

Présentiel pour openSpace

Table des matières

1	Analyse préliminaire	4
1.1	Introduction	4
1.2	Organisation	4
1.3	Objectifs.....	4
	Planification initiale	6
2	Analyse / Conception.....	7
2.1	Concept	7
2.1.1	Maquette schématique de l'application.....	8
2.1.2	Cas d'utilisation.....	8
2.2	Stratégie de test.....	9
2.2.1	Fonctionnement	9
2.2.2	Erreurs	10
2.2.3	Paramètres de connexion	10
2.3	Planification	10
2.4	Dossier de conception	11
2.4.1	Hardware	11
2.4.2	Software	11
2.4.3	Schéma de montage boîtier	12
2.4.4	Conception du code.....	13
3	Réalisation.....	15
3.1	Dossier de réalisation	15
3.1.1	Projet Arduino.....	16
3.1.2	Projet application	16
3.2	Description des tests effectués.....	16
3.2.1	Méthode.....	17
3.2.2	Presser le bouton.....	17
3.2.3	Changer le statut sur l'application.....	19
3.2.4	Changer le statut Skype	21
3.2.5	Déconnecter Skype	22
3.2.6	Déconnecter le boîtier	23
3.2.7	Déconnecter l'application.....	23

3.2.8	Cliquer sur le message « boîtier déconnecté »	24
3.2.9	Ouvrir la page de paramètres	24
3.2.10	Changer les paramètres	24
3.3	Erreurs restantes	25
3.4	Liste des documents fournis	25
4	Conclusions	26
5	Annexes.....	27
5.1	Résumé du rapport du TPI / version succincte de la documentation	27
5.2	Sources – Bibliographie.....	27
5.3	Journal de travail	27
5.4	Manuel d'Installation	27
5.5	Manuel d'Utilisation.....	27
5.6	Archives du projet.....	27

1 Analyse préliminaire

1.1 Introduction

Ce chapitre décrit brièvement le projet, le cadre dans lequel il est réalisé, les raisons de ce choix et ce qu'il peut apporter à l'élève ou à l'école. Il n'est pas nécessaire de rentrer dans les détails (ceux-ci seront abordés plus loin) mais cela doit être aussi clair et complet que possible (idées de solutions). Ce chapitre contient également l'inventaire et la description des travaux qui auraient déjà été effectués pour ce projet.

Ces éléments peuvent être repris des spécifications de départ.

Prenant peu de place et facilitant la communication entre employés, les open space se répandent. Malheureusement il y a aussi des effets négatifs comme le fait qu'une personne soit trop accaparée par les autres et aie par conséquent moins de temps pour travailler. Ce projet vise à réduire ce souci en proposant un indicateur de disponibilité lumineux lié par Bluetooth au poste de travail de la personne ainsi qu'à son Skype.

Ayant un grand intérêt pour l'électronique et la programmation, ce projet me motive beaucoup. Autre fait motivant :

J'ai pu utiliser un Arduino communiquant avec un PC en Bluetooth lors de mon projet de pré-TPI ce qui me permettra d'aisément mettre en place cette partie du système. La liaison avec Skype est quelque chose d'encore inconnu pour moi, ce qui m'intéresse car nouveau.

1.2 Organisation

Les personnes prenant part au projet sont les suivantes :

Fonction	Nom	E-mail	Téléphone
Candidat	Kalbfuss Cyril	cyril.kalbfuss@cpnv.ch	079 839 39 25
Chef de projet	Ithurbide Julien	julien.ithurbide@cpnv.ch	079 255 67 08
Expert 1	Malherbe Roger	r.malherbe@rmsoft.ch	079 230 72 37
Expert 2	Folomietow Borys	borys@folomietow.ch	076 366 45 06

Le temps nécessaire au développement de certaines parties étant très incertains, je vais travailler avec une méthode agile. Avec le chef de projet, nous fixerons chaque semaine un nouveau sprint.

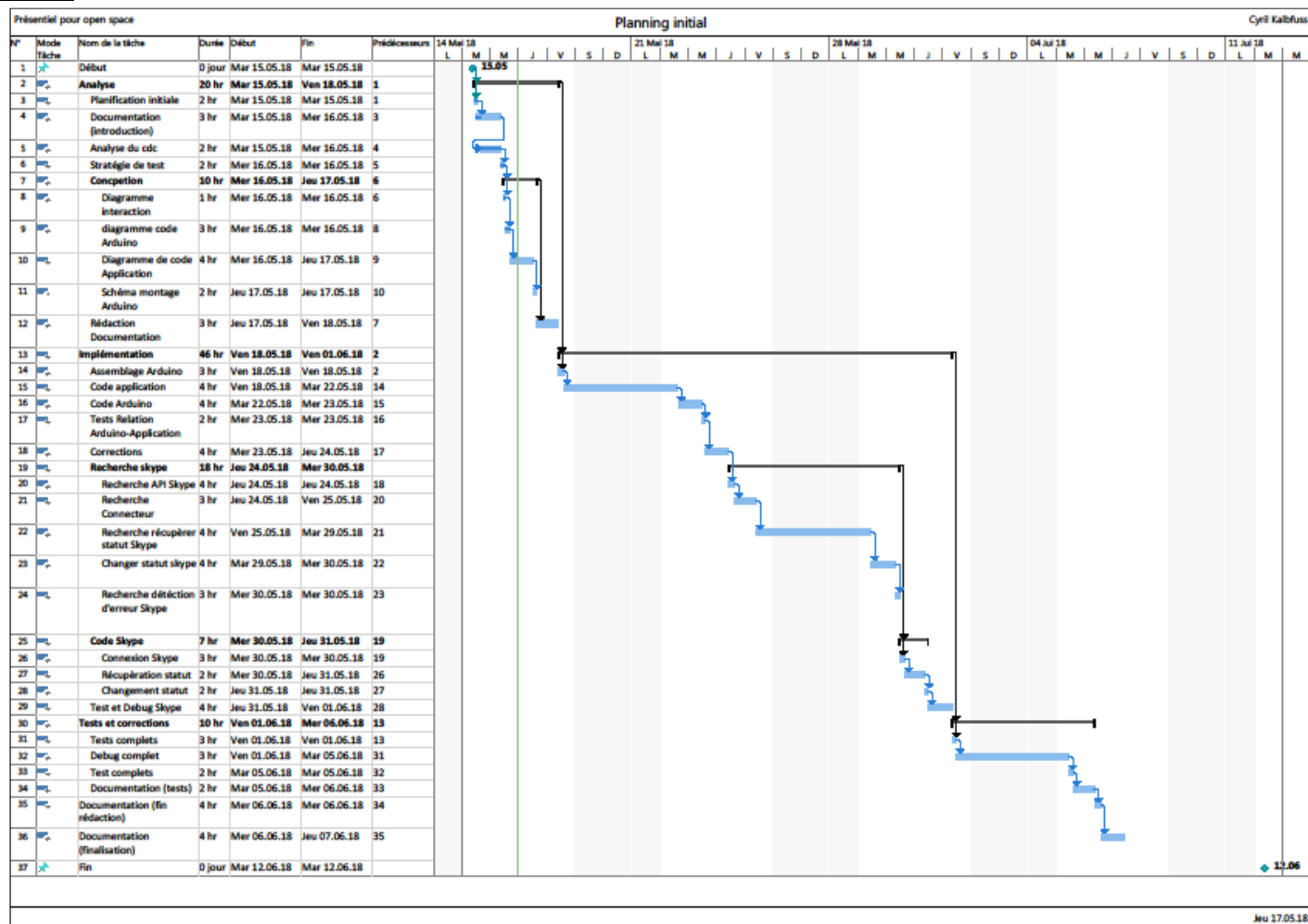
1.3 Objectifs

Ce chapitre énumère les objectifs du projet. L'atteinte ou non de ceux-ci devra pouvoir être contrôlée à la fin du projet. Les objectifs pourront éventuellement être revus après l'analyse.

