

DISPOSITIVO IoT PER IL MONITORAGGIO DI PERSONE NON AUTO-SUFFICIENTI CON NOTIFICA SU SMARTPHONE

INFORMATICA INDUSTRIALE E IOT

ANNO ACCADEMICO

2017/2018

Tesina di:

MATTEO PULEGA

Sommario

Introduzione	2
Sensori e attuatori	3
Sensore infrarosso PIR HC-SR501	3
Fotoresistenza LDR	4
LED	5
Schema di massima di collegamento ad Arduino	5
Microcontrollori e unità di processing/bridge	6
Arduino	6
Processing	7
Telegram	7
Protocolli di comunicazione	8
Trasmissione seriale	8
Protocollo HTTP	8
Prototipo	10
Funzionamento	10
Bot Telegram	11
Invio del messaggio	11
Immagine del prototipo	14

Introduzione

In alcuni nuclei familiari è possibile avere componenti non completamente auto-sufficienti, quali anziani, malati o con difficoltà motorie. Prendendo come esempio un caso personale, si è voluto individuare un metodo per monitorare e successivamente notificare la presenza ed eventuali problemi di un anziano con difficoltà motorie, in questo specifico caso, nella stanza da bagno. I messaggi che verranno inviati corrisponderanno alla presenza della persona in bagno e al caso di pericolo dato dall'assenza di movimento per un periodo di tempo prestabilito, con ulteriori messaggi che facciano ricordare la necessità di aiuto.

L'idea principale consiste nell'inviare un messaggio via Telegram, un servizio di messaggistica istantanea, sfruttando la creazione di un gruppo al quale possano partecipare vari componenti della famiglia e un bot adibito all'invio delle notifiche. In questo modo uno o più famigliari avranno la possibilità di andare in aiuto dell'anziano. In particolare, è stato preso come caso specifico un eventuale malessere che possa far perdere i sensi: tramite un **sensore infrarossi** è possibile infatti individuare se non vi sono più movimenti e, dopo un determinato periodo di tempo, notificare i famigliari del possibile pericolo.

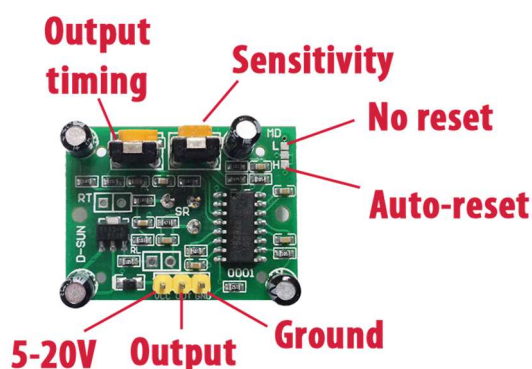
Viene utilizzato inoltre un altro sensore, una **fotoresistenza**, per un primo accertamento della presenza di qualcuno in bagno: questa infatti, nel caso reale, verrebbe inserita all'interno della scatola da parete in cui vi sono gli interruttori e una luce che si attiva all'accensione della lampada da parete presente nella stanza. Ovviamente vi sono innumerevoli altri tipi di sensori che si potrebbero utilizzare per questo tipo di controllo.

Fotoresistenza e sensore infrarossi sono collegati ad un microcontrollore, Arduino, insieme ad un **led** di colore rosso. Tale luce serve per segnalare in modo visivo la necessità di aiuto in caso di malessere, eventualmente installata al di fuori del bagno. Passato un intervallo di tempo prestabilito, verrà inviato un messaggio di allerta al gruppo Telegram.

È stata infine utilizzata come unità di **bridge** un computer con software Processing che potesse inviare richieste HTTP per il servizio di messaggistica via bot di Telegram.

Sensori e attuatori

Sensore infrarosso PIR HC-SR501



Il PIR HC-SR501 è un sensore a infrarossi in grado di rilevare il movimento di un corpo caldo e fornire di conseguenza un segnale alto a 3V. PIR è l'abbreviazione di "Passive Infra-Red". Il sensore è protetto da una semisfera di plastica, utilizzata per raccogliere tutte le fonti infrarosse che lo circondano.

