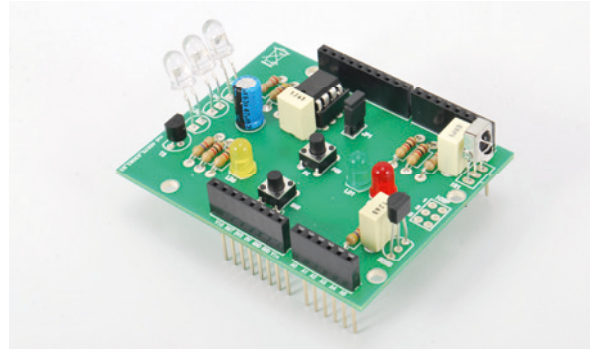
ArdIR Shield *(cod. FT1219K)*



Principio di funzionamento

Il funzionamento alla base di ArdIR è quello di un normale telecomando "universale": inizialmente necessita una fase di apprendimento per memorizzare i codici del (dei) telecomando (i) e in seguito, quando richiesto, tali codici vengono trasmessi all'apparecchio da gestire. Il *plus* sta nel fatto che ArdIR non si pilota tramite una tastiera fisica, ma con quella "virtuale" presentata nella pagina HTML/JS esposta dal suo Web Server con protocollo HTTP, quindi visibile collegandosi da smartphone o PC tramite un browser.

A livello pratico, il sistema hardware di principio potrebbe semplicemente essere composto da Arduino collegato ad un apposito shield dotato di ricevitore (per la fase di apprendimento) e LED emittenti nell'infrarosso. Per aggiungere la connettività alla rete, però, Arduino non basta: è necessario utilizzare la board **RandA** che alla semplicità di Arduino aggiunge la potenza di calcolo e la quantità di memoria offerte da Raspberry Pi, che qui sono necessarie per gestire le funzionalità del Web Server. Rispetto all'utilizzo di Raspberry Pi e Arduino come schede "separate", RandA aggiunge diversi vantaggi:

• l'installazione del software della Raspberry Pi include diverse librerie per la comunicazione con Arduino per facilitare l'integrazione tra i due sistemi; ad esempio, Arduino può direttamente trasmettere comandi Linux alla Raspberry Pi e scrivervi/leggervi file;

• la programmazione di Arduino può avvenire da Raspberry Pi, il cui software a corredo comprende anche l'IDE; ma è disponibile anche un IDE per PC "modificato" in grado di connettersi ad Arduino da remoto, nel momento in cui la Raspberry Pi sia collegata ad una rete;

• nel pacchetto software viene già incluso, configurato e funzionante, il server web con a corredo diversi applicativi e delle pagine web di esempio utilizzabili come base per sviluppare le vostre applicazioni;

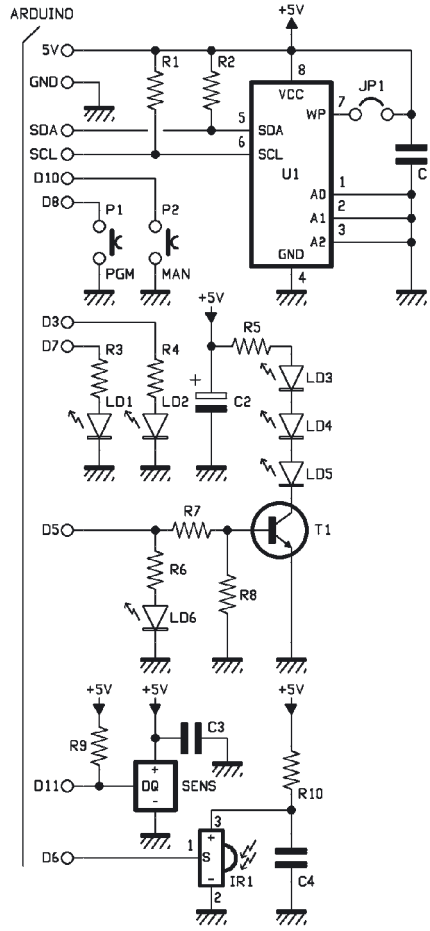
• è stato aggiunto un orologio in tempo reale (RTC) completo di batteria tampone, utile se si vuole -ad esempio- programmare delle attivazioni degli apparecchi basate sull'orario;

• dal punto di vista hardware, il sistema complessivo risulta più compatto: può venire alimentato da un'unica sorgente e non vi sono cablaggi "volanti" di collegamento tra una scheda e l'altra.

