

Shield Irrighino: centralina di irrigazione basata su Arduino Yún

(cod. FT1267K)

Lo shield Irrighino è una centralina di irrigazione basata su Arduino Yún e interfacciabile a una scheda a relè (max. 4) per il controllo di elettrovalvole. È in grado di gestire un numero configurabile di uscite, un sensore per la pioggia ed è programmabile tramite un'interfaccia web raggiungibile da qualsiasi PC o smartphone collegato alla rete domestica.

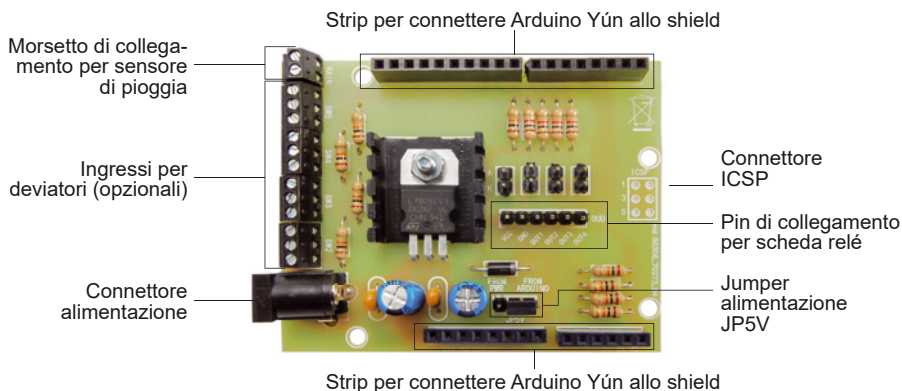


Schema elettrico

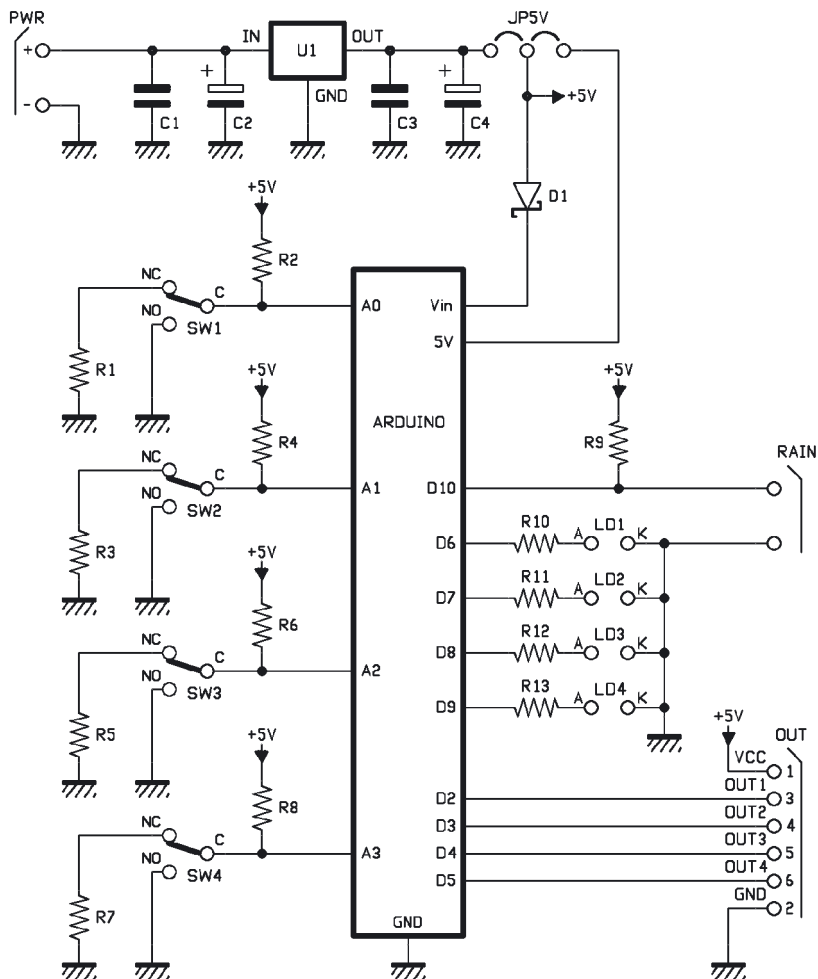
Affinché Irrighino sia in grado di gestire un impianto di irrigazione, è necessario collegarlo ad alcuni elementi esterni. Lo shield consente di gestire un impianto con un massimo

di quattro elettrovalvole, un sensore di pioggia e un pannello di comando esterno con quattro interruttori e quattro LED di stato. Lo schema di tale shield è composto da un

regolatore di tensione, indispensabile ad alimentare Arduino (tramite il contatto Vin) da un certo numero di uscite per LED e uscite per la gestione delle elettrovalvole, nonché da quattro