Nella parte posteriore sono presenti due **trimmer**:

- Controllo del delay (output timing), ovvero quanto tempo rimane alto il segnale dopo il movimento (minimo 3 secondi, massimo 5 minuti). Ruotando in senso orario si ottiene il massimo del tempo, al contrario per il minimo del tempo.
- Regolazione della sensibilità, ovvero a che distanza è necessario rilevare il movimento (minimo 3 metri, massimo 7 metri). Ruotando in senso orario si raggiunge la minima distanza, al contrario per la massima.

Questo sensore possiede inoltre un **jumper** costituito da 3 poli, collegabili in due modi:

- Pin centrale + pin L (no reset): il segnale dopo un movimento diventa alto e rimane alto per tanti secondi quanti sono stati settati tramite il trimmer apposito. Se vi fosse un ulteriore movimento durante il periodo di segnale alto, il tempo non viene resettato.
- Pin centrale + pin H (auto reset): ad ogni movimento rilevato il segnale viene settato ad alto e lo rimane fino al termine del tempo stabilito. Ad ogni nuovo movimento il tempo viene resettato.

I tre **pin** sul lato opposto rispetto ai due trimmer sono rispettivamente alimentazione (5 – 20V), Output e Ground (massa).

Caratteristiche principali del sensore PIR HC-SR501:

- Tensione di alimentazione: 5–20 V
- Distanza rilevamento: 3–7 metri
- Tensione di uscita: 0-3,3 V
- Durata del segnale alto: 3-300 secondi
- Angolo di rilevamento: 110°

Fotoresistenza LDR

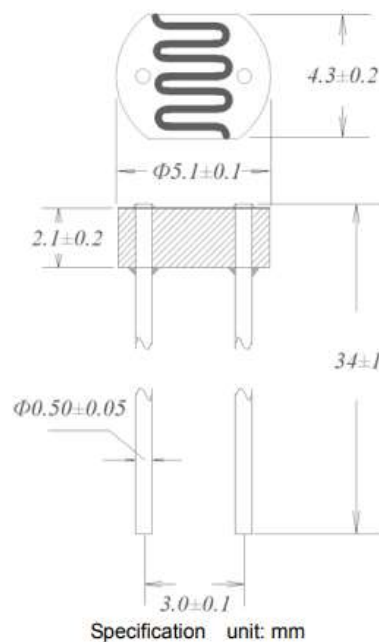
La fotoresistenza LDR è un sensore composto da uno strato semiconduttore che reagisce alla radiazione luminosa: al variare dell'intensità luminosa varia la propria conduttività. Il valore rilevato varia da 0 KOhm alla sua massima resistenza di 10,50 o 100 KOhm in base al tipo di fotoresistenza.

Le fotoresistenze sono caratterizzate da una curva di sensibilità dipendente dal colore al quale sono maggiormente sensibili e dai valori di resistenza al buio e alla luce: valore di buio (pochi MΩ) e valore di luce (alcuni KΩ).



Caratteristiche principali:

- Tensione operativa (max): 150V DC
- Wattaggio (max): 90mW
- Resistenza (luminosità - 10Lux): 5~10KΩ
- Resistenza (oscurità): 0.5 MΩ
- Tempo di risposta (fase di ascesa): 20ms
- Tempo di risposta (fase di discesa): 30ms
- Diametro: 5mm
- Peso: 0,2g



