

Sistemi Embedded e Internet of Things
Smart Dumpster

Alessandro Marcantoni

AA. 2019/2020

1 Analisi

Si vuole realizzare un sistema IoT che implementi una versione semplificata di uno *smart dumpster*, ovvero un cassonetto dei rifiuti "intelligente". Il sistema complessivamente è costituito da 5 sotto-sistemi:

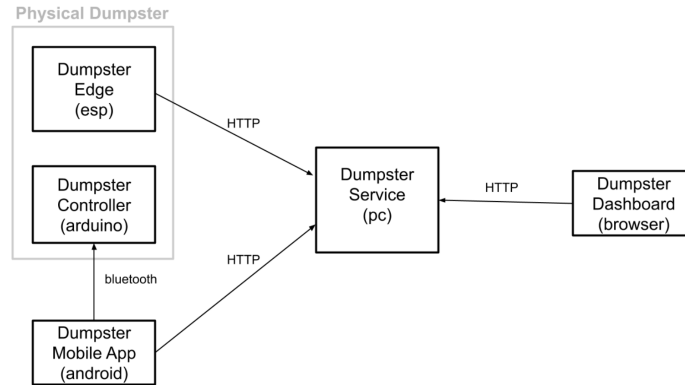


Figura 1: Architettura dello Smart Dumpster

1.1 Dumpster Controller (Arduino)

Sistema embedded che controlla il cassonetto e interagisce con utenti dotati di dispositivo mobile su cui è in esecuzione la parte Mobile App (via Bluetooth).

1.2 Dumpster Mobile App (Android)

Permette all'utente di interagire con il cassonetto e il processo di deposito di un rifiuto. Interagisce sia con la parte Controller (via Bluetooth), sia con la parte Service (via HTTP).

1.3 Dumpster Edge (ESP)

Sistema embedded sempre parte del cassonetto che ha come compito monitorare la quantità di rifiuti depositati e interagire con la parte Service (via HTTP).

1.4 Dumpster Service

Servizio REST che tiene traccia dello stato del cassonetto.

1.5 Dumpster Dashboard (Browser)

Front end per visualizzazione/osservazione/analisi dati.

2 Dumpster Controller

Il Controller è dunque rappresentato da un microcontrollore *Arduino UNO*. La board include i seguenti elementi:

- Tre led verdi L_A , L_B , L_C
- 1 servo motore M con cui si attua l'apertura e la chiusura dello sportello
- 1 modulo Bluetooth HC-06

che possono essere disposti nel seguente modo:

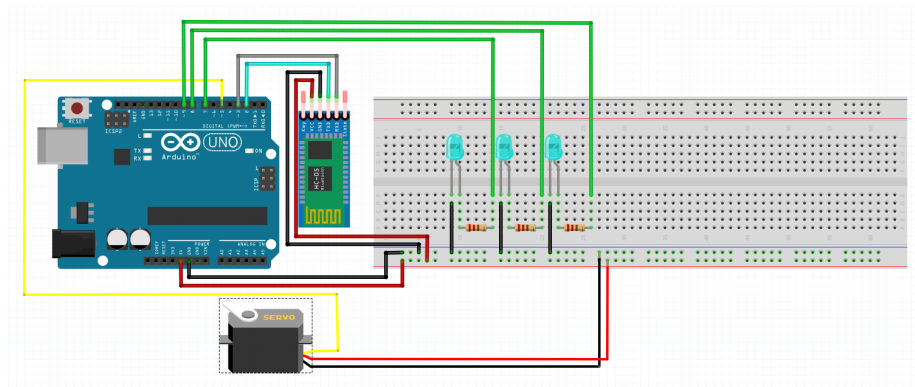


Figura 2: Schema Fritzing del Dumpster Controller

La logica di funzionamento del *Controller* doveva essere implementata in termini di **macchina a stati finiti sincrona**. Di seguito è perciò riportato il diagramma di stato della macchina:

