Relazione Smart Door

Progetto di: Luca Mondaini, Nicolò Pracucci, Daniele Tentoni

Sistemi Embedded e Internet of Things

# 

[**Schema Fritzing**](#_2cefjjvcuput) **2**

[**Sottosistema Arduino**](#_lkdprjfzt117) **3**

[**Diagramma degli stati Arduino**](#_rovk6l4mpm2p) **4**

[**Sottosistema Raspberry**](#_c8bc2zaybksq) **5**

[**Sottosistema Android**](#_oat313gwle2j) **6**

# 

# Schema Fritzing

# 

# Sottosistema Arduino

Si vuole creare tramite Arduino un software che permetta il funzionamento di una porta intelligente.

All’inizio lo stato iniziale dell’ambiente sarà la porta chiusa in attesa di una connessione da parte dell’applicazione che userà l’utente(stato di IDLEE).

In questo stato il sistema si troverà spento con il servomotore (che funge da simulatore di porta) impostato su 0 gradi in attesa di ricevere un segnale per partire.

Appena l’utente invierà una richiesta di connessione tramite l’app per smartphone al bluetooth si potra passare allo stato di WAIT.

Durante questo stato si avvierà il sensore di prossimità con il compito di identificare se l’utente si trova effettivamente davanti alla porta di ingresso ad una distanza minima di MIN\_DIST per una quantità di tempo di MIN\_SEC\*WAIT\_DOOR/1000 secondi (nel programma equivalgono a 3 secondi).

Confermato ciò, Arduino manderà un messaggio a Raspberry “Qualcuno è rimasto davanti alla porta per MIN\_SEC\*WAIT\_DOOR/1000 secondi” e all’applicazione “@pres@”.

Fatto ciò, si potrà passare allo stato di PRESENT.

Nello stato di PRESENT Arduino, Raspberry e l’app si occuperanno dell’autenticazione dell’utente nel sistema.

Dall’app, Arduino riceverà le credenziali dell’utente, successivamente le invierà a raspberry (che fungerà da server), Raspberry notificherà Arduino sulla correttezza delle credenziali inserite.

In caso affermativo risponderà “logok” in caso negativo “logno”.

Arduino notificherà successivamente l’applicazione che agirà di consenguenza.

Si passa così alla stato PASSATO.

In questo stato si attiverà il PIR, in caso rilevi l’utente Arduino notificherà Raspberry con la frase “Il PIR ha rilevato il movimento di qualcuno nella stanza”, notificherà anche l’applicazione con “@visto@” e passerà allo stato di VISTO.

Nel caso in cui il PIR non rilevi nessun movimento entro MAX\_DELAY\*WAIT\_DOOR/100 secondi (nell’applicazione equivale a 9 secondi), notificherà l’applicazione con “@exit@” e la Raspberry con la frase “Il PIR non ha rilevato il movimento di qualcuno nella stanza.”

Nello stato di VISTO Arduino attiverà il sensore di temperatura.

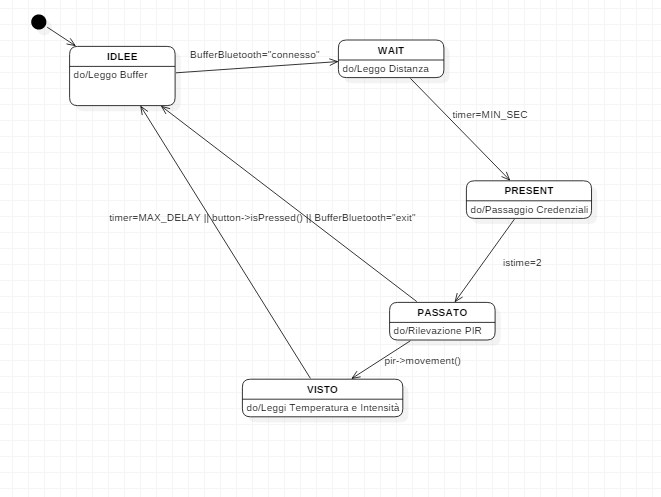
Ogni misurazione che farà il sensore di temperatura verrà notificato sia all’applicazione che a Raspberry (che gestisce il sito internet) in modo che si potranno visualizzare le informazioni in tempo reale.

Se dall’applicazione arriva in seriale un messaggio contenente un valore da 0-100 (percentuale di intensità del sensore), Arduino provvederà ad attuarla al led rosso.

Arduino comunicherà questo cambiamento anche a Raspberry.

