

Année scolaire 2021 - 2022

Projet de fin d'année

Affichage Tête Haute

Étude portée sur la réalisation d'un ATH,
adapté à un casque.

Nom : MOYSE Florie-Anne et PROUX Marc

Formation : CPB2 – ENSISA

Professeur Référent : Jean-Philippe Lauffenburger

Fiche descriptive du projet de CPB 2

Les étudiants

Noms et Prénoms : MOYSE Florie-Anne, PROUX Marc

Année universitaire : 2021 – 2022

Le rapport de projet

Titre du projet :

Affichage Tête Haute (ATH) appliqué à un casque

Mots-clés pour décrire le sujet de votre mémoire (5 maximum) :

Optique, Système embarqué, Informatique, électronique

Court résumé de votre mémoire (5 lignes maximum) :

Ce mémoire a pour but de présenter nos recherches quant à la réalisation d'un Affichage Tête Haute adapté à un casque. Il résumera nos recherches sur l'histoire et le développement des différents types d'ATH ainsi que nos études sur un système optique adapté à notre projet. Il présentera également les solutions que nous avons choisies en terme d'Hardware et de Software pour la réalisation de cet ATH.

Formulaire d'information sur le plagiat

Dans le règlement des examens validé par la CFVU du 2 octobre 2014, le plagiat est assimilé à une fraude.

Le plagiat consiste à reproduire un texte, une partie d'un texte, des données ou des images, toute production (texte ou image), ou à paraphraser un texte sans indiquer la provenance ou l'auteur. Le plagiat enfreint les règles de la déontologie universitaire et il constitue une fraude. Le plagiat constitue également une atteinte au droit d'auteur et à la propriété intellectuelle, susceptible d'être assimilé à un délit de contrefaçon.

En cas de plagiat dans un devoir, dossier, mémoire ou thèse, l'étudiant sera présenté à la section disciplinaire de l'université qui pourra prononcer des sanctions allant de l'avertissement à l'exclusion.

Dans le cas où le plagiat est aussi caractérisé comme étant une contrefaçon, d'éventuelles poursuites judiciaires pourront s'ajouter à la procédure disciplinaire.

Nous soussignés MOYSE Florie-Anne et PROUX Marc

Etudiants à l'Université de Haute Alsace en : CPB ENSISA

Niveau d'études : BAC + 2

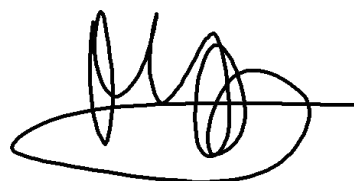
Formation ou parcours : Cycle Post Bac

Reconnaissons avoir pris connaissance du formulaire d'information sur le plagiat.

Fait à Brunstatt

Le 24/03/2022

Signatures :

A stylized, handwritten signature in black ink, consisting of a large, sweeping 'M' followed by a series of loops and a final upward stroke.A stylized, handwritten signature in black ink, featuring a large, horizontal oval shape with several vertical loops and a final horizontal stroke extending to the right.

Moyse Florie-Anne
Proux Marc
CPB2 – ENSISA

Remerciements

Nous tenions à remercier particulièrement les personnes suivantes, qui nous ont accompagnés et aidés dans la réalisation de ce projet de fin d'année :

- M. Jean-Philippe LAUFFENBURGER, Enseignant chercheur à l'ENSISA, responsable de ce projet.
- M. Pierre-Jean LAPRAY, Enseignant chercheur à l'ENSISA, qui nous a permis de réaliser la partie optique de ce projet.

Table des matières

TABLE DES MATIERES	5
LISTE DES FIGURES.....	6
LISTE DES TABLES.....	8
INTRODUCTION	9
HISTOIRE	11
1. DEFINITION ET ORIGINE :	11
2. SYSTEMES EXISTANTS :	12
SOLUTIONS OPTIQUE.....	16
1. GOOGLE GLASS :	16
2. HOLOLENS :	18
3. FORM SWIM GLASSES:	19
4. EYE RIDE :	20
5. NOTRE SYSTEME OPTIQUE.....	21
SOLUTIONS MATERIELS	25
1. MICROCONTROLEUR.....	25
2. SYSTEME DE COMMUNICATION :	27
3. SYSTEME D’AFFICHAGE	27
4. ALIMENTATION.....	28
SOLUTIONS LOGICIEL.....	30
1. PROGRAMMATION ARDUINO.....	30
a. Initialisation du système.....	30
b. Fonction principale	31
c. Fonctions secondaires	32
2. PROGRAMMATION ANDROID.....	34
a. Interfaces graphiques.....	34
b. Programmation	35
SECURITE ET DERIVE	40
CONCLUSION	41
BIBLIOGRAPHIE	ERREUR ! SIGNET NON DEFINI.

Liste des Figures

FIGURE 1 – DIAGRAMME PIEUVRE	10
FIGURE 2 – AVION DE CHASSE MOSQUITO	11
FIGURE 3 – CUTLASS SUPREME	11
FIGURE 4 – ATH AIDE AU DECOLLAGE	12
FIGURE 5 – GOOGLE GLASS	13
FIGURE 6 – HOLOLENS	13
FIGURE 7 – VITURE LUNETTES XR	14
FIGURE 8 – MOJO LENS	14
FIGURE 9 – SMART SWIM GOOGGLES	15
FIGURE 10 – EYE RIDE	15
FIGURE 11 – SCHEMA DU SYSTEME OPTIQUE DES GOOGLE GLASS	16
FIGURE 12 – SCHEMA GOOGLES GLASS (2)	17
FIGURE 13 – UTILISATION DU SEPARATEUR DE FAISCEAU DANS LES GOOGLE GLASS	17
FIGURE 14 – LENTILLE HOLOGRAPHIQUE	18
FIGURE 15 – SCHEMA DU CHEMIN OPTIQUE D'UN RAYON DAN SUNE LAME A FACE PARALLELE ET DANS UN GUIDE D'ONDE	18
FIGURE 16 – VUE DE L'HOLOLENS SANS LA VISIERE	18
FIGURE 17 – HOLOLENS	18
FIGURE 18 – FORM GLASSES	19
FIGURE 19 – VU DE L'ECRAN DE LUNETTE FORM	19
FIGURE 20 – ÉCRAN TRANSPARENT UTILISABLE DANS UN TEL PROJET	19
FIGURE 21 – EYE RIDE SUR LEQUEL ON APERÇOIT LE SEPARATEUR DE FAISCEAUX CUBIQUE	20
FIGURE 22 – LE NANO HD OLED DEVELOPPE PAR EYE LIGHT	20
FIGURE 23 – VUE AVEC L'EYE RIDE	20
FIGURE 24 – SEPARATEUR DE FAISCEAUX CUBIQUE	21
FIGURE 25 – MIROIR INCLINE A 45°	21
FIGURE 26 – ÉCRAN	21
FIGURE 27 – SCHEMA DE NOTRE MONTAGE	21
FIGURE 28 – SIMULATION OPTIQUE FAIT EN TP VU DU DESSUS	22
FIGURE 29 – EFFET DU DIVISEUR DE FAISCEAU (LAME) SUR UN LASER	22
FIGURE 30 – SIMULATION DE SEPARATEUR DE FAISCEAU AVEC UNE NAPPE EN PLASTIQUE	23
FIGURE 31 – SCHEMA SEPARATEUR DE FAISCEAUX CUBIQUE ET LAME	23
FIGURE 32 - SCHEMA COLLIMATEUR	24
FIGURE 33 – RESULTAT DE L'OBSERVATION OPTIQUE	24
FIGURE 34 – ARDUINO NANO	25

FIGURE 35 – RASPBERRY ZERO W	26
FIGURE 36 – MODULE HC-05	27
FIGURE 37 – ECRAN OLED	27
FIGURE 38 – SCHEMA ELECTRIQUE	28
FIGURE 39 – SCHEMA PONT DIVISEUR DE TENSION	29
FIGURE 40 – ARDUINO INITIALISATION	30
FIGURE 41 – ARDUINO MAIN LOOP	31
FIGURE 42 – ARDUINO FONCTION SMS	33
FIGURE 43 – ARDUINO FONCTION BT_STATE	33
FIGURE 44 – ARDUINO FONCTION CLEAR_TEXT_AREA	34
FIGURE 45 – ECRAN DECONNECTE	35
FIGURE 46 – ECRAN CONNECTE	35
FIGURE 47 – APPLICATION INITIALISATION	36
FIGURE 48 – APPLICATION CONNEXION	36
FIGURE 49 – APPLICATION ENVOIE MESSAGE CONSOLE	37
FIGURE 50 – APPLICATION RECEPTION SMS	38
FIGURE 51 – APPLICATION « GET_SENDER »	38

Liste des tables

TABLEAU 1 – CAHIER DES CHARGES.....	9
TABLEAU 2 – ALIMENTATION.....	28
TABLEAU 3 – BIBLIOTHEQUE ARDUINO	31
TABLEAU 4 – CLE D'IDENTIFICATION	32

Dans le cadre de notre projet, nous avons décidé d'étudier les différentes méthodes de réalisation d'un Affichage Tête Haute, dans l'optique de l'appliquer à un casque. En effet, de plus en plus d'entreprises se tournent vers le développement de tel système, que ce soit pour le militaire, l'aviation, l'automobile ou encore, la conquête du Metaverse, faisant de l'étude de l'ATH, un projet d'actualité.

Durant cette étude, nous travaillerons sur l'histoire de l'ATH, son origine et sa démocratisation. Nous travaillerons par la suite à la réalisation d'un ATH devant répondre au cahier des charges ci-dessous.

Fonctions de services	Critères
Affichage d'information	Le système est capable d'afficher les informations clés dans le champs de vision de l'utilisateur.
Utilisation	La mise en oeuvre du système doit être facile pour l'utilisateur et l'application téléphonique claire.
Communication	Le système est capable de se connecter à tout type de téléphone équipé du Bluetooth et de l'application ATH.
Autonome	Le système est capable de récupérer, sans intervention humaine, les données souhaitées et de les afficher de manière efficace et lisible.
Autonomie	Le système possède une batterie avec une autonomie suffisante pour permettre une utilisation prolongé sans interruptions.
Alimentation	Le système est capable de se recharger sur les réseaux électriques locaux.
Eco-Responsable	La fabrication et l'utilisation du système doit avoir une empreinte carbone faible.
Sécurité	L'utilisation du système ne doit pas représenter un danger pour l'utilisateur ou les individus et infrastructures environnant.

