

Report progetto fisica dei sistemi complessi

Alessandro Cordella
Natale Vadalà
alessandro.cordella@studio.unibo.it
natale.vadala@studio.unibo.it

Novembre 2018

Indice

1	Introduzione	3
2	Hardware	3
3	Software	3
3.1	Server	3
3.1.1	main.js	3
3.1.2	routes.js	4
3.1.3	AMSpi	4
3.2	Client	4
3.2.1	ex3-1.html	4
3.2.2	css	4
3.2.3	js	4
4	Ulteriori sviluppi e Conclusioni	4

1 Introduzione

L'obiettivo del presente lavoro è stato quello di costruire un robot capace di muoversi su ruote. Il dispositivo ottenuto è controllabile da un'interfaccia web ed è possibile visionare ciò che è posto di fronte al robot tramite una webcam.

Nelle seguenti sezioni verranno descritti i componenti hardware utilizzati, le tecniche software utilizzate e verranno presentati eventuali sviluppi futuri.

2 Hardware

3 Software

Per garantire il controllo del dispositivo tramite un'interfaccia web è stato necessario progettare un server al quale un client può collegarsi per ottenere il controllo del dispositivo. È stato utilizzato il framework javascript *NodeJs* per gestire sia la parte client che server dell'applicazione. Nelle seguenti sezioni verrà descritta la struttura gerarchica delle cartelle e le funzionalità presenti nei file al loro interno.

3.1 Server

I file presenti all'interno della cartella *server* contengono le funzioni per gestire la parte server del sistema, mentre all'interno delle sue sottocartelle sono presenti altri file che verranno descritti in seguito.

3.1.1 main.js

Il file *main.js* è il file principale del server. Al suo interno sono presenti le funzioni di inizializzazione del sistema e per la gestione dello streaming video.

Per avviare il server è necessario digitare il seguente comando:

```
node main.js ciao 8081 8082 8083
```

Listing 1: Avvio server

- **node** nodejs;
- **main.js** file del server;
- **ciao** parola chiave; _____
- **8081 8082 8083** qualcosa _____

cambiare
ciao

scrivere

Una volta lanciato il comando, sarà possibile accedere all'interfaccia web digitando l'indirizzo IP del raspberry seguito da *:8083*.

192.168.1.10:8083

Listing 2: Esempio indirizzo server

3.1.2 routes.js

Questo file è quello che gestisce la comunicazione client server, associando ad ogni interazione del client sulla pagina web una specifica azione del server. Gestisce quindi le funzioni che regolano la webcam e il movimento. In particolare viene utilizzata una libreria di nodejs per eseguire la libreria Python *AMSpi* per la gestione dei motori.

3.1.3 AMSpi

AMSpi è la libreria Python utilizzata per controllare i motori. Al suo interno è presente il file *movements.py*. È il file che usa le funzioni della libreria per controllare i motori.

3.2 Client

I file presenti all'interno della cartella *public* sono quelli che gestiscono la parte client dell'applicazione.

3.2.1 ex3-1.html

È il file contenente la struttura del sito web.

immagine

3.2.2 css

All'interno di questa cartella sono presenti i file che regolano lo stile della pagina.

3.2.3 js

In questa cartella sono contenuti i file che regolano le interazioni dell'utente con la pagina e comunicano al server le azioni intraprese da esso.

4 Ulteriori sviluppi e Conclusioni