

## SOMMAIRE

[1 Description de l'environnement](#)

[2 Base programmable compatible arduino](#)

[3 Programmation en langage graphique Scratch™](#)

[4 - programmation série avec un protocole ORION](#)

[5 - Piloter des commandes sur le robot avec le protocole ORION](#)

[6 - PILOTER les moteurs du mbot, comment faire?](#)

[7 - PILOTER les deux LEDS du mbot, comment faire?](#)

[8 - piloter l'avertisseur sonore](#)

[9 - RECEVOIR les informations d'un capteur](#)

[10 - Lire les valeurs du capteur ultra-son](#)

[11 - Lire les valeurs du capteur de lumière](#)

[12 - Lire les valeurs du capteur de ligne](#)

[13 Réaliser le programme pour piloter le mbot avec app inventor.](#)

[14 - L'interface pour pouvoir connecter la tablette au robot en Bluetooth](#)

[15 - Piloter le robot grâce aux requêtes envoyées.](#)

[16 Lire les valeurs des capteurs du Mbot avec une application pour les afficher](#)

[17 Afficher les valeurs du capteur ultra-son](#)

[18 Afficher les valeurs du capteur de lumière](#)

[19 Afficher les valeurs du capteur de suiveur de ligne](#)

[20 L'application MbotLED](#)

[21 L'application Mbot Pilot 1.0](#)

[22 L'application Mbot Pilot capteur 2.0](#)

[23 Conclusion](#)

## 1 Description de l'environnement

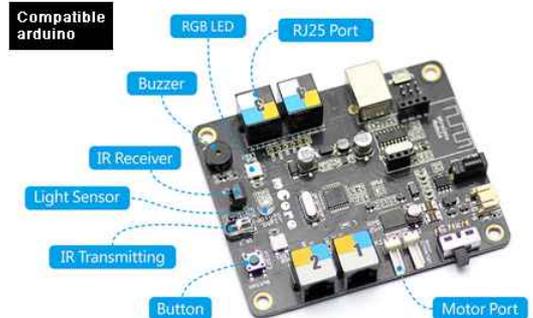
Le robot Mbot (Bluetooth Version) est un robot avec une platine principale architecturée sur un cœur compatible arduino UNO-328 (base Atmega328).



## 2 Base programmable compatible arduino

La platine dispose d'une prise USB afin que vous puissiez la programmer via l'environnement de l'Arduino. Elle dispose également:

- D'un buzzer
- De 2 Leds RGB
- D'une Led de transmission IR
- D'une Led de réception IR
- D'un capteur de lumière
- D'un bouton poussoir
- De 2 connecteurs permettant de la relier aux moteurs du robot
- D'un module Bluetooth™ embrochable (livré)
- D'un interrupteur "M/A"
- D'un connecteur permettant de la relier à un support de pile (le support est livré)
- D'un connecteur pour la relier à une batterie LiPo (batterie non livrée)
- De 4 connecteurs RJ45 permettant de lui adjoindre des modules d'extension.

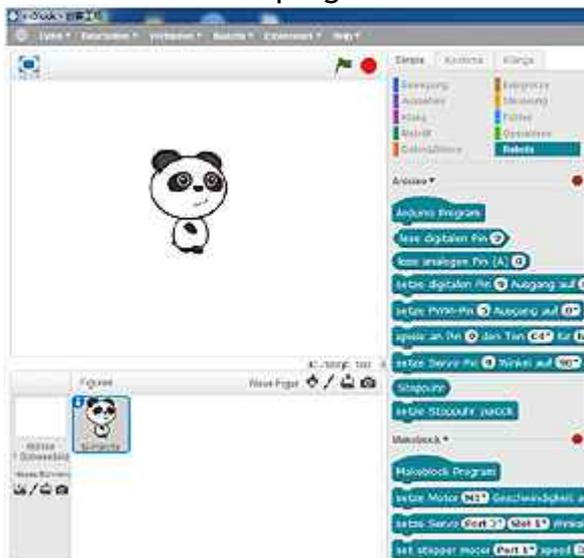


Sont également livré avec le robot "mBot":