Tableau 1 – Cahier des charges

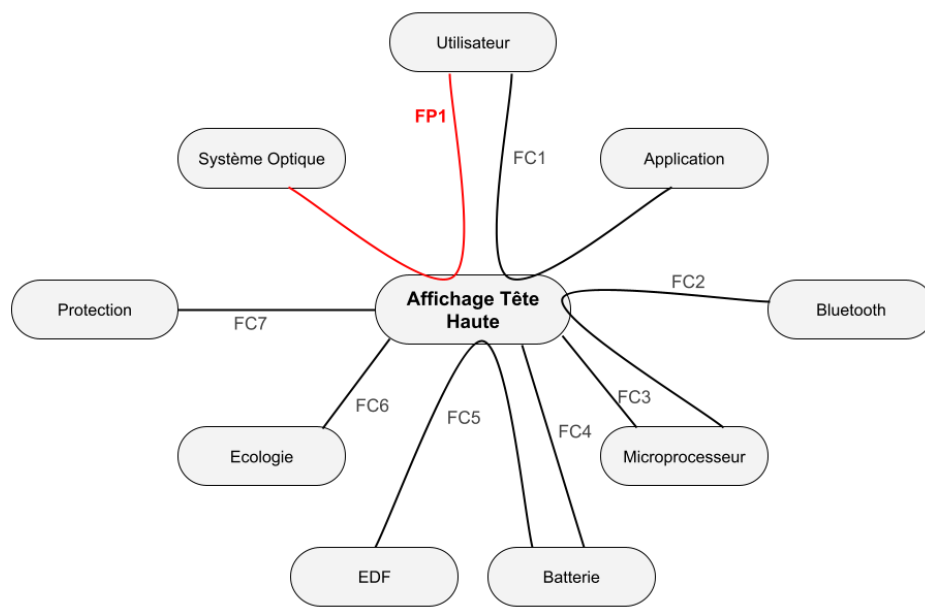


Figure 1 – Diagramme pieuvre

1. Définition et Origine :

L’Affichage Tête Haute est un système permettant la projection d’informations clés dans le champ de vision de l’utilisateur, lui permettant d’accéder à ces informations tout en restant conscient de son environnement.

Le type de données va dépendre du domaine d’utilisation de l’ATH (militaire, automobile, jeux vidéo, etc...)

L’affichage tête haute tire ses origines de l’aviation militaire. C’est durant la seconde guerre mondiale, en 1940, que De Havilland, avionneur britannique, applique une nouvelle technologie au Mosquito, un avion de chasse.

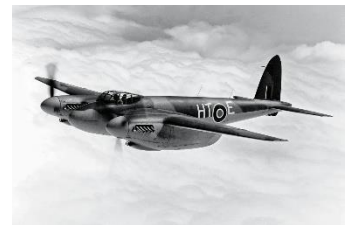


Figure 2 – Avion de chasse Mosquito

A l’intérieur de l’avion se trouve une plaque de verre sur laquelle on peut voir les informations données par le radar, permettant ainsi au pilote de rester attentif à son entourage, tout en gardant des informations importantes dans son champ de vision.

Aujourd’hui encore, des technologies similaires se retrouvent dans quantité d’avions de chasses modernes.

Ce sont ensuite les Américains qui ont pris le relais, avec notamment General Motors, un constructeur automobile, précurseur dans le domaine. C’est en 1965 qu’il imagine équiper une corvette d’un affichage tête haute, mais l’idée est abandonnée.



Figure 3 – Cutlass Supreme

C’est finalement près de 25 ans plus tard, en 1988, que les premiers modèles de voitures pouvant être équipés d’ATH, arrivent sur le marché : le Cutlass Supreme conçu par Oldsmobile.

En parallèle, des constructeurs japonais développent également leur propre système. Puis c'est au tour de Toyota, en 1991, avec le modèle la Crown Majesta.

L'ATH a également continué son évolution dans l'aviation :

- Civile : la compagnie Air France, par exemple, équipe depuis 2009 ses appareils, permettant une aide au décollage et à l'atterrissage.
- Militaire : avec le développement, depuis 2010, de casques avec des dispositifs permettant la projection d'un certain nombre d'informations sur la visière.



Figure 4 – ATH aide au décollage

2. Systèmes existants :

Grace au développement de composants électroniques et optiques de plus en plus performants et de plus en plus petits, certaines entreprises décident de reprendre le principe d'ATH et de l'appliquer à des systèmes plus « portatifs » et polyvalents, tels que des casques de moto, des lunettes ou encore, des lentilles de contact. C'est ce que fit google en 2011 avec son projet Google Glass. L'objectif : créer une paire de lunette équipée d'un système de réalité augmentée, afin de permettre l'affichage d'informations en superposition au champ de vision de l'utilisateur. Mais ce projet fut abandonné en 2015 suite à un cout trop élevé et un manque d'applications dédiées. Néanmoins, ce projet fut repris en 2017, mais orienté à un usage professionnel, et non plus grand publique.



Figure 5 – Google Glass

En parallèle, l'entreprise Microsoft sort en 2017 le premier modèle de sa gamme HoloLens.



Figure 6 – HoloLens

Il s'agit d'un casque utilisant la réalité mixée, permettant à l'utilisateur d'interagir avec des éléments virtuels, tout en interagissant avec son environnement (à l'inverse de la réalité virtuelle où tout se passe dans un univers virtuel). Bien qu'initialement développé pour l'industrie ou encore la médecine, l'HoloLens reste disponible pour le grand public pour un usage privé tel que le gaming, le streaming, les visio-conférences ou encore la programmation.

Plus récemment, une start-up nommée Viture annonce en Mars 2022 son projet de lunette à réalité augmentée, visant le gaming et le streaming.



Figure 7 – Viture Lunettes XR

Composé de lunettes XR et d'un boîtier de contrôle se plaçant autour du cou, ce produit permettait à son utilisateur de jouer à ses jeux favoris sans télévision ni console, grâce aux solutions cloud proposées par les différents fournisseurs (stadia, xbox cloud, geforce now) dans un espace virtuel. A l'inverse d'un casque VR, ce produit ne proposera pas à l'utilisateur une immersion totale dans le jeu, mais il permettra à ce dernier d'utiliser également les lunettes comme un simple accessoire de mode.

Pour pousser le concept à son paroxysme, la start-up Mojo Vision s'est lancée en 2015 dans la recherche et le développement de lentilles de contact équipées de technologie de réalité augmentée. Leur objectif étant de créer un produit ultra portable et ultra discret, permettant l'affichage d'informations cruciales pour l'utilisateur selon ses besoins et ses activités (sport, business, etc..) et ce, sans perturber son champ de vision. Un de leur objectif étant également d'assister les personnes malvoyantes avec leur système. De plus, à l'image de lentille de contact classique, ce produit pourra être adapté selon les corrections oculaire de l'utilisateur.

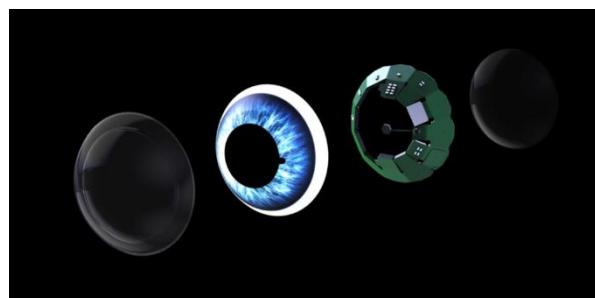


Figure 8 – Mojo Lens

Pour un usage sportif également, la start-up FROM développa les Smart Swim Googles, une paire de lunette de piscine équipée d'ATH permettant l'affichage de statistiques relatives à l'entraînement au nageur ou des programmes et objectifs à suivre.



Figure 9 – Smart Swim Googles

Dans le domaine des transports, nous retrouvons de nombreux modèles d'ATH dans les voitures, directement intégrés par les concessionnaires ou des systèmes modulaires trouvables sur le marché. Néanmoins, de tels systèmes restent très peu développés dans le domaine de la moto, notamment à cause des contraintes dues au manque d'espace. C'est pourquoi en 2016, une start-up française du nom d'Eye Light développe et commercialise un système d'ATH modulaire nommé Eye Ride, adaptable à tout modèle de casque de moto.



Figure 10 – Eye Ride

Comme le montre les systèmes évoqués précédemment, et tous ceux également dont nous n'avons pas parlé, il existe plusieurs solutions pour la réalisation d'un tel projet.

Nous avons tout spécialement décidé de nous pencher sur 4 de ces systèmes, qui ont, pour la même intention, eux des idées très différentes.

1. Google Glass :

A commencer par les Google Glass qui se rapprochent le plus du système que nous essayons de créer.

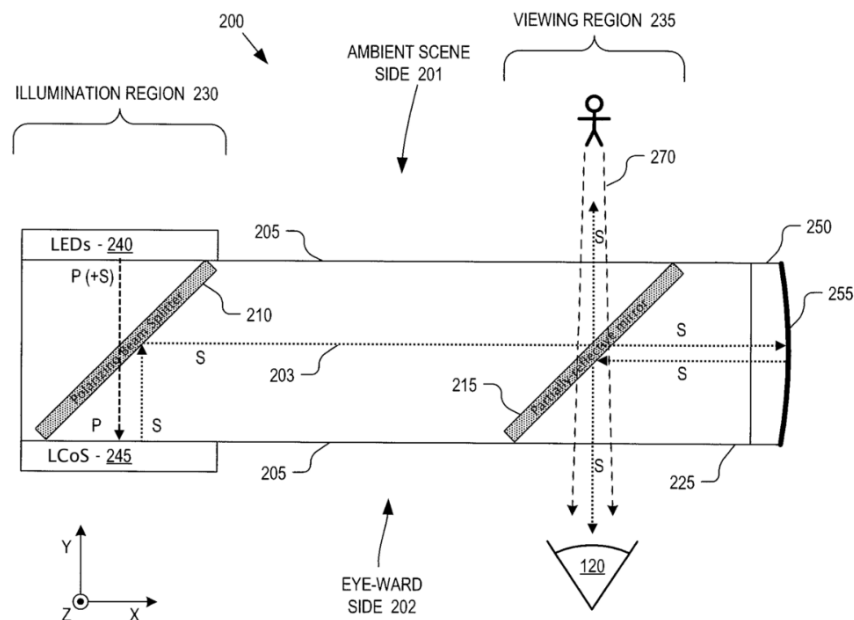


Figure 11 – Schéma du système optique des Google Glass

- 120 - l'œil du spectateur
- 200 - Oculaire à affichage tête haute comprenant :
 - 201 - côté scène ambiante de l'oculaire
 - 202 - côté oculaire de l'oculaire
 - 203 - chemin de propagation de la lumière vers l'avant
 - 204 - chemin de propagation de la lumière inverse
 - 205 - composants de maintien du cadre
 - 210 - séparateur de faisceau polarisant à couplage (PBS)
 - 215 - miroir semi-argenté partiellement réfléchissant à couplage de sortie
 - 230 - région d'éclairage
 - 235 - région de visualisation
- 240 - ensemble d'éclairage comprenant un réseau de LED avec des LED rouges, vertes et bleues et un système de conversion de polarisation composé d'un réflecteur, d'un coin avec un réseau de microlentilles, d'un diffuseur polarisant et d'un polariseur à grille croisée. [1]
- 245 - Affichage à cristaux liquides sur silicium (LCoS)
- 250 - lentille de collimation plan-convexe avec correction d'astigmatisme ; Rayon vertical de 81,87 mm, rayon horizontal de 83,20 mm
- 255 - réflecteur, formé à l'aide d'un revêtement réfléchissant sur la lentille 250
- 270 - scène extérieure

