

ALLART Julien – VERNIER Paul – VIEUBLED Tanguy  
IMA 3

## Rapport de Projet

Robe et pantalon pour lutter  
Contre le harcèlement



Alexandre BOE - Xavier REDON - Thomas VANTROYS  
2018 – 2019



# Sommaire

Introduction	page 3
I- Cahier des charges	page 4
Besoin fonctionnel Contraintes technique	
II- Conception & Réalisation	page 5
Application Capteur	
Bilan & Perspectives	page 7
Annexe	page 8

# Introduction

Dans le cadre de notre formation d'ingénieur à Polytech Lille en Informatique Microélectronique et Automatique (IMA3), nous nous sommes penchés sur le développement d'un projet qui doit permettre de lutter contre le harcèlement.

En effet, Le harcèlement est un phénomène très largement répandu dans notre société, particulièrement dans les soirées ou encore dans les transports en commun. Certaines personnes prétendent qu'il n'est pas possible d'éradiquer ce phénomène. Or nous pensons le contraire. Pour contrer ce phénomène, nous allons concevoir un système visualisant en temps réel les points de contact sur l'utilisateur et identifiant les cas de harcèlement le cas échéant.

Ce projet se déroule sur une période de deux ans. En fin de première année, nous présenterons un prototype fonctionnel. Ce prototype sera une première version de notre système, il concentrera les fonctionnalités décidées dans notre cahier des charges et l'idée de son fonctionnement.

Dans ce rapport, vous trouverez les différentes étapes d'élaboration de notre projet à savoir les décisions et les compromis que nous avons dû faire. Dans un premier temps, nous présenterons le contexte propre du projet. Ensuite, nous dévoilerons le cahier des charges régissant ce projet. Et dans une dernière partie, nous développerons les différentes parties de réalisations.

# I- Cahier des charges

Après avoir analysé notre projet en regardant des systèmes existants ou offrant un résultat similaire, nous avons pu identifier nos objectifs et dégager plusieurs axes de réalisation. A l'issue du semestre 6, nous proposerons un prototype répondant aux spécifications techniques et aux besoins que nous avons fixé dans notre cahier des charges.

## ➤ Besoin Fonctionnel

Le système à concevoir doit permettre de visualiser les points de contact sur le corps d'une personne. Les informations doivent être visionnées en temps réel, nous devons donc traiter les données et les rendre disponible immédiatement.

Une interface à développer pourra être utile pour communiquer avec le système et afficher les résultats du traitement des données. Aussi, elle doit informer son utilisateur s'il est en situation de harcèlement. Cette interface sera une application pour smartphone Android

## ➤ Contraintes Techniques

Notre système comporte tout d'abord des spécificités. En effet, il sera composé d'un grand nombre de capteur pour la détection des points de contact. Il devra donc être léger, simple à porter sans gêner son utilisateur et surtout être discret.

Ensuite les contraintes de réalisation sont l'utilisation de l'environnement Arduino pour exploitation des informations délivrées par les capteurs de force. Pour le développement de l'application Android, on utilisera l'utilitaire MIT App Inventor. Un module de communication Bluetooth sera employé pour la communication entre le système et l'application.

# II- Conception & Réalisation

Grâce à MIT App Inventor 2, nous avons pu développer une première application Android pour notre système. Le but premier de l'application est de visualiser les points de contact sur le corps de l'utilisateur.

Nous avons commencé par développer une application (Code voir annexe 2) via le moniteur série Arduino pour tester les différents capteurs de force commandés.

