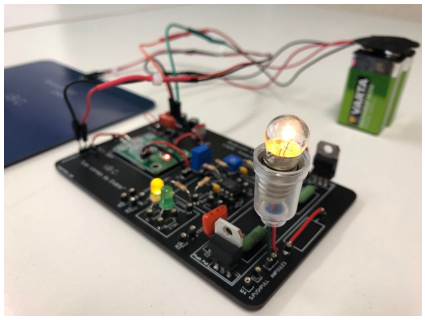
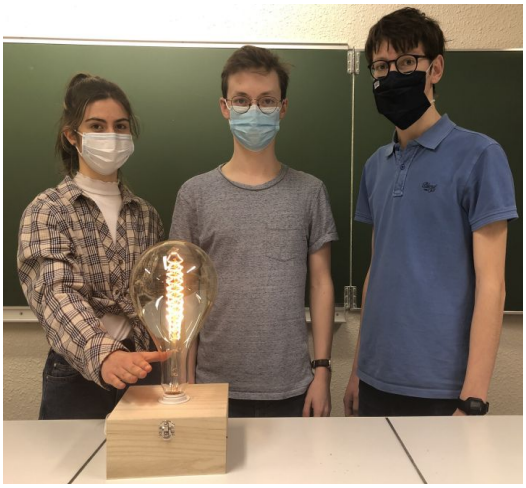
Concours CGénial Sciences à l'école - édition 2020/2021

Intra Body Communication (IBC)
"VOIR" et Utiliser les OEM rayonnées par le corps humain





Problématique

" Comment transmettre des données binaires par le corps humain ? "

L'équipe "Bodycom 2.0" (2020/2021)

- Nahomé VESVARD (chercheuse en physique et électronique)
- Clément DESJONQUERES (développeur Java / Android)
- Marin LUET (concepteur 3D et développeur Arduino)

Encadrés par M. Levionnais, enseignant en Sciences-Physiques

Un grand Merci à nos partenaires qui nous ont soutenu et aidé dans notre projet :

- **Société ORANGE** : prêt d'un analyseur de spectre portable, simulations logiciels, échange d'informations, aide à la conception de circuits électroniques
- **Société NXP** (Semiconducteurs) : don de téléphones (Xiaomi 6), échange d'informations
- **GREYC** (Laboratoire de recherche en sciences du numérique) / **UNICAEN** (Université de Caen) : prêt d'un générateur/oscilloscope numérique, échange d'informations
- **Fablab Le Dôme** : impression 3D, découpe LASER
- **CGénial** : participation à l'achat d'un oscilloscope numérique 4 voies en 2019/2020

SOMMAIRE

La page des vidéos 2019/2020 et 2020/2021 : **https://www.ibc.ovh** (ibc.ovh)

Introduction / Résumé

I. L'Interface Homme-Machine (IHM) et son comportement électrique (p5)
   1. L'IHM se comporte comme une résistance (p5)
   2. L'IHM se comporte comme un condensateur (p6)
   3. Comment transmettre un signal électrique à travers l'IHM (choix de la fréquence) ? (p6)

II. Le dispositif imaginé pour venir en aide aux personnes en situation de handicap (p8)
   1. Schéma détaillé du dispositif imaginé (p8)
   2. Emission des données binaires à transmettre vers le corps (p11)
   3. Réception des données binaires transmises par le corps (p12)
   4. Problème : un câble en moins ! (p13)
   5. Conception d'antennes émettrices et réceptrices (p15)

III. Le corps humain se comporte comme une antenne rayonnante (p15)
   1. Mise en évidence du rayonnement électromagnétique du corps humain (p15)
   2. Comment capter un faible rayonnement ? (p17)
   3. Test de réception du rayonnement (p18)
   4. « Voir » les OEM : "Les cornes du Diable" et la boîte magique (p19)

IV. Utiliser les OEM rayonnées par le corps humain : le dispositif final (p21)
   1. Le circuit de réception intégré au Smartphone (p21)
   2. L'aspect logiciel et application mobile (p22)
   3. Les maquettes 3D de démonstration (p23)

Conclusion

Annexes

1. Sources
2. Protocoles informatiques et schémas de données
3. Extrait du code Java
4. Schémas des circuits électroniques

Introduction / Résumé

Tous les jours nous effectuons des tâches comme ouvrir une porte, appeler un ascenseur, valider un titre de transport, effectuer un paiement bancaire, etc. Ces tâches en apparence simples peuvent s'avérer très compliquées pour une personne en situation de handicap (amputée, paralysée, malvoyante…), d'autant plus qu'elles sont répétitives. **Cynthia**, notre camarade de terminale (en 2019/2020), atteinte de la maladie de Charcot-Marie-Tooth (maladie dégénérative) nous a expliqué que le simple fait de valider un titre de transport peut s'avérer très compliqué à cause de ses béquilles, de son manque d'équilibre, et de la malformation de ses mains.

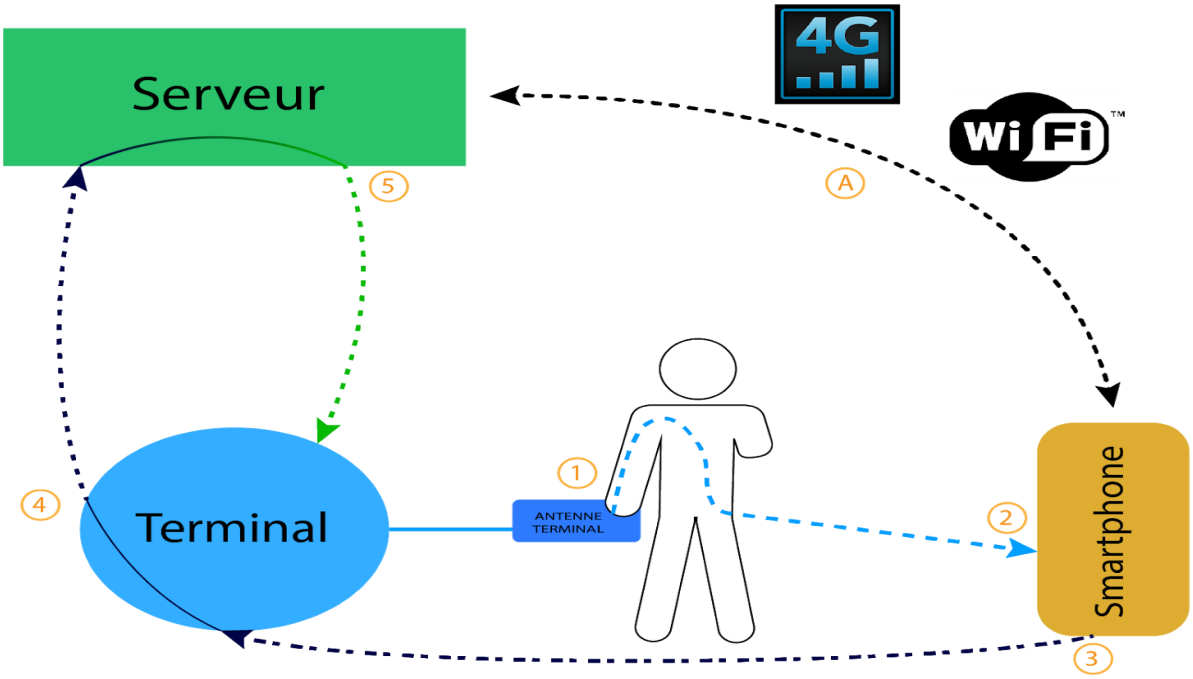
Ainsi notre objectif a été de développer un dispositif permettant de faciliter le quotidien des personnes en situation de handicap, mais également utile à tous, handicapés ou non. Ce projet consiste à utiliser **une partie du corps humain** (la main, le bras, etc) comme **vecteur de transmission des données**, afin d'établir des communications avec des dispositifs électriques, sans utiliser de puces dans le corps (donc non invasif) et sans utiliser de méthodes biométriques (pas de fichage).

Exemple : la validation d'une transaction CB (ou d'un ticket de bus, de métro…)	
Sans contact <u>actuel</u> (le NFC*) : une personne approche son **Smartphone** du **Terminal**, et la transaction CB (ou le ticket de bus, de métro…) est validée * NFC = Near Field Communication	 Un Terminal NFC
Notre projet : le sans contact du <u>futur</u> (l'IBC*) : une personne (handicapée ou non) positionne une partie de son **corps** (main, bras, jambe…) sur l'antenne du **Terminal**, et la transaction est validée. Le **Smartphone** reste dans la **<u>poche</u>** (ou dans un sac) de l'utilisateur. * IBC = Intra Body Communication	 L'antenne de notre Terminal IBC

" Comment transmettre des données binaires par le corps humain ? "

Vidéos de démonstration à voir sur https://www.ibc.ovh (ibc.ovh)

Voici le schéma simplifié du fonctionnement de notre projet :



Les <u>lettres</u> et <u>numéros</u> indiqués sur le schéma correspondent aux étapes ci-après.

Nous prendrons ici l'exemple de la **validation d'un ticket de bus**

<u>Etapes</u> :

A) L'utilisateur peut acheter des tickets de bus avec son smartphone en communiquant avec le **Serveur** à l'aide d'une connexion 4G, 3G, ou par Wi-Fi.

1) L'utilisateur (en situation de handicap ou non) fait une **<u>action volontaire</u>** : il approche une partie de son corps de l'antenne du **Terminal** (situé dans le bus)

2) Une connexion est effectuée entre le **Terminal** et le **Smartphone** de l'utilisateur (qui est dans sa poche ou dans un sac), en **passant par le corps** de l'utilisateur

3) Le **Smartphone** de l'utilisateur envoie une réponse par Bluetooth au Terminal

4) Le **Terminal** transmet le message reçu au **Serveur** (situé en dehors du bus) par le réseau Internet

5) Le **Serveur** analyse la réponse, et envoie au **Terminal** le résultat : **Autorisé** ou **Refusé** (toujours par le réseau Internet)

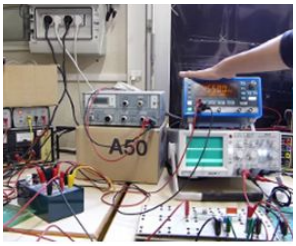
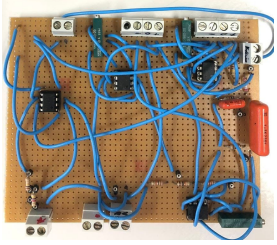
6) Le ticket de bus est alors validé (ou non). Si nécessaire le **Terminal** commande l'ouverture d'une porte ou d'une barrière (ce serait le cas dans le métro par exemple, ou à un péage autoroutier…)

Résumé des objectifs pour CGénial 2020 et 2021 :

- Montrer que le corps humain peut être utilisé comme antenne émettrice pour transmettre des données binaires
- Développer un émetteur pour envoyer les bits vers le corps humain
- Utiliser un capteur pour détecter le rayonnement émis par le corps humain
- Programmer le Terminal et le Serveur en Java
- Développer une application mobile Android en Java qui interprète les données binaires ayant transitées par le corps humain
- Rendre le dispositif autonome et miniaturisé pour qu'il puisse être intégré à un vrai Smartphone
- Modéliser en 3D la coque du Smartphone pour accueillir le circuit de réception
- Modéliser en 3D une barrière ouvrante pour rendre concret l'action réalisable une fois la transaction validée (en y intégrant un microcontrôleur Arduino)
- Développer l'aspect démonstration en créant des dispositifs pour "Voir" le rayonnement du corps humain et expliciter notre travail

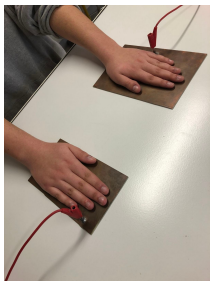
De plus, dans sa version finale, notre dispositif **fonctionne à distance**, sans qu'il y ait de contact direct entre le corps humain et l'antenne du Terminal. Dans un contexte malheureux de propagation de virus, ce dispositif serait parfaitement adapté.

Voici en 4 photos l'évolution du projet :

Les débuts	Les premières plaquettes électriques	Miniaturisation avec circuit imprimé et composants traversants, mais l'alimentation restait externe	Miniaturisation avec circuit imprimé et composants de surface, avec alimentation interne par le Smartphone lui-même
			

I- L'Interface Homme-Machine (IHM) et son comportement électrique

1) L'IHM se comporte comme une résistance



Pour faire passer des données à travers le corps humain depuis un dispositif électrique il est nécessaire de faire un **lien entre la machine (les dispositifs électriques) et le corps humain** : ce lien sera fait dans un premier temps avec deux **antennes,** chacune étant constituée d'une simple **plaque de cuivre** sur laquelle on pose une partie du corps (la main, le bras, etc.), la peau étant directement en contact avec les plaques de cuivre.

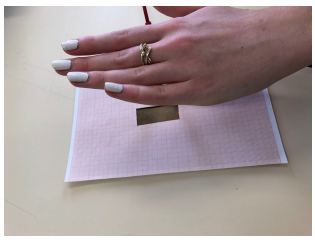
Interface Homme-Machine :

IHM = {Antenne plaque de cuivre - Corps humain - Antenne plaque de cuivre}

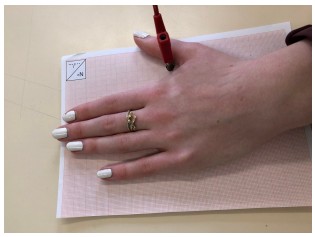
La première expérience a consisté à mesurer la résistance de l'**IHM** avec un multimètre. En fonction de la personne et de la partie du corps posée sur les antennes cela varie beaucoup, entre **50 kΩ** et **1 MΩ**.

Ainsi si on applique une tension de 10 V entre deux parties du corps l'intensité sera au maximum $I = \frac{U}{R} = \frac{10}{50.10^3}$ = 0,2 mA << 3 mA (seuil de sensation douloureuse). Le projet ne présente donc **pas de danger** du point de vue électrique.

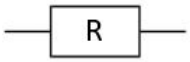
Nous avons ensuite souhaité voir si la **surface de la main en contact** avec la plaque avait une influence, car une personne handicapée peut avoir des difficultés pour poser la main **à plat** sur l'antenne, c'est le cas de **Cynthia** (photo de droite), la surface en contact est donc diminuée.