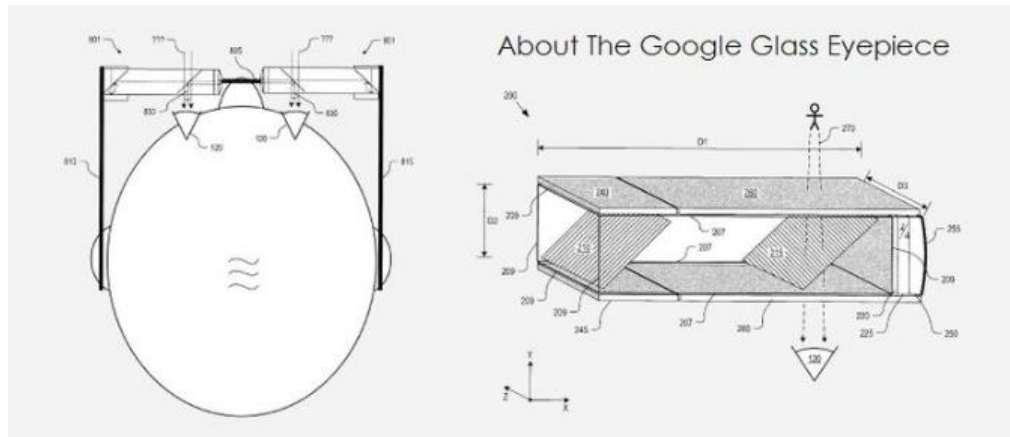


Figure 12 – Schéma Googles Glass (2)

Les Google glass peuvent être vues comme composées de deux parties. Celle de gauche fonctionne comme un interrupteur, le réseau LED émet des faisceaux polarisés selon S et P, seuls les P passent au travers du séparateur de faisceaux polarisant, et arrivent jusqu'aux cellules de cristaux liquides qui renvoient à leur tour des faisceaux soit P, soit S. Si les faisceaux renvoyés sont polarisés selon P, ils passent à nouveau au travers du séparateur, et forment ainsi une boucle. Aucune image ne parvient donc à l'utilisateur. Alors que si les faisceaux sont polarisés selon S, ils seront réfléchis, et passeront dans la « Partie droite » du système.

La partie droite est celle qui nous intéresse pour notre projet : les rayons qui arrivent sont retransmis dans l'œil de l'utilisateur afin de former une image. Cette seconde partie ressemble au système que nous avons mis en place.

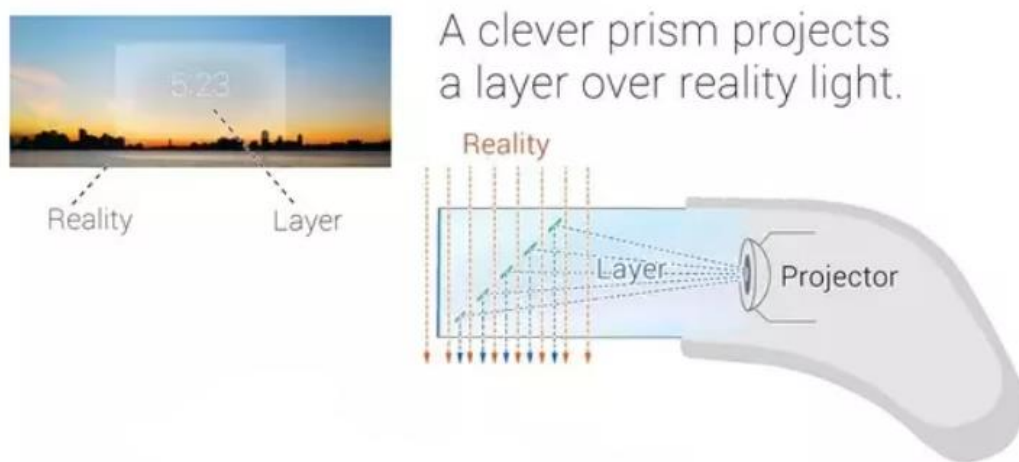


Figure 13 – Utilisation du séparateur de faisceau dans les Google Glass

Dans notre système, ce n'est pas un projecteur, mais un écran qui fait office de source lumineuse.

2. HoloLens :

Le second système intéressant, utilise une tout autre méthode pour transmettre des images à l'utilisateur. Il s'agit de l'HoloLens de Microsoft, le premier ordinateur holographique entièrement autonome.

Le HoloLens, lui, est doté d'une lentille holographique et utilise un guide d'onde.



Figure 14 – Lentille holographique

La lentille Holographique, permet de créer des images en 3D, c'est une lentille avec une forme double concave, les bords sont plus épais que le centre. Lorsqu'un faisceau traverse cette lentille, il se diffuse, se déploie de l'autre côté créant une image avec un effet de profondeur, un hologramme.

Afin d'emmener les faisceaux jusqu'aux lentilles, Microsoft a opté pour le guide d'onde.

Les guides d'onde, en optiques, sont des dispositifs composés de trois couches, qui permettent de confiner et propager la lumière dans une direction bien précise. Les confiner, car ils se reflètent totalement de façon successive sur les deux faces, et les propager grâce au phénomène de résonance dû aux interférences.

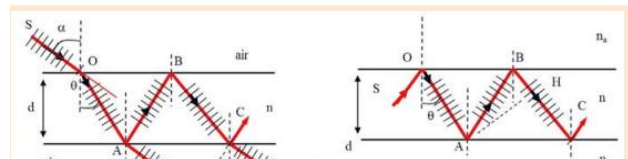


Figure 15 – Schéma du chemin optique d'un rayon dans une lame à face parallèle et dans un guide d'onde

Toutefois, pour éviter que la lumière ne s'autodétruise, il est nécessaire que les interférences soient constructives, ce qui implique un déphasage de 2π .

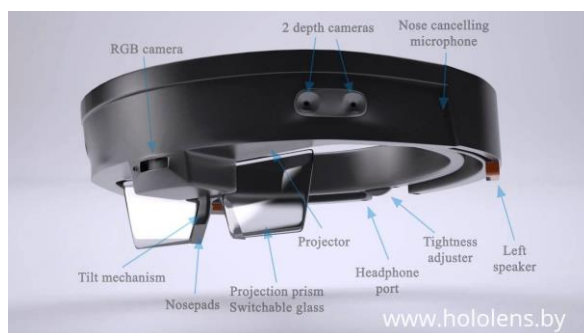


Figure 16 – Vue de l'Hololens sans la visière

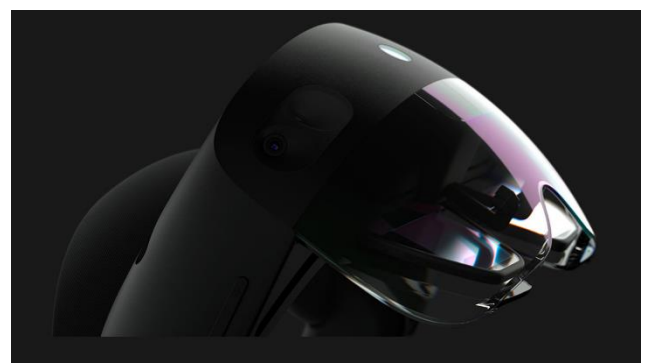


Figure 17 – Hololens

Cette solution, sur laquelle nous possédons peu d'information, n'est pas celle qui nous paraît la plus adaptée non plus pour notre système.

3. FORM swim glasses:



Figure 18 – FORM Glasses

Un troisième système sur lequel nous avons basé nos recherches sont les lunettes de piscine connectées, notamment les FORM glasses. Ces lunettes, sur lesquelles nous avons également des informations assez limitées, utilisent également le guide d'onde. La solution choisie afin d'afficher les informations dans le champ de vision de l'utilisateur semble être un écran transparent.

Cette solution est intéressante, et paraît être la plus simple, mais elle correspond moins à notre projet car bien plus coûteuse et très limitée dans l'affichage, comme pour l'utilisation de couleur par exemple.



Figure 19 – vu de l'écran de lunette FORM

Un tel écran aurait pu être utilisé et placé juste devant l'œil de l'utilisateur :

Soyez le premier à évaluer ce produit

Mini Écran OLED Transparent 128x64 SparkFun Qwiic

par SparkFun Electronics

HT 41,30 €
TTC 49,56 €

Vous avez trouvé un meilleur prix? Lancez l'algorithme de prix

Qté: 1

ACHETER

Notifiez-moi lorsque disponible

Aperçu du produit

- Mini Écran OLED Transparent 128x64 SparkFun Qwiic
- Permet d'afficher des images personnalisées sur un écran transparent
- Eclaire de façon lumineuse dans le noir et toujours visible à la lumière du jour
- Zone d'affichage de 128 x 64 pixels
- Fournit une zone transparente de 128 x 56 pixels

Code de Produit: RB-Spa-1712

Demander à nos experts
Comparatif Affichage LCD

Figure 20 – Écran transparent utilisable dans un tel projet

4. Eye Ride :

Le dernier système se rapproche grandement de celui que nous avons imaginé : il s'agit du produit un affichage tête haute conçu pour les casques de motos, par une start-up française : Eye light.



Figure 21 – Eye Ride sur lequel on aperçoit le séparateur de faisceaux cubique



Figure 22 – Le nano HD Oled développé par Eye Light

Il est fait d'un écran, l'un des plus petit qu'il existe, le nano HD Oled que l'entreprise a développé, et d'un séparateur de faisceaux cubique semblable à celui des Google glass, la différence avec celle-ci est que l'écran projette directement sur le séparateur de faisceaux. (Les images sont sûrement retournées sur l'écran afin d'être vue dans le bon sens par l'utilisateur).



Figure 23 – Vue avec l'Eye Ride

5. Notre système optique

En nous inspirant de ces systèmes et de nos recherches, nous avons imaginé notre propre prototype.

