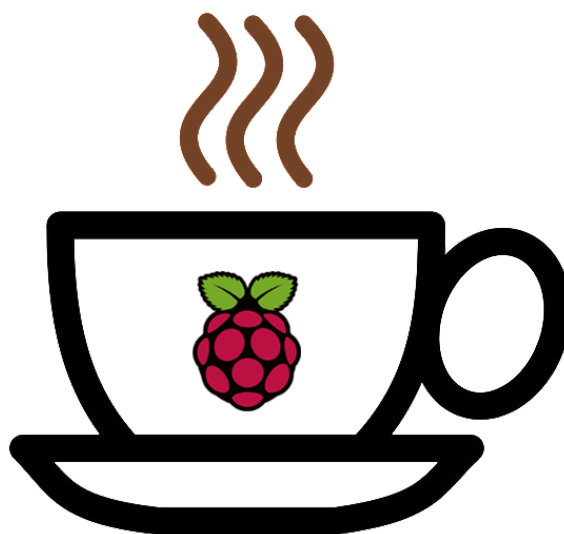


piCoffee
—



Descrizione

L'obiettivo di questo progetto è quello di riciclare una vecchia macchina del caffè per controllarla attraverso lo smartphone. Questo è possibile modificando lo schema elettrico all'interno del dispositivo, collegandolo ad un Raspberry che farà da server web e rimarrà in attesa delle nostre indicazioni.

I vantaggi che porta questo tipo di modifica sono:

1. Un risparmio energetico: la macchina in questione è abbastanza datata e, lasciandola accesa, consumerebbe molta corrente. Con la modifica effettuata, viene data la possibilità di fare il caffè con la massima efficienza e in tempi brevi, in quanto la sequenza di operazioni da eseguire è automatizzata.
2. La pianificazione della preparazione di un caffè: tramite **crontab** e l'interfaccia fornita, l'utente può programmare l'erogazione della miscela.

Strumenti e ambienti di sviluppo utilizzati

- Relè relay 5V - 230V
- Relè monostabile
- Raspberry Pi 3 Model B
- Saeco Nina cappuccino Type Sin026X
- Kit di cavi Elgoo
- Atom
 - IDE per HTML e PHP
- Putty
- FileZilla
- Bash Debian

