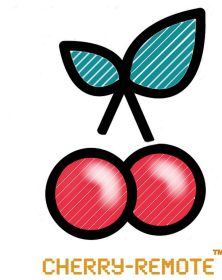


Cherry Remote

Fabian Greavu, Silvia Raggi

24 Luglio 2018



1 Abstract

In questo documento verrà presentato il progetto **Cherry Remote** dettagliandone il motivo, l'uso, i componenti, l'architettura e la sua realizzazione fisica.

2 Introduzione

Una delle semplici tecnologie con cui si ha a che fare ogni giorno è quella ad infrarossi per la comunicazione di semplici segnali tra un dispositivo client ed uno server: un telecomando.

E' ormai molto comune avere in casa un insieme di dispositivi controllati da remoto aventi un proprio telecomando, il che implica un numero alto di

dispositivi ognuno adibiti alla comunicazione con il proprio dispositivo. Nasce dunque la necessità di possedere una modalità di controllo univoca su tutti questi dispositivi, veloce da configurare, sempre disponibile e facile all'uso.

3 Target

L'applicazione è destinata ad un uso domestico per il controllo di monitor, decoder, condizionatori oppure ad ambienti lavorativi per l'uso anche di apparecchiature più avanzate (basta che abbiano un supporto ad infrarossi per la trasmissione dei dati).

4 Funzionamento

L'utente connesso alla propria rete wifi di casa apre la pagina specifica dell'applicazione ed ha da subito una visione diretta su un telecomando composto da pulsanti per comandare un certo dispositivo in casa. Vi sono diversi modelli grafici per l'uso di un telecomando ma tutti incorporano pulsanti di interazione. In alto è disponibile la lista dei telecomandi attraverso la quale si può scegliere un diverso telecomando. Ai lati della lista vi è un pulsante per cancellare il telecomando attuale, ed un pulsante di aggiunta di un nuovo telecomando (per far ciò è necessario possedere fisicamente il telecomando da replicare). Premendo quest'ultimo l'interfaccia cambia e vengono richiesti un nome per il telecomando, la scelta di un modello grafico da applicare e la scelta del controller da utilizzare. Appena viene scelto un modello si può iniziare la fase di registrazione pulsanti: dal modello del telecomando scelto, premendo uno dei vari pulsanti viene messo a disposizione un periodo di tempo (circa 5 secondi) per premere il tasto corrispettivo sul proprio telecomando puntandolo verso il controller, che ne salverà i dati. Una volta terminata la registrazione basterà premere il tasto "Confirm" per aggiungere il nuovo telecomando (oppure Cancel in ogni momento per tornare ai telecomandi).

Per l'uso di un telecomando basta selezionarlo dalla lista in alto ed utilizzare i suoi pulsanti. Il sistema invierà da sé il codice infrarossi specifico. Questa sezione viene approfondita successivamente nel capitolo 6.

5 Realizzazione

Per la realizzazione sono stati usati i seguenti linguaggi di programmazione:

- **HTML 5.0** per la parte di design e UI
- **Javascript** per legare interazioni utente ad eventi preimpostati
- **JQuery** per navigare all'interno dell'html
- **AJAX** per le richieste al server php e per i pacchetti HTTP POST verso il controller
- **PHP** per la comunicazione al Database
- **SQL** per le query inviate al Database
- **C++** per il programma embedded nel controllore

Mentre i software utilizzati comprendono WebStorm (HTML, javascript, JQuery, ajax), Google Chrome (debug, testing), Notepad++ (php), MySQL Workbench e PHPMyAdmin (sql), Arduino IDE (c++ controller). Infine lo scambio dei dati attraverso i vari componenti avviene tramite file JSON.

