SMASHBOX

**FUNZIONAMENTO GENERALE:**

SMASHBOX è un sistema che simula l’utilizzo delle classiche cassette di sicurezza della banca in modo più accessibile e smart rispetto allo scenario attuale.

Si compone di un N cassette di sicurezza (chiamate ACQUISITION) (il nostro prototipo ne comprende 2) e di una supplementare che funge da controllo per le altre (chiamata CENTRAL).

Ogni cassetta di sicurezza può essere registrata da un cliente tramite l’impronta digitale della mano, in modo da avere una chiave univoca e personale. Ogni cassetta ha la possibilità di tracciare lo stato interno della cassetta con dei parametri come temperatura, umidità, verificare se la cassetta è aperta o meno e se c’è stata una tentata manomissione. Il monitoring di questi parametri è utile per notificare il cliente in caso di tentativo di furto (manomissione) o di pericolo (aumento temperatura). Inoltre viene fornito un log con le diverse date e orari di apertura.

La notifica dei log e degli allarmi sfrutta un bot Telegram che viene registrato nel momento in cui il cliente acquista la possibilità di utilizzare una cassetta.

In caso di allarme il sistema si mette in quella che viene definita “SAFE-MODE”, ovvero tutte le cassette vengono bloccate, e solo un operatore può effettuare lo sblocco tramite un pulsante dedicato presente sul CENTRAL.

(FOTO SMASHBOX)

# MACRO COMPONENTI software

1. **LOCAL**: parte di acquisizione dei dati di monitoring e spedizione verso il BRIDGE.
2. **BRIDGE**: parte di recupero e inoltro verso il CLOUD.
3. **CLOUD**: gestione in cloud dei Digital Twin.
4. **CLIENT**: agenti esterni che possono accedere al CLOUD tramite interfaccia

# DETTAGLIO COMPONENTI

**LOCAL**

Questa sezione è divisa in due parti:

* Acquisition: la board scelta per leggere i dati dai sensori. Gestisce strettamente l’acquisizione dei dati dai sensori e l’inoltro tramite un canale (seriale o Wi-Fi) verso il Bridge.
* Central: Un Arduino dedicato gestisce il riconoscimento dell’impronta digitale. Centralizzata perché ci sarà un unico riconoscitore per tutte le cassette.

**ACQUISITION**

Questo componente è composto da

* Arduino
* Sensore di temperatura e umidità (Adafruit\_AHTX0)
* Accelerometro (ADXL345)
* 2 sensori di presenza (sportello e presenza oggetto)
* Un elettromagnete sostiene il blocco dello sportello quando è in modalità lock.

**CENTRAL:**

L’altra parte del LOCAL, quella centralizzata che effettua sia le registrazioni che le richieste di sblocco da parte dell’utente.

Questa è composta da:

* Sensore di impronte (Adafruit\_Fingerprint)
* Display di servizio
* Un pulsante per resettare la safe-mode (v. dopo)

**BRIDGE**

Il Bridge è uno script Python che ha due endpoint che lavorano in sinergia: uno lato LOCAL e uno lato CLOUD. L’idea di base è che il BRIDGE passi continuamente i dati che arrivano da LOCAL verso CLOUD (e viceversa). Il pacchetto di monitoring arriverà in un'unica trasmissione gestita da un protocollo (v. dopo). Il BRIDGE spacchetta tramite un Parser i dati ed effettua i comandi di update (HTTP/MQTT) verso il CLOUD in maniera separata.

Il BRIDGE è anche il punto di comunicazione tra CENTRAL e le varie ACQUISITIONS, dal momento che ci sono casi in cui il CENTRAL deve comunicare un informazione alle possibili N boards che sono le varie cassette.

**CLOUD (THINGSBOARD)**

La piattaforma scelta per il Cloud è Thingsboard. Qui avverrà il monitoring dei dati nel corrispondente Digital Twin e la gestione dei trigger scaturiti sulla base dei dati che arrivano dal Bridge.

Sostanzialmente raccoglie di ogni cassetta e comunica a tutte le altre di intervenire in caso di Safe Mode.

**TELEGRAM BOT**

E’ un endpoint in cui l’utente viene notificato di aperture della propria cassetta, di eventuali allarmi. Inoltre viene data la possibilità al cliente tramite comando al bot di capire quali sono stati i vari log della propria cassetta registrata.