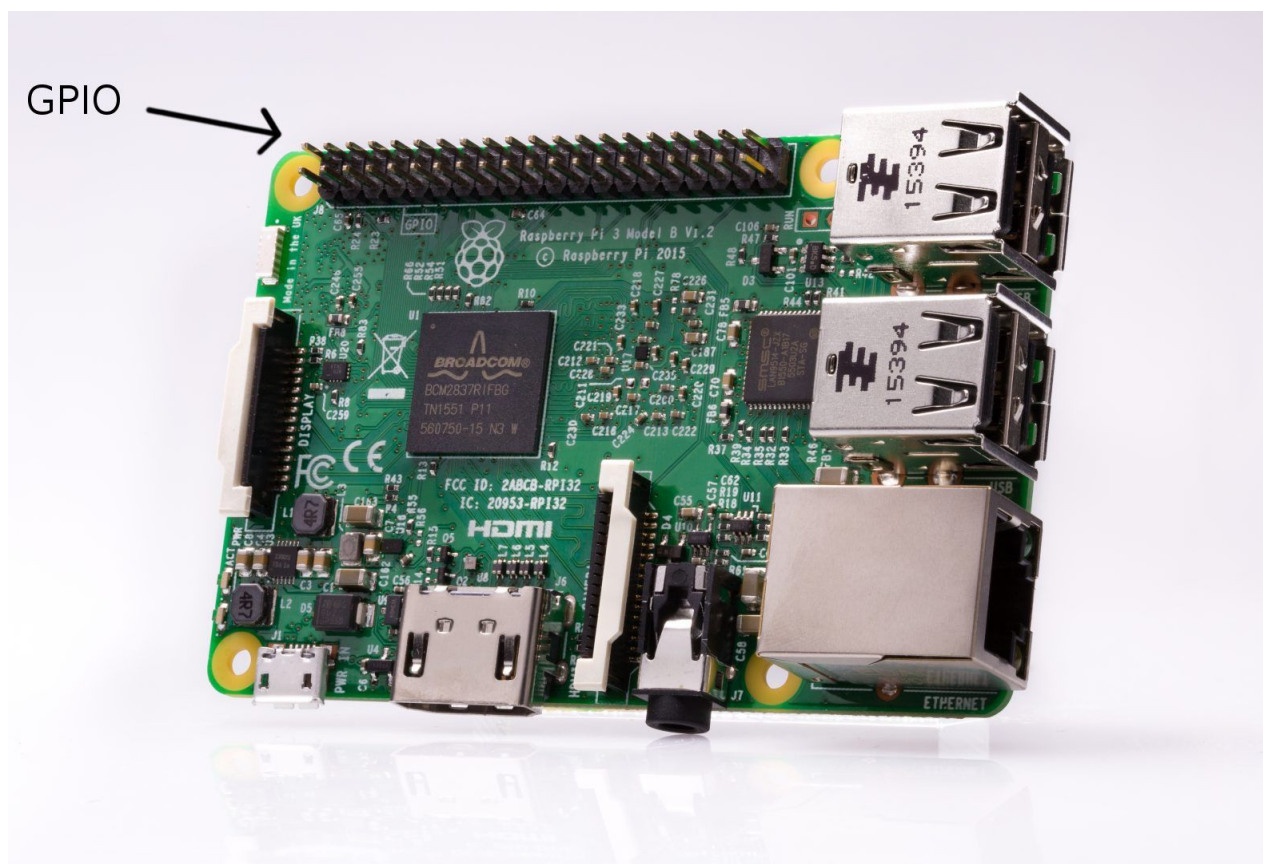
Cos'è il Raspberry Pi

Il Raspberry Pi è un single-board computer sviluppato nel Regno Unito dalla Raspberry Pi Foundation. In esso possono essere installati diversi OS, il più comune è Raspbian, un versione di Debian modificata appositamente per il dispositivo.

L'ultima versione uscita nel marzo del 2018 ha un costo di soli € 37.

I suoi utilizzi possono essere i più disparati: come computer economici nei paesi di sviluppo, come mediacenter, o in generale, per progetti fai-da-te. Nel mio caso servirà da centro di comando: riceverà le richieste provenienti dal dispositivo connesso e trasmetterà corrente alla macchina del caffè, se necessario.

Lo specifico hardware che mette in comunicazione elettronica e informatica è la riga di PIN GPIO (General Purpose Input/Output). Infatti i questi tipi di PIN possono essere impostati in modo da inviare corrente o riceverla. L'invio verrà utilizzato per accendere la macchina, mentre la ricezione per capire quando la caldaia è sufficientemente calda.



Progettazione Elettrica

Funzionamento normale

La macchina comincia a fare il caffè quando la manopola centrale viene girata verso destra. Dietro di essa, infatti, c'è uno switch che attiva il circuito e fa partire la pompa dell'acqua, che passerà per la caldaia e infine nel portafiltro, dove avverrà l'infusione. Un led sopra la manopola mostra quando la caldaia è abbastanza calda per produrre il caffè. La spia si illumina quando il termostato raggiunge la temperatura utile.

Funzionamento modificato

Alimentazione

L'alimentazione avviene tramite due spine elettriche. Una fornirà corrente al Raspberry e una alla macchina del caffè. La macchina non è sempre accesa, ma verrà alimentata solo quando lo decide il Raspberry. La fase elettrica viene normalmente interrotta dal *computerino*, ma quando si decide di accenderla, il PIN fornirà tensione e chiuderà il circuito, facendo partire il dispositivo. Non è stato possibile far alimentare il tutto da una sola spina perché il mini-computer non sarebbe in grado di fornire la corrente necessaria alla macchina del caffè, neanche con il supporto del relè.

Pompa dell'acqua

La pompa dell'acqua deve funzionare solo quando la caldaia è calda. Il PIN 2 è collegato al relè, che a sua volta è collegato alla pompa, rimpiazzando la manopola. Quest'ultima viene attivata con il comando `gpio write 2 0` e disattivata da `gpio write 2 1`. Tra un comando e l'altro viene usata la funzione `sleep(durata_in_secondi)` che serve a fermare l'esecuzione del codice PHP e di conseguenza ad erogare la giusta quantità di caffè, poiché il comando di spegnimento non viene eseguito subito, ma solo dopo il tempo definito da *durata*, che è stato precedentemente scelto dall'interfaccia.