Notre objectif est de créer un système optique permettant l'affichage d'informations en superposition au champ de vision de l'utilisateur. Le tout adaptable à notre prototype de casque, donc en prenant en compte les contraintes de dimensions et d'espace.

Ce système est composé d'un écran, d'un miroir, et d'un séparateur de faisceaux cubique.

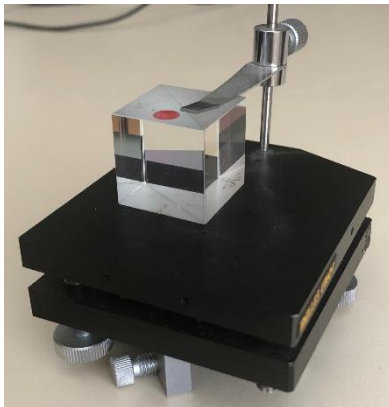


Figure 24 – Séparateur de faisceaux cubique

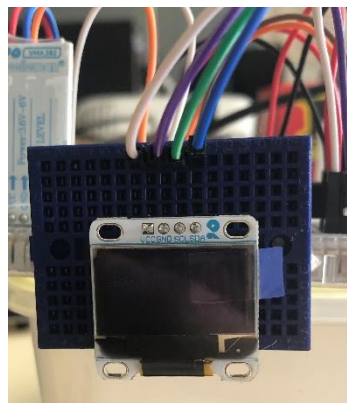


Figure 26 – Écran

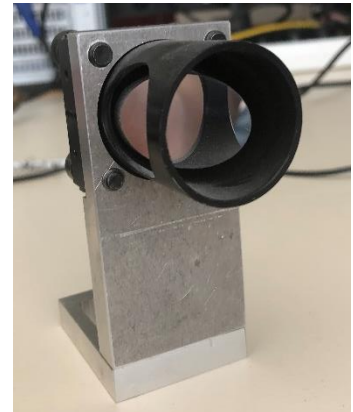


Figure 25 – Miroir incliné à 45°

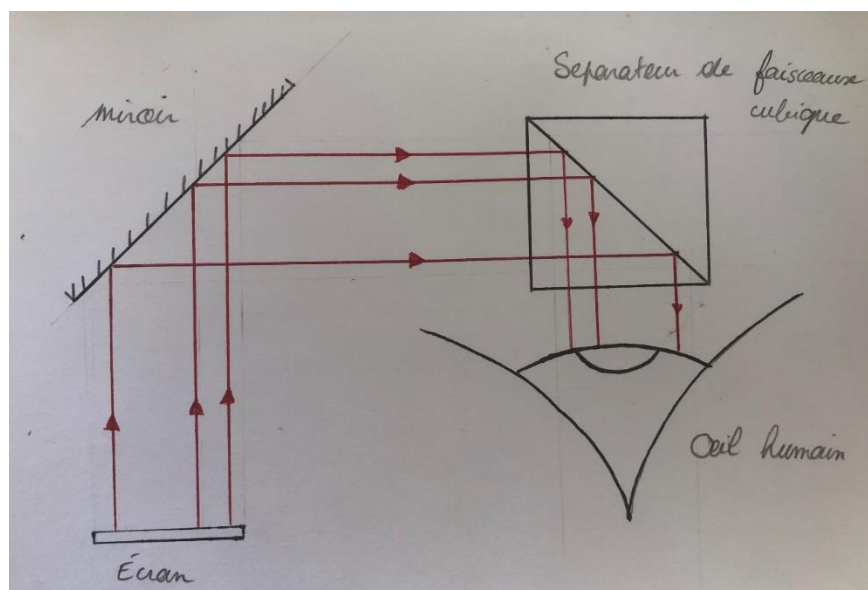


Figure 27 – Schéma de notre montage

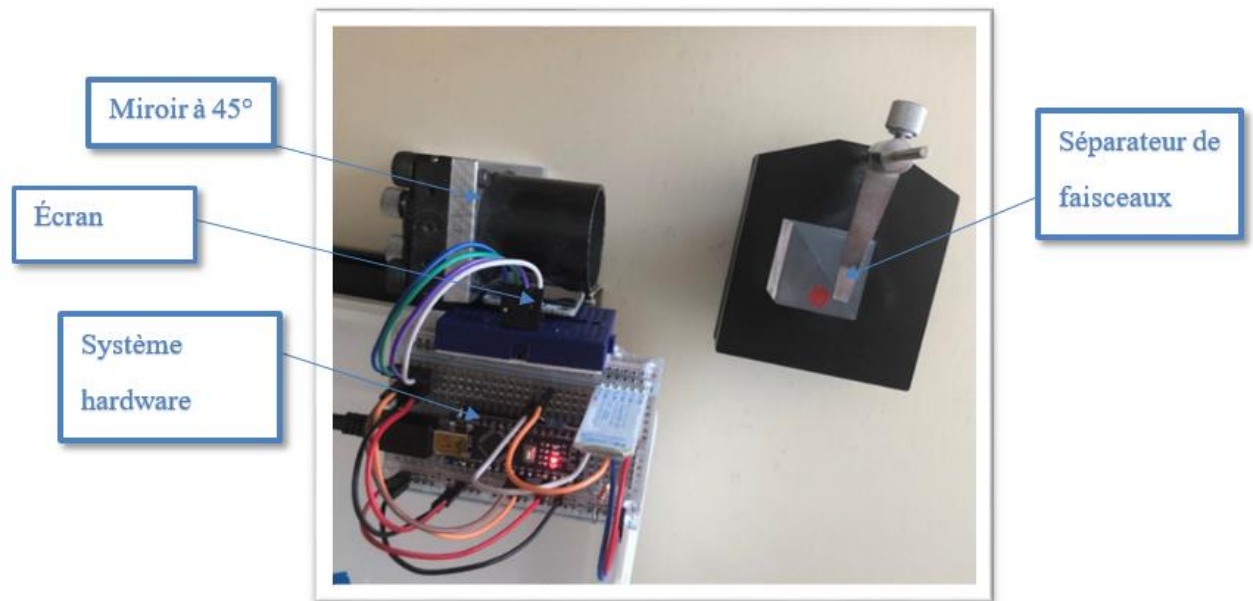


Figure 28 – Simulation optique fait en TP vu du dessus

Dans notre système, nous avons obligatoirement besoin d'un diviseur de faisceaux.

Un diviseur de faisceau est un composant utilisé en optique, afin de diviser un faisceau de lumière incident, qu'il provienne d'un laser ou d'une autre source.

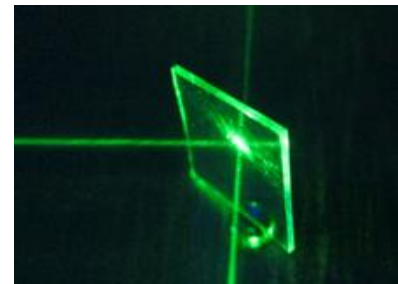


Figure 29 – Effet du diviseur de faisceau (lame) sur un laser

Il existe 3 types de miroirs semi transparents ou séparateurs de faisceaux :

- Les cubiques
- Les lames
- Les prismes ou filtres dichroïques

Ces deux derniers divisent les faisceaux lumineux selon l'angle d'incidence et la longueur d'onde. C'est celui qui nous intéresse le moins dans le cadre de ce projet.

Les lames en verres, dont la surface est revêtue d'une fine couche métallique la plupart du temps, sont conçues pour un angle d'incidence de 45° , impliquant un décalage des chemins optiques.

Les cubiques sont les séparateurs de faisceaux les plus courants, ils sont constitués de deux prismes triangulaires collés afin de former un cube. Effectivement un prisme divise la lumière, faisant par exemple apparaître un arc en ciel dans le cas de la lumière blanche. La lumière venant frapper le prisme est réfractée selon l'indice de réfraction et la longueur d'onde. Si l'on ajoute alors un second prisme là où la lumière sort du premier, elle se retrouve de nouveau réfractée de manière identique, mais dans le sens opposé. Ainsi les différents faisceaux sont recombinaison. Pourtant ils ne sont pas tous réfléchis dans la même direction, on obtient au final deux faisceaux perpendiculaires (comme montré sur les schémas).

On peut aussi noter que le chemin inverse est possible, on peut utiliser ces dispositifs afin de combiner deux faisceaux ensemble.

Pour débiter nos essais, nous avons simulé un miroir semi transparent. À l'aide d'une nappe en plastique transparente nous avons construit un cube avec une diagonale au centre, imitant les deux prismes assemblés. Toutefois ce dispositif est loin d'être optimisé, il y a de l'air à l'intérieur, les bords ne sont pas droits et très facilement déformables.



Figure 30 – Simulation de séparateur de faisceau avec une nappe en plastique

Il nous a quand même permis de faire nos premiers essais, et de remarquer expérimentalement que l'image renvoyée était inversée.

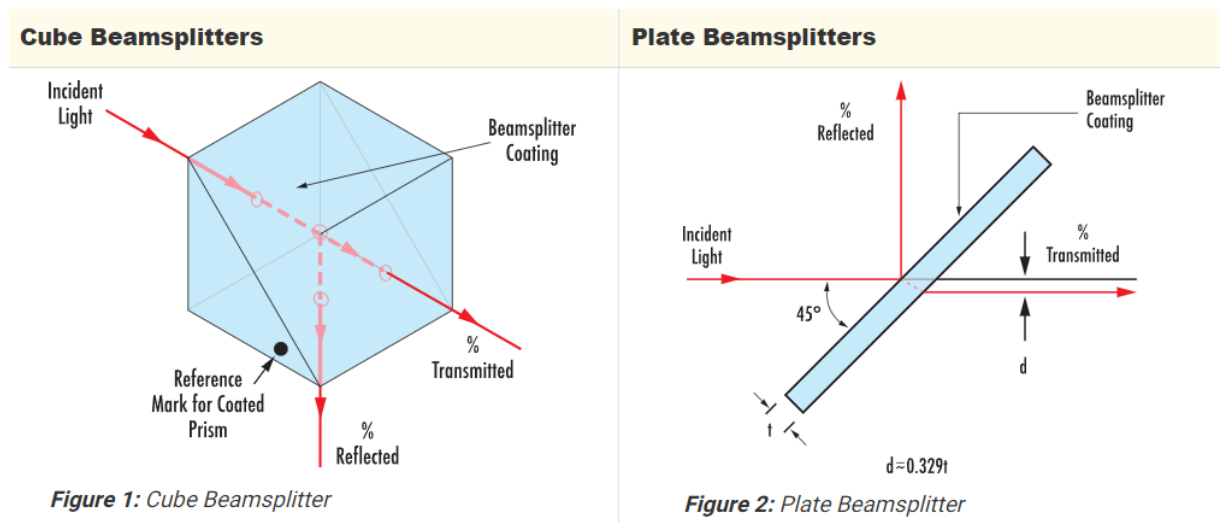


Figure 31 – Schéma séparateur de faisceaux cubique et lame

Remarque :

Afin que l'image soit la plus nette possible et que le confort soit maximal (en considérant l'œil comme une simple lentille convergente dont la rétine est le foyer) il est possible de mettre en place deux solutions :

-La première étant d'utiliser une simple lentille convergente qui permettrait d'envoyer tous les faisceaux lumineux exactement au niveau de la rétine. Cette solution est celle qui nous paraît la plus compliquée. Effectivement le système devrait être modifié à nouveaux pour chaque utilisateur, pour correspondre avec la distance entre son œil et le système, et l'emplacement exact de sa rétine.

