

Projet BAC STI2D SIN Série 2018

Gonfleur Autonome en Energie

Sommaire :

I. Cahier des charges

1. Répartition du projet
2. Diagramme d'exigence
3. Diagramme de cas d'utilisation

II. Planning : Diagramme de Gantt

III. Choix des composants

IV. Etude SysML

1. Diagramme de définition de bloc
2. Diagramme de bloc interne
3. Diagramme de séquence

V. Création de l'application

1. Ordinogramme
2. App Inventor : Initialisation
3. App Inventor : Haut : Initialisation
4. App Inventor : Haut : Clavier numérique
5. App Inventor : Haut : Bs, virgule
6. App Inventor : Haut : Connection, Liste pression
7. App Inventor : Haut : Supprimer, Démarrer
8. App Inventor : Bas : Ajouter pression, Delete longclick
9. App Inventor : Bas : Suppr pression
10. App Inventor : Quitter app

VI. Création du Psoc

1. Topdesign
2. Programme

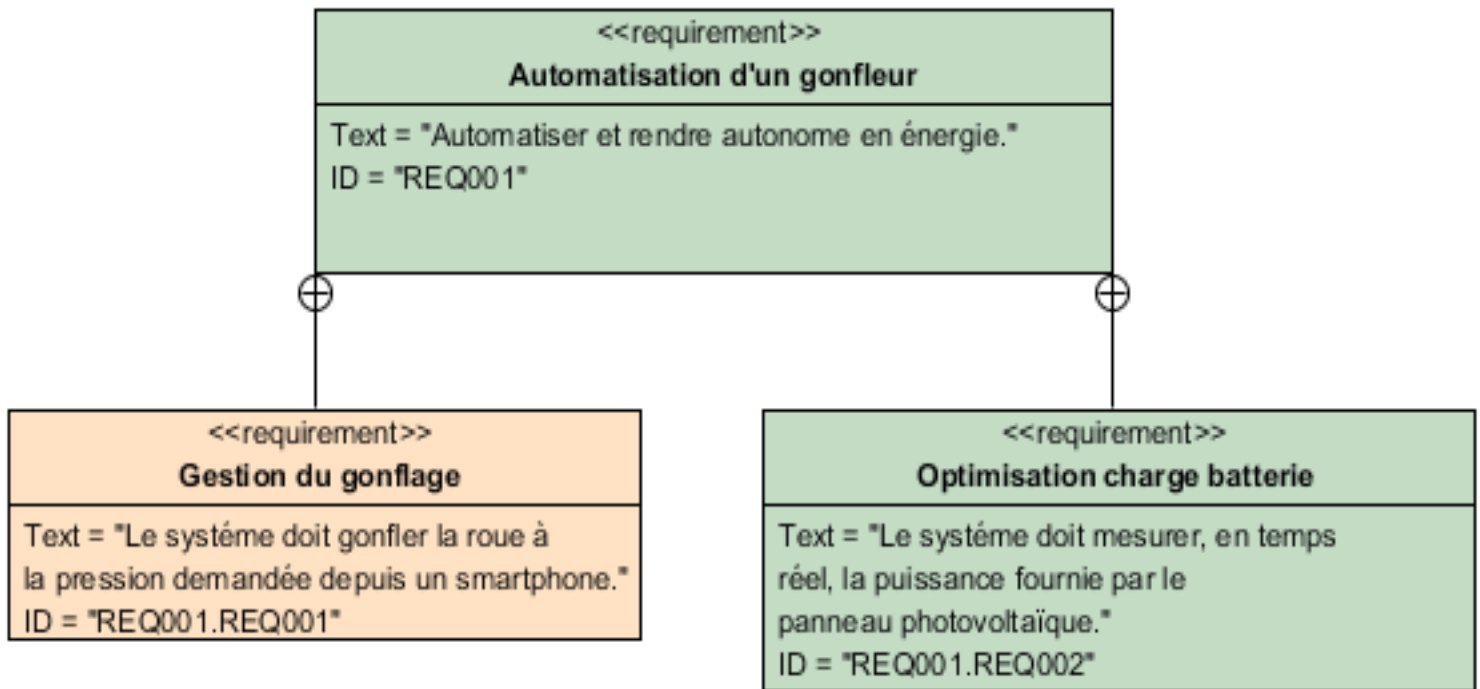
VII. Problèmes rencontrés

VIII. Test d'envoi

IX. Amélioration possible

I. Cahier des charges :

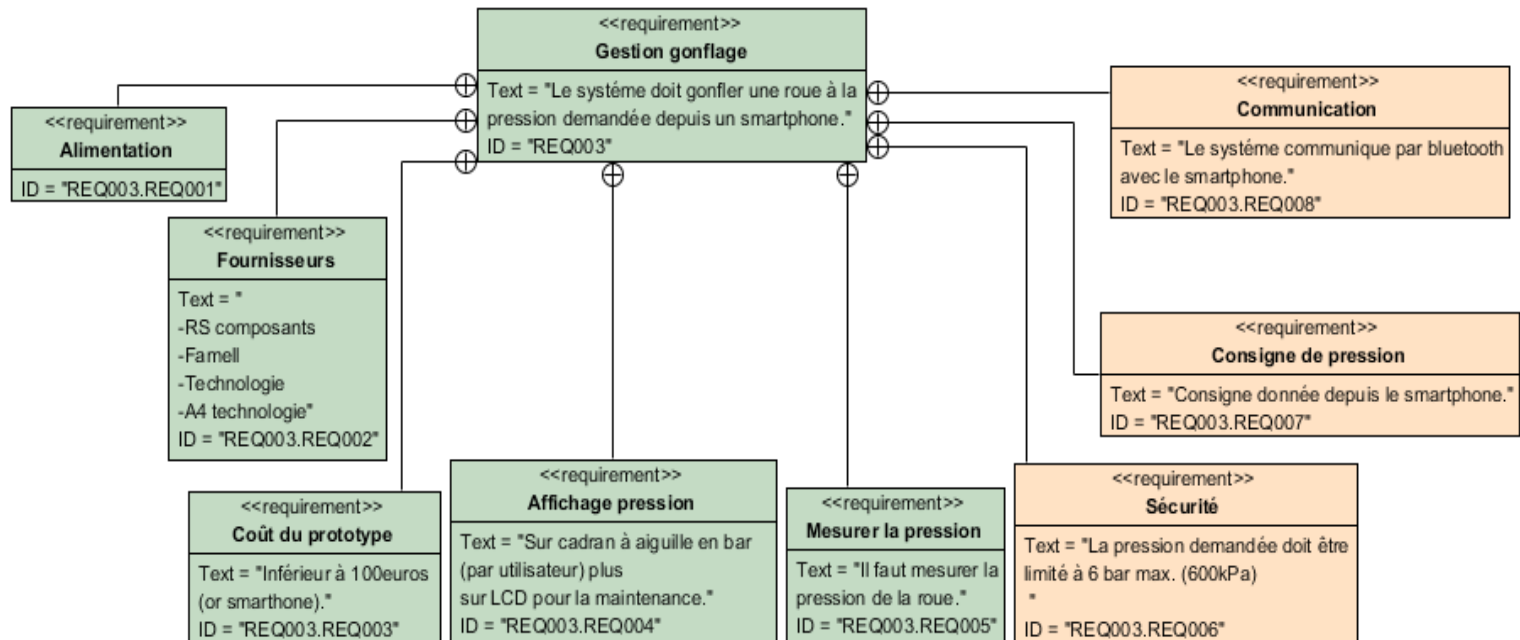
1/Répartition du projet



Ici nous avons une vue générale de notre projet, pour une optimisation de temps et d'efficacité le projet est découpé en deux parties. Le but de mon groupe est de gérer toute la gestion du gonflage.

