

Rapport projet

Bracelet de chute

2020-2021



25 Mai

L3-EEA

Créé par : Antony Corentin
Grossard Mathieu
Lange Theo Phil
Schultz Quentin

Professeur responsable :
Hermann Gilles

 **FST**
Faculté des Sciences et Techniques
UNIVERSITÉ HAUTE-ALSACE

Les remerciements :

Nous sommes Corentin Antony, Theo Phil Lange, Mathieu Grossard et Quentin Schultz. Nous sommes ensemble pour travailler sur notre projet qui est un bracelet de chute.

Nous tenons à remercier toutes les personnes actives sur les différents forums Arduino qui nous ont permis de peaufiner notre code pour le projet ainsi que les différentes fiches techniques de nos modules qui possèdent pour la plupart un code de démarrage.

Ainsi que le corps enseignant, plus particulièrement Mr. Hermann qui nous a imprimé notre boîte et nous a conseillé sur certains branchements de nos composants.

Sommaire

INTRODUCTION	3
1. CONTEXTE ET CAHIER DES CHARGES.....	4
2. ETUDES	5
2.1. PUBLIC VISE :	6
2.2. COMPOSANTS UTILISE :	7
3. REALISATIONS :.....	17
3.1. PLANNING :	18
3.2. BUT DU FONCTIONNEMENT :	19
3.3. FORME FINAL :	21
4. BILAN ET PERSPECTIVES :.....	24
4.2. CONCLUSION :	25
4.3. LEXIQUE :.....	26
4.4. WEBOGRAPHIE :	27
5. ANNEXE	28
5.1. CODE DU PROJET.....	28
5.2. ECRANS DIFFERENTS	36
5.3. SENSIBILITE DU CAPTEUR AVEC DIFFERENTES ETUDES	37

Introduction :

Nous avons dû pendant cette période restreinte de la Covid-19 réfléchir sur une thématique de détecteur de chute pour certaines personnes qui sont à notre choix. Pour cela en début n'ayant pas le matériel disponible, nous avons réfléchi sur ce projet à distance en utilisant Discord pour pouvoir communiquer entre nous. Mais aussi choisir le but du projet, les personnes visées ainsi que de son fonctionnement recherché.

Nous sommes partis sur un bracelet de chute pour un certain groupe d'âge qui sera décrit plus loin. Ce bracelet de chute a été voulu pour ressembler à une montre pour que la personne puisse la porter sur son poignet. Ce rapport va donc contenir notre démarche de réflexion sur ce projet.

Nous allons alors parler de notre étude sur le projet, comme le public visé, puis nous verrons les étapes de réalisation du projet. Et enfin, le bilan de cette étude.

1. Contexte et cahier des charges :

Nous avons pour but du projet de détecter une chute qui concerne les personnes âgées. C'est-à-dire que lorsque la personne tombe, il se peut qu'elle ne puisse plus se lever. Pour cela notre bracelet est prévu pour communiquer avec une centrale qui prévient les autorités compétentes ou les proches. Si un jour le produit pourrait sortir, les clients pourraient bénéficier du bracelet pour X prix, et par exemple d'un abonnement de **10€** par mois.

Pour cela nous avons fait plusieurs choix ou plutôt nous imposer des contraintes. Nos choix ou contraintes sont :

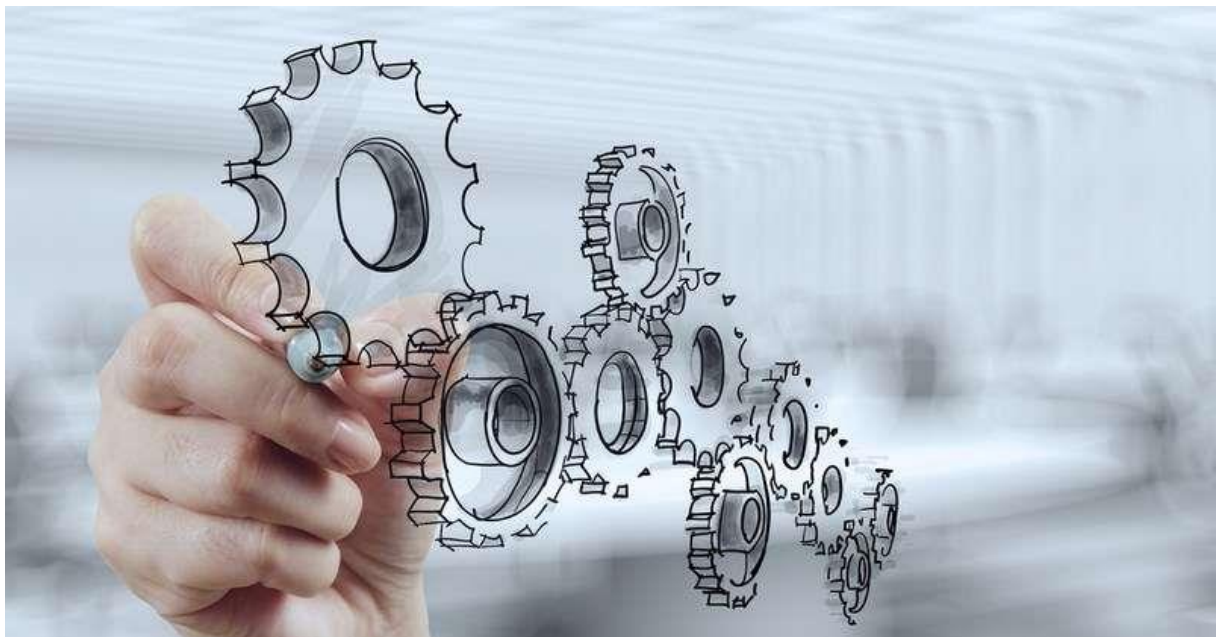
- Les personnes âgées sont des personnes actives alors pour cela si elles se déplacent hors de leur maison nous sommes censés pouvoir les localiser ;
- Elles n'ont peut-être pas d'appareils de communications, c'est-à-dire dans notre cas un téléphone portable pour prévenir la centrale ;
- Si le bracelet envoie un message mais que la personne se fait aider ou se relève toute seule elle doit avoir la possibilité d'informer que tout va bien ;
- Cependant, si c'est une erreur elle doit être aussi capable de le dire ;
- Mais si elle a besoin d'aide car elle ne se sent pas bien, elle peut avoir la possibilité de prévenir ;
- Et pour finir, un moyen capable de prévenir les personnes aux alentours à l'aide d'un son ;

Il s'agit donc de toutes les contraintes que nous nous sommes dans un sens imposées lors de l'élaboration du projet.

2. Etudes :

Nous allons dans cette partie expliquer notre procédé de réflexion pour notre étude. Que cela concerne notre démarche de réflexion du public visé, le fonctionnement voulu de notre projet et donc nous aurons une grande partie avec les différents composants utilisés.

Cette partie comportera le but d'utilisation du composant, ce que nous avons souhaité faire, ce qui a pu être fait et ce que l'on n'a pas pu réaliser finalement.



2.1. Public visé :

Nous avons à un moment donné dû nous poser pour discuter du type de public visé. Notre public serait-il un public âgé ou sportif. Après une étude du marché nous nous sommes rendu compte qu'il serait plus important de viser un public âgé, notamment à cause de la hausse de cette tranche d'âge et aussi de personnes dépendantes.

Par contre les sportifs sont moins dépendants de ce genre de produit même s'il y en a une demande celle-ci est moins importante et surtout au vu des situations de confinements répétitifs la cible de l'étude est mieux axée pour une tranche d'âge nécessaire pour leur protection.

Puisque nous avons choisi de viser les groupes de personnes âgées nous devons donc décider de leur besoin, comment l'utiliser et d'autres contraintes. Ainsi nous avons d'abord discuté de la forme que cela devrait prendre et nous nous sommes concentrés pour un bracelet type montre. Etant donné qu'elle est plus utilisée qu'une ceinture ou encore un collier. Nous avons dû décider de la manière de communiquer et celle-ci a pris plus de temps, car on a dû se concerter pour se mettre d'accord entre le choix d'une personne à mobilité réduite ou une personne dynamique mais risquant de faire des chutes. Au final nous avons choisi une personne pouvant se déplacer, cela signifiait que l'on n'allait pas faire une communication entre deux appareils dont l'un par un moyen ou un autre communique aux urgences. Donc nous avons décidé d'utiliser une communication par GSM¹ c'est-à-dire une communication numérique à l'aide d'une carte SIM¹.

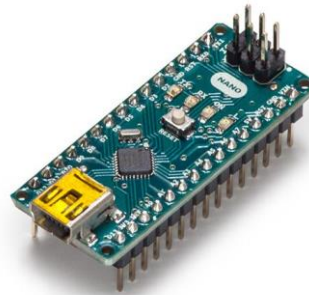
Comme nous avons décidé de viser les personnes âgées en mouvement nous devons être capable de la localiser. C'est pourquoi nous avons intégré un GPS¹ qui utilise les satellites pour repérer la personne, ainsi ces coordonnées seront envoyées par SMS. Nous avons rajouté un buzzer pour informer les personnes aux alentours afin qu'ils puissent se rendre compte que la personne est tombée.

