

In omaggio
SOLUZIONI EMBEDDED CONTROL 16-BIT
MICROCHIP

Mensile di progettazione elettronica, attualità scientifica, novità tecnologiche.

€ 6,00

Elettronica In

oltre l'elettronica

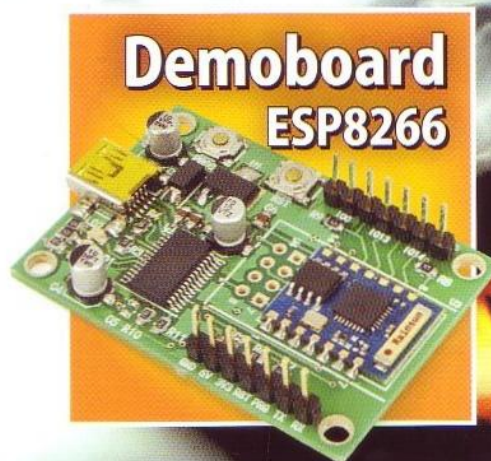
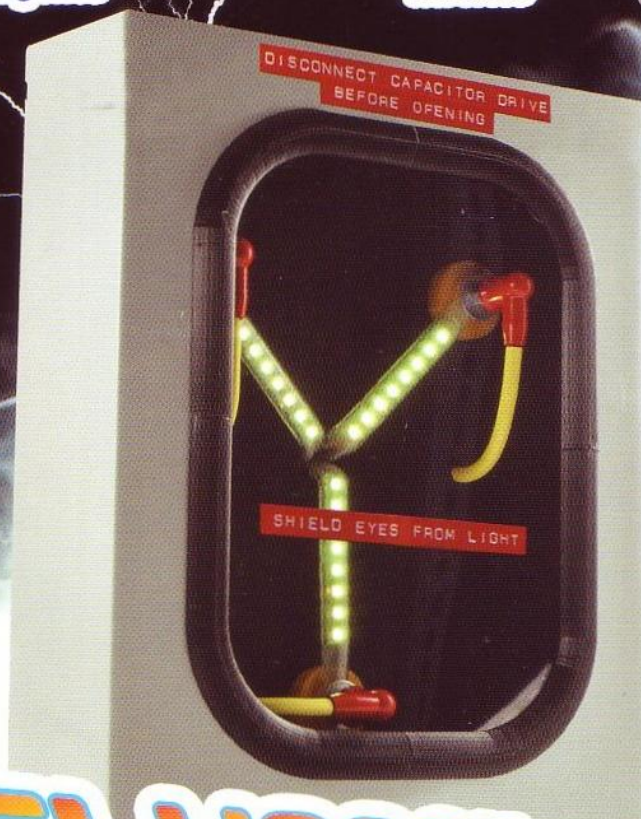
www.elettronica.in.it

Anno XXI - n. 199
Ottobre 2015

■ Smartem,
il termostato
intelligente

■ Fishino,
l'Arduino diventa
wireless

■ Win10:
proviamolo con Arduino



FLUSSO CANALIZZATORE

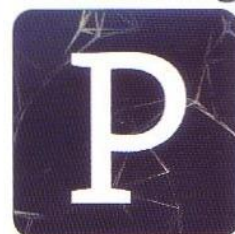
■ Wi-Lamp,
la lampada Wi-Fi

■ L'IoT si programma
in Python con Viper

■ LoRa: lo shield
RF-Long Range

■ Domotica Open-Source
con i moduli UDOiT

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
Estero 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.D.I.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

FabLab e scuole, un'occasione da non perdere

Oltre che essere un paese di santi, poeti, eroi e navigatori, l'Italia è anche un paese di makers e di FabLab. Lo evidenziano i numeri del portale ufficiale del movimento - fablabs.io - gestito dalla FabFoundation, nonché quelli del primo censimento su questo fenomeno condotto da Massimo Menichinelli e Alessandro Rannellucci per conto della Fondazione Make in Italy CdB (scaricabile dal sito della Fondazione stessa). Numeri che evidenziano come il nostro paese sia al secondo posto al mondo per laboratori attivi con circa (ad oggi) un centinaio di FabLab.

Laboratori dove migliaia di persone, associazioni di vario genere, scuole, enti locali e aziende stanno creando qualcosa di completamente nuovo: luoghi attrezzati con macchinari di fabbricazione digitale (stampanti 3D, macchine per il taglio laser, fresatrici, strumentazione elettronica e altro) dove chiunque può realizzare i propri progetti ma anche imparare ad utilizzare queste macchine ed avvicinarsi al mondo della fabbricazione digitale. Luoghi in cui si impara e si insegna, dove ci si confronta e si collabora, con ricadute sul territorio che non sono ancora quantificabili in considerazione dell'ancora troppo breve periodo di attività. Il dato che emerge dalla ricerca è quello di una grande varietà di contesti, modalità e obiettivi, con modelli di business anche molto distanti tra loro e con una precarietà (dal punto di vista economico) piuttosto accentuata.

Insomma, a distanza di tre anni dalla nascita del primo FabLab italiano, molti laboratori sono ancora alla ricerca di un ruolo preciso e di una propria identità. Da questo punto di vista (ma anche da quello finanziario) una grande occasione potrebbe venire dal recente stanziamento di ben 45 milioni di euro da parte del MIUR (il Ministero dell'Istruzione) a favore della creazione di nuovi laboratori per l'innovazione e il lavoro, entità di cui dovrebbero fare parte scuole, enti locali e

aziende/associazioni del territorio. Una nuova generazione di laboratori aperti anche in orario extra scolastico, pensati per essere palestre di innovazione e spazi dove mettere in campo attività di orientamento al lavoro e di alternanza, ma anche progetti contro la dispersione scolastica e per il recupero dei Neet, i giovani non inseriti in percorsi di studio né nel mondo del lavoro. Nel bando si chiarisce che la valutazione dei progetti terrà conto in particolare della capacità di favorire il rapporto con il mondo del lavoro e diffondere le nuove competenze, fra cui quelle digitali.

Sembra quasi che nelle sue dichiarazioni il Ministro Giannini descriva un FabLab: "L'apertura al territorio sarà uno dei caratteri fondamentali di questi laboratori che potranno essere realizzati anche in spazi esterni alle scuole e saranno attivi oltre l'orario scolastico. Saranno luoghi dove i nostri ragazzi potranno scoprire i loro talenti e le loro vocazioni attraverso l'acquisizione di competenze trasversali, conoscenze pratiche e attraverso l'educazione all'autoimprenditorialità. Stiamo costruendo una risposta concreta al tema della disoccupazione giovanile e alla dispersione. Mettiamo in mano agli studenti gli strumenti per orientarsi al lavoro e per crearlo loro stessi con una didattica che guarda ai settori strategici del Made in Italy, legata alla vocazione produttiva, sociale e culturale di ciascun territorio".

Che dire? Sicuramente un'occasione preziosa per lo sviluppo e il consolidamento dei FabLab, in un contesto più strutturato e con obiettivi e finanziamenti certi.

Insomma, un'occasione da non perdere.



Arsenio Spadoni

(Arsenio.Spadoni@elettronica.in.it)

Sommario

Ottobre 2015
n. 199



IN COPERTINA:
Il flusso canalizzatore di
"Ritorno al Futuro"
in un'interpretazione
moderna basata sulla scheda
Arduino.

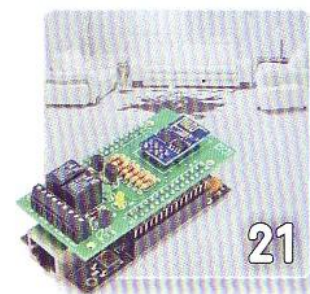
ELENCO INSERZIONISTI

- Fiera di Erba
- Fiera di Genova
- Fiera di Scandiano
- Fiera di Verona
- Fortronic
- Futura Elettronica
- Maker Faire
- Microchip
- National Instruments
- Rimini Mini Maker Faire
- RM Elettronica
- SMAU
- www.mdsrl.it

ARTICOLI

21 DOMOTICA **SMARTEM, IL TERMOSTATO INTELLIGENTE**

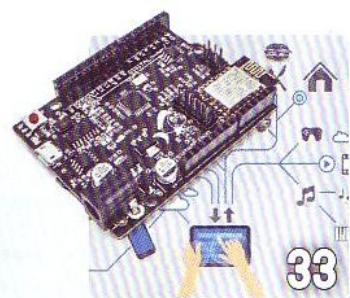
Impariamo a utilizzare e programmare il nuovo modulo EM1001, l'ultimo nato in casa Tibbo, attraverso un progetto pratico di home automation: il termostato smart. Prima puntata.



21

33 ARDUINO **FISHINO, L'ARDUINO DIVENTA WIRELESS**

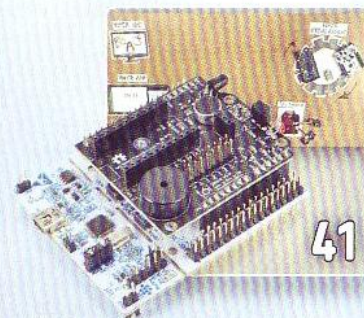
Proseguiamo nella presentazione della board Fishino, descrivendo le principali funzioni delle librerie e proponendo esempi d'uso. Seconda puntata.



33

41 PROGRAMMAZIONE **L'IoT SI PROGRAMMA IN PYTHON CON VIPER**

VIPER è una suite software per la programmazione di oggetti interattivi, pronti per il cloud e l'Internet of Things. VIPER permette di sviluppare in Python sulle più diffuse piattaforme di prototipazione, usando i paradigmi e le feature tipici della programmazione ad alto livello.



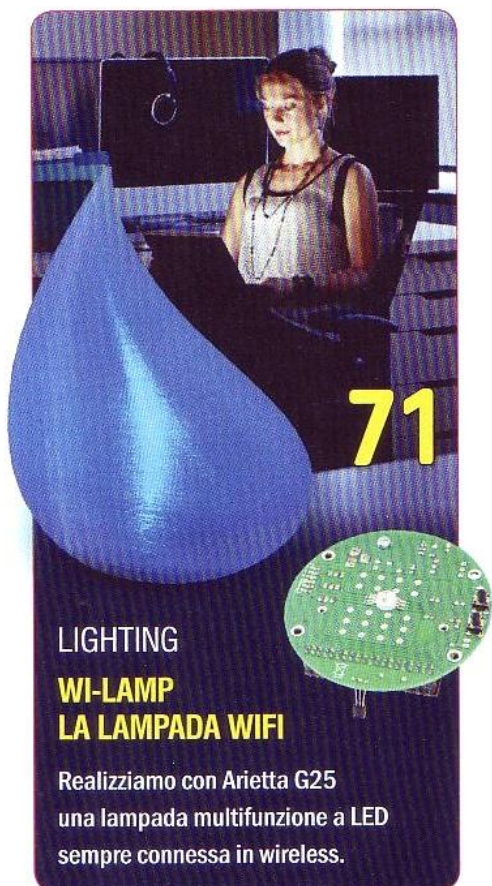
41

53 WIRELESS **LoRa SHIELD**

Dotiamo Arduino di un modulo per comunicazioni long-range basato sulla tecnologia Semtech. Prima puntata.



53



71

LIGHTING
WI-LAMP
LA LAMPADA WIFI

Realizziamo con Arietta G25 una lampada multifunzione a LED sempre connessa in wireless.

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
- 12 COMPONENTI & SISTEMI
- 15 EVENTI
- 17 APPUNTAMENTI
- 111 SCIENZA & TECNOLOGIA
- 125 RINNOVABILI

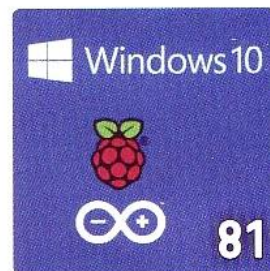
64 GADGET FLUSSO CANALIZZATORE

Riproduciamo in chiave moderna l'apparato che nel film RITORNO AL FUTURO consentiva di viaggiare nel tempo.



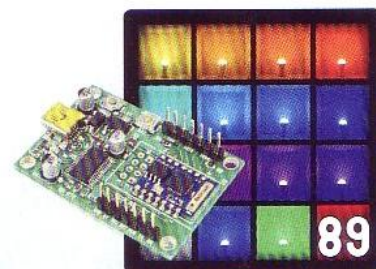
81 ARDUINO RELE' GEOLOCALIZZATO CON VIRTUAL SHIELD

Sperimentiamo lo strumento Virtual Shield per Arduino presente nell'ultimo sistema operativo Microsoft, per realizzare un'applicazione di geo-fence basata su smartphone, per attivare un relé quando il telefono si trova in una certa zona.



89 WIRELESS DEMOBOARD ESP-03

Sperimentiamo e prototipiamo applicazioni con la board wireless basata sul modulo ESP8266.



101 DOMOTICA DOMOTICA OPEN SOURCE CON I MODULI UDOIT

Facciamo conoscenza con il telecomando e l'attuatore relé della soluzione di domotica Open Source di UDOIT basata sui moduli RFTide ed introduciamo l'utilizzo delle "Scene".



115 DIDATTICA CORSO: SVILUPPARE APP ANDROID CON PROCESSING

Come realizzare app native per dispositivi Android tramite l'ambiente di sviluppo Processing. In questa puntata iniziale installiamo l'ambiente di lavoro e realizziamo la nostra prima app. Prima puntata.



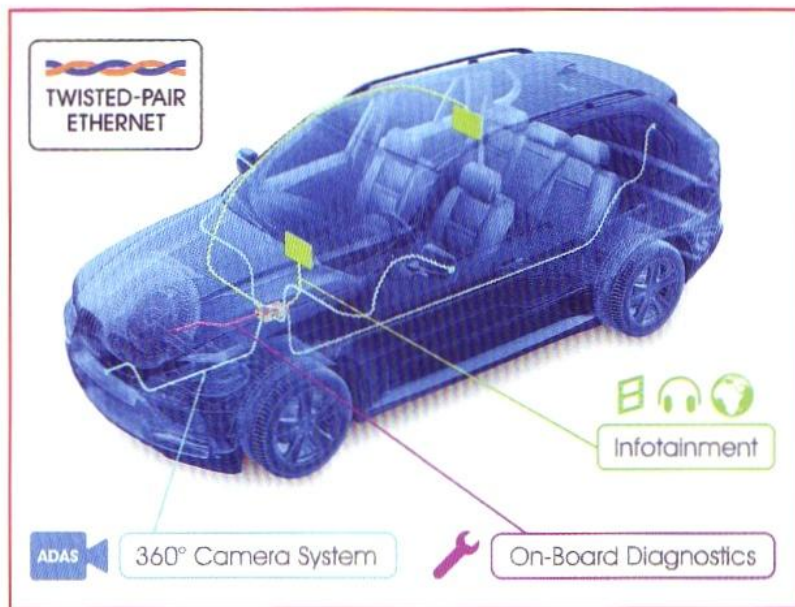
Queste pagine sono dedicate alle richieste, ai suggerimenti ed alle segnalazioni dei lettori. Raccomandiamo, per quanto possibile, di proporre argomenti di interesse generale. Contattateci numerosi all'indirizzo: redazione@elettronica.in.it

Tecnologia Ethernet anche in auto

Come mai nelle automobili vengono utilizzati bus CAN e LIN anziché la più performante tecnologia Ethernet?

Mario Moreschi - Roma

Le tecnologie CAN, LIN e FlexRay si sono affermate in campo automobilistico quando ancora nessuno parlava di automobili "connesse". Oggi questi bus controllano il funzionamento di milioni di automobili collegando tra loro attuatori, sensori e centraline di controllo. Con le prime applicazioni di infotainment audio e video, ma soprattutto con le esigenze tecnologiche legate allo sviluppo di vetture a guida autonoma, la banda di questi bus - dell'ordine dei megabit al secondo - è apparsa subito insufficiente, spingendo verso l'adozione di nuove soluzioni. E quale tecnologia poteva garantire velocità, sicurezza e integrazione se non quella Ethernet? Per tutte queste nuove applicazioni è infatti necessario che la vettura disponga di una rete in grado di operare alla velocità di almeno 100 Mb al secondo. In futuro la tecnologia Ethernet fornirà la dorsale di rete per i veicoli a guida autonoma e le vetture connesse, grazie all'elevata banda passante, alla velocità di comunicazione, al peso ridotto e all'ottimo rapporto prezzo/prestazioni, tutti requisiti indispensabili per queste applicazioni. Su questo fronte sono impegnati da tempo numerosi Produttori di semiconduttori, alcuni dei quali (NXP, Freescale e Broadcom) si sono alleati per dare vita all'Open Alliance Group, con lo scopo di rendere la tecnologia Ethernet adatta anche ai severi requisiti per l'impiego automobilistico. Proprio di recente NXP ha reso disponibili una serie di prodotti basati sullo standard BroadR-Reach, un physical layer Ethernet definito dall'Open Alliance Group, adatto all'impiego automobilistico: i transceiver Ethernet (famiglia TJA1100) e gli switch Ethernet (famiglia SJA1105). L'approccio modulare utilizzato da NXP con switch



e transceiver consente combinazioni flessibili, efficienti e convenienti, consentendo alle Case automobilistiche di creare soluzioni ottimali per una vasta gamma di architetture di rete - dalle utilitarie alle automobili di lusso al top della gamma.

Questo consentirà anche di spianare la strada in futuro a nuove architetture distribuite di rete e a nuove soluzioni di networking.

Il transceiver supporta modalità automotive a basso consumo: quando il motore è spento il sistema va in modalità sleep; tuttavia il sistema resta alimentato e ritorna attivo quando viene rilevata attività sulla rete.

Lo switch utilizza tecnologia Ethernet deterministica (Deterministic Ethernet Technology) per ridurre la latenza dei messaggi, indispensabile in applicazioni quali la guida autonoma, in cui la comunicazione deterministica è vitale per ragioni di efficienza operativa e sicurezza. Questa tecnologia recepisce l'attuale tendenza all'aumento della larghezza di banda, fino a un gigabit, garantendo nel contempo una elevata affidabilità dei sistemi di controllo collegati in rete ed elevata capacità anche in caso di guasti; il sistema si compone di diversi standard tra cui Ethernet (IEEE 802.3), Ethernet

Time-Triggered (SAE AS6802), Audio Video Bridging (AVB) e Time-Sensitive Networking (TSN).

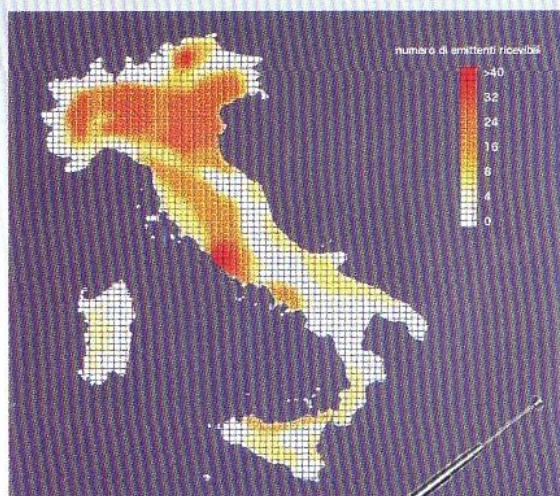
È opinione comune che la tecnologia Ethernet non sostituirà quelle CAN, LIN e FlexRay ma si integrerà con esse garantendo una dorsale per tutte le applicazioni ad elevata velocità.

Se la radio digitale non si collega a Internet

Ho acquistato una radio digitale con l'intenzione di ricevere i programmi radio disponibili su Internet anziché utilizzare il mio PC. Con mia sorpresa ho constatato che ciò non è possibile e le stazioni che ricevo sono relativamente poche e quasi tutte italiane...

Marco Luraghi - Milano

Sicuramente la radio che hai acquistato non è adatta a collegarsi, tramite Wi-Fi, alla rete Internet e ricevere le migliaia di stazioni radio commerciali e le web radio che - da tutto il mondo - trasmettono in streaming utilizzando la rete. Sicuramente l'evoluzione



Copertura della Radio Digitale nel nostro paese aggiornata a giugno 2014.

Radio Digitale DR315 in grado di ricevere le emissioni FM con RDS, quelle DAB/DAB+ e le web radio che trasmettono in streaming utilizzando Internet.



tecnologica in questo campo e nel settore del broadcast ha generato un po' di confusione e spesso, il termine "radio digitale" viene utilizzato in modo improprio. Vediamo dunque, di fare un po' di chiarezza. Per "Digital Radio" o "Radio Digitale" si intende l'evoluzione della radio analogica FM; si tratta di un ricevitore radio vero e proprio ma che opera con tecnologia digitale utilizzando il Digital Audio Broadcasting, meglio conosciuto col suo acronimo DAB e DAB+. Un'evoluzione simile a quella della TV analogica diventata recentemente DVB-T. La trasmissione avviene sulla banda III

(174-240 MHz) e sulla banda L (1452 - 1467,5 MHz). Come nella TV analogica, i canali radiofonici utilizzano dei multiplex che li contengono; la banda III, quella utilizzata attualmente nel nostro paese, è composta da 38 canali, identificati da numeri (5...13) e blocchi (A...F) mentre la banda L terrestre ha 9 canali (LA, LB, LC... LI). La ricezione avviene in maniera completamente automatica, l'apparecchio scansiona le bande e propone una lista di emittenti disponibili tra le quali scegliere. Esiste ovviamente anche la possibilità di memorizzare le stazioni preferite.

La radio digitale, oltre ad offrire dati e servizi, permette la diffusione dei programmi audio con ottima qualità e senza alcuna interferenza o disturbo. Questo elevatissimo livello sonoro è percepibile anche in auto. Per quanto riguarda la codifica audio, attualmente in Italia viene utilizzato lo standard DAB+ che utilizza un codec più efficiente (HE-AAC+, famiglia MPEG4) che, tra l'altro, consente il trasporto di un numero maggiore di programmi all'interno di un singolo multiplex. La radio digitale, dunque, utilizza un ricevitore a radiofrequenza a tutti gli effetti e come tale necessita di una buona antenna. Dal momento che, al contrario della TV analogica, le trasmissioni della banda FM non verranno sopresse, quasi tutte le radio digitali DAB/DAB+ sono in grado di ricevere la banda di frequenza 88-108 MHz nella quale operano, e continueranno ad operare, le emittenti commerciali che tutti noi conosciamo.

Le stazioni FM utilizzano un sistema prettamente analogico integrato con qualche informazione digitale per la quale viene utilizzato lo standard RDS (Radio Data System); questa tecnologia consente di inviare diversi tipi di informazioni tra cui l'identificativo della stazione radio, aggiornamenti sul traffico, eccetera. Queste informazioni consentono al ricevitore di essere più "smart", cercando, ad esempio, una frequenza alternativa nel momento in cui il segnale diventa troppo debole. Questa funzione è molto importante soprattutto per le autoradio. E veniamo ai dispositivi "Radio Internet". Anche in questo caso si tratta di ricevitori radio che però sono in grado di connettersi solamente ad una rete Wi-Fi (di cui debbono avere le credenziali). Naturalmente esiste anche la possibilità di connettersi via cavo al router Internet. Tutte le principali radio del mondo, oltre ad inviare il proprio segnale nell'etere, trasmettono in forma digitale il proprio

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

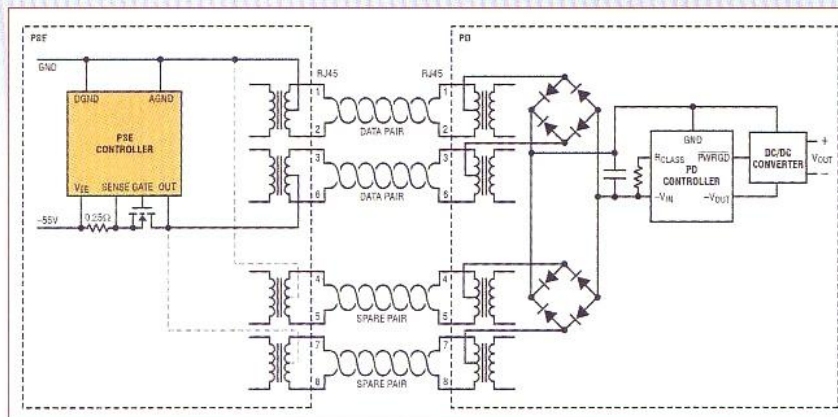
palinsesto attraverso Internet; in alcuni casi si tratta di radio tradizionali, ricevibili via etere in FM, che ampliano il proprio raggio di ascolto ripetendo le trasmissioni in linea; in altri casi si tratta di emittenti, le cosiddette web radio, che mettono a disposizione i propri programmi esclusivamente su Internet. Il flusso di dati – lo stream – può essere ricevuto con un'applicazione che funziona su PC o tablet o con un apparato specifico, la "Radio Internet", appunto. Naturalmente, come su PC, esiste la possibilità di selezionare e suddividere per genere, regione, ecc. le emittenti nonché di memorizzarne un certo numero (anche qualche centinaio). In commercio esistono apparati, genericamente identificati col termine Radio Digitali, che sono in grado di ricevere tutte queste tre categorie di emissioni, altri solamente due o una sola. Da quanto ci dici, il tuo ricevitore è in grado di ricevere solamente le emissioni DAB/DAB+ e quelle FM con RDS; per le web radio, dunque, dovrai continuare ad utilizzare il tuo PC o tablet.

Fino a 90 W con PoE++ di Linear Technology

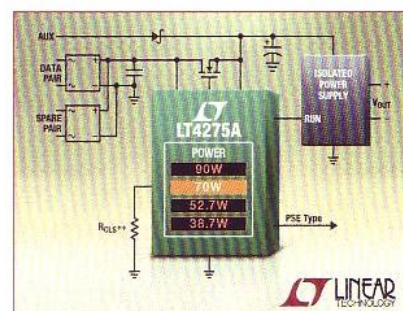
Il mese scorso in questa rubrica avete trattato l'argomento PoE senza citare la tecnologia PoE++ di Linear Technology con la quale è possibile trasferire fino a 90 watt...

Giovanni Regina – Milano

Nel rispondere al lettore abbiamo citato solamente le tecnologie PoE attualmente normate dall'IEEE, ovvero la IEEE 802.3af e la IEEE 802.3at; la tecnologia LTPoE++, nata nel 2011, pur mantenendo la compatibilità con i dispositivi IEEE, è



una tecnologia proprietaria di Linear Technology, produttore californiano specializzato in circuiti di alimentazione. Come puoi vedere nella tabella pubblicata, la tecnologia LTPoE++ è in grado di fornire 38,7W; 52,7W; 70W e, appunto, 90 watt su un cavo CAT5e a 4 coppie; LTPoE++ offre una soluzione "plug and play" valida e sicura, in grado di ridurre la complessità di progettazione dei PSE (Power Sourcing Equipment) e dei PD (Powered Device). Anche in questo caso prima di fornire alimentazione alla linea, il PSE deve verificare l'esistenza di una resistenza di segnetura. In tabella 1 sono evidenziate le differenze tra gli standard PoE, PoE+ e PoE++ mentre lo schema elettrico illustra una tipica configurazione PoE a 4 coppie. Attualmente l'LT4266 è il dispositivo più performante per quanto riguarda i controller PSE (power sourcing equipment) mentre tra i PD controller il modello più avanzato è l'LT4275 (vedi schema). Lo standard LTPoE++ utilizza uno schema di classificazione che consente ai controller PSE LTPoE++ e ai controller PD LTPoE++ di comunicare tra loro in modo



semplice, mantenendo l'interoperatività con le apparecchiature IEEE standard. A differenza dei controller PD tradizionali che integrano il MOSFET di potenza, l'LT4275 controlla un MOSFET esterno per ridurre in modo significativo la dissipazione generale di calore del PD e ottimizzare il rendimento, aspetto importante soprattutto ai livelli di potenza più elevati. Questo approccio innovativo consente agli utenti di dimensionare il MOSFET in base ai requisiti di riscaldamento ed efficienza specifici dell'applicazione, consentendo l'uso di MOSFET con RDS(ON) di soli 30 milliOhm. L'LT4275 riconosce un PSE sia come hardware di Tipo 1, conforme al livello di potenza di 13W dello standard 802.3af IEEE, sia come hardware di Tipo 2, conforme al livello di potenza di 25,5W dello standard IEEE 802.3at e sia come hardware LTPoE++ conforme ai livelli di potenza compresi tra 38,7W e 90W e trasmette la potenza in modo corrispondente. Per assegnare la potenza in modo efficiente, gli utenti dei dispositivi PD possono configurare una classificazione che rappresenta l'uso della potenza da parte del dispositivo PD. Un pin di potenza ausiliario programmabile con errore della "signature" offre supporto fino a 9V. L'LT4275 include inoltre un'uscita "power good", una resistenza on board di tipo signature, blocco della sottotensione e protezione termica completa.

DEVICE	PSE						
	STANDARD	802.3AT		LTPoE++			
	TYPE	TYPE 1	TYPE 2	38.7W	52.7W	70W	90W
PD	802.3AT	TYPE 1	13W	13W	13W	13W	13W
		TYPE 2	13W	25.5W	25.5W	25.5W	25.5W
	LTPoE++	38.7W	13W	25.5W	38.7W	38.7W	38.7W
		52.7W	13W	25.5W	—	52.7W	52.7W
		70W	13W	25.5W	—	—	70W
		90W	13W	25.5W	—	—	90W

LEVEL	NO-LOAD POWER REQUIREMENT	AVERAGE EFFICIENCY REQUIREMENT
I	used if you do not meet any of the criteria	
II	no criteria was ever established	no criteria was ever established
III	≤10 Watts: ≤0.5W of No Load Power 10~250 Watts: ≤0.75W No Load Power	≤1Watt: ≥ Power x 0.49 1~49 Watts: ≥[0.09 x Ln[Power]] + 0.49 49~250 Watts: ≥84%
IV	0~250 Watts: ≤0.5W No Load Power	≤1Watt: ≥ Power x 0.50 1~51 Watts: ≥[0.09 x Ln[Power]] + 0.5 51~250 Watts: ≥85%
V	Standard Voltage Ac-Dc Models (>6Vout)	
	0~49 Watts: ≤0.3W of No Load Power	≤1 Watt: 0.48 x Power + 0.140
	50~250 Watts: ≤0.5W of No Load Power	1~49 Watts: [0.0626 x Ln[Power]] + 0.622
		50~250 Watts: ≥87%
	Low Voltage Ac-Dc Models (<6Vout)	
	0~49 Watts: ≤0.3W of No Load Power	≤1 Watt: 0.497 x Power + 0.067
	1~49 Watts: [0.0750 x Ln[Power]] + 0.561	
	50~250 Watts: ≥86%	

Tabella 1 - Livelli energetici I ÷ V previsti dal Programma ENERGY STAR.

SINGLE-VOLTAGE EXTERNAL AC-DC POWER SUPPLY ¹ , BASIC-VOLTAGE		
Nameplate Output Power [P _{out}]	Minimum Average Efficiency in Active Mode [expressed as a decimal]	Maximum Power in No-Load Mode [W]
P _{out} ≤ 1 W	$0.5 \times P_{out} + 0.16$	0.100
1 W < P _{out} ≤ 49 W	$0.071 \times \ln(P_{out}) - 0.0014 \times P_{out} + 0.67$	0.100
49 W < P _{out} ≤ 250 W	0.880	0.210
P _{out} > 250 W	0.875	0.500

Tabella 2
Livello energetico VI previsto dal Programma ENERGY STAR.

World Voltage Guide

Mi capita spesso di recarmi all'estero, specialmente in Africa, e non so mai che tipo di presa e quale tensione di alimentazione troverò. Esiste una guida mondiale per questa mia esigenza?

Mauro Lucchini - Torino

L'efficienza energetica degli alimentatori

Tra i tanti simboli e indicazioni presenti sugli alimentatori esterni, non riesco a trovare il significato della scritta "EFFICIENCY LEVEL" seguita da un numero romano. Di cosa si tratta?

Giovanni Manoli - Napoli

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

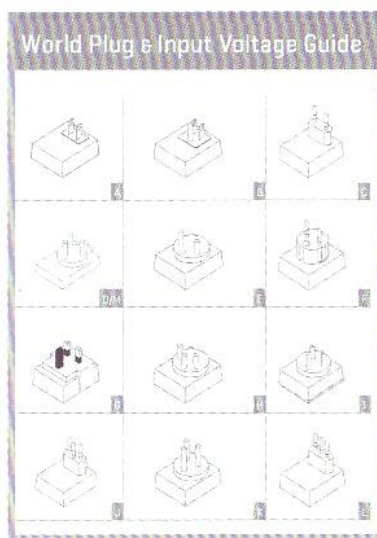
La scritta con l'indicazione del livello di efficienza che trovi nel tuo alimentatore fa riferimento al programma ENERGY STAR, un'iniziativa nata da una partnership tra il Dipartimento statunitense per l'energia (DOE) e l'Agenzia statunitense per la protezione dell'ambiente (EPA). Scopo del programma è quello di migliorare l'efficienza energetica degli alimentatori e degli adattatori di rete i quali consumano una discreta quantità di energia anche in assenza di carico.

I numeri romani (I-V) indicano l'efficienza energetica degli alimentatori; attualmente la maggior parte degli alimentatori certificati in commercio risponde ai requisiti delle classi IV e V (vedi tabella 1).

Recentemente è stata approvata una normativa ancora più stringente (tabella 2) identificata dal numero romano VI. In questo caso il consumo a riposo (nella categoria fino a 49 W) non può superare i 0,1 W (0,2 W fino a 250 W di uscita).

Negli Stati Uniti, a partire dal febbraio 2016, gli alimentatori dovranno rispettare obbligatoriamente i vincoli imposti dal livello VI, il cui assorbimento a riposo è 1/3 rispetto al passato.

Puoi trovare questa guida all'indirizzo <http://www.cui.com/catalog/resource/world-plug-and-input-voltage-guide.pdf> dove sono descritte le prese utilizzate in tutti i paesi del mondo nonché la tensione e la frequenza in uso.





Maggior velocità all'esecuzione del codice con LabVIEW 2015

È stata ufficialmente rilasciata la versione 2015 del software grafico di progettazione di sistemi LabVIEW. L'ultima versione di LabVIEW aggiunge velocità all'esecuzione del codice, semplifica lo sviluppo e implementa nuovi strumenti di debug. LabVIEW 2015 mantiene e accresce lo standard e il modo in cui gli utenti di LabVIEW in-

teragiscono con qualsiasi hardware attraverso il riutilizzo del medesimo codice e degli stessi processi, risparmiando tempo e denaro, consentendo di affrontare al meglio l'evoluzione tecnologica, i nuovi requisiti e le pressioni imposte dal mercato. LabVIEW 2015 migliora anche la curva di apprendimento per un approccio software-designed col

quale creare rapidamente sistemi potenti, flessibili e affidabili. Con tre differenti suite applicative che includono un anno di accesso illimitato ai corsi di formazione e certificazione, gli sviluppatori hanno accesso come non mai alle risorse software e di formazione per costruire più rapidamente sistemi migliori. www.ni.com

MAX9286, per visione a 360° nelle applicazioni ADAS

Maxim Integrated ha presentato il deserializzatore quadruplo MAX9286 che consente di progettare sistemi di visione panoramica a 360 gradi (surround view) per applicazioni ADAS (Advanced Driver Assistance Systems) caratterizzati da un minor numero di componenti e da un time to market più breve. Un singolo deserializzatore MAX9286 Gigabit Multimedia Serial Link (GMSL) è infatti in grado di ricevere e sincronizzare automaticamente i segnali video provenienti da un massimo di quattro telecamere.

Il MAX9286 fornisce i dati in uscita mediante un'interfaccia standard CSI-2 a quattro vie, rendendoli disponibili al processore video. Il dispositivo consente di ridurre lo spazio occupato sulla scheda ed il numero dei componenti, poiché sostituisce quattro deserializzatori discreti ed una FPGA. Questo deserializzatore ad alta integrazione permette inoltre di fare a meno del software aggiuntivo necessario per la sincronizzazione



delle telecamere. Il MAX9286 si presta in modo ottimale anche ai sistemi di visione industriali ed ai sistemi di telecamere 3D.

Il chip MAX9286 è disponibile in contenitori TQFN e QFN da 56 pin di 8 mm x 8 mm e può operare nella gamma di temperatura compresa tra -40 e +105 gradi Celsius. Sono disponibili esemplari certificati AEC-Q100. Un kit di valutazione dotato di processore SoC e progetti di riferimento specifici per applicazioni ADAS, sviluppati in collaborazione con partner dell'ecosistema, aiutano i progettisti nel loro lavoro.

www.maximintegrated.com

MLX91208CAV, sensore di corrente fino a 1000A da Melexis

Melexis ha aggiunto un altro dispositivo alla propria serie MLX91208 di sensori di corrente programmabili ad effetto Hall, con l'introduzione di una versione a corrente elevata.

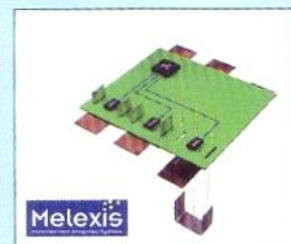
Ottimizzato per i campi molto elevati tipici delle applicazioni HEV (Hybrid and Electric Vehicle), il sensore MLX91208CAV può supportare una corrente primaria anche di 1000A. Questo dispositivo complementa i prodotti MLX91208 esistenti che coprono funzioni caratterizzate da bassi campi magnetici e da alti campi magnetici.

Grazie alla tecnologia proprietaria adottata, a concentratore magnetico integrato (IMC), i sensori della serie MLX91208 di prodotti qualificati AEC-Q100 possono misurare in modo accurato la corrente senza richiedere l'inclusione dei voluminosi nuclei ferromagnetici esterni

che sono necessari con i sensori di corrente ad effetto Hall convenzionali.

Basato sull'effetto Hall, il dispositivo MLX91208CAV fornisce un meccanismo di misura della corrente economicamente conveniente e completamente contactless che soddisfa i parametri richiesti. Questo dispositivo fornisce intrinsecamente l'isolamento galvanico imposto dalle applicazioni ad alta tensione, eliminando al contempo ulteriori sorgenti di perdite di potenza che si hanno con la tecnologia shunt.

www.melexis.com



LTC5549, mixer ad ampia larghezza di banda da 2GHz a 14GHz



Linear Technology ha presentato l'LTC5549, un mixer a doppio bilanciamento in grado di funzionare come convertitore verso l'alto o verso il basso in un ampio range di frequenza RF da 2GHz a 14GHz. L'LTC5549 offre un'eccellente linearità di 24,4dBm IIP3 a 9GHz. Consente di realizzare progetti di trasmettitori e ricevitori a microonde con un buffer LO integrato che richiede solo

un livello di segnale di 0dBm, eliminando in modo efficace un circuito amplificatore LO esterno ad alta potenza. L'LTC5549, inoltre, è dotato di un duplicatore di frequenza commutabile integrato su chip. L'LTC5549 utilizza un trasformatore balun integrato a banda larga, ottimizzato per estendere la larghezza di banda della frequenza RF da 2GHz a 14GHz, consentendo al

tempo stesso il funzionamento single-ended. Le prestazioni dell'LTC5549 migliorano una vasta gamma di applicazioni a microonde, tra cui microonde backhaul, stazioni base LTE-Advanced a banda elevata senza licenza, radio satellitari a banda larga, sistemi radar, transceiver a banda X e Ku, apparecchiature di test e modem satellitari.

www.linear.com

Atmel espande la linea di MCU ARM Cortex M0+ con la serie L22

Atmel ha annunciato la disponibilità della nuova serie di microcontrollori SAM L22 che amplia la popolare famiglia di MCU SMART SAM L basata su tecnologia ARM Cortex M0+ per applicazioni sicure e ultra low power. La famiglia Atmel SMART SAM L è al primo posto nella speciale classifica ULPBench (Ultra Low Power Bench) di EEMBC, società indipendente di valutazione. I nuovi microcontrollori combinano un consumo bassissimo con la tecnologia touch e sono in grado di controllare display LCD con un massimo di 320 segmenti, risultando particolarmente adatti in applicazioni a bassissimo consumo come termostati, contatori per acqua, gas, elettricità, dispositivi medicali portatili, domotica e controllo accessi. Un altro importante campo di applicazione riguarda l'Internet delle Cose (IoT), con tantissimi

dispositivi connessi alimentati a batteria che necessitano di componenti con limitato consumo e in grado di garantire la massima sicurezza dei dati. La serie SAM L22 offre funzioni di sicurezza avanzate grazie alla tecnologia AES a 256 bit, al controllo di ridondanza ciclico (CRC), ad un vero generatore numerico casuale ed alla protezione e controllo antimanomissione della memoria Flash che garantisce una memorizzazione ed una gestione dei dati particolarmente sicura.

Per ottenere il più basso consumo energetico possibile, i dispositivi della nuova serie utilizzano la tecnologia proprietaria picoPower di Atmel con le periferiche intelligenti a basso consumo che lavorano in maniera indipendente dalla CPU in modalità sleep.

www.atmel.com

One third the power of existing solutions
35µA/MHz in active mode and 200nA in sleep mode



F1358, demodulatore integrato con predistorsione digitale per reti TDD



IDT ha presentato oggi un nuovo dispositivo a radiofrequenza che riduce notevolmente i consumi energetici, il costo totale e l'area occupata sulla scheda dei sistemi elettronici utilizzati nelle stazioni radio base delle reti cellulari TDD (Time Division Duplexing). Funzionante su uno spettro di frequenze da 3200 MHz a 4000 MHz, il demodulatore integrato con predistorsione digitale (DPD) F1358 offre i più elevati li-

velli di linearità e densità di integrazione disponibili sul mercato, che permettono di migliorare le prestazioni della catena di linearizzazione del segnale nei trasmettitori con predistorsione digitale utilizzati nelle stazioni radio base delle reti cellulari.

Grazie all'attenuatore a passi digitali (DSA) e al commutatore a due vie (SP2T) integrati al suo interno, questo nuovo dispositivo, che si aggiunge alla famiglia di prodotti RF di IDT, si affianca ad altri due diffusi demodulatori ad alte prestazioni: il modello F1320, funzionante nella banda da 400 MHz e 1200 MHz, e il modello F1370, funzionante nella banda da 1300 MHz a 2900 MHz.

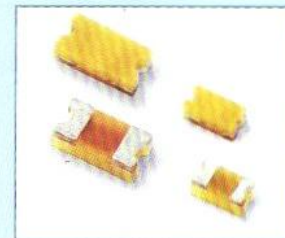
www.idt.com

Le scariche fino a 30 kV non sono più un problema

Littelfuse ha introdotto i soppressori ESD Serie XGD XTREME-GUARD, disponibili nelle misure 0402 e 0603, in scatole da montaggio con superficie superiore piatta. Basato su una rivoluzionaria tecnologia ESD, protegge le componenti elettroniche sensibili dalle scariche elettrostatiche fino a 30 kV ed è adatto per applicazioni ad alta tensione fino a 32 VCC. Se correttamente integrata nella progettazione di un circuito, la Serie XGD può assorbire tranquillamente ESD ripetitive quasi quattro volte il livello massimo specificato nella norma internazionale IEC61000-4-2 senza degrado delle prestazioni. La capacità estremamente bassa rende la Serie XGD adatta per

l'uso in circuiti dati ad alta frequenza e ad alta velocità, video, antenna e I/O. Anche dopo diverse ESD, mostra una dispersione bassissima di corrente (<1 nA), assicurando perdite di potenza minime, è dunque adatta per la protezione delle componenti elettroniche alimentate a batteria come i wearable.

www.littelfuse.com



On Semiconductor espande il portafoglio di prodotti innovativi per automotive

ON Semiconductor ha introdotto una serie di nuovi circuiti integrati conformi alle norme AEC-Q100 adatti a supportare progetti di nuova generazione in ambito automotive.

La famiglia di line driver/receiver multi-point M-LVDS (low voltage differential signaling) NBA3N200/1/6S opera con una tensione di alimentazione di 3,3 V. Entrambi i dispositivi NBA3N200S NBA3N201S offrono una velocità di 200

megabit al secondo (Mbps) e presentano un range in tensione in modalità common-mode compreso tra -1V e 3,4 V. Il doppio regolatore di tensione NCV8154 con dropout di



140 mV, accetta una tensione d'ingresso da 1,9 V a 5,25 V e dispone di due pin d'ingresso indipendenti.

Il regolatore di tensione a basso drop-out (LDO) NCV8170 è stato progettato specificamente per le applicazioni portatili a batteria, quali i sistemi di accesso senza chiave, con un consumo di corrente di soli 500 nA (nanoampere).

Il regolatore LDO NCV8715 da 50 mA presenta una elevata

stabilità, una tensione massima di ingresso di 24 V e un consumo di corrente di appena 4,7 µA in qualsiasi condizione operativa.

On Semiconductor ha introdotto anche dei MOSFET singoli a canale N in grado di fornire un valore incredibilmente basso di RDS(on) che riduce al minimo le perdite di conduzione e migliora il livello generale di efficienza energetica.

www.onsemi.com

PIC24F GB4, più sicurezza e più memoria per la nuova famiglia di PIC

Microchip annuncia l'ampliamento del suo portfolio di microcontroller (MCU) eXtreme Low Power (XLP) PIC. Le funzionalità della nuova famiglia PIC24F "GB4" includono un motore di crittazione hardware integrato con opzioni sia OTP che Key RAM per l'archiviazione sicura delle chiavi, memoria Flash fino a 256 KB e il pilotaggio diretto di display LCD a segmenti, in package 64-, 100- o 121-pin. Flash dual-partition con capacità Live Update permettono ai dispositivi di gestire due applicazioni software indipendenti, e la simultanea programmazione di una partizione durante l'esecuzione di codice applicativo dall'altra. Queste funzionalità avanzate rendono la famiglia PIC24F "GB4" ideale per progettisti di applicazioni industriali, computer, medicali/fitness e portatili che richiedano trasferimento e archivia-



zione dati sicuri, e una lunga durata delle batterie.

Aspetti salienti:

- Gli MCU PIC24F "GB4" proteggono dati embedded in un range ampio di applicazioni low-power
- Assicurano l'integrità dei dati embedded senza sacrificare il consumo di potenza
- Offrono due opzioni di crypto-key per l'archiviazione: One-Time-Programmable (OTP) o Key RAM
- Motore di crittazione hardware core-independent e generatore di numeri casuali
- Supportano il pilotaggio diretto di LCD fino a 512 segmenti.

www.microchip.com

Nuovi MOSFET di potenza "quasi perfetti" da STMicroelectronics

STMicroelectronics ha esteso la sua nota serie di Power MOSFETs a canale N MDmesh M2 con l'introduzione di una nuova famiglia di dispositivi che offrono alte efficienze nel settore degli alimentatori per server, laptop, telecomunicazioni e applicazioni di consumo, soprattutto in condizioni di carichi leggeri. Con questi nuovi dispositivi, i progettisti possono realizzare dispositivi switching di potenza meno pesanti e più compatti, che consentono di raggiungere più facilmente i valori di efficienza energetica richiesti dalle attuali norme. I nuovi dispositivi 600V MDmesh M2 EP di STSemiconductors combinano un layout strip-based con una nuova struttura verticale migliorata e un processo di diffusione ottimizzato, per avvicinarsi allo switch ideale, con una resistenza in conduzione

molto bassa e una perdita di commutazione allo spegnimento altrettanto bassa. Si tratta di componenti adattati per convertitori ad altissima frequenza ($f > 150$ kHz), ideali per le applicazioni più esigenti nel campo dei PSU (Power Supply Unit). Ideali per entrambe le topologie hard- e soft-switching, incluse le topologie risonanti LLC, i nuovi dispositivi offrono perdite di commutazione estremamente basse, soprattutto in condizioni di carico limitato.

www.st.com



Verso il "cavo unico": il primo controllore USB Tipo C Power Delivery di TI

Texas Instruments ha presentato il primo controllore USB Tipo C e USB Power Delivery (PD) all-in-one, che integra un interruttore sulla porta di potenza e un multiplexer sulla porta dati. Il controllore USB PDTPS65982 è l'unico circuito integrato in commercio che copre tutto il percorso di potenza, fungendo da porta singola o doppia e consentendo un'ampia gamma di implementazioni di alimentazione in modalità host o dispositivo. L'interfaccia USB (Universal Serial Bus) è ampiamente diffusa su smartphone e altri dispositivi elettronici personali, oltre che su computer e periferiche, ad esempio le stampanti. Con i nuovi prodotti USB Tipo C di TI, i progettisti



possono offrire ai consumatori apparecchiature che non richiedono più diversi cavi per diversi tipi di porte USB. Ciascuno dei nuovi dispositivi di TI è conforme alla USB Type-C Specification 1.1 (Engineering Change Notice del 22 giugno 2015) pubblicata dall'USB Implementers Forum (USB-IF). www.ti.com

TPD4152F, modulo di potenza HV-IPD per controllo motori

Toshiba Electronics Europe ha ampliato la sua offerta di dispositivi di potenza intelligenti ad alta tensione (HV-IPD, High-Voltage Intelligent Power Device) con un nuovo circuito integrato compatto a montaggio superficiale (SMD) con tensione di lavoro nominale di 600 V dedicato al pilotaggio di motori elettrici compatti. Il dispositivo di potenza TPD4152F permette di pilotare motori in corrente continua senza spazzole (brushless DC) di bassa potenza. È un prodotto particolarmente indicato per pilotare motori utilizzati in condizionatori d'aria, purificatori d'aria, ventole e caldaie. Il dispositivo TPD4152F eroga una corrente massima nominale di 0,7 A. Al suo interno sono integrati un circuito

PWM, un circuito di decodifica a 3 fasi, il traslatore di livello per il ramo superiore del circuito di pilotaggio, il ramo inferiore del circuito di pilotaggio e le uscite IGBT. Il dispositivo è incapsulato all'interno di un contenitore SMD con 31 piedini (HSSOP31) che mantiene separati i piedini a bassa tensione da quelli ad alta tensione. Ha dimensioni di 17,5 mm x 8,4 mm x 2 mm e occupa sulla scheda il 60% di spazio in meno rispetto agli altri dispositivi di potenza intelligenti esistenti in contenitore DIP 26 offerti da Toshiba e altre aziende. Il dispositivo TPD4152F è provvisto anche di funzioni di protezione da sovratensioni e sottotensioni e integra un circuito di spegnimento per sovraccarico termico. www.toshiba.semicon-storage.com



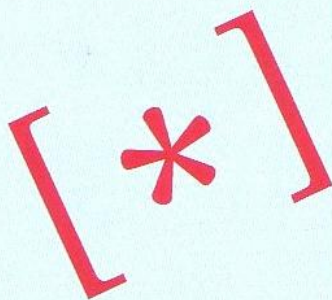
Festival della Scienza

Il Festival della Scienza di Genova è un punto di riferimento per la divulgazione della scienza nel nostro paese, un'occasione di incontro per ricercatori, appassionati, scuole e famiglie.

È anche uno dei più grandi eventi di diffusione della cultura scientifica a livello internazionale.

Un appuntamento per tutti, con incontri, laboratori, spettacoli e conferenze per raccontare la scienza in modo innovativo e coinvolgente, con eventi interattivi e trasversali. Undici giorni in cui le barriere fra scienze matematiche, naturali e umane, verranno abbattute e la ricerca si potrà toccare, vedere, capire senza confini.

Un nuovo paradigma: il Festival propone ogni anno eventi ispirati alle questioni più attuali e scottanti del dibattito scientifico, prime assolute di spettacoli e mostre dedicate all'incontro tra arte e scienza, riservando una particolare



attenzione alle novità della ricerca più avanzata e ai ricercatori dei Paesi emergenti.

Intimamente legato alla città di Genova e alla regione Liguria, il Festival è tuttavia un evento a carattere globale, con un intrinseco carattere internazionale. Gli incontri con grandi ospiti nazionali e internazionali imprezioscono ogni anno le giornate dedicate alla scienza, dando vita a collaborazioni durature con personalità e istituzioni di tutto il mondo.

A Genova dal 22 Ottobre al 1 Novembre 2015.

www.festivalscienza.it

LinuxDay 2015

Anche quest'anno torna la manifestazione più popolare e partecipata in Italia per la promozione, la diffusione, la conoscenza e l'approfondimento del software libero: il Linux Day.

La data per l'edizione 2015 è sabato 24 ottobre, nella quale come sempre decine di eventi si svolgeranno in contemporanea in tutto il Paese.

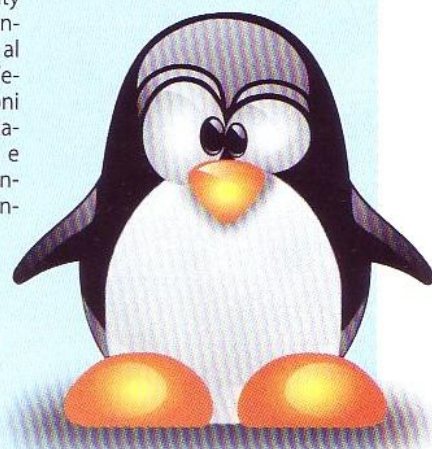
Quest'anno il tema cardine del Linux Day è la partecipazione attiva alla community freesoftware: i modi per contribuire al miglioramento, al perfezionamento ed all'estensione delle applicazioni che tutti usiamo quotidianamente, non solo scrivendo e modificando codice ma anche revisionando documentazione, pacchettizzando le nuove versioni, traducendo i testi e molto altro.

L'evento chiama dunque all'appello coloro - ed in Italia sono tanti - che ad oggi già collaborano fattivamente con i

progetti più noti e diffusi, non solo per farli conoscere ma soprattutto per illustrare da dove iniziare e come fare per portare il proprio contributo diventando parte attiva della comunità.

L'elenco aggiornato delle tantissime attività che si svolgeranno in Italia è disponibile sulle pagine del sito dell'evento.

www.linuxday.it



Rimini Mini Maker Faire, una Maker Faire da spiaggia!

Sabato 14 e domenica 15 novembre 2015 appuntamento con la prima edizione di Rimini Beach Mini Maker Faire, un evento dedicato ai "makers", un termine universale che unisce persone di diversa formazione ma che condivide l'interesse per l'apprendimento e lo sviluppo creativo.

Grazie ad una nuova dimensione sociale, creata dal Fab Lab, luoghi in cui si progetta, si sperimenta, si costruisce, i makers creano continuamente momenti di aggregazione e collaborazione, aprendo nuove frontiere al fai-da-te.

Rimini Beach Mini Maker Faire è affiliata al circuito "Maker Faire", un brand nato in America, dedicato al mondo della creatività tecnologica di proprietà di Maker Media, editore della rivista MAKE, portavoce e punto di riferimento del movimento Makers in tutto il mondo.

Un'occasione imperdibile per conoscere questa realtà così dinamica fatta di ricerca, innovazione e condivisione; i visitatori potranno testare il proprio talento di artigiano del terzo millennio grazie ai tantissimi stimoli proposti nella due giorni riminese: workshop, presentazioni, laboratori didattici, "pillole" e naturalmente materiali, strumenti, componenti, elettronica per la realizzazione di progetti DIY (do-

it-yourself). L'obiettivo dichiarato è quello di riunire in un unico spazio tutte le idee, visioni e opportunità offerte dall'applicazione delle tecnologie, digitali ma non solo.

Alla Rimini Beach Mini Maker Faire trovano spazio anche idee e prodotti per ambiti diversi dal tema principale della call.

Nell'ambito dell'evento sono previsti settori commerciali e focus specifici sui trend del momento: stampanti 3D, droni, videogame, fotografia, robotica, videomaker, Internet of Things, musica.

Completano la rassegna, Expo Elettronica, la mostra mercato di elettronica professionale e di consumo più famosa d'Italia e CAB - Recycle & Co, un'esposizione di artigiani e artisti contemporanei che recuperano e trasformano prodotti e materiali destinati alla distruzione, facendoli rivivere in piccole opere d'arte uniche e in oggetti utili, una *best practice* per limitare i rifiuti e ridurre i

consumi energetici a salvaguardia del nostro pianeta.

Alla manifestazione di Rimini saremo presenti anche noi con i prototipi dei nostri ultimi progetti, dalla Drink Maker alla Choco 3Drap.

www.makerfairerimini.it/



■ 3-5 NOVEMBRE 2015

European Utility Week

Vienna • Austria

L'European Utility Week è il più importante evento per tutte le società di servizi europee e per tutte quelle società che progettano, sviluppano e producono sistemi e soluzioni tecnologiche in questo ambito.

Un evento che offre un'occasione unica di incontro e confronto tra i professionisti del settore delle Utilities, una vetrina prestigiosa per presentare nuovi prodotti e tecnologie ma anche un luogo dove sviluppare partnership commerciali con i fornitori leader del settore.

Quest'anno la manifestazione – che cambia sede ogni anno – si svolgerà a Vienna, presso il Messe Wien Congress Center dal 3 al 5 novembre; sono previsti più di 10.000 visitatori, per la maggior parte professionisti del settore. Questi gli altri numeri della manifestazione: 420 espositori provenienti da tutto il mondo, oltre 400 relatori, 150 case study presentati durante le conferenze e i workshop,

50 startup con progetti innovativi; il tutto su una superficie complessiva di 20.000 mq, tra area espositiva e zone destinate alle conferenze. L'aspetto convegnistico è in ogni caso quello che caratterizza maggiormente l'European Utility Week, con il tentativo di interpretare (e possibilmente anticipare) quella che sarà l'evoluzione del mercato di questo importantissimo settore industriale. Molto interessante sarà, da questo punto di vista, la *Strategic Conference*, con numerose sessioni dedicate ai business model, alle strategie e all'evoluzione delle normative.

Grid & Renewables Integration: si occuperà delle problematiche riguardanti l'integrazione nella rete distributiva delle fonti rinnovabili di energia le cui caratteristiche (discontinuità di funzionamento, diffusione sul territorio, ecc.) sono molto diverse dalle attuali centrali di produzione sulle quali il sistema distributivo è "tarato".

Big Data & Analytics and Internet of Things (IoT): si occuperà di sicurezza e privacy dei dati ma, soprattutto, di

come trarre il massimo valore dai dati raccolti sul campo dai dispositivi di telelettura, la prima applicazione, in assoluto, dell'IoT.

Energy Storage: è la nuova frontiera nel campo della gestione dell'energia che, a giudizio di molti, insieme alla crescente diffusione delle fonti rinnovabili, rivoluzionerà il mercato.

Smart Homes & End User Engagement: le nuove tecnologie cambieranno sicuramente i Business Model del futuro col coinvolgimento degli utenti che assumeranno il doppio ruolo di consumatori e produttori.

Altre conferenze riguarderanno le *Smart Cities* e lo *Smart Metering & Gas Metering*. Non mancheranno i premi e i riconoscimenti alle aziende più innovative, ai giovani talenti, alle start-up, ecc.

www.european-utility-week.com

European Utility Week



■ 17-19 NOVEMBRE 2015

CARTES 2015

Parigi • Francia

Da 30 anni, Cartes Secure Connexiones è l'evento leader nel mondo per le soluzioni sicure di pagamento, identificazione e mobilità. Nel 2014 la mostra è stata visitata da oltre 20.000 persone provenienti da 165 paesi differenti ed ha visto la presenza di 450 espositori con l'80% di visitatori ed espositori provenienti dall'estero. Quest'anno l'evento metterà in campo oltre 145 conferenze e 7 tavole rotonde con 160 relatori di altissimo livello; alle conferenze si sono già iscritte 1.600 persone.

Cartes Secure Connexiones è stata fondata 30 anni fa al fine di promuovere una tecnologia che era stata appena creata: la chip card. Da allora il mondo è profondamente cambiato con molteplici nuove tecnologie che sono emerse in questo settore. Per questo motivo la manifestazione, da quest'anno, cambierà nome in TRUSTECH, con un'espressione che riflette questa evoluzione. La fiducia (TRUST) nella sicurezza dei dati sensibili è fondamentale per lo sviluppo di nuove applicazioni in un mondo sempre più connesso. TRUSTECH mira ad essere il fulcro della comunità internazionale che opera in questo settore con lo scopo di aiutare a definire le strategie future, sviluppare le modalità d'uso delle tecnologie e generare nuove oppor-



tunità di business. Come suggerisce lo slogan di quest'anno, "Unlimited Innovation", l'evento mette in mostra le più recenti e significative innovazioni provenienti da tutto il mondo, aiutando a sviluppare le modalità d'uso di tali tecnologie.

Di fronte alle rapide trasformazioni innescate dalle tecnologie digitali, TRUSTECH vuole distinguersi come evento globale da visitare obbligatoriamente, dove convergono tutte le soluzioni tecnologiche riguardanti la sicurezza delle connessioni e dove il futuro del settore prende forma.

www.cartes.com

■ 3-5 NOVEMBRE 2015

Sicurezza 2015

Milano

Dopo aver confermato, grazie all'edizione 2014, il proprio ruolo di evento leader in Italia e tra i primi in Europa, SICUREZZA si ripresenta al mercato nel 2015 con un'edizione speciale, caratterizza-

ta da una nuova collocazione negli anni dispari e un innovativo progetto espositivo che punta sempre di più all'integrazione tra sistemi e tecnologie.