Interfaccia web

Per comunicare con lo sketch dall'esterno del "regno" di Arduino, come detto, abbiamo sfruttato l'"altra metà" di RandA, ovvero la Raspberry Pi e gli strumenti software che il mondo Linux mette a disposizione, in particolare il programma Apache Tomcat: esso è un web server, o meglio un web *application server* in quanto oltre a gestire tutte le funzionalità di un server, permette di eseguire programmi (scritti in Java e in questo contesto identificati come *servlets*) in base a pre-

Schema elettrico



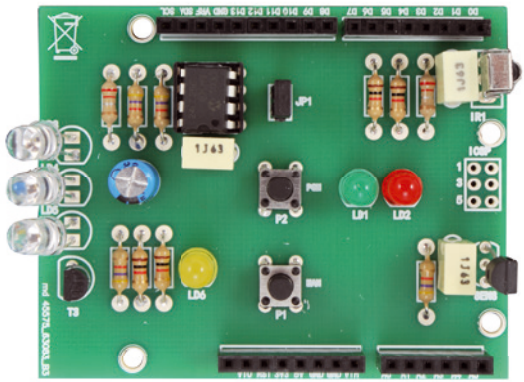
cise chiamate effettuate dal client remoto. L'uso di questi programmi permette di gestire la comunicazione seriale della Raspberry Pi e quindi di comunicare con Arduino. Nel nostro caso sfruttiamo la servlet "SerialIO", fornita nell'installazione del software di RandA perché già utilizzata da altre pagine web (Arduino Console, Arduino I/O management,…) comprese nell'installazione. Questa servlet accetta le richieste in arrivo dal client e le filtra sulla base del parametro *"cmd"*, per chiamare la relativa funzione che a sua volta trasmette le informazioni sulla seriale della Raspberry Pi, verso Arduino. In seguito alla trasmissione, attende la (eventuale) risposta in arrivo da seriale (quindi da Arduino) e la ritrasmette al client. Per capirne meglio i dettagli, se si mastica un po' il java, si può naturalmente editare il sorgente della servlet (SerialIO.class) che si trova nella cartella */home/apache-tomcat-7.0.47/webapps/RandA/WEB-INF/classes/ArduIO*. Tenete solo presente che, se voleste modificarla a piacimento, bisogna rifarne il *deploy* su Tomcat (al proposito rimandiamo alla consultazione della guida su *https://tomcat.apache.org/tomcat-7.0-doc/appdev/deployment.html*).

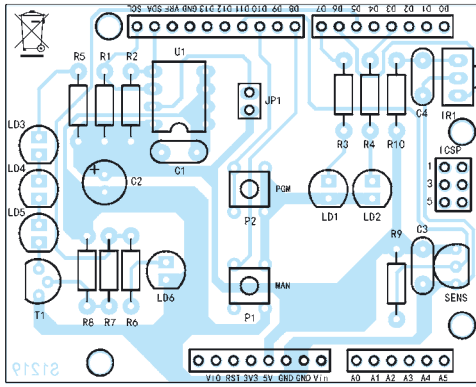
In **Fig. 5** vediamo la schematizzazione complessiva del flusso dei dati e degli attori che ne prendono parte, dalla pagina web al micro di Arduino, in cui si nota la funzione di interfaccia della servlet SerialIO tra il server web e la porta seriale (descritta da /dev/ttyS0) della Raspberry Pi.

Nella stessa figura possiamo adesso spostarci sulla sinistra per arrivare al lato client, ovvero alla pagina web da visualizzare in remoto. Questa pagina è suddivisa in due parti: una globale in html, utilizzata più che altro per costruire la presentazione (ovvero la grafica), ed una sezione di codice *javascript* che si occupa della gestione degli eventi (in pratica i "clic" sui tasti) attivati dall'utente.

