GNG 2501

Manuel d’utilisation et de produit pour le projet de conception Occupation de salle de bain

SENSORPRO BY FLAMA SENSOR Inc.

Soumis par:
Flama Sensor Inc. FA1-3
Lili Rose Théoret , 300342096

Thillo Aïssata Ameth Gaye, 300287192

Fatoumata Sarr, 300356079

Matéo Blais, 300266658

Mohamed Luc Aurel Degnon, 300260558

Le 2 décembre 2024
Université d’Ottawa

Table des matières

Table des matières i

Liste de figures iii

Liste de tableaux 3

1 Introduction 1

2 Aperçu 3

2.1 Conventions 4

2.2 Mises en garde & avertissements 4

3 Pour commencer 4

3.1 Considérations pour la configuration 6

3.2 Considérations pour l'accès des utilisateurs 7

3.3 Accéder/installation du système 8

3.4 Organisation du système & navigation 8

3.5 Quitter le système 9

4 Utiliser le système 10

4.1 <Entrer dans la salle de bain> 10

4.2 <Sortir dans la salle de bain> 10

5 Dépannage & assistance 10

5.1 Messages ou comportements d'erreur 11

5.2 Entretien 11

5.3 Assistance 11

6 Documentation du produit 13

6.1 Sous-système à l'intérieur de la salle de bain 13
6.1.1 NDM (Nomenclature des Matériaux) 13
Tableau 1 : Nomenclature des matériaux et des composantes du sous-système intérieur 20
6.1.2 Liste d'équipements 13
6.1.3 Instructions 15
6.2 Sous-système à l'extérieur de la salle de bain 18
6.2.1 NDM (Nomenclature des Matériaux) 18
Tableau 2 : Nomenclature des matériaux et des composantes du sous-système extérieur 23
6.2.2 Liste d'équipements 19
6.2.3 Instructions 20
6.3 Essais & validation 23
7 Conclusions et recommandations pour les travaux futurs 25
8 Bibliographie 27
APPENDICES 28
9 APPENDICE I: Fichiers de conception 28
10 APPENDICE II: Autres Appendices 28
10.1 Annexe 1 – Code pour l'ESP32 du sous-système intérieur 28
10.2 Annexe 2 – Code pour l'ESP32 du sous-système extérieur 31

Liste de figures

Figure 1. Sensor Pro en fonctionnement avec ces deux dispositifs interne et externe 3

Figure 2. Emplacement suggérer du dispositif 6

Figure 3. Plan prévisualisé de découpe laser du détecteur intérieur 15

Figure 4. Plan mise à jour sur AutoCAD 16

Figure 5. Boîtier électrique intérieur 17

Figure 6. Résultat de la soudure du dispositif intérieur 18

Figure 7. Plan de découpe laser sur MakerCase 20

Figure 8. Plan mise à jour sur AutoCAD 21

Figure 9. Résultats de la soudure du dispositif externe 22

Liste des Tableaux

Tableau 1. Nomenclature des matériaux et des composantes du sous-système intérieur

Tableau 2. Nomenclature des matériaux et des composantes du sous-système extérieur

Tableau 3.

Tableau 4.

1 Introduction

Le manuel d'utilisation et de produit pour le SensorPro, conçu par Flama Sensor Inc., répond à un besoin spécifique identifié dans des environnements tels que les écoles pour personnes sourdes ou malentendantes. Ces utilisateurs rencontrent des défis liés à l'indication d'occupation des salles de bain, en raison de mécanismes de verrouillage automatiques peu adaptés à leur accessibilité. Ce projet vise à fournir une solution intuitive, sécurisée et facile à installer qui signale de manière claire et visuelle l'occupation des toilettes, tout en garantissant une confidentialité et une sécurité optimales.

Le développement de ce dispositif repose sur plusieurs hypothèses. Premièrement, le système doit être automatique, sans intervention directe des utilisateurs, et accessible à une variété de publics, tout en étant particulièrement utile aux personnes malentendantes. Deuxièmement, sa conception doit minimiser les coûts tout en garantissant une fiabilité élevée et une installation non intrusive. Enfin, la solution doit être facilement intégrable dans des infrastructures existantes.

Le document est structuré comme suit : dans un premier temps la section *Aperçu* décrit les conventions de conception et les avertissements de sécurité liés à l'installation et à l'utilisation. La partie *Pour Commencer* fournit des instructions pour la configuration et l'accès, suivie de la section *Utilisation*, qui détaille les fonctionnalités du système. Une section est dédiée à la *Documentation du produit*, expliquant en détail le processus de fabrication et les choix techniques. Enfin, la section *Dépannage et assistance* guide l'utilisateur pour résoudre les problèmes éventuels et accéder au support technique, tandis que la *Conclusion* propose des pistes d'améliorations.

L'objectif principal de ce manuel est de guider les futurs utilisateurs dans l'installation et la maintenance du SensorPro. Ce document s'adresse principalement aux responsables techniques et

aux utilisateurs particuliers. Ce manuel vise à offrir une expérience utilisateur optimale tout en respectant les normes de sécurité et d’accessibilité pour répondre aux besoins diversifiés des utilisateurs.

2 Aperçu

Au Consortium Centre Jules-Léger il y a une porte de toilette qui se verrouille automatiquement lorsqu'elle se ferme et les employés doivent l'ouvrir avec une clé. Dans une école de personnes sourdes ou malentendantes ça ne sert à rien de cogner à la porte pour savoir s'il y a quelqu'un à l'intérieur et cela devient gênant lorsqu'il y a une personne sourde dedans.

La tâche qui nous a été donné était de concevoir un dispositif qui permet d'indiquer s'il y a vraiment quelqu'un à l'intérieur. De plus, il doit y avoir une sorte d'indication à l'extérieur de la salle pour indiquer l'occupation. Cette indication doit être accessible pour des personnes malentendantes. Il doit être intuitif et ne doit pas empêcher la porte de fermer correctement et il doit aussi être facile à installer et à maintenir.

Notre solution apporte une simplicité d'opération et de conception en réduisant les coûts et en utilisant peu de composantes tout en remplissant son rôle fondamental de détection de présence et de signal instantané à l'indication extérieur pour la préservation de l'intimité.

