena trovano un oggetto interessante richiamano gli altri robot e cooperano per prelevare un po' come fanno le formiche quando trovano qualcosa di utile da condividere con le altre.

[www.nasa.gov](http://www.nasa.gov)







# Conoscere e usare



# kicad

Concludiamo questo corso terminando l'analisi di Pcbnew e, contestualmente, lo sviluppo del progetto pratico. Sesta ed ultima puntata.

di Francesco Ficili e Vincenzo Germano



**N**ella puntata precedente abbiamo continuato l'analisi dello strumento Pcbnew, che avevamo iniziato ad affrontare già nella quarta puntata, quando ci siamo occupati dell'editor dei layout, dei riempimenti e della gestione di librerie e zone; sempre nella scorsa puntata, siamo andati avanti nella realizzazione del progetto pratico, completando la fase di posizionamento dei componenti. In queste pagine si conclude il corso dedicato a KiCad: completeremo la nostra panoramica sullo strumento Pcbnew e termineremo la realizzazione del nostro progetto pratico. Infine, vedremo come realizzare i file gerber necessari alla realizzazione dei nostri circuiti stampati.

## DEMOBOARD PIC18: IL ROUTING

Non ci soffermiamo in dettaglio sul routing del progetto pratico, in quanto l'argomento è già stato trattato nelle puntate precedenti ed una trattazione dettagliata del routing di una scheda di queste dimensioni risulterebbe eccessivamente pesante. Ci limiteremo, quindi, ad evidenziare qui di seguito, alcune linee guida valide per tutti i casi.

- Prima di eseguire lo sbroglio della scheda, bisogna posizionare tutti i moduli nella maniera più conveniente.
- Impostare le "design rules" di progetto (*Preferenze/Preferenze Circuito Stampato*), tenendo presenti:
  - dimensioni delle piste;
  - isolamento delle piste;



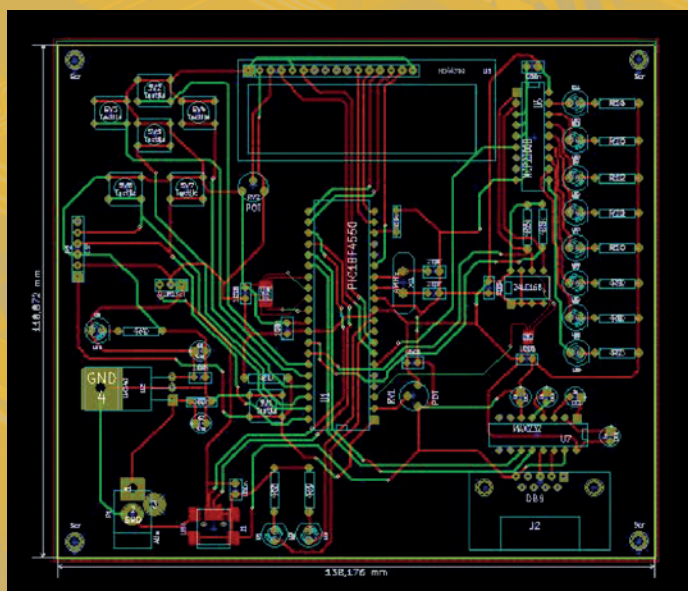


FIG. 1 - Layout della demoboard PIC18F4550.

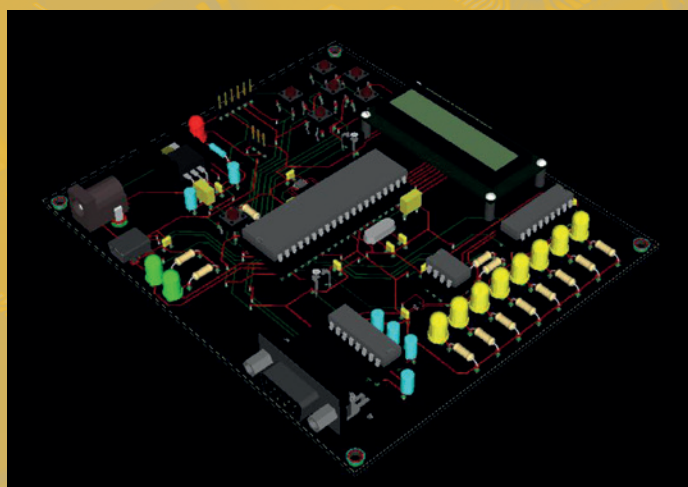


FIG. 2 - Rappresentazione 3D della demoboard PIC18F4550.