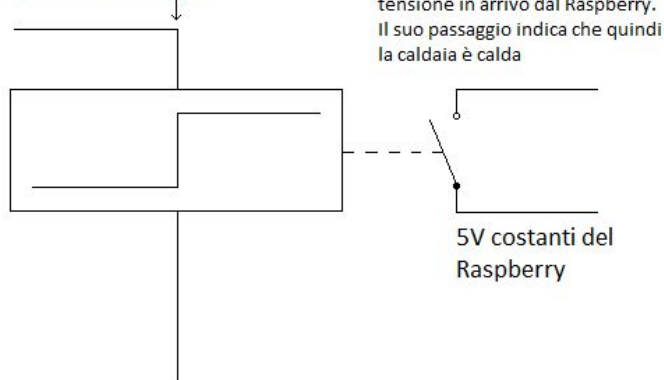
Caldaia/Termostato

La caldaia è controllata da un termostato e quando quest'ultimo rileva la temperatura adatta, accende un led vicino alla manopola. Far ricevere al Raspberry il segnale non è stato facile, visto che non può ricevere corrente a 230V, ma solo a 5V.

Per ovviare al problema, ho collegato al Raspberry un secondo relè. Da un lato ho connesso il cavo che faceva illuminare la spia. Quando da qui passerà tensione, il relè si ecciterà e chiuderà il circuito, accendendo il led. Allo stesso tempo, chiudendosi, lascerà passare la corrente proveniente dal PIN 0, sempre attivo e adibito solo a questa mansione; in questo modo l'energia elettrica andrà direttamente al PIN 1 che è in ascolto.

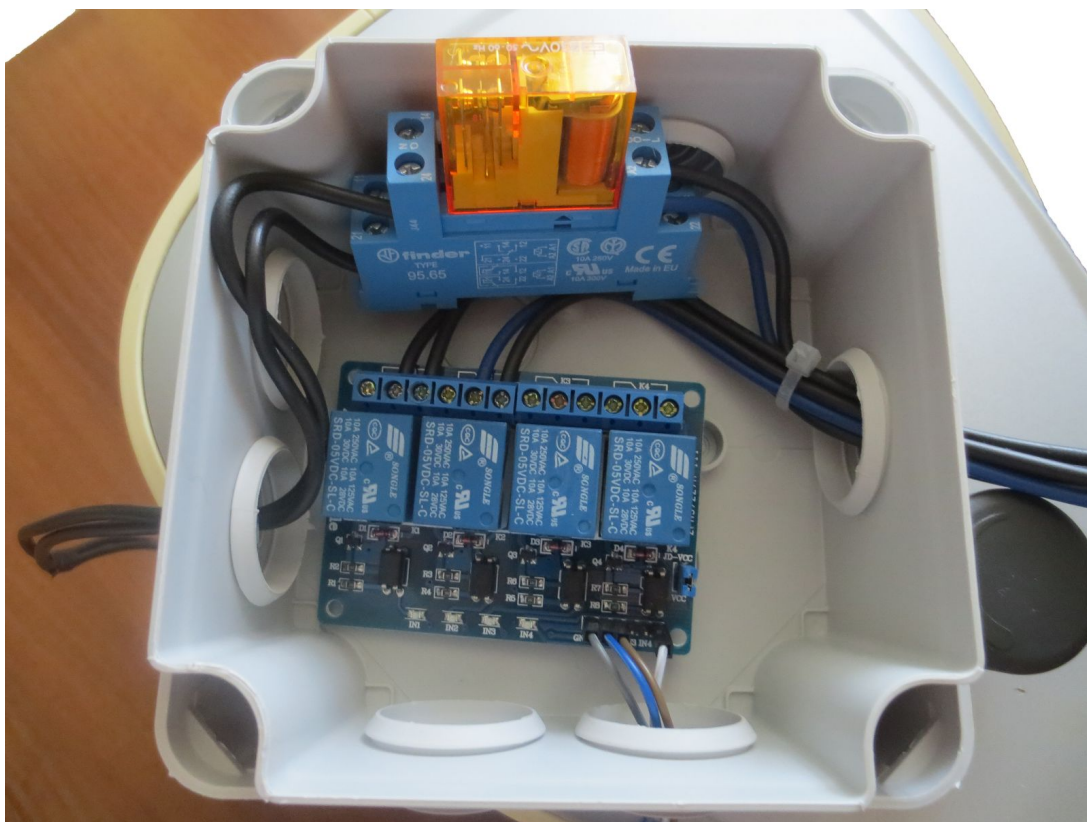
Quando il PIN 1 riceve 1, allora la caldaia è pronta.