LED

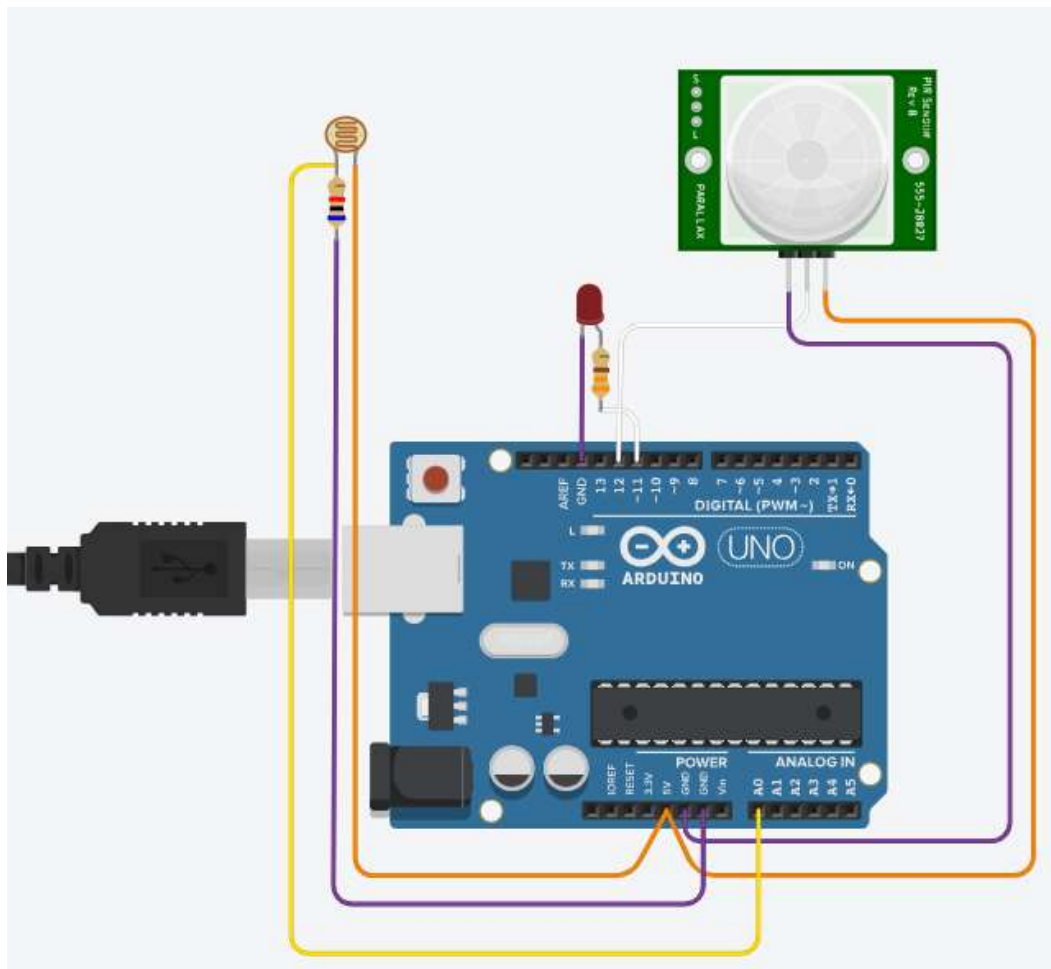
Abbreviazione di Light Emitting Diode, il LED è un attuatore semplice che riporta un segnale digitale in un mondo analogico. Possiede due gambe: quella lunga è l'anodo, positivo, mentre quella corta il catodo, negativo.

Il colore scelto per questo progetto è stato il rosso e di conseguenza la caduta di tensione del LED è di 1,8 V. Collegando direttamente il diodo senza resistenze vorrebbe indurre una corrente $i = (5 - 1.8) / \text{Resist del filo}$, troppo alta. Il valore minimo per far accendere un LED è di 20 mA, di conseguenza per limitare l'ampereaggio è necessario usare una resistenza almeno di:

$$R = (5 - 1.8) / 0.02 = 160 \text{ ohm}$$

Per il prototipo verrà usata una resistenza da 330 ohm. Lo stesso ragionamento è stato effettuato per la fotoresistenza, a cui verrà applicata una resistenza di 6,4 Kohm.