Schema elettrico



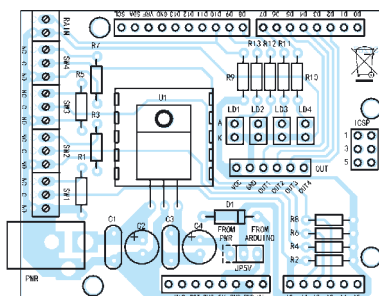
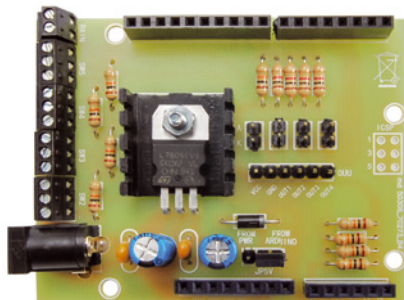
ingressi per i deviatori del pannello di comando, che sono opzionali.

PWR è il connettore jack di alimentazione, cui va for-

nita una tensione continua di valore compreso fra 8 e 15 volt, che viene filtrata dai condensatori C1 e C2 e stabilizzata a 5 volt dal

regolatore integrato U1. Il piedino OUT di quest'ultimo fornisce i propri 5 V riferiti a GND al lato sinistro del ponticello JP5V, che

Piano di montaggio



Elenco Componenti:

C1, C3: 100 nF ceramico
 C2: 100 μ F 25 VL
 elettrolitico
 C4: 100 μ F 25 VL
 elettrolitico
 R1 ÷ R9: 10 kohm
 R10 ÷ R13: 1 kohm
 U1: 7805
 D1: 1N5819

Varie:
 - Plug alimentazione
 - Morsetto 2 vie passo
 2,54mm
 - Morsetto 3 vie passo
 2,54mm (4 pz.)
 - Vite 10 mm 3 MA
 - Dado 3 MA
 - Dissipatore

- Strip M/F 6 vie
 - Strip M/F 8 vie (2 pz.)
 - Strip M/F 10 vie
 - Strip M/F 2x3 vie
 - Strip maschio 3 vie
 - Strip maschio 2 vie (4 pz.)
 - Strip maschio 6 vie
 - Jumper
 - Circuito stampato S1267

serve a decidere la fonte di alimentazione del sistema: se vogliamo alimentare Arduino Yún attraverso lo shield bisogna chiudere il jumper tra il centrale e il contatto di sinistra (in questo caso i 5V giungono dal diodo Schottky D1), mentre se vogliamo alimentare il tutto tramite la Yún, quest'ultima va alimentata con un proprio alimentatore e il jumper va chiuso a destra. In quest'ultimo caso non bisogna alimentare il jack PWR dello shield, perché lo shield stesso viene alimentato

dal 5V di Arduino, sempre tramite D1. Indipendentemente da quale sia la fonte di alimentazione, la linea dei 5 volt alimenta anche i resistori di pull-up R2, R4, R6, R8 degli ingressi destinati a deviatori ed R9, relativa all'ingresso riservato al sensore di pioggia RAIN. Le uscite fanno capo a OUT1, OUT2, OUT3 e OUT4 (rispettivamente D2, 3, 4, 5 di Arduino) e forniscono il livello logico alto quando devono essere aperte le elettrovalvole; sono pensate per comandare schede a relé,

che accettano segnali TTL e alimentano in proprio le bobine dei relé.