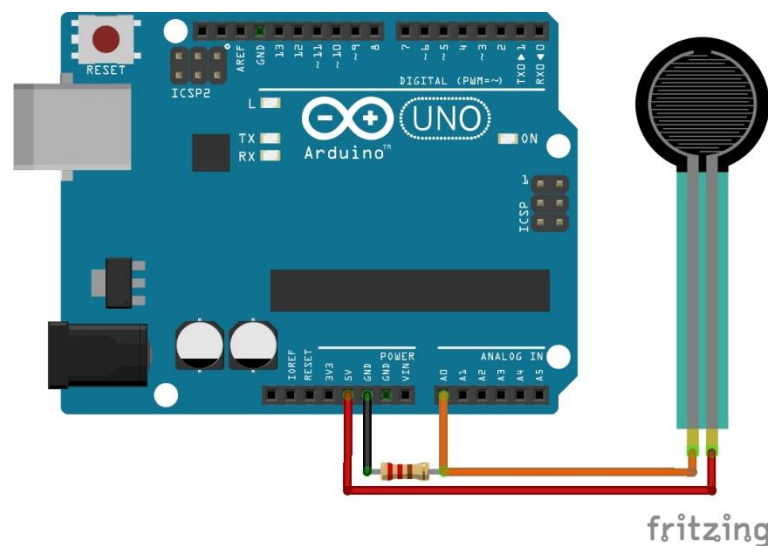


Figure 1 - Schéma Arduino avec un seul capteur

Grâce à cela, on a pu déterminer différentes valeurs en fonction de la force appliquée sur le capteur. Afin d'éliminer les « bruits » pouvant parasiter notre système, on ne prend pas en compte les valeurs en dessous de 250. Ainsi, on peut alors observer la force du point de contact sur la personne.

Type de contact	Caresse	Main posée	Tenir qqch
Force en Newton	0	1-3	Sup 4
Valeur	250 - 400	500 - 700	Sup 800

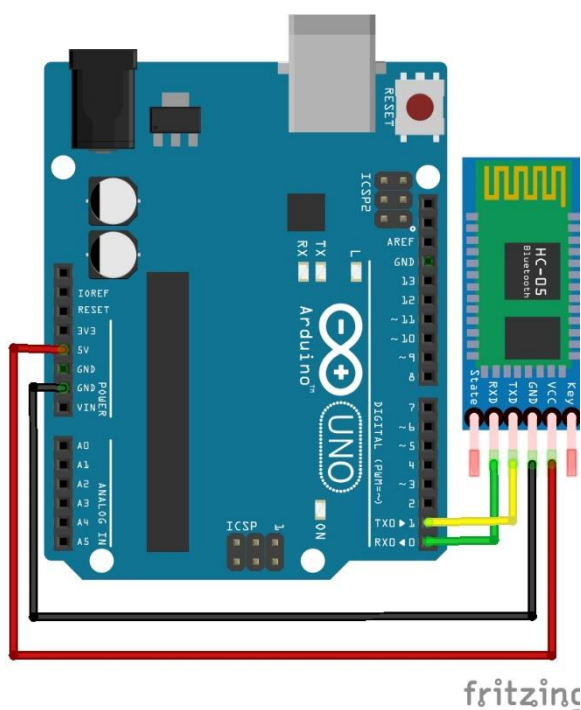


Figure 2 - Schéma Arduino avec module Bluetooth

Après ces tests, nous avons ajouté un module de communication Bluetooth pour relier notre système à notre application développée avec l'outil MIT App Inventor 2.

Nous avons commencé par établir la connexion entre l'application et le téléphone Android. Après, nous avons cherché à transmettre les valeurs de capteur via le module Bluetooth. L'un des problèmes que nous avons rencontrés immédiatement, ne permettant pas de visualiser deux touches distinctes à deux endroits différents. (Code voir annexe 3).

Nous avons donc repensé le programme Arduino avec une solution permettant de pallier à ce problème majeur. En effet, l'intérêt de ce projet est justement de visualiser les points de contact qu'ils soient isolés ou multiples. Le nombre de capteur à notre disposition était limité. Pour autant, ce ne fût pas un problème pour la réalisation de notre prototype. Le programme et l'application développés par nos soins, sont programmés pour fonctionner avec 3 capteurs qui se positionnent sur un avant-bras (Schéma voir annexe 1).

On a implémenté une fonction qui permet d'utiliser jusqu'à  $n$  capteurs. On utilise une boucle `for` pour lire les valeurs des capteurs. Lorsqu'un capteur est activé, on incrémente la variable `sum` qui nous permet de savoir quel capteur est activé. Cette variable est transmise par Bluetooth et l'application traite le résultat de la variable `sum`.