Finalement nous avons aussi rajouté deux boutons, un bouton pour annuler l'envoi de SMS. L'autre bouton au cas où la personne a besoin d'aide même si elle n'est pas tombée.

¹ P.26

2.2. Composants utilisé :

Carte Arduino NANO :

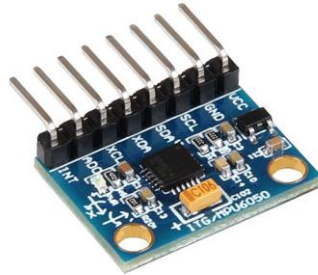


Nous avons décidé d'utiliser comme microcontrôleur la carte d'Arduino NANO par son prix faible et sa petite taille mieux adaptée pour notre étude. Malgré son coût plus faible il ne faut pas croire à une faible performance. Au contraire l'Arduino NANO possède 14 pins digitaux à travers lesquels elle fournit 40 mA² maximum. En outre, elle intègre 2 pins analogiques de plus que la UNO. Cela fait donc 8 pins analogiques pour la Nano. En ce qui concerne la capacité, elle est dotée d'une mémoire Flash² de 32Kb² et 2Kb² de mémoire dynamique. En utilisant le pin 30 on peut connecter une alimentation externe comprise entre 6 et 20 Volt² que l'on abordera plus tard.

Le choix de la NANO est aussi en occurrence avec l'environnement de travail qui est Arduino IDE. Celui-ci nous permet de créer nos programmes et de les transférer à la carte à l'aide de la prise USB. Mais cela est aussi en corrélation avec nos études puisque nous avons depuis le début de cette année utilisé cette interface pour la programmation robotique ou de projet.

² P.26

Accéléromètre et gyroscope 3 axes :



Nous utilisons l'accéléromètre pour détecter en partie la chute ou tout du moins une accélération rapide du mouvement. Nous pouvons sans aucun doute dire que c'est notre capteur le plus important et le plus essentiel à la réalisation de notre projet car c'est lui qui nous détecte principalement si nous avons une chute ou non. L'accéléromètre fonctionne avec un bus I2C³.

Nous avons souhaité l'utiliser aussi pour détecter en partie des mouvements ou des gestes pouvant mettre la personne en péril. Cela signifie le fait de monter sur une chaise, un escabeau principalement. Mais certains problèmes se sont posés. Ces problèmes sont pour la plupart une lecture des données brutes retransmise par l'accéléromètre et les adapter à nos besoins.

Nous avons uniquement réussi à détecter une accélération et un choc et non de prévenir un risque pouvant mettre en danger la personne. Nous supposons que cela vient de différents facteurs comme le fait que nous le portons sur le poignet ainsi si une personne lève son bras notre bracelet détecterait aussi un risque. Nous avons aussi le problème de monter des escaliers puisque le principe serait le même que monter une échelle. C'est-à-dire une avancée en x et y par exemple.

Au final nous avons réussi à transformer les valeurs brutes en valeur à étudier et avec quelques tests de simulations nous avons pu déclarer une sensibilité de chute à 23. {Cf. Sensibilité du capteur avec différentes études}

³ P.26

Capteur de vibrations :

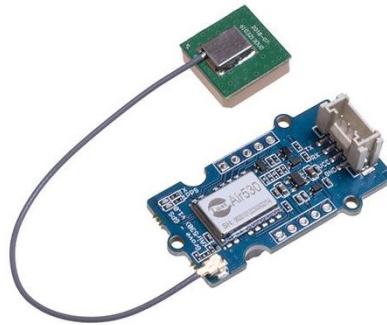


En complément de l'accéléromètre nous avons décidé d'utiliser un capteur de vibrations. Celui-ci est censé nous faire une confirmation de la chute. L'idée que nous avons eue, c'est que l'accéléromètre peut nous permettre de détecter une accélération mais pas sûr pour une chute alors on complète le montage avec un deuxième capteur, le capteur de vibrations qui va détecter les vibrations dues au choc. Par contre en pratique certains problèmes se sont posés.

Les problèmes que l'on a rencontrés sont en fait dû au fonctionnement lui-même du capteur de vibrations. Celui-ci fonctionne concrètement par le roulement d'une bille et est censé changer d'état lors de vibrations ou choc assez fort. Le problème c'est qu'il fonctionne très bien mais que dans une position donnée car si on le change la bille va pas se trouver au bon endroit et sera toujours en état haut. C'est-à-dire que le détecteur va toujours détecter des vibrations.

Nous avons donc dû essayer de la placer le mieux possible dans notre montage mais aussi adapter au mieux notre code puisqu'il est en quelque sorte la deuxième sécurité même s'il est possible que le code final ne l'utilise pas. Au final pour notre projet nous avons retiré ce capteur car il n'est pas adapté à ce type d'études puisqu'il est mieux adapté pour un déplacement d'un seul axe.

Module GPS⁴ Grove :



Pour la détection de la personne âgée qui est toujours mobile nous avons donc décidé d'utiliser un module GPS⁴ qui utilise les satellites pour obtenir des coordonnées mais ces coordonnées sont fausses dans le cas où la personne est en intérieur. C'est pourquoi le standard est formé pour pouvoir reconnaître de bonnes coordonnées ou des mauvaises et bien diriger les urgences. Le GPS⁴ fonctionne en communication série avec RX et TX.

Je parle de standard car on envoie un message à l'aide d'un module que l'on verra plus tard. Ce message comprend le message je suis tombé suivi d'une trame complète qui est la trame des coordonnées. Nous n'avons pas réussi à extraire uniquement les coordonnées GPS⁴ c'est pourquoi on s'est rabattu sur la deuxième solution qui est d'envoyer le moins d'informations nécessaires. Pour cela on extrait uniquement la trame la plus importante celle qui contient les coordonnées GPS⁴ et ceci sera inclus dans un message qui sera envoyé au standard qui est formé à lire et utiliser ses coordonnées.

Le standard est adapté à comprendre ces messages et à répondre au différent modèle qui seront selon le client, prévenir uniquement les proches, prévenir juste les urgences ou prévenir les deux mais d'autres cas peuvent se passer que l'on verra plus loin.

⁴ P.26

Buzzer :



Nous avons choisi d'utiliser un buzzer dans le but de prévenir la personne âgée que son bracelet à détecter une chute au cas où ce ne serait pas le cas, juste un mouvement trop brusque par exemple.

Le second but est de servir comme prévention pour les personnes aux alentours. Sachant que nous sommes sur une personne mobile. Elle peut faire ses courses toute seules, sortir se promener ou aller chez des proches. Le buzzer permettrait alors de faire réagir les personnes aux alentours qu'il y a un souci dans la même optique qu'un bracelet d'alarme de sécurité.

Nous n'avons pas rencontré de problème lors de l'utilisation du buzzer puisque son principe est simple. Une tension d'entrée permet de faire sortir un son du buzzer. Ayant un principe si simple la programmation n'a pas été compliqué. Le seul petit problème est lors de la création de la boîte qui est de faire un espace vide pour permettre que le son puisse sortir de la boîte et que l'on peut l'entendre. Cette boîte sera dans une autre partie qui vient plus tard.

Bouton poussoir :



Nous avons décidé d'intégrer à notre étude deux boutons poussoir. Chaque bouton a une utilité différente. Ces deux boutons se trouvent à gauche et à droite de l'écran OLED que l'on verra plus tard. L'un des boutons sert à annuler l'envoi des messages avant qu'il parte dans le cas où elle peut se relever ou si on l'aide. L'autre est utilisé par danger ou mauvaises situations. C'est-à-dire si la personne ne se sent pas bien même si elle n'est pas tombée ou si par exemple elle ne peut plus bouger par faiblesse.

Le fonctionnement des deux boutons est très simple puisqu'il joue sur une interruption. Lorsque l'on appuie sur le bouton un changement d'état arrive. Ce changement pris en compte va diriger l'envoi de message. Dans le cas du bouton pour d'interruptions du message on aura un temps donné pour laisser la possibilité à la personne d'interrompre l'envoi du message dans le cas d'une chute comme le fait d'appuyer sur le SOS. Dans l'autre cas « SOS ! » suivi des coordonnées GPS que le standard saura utiliser.

Nous n'avons pas eu de problèmes lors de l'utilisation de ces deux boutons puisque leur principe est le même qui est une interruption suivie d'un message. Seulement les messages peuvent être différents. L'utilisation des interruptions a en plus été bien traiter peu de temps avant le projet donc leur utilisation n'était pas une découverte.

Module GSM⁵ :



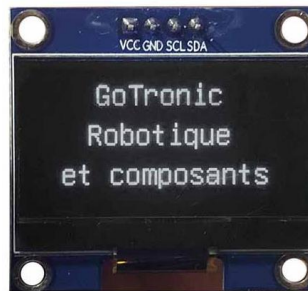
Comme mentionné plus tôt la communication doit pouvoir se faire à n'importe quel moment et si possible n'importe où. C'est pour cela que l'on utilise un module GSM⁵ utilisant une carte SIM⁵. Ce module nous permet de contacter la centrale qui possède les numéros des clients et est formée pour réagir au différent message pouvant arriver. Le GSM⁵ communique en série avec RX et TX comme le GPS⁵.

