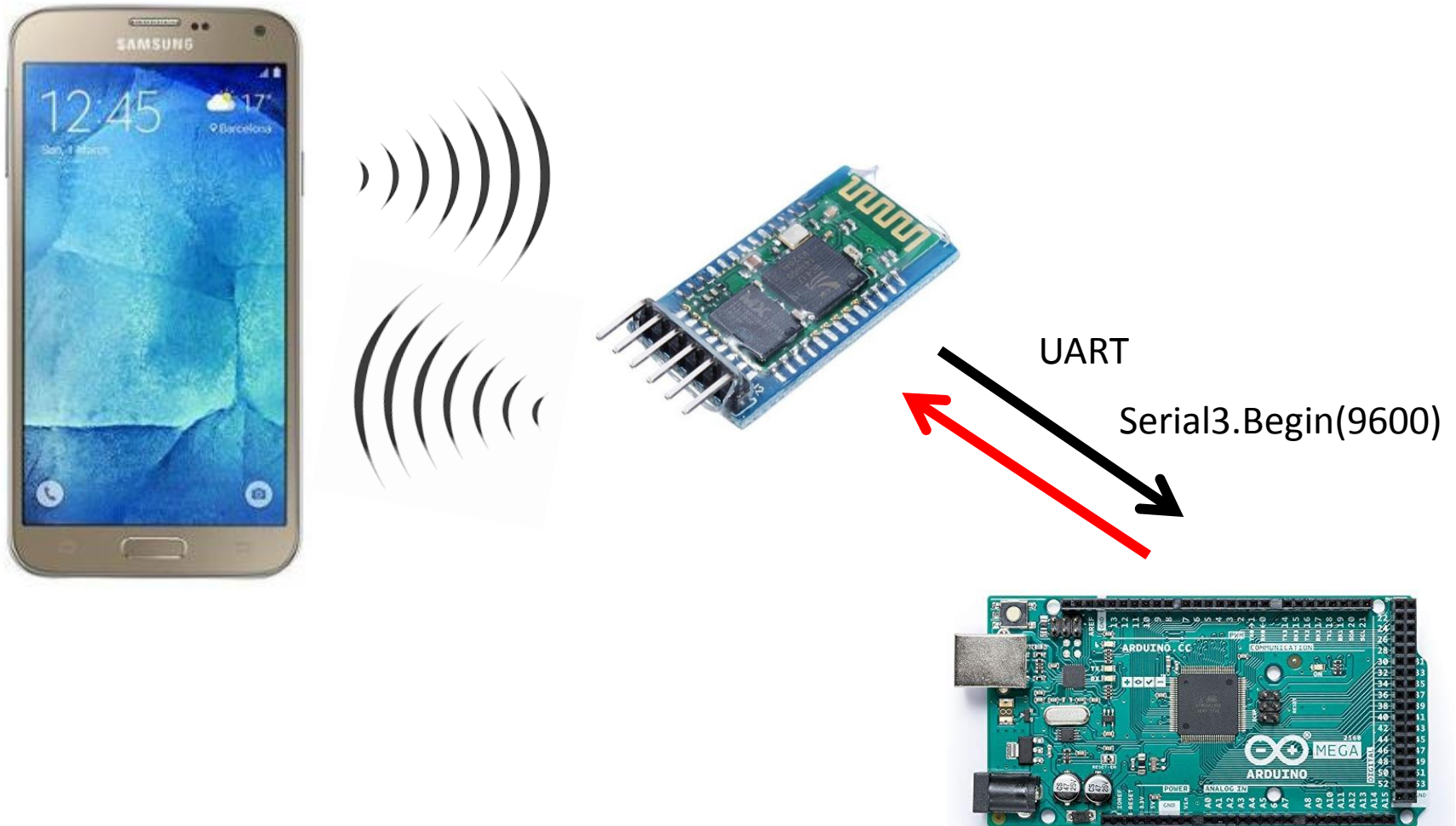
