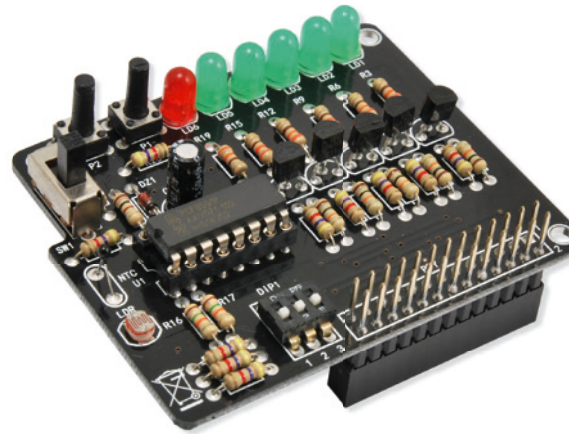
SHIELD I/O PER RASPBERRY PI
ANALOGICO/DIGITALE *(cod. FT1060M)*



Scheda di sperimentazione analogico/digitale, appositamente progettata per essere utilizzata negli esperimenti descritti nel libro Raspberry PI cod. 8330-RASPBOOK1 (acquistabile su www.futurashop.it). Basata sull'integrato convertitore AD/DA a 8 bit PCF8591, dispone di sensore di temperatura (NTC), fotoresistenza, 6 LED, 2 pulsanti, 1 interruttore.
La scheda che si innesta sul connettore GPIO di Raspberry Pi porta a bordo un convertitore ADC a quattro canali a 8 bit ed un convertitore DAC, sempre a 8 bit.
Tramite questa board è possibile realizzare un sistema di controllo remoto e di acquisizione dati che ci consentirà di illustrare tutte le problematiche software relative, dal pannello web alla gestione dei protocolli di comunicazione, all'integrazione dei dati mediante database.
Il circuito della scheda di espansione utilizza la tensione di 3,3 V per alimentare l'integrato PCF8591 un convertitore ADC/DAC a 8 bit, i sensori fotoresistenza e fotocellula, i pulsanti e l'interruttore. Data la bassa corrente erogabile dalla linea a 3,3 V di Raspberry Pi, abbiamo preferito interporre un circuito di potenza tra le uscite digitali ed i LED. I transistor di potenza sono alimentatati dalla linea di tensione a 5V.
Il bus di comunicazione I^2C funziona anch'esso con i livelli a 3,3 V.
Il pin 2 di Raspberry Pi è in grado di erogare 500 mA nella ev. 1 e 300 mA nella rev. 2.