5.1 Componenti principali

Per la giusta realizzazione del progetto saranno necessari i seguenti (i corrispettivi numeri si riferiscono alla numerazione nell'immagine descrittiva):

- **1. Raspberry pi:** da usare come server e centro database
- **2. Database:** viene usato dall'applicativo web per salvare/scrivere dati
- **3. Smartphone:** da usare come interfaccia per il controllo dei dispositivi IR
- **4. Rete wireless:** una qualsiasi rete personale wifi (chi ha una connessione ad internet ne è in possesso) a cui collegare tutti i dispositivi

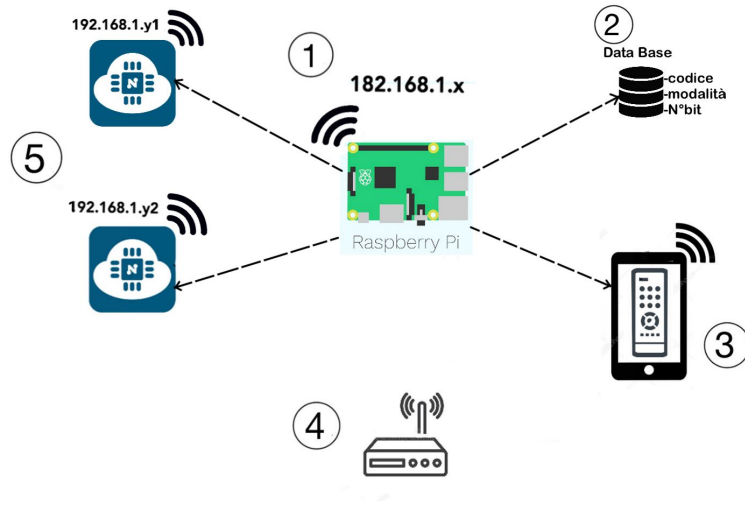


Figure 1: Componenti.

- **5. NodeMcu:** basta una qualsiasi versione con connettività wifi funzionante da usare per trasmettere codici ai dispositivi tramite ricevitori/emettitori IR

5.2 Cenni IR e tecnologia

Nel programma embedded nel microcontrollore NodeMCU vi è una libreria usata in questo progetto: IRremoteESP8266. Essa espone metodi di lettura e scrittura dei segnali IR secondo la loro tipologia. Come esempio il comando di standby di monitor di tipo LG è 20DF10EF (hex) chiamando il metodo sendNEC della libreria, mentre la maggior parte dei segnali IR sono di tipo RAW, quindi vettori di byte con lunghezza maggiore di 500. Cherry remote funziona con entrambi.

5.3 Comunicazione tra componenti

Nella fase di uso di un certo telecomando ogni tap su un pulsante genera una richiesta HTTP POST verso l'indirizzo IP del controller (NodeMCU)

sulla porta 80 nella scheda channel_ir. Il pulsante stesso contiene il codice da inviare (hex oppure raw) che viene ricavato usando JQuery e inviato sotto formato JSON.

Nella fase di registrazione di un comando si invia una richiesta HTTP POST verso l'indirizzo IP del controllore, porta 80 scheda channel_ir e viene dato un tempo massimo di 5 secondi per la risposta. In caso positivo il controller risponde con un file JSON avente il codice (hex o raw) del segnale IR appena letto.

Quando infine bisogna chiamare il codice in PHP per comunicare con il Database si effettuano sempre richieste HTTP POST oppure GET inviando file JSON contenente i dati prestabiliti.

La relazione fra i vari moduli (con il loro linguaggio di programmazione) viene proposto di seguito:

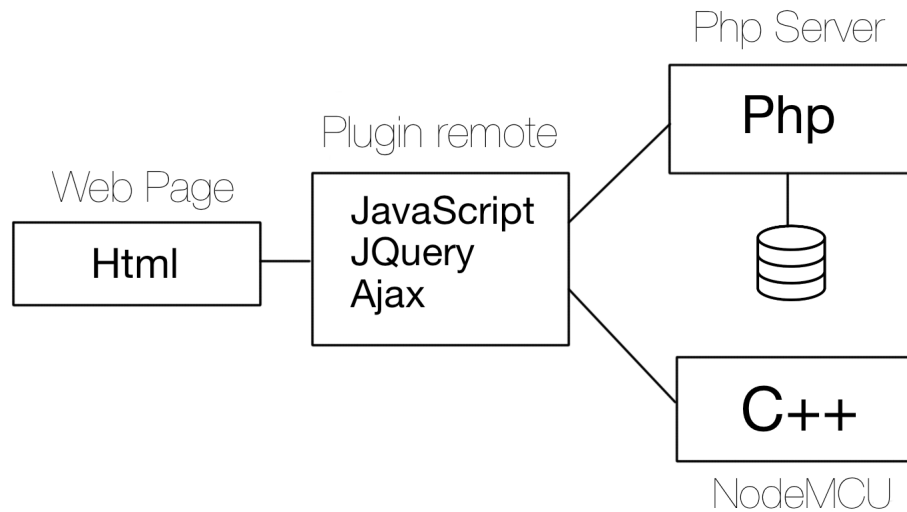


Figure 2: Relazione moduli.

