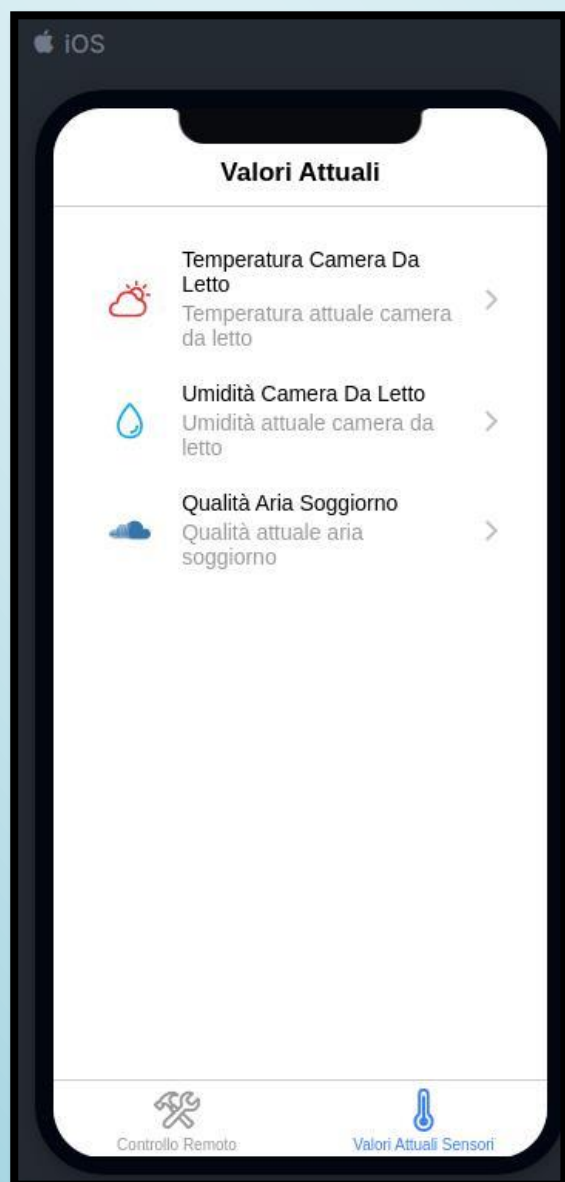


RELAZIONE



PROGETTO: ARDUINO & MQTT PER RENDERE LA CASA PIU' SMART

Docenti:

Prof. Vincenzo Catania

Prof. Maurizio Palesi

REALIZZAZIONE A CURA DI:

Arciprete Andrea

1000009556

Chiavetta Daniela

1000009554

Anno Accademico: 2020/2021

INGEGNERIA INFORMATICA

SCOPO DEL PROGETTO

L'obiettivo di base del progetto è quello di utilizzare il protocollo MQTT per rendere più smart la casa, introducendo funzionalità quali controllo remoto delle luci, possibilità di disattivare un allarme a distanza.

MQTT è, infatti, un protocollo di comunicazione di tipo publish-subscribe, progettato per essere utilizzato in situazioni in cui i dispositivi che parlano tra loro hanno una potenza di calcolo limitata o sono collegati tramite reti inaffidabili, soggette a latenza e ritardi.

Proprio per questo motivo fa a caso nostro, in quando i moduli ESP che abbiamo utilizzato sono limitati in termini di risorse.

In aggiunta, abbiamo voluto realizzare un'applicazione mobile usando il framework Angular, per consentire all'utente finale di gestire le luci, l'allarme di fumo e di temperatura da remoto e di conoscere la temperatura, l'umidità e la qualità dell'aria all'interno della propria casa.

COMPONENTI

Per realizzare il progetto, abbiamo utilizzato i seguenti componenti:

- 2 schede Arduino MEGA



- 2 moduli ESP 8266 (dal momento che le schede Arduino Mega in nostro possesso non sono dotate di un modulo WiFi)



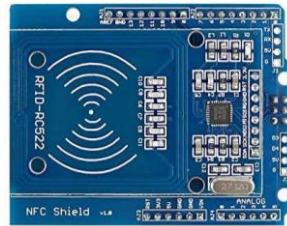
- Due buzzer (per generare il suono dell'allarme)