-La solution qui apparaît comme la plus pratique est l'utilisation d'un collimateur. Un collimateur est un dispositif optique permettant de passer d'une source lumineuse à un ensemble de faisceaux tous parallèles les uns avec les autres. Avec un tel dispositif, tous les rayons semblent venir de l'infini et il n'est pas nécessaire de faire un focus pour que l'image soit nette. L'œil, tel une lentille, fait converger tous les rayons parallèles directement sur la rétine.

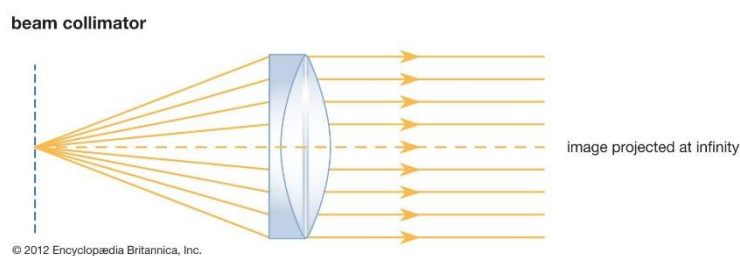


Figure 32 - Schéma collimateur

Dans notre cas pour une approche théorique on supposera que les faisceaux émis par la dalle LED sont tous parallèles.

Avec les éléments optiques mis à notre disposition, nous avons pu réaliser une simulation et ainsi obtenir l'image suivante :



Figure 33 – Résultat de l'observation optique

Notre ATH étant un produit nomade, il nécessitera une source d'électricité portable, pouvant se recharger sur les réseaux électriques domestiques.

Afin de récupérer les informations que l'utilisateur souhaiterait afficher sur son champ de vision, notre ATH devra être équipé d'un système de communication compatible avec les différents modèles de Smartphone.

Le but de notre système étant l'impression de ces informations dans le champ de vision de l'utilisateur, il devra posséder un composant qui, couplé au système optique, permettra l'affichage de ces données.

Enfin, pour permettre la récupération, la gestion et l'affichage des données, notre système nécessitera un microcontrôleur.

1. Microcontrôleur

Le microcontrôleur est le centre névralgique de notre ATH. C'est lui qui récupérera les données transmises par le téléphone de l'utilisateur et les affichera selon sa programmation dans le champ de vision de ce dernier.

Il devra être suffisamment puissant pour réaliser sa tâche tout en étant suffisamment compact pour se loger dans un casque.

Pour commencer, notre ATH aura besoin d'afficher uniquement du texte tel que le contenu d'un SMS ou encore l'heure. Nous avons donc choisi d'utiliser la carte Arduino Nano.

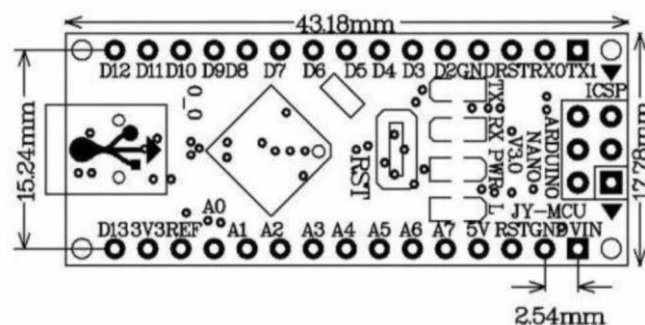


Figure 34 – Arduino Nano

2. Système de communication :

L'Arduino nano étant dépourvu de tout système de communication sans fil, il est nécessaire de lui rajouter un composant électronique capable d'assurer la liaison entre le microcontrôleur et le téléphone. Nous avons décidé de nous tourner vers le module Bluetooth HC-05.

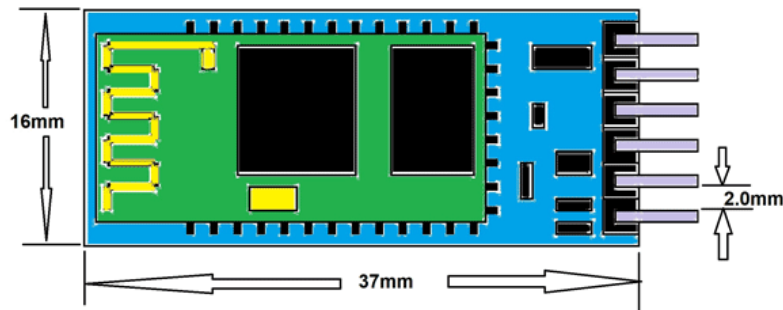


Figure 36 – Module HC-05

Ce module a l'avantage d'être également très compact et d'être compatible avec les protocoles Bluetooth des différents Smartphones. Ce module est un module « Maître », ce qui signifie qu'il peut recevoir des demandes d'appareillage mais aussi en émettre et se connecter automatiquement à un appareil dit « Esclave ». Dans notre cas, nous n'aurons pas besoin de cette fonctionnalité « Maître ». Nous le configurerons donc en mode « Esclave ».

3. Système d'affichage

Afin d'accomplir sa fonction, notre prototype doit être équipé d'un écran qui, grâce au système optique, permettra l'affichage des informations dans le champ de vision de l'utilisateur. Cet écran devra répondre à deux contraintes, être petit, mais posséder une bonne résolution afin de permettre un affichage net et adapté.



Figure 37 – Ecran OLED

Nous avons décidé de nous tourner vers ce modèle d'écran qui a l'avantage d'être compact (27x27x4,1 mm) et de posséder une résolution suffisante pour l'usage demandé (128 x 64 pix). Cet écran possède un bus I2C, compatible avec Arduino via les pilotes SSD1306.

4. Alimentation

Afin d'assurer la portabilité de notre ATH, il est nécessaire d'équiper notre système d'une source d'alimentation adaptée. Elle devra donc correspondre aux exigences énergétiques des différents composants.

Modules	Tension d'Alimentation
Arduino Nano	4V – 6V
HC-05	3,6V – 5V
Ecran OLED	3V – 5V

Tableau 2 – Alimentation

Le module Bluetooth et l'écran seront alimentés via la prise 5V de l'Arduino. Cette dernière sera alimentée en 5V également via son port mini-USB grâce à une batterie LiPo.

Cependant, l'entrée RXD de module HC – 05 ne supporte qu'une tension de 3,3V. Il sera donc nécessaire d'utiliser un pont diviseur de tension à son niveau pour réduire la tension de sortie de l'Arduino qui est de 5V.

Nous avons ainsi réalisé le circuit suivant :

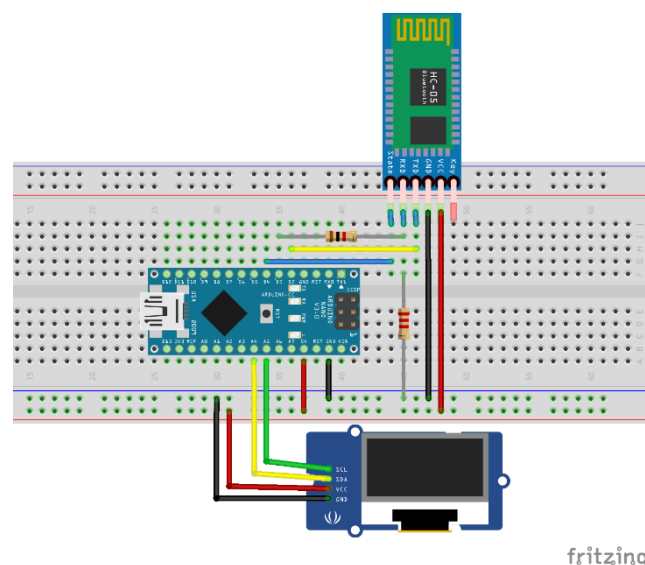


Figure 38 – Schéma électrique

Afin de réaliser le pont diviseur de tension, nous avons placé une résistance de $1\text{k}\Omega$ entre le pin 3 et l'entrée RXD du HC – 05 et une résistance de $2,2\text{ k}\Omega$ entre cette même entrée et la masse de l'Arduino. Pour vérifier que la tension de sortie est bien celle souhaitée, nous appliquons la formule du pont diviseur de tension.

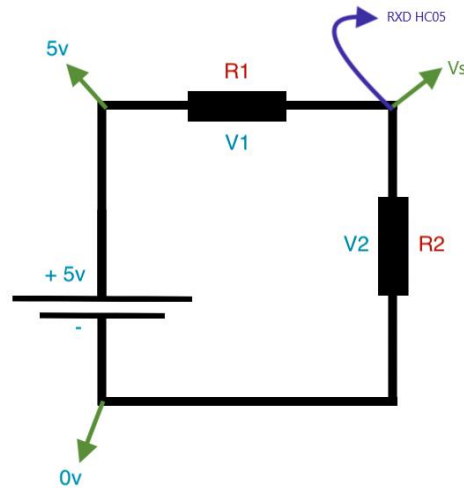


Figure 39 – Schéma pont diviseur de tension

$$V_s = V_e \times \frac{R_2}{R_1 + R_2} = 5 \times \frac{2,2}{2,2 + 1} = 3,4\text{ V}$$

On retrouve donc bien une valeur tolérable pour le système.

Afin d'assurer le bon fonctionnement de l'ATH, notre système devrait être programmé de manière à assurer la bonne communication entre l'Arduino et ses modules avec le téléphone. Pour assurer cela, nous avons dû programmer l'Arduino pour répondre au cahier des charges, ainsi qu'une application sous Android pour récupérer et envoyer les données souhaitées, et ce selon le protocole de communication établie.