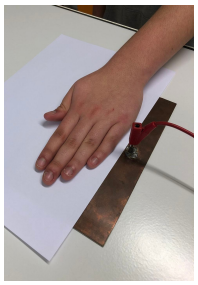
Pour cette étude nous avons découpé dans une feuille de papier millimétré la surface voulue puis nous avons posé la feuille sur la plaque de cuivre. Le cobaye pose sa main sur la surface découpée de la feuille, ainsi seule cette surface est en contact direct avec la plaque de cuivre. On relève alors la valeur de la résistance à l'ohmmètre et on fait des séries de 10 mesures pour obtenir une moyenne et plus de précision.



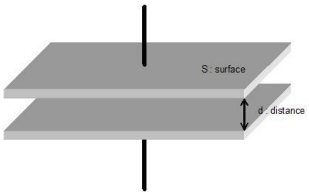
Nous avons observé une **augmentation de la résistance** lorsque la **surface diminue**. A ce stade l'**IHM** est modélisable par une simple résistance qui dépend de la personne et de la surface du corps en contact avec la plaque de cuivre.



2) L'IHM se comporte comme un condensateur



Notre 1er modèle est **trop limité**, en effet le contact entre les antennes et le corps humain ne se fera **jamais** par contact direct avec la peau car dans notre dispositif il y aura toujours la **présence de vêtements** ou **de coques de protection isolantes** entre le corps et les antennes. Pour **simuler cet isolant** nous avons utilisé des feuilles de papier car c'est facile de faire varier l'épaisseur.

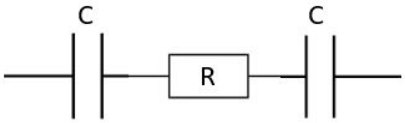


Un **condensateur** plan se forme alors par géométrie car un condensateur est un composant fait de deux armatures conductrices séparées par un isolant. La 1ère armature conductrice est la **plaque de cuivre**, l'**isolant** est la feuille de papier et la 2ème armature conductrice est la partie du **corps humain** posée sur la feuille.

On a calculé la <u>capacité</u> du dispositif grâce à la formule du condensateur plan : $C = \varepsilon \times \frac{S}{d}$ avec ε la permittivité de l'isolant, S la surface de la plus petite des plaques, et d la distance entre celles-ci. Pour une main de 80 cm², une feuille de 0,1 mm d'épaisseur et en considérant que l'isolant est assimilable au vide, on obtient : $C = (8{,}9.10^{-12} \times 80.10^{-4})/(0{,}1.10^{-3}) = 7\times10^{-10}$ F. Donc un ordre de grandeur de 10^{-9} F = **1 nF.** Si la distance augmente à 1 cm la capacité diminue à **10^{-11} F**. Ce sont des petites capacités mais nous verrons que cela aura malgré tout une **grande importance** pour la transmission des données.

IHM = {Antenne plaque de cuivre - Isolant - Corps humain - Isolant - Antenne plaque de cuivre}

L'ensemble est donc modélisé de la façon suivante (schéma) :

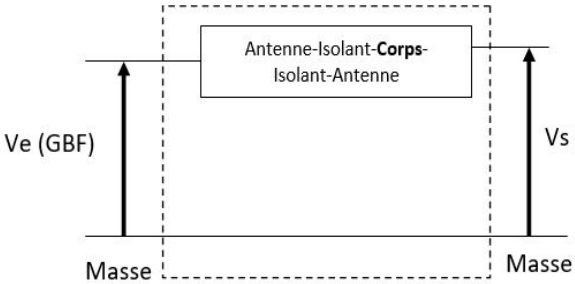


3) Comment transmettre un signal électrique à travers l'IHM (choix de la fréquence) ?

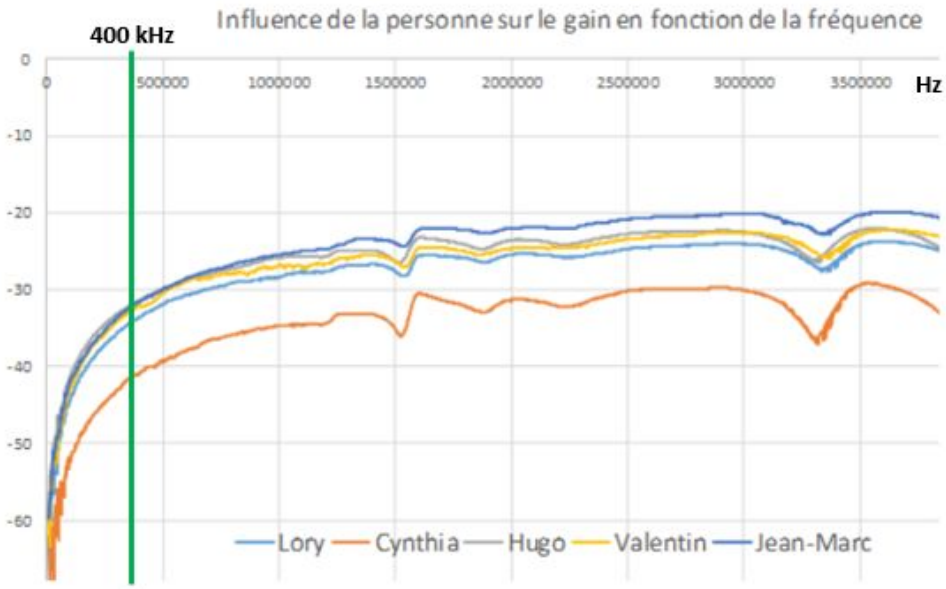
Nous aboutissons à un modèle où notre **IHM** a un comportement **résistif** et **capacitif**. Or une fois chargé un condensateur ne laisse plus passer le courant continu, il se comporte comme un **interrupteur ouvert**. Il faut donc soumettre l'**IHM** à une tension variable.

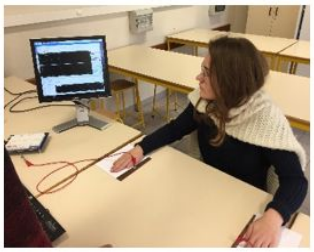
Nous avons alors souhaité connaître le comportement de l'**IHM** en fonction de la **fréquence** et de la **personne**.

<u>**Dispositif**</u> : **Ve** est un signal sinusoïdal d'amplitude 1,0 V et de fréquence variable. On mesure l'amplitude de **Vs**. Les mesures ont été automatisées grâce au matériel prêté par le GREYC / UNICAEN permettant de faire varier automatiquement la fréquence de Ve.



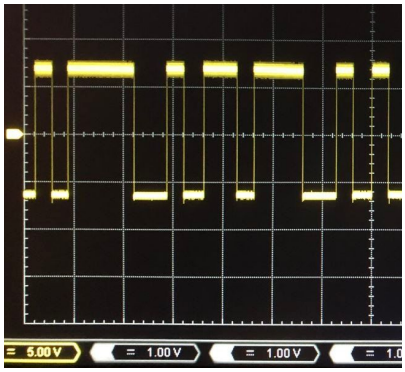
On trace ensuite $Gain(dB) = 20 \times log(\frac{Vs}{Ve})$





On constate que **les basses fréquences sont très atténuées**. Au-dessus de 400 kHz environ le gain est assez élevé, ainsi on pourra utiliser un signal de **fréquence supérieure à 400 kHz** pour **transmettre les données**. On note également que le gain dépend de la personne : il est **plus faible pour Cynthia** qui **ne peut pas poser ses mains bien à plat** sur l'antenne du fait de son handicap ; le gain devra rester assez élevé pour que le dispositif final fonctionne quelle que soit la personne et avec la présence des vêtements.

L'information que nous voulons transmettre est un **signal binaire**, donc une suite de **0 et 1**, qui sera obtenue par une tension qui utilise la norme de transmission série **RS232** : un **1** correspond à une **tension positive** et un **0** à une **tension négative**. La fréquence des 0 et 1 dépend du débit de transmission exprimé en bauds. Nous utiliserons un débit maximum de **9600 bauds** (valeur classique pour la transmission série), ce qui correspond à 9600 bits / seconde donc à **4800 Hz** en cas d'alternance successive de 0 et 1.

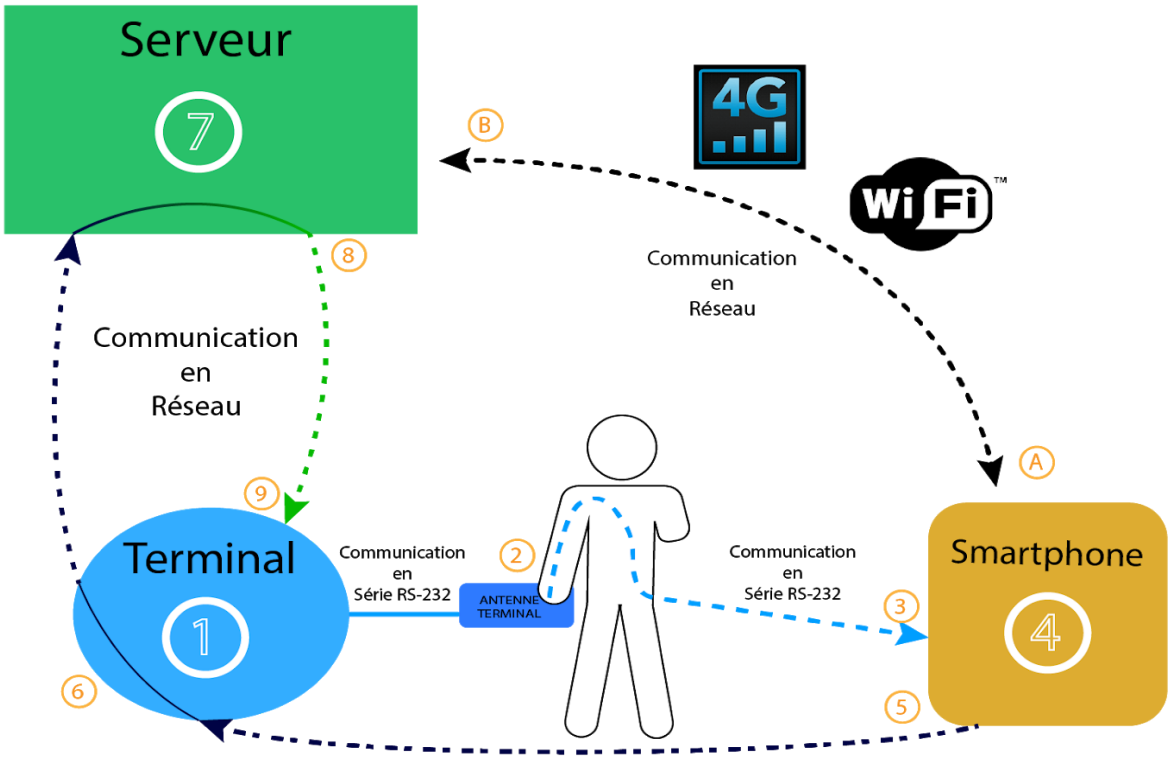


<u>**Conclusion**</u> : la fréquence du signal binaire est trop faible pour ne pas être fortement atténuée par l'**IHM**. Il faudra donc utiliser une **porteuse** de plus **haute fréquence** et **moduler l'amplitude** de la porteuse par le signal binaire à transmettre. Nous fixons notre choix sur une fréquence de **455 kHz** pour plusieurs raisons :

- Cette fréquence est supérieure à 400 kHz donc elle sera peu atténuée
- Orange nous la conseille car elle n'est utilisée par aucune autre application ou radio
- Chaque application doit avoir sa propre fréquence porteuse

II- Le dispositif imaginé pour venir en aide aux personnes en situation de handicap

1) Schéma détaillé du dispositif imaginé



Les lettres et numéros indiqués sur le schéma correspondent aux étapes ci-dessous.

Point de vue du développeur :

Un protocole est une construction logique de données afin de créer un message lisible par un programme. C'est un moyen efficace pour transmettre les données sur **une seule chaîne de caractères**. Pour notre système, nous avons construit **nos propres protocoles** afin de gagner en longueur de caractères envoyés. Ainsi, nous avons pu gagner suffisamment de caractères pour obtenir la **transmission la plus rapide possible**.

*Remarque : les protocoles que nous avons créés (**Protocole-1 à Protocole-4**), l'organisation des données et les permissions sont détaillés en **Annexe 2**, et un extrait du code **Java** se trouve en **Annexe 3**.*

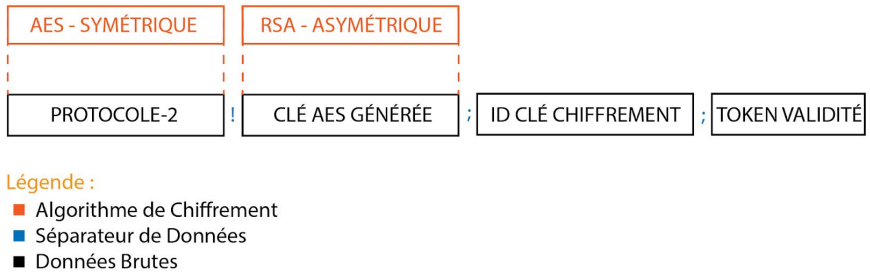
Les étapes :

A) Au premier lancement de l'**application pour Smartphone**, celle-ci va générer une **clé de chiffrement**, et se déclarer au **Serveur**. Le compte utilisateur est ainsi créé, le Serveur possède la **clé** permettant de **déchiffrer les messages** de ce Smartphone. Ce message utilise le **Protocole-4**.

B) Le **Serveur** enregistre dans son fichier de stockage “Utilisateur” les **informations de l’utilisateur** (son nom, son nombre de tickets, etc..) et dans le fichier des “Clés”, la **clé de chiffrement** fournie, ainsi que l’**identifiant de clé de chiffrement**. Ensuite, il envoie les valeurs à l’utilisateur lorsque celui-ci le demande.