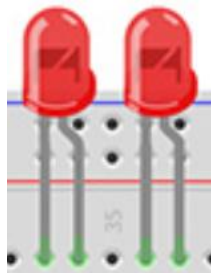
- Un sensore di temperatura



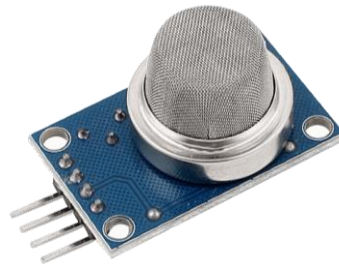
- Un sensore RFID



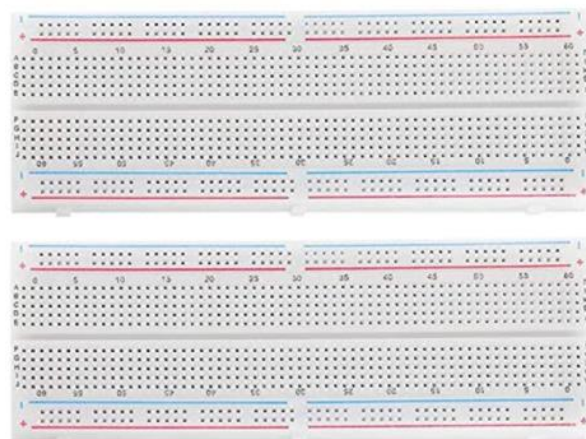
- Due led



- Un sensore di fumo



- 2 breadboard



ARCHITETTURA A BASSO LIVELLO

Per realizzare il nostro progetto abbiamo collegato a ciascuna scheda Arduino Mega un modulo ESP 8266 per usufruire delle connettività WiFi.

Alla prima scheda Arduino MEGA, oltre al modulo ESP, abbiamo collegato un buzzer per generare il suono dell'allarme, un sensore di temperatura, un sensore RFID e un led per implementare il controllo a distanza.

Alla seconda scheda Arduino MEGA, oltre al modulo ESP, abbiamo collegato un buzzer per generare il suono dell'allarme, un led per il controllo a distanza e un sensore di fumo. Questo per quanto concerne l'architettura a basso livello.

ARCHITETTURA AD ALTO LIVELLO

Abbiamo realizzato una piattaforma Middleware, utilizzando il framework Spring, che riceve i dati acquisiti dai sensori e pubblicati su opportuni topic dai moduli ESP ed espone delle API, che possono essere sfruttate dalle applicazioni IOT per accedere alle grandezze misurate dai sensori o per mandare dei comandi di attuazione, che poi si tradurranno in una publish su un opportuno topic.

Abbiamo, altresì, realizzato un'applicazione IOT utilizzando il framework Ionic, che verrà utilizzata dall'utente finale per poter gestire le luci, l'allarme di fumo e di temperatura da remoto e conoscere la temperatura, l'umidità e la qualità dell'aria all'interno della propria casa.

L'applicazione sviluppata è semplice da usare e intuitiva, si compone di tre schermate: la prima in cui sono presenti due pulsanti che consentono di accendere/spegnere il led presente nel soggiorno e accendere/spegnere il led presente nella camera da letto, la seconda che consente di richiedere i valori di temperatura, di umidità, di qualità dell'aria all'interno della casa, in particolare nelle stanze ove sono posizionate le due schede Arduino, la terza, che di fatto è una modale che viene mostrata soltanto se viene generato un allarme. Tale schermata/modale consente di disabilitare, qualora dovesse suonare, l'allarme di fumo e di temperatura da remoto.

COMUNICAZIONE

- Ciascuna scheda Arduino MEGA comunica con il modulo ESP 8266 a cui è collegata tramite la **seriale**. Questa comunicazione tramite seriale è necessaria in quanto rappresenta l'unico modo attraverso il quale la scheda Arduino può inviare e ricevere dei dati dal modulo ESP 8266.
- Il middleware e i due moduli ESP utilizzano il protocollo MQTT e, dunque, non comunicano direttamente, ma la loro comunicazione è mediata dal MQTT Broker.
- L'applicazione IOT, realizzata utilizzando il framework Ionic, utilizza le API REST esposte dalla piattaforma Middleware (applicazione Java) per il controllo da remoto e anche il protocollo MQTT. Ci è servito utilizzare il protocollo MQTT in virtù del fatto che esso

prevede un meccanismo di interazione di tipo interrupt e dunque consente di notificare all'applicazione IoT eventuali allarmi.

Vediamo un po' più in dettaglio il funzionamento.

MODULO ESP & ARDUINO

Ciascun modulo ESP (che funge sia da client publisher sia da client subscriber) lo abbiamo programmato con Arduino IDE, ossia abbiamo scritto un codice C e poi lo abbiamo caricato all'interno del modulo ESP. Nel codice C abbiamo incluso la libreria PubSubClient.h che mette a disposizione diverse funzioni per poter instaurare una connessione con il Broker, fare publish su un certo topic e sottoscrivere ad uno o più topics. Dunque, la parte relativa alla comunicazione con il Broker l'abbiamo implementata nel codice C che poi abbiamo caricato all'interno del modulo ESP, mentre la parte relativa alla logica applicativa l'abbiamo implementata nel codice C che poi abbiamo caricato all'interno della scheda Arduino MEGA.