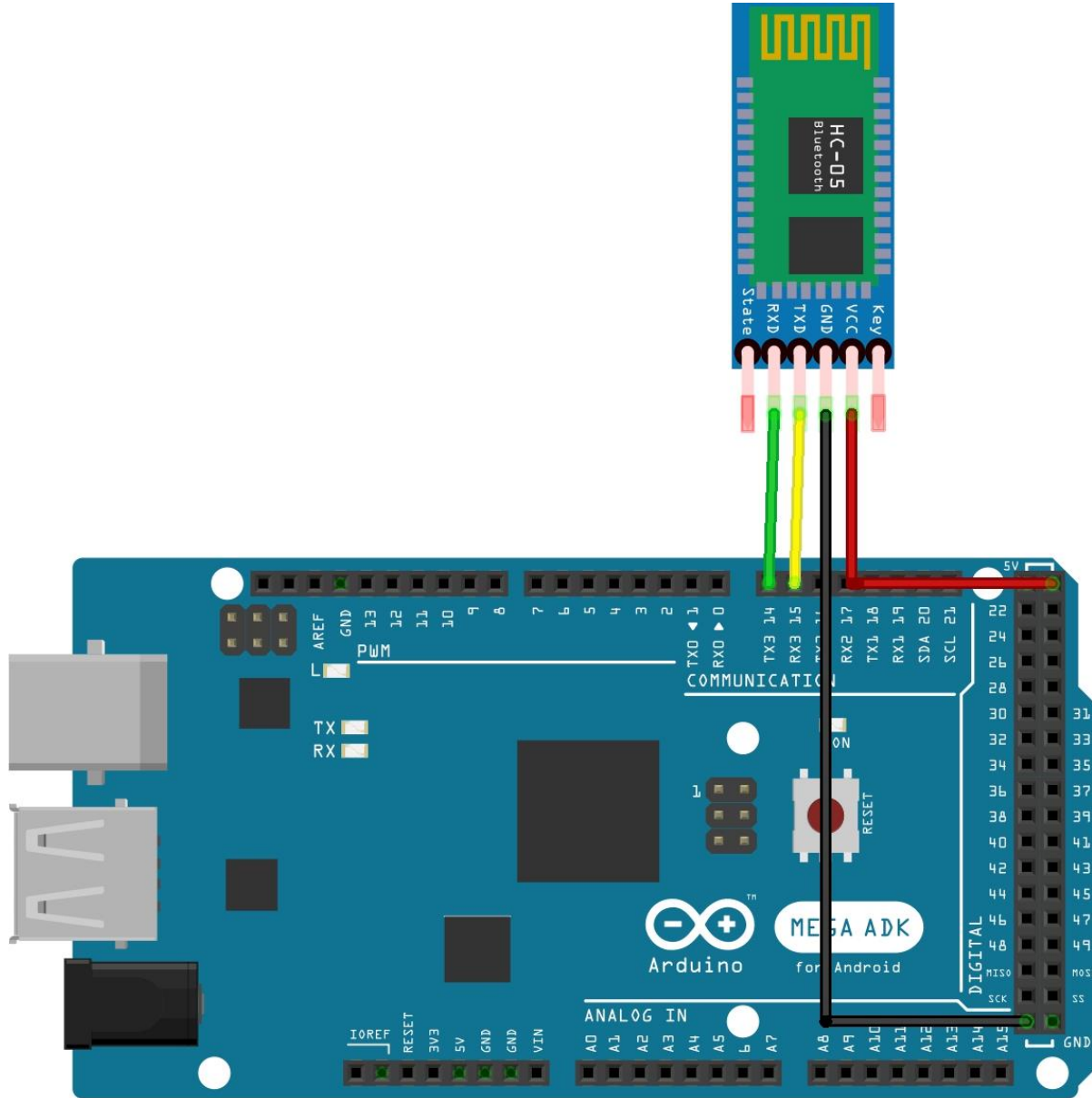