Il meccanismo utilizzato è

Piano di montaggio



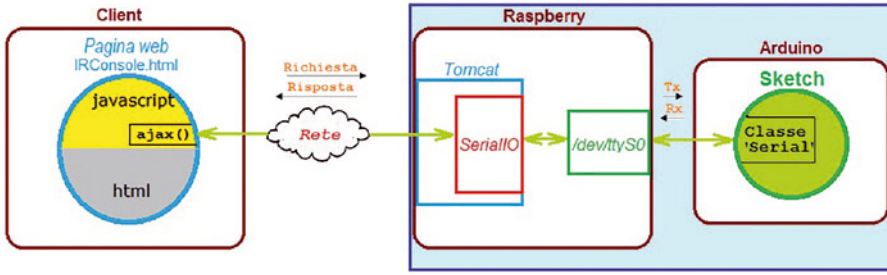


Elenco Componenti:

R1, R2: 4,7 kohm
R3, R4: 1 kohm
R5: 22 ohm
R6, R7: 1 kohm
R8, R9: 4,7 kohm
R10: 22 ohm
C1: 1 µF 63 VL poliestere
C2: 47 µF 16 VL elettrolitico
C3: 1 µF 63 VL poliestere
C4: 1 µF 63 VL poliestere
LD1: LED verde 5 mm
LD2: LED rosso 5 mm
LD3: LED IR 5 mm
LD4: LED IR 5 mm
LD5: LED IR 5 mm
LD6: LED giallo 5 mm
IR1: IR38DM
SENS: DS18B20+
T1: BC547
U1: 24LC256
P1: Microswitch
P2: Microswitch

Varie:
- Strip M/F 6 vie
- Strip M/F 8 vie (2 pz.)
- Strip M/F 10 vie
- Strip M/F 2x3 vie
- Strip maschio 2 vie
- Jumper
- Zoccolo 4+4
- Circuito stampato S1219

Fig. 5



molto semplice: ad ogni tasto dell'interfaccia grafica si associa una funzione javascript che sarà invocata ogni volta che il tasto viene attivato. Questa funzione può effettuare elaborazioni in locale (sulla macchina *client*) oppure, nel nostro caso in cui vogliamo trasmettere i comandi alla scheda RandA, chiamare a sua volta la funzione ajax() che, con opportuni parametri, instaura un colloquio col server, o meglio con la servlet SerialIO. Più specificamente il client invia una richiesta al server, che come visto attiverà una funzione sulla *servlet* che a sua volta trasformerà in una comunicazione seriale ad Arduino; la risposta di quest'ultimo, seguendo il percorso inverso, viene poi intercettata e restituita dalla funzione ajax() stessa (sotto forma di stringa di caratteri), per essere eventualmente visualizzata e dare all'utente il risultato richiesto o almeno un responso sull'esito dell'operazione.

Preparazione del sistema

Passando al lato pratico, la prima cosa da fare è realizzare lo shield ArdIR, saldando sull'apposito circuito stampato i pochi componenti necessari; non vi sono criticità particolari. Per quanto riguarda il montaggio dei componenti elettronici (fate soprattutto attenzione alla polarità!). L'unico suggerimento è quello di non saldare a filo scheda i diodi all'infrarosso LD3, LD4, LD5 ma di lasciarne i reofori lunghi almeno 1-2 cm in modo da poterne ottimizzarne l'orientamento verso l'apparecchio da gestire.

Per quanto riguarda la programmazione dello sketch ArdIR sul micro di Arduino, dopo avere alimentato e collegato RandA alla rete locale in modo da potervi accedere tramite TCP/IP, possiamo scegliere di procedere in due modi:

• installare su PC la IDE di Arduino modificata (inclusa nella distribuzione di RandA), che permette di "vedere" la scheda Arduino di RandA al relativo indirizzo IP (verificare che sia selezionato sul menu Strumenti → Porta seriale); in seguito è sufficiente caricare lo sketch e programmare Arduino come di consueto;

• utilizzare l'IDE di Arduino per Raspberry Pi: collegarsi a Randa tramite *ssh* (ad esempio utilizzando MobaXterm) e trasferirvi lo sketch sul filesystem (ad esempio nella cartella *home/pi/sketchbook/*); in seguito lo si può aprire e compilare tramite la IDE di Arduino "locale".

In quest'ultimo caso, Arduino viene "visto" sulla porta seriale /dev/ttyS0 (l'unica disponibile).