In caso l’utente clicchi sul bottone exit o l’utente chiuda l’applicazione, Arduino ritornerà nello stato di IDLEE notificando a Raspberry “L'utente ha effettuato l'uscita dall'applicazione, la porta verrà chiusa.” e all’applicazione “@exit@”.

# Diagramma degli stati Arduino



# Sottosistema Raspberry

Il sottosistema Raspberry fungerà da web server: nella cartella /var/www/html della nostra Raspberry è stata inserita la cartella /IoT che contiene la pagina index.php e tutti i suoi componenti in modo tale che collegandosi a [http://IP\_RASPBERRY/IoT](http://ip_raspberry/IoT) verrà visualizzata la pagina di log del nostro sistema SmartDoor.

Tramite il Makefile all’interno del progetto Java, la Raspberry compilerà tutto il codice e avvierà in automatico l’applicazione. Una volta avviata l’applicazione, questo sottosistema cercherà e si connetterà alla porta seriale alla quale sarà collegato il [sottosistema Arduino](#_lkdprjfzt117).

Dal momento che questi due sottosistemi si saranno collegati, la Raspberry resterà in ascolto sulla porta seriale dalla quale Arduino gli manderà i vari stati in modo che essa si possa comportare di conseguenza.

Gli stati sono:

* **PRESENT:**

quando un utente si presenta davanti alla porta e si connette al bluetooth la Raspberry dovrà aggiornare il proprio file di log notificando la presenza di una persona.

* **ACCOUNT:**

quando un utente presente davanti alla porta e connesso al bluetooth prova a loggarsi, la Raspberry dovrà controllare se quell’utente può accedere o meno.

Nel caso possa accedere, notificherà nel log l’accesso, sennò notificherà il tentativo errato e farà lampeggiare una volta il led rosso.

* **VISTO:**

quando un utente entra dentro la stanza e viene captato dal sensore di movimento, la Raspberry lo notificherà nel log.

Inoltre la Raspberry accenderà il led verde.

* **TEMP:**

ogni volta che Arduino segnalerà l’aggiornamento del valore della temperatura che c’è nella stanza, la Raspberry aggiornerà il valore anche nel sito tramite il file temp.txt.

* **INT:**

ogni volta che un utente modificherà l’intensità luminosa del led della stanza, la Raspberry aggiornerà il valore anche nel sito tramite il file int.txt .

* **EXITPIR:**

se dopo 9 secondi da quando la porta viene aperta nessuno viene rilevato dal sensore di movimento nella stanza, la Raspberry lo notificherà nel log.

* **EXIT:**

ogni volta che un utente esce dalla stanza e tramite l’applicazione [Android](#_oat313gwle2j) chiude la porta, la Raspberry lo notificherà nel log.

Per poter vedere continuamente gli aggiornamenti del log, index.php è stata fatta in modo che si aggiornerà automaticamente ogni secondo.

# Sottosistema Android

Il sottosistema Android consentirà all’utente di avere un’interfaccia al sistema che gli permetta di aprire la porta, regolare l’intensità di una fonte luminosa al suo interno e di avere un feedback costante riguardo la temperatura interna. L’applicazione sviluppata potrà comunicare unicamente con il sottosistema Arduino, tramite il controller bluetooth installato su di esso.

Per iniziare a lavorare è necessario instaurare la connessione bluetooth. Prima di avviare l’applicazione, associare il dispositivo bluetooth tramite il pannello delle impostazioni. Nel caso dei test tale dispositivo è denominato: isi-21. Successivamente, è possibile avviare il sottosistema. Nella parte bassa dell’applicazione si trova una label per fare il log di alcune comunicazione utili per l’utente. Cliccare sull’unico bottone abilitato di instaura connessione. Qui si cercherà di instaurare effettivamente la connessione con il dispositivo precedentemente associato, e in caso affermativo e di permanenza davanti al sensore di prossimità per tre o più secondi, verrà salutato l’utente e saranno abilitate sull’applicazione le textbox per l’immissione del nome utente e della password. Cliccando sul bottone accedi vengono inviate le credenziali ad arduino, e nel caso in cui siano corrette, viene richiesto all’utente di avanzare dentro alla stanza e farsi notare dal sensore di presenza.

Quando l’applicazione riceverà il segnale che l’utente si è fatto notare, verrà creata la seconda activity. Da qui, tramite lo scorrere dello slider, sarà possibile regolare l’intensità della luce e rilevare in tempo reale la temperatura della stanza.

visualizzare il log sugli accessi e la temperatura e la luminosità in tempo reale.