- Un module de détection ultrason
- Un module de suivie de ligne au sol
- Une télécommande IR (pile à ajouter)

## 3 Programmation en langage graphique Scratch™

Votre robot "mBot" est facilement programmable en langage graphique Scratch™

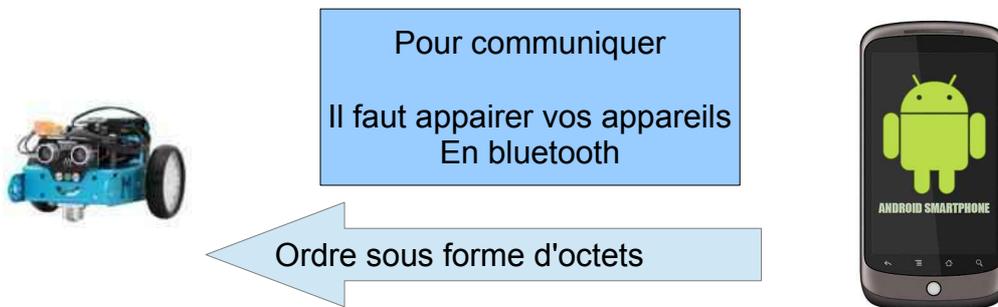


## 4 - programmation série avec un protocole ORION

Le Mbot possède aussi un protocole de communication ORION qui permet de communiquer entre une interface et la platine du robot.

C'est cette communication qui va nous intéresser. Certes, elle est un peu complexe mais elle va nous permettre de communiquer sans charger un programme dans le Mbot puisqu'il existe déjà.

Pour communiquer, il faut juste appairer votre robot à une tablette ou smartphone.



Le protocole ORION est basé sur un codage de plusieurs octets qui vont réaliser des instructions au robot pour qu'il puisse soit activer des commandes, soit envoyer des valeurs des capteurs par exemple. On utilisera ce protocole série pour communiquer en Bluetooth, c'est exactement la même chose.

## 5 - Piloter des commandes sur le robot avec le protocole ORION

Afin de piloter notre robot il nous faudra envoyer des commandes sous forme de plusieurs octets qui ont des significations précises.

## 6 - PILOTER les moteurs du mbot, comment faire?



Il faut envoyer plusieurs octets au robot en série ou en Bluetooth.

J'ai mis dans le tableau le code **hexadécimal** et l'équivalent en **décimal**. On verra plus tard que l'on utilisera uniquement le code décimal

Description	begin_code_1	begin_code_2	longueur	indice	action	dispositif	Port	speed_low	speed_high
Moteur gauche dans le sens avancer	Code hex FF	Code hex 55	Code hex 06	0	2	Code hex 0a	9	Code hex 00	Code hex FF
	code déc 255	code déc 85	code déc 06	0	2	code déc 10	9	code déc 0	code déc 255
Moteur gauche dans le sens arrière	Code hex FF	Code hex 55	Code hex 06	0	2	Code hex 0a	9	Code hex FF	Code hex 00
	code déc 255	code déc 85	code déc 06	0	2	code déc 10	9	code déc 255	code déc 0
Moteur droit dans le sens avancer	Code hex FF	Code hex 55	Code hex 06	0	2	Code hex 0a	10	Code hex FF	Code hex 00
	code déc 255	code déc 85	code déc 06	0	2	code déc 10	10	code déc 255	code déc 0
Moteur droit dans le sens reculer	Code hex FF	Code hex 55	Code hex 06	0	2	Code hex 0a	10	Code hex 00	Code hex FF
	code déc 255	code déc 85	code déc 06	0	2	code déc 10	10	code déc 0	code déc 255

En réalisant des tests, je me suis aperçu que lorsque l'on envoie une commande au Mbot, celui-ci renvoie 2 octets ( C'est un constat ). Cela peut engendrer des erreurs surtout si on demande des informations sur les capteurs. J'ai trouvé une solution ( Détails plus loin ) qui permet de gérer l'erreur qui en résulte.