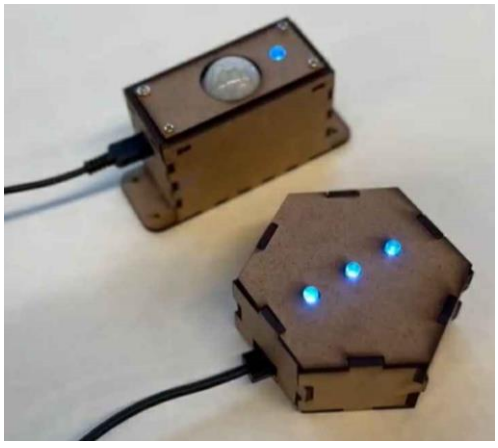


Figure **1**. Sensor Pro en fonctionnement avec ces deux dispositifs interne et externe

2.1 Conventions

Notre projet ne comporte pas de fonction de commande car il est automatique aussi loin qu'il est fonctionnel d'où son utilité. En effet il détecte du mouvement et le signale automatiquement en allumant les LEDs à l'extérieur et pour ce faite les utilisateurs n'ont rien de particulier à faire de particulier.

2.2 Mises en garde & avertissements

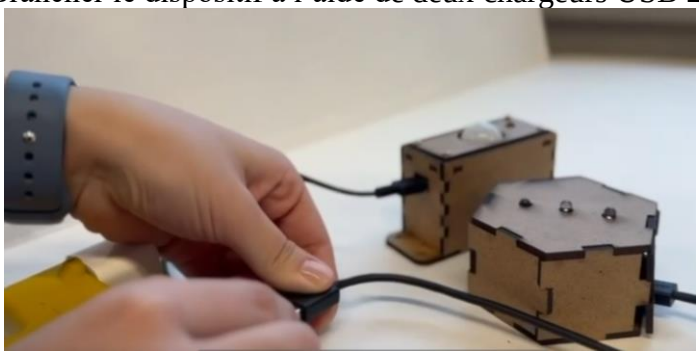
Notre produit comporte des restrictions importantes pour garantir la sécurité de l'utilisateur. Nous recommandons d'installer les boîtiers intérieur et extérieur au plafond de la salle de bain pour éviter toute manipulation risquée, ce qui pourrait altérer son fonctionnement.

De plus, une LED dans le boîtier intérieur indique l'état du détecteur de mouvement : elle s'allume lorsqu'il est fonctionnel. Si la LED est éteinte, cela signifie qu'il y a un problème (mauvais branchement, défaillance du détecteur, etc.), et l'utilisateur pourra signaler ce dysfonctionnement pour une intervention rapide.

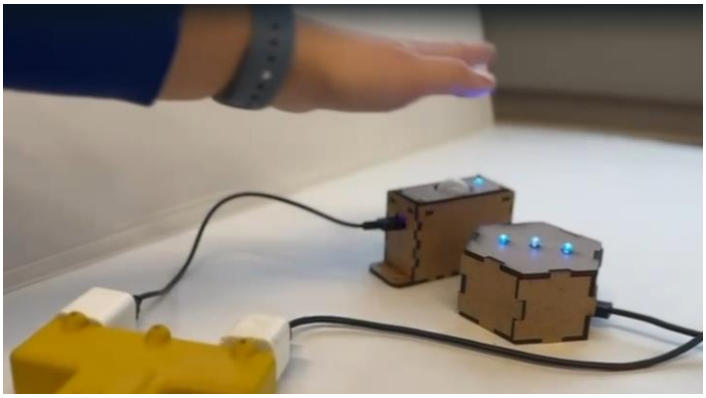
3 Pour commencer

Voici ci-dessous une présentation générale étape par étape de l'utilisation du dispositif

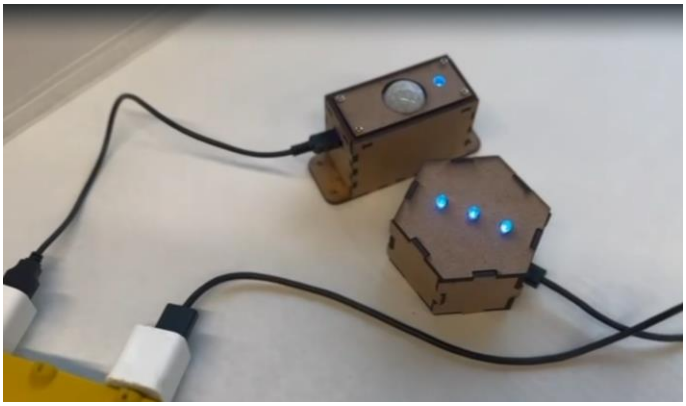
Étape 1 : Brancher le dispositif à l'aide de deux chargeurs USB 2.0 Micro B



Étape 2 : Tester le détecteur de mouvement HC-SR501 PIR



Étape 3 : Si les lumières s'allument vous pouvez installer les deux dispositifs aux endroits qui vous conviennent



3.1 Considérations pour la configuration

Le prototype SensorPro est conçu pour être installé dans une salle de bain. Il se compose de deux boîtiers : un boîtier intérieur et un boîtier extérieur, à installer au plafond. Le boîtier intérieur est connecté à un détecteur de mouvement et contient une LED qui sert à indiquer le bon fonctionnement du dispositif. Si la LED est allumée, cela signifie que le détecteur est fonctionnel. Si elle est éteinte, cela peut indiquer un problème, comme un branchement incorrect, un détecteur défectueux ou une panne d'alimentation.

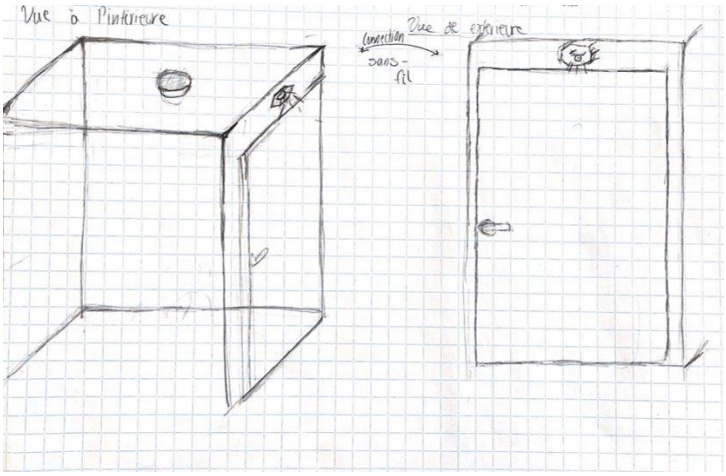


Figure 2. Emplacement suggérer du dispositif

Pour l'installation, il est recommandé de fixer ces boîtiers aux endroits appropriés au plafond de manière à éviter toute manipulation accidentelle du dispositif, ce qui pourrait nuire à son bon fonctionnement ou causer des risques pour l'utilisateur. Les connexions entre les boîtiers doivent être effectuées avec soin, en s'assurant que tout est bien raccordé avant de tester le système.