Au lieu d'envoyer précisément chaque valeur de force des capteurs, on envoie une trame qui contient quel capteur est activé mais aussi des informations sur la durée des points de contact. La trame est composée de plusieurs chiffres (1 ou 0). Sa longueur dépend du nombre de capteur. Par exemple, avec 3 capteurs, on peut recevoir la trame suivante : 011. On sait alors que le capteur 1 n'est pas touché mais que le 2 et le 3 le sont. La variable `tempContact` nous permet de savoir si le capteur est activé pendant 4 secondes consécutives ou pendant 10 secondes non-consécutives sur 30 secondes. On interrompt l'envoi des données pour indiquer qu'on est en présence de harcèlement.

Nous avons pensé à différents scénarios qui suivant différents critères peuvent nous renseigner sur la présence ou non de harcèlement. Ces scénarios devaient nous aider à coder notre projet. Malheureusement, MIT App Inventor nous a limité pour la création et le développement de l'application.

# Bilan & Perspectives

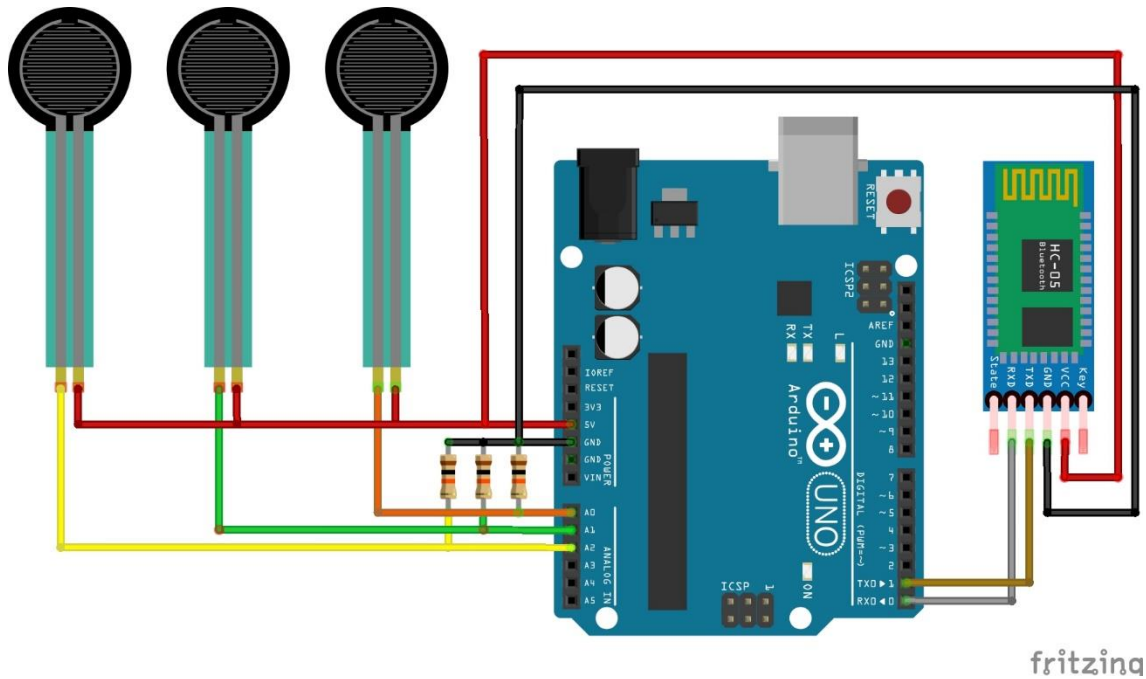
Ce projet nous a réellement intéressés. Il nous a permis de développer nos connaissances et nos compétences dans la gestion et réalisation de projet. Nous avons pu développer un prototype permettant de répondre aux objectifs fixés par notre cahier des charges. Celui-ci peut détecter des points de contact sur le corps d'un utilisateur. Etant limité par le nombre de capteur, nous avons conçu le prototype pour être disposé sur un avant-bras. Cependant notre code permet d'intégrer autant de capteur que d'entrée analogique sur la carte Arduino.

Néanmoins l'outil MIT App Inventor, nous a vraiment limités pour le développement de notre application. En effet, MIT ne nous permettait pas de traiter nos données efficacement.