Ces éléments peuvent être repris des spécifications de départ.

- Le boîtier a une LED qui doit s'allumer en vert, orange ou rouge.
- Un bouton sur le boîtier permet de changer la couleur.
- Une application permet de changer cette couleur.
- L'application peut communiquer en USB avec le boîtier.
- L'application peut communiquer en Bluetooth avec le boîtier.
- L'application a un paramètre pour changer entre USB et Bluetooth.
- L'application affiche la couleur de la LED (Si on change sur le boîtier, ça change sur l'application).
- L'application peut connaître le statut Skype et change la couleur en fonction.

Planification initiale



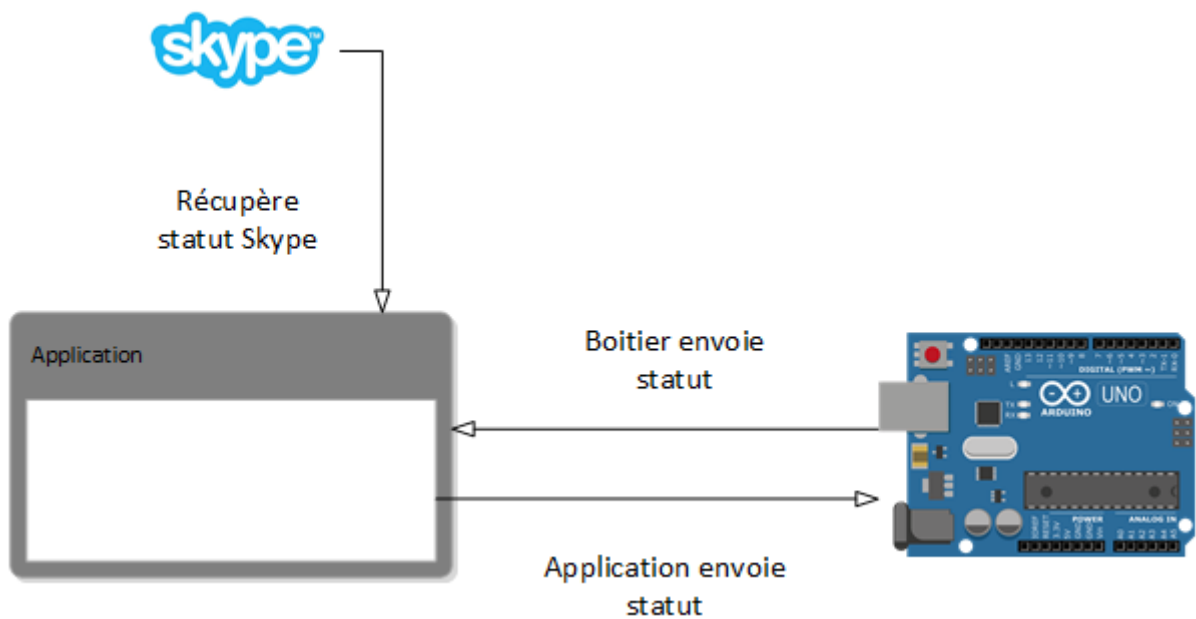
2 Analyse / Conception

2.1 Concept

Le concept complet avec toutes ses annexes:

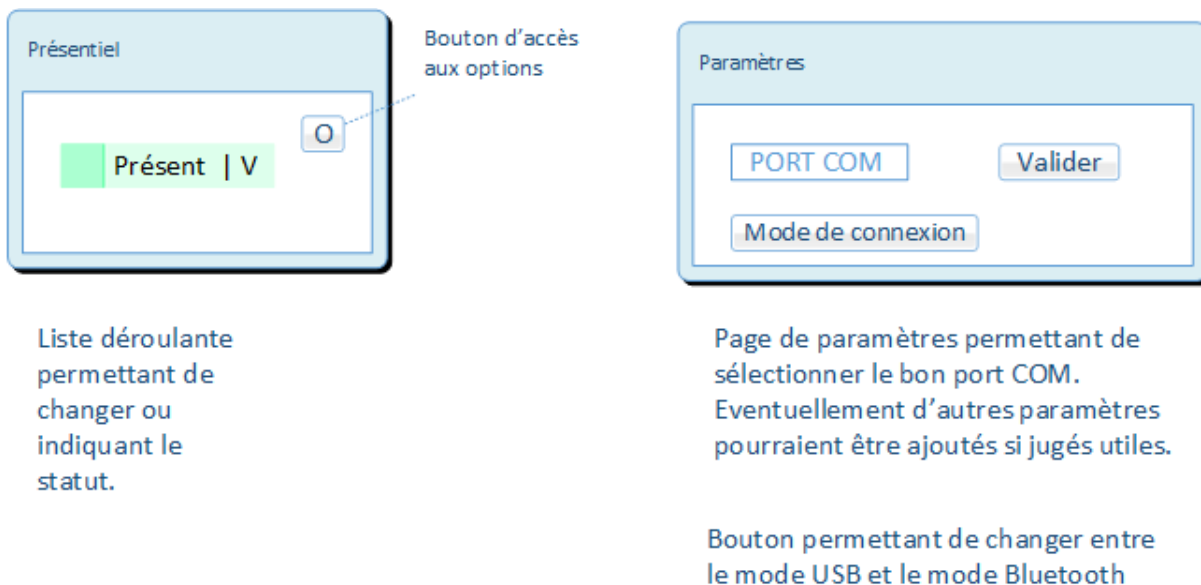
Par exemple :

- *Multimédia: carte de site, maquettes papier, story board préliminaire, ...*
- *Bases de données: interfaces graphiques, modèle conceptuel.*
- *Programmation: interfaces graphiques, maquettes, analyse fonctionnelle...*
- *...*



Ce schéma montre les interactions entre les différents composants du projet. Lors d'un changement de statut côté boitier, l'application adapte son affichage et lors d'un changement côté application, le boitier change sa couleur et Skype modifie son statut. L'application récupère le statut Skype et se met à jour en fonction.