La communication entre le boîtier intérieur et le boîter extérieur se fait via des microcontrôleurs ESP32, soit un dans chaque boîtier. Ces microcontrôleurs communiquent via Bluetooth. Le microcontrôleur se situant dans le boîter intérieur reçoit du détecteur de mouvements les données d'entrée, donc la présence de quelqu'un dans la salle de bain. Lorsque quelqu'un se trouve dans la salle de bain, il allume la LED du boîtier intérieur. En tout temps, ces données d'entrée sont envoyées au microcontrôleur situé dans le boîtier extérieur, qui se charge alors d'allumer les LEDs à l'extérieur lorsque qu'un mouvement est détecté dans la salle de bain. Dans chaque boîtier, la communication des données se fait par des fils électriques. Toutes les composantes sont reliées entre elles par ces fils et par de la soudure à l'étain sans plomb.

3.2 Considérations pour l'accès des utilisateurs

Le produit est accessible à tous types d'utilisateurs. Bien qu'il fût conçu principalement pour des personnes malentendantes, SensorPro peut satisfaire tout public car il vient à point nommé pour sauver du temps à toute personne qui veut urgemment utiliser une salle de bain car ce dernier a la possibilité de détecter son occupation (ou sa non-occupation) a une distance raisonnable de la salle de bain à travers le signal lumineux. Le SensorPro peut aussi être très utile

pour détecter l'occupation des places parking et faciliter la vie des utilisateurs de manières diverses et variées dans l'exercice de leurs tâches quotidiennes.

3.3 Accéder/installation du système

SensorPro n'est pas une application en ligne à laquelle nos utilisateurs doivent s'inscrire. Il arrive déjà programmer en fonction des exigences de la clientèle et est aussitôt prêt à être utiliser après avoir été installer.

3.4 Organisation du système & navigation

Composantes Mecaniques principales :

- Le boitier intérieur MDF, contenant le détecteur de mouvement et une LED d'état
- Le boitier extérieur MDF, avec les 3 LEDs d'indications
- Les connexions sans fils entre les deux boîtiers via bluetooth grâce au ESP32

Accessoires et pièces jointes :

- Câbles USB pour l'alimentation
- Résistances et LEDs
- Vis de fixation pour les boitiers aux murs ou plafond

Connexions aux fonctionnalités principales :

- Le détecteur de mouvement dans le boîtier intérieur est connecté grâce à des fils électriques au premier ESP32

- Les résistances et les LEDs du boîtier extérieur sont connectées au second ESP32 avec des fils électriques

Composantes Logiciels principales :

- Un système automatique sans intervention manuelle
- Les microcontrôleurs ESP32 gèrent la détection et l'envoi des données

Cheminement des fonctionnalités :

Détection de mouvement → Allumage de la LED intérieure → Transmission Bluetooth au boîtier extérieur → Allumage des LED extérieures.

3.5 Quitter le système

De manière générale, le SensorPro demeure actif et ne nécessite aucune action particulière pour être mis en veille.

Cependant il peut s'ajuster à des systèmes de sorte qu'il soit possible d'être mis en mode repos. Par exemple dans une salle de bain s'il est alimenté par le même système que la lumière centrale, il ne sera pas opérationnel à chaque fois que la lumière sera éteinte.

4 Utiliser le système

Les sous-sections suivantes fournissent des instructions détaillées, étape par étape, sur la façon d'utiliser le dispositif Sensor Pro. Étant donné qu'il s'agit d'un prototype totalement automatique les seules interactions sont de rentrer et sortir de la pièce pour permettre la détection.

4.1 <Entrer dans la salle de bain>

Ici l'utilisateur doit rentrer dans la salle de bain et se tenir en dessous du capteur et grâce au mouvement si la DEL à l'intérieur s'allume cela veut dire que l'indicateur extérieur d'occupation est allumé aussi, et que le système fonctionne durant la durée de la minuterie. Cette dernière est remise à zéro à chaque fois qu'un autre mouvement est détecté pour assurer la continuité de l'occupation même dans les moments où un mouvement n'est pas détecté.

4.2 <Sortir dans la salle de bain>

Ensuite quand l'utilisateur à terminer, il peut se laver les mains et le détecteur captera les mouvements relatifs à ce déplacement également s'il est bien placé au centre du plafond. Seulement lorsque l'utilisateur sort de la salle de bain et que le détecteur ne "voit" plus rien que les DEL s'éteindront au bout de la minuterie personnalisée par le client.

5 Dépannage & assistance

Dans le cas où le système ne fonctionne pas c'est à dire que la LED connectée au détecteur de mouvement ne s'allume pas, essayer de redémarrer le système d'alimentation du SensorPro. Si le problème persiste, veuillez contacter l'équipe d'installation du Produit SensorPro afin que le panne soit détectée et réglée.

5.1 Messages ou comportements d’erreur

Défense de manipuler, secouer ou ouvrir le dispositif car cela risquerait de déconnecter les branchements faits en interne. Pour parer à ces éventualités le dispositif doit être hors de portée des enfants et positionné en hauteur pour parrer à toute manipulation accidentelle. Le dispositif intérieur possède une LED qui indiquent lorsque son état de fonctionnement pour une intervention rapide en cas de panne.

5.2 Entretien

L'entretien du système de détection de mouvement développé pour les salles de bain est conçu pour être simple et ne requiert pas beaucoup d'efforts :

Nettoyage léger : Un simple nettoyage à l'aide d'un chiffon doux suffit pour enlever les poussières sur le capteur, à effectuer une fois par mois. Cela garantit que le capteur reste sensible aux mouvements.

Vérification visuelle : Une fois par trimestre, il est conseillé de faire une vérification rapide des connexions pour s'assurer que les câbles ne sont ni débranchés ni endommagés.

Test ponctuel : Vous pouvez vérifier que le système fonctionne bien en simulant un mouvement devant le capteur, simplement pour vous assurer que tout fonctionne correctement. Cela peut se faire de manière occasionnelle, sans fréquence définie.

5.3 Assistance

Pour garantir une assistance rapide et efficace, voici comment les utilisateurs peuvent obtenir de l'aide :

Support Technique : En cas de besoin d'assistance technique, contactez l’équipe de support à l'adresse suivante : support@flamasensor.com. Le responsable du support est Thillo Aïssata A. Gaye.

Support de Production : Pour toute question sur la production ou l'entretien, Mateo Blais est la personne à contacter à l'adresse : production@flamasensor.com.

Gestion des Incidents de Sécurité : En cas d’incident de sécurité, tel qu'une détection incorrecte, informez rapidement l’administration via les contacts fournis pour garantir une intervention immédiate.