***/\* MQTT SERVER(MQTT)***

*C’è un ulteriore blocco, che sarà un broker MQTT sul server dove appoggiarsi per scambiare dati tra BRIDGE e THINGS. L’idea è che i trigger di allarme siano mandati su questo server (test.mosquitto.org) e recuperati dal BRIDGE sullo stesso. Questo perché Thingsboard nella sua versione free possiede soltanto la versione per scambiare i dati verso il server e non dal server.*

***SERVER CUSTOMERS (FIREBASE)***

*L’idea di accedere ad un server per visualizzare il log e di poter sottoscrivere per una nuova cassetta potrebbe essere fatta tramite una app che fa delle richieste ad un Server dedicato in cui sono contenuti i log delle cassette. Lo stesso server contiene anche la lista dei clienti registrati che tentano di accedere allo sblocco. Questo non solo può essere usato per il log, ma anche come backup dei clienti registrati sul central per accedere alle cassette. Tutto questo sarà gestito con un Real Time Database di Firebase.*

*Lo schema generale potrebbe essere quello linkato al repository: \Doc\Blocks \*/*

# FUNZIONAMENTO

**ANALISI GENERALE DEL SISTEMA:**

Il sistema è composto da tre componenti principali: due cassette di sicurezza (ACQUISITION), una cassetta centralizzata (CENTRAL), un bridge software (scritto in Python) e un bot Telegram.  
Quando viene acceso, le cassette ACQ e la centralina CNTRL entrano in una fase di attesa in cui si mettono in comunicazione con il bridge. Il bridge identifica correttamente le unità e abilita la centralina a iniziare il processo di registrazione delle impronte digitali.  
In questa fase, il sistema associa un'identità a ogni cassetta (simulando la registrazione di un utente) e inizia la comunicazione tra le parti.

Una volta assegnati gli ID, le cassette iniziano a rilevare vari parametri ambientali e di sicurezza. I dati vengono inviati regolarmente al bridge, che li inoltra al server. Tali cassette sono sbloccabili tramite il riconoscimento dell’impronta digitale precedentemente registrata.

Il sistema può funzionare in due modalità: normale (IDLE) e di allarme (SAFE MODE). La modalità SAFE MODE viene attivata se vengono rilevati eventi sospetti come urti, aperture non autorizzate, o valori anomali di temperatura/umidità. In tal caso, il sistema reagisce localmente (con LED e blocchi fisici) e invia una notifica anche al bot Telegram.

**ANALISI TENICA DEL SISTEMA:**

Il sistema avviene su fasi che si succedono una dietro l’altra. Il seguente schema a blocchi descrive il processo delle varie parti.

// TODO: INSERIRE UNO SCHEMA A BLOCCHI

**RECOGNITION**

Allo startup, il BRIDGE effettua una scansione delle seriali che sono collegate al sistema. L’obiettivo è capire quali delle seriali rappresenta ACQ e quale rappresenta CNTRL.

Per farlo, CNRTL manda sulle seriale un pacchetto identificativo (7 byte) di tutti 0 e come ACQ si mettono in “ascolto” del BRIDGE stesso. Valutando quale serial ha inviato il pacchetto identificativo, BRIDGE riesce a capire quali sono le seriali delle cassette, e quale quella di CNTRL.

*from central\_proto.ino, in setup ()*

  /\* Sending a "0" to let the bridge recognize Central \*/

Serial.write(idle\_packet\_init, sizeof(idle\_packet\_init));

A questo punto, la parte CNTRL che era in attesa viene risvegliata dal bridge tramite l’invio sulla seriale (quella appunto riconosciuta) un carattere identificativo per procedere. Le ACQ restano ancora in attesa.

Se Central viene alimentato prima dell’avvio del bridge stesso, verrà visualizzato il messaggio seguente sul display di servizio:

“Wait for bridge…”

**ENROLL**

Quello che le ACQ stanno attendendo è l’ID. Poiché si sta simulando il primo power-on del sistema, le cassette fin quando non sono registrate, non possono partire ad acquisire i dati.

*from acquisition.ino, in setup ()*

Immagine che contiene testo, schermata, Carattere, software

Il contenuto generato dall'IA potrebbe non essere corretto.

Come si nota, ogni singola ACQ aspetta un pacchetto identificativo di 10 byte da parte del BRIDGE, con il primo valore (buffer[0]) contenente l’ID stesso. Fino a quando non arriva questo pacchetto, il sensore non è partito.

Il CNTRL, risvegliato, entra in fase di enroll, dove aspetta due registrazioni di due cliente differenti (come le cassette del sistema). La registrazione avviene tramite impronta digitale, che viene scannerizzata in due passaggi per accertarsi una corretta cattura dell’immagine.

Ogni volta che viene registrata una cassetta, da CNTRL viene inviato verso BRIDGE un pacchetto con di 2byte, uno che identifica l’header (PACKET\_ENROLL) e l’altro byte l’ID stesso. Questo ID è generato casualmente all’interno del CNTRL in un range compreso tra 1 e 128.

Il bridge riceve questo pacchetto con l’ID, lo estrae e lo invia nel pacchetto di 10 byte (tutti 0 tranne il meno significativo contenente proprio l’ID) ad una delle box non ancora registrate. Si traccia quali sono registrate e quali no, tramite un dizionario Python in RAM.

Successivamente, viene ripetuta la registrazione di una nuova impronta in modo da replicare il processo per il secondo cliente/cassetta.

A questo punto ACQ riceve i 10 byte, assegna l’ID e termine la sua routine di setup().

Anche CNTRL termina la sua routine di setup().