- fori di via;
- anular ring dei fori di via.
- Durante lo sbroglio evitare di realizzare piste con angoli retti (in alta tensione causano l'innescio di scariche).
- Evitare percorsi troppo lunghi e loop.
- Prevedere sempre almeno un piano di massa per il PCB.
- Prestare attenzione alla posizione delle serigrafie, evitando, per quanto possibile, che queste si sovrappongano alle piazzole e alle piste ed alle vie.

In Fig. 1 è visibile la scheda del progetto pratico a routing ultimato, mentre in Fig. 2 è visibile la rappresentazione 3D creata dal 3D-viewer di KiCad. Per la creazione del render 3D, sono state utilizzate le librerie 3D realizzate da Walter Lain, scaricabili dal suo sito web <http://smisioto.no-ip.org/elettronica/kicad/kicad.htm>.

Da entrambe le immagini (Fig. 1 e Fig. 2) è stato rimosso il piano di massa, per permettere di vedere in maniera più chiara il routing. Per visualizzare il rendering 3D del PCB è sufficiente selezionare, dal menu *Visualizza*, l'opzione "Visualizzazione 3d": usando i modelli 3D associati ai footprint, Pcbnew genererà il render 3D della scheda.

### VERIFICA PCB: CONTROLLO DELLE REGOLE DI PROGETTO

Una volta completato il routing, Pcbnew offre la possibilità di eseguire un check automatico sul progetto, il quale verifica:

- il rispetto delle regole imposte in fase di progetto;
- la presenza di nodi non connessi.

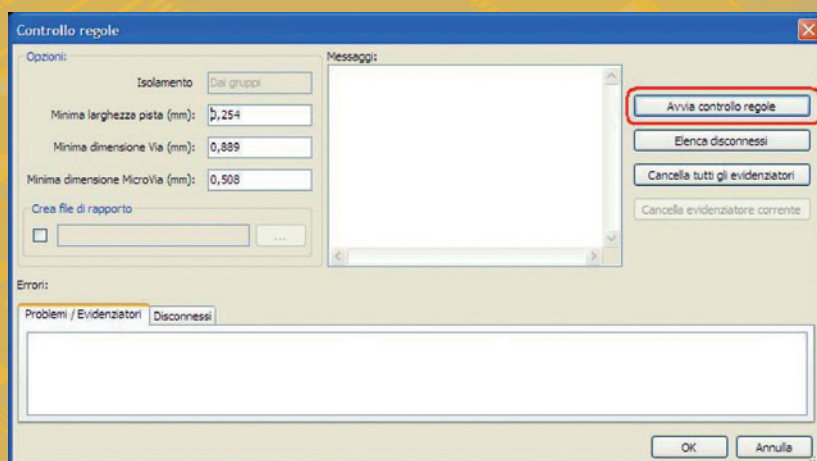


FIG. 3 - Finestra di dialogo "Controllo regole".

Pcbnew offre altresì la possibilità di rieseguire automaticamente il riempimento delle zone. Per avviare il controllo delle regole bisogna selezionare l'icona con il simbolo della coccinella e la spunta verde, che si trova in alto sulla Project bar, a fianco dell'icona per il caricamento della netlist, oppure bisogna impartire il comando omonimo che si trova sotto il menu *Strumenti* (*Controllo Regole*). Una volta eseguita questa



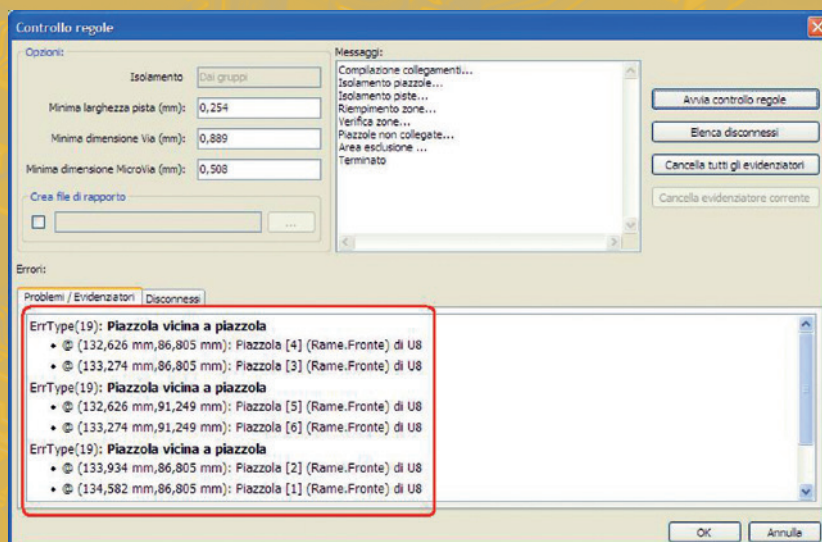


FIG. 4 - Esempio di errore sulla distanza di guardia.