Apriamo una finestra sull'evoluzione dell' "Internet of Things" (IoT) in cui le "cose" diventano smart, dialogano tra loro e tutto si collega in una unica rete, SICUREZZA 2015, insieme alla security, suo primario core business, dedicherà attenzione a prodotti e soluzioni legati a settori sinergici come



la domotica, la sicurezza stradale e la cyber security. Spazio anche all'urban lighting, ai veicoli speciali, ai sistemi GPS, e ai prodotti e servizi per la rilevazione e lo spegnimento degli incendi. Perché in un mondo in continua evoluzione, non solo sicurezza fisica e logica, ma la security diventa un driver strategico di business, fondamentale non solo per la protezione e il controllo, ma anche per il monitoraggio delle performance e la gestione da remoto.

Unico evento internazionale dedicato al settore in calendario nel secondo semestre di quest'anno, la manifestazione, grazie alle numerose riconferme e alla fiducia mostrata da nuove realtà, si mostra in sintonia con un mercato sempre più vitale e ricco di opportunità.

Lo dicono i numeri: i preconsuntivi 2014 di ANIE Sicurezza confermano il trend positivo degli ultimi tre anni, con una crescita del fatturato aggregato rispetto all'anno precedente del 5%. L'industria italiana fornitrice di tecnologie per la Sicurezza e Automazione edifici rappresentata da ANIE Sicurezza a fine 2014 vale un fatturato totale aggregato pari a circa 2 miliardi di euro, confermandosi stabile malgrado il fenomeno generale del downpricing. Tra i comparti del settore cresce maggiormente la Videosorveglianza, con un incremento su base annua vicino al 10%, ma dà ottimi segnali anche il segmento Antintrusione con un incremento vicino al 7%, nel confronto annuo. Più contenuta rispetto alla media del comparto la crescita del segmento Antincendio, vicina al 2% annuo.

Per quanto riguarda la manifestazione, nel 2014 Sicurezza ha visto un incremento del 33% della superficie espositiva, del 23% del numero di espositori (a quota 512) e del 26% dei visitatori che sono stati 20.301. Nu-

■ 24-26 NOVEMBRE 2015

SPS IPC Drives

Norimberga • Germania

Dal 24 al 26 novembre 2015 Norimberga ospiterà la 26^a edizione di SPS IPC Drives, la manifestazione che copre l'intera gamma di prodotti e servizi di automazione elettrica, offrendo una visione completa di tutta l'offerta del settore. Le premesse per un'altra fiera di grande successo ci sono tutte: oltre 1.600 espositori, sia locali che stranieri - tra i quali tutti i principali Key Player - presenteranno prodotti, soluzioni e tecnologie d'avanguardia per affrontare le sfide del futuro, confermando così per la Mostra di Norimberga il ruolo di piattaforma di innovazione per l'automazione industriale leader in Europa. Industry 4.0 è uno dei temi centrali della fiera di quest'anno presso molti espositori. Per la prima volta SPS IPC Drives propone nel padiglione 3A "Area Industry 4.0", con la possibilità per il visitatore di ricevere informazioni esaustive su Industry 4.0 - la produzione digitalizzata collegata in rete e omogenea:

- Lo stand congiunto e Forum "Automation"

meets IT" presenta modelli di business basati su dati nonché soluzioni di automazione basate sull'IT, sulla via della produzione digitale del futuro.

- Lo stand congiunto "MES goes Automation" mostra come è possibile ottimizzare l'elaborazione degli ordini ed i processi produttivi con l'impiego del MES.
- La speciale area espositiva di SmartFactoryKL illustra la fabbrica intelligente indipendente dal produttore.

Queste le categorie merceologiche più importanti che caratterizzano SPS IPC Drives:

- Sistemi di controllo
- IPC
- Sistemi e componenti di azionamento
- Dispositivi di interfaccia uomo-macchina (HMI)
- Comunicazione industriale
- Software industriale
- Tecnologia di interfacciamento
- Infrastrutture meccaniche
- Sensoristica

Le associazioni ZVEI, nel padiglione 2, e VDMA, nel padiglione 3, hanno collaborato nell'organizzazione delle varie conferenze in programma nonché alle tavole rotonde sui trend del momento. Gli stand congiunti "AMA Centro per i sensori, tecnologia di misura e di collaudo" e "Elaborazione industriale dell'immagine" del VDMA offrono ai visitatori approfondite informazioni sullo stato dell'arte di queste tecnologie.

www.mesago.de



mero che l'edizione 2015 dovrebbe ulteriormente incrementare.

"Con SICUREZZA 2015 proponiamo un format sempre più vicino alle richieste delle aziende, e le adesioni ad oggi preannunciano già che il nuovo corso degli anni dispari rappresenta una evoluzione importante, soprattutto in termini di internazionalizzazione" – afferma Giuseppe Garri, Exhibition Manager di SICUREZZA. "Punteremo sulla rappresentatività dell'offerta,

sulla valorizzazione dell'elevata ricerca tecnologica alla base delle soluzioni oggi disponibili, ma anche sull'attenzione all'evoluzione del mercato, con affondi su tematiche apparentemente più trasversali come l'Internet of Things e la Cyber Security. E presenteremo questo mondo ai buyer esteri provenienti dai mercati più interessanti per l'export – Bacino del Mediterraneo, Medio Oriente e Nord Africa - grazie a un investimento mirato

e ottimizzato mediante la nostra piattaforma di business matching EMP (Expo Matching Programme). L'obiettivo a cui puntiamo è quello di essere vicino a tutte le imprese come partner, grazie anche alla sintonia con le associazioni dei produttori, ANIE Sicurezza e ASSOSICUREZZA, insieme alle quali stiamo costruendo il progetto puntando a diventare bacino di riferimento per tutti i mercati del sud Europa".
www.sicurezza.it

MOSTRE MERCATO

ISOLA DELLA SCALA (VR)
ElettroBIT & ExpoRadio
Palafiere - Via Bastia
Organizzazione: Openoffice Srl
Tel: 0308376078 - Fax: 0308376078
info@fierelettronica.it - www.fierelettronica.it
31 Ottobre - 01 Novembre 2015

ERBA (CO)
FIERA DELL'ELETTRONICA
Lario Fiere - Erba (CO)
Organizzazione: Lariofiere
Tel: 031637414
erbaelettronica@lariofiere.com
www.lariofiere.com
07 e 08 Novembre 2015

CHIETI
MOSTRA MERCATO DELL'ELETTRONICA
Centro espositivo e di servizi Piazza G. B. Vico, 3 - Tel: 3286467529
Organizzazione: G.E.F.E. Fiere
fiere@gefe.it - www.gefe.it
14 e 15 Novembre 2015

LIVORNO
FIERA DELL'ELETTRONICA DI LIVORNO
Pala Livorno - Organizzazione: Prometeo
Tel: 057122266 - Fax: 05710831142
info@prometeo.tv - www.prometeo.tv
14 e 15 Novembre 2015

BOLOGNA
MONDO ELETTRONICA
Bologna Fiere - Tel: 054527548 - Fax: 0545291179
Organizzazione: Expo Fiere Srl
info@expositionservice.it
www.mondoelettronica.net
21 e 22 Novembre 2015

PESCARA
FIERA MERCATO NAZIONALE DEL RADIOAMATORE DI PESCARA
Centro IBISCO - Città S. Angelo (PE)
Organizzazione: Ari Pescara - Tel: 0854714835
arifiera@aripescara.org - www.aripescara.org
28 e 29 Novembre 2015

BASTIA UMBRA (PG)
EXPO ELETTRONICA
Quartiere Fieristico - Bastia Umbra (PG)
Organizzazione: Blu Nautilus srl
Tel: 054153295 - Fax: 0541439584
info@blunautilus.it - www.expoelettronica.it
31 Ottobre - 01 Novembre 2015

FERRARA
LA GRANDE FIERA CAMPIONARIA
Quartiere Fieristico - Ferrara
Organizzazione: Expo Fiere Srl
Tel: 054527548 - Fax: 0545291179
info@mondoelettronica.net
www.mondoelettronica.net
07 e 08 Novembre 2015

RIMINI
EXPO ELETTRONICA
Rimini Fiera - Via Emilia 155
Organizzazione: Blu Nautilus srl
Tel: 0541439573 - Fax: 0541439584
info@expoelettronica.it - www.expoelettronica.it
14 e 15 Novembre 2015

CHIUDUNO (BG)
FIERA DELL'ELETTRONICA
Quartiere fieristico Chiusduno
Organizzazione: Open Office srl
Tel: 0308376078
info@fierelettronica.it - www.fierelettronica.it
14 e 15 Novembre 2015

VERONA
ELETTRO EXPO
Quartiere Fieristico di Verona
Organizzazione: Veronafiere
Tel: 0458298111 - Fax: 0458298288
info@veronafiere.it - www.elettroexpo.it
28 e 29 Novembre 2015

LODI
FIERAMERCATO DELL'ELETTRONICA
Fiera di Lodi - Via dell'Artigianato 4
Organizzazione: Printmix - Tel: 3356882801
info@fieramercatodellelettronica.it
www.fieramercatodellelettronica.it
28 e 29 Novembre 2015

PADOVA
FIERA DELL'ELETTRONICA
Quartiere Fieristico - Padova
Organizzazione: PadovaFiere - Tel: 049840465
massimo.bressan@padovalfiere.it
www.padovalfiere.it
31 Ottobre - 01 Novembre 2015

FIRENZE
ELETTRONICA FIRENZE
OBIHALL - Via Fabrizio De Andrè - Firenze
Organizzazione: Prometeo
Tel: 057122266 - Fax: 05710831142
info@prometeo.tv
www.prometeo.tv
07 e 08 Novembre 2015

RIMINI
RIMINI BEACH - MINI MAKER FAIRE
Rimini Fiera - Via Emilia 155
Organizzazione: Blu Nautilus srl
Tel: 0541439573 - Fax: 0541439584
info@expoelettronica.it - www.expoelettronica.it
14 e 15 Novembre 2015

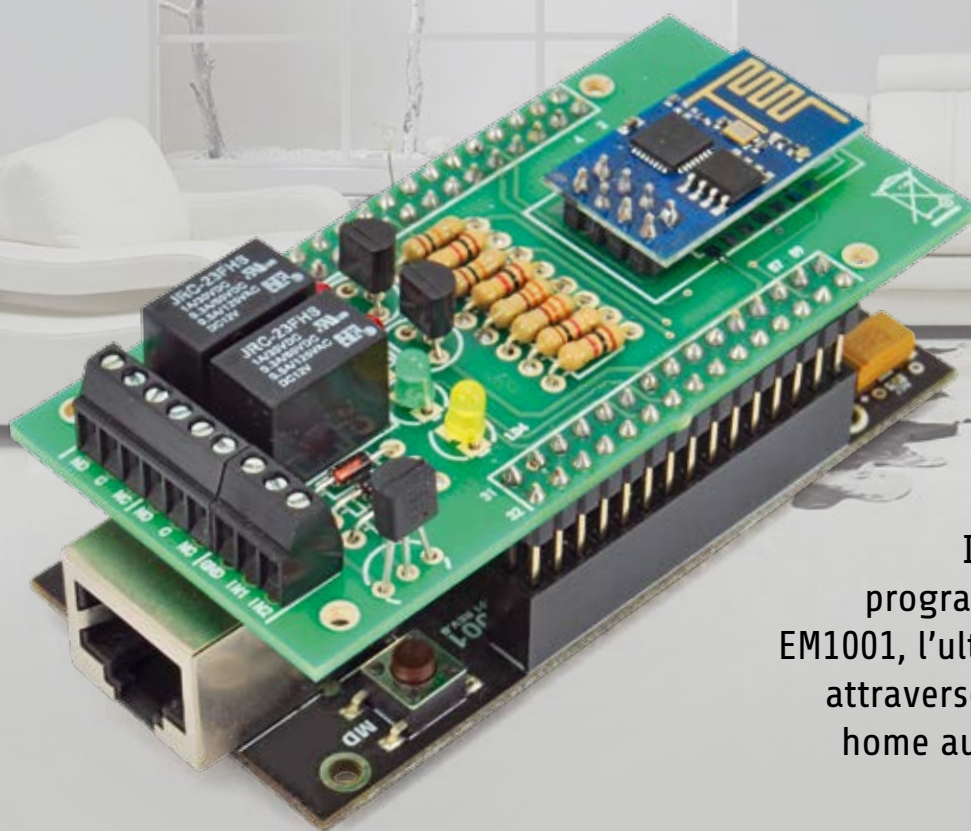
PORDENONE
RADIOAMATORE 2
Quartiere Espositivo, Tel: 0434232111
Organizzazione: Pordenone Fiere Spa
info@fierapordenone.it
www.radioamatorepordenone.it
21 e 22 Novembre 2015

PARMA
FIERA DELL'ELETTRONICA
Centro Fiere, Parma
Organizzazione: G.E.F.E. Fiere
Tel: 3286467529
fiere@gefe.it - www.gefe.it
28 e 29 Novembre 2015

L'elenco aggiornato di tutte le Mostre Mercato del 2015 è disponibile sul sito www.elettronica.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.elettronica.in/myMain/Fiere/p_fiere.asp

SMARTEM, IL TERMOSTATO INTELLIGENTE



Impariamo a utilizzare e programmare il nuovo modulo EM1001, l'ultimo nato in casa Tibbo, attraverso un progetto pratico di home automation: il termostato smart. Prima puntata.

..... di SAVINO GIUSTO

S secondo le stime di ABI Research, una delle più importanti società di ricerca di mercato, entro la fine del 2020 oltre 30 miliardi di dispositivi saranno collegati ad Internet in modalità wireless. Attualmente, il numero complessivo di dispositivi connessi si aggira intorno ai 10 miliardi. Si tratta di un trend in continua crescita, associato ad una tecnologia relativamente recente conosciuta con il nome di IoT (Internet of Thing o Internet delle Cose). Durante il CES (Consumer Electronics Show) 2015, il presidente di Samsung Electronic ha detto che l'IoT ha il

potenziale per trasformare la nostra economia, la società e il modo in cui viviamo le nostre vite. Questa affermazione lascia intendere che gli investimenti in questo settore nei prossimi anni saranno elevati; tra le prime 10 società che al mondo hanno maggiormente investito in questo settore spiccano Apple, Google, Intel e Microsoft.

I principali settori applicativi dell'Internet delle Cose sono Home Automation, Automotive, Robotica, Biomedicale, Monitoraggio industriale, Telesorveglianza, Smart Grid, Telematica.



Fig. 1 - Il termostato "smart" Nest di Google.

Attualmente, il principale settore trainante è quello relativo all'Home Automation: si punta, cioè, a rendere "smart" elettrodomestici, controlli di temperatura e apparecchiature elettroniche in generale, connettendoli ad Internet in modo da poterli controllare e programmare in modalità remota. Un significativo esempio di dispositivo di Home Automation è sicuramente rappresentato dal termostato sviluppato dalla Nest Labs, società acquisita successivamente da Google agli inizi del 2014; si tratta di un termostato elettronico programmabile, capace di auto-apprendere le abitudini di climatizzazione degli occupanti un'abitazione, con l'obiettivo di ottimizzare il consumo energetico. Il dispositivo può essere programmato e consultato da remoto tramite un qualunque dispositivo mobile, grazie alla connessione WiFi che gli permette di connettersi ad Internet. In queste pagine e nel seguito, in programma il mese prossimo, vi mostreremo come realizzare un termostato smart (che chiameremo per brevità

Smartem), tramite il quale sia possibile l'attivazione da remoto del proprio sistema di riscaldamento o di climatizzazione. Per una simile applicazione è necessario selezionare in maniera opportuna il tipo di piattaforma (microcontrollore) da impiegare. Le caratteristiche principali che il dispositivo dovrà avere sono:

- ridotte dimensioni, in modo da poter essere integrato in un box simile a quello dei normali termostati;
- basso consumo, in modo da poter essere alimentato anche a batteria;
- disponibilità di un buon numero di I/O per eventuali espansioni future;
- predisposizione alle tecnologie Ethernet per la connessione ad Internet;
- semplicità di programmazione.

Per dare vita a questo progetto abbiamo selezionato per voi il nuovo modulo EM1001 sviluppato dalla Tibbo Technology; in questo articolo analizzeremo l'ambiente di sviluppo del modulo (TIDE) e faremo alcuni esempi di programmazione. Basandoci su quanto avre-

mo appreso, nella prossima puntata costruiremo insieme il termostato smart.

LA PIATTAFORMA EM1001

Il modulo EM1001 rappresenta l'evoluzione del precedente EM1000 della Tibbo, sul quale potete trovare un valido corso di programmazione di 8 puntate presentate su Elettronica In, dal fascicolo n° 122 al n° 129. Per agevolare il confronto, in Fig. 2 e Fig. 3 pubblichiamo le foto delle piattaforme, rispettivamente, EM1001 ed EM1000. La principale differenza che si può notare è sicuramente il fatto che la nuova piattaforma EM1001, rispetto al precedente modello, si presenta come un modulo "auto-sufficiente", nel senso che dispone già di un alimentatore switching che da 12Vcc ricava 3,3 V, oltre che di un connettore RJ45 per la connessione ethernet 10/100BaseT, di tre LED di stato e di un pulsante. Il modulo EM1000 richiede, per poter essere utilizzato, una scheda madre su cui inserirlo e sulla quale sono presenti i suddetti componenti, mentre l'EM1001 è esso stesso la host board.

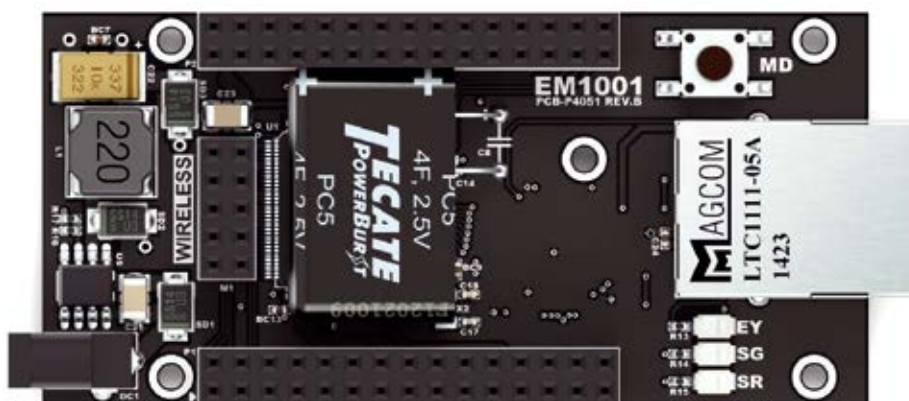


Fig. 2 - Layout del nuovo modulo EM1001.

Questo sicuramente rappresenta un sensibile miglioramento che gli utilizzatori di questo modulo non potranno che apprezzare, considerato che il costo dei due moduli è rimasto praticamente invariato, mentre le dimensioni risultano leggermente superiori (75 x 36 x 17,5mm).

Passiamo ora a descrivere le caratteristiche "di targa" del dispositivo, che è dotato di 3 connettori femmina a passo 2,54mm disposti a "U":

- porta Ethernet 10/100BaseT con connettore RJ45;
- predisposizione per l'interfacciamento con il modulo WiFi GA1000, molto utile per tutte le applicazioni che richiedono un trasferimento dati di tipo wireless;
- ampia dotazione di porte seriali UART (fino a 4) con la possibilità di impostare una velocità di trasferimento fino a 921.600 bps; tale velocità rappresenta un plus dell'EM1001, perché la maggior parte dei microcontrollori consente di raggiungere un massimo di 115.200 bps;
- una generosa dotazione di memoria Flash (1.024 kB) che può essere ripartita tra il firmware, l'applicazione e i dati;
- 2 kB di memoria EEPROM per la memorizzazione di eventuali dati di configurazione della vostra applicazione;
- fino a 54 linee configurabili come ingresso o uscita, di cui 8 impostabili ad interrupt;
- RTC (Real Time Clock); si tratta di un circuito integrato con funzione di orologio in grado di scandire il passare delle ore e dei giorni.

Ovviamente l'RTC deve essere alimentato anche quando la scheda non lo è; per ovviare a questo problema è necessario fornire alimentazione esterna con una batteria di backup oppure la casa produttrice mette a disposizione il modello EM1001 dotato di un supercondensatore che funge da batteria quando il dispositivo è privato dell'alimentazione. Un dettaglio maggiore delle funzionalità delle singole linee si può avere osservando la **Fig. 4**, che mostra la pin-out del modulo.

L'AMBIENTE DI SVILUPPO TIDE

Altro punto di forza dei prodotti di casa Tibbo, ed in particolare del modulo EM1001, è la disponibilità da parte del programmatore di una suite completa per la programmazione, un linguaggio di facile comprensione (di tipo Basic-like), un ambiente di sviluppo sofisticato con debugging incluso e una nutrita gamma di oggetti e librerie; il tutto messo a disposizione gratuitamente. All'indirizzo web

<http://tibbo.com/basic/product/tide.html> potete trovare l'eseguibile per l'installazione dell'intero pacchetto applicativo.

Il fulcro della programmazione del modulo EM1001 (così come dell'EM1000) è il TIDE, acronimo che sta per Tibbo Integrated Development Environment, ossia l'ambiente di sviluppo che utilizzeremo prossimamente per trasformare le nostre idee in codice operativo per il dispositivo. Già al primo utilizzo, questo IDE appare leggero, veloce e gradevole alla vista; questo però non deve trarre in inganno, poiché il TIDE è al tempo stesso un potente strumento dotato di debugging integrato che non richiede programmi esterni.

La **Fig. 5** mostra le principali sezioni in cui è suddivisa la finestra del TIDE, le relative funzionalità e le particolarità. La tag-line del produttore lascia ben sperare chi si avvicina per la prima volta a questo strumento: "Don't study TIDE, explore it" ("Non studiare TIDE, esploralo"), dice, invitando il neofita ad usare lo strumento piuttosto



Fig. 3 - Layout del precedente modulo EM1000.

che leggere centinaia di pagine di manuale.

Noi che abbiamo utilizzato il TIDE per programmare il modulo EM1001 possiamo in effetti confermare la semplicità dell'approccio.

Come vedremo nel seguito, dopo aver scritto il nostro codice sarà sufficiente la pressione del tasto funzione F5 per eseguire in serie le operazioni di compilazione, upload ed esecuzione sul dispositivo. Procediamo ora con il primo esempio pratico, il classico Hello World, con il quale vi mostreremo come sia semplice creare un progetto, compilarlo, programmare il modulo ed effettuare successivamente il debug. Nella fattispecie, il nostro codice di esempio farà lampeggiare il LED rosso pre-

sente sulla scheda al momento della pressione del pulsante, mentre per il resto del tempo sarà acceso il LED verde; inoltre, all'avvio del dispositivo sia il LED verde sia il rosso lampeggeranno per tre volte sincronizzati.

CREAZIONE DI UN NUOVO PROGETTO

Per creare un nuovo progetto dobbiamo impartire il comando *New Project* del menu *File*, allorché ci apparirà la finestra riportata in Fig. 6. In essa i campi da compilare sono:

- *Platform*; selezionate EM1001;
- *Project Name*; assegnate un nome al progetto (per esempio Hello World);
- *Location*; se volete cambiare il percorso predefinito dal TIDE

per il progetto, assegnatene uno personalizzato facendo clic sul pulsante *Browse*;

- *Target Address*; per individuare il dispositivo da programmare dovete collegarlo alla stessa LAN cui è collegato il PC che usate per la programmazione.

A questo punto potete fare clic sul pulsante *Select* e sarà avviata l'applicazione Device Explorer, la cui finestra di dialogo è mostrata nella Fig. 7. Essa permette di selezionare il dispositivo di interesse e risulta molto comoda quando si deve effettuare l'aggiornamento dell'applicazione o del firmware di diversi dispositivi connessi in rete. Facendo clic sul pulsante *Select* si ritorna alla finestra *New Project*, dalla quale potete fare clic sul pulsante *OK* per avviare la creazione del progetto.

SCRITTURA DEL CODICE

Prima di iniziare a descrivere il codice del progetto Hello World è doverosa una precisazione sull'architettura del modulo EM1001: la Tibbo ha creato per questa scheda un sistema operativo, denominato TiOS (Tibbo Operative System), su cui "gira" la nostra applicazione.

TiOS è un sistema a due processi: il primo è trasparente al programmatore e si occupa di operazioni quali gestione della memoria e stack TCP, mentre il secondo processo è in pratica una macchina virtuale che esegue l'applicazione utente. Riportate il **Listato 1** all'interno dell'editor di testi del TIDE; tale codice, per quanto semplice, ci permette di introdurre alcuni concetti fondamentali, tra cui

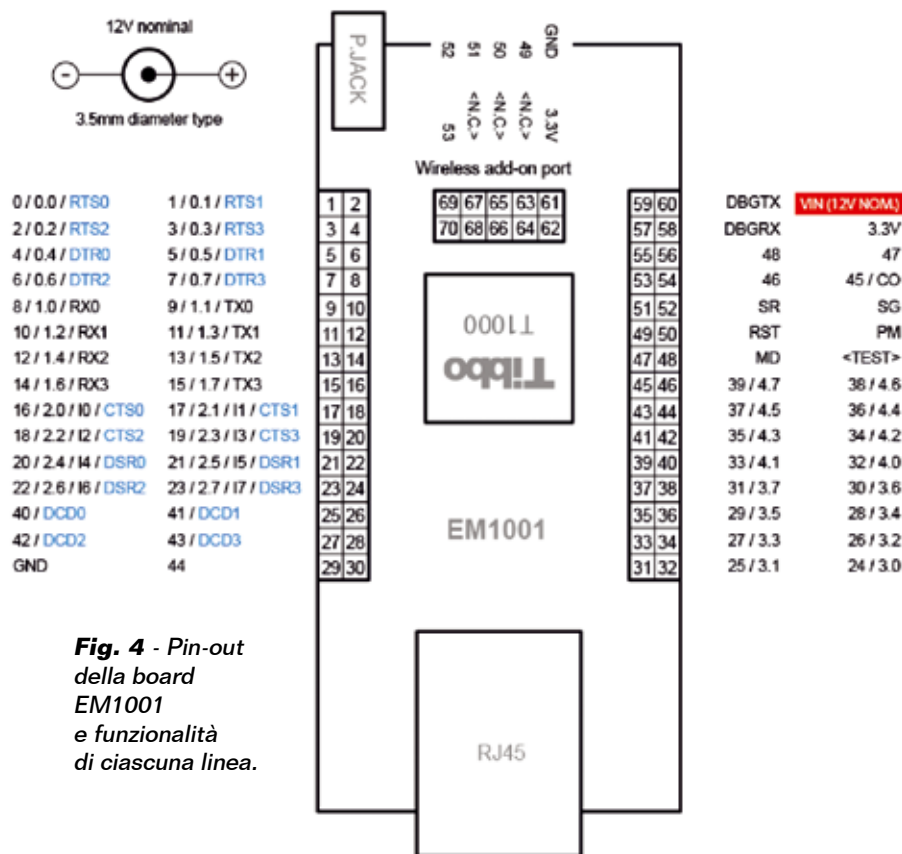


Fig. 4 - Pin-out della board EM1001 e funzionalità di ciascuna linea.

la gestione degli eventi e l'uso degli oggetti. Ma prima di spiegarli in dettaglio proviamo ad eseguire il codice sul dispositivo premendo il tasto funzione F5; il risultato della compilazione ed eventuali messaggi di errore sono riassunti nella finestra Output del TIDE; se non sono presenti errori, il codice sarà in esecuzione sul modulo EM1001. Il funzionamento dell'applicativo può essere riassunto come segue.

- All'avvio viene richiamato l'evento *on_sys_init*, il cui codice è eseguito una

sola volta; nella fattispecie, viene usato l'oggetto *pat* ed il metodo *play* per eseguire un pattern di lampeggio dei due LED di sistema: vengono fatti lampeggiare entrambi i LED rosso e verde per tre volte. Si tratta di un sistema molto veloce ed efficiente per segnalare gli stati dell'applicativo tramite diodi luminosi. Il primo parametro del metodo *play* serve per costruire il pattern ed è possibile scegliere tra i seguenti:

- '-' spegne entrambi i LED;

- 'R' accende il LED rosso e 'G' quello verde;
- 'B' accende entrambi i LED;
- '~' crea un loop con il pattern successivo;
- '*' raddoppia la frequenza di lampeggio, mentre '**' la quadruplica;

Il secondo parametro specifica se il pattern può interrompere un eventuale altro pattern in esecuzione (PL_PAT_CANINT). Molto utile è l'help, richiamabile semplicemente con il tasto funzione F1 dopo essersi

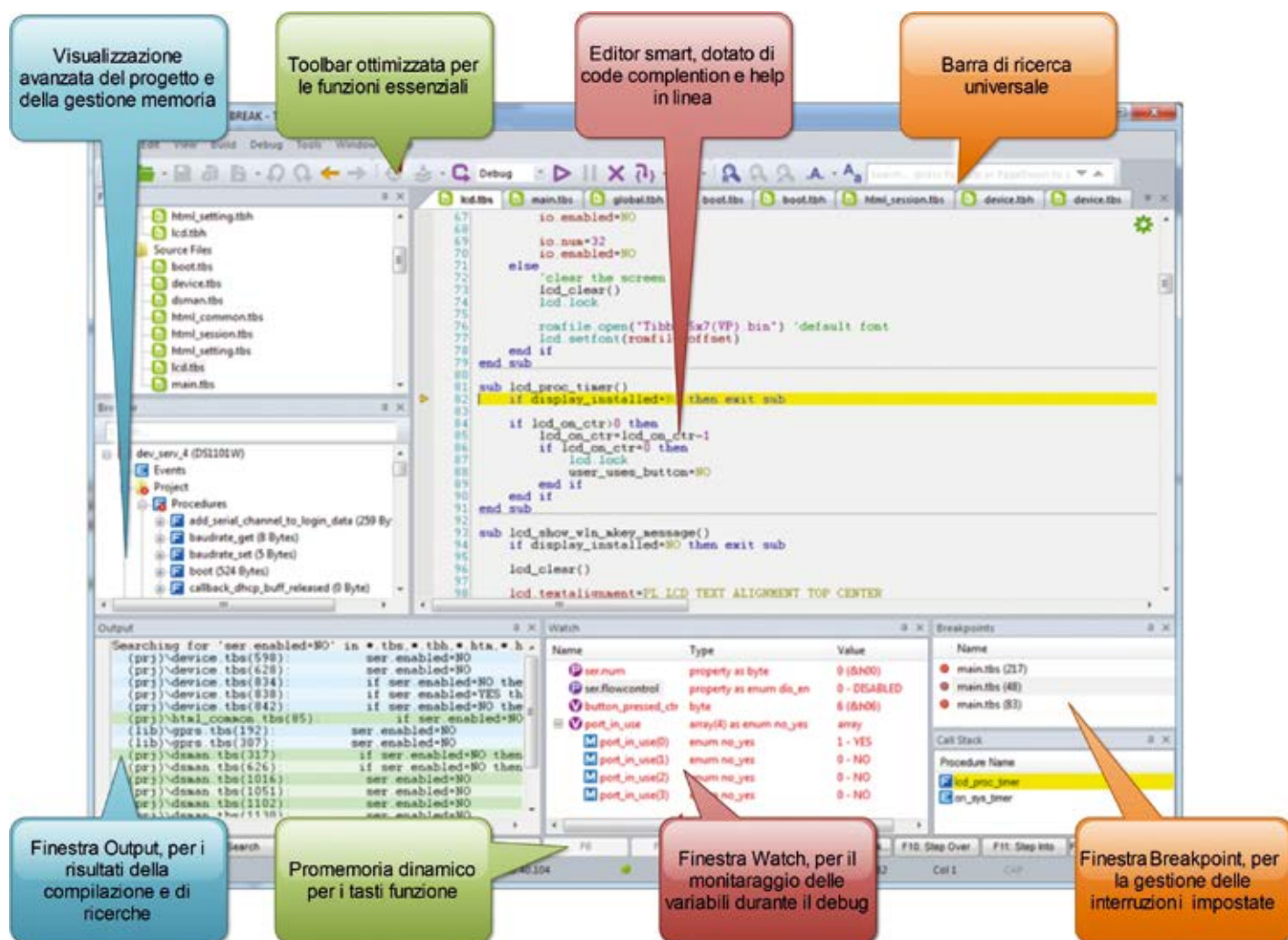


Fig. 5 - L'ambiente di sviluppo TIDE, con evidenziati i suoi principali punti di forza.

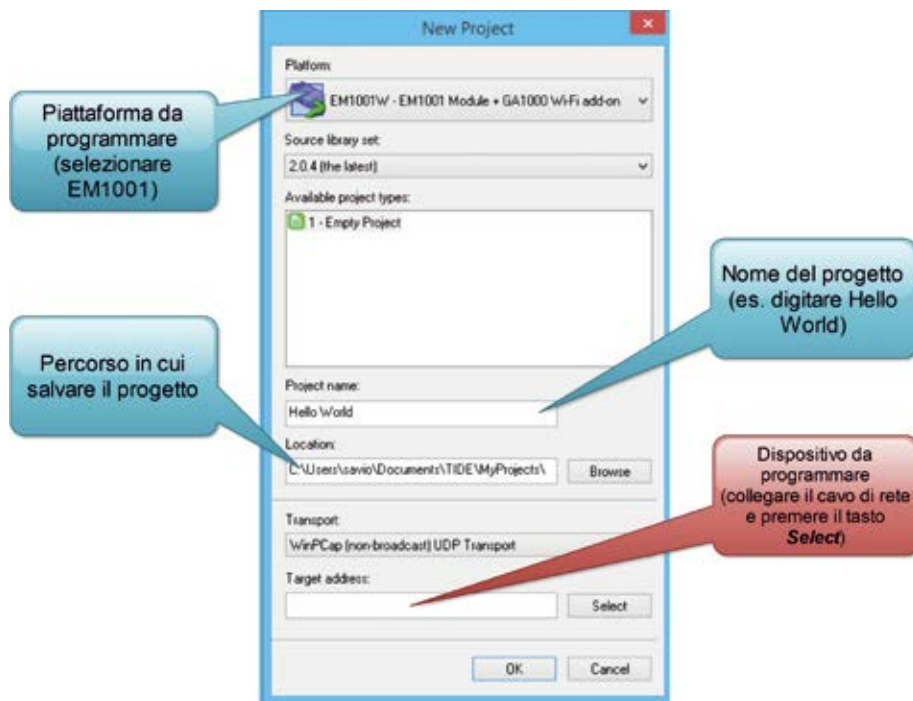


Fig. 6 - Selezionare File>New Project per accedere alla finestra di configurazione del nuovo progetto.

posizionati sull'oggetto di interesse.

- L'altro evento che è stato inserito nell'applicazione è *on_pat*. Si tratta di un evento dell'oggetto *pat* che viene fatto scattare nel momento in cui termina una sequenza di LED. Da quanto descritto per l'evento *pat*, si comprende che in assenza di altro evento esterno viene acceso il LED verde in maniera fissa.
- Infine abbiamo inserito l'evento *on_button_pressed*, che come dice lo stesso nome sarà eseguito tutte le volte che sarà premuto il pulsante presente sulla board. La presenza dell'asterisco nel pattern ci fa intendere che il LED rosso lampeggerà per tre volte ad una frequenza doppia rispetto al normale.

La programmazione con il TIDE è semplificata grazie alla presenza di oggetti e

metodi perfettamente documentati nell'help (richiamabile tramite il tasto F1). Generalmente qualsiasi oggetto deve essere inizializzato nella routine *on_sys_init* che è pre-

sente in qualunque applicativo. Ogni oggetto dispone di solito di uno o più metodi che ci permettono di gestirlo; la finestra Browser-Project di TIDE ci mostra l'elenco di tutti gli eventi disponibili.

DEBUGGING DELL'APPLICAZIONE

Quando il codice dell'applicazione inizia a diventare più complicato di quello basilare finora proposto, sorge la necessità di effettuarne il debug e monitorare come cambia il valore assunto dalle variabili. Come già anticipato nei paragrafi precedenti, questo può essere fatto con il TIDE in maniera semplice e senza alcun hardware aggiuntivo; vediamo come: supponiamo di voler verificare se l'applicazione presentata nel paragrafo precedente si comporta correttamente alla pressione del tasto, cioè entra nella routine *on_button_pressed*. In tal



Fig. 7 - Dopo aver collegato il dispositivo da programmare nella stessa rete LAN del PC su cui è installato TIDE, è possibile selezionare il modulo interessato.

caso dobbiamo posizionarci alla riga 10 e premere il tasto funzione F9, come suggeritoci dalla finestra di lavoro del TIDE nella barra delle shortcut; in questo modo comparirà un pallino rosso nella parte sinistra della riga selezionata, come evidenziato nella Fig. 8. Se riavviamo l'applicazione tramite F5 e poi premiamo il tasto vedremo che l'esecuzione si arresta proprio in corrispondenza della riga 10.

ESEMPI

DI PROGRAMMAZIONE: APERTURA DI UN SOCKET TCP

L'esempio di applicazione presentato nel paragrafo precedente, per quanto semplice, ci è stato di ausilio per muovere i primi passi con l'ambiente di sviluppo TIDE. Seguono ora alcuni esempi di programmazione volti alla gestione dell'interfaccia Ethernet. La semplicità del mondo Tibbo non si esaurisce all'IDE, ma semplifica notevolmente anche la scrittura di codice orientato alla comunicazione TCP/IP, astruendo il pro-



Anche il tavolo è Smart!

IKEA disegna la cucina del futuro, che sarà smart non solo negli elettrodomestici ma anche nel tavolo. Smart Table, così si chiama l'ultima novità, è uno speciale tavolo a realtà aumentata che spiegherà, passo passo, come preparare una ricetta indicando tutti gli ingredienti necessari e in che ordine vanno utilizzati, oppure ci dirà come affettare una verdura alla perfezione. Smart Table IKEA nasconde una bilancia, un piano di cottura a induzione e, soprattutto, riconosce gli ingredienti appoggiati sulla sua superficie, grazie a un sistema di riconoscimento d'immagine basato su una telecamera e un proiettore fissato al soffitto. Nella proposta IKEA non manca la gestione automatica dei rifiuti e del riciclo per evitare lo spreco delle risorse e rispettare l'ambiente. Il tavolo smart può anche suggerire ricette preparabili con gli ingredienti che vi si appoggiano, registra le preparazioni alimentari che avvengono su di esso per poi condividerle mediante i social, calcolare e tenere il tempo di cottura, mantenere i cibi in caldo e...caricare lo smartphone. Cosa chiedere di più?

Listato 1 - Pilotaggio dei LED e gestione degli eventi.

```

1.  sub on_sys_init
2.  ` Questo evento è generato una sola volta all'avvio del dispositivo
3.    pat.play("B-B-B---", PL_PAT_CANINT)
4.  end sub
5.
6.
7.  `=====
8.  sub on_button_pressed
9.  ` Questo evento è eseguito ogni volta che si preme il tasto
10.    pat.play("~R-R-R-", PL_PAT_CANINT)
11.  end sub
12.
13.
14.  `=====
15.  sub on_pat
16.  ` Questo evento è generato ogni volta che termina una sequenza di led
17.  ` avviata tramite la funzione pat.play
18.    pat.play("~G", PL_PAT_CANINT)
19.  end sub

```

grammatore da tutti i compiti di gestione dello stack. Vedremo in questo esempio come "aprire" un socket di tipo TCP in ascolto sulla porta 2000 che permetta lo scambio di dati per il comando da remoto delle linee della board (per maggiori informazioni su che cosa si intenda per socket vi invitiamo a fare riferimento al riquadro "I network Socket"). Il codice completo è riportato nel Listato 2 e la sua struttura è molto simile a quanto analizzato nel Listato 1, a parte il fatto che è stato introdotto

I network socket

Un socket internet è caratterizzato da una combinazione univoca formata da:

- indirizzo del socket locale costituito da indirizzo IP e porta locale;
- indirizzo del socket remoto costituito da indirizzo IP e porta remota;
- protocollo, che generalmente può essere uno tra TCP e UDP.

Tramite un socket è possibile implementare uno scambio di informazioni tra due sistemi. Se il protocollo impiegato è di tipo TCP (quindi orientato alla connessione) si parla di Stream Socket, mentre se si adotta l'UDP si utilizza il termine Datagram Socket.

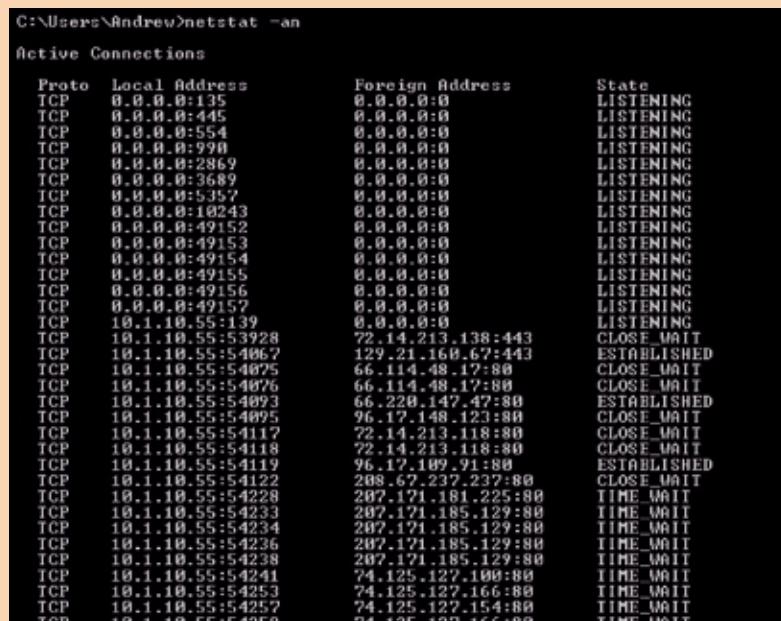
Un'applicazione (in esecuzione su un PC o modulo embedded) che fornisce un particolare tipo di servizio è generalmente definita server e crea un socket che è in ascolto su una o più porte (definito stato listening) e

può gestire simultaneamente un certo numero di connessioni in ingresso. Per ciascuna connessione viene creato un socket dedicato (definito stato established) nel momento in cui un client fa richiesta di connessione.

Il server può creare differenti socket TCP concorrenti, con la stessa porta e il medesimo indirizzo IP locale, per servire differenti client; ciascuno di essi è trattato come un socket differente poiché, pur restando uguali porta e IP locali del server, cambiano porta e IP remoti dei client.

Un comodo strumento per visualizzare tutti i socket aperti sul proprio PC è *netstat*.

La figura sotto mostra l'output del comando "*netstat -an*": ciascuna riga rappresenta, come si è spiegato, un socket individuato da protocollo (PROTO), indirizzo locale (Local Address) e indirizzo remoto (Foreign Address) con la relativa indicazione dello stato (es. ESTABLISHED).



Proto	Local Address	Foreign Address	State
TCP	0.0.0.0:135	0.0.0.0:0	LISTENING
TCP	0.0.0.0:445	0.0.0.0:0	LISTENING
TCP	0.0.0.0:554	0.0.0.0:0	LISTENING
TCP	0.0.0.0:990	0.0.0.0:0	LISTENING
TCP	0.0.0.0:2869	0.0.0.0:0	LISTENING
TCP	0.0.0.0:3689	0.0.0.0:0	LISTENING
TCP	0.0.0.0:5357	0.0.0.0:0	LISTENING
TCP	0.0.0.0:10243	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49152	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49153	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49154	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49155	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49156	0.0.0.0:0	LISTENING
TCP	0.0.0.0:49157	0.0.0.0:0	LISTENING
TCP	10.1.10.55:139	0.0.0.0:0	LISTENING
TCP	10.1.10.55:53928	72.14.213.138:443	CLOSE_WAIT
TCP	10.1.10.55:54067	129.21.160.67:443	ESTABLISHED
TCP	10.1.10.55:54075	66.114.48.17:80	CLOSE_WAIT
TCP	10.1.10.55:54076	66.114.48.17:80	CLOSE_WAIT
TCP	10.1.10.55:54093	66.220.147.47:80	ESTABLISHED
TCP	10.1.10.55:54095	96.17.148.123:80	CLOSE_WAIT
TCP	10.1.10.55:54117	72.14.213.118:80	CLOSE_WAIT
TCP	10.1.10.55:54118	72.14.213.118:80	CLOSE_WAIT
TCP	10.1.10.55:54119	96.17.109.91:80	ESTABLISHED
TCP	10.1.10.55:54122	208.67.237.237:80	CLOSE_WAIT
TCP	10.1.10.55:54228	207.171.181.225:80	TIME_WAIT
TCP	10.1.10.55:54233	207.171.185.129:80	TIME_WAIT
TCP	10.1.10.55:54234	207.171.185.129:80	TIME_WAIT
TCP	10.1.10.55:54236	207.171.185.129:80	TIME_WAIT
TCP	10.1.10.55:54238	207.171.185.129:80	TIME_WAIT
TCP	10.1.10.55:54241	74.125.127.100:80	TIME_WAIT
TCP	10.1.10.55:54253	74.125.127.166:80	TIME_WAIT
TCP	10.1.10.55:54257	74.125.127.154:80	TIME_WAIT
TCP	10.1.10.55:54259	74.125.127.166:80	TIME_WAIT

Output del comando "*netstat -an*" per la visualizzazione di tutti i socket TCP attivi su un PC.

un nuovo oggetto sock che rappresenta il cuore di tutte le comunicazioni TCP/IP. Padroneggiare questo oggetto ci permetterà di telecomandare

l'intero sistema e telecomandare qualsiasi dispositivo connesso alla board EM1001 sia da PC sia da dispositivi mobile. L'evento *on_sys_init*, come già

spiegato in precedenza viene richiamato solo all'avvio e quindi dobbiamo eseguire le operazioni di inizializzazione di tutti gli oggetti che saranno utilizzati nel seguito.

In particolare, facendo riferimento al **Listato 2**:

- la riga 4 permette di impostare l'indirizzo IP assegnato al modulo (modificare questo valore per adattarlo alla vostra sotto-rete);
- la riga 6 serve per indirizzare il socket di interesse, poiché è possibile istanziare fino a 16 socket contemporaneamente;
- le righe 7-9 servono per definire la dimensione del buffer per la ricezione e trasmissione dei dati; il parametro passato come argomento indica il numero di pagine riservate, quindi per conoscere il numero di byte bisogna usare la seguente formula $N \times 256 - 16$ (nell'esempio presentato $4 \times 256 - 16 = 1008$ byte);
- le righe 12-13 servono per indicare che si vuole aprire un socket TCP in ascolto sulle porte 2000 e 3000;
- la riga 14 indica al modulo di accettare le connessioni in entrata da qualunque indirizzo IP.

Ora che l'oggetto sock è stato correttamente istanziato e configurato è necessario gestire la ricezione dei dati. Anche in questo caso TIDE ci mette a disposizione un evento (*on_sock_data_arrival*) che scatta in automatico ogni volta che sono presenti dei dati nel buffer di ricezione.

All'interno di questa subroutine dobbiamo inserire il codice per il controllo delle linee e per fare ciò definiamo un

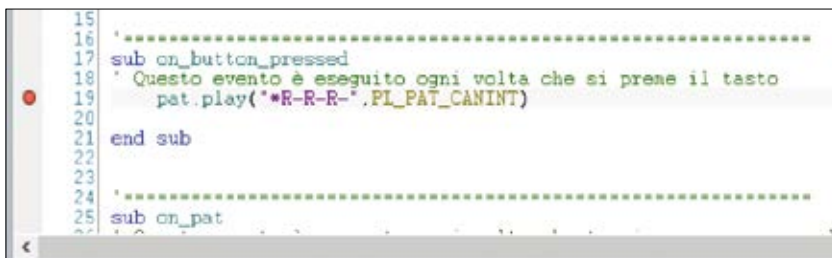


Fig. 8 - Per inserire un breakpoint è sufficiente posizionarsi sulla riga desiderata e premere il tasto funzione F5.

semplice protocollo formato da 3 soli caratteri: XYZ; in esso, X e Y rappresentano la linea (per esempio XY = 32 indica la linea 32 della board) e Z indica lo stato della stessa (1 per lo stato logico alto e 0 per il livello basso).

Sempre con riferimento al

Listato 2:

- le righe 23-24 si occupano di estrarre l'informazione della linea e del suo stato rispettivamente; il metodo utilizzato è `sock.getdata(N)`, dove N è il numero di byte da estrarre dal buffer di ricezione (il dato estratto dal metodo `getdata` è fornito come stringa ed è perciò necessario trasformarlo per indirizzare la linea da pilotare, tramite la funzione `val`);
- le righe 25-27 impiegano l'oggetto `io` per il controllo della linea, la quale viene dapprima indirizzata tramite il metodo `io.num` e successivamente configurata come uscita tramite il metodo `io.enable = YES` (se avessimo voluto configurarla come ingresso sarebbe stato sufficiente assegnare NO); infine si imposta il valore tramite `io.state`;
- le righe 28-29 implementano invece una risposta del modulo che ci conferma l'avvenuta esecuzione del comando; i metodi da usare sono `sock.setdata("OK")` che inserisce nel buffer di tra-

missione il dato e `sock.send` che invia effettivamente il dato.

DEBUGGING TRAMITE SOFTWARE IO NINJA

Ora che avete visto come creare un socket TCP e scambiare

dati tra il server (EM1001) ed il client, vi state probabilmente chiedendo come poter verificare il funzionamento dell'applicazione e quindi controllare le nostre linee di uscita. Ancora una volta la Tibbo vi viene incontro con un'applicazione dal nome curioso **IO Ninja**, scaricabile gratuitamente dal link web <http://tibbo.com/ioninja/>. Dopo aver terminato il download e la relativa installazione, avviate l'applicazione e impartite il comando *File>New Session*.

Listato 2 - Apertura di un socket TCP sulla porta 2000 per il controllo delle linee.

```

1. sub on_sys_init
2. ' Questo evento è generato una sola volta all'avvio del dispositivo
3.   dim w as word
4.   net.ip="10.218.136.154" 'Indirizzo IP del dispositivo
5.   'Allocazione della memoria per il socket da istanziare
6.   sock.num=0
7.   sock.rxbufrrq(4) 'buffer di ricezione 4*256-16=1008 bytes
8.   sock.txbufrrq(4) 'buffer di trasmissione 4*256-16=1008 bytes
9.   sys.buffalloc 'Allocazione della memoria
10.  'Impostazione della tipologia di socket
11.  sock.num=0

12.  sock.protocol=PL_SOCKET_PROTOCOL_TCP 'Tipologia TCP
13.  sock.localportlist="2000,3000" 'Porte in ascolto
14.  sock.inconmode=PL_SOCKET_INCONMODE_ANY_IP_ANY_PORT
15.  'Sequenza di avvio
16.  pat.play("B-B-B-----",YES)
17. end sub
18. '=====
19. sub on_sock_data_arrival
20. ' Questo evento è richiamato quando sono presenti dati nel buffer di ricezione
21.   dim linea as word
22.   dim stato as word
23.   linea = val(sock.getdata(2))
24.   stato = val(sock.getdata(1))
25.   io.num = linea
26.   io.enabled = YES
27.   io.state = stato
28.   sock.setdata("OK")
29.   sock.send
30. end sub

```

Link Utili

Manuale on-line Tibbo: <http://goo.gl/Lbsp03>

Corso di programmazione Tibbo: <http://goo.gl/RyQ1IO>

Download di TIDE: <http://goo.gl/7Gy2qj>

Funzionamento dei socket: <http://goo.gl/KZkkOu>

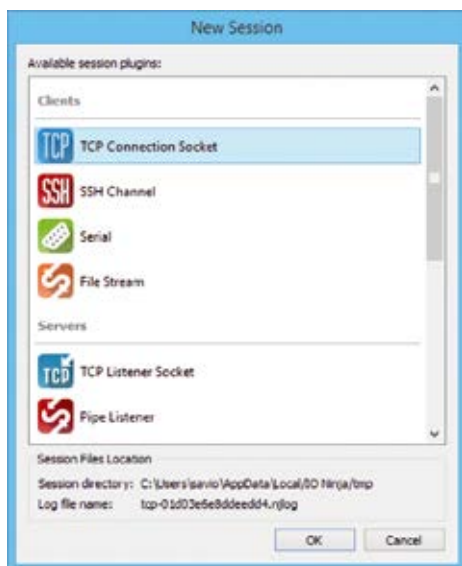


Fig. 9 - Dopo aver avviato l'applicazione IO Ninja, selezionate File>New Session e quindi TCP Connection Socket.

Nella schermata che si presenta (Fig. 9) selezionate TCP Connection Socket.

La Fig. 10 ci mostra la suc-

cessiva finestra; tutto ciò che bisogna fare è riassunto di seguito (avviate preventivamente l'applicazione sulla board EM1001 con il codice riportato nel Listato 2 tramite il solito tasto F5).

1. Digitate nel campo *Address* l'indirizzo IP e la relativa porta a cui connettersi, la stessa che avete indicato nel codice dell'applicazione (ad esempio, 10.218.136.153:2000).
2. Fate clic sull'icona con le due spine per avviare la connessione con EM1001 (apparirà a video il messaggio di avvenuta connessione, che ci conferma l'instaurarsi del socket).
3. Digitate il comando nella forma spiegata nel paragrafo precedente (ad esempio 320 per ordinare al modulo

EM1001 di mettere a zero la propria linea 32).

4. Fate clic sul pulsante *Send* per inviare effettivamente il comando.
5. Attendete la risposta di "OK" del modulo EM1001, che vi conferma che il comando è stato eseguito con successo.

DEBUGGING TRAMITE PORTA SERIALE

Il codice riportato nel Listato 2 è un'ottima base per l'introduzione di un altro oggetto molto utile, che serve per il controllo delle quattro porte seriali di cui è dotato l'EM1001: stiamo parlando dell'oggetto *ser*. Alcuni possibili impieghi di un'interfaccia seriale asincrona (UART) sono convertitore seriale-ethernet, una porta consolle

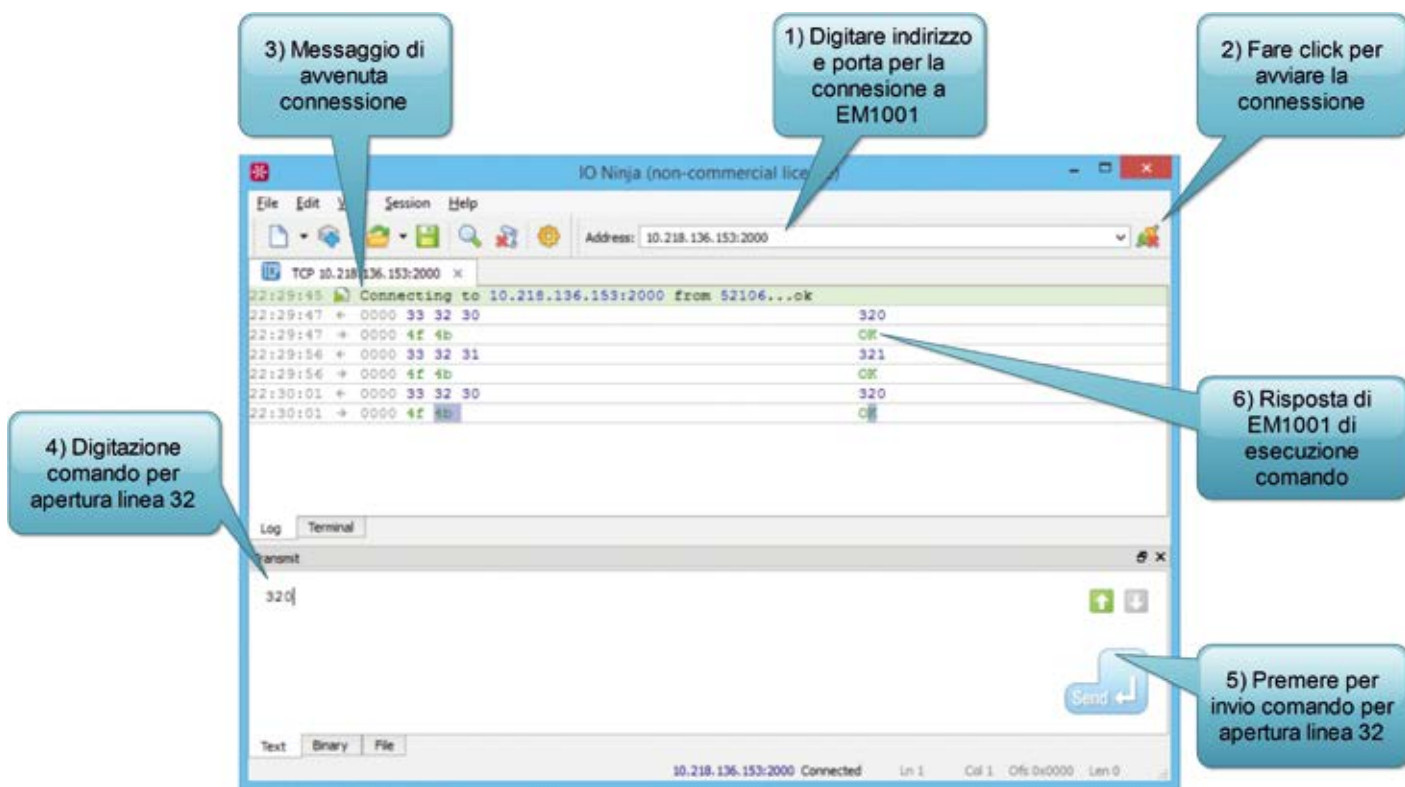


Fig. 10 - Sequenza di operazioni da eseguire con il software IO Ninja per comunicare con il modulo EM1001 tramite socket TCP sulla porta 2000.

Listato 3 - Debugging tramite seriale.

```
1. sub on_sys_init
2. .... `Stesso codice del listato 2 per allocazione memoria dell'oggetto sock
3. ` Allocazione memoria per interfaccia seriale
4.     ser.num=0
5.     ser.rxbufrrq(4)
6.     ser.txbufrrq(4)
7.     sys.buffalloc
8. .... `Stesso codice del listato 2 per impostazioni dell'oggetto sock
9. ` Impostazione parametri per interfaccia seriale
10. ser.flowcontrol=ENABLED
11. ser.baudrate=ser.div9600/4
12. ser.enabled=YES
13. `Sequenza di avvio
14. pat.play("B-B-B-----",YES)
15.     ` Messaggio di avvio
16.     ser.setdata("Avvio Applicazione...")
17.     ser.send
18. end sub
19. `=====
20. sub on_sock_data_arrival
21. ` Questo evento è richiamato quando sono presenti dati nel buffer di ricezione
22. ....
23. dim messaggio_debug as string
24. ....
25. messaggio_debug = "Linea: " + str(linea) + " Stato: " + str(stato)
26. ser.setdata(messaggio_debug)
27. ser.send
28. end sub
```

(per uso debug) e soprattutto per il controllo di periferiche/ sensori con interfaccia UART. Supponiamo quindi di voler estendere il codice precedente, usando la seriale 1 del modulo come porta console e cioè inviando ad essa dei messaggi di debug che ci indichino le varie operazioni svolte dalla nostra applicazione: praticamente sostituirò il software IO Ninja. I due strumenti, IO Ninja e porta console, non si escludono però a vicenda poiché il primo può essere impiegato durante la fase di sviluppo del codice mentre il secondo per fare debug on-site. Esaminiamo le righe di codice riportate nel **Listato 3** (per semplicità di lettura non sono state riportate le istruzioni già presentate nel **Listato 2** e relative all'oggetto sock):

- la riga 4 serve a selezionare il numero di porta seriale da utilizzare; come già spiegato nell'introduzione il modulo dispone di 4 interfacce, in questo esempio

utilizzeremo la prima (vale a dire *ser.num = 0*);

- le righe 5-6 servono a dimensionare i buffer di ricezione e trasmissione della porta seriale, proprio come fatto per il socket;
- l'istruzione della riga 7 è comune agli oggetti *sock* e *ser* e quindi sarà inserita una sola volta nel codice;
- le righe 10-12 impostano la velocità ed il controllo del flusso dati della porta seriale;
- dopo aver inizializzato l'oggetto, le righe 16-17 si occupano di inviare un messaggio di "Avvio Applicazione" su seriale, che ci conferma il corretto avvio del modulo.