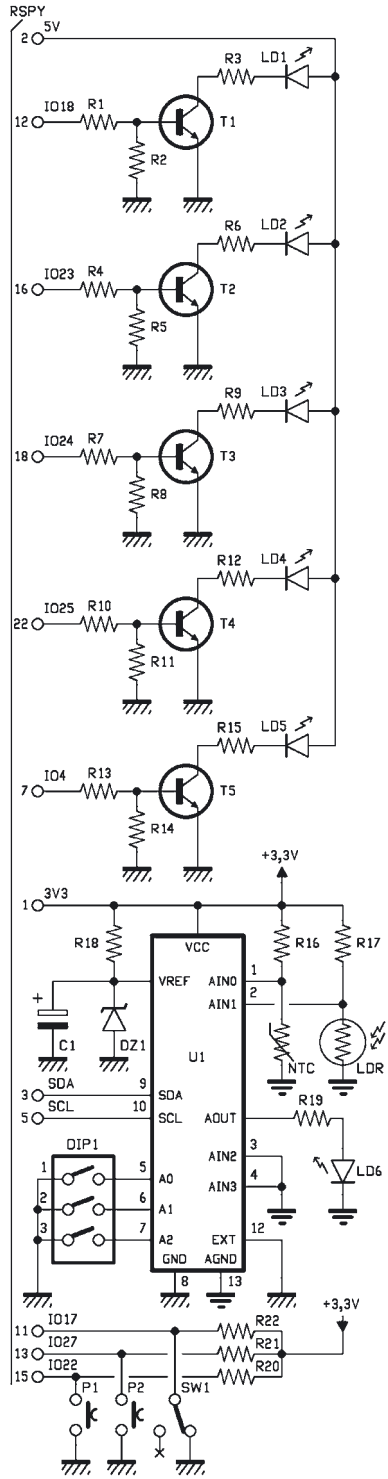
Termostato

L'elemento principale della funzione termostato è la resistenza NTC da 10 kohm, collegata, tramite il partitore di tensione composto da R16 e dall'NTC stessa, all'ingresso analogico 0 dell'integrato PCF8591. La tensione di riferimento per la misura degli ingressi analogici è fornita dalla resistenza di limitazione R18, dal diodo Zener e dal condensatore elettrolitico C1, che mantiene stabile la tensione. Il circuito del termostato è completato dai due LED e dai relativi circuiti di potenza: LD2, che simula l'impianto di condizionamento, collegato al pin 16 (GPIO23), e LD3 collegato al pin 18 (GPIO24) per il riscaldamento. Ciascun circuito di potenza è realizzato con un transistor BC547. Quando il livello del pin di uscita digitale del GPIO è basso, non vi è tensione sulla base del transistor che, in questa condizione, non conduce. Quando il livello del pin assume il valore alto, la tensione sulla base del transistor supera il livello di soglia necessario a portare in conduzione il transistor, che porta a massa la resistenza di limitazione della corrente, permettendo al LED di essere alimentato dalla linea di tensione a 5 V, tramite la resistenza di limitazione da 330 ohm.

Misurazione della luminosità

Dal punto di vista "elettrico" la misura della luminosità richiede solamente il collegamento del sensore al pin AIN1 della fotoresistenza, mediante il

Schema elettrico



partitore di tensione composto dalla fotoresistenza stessa e da R17. Al variare della luminosità varia il valore resistivo della fotoresistenza e, di conseguenza, il valore di tensione misurato sull'ingresso analogico.

LED Flip/Flop

Il LED Flip/Flop è controllato dal lato hardware dal pulsante P1 collegato al pin 15 (GPIO22) del connettore della Raspberry Pi. Il LED corrispondente è LD1, collegato al pin 12 (GPIO18), alimentato per mezzo del transistor T1.

Interruttore AUX

L'interruttore SW1 è collegato al pin 11 (GPIO17) del connettore di Raspberry Pi. Il LED corrispondente è LD5, collegato al pin 7 (GPIO4), alimentato per mezzo del transistor T5.

Messaggio

L'esistenza di un messaggio da leggere viene evidenziato dall'accensione del LED LD4, collegato al pin 22 (GPIO25), tramite il transistor T4. Per ascoltare il messaggio si preme il pulsante P2, collegato al pin 13 (GPIO27).

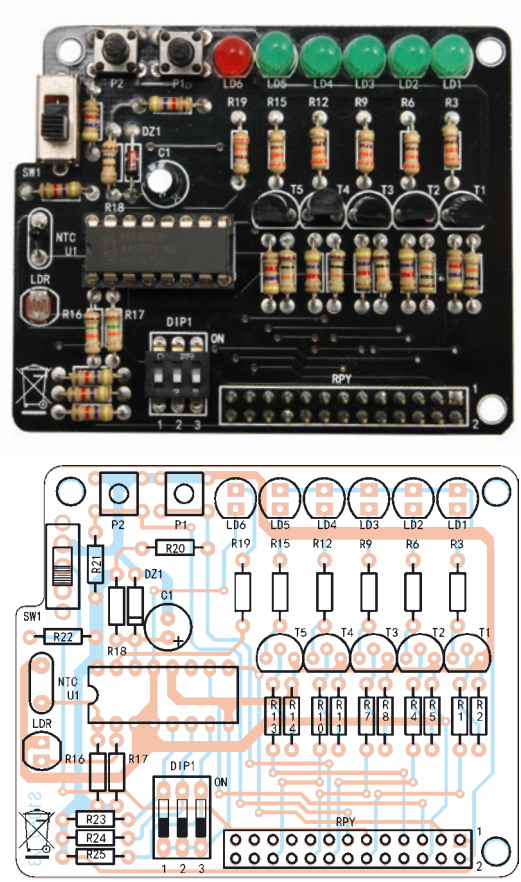
L'uscita DAC, sul piedino AOUT del PCF8591, è collegata al LED LD6, tramite la resistenza di limitazione R19. A seconda del livello di tensione impostato sull'uscita AOUT, LD6 si accende con una luminosità più o meno accentuata.