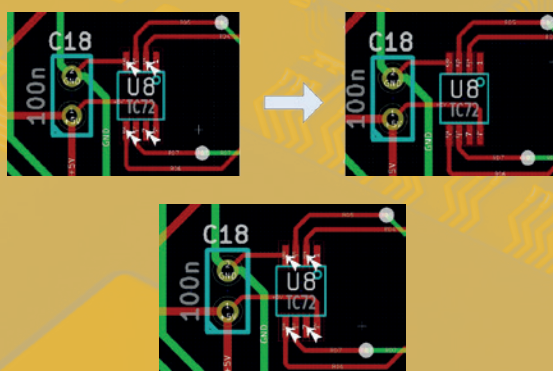


FIG. 5 - Errore sulla distanza di guardia (sopra) e relativa correzione (sotto).

operazione, si aprirà la finestra illustrata nella Fig. 3. Il controllo delle regole di progetto si avvia facendo clic sul pulsante *Avvia Controllo Regole*, come evidenziato sempre in Fig. 3.

Le opzioni per il controllo sono riportate nel riquadro a sinistra in alto (è anche possibile generare e salvare un file contenente il rapporto); nello specifico, vengono controllati:

- gli isolamenti (dalle preferenze impostate in fase di progettazione e dalle proprietà dei moduli);
- le larghezze delle piste;
- la dimensione delle vie (vias);
- la dimensione delle microvie (microvias).

In Fig. 4 è riportato un esempio di controllo che restituisce un errore sulla distanza di guardia tra le piazzole di un componente SMD. L'errore e la relativa correzione sono riportati in Fig. 5: come si può vedere nell'im-

agine sopra, i delimitatori delle distanze di guardia per le piazzole si compenetrano tra di loro, e quindi è stata violata la regola imposta in sede di progetto sulla distanza di guardia tra le piazzole.

Per risolvere il problema è sufficiente cambiare il valore dell'isolamento tra le piazzole, operando dalla finestra di pro-

prietà del modulo, portandolo ad un decimo di mm, come illustrato in Fig. 6.

È anche possibile verificare solo la presenza di nodi disconnessi, evitando di eseguire il check su tutte le regole: ciò si ottiene facendo clic sul pulsante corrispondente.

## GENERAZIONE DEI FILE DI OUTPUT

Una volta sicuri che non siano state violate delle regole di progetto e che nel layout non siano presenti nodi disconnessi, è possibile

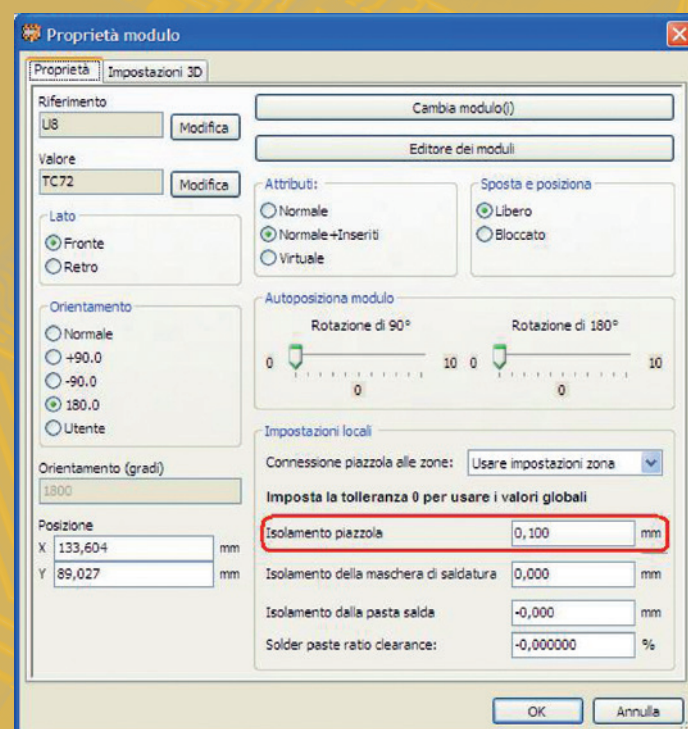


FIG. 6 - Modifica della proprietà del modulo.