6 Documentation du produit

6.1 Sous-système à l'intérieur de la salle de bain

6.1.1 NDM (Nomenclature des Matériaux)

Tableau 1 : Nomenclature des matériaux et des composantes du sous-système intérieur

Item n°	Qté	Description	Matériau	Masse	Fournisseur	Prix
1.	1	HC-SR501 Pir Motion IR Sensor (5x): Détecteur de mouvement qui s'étend sur 15 pied.	Ampoule, incandescent, board	1,58 onces	Amazon : Lien	13,99$ + tx
2.	½	Matériau MDF pour les boîtes	MDF	Environ 200g	MakerStore	2$
3.	1	ESP32 WROOM-32E	Board	9g	MakerStore	12$
4.	½	Paquet de 10 câbles mâle-femelle	Câble	1g	MakerStore	0,50$
5.	1	Câbles UBS-USB A micro	Câble	15g	Maison	1$
6.	1	Adaptateur Light Bulb Socket – USB	Plastique	42,52g	Amazon: Lien	17,99$ + tx
Total						51,64$ tx incluses

6.1.2 Liste d'équipements

- 1 détecteur de mouvement HC-SR501
- Matériau MDF
- Machine de découpe laser
- 1 microcontrôleur ESP32
- 5 câbles mâle-femelle

- 1 LED bleue
- 1 résistance de 220 Ω
- Matériel de soudure
    - Fer à souder
    - Étain sans plomb
    - Pince coupante
    - Troisième main
    - Tapis de soudage
    - Extracteur pour soudage
    - Laine de laiton
- 8 vis cruciforme
    - 4 pour visser le couvercle au reste du boitier
    - 4 pour visser le sous-système au plafond
- 1 câble USB-USB a micro
- 1 adaptateur Light Bulb Socket – USB
- Arduino IDE
- AutoCAD
- Inkscape

Il a été considéré d'utiliser du plastique pour imprimer en 3D le boitier intérieur. Cependant, ce matériau ne permet pas d'utiliser des vis pour fixer le sous-système en place. Le matériau MDF permet l'utilisation de vis, raison pour laquelle il a été choisi à la place du plastique pour impression 3D.

6.1.3 Instructions

6.1.3.1 Partie mécanique

Après plusieurs itérations avec le MakerCase du MakerSpace on s'est rendu compte qu'il ne remplissait pas tous les besoins nécessaires pour le détecteur intérieur alors nous avons utilisé Boxe.py qui est un générateur de boîtes open source et qui convient parfaitement à nos besoins.

1. Ouvrir le générateur de boîte Boxe.Py et sélectionner "Boîte Électrique"
2. Paramétrer les options comme suit :
   - Dimensions en millimètres : **X = 70** ; **Y = 30** ; **H = 48**
   - **Décocher** la case "**extérieur**" car on veut des dimensions en mm intérieurs *(Note : ceci est très important car sinon la boite sera plus petite !)*
   - 10 mm pour les côtés des triangles supportant le couvercle
   - 1mm pour le diamètre des trous de vis du couvercle intérieur
   - 2mm pour le diamètre des trous de vis du couvercle
   - 2mm pour le diamètre des trous de vis pour le montage
   - Cocher la case pour ajouter des points de montage externe
   - Mettre une distance des trous à votre convenance mais j'ai utilisé 7mm
   - Aller dans réglages par défaut et sélectionner d'enregistrer le fichier au format DXF *(Note : avant d'enregistrer le fichier vous pouvez le prévisualiser avec le format SVG)*

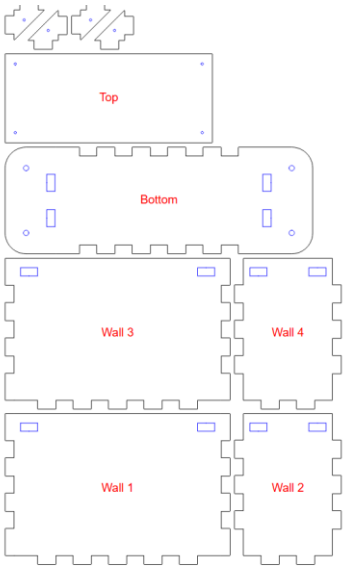


Figure 3. Plan prévisualisé de découpe laser du détecteur intérieur

3. Une fois que tout est paramétrer cliquer sur "Générer" et enregistrer le fichier DXF

4. Mais d’abord il faut convertir le fichier DXF en DWG pour que AutoCAD puisse le lire grâce à un convertisseur en ligne
5. Utiliser AutoCAD pour ajouter le trou de 12 mm de diamètre sur le couvercle pour laisser passer le détecteur de mouvement et le trou pour de 12 x 7 mm pour le câble d’alimentation sur l’un des deux murs

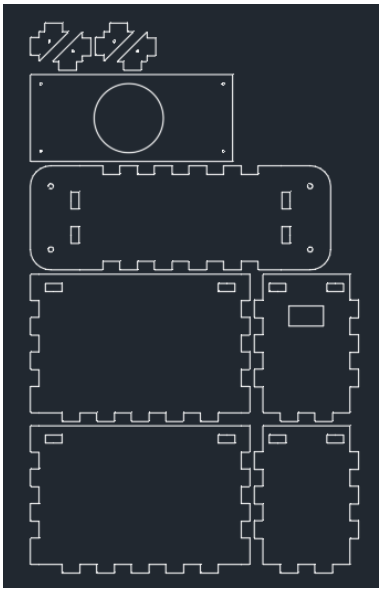


Figure 4. Plan mise à jour sur AutoCAD

6. Lorsque les changements ont été fait il faut reconvertir le DWG en DXF pour pouvoir l’importer sur Inkscape
7. **Important** : il faut sélectionner tous les traits et changer leurs épaisseurs à **0.001 inch** (ceci est une étape cruciale pour la découpe laser)
8. Enregistrer au format PDF 1.4
9. Ouvrir le fichier sur l’ordinateur connecté à la découpeuse laser et imprimer pour le lui envoyer afin de commencer la découpe.