I. Cahier des charges :

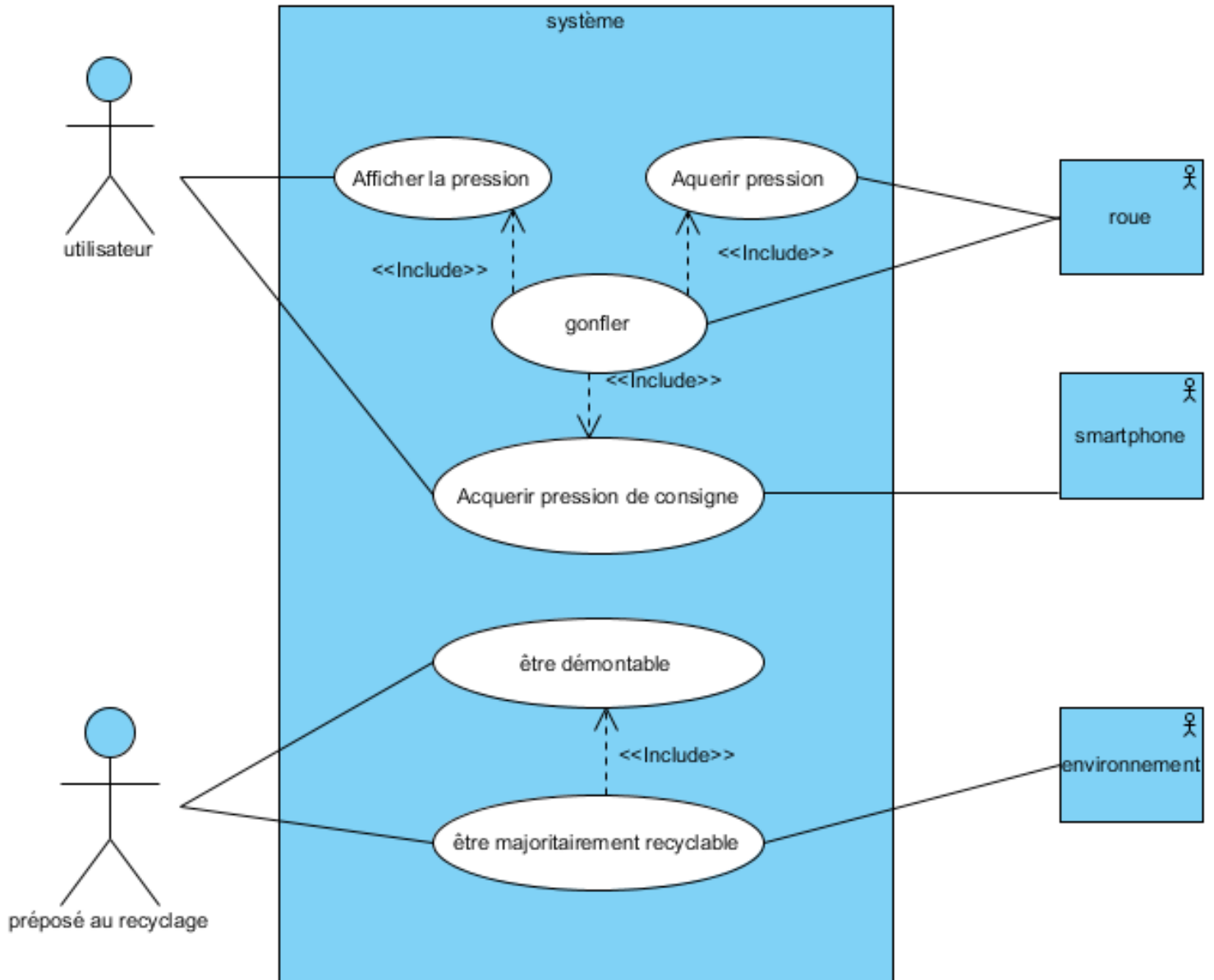
2/Diagramme d'exigence



Un diagramme d'exigence permet d'exposer les différentes contraintes que doit respecter le système. L'objectif de ce diagramme est d'organiser toutes les exigences auxquelles le système doit se plier, afin d'en avoir une vue plus précise, et surtout plus logique. Il pose les limites, les bases et les consignes du client. Nous avons ici une vue plus précise de notre partie, elle permet de recenser toute les "exigences". Ma part du projet et de créer l'application, la communication du smartphone au gonfleur et la réception de la donnée envoyée.