1. Programmation Arduino

L'Arduino permet de récupérer et traiter les informations envoyées par le téléphone et de gérer les composants du système. Nous avons pu diviser son code de programmation en trois parties.

a. Initialisation du système

L'objectif de cette partie du code est d'initialiser les variables et les modules nécessaires au bon fonctionnement du système, à savoir le module HC – 05 et l'écran I2C.

```

/*
 *
 * File name : ATH.ino
 *
 * Description :  Retrieves and processes the information sent by the
 *               application in order to display it on the ATH.
 *
 * Coder : Marc PROUX
 *
 */

#include <SoftwareSerial.h>
#include <Adafruit_SSD1306.h>
#include <Wire.h>

#define pixWidth 128 //Defines the number of pixels wide
#define pixHeight 64 //Defines the number of pixels height
#define resetPin -1 //Define de reset Pin (-1 for none)
#define i2cAddress 0x3C //Sets the address of the i2C display
Adafruit_SSD1306 ecran(pixWidth, pixHeight, <Wire>, resetPin);

#define rxPin 2 //Set the pin 2 as RX
#define txPin 3 //Set the pin 4 as TX
#define baudrate 9600 //Set the HC05 baudrate to 9600
#define state 4 // Set the HC05 state pin to 4

SoftwareSerial hc05(rxPin , txPin);
String entry; //Create the Entry variable that will retrieve the HC05 message

void setup() {
    pinMode(rxPin, INPUT);
    pinMode(txPin, OUTPUT);
    pinMode(state, INPUT);
    Serial.begin(19200);
    hc05.begin(baudrate);

    // =====
    //      Initialization Display
    // =====
    ecran.begin(SSD1306_SWITCHCAPVCC, i2cAddress);
    ecran.clearDisplay(); // Clear the display buffer
    ecran.display(); // Display the buffer on screen
    ecran.setTextColor(WHITE, BLACK); // Set the font color as Black (Pix off) and the Text color to White (Pix on)
}

```

Figure 40 – Arduino initialisation

Nous commençons par faire appel aux bibliothèques nécessaires au programme :

Nom	Rôle
SoftwareSerial	Permet de définir d'autres PIN comme port Serial
Adafruit_SSD1306	Bibliothèque permettant le contrôle d'écran I2C sous protocole SSD1306
Wire	Assure la communication I2C

Tableau 3 – Bibliothèque Arduino

Nous configurons par la suite l'écran I2C et le module HC – 05. Pour l'écran, nous définissons sa résolution, son PIN de réinitialisation (à -1 car nous n'en n'avons pas) ainsi que son adresse I2C et pour le module Bluetooth, les PIN RX et TX pour la communication Serial, sa vitesse de communication ainsi que l'entrée qui va pouvoir récupérer l'état logique de connexion du module au téléphone.

Dans cette partie d'initialisation, nous déclarons également une variable de type String qui contiendra les données envoyées par le téléphone et nous préparons l'écran en vidant son buffer, effaçant tout élément affiché et en définissant la couleur du texte tel que ce soit pixel allumé (White) sur fond de pixels éteints (Black). Ayant choisi un écran bleu, la véritable couleur d'affichage ne sera pas blanche, mais bleue.

b. Fonction principale

Cette fonction est la fonction principale de l'Arduino. Durant toute la durée de fonctionnement du système, elle se répétera de manière infinie.

```
// =====
//           Main loop
// =====

void loop() {
  while (hc05.available()){ // While HC05 sending information to Arduino
    entry += char(hc05.read()); // Add the char sent by the HC05 to the entry string
    Serial.println(entry);
  }
  //Serial.println("End");
  delay(200);
  if (entry!=""){
    sms(); // Call the notif function
    Serial.println("Begin sms");
  }
  bt_state(); // Call the bt_state function
}
```

Figure 41 – Arduino main loop

Cette fonction a 3 objectifs. Le premier est de détecter la réception d'un message par le module HC – 05. Si la carte détecte une transmission, le programme lancera une boucle qui permettra d'ajouter chaque caractère envoyé par le téléphone à la variable « entry ». Le programme sort automatiquement de cette boucle quand le message est entièrement transmis.

Son deuxième objectif est de, s'il y a des éléments dans la variable « entry » et donc, un message reçu, faire appel à la fonction « sms() ».

Enfin, sa dernière fonction est de vérifier l'état de la connexion Bluetooth en faisant appel à la fonction « bt_state() ».

c. Fonctions secondaires

La troisième et dernière partie de notre code est composée de trois fonctions secondaires, mises en jeu par la fonction principale, afin de rendre notre système fonctionnel.

➤ *Fonction sms()*

Le but de cette fonction est de décomposer le message reçu, afin de le traiter comme il se doit. Dans le cadre de ce projet, nous avons décidé de faire un système n'affichant que les sms réceptionnés par le téléphone et les messages envoyés depuis la console à destination de notre ATH. On peut cependant d'ores et déjà imaginer une évolution du système avec un plus large panel d'informations envoyées par le téléphone, tels que le niveau de batterie ou l'heure. Nous avons donc décider d'établir un protocole de communication pour que notre système puisse ultérieurement déterminer la nature de chaque information et de les traiter en conséquence. Chaque partie d'information sera donc précédée d'une clé d'identification. Les premières clés établies sont résumées ci-dessous :

Nom	Description	Clé
SMS	SMS entrant	0x6D
Sender	Emetteur du SMS	0x73
Notif	Message envoyé via la console	0x6E
Time	Heure	0x74

Tableau 4 – Clé d'identification

Comme évoqué, ce système sera utilisé uniquement comme voie d'amélioration. Pour ce projet, nous traiterons uniquement les SMS entrants et les messages envoyés via la console, qui seront eux aussi considérés comme des SMS envoyés par « System ». Nous utiliserons donc uniquement les clés 0x6D et 0x73

```
// =====
//          Sms function
// Will process the entry information to
// display it on the screen
// =====

void sms() {
    String sender = entry.substring(5, entry.indexOf("0x6D")); // Get the id of the sender located between code 0x73 and 0x6D
    String message = entry.substring(entry.indexOf("0x6D")+5); // get the id of the sender located after the code 0x6D
    Serial.println(sender);
    Serial.println(message);
    clear_text_area(); // Call the clear_text_area function
    ecran.setCursor(0, 15);
    ecran.print(">>> ");
    ecran.print(sender); // Place the sender name in the screen buffer
    ecran.setCursor(0, 30);
    ecran.print(message); // Place the message in the screen buffer
    ecran.display(); // Display the buffer on screen
    delay(5000);
    clear_text_area(); // Call the clear_text_area function
    entry=""; // Clear the entry variable
}
}
```

Figure 42 – Arduino fonction sms

Après avoir déterminé la nature de l'information en découpant la variable « entry » grâce aux clés d'identification, la fonction « sms () » devra l'afficher sur l'écran. Pour ce faire, elle commencera par effacer la zone prévue pour le texte grâce à la fonction « clear_text_area() » puis affichera le « Sender » et « message » en dessous.

➤ Fonction bt_state()

Cette fonction a pour objectif de vérifier si le système est toujours bien connecté au téléphone. Pour ce faire, elle récupérera l'état logique sur la broche 4, HIGH quand l'ATH est connecté et LOW sinon. Cet état logique sera récupéré à chaque itération de la fonction principale.

```
// =====
//          bt_state function
// Checks if the HC05 is connected to
// the phone and blocks the program
// otherwise
// =====

void bt_state() {
    if (digitalRead(state) == 0) { // If the HC05 state is LOW, this means that it is not connected
        ecran.clearDisplay();
        ecran.display();
        ecran.setCursor(0, 32);
        ecran.print("Connect to phone"); // Asks to user to connect the system
        ecran.display();
        while (1) { // Infinite that loop // Infinite loop that blocks the program until the module is reconnected
            if (digitalRead(state) == 1) { // If the HC05 state is HIGH, this means that it is not connected
                ecran.setCursor(0, 32);
                ecran.clearDisplay();
                ecran.display();
                ecran.print("Connected"); // Notifies the user that they are back online
                ecran.display();
                delay(2000);
                clear_text_area();
                break; // Exits the infinite loop
            }
        }
    }
}
}
```

Figure 43 – Arduino fonction bt_state

Si la broche renvoie un état LOW, la fonction bloquera le système via une boucle infinie, et affichera sur l'écran de l'utilisateur un message lui demandant de se reconnecter. A chaque itération de la boucle, l'état logique sera de nouveau vérifié, et si elle devient HIGH, le système affichera pendant quelques secondes un message confirmant la reconnexion et sortira de la boucle créée précédemment.

➤ *Fonction clear_text_area()*

Le rôle de cette fonction est d'effacer la zone dédiée à l'affichage du texte. Une telle zone permet de laisser le haut de l'écran libre pour l'affichage en continu d'autres informations telles que l'heure qu'il ne faudrait pas effacer durant l'utilisation.

```
// =====  
//          clear_text_area  
// Clear the part of the screen intended  
//      for displaying notifications  
// =====  
  
void clear_text_area() {  
    for (int width = 0; width < 128; width++) {  
        for (int height = 15; height < 64; height++) {  
            ecran.drawPixel(width, height, BLACK); // Turns all pixels in the selected area off  
        }  
    }  
    ecran.display();  
}
```

Figure 44 – Arduino fonction clear_text_area

Composée de deux boucles for imbriquées, la fonction passera sur chacun des pixels de la zone afin de les éteindre, et donc d'effacer tous éléments présents dans cette dernière.

2. Programmation Android

Afin de récupérer les notifications du téléphone et de permettre à l'utilisateur d'avoir la main sur son ATH, nous avons dû développer une application Android. Nous la programmerons à l'aide de l'outil de google App Inventor, un logiciel de développement facile à utiliser grâce à son système de programmation par bloc, à l'instar de Scratch. Composé de deux interfaces différentes, notre code se décompose en 5 fonctions.

a. Interfaces graphiques

Comme indiqué ci-dessus, notre application se décompose en deux interfaces graphiques. La première quand le téléphone n'est pas connecté à l'ATH, et le deuxième quand il l'est.

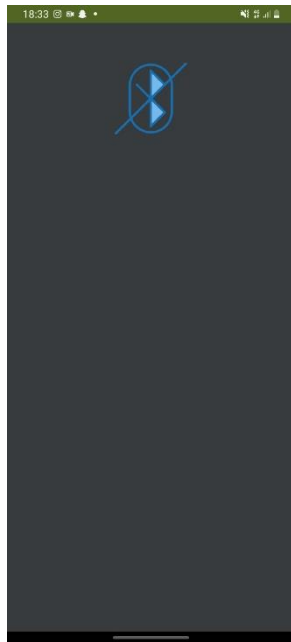


Figure 45 – Ecran déconnecté

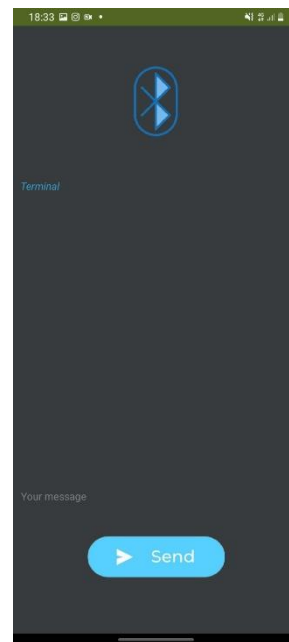


Figure 46 – Ecran connecté

Le premier écran est simplement équipé d'un bouton servant à la connexion Bluetooth. La seconde quant à elle, est équipée du même bouton, servant cette fois ci à la déconnexion et d'une console grâce à laquelle nous pouvons envoyer des messages à l'ATH. A noter que, lorsqu'on appuie sur le bouton de déconnexion, nous retournons sur la première interface, et la console est vidée.

b. Programmation

Afin de répondre au mieux aux demandes de l'utilisateur, nous avons divisé le code de l'application en 5 parties. L'initialisation, la connexion, l'envoi de données via la console, la réception de SMS, et la fonction de recherche de l'émetteur du message dans l'annuaire de l'utilisateur.