Collegamenti

Per comandare le elettrovalvole è necessario utilizzare dei relé esterni funzionanti a 5V e comandati dai pin di output di Arduino. Per questo scopo, lo shield prevede il connettore OUT, che riporta positivo d'alimentazione 5V, massa e quattro pin digitali di Arduino per l'attivazione di altrettanti relé. Il connettore è pensato per pilotare il modulo 4 relé 5Vdc com-

mercializzato da Futura Elettronica (cod. RELAY-4CH). Tale scheda è dotata di 4 relé con bobina a 5 V e scambio che consente di commutare carichi elettrici che funzionano con un massimo di 240Vca e assorbono fino a 10A (max); ciascun relé viene attivato portando l'ingresso di comando a livello logico alto (da 1,6 a 5 Vcc), quindi la scheda è ideale per attuare i segnali logici forniti dallo shield, anche perché ha un assorbimento, su ciascun input, di pochi milliampere, compatibile anche con le logiche CMOS e CMOS HC. Gli ingressi della scheda 4 relé sono tutti optoisolati, così che laddove

sia necessario è possibile separare galvanicamente la scheda dal circuito di comando; ogni optoisolatore ha l'anodo del diodo emettitore di ingresso (pin 1) collegato al rispettivo input tramite un resistore di limitazione della corrente e il catodo in comune con i catodi dei LED degli altri fotoaccoppiatori e connesso al contatto COM di ingresso. Se si desidera rinunciare all'isolamento, si devono ponticellare il contatto GND e il COM. I capi del secondario saranno collegati uno direttamente al comune delle elettrovalvole e uno al polo comune dei relé, come indicato nella **figura 1**. Se

disponete di una sola elettrovalvola, il collegamento è quello visibile in **Fig. 2**, dove trovate l'esempio di collegamento valido per un'elettrovalvola pilotata da una scheda relé singola.

Il sensore di pioggia è collegato all'ingresso RAIN, in condizioni di asciutto lascia che il pin D10 di Arduino assuma l'1 logico, mentre quando è bagnato, la resistenza dell'acqua fa partitore con R9 e D10 presenta una tensione tanto bassa da corrispondere allo zero logico.

Il sensore può essere collegarlo alla morsettiera etichettata come RAIN. È possibile dotare Irrighino

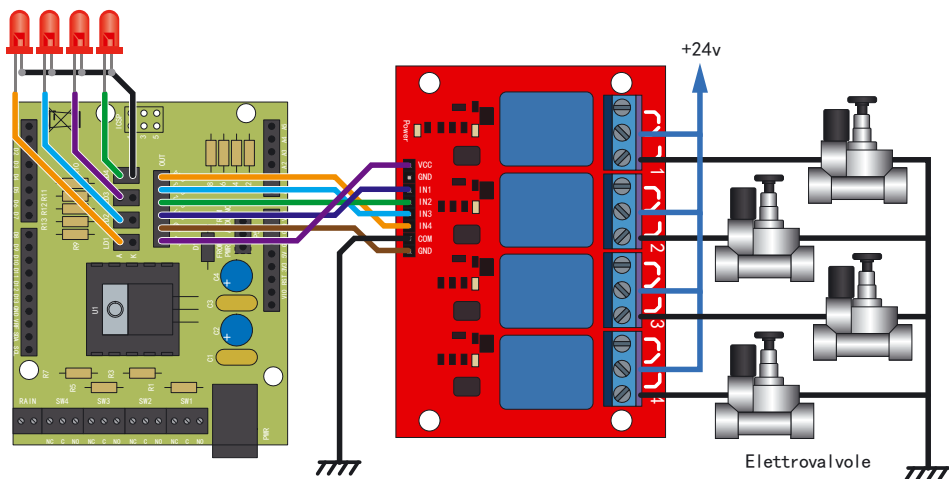
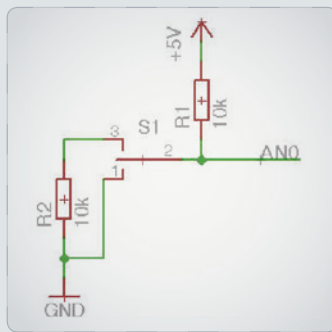


Fig.1. Lo schema di cablaggio dello shield con la scheda a 4 relé che vengono utilizzati per comandare fino a quattro elettrovalvole; lo schema è valido a prescindere dalla tensione di funzionamento delle elettrovalvole, che può essere da 12 a 220 Vca.