Les perspectives de ce projet sont nombreuses. Tout d'abord, nous développerons notre propre application pour utiliser plus efficacement le système « Don't touch me ». Ensuite, on étendra notre système pour un vêtement complet (Robe). Le système comportera une batterie pour le rendre mobile en gardant en tête qu'il ne gêne pas et qu'il se ne repère pas.

# Annexe

## Annexe 1 – Montage Arduino





## Annexe 2 – Code test capteur

```
int const A = 4;      // nombre de capteurs
int fsrPin[A];
int fsrReading[A];
int fsrVoltage[A];
long fsrForce[A];

int calculForce(int fsrVoltage) {
    unsigned long fsrResistance, fsrConductance;
    int fsrForce;
    fsrResistance = 5000 - fsrVoltage;
    fsrResistance *= 10000;
    fsrResistance /= fsrVoltage;
    fsrConductance = 1000000;
    fsrConductance /= fsrResistance;
    if (fsrConductance <= 1000) {
        fsrForce = fsrConductance / 80;
    } else {
        fsrForce = fsrConductance - 1000;
        fsrForce /= 30;
    }
    return fsrForce;
}

void setup() {
    Serial.begin(9600);
    for(int i = 0; i < A; i++) {
        fsrPin[i] = i;
    }
}

void loop() {
    for(int i = 0; i < A; i++) {
        fsrReading[i] = analogRead(fsrPin[i]);
        Serial.print("Analog reading ");
        Serial.print(i);
        Serial.print(" : ");
        Serial.println(fsrReading[i]);
        fsrVoltage[i] = map(fsrReading[i], 0, 1023, 0, 5000);
        if (fsrVoltage[i] == 0) {
            Serial.println("No pressure");
        } else {
            fsrForce[i] = calculForce(fsrVoltage[i]);
            Serial.print("Force in Newtons : ");
            Serial.println(fsrForce[i]);
        }
    }
    Serial.println("-----");
    delay(1000);
}
```