Le linee del bus I²C, SDA ed SCL, sono collegate rispettivamente ai pin 3 e 5 del connettore GPIO di Raspberry Pi. I pin di indirizzamento 3, 6 e 7 sull'integrato sono colle-

Piano di montaggio

Elenco Componenti:

R1: 4,7 kohm
R2: 10 kohm
R3: 330 ohm
R4: 4,7 kohm
R5: 10 kohm
R6: 330 ohm
R7: 4,7 kohm
R8: 10 kohm
R9: 330 ohm
R10: 4,7 kohm
R11: 10 kohm
R12: 330 ohm
R13: 4,7 kohm
R14: 10 kohm
R15: 330 ohm
R16: 15 kohm
R17: 15 kohm
R18: 100 ohm
R19: 330 ohm
R20: 4,7 kohm
R21: 4,7 kohm
R22: 4,7 kohm
C1: 4,7 μF 100 VL elettrolitico
DZ1: Zener 3,3V 400mW
T1: BC547
T2: BC547
T3: BC547
T4: BC547
T5: BC547
LD1: LED 5 mm verde
LD2: LED 5 mm verde
LD3: LED 5 mm verde
LD4: LED 5 mm verde
LD5: LED 5 mm verde
LD6: LED 5 mm rosso
U1: PCF8591
P1: Microswitch
P2: Microswitch
SW1: Deviatore slitta verticale
DIP1: Dip-Switch 3 vie



NTC: NTC 10 kohm
LDR: Fotoresistenza 2-20 kohm

Varie:
- Zoccolo 8+8
- Strip Femmina 13x2 vie
  per Raspberry Pi
- Distanziale F/F (3 pz.)
- Vite 8 mm 3 MA (6 pz.)
- Circuito stampato

gabili a massa, ciascuno con un jumper, in modo da poter modificare l'impostazione dell'indirizzo del convertitore. Nel nostro progetto li teniamo tutti collegati a massa per mezzo dei selettori del DIP1 posizionati su ON.
I due ingressi analogici non utilizzati AIN2 e AIN3 sono collegati a massa, per evitare valori flottanti.
Per prima cosa, se Raspberry Pi è in funzione, spegniamola con il comando:

shutdown -h now

oppure

halt

e poi togliamo l'alimentazione. Colleghiamo il connettore RSPY dello shield sul connettore GPIO di Raspberry Pi e ridiamo tensione. Ora dobbiamo attivare i moduli di gestione del bus I²C (nel mondo Windows sono chiamati driver).
A tal proposito dobbiamo sottolineare la diversità di approccio ai moduli di gestione delle periferiche esterne nei mondi Windows e GNU/Linux che illustriamo in Appendice 1. Il modulo di gestione del bus I²C è un device come quelli che gestiscono tutte le risorse ed i dispositivi collegati a Raspberry Pi, e fa parte di quelli compilati nel kernel di Raspian come moduli esterni e non facenti parte integrale del kernel. Oltre a questo, nella distribuzione di Raspbian il modulo di gestione del bus I²C è anche blacklisted ovvero "oscurato" all'utilizzo normale. Questo perché le linee del bus condividono alcuni pin digitali del GPIO che vengono privilegiati nella configurazione predefinita. Per … resuscitare e rendere utilizzabile il modulo di gestione del bus I²C è necessario toglierlo dalla blacklist e poi "aggiungerlo" all'insieme di moduli conosciuti dal kernel.
Apriamo il file di configurazione che contiene l'elenco dei moduli blacklisted (oscurati), con il comando (**Fig. 1**):