5.4 Database

Viene proposto e dettagliato lo schema seguente del database:

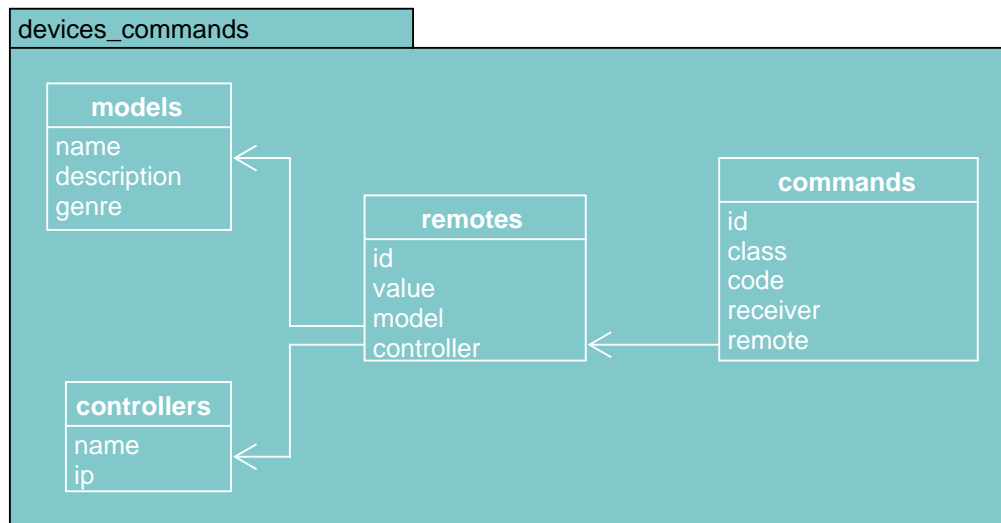


Figure 3: Data Base.

- **models:** la tabella dei modelli grafici con descrizione allegata che si possono scegliere per il proprio telecomando. Ogni modello ha un genere di appartenenza (o categoria)
- **controllers:** la tabella dei controller, ovvero dei dispositivi NodeMCU connessi e funzionanti aventi il proprio indirizzo IP
- **remotes:** la tabella dei telecomando. Ognuno di essi ha un modello (per la UI) e un controller (per inviare i segnali) di riferimento, oltre ad un nome proprio
- **commands:** la tabella dei comandi. Per ogni telecomando si ha dai 5 ai 20 comandi (segnali IR, uno per pulsante) da inviare al controllore. Ogni comando ha un codice (hex o raw), la classe (per rintracciare il pulsante html di riferimento), il receiver type e il telecomando al quale fa riferimento

5.5 Intallazione

Per iniziare l'installazione occorre possedere la lista dei componenti del progetto.

Si inizia dunque a configurare il RPI per essere attivo e funzionante (installando apache2, php7 e mysql). Una volta configurato in modo che si colleghi automaticamente alla wifi bisogna trasferire tutto il progetto sulla cartella apposita, in modo da avere disponibili nella LAN le pagine web del Remote Cherry. Come ultima cosa, creare il database con il file SQL fornito.

Si procede dunque con la configurazione dei controllori NodeMCU. In caso di hobbistica/DIY viene allegato lo schema del circuito e il codice sorgente da caricare tramite l'IDE di Arduino, mentre per i consumatori finali verrà dato il dispositivo finale dotato di un proprio design (stampa 3D o altro). Anche lui avrà il campo SSID e password da configurare per connettersi direttamente alla wifi di casa.

A questo punto bisogna rendere statici gli indirizzi IP dei controllori NodeMCU e a tal scopo bisogna vedere caso per caso il tipo di Router a disposizione (in generale entrare nella sezione accessi o dhcp).