Schema di massima di collegamento ad Arduino



Schema realizzato tramite Tinkercad.com

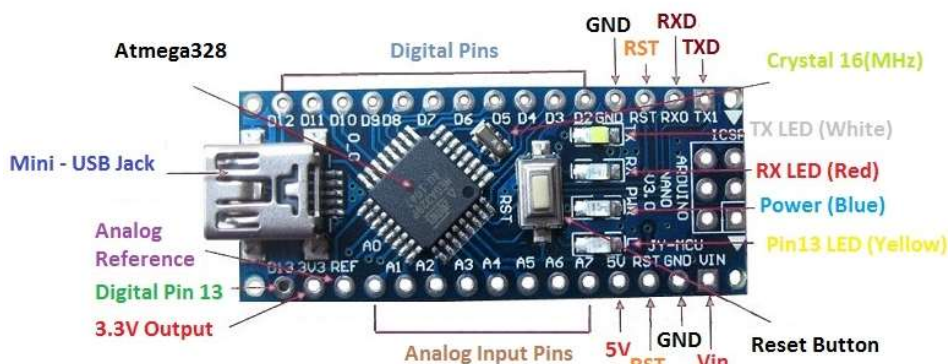
Microcontrollori e unità di processing/bridge

Arduino

Come microcontrollore a cui sono connessi i vari sensori ed attuatori è stato utilizzato Arduino, una piattaforma hardware low-cost programmabile, con cui è possibile creare circuiti di ogni tipo per molte applicazioni. Si basa su un Microcontrollore della ATMEL, l'ATMega168/328: **Arduino Nano** per esempio, impiegato nel prototipo, monta un ATMega328.

La scheda Arduino è in grado di interagire con l'ambiente in cui si trova ricevendo informazioni da una grande varietà di sensori, ma allo stesso tempo può comandare luci, LED, motori ed altri attuatori. I progetti basati su Arduino, che si basano su linguaggio di programmazione derivato dal C/C++, possono essere indipendenti oppure interfacciati con altri software come Processing.

Ogni programma che si scrive su Arduino necessita di una funzione `setup()`, eseguita una sola volta, ed un `loop()`, eseguito infinite volte finché non si toglie l'alimentazione al dispositivo.



I pin principali ed utilizzati nel progetto sono i seguenti:

- Pin di uscita corrente a 5V
- GND: massa o terra
- Analog Input Pins: possono percepire molto precisamente una corrente DC tra 0 e 5V, restituendo un valore da 0 a 1023.
- Digital Pins: pin digitali programmabili per essere input o output. Percepiscono se è presente o no corrente restituendo LOW se non c'è corrente e HIGH se c'è corrente.

Processing

Per la realizzazione del prototipo, il microcontrollore è stato collegato tramite porta USB ad un laptop, per utilizzare quest'ultimo come unità di bridge tra Arduino e il lato server. Come supporto per questa implementazione, viene utilizzato Processing, uno sketchbook software flessibile che può interagire con Arduino e permette di sviluppare diverse applicazioni nel contesto delle arti visive. In questo caso però non viene usato per realizzare un applicativo visivo, ma soltanto per l'invio di richieste HTTP via internet.

Basandosi sul linguaggio Java ma usando una sintassi semplificata, l'ambiente di sviluppo permette di realizzare degli sketch: generalmente ogni sketch contiene almeno una classe principale con una funzione setup, che verrà invocata una volta sola all'avvio dell'applicativo, e una draw, eseguita ad ogni frame.

Varie librerie e strumenti permettono di estendere le capacità di Processing; in questo progetto per esempio vengono utilizzate le librerie *http.requests* e *processing.serial* per rispettivamente inviare richieste HTTP e comunicare con Arduino tramite porta Seriale.

Telegram

Telegram, un servizio di messaggistica istantanea basato su cloud, è stato utilizzato come lato server del progetto, in quanto risulta necessario inviare messaggi ai famigliari riguardo l'eventuale malessere del parente. Una delle principali differenze tra Telegram ed altri servizi, quali Whatsapp, risiede nell'utilizzo del cloud: è possibile infatti visualizzare su dispositivi diversi la stessa chat contemporaneamente. Un'altra funzione che viene utilizzata nel progetto è rappresentata dalla possibilità di creare gruppi in cui parteciperanno più persone e potranno comunicare tra di loro.

