

Elettronica In

www.elettronica.in.it

€ 6,00 - Anno XXI - n. 201 - DICEMBRE 2015 / GENNAIO 2016

- Fish'n Tweet: leggiamo i Tweet con Fishino
- Trasforma Raspberry Pi in un oscilloscopio
- Breakout board NE555
- LoRa shield, il firmware
- STIMA: misurazione di parametri meteo e ambientali



Open Wheels 2.0: il Balancing Robot



- SmartEverything: tutto diventa smart
- Alla scoperta di OpenSCAD
- Media Center per Raspberry Pi con OSMC
- Smartem, il termostato intelligente

13
progetti
160
pagine



SVILUPPARE APP ANDROID CON
Processing



Direttore responsabile:

Arsenio Spadoni

(Arsenio.Spadoni@elettronica.in.it)

Redazione:

Gabriele Daghetta, Paolo Gaspari, Boris Landoni,
Marco Magagnin, Davide Scullino,
Alessandro Sottocornola
(redazione@elettronica.in.it)

Grafica:

Alessia Sfulcini, Filippo Gori
(Alessia.Sfulcini@elettronica.in.it)

Ufficio Pubblicità:

Monica Premoli (0331-752668)
(Monica.Premoli@elettronica.in.it)

Ufficio Abbonamenti:

Elisa Guarniero (0331-752668)
(Elisa.Guarniero@elettronica.in.it)

DIREZIONE, REDAZIONE, PUBBLICITÀ:

FUTURA GROUP srl - Divisione Editoriale
via Adige 11 - 21013 Gallarate (VA)
Telefono 0331-752668 Fax 0331-792287

Abbonamenti:

Annuo 10 numeri Euro 45,00

Esteri 10 numeri Euro 45,00 (digitale)

Le richieste di abbonamento vanno inviate a:

FUTURA GROUP srl

via Adige 11, 21013 Gallarate (VA)
tel. 0331-752668.

Distribuzione per l'Italia:

S.O. Di P. Angelo Patuzzi S.p.A.

via Bettola 18 - 20092 Cinisello Balsamo (MI)
Telefono 02-660301 Fax 02-66030320

Stampa:

ROTO3 Spa - Via Turbigo, 11/b
20022 CASTANO PRIMO (MI)

Elettronica In:

Rivista mensile registrata presso il Tribunale di Milano con il n. 245 il 3-05-1995.

Prezzo di copertina Euro 6,00. Gli arretrati nei formati cartaceo e digitale (pdf) sono acquistabili sul sito della rivista al prezzo di Euro 6,00.

Poste Italiane Spa Spedizione in abbonamento Postale - D.L. 353/2003 (conv. in L. 27/02/2004) art.1 comma 1 - DCB Milano.

FUTURA GROUP srl è iscritta al Registro Operatori della Comunicazione n. 23650 del 2/7/2013.

Impaginazione ed immagini sono realizzati in Desktop Publishing con programmi Adobe InDesign e Adobe Photoshop per Windows.

Tutti i contenuti della Rivista sono protetti da Copyright. Ne è vietata la riproduzione, anche parziale, la traduzione e più in generale la diffusione con qualsiasi mezzo senza l'autorizzazione scritta da parte dell'Editore. I circuiti, il firmware ed il software descritti sulla Rivista possono essere realizzati solo per uso personale, ne è proibito lo sfruttamento a carattere commerciale e industriale. Tutti possono collaborare con Elettronica In. L'invio di articoli, materiale redazionale, programmi, traduzioni, ecc. implica da parte del Collaboratore l'accettazione dei compensi e delle condizioni stabilite dall'Editore (www.elettronica.in.it/ase.pdf). Manoscritti, disegni e foto non richiesti non verranno in alcun caso restituiti. L'utilizzo dei progetti e dei programmi pubblicati non comporta alcuna responsabilità da parte della Società Editrice.

Futura Group
Edizioni

© 2015 FUTURA GROUP srl

Per un 2016 ancora più frizzante

Chiudiamo il 2015 e salutiamo il 2016 con un numero davvero eccezionale: 160 pagine di progetti, corsi, articoli e informazioni dal mondo dell'elettronica.

E questo dopo aver proposto nel corso del 2015 - il ventesimo di vita della nostra testata - una serie di progetti unici, innovativi e in molti casi anche divertenti.

Tra tutti ricordiamo *Fishino*, la scheda compatibile al 100% con Arduino Uno, con in più una connessione Wi-Fi, RTC e microSD, ma soprattutto con tutti gli strumenti software per una facile connessione a Internet. Una board, scusate la presunzione, che se fosse stata proposta dal gruppo di Banzi avrebbe "spaccato", una scheda di cui si sentiva il bisogno, specie in un mondo sempre più connesso.

A proposito di *Fishino*, ricordiamo il Contest che abbiamo promosso a partire da questo mese con ben tre oscilloscopi in palio: il premio per quanti proporranno le applicazioni più interessanti e innovative con questa scheda.

Tornando all'anno che sta per finire, come non ricordare la *Drink Maker*, la *3D Vertex*, la piattaforma domotica *UDoIT*, la board *EthernetMP3*, il "*Flusso Canalizzatore*", le board *LoRa* e le tantissime *Breakout Board*?

Tutti progetti che utilizzano quanto di meglio (e di più recente) i Produttori di Semiconduttori stanno proponendo al mercato. D'altra parte per realizzare progetti con caratteristiche innovative non si può prescindere dallo stato dell'arte della tecnologia elettronica.

Anche in questo numero, ecco dunque una sfilza di progetti in

grado di soddisfare qualsiasi palato: da *Stima*, il dispositivo per la misurazione dei parametri meteo e ambientali con connessione Bluetooth, a *Presepio*, il progetto di presepe Arduino-based con effetti luminosi e sonori, da *Fish'n Tweet*, per controllare le uscite di una board tramite Twitter (e Fishino), a *Open Wheels 2.0*, il nuovo balancing robot evoluto.

E poi, ancora, gli sviluppi del progetto domotico con *UDoIT*, la *breakout board* con NE555, l'articolo dedicato al software dello *shield LoRa*, l'evoluzione del termostato intelligente *Smartem* e la board per trasformare Raspberry Pi in un oscilloscopio. E come non bastasse, ecco la presentazione della scheda *SmartEverything*, l'introduzione a *OSMC*

(per trasformare Raspberry Pi in un prestante Media Center) e la presentazione dell'iniziativa *Precision Labs* di Texas Instruments. Per quanto riguarda i corsi, oltre alla terza puntata per imparare a realizzare *App Android con Processing*, iniziamo ad occuparci di *OpenSCAD*, il software di modellazione e di editing tridimensionale.

Insomma, un numero davvero imperdibile!

Buona lettura e buon 2016.



Arsenio Spadoni

(Arsenio.Spadoni@elettronica.in.it)

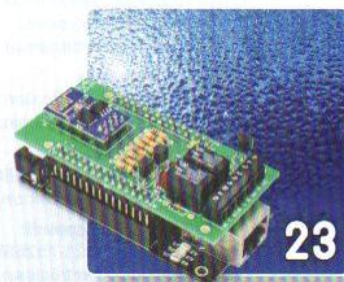
Sommario

Dicembre 2015
Gennaio 2016
n. 201

ARTICOLI

23 DOMOTICA SMARTEM, IL TERMOSTATO INTELLIGENTE

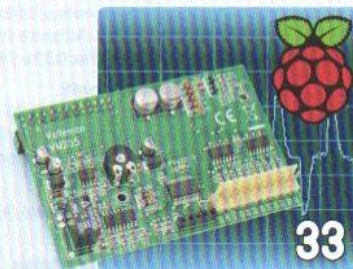
Dopo avervi descritto il nuovo modulo EM1001 Tibbo e il suo impiego in un termostato, vi descriveremo le modalità di upload dei dati su un server esterno e la trasmissione WiFi. Terza ed ultima puntata.



23

33 STRUMENTAZIONE OSCILLOSCOPIO PER RASPBERRY PI

Trasformiamo il nostro Raspberry Pi modello B in un oscilloscopio in grado di visualizzare sia segnali analogici che digitali.



33

43 LABORATORIO BREAKOUT BOARD NE555

Permette di realizzare circuiti astabili e monostabili sfruttando il più popolare tra i circuiti integrati timer.

66 DIDATTICA TORNANO I CORSI FUTURA ACADEMY

Teoria, laboratori pratici e supporto di tutor per chi vuole acquisire competenze utili a scuola o nel mondo del lavoro o anche solo per nutrire la propria passione.

49 NETWORKING STIMA: MISURAZIONE DI PARAMETRI METEO E AMBIENTALI

Monitoriamo temperatura, umidità e altri parametri pubblicandone i valori su un servizio web pubblico. Prima puntata.



43



59

59 WIRELESS LORA SHIELD

Fornisce ad Arduino la comunicazione long-range basata su tecnologia LoRa. Terza ed ultima puntata.



49



66



IN COPERTINA:

La versione natalizia del Balancing Robot 2.0 e Presepio, la centralina basata su Arduino, con effetti luminosi e sonori per realizzare un tecnopresepe.

ELENCO INSERZIONISTI

- Expo Elettronica - Blu Nautilus
- Fiera di Novegro
- Futura Elettronica
- Microchip
- National Instruments
- RM Elettronica
- Robot & Maker
- www.mdsrl.it

69 WIRELESS FISH'N TWEET

Controlliamo le uscite di uno shield di espansione Arduino montato sulla nostra Fishino Uno, tramite Twitter.



69

82 GADGET PRESEPIÙ: IL PRESEPE CON ARDUINO

Centralina capace di simulare ciclicamente l'avvicinarsi del giorno e della notte, pilotando opportunamente fino a sei uscite. Può anche controllare strip NeoPixel e, per rendere l'atmosfera più natalizia, riprodurre musiche a vostra scelta.



82

93 LABORATORIO PRECISION LABS

Texas Instruments supporta i progettisti di elettronica analogica con una importante iniziativa on-line: ne parliamo con Art Key, a capo di questo progetto.



93

La tiratura di questo numero è stata di 18.500 copie.



Mensile associato all'USPI,
Unione Stampa Periodica Italiana

RUBRICHE

- 01 EDITORIALE
- 05 ELETTRONICA INSIEME
- 13 EVENTI
- 15 APPUNTAMENTI
- 19 COMPONENTI & SISTEMI
- 149 SCIENZA & TECNOLOGIA
- 157 RINNOVABILI

97 APPLICAZIONI MEDIACENTER CON OSMC

Trasformiamo, grazie a un sistema operativo Linux-like, Raspberry Pi in un prestante ed economico Media Center.



97

105 ROBOTICA OPEN WHEELS 2.0

Scheda scalabile per prototipare applicazioni di motion control sia di balancing robot sia di qualsiasi apparato su ruote.



105

121 DOMOTICA DOMOTICA OPEN SOURCE CON I MODULI UDOIT

Attiviamo le funzionalità WiFi per controllare il sistema UDOIT senza cavi di rete, sia come client che come Access Point.



121

131 DIDATTICA CONOSCIAMO OPENS CAD

Chi stampa oggetti 3D ha bisogno di programmi di modellazione ed editing tridimensionale per creare e ritoccare file; impariamo assieme ad usare OpenSCAD. Prima puntata.



131

141 HI-TECH SMART EVERYTHING: TUTTO DIVENTA SMART

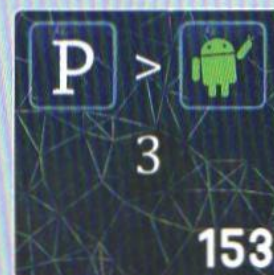
L'IoT è alle porte ed è il momento di cominciare a sperimentare con questa scheda Arduino-like dotata di vari sensori e connettività wireless.



141

153 DIDATTICA CORSO: SVILUPPARE APP ANDROID CON PROCESSING

Impariamo a utilizzare i sensori integrati nel nostro smartphone come l'accelerometro, il sensore di prossimità e quello di illuminazione ambiente, ecc. Terza puntata.



153

Queste pagine sono dedicate alle richieste, ai suggerimenti ed alle segnalazioni dei lettori. Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale. Contattateci numerosi all'indirizzo: redazione@elettronica.in.it

I primi prodotti per il protocollo Thread

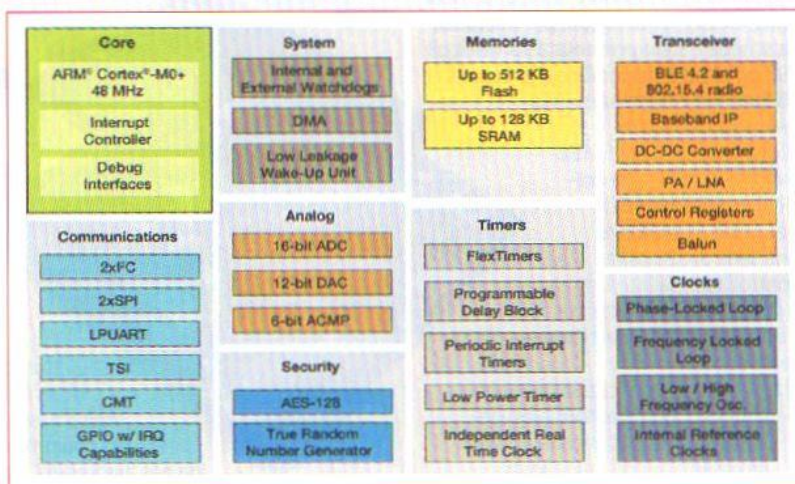
Dopo l'annuncio dell'estate 2014, non ho più sentito parlare di Thread, il protocollo wireless per collegare in rete elettrodomestici e sistemi per home automation. Anche questo standard è finito nel dimenticatoio?

Mauro Luppi - Roma

Poco più di un anno per la diffusione di un protocollo wireless è un tempo decisamente troppo breve. Ricordiamo che Thread Group è nato nel luglio del 2014 su iniziativa di Nest Labs (Google) e di poche altre aziende (Yale Security, Silicon Labs, Samsung Electronics, Freescale Semiconductor, Big Ass Fans e ARM) con l'obiettivo di sviluppare un protocollo per reti mesh a basso consumo per home automation, basato sullo standard IEEE 802.15.4 e in grado di sostituire tutti gli attuali protocolli di comunicazione (ZigBee, Z-Wave, Bluetooth e WiFi) utilizzati in questo ambito, superandone le numerose limitazioni (eccessivo consumo, scarsa sicurezza, basse prestazioni, ecc.).

Thread è un protocollo open che gestisce nativamente IPv6 (la nuova generazione degli indirizzi del protocollo Internet), che definisce i primi due livelli del modello OSI e che permette di connettere in modo sicuro oltre 250 dispositivi ad una rete mesh wireless a basso consumo. Le piattaforme IoT esistenti e milioni di device 802.15.4 già sul mercato possono funzionare sulle reti Thread in quanto è sufficiente un aggiornamento firmware. Le attuali tecnologie ZigBee e Z-Wave non sono basati su IP e non possono funzionare facilmente con gli standard basati su protocolli Internet anche se esiste un

THREAD



livello di adattamento che abbina IPv6 agli standard di rete non basati su IP e che prende il nome di 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks).

Nell'anno e mezzo trascorso dalla nascita, l'Associazione ha pubblicato (luglio 2015) le specifiche tecniche di Thread ed ha visto il numero di associati salire ad oltre ottanta. Il gruppo ha anche pianificato l'avvio di un programma di certificazione che dovrebbe partire a breve. Infine sono stati annunciati i primi prodotti. Ad esempio, Freescale Semiconductor, uno dei promotori dell'iniziativa, ha presentato il nuovo Kinetis KW41Z, un microcontrollore con connettività wireless per applicazioni in ambito Internet of Things (IoT) dotato di radio multi-mode in grado di gestire simultaneamente sia lo standard Thread che quello Bluetooth Smart/BLE.

Lo standard LVDS

Sento spesso parlare di standard LVDS, specie in ambito video. Quali sono le caratteristiche di questa tecnologia?

Roberto Minerva - Roma

Lo standard LVDS (Low-voltage differential signaling), noto anche come TIA / EIA-644, è uno standard che

specifica le caratteristiche elettriche di un protocollo di comunicazione seriale di tipo differenziale a bassa potenza. L'LVDS può funzionare a velocità molto elevate utilizzando economici cavi twisted pair in rame. Poiché LVDS è una specifica a livello fisico, molti standard e le applicazioni di comunicazione dati la utilizzano, aggiungendo il layer dati, come definito nel modello OSI. LVDS è stato introdotto nel 1994, ed è diventato popolare in prodotti come televisori LCD, sistemi di infotainment automotive, telecamere industriali e visione artificiale, computer portatili e tablet, sistemi di comunicazione. Le applicazioni tipiche sono video ad alta velocità, grafica, trasferimenti di dati. La bassa tensione differenziale utilizzata, circa 350 mV, consente ai sistemi LVDS di consumare pochissima energia rispetto ad altre tecnologie di trasporto dati. Ad esempio, rispetto ad uno standard simile (RS-422), il consumo di energia è di appena 8,75 mW contro i



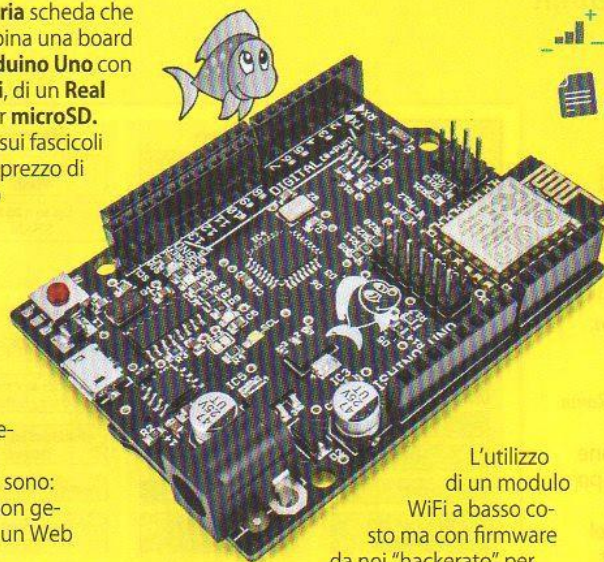
PROGETTA & VINCI CON FISHINO! TRE OSCILLOSCOPI IN PALIO

Fishino UNO è la rivoluzionaria scheda che tutti aspettavano e che combina una board compatibile al 100% con **Arduino Uno** con l'aggiunta di un **modulo WiFi**, di un **Real Time Clock** e di un lettore per **microSD**. La scheda è stata presentata sui fascicoli 198 e 199 ed è disponibile al prezzo di 36 Euro IVA compresa presso www.futurashop.it.

L'integrazione della funzionalità WiFi (oltre che dell'RTD e della microSD) permette di realizzare una serie di dispositivi controllabili via Internet o in grado di connettersi e trasmettere dati rilevati in precedenza.

Tra le possibili applicazioni ci sono:

- sistemi di home automation gestibili via Internet tramite un Web Browser;
- data logger portatili in grado di connettersi e scaricare i dati sulla rete quando si entra nel campo di copertura di una rete WiFi;
- robot controllabili via rete e in grado di trasmettere tramite essa i dati rilevati da sensori.



L'utilizzo di un modulo WiFi a basso costo ma con firmware da noi "hackerato" per ottenere prestazioni elevate e capace di funzionare anche come access point, ovvero senza la necessità di una struttura di rete WiFi esistente, permette il controllo via cellulare in qualsiasi momento, anche in assenza di copertura di rete, rendendo il dispositivo sempre interattivo.



La possibilità di eseguire l'aggiornamento degli sketch via Internet, già prevista a livello hardware, permetterà inoltre di avere dispositivi sempre aggiornati senza la necessità di doverli collegare fisicamente ad un computer. Per incentivare l'uso di questa board e proporre alla nostra community applicazioni originali, abbiamo dato vita a questa iniziativa che **PREMIA CON TRE OSCILLOSCOPI**, le migliori applicazioni basate sulla board Fishino. Tutti i progetti verranno presentati on-line mentre quelli che si classificheranno ai primi tre posti verranno pubblicati anche sulla rivista. Tutte le informazioni relative a questa iniziativa sono disponibili alla pagina:

www.elettronica.in/fishinocontest
La scadenza di questo concorso è fissata al **31 marzo 2016**. Affrettatevi!

Tabella 1
Parametri di un segnale differenziale

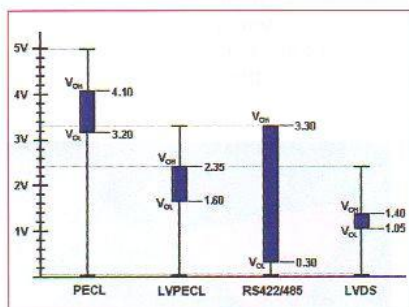
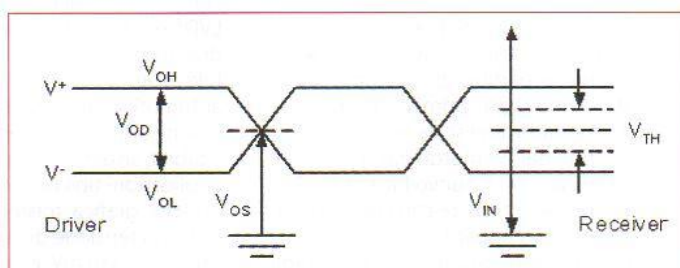


Tabella 2 - Confronto tra differenti tecnologie differenziali

90 mW. In una tipica implementazione, il trasmettitore inietta infatti una corrente costante di 3,5 mA che passa attraverso una resistenza di terminazione di 100/120 Ohm dando luogo ad una differenza di potenziale di 350 mV. Nell'LVDS il valore assoluto della differenza di tensione misurata tra le due linee di uscita del driver è compreso tra 247 e 454 mV, con 350 mV di valore tipico.

Come in tutti i sistemi differenziali, la riduzione del rumore è dovuta al flusso di corrente, uguale ed opposto nei due cavi, che crea campi elettromagnetici che tendono ad annullarsi reciprocamente. Lo

stesso avviene per le interferenze elettromagnetiche captate: il rumore incide in ugual misura su ciascun conduttore per poi annullarsi.

La bassa tensione differenziale consente di utilizzare circuiti integrati alimentati a 2,5 volt o meno. La tensione di offset è la tensione di modo comune del segnale differenziale, ed è essenzialmente la tensione media delle due linee della coppia differenziale rispetto a terra. I valori minimi e massimi per VOS secondo lo standard sono 1,125 e 1,375 V, con valore tipico di 1,2 V. Con un'oscillazione di tensione di 350 mV centrata a 1,2 V, è disponibile un margine di 1,025 V su ciascun lato del segnale. Con questo margine, il ricevitore respinge efficacemente il rumore di modo comune.

LVDS è diventato popolare nella metà degli anni '90, prima di allora le risoluzioni dei monitor dei computer non erano così elevate da richiedere una grande velocità di trasferimento.

Attualmente, nei dispositivi più performanti, l'LVDS è in grado di raggiungere una velocità di quasi 1 Gbps; per esigenze

OSCILLOSCOPIO 2 CANALI 200 MHz CON GENERATORE DI FUNZIONI

- ✓ Frequenza di campionamento fino a 1 GS/s
- ✓ Display LCD da 7" 64.000 colori 800x480 pixel
- ✓ Generatore di forme d'onda arbitrarie fino a 25 MHz con convertitore A/D 12-bit
- ✓ Funzione FFT integrata
- ✓ Software per l'analisi e il controllo in tempo reale con il PC
- ✓ Porta USB Host / USB Device

1° PREMIO



2° PREMIO

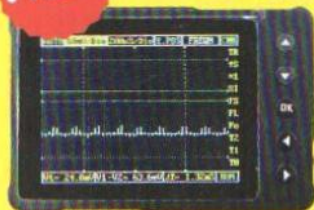


OSCILLOSCOPIO 2 CANALI 70 MHz

- ✓ Frequenza di campionamento fino a 1 GS/s
- ✓ Display LCD da 7" 64.000 colori 800x480 pixel
- ✓ Funzione FFT integrata
- ✓ Software per l'analisi in tempo reale con il PC
- ✓ Porta USB Host

3° PREMIO

DSO Nano



OSCILLOSCOPIO TASCABILE MONOCANALE CON MEMORIA - 1MHz

- ✓ Frequenza di campionamento 1 MS/s
- ✓ Slot per micro SD Card con possibilità di salvare le forme d'onda
- ✓ Generatore di segnale da 10 Hz a 1 MHz
- ✓ Display LCD TFT 2,8" a colori
- ✓ Connessione USB e batteria ricaricabile al litio
- ✓ Completo di puntali ad uncino e a singolo terminale

ancora più spinte è possibile fare ricorso a più coppie, sino a 8, per una velocità di trasferimento di 7/8 Gbps. Basso consumo di energia, elevata immunità ai rumori ed elevata velocità di trasferimento, rendono ancora attuale e molto utilizzata questa tecnologia.

LoRaWAN per reti low-cost IoT gestite dagli utenti

Sto seguendo con interesse gli articoli e i progetti pratici basati sulla tecnologia LoRa. Vorrei realizzare una rete wireless IoT per la mia città da condividere con altri amici. Potete darmi qualche consiglio sul migliore hardware e software da utilizzare?

Antonio Castiglioni - Legnano (MI)

Per realizzare la tua rete puoi ispirarti all'ultimo dei tre articoli che abbiamo dedicato a questa tecnologia e che trovi in questo stesso numero della rivista. Puoi anche dare un'occhiata

all'iniziativa *The Network Things* (<http://thethingsnetwork.org/>) che si propone di realizzare una rete globale, aperta, gratuita e low cost, per acquisire i dati degli oggetti dotati di connettività wireless long-range. L'ideatore di questa iniziativa, Wienke Giezeman, è riuscito in sole sei settimane a coprire l'intera città di Amsterdam utilizzando 10 gateway LoRaWAN di tipo commerciale che hanno un costo unitario compreso tra mille e 2 mila euro. Ora, *The Things Network*, con un'iniziativa sbarcata su Kickstarter, si propone di realizzare dei gateway molto più economici, del costo unitario di circa 200 euro. Per quanto riguarda il funzionamento dell'infrastruttura, vengono forniti componenti open source per il routing e la gestione dei dati, compatibili con la specifica LoRaWAN 1.1. I due componenti principali della architettura di rete sono il *The Things Router* e il *The Things Handler*. I pacchetti di dati provenienti dai nodi e ricevuti dai gateway, vengono inoltrati a uno o più Router, in funzione della configurazione effettuata dal proprietario del gateway. Il

gateway fornito da *The Things Network* è pre-configurato con il Router di default ospitato dalla Fondazione per una installazione plug-and-play. I Router pubblicano i pacchetti utilizzando il protocollo MQTT, il più adatto a questo tipo di messaggi, con funzionamento sia in uplink che in downlink.

Per la gestione dei dati viene utilizzato il tool node RED che consente una facile connessione alle piattaforme di cloud più diffuse quali IBM Bluemix, AWS IoT, FIWARE, Parse.com, IFTTT e OpenSensors.io.

L'iniziativa di Wienke Giezeman ha riscosso subito un grande successo, tanto che in decine di città di tutto il mondo sono nate delle community che hanno dato vita a reti locali LoRaWAN per le applicazioni più disparate.

Sfruttando questa piattaforma – completamente gratuita – potresti realizzare in pochissimo tempo la tua rete; per una città come la tua dovrebbe essere sufficiente un solo gateway, tre al massimo.

Per quanto riguarda gli end-device, anche in questo caso puoi trarre ispirazione dai nostri progetti oppure aspettare i dispositivi proposti da *The Things Network* che, oltre al gateway low-cost, ha in fase di sviluppo due unità remote, la prima basata su Arduino, e programmabile con il tradizionale IDE, la seconda molto più piccola, a forma di portachiavi, dotata di sensori di movimento e di temperatura. Ovviamente entrambe open source, sia nell'hardware che nel software, così come il gateway.



KNX per domotica e building automation

Ho letto recentemente di tre nuovi chip di On Semiconductor compatibili con lo standard KNX. Di cosa si tratta esattamente?

Luca Nosetti - Milano

I dispositivi cui fai riferimento sono i modelli NCN5110, NCN5121 e NCN5130, ricetrasmittitori twisted-pair certificati KNX e adatti a realizzare impianti per l'automazione di edifici.

Lo standard KNX è il primo standard aperto al mondo di building automation, indipendente dalla piattaforma e approvato come standard europeo (EN 50090 - EN 13321-1) e mondiale (ISO/IEC 14543). Lo standard è stato sviluppato da KNX Association sulla base dell'esperienza dei suoi predecessori BatiBUS, EIB e EHS.

Questo standard viene largamente utilizzato in tutte le applicazioni della casa e dell'edificio, che vanno dall'illuminazione al controllo di diversi sistemi di sicurezza, riscaldamento, ventilazione, aria condizionata, monitoraggio, allarme, controllo dell'acqua e misurazione intelligente. BatiBUS ebbe successo soprattutto in Francia, Italia e Spagna, EIB più nei paesi tedeschi e nordici, EHS era invece la soluzione preferita dai produttori di elettrodomestici bianchi e bruni. Nel 1997, i tre consorzi decisero di associarsi per sviluppare insieme il mercato delle "Case intelligenti". Lo scopo era quello di trovare un nuovo standard industriale comune da proporre come standard internazionale. La specifica KNX è stata pubblicata nel primo trimestre del 2002 dall'Associazione KNX; è basata sulla specifica EIB, completata con nuovi meccanismi di configurazione e con mezzi di comunicazione originaria-

mente sviluppati da BatiBUS ed EHS. Per i soci di KNX Association, l'accesso al sistema è privo di royalty e può essere implementato su qualsiasi piattaforma; inoltre tutti i prodotti marchiati con il logo KNX sono certificati a garanzia della totale compatibilità, interconnessione ed interoperabilità dei dispositivi stessi. Entrando più nello specifico, possono essere utilizzati differenti mezzi di trasmissione, in combinazione con una o più modalità di configurazione, permettendo a ciascun costruttore di scegliere la combinazione più adeguata al segmento di mercato a cui essa è indirizzata ed all'applicazione. Attualmente vengono utilizzati i seguenti mezzi trasmissivi:

- **Doppino intrecciato (KNX TP):** lo standard KNX è trasmesso mediante un cavo bus separato, gerarchicamente strutturato in linee ed aree. Il Twisted Pair di tipo 0 (TP-0) è basato su cavo a conduttori intrecciati con bitrate di 4800 bits/s, simile al BatiBUS. I prodotti certificati KNX TP-0 funzionano sulla stessa linea bus dei componenti certificati BatiBUS ma non scambiano informazioni con essi. Il Twisted Pair di tipo 1 (TP-1) utilizza lo stesso mezzo trasmissivo ma con bitrate di 9600 bit/s, simile allo EIB. I prodotti certificati EIB e KNX TP-1 funzionano e comunicano fra di loro sulla stessa linea bus.

- **Modulazione su rete elettrica (KNX PL):** lo standard KNX viene trasmesso sulla preesistente rete elettrica mediante onde convogliate. Il Power Line a 110 kHz (PL-110) presenta un bitrate di 1200 bit/s, simile a quello di EIB. I prodotti certificati EIB e KNX PL-110 funzionano e comunicano fra di loro sulla stessa rete di distribuzione dell'alimentazione elettrica. Il Power Line a 132 kHz (PL-132) presenta un bitrate di 2400 bits/s ed è ispirato da EHS. I componenti certificati KNX PL-132 ed EHS 1.3a funzionano sulla stessa rete



ma non comunicano fra loro senza un convertitore di protocollo dedicato.

- **Radiofrequenza (KNX RF):** lo standard KNX viene trasmesso attraverso il segnale radio. I dispositivi possono essere uni- o bidirezionali e presentano un bitrate di 38,4 kbit/s, sviluppato direttamente all'interno della piattaforma standard KNX.

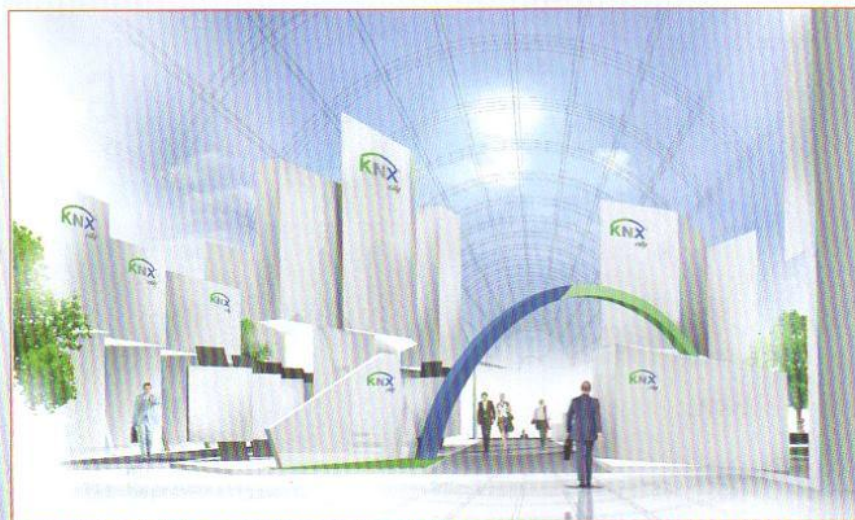
- **IP/Ethernet (KNX IP):** molto diffuso, può essere utilizzato in unione con la specifica "KNXnet/IP", che permette l'incanalamento o il routing di frame KNX incapsulati in IP.

KNX può essere utilizzato sia negli edifici nuovi sia in quelli preesistenti. Le installazioni KNX possono essere facilmente ampliate ed adattate secondo le nuove esigenze, in poco tempo e con investimenti finanziari minimi (es: quando subentrano nuovi inquilini in un edificio commerciale). KNX offre diverse modalità di configurazione per la realizzazione di progetti KNX, suddivise in livelli di difficoltà. La modalità di configurazione E-Mode (easy mode) si utilizza per semplici installazioni ed è adatta per gli installatori elettrici con competenze di base su KNX. La modalità S-Mode (System Mode) è adatta per gli installatori elettrici e gli integratori di sistema, i quali hanno conseguito una formazione per l'utilizzo dello strumento ETS e che sono in grado di realizzare installazioni complesse. Le modalità di configurazione sono:

- **Configurazione semplice (E-mode):** la configurazione si attua senza l'aiuto di un PC ma usufruendo di un controller centrale, di dischi codificatori o mediante pulsanti. I prodotti compatibili per la modalità di installazione E-mode permettono normalmente delle funzioni limitate e sono stati riservati ad installazioni di media grandezza.

- **Configurazione di sistema (S-mode):** la progettazione, l'installazione e la configurazione avviene mediante un PC su cui è installato l'ETS Software, mentre i dati di prodotto del costruttore sono contenuti nel database ETS. La modalità S-mode è riservata agli installatori elettrici ed integratori di sistema certificati KNX ed è applicata in grandi e complesse installazioni.

Molti costruttori di prodotti KNX offrono gateway con altre reti, per esempio con



altri sistemi d'automazione dell'edificio, reti telefoniche, reti multimedia, reti IP, etc. Il sistema KNX può essere mappato per gli oggetti BACnet (come previsto dallo standard internazionale ISO16484-5) oppure offre la possibilità d'interfacciamento con la tecnologia DALI. Maggiori informazioni sono disponibili sul sito: www.knx.it

I vantaggi del nitruro di gallio

Sento sempre più spesso parlare di transistor realizzati con nuovi materiali e nuovi composti, GaN, SiC, ecc. Quali i vantaggi rispetto ai tradizionali semiconduttori al silicio?

Marino Sanna – Cagliari

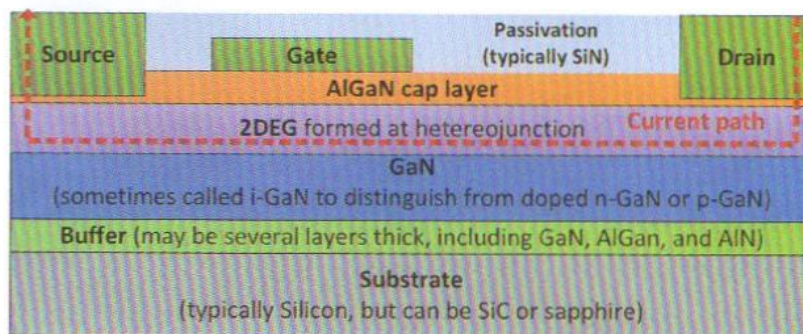
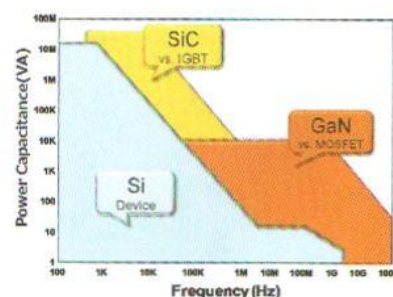
La storia dei semiconduttori inizia negli anni '50 del secolo scorso con i primi transistor realizzati utilizzando il germanio, ben presto, però, sostituito dal silicio, materiale con caratteristiche nettamente superiori sotto molti punti di vista. Ed è proprio il silicio, che ha caratterizzato la storia dei transistor prima e dei circuiti integrati poi. Grazie alle sue eccellenti caratteristiche, il silicio ha permesso la nascita di un'industria di altissima tecnologia con processi di fabbricazione sempre più spinti, dando un contributo decisivo allo sviluppo di prodotti miniaturizzati ad alte prestazioni che sono diventati una parte importante nella nostra vita quotidiana. Negli ultimi anni, tuttavia, la ricerca di base e quella applicativa ha sondato la possibilità di realizzare semiconduttori utilizzando altri materiali; tra questi, il GaN (nitruro di gallio) e il SiC (carburo di silicio) sono quelli che hanno offerto le prestazioni migliori, specie per quanto riguarda i dispositivi di potenza e quelli ad alta frequenza, tanto che questi materiali sono stati iscritti di diritto nella categoria dei "Next Generation Power Semiconductors". Rispetto al silicio, GaN e SiC hanno un gap di banda più ampio (Si: 1,1V, SiC: 3,3V, GaN: 3,4V); per questo motivo, i dispositivi realizzati con questi materiali vengono anche definiti "Wide Band Gap Semiconductors". Altra differenza rispetto al silicio riguarda la figura di merito ($\epsilon \mu e E_c^3$) che nel caso del SiC è 440 volte più grande, e in quello del GaN è 1130 volte superiore. Poiché i dispositivi SiC hanno perdite significativamente inferiori rispetto ai dispositivi in silicio, l'utilizzo di questo materiale consente di ridurre drasticamente le dimensioni delle apparecchiature. In un dispositivo SiC,

Characteristics of Silicon / SiC / GaN

Substance	Si	SiC	GaN
Bandgap eV	1.1	3.3	3.4
Electron Mobility cm ² /Vs	1350	700	1500
Electric Field for Breakdown kV/cm	0.3	3.0	3.3
Figure of Merit	1	440	1130

Characteristics of GaN / SiC compared to Silicon:

High Breakdown voltage: Up to 10 times higher	Compact: Reduced up to 1/1000
High Heat Resistance: Up to 1000°C	High Speed: Up to 100MHz



metà del silicio viene sostituito con carbonio, dando luogo ad una struttura cristallina più precisa e stabile. In questo modo la tensione di rottura risulta di un ordine di grandezza superiore al silicio, consentendo l'impiego di strati molto più sottili.

Nel caso del GaN la tensione di breakdown è ancora più alta e la struttura utilizzata (GaN su Si o GaN su SiC) presenta una maggiore stabilità. I sottili strati così ottenuti consentono di ottenere ottime prestazioni anche in alta frequenza.

Ad oggi, dunque, la tecnologia GaN su Si, offre come minimo 8 volte la densità di potenza rispetto all'attuale tecnologia, con un miglioramento dell'efficienza dal 40% al 70%.

Per quanto riguarda le ricadute commerciali, da pochissimo tempo sono disponibili transistor RF di potenza realizzati con queste tecnologie, in particolare con la tecnica del nitruro di gallio su carburo di silicio. Ad esempio, all'inizio di quest'anno, Infineon ha presentato il suo portfolio di brevetti relativo alla tecnologia GaN e ha annunciato l'espansione della sua offerta GaN-on-Silicon (GaN/Si) con processo epitassiale GaN/Si a 100V-600V, tecnologie in gran parte derivanti dall'acquisizione di International Rectifier. Sempre Infineon ha presentato all'ultima edizione di EuMW i suoi primi dispositivi RF con tecnologia "GaN on SiC" che garantiscono un'efficienza superiore del 10% e, soprattutto, una densità di potenza cinque volte superiore rispetto alla tecnologia LDMOS utilizzata

finora. Tutto ciò avrà un significativo impatto sulle dimensioni e sui requisiti di alimentazione degli amplificatori di potenza impiegati oggi nelle base-station che operano nelle gamme di frequenza 1,8-2,2 GHz e 2,3-2,7 GHz. I nuovi dispositivi consentono di operare con una larghezza di banda doppia rispetto ai transistor RF LDMOS, permettendo alle base-station di supportare più frequenze operative. Essi presentano anche una maggiore larghezza di banda istantanea che consente agli operatori di realizzare tecniche di aggregazione dei dati, offrendo servizi in 4.5G.

Ricordiamo infine che la tecnologia GaN ha avuto un ruolo fondamentale nello sviluppo e nella realizzazione dei dispositivi laser blu e, soprattutto, dei LED a luce bianca.

Web Forum e supporto tecnico

Hai un problema con uno dei circuiti pubblicati? Vorresti effettuare una modifica? Accedi al nostro Web Forum dove i nostri tecnici (ma anche gli altri lettori) ti aiuteranno a chiarire qualsiasi dubbio di natura tecnica.

Collegati a:

www.elettronica.in/webforum

Corsi Microchip Technology 2016



Per imparare ad utilizzare più rapidamente un microcontrollore o un dispositivo digitale, la soluzione migliore è quella di frequentare i workshop e gli hands-on che tutti i produttori di se-

miconduttori organizzano periodicamente.

Anche Microchip organizza seminari ed eventi formativi presso gli RTC (Regional Training Centers), centri di formazione distribuiti in tutto il mondo ed attrezzati con i più recenti strumenti di sviluppo, nei quali sono presentate le tecniche di controllo più avanzate da esperti istruttori Microchip.

Negli RTC si tengono corsi pratici base, intermedi ed avanzati, sulle linee di prodotti Microchip, per aiutare i progettisti ad accelerare i tempi di sviluppo e ridurre il costo delle loro applicazioni embedded. I corsi offrono un'ampia varietà di argomenti che aiutano i progettisti a mantenersi aggiornati nel mondo competitivo dei sistemi di controllo embedded, riuscen-

do ad anticipare le tendenze del mercato industriale per implementare le tecnologie più innovative. Durante lo svolgimento dei corsi si ha tutto il tempo per un'interazione diretta con i tecnici Microchip.

I primi corsi del 2016 prevedono il 12 gennaio, in mattinata, l'Introduzione ai sistemi di sviluppo Microchip e nel pomeriggio l'Introduzione ai microcontrollori Microchip della famiglia High-Performance 8-bit (architettura e set di istruzioni assembly); il 13 gennaio è in programma un corso sull'utilizzo del compilatore MPLAB XC8 mentre il 14 gennaio il corso tratta la Configurazione e l'utilizzo delle periferiche dei microcontrollori Microchip High-Performance 8-bit in linguaggio C.

Gli eventi sono in programma presso l'RTC di Legnano (MI).

www.microchip.com



Seminario pratico STM32L4

Questa iniziativa, promossa da Silica e STMicroelectronics, ha lo scopo di fare conoscere e insegnare a realizzare applicazioni pratiche con i microcontrollori della famiglia STM32L4 di ST, basati su tecnologia ARM Cortex-M4, in grado di offrire elevatissime prestazioni con un consumo di energia particolarmente contenuto (Ultra-Low-Power).

Il seminario - tenuto da personale tecnico di ST - combina la teoria con esercitazioni pratiche, coinvolgendo anche addetti al marketing in grado di offrire una comple-

ta panoramica sui prodotti STM32.

Ogni partecipante riceverà un kit omaggio STM32L4 Discovery e un set completo di lezioni e manuali; gli esercizi pratici consentiranno di evidenziare la tecnologia alla base dei prodotti STM32L4, mettendo in evidenza le possibili applicazioni in campo industriale e nel settore dell'elettronica indossabile.



Verranno testate le funzionalità più significative, dalle linee di comunicazione al clock, e analizzate le modalità d'uso più frequenti, compresi trucchi e scorciatoie.

Il seminario, della durata di una giornata, è gratuito, previa registrazione sul sito del seminario.

A Milano il 16 e il 17 febbraio 2016 e a Napoli il 17 marzo 2016.

www.st.com

RM ELETTRONICA

Forniture per hobbisti, scuole e industria

Distributore autorizzato:

**FUTURA
ELETTRONICA**

**ELSE
Kit**

**NUOVA
ELETTRONICA**

Il tuo punto
di riferimento
per l'elettronica
a ROMA

Via Val Sillaro 38 • 00141 Roma • Tel: 06-8104753



ELETTRONICA IN - Dicembre 2015 / Gennaio 2016

■ 23-25 FEBBRAIO 2016

EMV 2016

Dusseldorf - Germania

EMV è la più importante manifestazione mondiale dedicata alla compatibilità elettromagnetica. Con le tecnologie elettroniche che diventano sempre più pervasive, dall'automotive all'industria, dalle applicazioni domestiche al wearable, le problematiche relative alle interferenze tra i vari dispositivi si fanno sempre più complesse, con la necessità di

valutare attentamente questo aspetto durante tutto il ciclo produttivo: progettazione, prototipazione, produzione e test. Nell'area espositiva della manifestazione di Dusseldorf, vengono proposti tutti gli strumenti hardware e software che aiutano i progettisti e le aziende a rispettare i parametri previsti dalle normative vigenti, mentre convegni e seminari portano all'attenzione dei professionisti del settore i più recenti sviluppi e i processi produttivi più adatti a raggiungere questi risultati nel modo più eco-

emv

Düsseldorf, 23. – 25.02.2016

nomico possibile. I numeri di EMV 2016 dovrebbero ricalcare quelli dell'edizione 2015, che ha visto la presenza di 130 espositori su una superficie complessiva di 4.000 mq circa; 74 provenivano dalla Germania mentre 56 provenivano dal resto del mondo.

2.643 sono stati, invece, i visitatori professionali.

<http://e-emc.com>



■ 17-19 GENNAIO 2016

Intersec 2016

Dubai • UAE

A cavallo tra Europa e Asia, tra aree geografiche con vocazioni differenti (R&D Europa e USA, manufacturing l'Asia), Intersec – che si svolge a Dubai – rappresenta l'ideale punto di incontro tra Aziende e mercati che hanno come riferimento la Sicurezza, attiva e passiva, e che trovano in quest'area anche un importante sbocco commerciale.

Giunta alla sua 18^a edizione, Intersec è diventata una delle più importanti fiere a livello mondiale per la Sicurezza e la protezione antincendio, in grado di coinvolgere quest'anno quasi 30 mila visitatori (per l'esattezza sono stati 27.303 nel 2015, provenienti da 113 differenti paesi). Grazie anche all'organizzazione e

alla promozione di Messe Frankfurt, la manifestazione di Dubai è in grado di mettere in mostra una gamma completa di prodotti in ambito Commercial Security, Information Security, Fire & Rescue, Safety & Health e Homeland Security & Policing, a cui fanno capo le cinque sezioni in cui si articola l'esposizione. Durante l'evento sono in programma anche numerose conferenze caratterizzate dalla presenza di



relatori di alto profilo e da temi di grande attualità, dalla cyber security in ambito industriale e commerciale agli strumenti per la sicurezza sul lavoro.

www.intersecexpo.com



■ 9-12 FEBBRAIO 2016

ISE 2016

Amsterdam • Olanda

Anche quest'anno sarà il centro fieristico RAI di Amsterdam ad ospitare la 13ª edizione di Integrated Systems Europe (ISE), la più importante manifestazione continentale per il settore A/V professionale e per i sistemi elettronici integrati. In considerazione del continuo e consistente incremento di visitatori che ha caratterizzato le ultime edizioni (51 mila presenze nel 2014 e quasi 60 mila l'anno scorso), nonché dell'occupazione di tutti gli spazi disponibili del complesso fieristico (compreso il nuovo Amtrium building, ancora in costruzione e già interamente prenotato da Samsung), quest'anno, per la prima volta, la manifestazione si svolgerà nell'arco di quattro giorni anziché tre, come consuetudine, dando tempo e modo ai visitatori di scoprire tutte le opportunità dell'evento e creare le migliori occasioni di business. Anche perché gli espositori sono più di mille ed in programma ci sono pure numerosi Convegni, Seminari e Laboratori pratici. In mancanza di ulteriori spazi, e considerati i numeri, la scelta di prolungare di un giorno la Fiera è stata quasi obbligatoria. ISE è sicuramente la manifestazione europea di tecnologia che ha avuto lo sviluppo più rapido: la prima edizione si è tenuta infatti a Ginevra nel 2004 con la partecipazione di 120 espositori e la presenza di 3.500 visitatori; da allora, in pochi anni, l'evento ha raggiunto più di 1.000 espositori e 60.000 visitatori. Una delle ragioni di

questo successo risiede nella presenza – in un unico luogo nello stesso periodo – di tutte le principali aziende del settore, le quali approfittano di un pubblico altamente qualificato per presentare le novità ed esporre i propri prodotti di punta. Bisogna poi considerare la crescente importanza di settori quali Smart Building & Residential Solutions, che si sono aggiunti ai tradizionali mercati del Professional Audio & Video e del Digital Signage.

Non bisogna dimenticare, infine, che la società che organizza l'ISE (Integrated Systems Events LLC) è una joint venture tra InfoComm International e CEDIA, le due associazioni leader a livello mondiale per l'industria audiovisiva. Un'industria, quella audiovisiva, che sempre di più sta convergendo verso il mondo dell'Information Technology, con strutture, standard e sistemi di trasmissione che sono gli stessi per i due settori. Forse anche in questo risiede la ragione del successo della manifestazione di Amsterdam.

"Nonostante l'allungamento della manifestazione a quattro giorni, abbiamo raggiunto il sold-out già da tempo. Ciò significa che anche per i nostri espositori e partner, questa è stata la decisione più corretta per espandere le opportunità di business di ISE" ha affermato Mike Blackman, Integrated Systems Events' Managing Director *"Siamo sicuri che questa decisione verrà accolta positivamente anche dai visitatori"* ha concluso. Come di consueto la giornata pre-fiera sarà dedicata alle conferenze di un giorno con, tra gli altri, l'AudioForum organizzato da Connessioni, il Digital Signage Summit e il Total Hospitality Technology Forum, dedicato all'innovazione tecnologica nell'hospitality (organizzato da Hotel Management International).

www.iseurope.org



■ 16-18 DICEMBRE 2015

SEMICON Japan 2015

Tokyo • Giappone



È uno dei più importanti eventi per l'industria giapponese dei semiconduttori, con una visione sull'intera catena produttiva che parte dai tools di progettazione per toccare le macchine e le tecniche di produzione dei wafer, fino al test, all'ispezione e al packaging. Tra le iniziative di quest'anno, una serie di nuovi padiglioni e conferenze che cercheranno di dare una più precisa visione del futuro, con l'Internet of Things in primo piano anche da queste parti. "World of IoT" si chiama l'iniziativa legata a questo tema, con i leader delle società di microelettronica intenti a confrontarsi su come trasformare in realtà questo paradigma.

Il Manufacturing Innovation Pavilion presenterà le aziende e le tecnologie all'avanguardia nella ricerca & sviluppo nel campo della microelettronica, mettendo in evidenza le innovazioni tecnologiche nei materiali, processi, metrologia, imballaggio e test.

Nel Sustainable Manufacturing Pavilion verranno analizzate le problematiche relative ad una crescita sostenibile del settore, con particolare riferimento alla capacità produttiva ed alla richiesta dei wafer da 200 mm di cui si prevede un aumento consistente della domanda.

Semicon Japan ospita anche padiglioni regionali giapponesi e di altre nazioni.

www.semiconjapan.org

■ 2-4 FEBBRAIO 2016

TUSEXPO 2016

L'Aia • Olanda

Sarà ancora l'Aia ad ospitare dal 2 al 4 Febbraio 2016 la seconda edizione di *The Unmanned Systems Expo (TUSEXPO)*, la fiera europea dedicata ai sistemi senza pilota e al relativo *World Forum*. TUSEXPO è focalizzata sul mercato in rapida crescita dei sistemi a guida autonoma e sulle applicazioni commerciali e civili in settori quali la sicurezza, l'agricoltura, l'industria, la ricerca scientifica, l'intrattenimento e la tutela dell'ambiente. Secondo gli organizzatori, a questa seconda edizione parteciperanno circa 2.500 visitatori internazionali, dopo che alla

prima edizione, nel febbraio del 2015, hanno partecipato oltre 1.500 visitatori e 75 giornalisti. La manifestazione prevede un'area espositiva e un'area convegnistica, oltre ad un'attività di matchmaking. Al centro delle conferenze, sia le più recenti tecnologie e le loro applicazioni, che gli aspetti normativi e legali legati all'utilizzo dei droni. L'area espositiva darà la possi-



**THE
UNMANNED
SYSTEMS
EXPO**
www.tusexpo.com

bilità ai visitatori di toccare con mano i più recenti prodotti di questo settore; sono previsti anche spazi per i centri di ricerca universitari e per le start-up che operano in questo ambito. La conferma come sede dell'evento della città dell'Aia è da mettere in relazione al cluster *The Hague Security Delta (HSD)*, riconosciuto a livello internazionale come il luogo ideale per sviluppare prodotti nel campo della sicurezza, con i sistemi a guida autonoma in grado di dare un forte impulso a questo settore.

www.tusexpo.com

MOSTRE MERCATO

FASANO (BR)

EXPO ELETTRONICA PUGLIA

Quartiere Conforama - SS16 uscita Savellieri Fasano BR - Organizzazione: ARI di BARI e di CASTELLANA GROTTE
Tel: 3476847524 - Fax: 080748931
info@aricastellana.it - www.aricastellana.it
07-08 Febbraio 2016

CASALE MONFERRATO (AL)

FIERA DELL'ELETTRONICA

Quartiere fieristico Casale Monferrato
Organizzazione: Open Office srl
Tel: 0308376078
info@fiereleletronica.it
www.fiereleletronica.it
07-08 Febbraio 2016

FIRENZE

ELETTRONICA FIRENZE

OBIHALL - Via Fabrizio De Andrè - Firenze
Organizzazione: Prometeo
Tel: 057122266 - Fax: 05710831142
info@prometeo.tv
www.prometeo.tv
07-08 Febbraio 2016

OSIMO (AN)

FIERA DELL'ELETTRONICA INFORMATICA E RADIOAMATORI

Grotte Center - Camerano AN
Organizzazione: Elettrotec - Tel: 071 8853021
info@elettronicaosimo.it
www.elettronicaosimo.it
07-08 Febbraio 2016

FERRARA

FERRARA MONDO ELETTRONICA

Quartiere Fieristico - Ferrara
Organizzazione: Expo Fiere Srl
Tel: 054527548 - Fax: 0545291179
info@mondoelettronica.net
www.mondoeleletronica.net
13-14 Febbraio 2016

CARRARA (MS)

EXPO ELETTRONICA

Quartiere fieristico - Carrara Fiere
Organizzazione: Blu Nautilus Srl
Tel: 0541439573 - Fax: 0541439584
info@expoeleletronica.it
www.expoeleletronica.it
13-14 Febbraio 2016

CENTOBUCCHI (AP)

33ª FIERA MERCATO DELL'ELETTRONICA

Centro Logistico Orlando Marconi
Organizzazione: Federvol
Tel: 3477433924 - Fax: 0734841316
federvol@gmail.com
www.federvol.it
20-21 Febbraio 2016

SCANDIANO (RE)

FIERA REGIONALE DELL'ELETTRONICA DI SCANDIANO

Centro Fieristico di Scandiano (RE)
Organizzazione: Comune di Scandiano
Tel: 0522764302 - Fax: 0522857592
entefiere@comune.scandiano.re.it
www.fierascandiano.it
20-21 Febbraio 2016

ROVIGO

F.I.E.R.A. - Fiera Informatica, Elettronica e Radiantismo

Centro Servizi Rovigo Fiere
Viale P. Adige 45
Organizzazione: Area Rebus
Tel: 042527402 - Fax: 042527065
fiera@arearebus.com - www.arearebus.com
27 e 28 Febbraio 2016

VICENZA

EXPO ELETTRONICA

Quartiere Fieristico di Vicenza
Organizzazione: Blu Nautilus Srl
Tel: 0541439573 - Fax: 0541439584
info@expoeleletronica.it
www.expoeleletronica.it
27 e 28 Febbraio 2016

LA SPEZIA

FIERA DELL'ELETTRONICA DI LA SPEZIA

Spezia Expo - La Spezia
Organizzazione: Prometeo
Tel: 057122266 - Fax: 05710831142
info@prometeo.tv - www.prometeo.tv
27 e 28 Febbraio 2016

L'elenco aggiornato di tutte le Mostre Mercato del 2015 è disponibile sul sito www.eleletronica.in sul quale è anche possibile scrivere un commento sulle fiere visitate.

L'elenco completo dei più importanti eventi nazionali e internazionali di elettronica, sicurezza e fonti rinnovabili è disponibile sul nostro sito: http://www.eleletronica.in/myMain/Fiere/p_fiere.asp



MLX73290-M, transceiver programmabile multi-canale sub-GHz

Melexis ha introdotto l'MLX73290-M, un transceiver RF multicanale pensato per operare nelle bande di frequenze ISM al di sotto del GHz comprese fra 300 MHz e 960 MHz; il prodotto fornisce due canali RF, ciascuno dotato di un trasmettitore RF e di un ricevitore RF ad elevata sensibilità. Il chip è totalmente programmabile attraverso la sua interfaccia periferica seriale (SPI).

Questo IC è particolarmente adatto per le applicazioni wireless al di sotto del GHz come l'automazione domestica/degli edifici, il monitoraggio della pressione dei pneumatici (TPMS), la lettura automatica delle utenze (AMR), i sistemi di allarme, i sistemi di accesso



remoto senza chiave (PKE), la diagnostica medica e la telemetria. Il suo livello di potenza in uscita varia fra -20dBm e 13dBm, mentre la sua sensibilità di ricezione può raggiungere -120 dBm con una ampiezza di banda di 15kHz. Con una velocità di trasmissione dati massima di 250kbps, il transceiver è

anche in grado di gestire velocità dati più elevate. Grazie all'alto grado di programmabilità, gli sviluppatori che fanno uso di questo IC dispongono di ampie possibilità quando si tratta di configurare e realizzare un'interfaccia di ingresso di tipo RF. È possibile regolare un'ampia varietà di parametri RF diversi (numero di canali, risoluzione di frequenza, uscita, deviazione della frequenza, ecc.) allo scopo di soddisfare criteri specifici di un'applicazione. L'MLX73290-M è fornito in un package QFN a 32 piedini da 5mm x 5mm caratterizzato da un intervallo di temperature operative compreso fra -40°C e 105°C.

www.melexis.com

Atmel presenta la versione Studio 7 del suo IDE



Atmel ha annunciato il rilascio di Atmel Studio 7, l'ambiente di sviluppo integrato (IDE) e gratuito della società per la progettazione del codice dei processori ATMEGA SMART basati su tecnologia ARM. Atmel ha anche lanciato Atmel START, una nuova e intuitiva piattaforma grafica per creare e configurare applicazioni embedded che consentono agli sviluppatori di creare piattaforme

software personalizzate. Con l'incremento della complessità e l'aumento delle funzionalità richieste, gli sviluppatori embedded devono utilizzare sistemi di sviluppo più performanti ma allo stesso tempo più facili da usare. Basato sull'ultima versione di Microsoft Visual Studio Shell, l'IDE Atmel Studio 7 riduce drasticamente il tempo complessivo di progettazione, fornendo miglioramenti significativi delle prestazioni per lo sviluppo e il debug con un'interfaccia utente facile da usare; il tutto al fine di fornire una più rapida risposta alle esigenze del mercato consumer, industriale, dei Maker e molti altri ancora.

www.atmel.com

SB0400 e SB0401, IC antibloccaggio per sistemi di frenatura di moto e scooter

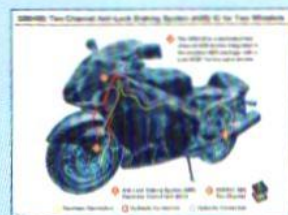
Freescale Semiconductor ha presentato i nuovi circuiti integrati SB0400 e SB0401 specificamente studiati per moto e scooter. Con questi minuscoli dispositivi (appena 7x7 mm) – da integrare nella centralina elettronica della moto (ECU) – non sarà più necessario fare ricorso ai molto più grandi circuiti integrati ABS sviluppati per le automobili, realizzando notevoli risparmi in termini di costi, peso e spazio.

Il nuovo IC SB0400 è adatto per motociclette con ABS sulle due ruote mentre il modello SB0401 è destinato a sistemi monoruota ed in genere a scooter e piccole moto.

I nuovi dispositivi utilizzano

la tecnologia analog-mixed SMARTMOS di Freescale, ed integrano un'interfaccia per il sensore di velocità, i driver per le valvole e la pompa motore e i relativi sistemi di sicurezza; il tutto nel più piccolo package disponibile attualmente sul mercato. Inoltre, entrambi i chip non necessitano di dissipatore, con un ulteriore risparmio di costi e spazio.

www.freescale.com



VersaClock 6, i generatori di clock di IDT con prestazioni al top

IDT ha presentato la famiglia di generatori di clock programmabili VersaClock 6, che offre una soluzione flessibile e a basso consumo per affrontare le più impegnative applicazioni ad alte prestazioni. Con un jitter di fase inferiore a 500 femtosecondi (fs), i generatori di clock VersaClock 6 offrono le migliori prestazioni mai raggiunte dalla famiglia VersaClock di IDT, grazie a una combinazione eccezionale di

caratteristiche di jitter, flessibilità e basso assorbimento. Con un jitter di fase RMS inferiore a 500 fs sull'intero intervallo di integrazione da 12 kHz a 20



MHz, i nuovi dispositivi soddisfano i requisiti più stringenti in termini di jitter e rumore di fase imposti dalle applicazioni conformi a numerosi standard, come 10G Ethernet, sistemi di storage SAS e SATA, PCI Express Gen 1/2/3, XAUI, SRI, nonché quelli relativi alle interfacce PHY delle FPGA di fascia alta di ultima generazione, il tutto con una dissipazione di potenza dimezzata rispetto a quella dei dispositivi

concorrenti. L'assorbimento di corrente di base di questi dispositivi, pari a 30 mA, semplifica il rispetto dei requisiti termici e riduce la dissipazione di potenza del sistema. Inoltre, i dispositivi VersaClock 6 hanno un formato compatibile con quello delle rispettive controparti della famiglia VersaClock 5, semplificando le realizzazioni di soluzioni scalabili con modifiche progettuali minime.

www.idt.com

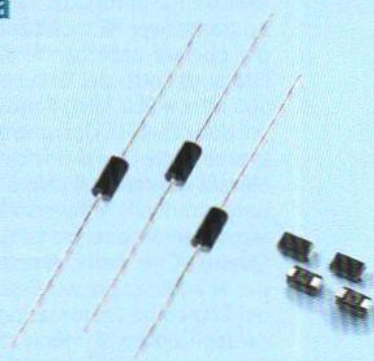
Tiristori di commutazione SIDAC per applicazioni a scarica capacitiva

Littelfuse ha presentato due nuove categorie di prodotti: le serie Kxxx0S1URP e Kxxx0GURP di tiristori di commutazione SIDAC unidirezionali. Un SIDAC (abbreviazione di Silicon Diode for Alternating Current, cioè diodo al silicio per corrente alternata) è un interruttore a semiconduttore utilizzato in combinazione con un condensatore

e un trasformatore in circuiti di accensione a scarica capacitiva (CDI) per la generazione di picchi di alta tensione. Le nuove serie offrono minori perdite di commutazione rispetto ai prodotti SIDAC esistenti standard e ad alta energia. La nuova progettazione di SIDAC offre una maggiore corrente a impulso ripetitivo (ITRM) a temperature

ambiente superiori, una migliore tensione di breakover/innesco rispetto alle caratteristiche di temperatura e minore perdita di commutazione. Questo consente ai circuiti esistenti di supportare uscite di tensione superiori e garantire uscite di tensione stabili a temperature ambiente elevate.

www.littelfuse.com



MCU a 32 bit con unità di calcolo in virgola mobile da Microchip

Microchip Technology ha annunciato una nuova serie di prodotti della famiglia di microcontrollori a 32 bit PIC32MZ che implementa un'unità hardware di calcolo in virgola mobile (FPU) ad alte prestazioni e bassa latenza, adatta per applicazioni intensive di calcolo a singola e doppia precisione.

La nuova serie PIC32MZ EF conta 48 differenti modelli e dispone anche di un convertitore analogico-digitale (ADC) a 12 bit e 18 Msps, adatto ad una vasta gamma di applicazioni a banda larga ed alta velocità. Inoltre, il PIC32MZ EF supporta un vasto set di istruzioni per DSP. La combinazione di una FPU a precisione doppia, un set di istruzioni DSP e un ADC ad elevata velocità contribuisce a migliorare la densità del codice, diminuisce la latenza e accelera le prestazioni in applicazioni ad alta intensità.

La serie PIC32MZ EF utilizza un core Imagination Technologies



MIPS M-Class con performance di 330 DMIPS (a 200 MHz) e 3,28 CoreMarks /MHz, insieme ad una memoria Flash da 2 MB max di tipo dual-panel che consente un aggiornamento sul campo, una RAM da 512 KB, nonché una nutrita serie di periferiche per la connettività, tra cui Ethernet 10/100 MAC, USB MAC/PHY ad alta velocità e due porte CAN. Come optional è anche disponibile un avanzato motore crittografico hardware ricco di funzionalità, con un generatore di numeri casuali a supporto di un elevato flusso di dati da autenticare, crittografare e decrittare negli standard AES, 3DES, SHA, MD5 e HMAC.

www.microchip.com

Telemaco2, processori dedicati alla telematica per veicoli

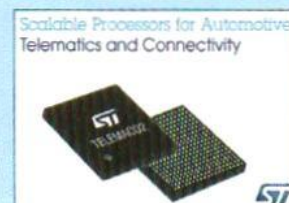
STMicroelectronics ha introdotto una nuova famiglia di processori per applicazioni specifiche di telematica e di connettività per impiego automotive. La nuova famiglia Telemaco2 offre una serie di microcontrollori programmati basati su Linux, scalabili e di dimensioni ridotte, dedicati ai sistemi in-car che migliorano la sicurezza, il comfort e i consumi. I sistemi basati su Telemaco2 sono in grado di effettuare chiamate automatiche in caso di incidente, garantire connettività continua vettura-cloud con diagnostica auto a distanza, nonché offrire indicazioni stradali in tempo reale.

La telematica è destinata a ricoprire un ruolo sempre più importante in tutti i veicoli, da quelli privati alle flotte aziendali e ai trasporti pubblici. Questa tendenza è anche sostenuta da importanti politiche governative, come

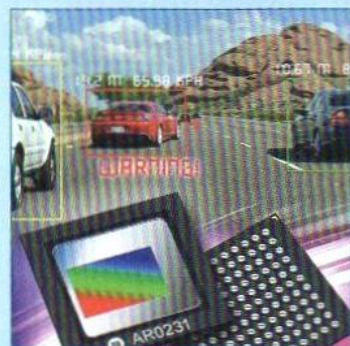
la recente norma della Comunità Europea che impone per tutte le nuove automobili vendute in Europa da aprile 2018 la tecnologia "eCall", che consente alla vettura di chiamare automaticamente i servizi di emergenza in caso di incidente e di fornire importanti informazioni come l'esatta posizione del veicolo e la direzione di marcia.

Fino ad oggi, questo settore della telematica ha utilizzato processori che non sono stati specificamente progettati per questo scopo, con funzionalità non necessarie e spreco di silicio.

www.st.com



AR0231AT, sensore di immagini di nuova generazione per automotive



ON Semiconductor ha annunciato la disponibilità di un sensore di immagini CMOS da 2,3 megapixel (MP) dotato della rivoluzionaria tecnologia LFM (LED Flicker Mitigation), che rappresenta un nuovo punto di riferimento per la realizzazione di sistemi di assistenza alla guida avanzati (ADAS) per auto. In grado di catturare immagini video ad elevata dinamica (HDR) in formato 1080p, il sensore

AR0231AT supporta la sicurezza funzionale delle applicazioni automobilistiche secondo le norme ASIL B (Automotive Safety Integrity Level B). La tecnologia LFM, un brevetto in corso di registrazione, elimina il tipico sfarfallio ad alta frequenza (flicker) proveniente dalla luce dei semafori a LED e dai fari a LED dei veicoli, onde garantire il corretto funzionamento degli algoritmi di iden-

tificazione dei segnali stradali in ogni condizione di illuminazione. Il sensore AR0231AT ha un formato ottico di 1/2,7 pollici (6,82 mm) e una matrice attiva di pixel in formato 1928 (orizzontale) x 1208 (verticale). Il sensore AR0231AT viene offerto in un contenitore iBGA-121 da 11 mm x 10 mm e i primi campioni di valutazione sono già disponibili.

www.onsemi.com

Nuova famiglia di processori AM57X Sitara di Texas Instruments

Per offrire agli sviluppatori integrazione, scalabilità e periferiche in un solo chip, Texas Instruments ha presentato la famiglia di processori AM57X Sitara, che comprende i dispositivi con le prestazioni più elevate nell'ambito di questa piattaforma di processori. I processori Sitara AM57X sono progettati per un'ampia gamma di applicazioni embedded e industriali grazie all'esclusiva architettura eterogenea con core ARM Cortex-A15 per calcolo ad alte prestazioni e al sistema operativo di alto livello (HLOS). Inoltre, i processori AM57X integrano processori di segnale digitali C66x (DSP) di TI per analisi e calcolo in tempo reale, unità in tempo reale programmabili (PRU) e core ARM Cortex-M4 per le funzioni di controllo, oltre ad acceleratori video e grafici per interfacce



utente avanzate e applicazioni multimediali, posizionando i processori AM57X ai vertici della loro fascia di mercato. La nuova famiglia di processori Sitara AM57X è stata concepita fin dall'inizio puntando su prestazioni elevate e massima integrazione. Il processore offre così agli sviluppatori prestazioni impareggiabili che superano di oltre il 40 per cento quelle dei processori ARM Cortex-A9 quad-core e del 280 per cento quelle dei normali processori ARM Cortex-A9 dual-core attualmente diffusi sul mercato embedded. I processori Sitara AM57X offrono il massimo livello di integrazione attualmente disponibile in commercio per funzionalità di calcolo, controllo in tempo reale, connettività e multimedialità. www.ti.com

Tre nuovi microcontrollori ottimizzati per USB

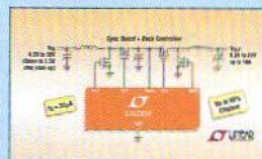
Toshiba Electronics Europe ha presentato tre nuovi microcontrollori ottimizzati per essere utilizzati in prodotti con interfaccia USB. I chip TPM066FWUG, TPM067FWQG e TPM068FWXBG sono gli ultimi modelli aggiunti alla serie di dispositivi Toshiba TX00 basati su core ARM Cortex-M0 e integrano al loro interno un controller USB. Questi microcontrollori hanno al loro interno 128 kB di memoria Flash e 16 kB di SRAM e sono stati progettati per essere utilizzati in svariate applicazioni, come dispositivi mobili e tablet, PC portatili, attrezzature per l'ufficio, sistemi industriali e concentratori per sensori (sensor hub). I nuovi chip supportano diverse interfacce

seriali come USB, SPI (Serial Peripheral Interface) e I2C in modalità Fast Mode Plus con velocità di trasmissione fino a 1 Mbps. I controller USB incorporati permettono di semplificare il collegamento e la gestione dei dispositivi USB. Possono anche funzionare come dei "sensor hub", trasferendo le informazioni fornite dai sensori a un controller principale o a un personal computer. www.toshiba.semicon-storage.com



LTC7812, controller DC/DC sincrono boost + buck

Linear Technology presenta l'LTC7812, un controller sincrono DC/DC a bassa corrente di quiescenza e a due uscite (boost + buck). Quando collegati in cascata, i controller indipendenti step-up (boost) e step-down (buck) regolano la tensione di uscita da una tensione di ingresso che può essere superiore, inferiore o uguale alla tensione di uscita, mantenendo la regolazione delle uscite in condizioni di avviamento a freddo e load dump. A differenza dei regolatori buck-boost convenzionali a singolo induttore, la soluzione boost + buck in cascata dell'LTC7812 fornisce una rapida risposta ai transienti con correnti di ingresso e di uscita continue con pulsanti



che sostanzialmente riducono la tensione di ripple e le interferenze elettromagnetiche (EMI). L'LTC7812 funziona da una tensione di ingresso compresa tra 4,5V e 38V; dopo l'accensione la tensione viene mantenuta a soli 2,5V. Il convertitore boost sincrono può generare tensioni di uscita fino a 60V e può funzionare con un duty cycle dello 0% (switch sincrono ON) per filtrare la tensione di ingresso quando richiesto per ottimizzare l'efficienza. www.linear.com

RZ/G, processori ad alte prestazioni grafiche

Renesas Electronics ha annunciato la disponibilità della serie di microprocessori RZ/G che consentono di sviluppare in modo estremamente rapido applicazioni ad alte prestazioni basate su sistemi operativi quali Android e Linux. La famiglia di dispositivi RZ/G è stata progettata per gestire l'interfaccia grafica utente di applicazioni industriali, per la domotica e per gli elettrodomestici, ed espande le caratteristiche tecniche dell'ormai ampiamente diffusa famiglia RZ/A (che supporta la grafica 2D fino alla risoluzione WXGA) allo scopo di supportare applicazioni che richiedono prestazioni più elevate quali l'elaborazione delle immagini, il riconoscimento delle immagini, la grafica in 3D e le funzioni multimediali fino alla risoluzione di 1080p.

La serie RZ/G è composta da due versioni principali: la versione RZ/G1E che include una coppia di CPU ARM Cortex-A7



che operano fino a 1,0 GHz e la versione RZ/G1M che include una coppia di CPU Cortex-A15 che operano fino a 1,5 GHz. Questa ampia gamma di dispositivi consente ai progettisti di selezionare la migliore combinazione tecnica per ottenere i migliori risultati in termini di prestazione oppure in termini di ottimizzazione dei consumi operativi. Ogni dispositivo della famiglia RZ/G include il supporto per la gestione del BUS esterno a 800 MHz DDR3, questo consente di avere un ampio margine di prestazione per applicazioni complesse quali l'elaborazione delle immagini di video streaming multipli e per il riconoscimento delle immagini evoluto quale, ad esempio, il riconoscimento facciale. www.renesas.eu

SMARTEM, IL TERMOSTATO INTELLIGENTE



Dopo avervi descritto il nuovo modulo EM1001 Tibbo e il suo impiego in un termostato, vi descriveremo le modalità di upload dei dati su un server esterno e la trasmissione WiFi. Terza ed ultima puntata.

di SAVINO GIUSTO

Nelle precedenti puntate abbiamo approfondito la conoscenza del nuovo modulo EM1001 di casa Tibbo (prima puntata) e posto le basi per la realizzazione di un sensore di temperatura smart, entrando nel dettaglio delle tecniche di programmazione per l'interfacciamento con il sensore di temperatura DS18B20 (seconda puntata). In questa terza puntata vi mostreremo come sia possibile condividere tali valori di temperatura tramite una piattaforma web. Avremmo potuto implementare un Web Server all'interno del nostro modulo EM1001, ma questo avrebbe posto alcuni problemi come la gestione di un indirizzo IP dinamico,

l'implementazione di un database per il salvataggio dei valori di temperatura e relative soglie, la gestione della sincronizzazione oraria. Per ovviare a tutte queste problematiche e rendere il progetto più semplice, ma al contempo ben strutturato, abbiamo deciso di "appoggiarci" a un server esterno verso cui inviare tutti i dati. Qualsiasi dispositivo che voglia accedere ai dati lo potrà fare semplicemente con una connessione Internet puntando verso il server esterno. Esistono diverse alternative, ma abbiamo selezionato per voi ThinkSpeak, perché è una soluzione di tipo open source (il codice sorgente è disponibile

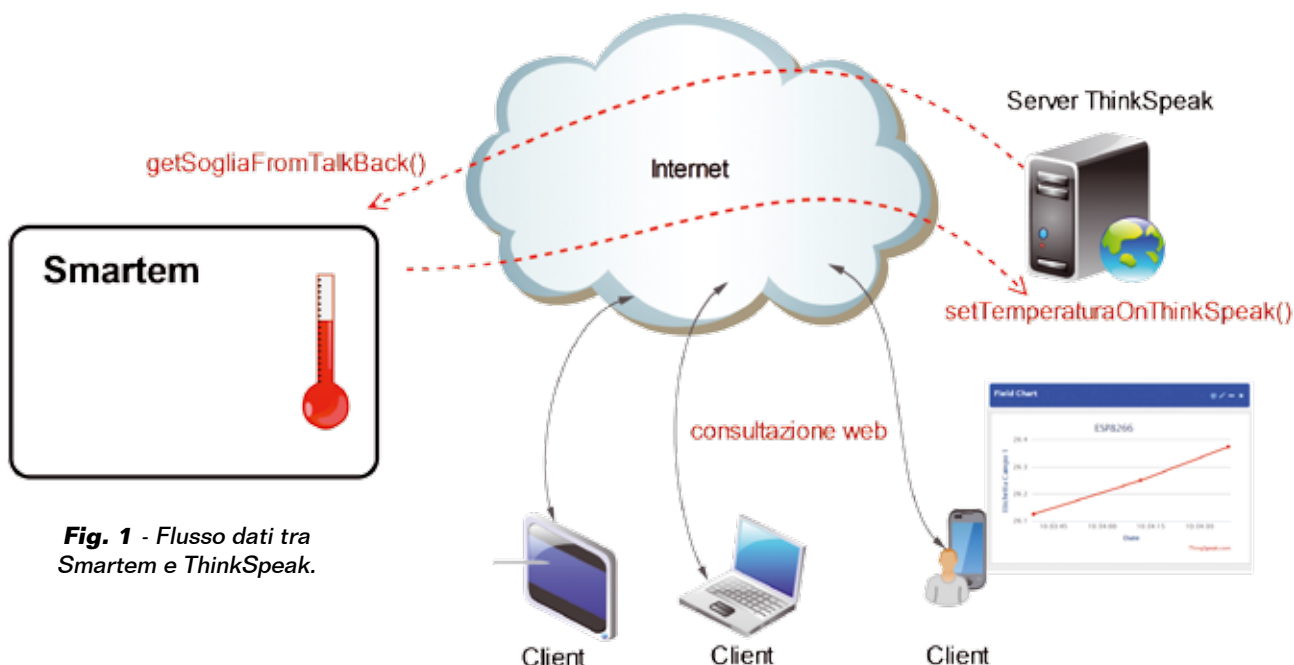


Fig. 1 - Flusso dati tra Smartem e ThinkSpeak.

per gli sviluppatori al seguente link <https://github.com/iobridge/ThingSpeak>). La Fig. 1 illustra quanto andremo a realizzare; come si può notare, sono state implementate all'interno di Smartem due funzioni:

1. *getSogliaFromTalkBack()*, la quale ha il compito di interrogare periodicamente il server e ottenere il valore di soglia impostato dall'utente che provocherà l'attivazione/disattivazione del sistema di riscaldamento;
2. *setTemperaturaOnThinkSpeak()*, la quale ha il compito di effettuare il caricamento sul server dei valori di temperatura.