Puisque nous avons décidé de créer des fonctions pour tester le fonctionnement de chaque composant à part nous avons rencontré un problème sur le GSM⁵. Ce problème était un problème de consommation. Lors de nos recherches et tests, la plupart de nos camarades de classe réussissait à envoyer un SMS alors que les pins de l'Arduino ne peuvent transmettre que 40mA⁵ maximum. Cela nous paraissait étrange sachant que la consommation du module est inférieure à 350mA⁵ et jusqu'à 2A⁵ en pic(appel).

Pour pallier ce problème on utilise un autre composant qui est un régulateur ajustable 1,25 à 30 Vcc. Ce régulateur une fois ajouté au projet et liant notre Arduino au GSM⁵ nous permet de fournir assez d'intensité (jusqu'à 3A) pour pallier la consommation du GSM⁵ qui est supérieure à ce que peut délivrer l'Arduino.

⁵ P.26

Afficheur OLED⁶ :



L'afficheur OLED⁶ nous permet de vérifier le bon fonctionnement du bracelet, puisque l'on a décidé qu'il sera toujours allumé même si malheureusement cela influence sa durée de vie. Nous avons voulu utiliser l'afficheur dans le même principe que l'écran d'une montre mais nous nous sommes rendu compte qu'il nous manquait un composant pour rendre cela possible. Pour pallier cela on n'affiche que le but du bracelet et selon laquelle des situations arrivent l'affichage change. L'OLED⁶ fonctionne avec un bus I2C⁶ comme on peut le voir avec SCL et SDA sur l'image ci-dessus.

Le seul problème rencontré a été d'apprendre à l'utiliser puisque celui-ci fonctionne avec une librairie qui est la U8glib. Nous avons vite trouvé comment faire un programme simple avec sur un blog web. Même s'il est vrai qu'avant de l'utiliser on a cherché à comprendre son fonctionnement. Puisque celui-ci est très différent d'un écran LCD Grove utilisé en TP.

⁶ P.26

Batterie :



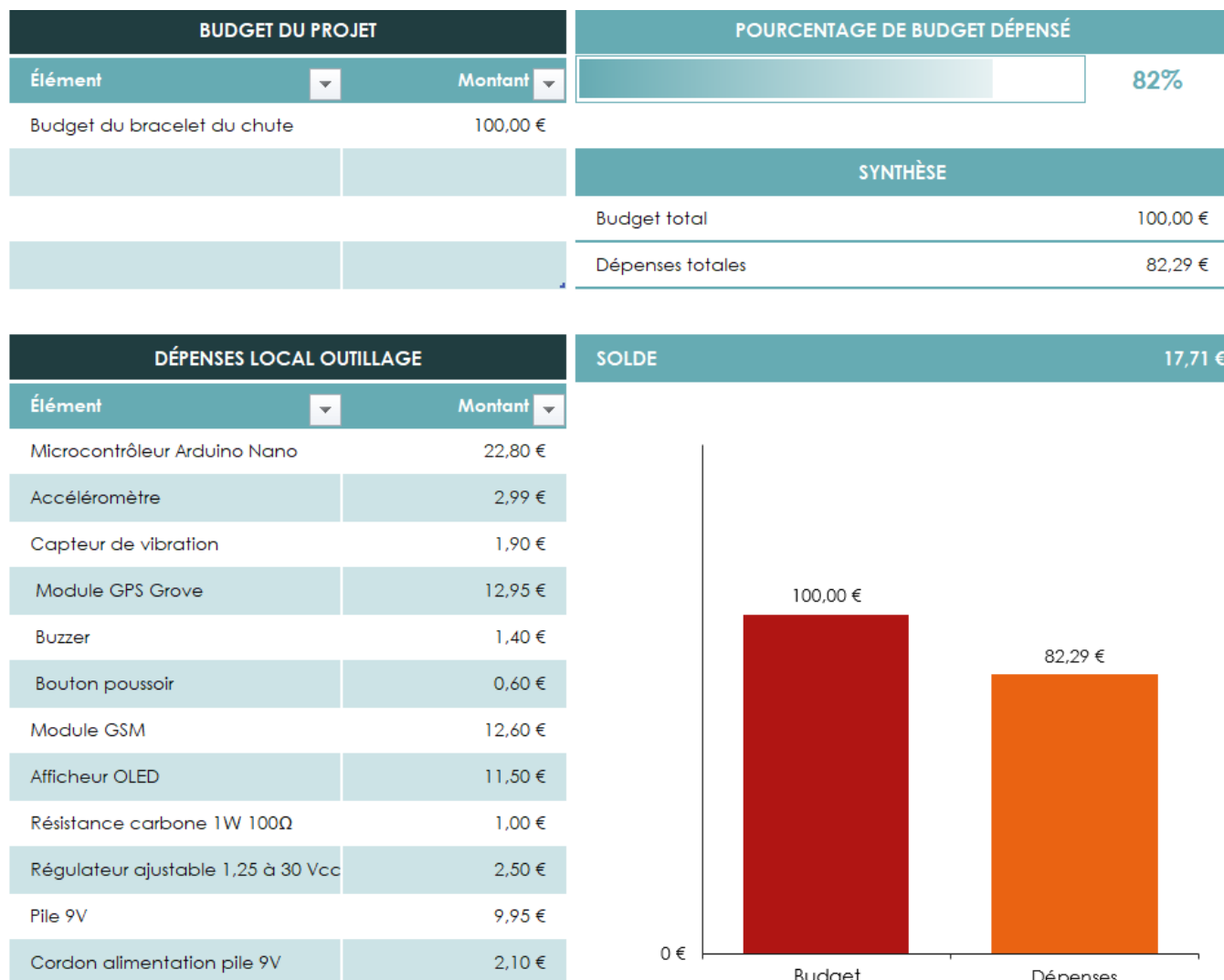
La pile de 9V⁷ a été notre choix pour un prototype de notre bracelet. Notre raison est la tension à fournir en batterie externe qui doit être entre 6 et 20 Volts pour faire fonctionner l'Arduino. En plus de cela il est plus simple de trouver en grande surface une pile 9V et un chargeur pour pouvoir en cas de décharge avoir une possibilité efficace de changer la batterie.

Après étude la pile procell que nous utilisons pour le prototype possède 673 mAh⁷. La pile a pour notre projet une faible autonomie mais selon le calcul de la consommation qui est capacité de la pile / consommation de l'appareil = autonomie en heures. On choisit d'utiliser pour l'autonomie le principe que le GPS⁷ et le GSM⁷ sont en veille la plupart du temps et nous donnons donc la durée d'autonomie la plus grande du bracelet.

Soit consommation de l'appareil = arduino (9.2) + accéléromètre (3.9) + capteur de vibration (1) + OLED (25) + GSM (1.2) + GPS (0.85) = 41.15mAh. Ainsi la durée de vie maximum pouvant être atteinte avec juste quelques composants toujours actifs nous avons autonomie = 673/41.15 = 16 heures environ. Il faut prendre en compte que cette durée est possiblement considérée comme assez longue mais en pratique l'autonomie est plus petite puisque si on considère qu'un message n'est pas envoyé mais que le GPS⁷ est actif, on a par exemple 88.3mAh⁷ soit une autonomie d'environ 8 heures et si un message est envoyé cette autonomie est encore plus petite.

⁷ P.26

Totale :



L'analyse nous montre que le budget initial a été respecté puisque le budget de départ est de 100€ et nous avons au final 82,29€ avec un solde restant de 17,71€. La raison pour cela c'est que nous avons voulu prendre le moins cher possible.