A giugno 2015 sono stati introdotti i bot, applicazioni di terze parti che agiscono all'interno di Telegram: fungono da interfaccia per eseguire codice e non richiedono un numero di telefono per funzionare. Il servizio di messaggistica offre infatti un servizio di API che permette di connettere i bot al server di Telegram. Alla creazione di un bot viene sempre generato un authorization token caratterizzante, utilizzato per avere la garanzia di essere in possesso delle autorizzazioni necessarie per la modifica.

L'utente può interfacciarsi con i bot inviando loro messaggi, comandi o usando direttamente richieste HTTPS. Nel progetto viene per l'appunto utilizzata questa ultima funzione, in quanto tramite richiesta HTTPS sarà possibile ordinare al bot di inviare un messaggio.

Protocolli di comunicazione

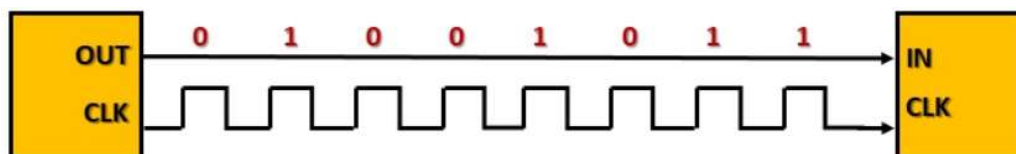
Trasmissione seriale

Arduino comunica con Processing tramite trasmissione seriale, una modalità di comunicazione nella quale vengono inviati su un unico cavo un bit alla volta in modo sequenziale. Il destinatario riceverà i bit nello stesso ordine in cui sono stati inviati dal mittente.

Rispetto alla comunicazione parallela, richiede un minor costo dato il minor numero di cavi utilizzati, i microcontrollori infatti generalmente non hanno molte porte I/O, è più tollerante agli errori di trasmissione ed è preferibile per collegamenti a lunga distanza. Al contrario, può essere preferibile utilizzare una comunicazione parallela se si ha necessità di velocità di trasmissione.

La comunicazione può avvenire tramite due modalità:

- Modalità sincrona: tutti i bit vengono trasmessi in sequenza su di un unico filo sincronizzandoli con un segnale periodico, clock, trasmesso su di un altro filo. Questa configurazione permette un trasferimento dati più veloce ma richiede anche l'utilizzo di un canale aggiuntivo.



- Modalità asincrona: mittente e destinatario usano due clock diversi e chi riceve i bit non può avere la garanzia sulla temporizzazione con cui arrivano i dati. È necessario sincronizzarsi tramite il pacchetto stesso inviato: il mittente invia inizialmente un bit di partenza, seguito dall'informazione vera e propria, successivamente un opzionale bit di parità e un bit finale di stop. Questo metodo di sincronizzazione rende la trasmissione dei dati molto meno efficiente rispetto a quella sincrona.

Protocollo HTTP

L'invio di richieste dal bridge Processing verso il bot Telegram avviene tramite protocollo HTTP, più precisamente HTTPS. È un protocollo a livello applicativo, ultimo livello del modello ISO/OSI per le reti di calcolatori: la sua funzione è quella di interfacciare e fornire servizi per i processi delle applicazioni.

La connessione avviene tramite server HTTP, il quale rimane in ascolto sulla porta 80. Si basa su un'architettura client-server: il client (utente tramite browser) effettua una chiamata al server (sito web), il quale restituisce una risposta. Al termine dello scambio, la connessione viene chiusa per lasciare spazio ad altre future connessioni: per questo motivo il sistema viene definito *stateless*, ovvero non mantiene in memoria gli stati precedenti ma solo quelli attuali.