➤ Initialisation

Tout comme pour la programmation de l'Arduino, la partie initialisation de l'application sert à définir les variables globales et à initialiser le Bluetooth.

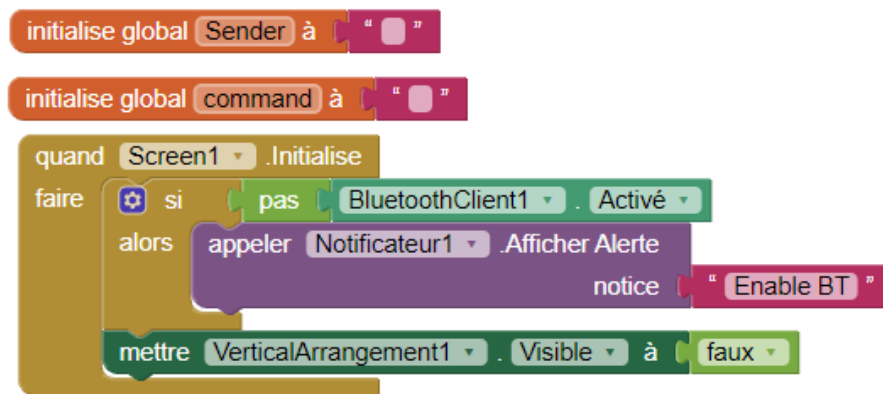


Figure 47 – Application initialisation

On définit ici la variable « Sender » qui contiendra le nom de l’émetteur et la variable « command » qui recevra l’ensemble des messages reçus qui seront affichés dans la console de l’application. Ensuite, lorsque l’écran s’initialise, nous vérifions que le Bluetooth est bien activé sur le téléphone de l’utilisateur et, cas échéant, une alerte lui est envoyée pour qu’il l’active. On finit par masquer la partie inférieure de l’écran, à savoir tous les éléments concernant la commande qui restent masqués tant que l’appareil n’est pas connecté à l’ATH.

➤ Fonction de connexion

Cette fonction est appelée quand le bouton de connexion est pressé. Elle a pour but de connecter ou de déconnecter le téléphone à l’ATH et de modifier l’interface graphique en fonction.

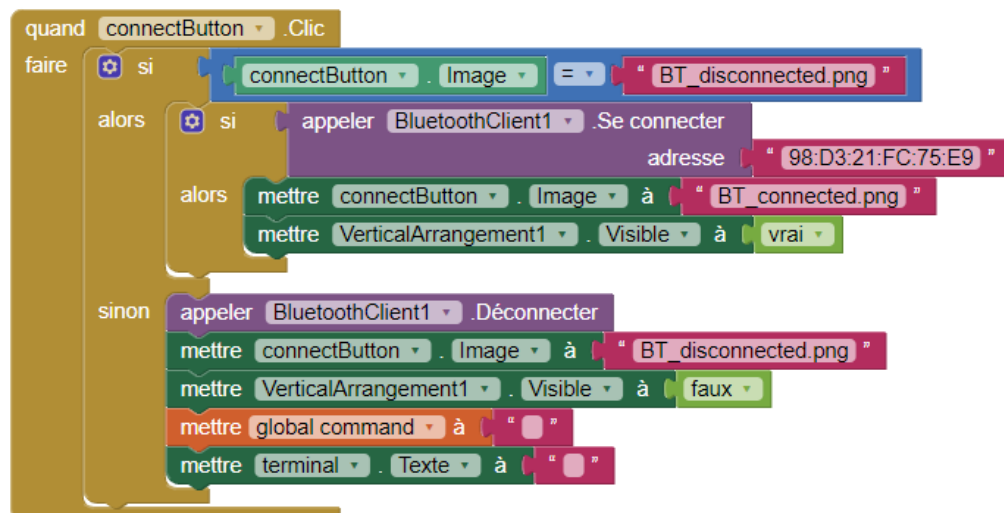


Figure 48 – Application connexion

Elle consiste à ce que, quand le bouton est pressé, le programme vérifie s'il est en mode déconnecté ou connecté. Dans le premier cas, l'application tentera d'établir une liaison Bluetooth avec l'ATH. S'il y arrive, il changera l'image du bouton en connecté, et affichera le terminal. Le cas échéant, il affichera un message d'erreur à l'utilisateur.

Dans le cas où le bouton est en mode connecté, l'application interrompt la connexion avec le système, masque et vide la console et change l'image du bouton en déconnecté.

➤ *Envoie via console*

Cette fonction est utilisée dans le cas où l'utilisateur presse le bouton « Send » situé sous la console afin d'envoyer un message à l'ATH via cette dernière.

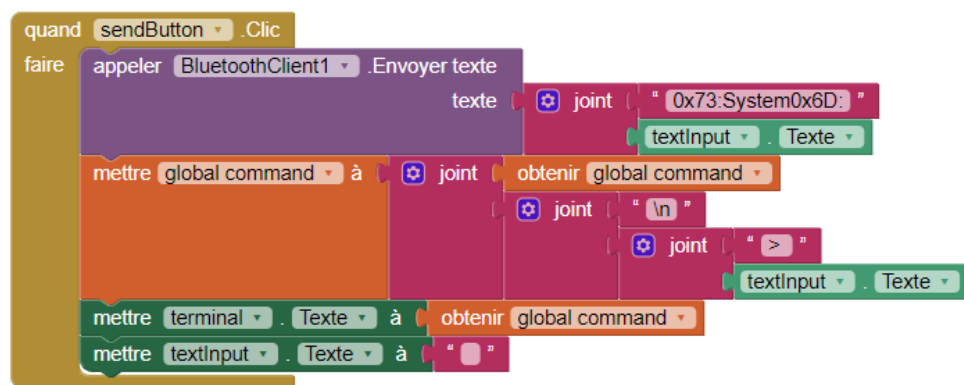
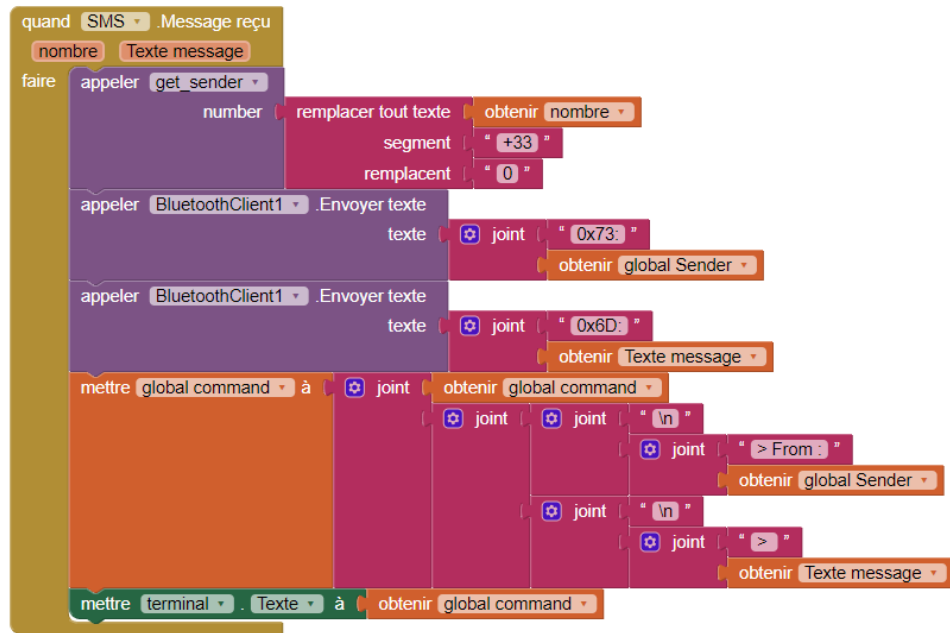


Figure 49 – Application envoie message console

La commande consiste à récupérer le texte entré par l'utilisateur dans la zone de texte, et de le compléter avec les Clés d'identification vues dans la programmation de l'Arduino. Etant donné que dans notre cas nous considérons les messages de la console comme des SMS envoyés par système, on rentre le nom de ce dernier directement à la suite de la Clé d'identification du « Sender ». On rajoute ensuite le contenu du message à la variable « command » qui est affichée en continue dans la console et qui permet ainsi un suivi des communications.

➤ *Réception SMS*

Cette fonction est déclenchée à la réception de chaque SMS, et a pour but d'encoder le message et le nom de l'émetteur selon le protocole établi et de l'envoyer à l'ATH pour affichage.



Durant son exécution, elle va faire appel à la fonction « get sender » afin de récupérer le nom de l'émetteur du SMS qu'elle enverra avec le message au système. Elle rajoutera ces informations à la variable `command` pour qu'elles soient affichées dans la console.

➤ *Fonction « get_sender »*

Appelée lors de l'exécution de la réception SMS, cette fonction a pour but de chercher dans l'annuaire de l'utilisateur, le nom de la personne ayant envoyé le message grâce à son numéro de téléphone.