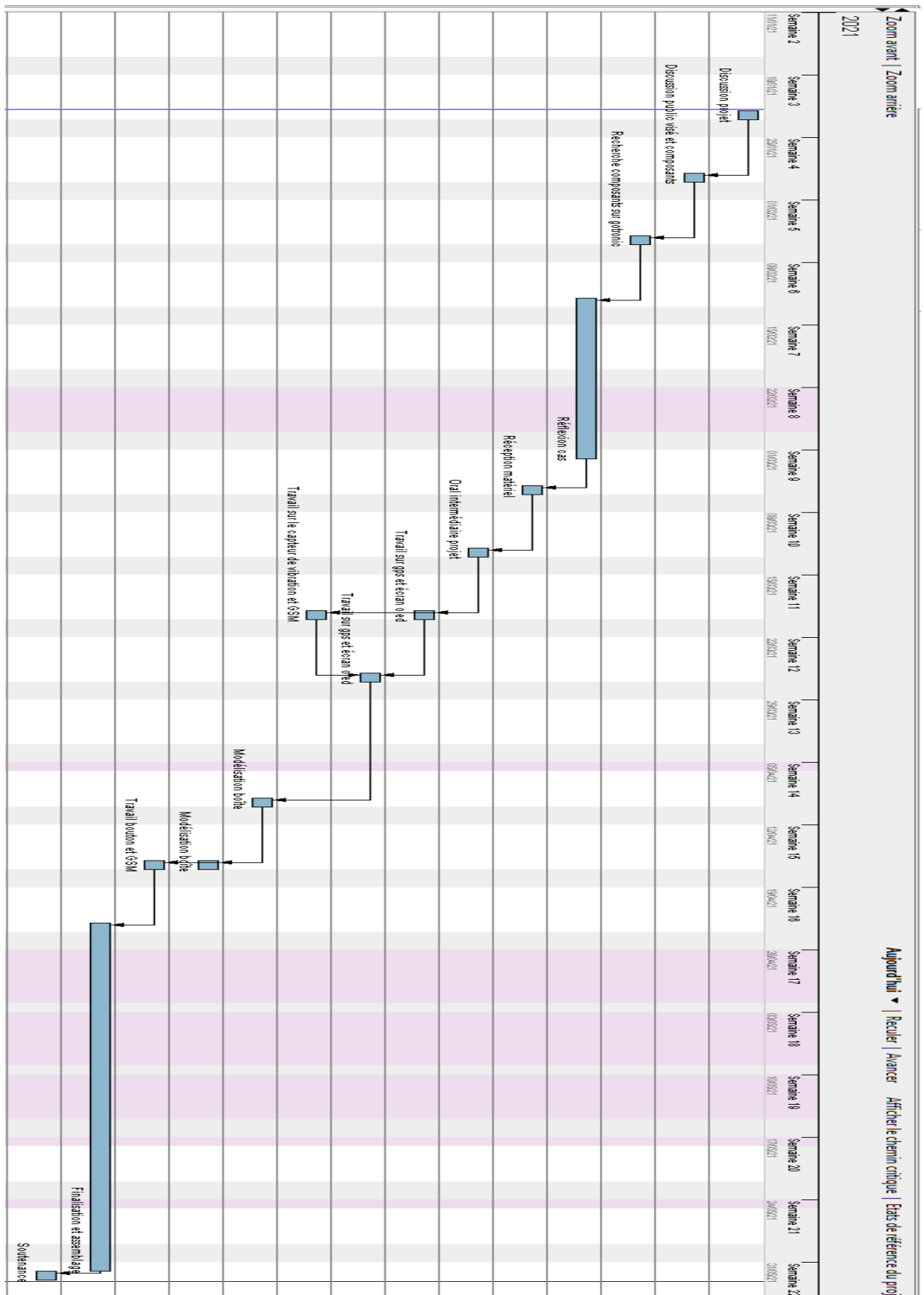
3. Réalisations :

Dans cette partie nous allons expliquer la réalisation du projet. C'est-à-dire nous allons décrire le fonctionnement de notre code dans cette étude. Ses différents buts et fonctions.

Ensuite nous expliquerons la démarche que nous avons eue pour la réalisation du contenant pour ce prototype et pourquoi nous l'avons fait de telles manières.

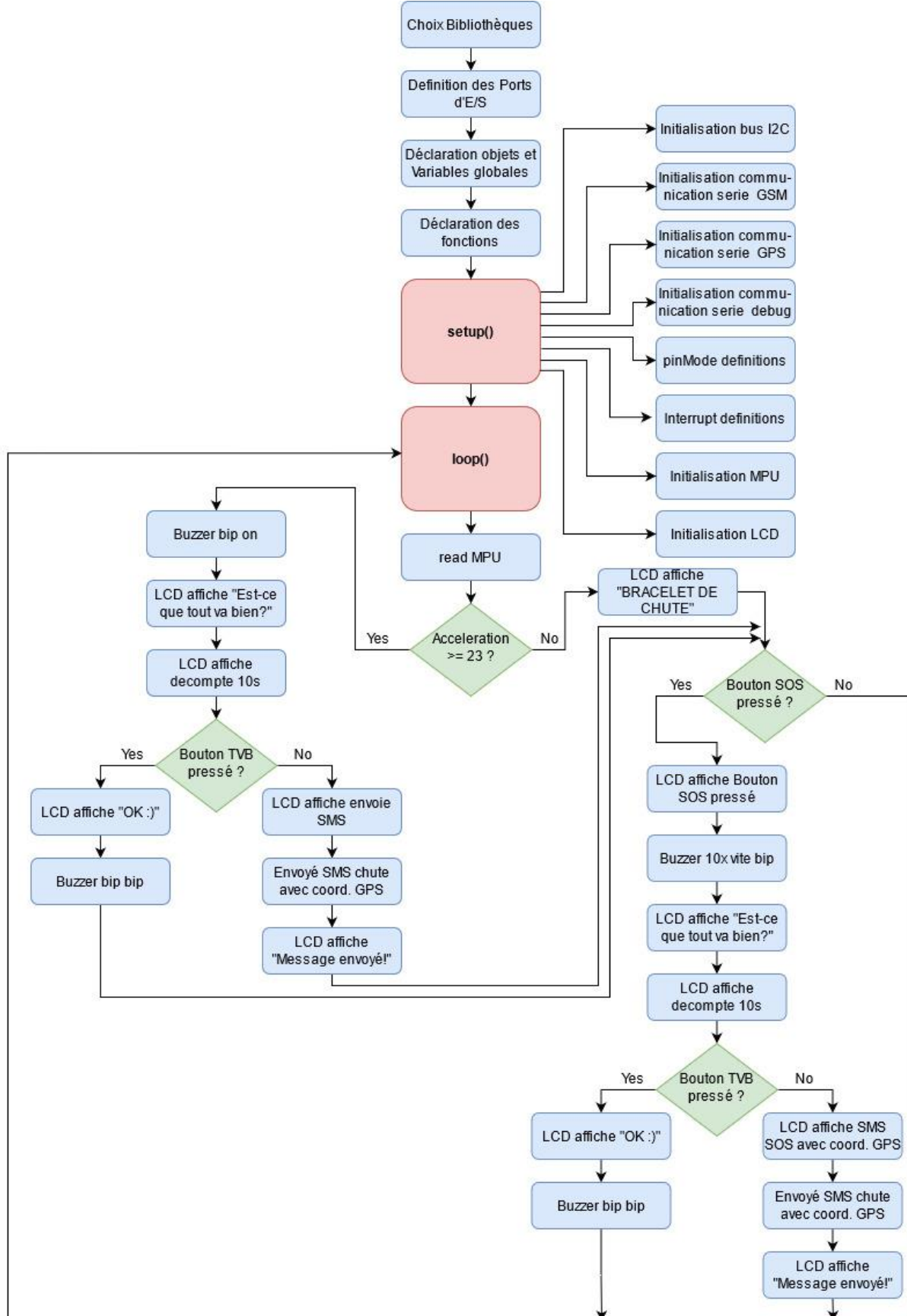


3.1. Planning :



3.2. But du Fonctionnement :

Tout d'abord commençons par un organigramme du code :



Dans ce qui suit, je vais présenter le programme. L'ensemble du code du programme est présenté en détail à l'annexe 5.1. En Annexe 5.2 vous pouvez voir tous les écrans différents de notre bracelet de chute.

Le programme affiche à l'utilisateur "BRACELET DE CHUTE" tant qu'aucun événement ne se produit. Le MPU demande constamment quelle accélération est présente. Si cette valeur est supérieure ou égale à 23, une chute est détectée. Maintenant, le buzzer est activé et l'utilisateur est interrogé par l'intermédiaire de l'écran : "Est-ce que tout va bien ? Un décompte est affiché pendant 10s.

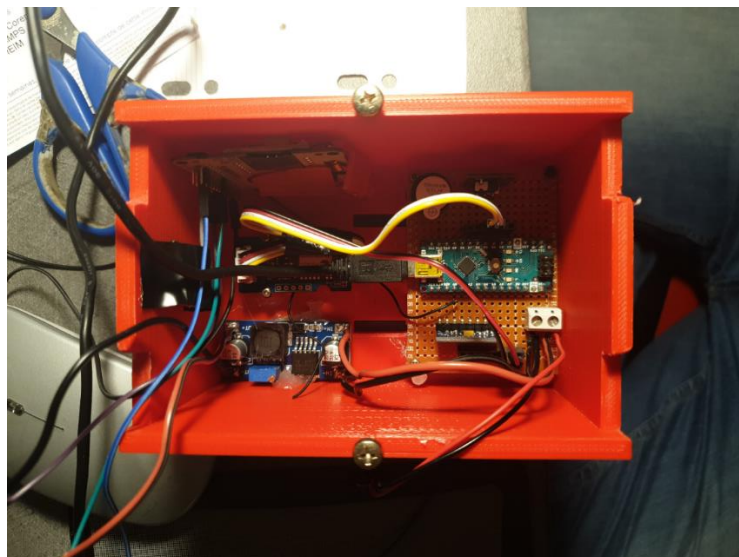
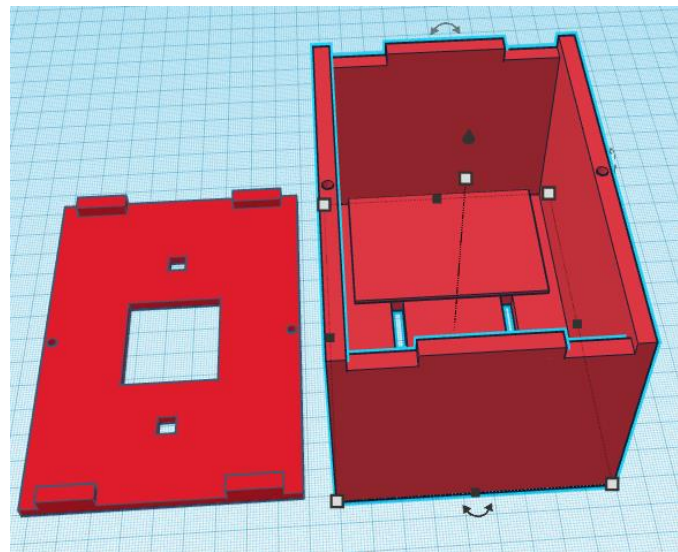
L'utilisateur a tout ce temps pour montrer qu'il va bien en cliquant sur le bouton TVB. S'il montre qu'il va bien, "OK :)" s'affiche sur l'écran, l'utilisateur entend deux longs bips du buzzer et le programme repart dans la boucle principale. Cependant, si le bouton TVB n'est pas pressé, un SMS est envoyé au panneau de contrôle indiquant qu'il y a eu une chute. Le SMS contient également les coordonnées GPS de l'utilisateur. Après avoir envoyé ce SMS, l'écran indiquera que le SMS a été envoyé et le programme retournera à la boucle principale.