1) **Terminal** : un **Token** (4 caractères aléatoires) est créé afin de garantir l’**authenticité et l’unicité de la demande**. Ensuite, le Terminal va autoriser les connexions **Bluetooth**, pour permettre aux utilisateurs de se connecter. Enfin, il va commencer la procédure de démarrage de la communication **Série** vers son antenne émettrice. Il va envoyer en boucle le **message** (**Protocole-1**) constitué de l’**ordre à effectuer** (par exemple “composter un ticket de bus”), du **Token** créé précédemment, de l’**adresse bluetooth du Terminal**, et d’un **nombre** dont la signification dépend de l’ordre à effectuer (sur l’exemple du ticket de bus ou métro ce nombre vaut 1 pour débiter un ticket). Cette boucle s’arrête lorsqu’une réponse bluetooth est réceptionnée, et que le Token fourni dans la réponse correspond à celui envoyé en série.
2) L’**utilisateur** crée la connexion entre le **Terminal** et le **Smartphone** en **posant une partie de son corps** sur **l’Antenne du Terminal**. Les données passent par le **corps humain**. Le Smartphone peut les lire depuis sa prise USB-C reliée au circuit récepteur.
3) Les données reçues ne sont pas exploitables en tant que telles. Elles correspondent au message envoyé en boucle. Elles contiennent donc plusieurs fois le message. L’application pour Smartphone doit donc extraire un message parmi ceux reçus pour l’exploiter. Tout d’abord pour s’assurer que l’utilisation du dispositif est bien le fruit d‘un **acte volontaire**, il attend une dizaine de messages dans les données reçues pour commencer à les exploiter. Cette attente correspond à environ 2 secondes pendant lesquelles l’utilisateur doit garder sa main sur l’antenne (cette durée est réglable). Enfin pour exploiter les données, il va utiliser le caractère de fin de protocole (#) en déduisant qu’un message complet se situe entre deux de ces caractères (**Protocole-1**).
4) **Smartphone** : le message est extrait, puis analysé. Il contient le Token, l’Adresse Bluetooth à contacter et les informations de transaction. Le Smartphone recrée ensuite un message en utilisant le **Protocole-2**. Celui-ci contient les instructions nécessaires au **Serveur** pour effectuer la transaction. Le **Smartphone** va ensuite constituer le **Protocole-3** (voir schéma ci-dessous) à envoyer vers le **Terminal**.

GESTION CRYPTOGRAPHIE COMMUNICATION - PROTOCOLE-3



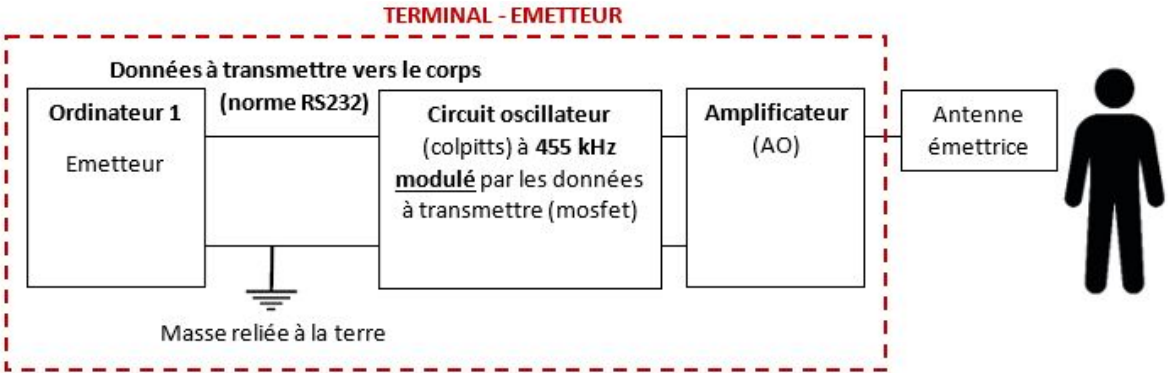
<u>Note</u> : l'utilisation de deux algorithmes cryptographiques différents permet d'optimiser le temps d'exécution sans diminuer la sécurité du dispositif. Il faut prendre en compte que la conception de ce protocole est faite par un smartphone, nous devons donc nous adapter à la puissance moyenne des smartphones du marché. Nous avons économisé près de 3 secondes d'exécution avec cette gestion.

L'AES est plus rapide d'exécution, comparé au RSA. Le désavantage de l'AES, est qu'il est symétrique. Une clé unique permet de chiffrer comme déchiffrer une donnée. Le RSA est lui asymétrique. Il y a dans ce cas deux clés : l'une chiffre, et l'autre déchiffre. Le RSA est donc plus sécurisé pour des transmissions de données chiffrées. Pour plus de rapidité, nous chiffrons le Protocole-2 en AES, et nous fournissons la clé ayant chiffré celui-ci dans le protocole. Cette clé est elle-même chiffrée en RSA, avec la clé permettant de chiffrer. L'autre clé RSA permettant de déchiffrer est stockée uniquement sur le serveur, soit le destinataire de ce protocole. Ainsi, seul lui pourra déchiffrer la clé AES, dans le but de déchiffrer le message chiffré avec cette clé.

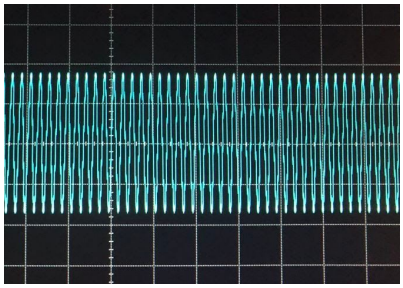
5) Le message préparé est envoyé par le **Smartphone** en **Bluetooth** de manière **sécurisée**.
6) Le **Terminal** reçoit le message par Bluetooth et **vérifie le Token**. S'il est correct, le **Terminal** transmet ce qu'il a reçu par Bluetooth au **Serveur** en utilisant une connexion réseau. Sinon, il attend une nouvelle connexion Bluetooth.
7) **Serveur** : lors de la réception des données du **Terminal**, le **Serveur** va commencer par **identifier son interlocuteur**. Ainsi, s'il ne le connaît pas, il va lui demander son identifiant et sa fonction. Le **Terminal** doit répondre forcément à cette demande par son identifiant et sa fonction de **Terminal** (**TYPE-Utilisateurs**). Une fois identifié, il va vérifier que celui-ci a la permission d'ordonner cette requête (**Permissions**). Enfin, s'il est autorisé, le Serveur va procéder à la **lecture de la requête**. Il va tout d'abord **déchiffrer** le **Protocole-2** à l'aide de la clé de chiffrement RSA accessible grâce à l'identifiant contenu dans le **Protocole-3**, puis va lire la demande. Si les conditions sont correctes, à savoir si le montant proposé n'est pas supérieur à la valeur actuelle de l'utilisateur et que le débit se déroule sans erreurs, le **Serveur** renvoie au **Terminal** un nouveau message sur le modèle du **Protocole-2** contenant la réponse : **Autorisé** ou **refusé**.
8) Le **Serveur** envoie au **Terminal** le message
9) Le **Terminal** reçoit le résultat : il change son affichage et **effectue ou non la demande** (par exemple : indiquer qu'un ticket de bus a été débité ou qu'une transaction bancaire a été effectuée, ouvrir une barrière dans le métro ou au péage, etc.) dépendant de la fonction du **Terminal** et de la réponse reçue.

2) Emission des données binaires à transmettre vers le corps

Schéma de principe



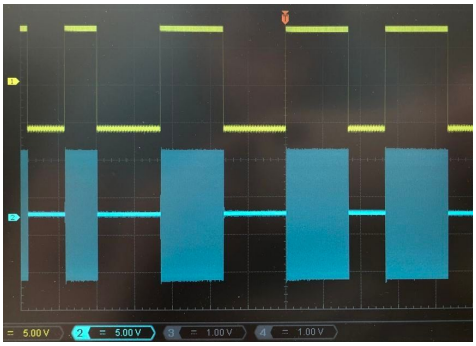
Fonctionnement



On a créé un circuit muni d'un **oscillateur** (type colpitts) qui fonctionne sur le principe des oscillations d'un circuit LC (bobine + condensateur), qui nous permet d'obtenir notre signal **porteur** à une fréquence de **455 kHz**.

$$L = 6{,}8\ \mu H \text{ et } C = 18\ nF \rightarrow f_0 = \frac{1}{2\pi\sqrt{LC}} = 455\ kHz$$

L'ordinateur **1** envoie un signal (qui contient les données à transmettre) vers l'oscillateur à 455 kHz qui contient aussi un transistor **mosfet**, ce dernier va jouer le rôle d'**interrupteur**.



Explications : le signal transmis va alterner tensions négatives et tensions positives en fonction des bits envoyés (les 0 et les 1), lorsque la tension est **positive**, le mosfet se comporte comme un **interrupteur fermé** et le circuit **oscille**, et lorsque la tension est **négative**, le mosfet se comporte comme un **interrupteur ouvert** et l'oscillateur **n'oscille pas**. On a ainsi modulé l'amplitude de la porteuse par le signal à transmettre, en **tout ou rien**. On distingue parfaitement les 0 et les 1 du signal binaire. On a ainsi fait de la **modulation numérique** de type **ASK** (**A**mplitude-**S**hift **K**eying en anglais). Le signal modulé est ensuite **amplifié** et envoyé vers l'**antenne** et le **corps humain**.

Point de vue du développeur

Le schéma protocolaire correspondant à l'émission par le **Terminal** est le suivant :
ORDRE:TOKEN_VALIDITE:ADDRESS_BLUETOOTH:TARIF#

En pratique, le protocole peut ressembler à cet EXEMPLE : **M:agft:04939582:1#**

Dans ce protocole, nous avons donc :

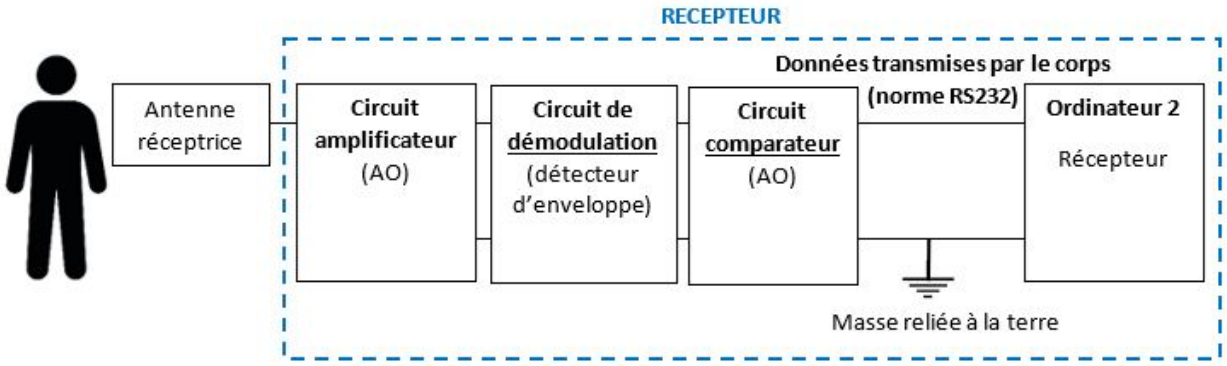
- **ORDRE** : l'ordre qui est le type d'action à effectuer (ouvrir une porte d'accès, valider un titre de transport, etc.) (représenté par la lettre **M** dans notre exemple)
- **TOKEN_VALIDITE** : une chaîne de 4 caractères aléatoires qui est renouvelée à chaque transaction. Cela permet d'éviter les doublons indésirables (**agft** dans notre exemple)
- **ADDRESS_BLUETOOTH** du **Terminal** : nous partons du principe que ce dispositif peut fonctionner avec plusieurs **terminaux**. Cela implique de devoir **différencier** les terminaux entre eux. Pour cela, nous communiquons dans le message l'adresse à joindre pour répondre. (**04939582** dans notre exemple)
- **TARIF** : le tarif qui doit être retiré sur le compte d'utilisateur (1 ticket, 10€, etc.) (**1** dans notre exemple)
- **#** : caractère unique signifiant la **fin du protocole**, donc la fin du message.

Nous avons fait le choix d'avoir un système **adaptable à chaque situation**. Cela permet par exemple de **différencier** un compostage de tickets de bus et une transaction bancaire.

Une fois le protocole créé, nous l'envoyons **en boucle** jusqu'à obtenir une réponse d'un récepteur.

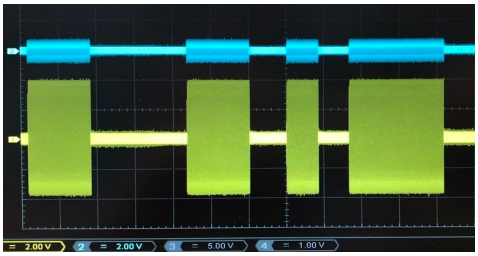
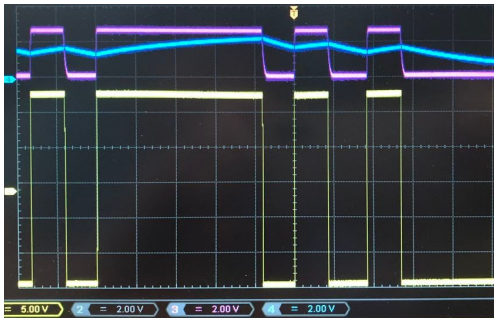
3) Réception des données binaires transmises par le corps

Schéma de principe



Fonctionnement

Après le passage dans le corps humain les données sont **captées** par l'**antenne réceptrice** puis **amplifiées**. Elles sont ensuite **démodulées** par un circuit détecteur d'enveloppe (qui fonctionne sur le principe de la charge et de la décharge d'un condensateur), puis un circuit **comparateur** permet de reconstituer un signal à la **norme série RS232** (les tensions positives doivent être comprises entre +3V et +25V et les tensions négatives entre -3V et -25V) avant d'être envoyées vers l'ordinateur **2** qui recevra les données.

Le signal reçu par l'antenne réceptrice (en bleu) est amplifié (signal jaune)	Le signal jaune précédent est démodulé (on obtient le signal violet) puis le signal binaire est reconstitué (en jaune) à partir de la comparaison du signal violet avec le signal bleu (circuit comparateur)
	

Note : le circuit détecteur d'enveloppe qui sert à démoduler sera remplacé par la suite par un circuit plus sensible.

Point de vue du développeur

Le programme lit les caractères arrivant un à un. Comme le programme de réception connaît le **protocole** utilisé pour l'émission, dès lors qu'il lit un "#", il commence à enregistrer car cela signifie que le message précédent vient de se terminer. Un nouveau message complet arrive. Ensuite, dès qu'il retrouve un "#", il est sûr de posséder la requête complète.