Una volta concluso ci si inserisce nel database gli indirizzi ip dei controllori nella tabella controllers (la tabella models invece è già popolata).

Per quanto riguarda l'alimentazione il RPI deve essere collegato alla presa di corrente con un alimentatore a 5V (1,5-3 Ampere MAX) mentre il NodeMCU può avere alimentazione equivalente oppure essere alimentata da due pile ricaricabili li-po a 3,7V collegate in serie (forniscono 7,4V). La disponibilità di questo componente verrà dunque a mancare dopo un certo periodo per cui accorgimenti di Power Management e gestione di librerie Wifi a risparmio energetico potrebbero essere introdotte per allungare il periodo di vita.

Cherry Remote è pronto all'uso!

6 Design

In questa sezione parliamo di come è stata realizzata la parte visiva del nostro Cherry Remote, delle opzioni che la pagina web può offrire a chi sta utilizzando il dispositivo e del design.

6.1 User Interface Design

Una volta digitato l'indirizzo Ip, l'utente avrà a disposizione la pagina web che gli permetterà di utilizzare i telecomandi. I colori che abbiamo applicato per caratterizzare i pulsanti e le specifiche sono stati scelti tra le palettes studiate appositamente per i siti web. La realizzazione grafica è stata pensata per essere semplice ed immediata, quindi con l'uso di icone minimali. La pagina di apertura è la seguente.

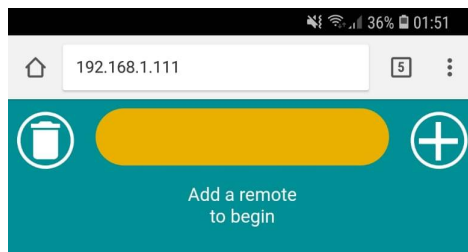


Figure 4: Schermata iniziale della pagina web.

È stata creata una prima Listbox (in index.html) e iniziando da destra verso sinistra abbiamo :

1. **Pulsante "Add Remote"**: questo pulsante è stato realizzato per avere la possibilità di aggiungere nuovi telecomandi. Una volta premuto il pulsante comparirà una nuova schermata. Guardiamo nello specifico le opzioni che l'utente ha a disposizione per l'aggiunta di un nuovo telecomando. La schermata sarà la seguente.

Questa Select box (in addRemote.html) ha lo stesso style e colori della pagina precedente. Abbiamo aggiunto le seguenti options:

- **Insert Info**: in questa sezione inseriamo l'Id del telecomando.
- **Choose Model**: qui abbiamo a disposizione i modelli selezionabili. È possibile scegliere modelli progettati appositamente per Tv, condizionatore o stereo attraverso un drop down menù.
- **Choose NodeMCU**: come abbiamo già detto possiamo avere a disposizione più di un Controller, quindi ad esempio, se vogliamo aggiungere il telecomando per il condizionatore in salotto basterà selezionare il Controller situato in quell'ambiente.

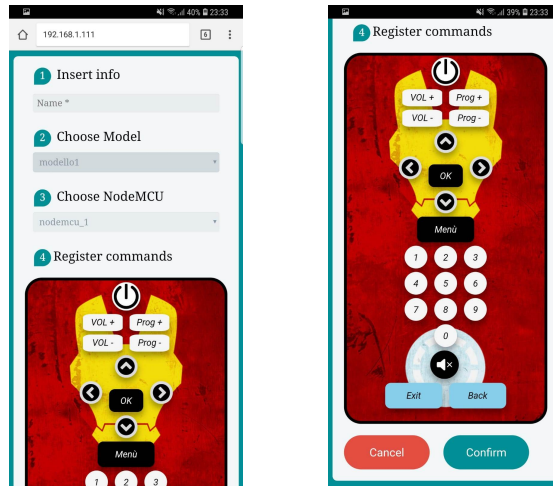


Figure 5: Schermata per l'aggiunta del telecomando.

- **Register commands:** scelto il telecomando possiamo registrare i singoli pulsanti (come specificato nel punto 4). Una volta registrato il pulsante questo si colorerà di verde, cioè confermerà la memorizzazione del codice, in modo da aiutare l'utente che sta effettuando l'operazione.
- **Pulsanti "Cancel" e "Confirm":** completata la fase precedente possiamo confermare o annullare l'operazione attraverso i buttons a fine pagina, in caso di cancellazione si ritornerà alla schermata iniziale.