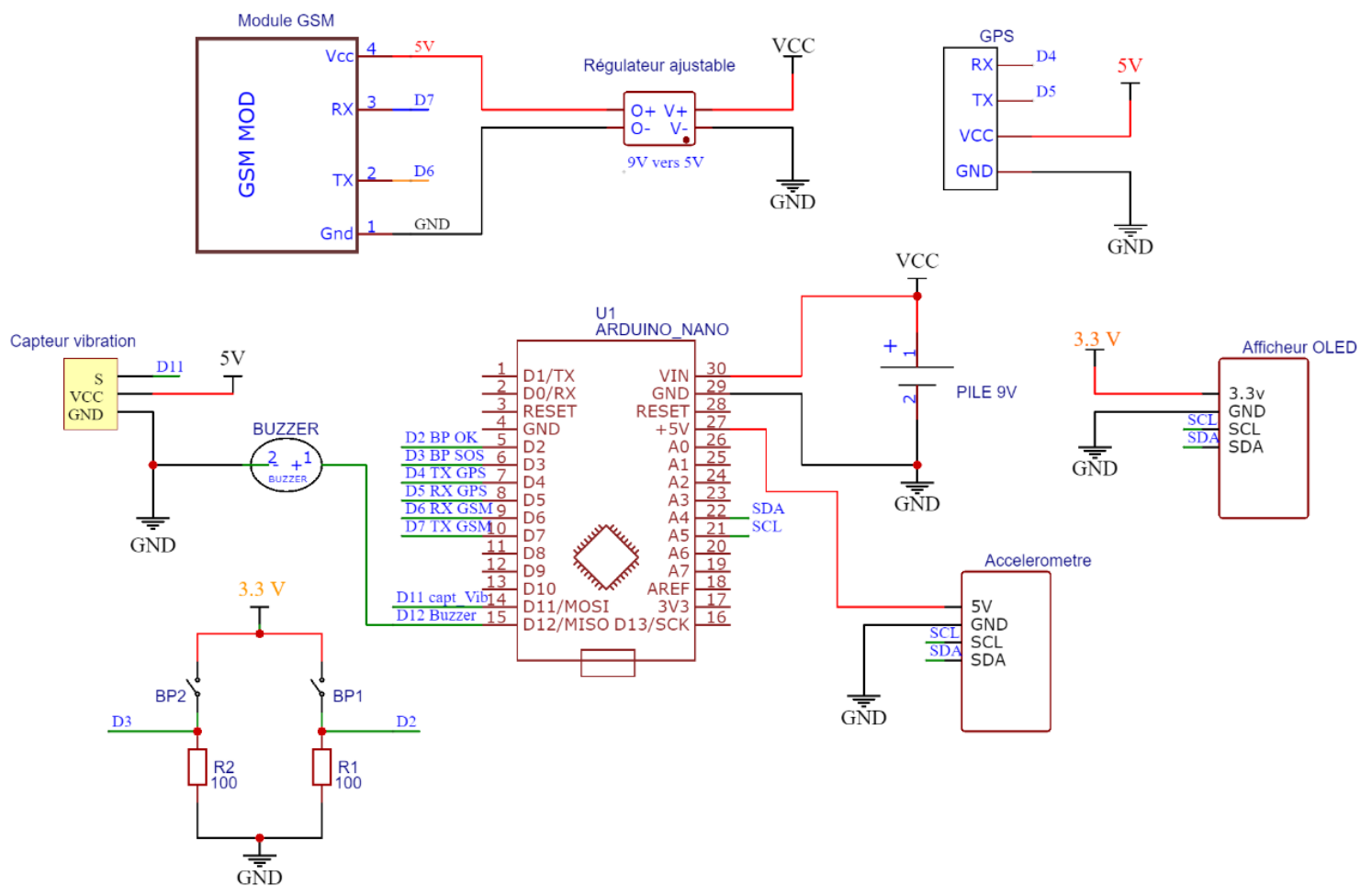
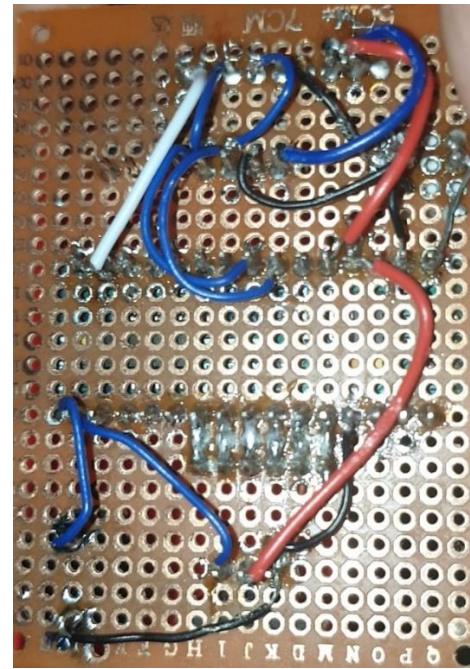
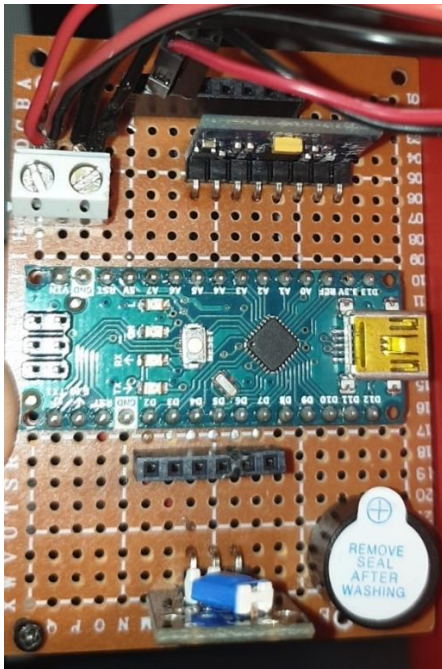
Un appel d'urgence peut également être envoyé via le bouton SOS. Le programme demande alors si le bouton SOS a été actionné. Si le bouton SOS a été actionné, l'écran indique que le bouton a été actionné et l'utilisateur entend dix bips rapides du buzzer. Après cela, il a 10s de temps pour montrer qu'il est ok en appuyant sur le bouton TVB.