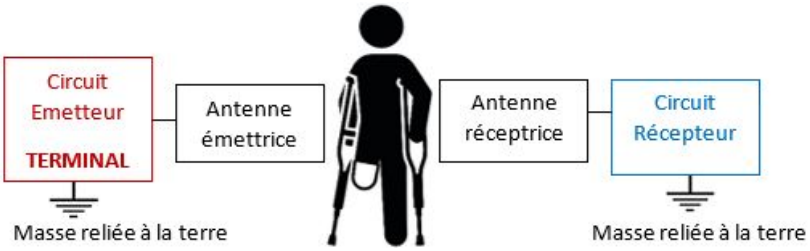
Lorsqu'on connaît la structure d'un protocole, il est très simple de retrouver les informations cachées dedans. Une fois les données analysées, l'ordre peut être exécuté.

*Remarque : les circuits électroniques que nous avons créés sont détaillés en **Annexe 4**. Ils ont été conçus avec le site web gratuit EasyEDA.com*

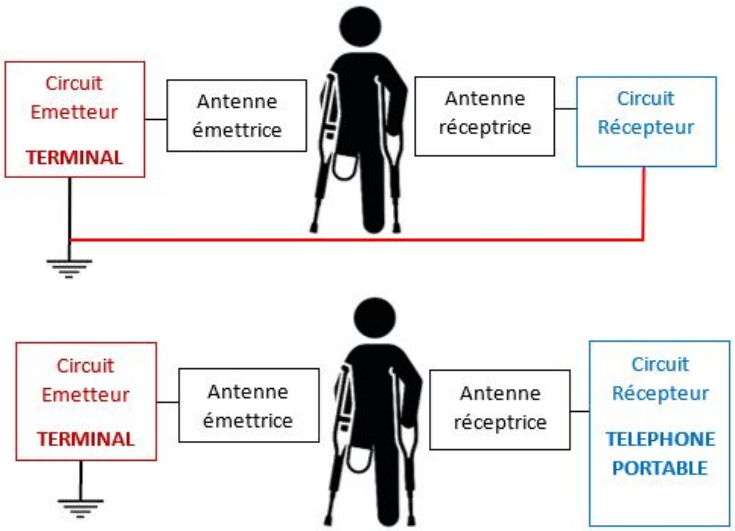
4) Problème : un câble en moins !

Le dispositif présenté précédemment pose une **difficulté** : si nous rendons le **récepteur autonome** en l'alimentant intégralement par **batterie** (pour simuler un téléphone portable) alors notre dispositif **ne fonctionne plus.**

Explications : dans notre dispositif précédent il existe toujours un **lien physique électrique** entre l'émetteur et le récepteur par le biais de la **prise de terre** des appareils électriques. En effet les deux circuits ont leur masse reliée à la prise de terre de façon à avoir une **référence commune** pour la mesure des tensions.



Cela permet de **fermer le circuit électrique**, l'ensemble formant un **circuit série** (c'est comme s'il y avait le câble en rouge ci-contre).



Si on déconnecte les circuits de la prise de terre alors le dispositif **ne fonctionne plus**. Or si on veut faire fonctionner le dispositif avec un récepteur **autonome** comme un **téléphone portable** (situé dans la poche) suivant le schéma ci-contre alors il n'y a plus de connexion possible entre le circuit récepteur et la terre. **Le câble (en rouge) disparaît** ! Il n'existe pas avec un téléphone portable.

Solution trouvée : la seule possibilité est de passer par les **ondes électromagnétiques** (OEM). On sait depuis longtemps que le corps humain a un comportement d'**antenne réceptrice** (il suffit de mettre la main sur une antenne de radio pour constater qu'on capte mieux) mais pour que notre dispositif fonctionne il faut également que le corps humain ait un comportement d'**antenne émettrice**. Or c'est ce que nous avons découvert à travers la **thèse de Kibret** (2016) **« The Human Body Antenna : Characteristics and its Application »** où l'homme est modélisé par une antenne droite cylindrique.

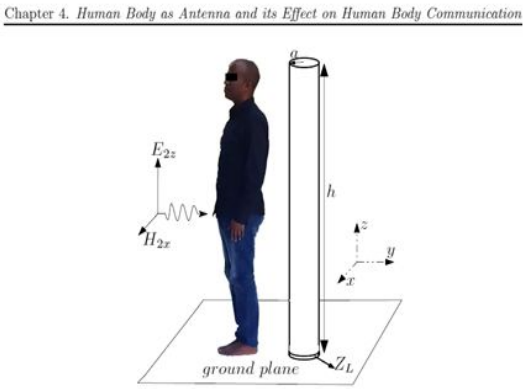
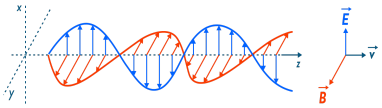
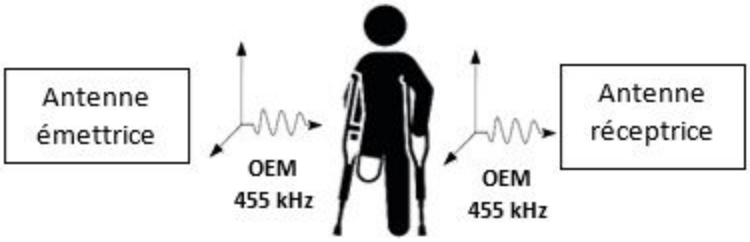


FIGURE 4.1: Cylindrical antenna model of the human body exposed to vertically polarised plane wave [53].

Notre idée est donc la suivante : le corps humain pourrait **recevoir une OEM** provenant de l'antenne émettrice du **Terminal** puis **réémettre une OEM** vers une **antenne réceptrice** que nous ajouterons dans le téléphone portable.



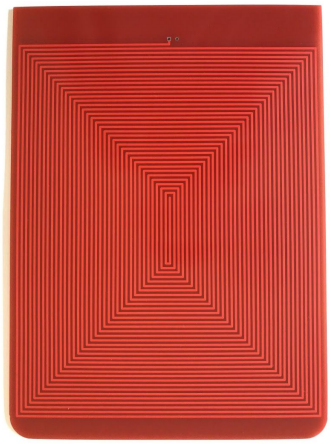


L'OEM émise par l'antenne émettrice doit être de **faible puissance** pour se propager sur une **courte distance** pour ne pas atteindre l'antenne réceptrice en l'absence du corps humain. En quelque sorte le corps humain servirait de **prolongement de l'antenne émettrice** pour que le signal atteigne l'antenne réceptrice.

5) Conception d'antennes émettrices et réceptrices

Pour la conception de nos antennes émettrices et réceptrices, il a fallu revoir l'idée d'utiliser une simple plaque de cuivre. On a eu besoin de s'appuyer sur **"la théorie des antennes filaires"**, qui dit que pour que l'antenne soit le plus efficace possible, l'ordre de grandeur de sa taille doit être égal à $\frac{\lambda}{2}$ (antenne dite demi-onde), avec $\lambda = \frac{c}{f}$ (c étant la vitesse des OEM dans le vide en m/s et f la fréquence des ondes en Hz). Ce type d'antenne est sensible au **champ électrique** de l'OEM. Ainsi, lorsqu'on calcule la valeur de λ à une fréquence de 455 kHz on obtient : $\lambda = \frac{3{,}00.10^8}{455.10^3} = 660\ m$ et donc $\frac{\lambda}{2}$ **= 330 m** ; voilà la longueur idéale pour émettre et recevoir au mieux les ondes, et c'est très long ! Bien évidemment, ce résultat est inexploitable pour notre dispositif que nous souhaitons garder à une **échelle raisonnable** (l'antenne réceptrice doit tenir dans un téléphone portable). De plus, rappelons que l'OEM émise ne doit pas atteindre l'antenne réceptrice en l'**absence du corps humain** donc on a pas besoin d'avoir un dispositif qui émet loin. On veut communiquer sur des courtes distances, comme pour la technologie NFC.

Par conséquent, on a décidé d'un **compromis** : on doit concevoir des antennes avec la plus grande longueur possible tout en restant en accord avec les dimensions qui conviennent au projet : donc une **antenne émettrice** de taille suffisante pour poser une partie du corps (choix de 20cm × 15cm) et une **antenne réceptrice** qui entre dans une coque de Smartphone (choix de 6cm × 6cm) à côté du circuit de réception. On a donc **enroulé le fil de l'antenne sur lui-même**. On a calculé la longueur total du fil des antennes :



- Grande antenne : environ **14 m**
- Petite antenne : environ **1,6 m**

Notre IHM évolue, c'est maintenant :

IHM = {Grande antenne filaire - Isolant - Corps humain - Isolant - Petite antenne filaire}

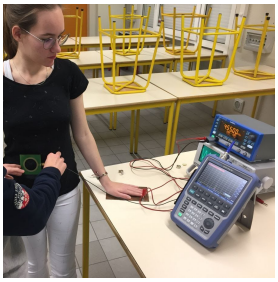
III- Le corps humain se comporte comme une antenne rayonnante

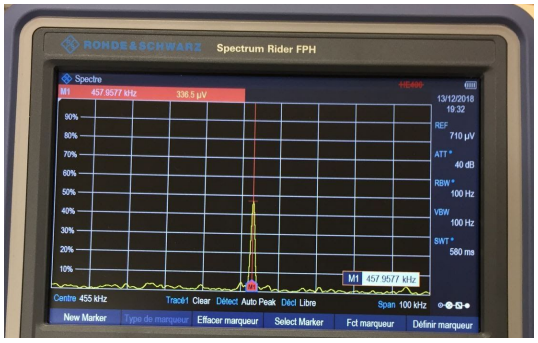
1) Mise en évidence du rayonnement électromagnétique du corps humain

Nous avons d'abord souhaité vérifier que le **corps humain** se comporte bien comme une **antenne rayonnante**, c'est-à-dire qu'il émet naturellement un rayonnement lorsqu'il en reçoit un.

Pour cela l'ingénieur d'Orange nous a prêté un analyseur de spectre portable et autonome (sur batterie), donc **sans aucun lien physique** avec le reste du circuit afin que seul le **rayonnement** soit mesuré.

<u>Expérience</u> : une **main est posée sur l'antenne émettrice** reliée au générateur de tension sinusoïdale de fréquence **455 kHz** et d'amplitude 10 V. Les vêtements (tee-shirt et pantalon) ont été quadrillés (8 cm x 8 cm) et numérotés et une antenne réceptrice est posée dans chaque zone. L'analyseur de spectre fait apparaître **un pic** à **455 kHz** ce qui confirme que **le corps émet un rayonnement** de même fréquence que ce qu'il reçoit. On mesure son amplitude, les valeurs obtenues sont faibles, de l'ordre de **100 µV**.





	250				260	
255	240	230	360	280	400	400
	440	300	230	200	170	
	300	300	180	320	320	
	210	270	280	130	125	
	180	280	200	180	130	
	160	200	180	200	130	
	140	130	120	150	160	

Pour présenter les résultats nous avons ensuite normalisé les valeurs obtenues en divisant toutes les tensions mesurées par la tension la plus faible. Les zones de valeur 1,0 sont donc celles qui émettent le moins (en bleu), et celles de valeur 2,0 ou 3,0 émettent deux ou trois fois plus (en rouge).

Résultats normalisés obtenus sur Lory :

Lory tee-shirt devant

	1,7				2,0	
1,7	2,3	2,0	1,7	1,7	1,7	2,0
	2,6	1,9	1,7	1,9	2,9	
	1,4	1,7	1,7	1,7	1,9	
	1,9	2,0	2,0	2,0	1,7	
	1,6	1,7	1,9	1,9	1,4	
	1,1	1,1	1,3	1,4	1,4	
	1,7	1,3	1,4	1,4	1,4	

Lory tee-shirt derrière

	1,7	1,6		2,0	1,7	
2,1	2,1	1,7	1,9	1,9	2,0	2,3
	1,9	1,9	2,0	1,9	2,0	
	1,9	1,8	1,8	1,7	1,7	
	1,9	1,6	1,4	1,6	1,6	
	1,3	1,4	1,3	1,4	1,1	
	1,4	1,7	1,4	1,7	1,4	
	1,6	1,3	1,1	1,3	1,4	

Lory pantalon devant

	1,4	1,3	1,1	1,7	1,9	
	1,7	1,7	1,7	1,9	2,0	
	1,9	1,9		1,9	1,9	
	2,1	2,1		2,3	1,9	
	1,7	1,7		2,0	2,0	
	1,9	1,7		1,7	1,6	
	1,6	1,7		1,9	1,6	
	1,6	1,6		1,6	1,4	
	1,4	1,4		1,6	1,4	
	1,3	1,3		1,3	1,4	
	1,3	1,0		1,1	1,1	

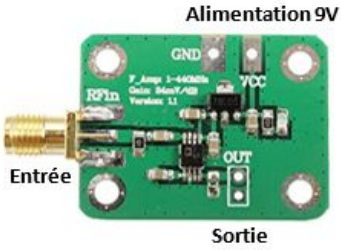
Lory pantalon derrière

	1,4	1,7	1,3	1,4	1,3	
	1,6	1,7	1,6	1,6	1,3	
	1,7	1,6	1,9	1,4	1,4	
	1,9	2,3		2,0	1,4	
	1,9	1,7		1,7	1,3	
	1,7	1,6		1,7	1,4	
	1,6	1,3		1,6	1,7	
	2,0	1,3		1,6	1,4	
	1,4				1,4	
	1,6				1,4	
	1,4				1,4	
	1,3				1,1	

Nous constatons que certaines zones ont tendance à moins rayonner : le bas du ventre, le bassin et le bas des jambes mais que l'ensemble reste **assez homogène** malgré tout.

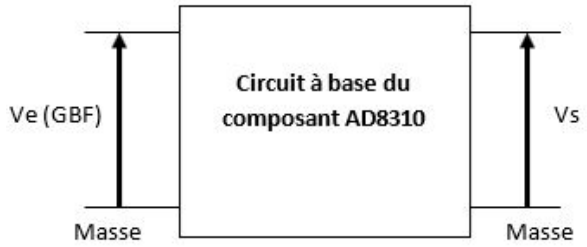
<u>Conclusion</u> **:** le corps humain se comporte bien comme une **antenne** qui émet **naturellement** une partie du rayonnement qu'il reçoit, et à la même fréquence. L'idée est donc d'**utiliser ce rayonnement pour transmettre des données binaires**. Cependant les tensions captées sont faibles, il faut mettre en place un **nouveau capteur plus sensible** pour remplacer le circuit détecteur d'enveloppe .