2.1.1 Maquette schématique de l'application



L'application se composera d'une fenêtre principale avec l'indicateur de statut accompagné d'une pastille de la couleur correspondante et d'un bouton pour accéder aux options. L'indicateur sera un menu permettant de changer de statut.

Une fenêtre d'options contenant le choix du mode de connexion (USB ou Bluetooth) et un champ pour indiquer le port COM. D'autres options pourraient être nécessaires au bon fonctionnement du programme et seraient ajoutées ici.

2.1.2 Cas d'utilisation

2.1.2.1 Boitier

Événement	Action
Bouton pressé	La LED passe à la couleur suivante (Bleu->Vert->Orange->Rouge->Bleu...).
	Un message est envoyé à l'application en serial.
Message serial reçu (changement de statut côté application)	La LED passe à la couleur indiquée dans le message.

2.1.2.2 Application

Événement	Action
Changement manuel du statut	L'indicateur change pour afficher le nouveau statut.
	Message serial envoyé au boitier.
	Mise à jour du statut Skype.

Changement des paramètres de connexion	Les paramètres sont enregistrés.
Réception d'un message du boîtier	L'indicateur change pour afficher le nouveau statut. Mise à jour du statut Skype.
Détection changement statut Skype (pas dû à l'application)	L'indicateur change pour afficher le nouveau statut. Message serial envoyé au boîtier.

2.1.2.3 Erreurs

Événement	Action
Côté application : Skype pas connecté	Affiche un message l'indiquant
Côté application : Boîtier pas connecté	Indique que le boîtier n'est pas connecté. Une aide est disponible.
Côté boîtier : Application pas connectée	Indique une couleur standby.

2.2 Stratégie de test

Des tests unitaires seront effectués tout au long du projet pour les différentes fonctionnalités. Des tests complets seront finalement effectués, portant sur le projet fini. Ces tests seront réalisés sur mon poste avec le prototype du boîtier. Un test sur un autre ordinateur sera effectué.

2.2.1 Fonctionnement

Test	Résultat attendu
Presser le bouton du boîtier. <i>Effectuer le test en USB et en Bluetooth.</i> <i>Effectuer le test avec Skype déconnecté.</i>	La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu...).
	Le statut sur l'application change en fonction.
Presser le bouton du boîtier. <i>Effectuer le test en USB et en Bluetooth.</i> <i>Effectuer le test avec Skype connecté.</i>	La LED change de couleur (Bleu->Vert->Rouge->Bleu...).
	Le statut sur l'application change en fonction.
	Le statut de Skype est changé en fonction.
Changer le statut sur l'application. <i>Effectuer le test avec et sans Skype.</i> <i>Effectuer le test en USB et en Bluetooth</i>	La LED prend la couleur correspondante.
	Le statut sur l'application change en fonction.

	Le statut de Skype est changé en fonction, si Skype est connecté.
Changer le statut sur Skype. <i>Effectuer le test avec et sans le boîtier.</i> <i>Effectuer le test en USB et en Bluetooth</i>	La LED prend la couleur correspondante. Le statut sur l'application change en fonction.

2.2.2 Erreurs

Test	Résultat attendu
Déconnecter Skype.	Un message l'indique sur l'application
Déconnecter le boîtier.	Un message indique que le boîtier est déconnecté sur l'application.
Cliquer sur le message précédemment obtenu.	Une page d'aide s'affiche. On peut la refermer.
Déconnecter l'application.	Le boîtier indique la couleur de standby.

2.2.3 Paramètres de connexion

Test	Résultat attendu
Cliquer sur le bouton de paramètres.	Une page de paramètres s'ouvre.
Changer le port COM et fermer la page. <i>Effectuer le test en USB.</i>	Le nouveau port est enregistré. La connexion est relancée.
Changer le port COM et fermer la page. <i>Effectuer le test en Bluetooth.</i>	Le nouveau port est enregistré. La connexion est relancée.
Cliquer sur le bouton de changement USB/Bluetooth.	Le mode de connexion est changé.

2.3 Planification

Révision de la planification initiale du projet :

- *planning indiquant les dates de début et de fin du projet ainsi que le découpage connu des diverses phases.*
- *partage des tâches en cas de travail à plusieurs.*

*Il s'agit en principe de la planification **définitive du projet**. Elle peut être ensuite affinée (découpage des tâches). Si les délais doivent être ensuite modifiés, le responsable de projet doit être avisé, et les raisons doivent être expliquées dans l'historique.*

2.4 Dossier de conception

Fournir tous les document de conception:

- *le choix du matériel HW*
- *le choix des systèmes d'exploitation pour la réalisation et l'utilisation*
- *le choix des outils logiciels pour la réalisation et l'utilisation*
- *site web: réaliser les maquettes avec un logiciel, décrire toutes les animations sur papier, définir les mots-clés, choisir une formule d'hébergement, définir la méthode de mise à jour, ...*
- *bases de données: décrire le modèle relationnel, le contenu détaillé des tables (caractéristiques de chaque champs) et les requêtes.*
- *programmation et scripts: organigramme, architecture du programme, découpage modulaire, entrées-sorties des modules, pseudo-code / structogramme...*

Le dossier de conception devrait permettre de sous-traiter la réalisation du projet !

2.4.1 Hardware

Pour ce projet je dispose des éléments Hardware suivants :

- PC CPNV (8Go RAM, disque dur 512Go)
- Module Bluetooth USB
- Arduino UNO R3
- Module Bluetooth HC-06
- Barrette LED RGB (Adafruit NeoPixels)
- Bouton
- 1x 470Ω résistance
- 3x 1kΩ résistances
- 1x 10kΩ résistance
- 1x 1μF condensateur non polarisé

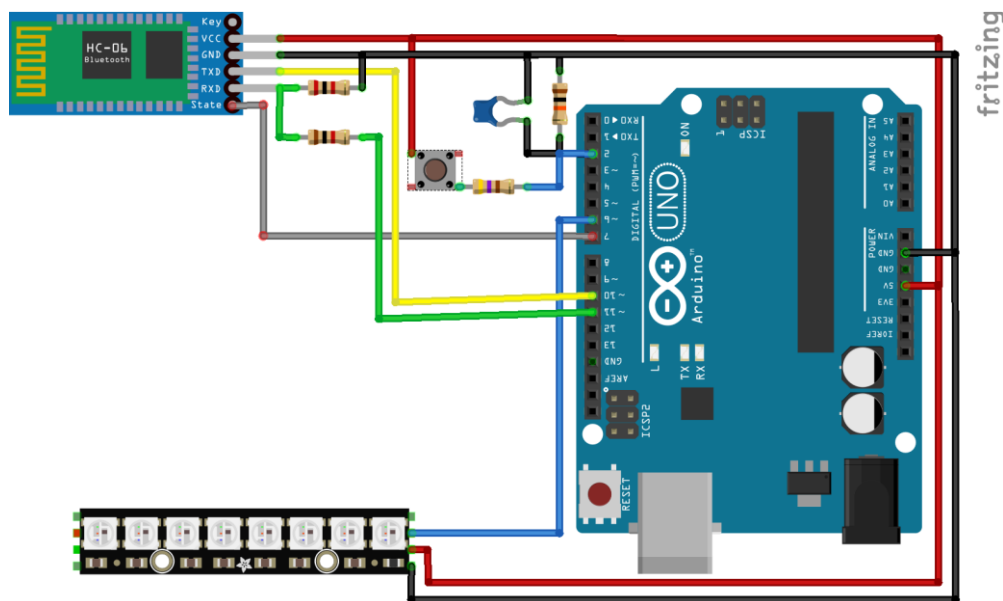
2.4.2 Software

Je dispose d'une machine Windows 7 pour le développement et l'application finale tournera sur cet OS.

