Alberto Simone s247818

Balduzzi Luca s248398

Canensi Emanuele s235263

Corso di Tecnologie per IoT

01UEIOA

Relazione progetto finale

Funzioni aggiuntive

Bot di telegram

Interfaccia molto user friendly tramite il semplice utilizzo di pulsanti in cui è possibile visualizzare e tenere traccia di tutti i valori gestiti dal nostro smart home controller.

È inoltre possibile sempre grazie ad appositi pulsanti settare lo stato dell'impianto di illuminazione e settare nuovi valori di temperatura per la soglia di attivazione degli impianti di riscaldamento e raffreddamento; si può ancora settare lo stato dell'antifurto.

Tutte le funzioni sono state associate a richiesta HTTP tramite la libreria requests: sia GET per poter visualizzare tutti i valori sia POST per poterli modificare.

Catalog

Il Catalog utilizzato è una estensione di quanto sviluppato nei precedenti script python dei vari laboratori software.

In particolare ci si è focalizzati sulla progettazione di nuovi endpoint per la gestione dello sketch finale presente sulla scheda Arduino e per permettere una migliore sinergia tra eventuali servizi facenti utilizzo del Catalog.

Endpoint GET:

1) /temperature: ritorna la temperatura attuale (ultimo valore registrato dal sensore);
2) /heating: ritorna la potenza percentuale del riscaldamento attivo;
3) /avgtemperature: ritorna la temperatura media registrata;
4) /cooling: ritorna la potenza percetuale del condizionatore attivo;

5) /presence: ritorna l'effettiva presenza di persone all'interno dell'abitazione;
6) /light: ritorna lo stato dell'illuminazione;
7) /antifurto: ritorna lo stato dell'antifurto.

Endpoint POST:

1) /luciON: accende l'impianto di illuminazione;
2) /luciOFF: spegne l'impianto di illuminazione;
3) /settemperature: imposta la temperatura di riferimento (permette l'azionamento del condizionatore o dell'impianto di riscaldamento),
4) /setantifurto: in base al valore passato come parametro setta lo stato dell'antifurto.

Una scelta migliore per rispettare totalmente il protocollo REST sarebbe stata quella di implementare le richieste PUT per l'aggiornamento di alcune variabili.

Arduino

Lo sketch di arduino da caricare sulla scheda è un'estensione degli sketch dei laboratori precedenti. La prima differenza è sui topic utilizzati: ora ne abbiamo due per ogni sensore o attuatore; uno viene usato per la comunicazione Arduino->Catalog, e l'altro per la comunicazione inversa con il subtopic /res che serve ad assegnare l'ID specifico ad ogni sensore in fase di registrazione al Catalog.
Successivamente troviamo altri due topic specifici per l'accensione/spegnimento delle luci /led/onoff e per il set del valore soglia di temperatura, al topic /tmp/change.
Il controllo e azionamento delle periferiche è gestito in modo uguale ai precedenti laboratori mentre si hanno estensioni nella sezione riguardante la comunicazione per via del numero di canali.
La comunicazione MQTT in JSON è stata gestita in modo manuale sia in scrittura che in lettura. In scrittura tramite una concatenazione di char* (si sono riscontrati diversi problemi nella concatenazione per il tipo di dato String) per poi trasformare la stringa così ottenuta in String (la funzione di pubblicazione sui topic MQTT richiede il tipo String). La lettura invece è stata eseguita tramite la creazione di una funzione split ad hoc sia per la registrazione che per il setup delle luci e della temperatura.
Dopo aver completato il corpo principale della funzione loop piuttosto che l'utilizzo di una una funzione delay all'interno della propria finestra temporale si è preferito: gestire la scrittura su LCD, check della presenza del sensore PIR (si può settare dal potenziometro una durata sufficiente che copra tutto il periodo di attività della loop) e il check della notify tramite la funzione mqtt.monitor(), la quale riceve solamente un messaggio ogni volta che viene invocata.
Essendo collegato il sensore di rumore ad una procedura di interrupt si è applicato il sensor fusion nel seguente modo: ad ogni lettura del sensore PIR si confronta lo stato del sensore di rumore, ottenendo così un valore più che sicuro.

Infine per evitare valori multipli o falsi positivi sono stati settati dei buffer circolari e delle soglie minime.
Sono stati riscontrati innanzitutto problemi di memoria della scheda, per questo motivo si è scelto di eseguire sia il coding che il decoding manualmente per poter togliere la libreria relativa al JSON.
Inoltre in un sistema reale sarebbe preferibile utilizzare un numero maggiore di schede, magari collegandole come master & slave per poter distribuire il carico di lavoro in maniera più uniforme. L'utilizzo di una sola scheda ci ha portato a considerare i devices come sensori, a discapito purtroppo di velocità di esecuzione e leggerezza.