Ogni dispositivo mobile o PC potrà accedere a tali dati tramite una rappresentazione grafica che mostra in real-time l'andamento della temperatura. La modalità con cui trasmetteremo i dati sarà di tipo wireless ed in particolare WiFi. Esistono differenti soluzioni in commercio e la stessa Tibbo ha una sua particolare proposta (cioè il modulo GA1000). Però abbiamo voluto presentarvi l'uso di un altro modulo (ESP-01) per

mostrare come sia semplice integrare EM1001 con altre tecnologie e produttori.

Per questo progetto ESP-01 e ThinkSpeak sono strettamente interconnessi; perciò abbiamo scelto di presentarvi nella prima parte dell'articolo il funzionamento del modulo WiFi, nella seconda parte la piattaforma ThinkSpeak ed infine nell'ultima parte il modo in cui essi, utilizzati insieme, riescono a rendere il nostro progetto "Smart".

IL MODULO ESP-01

Il modulo (Fig. 2) è basato sull'integrato ESP8266 e da qualche tempo viene molto utilizzato in virtù dell'elevato rapporto prestazioni/prezzo. L'ESP8266 è un chip che soddisfa pienamente le principali richieste del mondo IoT, vale a dire:

1. elevata integrazione (appena 18mm x 13mm);
2. costo ridotto;
3. connessione senza fili (di tipo WiFi);
4. elevata flessibilità di programmazione; infatti, il modulo può essere utilizzato sia come periferica per un microcon-

trollore esterno (che utilizzeremo in questo articolo) oppure in modalità stand-alone, poiché esso ha la possibilità di essere programmato.

La shield che abbiamo realizzato e descritto nella seconda puntata prevede anche il socket per inserire tale componente. La Fig. 3 riporta la pin-out, mentre la connessione con il modulo EM1001 è visualizzata nello schema elettrico pubblicato nella seconda puntata, cui rimandiamo (fascicolo n° 200); come si può constatare, si tratta di una interfaccia UART con livelli 3,3 V, quindi compatibili con Tibbo. Le istruzioni impartite al modulo ESP-01 sono inviate sotto forma di

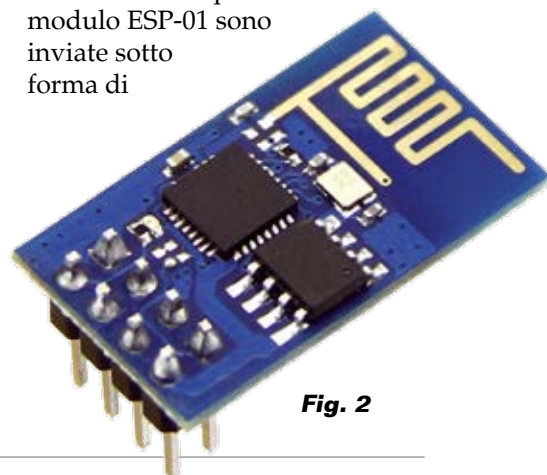


Fig. 2

comandi AT Hayes, cioè i tipici comandi usati in passato sui modem telefonici. La **Tabella 1** riporta i principali comandi AT che sono anche quelli che abbiamo usato per la sua programmazione. Vediamo in dettaglio il loro utilizzo. Ogni comando deve sempre essere seguito dai caratteri di Carriage return (codice ASCII 13) e Line feed (codice ASCII 10) affinché venga effettivamente interpretato. Per accertarsi che il modulo risponda correttamente è sufficiente usare AT; se tutto è corretto, il modulo ci risponderà con OK. Il comando `AT+CIPMUX=1` permette di configurare il dispositivo connessioni IP multiple. Il terzo comando che ci servirà utilizzare è `AT+CWJAP` necessario per collegare il modulo al nostro access point. La sintassi corretta prevede:

```
AT+CWJAP="<access_point_name>","<password>"
```

dove `access_point_name` deve essere sostituito con l'identificativo SSID della nostra rete WiFi, mentre il campo `password` ha un ovvio significato. Esso deve essere eseguito all'avvio del dispositivo. Prima di poter effettuare qualsiasi comunicazione con il server scelto, è necessario creare la relativa connessione; questo viene fatto tramite il seguente comando:

```
AT+CIPSTART=<id>,<type>,<addr>,<port>
```

dove `id` indica l'identificativo della connessione, `type` il tipo di protocollo UDP o TCP, `addr` l'indirizzo IP del server a cui connettersi ed infine `port` la porta tramite la quale stabilire la comunicazione. Ad esempio, per poter effettuare un collegamento

alla pagina web di Google, sarà necessario inviare al modulo ESP8266 la seguente istruzione:

```
AT+CIPSTART=4,"TCP","google.com",80
```

Dopo aver stabilito la connessione con il server, il comando che più utilizzeremo è sicuramente quello che predispone il modulo a ricevere comandi per interagire con il server:

```
AT+CIPSEND=<id>,<length>
```

dove `id` rappresenta l'identificativo della connessione selezionata, mentre `length` indica la lunghezza complessiva dei caratteri da inviare per un massimo di 2.048 caratteri. Per esempio, se vogliamo richiedere la pagina web `www.google.it`, dovremo usare il seguente comando:

```
AT+CIPSEND=4,18
```

Il numero 18 non è casuale, ma rappresenta il numero di caratteri presenti nel comando HTTP per richiedere una pagina web e cioè: `GET / HTTP/1.0\r\n\r\n(\r: carriage return; \n: line feed)`. Esso è proprio il successivo comando che dobbiamo inviare. A questo punto il modulo ci risponderà con:

```
+IPD,4,530:
HTTP/1.0 302 Found
Cache-Control: private
Content-Type: ...
```

Come in tutte le cose, alla teoria dobbiamo associare sempre la pratica. È questo il motivo per cui abbiamo realizzato un firmware di test che consente di inviare i comandi AT al modulo ESP-01 tramite lo stesso Smarterm che abbiamo presentato nella precedente puntata. Il codice (lo vedete nel **Listato 1**) rende il dispositivo un converti-

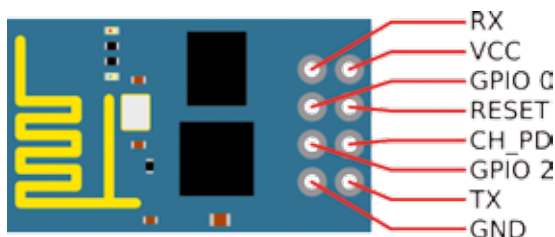


Fig. 3
Connessioni del
modulo ESP-01.

Tabella 1 - Principali comandi ESP8266 utilizzati.

Funzione	Comando
Corretto funzionamento	AT
Connessioni TCP/UDP	AT+ CIPMUX? AT+ CIPMUX=0 AT+ CIPMUX=1
Collegamento access point	AT+CWJAP? AT+CWJAP="SSID", "Password"
Crea una connessione con il server	AT+CIPSTART= <id>,<type>,<addr>,<port>
Invio dati TCP/IP	AT+CIPSEND=<id>,<length>
Chiusura connessione TCP/UDP	AT+CIPCLOSE=<id> o AT+CIPCLOSE
Riavvio del modulo	AT+RST
	AT+CI0BAUD? Supported: 9600, 19200, 38400, 74880, 115200, 230400, 460800, 921600

Listato 1 - Codice di prova per il modulo ESP8266 e EM1001

```

Osub on_sys_init
  dim w as word
  net.ip="10.218.136.155"  '«««««««« modificare con un IP compabile con la propria sottorete
  sock.num=0
  sock.rxbufreq(4)        'we ask for 4*256-16=1008 bytes for the RX buffer of socket 0
  sock.txbufreq(4)        'we ask for 4*256-16=1008 bytes for the TX buffer of socket 0

  sock.num=0
  ser.rxbufreq(4)          'we ask for 4*256-16=1008 bytes for the RX buffer of the serial port
  ser.txbufreq(4)          'we ask for 4*256-16=1008 bytes for the TX buffer of the serial port

  sys.buffalloc            'And now ACTUAL buffer memory allocation
  sock.num=0
  sock.protocol=PL_SOCKET_PROTOCOL_TCP      'We set TCP
  sock.localportlist="2000,3000"            '«««««««« Listening ports. CHANGE AS NEEDED!
  sock.inconmode=PL_SOCKET_INCONMODE_ANY_IP_ANY_PORT  'Accept connection from any host

  ser.num=0
  ser.baudrate=ser.div9600                  'We want to run at 9600bps
  ser.enabled=YES                          'The serial port is now ON!
end sub

sub on_sock_data_arrival
  dim rx_buffer as string
  rx_buffer = sock.getdata(ser.txfree)
  ser.setdata(rx_buffer + chr(13) + chr(10))
  ser.send
end if
end sub

sub on_ser_data_arrival
  dim rx_buffer as string
  rx_buffer = ser.getdata(sock.txfree)
  sock.setdata(rx_buffer)                  'Rimbalza sul socket
  sock.send
end sub

```

tore seriale/ethernet, permettendo l'invio manuale di comandi tramite EM1001 e il modulo basato sull'ESP8266. Per l'invio manuale dei comandi possiamo far riferimento alla procedura descritta nel paragrafo "DEBUGGING TRAMITE SOFTWARE IO NINJA" della prima puntata. In questo modo avrete la possibilità di prendere dimestichezza con il modulo e provare i comandi appena descritti.

LA PIATTAFORMA THINKSPEAK

ThinkSpeak è una piattaforma applicativa appositamente sviluppata per semplificare lo sviluppo di dispositivi IoT. ThinkSpeak permette di realizzare un'applicazione tramite i dati raccolti da uno o più sensori dislocati in posizioni differenti. Le principali caratteristiche che lo contraddistinguono sono:

1. raccolta dati in real-time;
2. elaborazione dati;
3. visualizzazione;
4. applicazioni e plug-in.

Alla base di ThinkSpeak ci sono i cosiddetti canali ("Channel"); un canale è dove vengono inviati i dati da memorizzare. Ogni canale è caratterizzato da 8 campi per i dati, 3 campi per la geolocalizzazione del sensore ed un campo di stato. Maggiori approfondimenti possono essere recuperati dalla guida online disponibile a questo indirizzo <https://thingspeak.com/docs>, dove potete anche visualizzare una dimostrazione live di un sensore di luce che pubblica dati su ThinkSpeak; notate che il grafico viene aggiornato in tempo reale ogni 10 secondi: questo è proprio quello che realizzeremo con il nostro

Smartem.

Ma prima di poter salvare dati sul canale è necessario registrarsi e crearne uno; ciò si effettua con la procedura di seguito descritta.

1. Andate all'indirizzo <https://thingspeak.com/> e fate clic su *Registrati*; sarà mostrata la schermata in **Fig. 5**. Immettete i dati richiesti e fate clic sul pulsante *Crea Account*.
2. Selezionate il pulsante *New Channel*; ci saranno mostrati i campi da riempire relativi al canale che vogliamo creare (**Fig. 6**). Tra i principali possiamo citare:
 - *Nome*, identifica il canale;
 - *Descrizione*, un breve descrizione dei dati relativi al sensore;
 - *Campo 1...8*, rappresentano i valori da visualizzare nel grafico (nel nostro caso particolare selezioniamo

- solo il primo campo che riporterà la temperatura);
- *Rendi Pubblico?*, permette di rendere pubblico il canale; in caso tale campo non sia spuntato il canale potrà essere visualizzato solo da chi possiede le credenziali per l'accesso.

A questo punto il canale risulta creato ed è già predisposto a ricevere e memorizzare i dati inviati (Fig. 7).

Ora vi starete chiedendo che misure sono messe in atto per garantire la sicurezza dei dati; anche da questo punto di vista la piattaforma risulta ben strutturata, poiché la scrittura dei dati è possibile solo se si possiede un'apposita chiave alfanumerica che è possibile recuperare selezionando il tab *Chiavi API*. Quella che ci interessa è la *chiave API di Scrittura*. La Fig. 8 illustra quanto descritto. Qualora tale chiave di sicurezza dovesse essere compromessa, sarà sufficiente fare clic sul pulsante *Genera Nuova Chiave di Scrittura* per generarne un'altra in automatico.

Memorizzare un valore su ThinkSpeak, a questo punto, è davvero molto semplice: è sufficiente richiamare il link https://api.thingspeak.com/update?api_key=YOUR_CHANNEL_API_KEY&field1=VAL e una volta ottenuto l'accesso, al posto di YOUR_CHANNEL_API_KEY dobbiamo inserire il valore della chiave di cui al punto precedente, mentre VAL deve essere sostituito con il valore numerico che vogliamo memorizzare (la temperatura nel nostro caso). Per effetto di ciò la stringa diventerà https://api.thingspeak.com/update?api_key=VZYA0JAFH0WNUPO8&field1=7. Se immettiamo questo indirizzo in un normale browser, ci verrà

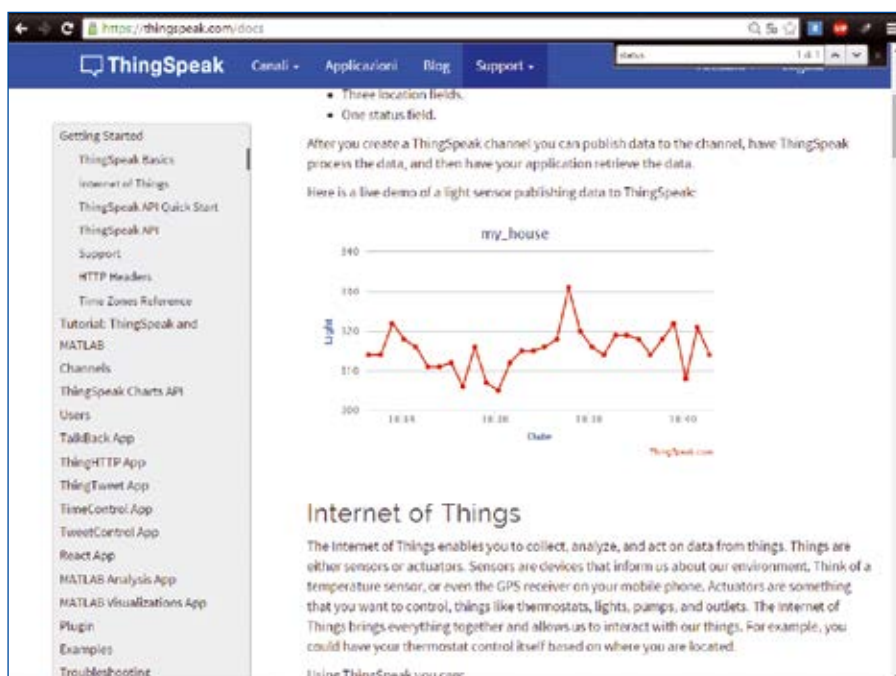


Fig. 4 - Guida online di ThinkSpeak (<https://thingspeak.com/docs>) ed esempio live di upload dei dati su un canale ThinkSpeak.

The screenshot shows the 'Registrati per iniziare ad utilizzare ThingSpeak' (Sign up to start using ThingSpeak) form. It includes input fields for 'ID Utente', 'Email', 'Fuso orario' (Time zone), 'Password', and 'Conferma Password'. There is a checkbox for 'Con la registrazione, l'utente accetta i Termini di Servizio e Privacy Policy' and a green 'Crea Account' button.

Fig. 5 - Schermata di registrazione a ThinkSpeak.

The screenshot shows the 'New Channel' creation page on ThingSpeak. It has a 'Name' field (filled with 'Smartem') and a 'Descrizione' field (filled with 'valore di temperatura rilevato da sensore'). Below are eight 'Campo' (Field) input boxes. To the right is an 'Aiuto' (Help) section titled 'ThingSpeak Channel' and 'Channel Settings' with detailed instructions on how to configure the channel, including fields for location, metadata, and tags.

Fig. 6 - Pagina per l'inserimento dei dati relativi al canale da creare.

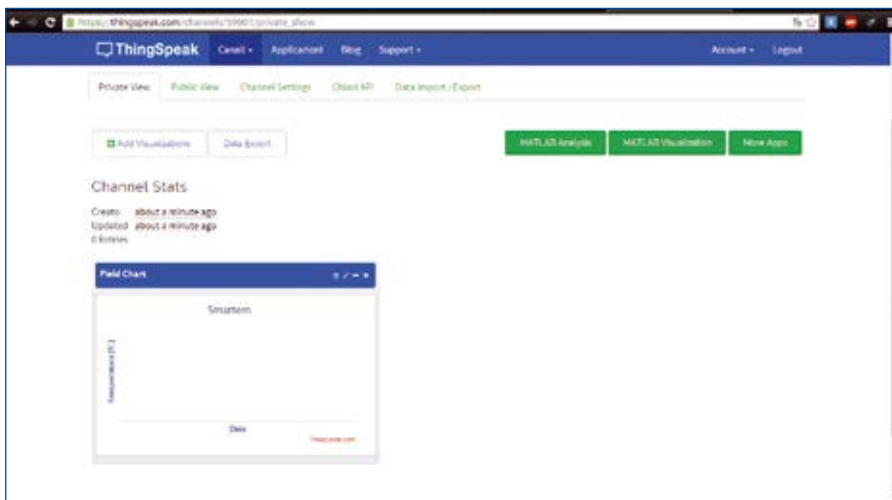


Fig. 7 - Cruscotto per il controllo e la configurazione del canale ThinkSpeak.

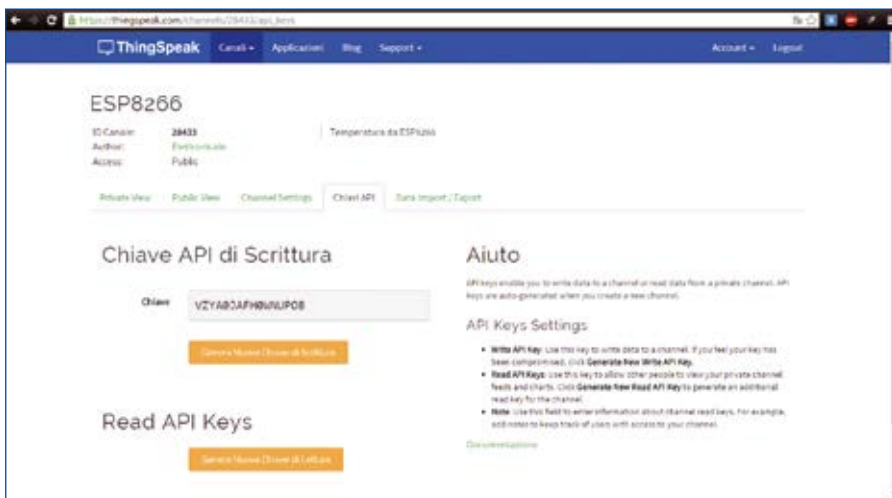


Fig. 8 - Pagina per la generazione e visualizzazione della chiave di scrittura.

restituito il valore 1 ad indicare che la memorizzazione è andata a buon fine e nella visualizzazione del grafico del canale com-

parirà in automatico un punto rosso, indicante il valore e il timestamp (Fig. 9). Il medesimo metodo dovrà essere seguito da

Fig. 9
Il punto rosso rappresenta il primo dato caricato sulla piattaforma ThinkSpeak.



qualsiasi dispositivo (che sia un PC, smartphone o modulo embedded) per caricare a intervalli regolari i valori acquisiti. Comincia quindi ad essere chiaro quanto l'uso della piattaforma ThinkSpeak ci abbia semplificato il compito della programmazione, infatti ci siamo risparmiati di sviluppare:

- il web server per la consultazione dei dati;
- il database per la loro memorizzazione;
- la gestione della sincronizzazione oraria.

Fino a questo punto abbiamo visto come inviare un dato dal nostro sensore al server ThinkSpeak. Potremmo però avere necessità di effettuare l'operazione opposta: pensiamo ad esempio al caso del nostro sensore di temperatura e alla necessità di dover gestire una soglia di intervento del termostato. Ovviamente, qualcuno potrebbe obiettare che tale valore dovrebbe essere memorizzato all'interno del modulo embedded, però questo pone un duplice problema:

1. necessità di una pagina web o un'interfaccia seriale per programmare la soglia;
2. necessità di avere accesso diretto al dispositivo, con tutte le complicazioni legate alla gestione di DNS dinamici.

Per ovviare a queste problematiche ThinkSpeak mette a disposizione un'applicazione denominata TalkBack (significa letteralmente "rispondere" poiché permette ad un qualsiasi dispositivo di interrogare il server ThinkSpeak chiedendogli il valore di un parametro). Per implementare questo meccanismo è necessario creare una TalkBack seguendo questa procedura:

1. selezionare *Applicazioni* →

- TalkBack* dall'interfaccia web;
2. selezionare il pulsante *New TalkBack*;
 3. nella nuova pagina che si apre bisogna:
 - scrivere il nome della *TalkBack*;
 - riportare in Log to Channel il canale precedentemente creato;
 - fare clic sul pulsante *Add Command* e quindi inserire in *Position* il valore 1 e in *Command String* il valore (ad esempio quello di soglia) che il nostro dispositivo embedded dovrà leggere.

La pagina configurata apparirà come in **Fig. 10**. Con lo stesso metodo usato per caricare un valore, basta usare il seguente indirizzo per interrogare ThinkSpeak:

https://api.thingspeak.com/talkbacks/TALKBACK_ID/commands/COMMAND_ID?api_key=API_KEY

dove TALKBACK_ID rappresenta l'identificativo della talkback, COMMAND_ID l'identificativo e API_KEY è la chiave API generata in automatico da ThinkSpeak. Tutte queste informazioni sono recuperabili dalla pagina di configurazione della talkback (come in **Fig. 10**).

INTERFACCIARE SMARTEM CON THINKSPEAK

Ora che abbiamo compreso le potenzialità di ThinkSpeak in merito alle applicazioni IoT e il modo di pilotare il modulo WiFi ESP-01, siamo pronti a combinare entrambe le funzionalità per rendere operativo Smartem. Ci concentreremo sulle modalità di programmazione all'interno dell'ambiente TIDE. Riprendendo il diagramma di flusso riportato nella precedente puntata in **Fig. 2**, possiamo vedere che la

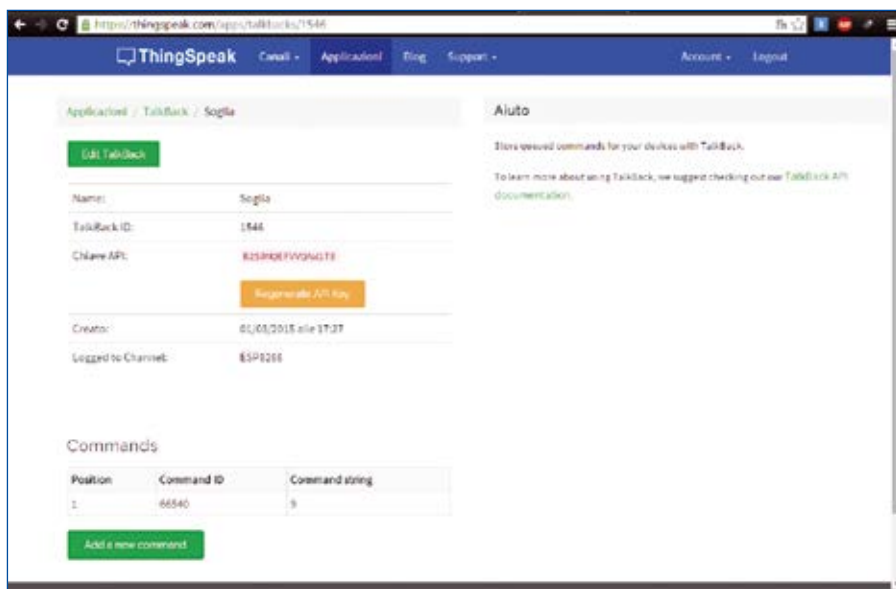


Fig. 10

funzione ON_SYS_TIMER richiama periodicamente tre funzioni principali:

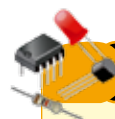
1. *getTemperaturaFromDS18B20()*, già analizzata nella seconda puntata;
2. *getSogliaFromTalkBack()*, riportata nel **Listato 1**, serve per recuperare il valore di soglia tramite TalkBack;
3. *SetTemperaturaToThinkSpeak()*, riportata nel **Listato 2**, serve per memorizzare il valore di temperatura sul server ThinkSpeak; questa funzione richiamata periodicamente permetterà di creare, graficamente, l'andamento della temperatura nel tempo.

Entrambe le funzioni 2 e 3 sono state raggruppate all'interno del file di progetto *ESP8266.tbs*, in modo da costituire una sorta di libreria che possa essere utilizzata anche per altri progetti. All'interno dello stesso file potrete notare la presenza di una terza funzione utilizzata da entrambe le prime due: *sendESP8266*. Essa ha il compito di semplificare la comunicazione tra EM1001 e ESP8266, che, ricordiamo, sono connessi tramite la seriale 0. Il prototipo della funzione è:

function sendESP8266(comando as string, porta as byte, attesa as word, risposta as string) as boolean

dove *comando* rappresenta la stringa da inviare tramite l'UART del Tibbo, *porta* indica la seriale del modulo EM1001, *attesa* rappresenta il valore del ritardo della funzione *ritardo()* ed infine *risposta* è la risposta attesa dal modulo con ESP8266. La funzione restituisce *true* se la risposta del modulo è effettivamente quella inserita nella chiamata a funzione, altrimenti *false*.

In base a quanto visto finora, sarà facile comprendere le istruzioni usate per l'upload della tempe-



per il MATERIALE

Il modulo ethernet programmabile (cod. EM1001) viene fornito assemblato ed è reperibile presso Futura Elettronica al prezzo di 59,00 Euro.

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

Listato 2 - Codice della funzione *getSogliaFromTalkBack()*

```
1. function getSogliaFromTalkBack(porta as byte, attesa as word) as string
2.     dim appo_string as string
3.     dim pos as word
4.     dim riavvia as boolean
5.     sendESP8266(talkBackString,0,6000,"")
6.     sendESP8266("AT+CIPSEND=4,72",0,6000,"")
7.     riavvia = sendESP8266(talkback2,0,6000,"ERROR")
8.     ser.num = porta
9.     ser.setdata(chr(13) + chr(10))
10.    ser.send
11.    ritardo(attesa)
12.    while (ser.rxlens > 255)
13.        ' Gestione delle stringhe superiori a 255
14.        appo_string = ser.getdata(255)
15.    wend
16.    if (ser.rxlens > 0) then
17.        appo_string = ser.getdata(ser.rxlens)
18.        pos = instr(1,appo_string,chr(13) + chr(10) + "OK",1)
19.        appo_string = left(appo_string,pos-1)
20.        getSogliaFromTalkBack = right(appo_string,2)
21.    end if
22.    sendESP8266("AT+CIPCLOSE",0,4000,"OK")
23.    if riavvia then
24.        ' Se restituisce ERROR resetta il chip ESP8266
25.        sendESP8266("AT+RST",0,120000,"Ready")
26.        sendESP8266("AT+CIPMUX=1",0,6000,"OK")
27.        io.num = PL_IO_NUM_30
28.        io.state = HIGH
29.    end if
30. end function
```

Listato 3 - Codice della funzione *setTemperaturaToThinkSpeak()*

```
1. sub setTemperaturaOnThinkSpeak(lung_temp as byte, temp_str as string)
2.     sendESP8266(thingSpeakString,0,10000,"Linked") ' Connessione al server
3.     sendESP8266("AT+CIPSEND=4," + str(lung_temp),0,4000,"OK")
4.     sendESP8266(thingSpeakString2 + temp_str,0,4000,"OK") ' Upload temperatura
5.     sendESP8266("AT+CIPCLOSE",0,4000,"OK")
6. end sub
```

Listato 4 - Funzione *sendESP8266()*

```
1. function sendESP8266(comando as string, porta as byte, attesa as word,
risposta as string) as boolean
2.     dim rx_buffer as string
3.     ser.num = porta
4.     ser.setdata(comando + chr(13) + chr(10))
5.     ser.send
6.     ritardo(attesa)
7.     if ser.rxlens > 0 then
8.         rx_buffer = ser.getdata(ser.rxlens)
9.         if (instr(1,rx_buffer,risposta,1) > 0) then
10.             sendESP8266 = true
11.         else
12.             sendESP8266 = false
13.         end if
14.         sock.setdata(rx_buffer + chr(13) + chr(10))
15.         sock.send
16.     end if
17. end function
```

ratura sul server ThinkSpeak (Listato 2):

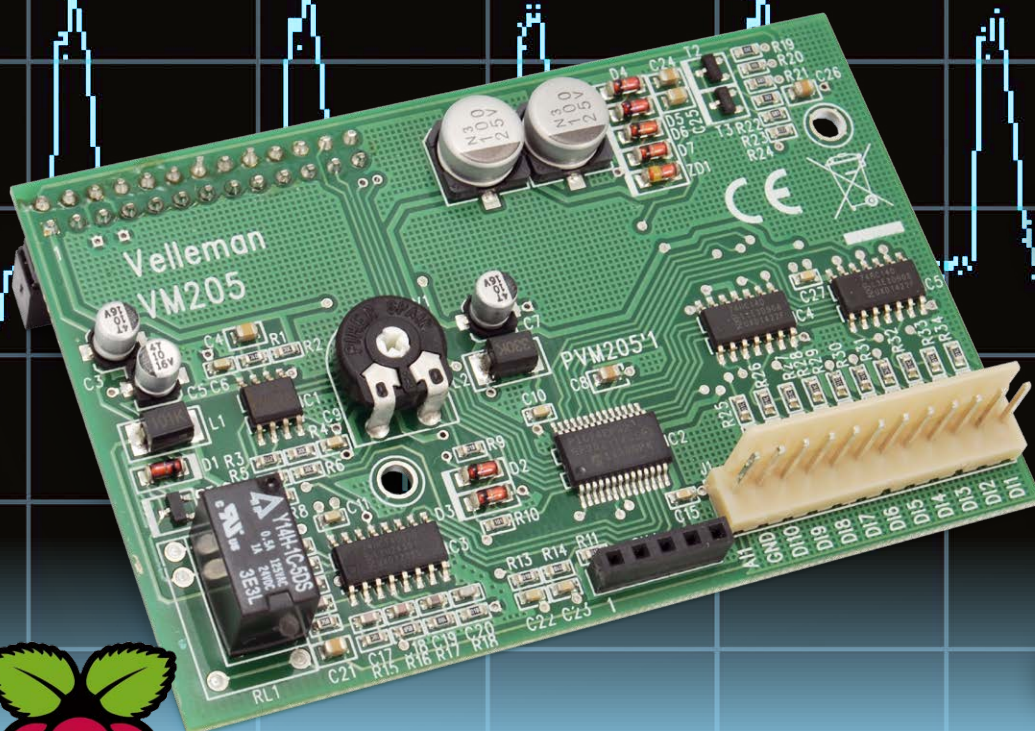
1. connessione al server tramite la stringa di connessione (AT+CIPSTART);
2. indicazione del numero di caratteri da inviare al server per l'upload (AT+CIPSEND);
3. invio della stringa di invio valore al server ThinkSpeak;
4. chiusura della connessione tramite il comando AT+CIPCLOSE.

Ogni volta che viene richiamata questa sequenza di comandi si memorizza un valore sul server; ripetendolo per un certo periodo si ottiene un grafico che riporta l'andamento della temperatura nel tempo. Più articolata è, invece, la funzione per recuperare il valore di soglia memorizzato sul server ThinkSpeak: in aggiunta al procedimento descritto, è necessario effettuare il parsing della risposta del server, all'interno del quale si trova anche il valore del campo richiesto. A questo scopo abbiamo usato le funzioni incluse nel linguaggio di Tibbo denominate *instr*, *left* e *right* (da riga 16 a riga 21 del Listato 2).

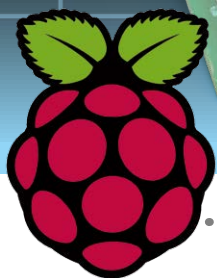
CONCLUSIONI

Siamo ormai giunti al termine di queste tre puntate in cui vi abbiamo mostrato l'intero processo seguito per la progettazione di un dispositivo IoT. Abbiamo selezionato per voi una piattaforma basata sul microcontrollore della Tibbo e lo abbiamo interfacciato con il sensore di temperatura della Maxim DS18B20; i valori di temperatura ottenuti sono stati inviati ad un server esterno per la memorizzazione e successiva consultazione. Ovviamente il progetto Smartem può essere espanso ulteriormente, dotando la shield realizzata di ulteriori sensori come umidità ed altri. ■

OSCILLOSCOPIO PER RASPBERRY PI



Trasformiamo il nostro Raspberry Pi modello B in un oscilloscopio in grado di visualizzare sia segnali analogici che digitali.



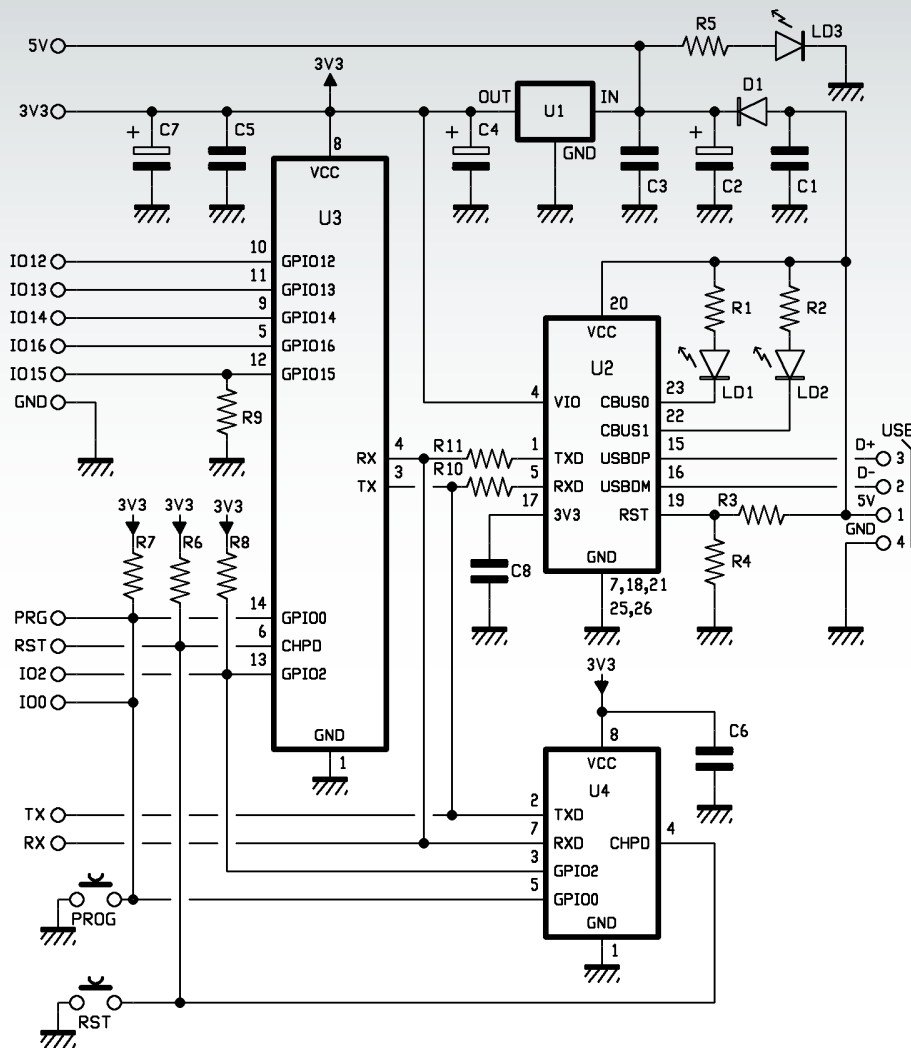
di MARCO MAGAGNIN

State costruendo il vostro primo robot rover? State armeggiando con encoder, segnali PWM, sensori di linea o distanza? Per analizzare questi componenti torna proprio utile un oscilloscopio, anche dalle caratteristiche non particolarmente professionali. Se avete nel cassetto un ... vecchio Raspberry del tipo precedente all'uscita di Raspberry Pi 2, allora la soluzione può essere il modulo che vi presentiamo in questo articolo. La scheda Velleman VM205, è uno shield appositamente progettato per Raspberry Pi (modello B), per ottenere un oscilloscopio digitale in grado di visualizzare, memorizzare e analizzare forme d'onda sia analogiche che digitali. Lo shield, offre tutte le funzionalità di un oscilloscopio standard, con un

canale analogico e ben dieci canali digitali. Come valore aggiunto il codice è disponibile gratuitamente, sia come applicazione pronta a funzionare, sia come codice sorgente e protocollo di comunicazione. A proposito, lo shield utilizza il bus SPI per comunicare con Raspberry Pi. Ma diamo un'occhiata alle specifiche:

- Sezione oscilloscopio:
 - massima frequenza di campionamento : 1MS/s
 - banda passante: DC - 200kHz (-3dB)
 - frequenza massima: 100kHz
 - impedenza d'ingresso : 100kohm // 20pF
 - massima tensione d'ingresso: 30Vp (AC + DC)
 - ingresso: AC+DC
 - risoluzione AD: 8 bit

[schema ELETTRICO]



- lunghezza buffer SPI: 800 campionamenti
- modalità di funzionamento: Run, Single
- trigger da 254 passi
- intervallo base dei tempi: divisione in 15 passi da 5µs a 200ms
- intervallo sensibilità di ingresso: divisione in 6 passi da 100mV a 5V
- sensibilità ingresso: 3mV
- risoluzione display
- marcatori time e Voltage
- lettura frequenza (tramite marcatori)
- Sezione analizzatore logico:
 - canali: 10
 - registrazione: 10x800 samples

- massima frequenza di campionamento: 1MS/s
- massima tensione d'ingresso: 5VDC

Da quanto si evince non ci troviamo di fronte ad uno strumento dalle caratteristiche particolarmente professionali, ma permette, a chi non ha particolari esigenze, di eseguire misure e controlli in ambiti dove le frequenze non sono particolarmente elevate. Quindi per effettuare misure audio e a fini didattici è, almeno nel nostro caso, un utile strumento di misura e controllo nella realizzazione di robot, anche dalle caratteristiche evolute come i robot che partecipano alle com-

petizioni della lega RoboCup. Le caratteristiche di questo oscilloscopio sono più che sufficienti per analizzare le forme d'onda di encoder, driver per motori e segnali provenienti da sensori di vario genere.

SCHEMA ELETTRICO

Il cuore del circuito è il microcontrollore Microchip PIC24EP32GP202T, su architettura a 16 bit. Per la descrizione generale del circuito rimandiamo alla esauriente descrizione contenuta nell'articolo "Il mio primo oscilloscopio digitale" nel numero 196 della rivista (Giugno 2015) in quanto il circuito elettrico corrisponde pressoché totalmente a quello del modello EDU09, sempre di Velleman. La principale differenza è la possibilità di analizzare segnali analogici e la scelta di comunicare con Raspberry Pi utilizzando il bus SPI. Di conseguenza il software a bordo del microcontrollore Microchip implementa la comunicazione SPI verso Raspberry Pi.

Nella Fig. 1 è visibile il layout dello shield e lo schema dei collegamenti per gli ingressi analogico e digitali.

INSTALLAZIONE

Una volta estratto dalla sua confezione sinceratevi che Raspberry Pi sia spenta ed installate lo shield sul connettore GPIO di Raspberry. Fate particolare attenzione a far combaciare i pin del connettore dello shield VM205 con i corrispondenti pin del connettore GPIO, come visibile in Fig. 2. Per il prosieguo dell'installazione presupponiamo che abbiate a disposizione una SD Card con installata una versione recente di Raspbian, il sistema operativo principale disponibile per Raspberry Pi, derivato da Debian.

Per approfondire l'argomento relativo al sistema operativo GNU/Linux e come preparare una SD Card con il sistema operativo Raspbian potete fare riferimento al libro "Raspberry Pi, il mio primo Linux embedded".

Inserite la SD Card nell'apposito alloggiamento sul microcontrollore Raspberry Pi ed accendete il tutto. Per evitare di dedicare un monitor HDMI a Raspberry Pi, oltre che tastiera e mouse, utilizzeremo Raspberry Pi da remoto, nella modalità nota come "headless".

Collegatevi a Raspberry Pi utilizzando il vostro PC con sistema operativo Microsoft Windows e gli strumenti Putty, WinSCP e Mobaxterm facilmente reperibili con una ricerca su internet ed installabili sia in modalità normale che su una chiavetta USB in modalità "portabile". Collegate Raspberry Pi alla rete di casa con un cavo di rete. Ora dovete individuare l'indirizzo IP di Raspberry Pi. Il modo più semplice è di aprire l'applicazione di configurazione del vostro router dove potete reperire l'indirizzo assegnato a Raspberry Pi. Collegatevi a Raspberry Pi in remoto con

Putty od anche con Mobaxterm con l'utente "root" e relativa password se lo avete attivato, oppure con l'utente predefinito "pi" e password "raspberrypi".

Una volta collegati, se non lo avete già fatto, potete attivare l'utente "root" con il comando:

```
sudo passwd
```

vi verrà chiesto di impostare e confermare una password per l'utente root. Ricordatevi di anteporre il comando "sudo" altrimenti modificherete unicamente la password per l'utente "pi". Potete accedere all'utente "root" con il comando:

```
su
```

dopo il quale vi verrà richiesta la password di root. Ora aggiornate il sistema operativo con i classici comandi:

```
apt-get update
apt-get upgrade
apt-get dist-upgrade
```

Una volta aggiornato il sistema è bene espandere il file system, se non lo avete ancora fatto, e

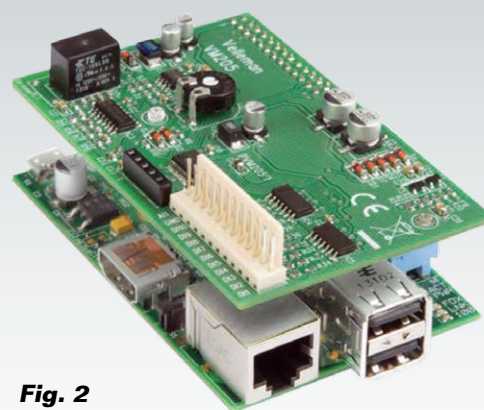


Fig. 2

poi attivare i driver per il bus SPI, installare il programma per la remotizzazione dello schermo ed il programma di gestione dell'interfaccia grafica dell'oscilloscopio. Per le prime due operazioni utilizzate il comando:

```
raspi-config
```

che apre il menu dello strumento di configurazione di Raspbian. Iniziamo con l'espandere il file system per occupare tutto lo spazio disponibile sulla SD Card. Per questo selezionate la voce di menu "Expand rootfs" e confermate con "Yes" nella pagina successiva, ed in quella successiva ancora rispondete con "No" alla proposta di eseguire il reboot immediatamente. Prima attivate il driver per la gestione del bus SPI, selezionate la voce di menu "Advanced Options" e, nella pagina successiva, selezionate "Enable SPI" (Fig. 3). Confermate con Yes a tutte le domande (Fig. 4, 5, 6 e 7), uscite dal menu di raspi-config con "Finish" e, questa volta, accettate la proposta di reboot. A reboot terminato dovrete ricollegarvi a Raspberry Pi. Per sincerarvi che tutto sia andato a buon fine ed il driver sia correttamente installato potete utilizzare lo strumento WinSCP, collegandovi come utente "root" ed andando alla cartella /etc/dev dove dovrete trovare le due entrate *spidev0.0* e *spidev0.1* come visibile in Fig. 8. In alternativa,

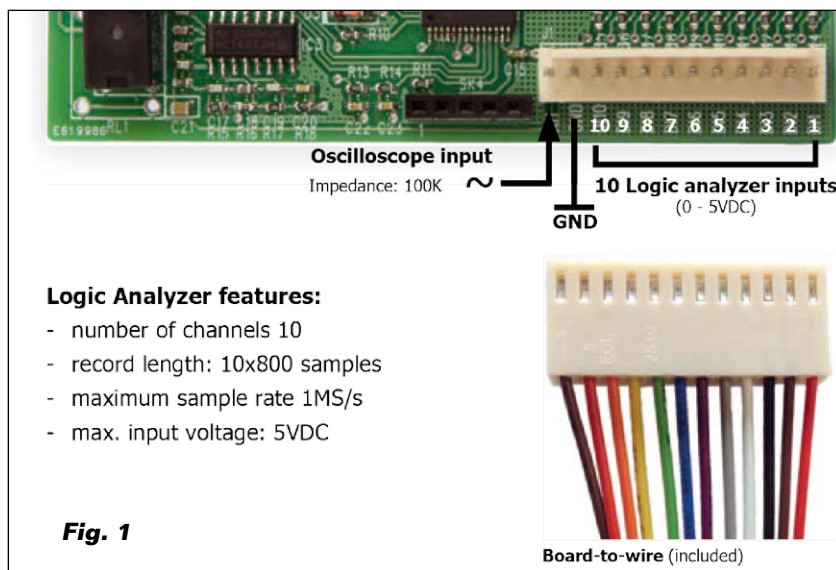
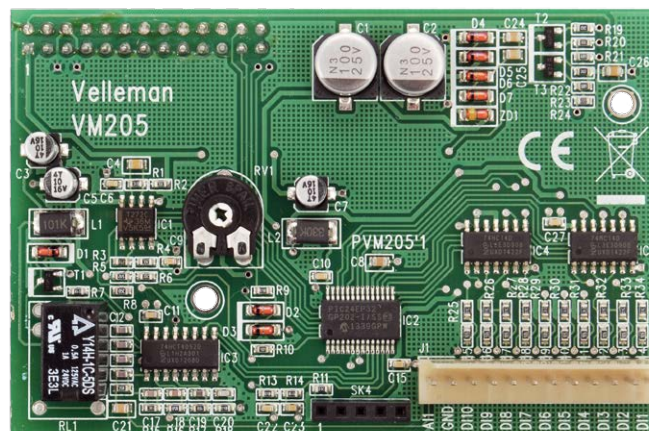
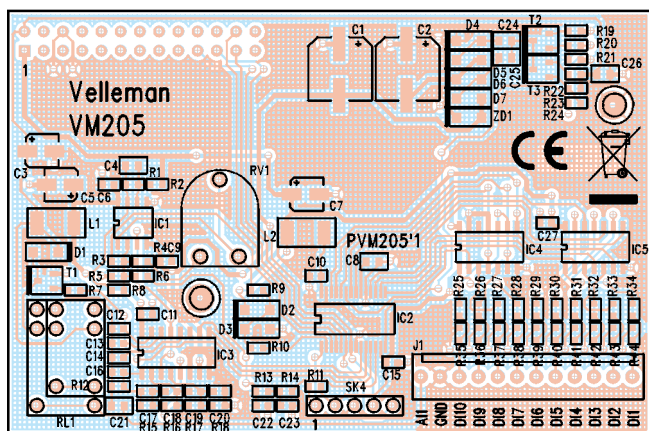


Fig. 1



Elenco Componenti:

R1, R11: 10 kohm (0603)
 R2, R4, R6: 20 kohm (0603)
 R3: 2,7 kohm (0603)
 R5: 5,1 kohm (0603)
 R7, R9: 1 kohm (0603)
 R8: 11 kohm (0603)
 R10: 100 ohm (0603)
 R12: 75 kohm (0603)
 R13, R14, R18: 1 kohm (0603)
 R15: 15 kohm (0603)
 R16: 7,5 kohm (0603)
 R17, R20: 1,5 kohm (0603)
 R19: 2,2 ohm (0603)
 R21, R23: 680 ohm (0603)
 R22, R24: 2,2 ohm (0603)
 R25÷R34: 10 kohm (0603)

R35÷R44: 1 Mohm (0603)
 RV1: Trimmer 5 kohm MO
 C1, C2: 100 μ F 25 VL elettrolitico ($\pm$ 8mm)
 C3: 10 μ F 16 VL elettrolitico ($\pm$ 4mm)
 C4: 1 μ F ceramico (0805)
 C5: 10 μ F 16 VL elettrolitico ($\pm$ 4mm)
 C6, C9÷C11: 100 nF ceramico (0603)
 C7: 10 μ F 16 VL elettrolitico ($\pm$ 4mm)
 C8: 10 μ F ceramico (0805)
 C12: 2,2 pF ceramico (0603)
 C13: 4,7 pF ceramico (0603)
 C14: 12 pF ceramico (0603)
 C15: 100 nF ceramico (0603)
 C16: 10 pF ceramico (0603)
 C17: 47 pF ceramico (0603)

C18: 100 pF ceramico (0603)
 C19: 470 pF ceramico (0603)
 C20: 680 pF ceramico (0603)
 C21: 1 μ F ceramico (0805)
 C22, C23, C27: 100 nF ceramico (0603)
 C24÷C26: 1 μ F ceramico (0805)
 D1÷D7: BAT85 (MiniMELF)
 ZD1: Zener 5,1V 400mW
 T1, T2: 2N2222A (SOT-23)
 T3: 2N2907A (SOT-23)
 L1: Induttanza 100 μ H (DO-214BA)
 L2: Induttanza 33 μ H (DO-214BA)
 IC1: TLV272ID

IC2: PIC24EP32GP202T-I/SS
 IC3: CD74HCT4052M
 IC4, IC5: 74HC14D
 RL1: Relè 5V 1 scambio

Varie:

- Connettore Board-To-Wire Maschio 12 vie da CS
- Connettore Board-To-Wire Femmina 12 vie
- Strip femmina 5 vie
- Connettore per Raspberry Femmina 2x13 vie
- Distanziali Nylon F/F 20mm 2 MA (2 pz.)
- Vite Nylon 2 MA (4 pz.)
- Circuito stampato

dalla finestra di terminale (PuTTY) potete dare il comando:

```
lsmod
```

che elenca tutti i driver installati tra i quali troverete il driver *spi-dev*. Ora è tempo di scaricare il software di gestione dell'oscilloscopio ed installarlo su Raspberry Pi. Andate all'indirizzo dove sono disponibili i file di download per l'oscilloscopio VM205:

<https://www.velleman.eu/support/downloads/?code=VM205>

Da questa pagina (**Fig. 9**) selezionate la voce "Software package

for VM205 with Pascal source code (Rev.10)" e salvate il file compreso in una cartella di comodo del vostro PC.

Dalla stessa pagina potete scaricare anche il manuale e gli altri documenti informativi sullo shield. Una volta scaricato decomprimitelo il file in una cartella di comodo, noi abbiamo nominato la cartella "osc", come visibile in **Fig. 10**.

Fatto questo, è necessario trasferire la cartella scompattata nel file system di Raspbian. Il modo più comodo che conosciamo è di utilizzare lo strumento WinSCP. Se non siete già collegati, configurate WinSCP per accedere a Raspberry Pi con l'indirizzo IP e l'utente

root. Vi verrà chiesta la password di root e poi si aprirà la schermata dello strumento, composta da due finestre affiancate.

Nella finestra di sinistra viene mostrato il file system su PC mentre in quella di destra è visibile il file system di Raspberry Pi.

Per spostare file e cartelle da una parte all'altra è sufficiente trascinarle.

Dunque posizionatevi sulla cartella lato PC che contiene i file scompattati mentre lato Raspberry Pi posizionatevi sulla cartella /home/pi.

Ora trascinate la cartella, nel nostro caso di nome "osc", dal file system del PC a quello di Raspberry

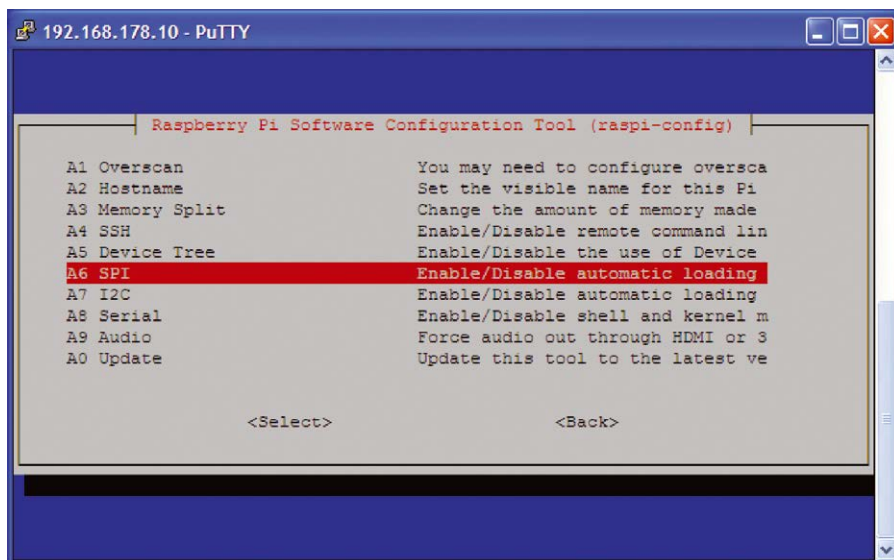


Fig. 3

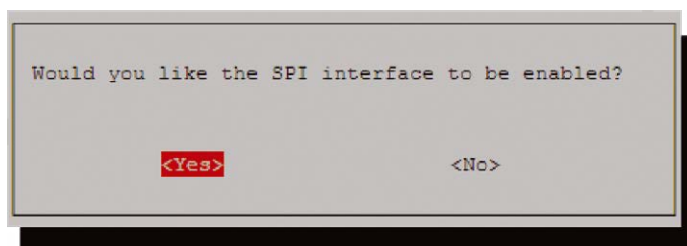


Fig. 4

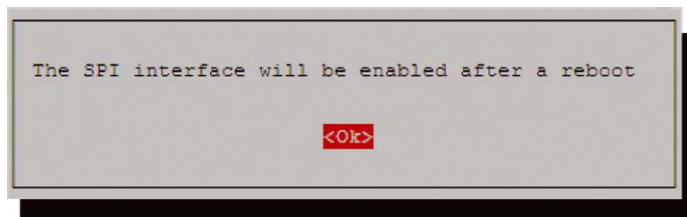


Fig. 5

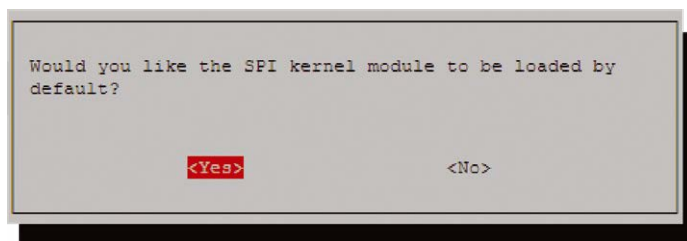


Fig. 6

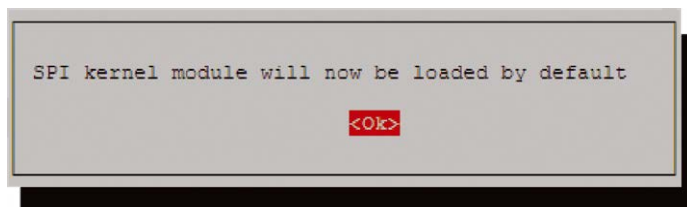


Fig. 7

Pi (Fig. 11). Prima di chiudere WinSCP dovete dare i privilegi di esecuzione al file "VM205" all'interno della cartella "osc".

Cliccate sul file VM205 col tasto destro del mouse e nel menu a tendina selezionate "Proprietà" e "spuntate" tutti i quadratini con i caratteri "x". In questo modo diamo la possibilità a tutti gli utenti di eseguire il file. Ora dobbiamo fare in modo di poterci collegare da remoto al desktop di Raspberry Pi. Su Raspberry Pi utilizzeremo il programma "tightvncserver" e Mobaxterm sul PC.

Per installare il pacchetto tightvncserver, dalla finestra di terminale (PuTTY) digitate il comando:

```
apt-get install tightvncserver
```

Durante l'installazione viene richiesta una password per collegarsi da remoto, sceglietene una di otto caratteri, digitatela e confermatela. Alla seconda richiesta di una password per collegarsi in

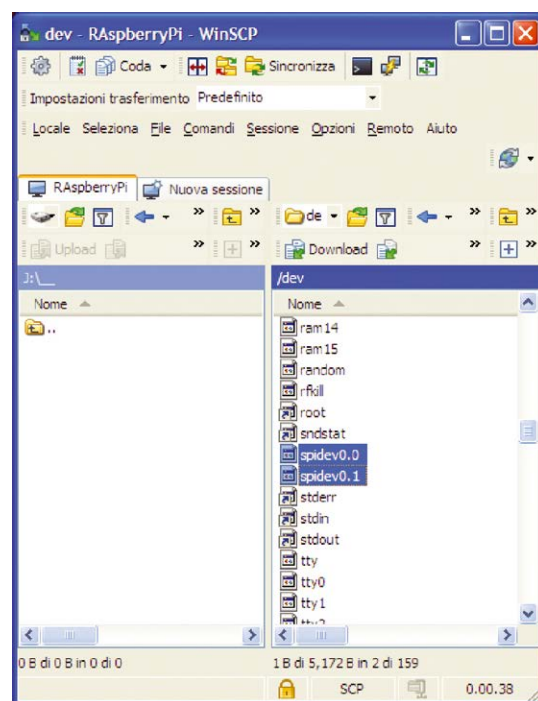
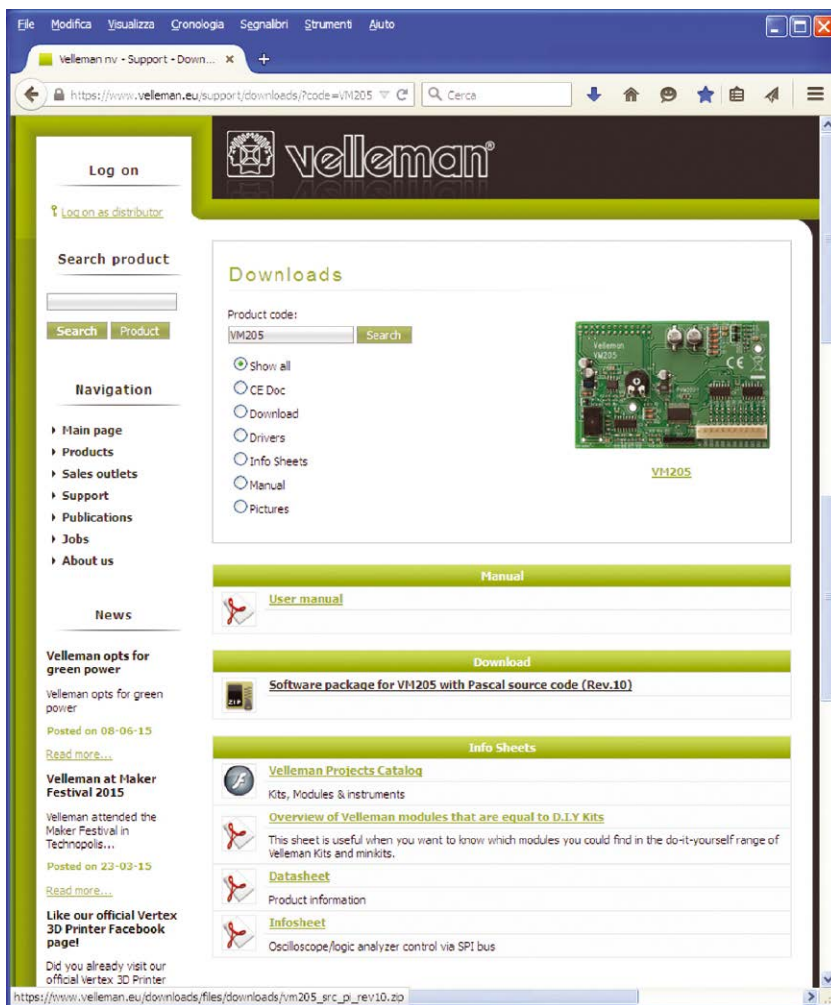


Fig. 8



sola lettura potete tranquillamente rispondere di no. Terminata l'installazione dovete attivare una sessione di terminale remoto con il comando (Fig. 14):

```
tightvncserver :1 -geometry
1024x768 -depth 24
```

A questo punto potete passare sul PC e lanciare il programma MobaXterm. Una volta aperto cliccate sulla grossa "X" rossa in alto a destra in modo da attivare il "server X" sul vostro PC. Cliccate sulla voce di menu "Session" e scegliete l'opzione VNC. Inserite l'indirizzo IP del vostro Raspberry Pi seguito dal suffisso :1 per esempio (Fig. 15):

```
192.168.0.10:1
```

Vi verrà richiesta la password per il collegamento remoto che avete impostato durante l'operazione di installazione di tightvncserver (Fig.16).

Per magia in MobaXterm si aprirà

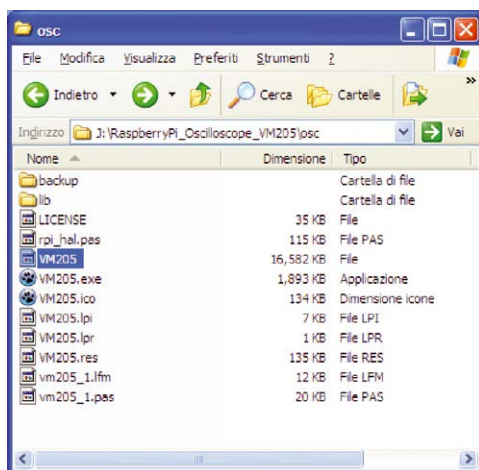


Fig. 10

Fig. 9

una scheda con all'interno il desktop di Raspberry Pi. Cliccate sull'opzione "FullScreen" per aprire il desktop a tutto terminale. Per tornare alla visualizzazione normale "appoggiate" la "freccia" del mouse sul bordo in alto dello schermo. Riapparirà una barra di menu sulla quale cliccate l'opzione "FullScreen".

Dal desktop di Raspberry Pi aprite una finestra di terminale che utilizzerete per lanciare l'interfaccia grafica dell'oscilloscopio. Nella finestra andate alla cartella che

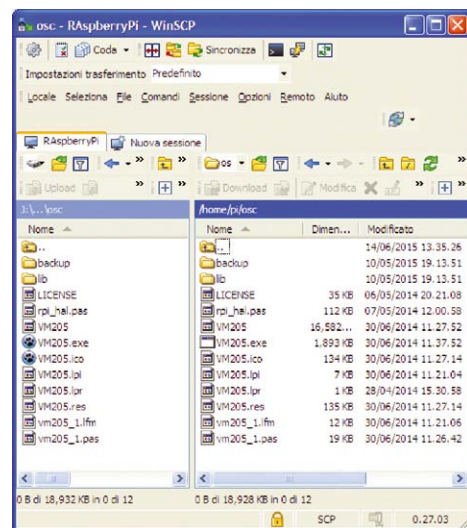


Fig. 11

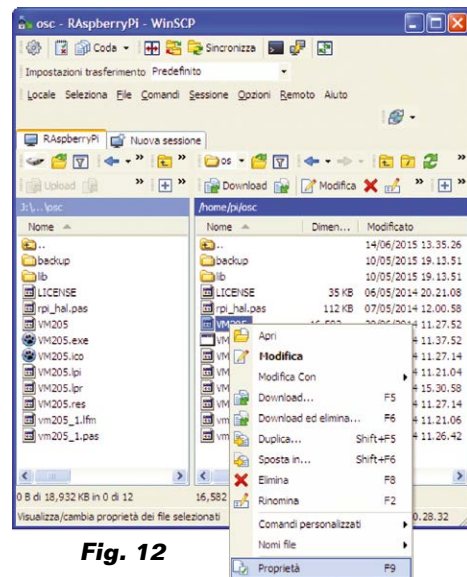


Fig. 12

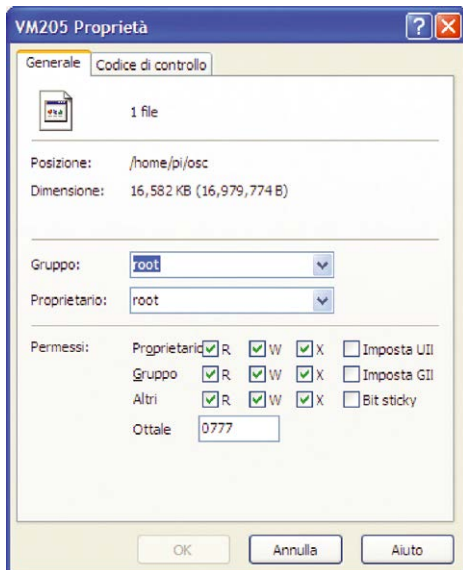


Fig. 13

contiene il programma "VM205" con il comando:

```
cd /home/pi/osc
```

e lanciate il programma con il comando (Fig. 17):

```
./VM205
```

A questo punto cliccate su "Run" per iniziare l'acquisizione dei dati.

TEST DELL'OSCILLOSCOPIO

Per verificare che tutto funzioni a dovere colleghiamo un cavetto al canale analogico dello strumento oscilloscopio, modifichiamo la base dei tempi, impostando il valore 20 msec/div. Impostiamo a 1 V/div la scala delle tensioni. A questo punto impugnate il terminale del cavetto che fa capo al canale analogico e toccate con un dito l'estremità. Nessun timore, non c'è pericolo. Dovreste vedere una forma d'onda, più o meno sinusoidale, come quella mostrata in Fig. 16. Ogni ciclo della forma d'onda copre un intervallo da 20ms (0.02 secondi). Dato che la frequenza è il reciproco dell'intervallo di tempo possiamo calcolare la frequenza di $1/0,02 = 50\text{Hz}$, ovvero la frequenza della rete

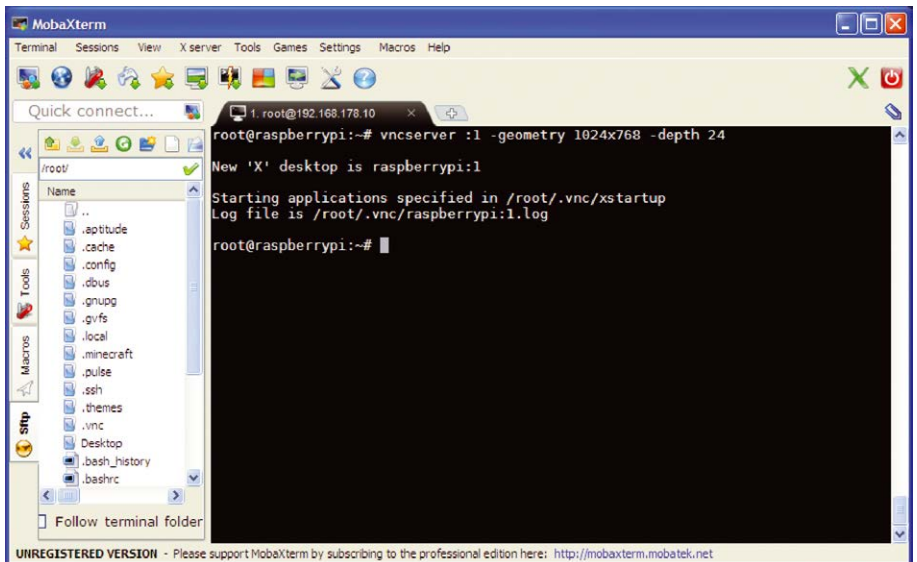


Fig. 14

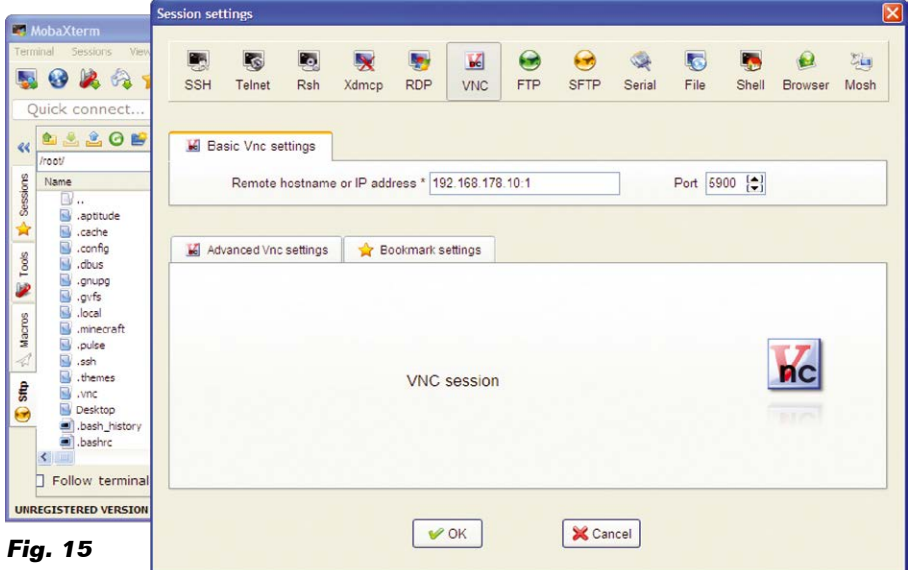


Fig. 15

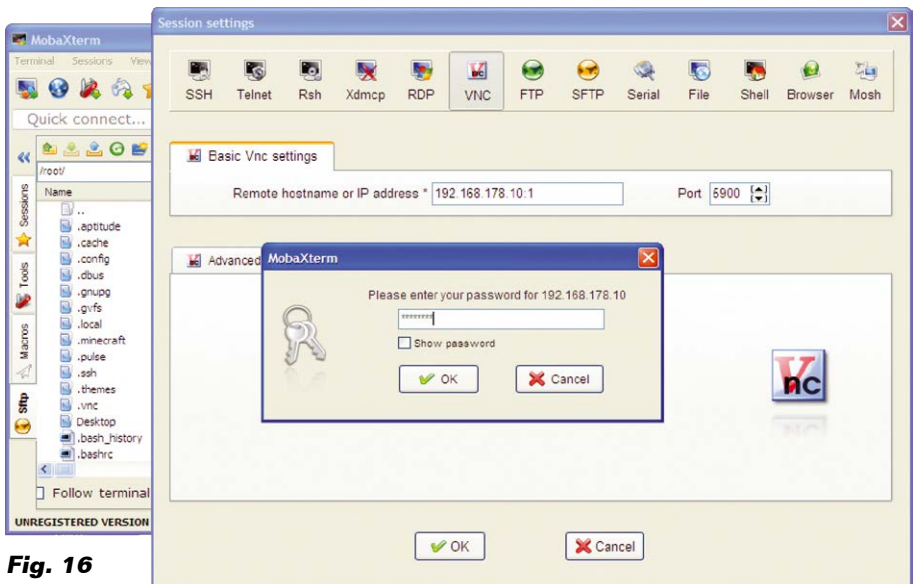


Fig. 16

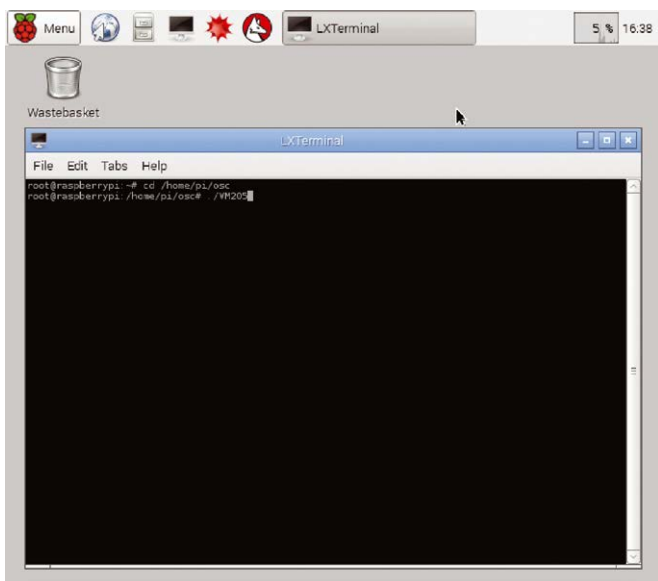


Fig. 17

elettrica. Siccome qualsiasi conduttore attraversato da una corrente alternata si trasforma in un trasmettitore di onde radio e il nostro corpo funge da ricevente, la forma d'onda ricevuta può essere evidenziata con un oscilloscopio, anche se un po' distorta.

Per sperimentare con il nostro oscilloscopio VM205 abbiamo deciso di utilizzare il modulo

EDU06, sempre prodotto da Velleman, che vi abbiamo presentato sul numero 197 della rivista. Il modulo EDU06 è stato appositamente progettato per testare gli oscilloscopi. Abbiamo collegato il modulo EDU06 all'oscilloscopio VM205 come visibile in

Fig. 19.

Di fatto abbiamo collegato l'uscita del generatore di onda sinusoidale all'ingresso analogico e

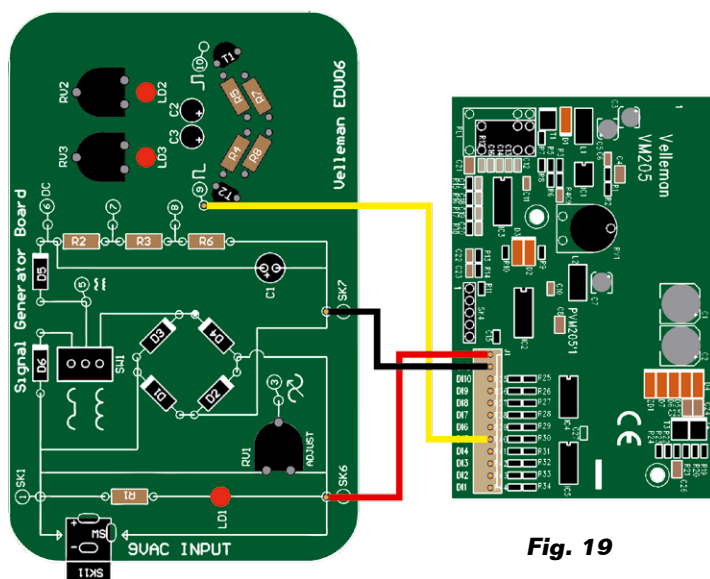


Fig. 19

dell'oscilloscopio e l'uscita del segnale ad onda quadra all'ingresso digitale numero 5 (segno di spunta in corrispondenza del valore "5" a destra della videata dell'oscilloscopio). Il risultato è la forma d'onda a 50 Hz nella sezione analogica, questa volta assai meglio definita che nel caso precedente ed il segnale digitale corrispondente all'onda quadra nella sezione digitale. ■

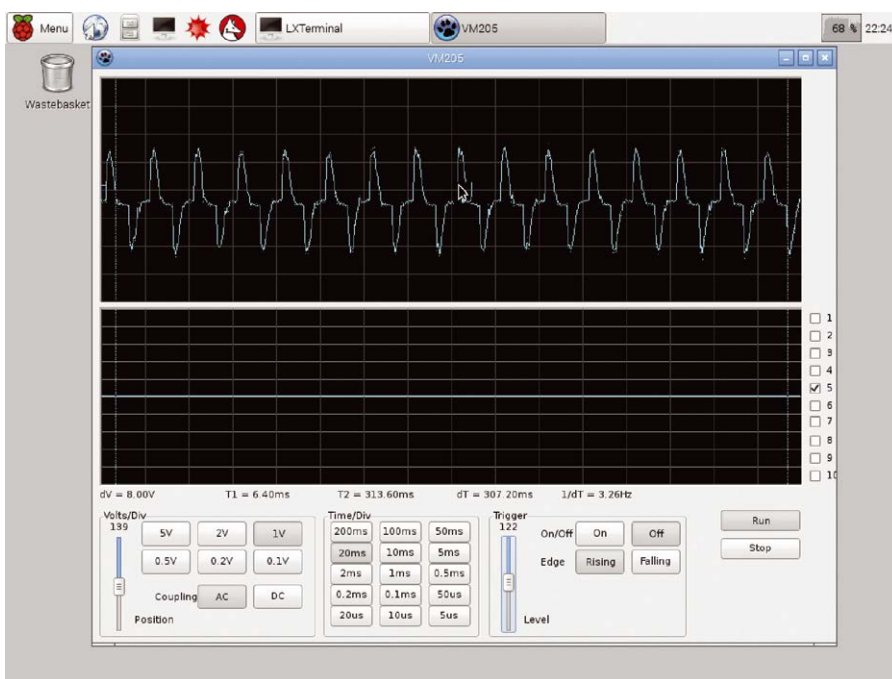


Fig. 18



per il MATERIALE

Questi progetti vengono prodotti dalla società Velleman e distribuiti in Italia da Futura Elettronica. L'oscilloscopio per Raspberry Pi B (cod. VM205) è disponibile al prezzo di 56,00 Euro, mentre la demoboard per oscilloscopio (cod. EDU06) costa Euro 24,00. Le schede vengono fornite montate e collaudate; il software dell'oscilloscopio VM205 è scaricabile dalla scheda on-line del prodotto.