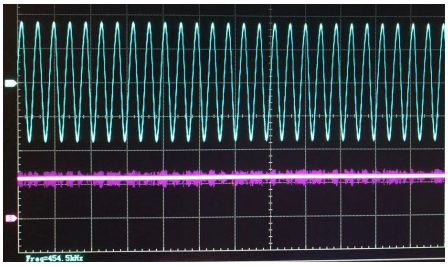
2) Comment capter un faible rayonnement ?



D'après Orange, afin de capter un faible rayonnement, on peut utiliser un **amplificateur logarithmique** comme le composant **AD8310** utilisé dans le domaine des radiofréquences. On a acheté un circuit à base de ce composant (ci-contre) et, pour comprendre son fonctionnement, nous avons tracé la tension de sortie en fonction de la tension d'entrée. Nous avons donc réalisé le montage suivant :

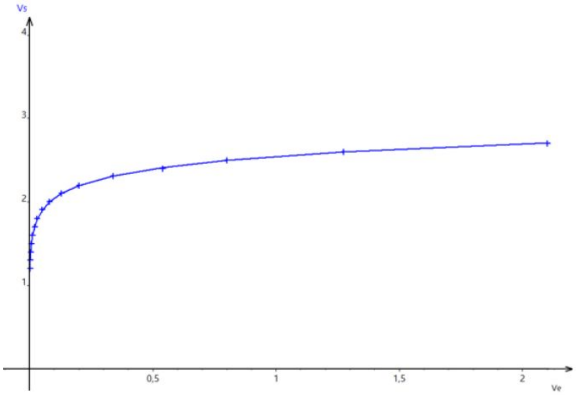
En **entrée** nous avons une **tension sinusoïdale** de **455 kHz** délivrée par un GBF, d'amplitude **Ve**. Puis en sortie nous avons une **tension continue** notée **Vs** mesurée au voltmètre. Enfin le circuit doit être alimenté en 9V pour fonctionner.



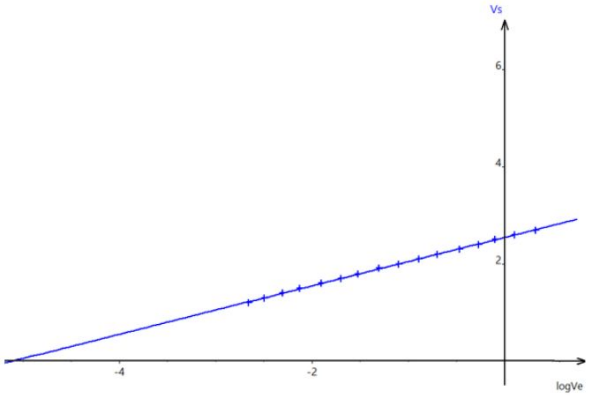


<u>**Expérience**</u> : nous avons fait varier **Ve de 0,002 V à 5 V** et mesuré Vs (qui varie de 1,15 V à 2,79 V), puis nous avons tracé Vs=f(Ve) et Vs=f(log(Ve)). Voici les courbes obtenues :

Vs = f(Ve)



Vs = f(log(Ve))



On constate que **Vs=f(log(Ve))** est une fonction **affine**, c'est pour cette raison qu'on appelle le composant **amplificateur logarithmique**. Nous avons modélisé la droite pour obtenir l'équation suivante :

Vs= a × log(Ve)+b avec a = 498 ± 5 mV et b = 2,545 ± 0,008 V

Ce circuit peut donner une tension de sortie entre **0,3 V** et **2,8 V**.

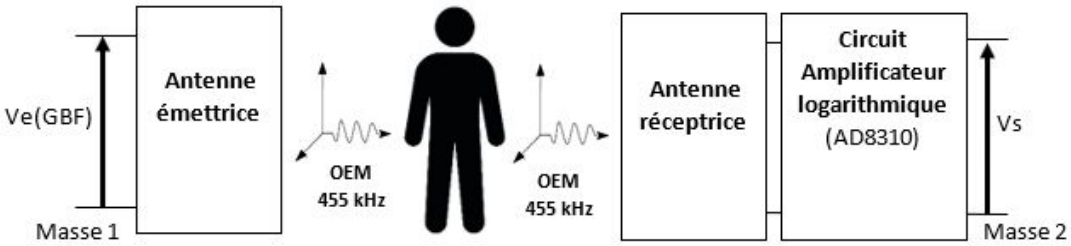
Pour Vs(min) = **0,3 V** : log(Ve) = $\frac{0,3-b}{a}$ = - 4,5 ; Ve = $10^{-4,5}$ V = **3×10^{-5} V = 30 µV**

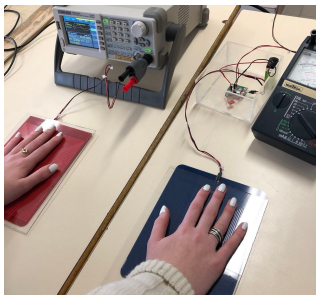
Pour Vs(max) = **2,8 V** : Ve = **5 V**

Nous remarquons qu'entre les deux valeurs de Ve, il y a un rapport supérieur à **100 000** ! L'intérêt de l'amplificateur logarithmique est donc d'être capable de détecter une **très grande gamme** de rayonnement en entrée, et donc de pouvoir aussi **capter de faibles niveaux de rayonnement**. Ce qui est très important car le niveau capté peut varier d'une personne à l'autre, changer en fonction de l'épaisseur des vêtements ou bien si le récepteur se trouve dans un sac au lieu de la poche. Nous espérons ainsi pouvoir **utiliser le rayonnement émis par le corps humain**.

3) Test de réception du rayonnement

Afin de **mettre en évidence** les ondes électromagnétiques rayonnées par le **corps humain** nous avons utilisé le dispositif suivant en utilisant l'**amplificateur logarithmique** :



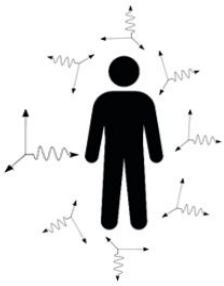


<u>Expérience</u> : Ve est une tension sinusoïdale à **455 kHz** et d'amplitude **10 V.** Vs est mesurée au **voltmètre**. Toute la partie réception est **autonome** (sur batterie donc les deux masses du circuit sont indépendantes). Une **main** est posée sur l'**antenne réceptrice** (droite) : on mesure une tension de **1 V** environ, puis on pose l'**autre main** sur l'**antenne émettrice** (gauche) : on mesure une tension de **1,5 V** environ : l'antenne réceptrice capte bien un **rayonnement supplémentaire émis par le corps humain** !

Remarquons que sans le corps humain la tension est de **1 V et non de 0,3 V** car nous baignons en permanence dans les OEM émises par toutes les radios, la TV, les téléphones portables, et les appareils électriques, et l'antenne réceptrice capte une partie de ce rayonnement.

<u>Conclusion</u> : nous allons pouvoir utiliser le composant amplificateur logarithmique AD8310 très sensible pour capter le rayonnement et démoduler le signal en remplacement du circuit détecteur d'enveloppe utilisé précédemment.

Au final, voici ce que nous pouvons conclure sur le **comportement électrique du corps humain** :

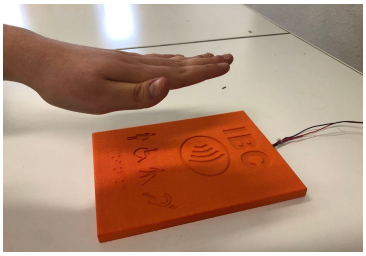


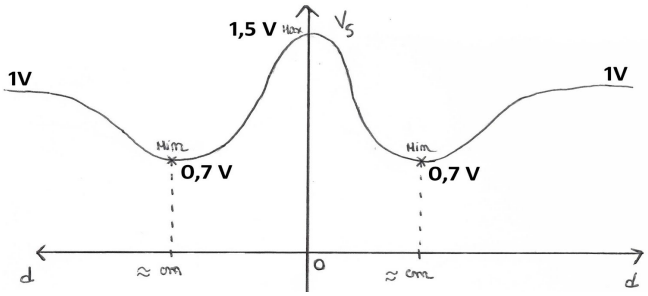
- Il a un caractère **résistif** en courant continu
- Il a un caractère **résistif et capacitif** en courant alternatif
- Il a un caractère **rayonnant** : il reçoit et émet des OEM

4) “Voir” les OEM : “Les cornes du Diable” et la boîte magique

Les cornes du Diable

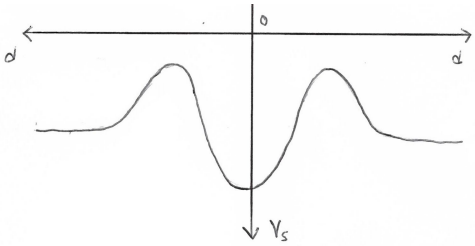
Lors de l’étude précédente nous avons pu observer un **phénomène “étrange”** : une **perturbation** des ondes reçues ! En effet, lors des mesures de la tension **Vs**, quand l’utilisateur approchait sa main de la plaque, **la tension diminuait** (vers 0,7 V) jusqu’à une certaine distance de la plaque (quelques cm) **pour ensuite ré-augmenter** jusqu’à atteindre un maximum (vers 1,5 V). Le même phénomène se produit aussi quand on **retire la main** : la tension diminue puis ré-augmente. Nous avons représenté ci-dessous l’allure du signal **Vs(d)** en appelant **d** la **distance entre la main et l’antenne émettrice (recouverte de sa coque de protection ci-dessous)**.





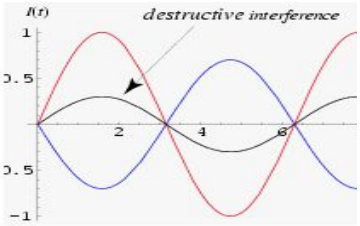
Ce phénomène était totalement **inattendu** puisqu’on pensait que la tension ne ferait qu’augmenter en approchant la main ! **Comment se fait-il que la tension diminue ?** Puisque le corps humain rayonne, plus la distance est faible avec l’antenne et plus le rayonnement capté devrait être important !

Après s’être renseigné auprès de l’ingénieur Orange, il s’est avéré que ce phénomène est connu des chercheurs et qu’on le nomme **phénomène des cornes du Diable** (il faut avoir un peu d’**imagination** mais en mettant le graphique précédent à l’envers il se dessine deux **“cornes”**, de plus ce phénomène est indésirable d’où le terme **“Diable”**).



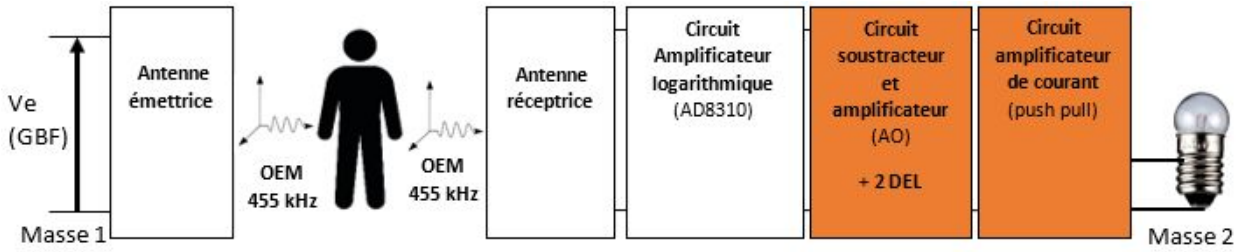
L’origine du phénomène est **méconnue**, d’autant plus qu’**il n’apparaît pas tout le temps** ! Nous avons effectué plusieurs tests dans des conditions similaires (même pièce et même environnement), le phénomène était **présent certains jours et d’autres non**. En changeant de pièce il pouvait **disparaître ou apparaître** ! De plus, la distance à laquelle se produit le changement de sens de la tension est variable également.

Certaines **hypothèses** peuvent être proposées : il est possible que des ondes “parasites” (obtenues par réflexion sur des objets par exemple) se superposent aux ondes transmises par le corps humain et créent alors une zone d’**interférence destructive** où le signal est atténué.

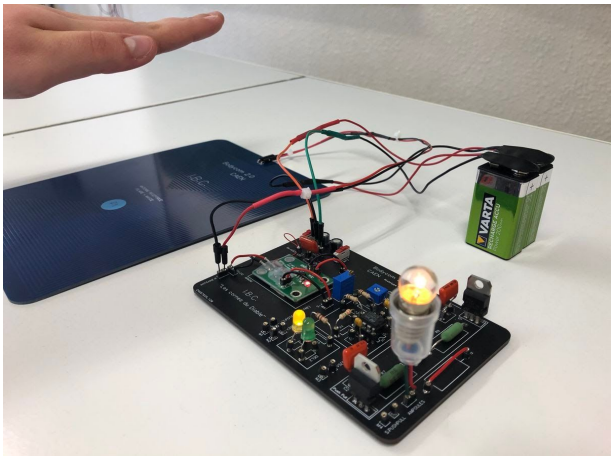


Nous avons décidé de **mettre en évidence cette étrange manifestation** en créant un circuit qui permettrait de **visualiser la baisse puis l'augmentation de la tension**.

Schéma de principe



Fonctionnement



La partie nouvelle est la partie en orange ci-dessus. Comme la tension de départ fournie par l'**AD8310** est de **1 V** environ, nous avons décidé de **soustraire 1 V** pour arriver à **zéro** grâce à un montage soustracteur. Quand on approche la main de l'antenne, si le phénomène des cornes du Diable est présent alors la tension diminue donc la tension à la sortie du soustracteur va devenir **négative**. Pour que cela soit **visuel** on a intégré **deux DEL** : une première DEL d'une certaine couleur s'allume quand la tension est négative et une seconde DEL d'une autre couleur s'allume quand la tension est positive. Ainsi la couleur perçue varie selon le signe de la tension et le phénomène est facilement observable. Nous avons aussi intégré un amplificateur de courant de façon à allumer ou éteindre une **ampoule à filament**.

Notons que le phénomène des "cornes du Diable" **n'est pas gênant pour notre dispositif final** puisque la tension obtenue quand on met la main très proche de l'antenne est supérieure à la tension loin de l'antenne.