## 7 - PILOTER les deux LEDS du mbot, comment faire?

Avec le même principe on peut aussi piloter les deux LEDS RGB du Mbot.

R= RED = Rouge (255 = rouge vif, 0 = noir )

G = Green = Vert (255 = vert vif, 0 = noir )

B = Blue = Bleu (255 = bleu vif, 0 = noir )

Si je mets les 3 valeurs à 255 alors la led est blanche

Si je mets les 3 valeurs à 0, ce sera très peu allumé.



Description	begin_code_1	begin_code_2	longueur	indice	action	dispositif	Port	fente	position	R	G	B
Allumer les 2 LED selon les valeurs RGB	Code hex FF	Code hex 55	Code hex 09	0	2	Code hex 08	7	2	0			
	code déc 255	code déc 85	code déc 9	0	2	code déc 8	7	2	0	R	G	B
Allumer la LED gauche selon les valeurs RGB	Code hex FF	Code hex 55	Code hex 09	0	2	Code hex 08	7	2	1			
	code déc 255	code déc 85	code déc 9	0	2	code déc 8	7	2	1	R	G	B
Allumer la LED droite selon les valeurs RGB	Code hex FF	Code hex 55	Code hex 09	0	2	Code hex 08	7	2	2			
	code déc 255	code déc 85	code déc 9	0	2	code déc 8	7	2	2	R	G	B



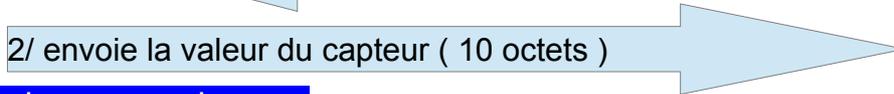
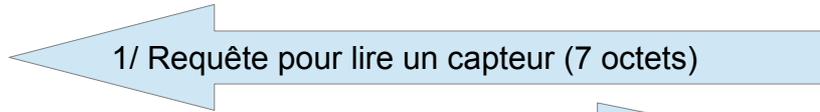
## 8 - piloter l'avertisseur sonore

Description	begin_code_1	begin_code_2	longueur	indice	action	dispositif	Tone Low	Tone High	Beat Low	Beat High
Faire sonner l'avertisseur	Code hex FF	Code hex 55	Code hex 07	0	2	Code hex 22			Code hex FF	Code hex 00
	code déc 255	code déc 85	code déc 7	0	2	code déc 34	TL	TH	code déc 250	code déc 0

Il vous suffit de modifier les valeurs Ton Low et Tone high pour changer la tonalité.

**9 - RECEVOIR les informations d'un capteur**

Afin de recevoir les informations d'un capteur sur le Mbot, je dois envoyer une requête au Mbot pour qu'il puisse m'envoyer les informations de la valeur du capteur.



**10 - Lire les valeurs du capteur ultra-son**

J'envoie une requête au robot pour lui demander qu'il me renvoie la valeur

Requête à Envoyer



Description	begin_code_1	begin_code_2	longueur	indice	action	dispositif	Port
Demander les valeurs du capteur ultra-son	Code hex FF	Code hex 55	Code hex 04	02	01	Code hex 01	03
	code déc 255	code déc 85	code déc 4	2	1	code déc 1	3

Recevoir les informations du capteur

Description	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	Octet 8	Octet 9	Octet 10
Recevoir les informations	Code hex FF	Code hex 55	Code hex 02	Code hex 02					Code hex 0a	Code hex 0d
	code déc 255	code déc 85	code déc 2	code déc 2	Val 1	Val 2	Val 3	Val 4	code déc 10	code déc 13

Ensuite le calcul est le suivant: Résultat = (0xVAL1<<24) + (0xVAL4<<16)+(0xVAL2<<8)+0xVAL3  
 Ox, c'est la valeur hexadécimal, et <<24 correspond à la place de l'octet car c'est une valeur sur 32 bits.