Tre posizioni, un solo pin

Per leggere lo stato di un interruttore a tre posizioni, è possibile usare un solo pin analogico di Arduino e due resistenze, collegandolo come mostrato nell'immagine in questo riquadro. Quando l'interruttore è in posizione centrale, il pin è collegato a 5V tramite R1 e il valore letto dall'istruzione `analogRead()` è vicino al massimo (1023). Quando l'interruttore è in posizione 3, il pin è collegato ad un partitore resistivo formato da R1 e R2; il valore letto sarà circa la metà (512). Infine quando l'interruttore è in posizione 1, il pin è collegato direttamente a massa e il valore letto è vicino a zero. Tale tecnica viene utilizzata nello shield utilizzato in questo progetto.



di un pannello di controllo fisico, per operare manualmente in locale senza bisogno di un computer: detto pannello prevede un interruttore a tre posizioni

(ON, AUTO, OFF) stabili e un LED di stato per ogni uscita. La posizione dell'interruttore viene letta tramite un pin analogico di Arduino Yún, come spiegato

nel box qui sopra, grazie al particolare collegamento di due resistenze che permettono di ottenere un valore di tensione per ciascuna posizione.

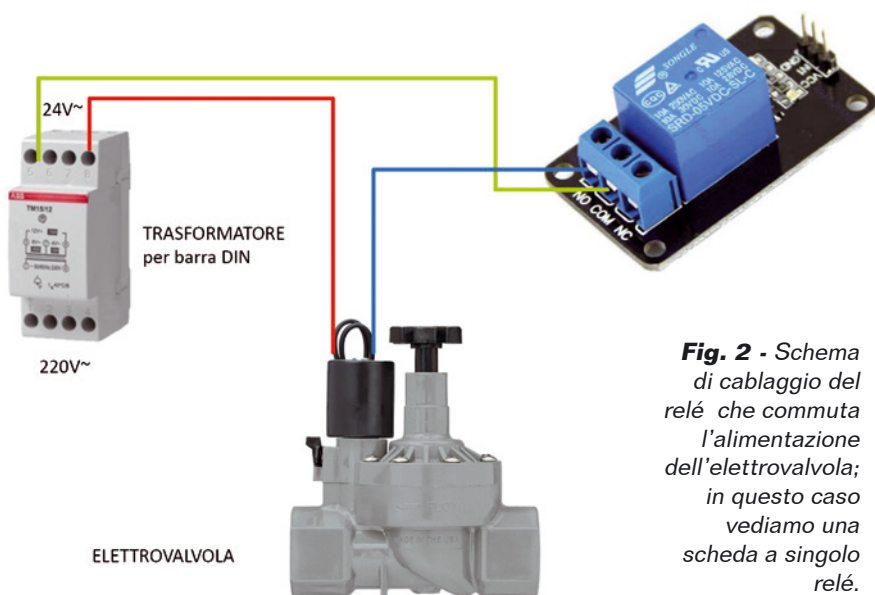
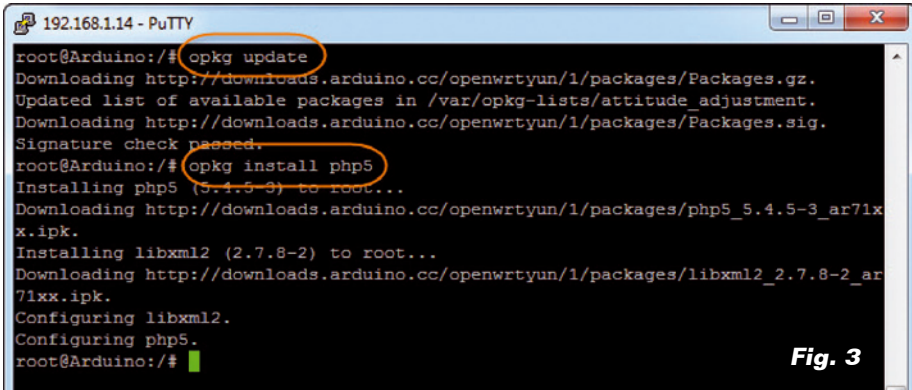