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

BREAKOUT BOARD

NE555

Permette di realizzare circuiti astabili e monostabili sfruttando il più popolare tra i circuiti integrati timer.

di BORIS LANDONI

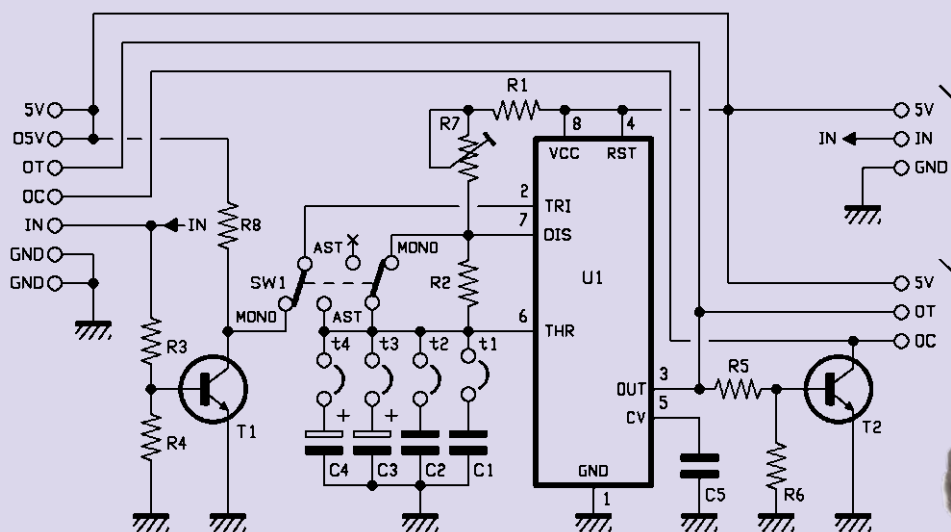


Uno dei circuiti integrati che hanno fatto la storia dell'elettronica allo stato solido è indubbiamente l'NE555, perché quando nacque permise di realizzare svariate funzioni particolari che prima potevano essere ottenute solo realizzando per ognuna un circuito specifico a componenti discreti; il 555 è un timer configurabile in vario modo che incorpora una coppia di comparatori le cui uscite sono collegate agli ingressi di SET e RESET di un flip-flop RS, in modo da realizzare multivibratori astabili e monostabili, ovvero bistabili. Ma non solo: la possibilità di traslare la tensione di riferimento della coppia di comparatori, e perciò quella alla quale avviene la commutazione, consente di far generare all'NE555, nella configurazione astabile, segnali di frequenza variabile, implementando un vero e proprio modulatore di frequenza (FSK) con pochissimi componenti esterni e con una minima occupazione di spazio sul circuito stampato. Quale tributo al successo riscosso dall'NE555, che

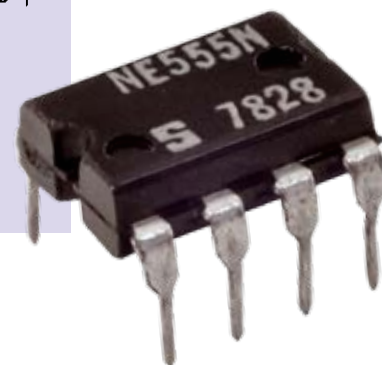
negli anni è stato peraltro migliorato nelle prestazioni (una ventina di anni fa lavorava, come astabile, a una frequenza massima di 200 kHz mentre i modelli prodotti attualmente si spingono a mezzo MHz), e in considerazione della grande utilità, sempre attuale, abbiamo dedicato al 555 una specifica breakout board, vale a dire una scheda di prototipazione in formato miniatura che permette di sperimentarne le funzionalità e di integrarle in altri circuiti anche solo per la fase prototipale. La versione del 555 utilizzata in questo circuito è quella SMD, per restare in dimensioni contenute, come abbiamo fatto in tutte le breakout board che avete visto pubblicate nei fascicoli dei mesi passati.

L'INTEGRATO 555

Prima di dare uno sguardo allo schema elettrico è opportuno spendere qualche paragrafo per analizzare lo schema interno dell'integrato, inizialmente prodotto dalla Signetics e poi da tante altre case costruttrici



L'originale 555 in contenitore DIP a 4+4 piedini. A fondo pagina, lo schema a blocchi interno del 555, che vale per tutti i case.



di semiconduttori tra cui Philips, SGS-Thomson (ora STM), Fairchild ecc.

Come mostra la **Fig. 1**, il componente contiene due comparatori aventi in comune un partitore di tensione multiplo che polarizza per l'operazionale in alto l'ingresso invertente e per quello inferiore il non-invertente. Dei comparatori vengono resi accessibili dall'esterno gli ingressi non-invertenti di quello superiore (i termini superiore e inferiore si riferiscono al potenziale di riferimento che ricevono dal partitore multiplo, quindi quello superiore riceve la tensione più elevata...), corrispondente al piedino 6 (TRESHOLD) e l'invertente di quello inferiore, corrispondente al piedino 2 (TRIGGER). Il nodo tra la prima e la seconda (considerate dall'alimentazione positiva) resistenza del

partitore multiplo di riferimento dei due comparatori (alimentato, come il resto degli stadi interni al 555, dal piedino 8) viene portato all'esterno tramite il piedino 5 (Control Voltage) e ciò permette di alterare le tensioni di riferimento dei comparatori, così da controllare con una tensione esterna la frequenza di lavoro nella configurazione astabile (operando lo shift di frequenza come si fa nei VCO, ossia gli oscillatori controllati in tensione) oppure la durata degli impulsi in quella da timer (monostabile).

Più esattamente, nella configurazione astabile, abbassando la tensione sul piedino 5 si accorcia il periodo della commutazione dell'uscita e quindi si incrementa la frequenza generata, viceversa lasciando salire la tensione fino al livello caratteristico (a vuoto, ossia senza alcun circuito collegato al piedino 5) o incrementandola, si allunga il periodo e si riduce la frequenza prodotta. Nella configurazione monostabile, invece, la tensione altera la durata degli impulsi, il che permette, pilotando il trigger con impulsi a frequenza costante, di ottenere un modulatore PWM.

Le uscite dei comparatori entrano una nel RESET (quella del compa-

ratore superiore) e nel SET (comparatore inferiore) di un flip-flop di tipo RS, che è un circuito logico la cui uscita diretta (Q) va a livello alto (corrispondente a circa il potenziale del piedino 8) quando il SET è a livello alto, ovvero assume lo zero logico se SET è posto a circa zero volt; l'uscita presenta un comportamento esattamente opposto quando ad essere stimolato è l'ingresso RESET, il quale, posto a livello logico alto, resetta il flip-flop, intendendo con ciò che ne azzerà lo stato di Q. Il flip-flop ha anche un'uscita complementata ($\bar{Q}$), la cui condizione logica è sempre l'inverso di quella Q; nel 555 è collegata, tramite una resistenza, alla base di un transistor NPN configurato ad open collector il cui emettitore è connesso a massa e il collettore al piedino DISCHARGE (7). Il transistor può commutare una corrente di collettore di 200 mA, utilizzato in modo sink (ad assorbimento di corrente). L'uscita diretta, Q, è invece collegata al piedino OUT (3) dell'integrato attraverso un buffer interno push-pull, capace di erogare un massimo di 200 mA. Il negativo di

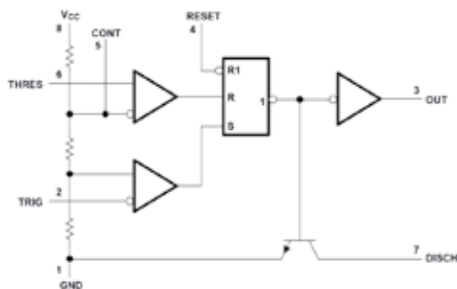


Fig. 1 - Schema interno del 555.

alimentazione, vale a dire la massa di riferimento dell'NE555 corrisponde al piedino 1, mentre al 4 è connesso il reset a logica invertita del flip-flop: questo piedino permette di resettare forzatamente il circuito dall'esterno applicando un livello logico basso.

SCHEMA ELETTRICO

Ora che sappiamo come è fatto il 555 possiamo vedere lo schema elettrico della breakout board e ci sarà facile comprendere come funziona l'integrato nelle due configurazioni rese possibili da essa: il multivibratore astabile e il monostabile. Nella scheda l'integrato (siglato U1) è montato nella classica configurazione che vede il piedino 1 collegato a massa e l'8 al positivo di alimentazione; anche il 4 (RESET invertito) è fisso al positivo, perché qui non c'interessa utilizzare la funzione di reset esterno.

Il Control Voltage non viene utilizzato e, conformemente alle indicazioni del costruttore, lo colleghiamo a un condensatore di filtro (C5) terminante a massa; l'uscita, invece, pilota un transistor che serve ad amplificare la corrente erogabile, fino a 500 milliampere. Il T2 è in configurazione open collector e può lavorare in circuiti alimentati a tensione non superiore a 45 volt cc; viene polarizzato in base tramite il partitore resistivo formato da R5 ed R6, cosicché si trova a riposo quando lo stato logico dell'uscita OUT è basso e in saturazione quando, viceversa, l'uscita si trova ad 1 logico (livello alto).

Veniamo alla parte più interessante del circuito, ossia i piedini 2, 6 e 7, che sono quelli che opportunamente configurati consentono di ottenere le due modalità di funzionamento da noi proposte, ossia la monostabile e la bistabile: nel circuito sono gestiti tramite un

doppio deviatore. Per ottenere la prima modalità bisogna spostare il deviatore su MONO, allorché il piedino DIS viene collegato a una batteria di condensatori selezionabile mediante quattro jumper siglati t1, t2, t3, t4; la stessa sezione del doppio deviatore cortocircuita, escludendola, R2, utilizzata nella configurazione astabile, oltre a unire i piedini 6 e 7.

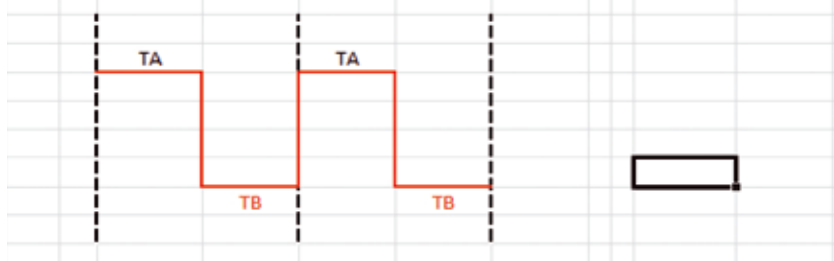
L'altra metà del doppio deviatore, invece, collega il piedino 2 al collettore del transistor T1, il quale funziona da interruttore statico per commutare lo stato dell'ingresso di trigger del 555; la base del transistor, tramite il partitore resistivo formato da R3 ed R4, viene polarizzata tramite la tensione che applicheremo all'ingresso IN. Quest'ultimo è dunque l'input di trigger, attivo nella sola modalità

monostabile (dato che nella posizione AST il deviatore esclude T1) e progettato per ricevere impulsi di tensione di valore compreso fra 3 e 12 volt (tipicamente TTL 0/5V).

Il funzionamento come monostabile si spiega così: partendo dalla condizione di condensatore di temporizzazione scarico (è quello collegato dai jumper tra il piedino 6 e massa...), alimentando il circuito, inizialmente THR si trova a zero volt e quindi, per effetto della polarizzazione interna dovuta al partitore multiplo, l'uscita del comparatore superiore si porta a zero logico risultando ininfluente ai fini dello stato del flip-flop. Il pin TRI in condizioni di riposo (nessuna tensione su IN) è al livello imposto dalla polarizzazione interna dell'integrato e perciò

Tabella 1

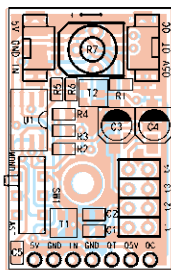
		ASTABILE					MONOSTABILE	
		Tmin = 1 kohm		Tmax = 1 Mohm			1 kohm	1 Mohm
C (uF)	C (F)	TAmin (ms)	TBmin (ms)	TAmax (ms)	TBmax (ms)	t	Tmin (ms)	Tmax (ms)
1 uF	0,000001	0,153153	0,15246	0,846153	0,15246		0,0011	1,1011
2,2 uF	0,000022	0,336937	0,335412	1,861537	0,335412		0,00242	2,42242
3,3 uF	0,000033	0,505405	0,503118	2,792305	0,503118	t1	0,00363	3,63363
4,7 uF	0,000047	0,719819	0,716562	3,976919	0,716562		0,00517	5,17517
5,8 uF	0,000058	0,888287	0,884268	4,907687	0,884268		0,00638	6,38638
10 uF	0,00001	1,53153	1,5246	8,46153	1,5246	t2	0,011	11,011
12 uF	0,000012	1,837836	1,82952	10,15384	1,82952		0,0132	13,2132
15 uF	0,000015	2,297295	2,2869	12,6923	2,2869		0,0165	16,5165
18 uF	0,000018	2,756754	2,74428	15,23075	2,74428		0,0198	19,8198
22 uF	0,000022	3,369366	3,35412	18,61537	3,35412		0,0242	24,2242
33 uF	0,000033	5,054049	5,03118	27,92305	5,03118	t3	0,0363	36,3363
47 uF	0,000047	7,198191	7,16562	39,76919	7,16562		0,0517	51,7517
58 uF	0,000058	8,882874	8,84268	49,07687	8,84268		0,0638	63,8638
68 uF	0,000068	10,4144	10,36728	57,5384	10,36728		0,0748	74,8748
100 uF	0,0001	15,3153	15,246	84,6153	15,246	t4	0,11	110,11



[piano di **MONTAGGIO**]

Elenco Componenti:

R1: 1 kohm (0805)	SW1: Deviatore 2 vie
R2: 220 kohm (0805)	
R3: 4,7 kohm (0805)	Varie:
R4: 10 kohm (0805)	- Strip maschio
R5: 4,7 kohm (0805)	- 4 vie (2 pz.)
R6: 10 kohm (0805)	- Jumper
R7: Trimmer 1 Mohm	- Strip maschio
C1: 3,3 μ F ceramico (0805)	- 7 vie 90°
C2: 10 μ F ceramico (0805)	- Connettore
C3: 33 μ F 6,3 VL elettrolitico (Δ E 4mm)	JST 3 vie
C4: 100 μ F 6,3 VL elettrolitico (Δ E 4mm)	passo 1.25mm da CS (2 pz.)
C5: 10 nF ceramico (0805)	- Connettore JST
T1: BC817	volante 3 vie
T2: BC817	passo 1.25mm (2 pz.)
U1: NE555DR	- Circuito stampato
	S1229



anche il secondo comparatore ha l'uscita a livello basso, perché la polarizzazione interna ne pone l'ingresso non-invertente a livello basso, pertanto anche l'uscita del 555 assume questo livello.

Alimentando il contatto IN della scheda, il transistor T2 va in saturazione e il suo collettore trascina il piedino 2 del 555 a un potenziale minore di quello imposto dal partitore interno, il che fa scattare l'uscita del relativo comparatore a 1 logico, il che setta il flip-flop, determinando il livello alto sul piedino OUT dell'U1.

Per effetto di ciò l'uscita complementata del flip-flop si porta a livello basso e lascia interdire il transistor interno, il cui collettore è collegato al piedino 7, lasciando quindi caricare il condensatore di temporizzazione.

Quando la differenza di potenziale su quest'ultimo supera quella che il partitore di tensione interno al 555 applica al non-invertente del comparatore superiore, quest'ultimo commuta la propria uscita dal livello basso a quello alto e resetta il flip-flop, riportan-

do a zero logico l'uscita OUT. Ciò avviene in un tempo pari a:

$$t = 1,1 \times R \times Ct$$

dove R è la resistenza di carica del condensatore di temporizzazione (Ct) inserito tra i piedini 6,7 e massa; essendo questa composta da R1 e dal trimmer R7, possiamo variare la durata dell'impulso con una certa flessibilità. La **Tabella 1** riporta tutte le possibili combinazioni tra i condensatori selezionati con i jumper e i valori minimi e massimi di R, corrispondenti rispettivamente ad avere il cursore tutto verso i piedini 6, 7 e tutto verso R1. Il tempo ottenibile dalla formula è in secondi se R è espressa in kohm e Ct in millifarad ($1 \text{ mF} = 1.000 \mu\text{F}$) ovvero R in Mohm e Ct in microfarad.

A questo punto lo stato logico basso all'uscita del flip-flop determina l'1 logico su /Q e la saturazione del transistor interno collegato al piedino 7, il quale quasi cortocircuita il condensatore di temporizzazione scaricandolo, fin quando non giunge un nuovo

impulso di comando sul trigger. Notate che se IN viene alimentato continuamente, quando l'uscita OUT torna a zero logico il monostabile viene retriggerato e comincia un nuovo ciclo di temporizzazione. Vediamo adesso il funzionamento astabile, ottenuto portando il doppio deviatore nella posizione AST: in questo caso il piedino 2 viene sconnesso dal collettore del T1 (quindi lo stato di IN risulta ininfluente) e da R6 viene rimosso il cortocircuito, cosicché i piedini 2 e 6 vengono uniti e il 7 risulta collegato solo ad R7 (che decide stavolta la frequenza di lavoro del multivibratore, sempre secondo la **Tabella 1**) e ad R2. THR si connette ad R2 e al condensatore scelto tramite i jumper, che diventa il Ct . La frequenza di commutazione dell'uscita dell'NE555 nella configurazione astabile è data dalla formula:

$$f = 1/(Ct \times 2R)$$

dove Ct è il condensatore selezionato e R la resistenza data dalla somma di R1 ed R7 (valgono le considerazioni già fatte per il monostabile). Vediamo subito come funziona questo multivibratore, supponendo di aver chiuso il doppio deviatore su AST e di alimentare il circuito in condizioni di condensatori scarichi. Trascurando le vicende di C5, che si carica quasi subito e quindi non disturba la tensione di riferimento applicata dal partitore multiplo ai comparatori interni al 555, la condizione iniziale del circuito vede i piedini 2 e 6 a circa zero volt e quindi l'uscita del comparatore inferiore a livello alto; ciò determina il livello logico alto sull'ingresso SET del flip-flop e lo stesso sull'uscita Q (la /Q si porta a zero logico lasciando interdito il transistor collegato al DISCHARGE e quindi il condensatore libero

di caricarsi). A questo punto Ct prende a caricarsi attraverso la corrente che fluisce nella serie R1/R7 ed anche in R2, fin quando supera la tensione di soglia del comparatore inferiore, che è pari a 1/3 di quella applicata al piedino 8; ora l'uscita del suddetto comparatore si porta a zero logico e libera il SET del flip-flop, la cui uscita rimane comunque a livello alto fin quando la tensione ai capi del Ct, continuando a crescere, non si porta a un valore superiore ai 2/3 della tensione di alimentazione del 555, ovvero alla soglia di commutazione del comparatore superiore. Quando ciò accade, l'uscita del comparatore superiore si porta a livello logico alto e resetta il flip-flop, la cui uscita diretta passa a zero logico mentre la /Q si porta a livello alto, mandando in saturazione il transistor interno collegato al piedino 7, il quale attraverso R2 scarica il condensatore Ct. La situazione rimane questa fin quando la tensione su detto condensatore non scende al disotto della soglia di commutazione del comparatore inferiore, allorché l'uscita di quest'ultimo torna a livello logico alto e setta il flip-flop, che riporta Q a 1 logico e /Q a zero. In questa condizione il condensatore riprende a caricarsi perché il piedino 7 torna aperto. La carica avviene nuovamente fino a 2/3 della tensione applicata tra i piedini 1 e 8 del 555, quindi il ciclo ricomincia. Per effetto di ciò abbiamo l'uscita OUT dell'integrato che si mantiene a livello alto nel periodo che serve al condensatore Ct affinché la tensione ai suoi capi passi da 1/3 a 2/3 della tensione di alimentazione (parliamo di 5Vcc, quindi il passaggio è tra 1,67 e 3,33 volt...) e a livello basso (zero volt circa) nel tempo che trascorre da quando il Ct, in fase di scarica, vede scendere la tensione ai propri capi da 2/3 della tensione

di alimentazione ad 1/3. I tempi dipendono dai valori delle resistenze R1, R2 ed R7 e sono tanto più simili quanto più R2 è piccola rispetto alla somma di R1 ed R7: infatti la carica avviene attraverso tutte e tre le resistenze mentre la scarica avviene solo attraverso R2. Va anche detto che la scarica al valore di soglia di 1,67 V è, parità di resistenza, più rapida della carica a 3,33 V, quindi per un certo valore di R2 la durata dell'1 logico all'uscita del 555 è pari a quella dello stato zero. Il ciclo descritto prosegue fin quando si mantiene la tensione di alimentazione, ovvero non si sposta la levetta del doppio deviatore. Concludiamo la descrizione del circuito dicendo che, trattandosi di una scheda di prototipazione, abbiamo previsto sia due connettori per l'alimentazione e i collegamenti di uscita e ingresso di trigger, sia una fila di piazzole a passo 2,54 mm che permettono di montare un pin strip maschio per innestare la scheda su hardware esistente. Il connettore dell'alimentazione è quello che porta 5V, GND e IN (l'ingresso TTL di trigger nella modalità monostabile) mentre l'altro, oltre a ripetere i 5V trasporta le uscite, che sono quella del 555 (OT) e il collettore del transistor T2 che realizza l'uscita di potenza (OC). Sul pin strip troviamo invece tutti i contatti dei due connettori.

REALIZZAZIONE PRATICA

Spiegato il funzionamento dei due circuiti ottenibili dalla breakout board, vediamo come realizzare quest'ultima a partire dal circuito stampato, ottenibile per fotoincisione seguendo le due tracce (il c.s. è a doppia faccia) dei lati rame scaricabili sotto forma di file immagine dal nostro sito www.elettronica.in. it insieme ai file del progetto. Ottenuta la basetta e procurati i

componenti, tutti per montaggio superficiale, potete iniziare a disporli ognuno con i terminali in corrispondenza delle rispettive piazzole (che avrete avuto cura di spalmare con pasta flussante allo scopo di agevolare le saldature) stagnandoli poi a partire dalle resistenze e dai condensatori non elettrolitici, per proseguire con il 555, i due transistor e i due connettori miniatura. Terminate con il doppio deviatore, gli elettrolitici e il trimmer. I jumper si realizzano unendo le rispettive piazzole con una goccia di stagno. L'orientamento dei componenti è visibile nel piano di montaggio pubblicato in queste pagine. Ricordate che il montaggio di componentistica SMD richiede un po' di manualità e una buona vista da vicino: se non l'avete procuratevi una lente d'ingrandimento, che vi aiuterà a verificare che i componenti più piccoli siano saldati correttamente e che non vi siano baffi di stagno a unire terminali dell'integrato o pad di componenti vicini. Alle doti "fisiche" dovete aggiungere del filo di lega saldante da 0,5 mm di diametro, una pinzetta per posizionare i componenti e un saldatore a punta finissima da non più di 20 watt. ■

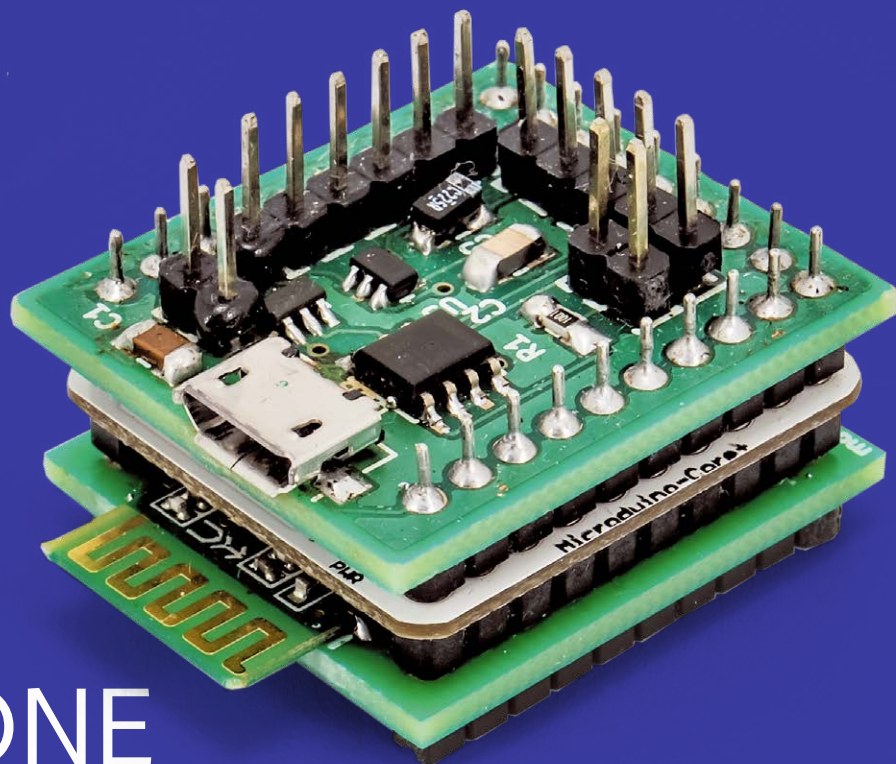


per il MATERIALE

Tutto il materiale può essere acquistato presso Futura Elettronica. La breakout con timer NE555 (cod. BREAKOUT016) viene fornita assemblata e compresa di cavetti al prezzo di Euro 11,00.

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

Monitoriamo
temperatura, umidità
e altri parametri
pubblicandone i valori
su un servizio web
pubblico. Prima puntata.



STIMA: MISURAZIONE DI PARAMETRI METEO E AMBIENTALI

di PAOLO PATRUNO E DANIELE CASTELLARI

Qual è il livello dell'inquinamento atmosferico vicino alla vostra casa o al luogo di lavoro? Quali la temperatura e l'umidità? Immaginate di poterle conoscere, condividerle immediatamente e confrontarle con altri luoghi della vostra città, in tempo reale... Potrebbe, e come, questa informazione, migliorare la qualità del nostro ambiente? Vediamo come connettere i dati, la conoscenza e le persone, per costruire utili indicatori aperti e applicazioni distribuite. Un termine che ricorre spesso ultimamente è Smart City, che significa territorio urbano che permette di soddisfare le esigenze dei cittadini, delle imprese e delle istituzioni, mediante l'ausilio di strumenti innovativi e partecipazione attiva. L'uso di tecnologie open e di sensoristica a basso costo aiuta lo sviluppo di questa rete. In quest'ottica



si colloca il progetto RMAP, che proponiamo di sfruttare mediante un sistema modulare basato su unità Bluetooth equipaggiate con sensori, che tramite smartphone accedono a Internet e si collegano al sito web corrispondente.

IL PROGETTO RMAP

Il progetto RMAP (Rete di Monitoraggio Ambientale Partecipativo) è promosso da istituzioni come ARPA - Agenzia Regionale per La Prevenzione e L'Ambiente dell'Emilia Romagna Servizio Idro-Meteo-Clima, ARPA Agenzia Regionale per La Prevenzione e L'Ambiente del Veneto, Dipartimento informatica Università di Bologna e Cineca (Consorzio Interuniversitario per il supercalcolo e l'innovazione tecnologica). Il progetto si pone come obiettivo di:

- raccogliere e distribuire dati ambientali rilevati dai cittadini;
- rendere disponibili questi dati anche ai servizi meteorologici, alle agenzie di prevenzione ambientale, alla Protezione Civile e ad istituti di ricerca;
- fornire feedback ai fornitori di dati in modo che essi abbiano servizi per testare e migliorare la qualità dei dati;
- divulgare temi scientifici e sensibilizzare il pubblico sui temi ambientali;
- coinvolgere scuole e università a scopi formativi;
- creare un circolo virtuoso tra enti formativi, pubbliche amministrazioni, aziende private e cittadini.

RMAP, di fatto, comprende una serie di standard atti a favorire lo scambio di dati ambientali tra istituzioni e tra cittadini e consiste in un insieme di specifiche riguardanti:

- protocollo di rilevamento

dati (www.raspibo.org/wiki/index.php/Gruppo_Meteo/DisciplinareStazione);

- collocazione sensori;
- accuratezza dei sensori;
- elaborazioni sui campionamenti;
- sistema per lo scambio dati (www.raspibo.org/wiki/index.php/Gruppo_Meteo/RFC-rmap);
- protocolli di trasmissione;
- formati dati;
- metadati.

I moduli Stima che qui presentiamo, aderiscono a tale standard e permettono di pubblicare i dati sul server <http://rmap.cc>; sono stati sviluppati all'interno delle attività di RasiBo, gruppo di maker della provincia di Bologna. Misurare temperatura e umidità non è sicuramente una novità, ma il progetto che presentiamo si distingue per l'approccio e le soluzioni tecniche adottate. La progettazione modulare per espandibilità e versatilità, l'adozione di protocolli e formati standard che lo rendono compatibile e integrabile, l'attenzione a concetti scientifici per il campionamento, le soluzioni completamente open, lo rendono speciale per chi vuole qualcosa più di un "giocattolo tecnologico" per fare misurazioni ambientali.

I MODULI STIMA

In questa prima puntata presentiamo il modulo Stima Bluetooth, che si appoggia a uno smartphone o a un PC dove un apposito software si occuperà dell'invio dei dati al server rmap.cc; il modulo acquisisce i dati e li invia tramite Bluetooth allo smartphone o al PC. Sul server i dati vengono archiviati e sono presenti alcuni tools per la visualizzazione grafica; il software utilizzato sul server è completa-

mente open. La suite di moduli Stima è molto ampia e permette di soddisfare differenti esigenze: vi sono moduli collegabili con cavo ethernet e alimentati via PoE, moduli alimentati a batteria che comunicano con un modulo radio e moduli basati su Raspberry Pi, che oltre a raccogliere dati collegati alla GPIO fungono da veri e propri server per l'archiviazione e la visualizzazione dei dati. Nelle prossime puntate ne presenteremo alcuni, partendo da quello che comunica (ed è alimentato) tramite ethernet, nonché quello con Raspberry Pi, che svolge tutte le funzioni di un server RMAP per l'acquisizione, archiviazione, visualizzazione e upload dei dati.

SENSORI

I sensori utilizzati per acquisire i parametri ambientali devono essere compatibili con il bus I²C, altrimenti si può risolvere il problema utilizzando un microcontrollore che adatta le letture (analogiche o digitali) e le elaborazioni (contatori, medie ecc.) rendendole disponibili su registri interrogabili tramite I²C-Bus. Ricordiamo che il protocollo I²C prevede l'utilizzo di un bus formato da due linee bidirezionali, chiamate SCL e SDA, le quali trasportano, rispettivamente, il segnale di sincronizzazione (clock) e i dati (la linea è bidirezionale). Abbiamo scelto il bus I²C in quanto:

- è diventato lo standard per una serie di integrati tra cui i sensori;
- vi si possono collegare fino a 127 dispositivi;
- la comunicazione è bidirezionale (read e write) con velocità sufficienti per i nostri scopi;
- la lunghezza operativa dei cavi è adeguata al nostro utilizzo (anche alcune decine di metri).

I sensori vengono interrogati tramite remote call procedure in formato json (json-rpc). Quando richiesto, tutti i sensori vengono interrogati "in parallelo" ossia i sensori vengono impostati e configurati all'accensione, poi, a richiesta vengono attivati e gli viene impartito il comando di lettura. Il driver del sensore attende il tempo necessario per avere la misura disponibile; siccome non tutti i sensori leggono nello stesso tempo, si attende il tempo necessario al sensore più lento, cosicché si effettuano tutte le letture. In questo modo si riescono a campionare tutti i sensori solitamente entro i 3 secondi e, considerando i tempi per la pubblicazione dei relativi dati sul server, si riescono a campionare i sensori ogni 5 secondi. Tenendo i sensori normalmente in modalità sleep, si riducono i consumi. Ogni sensore può restituire valori multipli (ad esempio temperatura e umidità). Di seguito, vedete un esempio di interrogazione e risposta di un sensore di temperatura:

```
(SEND): {"jsonrpc": "2.0",
"method": "getjson", "params":
{"node": 1, "type": "TMP",
"driver": "I2C", "address": 72},
"id": 0}
(RECEIVE): {"jsonrpc": "2.0", "result":
":{"B12101":30633},"id":0}
```

B12101 indica che il numero che segue è una temperatura espressa in centesimi di gradi Kelvin, quindi 33,18 °C. Le "remote procedure" disponibili sono documentate sul wiki del progetto, accessibile da Internet alla pagina www.raspibo.org/wiki/index.php/Gruppo_Meteo/RemoteProcedure.

JSON RPC

Il protocollo JSON-RPC (Remote Procedure Call in formato json) funziona inviando una richiesta a un server che lo implementa;

il client in questo caso è tipicamente un software che intende richiamare un singolo metodo su un sistema remoto. Possono essere passati parametri multipli al metodo remoto come array di oggetti; il metodo stesso, poi, può esso stesso ritornare parametri multipli. Un metodo remoto è invocato inviando una richiesta a un servizio remoto usando HTTP o un socket TCP/IP o una porta seriale, come trasporto. Tutti i parametri trasferiti di ogni tipo sono singoli oggetti, serializzati usando JSON. Una richiesta è una chiamata a uno specifico metodo disponibile sul sistema remoto; deve contenere tre specifiche proprietà:

- *method*, che è una stringa col nome del metodo da invocare;
- *params*, che è un array di oggetti come parametri al metodo invocato;
- *id*, che è un valore di qualsiasi tipo, usato per riferire la risposta alla richiesta a cui si sta rispondendo.

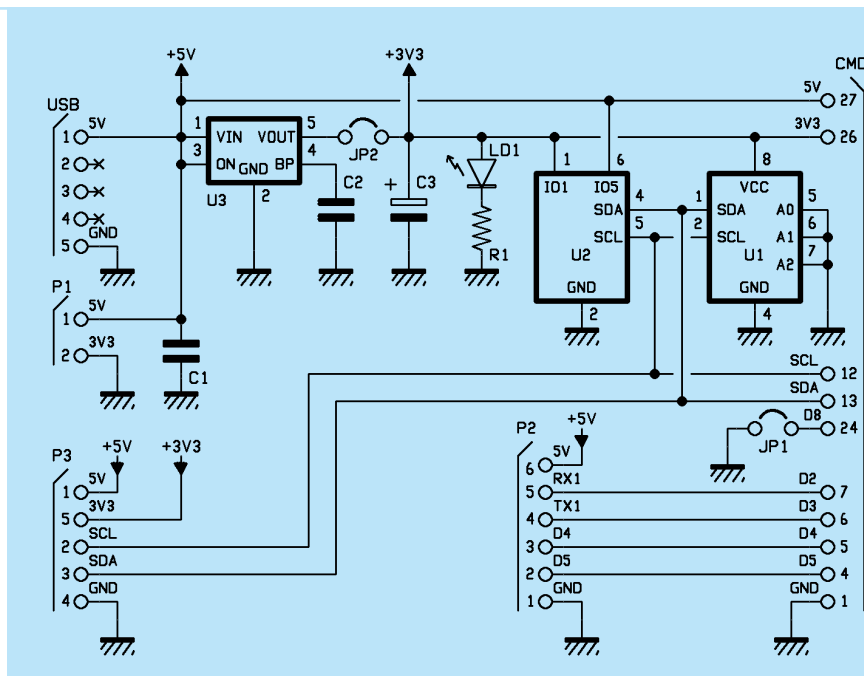
Il server che riceve la richiesta deve rispondere con una risposta valida a tutte le richieste ricevute. Una risposta deve contenere le

proprietà descritte di seguito.

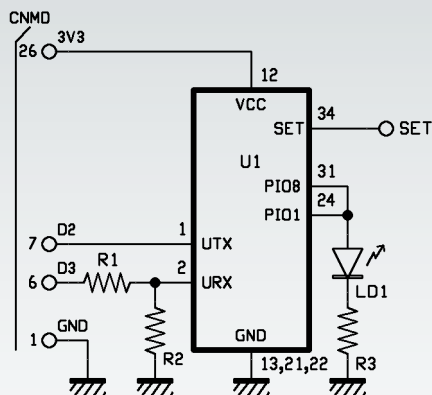
- *result*: i dati ritornati dal metodo invocato. Se c'è un errore invocando il metodo, il valore deve essere null.
- *error*: uno specifico codice di errore se l'invocazione del metodo ha dato luogo a un errore, altrimenti null.
- *id*: l'id della richiesta a cui si sta rispondendo.

TRASPORTI

Il concetto di trasporto in Stima è simile, ma non rigidamente aderente ai concetti del modello ISO-OSI e possono essere attivi o passivi. Nel caso dei trasporti passivi, il compito è fornire un canale logico-affidabile di comunicazione end-to-end per fornire servizi al sopostante livello che in Stima è JsonRPC. Nel caso dei trasporti attivi, corrisponde al protocollo (Session Layer) per la pubblicazione dei dati su un server (broker). In pratica i trasporti passivi permettono di eseguire procedure remote codificate in formato json specifiche dell'implementazione Stima; quelli attivi permettono la pubblicazione su server (broker) dei messaggi aderenti allo standard R-MAP.



[schema elettrico
STIMA-Bluetooth]



Trasporti Passivi sono:

- serial;
- TCP/IP;
- Bluetooth (Serial Port Profile);
- NRF24.

Trasporti attivi sono invece MQTT e AMQP.

UN SISTEMA MODULARE

Passiamo ora a vedere come è stato progettato il sistema sul piano hardware; daremo un'occhiata generale per poi arrivare ai dettagli e capire meglio come funziona. Il progetto è stato sviluppato utilizzando il software open source di progettazione elettronica KiCad (gli abbiamo dedicato un corso nei fascicoli dal n° 184 al 189); per l'apertura dei file, disponibili su github all'indirizzo <https://github.com/r-map>) è necessaria l'ultima versione ad oggi rilasciata di KiCad, in quanto il formato dei file è cambiato rispetto alle versioni precedenti. A livello hardware abbiamo progettato differenti moduli: un modulo è composto da alcune board ed è collegabile a uno o più sensori. I moduli si differenziano tra loro soprattutto per la modalità con cui comunicano con il server; i dati possono essere inviati via Bluetooth, oppure via LAN, rete mobile e via dicendo.

MODULO STIMA BLUETOOTH

Il modulo è composto da tre board, che sono una Microduino core+, una Stima-I2C e una Stima-Bluetooth. E' possibile collegare più sensori tra quelli al momento supportati (sensori per la temperatura, umidità, pressione, precipitazione e vento); una breakboard per alcuni sensori verrà presentata nelle prossime puntate; per il momento sulla board Stima-I2C è già presente un sensore di temperatura con cui poter fare tutti i test.

BOARD MICRODUINO CORE+

Il cuore di ogni modulo è la board Microduino core+, basata su un microcontrollore Atmega644P; ha 64 k di flash necessari a contenere tutto il software per la lettura dei sensori e la gestione dei protocolli di comunicazione, 4k di SRAM, due porte seriali che utilizzeremo per la comunicazione wireless e il debug. Questa board, nata da un progetto su Kickstarter di microduino.cc è molto semplice infatti vediamo oltre al micro, un quarzo, qualche condensatore ed un led, insomma il minimo necessario che viene integrato di volta in volta con l'uso di shield. Notate che la cpu viene alimentata a 5 V semplicemente con una porta usb del pc o batteria esterna come quelle che si usano per gli smartphone; dato che alcune periferiche vanno alimentate a 3.3 V è necessario utilizzare uno shield appositamente progettato per fornire questa tensione. Lo shield Microduino-USBTTL verrà utilizzato solo per caricare il firmware nella fase propeedeutica e lo potremo poi riutilizzare per la programmazione di altri moduli Stima. E' utile anche per ottenere tutta la messaggistica di debug inviata sulla porta seriale 0. Un consiglio, caricate il firmware da subito ed una volta terminata la saldatura sarete già pronti per i test!

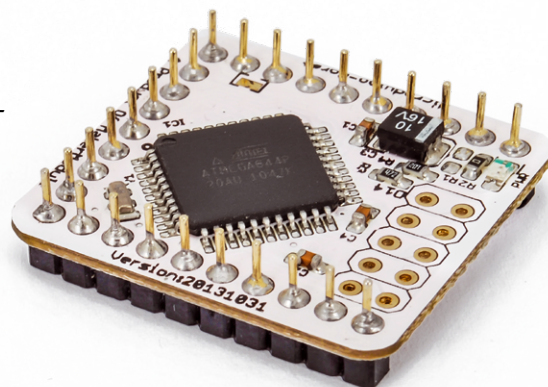
BOARD STIMA-I2C

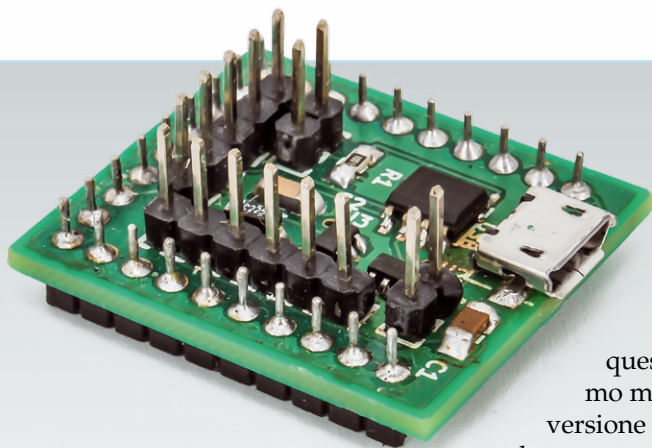
Questa board, assieme a Microduino core+, è utilizzata in tutte le configurazioni di Stima perché a bordo si trovano connettori per l'alimentazione, il bus I²C ecc. Abbiamo detto che la board può essere alimentata a 5V tramite il connettore micro USB, ma se lo preferite è presente anche il connettore P1, al quale potete collegare un alimentatore stabilizzato.

Il connettore P3 permette di collegare al modulo altre periferiche I²C; su di esso sono presenti entrambe le tensioni di alimentazione. Se si vuole connettere un sensore remoto è consigliabile collegarlo all'uscita a 5 volt e utilizzare uno stabilizzatore a 3,3 V vicino al sensore stesso, per evitare cadute di tensione lungo la linea.

Il connettore P4 viene utilizzato per forzare, al boot, l'aggiornamento delle variabili di configurazione memorizzate sulla EEPROM del micro. P2 aggiunge la possibilità di collegare alla board alcuni relé, rendendo il tutto predisposto per svolgere alcune funzioni di domotica.

Per alimentare alcune periferiche come, ad esempio, il modulo Bluetooth, e' necessaria una tensione di 3,3 V; per questo motivo abbiamo utilizzato uno stabilizzatore con output a 3,3 volt di ridotte dimensioni REG113NA





(U3), lo stesso utilizzato anche dagli shield di Microduino. All'uscita di tale integrato è collegato anche un LED (D1) che ne indica il regolare funzionamento. Per proteggere i componenti da scariche elettrostatiche (tipicamente quelle dei fulmini) sono stati collegati ai pin del bus I²C e delle linee di alimentazione alcuni diodi di protezione (scaricatori) racchiusi nel componente ESDA6V1-5SC6 (U2) i quali si occupano di assorbire eventuali eccessi di tensione.

Per darvi la possibilità di testare subito la board collegandola al PC o allo smartphone, è già presente un sensore di temperatura I²C modello TMP275(U4); già dalla prima connessione, in attesa di ampliare il vostro set di sensori, avrete la possibilità di verificare che tutto il sistema sia regolarmente funzionante. Non collegate contemporaneamente questo shield con lo shield Stima-I2C in quanto entrambi alimentano la linea a 3.3V e si potrebbero causare danni.

BOARD STIMA-BLUETOOTH

Vediamo più in dettaglio lo schema del modulo Stima-Bluetooth: si tratta di uno shield equipaggiato con una board preassemblata HC05 (lo shield è compatibile anche con la versione HC06) e pochi componenti aggiuntivi. Il modulo HC05 trasporta su Bluetooth la comunicazione seriale (profilo SPP) rendendo Stima capace di comunicare sia col PC che con lo smartphone; in

questo modo possiamo montare Stima in versione mobile sulla bicicletta, oppure stabilmente sul terrazzo. L'HC05 richiede una tensione di alimentazione di 3,3 V a differenza del microcontrollore che è alimentato a 5 V perciò è sempre necessario abbinarlo allo shield Stima-I2C o USBTTL. Il microcontrollore invia i dati da pubblicare sulla seriale 1, lasciando disponibile per il debug e la programmazione la porta seriale 0. Sulla board è presente anche un LED che lampeggia per indicare lo stato della connessione Bluetooth oltre ad un partitore di tensione che riduce a 3,3 V i segnali in arrivo dall'ATmega. Volendo è possibile realizzare un prototipo come mostrato nella Fig. 1 utilizzando una breakout board per hc05 e una SparkFun Digital Temperature Sensor Breakout - TMP102.

REALIZZAZIONE PRATICA

Per rendere le loro dimensioni compatibili con Microduino, le board montano alcuni componenti SMD il cui montaggio richiede un po' di manualità ed esperienza, oltre ad attrezzatura adeguata. Procurati tutti i componenti, vi consigliamo di iniziare con il connettore USB, che richiede un posizionamento preciso ed ha pin al limite del contenitore; proseguite poi con i circuiti integrati e i componenti passivi. A completamento delle schede vanno poi montati i connettori THT, ponendo attenzione ai connettori P1, P2, P3, P4 di Stima-I2C, che vanno saldati sulla faccia opposta a quella che ospita i componenti SMD. La saldatura della board Bluetooth non richiede particolari indicazioni; sebbene non sia strettamente necessario, vi consigliamo di saldare tutti i pad della scheda HC05 per renderla più resistente meccanicamente, dato che l'antenna sporge rispetto al profilo di Microduino. Completate la saldatura delle board e verificate attentamente che non

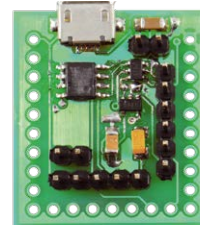
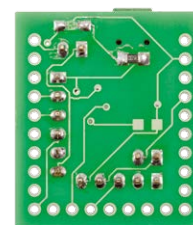
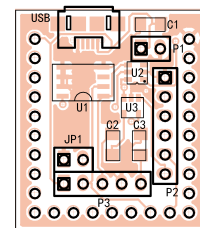
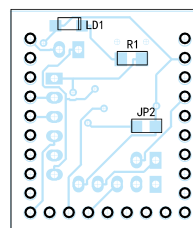
[piano di montaggio STIMA-I2C]

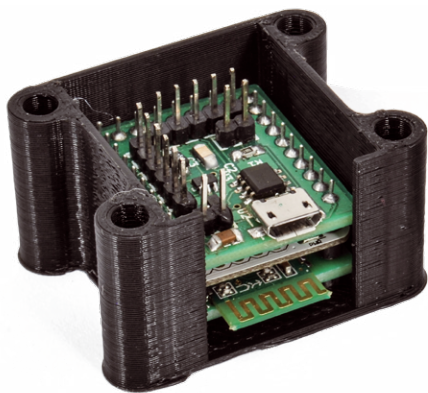
Elenco Componenti:

R1: 150 ohm (1206)
C1: 1 μ F ceramico (1206)
C2: 10 nF ceramico (1206)
C3: 2,2 μ F 16 VL tantalio (1206)
LD1: LED giallo (1206)
U1: TMP275AID
U2: ESDA6V1-5SC6
U3: REG113NA

Varie:

- Connettore Micro-USB da CS
- Pin strip femmina torniti 27 vie
- Strip maschio 5 vie
- Strip maschio 6 vie
- Strip maschio 2 vie (2 pz.)
- Jumper
- Circuito stampato S1232





vi siano “baffi” di lega saldante a provocare connessioni indesiderate. Montati gli shield, impilateli mantenendo il microduino core+ al centro. Aggiungete anche lo shield USBTTL per caricare il firmware e poter leggere i messaggi di debug. Una volta caricato il firmware potete rimuovere questo shield e saldare il ponticello JP2 sullo shield Stima-I2C per attivare l'alimentazione a 3.3V del bus microduino. Non collegate contemporaneamente lo shield USBTTL con lo shield Stima-I2C col ponticello saldato in quanto entrambi alimentano la linea a 3.3V e si potrebbero causare danni. Abbiamo progettato un interessante box per il modulo Stima-Bluetooth e i sensori (presentati nella prossima puntata); è composto da alcuni elementi impilabili a seconda delle esigenze. Ora lo presentiamo nella versione con un comodo attacco a elastico

da usare ad esempio sul manubrio della bici per monitorare il percorso delle nostre escursioni. Sono disponibili i file .stl per stamparlo con una stampante 3D: i file; i file (il progetto è di Mirco Bergamini) si scaricano da <https://github.com/r-map/rmap/releases>.

IL FIRMWARE RMAP

I dettagli del firmware dei moduli Stima verrà esaminato nella prossima puntata. Per ora basti sapere che esiste un file di configurazione per le opzioni a tempo e che altre configurazioni sono salvate runtime su eeprom tramite json-rpc. Seguendo le istruzioni seguenti vi troverete installati tutti i file e le configurazioni necessarie per far funzionare l'IDE originale di arduino anche con le board microduino; questa è una grandissima comodità! Qui parleremo, invece, del software da installare nel PC per interfacciarsi con il server rmap.cc, chiamato Applicazione RMAP. Questo programma interagisce con l'utente tramite interfaccia grafica e permette la pubblicazione di dati ambientali sul server rmap.cc. I dati possono essere rilevati manualmente e a vista, oppure con il dispositivo Stima-Bluetooth in modo automatico. L'app guida attraverso una

sequenza di fasi:

- identificazione della propria posizione;
- inserimento dei dati manuali;
- connessione e attivazione del dispositivo Stima-Bluetooth per il rilevamento automatico dei dati;
- gestione della coda dei dati rilevati e connessione e invio dei dati al server.

L'app ha due modalità di funzionamento: una interattiva e una in background. La modalità di esecuzione in background non richiede la presenza o l'intervento dell'utente e può essere concorrente all'esecuzione di altre applicazioni. Nello sviluppo dell'applicazione RMAP sono stati utilizzati Django e la libreria Kivy: Django è un web framework open source per lo sviluppo di applicazioni web, scritto in linguaggio Python; fornisce funzionalità che facilitano lo sviluppo rapido di applicazioni per la gestione di contenuti. Django fornisce una soluzione integrata di amministrazione dei contenuti che può essere inclusa come parte di ogni sito basato su Django. Le principali funzionalità incluse sono:

- astrazione del database relazionale ad oggetti;
- robusta API per la gestione del database;
- sistema template basato su tag con ereditarietà dei template;
- gestore di URL basate su espressioni regolari;
- sistema “middleware” per lo sviluppo di funzionalità aggiuntive; ad esempio, componenti middleware che forniscono caching, compressione dell'output, normalizzazione dell'output, protezione CSRF e supporto per la sessione sono inclusi nella distribuzione principale di Django;
- supporto per localizzazione, incluse traduzioni dell'interfaccia

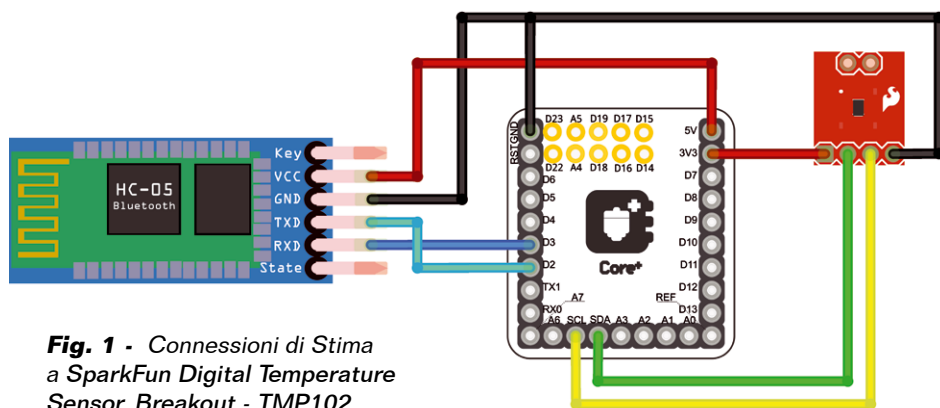
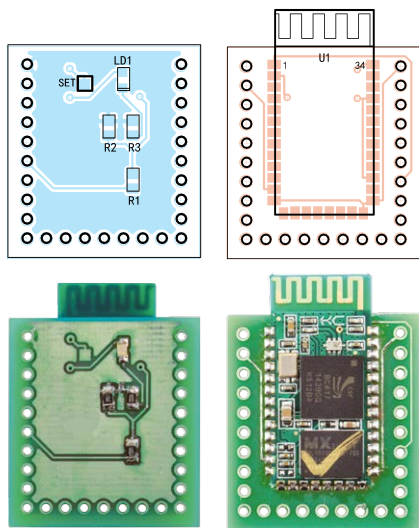


Fig. 1 - Connessioni di Stima a SparkFun Digital Temperature Sensor. Breakout - TMP102.

[piano di montaggio STIMA-Bluetooth]



Elenco Componenti:

R1: 8,2 kohm (0805)
R2: 15 kohm (0805)
R3: 150 ohm (0805)
LD1: LED rosso (0805)
U1: Modulo HC05 o HC06

Varie:

- Pin strip femmina torniti 27 vie
- Circuito stampato S1233

amministrativa, in molte lingue;

- documentazione accessibile dall'interfaccia amministrativa (compresa documentazione automaticamente generata dai modelli e librerie template aggiunte dalle applicazioni);
- sistema di gestione degli utenti e loro autenticazione nell'applicazione web;
- sistema per la creazione e la validazione di form HTML.

Quanto a Kivy, è una libreria Python per lo sviluppo di applicazioni su device mobili e/o multi-touch con un'interfaccia utente naturale (NUI); funziona su Android, iOS, Linux, OS X, e Windows. Distribuito con la licenza MIT, Kivy è un software libero

e open source. Cellulari e tablet hanno portato con essi un cambiamento drammatico nell'uso delle applicazioni; la compatibilità è diventata essenziale e ha aumentato il tipo di interazione che gli utenti si aspettano: gesti, multi-touch, animazioni e penne magiche.

Kivy è una soluzione open source Python che copre queste esigenze di mercato con un approccio di sviluppo facile e rapido da imparare. Kivy sta crescendo rapidamente e guadagnando attenzione come alternativa alle piattaforme di sviluppo standard.

UTILIZZO DI RMAP

Eseguiamo l'applicazione e vediamo come si presenta; l'interfaccia è schietta, per poter essere eseguita in ambienti differenti e bisogna dedicarsi un poco per appropriarsi di tutte le funzioni disponibili. In alto abbiamo i pulsanti di navigazione tra le pagine e quelli per attivare alcune funzioni; al centro le funzioni specifiche della pagina corrente e in basso i messaggi di stato e di errore, oltre ad alcune funzioni chiave da eseguire prima di abbandonare quella pagina. Per pubblicare i dati sul server RMAP.cc bisogna prima di tutto registrarsi: facendo clic sul pulsante Registrazione si apre la pagina web della registrazione, localizzata a <http://rmap.cc/registrazione/register/>.

Completata la registrazione, sarete in possesso di uno user e di una password con cui accedere; entrate nel menu Impostazioni e in "Rmap" inserite "MQTT user" e "MQTT password" ottenuti durante la registrazione a rmap.cc. Per utilizzare Stima-Bluetooth il dispositivo va associato col telefono/PC come qualsiasi altro dispositivo Bluetooth. Seguite le istruzioni di installazione specifiche per la

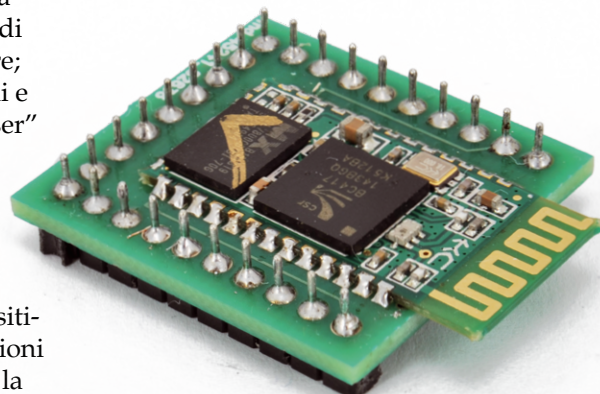
vostra piattaforma. Iniziamo la navigazione delle varie pagine dell'App presentando, per il momento, solo una sintesi delle funzioni; un manuale d'uso più dettagliato è consultabile nella prima pagina della stessa App.

Pagina "Avvia"

Il pulsante "Visualizzazione grafici" apre il browser con una URL per la visualizzazione di un grafico con i dati delle ultime ore.

Pagina "Posizione"

Qui viene visualizzata una mappa, contenente un simbolo di forma circolare al centro, la cui funzione è di "mirino". È possibile muoversi sulla mappa e "zoomare" per portare il mirino precisamente sulla nostra posizione e una volta trovata la posizione corretta si può salvare col pulsante "Salva posizione". Se disponiamo di un GPS (come nell'uso di uno smartphone android) possiamo, dopo averlo abilitato nel menu di impostazioni del telefono, attivarne l'uso nella nostra applicazione col bottone "Avvia GPS". Se il GPS è attivo, il "mirino" si porta continuamente sulla posizione rilevata dal GPS. Con il pulsante "Salva posizione" è possibile impostare la posizione segnata dal "mirino" come nostra "Location" e un "segnaposto" sulla mappa ne visualizzerà



Il server **rmap.cc**

Il servizio RMAP permette di inviare al server **rmap.cc** dati che poi vengono, ad esempio, visualizzati su grafici come quello visibile in **Fig. 2**, dove appare un tipico grafico della temperatura ambiente rilevata.

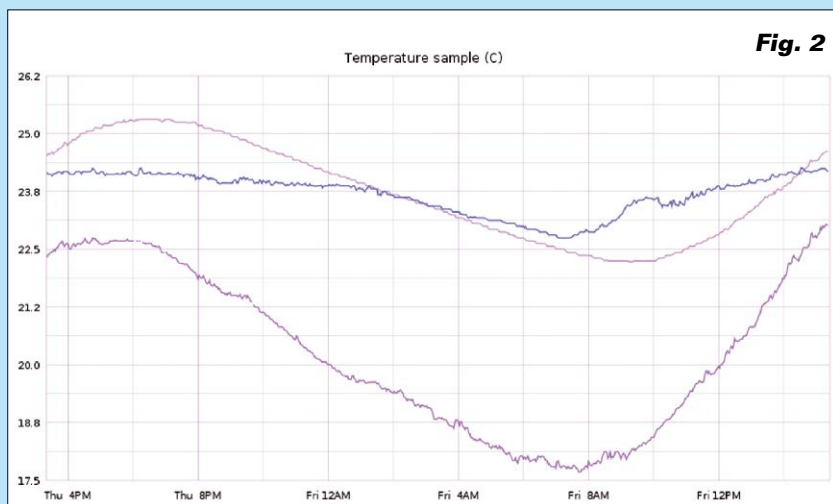


Fig. 2

Nella **Fig. 3** è invece visibile una schermata contenente una mappa con i punti stazione, cliccando sui quali è possibile vedere i rispettivi parametri inviati.

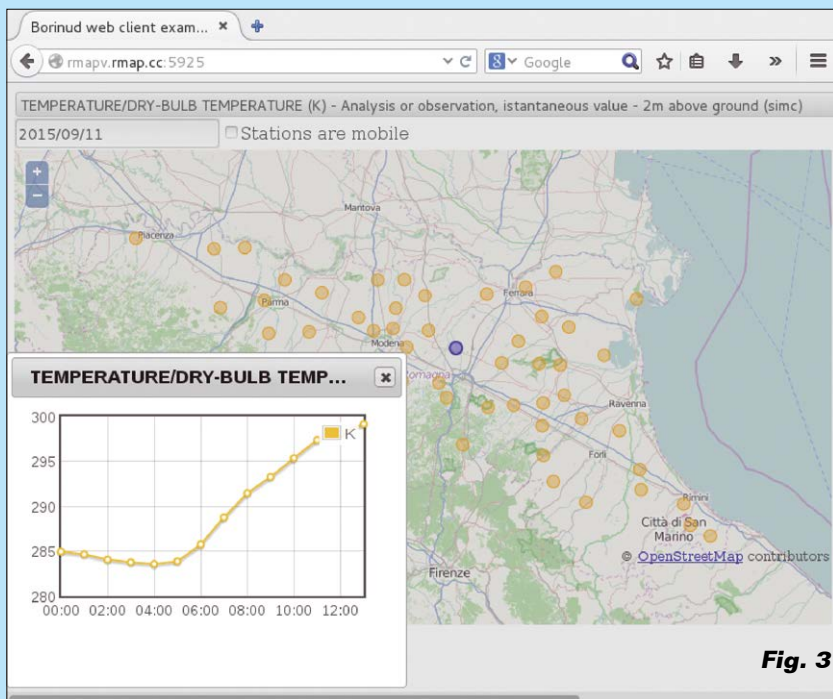


Fig. 3

la posizione permanentemente salvata. Se nelle impostazioni è stata selezionata una stazione di tipo mobile, allora è possibile selezionare il successivo gruppo di misure che verranno effettuate come un "viaggio" dove la "location" viene costantemente aggiornata dal GPS; il pulsante "Inizia Viaggio" serve a questo scopo.

Pagina "Inserisci dati"

In questa pagina è possibile inserire i dati rilevati con osservazioni a vista e quindi manualmente relativamente alla "location" corrente definita dal GPS del cellulare. Per ulteriori approfondimenti fate riferimento al manuale d'uso dell'applicazione.

Pagina "Dati automatici"

Il pulsante "Configura" permette di salvare sulla EEPROM del modulo Stima-Bluetooth le configurazioni necessarie al funzionamento del modulo, quali la definizione della sensoristica presente sul modulo; questa operazione va fatta una tantum dopo il caricamento del firmware sul modulo Stima-Bluetooth. Il bottone "Avvia trasporto" permette di verificare la possibilità di creare il link radio tra il nostro PC/smartphone e il modulo stima-Bluetooth e lo stato viene visualizzato in fondo allo schermo. Il pulsante "Prendi dati" permette di effettuare un campionamento dei sensori e acquisirlo per l'invio al server. Il pulsante "Campionamento ON" attiva il campionamento automatico dei sensori alla frequenza impostata nella configurazione. I dati rilevati dai sensori vengono visualizzati al centro della pagina.

Pagina "Pubblica"

In questa pagina vengono visualizzati i dati che sono accodati per la pubblicazione sul server. Infatti è possibile effettuare misure anche senza connessione a Internet, memorizzando i dati sul proprio dispositivo; quando la connessione a Internet tornerà attiva, verranno automaticamente inviati al server. I dati hanno un proprio timestamp, quindi l'ora della misura sarà sempre quella corretta. Il pulsante "Connetti" attiva i tentativi di connessione al server; lo stato della connessione viene visualizzato in fondo allo schermo. Il pulsante "Esegui in background" è importante in quanto permette di eseguire l'applicazione RMAP in background, ossia liberando lo schermo (in particolare di uno smartphone) per fare tutte le altre operazioni cui è adibito il nostro dispositivo. Alla sua attivazione,

sullo smartphone sarà presente un servizio "Rmap background" dal nome "station". Nelle notifiche sarà presente la messaggistica di diagnostica relativa allo stato del link Bluetooth e alla connessione al server.

COMPILAZIONE FIRMWARE STIMA-BLUETOOTH

Passiamo alla compilazione dello sketch per Microduino core+:

- installare l'IDE Arduino 1.6.5 da <https://www.arduino.cc/en/Main/Software> o tramite la propria distribuzione;
- scaricare l'ultima versione del software stima (file *stima-sketchbookbluetooth*) da <https://github.com/r-map/rmap/releases>;
- scompattare il file zip;
- aprire l'IDE Arduino e in File -> Impostazioni -> percorso della cartella degli sketch selezionare la cartella sketchbook appena scompattata dal file scaricato;
- chiudere e riaprire l'IDE;
- aprire il menu Strumenti e scegliere:
 1. Scheda: Microduino Core+ (644pa);
 2. Processore: ATmega644pa@16M5V;
 3. Porta: (quella disponibile);
- dal menu Sketch impartire il comando Verifica e compila.

Ora per uploadare il firmware sul microprocessore dovete collegare alla Microduino core+ lo shield USB/TTL e connettere al PC il tutto; a questo punto, nell'IDE di arduino impartite il comando Sketch -> Carica.

INSTALLAZIONE STIMA RMAP

L'installazione su Android è semplicissima: è sufficiente ricercare tra le app su google play "rmap" (<https://play.google.com/store/apps/details?id=org.test.rmap>), procedere all'installazione e accoppiare

il dispositivo HC-05 inserendo come pin "1234". L'installazione in ambiente Linux richiede la disponibilità di alcuni pacchetti e del comando pip. Prima di tutto bisogna installare Kivy seguendo le istruzioni che trovate alla pagina <http://kivy.org/docs/installation/installation-linux.html>. Il comando pip nelle distribuzioni Linux più diffuse lo si ottiene installando il pacchetto python-pip; per installare da utente non privilegiato l'ambiente RMAP si può usare virtualenv e pip; da terminale, bisogna eseguire:

```
virtualenv --system-site-packages
rmap
source rmap/bin/activate

pip install django
pip install configobj
pip install rmap
```

Per accoppiare il dispositivo Stima-Bluetooth cosigliamo di utilizzare Blueman-manager contenuto nel pacchetto Blueman; seguendo pochi passi, con l'interfaccia grafica risulta facile accoppiare il dispositivo HC-05 inserendo come pin "1234"; il device RFCOM0 viene utilizzato per la comunicazione seriale. Ora bisogna attivare l'interfaccia utente grafica:

```
source rmap/bin/activate
rmapgui
```

Windows

L'installazione in windows è molto semplice in quanto il file da scaricare è autoscompattante e comprende tutto l'ambiente necessario a Rmap. Scaricare quindi il file *rmapwindows* da <https://github.com/r-map/rmap/releases> ed eseguirlo per scompattarlo. Tramite le apposite interfacce di Windows procedere all'accoppiamento del dispositivo bluetooth HC-05 inserendo come pin "1234" e richiedere la creazione della relativa porta seriale COM13 Per far partire

l'applicazione a questo punto basterà eseguire il file *rmapgui* contenuto nella cartella *rmap*.

Mac OS X

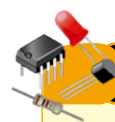
Per l'installazione in MacOS X, prima di tutto bisogna installare Kivy sul sistema operativo di Apple seguendo le istruzioni reperibili alla pagina web <http://kivy.org/docs/installation/installation-macosx.html> e poi installare gettext da <http://www.ellert.se/twain-sane/>. Fatto ciò, si impartisce:

```
kivi -m pip install django
kivi -m pip install configobj
kivi -m pip install --upgrade rmap
```

Dopo aver accoppiato il dispositivo Bluetooth inserendo come pin "1234", si può attivare il programma:

```
kivy /Applications/Kivy.app/Contents/Resources/venv/bin/rmapgui
```

Con questo abbiamo concluso; riprenderemo il discorso nella prossima puntata. ■



per il MATERIALE

Tutti i componenti utilizzati in questo progetto sono di facile reperibilità. I master dei circuiti stampati possono essere scaricati dal sito della rivista. La board Microduino Core ATMEGA644PA (cod. MDUINOCORE+) è in vendita al prezzo di Euro 15,50, il modulo Bluetooth Transceiver (cod. HC05) costa Euro 9,00 e il sensore di temperatura (cod. TMP102) è disponibile a Euro 10,50. Questi ultimi componenti si possono acquistare presso Futura Elettronica.

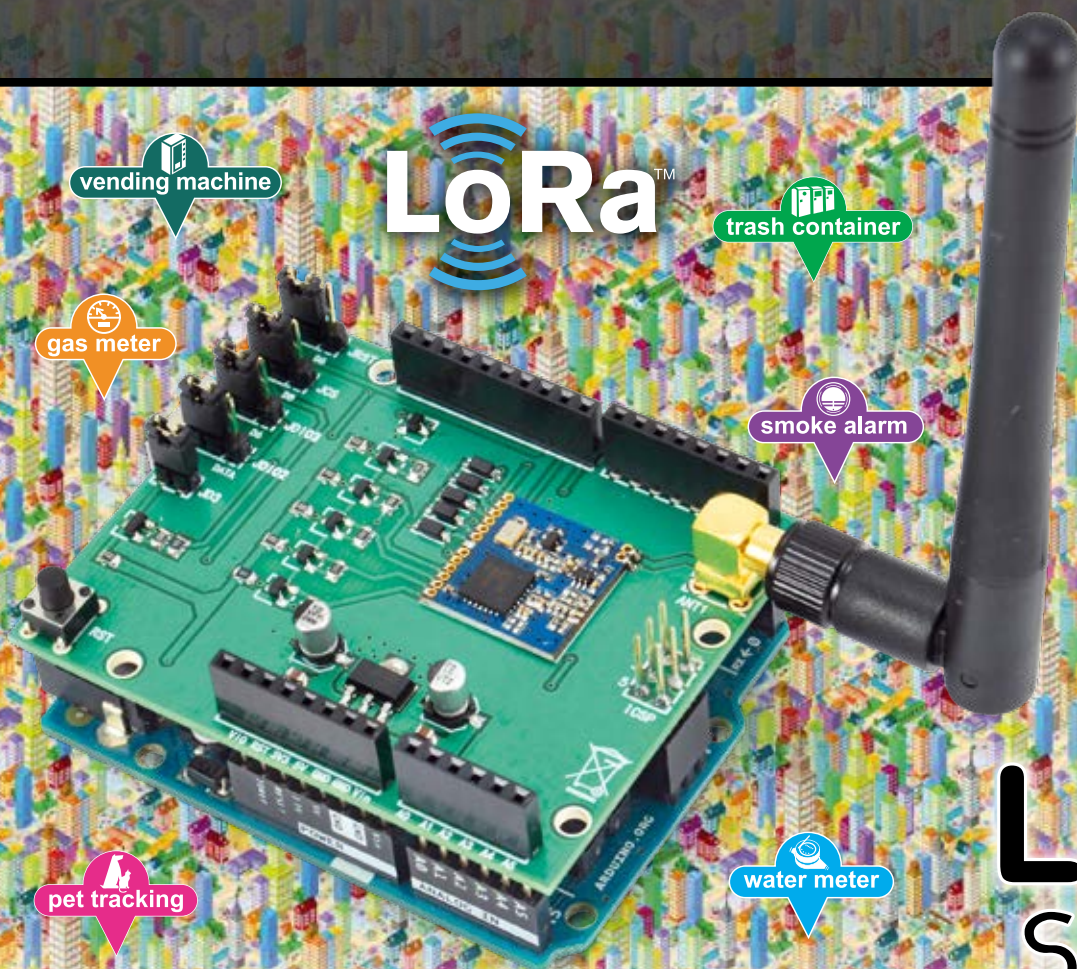
Il materiale va richiesto a:
Futura Elettronica, Via Adige 11,
21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287
<http://www.futurashop.it>

LoRa™

Fornisce ad Arduino la comunicazione long-range basata su tecnologia LoRa. Terza ed ultima puntata.

LoRa SHIELD

dell'ing DANIELE DENARO



In quest'ultimo articolo vedremo come utilizzare la scheda proposta nel fascicolo n° 199 per realizzare una rete composta da nodi indirizzabili e sicuri. Come abbiamo visto nel precedente articolo, l'integrato SX1278 della Semtech, su cui è basato lo shield, permette una sofisticata ed efficiente comunicazione in tecnologia LoRa (Long Range transmission). Questa comunicazione è bidirezionale ed ha un consistente raggio di copertura: un centinaio di metri all'interno di edifici o circa un chilometro in campo aperto. La scheda opera nella banda dei 433 MHz ed è a basso costo; è quindi orientata ad applicazioni che vanno dalla "home automation" alla sicurezza, fino al monitoraggio e all'automazione in agricoltura.

Poiché la comunicazione è bidirezionale, la board si presta ad architetture di rete a "stella" o anche a "reticolo": nel primo caso c'è un nodo privilegiato cui gli altri fanno riferimento, comunicando esclusivamente con esso, mentre nel secondo ogni nodo può decidere di comunicare con qualunque altro. Nella puntata precedente abbiamo visto come programmare la scheda per una comunicazione punto-punto; in tale applicazione l'attenzione è

stata posta esclusivamente sulla verifica delle condizioni di comunicazione e della portata raggiungibile. Infatti l'applicazione era basata su un trasmettitore che lanciava un messaggio ed un ricevitore che ne faceva l'eco; non era presente alcuna forma di indirizzamento delle comunicazioni e volendo utilizzare più sistemi insieme l'unica discriminante delle trasmissioni delle singole unità poteva essere la frequenza portante.

È evidente che per realizzare qualunque struttura di rete è necessario un sistema di indirizzamento delle singole comunicazioni. Ma non basta: a causa della portata, che non è limitata a pochi metri, la trasmissione è facilmente intercettabile e quindi necessita di un sistema di protezione.

La Semtech ed altri soggetti della cosiddetta "LoRa Alliance" hanno progettato un protocollo di rete chiamato LoRaWAN; è abbastanza complesso e in esso le periferiche con sensori e attuatori ("End-device") sono collegate a dei Gateway che instradano il traffico sulla rete Internet fino al server dedicato alla loro gestione. Il protocollo è complesso a causa delle molte variabili che sono gestite; prima fra tutte la gestione dell'alimentazione delle periferiche,

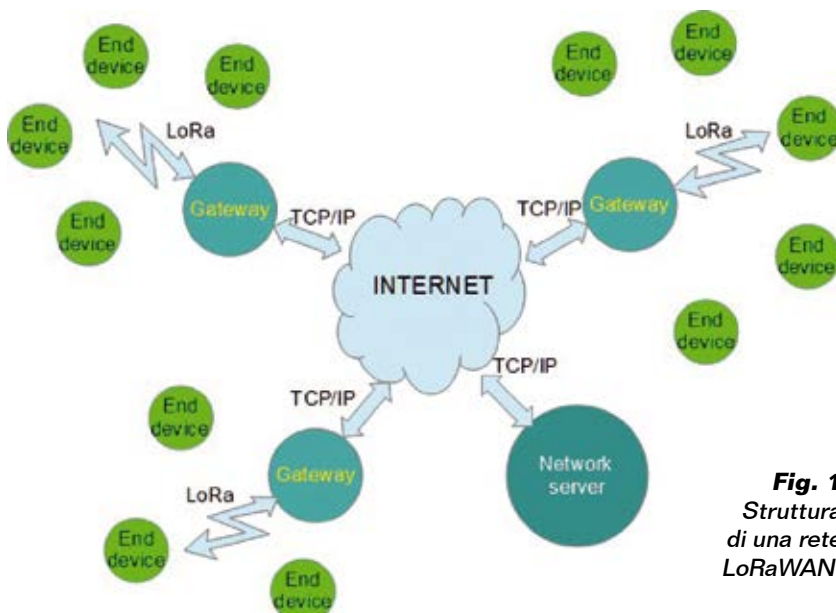


Fig. 1
Struttura
di una rete
LoRaWAN.

che si tende a ridurre al massimo. Inoltre sono gestiti in modo adattativo sia il salto di frequenza che

il data-rate. Il protocollo prevede anche la crittografia AES con chiave di 128 bit e prevede tre di-

verse tipologie di "End-device": classe A, classe B e classe C. La classe A è quella di base e prevede l'iniziativa da parte della periferica che, dopo la trasmissione apre due piccole finestre temporali per la risposta da parte del server e poi si spegne. La classe B ha la possibilità di avere delle temporizzazioni di ascolto, mentre la C può stare sempre in ascolto, perché evidentemente non ha problemi di alimentazione. Le specifiche del protocollo LoRaWAN possono essere scaricate dal sito della Semtech. Il protocollo LoRaWAN è troppo complesso per una struttura più limitata come può essere una piccola rete privata per utilizzo

Come proteggere le informazioni

La crittografia, ovvero l'attività di camuffare le informazioni verso occhi ed orecchie indiscrete, è antica praticamente come l'uomo. Senza contare i sistemi fisici di occultamento, come gli inchiostri simpatici di infantile memoria, già Giulio Cesare utilizzava regolarmente la crittografia a sostituzione di caratteri nei suoi messaggi.

Si può dire che la crittografia nasce proprio con il metodo della sostituzione di ogni carattere con un altro secondo uno schema che prende il nome di "chiave". Non è il metodo ad essere segreto, ma la chiave, senza la quale non è possibile rendere intellegibile il testo. Ma la crittografia a sostituzione di carattere ha un fondamentale punto debole: lascia inalterata l'informazione elementare, cioè il carattere. Per cui, nonostante la notevole complicazione dello schema di spostamento, avvenuto nel corso dei secoli, l'analisi statistica della frequenza dei caratteri nel messaggio codificato, permette di ricostruire la chiave. Infatti, poiché ogni lingua ha una particolare distribuzione media dei singoli caratteri, se i messaggi sono molti o molto lunghi, è possibile associare il carattere vero al

carattere di camuffamento.

Ma cosa succede se il metodo di sostituzione cambia continuamente secondo una meta-chiave? E' questo il metodo utilizzato dalla macchina a rotori Enigma, utilizzata dai tedeschi nella seconda guerra mondiale ed alla cui decrittazione ha lavorato il famoso matematico e precursore dell'informatica Alan Turing. Questa volta la distribuzione dei caratteri nel testo cifrato è uniforme e non fornisce alcuna indicazione. È necessario ricorrere ad altri trucchi, come l'intuizione di parte del contenuto o la ricerca esaustiva ("forza bruta").

L'avvento dei computer, se da un lato ha aumentato a dismisura le esigenze di protezione - se non altro a causa dell'esplosione di comunicazione - d'altro canto ha fornito dei meccanismi formidabili di protezione: invece di trasformare i caratteri, vengono modificati i singoli bit. Già questo garantirebbe una notevole assenza di schema nel testo crittato, ma se poi ci aggiungiamo ripetute azioni non lineari con l'ausilio di chiavi particolarmente lunghe, il testo finale è praticamente non decodificabile in tempi umani ricorrendo al sistema della forza

bruta. In pratica l'informazione di tutto il messaggio risulta sparpagliata come un rumore. Solo l'applicazione della chiave insieme al procedimento inverso può ricostruire il messaggio.