nano /etc/modprobe.d/raspi-blacklist.conf

Nano è un editor di testo minimale che funziona in ambiente terminale.

Eliminiamo il modulo I²C dalla blacklist cancellando la riga o, come abbiamo preferito noi, commentandola con un "#" (**Fig. 2**).
Premiamo Ctrl-X e poi Y alla richiesta di salvare il file dopo le modifiche.
Eseguiamo un reboot per rendere effettive le modifiche.
Ora dobbiamo fare in modo che il modulo "liberato" venga caricato e diventi parte integrante del kernel. Per questa operazione abbiamo due possibilità. La prima ci permette di caricare il modulo a comando, ed ha validità per tutto il tempo nel quale Raspberry Pi rimane acceso. Al

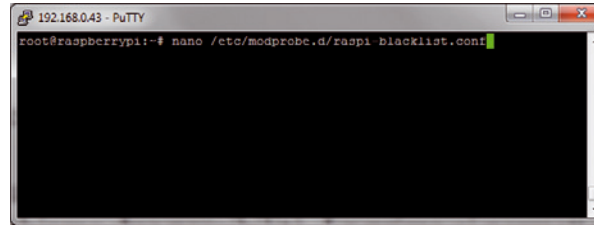


Fig. 1

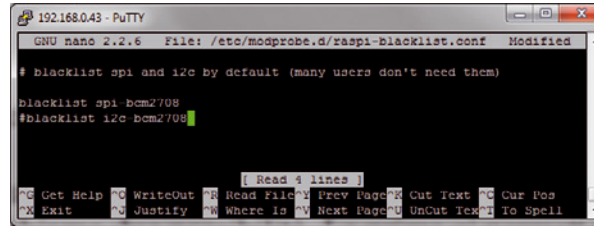


Fig. 2

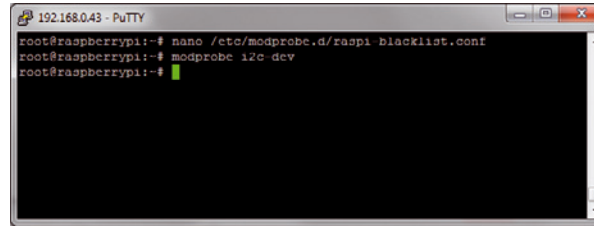


Fig. 3

boot successivo il modulo dovrà essere ricaricato di nuovo a comando. La seconda ci permette di caricare il modulo direttamente al boot del sistema operativo, e renderlo disponibile alle applicazioni subito dopo il boot, condizione indispensabile in un sistema server unattended.

La prima possibilità richiede l'utilizzo del comando modprobe. Scriviamo (**Fig. 3**):

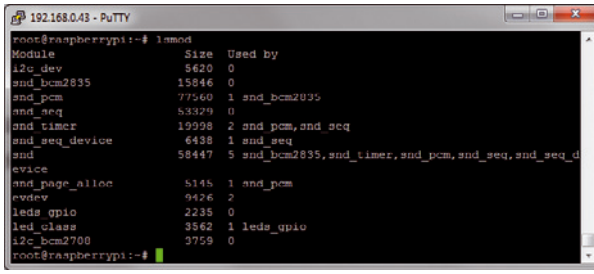


Fig. 3

modprobe i2c-dev

Possiamo vedere il buon esito dell'attivazione dei driver con il comando che mostra la lista di tutti i moduli installati (**Fig. 4**):