Per verificare la corretta esecuzione del programma, possiamo ancora una volta sfruttare l'applicativo IO Ninja (in alternativa si possono usare Hyperterminal, PuTTY o altri simili), ma a differenza dell'esempio precedente dobbiamo scegliere l'opzione

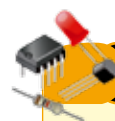
Serial dal menu riportato in **Fig. 9**. L'unica differenza nella finestra mostrata, rispetto al caso della connessione TCP, è nella barra dell'indirizzo IP che in questo caso viene sostituita con i parametri connessione seriale:

- Port; selezionare la porta seriale per la comunicazione con EM1001;
- Baud Rate; 9600;
- Flow Control; RTS/CTS.

CONCLUSIONI

Ora che abbiamo appreso le caratteristiche hardware della board EM1001, le funzionalità dell'ambiente di sviluppo TIDE ed i primi esempi operativi, siamo pronti a progettare un sistema basato su EM1001 che sia in grado di controllare la temperatura all'interno della nostra abitazione ed attivare o disattivare di conseguenza il sistema di riscaldamento, il tutto condotto con un controllo remoto del sistema.

Avrete ormai capito che abbiamo messo le basi per la costruzione del nostro termostato intelligente Smartem. ■

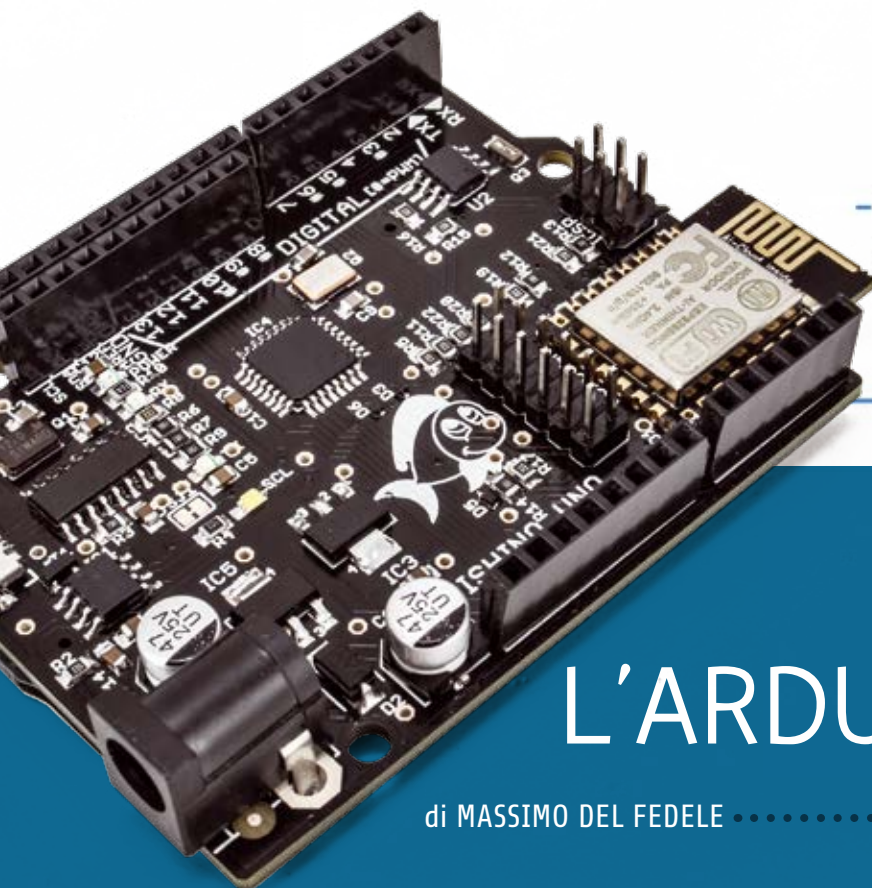


per il MATERIALE

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>

Continuamo la presentazione della board Fishino, mostrando le principali funzioni delle librerie e degli esempi d'uso. Seconda puntata.



FISHINO, L'ARDUINO DIVENTA WIRELESS

di MASSIMO DEL FEDELE

Nel numero scorso abbiamo presentato la scheda **Fishino UNO**, una board compatibile con la diffusissima Arduino UNO dotata però di connettività WiFi, slot per microSD ed RTC incorporati.

In questo secondo articolo iniziamo la descrizione delle librerie software disponibili, mostrando le principali funzioni con alcuni semplici esempi d'uso.

Come anticipato, sia il firmware della scheda che le librerie software sono in continua fase di sviluppo, quindi consigliamo di eseguire spesso l'aggiornamento di entrambi.

LE LIBRERIE

Per poter sfruttare tutte le caratteristiche di **Fishino** occorre ovviamente disporre di una serie di librerie software che ne gestiscano tutti i componenti aggiuntivi. Se per la scheda SD card e il Real Time Clock (RTC) esistono già nella suite di Arduino le corrispondenti librerie, questo non vale per il modulo WiFi ESP12, per il quale ne abbiamo sviluppate di apposite.

Inizieremo quindi da queste ultime, fornendo comunque successivamente anche qualche dettaglio su quelle già disponibili nell'IDE.

Le librerie fornite e liberamente scaricabili dal sito sono:

- Libreria '**Fishino**'
- Libreria '**FishinoWebServer**'
- Libreria '**Flash**'

quest'ultima libreria, che abbiamo inserito per comodità nel download pur essendo reperibile in rete, è necessaria per il funzionamento delle due precedenti.

LIBRERIA 'FISHINO'

Partiamo con la descrizione della libreria (che potete scaricare dal sito della rivista www.elettronica.in.it) che contiene tutta la gestione a basso e medio livello del modulo

WiFi di Fishino.

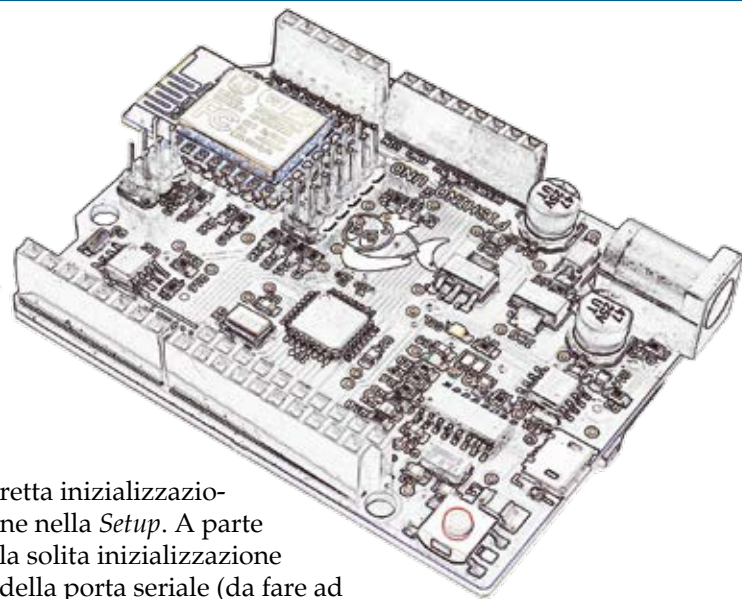
Questa definisce 3 classi:

- **FishinoClass** e la relativa variabile globale **Fishino**
- **FishinoClient**
- **FishinoServer**

Il progetto è in continua evoluzione e ben supportato da una comunità molto attiva anche su FaceBook (<https://www.facebook.com/groups/fishino>); anche grazie a questa in seguito verranno aggiunte la **FishinoUdp**, **FishinoAnalog**, **FishinoDigital** e **FishinoSerial** per gestire rispettivamente le comunicazioni internet tramite socket Udp, l'ingresso analogico, gli I/O digitali e la porta seriale hardware aggiuntiva presenti sul modulo WiFi.

Iniziamo la descrizione della classe **FishinoClass** (istanza singola nella variabile globale **Fishino**), con degli esempi pratici di utilizzo delle varie funzioni. Con **bool Fishino.reset()** si inizializza il modulo WiFi inviandogli un reset software. Obbligatorio ad inizio sketch per garantire un avvio corretto del modulo. Ritorna TRUE se il modulo è stato correttamente inizializzato, FALSE altrimenti.

La funzione di reset esegue inoltre un controllo sulla versione del firmware installata. In caso di versione troppo datata viene inviato un messaggio di errore alla porta seriale ed il programma viene bloccato. Nel **Listato 1** vediamo un esempio di cor-



retta inizializzazione nella *Setup*. A parte la solita inizializzazione della porta seriale (da fare ad inizio setup), si notano:

- l'inizializzazione dell'interfaccia SPI
- il **Fishino.reset()** dell'inizializzazione del modulo

La prima è stata lasciata volutamente manuale per poter cambiare la velocità di comunicazione, nel caso siano presenti altri shields che utilizzano la stessa interfaccia. In questo caso si è impostata la massima velocità disponibile.

La sezione contenente la chiamata **Fishino.reset()** inizializza il modulo e visualizza un messaggio di corretta inizializzazione sulla seriale o, in caso di problemi, visualizza l'errore e blocca lo sketch. Attenzione, il modulo WiFi **NON** parte senza que-

Listato 1

```
void setup()
{
  // apre la porta seriale e ne attende l'apertura
  // consigliabile da eseguire come primo comando per poter visualizzare
  // eventuali messaggi di errore sul monitor seriale
  Serial.begin(115200);

  // attende l'apertura della porta seriale.
  // Necessario solo per le boards Leonardo
  while (!Serial)
    ;

  // inizializza il modulo SPI
  SPI.begin();
  SPI.setClockDivider(SPI_CLOCK_DIV2);

  // resetta e testa il modulo WiFi
  if(Fishino.reset())
    Serial.println("Fishino WiFi RESET OK");
  else
  {
    Serial.println("Fishino RESET FAILED");

    // attende per sempre
    while(true)
      ;
  }

  Serial.println("Fishino WiFi web server");

  .....<resto dello sketch>...
```

Listato 2

```
...<parte precedente dello sketch>...

Fishino.setMode(STATION_MODE);

...<resto dello sketch>...
```

Listato 3

```
...<parte precedente dello sketch>...
while(true)
{
  if(Fishino.begin("MIO_SSID", "MIA_PASSWORD")) {
    Serial.println("Connected to MIO_SSID");
    break;
  }
  else {
    Serial.println("Failed connecting to MIO_SSID");
    Serial.println("Retrying.....");
  }
}
...<resto dello sketch>...
```


sto comando.

Le funzioni **bool Fishino.setMode(uint8_t mode)** e **uint8_t Fishino.getMode(void)** impostano (o leggono) la modalità di funzionamento del modulo (**Listato 2**), che può essere una delle seguenti:

- **STATION_MODE**
modalità stazione. Richiede la presenza di un router WiFi a cui connettersi. È la modalità normale.
- **SOFTAP_MODE**
Permette la creazione di un access point a cui connettersi. Utile in mancanza di un'infrastruttura di rete esistente.
- **STATIONAP_MODE**
Modalità doppia, il modulo funziona sia da stazione, collegandosi ad un router esistente, che da access point.

Per eseguire la connessione all'access point/router si utilizza **bool Fishino.begin(SSID, PASSWORD)**, dove al posto di SSID va inserito il punto di accesso e al posto di PASSWORD la chiave per accedervi (quest'ultima può essere una stringa vuota se non è richiesta).

Per controllare se la board Fishino è correttamente connessa il comando è **uint8_t Fishino.status()**. La funzione ritorna TRUE se la connessione ha avuto successo, FALSE altrimenti. Nello spezzone di codice presente nel **Listato 3** viene tentata la connes-

CARATTERISTICHE TECNICHE

- Alimentazione: 12 Vcc o USB
- Completamente compatibile con Arduino Uno
- Scheda WiFi a bordo, con possibilità di funzionamento in modalità stazione, access point o entrambe contemporaneamente
- Interfaccia per schede di memoria MicroSD a bordo
- RTC (Real Time Clock) a bordo con batteria al litio di mantenimento
- Sezione di alimentazione a 3,3 V potenziata
- Connettore aggiuntivo sfalsato in modo da risolvere il problema dell'incompatibilità di Arduino con le schede millefori.

Listato 4

```
uint32_t connectTime;
void setup()
{
    ....<parte precedente dello sketch>....
    connectTime = millis();
}

void loop()
{
    // controlla la connessione ogni 10 secondi
    if(millis() - connectTime > 10000) {

        // resetta il tempo
        connectTime = millis();

        // controlla se connesso
        uint8_t stat = Fishino.status();

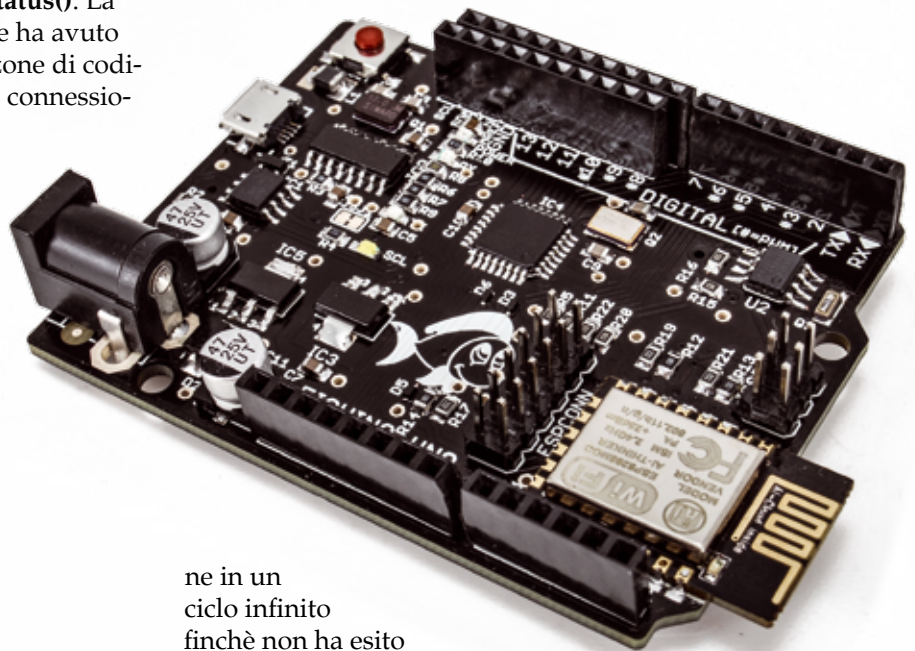
        // se non connesso, tenta la riconnessione
        if(stat != STATION_GOT_IP) {
            if(Fishino.begin("MIO SSID", "MIA_PASSWORD"))
                stat = STATION_GOT_IP;
        }

        // se connesso, salva i dati sul server
        if(stat == STATION_GOT_IP) {
            salvaDatiSulServer(); // funzione da definire!!!
        }

        // qui legge i sensori e li memorizza sulla SD
        leggiSensoriEMemorizza(); // funzione da definire!!!
    }
}
```

Listato 5

```
Fishino.config(IPAddress(192, 168, 1, 251));
```



ne in un ciclo infinito finché non ha esito positivo.

Questo tipo di connessione (eseguita nella *Setup*) è adatto ad una postazione fissa, ovviamente. Nel caso si utilizzi il Fishino in mobilità, è consigliabile spostare la connessione nel *loop()* e tentarla ogni tanto mentre si fanno altre attività.

In questo modo è possibile, ad esempio, raccogliere

Listato 6

```
Serial.print("Il mio IP è : ");  
Serial.println(Fishino.localIP());
```

dei dati da un sensore, memorizzarli sulla scheda SD e, quando viene rilevata una connessione funzionante, inviarli ad un computer remoto (**Listato 4**). In questo esempio (volutamente abbreviato), nella *setup()* viene letto il tempo corrente e salvato nella variabile *connectTime*; successivamente nel *loop()* viene controllato il tempo passato (*millis()* - *connectTime*) e quando questo supera i 10 secondi viene eseguito un test sulla connessione; se non connesso si tenta la connessione al server e, in caso di successo viene eseguita una funzione (da definire) che salva in rete i dati letti in precedenza.

Il loop continua successivamente tramite un'altra funzione (anch'essa da definire) che legge qualche sensore e memorizza i dati localmente, ad esempio su una scheda SD.

Con uno sketch simile è possibile quindi realizzare un semplice datalogger che non solo legge e memorizza su scheda SD i dati ma che, in presenza di una connessione di rete, è in grado di salvarli in modo totalmente automatico ad intervalli di tempo predefiniti.

Per configurare un IP statico ed eventualmente i servers DNS, il gateway e la subnet della rete locale si utilizzano queste funzioni:

```
bool Fishino.config(IPAddress local_ip)  
bool Fishino.config(IPAddress local_ip, IPAddress  
dns_server)  
bool Fishino.config(IPAddress local_ip, IPAddress  
dns_server, IPAddress gateway)  
bool Fishino.config(IPAddress local_ip, IPAd-  
dress dns_server, IPAddress gateway, IPAddress  
subnet)
```

In pratica, la prima è adoperata per impostare un IP statico, se necessario.

Nel **Listato 5** vediamo come è possibile impostare un IP statico su 192.168.1.251

Se non utilizzata l'IP sarà richiesto dinamicamente al router.

E' ovviamente possibile anche disconnettersi dalla rete WiFi. Il comando per eseguire questa operazione è **bool Fishino.disconnect(void)**.

Qui di seguito, invece, alcune funzioni utilizzate per controllare i parametri della connessione, in particolare per leggere il MAC del modulo WiFi la funzione è **const uint8_t* Fishino.macAddress(void)**

Mentre per la lettura dell'IP acquisito dal modulo WiFi (utile nel caso si sia impostato un IP dinamico) il comando da richiamare è **IPAddress Fishino.localIP()** come mostrato ad esempio nel **Listato 6**. Per leggere la maschera della sottorete e dell'indirizzo IP del gateway potete richiamare queste funzioni:

IPAddress Fishino.subnetMask()

IPAddress Fishino.gatewayIP()

Le funzioni indicate sopra sono state nominate in modo assolutamente simile a quelle delle analoghe funzioni delle librerie Ethernet e WiFi di Arduino,

Listato 8

```
Serial.print("Sono connesso a : ");  
Serial.println(Fishino.SSID());
```

per poter semplificare il porting del codice esistente. Tuttavia le potenzialità superiori di **Fishino**, ed in particolar modo la possibilità di funzionare anche in modalità Access Point senza bisogno di un'infrastruttura esistente, hanno reso necessario studiare nuove funzioni per quanto riguarda la modalità **Stazione** tra cui:

```
bool Fishino.setStaIP(IPAddress ip)  
bool Fishino.setStaMAC(uint8_t const *mac)  
bool Fishino.setStaGateway(IPAddress gw)  
bool Fishino.setStaNetMask(IPAddress nm)
```

Mentre per la modalità **Access Point** sono state create:

```
bool Fishino.setApIP(IPAddress ip)  
bool Fishino.setApMAC(uint8_t const *mac)  
bool Fishino.setApGateway(IPAddress gw)  
bool Fishino.setApNetMask(IPAddress nm)  
bool Fishino.setApIPInfo(IPAddress ip, IPAd-  
dress gateway, IPAddress netmask)
```

In particolare, l'ultima permette di impostare tutti i parametri IP del **Fishino** utilizzato come router

Listato 7

```
Fishino.setApIPInfo(  
    IPAddress(192, 168, 100, 1), // IP del Fishino  
    IPAddress(192, 168, 100, 1), // Gateway del Fishino, solitamente come l'IP  
    IPAddress(255, 255, 255, 0) // Netmask (maschera di sottorete) del Fishino  
);
```


Listato 9

```
uint8_t n = Fishino.scanNetworks();
if(n){
    Serial.print("Trovate ");
    Serial.print(n);
    Serial.println(" reti wifi:");
    for(int i = 0; i < n; i++) {
        Serial.print("Rete #");
        Serial.print(i);
        Serial.print(" : ");
        Serial.println(Fishino.SSID(i));
    }
}
else
    Serial.println("Nessuna rete WiFi trovata");
```

WiFi in un comando singolo (**Listato 7**).

Vedremo come usarle a fine articolo con un esempio completo. Per poter leggere i dati della connessione WiFi, quali lo SSID del router a cui ci si è connessi, il MAC del medesimo (BSSID), la potenza in dBm del segnale (RSSI) ed il tipo di protezione della rete, potete utilizzare queste funzioni:

const char* Fishino.SSID()

const uint8_t* Fishino.BSSID()

int32_t Fishino.RSSI()

uint8_t Fishino.encryptionType()

Come mostrato ad esempio nel **Listato 8**.

Esistono poi alcune funzioni utilizzate per eseguire una lista delle reti WiFi disponibili con le loro caratteristiche:

uint8_t Fishino.scanNetworks()

Questa operazione esegue una scansione delle reti WiFi disponibili e ritorna il numero di reti trovate. Una volta eseguita la *scanNetworks*, è possibile utilizzare le seguenti funzioni, che hanno come parametro il numero della rete da esaminare (numero di reti ritornate da *scanNetworks()* - 1).

La funzione **const char* Fishino.SSID(uint8_t net-**

workItem) ritorna invece lo SSID, ovvero il nome della rete richiesta come mostrato nel **Listato 9**.

Questo esempio stampa sulla seriale un'elenco delle reti wireless trovate.

Per sapere il tipo di protezione della rete il comando da usare è **uint8_t Fishino.encryptionType(uint8_t networkItem)**.

E' possibile anche sapere la potenza del segnale della rete richiesta con **int32_t Fishino.RSSI(uint8_t networkItem)**.

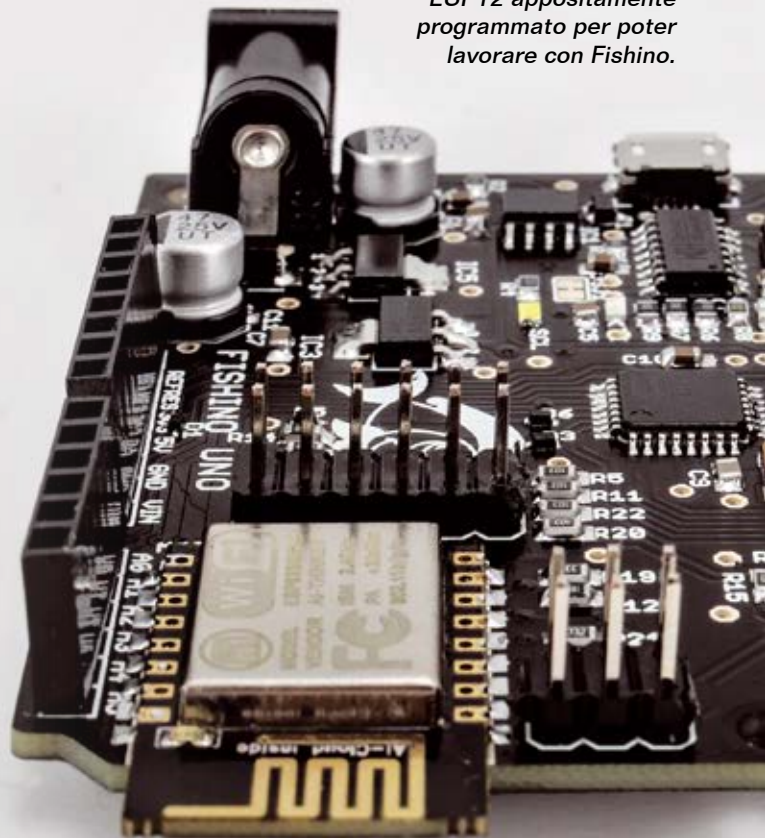
Nella classe **FishinoClass** sono presenti altre funzioni meno utilizzate che tralasciamo per brevità. Il codice della libreria è comunque ben commentato e di facile interpretazione.

CLASSI FISHINOCIENT E FISHINOSERVER

Queste due classi sono l'equivalente delle EthernetClient/WiFiClient ed EthernetServer/WiFiServer delle shield ethernet e WiFi di Arduino, e l'uso è praticamente identico.

Ad esempio, per inviare una richiesta ad una pagina web e stampare sulla seriale la risposta vediamo il **Listato 10**.

Dettaglio del modulo WiFi ESP12 appositamente programmato per poter lavorare con Fishino.



Listato 10

```
// tenta la connessione al server
FishinoClient client;
if (client.connect("www.google.com", 80)) {

    Serial.println("connected to server");

    // esegue un request Http
    client.println("GET /search?q=arduino HTTP/1.1");
    client.println("Host: www.google.com");
    client.println("Connection: close");
    client.println();

    // legge la risposta finchè il client resta connesso
    do {
        // finchè ci sono bytes in arrivo.....
        while (client.available()) {

            // legge un carattere dal server
            char c = client.read();

            // .... e lo invia alla seriale
            Serial.write(c);

        }
        while(client.connected());
        Serial.println("Client disconnected");
    }
}
```



```
#include <Flash.h>
#include <FishinoUdp.h>
#include <FishinoSockBuf.h>
#include <Fishino.h>
#include <SPI.h>

// CONFIGURAZIONE SKETCH -- ADATTARE ALLA PROPRIA RETE WiFi //
// WiFi SSID e PASSWORD
// potete cambiarle entrambe, verranno utilizzate
// per la creazione dell'infrastruttura WiFi
#define My_SSID "FISHINO"
#define My_PASS ""
// FINE CONFIGURAZIONE

// crea un server in ascolto sulla porta 80 (HTTP standard)
FishinoServer server(80);

void setup()
{
    // apre la porta seriale
    Serial.begin(115200);

    // attende l'apertura della porta seriale.
    // Necessario solo per le boards Leonardo
    while (!Serial)
        ;

    // inizializza il modulo SPI
    SPI.begin();
    SPI.setClockDivider(SPI_CLOCK_DIV2);

    // resetta e testa il modulo WiFi
    if(Fishino.reset())
        Serial << F("Fishino WiFi RESET OK\r\n");
    else
    {
        Serial << F("Fishino RESET FAILED\r\n");

        // attende per sempre in caso di errore
        while(true)
            ;
    }

    Serial << F("Fishino WiFi AP web server\r\n");

    // imposta la modalità SOFT AP (crea una rete autonoma)
    Fishino.setMode(SOFTAP_MODE);

    // ferma il server DHCP, necessario per impostare l'IP della rete
    Fishino.softApStopDHCPserver();

    // imposta i parametri IP dell'access point
    // in questo caso la rete viene creata su 192.168.100.0-255
    // ed il Fishino assume l'IP 192, 168, 100, 1
    Fishino.setApIPInfo(
        IPAddress(192, 168, 100, 1), // IP
        IPAddress(192, 168, 100, 1), // gateway
        IPAddress(255, 255, 255, 0) // netmask
    );

    // imposta i parametri di connessione WiFi, ovvero nome della rete(SSID)
    // e password. Se non avete modificato l'esempio, la rete sarà chiamata FISHINO
    // e sarà una rete aperta, senza password
    Fishino.softApConfig(My_SSID, My_PASS, 1, false);

    // riavvia il server DHCP in modo da poter fornire gli indirizzi
    // in automatico a tutte le stazioni che si connettono
    Fishino.softApStartDHCPserver();

    // inizia l'attesa delle connessioni
    server.begin();
}

void loop()
{
    // attende nuovi clienti
    FishinoClient client = server.available();

    if (client)
    {
        Serial.println("new client");

        // ogni richiesta http termina con una linea vuota
        boolean currentLineIsBlank = true;
        while (client.connected())
        {
            if (client.available())
            {
                char c = client.read();
                Serial.write(c);
            }
        }
    }
}
```

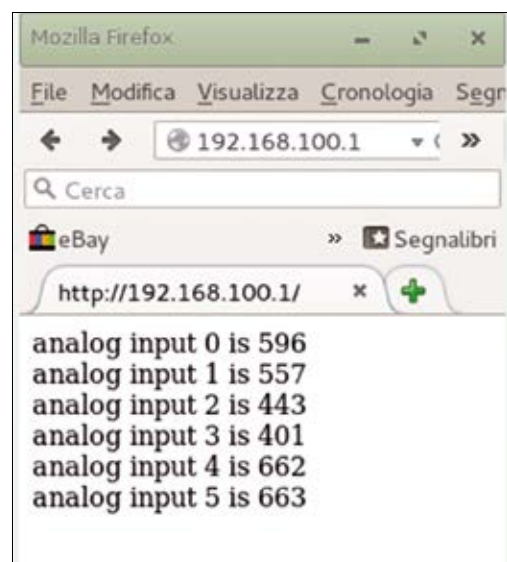
A conclusione dell'articolo, presentiamo un esempio completo che mostra una delle caratteristiche più interessanti di **Fishino**, ovvero la possibilità di creare una propria infrastruttura di rete senza

bisogno di un router esterno, alla quale connettersi in mobilità, per esempio con un cellulare. Un'applicazione simile potrebbe essere usata, ad esempio, per monitorare alcuni sensori all'aperto

Fig. 1



Fig. 2



```

// se si è arrivati a fine linea (carattere 'newline' ricevuto
// e la linea è vuota, la richiesta http è terminata
// quindi è possibile inviare una risposta
if (c == '\n' && currentLineIsBlank)
{
    // invia uno header standard http
    client.println("HTTP/1.1 200 OK");
    client.println("Content-Type: text/html");
    client.println("Connection: close"); // la connessione verrà chiusa automaticamente una volta inviata la risposta
    client.println("Refresh: 5"); // aggiorna la pagina automaticamente ogni 5 secondi
    client.println();
    client.println("<!DOCTYPE HTML>");
    client.println("<html>");

    // invia il valore di tutti i pins analogici
    for (int analogChannel = 0; analogChannel < 6; analogChannel++)
    {
        int sensorReading = analogRead(analogChannel);
        client.print("analog input ");
        client.print(analogChannel);
        client.print(" is ");
        client.print(sensorReading);
        client.println("<br />");
    }
    client.println("</html>");
    break;
}
if (c == '\n')
{
    // inizio di una nuova linea
    currentLineIsBlank = true;
}
else if (c != '\r')
{
    // sono stati ricevuti dei caratteri nella linea corrente
    currentLineIsBlank = false;
}
}
// lascia tempo al browser per ricevere i dati
delay(1);

// chiudi la connessione
client.stop();
Serial.println("client disconnected");
}
}

```

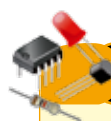
tramite un cellulare da una certa distanza, realizzando così dispositivi completamente portatili. Un'altra interessante applicazione potrebbe essere un comando remoto via WiFi sempre tramite browser web sul cellulare.

L'esempio, nel **Listato 11** crea una rete WiFi "volante", con nome (SSID) **'FISHINO'**, senza password (rete open) ed avvia un piccolo server che su richiesta fornisce una lettura dei sei ingressi analogici di Fishino. Una volta lanciato lo sketch, occorre selezionare la rete wireless **FISHINO** tra le reti disponibili (**Fig. 1**) e aprendo l'indirizzo **192.168.100.1** sul browser si ottiene il risultato visualizzato in **Fig. 2**. Gli esempi qui riportati sono comunque contenuti, insieme ad altri, nella libreria **Fishino**.

Un'ultima nota sugli I/O occupati dalle estensioni, e che non vanno utilizzati come I/O quando sono attivi i componenti aggiuntivi. Il modulo **WiFi** utilizza i seguenti pins: **7, 10, 11, 12 e 13**. È disattivabile completamente con un ponticello tra il pin **CH_PH** del connettore **ESP** e la massa. La scheda **microSD** utilizza i seguenti pins: **4, 11, 12 e 13** ed impone che il pin **7** sia impostato ad output digitale. Per libe-

rare i ports usati basta non inserire alcuna scheda nel connettore. Il modulo **RTC** comunica via i2c utilizzando i pins **SCL** e **SDA**, abbinati nell'UNO ai ports analogici **A4** ed **A5**.

Continueremo in un prossimo articolo con la descrizione della libreria **FishinoWebServer** che permette la realizzazione di un piccolo ma completo server web, che è la base dell'esempio di Home Automation mostrato in breve nel numero scorso. ■



per il MATERIALE

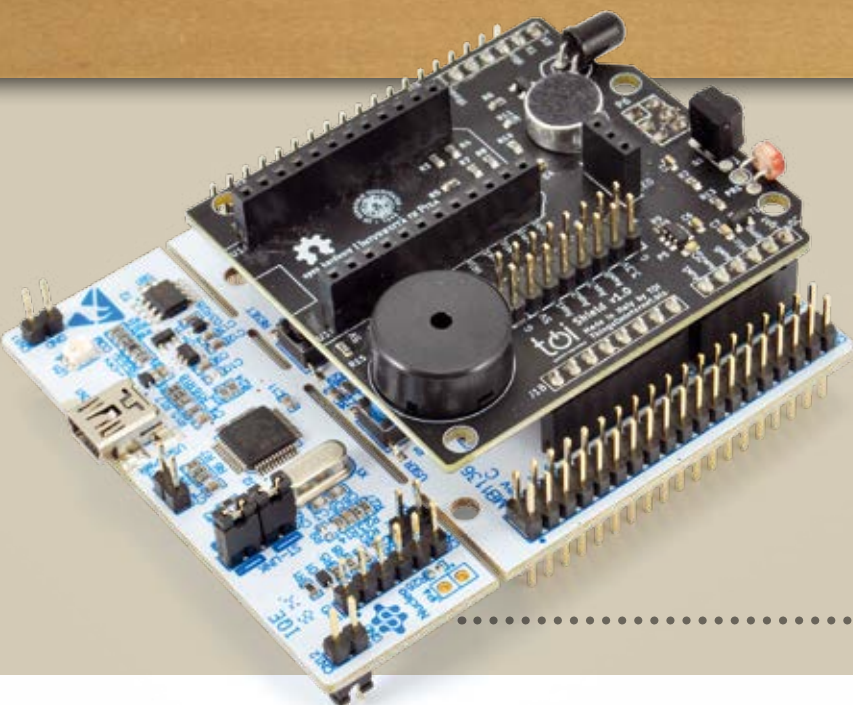
La board Fishino (cod. FISHINOUNO) viene fornita montata e collaudata. Può essere acquistata presso Futura Elettronica al prezzo di Euro 36,00. Il prezzo si intende IVA compresa.

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

L'IoT SI PROGRAMMA IN PYTHON CON VIPER



VIPER APP



VIPER è una suite software per la programmazione di oggetti interattivi, pronti per il cloud e l'Internet of Things. VIPER permette di sviluppare in Python sulle più diffuse piattaforme di prototipazione, usando i paradigmi e le feature tipici della programmazione ad alto livello.

..... di LUIGI FRANCESCO CERFEDA

Al giorno d'oggi siamo tutti connessi ad Internet. Non solo le persone, tutto è connesso o può essere connesso: è la rivoluzione dell'IoT. Televisioni, lampade e termostati "smart" si stanno affacciando prepotentemente nelle nostre case e già oggi è possibile avere sveglie sincronizzate con il servizio di traffico cittadino e lavatrici che decidono per noi la funzione di lavaggio migliore in base alle condizioni atmosferiche.

Se oggi sono pochi milioni gli oggetti di questo tipo fra pochi anni saranno miliardi. Tutti i maggiori produttori di oggetti di largo consumo stanno pensando a come acquisire informazioni utili e come analizzare il grande quantitativo di dati che questi oggetti si scambieranno al fine di offrire nuovi servizi. La maggior parte di questi dispositivi ha al suo interno un microcontrollore che permette di far funzionare l'apparecchio, ma anche di colle-

garlo alla rete. Tuttavia, affinché questo "Internet of Things" sia pronto per fare il salto di qualità, sono necessari progetti pratici in grado di rendere intelligenti anche gli oggetti più semplici della vita quotidiana.

Ad esempio, è possibile collegare il tostapane alla Rete? E la ciotola del cane? E la culla del bebè? Tutto questo è certamente fattibi-

le dal punto di vista tecnico. Di sicuro non mancano le piattaforme di prototipazione come Arduino, la ST Nucleo, il Core ed il Photon della Particle. Inoltre, affinché ogni oggetto della vita quotidiana diventi intelligente è necessario che sia munito anche di sensori e attuatori in grado di renderlo interattivo.

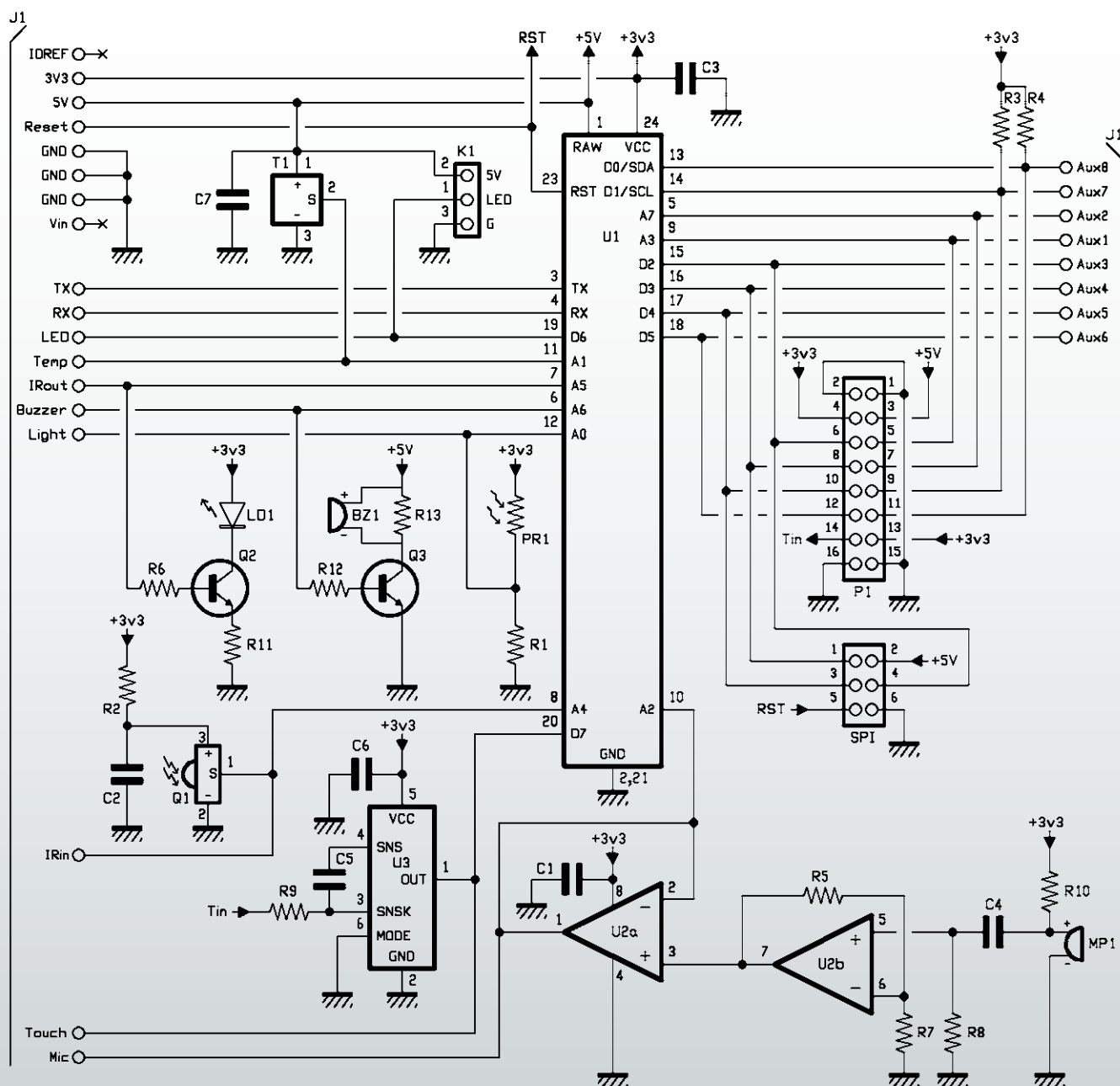
Grazie al TOI Shield, ogni ogget-

to di uso comune potrà scambiare informazioni con l'ambiente e, ovviamente, le persone. Vedremo il TOI Shield nel dettaglio nei paragrafi a seguire.

ABBATTERE LE BARRIERE TRA HARDWARE E SOFTWARE

Parte delle difficoltà che ostacolano lo sviluppo dell'IoT risiede nelle barriere "linguistiche" tra

[schema ELETTRICO]





VIPER e Kickstarter

VIPER ed il TOI Shield sono due progetti sviluppati dalla startup TOI, acronimo di Things on Internet. TOI vede nel proprio progetto la possibilità di un cambiamento di prospettiva all'interno del mondo dell'IoT: per TOI infatti non c'è una rete di oggetti che comunicano tra loro, ma una rete in cui l'uomo sta al centro e porta i propri oggetti su internet perché da loro vuole ricevere le informazioni che gli servono (conoscere la temperatura di casa, sapere se il cane sta abbaiando infastidendo i vicini, sapere se annaffiare le piante o prendere l'ombrello nonostante la mattinata di sole). La startup è cresciuta nel FABLAB Pisa ed ha al suo fianco professori universitari, aziende e più di 50 alpha tester. Il team è nato 4 anni fa nell'Università di Pisa, dove è stato concepito un nuovo paradigma per la progettazione e per l'Internet delle cose. Adesso è diventato realtà gra-

zie al successo della campagna Kickstarter conclusa a Marzo 2015.

Oggi il progetto VIPER è cresciuto, e si è trasformato in una startup con sede a New York. Il core team di VIPER è composto da quattro elementi, tre ingegneri e uno sviluppatore informatico: Daniele Mazzei del Centro di ricerca "E. Piaggio", Giacomo Baldi dell'Azienda Ospedaliera Universitaria pisana, Gualtiero Fantoni e Gabriele Montelisciani del dipartimento di Ingegneria Civile e Industriale. Al core team si aggiunge poi un gruppo di giovanissimi entusiasti che collaborano sia nell'area tecnica che nell'area di comunicazione. E poi gli Evangelists, persone che si stanno innamorando della filosofia di VIPER e che si propongono di "diffondere il verbo" in Italia e all'estero. Con le loro conoscenze tecniche e di mercato, aiutano VIPER a crescere nella comunità maker, ma anche in ambito industriale. Su Kickstarter i cercatori di fondi raccontano il loro progetto e chiedono un finanziamento alla folla, se il progetto piace i

backers (sostenitori) fanno delle donazioni per vederlo realizzato ed in cambio ricevono una scheda, un libro, una maglietta o un prodotto. TOI ha messo VIPER ed il TOI Shield su Kickstarter. La scelta di utilizzare la piattaforma americana deriva dalla volontà di creare un gruppo di developer per accelerare e migliorare il prodotto. L'obiettivo era raggiungere un finanziamento di 20.000 euro in circa 30 giorni per migliorare ulteriormente il progetto, ma anche cercare co-sviluppatori ed evangelisti, sempre in un'ottica open source. E TOI ce l'ha fatta!

La campagna Kickstarter è stata un importante banco di prova per comprendere il potenziale di mercato di VIPER. Oltre 400 utenti da 25 paesi diversi hanno creduto nel prodotto. A singoli finanziatori, quali designer professionisti, maker e ricercatori, si sono affiancate importanti aziende nazionali e internazionali operanti nei settori dello sviluppo software, elettronica embedded e consumer electronics.

L'uomo, che vuole prototipare in modo semplice e flessibile le sue idee, e l'hardware, che richiede istruzioni specifiche. Il vero problema è che servirebbero competenze informatiche "spinte", fin troppo spinte per il mercato a cui queste board elettroniche si riferiscono. D'altra parte se vogliamo oggetti interattivi dobbiamo sforzarci di realizzare dispositivi capaci di eseguire molte attività contemporaneamente. Ciò richiede di apprendere paradigmi come il real-time, gli interrupt, le callback, non semplicissimi da imparare e da gestire. Ma se da una parte le schede di prototipazione sono adesso accessibili a tutti, dall'altra molte persone incontrano non poche difficoltà quando approcciano a tali schede: in primo luogo devono saper programmare bene in C o negli altri linguaggi di programmazione che ogni scheda richiede.

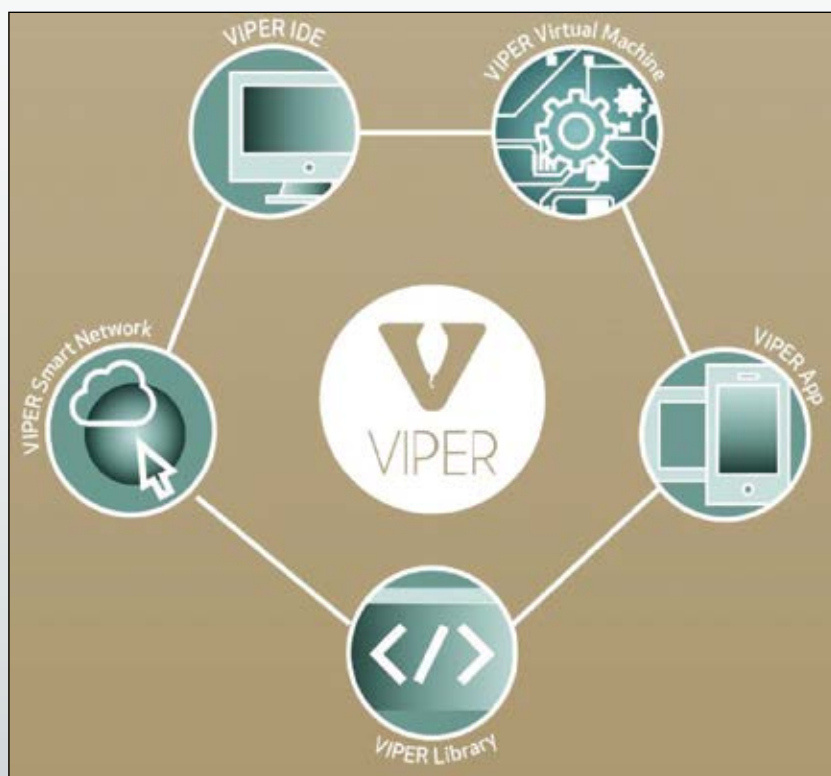


Fig. 1 - La VIPER suite.

VIPER: dettagli tecnici

Uno dei vantaggi più semplici da intuire nel confronto tra Arduino e VIPER è la differenza tra *delay()* e *sleep()*. In Arduino/Wiring usare la funzione *delay()* ha un effetto collaterale: interrompe il ciclo di esecuzione di uno "sketch" per il numero di millisecondi assegnato come argomento. Cioè Arduino letteralmente si blocca e non rileva alcun input così come non invia alcun output. In VIPER, invece, la funzione *sleep()* sospende il thread corrente ma tutti gli altri threads continuano ad essere eseguiti! Dovendo girare su microcontrollori, la VIPER VM estende Python con delle funzioni particolari (built-in functions) in grado di gestire i pin di Input e Output dei microcontrollori. Queste funzioni ricordano quelle tipiche del linguaggio Arduino, ma sono più flessibili. Vediamone alcune. La funzione *analogRead()* è fornita come built-in per facilitare il passaggio da Arduino/Wiring a VIPER. Comunque, la sintassi preferibile per leggere i valori dai pin analogici in VIPER è:

```
import adc # import the adc driver
x = adc.read(pin, samples=1)
```

Se "samples" è uguale a 1 (o non espresso), la funzione ritorna il valo-

re intero letto sul pin. Se "samples" è maggiore di 1, ritorna una tupla di interi con dimensione uguale a quella impostata.

La funzione *analogWrite()* di Arduino fornisce una semplice interfaccia per la gestione della PWM, impropriamente chiamata "analog output". In particolare, permette di variare il duty cycle dell'onda quadra ma non fornisce nessun controllo sul periodo dell'onda quadra. VIPER, invece, attraverso una sintassi molto semplice, permette di controllare sia il periodo che il duty cycle dell'onda. Anche qui la funzione *analogWrite()* è fornita come built-in per facilitare il passaggio da Arduino/Wiring a VIPER. Tuttavia il modo migliore di usare il PWM in VIPER è:

```
import pwm # import the pwm driver
pwm.write(pin, period, pulse, time_
unit=MILLIS)
```

Lo stato del pin è periodicamente alternato tra LOW e HIGH in base ai parametri impostati. In particolare:

- "period" imposta la durata dell'onda quadra
- "pulse" imposta la durata del duty cycle
- "time_unit" imposta l'unità di misura di "period" e "pulse"

Questo esclude già una buona parte delle persone che potrebbero vedere nell'IoT un rinnovamento e un'evoluzione della propria professionalità: artigiani, webdesigner, ma anche artisti e vetrinisti.

Dalla parte opposta della barricata anche per i programmatori più esperti l'attuale situazione spesso risulta essere poco flessibile, perché basta che abbiano la necessità di cambiare scheda per fare il proprio progetto, che devono ricominciare a scrivere il codice dall'inizio nel linguaggio di programmazione che la nuova scheda richiede.

Per far sì che ogni utente possa programmare i propri oggetti interattivi occorre cambiare

approccio alla loro programmazione.

Nella speranza di semplificare il modo con il quale queste "cose" sono progettate e collegate ad Internet, VIPER riunisce una serie di componenti software per automatizzare meglio il processo di sviluppo di oggetti interattivi. VIPER, disponibile per il download in modalità open source al sito www.viperize.it/download, è una suite di sviluppo multiplatforma (Linux, Windows e Mac) che permette di programmare la maggior parte delle schede a 32 bit attualmente presenti sul mercato: dalle schede professionali usate in campo industriale alle schede di prototipazione per hobbysti più note al momento

come Arduino DUE, UDOO, Particle e ST Nucleo.

Inoltre VIPER si programma in Python, che è tra i linguaggi di programmazione più semplici e intuitivi esistenti. Python infatti è usato sia nelle scuole che nelle università e nel mondo professionale.

Grazie alla sua sintassi asciutta e potente ed al supporto multiplatforma, è utilizzato per moltissime tipologie di applicazioni: dal networking, al web, alla grafica. La peculiarità di VIPER è che riesce a coniugare la semplicità d'uso a performance di tipo professionale proprio come è successo per Photoshop, il più noto software di editing di foto, oggi usato indistintamente da studenti, fotoamatori e fotografi professionisti.

VIPER garantisce un rapido sviluppo e un'efficace integrazione con sensori, attuatori e servizi cloud riducendo tempi ed energie per lo sviluppo.

Il codice utilizzato in fase di prototipazione, infatti, può essere riutilizzato sia per l'elettronica selezionata per la produzione di serie, che per futuri aggiornamenti del progetto o addirittura su un'altra scheda fra quelle supportate.

VIPER sta per "Viper Is Python Embedded in Realtime", acronimo ricorsivo che descrive chiaramente i principi di base del progetto.

L'idea di Viper nasce dall'esigenza di ottimizzare tali processi, attraverso un'interfaccia web per gestire e programmare le schede, una macchina virtuale "real time" e una vasta libreria di funzioni pronte per l'uso.

La suite include inoltre un'App per mobile che permette di controllare gli oggetti VIPER attraverso gli smartphone senza dover sviluppare App specifiche

per ogni progetto. Il tutto è già "nativamente connesso al cloud". Tale configurazione permette di ridurre tempi e le energie per lo sviluppo.

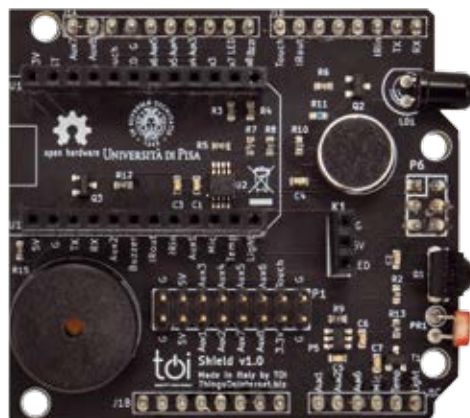
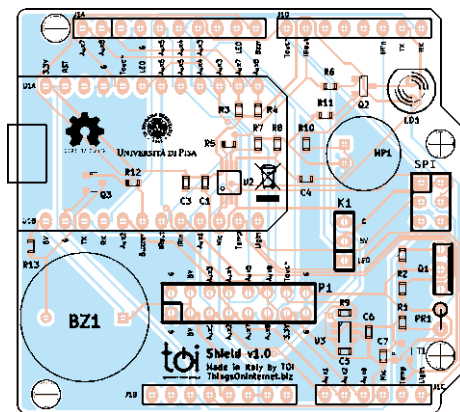
Nel dettaglio VIPER è composta da:

- VIPER IDE: un ambiente di sviluppo browser-based e multi piattaforma, con sincronizzazione e storage dei progetti nel cloud;
- VIPER VM: una Macchina Virtuale Real-Time per microcontrollori a 32 bit, scritta in Python 3, con supporto per il multi-threading. Compatibile con tutte le schede basate su chip ARM 32 bit come Arduino Due, UDOO, Particle, STNucleo.
- VIPER Library: insieme di moduli che include CC3000 del Wi-Fi dello Spark Core e dello shield wifi di Adafruit, la stampante termica della Adafruit/Sparkfun, l'anello di LED NeoPixel, RTTTL smart melody player, una libreria per segnali di tipo Streams, come pure i protocolli TCP ed UDP.
- VIPER APP: un'app, disponibile in versione sia iOS sia Android, che fa da interfaccia per pilotare la board programmate con VIPER, senza dover usare interruttori o potenziometri. L'interfaccia di comando è personalizzabile per ogni singolo progetto, perché tecnicamente l'app è un client HTML che visualizza i templa-

te definiti dagli script Python all'interno della memoria dei VIPER Objects.

Rispetto ad altre soluzioni, VIPER è nativamente cross-platform e compatibile con tutte le board a 32bit basate su ARM32, con un'infinità di sensori e kit per il rapid prototyping. Inoltre integra un sistema operativo real-time e consente il data analytics su board e componenti. Diversamente da chi fa prototipazione di prodotto con suite di sviluppo Arduino o RaspberryPi, VIPER rende facili da usare i componenti industriali per creare prodotti commerciali pronti per il mercato. Con poche righe di codice, gli utenti possono sviluppare una vasta gamma di

[piano di MONTAGGIO]



Elenco Componenti:

R1: 20kohm (0603)
R2: 390 ohm (0603)
R3, R4: 4,7 kohm (0603)
R5: 100 kohm (0603)
R6: 3,6 kohm (0603)
R7: 470 ohm (0603)
R8, R9: 10 kohm (0603)
R10: 2,2 kohm (0603)
R11: 6,8 ohm (0603)
R12, R13: 1 kohm (0603)
C1÷C4: 100 nF ceramico (0603)

C5: 10 nF ceramico (0603)
C6, C7: 100 nF ceramico (0603)
PR1: Fotoresistenza 500kohm (VT90N2)
MP1: Microfono ABM-707-RC
T1: MCP9700AT-E/TT
Q1: TSOP34838
Q2, Q3: BC818
U1: Spark Core
U2: LMV358IDGKRG4
U3: AT42QT1010-TSHR
BZ1: Buzzer ABT-436-RC

LD1: LED IR SFH-4546

Varie:

- Strip maschio 2x8 vie
- Strip maschio 6 vie
- Strip maschio 8 vie (2 pz.)
- Strip maschio 10 vie
- Strip femmina 12 vie (2 pz.)
- Strip femmina 3 vie
- Strip femmina 2x3 vie
- Circuito stampato S1226

STMicroelectronics e STNUCLEO

Oggi tutti i produttori di microcontrollori e soluzioni professionali hanno iniziato a seguire la strada tracciata da Arduino proponendo piattaforme di prototipazione basate sui loro microcontrollori ed espressamente compatibili con il pinout di Arduino. Una di queste è la STM32 Nucleo di ST Microelectronics (**Fig. A**). Sostanzialmente, la scheda è composta da due macroblocchi: un programmatore ST-Link, usato per caricare il firmware sul microcontrollore target ed eseguire il debugging, e la scheda di sviluppo vera e propria. Quest'ultima presenta due gruppi di connettori: i connettori Morpho, su cui sono rigirati tutti i pin del micro target e 4 connettori compatibili con il pinout di Arduino, permettendo quindi l'aggancio del

TOI Shield. Le schede di sviluppo Nucleo STM32 STMicroelectronics rappresentano per gli utenti un modo conveniente e flessibile per provare nuove idee e creare prototipi con qualsiasi linea di microcontrollori STM32, scegliendo tra le varie combinazioni di prestazioni, consumo elettrico e funzionalità.

La board STM32 Nucleo è la piattaforma ideale per gli sviluppatori embedded grazie alla flessibilità, convenienza, facilità d'uso, connettività e compatibilità con le più diffuse risorse del settore. Il fatto di essere compatibile con Arduino, però, non deve trarre in inganno. Questa board, infatti, non è una piattaforma entry-level e certamente la sua programmazione out-of-the-box risulta ostica ai neofiti.

E qui entra in gioco VIPER, in grado di abilitare anche i meno esperti alla programmazione della scheda.

Tra l'altro, VIPER si basa su ChibiOS (www.chibios.org), un sistema operativo real-time per microcontrollori, rilasciato in open source alcuni anni fa da Giovanni di Sirio, ingegnere elettronico che lavora a STMicroelectronics. Ovviamente tra Giovanni e il team di VIPER è iniziata una stretta collaborazione.

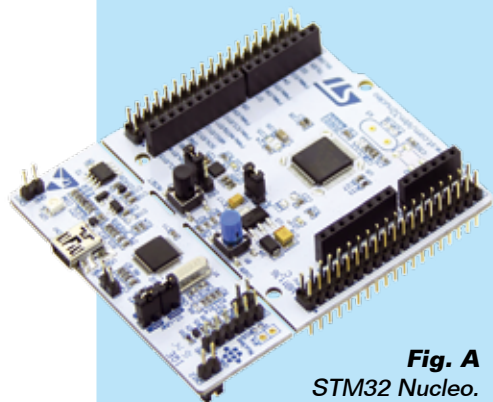


Fig. A
STM32 Nucleo.

applicazioni del mondo dell'internet degli oggetti, che vanno dalla domotica al fitness, dalla robotica alla nautica, dalle applicazioni industriali ai controlli manutentivi da remoto, fino all'agricoltura intelligente.

COME FUNZIONA VIPER

In estrema sintesi, Viper funziona cancellando il firmware standard delle board su cui viene installato e sostituendolo con un sistema operativo embedded (ChibiOS) con multithreading,

sul quale si appoggia una virtual machine Python 3.

L'operazione di "flash" del firmware nella board, che diventa quindi "viperizzata", si esegue direttamente dall'interfaccia dell'ambiente di programmazione di VIPER.

L'ambiente di sviluppo comprende le librerie per gestire nativamente i componenti hardware e software più utili: dagli shield Wi-Fi ai protocolli di rete, dalla sensoristica in generale ai servizi cloud per l'Internet of

Things come ad esempio Carriots (www.carriots.com). VIPER è in grado di connettere gli oggetti alla rete via TCP e UDP permettendo l'uso dell'HTTP e, sulle schede dove è consentita la crittografia, anche in HTTPS. Così i dati che passano fra gli oggetti e la rete sono al sicuro! VIPER ad oggi gira su Arduino Due, Spark Core, Spark Photon, UDOO, ST Nucleo, ma potenzialmente su una miriade di altre schede: tutte quelle basate sull'architettura ARM. Infatti, indipendentemente dalla scheda tu scrivi in Python, poi VIPER adatta il tuo programma alla scheda che hai scelto.

Usare VIPER è molto semplice:

1. si scarica l'installer per Windows, Linux o iOS dalla pagina di download di VIPER (www.viperize.it/download);
2. si installa e lancia l'applicazione tramite l'icona apparsa sul desktop;
3. si crea un utente VIPER attraverso il pulsante dedicato in alto a destra. Controlla la tua mail e verifica il tuo nuovo account cliccando sull'apposito link (registrarsi permette di partecipare alla VIPER community, salvare i propri progetti on-line nel VIPER Cloud e ricevere automaticamente gli aggiornamenti);
4. si connette la board ed eventualmente la si rinomina a piacimento in modo da poterla riconoscere facilmente in futuro;
5. per rendere la board utilizzabile bisogna "viperizzarla", cioè installare su di essa la VIPER VM (Virtual Machine), attraverso l'apposito pulsante nella topbar del VIPER IDE;
6. si crea un nuovo progetto o si clona un esempio (**Fig. 2**), si compila lo script e lo si carica nella board.