Figure 5. Boîtier électrique intérieur

(Pour de plus amples détails sur la procédure pour Inkscape et l'utilisation de la découpe laser *Epilog Mini 24* ou *Epilog Helix* visiter le mode d'emploi du MakerSpace : Laser Cutting Beginner)

6.1.3.2 Partie électronique

1. Couper un câble mâle-femelle en trois pour avoir les deux extrémités raccourcies et la longueur inutile séparée. Seules les deux extrémités seront utilisées pour la suite, jeter la longueur inutile. Répéter 5 fois.
2. Souder la patte courte de la DEL bleue à une extrémité mâle découpée à l'étape 1.
3. Connecter cette extrémité à une pin GND du ESP32.
4. Souder la patte longue de la DEL bleue à une des pattes de la résistance.
5. Souder l'autre patte de la résistance à une extrémité mâle découpée à l'étape 1.
6. Connecter cette extrémité à une pin GPIO du ESP32. Les instructions de ce document utilisent le GPIO 23.
7. Avec les extrémités restantes découpées à l'étape 1, les souder pour former trois câbles raccourcis.
8. Connecter l'extrémité d'un câble de l'étape 7 à la pin VCC du détecteur de mouvements, et l'autre extrémité à la pin 3.3V du ESP32.
9. Connecter l'extrémité d'un câble de l'étape 7 à la pin GNG du détecteur de mouvements, et l'autre extrémité à une pin GNG du ESP32.
10. Connecter l'extrémité d'un câble de l'étape 7 à la pin OUT du détecteur de mouvements, et l'autre extrémité à une pin GPIO du ESP32. Les instructions de ce document utilisent le GPIO 13.
11. Brancher l'extrémité USB a micro du câble USB-USB a micro dans le ESP32, et l'extrémité USB dans l'ordinateur muni de Arduino IDE.

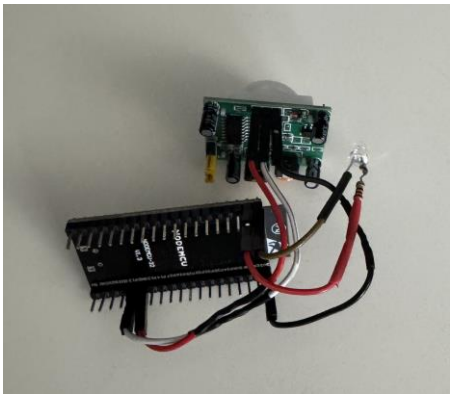


Figure 6. Résultat de la soudure du dispositif intérieur

(Les étapes suivantes se font sur le Arduino IDE)

12. Aller dans les réglages de Arduino IDE, et ajouter le manager URL du ESP32 : https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
13. Créer un nouveau sketch vide.
14. Choisir comme Board le ESP32 Dev Module.
15. Programmer le ESP32 : (pour avoir le code détaillé, voir l'annexe 1)
    a. Fonction BT_Status() qui permet d'afficher via les DELs intégrés au ESP32 et via le moniteur en série si la connexion Bluetooth est établie
    b. Fonction SlaveConnect() qui permet de connecter le ESP32 en tant que master au ESP32 du sous-système extérieur.
    c. Fonction set() qui permet de fixer les paramètres et d'initialiser la connexion Bluetooth.
    d. Fonction loop() qui permet de recevoir les données d'entrée du capteur de mouvements et de les envoyer au sous-système extérieur via Bluetooth, et d'allumer la DEL en conséquence.

6.2 Sous-système à l'extérieur de la salle de bain

6.2.1 NDM (Nomenclature des Matériaux)

Tableau 2 : Nomenclature des matériaux et des composantes du sous-système extérieur

Item n°	Qté	Description	Matériau	Masse	Fournisseur	Prix
2.	½	Matériau MDF pour les boîtes	MDF	Environ 200g	MakerStore	2$
3.	1	ESP32 WROOM-32E	Board	9g	MakerStore	12$
4.	½	Paquet de 10 câbles mâle-femelle	Câble	1g	MakerStore	0,50$

5.	1	Câbles UBS-USB A micro	Câble	15g	Maison	1$
Total						15,50$ tx incluses

6.2.2 Liste d'équipements

- Matériau MDF
- Machine à découpe laser
- 1 microcontrôlleur ESP32
- 2 câbles mâle-femelle
- 3 LED bleue
- 3 résistance de 220 Ω
- Matériel de soudure
    - Fer à souder
    - Étain sans plomb
    - Pince coupante
    - Troisième main
    - Tapis de soudage
    - Extracteur pour soudage
    - Laine de laiton
- 1 câble USB-USB a micro
- 1 petite plaque perforée verte
- Arduino IDE
- AutoCAD
- Inkscape

6.2.3 Instructions

6.2.3.1 Partie mécanique

1. Ouvrir le générateur de boite MakerCase et sélectionner l'option boite à polygone
2. Insérer les dimensions suivantes dans chacune des boites et assurer vous d'être changé les unités en milimetre:
   Sides : 6
   Inside Diameter: 65 mm
   Inside Height: 55mm
   Materiel Thickness; 3mm
   Open or Closed: Closed
   Edge Joints: Finger
   Finger size: 12.4
3. Télécharger le ficher sous forme DXF en cliquant sur *Download Box Plans.*

   Voici à quoi votre fichier devrait ressembler :

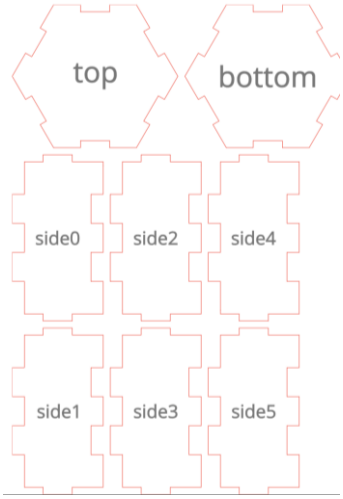


Figure 7. Plan de découpe laser sur MakerCase

4. Convertissez le ficher de DXF à DWG grâce à un convertisseur en ligne.
5. Importer le fichier DWG dans le logiciel AutoCad.
6. **IMPORTANT :** Une fois le dessin ouvert dans Autocad assurer vous d'enlever les mots écrits sur chacune des pièces sinon ils se retrouveront sur les pièces une fois qu'elles seront coupées au laser.

7. Ajouté un cercle de 5mm de diamètre au centre de la pièce du haut (forme hexagonale), ensuite ajouté en deux autres à une distance de 1,5cm.
8. Ajouté un rectangle de 12 x 7 mm pour le câble d'alimentation sur l'un des côtés.

