

A dark blue vertical bar runs along the left edge of the page. A blue arrow points to the right from this bar, containing the date.

21/01/2021

# Rapport Final

Emergency call button

Enseignants : Aomar Ousmani

Massinissa Hamidi

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Louni Ghezlane Nadine  
Alexandre Fred Karim  
IOT

INTRODUCTION.....	1
Etat de l'art .....	3
Ancienne perspective + Difficultés rencontrées.....	<b>Erreur ! Signet non défini.</b>
Nouvelle perspective .....	<b>Erreur ! Signet non défini.</b>
Composants utilisés .....	<b>Erreur ! Signet non défini.</b>
Produit final .....	<b>Erreur ! Signet non défini.</b>
Conclusion.....	8

## INTRODUCTION

Nous vivons dans une époque où les objets connectés se comptent par milliers mais peu d'entre eux s'avèrent être réellement utiles aux personnes âgées ou handicapées. Comment pourrait une personne muette, clouée au lit, contacter une auxiliaire de vie présente dans une autre pièce ? Comment une personne âgée pourrait facilement contacter une personne dans un laps de temps extrêmement court ? Aujourd'hui, certains concepts d'objets connectés permettent de résoudre et au minimum, de contourner ces problèmes. En effet, si nous ne pouvons soigner ces personnes, essayons au moins de les aider.

Comme l'indique le titre, notre objet connecté serait un « Emergency Call Button » « *Bouton d'appel d'urgence* ».

Il arrive souvent qu'une personne malade ou âgée se retrouve dans une situation où sa mobilité est restreinte. Imaginons maintenant que le malade veuille faire appel à une auxiliaire où n'importe qui susceptible de l'aider, ne serait-il pas plus simple de pouvoir effectuer cet appel de détresse en appuyant sur un petit bouton à proximité ?

En effet, ce petit bouton pourrait intervenir dans énormément de situations où le malade ne peut crier "à l'aide" !

Les boutons d'urgence peuvent être utilisés un peu partout : à la maison, dans la rue, à l'hôpital. De ce fait, on peut facilement s'en procurer surtout en ligne dans les grandes boutiques comme Amazon, Cdiscount etc. où les frais de livraison sont assez souvent pris en compte. Nous verrons aussi par la suite certains modèles de 3 gammes différentes. Enfin, nous étudierons le marché afin de faire ressortir les éléments importants.

## CE QUE L'ON TROUVE SUR LE MARCHE



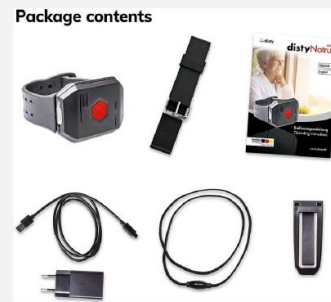
29,99€

- Recepteur et emetteur
- Utilisation simple pour l'envoi d'un signal d'urgence
- Limite par la distance et les obstacles



97,38€

- GPS inclus
- Permet de contacter jusqu'à 3 personnes par appel/sms
- Detection de mouvement



126,51€

- Appelle 5 personnes successivement jusqu'à reception de l'appel
- Application incluse
- Appels entrant et sortant

Le marché des boutons d'appels d'urgence (BAU pour simplifier) est très vaste. On a pu trouver sur le net beaucoup de modèles et d'objets visant à aider les personnes en détresse, certains sont plus performants, offrent plusieurs options et sont assez coûteux, d'autres sont moins performants mais accessibles à tous (niveau budget). Alors que les modèles bas de gamme sont peu coûteux et très simplistes, plus ce dernier se verra ajouter des options comme le GPS ou un système de communication plus élaboré, il devient par conséquent plus cher.