L'application sera programmée avec Visual Studio et le code du boîtier sera fait avec l'IDE Arduino.

Skype for business sera utilisé en tant que partie du projet.

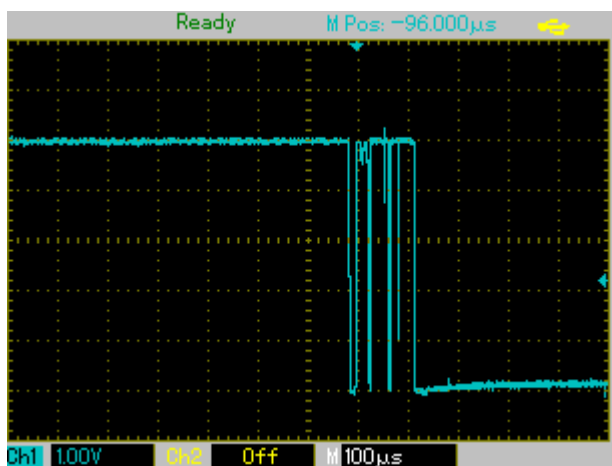
2.4.3 Schéma de montage boîtier



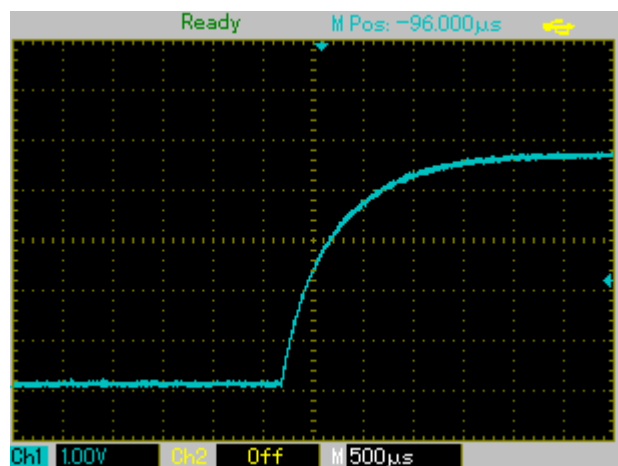
Le module Bluetooth HC-06 est connecté aux pins 10 et 11 car les pins rx/tx 0 et 1 sont utilisés par la connexion USB. La patte Rx du module ne supporte pas plus de 3.3v, c'est pour cela qu'elle est connectée avec un diviseur résistif utilisant trois résistances de 1kΩ. Il est important de noter que le Rx du module va vers le Tx de l'Arduino et le Tx du module va vers le Rx de l'Arduino.

La barrette de LEDs se connecte à l'alimentation, la terre et utilise un pin pour être contrôlée.

Le bouton est connecté à un pin et à l'alimentation ainsi qu'à une résistance 10kΩ en « pull-down » et un filtre passe bas pour filtrer les rebonds (voir images ci-dessous).



Sans filtre

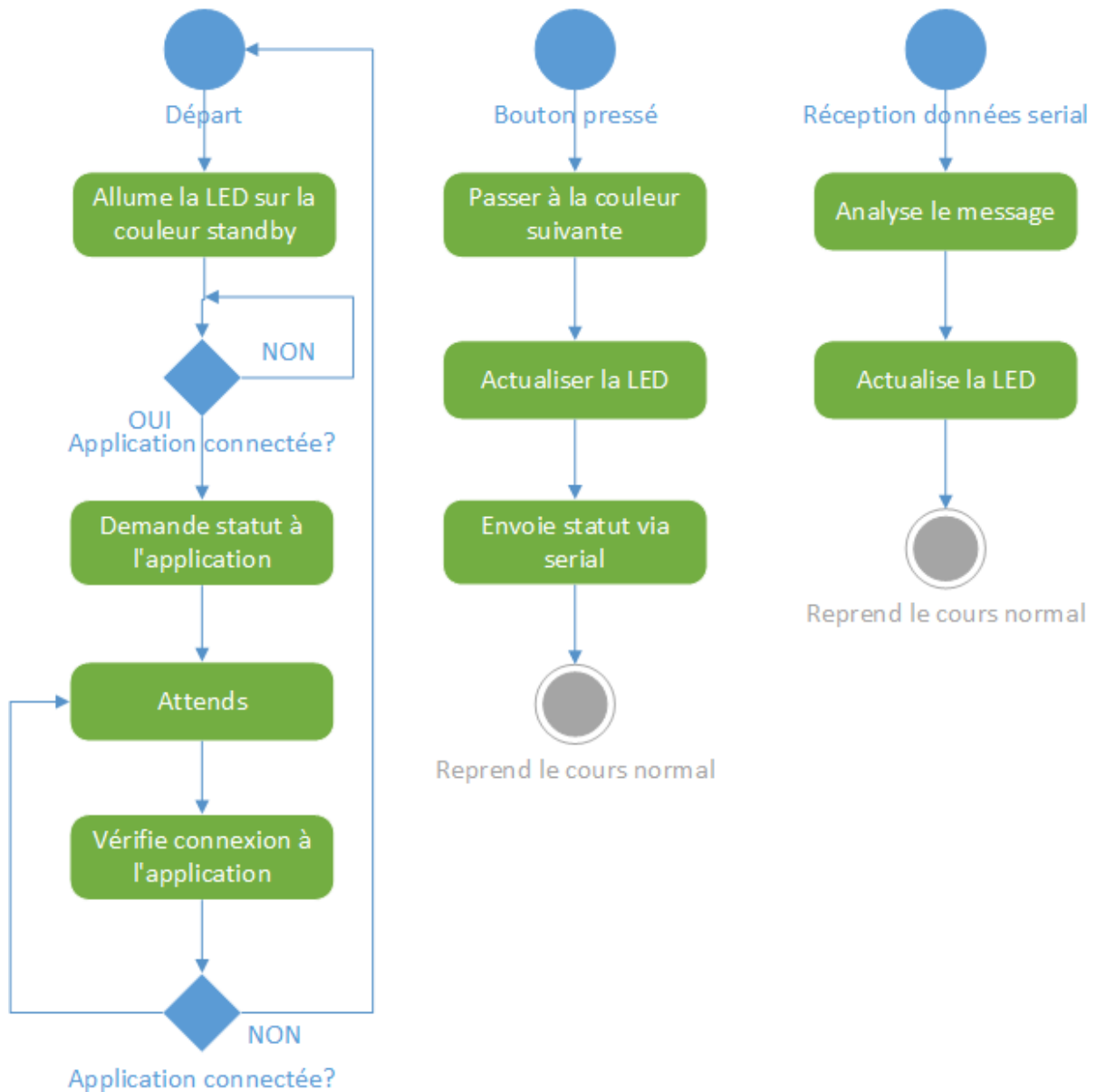


Avec filtre

On voit bien ici les rebonds créés par le mécanisme du bouton (gauche) et l'effet du filtre anti-rebonds (droite). Les rebonds posent le problème suivant : étant donné qu'il y a plusieurs « montées », le programme compte comme plusieurs appuis sur le bouton.