I. Cahier des charges :

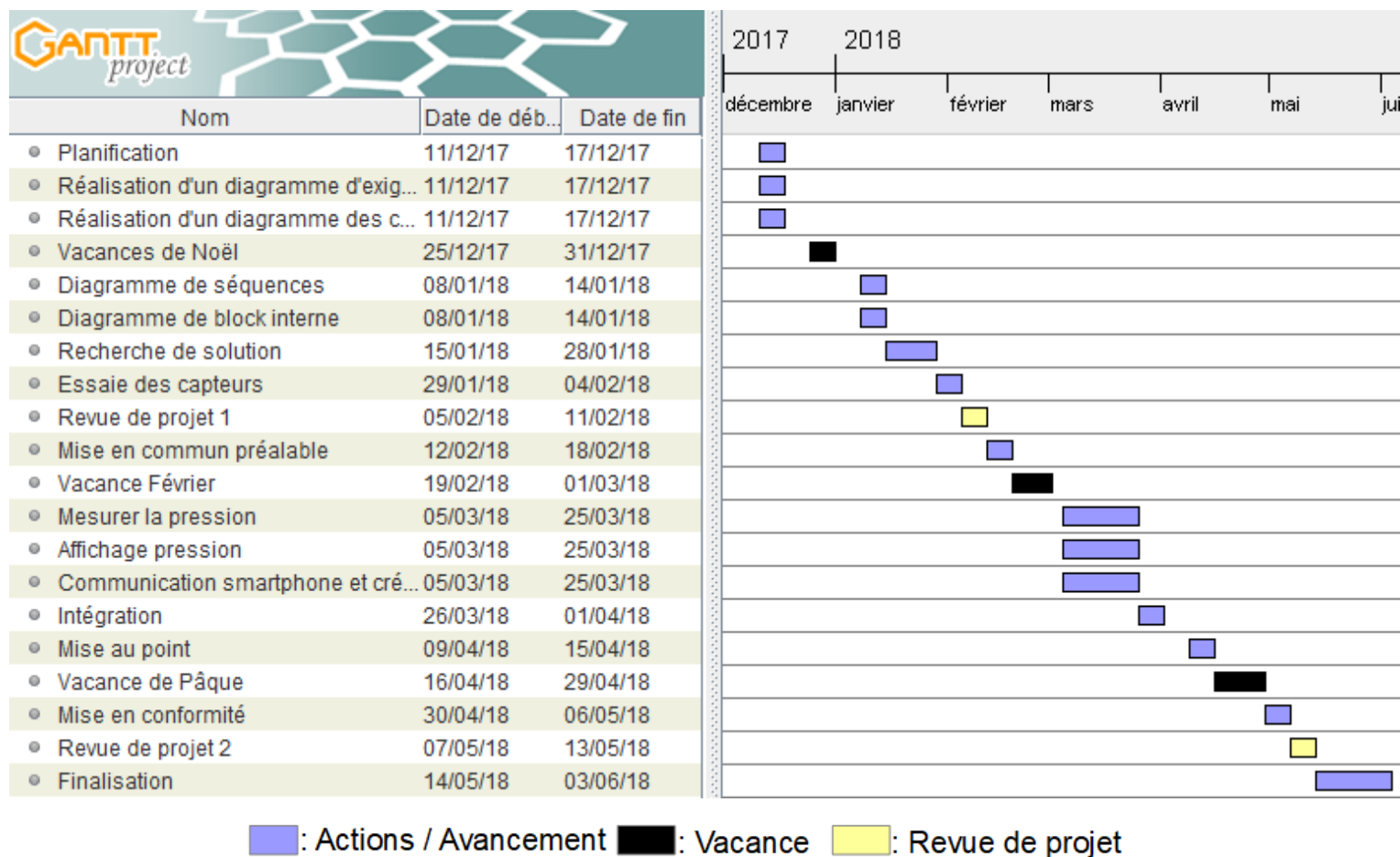
3/Diagramme de cas d'utilisation



Cet outil nous permet d'identifier les cas d'utilisation en se plaçant du point de vue d'un acteur et déterminer pourquoi il se sert du système. Il nous montre les services rendu par le système.

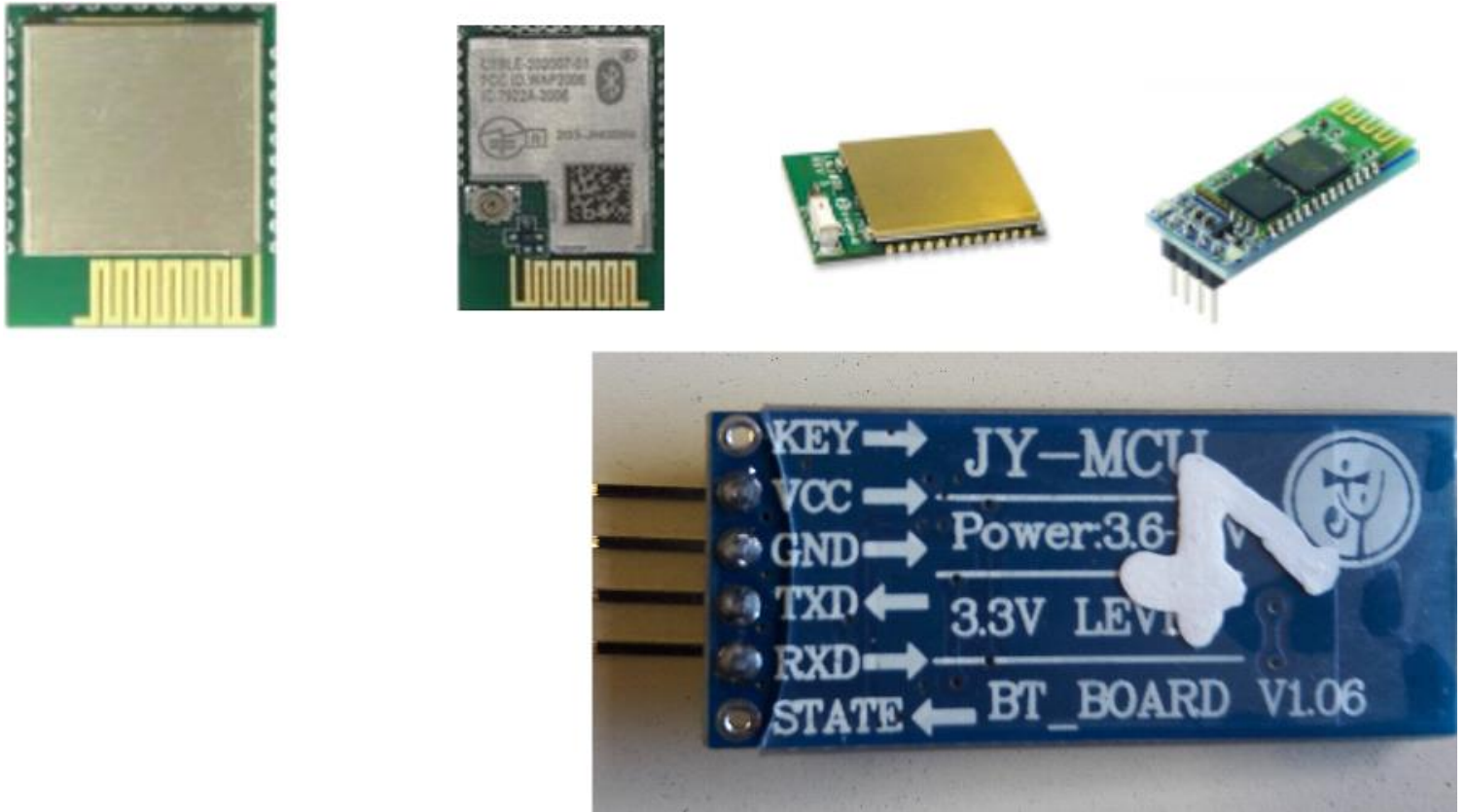
II. Planning :

Diagramme de Gantt :



Ce diagramme illustre les taches réparties de notre groupe, les vacances ainsi que les revues de projet. Notre projet a duré environ 70h.

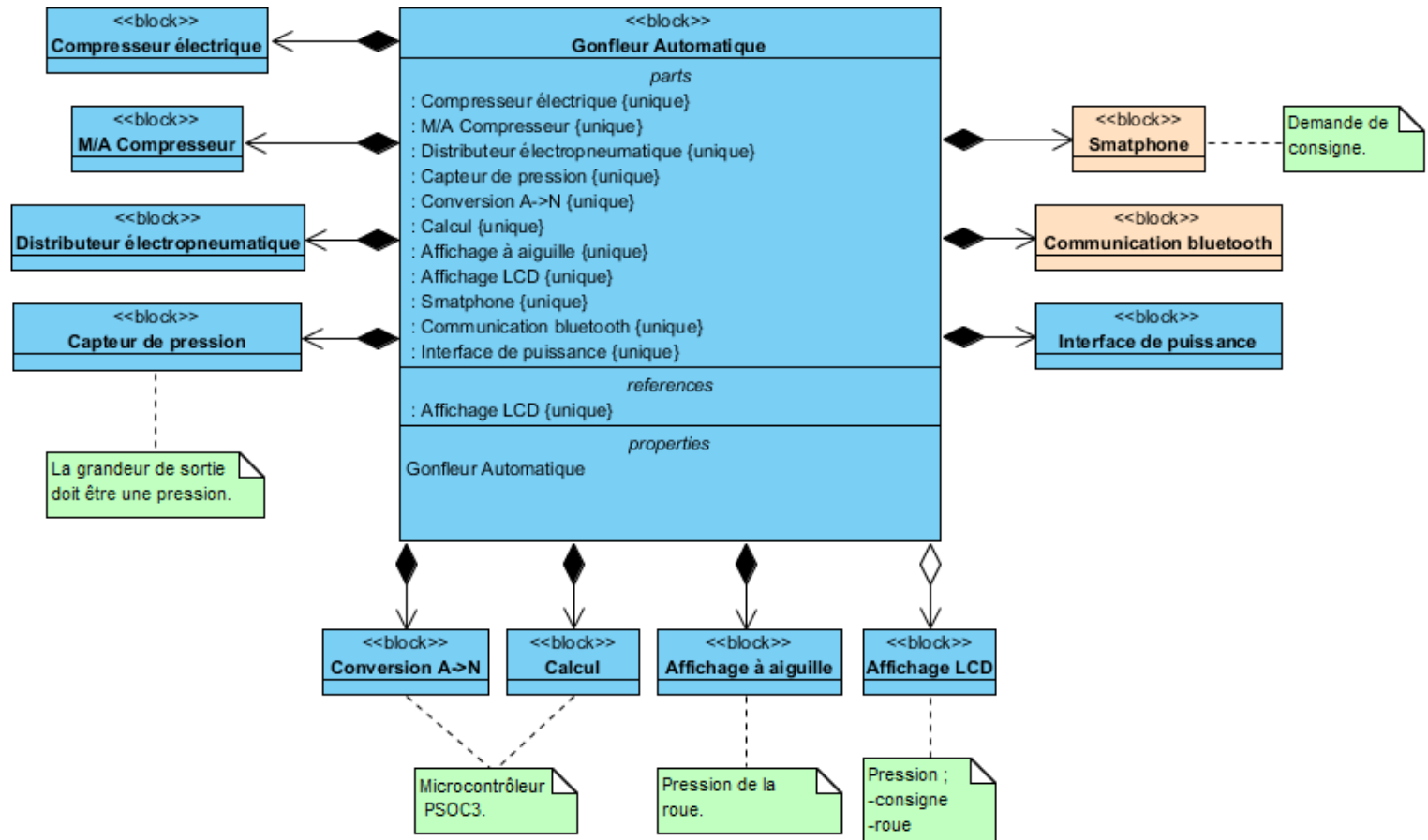
III. Choix des composants :