La richiesta viene effettuata tramite URL, Uniform Resource Locator, che permette di localizzare le risorse sul server:

[http://\[username\[:password\]@\]host\[:porta\]</percorso>\[?querystring\]\[#fragment\]](http://[username[:password]@]host[:porta]</percorso>[?querystring][#fragment])

- http: protocollo utilizzato a livello applicativo. Oltre ad HTTP esistono anche altri protocolli, come HTTPS e FTP;
- Username e password: credenziali di autenticazione per l'accesso alla risorsa, opzionali. Risulta rischioso inviarle tramite url in quanto verrebbero inviate al server in chiaro;
- Host: indirizzo IP o nome di dominio del server;
- Porta: la porta standard è 80 e si può omettere, ma se si desidera è possibile modificarla;
- percorso: percorso per raggiungere la risorsa all'interno del server, ultimato con il nome della risorsa desiderata;
- querystring: stringa di caratteri che consente di passare al server vari parametri;
- fragment: indica una parte all'interno della risorsa.

Più nello specifico, Telegram supporta i metodi GET e POST di HTTP; i parametri possono essere inviati tramite URL query string, formato x-www-form-urlencoded, formato json o form-data. Tutte le richieste ai bot di telegram devono utilizzare il servizio HTTPS e necessita la seguente forma:

https://api.telegram.org/bot<token>/METHOD_NAME

HTTPS è un protocollo di comunicazione derivante da HTTP ma che garantisce riservatezza, tramite cifratura con crittografia asimmetrica dei dati. È un protocollo stateful, in quanto sia client che server devono tenere traccia delle chiavi private per tutta la sessione, per eventuali altre iterazioni. HTTPS garantisce inoltre autenticazione del server e integrità dei dati scambiati, caratteristiche fondamentali per una comunicazione sicura via internet. La porta di default che utilizza è 443.

Prototipo

Funzionamento

Il microcontrollore utilizzato per il progetto è Arduino NANO, molto simile alla sua versione base ma di dimensioni ridotte: il motivo della scelta risiede nel voler installare successivamente il prototipo in un contenitore di piccole dimensioni nei pressi della stanza da bagno.

Nel caso specifico, all'interno della scatola da parete contenente l'interruttore, è presente un led che mostra all'esterno l'accensione della lampada da parete interna al bagno (Figura 1); come primo sensore è stata scelta quindi una **fotoresistenza** che verrà inserita nella scatola e reagirà all'accensione della luce nella stanza. Una volta rilevata l'accensione, un **sensore a infrarossi** permetterà di individuare movimenti all'interno del bagno.

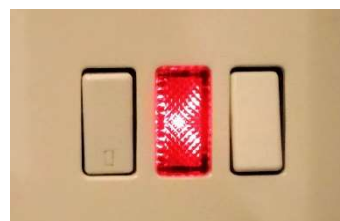


Figura 1

Gli stati principali del sistema, realizzati su scheda Arduino, sono cinque. Identificando con LIGHT la fotoresistenza e PIR il sensore a infrarossi, gli stati possibili sono:

- *LIGHT OFF + PIR OFF*: bagno libero / liberato;
- *LIGHT OFF + PIR ON*: considerato sempre bagno libero / liberato;
- *LIGHT ON + PIR OFF*: luce accesa;
- *LIGHT ON + PIR ON*: bagno occupato, rilevato movimento;
- *LIGHT ON + PIR OFF con iniziale movimento*: nessun movimento della persona, potenziale pericolo.

Dopo un iniziale movimento rilevato, se susseguono 10 secondi di inattività una luce di emergenza al di fuori del bagno si illuminerà (nel prototipo è stato utilizzato un **led rosso**). Dopo un totale di 20 secondi, ed ogni 10 secondi aggiuntivi, verrà inviato un messaggio di PERICOLO per avvertire i famigliari.

L'allarme si disattiverà quando la luce del bagno verrà spenta e non quando si rileverà di nuovo movimento: questa scelta è stata presa in quanto potrebbe risultare utile continuare a ricevere messaggi anche durante il periodo di soccorso, per eventuale sicurezza. Una possibile futura implementazione potrebbe consistere nell'inserimento o di un pulsante all'interno del bagno o un qualunque altro sensore utile nel caso specifico di applicazione.