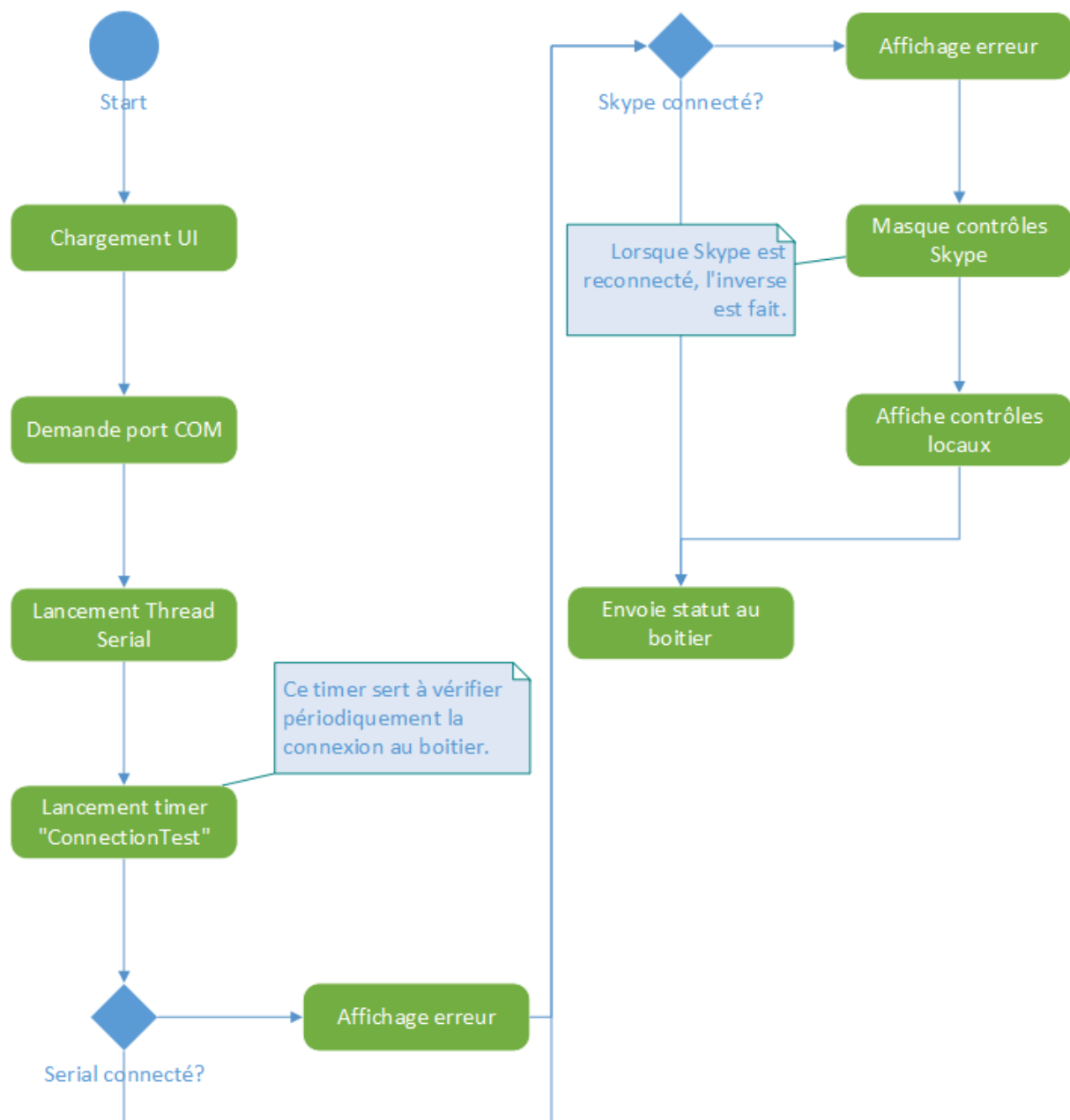
2.4.4 Conception du code

2.4.4.1 Boitier



Lorsque le boitier s'enclenche, il s'allumera avec une couleur d'attente pouvant indiquer qu'il n'y a personne. Si le boitier ne détecte pas de connexion avec l'application, il reste ou revient sur cette couleur. Lorsque quelqu'un pèse sur le bouton ou que le boitier reçoit un message serial, l'attente s'interrompt et le boitier change de couleur.

2.4.4.2 Application



Le logiciel C# sera un projet Visual studio WPF car c'est dans ce type de projet que les contrôles de l'API de Skype for business sont présents. Ceux-ci permettent une interaction quasi-automatique entre l'application et Skype.

La partie principale du programme lance un thread qui gèrera le serial et un qui gèrera Skype. Puis il initialise le statut en fonction des choses connectées.

Les threads serial et Skype enverront des événements lors de la réception d'un message ou d'un changement de statut.

Un timer est également lancé afin de vérifier périodiquement l'état des connexions.

3 Réalisation

3.1 Dossier de réalisation

Décrire la réalisation "physique" de votre projet

- les répertoires où le logiciel est installé
- la liste de tous les fichiers et une rapide description de leur contenu (des noms qui parlent !)
- les versions des systèmes d'exploitation et des outils logiciels
- la description exacte du matériel
- le numéro de version de votre produit !
- programmation et scripts: librairies externes, dictionnaire des données, reconstruction du logiciel - cible à partir des sources.

NOTE : Evitez d'inclure les listings des sources, à moins que vous ne désiriez en expliquer une partie vous paraissant importante. Dans ce cas n'incluez que cette partie...

Lors de ce projet, les divers documents et dossiers (projet de code, etc...) sont enregistrés sur le lecteur réseau dont je dispose et sont régulièrement synchronisés sur GitHub.