In tutti i casi, se non l'avete mai fatto dovrete anche scaricare la libreria OneWire (necessaria per la gestione del sensore di temperatura DS18B20) e copiarla sotto la cartella libraries d'installazione dell'IDE (se in Windows) oppure in /usr/share/arduino/libraries (se su Raspberry Pi). Le altre librerie del progetto sono invece già comprese con l'IDE per entrambe le piattaforme.

Notare che l'IDE di Arduino di Raspberry Pi dovrebbe essere incluso se vi procurate una SD-card con il sistema RandA già installato; altrimenti, accertatevi che la scheda sia collegata in Internet durante l'installazione del pacchetto RandA dato che gli strumenti di Arduino vengono scaricati on-line. Per verificarlo, potete digitare su una console della Raspberry Pi il comando:
ping 8.8.8.8 ove 8.8.8.8 è l'indirizzo IPv4 del server DNS primario di Google. Se l'esito è positivo, la scheda Raspberry Pi "vede" Internet.

Da alcune prove eseguite, consigliamo di utilizzare il primo sistema: la IDE su Raspberry è decisamente più lenta, soprattutto nella compilazione, ma d'altra parte la potenza di calcolo non può essere paragonata a quella di un PC!

Dopo aver programmato lo shield con lo sketch ArdIR e dopo averlo inserito su RandA, si può effettuare subito un primo collaudo del sistema. Innanzitutto si può verificare che, dopo ogni reset (oppure ogni programmazione) il LED rosso lampeggia tre volte, ad indicare che lo sketch è caricato e pronto all'uso. Quindi pigiare P2 sino a veder lampeggiare due volte il LED rosso per entrare nella modalità locale di funzionamento. Successivamente pigiamo P1 per entrare in modalità di acquisizione, ed il LED rosso si accenderà fisso: puntiamo quindi il telecomando verso il ricevitore IR1 ad una distanza

di 5-10cm e ne azioniamo il comando che desideriamo sia appreso dalla nostra scheda (esempio, cambio canale del televisore).
Il LED rosso dovrebbe spegnersi, ed accendersi invece quello verde a conferma dell'avvenuta operazione. In caso contrario, ripetere l'operazione. Se nuovamente non andasse a buon fine bisogna verificare che il telecomando funzioni e che la frequenza di quest'ultimo sia di 36-38 kHz; nel dubbio potete provare con un altro telecomando.
Ammesso che l'operazione sia andata a buon fine, dopo cinque secondi il LED verde si spegnerà e torniamo nello stato di attesa: proviamo quindi a trasmettere il comando appena acquisito verso l'apparecchio relativo al telecomando che avevamo utilizzato, avendo cura che i LED all'infrarosso LD3-5 siano rivolti verso l'apparecchio ad una distanza possibilmente non superiore al metro (almeno per la prima prova). Pigiamo quindi P2 sino a vedere i due lampeggi del LED rosso, e quindi pigiamo nuovamente P2: il LED rosso produrrà ulteriori tre lampeggi al termine dei quali verrà effettuata la trasmissione verso l'apparecchio. Notare che contestualmente il LED blu dovrà accendersi brevemente. Se l'apparecchio "di prova" ha eseguito il comando che avevamo fatto acquisire, il collaudo può ritenersi concluso.
N.B. Dopo aver effettuato la programmazione del telecomando inserire il JUMPER JP1 per impedire che altre programmazioni sovrascrivano i dati precedentemente inseriti. Per aggiungere un altro telecomando è necessario rimuovere temporaneamente il JUMPER JP1 per poi rimetterlo in posizione al termine della programmazione.

Spostiamo il controllo sul web

L'ultimo passo da implementare è quello di dare l'accesso alla nostra scheda da remoto: come già visto, sfruttiamo il web server di RandA (Tomcat) per rendere visibili le pagine HTML che vogliamo rendere pubblicabili sui client remoti. Di conseguenza, a livello di installazione bisogna solo preoccuparsi di depositarle nella cartella:
/home/apache-tomcat-7.0.47/webapps/RandA/
da dove Tomcat la preleverà per trasferirla al client che la richiede.
La pagina creata appositamente per questa applicazione, che abbiamo chiamato IRConsole.html, è visibile in **Fig. 6**: lo stile grafico è stato impostato sulla falsa riga delle altre pagine web già realizzate per il controllo remoto della board RandA; inoltre abbiamo