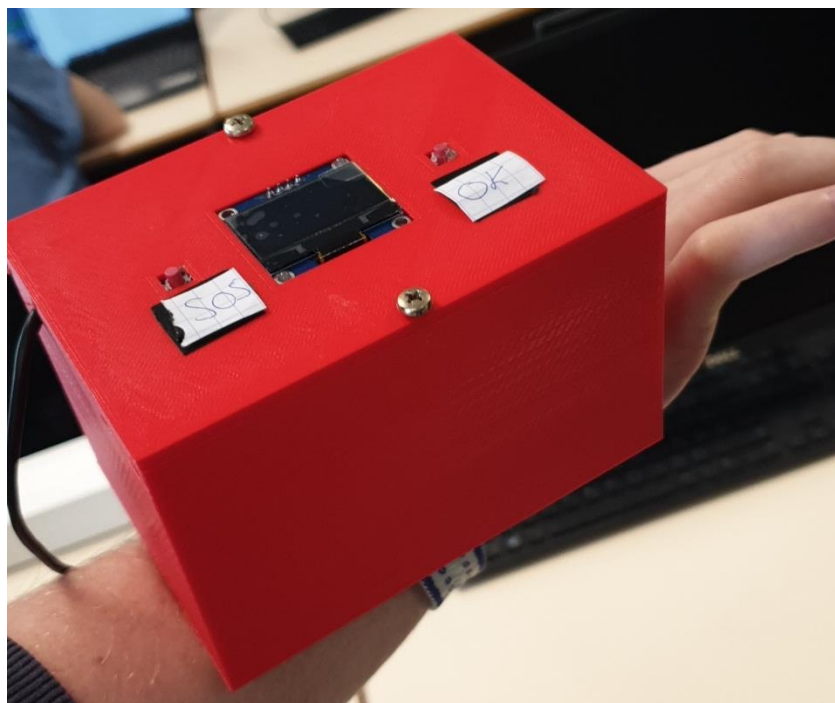
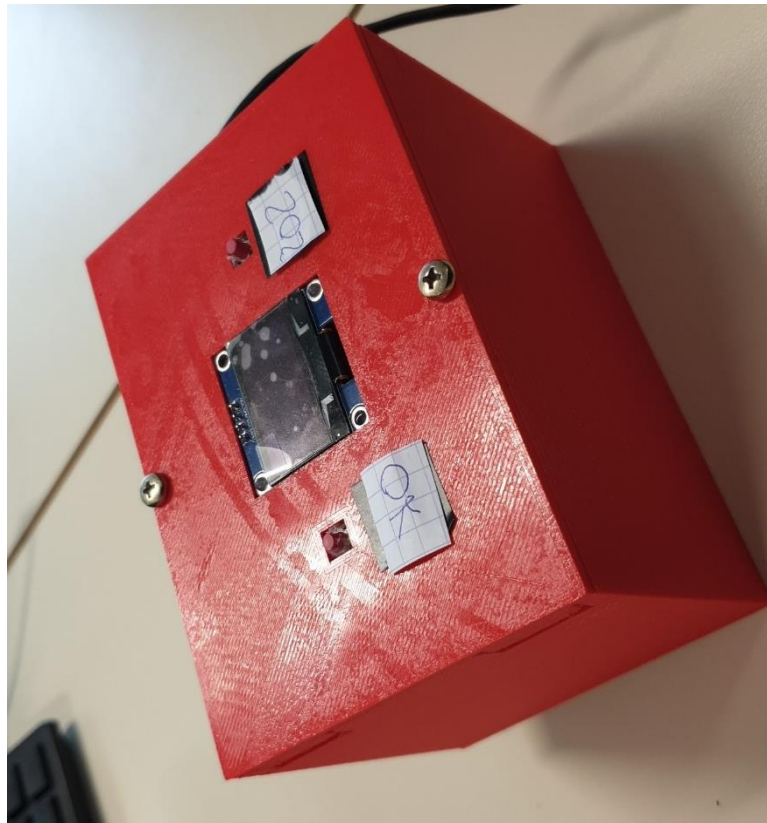
Si le bouton TVB est pressé dans les 10s, l'écran affiche "OK :)", l'utilisateur entend deux longs bips du buzzer et le programme revient à la boucle principale. Cependant, si le bouton TVB n'est pas pressé, un SMS avec "SOS" et les coordonnées GPS de l'utilisateur est envoyé au centre de contrôle. Ensuite, il est indiqué que le SMS a été envoyé et le programme retourne dans la boucle principale. Maintenant la boucle principale recommence avec la demande d'accélération de la part du MPU. Pour toute cette partie Cf. Code du projet

3.3. Forme Final :

Nous nous sommes réparti les tâches pour finaliser le projet. Quentin et Mathieu se sont chargé de la modélisation 3D de la boîte, quant à Corentin, il s'est occupé de souder les composants entre eux et Théo de rédiger le code final. Pour la forme finale du bracelet nous avons décidé de faire une boîte rectangulaire sur 13x9x8,35cm. La boîte est assez grande comparée à l'objectif de départ mais cela nous permet d'avoir tous nos composants à l'intérieur. Nous avons aussi fait des encoches dans le bas de la boîte pour pouvoir y monter une sangle de bracelet. Ce qui suit maintenant seront des images de la boîte en 3D numériques et ensuite réels. :







4. Bilan et perspectives :

Nous avons choisi de créer un bracelet qui détecte les chutes pour les personnes âgées qui sont toujours en mobilité dans la vie de tous les jours. Pour cela on avait un budget initial de 100€ et nous voulions faire un prototype le plus petit possible.

En ayant ces deux points en tête nous avons eu au total une commande de 82,29€ pour l'ensemble de nos composants qui regroupe la détection de la chute, la communication ainsi que la géolocalisation. En pratique nous nous sommes rendu compte qu'un composant ne pouvait pas être utilisé dû à la forme de notre projet c'est-à-dire le capteur de vibration que l'on a remarqué à la fin lors de différents tests.

L'impact de notre projet est d'amener un sentiment de sécurité aux personnes âgées ainsi que leur entourage par ce bracelet qui permet de contacter une équipe en cas de besoin.

Au final nous avons encore un certain nombre d'améliorations qui peuvent être apportées et nous allons n'en citer qu'un certain nombre. La première amélioration est sans aucun doute la forme de la boîte, une forme plus compacte et plus ronde. La deuxième serait l'intégration d'un module d'horloge en temps réel comme le DS3231. La troisième est possiblement un capteur de choc plus adapté qui permet d'exploiter plus la vibration. La quatrième est un nouveau bouton qui permettrait de changer le mode de détection de chute, c'est-à-dire que les gestes brusques ne seront pas des chutes.

4.2. Conclusion :

GROSSARD Mathieu :

Le projet durant cette période, ma permit de me rendre compte encore plus qu'avant l'importance d'une bonne répartition des tâches de travail pour un avancement efficace et intéressant d'un projet que ce soit au niveau professionnel ou personnel. Cela m'a aussi permis de faire quelque rappel et complément sur la création de contenant 3D. Tout en complétant mon savoir-faire sur la programmation et d'intégrer plusieurs petits codes pour en créer un grand qui gère toutes les conditions voulu lors de la création.

LANGE Theo Phil:

Le projet m'a beaucoup aidé à améliorer mes compétences linguistiques. Grâce au travail permanent avec d'autres étudiants, j'ai pu avoir des contacts fréquents pendant la situation de Corona et nous avons bien travaillé ensemble et aussi ri.

J'aime vraiment travailler sur des projets, développer et mettre en œuvre des idées. C'était donc amusant pour moi de développer le projet de plus en plus loin et d'être capable de livrer un appareil fonctionnel à la fin.

Les projets sont également un bon moyen pour moi de pratiquer la structure et la planification. Dans l'ensemble, j'ai pu acquérir beaucoup d'expérience.

Schultz Quentin :

Ce projet est une bonne introduction pour la continuité de mes études, en particulier au niveau du travail en groupe. Ce projet a été très formateur, il a permis de mobiliser des connaissances acquises au cours de notre formation et d'explorer de nouveaux domaines non enseignés comme avec Gantt. Les différentes recherches que nous avons faites ont été très enrichissantes. Cela nous a permis de prendre conscience de l'importance de considérer les diverses parties prenantes lors de la réalisation d'un projet et les réalisations du code et de la modélisation de la boîte.

ANTONY Corentin :

Ce projet s'est révélé très enrichissant dans la mesure où il a consisté en une approche concrète de notre formation, nous avons pris les devants en proposant notre propre vision du projet en le réalisant de A à Z. De plus, ce projet nous a permis d'appliquer nos connaissances en électronique, programmation Arduino et même en soudure. Malgré cette période nous avons fait de notre mieux afin de finaliser au maximum ce projet en nous répartissant équitablement les tâches. Même si le temps fut bref, j'ai aimé travailler avec mes camarades et ça a été pour moi une belle expérience.

4.3. Lexique :

Ampère : L'ampère (symbole A) est l'unité de mesure du Système international d'unités de l'intensité du courant électrique. Un courant d'un ampère correspond au transport d'une charge électrique d'un coulomb par seconde à travers une surface.

E/S : E/S correspond à entrer et sortie, ici cela correspond au différent pin de l'Arduino qui seront déclaré comme une entrée d'informations ou une sortie d'informations.

GPS : Le Global Positioning System (GPS) est un système de géolocalisation mondial. Ainsi, grâce aux satellites et à des récepteurs appelés également GPS, une personne peut non seulement savoir où elle se trouve de façon très précise mais également trouver son chemin pour aller à un endroit.

GSM : Abréviation de Global System for Mobile communication, qui désigne un téléphone portable.

I2C : Inter Integrated Circuit développé au début des années 80 par Philips Semiconducteur pour permettre de relier facilement à un microprocesseur les différents circuits d'un téléviseur moderne. Son But : faire communiquer entre eux des composants électronique très divers grâce à seulement 3 fils : Signal de donnée : SDA ; Signal d'horloge : SCL; Signal de référence électrique : masse

SIM : Une carte SIM (de l'anglais : subscriber identity/identification module) est une puce contenant un microcontrôleur et de la mémoire. Elle est utilisée en téléphonie mobile pour stocker les informations spécifiques à l'abonné d'un réseau mobile¹, en particulier pour les réseaux GSM, UMTS et LTE.

Kb : Un kilobit est une unité d'information (symbole kbit ou kb). La valeur standard d'un kilobit est $10^3 \text{ bit} = 1\,000 \text{ bit}$. Les kilobits sont généralement utilisés pour exprimer la vitesse de communication numérique

Mémoire Flash : La mémoire flash est une mémoire de masse à semi-conducteurs réinscriptible, c'est-à-dire une mémoire possédant les caractéristiques d'une mémoire vive mais dont les données ne disparaissent pas lors d'une mise hors tension. La mémoire flash stocke dans des cellules de mémoire les bits de données qui sont conservées lorsque l'alimentation électrique est coupée.

OLED : Oled est l'acronyme de Organic Light-Emitting Diodes (en français, Diode électroluminescente organique). Un film organique (composé carboné) est pris en sandwich entre deux autres films jouant le rôle d'électrodes. Lorsqu'un courant est appliqué, le composé organique s'illumine.