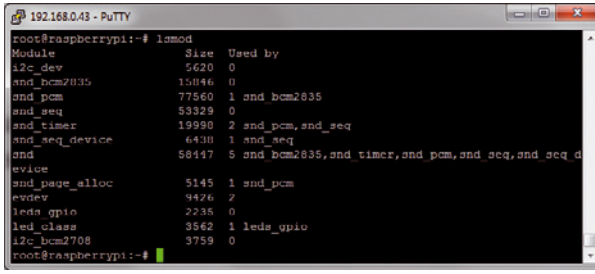


Fig. 4

lsmod

Dato che in GNU/Linux tutto (o quasi) è un file, se andiamo nella cartella /dev vediamo apparire i file di collegamento ai device I²C-0 ed I²C-1 (**Fig. 5**). Il comando modprobe permette di caricare e di scaricare i moduli a run time e mantiene i suoi effetti fintanto che Raspberry Pi è acceso. In caso di spegnimento, o anche solo di reboot, il nostro modulo dovrà essere ricaricato manualmente. Questa condizione non è adatta a funzionare con un'applicazione stand alone, che deve funzionare in modo automatico. Il comando modprobe, con l'opzione remove, può essere utilizzato anche per disattivare un modulo caricato in precedenza.

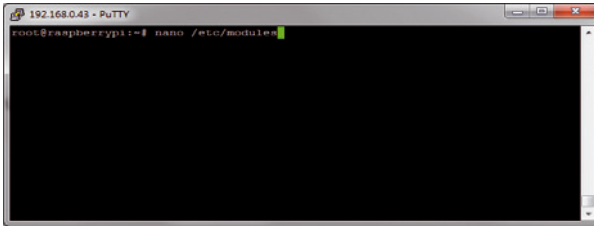


Fig. 5

modprobe -r i2c-dev

Se si desidera che i moduli vengano caricati all'accensione di Raspberry Pi è necessario abilitare il caricamento permanente del driver I²C, che si realizza modificando opportunamente il file di configurazione /etc/modules, che contiene la lista dei driver da caricare al momento del boot (**Fig. 6**):

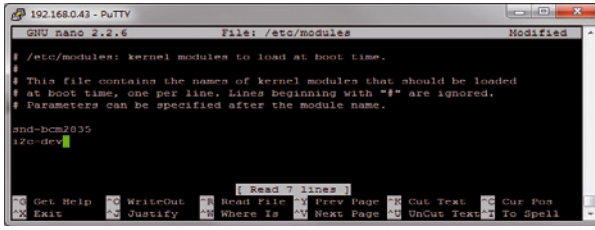


Fig. 6

nano /etc/modules

ed aggiungere una nuova linea al file di configurazione

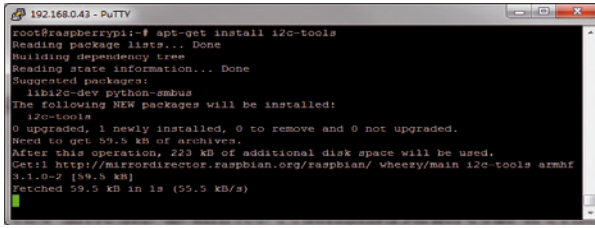


Fig. 7

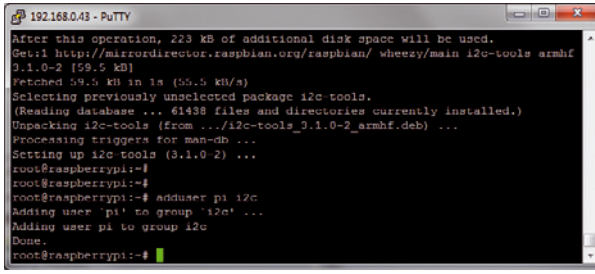


Fig. 8

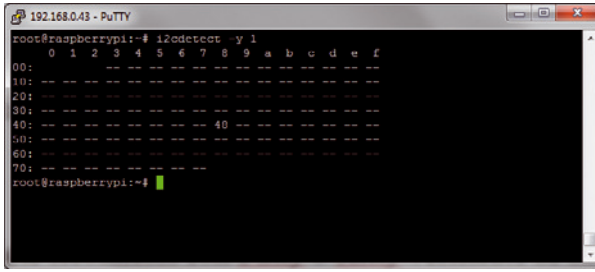


Fig. 9

che contiene (**Fig. 6**):