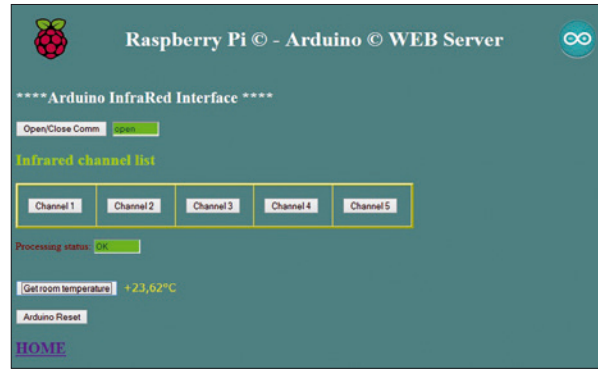


***Fig. 6** - La pagina web della nostra applicazione.*

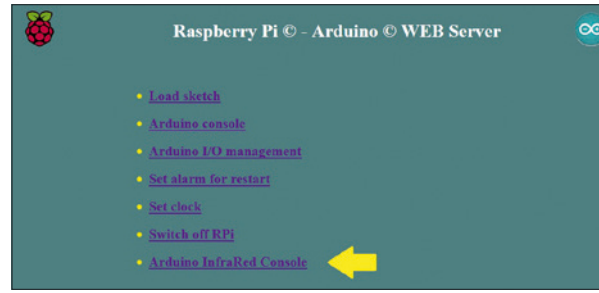


***Fig. 7** - La pagina principale.*

Come aggiungere i canali al telecomando

Il progetto è predisposto per acquisire fino a 127 differenti canali, quindi più di quello che servirà nella pratica; potete comunque utilizzare e configurare solo quelli di cui avete bisogno.
Per modificare il numero di pulsanti mostrati nella pagina web, bisogna editare il file IRConsole.html ed agire sul codice compreso tra i commenti *<!--Channel list begin -->* e *<!-- Channel list end -->*. Ad esempio, per incrementare il numero dei pulsanti in fila a quelli presenti, si può aggiungere il seguente codice prima del tag </tr>:

```
<td width="6%">
<p>      <input id="ChXX" name="NomeXX"
type="button" value="Nome pulsante" /></p>
</td>
```

Invece per creare un'altra fila sottostante bisogna aggiungere una riga nella tabella che li contiene: è sufficiente ricopiare tutto il codice tra i tags <tr> e </tr>, compresi.
Se però non vi sentite sicuri di quello che fate, consultate prima una guida su HTML e, nello specifico, la parte riguardante la disposizione degli elementi su tabelle (es. *http://www.html.it/pag/16055/tabella-struttura-di-base*): farete prima ed il risultato sarà migliore. I più esigenti vorranno magari alterare anche la grafica per creare un layout corrispondente ad un telecomando "vero"...
In ogni caso, per ogni pulsante che inserite dovrete:

1) assegnare un identificatore ed un nome univoci (campi "id" e "name" rispettivamente); in particolare, l'identificatore dev'essere nel formato ChX, ove X è un numero progressivo (es. Ch6, Ch7, Ch8,...).
2) aggiungere la seguente riga nella sezione di codice javascript (in alto nel file), nella funzione window.onload, dopo le altre righe simili:
   `el=document.getElementById("<ChX>"); el.onclick=setPin;`
   dove `<ChX>` è l'identificatore del nuovo pulsante come descritto al punto 1

Come ultima cosa, sempre nella sezione javascript, dovrete aggiornare la variabile NCHANNELS con il numero totale di pulsanti dei canali:
`var NCHANNELS = <numero totale pulsanti canali>;`
Dal "lato Arduino", bisogna infine ricordarsi di aggiornare con tale numero anche l'omonima costante dichiarata nello shield:
`#define NCHANNELS <numero totale pulsanti canali>`
e quindi riprogrammare la scheda attraverso l'IDE.