**3.2 SAFE MODE**

Prima di descrivere cosa fanno nella loro parte applicativa le cassette ACQ e il CNTRL, è necessario definire in quali modalità il sistema (nell’intero) può ritrovarsi:

* IDLE (funzionamento normale)
* SAFE MODE (caso di allarme)

Il secondo caso viene triggerato dal server: quando i dati di ACQ giungono a questo, vengono elaborati. In particolare, se viene effettuata una manomissione (accelerometro) oppure temperatura e umidità superano alcune soglie, tramite un broker MQTT viene inviato al BRIDGE un messaggio, il quale notifica l’intera parte “local”: sia CNTRL che ACQ.

In particolare questo avviene…

La prima si mette in modalità SAFE MODE da cui si può uscire soltanto con una operazione manuale da parte di un operatore (simulato con un push button sul retro). Le seconde entrano anch’esse in modalità SAFE MODE (led lampeggianti) in cui il pacchetto viene riempito da tutti 0 tranne il Lock.

Un messaggio di allarme viene inviato anche al BOTTG.

**MAIN LOOP ACQUISTION:**

Una volta che il CNTRL ha assegnato l’ID, ogni singola ACQ rileva in real-time i vari dati dai sensori installati all’interno delle cassette. In particolare:

* Temperatura + Umidità
* Accelerometro
* Presenza oggetto
* Rilevamento sportello aperto o no

I dati vengono mandati sulla seriale ad ogni ciclo al BRIDGE in un pacchetto di 7 byte:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID (1 byte)** | **Presence (1 byte)** | **Temperature (1 byte)** | **Humidity (1 byte)** | **Infra (1 byte)** | **Lock (1byte)** | **Open (1byte)** |

Il main loop di acquisizione è continuo, alla fine di ogni ciclo verranno comunque spediti al BRIDGE i 7 byte. Quello che può cambiare è il contenuto di questo pacchetto, sia nella rilevazione (i dati cambiano in maniera naturale sulla base di quello che devono rilevare), oppure se si è in modalità SAFE MODE.

**MAIN LOOP CENTRAL:**

**3.2 SAFE MODE**

Le cassette e il CNTRL funzionano in due modalità

* IDLE (funzionamento normale)
* SAFE MODE (caso di allarme)

Il secondo caso viene triggerato dal server: quando i dati di ACQ giungono al server, questi vengono controllati. In particolare, se viene effettuata una manomissione (accelerometro mosso) oppure temperatura e umidità superano alcune soglie, tramite un messaggio MQTT viene inviato al BRIDGE che lo raccoglie e questo notifica la parte local: sia CNTRL che ACQ. La prima si mette in modalità SAFE MODE da cui si può uscire soltanto con una operazione manuale da parte di un operatore (simulato con un push button sul retro). Le seconde entrano anch’esse in modalità SAFE MODE (led lampeggianti) in cui il pacchetto viene riempito da tutti 0 tranne il Lock.

Un messaggio di allarme viene inviato anche al BOTTG.

**3.3 ACQUISITION**

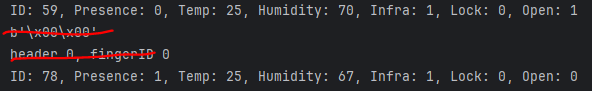
Una volta che il CNTRL ha assegnato l’ID, ogni singola ACQ prende i dati dai sensori

* Temperatura + Umidità
* Accelerometro
* Presenza oggetto
* Rilevamento sportello aperto o no

Se in modalità IDLE (e quindi non in SAFE MODE) Tutti questi dati vengono mandati ad ogni ciclo al bridge in un pacchetto di 7 byte.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **ID (1 byte)** | **Presence (1 byte)** | **Temperature (1 byte)** | **Humidity (1 byte)** | **Infra (1 byte)** | **Lock (1byte)** | **Open (1byte)** |

ESEMPIO DAL TERMINALE:



Altrimenti, SAFE MODE vuol dire che il pacchetto sarà costituito da 7 byte, ma tutti a 0 tranne il lock uguale a 1.



**3.4 FINGERPRINT MODE (CHECK)**

A questo punto Central aspetta che qualcuno voglia sbloccare le cassette tramite il riconoscimento di un impronta digitale. Una volta appoggiato il dito sul sensore, la cassetta verrà sbloccata sulla base di quale dito è stato registrato (ricordando che ogni dito simula un cliente di una cassetta).

Una volta sbloccata, una cassetta si apre e il LED assegnato a tale cassetta è acceso. Si può simulare la presenza dell’oggetto inserendo al proprio interno un ostacolo da rilevare. La cassetta va aperta (sensore di apertura riconosce che lo sportello è stato aperto solo se è molto aperto) a questo punto il magnete si riattiva, pronto per la richiusura.

Lo sblocco avviene quando il central riconosce che esiste una delle cassette registrate. Se c’è questo match, allora

**3.4 ALLARMI PER SAFE MODE**

Durante l’acquisizione dei dati, il server può elaborare alcuni campi per triggerare un allarme. In particolare, viene armato un warning quando i campi della temperature e di manomissione (infrangment) vagliano una soglia predefinita. Da ACQ infatti verranno spediti i dati in presi in real time, e il server una volta elaborati, verificherà se ci sono gli estremi per l’invio.