Questo tipo di crittografia è detta simmetrica, perché la stessa chiave è utilizzata per la codifica e la decodifica. Ne consegue che la chiave deve essere comunicata tra gli utenti in modo sicuro. Attualmente il migliore sistema a chiave simmetrica è il sistema AES con chiave di 256 bit. Si stima che con i computer attuali i tempi di decifrazione con la forza bruta siano di milioni di anni. Ma come si fa se la chiave non può essere comunicata in modo sicuro? Questa, per esempio, è la comune realtà dell'e-commerce, dove non è pensabile che per acquistare un prodotto si debba chiedere "per lettera assicurata" una chiave da utilizzare. Per fortuna negli ultimi decenni è stata sviluppata la crittografia asimmetrica, anche detta a chiave pubblica. Questa è particolarmente impegnativa a livello computazionale e viene utilizzata soprattutto come canale sicuro per la comunicazione della chiave simmetrica utilizzata per l'effetti-

Indirizzo automaticamente costruito avendo fissato a 64 il numero massimo di periferiche ed il codice di rete



Fig. 2 - Esempio di costruzione dell'indirizzo.

Tabella 1 - Indirizzamento.

setNetAddress(767,64)	Definisce una rete di indirizzo 767 e con possibili 63 sub-indirizzi (0 non è contemplato) (ritornerebbe false se al posto di 767 ci fosse 3400)
sendNetMess(7,1,"Test",4)	Spedisce il messaggio "Test" al sub-indirizzo 7 (lui è 1) (l'indirizzo di rete è stato definito dalla precedente funzione)
receiveNetMess(1,7,buff,8)	Attende messaggio da 7 per lui (indirizzo 1). Se al posto di 7 ci fosse 0, riceverebbe da tutti.

locale. Per questo motivo abbiamo pensato di completare la libreria LORA con l'imple-

mentazione di una struttura di indirizzi ed una protezione con crittografia AES.

LIBRERIA LORA: INDIRIZZAMENTO

Nel protocollo da noi definito, è stato inserito un indirizzo del destinatario ed uno del mittente. L'indirizzo è a 16 bit (word) che permettono 65.534 possibilità (lo 0 non è contemplato).

Per rendere più pratico il suo utilizzo, l'indirizzo è stato suddiviso in un indirizzo di rete ed un sub-indirizzo di "device"; però la suddivisione non è fissa e può essere decisa dall'utente, nel senso che la porzione dell'indirizzo riservata alla rete e di conseguenza quella delle periferiche, è variabile. In pratica si tratta di definire il massimo numero delle periferiche indirizzabili (che sia una potenza di 2) e un indirizzo

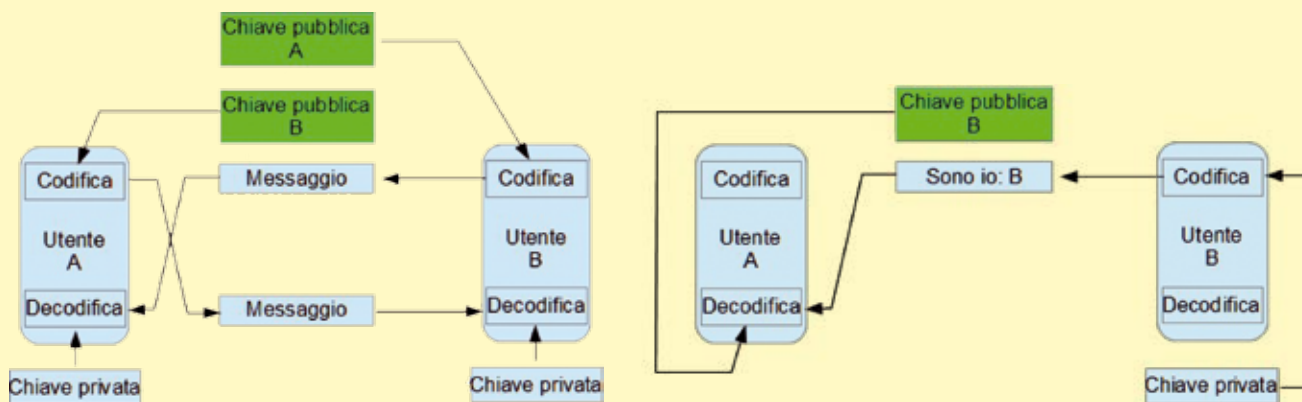


Fig. A - Crittografia a chiave pubblica e firma elettronica

vo trasferimento del messaggio. Oppure viene utilizzata per garantire l'originalità del mittente (firma digitale, **Fig. A**) o l'integrità del messaggio. Infatti un altro aspetto della sicurezza è sapere se il messaggio è stato spedito dal soggetto indicato o da qualcuno che si è sostituito a lui, oppure ha modificato il messaggio originale. Questo è il cosiddetto problema del "man in the middle".

La crittografia a chiave pubblica è piuttosto complessa ed utilizza in sostanza l'aritmetica dei numeri primi o delle funzioni "ellittiche" e soprattutto fa ampio uso dell'aritmetica "modulare" (resto

della divisione o aritmetica dell'orologio); la sostanza può essere riassunta così:

- le chiavi sono due e unidirezionali (una codifica, l'altra decodifica);
- le due chiavi sono in qualche modo collegate, anche se non in modo decodificabile;
- le due chiavi sono intercambiabili (se la seconda codifica, la prima decodifica e viceversa).

C'è infine un ulteriore tipo di protezione: quello necessario per evitare che un soggetto esterno copi brutalmente il messaggio crittato e lo invii in secondo tempo senza esigenza di comprensione.

Questo è un evento caratteristico dei radiocomandi unidirezionali che inviano sempre lo stesso messaggio per azionare un apparato: ad esempio per aprire le portiere di un'auto. Poiché il messaggio è sempre uguale ed ottiene sempre lo stesso effetto, non c'è crittografia che tenga e basta ripeterlo così com'è. Per risolvere il problema è sufficiente rendere il messaggio sempre diverso sincronizzandolo con il ricevitore: è ciò che viene realizzato con il "rolling code". Un'alternativa più complessa è quella di rendere il radiocomando bidirezionale e fargli intraprendere un dialogo di "hand shaking".

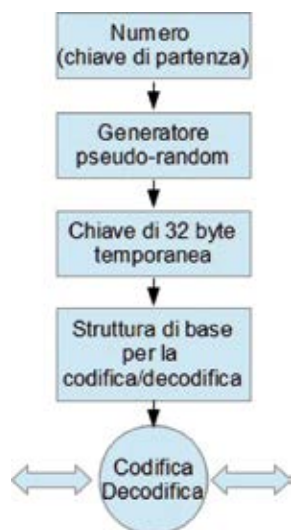


Fig. 3 - Crittografia AES 256 per LoRa.

di rete che sia compreso nello spazio che rimane nella "word" di indirizzo effettivo. Se l'indirizzo di rete eccede, la funzione ritornerà "false". Ovviamente tutti i nodi faranno riferimento a questa struttura con lo stesso numero di rete e diverso indirizzo di nodo.

Nella **Tabella 1** è mostrato un esempio di utilizzo di funzioni di spedizione e ricevimento.

LIBRERIA LORA: CRITTOGRAFIA

Anche se l'applicazione che potreste avere in mente non ha veri vincoli di sicurezza, non è opportuno far viaggiare in chiaro pacchetti radio alle distanze raggiungibili dalle trasmissioni LoRa. Per questo motivo abbiamo pensato a una soluzione che fosse ragionevolmente sicura ma anche praticabile su un hardware limitato come Arduino.



Fig. 4 - Payload LoRa.

Per fortuna in rete è possibile scaricare un software di crittografia AES adattato per Arduino.

Questo software, che utilizza la memoria flash di programma per alcune strutture di base, è stato da noi leggermente modificato nella gestione della chiave.

La crittografia AES utilizzata è su chiave di 256 bit, ma invece di fornire un array di 32 byte, la chiave è generata in maniera pseudo-random a partire da un valore-seme del generatore random di Arduino. Questa modalità riduce l'occupazione della RAM ad un solo numero (integer), ma ha la controindicazione di essere utilizzabile solo per Arduino. Infatti se si utilizzano librerie matematiche differenti, la produzione di chiavi pseudo-casuali a partire dallo stesso seme fornirebbe chiavi non coincidenti tra le varie schede della stessa rete.

La crittografia AES a 256 bit è considerata molto sicura, infatti l'algoritmo compie un certo numero di spostamenti e permutazioni con funzioni non lineari, producendo una sequenza di byte completamente mascherata e paragonabile a rumore puro. Solo l'utilizzo della chiave permette il procedimento inverso. Tuttavia la libreria utilizzata ap-

pesantisce pochissimo Arduino in termini sia di elaborazione che di occupazione di memoria.

Oltre alla crittografia AES, è stato aggiunto un byte al messaggio; tale byte ha un valore casuale ed ha lo scopo di rendere unici messaggi identici che si ripetono. Più che altro è stato pensato per consentire un'eventuale implementazione di una sicurezza aggiuntiva tipo rolling-code. Tenete presente che a causa del tipo di crittografia utilizzato, il valore di un byte si sparpaglia su tutto il messaggio codificato.

La chiave crittografica è definita direttamente nella funzione di attivazione della libreria LoRa. La definizione della chiave è:

Chiave	
begin(33333)	Inizializza la libreria con la chiave 33333. La funzione verifica la presenza della scheda e ritorna false se non è agibile.

PACCHETTO

Il pacchetto inviato e ricevuto in modalità LoRa (ovvero il payload) è quindi composto come si può vedere in **Fig. 4**.

La funzione di spedizione lo costruisce aggiungendo al messaggio i due indirizzi (previa ricostruzione in formato word) ed un byte di valore casuale. Quindi lo

Tabella 2 - Funzioni LoRa basilari.

<code>bool begin(unsigned int keyval);</code>	Inizializza e definisce la chiave per la crittografia
<code>bool setNetAddress(unsigned int netAdd, unsigned int devRange);</code>	Definisce la struttura di rete
<code>int sendNetMess(unsigned int devSubAdd, unsigned int sendSubAdd, byte *mess, int lmess);</code>	Spedisce un messaggio
<code>void receiveMessMode();</code>	Si pone in continuous receiving mode
<code>int receiveNetMess(unsigned int devSubAdd, unsigned int sendSubAdd, byte *buff, byte maxlen);</code>	Verifica se è arrivato un messaggio a lui indirizzato. Ritorna 0 se non è arrivato nulla. Se sendSubAdd è 0 accetta messaggi da tutti altrimenti restituisce -1 se il mittente non corrisponde. Questa funzione va eseguita ripetutamente dopo aver utilizzato la precedente funzione.
<code>unsigned int getSubNetSender();</code> <code>byte getMarker();</code>	Restituisce il mittente Restituisce il marker

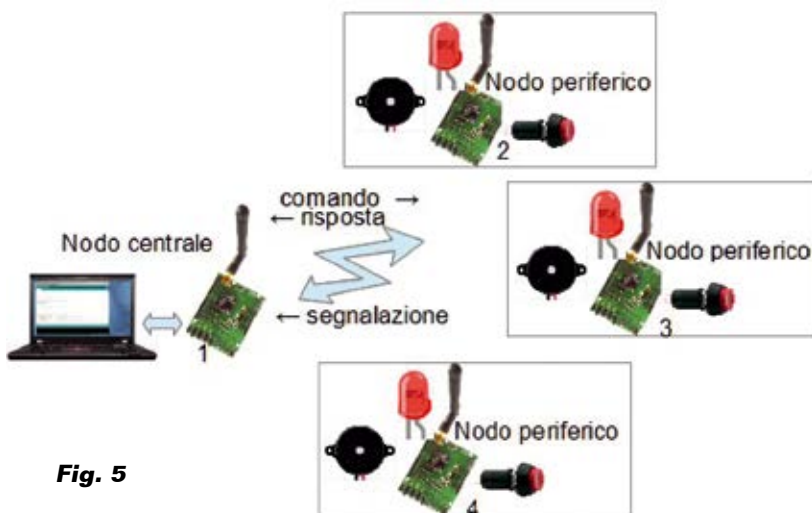


Fig. 5

codifica (con l'eccezione dell'indirizzo destinatario) e lo spedisce. Il destinatario non è codificato per velocizzare il riconoscimento all'arrivo. Appena un pacchetto è arrivato, la funzione di rice-

zione per prima cosa controlla il destinatario: se coincide con il suo indirizzo procede, altrimenti lo scarta. Se è di sua competenza lo decodifica e verifica se lo aspettava dal mittente indicato

(può anche accettare messaggi da qualunque mittente). Alla fine restituisce il numero di byte del messaggio contenuto nel buffer predisposto. Restituisce 0 se non è in arrivo alcun messaggio, oppure se il messaggio non è di sua competenza. Nel caso di messaggio arrivato si può chiedere il mittente o il marker casuale.

FUNZIONI DI LIBRERIA LORA

Le funzioni basilari per utilizzare la scheda in modalità LoRa all'interno di una struttura di rete così definita sono veramente poche. La **Tabella 2** ne propone il riassunto. Ora vi mostriamo un semplice esempio che evidenzia la semplicità d'uso: è contenuto nella

Listato 1 - Sketch del nodo centrale

```
#include <LORA.h>
#include <SPI.h>

#define LORANET 333 // indirizzo di rete
#define DEVRANGE 16 // range di indirizzi di nodo
#define THISADD 1 // indirizzo di questo nodo
#define KEYVAL 111 // valore per la generazione della chiave AES

#define RXTIMEOUT 500 // timeout di replay

#define inplen 64 // buffer per la spedizione
#define reclen 64 // buffer per la ricezione

LORA LR; // Class LORA instance
boolean SHIELD=true;
:
:
void setup()
{
  Serial.begin(9600);
  if (!LR.begin(KEYVAL))
  {Serial.println("No LoRa shield detected! Stop!");SHIELD=false;return;}
  LR.setNetAddress(LORANET,DEVRANGE);
  Serial.print("Net address : ");Serial.println(LORANET);
  Serial.print("Devices range: ");Serial.println(DEVRANGE);
  Serial.print("My address : ");Serial.println(THISADD);
  Serial.println("LoRa transmitter ready...");
  SX.setPower(PWR);
  showConfig();
  LR.receiveMessMode(); // set shield in continuous receiving mode
}

void loop()
{
  delay(200);
  if (!SHIELD) return; // se la scheda non è operativa non può fare nulla
  int nb;
  if ((nb=getInput())>0) //Se è stato inserito un comando
  {if (sendBuff(nb))getReplay();}
  if ((nb=getMess())>0) {} //Se è arrivato un messaggio spontaneo lo stampa
}
```

Listato 2 – Sketch per i nodi periferici

```
#include "LORA.h"
#include <SPI.h>
/* Simulazione di sensori ed attuatori*/
#define psound 9          // pin for buzzer
#define pingo 8           // pin for push-button
#define pinf 7            // pin for signaling led
#define LORANET 333       //indirizzo di rete
#define DEVRANGE 16       //range di indirizzi di nodo
#define THISADD 2         //indirizzo di questo nodo (gli altri nodi lo avranno diverso)
#define KEYVAL 111        // valore per la generazione della chiave AES
#define MASTER 1          // indirizzo del nodo centrale
LORA LR;                  // Class LORA instance
boolean SHIELD=true;
char recbuff[reclen];     //receiving buffer
char sendbuff[reclen];    //transmitting buffer
:
:

void setup()
{
  pinMode(pingo, INPUT_PULLUP);
  digitalWrite(pinf, 0);
  pinMode(pinf, OUTPUT);
  pinMode(psound, OUTPUT);
  Serial.begin(9600);
  if (!LR.begin(KEYVAL))
  {
    Serial.println("No LoRa shield detected! It can't perform echo!"); SHIELD=false; return;
  }
  Serial.println("LoRa receiver.");
  LR.setNetAddress(LORANET, DEVRANGE);
  Serial.print("Net address : "); Serial.println(LORANET);
  Serial.print("Devices range: "); Serial.println(DEVRANGE);
  Serial.print("My address : "); Serial.println(THISADD);
  SX.setPower(PWR);
  showConfig();
  Serial.println("Waiting for message...");
  LR.receiveMessMode(); // set shield in continuous receiving mode
}

void loop()
{
  int nb;
  if (!SHIELD) {delay(200); return;} // se la scheda non è operativa non può fare nulla
  if ((nb=getMess())>0) //Se riceve un comando lo esegue e risponde
  {
    decode(nb); LR.receiveMessMode(); // poi si rimette in ricezione
  }
  if (getInput()) //Se il pulsante/interruttore è premuto manda un avviso
  {
    sendWarn(); LR.receiveMessMode(); // poi si rimette in ricezione
  }
}
```

libreria e in questa sede ne vedremo le caratteristiche principali, lasciando ai lettori l'analisi del listato completo. L'esempio è formato da due sketch e rappresenta una struttura a stella in cui un nodo interroga

gli altri, ma può ricevere anche degli avvisi spontanei nel caso che sulla periferica si realizzi un certo evento. Nella nostra simulazione l'evento è l'azione su un pulsante (**Fig. 5**). Il primo sketch rappresenta il nodo centrale ed ha indirizzo 1 (indirizzo di nodo all'interno di una rete con 15 possibili nodi). Lo sketch inizializza la libreria e la scheda e poi si mette in attesa di un input da console o di un messaggio da una periferica. In caso di input da console (ovvero un comando) invia il corrispondente messaggio ed attende la risposta che visualizzerà sulla console. Nel caso di messaggio spontaneo, lo visualizzerà sulla console. Trovate il codice corrispondente nel **Listato 1**. I comandi inseribili sono relativi alla richiesta di un eco, ad accendere/spegnere un LED, a far suonare un cicalino o a verificare l'azione su un pulsante o un interruttore. Lo sketch del nodo periferico (**Listato 2**), invece, non fa altro che attendere l'arrivo di un messaggio e verificare l'azione su un pulsante o un interruttore. Nel caso arrivi un messaggio (co-

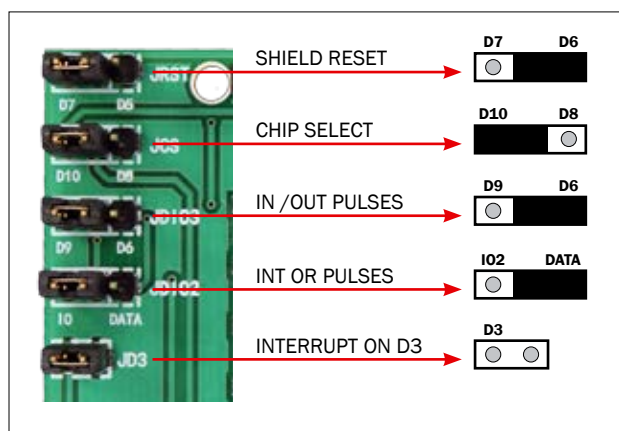


Fig. 6 - Impostazione della libreria.

mando) lo esegue e risponde. Nel caso verifichi il cambiamento di stato di un pulsante o interruttore, spedisce autonomamente un messaggio di attenzione.

CONCLUSIONI

In queste tre puntate abbiamo analizzato un componente molto sofisticato e flessibile come l'SX1278, sul quale abbiamo realizzato una shield di basso costo. Lungi dal pensare di aver esaurito le caratteristiche e le potenzialità di questa innovativa tecnologia, speriamo di aver realizzato una pratica ed efficiente base di lavoro per sviluppare interessanti applicazioni. Sul sito di GitHub (github.com/open-electronics/LoRa) è stata aperta una pagina relativa a questo progetto, dove oltre alla libreria potrete trovare applicazioni che sia noi ma soprattutto voi, vorrete condividere. Le applicazioni, lo abbiamo detto, sono pressoché infinite: dal sistema di allarme casalingo alla gestione dei sensori/attuatori di una piccola impresa. Noi, per conto nostro, stiamo già pensando ad una miniaturizzazione del transceiver che appoggiandosi sempre su Arduino (in formato miniaturizzato), possa risolvere situazioni dove lo spazio e/o l'alimentazione sono critiche. Per esempio un radiocomando bidirezionale tascabile. Poiché pensiamo che questa tecnologia sia valida in termini di rapporto prestazioni/prezzo, vi invitiamo a scatenare la vostra fantasia e a suggerirci o a sottoporci progetti in tema. Vi ricordiamo che con l'utilizzo di RandA su Raspberry la scheda LoRa può diventare un completo gateway per controllare periferiche WorldWide. ■



per il MATERIALE

La shield Long Range (cod. FT1190M) comprensiva di modulo radio e tutti i componenti necessari al suo funzionamento, è disponibile a 39,50 Euro. Il modulo DRF1278F è disponibile anche separatamente a 19,00 Euro. Arduino UNO (cod. ARDUINOUNOREV3) viene venduto a 24,50 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>

CORSI FUTURA ACADEMY

Puoi scegliere
tra numerosi argomenti:

- Stampa 3D
- Raspberry Pi
- Droni
- Arduino
- Robotica
- Eagle
- Elettronica
- Fotovoltaico

... e molti altri ancora!



NON ASPETTARE.

Se ti iscrivi in anticipo avrai diritto ad uno sconto!

INFORMAZIONI DETTAGLIATE E ISCRIZIONI SU
www.futura.academy

TORNANO I CORSI FUTURA ACADEMY

Teoria, laboratori pratici e il supporto di tutor che accompagnano i partecipanti in un percorso formativo rivolto a chi vuole acquisire competenze utili a scuola, nel mondo del lavoro o anche solo per nutrire la propria passione.

a cura della REDAZIONE

Quando, in collaborazione con Futura Elettronica, abbiamo iniziato ad allestire il progetto Futura Academy, lo abbiamo fatto quasi per scommessa: l'intento era portare, oltre che tramite le pagine cartacee della nostra rivista e quelle web del nostro sito www.elettronica.in.it e del relativo blog <http://blog.elettronica.in.it>, un po' della nostra ventennale esperienza in ambito elettronico e tecnologico e un pizzico dell'entusiasmo che ha animato la nostra attività.

Siamo partiti allestendo una capiente aula con i più moderni supporti multimediali e di connettività wireless e cablata, e reperendo insegnanti qualificati nelle discipline che ritenevamo le più interessanti per un pubblico di speri-

mentatori. A distanza di quasi un anno abbiamo "aggiustato" il tiro seguendo anche le vostre richieste e i suggerimenti circa gli argomenti da affrontare; così ne è scaturita un'ampia offerta formativa che spazia dall'elettronica

(di base e avanzata) all'utilizzo delle due schede di prototipazione "regine" ossia Arduino e Raspberry Pi, fino alla stampa 3D, alla robotica, ai droni (multicotteri), ai CAD elettronici (ORCAD e Eagle) e alla programmazione in C/

<http://futura.academy>



C++ e dei microcontrollori Microchip tramite l'IDE MPlab. Ma abbiamo avuto un occhio di riguardo anche per chi inizia e vuole un po' più pratica e meno teoria, venendo incontro alle esigenze degli sperimentatori con corsi pratici sulla realizzazione di semplici circuiti sperimentali.

Dobbiamo dire che il nostro pubblico ci ha dato ancora una volta una certa soddisfazione, perché in molti hanno seguito l'iniziativa e presenziato ai corsi. Ma che cos'è esattamente Futura Academy?

Con questo nome, scelto forse con un po' di presunzione, abbiamo battezzato un progetto di formazione che consiste in Corsi di Elettronica, Informatica e Tecnologie Innovative, ma anche di laboratori pratici come il "corso di costruzione della 3Drag", dove chi vuole veder nascere sotto i propri occhi la propria stampante 3Drag può farlo con l'aiuto dei nostri esperti, invece che essere lasciato solo ad affrontare l'impresa o accontentarsi di avere in mano un prodotto già fatto e privarsi così della soddisfazione di avere a casa qualcosa di autocostruito.

Futura Academy è un'iniziativa che rientra a pieno titolo in quella che è la nostra "mission": divulgare e diffondere l'elettronica e le tecnologie innovative a tutti i livelli

e con tutti gli strumenti che il caso richiede. Il supporto di Futura Elettronica -azienda impegnata nella progettazione, produzione e commercializzazione di dispositivi elettronici- permette di avere a disposizione non solo apparecchiature innovative e utili dal punto di vista didattico, ma anche strumenti di sviluppo e attrezzature. Non funziona una scheda? Si è rotto un cavo? Serve un saldatore? Nessun problema, perché dietro l'aula corsi si può contare sul nutrito magazzino e -se è il caso- sul supporto dei tecnici di Futura Elettronica. Inoltre quanti frequentano i corsi non dovranno preoccuparsi di questo aspetto e potranno, nel caso, acquistare al termine del corso, gli stessi strumenti utilizzati per seguire le lezioni, per utilizzarli a casa propria e approfondire i concetti appresi. Tra gli scopi di Futura Academy, c'è la collaborazione con le istituzioni scolastiche per fornire quelle competenze che per i motivi più vari, il "pubblico" non riesce a soddisfare; uno degli scopi di Futura Academy è, infatti, trasmettere le nostre competenze anche agli insegnanti delle scuole a indirizzo tecnico, affinché loro stessi possano poi metterle a disposizione dei propri studenti per arricchirne la formazione e il bagaglio

culturale.

Per quanto riguarda la tipologia e il livello dei corsi, prevediamo corsi di base e avanzati: i primi sono dedicati a chi si avvicina per la prima volta al tema trattato o che ha ancora poca dimestichezza con la tecnologia presentata, mentre nei laboratori avanzati si approfondisce l'argomento di cui si conoscono già i principi di base, studiando e sviluppando concretamente applicazioni o progetti in tema.

Ai corsi abbiamo deciso di dare un'impostazione tale da renderli accessibili a tutti coloro che hanno voglia di imparare e che si sentono predisposti all'apprendimento di tecnica e tecnologia.

Per consentire l'accesso anche a chi lavora, gran parte dei corsi si svolge il sabato.

Al termine di ogni corso viene rilasciato un attestato di frequenza che certifica la regolare partecipazione.

I corsi di Futura Academy sono svolti da Tutor esperti, selezionati attentamente dal team di Futura Elettronica. Ogni docente organizza la giornata di corso alternando, ove previsto, momenti di spiegazione teorica all'utilizzo pratico delle macchine, dei software e delle schede relative al corso di for-



mazione prescelto.

È possibile consultare in ogni momento il calendario aggiornato su <http://futura.academy/>, dove è altresì possibile vedere i corsi disponibili e i relativi dettagli, quindi accedere al modulo di partecipazione tramite cui iscriversi.

È anche possibile scegliere direttamente l'argomento di proprio interesse ed accedere alla relativa pagina web, per visualizzare i dettagli del programma e le date previste.

Dunque cosa aspettate? Volete avvicinarvi all'elettronica ma non ve la sentite di fare gli autodidatti? Volete approfondire la conoscenza di un argomento riguardante hardware, software o tecnologia? Futura Academy può accompagnarvi lungo il percorso con i suoi tutor, pronti a guidarvi passo dopo passo!

FISH'N TWEET

Controlliamo le uscite di uno shield di espansione Arduino montato sulla nostra Fishino Uno, tramite Twitter.

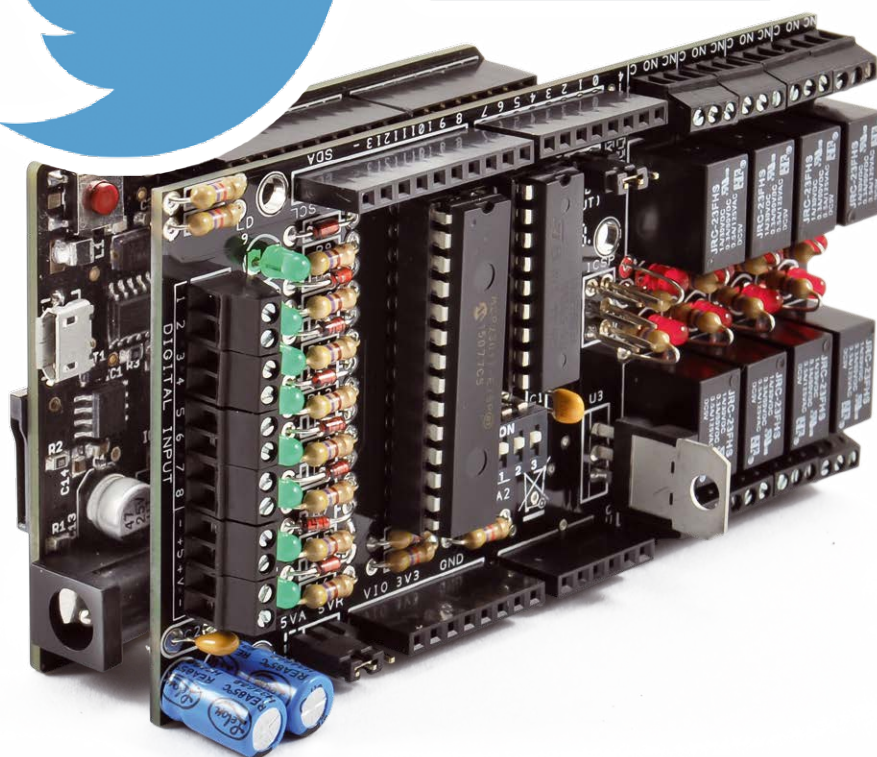


Elettronica In
@elettronica_in

#fishino 24 ON

Reply Elimina Favorite Altro

21 ottobre 2015 12:36



di MASSIMO DEL FEDELE

Avete conosciuto Fishino, il nostro Arduino dotato di connettività WiFi ed RTC, nel fascicolo n° 199; ora è venuto il momento di vedere un'applicazione pratica della scheda, che permette, in modo molto economico e semplice, la gestione remota di apparecchiature elettriche quali lampadine, caldaie, condizionatori, antifurto, eccetera. L'unicità del sistema è costituita dal fatto di non richiedere apparecchiature di rete o server remoti esterni, ma solo una scheda Fishino UNO ed eventualmente uno o più shield per la gestione dei carichi applicati, che nel nostro caso

Ricerca avanzata

Fig. 1

Parole

Tutte queste parole

Questa frase esatta

Una di queste parole

Nessuna di queste parole

Questi hashtag

Scritto in

Persone

Da questi utenti

Fig. 2



sono I/O Expander basati su MCP23017 (descritti nel fascicolo n° 181, cui rimandiamo per gli approfondimenti del caso). Questo progetto, non realizzabile con un semplice Arduino nemmeno se dotato di shield WiFi o Ethernet Shield, è stato reso possibile grazie alla capacità di Fishino di effettuare connessioni sicure con il protocollo https, disponibili solitamente su apparecchi di livello superiore e decisamente più costosi; restare in tema Arduino, attualmente per realizzare un progetto equivalente occorrono una YÙN e un software decisamente più complesso. Prima di entrare nel merito del progetto vediamo i vari modi per controllare apparecchiature a distanza.

Controllo via SMS

Vantaggi:

- il ricevitore può essere reso portatile ed indipendente da infrastrutture di rete;
- risposta pressoché istantanea;
- gestibile da qualsiasi cellulare, anche non recente;
- portata limitata solo dalla copertura radiomobile.

Svantaggi:

- richiede una SIM sull'apparecchio controllato, che solitamente scade se non usata;
- richiede un dispositivo GSM, non economicissimo, per la ricezione;
- richiede una copertura GSM, cosa non sempre scontata.

Controllo via moduli radio

Vantaggi:

- il ricevitore può essere reso portatile e indipendente da infrastrutture di rete;
- risposta pressoché istantanea;
- molto economico.

Svantaggi:

- portata molto limitata e fortemente influenzata dalla presenza di eventuali ostacoli.

Controllo via Internet

Vantaggi:

- può essere reso portatile al costo di un ricevitore GPRS;
- può essere effettuato pressoché da qualsiasi PC o telefono cellulare;
- molto economico, se presso la destinazione è presente un'infrastruttura di rete;
- può diventare bidirezionale senza costi aggiuntivi e senza particolare difficoltà.

Svantaggi:

- richiede la presenza di un'infrastruttura di rete in ricezione o un costo aggiuntivo per il GPRS;
- la stabilità dipende dalla connessione Internet;
- i tempi di accesso possono essere variabili a seconda della connessione e/o del metodo utilizzato;
- spesso richiede ricevitori complessi e/o un computer di appoggio;
- richiede un IP pubblico sull'apparecchio ricevente o l'uso di un servizio intermedio di terze parti;
- possono esserci problemi di sicurezza che richiedono accorgimenti particolari per evitare utilizzi non autorizzati.

Con questo progetto abbiamo cercato di unire i vantaggi del controllo via Internet, eliminando alcuni degli svantaggi di tale tecnologia, sfruttando un servizio ben noto a tutti: Twitter. In pratica, i dispositivi vengono azionati inviando un semplice tweet con un formato particolare, dall'utente preimpostato (o più utenti, con qualche piccola modifica al software). Con questo sistema risolviamo le seguenti problematiche del controllo via Internet:

- la stabilità della connessione; nel caso la connessione sia momentaneamente assente, il comando viene solo posticipato ed eseguito al suo ripristino;
- il ricevitore risulta largamente semplificato, dal lato sia hardware che software;
- non è richiesto alcun IP pubblico né un servizio di rete intermedio;
- la sicurezza è garantita dall'infrastruttura di Twitter, che lavora con connessioni sicure.

Come per tutte le cose, anche questo sistema non è esente da svantaggi, che sono principalmente la non immediatezza dell'attuazione (il tempo di propagazione di un tweet è di qualche secondo fino ad un paio di decine di secondi) ed alcune limitazioni (aggirabili) sull'invio di tweet duplicati.

Vediamo innanzitutto l'utilizzo, per poi passare alla descrizione del sistema.

Il controllo avviene tramite l'invio di un Tweet con il seguente formato:

#hashtag porta ON/OFF

ove **'#hashtag'** è un hashtag impostabile nel software (nel nostro caso abbiamo preimpostato **#fishino**), **'porta'** è un numero di porta da attivare (vedremo i dettagli in seguito) e **'ON'** o **'OFF'** sono i comandi di accensione/spengimento della medesima. Esempio:

#fishino 24 ON accende l'uscita 24
#fishino 10 OFF spegne l'uscita 10

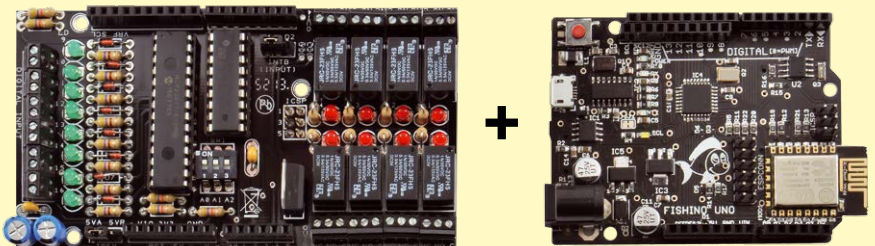
Come detto sopra, tra l'invio del tweet e l'esecuzione del comando passa un tempo stimabile in 5-20 secondi, dipendente dalla congestione della rete di Twitter. L'altro svantaggio di cui parliamo sopra è l'impossibilità di

Il progetto si propone di controllare utilizzatori elettrici o gestibili elettricamente tramite l'invio di tweet contenenti le istruzioni per governare i relé di schede di I/O applicabili a Fishino: si tratta di shield FT1079 della Futura Elettronica, sviluppati intorno all'I/O Expander MCP23017 della Microchip. Questo integrato dispone in totale di 16 pin (suddivisi a livello logico in 2 diverse porte A e B) configurabili individualmente come ingresso o uscita. Nello shield gli 8 pin della porta A sono impostati come uscite e gli 8 pin della porta B come ingressi. Le uscite pilotano gli ingressi di un line-driver ULN2803, progettato appositamente per pilotare carichi induttivi come le bobine dei relé (infatti al suo interno dispone di diversi stadi a transistor dotati di diodi di protezione della giunzione base-collettore). Gli ingressi sono invece collegati ad apposite sezioni circuitali che realizzano gli input digitali. Ogni ingresso dispone di un

diodo di protezione, una resistenza di pull-up e un LED che ne segnala lo stato. Comunque in questa applicazione non usiamo gli input.

La gestione dell'MCP23017 avviene da I²C Bus, realizzato con i piedini 12 (SCL) e 13 (SDA), i quali sono collegati ai relativi pin di Fishino. L'MCP23017 offre la possibilità di generare un interrupt a fronte della variazione di almeno un pin programmato come input; nel nostro caso non abbiamo sfruttato questa funzione perché non ci serve acquisire lo stato degli ingressi. L'indirizzo I²C dell'MCP23017 viene selezionato tramite il dip-switch SW1.

Lo shield si alimenta tramite il Vin di Fishino, grazie ad un regolatore interno, basato su un L78L05, che ricava 5 V stabilizzati; per attivare questa opzione, il jumper JP5V va chiuso su 5VR. Chiudendolo su 5VA, lo shield riceve i 5 V direttamente dalla linea +5V di Fishino; ciò è possibile solo se il numero di schede è limitato a due.



inviare tweet duplicati in breve sequenza; per esempio:

#fishino 24 ON accende l'uscita 24
#fishino 24 OFF spegne l'uscita 24
#fishino 24 ON accende l'uscita 24

Il terzo tweet non viene accettato ma segnalato con un "hai già inviato questo tweet". A ciò si può ovviare molto semplicemente aggiungendo qualche carattere a caso dopo la scritta ON:

#fishino 24 ON 123456 accende l'uscita 24

I caratteri supplementari (123456) vengono trascurati dal ricevitore.

FUNZIONAMENTO

Partiamo dall'invio del tweet, che viene memorizzato e propagato

da Twitter. Chiunque può vedere, ricevere e cercare il tweet nella rete da qualsiasi parte del mondo. Utilizzando un PC o un telefono cellulare si può effettuare la ricerca, per esempio, impostando come parametri il mittente del tweet e l'hashtag (**Fig. 1**).

Il risultato della ricerca è qualcosa di simile alla **Fig. 2**.

Come si nota, vengono mostrati i tweet corrispondenti al criterio di ricerca, dall'ultimo arrivato a quelli precedenti.

Analizziamo un po' più in dettaglio un singolo tweet; ogni messaggio contiene:

- un mittente (@mdelfede in questo caso);
- nessuno, uno o più hashtag

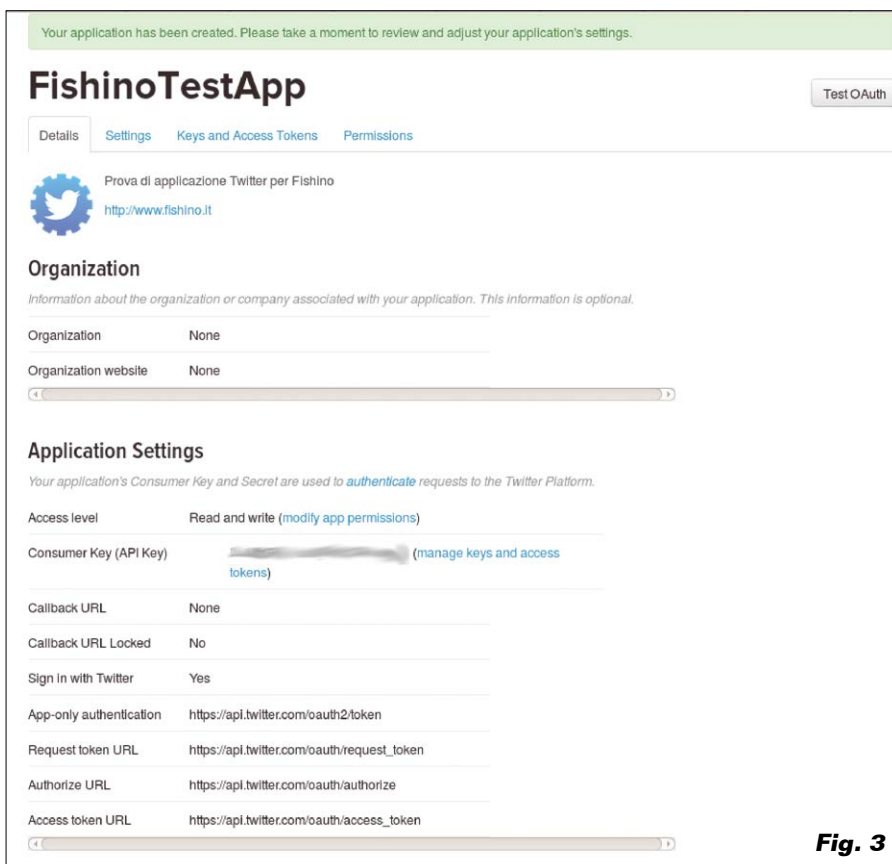


Fig. 3

(#fishino in questo caso);

- un messaggio di testo, comprendente gli hashtag e altro;
- un timestamp, ovvero data ed ora di invio.

Il tweet può anche contenere riferimenti ad altre persone, immagini, URL, eccetera.

Nel nostro sistema sono importanti il **mittente**, utilizzato per consentire o negare l'accesso, gli **hashtag**, utilizzati per differenziare un tweet normale da uno di comando al sistema, il testo, che contiene il comando vero e proprio, ed infine il **timestamp**, o data e ora di invio, che serve per discriminare i comandi già ese-

guiti da quelli ancora da eseguire. Ovviamente non possiamo utilizzare una ricerca simile tramite un PC per controllare i nostri dispositivi, ma abbiamo bisogno di qualcosa di più "digeribile" da un sistema automatico, ed il più semplice possibile.

Per questo ci vengono in aiuto le API (Application Programming Interfaces) fornite da Twitter che permettono di effettuare, tra le altre cose, delle ricerche sui tweet.

LE API DI TWITTER

Le API in questione sono molto estese, permettendo di effettuare moltissime operazioni, sia di ricerca che di invio di tweet tra-

mite software. Attualmente, per il nostro controller ci interessano solo le API di ricerca.

Il funzionamento di tutte le API è comunque abbastanza omogeneo e, una volta capiti i meccanismi di un loro sottoinsieme, risulta facilissimo utilizzarle.

Iniziamo con il superare il primo scoglio incontrato durante lo sviluppo, e che è lo scoglio più grosso che ha obbligato molti progetti analoghi ad utilizzare sistemi molto più complessi di Fishino e/o PC di appoggio per la ricerca dei tweet: la sicurezza. Twitter utilizza, per rendere sicuro il suo sistema, principalmente due meccanismi i quali prevedono che:

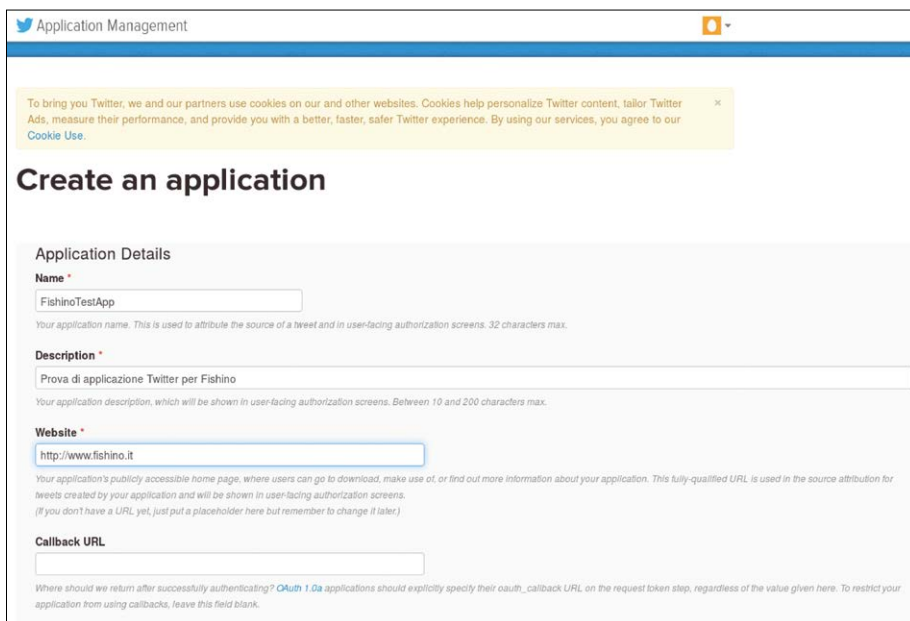
- la connessione al server delle API debba avvenire con protocollo sicuro https;
- l'utente delle API debba autenticarsi in uno dei due modi che vedremo in seguito.

Se il secondo requisito è relativamente semplice da soddisfare, il primo impone -per contro- l'utilizzo di un sistema in grado di effettuare connessioni con protocollo sicuro HTTPS. Questo esclude automaticamente gli Ethernet Shield e WiFi Shield di Arduino, per esempio, che non sono in grado di utilizzarlo. Invece Fishino è in grado di eseguire connessioni https anche se col limite di una connessione per volta, vista la grande richiesta di risorse di memoria necessaria per la crittografia.

Vediamo, nel **Listato 1**, una

Listato 1

```
GET
/1.1/search/tweets.json?count=1&q=%23fishino+from%3Amdelfede&result_type=recent&max_id=647068135569387520&since_id=646038066440896517 HTTP/1.1
Host: api.twitter.com
User-Agent: Fishino Twitter Client
Authorization: Bearer AAAAAAAAAAAAAAAAAA(altri caratteri della chiave di autorizzazione)
```

Application Management

To bring you Twitter, we and our partners use cookies on our and other websites. Cookies help personalize Twitter content, tailor Twitter Ads, measure their performance, and provide you with a better, faster, safer Twitter experience. By using our services, you agree to our [Cookie Use](#).

Create an application

Application Details

Name *
FishinoTestApp
Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

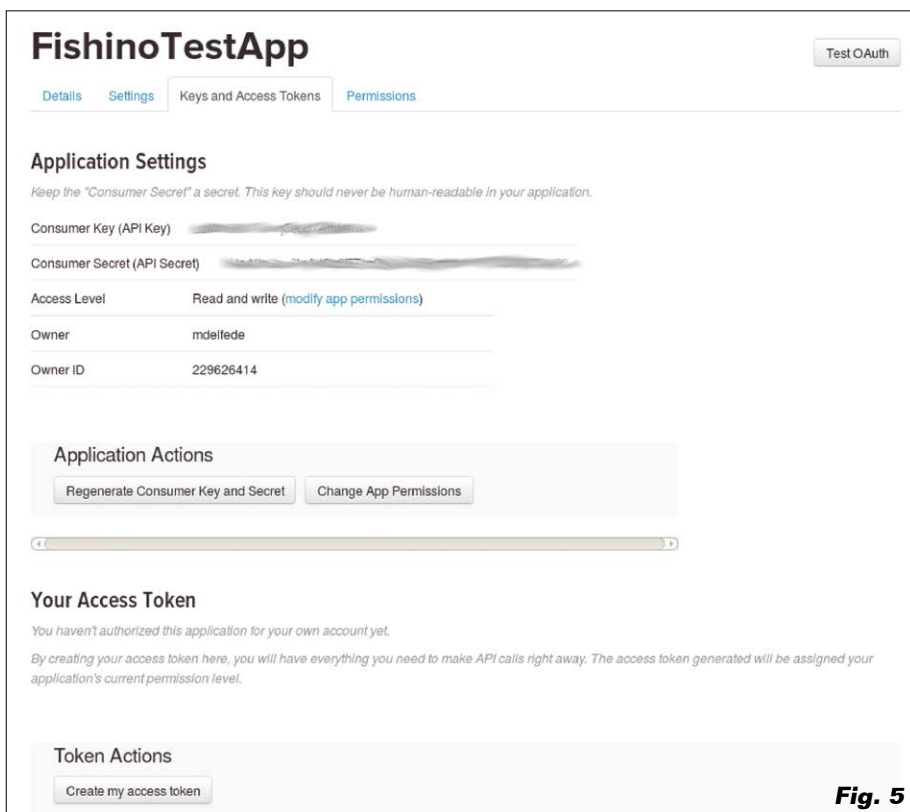
Description *
Prova di applicazione Twitter per Fishino
Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

Website *
<http://www.fishino.it>
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens. (If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Where should we return after successfully authenticating? OAuth 1.0a applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Fig. 4



FishinoTestApp

Details Settings Keys and Access Tokens Permissions

Application Settings

Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) [REDACTED]

Consumer Secret (API Secret) [REDACTED]

Access Level: Read and write (modify app permissions)

Owner: mdelfede

Owner ID: 229626414

Application Actions

Regenerate Consumer Key and Secret Change App Permissions

Your Access Token

You haven't authorized this application for your own account yet.

By creating your access token here, you will have everything you need to make API calls right away. The access token generated will be assigned your application's current permission level.

Token Actions

Create my access token

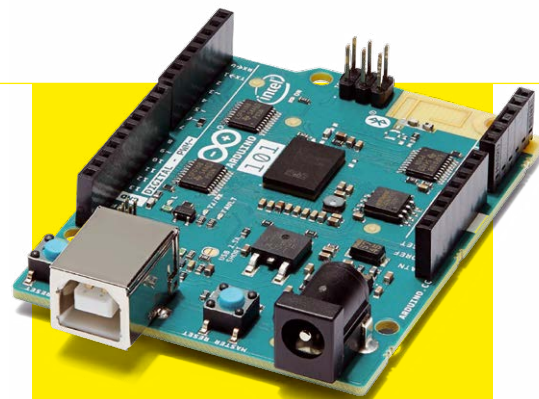
Fig. 5

richiesta https tipica alle API di Twitter, che andremo a esaminare in dettaglio. La prima riga è la richiesta vera e propria:

- **GET /1.1/search/tweets.json?** è la richiesta alle API di ricerca dei tweet;
- **count=1:** è il numero di risul-

tati richiesti; ne vogliamo uno per volta;

- **q= %23fishino+from %3Amdelfede** significa che cerchiamo l'hashtag (%23, #) fishino proveniente (%3A,@) da mdelfede;
- **result_type=recent** significa



Arduino sposa Intel

All'ultimo Rome Maker Faire, Intel e Massimo Banzi hanno presentato Genuino 101, una scheda di sviluppo pensata per la scuola, il mondo dei maker e per quanti si avvicinano per la prima volta alla programmazione. Genuino 101 è basata su Intel Curie, un SoC a basso consumo dalle dimensioni ridottissime, nato per applicazioni wearable; esso implementa un modulo Bluetooth LE e un accelerometro giroscopio a 6 assi. Il core è un Quark x86 a 32 bit con clock a 32 MHz nel quale gira un sistema operativo RTOS ViperOS. La board utilizza anche un core ARC (Argonaut RISC Core) a 32 bit anch'esso con clock a 32 MHz. Per il resto la basetta ha lo stesso form-factor di Arduino UNO e dispone di 14 pin digitali di I/O (di cui 4 possono essere utilizzati come uscite PWM), 6 ingressi analogici, un connettore USB per la comunicazione seriale e per il caricamento degli sketch, un jack di alimentazione e un connettore ICSP con SPI e I²C. Le linee di I/O funzionano a 3,3V ma sono protette contro le sovratensioni fino a 5 V. Genuino 101 sarà adottata da Creative Technologies in the Classroom, programma di physical computing attualmente implementato in oltre 300 scuole.

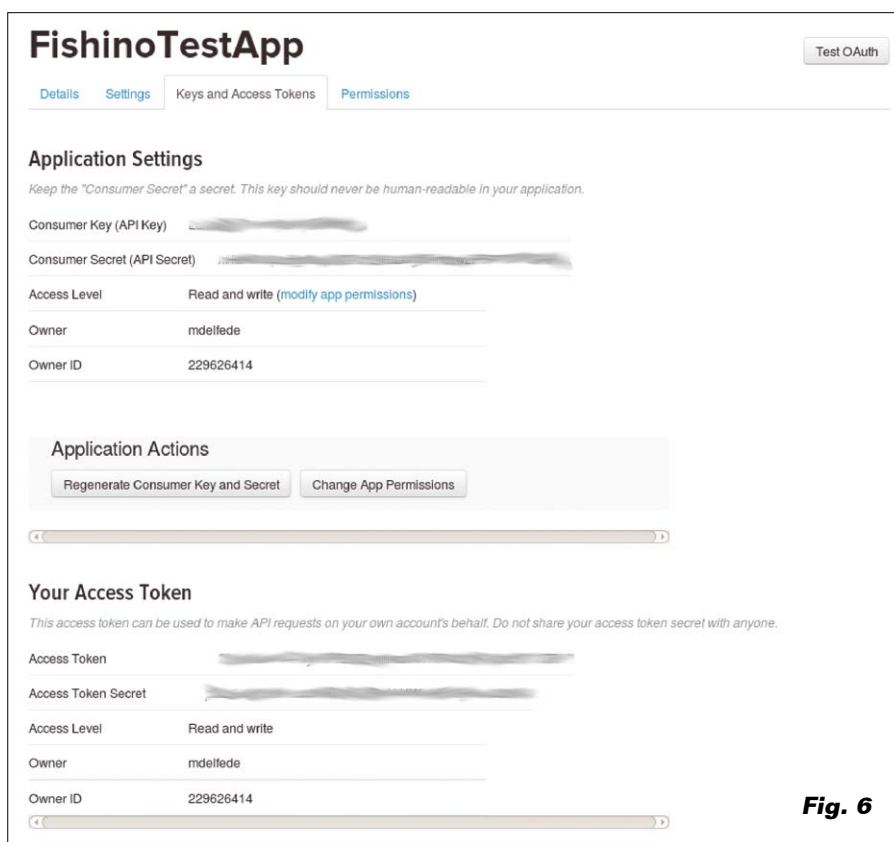


Fig. 6

che vogliamo i risultati in ordine di tempo e non di popolarità;

- **max_id= 647068135569387520;** gli ID sono numeri di sequenza contenuti in ogni messaggio Twitter, e che forniscono la sequenza temporale dei medesimi; in questo caso vogliamo cercare un messaggio non successivo al numero impostato;
- **since_id= 46038066440896517**

vuol dire che vogliamo un messaggio successivo al numero impostato;

- **HTTP/1.1** chiude la query http specificando il formato del protocollo.

La seconda e la terza riga indicano rispettivamente l'host cui viene inviata la richiesta (*api.twitter.com*) e l'applicazione che la

sta effettuando (Fishino Twitter Client); quest'ultima è arbitraria e serve solo ad identificare l'applicazione che utilizza le API. La quarta riga è molto importante, perché è quella utilizzata per l'autenticazione della richiesta sulla rete twitter:

Authorization:
Bearer AAAAAAAAAAAAAAAAAAAAAA
 (altri caratteri della chiave di autorizzazione)

Come già detto Twitter utilizza due metodi per rafforzare la sicurezza delle connessioni: il protocollo https (che impedisce gli attacchi tipo "man in the middle") e delle chiavi di autenticazione. Queste ultime sono di due tipi: personali o di applicazione (application-only); con il primo tipo di autenticazione il programma accede all'API come se fosse un utente specifico di Twitter ed agisce in suo nome; può, per esempio, inviare tweet come se fossero stati inviati da un PC. Il secondo tipo di autenticazione è invece legato all'applicazione, quindi parzialmente anonimo; non permette l'invio di messaggi ma solo alcune operazioni: per esempio la ricerca dei tweet. Nel nostro progetto abbiamo utilizzato il secondo metodo, che da un lato è più limitato perché Twitter pone dei limiti temporali all'utilizzo delle API: non è possibile eseguire oltre 180 operazioni ogni 15 minuti come utente o come applicazione. Quindi, se eseguiamo 180 richieste in 15 minuti come applicazione ce ne rimangono altrettante a disposizione come utente. 180 richieste ogni 15 minuti equivalgono ad un intervallo di 5 secondi tra 2 richieste.

Superare questi limiti vuol dire rischiare di farsi bloccare l'account Twitter, quindi è opportuno evitare richieste troppo frequenti.

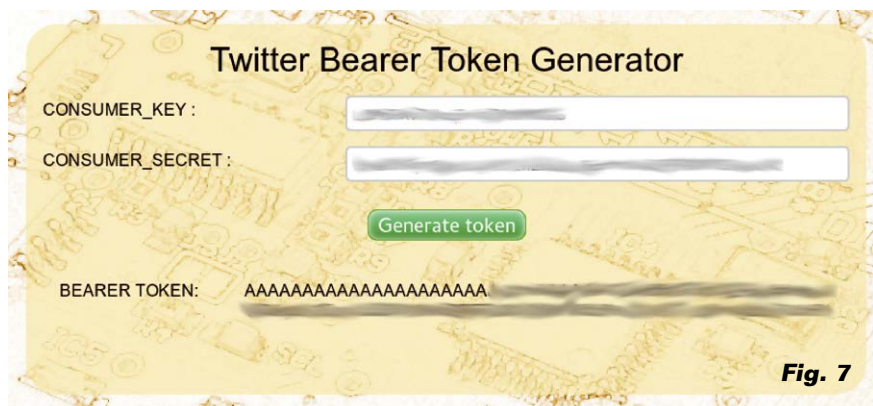


Fig. 7

Listato 2

```
////////////////////////////////////  
////////////////////////////////////  
// CONFIGURATION DATA          -- ADAPT TO YOUR NETWORK !!!  
// DATI DI CONFIGURAZIONE -- ADATTARE ALLA PROPRIA RETE WiFi !!!  
  
////////////////////////////////////  
////////////////////////////////////  
// CONFIGURATION DATA          -- ADAPT TO YOUR NETWORK !!!  
// DATI DI CONFIGURAZIONE -- ADATTARE ALLA PROPRIA RETE WiFi !!!  
  
// here put SSID of your network  
// inserire qui lo SSID della rete WiFi  
#define MY_SSID    "CASA"  
  
// here put PASSWORD of your network. Use "" if none  
// inserire qui la PASSWORD della rete WiFi -- Usare "" se la rete non è protetta  
#define MY_PASS    "unapassword"  
  
// here put required IP address of your Fishino  
// comment out this line if you want AUTO IP (dhcp)  
// NOTE : if you use auto IP you must find it somehow !  
// inserire qui l'IP desiderato per il fishino  
// commentare la linea sotto se si vuole l'IP automatico  
// nota : se si utilizza l'IP automatico, occorre un metodo per trovarlo !  
#define IPADDR     192, 168, 1, 251  
  
// here put your application Bearer authorization key  
// see https://dev.twitter.com/oauth/application-only for details  
// inserire qui la chiave "bearer" di autenticazione dell'applicazione  
// vedere https://dev.twitter.com/oauth/application-only per dettagli  
#define BEARER     "AAAAAAAAAAAAAAAAAAAAA....."  
  
// here put your Twitter user name; if empty the application will react to ANY user  
// inserire qui il nome utente ai cui tweets il programma si attiva; se vuoto, il programma si attiva con TUTTI gli utenti  
#define TWITTER_USER "pippo"  
  
// here put twitter hashtag needed to trigger the event (without the # char!)  
// inserire qui l'hashtag necessario ad avviare l'applicazione (senza il carattere #!)  
#define TWITTER_HASHTAG "fishino"  
  
// here put the delay in milliseconds between requests  
// there is a limit of about 180 requests every 15 minutes, so don't put a too short time here  
// inserire qui il ritardo tra due richieste successive  
// c'è un limite di circa 180 richieste ogni 15 minuti, quindi non inserire un tempo troppo breve  
#define REQUESTS_INTERVAL 5000
```

OTTENERE UNA CHIAVE 'BEARER' DI AUTORIZZAZIONE

Vediamo come creare un'applicazione Twitter e ottenere la relativa chiave di autorizzazione. Innanzitutto occorre essere registrati a Twitter e aprire l'indirizzo <https://apps.twitter.com/>.

Facendo clic sul pulsante Create New App appare la schermata di Fig. 3. Qui occorre riempire alcuni campi obbligatori quali:

- **Nome applicazione (Name);** in questo caso, **FishinoTestApp**;
- **Descrizione applicazione**

(Description); in questo caso "Prova di applicazione Twitter per Fishino";

- **Sito web (Website)** è un sito che dovrebbe contenere i dettagli dell'applicazione; è obbligatorio, ma può essere anche solo un segnaposto; in questo caso è www.fishino.it;
- **Callback URL**; non necessaria per i nostri scopi (lasciare in bianco).

In fondo alla pagina ci sono le usuali condizioni del servizio da accettare (selezionare la

checkbox) e il pulsante per creare l'applicazione (Create your Twitter application); cliccando appare la pagina di Fig. 4.

L'applicazione è stata creata e viene fornita (qui l'abbiamo oscurata) la 'consumer API key', che è una delle chiavi necessarie all'utilizzo delle API. Poiché vogliamo usare la 'Bearer token', ovvero la chiave necessaria in modalità 'Application only', serve un ulteriore passo: in alto cliccate sul tab 'Keys and access tokens'; vi apparirà la schermata in Fig. 5, che informa che l'applicazione

Listato 3

```
if (!client.connected() && !client.available())
{
    Serial.println("\nStarting connection to server...");
    if(client.connect("api.twitter.com", 443))
    {
        Serial.println("connected to server");

        // run the query looking from given hashtag and user
        doQuery(client, "%23" TWITTER_HASHTAG);
    }
}
```

Listato 4

```
// run a query on twitter
// esegue la query a Twitter
void doQuery(FishinoSecureClient &client, const char *query)
{
    // if walkback is finished (nextID == -1) we reset sinceID to next found maximum ID
    // otherwise we let it as it is and continue the walk back
    // se il walkback è finito (nextID == -1) re-impostiamo il sinceID a dopo
    // il massimo ID trovato
    // altrimenti lasciamo tutto come sta e continuiamo col walkback
    if(nextID == (uint64_t)-1)
        sinceID = maxID;

    client.print("GET /1.1/search/tweets.json?count=1&q=");
    client.print(query);
    if(strlen(TWITTER_USER))
    {
        client.print("&from%3A");
        client.print(TWITTER_USER);
    }
    client.print("&result_type=recent");
    if(nextID != (uint64_t)-1)
    {
        client.print("&max_id=");
        client << nextID;
    }
    client.print("&since_id=");
    client << sinceID;
    client.println(" HTTP/1.1");

    client.println("Host: api.twitter.com");
    client.println("User-Agent: Fishino Twitter Client");

    // send auth key to server
    // invia la chiave di autorizzazione al server
    client.println("Authorization: Bearer " BEARER);
    client.println();

    // reset nextID - it will be filled by parser if more results are pending
    // azzerare il nextID - verrà riempito dal parser se ci sono altri risultati da leggere
    nextID = -1;
}
```

non è ancora stata autorizzata (voce sotto **'Your Access Token'**). Facendo clic sul pulsante **'Create my access token'** in basso si ottiene la schermata di **Fig. 6**. Sotto la voce **'Your Access Token'** appaiono ora due chiavi che vanno utilizzate per l'uso dell'applicazione in modalità normale (**non Application-only**), vale a dire come se foste voi ad eseguire

i comandi. L'ultimo passo per ottenere la chiave **Application-only**, che è poi quella che useremo nella nostra applicazione; sarebbe il più complicato, visto che nelle pagine della documentazione delle API viene spiegato come fare ma, purtroppo, non viene fornito un sistema per farlo in modo semplice. Abbiamo quindi provveduto a realizzare

una pagina all'indirizzo *www.fishino.it/bearer.php* che permette di ottenere in modo facilissimo la cosiddetta **'Bearer token'**: È sufficiente inserire la **'consumer_key'** e la **'consumer_secret'** dell'applicazione, ricavate in precedenza, e premere il pulsante **'Generate token'** per ottenere la chiave da copiare ed incollare nello sketch. Ultima nota: la chiave appare sullo schermo in due linee per motivi di larghezza di pagina, ma è da copiare e incollare in una sola linea.

LO SKETCH PER FISHINO

Passiamo ora ad esaminare lo sketch (**Listato 2**) partendo dall'inizio del file, dove abbiamo scelto di configurare pochi elementi per dare modo agli "impazienti" di provare subito il sistema. Le impostazioni sono poche e semplici; chi già conosce le applicazioni dimostrative di Fishino noterà che lo stile è sempre lo stesso e i contenuti variano di poco; in particolare, verso la fine del gruppo di impostazioni si notano quelle rilevanti per Twitter, ovvero la Bearer Token di cui abbiamo parlato al paragrafo precedente (**BEARER**), il nome utente (**TWITTER_USER**) da cui devono provenire i tweet, l'hashtag che attiva il programma (**TWITTER_HASHTAG**), da inserire senza il carattere di cancelletto (#) e l'intervallo tra due richieste al sistema. Per quanto riguarda quest'ultimo, per rispettare il limite di 180 richieste in 15 minuti abbiamo impostato una richiesta ogni 5 secondi circa (#define REQUESTS_INTERVAL 5000). Sugeriamo di non scendere sotto il valore 5.000 (millisecondi) se non per brevi prove.

DESCRIZIONE DELLO SKETCH

Una buona parte dello sketch si occupa dell'inizializzazione della

Listato 5

```
// analyze data and act on devices
// analizza i dati in arrivo da Twitter
void parserCallback(uint8_t filter, uint8_t level, const char *name, const char *value)
{
    // check for user name
    // controlla il nome utente
    if(level == 3 && !strcmp(name, "screen_name") && !strcmp(value, "\"" TWITTER_USER "\""))
        gotUser = true;

    // check twitter message - MUST contain the requested hashtag, a port number and word ON or OFF
    // controlla il testo del messaggio - DEVE contenere l'hashtag richiesto, un numero di porta e le parole ON o OFF
    else if(level == 2 && !strcmp(name, "text") && value && value[0] && !strnicmp(value + 1, "#" TWITTER_HASHTAG, strlen(TWITTER_HASHTAG) + 1))
    {
        // hashtag found, skip it
        // trovato l'hashtag, lo salta
        const char *p = value + strlen(TWITTER_HASHTAG) + 3;

        // skip spaces after hashtag
        // salta gli spazi dopo l'hashtag
        while(*p && isspace(*p))
            p++;

        // read port number - sets it to 0xff if not found
        // legge il numero di porta - lo imposta a 0xff se non trovato
        if(isdigit(*p))
        {
            port = 0;
            while(isdigit(*p))
                port = port * 10 + *p++ - '0';
        }
        else
            port = 0xff;

        // if port not found, just leave
        // se non trova il numero di port ritorna
        if(port == 0xff)
            return;

        // skip spaces after port number
        // salta gli spazi dopo il numero di porta
        while(*p && isspace(*p))
            p++;

        // check for ON or OFF keywords - if not found sets port to 0xff and leave
        // cerca le parole chiave ON o OFF - se non trovate imposta la porta a 0xff ed esce
        if(!strnicmp(p, "on", 2))
            command = true;
        else if(!strnicmp(p, "off", 3))
            command = false;
        else
            port = 0xff;
    }

    // look if there are previous tweets to handle
    // cerca se deve elaborare tweets precedenti
    else if(level == 1 && !strcmp(name, "next_results"))
    {
        // get next tweet id to handle
        const char *s = strstr(value, "\"?max_id=");
        if(s)
        {
            s += 9;
            nextID = readID(s);
        }
    }

    // get max id of search, needet to go further after walk back
    // cerca il massimo ID corrispondente alla ricerca, necessario per proseguire dopo la ricerca all'indietro
    else if(level == 2 && !strcmp(name, "id"))
    {
        uint64_t id = readID(value);
        if(id > maxID)
            maxID = id;
    }
}
```

Listato 6

```
// waits for data from twitter and parse them - return true on success, false if no data
bool doReceive(FishinoSecureClient &client, JSONStreamingParser &parser)
{
    // wait 1 second max for data from server
    // attende al massimo 1 secondo i dati dal server
    unsigned long tim = millis() + 1000;
    while(!client.available() && millis() < tim)
        ;

    // if no data is available, close the connection
    // se non ci sono dati, chiude la connessione
    if(!client.available())
        return false;

    // skip data up to 2 consecutive cr
    bool cr = false;
    while(client.available())
    {
        if(cr)
        {
            if(client.read() == 0x0d && client.read() == 0x0a)
                break;
            else
                cr = false;
        }
        else if(client.read() == 0x0d && client.read() == 0x0a)
            cr = true;
    }

    while(client.available())
    {
        char c = client.read();
        parser.feed(c);
    }
    return true;
}
```

scheda, della connessione alla rete WiFi, eccetera.

Per brevità rimandiamo agli articoli di presentazione di Fishino; qui vediamo in dettaglio solo le parti più interessanti dello sketch, ovvero la gestione della connessione sicura alle API di Twitter e l'utilizzo delle schede di espansione I/O.

Per quanto riguarda la connessione alle API, notiamo subito una differenza rispetto agli sketch visti in precedenza:

```
// The web secure client used
to access twitter API
// il client https usato per
l'accesso all'API di Twitter
FishinoSecureClient client;
```

Qui si nota l'utilizzo del FishinoSecureClient al posto del precedente FishinoClient. Questo client viene utilizzato per le connessioni sicure https.

La connessione alle API avviene tramite lo spezzone di codice contenuto nel **Listato 3**: prima viene controllato se il client è ancora connesso dal loop precedente e/o se ha ancora dei dati da leggere, visto che i dati sono mantenuti in un buffer nel client anche dopo la chiusura dello stesso. In caso contrario viene aperta la connessione al sito *api.twitter.com* sulla porta 443, che è quella utilizzata nelle connessioni https.

Se la connessione ha esito positivo, viene inviata la richiesta vera e propria tramite la funzione `doQuery()`, che ha come parametri il client e l'hashtag da cercare. La funzione non fa altro che inviare i dati come visto durante la spiegazione delle API (**Listato 4**). La particolarità di questa funzione, a parte l'invio dei dati di ricerca veri e propri, sta nella gestione dei campi **since_id** e **max_id** che consentono di restringere la ricerca in ben precisi limiti temporali.

Listato 7

```
// if port has been set, change its output
// se la porta è stata impostata, ne modifica l'output
if(port != 0xff)
{
    // if no expanders found, check if pin is free on Fishino
    // and set its output (must be changed on Mega!)
    if(!numExpanders)
    {
        if(port == 4 || port == 7 || port >= 10)
            Serial << F("INVALID PORT #") << port << F(" -- COMMAND DISCARDED\n");
        else
        {
            Serial << "\n\nSETTING PORT " << port << " TO " << (command ? "HIGH" : "LOW") << "\n";
            pinMode(port, OUTPUT);
            digitalWrite(port, command ? HIGH : LOW);
        }
    }
    else
    {
        uint8_t board = port / 8;
        if(board >= numExpanders)
            Serial << F("INVALID PORT #") << port << F(" -- COMMAND DISCARDED\n");
        else
        {
            Serial << "\n\nSETTING PORT " << port << " TO " << (command ? "HIGH" : "LOW") << "\n";
            port %= 8;
            expanders[board]->digitalWrite(15-port, command ? HIGH : LOW);
        }
    }
}
```


Per motivi di spazio e velocità, ad ogni richiesta scegliamo infatti di ricevere un solo tweet, che è l'ultimo presente in rete tra i due ID forniti; giocando su questi ultimi è possibile andare a ritroso richiedendo i tweet precedenti, fino all'esaurimento dei medesimi. Finita la ricerca all'indietro, viene reimpostato il **since_id** in modo che nelle richieste successive vengano selezionati solo i tweet più nuovi.

Una volta inviata la richiesta, occorre ricevere ed analizzare i dati. Le API di Twitter rispondono con una serie di dati in formato JSON e di dimensioni molto variabili, anche piuttosto grandi. Vista la limitatezza delle risorse di memoria della scheda, una memorizzazione di questi dati è impensabile; occorre quindi analizzarli man mano che arrivano. Per far questo abbiamo scritto una libreria completa per il parsing (scansione) dei dati in arrivo in modalità streaming, ovvero carattere per carattere, senza memorizzazioni intermedie.

Questa libreria, **JSONStreamingParser**, è diversa dalla maggior parte delle librerie analoghe presenti in rete visto che, oltre all'analisi dei dati in tempo reale, non memorizza nemmeno i risultati, se non quello corrente.

In pratica, i dati JSON hanno un formato simile a questo:

```
{
  nome1 : dato1,
  nome2 : {nome3 : dato3, nome4 : dato4},
  nome5 : [dato6, dato7, dato8]
}
```

Ovvero, i dati possono contenere valori semplici (stringhe, numeri, eccetera) oppure liste di dati oppure array di dati. Il formato, arrivando ai dati semplici, è comunque sempre lo stesso:

```
nome : dato
```

Listato 8

```
// find all available IOExpander boards
void findIOExpanders(void)
{
  numExpanders = 0;
  for(uint8_t addr = 0; addr < 8; addr++)
  {
    IOExpanderMCP23017 *exp = new IOExpanderMCP23017;

    // set expander address
    exp->begin(addr);

    // this is because expander board hasn't reset connected to arduino one
    // and it checks register 0 upon init to see if board is present
    // (register should be 0xff, all A port as input)
    for(int i = 0; i < 16; i++)
      exp->pinMode(i, INPUT);

    if(exp->init())
    {
      // expander found, add to list and setup pins
      expanders[numExpanders++] = exp;

      // set pin modes
      for(int i = 0; i < 8; i++)
        exp->pinMode(15-i, OUTPUT);
    }
    else
      // not found, delete expander object
      delete exp;
  }
  if(numExpanders)
    Serial << F("Found ") << numExpanders << F(" I/O expander boards\n");
  else
    Serial << F("No expander boards found - using Fishino's I/O pins\n");
}
```

Il parser, quando determina un nome, esegue la scansione del dato; se questo è un tipo semplice, chiama una funzione predefinita con i seguenti parametri:

```
void callback(uint8_t filter,
uint8_t level, const char *name,
const char *value
```

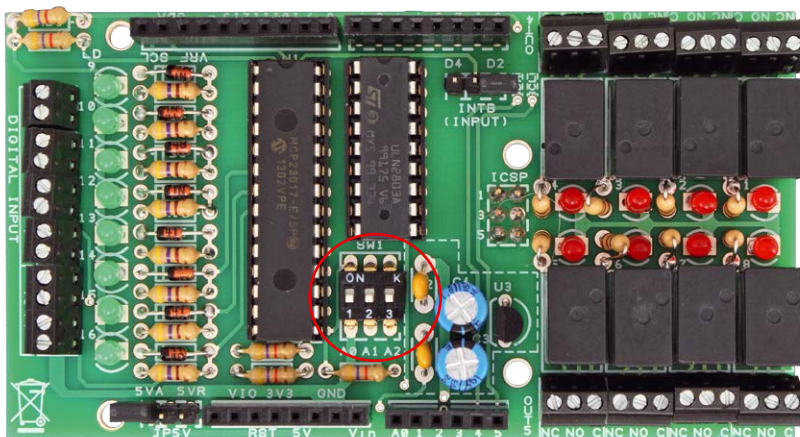
dove il parametro 'filter' è previsto per future espansioni, il parametro 'level' indica il livello di nidificazione del dato ed i parametri 'name' e 'value' indicano nome e valore del dato. Se il dato non è semplice ma costituito da una lista o da un array, il campo 'value' assume il valore "<LIST>" o "<ARRAY>" e la scansione prosegue al suo interno. Creando opportune funzioni di **callback** è quindi possibile analizzare al volo il contenuto del **JSON** ricevuto ed impostare le

variabili opportune nello sketch. L'attuale funzione di callback è contenuta nel **Listato 5**.

La funzione sembra complessa, ma in realtà non fa altro che sfruttare i parametri 'level' e 'name' per determinare se sono presenti o meno i campi richiesti nel tweet e, nel caso, ne estrae ed analizza i valori.

Qui si nota che viene controllato l'utente che ha inviato il tweet, gli hashtag per vedere se c'è quello impostato, il parametro 'text' per leggere i comandi inviati e gli ID utilizzati per controllare la sequenza temporale dei tweet.

Come dicevamo, la funzione viene richiamata dal **JSONStreamingParser** ogni volta che questo legge un valore dallo stream. Il parser, a sua volta viene utilizzato dalla funzione **doReceive()**



Con SW1 si imposta l'address I²C dello shield I/O Expander: notate la corrispondenza tra A0, A1, A2 e i singoli dip.

come mostrato nel **Listato 6**.

Questa funzione non fa altro che leggere i dati dal client, saltare il preambolo HTTP (la cui fine è indicata da una linea vuota) ed inviare i restanti caratteri, uno alla volta, al parser, per l'analisi. Una volta completata l'analisi dello stream di dati, in caso di comando valido la variabile globale 'port' conterrà il numero di porta da modificare e la variabile globale 'command' (booleana) sarà impostata a true se vorremo portare l'uscita a, valore HIGH, oppure false se la vorremo a LOW.

Vediamo la parte che attiva l'uscita, interessante per il fatto che prevede la possibilità di applicare gli I/O Expander a Fishino (**Listato 7**). Qui si nota per prima cosa il controllo sulla validità della porta (port); se è a 0xff il comando è da scartare, altrimenti viene proseguita l'analisi. Successivamente il codice agisce in modo differente se sono presenti o meno le espansioni. Nel primo caso, determina il numero di scheda di espansione (port / 8) e la porta sulla medesima (port % 8). Per esempio, un valore di port pari a 33 corrisponde alla scheda n° 4 (33/8) ed alla sua porta n° 1 (33 % 8 = 33 - 4 •

8 = 1). L'operatore '%' fornisce infatti il resto della divisione. Nel caso non siano presenti le espansioni, lo sketch utilizza il numero di porta direttamente come I/O digitale del Fishino; controlla che sia un valore valido e che non vada a modificare pin utilizzati per altri scopi ed agisce su quello.

UTILIZZO DELLE SCHEDE DI ESPANSIONE I/O

Lo sketch prevede, in modo completamente automatico, il riconoscimento di eventuali schede I/O Expander anche connesse in cascata, fino ad un massimo di 8. L'importante è avere l'accortezza di fornire indirizzi I²C differenti ad ogni scheda.

All'avvio lo sketch controlla, nel setup(), se sono presenti le schede e le numera in base all'indirizzo I²C. Se ne trova, assegna i port di uscita in sequenza: da 0 a 7 per la prima scheda, da 8 a 15 per la seconda e così via.

Nel caso non venga rilevata alcuna scheda di espansione, lo sketch va in modalità standalone utilizzando tutti gli I/O liberi di Fishino, e quindi le uscite 0, 1, 2, 3, 5, 6, 8 e 9, essendo le altre riservate al funzionamento delle periferiche incluse.