La communication entre le smartphone et le gonfleur se fait par Bluetooth, le gonfleur a besoin d'un module Bluetooth pour communiquer. Pour faire le tri entre les nombreux composants que j'ai sélectionné, j'ai dû fixer plusieurs critères tel que la compatibilité, le prix, la caractéristique. Après analyse, le composant final possède des broches, n'est pas BLE et remplit parfaitement toutes les conditions de fonctionnement pour notre système. Il s'agit du composant JY-MCU : BT_BOARD V1.06 à un prix de 4,31€.

IV. Etude SysML :

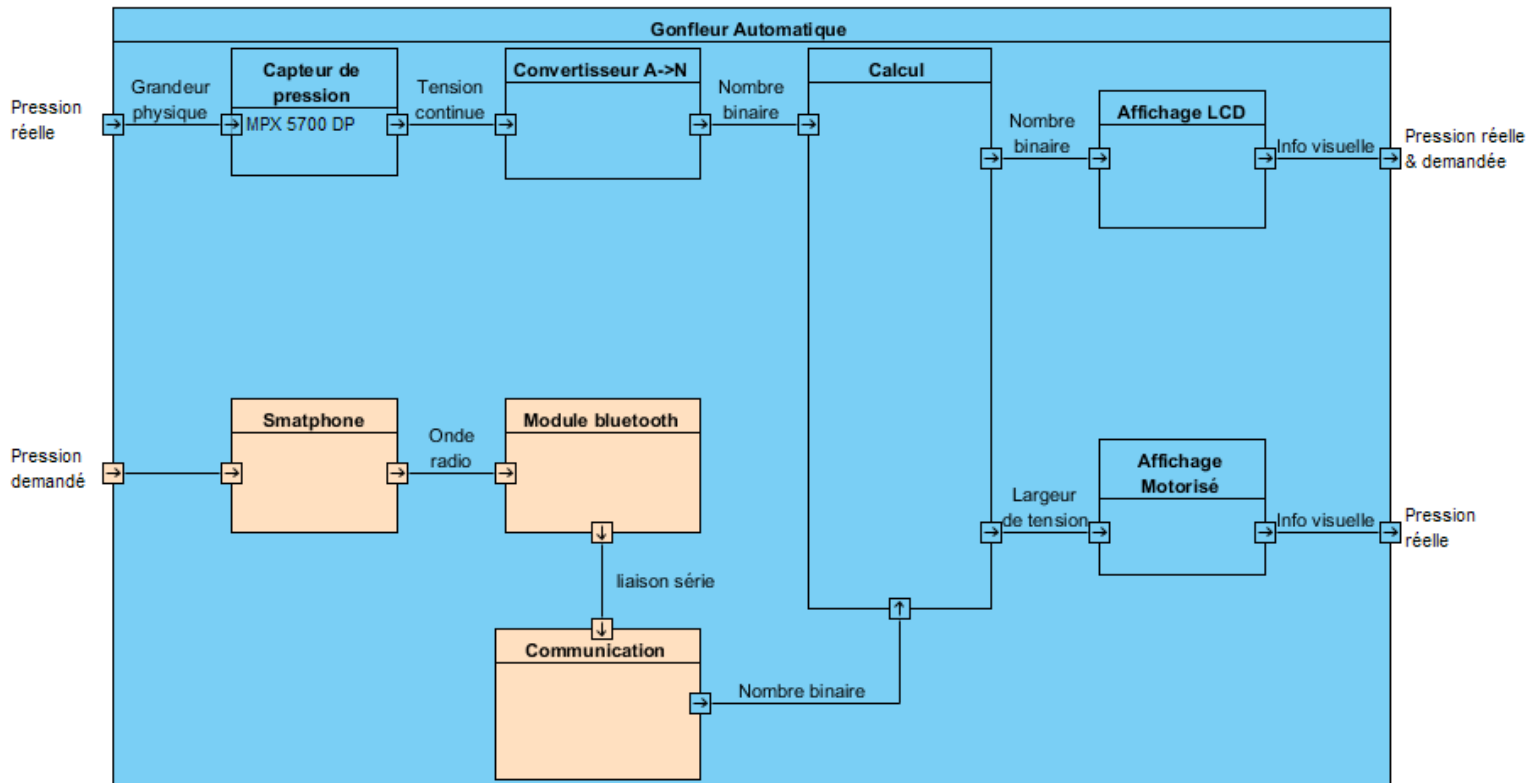
1/Diagramme de définition de bloc



Il sert à représenter le type d'éléments matériels et logiciels d'un système par l'intermédiaire de bloc, cela permet une meilleure organisation du système. L'affichage LCD est optionnel car il est pour la maintenance du gonfleur.