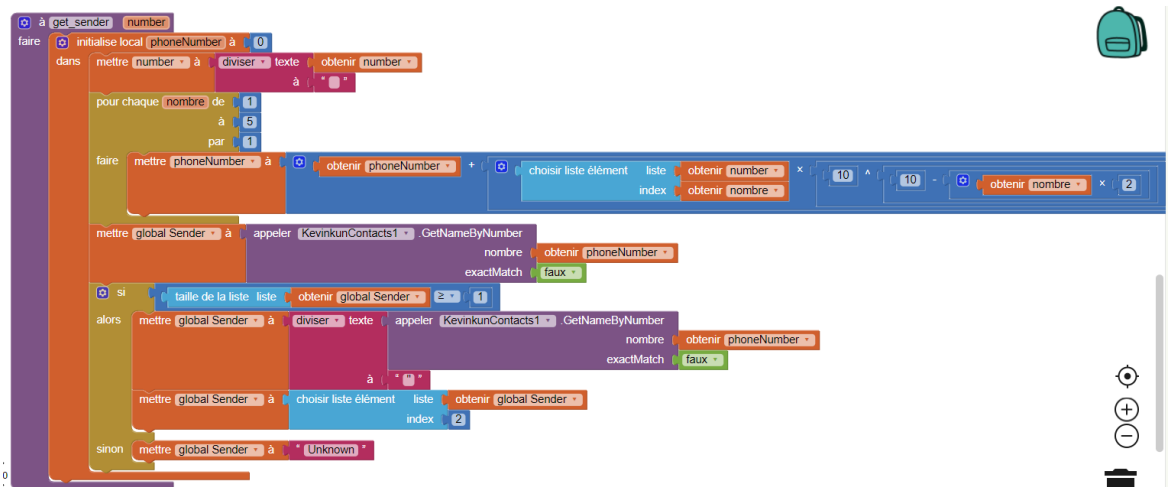


Figure 51 – Application « get_sender »

La première partie de cette fonction consiste à convertir la variable de type String contenant le numéro de téléphone en une variable int traitable par la bibliothèque qui effectue la recherche. Si un numéro est trouvé, la bibliothèque nous retourne une variable du type String semblable à : `[["NAME"],["NUM"]]`. Si aucun contact n'est trouvé, la bibliothèque retourne une variable vide. Dans le premier cas, on divise la variable en une liste en se servant du guillemet comme élément séparateur, et on récupère l'élément d'indice 2, soit le nom de la personne que l'on écrit dans la variable global « Sender ». Dans le deuxième cas, on modifie la valeur de cette variable en « Unknown ».

Ainsi, la combinaison de notre programmation de l'Arduino et celle de l'application, nous permettent de faire fonctionner notre système comme demandé par le cahier des charges.

La réalité augmentée, bien qu'étant une technologie se développant de plus en plus, reste une invention récente. Il y a peu d'études sur les risques et conséquences de son utilisation au quotidien.

On peut supposer qu'il y a des risques du côté social, tel qu'un renfermement sur soi ou une isolation, mais bien moindre que ceux liés à la réalité virtuelle.

D'après un rapport de l'ANSES (l'Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail), dans un rapport sorti en 2021, la réalité augmentée pourrait causer certains maux divers et variés, tel que des vertiges, de la dépendance, des effets neurologiques et sur le développement.

La cybercinétose est également évoquée, il s'agit d'une perturbation de la perception visuelle pouvant être comparée au mal des transports.

Malheureusement, le manque d'études scientifiques sur le sujet nous empêche de tirer de vraies conclusions.

Une étude menée par le Vias Institut, a pour but d'observer les « Effets de l’Affichage Tête Haute intégré dans le casque » (sur un simulateur moto et avec le dispositif EyeLight). Lors de cette étude plusieurs paramètres sont étudiés, sur un petit groupe d'individus, comme : la distraction pendant la conduite, le champ de vision, le confort par exemple.

Il s'avère que les résultats sont très positifs pour ce dispositif en particulier, le EyeLight. Toutefois ils tiennent à mettre en garde les utilisateurs car il y a peu de réglementation quant à cette nouvelle technologie ; il est important de mener des études sur le fonctionnement à la fois cognitif et contextuel des utilisateurs.

En conclusion trop peu de recherches ont été menées pour que nous puissions vraiment connaître quels pourraient être les effets néfastes de l'exposition à la réalité augmentée.

Conclusion

Bien que l’Affichage Tête Haute soit un concept apparu au cours des années 40, il reste encore aujourd’hui très peu développé, et son application dans notre vie de tous les jours est encore au stade de prototype. Au cours de ce projet, nous avons pu en apprendre plus sur l’histoire de l’ATH et sur les différents systèmes en cours de démocratisation.

Nous avons pu également mettre en œuvre et approfondir nos connaissances en optique et informatique, afin de réaliser un prototype fonctionnel de notre ATH. Cette étude et nos cas pratiques nous ont également permis de déterminer des trajectoires d’amélioration pour notre système à commencer par la gestion de plus de données, qui nécessitera surement de se tourner vers un microprocesseur ayant une plus grande mémoire dynamique.

Bibliographie

EyeLights. 2022. EyeLights | GPS Affichage tête haute. *EYELIGHTS*. [En ligne] 2022. [Citation : 15 04 2022.] <https://eye-lights.com/>.

2022. Google Glass. *Wikipedia*. [En ligne] 21 05 2022. [Citation : 15 04 2022.] https://en.wikipedia.org/wiki/Google_Glass.

Jérôme. 2021. Afficheur OLED i2c Arduino (avec contrôleur SSD1306 monochrome 128×64 pixels) : branchement, librairie, fonctions graphiques, et exemples de code. *Passion Electronique*. [En ligne] 12 09 2021. [Citation : 09 05 2022.] <https://passionelectronique.fr/ecran-oled-i2c-arduino/>.

2022. Microsoft HoloLens. *Wikipedia*. [En ligne] 10 05 2022. [Citation : 15 04 2022.] https://en.wikipedia.org/wiki/Microsoft_HoloLens.

Mojo Vision. 2022. Mojo Vision. *Mojo*. [En ligne] 2022. [Citation : 15 04 2022.] <https://www.mojo.vision/>.

Velleman nv. 2021. 0.96 INCH OLED SCREEN WITH I2C FOR ARDUINO ®. [En ligne] 21 12 2021. [Citation : 17 05 2022.] https://www.velleman.eu/downloads/29/vma438_a4v01.pdf.

—. **2019.** BLUETOOTH® HC-05 TRANSMISSION MODULE. [En ligne] 08 05 2019. [Citation : 17 05 2022.] https://www.velleman.eu/downloads/29/vma302_a4v02.pdf.

VITURE: Next Gen XR Glasses. *VITURE*. [En ligne] [Citation : 15 04 2022.] <https://www.viture.com/>.

Xukyo. 2019. Votre Arduino communique avec le module HC-05. *Aranacorp*. [En ligne] 31 07 2019. [Citation : 17 05 2022.] <https://www.aranacorp.com/fr/votre-arduino-communique-avec-le-module-hc-05/>.

[En ligne] / aut. FORM Athletica, Inc // FORM Athletica. - 2022. - 15 04 2022. - <https://www.formswim.com/blogs/media?page=3>.

À propos de HoloLens 2 [En ligne] / aut. Microsoft // Microsoft. - 11 05 2022. - 15 05 2022.

À quoi sert un séparateur de faisceau ? [En ligne] / aut. Staff Moyens // moyens.net. - 02 2022. - 15 04 2022. - <https://www.moyens.net/strategie/a-quoi-sert-un-separateur-de-faisceau-5-faits-en-bref/>.

Affichage tête haute [En ligne] // Wikipedia. - 02 12 2021. - 03 03 2022. - https://fr.wikipedia.org/wiki/Affichage_t%C3%AAte_haute.

Augmented Reality Display Glasses [En ligne] / aut. BAE System // BAE System. - <https://www.baesystems.com/en/product/ar-glasses>.

AVIS de l'Agence nationale de sécurité sanitaire de l'alimentation, de l'environnement et du travail relatif aux « effets sanitaires liés à une exposition aux technologies de réalité virtuelle et/ou de réalité augmentée » [Rapport] / aut. Anses. - Maisons-Alfort : [s.n.], 2021.

collimator [En ligne] / aut. Gregersen Erik // Britannica. - The Editors of Encyclopaedia Britannica . - 02 05 2022. - <https://www.britannica.com/technology/collimator>.

Effets de l’Affichage Tête Haute intégré dans le casque [Rapport] / aut. Aline Delhayé Sofie Boets, Stéphane Espié, Martijn Teuchies. - [s.l.] : Institut Vias –Centre de Connaissance Sécurité Routière, 2021.

Filtre dichroïque [En ligne] // Wikipedia. - 06 05 2022. - 15 04 2022. - https://fr.wikipedia.org/wiki/Filtre_dichro%C3%AFque.

Google Glass: It's all about the Eyepiece Stupid [En ligne] / aut. Purcher Jack // Patently Mobile. - 21 03 2013. - 10 05 2022. - <https://www.patentlymobile.com/2013/03/google-glass-its-all-about-the-eyepiece-stupid.html>.

HoloLens 2 En savoir plus sur les fonctionnalités de HoloLens 2 et consulter les spécifications techniques. [En ligne] / aut. Microsoft // Microsoft. - 2022. - 12 05 2022. - <https://www.microsoft.com/fr-fr/hololens/hardware>.

How does Google Glass project the image onto the glass? [En ligne] / aut. Missfeldt Martin // Quora. - 02 2013. - 02 05 2022.

<https://www.edmundoptics.com/knowledge-center/application-notes/optics/what-are-beamsplitters/> [En ligne] / aut. Edmund Optics Inc. // Edmund Optics . - Edmund Optics Inc., 2022. - 15 04 2022. - <https://www.edmundoptics.com/knowledge-center/application-notes/optics/what-are-beamsplitters/>.

Introduction à la polarisation [En ligne] / aut. Edmund Optics // Edmund Optics. - 2022. - 15 05 2022. - <https://www.edmundoptics.fr/knowledge-center/application-notes/optics/introduction-to-polarization/>.

Introduction To Splitters [En ligne] / aut. Teledyne Photometrics. // Teledyne Photometrics.. - 2022. - 15 04 2022.

Mini Écran OLED Transparent 128x64 SparkFun Qwiic [En ligne] // Robotshop. - 2020. - 18 05 2022. - <https://www.robotshop.com/eu/fr/mini-ecran-oled-transparent-128x64-sparkfun-qwiic.html>.

Miroir semi-réfléchissant [En ligne] // Wikipedia. - 10 04 2022. - 12 05 2022. - <https://www.formswim.com/blogs/media?page=3>.

Prisms [En ligne] / aut. NITTO OPTICAL CO., LTD. // NITTO OPTICAL CO., LTD.. - 16 04 2022. - https://www.nitto-optical.co.jp/english/products/basic_prism/index.html.

Qu'est-ce qu'une lentille holographique ? [En ligne] / aut. Spiegato // Spiegato. - 2022. - 13 05 2022. - <https://spiegato.com/fr/quest-ce-quune-lentille-holographique>.

Réalité virtuelle, réalité augmentée : quels risques ? quelles bonnes pratiques adopter ? [En ligne] / aut. Anses // anses. - 24 06 2021. - 03 05 2022. - <https://www.anses.fr/fr/content/r%C3%A9alit%C3%A9-virtuelle-r%C3%A9alit%C3%A9-augment%C3%A9e-quels-risques-queles-bonnes-pratiques-adopter>.