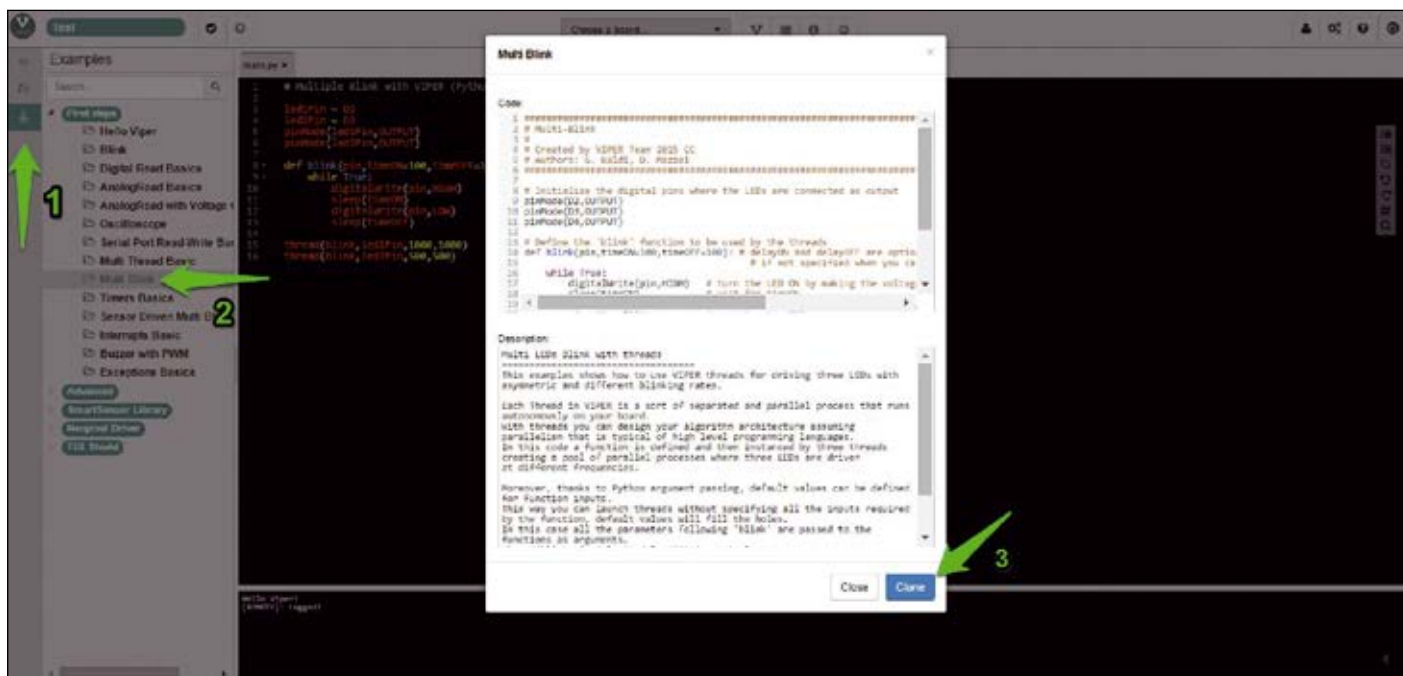


Fig. 2 - Clonazione di un progetto in VIPER.

Un esempio molto semplice in cui si possono intravedere le potenzialità di VIPER, è il “Multiple LED Blink” (Listato 1). Far lampeggiare un LED è conosciuto fra gli addetti come l’“Hello World” dei dispositivi embedded... ma con VIPER possiamo farlo decisamente più complesso, più divertente ed addirittura con meno fatica. Siamo infatti in grado di far lampeggiare molti LED a frequenze diverse utilizzando diversi thread separati, in poche righe di codice.

Sì, avete letto bene: possiamo usare i thread su schede tipo Arduino! Non dovremo più scrivere tutto il nostro codice in un singolo loop logico, tipico della programmazione imperativa. Con VIPER si può procedere ad una implementazione multithread reale! Questo esempio mostra come usare i thread di Viper per guidare tre LED con frequenze di lampeggiamento asimmetriche e diverse tra loro. Una volta che viene eseguito lo script, vengono creati tre thread

(oltre al thread principale, che è sempre disponibile), ognuno dei quali esegue una specifica istanza della funzione “blink”, con parametri diversi. Ogni thread in VIPER è una sorta di processo separato e parallelo che funziona autonomamente sulla vostra scheda. Un thread richiede di eseguire una funzione come input per la definizione. La stessa funzione può essere istanziata da vari thread dandovi la possibilità di scrivere codice molto conciso e leggibile. Con i

Listato 1

```
# Initialize the digital pins where the LEDs are connected as output
pinMode(D2,OUTPUT)
pinMode(D3,OUTPUT)
pinMode(D4,OUTPUT)

# Define the 'blink' function to be used by the threads
# delayON and delayOFF are optional parameters, used as default if not specified when you call the function
def blink(pin,timeON=100,timeOFF=100):
    while True:
        digitalWrite(pin,HIGH) # turn the LED ON by making the voltage HIGH
        sleep(timeON)          # wait for timeON
        digitalWrite(pin,LOW)  # turn the LED OFF by making the voltage LOW
        sleep(timeOFF)         # wait for timeOFF

# Create three threads that execute instances of the 'blink' function.
thread(blink,D2)              # D2 is ON for 100 ms and OFF for 100 ms, the default values of delayON and delayOFF
thread(blink,D3,200)          # D3 is ON for 200 ms and OFF for 100 ms, the default value of delayOFF
thread(blink,D4,500,200)      # D4 is ON for 500 ms and OFF for 200 ms
```


thread è possibile progettare l'architettura del vostro algoritmo sfruttando il parallelismo che è tipico dei sistemi ad alto livello. Inoltre, grazie ai parametri di passaggio in Python, i valori di default possono essere definiti per gli ingressi di ogni funzione. In questo modo è possibile avviare i thread senza specificare tutti gli input richiesti dalla funzione. In assenza di valori esplicitati, i valori di default serviranno a "riempire i buchi". In questo caso tutti i parametri della funzione 'blink' vengono passati alle funzioni come argomenti.

TOI SHIELD

TOI Shield è uno shield compatibile con i pin-out di Arduino e Particle (Fig. 3) ed è nativamente supportato dall'ambiente di sviluppo VIPER. Presenta un insieme di sensori e attuatori di uso comune (Fig. 4), tra cui sensori per il rilevamento del tocco, LED a infrarossi, microfono, sensore di luminosità e sensore di temperatura.

Inoltre, sono presenti delle porte denominate "Aux" che consentono l'uso di altri sensori come quelli di Groove, ThinkerKit, Phidgets, e le strisce LED NeoPixel di Adafruit.

Avere una serie di sensori a bordo e tutte le porte di comunicazione più usate pronte per essere collegate consente ai

makers, artigiani e progettisti di sviluppare facilmente i loro oggetti intelligenti senza doversi curare del circuito elettrico e delle saldature o dei cavetti volanti nella breadboard. Basta collegare un Arduino Due o un Photon o una ST Nucleo allo shield ed iniziare a lavorare con uno degli esempi VIPER che si trovano nella sua libreria. Il routing garantisce la compatibilità del TOI shield con la maggior parte degli shield di Arduino come gli shield Ethernet, Wi-Fi e BLE. Ciò significa che il TOI Shield si può facilmente agganciare in cima alla vostra "pila di shield" senza avere alcun problema di compatibilità dei pin. La scheda dispone di sensori di temperatura, suono, luce, ma anche un touch capacitivo e un ricevitore IR. A bordo ha anche un cicalino piezoelettrico e un LED emettitore IR (attuatori). Le porte di comunicazione a bordo sono SPI e I²C-Bus; completano la scheda una linea Analog Aux e una Digital Aux, oltre a un'uscita PWM e una Porta per strisce LED Neopixel Adafruit.

Ricevitore IR

Il ricevitore IR è basato sul componente TSOP34838 che può essere utilizzato per leggere segnali dai comuni dispositivi IR a 38 kHz, come televisori, condizionatori e home theatre. Il

ricevitore IR richiede un pin di ingresso digitale ed è connesso al pin D2 sul Arduino e al pin A4 su layout Particle.

Temperatura

Il sensore di temperatura appartiene alla serie MCP9700/9701 e converte la temperatura in tensione analogica. Un vantaggio chiave di questa serie di termistori è la resistenza alle capacità parassite e l'abilità di azionare grandi carichi capacitivi. Il pin di uscita dell'integrato è collegato ad A4 su Arduino e A1 sui formati Particle.

Sensore di luce

Il TOI Shield integra un fotoresistore, noto anche come resistenze luce dipendente (LDRs). Le LDR sono resistenze variabili a basso costo dove la resistenza cambia a seconda della quantità di luce che colpisce la sua superficie. In ambienti scuri la resistenza è alta; in ambienti luminosi la resistenza è bassa.

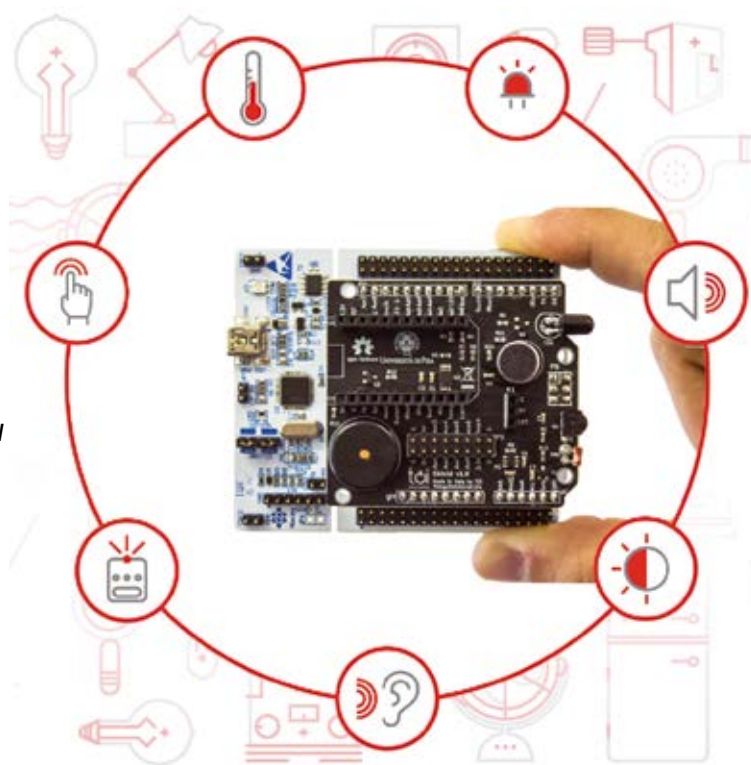
Possono essere utilizzati per rispondere agli eventi quali la transizione dal giorno alla notte (e viceversa), per applicazioni di automazione domestica e giardinaggio, e sono spesso utilizzati per controllare l'illuminazione stradale.

Il circuito è un semplice partitore di tensione che consente una lettura abbastanza precisa della

Fig. 3 - TOI Shield è multiboard compatibile.



Fig. 4
TOI Shield
su ST
Nucleo.



resistenza del fotoreistore. La resistenza opposta alla LDR è stata impostata a 20k perché l'LDR scelto ha una resistenza che varia da 12k a 36k ohm. Il sensore è collegato ad A5 su Arduino e A0 sui formati di Particle.

Trasduttore acustico

Un piccolo microfono electret amplificato con un guadagno di circa 200 volte viene utilizzato per consentire il rilevamento di voci, suoni ambientali e rumori simili. Il circuito implementato è basato su un semplice stadio di amplificazione seguito da un buffer. L'uscita dell'amplificatore del microfono è collegata al pin A3 Arduino e all'A2 del Particle.

Touch capacitivo

Il TOI Shield integra un sensore touch capacitivo basato sul chip AT42QT1010, un componente integrato utilizzato per convertire qualsiasi superficie capacitiva in un pulsante. Il chip gestisce il monitoraggio di una zona conduttiva e viene attivato

quando questa viene toccata. Fino a quando viene rilevato un tocco (ad esempio da un dito), l'AT42QT1010 mantiene il segnale in uscita a livello alto. Quando il tocco si interrompe, il segnale è portato al valore basso. L'uscita del sensore AT42QT1010 è collegata al pin D7 sia su Arduino che su Particle. Il pin del touch può essere collegato ad una superficie metallica da sensorizzare attraverso il connettore di espansione (P1).

Buzzer Piezoelettrico

Questo trasduttore montato sul TOI Shield serve per generare segnali acustici, toni e allarmi. Il cicalino è alimentato attraverso la linea 5 V per evitare interferenze con i sensori. Il buzzer è guidato attraverso i pin collegati al piedino D8 su Arduino e sull'A6 di Particle.

LED emettitore IR

Un LED IR guidato da un BJT è collegato al pin D6 su Arduino e al pin A5 sui Particle. Un'onda PWM è utilizzata per creare impulsi a infrarossi che possono essere ricevuti e decodificati da comuni ricevitori IR.

LED Adafruit Neopixel

Nel TOI Shield è stato inclusa anche una porta dedicata al collegamento delle strisce LED Neopixel di Adafruit. Il pin di alimentazione è collegato al Vin dei Particle e al pin 5V di Arduino. I Pin da utilizzare sono (GND, Vin, LED control pin). Il pin di comando per il controllo digitale del LED è collegato al pin D9 su Arduino e al pin D6 sui Particle. Se la configurazione include più di 24 LED, al fine di evitare eventuali danni alla vostra scheda, è necessario utilizzare un alimentatore a 5V esterno per alimentare i LED.

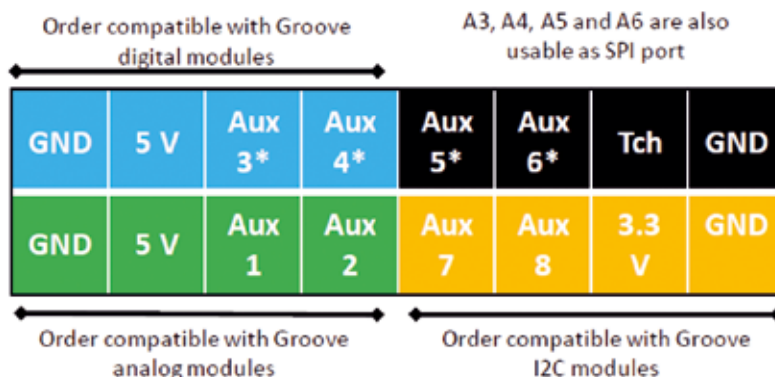


Fig. 5 - Expansion Connector.

AUX PINs PROPERTIES					
CHANNEL\FEATURES	ARDUINO DUE	ST NUCLEO	Particle Core	Particle Photon	NOTE
AUX 1	ADC, DIO	ADC, DIO, PWM	ADC, DIO	ADC, DAC, DIO	ADC guaranteed on all boards
AUX 2	ADC, DIO	ADC, DIO, PWM	ADC, PWM, DIO	ADC, PWM, DIO	ADC guaranteed on all boards
AUX 3	PWM, DIO, SPI_MOSI*	ADC, DIO, PWM, SPI_MOSI	DIO (5V tolerant)	DIO, PWM, CAN Rx, SPI_MOSI	*Connected also with Arduino SPI connector SPI_MOSI
AUX 4	PWM, DIO, SPI_MISO*	ADC, DIO, PWM, SPI_MISO	DIO (5V tolerant)	DIO, PWM, SPI_MISO	*Connected also with Arduino SPI connector SPI_MISO
AUX 5	PWM, DIO, SPI_SCK*	ADC, DIO, PWM, SPI_SCK	DIO (5V tolerant)	DIO, SPI_SCK	*Connected also with Arduino SPI connector SPI_SCK
AUX 6	ADC, DIO	ADC, DIO, PWM	DIO (5V tolerant)	DIO, SPI_SS	
AUX 7	DIO, I2C SCL	DIO, I2C SCL	DIO, PWM, I2C SCL	DIO, PWM, I2C SCL, CAN Tx	I2C guaranteed on all boards
AUX 8	DIO, I2C SDA	DIO, I2C SDA	DIO, PWM, I2C SDA	DIO, PWM, I2C SCL	I2C guaranteed on all boards

Tabella 1 - Proprietà dei pin Aux in base alla board utilizzata.

Expansion Connector

Si tratta di un connettore maschio 8X2 (Fig. 5) su cui sono disponibili tutti i pin Aux (Tabella 1) in una configurazione compatibile con le uscite analogiche e digitali dei moduli Seedstudio Groove. Il connettore di espansione permette anche un facile utilizzo di moduli Phidgets e Thinkerkit, e mette a disposizione anche le linee di alimentazione 3,3V, 5V e GND. Il connettore di espansione consente anche di esporre i pin SPI ma

purtroppo questa funzione, disponibile in Arduino e nella Nucleo, non è disponibile sul Photon della Particle a causa di limitazioni hardware. Il TOI shield include anche una porta su cui utilizzare l'I²C. La porta ha un ordine pin compatibile con i sensori e attuatori I²C Groove (3,3, SCL, SDA, GND) che consente l'uso *plug and play* di questi moduli. I dispositivi I²C di altri produttori possono essere utilizzati facendo attenzione ad effettuare i collegamenti adeguati.

TOI Shield Library

Il TOI shield è stato sviluppato per semplificare la vita agli sviluppatori di nuovi oggetti intelligenti. Per questo motivo il team di VIPER ha sviluppato una libreria di controllo dedicata al TOI Shield.

La libreria di controllo del TOI Shield viene rilasciata insieme con la suite VIPER e comprende varie funzioni di alto livello pronte da essere utilizzate. Alcuni esempi: la media dei segnali dei sensori e sistemi di

Listato 2

```
#####
# TOI Shield basics
#
# Created by VIPER Team 2015 CC
# Authors: L. Rizzello, G. Baldi, D. Mazzei
#####

import streams
import adc
from drivers.toishield import toishield

streams.serial()

# toishield defines pin names in a board independent manner
# let's use them to read raw sensors values

while True:
    print(" Microphone:",adc.read(toishield.microphone_pin))
    print("    Light:",adc.read(toishield.light_pin))
    print("Temperature:",adc.read(toishield.temperature_pin))
    print("    Touch:",digitalRead(toishield.touch_pin))
    # aux pins are also accessible!
    print("    AUX1:",adc.read(toishield.aux1.ADC))
    print("-"*40)
    sleep(500)

# this scripts runs on every supported board, without a single change...cool isn't it? :)
```



Fig. 6 - Casi di studio VIPER e TOI Shield.

filtraggio dei dati da esso provenienti.

Esistono anche serie di funzioni intelligenti come ad esempio lo "smart motion detection" che permette di utilizzare il sensore di luminosità come sensore di presenza o il "doppio clic" che consente il rilevamento di due tocchi ravvicinati su un pulsante attraverso il sensore capacitivo. Vediamo un semplice esempio di utilizzo della libreria TOI Shield su VIPER nel **Listato 2**.

Infine analizziamo ora alcuni possibili scenari d'uso (**Fig. 6**): l'unione tra VIPER e TOI Shield potrà aiutare le persone a monitorare la propria casa in maniera wireless senza dover fare tracce nei muri, permettendo di sapere, ad esempio, se i bambini sono rientrati da scuola, di spegnere il riscaldamento se è stato lasciato inavvertitamente acceso o di controllare dal lavoro se è stato chiuso il gas, il tutto sul proprio PC o smartphone.

Prossimamente proporremo alcuni semplici progetti pratici sviluppati con firmware scritto in VIPER (**Fig. 7**). ■

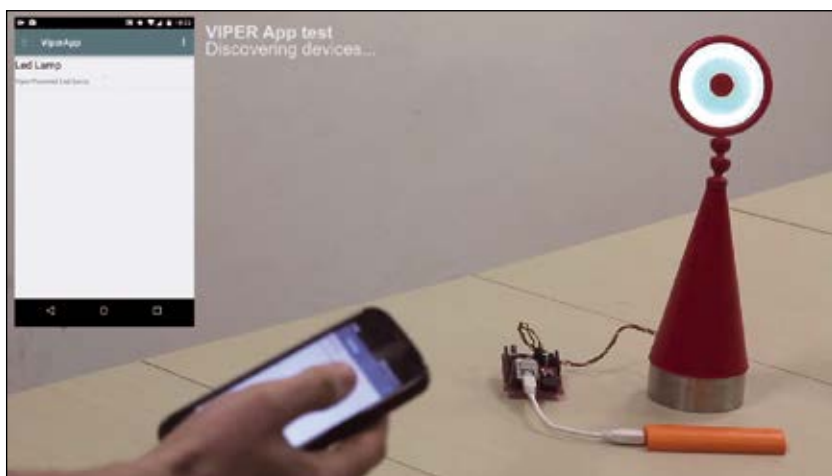
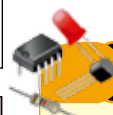


Fig. 7 - Esempio VIPER App.



per il MATERIALE

Il Toishield multiplatforma (cod. TOISHIELD), viene fornito assemblato, ed è disponibile presso Futura Elettronica al prezzo di 44,00 Euro. La board nucleo F401RE (cod. NUCLEOF401RE) viene venduta a 14,90 Euro. I prezzi si intendono IVA compresa.

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

LoRa™



Dotiamo
Arduino di un
modulo per
comunicazioni
long-range
basato sulla
tecnologia
Semtech.
Prima puntata.

LoRa SHIELD

dell'ing DANIELE DENARO

Abbiamo avuto modo di parlarvi (nel numero 191) delle due tecnologie che si contendono il mercato della comunicazione wireless a lungo raggio (alcuni chilometri) e basso consumo: la tecnologia SigFox e la LoRa; su quest'ultima si basa la demoboard che abbiamo realizzato (e proposto nel fascicolo numero 194), che fa uso di un modulo LoRa realizzato dall'Aurel. Ora è venuto il momento di trasportare la tecnologia long-range wireless sulla piattaforma Arduino e per questo vi proponiamo uno shield che utilizza un modulo RTX LoRa ed è corredato dalla relativa libreria software per un utilizzo immediato. Prossimamente lavoreremo anche con la tecnologia SigFox, per la quale Atmel (casa produttrice dei microcontrollori di Arduino) ha preparato uno shield per Arduino, che presenteremo appena disponibile.

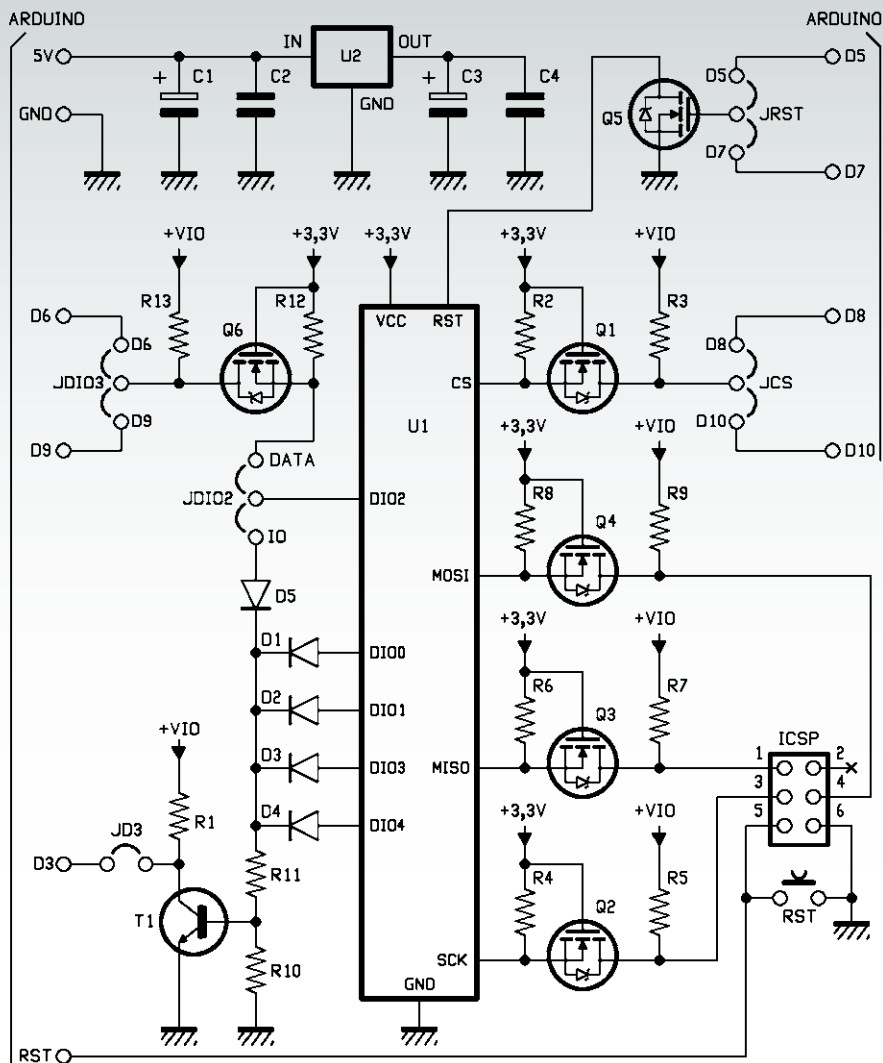
Prima di introdurre la scheda, è necessario riprendere brevemente i concetti base di queste due tecnologie. Innanzitutto chiariamo le motivazioni che hanno spinto alla ricerca di una comunicazione

a lungo raggio ma a basso consumo:

- realizzare soluzioni IoT (Internet of Things), ovvero collegare via radio sensori ed attuatori distribuiti in quantità, ma senza ricorrere a Wi-Fi o WiMax, non adatte alla tipologia minimale dei sensori/attuatori e dispendiose in termini di energia, nonché soggette a saturazione;
- avere bassi costi per sensore/attuatore wireless.

Le due tecnologie sono differenti in termini di tecnica di trasmissione e di filosofia: mentre SigFox utilizza la tecnica della banda strettissima (UNB=Ultra Narrow Band), LoRa utilizza il concetto opposto dello spettro di emissione distribuito (Spread Spectrum); inoltre SigFox contempla una struttura a stella dove una stazione ricevente, più sofisticata dei singoli terminali, concentra la ricezione e la instrada su normale rete Internet.

In questa architettura i terminali sono, in genere, solo trasmettenti. Al contrario, LoRa ha una architettura orientata al punto-punto, e, con adeguato



software potrebbe realizzare una rete distribuita come Internet. Anche in questo caso si può pensare ad un instradamento su Internet realizzato da uno o più nodi di questa struttura, ma ciò non è determinante come in SigFox; insomma, LoRa è una struttura più flessibile di SigFox. In ogni caso le implementazioni di ambedue i sistemi fanno ampio uso della tecnica SDR (Software Defined Radio), che consiste nella gestione software delle componenti dei ricetrasmittitori. I circuiti integrati sono perciò estremamente flessibili e configurabili (anche dinamicamente) sia in termini di frequenza fondamentale che di potenza

e di molti altri parametri; possono realizzare, quindi, sistemi adattativi. In particolare l'implementazione LoRa della Semtech è molto flessibile, come vedremo più avanti. Dal punto di vista della comunicazione, ambedue i sistemi irradiano su frequenze minori di 1 GHz: SigFox, tipicamente sugli 868 MHz (in Europa) mentre LoRa sugli 868 MHz o sui 433 MHz (bande libere senza licenza). La potenza di emissione è decisamente bassa: dell'ordine di $10\div 100$ mW (corrispondenti a $10\div 20$ dBm). Altra particolarità è che i dispositivi SigFox e LoRa, quando non trasmettono, praticamente non consumano elettricità: sono

quindi l'ideale per le applicazioni alimentate con batterie, elementi fotovoltaici o con soluzioni di "energy harvesting". La comunicazione può avvenire su distanze di chilometri o decine di chilometri. Ma come fanno, i dispositivi, a comunicare a queste distanze con una potenza così bassa? Ebbene, malgrado le diverse tipologie di modulazione adottate, LoRa e SigFox sono accomunati da un'altissima immunità ai disturbi ed alle interferenze, tali da ammettere sensibilità dell'ordine di -126 dBm ($10 \div 13$ mW). Ma, ovviamente nulla è gratis, e il prezzo da pagare è una bassissima velocità di comunicazione: alcune decine di bit al secondo (bps). Ma ciò non ha importanza nel tipo di utilizzo di questi sistemi, visto che è orientato a sensori/attuatori che devono scambiare brevissimi messaggi in maniera episodica (tipicamente qualche centinaio di volte al giorno).

MODULAZIONE

Lungi da pretendere una trattazione esaustiva di una problematica molto vasta e complessa come la trasmissione di segnali, facciamo il punto solo sui concetti base e sulla terminologia. I segnali digitali possono modulare la frequenza portante sostanzialmente in modalità:

- AM (modulazione di ampiezza) detta anche ASK (Amplitude Shift Keying) di cui la più semplice implementazione è la OOK (On/Off Keying) nella quale allo zero corrisponde assenza di portante e ad uno la presenza di portante;
- FM (modulazione di frequenza) detta anche FSK (Frequency Shift Keying); spesso usata nella forma GFSK (Gaussian FSK) dove viene applicato un filtro gaussiano al segnale

SDR: la radio definita da software

La tecnologia "Software Defined Radio" introduce la programmabilità in ciò che finora era un vero tempio dell'hardware: l'elettronica degli apparati di radiocomunicazione. Complice il passaggio, ormai pressoché totale, dei segnali modulanti dall'analogico al digitale, e l'avanzare delle tecnologie digitali nella sintesi di frequenza, nei filtri e nella conversione del segnale, è inevitabile immaginare

un apparato radio composto da moduli configurabili sia nella loro funzionalità che nel loro collegamento. Ecco allora i sistemi SDR, che possono dirsi sistemi radio generici, adattabili di volta in volta a frequenze, modulazioni, sensibilità e potenze varie mediante configurazione software.

La famiglia SX12xx della Semtech rappresenta un esempio di questo approccio.

che con accorgimenti di modulazione e filtri.

In SigFox la frequenza della portante in Europa è a 868 MHz, la modulazione è GFSK e il data-rate di 100 bps, con una larghezza dello slot di comunicazione di appena 100 Hz; quindi più terminali possono trasmettere contemporaneamente allocando lo slot disponibile. Poiché lo spettro è strettissimo, tutta l'energia è concentrata e di conseguenza ne basta poca perché il segnale sia d'ampiezza superiore al rumore.

per ammorbidire gli impulsi quadrati e ridurre così lo spettro; in questo modo la FSK modula quasi in modo analogico; la FM ha uno spettro maggiore rispetto alla AM ma è meno soggetta ai disturbi;

- PM (modulazione di fase) detta anche PSK (Phase Shift Keying); è la più efficiente ma anche la più complicata.

Esiste poi la modalità QAM che coniuga AM e PM (a diversi angoli di sfasamento) consentendo la trasmissione di più bit contemporaneamente e quindi aumentando il rateo dati.

Il segnale digitale modulante, poi, può avere diverse codifiche: NRZ (Non Return to Zero, il più semplice), RZ e ARZ (Always Return to Zero: in pratica il PWM), Manchester, codifiche bipolari, ecc. La scelta della codifica influisce sullo spettro e su altre caratteristiche come la presenza o meno di una componente continua o la possibilità di perdere il sincronismo a causa di lunghe sequenze di zeri (come per NRZ).

Ma, in ogni caso, qualunque modulazione disperde la frequenza portante in una banda centrata sulla frequenza base. Ne consegue una dispersione di energia e una maggiore sensibilità al rumore o alle interferenze.

ULTRA NARROW BAND

Una possibile soluzione è restringere moltissimo lo spettro di modulazione ad alcune decine di hertz, sia riducendo il data-rate,

SPREAD SPECTRUM

Un'alternativa è distribuire un segnale (ancora una volta con un basso rateo) su una banda molto

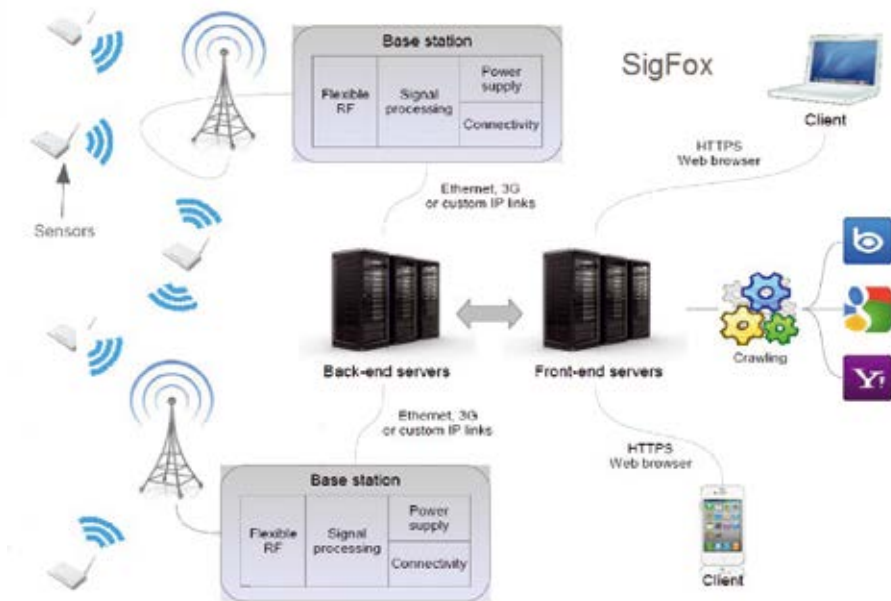


Fig. 1 - Schema del sistema SigFox.

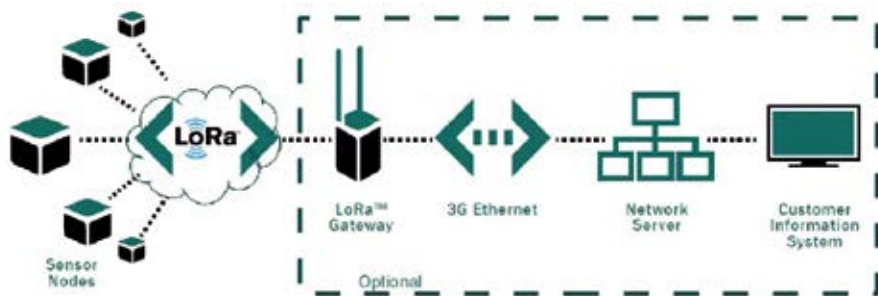


Fig. 2 - Schema del sistema LoRa.

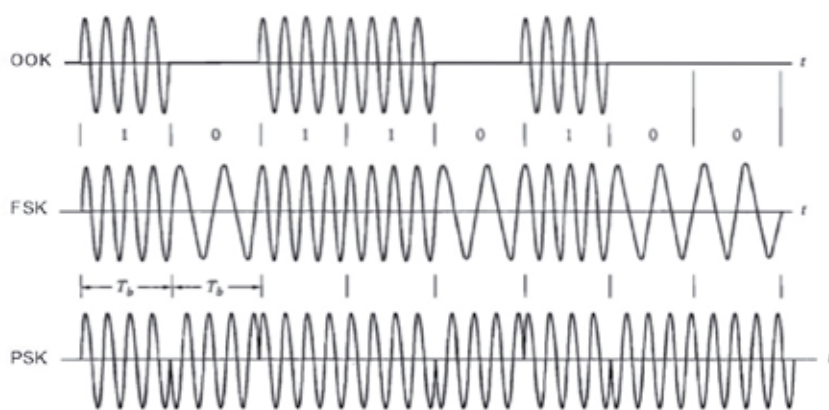


Fig. 3 - Le modulazioni di ampiezza, frequenza e fase.

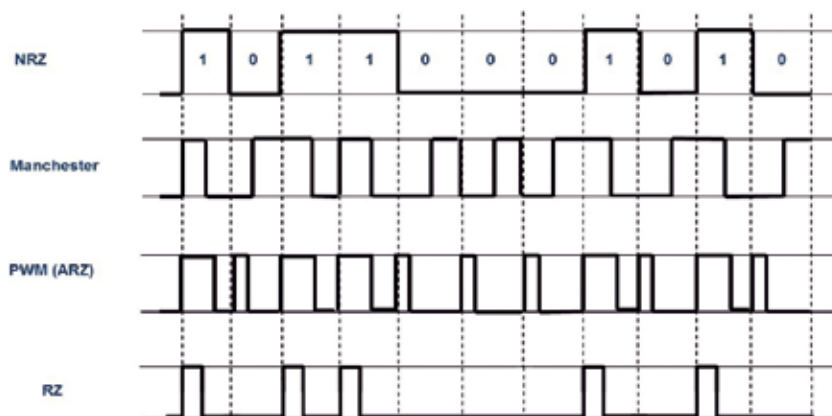


Fig. 4 - Codifiche del segnale modulante.

superiore a quella necessaria, utilizzando una qualche funzione di dispersione. La ricostruzione avviene tramite una funzione inversa.

Quella di dispersione è in genere una funzione pseudo-randomica, oppure una funzione che provoca un continuo salto di frequenza (Frequency Hopping) o, infine, una combinazione delle due tecniche. Il risultato è una dispersione del segnale a livello del rumore, il che rende difficile l'intercettazione e l'interferenza. Anche in questo caso l'energia è molto bassa, sebbene lo spettro sia molto ampio, perché non importa se il segnale si trova allo stesso livello del rumore, dato che il segnale può essere ricostruito dall'elaborazione. Questa tecnica è stata mutuata dal settore militare ed infatti

realizza una sorta di "camouflage" via etere. Nel LoRa, la banda di dispersione può essere fissata da 7,8 kHz a 500 kHz.

Inoltre ogni byte del segnale è modulato su più frammenti radio (Spreading Factor). Questo numero può essere definito da 64 a 4.096.

Per cui il rateo effettivo può andare da circa 18 bps (minima banda, massimo spreading factor) a 78 kbps (massima banda, minimo spreading factor).

Ma anche la sensibilità ed il valore SNR (segnale/rumore) ne sono influenzati; in particolare, un aumento dello spreading factor aumenta il valore SNR e della sensibilità, mentre un aumento della banda diminuisce la sensibilità. Nello specifico, si va da -5 dB a -20 dB per l'SNR e da -124 dBm a -134 dBm per la sensibilità.

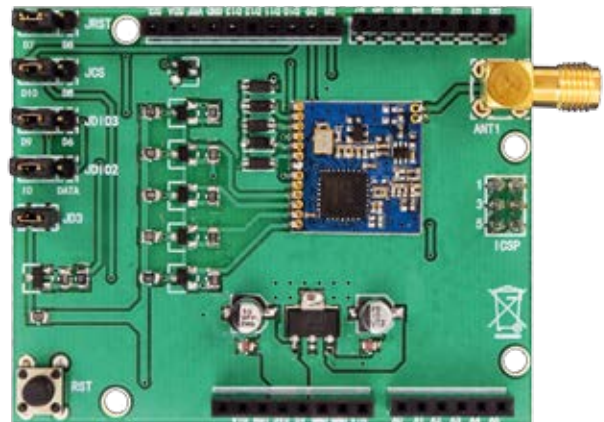
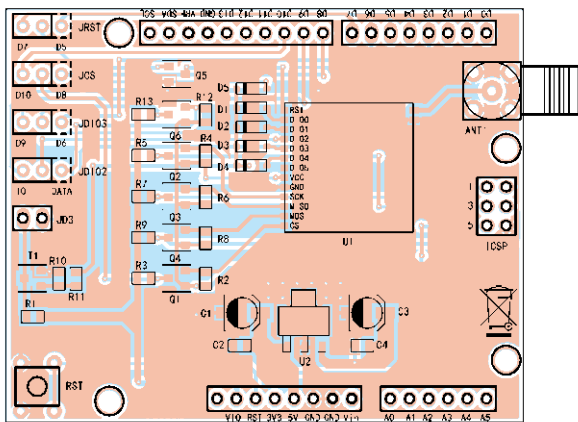
LO SHIELD PER ARDUINO

Lo shield che abbiamo preparato utilizza la tecnologia LoRa e si basa su un modulo DRF1278F della Dorji, il quale a sua volta monta il chip SX1278 della Semtech, che è il cuore del sistema. L'SX1278 è un componente molto sofisticato e versatile, che fa ampio uso dell'SDR, per cui è completamente configurabile.

La prima cosa da mettere in risalto è che lo SX1278 non è dedicato al solo sistema LoRa, ma può trasmettere e ricevere anche nelle classiche modalità FSK o OOK. In questi casi la comunicazione può avvenire con un protocollo a pacchetto con codifica NRZ, oppure perfino con una codifica qualunque, inserendo (o ricevendo) gli impulsi direttamente attraverso un pin di I/O. In quest'ultimo caso il chip funziona come semplice transceiver. L'SX1278 fa parte della famiglia SX12xx, e si distingue dagli altri membri per il range di frequenza della portante, che può andare da 123 a 525 MHz; gli altri possono arrivare anche a 1.020 MHz.

In questa prima parte descriveremo la struttura dello shield e l'utilizzo di esso nella modalità OOK o FSK, mentre nella seconda parte affronteremo la comunicazione in modalità LoRa.

La struttura della scheda è abbastanza semplice perché, oltre a contenere la schedina DRF1278F, ha un alimentatore riduttore da 5 a 3,3V, alcuni adattatori di livello (perché SX1278 lavora a 3,3V) e una combinazione in logica OR per una eventuale gestione degli interrupt che l'SX1278 può produrre sui suoi piedini di I/O. Inoltre, alcuni ponticelli permettono la scelta e l'abilitazione dei segnali con i pin di Arduino. Per esempio il chip select (negato) può essere collegato al pin D8 o D10 di Arduino, mentre il reset



Elenco Componenti:

R1 ÷ R10: 10 kohm (0805)
 R11: 4,7 kohm (0805)
 R12, R13: 10 kohm (0805)
 C1, C3: 47 μ F 6,3 VL elettrolitico ($\varnothing$ 4mm)
 C2, C4: 100 nF ceramico (0805)
 U1: DRF1278F
 U2: TC1262-3.3VDB

RST: Microswitch
 Q1 ÷ Q6: BSS138 (SOT23)
 T1: BC817 (SOT23)
 D1 ÷ D5: MMSD4148T1G

Varie:
 - Connettore SMA da CS
 - Strip maschio 2 vie (1 pz.)

- Strip maschio 3 vie (4 pz.)
 - Jumper (5 pz.)
 - Strip Maschio/Femmina 3 vie (2 pz.)
 - Strip Maschio/Femmina 6 vie
 - Strip Maschio/Femmina 8 vie (2 pz.)
 - Strip Maschio/Femmina 10 vie (1 pz.)
 - Antenna stilo cod. ANTSMAGSM
 - Circuito stampato S1190

di SX1278 può essere collegato al pin D5 o D7. L'OR degli interrupt può essere collegato al pin D3, che è uno dei due possibili interrupt esterni di Arduino. Ma il DIO2 di SX1278 può anche essere usato come input/output di impulsi (come vedremo più avanti), e può essere collegato a D6 o D9. Infine, il bus SPI per comunicare con lo SX1278 è collegato al connettore ICSP di Arduino.

È stato necessario inserire un alimentatore riduttore da 5 a 3,2 V perché l'assorbimento in fase di trasmissione (alla massima potenza) può arrivare a 120 mA e il convertitore a 3,3V di Arduino non è sufficiente a garantire l'assorbimento.

È opportuno tener presente che, mentre l'SX1278 ha due uscite per l'antenna, la scheda DRF1278F ne riporta all'esterno solo una: quella dedicata al range maggiore di potenza (uscita "boost"). Questa risulta collegata al connettore d'antenna in formato

SMA. Ne consegue che per avere segnale in antenna è necessario abilitare questa uscita (che per impostazione predefinita non è abilitata) tramite la configurazione di un registro dell'SX1278 (registro 0x09).

La configurazione dell'SX1278 comporta l'aggiornamento dei suoi registri interni tramite il bus SPI. I registri di configurazione sono ben 112, di cui la maggior parte gestisce più funzionalità mediante raggruppamenti di bit. Inoltre alcuni hanno un diverso

comportamento, o, più precisamente, hanno più copie, a seconda della modalità: LoRa o FSK/OOK. Nel caso di dati pacchettizzati, questi vengono inseriti in una coda FIFO (First In First Out) tramite il bus SPI e poi estratti e serializzati dal chip nel momento della trasmissione. Viceversa, nel caso della ricezione, vengono accumulati dal chip nella coda da cui possono essere estratti tramite il bus SPI.

La coda è diversamente configurata e gestita nel caso della

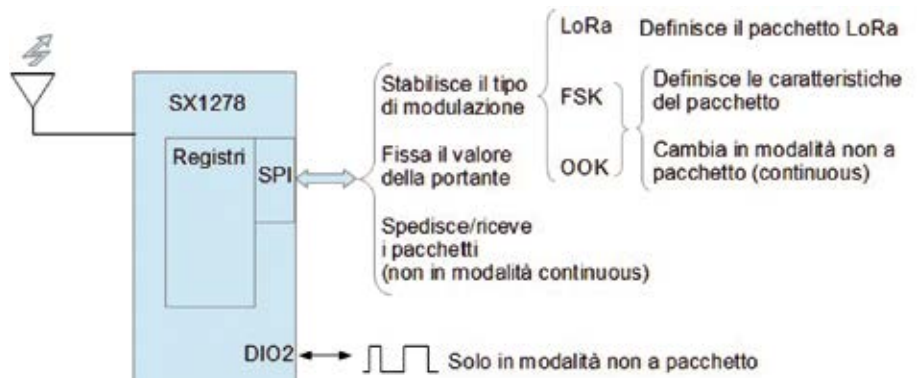


Fig. 5 - Modalità di funzionamento dell'SX1278.

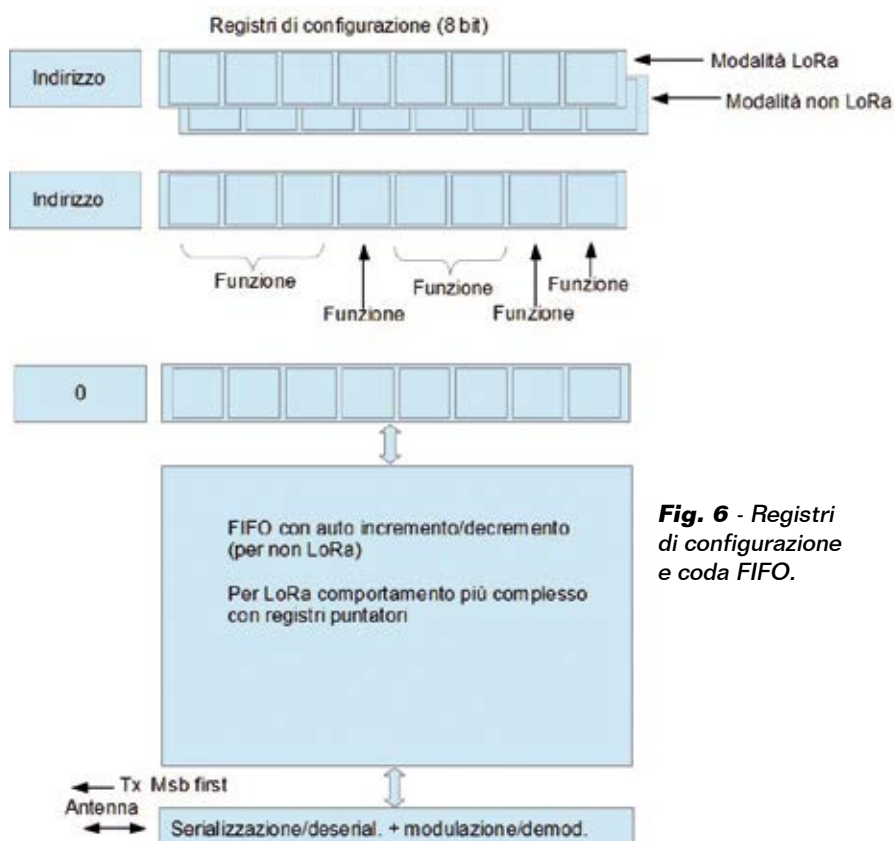


Fig. 6 - Registri di configurazione e coda FIFO.

modalità LoRa rispetto alla modalità non LoRa (FSK/OOK). La porta di comunicazione rimane, comunque, lo speciale registro di indirizzo 0.

Oltre alla porta per la coda FIFO (registro 0), uno degli altri registri comuni alle due modalità LoRa/NonLoRa è il registro 1, che definisce le seguenti modalità:

- bit 7; 0 = FSK/OOK 1=LoRa (predefinito 0=non LoRa);
- bit 6,5; 00=FSK 01=OOK (predefinito 00);
- bit 4,3 ; riservato o non applicabile al modello 78;
- bit 2,1,0; modalità operativa del chip:
 - 000 (0) Sleep mode;
 - 001 (1) Standby mode (predefinito);
 - 010 (2) FSTX mode (pronto a trasmettere);
 - 011 (3) TX (trasmette);
 - 100 (4) FSRX (pronto a ricevere);
 - 101 (5) RX (riceve pacchetto

in FSK/OOK mode o riceve in modo continuo in LoRa mode);

- 110 (6) RX SINGLE (solo in LoRa mode), riceve singolo pacchetto;

- 111 (7) CAD (solo in LoRa mode), rilevazione attività nel canale.

Altri importanti registri sono 0x06,0x07,0x08, che definiscono insieme la frequenza della portante in Hz. In Fig. 7 sono raggruppati i registri comuni alle due modalità.

Per un esame dettagliato si rimanda al Datasheet della serie SX12xx <http://www.semtech.com/images/datasheet/sx1276.pdf>.

I registri 0x40 e 0x41 sono destinati a configurare i pin di I/O che sono 6 (DIO0-DIO5); purtroppo queste configurazioni hanno un diverso significato a seconda della modalità. In ogni caso sono pin di output, eccetto DIO2, che può essere bidirezionale quando è configurato per la modalità *continuous*.

Gli altri pin sono soprattutto utili per veicolare all'esterno dei segnali di interrupt che comunque sono visualizzabili sui registri di flag 0x3E, 0x3F (per OOK/FSK mode) o 0x12 (per LoRa mode). L'uso del interrupt permette una programmazione ad eventi, ma

Address	Register Name		Reset (POR)	Default (FSK)	Description	
	FSK/OOK Mode	LoRa™ Mode			FSK Mode	LoRa™ Mode
0x00	RegFifo		0x00		FIFO read/write access	
0x01	RegOpMode		0x01		Operating mode & LoRa™ / FSK selection	
0x06	RegFrfMsb		0x0C		RF Carrier Frequency, Most Significant Bits	
0x07	RegFrfMid		0x00		RF Carrier Frequency, Intermediate Bits	
0x08	RegFrfLsb		0x00		RF Carrier Frequency, Least Significant Bits	
0x09	RegPaConfig		0x4F		PA selection and Output Power control	
0x0A	RegPaRamp		0x09		Control of PA ramp time, low phase noise PLL	
0x0B	RegOcp		0x2B		Over Current Protection control	
0x0C	RegLna		0x20		LNA settings	
0x40	RegDioMapping1		0x00		Mapping of pins DIO0 to DIO3	
0x41	RegDioMapping2		0x00		Mapping of pins DIO4 and DIO5, CkOut frequency	
0x42	RegVersion		0x12		Semtech ID relating the silicon revision	
0x4B	RegTcxo		0x09		TCXO or XTAL input setting	
0x4D	RegPaDac		0x84		Higher power settings of the PA	
0x5B	RegFormerTemp		-		Stored temperature during the former IQ Calibration	
0x01	RegAgcRef		0x13		Adjustment of the AGC thresholds	
0x02	RegAgcThresh1		0x0E			
0x03	RegAgcThresh2		0x5B			
0x04	RegAgcThresh3		0x0B			
0x70	RegPll		0x00		Control of the PLL bandwidth	
others	RegTest		-		Internal test registers. Do not overwrite	

Fig. 7 - Registri base e comuni.

con l'interrogazione dei registri di flag si può, comunque, tenere sotto controllo la comunicazione, anche perché lo SX1278 gestisce molti passi in automatico.

COMUNICAZIONE IN MODALITÀ OOK O FSK

Come si è detto, la comunicazione in modalità OOK o FSK può avvenire sotto forma di pacchetti o in forma libera con l'intervento di Arduino che avrà il compito di serializzare/deserializzare i dati attraverso il pin D6/D9 che può essere collegato al DIO2 dell'SX1278.

Ma è molto più comodo ed efficiente far gestire la comunicazione completamente all'SX1278 utilizzando la modalità a pacchetto e caricando/scaricando i dati dalla coda tramite SPI. I pacchetti possono essere di lunghezza fissa o variabile.

Il pacchetto a lunghezza fissa è composto da:

- un preambolo (da 0 a 65.536 byte) composto da una sequenza di byte uguali (la lunghezza è definita dai registri 0x25, 0x26); il tipo di byte può essere 0xAA (10101010) o 0x55 (01010101) (registro 0x27);
- campo di indirizzo (Sync), la cui lunghezza (da 0 a 8 byte) è definita nel registro 0x27 e i singoli byte sono definiti nei registri da 0x28 a 0x2F (ma non possono avere valore 0); il campo può definire un indirizzo (di rete o del terminale) perché quando è presente fa scartare automaticamente in ricezione il pacchetto se non corrisponde;
- Payload (lunghezza definita in 0x32); è l'unico che finisce nella coda FIFO e può avere lunghezza minima di 1 byte e massima di 2048 byte ed è composto da:
 - indirizzo utente (1 byte

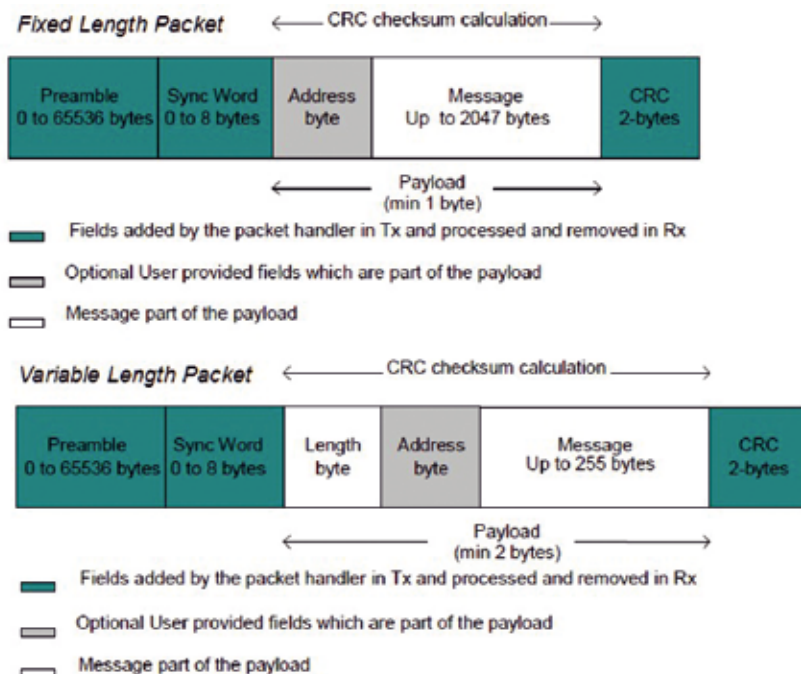


Fig. 8 - Composizione dei pacchetti.

- opzionale); è un ulteriore indirizzo.
- messaggio;
- CRC opzionale di 2 byte calcolato automaticamente dal chip.

Per quanto riguarda l'indirizzo utente, viene verificato se il registro 0x30 è configurato per "AddressFiltering"; in questo caso il pacchetto viene scartato se il byte di indirizzo non corrisponde a quello configurato nel registro 0x33. L'AddressFiltering può avvenire anche con un ulteriore controllo, prendendo in considerazione anche il registro 0x34 dove è stato inserito il "BroadcastAddress".

Il pacchetto di lunghezza variabile si differenzia per avere il primo byte del payload che indica la lunghezza e quindi consente un messaggio di 255 byte massimo. Nel caso di comunicazione in modalità FSK/OOK, un settaggio fondamentale è quello della coppia di registri 0x02, 0x03 i quali definiscono in maniera precisa il baud-rate della codifica NRZ. Infatti in modalità LoRa il

protocollo proprietario ha altri parametri.

Come vedete, lo SX1278 è estremamente configurabile in termini di tipo di modulazione, frequenza portante, baud-rate e insieme ad altri moltissimi parametri consente un preciso "tuning" della comunicazione, anche se molti parametri sono definiti con valori predefiniti mediante soddisfacenti. È quasi un modulo radio universale, nella logica della filosofia SDR.

Poiché la configurazione è notevolmente variabile, non possiamo esaurire l'argomento in questa sede; chi vorrà potrà consultare il data-sheet dell'SX1278.

UTILIZZIAMO LA SCHEDA PER COMANDARE LE PRESE RADIO-CONTROLLATE

Nel numero 196 di Elettronica In abbiamo descritto, in un apposito articolo, i protocolli di comunicazione delle prese di alimentazione a 220V radio-controllate (come i sistemi Avidsen o Velleman) e una realizzazione per controllare queste ultime tramite Arduino collegato a dei moduli radio a

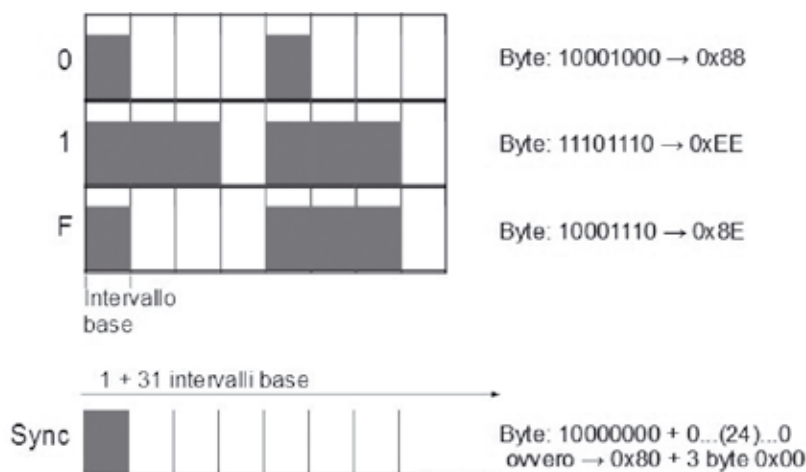


Fig. 9 - Codifica degli impulsi dei sistemi radio-controllati.



Fig. 10 - Pacchetto per sistemi radio-controllati.

basso costo. Nulla ci vieta di tentare la stessa strada con questa scheda, che contiene un così performante e flessibile sistema radio. La prima tentazione sarebbe quella di utilizzare la scheda

nella modalità "continuous" e quindi come un normale modulo radio utilizzando la libreria software, descritta nell'articolo, per far costruire ad Arduino la sequenza degli impulsi previsti

dal protocollo di comunicazione. Ma possiamo fare molto meglio: se vi ricordate le caratteristiche del protocollo utilizzato da questi sistemi, potete notare che, in effetti, la codifica dei valori three-state (0,1,Float) può essere immaginata come una sequenza di 8 bit, il cui baud-rate corrisponde alla frequenza dell'impulso base (vedi Fig. 9).

A questo punto teniamo presente che la codifica utilizzata nei pacchetti è di tipo NRZ e non essendo una trasmissione RS232, non ha i bit di start/stop, né la parità. Allora possiamo immaginare di trasformare il singolo frame del protocollo del radio-controllo in una sequenza di byte, ovvero un pacchetto. Basta porre a zero il preambolo e il campo Sync (indirizzo) e disabilitare il CRC. In realtà il pacchetto sarà composto da 4 ripetizioni di questo frame, come esige il proto-

Tabella 1 - Libreria per Arduino: Classe SX1278 (funzioni generali e per la modalità FSK/OOK).

Inizializza e reset	void begin(); void restart();
Attiva la modalità LoRa	void startModeLORA();
Attiva la modalità FSK/OOK	void startModeFSKOOK();
Decide se FSK (0) o OOK (1)	void setModulation(unsigned char mod); int readModulation();
Attiva il particolare modo operativo: SLEEP =0 STBY =1 FSTX =2 (pronto a trasmettere) TX =3 (trasmette) FSRX =4 (pronto a ricevere) RX =5 (se FSK/OOK) RXCONT (se LoRa) (riceve) RXSING =6 (solo LoRa) CAD =7 (solo LoRa)	void setStat(byte s); int readStat();
Frequenza portante (in Mhz)	void setFreq(float freq); float readFreq();
Bit Rate (se in modalità FSK/OOK)	void setBPS(int bps); unsigned int readBPS();
Potenza di trasmissione; p può essere: 1= 7dBm(5mW) ; 2= 10dBm(10mW); 3= 13dBm(20mW) 4= 17dBm(50mW)(def); 5=20dBm(100mW)	void setPower(byte p);
Abilita(1)/disabilita(0) la modalità a pacchetto (def). Se disabilitata va in modalità "continuous"	void setPackNoPack(byte pk);
Definisce la funzionalità di ognuno dei 6 pin di I/O (per il codice val vedere tabella corrispondente sul datasheet)	void setIO(byte nio,byte val);
Lunghezza e tipo del preambolo del pacchetto	void setPreamble(unsigned int len, byte pol);
Indirizzo (rete): se presente, lunghezza, array di valori	void setSync(byte on,byte len, byte val[]);
Carica la coda FIFO con i dati da spedire	void dataToSend(byte data[],int len);
Legge dalla la coda FIFO i dati ricevuti	void dataReceived(byte data[],int len);
Legge il primo o il secondo registro di flag	byte getFlags(byte nreg);
Legge il flag n dal primo o dal secondo registro dei flag	byte getFlag(byte nflag, byte nreg);

Tabella 2 - Libreria per Arduino: Classe REMOTEC.