IV. Etude SysML :

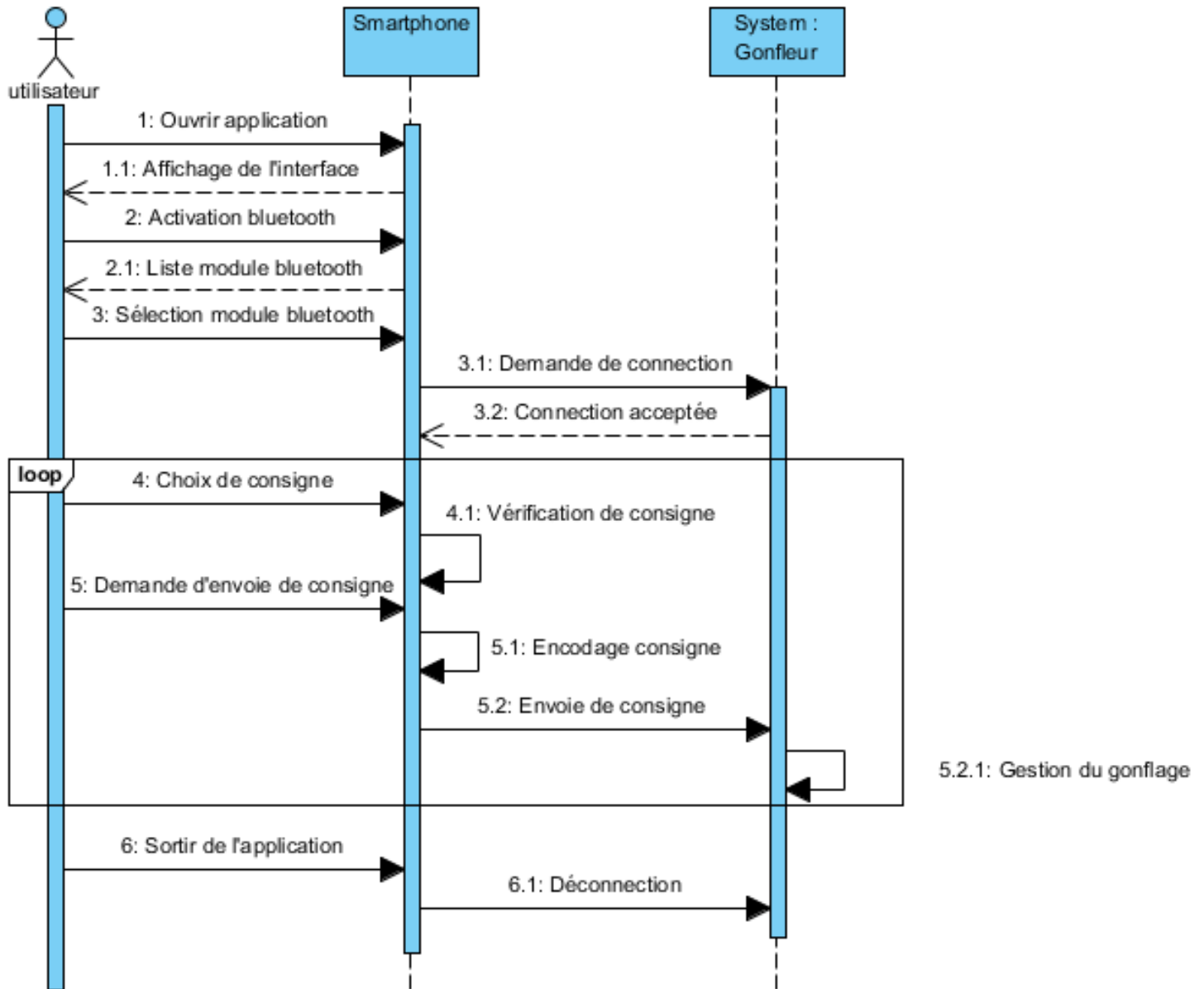
2/Diagramme de bloc interne



Le diagramme de bloc interne modélise la structure interne d'un bloc. Il décrit les flux et interactions entre des blocs qui caractérisent les fonctions

IV. Etude SysML :

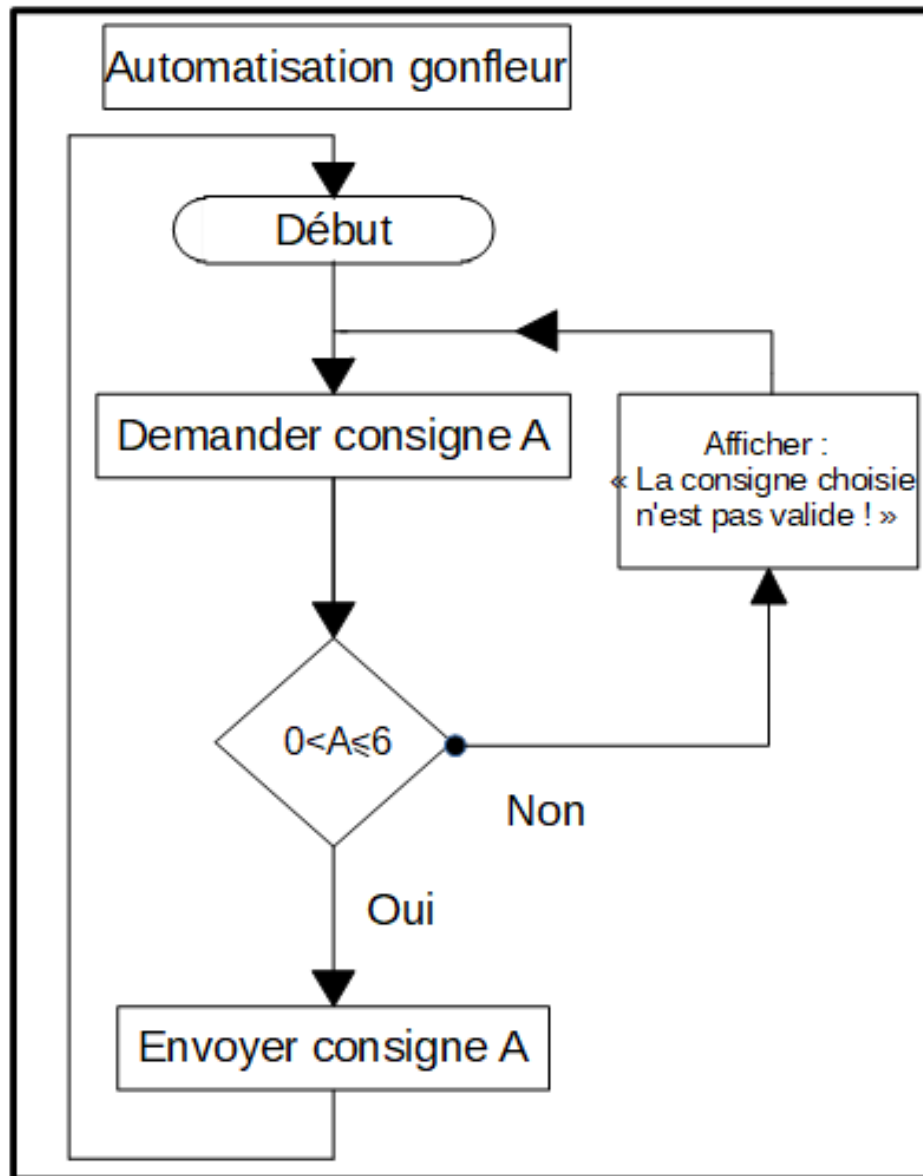
3/Diagramme de séquence



Les diagrammes de séquences sont la représentation graphique des interactions entre les acteurs et le système selon un ordre chronologique.

V. Création de l'application :

1/Ordinogramme



Avant la création de mon application j'ai décidé de créer cet ordinogramme. Pour une lisibilité plus simple, et une meilleure compréhension du programme de l'application.