Notre circuit permet également de "VOIR" la transmission des 0 et des 1 binaires par allumage et extinction de l'ampoule au rythme de la transmission des données (en choisissant une fréquence des données de quelques Hertz pour que l'ampoule clignote lentement).

La boîte magique

Nous avons ensuite voulu créer un dispositif encore plus spectaculaire, à l'aide d'une **énorme ampoule à filament** de 30 cm de haut, alimentée en 220 V. L'aide de la société Orange nous a permis de réaliser ce dispositif de toute pièce que nous avons appelé "la boîte magique".

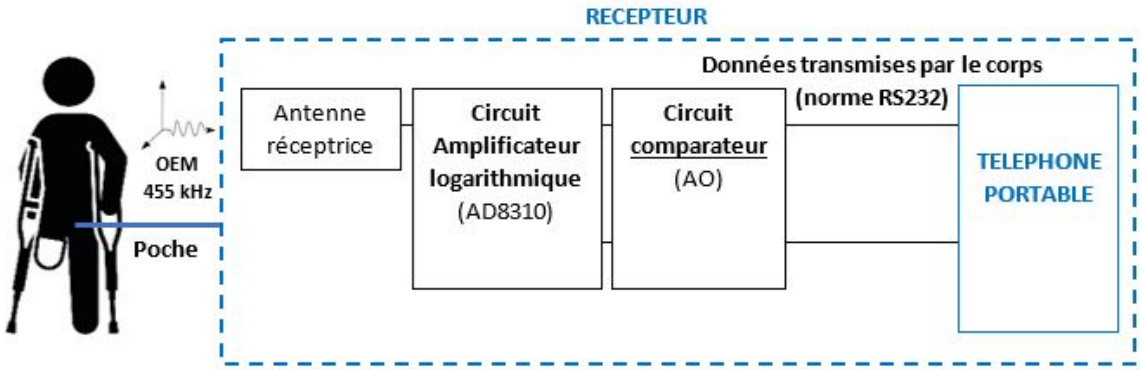
Nous n'allons pas vous dévoiler ses secrets tout de suite mais sachez que notre boîte magique peut émettre de la lumière mais également du son.

Si vous souhaitez dès maintenant la voir en action, rendez-vous sur notre site **https://www.ibc.ovh**

IV - Utiliser les OEM rayonnées par le corps humain : le dispositif final

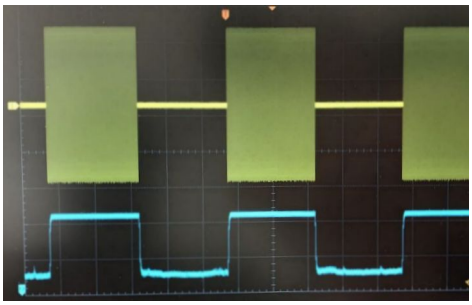
1) Le circuit de réception intégré au téléphone portable

Schéma de principe



Fonctionnement

Les ondes émises par le corps humain sont réceptionnées par l'**antenne réceptrice** qui reçoit donc un signal à **455 kHz modulé en amplitude** qui correspondant aux **0** et **1**. Quand un 0 arrive, l'amplificateur logarithmique donne une tension d'environ **1 V** et quand un 1 arrive il délivre une tension d'environ **1,5 V**, ainsi on peut **distinguer les 0 et 1**. Le circuit **comparateur** permet ensuite de reconstituer un signal à la **norme série RS232** avant d'être envoyé vers le **téléphone portable** qui recevra les données et les analysera.



2) L'aspect logiciel et application mobile

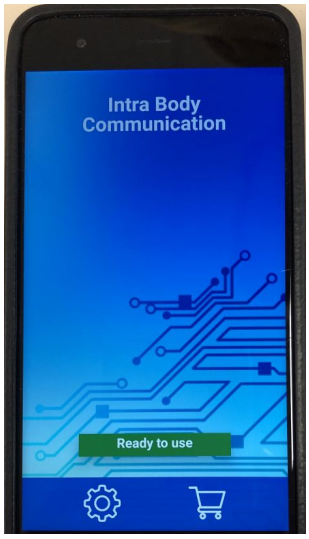
Toute la partie logiciel (Terminal et Serveur) et application mobile Android a été développée en langage **Java** et comporte environ 16000 lignes de code qui permettent de gérer le fonctionnement des **trois éléments principaux** :

le **Terminal,** le **Serveur,** le **Smartphone**

Communications mobiles

L'**application Android** est capable de lire des données par Série en lisant les entrées dans le port USB du téléphone en provenance du circuit électronique et de son antenne réceptrice.

Ergonomie de l'interface mobile



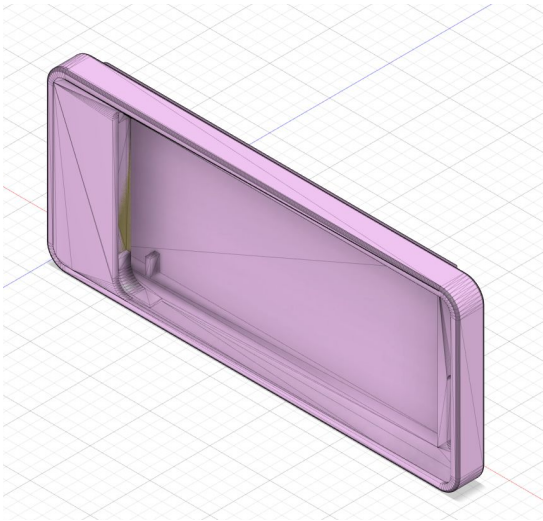
L'application **Android** possède deux interfaces : une première qui sert de menu, et une seconde qui permet de visionner et d'augmenter son stock de tickets.

Ergonomie de l'interface du Terminal

Pour le côté démonstratif nous avons choisi d'afficher les informations à chaque **étape du processus.** Des informations **vocales ont été ajoutées** pour guider les personnes **malvoyantes**. Les malvoyants sont donc capables d'utiliser ce service.

3) Les maquettes 3D de démonstration

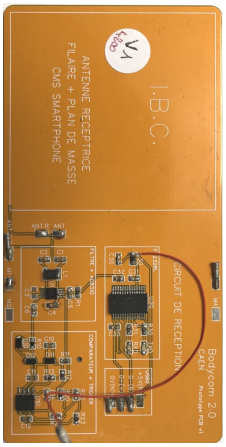
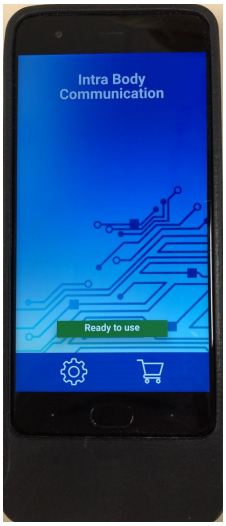
La coque du Smartphone



Nous avions besoin d'une **coque spéciale** pour contenir un vrai Smartphone, le circuit de réception et son antenne réceptrice, ainsi que le connecteur pour relier le port USB-C du Smartphone au circuit récepteur. Plusieurs coques de tests ont été réalisées, elles ont été conçues avec freeCAD, un logiciel de modélisation 3D open source.

Cahier des charges de la coque :

- adaptée à la forme du Smartphone offert par NXP (modèle Xiaomi 6)
- plus profonde qu'une coque usuelle, pour y placer le circuit électronique de réception
- munie d'une extension dans la partie inférieure pour cacher le connecteur USB-C
- légère, souple, et robuste
- des bords arrondis pour être agréable à tenir

Le circuit de réception final et son antenne réceptrice (recto/verso)		La coque modélisée et imprimée en 3D	Le circuit au fond de la coque	Le Smartphone dans la coque au dessus du circuit
				

La barrière ouvrante

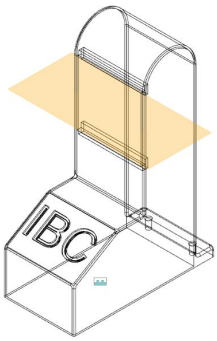
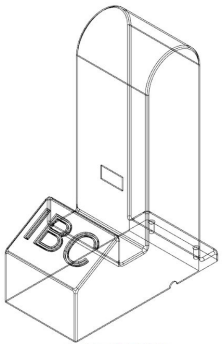
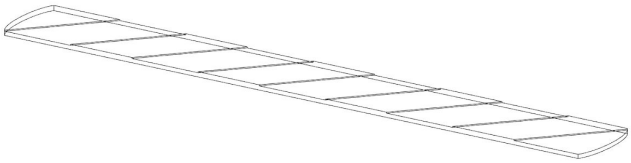
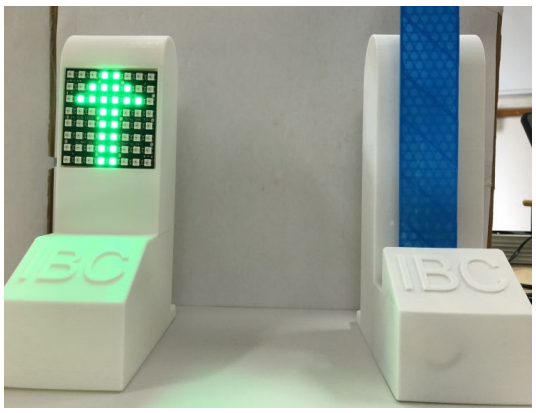
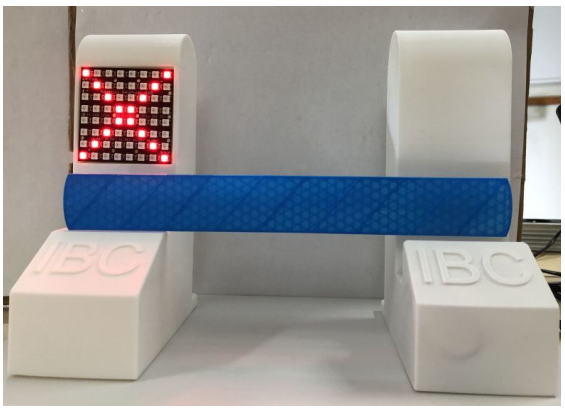
L'idée est d'obtenir une **action concrète** observable lorsqu'une transaction est validée par notre dispositif IBC (validation d'un paiement CB, d'un ticket de bus ou de métro, d'un péage autoroutier, etc.). Pour cela nous avons choisi d'ouvrir ou de fermer une barrière pilotée par un servomoteur relié à un microcontrôleur Arduino lui-même piloté par le Terminal de notre dispositif final :

IBC -> Terminal -> Arduino -> Servomoteur -> Ouverture ou fermeture de la barrière

Cahier des charge de la barrière :

- poteaux de hauteur 20 cm pour être bien visibles, 30 cm pour la barrière elle-même
- encoche dans le poteau de droite pour permettre la rotation de la barrière
- emplacement pour le panneau de LED dans le poteau de gauche
- poteaux creux pour contenir et cacher le microcontrôleur et les fils électriques

Les poteaux et la barrière ont été conçus sur Fusion 360, un autre logiciel de modélisation 3D que nous avons trouvé plus pratique à l'usage, il n'est pas open source mais propose néanmoins une version d'essai gratuite de 1 an.

Poteau de gauche	Poteau de droite	Barrière
		
		

Conclusion

Nous sommes très satisfaits d'être parvenus à réaliser un dispositif fonctionnel qui utilise le rayonnement du corps humain pour communiquer en transmettant des données binaires ; d'autant plus qu'il serait utile à tous et faciliterait une partie de la vie des personnes en situation de handicap. De plus, notre dispositif utilise les communications réseau comme le fait un vrai Terminal NFC sans contact pour joindre un Serveur et nous sommes parvenus à miniaturiser le récepteur pour qu'il puisse entrer dans une coque de Smartphone. Nous allons compléter notre site web pour présenter d'autres vidéos de démonstrations, et nous souhaitons mettre à disposition nos circuits électroniques, et code Java en open source pour que d'autres projets se développent en utilisant le même phénomène physique du rayonnement du corps humain. Beaucoup d'applications sont à imaginer.

MERCI

Merci à l'équipe "Bodycom 2.0" de 2019/2020 (seuls 2 membres sont encore présents cette année, rejoints par Marin)

L'équipe "Bodycom 2.0" (2019/2020)

- Cynthia KIMPAMBOUDI
- Nahomé VESVARD
- Clément DESJONQUERES
- Lory AUDRAIN

Avec la participation d'Octave PLESSIS (vidéo) et d'Alis BOSSUYT (design)

ANNEXES

Note : consultez également notre site web **https://www.ibc.ovh** (ibc.ovh) pour avoir plus de détails sur les circuits électroniques, les maquettes 3D, et le code source Java

Annexe 1 : Sources

- Thèse de Kibret (2016) The Human Body Antenna : Characteristics and its Application : http://vuir.vu.edu.au/31012/
- Thèse Spectroscopie diélectrique hyperfréquence de cellules uniques cancéreuses: de l'optimisation du capteur en sensibilité : https://tel.archives-ouvertes.fr/tel-01561608/document
- Thèse de Zimmerman "Personal Area Networks (PAN) : Near-Field Intra-Body Communication" : https://dspace.mit.edu/handle/1721.1/29101
- NFC : https://fr.wikipedia.org/wiki/Communication_en_champ_proche
- DAS (Débit d'Absorption Spécifique) du NFC : https://books.google.fr/books?id=FC7fCwAAQBAJ&pg=SA2-PA24&lpg=SA2-PA24&dq=sar+nfc+13.56&source=bl&ots=l0E3HSJbu4&sig=G4ke5cwAclaP79zeiMnig8wdu3w&hl=fr&sa=X&ved=0ahUKEwiShcPrkOzZAhWIJcAKHYH6DU8Q6AEIPjAC#v=onepage&q=sar%20nfc%2013.56&f=false
- Principes d'électroniques de Albert Paul Malvino (montage oscillateur Colpitts p681)
- http://www.elektronique.fr/cours/composant-condensateur.php
- http://www.elektronique.fr/cours/composant-bobine.php
- https://fr.wikipedia.org/wiki/Circuit_LC
- http://www.elektronique.fr/cours/AOP/aop-montages-de-base.php
- http://electronique.aop.free.fr/AOP_lineaire_NF/3_ampli_inverseur.html
- http://electronique.aop.free.fr/AOP_sature/1_comparateur_simple.html
- http://electronique.aop.free.fr/AOP_lineaire_NF/5_amplidiff_1.html
- Conception de circuits électroniques : https://easyeda.com