la spia luminosa è collegata in parallelo al relè. Quando si accende passa corrente anche qui



Manopola

La nuova funzione della manopola riguarda la sicurezza. Girandola verso destra, la macchina del caffè sarà disponibile a tutte le nostre richieste. Se per qualche motivo dovessimo incappare in qualche malfunzionamento (Es. il dispositivo non smette di fare caffè, la caldaia si scalda all'infinito, ecc), possiamo posizionare la manopola al centro. In questo status qualsiasi azione in corso viene ignorata, ad esempio l'erogazione del caffè. Nel frattempo il codice proseguirà la sua esecuzione, quindi o si aspetta la sua fine e si riposiziona la manopola, o, nel peggiore dei casi, si staccano le spine. Fisicamente la macchina del caffè è come se si spegnesse, allo stesso modo di quando veniva spenta dal tasto posto sul lato destro del rivestimento.



Relè in posizione

Progettazione Informatica

WiringPi

WiringPi è una libreria di accesso a PIN GPIO scritta appositamente per i vari modelli di Raspberry. Il software include una serie di comandi eseguibili da shell (**gpio**) che permettono di controllare i PIN GPIO.

Es.

```
gpio mode 1 out #Il pin numero 1 è impostato in modalità output.
```

```
gpio write 1 1 #Il pin numero 1 è in tensione.
```

Al boot del Raspberry vengono eseguiti dei comandi che servono a impostare i PIN in maniera corretta e renderli pronti all'uso da parte delle pagine PHP. Il file che permette questo è `rc.local` che si trova nella cartella `/etc/`.

Sequenza di comandi:

```
gpio mode 0 out #Imposta il PIN 0 in modalità output  
gpio write 0 1 #Mette il PIN 0 in tensione  
gpio mode 3 in #Imposta il PIN 3 in modalità input  
gpio mode 2 out #Imposta il PIN 2 in modalità output  
gpio write 2 1 #Mette il PIN 2 in tensione  
gpio mode 1 out #Imposta il PIN 1 in modalità output  
gpio write 1 1 #Mette il PIN 1 in tensione
```

Materialize

Materialize è un framework front-end responsive basato sul Material Design, lo stile progettato da Google. Esso può essere considerato una valida alternativa a Bootstrap, altro framework responsive. Entrambi, infatti, si basano sull'uso del sistema a griglia a 12 colonne.

Il framework fornisce un file CSS e un file JavaScript con stili e animazioni già pronti da utilizzare nelle proprie pagine.

Crontab

Crontab è un comando installabile nei sistemi operativi Linux che consente la pianificazione di comandi, che possono essere mandati in esecuzione periodicamente. Il singolo evento nella crontab viene chiamato cronjob. Il mio intento è quello di dare la possibilità all'utente di programmare l'uscita del caffè. Perché ciò accada è stato indispensabile modificare i privilegi dell'utente *www-data* per permettergli di lanciare il comando *php-cgi*, che è necessario per eseguire una pagina PHP con parametri GET da linea di comando. In particolare, ho dovuto aggiungere *www-data* alla lista dei *sudoers*, cioè utenti che possono usare il comando *sudo* senza bisogno di inserire la password. La sicurezza in questo caso non è una priorità visto che la macchina del caffè è utilizzabile solo in rete locale.

```
* * * * * sudo php-cgi /var/www/html/comandi/on.php durata=22
```

Esempio di cronjob

```
.----- [m]inute: minuto (0 - 59)
| .----- [h]our: ora (0 - 23)
| | .----- [d]ay of month: giorno del mese (1 - 31)
| | | .----- [mon]th: mese (1 - 12) OPPURE jan,feb,mar,apr...
| | | | .---- [w]eek day: giorno della settimana (0 - 6) (domenica=0 o 7) OPPURE sun,mon,tue,wed,thu,fri,sat
| | | | |
* * * * * comando da eseguire
```

Com'è strutturato lo schema temporale in un comando cronjob


Per rendere lo schema temporale comprensibile a tutti, ho aggiunto una libreria chiamata **cRonstrue**, la quale mette a disposizione delle funzioni JavaScript che trasformano la stringa numerica in una stringa leggibile.

Es.

