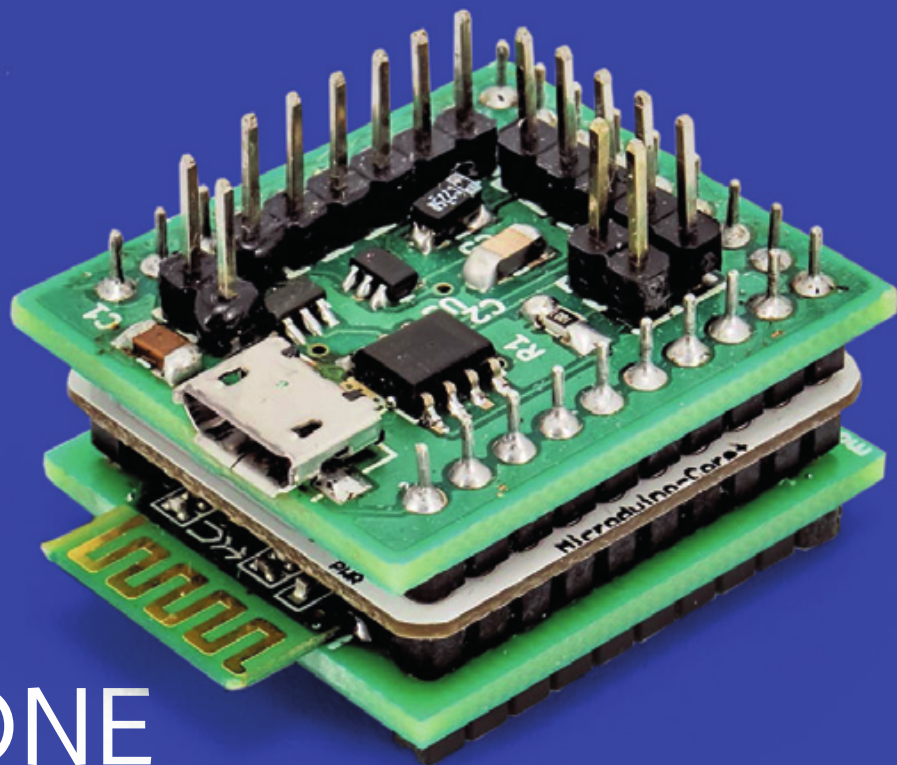


Monitoriamo temperatura, umidità e altri parametri pubblicandone i valori su un servizio web pubblico. Prima puntata.



# STIMA: MISURAZIONE DI PARAMETRI METEO E AMBIENTALI

di PAOLO PATRUNO E DANIELE CASTELLARI

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



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

### IL PROGETTO RMAP

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

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

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

- protocollo di rilevamento

dati ([www.raspibo.org/wiki/index.php/Gruppo\\_Meteo/DisciplinareStazione](http://www.raspibo.org/wiki/index.php/Gruppo_Meteo/DisciplinareStazione));

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

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

### I MODULI STIMA

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

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

### SENSORI

I sensori utilizzati per acquisire i parametri ambientali devono essere compatibili con il bus I<sup>2</sup>C, altrimenti si può risolvere il problema utilizzando un microcontrollore che adatta le letture (analogiche o digitali) e le elaborazioni (contatori, medie ecc.) rendendole disponibili su registri interrogabili tramite I<sup>2</sup>C-Bus. Ricordiamo che il protocollo I<sup>2</sup>C prevede l'utilizzo di un bus formato da due linee bidirezionali, chiamate SCL e SDA, le quali trasportano, rispettivamente, il segnale di sincronizzazione (clock) e i dati (la linea è bidirezionale). Abbiamo scelto il bus I<sup>2</sup>C in quanto:

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

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

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

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

## JSON RPC

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

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

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

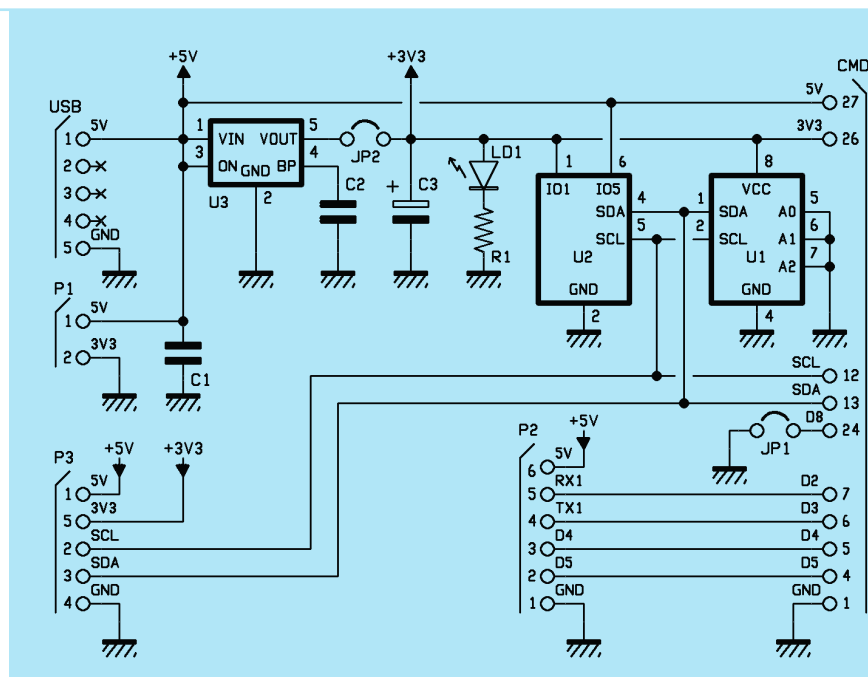
Il server che riceve la richiesta deve rispondere con una risposta valida a tutte le richieste ricevute. Una risposta deve contenere le

proprietà descritte di seguito.

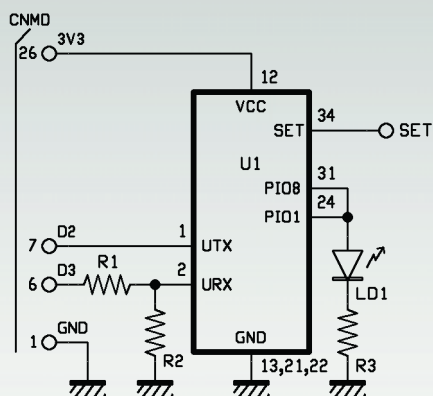
- *result*: i dati ritornati dal metodo invocato. Se c'è un errore invocando il metodo, il valore deve essere null.
- *error*: uno specifico codice di errore se l'invocazione del metodo ha dato luogo a un errore, altrimenti null.
- *id*: l'id della richiesta a cui si sta rispondendo.

## TRASPORTI

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



## [schema elettrico STIMA-Bluetooth]



Trasporti Passivi sono:

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

Trasporti attivi sono invece MQTT e AMQP.