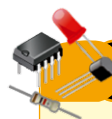
In questo caso il numero di porta corrisponde al numero dell'I/O digitale di Fishino.

La numerazione delle schede avviene tramite lo spezzone di codice contenuto nel setup (**Listato 8**).

Si nota il ciclo che prova tutti gli indirizzi I²C possibili (da 0 a 7); per ogni indirizzo crea una variabile IOExpanderMCP23017 (in modo dinamico, come puntatore), ne esegue il test di presenza e, nel caso funzioni la memorizza in un array di puntatori. Se la scheda non è presente elimina la variabile e riparte con la successiva. In questo modo le schede vengono rilevate in ordine di indirizzo I²C.

Con questo abbiamo concluso la descrizione dell'applicazione **Fish'n Tweets!**

In un prossimo articolo vedremo come fare l'operazione inversa, ovvero far inviare dei tweet alla scheda quando alcuni pin di ingresso cambiano di stato, cosa utilissima per esempio per realizzare degli allarmi o monitorare eventi in un luogo remoto. ■



per il MATERIALE

Tutto il materiale necessario per realizzare questo progetto è disponibile presso Futura Elettronica. Lo shield Arduino I/O expander in kit (cod. FT1079K) è disponibile a Euro 34,00. La board Fishino (cod. FISHINOUNO) viene fornita montata e collaudata al prezzo di Euro 36,00. Software e sketch sono scaricabili gratuitamente sul sito www.elettronica.in.it. 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>

PRESEPINO: IL PRESEPE CON ARDUINO

di BORIS LANDONI





Centralina capace di simulare ciclicamente l'avvicinarsi del giorno e della notte, pilotando opportunamente fino a sei uscite (con cui alimentare LED o piccoli motori) corrispondenti alla luce del giorno, al bagliore delle stelle, al fuoco delle case, all'accensione della stella cometa e a due dispositivi mobili. Può, inoltre, controllare strip NeoPixel e, per rendere l'atmosfera più natalizia, riproduce musiche a vostra scelta.

Come ogni anno, all'approssimarsi delle Feste Natalizie pensiamo a un progetto a tema che possa renderle originali e più calorose. Dato che da tanti anni non proponevamo più quello che è poi uno dei classici dell'elettronica -il controllo delle luci del presepe- abbiamo deciso di optare per una centralina del genere, ma ci è sembrato giusto farlo in chiave moderna ed attuale; e per attuale intendiamo con un hardware conosciuto e apprezzato come quello che migliaia di sperimentatori hanno trovato in Arduino. Ecco nascere, così, Presepio, ossia una centralina di controllo luci e sonoro per presepe, il cui circuito è basato su un ATmega 328 (quindi l'hardware di Arduino UNO) e nasce per comandare direttamente dei LED, con tutti i vantaggi del caso, primo fra tutti il fatto che lavora a bassa tensione e può quindi essere collocata anche a portata dei bambini, senza che ne derivi pericolo per la loro salute. Poi oggi, con la grande disponibilità di strip a LED, realizzare un pre-

sepe automatizzato piccolo o grande che sia, risulta semplicissimo. Oltre alle luci, il nostro Presepio permette di gestire anche l'audio, grazie a un innovativo modulo riproduttore di MP3 montato sulla scheda, capace di farvi ascoltare in altoparlante (seppure in mono) i brani contenuti in una SD-Card che potrete scegliere secondo le vostre preferenze.

IL PROGETTO

Ma andiamo con ordine e vediamo innanzitutto le funzioni disponibili, che differenziano la nostra centralina da quelle per il semplice controllo delle due luci che realizzano la simulazione di alba e tramonto: con Presepio è possibile realizzare un controllo che va oltre il piccolo o grande presepe di casa, per arrivare ai presepi allestiti nelle chiese, negli oratori, nei locali delle comunità e nei luoghi aperti. Per questo motivo abbiamo realizzato un circuito decisamente prestante, in grado di pilotare quattro carichi luminosi a LED funzionanti a 12 volt e che assorbano una corrente

individuale di 6 ampere; questa è anche la massima corrente che la scheda può gestire e può essere aumentata stagnando abbondantemente le piste che portano dalla morsettiera di alimentazione ai MOSFET e da questi alle relative morsettiere di uscita. Per corrente individuale si intende che ciascun canale può commutare 6 ampere, ma per come è costruito lo stampato non è possibile assorbire 6 ampere costantemente da tutte le uscite insieme. Questi valori, considerato che il circuito nasce per pilotare strip a LED o comunque composizioni di LED, sono più che accettabili e per un presepe da casa sono sovrabbondanti, pur bastando anche per una struttura all'aperto, grazie all'efficienza luminosa dei LED. Le uscite della centralina sono sei, ma il firmware che forniamo contempla la gestione delle prime quattro per implementare la simulazione della luce del giorno, di quella delle stelle, dei fuochi della capanna/grotta (di Betlemme) e delle case dei pastori, ed infine di illuminare la stella cometa. Le luci si accendono e spengono gradatamente seguendo un ciclo che simula un'intera giornata. A queste uscite di base potete aggiungere, intervenendo sullo sketch per l'ATmega 328, le altre due, cui potete assegnare il controllo di vari accessori del presepe come una ruota a pale, il movimento di personaggi animati o animali, ecc.

Per quanto riguarda le uscite di base, abbiamo suddiviso la sequenza completa in quattro fasi denominate giorno, tramonto, notte e alba. La durata delle fasi può essere regolata mediante trimmer, e l'escursione corrispondente si può reimpostare intervenendo sullo sketch (ma in maniera uguale per tutte); ciò vale dunque per il giorno e la

notte, ma anche per le due fasi di transizione (tramonto e alba). Queste ultime due sono le fasi più suggestive: durante il tramonto, la luminosità del giorno (realizzata con i LED collegati all'uscita SOLE) diminuisce a poco a poco, mentre nel cielo iniziano a illuminarsi le stelle (si attiva l'uscita omonima); a un certo punto, prima che il ciclo si concluda, si accendono i fuochi delle case e della capanna (uscita FUOCO). Tra l'altro, il nostro circuito è in grado di simulare il tremolio della legna che arde, pilotando con un'opportuna sequenza di impulsi l'uscita FUOCO. Quando tutte le stelle in cielo sono completamente illuminate, appare anche la stella cometa (accesa dall'uscita COMETA). Ovviamente, salvo piccoli dettagli, durante l'alba si spengono gradatamente tutte le luci luminose, mentre aumenta lentamente sino a raggiungere la massima luminosità la luce del giorno.

SCHEMA ELETTRICO

Scendiamo ora negli aspetti più tecnici, analizzando il circuito, che è basato sul microcontrollore Atmega 328 nel quale gira l'apposito sketch che implementa il controllo delle uscite e del modulo che realizza il sonoro del presepe; il micro, siglato U1, all'inizializzazione imposta PB0 come uscita per l'invio dei comandi seriali a un'eventuale strip Neopixel, e PB1, PB2, PB3, PD3, PD5, PD6 come output per il controllo delle uscite a MOSFET a canale N (i transistor sono dei robusti STP36N06, da 36 ampere di I_d e 60V di V_{ds}) che materialmente commutano l'alimentazione sui LED o strip a LED corrispondenti. Le linee del micro utilizzate per gestire le uscite pilotano ciascuna il gate

del proprio MOSFET attraverso un resistore da 1 kohm e ciascuna di esse e massa è applicato un LED (con rispettiva resistenza di limitazione della corrente) che accendendosi permette di monitorare lo stato del rispettivo transistor; ciascuna delle uscite fornisce impulsi rettangolari TTL-compatibili di frequenza costante e modulati in larghezza, secondo la tecnica PWM: ciò consente di variare il valore medio della tensione applicata dai MOSFET ai carichi loro collegati e di conseguenza la potenza e la luminosità, con un'elevatissima efficienza in quanto la resistenza in stato di ON (piena conduzione) dei MOSFET è bassissima (appena 40 milliohm) e quindi la potenza dissipata, a parità di potenza trasferita, è limitatissima se confrontata a quella che comporterebbe un pilotaggio serie a tensione continua, dove il transistor dovrebbe sobbarcarsi la differenza tra la tensione di alimentazione e quella applicata al carico.

I segnali PWM vengono generati dai moduli PWM interni al microcontrollore, opportunamente impostati dal firmware per ciascuna uscita.

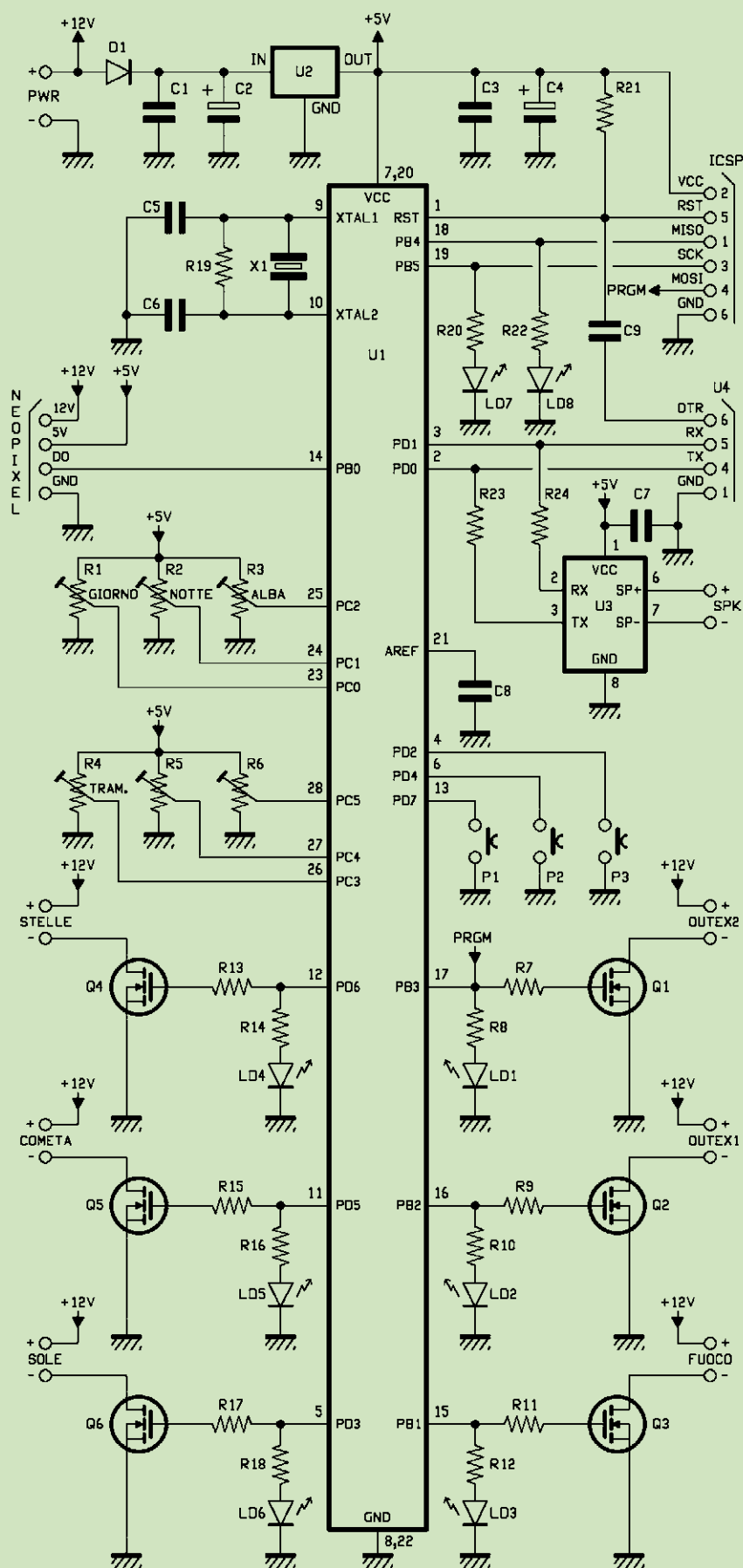
L'attuale configurazione circuitale prevede di alimentare le strip di LED collegandone positivo e negativo rispettivamente a + e - della relativa uscita, tuttavia siccome i MOSFET possono "reggere" tensioni drain-source in interdizione pari a 60 volt, è anche pensabile partire da un'alimentazione esterna anche a 24÷36 volt c.c. in modo da ridurre l'assorbimento a parità di potenza elettrica gestita. Se l'alimentazione non viene presa dal circuito, la striscia di LED va connessa con il + al positivo dell'alimentatore scelto e il - all'uscita della nostra centralina, fermo restando che la

massa dell'alimentatore e della scheda vanno unite.

Abbiamo detto in precedenza che la sequenza completa generata dal nostro circuito è composta da quattro fasi (giorno, tramonto, notte e alba) la cui durata può essere regolata indipendentemente tramite gli appositi trimmer alimentati a 5 volt e la cui tensione del cursore viene letta da un I/O del microcontrollore configurato come input e acquisito tramite l'A/D converter interno all'ATmega; nel nostro progetto, l'ADC viene assegnato alle linee PC0÷PC5, con le quali si acquisiscono in scansione le tensioni dei sei trimmer presenti, vale a dire R1, R2, R3, R4 per le funzioni anzidette ed R5 e R6 disponibili per funzioni assegnabili a piacimento programmando opportunamente il microcontrollore.

Per come è strutturato il firmware, la durata dei rispettivi cicli è direttamente proporzionale alla tensione applicata dal relativo trimmer, quindi portando il cursore di R1 verso massa, per esempio, si accorcia la durata del giorno, la quale invece aumenta portandolo verso il positivo dei 5 V.

Attenzione ad un particolare: i trimmer R1, R2, R3, R4 non controllano direttamente il PWM delle uscite per le fasi, ma ne determinano i tempi di esecuzione; il duty-cycle dei segnali PWM alle uscite SOLE, STELLE, COMETA e FUOCO viene gestito secondo la periodicità prevista da apposite subroutine del firmware. Questo funzionamento è logico in quanto le fasi descritte coinvolgono più di un'uscita e, nel caso della transizione giorno/notte (TRAMONTO), per esempio, coinvolgono l'uscita SOLE e quella STELLE, che vanno in dissolvenza, come è, anche nel passaggio notte/giorno (ALBA),



Listato 1

```
if(runtime>time_trim+200){
  for (int i=0;i<6;i++){
    trimvalue[i]=(ReadTrimmer(i));
    trimvalue[i]=trimvalue[i]*100;
    delay (1);
  }
  time_trim=millis();
}

int ReadTrimmer (int id){
  return (analogRead(trimmer[id]));
}
```

seppure al contrario.

I trimmer da R1 a R4 servono dunque per gestire la durata del GIORNO (R1), della NOTTE (R2), dell'ALBA (R3) e del TRAMONTO (R4). Il led LD8, pilotato dalla linea PB4, si accende brevemente quando la centralina passa da una fase all'altra.

La durata di ogni singola fase viene impostata tramite l'apposito trimmer e la durata massima di ogni fase è di circa 100 secondi; per l'esattezza, la durata è ottenuta dalla lettura dell'ADC su 10 bit (tra 0 e 1.023) considerata in millisecondi, moltiplicata per 100 dal firmware. Se volete tempi differenti potete editare lo sketch nella porzione che si occupa di determinare la temporizzazione, che per praticità riportiamo nel **Listato 1**. Si tratta sostanzialmente della riga:

*trimvalue[i]=trimvalue[i]*100;*

Qui potete modificare il valore 100 sostituendolo con ciò che vi aggrada: ad esempio, inserendo 200 ottenete il raddoppio dei tempi, mentre con 50 li dimezzate.

Quanto agli altri due trimmer, sono personalizzabili da firmware, così come le uscite OUTEX1 e OUTEX2; con lo sketch da noi fornito, R5 è libero mentre R6 è stato utilizzato per gestire il volume di riproduzione dell'audio. Alle uscite a MOSFET possono essere collegate sia delle strip a LED, sia lampadine a 12 volt di

piccola potenza, ma anche singoli LED, motori elettrici per azionare ruote e altri meccanismi, come anche pompe per far muovere l'acqua in un laghetto o un fiume e tutto quello che la fantasia suggerisce per realizzare un presepe più "vivo".

Restando in tema uscite, è il caso di parlare di quella dedicata al controllo di LED NeoPixel, che permette di aggiungere ulteriori animazioni luminose, come ad esempio la coda pulsante di una cometa. Le strip NeoPixel vengono gestite in parallelo con una sola linea del microcontrollore (più in generale, di Arduino), che nel nostro caso è facilmente assegnabile a piacimento specifican-

dola nello sketch; la comunicazione è unidirezionale e gestisce un gruppo di LED.

NeoPixel è una particolare soluzione a LED RGB "intelligenti" per il controllo, ad esempio, tramite Arduino; ogni LED NeoPixel dispone di controller a bordo, interfacciabile con una sola linea dati. La libreria proprietaria che l'Adafruit (www.adafruit.com) rende disponibile liberamente facilita l'integrazione nell'ambiente Arduino.

Ogni LED RGB può essere gestito individualmente tramite un apposito comando inserito nella stringa seriale e può produrre fino a 256 tonalità del proprio colore, determinando un totale di

Listato 2

```
void gestAudio(int track){

  digitalWrite (LEDPLAY,LOW);
  if (playing==true){
    digitalWrite (LEDPLAY,HIGH);
    track++;
    if (track==5) track=1;
    if (stopVol==false){
      if(runtime>time_volume_running+200){
        if (startVol==false){
          if (volume>3) {
            volume=volume-3;
            Serial.print("dec ");
            Serial.println(volume);
          }
          else
          {
            playtrack(track);
            startVol=true;
          }
        }
        else
        {
          if (volume<(trimvalue[TRIMMER_VOL]/3500)) {
            volume=volume+3;
            Serial.print("inc ");
            Serial.println(volume);
          }
          else
          {
            startVol=false;
            stopVol=true;
            Serial.println("finito ciclo");
          }
        }
      }
      time_volume_running=millis();
      setvolume (volume);
    }
  }
}
```


16.777.216 combinazioni di colore. Una particolarità dei LED NeoPixel è che possono essere collegati in cascata in modo che la linea dati da uno passi al successivo, però il prezzo da pagare è che oltre un certo numero di LED, la velocità di gestione deve ridursi sensibilmente; a causa di ciò, se si devono realizzare matrici per mostrare della grafica veloce, occorre impiegare molte linee con pochi LED ciascuna. Ma questa limitazione non interessa il nostro progetto, nel quale peraltro non è stata prevista una funzione predefinita per l'uscita NeoPixel, che potrete invece aggiungere editando lo sketch.

Il canale dati usato per la comunicazione con i LED NeoPixel e quindi con le strip, è simile al tipo oneWire; la comunicazione avviene a un massimo di 800 kbps.

Per ogni strip di LED è possibile impostare a piacimento la frequenza di refresh, in modo da rendere impercettibili determinati giochi di luce, fermo restando il limite dettato dalla quantità di strip collegate.

Il protocollo di comando del sistema NeoPixel prevede l'invio di gruppi di tre byte in una stringa di 24 bit, ognuna delle quali contiene lo stato di illuminazione di ciascun colore base (prima gli otto bit del verde, poi quelli del rosso e infine quelli del blu). L'alimentazione prevista per i LED NeoPixel è a 5 volt, prelevabili dal contatto 5V del connettore della scheda Presepio, dato che l'assorbimento di ciascuna strip normalmente non è elevato e che i LED tricolore NeoPixel vengono accesi alternativamente. La massa di riferimento di alimentazione e dati (è unica e fa capo al contatto G della strip) è sempre quella del microcontrollore e va quindi connessa a

Il componente che utilizziamo per la riproduzione dell'audio permette di far riprodurre brani musicali MP3 senza dover utilizzare hardware costoso e impegnativo sul fronte dell'elaborazione: infatti il DFR0299 è un completo decoder che attinge a un supporto SD-Card, decifra l'audio e poi lo amplifica on-board pronto per inviarlo a un altoparlante (da 8 ohm di impedenza) con quasi 3 watt di potenza. È quindi l'ideale quando si desiderano riprodurre file audio standard come gli MP3 ma non si dispone delle risorse di calcolo (CODEC compreso) che la loro elaborazione diretta richiederebbe.

Il modulo nasce per essere gestito da Arduino, il quale di fatto si limita a inviargli comandi seriali per gestire la riproduzione; tutto il resto lo fa il DFR0299. Questo dispositivo può essere utilizzato in stand-alone, alimentandolo a 5 volt (la corrente richiesta per erogare 3 watt su 8 ohm, che si ottengono a 5 volt, è 0,45 ampere), collegandogli un altoparlante e dei pulsanti. Oppure può essere usato in combinazione con un Arduino UNO o qualsiasi altro dispositivo dotato di porta seriale TTL. Dispone di controllo tramite pin di I/O, tramite seriale o tramite ingresso analogico. Supporta fino a 100 cartelle; ogni cartella può contenere fino a 255 brani MP3. La regolazione del volume, sia essa ottenuta per via seriale o localmente tramite pulsanti, prevede 30 livelli ($\pm$ dalla posizione centrale). L'alimentazione prevista è da 3,2V a 5V in continua (la corrente

Piedinatura del modulo DFR0299.



assorbita in standby è pari a 20 mA). Le principali caratteristiche del modulo sono:

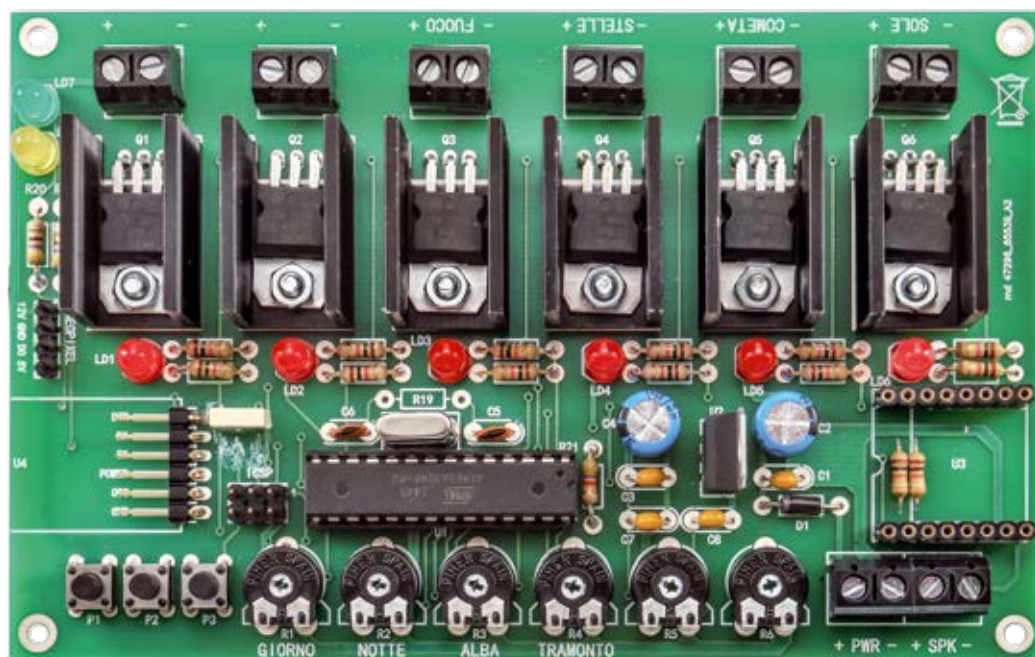
- decodifica MP3 11172-3 e ISO13813-3 layer3 audio decoding;
- possibilità di equalizzazione Pop, Rock, Jazz, Classica;
- frequenze di campionamento in riproduzione (kHz): 8/11,025/12/16/22,05/24/32/44,1/48
- DAC a 24 -bit, che supporta una gamma dinamica di 90 dB e un rapporto segnale/rumore di 85dB;
- supporto file system FAT16 e FAT32;
- massima capacità di memoria indirizzabile pari a 32 GB, 32G di USB disk, blocchi di 64 MB Flash NOR;
- dimensioni (mm): 20 x 20 x 13.

Per quanto nel nostro progetto lo utilizziamo in mono, il modulo dispone di un decoder MP3 stereo le cui uscite sono disponibili a basso livello e impedenza relativamente alta per pilotare amplificatori, ingressi di mixer, cuffie a 300 ohm ecc. L'amplificatore di potenza incorporato è mono e amplifica la miscelazione dei canali destro e sinistro. La tabella in questo riquadro riporta la piedinatura del modulo.



Il modulo può essere utilizzato autonomamente con soli quattro pulsanti: S1 e S2 hanno la duplice funzione di controllo volume (premuti a lungo) o di skip tra brani (pressione breve).

N°	Pin	Description	Note
1	VCC	Input Voltage	DC3.2~5.0V;Type: DC4.2V
2	RX	UART serial input	
3	TX	UART serial output	
4	DAC_R	Audio output right channel	Drive earphone and amplifier
5	DAC_L	Audio output left channel	Drive earphone and amplifier
6	SPK2	Speaker-	Drive speaker less than 3W
7	GND	Ground	Power GND
8	SPK1	Speaker+	Drive speaker less than 3W
9	IO1	Trigger port 1	Short press to play previous (long press to decrease volume)
10	GND	Ground	Power GND
11	IO2	Trigger port 2	Short press to play next (long press to increase volume)
12	ADKEY1	AD Port 1	Trigger play first segment
13	ADKEY2	AD Port 2	Trigger play fifth segment
14	USB+	USB+ DP	USB Port
15	USB-	USB- DM	USB Port



Elenco Componenti:

R1÷R6: Trimmer 10 kohm MO
 R7÷R18: 1 kohm
 R19: -
 R20, R22: 470 ohm
 R21: 4,7 kohm
 R23, R24: 1 kohm
 C1, C3: 100 nF ceramico
 C2, C4: 220 µF 25 VL
 elettrolitico
 C5: 15 pF ceramico
 C6: 15 pF ceramico
 C7÷C9: 100 nF ceramico
 D1: 1N4007
 U1: ATMEGA328P (MF1230)
 U2: 7805
 U3: Modulo DFPlayer Mini
 (DFR0299)
 U4: Convertitore USB-TTL

GND. Per gestire eventualmente strip molto lunghe e quindi con un assorbimento più elevato, sul connettore siglato NEOPIXEL abbiamo portato anche i 12 volt, prelevati direttamente a monte del diodo D1. A questa uscita è possibile ad esempio collegare un regolatore switching in grado di fornire la corrente necessaria all'intera fila di led NEOPIXEL. Passiamo adesso all'audio del presepe, che viene ottenuto grazie a un valido modulo siglato DFR0299 (lo trovate su www.futurashop.it/mini-riproduttore-mp3-montato-9145-dfr0299) controllabile tramite interfaccia seriale a livello TTL; nel nostro circuito utilizza la seriale hardware del microcontrollore, la stessa impiegata in fase di programmazione per caricare lo sketch sul microcontrollore, ma ciò non costituisce un problema in quanto le linee RX e TX dell'ATmega vengono usate per una funzione alla volta, ovvero una volta che è stato programmato il micro, lo

sketch le utilizza per gestire il modulo. Il DFR0299 nello schema lo trovate siglato U3; è in grado di riprodurre direttamente i file MP3 e WAV memorizzati su una SD-Card della capacità massima di 32 GB purché formattata con FAT16 o FAT32, inserita nell'apposito slot. La particolarità di questo modulo è che è stato progettato e realizzato per il mondo Arduino: anche nella scelta del modulo, vedete, è evidente il vantaggio di avere a bordo della nostra scheda un'architettura Arduino. Per la gestione del modulo è chiaramente stata sviluppata una libreria specifica (scaricabile anche dal nostro sito www.elettronica.in insieme agli altri file del progetto) che il nostro sketch include, la quale permette di decidere quale file riprodurre, regolare il volume ecc. Il modulo contiene un decoder in grado di decomprimere l'audio in formato MP3 e un controller in grado di accedere via SPI

ai dati contenuti nella SD-Card: man mano che lo stream di dati viene letto, il decoder lo trasforma in audio non compresso, che poi viene amplificato da un piccolo finale integrato, mono con uscita a ponte, da 3 watt, che sono una potenza più che sufficiente a pilotare un altoparlante in grado di farsi sentire nell'ambiente. Se serve maggior potenza, utilizzate l'uscita audio (+ e - SPK) per pilotare un amplificatore di potenza cui applicare una cassa acustica; in alternativa potete prendere l'audio dai contatti 4 e 5, che sono le uscite audio a basso livello dei canali, rispettivamente, LEFT e RIGHT. In questo caso potete contare su un audio stereo. Tutte le funzioni del modulo U3 si governano da seriale con specifiche istruzioni che il microcontrollore impartisce; quelle riguardanti il volume vengono generate leggendo l'ADC dell'ATmega quando campiona la tensione sul cursore del trim-

(FTDI5V)

LD1÷LD6: LED rosso 5 mm

LD7: LED verde 5 mm

LD8: LED giallo 5 mm

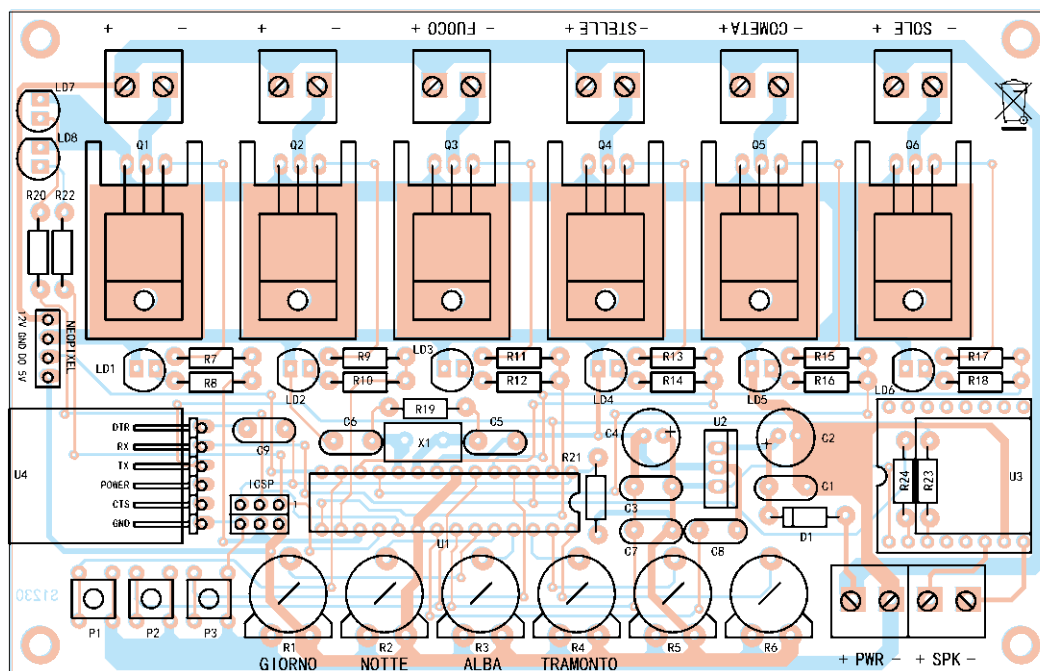
X1: Quarzo 16 MHz

Q1÷Q6: STP36NF06

P1÷P3: Microswitch

Varie:

- Morsetto 2 poli passo 5,08mm (8 pz.)
- Strip femmina 8 vie (2 pz.)
- Strip maschio 3 vie (2 pz.)
- Strip maschio 4 vie
- Strip maschio 6 vie 90°
- Zoccolo 14+14
- Dissipatore (6 pz.)
- Vite 10 mm 3 MA (6 pz.)
- Dado 3 MA (6 pz.)
- Circuito stampato S1230



mer R6. Abbiamo anche previsto di utilizzare, tra i comandi di gestione della riproduzione (Play, Stop, Pause, FWD ecc.) solo quello di Play/Stop, che viene implementato con i tre pulsanti collegati al microcontrollore tramite le linee PD2, PD4 e PD7 (tutte inizializzate come input dotati di pull-up interno) che hanno tutti la stessa funzione, ovvero quella di abilitare/disabilitare l'audio (**Listato 2**). Più esattamente, la pressione di uno dei pulsanti P1, P2, P3 quando è in riproduzione un brano causa l'arresto del medesimo e la successiva pressione fa ricominciare la riproduzione dall'inizio del brano arrestato in precedenza. Quando l'audio è in riproduzione, il LED LD7 viene acceso dalla linea PB5 del microcontrollore, la quale nel normale funzionamento è utilizzata come output; le linee PD0 e PD1, facenti capo all'UART, vengono accoppiate al modulo U3 mediante resistenze per evitare che in program-

mazione, ovvero quando alla scheda si applica il convertitore Seriale/USB, vi siano interferenze tra le tensioni dei due moduli. I file dell'audio che volete sia riprodotto devono essere scritti sulla SD-Card con nome nel formato 001.mp3, 002.mp3, 003.mp3 e 004.mp3, perché il primo file viene richiamato durante la fase GIORNO, il secondo durante il TRAMONTO, il terzo quando viene acceso il FUOCO (fuoco tremolante) e il file 004.mp3 viene richiamato quando arriva l'ALBA.

Durante il passaggio tra una fase e l'altra, il volume di riproduzione viene abbassato gradualmente fino al termine del brano e viene avviata, con volume in graduale aumento, la riproduzione del brano successivo (tipo ciò che avviene alla radio o negli album musicali); questo vale nel passaggio dal brano da una fase alla seguente.

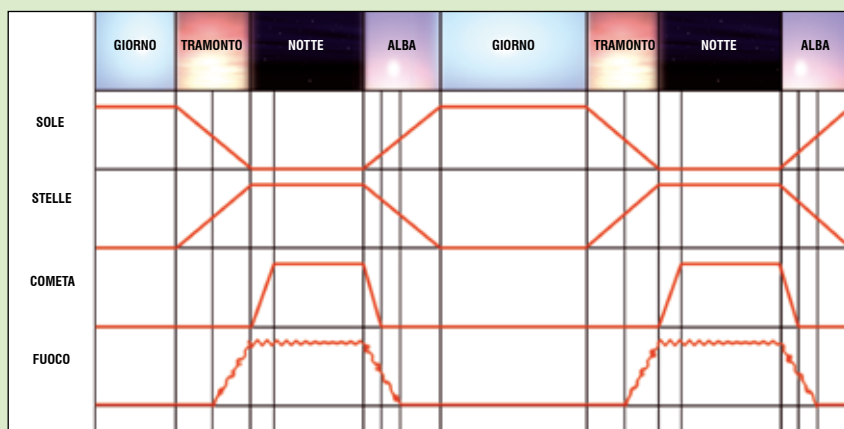
Detto ciò completiamo la descrizione del circuito con le parti

finora tralasciate, ossia l'alimentazione, l'oscillatore dell'ATmega 328P e la programmazione in-circuit; per quanto riguarda la sezione di alimentazione, il circuito va alimentato con 12÷14 Vcc (la corrente richiesta dipende da quanti utilizzatori collegate alle uscite) tramite la morsettieria +/- PWR, la cui tensione giunge direttamente alle uscite a MOSFET e al connettore per NeoPixel. La tensione di alimentazione principale giunge anche all'anodo del diodo D1, inserito per proteggere il resto del circuito dall'inversione di polarità, e viene applicata all'ingresso del regolatore di tensione U2, compito del quale è ricavare i 5 volt stabilizzati con cui funzionano il microcontrollore, i componenti di contorno e il modulo audio U3. L'eventuale converter Seriale/USB si alimenterà con la tensione dell'USB del computer cui verrà interfacciato. I condensatori posti sull'alimentazione d'ingresso servono a filtrare gli

Funzionamento delle uscite

La sequenza di accensione delle luci controllate dal nostro circuito. Di giorno l'unica lampada (o serie di lampade) accesa è quella che simula il sole e che illumina, appunto, "a giorno" il presepe. Questa lampada resta accesa per un tempo compreso tra 0 e 102 secondi a seconda di come viene settato il trimmer R1. Scaduto il tempo ha inizio una nuova fase: il tramonto, la cui durata dipende dal trimmer R4. Agendo su detto trimmer è possibile impostare un valore compreso tra 0 e 102 secondi circa. A poco a poco la luce del giorno si attenua mentre aumenta la luminosità delle lampade che simulano le stelle, sino al completo spegnimento del sole che coincide con la massima illuminazione delle stelle. Esattamente a metà di questa fase di transizione iniziano ad accendersi i fuochi delle case che raggiungono la massima luminosità al termine del ciclo. Da notare che questa uscita genera una luce tremolante che simula il bagliore del fuoco. Infine, al termine del ciclo, si illumina gradatamente anche la luce che simula la stella cometa. Il passaggio dalla minima alla massima luminosità avviene in tempo pari ad 1/4 di quello impostato per il TRAMONTO.

A questo punto ci troviamo in piena notte con le stelle in cielo che brillano, la cometa completamente illuminata ed i fuochi nelle case accesi col loro tipico tremolio. La durata di questa fase (compresa, come per il giorno, tra 0 e 102 secondi circa), che è la NOTTE, viene impostata mediante il trimmer R2. Allo scadere del tempo impostato, ha inizio la quarta ed ultima fase: l'alba. Gradatamente l'intensità luminosa delle stelle si abbassa mentre aumenta la luminosità del giorno sino al completo spegnimento delle stelle ed alla completa accensione delle lampade che simulano la luce del giorno. All'inizio di questa fase anche la cometa si spegne gradatamente ma molto più velocemente tanto che, trascorso un periodo pari ad 1/4 del tempo impostato per l'alba, la cometa risulta completamente spenta. Sempre all'inizio di questa fase di transizione, anche la luminosità dei fuochi inizia a calare sino allo spegnimento. In questo caso il passaggio dalla massima luminosità allo spegnimento completo avviene in un tempo pari ad 1/2 di quello impostato per l'ALBA, mediante il trimmer R3. A questo punto abbiamo simulato un ciclo di 24 ore ed il sistema si appresta a ripetere all'infinito la sequenza programmata.



impulsi creatisi sulle piste per effetto dell'assorbimento dei LED e in generale dei carichi collegati alle uscite a MOSFET; ciò è necessario in quanto la pulsazione dell'alimentazione dei diodi è a frequenza elevata e altrimenti i disturbi (che poi sono abbassamenti di tensione pur lievi concomitanti con l'accensione dei singoli LED) potrebbero interferire con il corretto funzionamento del micro-

controllore. Analoga è la funzione dei condensatori C3 e C4. Veniamo ora all'interfaccia per la programmazione in-circuit, da utilizzare solo se volete caricare il bootloader in un microcontrollore vergine; a riguardo va detto che il microcontrollore fornito dalla Futura Elettronica (www.futurashop.it) con il kit è già programmato con bootloader e sketch, ma chi preferisce procurarsi da sé l'ATmega 328P

dovrà prima caricare il bootloader per poter trasferire lo sketch dall'IDE di Arduino.

Il connettore preposto è quello siglato ICSP e ad esso fanno capo le linee di reset (RST), MISO (ingresso dati del programmatore e uscita del micro U1), MOSI (uscita dati del programmatore e ingresso dati del micro), SCK (clock della comunicazione seriale sull'SPI formato da MISO e MOSI), oltre che la massa e la Vcc, che durante la programmazione viene fornita dal programmatore per alimentare il circuito.

Ricordiamo che quando si programma il bootloader il circuito deve essere alimentato attraverso la morsettiera +/- PWR. Concludiamo con l'oscillatore del microcontrollore, che è impostato come clock ricavato dal quarzo X1, in parallelo al quale troviamo la resistenza R19; i condensatori C5 e C6 completano l'oscillatore, che funziona a 16 MHz, frequenza più che sufficiente a gestire l'intero ciclo di lavoro della centralina.

REALIZZAZIONE PRATICA

Passiamo adesso alla costruzione della scheda, che richiede la preparazione (o l'acquisto, visto che il materiale è tutto disponibile presso Futura Elettronica) di un circuito stampato a doppia faccia le cui tracce sono scaricabili dal nostro sito web insieme agli altri file del progetto; incisa e forata la basetta, realizzate le vie (ossia le interconnessioni tra piazzole comuni alle due facce) si possono montare i componenti iniziando da quelli a più basso profilo (resistenze e diodo D1) e proseguendo con lo zoccolo del microcontrollore, il pin-strip a 6 poli con terminali a 90° per il convertitore Seriale/USB (facol-

Guarda Presepio in funzione su YouTube!



Vuoi vedere il nostro presepe 3D realizzato con la stampante 3Drag e illuminato

da Presepio? Nulla di più semplice: collegati alla pagina <https://youtu.be/X0lvpg3FLGg>. Dal nostro canale YouTube <https://www.youtube.com/user/ElettronicaIN/videos> potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.



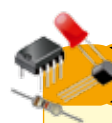
tativo), i trimmer, i pulsanti da c.s. e i LED, da orientare come indicato nel piano di montaggio, rammentando che il catodo è l'elettrodo situato dal lato smussato del contenitore. Ora inserite e saldate i due pin-strip femmina a 8 poli ciascuno per montare il modulo audio, il pin-strip per l'ICSP (4 poli), le otto morsettiere e il regolatore 7805, che va tenuto in piedi con il lato delle scritte rivolto allo zoccolo del microcontrollore. Ora bisogna montare i sei MOSFET, ognuno dei quali va inserito nei rispettivi fori dello stampato dopo averne piegato ad angolo retto i terminali ed averlo appoggiato (con il lato scritte verso l'alto) a un dissipatore di calore ad "U" tipo ML26 (22° C/W di resistenza termica). Completate le saldature e verificato che ogni componente sia montato correttamente e nel verso previsto, inserite il modulo DFR0299 nei propri pin-strip

(orientandolo in modo che il porta-SD sia rivolto all'esterno del circuito) e il microcontrollore, la cui tacca di riferimento deve essere rivolta al regolatore integrato U2. Nel DFR0299 bisogna inserire una microSD della capacità desiderata, con già caricati (fatelo su un computer) i file audio desiderati; per essi vale quanto detto qualche paragrafo indietro, rammentando che per ovvi motivi sarebbe meglio che la durata di ciascuno non ecceda quella impostata per la fase che deve accompagnare, quindi, ad esempio, il motivo suonato per il giorno deve durare non più di quanto impostato per il giorno. Se il file dura meno in ogni caso non è un problema, in quanto il brano viene riprodotto ciclicamente fino alla fine del tempo impostato per la relativa fase e quindi sfumato per tempo.

Ricordate che i brani vanno preferibilmente inseriti nella root della microSD, ma il modulo è comunque in grado di cercare nelle cartelle, fermo restando che l'ordine di riproduzione, se nella scheda esistono altri file audio, non sarebbe più garantito.

A questo punto la centralina può o meno richiedere la programmazione in base a come l'avete realizzata: se avete acquistato un microcontrollore vergine dovete, con il programmatore ATMEL o una scheda Arduino impostata come programmatore (tecnica di programmazione stand-alone), caricare il bootloader utilizzando, allo scopo, la connessione ICSP. Fatto ciò, rimuovendo il programmatore e montando il modulo d'interfaccia Seriale/USB nel connettore U4 (si tratta dell'FTDI5V, un modulo commercializzato dalla Futura

Elettronica, www.futurashop.it, basato sull'integrato FT232RL della FTDI) si collega la presa USB di quest'ultimo a quella di un PC nel quale sia installato l'IDE di Arduino e si carica lo sketch (l'IDE vede la nostra scheda come una Arduino UNO, perché il bootloader caricato nell'ATmega 328P è quello). Tutto ciò non è richiesto se si acquista il kit di montaggio, dove il microcontrollore viene fornito programmato con le funzionalità base (4 uscite gestibili da 4 trimmer). Comunque per rendere il progetto facilmente adattabile alle vostre esigenze diamo lo sketch, che potete scaricare dal nostro sito, insieme ai file del progetto. Anche se avete acquistato il kit ma non vi accontentate della configurazione base e volete provare ad apportare qualche modifica, dovete procurarvi e montare il convertitore Seriale/USB. Bene, a questo punto non ci resta che augurarvi buon divertimento e soprattutto Buone Feste! ■



per il MATERIALE

Tutto il materiale di questo progetto può essere acquistato presso Futura Elettronica. La scheda presepe con Arduino (cod. FT1230K) viene venduta già programmata a Euro 39,00; il convertitore USB/TTL (cod. FTDI5V) è disponibile a Euro 19,00, mentre il mini riproduttore MP3 (cod. DFR0299) costa Euro 12,00. 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>

PRECISION LABS

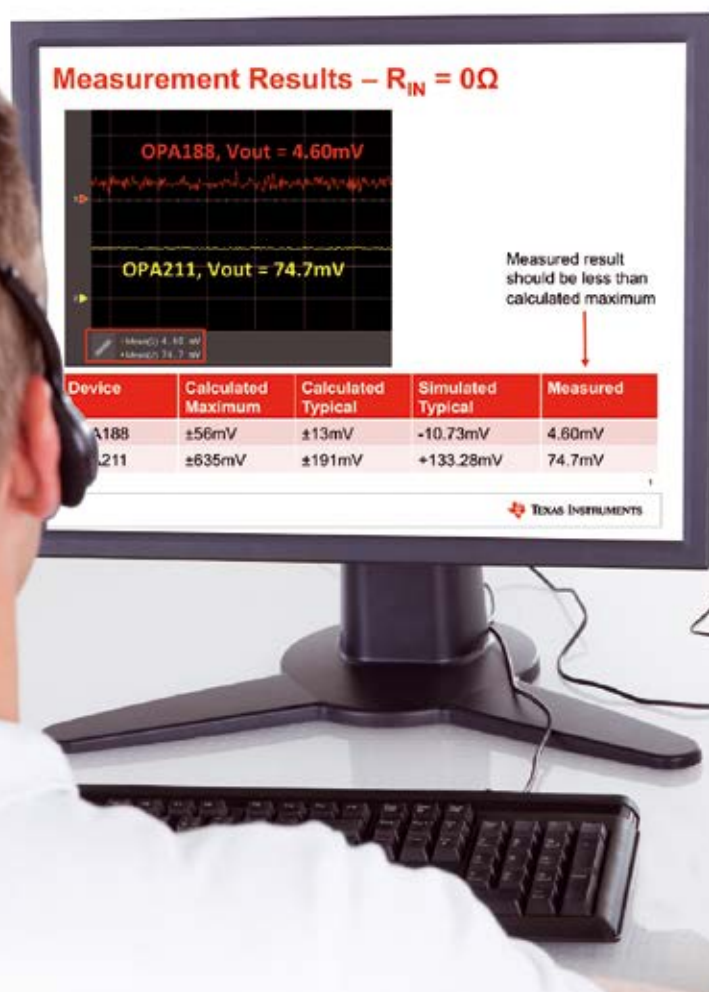
I CORSI ON-LINE DI ELETTRONICA ANALOGICA DI TEXAS INSTRUMENTS

di ARSENIO SPADONI

Texas Instruments supporta i progettisti di elettronica analogica con una importante iniziativa on-line: ne parliamo con Art Key, a capo di questo progetto.

Avviiata nei primi mesi dell'anno, questa importante iniziativa di Texas Instruments si è rapidamente affermata nella community di ingegneri e progettisti che operano nel campo dell'elettronica analogica, un settore dell'elettronica che negli ultimi anni è stato "offuscato" dall'impetuoso avanzare del digitale. Ormai tutti

hanno qualche competenza di elettronica digitale e di programmazione, mentre sono sempre di meno quelli che hanno competenze in ambito analogico. Forse proprio questa è stata la ragione che ha spinto TI a dare vita a questo progetto. Non bisogna infatti dimenticare che tutti i dati e i segnali, prima di assumere una forma digitale, hanno una



natura analogica e debbono essere trattati, amplificati ed elaborati con tecniche analogiche, prima di essere trasformati in segnali digitali.

L'iniziativa, che prende il nome di Precision Labs, è dedicata agli ingegneri che operano in ambito analogico ed agli studenti di ingegneria elettronica; tuttavia, in considerazione del fatto che i corsi partono da concetti di base, anche gli hobbysti, i semplici studenti ed i makers possono trarre profitto da questa iniziativa. Le competenze tecniche acquisite potranno essere sfruttate per migliorare e accelerare il lavoro di progettazione.

Le risorse messe a disposizione sono completamente gratuite ed attualmente comprendono più di trenta hands-on con i relativi video.

Texas Instruments mette anche a disposizione una serie di strumenti per poter realizzare in pratica gli esperimenti descritti nei moduli didattici. In particolare, il *TI Precision Labs – Op Amp Evaluation Module* (TI-PLABS-AMP-EVM), del costo di 199 dollari, incorpora tutti gli strumenti per dare vita agli esperimenti; a supporto di questa iniziativa sono anche disponibili altre risorse gratuite quali il *TI Design reference designs* e il *TINA-TI SPICE model simulator*.

Per alimentare i circuiti ed effettuare le misure, è stato previsto l'impiego dello strumento Virtual Bench di National Instruments e del software *Bode Analyzer Software*, appositamente realizzato da TI per *Virtual Bench*.

Questa combinazione, come ha osservato **Chad Chesney**, director of data acquisition marketing di National Instruments "Fornisce un modo più coinvolgente e accessibile per imparare e applicare concetti analogici. L'impiego di un'unica,

intuitiva, interfaccia utente, garantisce una migliore comprensione delle prestazioni complessive".

Naturalmente gli utenti potranno utilizzare qualsiasi altra strumentazione di laboratorio in loro possesso, purché abbia le caratteristiche necessarie.

Questi, in sintesi, i vantaggi dell'iniziativa Precision Labs:

- **Apprendere dall'esperienza:** consente di vedere le nozioni teoriche applicate al mondo reale con esperimenti pratici e video.
- **Ambiente di apprendimento personalizzato:** consente di scegliere tracce su argomenti specifici quali rumore e larghezza di banda, o su temi legati agli interessi o alle esigenze degli utenti.
- **Accelera le competenze dei neo-laureati:** facilita la transizione dalle nozioni teoriche apprese durante il corso di studi alle esigenze di progettazione del mondo reale.
- **Utile materiale didattico:** include un manuale di laboratorio e una cartella di lavoro scaricabili gratuitamente, così come l'*Analog Engineer's Pocket*

Reference di TI, che contiene le principali formule utilizzate in ambito elettronico.

- **Supporto di esperti:** il forum *TI Precision Labs support forum* è disponibile all'interno della TI E2E Community per rispondere a qualsiasi domanda relativa a questa iniziativa.

Per accedere alle risorse di Precision Labs è necessario registrarsi e creare un account (assolutamente gratuito) myTI.

Precision Labs è inserito nell'area didattica TI Training Center del sito di Texas Instruments dove sono disponibili numerose altre risorse che TI mette a disposizione di ingegneri e tecnici alla ricerca di soluzioni per lo sviluppo dei propri progetti.

Per conoscere meglio alcuni aspetti di questa iniziativa di TI, abbiamo incontrato **Art Kay**, Linear Applications Manager, High Performance Linear di Texas Instruments.

D: Che cos'è esattamente Precision Labs?

R: Precision Labs è un'iniziativa di Texas Instruments che, a pochi



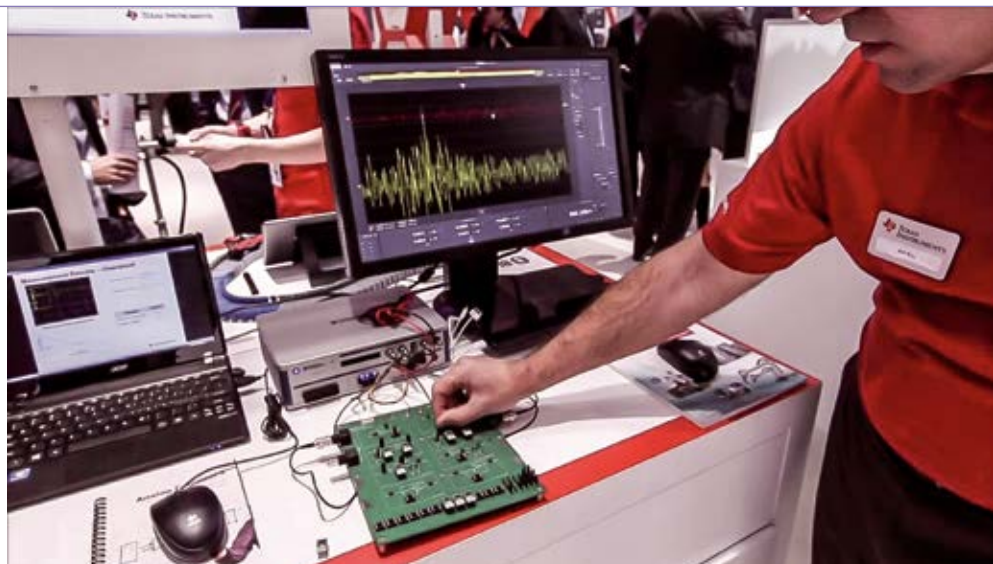
mesi dal suo lancio, è già molto famosa ed apprezzata all'interno della community degli ingegneri. Sappiamo che molti ingegneri che operano in ambito analogico, sia come progettisti che come sistemisti, utilizzano Precision Labs – Op Amps per la formazione interna. Il nostro team che lavora a questa iniziativa sta sviluppando nuovi materiali ed entro la fine dell'anno verranno proposti nuovi video e nuovi corsi. TI attribuisce grande valore a questo genere di contenuti, e per il futuro abbiamo in programma di arricchire la gamma degli argomenti trattati.

D: Quale hardware bisogna possedere per realizzare i vari esperimenti? Ed è possibile utilizzare oscilloscopi e board differenti?

R: Gli esperimenti utilizzano componenti standard che possono essere montati su normali breadboard ed anche per il test e le misure è possibile fare ricorso ad una dotazione standard. Il modulo di valutazione Precision Labs – Op Amp Evaluation Module (TI-PLABS-AMP-EVM) – rappresenta un sistema semplice e pratico per realizzare gli esperimenti senza dover effettuare manualmente i collegamenti tra i vari componenti. Maggiori informazioni su questi elementi sono disponibili sul nostro sito. Nel caso l'utente non disponga della strumentazione necessaria, un'ottima soluzione da prendere in considerazione è il National Instruments Virtual Bench. Si tratta di uno strumento compatto che racchiude al suo interno tutto quanto serve per realizzare questo genere di esperimenti.

D: Quale feedback avete ricevuto finora dagli utenti?

R: I feedback sulla nostra inizia-



tiva TI Precision Labs sono tutti ampiamente positivi. Diversi ingegneri ci hanno detto che sarebbero stati contenti se avessero avuto a disposizione uno strumento di formazione come questo anni fa, e ci hanno chiesto ulteriori video e contenuti. Abbiamo anche ricevuto richieste per introdurre corsi TI Precision Labs tra gli insegnamenti universitari, il che rappresenta un'importante conferma della qualità dei nostri contenuti.

D: A quali nuovi esperimenti state lavorando? E gli utenti possono richiedervi specifici argomenti?

R: Stiamo già sviluppando e preparando nuovi esperimenti e video; tra questi alcuni moduli di formazione sulla dissipazione negli amplificatori di potenza e sull'auto-riscaldamento. Vogliamo che TI Precision Labs diventi la risorsa di riferimento per l'aggiornamento nell'ambito degli amplificatori e del segnale analogico. A questo proposito abbiamo anche dato vita al forum TI Precision Labs – Op Amps Forum sulla nostra community per ingegneri E2E, dove è possibile porre domande e fornire feedback sull'offerta formativa. Insomma... diteci la vostra!

D: Ritiene che questa iniziativa sia adatta anche ai makers?

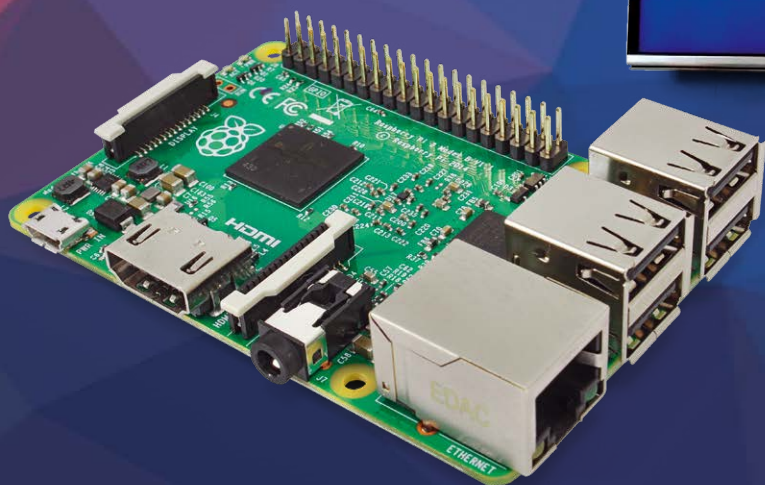
R: Sicuramente. Iniziamo infatti con i concetti di base per poi continuare fino agli argomenti più avanzati. Questo rende semplice la fruizione anche da parte dei makers, che possono partire dal livello più adatto alle proprie conoscenze. Alcune persone potrebbero voler una panoramica generale, per poi poter scegliere un argomento specifico da approfondire, altri potrebbero dare una rapida occhiata agli argomenti di base per poi concentrarsi sugli aspetti più complessi.

D: Texas Instruments ha in programma iniziative simili su altre tecnologie?

R: Certo! TI ha appena lanciato un nuovo Training Portal che contiene video tecnici che spaziano tra diversi argomenti di progettazione analogica e digitale, come ad esempio *understanding current shunt monitors* e *how to avoid common-error amplifier pitfalls*, con l'aggiunta di un nuovo video di formazione quasi ogni giorno. Questo il link per accedere a Precision Labs: <http://www.ti.com/lstds/ti/amplifiers-linear/precision-amplifier-precision-labs.page>

INTRODUZIONE A OSMC

Trasformiamo, grazie a un sistema operativo Linux-like, Raspberry Pi in un prestante ed economico Media Center.



..... di SAMUELE BRUNINI e BORIS LANDONI

Qualche anno fa un team di sviluppatori realizzò un programma chiamato Kodi Entertainment Center, per trasformare in Home Theatre o Media Center la famosa console videogame Xbox di Microsoft, sfruttando le ottime qualità multimediali e la potenza di calcolo di quest'ultima. Kodi Entertainment Center nacque come alternativa all'Xbox Media Player (XBMP), il programma originale Microsoft per la riproduzione di contenuti multimediali, che equipaggiava l'Xbox, rispetto al quale aveva un'interfaccia grafica ed un core rinnovati; perciò al grosso del pubblico divenne noto come Xbox Media Center (XBMC). Una prerogativa di tale software era che, sebbene fosse nato per la Xbox, si poteva installare su PC con sistema operativo Windows così da realizzare un completo Media Center o un Home

Theatre; ma siccome Kodi Entertainment Center nasceva per la Xbox, si creò attrito con Microsoft, la quale rivendicò l'uso esclusivo del nome Xbox e l'installazione del programma solo su Xbox. Gli sviluppatori rinunciarono al nome XBMC e il programma venne chiamato semplicemente Kodi, per rompere ogni ponte col passato. Liberatosi dai vincoli "legali" Kodi poté essere sviluppato e reso disponibile anche per Mac OS, Linux, Apple TV, Raspberry Pi e Android.

DA APPLICAZIONE A SISTEMA OPERATIVO

Kodi è stato, nel tempo, sviluppato anche come sistema operativo multimediale autonomo, basato su Linux (la distribuzione è Lubuntu) e perciò chiamato XBMCbuntu. Su questa scia è stato creato OSMC (Open Source Media Center), del quale parliamo in questo articolo, che è un media player (riproduttore di file audiovisivi) basato su Linux, ispirato dal progetto Kodi.

QUALCOSA SU KODI

Prima di passare alla descrizione di OSMC e alla sua installazione, spendiamo qualche paragrafo

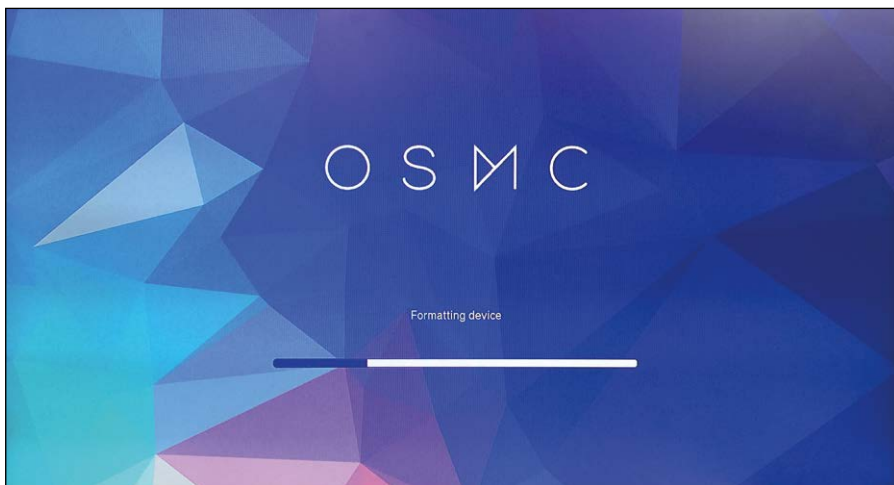


Fig. 1

medesime caratteristiche di Kodi ed anche di XBMC.

RASPBERRY PI DIVENTA UN MEDIA CENTER

In questo articolo vi proponiamo di installare OSMC su Raspberry Pi per realizzare un sistema multimediale che rispetto, ad esempio, a una smart TV:

- permette di riprodurre la maggior parte dei formati video, immagine e musica, grazie a codec facilmente installabili;
- offre la possibilità di installare sorgenti da cui scaricare programmi (sono circa 30.000 e continuano ad aumentare);
- permette l'integrazione con i social network;
- consente la condivisione di file all'interno della rete attraverso un server HTTP;
- essendo basato su Linux è possibile fargli svolgere compiti o eseguire applicazioni normalmente utilizzate su Linux (la gestione è possibile anche da linea di comando);
- è completamente gestibile in remoto via SSH;
- contiene funzioni che hanno solo alcuni dispositivi come l'Airplay di Apple, quindi ha una buona compatibilità con dispositivi mobili come smartphone e tablet;
- è totalmente personalizzabile secondo le proprie esigenze;
- supporta la maggior parte degli adattatori Bluetooth USB, il che permette di scambiare file con OSMC o riprodurre su di esso musica dal nostro cellulare senza ricorrere a cavi dedicati.

Inoltre Raspberry Pi dispone dei GPIO, molto utili se vogliamo realizzare qualche progetto con LED IR o implementare modi per interagire con altre unità, o progetti come Ambilight, piena-

sulle funzionalità multimediali di Kodi, perché sono le stesse. Kodi può essere utilizzato per riprodurre la maggior parte dei formati multimediali. Riportiamo solo alcuni tra i più famosi formati:

- AVI, MPEG, WMV, QuickTime, MP4, AAC;
- video MPEG-1/2/4, DivX, XviD, H.264, RealMedia e QuickTime, MJPEG, RealVideo;
- immagini RAW, JPEG, GIF, BMP, PNG, TIFF, ;
- audio: MP3, AAC, MIDI, WAV/WAVE, MP2, Vorbis, RealAudio, Compact Disc Digital Audio(CDDA);
- sottotitoli: AQTtitle, CC, JACOSub, MicroDVD, MPsub, OGM, VOSub, VPlayer.

Kodi riproduce tutti i file multimediali direttamente dalle unità di memoria di massa, dalla root dell'hard-disk, oppure da una cartella o una partizione di Windows condivisa attraverso il sistema Samba (SMB) o, ancora, da iTunes attraverso il protocollo di comunicazione DAAP. Inoltre utilizza la porta ethernet e l'eventuale accesso ad Internet; se rileva il collegamento a Internet, è possibile richiedere al database IMDb copertine e recensioni di film, e al database CDDb i titoli delle tracce di CD audio, la riproduzione di video in streaming e l'ascolto di qualsiasi radio

on-line. Ancora, Kodi può gestire elenchi di tracce, ha funzioni di galleria d'immagini, previsioni del tempo e dispone di numerosi effetti grafici coordinati con i file audio.

Kodi supporta come plug-in gli script Python, permettendo all'utente di aggiungere funzionalità; gli script più utilizzati sono guide TV, previsioni del tempo, orari ferroviari, script per controllare software e hardware PVR (MediaPortal, MythRV, TiVo, Reeplay), browser di stazioni di radio on-line (ad esempio SHOUTcast), browser di Internet TV, client di posta elettronica, programmi di file sharing P2P (BitTorrent, IRC) e altri.

E ORA ...OSMC

La differenza sostanziale tra OSMC e Kodi è che OSMC non è un'applicazione che gira su un sistema operativo, come ad esempio Windows o Linux, ma un sistema operativo con vocazione multimediale. Infatti esso è in grado di gestire l'hardware (telecomandi, WiFi, Bluetooth, tastiere...) e le risorse di sistema, oltre che di tenersi aggiornato. OSMC si installa dunque come sistema operativo: sull'unità di memoria di massa usata per il boot.

OSMC è basato sulla distribuzione di Linux Debian Jessie e mantiene gli stessi plugin e le

mente supportati da OSMC. Abbiamo scaricato e testato OSMC su Raspberry Pi 2 e dobbiamo dire che c'è stato un grosso miglioramento rispetto alla versione per Raspberry Pi, in quanto le caratteristiche della nuova Raspberry Pi 2 permettono di contare su un hardware più solido, soprattutto quando si vogliono utilizzare servizi di streaming da Internet, che richiedono molte risorse.

Il sistema operativo è molto fluido e non presenta particolari problemi; oltretutto questo progetto open source è valido e la piattaforma viene mantenuta sempre aggiornata, fornendo quel supporto che spesso per l'utente determina la scelta di un sistema operativo.

Se volete provare OSMC, di seguito vi spieghiamo come fare.

SETUP-INSTALLAZIONE

Vediamo come si installa OSMC, partendo dal presupposto di disporre di una scheda Raspberry Pi 2 dotata di monitor HDMI, mouse e tastiera, e di aver inserito la SD-Card nell'apposito lettore; ricordiamo che nella SD dovete aver installato il file scaricato dal sito ufficiale alla pagina <https://osmc.tv/download/images/>, la quale riporta le immagini dei

Sta per arrivare il Wi-Fi superveloce

Il 'Tera-Wifi', il sistema di comunicazione wireless ultraveloce potrebbe presto divenire realtà, soprattutto dopo la pubblicazione della ricerca su Nature Photonics effettuata dal team guidato da Daniel Mittleman, dell'università americana Brown. I ricercatori hanno infatti creato i primi dispositivi capaci di smistare i dati trasmessi con tale tecnologia al fine di renderla in un futuro molto vicino commerciabile. L'innovativa tecnologia Wi Fi si basa sull'uso di onde a frequenze più alte delle attuali, i Terahertz, che consentirebbero il trasporto di dati almeno 100 volte superiore



rispetto al WiFi classico. Al momento però mancano dispositivi capaci di diffondere tale tecnologia, ma il lavoro dei ricercatori della Brown University potrebbe costituire un tassello fondamentale verso la costruzione di un multiplexer di dati ad alte frequenze, che permetterà ad esempio di trasportare il segnale verso migliaia di telefoni su una singola fibra ottica.

"Siamo alla prima generazione, abbiamo creato un dimostratore e ci sono ancora molte cose da migliorare e studieremo per agire in tal senso", rivela Nicholas Karl, uno degli autori della ricerca.

file. Per la creazione dell'SD card potete utilizzare WinDiskImage. A questo punto inserite la chiavetta WiFi in una qualsiasi porta USB della Raspberry Pi e collegate il cavo HDMI alla Raspberry Pi e al televisore/monitor, quindi alimentare Raspberry Pi. Appena acceso il sistema, ci troviamo davanti la schermata in **Fig. 1**. Attendiamo circa 5 minuti

che l'installazione sia terminata, condizione evidenziata dal fatto che la barra di progresso blu si allunga fino all'estremità a destra. Il sistema si riavvierà e appena acceso dovremo scegliere la lingua (con la tastiera) e dare poi la conferma facendo clic su *yes* (**Fig. 2**).

La pagina successiva ci propone l'accordo di licenza, che accet-



Fig. 2

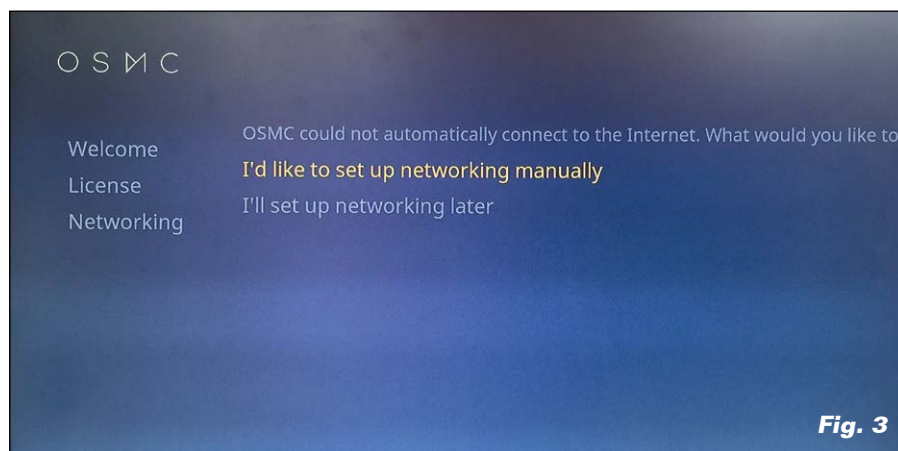


Fig. 3

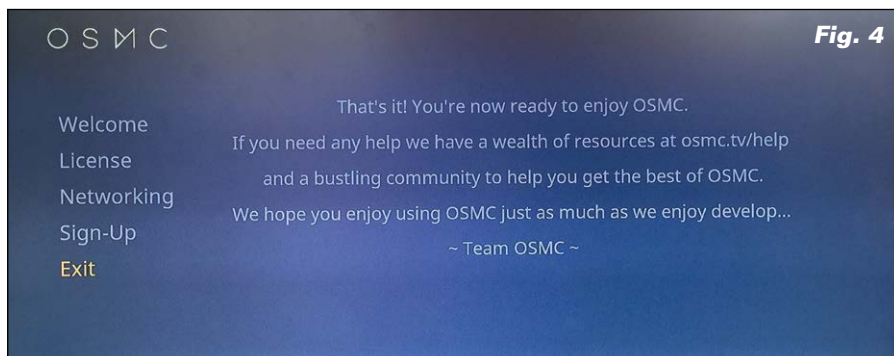


Fig. 4



Fig. 5

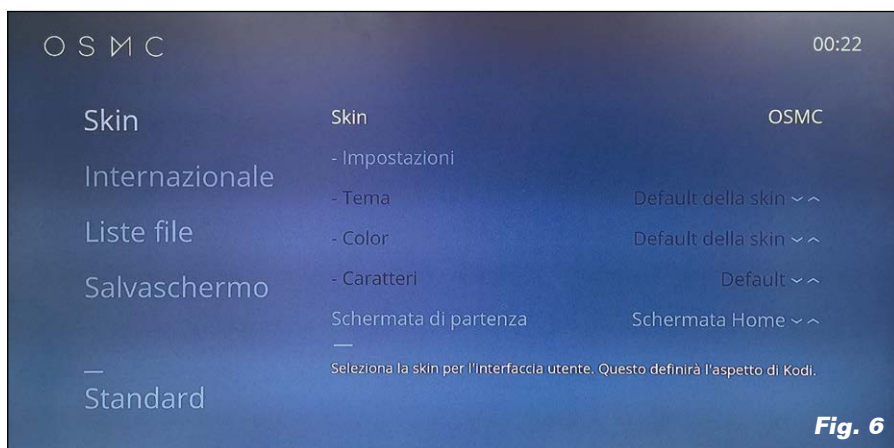
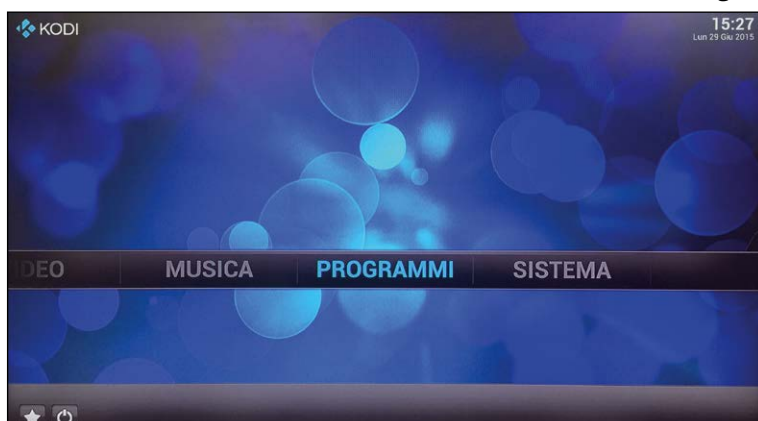


Fig. 6

Fig. 7



Fig. 8



teremo cliccando su *I Agree*; a questo punto appare una nuova schermata come quella visibile in **Fig. 3**, che ci informa dell'impossibilità per il sistema operativo di connettersi a Internet e ci propone due possibilità: qui clicchiamo su *I'd like to set up networking manually* perché configureremo in seguito la connessione ad Internet, poi, nella nuova schermata che apparirà (**Fig. 4**), clicchiamo su *Exit* (ovvero premiamo il tasto Invio) per uscire.

CAMBIO GUI

Per cambiare l'interfaccia utente (GUI) di OSMC nella classica di Kodi, bisogna cliccare su *Impostazioni* > *Aspetto*, allorché appare la schermata di **Fig. 5**. Fatto ciò appare una nuova schermata di configurazione nella quale clicchiamo su *Skin* e poi nuovamente su *Skin* (**Fig. 6**). Ora si aprirà la schermata in **Fig. 7**, dove dobbiamo cliccare su *Confluence*.

WIFI

Vediamo adesso come configurare la connessione WiFi in modo da permettere a OSMC di accedere a Internet per gestire e riprodurre gli stream audio e video in arrivo dalla Rete. Il primo passo da compiere, dalla GUI, è andare nella scheda *Programmi* (**Fig. 8**)

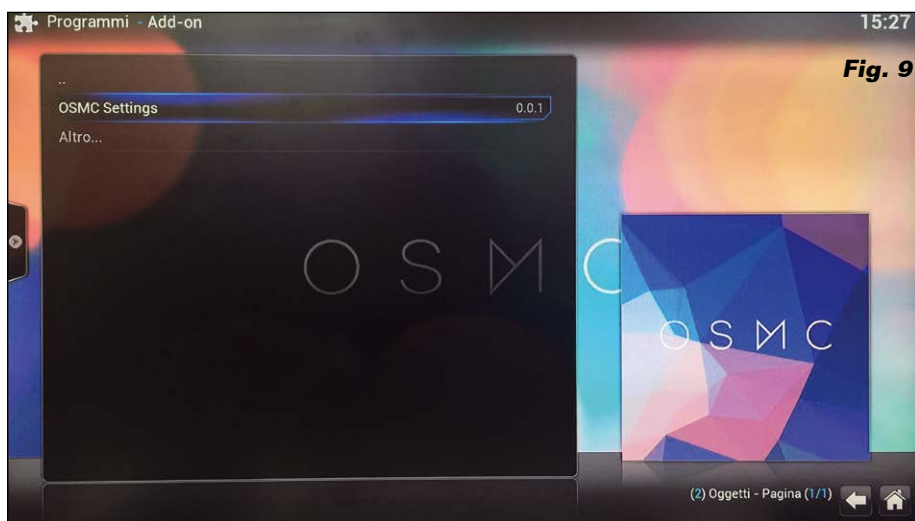


Fig. 9

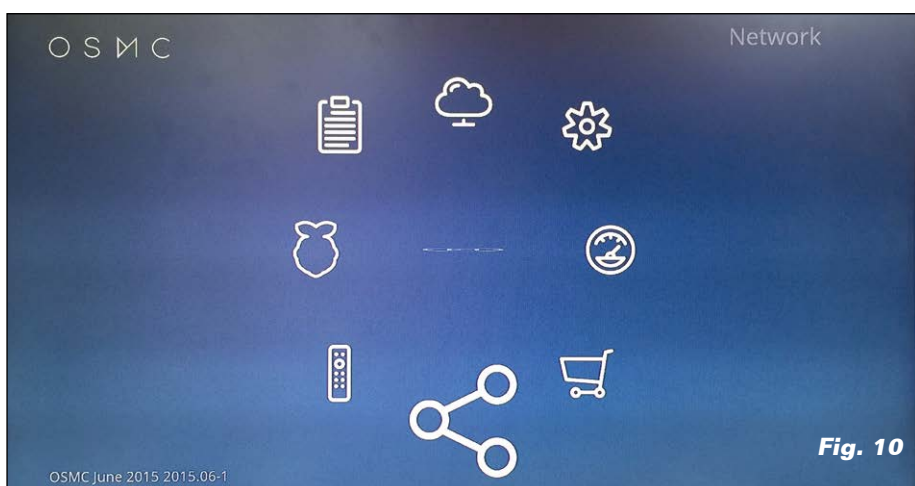


Fig. 10

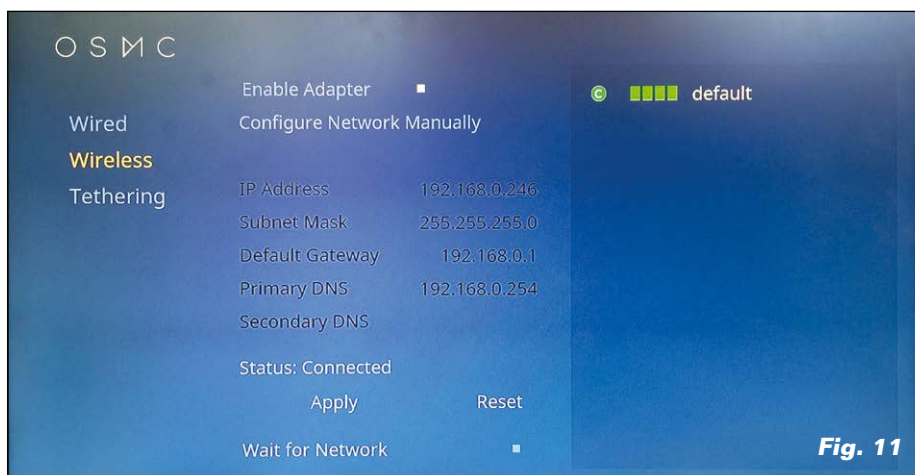


Fig. 11

facendo clic sulla relativa voce di menu nella schermata di avvio. Ci troveremo davanti alla schermata di **Fig. 9**, nella quale dobbiamo cliccare su *OSMC Settings*. Fatto tutto questo, ci appare la schermata di **Fig. 10**, nella quale

ora clicchiamo sull'icona centrale in basso (*Network*), allorché ci appare una schermata (**Fig. 11**) nella quale dobbiamo portarci nella scheda *Wireless* e qui verificare che la voce *Enable Adapter* sia spuntata (se non lo è, bisogna

cliccarvi per attivarla). Ora facciamo clic su *Wireless* per collegarci alla rete desiderata (**Fig. 12**); a sinistra della schermata abbiamo tre voci, corrispondenti ciascuna alla rete cui collegarsi (*Wired* è la LAN accessibile dalla presa RJ45 della Raspberry Pi).