   Voici à quoi votre sketch sur le logiciel Autocad devrait ressembler :

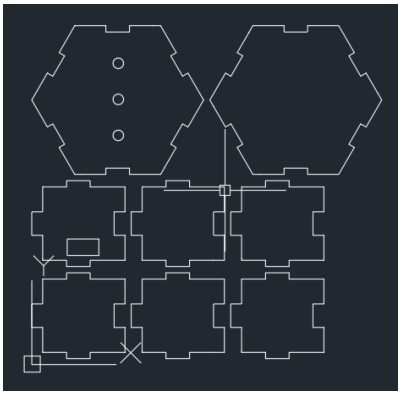


Figure 8. Plan mise à jour sur AutoCAD

9. Reconvertir le DWG en DXF pour pouvoir l'importer sur Inkscape
10. **Important** : il faut sélectionner tous les traits et changer leurs épaisseurs à **0.001 inch** (ceci est une étape cruciale pour la découpe laser)
11. Enregistrer au format PDF 1.4
12. Ouvrir le fichier sur l'ordinateur connecté à la découpeuse laser et imprimer pour le lui envoyer afin de commencer la découpe.
13. Une fois les pièces coupées utilisé de la colle pour bois afin de pouvoir former le boitier

6.2.3.2 Partie électronique

1. Couper un câble mâle-femelle en trois pour avoir les deux extrémités raccourcies et la longueur inutile séparée. Seules les deux extrémités seront utilisées pour la suite, jeter la longueur inutile. Répéter 2 fois.
2. Placer les 3 DELs et les 3 résistances en alternance sur la plaque perforée verte. S'assurer que les pattes longues des DELs soient toutes du même côté.
3. Souder chaque patte longue de chaque DEL à la patte de la résistance voisine.
4. Souder toutes les pattes courtes des DELs ensemble pour qu'elles soient en parallèle. Garder le bout d'une patte libre.
5. Souder le bout de la patte libre d'une DEL de l'étape précédente à une extrémité mâle découpée à l'étape 1.
6. Connecter l'extrémité mâle à une pin GND du ESP32.

7. Souder toutes les pattes libres des résistances ensemble pour qu'elles soient en parallèle. Garder le bout d'une patte libre.
8. Souder le bout de la patte libre d'une résistance de l'étape précédente à une extrémité mâle découpée à l'étape 1.
9. Connecter l'extrémité mâle à une pin GPIO du ESP32. Les instructions de ce document utilisent le GPIO 12.
10. Brancher l'extrémité USB a micro du câble USB-USB a micro dans le ESP32, et l'extrémité USB dans l'ordinateur muni de Arduino IDE.

Figure 9. Résultats de la soudure du dispositif externe

(Les étapes suivantes se font sur le Arduino IDE)

11. Aller dans les réglages de Arduino IDE, et ajouter le manager URL du ESP32 (si ce n'est pas déjà fait) : https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json
12. Créer un nouveau sketch vide.
13. Choisir comme Board le ESP32 Dev Module.
14. Programmer le ESP32 : (pour avoir le code détaillé, voir l'annexe 2)
    a. Fonction BT_Status() qui permet d'afficher via les DELs intégrés au ESP32 et via le moniteur en série si la connexion Bluetooth est établie
    b. Fonction set() qui permet de fixer les paramètres et d'initialiser la connexion Bluetooth.
    c. Fonction loop() qui permet de recevoir les données du système intérieur via Bluetooth et d'allumer et d'éteindre les DELs en conséquence.

6.3 Essais & validation

Tableau 3. Essais

Test	Simulations d'essais	Résultats attendus	Résultats obtenus	Commentaires
Test de la sensibilité du capteur	Positionner le capteur PIR à différentes distances (1 m, 2 m, 2,5 m, 3 m, 5 m) et vérifier la détection avec différentes vitesses et angles de mouvement.	Le capteur doit détecter efficacement les mouvements jusqu'à 3 m de distance	Le capteur détecte les mouvements humains efficacement jusqu'à 2,5 m avec une fiabilité de 90 %. Au-delà de 2,5 m, fiabilité réduite à 70 %.	La détection est peu fiable pour des objets lancés dans le champ du capteur, surtout au-delà de 33 cm. La vitesse de mouvement influence légèrement la détection (les mouvements très lents sont moins détectés), mais pour des mouvements humains normaux, la détection reste efficace.
Test de la connexion capteur-DEL	Déclencher un mouvement et vérifier l'allumage immédiat de la DEL bleue, avec des répétitions pour tester la cohérence de la réponse	La DEL bleue doit s'allumer instantanément après chaque détection	La DEL s'allume immédiatement après chaque détection ; cohérence observée dans les résultats du moniteur série	Une légère latence est présente lors de l'extinction de la DEL entre deux détections consécutives. Une optimisation du code pourrait réduire cette latence pour une meilleure réactivité du système.
Test de la durée de détection	Détecter un mouvement continu pour des périodes	La DEL doit rester allumée pendant toute la durée de	La DEL reste allumée pour une période comprise entre 42 et 43	Le capteur devient inactif pendant 42 à 43 secondes après une détection initiale, puis se réactive. Ce comportement pose

	prolongées (41 s, 1 min, 2 min) et vérifier la durée affichée dans le moniteur série	détection et s'éteindre une fois la période écoulée	secondes avant de s'éteindre, indépendamment des mouvements en cours	un problème de latence, notamment en fin de période, qui pourrait causer des interruptions gênantes pour l'utilisateur.
Test de la distance entre les deux sous-systèmes	Détecter un mouvement lorsque les deux sous-systèmes sont séparés par une certaine distance (< 1 m, 2 m, 5 m, 5 m avec mur)	La DEL bleue doit s'allumer quasi instantanément peu importe la distance qui la sépare du détecteur de mouvements	La DEL s'allume immédiatement après chaque détection, qu'il y ait de la distance (jusqu'à 5 mètres) ou un mur séparant les sous-systèmes	La connexion Bluetooth semble rester stable, peu importe la distance entre 0 et 5 mètres qui sépare les deux modules ESP32, et ce, qu'il y ait un mur ou non. Si l'un ou les deux sous-systèmes sont éteints puis rallumés, la connexion se remet rapidement en place.

7 Conclusions et recommandations pour les travaux futurs

Lors du développement de notre prototype, nous avons acquis des compétences et des connaissances essentielles, comme l'utilisation d'une découpe laser pour les boîtiers, la soudure des composants électroniques sans plaque de prototypage, ainsi que la maîtrise de la connexion Bluetooth avec des microcontrôleurs ESP32. Ces expériences nous ont permis de mieux comprendre les défis de conception et d'intégration technologique, ainsi que l'importance de choisir des composants adaptés à nos besoins spécifiques. Par exemple, l'apprentissage du soudage a permis une intégration plus efficace des LED et des résistances, tandis que la compréhension de la connexion Bluetooth a amélioré la communication entre les modules internes et externes du système

.

- Travaux Futurs:

Pour améliorer le prototype et maximiser son potentiel, voici quelques recommandations de travaux futurs :

1. Amélioration de la Durée de Vie des Composants : Le choix des matériaux pour les boîtiers (MDF) pourrait être revue pour une option plus durable, comme du plastique recyclé ou de l'aluminium. Cela améliorerait la résistance à l'humidité et la durabilité dans le temps, surtout dans des environnements comme les salles de bain.
2. Implémentation de Fonctionnalités de Sécurité : Le système pourrait être enrichi avec une fonctionnalité de détection de sécurité avancée. Par exemple, l'intégration d'un détecteur de chaleur ou de fumée permettrait de donner une alerte en cas de température

anormalement élevée, ajoutant une dimension de sécurité supplémentaire pour les utilisateurs.