Bot Telegram

Per la creazione del bot è stato utilizzato **@BotFather**, un bot dedicato alla creazione dei nuovi bot. Al termine della creazione viene fornito il token per l'accesso alle HTTP API.



Il bot generato, @SafeNotifyMe_bot, viene utilizzato per inviare messaggi all'interno di un gruppo composto dai famigliari o dalle persone che offriranno aiuto alla persona non autosufficiente. Una volta inserito il bot nel gruppo, per identificare l'id di quest'ultimo, fondamentale per inviare il messaggio, è necessario inviare una richiesta HTTPS su browser con il seguente url:

<https://api.telegram.org/bot<token>/getUpdates>

Come risposta si ottiene una lista di informazioni relative al bot, tra le quali si trovano quelle riguardo al gruppo creato a cui fa parte il bot e i famigliari.

```
{"id":-361136352,"title":"FamilySafeNotify","type":"group","all_members_are_administrators":true}, {"date":1551191718,"group_chat_created":true}]}}
```

Avendo il token del bot, l'id del gruppo, il testo del messaggio da inviare e conoscendo il metodo utilizzato dall'api di Telegram, sendMessage, è possibile inviare tramite Processing una richiesta GET al bot per ordinargli di trasmettere il messaggio al gruppo desiderato.

https://api.telegram.org/bot<token>/sendMessage?chat_id=<groupid>&text=<message-text>

Invio del messaggio

Il messaggio viene inviato al gruppo in seguito ad una serie di passaggi:

a) Arduino

Dopo aver inizializzato i pin dei sensori come input o output e calibrato il sensore a infrarossi (devono passare intorno a 15 secondi dall'accensione di Arduino) nella funzione *setup()*, il sistema inizia il *loop* per identificare prima se la luce è accesa e se lo è identificare l'eventuale movimento all'interno del bagno.

```
if(<luce spenta>)          /*LIGHT OFF + PIR OFF e LIGHT OFF + PIR ON */
else                       /* luce accesa */
    if(<rilevato movimento>) /* LIGHT ON + PIR ON */
    else                   /* LIGHT ON + PIR OFF */
        if(<passati 10 secondi>) /* danger!! Invia messaggio */
                                /* continua a inviare */
                                /*terminato pericolo*/
        if(<luce spenta>)
```

b) Arduino → Processing

Per inviare il messaggio a Processing, è necessaria una comunicazione seriale, configurabile su Arduino e Processing rispettivamente con i seguenti comandi:

- lato Arduino `Serial.begin(9600);`
- lato Processing `String portName = Serial.list()[0];`
`Serial myPort = new Serial(this, portName, 9600);`

dove 9600 rappresenta il rateo bit per secondo utilizzato.

Le informazioni necessarie a Processing per comunicare con il bot di Telegram consistono nel momento in cui si ha qualcuno che sta utilizzando il bagno, non vi sono più movimenti successivi e quando la luce viene spenta dopo un momento di inattività, rappresentante il termine del soccorso dato. In generale, una stringa inviata via seriale ha la seguente forma:

```
Serial.print("PERICOLO CONTINUO.");
```

c) Processing

La funzione di Processing consiste nell'interpretare il pacchetto ricevuto da Arduino ed inviare una richiesta HTTPS al bot di Telegram. Nel codice sottostante viene mostrato come il programma, nella funzione `draw()` che viene eseguita ad ogni frame, processa il pacchetto ricevuto: a seconda di una determinata sotto-stringa presente in esso, genera un messaggio che andrà in pasto alla funzione `send_msg()`, adibita alla richiesta HTTPS.

```
String val="";
if ( myPort.available() > 0){
  val = myPort.readString();
  print(val);

  //controllo dei pacchetti in arrivo da ARDUINO
  String msg;
  String msg1;

  if( val.contains("PRONTO") ){
    msg1 = "SISTEMA+PRONTO.";
    send_msg(msg1);
  }

  msg = "CONTINUO";
  if( val.contains(msg) ){
    msg1 = "****DANGER:+Inattivita'+da+";
    sec += 10;
    msg1 = msg1 + sec + "+secondi.";
    send_msg(msg1);
  }
}
```