**11 - Lire les valeurs du capteur de lumière**

J'envoie une requête au robot pour lui demander qu'il me renvoie la valeur

Requête à Envoyer



Description	begin_code_1	begin_code_2	longueur	indice	action	dispositif	Port
Demander les valeurs du capteur ultra-son	Code hex FF	Code hex 55	Code hex 04	05	01	Code hex 03	03
	code déc 255	code déc 85	code déc 4	5	1	code déc 3	3

Recevoir les informations du capteur

Description	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	Octet 8	Octet 9	Octet 10
Recevoir les informations	Code hex FF	Code hex 05	Code hex 02	Code hex 55					Code hex 0a	Code hex 0d
	code déc 255	code déc 05	code déc 2	code déc 85	Val 1	Val 2	Val 3	Val 4	code déc 10	code déc 13

Ensuite le calcul est le suivant: Résultat = (0xVAL1<<24) + (0xVAL4<<16)+(0xVAL2<<8)+0xVAL3  
 Ox, c'est la valeur hexadécimal, et <<24 correspond à la place de l'octet car c'est une valeur sur 32 bits

12 - Lire les valeurs du capteur de ligne

J'envoie une requête au robot pour lui demander qu'il me renvoie la valeur

Requête à Envoyer

Description	begin_code_1	begin_code_2	longueur	indice	action	dispositif	Port
Demander les valeurs du capteur ultra-son	Code hex FF	Code hex 55	Code hex 04	hex 60	01	Code hex 11	02
	code déc 255	code déc 85	code déc 4	96	1	code déc 17	2

Recevoir les informations du capteur

Le capteur de ligne renvoie 4 informations différentes.

Soit les 2 cellules renvoient une information pour dire que la ligne est blanche (64,64)

Soit la cellule droite renvoie une ligne noire et celle de gauche une ligne blanche (0,64)

Soit la cellule droite renvoie une ligne blanche et celle de gauche une ligne noire (128,63)

Soit les 2 cellules renvoient une information pour dire que la ligne est noire. (0,0)

Description	Octet 1	Octet 2	Octet 3	Octet 4	Octet 5	Octet 6	Octet 7	Octet 8	Octet 9	Octet 10
2 Capteurs allumés Ligne blanche	Code hex FF code déc 255	Code hex 60 code déc 96	Code hex 02 code déc 2	Code hex 55 code déc 85	00	00	Code hex 40 code déc 64	Code hex 40 code déc 64	Code hex 0d code déc 13	Code hex 0a code déc 10
Gauche allumée Droite éteinte							Code hex 00 code déc 0	Code hex 40 code déc 64		
Gauche éteinte Droite allumée							Code hex 80 code déc 128	Code hex 3F code déc 63		
2 capteurs éteints Ligne noire							Code hex 00 code déc 0	Code hex 0 code déc 0		

Les deux LED sont éteintes qui indiquent qu'il y a une ligne noire. Le Mbot quand on lui demande renvoie les valeurs 0 et 0 sur les octets 7 et 8.



Ici la led droite est allumée, donc c'est blanc. Le capteur renvoie les valeurs 128 et 63 sur les octets 7 et 8

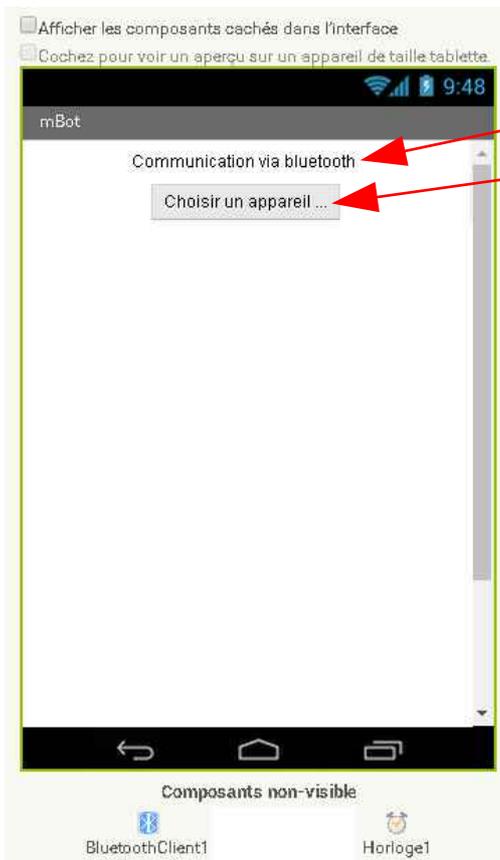
Les 2 Leds sont allumées, donc c'est blanc. Le capteur renvoie 64 et 64 sur les octets 7 et 8