Annexe 2 : Protocoles informatiques et schéma de données

Présentation

Pour transmettre nos données, nous avons dû créer une série de protocoles répondant au mieux à nos attentes. Ainsi, chaque protocole sera présenté dans la partie *Liste des Protocoles* de la sorte :

Nom de référence du protocole (utilisé dans ce compte-rendu) :

➔ Schéma : **PARTIE_1**:**PARTIE_2**:**PARTIE_3**

- Caractère(s) de séparation : ":" *Énonciation du ou des caractères de séparation*
- Parties :
  - **PARTIE_1** : *Description de la partie*
  - **PARTIE_2** : *Description de la partie*
  - **PARTIE_3** : *Description de la partie*

Ensuite, il nous a fallu trouver et organiser les données que nous allions utiliser et stocker. Dans la partie ***Organisation des données***, vous retrouverez chaque donnée sous cette forme :

Nom de référence de la donnée (utilisée dans ce compte-rendu) :

➔ Emplacement de stockage : *(Serveur, Terminal, Téléphone Client)*
➔ Définition de la donnée : *L'utilité de la donnée*
➔ Sensibilité de la donnée à la sécurité : *(Normal, Sensible, Très sensible)*
➔ Fréquence de transit : *(À chaque utilisation, une seule fois, sur demande)*
➔ Propre à : (*Serveur, Terminal, Téléphone Client)*

Enfin, pour augmenter la sécurité, nous avons mis au point une **gestion des permissions**. Chaque action demandée au Serveur implique une vérification des permissions. Dans la partie Permissions, elles seront énumérées sous cette forme :

Nom de la Permission :

- *Liste des types d'utilisateurs ayant cette permission*

Liste des protocoles

Protocole-1 : Schéma : **ORDRE:TOKEN_VALIDITE:ADRESSE_BLUETOOTH:TARIF#**

- Caractère de séparation :
  - ":" Sépare chaque parties
  - "#" Termine le protocole
- Parties :
  - **ORDRE** : l'ordre à effectuer
  - **TOKEN_VALIDITE** : le code aléatoire généré à chaque utilisation
  - **ADRESSE_BLUETOOTH** : l'adresse bluetooth du Terminal
  - **TARIF** : Est le tarif à payer

➔ Exemple : M:agft:04939582:1#

Protocole-2 : Schéma : **ACTION:REQUETE:ID_UTILISATEUR:ARGUMENTS…**

- Caractère de séparation : ":"
- Parties :
  - **ACTION** : Soit "GET", soit "POST", permet de connaître l'intention de l'utilisateur : Modifier des données ou les obtenir
  - **REQUEST** : dérivé de l'**ORDRE** ci-dessus, complété avec quelques options supplémentaires réservées à l'Application mobile.
  - **ID_UTILISATEUR** : L'identifiant secret de l'utilisateur
  - **ARGUMENTS…** : Données complémentaires à la requête, séparées par des ":". Le nombre d'argument n'est pas limité

Protocole-3 : Schéma : **PROTOCOLE_2_CHIFFRÉ;ID_CLE_CHIFFREMENT;TOKEN_VALIDITE**

- Caractère de séparation : ";"
- Parties :
  - **PROTOCOLE_2_CHIFFRÉ** : Protocole-2 chiffré en AES avec une clé générée à la création.
  - **ID_CLE_CHIFFREMENT** : Identifiant permettant au serveur de retrouver la clé de chiffrement de l'utilisateur
  - **TOKEN_VALIDITE** : Le code de validité reçu du Protocole-1

Protocole-4 : Schéma :
ID_UTILISATEUR#NOM_COMPLET#TYPE_D'UTILISATEUR#CLE_CHIFFREMENT#ID_CLE_CHIFFREMENT

- Caractère de séparation : "#"
- Parties :
  - **ID_UTILISATEUR** : L'identifiant secret de l'utilisateur
  - **NOM_COMPLET** : Le nom complet de l'utilisateur
  - **TYPE_D'UTILISATEUR** : Le type d'utilisateur. Il existe trois types : CLIENT, TERMINAL, SERVEUR
  - **CLE_CHIFFREMENT** : La clé de chiffrement AES
  - **ID_CLE_CHIFFREMENT** : Identifiant permettant au serveur de retrouver la clé de chiffrement de l'utilisateur

Organisation des données

ID-Utilisateur :

- ➔ Emplacement de stockage : **Serveur** et **Téléphone Client**
- ➔ Définition de la donnée : Identifiant de l'utilisateur. Cette donnée permet d'accéder au compte de l'utilisateur. Elle doit rester secrète : elle est chiffrée à chaque transit. Le seul moyen de la découvrir à la réception est de connaître la clé de chiffrement (CLE-Chiffrement) personnelle de l'utilisateur.
- ➔ Sensibilité de la donnée à la sécurité : **Très sensible**
- ➔ Fréquence de transit : **À chaque utilisation**, chiffrée
- ➔ Propre à : **Téléphone Client**

CLE-Chiffrement :

- ➔ Emplacement de stockage : **Serveur** (sur un fichier à part chiffré) et **Téléphone Client**
- ➔ Définition de la donnée : Clé permettant de chiffrer un message en utilisant l'algorithme de cryptographie symétrique AES.
- ➔ Sensibilité de la donnée à la sécurité : **Très sensible**
- ➔ Fréquence de transit : **Une seule fois**
- ➔ Propre à : **Téléphone Client**

ID-CLE-Chiffrement :

- ➔ Emplacement de stockage : **Serveur** et **Téléphone Client**
- ➔ Définition de la donnée : Identifiant permettant au serveur de retrouver la clé de chiffrement (CLE-Chiffrement) de l'utilisateur, et ainsi déchiffrer le message reçu.
- ➔ Sensibilité de la donnée à la sécurité : **Très sensible**
- ➔ Fréquence de transit : **À chaque utilisation**
- ➔ Propre à : **Téléphone Client**

NOM-Utilisateur :

- ➔ Emplacement de stockage : **Serveur**, **Terminal** et **Téléphone Client**
- ➔ Définition de la donnée : Identifiant nominatif de chaque élément, les différenciant de manière lisible.
- ➔ Sensibilité de la donnée à la sécurité : **Normal**
- ➔ Fréquence de transit : **À chaque utilisation**
- ➔ Propre à : **Serveur**, **Terminal** et **Téléphone Client**

TYPE-Utilisateur :

- ➔ Emplacement de stockage : **Serveur**, **Terminal** et **Téléphone Client**
- ➔ Définition de la donnée : Identifiant différenciant un Serveur, d'un Terminal, d'un Téléphone Client pour chaque utilisateur.
- ➔ Sensibilité de la donnée à la sécurité : **Normal**
- ➔ Fréquence de transit : **À chaque utilisation**
- ➔ Propre à : **Serveur**, **Terminal** et **Téléphone Client**

NB-Ticket :

- ➔ Emplacement de stockage : **Serveur**
- ➔ Définition de la donnée : Nombre de tickets que possède chaque Téléphone Client.
- ➔ Sensibilité de la donnée à la sécurité : **Normal**
- ➔ Fréquence de transit : **Sur demande**
- ➔ Propre à : **Téléphone Client**

NB-Argent :

- ➔ Emplacement de stockage : **Serveur**
- ➔ Définition de la donnée : Montant d'argent contenu sur le compte de chaque Téléphone Client
- ➔ Sensibilité de la donnée à la sécurité : **Sensible**
- ➔ Fréquence de transit : **Sur demande**
- ➔ Propre à : **Téléphone Client**

Permissions

Note : Le Serveur possède toutes les permissions. Pour des raisons de clarté, nous ne l'avons pas précisé pour chaque permission.

Obtenir n'importe quelle information :

- Téléphone Portable (concerné par l'information)
- TERMINAL

Modifier le nom d'utilisateur :

- Téléphone Portable (concerné)

Acheter des tickets :

- Téléphone Portable (concerné)

Débiter des tickets :

- Terminal

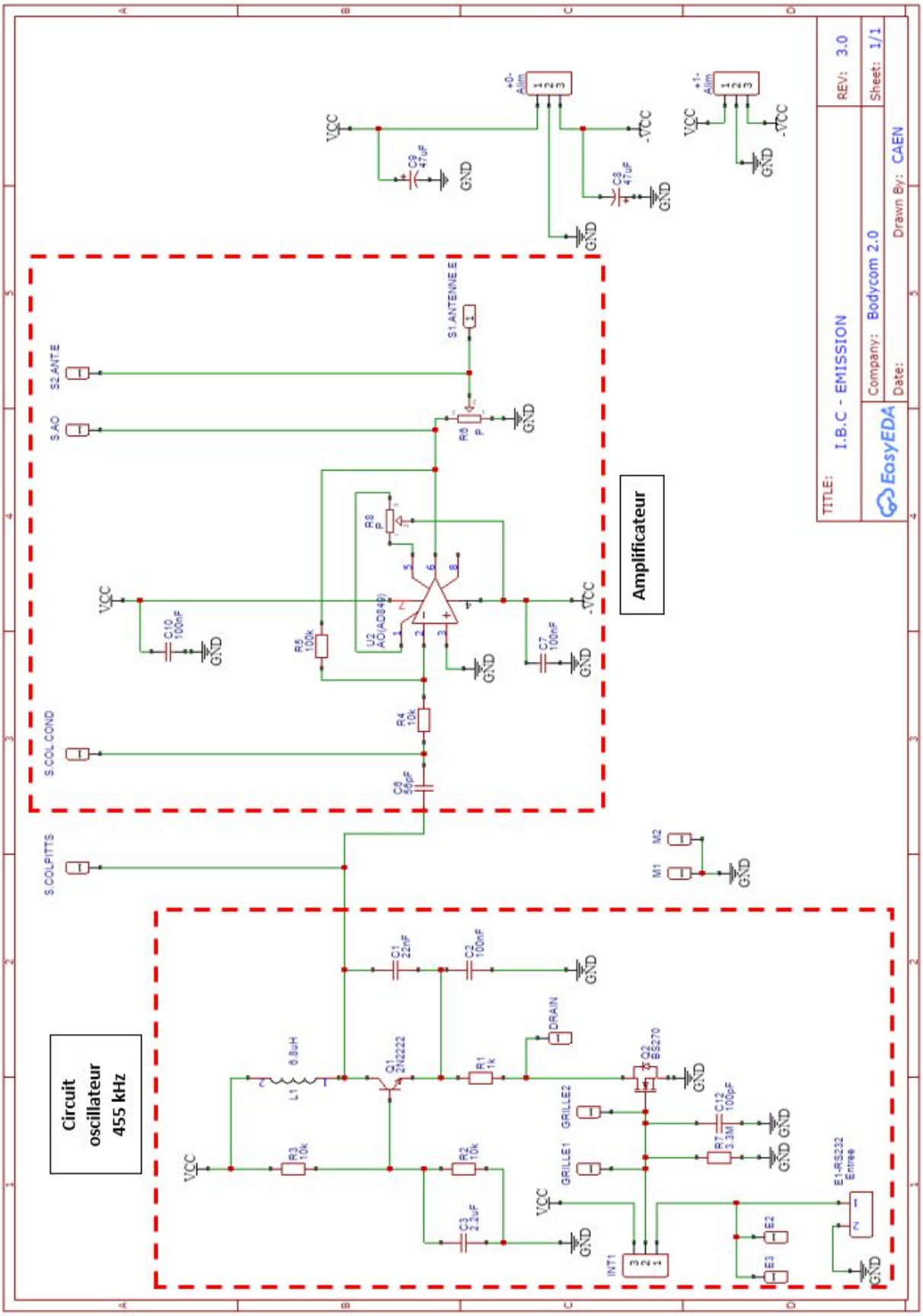
Annexe 3 : Extrait du code Java pour envoyer les données depuis le Terminal vers le corps humain et recevoir la réponse par Bluetooth