CONNESSIONE VIA ETHERNET

Se ci "accontentiamo" della connessione via cavo tramite la porta ethernet, ci basta connettere il cavo ethernet alla presa LAN della Raspberry Pi: la connessione avverrà automaticamente perché OSMC provvederà a tutto e imposterà i parametri di rete.

COLLEGARSI TRAMITE SMARTPHONE

OSMC consente anche la connessione ad uno smartphone, ad esempio per scaricare e riprodurre brani nella memoria di quest'ultimo come fosse un moderno impianto hi-fi car multimediale. Per funzionare, le relative applicazioni devono essere in esecuzione su uno smartphone collegato alla stessa rete WiFi cui è collegata la Raspberry Pi con installato OSMC.

Il sistema operativo supporta applicazioni (app) per telefoni Android e Apple iOS; va notato che le applicazioni da noi utilizzate non sono le uniche presenti sullo store, però sono quelle che consigliamo perché le abbiamo testate e risultano ben funzionanti. Delle altre non abbiamo avuto esperienza pratica, quindi non possiamo esprimerci.

App Iphone

Per connettersi a Raspberry Pi con OSMC mediante i telefoni di Apple, bisogna cercare in Appstore "Sybu for Kodi and XBMC", quindi scaricare e aprire la relativa applicazione; a questo punto attendiamo che l'iPhone si

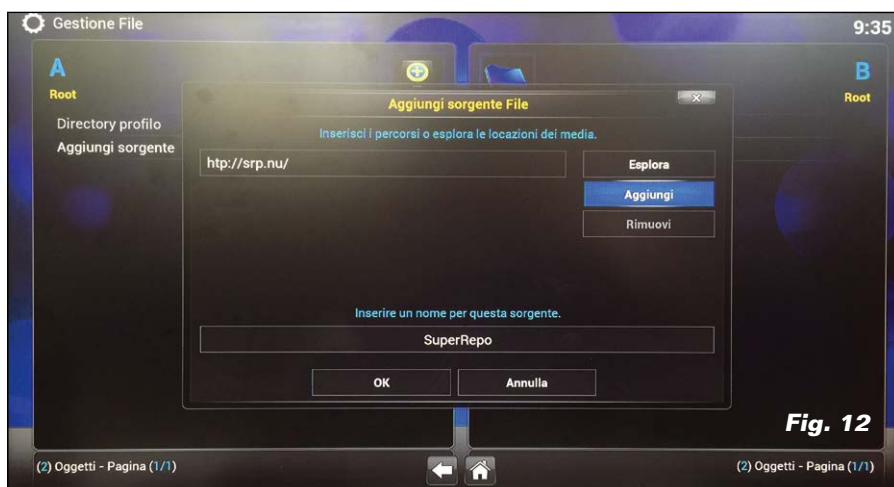


Fig. 12

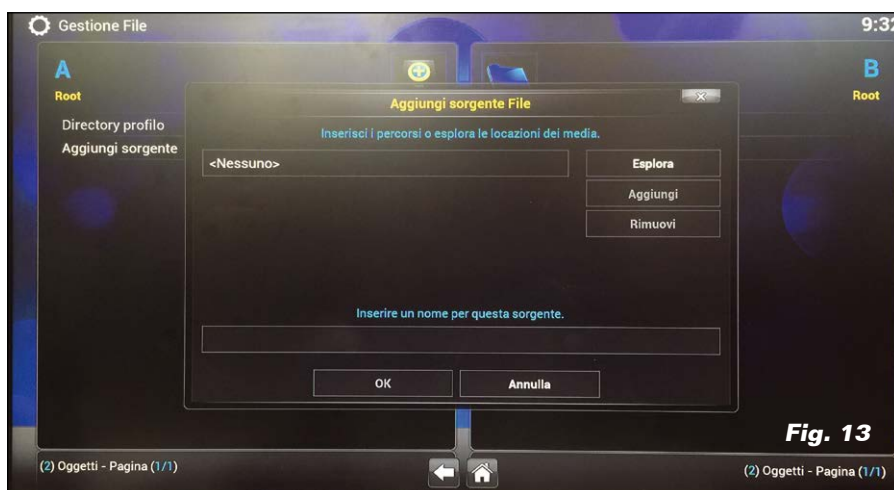


Fig. 13

connetta al nostro media center. Ora possiamo utilizzarlo come un vero e proprio telecomando per gestire il nostro media center.

App Android

Vediamo adesso la procedura per connettere uno smartphone Android (la maggior parte degli apparati in commercio, come Samsung, Lenovo, Asus, LG...); la prima cosa da fare è cercare in PlayStore l'applicazione "Yatse, the XBMC", quindi scaricarla ed aprirla.

Al solito attendiamo che il nostro sistema Raspberry Pi con OSMC venga rilevato, quindi selezioniamo il nostro dispositivo e clicchiamo su OK. Ora possiamo utilizzare lo smartphone come un telecomando per gestire il nostro media center.

INSTALLARE UNA REPOSITORY E GLI ADD-ON

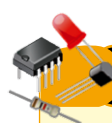
Passiamo adesso all'installazione di un repository da utilizzare in OSMC; per eseguire questa operazione è necessaria una connessione a Internet attiva e disponibile a Raspberry Pi. I passi da compiere sono:

- 1) andare su *SISTEMA>Gestione File* e, nella finestra che si apre (*Aggiungi sorgente File*, Fig. 13) clicchiamo su *Aggiungi*;
- 2) nella schermata che appare, dove c'è scritto *<Nessuno>* inserire l'URL della sorgente cui attingere e sotto il nome della sorgente, cliccare su *OK*;
- 3) andare nel percorso *Sistema>Impostazioni>Add-on* e qui clicchiamo su *Installa da un file zip*;
- 4) cliccare sul nome che abbiamo assegnato alla sorgente prece-

dentemente (nel nostro caso bisogna cliccare su *helix>repositories>superrepo>superrepo.kodi.helix.repositories.0.7.03.zip*); così verrà installato un add-on;

- 5) cliccare su *Scarica add-on>superRepo Repositories[Helix][v7]>add-on repository*;
- 6) scegliere la categoria degli add-on che intendiamo installare; per installarla, dobbiamo semplicemente cliccarci sopra;
- 8) tornare indietro alla scheda *scarica add-on*, dove adesso troveremo la categoria appena installata; se ci cliccassimo sopra per aprirla non conterrebbe alcunché, quindi dobbiamo cliccarci su con il pulsante destro del mouse e cliccare prima su *controlla aggiornamenti* e poi su *forza refresh*.

Adesso possiamo installare qualsiasi add-on contenuto in quella categoria; questo add-on poi lo troveremo nella scheda corrispondente. Se avremo installato un add-on inerente ai video, andremo nella scheda *Video>add-on video* e troveremo i nostri add-on installati. Ora siete pronti: buon divertimento con OSMC!



per il MATERIALE

Il Kit Raspberry Pi OSMC (cod. RASPKODI) comprende un Raspberry Pi 2, una micro SD card da 8GB con il sistema operativo installato, un cavo HDMI, un dongle WiFi, un contenitore plastico e un alimentatore switching, ed è disponibile presso Futura Elettronica al prezzo di 89,00 Euro 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>

Scheda scalabile per prototypare applicazioni di motion control sia di balancing robot sia di qualsiasi apparato su ruote.

OPEN WHEELS 2.0

dell' Ing. MIRCO SEGATELLO

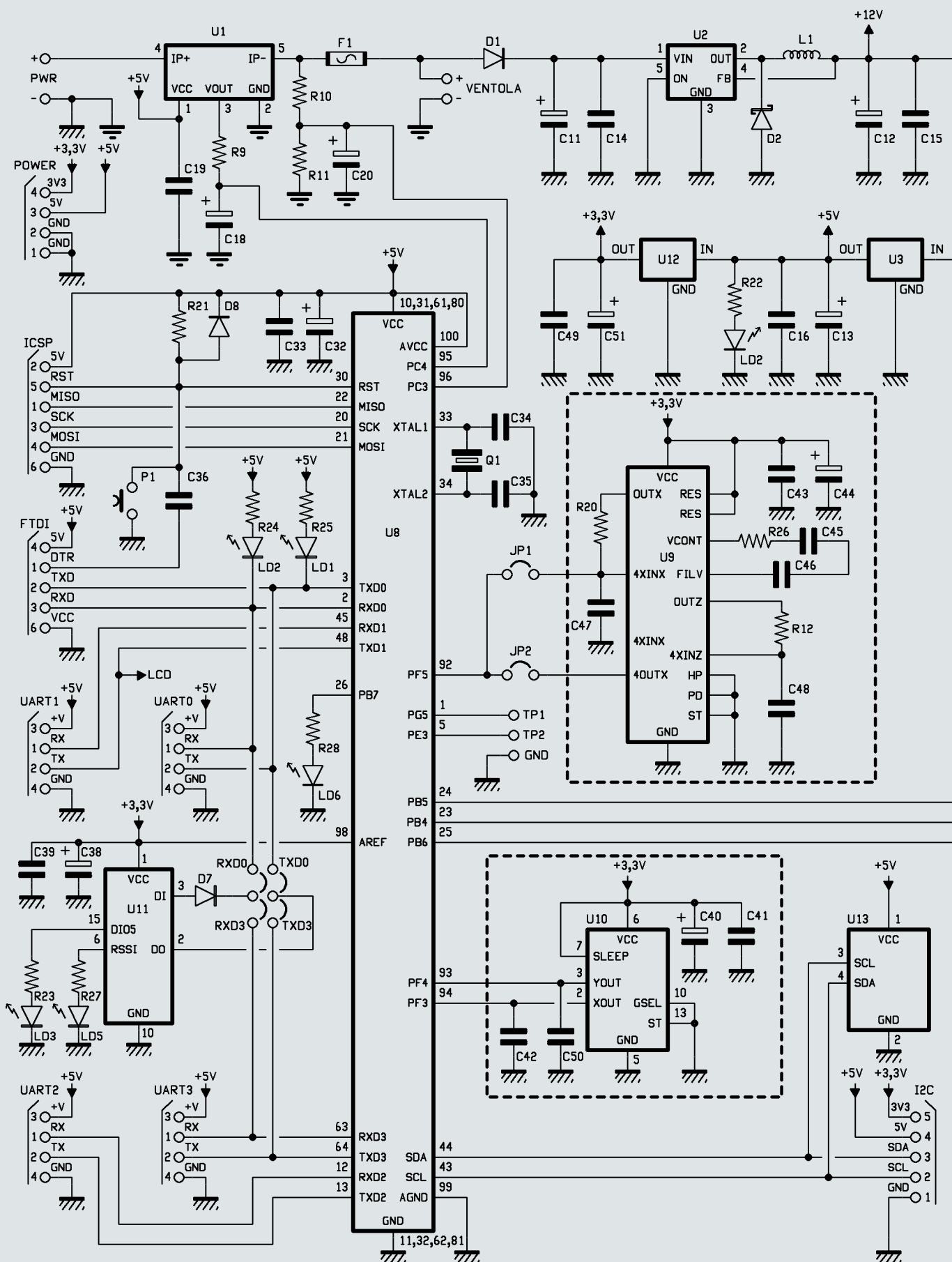
Era il 2013 quando presentammo una scheda di controllo per realizzare un clone del famoso SegWay: era il progetto OpenWheels. In questi ultimi anni l'attenzione verso i sistemi che permettono di stabilizzare un sistema intrinsecamente instabile non si è attenuata, anzi, basta guardare i numerosi progetti proposti in rete per vedere quanta gente si sia cimentata nel realizzare semplici (e non) robot equilibristi, meglio noti col termine anglosassone "balanced robot". Dal canto nostro, abbiamo evoluto il nostro primo progetto mettendo a frutto le nuove tecnologie che il mercato ci mette a disposizione; così è nata una nuova scheda che abbiamo chiamato OpenWheels2, la quale consiste in una board all-in-one utile a realizzare ogni vostro progetto rivolto alla movimentazione di robot. La scheda che andiamo a presentare è infatti equipaggiata (ed espandibile) con tutto l'occorrente per realizzare robot anche complessi, quindi non solo del tipo a bilanciamento. Il progetto è stato dirottato dal controllo di un clone di SegWay verso un robot bilanciato che chiameremo Personal Robot.

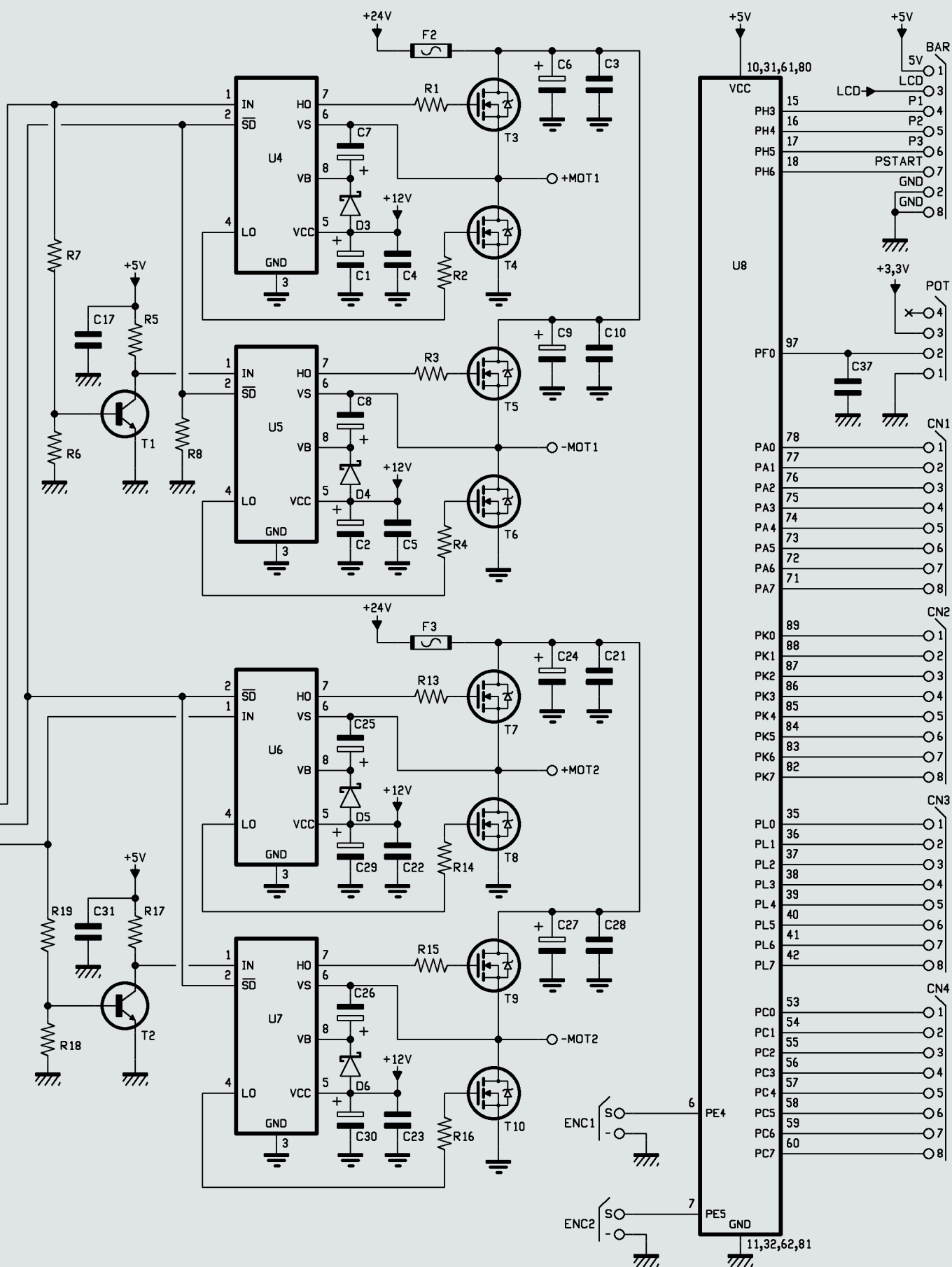
CARATTERISTICHE

La principale innovazione riguarda il processore, che passa da un ATmega328P ad un ATmega2560 mantenendo la compatibilità con Arduino e aumentando considerevolmente i pin disponibili e i moduli UART, molto utili per le funzioni di connettività. Il modulo IMU è ora più prestante, mentre gli stadi di alimentazione e a MOSFET sono rimasti invariati. La nuova scheda ha dimensioni simili a quelle della precedente e i connettori sono invariati, un facile aggiornamento a quanti abbiano realizzato OpenWheels.

Le caratteristiche della nuova scheda sono:

- compatibile con Arduino 2560;
- predisposta per utilizzare moduli XBee e Bluetooth;
- supporta sino a due motori DC con corrente massima di 20A (30A di picco);
- dispone di sensori di tensione e corrente sull'alimentazione;
- predisposta con ingresso per encoder;
- supporta moduli IMU sia analogici che digitali.





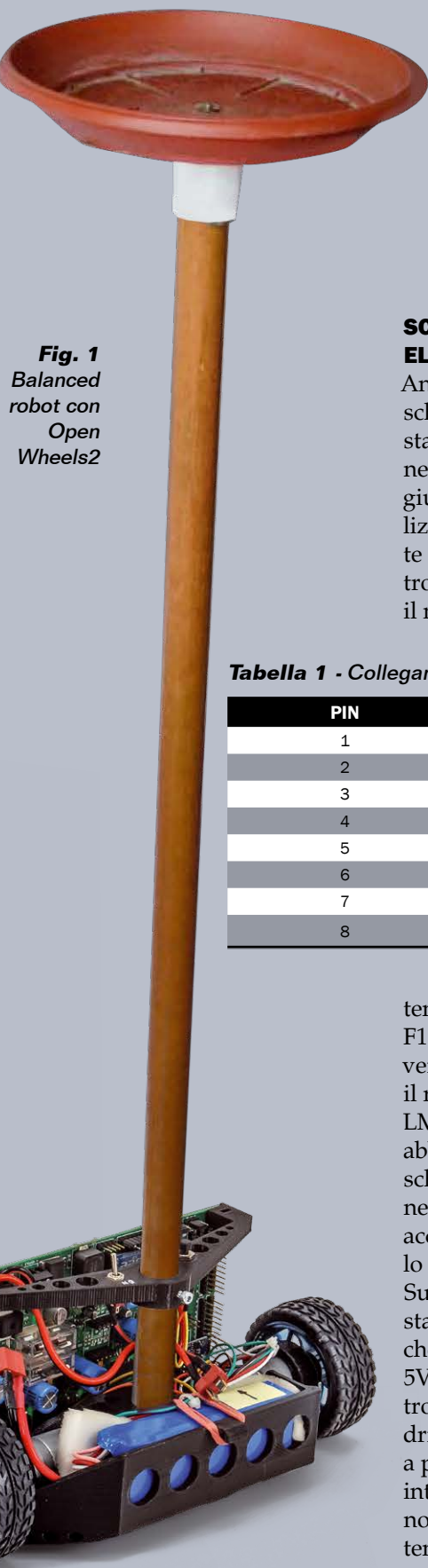
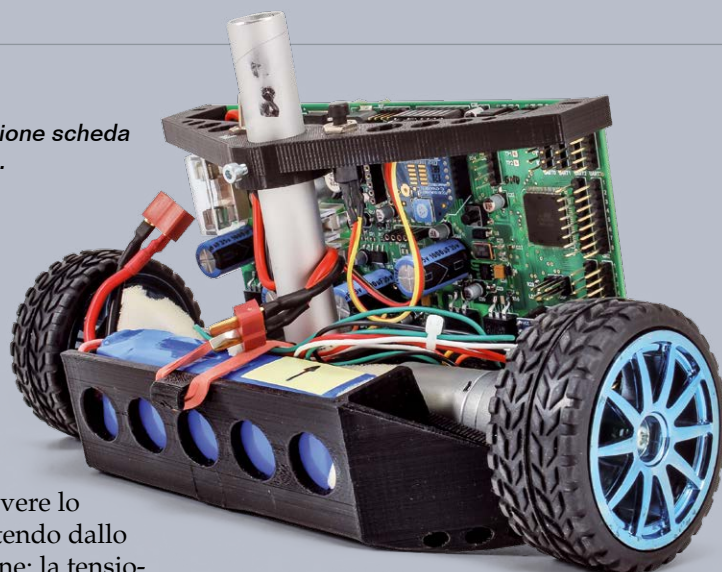


Fig. 1
Balanced
robot con
Open
Wheels2

Fig. 2
Installazione scheda
e motori.



SCHEMA ELETTRICO

Andiamo ora a descrivere lo schema elettrico, partendo dallo stadio di alimentazione: la tensione applicata ai contatti + e - PWR giunge all'integrato ACS758, utilizzato per la misura della corrente assorbita, all'uscita del quale si trovano il partitore R10/R11 che il micro utilizza per misurare la

di quella dell'alimentazione, così da pilotare il gate dei MOSFET. Per ciascun motore vengono utilizzati due integrati IRS2184 e quattro MOSFET pilotati con soli due segnali: uno è il segnale che abilita il driver (shutdown)

Tabella 1 - Collegamenti del connettore BAR.

PIN	Funzione
1	+5V
2	GND
3	LCD
4	P1 (-)
5	P2 (+)
6	P3 (E)
7	PSTART
8	GND

Tabella 2 - Collegamenti potenziometro sterzo connettore POT.

PIN	Funzione
1	GND
2	POT
3	+5V
4	+3V3

tensione della batteria, il fusibile F1 e il diodo di protezione dall'inversione di polarità, che precede il regolatore di tensione switching LM2576 (U2) utilizzato per abbassare la tensione a 12V. La scheda nasce per un'alimentazione a 24V e, senza alcuna modifica, accetta fino a 11,5V; bypassando lo switching è possibile scendere. Sulla linea dei 12V troviamo due stabilizzatori di tensione lineari che forniscono rispettivamente i 5V e i 3,3V necessari al microcontrollore e al modulo IMU. I due driver motori, in configurazione a ponte H, sono realizzati con integrati IRS2184, i quali sfruttano un sistema di elevamento di tensione a pompa capacitiva per ottenere una tensione maggiore

mentre il secondo è il segnale PWM di comando configurato in modalità locked-antiphase. In questa modalità, il segnale PWM giunge a entrambi i rami del ponte H ma in opposizione di fase: così la ruota rimane ferma con un duty-cycle del 50%. Ruoterebbe alla massima velocità in una direzione con un duty-cycle di 0% e alla massima velocità in senso opposto con duty-cycle del 100%. Nel caso di balanced-robot in cui le ruote presentano continui cambi di direzione (per mantenere la posizione verticale); tale modalità permette di far lavorare i MOSFET di potenza con un duty-cycle attorno al 50%, evitando segnali impulsivi molto rapidi come accade nella classica mo-

dalità sign-magnitude, in cui un segnale è adibito alla direzione ed uno al PWM. In questi ultimi driver, quando il robot si trova nella posizione di equilibrio il segnale PWM ha un duty-cycle molto basso ed il segnale di direzione ha continui cambiamenti di stato; se per motori che assorbono poche centinaia di milliampere ciò non costituisce un problema, nel caso di motori che assorbono diversi ampere può diventarlo. I MOSFET previsti nel progetto sono del tipo AUIRF2804, capaci di gestire correnti impulsive di 200A e correnti continue di almeno 10A, senza riscaldare apprezzabilmente; abbiamo comunque previsto un connettore cui connettere una ventola per il raffreddamento forzato. Tre, sono i fusibili a protezione della scheda: uno da 1A per i circuiti di comando e due sulle due linee dei motori; questi ultimi vanno scelti in modo da interrompere il circuito di potenza in presenza di picchi di corrente eccessivi, solitamente presenti al momento dell'alimentazione da fermo dei motori. Il microcontrollore è un ATmega2560, lo stesso montato sulla scheda Arduino Mega, che, rispetto all'ATmega328 (della Arduino UNO), dispone di un maggiore numero di pin e di ben quattro moduli UART per la comunicazione seriale hardware. Gli ingressi analogici sono utilizzati per la lettura della corrente e della tensione di alimentazione, oltre che del potenziometro dello sterzo (se presente) e dell'accelerometro e del giroscopio analogici, qualora siano presenti. È previsto un connettore di nome BAR, cui connettere dei pulsanti e un eventuale display LCD secondo la **Tabella 1**, ed un connettore per il potenziometro dello sterzo (**Tabella 2**). Sulla scheda sono presenti i so-

cket per l'accelerometro analogico MMA7361 e per il giroscopio analogico LPR403, oltre a quello per il modulo XBee; quattro connettori rendono disponibili i segnali TX ed RX degli altrettanti UART del microcontrollore, con relativa alimentazione. La seriale principale Serial0 fa capo al connettore FDTI, al quale può essere connesso un convertitore USB/seriale utile alla programmazione del microcontrollore, mentre la seriale Serial3 viene utilizzata per la telemetria e connessa al socket per il modulo XBee tramite i jumper XB1TX e XB1RX; la seriale Serial1 viene invece usata per il display LCD. Nella parte inferiore della scheda troviamo il connettore per il modulo IMU digitale MPU-6050 della InvenSense e quello per la porta di comunicazione I²C-Bus, al quale collegare qualsiasi altro modulo IMU digitale. È presente anche un connettore per l'ingresso dei segnali provenienti da due encoder ed uno per l'alimentazione generica di moduli esterni. Tutti i pin dell'ATmega2560 non utilizzati nel progetto sono comunque resi disponibili sui

Tabella 3 - Funzionamento dei LED

LED	Funzione
LD1	UART0 TX (USB/seriale monitor)
LD2	Presenza 5V
LD3	XBee ASS
LD4	UART1 RX (USB/serial monitor)
LD5	XBee RSSI
LD6	Connesso al pin 13 di Arduino

connettori CN1,CN2,CN3 e CN4, utili per ogni eventuale espansione che desideriate implementare. Per ultimo abbiamo il connettore ICSP, che, come vedremo, sarà utilizzato per caricare il bootloader sul microcontrollore.

Sono stati implementati diversi LED per segnalare lo stato di funzionamento, come riportato nella **Tabella 3**.

Conclusa la descrizione dello schema, passiamo all'utilizzo pratico della scheda cimentandoci nella realizzazione di un balance-robot particolarmente sofisticato, in grado di mantenersi perfettamente bilanciato anche nelle situazioni più critiche.

LA MECCANICA DEL ROBOT

Partiamo dalla meccanica, stampata con la nostra 3Drag ed opportunamente disegnata per accogliere l'elettronica (i file STL li potete scaricare dal nostro sito

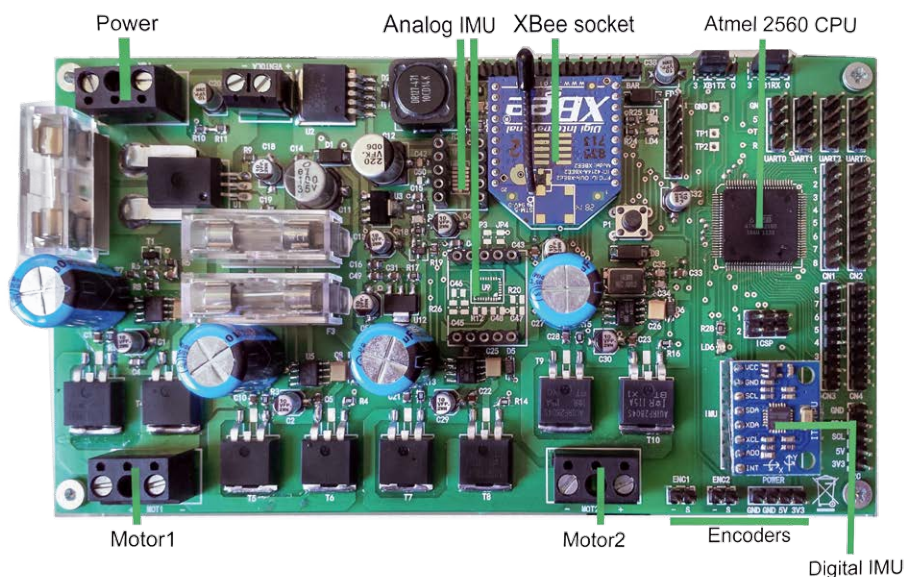


Fig. 3 - Connettori della scheda di controllo.

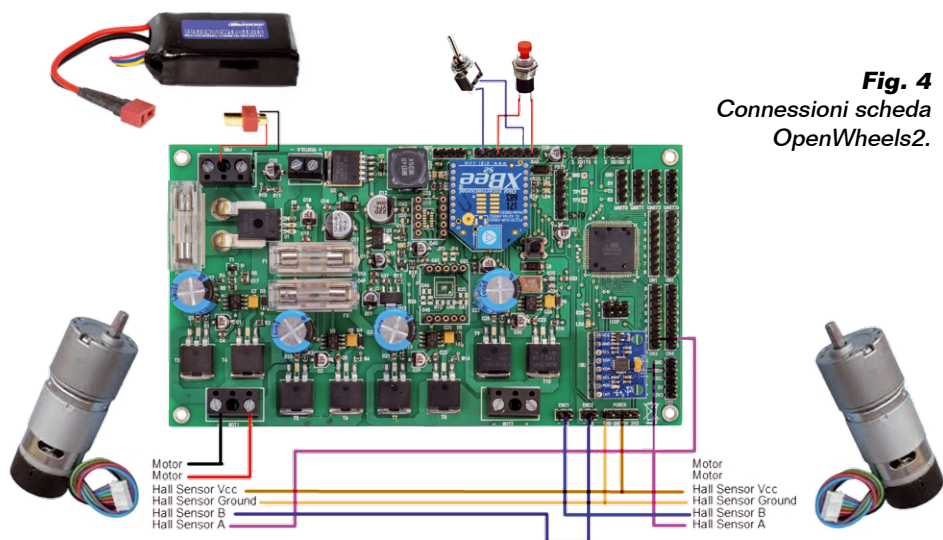


Fig. 4
Connessioni scheda
OpenWheels2.

www.elettronica.in.it). Su questo basamento andremo a fissare i motori, la batteria e la scheda OpenWheels2. Sulla base fissiamo un'asta ricavata da un tubo di alluminio, sulla cui sommità abbiamo posto per le prove un piccolo vassoio che ci permetterà di contenere anche degli oggetti o, come abbiamo fatto noi, dei drink; nulla vieta di applicare qualsiasi altra cosa.

Alla scheda di controllo vanno collegati i seguenti dispositivi:

- 2 motoriduttori con encoder e ruote (2846-BALANCINGBOT);
- batteria LiPo 1.800 mAh/11,1 V (5400-LIPO1800);
- connettore maschio/femmina (7300-TCONNMF);
- deviatore unipolare a pulsante (8220-8701);
- deviatore unipolare a levetta per c.s. a 90° (8220-TS-8);
- 2 moduli Xbee 2 mW serie 2 (6168-XBEE2MW);
- XBEE EXPLORER USB (6168-XBEE2DONGLE2);
- modulo Bluetooth controller DAGU (2586-DG010);
- accelerometro 3 assi + Giroscopio a 3 assi (2846-MPU6050);
- convertitore FTDI USB/seriale 3,3V e 5V (7300-FTDI5V);
- fusibile (F2) 1A;
- fusibile (F1,F3) 5A.

I moduli XBee e Bluetooth servono per la telemetria. I componenti si trovano tutti presso Futura Elettronica (www.futurashop.it) e tra parentesi trovate i rispettivi codici.

Nel caso si utilizzasse un'asta lunga, potrebbe essere necessario prevedere un piccolo peso di circa 100 g per bilanciare il peso alla base dell'asta, dovuto ai motori ed alla batteria; quest'ultima avremmo potuto posizionarla in alto, ma avrebbe richiesto lunghi cavi che avrebbero introdotto una resistenza elettrica deleteria. Anche se non sembra, il peso posto in alto migliora la stabilità minimizzando nel contempo piccole oscillazioni dovute alla cedevolezza dei materiali.

A questo punto fissate i motori al supporto stampato in 3D, le relative ruote e la scheda elettronica, come visibile nelle foto dell'articolo; l'unica precauzione è fissare a dovere la scheda. Per quanto riguarda i collegamenti elettrici, occorre fare molta attenzione, soprattutto per i motori che hanno incorporato l'encoder; riferitevi alla Fig.3.

È previsto un interruttore che sarà utilizzato per armare il robot connesso secondo le indicazioni della Tabella 4 ed un pulsante per il reset degli allarmi. Se volete,

potete inserire un interruttore di accensione in serie alla batteria. La Fig. 4 illustra il cablaggio.

IL FIRMWARE

Il firmware è derivato da quello di OpenWheels, migliorato ed arricchito di nuove funzionalità. Per caricarlo è necessario installare il bootloader nell'ATmega; allo scopo possiamo usare un apposito programmatore, oppure una comune scheda Arduino usata come programmatore.

In quest'ultimo caso aprite l'IDE di Arduino e caricate lo sketch di esempio denominato *Arduinolisp.ino* sulla scheda Arduino, quindi impostate come scheda target "Arduino Mega2560 or Mega ADK" accedendo al menu Strumenti>tipo di Arduino, poi impostate Arduino come programmatore (Strumenti>Programmatore>Arduino as ISP).

A questo punto avviate la programmazione con il comando Strumenti>Scrivi il Bootloader. A procedura completata (sono richiesti alcuni minuti) sconnettete il programmatore ed interfacciatevi direttamente al PC tramite il connettore FDTI con l'ausilio di un convertitore USB/Seriale: la vostra scheda sarà vista dall'IDE come una Arduino Mega.

Per verificare che tutto sia funzionante, caricate lo sketch di esempio *Blink.ino* e accertatevi che il LED LD6 lampeggi.

Per il momento potete omettere il modulo XBee ed assicurarvi che il modulo IMU sia correttamente installato; potete quindi fare alcuni test per verificare che tutto funzioni (consigliamo di farlo prima di installare la scheda ed effettuare le connessioni).

Testiamo i sensori caricando sulla scheda lo sketch di nome *PersonalRobot_TestSensor.ino* disponibile con i file del progetto su www.elettronica.in.it: i dati

Tabella 4 - Collegamenti elettrici motore sinistro

Componente	pin usati
Interruttore stand-by/Go	Connesso tra i pin 2 e 7 del connettore BAR
Pulsante di reset (NA)	Connesso tra i pin 4 e 8 del connettore BAR

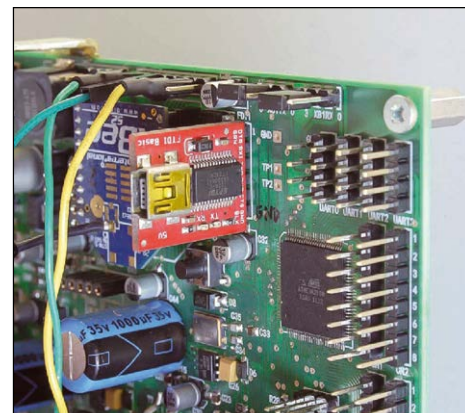
Tabella 5 - Collegamenti per caricare il bootloader usando una scheda Arduino come programmatore.

Arduino (programmatore)	Openwheel2 (da programmare)
Arduino pin D12/MISO	ICSP pin 1
Arduino pin +5V	ICSP pin 2
Arduino pin D13/SCK	ICSP pin 3
Arduino pin D11/MOSI	ICSP pin 4
Arduino pin D10/RESET	ICSP pin 5
Arduino pin GND	ICSP pin 6

acquisiti saranno visualizzati direttamente su SerialMonitor di Arduino in modo grezzo, senza alcuna post-elaborazione. Con un semplice jumper potete simulare i pulsanti cortocircuitando gli ingressi P1, P2, P3, PStart verso il pin GND; connettendo un display seriale sui pin +5V, GND, LCD del connettore BAR potrete verificarne il funzionamento. Ricordiamo che il display LCD è supportato sia nell'hardware che nel software, ma non utilizzato in questo progetto in quanto, come vedremo, abbiamo previsto altri sistemi per la visualizzazione dei dati. Se sono collegati i motori, è possibile testarli caricando lo sketch *OpenWheelsTestMotorV2.ino*; il comando dei motori avviene da SerialMonitor con il semplice invio di caratteri. La cosa importante è che i motori ruotino nella direzione corretta e che il robot avanzi quando le ruote girano in senso orario.

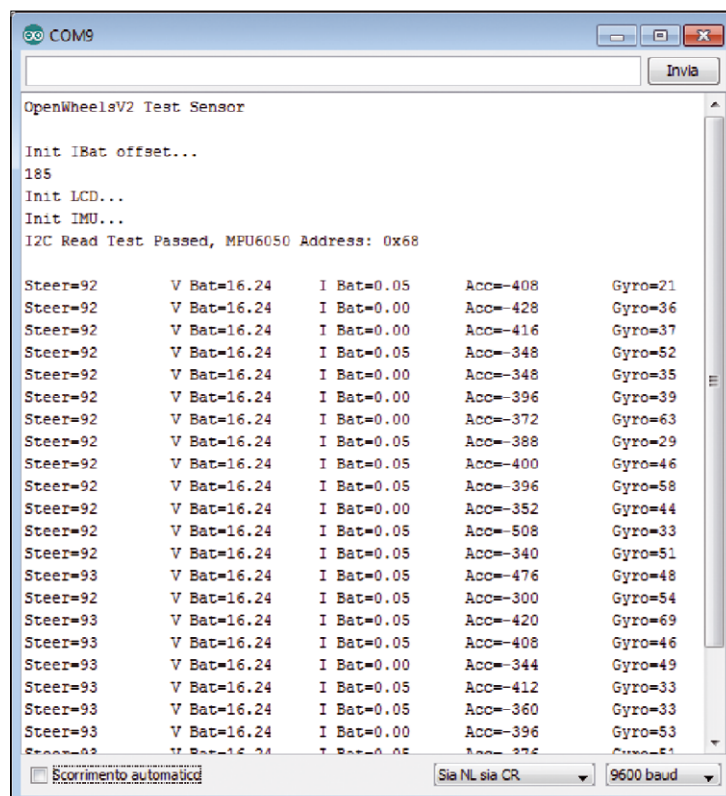
Una delle più importanti innovazioni rispetto alla precedente versione è l'utilizzo di un modulo IMU digitale basato sull'integrato MPU-6050 della InvenSense, che contiene un accelerometro e un giroscopio entrambi a tre assi, oltre ad un microprocessore idoneo ad elaborare i dati acquisiti. La precisione e la risoluzione sono enormemente aumentati ed il livello di rumore che affligge da sempre gli accelerometri, qui

è molto basso. Questo modulo che spesso si trova in commercio come breakout board con la sigla GY-521, è un ottimo compromesso tra prestazioni e costi; inoltre la sua interfaccia di comunicazione I²C è perfetta per l'abbinamento con Arduino. Come sempre la lettura dell'accelerometro e del giroscopio vanno fuse assieme al fine di minimizzare l'inevitabile drift dell'offset del giroscopio e limitare il rumore dell'accelerometro; tra le varie soluzioni, il

Fig. 5 - Montaggio del convertitore USB/seriale.

filtro complementare si è rivelato efficace e facile da implementare (chi volesse approfondire l'argomento troverà interessante questo tutorial in lingua italiana: <http://www.gioblu.com/tutorials/sensori/193-filtro-complementare-e-filtro-di-kalman>).

Il firmware necessario a far funzionare il nostro robot si chiama *PersonalRobot_V10.ino* ed è scaricabile insieme a tutti gli altri file del progetto. Partiamo dicendo che nel progetto la scheda

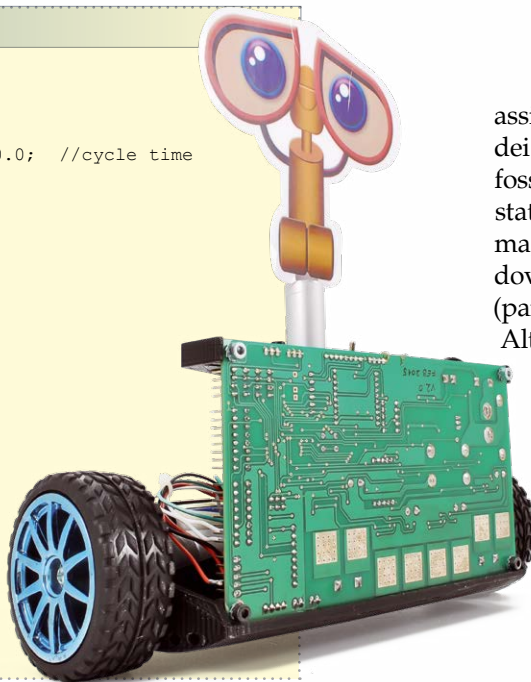
Fig. 6
Serial monitor con i dati rilevati durante il test della sensoristica.

Listato 1

```
void loop()
{
  time2=micros();
  dt = float(time2-time1)/1000000.0; //cycle time
  time1=time2;

  SoftStart();
  EstimateAngle();
  EstimatePos();
  Status();
  PID();
  SetMotor();
  SetZero();
  Telemetry();

  if (digitalRead(P1_pin) == 0)
  {
    // reset error
    statusFlag=0;
    errorIMU=0;
    Integral=0;
    encoderCounterLeft=0;
    encoderCounterRight=0;
  }
  delay(18); //about 50Hz
}
```



è posta in verticale, ma nulla vieta di posizionarla in orizzontale; in tal caso è necessario configurare il firmware lasciando la sola riga di codice utile:

```
//horizontal board
//Wire.write(MPU6050_ACCEL_YOUT_H);

//vertical board
Wire.write(MPU6050_ACCEL_ZOUT_H);
```

Lo sketch è suddiviso in blocchi, ciascuno dei quali ha un file distinto in base alla funzione svolta; il file *PersonalRobot_V10.ino* contiene il codice principale dal quale sono richiamate le funzioni via-via necessarie.

Il ciclo principale contiene solo le chiamate alle funzioni e la misura del tempo di ciclo (**Listato 1**). L'intero ciclo di stabilizzazione dura circa 2 ms, ma con una funzione delay lo portiamo a un più ragionevole 20 ms, ovvero 50 volte al secondo; la telemetria influenza molto il tempo di ciclo, che comunque viene ricalcolato ad ogni ciclo. Nel file *EEPROM.ino* sono presenti le funzioni necessarie a leggere e scrivere nella memoria EEPROM dell'ATmega2560: nel nostro caso, tutti i parametri

del controllo PID ed i valori di offset dei sensori. Nel file *IMU.h* sono contenute le funzioni che permettono di inizializzare i sensori, misurarne l'offset e i valori istantanei. In applicazioni di questo tipo il firmware è molto critico e contempla diversi parametri che devono assumere un ben preciso valore a seconda dell'hardware utilizzato; è sufficiente modificare, ad esempio, l'altezza dell'asta del robot, per modificarne il baricentro ed il momento d'inerzia; immaginate cosa può accadere sostituendo ad esempio i motori o variando qualche peso.

Oltre ai parametri canonici del controllo PID, ce ne sono alcuni meno critici ma ugualmente importanti, che fanno la differenza tra un robot che sta in verticale ed uno che rimane in verticale anche se gli si appoggia un peso all'improvviso; in particolare, nella funzione *MPU6050_init()* osserviamo la riga di impostazione del modulo *MPU6050_write(MPU6050_CONFIG, 0x04)* che imposta un'azione di filtro dei dati dell'accelerometro e del giroscopio alla frequenza di 20 Hz (parametro *DLPF_CFG=4*)

assicurando una pulizia, in lettura, dei rumori indesiderati. Se il robot fosse stato più piccolo e fossero stati necessari tempi di reazione maggiori, questo valore avrebbe dovuto necessariamente salire (parametro *DLPF_CFG* più basso).

Altro parametro relativamente importante è la frequenza di taglio del filtro complementare che fonde le letture dei sensori. La riga in cui viene definito è questa: *float Comp_Filter_Constant = 0.01*; valori più bassi rallentano la misura dell'angolo (ma attenuano di più il rumore) mentre valori più alti permettono di ottenere una misura angolare più veloce ma maggiormente affetta da rumore.

La sezione che gestisce gli encoder è contenuta nel file *Encoder.ino*, sezione implementata ex-novo proprio per OpenWheels2: se si realizza un balanced robot, diventa importante non solo la stabilità verticale, ma anche il mantenimento di un preciso punto rispetto al suolo, altrimenti può capitare che il robot se ne vada in giro per la stanza senza controllo a causa del drift di qualche sensore. Gli encoder forniscono informazioni circa la rotazione delle ruote, quindi è possibile valutare quando il robot si è mosso. Utilizzando gli encoder su entrambe le ruote possiamo anche avere informazioni sulla differenza di rotazione tra le due, cosa che porterebbe il robot a ruotare su se stesso a causa delle differenze tra i motori. Per questo motivo vengono determinate le variabili **WheelsPos (DEG)**, che fornisce la posizione delle ruote in gradi, **WheelsVel**, che fornisce la velocità di rotazione delle ruote in gradi al secondo, e la **encoder-Dif**, che fornisce indicazione sulla differenza di rotazione tra le ruote in gradi.

A ciascuno dei due encoder è associato un interrupt:

```
attachInterrupt(0, EncoderLeft,
FALLING); //encoder pin2 int0 (ENC1)
attachInterrupt(1, EncoderRight,
FALLING); //encoder pin3 int1 (ENC2)
```

Per sapere se la ruota sta girando in senso orario (e quindi incrementare il conteggio) oppure in senso antiorario (per decrementare il conteggio) dobbiamo sfruttare entrambi i segnali in quadratura forniti dagli encoder. Il secondo segnale è quindi connesso ad un ingresso digitale della scheda e la sua lettura durante l'interruzione fornisce indicazioni sul verso di rotazione della ruota (vedere **Listato 2**).

Il file denominato *PID.ino* è il cuore di tutto il sistema, perché contiene le righe di codice necessarie alla stabilizzazione del robot (**Listato 3**). Si tratta, in definitiva, di un calcolo che tiene conto dei vari contributi in base a parametri che l'utente dovrà inserire in modo molto preciso. Vediamone il significato:

- Pval, controllato dal parametro KP, è il valore proporzionale (più il robot si inclina e più è necessario compensare);
- Ival, controllato dal parametro KI, è il valore integrativo e viene usato solo in cloni di SegWay e controlla la funzione per cui ad andatura costante è possibile riportare il manubrio in posizione verticale;
- Dval, controllato dal parametro KI, è il valore derivativo e fa sentire il suo effetto tanto quanto maggiore è la tendenza a cadere;
- Ppos, controllato dal parametro KO, è il valore proporzionale per il posizionamento; maggiore è questo valore e maggiore sarà la precisione con la quale il robot mantiene la posizione rispetto al suolo;

- Pvel, controllato dal parametro KV, è il valore che insieme al valore Ppos permette di evitare che il robot si allontani dalla posizione iniziale.

Nel software di telemetria c'è la possibilità di impostare il parametro STEER, utile per impostare la sensibilità dello sterzo nel caso il progetto riguardi il clone di un Segway. Un ulteriore parametro denominato Kdm permette di mantenere l'orientamento originale del robot compensando eventuali differenze nella rotazione delle ruote. Ovviamente gli ultimi tre valori devono essere azzerati in tutti i casi in cui si vuole che il robot si sposti, magari perché comandato a distanza.

Analizzando il codice troviamo anche una riga utilizzata per compensare il livello di carica della batteria: $drive = drive * VBatAlim / VBatLevel$; infatti a parità di potenza calcolata, quella effettiva sulle ruote, dipenderà dal livello di carica della batteria e noi non vogliamo che il robot si comporti in modo differente a seconda del

livello della batteria.

Il file *Motor.ino* (**Listato 4**) contiene le impostazioni e le funzioni per gestire i motori, primo fra tutti l'impostazione dei registri interni dell'Atmel, al fine di comandare i motori con un segnale PWM ad alta risoluzione (10bit) invece dei canonici 8bit usati per default in Arduino; inoltre la frequenza del PWM viene portata a valori di 16 kHz, assicurando un movimento più fluido e silenzioso dei motori. Sempre in questa sezione, troviamo altri due parametri denominati *steerSensibilty* e *speedSensibility* che, come suggerito dai loro nomi, definiscono la sensibilità rispettivamente nella sterzata e nella velocità, qualora il robot venga comandato a distanza (**Listato 5**). In questa sezione troverete anche la compensazione della cosiddetta *motor dead zone*, che è il livello di tensione sotto il quale le ruote non girano. Se ad esempio il robot ha bisogno del 5% di potenza sui motori, ma questa non è sufficiente a vincere gli attriti, le ruote non gireranno, con il risultato che il robot tenderà a cadere e richie-

Listato 2

```
void EncoderRight()
{
  // If pinA and pinB are both high or both low, it is spinning
  // forward. If they're different, it's going backward.
  if (PINL & 0b00000010)
    encoderCounterRight++;
  else
    encoderCounterRight--;
}
```

Listato 3

```
Pval = float(KP) * Angle;
Ival = float(KI) * Integral;
Dval = float(KD) * Gyro_Rate;
Ppos = float(KO) * wheelsPos;
Pvel = float(KV) * wheelsVel;
drive = Pval + Ival + Dval + Ppos + Pvel;
drive = drive * VBatAlim/VBatLevel; // for compensate Battery level
drive = constrain(drive, -250.0, 250.0); //limit drive values
```

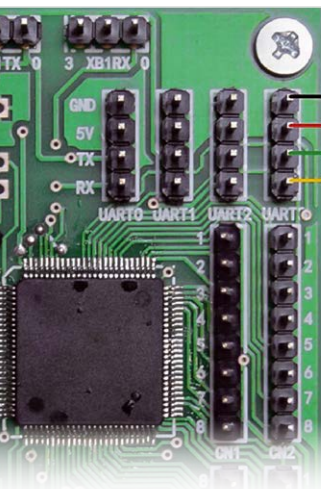


Fig. 7
Collegamento del convertitore USB/seriale per telemetria via cavo.

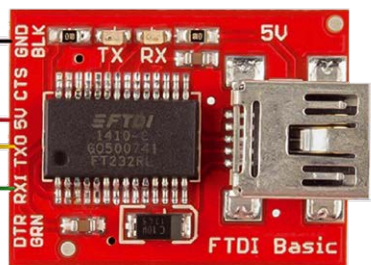
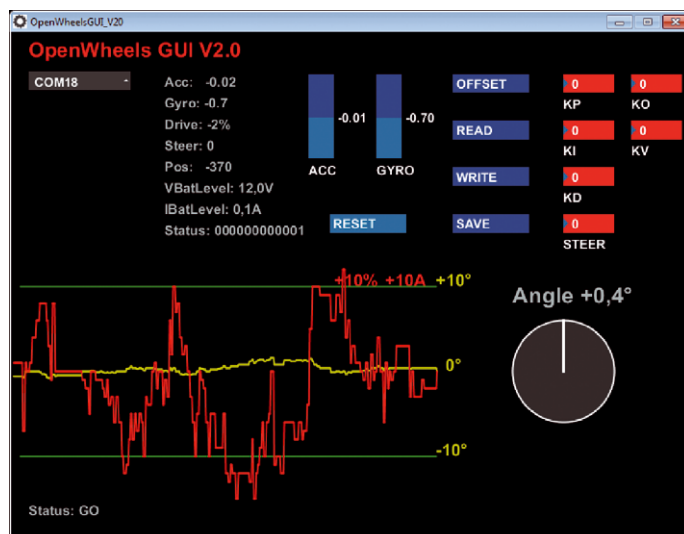


Fig. 8
Software per telemetria.



derà un'ulteriore correzione, con il risultato di un bilanciamento incerto. Una semplice prova con il robot appoggiato al suolo vi permetterà di determinare il minimo valore di potenza alle ruote affinché il robot si muova. La riga di codice che definisce questo parametro è: `int deadZoneMotor = 10`. La sezione *SerLCD.ino* attualmente non è utilizzata ed è stata lasciata per la compatibilità con il precedente hardware. Il file *telemetry.ino* contiene le funzioni utilizzate per le comunicazioni con la scheda; nello specifico, per richiedere dei dati o impostare dei parametri. In ogni caso è l'apparecchiatura esterna a dover far richiesta di dati, in quanto Openwheels2 non invia in automatico alcun dato; si veda, al riguardo, la **Tabella 6**. Ulteriori parametri di contorno sono usati per controllare lo stato di funzionamento ed eventualmente intervenire con degli allarmi; è il caso dei seguenti tre valori di tensione riferiti rispettivamente al valore massimo, minimo e nominale di alimentazione:

```
float VBatLmax = 12.6; //Max batt Volt
float VBatLmin = 11.5; //Min batt Volt
float VBatAlim = 11.8; //Nominal batt Volt
```

Tali valori sono usati, oltre che per gli allarmi, anche per la funzione di compensazione di potenza sui motori.

Notare come a causa del driver di potenza la tensione minima di funzionamento non può essere inferiore a 11,5 volt, pena l'impossibilità di portare i MOSFET nello stato di ON. Un ulteriore valore riguarda la massima corrente oltre la quale interviene lo shutdown dei motori; la funzione interviene in circa 20 ms, quindi non ha la stessa rapidità dei fusibili, ma offre una valida protezione nella maggior parte dei casi:

```
float IBatMAX = 3.0;
// IBat MAX for shutdown MOTOR
```

Ulteriori interventi di protezione riguardano le variabili riportate nella **Tabella 7**. Il primo bit della variabile `statusFlag` non contiene un errore ma indica se il robot è in stato di standby oppure è armato, ovvero attivo e funzionante; questa funzionalità è gestita tramite l'apposito interruttore. Nello stato di standby tutti i sistemi sono pienamente funzionanti, compresa la telemetria, ma il driver è disabilitato e le ruote non possono girare: potete quindi gestire il robot senza la preoccupazione di fare danni.

Listato 4

```
void initMotor()
{
    pinMode(ShutDown_pin, OUTPUT);
    digitalWrite(ShutDown_pin, LOW); //ShutDown driver at startUP
    pinMode(PWMLLeft_pin, OUTPUT);
    pinMode(PWMLRight_pin, OUTPUT);

    // Phase and Frequency correct PWM on TIMER1
    //PWM_frequency = (16000000)/(2*Prescale_factor*ICR1)
    //ICR1=(16000000)/(2*Prescale_factor*PWM_frequency)
    // ICR1=400 fPWM=20000
    // ICR1=500 fPWM=16000
    // ICR1=666 fPWM=12012
    // ICR1=1024 fPWM=7812
    TCCR1A = _BV(COM1A1) | _BV(COM1B1);
    TCCR1B = _BV(WGM13) | _BV(CS10);
    ICR1=500;

    // *****
    // limit for driver compatibiliti (5% di ICR1A) < drivePWM < (95% di ICR1A)
    // stop motor driver=stallMotor=ICR1/2;
    // *****
    maxPWM = float(ICR1)*0.80/2; //0.95
    stallMotor=ICR1/2; //25
    OCR1A = stallMotor; //0=0% ICR1/2=50% ICR1=100%
    OCR1B = stallMotor;
}
```


Listato 5

```
void SetMotor()
{
    Steer = steerSensibility * float(int(dataX)-127);
    Speed = speedSensitivity * float(int(dataY)-127);

    drivePWM_L = int( (drive*goStart) - Kdm*float(encoderDif) + Steer );
    drivePWM_R = int( (drive*goStart) + Kdm*float(encoderDif) - Steer );

    // compensate motor dead band
    if (drivePWM_L>0) drivePWM_L += deadZoneMotor;
    if (drivePWM_L<0) drivePWM_L -= deadZoneMotor;
    if (drivePWM_R>0) drivePWM_R += deadZoneMotor;
    if (drivePWM_R<0) drivePWM_R -= deadZoneMotor;

    dutyCycleLeft = stallMotor + drivePWM_L; // + Steer;
    dutyCycleRight = stallMotor + drivePWM_R; // - Steer;

    dutyCycleLeft = constrain(dutyCycleLeft, stallMotor-maxPWM, stallMotor+maxPWM);
    dutyCycleRight = constrain(dutyCycleRight, stallMotor-maxPWM, stallMotor+maxPWM);
    OCR1B = dutyCycleLeft; //set PWM SX 250+-250
    OCR1A = dutyCycleRight; //set PWM DX 250+-250
}
```

Il LED LD6 lampeggia durante uno stato di errore; solo premendo il pulsante di reset (o tramite comando remoto) è possibile resettare gli allarmi. Vista la complessità dell'impostazione dei parametri e la necessità di visualizzare lo stato operativo, abbiamo implementato un apposito software (*OpenWheelsGUI_V20*) per la gestione in remoto di tutte le principali operazioni, che permetta di visualizzare tutti i dati di telemetria anche graficamente. Il software è stato scritto con Processing, perché è un ambiente di sviluppo che supporta diversi sistemi operativi.

Per poter utilizzare il software di configurazione dovete montare il convertitore USB/Seriale, così da dialogare con la scheda tramite la seriale3 del microcontrollore, disponibile sul connettore UART3: è sufficiente cablare le linee TX,RX e GND; la linea +5V non è necessaria perché la scheda ha alimentazione propria. Il software è molto intuitivo e una volta avviato è sufficiente selezionare la seriale utilizzata per la comunicazione: il software provvederà, ad intervalli regolari, ad interrogare il robot. Tutti i principali dati interni sono visualizzati, mentre il valore dell'inclinazione, la corrente assorbita e la potenza dei motori sono anche visibili in un grafico. I campi in rosso permettono l'immissione dei parametri semplicemente trascinandovi sopra il puntatore del mouse. Il pulsante RESET consente il reset di tutti gli allarmi, OFFSET consente la determinazione dell'offset dei sensori, READ permette di leggere i parametri PID attualmente utilizzati, WRITE invia i parametri a video, mentre il pulsante SAVE consente di salvare nella memoria EEPROM del microcontrollore i parametri PID attualmente utilizzati.

CONFIGURAZIONE TELEMETRIA WIRELESS

La telemetria/configurazione wireless è decisamente più comoda, visto che il controllo completo del robot può avvenire in tempo reale mentre è in bilanciamento, è quindi semplice modificare i parametri al volo e vederne il comportamento in tempo reale.

La telemetria che consente la maggior portata può essere ottenuta con moduli XBee: anche la serie 2 da 2mW di potenza assicura una buona portata delle comu-

nicazioni. I moduli XBee vanno programmati prima di essere utilizzati, perciò è necessario scaricare il software XCTU della Digi all'indirizzo <http://www.digi.com/products/wireless-wired-embedded-solutions/zigbee-rf-modules/xctu>. Per la nostra applicazione è sufficiente che i due moduli comunichino in modalità *transparent mode*. In questa modalità due moduli (e solo loro) possono sostituire a tutti gli effetti una comunicazione cablata. Per prima cosa inserite uno dei due moduli XBee nell'USB dongle

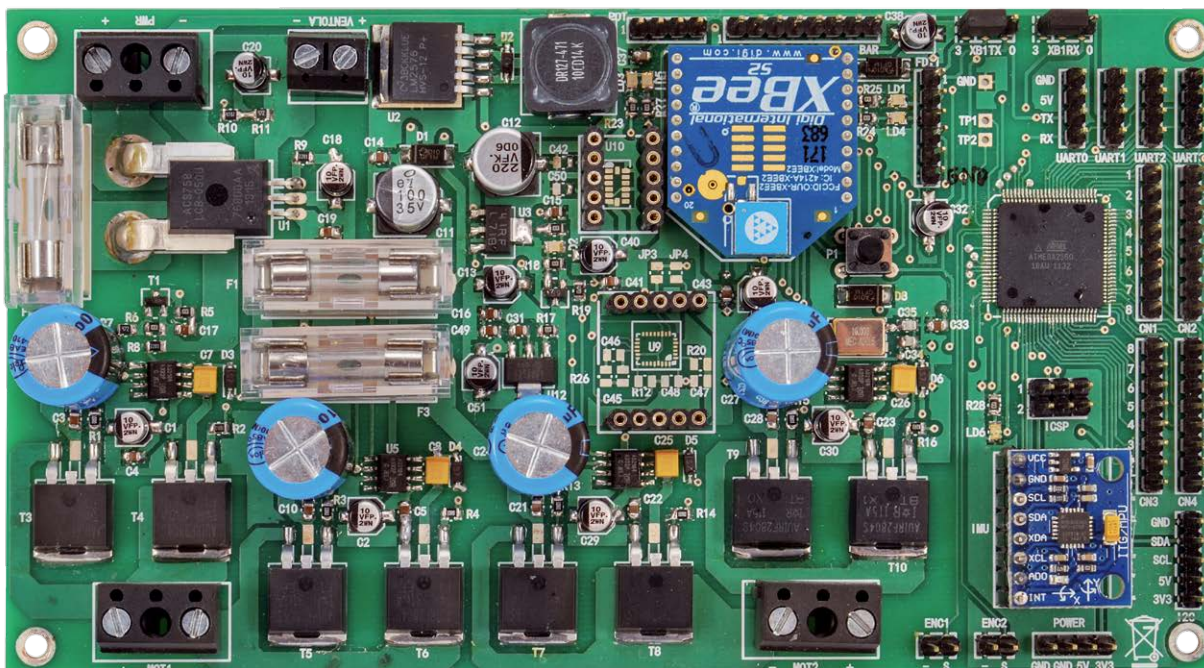
Tabella 6 - Protocollo di scambio dati con OpenWheels2.

Carattere ricevuto	Funzione svolta
A	risposta con tutti i parametri dei sensori e dello stato di funzionamento
E	risposta con i parametri PID attuali
W+(PID parameters)	ricezione parametri PID settati dall'utente
S	salvataggio dei parametri PID attuali in EEPROM
Z	set offset sensori
D+dataXY	ricezione valori per comando movimentazione
C	reset errori ed encoder

Tabella 7 - Segnalazioni di allarmi.

Allarme	Stato
GO_Standby	avvisa di robot armato/standby
VBat_under	tensione di alimentazione troppo bassa
VBat_over	tensione di alimentazione troppo alta
IBat_over	corrente assorbita supera il limite impostato
Steer_error	errore lettura potenziometro dello sterzo (solo per cloni di SegWay)
Acc_error	errore lettura accelerometro
Gyro_error	errore lettura giroscopio
Angle_error	la misura dell'angolo eccede i limiti ammessi
EEPROM_error	si è verificato un errore nella lettura dei dati in EEPROM*
Speed_error	la velocità delle ruote eccede il valore massimo

* alla prima messa in servizio non vi sono dati in memoria quindi si verifica l'errore.



Elenco Componenti:

R1 ÷ R4, R7: 4,7 ohm (0805)
 R5, R6: 1 kohm (0805)
 R8, R9: 2,2 kohm (0805)
 R10: 47 kohm (0805)
 R11, R13 ÷ R16, R19: 4,7 ohm (0805)
 R12: 33 kohm (0805)
 R17, R18: 1 kohm (0805)
 R20: 33 kohm (0805)
 R21, R26: 10 kohm (0805)
 R22 ÷ R25: 1 kohm (0805)
 R27, R28: 1 kohm (0805)
 C1, C2: 10 μ F 35 VL elettrolitico (B)
 C3 ÷ C5, C10: 100 nF ceramico (0805)
 C6, C9: 1000 μ F 35 VL elettrolitico

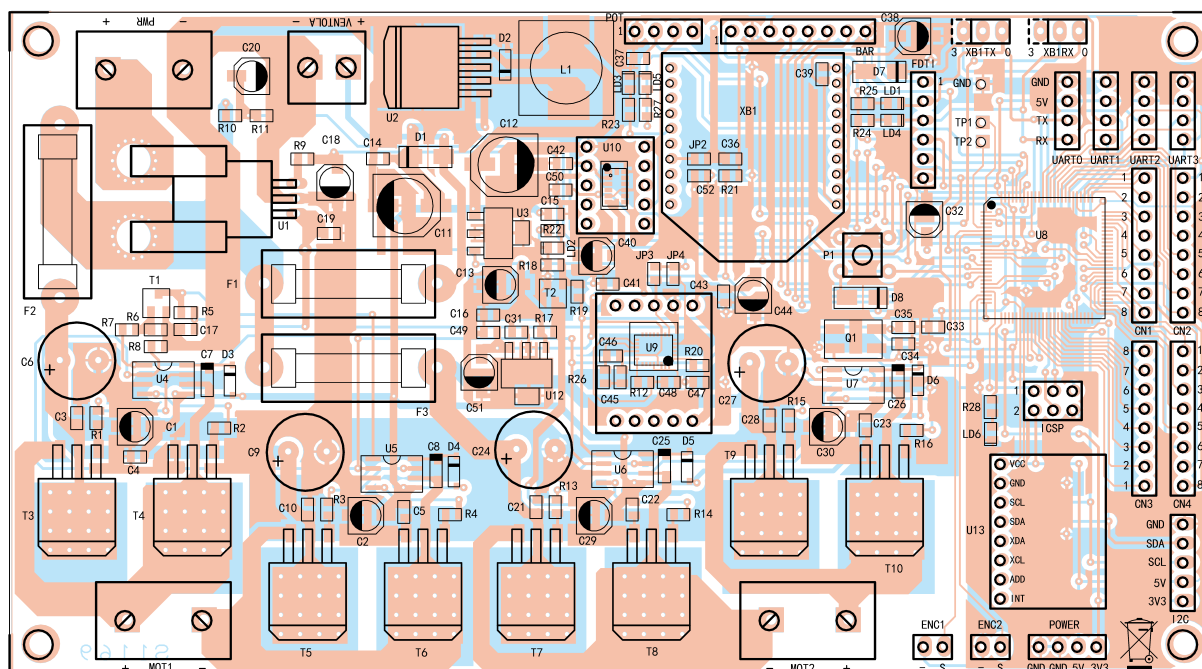
C7, C8: 10 μ F 25 VL tantalio (A)
 C11: 100 μ F 35 VL elettrolitico (F)
 C12: 220 μ F 35 VL elettrolitico (F)
 C13: 10 μ F 35 VL elettrolitico (B)
 C14 ÷ C17: 100 nF ceramico (0805)
 C18: 10 μ F 35 VL elettrolitico (B)
 C19: 100 nF ceramico (0805)
 C20: 10 μ F 35 VL elettrolitico (B)
 C21 ÷ C23: 100 nF ceramico (0805)
 C24, C27: 1000 μ F 35 VL elettrolitico
 C25, C26: 10 μ F 25 VL tantalio (A)
 C28, C31: 100 nF ceramico (0805)
 C29, C30: 10 μ F 35 VL elettrolitico (B)
 C32: 10 μ F 35 VL elettrolitico (B)

C33: 100 nF ceramico (0805)
 C34, C35: 10 pF ceramico (0805)
 C36, C39, C41, C43: 100 nF ceramico (0805)
 C37: 220 nF ceramico (0805)
 C38, C40: 10 μ F 35 VL elettrolitico (B)
 C40: 10 μ F 35 VL elettrolitico (B)
 C42: 3,3 nF ceramico (0805)
 C44: 10 μ F 35 VL elettrolitico (B)
 C45: 470 nF ceramico (0805)
 C46: 10 nF ceramico (0805)
 C47 ÷ C49: 100 nF ceramico (0805)
 C50: 3,3 nF ceramico (0805)
 C51: 10 μ F 35 VL elettrolitico (B)
 C52: 100 nF ceramico (0805)

ed impostate i parametri ID=0 e MY=0, quindi leggete i valori SH e SL. A questo punto scambiate i due moduli, trascrivete nuovamente i valori SH e SL ed impostate sui campi DH e DL i valori letti sul precedente modulo: DH=SH e DL=SL. A questo punto scambiate nuovamente i due moduli ed inserite nel campo DH e DL del primo modulo i valori SH e SL del secondo modulo. In questo modo l'indirizzo di destinazione (D) del primo modulo coinciderà con l'indirizzo fisico del secondo

modulo (S) e viceversa. A questo punto i due moduli sono pronti a comunicare. Non dimenticate di posizionare i due jumper XB1TX e XB1RX verso il "3" per mettere in comunicazione il modulo con la Seriale3 dell'Atmel adibita alla telemetria. Installate il secondo modulo su Openwheels2 e lasciate il primo connesso al PC, avviate il software OpenWheelsGUI_V20 e selezionate la porta seriale relativa al dongle XBee: partirà immediatamente la lettura dei parametri. È possibile visualizzare tutti i

parametri di funzionamento compreso un grafico in tempo reale dell'angolo di inclinazione, della potenza sui motori e della corrente assorbita. Una riga di testo in basso riporta lo stato di funzionamento ed eventuali messaggi di errore. Esiste anche la possibilità di utilizzare la telemetria via Bluetooth: in questo caso occorre attrezzarci di conseguenza dotando la scheda Openwheels 2 con un modulo BT. Per evitare di dover mettere mano allo sketch abbiamo utilizzato



D1: GF1M
D2÷D6: BAT42W
D7, D8: GF1M
T1: BC817
T2: BC817
T3÷T10: AU1RF2804S
Q1: Quarzo 16 MHz
U1: ACS758LCB-050U-PFF
U2: LM2576HVS-12/NOPB
U3: LM2937IMP-5.0/NOPB
U4÷U7: IRS2184S
U8: ATMEGA2560
U9: LPY503AL

U10: MMA7361
U11: XBEE1MWCHIP
U12: TC1262-3.3VDB
U13: MPU6050 (GY-521)
L1: Induttanza 470 μ H 850 mA
F1: Fusibile 1A
F2, F3: Fusibile 10A
LD1, LD2: LED verde (0805)
LD3÷LD5: LED rosso (0805)
LD6: LED giallo (0805)
P1: Microswitch

Varie:
- Porta fusibile da CS (3 pz.)

- Morsettiera 2 poli passo 5 mm
- Morsettiera 2 poli passo 10 mm (3 pz.)
- Strip maschio 2 poli (2 pz.)
- Strip maschio 3 poli (4 pz.)
- Strip maschio 4 poli (6 pz.)
- Strip maschio 5 poli (1 pz.)
- Strip maschio 6 poli (1 pz.)
- Strip maschio 8 poli (5 pz.)
- Strip femmina 5 poli (4 pz.)
- Strip femmina 8 poli (1 pz.)
- Strip femmina 10 poli passo 2 mm (2 pz.)
- Jumper (2 pz.)
- Circuito stampato S1169

un modulo Bluetooth che una volta configurato non richiedesse ulteriori istruzioni per il funzionamento, come il modulo DG010 della Dagurobot, che va connesso sempre alla Seriale3 tramite il connettore UART3 (Fig. 10). Questo modulo viene venduto preconfigurato per il funzionamento con protocollo SPP (Serial Port Profile) alla velocità di 9.600 bps; l'SPP permette al modulo di trasferire serialmente dei dati esattamente come farebbe un cavo e analogamente al funzionamento

dei moduli XBee in transparent mode. Per la nostra applicazione è necessario impostare una velocità di comunicazione a 115 kbps e il nome della periferica in "PRobot"; allo scopo bisogna interfacciare il modulo al PC tramite il convertitore USB/Seriale. Usate un software che possa inviare stringhe tramite la seriale come CoolTermWin oppure lo stesso Serial-Monitor di Arduino, impostando la velocità di comunicazione a 9.600 bps. Inviare quindi la stringa AT senza aggiungere alcun fine

riga o ritorno a capo: se il modulo risponde con OK significa che la comunicazione è stabilita. Modificate il nome del dispositivo con il comando AT+NAMEPRobot e la velocità di comunicazione con il comando AT+BAUD8. A questo punto il modulo è configurato e pronto a funzionare nella modalità richiesta. Anche il PC dovrà disporre della comunicazione Bluetooth; applicando il dongle è possibile abilitare alla comunicazione Bluetooth qualsiasi PC. Se riscontrate problemi con i driver

Guarda OpenWheels in azione su YouTube!



Vuoi vedere il nostro balancing robot in azione? Nulla di più semplice: collegati alla pagina

<https://youtu.be/FagvW4DdBP1>. Dal nostro canale YouTube <https://www.youtube.com/user/ElettronicaIN/videos> potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato come il nostro segway opensource OpenWheels.



di Windows, potete installare il software Bluesoleil della IVT Corporation. Non appena alimenterete il modulo il LED rosso si accenderà a luce fissa, mentre il LED blu lampeggerà; non appena vi sarete connessi al modulo il LED blu rimarrà acceso a luce fissa, ad indicare l'abilitazione alla transmis-

sione. Per utilizzare la telemetria per PC è sufficiente specificare nel software la porta seriale assegnata alla comunicazione SPP del dongle BT.

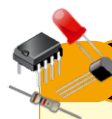
MESSA IN SERVIZIO

Con il software di telemetria avviato, è opportuno impostare tutti i parametri del controllo PID a zero e procedere alla determinazione dell'offset; posizionate il robot esattamente in equilibrio in modo stabile e senza vibrazioni e premete il pulsante OFFSET: la taratura è automatica e richiede due secondi. Per evitarvi di ripetere l'operazione ad ogni accensione, i parametri vengono salvati nella memoria permanente del micro. Ora portate il robot dallo stato di stand-by a quello di Go e resettate eventuali errori; non vi resta che agire sui parametri del PID per ottenere la stabilizzazione. Non esiste una procedura certa e sicura per determinare immediatamente i parametri corretti; il più delle volte serve molta pazienza e diverse prove per arrivare ad una stabilizzazione soddisfacente. La procedura ideale prevede che, partendo con tutti i parametri nulli, si alzi il solo parametro KP finché il robot non cominci ad oscillare; successivamente si abbassa il valore di KP in modo che il robot non oscilli, quindi si alza il valore di KV. Già questi due parametri dovrebbero garantire la stabilizzazione e solo in un secondo momento si alza il valore di KO per assicurare che il robot mantenga la posizione. In un secondo momento potete provare ad alzare anche il valore di KD per verificare se la stabilizzazione migliora.

Con il robot da noi proposto è stato sufficiente impostare: KP=160, KO=5 e KV=12 (i rimanenti parametri sono a zero).

CONTROLLO REMOTO

Non poteva mancare la possibilità di controllare a distanza la movimentazione del robot: la filosofia di base è ingannare il robot facendogli credere di essere perfettamente bilanciato, anche se in realtà è leggermente (parliamo di decimi di grado) inclinato, pertanto tendente a cadere. È sufficiente aggiungere o togliere un determinato valore alla variabile Pvel affinché il robot avanzi o indietreggi, mentre, per farlo sterzare, dobbiamo modificare il valore del PWM dei due motori incrementando l'uno e diminuendo l'altro, in modo che la somma rimanga costante. Sommando i due effetti si può far virare il robot e quindi comandarlo in



per il MATERIALE

I prodotti presentati in questo progetto possono essere acquistati presso Futura Elettronica. La board Open Wheels 2.0 (cod. FT1169M) costa 145,00 Euro, la batteria Lipo 1800 mAh/11,1V (cod. LIPO1800) è disponibile a 29,00 Euro. Il modulo Xbee 1 mW con antenna integrata (cod. XBEE1MWTANT) costa 36,00 Euro, quello da 2 mW completo di antenna (cod. XBEE2MW) 36,00 Euro, mentre l'Explorer Dongle V2 (cod. XBEE2DONGLE2) è in vendita a 33,50 Euro. Il modulo Bluetooth controller DAGU (cod. DG010) è disponibile a 35,00 Euro, il convertitore FTDI USB-Seriale 3,3V/5V (cod. FTDI5V) costa 19,00 Euro, il motoriduttore con encoder e ruote (cod. BALANCINGBOT) è in vendita a 74,00 Euro.

Il materiale va richiesto a:

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

Tel: 0331-799775 • Fax: 0331-792287
<http://www.futurashop.it>

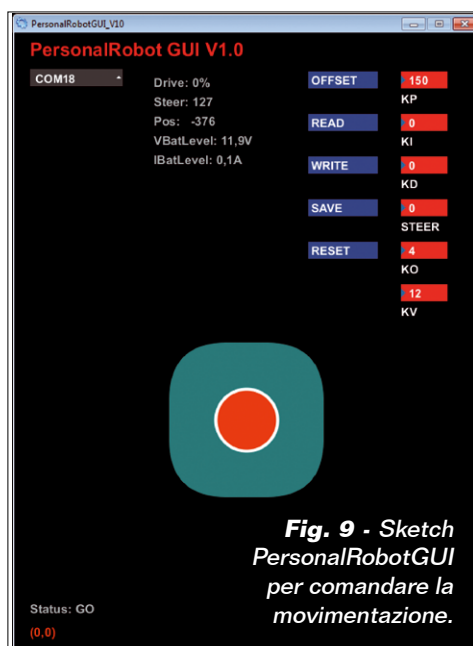


Fig. 9 - Sketch PersonalRobotGUI per comandare la movimentazione.