Inizializza (tutto)	void begin();
Switch on/off prese Avidsen	void avidsenSet(byte address[5], byte socket, byte onoff);
Switch on/off prese Velleman	void vellemanSet(byte address[7], byte socket, byte onoff);

collo del radio-controllo, per un totale di 64 byte. Il baud rate sarà definito in base all'inverso della lunghezza dell'impulso base. Vediamo di riassumere i passi necessari per realizzare questo "remote-controller".

1. Abilitare l'uscita boost collegata all'antenna, che per impostazione predefinita non è abilitata (reg. 0x09); con lo stesso registro si può regolare la potenza di trasmissione (quella predefinita è pari a 17 dBm, ossia 50 mW) che si può ridurre a 2 dBm(1,5 mW) in step di 1 dBm, oppure si può abilitare la extra potenza massima di 20 dBm (100 mW).
2. Porre il chip in modalità operativa "Sleep" (reg. 0x01) e attivare la modalità non LoRa (reg. 0x01); infatti questa fondamentale commutazione può essere effettuata solo quando il chip è in condizione "Sleep".
3. Settare la modalità OOK (reg. 0x01).
4. Definire la frequenza portante a 433,92 MHz (reg.

0x06,0x07,0x08).

5. Definire il baud-rate (per esempio 3.800 per Advdsen) (reg. 0x02,0x03).
6. Definire il pacchetto di lunghezza fissa di 64 byte senza preambolo, senza indirizzo e senza CRC (reg. 0x25, 0x26, 0x27, 0x30, 0x32).
7. Caricare i dati nel FIFO.
8. Portare il chip allo stato FSTX (pronto a trasmettere, reg. 0x01) ed attendere alcune decine di microsecondi.
9. Portare il chip allo stato TX (trasmette, reg. 0x01).
10. Portare eventualmente il chip allo stato "Sleep" (non necessario perché l'SX1278 si porta automaticamente allo stato "Standby" appena finita la trasmissione del pacchetto).

La qualità e l'efficienza dello SX1278 è decisamente maggiore dei moduli radio a basso costo presi in considerazione nel precedente articolo, anche se la

scheda ha un costo abbastanza contenuto; ed inoltre scarica completamente Arduino dal compito gravoso di serializzare il treno di impulsi. Ne risulta un sistema affidabile e di buona portata radio. Tenete presente che quella proposta è solo una delle applicazioni possibili dell'SX1278.

LA LIBRERIA PER ARDUINO

Come avete potuto notare, la gestione del componente SX1278 non è affatto semplice a causa dei numerosi registri e delle moltissime funzionalità; abbiamo, perciò, scritto una libreria per semplificarne l'uso con Arduino. Sono state definite, innanzitutto, le funzioni base per leggere e scrivere nei registri appoggiandosi alla classica libreria SPI per Arduino. Successivamente sono state implementate le funzioni per i principali settaggi dell'SX1278. La libreria è comunque un cantiere aperto per l'implementazione delle funzionalità più

Listato 1

```

/*****
* Commandi (seriale a 9600):
*   ? : risponde con il nome dello sketch (utile per RandA)
*   onx : switch on; dove x=1,2,3,4,5 (1,2,3 per Velleman) (reply "ON")
*   ofx : switch off; dove x=1,3,3,4,5 (1,2,3 per Velleman) (reply "OF")
*   av : predisporre lo sketch per il sistema r Avidsen (default) (reply "OK")
*   ve : predisporre lo sketch per il sistema Velleman (reply "OK")
*   adxxxxxx : definisce l'indirizzo (three state es.: ad0200220) (solo 5 char per Avidsen)
*   pwx : regola la potenza di trasmissione;
*           dove x=1(5mW), 2(10mW), 3(20mW), 4(50mW) (default), 5(100mW)
*****/

#include "SX1278.h"
#include "REMOTEC.h"
#include <SPI.h>

#define ln 10

REMOTEC RC;

/* valori di default (inizialmente su Advisen con indirizzo 2020222) */
byte addAv[5]={2,0,2,0,0};int alenAv=5;
byte addVl[7]={2,0,2,0,2,2,2};int alenVl=7;
byte* add=addAv;
int alen=alenAv;

int mode=1; //modo Advdsen

```

(continua)

Listato 1 (segue)

```
char buff[32];

byte sock;
byte onoff;

void setup()
{
  Serial.begin(9600);
  RC.begin();
  setmodeavidsen();
}

void loop()
{
  int c=Serial.available();
  if(c>0)
  {
    Serial.readBytesUntil(ln,buff,32);
    if (buff[0]=='?') itsMe();
    if (strncasecmp(buff,"on",2)==0) commandSend(1);
    if (strncasecmp(buff,"of",2)==0) commandSend(0);
    if (strncasecmp(buff,"av",2)==0) setmodeavidsen();
    if (strncasecmp(buff,"v1",2)==0) setmodevelleman();
    if (strncasecmp(buff,"ad",2)==0) setAddress();
    if (strncasecmp(buff,"pw",2)==0) setpower();
  }
}

/* Funzione di riconoscimento; permette di sapere quale sketch è caricato su Arduino */
void itsMe()
{
  char name[]=__FILE__;
  char* c=strchr(name,'.');
  if (c != NULL) *c='\0';
  Serial.println(name);
}

void setmodeavidsen()
{
  mode=1;
  add=addAv;
  alen=alenAv;
  Serial.println("OK");
}

void setmodevelleman()
{
  mode=2;
  add=addV1;
  alen=alenV1;
  Serial.println("OK");
}

void commandSend(byte onoff)
{
  byte sock=atoi(&buff[2]);
  if (mode==1) RC.avidsenSet(add,sock,onoff);
  if (mode==2) RC.vellemanSet(add,sock,onoff);
  if (onoff==1)Serial.println(" ON"); else Serial.println(" OF");
}

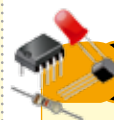
void setAddress()
{
  int i;
  if (mode==1){for(i=2;i<7;i++) addAv[i-2]=buff[i]-'0';}
  if (mode==2){for(i=2;i<9;i++) addV1[i-2]=buff[i]-'0';}
  Serial.println("OK");
}

void setpower()
{
  byte power=atoi(&buff[2]);
  if ((power<1)|(power>5)){Serial.println("NK");return;}
  SX.setPower(power);
  Serial.println("OK");
}
```

sofisticate. Nella **Tabella 1** potete vedere elencate le principali funzioni. Sulla base di questa libreria (classe SX1278) è stata creata una ulteriore libreria (classe REMO-TEC) finalizzata al controllo remoto delle prese 220V.

Nella **Tabella 2** sono descritte le funzioni di questa libreria.

Ne consegue che lo sketch del **Listato 1** è molto semplice. Lo sketch legge dalla seriale il comando e lo esegue rispondendo sulla seriale con una stringa "ON", "OF" o "OK". Lo sketch è stato pensato (e verificato) anche per un utilizzo con RandA. Potete immaginare la quantità di utilizzi che si possono facilmente implementare: dall'accensione/spegnimento programmato agli allarmi o alla gestione intelligente di carichi. Bene, con questa puntata abbiamo concluso; nella prossima analizzeremo le modalità di ricezione FSK/OOK e soprattutto affronteremo la modalità LoRa che permette a due schede di comunicare su lunghe distanze. ■



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>



Riproduciamo in chiave moderna l'apparato che nel film **RITORNO AL FUTURO** permetteva di viaggiare nel tempo.

di **BORIS LANDONI**

Chi ha qualche "decade" sulle spalle (ma anche i più giovani, che lo hanno conosciuto e visto grazie allo streaming su YouTube) conosce *Ritorno al futuro*, il famoso film che è poi diventato una saga... con tre episodi. Il grande fascino del film - diretto da Robert Zemeckis e prodotto nientemeno che da Steven Spielberg - stava nella bizzarra e avveniristica avventura di un giovane, Marty McFly (interpretato da Michael J. Fox), che con l'aiuto di uno scienziato dall'aria strampalata - tale "Doc" Emmett L. Brown (al secolo Christopher Lloyd) - riusciva a proiettarsi nel futuro ed anche nel passato usando come macchina del tempo un accrocchio

realizzato da Doc e viaggiando a bordo di un'automobile divenuta un cult: la DeLorean DMC-12, scelta dal professore, convinto che: "Dovendo trasformare un'automobile in una macchina del tempo perché non usare una bella automobile?". Quest'auto, novella capsula del tempo, per poter effettuare il salto doveva ricevere una potenza elettrica di 1,21 Gigawatt: un'enormità che veniva poi utilizzata dal "flusso canalizzatore" (nel film originale *flux capacitor*), situato dietro i sedili della DeLorean. Originariamente, il flusso canalizzatore veniva alimentato da del plutonio, mentre nei film successivi occorreva semplicemente intro-

FLUSSO» CANALIZZATORE

dure della comune spazzatura nel dispositivo di conversione *Mr. Fusion* per attivare la fusione nucleare e sviluppare la potenza necessaria. Nel film, dopo aver attivato i circuiti temporali (anche detti tempo-circuiti) ed aver impostato data e ora destinazione, McFly doveva accendere il motore e accelerare fino a 88 miglia orarie (141,6 km/h), finché il flusso canalizzatore non si attivava. La DeLorean, una volta effettuato il salto nel tempo, si trovava nella stessa posizione fisica sulla Terra al momento della partenza. I cultori di Ritorno al futuro ricorderanno che il pannello visualizzatore dello stato del flusso ca-

nalizzatore era composto da una stella a tre punte formata da lampade, cui erano applicati tre cavi con le classiche pipe isolanti in gomma. Ricorderanno pure che nel primo episodio, la data scelta per il viaggio sperimentale nel futuro -dal destino incerto per i due protagonisti- era il 21 ottobre 2015; sì, proprio un giorno di questo mese. Ritorno al futuro ha stimolato la curiosità e la creatività di molti cultori, tanto che per questa data sono stati organizzati addirittura degli eventi commemorativi, alcuni dei quali favoriscono suggestivi confronti con quello che il film proponeva come scenario all'arrivo nel futuro di Marty McFly



Fig. 1 – L'originale pannello del Flusso Canalizzatore di Ritorno al futuro.

e Doc: per esempio la videocomunicazione (che se vogliamo, oggi si traduce in Skype) e le auto volanti -tra l'altro ipotizzate anche in film come *Minority Report*- che purtroppo per gli sceneggiatori e per fortuna per noi, ancora non esistono...

Ma del resto negli anni '80 del secolo scorso, come anche prima, il progresso aveva spinto i più ottimisti registi e sceneggiatori a sognare un'evoluzione ben più rapida di quella che le leggi della Fisica imponevano ed hanno imposto; d'altra parte Ritorno al futuro non fu la sola evasione futuristica, perché la conquista della Luna e dello Spazio in generale spinsero il cinema e la televisio-

ne a proporre serie come *Spazio 1999* (dove si teorizzava una Luna, ormai deposito di scorie radioattive prodotte dalle centrali nucleari sulla Terra e popolata da 300 persone abitanti la futuristica Base lunare Alpha, strappata all'orbita terrestre e proiettata in un viaggio interminabile nello Spazio) e *Star Trek* o *Guerre Stellari*.

Il fatto che in *Ritorno al futuro* il primo viaggio nel tempo avesse come destinazione il 21 ottobre 2015, ci ha fatto scivolare nell'entusiasmo che sta travolgendo i cultori della trilogia e che ha stimolato la parte Maker che ognuno di noi porta dentro di sé. Dalla passione all'idea il passo è stato breve, come rapida è stata la traduzione nella pratica: un po' di ore in laboratorio ed è nato il progetto proposto in queste pagine, che è una rivisitazione della macchina per il flusso canalizzatore in chiave moderna, con la quale anche noi vogliamo omaggiare il grande sogno che Robert Zemeckis ha condiviso con i fan di *Ritorno al futuro* e la passione suscitata da una versione di macchina del tempo più realistica di quelle proposte in altri film, decisamente improbabili, sicuramente perché in quegli

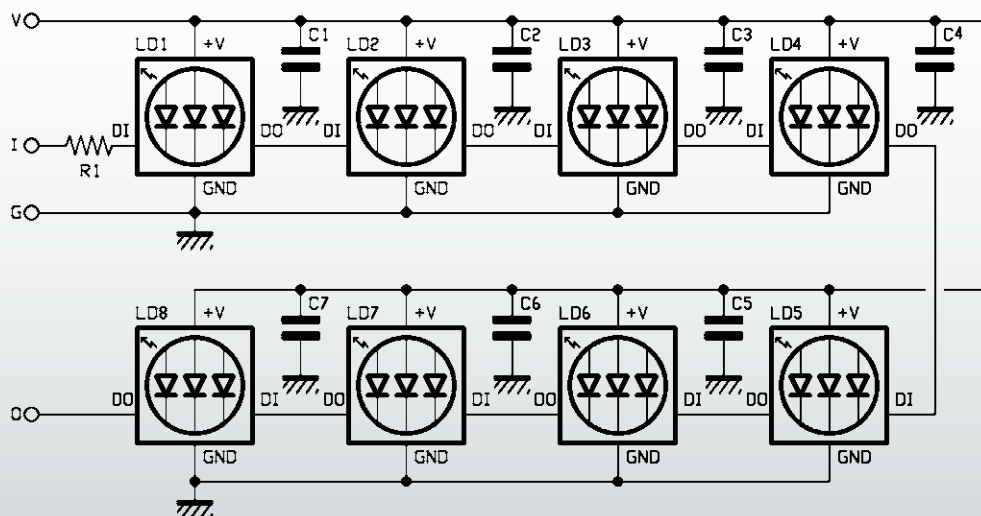
anni l'elettronica ci avrebbe fatto credere quel che un po' poi è stato, cioè che fosse la tecnologia per realizzare ciò che un tempo era impossibile.

E in un certo senso il fascino dell'elettronica è un po' questo, perché diversamente da altre discipline, essendo qualcosa che non si vede e che non è immediato spiegare, noi umani siamo portati ad affidarle i nostri sogni e le nostre speranze (diversamente dalla meccanica, per esempio, la quale è visibile e più immediata e perciò si presta meno a immaginare con essa creazioni fantascientifiche).

IL NOSTRO PROGETTO

Per simulare il macchinario del flusso canalizzatore che proiettava nel tempo la DeLorean, abbiamo fatto uso dell'ormai immancabile Arduino Uno, che con caricato un apposito sketch, e questa volta senza bisogno di alcuno shield, pilota tre strip, ciascuna con 8 LED Neopixel. Le strip vengono gestite in parallelo con una sola linea di Arduino, che nel nostro caso è facilmente modificabile a piacimento specificandola nello sketch; la comunicazione è unidirezionale e gestisce un gruppo di LED, che

[schema ELETTRICO]



Dalla celluloide alla realtà



Come spesso accade quando, in un racconto letterario o in un film, si azzarda una previsione sul mondo di domani, anche nel celebre Ritorno al futuro il regista e lo sceneggiatore hanno previsto scene che ci proponevano una loro visione del tempo a venire e delle cose innovative che sarebbero arrivate con esso. Ma un po' come in Spazio 1999 (serial inglese) si teorizzava già nel 1996 una base lunare abitata da umani e ad oggi abbiamo appena la ISS che vaga nello Spazio intorno alla Terra, anche in Ritorno al futuro abbiamo visto cose che ci hanno strabiliato, e che in realtà "hanno da venire". È suggestivo provare a fare un elenco di quel che si è avverato e di quanto no. Sicuramente ci sono oggi cose che il film aveva previsto e che oggi troviamo realmente. La prima è la videocomunicazione, permessa dai servizi di chat video come Skype; la seconda sono i sistemi di sicurezza con accesso a parametri biometrici, vale a dire tecnologie di identificazione basate sul riconoscimento delle impronte digitali, della conformazione del viso e dell'iride ecc. La terza sono gli schermi piatti (LCD, OLED ecc.) e la multivisione o, se vogliamo, la tecnologia PIP (Picture In Picture) e quella di visione multipla usata nella videosorveglianza.

Quarta invenzione sono gli schermi flessibili, ad esempio come quelli panoramici dei più moderni TV a schermo curvo e quelli OLED di smartphone come Samsung Galaxy.

E come non parlare dei PC Tablet, quinta previsione di Ritorno al futuro, cui si sono aggiunti i video occhiali, visione quasi onirica dei Google Glass.

Gli ologrammi in 3D, peraltro proposti in altri film (per esempio Total Recall, con Arnold Schwarzenegger...) sono oggi possibili grazie al laser olografico.

Ottava innovazione sono i videogiochi che come dispositivo di input utilizzano, invece dei joystick, sistemi di riconoscimento dei gesti: parliamo ad esempio di Microsoft Kinect o della Wii e di altri sistemi con sensori indossabili.

Nona previsione avverata è lo Slamball, uno sport di squadra ispirato al basket, dal quale si distingue da esso perché sul campo vi sono quattro tappeti elastici posti sotto ogni canestro, che permettono ai giocatori di amplificare il salto ed effettuare schiacciate. Insomma, un basket acrobatico...

Al decimo posto troviamo i vestiti tecnologici, precursore dei quali è quello indossato da McFly: speciali fibre, sensori e attuatori consentono di adattarsi al corpo e di comunicare le condizioni della persona che li indossa, nell'ottica dell'elettronica indossabile e dell'IoT. Cestini della spazzatura robotizzati che inseguono Doc in una delle scene



più divertenti del film, possono essere paragonati ai robot spazzini da tempo introdotti da vari centri di ricerca (per esempio dalla scuola Sant'Anna di Pisa).

Dodicesima previsione avverata sono i droni che filmano: in Ritorno al futuro ci sono piccoli aerei che inseguono la notizia, la filmano e la diffondono e nel film riprendono il processo in tribunale; oggi i droni si usano moltissimo e i multicotteri amatoriali dotati di telecamera per ripresa aerea impazzano.

A Marty McFly venne annunciata nel film la scomparsa dei laser disc, i predecessori del dvd, più grandi però, utilizzati nei video juke box e che rimasero in commercio fino al 1998.

Marty e Jennifer McFly nel 2015 avevano una casa dove tutto è connesso e comandato; oggi è realtà con il sempre crescente ricorso alla domotica e alle tecnologie smart e all'IoT. Analogo è il discorso per il sistema Homechat presentato all'ultimo Ces di Las Vegas dalla LG, che permette di scambiarsi messaggi con gli elettrodomestici, come se fossero una persona.

E da ultimo, hoverboard, lo skateboard volante tipo overcraft di Ritorno al futuro, potrebbe entrare in commercio davvero entro la fine di quest'anno. Così ha promesso la casa Haltek Industries.

nel nostro caso in totale sono 24. Il collegamento dell'insieme è illustrato in queste pagine nello schema di cablaggio. Prima di proseguire è opportuno spendere qualche parola sulla tecnologia Neopixel, perché permette di realizzare LED RGB "intelligenti" con controller a bordo, facilmente integrabili nell'ambiente Arduino grazie a librerie proprietarie che l'Adafruit (www.adafruit.com) rende disponi-

bili liberamente. 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 riguar-

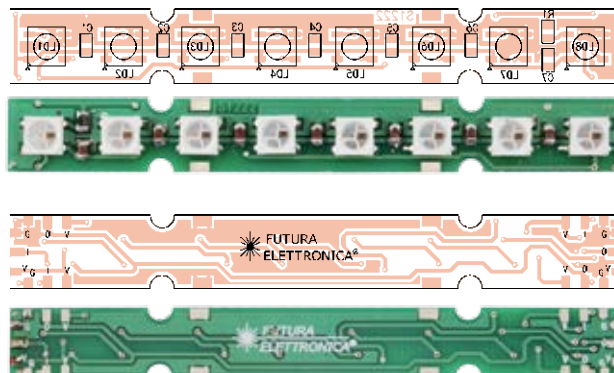
da il nostro progetto. Ogni LED RGB può essere gestito individualmente tramite un apposito comando inserito nella stringa seriale e può produrre fino a 256 tonalità del proprio colore, determinando un totale di 16.777.216 combinazioni di colore. Neopixel è praticamente una soluzione che prevede l'integrazione di un driver e del rispettivo LED RGB in un case SMD, consentendo il comando diretto LED per LED.

Elenco Componenti:

R1: 470 ohm (0805)
C1 ÷ C7: 100 nF ceramico (0805)
LD1 ÷ LD8: LED NeoPixel

Varie:

- Circuito stampato S1222



Il canale dati usato per la comunicazione con i LED neopixel e quindi con le strip è simile al tipo oneWire. L'alimentazione prevista per i LED Neopixel è a 5 volt; la comunicazione avviene a un massimo di 800 kbps.

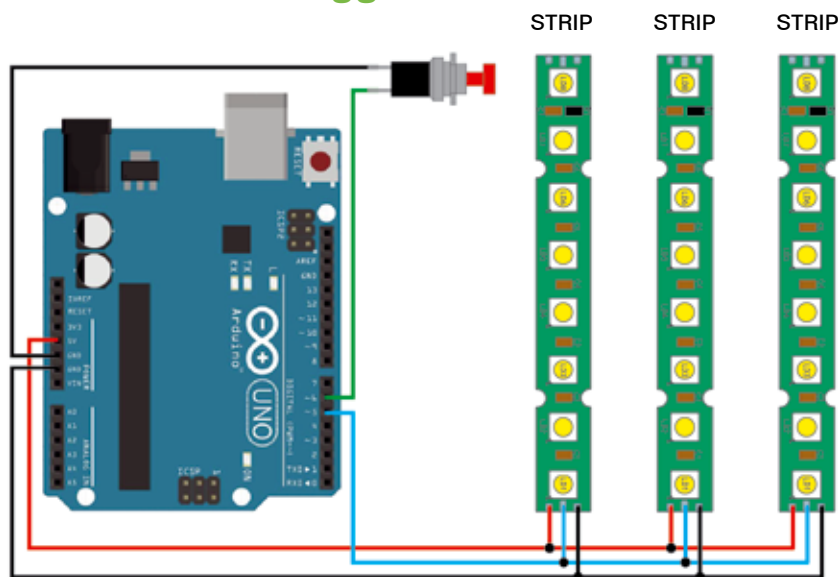
Per ogni strip di LED è possibile impostare a piacimento la frequenza di refresh, in modo da rendere impercettibili determinati giochi di luce. Nel nostro caso la frequenza di scansione dei LED è, per ciascuna strip, di 400 Hz.

Più strip possono essere collegati in cascata o in parallelo per realizzare vari effetti, ma in questo caso tale configurazione non interessa; sappiate comunque che più strip

si connettono a un unico canale dati, più, fermo restando il massimo data-rate consentito, verrà limitata la frequenza di refresh. In breve, la frequenza di refresh e quindi la velocità di accensione/spegnimento dei singoli LED, è inversamente proporzionale alla quantità di LED da gestire. 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 verde). Analizziamo dunque lo schema elettrico della strip, dove appare l'estrema

semplicità della realizzazione: ogni LED intelligente è collegato in cascata, dato che la linea dati entrante nel terminale DI esce dal DO, che ne ripete i dati. L'alimentazione è a 5 volt (V della strip), prelevabili dal contatto 5V di Arduino, dato che l'assorbimento di ciascuna strip non raggiunge i 200 milliampere 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 di Arduino e va a GND di quest'ultima scheda. I numerosi condensatori posti sull'alimentazione servono a filtrare gli impulsi creatisi sulle piste per effetto dell'assorbimento dei LED quando si accendono; 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 di Arduino. Torniamo adesso ad Arduino e vediamo che oltre alle tre strip in parallelo ed alla board abbiamo collegato un pulsante che ci serve a scegliere tra i giochi di luce previsti dallo sketch; il pulsante è normalmente aperto ed è connesso sul pin 6 di Arduino e a massa (al corrispondente pin dell'Atmega è abilitata da software la resistenza di pull-up, così da risparmiarci un

Schema di cablaggio



resistore esterno e semplificare il cablaggio). Il tutto si alimenta da USB, quindi da un PC, ma è anche possibile alimentare arduino tramite l'apposito plug; in questo caso consigliamo di utilizzare un alimentatore con una tensione di uscita non superiore ai 7,5V, in modo da non "stressare" eccessivamente il regolatore interno di Arduino. Una volta fornita l'alimentazione, Arduino carica lo sketch e controlla periodicamente lo stato del pulsante; contemporaneamente avvia il gioco di luci predefinito, che prevede i LED accesi di bianco che si spostano da periferia a centro tutti sincronizzati a convergere. Una pressione del pulsante fa accendere alla stessa maniera i LED di rosso, un'ulteriore pressione fa altrettanto con i LED a luce verde e un nuovo intervento sul tasto ripete il gioco con i LED a luce blu. Ulteriori pressioni producono altri giochi luminosi per un massimo di 10 in totale. Tra questi abbiamo realizzato un gioco che riproduce fedelmente l'effetto del flusso canalizzatore del film. Giunti al decimo, si ricomincia con quello predefinito all'accensione.



Fig. 2 – Il nostro prototipo.

REALIZZAZIONE PRATICA

Siccome volevamo realizzare qualcosa che replicasse il più fedelmente possibile il quadro del Flusso Canalizzatore del film, il nostro progetto doveva avere, oltre a un'elettronica, una parte meccanica che fosse la più somigliante possibile all'originale; siccome il quadro elettrico del film era il classico di metallo con finestratura in vetro e guarnizione in gomma, abbiamo quindi realizzato un "falso" composto da una scatola grigia in cartone verniciato di grigio, in luogo del quadro elettrico, applicando poi una finestra in foglio di acetato spesso con finta guarnizione di gomma stampata in 3D con PLA nero mediante la nostra stampante 3Drag.

La scatola potete comunque realizzarla in plastica, sempre verniciandola di grigio.

Le strip sono state disposte a stella a tre punte ed applicate a un doppio fondo in cartone ondulato verniciato di nero, mentre Arduino è stato montato dietro (fra il doppio fondo e il fondo) fermandolo con distanziali incollati al fondo della scatola mediante colla a caldo. Per rendere più uniforme la luce emessa e più somiglianti le strip alle lampade tubolari a scarica usate in Ritorno al futuro, abbiamo inserito ogni strip in una guaina in plastica trasparente ottenuta da uno spezzone di tubo trasparente per annaffiare, del diametro interno di poco maggiore alla larghezza della strip. Si può usare sia del tubo liscio e trasparente, sia del tubo traslucido o con la superficie lavorata.

I fili di collegamento delle strip escono da un foro al centro della stella, almeno quelli veri, ossia i tre che portano ad Arduino; possono essere ottenuti con pezzi di piattina a tre fili, collegati poi in parallelo a tre punte (o jumper per

Guarda il Flusso canalizzatore su YouTube!



Vuoi vedere in azione il pannello per il Flusso canalizzatore? Nulla di più semplice:

collegati alla pagina <https://youtu.be/-FruWlrKeWl>. Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.



Arduino) inserite nei connettori di espansione di Arduino, nella posizione indicata dallo schema di cablaggio.

I fili applicati dall'altra estremità di ciascuna strip sono solo scenografici (sono del tutto fittizi) e simulano i cavi delle estremità delle lampade a scarica del quadro visibile nel film; si fissano ognuno con degli isolatori a pipa (rossi) da candela di accensione dei motori



per il MATERIALE

Tutto il materiale necessario per realizzare questo progetto è disponibile presso Futura Elettronica, in particolare la board Open Source Arduino UNO (cod. ARDUINOUNOREV3) è in vendita al prezzo di 24,50 Euro, mentre gli stick con i led indirizzabili (cod. STICK8NEOPIXEL) vengono forniti a 6,50 Euro l'uno. 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>

a benzina applicati su viti metalliche strette su isolatori tondi da 40 mm di diametro stampati in 3D e incollati al cartone nero. I fili finti che nella macchina originale del film porterebbero l'alta tensione, possono essere ricavati verniciando di giallo dei tubi di gomma trasparente del diametro di 6÷8 mm, che avrete cura di infilare nel cartone. Il risultato è quello visibile in Fig. 2, che potete confrontare con l'originale in Fig. 1 per constatarne la somiglianza.

LO SKETCH

Per ottenere i giochi di luce va caricato in Arduino, mediante l'apposito IDE e la connessione USB al computer, lo sketch che rendiamo disponibile sul nostro sito Internet www.elettronica.in.it

insieme agli altri file del progetto. Lo sketch fa uso della libreria di Adafruit NeoPixel.h, che viene inclusa all'inizio, prima della definizione dei pin e delle variabili, con la riga:

```
#include <Adafruit_NeoPixel.h>
```

Subito dopo viene assegnato il pin per la connessione del pulsante di selezione dei giochi di luce (in questo caso il 6), il che si ottiene mediante l'istruzione:

```
#define BUTTON_PIN 6
```

La comunicazione con le strip a LED è assegnata al pin Digital I/O 5 (D5) mediante l'istruzione:

```
#define PIXEL_PIN 5
```

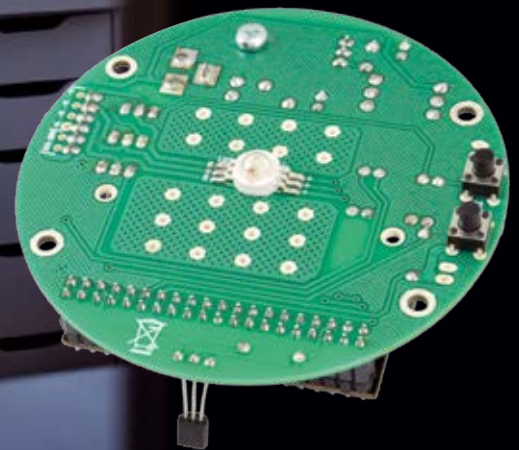
Dunque, se desiderate cambiare linea di Arduino, per esempio perché intendete utilizzare la Digital I/O 5 per altri scopi, modificate (editando lo sketch con l'IDE di Arduino) questa linea scrivendo al posto di 5 il numero del pin digitale desiderato, quindi salvate lo sketch e ricaricarlo in Arduino. Infine, nello sketch viene definito il numero di LED da pilotare per ogni strip, che è 8:

```
#define PIXEL_COUNT 8
```

A questo punto il firmware può avviarsi e gestire sia la lettura in loop del pulsante, sia la visualizzazione sulle strip di LED, ottenuta attraverso una struttura switch/case e l'utilizzo della libreria Adafruit. ■

The advertisement features a large green banner with the text "ElettroExpo" in white, stylized font. Below it, in white text on a green background, is "VERONA 28/29 NOVEMBRE 2015". Underneath, in green text on a white background, is "53^ FIERA DELL'ELETTRONICA DELL'INFORMATICA E DEL RADIOAMATORE". At the bottom left, the website "www.elettroexpo.it" is listed, along with logos for "organized by VERONAFIERE" and "GFS". On the right, a black circular badge with white text reads "MERCATINO DEL PRIVATO E DELLA FOTOGRAFIA". The background is decorated with green circuit board traces and components.

Realizziamo
con Arietta G25
una lampada
multifunzione a LED
sempre connessa in
wireless.



di LUCA BELLAN

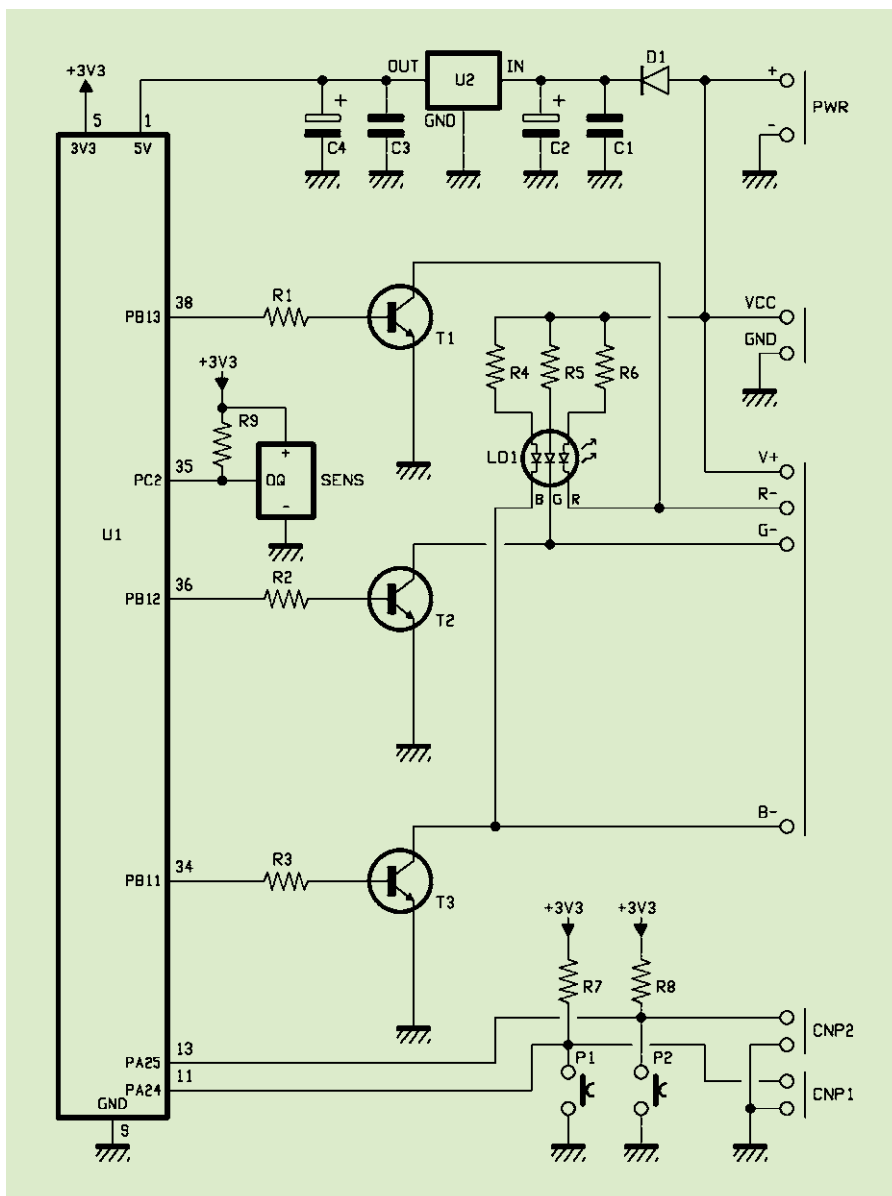
Wi-LAMP LA LAMPADA WiFi

Già da diversi anni esistono sistemi per automatizzare la casa: irrigazione, serramenti, illuminazione, climatizzazione e sistemi d'allarme, tutto può essere centralizzato e programmato. Negli ultimi tempi le classiche centraline di controllo hanno lasciato il posto a sistemi intelligenti sempre connessi alla rete WiFi e accessibili da remoto tramite dispositivi mobili come smartphone o tablet.

Il progetto qui proposto segue questa tendenza ed è dedicato all'illuminazione, perché si tratta di una lampada RGB (capace quindi di assumere tutti i colori) governabile da dispositivi connessi in WiFi. Il cuore della lampada, che abbiamo battezzato Wi-Lamp, è una

scheda ad interfaccia wireless, che monta un LED RGB di potenza e dispone anche di una morsettiera che permette, in alternativa, di comandare strisce RGB, permettendoci così di illuminare interi locali. Per l'uso come lampada singola, il circuito della Wi-Lamp può essere inserito in una lampada commerciale, come un'abat-jour, oppure in un involucro della foggia desiderata, stampato con una stampante 3D; a tal riguardo ricordiamo che cercando "lamp" sul sito www.thingiverse.com si trovano molti esempi.

Dopo averla montata e configurata, Wi-Lamp sarà sempre connessa alla nostra rete WiFi e potrà essere controllata tramite tablet o smartphone, o in generale



con qualsiasi dispositivo dotato di browser e connesso alla sua stessa rete: basterà semplicemente digitare l'indirizzo <http://wi-lamp> (oppure l'IP fornito in fase di installazione) per accedere al pannello di controllo web della lampada e selezionare una tra le varie modalità di funzionamento disponibili, che sono le seguenti:

- **OFF** = la lampada viene spenta, anche se la scheda Arietta rimane accesa in attesa di comandi;
- **Self** = facendo scorrere il dito sul color picker possiamo selezionare il colore e l'intensità voluti;

- **Fade** = la lampada sfumerà da un colore all'altro in un ciclo infinito; agendo sulla barra orizzontale (slider) possiamo controllarne la velocità;
- **Party** = la lampada lampeggerà con colori random a una velocità impostabile tramite la barra orizzontale;
- **Sunset** = modalità utile nel caso in cui volessimo andare a letto senza doverci ricordare di spegnere la luce; possiamo selezionare la durata del "tramonto" (da 1 a 20 minuti) tramite lo slider e farlo partire

premendo il pulsante "Sunset Start" (la lampada si accenderà immediatamente con il colore bianco e sfumerà in tutti i colori del tramonto fino a spegnersi completamente);

- **Sunrise** = è una modalità simile a Sunset, ma con colori più adatti al mattino; agendo sullo slider possiamo scegliere la durata dell'alba (da 1 a 20 minuti), mentre il campo "Starting Time" serve per scegliere l'ora di partenza; raggiunta l'ora impostata la lampada (da spenta) inizierà ad accendersi gradatamente mostrando tutti i colori dell'alba fino a raggiungere la completa accensione (colore bianco);
- **Temperature** = grazie al sensore DS18B20 a bordo scheda possiamo ricavare la temperatura nella stanza e colorare la lampada su una sfumatura che va dal blu (temperatura minima impostata) al rosso (temperatura massima impostata); impostando, ad esempio, la temperatura minima a 10°C e la temperatura massima a 30°C, se la temperatura rilevata fosse intorno ai 10°C (o minore), la lampada si colorerebbe completamente di blu;
- **Fire** = la lampada simula la fiamma di un camino;
- **Settings** = in questa sezione è possibile cambiare le modalità che vengono richiamate alla pressione dei due pulsanti presenti sul PCB della lampada; possiamo inoltre spegnere completamente la lampada (inclusa Arietta) premendo il pulsante "Shutdown Lamp";
- **Auto** = tramite questa modalità possiamo programmare la lampada perché si accenda in un determinato giorno della settimana (o tutti), ad un determinato orario, per una durata e un colore voluti; dopo aver impostato tutti i parametri occorre premere il pulsante "Create" per salva-

Fig. 1

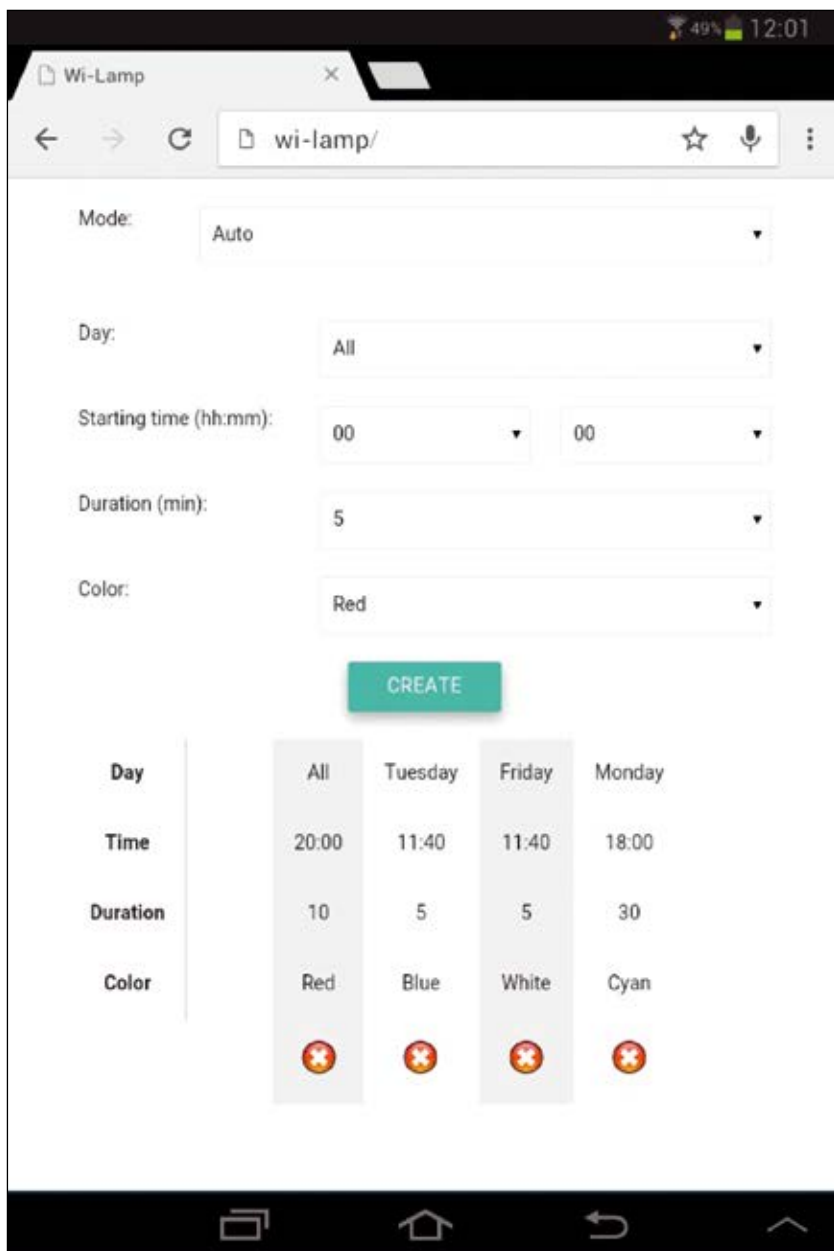
re la programmazione e, per cancellare una programmazione, occorre invece premere l'icona tonda rossa con la X (Fig. 1).

La modalità Auto permette anche di simulare la presenza di gente in casa, accendendo e spegnendo sistematicamente la luce in una o più stanze: un ottimo deterrente contro i furti. Sunrise e Auto funzionano solamente se la Wi-Lamp viene lasciata sulle medesime modalità: per esempio se volessimo svegliarci alle ore 07:30 con i colori dell'alba, dopo aver impostato durata e orario dovremo lasciare la lampada tutta la notte in modalità Sunrise.

SCHEMA ELETTRICO

Il cuore della Wi-Lamp è una board Arietta G25 (nello schema elettrico è siglata U1): si tratta di un piccolissimo (53x25mm) modulo Linux embedded tutto italiano che con un processore a 400 MHz e 128 Mb di RAM DDR2 è in grado di svolgere il compito per cui è stato scelto. Lo storage è fornito da una SD Card (4GB o più) contenente la distribuzione di Linux, mentre la connettività è data dal modulo WiFi dedicato che utilizza un chip RaLink RT5370N. Oltre all'Arietta, nello schema elettrico troviamo una sezione di alimentazione, facente capo a U2, che permette di ottenere i 5V con cui viene alimentata Arietta, partendo dalla tensione applicata tramite il plug PWR o la morsettiera prevista.

Il LED RGB di potenza viene comandato con tensioni PWM tramite i pin 34, 36, 38 di Arietta, i cui segnali vengono amplificati dai transistor BD139 T1, T2, e T3, il cui collettore viene portato anche sulla morsettiera, per poter collegare eventuali strip a LED. I pulsanti sono collegati, tramite le resistenze di pull-up R7 e R8, ai pin 11 e 12 di Arietta, mentre il sensore di temperatura DS18B20 è collegato al pin 35.

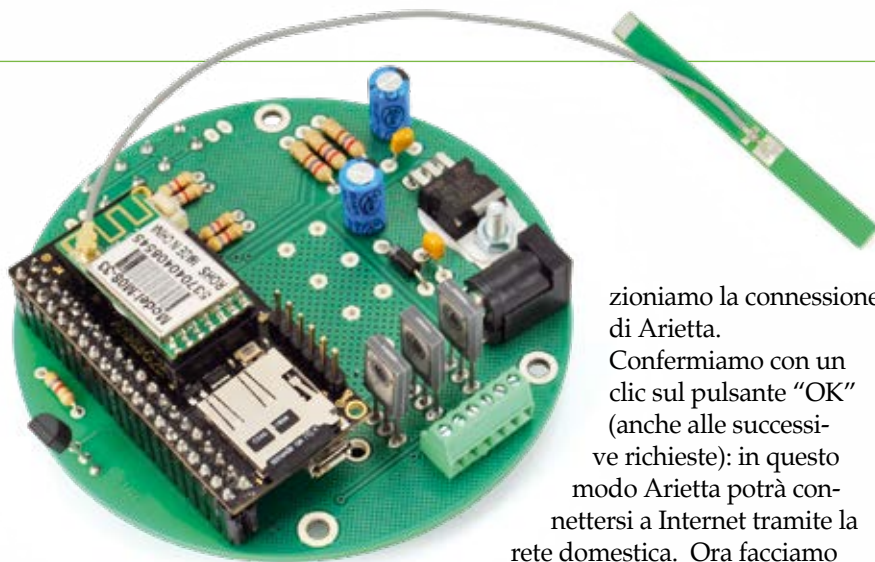


INSTALLAZIONE E CONFIGURAZIONE

Per preparare l'SD Card di Arietta possiamo scaricare sia l'immagine di Linux già pronta dal nostro sito www.elettronica.in.it, sia l'ultima immagine disponibile per Arietta dalla sezione "Download" del sito www.acmesystems.it (il file è "arietta_with_wifi.img.zip"). In tutti i casi dobbiamo scrivere l'immagine sulla SD Card con il software Win32DiskImager reperibile gratuitamente a questo indirizzo: sourceforge.net/projects/win32diskimager/. Una volta preparata l'SD, inse-

riamola in Arietta, montiamo il modulo WiFi (inclusa l'antenna) e colleghiamo il nostro PC tramite un cavo microUSB.

Durante la prima connessione, se l'installazione dei driver è andata a buon fine, in Gestione Dispositivi (da Windows Vista in poi...) del nostro computer visualizzeremo il dispositivo "USB Ethernet/RNDIS Gadget" nella sezione "Schede di rete"; nel caso in cui il nostro computer non riconosca Arietta, facciamo clic con il tasto destro sull'icona "dispositivo sconosciuto", selezioniamo "Aggiornamento



software driver...”, poi “Cerca il software driver nel computer” e successivamente “Scegli manualmente da un elenco di driver di dispositivo computer”. Scorriamo nell'elenco di sinistra fino ad individuare e selezionare la voce “Microsoft Corporation” oppure “Microsoft Corp.”, nell'elenco di destra selezioniamo “Remote NDIS based Internet Sharing Device” e completiamo con l'installazione facendo clic su “Avanti”: dopo qualche istante dovrebbe comparire il dispositivo nella sezione “Schede di rete”.

Dopo aver condiviso la nostra LAN con Arietta tramite il PC, dobbiamo condividere la connessione Internet; allo scopo spostiamoci in Pannello di controllo, selezioniamo “Rete e Internet”, successivamente “Centro connessioni di rete e condivisione” e facciamo clic sulla voce a sinistra “Modifica impostazioni scheda”. Nella finestra che si aprirà, tra le varie icone troveremo quella della connessione Internet e la connessione di Arietta (che potrebbe chiamarsi “Connessione alla rete locale LAN 2”): facciamo clic con il tasto destro sulla connessione Internet, selezioniamo “Proprietà” e facciamo clic sulla scheda “Condivisione”; attiviamo poi con un segno di spunta la voce “Consenti ad altri utenti in rete di collegarsi tramite la connessione Internet di questo computer” e nella casella “Connessione rete domestica” sele-

zioniamo la connessione di Arietta. Confermiamo con un clic sul pulsante “OK” (anche alle successive richieste): in questo modo Arietta potrà connettersi a Internet tramite la rete domestica. Ora facciamo clic con il tasto destro del mouse sulla connessione di rete di Arietta, selezioniamo la voce “Proprietà”, scorriamo la lista degli elementi fino a trovare la voce “Protocollo Internet versione 4 (TCP/IPv4)”, selezioniamola e facciamo clic sul pulsante “Proprietà”: nella scheda che si aprirà, clicchiamo su “Utilizza il seguente indirizzo IP” e compiliamo il campo “Indirizzo IP” con 192.168.10.20 e il campo “Subnet mask” con 255.255.255.0, poi confermiamo con il pulsante “OK” e chiudiamo la finestra della connessione di Arietta.

Se volete approfondire la connessione tra Arietta e PC, visitate la pagina web www.acmesystems.it/arietta_getting_started. Quanto alla condivisione dell'accesso Internet con le periferiche USB, trovate una guida dettagliata su <http://windows.microsoft.com/en-au/windows/using-internet-connection-sharing#1TC=windows-7>.

Ora possiamo connetterci in SSH ad Arietta tramite PuTTY: apriamo il programma e inseriamo 192.168.10.10 come indirizzo IP, lanciamo la connessione e connettiamoci con utente “root” e password “acmesystems”; nel caso avessimo scelto l'immagine già pronta di Wi-Lamp, dovremo inserire i dati della nostra connessione WiFi per fare connettere direttamente Arietta quando sarà montata sul PCB della lampada.

Se disponiamo di una connessione WiFi con protezione WPA (la mag-

gior parte dei casi) editiamo il file di configurazione con il comando `nano /etc/wpa_supplicant.conf`. Inseriamo il nome e la password della nostra rete negli appositi spazi, poi chiudiamo e salviamo il file con CTRL+X, Y e INVIO. Se invece disponiamo di una connessione WiFi con protezione WEP, dobbiamo aprire il file delle interfacce di rete con `nano /etc/network/interfaces`. Commentiamo le seguenti righe con il cancelletto (#) ad inizio riga:

```
wireless-essid any
pre-up wpa_supplicant -i wlan1 -c /
etc/wpa_supplicant.conf -B
post-down killall -q wpa_supplicant
```

Decommentiamo le seguenti righe rimuovendo il cancelletto (#):

```
wireless-essid WiFi_Name
wireless-mode managed
wireless-key s: WiFi_Password
```

Inseriamo al posto di “WiFi_Name” e “WiFi_Password” i dati della nostra rete WiFi, poi chiudiamo e salviamo il file con CTRL+X, Y e INVIO. In tutti e due i casi dobbiamo commentare la riga `gateway 192.168.10.20` contenuta all'interno del file `/etc/network/interfaces`; se non fosse commentata, Wi-Lamp non potrebbe connettersi ad Internet tramite la nostra rete WiFi per sincronizzare l'orologio di sistema. Con la riga in questione commentata, non potremmo più raggiungere Arietta con la connessione USB ma solamente tramite la connessione WiFi.

Lanciamo il seguente comando per attivare la connessione WiFi di Arietta: `ifdown wlan1 && ifup wlan1`. Dopo qualche istante possiamo verificare l'effettiva connessione e trovare l'IP della nostra lampada con il comando (leggiamo la riga “wlan1”, parametro “inet addr”): `ifconfig`. Se tutto è andato a buon fine e Arietta si è connessa alla

nostra rete WiFi, spegniamola con `shutdown -h now`. Nel caso avessimo scelto di scaricare l'immagine da preparare sul sito della ACME Systems, dopo esserci connessi come "root" con PuTTY spostiamoci nella home dell'utente acme con `cd /home/acme`.

Togliamo la verifica dei certificati per poter scaricare la repository, con il comando `git config --global http.sslverify false` e scarichiamo da GitHub i file necessari per l'installazione e il funzionamento di Wi-Lamp, tramite il comando `git clone https://github.com/open-electronics/wi-lamp.git`. Ora spostiamoci nella cartella del setup con `cd wi-lamp/utills` e lanciamo l'installazione di Wi-Lamp con il comando `bash setup.sh WiFi_Name WiFi_Password [WPA/WEP]` avendo cura di sostituire "WiFi_Name" e "WiFi_Password" con i dati della nostra rete WiFi, mentre come ultimo parametro indichiamo WPA se abbiamo una rete WiFi con questo tipo di protezione, altrimenti indichiamo WEP. Un esempio del comando per lanciare il setup potrebbe essere: `bash setup.sh MiaRete MiaPassword WPA`. Dopo aver premuto il pulsante INVIO partirà la procedura di installazione, che durerà circa 40 minuti e utilizzerà la connessione Internet del vostro PC condivisa con Arietta tramite USB. Il file `setup.sh` si occupa di fare l'update e l'upgrade della distribuzione Linux per aggiornare tutti i pacchetti, copiare il file di configurazione dei pin di Arietta nel boot, installare NTP per sincronizzare la data e l'ora di Arietta con Internet, installare le librerie di Python per utilizzare i pin configurabili di Arietta, cambiare l'hostname al sistema in modo da poter raggiungere la lampada digitando `http://wi-lamp` dal browser, copiare tutti i file del pannello di controllo web sotto `/var/www` facendo un backup dei file preesistenti e infine configurare

la rete WiFi con i parametri che gli abbiamo passato inizialmente (nome rete, password e tipo protezione). Alla fine del setup comparirà un paragrafo sul terminale che inizia con "Setup is finished, Arietta will now shutdown." e verrà indicato l'IP con cui raggiungere la lampada se l'indirizzo `http://wi-lamp` non funzionasse per problemi di restrizioni sul router di casa. Qualsiasi tasto spegnerà Arietta.

REALIZZAZIONE PRATICA

Passiamo alla costruzione della lampada: per prima cosa dovete preparare lo stampato seguendo le tracce pubblicate nel nostro sito web www.elettronica.in. Ottenuto il c.s. disponetevi i componenti a partire dal LED da 3W e dai due pulsanti, che vanno sul lato saldature; gli altri componenti (inclusa Arietta) vanno saldati dal lato opposto. Arietta va applicata mediante quattro colonnine distanziali da 25 mm, dopo averla installata e

configurata con il PC, come descritto nei paragrafi precedenti.

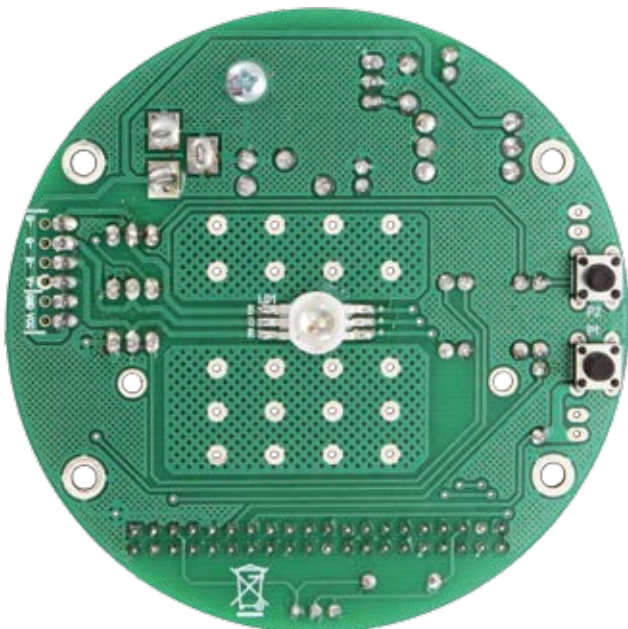
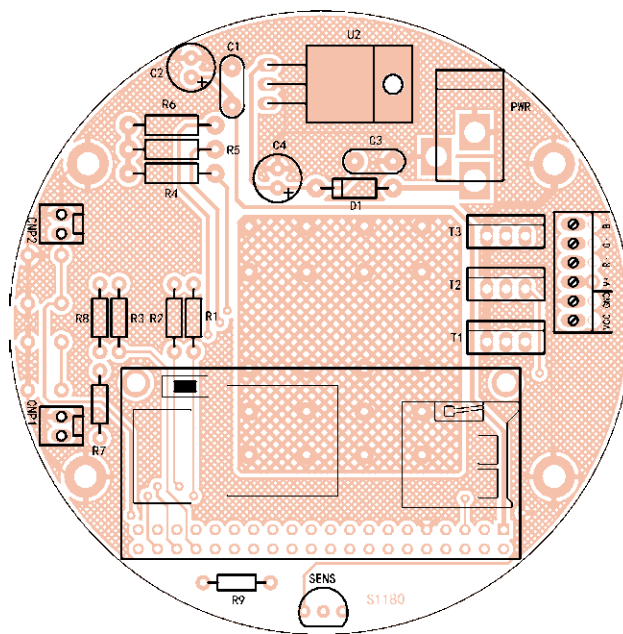
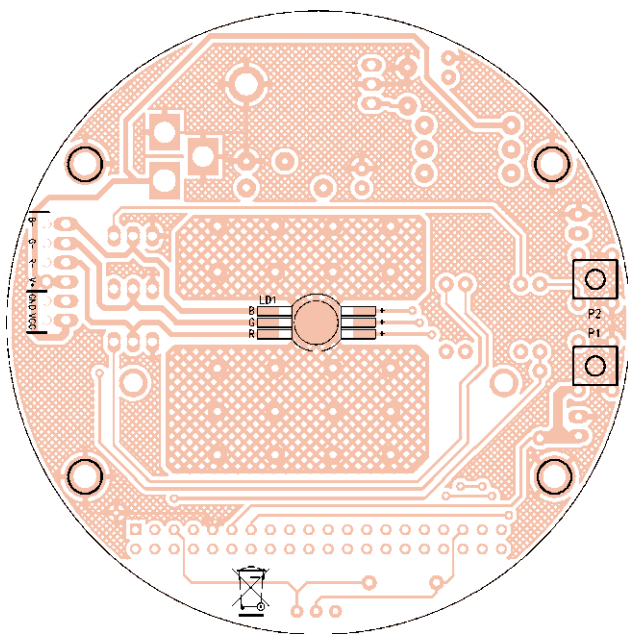
Il DS18B20 è preferibile tenerlo sollevato il più possibile dal c.s. ad evitare che la misura della temperatura venga influenzata dal calore prodotto dal LED di potenza. Completato il circuito, montiamo il corpo della lampada e alimentiamo tutto con un alimentatore da 12V, 2A. Nel caso volessimo collegare una striscia LED RGB, dovremo prevedere un alimentatore in grado di erogare la corrente richiesta. Alimentata, la lampada si colorerà per un paio di secondi di bianco per poi spegnersi: da quel momento potremo raggiungerla da qualsiasi dispositivo tramite il suo nome oppure l'indirizzo IP che abbiamo ricavato in fase di configurazione. Se volessimo spegnere completamente la lampada, dopo aver utilizzato il pulsante "Shutdown Lamp", dovremo attendere qualche secondo prima di scollegare l'alimentatore (la lampada si colorerà di bianco fisso).

Configurazione pin di Arietta

Arietta G25 dispone di ben 40 pin utilizzabili per diverse funzionalità: tre linee seriali, due bus I²C, quattro bus SPI, quattro convertitori A/D, quattro linee PWM, un collegamento "1 wire" e un collegamento per il bus audio. Siccome alcuni sono utilizzati da più funzionalità, per non creare conflitti abbiamo a disposizione un configuratore visuale dei pin raggiungibile all'indirizzo `http://dts.acmesystems.it/arietta/`: tramite questa pagina web è possibile impostare i pin che utilizziamo nelle varie modalità, in modo da creare un file di configurazione chiamato `acme-arietta.dtb` che dovremo scaricare tramite il menu "Compile" e il pulsante "Compile". Una volta scaricato, dovremo posi-

zionarlo nella root dell'SD Card di Arietta; nel caso fossimo connessi tramite SSH dovremo spostarci nella cartella di boot e scaricare direttamente il file dal sito della ACME Systems con i due comandi: `cd /boot`
`wget http://dts.acmesystems.it/arietta/queue/[percorso generato]`
`-O acme-arietta.dtb`

e riavviare Arietta per rendere effettive le modifiche. Per realizzare Wi-Lamp abbiamo attivato le voci PWM0, PWM1, PWM2 per controllare il LED RGB, la PC2 per collegare il DS18B20, mentre per i due pulsanti non abbiamo selezionato alcuna voce, collegandoli direttamente ai pin GPIO.



Elenco Componenti:

R1 ÷ R3: 4,7 kohm
 R4 ÷ R6: 270 ohm 1/2W
 R7 ÷ R9: 4,7 kohm
 C1, C3: 100 nF multistrato
 C2, C4: 100 µF 25 VL elettrolitico
 U1: ARIETTA-G25
 U2: 7805
 D1: 1N4007

T1 ÷ T3: BD139
 LD1: LEDRGB3W
 SENS: DS18B20
 P1, P2: Microswitch

Varie:
 - Modulo WIFI-2ANT-01
 - Plug alimentazione

- Morsettiera 6 poli passo 2,54mm
 - Vite 3MA 8mm
 - Dado 3MA (5 pz.)
 - Torretta M/F 25mm (4 pz.)
 - Strip femmina 20 poli (2 pz.)
 - Strip maschio 20 poli (2 pz.)
 - Circuito stampato S1180

FUNZIONAMENTO

Il codice sorgente della Wi-Lamp (Fig. 2) è così organizzato:

- **common/**: cartella con le librerie Javascript di uso comune, CSS e immagini per il pannello di controllo web;
- **engine/**: cartella che contiene i file *lamp.py* e *buttons.py* eseguiti ad ogni avvio di Arietta; il primo gestisce tutte le modalità della lampada mentre il secondo rileva la pressione di uno dei due pulsanti sul PCB e cambia la modalità della lampada; *lampio.py* gestisce la lettura e la scrittura dei file di testo e viene richiamato dai primi due file, mentre *setled.sh* si occupa di colorare effettivamente il LED RGB;
- **storage/**: questa cartella contiene i file di testo contenenti le impostazioni delle modalità:
 - *auto*: contiene tutte le schedulazioni per l'accensione automatica della lampada divise per riga, nel formato giorno/hh:mm/durata/R,G,B;
 - *buttons*: contiene le due modalità, separate dalla virgola, richiamabili dai due pulsanti sul PCB;
 - *color*: contiene il colore che la lampada avrà in modalità Self, nel formato R,G,B;
 - *mode*: contiene la modalità attuale della lampada, le modalità partono dalla numero zero (OFF) e sono numerate in sequenza;
 - *shutdown*: contiene 1 se abbiamo premuto il pulsante "Shutdown Lamp", 0 altrimenti;
 - *speed*: contiene i valori da 0 a 4 per regolare la velocità sia in Fade che in Party;
 - *sunrise_duration*: contiene la durata in minuti (da 1 a 20) dell'alba;
 - *sunrise_time*: contiene l'ora e i minuti di partenza dell'alba nel formato hh:mm;

Anche per Halloween...

