

Arduino SmartHouse with web-server and Bluetooth

WHAT

Le specifiche del progetto richiedevano di realizzare un sistema di controllo per una SmartHouse.

Il progetto è diviso in 2 parti, **Server** e **Client**:

1) Il **Server** Controllore della SmartHouse deve essere realizzato tramite un microcontrollore Arduino Mega 2560.

Una volta che il Client si è connesso al Server, quest'ultimo potrà ricevere due tipi di comandi:

- Comandi di **Setup**: per impostare i nomi dei vari canali(pin) utilizzati.
- Comandi di **Controllo**: per accendere/spegnere luci(simulate con dei led) e leggere da canali digitali ed analogici.

La comunicazione tra il Server ed il Client deve avvenire tramite connessione bluetooth.

Oltre a questo, il Server deve avvalersi delle **ISR**(Interrupt Service Routine) per gestire la comunicazione input/output con il Client.

Inoltre, tutti i comandi di Setup devono essere salvati sulla eeprom(Memoria non volatile) di Arduino.

2) Il **Client** deve instaurare una connessione bluetooth al Server. Dopodichè tramite shell di comando l'utente può inviare i comandi di Setup e Controllo al Server.

Il front-end del Client può essere realizzato sia tramite shell di comando o tramite un WebServer, l'importante è mantenere il back-end in linguaggio C.

La realizzazione è stata effettuata sia tramite shell di comando, sia tramite WebServer, di default è impostata la shell.

HOW

Come già accennato sopra per realizzare il progetto ci siamo avvalsi dei seguenti dispositivi:

- PC
- Arduino Mega 2560
- Dongle Bluetooth HC-05

- 7 LED per Arduino
- 10 Resistori di resistenza tra 100Ω e $2k\Omega$
- 1 BreadBoard
- Diversi Jumper(Ne bastano 6, i necessari per il Dongle Bluetooth)

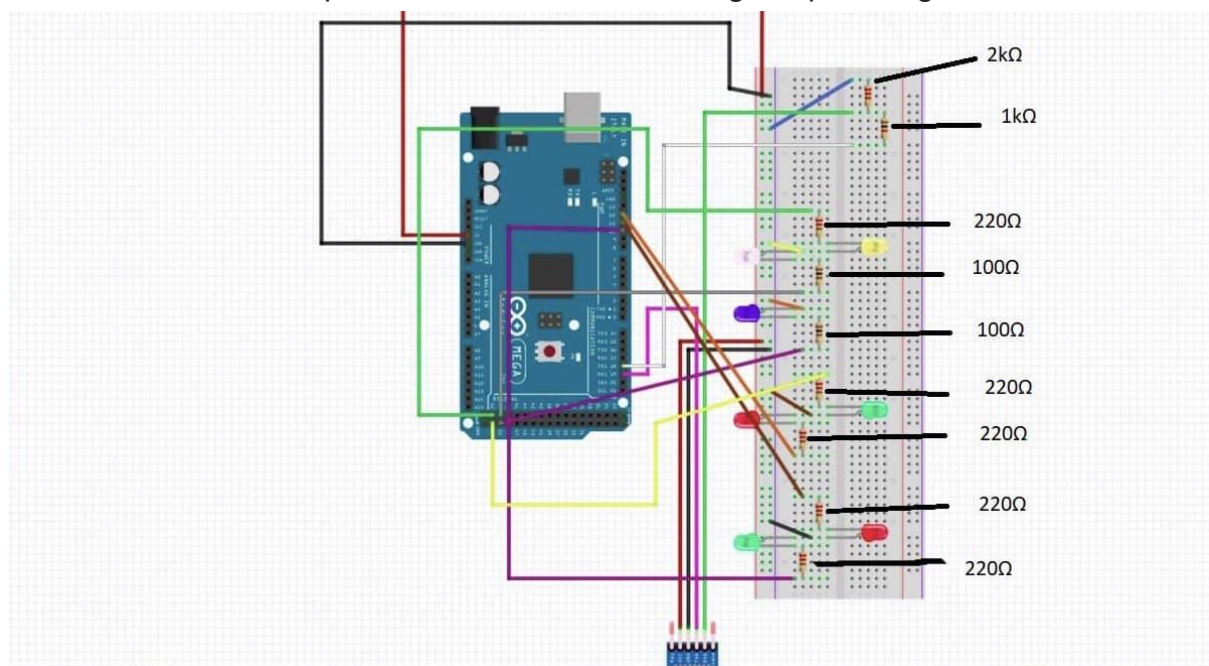
Il Software utilizzato è il seguente:

- Lato Server completamente sviluppato in linguaggio C
- Lato Client:
 - 1)Comunicazione e shell di comando sono stati sviluppati in C
 - 2)Il WebServer è stato realizzato in HTML+CSS+Javascript

HOW TO RUN

Prima di far partire il programma dobbiamo avere degli accorgimenti e compiere i seguenti tasks:

- 1)Ricostruire un modello Arduino avvalendosi dei tools sopra elencati.
Per la realizzazione si può fare riferimento all'immagine qui di seguito.



- 2)Dopo aver finito tutti i collegamenti, possiamo collegare Arduino ad un alimentatore e recarci al PC.
- 3)Dobbiamo Scoprire l'indirizzo MAC del Dongle Bluetooth per aprire una connessione con il Client.
Per far ciò possiamo avvalerci di un tool di Ubuntu chiamato **hcitool**.

Quindi dopo aver aperto la shell di comando di Ubuntu digitare:

```
sudo hcitool scan
```

Vi apparirà una schermata come la seguente:

```
(base) lorenzo@lorenzo-latitude:~$ sudo hcitool scan
[sudo] password di lorenzo:
Scanning ...
    98:D3:51:F5:E0:93    HC-05
                        realme 7
(base) lorenzo@lorenzo-latitude:~$
```

Quindi copiamo l'indirizzo MAC del tipo "98:D3:51:F5:E0:93" e ci rechiamo all'interno della cartella **client**.

Nella cartella **client**, editare il file **avr_client_functions.c**:

Sostituire alla **riga 27**:

char dest [18] = "98:D3:51:F5:E0:93"

Dove tra gli apici dovete inserire l'indirizzo MAC del vostro dongle Bluetooth.

4) Scaricare librerie necessarie alla connessione bluetooth:

Dobbiamo scaricare una libreria necessaria per la connessione Bluetooth.

Aprire la shell di comando e digitare:

```
sudo apt-get install libbluetooth-dev
```

5) Adesso dobbiamo scaricare una libreria per la connessione con il web server

Aprire la shell di comando e digitare:

```
sudo apt-get install libwebsockets-dev
```

6) Ora possiamo compilare i codici Server e Client con i relativi **make**

(**make main.hex** per il Server).

7) ./main per far partire il client

8) Per conoscere i vari comandi digitare "help", una volta stabilita la connessione.

Created by Lorenzo Ciarpaglini