### UN SISTEMA MODULARE

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

### MODULO STIMA BLUETOOTH

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

### BOARD MICRODUINO CORE+

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

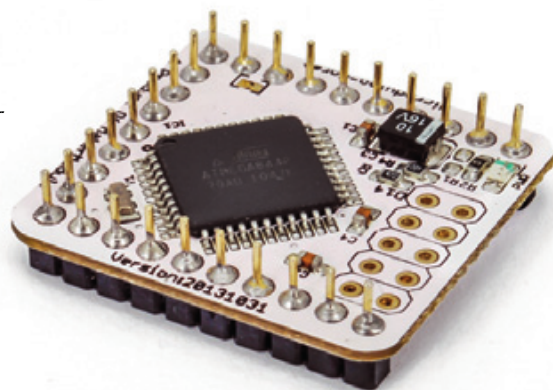
### BOARD STIMA-I2C

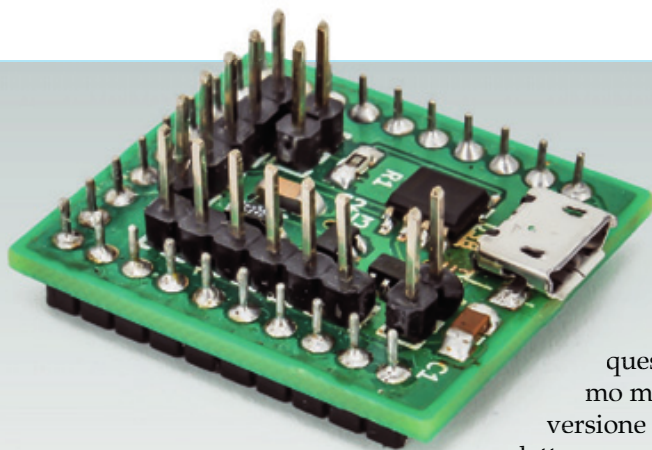
Questa board, assieme a Microduino core+, è utilizzata in tutte le configurazioni di Stima perché a bordo si trovano connettori per l'alimentazione, il bus I2C ecc. Abbiamo detto che la board può essere alimentata a 5V tramite il connettore micro USB, ma se lo preferite è presente anche il connettore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

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

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

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





(U3), lo stesso utilizzato anche dagli shield di Microduino. All'uscita di tale integrato è collegato anche un LED (D1) che ne indica il regolare funzionamento. Per proteggere i componenti da scariche elettrostatiche (tipicamente quelle dei fulmini) sono stati collegati ai pin del bus I<sup>2</sup>C e delle linee di alimentazione alcuni diodi di protezione (scaricatori) racchiusi nel componente ESDA6V1-5SC6 (U2) i quali si occupano di assorbire eventuali eccessi di tensione. Per darvi la possibilità di testare subito la board collegandola al PC o allo smartphone, è già presente un sensore di temperatura I<sup>2</sup>C modello TMP275(U4); già dalla prima connessione, in attesa di ampliare il vostro set di sensori, avrete la possibilità di verificare che tutto il sistema sia regolarmente funzionante. Non collegate contemporaneamente questo shield con lo shield Stima-I2C in quanto entrambi alimentano la linea a 3.3V e si potrebbero causare danni.

### BOARD STIMA-BLUETOOTH

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

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

### REALIZZAZIONE PRATICA

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

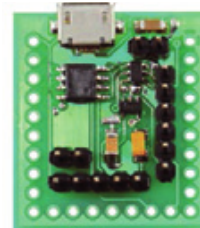
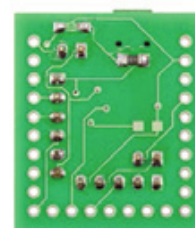
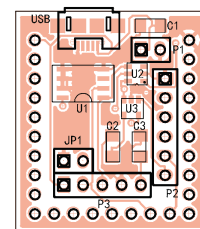
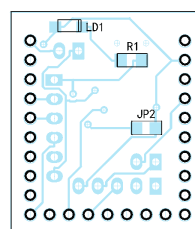
## [piano di montaggio STIMA-I2C]

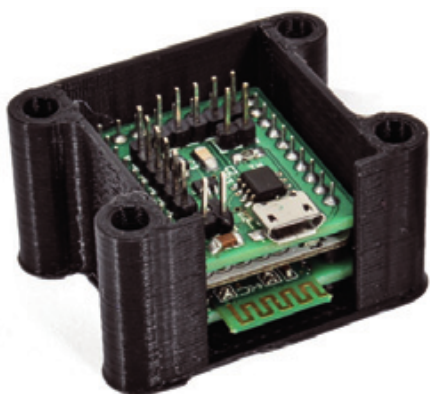
### Elenco Componenti:

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

#### Varie:

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





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

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

### IL FIRMWARE RMAP

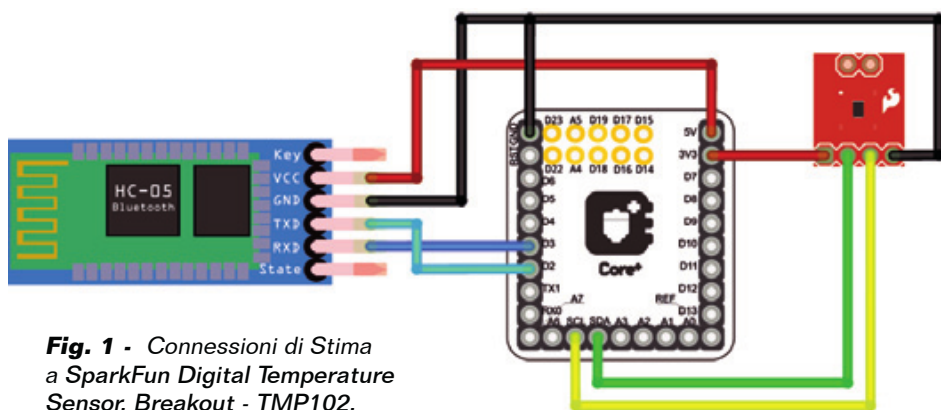
I dettagli del firmware dei moduli Stima verrà esaminato nella prossima puntata. Per ora basti sapere che esiste un file di configurazione per le opzioni a tempo e che altre configurazioni sono salvate runtime su eeprom tramite json-rpc. Seguendo le istruzioni seguenti vi troverete installati tutti i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

sequenza di fasi:

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

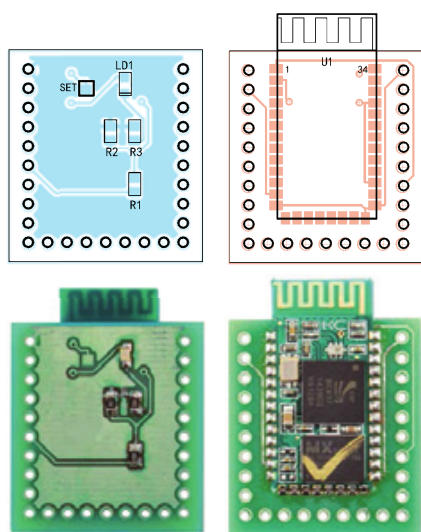
L'app ha due modalità di funzionamento: una interattiva e una in background. La modalità di esecuzione in background non richiede la presenza o l'intervento dell'utente e può essere concorrente all'esecuzione di altre applicazioni. Nello sviluppo dell'applicazione RMAP sono stati utilizzati Django e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

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



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

## [piano di montaggio STIMA-Bluetooth]



### Elenco Componenti:

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

Varie:

- Pin strip femmina torniti 27 vie
- Circuito stampato S1233

amministrativa, in molte lingue;

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

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

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

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

### UTILIZZO DI RMAP

Eseguiamo l'applicazione e vediamo come si presenta; l'interfaccia è schietta, per poter essere eseguita in ambienti differenti e bisogna dedicarsi un poco per appropriarsi di tutte le funzioni disponibili. In alto abbiamo i pulsanti di navigazione tra le pagine e quelli per attivare alcune funzioni; al centro le funzioni specifiche della pagina corrente e in basso i messaggi di stato e di errore, oltre ad alcune funzioni chiave da eseguire prima di abbandonare quella pagina.

Per pubblicare i dati sul server RMAP.cc bisogna prima di tutto registrarsi: facendo clic sul pulsante Registrazione si apre la pagina web della registrazione, localizzata a <http://rmap.cc/registrazione/register/>.