Communication Bluetooth entre téléphone et Arduino



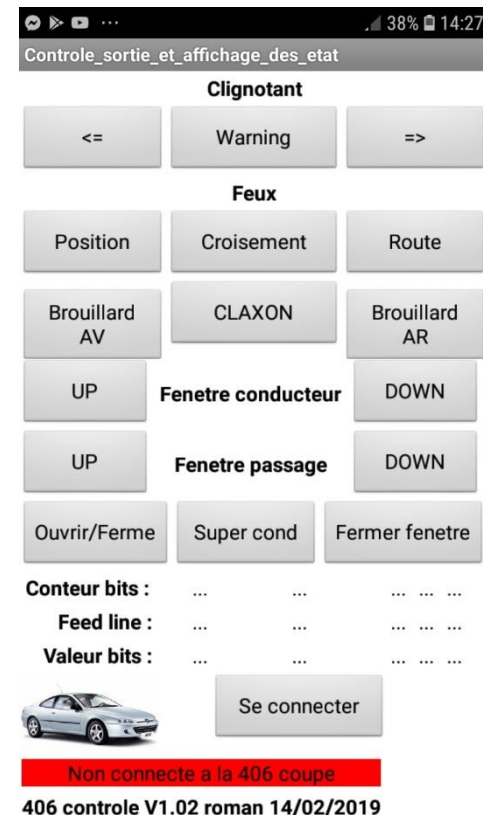
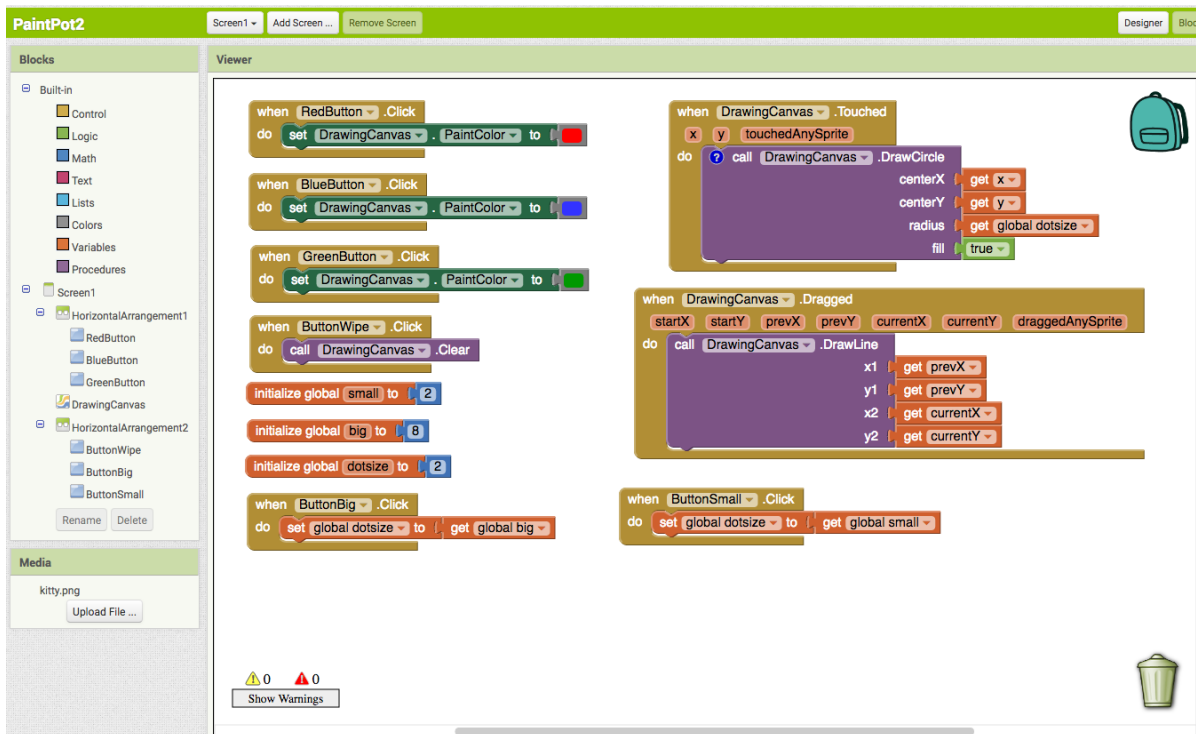
Connection



Application téléphone

Pour faire une application pour un téléphone Android rapidement j'ai utilisé MIT App Inventor (ça marche sur un navigateur internet) <https://appinventor.mit.edu/>

L'application n'est pas parfaite il y a des choses à améliorer mais pour faire les essais ça suffit



Application téléphone

test1

Controle_sortie Add Screen ... Remove Screen

Designer Blocks

Blocks

Built-in

- Control
- Logic
- Math
- Text
- Lists
- Dictionaries
- Colors
- Variables
- Procedures