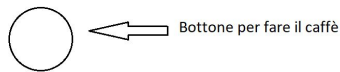
```
cronstrue.toString("0 23 * * 1,5", { locale: "it" })
//Il valore della stringa è: "Alle 23, il Lunedì e il Venerdì"
```


Funzionamento

L'utente avrà a disposizione 3 interfacce: caffè veloce, programmazione e opzioni

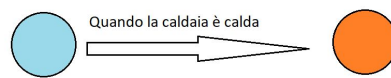
 piCoffee
Caffè veloce | Programma il caffè | Opzioni

Spiegazione del pulsante



Scegli il caffè che più ti piace

corto  lungo  Switch = 2 radio button




Spiegazione del pulsante

Caffè veloce:

- L'utente può premere il tasto per erogare il caffè in maniera efficiente, in quanto tutte le operazioni sono effettuate nella più rapida successione
- Attraverso uno switch può scegliere se fare un caffè espresso o un caffè lungo
- Una piccola icona fornisce lo stato della caldaia
 - Azzurra e ferma: caldaia fredda
 - Rossa e pulsante: caldaia calda

Caffè programmato:

 piCoffee
Caffè veloce | Programma il caffè | Opzioni

Scegli il caffè che più ti piace

corto ☐ lungo

Programma il caffè in un singolo giorno ☐ Programma il caffè ogni settimana nello stesso giorno

↓
☐ Lunedì
☐ Martedì
☐ Mercoledì
☐ Giovedì
☐ Venerdì
☐ Sabato
☐ Domenica

↓
Form che apre un calendario e fa scegliere la data

Form che apre un orologio e fa scegliere l'ora



Pianifica l'evento

Pianificazioni in corso

☐ Elenco delle pianificazioni in corso




Elimina le programmazioni selezionate

- Possibilità di scegliere il tipo di caffè
- Le programmazioni sono differenti, la prima programma il caffè un giorno solo (con rinnovo annuale), mentre l'altra si ripete ogni settimana nei giorni/o selezionati/o
- L'ora e il giorno sono scelti attraverso un form speciale di Materialize, che ritorna una stringa.

Opzioni

- Permette la sola accensione (e quindi spegnimento). Può essere necessario se si devono fare molti caffè, così la caldaia non dovrà riaccendersi ogni volta, evitando così di raffreddarsi.
- Un messaggio ricorda di spegnere la macchina. Lasciarla accesa potrebbe consumare molta corrente

 piCoffee
Caffè veloce | Programma il caffè | Opzioni

Accensione e spegnimento

Accendi la caffettiera

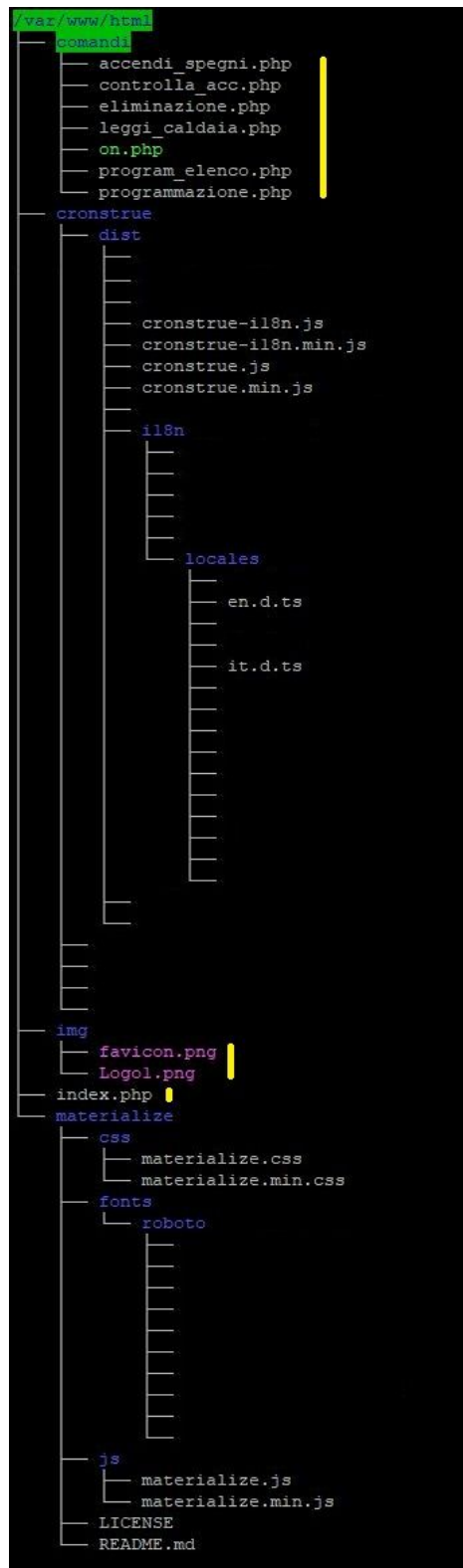
Spegni la caffettiera

Messaggio che ricorda di spegnere la macchina del caffè

N.B.