V. Création de l'application :

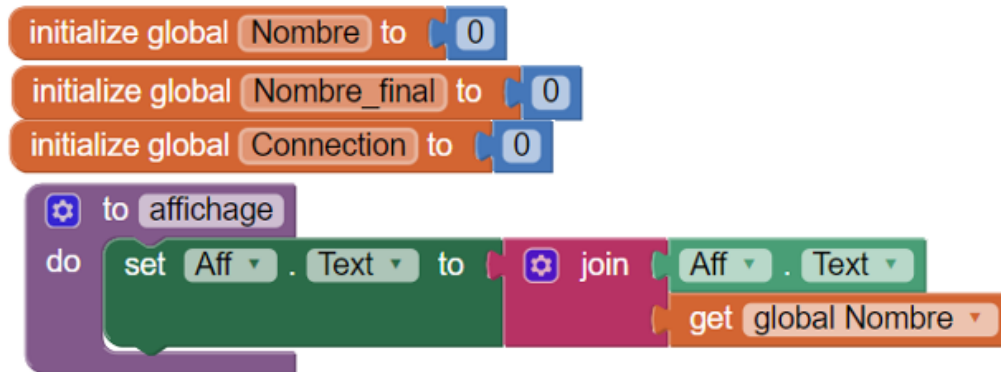
2/App Inventor : Initialisation



Initialisations du timer pour "refresh" la barre de notif. Initialisation variable "list" de la liste de pression. Création de mémoire TinyDB 1 pour le stockage des données.

V. Création de l'application :

3/App Inventor : Haut

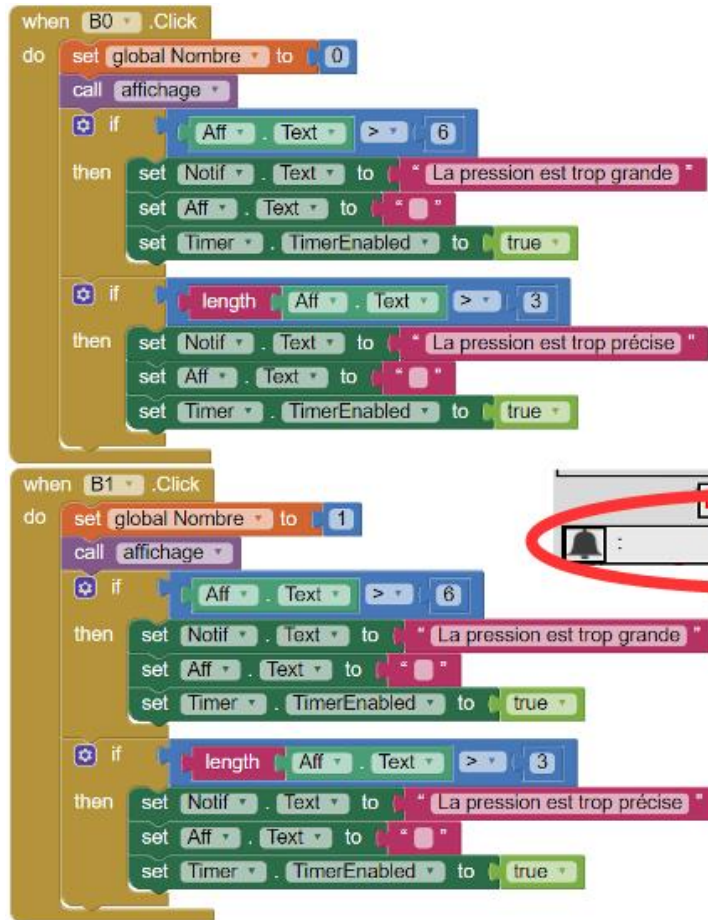


Gonfleur APP			
1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

Création du programme qui permet de coller le chiffre saisi avec les caractères déjà existant dans le label. Création d'une interface personnalisée pour l'appli.

V. Création de l'application :

4/App Inventor : Haut : Clavier numérique



1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

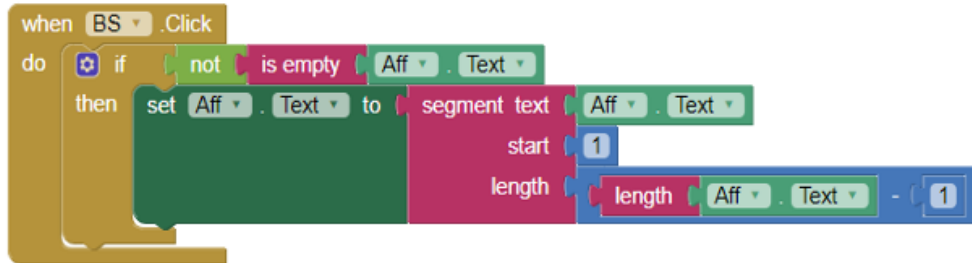


1	2	3
4	5	6
7	8	9

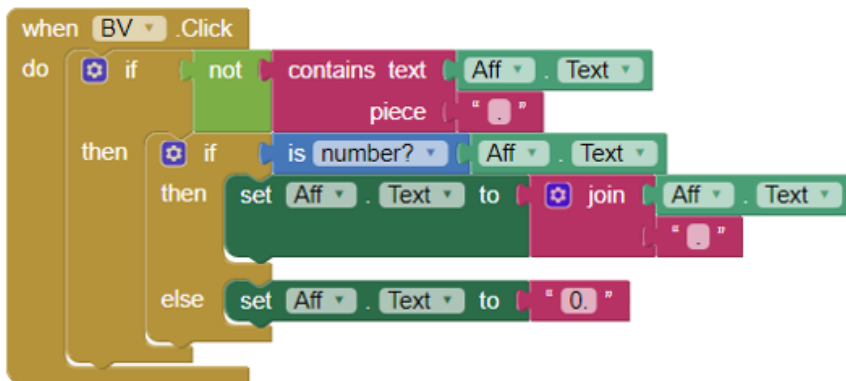
Création des boutons de saisi de chiffre, des alertes correspondantes et activation du "refresh" de la barre de notification.

V. Création de l'application :

5/App Inventor : Haut : Bs, virgule



1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression



1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

Création du bouton "BS" qui supprime le caractère précédemment saisi dans le label et du bouton "BV" qui permet de rajouter une virgule ou un raccourci "0.".

V. Création de l'application :

6/App Inventor : Haut : Connection, Liste pression

when Connection .BeforePicking

do set Connection .Elements to BluetoothClient1 .AddressesAndNames

when Connection .AfterPicking

do set Connection .Selection to call BluetoothClient1 .Connect address Connection .Selection

if BluetoothClient1 .IsConnected

then set Etat .Text to "Connecté"

set Etat .TextColor to green

set global Connection to 1

else set Etat .Text to "Déconnecté"

set Etat .TextColor to red

set global Connection to 0

Ajouter pression Supprimer pression

Déconnecter

when ListP .AfterPicking

do set Aff .Text to ListP .Selection

1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

Création du bouton "Connection" avec son alerte et du bouton "Liste de pression" qui affichera une liste de pression enregistrée au préalable.

V. Création de l'application :

7/App Inventor : Haut : Supprimer, Démarrer

```

when BSUPPR .Click
do
  set Aff . Text to " "
  set global Nombre to 0

when BGONFL .Click
do
  if 0 ≠ get global Connection
  then
    if is empty Aff . Text
    then
      set Notif . Text to "Saisir pression (bar)"
      set Timer . TimerEnabled to true
    else
      set global Nombre_final to 10 × Aff . Text
      set Notif . Text to "La pression est envoyée"
      call BluetoothClient1 .Send1ByteNumber
        number get global Nombre_final
    else
      set Notif . Text to "Vous n'êtes pas connecté"
      set Timer . TimerEnabled to true
  
```

1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

1	2	3	Connection
4	5	6	Démarrer
7	8	9	Supprimer
.	0	←	Liste pression

Création du bouton "BSUPPR" qui permet de supprimer tous les caractères présents dans le label et du bouton "BGONFL" qui permet, après plusieurs vérifications de saisie, d'envoyer la consigne présente dans le label après l'avoir multiplié par 10.