i2c-dev

Premete CTRL X e poi Y per salvare le modifiche al file ed uscire (**Fig. 7**).
Ora installiamo il pacchetto I²C-tools che ci fornisce una serie di funzioni utilizzabili a linea di comando per verificare il funzionamento del bus I²C (**Fig. 8**):

apt-get install i2c-tools

Aggiungiamo il nostro utente pi al gruppo I²C (**Fig. 9**):

adduser pi i2c

Eseguiamo il reboot di Raspberry Pi per attivare le nuove configurazioni con il comando:

reboot

Ricordatevi di ridare il comando:

modprobe i2c-dev

Dopo che Raspberry Pi si è riavviata e vi siete riconnessi con Putty o Kitty verificate se sul bus I²C è visibile il convertitore ADC con il comando

i2cdetect -y 0 per Raspberry Pi rev. 1 oppure
i2cdetect -y 1 per Raspberry Pi rev. 2

Dovrete ottenere un risultato simile a quello visibile in Fig. 13 dove l'indirizzo 0x48 identifica il convertitore ADC.
Il punto centrale del partitore di tensione del quale fa parte la NTC è collegato al canale AIN0 del convertitore ADC, il punto centrale del partitore di tensione del quale fa parte la fotoresistenza è collegato al canale AIN1.
Sperimentiamo ora il funzionamento del convertitore utilizzando la linea di comando da terminale.
Usiamo i comandi:

i2cset –y 0 0x48 0x00 per Raspberry Pi rev. 1 oppure
i2cset –y 1 0x48 0x00 per RaspBerry Pi rev. 2

per istruire il convertitore a restituire i valori letti sul canale 0 ADC.
Per leggere effettivamente il valore esadecimale sul canale 0 utilizziamo il comando:

i2cget –y 0 0x48 per Raspberry Pi rev. 1 oppure
i2cget –y 1 0x48 per RaspBerry Pi rev. 2

che ci restituisce un byte che rappresenta il valore digitale della tensione misurata ai capi della resistenza NTC, nel nostro caso "0x70" (**Fig. 10**).
Abbiamo dato il comando due volte perché il convertitore

ADC alla prima lettura restituisce sempre 0x80. Per verificare il funzionamento provate a dare diversi comandi tenendo la resistenza NTC tra le dita. Vedrete variare i valori esadecimali letti.
Ora installate il pacchetto “bc” che è un programma generico di calcolo a riga di comando: permette calcoli e conversioni in tutte le basi e val la pena di provarlo perché può risolvere molte situazioni:

apt-get install bc

Per convertire il valore esadecimale nel corrispondente valore decimale ora possiamo usare il comando:

echo ‘ibase=16;obase=A;70’ | bc

che ci restituirà il valore 112 (**Fig. 11**)
Il comando stampa sul terminale il risultato della conversione da base 16 a base 10 (A in esadecimale) del valore 70 (è obbligatorio utilizzare le maiuscole per le lettere) e lo passa al programma “bc”. Per esempio, per le conversioni da binario a esadecimale utilizzate il comando:

echo ‘ibase=2;obase=10000;bbbbbbbb’ | bc

sostituendo bbbbbbbb con il valore binario che volete convertire
Per convertire da esadecimale a binario utilizzate il comando:

echo ‘ibase=16;obase=2;hh’ | bc

sostituendo hh con il valore che volete convertire.
Possiamo eseguire lo stesso test sul canale AIN1 del convertitore ADC per sincerarci del corretto funzionamento del circuito della fotoresistenza.
Per provare l’uscita DAC del convertitore usiamo il comando (**Fig. 12**):

i2cset –y 1 0x48 0x40 255

Per il significato dei valori dei registri vi rimandiamo al riquadro dedicato all’integrato PCF8591. In questo comando 0x48 è l’indirizzo in esadecimale sul bus I²C del convertitore, il secondo valore 0x40 alza il bit che abilita il DAC ad emettere sul pin AOUT la tensione corrispondente al valore esadecimale impostato, mentre il valore 255 è il valore da passare al DAC per essere convertito in tensione. Dopo il comando dovremmo vedere il

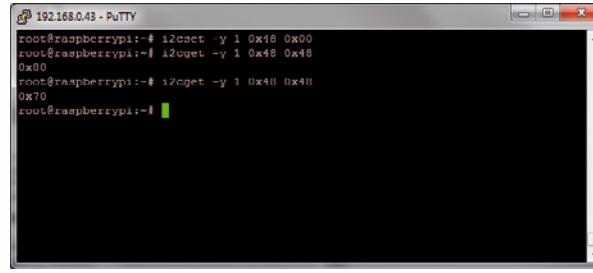


Fig. 10

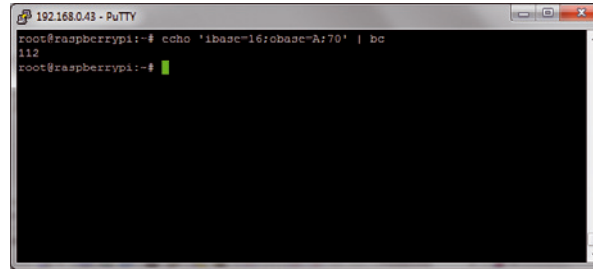


Fig. 11

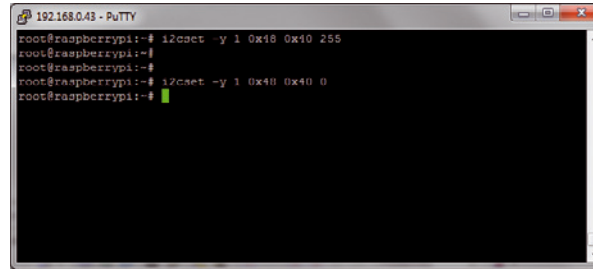
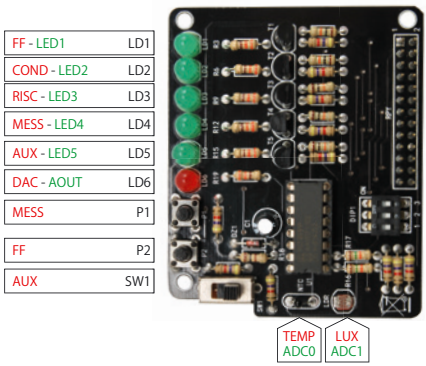
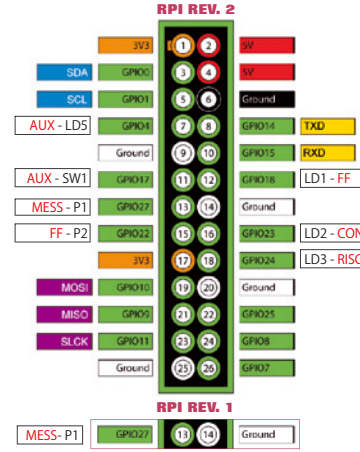


Fig. 12

Schema comandi, sensori e LED





LED rosso accendersi.
Per spegnerlo usiamo il comando:

i2cset –y 1 0x48 0x40 0

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. 0331-778112
Web site:
www.futurashop.it
Info tecniche:
www.futurashop.it/Assistenza-Tecnica