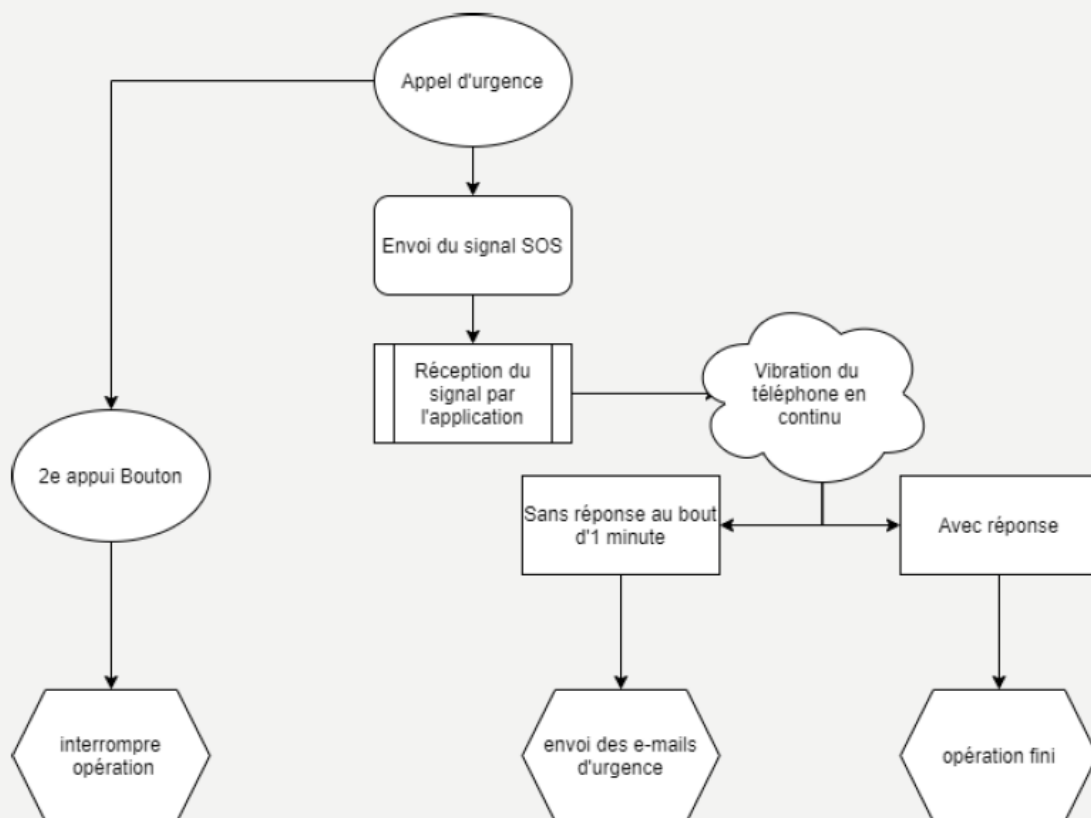
Il est tout de même intéressant de noter que les modèles qui se vendent le plus sont justement les modèles bas de gamme. On a ensuite compris qu'ils sont justement beaucoup plus vendus car bien que les modèles plus chers soient bien plus complets : GPS. Capteur de chute / température, ils sont peu vendus car justement, les malades sont souvent déjà surveillés par des appareils destinés pour cela (hôpital ou personnel). Les modèles basiques sont pour cela bien plus vendus.

## Ancienne perspective

Au tout début, notre objectif était de créer un schéma où l'on utilisait 2 ESPs maître-esclave (pour la connexion WiFi) et une application mobile qui permettait de gérer l'ensemble. Nous nous sommes ensuite vite rendu compte dans un premier temps qu'un 2<sup>e</sup> ESP était inutile.

Notre objectif était donc celui de garder un ESP qui se connecterait par lui-même au réseau et de plus l'on aurait une application mobile qui prendrait la communication entre le malade et le possible aide-soignant.

### • Diagramme d'utilisation:



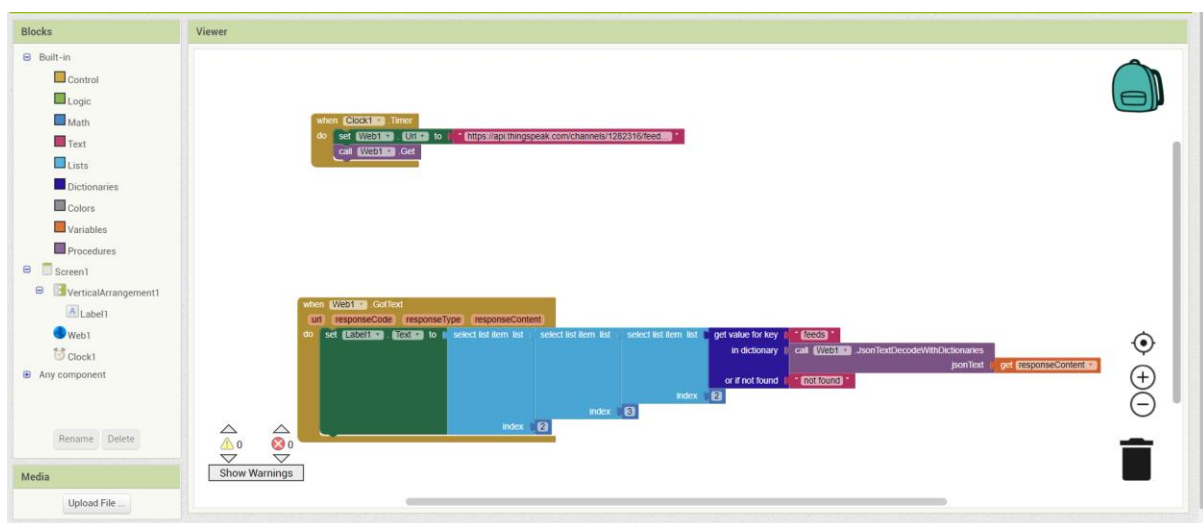
## Difficultés Rencontrées

Nous nous sommes donc mis début décembre sur la conception d'une application mobile. Nous avons au début commencé par suivre des tutoriels sur ANDROID STUDIO, cependant nous n'avancions qu'à très petit pas car nous partions de 0 en conception d'application android.