## Tracciatore

FIG. 7 - Generatore file di output.

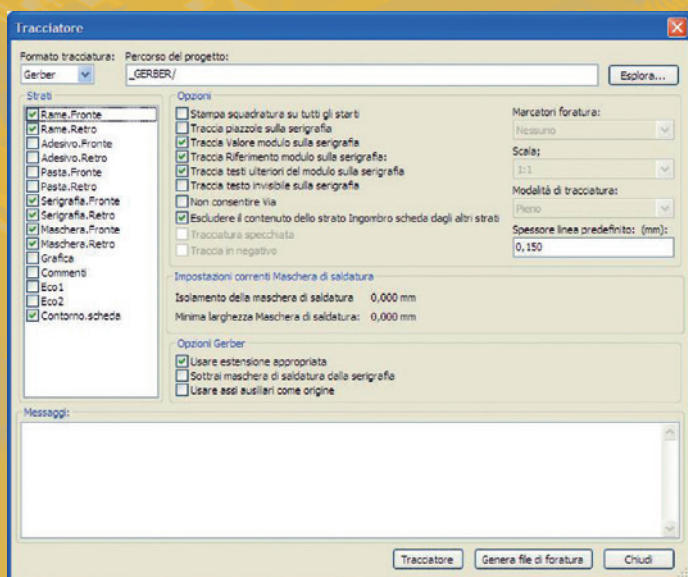


FIG. 8 - Finestra di dialogo Tracciatore.

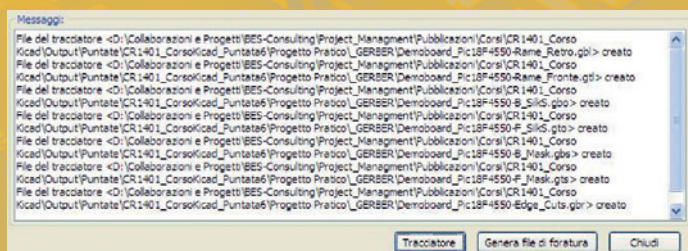


FIG. 9 - Generazione dei file di output.

passare alla fase di realizzazione dei file di fabbricazione. Per farlo, è sufficiente selezionare l'icona a forma di stampante con la lettera P sovrapposta, che è presente, sempre all'interno della Project bar, a sinistra dello zoom, come indicato in Fig. 7 (oppure selezionare l'opzione File/Tracciatore dal menu principale).

Una volta selezionato lo strumento, si aprirà la finestra di dialogo mostrata in Fig. 8. Vediamo quali sono i passi da compiere per generare correttamente i nostri file di output. Per prima cosa è necessario impostare il percorso all'interno del quale salvare i file generati; per farlo, è sufficiente cliccare sul pulsante "Esplora..." che farà comparire una finestra di dialogo di Windows attraverso la quale è possibile impostare il percorso preferito (viene anche fornita la possibilità di

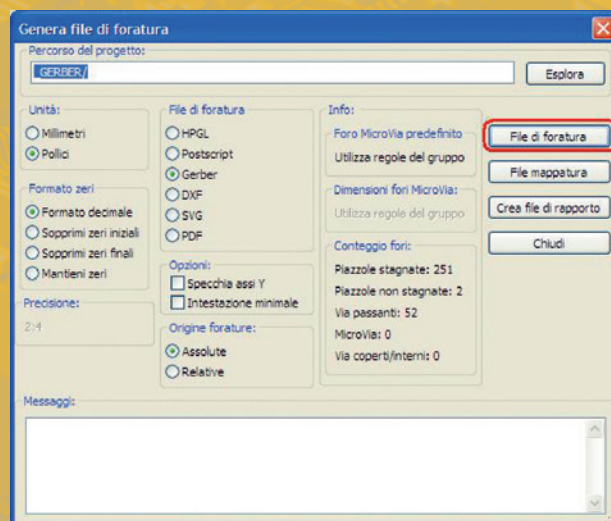


FIG. 13 - Finestra di generazione dei file di foratura.

selezionare un percorso relativo, in modo da non dover modificare questa impostazione qualora si portasse il progetto su un'altra macchina). Una volta impostato il percorso di salvataggio dei file, è necessario scegliere il formato dell'output; KiCad è in grado di generare file di output con i seguenti formati (selezionabili dal menu a tendina sotto la voce "Formato Tracciatura"):

- Gerber;
- PostScript;
- SVG;
- DXF;
- HPGL.