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

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

### Pagina "Avvia"

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

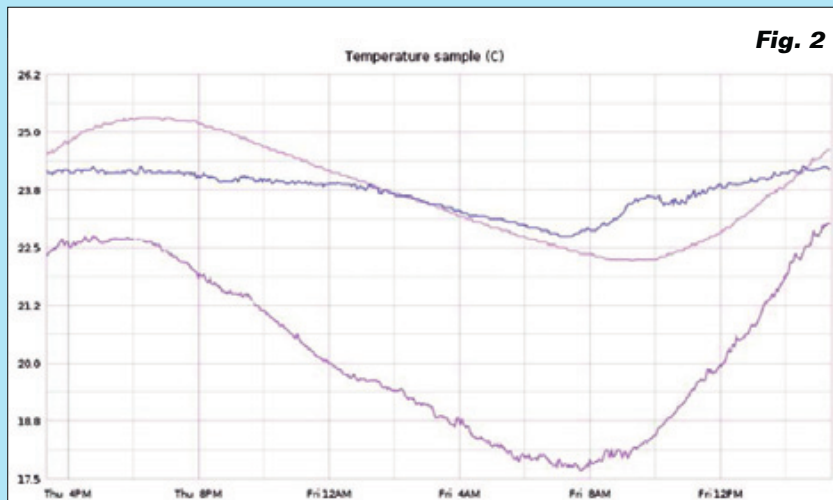
### Pagina "Posizione"

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

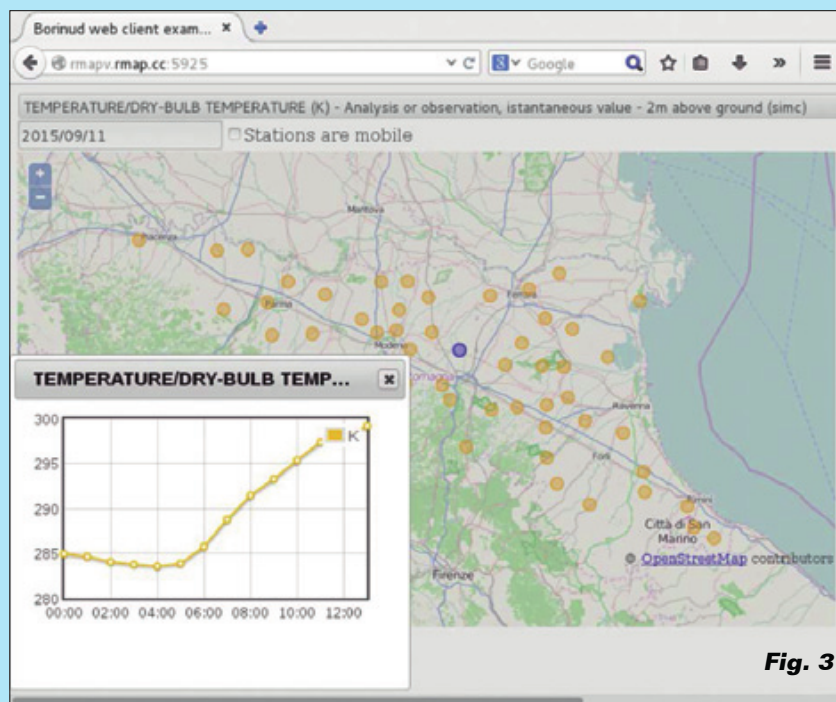


## Il server rmap.cc

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



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



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

### Pagina "Inserisci dati"

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

### Pagina "Dati automatici"

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

### Pagina "Pubblica"

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



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

## COMPILAZIONE FIRMWARE STIMA-BLUETOOTH

Passiamo alla compilazione dello sketch per Microduino core+:

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

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

## INSTALLAZIONE STIMA RMAP

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

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

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

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

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

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

## Windows

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

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

## Mac OS X

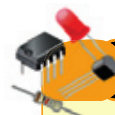
Per l'installazione in MacOS X, prima di tutto bisogna installare Kivy sul sistema operativo di Apple seguendo le istruzioni reperibili alla pagina web <http://kivy.org/docs/installation/installation-macosx.html> e poi installare gettext da <http://www.ellert.se/twain-sane/>. Fatto ciò, si impartisce:

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

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

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

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



## per il MATERIALE

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

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