- Tutte le richieste al server avvengono tramite AJAX perché dà la possibilità di ottenere informazioni in tempo reale senza ricaricare la pagina (Es. se la caldaia è in temperatura)
- Le sezioni “Caffè veloce”, “Programma il caffè” e “Opzioni” fanno parte della pagina **index.php**

Albero delle pagine



Le pagine selezionate in giallo sono quelle scritte od elaborate da me, le altre sono le due librerie che ho utilizzato. Ho omesso i file che non mi sono serviti direttamente.

Le pagine contenute in “comandi” lanciano un comando nella shell del Raspberry Pi tramite la funzione PHP `system(comando,output)`. Nel primo parametro ho sempre scritto un’operazione di lettura e scrittura delle GPIO.

Es.

```
gpio read 3
```

```
gpio write 0 1
```

Il nome dei file è abbastanza esplicativo:

- `accendi_spegni.php`, accende la macchina del caffè se è spenta o la spegne se è accesa
- `controlla_acc.php`, controlla se il dispositivo è acceso
- `eliminazione.php`, elimina gli eventi pianificati
- `leggi_caldaia.php`, ritorna 0 se la caldaia è fredda, 1 se calda
- `on.php`, prepara il caffè e accende la macchina se spenta
- `program_elenco.php`, ritorna l’elenco degli eventi pianificati
- `programmazione.php`, programma un evento



Macchina del caffè completata

Esempio di funzionamento

Dopo essersi connessi al sito web del Raspberry, tramite il suo indirizzo IP, si sceglie un caffè espresso dallo *switch* e si tocca sul bottone con il simbolo della tazza.

In quel momento parte una richiesta AJAX alla pagina *on.php*, con un parametro GET, chiamato *durata*. Se si è scelto il caffè espresso, la variabile sarà di 22 secondi, mentre per quello lungo di 35 secondi.

```
url="comandi/on.php?durata="+encodeURIComponent(durata_coffee);  
richiesta.open("GET",url,false) //ASINCRONO
```

Url e funzione utilizzate nella richiesta. *durata_coffee* vale 22 in questo esempio

A questo punto la pagina *on.php* accende la macchina del caffè e pertanto inizia a scaldarsi. Dopo questo comando parte un ciclo while, che si ripete ogni 3 secondi e che ha come condizione `gpio read 3`, cioè finché il termostato non segna che la caldaia è in temperatura.

```
while (system("gpio read 3")==0) { //Finchè la risposta del server è negativa (cioè 0),  
il  
                                //ciclo si ripete. In questo modo evita che la pompa  
                                //venga attivata subito  
    sleep(3);  
}
```

Quando il programma esce dal ciclo, fa partire la pompa dell'acqua. Il comando di spegnimento, però, è preceduto dalla funzione `sleep(tempo in s)`. Il parametro passatole è appunto la durata, decisa preventivamente con lo *switch*.

Dopo lo *sleep* viene spenta la pompa. Se prima di toccare sul simbolo della tazza del caffè la macchina era già accesa, la lascia nello stesso stato, altrimenti la spegne.

Nel frattempo, l'interfaccia comunica che il caffè è in preparazione e si aggiunge un'animazione di caricamento. Inoltre un simbolo mostra quando la caldaia è in temperatura.

Conclusa l'erogazione del caffè, appare un messaggio di avvenuta erogazione.

Fonti

- <https://crontab.guru/>
- <http://bradymholt.github.io/cRonstrue/>
- https://en.wikipedia.org/wiki/General-purpose_input/output
- <https://www.raspberrypi.org/documentation/usage/gpio/>
- <https://it.wikipedia.org/wiki/Crontab>
- <https://materializecss.com/>
- <https://www.raspberrypi.org/products/raspberry-pi-3-model-b/>
- <https://www.raspberrypi.org/documentation/linux/usage/cron.md>
- <https://stackoverflow.com>
- <http://php.net/>