d) **Processing → Telegram**

`send_msg()` è una funzione con l'obiettivo di generare un corretto url, unendo il messaggio desiderato, e inviare una richiesta GET tramite il comando della libreria `http.requests`.

```
void send_msg(String message){  
    String url = "https://api.telegram.org/bot761484947:AAHUXod....  
    url = url + message;  
    println("\n");  
    System.out.println(url);    //stampo a video il link + messaggio  
    GetRequest get = new GetRequest(url);  
    get.send();  
}
```

e) **Telegram**

Una volta che la richiesta è stata inviata, il messaggio viene ricevuto dagli utenti all'interno del gruppo, i quali potranno interagire fra di loro per andare a supportare la persona bisognosa.

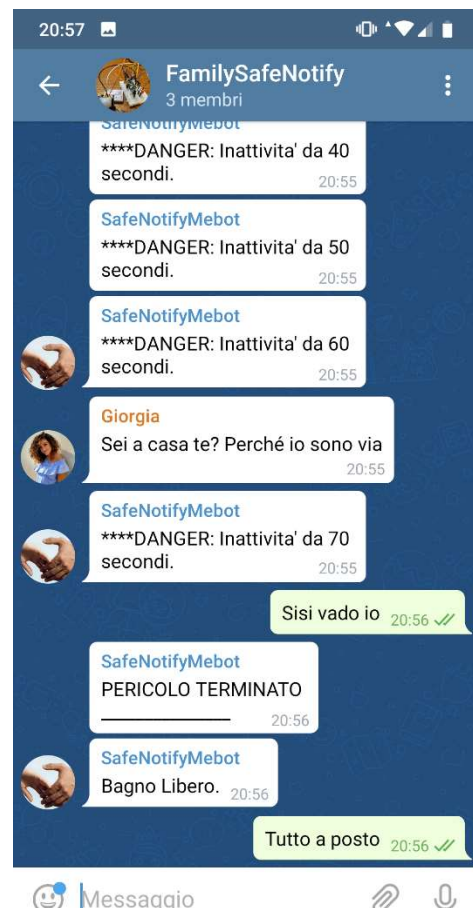
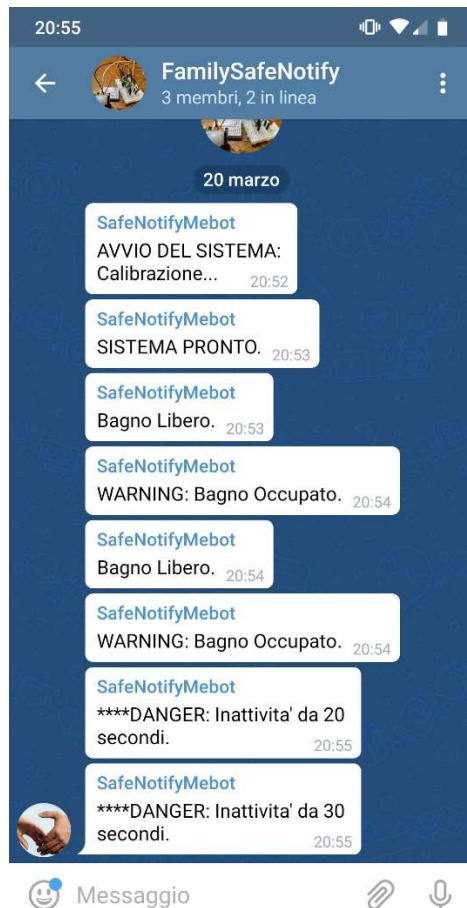


Immagine del prototipo