V. Création de l'application :

8/App Inventor : Haut : Ajouter pression, Delete longclick

The image displays two sets of code blocks for an App Inventor application, along with a visual representation of the user interface.

Code for 'AjouterP' .Click:

- when **AjouterP** .Click
- do
 - if **Aff** .Text \neq 0
 - then
 - if **Aff** .Text \neq " "
 - then
 - add items to list **list** (get **global List**)
item **Aff** .Text
 - call **TinyDB1** .StoreValue
tag "dbpression"
valueToStore (get **global List**)
 - set **SupprP** .ElementsFromString to (replace all text (list to csv row **list** (get **global List**))
segment " "
replacement " ")
 - set **Aff** .Text to " "

Code for 'Delete' .LongClick:

- when **Delete** .LongClick
- do
 - call **TinyDB1** .ClearAll
 - call **TextToSpeech1** .Speak
message "RAM effacée "
 - set **Timer** .TimerEnabled to true
 - close application

UI Representation:

The visual representation shows two states of the application interface. The top state shows a button labeled "Ajouter pression" (highlighted in red) and a button labeled "Suppr pression" (highlighted in red). The bottom state shows the same buttons, but the "Ajouter pression" button is now disabled (greyed out) and the "Suppr pression" button is highlighted in red.

Création du bouton "AjouterP" qui permet de mémoriser la consigne saisie dans le label et de l'afficher dans la liste de pression.

Le bouton "Delete" lui permet d'effacer toutes les données mémorisées.

V. Création de l'application :

9/App Inventor : Bas : Suppr pression

The image displays two event-driven code blocks in a visual programming language (Scratch-style) for App Inventor, along with a UI element.

Top Code Block: when Notifier1 .AfterChoosing

- choice** (block)
- do** (block)
 - if** (block)
 - get choice** (block) = **" Oui "** (text)
 - then** (block)
 - remove list item list** (block)
 - index** (block)
 - get global List** (block)
 - index in list thing** (block)
 - list** (block)
 - SupprP** (block) . **Selection** (block)
 - set SupprP** (block) . **ElementsFromString** (block) to **replace all text** (block)
 - list to csv row list** (block)
 - list** (block)
 - get global List** (block)
 - segment** (block) = **" "** (text)
 - replacement** (block) = **" "** (text)

Bottom Code Block: when SupprP .AfterPicking

- do** (block)
 - call Notifier1** (block) . **ShowChooseDialog** (block)
 - message** (block)
 - join** (block)
 - " Veux-tu supprimer " "** (text)
 - SupprP** (block) . **Selection** (block)
 - " "** (text)
 - title** (block) = **" Suppression de la pression "** (text)
 - button1Text** (block) = **" Non "** (text)
 - button2Text** (block) = **" Oui "** (text)
 - cancelable** (block) = **false** (boolean)

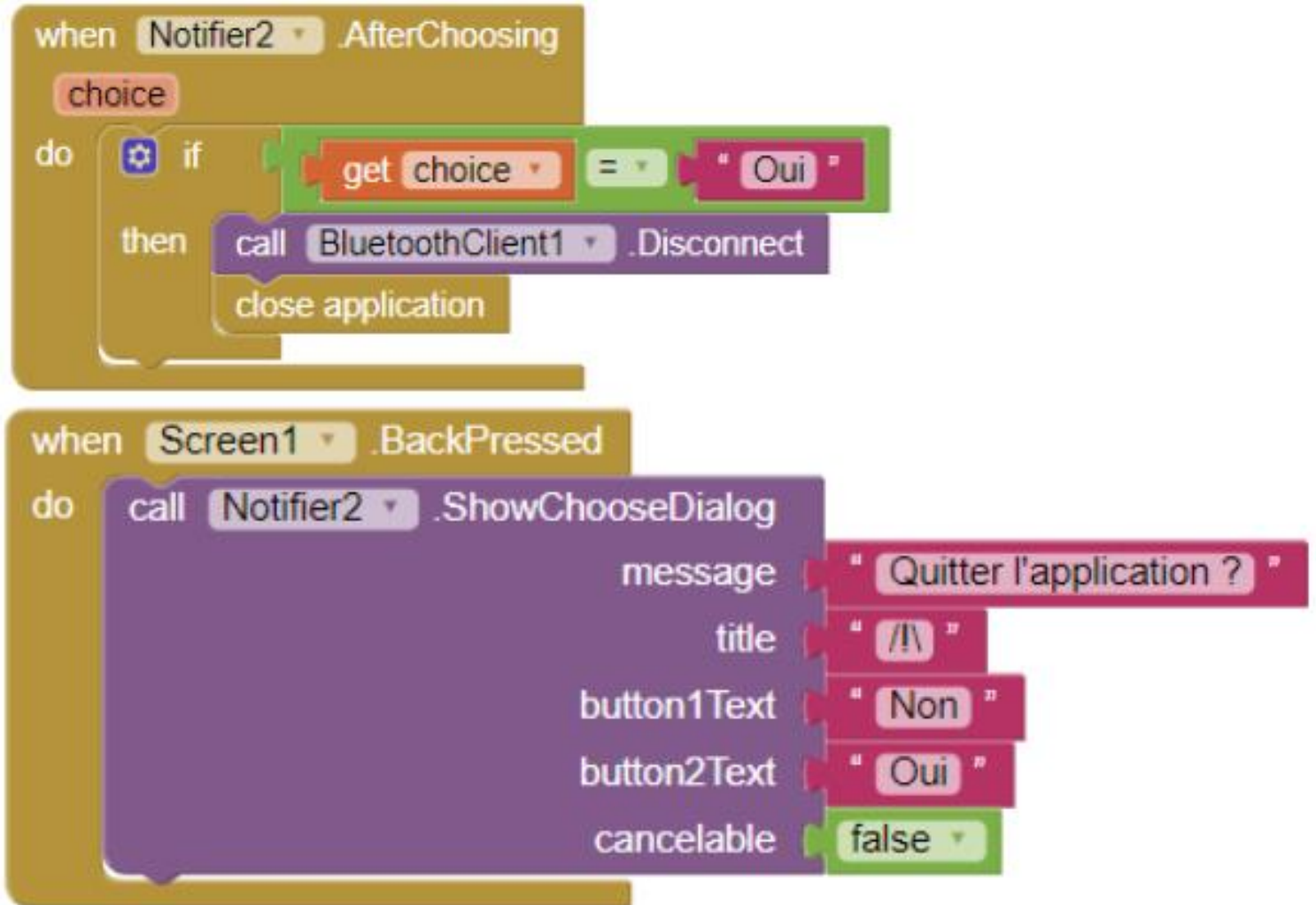
A horizontal container holding three buttons:

- Ajouter pression** (light gray button)
- Suppr pression** (red button)
- P** (red button)

Création du bouton "SupprP" qui supprime la consigne enregistrée de Liste de pression.