Le dossier est organisé de la manière suivante :

Doc	Contient les documents utilisés pour la documentation.
Archives	Contient des sauvegardes de la documentation.
Conception	Contient les documents de conception (pseudo codes, schéma de montage, ...)
Images	Contient les images utilisées pour la documentation.
Planning	Contient les fichiers pour le planning initial et suivants.
Documentation.docx	Fichier Word de la documentation.
JournalTravail.xlsx	Fichier Excel du journal de travail.
Code	Contient les dossiers des projets de code.
Boitier	Contient le projet Arduino IDE du boitier.
Presentiel	Contient le projet Visual Studio de l'application
Légende :	Dossier Fichier

3.1.1 Projet Arduino

La programmation du boîtier a été effectuée avec l'IDE Arduino (version 1.6.5). Le projet ne comporte qu'un fichier de code mais ce dernier utilise deux bibliothèques :

- SoftwareSerial.h
- Adafruit_NeoPixel.h

La première permet de gérer la communication serial et est donc utilisée pour la communication USB et Bluetooth.

La seconde permet de gérer la barre LED et est proposée par le fabricant.

Concernant le boîtier physique, le montage montré dans la partie conception a été suivi et un boîtier imprimé en 3D a été créé sur mesure pour accueillir ce prototype.

L'Arduino doit être alimenté via le câble USB pour fonctionner. Ce dernier ne doit toutefois pas forcément être branché sur un ordinateur : Une batterie externe fonctionne parfaitement.

3.1.2 Projet application

L'application a été développée en C# sous forme d'un projet WPF avec le logiciel Microsoft Visual Studio Entreprise 2017 (Version 15.5.7). Les différents fichiers et dossiers du projet sont organisés sous la forme suivante :

[presentiel.sln](#)

[presentiel](#)

properties	Contient les settings utilisés.
referencies	Contient les bibliothèques importées.
Resources	Contient les fichiers importés en tant que ressource.
utils	Un dossier contenant les classes « utiles » du projet.
serialCom.cs	La classe du gestionnaire de communication serial.
MainWindow.xaml	Fichier de l'UI. Décrit les différents éléments graphiques.
MainWindow.xaml.cs	Fichier du code lié à l'UI.

Pour ce projet, j'ai importé la bibliothèque suivante :

- Microsoft Lync Client SDK

Celle-ci est fournie par Microsoft mais il faut éditer le fichier d'installation fourni car ce dernier ne reconnaît pas Skype for Business en tant que Lync.

En plus de cette bibliothèque, j'utilise une partie des références de base de System et System.Windows. notamment Threading et IO.Ports pour la communication.

3.2 Description des tests effectués

Pour chaque partie testée de votre projet, il faut décrire:

- les conditions exactes de chaque test
- les preuves de test (papier ou fichier)
- tests sans preuve: fournir au moins une description

3.2.1 Méthode

Les tests suivants ont été réalisés sur le PC de développement avec le prototype de boîtier ainsi que l'application lancée depuis Visual Studio. Pour les tests en connexion Bluetooth, le boîtier est branché à une batterie externe au lieu de PC pour s'assurer que ce soit bien ce mode de présentation qui soit utilisé. Des images illustrent les différents états résultants du test effectué.

3.2.2 Presser le bouton

Éléments connectés : Boîtier, Application, Connexion USB

Résultat attendu :

- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).
- Le statut de l'application prend la même couleur et le statut correspondant.

Résultat obtenu :

- Le boîtier se comporte correctement.
- L'application se comporte correctement.



Éléments connectés : Boîtier, Application, Skype, Connexion USB

Résultat attendu :

- La LED change de couleur (Bleu->Vert->Rouge->Bleu->...).
- Le statut de l'application prend la couleur et le statut correspondants (Vert->Rouge->Rouge->Vert->...) Bleu n'étant pas disponible pour Skype.
- Le statut Skype correspond à l'application.

Résultat obtenu :

- Le boîtier se comporte correctement.
- L'application se comporte correctement.

- Skype se comporte correctement.



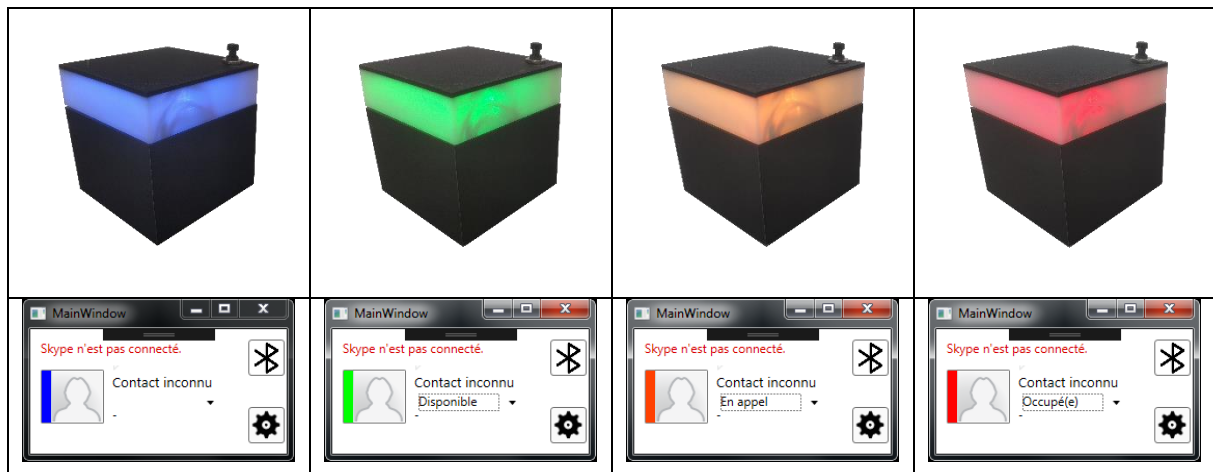
Éléments connectés : Boitier, Application, Connexion Bluetooth

Résultat attendu :

- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).
- Le statut de l'application prend la même couleur et le statut correspondant.

Résultat obtenu :

- Le boitier se comporte correctement.
- L'application se comporte correctement.



Éléments connectés : Boitier, Application, Skype, Connexion USB

Résultat attendu :

- La LED change de couleur (Bleu->Vert ->Rouge->Bleu->...).

- Le statut de l'application prend la couleur et le statut correspondants (Vert->Rouge->Rouge->Vert->...) Bleu n'étant pas disponible pour Skype.
- Le statut Skype correspond à l'application.

Résultat obtenu :

- Le boîtier se comporte correctement.
- L'application se comporte correctement.
- Skype se comporte correctement.