Controle_sortie

- TableArrangement1
 - Button_clignotant_1
 - Button_clignotant_2
 - Label_clignotant
 - Button_WARNING
- TableArrangement2

Media

- 325.png

Viewer

do call testeDesEtat

when Button_menu_parametre Click

do open another screen screenName Ecran_De_Parametre

initialize global conneteVoiture to 0

initialize global etatBluetooth to 0

initialize global dejaconnecter to 0

when Connection_Deconnection Click

do if get global conneteVoiture = 0

then if get global etatBluetooth = 1

then set global conneteVoiture to 2

call messageEtatConnection

set Connection_Deconnection Enabled to call BluetoothClient1 Connect address 20:16:05:09:05:68

if get global conneteVoiture = 1

then call messageEtatConnection

else call testeDesEtat

else call testeDesEtat

else call BluetoothClient1 Disconnect

call TextToSpeech1 Speak

message Déconnecter de la 406 coupée

call Notifier1 ShowAlert

notice Vous avez demande a etre deconnecte

call testeDesEtat

set global dejaconnecter to 0

when Click_envoie_des_etat_des_button Times

do set global delaytime to 0

if get global conneteVoiture = 1

then modulo of get global delaytime + 2 = 0

then call envoyer_les_donnee_par_bleutooth

call envoyer_les_donnee_par_bleutooth_en_byte

if get global changement_etat_lors_de_envoie = 1

then set global changement_etat_lors_de_envoie to 0

else set global changement_etat_lors_de_envoie to 1

set message_etat_de_connection TextColor to black

call TaiFunFlashlight On

else set message_etat_de_connection TextColor to red

call TaiFunFlashlight Off

else call TaiFunFlashlight Off

to messageEtatConnection

do if get global conneteVoiture = 0

then set message_etat_de_connection BackgroundColor to red

set Connection_Deconnection Text to Se connecter

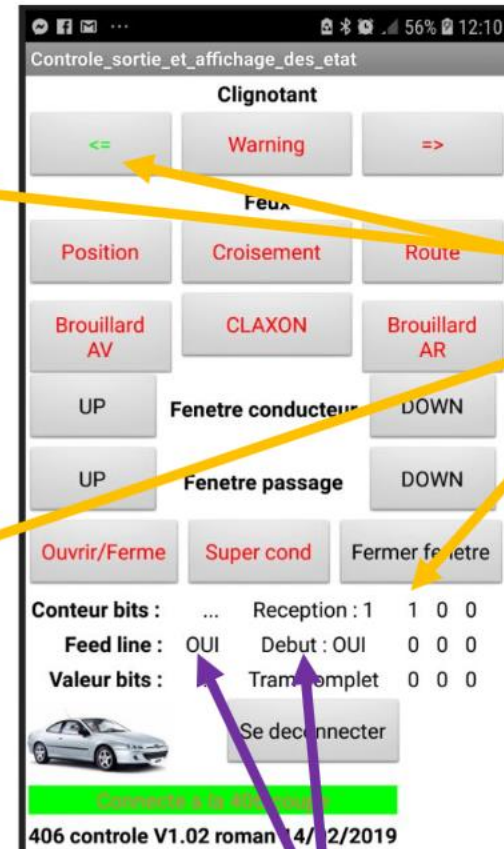
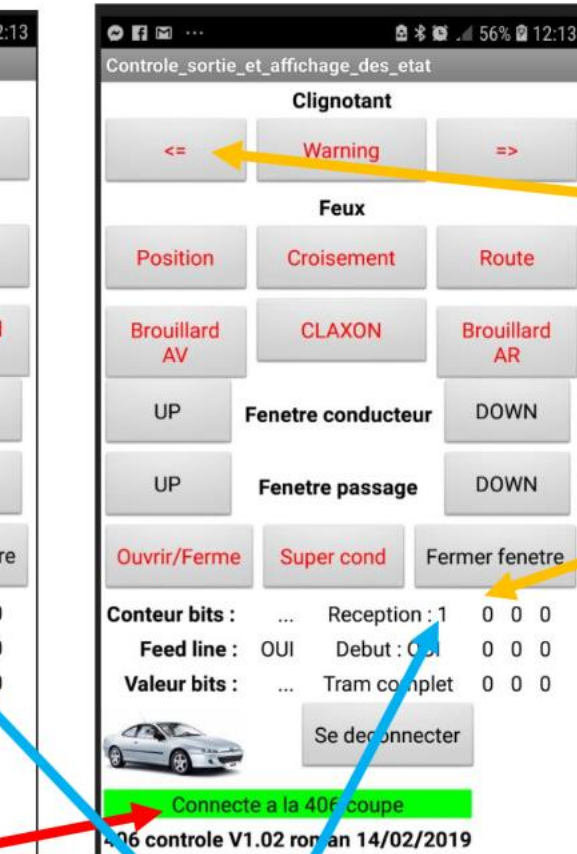
set message_etat_de_connection TextColor to black

