

Diario di lavoro

Luogo	Canobbio
Data	16.03.2020

Lavori svolti

Oggi per prima cosa mi sono scritto tutte le cose che mi mancano da fare, in ordine, su come procedere, così da cercare di ottimizzare il mio lavoro e cercare di andare avanti più veloce possibile.

Dopo di che ho deciso di invertire le cose che dovevo fare rispetto a quanto scritto nel diario precedente, quindi devo fare un ultimo aggiustamento al codice del backend.

Nella classe `SerialThread`, dopo aver inizializzato la comunicazione seriale, devo inizializzare l'Arduino con i dati del database.

```
/**
 * Setup initial commands.
 */
private void setupInitCommands() {
    JdbcConnector connector = null;
    try {
        connector = new JapiConnector();
        connector.openConnection();
        Connection connection = connector.getConnection();

        AccGenerator.initAcc(connection, this);

    } catch (IOException | ClassNotFoundException | SQLException ignored) {

    } finally {
        if (connector != null) {
            connector.close();
        }
    }
}
```

Ho creato questo metodo che apre una comunicazione con il database e richiama il metodo `initAcc` il quale legge tutti i dati dal database ed aggiunge dei comandi alla coda di `SerialThread`, così che vengano subito inviati.

Dopo di che sono passato ad uploadare il database sul raspberry, ma prima di fare ciò ho aggiunto tutti i commenti al codice sql.

Per poi tramite eseguire tutti i file con il comando:

```
mysql -u root -p1234qwer < src/sql/db.sql
mysql -u root -p1234qwer < src/sql/triggers_generator.sql
mysql -u root -p1234qwer < src/sql/triggers_group.sql
mysql -u root -p1234qwer < src/sql/triggers_group_permission.sql
mysql -u root -p1234qwer < src/sql/triggers_mic.sql
mysql -u root -p1234qwer < src/sql/triggers_permission.sql
mysql -u root -p1234qwer < src/sql/triggers_remote.sql
mysql -u root -p1234qwer < src/sql/triggers_user.sql
mysql -u root -p1234qwer < src/sql/triggers_user_group.sql
```

Dopo di che ho provato ad avviare l'applicativo tramite gradle, utilizzando il comando:

```
cd freqline/src/freqline-be
./gradlew run
```

Avendo collegato il raspberry all'arduino.

Dopo di che con il mio portatile ho fatto la richiesta (tramite la mia rete casalinga) per provare a vedere su funzionasse, quindi ho eseguito il login ed ho provato ad accendere il generatore

```
curl -c c -b c http://10.8.16.191:8080/action/login -X POST -d "username=admin&password=123qwe"
curl -c c -b c http://10.8.16.191:8080/action/generatorStatus -X POST -d "status=1"
curl -c c -b c http://10.8.16.191:8080/action/generatorStatus -X POST -d "status="
```

Il generatore si accende correttamente e funziona. Quindi ho deciso di iniziare a spostare anche il front-end sul server. Per cui ho dovuto fare delle modifiche sia al back-end che al front-end. Per esempio gli elementi legati al CORS.

Quindi per prima cosa ho iniziato a configurare NGINX per esporre il front-end sulla porta 80. Ho deciso di utilizzare http solamente siccome l'applicativo viene utilizzato in una rete isolata e non contiene dati sensibili. Anche perché altrimenti dovrei creare una Certificate Authority per eseguire la certificazione per HTTPS e poi installare su tutti i dispositivi il certificato.

Per esporre il front-end ho dovuto prima di tutto modificare la variabile globale **baseUrl** che si trova nel file **src/freqline-be/src/main/webapp/assets/js/app.js**, modificando l'indirizzo localhost in **10.8.16.191**. Poi ho dovuto aggiornare gli elementi per il cors nel file **src/freqline-be/src/main/java/ch/giuliosbosco/freqline/servlets/BaseServlet.java**.

Infine ho iniziato a configurare il raspberry per emettere un segnale Wifi.

Quindi ho dovuto per prima cosa configurare l'ip statico per la scheda di rete wireless, inserire le seguenti linee nel file **/etc/dhcpd.conf**

```
interface wlan0
static ip_address=192.168.0.10/24
denyinterfaces eth0
denyinterfaces wlan0
```

Dopo di che ho installato i seguenti pacchetti:

```
sudo apt install dnsmasq hostapd
```

Poi ho dovuto fermare i servizi installati, per configurarli come segue. Prima cosa fare una copia del file di configurazione di **dnsmasq**

```
mv /etc/dnsmasq.conf /etc/dnsmasq.conf
```

Ed inserire nel file:

```
interface=wlan0
dhcp-range=192.168.4.2,192.168.4.20,255.255.255.0,24h
```

poi avviare dnsmasq

```
systemctl start dnsmasq
```

ed infine configurare hostapd per il controller del wifi:

```
interface=wlan0
driver=nl80211
ssid=freqline
hw_mode=g
```

```
channel=7
wmm_enabled=0
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
wpa=2
wpa_passphrase=freqline
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
```

Sia il SSID che la password della rete saranno freqline.

Poi configurare il file default (**/etc/default/hostapd**):

```
DAEMON_CONF="/etc/hostapd/hostapd.conf"
```

Ed avviare hostapd

```
sudo systemctl unmask hostapd
sudo systemctl enable hostapd
sudo systemctl start hostapd
```

Sono arrivato a testare la rete, il telefono si collega alla rete wireless, tramite WPA2.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Sono circa 8 ore avanti alla pianificazione, attività 17.

Programma di massima per la prossima giornata di lavoro

Testare che il tutto funzioni via wireless con il telefono ed il computer.