2. **Barra di selezione:** una volta aggiunti i singoli telecomandi, questi saranno salvati con l'apposito nome (Id) nel Database e resi quindi disponibili. Sarà possibile utilizzarli selezionandoli tramite il drop down menù.
3. **Pulsante di "Delete":** il bottone con l'icona del cestino offre l'opportunità di poter eliminare un telecomando qualora questo non sia più utile. Prima di effettuare l'operazione apparirà una finestra di conferma per la cancellazione.

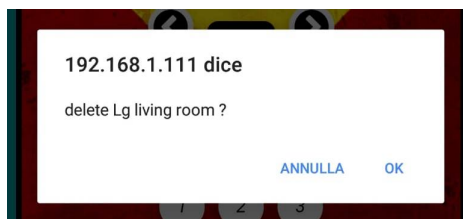


Figure 6: Conferma di eliminazione del telecomando.

6.2 Model Design

Per il Cherry Remote sono stati realizzati 4 tipi di telecomandi: due per la tv, uno per il condizionatore e uno per lo stereo (ognuno situato in una propria pagina html). I modelli hanno background, style e pulsanti molto diversi tra di loro, in modo da offrire non solo un design giovanile ed accattivante, ma un dispositivo che può soddisfare anche gli utenti più esigenti. Esempio portante lo sono i due telecomandi per la Tv: in Figure 6. abbiamo un primo telecomando piuttosto completo, dotato di un tasto menù e dei relativi pulsanti per muoversi nella schermata del proprio televisore; il secondo telecomando è più semplice e dispone di un solo tasto source, quindi ad esempio può essere usato per il decoder.

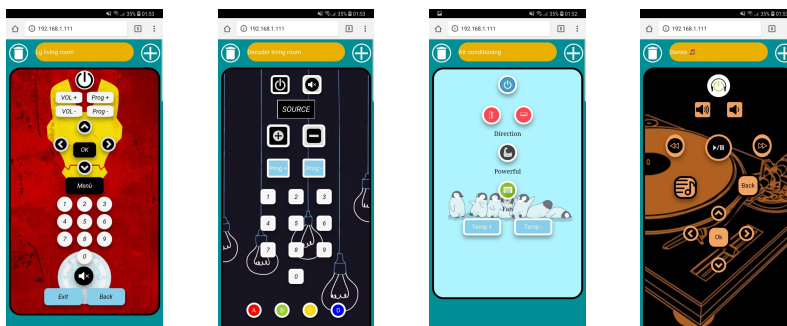


Figure 7: I modelli del Cherry Remote

6.3 Controller Design

Parliamo dei Controllers. Questi, essendo dispositivi fisici che devono essere collocati in casa o in ufficio, dovranno avere anche loro un design adatto a tutti gli ambienti, ed è per questo che abbiamo pensato ad un possibile contenitore che possa essere di gradimento a coloro che vorranno possedere tale dispositivo. Il concept ideato è minimale, sostanzialmente abbiamo un involucro nero che conterrà il nostro Controller, con una sola piccola fessura per il led.



Figure 8: Concept per i Controllers

7 Considerazioni finali

Il progetto è stato testato con telecomandi di tipo LG (sottotipo di NEC) e SAMSUNG (sottotipo di RC6) utilizzando smartphone con sistema operativo android e ios.

Entrando in un contesto più tecnico le componenti hardware utilizzate sono le seguenti:

- **Raspberry PI 3**
- **NodeMCU V3**
- **Emettitore IR** standard
- **Ricevitore IR** modulo XC-4427
- **Fili** di connessione semplici per arduino

Mentre le componenti software sono le seguenti:

- Apache2 2.4.25-3+deb9u5
- mariadb-server-10.1 10.1.23-9+deb9u1
- mariadb-client-10.1 10.1.23-9+deb9u1
- php7.0 7.0.30-0+deb9u1
- Arduino IDE 1.8.5
- WebStorm 2018.1.1
- Google Chrome Versione 67.0.3396.99