De plus le combo covid+partiel ne nous aidait pas du tout dans le sens où notre temps était limité.

Après environ 2 semaines de frustration et de non-productivité, nous nous sommes orientés vers une plateforme de conception d'application d'android appelée APP INVENTOR.

APP INVENTOR était un peu plus simple dans le sens où il fallait simplement utiliser des instructions logiques préprogrammées. Cependant nous avons réalisés bien trop tard que le souci est qu'une fois l'application créée, il est très compliqué de le débbugger. Notre application était au début opérationnel puis sont venus les bugs, bref c'était un enfer. Voici quelques captures depuis la plateforme d'APP INVENTOR :



Face à ce mur, nous avons décidé de regarder ailleurs, et nous sommes tombés sur IFTTT !

## NOUVELLE PERSPECTIVE

IFTTT (IF This Than That) est un service web permettant à ses utilisateurs de créer des chaînes d'instruction simples appelées applets.

Nous n'avons donc pas besoin de « créer » une application. Il suffit simplement à notre ESP8266 de contacter le serveur IFTTT puis grâce aux applets utilisées nous pourrions interagir avec le téléphone. Nous utilisons justement 3 applets :

- Le premier permet d'enlever le mode « silencieux » du téléphone si ce dernier était activé :



### **Set ringtone volume**

This Action will set the ringtone volume of your Android device.

#### **Volume**

100%



- Le second nous permet d'envoyer une notification personnalisée :



### **Send a notification from the IFTTT app**

This action will send a notification to your devices from the IFTTT app.

#### **Message**

J'ai besoin de votre aide de toute urgence.

**OccurredAt**

**Add ingredient**

- La 3<sup>e</sup> et plus importante nous permet d'appeler et de diffuser un message vocal personnalisé :



### **Call my device**

This action will call your device with the IFTTT app installed and speak a message provided by your Applet.

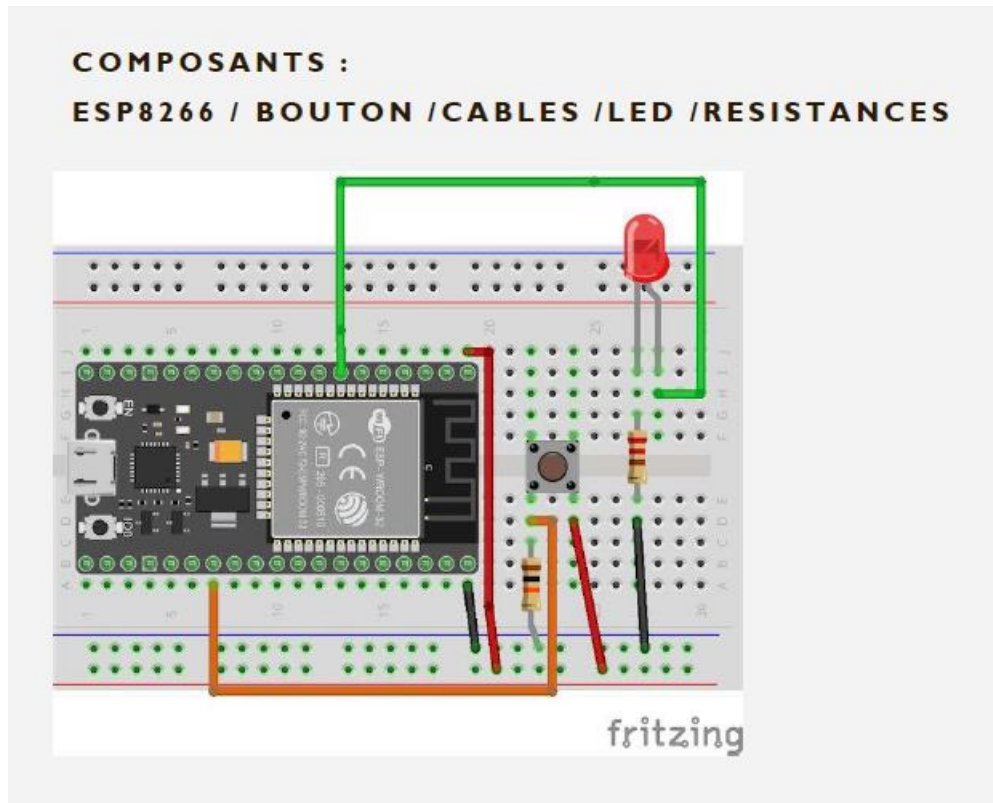
#### **Voice message**

J'ai besoin de votre aide de toute urgence.  
Je répète, j'ai besoin de votre aide de toute urgence. **OccurredAt**