Halloween è alle porte e possiamo sfruttare l'occasione per "vestire" la nostra Wi-Lamp con una zucca o un teschio realizzato con la stampante 3D 3Drag. Per stampare la zucca possiamo utilizzare il file reperibile su www.thingiverse.com/thing:134066 mentre per il teschio possiamo utilizzare il file www.thingiverse.com/thing:518748.

Dopo aver realizzato la zucca o il teschio, posizioniamoci all'interno Wi-Lamp e sorprendiamo i nostri amici attivando la modalità Fire; se la festa dovesse prendere un ritmo più vivace potremo contare sulla modalità Party e, alla fine, passare a una più rilassante modalità Fade.



- *sunset_duration*: contiene la durata in minuti (da 1 a 20) del tramonto;
- *sunset_start*: contiene 1 se abbiamo premuto il pulsante per fare partire il tramonto, altrimenti contiene 0;
- *temperature*: contiene i valori di temperatura massima e minima impostati nella modalità Temperature, separati da virgola;
- *temperature_sensor*: ad ogni avvio viene popolato con il nome del sensore di temperatura rilevato da Arietta;
- **utils/**: questa cartella contiene il file di configurazione dei pin di Arietta che viene copiato durante la fase di installazione nella directory /boot, il file *setup.sh* che prepara arietta per Wi-Lamp e installa tutti i componenti, e il file *startup.sh* che viene eseguito ad ogni avvio di Arietta lanciando i file *lamp.py* e *buttons.py*;
- **index.php**: è il pannello di controllo web della Wi-Lamp;
- **wi-lamp.php**: si occupa di ricevere comandi in asincrono da *index.php* e popolare tutti i file nella cartella *storage/*.

Il file *lamp.py* può essere quindi considerato lo script principale della

Wi-Lamp: ogni secondo controlla se la modalità è cambiata, mentre ogni 0,15 secondi estrae i dati inerenti alla modalità corrente; se fossimo in modalità Fade verrebbe letto il file *speed* per controllare la velocità del fading. Approfondiamo ora alcuni aspetti delle modalità: OFF setta semplicemente i colori R, G e B a zero ogni 0,8 secondi per tenere spenta la lampada. Self legge continuamente il file *color* e ogni 0,2 secondi setta il valore letto al LED. Fade, ogni volta che viene richiamata, manda avanti di uno step la sfumatura corrente abbassando un colore e alzando quello successivo: si parte infatti da una situazione iniziale del tipo 255, 0, 0 (rosso) per poi sfumare lentamente a 0, 255, 0 (verde) e concludere il giro con 0, 0, 255 (blu); la velocità con cui viene richiamata questa funzione dipende dal valore contenuto del file *speed*. Party randomizza i tre colori del LED e li setta con un intervallo di tempo dipendente dal valore contenuto nel file *speed*. Sunset esegue essenzialmente tre operazioni, una esclude l'altra: se è stato premuto il pulsante di avvio del tramonto (leggendo dal file *sunset_start*), imposta la lampada in modalità "occupata", se la lampada è in modalità "occupata" ma è trascorsa la durata

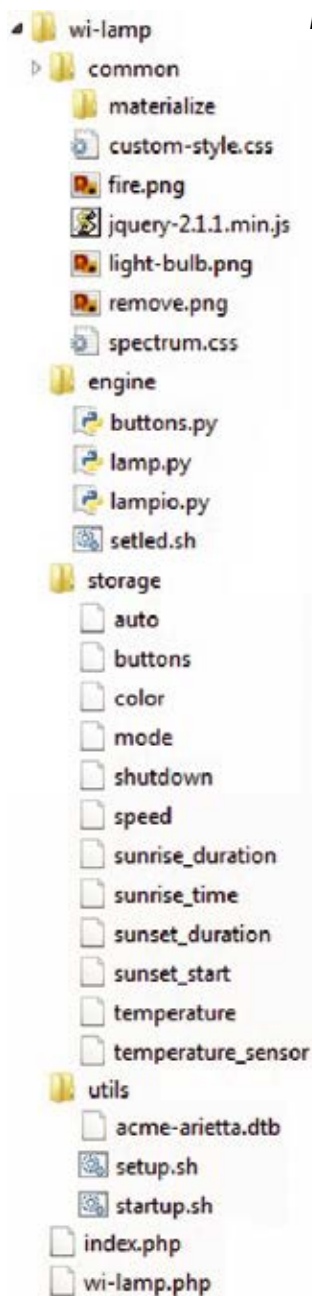


Fig. 2

lampada, da spenta, passa tutte le sfumature tipiche dei colori dell'alba fino ad accendersi completamente di bianco; anche qui è presente la durata in minuti impostabile (file *sunrise_duration*). Temperature legge ogni 0,8 secondi la temperatura misurata dal sensore e imposta la lampada del colore ricavato dalla sfumatura blu-rosso e dai valori di temperatura minima e massima precedentemente impostati sul file *temperature*. Fire randomizza un colore dalla sfumatura rosso-giallo ogni 0,1 secondi, non è possibile impostare la velocità di questa animazione. La modalità Auto (chiamata Automatic nel codice sorgente) estrae ogni 0,8 secondi la lista delle schedulazioni di accensione della lampada dal file *auto*, le analizza una ad una per capire se il giorno e l'orario di partenza coincidono con quelli attuale e, nel caso, imposta la lampada in modalità occupata per la durata impostata nella schedulazione settando il LED del colore preimpostato; alla fine spegne il LED e "libera" la lampada, che torna ad essere pronta per un'altra accensione automatica.

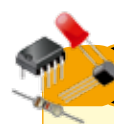
PERSONALIZZAZIONE

Se volessimo connettere più lampade alla stessa rete WiFi, dovremo cambiare l'hostname di ciascuna per non avere conflitti; connettiamoci quindi tramite SSH alla Wi-Lamp cui vogliamo cambiare nome (se è già connessa alla rete WiFi possiamo accedervi tramite l'indirizzo IP fornito in fase di configurazione oppure con l'hostname "wi-lamp"). Editiamo i seguenti file per cambiare nome alla lampada:

```
nano /etc/hosts
nano /etc/hostname
```

Chiudiamo e salviamo i file con CTRL+X, Y e INVIO; infine riavviamo Arietta con *reboot*. Disponendo di un servizio DNS gratuito come

www.noip.com e agendo sulla pagina di configurazione del nostro router, possiamo inoltrare all'esterno la porta 80 dell'indirizzo IP della nostra lampada (Port Forwarding): sarà così possibile controllare Wi-Lamp anche da remoto. Per customizzare la lampada possiamo agire su diversi parametri editando il file *lamp.py* con il comando `nano /var/www/engine/lamp.py`. Nella sezione "Properties declaration" troviamo l'array *StartColor* che contiene i valori di partenza del LED quando viene caricata una nuova modalità: i valori sono R, G, B e vanno da 0 (spento) a 255 (colore al massimo della potenza). Tutte le modalità partono con il LED spento, tranne *Fade* che parte con il LED colorato di rosso (per poi sfumare al giallo). Gli array *SpeedFade* e *SpeedParty* controllano la pausa in secondi prima del passaggio ad un nuovo colore o sfumatura delle due rispettive modalità; tramite lo slider dal pannello di controllo web viene scritto un valore da 0 a 4 nel file *speed* che successivamente è utilizzato per trovare la pausa: possiamo modificare questi valori per variare le velocità di *Fade* e *Party*. ■



per il MATERIALE

La lampada WiFi è disponibile in scatola di montaggio e comprende il modulo Arietta G25, il modulo WiFi, il LED di potenza da 3W e la micro SD card contenente il sistema operativo. Il kit (cod. FT1180K) viene venduto a 69,00 Euro. Arietta G25 da 128 MB (cod. ARIETTA-G25) e il modulo WiFi (cod. WIFI-2ANT-01) sono disponibili anche separatamente rispettivamente a 24,40 Euro e 14,00 Euro.

Il materiale va richiesto a:

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

Tel: 0331-799775 • Fax: 0331-792287
<http://www.futurashop.it>

del tramonto (i minuti impostabili dal pannello web letti dal file *sunset_duration*) libera la lampada e la setta su OFF; mentre se la lampada è "occupata" ma non sono ancora trascorsi i minuti del tramonto, si occupa di mandare avanti la sfumatura che dal bianco passa attraverso tutti i colori del tramonto fino a spegnersi completamente. Sunrise è una modalità molto simile a Sunset, con la differenza che l'animazione dell'alba parte al raggiungimento dell'orario impostato dal pannello di controllo (file *sunrise_time*) e la



Windows 10



Sperimentiamo lo strumento Virtual Shield per Arduino presente nell'ultimo sistema operativo Microsoft, per realizzare un'applicazione di geo-fence basata su smartphone, per attivare un relé quando il telefono si trova in una certa zona.

RELE' GEOLOCALIZZATO CON VIRTUAL SHIELD

di FABRIZIO GUGLIELMINO

Nei due fascicoli precedenti vi abbiamo introdotto dei progetti correlati con l'ingresso di Microsoft nel mondo dei Makers ed in particolare ci siamo soffermati sulla distribuzione di Windows per Raspberry Pi, chiamata Windows 10 IoT Core e nata per semplificare il ciclo di sviluppo delle applicazioni. Invece ora vediamo quanto è stato fatto per Arduino e in particolare vedremo l'utilizzo di Windows Virtual Shield, realizzando un semplice progetto per dimostrarne le potenzialità. Ciò che vogliamo realizzare è, pur nella sua semplicità, utilizzabile in contesti reali: si tratta di un relé che si attiva quando si è in prossimità di un punto prestabilito. Abbiamo citato questo tipo di utilizzo del Virtual Shield già nell'articolo precedente, come soluzione per il comando di apertura del cancello del garage in luogo del classico telecomando; nel progetto qui descritto, quando siamo nelle vicinanze sarà Arduino, installato in auto, che invierà per noi il comando

utilizzando uno smartphone dotato di ricevitore GPS (quest'ultimo fornirà i dati sulla posizione).

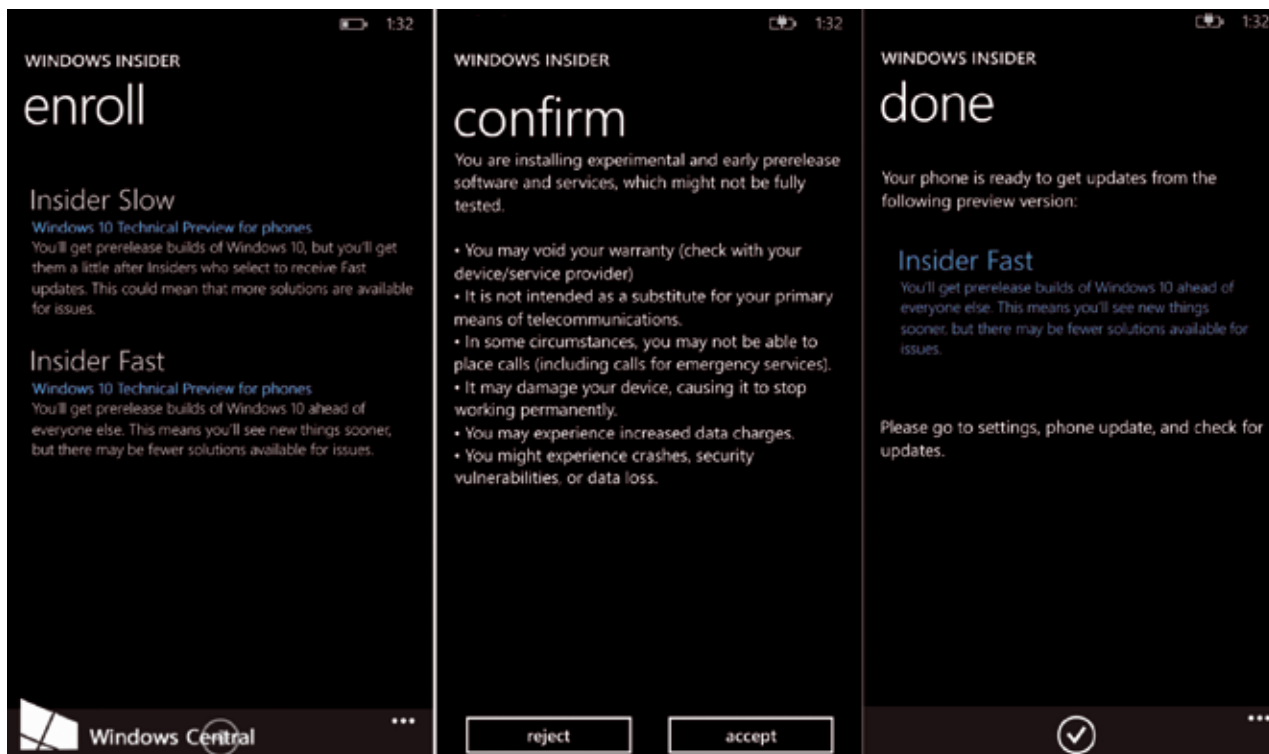
MATERIALE NECESSARIO

Il progetto richiede questi dispositivi:

- Arduino Uno o una scheda compatibile;
- modulo Bluetooth FT1018M con RN42 (presentato nel numero 168 della rivista);
- smartphone con Windows Phone 10 preview (vanno bene per esempio Lumia 520, Lumia 635/630);
- PC con Windows 10.

Dato che si tratta di un progetto didattico su Windows Virtual Shield useremo una breadboard sulla quale costruiremo il circuito con il relé comandato da Arduino. I componenti esterni con cui andiamo ad interfacciare Arduino sono semplicemente un relé e quel poco che serve per ottenerne l'attivazione (di fatto, un transistor e una resistenza).

Fig. 1



PREPARAZIONE DELL'AMBIENTE

La preparazione dell'ambiente è abbastanza articolata e si suddivide in tre step principali:

1. configurazione dell'ambiente di sviluppo su PC;
2. configurazione del telefono;
3. configurazione dell'IDE di Arduino.

Iniziamo col predisporre l'ambiente di sviluppo sul PC basato su Windows 10: allo scopo dovremo scaricare e installare Visual Studio 2015 (la versione Community è gratuita ed è scaricabile da <https://www.visualstudio.com/it-it/vs-2015-product-editions>.

Una volta installato l'ambiente di sviluppo, lo utilizzeremo per installare la Universal App "Virtual Shield for Arduino" sul telefono. Questa App, fornita in forma sorgente, si occupa di gestire la comunicazione con Arduino, esponendo i sensori e le componenti utilizzate del telefono. I sorgenti dell'App sono disponibili su GitHub all'indirizzo <https://github.com/ms-iot/virtual-shields-arduino>; per chi conosce GitHub è possibile fare il clone del repository (Visual Studio supporta GitHub direttamente dall'ambiente). Per chi non avesse dimestichezza con GitHub, l'approccio più semplice

è scaricare lo zip con i sorgenti e fare doppio clic su *Shield.sln*, così da aprirli con Visual Studio. Caricati i sorgenti con Visual Studio, è il momento di provare ad eseguire "Build" (o "Compila" per la versione italiana); se non rilevate errori, avete completato la configurazione del PC. Passiamo quindi al telefono: in questo caso l'operazione è un po' più semplice. L'ideale è disporre di un telefono non utilizzato come nostro telefono principale, perché al momento della pubblicazione di questo articolo la versione di Windows 10 per smartphone è ancora in preview e Microsoft non dà alcuna garan-

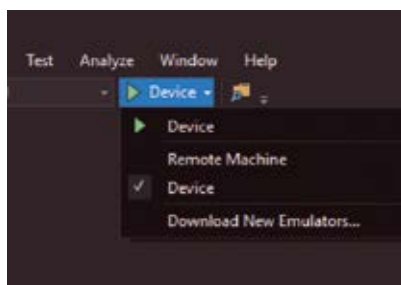


Fig. 2



Fig. 3

zia sulla sua affidabilità: installarla significa accettare il rischio che il telefono possa funzionare male. Per installare la preview di Windows 10 è sufficiente scaricare dallo store l'App "Windows Insider", l'applicazione una volta eseguita procede in modo semi automatico, l'unico requisito è essere registrati al programma Insider Preview ma anche questa registrazione è gratuita e si può fare seguendo le indicazioni dell'App. Prima di avviare il processo di installazione di Windows 10 preview, l'App ci chiederà se usare l'aggiornamento "Fast" o "Slow": nel primo caso il sistema riceverà aggiornamenti più frequenti ma potenzialmente meno stabili; nel secondo caso ci saranno meno aggiornamenti, ma più stabili. Per il nostro test, entrambe le condizioni vanno bene, quindi lasciamo a voi la decisione (Fig. 1). Ad installazione terminata siamo pronti a collegare il telefono al computer e, utilizzando Visual Studio, ad installare "Virtual Shield for Arduino"; se i passi precedenti sono stati fatti senza problemi, l'operazione consiste semplicemente nel cliccare sulla freccina tipo "play" vicino alla scritta "Device" che si trova nella barra supe-

riore (Fig. 2). Nota importante: può capitare che Visual Studio dia un errore perché la versione per la quale è configurato il progetto è successiva a quella del sistema operativo presente sul telefono; purtroppo questo è un problema noto e bisogna intervenire manualmente per risolverlo. Nel caso vi trovaste in questa condizione, dovete per prima cosa annotarvi la versione che Visual Studio ha rilevato nel telefono, che verrà mostrata contestualmente all'errore, aprire con notepad il file *Shield.csproj* contenuto nella cartella dei sorgenti ed inserire la versione letta nel tag *TargetPlatformMinVersion*; per esempio se leggessimo 10.0.10149.0, dovremmo impostare il tag così: `<TargetPlatformMinVersion>10.0.10149.0</TargetPlatformMinVersion>`. Questo serve ad indicare che la versione minima supportata dal progetto è quella specificata e, coincidendo con quella del telefono, non si presenterà più il problema. Ora non resta che configurare la parte Arduino: per prima cosa bisogna scaricare una versione recente dell'IDE (dalla 1.6 in su). Installato l'IDE dobbiamo scaricare la libreria *ArduinoJson*; per fortuna nelle versioni recenti

è stato inserito il *LibraryManager* (Fig. 3) accessibile dal menu *Sketch->Include Libraries->Manage Libraries...*: con questo comando si apre un popup dal quale scegliere la libreria da installare, filtrando *ArduinoJson* viene mostrata la libreria che stiamo cercando e basterà impartire il comando *Install* per installarla (Fig. 4). L'ultimo step è scaricare ed installare le librerie per il virtual shield, che purtroppo non sono ancora inserite all'interno del *Library Manager* e pertanto bisogna procedere con l'installazione manuale. L'operazione, per fortuna, non è eccessivamente complessa: si tratta di scaricare il repository presente all'indirizzo <https://github.com/ms-iot/virtual-shields-arduino> e copiare tutto il contenuto nella cartella dove l'IDE tiene le librerie (tipicamente *Documents\Arduino\libraries*).

L'IMMANCABILE HELLO WORLD

Giunti a questo punto dovremmo avere l'ambiente pronto per il primo test, che, come da tradizione, sarà il classico Hello World. Il "quasi" è d'obbligo perché non abbiamo ancora messo mano all'hardware; in realtà in questa prima fase, dello schema dell'applicazione, mostrato in Fig. 5,

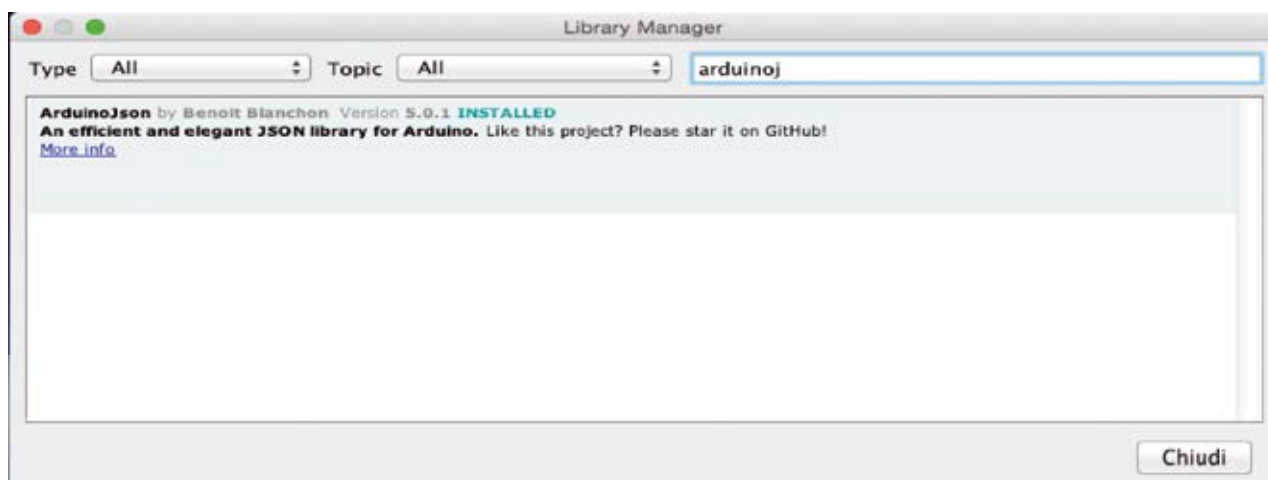


Fig. 4

Listato 1

```
#include <ArduinoJson.h>
#include <VirtualShield.h>
#include <Text.h>
VirtualShield shield;
Text screen = Text(shield);
void refresh(ShieldEvent* shieldEvent)
{
    screen.clear();
    screen.printAt(1, "Hello World");
}
void setup()
{
    shield.setOnRefresh(refresh);
    shield.begin();
}
void loop()
{
    shield.checkSensors();
}
```

ci basta collegare il modulo Bluetooth alla scheda Arduino. Il primo test è molto semplice e consiste nel creare uno sketch che utilizzerà il display del telefono per scrivere "Hello World". Il codice da caricare è quello nel **Listato 1**.

In breve, viene creato l'oggetto di tipo VirtualShield (chiamato shield nell'esempio), che rappresenta l'interfaccia di comunicazione con il telefono. L'oggetto viene usato insieme alla classe Text, che espone le funzionalità per interagire con il display del telefono. Una particolarità dell'esempio è che le chiamate per visualizzare il testo non sono fatte direttamente nella funzione loop ma sono nella funzione refresh; questa viene a sua volta passata come parametro al metodo setOnRefresh. Tale

meccanismo delega all'oggetto shield la chiamata della funzione quando è necessario (per esempio all'avvio del programma o dopo la connessione bluetooth). Per veder funzionare l'esempio è necessario abbinare il modulo Bluetooth al telefono: la procedura è quella classica che si fa per abbinare i dispositivi Bluetooth. Dalle impostazioni si seleziona "Dispositivi" poi "Bluetooth" e da questa videata si attiva il Bluetooth (se spento), attendendo qualche secondo mentre vengono attivati i dispositivi nel raggio di azione. Il nome del dispositivo da abbinare sarà qualcosa tipo RN42-1234 (dove 1234 è diverso per ogni modulo); selezionandolo avviene l'abbinamento. Ora avviamo l'App "Virtual Shield for Arduino" sul telefono,

alla partenza della quale viene proposto lo schermo nero (**Fig. 6**). Questo è il comportamento corretto perché l'area nera è quella resa disponibile ad Arduino per visualizzare contenuti testuali o grafici; in basso a destra è presente il solito tasto con i tre puntini "...", che indica la presenza di un menu a scomparsa. Selezioniamolo e verrà presentato un menu con alcuni pulsanti, dei quali quello che ci interessa è "Settings". La sezione "Connections" è quella che cerchiamo, dove saranno visibili tutti i dispositivi Bluetooth abbinati con il cellulare tra i quali il nostro modulo. Selezionandolo e premendo "Connect" concludiamo la fase di predisposizione. A questo punto, tornando alla videata principale (con il tasto back) dovremmo vedere apparire "Hello World" inviato dal nostro sketch caricato su Arduino.

IL PROGETTO

Ma vediamo ora come attivare un relé quando siamo in prossimità di una determinata zona. Il circuito da realizzare è quello di **Fig. 5**, dove troviamo un transistor usato in modalità switching per comandare un relé. Ai due contatti della morsettiere, facenti capo allo scambio normalmente aperto del relé (C-NA) si collega il pulsante del trasmettitore del telecomando dell'apricancello. Quando il relé verrà eccitato, il contatto risulterà chiuso e attiverà la trasmissione da parte del telecomando, il che invierà il comando di apertura del cancello motorizzato.

I passaggi logici che descrivono il funzionamento del circuito sono:

1. lettura periodica della posizione GPS;
2. calcolo della distanza rispetto ad un punto prestabilito (il cancello da aprire);
3. attivazione del relé se la di-

Listato 2

```
void gpsEvent(ShieldEvent* shieldEvent)
{
    float distance = calc_dist(gps.Latitude, gps.Longitude, myLatitude, myLongitude);
    String diststr = String("Distanza ") + String(distance) + String(" mt");

    screen.printAt(3, diststr);
    if (distance < threshold){
        screen.printAt(5, "Rele' ON", GREEN);
        digitalWrite(PIN_RELE, HIGH);
    }else{
        screen.printAt(5, "Rele' OFF", RED);
        digitalWrite(PIN_RELE, HIGH);
    }
}
```

stanza è inferiore ad una soglia prestabilita.

Per la lettura della posizione utilizziamo una classe delle Virtual Shield specificatamente fatta per questo scopo; come per Text viene inizializzata passando l'oggetto shield del quale farà uso per comunicare con il telefono:

```
Geolocator gps =  
Geolocator(shield);
```

La variabile gps rappresenta l'oggetto che utilizziamo per leggere la posizione; la lettura periodica avviene tramite il meccanismo già visto delle callback. Nel setup del nostro sketch avremo la chiamata che imposta la funzione da chiamare periodicamente per leggere la posizione; nel nostro caso effettueremo la lettura ogni 15 secondi. Notate che è importante bilanciare questo tempo in modo da non effettuare letture troppo frequenti, che penalizzerebbero l'autonomia, visto che parliamo di un dispositivo da alimentare a batteria. Allo stesso tempo, se facciamo letture troppo rade genereremo ritardi nell'attivazione del relé rispetto all'ingresso nell'area di prossimità dell'obiettivo e per prudenza dovremo avvicinarci solo dopo aver visto che il cancello si apre.

La linea di codice che imposta la callback è `gps.setOnEvent(gpsEvent)`.

Vediamo quindi la funzione `gpsEvent`, che è quella che abbiamo richiesto venga chiamata quando ci sono aggiornamenti nella lettura della posizione: è riportata nel **Listato 2**.

Le coordinate relative alla posizione vengono rese disponibili in `gps.Latitude` e `gps.Longitude`; avremo precedentemente impostato nel sorgente altre due coordinate corrispondenti al punto rispetto al quale calcolare

la distanza (nel sorgente sono `myLatitude` e `myLongitude`). La funzione calcola quindi la distanza tra le coordinate lette e quelle di riferimento; per questo chiamiamo un'altra funzione, `calc_dist`, che utilizza una formula matematica per il calcolo del caso (la funzione è riportata nel sorgente completo che trovate nel **Listato 4**). Ottenuta la distanza, creiamo una stringa (`diststr`) che utilizzeremo per visualizzare

quanto letto sul display del cellulare. Il vero cuore della funzione è il test che controlla se la variabile `distance` è minore della soglia prestabilita (indicata da `threshold`) e nel caso attiva il relé collegato al pin definito con `PIN_RELE`. L'esempio è volutamente semplificato: quando entriamo nell'area di interesse (che possiamo considerare un "geo-fence"), il relé rimane eccitato finché permaniamo al suo interno; ovvia-



Il lago diventa smart

Lake George è al centro di un ambizioso progetto di Ricerca Ambientale che prevede il dispiegamento di centinaia di sensori che stanno fornendo una enorme quantità di dati, la cui elaborazione potrà aiutare a preservare le caratteristiche di questo, come degli altri bacini degli Stati Uniti. Il Progetto Jefferson presso il Lake George, frutto di una collaborazione tra IBM Research, il Rensselaer Polytechnic Institute e il Fondo per Lake George si basa sull'applicazione dell'Internet of Things. Il progetto ha già permesso di scoprire che inquinanti come i cloruri (provenienti dal sale antigelo sparso sulle strade) sono triplicati, le alghe sono aumentate di un terzo e di affinare modelli matematici di sviluppo del suo ecosistema. I sensori distribuiti sul fondo del lago hanno confermato recentemente l'esistenza di "un'onda subacquea fantasma" che può essere alta fino a 10 metri e corre a circa 3 metri sotto la superficie del lago, lungo più di 50 Km. L'infrastruttura informatica che abilita il progetto Jefferson coinvolge molteplici piattaforme, che vanno da un supercomputer IBM Blue Gene/Q, che si trova in un data center nel campus Rensselaer, a elementi di intelligent computing e altre tecnologie Internet of Things situate su diverse piattaforme di sensori in e intorno al lago.

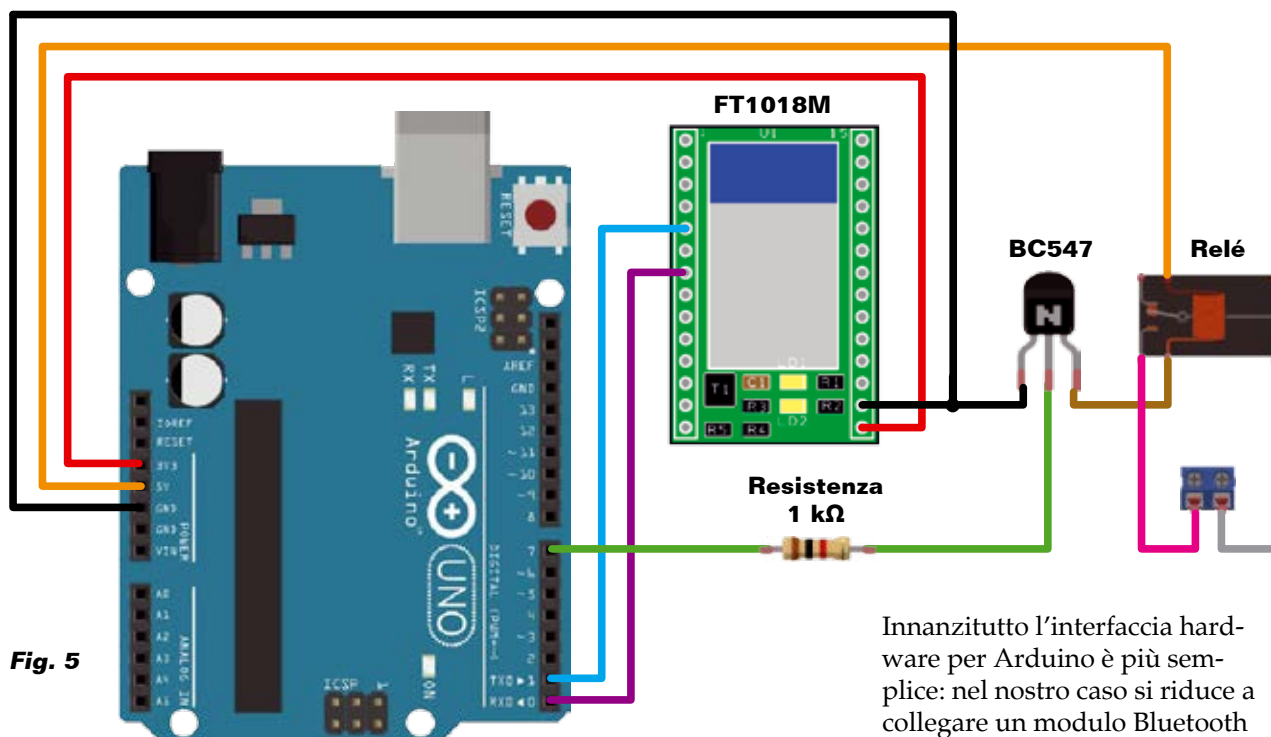


Fig. 5

mente è possibile introdurre una gestione per questa condizione ma abbiamo preferito mantenere il codice leggibile. Agganciare la callback non è però sufficiente se non impostiamo una chiamata periodica che scateni la lettura della posizione. Come è naturale in un programma per Arduino, questa attività viene svolta nel loop: lo vediamo nel **Listato 3**, che ne riporta la relativa porzione di codice.

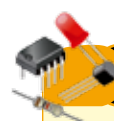
La richiesta di lettura delle coordinate della posizione avviene con la chiamata `gps.get()`, per gestire la frequenza di lettura

utilizziamo la funzione `millis()` in modo da effettuare la chiamata solo quando sono trascorsi almeno 15 secondi dalla richiesta precedente.

CONCLUSIONI

Coloro tra voi che hanno esperienza con Arduino potrebbero chiedersi quale sia la differenza tra usare uno shield GPS fisicamente montato sulla scheda e utilizzare il Virtual Shield. Funzionalmente avremmo lo stesso risultato, ma l'utilizzo di Windows Virtual Shield porta una serie di vantaggi.

Innanzitutto l'interfaccia hardware per Arduino è più semplice: nel nostro caso si riduce a collegare un modulo Bluetooth e un relé. Il nostro progetto, inoltre, utilizza il display dello smartphone per visualizzare dei messaggi; se volessimo realizzare il progetto in modo "classico" dovremmo preoccuparci di interfacciare un display e magari procurarci un case per posizionarlo, considerato che il caso d'uso tipico del nostro progetto è quello di comando dell'apertura cancello da posizionare in auto,



per il MATERIALE

Tutto il materiale necessario per realizzare questo progetto è disponibile presso Futura Elettronica, in particolare la board Open Source Arduino UNO (cod. ARDUINOUNOREV3) è in vendita al prezzo di 24,50 Euro mentre il modulo Bluetooth con RN42 (cod. FT1018M) viene fornito a 37,00 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>

Listato 3

```
const long interval = 1000 * 15; //15 secondi
long nextGPS = 0;

void loop()
{
  if (millis() > nextGPS) {
    nextGPS = millis() + interval;

    gps.get();
  }

  shield.checkSensors();
}
```

come abbiamo anticipato nell'introduzione. Per questo utilizzo abbiamo bisogno che Arduino sia in grado di accedere ai dati di geolocalizzazione solo quando anche noi siamo in auto; usare un modulo GPS integrato avrebbe senso nel caso di un antifurto, ma in questo caso invece possiamo risparmiare l'acquisto del componente.

Infine, il fatto di usare uno shield virtuale che dispone di una serie molto nutrita di sensori ci permette di estendere in futuro il nostro progetto senza agire sul circuito. Potremmo per esempio fare emettere un suono o sintetizzare un messaggio vocale quando siamo prossimi al cancello e per questo basterebbero poche righe di codice in più nello sketch.



Fig. 6

Listato 3 – sketch completo

```
#include <ArduinoJson.h>
#include <VirtualShield.h>
#include <Text.h>
#include <Colors.h>
#include <Geolocator.h>

VirtualShield shield;
Text screen = Text(shield);
Geolocator gps = Geolocator(shield);
// Coordinate del punto rispetto al quale misurare la distanza
float myLatitude = 45.092949;
float myLongitude = 7.523704;
// Soglia entro la quale attivare il Relè
float threshold = 50;
// Pin al quale è collegato il Relè
int PIN_RELE = 7;
void gpsEvent(ShieldEvent* shieldEvent)
{
    float distance = calc_dist(gps.Latitude, gps.Longitude, myLatitude, myLongitude);
    String diststr = String("Distanza ") + String(distance) + String(" mt");

    screen.printAt(3, diststr);
    if (distance < threshold){
        screen.printAt(5, "Rele' ON", GREEN);
        digitalWrite(PIN_RELE, HIGH);
    }else{
        screen.printAt(5, "Rele' OFF", RED);
        digitalWrite(PIN_RELE, HIGH);
    }
}

void setup()
{
    shield.begin();
    screen.clear(ARGB(0,0,0));
    screen.printAt(1, "Rele' Geolocalizzato");
    gps.setOnEvent(gpsEvent);
    pinMode(PIN_RELE, OUTPUT);
}

const long interval = 1000 * 15; //15 secondi
long nextGPS = 0;
void loop()
{
    if (millis() > nextGPS) {
        nextGPS = millis() + interval;

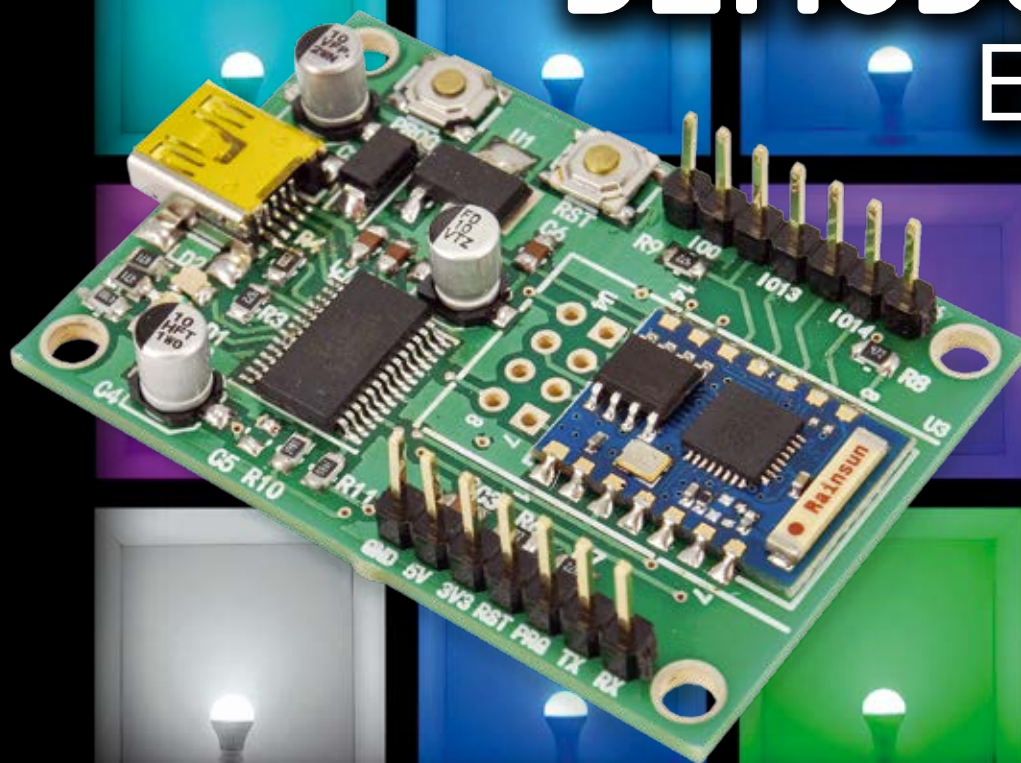
        gps.get();
    }

    shield.checkSensors();
}

// Calcolo della distanza tra due coordinate (in metri)
float calc_dist(float flat1, float flon1, float flat2, float flon2)
{
    float dist_calc=0;
    float dist_calc2=0;
    float diflat=0;
    float diflon=0;
    diflat=radians(flat2-flat1);
    flat1=radians(flat1);
    flat2=radians(flat2);
    diflon=radians(flon2)-(flon1));
    dist_calc = (sin(diflat/2.0)*sin(diflat/2.0));
    dist_calc2= cos(flat1);
    dist_calc2*=cos(flat2);
    dist_calc2*=sin(diflon/2.0);
    dist_calc2*=sin(diflon/2.0);
    dist_calc +=dist_calc2;
    dist_calc=(2*atan2(sqrt(dist_calc),sqrt(1.0-dist_calc)));
    dist_calc*=6371000.0;
    return dist_calc;
}
```


DEMOBOARD

ESP-03



Sperimentiamo
e prototipiamo
applicazioni
con la board
wireless
basata sul
modulo
ESP8266.

dell'ing. MIRCO SEGATELLO

Riprendiamo la trattazione del chip ESP8266, di cui abbiamo già parlato nel fascicolo di settembre 2015, dove abbiamo descritto la nuova scheda Fishino i cui elementi principali erano il microcontrollore ATmega328 e il modulo ESP03, quest'ultimo basato proprio su un ESP8266. Il modulo, in questo caso, fungeva solo da interfaccia WiFi e la board lasciava al chip Atmel l'onere di gestire le funzionalità richieste tramite appositi comandi. La piccola scheda ESP03 contiene, oltre all'interfaccia WiFi, anche un microcontrollore programmabile grazie al quale abbiamo potuto realizzare questa demoboard che ci consente di creare applicazioni wireless in maniera ancora più semplice e con un numero ridotto di componenti. Con questa scheda è come avere un microcontrollore con il WiFi integrato pronta all'uso. Non male... vero?! Come accennato il modulo è del tutto autonomo, però è necessario potervi accedere da PC per poterlo programmare; abbiamo pertanto previsto

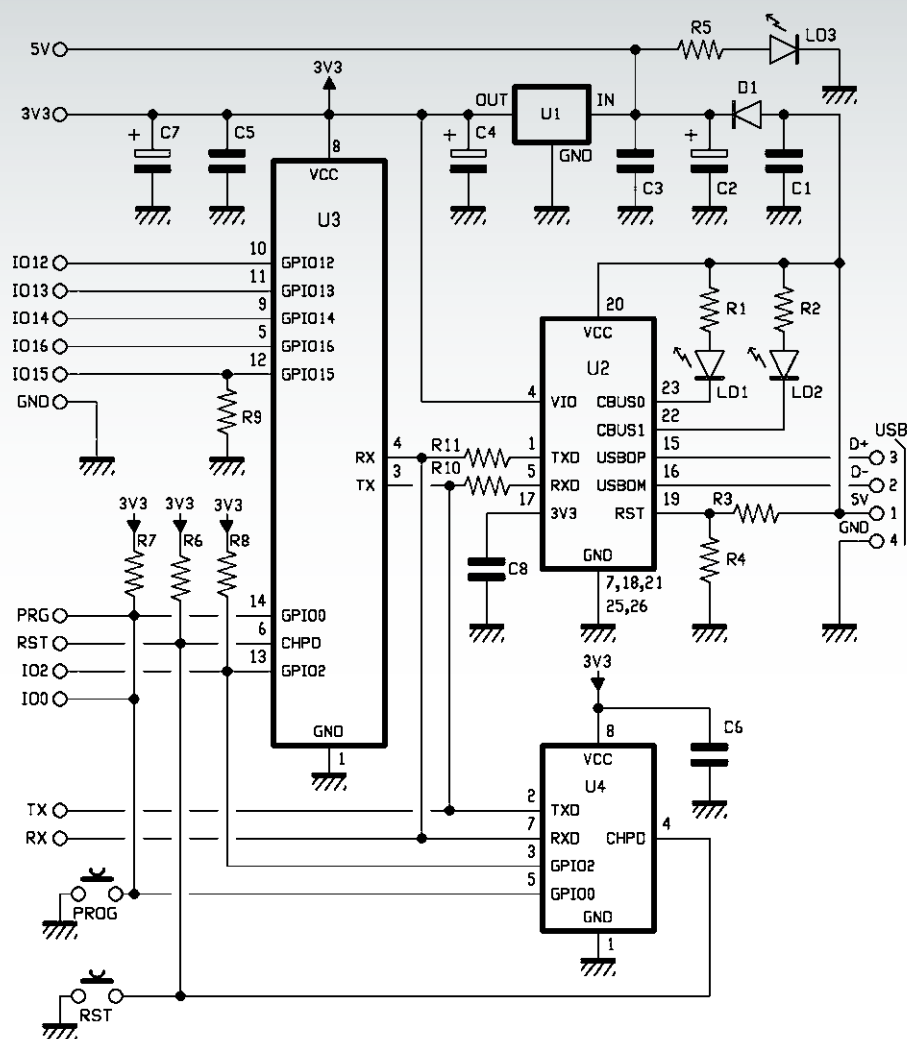
un convertitore USB/Seriale che consente di scaricare all'interno della memoria flash del modulo ESP8266 il programma che dovrà eseguire.

SCHEMA ELETTRICO

La demoboard che proponiamo in questo articolo contiene l'essenziale per creare semplici applicazioni, ovvero il convertitore USB/seriale e pochi altri componenti necessari al funzionamento. Tra questi troviamo uno stabilizzatore di tensione, alcuni connettori strip e dei pulsanti che semplificano la gestione del modulo.

Dall'analisi dello schema è possibile notare che un pulsante (PROG) è connesso al pin GPIO0 del chip ESP8266 ed è utilizzato per attivare la funzionalità di boot mode del modulo, mentre il pulsante RST, connesso al pin CH_PD (Power Down), è utilizzato come reset hardware del modulo. Gli strip, semplicemente rendono disponibili i pin del chip ESP8266, facilitando, mediante dei cavetti jumper,

[schema ELETTRICO]



la connessione a periferiche esterne. La scheda supporta l'utilizzo dei moduli ESP01 (Fig. 1) e ESP03 (Fig. 2); ricordiamo che, mentre il primo utilizza un connettore strip e quindi può essere facilmente sostituito, il secondo va saldato direttamente sullo stampato, il che ne complica l'eventuale sostituzione.

Per contro, l'ESP01 mette a disposizione un esiguo numero di pin. In questo articolo faremo riferimento all'utilizzo della board con il modulo ESP03. Dall'analisi della configurazione interna del chip ESP8266 (che è il cuore del modulo ESP-03) è possibile notare come esso disponga di tutte le funzionalità tipiche di un microcontrollore con architettura a 32 bit, oltre al

modulo di comunicazione UART dispone dell'interfaccia SPI ad alta velocità e dell'interfaccia I2C (Fig. 3). Tutti i pin sono configurabili come ingresso o uscita ed alcuni anche come ingressi analogici mentre le uscite analogiche possono essere ottenute tramite il segnale PWM disponibili in alcuni pin. Le caratteristiche elettriche e le prestazioni sono paragonabili a quelle del microcontrollore Atmega328 montato su Arduino Uno, il che permette di sviluppare applicazioni simili, con l'unica differenza che l'alimentazione è a 3,3 volt, contro i 5 volt usati sulla scheda Arduino.

CONFIGURARE IL MODULO

Vediamo ora come configurare

il modulo per il funzionamento in stand-alone, operazione non complicata ma che va eseguita con una certa attenzione. Le soluzioni sono essenzialmente due, sviluppate in collaborazione dalla comunità, che si è interessata al modulo e che ha sfruttato l'SDK fornito dall'azienda produttrice per compilare un nuovo firmware. La prima soluzione è il progetto chiamato NodeMCU il cui sito di riferimento (in lingua cinese ma con opzione in inglese) è http://nodemcu.com/index_cn.html. I sorgenti del progetto e tutti i file attinenti sono disponibili presso il sito: <https://github.com/nodemcu/nodemcu-firmware> e <https://github.com/nodemcu/nodemcu-flasher>. Per prima cosa è necessario installare il firmware in grado di supportare il funzionamento in stand-alone e per fare questo utilizziamo un apposito software di nome *ESP8266Flasher.exe* disponibile per sistemi Windows a 32 e 64 bit (Fig. 4).

Assicuratevi che il modulo sia connesso al PC, avviate il software e impostate un Baudrate di 9600 (ma anche Baudrate più elevati sono consentiti), una dimensione della Flash di 512 kByte, una velocità di 40MHz e selezionate come SPI il modo QIO (Fig. 5). Il firmware da noi usato è contenuto nel file *nodemcu_integer_0.9.5_20150318.bin* quindi apritelo dall'apposito menu (Fig. 6). Nella demoboard, premete e tenete premuto il pulsante PRG, quindi agite per un attimo sul pulsante RST per avviare il modulo in modalità di boot mode, quindi cliccate sul pulsante Flash del software per avviare la scrittura del firmware; dalla scheda LOG del software potrete osservare le operazioni eseguite.

Appena avviato il download potete rilasciare il pulsante PROG. Se tutto è andato a buon fine com-

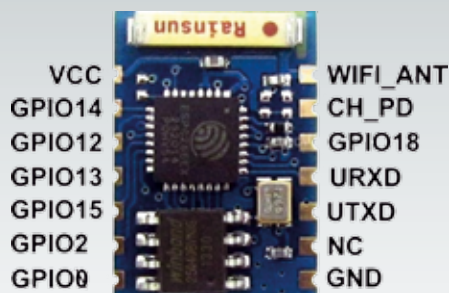


Fig. 1 - Piedinatura del modulo ESP03.



Fig. 2 - Piedinatura del modulo ESP01.

parirà una piccola icona verde in basso a sinistra. A questo punto è possibile interagire con il modulo tramite l'utilizzo di opportuni ambienti di programmazione, il più semplice ed immediato dei quali si chiama *ESP8266LuaLoader* ed è disponibile in Internet a questo indirizzo: <http://benlo.com/esp8266/>; il file da avviare si chiama *LuaLoader.exe* e si trova su <http://benlo.com/esp8266/index.html#LuaLoader>.

Non appena avviato questo IDE e stabilita la comunicazione con il modulo (dal menu *Settings*), sarete in grado di lavorare con esso in modo semplice ed immediato. Ora connettete un LED con in serie una resistenza da 220 ohm tra il pin GPIO2 e GND e provate a gestirla: utilizzate i comandi del riquadro azzurro per impostare il pin come uscita (pulsante *Set*) ed accendere e spegnere il LED con i pulsanti 0 e 1.

Usate i comandi del riquadro arancione per impostare la rete WiFi; il pulsante *Survey* permette di conoscere gli AP (Access Point WiFi) disponibili: inserendo il nome della rete e la password è possibile accedervi. Il pulsante *GEP IP* mostrerà l'indirizzo IP assegnato dal router al modulo

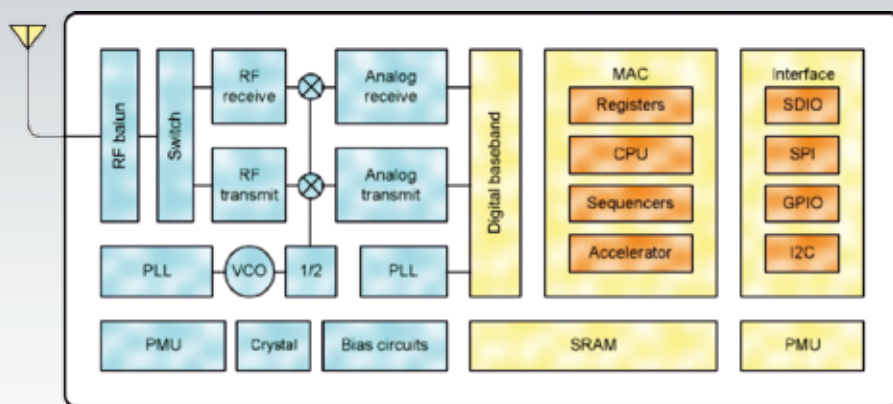


Fig. 3 - Architettura interna del chip ESP8266.

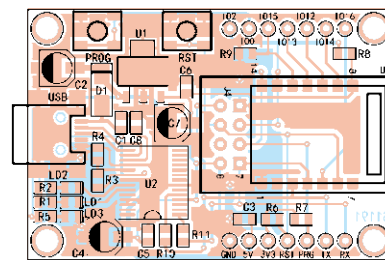
(Fig. 7). Questi primi esempi permettono di testare il funzionamento del modulo e di tutto l'hardware ma ancora il modulo non è del tutto autonomo. Infatti al riavvio il modulo è impostato per avviare lo sketch di nome *init.lua* che conterrà tutte le istruzioni necessarie a svolgere il compito previsto. Vediamo come fare quindi a programmare il modulo in maniera permanente. Partiamo con un esempio: aprite con un normale editor di testo, come il Blocco Note, il file

WebServer_toggle_pin.lua che trovate nella cartella esempio del programma *LuaLoader*, modificate le voci SSID e PASS della riga *wifi.sta.config("SSID","PASS")* con i dati della vostra rete, rispettivamente il nome e la password di accesso. Nel riquadro arancione selezionate il file che volete avviare (nel nostro esempio *WebServer_toggle_pin.lua*) e cliccate sul comando *do file* per far eseguire le istruzioni contenute (Fig. 8). Aprite il browser ed inserite l'indirizzo IP che vedete sulla scher-

[piano di MONTAGGIO]

Elenco Componenti:

- R1, R2, R5: 470 ohm (0805)
- R3: 4,7 kohm (0805)
- R4: 10 kohm (0805)
- R6÷R9: 4,7 kohm (0805)
- R10, R11: 33 ohm (0805)
- C1: 100 nF ceramico (0805)
- C2: 22 µF 6,3 VL elettrolitico (Ø4mm)
- C3, C5, C6, C8: 100 nF ceramico (0805)
- C4, C7: 22 µF 6,3 VL elettrolitico (Ø4mm)
- C8: 100 nF ceramico (0805)
- U1: TC1262-3.3VDB
- U2: FT232RL
- U3: ESP03
- U3: ESP01
- RST: Microswitch > Farnell: 2320079
- PROG: Microswitch > Farnell: 2320079
- D1: GF1M
- LD1: LED giallo (0805)
- LD2: LED rosso (0805)
- LD3: LED verde (0805)



- Varie:
- Connettore mini USB
 - Circuito stampato S1191

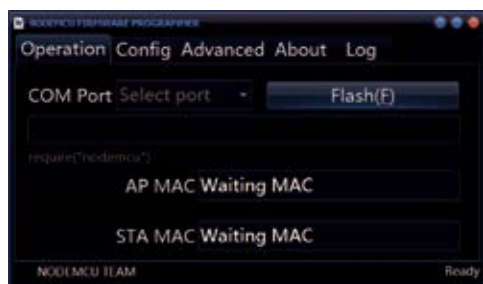


Fig. 4 - Il software usato per caricare il firmware.



Fig. 5 - Impostazioni di programmazione.

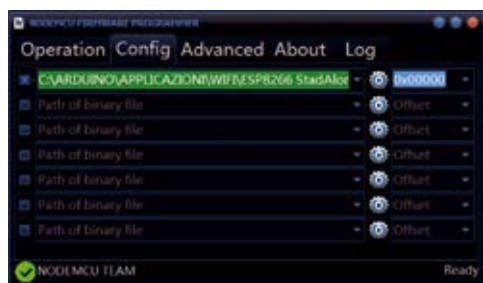


Fig. 6 - Apertura del file .bin contenente il firmware.

matà, avrete così la possibilità di accendere e spegnere da remoto il LED connesso a GPIO2 (Fig. 9). Come già detto, il modulo al riavvio è impostato per avviare lo sketch di nome *init.lua* che conterrà tutte le istruzioni necessarie al compito da svolgere, pertanto quando avete deciso il file da usare, dovreste rinominarlo con il nome *init.lua* e caricarlo sul modulo: ad ogni avvio verrà avviato questo sketch ed eseguite le istruzioni in esso contenute. Non è più necessario che il modulo sia connesso al PC, pertanto è sufficiente alimentare la scheda con una tensione di 5V tramite la porta USB ed il gioco è fatto. Per tale funzione possono risultare utili le batterie per la ricarica di

smartphone, delle quali esistono anche delle versioni avanzate che permettono la ricarica tramite energia solare.

Con questi prodotti è possibile alimentare la scheda praticamente per un periodo infinito, purché sia assicurata sufficiente insolazione per la ricarica della batteria. Ovviamente anche un semplice alimentatore stabilizzato a 5V va benissimo, come quelli usati come caricabatteria per smartphone. Vediamo ora altri esempi, utilizzando un software di gestione alternativo: si tratta dell'applicazione chiamata *ESPlorer* scaricabile all'indirizzo <https://github.com/4refr0nt/ESPlorer/>.

Anche questo, come gli altri programmi, non necessita di installa-

zione ma richiede che sul PC sia installato l'ambiente java; il file da avviare si chiama *ESPlorer.jar*.

Prendete il file di esempio di nome *WebServer_toggle_pins.lua* e rinominatelo in *init.lua*, quindi apritelo dal menu

File>Open from disk e caricatelo sul modulo con il comando *Save to ESP* (Fig. 10), poi riavviate il modulo con il comando *Reset ESP*, oppure lanciate il file con il comando *DoFile*; sulla schermata comparirà l'indirizzo IP assegnato dal router al modulo: inserite questo indirizzo nel browser per verificare il funzionamento del webserver (Fig. 11).

Analizziamo il listato per meglio comprenderne il funzionamento. Le prime righe sono utilizzate

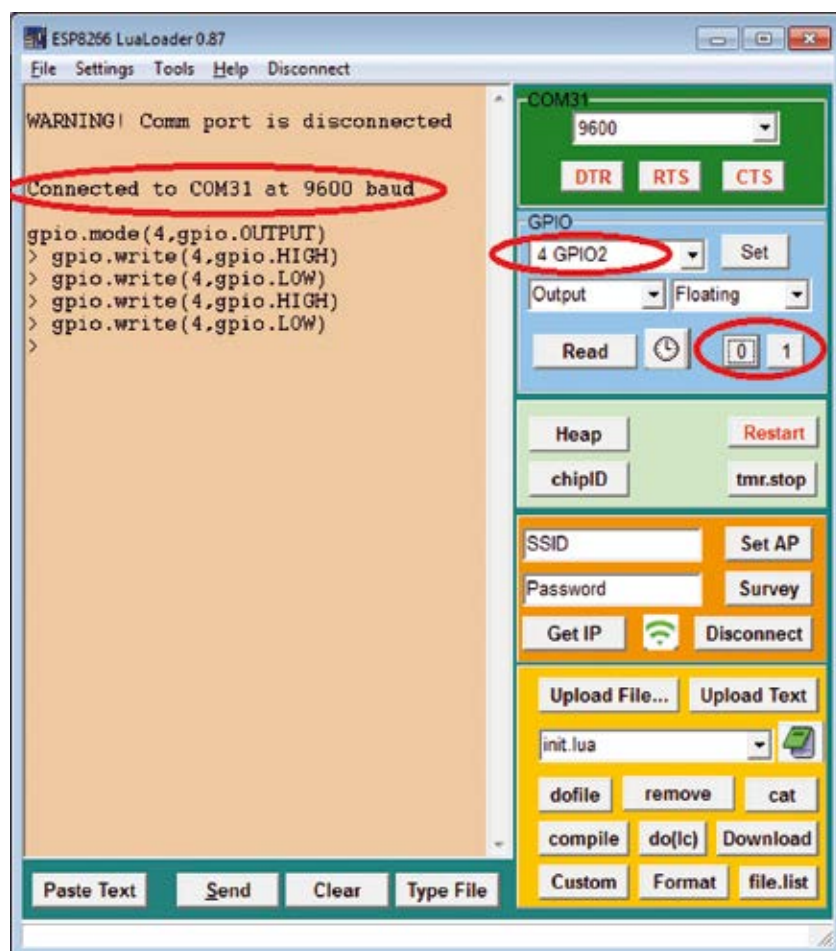


Fig. 7 - LuaLoader: gestione GPIO.