3. Développer une Application Mobile : Une application mobile dédiée permettrait aux utilisateurs de configurer le dispositif et de recevoir des notifications en temps réel sur l'état d'occupation de la salle de bain, améliorant ainsi l'expérience utilisateur.
4. L'intégration d'une alimentation par énergie solaire aurait été un ajout innovant permettant de rendre le produit plus éco-responsable et d'éliminer la nécessité d'une alimentation électrique externe, en particulier dans des environnements bien éclairés.

Ces recommandations et pistes futures permettraient non seulement d'améliorer la fiabilité et la fonctionnalité du produit, mais également de proposer une meilleure expérience utilisateur, garantissant un système plus robuste et plus adapté aux divers besoins des utilisateurs.

8 Bibliographie

Générateur de plan de boite Box Py :

https://boxes.hackerspace-bamberg.de/?language=fr

Guide de Découpe Laser CEED :
https://en.wiki.makerepo.com/wiki/Digital_technologies/Laser_cutting/Laser_cutting-_Beginner

APPENDICES

9 APPENDICE I: Fichiers de conception

Tableau 4 : Documents référencés

Nom du document	Emplacement du document et/ou URL	Date d'émission
Page MakerRepo	https://makerepo.com/Mateobl24/2273.gng-2501-flama-sensor-inc	
Page Box.Py	https://boxes.hackerspace-bamberg.de/ElectronicsBox?language=fr	
CEED How to Laser Cut	https://en.wiki.makerepo.com/wiki/Digital_technologies/Laser_cutting/Laser_cutting-_Beginner	

10 APPENDICE II: Autres Appendices

10.1 Annexe 1 – Code pour l'ESP32 du sous-système intérieur

```
#include "BluetoothSerial.h"                                    // BT: Include the Serial bluetooth library
#define LED_BT_BLUE 2                                           // BT: Internal LED (or LED on the pin D2) for the
connection indication (connected LED ON / disconnected LED OFF)
#define LED_BT_RED 15                                           // BT: LED (LED on the pin D4) for the connection
indication (connected LED OFF / disconnected LED ON)
unsigned long previousMillisReconnect;                          // BT: Variable used for comparing millis
counter for the reconnection timer
bool ledBtState = false;                                        // BT: Variable used to chage the indication LED state
bool SlaveConnected;                                            // BT: Variable used to store the current connection
state (true=connected/false=disconnected)
int recatt = 0;                                                 // BT: Variable used to count the reconnection attempts

// BT: Bluetooth availabilty check
#if !defined(CONFIG_BT_SPP_ENABLED)
```

```
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
#endif

String myName = "ESP32-BT-Master";                                  // BT: Variable used to store the
SERVER(Master) bluetooth device name; just for prinitng
String slaveName = "ESP32-BT-Slave";                                // BT: Variable used to store the CLIENT(Slave)
bluetooth device name; just for prinitng; just for printing in this case
String MACadd = "D0:EF:76:56:E3:6E";                                // BT: Variable used to store the CLIENT(Slave)
bluetooth device Mac address; just for prinitng; just for printing in this case
uint8_t address[6] = { 0xD0, 0xEF, 0x76, 0x56, 0xE3, 0x6E };                  // BT: Variable used to store the
CLIENT(Slave) MAC address used for the connection; Use your own andress in the same format

BluetoothSerial SerialBT;                                           // BT: Set the Object SerialBT

int LED_PIN = 23;  // pin LEDs
int PIR_Pin = 13; // pin détecteur de mouvements
bool mouvementDetecte = false; // status de détection de mouvements (true = mouvement détecté ; false = aucun
mouvement détecté)
int sensorValue = 0;  // valeur du capteur (HIGH = mouvement détecté ; LOW = aucun mouvement détecté)



// BT: Bt_Status callback function
void Bt_Status(esp_spp_cb_event_t event, esp_spp_cb_param_t *param) {
  if (event == ESP_SPP_OPEN_EVT) {                                  // BT: Checks if the SPP connection is open, the
event comes// event == Client connected
    Serial.println ("Client Connected");                           // BT: Write to the serial monitor
    digitalWrite (LED_BT_BLUE, HIGH);                               // BT: Turn ON the BLUE bluetooth indication LED
(solid light)
    digitalWrite (LED_BT_RED, LOW);                                 // BT: Turn OFF the RED bluetooth indication LED
    SlaveConnected = true;                                          // BT: Set the variable true = CLIENT is connected to
the SERVER
    recatt = 0;                                                     // BT: Reset the reconnect attempts counter
  }
  else if (event == ESP_SPP_CLOSE_EVT) {                            // BT: event == Client disconnected
    Serial.println("Client Disconnected");                          // BT: Write to the serial monitor
    digitalWrite(LED_BT_RED, HIGH);                                 // BT: Turn ON the RED bluetooth indication LED
(solid light)
    digitalWrite(LED_BT_BLUE, LOW);                                 // BT: Turn OFF the BLUE bluetooth indication
LED
    SlaveConnected = false;                                         // BT: Set the variable false = CLIENT connection lost
  }
}

void setup() {
```

```
  pinMode(LED_PIN, OUTPUT);  // LEDs = output
  pinMode(PIR_Pin, INPUT); // détecteur = input
 pinMode(LED_BT_BLUE, OUTPUT);                                    // BT: Set up the onboard LED pin as output
 pinMode(LED_BT_RED, OUTPUT);                                     // BT: Set up the onboard LED pin as output
 digitalWrite(LED_BT_RED, HIGH);                                  // BT: Turn ON the RED LED = no connection
established
 SlaveConnected = false;                                          // BT: Set the variable false = CLIENT is not connected
 Serial.begin(115200);                                            // Sets the data rate in bits per second (baud) for serial
data transmission

 // BT: Define the Bt_Status callback
 SerialBT.register_callback(Bt_Status);
 // BT: Starts the bluetooth device with the name stored in the myName variable as SERVER(Master)
 SerialBT.begin(myName, true);
 Serial.printf("The device \"%s\" started in master mode, make sure slave BT device is on!\n", myName.c_str());
 SlaveConnect();                                                  // BT: Calls the bluetooth connection function to cnnect
to the CLIENT(Slave)
}

void SlaveConnect() {                                             // BT: This function connects/reconnects to the
CLIENT(Slave)
 Serial.println("Function BT connection executed");                // BT: Write to the serial monitor
 Serial.printf("Connecting to slave BT device named \"%s\" and MAC address \"%s\" is started.\n", slaveName.c_str(),
MACadd.c_str());  // BT: Write to the serial monitor
 SerialBT.connect(address);                                       // BT: Establishing the connection with the
CLIENT(Slave) with the Mac address stored in the address variable
}

void loop() {
 sensorValue = digitalRead(PIR_Pin);  // lire l'input du détecteur

 if (!SlaveConnected) {                                           // BT: Condition to evalute if the connection is
established/lost
   if (millis() - previousMillisReconnect >= 10000) {             // BT: Check that 10000ms is passed
     previousMillisReconnect = millis();                          // BT: Set previousMillisReconnect to current millis
     recatt++;                                                    // BT: Increase the the reconnection attempts counter +1
     Serial.print("Trying to reconnect. Attempt No.: ");          // BT: Write to the serial monitor
     Serial.println(recatt);                                      // BT: Write the attempts count to the serial monitor
     Serial.println("Stopping Bluetooth...");                     // BT: Write to the serial monitor
     SerialBT.end();                                              // BT: Close the bluetooth device
     Serial.println("Bluetooth stopped !");                       // BT: Write to the serial monitor
     Serial.println("Starting Bluetooth...");                     // BT: Write to the serial monitor
     SerialBT.begin(myName, true);                                // BT: Starts the bluetooth device with the name
stored in the myName variable as SERVER(Master)
     Serial.printf("The device \"%s\" started in master mode, make sure slave BT device is on!\n", myName.c_str());
```

```
    SlaveConnect();                                          // BT: Calls the bluetooth connection function to cnnect
to the CLIENT(Slave)
   }

 }
 // BT: Data send/receive via bluetooth
  if (sensorValue == HIGH && !mouvementDetecte) {  // Mouvement détecté
    Serial.println("Mouvement détecté");
    digitalWrite(LED_PIN, HIGH);
    mouvementDetecte = true;  // Mettre le statut de mouvement à détecté
    SerialBT.write(1); // Envoie le signal Bluetooth pour allumer les LEDs
    delay(5000);
   }

   else if (sensorValue == LOW && mouvementDetecte) {  // Aucun mouvement détecté
    Serial.println("Aucun mouvement détecté");
    digitalWrite(LED_PIN, LOW);
    mouvementDetecte = false; // Mettre le statut de mouvement à non-détecté
    SerialBT.write(0); // Envoie le signal Bluetooth pour éteindre les LEDs
   }
 if (SerialBT.available()) {                                 // BT: Checks if there are data from the bluetooth
available
   Serial.write(SerialBT.read());                            // BT: Write the data to the serial monitor

 }
}
```