Show Warnings

9 20

Pour que le téléphone se connecte au module Bluetooth lorsque le bouton **Se connecter** est appuyé, il faut se rendre sur la page **Controle_sortie** puis **Blocks** et en haut on trouve un block qui s'appelle **BluetoothClient1.Connect1** on remplace l'adresse MAC par la sienne.

Application téléphone



Indique état
d'une sortie
de arduino

Quand les données sont
envoyées par le téléphone
le message clignote

La valeur change successivement
pour indiquer que l'on reçoit
des données de la part d'Arduino

Indique si les données envoyées par
Arduino sont correctes
(je regarde si les données commencent
par un \n et finissent par \r)

Donne envoyer par le téléphone

0 12345 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21
\r 00000 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 \n

0 = \r (Début de tram)

1 = Bit qui change d'état a chaque envoyé des donne

2 = ID_Bit1

3 = ID_Bit2

4 = ID_Bit3

5 = ID_Bit4

6 = Clignotant gauche BUTTON STATE APPLICATION

7 = Clignotant droit BUTTON STATE APPLICATION

8 = Feux de route BUTTON STATE APPLICATION

9 = Feux de croisement BUTTON STATE APPLICATION

10 = Feux de position BUTTON STATE APPLICATION

11 = Brouillard avant BUTTON STATE APPLICATION

12 = Brouillard arrière BUTTON STATE APPLICATION

13 = Fenêtre conducteur monte BUTTON STATE APPLICATION

14 = Fenêtre conducteur décente BUTTON STATE APPLICATION

15 = Fenêtre passager monte BUTTON STATE APPLICATION

16 = Fenêtre passager décente BUTTON STATE APPLICATION

17 = Klaxon BUTTON STATE APPLICATION

18 = Ouvrir fermer la voiture BUTTON STATE APPLICATION

19 = Super condanation BUTTON STATE APPLICATION

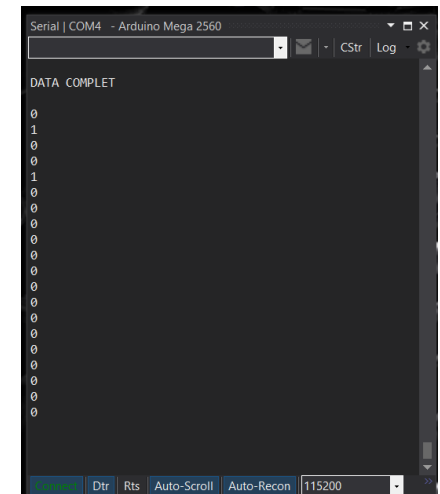
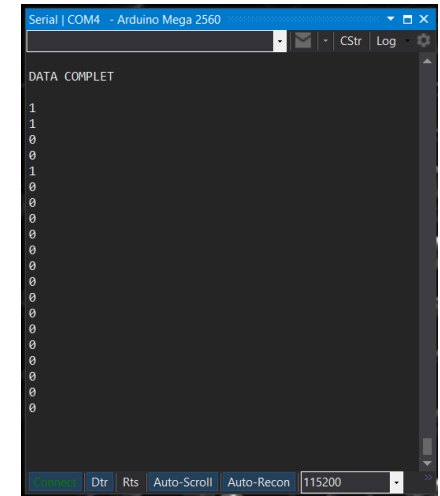
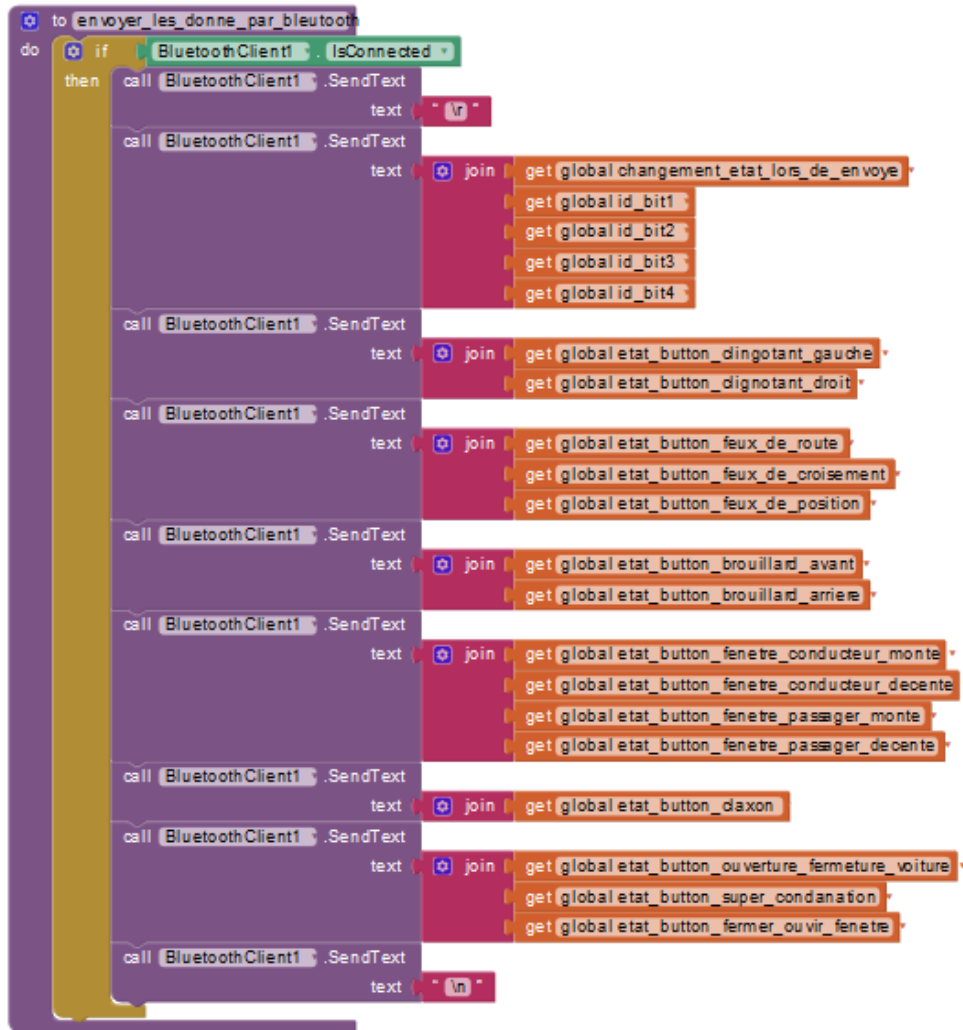
20 = Ouvrir fermer tout les fenêtre BUTTON STATE APPLICATION

21 = \n (fin de la tram)