In virtù di quanto detto, se il modulo ESP si è sottoscritto ad un certo topic, non appena il publisher (piattaforma Middleware) pubblica su quel topic, il Broker inoltrerà al modulo ESP ciò che riceve. Il modulo ESP, ricevuto tale messaggio (PUBLISH), lo processa e tramite la seriale manderà qualcosa alla scheda Arduino, che sulla base di ciò che riceve comanderà opportunamente gli attuatori.

Se, invece, i sensori collegati alla scheda Arduino acquisiscono il valore di una grandezza, la scheda Arduino tramite la seriale trasmette il valore acquisito al modulo ESP, che pubblicherà su un opportuno topic il valore acquisito dai sensori, che gli è stato trasmesso dalla scheda Arduino. Tutti i sottoscrittori al topic (nel nostro caso la piattaforma Middleware) riceveranno tale valore.

MODULO ESP & MODULO ESP

Abbiamo implementato anche una comunicazione mediante il protocollo MQTT tra i due moduli ESP (comunicazione sempre mediata dal Broker). In particolare, quando l'utente avvicina il tag RFID al lettore RFID, se l'esito della scansione è positivo, l'Arduino MEGA 1 lo comunica tramite seriale al modulo ESP 1, che pubblicherà su un certo topic, a cui si è sottoscritto l'altro modulo ESP. L'altro modulo ESP non appena riceve il messaggio pubblicato dal primo modulo ESP tramite seriale manderà un comando di attuazione all'Arduino Mega 2 che provvederà opportunamente ad accedere il led in soggiorno.

PIATTAFORMA MIDDLEWARE

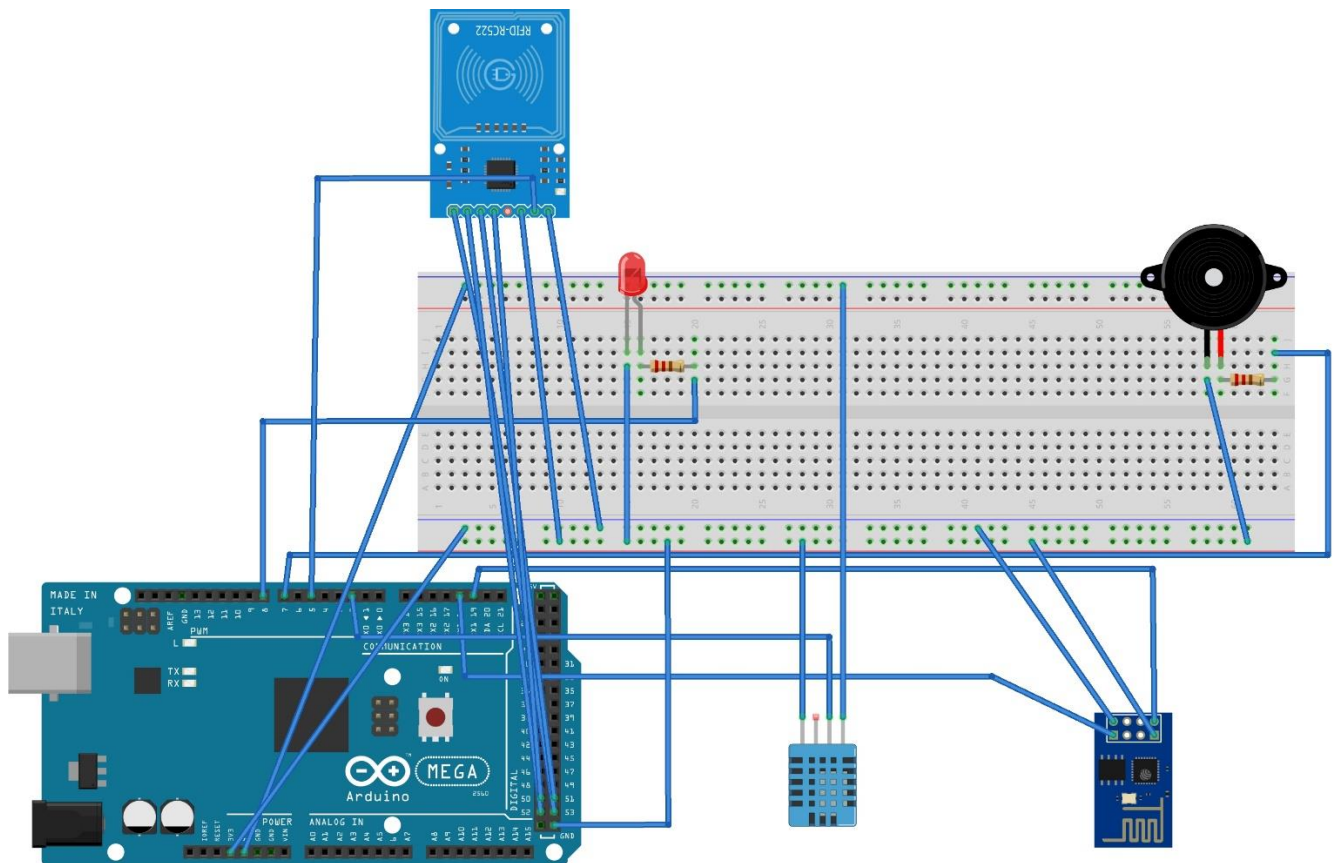
La piattaforma middleware da un lato pubblica messaggi su vari topic, topic a cui si sono sottoscritti i due moduli ESP. In tal modo, ciascun modulo ESP non appena riceve il valore pubblicato dal Middleware (si ricorda che la comunicazione è sempre mediata dal Broker) trasferisce quanto ricevuto tramite la seriale alla scheda Arduino, che comanderà opportunamente gli attuatori.