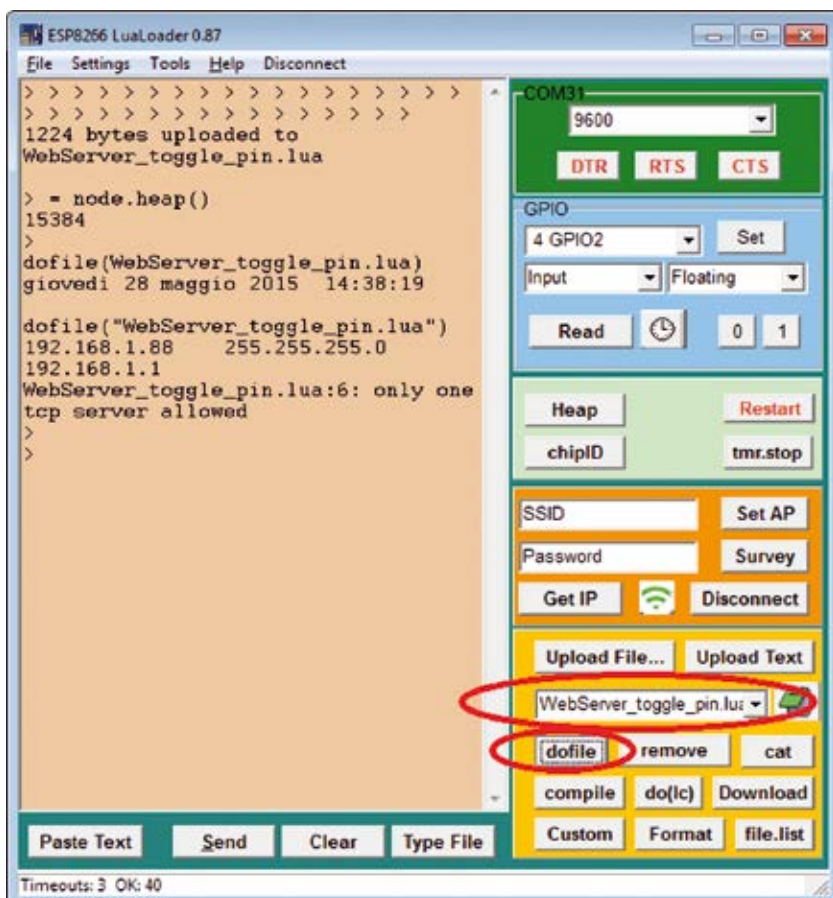


Fig. 8 - Gestione file in LuaLoader.

per la connessione alla rete WiFi (in questo caso al router di casa) e, come abbiamo visto, dovranno essere modificate aggiungendo il nome della rete e la password di accesso; lo sketch provvederà alla stampa dell'indirizzo IP assegnato dal router:

```
wifi.setmode(wifi.STATION)
wifi.sta.
config("SSID","PASSWORD")
print(wifi.sta.getip())
```

Le successive righe servono ad impostare i pin GPIO come uscite e per avviare il server mettendolo in ascolto sulla porta 80, predefinita per le richieste html:

```
led1 = 3
led2 = 4
gpio.mode(led1, gpio.OUTPUT)
gpio.mode(led2, gpio.OUTPUT)
srv=net.createServer(net.
TCP)
```

```
srv:listen(80,function(conn)
```

Quando arriva una richiesta dal client (il browser) essa verrà elaborata dalla funzione:

```
conn:on("receive",
function(client,request)
```

che provvederà a verificare se la richiesta è compatibile con lo standard html quindi provvederà a costruire la stringa di risposta, ovvero la pagina che sarà inviata al client per la sua visualizzazione. Dalla richiesta del client viene anche estrapolata la stringa relativa al comando (valore ON o OFF) usato per attivare o disattivare i LED; il codice corrispondente è quello di seguito:

```
if(_GET.pin == "ON1")then
  gpio.write(led1, gpio.
HIGH);
elseif(_GET.pin == "OFF1")
```

```
then
  gpio.write(led1, gpio.LOW);
elseif(_GET.pin == "ON2")
then
  gpio.write(led2, gpio.
HIGH);
elseif(_GET.pin == "OFF2")
then
  gpio.write(led2, gpio.LOW);
end
```

La programmazione non è forse immediata ma prendendo spunto dai numerosi esempi forniti e con alcune prove si potranno ottenere tutte le funzioni desiderate. Ad esempio un altro interessante sketch che consente di utilizzare il modulo come client web per richiedere data e ora ai server di Google, il file si chiama *googleTime.lua* (Fig. 12).

In questo caso lo sketch inizia creando la connessione al server di Google con la riga:

```
conn=net.
createConnection(net.TCP, 0)
conn:connect(80,'google.
com')
```

Non appena è avvenuta la connessione (*conn:on*) viene inviata la richiesta HTTP verso il server (Listato 1).

La funzione successiva si occupa di elaborare la stringa di ritorno ed estrapolare i dati richiesti, per poi visualizzarli a video (Listato 2). Come finezza viene anche riportato il tempo necessario a completare la richiesta.



Fig. 9 - WebServer dell'esempio WebServer_toggle_pin.lua.

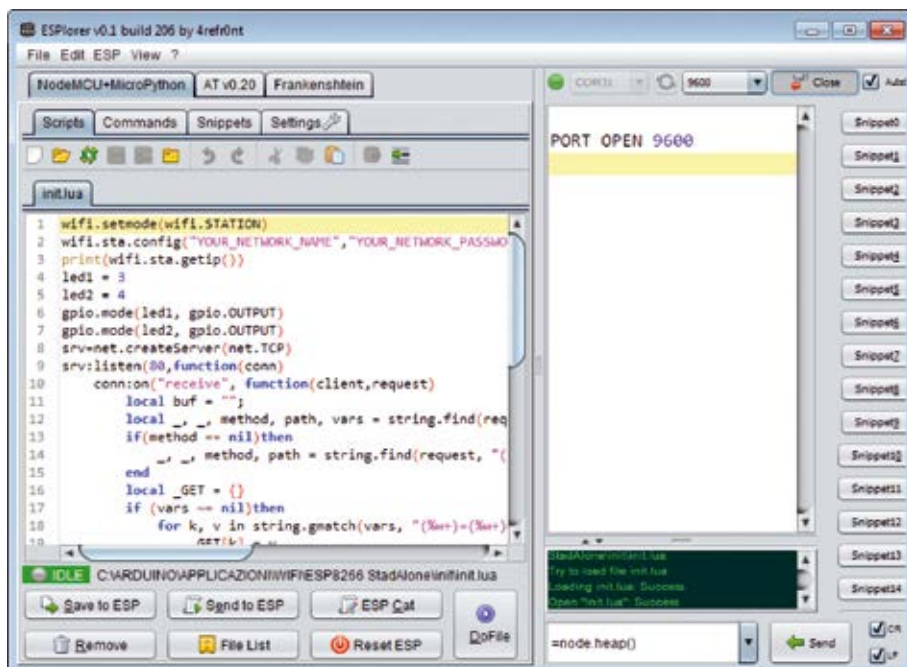


Fig. 10 - ESPlorer: apertura file di esempio.

lare l'ultima versione dell'IDE di Arduino (al momento in cui scriviamo la 1.6.5); nelle release recenti l'IDE, oltre ad essere multi-piattaforma è diventato anche multi board, infatti viene usato per programmare diversi tipi di schede semplicemente installando dei pacchetti aggiuntivi con la procedura descritta qui di seguito. Aprite l'IDE e selezionate il menu *File>Impostazioni*, quindi nel campo *Additional board manager URLs* della finestra che vi appare inserite l'indirizzo http://arduino.esp8266.com/package_esp8266com_index.json. Ora, nella finestra accessibile con il comando di menu



Fig. 11 - WebServer dell'esempio WebServer_toggle_pins.lua.

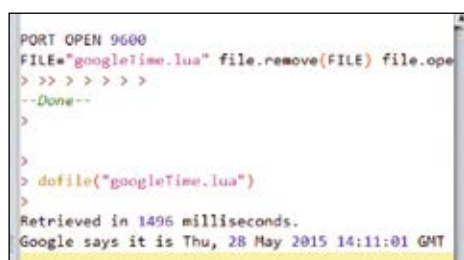


Fig. 12 - WebClient Time dal server Google.

LAVORARE CON L'IDE ARDUINO

Come avete visto, pur non essendovi grosse difficoltà, la programmazione non è proprio della massima semplicità; per questo motivo la comunità si è mossa per rendere l'IDE di Arduino compatibile con questo modulo (in fin dei conti il microcontrollore a bordo non è tanto diverso da quello installato in una scheda Arduino). Per prima cosa occorre instal-

Listato 1

```
conn:on("connection",function(conn, payload)
  conn:send("HEAD / HTTP/1.1\r\n"..
    "Host: google.com\r\n"..
    "Accept: */*\r\n"..
    "User-Agent: Mozilla/4.0 (compatible; esp8266 Lua;)"..
    "\r\n\r\n")
end)
```

Listato 2

```
conn:on("receive", function(conn, payload)
  print('\nRetrieved in '..((tmr.now()-t)/1000).. ' milliseconds.')
  print('Google says it is '..string.sub(payload,string.find(payload,"Date: ")
    +6,string.find(payload,"Date:")+35))
  conn:close()
end)
```

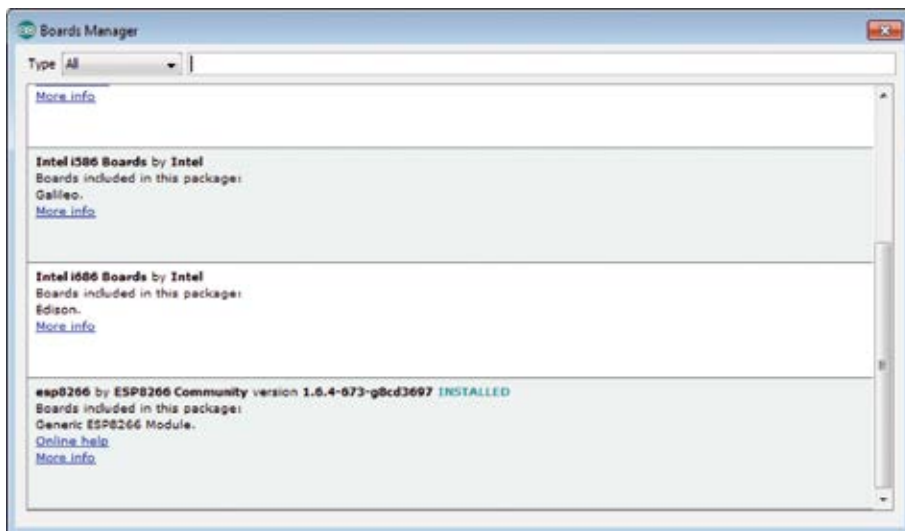


Fig. 13 - Installazione del pacchetto per ESP8266.

Strumenti>Scheda>Board manager, installate il pacchetto *esp8266 by ESP8266 Community* (Fig.13). Il supporto per il chip ESP8266 è adesso disponibile.

Se avete incontrato dei problemi in questa procedura potete visionare il sito Internet di riferimento all'indirizzo <https://github.com/esp8266/arduino>; se tutto è andato bene, invece, riavviate l'IDE e selezionate *Strumenti>Scheda>Generic ESP8266 Module* che ora è presente; assicuratevi che le impostazioni siano le stesse mostrate in Fig. 14 (Generic ESP8266 Module, Serial, 80 MHz, 40 MHz, QIO, 512000, 512K) e aprite l'esempio blink relativo al lampeggio di un LED (si apre dal menu *File>Esempi>01.Basic*), modificate il pin utilizzato (13) con il nostro GPIO2 (2) al quale avrete connesso un LED con una resistenza in serie. A questo punto dovete trasferire il programma nel modulo ESP, che quindi dobbiamo portare in boot mode con la solita procedura: mantenete premuto il pulsante PROG (GPIO0) e premete per un attimo RST (CH_PD). Cliccate sul pulsante *Carica* (quello con la freccia) come fareste per caricare uno sketch su Arduino, ed attendete il completamento delle operazioni; non appena avviata la procedura di download potete rilasciare il pulsante PROG. Come avrete intuito non abbiamo usato la funzione di autoreset che viene svolta manualmente quando portiamo il modulo in boot mode. Se avete problemi durante il caricamento potete ridurre la velocità da 512000 a 115200 Baud. A caricamento terminato, premete il pulsante di RST per riavviare il modulo e far partire lo sketch; il LED inizierà a lampeggiare. Se tutto ha funzionato potete subito sperimentare altre applicazioni, la più classica delle quali è il web

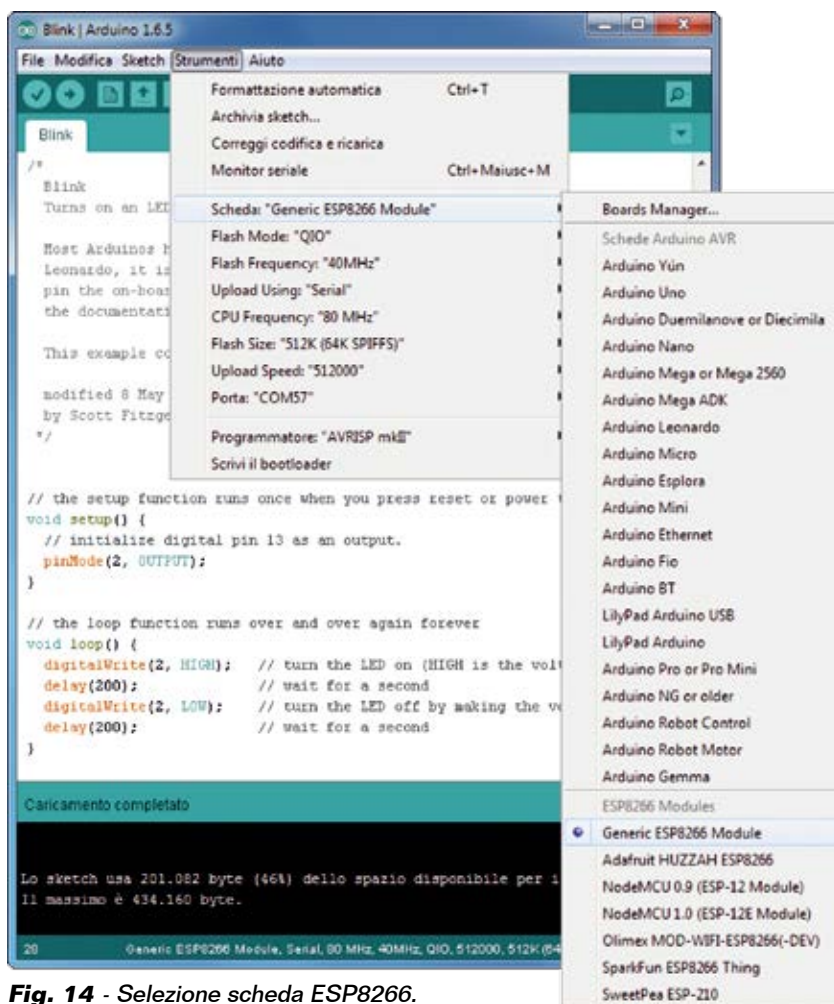


Fig. 14 - Selezione scheda ESP8266.

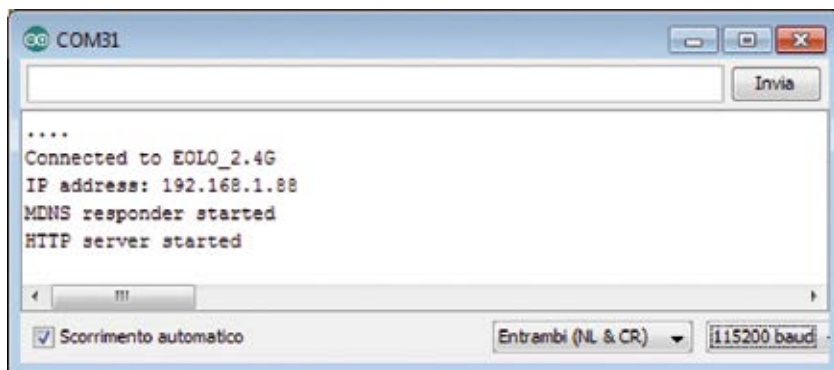


Fig. 15 - Indicazione della ESP8266 board con LED su GPIO2 lampeggiante.

server con lo sketch *HelloServer*, disponibile tra gli esempi del modulo ESP8266. Per caricare ed avviare lo sketch seguite la procedura appena descritta, con una variante: prima di resettare il modulo aprite Serial Monitor ed impostate la velocità di comunicazione a 115200 Baud; ciò vi permetterà di vedere l'indirizzo

IP assegnato dal vostro router al modulo. Aprite il browser ed inserite l'indirizzo appena visualizzato per vedere il web server in funzione. Gli esempi disponibili sono veramente numerosi e comprendono praticamente ogni applicazione significativa. Abbiamo comunque deciso di proporre una nostra applicazione



Arriva il WiFi per auto

Nasce in America in "casa" Autonet Mobile (www.autonetmobile.com) e consente di avere in auto una connessione Wi-Fi per l'accesso ai servizi multimediali di bordo, ma anche per accedere a Internet attraverso un router che si appoggia alle reti cellulari a banda larga 3G e 2,5G. Il router integra un Hot Spot WiFi, che permette di collegare smartphone, tablet e PC portatili, la cui copertura è notevole: circa 30 metri, il che significa potervi accedere anche ad auto ferma quando si è all'esterno. Per evitare intrusioni nel WiFi, il router

implementa la crittografia WEP. Autonet Mobile ha offerto al grande pubblico la sua tecnologia in collaborazione con la FIAT-Chrysler, fornendo a quest'ultima il sistema uConnect, che è disponibile anche sulle vetture (ad esempio la Giulietta) vendute in Italia; sta anche stringendo accordi con altre case automobilistiche come VAG, che monta uConnect sulle auto vendute in USA.



personalizzata che consiste in una semplice lampada di cui si può controllare da smartphone accensione, spegnimento e variazione di colore. Per prima cosa dobbiamo connettere un LED RGB alle uscite PWM del modulo ESP03: per la precisione i pin 12, 13, 14, i quali comanderanno rispettivamente i tre colori primari rosso, verde e blu. Ciascun LED necessiterà di una resistenza in serie, tale da garantire una corrente di circa 10 mA.

La formula generale per il calcolo della resistenza è $R = (V_{cc} - V_F) / 0,01$ dove V_{cc} è la tensione di alimentazione (nel nostro caso 3,3 V)

mentre V_F è la caduta di tensione sul diodo. Nel caso del LED rosso, la V_F vale circa 1,7 volt per cui la resistenza serie vale circa 150 ohm; per i diodi verde e blu la caduta di tensione è di 3 volt, quindi la resistenza è di circa 33 ohm. Preparato l'hardware, occupiamoci della parte software, ovvero dello sketch da caricare sul modulo; visto il tipo di applicazione, pare naturale implementare una comunicazione di tipo UDP (User Datagram Protocol) che, pur non garantendo la ricezione corretta dei dati (non implementa la conferma di ricezione né il re-invio dei dati corrotti), ha il vantaggio

di non soffrire di tempi di latenza e prevede la possibilità dell'invio dei dati da un server a più client contemporaneamente, senza la necessità di stabilire una comunicazione, ovvero senza che siano inseriti gli indirizzi IP.

È proprio quest'ultima peculiarità che interessa nel nostro progetto, perché se dovessimo installare più lampade nella stessa stanza, tutte sarebbero controllate all'unisono con la stessa applicazione; lo stesso ponendole in più stanze, purché ogni lampada abbia accesso alla stessa rete WiFi.

Ciascuna demoboard dovrà essere alimentata autonomamente e connessa alla rete WiFi domestica in attesa di messaggi UDP che saranno i comandi per gestire il LED RGB. Lo sketch per ESP8266 è facilmente comprensibile (il listato completo lo potete scaricare dal sito elettronica.in.it); possiamo analizzarne le righe essenziali: per la comunicazione UDP abbiamo scelto la porta tuttofare 8080 e questo è l'unico parametro necessario affinché solo comunicazioni indirizzate alla stessa porta possano funzionare.

L'indirizzo 0.0.0.0 è invece un indirizzo generico, ad indicare una trasmissione verso tutti:

```
unsigned int localPort = 8080;
```



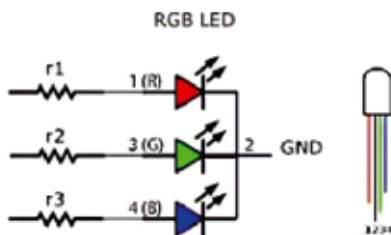


Fig. 16 - Pinout di un LED RGB a catodo comune.

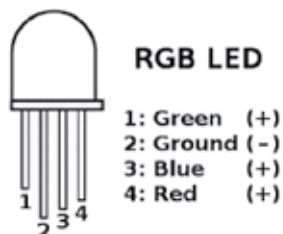


Fig. 17 - Pinout di un altro tipo di LED RGB a catodo comune.

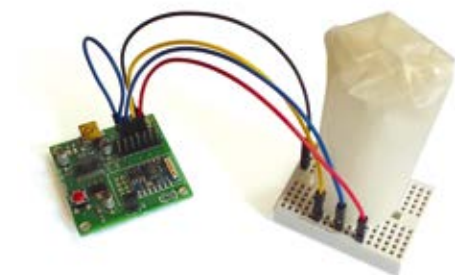


Fig. 18 - Circuito di test dell'esempio RGBlamp: la copertura bianca serve ad "amalgamare" le luci rossa, verde e blu del LED ottenendo una colorazione omogenea.

```
IPAddress serverIP(0,0,0,0);
// Broadcast communication
```

La parte del programma relativa al setup provvede all'accensione a luce rossa del LED non appena la scheda viene alimentata; segue il tentativo di connettersi alla rete WiFi. Prima di caricare il programma è necessario impostare nome utente e password per l'accesso alla rete domestica di casa. Come al solito, lo sketch va personalizzato inserendo l'SSID e la password della vostra rete. Lo sketch provvede la stampa dell'indirizzo IP assegnato da router (anche se questa informazione non è

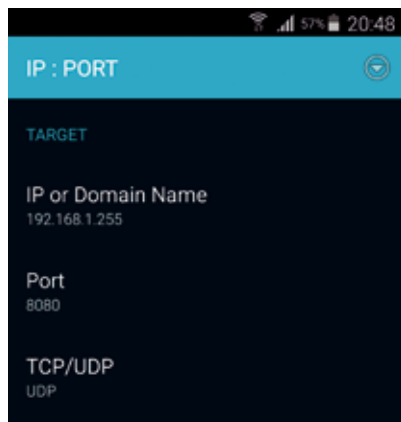


Fig. 19 - Schermata delle impostazioni dell'app TCP UDP RGB.

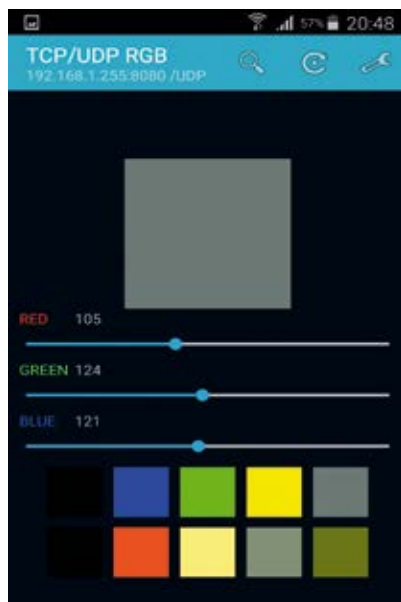


Fig. 20 - Schermata principale dell'app TCP UDP RGB.

necessaria); segue poi l'avvio del client UDP. Abbiamo previsto che, appena alimentata, la scheda faccia accendere la lampada a luce rossa e, una volta terminata con successo la connessione alla rete, emetta un lampeggio a luce verde per due secondi (**Listato 3**). Il programma vero e proprio è contenuto nella funzione loop e provvede semplicemente a leggere i dati ricevuti e a processarli (**Listato 4**). La variabile *packetSize* assume valore maggiore di zero quando sono arrivati dei messaggi; nel nostro caso questi ultimi saranno semplicemente tre byte conte-

Listato 3

```
void setup()
{
  setColor(255,0,0);
  Serial.println();
  Serial.begin(9600);

  // We start by connecting to a WiFi network
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, pass);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println();
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

  Serial.println("Starting UDP");
  udp.begin(localPort);
  Serial.print("Local port: ");
  Serial.println(udp.localPort());
  setColor(0,255,0);
  delay(2000);
  setColor(0,0,0);
}
```

Listato 4

```
void loop()
{
  int packetSize = udp.parsePacket();
  if (packetSize)
  {
    int len = udp.read(packetBuffer, 255);
    if (len > 0) packetBuffer[len] = 0;
    int R = packetBuffer[0];
    int G = packetBuffer[1];
    int B = packetBuffer[2];
    setColor(R,G,B);
  }
  delay(100);
}
```

nenti ciascuno i valori RGB della luminosità.

Questi tre byte vengono letti e convertiti in numeri interi con valore compreso tra 0 e 255 e quindi utilizzati per impostare la luminosità dei tre LED richiamando la funzione *setColor*.

La funzione *setColor* utilizza i tre valori per impostare i tre canali PWM tramite la nota funzione *analogWrite* sui tre pin assegnati. Per inviare il comando per la gestione del LED RGB abbiamo pensato di usare una specifica app disponibile per dispositivi Android, chiamata TCP UDP RGB e reperibile sul web all'indirizzo <https://>

12^a FIERA REGIONALE DELL'ELETTRONICA SCANDIANO

(REGGIO EMILIA)

24/25 OTTOBRE
CENTRO FIERISTICO 2015



**MERCATINO DELLE PULCI
RADIOAMATORIALI**

**TELEFONIA - COMPONENTISTICA
COMPUTER - HI-FI CAR - RADIANTISMO
CB e OM - VIDEOREgistRAZIONE**

ORARI

**Sabato 24 Ottobre 9 - 18,30
Domenica 25 Ottobre 9 - 18**



**INGRESSO euro 7,50 RIDOTTO euro 5
GRATUITO per disabili e ragazzi fino a 14 anni**

**INFO: 0522/764302-857436
www.fierascandiano.it**

Guarda la lampada RGB su YouTube!



Vuoi vedere come funziona la lampada a LED RGB gestita da smartphone? Nulla di più semplice: collegati alla pagina <https://youtu.be/QcWQQ60qPrU>. Potrai anche vedere i video di moltissimi altri nostri progetti proposti in passato.



play.google.com/store/apps/details?id=nextprototypes.tcpudpRGB. Si tratta di un'app molto semplice ma estremamente efficace e facilissima da usare; le uniche impostazioni che richiede sono l'indirizzo IP e la porta di comunicazione. Per la nostra rete domestica, l'indirizzo IP da assegnare vale 192.168.1.255 e porta 8080, ovvero la trasmissione avverrà sulla porta 8080 di tutti i dispositivi il cui indirizzo IP inizia con 192.168.1.xxx (Fig. 19). Dalla schermata principale (Fig. 20) è possibile inviare i comandi di gestione del LED RGB scegliendo separatamente i valori di rosso, verde e blu, con la possibilità di salvare sino a 10 valori personalizzati. Per ottenere un miglior risultato, dal punto di vista luminoso, consigliamo di racchiudere il LED RGB con della carta semitrasparente che svolge la funzione di diffusore. Amplificando il segnale in uscita dal modulo ESP con dei transistor, sarebbe possibile utilizzare un LED RGB di potenza, così da ottenere un flusso luminoso sufficiente per creare una bella luce diffusa personalizzabile nella stanza.



per il MATERIALE

La Demoboard con ESP03 (Cod. FT1191M) è in vendita presso Futura Elettronica. Viene fornita montata, collaudata e comprensiva del modulo Wi-Fi al prezzo di 23,00 Euro. Il prezzo si intende IVA compresa.

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



DOMOTICA OPEN SOURCE con i moduli UDOiT

Facciamo conoscenza con il telecomando e l'attuatore relè della soluzione di domotica Open Source di UDOiT basata sui moduli RFTide ed introduciamo l'utilizzo delle "Scene".

di MARCO MAGAGNIN

Una delle principali funzioni che ci si aspetta da un sistema di automazione domestica è la possibilità di attivare delle "utenze", siano esse semplici lampade, elettrodomestici od ulteriori sistemi di gestione dell'abitazione, come antifurti, controllo accessi, luci notturne, irrigazione del giardino e dell'orto. La collezione UDOiT prevede una serie di attuatori, alcuni disponibili ed altri in via di sviluppo oltre ad un modulo telecomando che permette di svincolare alcuni comandi dall'interfaccia web.

ATTUATORE RELÉ

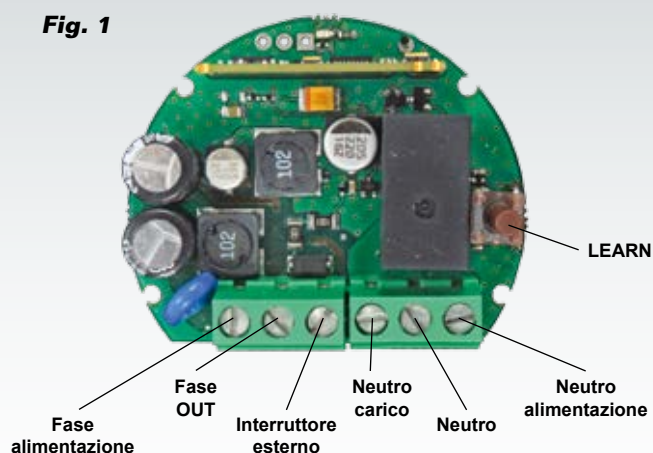
Il primo attuatore disponibile è il dispositivo "Relè", che permette di attivare carichi fino a 10 A con tensione fino a 220 V. A questo proposito è bene precisare che quando si decide di installare un impianto

domotico la parte in bassa tensione non è soggetta a certificazione mentre lo è tutta la parte di impianto che in qualche modo si interfaccia alla rete a 220 V, che pertanto deve essere installata e certificata da un installatore autorizzato.

Torniamo al dispositivo relè, visibile nella foto di **Fig. 1**, che dispone di un contenitore esterno disegnato in modo da poter essere installato in una normale cassetta da incasso per impianti elettrici di tipo 503E. A tale scopo per questo dispositivo è stato realizzato un modulo a radiofrequenza miniaturizzato, e tutta la realizzazione è anch'essa particolarmente compatta.

Sempre nella **Fig. 1** sono visibili le posizioni dei connettori e del tasto di learning presenti sul dispositivo, in particolare:

Fig. 1



- Un microinterruttore da utilizzare per il processo di "Learning";
- Un connettore con morsetti a vite che riportano le uscite del relè;
- Un connettore con morsetti a vite per alimentare il modulo dalla rete a 220 V;
- Un connettore con morsetti a vite per collegare un pulsante di comando esterno (questo connettore è stato predisposto per usi futuri, al momento il firmware non implementa questa funzionalità).

Il firmware del modulo implementa i comandi di gestione del dispositivo che utilizzano il protocollo RFTide standard e quindi il dispositivo relè si inserisce perfettamente nell'architettura UDOiT che vi abbiamo presentato negli articoli precedenti. Sono previste diverse possibilità di comando e gestione del modulo. Il comando di attivazione o disattivazione del relè, prevede un pacchetto di risposta con lo stato del relè. Il pacchetto di comando prevede la configurazione visibile in Fig. 2

All'invio di un pacchetto di attivazione o disattivazione del relè fa seguito un pacchetto di risposta da parte del modulo con il contenuto visibile in Fig. 3. Nel caso il relè venga attivato mediante l'interruttore esterno, che funziona in modo bistabile, ovvero con l'interruttore collegato alla fase il relè è attivato mentre lasciato "floating" il relè è disattivato, l'even-

Comando	Indirizzo	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x00	0x00	Indirizzo in esadecimale	0x30	0x02	On/Off 0x01/0x00	0x00	0x00	0x00	0x00

Fig. 2

Comando	Indirizzo	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x00	0x00	Indirizzo in esadecimale	0x30	0x01	On/Off 0x01/0x00	0x00	0x00	0x00	Vers. Firmware

Fig. 3

to viene comunicato alla rete RFTide con l'invio di un pacchetto nello stesso formato del pacchetto di risposta visibile in Fig. 3, che riporta lo stato finale del dispositivo gestito dall'applicazione di controllo ospitata dal gateway. Per aggiungere alla rete un dispositivo attuatore Relè si utilizza la funzione "Add Device" della applicazione di gestione, che si trova nella pagina "Settings", come visibile in Fig. 4. Alla pagina "Settings", come già descritto nell'articolo precedente sul numero 198 della rivista, viene richiesta un'autenticazione con userid e password.

Nella Fig. 5 si nota che il dispositivo è caratterizzato da un nome, di lunghezza massima quattro caratteri, una descrizione, che apparirà nelle pagine web di gestione dei dispositivi della rete, un indirizzo, che deve essere univoco nell'intero sistema, ed il tipo di dispositivo, nel nostro caso "Relè".

Abbiamo poi la possibilità di scegliere se vedere o meno l'entrata del dispositivo nella pagina web di gestione dell'applicazione, cosa abbastanza ovvia per un dispositivo di attuazione carichi e, in questo caso scegliere una posizione nell'elenco (campo "Order").

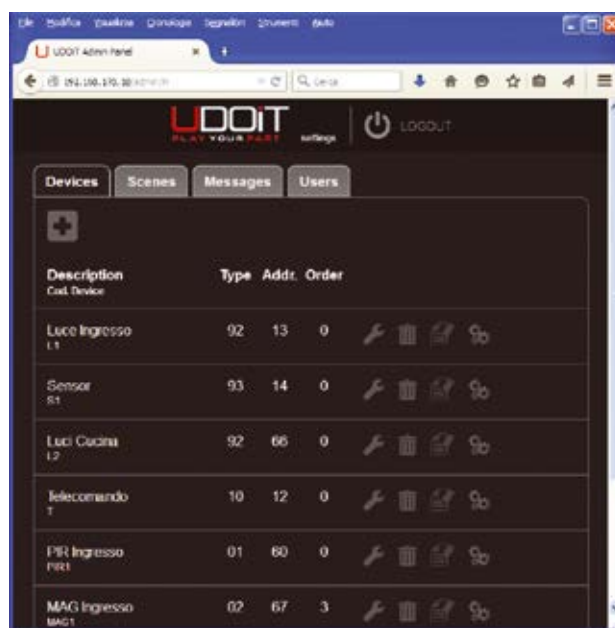


Fig. 4

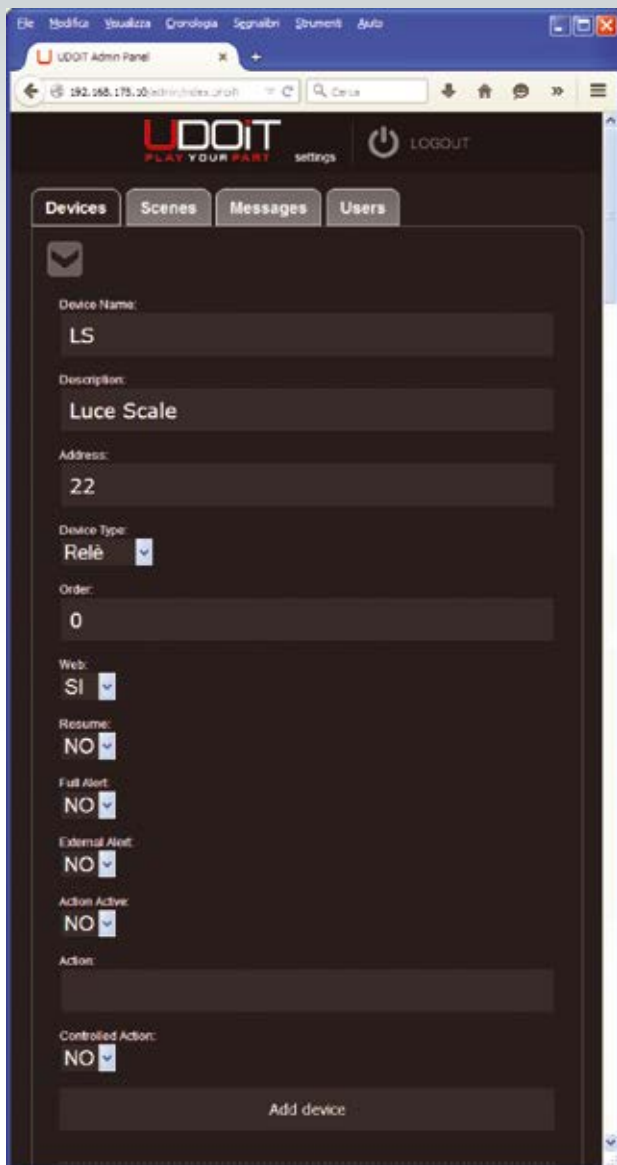


Fig. 5

I campi successivi servono per fare entrare o meno il dispositivo nella rete di controllo e sicurezza e per pilotare, in caso di attivazione, l'esecuzione di un processo personalizzato. Queste ultime funzionalità sono un po' più complesse da configurare e personalizzare e le affronteremo in seguito in un articolo dedicato. Una volta configurata l'anagrafica del dispositivo e salvata con il pulsante "Add device" possiamo vedere in Fig. 6 come si presenta sulla pagina web: un'entrata di una riga con il nome ed il pulsante per l'attivazione e disattivazione del relè. Il codice colore e la dicitura ci informano anche sullo stato attuale del relè stesso. Per cambiare lo stato del relè si "clicca" sul pulsante di stato corrispondente al dispositivo o su un dispositivo mobile si "tappa" sullo stesso. Questo però non prima di avere eseguito l'operazione di "learning" del dispositivo stesso. L'operazione di learning è necessaria per associare alla rete il dispositivo appena

descritto mediante un processo di apprendimento, durante il quale il dispositivo viene personalizzato con il Network ID di appartenenza (quello del Dongle, Master della rete) e l'indirizzo di identificazione del dispositivo stesso, una sorta di cognome e nome che permetteranno di riconoscere univocamente il dispositivo nella rete e impediranno ai dispositivi presenti su reti adiacenti di interferire.

Il processo di apprendimento avviene come segue:

- Nella pagina "Devices" del pannello "Settings" si "prenota" l'operazione di apprendimento cliccando sull'icona a forma di foglio di carta e matita ("Learn Device" in corrispondenza del dispositivo per il quale si vuole eseguire l'associazione Fig. 7):
- A questo punto si preme il pulsante presente sul circuito stampato per circa cinque secondi, al rilascio si accende un LED rosso;
- Entro 15 secondi cliccare sul pulsante "OK" della finestra "Pop-Up" sulla pagina web;
- Il LED lampeggia per qualche secondo;
- Il dispositivo è in grado di essere riconosciuto ed operare sulla rete;
- In caso non venga avviata la procedura di apprendimento il dispositivo esce dallo stato di apprendimento dopo circa cinque secondi.

Se è stata scelta l'opzione WEB "SI" lo stato del dispositivo sarà visibile nella pagina di gestione web del sistema (Fig. 6).

TELECOMANDO

Il telecomando è un dispositivo particolare, che permette di eseguire delle azioni sugli attuatori. Ma solo per il tramite del gateway e dell'applicazione di gestione, che deve essere configurata in modo opportuno. Il telecomando, come visibile in Fig. 8 dispone di quattro tasti di comando ai quali possono essere associate quattro diverse funzioni, un LED di segna-



Fig. 6

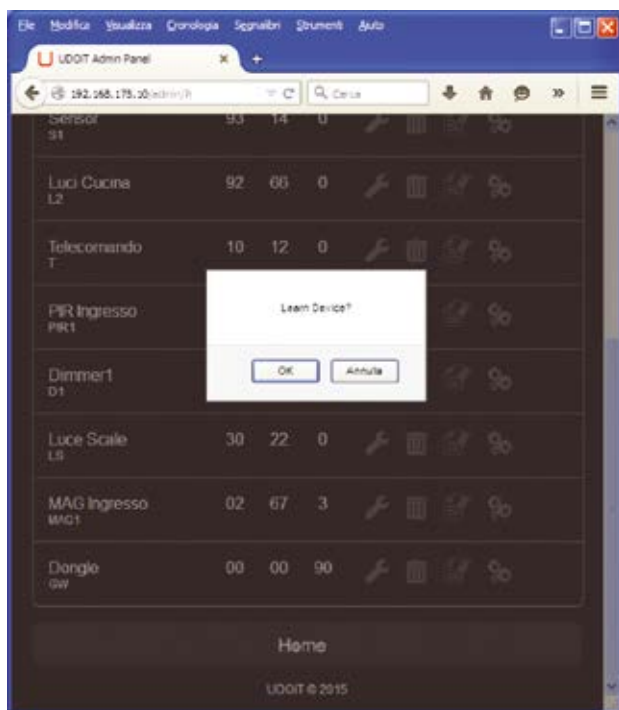


Fig. 7

zione ed è alimentato da una batteria a bottone da 3V. Lo schema di funzionamento complessivo è visibile in Fig. 9. Ovviamente anche il modulo RF del telecomando utilizza il protocollo di comunicazione RFTide con i contenuti visibili in Fig. 10.

A parte i soliti dati di riconoscimento del dispositivo, tipo di device nel byte 0 ed indirizzo nel byte indirizzo sono presenti due importanti informazioni, il numero del tasto premuto nel byte 1 e il codice univoco di identificazione nei byte 5 e 4 espressi secondo la convenzione Big-Endian. Il codice univoco di identificazione viene assegnato in fabbrica e serve a "certificare" la provenienza dei comandi inviati. Facciamo

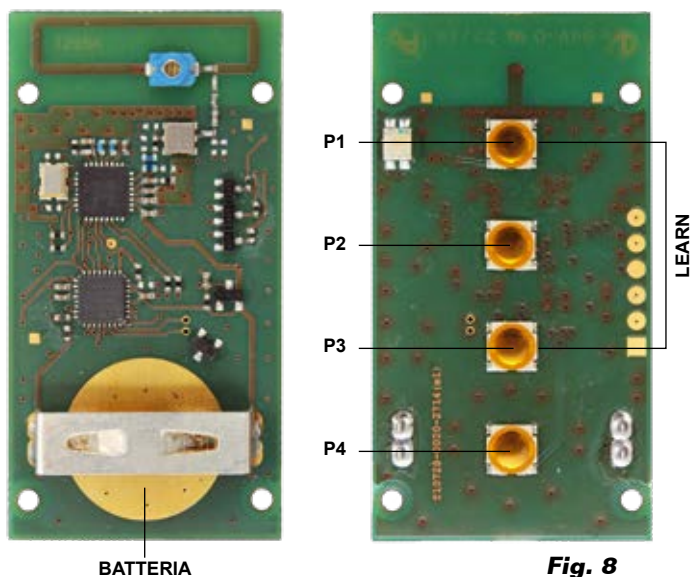
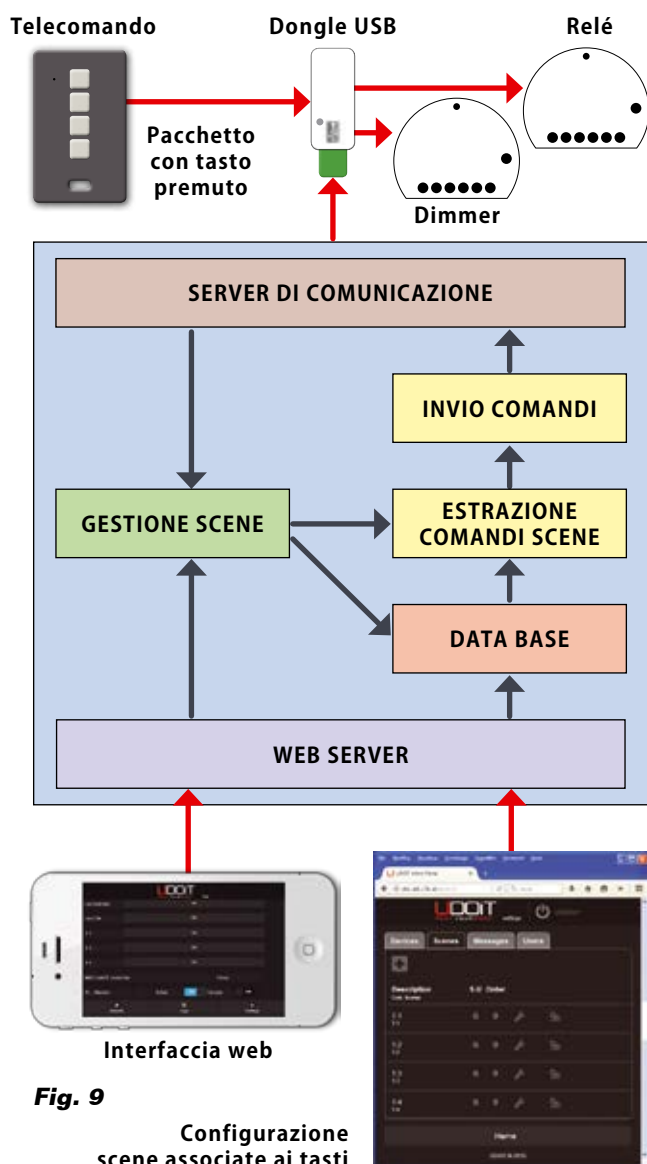


Fig. 8

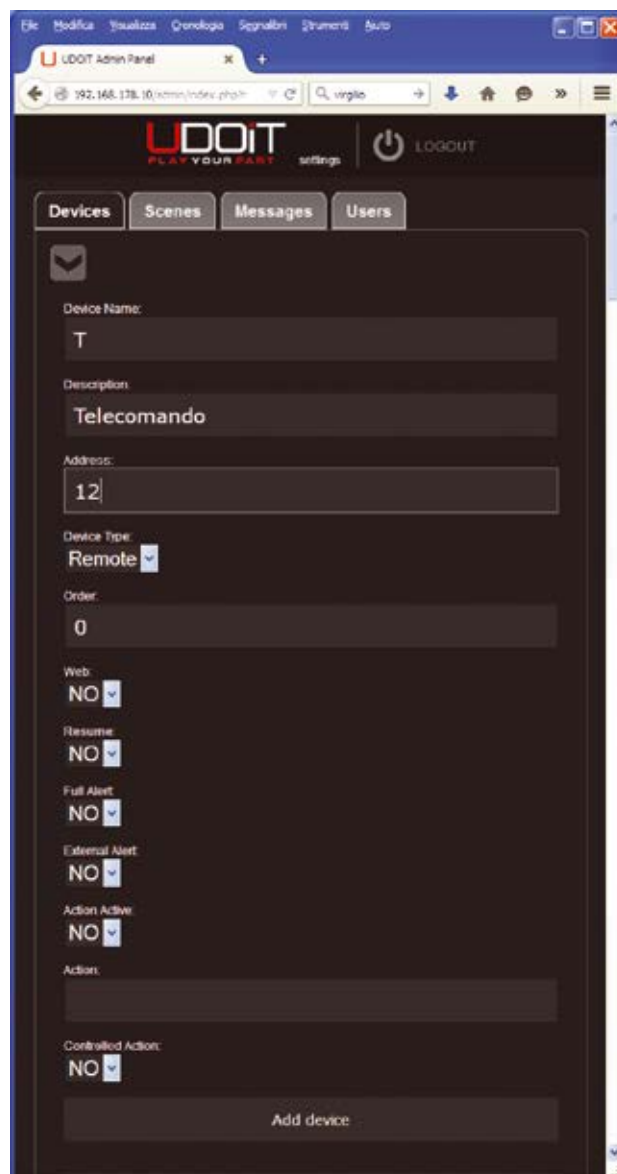
il caso di un telecomando correttamente associato all'applicazione con il proprio indirizzo. In caso di "smarrimento" del telecomando la soluzione più logica sembra essere quella di procurarsi un nuovo telecomando ed eseguire su quest'ultimo l'operazione di learning mantenendo l'indirizzo e le configurazioni associate al vecchio telecomando. Questa soluzione ha una controindicazione da non sottovalutare; chiunque ritrovi il vecchio telecomando potrà utilizzarlo per impartire i comandi esattamente come con quello nuovo, con gravi controindicazioni in termini di sicurezza. L'alternativa di cancellare il vecchio telecomando e configurare quello nuovo con un indirizzo diverso costringe ad un grosso lavoro di riconfigurazione. La soluzione adottata è quella di "personalizzare" ciascun telecomando con un proprio codice identificativo e di verificare ad ogni comando che questo provenga dal telecomando desiderato. In questo modo i comandi provenienti dal vecchio telecomando saranno semplicemente ignorati. Nel seguito descriveremo il processo per fare in modo che l'applicazione riconosca il telecomando corretto. Tornando allo schema di Fig. 9 notiamo che quando viene premuto uno dei tasti viene inviato un pacchetto contenente il numero del tasto premuto. Il pacchetto viene letto e recepito dal Dongle, master della rete, che lo consegna all'applicazione di gestione. In un'apposita tabella dell'applicazione di gestione sono configurate delle "scene". Nel mondo della domotica, ciascuna "scena", o "scenario", è composto da una lista di comandi elementari che devono essere eseguiti in un unico processo, come per esempio accendere tre lampade o spegnere tutte le luci e attivare l'antifurto prima di uscire. Di solito, come anche nel caso dell'applicazione di gestione della soluzione UDOiT si possono inserire quante scene si vogliono e configurarle in modo molto flessibile per gestire gli attuatori presenti. Nel caso del telecomando, quando viene aggiunto all'applicazione, il processo di aggiornamento genera automaticamente quattro scene, una per ogni tasto disponibile, che possono poi successivamente essere personalizzate in modo da eseguire i comandi desiderati. Una volta ricevuto il pacchetto inviato dal telecomando, l'applicazione lo elabora in modo da "lanciare" la scena corrispondente al tasto premuto. Il lancio di una scena consiste nel reperire la lista dei comandi che costituiscono la scena stessa ed inviarli in sequenza nella rete per mezzo del "Dongle", master della rete. Per aggiungere un telecomando alla rete si utilizza sempre la funzione "Add Device" dell'applicazione di gestione, che si trova nella pagina "Settings". In questo caso si dovrà scegliere come tipo dispositivo "Remote" (Fig.11). Ovviamente scegliamo di non visualizzare il dispositivo sulla pagi-



na web. Si consiglia di utilizzare un nome corto per il telecomando in quanto lo stesso nome verrà utilizzato come prefisso per i nomi delle scene. Un nome di una sola lettera va benissimo: se assegnamo al telecomando il nome "T" le scene associate assumeranno i nomi da T-1 a T-4. Andando alla pagina "Scenes" della funzione "Settings", come nella Fig. 12, possiamo vedere le quattro scene generate automaticamente ed associate ai tasti del telecomando.

Il processo di learning del telecomando include anche la memorizzazione del codice identificativo. La prima parte del processo è identica a quella degli altri dispositivi. Si prenota la funzione di learning per il teleco-

mando. Per portare il telecomando in stato di learning si devono premere contemporaneamente i tasti 1 e 3 per qualche secondo finché il LED da blu diventa rosso. Cliccando su OK vedremo il LED lampeggiare per qualche attimo, al termine l'operazione di learning è conclusa. Il codice identificativo viene assegnato con il pacchetto di risposta al comando di learning, un pacchetto con la configurazione di Fig. 10 ma con il numero tasto (byte 1) uguale a zero. Per associare un



Comando	Indirizzo	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
0x00	0x00	Indirizzo in esadecimale	0x10	Tasto Premuto 0x01/0x04	Livello batteria	Livello batteria	SerialID LSB	SerialID MSB	Vers. Firmware
									0x00

Fig. 10



Fig. 12

diverso telecomando allo stesso indirizzo, come nel caso di smarrimento o furto, basta ripetere l'operazione descritta. Un nuovo processo di learning disassocia dall'applicazione l'identificativo seriale del vecchio telecomando ed identificativo. Il vecchio telecomando non avrà più effetti sull'applicazione. Ora possiamo configurare le scene in modo da personalizzare ciascun tasto del telecomando per fargli eseguire i comandi desiderati. Prima, però è opportuno approfondire il processo di configurazione ed il funzionamento delle scene.

LE SCENE

Nei sistemi di domotica e di automazione in generale per "scene" o "scenari" si intende la creazione di sequenze di operazioni preconfigurate che possono essere innescate ed eseguite a fronte di un evento, come l'attivazione di un sensore, o di una specifica richiesta dell'operatore, come la pressione di un tasto del telecomando. Dal punto di vista operativo le scene vengono configurate utilizzando la funzione "Scenes", anch'essa raggiungibile dalla pagina "Settings". Prima però è necessaria una premessa per descrivere con quali "mattoni" si costruiscono le sequenze di comandi che costituiscono le scene. Torniamo alla descrizione dei dispositivi che entrano a far parte della rete. Come abbiamo visto in precedenza il protocollo di comunicazione dei dispositivi "attuatori" prevede dei comandi in grado di variare lo stato degli attuatori stessi, nel caso del Relè, la possibilità di attivarlo e disattivarlo. Questi "comandi" sono gestiti dall'applicazione di controllo. Per facilitare la gestione

da parte dell'utente, a ciascun "comando", in fase di descrizione dell'attuatore, viene associato un nome mnemonico, che ne faciliti l'identificazione e che viene memorizzato, assieme al comando corrispondente, in una delle tabelle del database dell'applicazione. Per esempio se identifichiamo un attuatore relè con il nome "REL1", in fase di salvataggio (o di aggiornamento, nel caso eseguiamo delle modifiche ad un attuatore esistente), verranno create due entrate nella tabella dei comandi; uno con nome REL1_ON al quale verrà associato il comando di attivazione del relè ed uno con nome REL1_OFF al quale verrà associato il comando di disattivazione del relè. Fatta questa premessa, andando alla pagina "Settings" e cliccando sulla tab "Scenes" si apre la funzione di gestione delle scene. Se abbiamo già configurato un telecomando, come nel nostro caso, vedremo quattro scene già presenti; quelle che sono state generate automaticamente quando abbiamo descritto il dispositivo telecomando. Per aggiungere una scena indipendente dal telecomando si deve cliccare sull'icona a forma di "+". Le scene associate ai tasti del telecomando possono essere attivate sia utilizzando i tasti del telecomando che dall'interfaccia web dell'applicazione. Tutte le scene possono anche essere attivate mediante script che a loro volta possono essere schedulati a tempo, associati all'attivazione di sensori o utilizzati come ponte di comunicazione con applicazioni esterne. Questo è un argomento avanzato che non affrontiamo al momento



Fig. 13



Fig. 14

ma tratteremo in seguito. Nella pagina di sintesi che espone la lista delle scene, visibile in Fig. 13, ciascuna scena è identificata dal nome e dall'ordine nel quale viene presentata nella pagina web di gestione dell'applicazione. Le icone a fianco di ciascuna entrata, permettono la "manutenzione" della scena.

In particolare la prima icona permette di modificare la descrizione ed il comportamento della scena, la seconda permette di comporre la lista di comandi da associare alla scena e la terza permette di eliminare la scena dal sistema. Si può notare che l'icona di eliminazione non è disponibile per le scene associate ai tasti del telecomando. Queste e le altre scene associate a funzioni di "sistema" sono create e cancellate in automatico e non possono essere eliminate dall'utilizzatore. Per esempio quando si elimina l'entrata associata ad un telecomando nella pagina "Devices" vengono cancellate anche le quattro scene associate a quest'ultimo. Per aggiungere una scena nel sistema si utilizza la funzione "Add Scene" (Fig. 14) dell'applicazione di gestione.

Nella stessa figura si vede che la scena è caratterizzata da un nome, di lunghezza massima quattro caratteri, una descrizione, che apparirà nelle pagine web di gestione del sistema. Abbiamo poi la possibilità di scegliere se vedere o meno l'entrata del dispositivo nella pagina web di gestione dell'applicazione

e in questo caso scegliere una posizione nell'elenco (campo "Order"). Scegliamo l'opzione "SI". I campi successivi servono per pilotare, in caso di attivazione, l'esecuzione di un processo personalizzato. Queste ultime funzionalità sono più complesse da configurare e personalizzare e le affronteremo in seguito in un articolo dedicato.

Per "salvare" la scena nel sistema clicchiamo sul pulsante "Add Scene". Una volta inserita una nuova scena è possibile associare la lista dei comandi che devono essere eseguiti quando viene attivata. Dalla finestra "Scenes" clicchiamo sull'icona a forma di tre ingranaggi (Configure Scene) in corrispondenza della scena che vogliamo configurare. In alto si apre la scheda visibile in Fig. 15. Nella riga in basso possiamo scegliere quale comando associare utilizzando il campo a tendina che presenta la lista di tutti i comandi disponibili nel sistema. Il primo campo permette di stabilire la posizione che vogliamo assegnare al comando all'interno della lista. L'ultimo campo permette di introdurre un ritardo tra l'esecuzione di un comando ed il successivo. Dato un comando, il ritardo viene applicato prima dell'esecuzione del comando. Per chiarire, se abbiamo una lista di due comandi e vogliamo introdurre un ritardo tra i due, dobbiamo specificare il valore del ritardo nel secondo comando. Questo dato è espresso in millesimi di secondo, ad esempio 2000 indica due secondi. I valori sono incrementali, quindi per aggiungere un ulteriore ritardo di due secondi



Fig. 15



Fig. 16

tra il secondo ed il terzo comando bisogna indicare il valore 4000. Una volta definito un comando lo aggiungiamo alla lista cliccando sull'icona a forma di "+" (Fig. 16). Se vogliamo modificare un comando già inserito nella lista lo facciamo direttamente sull'entrata della lista e lo confermiamo cliccando sull'icona a forma di chiave inglese.

Per cancellare un comando della lista clicchiamo sull'icona a forma di bidone della spazzatura in corrispondenza del comando da eliminare. Per confermare il tutto clicchiamo sull'icona in alto a forma di freccia orientata verso il basso.

Con queste funzionalità creiamo due scene, una con il comando che attiva il relè ed una con il comando che lo disattiva. In alternativa, se abbiamo un telecomando descritto nel sistema, possiamo configurare le scene associate ai pulsanti. Andiamo ora alla pagina principale di gestione del sistema, dove troviamo i pulsanti associati alle scene create, come visibile in Fig. 17. Se clicchiamo sui pulsanti associati alle scene vedremo attivarsi e disattivarsi il relè associato. Se le scene corrispondono ai tasti del telecomando otterremo lo stesso effetto premendo i corrispondenti tasti del telecomando. Ovviamente il tutto funziona in modo disgiunto. Possiamo attivare il relè premendo un tasto del telecomando e disattivarlo dall'interfaccia web.

FUNZIONE ALLARME

La gestione degli allarmi è un primo esempio preconfezionato della integrazione possibile tra sensori, attuatori, scene e script di controllo personalizzati. Lo

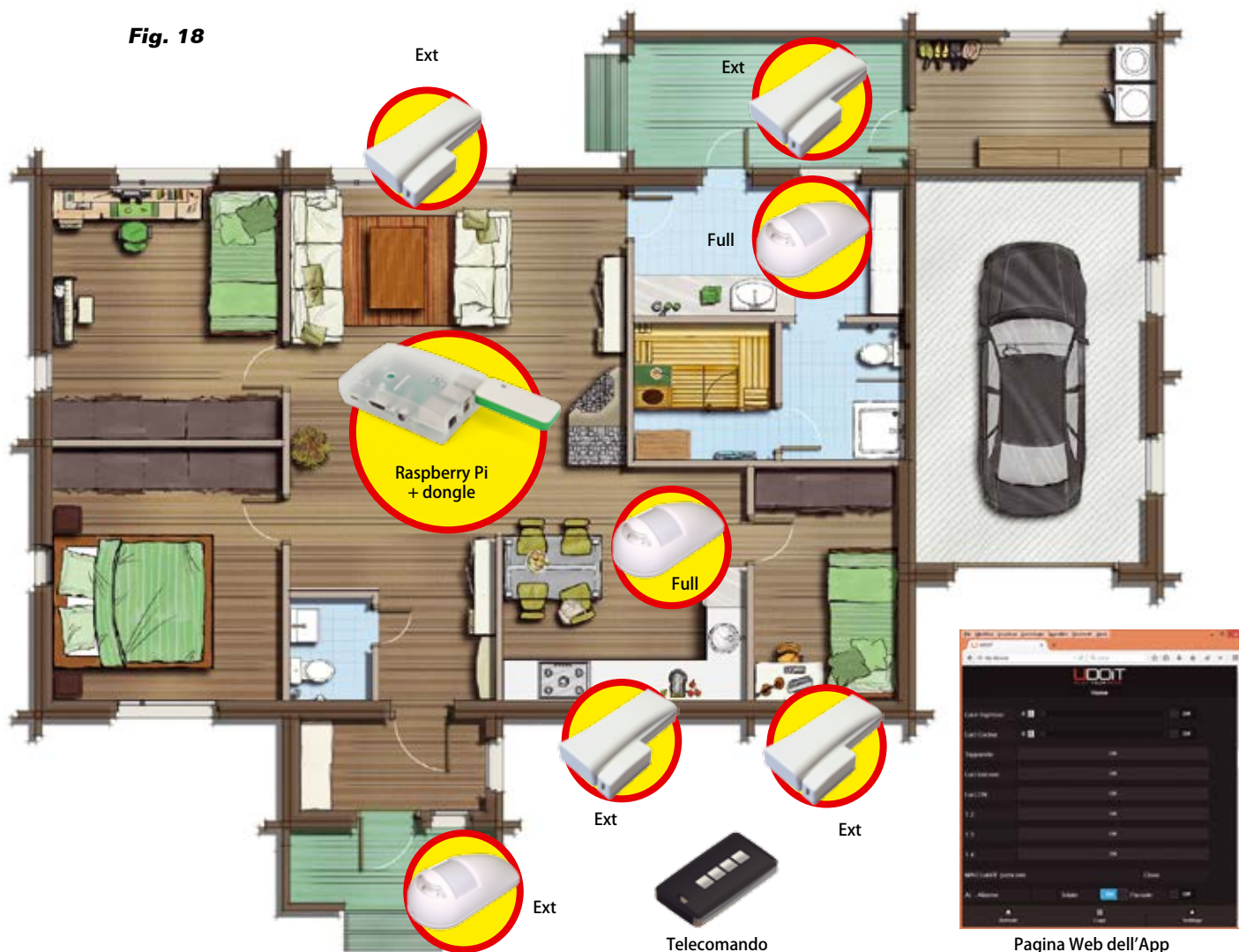
scopo è quello di proteggere o comunque di tenere sotto controllo un'ipotetica abitazione con una rete di sensori opportunamente predisposti. Questa funzionalità preconfezionata prevede due livelli di "controllo", un livello di controllo periferico, o "esterno", al quale possiamo associare i sensori collegati a porte e finestre esterne in modo da intercettare eventuali entrate ed uscite dall'abitazione ed un livello di controllo "totale" al quale possiamo associare anche sensori in grado di intercettare eventuali movimenti all'interno dell'abitazione. Come nell'esempio visibile in Fig. 18 possiamo prevedere dei sensori magnetici per rilevare l'apertura di porte e finestre esterne e dei sensori PIR per rilevare movimenti all'interno dell'abitazione. Come visibile in Fig. 19 l'interfaccia grafica prevede un'entrata predefinita per l'attivazione e disattivazione delle funzionalità di controllo totale ed esterno. Si attiva o disattiva ciascuna funzionalità cliccando sul corrispondente cursore ON/OFF. Nel caso sia attiva la funzionalità di controllo esterno e si decide di attivare il controllo totale, automaticamente verrà disattivata la funzionalità di controllo esterno. Lo stesso comportamento nel caso opposto.