Una volta selezionato il formato preferito (in genere è Gerber), è possibile selezionare dall'elenco "strati", nella parte sinistra della finestra, i layer che si desidera tracciare (basta mettere o togliere il segno di spunta nella checkbox relativa ai layer). A questo punto si può intervenire sulle opzioni, presenti nei due riquadri "Opzioni" ed "Opzioni Gerber". Una volta completati tutti i settaggi è sufficiente fare clic sul pulsante "Tracciatore" per generare i file di output. Il tool ci fornirà un report relativo alla generazione dei file, come quello visibile in Fig. 9.

Oltre ai file relativi ai vari strati, è necessario generare anche i file di foratura. Questo tipo

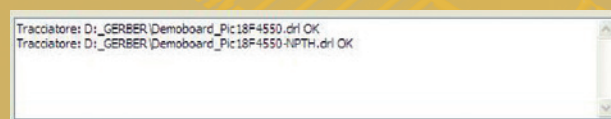


FIG. 11 - Generazione dei file di foratura.



di file si genera con un processo a parte, che può essere selezionato facendo clic sul pulsante "Genera File di Foratura", presente nel tool di generazione.

Anche in questo caso verrà aperta una finestra specifica (Fig. 10), dalla quale è possibile impostare una serie di opzioni per la generazione dei file, nonché il percorso di salvataggio degli stessi. Da tenere presente che i file di foratura generati devono essere coerenti con il formato scelto per i file relativi ai layer, quindi se generiamo dei file gerber anche il file di foratura dovrà avere questo tipo di estensione. Tra le varie opzioni disponibili, è possibile selezionare l'unità di misura (i pollici sono l'unità di misura comunemente accettata per questo tipo di lavorazioni). Una volta completati i settaggi, è sufficiente premere il tasto "File di Foratura" per avviare la generazione dei file (anche in questo caso, il risultato dell'operazione viene riepilogato nella finestra di dialogo, come rappresentato in Fig. 11).

## CREAZIONE DEI FILE DI POSIZIONAMENTO

Pcbnew permette anche di creare i file di posizionamento per il montaggio tramite pick&place, nel caso la scheda sia di una certa dimensione e sia stata progettata prevedendo la tecnologia a montaggio superficiale. Per accedere a questa opzione bisogna selezionare la voce "Esportazioni per fabbricazione/Module Position" presente sotto il menu *File*. Una volta selezionata questa voce, sarà visualizzata la finestra di Fig. 12. Da qui, come di consueto, è possibile scegliere il percorso all'interno del quale salvare i file generati dall'elaborazione, oltre ad altre opzioni, come ad esempio l'unità di misura, la possibilità di generare un file di posizionamento per ogni strato della scheda (in alternativa ad un file per scheda, nel caso di un bundle).

Una volta completate le impostazioni, è possibile generare i file di posizionamento semplicemente facendo clic sul pulsante "OK" (in Fig. 13 è riportato l'output della finestra di dialogo).

## CREAZIONE DELLA BOM

Una volta completata la realizzazione del nostro circuito stampato, avremo la necessità di approvvigionarci dei componenti necessari

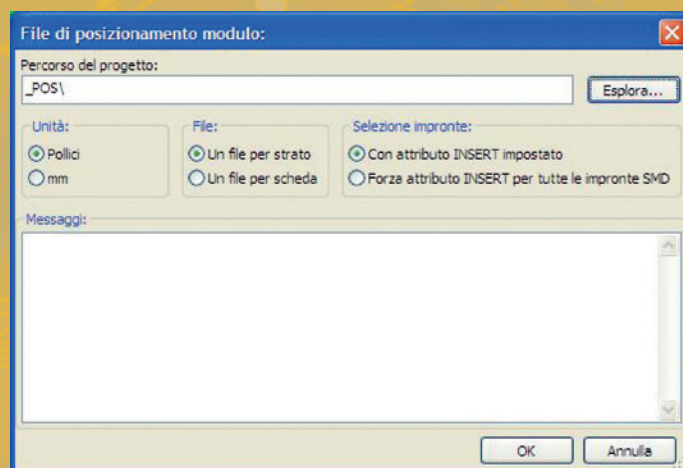


FIG. 12 - Finestra di generazione file di posizionamento.