Volts : Le volt (symbole : V) est une unité, de force électromotrice et de différence de potentiel (ou tension) et dérivée du SI.

4.4. Webographie :

Réalisé par Steeve et Hervé, <https://www.redohm.fr/2015/03/utilisation-du-module-gps-grove-sen10752p/>, 10 avril 2021, Utilisation du module GPS Grove SEN10752P

Réalisé par tiptop administrateur du site, http://tiptopboards.free.fr/arduino_forum/viewtopic.php?f=2&t=28, 9 avril 2021,

Module gyroscope et accéléromètre MPU6050 6 axes + Arduino

Réalisé par Path, <https://www.robot-maker.com/forum/topic/12105-detecter-les-collisions-avec-un-accelerometre/>, 3 avril 2021, Détecter les collisions avec un accéléromètre

Réalisé par Iglinux, <http://lglinux3d.e-monsite.com/pages/arduino/arduino.html>, 5 mars 2021, Arduino avec Oled ssd1306 et U8Glib

Un Wiki donc pas trouvé l'auteur, <https://wiki.seeedstudio.com/Grove-GPS/>, 19 février 2021, Grove – GPS

Réalisé par jojo, <https://www.circuitstoday.com/interface-gsm-module-with-arduino>, 23 avril 2021, How to Interface GSM Module to Arduino-Send and Receive SMS

5. Annexe

5.1. Code du projet

```
1  /*****
2  * Project: Bracelet de chute - Projet Etat de l'art
3  * Authors: Corentin Antony, Mathieu Grossard, Quentin Schultz, Theo Phil Lange
4  *****/
5  #include "U8glib.h"           //Librairie et définition LCD SH1106 128X64
6  #include "Wire.h"
7  #include "I2Cdev.h"
8  #include "MPU6050.h"         // MPU6050
9  #include <SoftwareSerial.h>   // GPS et GSM
10
11
12 /*****
13 * PIN DEFINITIONS
14 *****/
15 #define PIN_BUZZER 12
16 #define PIN_CAPTEUR_VIBRATION 11
17 #define PIN_BUTTON_TV8 2
18 #define PIN_BUTTON_SOS 3
19 #define PIN_GPS_RX 5
20 #define PIN_GPS_TX 4
21 #define PIN_GSM_RX 6
22 #define PIN_GSM_TX 7
23
24
25 /*****
26 * GLOBAL VARIABLE AND OBJECT DEFINITIONS
27 *****/
28 MPU6050 AccelGyro;            // Objet du MPU6050
29 U8GLIB_SH1106_128X64 LCD(U8G_I2C_OPT_NONE); // I2C / TWI
30 SoftwareSerial SoftSerial_GPS(PIN_GPS_RX, PIN_GPS_TX); // objet Serial GPS
31 SoftwareSerial SoftSerial_GSM(PIN_GSM_RX, PIN_GSM_TX); // objet Serial GSM
32
33 unsigned char buffer[64];     // buffer array for data receive over serial port
34 int count = 0;                // counter for buffer array
35
36 float AccelGyro_shock_float = 0.0;
37 volatile bool bool_ButtonTV8 = 0;
38 volatile bool bool_ButtonSOS = 0;
39
40
```

```
41 ▾ /*****
42  * FUNCTIONS
43  *****/
44 void LCD_initialize()
45 ▾ {
46     /* La fonction permet initiaaLiser L'ecran LCD */
47
48     // assign default color value
49 ▾ if ( LCD.getMode() == U8G_MODE_R3G3B2 ) {
50     LCD.setColorIndex(255);                // white
51     }
52 ▾ else if ( LCD.getMode() == U8G_MODE_GRAY2BIT ) {
53     LCD.setColorIndex(3);                // max intensity
54     }
55 ▾ else if ( LCD.getMode() == U8G_MODE_BW ) {
56     LCD.setColorIndex(1);                // pixel on
57     }
58 ▾ else if ( LCD.getMode() == U8G_MODE_HICOLOR ) {
59     LCD.setHiColorByRGB(255,255,255);
60     }
61 }
62
63 void LCD_draw(float shock=0.0, int choice=0, int decompete=0)
64 ▾ {
65     /* La fonction permet d'afficher differentes ecrans predefiné sur Le LCD */
66
67     /* commandes pour L'affichage LCD */
68 ▾ /*
69     * u8g.setFont(u8g_font_unifont);
70     * u8g.setFont(u8g_font_osb21);
71     * u8g.drawStr( 0, 22, "Hello World!");
72     */
73
74     LCD.setFont(u8g_font_unifont); // choisir type des caractères
75
76 ▾ switch(choice){
77 ▾     case 0: {
78         // L'affichage pour une utilisation debug
79         LCD.setPrintPos(0,20);           // curseur a la position (x,y)
80         LCD.print("Bracelet de");       // texte affiché premier ligne sur L'ecran
81         LCD.setPrintPos(0,40);
82         LCD.print("chute");
83         LCD.setPrintPos(80,60);
84         LCD.print(shock);               // affiche shock actuelle
85     }
86     break;
87 ▾     case 1: {
88         LCD.setPrintPos(0,40);
89         LCD.print("Envoyer message");
90         LCD.setPrintPos(50,60);
91         LCD.print("SOS!");
92     }
93     break;
94 ▾     case 2: {                               // l'ecran teste pour une message tout va bien
95         LCD.setPrintPos(0,40);
96         LCD.print("Envoyer message");
97         LCD.setPrintPos(0,60);
98         LCD.print(" tout va bien!");
99     }
100     break;
```

```
101 v   case 3: {
102     LCD.setPrintPos(0,40);
103     LCD.print("Message envoye!");
104 }
105 break;
106 v   case 4: {
107     // L'affichage en mode normal
108     LCD.setPrintPos(35,20);
109     LCD.print("BRACELET");
110     LCD.setPrintPos(60,40);
111     LCD.print("DE");
112     LCD.setPrintPos(47,60);
113     LCD.print("CHUTE");
114 }
115 break;
```

```
116 v   case 5: {
117     LCD.setPrintPos(20,20);
118     LCD.print("Est-ce que");
119     LCD.setPrintPos(10,40);
120     LCD.print("tout va bien?");
121     LCD.setPrintPos(60,60);
122     LCD.print(decompte);
123     LCD.print("s");
124 }
125 break;
126 v   case 6: {
127     LCD.setPrintPos(40,40);
128     LCD.print("OK! :)");
129 }
130 break;
131 v   case 7: {
132     LCD.setPrintPos(20,40);
133     LCD.print("Bouton SOS");
134     LCD.setPrintPos(40,60);
135     LCD.print("presse");
136 }
137 break;
138 v   default: {
139     LCD.setPrintPos(60,40);
140     LCD.print("ERREUR");
141 }
142 }
143 }
144
```

```
145 void MPU_read()
146 {
147     /* Lire Les données brutes acceleration / gyro */
148     int16_t ax, ay, az;           // mesures brutes
149     int16_t gx, gy, gz;          // pour le gyroscope
150     //int16_t ax_moy, ay_moy, az_moy; // moyennes
151     //int16_t gx_moy, gy_moy, gzv;
152     AccelGyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
153     //ax_moy = ax;
154
```

```
155  /* Autre méthodes pour cela */
156  /*
157   * //accelgyro.getAcceleration(&ax, &ay, &az);
158   * //accelgyro.getRotation(&gx, &gy, &gz);
159   */
160
161  /* Possibilité debug pour les valeurs de l'accyleroscope (accel/gyro x/y/z) */
162  /*
163   * Serial.print("a/g:\t");
164   * Serial.print(ax); Serial.print("\t");
165   * Serial.print(ay); Serial.print("\t");
166   * Serial.print(az); Serial.print("\t");
167   * Serial.print(gx); Serial.print("\t");
168   * Serial.print(gy); Serial.print("\t");
169   * Serial.print(gz); Serial.print("\t");
170   * // x > 0 si bascule en avant (tanguage)
171   * // y > 0 si roulis du côté gauche
172   * // z > 0 si orienté en haut
173   */
174
175  //== module détecteur de chocs ou d'accélération (sans 1g)
176  AccelGyro_shock_float = (float)ax*ax + (float)ay*ay + (float)az*az;
177  AccelGyro_shock_float = abs(sqrt(AccelGyro_shock_float)/1000-16); //retirer 1g et absolu
178  Serial.print(AccelGyro_shock_float); Serial.println("");
179
180  /* Possibilité debug la valeur de chock en étoiles */
181  /*
182   * String etoiles = "*****";
183   * String nb_etoiles = etoiles.substring(0,(int)AccelGyro_shock_float);
184   * Serial.print(AccelGyro_shock_float);
185   * Serial.print(" ");
186   * Serial.println(nb_etoiles); //module
187   */
188
189  delay (100);
190  }
191
```

```
192 void clearBufferArray()
193 {
194   /* fonction pour effacer le buffer array */
195   for (int i=0; i<count;i++)
196   {
197     buffer[i]=NULL;
198   } // clear all index of array with command NULL
199 }
200
```



```
201 String GPS_get()
202 {
203     /* récupérer des coordonnées GPS avec une masque string */
204     bool done = 0;
205     char currentchar = '.';
206     while(done == 0){
207         if(SoftSerial_GPS.available()){
208             currentchar = SoftSerial_GPS.read();
209             if(currentchar=='$'){
210                 count = 0;
211                 buffer[count++] = '$';
212             }
213             if(currentchar=='*'){
214                 if(buffer[1]=='G' && buffer[2]=='N' && buffer[3]=='G' && buffer[4]=='G' && buffer[5]=='A'){
215                     Serial.write(buffer,count);
216                     Serial.println(" ");
217                     String GPS = buffer;
218                     Serial.print(GPS);
219                     return GPS;
220                     done = 1;
221                 }
222             }
223             if(currentchar!='*' && currentchar!='$'){
224                 buffer[count++] = currentchar;
225             }
226         }
227     }
228 }
229
```

```
230 void GSM_sendSMS(int choix=0) // ajouter parametre pour donner coordonnées GPS
231 {
232     /* Cette fonction permet d'envoyer une SMS. Avec le premier paramètre on peut choisir le typ du SMS, il y a :
233      * - choix = 0: Je suis tombé avec coordonnées GPS
234      * - choix = 1: Tout va bien
235      * - choix = 2: SOS
236      */
237     Serial.println("Get GPS...");
238     String GPS = GPS_get();
239     Serial.println("coordonnées GPS récupéré...");
240     Serial.println("Sending SMS..."); // Show this message on serial monitor
241     String message;
242     switch(choix){
243         case 0: { message = "Je suis tombe *COORDINATES GPS* "; message += GPS; Serial.println(message); }
244             break;
245         case 1: message = "Tout va bien!"; // option pour tester
246             break;
247         case 2: { message = "SOS! *COORDINATES GPS* "; message += GPS; Serial.println(message); }
248             break;
249         default: Serial.println("ERROR - Choisir type de SMS");
250     }
251     SoftSerial_GSM.print("AT+CMGF=1\r"); // Set the module to SMS mode
252     delay(100);
253     SoftSerial_GSM.print("AT+CMGS=\"+33769542703\"\r"); // numero de telephone du destinaire
254     delay(500);
255     SoftSerial_GSM.print(message); // message a l'envoyer pour le destinaire
256     delay(500);
257     SoftSerial_GSM.print((char)26);
258     delay(500);
259     SoftSerial_GSM.println();
260     Serial.println("Text Sent.");
261     delay(500);
262 }
```

```
266 /*****
267  * SETUP
268  *****/
269 void setup(){
270     Wire.begin();           // Initialisation bus I2C
271     SoftSerial_GSM.begin(9600); // Initialisation communication GSM
272     SoftSerial_GPS.begin(9600); // Initialisation communication GPS
273     Serial.begin(9600);      // Initialisation communication PC sur console serie
274     Serial.println("-----SETUP-----");
275     Serial.println(AccelGyro.testConnection() ? "MPU6050 connection reussie" : "MPU6050 connection echec");
276     Serial.println("Initialisation bus serie fini");
277
278     pinMode(PIN_BUZZER, OUTPUT);
279     pinMode(PIN_CAPTEUR_VIBRATION, INPUT);
280     attachInterrupt(digitalPinToInterrupt(PIN_BUTTON_TV), ISR_toutVaBien, FALLING);
281     attachInterrupt(digitalPinToInterrupt(PIN_BUTTON_SOS), ISR_SOS, FALLING);
282     Serial.println("Definition des PINS fini..");
283
284     AccelGyro.initialize();
285
286     Serial.println("Initialisation d'Accelometre + Gyroscope fini..");
287
288     LCD_initialize();
289     Serial.println("Initialisation de LCD fini");
290
291     Serial.println("-----LOOP-----");
292 }
293
294
```

```
295 /*****
296  * LOOP
297  *****/
298 void loop(){
299     MPU_read(); // Lit les valeurs de l'accelerometre
300
301     //bool CapVib_bool = digitalRead(PIN_CAPTEUR_VIBRATION); // CAPTEUR VIBRATION
302     if(AccelGyro_shock_float >= 23){ // definition l'intensité d'une chock qui est une chute (ici 23)
303         /* Si un choc se produit, on active le buzzer */
304         digitalWrite(PIN_BUZZER,HIGH);
305         bool_ButtonTVB = 0; // on traite juste le bouton TVB si nous sommes apres une chute
306                             // ca evite que nous avons une chute qui n'est pas traité
307
308         for(int decompote = 10; decompote>=0; decompote--){
309             // Apres une chute la person a dix secondes pour indiquer avec le bouton que tout va bien
310             LCD.firstPage();
311             do{
312                 LCD.draw(AccelGyro_shock_float,5,decompote);
313             }while( LCD.nextPage() );
314
315             if(bool_ButtonTVB){ break; }
316             delay(1000);
317         }
318         digitalWrite(PIN_BUZZER,LOW);
319     }
320 }
```

```
320     if(bool_ButtonTVB){ // si La person indique que tout va bien on envoie pas une message
321         LCD.firstPage();
322         do{
323             LCD_draw(AccelGyro_shock_float,6);
324         }while( LCD.nextPage() );
325         delay(300);
326         for(int i=0; i<2; i++){
327             digitalWrite(PIN_BUZZER,HIGH);
328             delay(300);
329             digitalWrite(PIN_BUZZER,LOW);
330             delay(300);
331         }
332
333         delay(2000);
334         bool_ButtonTVB = 0;
335     }
336     else{ // si La person n'indique pas que tout va bien on envoie une message SOS
337         LCD.firstPage();
338         do{
339             LCD_draw(AccelGyro_shock_float,1);
340         }while( LCD.nextPage() );
341
342         GSM_sendSMS(0);
343
344         LCD.firstPage();
345         do{
346             LCD_draw(AccelGyro_shock_float,3);
347         }while( LCD.nextPage() );
348
349         delay(2000);
350         bool_ButtonTVB = 0;
351     }
352 }
353 else{
354     LCD.firstPage();
355     do{
356         LCD_draw(AccelGyro_shock_float,4);
357     }while( LCD.nextPage() );
358 }
359
```

```
360 /* Routine pour tester Le bouton Tout va Bien */
361 ~ /*
362  * if(bool_ButtonTVB){
363  *     LCD.firstPage();
364  *     do{
365  *         LCD_draw(AccelGyro_shock_float,2);
366  *     }while( LCD.nextPage() );
367  *
368  *     for(int i=0; i<2; i++){
369  *         digitalWrite(PIN_BUZZER,HIGH);
370  *         delay(300);
371  *         digitalWrite(PIN_BUZZER,LOW);
372  *         delay(300);
373  *     }
374  *     GSM_sendSMS(1);
375  *
376  *     LCD.firstPage();
377  *     do{
378  *         LCD_draw(AccelGyro_shock_float,3);
379  *     }while( LCD.nextPage() );
380  * }
```

```

381  *   delay(2000);
382  *   bool_ButtonTVB = LOW;
383  * }
384  */
385