3.2.3 Changer le statut sur l'application

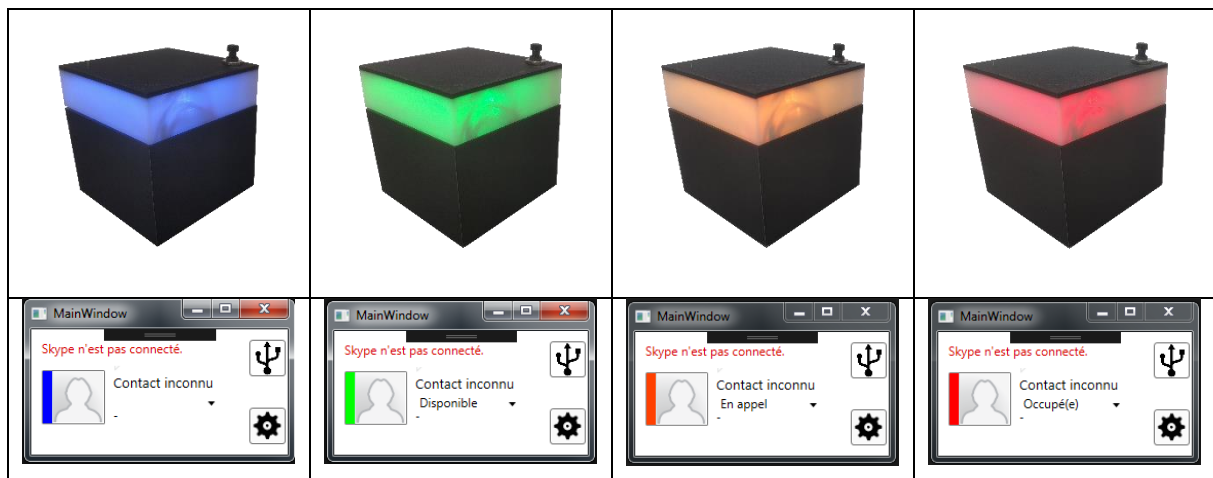
Éléments connectés : Boîtier, Application, Connexion USB

Résultat attendu :

- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).

Résultat obtenu :

- Le boîtier se comporte correctement.



Éléments connectés : Boitier, Application, Skype, Connexion USB

Résultat attendu :

- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).
- Le statut Skype correspond à l'application.

Résultat obtenu :

- Le boitier se comporte correctement.
- Skype se comporte correctement.



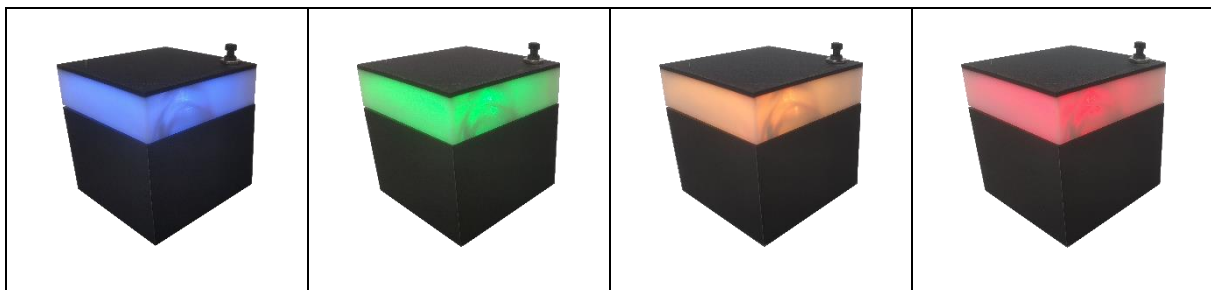
Éléments connectés : Boitier, Application, Connexion Bluetooth

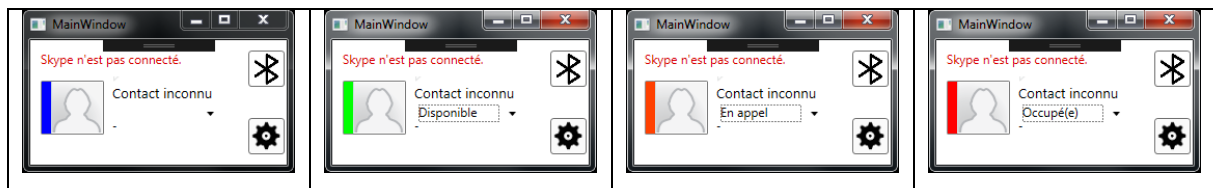
Résultat attendu :

- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).

Résultat obtenu :

- Le boitier se comporte correctement.





Eléments connectés : Boitier, Application, Skype, Connexion Bluetooth

Résultat attendu :

- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).
- Le statut Skype correspond à l'application.

Résultat obtenu :

- Le boitier se comporte correctement.
- Skype se comporte correctement.



3.2.4 Changer le statut Skype

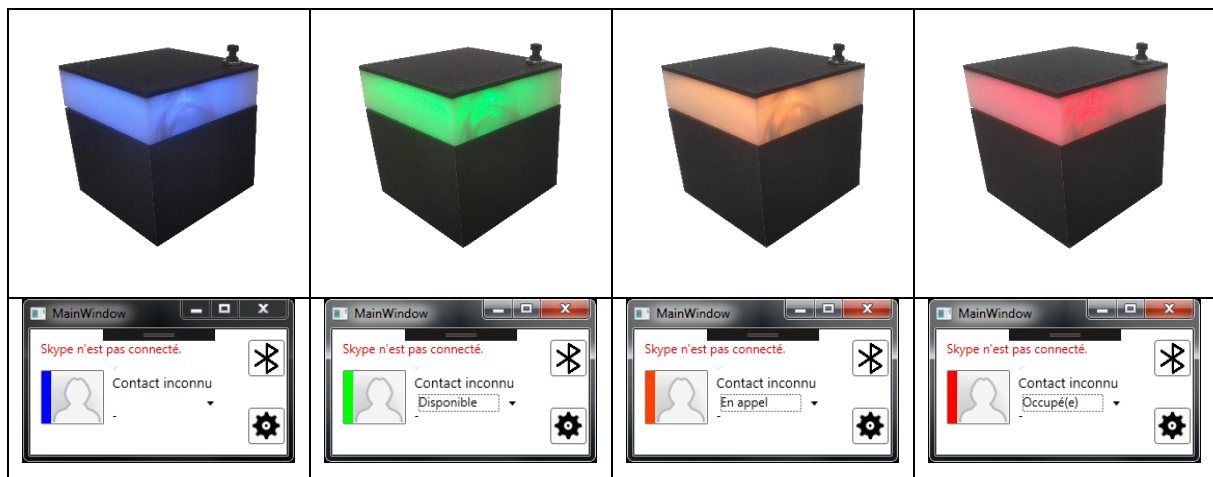
Eléments connectés : Boitier, Application, Skype, Connexion USB

Résultat attendu :

- Le statut de l'application correspond au statut Skype.
- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).

Résultat obtenu :

- L'application se comporte correctement
- Le boitier se comporte correctement.



Eléments connectés : Boitier, Application, Skype, Connexion Bluetooth

Résultat attendu :

- Le statut de l'application correspond au statut Skype.
- La LED change de couleur (Bleu->Vert->Orange->Rouge->Bleu->...).

Résultat obtenu :

- L'application se comporte correctement
- Le boîtier se comporte correctement.