V. Création de l'application :

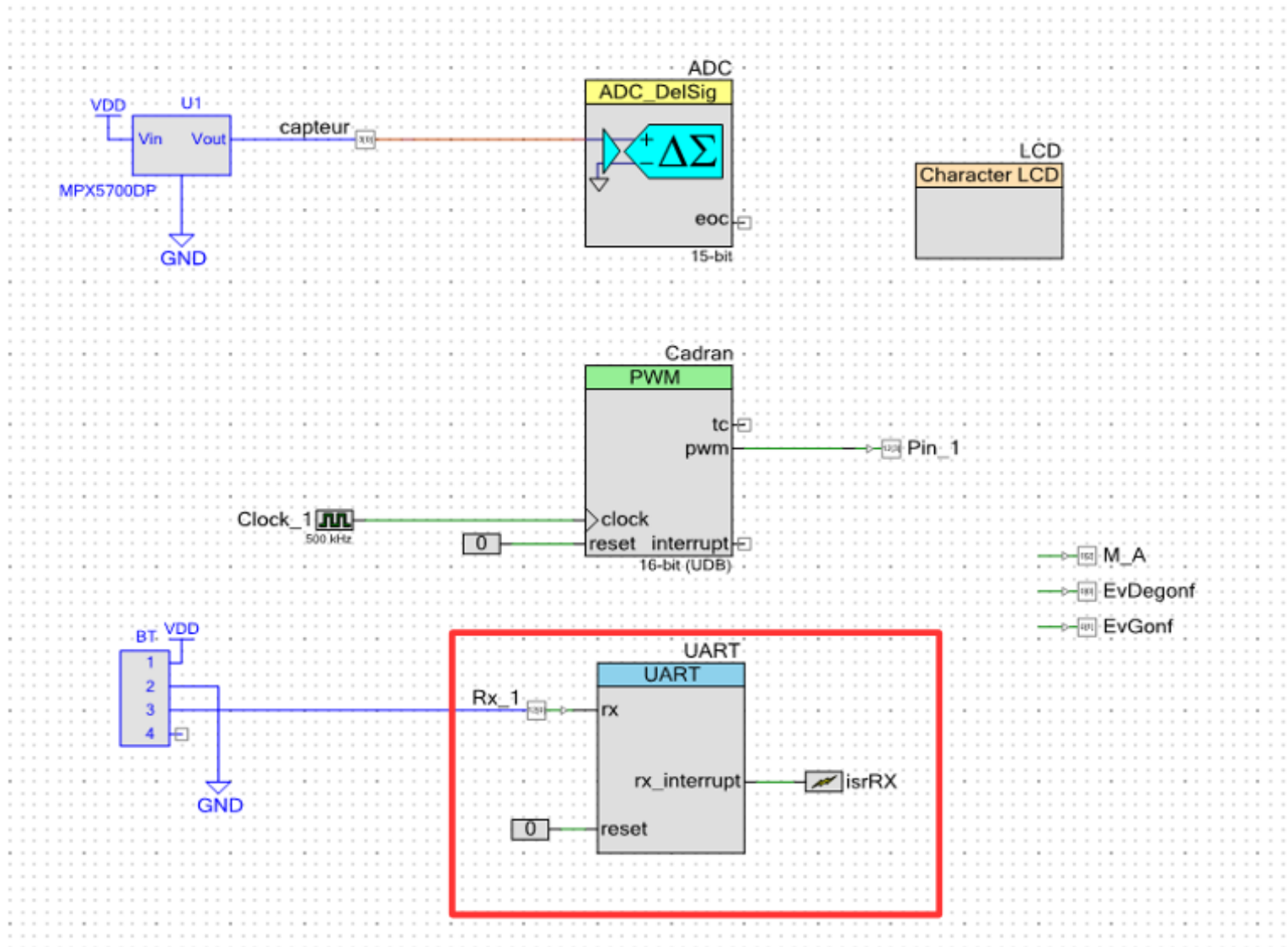
10/App Inventor : Quitter app



Création d'un petit programme qui permet de fermer l'application et de se déconnecter du module Bluetooth après avoir "backpressed" l'écran de l'application.

VI. Création de l'application :

1/Topdesign



Ces composants sont ceux nécessaires pour mon programme PsoC. L'interruption est nécessaire car le module Bluetooth ne sait pas quand est-ce qu'il va recevoir la consigne.

VI. Création de l'application :

2/Programme

```
10| * =====
11| */
12| #include <device.h>
13| #include <stdio.h>
14| cystatus etat;
15|
16| #define quV      153
17| #define offset  199000// en uV
18| #define S       6300 //en uV
19|
20| uint16 valeur,pas,consigne;
21| uint32 pression,N;
22| int32 ecart;
23| uint8 buffer[17];
24| uint8 demande;
25|
26| CY_ISR(InterruptRX)
27| {
28|     consigne=UART_GetChar();
29|     consigne=consigne*10;
30|     demande=1;
31| }
32|
33|
34| void acquisition(void);
35| void calcul_pression(void);
36| void calcul_ecart(void);
37| void affichage(void);
38| void gonflage(uint16 duree);
39| void degonflage(uint16 duree);
40|
41|
42| void main()
43| {
44|     UART_Start();
45|     isrRX_StartEx(InterruptRX);
46|     CyGlobalIntEnable;
47|     Caution_Start();
48|     Clock_1_Start();
49|     ADC_Start();
50|     ADC_StartConvert();
51|     LCD_Start();
52|     LCD_Position(0,0);
```

Mon programme comporte la réception de la consigne et de l'interruption.

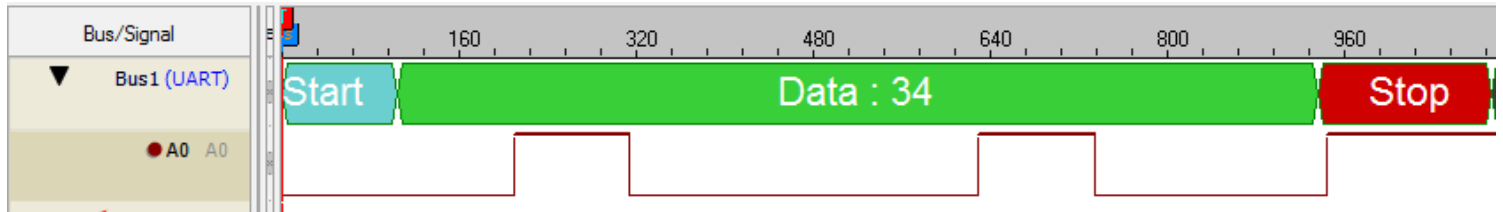
VII. Problèmes rencontré :



J'ai rencontré des problèmes au niveau de mon application smartphone tel que l'assemblage de caractère pour la consigne. Mais le plus gros problème rencontré est certainement la cassure du moteur car il n'avait pas de soupape de sécurité pour évacuer l'excédent de pression.

VIII. Test d'envoi :

Trame I2C



Nous pouvons voir le bit de start et de stop avec la consigne sur 8bit qui est égal à 3,4 bar (x10 pendant l'envoi par l'application). L'application multiplie par 10 la consigne pour réduire son nombre de caractère, "3,4" fait 3 caractères tandis que "34" fait 2 caractères.

Réduire son nombre de caractère permet de diminuer le taux d'erreur car la vitesse d'envoi du module Bluetooth n'est pas exactement la même que celle du microcontrôleur.

IX. Amélioration possible :

Dans la mesure du possible utiliser des composants ROHS avec des soudures sans plomb et sans métaux lourds.

Mettre en place ce système compatible pour les personnes à mobilité réduite.

Rajouter une fonctionnalité à l'application comme une connexion automatique du smartphone au module Bluetooth du gonfleur.

Mettre en place le même type de gonfleur automatique cette fois retravaillé pour d'autres moyens de transport, comme sur autoroute pour des pneus de voiture par exemples.