Fig. 2 - Schema di cablaggio del relé che commuta l'alimentazione dell'elettrovalvola; in questo caso vediamo una scheda a singolo relé.



```

192.168.1.14 - PuTTY
root@Arduino:/# opkg update
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/Packages.gz.
Updated list of available packages in /var/opkg-lists/attitude adjustment.
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/Packages.sig.
Signature check passed.
root@Arduino:/# opkg install php5
Installing php5 (5.4.5-3) to root...
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/php5_5.4.5-3_ar71x.ipk.
Installing libxml2 (2.7.8-2) to root...
Downloading http://downloads.arduino.cc/openwrtyun/1/packages/libxml2_2.7.8-2_ar71xx.ipk.
Configuring libxml2.
Configuring php5.
root@Arduino:/#

```

Fig. 3

Ogni interruttore del pannello serve, dunque, a forzare manualmente una condizione all'uscita corrispondente e quindi nell'elettrovalvola corrispondente; la condizione assunta viene segnalata dal relativo LED. SW1 si riferisce e OUT1, il cui stato è segnalato da LD1, SW2 a OUT e LD2 e via di seguito.

Portando l'interruttore sulla posizione di ON, la relativa uscita viene attivata, mentre portandolo sulla posizione di OFF l'uscita viene disattivata. Lasciandolo nella posizione intermedia (AUTO) l'uscita segue la programmazione impostata. Il pannello ha priorità rispetto all'interfaccia grafica, quindi impostando sul pannello un'uscita ad OFF essa risulterà disabilitata sull'interfaccia grafica e non sarà possibile modificare lo stato.

Lo shield prevede già le morsettiere cui collegare i quattro interruttori e i pin cui connettere i LED; per questi ultimi è necessario fare attenzione a collegare correttamente l'anodo e il catodo ai pin come indicato dalla serigrafia (l'anodo corrisponde normalmente al terminale più lungo).

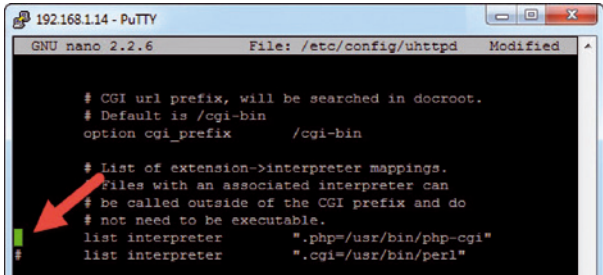
Lo shield è anche dotato di un regolatore di tensione (5V) che consente di alimentare tutti i componenti esterni in maniera autonoma rispetto ad Arduino.

Tramite il jumper JP5V è possibile scegliere se utilizzare l'alimentazione proveniente da Arduino o un alimentatore esterno (9-12V corrente continua) collegato direttamente allo shield.

Preparazione di Arduino

Per poter utilizzare Irrighino, è necessario per prima cosa installare alcuni pacchetti software nel sistema operativo Linux.

Utilizzando un client SSH



```

GNU nano 2.2.6 File: /etc/config/uhttpd Modified
# CGI url prefix, will be searched in docroot.
# Default is /cgi-bin
option cgi_prefix /cgi-bin

# List of extension->interpreter mappings.
# Files with an associated interpreter can
# be called outside of the CGI prefix and do
# not need to be executable.
list interpreter ".php=/usr/bin/php-cgi"
list interpreter ".cgi=/usr/bin/perl"
#

```

Fig. 4 - Cancellazione del carattere #.

(se siete sotto Windows potete utilizzare PuTTY, su Linux e MacOS esiste il comando ssh) colleghiamoci in console ad Arduino Yún e inseriamo le credenziali di accesso (per impostazione predefinita, root è l'utente e arduino la password). Assicuriamoci inoltre che lo Yún sia connesso ad Internet.