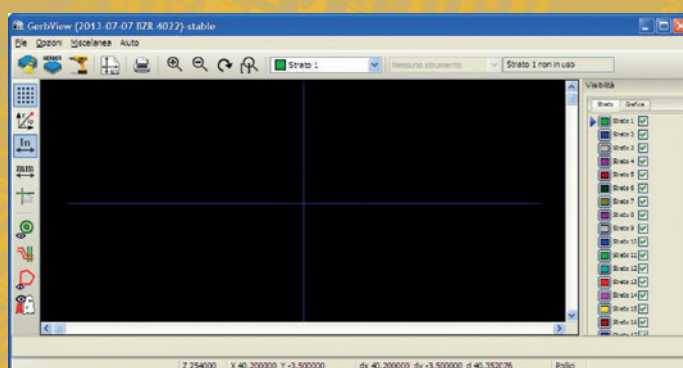


FIG. 13 - Generazione dei file di posizionamento.

alla sua realizzazione; KiCad ci viene incontro in questa fase, fornendoci, tra le esportazioni per la fabbricazione, la possibilità di generare la BOM (acronimo di Bill Of Material) del progetto. Generarla è semplicissimo: basta selezionare la voce "Esportazioni per Fabbri- cazione/Elenco Materiali" dal solito menu *File*, come illustrato in Fig. 14.

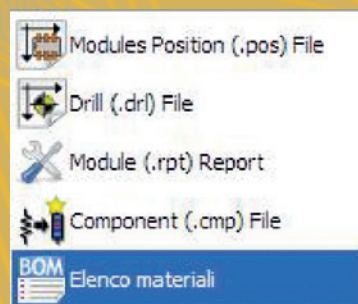


FIG. 14 - Generazione della BOM.

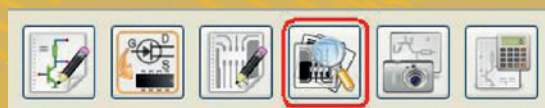


FIG. 15 - Icona di lancio di Gerbview.



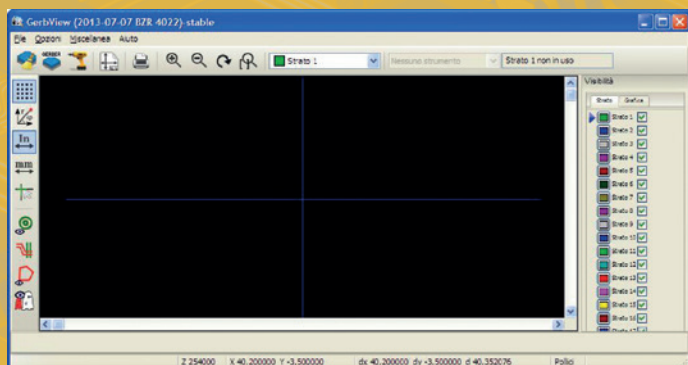


FIG. 16 - Finestra di dialogo Gerberview.

Dopo aver selezionato questa voce, verrà visualizzata una finestra di dialogo di Windows, che permette di selezionare il path all'interno del quale salvare la BOM. Il formato di esportazione è un file .csv contenente i riferimenti dei moduli, la quantità, il package ed il valore assegnato durante lo schematic design.

### ISPEZIONE DEI FILE GERBER: GERBVIEW

Analizziamo adesso l'ultimo tool della suite di KiCad (non considerando il calcolatore per

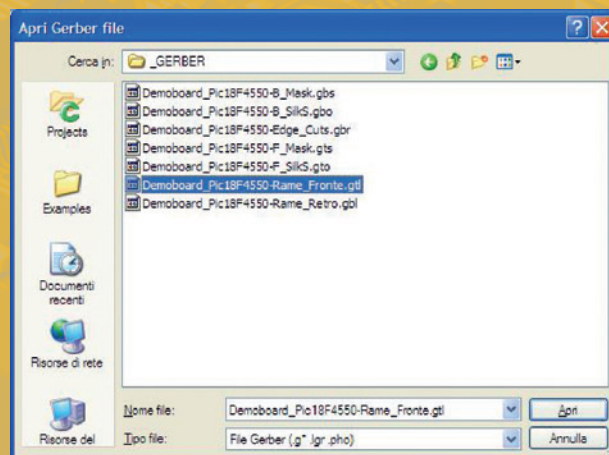


FIG. 17 - finestra di dialogo per il caricamento dei file in Gerberview.