Come abbiamo già descritto per il telecomando, anche alle funzionalità di controllo sono associate delle scene predefinite che vengono innescate quando i sensori si attivano. In particolare a ciascuna funzionalità sono



Fig. 17

Fig. 18



associate due scene. La logica dell'applicazione prevede che quando uno dei sensori associati alla funzionalità si attiva viene eseguita la prima scena associata alla funzionalità quella che ha suffisso -ON nel nome, ovvero ALT-ON per il controllo totale e ALP-ON per il controllo interno. Successivamente, sempre la logica predefinita dell'applicazione prevede che dopo un intervallo di un minuto vengono attivate le corrispondenti scene con il suffisso -OFF nel nome, rispettivamente ALT-OFF e ALP-OFF (Fig. 20). In questo modo, per esempio, controllando uno o più attuatori Relè, è possibile attivare una sirena per un minuto, oppure innescare l'ingresso ausiliario di un secondo sistema antifurto o ancora accendere luci.

Utilizzando poi le funzionalità avanzate di personalizzazione mediante script, che descriveremo nei prossimi articoli, è possibile inviare messaggi mail, o via applicazioni social, o interagire con applicazione esterne, sia locali che disponibili per mezzo della rete internet. Per attivare la struttura di controllo è necessario utilizzare la pagina "Devices" della funzione "Settings".

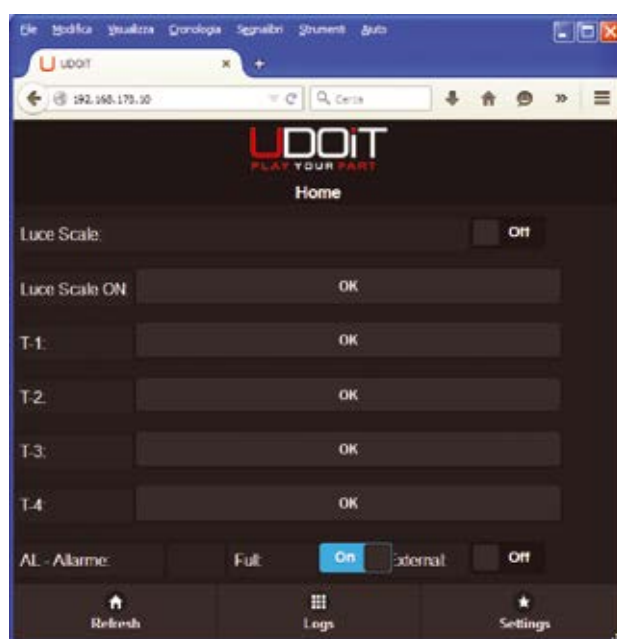


Fig. 19



Fig. 20

Per ciascuno dei sensori che si vuole far partecipare alla struttura di controllo è necessario aprire la funzione di update del sensore stesso, cliccando sull'icona a forma di chiave inglese, in corrispondenza del sensore desiderato ed impostare a "SI" uno o entrambi i campi "Total alert" e "External alert". Questa operazione deve essere eseguita per tutti i sensori che si vogliono far partecipare alla rete di controllo. Successivamente è necessario configurare le scene associate alle funzioni attivate, per esempio per attivare e disattivare uno o più relè, secondo le modalità descritte nel paragrafo precedente dedicato alle scene (Fig. 21 e Fig. 22).

Fig. 21

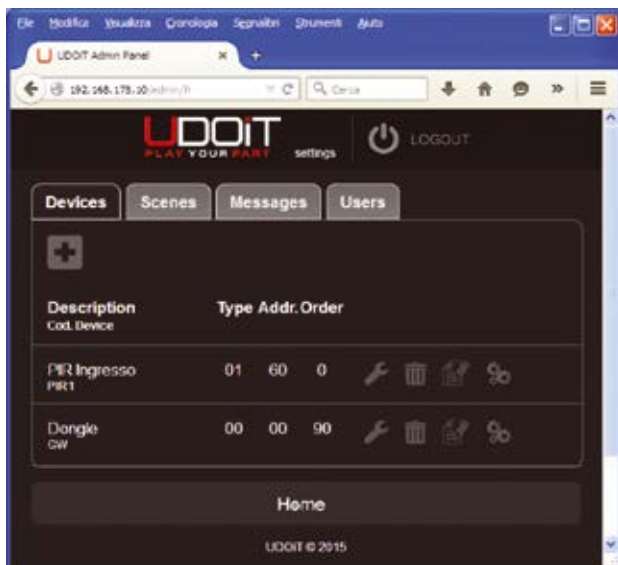


Fig. 22



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

<http://www.futurashop.it>

Due petawatt per il super laser giapponese

Il laser, messo a punto da un team di fisici e ingegneri della Osaka University, in Giappone, teorizzato e descritto nel 2012 sulle riviste specializzate, è stato finalmente realizzato.

Il laser Lfex (Laser for Fast Ignition Experiment), lungo circa 100 metri, è il dispositivo più potente al mondo, anche se l'impulso da 2 petawatt è durato solamente un bilionesimo di secondo.

Il laser, che sprigiona una potenza record, il doppio di quella generata dall'Università del Texas, che si ferma ad un petawatt, non richiede molta energia per funzionare, e ha bisogno solamente di pochi Joule.

"L'energia del Texas Petawatt, dai 150 ai 200 Joules, è come una tazza di caffè o un servizio nel tennis piuttosto potente", ha precisato Michael Donovan,

direttore di Texas Petawatt. "È l'energia impiegata da una lampadina da 100 watt in due secondi", conclude.

Comunque i ricercatori hanno rivelato che la configurazione da due petawatt rappresenta solo un inizio, in quanto l'obiettivo è quello di mettere a punto molto presto un super-laser da dieci petawatt.

<http://iopscience.iop.org/>

Marte: le manovre per accogliere InSight

Al via su Marte le operazioni per accogliere InSight, la missione della Nasa programmata per il 28 settembre 2016, al fine di esplorare il Pianeta rosso.

Il satellite Mro (Mars Reconnaissance Orbiter) compirà un cambio di orbita per garantire le comunicazioni con InSight quando scenderà sulla superficie di Marte. I propulsori di Mro, che orbita attorno al Pianeta rosso dal 2006, si accenderanno per 77 secondi e ciò

basterà per garantire le comunicazioni con il lander InSight, che analizzerà la struttura interna di Marte.

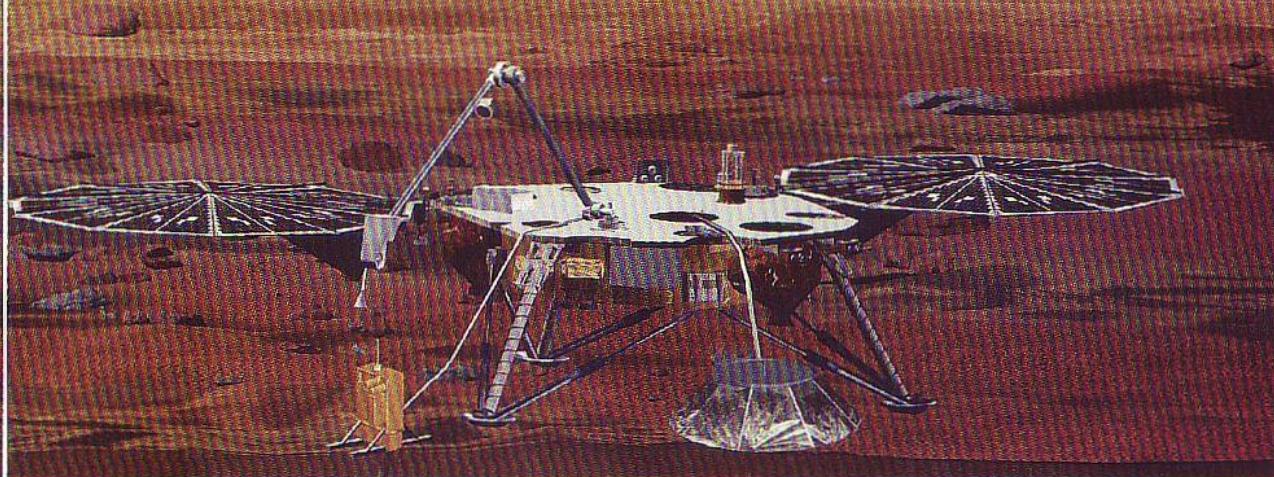
"Tramite il cambiamento d'orbita Mro potrà ascoltare InSight durante l'atterraggio. La manovra lo farà posizionare al posto giusto al momento giusto", ha spiegato il responsabile del satellite Dan Johnston, del Jet Propulsion Laboratory della Nasa.

Il satellite raccoglierà i dati della discesa di InSight per inviarli sulla Terra e

Mro continuerà nel suo lavoro di analisi di Marte dalla sua orbita tramite Sharad (Shallow Subsurface Radar), strumento realizzato da Thales Alenia Space per conto dell'Agenzia Spaziale Italiana (Asi) e l'università Sapienza di Roma.

InSight dovrà posizionarsi stabilmente sulla superficie di Marte ed esaminare la struttura interna del Pianeta Rosso.

www.jpl.nasa.gov



LISA Pathfinder: "ascoltare" l'Universo

"Ascoltare" il suono dell'Universo. O per lo meno verificare che sia possibile farlo. È questo l'ambizioso obiettivo di LISA Pathfinder, il primo osservatorio ESA - selezionato all'interno del programma Cosmic Vision - interamente dedicato allo studio delle onde gravitazionali, previste dalla teoria della relatività di Albert Einstein oltre un secolo fa. La missione, realizzata con un importante contributo italiano, ha appena brillantemente concluso la fase di test al centro IABG di Ottobrunn in Germania ed

è dunque stata ufficialmente presentata alla stampa. Il "cuore" dello strumento - i sensori inerziali, realizzati da CGS (Compagnia Generale per lo Spazio) con finanziamento dell'Agenzia Spaziale Italiana e su progetto dei ricercatori dell'Università di Trento e dell'INFN - era stato consegnato lo scorso novembre, con una cerimonia a Milano. Adesso il satellite è pronto a partire alla volta dello spazioriparto europeo di Kourou, nella Guayana francese, da dove sarà lanciato alle prime ore del mattino del prossimo 27 novembre. A portarlo in orbita sarà un vettore VEGA, il "piccolo-grande" razzo ESA "Made in Italy" - che porterà così a sei il conto dei lanci effettuati.

Lo scopo di LISA è mettere alla prova il concetto di rivelazione di onde gravitazionali dallo spazio dimostrando che è possibile controllare e misurare con una precisione altissima il movimento di due masse in condizioni di caduta libera.

"Rivelare le onde gravitazionali - ha spiegato Stefano Vitale, dell'università di Trento e dell'Istituto Nazionale di Fisica Nucleare (INFN), nonché PI del LISA Technology Package (LTP) - permetterà di studiare fenomeni estremi e oggetti difficili da osservare, come i buchi neri. Sarà anche possibile - ha aggiunto Vitale - fare la stratigrafia dell'universo fino a ricostruire l'origine delle galassie e dei buchi neri che possiamo osservare oggi".

www.asi.it

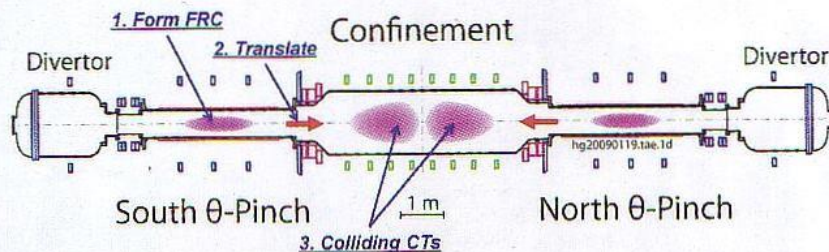
Tri Alpha Energy, ulteriori progressi sulla via della fusione nucleare

L'ultima relazione sullo stato delle ricerche risale al 2012, da allora più nulla, sino a pochi giorni fa quando, al simposio organizzato per commemorare il suo fondatore Norman Rostoker, Tri Alpha Energy ha annunciato di aver realizzato un prototipo del piccolo reattore nucleare C-2U e di essere riuscita a mantenere stabile a 10 milioni di gradi Celsius una massa di combustibile da fusione. Una "svolta" nel lungo percorso verso il raggiungimento della fusione nucleare controllata, così l'ha definita la rivista scientifica Science, una notizia che riaccende l'interesse nei confronti della corsa per l'energia pulita

dopo l'annuncio di fine 2014 da parte di un'altra società privata, la Lockheed Martin, sul proprio progetto di fusione nucleare Skunk Works.

La Tri Alpha Energy utilizza un reatto-

re relativamente piccolo composto da una camera centrale dove due anelli di plasma incandescente "sparati" da due cannoni laterali si fondono raggiungendo una temperatura di 10 milioni di gradi



Droni: scende in campo anche Intel con un investimento di 60 milioni di dollari

Attraverso un originale annuncio video realizzato grazie a un drone, Intel Capital ha comunicato di aver investito oltre 60 milioni di dollari in Yuneec International, leader mondiale nell'aviazione elettrica. Questo annuncio riguarda uno dei più grandi investimenti mai compiuti nell'industria dei droni.

La comunicazione, ripresa dal drone Typhoon 4K, è stata fatta dal CEO di Intel Corporation Brian Krzanich e dal CEO di Yuneec International Tian Yu. Krzanich e Yu hanno, inoltre, confermato che le due aziende collaboreranno su prodotti futuri. *"Siamo convinti che i droni abbiano il potenziale di cambiare le nostre vite in positivo in molti modi - e anche*

grazie alla tecnologia Intel Inside, i droni intelligenti di Yuneec creeranno straordinarie opportunità e possibilità", ha commentato Krzanich. *"Siamo felici di collaborare con un'azienda leader in questo nuovo e importante settore".*

Yuneec International utilizzerà questi fondi per accelerare lo sviluppo e il marketing dei prodotti.

"Con le sue avanzate politiche di sviluppo industriale-tecnologico e di controllo della produzione, Yuneec ha le potenzialità per espandere il mercato

consumer dei droni", ha aggiunto Tian Yu, CEO di Yuneec International. *"I nostri droni sono pronti per essere usati così come escono dalla confezione e garantiscono volo e riprese video stabili e fluidi".*

www.intel.com



Supercomputer da un exaflop, l'obiettivo di IBM e GENCI

IBM e GENCI, l'agenzia francese per l'High Performance Computing, hanno siglato un accordo di collaborazione per accelerare il cammino verso il calcolo su scala di exaflops, ossia la capacità di un sistema di eseguire almeno un exaflop, un miliardo di miliardi di operazioni aritmetiche in virgola mobile

Celsius. Il confinamento della sostanza incandescente (nessun materiale sarebbe in grado di contenerlo) viene ottenuto da un campo magnetico lineare prodotto dal flusso dello stesso plasma. Secondo i ricercatori il processo si è mantenuto stabile per 5 millisecondi, contro i 0,3 millisecondi del passato. La società californiana ha già messo in cantiere una macchina più grande e potente che nelle intenzioni dei ricercatori dovrebbe raggiungere una temperatura di 3 miliardi di gradi, più che sufficiente per la fusione del boro, materiale che non genera neutroni radioattivi.

<http://www.ialtenenergy.com/tri-alpha-energy.html>

(flops), al secondo. Attualmente, i sistemi più veloci del mondo eseguono tra dieci e 33 petaflops, ovvero da dieci a 33 milioni di miliardi di operazioni al secondo, circa dall'uno al tre per cento della velocità in scala di exaflops.

Per dare un'idea, se il calcolo su scala di exaflops è l'equivalente di un'automobile che raggiunge i 1000 km all'ora, i sistemi più veloci di oggi viaggiano a una velocità compresa tra 10 e 33 km orari.

La collaborazione, che avrà una durata di almeno 18 mesi, è incentrata sulla preparazione di applicazioni scientifiche complesse per i sistemi ora in via di sviluppo, destinati a superare i 100 petaflops, un importante passo avanti sulla strada verso la scala di exaflops. Lavorando a stretto contatto con gli esperti di supercomputing di IBM, GENCI avrà accesso ad alcune delle più avanzate tecnologie di High Performance Computing, frutto dell'ecosistema OpenPOWER in continua espansione.

Supportato da oltre 140 membri della OpenPOWER Foundation e da migliaia di sviluppatori in tutto il mondo, l'ecosistema

OpenPOWER comprende un'ampia varietà di soluzioni di calcolo, sia proprietarie IBM che Open Source, basate sulla tecnologia del processore POWER.

La collaborazione punterà a studiare l'impatto di alcune delle innovazioni di OpenPOWER, ad esempio valuterà il collegamento degli acceleratori GPU di NVIDIA ai processori POWER tramite l'interconnessione ad alta velocità NVLink di NVIDIA, o come gli switch Mellanox InfiniBand EDR a 100 Gb/s possano sfruttare la Coherent Application Processor Interface (CAPI) di IBM per migliorare radicalmente le prestazioni delle applicazioni. Inoltre, gli esperti di GENCI e delle organizzazioni di ricerca francesi, insieme a IBM, prevedono di approfondire la conoscenza dell'evoluzione dei modelli di programmazione, considerando l'integrazione di MPI e OpenMP come un primo passo verso una programmazione con un modello a memoria condivisa distribuita.

www.ibm.com



>



1

SVILUPPARE APP android con Processing

dell'ing.
MIRCO SEGATELLO

Come realizzare app native per dispositivi Android tramite l'ambiente di sviluppo Processing. In questa puntata iniziale installiamo l'ambiente di lavoro e realizziamo la nostra prima app.

Leggendo i dati relativi alle vendite nel mercato internazionale dell'elettronica di consumo, risulta evidente che in Italia il numero dei telefoni cellulari per abitante è tra i più elevati. Gli Italiani, infatti, prediligono la telefonia cellulare per mantenersi in contatto e poter usufruire dei molteplici servizi offerti dai social network. La tecnologia ha fatto passi enormi e in poco tempo si è passati da semplici cellulari, utilizzati solo per telefonare, a veri e propri computer portatili con funzioni integrate di fotografia e riprese video al cui interno gira un vero e proprio sistema operativo, in grado di gestire

specifiche applicazioni. A questo punto non li chiamiamo più telefoni cellulari ma smartphone, proprio per mettere in evidenza le capacità multimediali, di comunicazione e di calcolo di cui dispongono. Per enfatizzare le qualità multimediali, di recente vengono offerti smartphone con display grandi (5,5") e processori sempre più potenti, dando luogo a ciò che il mercato chiama "phablet" (phone tablet), un qualcosa a metà strada tra telefono e tablet. Gli smartphone spopolano fra i giovani, ma sono graditi da gente di tutte le età, soprattutto da chi ne fa uno strumento di lavoro indispensabile.

La battaglia tra i vari produttori di smartphone è molto serrata e si gioca tutto sull'innovazione e sulla ricchezza di funzioni, ma la grossa novità di questa generazione di telefoni è la possibilità di installare ulteriori programmi, oltre a quelli già forniti al momento della vendita. Tra le prime aziende ad intuire questo business ci fu la Apple ma, ben presto, quasi tutti i produttori di smartphone proposero la loro soluzione. Attualmente il mercato degli smartphone vede un buon 50% con installato il sistema operativo della Google, denominato Android, e a seguire il sistema operativo iOS della Apple; una piccola quota di mercato se la aggiudicano la Nokia con il suo Symbian, la Microsoft con il Windows phone, oltre ad altri sistemi operativi poco diffusi.

Senza entrare nell'accesa discussione su quale sia il miglior sistema operativo, possiamo senz'altro affermare che il mondo Android meglio si sposa con la filosofia dei Maker, perché da sempre Android è considerato un sistema "aperto" in cui l'accesso alle risorse hardware e software è risultato facilitato. Non a caso esistono diverse schede di sviluppo come ad esempio Google ADK, Arduino ADK e Sparkfun IOIO, che possono essere connesse via USB con uno smartphone Android e poter così sviluppare specifiche applicazioni in modo semplice e veloce, grazie anche ad applicazioni già pronte. Sicuramente vi sarà capitato di prendere in mano il vostro smartphone e di chiedervi come fare per realizzare una vostra app (abbreviazione del termine informatico applicazione) per poter interagire con i vostri progetti; ebbene, la cosa non è poi tanto dif-

ficile, come vi dimostreremo in questo corso. Prima di entrare nel merito facciamo una piccola precisazione, distinguendo tra app "native" e app "non native": le prime sono scritte in un linguaggio di programmazione specifico per il sistema operativo (Android, iOS, Windows Phone, Symbian, Blackberry OS, ecc.) su cui devono girare, mentre le seconde sono quelle che si utilizzano tramite browser internet (web app) ma anche quelle che vengono eseguite sfruttando degli ambienti di esecuzione intermedi.

Per realizzare un'app nativa per Android esistono diversi sistemi, ma tutti si basano sull'utilizzo dell'SDK ufficiale rilasciato da Google. Con il termine SDK (acronimo di Software Development Kit) viene inteso un kit di sviluppo software, tramite il quale qualsiasi utente potrà realizzare le proprie applicazioni. Questo software può essere gestito a linea di comando, ma per una più agevole scrittura del codice sorgente, si preferisce utilizzare un ambiente di sviluppo con un'interfaccia più intuitiva (IDE) come, ad esempio, Eclipse.

Google, per agevolare lo sviluppo di applicazioni, per il suo sistema operativo, ha da poco reso disponibile l'ambiente di sviluppo denominato Android Studio, il quale, similmente ad Eclipse e tramite l'SDK, permette lo sviluppo a livello professionale di applicazioni Android. Chi si fosse cimentato nell'utilizzo di questi ambienti di sviluppo (magari cercando qualche tutorial su Youtube) si sarà reso conto che non è proprio così immediato sviluppare un'applicazione, perché occorre fare pratica con i comandi e con la gestione del pannello di comando, oltre a sapersi destreggiare tra i vari menu e sotto menu.

Un'alternativa decisamente più alla portata degli hobbisti è rappresentata dalla disponibilità di un ambiente di sviluppo on-line denominato App Inventor, creato da Google nel 2009 per consentire praticamente a tutti di programmare (creare) piccole applicazioni ad uso personale. Poi Google, nel dicembre del 2011, decise di chiudere il progetto App Inventor e di passarlo al MIT (Massachusetts Institute of Technology) che lo chiamò MIT App Inventor Edu. Da dicembre 2013 si è passati a MIT App Inventor 2 il cui sito di

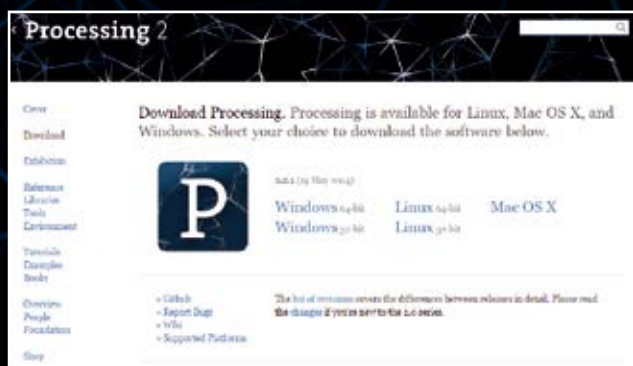
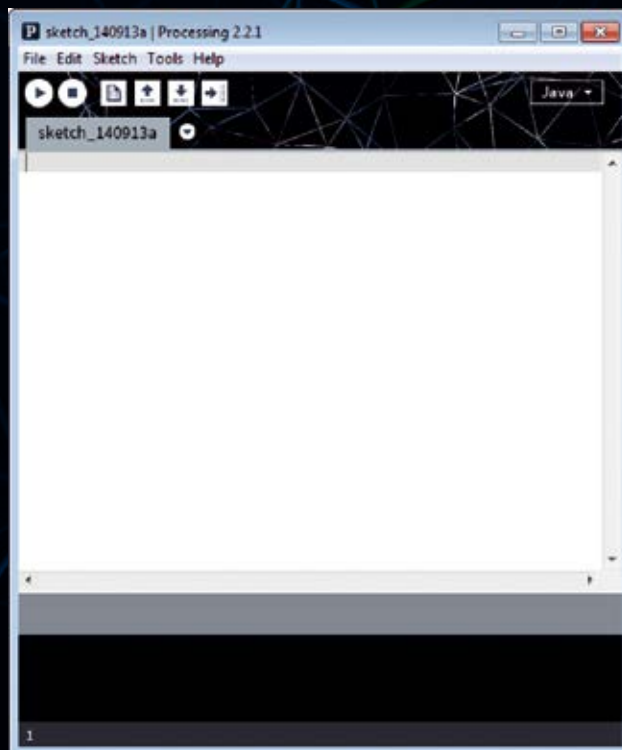


Fig. 1 - Download di Processing.

Fig. 2 - Processing PDE.

riferimento è <http://ai2.appinventor.mit.edu>. In questo ambiente di sviluppo la programmazione avviene, similmente al software (sempre del MIT) scratch, con un linguaggio iconico molto semplice ed intuitivo. Esiste anche un'ulteriore opzione per sviluppare applicazioni per Android e consiste nell'utilizzare l'ambiente di sviluppo Processing, la cui descrizione è proprio l'obiettivo di questo corso. Processing è un linguaggio di programmazione rivolto alla grafica, con il quale è possibile realizzare animazioni, immagini o visualizzare dati, è utilizzato da moltissimi designer, ingegneri, artisti, per via della sua immediatezza e per la facilità di utilizzo. Processing era inizialmente un progetto del MIT Media Lab passato poi, dal 2004, ad una fondazione non-profit che continua a supportarlo pur facendolo rimanere open-source. Insieme al linguaggio viene distribuito anche un semplice IDE (Integrated Development Environment) per permettere di programmare senza dover scaricare e configurare altri strumenti di sviluppo. Processing è multi-piattaforma e può essere utilizzato in ambiente Windows, Linux, Mac purché sia installato Java. Una volta installato ed aperto, scoprirete che l'interfaccia PDE (Processing Development Environment) è molto simile a quella di Arduino, basata sulla stessa semplicità e con un linguaggio di programmazione molto simile, infatti la piattaforma di sviluppo Wiring usata in Arduino è derivata proprio da Processing. Con Processing possiamo sviluppare su diversi sistemi operativi potendo scegliere quale compilatore usare a seconda del target del progetto. La modalità di programmazione predefinita al momento è Java, ma altre modalità di programmazione, come JavaScript e Android, possono essere aggiunte selezionando "Aggiungi Modalità ..." dal menu nell'angolo superiore destro del PDE. Proprio la familiarità con l'ambiente di sviluppo Arduino e la similitudine nel linguaggio di programmazione lo rendono un modo semplice ed intuitivo con cui sviluppare applicazioni native per Android. Dopo questa breve introduzione siamo pronti per iniziare a descrivere tutte le operazioni necessarie per poter scrivere il nostro primo programma per Android.

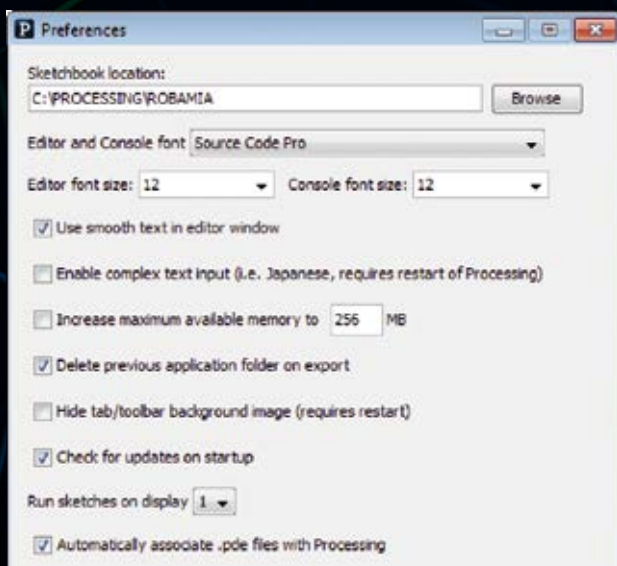


Visto che siamo una rivista di elettronica applicata, gli esempi saranno orientati all'interfacciamento con il mondo esterno; nello specifico, con Arduino-board e le sue shield.

PREPARAZIONE DELL'AMBIENTE DI LAVORO

Per il nostro esempio faremo riferimento ad un PC desktop in cui è installato il sistema operativo Windows 7 a 64bit. Poiché il linguaggio di programmazione per Android altro non è che un "dialetto" del linguaggio Java, avremo bisogno che sul computer in uso sia installato il JDK (java SE development kit), reperibile dal sito ufficiale: www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html, nel nostro caso è stata installata la versione jdk-7u21-windows-x64. Processing invece può essere scaricato dal sito: <http://processing.org>; per questo esempio abbiamo usato la versione 2.2.1.

Per creare un'applicazione Android ci si appoggia all'SDK ufficiale Android scaricabile all'indirizzo: <http://developer.android.com/sdk>; selezionate la voce "GET THE SDK FOR AN EXISTING IDE" e cliccate su "Download the stand-alone Android SDK tools for Windows"; il file da scaricare è *installer_r23.0.2-windows.exe*. Quest'ultimo è solo un pacchetto auto scom-



pattante i cui file possono essere copiati in qualunque posizione nel computer, ma ricordate che se userete una cartella di sistema, ad un certo punto avrete bisogno delle autorizzazioni di amministratore. A questo punto potete avviare Processing (Fig. 1) ed installare la modalità Android grazie al menu in alto a destra denominato *Add Mode* (Fig. 2). Se per qualche motivo la procedura dovesse fallire potete scaricare manualmente il pacchetto all'indirizzo: <https://github.com/processing/processing-android>. Questo pacchetto andrà scompattato all'interno di una cartella di nome *modes* che poi copierete all'interno della cartella usata per salvare gli sketch. Sempre dal PDE di Processing in *file-preferences* dovrete aver specificato il nome della cartella che userete per salvare i vostri programmi (Fig. 3) all'interno della quale è presente la cartella *modes* e la cartella *libraries* che conterrà le librerie aggiuntive.

Per essere certi che Processing raggiunga tutti i file di cui ha bisogno, è necessario impostare le variabili ambiente; allo scopo dovete andare in "Pannello di controllo\Sistema e sicurezza\Sistema\impostazioni di sistema avanzate\variabili d'ambiente", selezionare Path ed aggiungere i seguenti percorsi: *C:\xxx\android-sdk*, *C:\xxx\android-sdk\platform-tools*, *C:\xxx\android-sdk\tools*, *C:\yyy\Java\jdk1.7.0_03\bin* dove *xxx* e *yyy* è il percorso in cui avrete installato rispettivamente l'SDK e Java.

A questo punto potete avviare Processing e selezionare la modalità Android; al primo

Fig. 3 - Processing preference.

avvio vi verrà chiesto di inserire il percorso nel quale avete installato l'SDK. In modalità Android i menu a tendina si modificano mettendo a disposizione ulteriori comandi, tra cui il menu specifico per le funzioni Android. Avviate *Android SDK Manager* (dal menu Android) per poter completare l'installazione dell'SDK. Se avete installato l'SDK in una cartella di sistema dovrete avviare manualmente l'SDK manager lanciando (pulsante destro del mouse, esegui come amministratore) il file *SDK Manager.exe* presente nella cartella *android\sdk*. Senza entrare troppo nei dettagli, diciamo che il sistema operativo Android è in continua evoluzione e costantemente aggiornato e ad ogni nuova release è associato un nome (Ice Cream Sandwich, Jelly Bean, KitKat) le cui funzioni specifiche sono rese disponibili tramite delle API (Application

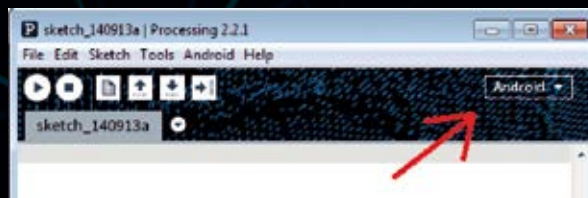


Fig. 4 - Processing in modalità Android.

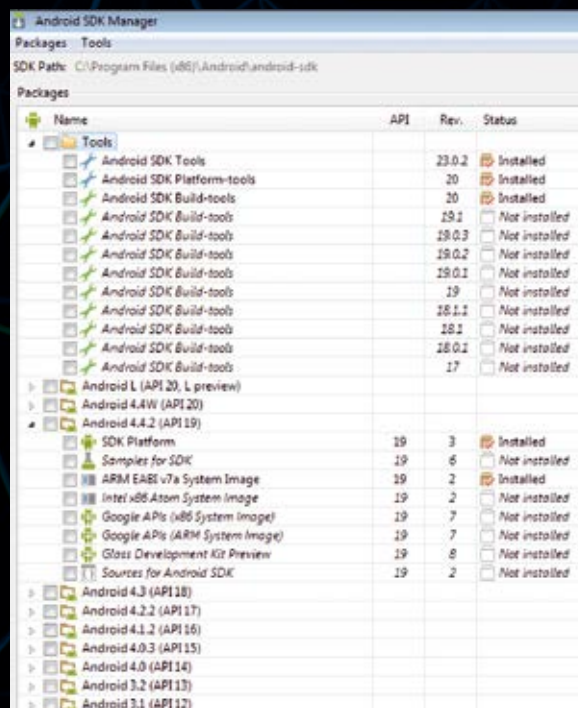


Fig. 5 - Android SDK manager.

Listato 1

```

/*ES1 Circle Draggert */

int circleWidth = 200;

void setup(){
  orientation(PORTRAIT);
  fill(255,100,0); //imposta il colore del disegno
}

void draw(){
  background(0); //imposta colore di sfondo del display
  ellipse(mouseX,mouseY,circleWidth,circleWidth); //disegna un cerchio
}

```

Programming Interface) permettendo agli sviluppatori di interagire con la piattaforma di sviluppo grazie ad un set di istruzioni predisposte. A ciascuna versione di Android corrisponde un relativo set di istruzioni ed in definitiva una versione delle API. Tramite l'SDK Manager avrete la possibilità di scaricare (è necessaria una connessione veloce ad internet) ed installare sia i Tools necessari al funzionamento dell'SDK, sia la versione

API più idonea. Nel nostro caso ipotizziamo di disporre di uno smartphone Samsung Galaxy S4 con installato Android versione 4.2.2, pertanto installeremo l'API 19 nelle due componenti essenziali *SDK Platform e ARM EABI V7a system Image*, oltre ai driver *Google USB Driver Package* (selezione in basso) e agli applicativi *Android SDK Tools, Android SDK platform-tools e Android SDK build-tools*. Confermate il download e l'installazione per completare l'operazione. Anche in questo caso non ci soffermeremo troppo sui dettagli della gestione dell'SDK, punto essenziale per gli sviluppatori professionisti, che devono cercare soluzioni



Fig. 6 – La nostra prima app.

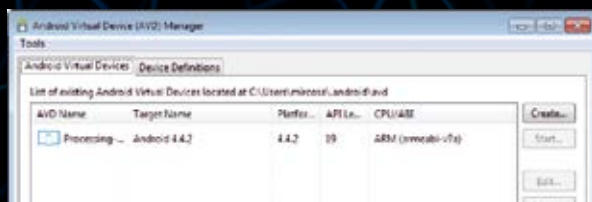
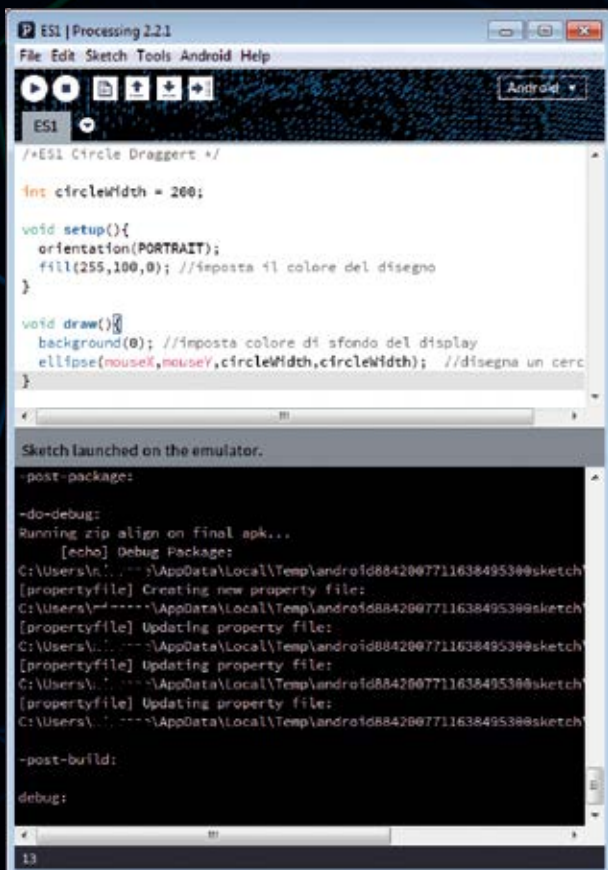


Fig. 7 – Android AVD Manager.



Fig. 8 – Creazione di un dispositivo nell'emulatore.



compatibili con diverse versioni del sistema operativo. Per il nostro semplice esempio possiamo dire che l'applicazione che andremo a sviluppare sarà compatibile con il

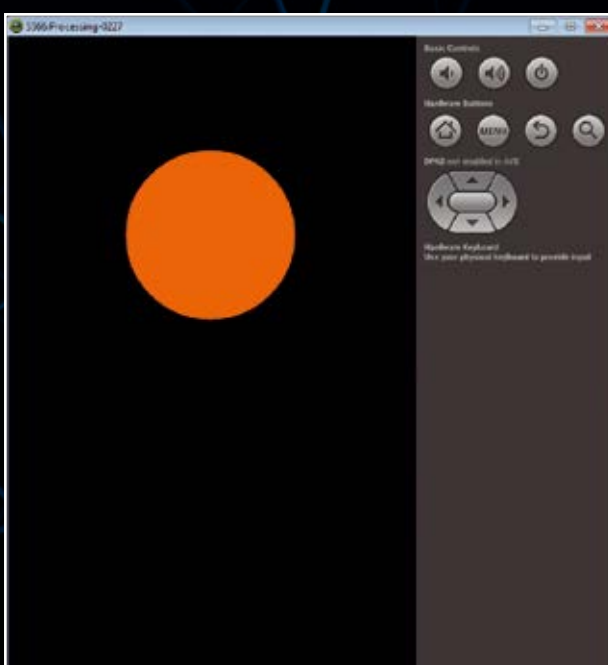


Fig. 10 – Avvio nell'emulatore.

Fig. 9 – Processing con applicazione avviata sull'emulatore.

nostro smartphone per il fatto che la sviluppiamo usando la stessa versione installata a bordo; vedremo oltre come sarà possibile definire per quali versioni di Android la nostra applicazione sarà compatibile.

La maggior parte delle API è retrocompatibile, il che significa che se sviluppiamo un'applicazione con API 15, questa girerà senza problemi in versioni più recenti; se state usando delle funzioni specifiche di questa API, l'applicazione andrà in crash quando si cercherà di avviarla in uno smartphone con sistema operativo più datato.

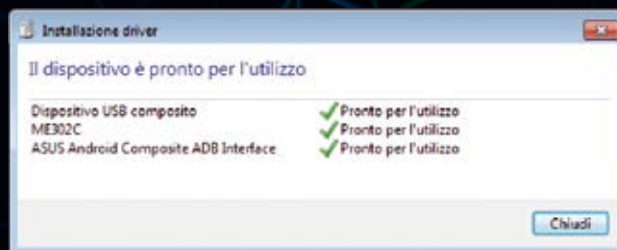
Se disponete di più dispositivi Android vi consigliamo di sviluppare la vostra applicazione con la versione API del dispositivo più vecchio che sicuramente girerà anche sul dispositivo più recente. La procedura per la preparazione dell'ambiente di lavoro è abbastanza lunga e non priva di insidie, quindi consigliamo di fare massima attenzione alla procedura descritta. Se tutto è andato a buon fine, siete pronti a scrivere la vostra prima applicazione.

Poche semplici righe di codice (**Listato 1**) sono sufficienti per realizzare un'applicazione che usa il display ed il tocco del dito, nello specifico, un pallino arancione che seguirà i movimenti del nostro dito sul display. Per quando riguarda i comandi base di Processing, si può fare riferimento al corso presentato nei fascicoli 159, 160, 161 e 162. Per questo corso focalizzeremo l'attenzione sui soli comandi specifici per Android; ulteriori dettagli sono disponibili nel repository https://github.com/processing/processing-android/wiki#About_Processing_for_Android.

La riga essenziale di questo listato è relativa al disegno di un cerchio (un'ellisse avente i due diametri uguali) di diametro *circleWidth*, disegnata nelle coordinate *mouseX*, *mouseY*, che il sistema fornisce quando il dito tocca lo schermo. Come al solito, salverete lo sketch nella cartella sketchbook indicata nelle preferenze; l'unica novità rispetto ad un progetto per il desktop è la presenza di un file dal nome *AndroidManifest.xml* creato automaticamente dal sistema, di cui ci occuperemo nel dettaglio in una prossima lezione. A questo punto vien voglia di vedere l'applicazione girare e nel fare questo abbiamo due opzioni disponibili nel menu *sketch*:

Fig. 12 – Installazione ADB driver.

Run on device e *Run on emulator*. Vediamo subito come sfruttare la seconda opzione, ovvero far girare l'applicazione direttamente nel computer in uso, richiamando un emulatore del dispositivo reale. Per prima cosa avviate *Android AVD Manager* e create un nuovo dispositivo con nome *Processing-0xxx* (xxx è la versione di Processing); indicate il *target* in relazione alla versione del sistema operativo che volete testare e definite il tipo di smartphone che volete emulare. In mancanza di un emulatore, il sistema ne creerà uno predefinito adatto alla versione API che avete installato. Ora siete pronti per avviare l'applicazione nell'emulatore; i tempi per l'avvio sono lunghi, anche alcuni minuti, perché viene simulata anche l'accensione del dispositivo; consigliamo pertanto di lasciare aperta la sessione per tutte le prove che vorrete fare. Può succedere di dover attendere alcuni minuti prima che l'emulatore sia correttamente avviato e di dover nuovamente lanciare il comando *run on emulator*. Nell'emulatore, il tocco del dito è simulato con il mouse; l'interfaccia grafica è facilmente verificabile, mentre l'uso delle periferiche hardware è testabile solo fino a un certo punto. Per i nostri esempi abbiamo selezionato come smartphone un Nexus4 con risoluzione 800x600, anche perché risoluzioni maggiori,



se il PC non dispone di un monitor di adeguata risoluzione non sono praticabili. Proviamo ora ad avviare l'applicazione nel nostro smartphone, seguendo attentamente le fasi successive. Per prima cosa è necessario abilitare una particolare funzione sul telefono affinché esso possa interagire con il PC in modalità di *debug*. A seconda della versione del sistema operativo la procedura è diversa e consiste nell'andare nelle impostazioni ed accedere alla sezione *Opzione sviluppatore* e selezionare *debug USB* e *Resta connesso*. Dalla versione 4 la voce *Opzione sviluppatore* è nascosta e per renderla visibile è necessario andare in *Impostazioni>Info sul dispositivo*, eappare per sette volte sulla voce *Versione build*. Collegate lo smartphone al PC tramite il cavo USB, vi verrà chiesto di installare i driver specifici per la funzione di *debug*, si tratta degli ADB driver (Android Debug Bridge) che forniscono un'interfaccia accessibile da PC per poter comunicare con il proprio smartphone. Nel caso di dispositivi Samsung è la stessa azienda produttrice a

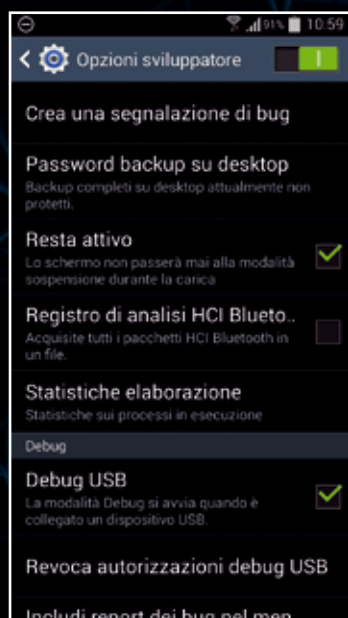


Fig. 11 – Opzioni sviluppatore.

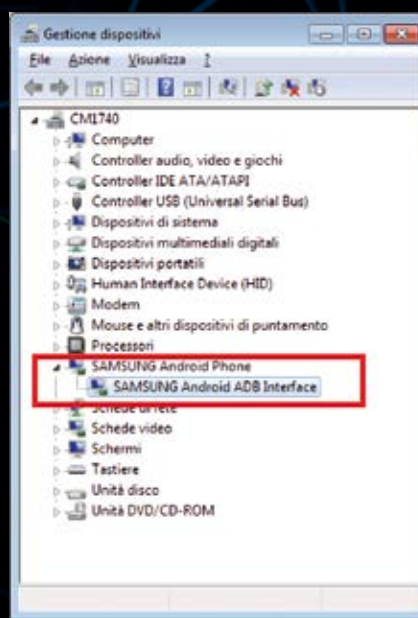
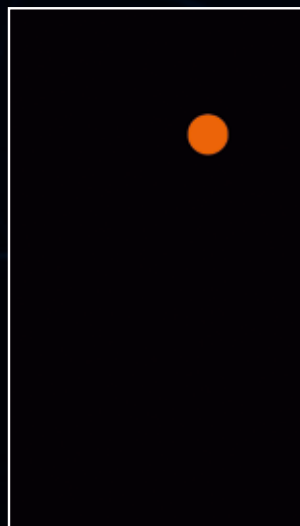


Fig. 13 – Verifica installazione ADB driver.

Fig. 14
Screenshot
dell'applicazione
del Listato 1.



mettere a disposizione questi driver all'indirizzo <http://developer.samsung.com/android/tools-sdks/Samsung-Android-USB-Driver-for-Windows> e analogamente per dispositivi di altre marche, ma alle volte sono sufficienti i driver disponibili tramite l'SDK; in questo caso è sufficiente indicare la cartella `usb-driver` contenuta all'interno dell'SDK, durante la fase di installazione dei driver. Potete verificare la corretta installazione dei driver accedendo alla sezione gestione dispositivi di Windows (Fig. 13) oppure lanciando il comando `adb devices` dal CMD di Windows; se sono settate le variabili d'ambiente basta digitare `adb devices` da prompt dei comandi.

Adesso è sufficiente cliccare su *Run on device* affinché l'applicazione venga caricata ed avviata sullo smartphone. Notate che l'app risulta correttamente installata sul dispositivo ed eseguibile a piacimento anche se sconnesso dal PC, ma solo su questo dispositivo; vedremo prossimamente come realizzare un'applicazione distribuibile. Tutta la procedura per preparare l'ambiente di lavoro richiede la massima attenzione; se incontrate problemi consultate il forum all'indirizzo <http://forum.processing.org/onetopic/android-problems.html>. Da poco è disponibile nel Play Store un'applicazione che permette di sviluppare app tramite Processing direttamente da dispositivi android: APDE (Android Processing IDE). Come indicato dagli autori, APDE (Android Processing Development Environment) è un ambiente di sviluppo integrato per la creazione di applicazioni su telefoni e tablet. APDE è in grado di editare, compilare ed eseguire un'applicazione direttamente sul dispositivo Android senza dover usare un PC o installare alcun SDK. Per il momento è in versione beta, però funziona bene e pensiamo possa avere un discreto successo. Si conclude questa prima puntata del corso; nella prossima analizzeremo le possibilità di personalizzazione dell'interfaccia grafica.

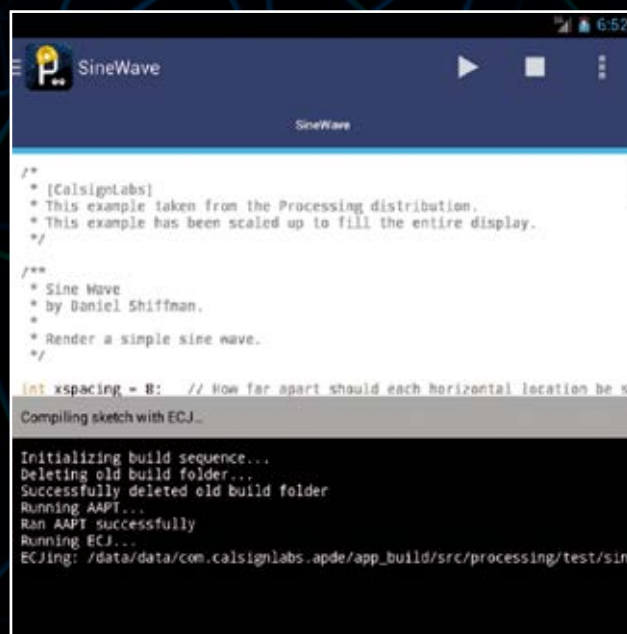


Fig. 15 – Processing per Android applicazione.

L'acqua? Ricaviamola dall'aria!

Durante una Conferenza Stampa che si è tenuta a fine Agosto presso il Padiglione Svizzero di EXPO 2015, la società svizzera di ingegneria SEAS (Société de l'Eau Aérienne Suisse SA) ha presentato gli ultimi sviluppi della propria tecnologia che consente di produrre, con grande efficienza, acqua di alta qualità dall'aria, catturando e trasformando l'umidità presente nell'atmosfera. Il progetto è stato sviluppato in collaborazione con l'Università di Pavia e la società IVR che produce i due sistemi attualmente commercializzati: ATW MOBILE e AWA MODULA, realiz-

zati presso gli stabilimenti IVR a Boca (Novara).

Il processo ideato da SEAS filtra l'aria ambientale condensando il vapore in acqua distillata che viene filtrata, sottoposta a un trattamento a raggi ultravioletti per eliminare i batteri e successivamente mineralizzata nuovamente per produrre un'acqua tra le più salubri, disponibile ovunque sia necessaria e non siano disponibili fonti alternative di approvvigionamento. L'acqua prodotta con questo metodo risulta meno inquinante rispetto a quella prodotta, ad esempio, dai dis-

salatori ed inoltre più sicura anche in caso di disastri ambientali (in quanto l'aria è più facile da purificare rispetto a un terreno contaminato).

Gli impianti SEAS sono in grado di produrre acqua di tre tipologie, da quella semi-potabile a quella potabile e mineralizzata (simile all'acqua minerale in bottiglia) a quella demineralizzata per uso farmaceutico o industriale.

I moduli SEAS possono essere alimentati anche con elettricità prodotta da generatori portatili o impianti fotovoltaici, che consentono l'utilizzo di questa tecnologia in zone remote o in caso di emergenza.

www.seas-sa.com



Ventotene, Capraia, le Canarie: le isole "efficienti" di Enel

Energicamente autosufficienti, dotate di sistemi di generazione elettrica innovativi, sostenibili sia dal punto ambientale che economico. Sono le "isole verdi" che Enel ha realizzato in Italia e Spagna: modelli di riferimento per l'accesso stabile e sicuro all'elettricità di località che il mare separa dalle reti di distribuzione nazionali e che possono essere replicati anche per paesi e aree isolati da infrastrutture di

distribuzione a causa di barriere geografiche come montagne e deserti.

Il modello realizzato nelle Canarie da Enel nel 2014 è un esempio già operante di come sia possibile rendere le isole 100% rinnovabili ed energeticamente indipendenti dall'importazione di combustibili fossili e dal collegamento con reti di distribuzione continentali. A El Hierro, la più piccola delle isole dell'arcipelago spagnolo,

Endesa ha infatti realizzato un impianto ibrido idro-eolico che ogni anno garantisce, oltre all'autosufficienza energetica, l'abbattimento di 18.700 tonnellate di CO2 e il risparmio di 1,8 milioni di euro di petrolio importati in passato dall'isola per il proprio fabbisogno energetico. In Italia, le isole di Ventotene e di Capraia offrono un altro esempio di come l'innovazione tecnologica possa generare benefici economici e ambientali: nell'isola al largo del litorale laziale Enel ha installato un sistema di storage con batterie agli ioni di litio mentre a Capraia è in funzione una centrale di produzione a biodisel a bassissimo impatto ambientale.

www.enel.it



Inaugurata la prima centrale elettrica che sfrutta la differenza termica delle acque oceaniche

È stata inaugurata e collegata alla rete di distribuzione elettrica il 21 agosto alle Hawaii la prima centrale OTEC (Ocean Thermal Energy Conversion) che sfrutta la differenza di temperatura tra l'acqua calda di superficie dell'oceano e l'acqua fredda in profondità, per generare elettricità pulita e inesauribile, 24 ore al

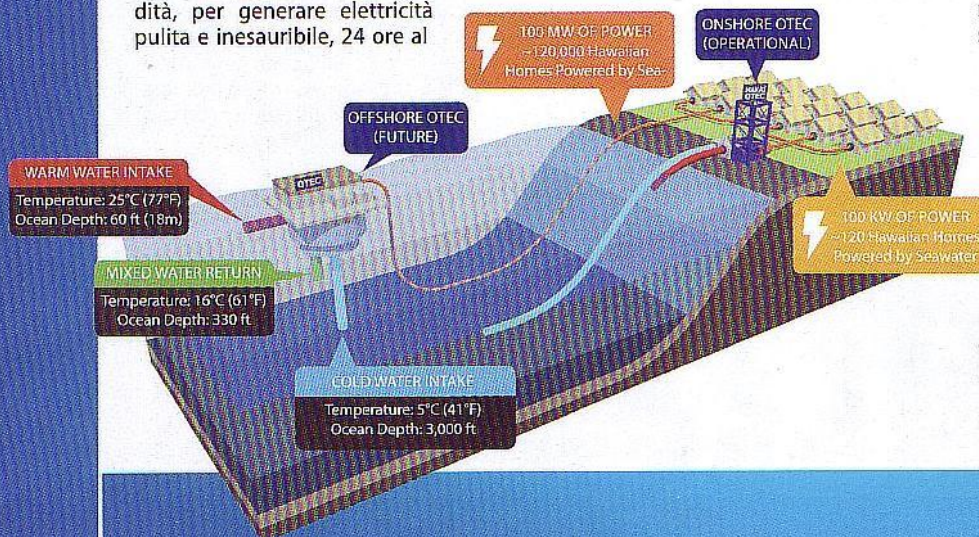
giorno, 365 giorni l'anno. L'impianto OTEC di NELHA (Hawaii) della Makai Ocean Engineering ha una potenza di 100 kWp, in grado di fornire energia a un centinaio di abitazioni circa.

Il principio è simile a quello di una centrale

termoelettrica a vapore: si utilizza un sistema chiuso nel quale circola un fluido in grado di evaporare alla temperatura dell'acqua di superficie (circa 25 gradi) generando vapore che mette in moto una turbina collegata ad un alternatore elettrico che produce elettricità; il vapore passa quindi attraverso un condensatore raffreddato dall'acqua aspirata dal fondo (circa 5 gradi) e ritorna in ciclo.

Attualmente i costi di realizzazione di una centrale di questo genere sono ancora troppo elevati rispetto agli altri sistemi di produzione di energia (compreso fotovoltaico ed eolico), ma un impiego su larga scala di questa tecnologia potrebbe portare ad una significativa riduzione dei costi, tanto da rendere queste centrali convenienti, specie nelle piccole isole del Pacifico e della fascia tropicale.

www.makai.com



Dagli Stati Uniti le turbine eoliche orizzontali

Grazie ad una "strozzatura" presente al centro, la velocità del flusso delle turbine aumenta dalle tre alle sei volte ed una rete posta in prossimità delle aperture evita agli uccelli di avvicinarsi troppo alle lame dell'impianto eolico che potrebbero rivelarsi mortali per gli animali.

Il sistema di manutenzione di tali dispositivi è meno costoso rispetto a quello delle turbine tradizionali ed è più facilmente gestibile, grazie alla minore distanza dal terreno.

"Il nostro obiettivo è quello di ridurre il

consumo energetico derivante dalle fonti fossili e l'unica soluzione per rispettare le restrizioni tra cui altezza, velocità del vento e salvaguardia degli uccelli è dato dal sistema di turbine orizzontali".

<http://sheerwind.com/>

Ecco una possibile soluzione per evitare che gli uccelli presenti nell'oasi naturalistica dell'atollo di Palmyra non vengano uccisi dalle pale delle turbine eoliche tradizionali di tipo verticale: costruire turbine eoliche orizzontali Invelox, inventate dalla SheerWind, un'azienda del Minnesota. Le turbine orizzontali, che hanno le sembianze di clessidre sdraiate su un fianco, lunghe 25 metri, con un grande ingresso circolare per il vento da una parte e uno scarico sul lato opposto, sono state acquistate dalla ONG Nature Conservancy proprio per Palmyra, dove nidificano oltre un milione di uccelli marini.



A San Francisco sorgerà la stazione di ricarica ad idrogeno più grande del mondo

Nella Baia di San Francisco nascerà presto la più grande stazione di rifornimento di idrogeno del mondo. Ogni tipologia di mezzo dotato di fuel cell ad idrogeno potrà dunque ricaricarsi nell'impianto comodamente.

Il dipartimento dei trasporti statunitense ha infatti stanziato i fondi per finanziare un'indagine, redatta con l'ausilio del Sandia National Laboratories, che valuterà l'effettiva convenienza del progetto che consentirà ad automobili, imbarcazioni, bus e ad ogni topologia di mezzo ad idrogeno di ricaricarsi nell'impianto.

"L'amministrazione marittima è impegnata a trovare tecnologie nuove ed efficienti dedicate al settore marittimo che riducano l'inquinamento e proteggano il nostro ambiente. Tale settore continua a fare progressi sul fronte delle energie rinnovabili e sulle opzioni verdi per il carburante. In fondo il nostro progetto spinge ad investire su un impatto ambientale dei combustibili marittimi sempre meno invasivo", ha spiegato Paul Jaenichen.

La stazione di ricarica ad idrogeno potrà erogare 1.500 chili di idrogeno al giorno e sarà la prima stazione di rifornimento a servire sia mezzi marini che terrestri.

www.sandia.gov



Stazione di ricarica wireless sulle autostrade inglesi

La società autostradale inglese sta preparando un sistema di ricarica wireless capace di ricaricare i veicoli a zero emissioni. A fornire la notizia la Highway England, ex UK Highways Agency.

Lo studio ed i primi test del progetto sono già stati realizzati su tratti di strada chiusi al pubblico. Le corsie green saranno costruite con spire

annegate nell'asfalto e cabine elettriche a bordo carreggiata. Se i test in corso daranno esito positivo, già entro 18 mesi i primi tratti di nastro asfaltato con corsie dedicate alle auto elettriche o ibride potranno entrare in funzione sulle autostrade inglesi, per essere poi estese sulle strade extraurbane. Il Ministro dei trasporti britannico Andrew Jones ha spiegato che il governo è pronto a fornire un finanzia-

mento di 500 milioni di sterline (circa 700 milioni di euro) per i prossimi cinque anni per porre l'Inghilterra all'avanguardia nel settore della mobilità sostenibile. Per Mike Wilson, capo ingegnere di Highway England, i primi esperimenti su strade chiuse al pubblico dovrebbero avere inizio entro fine 2015.

www.gov.uk