**Add ingredient**










## HARDWARE

De là, nous pouvons garder cette configuration là pour le prototype :



### Composants :

Materials

Image	Name	Part Number	Price	Count	Link
	ESP8266	1	7.99	1	
	led	2	7,99	1	
	resistance	3	Included in starter arduino pack	1	
	boutton	4	(Included in starter arduino pack)	1	
	câbles male femelle	5	(included in starter arduino pack)	Severals	



## Programmation de l'ESP8266

```
#include <ESP8266WiFi.h>

const char* ssid = "";
const char* password = "";
// Unique IFTTT URL resource
const char* resource = "/trigger/emergency_call_button/with/key/gXsg2b0dWW69zZjsRUd6M6IfNLqlhUGvtqDQw02TJDn"; //Important
int keyIndex = 0;
int LED = D3;
int BUTTON = D2;
const char* server = "maker.ifttt.com";

void setup(){
  Serial.begin(9600); //set rate of serial monitor
  initWifi();
  pinMode(LED,OUTPUT); //set LED pin as output
  pinMode(BUTTON,INPUT_PULLUP); //set button pin as input
}
```

---

D'abord, nous commençons par les initialisations. La variable ressource contient la clé qui permettra de communiquer avec le serveur IFTTT. Nous utilisons la bibliothèque ESP8266 pour le wifi.

```
void loop() {
  if(digitalRead(BUTTON) == LOW) {
    digitalWrite(LED,HIGH);
    delay (1000);
    digitalWrite(LED,LOW);
    delay (1000);
    digitalWrite(LED,HIGH);
    delay (1000);
    makeIFTTRequest();

  }else{

    digitalWrite(LED,LOW);

  }
}
```

---

Si on appuie sur le bouton, la led va flasher puis soumettre la requête. Sinon la led est OFF.

```

void initWifi() {
    Serial.print("Connecting to: ");
    Serial.print(ssid);
    WiFi.begin(ssid, password);

    int timeout = 10 * 4; // 10 seconds
    while(WiFi.status() != WL_CONNECTED && (timeout-- > 0)) {
        delay(250);
        Serial.print(".");
    }
    Serial.println("");

    if(WiFi.status() != WL_CONNECTED) {
        Serial.println("Failed to connect, going back to sleep");
    }

    Serial.print("WiFi connected in: ");
    Serial.print(millis());
    Serial.print(", IP address: ");
    Serial.println(WiFi.localIP());
}

```

Cette fonction nous permet d'initialiser le WiFi.

```

void makeIFTTRequest() {
    Serial.print("Connecting to ");
    Serial.print(server);

    WiFiClient client;
    int retries = 5;
    while(!client.connect(server, 80) && (retries-- > 0)) {
        Serial.print(".");
    }
    Serial.println();
    if(!client.connected()) {
        Serial.println("Failed to connect, going back to sleep");
    }

    Serial.print("Request resource: ");
    Serial.println(resource);
    client.print(String("GET ") + resource +
        " HTTP/1.1\r\n" +
        "Host: " + server + "\r\n" +
        "Connection: close\r\n\r\n");

    int timeout = 5 * 10; // 5 seconds
    while(!client.available() && (timeout-- > 0)){
        delay(100);
    }
    if(!client.available()) {
        Serial.println("No response, going back to sleep");
    }
    while(client.available()){
        Serial.write(client.read());
    }

    Serial.println("\nclosing connection");
    client.stop();
}

```

*Cette dernière fonction nous permet de faire la quête au serveur IFTTT*

Enregistrement terminé

Et finalement nous avons le prototype final :



## CONCLUSION

Bien que simple sur le papier ce projet d'IOT nous a pris du temps et énormément. Nous avons perdu beaucoup de temps sur la conception d'une application que nous n'avons pas finalement pu créer. Cependant, nous restons satisfaits dans le sens où les fonctionnalités marchent et nous n'avons pas trop changé l'essence du projet.

Nous profitons pour remercier nos professeurs qui nous ont permis de réaliser une première immersion dans le monde des objets connectés. L'expérience fut difficile mais enrichissante. Nous remercions aussi nos camarades qui nous ont aidés et conseillés et parfois à des heures improbables.

Il y a beaucoup de points sur lesquels l'ECB pourrait être amélioré et il est loin d'être parfait. Nous avons pensé par exemple à implémenter de la géolocalisation cependant, ayant déjà déboursé plus de 70 euros pour le projet (achat du kit, de plusieurs ESP -il y a eu des accidents-), nous nous sommes entendus pour nous arrêter là.

Ainsi se termine donc notre compte rendu final pour l'EMERGENCY CALL BUTTON !