PCB ed il convertitore Bitmap/Sikscreen come degli strumenti facenti parte della toolchain, ma più che altro come delle utility a corredo), ossia il visualizzatore di file gerber, Gerberview. Come dice anche il nome, si tratta di un semplice visualizzatore che ci permette di accertare che i file gerber generati siano effettivamente corretti ed evitare, di conseguenza, di mandare in lavorazione dei file difettosi,

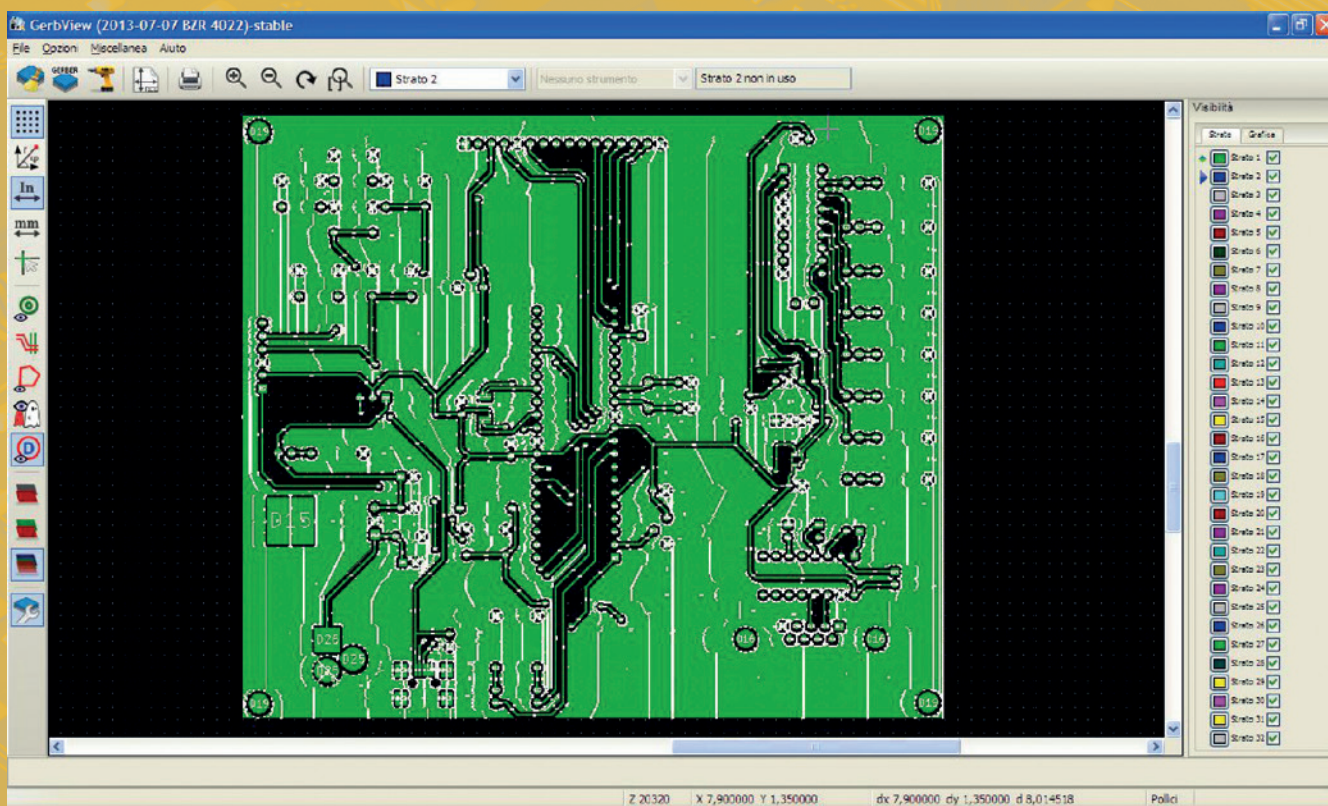


FIG. 18 - Visualizzazione di un file gerber tramite Gerberview.



che produrrebbero dei PCB errati, con evidente sperpero di tempo e denaro.