Per prima cosa eseguiamo il comando per aggiornare la lista dei pacchetti disponibili per l'installazione, che è:

```
opkg update
```

opkg è il package manager della nostra distribuzione Linux; tramite que-

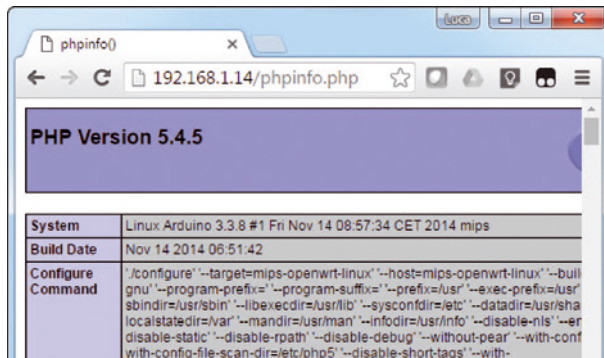


Fig. 5 - Pagina informazioni dell'interprete php.

sto comando è possibile installare o disinstallare applicazioni e programmi. Per installare l'interprete php (versione 5) possiamo quindi digitare il comando `opkg install`

php5. Il gestore pacchetti scaricherà l'ultima versione disponibile –insieme ad eventuali prerequisiti– e la installerà nel sistema (Fig. 3).

Utilizziamo sempre opkg

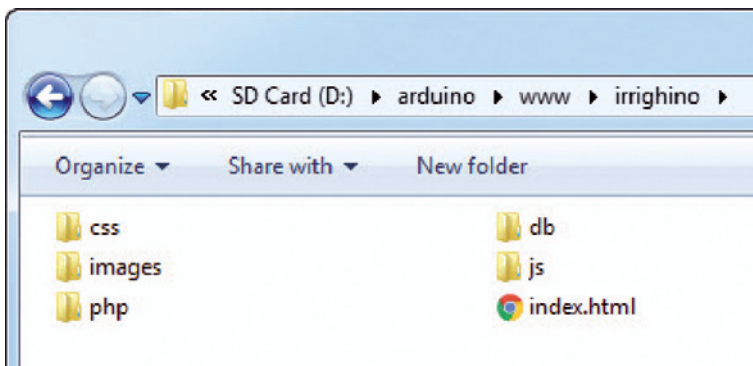
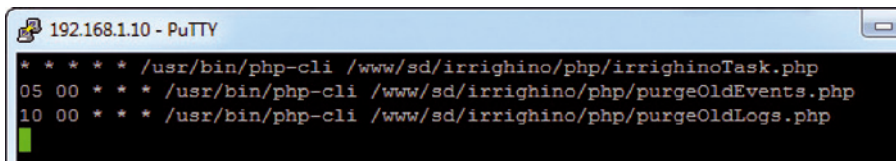


Fig. 6

Fig. 7



per installare tutti gli altri pacchetti necessari ad ir-righino:

```
opkg install php5-
cgi php5-cli php5-
mod-curl php5-mod-
json php5-mod-pdo
php5-mod-pdo-sqlite
zoneinfo-core zo-
```

neinfo-europe

Attiviamo ora l'inteprete php all'interno del server web **uhttpd** modificando il suo file di configurazione con il comando:

```
nano /etc/config/
```

uhttpd

Dobbiamo cancellare il carattere di cancelletto (#) all'inizio della riga relativa all'estensione **.php** come illustrato dalla **Fig. 4**. Premiamo contemporaneamente CTRL e X, quindi digitiamo Y per confermare la modifica effettuata. Infine riavviamo il server web con il comando:

```
/etc/init.d/uhttpd
restart
```

Per verificare che ora il nostro Yún sia in grado di eseguire pagine php, possiamo crearne una di esempio digitando i se-

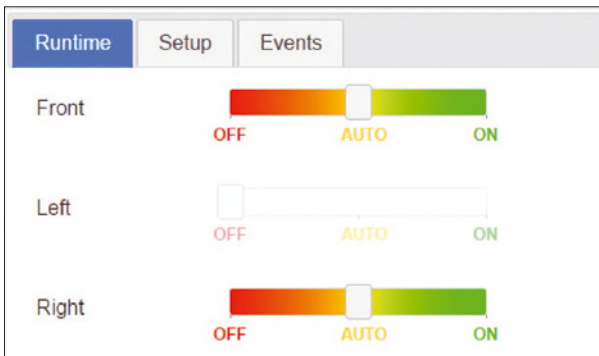


Fig. 8 L'interfaccia web.