Application téléphone

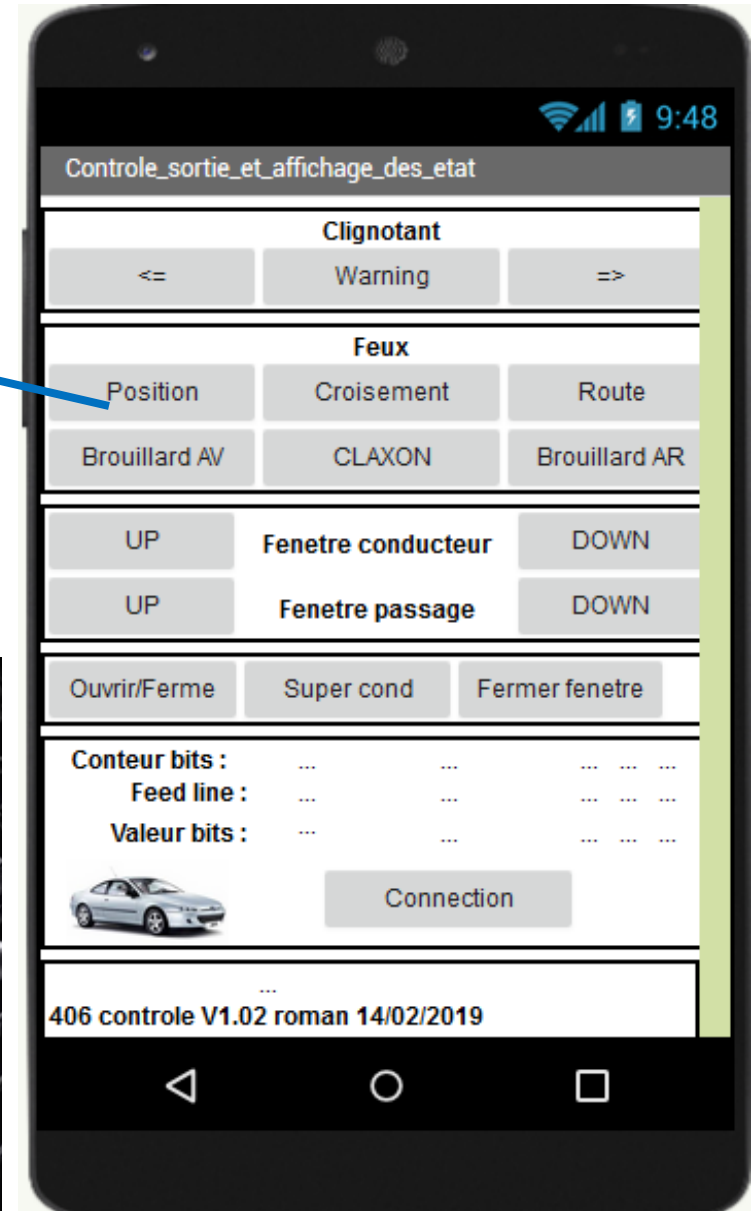
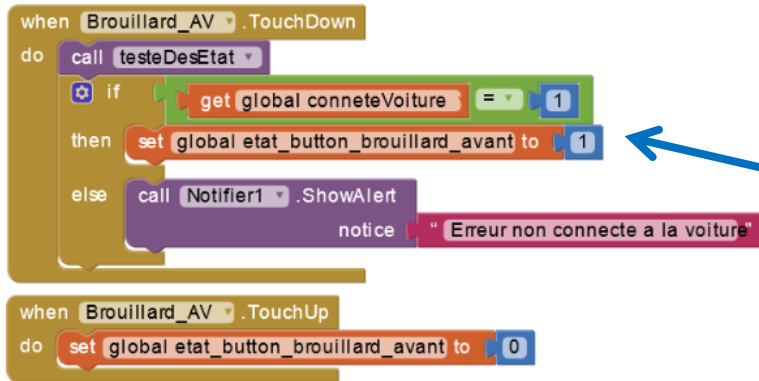
Le bloque qui envoyer les donne MIT App Inventor

Moniteur série de Arduino

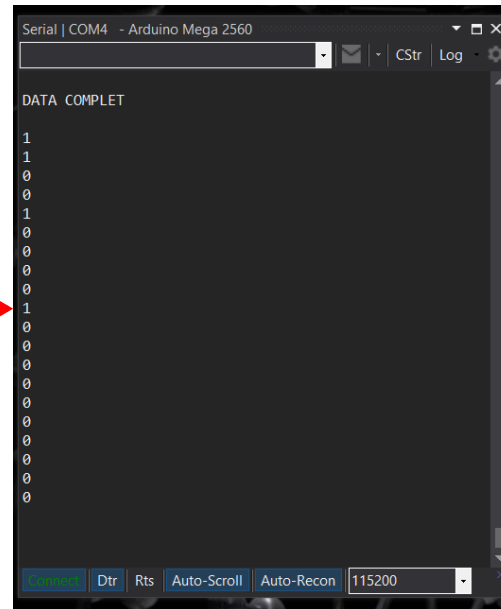


Application téléphone

Le bloque qui regarde l'état du bouton dans MIT App Inventor




J'ai appuyer sur un
Bouton de position
dans l'application téléphone



Donne envoyer par Arduino

A chaque fois que Arduino reçoit des données du téléphone et qu'il sont corrects il envoie l'état des sorties (clignotant, des feux ..)