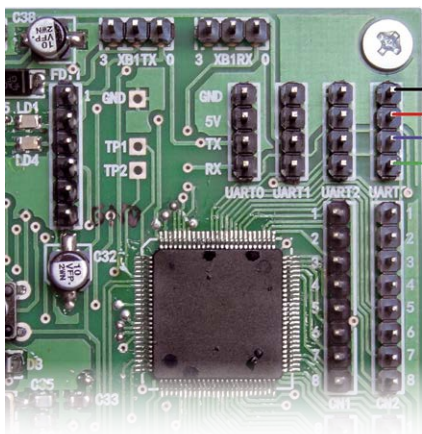


Fig. 10
Connessione modulo Bluetooth.

bilità dell'IDE di Processing di gestire più piattaforme di sviluppo (potete approfondire l'argomento con il corso pubblicato attualmente nelle nostre pagine). L'app (Fig. 9) non è particolarmente complessa o esteticamente sofisticata, ma fa il suo dovere, ricevendo i dati di telemetria e inviando i dati del joystick virtuale, implementato sul display e facilmente azionabile con il pollice. L'App è stata scritta per uno smartphone Samsung S4 con display FHD e display AMOLED, per cui abbiamo preferito uno sfondo di colore molto scuro per evitare un eccessivo consumo di corrente durante il funzionamento. Per utilizzare lo smartphone, come telecomando dovete installare l'App di nome *PRobot*. apk che troverete tra i file del progetto. Non essendo stata prelevata dallo store ufficiale, l'installazione potrà avvenire solo se è stata attivata la modalità di installazione da sorgenti sconosciute, di solito disponibile nella sezione sicurezza delle impostazioni.

Per prima cosa dovete associare la periferica Bluetooth accedendo alla sezione BT nelle impostazioni del telefono (la password è 1234); a questo punto potete avviare l'App, che è configurata per

qualsiasi direzione con qualsiasi velocità (nei limiti consentiti dalla meccanica e della fisica). Come già accennato i parametri KO e Kdm devono essere posti a zero altrimenti il robot interpreterà i comandi di movimento come dei disturbi e cercherà di compensarli. Anche in questo caso abbiamo scritto un apposito software di nome *PersonalRobotGUI_V10* che permette di mantenere sotto controllo i principali parametri e nel contempo di comandare il movimento tramite un semplice joystick virtuale, da azionare con il mouse.

Vista la criticità del robot, i comandi dovranno essere gradualmente e fluidi e di certo non ci si può aspettare di avere un controllo immediato come quello di un robot a quattro ruote.

CONTROLLO DA ANDROID

Abbiamo previsto la possibilità di comandare il robot tramite un'app Android, scritta grazie alla possi-



Fig. 11
Configurazione modulo Bluetooth.

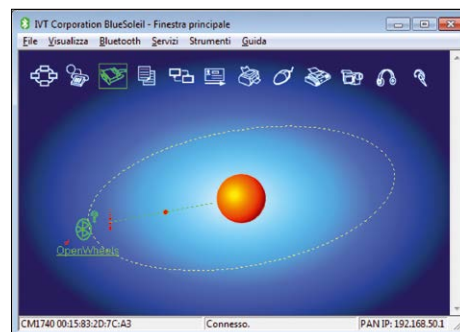


Fig. 12 - Software BlueSoleil per il Bluetooth.

ricercare la periferica PRobot tra quelle associate e collegarvi a essa (Fig. 14). Non serve connettersi ad alcun altro dispositivo. Instaurata la connessione, sul display saranno visualizzati i dati di telemetria e potrete azionare il joystick per comandare il robot.

Bene, con questo abbiamo detto tutto; ora sta a voi dare forma al progetto, variando la meccanica o il firmware a piacimento.

Fig. 13 - Connessione al modulo Bluetooth da PC.

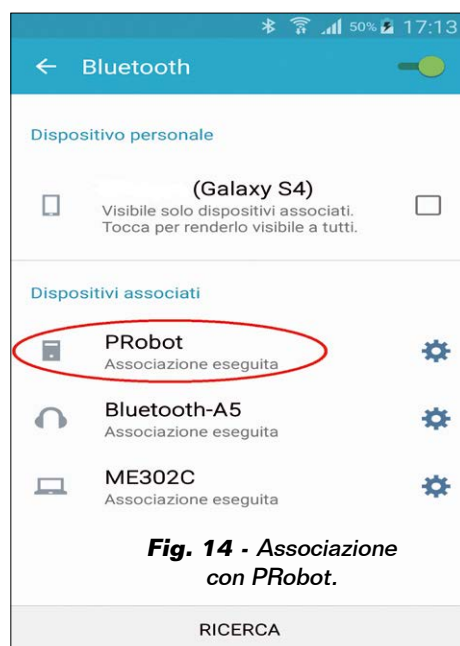
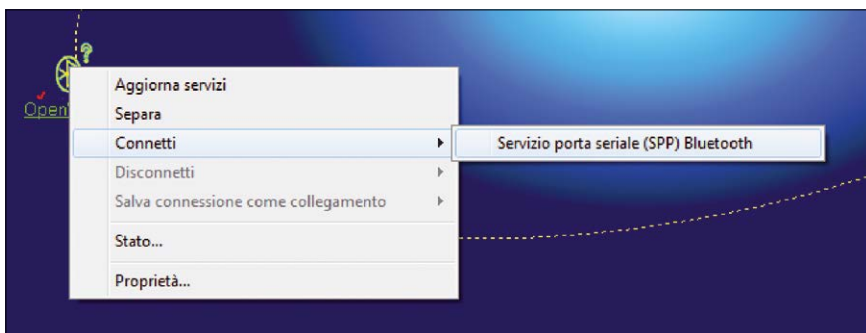
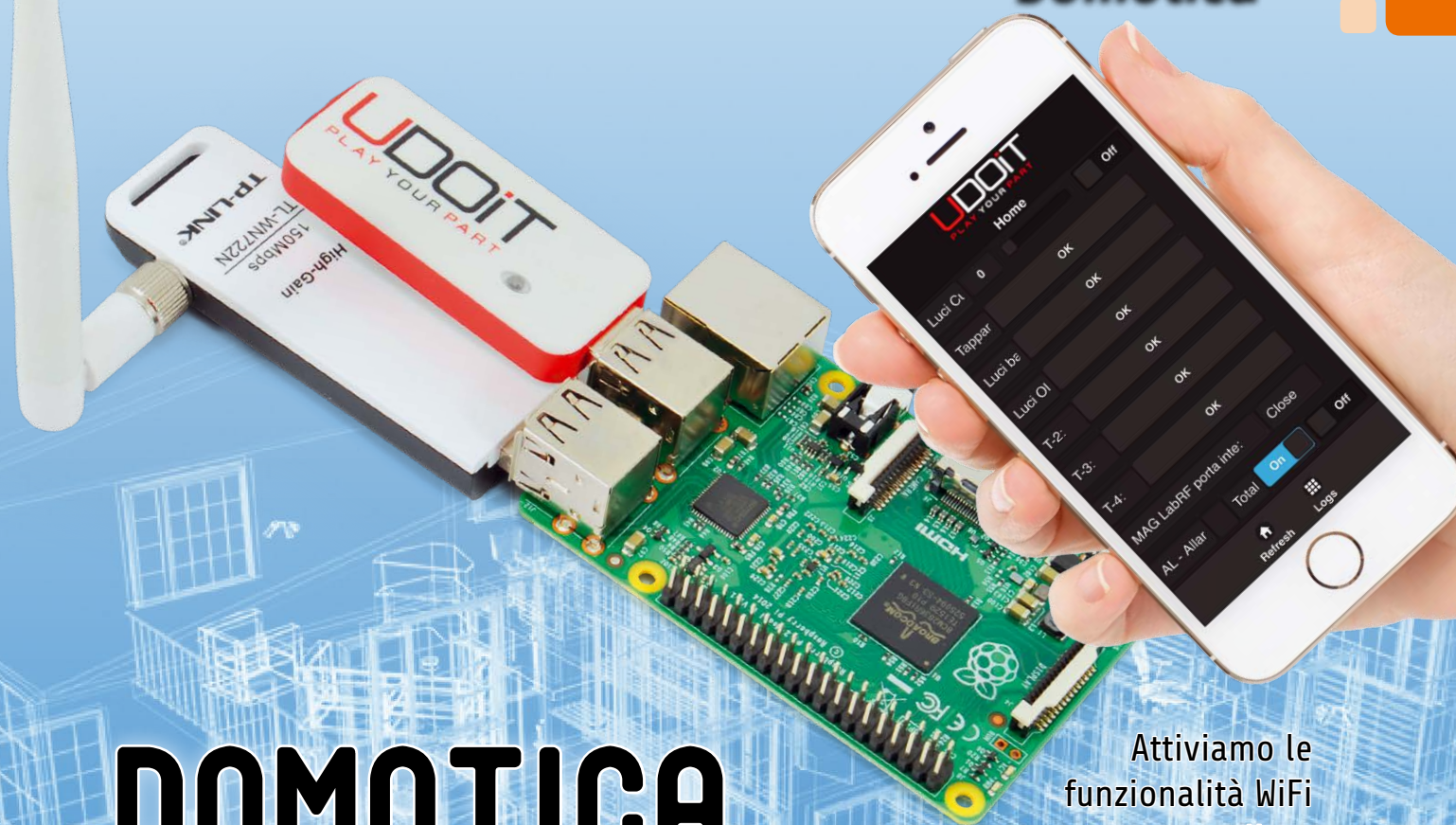


Fig. 14 - Associazione con PRobot.



DOMOTICA OPEN SOURCE con i moduli UDOiT

Attiviamo le funzionalità WiFi per controllare il sistema UDOiT senza cavi di rete, sia come client che come Access Point.

di MARCO MAGAGNIN

Non sempre è possibile raggiungere il luogo dove vogliamo posizionare il nostro sistema di controllo domestico con un cavo di rete collegato al router della rete di casa. E non sempre si desidera inserire il sistema di controllo nella rete domestica ma si preferisce gestire una rete differente, in wifi. Per esempio se vogliamo posizionare il nostro sistema di controllo in un luogo non dotato di connessione internet, come un box auto, una serra od un giardino. In tutti i casi è necessario attivare le funzionalità di rete wireless su Raspberry Pi, che funge da sistema di supporto all'applicazione di controllo domestico. In entrambe le configurazioni è necessario dotarsi di un "dongle" wifi, da scegliersi in base alle esigenze. Nel primo caso si dovrà configurare

l'interfaccia di rete wifi a funzionare come Hot-Spot, od Access Point, nel secondo caso si dovrà configurare l'interfaccia come client wifi della rete di casa ma con assegnato un indirizzo statico per poter raggiungere facilmente l'interfaccia web. Ovviamente il contenuto di questo articolo vale in generale, per qualunque utilizzo si voglia fare di Raspberry Pi, e non solo in associazione all'applicazione UDOiT. Cercheremo di toccare tutti gli argomenti che possono aiutare nelle configurazioni citate, in quanto la configurazione di una connessione wifi è soggetta a molteplici variabili, condizionate dal tipo di apparecchiature alle quali ci si vuole collegare, dal tipo di autenticazione che si vuole implementare e dal tipo di dispositivo che si utilizza per la connessione.

ACCESS POINT

In Fig. 1 è visibile uno schema di riferimento dell'architettura di rete wireless nella quale Raspberry Pi appare come Access Point e gli altri dispositivi come client associati allo stesso. La configurazione dell'interfaccia wifi di Raspberry Pi in modalità Access Point richiede l'installazione di pacchetti software dedicati e delle loro dipendenze, nella fattispecie un pacchetto che fornisca il servizio di Access Point e di autenticazione/sicurezza ed un pacchetto per la gestione del server DHCP, quest'ultimo necessario per fornire gli indirizzi IP dinamici ai client che si collegano all'applicazione di gestione. Questi pacchetti, nella fattispecie "hostapd" e "isc-dhcp-server" sono già installati nell'immagine del sistema operativo scaricabile dal sito UDOiT. Sono solo da risvegliare, e tra breve vedremo come. Il punto "delicato" per la messa in funzione dell'Access Point è la scelta del dongle wifi. I dongle, come tutte le interfacce di rete wifi, USB e non, non sono tutti uguali, come visibile in Fig. 2. A parte forma, dimensioni e tipo di interfaccia le loro differenze principali risiedono nel chip adottato e nelle relative funzionalità rese disponibili. La diversità di queste caratteristiche è la causa principale dei mal di testa provocati a chi si avventura nel campo della configurazione delle reti wireless tentando di associare dispositivi differenti come smartphone di case diverse, tablet, PC, televisori, lavatrici ecc. Pressoché tutti i chip hanno la possibilità di essere utilizzati per collegarsi come client ad una rete wifi esistente. Come è noto la modalità client non è l'unica possibile per accedere ad una rete. Sono possibili anche la cosiddetta modalità AD-HOC, nella quale due sistemi possono essere

Fig. 2



collegati "alla pari" senza distinzione tra Access Point e client. Una modalità monitor, che permette di "ascoltare" cosa succede nell'etere circostante, senza collegarsi fisicamente ad alcuna rete e, tra le ulteriori altre, anche la modalità AP o Access Point, per l'appunto, che permette di configurare la periferica ed il relativo sistema ospitante come un Access Point in grado di gestire l'autenticazione, e la sicurezza di accesso ai client che si vogliono collegare. Approfittiamo qui per approfondire un altro argomento. Anche le tipologie e le caratteristiche dei sistemi di autenticazione non sono implementati tutti allo stesso modo sui diversi chip. A parte l'età, in quanto sui chip "vecchi" non sono implementati i sistemi di sicurezza più recenti, spesso ci si trova a confrontarsi anche con la lunghezza delle password. Per esempio non si riescono ad associare ad un Access Point con una chiave di rete di 64 caratteri alcuni tipi di dispositivi (mai capitato?) che permettono l'utilizzo solo di password più corte (per esempio 32 caratteri), altri dispositivi non accettano l'associazione tramite tasto (o funzione) WPS, e così di seguito. Tutta questa premessa per consigliarvi di scegliere con cura il dongle wifi da utilizzare per trasformare Raspberry Pi in un Access Point. Per inciso, la sicurezza del sistema di protezione della rete wifi non è di secondaria importanza, in quanto una rete wifi è esposta a chiunque sia in grado di intercettare il segnale wifi. Una protezione WPA2 ed una chiave lunga e complessa mette al riparo da "orecchie" indiscrete, ed in genere ve ne sono. Prima di tutto deve essere un dongle compatibile con i driver presenti nel sistema operativo Raspian, ricordatevi di aggiornare sempre il sistema con i comandi:

```
apt-get update
apt-get upgrade
apt-get dist-upgrade
```

In questa scelta può essere utile consultare la lista delle periferiche per le quali è stata verificata la compatibilità pubblicata all'indirizzo (Fig. 3):

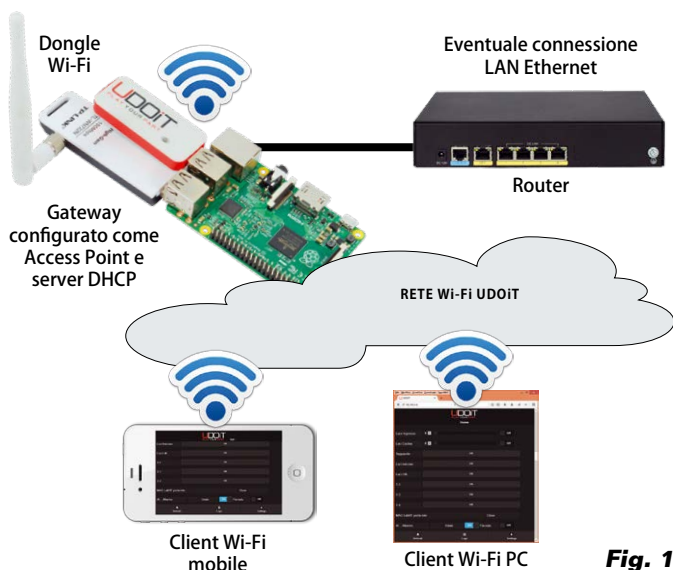


Fig. 1



Fig. 3

http://elinux.org/RPi_USB_Wi-Fi_Adapters

La lista non riporta informazioni in modo sistematico sui dongle che offrono funzionalità come Access Point ma è possibile identificare quelli sicuramente "certificati". Un metodo è quello di utilizzare la funzione del browser "Trova" cercando "AP". Il nostro consiglio è di scegliere tra uno di questi. Noi abbiamo utilizzato un TL-WN722N della TP-LINK. Di seguito vi indichiamo il metodo per verificare se il wifi dongle funziona anche come Access Point. Potete utilizzare questo metodo per verificare se potete utilizzare qualche dongle wifi che avete già in casa. Per prima cosa avete bisogno del gateway Raspberry Pi configurato con l'immagine che contiene l'applicazione, con il dongle master di rete UDOiT ed il dongle wifi inseriti nelle rispettive prese USB. Per questa operazione Raspberry Pi deve essere collegata alla rete via cavo, ricordandovi di configurare un IP statico come descritto nel numero 200 della rivista. Per verificare che il dongle wifi sia stato riconosciuto correttamente dal sistema operativo Raspbian digitate il comando:

```
cat /proc/net/dev
```

oppure

```
ifconfig
```

In entrambi i casi vi dovrebbe apparire un'interfaccia con nome "wlan0". In Raspbian i device di rete non seguono le regole degli altri device come SDCard, dischi, ecc. Per il network il file descrittore è in /etc/net/tun ma le interfacce disponibili sono reperibili in /proc/net/dev. Se trovate l'interfaccia

di rete wlan0 è tutto OK (Fig. 4). Altrimenti bisogna capire il perché dell'intoppo. Per lo più si tratta di un dongle con un chip per il quale non è presente il driver. Potete indagare l'accaduto con i comandi (Fig. 5):

```
lsusb
```

```
e
```

```
dmesg
```

In genere l'unica operazione possibile per non impazzire è utilizzare un dongle con un chip per il quale è disponibile il driver. Per approfondire l'argomento potete fare riferimento al libro "Raspberry Pi – Il mio primo Linux embedded". A questo punto date il comando (Fig. 6):

```
iw list
```

Nella sezione "Supported Ciphers" sono elencate tutte le forme di autenticazione supportate dal dongle. Nella sezione "Supported interface modes" sono elencati i "modi" supportati dal dongle, tra i quali troviamo "AP". Nelle altre sezioni troviamo le frequenze e le velocità di comunicazione supportate. Il primo passo è fatto. Ora è necessario configurare il servizio "Hostapd" ed il DHCP server. Cominciamo dal server "Hostapd". Il suo file di configurazione si trova in:

```
/etc/hostapd/hostapd.conf
```

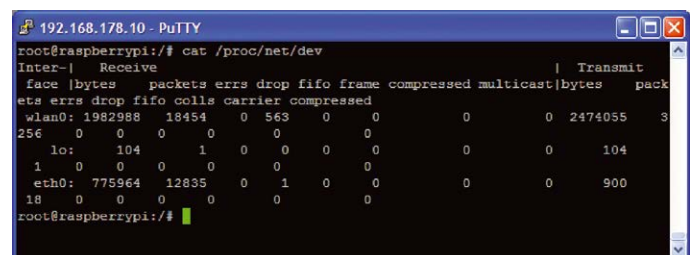


Fig. 4

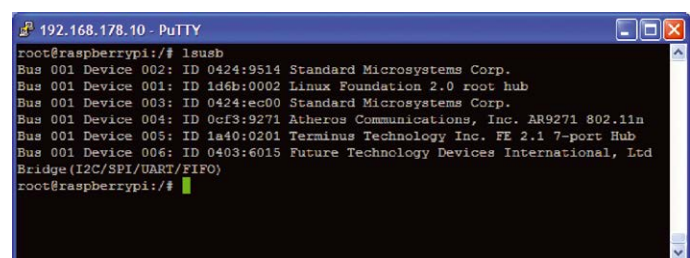


Fig. 5

```

192.168.178.10 - PuTTY
root@raspberrypi:/# iw list
Wiphy phy0
  Band 1:
    Capabilities: 0x116e
                  HT20/HT40
                  SM Power Save disabled
                  RX HT20 SGI
                  RX HT40 SGI
                  RX STBC 1-stream
                  Max AMSDU length: 3839 bytes
                  DSSS/CKK HT40
                  Maximum RX AMPDU length 65535 bytes (exponent: 0
x003)
                  Minimum RX AMPDU time spacing: 8 usec (0x06)
                  HT TX/RX MCS rate indexes supported: 0-7
    Frequencies:
      * 2412 MHz [1] (20.0 dBm)
      * 2417 MHz [2] (20.0 dBm)
      * 2422 MHz [3] (20.0 dBm)
      * 2427 MHz [4] (20.0 dBm)
      * 2432 MHz [5] (20.0 dBm)
      * 2437 MHz [6] (20.0 dBm)
      * 2442 MHz [7] (20.0 dBm)
      * 2447 MHz [8] (20.0 dBm)
      * 2452 MHz [9] (20.0 dBm)
      * 2457 MHz [10] (20.0 dBm)
      * 2462 MHz [11] (20.0 dBm)
      * 2467 MHz [12] (20.0 dBm)
      * 2472 MHz [13] (20.0 dBm)
      * 2484 MHz [14] (disabled)
    Bitrates (non-HT):
      * 1.0 Mbps
      * 2.0 Mbps (short preamble supported)
      * 5.5 Mbps (short preamble supported)
      * 11.0 Mbps (short preamble supported)
      * 6.0 Mbps
      * 9.0 Mbps
      * 12.0 Mbps
      * 18.0 Mbps
      * 24.0 Mbps
      * 36.0 Mbps
      * 48.0 Mbps
      * 54.0 Mbps
    max # scan SSIDs: 4
    max scan IEs length: 2257 bytes
    Coverage class: 0 (up to 0m)
    Supported Ciphers:
      * WEP40 (00-0f-ac:1)
      * WEP104 (00-0f-ac:5)
      * TKIP (00-0f-ac:2)
      * CCMP (00-0f-ac:4)
      * 00-0f-ac:10
      * 00-0f-ac:8
      * 00-0f-ac:9
      * CMAC (00-0f-ac:6)
      * 00-0f-ac:13
      * 00-0f-ac:11
      * 00-0f-ac:12
    Available Antennas: TX 0x1 RX 0x1
    Configured Antennas: TX 0x1 RX 0x1
    Supported interface modes:
      * IBSS
      * managed
      * AP
      * AP/VLAN
      * monitor
      * mesh point
      * P2P-client
      * P2P-GO
    software interface modes (can always be added):
      * AP/VLAN
      * monitor
    valid interface combinations:
      * #{ managed, P2P-client } <= 2, #{ AP, mesh po
int, P2P-GO } <= 2,
        total <= 2, #channels <= 1
    Device supports TX status socket option.
    Device supports HT-IBSS.
root@raspberrypi:/#

```

Fig. 6

Per aprirlo utilizziamo lo strumento WinSCP e di seguito vediamo il suo contenuto.

```

interface=wlan0
driver=nl80211
#driver=rtl871xdrv
#driver=ath9k_htc
ssid=UDOiT
hw_mode=g
channel=6
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=raspberry
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP

```

Nell'immagine scaricabile dal sito UDOiT il server è già configurato. Potete modificare il nome dell'Access Point visibile all'esterno modificando il campo "ssid". Il driver "nl80211" è il vecchio driver predefinito per interfacciare la maggior parte dei

```

/etc/network/interfaces - RaspberryPi - Editor - Wi...
auto lo

iface lo inet loopback

## Configurazione connessione Ethernet
## con indirizzo IP statico
iface eth0 inet dhcp
iface eth0 inet static
    address 192.168.1.10
    netmask 255.255.255.0
    network 192.168.1.0
    broadcast 192.168.1.255
# commentare se si utilizza anche dongle wifi
# client altrimenti non funziona checkwifi.sh
gateway 192.168.1.1

# allow-hotplug wlan0

## Configurazione come Access Point
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0

## Configurazione come client wifi
# iface wlan0 inet static
#   address 192.168.1.12
#   netmask 255.255.255.0
#   gateway 192.168.1.1
#   wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet dhcp

```

Fig. 7


```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multi
# user session.  Make sure that the script will "exit 0" on succ
# value on error.
#
# In order to enable or disable this script just
# bits.
#
# By default this script does nothing.
#
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi

sudo service hostapd start
sleep 2
sudo service isc-dhcp-server start
sleep 2

bash /home/RF/RF_init start

exit 0
```

dongle usb e farli funzionare come Access Point. Se non dovesse funzionare con il vostro dongle, provate con gli altri due, uno alla volta, decommentando quello che si desidera utilizzare e commentando gli altri. Se nessuno dovesse funzionare, non resta che ricercare in internet, oppure cambiare dongle. L'entrata "wpa_key_mgmt=WPA-PSK" configura il tipo di protezione da applicare alla connessione, mentre nel campo "wpa_passphrase=raspberry" potete configurare la vostra password di protezione. Se eseguite qualche modifica, ricordatevi di salvare il file modificato. Ora tocca al server DHCP. Il file di configurazione lo troviamo in "/etc/dhcp/dhcpd.conf" del quale di seguito è visibile la sezione che ci interessa.

```
subnet 192.168.42.0 netmask 255.255.255.0 {
    range 192.168.42.100 192.168.42.150;
    option broadcast-address 192.168.42.255;
    option routers 192.168.42.1;
    default-lease-time 600;
    max-lease-time 7200;
    option domain-name "local";
    option domain-name-servers 8.8.8.8, 8.8.4.4;
}
```

In questa configurazione il parametro subnet indica il primo indirizzo della classe che si vuole assegnare al DHCP, seguito dalla maschera di rete. Il parametro "range" indica quali indirizzi vogliamo siano assegnati automaticamente a chi si collega chiedendo il servizio DHCP per farsi assegnare un indirizzo. Chiaramente non può essere assegnato ai dispositivi

Fig. 8

alcun indirizzo statico che "caschi" nell'intervallo configurato. Il parametro "option-routers" riporta l'indirizzo assegnato a Raspberry Pi, per intenderci quello che si utilizza per collegarsi a Raspberry Pi dal browser web. Nel parametro "option domain-name-servers" si configurano gli indirizzi dei server DNS (Domain Name Server) che permettono la risoluzione dei nomi trasformandoli nei corrispondenti indirizzi IP, questo nel caso si voglia utilizzare Raspberry Pi come Access Point ma che funzioni anche come router verso la rete internet. La configurazione è un po' complessa, per il momento sono stati configurati gli indirizzi dei server DNS messi a disposizione da Google. Anche in questo caso, se eseguite delle modifiche, salvate il file prima di chiuderlo. Ora dobbiamo assegnare l'indirizzo IP statico che abbiamo scelto per Raspberry Pi nel file di configurazione delle interfacce di rete, per chi ci segue da un po' su questa rivista, il mitico file /etc/network/interfaces, di seguito potete vedere riprodotta la sezione che descrive l'interfaccia Wlan0 (Fig. 7). Per gli utenti Windows, usate sempre WinSCP per navigare nel file system ed editare il file:

```
#allow-hotplug wlan0
iface wlan0 inet static
    address 192.168.42.1
    netmask 255.255.255.0
```

In "address" configurate lo stesso indirizzo IP statico che avete configurato nel file di configurazione del server DHCP. Salvate e chiudete. Avete quasi finito. Non resta che fare in modo che i due server vengano lanciati al boot di Raspberry Pi. La scelta

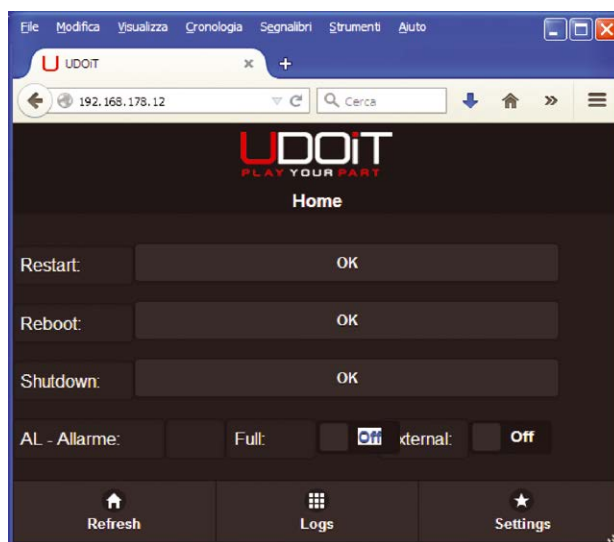


Fig. 9

fatta nel sistema è di eseguire lo start dei server “hostapd” e “isc-dhcp-server” al termine del processo di boot, poco prima del lancio dei server della applicazione. L’ultimo script che viene lanciato durante il processo di boot è /etc/rc.local, nel quale si possono inserire i comandi necessari al lancio dei due server. Inserite i comandi prima di “bash /home/RF/RF_init start” che è il comando di esecuzione dei server dell’applicazione UDOiT (Fig. 8).

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
```

```
sudo service hostapd start
sleep 2
sudo service isc-dhcp-server start
sleep 2
```

```
bash /home/RF/RF_init start
```

```
exit 0
```

I due comandi sleep servono a dare il tempo ai server di partire e configurarsi con comodo. Salvate anche questo file ed eseguite un

reboot

del sistema. Attendete il tempo necessario al sistema per ripartire ed attivare tutti i suoi componenti. Poi da un browser su un tablet, uno smartphone

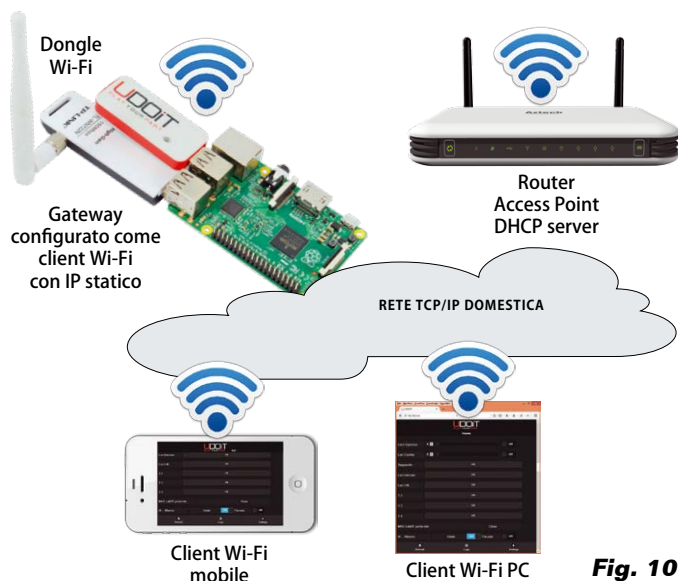


Fig. 10

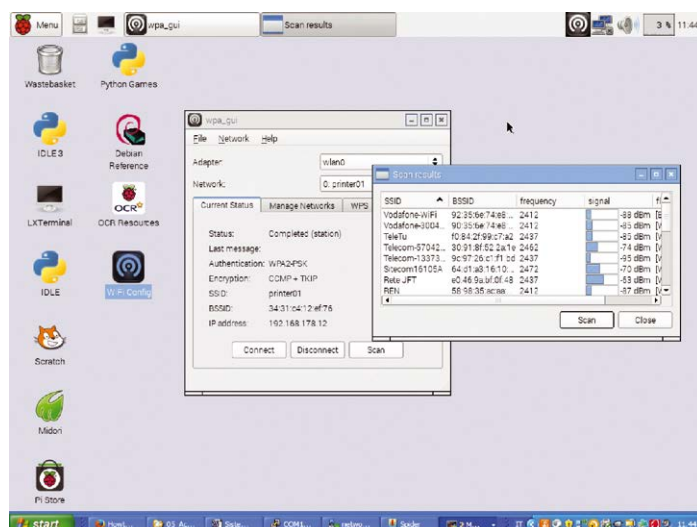


Fig. 11

od un qualunque dispositivo dotato di connettività wifi, aprite il configuratore della rete wifi, cercate l’SSID che avete configurato per Raspberry Pi, inserite la password, aprite il browser ed inserite l’indirizzo IP che avete configurato. Dovreste trovarvi la pagina di comando del sistema di automazione domestica (Fig. 9).

CONNESSIONE WIFI CLIENT

La seconda alternativa è di configurare Raspberry Pi come client della vostra rete domestica.

In questo caso dobbiamo seguire un percorso diverso, che richiede di assegnare un IP statico a Raspberry Pi (al di fuori dell’intervallo degli indirizzi distribuiti dal DHCP del vostro router) e configurare le credenziali di accesso alla vostra rete come visibile nello schema di Fig. 10. In questo caso, per raggiungere il nostro obiettivo, dobbiamo tenere presente alcune condizioni che possono richiedere una “personalizzazione” della configurazione che andiamo descrivendo.

A parte la scelta del dongle wifi, assai più vasta che nel caso della configurazione come Access Point, bisogna tenere presente il tipo di accesso che è stato configurato sul router, in particolare il tipo di protezione ed il tipo di password utilizzata. In secondo luogo è di gran lunga preferibile assegnare a Raspberry Pi un indirizzo IP statico, appartenente alla stessa classe assegnata alla rete domestica, e che non sia all’interno dell’intervallo degli indirizzi assegnati dal server DHCP. Potremmo poi dover configurare manualmente i riferimenti ai server DNS, in quanto di solito vengono configurati durante il processo di assegnazione dell’indirizzo IP dinamico da parte del DHCP, cosa che non avviene nel nostro caso. Infine, un ulteriore tema è il collegamento automatico alla rete domestica all’accensione di

PiDrive - HARD DISK per Raspberry Pi

Come è noto il supporto predefinito per il sistema operativo di Raspberry Pi è la schedina SD Card, in formato normale o micro, a seconda dei modelli. Questo va più che bene per un uso hobbistico o per applicazioni che non richiedono particolari accorgimenti in termini di sicurezza dei dati e di prestazioni. Nei nostri articoli abbiamo sempre sottolineato il fatto che le SD Card presentano dei limiti nel numero di scritture, cosa di poca importanza nell'utilizzo, per esempio in una macchina fotografica, ma che può rappresentare un serio limite in applicazioni che fanno, per esempio, un pesante utilizzo di database. Nel tempo abbiamo descritto come utilizzare un hard disk per la partizione di root, al posto della corrispondente partizione sulla SD Card. In quest'ottica Western Digital ha approntato una soluzione al problema particolarmente aderente alle caratteristiche di Raspberry Pi. La soluzione, dal nome commerciale di PiDrive prevede:

- un disco fisso da un Terabyte, con interfaccia



USB 3.0 modificata per adattarsi alle porte USB di Raspberry Pi, in modo da ottimizzare le prestazioni anche in assenza della porta SATA;

- un alimentatore dotato di doppia uscita per alimentare in contemporanea sia Raspberry Pi che l'hard disk;
- una micro SD Card, con adattatore, sulla quale ospitare la partizione di boot, obbligatoria per la partenza del sistema operativo di Raspberry Pi ed utilizzabile su tutte le versioni di Raspberry Pi.

Raspberry Pi oppure dopo un eventuale reset del router di casa. Affrontiamo questi argomenti, uno alla volta. Come prima operazione configuriamo l'accesso alla rete locale.

Il file di configurazione è:

```
/etc/wpa_supplicant/wpa_supplicant.conf
```

Nel nostro caso lo troviamo già preconfigurato per una connessione con chiave WPA2. I parametri che configurano la connessione all'Access Point sono i seguenti:

```
network={
ssid="SSID_RETE_WIFI"
psk="PASSWORD_RETE_WIFI"
proto=RSN
key_mgmt=WPA-PSK
pairwise=CCMP
auth_alg=OPEN
}
```

Per una configurazione di rete leggermente diversa elenchiamo alcune alternative ai parametri citati, per configurazioni notevolmente differenti qualche ricerca in internet può essere d'aiuto, oppure si può utilizzare il configuratore wifi

disponibile nel desktop di Raspberry Pi. Questa possibilità richiede di collegare Raspberry Pi ad un monitor HDMI (va bene anche il televisore, se dotato di un'entrata HDMI) oltre che a tastiera e mouse. Per attivare il desktop, dopo che è terminato il caricamento del sistema operativo, eseguire il login con utente "pi" e password "raspberrypi" ed in seguito eseguire il comando "startx". Cliccate sull'applicazione WiFi config e successivamente sul pulsante "scan" (Fig. 11).

Di seguito il significato di alcuni parametri:

- **proto** – RSN, come nel nostro caso indica l'utilizzo di uno schema di sicurezza WPA2; il valore WPA indica l'utilizzo di uno schema di sicurezza WPA1;
- **key_mgmt** - WPA-PSK pre-shared key è la modalità utilizzata nelle reti domestiche e nei piccoli uffici, WPA-EAP è invece utilizzata nelle organizzazioni con una gestione della sicurezza centralizzata, per esempio utilizzando un server Radius;
- **pairwise** – esprimono la caratteristica delle chiavi di autenticazione tra access point e device: CCMP nel caso di WPA2 oppure TKIP nel caso di WPA1;
- **auth_alg** – algoritmo utilizzato per l'auten-

ticazione. Per la protezione WPA e WPA2 il parametro è OPEN, altre configurazioni sono SHARED e LEAP.

Se la vostra rete domestica ha uno schema di autenticazione diverso da WPA2 potete utilizzare l'applicazione "WiFi-config", come indicato in precedenza. Per approfondire l'argomento relativo alla configurazione dei parametri di connessione agli Access Point potete consultare il documento all'indirizzo:

http://www.cs.upc.edu/lcsi/Manuales/wireless/files/wpa_supPLICANT.conf

Questo per quanto riguarda la configurazione per l'accesso al router della connessione di rete domestica. Di solito dopo la connessione al router i dispositivi sono configurati per richiedere al server DHCP del router un indirizzo IP dinamico e le relative configurazioni associate. Nel caso della applicazione in esame è invece pressoché necessario configurare Raspberry Pi con un indirizzo IP statico, in modo da poter accedere via browser all'applicazione, sempre con lo stesso URL. Per configurare un indirizzo IP statico si deve modificare il file `/etc/network/interfaces`, in particolare nella sezione `"wlan0"`, dove `wlan0` è il descrittore con il quale è identificato il dongle wifi. In caso siano attive sia la connessione via cavo che la connessione wifi sulla stessa rete domestica, commentate l'entrata `"gateway"` per l'interfaccia `"eth0"`.

```
# /etc/network/interfaces

auto lo
iface lo inet loopback

iface eth0 inet dhcp

allow-hotplug wlan0
iface wlan0 inet static
address 192.168.1.12
netmask 255.255.255.0
gateway 192.168.1.1
wpa-conf /etc/wpa_supplicant/wpa_supplicant.conf

iface default inet dhcp
```

L'ultima riga della sezione `wlan0` rimanda al file di configurazione `"wpa_supplicant.conf"` della connessione wifi per il riconoscimento della rete alla quale collegarsi e lo scambio delle credenziali

per l'autenticazione. Se con la configurazione così impostata non si riesce ad accedere all'esterno, il problema può essere la risoluzione dei nomi, per la mancata configurazione dei server DNS. Si può sopprimere aggiungendo in testa al file `/etc/resolv.conf` gli indirizzi dei server DNS resi disponibili da Google:

```
nameserver 8.8.8.8
nameserver 8.8.4.4
```

Infine un ultimo problema che potrebbe presentarsi, anche se non necessariamente. All'accensione di Raspberry Pi, se l'Access Point è attivo le configurazioni che sono state impostate sino ad ora permettono la connessione automatica all'Access Point stesso. Se però al momento dell'accensione di Raspberry Pi l'Access Point non è attivo oppure, se la connessione viene persa dopo che è stata instaurata, può accadere che non venga più riattivata automaticamente. Per sopperire a questo inconveniente è possibile realizzare una procedura di recupero automatico della connessione wifi. In caso contrario sarebbe necessario riattivare la connessione da linea di comando, oppure dallo strumento `"WiFi-config"`, oppure ancora con un bel `"reboot"`. Il `"salvagente"` che andiamo a descrivere consiste di uno script che verifica se è possibile accedere al router mediante un ping. In caso positivo lo script termina, mentre in caso negativo, disattiva l'interfaccia di rete per riattivarla poco dopo, in modo da reinnescare il collegamento automatico. Questo script deve essere eseguito a intervalli periodici, per esempio ogni cinque minuti. Per questo si utilizza lo schedatore interno cron. Cron è stato descritto nel libro `"Raspberry Pi – Il mio primo Linux embedded"` ma ne riassumiamo brevemente l'utilizzo nel seguito. Partiamo dallo script che vediamo listato di seguito. Iniziate col copiare le istruzioni in un file di nome `"checkwifi.sh"` salvandolo nella cartella `/home/RF/`, il per-

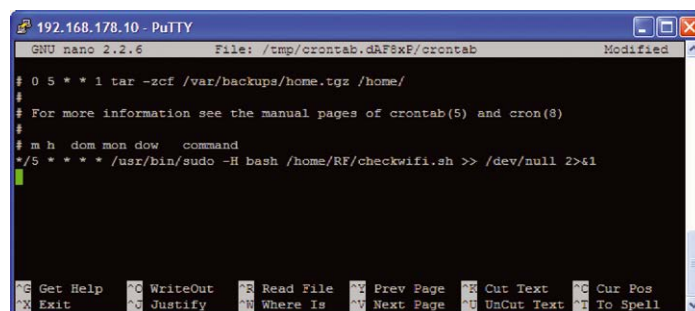


Fig. 12

Fig. 13

corso completo sarà:

`/home/RF/checkwifi.sh`

`#!/bin/bash`

`cd /home/RF`

`ping -c4 192.168.1.1 > /dev/null`

`if [ $? != 0 ]`

`then`

`echo "No network connection, restarting wlan0"`

`/sbin/ifdown 'wlan0'`

`sleep 5`

`/sbin/ifup --force 'wlan0'`

`fi`

Il primo comando dello script esegue un ping all'indirizzo dell'Access Point. Per inciso è necessario modificare lo script nel caso l'indirizzo IP non corrisponda a quello dell'Access Point, codificando l'indirizzo IP corretto. In caso il ping abbia esito positivo lo script termina senza eseguire alcuna azione. Nel caso il ping non riceva risposta, lo script disabilita e riabilita, dopo un piccolo intervallo, l'interfaccia della rete wifi, in modo da provocare la riconnessione. Ora bisogna fare in modo che lo script venga eseguito ad intervalli regolari e, per completezza, all'accensione di Raspberry Pi. Per ottenere questo risultato è necessario eseguire un altro paio di configurazioni. Una nel file `/etc/`

`rc.local` per eseguire la connessione all'accensione di Raspberry Pi, come vediamo di seguito:

```
# Print the IP address
_IP=$(hostname -I) || true
if [ "$_IP" ]; then
    printf "My IP address is %s\n" "$_IP"
fi
```

```
bash /usr/bin/sudo -H /home/RF/checkwifi.sh >> /dev/null 2>&1
```

```
bash /home/RF/RF_init start
```

```
exit 0
```

La seconda configurazione permette di eseguire il controllo della connessione ad intervalli regolari, diciamo ogni cinque minuti, per non interferire troppo con le elaborazioni normali. Per queste funzionalità utilizziamo il servizio di schedulazione cron che è il servizio predefinito in GNU/Linux per la gestione dell'esecuzione di programmi e comandi a tempo. Per funzionare, cron si appoggia a una tabella che contiene le "istruzioni" circa i programmi da eseguire e in quali istanti lanciarli. La tabella utilizzata da cron si gestisce con la funzionalità `crontab` la cui sintassi è la seguente:

`crontab <opzione>` dove opzione può assumere i valori `-e`, `-l` e `-r`.

Il comando:

```
crontab -e
```

ci permette di aprire il file di configurazione predefinito di `crontab`.

Ogni riga del file di configurazione ha la seguente struttura:

* * * * * comando da eseguire

- - - - -

```
| | | |
| | | | +---- giorno della settimana (0-6) (0=Domenica)
| | | +----- mese (1-12)
| | +----- - giorno del mese (1-31)
| +----- - ora (0-23)
+----- - min (0-59)
```

Ciascun parametro può essere configurato come serie di valori es: 0,3,7,9 come un intervallo es: 3-7 oppure un misto dei due, esempio: 0,3,4-6,9 oppure ancora come frequenza `*/5` (ogni 5 minuti, ore, ecc.).

Fig. 14

Nel nostro caso vogliamo che ogni cinque minuti venga eseguito lo script `checkwifi.sh` per verificare lo stato in essere della connessione e, nel caso, ripristinarla. Nella finestra che ci mostra il contenuto del file aggiungete in fondo al testo, come visibile in **Fig. 12**, il comando indicato di seguito:

```
*/5 * * * * /usr/bin/sudo -H /home/RF/checkwifi.  
sh >> /dev/null 2>&1
```

che significa: far eseguire lo script ogni cinque minuti di ogni ora, giorno e mese. Usciamo dal file con <CTRL>X, rispondiamo "Y" alla richiesta di salvataggio e confermiamo il nome del file con "Invio". D'ora in avanti ogni cinque minuti verrà lanciato lo script.

Per modificare le impostazioni di cron richiamiamo la funzionalità:

```
crontab -e
```

Per vedere le impostazioni utilizziamo l'opzione:

```
crontab -l
```

mentre per interrompere l'esecuzione automatica digitiamo:

```
crontab -r
```

Un reboot metterà tutto in funzione.

Prima di terminare indichiamo la soluzione di un ulteriore problema che potrebbe verificarsi, per esempio, a seguito di uno spegnimento di Raspberry Pi causato da una caduta della tensione di alimentazione. In questo caso la partizione di boot non viene smontata in modo corretto ed il tutto potrebbe risultare in una partizione "sporca". Ve ne potete accorgere con il comando:

```
dmesg | grep fsck
```

e ricercando la dicitura (**Fig. 13**):

```
FAT-fs (mmcblk0p1): Volume was not properly  
unmounted. Some data may be corrupt. Please run  
fsck
```

Anche in questo caso lo strumento per risolvere il problema è già preinstallato nell'immagine scaricabile dal sito UDOiT. Dalla finestra di terminale aperta con PuTTY ed alla quale vi siete loggati

come utente "root" (password "pi") smontate la partizione di root con il comando:

```
umount /boot
```

Andate alla cartella "dosfstools" con il comando:

```
cd /home/pi/dosfstools
```

e date il comando:

```
./fsck.fat -V /dev/mmcblk0p1
```

rispondete "1" al prompt, ed al termine date i comandi:

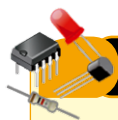
```
./fsck.fat -a /dev/mmcblk0p1  
mount /boot
```

Il tutto è visibile in **Fig. 14**.

Per ora ci fermiamo qui, nei prossimi articoli affronteremo l'argomento relativo alla personalizzazione del comportamento del sistema di controllo domestico, mediante l'utilizzo degli script che, in combinazione con le funzionalità messe a disposizione dalle "scene", permettono di gestire tutta l'applicazione con estrema flessibilità.

Ricordiamo che quanto descritto in questo articolo si applica a Raspberry Pi indipendentemente dall'applicazione di controllo domestico.

Per qualunque utilizzo che richieda di connettere Raspberry Pi alla rete wifi sia come client che in funzione di Access Point, potete fare riferimento a queste note. ■



per il MATERIALE

I prodotti presentati in questo progetto possono essere acquistati presso Futura Elettronica. Il dongle USB (cod. RFT868USB) costa 39,00 Euro, il sensore magnetico (cod. RFT868MAG) è disponibile a 49,00 Euro, il sensore di movimento (cod. RFT868PIR) è in vendita a 55,00 Euro, il modulo a relé (cod. RFT868REL) costa 55,00 Euro e il modulo a dimmer (cod. RFT868DIM) è disponibile a 55,00 Euro. Il radiocomando a 4 canali (cod. RFT8684CH) è in vendita a 39,00 Euro, e la Raspberry Pi 2 Tipo B (cod. 8326274RS) a 44,00 Euro.

Il materiale va richiesto a:

Futura Elettronica, Via Adige 11, 21013 Gallarate (VA)
Tel: 0331-799775 • Fax: 0331-792287 - www.futurashop.it



Chi stampa oggetti 3D ha bisogno di programmi di modellazione ed editing tridimensionale per creare e ritoccare i file; un buon esempio è OpenSCAD, che permette tra l'altro di elaborare molti dei file scaricati da Thingiverse. Prima puntata.

CONOSCIAMO OpenSCAD

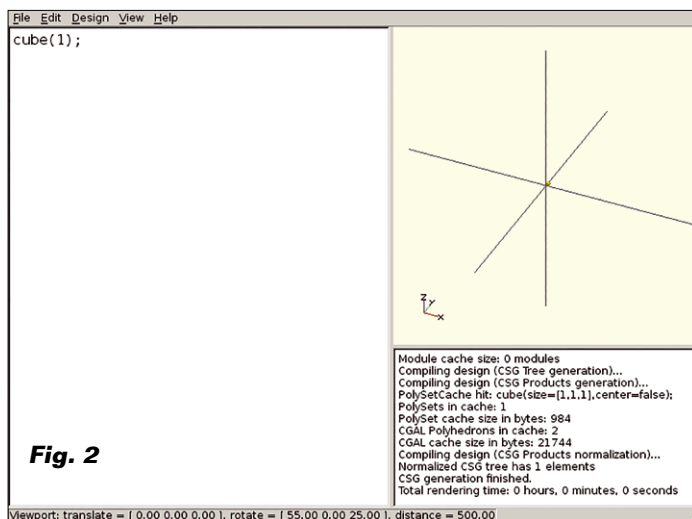
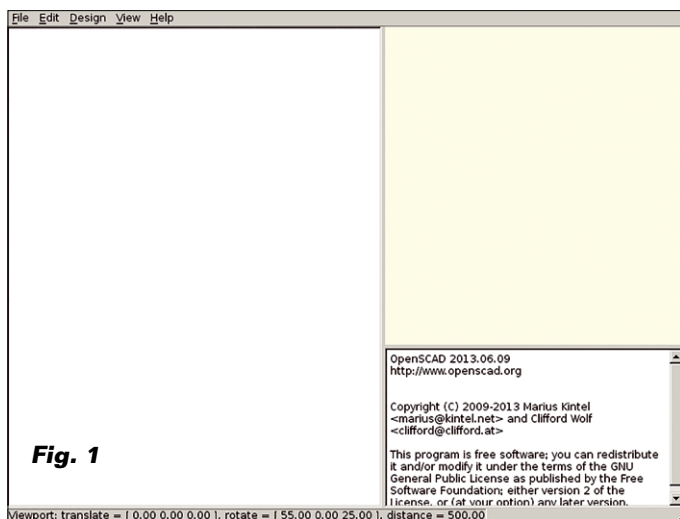
a cura di L.I.F.O. (Laboratorio Informatico Free&Open)

Per modellazione 3D, si intende la progettazione di una forma tridimensionale, la generazione del modello e la sua rappresentazione in un formato elaborabile da un computer; generalmente è il primo passo di procedure anche molto diverse tra loro: ad esempio serve per fabbricare oggetti fisici con strumenti a controllo numerico, ma anche per usi puramente "virtuali" come la realizzazione di immagini o filmati in grafica 3D. Esistono due tecniche di modellazione 3D: manuale e algoritmica. Un modello può essere ottenuto anche per scansione. Le tecniche manuali sono ispirate a quelle tradizionali delle arti plastiche, nelle quali si manipola un materiale fino a fargli ottenere la forma voluta; nelle algoritmiche, invece, si usano tecniche di programmazione per

descrivere l'oggetto. Le tecniche algoritmiche sono più adatte sia ad ottenere precisione che a riprodurre comportamenti simili più volte.

Anche per la rappresentazione in formato elaborabile dal computer esistono svariati metodi: uno dei più diffusi, ad esempio usato da vari programmi famosi come Blender o Google Sketchup, è l'uso di mesh poligonali che descrivono la superficie esterna del modello. Però ne esistono svariati altri e tra questi, degna di nota è la geometria solida costruttiva (Constructive Solid Geometry - CSG) nella quale si descrive l'intero volume del modello costruendolo a partire da primitive semplici come cubi, cilindri, sfere, combinate tra di loro mediante unioni, intersezioni e differenze.

Come si può immaginare, ottenere una rappre-



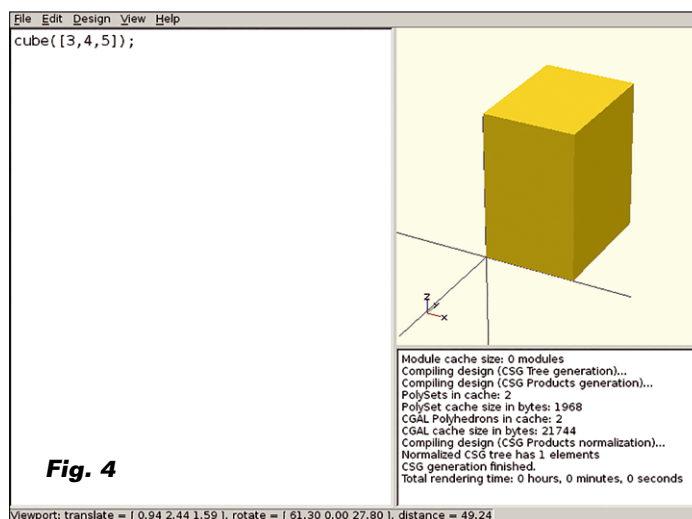
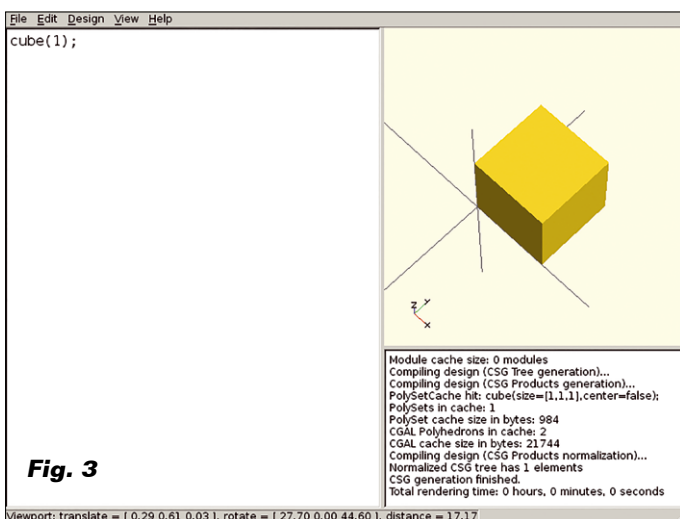
sentazione realistica di forme organiche con questo sistema richiede molto lavoro; per contro, si tratta di un sistema molto adatto alla descrizione di parti meccaniche: infatti queste tendono ad essere facilmente scomponibili in primitive geometriche, e sono più naturalmente rappresentate da oggetti naturalmente solidi che non da superfici delle quali ci si deve preoccupare di definire cosa si debba considerare "interno". A seconda dell'obiettivo che si vuole raggiungere, infatti, si deve prestare attenzione a diversi aspetti della modellazione: se nella grafica 3D sono importanti l'aspetto esteriore del modello e le proprietà visuali delle sue superfici, lavorando su modelli che devono essere riprodotti fisicamente è importante la struttura vera e propria dell'oggetto, ed in questo una rappresentazione basata su solidi anziché superfici è indubbiamente di aiuto.

OpenSCAD è un programma libero per la creazione di modelli 3D basato sulla geometria solida

costruttiva, che quindi eccelle nella modellazione di oggetti in cui è richiesta precisione: a partire dai più semplici come una scatola rettangolare fino a strutture o meccanismi complessi.

L'altra caratteristica immediatamente evidente di OpenSCAD è che i modelli non vengono creati tramite un'interfaccia basata sull'utilizzo del mouse, ma sono descritti in modo algoritmico in un semplice linguaggio di programmazione: questa è una delle interfacce utente più semplici che permettono di ottenere la precisione richiesta dalle applicazioni CNC, ed inoltre fornisce utili strumenti per la realizzazione di pezzi composti da parti ripetute o descrivibili con formule matematiche.

Queste caratteristiche, assieme alla facilità d'uso che lo caratterizzano, rendono OpenSCAD una buona scelta per l'hobbista che vuole aggiungere ai suoi progetti di elettronica strutture o anche meccaniche prodotte con una stampante 3D come la 3Drag o con altri strumenti a controllo numerico.



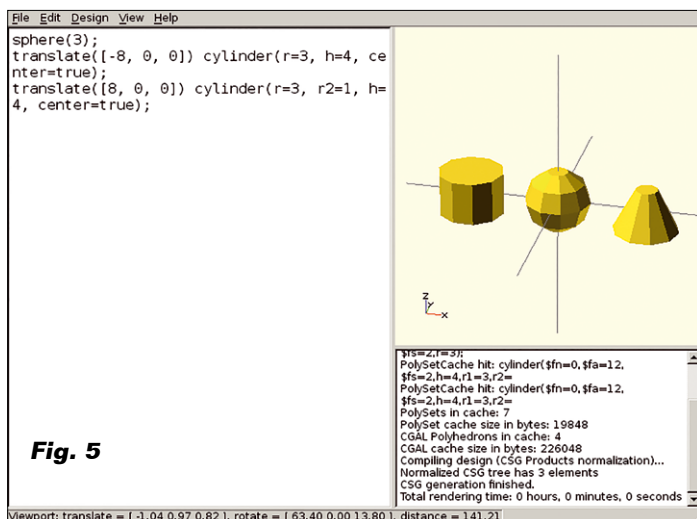


Fig. 5

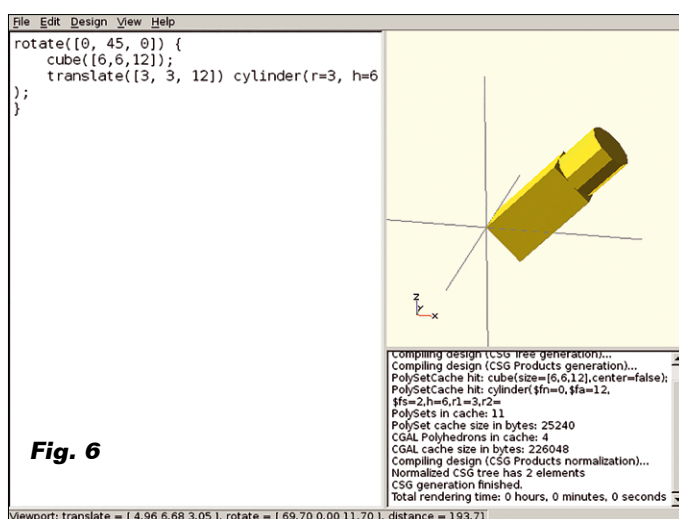


Fig. 6

Grazie a tutte queste caratteristiche, OpenSCAD è uno dei programmi più usati per la realizzazione dei modelli 3D che si possono trovare sui vari siti di condivisione (come ad esempio Thingiverse), anche in virtù della facilità di realizzazione di modelli parametrici e facilmente personalizzabili. Nella prima puntata di questo tutorial vedremo le funzionalità di base di OpenSCAD, sufficienti comunque alla realizzazione di contenitori personalizzati per i propri progetti.

IL CUBO UNITARIO E ALTRI SOLIDI

Lanciando OpenSCAD, notiamo subito l'interfaccia divisa in tre parti: a sinistra c'è l'area che ospita il codice sorgente, mentre a destra si trovano l'area dedicata alla visualizzazione del modello renderizzato ed una sezione dedicata al log dei messaggi del programma, che ci dà il benvenuto ricordando la licenza (GNU GPL) sotto cui è rilasciato il programma (Fig. 1). Se non sono già visibili, chiediamo a OpenSCAD di visualizzare gli assi x, y e z nella finestra di renderizzazione (menu View → Show Axes) e possiamo procedere a descrivere il nostro primo modello. Nella sezione di sinistra scriviamo:

```
cube(1);
```

renderizziamo cliccando su Design → Compile e possiamo vedere apparire il nostro modello (Fig. 2). OpenSCAD funziona con delle unità assolute, ma gli strumenti successivi generalmente le interpretano come millimetri, quindi il nostro cubo da 1 di lato risulta veramente piccolo; per poterlo vedere chiaramente dovremo avvicinare un po' il punto di vista: con il mouse nella finestra di rendering possiamo usare la rotella per avvicinarsi ed allontanarsi, trascinare con il tasto sinistro per ruotare la visuale e trascinare con il tasto destro per traslarla (Fig. 3).

Dopo aver ammirato il nostro cubo da tutti i lati, procediamo a vedere un po' meglio le primitive geometriche di OpenSCAD. Innanzitutto va detto che l'istruzione `cube` serve in realtà a disegnare dei parallelepipedi rettangoli (Fig. 4): le misure sui tre assi vengono indicate come un vettore, ovvero una tripletta di numeri tra parentesi quadre:

```
cube([3,4,5]);
```

Le altre due primitive solide più usate sono `sphere`, il cui unico parametro è il raggio, e `cylinder`, che in realtà può disegnare anche tronchi di cono, specificando i raggi delle basi inferiore (`r`) e superiore (`r2`).

Molte primitive accettano poi il parametro `center`, che specifica di disegnare il modello centrato su tutti gli assi, e non a partire dal piano dell'origine:

```
sphere(3);
translate([-8, 0, 0]) cylinder(r=3, h=4, center=true);
translate([8, 0, 0]) cylinder(r=3, r2=1, h=4,
center=true);
```

Qui abbiamo anche visto la nostra prima trasformazione, `translate()`, che si applica all'elemento successivo e lo trasla secondo il vettore specificato come parametro.

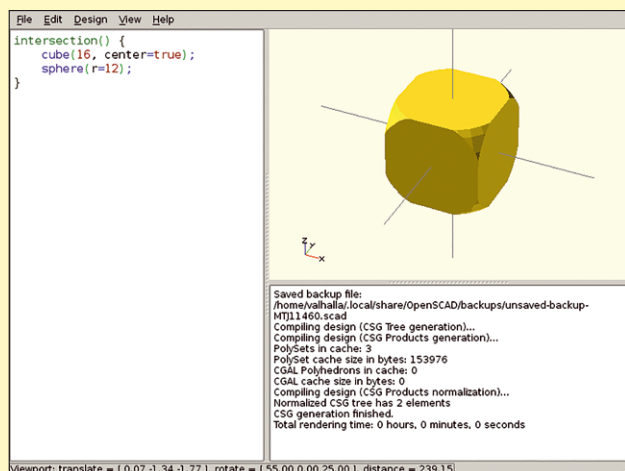
Una trasformazione si può applicare anche a più elementi, racchiudendoli tra parentesi graffe: applichiamo una traslazione ad un cilindro ed una rotazione di 45° (Fig. 6) sia al cilindro traslato che ad un parallelepipedo:

```
rotate([0, 45, 0]) {
    cube([6,6,12]);
    translate([3, 3, 12]) cylinder(r=3, h=6);
}
```

Intersezioni

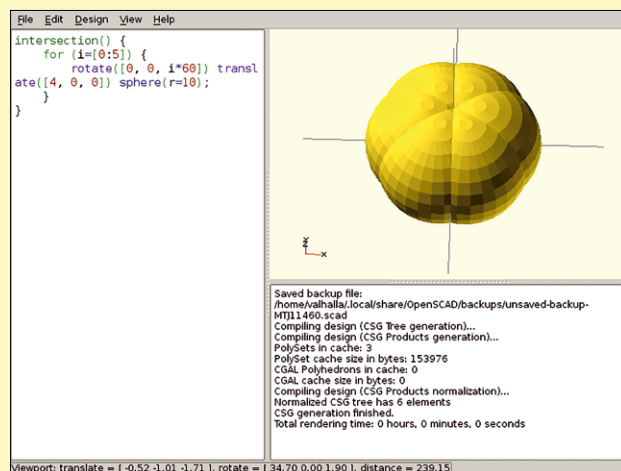
Oltre all'unione e alla differenza di solidi, un'operazione comune è la loro intersezione: ad esempio un'intersezione tra un cubo ed una sfera è un metodo per ottenere un cubo con gli angoli arrotondati:

```
intersection() {
  cube(16, center=true);
  sphere(r=12);
}
```



Però bisogna fare attenzione nel caso si vogliano intersecare più oggetti mediante un ciclo for: dato che OpenSCAD effettua un'unione implicita degli oggetti contenuti nel ciclo, il risultato è che questa unione verrà usata come oggetto unico nell'intersezione.

```
intersection() {
  for (i=[0:5]) {
    rotate([0, 0, i*60]) translate([4, 0, 0]) sphere(r=10);
  }
}
```



La rotazione è descritta dalla trasformazione `rotate`, che prende come parametro un vettore con gli angoli dei quali effettuare le tre rotazioni rispetto ai tre assi. È importante ricordare che il centro di rotazione è sempre l'origine degli assi, quindi in molti casi è pratico preparare prima la componente con l'inclinazione giusta, poi traslarla nella sua posizione definitiva.

Altre trasformazioni semplici sono `scale`, che moltiplica le dimensioni del modello per il valore fornito come parametro, e `resize`, che invece le

adatta fino a raggiungere i valori forniti.

Ad esempio, sia `scale([0.5, 1, 1])` `sphere(r=10)` che `resize([5, 0, 0])` `sphere(r=10)` disegneranno una sfera deformata per essere lunga 5 sull'asse delle x e 10 lungo gli altri assi. Infine, `mirror` serve per ottenere una copia speculare di un modello e prende come parametro il vettore perpendicolare al piano rispetto al quale specchiare il modello. Come nel caso della rotazione, questo piano passerà necessariamente per l'origine.

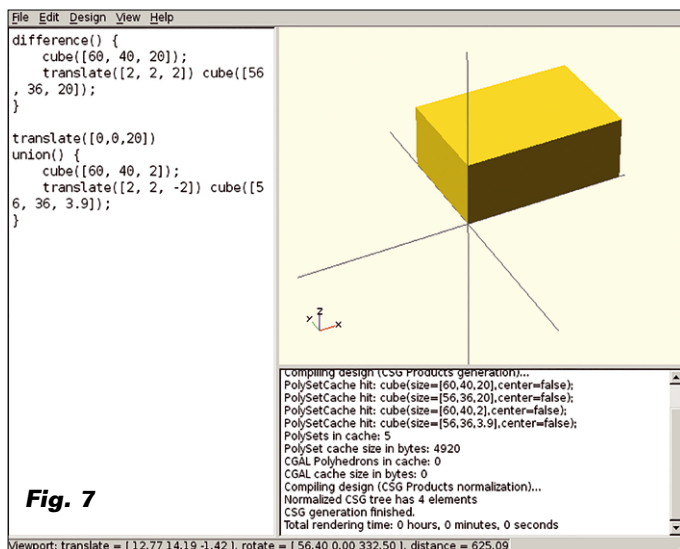


Fig. 7

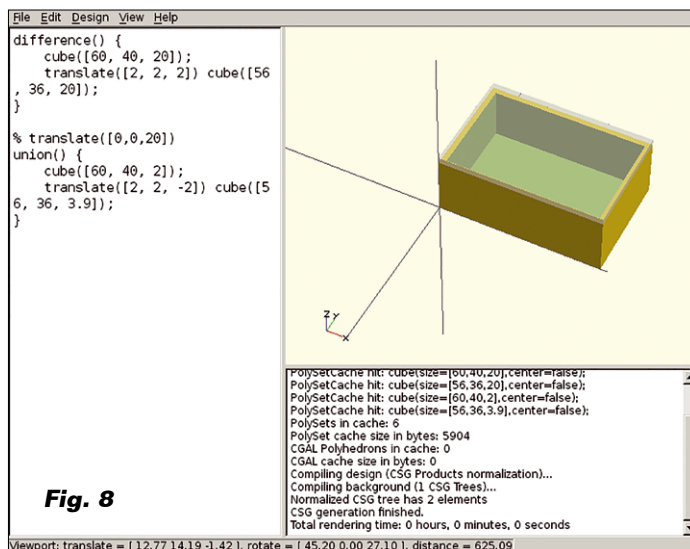
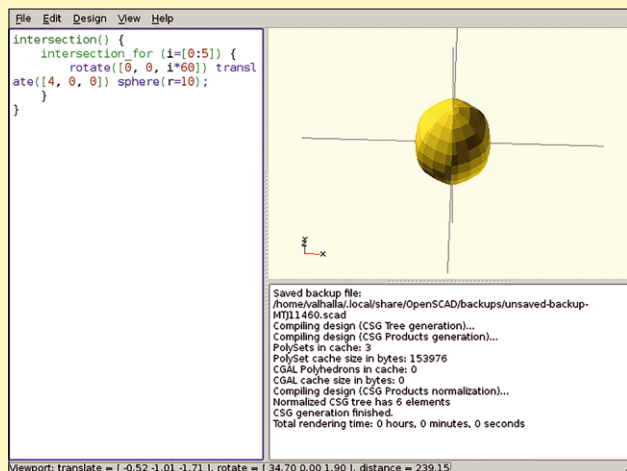


Fig. 8

Per ovviare a questo problema esiste un'istruzione apposta: `intersection_for`, che effettua un ciclo `for` effettuandone implicitamente non l'unione ma l'intersezione, in modo da essere compatibile con l'operazione di intersezione.