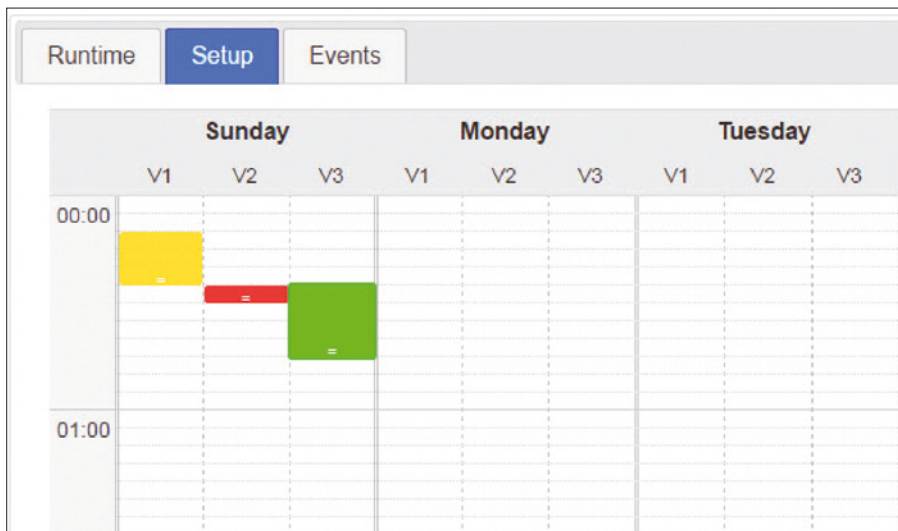


Fig. 9


```

30 // ----- Output configuration -----
31
32 define ("OUTPUTS_NUMBER", 2)
33
34 $outputs = array();
35 $outputs[0]["name"] = "Giardino";
36 $outputs[0]["baseColor"] = "#FFD800";
37 $outputs[0]["borderColor"] = "#E5BF00";
38
39 $outputs[1]["name"] = "Fiori balcone";
40 $outputs[1]["baseColor"] = "#FF0000";
41 $outputs[1]["borderColor"] = "#E00000";

```

Fig. 10

guenti comandi:

```

cd /www/
echo "<?=phpinfo()";
?>" >phpinfo.php

```

Se ora apriamo un browser sul nostro computer e digitiamo nella barra degli indirizzi l'indirizzo IP di Arduino Yún seguito da `/phpinfo.php`, dovremmo vedere la pagina di informazioni dell'interprete php come in Fig. 5.

Installazione

Scarichiamo ora l'ultima versione del software dal repository Github. Collegiamoci all'indirizzo <https://github.com/lucadentella/irrishino> e clicchiamo sul pulsante **Download ZIP**: dopo qualche istante avremo sul nostro PC un file compresso che contiene tutto il necessario per installare Irrighino.

Prendiamo una scheda

micro SD e inseriamola nel nostro PC; dobbiamo creare in essa una cartella chiamata **arduino** e, all'interno di quest'ultima, una cartella chiamata **www**. Copiamo ora la cartella **Website**, contenuta nell'archivio scaricato da Github, all'interno della cartella **www**; infine rinominiamo la cartella **Website** in **irrishino**. Se tutti i passaggi sono stati effettuati correttamente, il contenuto della scheda micro SD sarà quello di Fig. 6 e potremo inserire la scheda di memoria nell'apposito slot del nostro Arduino.

Sempre collegati ad Arduino via SSH, digitiamo il comando `crontab -e` e per modificare la configurazione dello schedulatore e, dopo aver premuto il tasto "i", inseriamo tre righe, come indicato in Fig 7. Per salvare la configurazione premiamo ESC, quindi di-

gitiamo `:wq` e premiamo INVIO. Le tre righe inserite attivano ogni minuto (***) lo script che controlla se vi sono schedulazioni da attivare e una volta al giorno (alle 00:05 e 00:10) gli script che puliscono gli eventi e i log.

Adesso ci resta solo un ultimo passo da fare: caricare lo sketch, contenuto nella cartella **Sketch** all'interno dell'archivio di Github. Utilizzando quindi l'IDE di Arduino, apriamo lo sketch *irrishino.ino* ed effettuiamo l'*upload* sul nostro Yún.