Per aprire Gerberview è sufficiente cliccare sulla relativa icona (la quarta da sinistra) nella toolbar di KiCad, come evidenziato in Fig. 15. Il tool si presenta come in Fig. 16; in alto è presente una toolbar minimale che permette di caricare file gerber e file di foratura, di eseguire zoom e di stampare. Sul lato sinistro è presente un'altra toolbar contenente alcune opzioni di visualizzazione, mentre a destra si trova la selezione degli strati da visualizzare. Per caricare un file gerber bisogna selezionare l'icona con la scritta "gerber" sovrapposta al simbolo del poligono stratificato; dopo aver effettuato questa selezione, si presenterà una finestra di dialogo per il caricamento dei file (Fig. 17).

Un esempio di file gerber visualizzato (si tratta del copper front della demoboard oggetto del nostro progetto pratico) è visualizzato in Fig. 18.

Come abbiamo fatto notare all'inizio di questo capitolo, il compito principale cui assolve Gerberview è l'ispezione dei file gerber generati; per fare un esempio, è possibile verificare che il file di foratura coincida con le piazzole dei piani di rame.

A questo scopo, una volta caricato un piano rame (come visto precedentemente) è possibile sovrapporre un file di foratura, caricandolo tramite l'icona posta a fianco a quella per il caricamento dei gerber file, e recante il simbolo di una trapano. Il risultato di questo tipo di

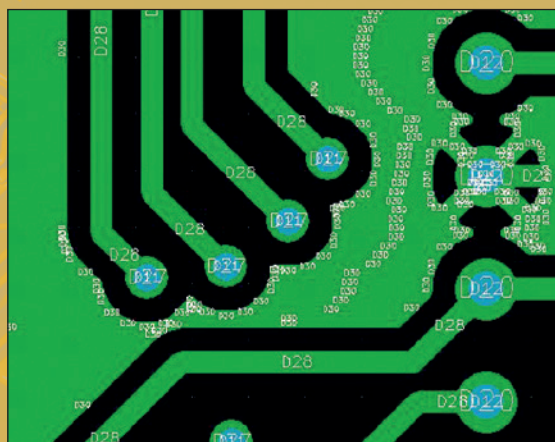


FIG. 19 - Sovrapposizione di un file di foratura ad un file gerber.

## Calcolatore per PCB

Un tool utile che KiCad mette a disposizione all'interno della suite è il calcolatore per PCB. Questo utile strumento non è altro che un insieme di tool di calcolo per PCB, tutti raggruppati sotto un unico strumento. L'utilità di calcolo (una cui schermata è riportata in Fig. 20), permette di:

- calcolare i valori del partitore resistivo dei regolatori variabili;
- calcolare la larghezza delle piste in funzione della corrente;
- calcolare i valori dell'isolamento elettrico in funzione della tensione;
- calcolare i parametri delle linee di trasmissione;
- calcolare i valori delle resistenze degli attenuatori RF.

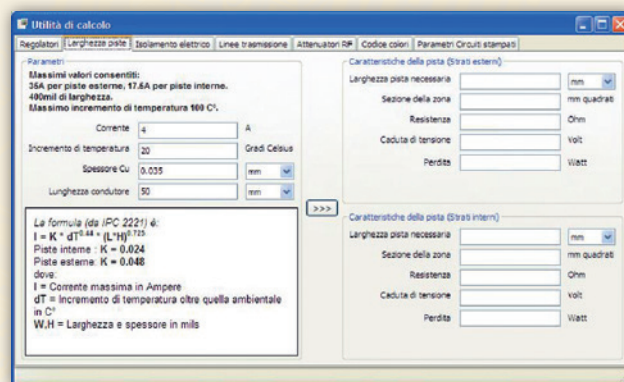


FIG. 20 - Calcolatore per PCB.

operazione è mostrato nel dettaglio riportato in Fig. 19: come è possibile notare, i fori sono perfettamente allineati con le piazzole, e siamo quindi sicuri che la realizzazione avverrà in maniera corretta.

Una volta completato il check anche sui file gerber, si può finalmente inviare il PCB in lavorazione presso l'azienda desiderata, attrezzata per realizzare lo stampato.

### CONCLUSIONI

Con questa sesta puntata abbiamo concluso il nostro viaggio alla scoperta di KiCad, completando la panoramica su Pcbnew e introducendo gli ultimi tool della suite.

Il progetto KiCad, che ci ha permesso di disporre di un prezioso software ECAD di progettazione di circuiti stampati che copre l'intera filiera della produzione, va avanti e ci auguriamo che il team di sviluppo che l'ha ideato e progettato possa offrire ancora, in futuro, delle versioni aggiornate di questo ottimo CAD elettronico open-source.