POLITECNICO DI TORINO

Alberto Simone s247818

Balduzzi Luca s248398

Canensi Emanuele s235263

Corso di Tecnologie per IoT

01UEIO

Relazione Laboratorio Hardware N°3

Scopo del laboratorio: comunicazione interfacce REST e MQTT lato Arduino.

1) Realizzare un server sulla Yun e comunicare tramite richieste HTTP. Le richieste sono state gestite con il seguente URL di base : http://<hostname>:<porta>/arduino, suddividendo le richieste per topic, in particolare http://<hostname>:<porta>/arduino/LED e http://<hostname>:<porta>/arduino/temperature. Lo scopo è quello di ricevere in base alle richieste il valore della temperatura tramite una richiesta GET sull'URL specifico mentre per il led vengono gestite le richieste POST per l'accensione e lo spegnimento del LED. Arduino risponde ad ogni comunicazione inviando un messaggio HTTP di avvenuta o mancata risposta tramite i codici 200 o 404 con body formattato secondo il coding senML.
2) Per questo esercizio sono stati realizzati due programmi, uno sketch arduino e uno script in python con il ruolo di server utilizzando il framework cherrypy. Quest'ultimo va a collezionare tutti i valori inviati da Arduino tramite richieste curl periodiche, ognuna con una lettura del valore di temperatura in formato senML. Il server infine deve gestire le richieste GET in arrivo restituendo l'intera lista formattata come un unico JSON.

Di seguito due porzioni di codice rappresentanti rispettivamente la gestione delle richieste lato server e creazione ed invio della richiesta da parte di arduino

```
def GET(self, *uri, **params):
        return json.dumps(self.requests)
def POST(self, *uri, **params):
        cherrypy.response.headers['Content-Type'] = 'application/json'
        request_body = cherrypy.request.body.read()
        request = json.loads(request_body)
        self.requests.append(request)
        return "Request saved!"
```

```
int postRequest(String data){
  Process p;
  p.begin("curl");
  p.addParameter("-H");
  p.addParameter("Content-Type: application/json");
  p.addParameter("-X");
  p.addParameter("POST");
  p.addParameter("-d");
  p.addParameter("param1=" + data);
  p.addParameter(url);
  p.run();
  return p.exitValue();
}
```

3) In questo esercizio vanno convertite tutte le funzionalità dell'esercizio precedente non più attraverso richieste HTTP bensì tramite comunicazione MQTT.

   I topic della comunicazione di MQTT sono /tiot/26/led e /tiot/26/tmp a cui Arduino deve registrarsi, nel primo per ricevere comandi riguardo la gestione del led mentre il secondo viene usato nel ruolo di publisher dove pubblica periodicamente ogni 10s informazioni sul valore di temperatura rilevato. Il tutto viene scambiato tramite stringhe JSON nel formato senML. Viene utilizzata inoltre la libreria ArduinoJSON per semplificare l'encoding delle stringhe.