## Annexe 3 – Code Projet

```
int const A = 3;
int fsrPin[A];
int fsrReading[A];
double sum = 0;
int tempContact = 0;
int message;
int timer = 0;
int timeTouche = 0;

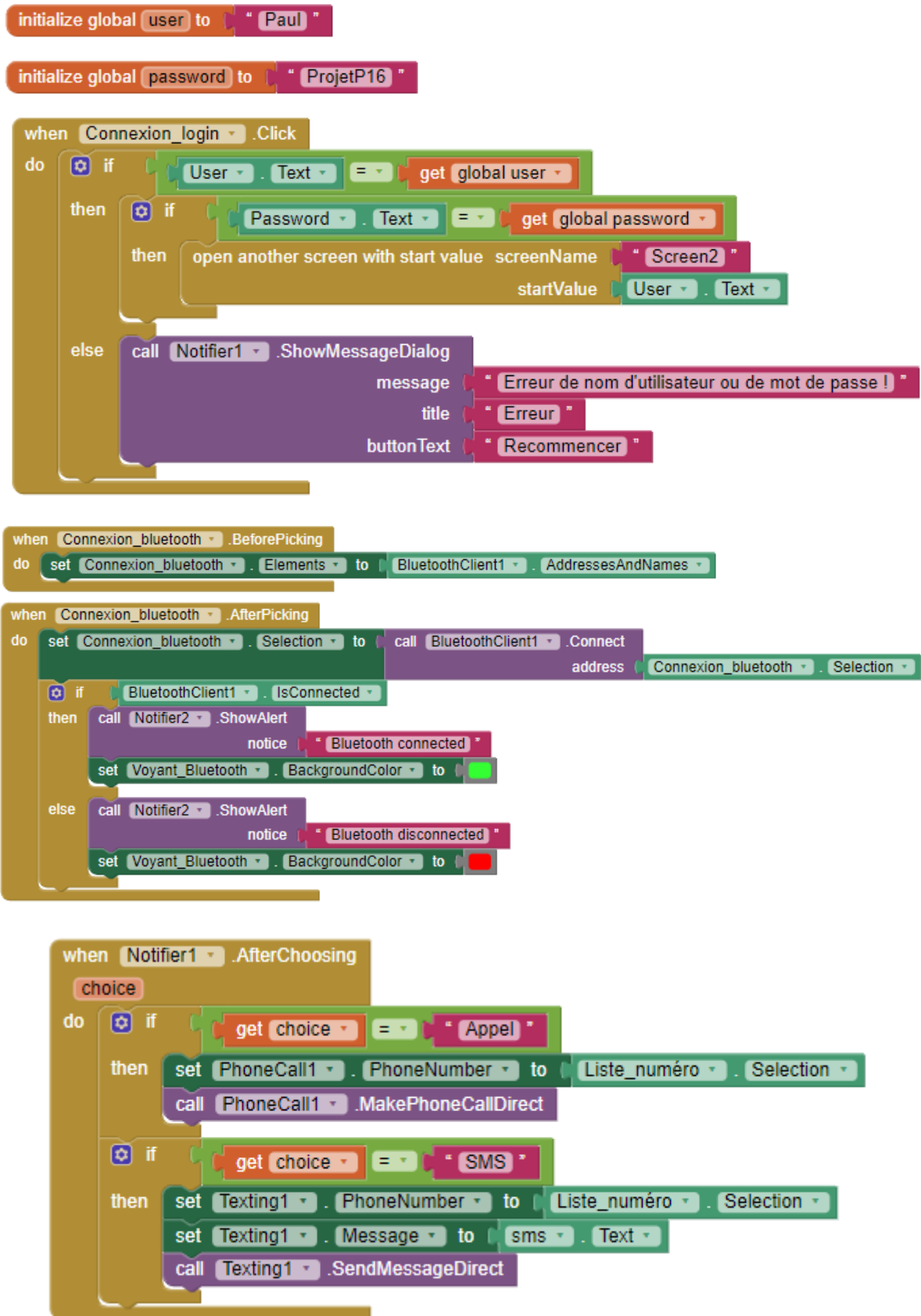
void setup() {
  Serial.begin(9600);
  for(int i = 0; i < A; i++) {
    fsrPin[i] = i;
  }
}

void loop() {
  for(int i = 0; i < A; i++) {
    fsrReading[i] = analogRead(fsrPin[i]);
    if(fsrReading[i] > 350) {
      sum = sum + pow(10, i);
    }
  }
  if(tempContact < 20) {
    Serial.println(sum);
  }
  if(sum > 0) {
    tempContact++;
    timeTouche++;
  }
  else tempContact = 0;
  if(tempContact >= 20) {
    while(message != 12) {
      Serial.println(2);
      delay(200);
      if(Serial.available() > 0) {
        message = Serial.read();
        if(message != -1) {
          Serial.println(message);
        }
      }
    }
  }
  tempContact = 0;
}
```

```
tempContact = 0;
message = 0;
}
if(timeTouche > 50) {
  Serial.println(12);
}
if(timer > 150) {
  timeTouche = 0;
  timer = 0;
}
sum = 0;
delay(200);
timer ++;
}
```

---

## Annexe 4 – Application Android



```

initialize global numero to create empty list

when Liste_numero .BeforePicking
do set Liste_numero . Elements to get global numero

when Liste_numero .AfterPicking
do set aff_num . Text to Liste_numero . Selection

when Bt_ajout_num .Click
do add items to list list get global numero
    item make a list Liste_numero . Selection
    numero . Text
set numero . Text to " "
set Liste_numero . Elements to get global numero

```

```

when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set Label2 . Text to call BluetoothClient1 .ReceiveText
    numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
    if is number? Label2 . Text
    then
        if Label2 . Text == 111
        then set C1 . BackgroundColor to red
            set C2 . BackgroundColor to red
            set C3 . BackgroundColor to red
            set global cptouche to get global cptouche + 1
        if Label2 . Text == 110
        then set C1 . BackgroundColor to red
            set C2 . BackgroundColor to red
            set C3 . BackgroundColor to white
            set global cptouche to get global cptouche + 1
        if Label2 . Text == 2
        then call BluetoothClient1 .Send1ByteNumber
            number 12
            call Notifier1 .ShowChooseDialog
            message " "
            title "Êtes vous en danger?"
            button1Text "Appel"
            button2Text "SMS"
            cancelable true

```