Dall'altro lato, la piattaforma middleware si sottoscrive ad alcuni topic, topic su cui i due moduli ESP faranno le publish.

fritzing

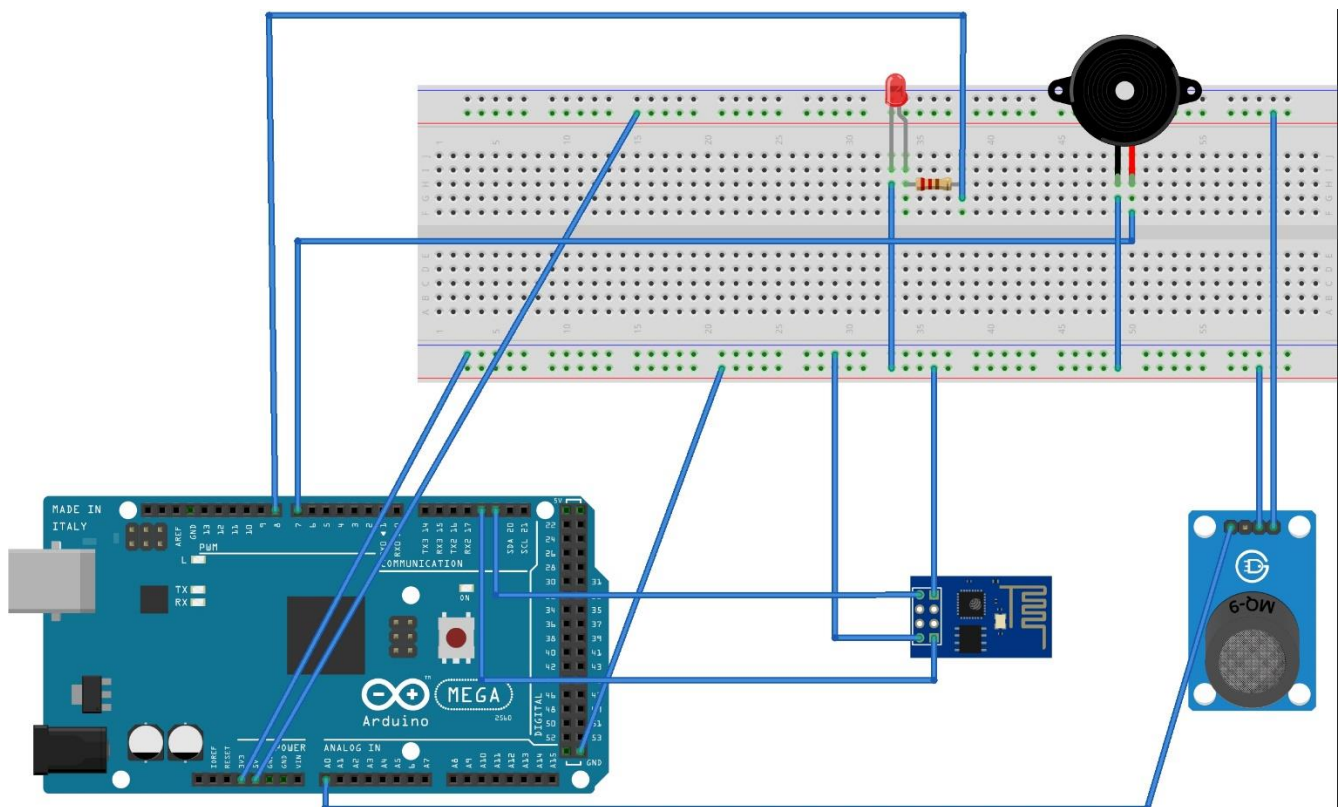
COLLEGAMENTI FINALI PER TESTARE LE FUNZIONALITA':

ARDUINO MEGA 1



fritzing

ARDUINO MEGA 2



fritzing