```
intersection() {
  intersection_for (i=[0:5]) {
    rotate([0, 0, i*60]) translate([4, 0, 0]) sphere(r=10);
  }
}
```



IL PRIMO MODELLO

Adesso, come nostro primo modello possiamo costruire una semplice scatola rettangolare, la cui base sarà composta dalla differenza tra due parallelepipedi, di cui uno più piccolo e traslato delle dimensioni della parete:

```
difference() {
  cube([60, 40, 20]);
  translate([2, 2, 2]) cube([56, 36, 20]);
}
```

Una cosa importante da ricordare è che quando si fanno differenze e unioni di solidi, questi ultimi non devono avere facce o parti di facce in comune, ma debbono intersecarsi. Ad esempio, se il secondo parallelepipedo del cubo sopra fosse stato `cube([56, 36, 18])` OpenSCAD non sarebbe stato in grado di stabilire come gestire la faccia superiore, e non avrebbe generato un modello valido. Per evitare questi problemi, nei casi in cui non si possa abbondare con le misure in questo modo, basta aggiungere o togliere 0.01 dalla dimensione opportuna: è una differenza abbastanza piccola da non notarsi nel modello finito, ma sufficiente ad OpenSCAD per distinguere interno ed esterno. Il coperchio, invece, sarà formato dall'unione di due parallelepipedi: uno largo come la scatola e l'altro più stretto, in modo da incastrarsi; ecco come viene costruito:

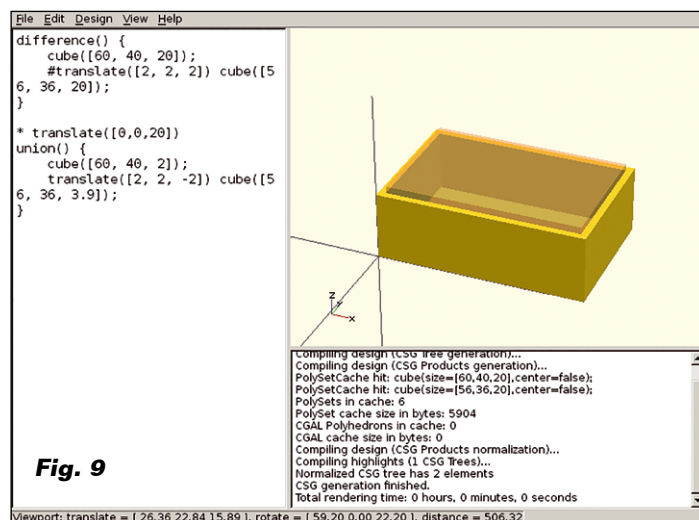


Fig. 9

```
translate([0,0,20])
union() {
  cube([60, 40, 2]);
  translate([2, 2, -2]) cube([56, 36, 3.9]);
}
```

Compilando questo file sembrerà di vedere un parallelepipedo pieno, come in Fig. 7. Possiamo però sfruttare i modificatori di OpenSCAD per visualizzare meglio la struttura dell'oggetto: premettendo un `%` ad un oggetto, questo verrà ignorato nella composizione del modello, ma sarà visualizzato in grigio trasparente, come si vede in Fig. 8.

Gli altri modificatori supportati sono `*`, che serve ad escludere totalmente una componente dal modello, `!` che al contrario fa sì che venga renderizzata solo la componente specificata e `#` che usa normalmente la componente, ma la disegna anche in rosa trasparente; quest'ultimo, ad esempio, è utile per verificare il corretto posizionamento di componenti che vengono sottratte all'oggetto con l'operazione `difference` (Fig. 9).

Come ci si può aspettare da un linguaggio di programmazione, è possibile aggiungere dei commenti; OpenSCAD usa i commenti in stile C++ e con `//` si può inserire un commento da una sola riga, mentre con `/*` e `*/` si delimitano commenti da più righe. Completiamo il nostro primo modello con qualche commento descrittivo e salviamo il file con estensione `.scad`. Prima di poterlo esportare nel formato STL per poterlo stampare, dobbiamo fare un ultimo passaggio: con il comando `Design → Compile and Render (CGAL)` facciamo calcolare correttamente il modello e dopo alcuni secondi, quando l'immagine sarà nuovamente apparsa, possiamo finalmente usare il comando `Design → Export as STL...`

Il risultato è visibile in Fig. 10.

Per poter realizzare anche il coperchio, basterà to-

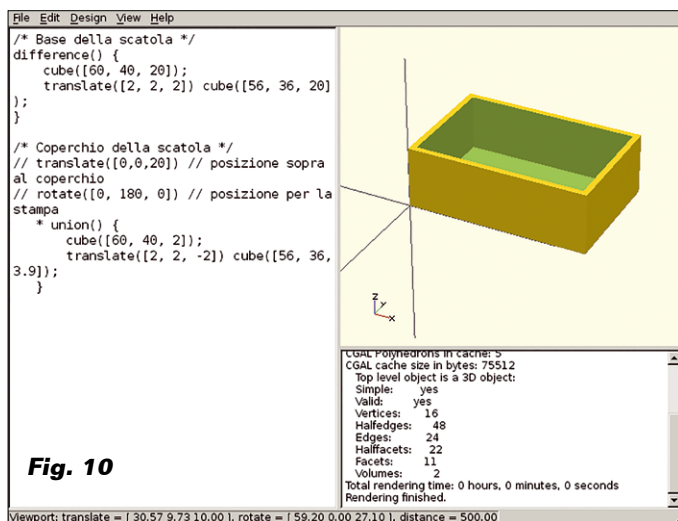


Fig. 10

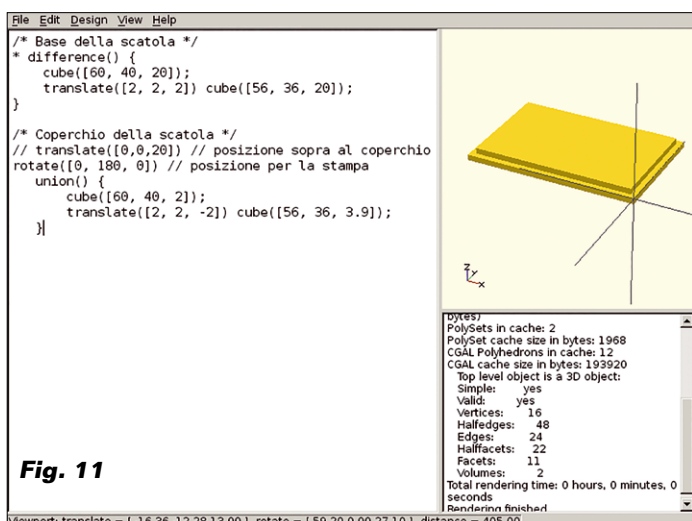


Fig. 11

gliere il modificatore * dal coperchio, metterlo alla base e poi decommentare la riga con la rotazione, la quale fa in modo che il coperchio sia stampabile senza bisogno di supporti; adesso basta lanciare di nuovo Design → Compile and Render (CGAL) e

quindi Design → Export as STL... (Fig. 11).

SCATOLA PER UN PROGETTO

Vediamo adesso una versione un po' più complessa della scatola, descritta in modo parametrico e sfrut-

Progettazione di modelli per la stampa 3D

Anche se il nostro modello è stato compilato a dovere ed esportato correttamente in STL, non è detto che sia ideale per la stampa 3D, la quale presenta vincoli

specifici e spesso diversi da quelli imposti da altri usi dell'STL come il rendering o la fabbricazione in CNC. La prima considerazione da fare ovviamente è: i dettagli sono sufficientemente grandi da apparire nell'oggetto stampato, considerando la precisione della stampante?

Una stampante per hobbisti come la nostra 3Drag lavora con strati spessi alcuni decimi di millimetro, per cui inserire dettagli più piccoli di mezzo millimetro risulta inutile e controproducente, dato che non verranno riprodotti.

Un'altra verifica ovvia ma importante, riguarda il fatto che l'oggetto risulti sufficientemente robusto o abbia punti deboli che potrebbero rompersi: tra

gli aspetti da controllare lo spessore delle pareti, che con le plastiche usate nella 3Drag dovrebbe essere di almeno un millimetro, e che non ci siano parti troppo sottili soggette a particolare sforzo.

Sotto questo aspetto è importante ricordare che le stampanti lavorano a strati orizzontali: un oggetto lungo e stretto disposto orizzontalmente è formato da filamenti paralleli che gli danno una discreta resistenza; lo stesso pezzo stampato verticalmente è diviso in strati più deboli e sarà quindi più facile romperlo. Un esempio di rottura è visibile nella foto in questo riquadro: dei due lati, di uguali dimensioni, quello stampato in orizzontale rimane intero, mentre quello stampato in

verticale si rompe con un taglio netto lungo uno degli strati.

Le stampanti che lavorano deponendo filamenti di materiale hanno ovviamente bisogno di una superficie di appoggio sulla quale deporli: se da un lato gli slicer possono ovviare al problema aggiungendo strutture di supporto, dall'altro tenere presente il vincolo in fase di progettazione permette di ottenere oggetti rifiniti meglio e con minore spreco di materiale. Da questo punto di vista, uno dei criteri da prendere in considerazione è l'angolo con cui salgono eventuali sporgenze (overhang): angoli all'incirca sopra i 45 gradi vanno generalmente bene, mentre sotto i 45 è facile vedere i filamenti cadere nel vuoto.



tando più potenzialità di OpenSCAD. Il sorgente completo di questo modello è disponibile nel **Lista-to 1**; (il file è scatola_arduino.scad) vediamo mano che funzionalità abbiamo usato e come. Innanzitutto prendiamo le misure di ciò per cui vogliamo creare la scatola; dato che vogliamo renderla parametrica salveremo queste misure con alcune variabili.

Una cosa cui si deve fare attenzione è che le variabili di OpenSCAD si comportano in modo peculiare e sono più simili a costanti parametriche che non alle variabili vere e proprie di altri linguaggi; in particolare, il loro valore viene calcolato in fase di compilazione e non varierà nel corso del rendering. Ad esempio, usando il comando `echo` per stampare testo nel log dei messaggi, il codice:

```
a = 1;
echo(a);
a = 2;
echo(a);
```

Per risolvere sia questo problema che il precedente può essere utile fare attenzione all'orientamento del pezzo in fase di stampa: a volte una semplice rotazione può eliminare gli effetti dell'overhang (l'abbiamo fatto per il coperchio della scatola semplice). In alternativa all'appoggio su un filamento completo, si può prevedere un buco rettangolare su una parete verticale, facendo perciò appoggiare i filamenti sulle pareti laterali: questa tecnica (ponte o bridge) può funzionare per piccole distanze, attorno al centimetro e dipendenti dalla velocità di stampa (velocità maggiori possono superare vuoti più lunghi, entro certi limiti). Anziché usare dei ponti,

però, può valere la pena progettare l'oggetto in modo tale che non ci siano situazioni del genere, come ad esempio nella nostra scatola per arduino dove i buchi per i connettori arrivano fino in cima al bordo (ma potrebbero essere chiusi da un opportuno coperchio). Un'ultima considerazione riguarda il risparmio di materiale usato: è vero che lo slicer può lavorare molto su questo aspetto, specialmente tramite le impostazioni per il fill di modelli solidi, ma in alcuni casi il modello stesso può essere modificato per ridurre il volume stampato. Un esempio limite può essere l'uso di reticoli (tenendo presente i vincoli sugli angoli di crescita) al posto di pareti piene.

Thingiverse Customizer

Navigando sui modelli per stampante 3D disponibili online, si può notare come OpenSCAD sia un programma molto usato: in particolare, è il linguaggio supportato dall'app Customizer di Thingiverse che permette di caricare modelli parametrici facilmente personalizzabili dagli utenti del sito.

La cosa più semplice che si possa fare è creare un file .scad che usi delle variabili dichiarate all'inizio, prima di ogni modulo, nel formato:

```
// Descrizione
variabile = valore_di_default;
```

Il Customizer permette molto altro: per approfondire l'argomento se ne può consultare la documentazione online all'indirizzo <http://customizer.makerbot.com/docs>.

fornirà come risultato quanto di seguito:

```
ECHO: 2
ECHO: 2
```

Tornando alla nostra scatola, non avremo problemi ad impostare le misure della nostra scheda (nell'esempio un Arduino UNO, che molti di voi avranno sottomano) tramite variabili il cui valore è impostato all'inizio del file.

Dopo aver inserito dimensioni della scheda nei tre assi, posizione dei buchi di supporto (sulle coordinate x ed y, relative alla scheda) e delle aperture da praticare nel bordo della scatola (nel nostro caso solo su un lato, per i connettori sporgenti di alimentazione e USB), possiamo impostare alcuni parametri come la larghezza delle pareti, il margine di tolleranza da lasciare attorno al contenuto e l'altezza dei supporti.

A questo punto abbiamo bisogno di una nuova funzionalità di OpenSCAD: dato che vogliamo ripetere i supporti in posizioni diverse, ci conviene usare un modulo per definire la loro forma una sola volta. I moduli corrispondono grosso-modo alle funzioni o alle macro di altri linguaggi di programmazione e per definirli basta usare la parola chiave `module` seguita da un nome, eventualmente dei parametri tra parentesi tonde e da una serie di istruzioni e componenti tra parentesi graffe.

Nel corso della costruzione del modello possono poi essere richiamati usandone semplicemente il nome, con i valori dei parametri tra parentesi tonde. Nel nostro caso, definiamo un modulo `pin` al quale passiamo come parametri la posizione x,y in cui dovrà essere posizionato il pin rispetto alla scheda. Il codice del pin sarà semplicemente l'unione di due cilindri, traslata nella posizione voluta.

Passiamo quindi a definire la scatola vera e propria:

Listato 1

```

/* ***** Board measurements ***** */

board_x = 69;
board_y = 53;
board_z = 13;

pins = [
    [14, 3],
    [15, 51],
    [66, 7],
    [66, 36]
];

// This assumes that all of the connectors are on one side.
connectors = [
    [3, 11],
    [31, 14]
];

/* ***** Build parameters ***** */

wall = 2;
ease = 1;
v_ease = 4;
pin_height = 2;

/* ***** Modules ***** */

module pin(x,y) {
    translate([x + wall + ease, y + wall + ease, wall - 0.01]) union() {
        cylinder(h=v_ease + pin_height, r=1);
        cylinder(h=v_ease, r=2.5);
    }
};

/* ***** Box ***** */

d = (wall + ease) * 2;

union() {
    difference () {
        cube([board_x + d, board_y + d, board_z + wall + v_ease]);
        translate([wall, wall, wall])
        cube([board_x + ease*2, board_y + ease * 2, board_z + v_ease + 1]);
        for ( c = connectors ) {
            translate([-1, c[0] + wall + ease, wall + v_ease])
            cube([wall + 2, c[1], board_z + 1]);
        }
    }
    for ( p = pins ) {
        pin(p[0],p[1]);
    }
}

```

come prima, la struttura principale sarà composta dalla differenza tra due cubi, con l'unica differenza che qui le misure sono calcolate a partire dai parametri impostati in precedenza.

Alla struttura principale, in questo caso vogliamo aggiungere i supporti per la scheda e rimuovere i buchi per i connettori sporgenti: in entrambi i casi abbiamo fatto uso di un ciclo for. In OpenSCAD i cicli for possono lavorare su un vettore oppure su

un range numerico: nel nostro caso lavoriamo sui vettori di pins e connectors. Se avessimo voluto lavorare su un range, avremmo potuto utilizzare una sintassi come:

```

for (i=[0:2]) {
    echo(i);
}

```

la quale stamperà:

0

1
2

Nel caso delle aperture per i connettori, in questo caso ci siamo limitati ad un lato soltanto, avendo una scheda con connettori sporgenti su più lati può essere un buon esercizio cambiare connectors in un vettore di vettori di vettori e ciclare su due livelli. Anche in questo caso, una volta finito e salvato il modello sarà sufficiente il comando Design → Compile and Render (CGAL) seguito dall'esportazione in STL e si avrà un file usabile con i consueti programmi per la stampa in 3D. Se vogliamo condividere il nostro modello anche su un sito che non supporti direttamente i formati .scad o .stl, possiamo anche salvare l'immagine presente nella finestra di preview: sempre dopo aver usato Compile and Render (CGAL) si può usare il comando Design → Export as Image...

CONCLUSIONI

In questa puntata abbiamo solo iniziato a sfiorare la superficie delle potenzialità di OpenSCAD, volendoci concentrare soprattutto sulle caratteristiche usabili immediatamente per la realizzazione di modelli semplici ma di uso pratico.

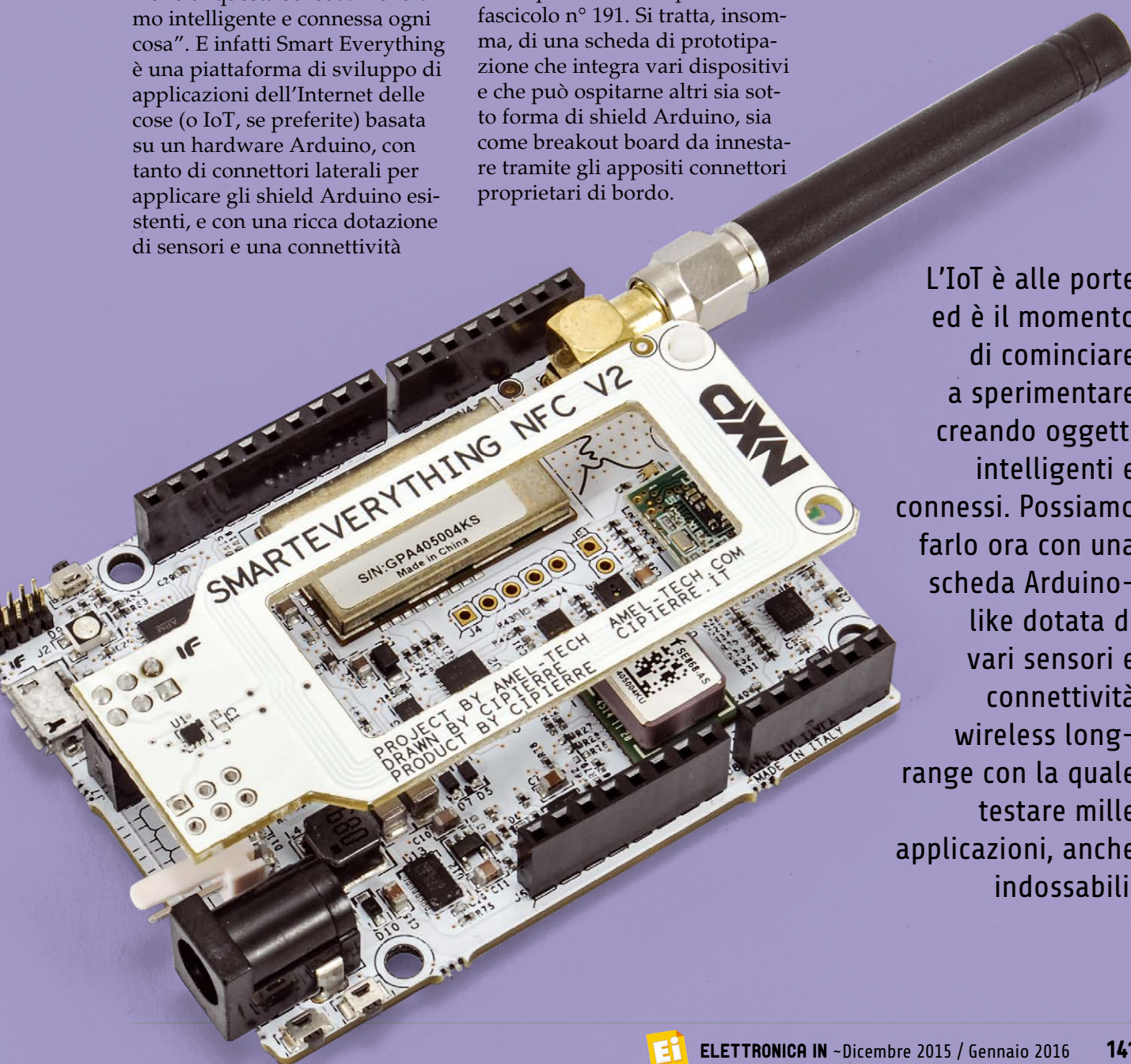
Nella prossima puntata di questo tutorial avremo modo di affrontare funzionalità più complesse, che rendono OpenSCAD un'ottima soluzione per esigenze di modellazione a qualunque livello. Ricordiamo che OpenSCAD è un programma di libero utilizzo (sotto licenza GNU GPL) e multiplatforma, disponibile nei repository della maggior parte delle distribuzioni Linux e scaricabile gratuitamente per Windows e OSX dal sito ufficiale <http://www.openscad.org/>. ■

SMART EVERYTHING: TUTTO DIVENTA SMART

di BORIS LANDONI

Il nome è di per sè un programma e rivela già la vocazione di questa scheda: "rendiamo intelligente e connessa ogni cosa". E infatti Smart Everything è una piattaforma di sviluppo di applicazioni dell'Internet delle cose (o IoT, se preferite) basata su un hardware Arduino, con tanto di connettori laterali per applicare gli shield Arduino esistenti, e con una ricca dotazione di sensori e una connettività

wireless, inizialmente basata sulla tecnologia long-range SigFox della quale abbiamo parlato nel fascicolo n° 191. Si tratta, insomma, di una scheda di prototipazione che integra vari dispositivi e che può ospitarne altri sia sotto forma di shield Arduino, sia come breakout board da innestare tramite gli appositi connettori proprietari di bordo.



L'IoT è alle porte ed è il momento di cominciare a sperimentare creando oggetti intelligenti e connessi. Possiamo farlo ora con una scheda Arduino-like dotata di vari sensori e connettività wireless long-range con la quale testare mille applicazioni, anche indossabili.



La scheda vista senza la bobina per l'NFC

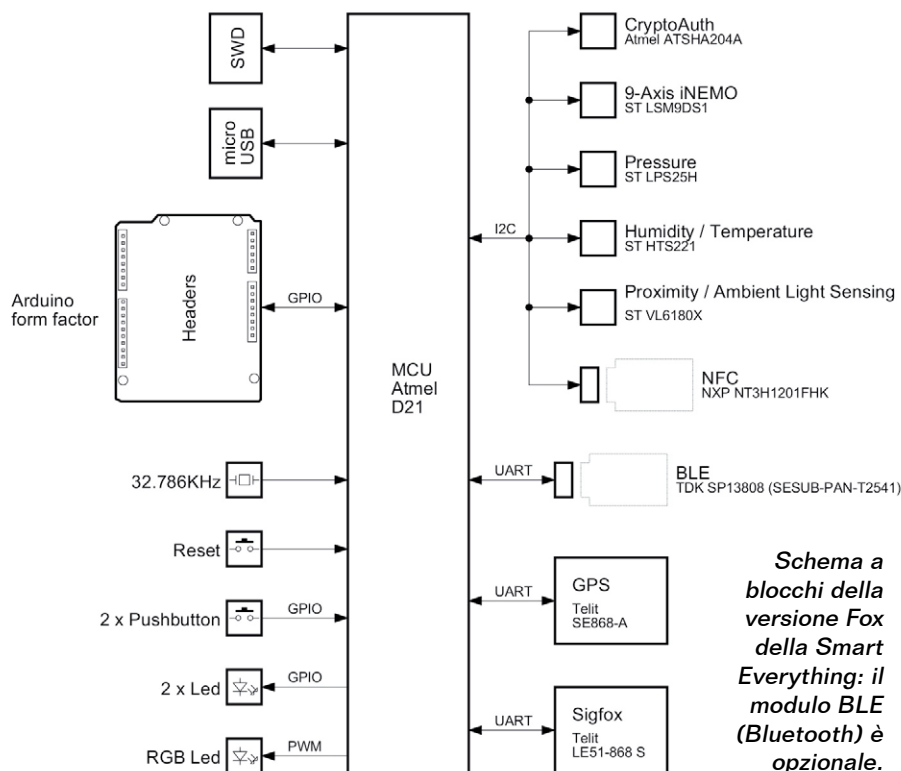
Al momento la Smart Everything è dotata di transceiver long-range basato sulla tecnologia SigFox e nel seguito di questo articolo faremo riferimento a tale versione; i creatori della scheda hanno comunque annunciato che prossimamente verranno offerte due alternative basate, una ancora su un link radio long-range ma con modulo LoRa Semtech, e una dotata di WiFi, per le applicazioni in cui consumo e portata non sono imperativi categorici. Questo non significa che la versione WiFi sprechi energia, dato che comunque andrà a utilizzare una soluzione WiFi dell'Atmel, che ottimizza i consumi (anche se non a livello del LoRa Semtech o del SigFox) grazie a una modalità di standby attivabile quando non serve ritrasmettere dati. In linea di massima le tre tecnologie di connettività sono tutte ottimizzate -pur nel loro contesto- nei consumi di elettricità e sono state previste per soddisfare le esigenze di diverse fasce di mercato. Una delle principali differenze tra esse riguarda il range di copertura della rete, in quanto la SigFox è orientata a reti nazionali o mondiali (similmente a quella telefonica cellulare), LoRa alle reti urbane a corto raggio e il WiFi alle applicazioni

localizzate.

La scheda SmartEverything è rivolta a due mondi: uno è quello industriale e professionale che trova all'interno di questa scheda una serie di tecnologie di un certo livello e innovative con cui sperimentare; il secondo è quello dei maker, diciamo "evoluti" che vogliono realizzare applicazioni

in ambito IoT e con una buona dose di sensoristica, perché il mondo dei maker negli ultimi anni è cambiato nella visione di chi produce elettronica, componenti e tecnologie, passando da un mondo di "smanettoni" a uno di sperimentatori sempre più interessati a finalizzare il proprio impegno.

Oggi c'è una particolare attenzione da parte del mondo industriale ad andare verso il mondo dei maker, perché viene visto come un mercato molto ricco e poi perché oggi sempre più il maker è il progettista di domani, perciò se acquisisce una tecnologia fin dall'inizio, poi se la porta dietro ed è probabile che la riutilizzi nel mondo del lavoro. Questo spiega perché la componentistica per il mondo dei maker è abbondantemente accettata anche dal mondo industriale, in quanto i componenti



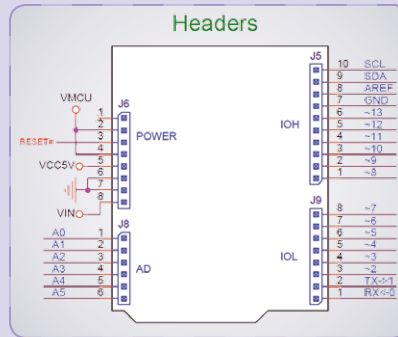
Schema a blocchi della versione Fox della Smart Everything: il modulo BLE (Bluetooth) è opzionale.

offerti ai maker, magari sotto forma di breakout board, sono professionali e vengono offerti allo sperimentatore affinché realizzi le proprie applicazioni con essi.

SmartEverything si colloca a cavallo tra il mondo dei maker e quello della progettazione di applicazioni industrializzabili. Può essere usata dall'industria per fare prototipi per testare un sensore o implementare un'applicazione, e può essere testata da un maker che vede in questa scheda un modulo Arduino dotato di uno shield corposo integrato; infatti, per disporre di tutti i componenti a bordo di SmartEverything (fermo restando che non di tutti l'industria offre uno shield), servirebbero almeno due o tre shield, più Arduino, naturalmente. Il tutto occuperebbe uno spazio nettamente maggiore di quello richiesto dalla nostra scheda.

Inoltre con SmartEverything un "utente evoluto" può realizzare un firmware basato su un sistema operativo Real Time leggero come freeRTOS; questo permette di gestire in maniera multitasking le varie risorse della scheda. Per lo sviluppo e il debugging può essere utilizzato il tool Atmel Studio.

Smart Everything nasce ad opera di Claudio Pino di Amel-Smart Everything nasce ad opera di Claudio Pino, ispirato dalla proposta di Marco Sangalli di Arrow Electronics (www.arroweurope.com) -tra i principali player della distribuzione elettronica- di realizzare una evaluation board da fornire ai propri clienti per prototipare applicazioni IoT basate sulla tecnologia SigFox integrata nei recenti moduli Telit. Dato che SigFox fornisce una connettività assimilabile a una rete cellulare ma basata su



La tecnologia wireless **SigFox**

Per incrementare la portata di un sistema radio si possono seguire due strade: aumentare la potenza del trasmettitore o migliorare la sensibilità del ricevitore. Nelle applicazioni IoT non è possibile aumentare la potenza del TX che, al contrario, deve essere la più bassa possibile per ridurre al minimo i consumi energetici. Da questo punto di vista potenze dell'ordine di 1-25 mW sulle frequenze sub-1 GHz sono accettabili. Con queste potenze e con una sensibilità standard del ricevitore dell'ordine di -110 dBm, la portata non supera i 30-300 metri. La sensibilità di un ricevitore è fortemente influenzata dal rumore intrinseco del dispositivo e da quello presente nell'etere. Per incrementare in maniera significativa la sensibilità non resta che alzare il rapporto S/N (segnale/rumore) ovvero, a parità di potenza del segnale, ridurre il rumore o incrementare la capacità del

dispositivo a valle del demodulatore di discernere il rumore ed eliminarlo.

Le soluzioni adottate si riferiscono a due particolari tecnologie: la spread spectrum e la Ultra Narrow Band. Nel caso dell'UNB, il principale problema da risolvere risiede nella stabilità di frequenza dell'apparato, mentre lo svantaggio è la banda passante particolarmente bassa (nel caso della tecnologia SigFox i dati vengono trasmessi a 100 bit/secondo). Tuttavia, nel caso di applicazioni IoT, i messaggi sono molto brevi per cui, anche con questa velocità, possono essere inviati in un paio di secondi. L'altro grande vantaggio risiede nella possibilità di gestire da parte di un singolo ricevitore un numero incredibilmente alto di trasmettitori, come dimostrano i grafici nei quali si vedono le emissioni di pochi TX convenzionali (con già alcune collisioni) paragonate con le emissio-

ni di 210 TX UNB senza alcuna collisione. Dal grafico in basso a sinistra in questo riquadro si vede che un ricevitore SigFox con tecnologia UNB integrata con Cognitive SDR (oltre a garantire una sensibilità di ben -126 dBm a cui corrisponde una portata che va da 1 a 10 km) è in grado di gestire fino a 3 milioni di dispositivi nel caso ognuno di questi invii 3 messaggi giornalieri occupando appena l'8% dello spettro del canale utilizzato.

Le tecnologie long-range wireless nascono come alternativa alle reti radiomobili esistenti, che per l'IoT sarebbero inutilizzabili in quanto i terminali richiederebbero troppa potenza, mentre gli oggetti da connettere devono consumare pochissimo, essendo alimentati quasi sempre a batteria o con soluzioni di energy harvesting. Pensiamo, ad esempio, ad un rilevatore di fumo per impianti antincendio o ad un sensore di temperatura e umidità di una serra: il rilevatore di fumo deve inviare un paio di messaggi al giorno per dire che è "in vita" e, nel caso di incendio, un segnale d'allarme.

Per oggetti di questo tipo, che rappresentano una grande fetta delle "Cose" da mettere in rete, è sicuramente più adatta una rete realizzata ad hoc, facilmente scalabile e altrettanto facilmente implementabile: e sicuramente di gran lunga più

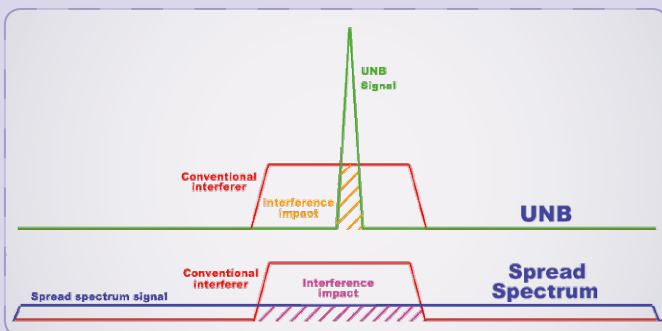
economica di una rete GSM.

L'azienda francese SigFox si è impegnata principalmente nella creazione di reti long-range che già coprono intere nazioni, lasciando a società partner la produzione dei dispositivi (chip e moduli) da utilizzare per realizzare i terminali; recentemente, ai produttori di chip già attivi in questo settore, si è aggiunta Atmel, che ha sviluppato il primo SoC certificato SigFox, low-cost dalle dimensioni particolarmente contenute che sicuramente darà un notevole impulso a questa tecnologia e alle reti in fase di realizzazione.

La tecnologia SigFox opera in banda ISM (868-869 MHz) e per rimanere nei limiti imposti la potenza emessa non supera i 10-25 mW. Sulle frequenze ISM non è possibile trasmettere in continuazione ma con un "active time" limitato; questo è anche uno dei motivi per il quale SigFox ha scelto un massimo di 140 messaggi al giorno per oggetto.

SigFox è una rete wireless già sperimentata in Francia e che gode di una copertura piuttosto ampia a livello europeo con i primi gateway che stanno per essere installati anche in Italia e la disponibilità di chip low-cost (come quello di Atmel) con i quali realizzare le prime interessanti applicazioni.

In figura vedete la configurazione di questa rete con una serie



sistemi long-range e perciò a bassissimo consumo energetico, è sembrato logico integrare nella scheda -oltre ad essa- una serie di fonti di dati; nello specifico, dato che i dispositivi long-range nascono principalmente per interconnettere oggetti intelligenti e quindi per l'Internet delle cose, e nello specifico per far viaggiare i dati relativi a dispositivi mobili e indossabili, il progettista ha voluto incorporare la sensoristi-

ca delle tipiche applicazioni IoT e quindi trasduttori per rilevare le principali grandezze fisiche. Per creare l'assortimento di sensoristica e la dotazione della scheda, Arrow ha interessato aziende di cui tratta i prodotti proponendo loro di sponsorizzare la realizzazione di SmartEverything, fornendo ciascuno un componente; queste aziende sono ST, che ha portato la sensoristica, Telit, che oltre alla

connettività SigFox ha fornito il ricevitore GPS, Atmel che ha messo a disposizione il microcontrollore, NXP che ha messo a disposizione il transceiver per realizzare l'NFC e TDK che ha fornito il modulo per il Bluetooth basato su un integrato della Texas Instruments.

La scheda è praticamente una Arduino Zero, quindi centrata su un micro Atmel SAMD21 basato su core ARM Cortex

di gateway radio in grado di colloquiare con gli oggetti (un milione e più per ciascun gateway) che si trovano nell'area di copertura radio.

I dati ricevuti vengono inviati via GSM o rete fissa ad un sistema centrale di memorizzazione ed elaborazione e "distribuiti" ai Clienti mediante API di Internet in grado di automatizzare la gestione dei dispositivi e implementare l'integrazione dei dati.

L'infrastruttura è talmente semplice e poco costosa, che la copertura dell'intero territorio francese è stata realizzata in poco più di un anno.

Si calcola che un terminale SigFox con chip di ultima generazione alimentato con due batterie a stilo da 2.700 mA sia in grado di funzionare per circa 20 anni inviando 140 messaggi al giorno.

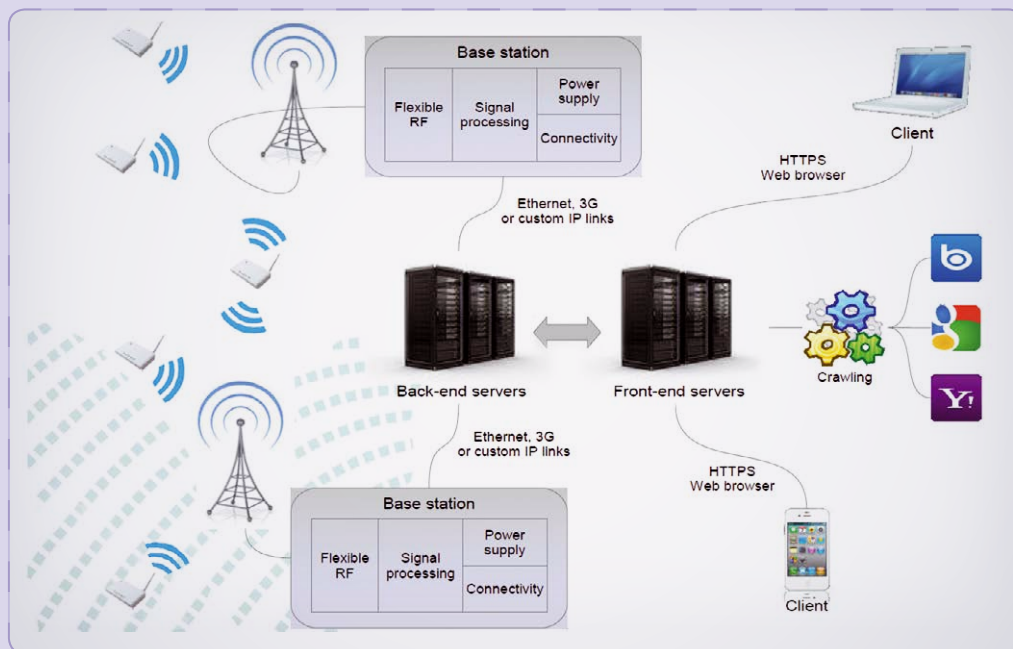
Dal punto di vista tecnico la rete SigFox prevede l'invio di un massimo di 140 messaggi per oggetto connesso al giorno. Ciascun messaggio è formato da un payload di 12 byte; nel caso di dati più lunghi, gli stessi vengono suddivisi in più messaggi in modo da rispettare i 12 byte previsti. Ciascun dispositivo è identificato da un ID di 32 bit e da PAC utilizzabile una sola volta per registrare l'ID SigFox.

I nuovi dispositivi Atmel, ed in particolare il chip ATA8520, integrano un TX a basso

consumo sia in trasmissione (appena 32,7 mA) che in modalità OFF (appena 5 nA); il chip può funzionare con una tensione compresa tra 1,9 e 3,2 V ed è in grado di erogare una potenza RF di +14,5 dBm. Inoltre il dispositivo integra lo stack SigFox, l'ID, il PAC ed è predisposto per un sistema crittografico AES in grado di garantire la massima sicurezza da questo punto di vista. La comunicazione avviene attraverso una porta SPI e il chip richiede circa 10 componenti esterni per funzionare. Nella classica configurazione questo dispositivo viene controllato da un micro AVR ad 8 bit (ATmega328P). Fin dalla sua nascita, la rete ed il protocollo SigFox sono

stati supportati dai dispositivi prodotti dalla società francese Telecom Design che attualmente ha a catalogo i moduli TD1204, TD1205 e TD1208, gateway e ricetrasmittitori operanti sulla frequenza europea ISM 868 MHz. A questi prodotti si sono aggiunti i dispositivi Telit (LE51-868S, un gateway SigFox a 868 MHz) e il ricetrasmittitore Silicon Labs Si446x in grado di operare tra 142 e 1.050 MHz e quindi di funzionare sia sulla frequenza europea ISM 868 MHz che su quella americana di 915 MHz. A tali frequenze di lavoro, questo compatto chip presenta una sensibilità di -127 dBm ed una potenza massima di

+16/+20 dBm (50-100 mW). A fianco della SigFox è recentemente scesa in campo Atmel che ha sviluppato il proprio SoC ATA8520 che ha ottenuto di recente l'omologazione SigFox. Il chip è in grado di operare a frequenze comprese tra 315 e 915 MHz e può essere controllato tramite interfaccia SPI. Il dispositivo Atmel, -che contiene lo stack SigFox e il relativo identificativo univoco a 32 bit, oltre al protocollo di sicurezza AES- è in grado di assicurare una portata compresa tra 15 e 30 km, grazie all'elevata sensibilità del ricevitore e alla potenza di uscita in trasmissione, che è di +14,5 dBm.



M0+ a 32 bit; la nostra board ha anche i connettori laterali standard Arduino Zero, però a differenza della Zero, il micro che dà vita all'SmartEverything è il SAMD21J, che ha un numero maggiore di I/O, il che è utile per comandare e leggere le periferiche a bordo della scheda. Il micro Atmel SAMD21 dispone internamente, nella SmartEverything, di 256 kB di memoria Flash di sistema, di 32 kB di

SRAM principale e di 12 kB di SRAM Low Power. Altra differenza con la Zero, che verrà meglio esposta più avanti in questo articolo, è l'assenza dell'EDBG a bordo.

L'HARDWARE VISTO DA VICINO

La scheda SmartEverything attualmente prodotta e chiamata Fox, utilizza la soluzione di connettività della rete wireless globale SigFox per realizzare

l'accesso all'IoT; allo scopo contiene un modulo Telit SigFox, oltre a un ricevitore GPS con antenna embedded finalizzato alla localizzazione dell'unità. Le versioni Lion e Wolf che verranno, avranno i link rispettivamente LoRa e WiFi.

La scheda inoltre utilizza il software di device management M2M di Telit per facilitare un collegamento rapido ed efficiente dei dispositivi al Cloud. La

Il link giusto per l'IoT

L'impiego di oggetti connessi non è certo una novità, ma la crescita del comparto sta accelerando rapidamente: si calcola che l'IoT genererà ricavi per 1200 miliardi entro il 2020 rispetto ai 200 miliardi di oggi. Gli oggetti collegati sono spesso molto semplici, isolati e funzionanti a batteria, con sensori che rilevano determinati eventi o informazioni una decina o un centinaio di volte al giorno e li inviano ad un sistema informatico centralizzato. Le informazioni possono riguardare qualsiasi cosa, dal consumo di energia, alla temperatura, umidità, posizione, presenze, dati sanitari e tanto altro ancora.

La sfida che i fornitori di soluzioni di connettività di rete devono affrontare sta nella loro capacità di fornire soluzioni adeguate a questi prodotti, che spesso le tecnologie esistenti non possono fornire, sotto l'aspetto sia dei consumi energetici, sia dei costi. Oggigiorno risulta evidente che le applicazioni legate all'Internet of Things hanno requisiti molto diversi rispetto ai cellulari ed agli smartphone; in quest'ultimo caso si procede verso la strada di una maggior larghezza di banda ed una maggior capacità di elaborazione a scapito della durata della batteria e di costi che risultano in ogni caso elevati, improponibili per infrastrutture con miliardi di oggetti connessi.

L'altra possibilità, quella della connet-

tività satellitare, presenta gli stessi problemi con costi elevati e consumi energetici incompatibili con lo sviluppo di diffuse reti low-cost. Anche le soluzioni di connettività a corto raggio (Wi-Fi, ZigBee, ecc.) presentano un elevato consumo energetico e sono molto complesse da gestire. La connettività Wi-Fi, per esempio, richiede la configurazione di ogni oggetto mentre infrastrutture come ZigBee richiedono un elevato numero di concentratori, con conseguente complessità di installazione e manutenzione e persino maggior consumo energetico. In conclusione, dunque, si torna a quelle che sono le caratteristiche della rete SigFox e più in generale dei sistemi "low power long range" o LPWA (Low Power Wide Area) che risultano i più adatti ad un rapido sviluppo di infrastrutture IoT. Tecnicamente anche un sistema cellulare GSM/GPRS/LTE potrebbe essere definito long range, ma sicuramente non è low power. Senza considerare il fatto che al contrario dei sistemi GSM e satellitari, i dispositivi LPWA operano su frequenze radio libere e non soggette a concessione governativa. Il rapido dispiegamento di soluzioni destinate all'Internet of Things richiede dunque infrastrutture dedicate con elevata portata ma con consumi ridotti, dell'ordine (in trasmissione) di pochi milliwatt, per garantire molti anni di funzionamento senza manutenzione di alcun tipo.

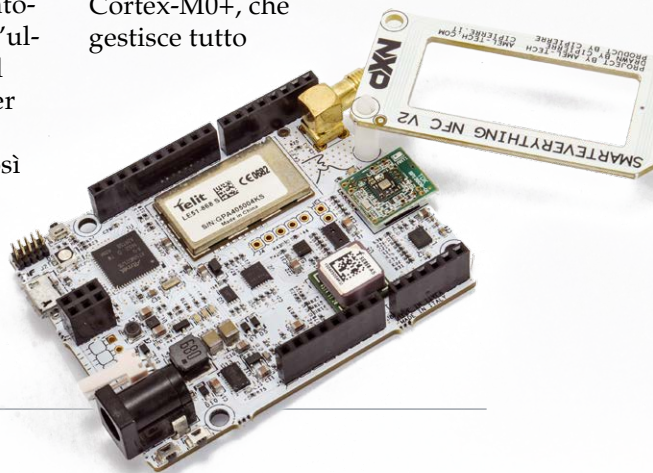
tecnologia di collegamento long-range SigFox è del tipo radio a banda strettissima UNB (Ultra Narrow Band) per la connessione dei dispositivi a una rete globale. L'UNB richiede pochissima energia e i dispositivi UNB operano sulle bande ISM disponibili a livello globale, che non richiedono alcuna licenza. Le funzionalità di questa scheda di sviluppo comprendono i sensori di prossimità, umidità, temperatura e accelerazione di STMicroelectronics i cui limiti di accelerazione possono essere selezionati per step di $\pm 2/4/8/16$ g. L'accelerometro

è ST LSM9DS1 triassiale che misura l'accelerazione lineare. Tutti i sensori comunicano sullo stesso canale I²C, che è lo stesso di Arduino Zero.

Il sensore di prossimità è un ST VL6180X che è insieme rilevatore di prossimità e luce; quest'ultimo misura con precisione il tempo che la luce impiega per viaggiare verso l'oggetto più vicino e tornare al sensore così da calcolarne la distanza. Combina un diodo emettitore IR, un circuito di misura e un sensore di luce in un solo package. Il sensore di umidità e

temperatura è un ST HTS221 e fornisce i dati di misura attraverso un canale digitale seriale direttamente al processore Atmel; l'elemento sensibile è un condensatore planare a dielettrico polimerico che rileva le variazioni di umidità (la misura è tra 0 e 100% RH) con una sensibilità di 0,004% rH/LSB e un'accuratezza di $\pm 4,5\%$ rH, tra 20 e $+80\%$ di rH. Per la misura della temperatura, il range è tra -40 e 120 °C e l'accuratezza di $\pm 0,5$ °C tra 15 e $+40$ °C.

Il trasduttore di pressione è sempre ST: si tratta di un LPS25H con elemento piezoresistivo dotato di logica a interfaccia I²C-Bus. L'elemento sensibile è una membrana sospesa all'interno di un singolo substrato di silicio. Il trasduttore rileva pressioni da 260 a 1.260 hPa con risoluzione di 1 Pa RMS e può sopportare sovrappressioni fino a 20 volte. La scheda dispone di un'interfaccia a basso consumo TDK Bluetooth per collegamenti short-range, che ne consente l'impostazione e il controllo da uno smartphone, oltre a un tag NXP NFC con interfaccia seriale I²C per l'autenticazione (infatti nella foto si vede con applicata la scheda bobina per l'NFC inserita nei connettori dedicati). La scheda SmartEverything comprende anche una CPU USB "Host orchestrator chip" di Atmel basata sull'ARM Cortex-M0+, che gestisce tutto



il traffico tra le periferiche. La crypto-autenticazione basata sul chip Atmel consente di implementare un completo algoritmo di sicurezza SHA-256 hash con MAC (Message Authentication Code). Sono compresi anche un'antenna Dynaflex a 868 MHz e i dispositivi di power management Linear Technology.

La scheda Fox nasce in un contesto low power: non è stato previsto il WiFi perché non è un sottosistema a basso consumo; tutta la componentistica usata è low-power perché l'obiettivo è ottenere dalla scheda grande autonomia con piccole batterie. Pur essendo di base una Arduino Zero, la Smart Everything non dispone dell'EDBG (*Atmel's Embedded Debugger*) dell'Atmel, che fornisce un'interfaccia per il debug che evita il ricorso a software aggiuntivi; l'EDBG offre anche il supporto per le porte COM virtuali per il bootloader della scheda e la sua programmazione. L'EDBG è un componente che fa da debugger a bordo della scheda, usato su Arduino Zero e sulle schede Atmel Xplained; la scelta di non montare l'EDBG dipende da varie considerazioni, la prima delle quali è che in produzione non ha senso avere un componente in più, che viene usato solo in fase di sviluppo. Inoltre l'EDBG è un componente realizzato da Atmel per le proprie schede Xplained Pro e per Arduino, ma non viene rilasciato per produttori terzi.

Il prezzo che si paga per non avere l'EDBG è che bisogna utilizzare un programmatore esterno nel momento in cui si vuole impiegare Atmel Studio come debugger e quando si vuole aggiornare il bootloader, perché l'EDBG dell'Arduino Zero, per quanto riguarda l'IDE Arduino, si utilizza solamente

per programmare il bootloader; una volta che il bootloader è a bordo, l'IDE Arduino comunica attraverso la porta USB nativa e a quel punto la SmartEverything si usa come una Arduino Zero, nel senso che anche se il bootloader viene programmato con l'EDBG nel caso di Arduino Zero e con il programmatore esterno Atmel Ice nel nostro caso, una volta che il bootloader è a bordo della scheda l'EDBG non serve più. Quindi la scheda si programma tramite la USB nativa esattamente come si fa con l'Arduino Zero e le altre schede Arduino che hanno la USB.

I GPIO sugli header standard Arduino sono gli stessi eccetto due, che non è stato possibile rimappare per come sono multiplexate le periferiche (è stato utilizzato un UART in più rispetto alla Zero) e quindi ci sono due I/O in più comunque gestiti dall'apposita libreria.

IL SOFTWARE

Sul lato della programmazione va detto che uno dei frutti della stretta collaborazione con il progetto Arduino è che questa scheda è una delle primissime

schede che sono già implementate all'interno del nuovo IDE il 1.6.5, che ha la capacità di integrare tutta una serie di terze parti. SmartEverything è la prima scheda di terze parti già integrata.

I creatori di SmartEverything hanno voluto integrare la scheda nell'IDE 1.6.5 per rendere facilmente utilizzabile la board; come ambiente di sviluppo era era disponibile Atmel Studio, che è ricco, completo e gratuito, ma per poter utilizzare la scheda con Atmel Studio era necessario un programmatore della Atmel, problema comunque risolvibile perché sono disponibili in commercio al costo di 40÷50 euro ottimi programmatori. Per quanto questa soluzione fosse accessibile, rimaneva il problema che andava bene per i maker "di fascia alta", i più evoluti, ma per quelli di livello medio e i principianti il programmatore Atmel e Atmel Studio non erano lo strumento ideale. Seguendo la filosofia Arduino, si è quindi pensato di fornire agli utenti un IDE semplificato e capace di rendere trasparenti all'utente molti pas-

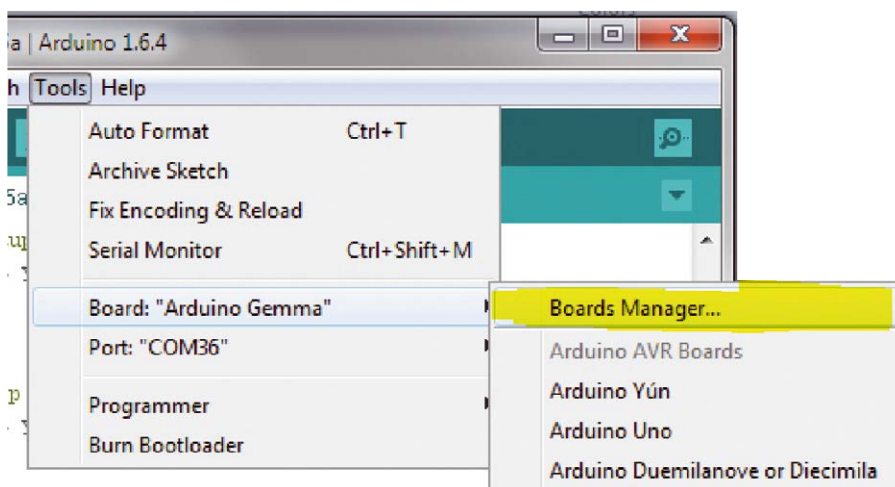


Fig. 1 - Selezione della scheda da Boards Manager dell'IDE.

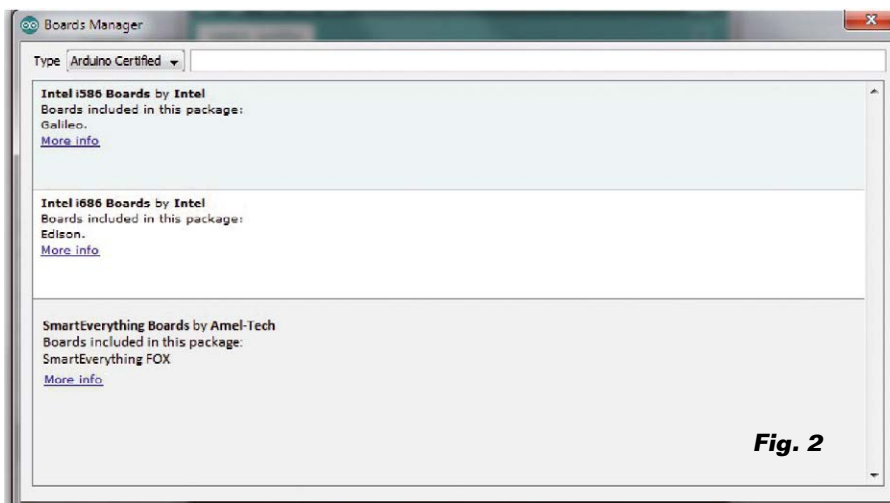


Fig. 2

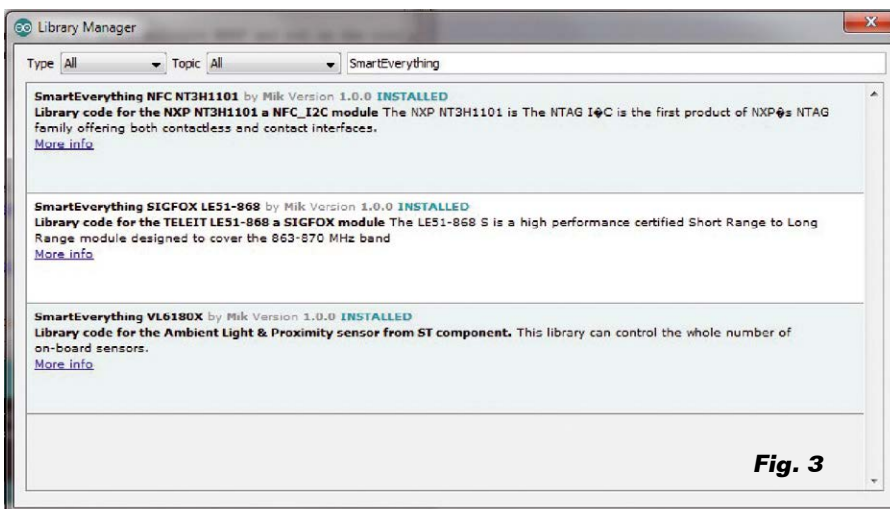


Fig. 3

Fatto ciò, si accede all'omonima finestra dalla quale si seleziona la scheda (Fig. 2).

Per quanto riguarda le librerie, attualmente per la Smart Everything Fox sono disponibili le seguenti:

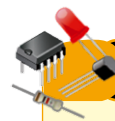
- SmartEverything NFC NT3H1101 (comunicazione con NFC);
- SmartEverything SIGFOX LE51-868 (comunicazione wireless SigFox);
- SmartEverything VL6180X (sensore di prossimità).

Altre sono in preparazione e saranno disponibili e costantemente aggiornate sul sito web della scheda; gli unici componenti della Smart Everything per cui ancora non sono disponibili le librerie sono Crypto e BLE.

Per includere le librerie per la SmartEverything, una volta aperto l'IDE si entra nel menu Sketch, quindi si impartisce il comando Include Library e poi Manage Libraries.... Aperto Library Manager, dalla relativa finestra si sceglie la libreria richiesta (Fig. 3).

saggi. Sfruttando la caratteristica del nuovo IDE Arduino di accettare schede di terze parti, i creatori di SmartEverything hanno sviluppato le librerie per integrare la scheda all'interno dell'IDE Arduino 1.6.5 e poi per sviluppare le librerie necessarie a utilizzare in modo semplice tutta la sensoristica onboard. Quindi ogni componente all'interno della scheda ha una libreria con diversi esempi in perfetto stile Arduino, che permette l'uso tramite ogni sketch; siccome le schede arduino vengono integrate in Atmel Studio, la scheda SmartEverything sarà

programmabile anche in Atmel Studio in due modi diversi: come una piattaforma SAMD21 partendo da scratch e utilizzando una libreria di Atmel, oppure in modo più semplice, utilizzandola come se fosse una scheda Arduino, quindi usando delle librerie che sono quelle che vengono dal mondo Arduino. Il primo passo da compiere una volta installato l'IDE è installare il core della Zero; il secondo è caricare la scheda SmartEverything; allo scopo (Fig. 1) si deve aprire il menu Tools e impartire il comando Boards -> Boards Manager...



per il MATERIALE

Tutto il materiale utilizzato in questo progetto è disponibile presso Futura Elettronica. La scheda smart everything fox (cod. MCS7561BN) è in vendita a Euro 199,00, il modulo bluetooth per smarteverything (cod. MCS7561BLE) costa Euro 25,00. 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>

Un nuovo supertelescopio per scoprire la vita nell'Universo

genere umano", ha affermato il presidente di Aura Matt Mountain.

Il telescopio sarà utilizzato da astronomi, scienziati e ricercatori per migliorare le conoscenze sull'origine e l'evoluzione dell'universo, fornendo, secondo gli auspici, evidenze sulla presenza della vita al di fuori della Terra. Hdst sarà dotato di un coronografo, disco che blocca la luce della stella

L'High-Definition Space Telescope, Hdst, è il nuovo supertelescopio spaziale gigante, progettato e costruito per scoprire se esiste la vita nell'Universo. Il programma è stato annunciato dall'Aura (Association of Universities for Research in Astronomy) di Washington, al fine di mettere in orbita prima del 2030 un telescopio con

un potere risolutivo mai raggiunto da nessuno strumento finora.

Il telescopio potrà osservare gli oggetti cosmici con una nitidezza cinque volte superiore a quella di Hubble e con una sensibilità cento volte maggiore. "Finalmente utilizzeremo un potente mezzo per realizzare scoperte eccezionali che cambieranno per sempre il

madre in modo da poter visualizzare direttamente la luce riflessa degli esopianeti orbitanti e di uno spettrografo che analizza la composizione atmosferica dei pianeti extraterrestri.

Hdst avrà grandi dimensioni, si prevede infatti la realizzazione di uno specchio primario di almeno dodici metri di diametro.

www.hdstvision.org

I Memristor sfidano le Flash Memory

In attesa dell'arrivo sul mercato delle memorie 3D XPoint di Intel, proseguono gli studi per trovare alternative alla tecnologia del silicio che, secondo alcuni, è ormai al capolinea. L'ultimo pretendente al trono è il cosiddetto memristor, oggetto degli studi dei ricercatori del Politecnico di Zurigo che sono riusciti a realizzare memristor di perovskite di soli 5 nanometri di spessore con tre stati resistivi stabili, il che significa che è possibile memorizzare i dati come 0,1 e 2, ovvero con un "trit" anziché un "bit". La ricerca, che è stata pubblicata sulla rivista ACS Nano, ha portato allo sviluppo di sistemi di commutazione non volatili

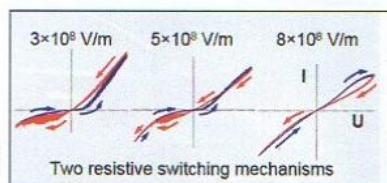
che possono essere attivati mediante un impulso di tensione.

"I dispositivi da noi messi a punto potrebbero essere utilizzati per una Information Technology di nuovo tipo, che non si basa soltanto sulla logica binaria ma che prevede l'informazione che si trova tra lo zero e l'uno," ha

affermato Jennifer Rupp, professoressa del Dipartimento dei Materiali al Politecnico di Zurigo. *"Tutto questo presenta interessanti implicazioni per quella che viene definita come logica fuzzy, che introduce una forma di incertezza nel trattamento delle informazioni digitali. Potremmo chiamarla informatica più soft."*

I ricercatori ritengono che questa capacità potrebbe essere adatta a quella che viene definita "neuromorphic computing", ovvero la simulazione dei neuroni del cervello umano.

<http://pubs.acs.org>



ExoMars alla scoperta dei segreti del Pianeta Rosso

Le nuove scoperte su Marte riaccendono l'interesse per il Pianeta, che sarà al centro della missione ExoMars realizzata con l'obiettivo principale di acquisire e dimostrare la capacità autonoma europea di eseguire un atterraggio controllato sulla superficie marziana, operare sul suolo marziano in mobilità di superficie, accedere al sottosuolo per prelevare campioni e analizzarli in loco. La missione, programmata per il 2016, indagherà sulle tracce di vita passata e presente su

Marte, la caratterizzazione geochemica del pianeta, la conoscenza dell'ambiente marziano e dei suoi aspetti geofisici e l'identificazione dei possibili rischi per le future missioni umane. L'ESA ha assegnato all'industria italiana la leadership principale di entrambe le missioni. Oltre alla responsabilità complessiva di sistema di tutti gli elementi, l'Italia dirige anche lo sviluppo del modulo di discesa di ExoMars 2016, denominato Schiaparelli, il "Drill" di due metri che perforerà il suolo marziano per il prelievo di campioni e il centro di controllo da cui verranno gestite le seguenti parti del Rover:

- DREAMS (Dust characterization, Risk assessment and Environment Analyser on the Martian Surface) suite di sensori per la misura dei parametri meteorologici (pressione, temperatura, umidità, velocità e direzione del vento, radiazione solare) e del campo elettrico atmosferico in prossimità della superficie di Marte. Esso avrà la possibilità di studiare le condizioni ambientali



La Tesla Model S diventa autonoma (o quasi)

Con l'ultimo upgrade del software (v.7.0) rilasciato da Tesla Motors, la Model S diventa una vettura quasi autonoma: elaborando i dati provenienti da telecamere, radar, sensore ad ultrasuoni e GPS, la vettura riesce ad orientarsi automaticamente lungo il percorso selezionato, cambiare corsia e regolare la velocità in relazione al traffico; una volta arrivati a destinazione, la Model S è in grado di identificare un parcheggio libero e parcheggiare da sola. L'ultima versione del software di Tesla consente al guidatore di prendere confidenza con le tecnologie di guida autonoma e di valutare la capacità del software di evitare pericoli; pur non potendo fare scomparire il traffico, la Model S così configurata può ridurre il carico di lavoro del conducente e rendere più facile, sicura e piacevole la guida.

Insieme con le nuove funzionalità di guida autonoma, l'innovativo cruscotto della vettura visualizza le informazioni che il software utilizza in tempo reale per determinare il comportamento del veicolo. Il pannello degli strumenti mostra un'im-

agine della strada sulla base dei dati rilevati dai sensori della vettura, fornendo al guidatore le stesse informazioni che la vettura utilizza per districarsi nel traffico; tra queste, la segnaletica orizzontale della carreggiata, la presenza di ostacoli o vetture nell'angolo cieco, quello della massima velocità consentita, l'allarme di collisione, il sistema di regolazione della velocità e il controllo della direzione. La Model S è in grado di mantenere la vettura nella corsia prescelta adeguando la velocità alle condizioni del traffico; per cambiare corsia è sufficiente azio-

nare l'indicatore di direzione: la Model S si sposterà automaticamente nella corsia prescelta quando le condizioni del traffico lo consentiranno.

La vettura è anche in grado di rilevare ostacoli presenti di lato, tipicamente vetture troppo vicine.

Infine la Model S è in grado di rilevare posti auto liberi e parcheggiare, facendosi carico anche di manovre complesse e difficili, controllando lo sterzo e la velocità del veicolo.

www.teslamotors.com



sul pianeta nel periodo, particolarmente interessante, in cui si prevede una forte presenza di polveri nell'atmosfera (Francesca Esposito INAF/Osservatorio Astronomico di Napoli);

- AMELIA (Atmospheric Mars Entry and Landing Investigation and Analysis), modellistica dell'atmosfera marziana impiegando i dati raccolti dai sensori durante la discesa del lander Schiaparelli sulla superficie marziana (Francesca Ferri CISAS- Università di Padova);
- MA_MISS (Mars Multispectral Imager for Subsurface Studies) spettrometro per l'analisi dell'evoluzione geologica e biologica del sottosuolo marziano, inserito all'interno del "Drill", che consentirà di analizzare la conformazione della superficie interna della perforazione effettuata dal "Drill" stesso (Maria Cristina De Sanctis (INAF/IAPS, Roma, SELEX GALILEO).

www.esa.int

Teletrasporto record: particella trasferita a 100 chilometri di distanza con la fibra ottica

Record nel "teletrasporto": l'informazione contenuta in una particella di luce è stata infatti trasferita attraverso una fibra ottica a 100 chilometri di distanza, ben quattro volte più lontano rispetto al record precedente. Il risultato, che apre importanti scenari sulla possibilità di creare una rete Internet super veloce basata sulla fisica quantistica, è stato pubblicato sulla rivista *Optica* dall'Istituto statunitense per gli Standard e la Tecnologia (Nist). Il teletrasporto effettuato nell'esperimento consta nel trasferimento delle proprietà fisiche di una particella di luce (fotone). I ricercatori le hanno trasmesse ad un

altro fotone attraverso 102 chilometri di fibra ottica. Precedenti esperimenti avevano teletrasportato fotoni su distanze superiori, ma attraversando l'aria, mentre l'utilizzo delle fibre ottiche convenzionali offre la possibilità di creare reti più flessibili.

La nuova tecnica di teletrasporto del Nist evita gli effetti dispersivi di dati della fibra ottica e potrebbe essere usata per produrre ripetitori quantici che rimbalzano periodicamente i dati consentendo una maggiore estensione della rete.

www.nist.gov

Eutelsat e Facebook collaborano per migliorare la connettività in Africa

Facebook ha lanciato in collaborazione con Eutelsat un progetto per migliorare ed estendere la connettività in Africa.

Il progetto nasce da un accordo con Spacem per usare un payload a banda larga del prossimo satellite AMOS-6.

L'iniziativa consentirà di coprire l'Africa sub-sahariana fornendo connettività ad una fascia di popolazione finora esclusa dal servizio.

"Facebook mira a connettere il mondo e crediamo che i satelliti abbiano un ruolo importante nell'abbattere i principali ostacoli che impediscono di connettere i popoli dell'Africa. Collaboreremo con Eutelsat a questo progetto per cercare nuovi modi di utilizzare i satelliti per connettere le persone nelle zone più remote del mondo, in modo più efficiente", ha spiegato Chris Daniels, vice presidente di internet.org.

Il sistema favorirà l'accesso a Internet individuale o collettivo, con costi accessibili.

Michel de Rosen, CEO e presidente di Eutelsat ha dichiarato: "Siamo entusiasti per

tale possibilità di fornire il servizio di accesso a Internet in Africa. Abbiamo importanti precedenti nella gestione di sistemi di satelliti ad alto rendimento (hts) e faremo in modo di fornire soluzioni Internet

affidabili a prezzi competitivi, in modo che sempre più utenti possano navigare e far parte della società dell'informazione".

<http://news.eutelsat.com>





>



3

SVILUPPARE APP android con Processing

dell'ing.
MIRCO SEGATELLO

Impariamo a utilizzare i sensori integrati nel nostro smartphone come l'accelerometro, il sensore di prossimità e quello di illuminazione ambiente, ecc. Terza puntata.

Sebbene conosciamo l'importanza della programmazione e le dedichiamo ampio spazio, la nostra è una rivista di elettronica quindi abbiamo deciso di focalizzare questo corso, oltre che sull'aspetto software, anche su quello hardware. Quindi in questa terza puntata ci occuperemo dell'utilizzo dei sensori presenti all'interno del nostro smartphone. Anche se potrebbe sembrare complicata, in realtà la gestione dei sensori è alquanto semplice ed una volta che vi sarete impadroniti della tecnica di lettura dei dati, potrete integrare queste periferiche hardware all'interno delle vostre applicazioni. Andando a vedere con attenzione le caratteristiche del nostro smartphone Samsung Galaxy S4, utilizzato per testare gli esempi proposti, scopriamo che i sensori disponibili sono: Accelerometro, Magnetometro, Giroscopio,

RGB light, Barometro, Prossimità, Gestuale, Temperatura & Umidità.

Cominciamo con il focalizzare la nostra attenzione sull'accelerometro, ossia quel sensore che -detto in parole povere- permette di orientare lo schermo in automatico ogni qualvolta ruotiamo il nostro smartphone. Si tratta di un sensore MEMS (acronimo di Micro Electro Mechanical Systems) che semplicemente misura l'accelerazione durante lo spostamento del vostro smartphone, ma anche a dispositivo immobile la forza di gravità agisce su di esso. Il sensore integra sia una parte elettronica che una parte meccanica; in altre parole potremmo immaginarlo costituito da una piccola massa (realizzata con materiale conduttivo) mantenuta sospesa all'interno del sensore da elementi elastici; un apposito circuito elettronico rileva la capacità

Listato 1

```

/* ES7 accelerometro e magnetometro */
// Based on sketch of Eric Pavey

// Imports per usare i sensori
import android.content.Context;
import android.hardware.Sensor;
import android.hardware.SensorEvent;
import android.hardware.SensorManager;
import android.hardware.SensorEventListener;

// Setup variabili
SensorManager mSensorManager;
MySensorEventListener accSensorEventListener;
MySensorEventListener magSensorEventListener;
Sensor acc_sensor;
float[] acc_values;
Sensor mag_sensor;
float[] mag_values;

void setup() {
  orientation(LANDSCAPE);
  fill(200);
  textSize(40);
}

void draw() {
  background(0);
  if (acc_values != null) {
    text(("Accelerometro: " + acc_values[0] + " " + acc_values[1] + " " + acc_values[2]), 10, 50);
  }
  else {
    text("Accelerometro: non disponibile", 10, 50);
  }
  if (mag_values != null) {
    text(("Magnetometro: " + mag_values[0] + " " + mag_values[1] + " " + mag_values[2]), 10, 100);
  }
  else {
    text("Magnetometro: non disponibile", 10, 100);
  }
}

// Override the parent (super) Activity class:
// States onCreate(), onStart(), and onStop() aren't called by the sketch. Processing is entered at
// the 'onResume()' state, and exits at the 'onPause()' state, so just override them:

void onResume() {
  super.onResume();
  println("RESUMED! (Sketch Entered...)");

  // Build our SensorManager:
  mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
  // Build a SensorEventListener for each type of sensor:
  magSensorEventListener = new MySensorEventListener();
  accSensorEventListener = new MySensorEventListener();
  // Get each of our Sensors:
  acc_sensor = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
  mag_sensor = mSensorManager.getDefaultSensor(Sensor.TYPE_MAGNETIC_FIELD);
  // Register the SensorEventListeners with their Sensor, and their SensorManager:
  mSensorManager.registerListener(accSensorEventListener, acc_sensor, SensorManager.SENSOR_DELAY_GAME);
  mSensorManager.registerListener(magSensorEventListener, mag_sensor, SensorManager.SENSOR_DELAY_GAME);
}

void onPause() {
  // Unregister all of our SensorEventListeners upon exit:
  mSensorManager.unregisterListener(accSensorEventListener);
  mSensorManager.unregisterListener(magSensorEventListener);
  println("PAUSED! (Sketch Exited...)");
  super.onPause();
}

// Setup our SensorEventListener
class MySensorEventListener implements SensorEventListener {
  void onSensorChanged(SensorEvent event) {
    int eventType = event.sensor.getType();
    if (eventType == Sensor.TYPE_ACCELEROMETER) {
      acc_values = event.values;
    }
    else if (eventType == Sensor.TYPE_MAGNETIC_FIELD) {
      mag_values = event.values;
    }
  }
  void onAccuracyChanged(Sensor sensor, int accuracy) {
    // do nuthin'...
  }
}

```


tra le armature del condensatore formato dalla massa flottante e dalla struttura fissa del sensore. Sottoponendo il sensore a delle accelerazioni (o all'attrazione della forza di gravità) la piccola massa flottante tenderà a spostarsi modificando la distanza tra le armature del condensatore; dalla variazione di capacità è possibile determinare lo spostamento della massa sospesa e quindi l'accelerazione impressa. Il tutto ovviamente avverrà lungo i tre assi X, Y e Z, potendo in questo modo determinare i movimenti del dispositivo in ogni direzione. Una curiosità: in caso di caduta libera dello smartphone, si verificherà l'unica situazione in cui nessuna forza agisce su di esso e l'accelerometro fornirà zero per ciascun asse.

Questa particolare situazione viene rilevata dal chip a bordo dell'accelerometro, fornendo prontamente segnalazione alla CPU principale, la quale, preparandosi alla caduta (ed eventuale rottura) provvederà a salvare i dati più importanti. Il giroscopio è un sensore che rileva la rotazione nei tre assi X, Y e Z dell'oggetto nel quale è installato; nel caso dello smartphone, viene utilizzato insieme all'accelerometro per determinare in maniera più precisa e dettagliata a quali movimenti è sottoposto l'apparato. Un altro sensore spesso presente nei nostri dispositivi "mobile" è il sensore di campo

magnetico terrestre, chiamato magnetometro, utilizzato per la funzione bussola e per l'orientamento automatico delle mappe.

Il sensore di prossimità è stato introdotto negli smartphone touch screen per disabilitare il display durante le telefonate, onde evitare che il contatto con l'orecchio possa attivare funzioni o comporre numeri indesiderati; entra in funzione quando si avvicina il telefonino al volto. Il sensore di luce è molto utile per regolare l'intensità dell'illuminazione del display in maniera del tutto automatica, ottimizzando i consumi della retroilluminazione del display. A livello di programmazione possiamo appoggiarci alle funzioni interne delle API; anche se la stesura dello sketch non è immediata, prendiamo spunto da alcuni esempi disponibili sul sito di Processing. Con il **Listato 1** visualizziamo sul display i valori grezzi (raw values) dei tre assi dell'accelerometro e dei tre assi del magnetometro; per l'accelerometro noterete dei numeri molto "ballerini" a causa del forte rumore intrinseco del sensore, mentre il magnetometro -pur fornendo valori più stabili- risulta molto influenzato da corpi ferromagnetici nelle vicinanze. Abbiamo testato questa applicazione (**Fig. 1**) direttamente sul dispositivo, perché non sarebbe molto significativo avviarla nell'emulatore in quanto non è

Accelerometer:
x: +0,384
y: -0,160
z: +9,494

Fig. 1 - Screenshot dell'applicazione del Listato 1.

Accelerometer :
x: +0,08
y: +0,13
z: +9,47
MagneticField :
x: +16,53
y: -7,63
z: -40,92
Light Sensor : 1.0
Proximity Sensor : 8.0

Fig. 2 - Screenshot dell'esempio MultipleSensors della libreria ketai.

Angle: 69°



Fig. 3 - Screenshot dell'applicazione del Listato 3.

Listato 2

```
/* ES8 uso libreria Ketai */
import ketai.sensors.*;

KetaiSensor sensor;
float accelerometerX, accelerometerY, accelerometerZ;

void setup()
{
  sensor = new KetaiSensor(this);
  sensor.start();
  orientation(LANDSCAPE);
  textAlign(CENTER, CENTER);
  textSize(50);
}

void draw()
{
  background(78, 93, 75);
  text("Accelerometer: \n" +
    "x: " + nfp(accelerometerX, 1, 3) + "\n" +
    "y: " + nfp(accelerometerY, 1, 3) + "\n" +
    "z: " + nfp(accelerometerZ, 1, 3), 0, 0, width, height);
}

void onAccelerometerEvent(float x, float y, float z)
{
  accelerometerX = x;
  accelerometerY = y;
  accelerometerZ = z;
}

boxColor=false;
}
```

in grado di fornire la lettura dei sensori. Analogamente a come abbiamo fatto per questo esempio, troverete in rete esempi specifici per ciascun tipo di sensore, alcuni interessanti che potete trovare a questo indirizzo: <https://github.com/jeffThompson/ProcessingTeachingSketches/tree/master/Android>.

Volendo semplificarci la vita, possiamo ricorrere ad un'apposita libreria che ci viene in aiuto nella stesura di applicazioni che utilizzano i sensori del nostro dispositivo; la libreria si chiama *Ketai Sensor Library for Android* ed è disponibile all'indirizzo web: <https://code.google.com/p/ketai/>. Questa libreria consente di implementare l'utilizzo di tutti i sensori presenti nel nostro dispositivo oltre ad altre importanti funzioni che utilizzeremo più avanti nel corso; per la sua installazione fate riferimento alle indicazioni riportate nelle scorse puntate. Apriamo l'esempio d'uso della libreria chiamato *accelerometer*, disponibile tra i file della libreria (**Listato 2**) e avviamolo sul dispositivo; in esso viene importata la libreria Ketai. Giusto per farvi vedere in pratica l'utilità della libreria Ketai, abbiamo provato a scrivere un'applicazione chiamata *Bussola*, utilizzando la lettura del magnetometro (**Listato 3**) e importando Ketai. Una semplice immagine reperita in Internet (l'abbiamo utilizzata come icona) ed alcune righe di codice sono state sufficienti per realizzare un'app utile e dall'aspetto gradevole

(Fig. 3). La libreria Ketai è una delle più complete e offre notevoli spunti per applicazioni anche più sofisticate come ad esempio quelle che utilizzano il modulo NFC, ma altrettanto interessante la possibilità di utilizzare SQLite. Spesso è necessario dover salvare una piccola quantità di dati in modo persistente all'interno del nostro dispositivo; ad esempio è possibile chiedere all'utente alcune preferenze di funzionamento, che ritroverà al successivo avvio dell'applicazione. Potremmo chiedere all'utente la lingua in uso piuttosto che il colore di fondo, e salvare questi dati per poi recuperarli al successivo avvio. Un'applicazione può creare il suo database privato SQLite, sul quale ha il controllo completo, ed utilizzarlo per scrivere e leggere i dati secondo necessità. La libreria Ketai può tornarci utile anche nello sviluppo dell'interfaccia grafica, per la quale mette a disposizione alcuni importanti strumenti. ■

Listato 3

```
/* ES9 bussola */
import ketai.sensors.*;

KetaiSensor sensor;
PVector magneticField;

float angle;
PImage img;

void setup()
{
  sensor = new KetaiSensor(this);
  sensor.start();
  sensor.list();

  magneticField = new PVector();
  orientation(PORTRAIT);
  textSize(80);

  angle=0.0;
  img=loadImage("bussola-antica.jpg");
  imageMode(CENTER);
}

void draw()
{
  background(0);
  float Y=magneticField.y;
  float X=magneticField.x;
  angle= -180*atan2(Y,X)/PI;
  //text("x:" + X + " y:" + Y, 0, 50);
  textAlign(CENTER);
  text("Angle: " + int(angle) + "°", width/2, 300);
  translate(width/2, height/2);
  rotate(angle*TWO_PI/360);
  image(img,0,0);
}

void onMagneticFieldEvent(float x, float y, float
long time, int accuracy)
{
  magneticField.set(x, y, z);
}
}
```


World Solar Challenge 2015: primo e secondo posto per i team olandesi

Dopo cinque giorni di gara e 3.000 chilometri percorsi utilizzando esclusivamente l'energia del sole, il team olandese Nuon Solar ha preceduto di quasi 10 minuti sul traguardo di Adelaide, Solar Team Twente, l'altra squadra olandese in gara, aggiudicandosi per la seconda volta consecutiva il primo posto nella classifica della World Solar Challenge. Il team Nuon Solar, dell'Università di Delft, aveva già vinto la precedente edizione, quella del 2013. Sin dalla partenza le due squadre si sono alternate alla testa della gara che quest'anno ha visto la presenza di ben 47 team provenienti da 25 nazioni. Nella prima giornata di gara, tuttavia, dopo un iniziale testa a testa, i due team erano scivolati al terzo e quarto

posto, superati dalla vettura Aurum dell'Università del Michigan.

Durante la seconda giornata il duo olandese riprendeva la testa, e nell'ultima frazione di gara Nuon difendeva il proprio vantaggio sia su Red One che su Aurum, chiudendo al primo posto con un tempo complessivo di 37 ore, 56 minuti e 56 secondi, ad una media oraria di circa 80 km/h.

www.worldsolarchallenge.org



Crescent Dunes: energia elettrica anche di notte

Sta per diventare operativo, nel Nevada, il primo impianto solare termico di grandi dimensioni con sistema di accumulo in grado di fornire energia elettrica per 10 ore dopo il tramonto del sole.

Si tratta dell'impianto da 110 MWp di Crescent Dunes, vicino a Tonopah, in Nevada, con 17.000 specchi che concentrano i raggi del sole su una torre alta 165 metri, simile a quella di Ivanpah, in California. Tuttavia, mentre nel caso di Ivanpah e di altri impianti simili, i raggi del sole riscaldano un serbatoio di vapore che aziona le turbine, l'impianto di Crescent Dunes riscalda una miscela di sali fusi che viene immagazzinata in serbatoi coibentati ed utilizzata prevalentemente di

notte o quando vi è maggior richiesta di elettricità. In pratica questo impianto funziona come una centrale a gas che può rapidamente essere messo in funzione fornendo il massimo dell'energia o spento altrettanto rapidamente. Nell'impianto la temperatura dei sali fusi arriva a 566 °C migliorando il rendimento complessivo rispetto ad altre soluzioni che funzionano a 400 °C. Per il futuro, SolareReserve, la società proprietaria dell'impianto, ha allo studio soluzioni in ceramica in grado di funzionare a 732°C con un ulteriore miglioramento del rendimento. Secondo la società, il problema

più complesso riguarda non tanto il sistema di accumulo quanto il posizionamento dei 17.000 specchi che vengono spostati ogni 60 secondi per garantire la massima concentrazione dei raggi solari. Impianti di questo tipo sono già stati realizzati in passato ma con potenze molto più basse, non in grado di offrire le economie di scala di Crescent Dunes. Attualmente il costo dell'energia prodotta è di circa 190 dollari al Megawatt ma la società conta di scendere a 125 dollari (senza contributi governativi) con il prossimo impianto di questo tipo in fase di costruzione in Sud Africa.

www.solarreserve.com



Rectenna ottica per aumentare l'efficienza delle celle solari

Ricercatori del Georgia Institute of Technology hanno sviluppato un prototipo perfettamente funzionante di rectenna ottica, un dispositivo che converte le onde elettromagnetiche in corrente continua combinando le funzionalità di un'antenna e di un raddrizzatore a diodi. L'antenna messa a punto è in grado di convertire anche le onde elettromagnetiche nello spettro del visibile, caratteristica che potrebbe portare a una nuova generazione di celle solari molto più efficienti.

Nessuno era mai riuscito a fare funzionare questi dispositivi, noti da più di 50 anni, a lunghezze d'onda ottiche, a causa delle dimensioni particolarmente ridotte che avrebbe dovuto avere gli elementi dell'antenna che inglobano i raddrizza-

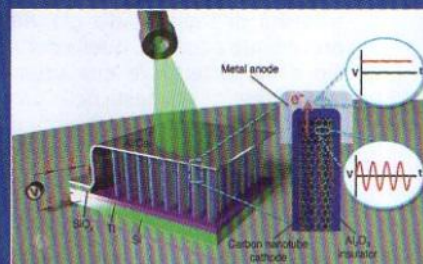
tori. Nella ricerca pubblicata sulla rivista *Nature Nanotechnology*, i ricercatori del Georgia Tech sono stati in grado di superare questa sfida grazie ad una struttura con nanotubi di carbonio a parete multipla i quali fungono da antenne per catturare la luce; ciascuno dei nanotubi ingloba sottilissimi rettificatori in grado di convertire le oscillazioni prodotte dalle onde luminose in corrente continua.

"Sulla base della nostra esperienza e dei calcoli teorici, credo che con questi dispositivi sia possibile realizzare celle solari con un'efficienza superiore al 40%" ha dichiarato Baratunde Cola, professore associato presso la Georgia Tech.

Attualmente l'efficienza di conversione energetica di questi dispositivi è ancora bassa, circa l'uno per cento, che va bene

per rivelatori ad alta temperatura, ma non per le comuni celle solari. Tuttavia, i ricercatori ritengono che con alcune tecniche di ottimizzazione allo studio, entro un anno si potranno produrre dispositivi commercialmente validi.

www.news.gatech.edu



AtlantikSolar, l'aereo solare progettato dagli studenti dell'ETH

Gli studenti di ingegneria del Politecnico di Zurigo hanno realizzato un aereo solare senza pilota che percorrerà 400 chilometri, sorvolando la foresta amazzonica, tentando così di battere il record mondiale di un aeromobile alimentato con l'energia fotovoltaica, dopo aver già battuto il primato di volo continuo per aeromobili solari inferiori ai 50 chilogrammi, con una tratta di ben 81,5 ore.

Il velivolo solare UAV, che pesa solamente 6,8 chilogrammi, chiamato AtlantikSolar, tenterà una nuova impresa decollando da Belem, in Brasile. Per l'occasione

l'aereo solare sarà equipaggiato con una serie di telecamere e sensori atmosferici. AtlantikSolar è munito di batterie ad alta densità energetica e moduli fotovoltaici. Di giorno l'energia viene parzialmente accumulata nelle batterie per poi essere utilizzata anche nelle ore notturne.

"Abbiamo già costruito un paio di velivoli solari e siamo in fase di test. Il nostro prossimo obiettivo è il volo a bassa quota, per mappare il terreno con le telecamere e costruire mappe in 3D", ha spiegato Roland Siegwart, docente di sistemi autonomi al Politecnico di Zurigo.

Il velivolo solare senza pilota potrà essere impiegato anche per monitorare zone colpite da catastrofi e per ispezionare luoghi a rischio, garantendo l'incolumità degli esseri umani.

www.atlantiksolar.ethz.ch



Wattway, la strada fotovoltaica francese

L'azienda Colas realizzerà in Francia una strada fotovoltaica, chiamata Wattway, basata su una tecnologia che rende i pannelli fotovoltaici indistruttibili, grazie ad una speciale resina che ricopre

il rivestimento sul quale transitano i veicoli. L'energia viene prodotta, senza consumare ulteriore spazio, sfruttando le superfici già occupate. Le celle fotovoltaiche sono utilizzate per fabbricare un rivestimento ricoperto da una speciale resina, che protegge il sistema, senza influire negativamente sulla circolazione dei veicoli su strada.

L'energia prodotta viaggia poi sotto terra per essere immessa in

rete. La commercializzazione dei pannelli Wattway inizierà all'inizio del 2016 in Francia, per poi espandersi anche all'estero, in particolare nel Nord America. Al momento la strategia di vendita di Wattway si concentra sulla proposta di installare lo speciale "pavimento" su superfici limitate, anche solo qualche centinaio di metri quadrati, per valutarne i benefici, visto che con soli 15 metri quadrati di superficie fotovoltaica si riesce a produrre energia sufficiente ad alimentare un semaforo. Da Colas fanno sapere che rivestendo il 2,5% dell'intera superficie stradale francese sarebbe possibile coprire il 10% del fabbisogno energetico nazionale.

www.wattwaybycolas.com



Storm Pulse: la moto elettrica con una autonomia di 380 km



Trenta studenti della University of Technology di Eindhoven hanno lavorato alla realizzazione di un motore elettrico da destinare ad una motocicletta che vanta un'autonomia di 380 chilometri. Per promuovere il veicolo elettrico è in programma nel 2016 un tour di 26.000 chilometri attorno al globo per testare e verificare su strada le potenzialità e gli eventuali problemi della moto elettrica. Storm Pulse utilizza un nuovo tipo di batteria modulare, divisa in 24 speciali cartucce per una capacità massima complessiva di 28,5 kWh.

La peculiarità di funzionamento di questo veicolo risiede nella possibilità di scegliere la quantità di moduli da utilizzare in base alle proprie esigenze, intervenendo personalmente sui moduli in maniera molto semplice e intuitiva. Il progetto dei 26mila chilometri dello Storm World Tour sarà realizzato, secondo gli auspici, in soli 80 giorni per poi essere riproposto l'anno successivo con un tour ancora più impegnativo di 40.000 chilometri.

www.tue.nl/en/