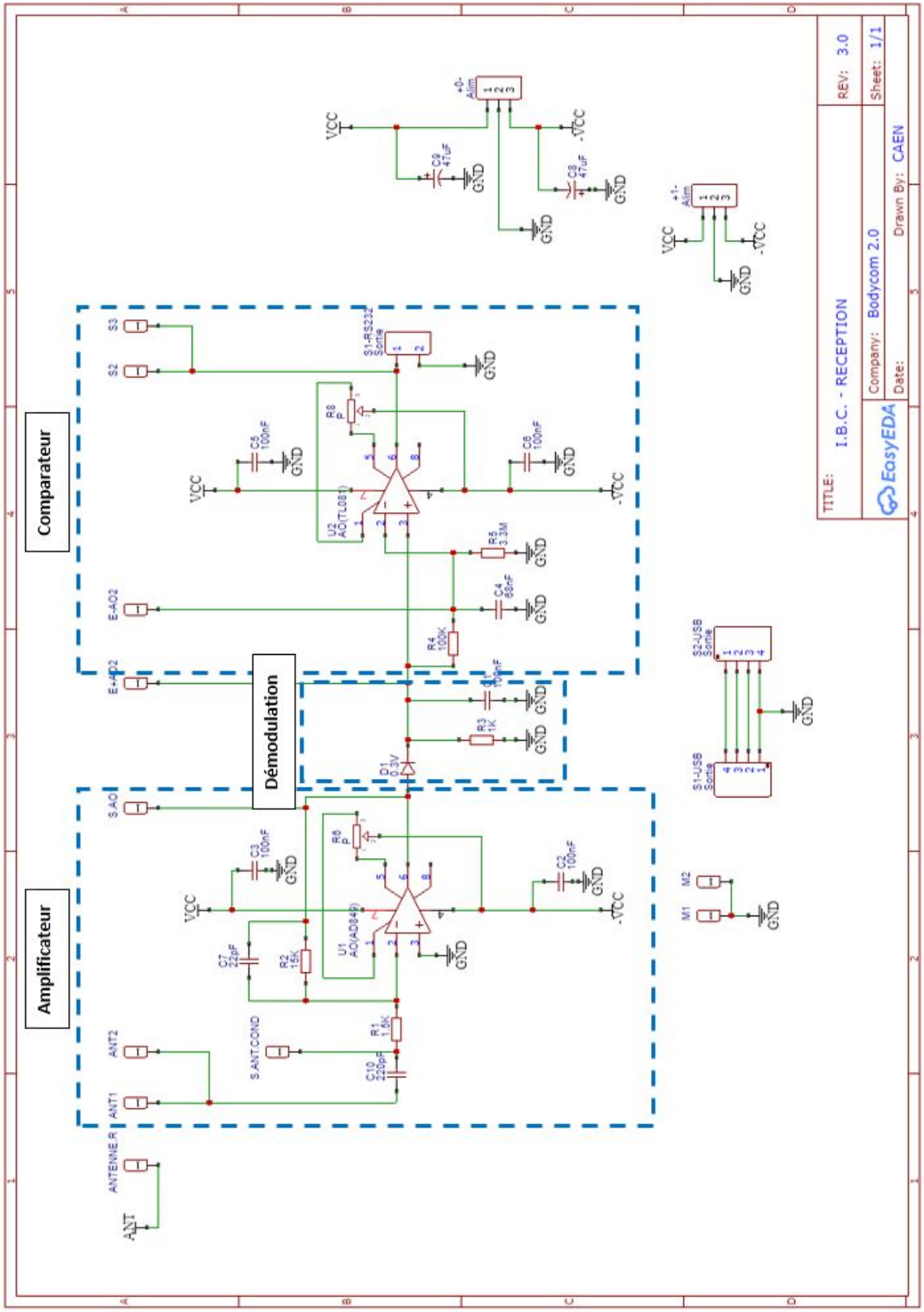
```
public void startProcess(boolean bool) {
    if (processThread != null && !processThread.isInterrupted()) {
        processThread.interrupt();
    }
    this.processThread = new Thread(() -> {
        Thread connectionThread = null;
        SerialHandler serial = null;
        BluetoothConnection bluetooth = null;
        try {
            EventHandler.callEvent(new StateEvent.Start());
            getLogger().info(" ======== Starting process ========");
            String token = UUID.randomUUID().toString().split("-")[2];
            getLogger().info("Generated token : " + token);
            getLogger().info(" ");
            EventHandler.callEvent(new TokenCreateEvent(token));
            EventHandler.callEvent(new BluetoothServiceEvent.Start.Pre());
            try {
                bluetooth = new BluetoothConnection();
            } catch (IOException e) {
                e.printStackTrace();
                EventHandler.callEvent(new BluetoothServiceEvent.Start.Fail(e));
                return;
            }
            EventHandler.callEvent(new BluetoothServiceEvent.Start.Post(bluetooth));
            connectionThread = new Thread(bluetooth::connect, "BT-Connection-Thread");
            connectionThread.start();

            serial = new SerialHandler("", getConfig().getBaud());

            try {
                IBCServerPacket serverPacket = new IBCServerPacket(IBCPacketType.getSelected(), token,
LocalDevice.getLocalDevice().getBluetoothAddress(),
IBCPacketType.getSelected().equals(IBCPacketType.M) ? 20 : 1);
                EventHandler.callEvent(new SerialSendEvent(serverPacket.toString()));
                while (!bluetooth.isConnected()) {
                    serial.write(serverPacket.toString());
                    if (processThread.isInterrupted()) {
                        throw new InterruptedException();
                    }
                }

                getLogger().info("Bluetooth data are available");

                getLogger().info("Sending to Server...");

                try (SocketConnection connection = new SocketConnection()) {
                    Thread.sleep(2000);
                    String msg = new String(bluetooth.readBytes());
                    EventHandler.callEvent(new BluetoothServiceEvent.Receive(msg));
                    getLogger().info(msg);
                    EventHandler.callEvent(new SocketServiceEvent.Send.Pre());
                    Message send = new Message(Message.Type.GET, Message.Request.QUERY,
TerminalUser.toUser(), msg);
                    connection.sendMessage(send);
```

```
                EventHandler.callEvent(new SocketServiceEvent.Send.Post(send));

                String response = connection.waitForAnswer();
                Message message = new Message(response);
                if (message.getRequest().equals(Message.Request.DECLARE)) {
                    getLogger().info("Declaration request received. Send information");
                    try {
                        connection.sendMessage(TerminalUser.declare());
                    } catch (IOException e) {
                        e.printStackTrace();
                    }
                    message = new Message(response = connection.waitForAnswer());
                    getLogger().info("Decrypted data received.");
                }
                EventHandler.callEvent(new SocketServiceEvent.Receive(response, message));
                User user = User.parse(message.getArgs()[0], serverPacket);
                EventHandler.callEvent(new UserEvent(user));
                EventHandler.callEvent(new StateEvent.SendResult());
                this.sendResults(user);
            } catch (IncorrectPacketFormatException e) {
                e.printStackTrace();
            }
        } catch (IOException e) {
            e.printStackTrace();
        }
    } catch (InterruptedException ignored) {
    } finally {
        if (serial != null)
            serial.disconnect();
        if (connectionThread != null) {
            connectionThread.interrupt();
        }
        if (bluetooth != null) {
            try {
                bluetooth.disconnect();
            } catch (IOException e) {
                e.printStackTrace();
            }
        }
        if (bool)
            this.navigate(ProcessPage.class, ((ProcessPage) this.currentController).demo ? "demo" :
"normal");
        else
            this.navigate(MainPage.class);
    }
  }, "PROCESS-Thread");
  this.processThread.start();
}

private void initServer(String[] args) {
  if (INIT)
    return;
  INIT = true;
  launch(args);
}
```

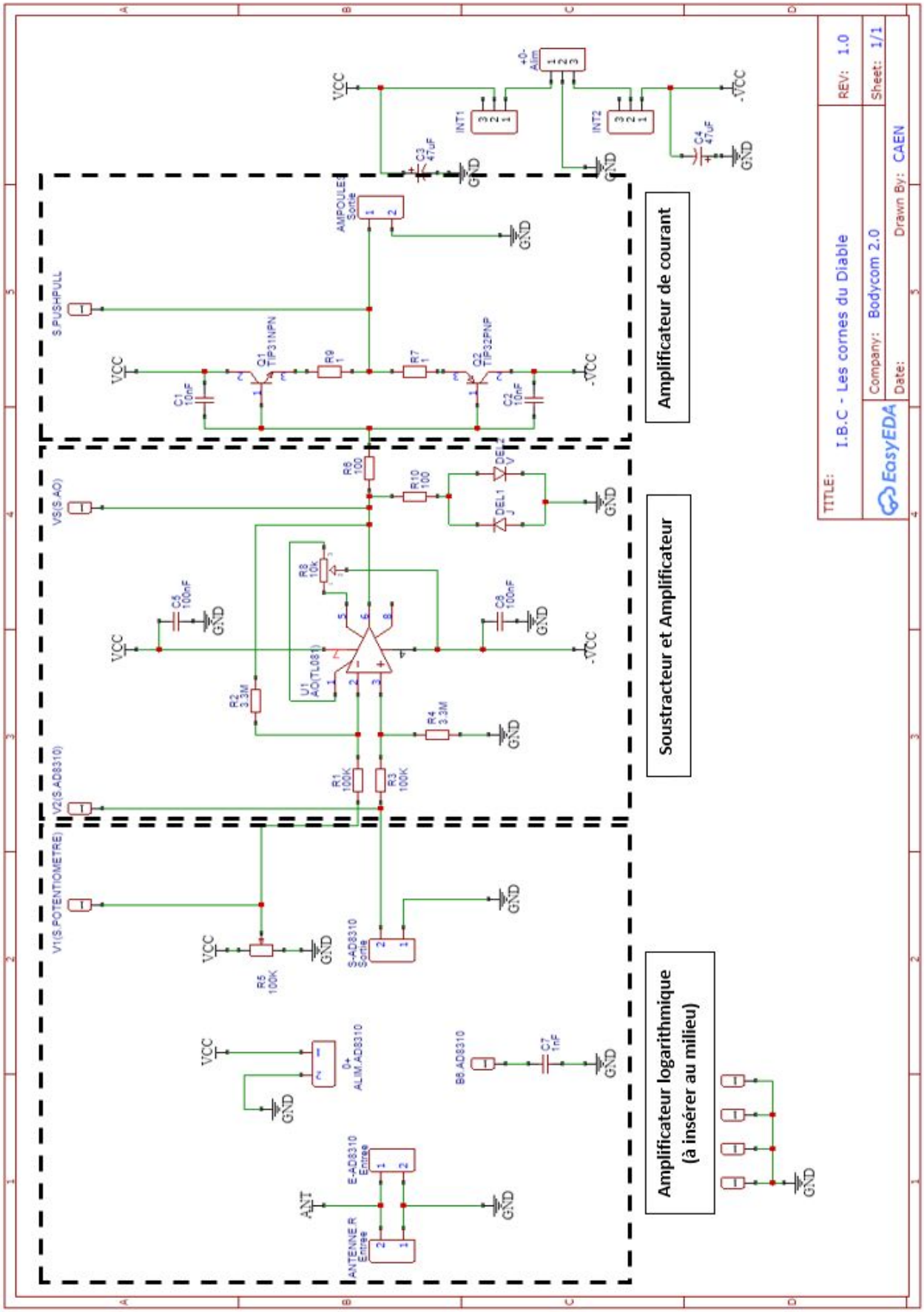
Annexe 4 : Schémas des circuits électroniques

Terminal : Emission des données binaires vers le corps humain

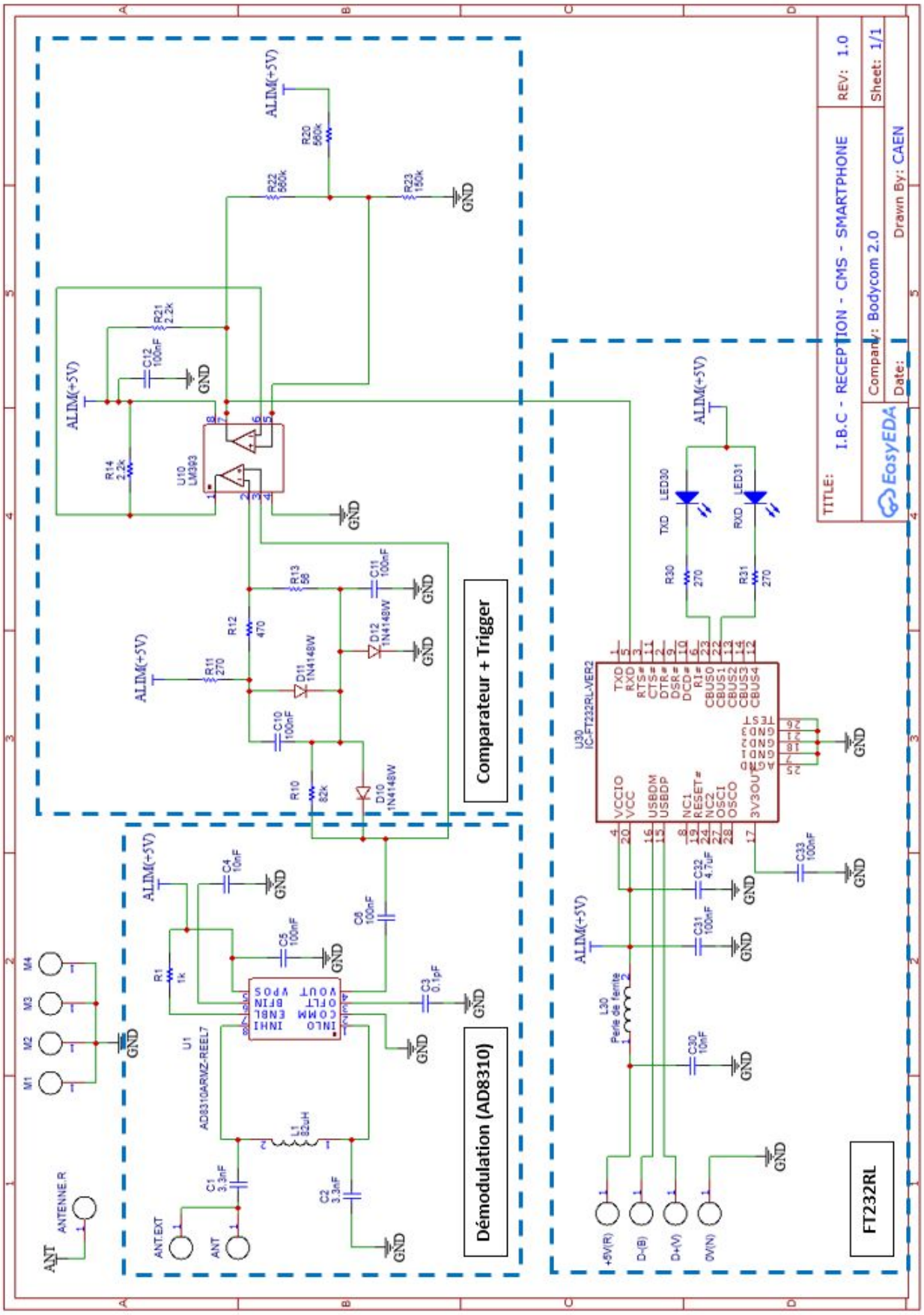




Dans la version finale intégrée à la coque du Smartphone l'amplificateur et le circuit de démodulation sont remplacés par l'amplificateur logarithmique (AD8310)


Amplificateur de courant
Soustracteur et Amplificateur
Amplificateur logarithmique
(à insérer au milieu)
TITLE: I.B.C - Les cornes du Diable
REV: 1.0
Company: Bodycom 2.0
Sheet: 1/1
Date:
Drawn By: CAEN
EasyEDA
S.PUSHPULL
AMPOULE Sortie
Q1 TIP31NPN
Q2 TIP32PNP
R9 1
R7 1
C1 10nF
C2 10nF
C3 47uF
C4 47uF
INT1
INT2
+0- Alim
VCC
-VCC
GND
V3(S.AO)
V2(S.AD8310)
V1(S.POTENTIOMETRE)
R6 100
R10 100
R8 10k
U1 AO(TL081)
R2 3.3M
R4 3.3M
R1 100K
R3 100K
C5 100nF
C8 100nF
DEL1
R5 100K
S-AD8310 Sortie
0+ ALIM.AD8310
B6.AD8310
C7 1nF
E-AD8310 Entree
ANTENNE.R Entree
ANT

Le dispositif final intégré à la coque du Smartphone



Plus de démonstrations sur **https://www.ibc.ovh** (ibc.ovh)

FIN

MERCI