10.2 Annexe 2 – Code pour l'ESP32 du sous-système extérieur

```
#include "BluetoothSerial.h"                                 // BT: Include the Serial bluetooth library
#define LED_BT 2                                            // BT: Internal LED (or LED on the pin D2) for the connection
indication (connected solid/disconnected blinking)
unsigned long previousMillis;                                // BT: Variable used for comparing millis counter (LED
blinking)
bool ledBtState = false;                                    // BT: Variable used to chage the indication LED state
bool MasterConnected;                                       // BT: Variable used to store the current connection state
(true=connected/false=disconnected)
String device_name = "ESP32-BT-Slave";                          // BT: Variable used to store the CLIENT(slave)
bluetooth device name
String MACadd = "D0:EF:76:56:E3:6E";                        // BT: Variable used to store the CLIENT(slave)
bluetooth Mac address; Use your own MAC address
```

```
// BT: Bluetooth availabilty check
#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run make menuconfig to and enable it
#endif
// BT: Serial Bluetooth availabilty check
#if !defined(CONFIG_BT_SPP_ENABLED)
#error Serial Bluetooth not available or not enabled. It is only available for the ESP32 chip.
#endif

BluetoothSerial SerialBT;                                   // BT: Set the Object SerialBT

int LED_PIN = 12;  // pin LEDs

// BT: Bt_Status callback function
void Bt_Status (esp_spp_cb_event_t event, esp_spp_cb_param_t *param) {

  if (event == ESP_SPP_SRV_OPEN_EVT) {                          // BT: Checks if the SPP Server connection is open, the
event comes
    Serial.println ("Client Connected");                       // BT: Write to the serial monitor
    digitalWrite (LED_BT, HIGH);                               // BT: Turn ON the bluetooth indication LED (solid light)
    MasterConnected = true;                                    // BT: Set the variable true = SERVER is connected to the
CLIENT
  }
  else if (event == ESP_SPP_CLOSE_EVT ) {                       // BT: Checks if the SPP connection is closed, the event
comes
    Serial.println ("Client Disconnected");                    // BT: Write to the serial monitor
    digitalWrite (LED_BT, LOW);                                // BT: Turn OFF the bluetooth indication LED
    MasterConnected = false;                                   // BT: Set the variable false = SERVER connection lost
  }
}

void setup() {
  pinMode(LED_PIN, OUTPUT);  // LEDs = output
  pinMode(LED_BT,OUTPUT);                                      // BT: Set up the onboard LED pin as output
  Serial.begin(115200);                                       // Sets the data rate in bits per second (baud) for serial data
transmission

  // BT: Define the Bt_Status callback
  SerialBT.register_callback (Bt_Status);
  // BT: Starts the bluetooth device with the name stored in the device_name variable
  SerialBT.begin(device_name);
  Serial.printf("The device with name \"%s\" and MAC address \"%s\" is started.\nNow you can pair it with Bluetooth!\n",
device_name.c_str(), MACadd.c_str());
}
```

```
void loop() {
  // BT: Blinking the bluetooth indication LED if the connection is not established
  if (!MasterConnected) {
    if (millis() - previousMillis >= 500) {                    // BT: Checks if 500ms is passed
      if (ledBtState == false) {                               // BT: Checks the leddState and toggles it
      ledBtState = true;
    }
    else {
      ledBtState = false;
    }
    digitalWrite(LED_BT, ledBtState);                          // BT: Set LED ON/OFF based onthe ledBtState variable
    previousMillis = millis();                                 // BT: Set previousMillis to current millis
    }

  }
  // BT: Data send/receive via bluetooth
  if (Serial.available()) {                                    // BT: Checks if there are data from the serial monitor available
    SerialBT.write(Serial.read());                             // BT: Sends the data via bluetooth
  }
  if (SerialBT.available()) {                                  // BT: Checks if there are data from the bluetooth available
    int message = SerialBT.read();

      if (message == 1) { // allumer les LEDs
        Serial.println("LEDs allumées");
        digitalWrite(LED_PIN, HIGH);
      }
      else if (message == 0) {
        Serial.println("LEDs éteintes");
        digitalWrite(LED_PIN, LOW);
      }
  }
}
```