aggiunto il suo riferimento anche su *index.html* (che permette di aprire le altre pagine con le applicazioni per la gestione di RandA da remoto) in modo che ospiti anche il collegamento alla nostra *IRConsole.html* (**Fig. 7**).
Una volta caricate le pagine nella cartella sopra citata, possiamo verificarne subito il corretto funzionamento con un PC connesso alla rete locale di RandA, digitando nella barra degli indirizzi del browser: *http://<Indirizzo IP RandA>/RandA/IRConsole.html*, dove <Indirizzo IP> è l'indirizzo IPv4 della scheda RandA nella LAN. Dovrebbe quindi venire caricata la pagina di **Fig. 6**, che come già detto corrisponde al file *IRConsole.html* precedentemente caricato.
Su questa pagina si può notare subito il primo tasto, relativo alla comunicazione seriale: esso serve ad attivare il canale di comunicazione visto in **Fig. 5**, dalla servlet alla seriale /dev/ttyS0 (e quindi ad Arduino). Pertanto, è la prima cosa da attivare ed attendere che la finestra a fianco riporti la scritta "open". Vale la pena ricordare che la seriale di Arduino viene anche utilizzata per la programmazione, quindi se abbiamo attivato la comunicazione con la pagina web non possiamo programmare la scheda, e viceversa. Allo scopo di non lasciare la seriale inutilmente impegnata sarebbe anche buona norma, prima di lasciare la pagina, ricordarsi di spegnere l'interfaccia agendo nuovamente sul tastino.
Subito sotto abbiamo collocato la lista dei tasti relativi ai vari canali del nostro telecomando virtuale; a tale proposito va detto che è possibile aggiungerne altri (fino a 127) e magari personalizzare i loro nomi, in modo che descrivano la funzionalità associata (ad esempio, "Accendi climatizzatore" al posto di "Channel 1"). Per fare ciò bisogna editare il file *IRConsole.html*, e modificare il codice compreso tra i commenti *<!--Channel list begin --> e <!-- Channel list end -->*. Per cambiarne il nome, è sufficiente cambiare il relativo campo "value". La procedura necessaria ad aggiungere dei pulsanti viene per chiarezza descritta a parte in questo stesso articolo (nel riquadro "Come aggiungere i canali al nostro telecomando").
Dopo la lista dei pulsanti-canale, vi è la casellina "Processing status" che riporta l'esito della comunicazione con RandA: per qualsiasi comando impartito, visualizza "wait" durante l'attesa di risposta, "fail" se il comando non è stato eseguito oppure se Arduino non ha risposto entro un certo periodo (vi è un timeout nella servlet SerialIO), oppure "OK" se il comando è stato eseguito con successo. Scendendo nella pagina incontriamo i due pulsanti descritti qui di seguito.
• *Get room temperature*: ad ogni "click" aggiorna il valore di temperatura rilevato dal sensore DS18B20 presente nello shield; questa informazione può essere utile se si utilizza ArdIR per il controllo del climatizzatore domestico. Tenere presente che alla prima interrogazione, non avendo ancora conversioni valide, potrebbe visualizzare il valore di default (85°C): non spaventatevi, ma cliccate di nuovo.
• *Arduino reset*: forza un reset del micro di Arduino. In caso di (improbabile) necessità, permette di fare "ripartire" lo sketch.
Il link "HOME", infine, mutuato dalle altre pagine web di RandA, consente di tornare alla pagina iniziale di **Fig. 7**.

L'articolo completo del progetto è pubblicato su: Elettronica In n. 197

A tutti i residenti nell'Unione Europea. Importanti informazioni ambientali relative a questo prodotto

Questo simbolo riportato sul prodotto o sull'imballaggio, indica che è vietato smaltire il prodotto nell'ambiente al termine del suo ciclo vitale in quanto può essere nocivo per l'ambiente stesso. Non smaltire il prodotto (o le pile, se utilizzate) come rifiuto urbano indifferenziato; dovrebbe essere smaltito da un'impresa specializzata nel riciclaggio. Per informazioni più dettagliate circa il riciclaggio di questo prodotto, contattare l'ufficio comunale, il servizio locale di smaltimento rifiuti oppure il negozio presso il quale è stato effettuato l'acquisto.

Prodotto e distribuito da:
FUTURA GROUP SRL
Via Adige, 11 - 21013 - Gallarate (VA)
Tel. 0331-799775
Fax. 0331-792287
Web site: *www.futurashop.it*
Info tecniche: www.futurashop.it/Assistenza-Tecnica