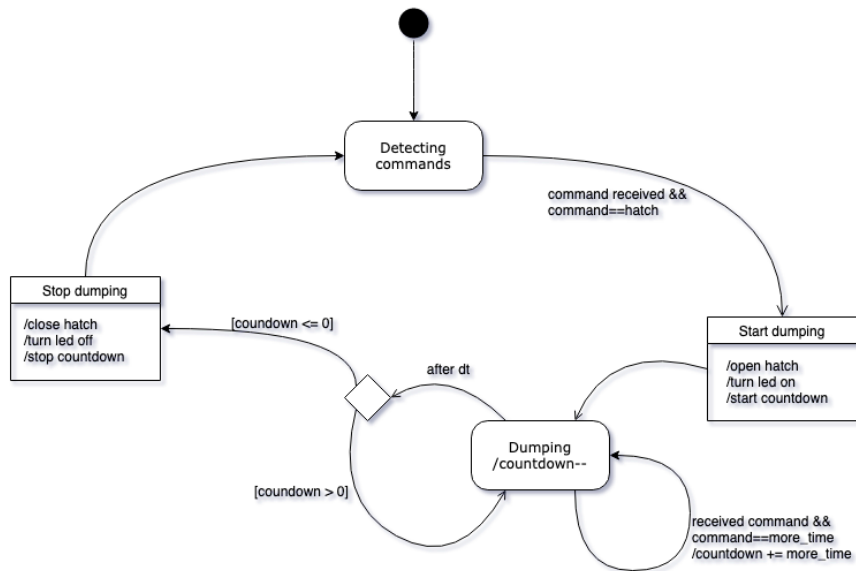


Figura 3: Diagramma di stato del Dumpster Controller

In particolare, quando la macchina viene avviata, si trova nello stato *Detecting Commands* che permette di rispondere ai comandi ricevuti via Bluetooth. Quando viene ricevuto il comando *hatch* (che indica l'intenzione da parte dell'utente di depositare dei rifiuti), la macchina passa nello stato *Start Dumping* che esegue le seguenti azioni:

- Apre il portello
- Accende il led corrispondente al tipo di rifiuto selezionato via Mobile App
- Inizia il countdown che indica il tempo per cui lo sportello rimarrà aperto

Successivamente, il Controller si sposta nello stato *Dumping* che aggiorna il countdown ed è in grado di gestire richieste da parte dell'utente di prolungare il tempo a disposizione per depositare i rifiuti. Una volta terminato il countdown, la macchina entra nello stato *Stop Dumping* che:

- Chiude il portello
- Spegne il led precedentemente acceso
- Ripristina il countdown
- Invia alla Mobile App la notifica di deposito avvenuto con successo

Una volta terminato, si ritorna nello stato iniziale *Detecting Commands*.

3 Dumpster Mobile App

La Mobile App consiste in un'applicazione per piattaforma Android che consente di interagire via Bluetooth con il Controller e via HTTP con il Service. L'obiettivo principale in questo caso era la semplicità d'uso in quanto l'applicazione smartphone risulta essere il metodo di interazione dell'utente con lo Smart Dumpster. L'interfaccia risulta perciò molto pulita come possiamo notare dalla figura sottostante.

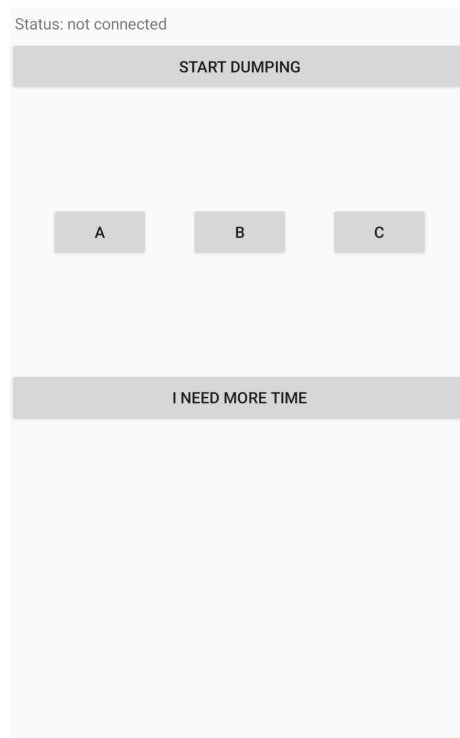


Figura 4: Schermata della Mobile App

All'apertura dell'applicazione, l'unico pulsante cliccabile risulta essere *START DUMPING* che permette di connettersi allo Smart Dumpster. Alla pressione del pulsante, l'applicazione automaticamente richiederà al Service (via HTTP) la disponibilità del Dumpster e, solamente una volta ottenuta la conferma, tenterà di connettersi al Controller (via Bluetooth).

Un aspetto importante dell'applicazione è l'utilizzo di chiamate non bloccanti: le transazioni HTTP vengono infatti effettuate sfruttando la libreria *Volley* che effettua richieste in modo **asincrono** e la connessione Bluetooth viene svolta mediante le funzioni messe a disposizione durante i laboratori le quali fanno uso di **Async Task**. Ciò permette di non bloccare il thread principale in attesa delle risposte e mantenere l'interfaccia grafica reattiva.

Una volta stabilita la connessione Bluetooth con il Controller, sarà possibile cliccare i pulsanti *A*, *B* e *C* che invieranno al Controller l'ordine di aprire il portello per il tipo di rifiuto corrispondente. Una volta selezionato il portello, sarà possibile cliccare il pulsante *I NEED MORE TIME* che richiederà al Controller del tempo aggiuntivo per terminare il deposito in corso. Alla conclusione del deposito, notificata da un messaggio del Controller (via Bluetooth), verrà informato il Service (via HTTP) il quale aggiornerà il numero di depositi effettuati.