3.2.5 Déconnecter Skype

Eléments connectés : Boitier, Application, Connexion USB

Résultat attendu :

- L'application indique que Skype est déconnecté.

Résultat obtenu :

- L'application se comporte correctement

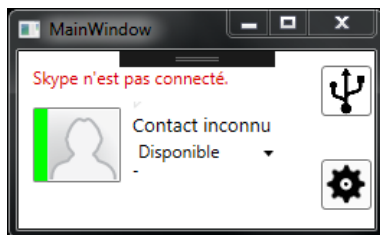
Eléments connectés : Boitier, Application, Connexion Bluetooth

Résultat attendu :

- L'application indique que Skype est déconnecté.

Résultat obtenu :

- L'application se comporte correctement



3.2.6 Déconnecter le boitier

Eléments connectés : Application, Connexion USB

Résultat attendu :

- L'application indique que le boitier est déconnecté.

Résultat obtenu :

- L'application se comporte correctement

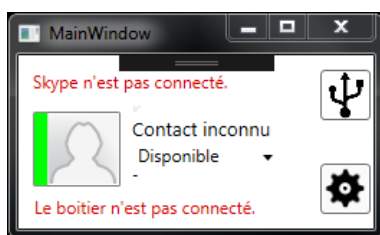
Eléments connectés : Application, Connexion Bluetooth

Résultat attendu :

- L'application indique que le boitier est déconnecté.

Résultat obtenu :

- L'application se comporte correctement



3.2.7 Déconnecter l'application

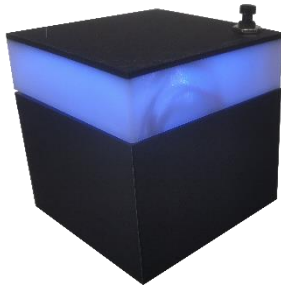
Eléments connectés : Boitier

Résultat attendu :

- La couleur passe au bleu (couleur stand-by).

Résultat obtenu :

- Le boitier se comporte correctement : Change sa couleur après quelques secondes de délai. (Image à droite)



3.2.8 Cliquer sur le message « boîtier déconnecté »

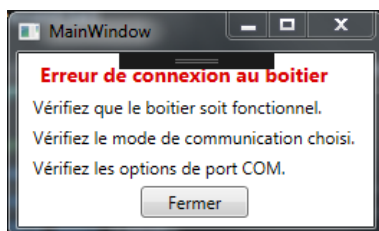
Éléments connectés : Application

Résultat attendu :

- Une page avec des informations s'affiche.
- Cette page se ferme lorsqu'on clique sur son bouton.

Résultat obtenu :

- L'application se comporte correctement.



3.2.9 Ouvrir la page de paramètres

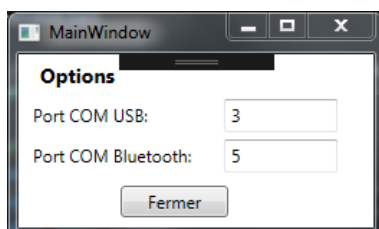
Éléments connectés : Application

Résultat attendu :

- La page des paramètres s'ouvre.
- Cette page se ferme lorsqu'on clique sur son bouton.

Résultat obtenu :

- L'application se comporte correctement.



3.2.10 Changer les paramètres

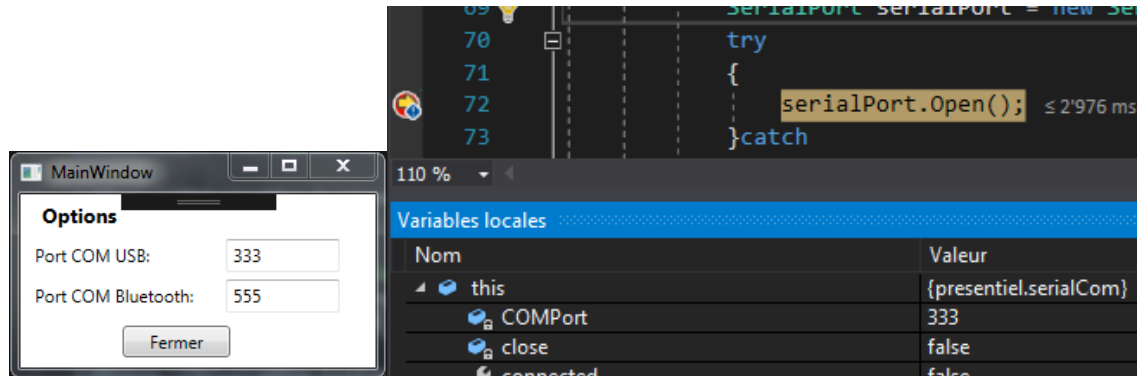
Éléments connectés : Boîtier, Application, Connexion Bluetooth/USB

Résultat attendu :

- Le changement est enregistré.
- Le changement s'applique à la communication serial.

Résultat obtenu :

- L'application se comporte correctement.



On voit ici que le paramètre est bien appliqué au relancement de la communication qui intervient à la fermeture de la page de paramètres. En revenant sur cette page, les paramètres restent inchangés (333 et 555).

3.3 Erreurs restantes

S'il reste encore des erreurs:

- *Description détaillée*
- *Conséquences sur l'utilisation du produit*
- *Actions envisagées ou possibles*

Lorsque l'on survole l'un des boutons de l'interface principale, l'animation de focus se déclenche. Cela a pour effet de changer le fond par un dégradé et donc de masquer le logo. Cette erreur n'a qu'une conséquence visuelle et aucune incidence sur le fonctionnement technique du produit.

Pour résoudre ce problème, il faudrait créer un style personnalisé pour ces boutons soit reprenant l'ensemble du style existant sans le focus et écrasant le style par défaut soit en créant un style contenant une nouvelle animation n'écrasant pas le style par défaut. Dans les deux cas cela devrait être assez long à mettre en place. Compte tenu du faible impact de cette erreur, elle n'a pas été corrigée.

3.4 Liste des documents fournis

Lister les documents fournis au client avec votre produit, en indiquant les numéros de versions

-
- *le rapport de projet*
 - *le manuel d'Installation (en annexe)*
 - *le manuel d'Utilisation avec des exemples graphiques (en annexe)*
 - *autres...*

4 **Conclusions**

Développez en tous cas les points suivants:

- *Objectifs atteints / non-atteints*
- *Points positifs / négatifs*
- *Difficultés particulières*
- *Suites possibles pour le projet (évolutions & améliorations)*

5 Annexes

5.1 Résumé du rapport du TPI / version succincte de la documentation

5.2 Sources – Bibliographie

Liste des livres utilisés (Titre, auteur, date), des sites Internet (URL) consultés, des articles (Revue, date, titre, auteur)... Et de toutes les aides externes (noms)

5.3 Journal de travail

Date	Durée	Activité	Remarques

5.4 Manuel d'Installation

5.5 Manuel d'Utilisation

5.6 Archives du projet

Media, ... dans une fourre en plastique