Utilizzo di irrishino

Da PC: aprite il browser Internet e nella barra degli indirizzi puntate l'indirizzo web `http://<ip_arduino_yun>/sd/irrishino`; dalla pagina che si aprirà è possibile accedere alla schermata di gestione della nostra centralina.

```

irrighino  config.h  debug_functions.c

// outputs, digital PINs connected to the relays
#define OUT_1      2
#define OUT_2      3
#define OUT_3      4

// leds, digital PINs that directly drive the panel leds
#define LED_1      7
#define LED_2      6
#define LED_3      5

// switches, analog PINs that read the 3-way panel switches
#define SWITCH_1   0
#define SWITCH_2   1
#define SWITCH_3   2

// rain sensor, digital PIN connected to the rain sensor
// (basically a 2-way switch, normally closed)
#define RAIN_SENSOR 8

```

Fig. 11

L'interfaccia è divisa in tre tab: **Runtime**, da dove è possibile comandare manualmente le uscite, **Setup**, dalla quale è possibile programmare le attivazioni su base settimanale, ed **Events**, dove viene visualizzato l'elenco degli eventi (attivazioni, modifiche alle configurazioni...) più recenti.

Ogni uscita può essere impostata in tre stati: **ON** - l'uscita viene attivata; **OFF** - l'uscita viene disattivata o **AUTO** - l'uscita segue la programmazione impostata. Se l'uscita viene configurata tramite il pannello esterno, i relativi comandi

sull'interfaccia web saranno disattivati: il pannello ha infatti priorità maggiore (vedi Fig. 8).

La fase di programmazione è molto semplice: gli eventi sono raccolti in una tabella nelle cui colonne sono rappresentati i giorni della settimana e le singole uscite (V1...). Ogni riga rappresenta un intervallo di 5 minuti: cliccando con il mouse su una cella è possibile definire un evento di attivazione. Tramite il mouse, l'evento può essere ridimensionato (per modificarne la durata) e spostato, anche su un'altra uscita. Cliccando su

un evento, questo viene selezionato e attorno ad esso viene visualizzato un bordo più scuro (vedi Fig. 9) che ne evidenzia la selezione.

È possibile selezionare più eventi o deselectionarli facendo nuovamente clic su di essi mentre si preme il tasto Shift.

Infine premendo il tasto CANC è possibile eliminare tutti gli eventi selezionati.

Personalizzazione di irrighino

Sebbene il progetto che potete scaricare da Github sia configurato per funzio-

nare *out of the box* insieme allo shield che abbiamo realizzato, Irrighino è stato pensato per essere facilmente personalizzabile in modo da poterlo adattare ad ogni necessità.

Apriamo il file **include.php**, presente all'interno della cartella php, è possibile modificare il numero di uscite, il loro nome e il colore assegnato, come illustrato in **Fig. 10**. È anche possibile modificare il comportamento in caso di pioggia (stop immediato delle schedulazioni in corso e/o nessuna nuova attivazione) o disabilitarlo completamente nel caso non sia previsto nel vostro impianto.

Anche i collegamenti tra

Arduino e i dispositivi esterni sono configurabili modificando lo sketch. Tutto il *mapping* tra i dispositivi e i pin cui sono collegati è stato definito in un apposito file, chiamato **config.h**, che riportiamo nel Listato 1. È quindi possibile modificare i pin cui sono collegati relé, interruttori, LED e sensore di pioggia semplicemente indicando, sempre nel file **config.h**, i pin di Arduino desiderati (vedere **Fig. 11**). Ricordate, nel caso appliciate ad Arduino altri shield oltre quello di Irrighino, di verificare sempre -prima di assegnare gli I/O in **config.h**- che non siano già utilizzati dalle rispettive applicazioni.

Accertatevi sempre che il numero di uscite configurato nell'interfaccia web corrisponda al numero di output (OUT_x) definiti nello sketch e ricordatevi anche di aggiornare lo sketch principale.

Infine, se non prevedete l'installazione di un pannello di controllo esterno è possibile inibire le funzioni associate al controllo manuale mettendo a **false** la costante USE_SWITCHES.

L'articolo completo del progetto è stato pubblicato su:
Elettronica In n. 207

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.

Distribuito da:

FUTURA GROUP SRL

Via Adige, 11 - 21013 Gallarate (VA) Tel. 0331-799775 Fax. 0331-792287

web site: www.futurashop.it

supporto tecnico: www.futurashop.it/Assistenza-Tecnica