4 Dumpster Edge

L'Edge è costituito da un SoC *ESP8266* la cui board presenta i seguenti elementi:

- Un led verde L_{avail} e un led rosso $L_{not-avail}$
- 1 sensore che misura il peso (simulato da un potenziometro)

La disposizione degli elementi è mostrata dalla seguente figura:

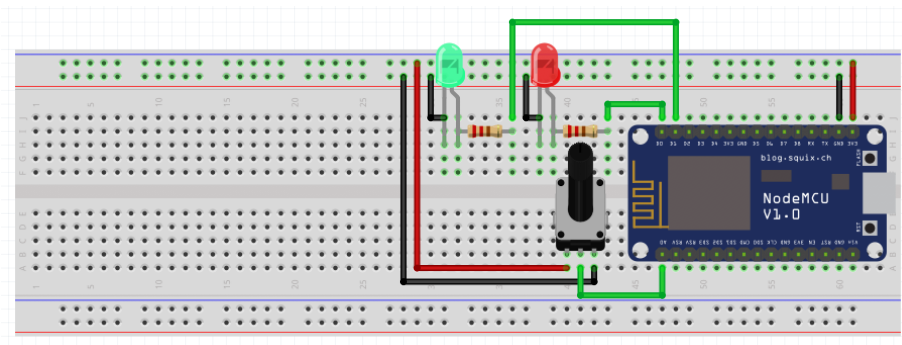


Figura 5: Schema Fritzing del Dumpster Edge

La tecnologia utilizzata per programmare il SoC è la stessa utilizzata per il Controller, ovvero la *piattaforma wiring Arduino*. Inoltre, anche l'Edge è stato progettato come **macchina a stati finiti** e di seguito è riportato il relativo diagramma di stato:

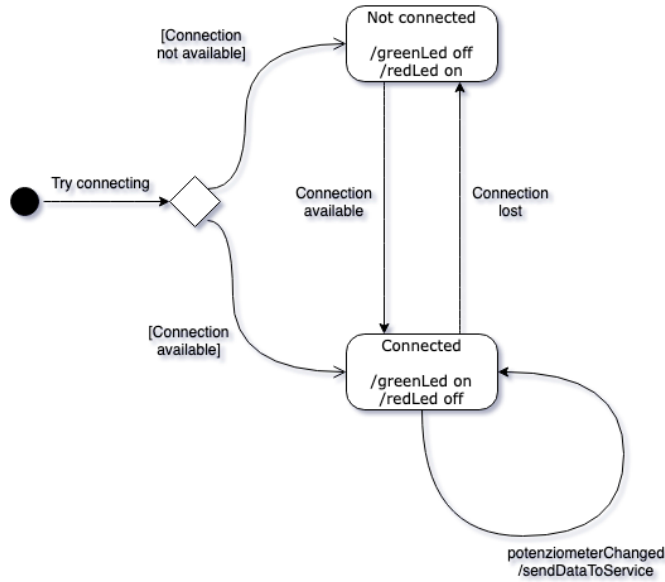


Figura 6: Diagramma di stato del Dumpster Edge

Come si può notare anche dal diagramma, il funzionamento dell'Edge è meno articolato rispetto a quello del Controller. All'avvio, infatti, se il SoC è in grado di connettersi al Service e il Dumpster è disponibile entra nello stato *Connected*, contraddistinto dal led L_{avail} acceso, altrimenti si troverà nello stato *Not Connected*, caratterizzato invece dal led $L_{not-avail}$ acceso. La perdita o il ristabilimento della connessione con il Service o il cambiamento di stato del Dumpster permettono all'Edge di passare da uno stato all'altro. Inoltre, quando la macchina si trova nello stato *Connected*, se viene rilevato un cambiamento nel valore del potenziometro questa invierà una richiesta HTTP al Service.

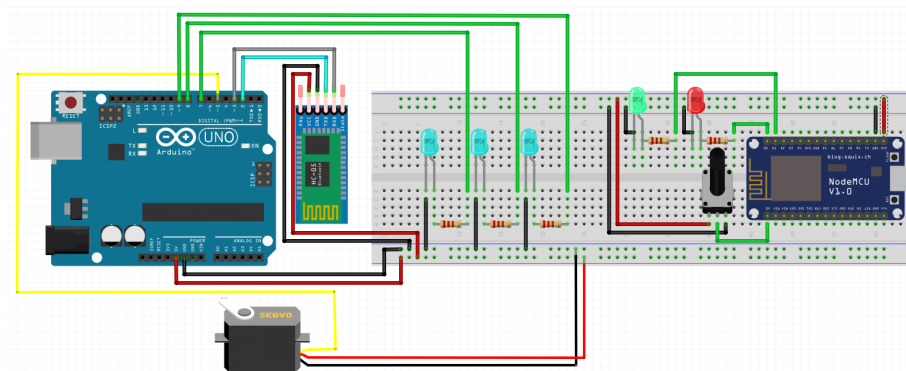


Figura 7: Schema completo del Dumpster

5 Dumpster Service

Il Service corrisponde ad un servizio REST che funge da "back end". Per la sua realizzazione si è scelto di utilizzare codice *PHP* in esecuzione su server *Apache*. La scelta è stata influenzata dall'esperienza acquisita con questa tecnologia. In particolare, il Service mette a disposizione le seguenti API:

- *is-dumpster-available* che restituisce lo stato del Dumpster (disponibile o no)
- *set-dumpster-available* e *set-dumpster-unavailable* che permettono rispettivamente di ripristinare la disponibilità del Dumpster e di interromperla
- *on-throw-success* che aggiorna il numero di depositi effettuati
- *update-weight* che inserisce un nuovo deposito con data e peso aggiornato
- *get-data* che restituisce i dati relativi al Dumpster (numero di depositi effettuati e storico dei depositi)

Per richiamare una delle precedenti API è necessario spostare la cartella *sd-service* all'interno del percorso *htdocs* di *XAMPP*. Successivamente sarà possibile accedervi da browser all'indirizzo "*localhost/sd-service/[API].php*". Per quanto riguarda i dati, essi sono mantenuti all'interno di file *.json* che garantiscono una maggiore immediatezza e semplicità d'uso rispetto ad un normale database relazionale. Troviamo infatti:

- *dumpster-availability.json* che mantiene lo stato del Dumpster (disponibile o no)
- *dumpster-data.json* che mantiene i dati relativi ai depositi (numero depositi e quantità depositate)

6 Dumpster Dashboard

La Dashboard rappresenta infine il "front end" per la visualizzazione, osservazione e analisi dei dati. È stata progettata come web app per la semplicità di interazione con il Service sviluppato in PHP. Le tecnologie utilizzate sono quelle di un classico sito web: *HTML* per la pagina, *CSS* per lo stile e *Javascript/JQuery* per la logica di funzionamento. Per garantire l'aggiornamento dinamico della pagina è stata impiegata la libreria *Ajax* mentre la libreria *Chart.js* ha permesso di visualizzare in modo opportuno i dati del Dumpster sotto forma di grafico.

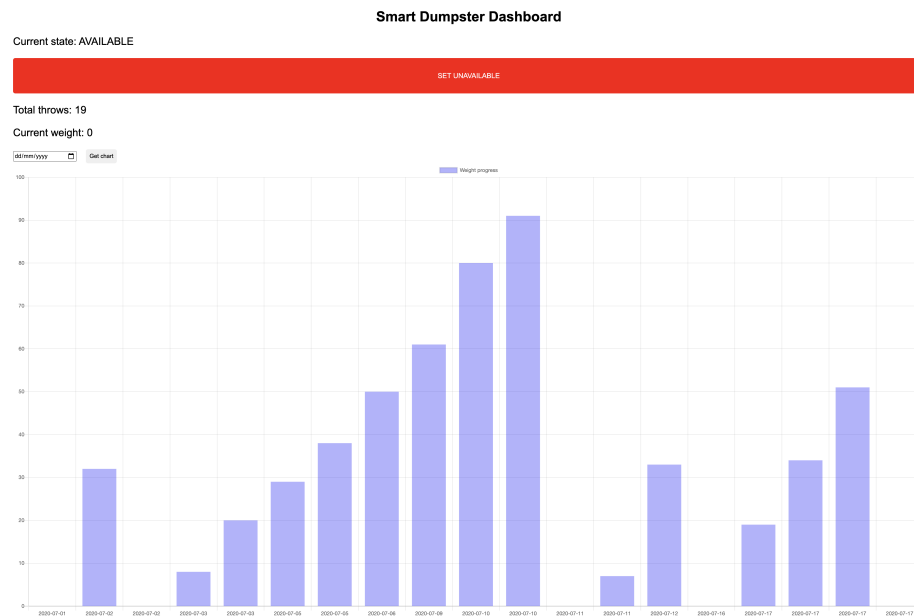


Figura 8: Schermata della Dumpster Dashboard

Come si può notare dalla figura precedente, la schermata permette di:

- Controllare lo stato del Dumpster
- Cambiare lo stato del Dumpster (il pulsante si aggiorna automaticamente in base allo stato corrente del Dumpster)
- Controllare il numero totale di depositi
- Controllare la quantità di rifiuti attualmente all'interno del Dumpster
- Visualizzare un grafico che rappresenta tutti i depositi fatti a partire da una certa data (all'avvio della Dashboard vengono visualizzati tutti)