Les données commencent par un `\n` et finissent par `\r`.

A chaque envoi l'état du bit 1 change 

`dataArray[6]` = Etat sortie clignotant gauche
`dataArray[7]` = Etat sortie clignotant droit
`dataArray[8]` = Etat sortie feux de position
`dataArray[9]` = Etat sortie feux de croisement
`dataArray[10]` = Etat sortie feux de route
`dataArray[11]` = Etat sortie feux de brouillard avant
`dataArray[12]` = Etat sortie feux de brouillard arrière
`dataArray[13]` = Etat voiture Ferme/Ouvert
`dataArray[14]` = Etat super condensation
`dataArray[15]` = Etat du klaxon

```
byte dataArray[20] = { 0 };
static byte a = 0;
//uint8_t dataArray[10];

if (a == 1)
{
    a = 0;
}
else {
    a = 1;
}

dataArray[0] = '\n'; // Nouvelle ligne

dataArray[1] = a; // a chaque envoi cette valeur doit changer

dataArray[2] = 1;
dataArray[3] = 0;
dataArray[4] = 1;
dataArray[5] = 1;

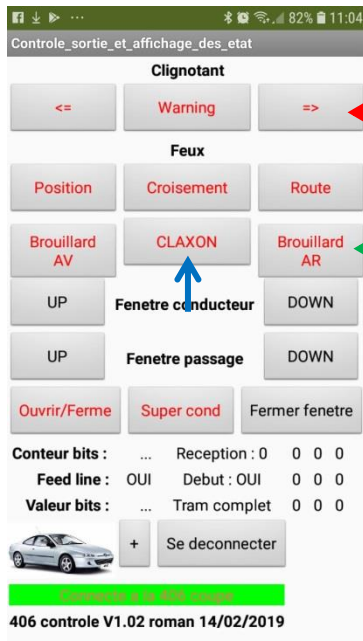
dataArray[6] = 0;
dataArray[7] = 0;
dataArray[8] = 0;
dataArray[9] = 0;
dataArray[10] = 0;
dataArray[11] = 0;
dataArray[12] = 0;
dataArray[13] = 0;
dataArray[14] = 0;
dataArray[15] = 0;
dataArray[16] = 0;
dataArray[17] = 0;
dataArray[18] = 0;
dataArray[19] = '\r'; //'\r' retour chariot
//dataArray[10] = 11; //'\n'

Serial3.write(dataArray, 20);
```

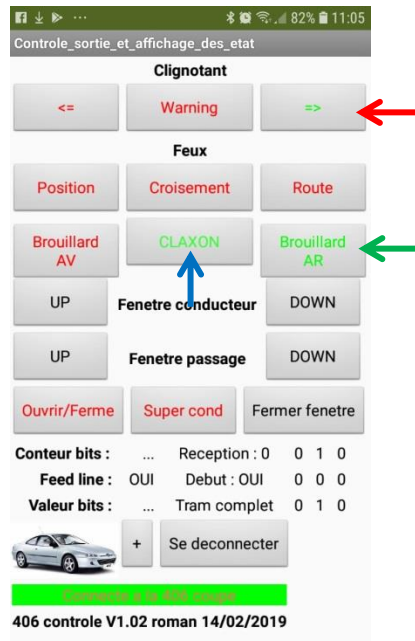
Donne envoyer par Arduino

J'ai change manuellement la valeur des bit a envoyer par Arduino et on vois le résultant dans l'application.

0 = rouge , 1 = vert



AVANT



APRES

```
dataArray[0] = '\n'; // Nouvelle ligne
dataArray[1] = a; // a chaque envoi
dataArray[2] = 1;
dataArray[3] = 0;
dataArray[4] = 1;
dataArray[5] = 1;

dataArray[6] = 0;
dataArray[7] = 1;
dataArray[8] = 0;
dataArray[9] = 0;
dataArray[10] = 0;
dataArray[11] = 0;
dataArray[12] = 1;
dataArray[13] = 0;
dataArray[14] = 0;
dataArray[15] = 1;
dataArray[16] = 0;
dataArray[17] = 0;
dataArray[18] = 0;
dataArray[19] = '\r'; // '\r' retour chariot
```