```

```

386  if(bool_ButtonSOS){
387      /* Envoie La message "SOS!" si La bouton a droit est presé */
388      LCD.firstPage();
389      do{
390          LCD_draw(AccelGyro_shock_float,7);
391      }while( LCD.nextPage() );
392
393      for(int i=0; i<10; i++){          // son indication que Le bouton SOS etait pressé
394          digitalWrite(PIN_BUZZER,HIGH);
395          delay(50);
396          digitalWrite(PIN_BUZZER,LOW);
397          delay(50);
398      }
399      delay(500);
400
401      bool_ButtonTVB = 0;              // on traite juste Le bouton TVB si nous somme apres une chute
402                                          // ca eveite que nous avons une chute qui n'est pas traité
403
404      for(int decompote = 10; decompote>=0; decompote--){
405          LCD.firstPage();
406          do{
407              LCD_draw(AccelGyro_shock_float,5,decompote);
408          }while( LCD.nextPage() );
409
410          if(bool_ButtonTVB){ break; }
411          delay(1000);
412      }
413

```

```

414  if(bool_ButtonTVB){
415      LCD.firstPage();
416      do{
417          LCD_draw(AccelGyro_shock_float,6);
418      }while( LCD.nextPage() );
419      delay(300);
420      for(int i=0; i<2; i++){
421          digitalWrite(PIN_BUZZER,HIGH);
422          delay(300);
423          digitalWrite(PIN_BUZZER,LOW);
424          delay(300);
425      }
426      delay(2000);
427  }
428  else{ // si la person n'indique pas que tout va bien on envoie une message SOS
429      LCD.firstPage();
430      do{
431          LCD_draw(AccelGyro_shock_float,1);
432      }while( LCD.nextPage() );
433
434      GSM_sendSMS(2);
435
436      LCD.firstPage();
437      do{
438          LCD_draw(AccelGyro_shock_float,3);
439      }while( LCD.nextPage() );
440
441      delay(2000);
442  }
443  bool_ButtonTVB = 0;
444  bool_ButtonSOS = 0;
445  }
446  }

```

```
447
448
449 /*****
450  * INTERRUPT ROUTINES
451  *****/
452 void ISR_toutVaBien(){
453     bool_ButtonTVB = HIGH;
454 }
455
456 void ISR_SOS(){
457     bool_ButtonSOS = HIGH;
458 }
459
460
461 /*****
462  * END
463  *****/
```

5.2. Ecrans différents



5.3. Sensibilité du capteur avec différentes études