La led gauche est allumée, la droite est éteinte. Le capteur renvoie la valeur 0 et 64 sur les octets 7 et 8.

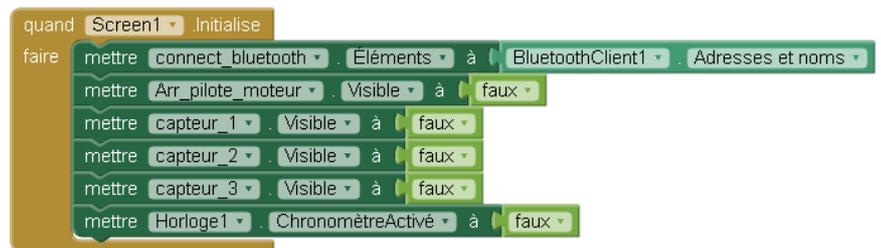
## 13 Réaliser le programme pour piloter le mbot avec app inventor.

- Aller sur app inventor <http://ai2.appinventor.mit.edu/>
- Il existe une version à installer
- sur la version à installer pas besoin de compte google.
- Commencer un nouveau projet.
- Vous devez créer votre interface sur designer.

## 14 - L'interface pour pouvoir connecter la tablette au robot en Bluetooth

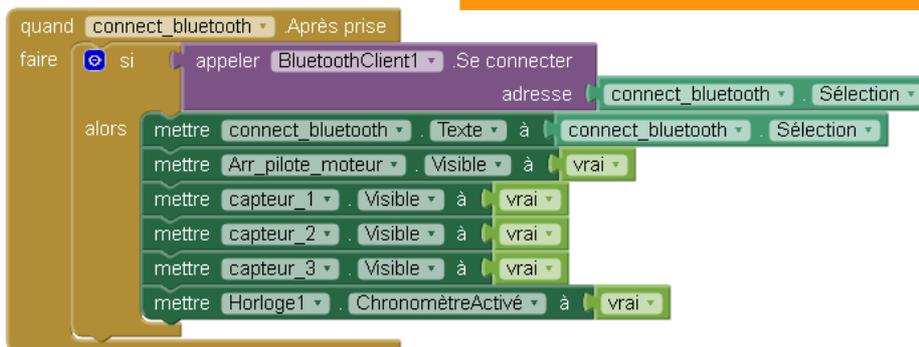


Créer un label pour écrire «communication via Bluetooth»  
Un bouton qui s'appelle connect-bluetooth



Quand l'écran s'initialise  
Connecter le bluetooth à l'adresse bluetooth  
Mettre les autres éléments visibles à faux pour éviter qu'il s'affiche à l'écran. Comme cela on évite que quelqu'un appuie pour demander des informations sans être connecté.

mettre un outil de connectivité BluetoothClient1



Quand j'appuie sur le bouton choisir un appareil, alors  
Si j'ai choisi mon appareil bluetooth dans la sélection de mon appareil  
Alors j'affiche le nom de mon appareil connecté  
Je mets tous mes éléments visibles à vrai pour qu'il s'affiche

## 15 - Piloter le robot grâce aux requêtes envoyées.

créer une interface dans un tableau avec des boutons. Pour rendre plus sympathique l'interface, mettre des images dans les boutons.



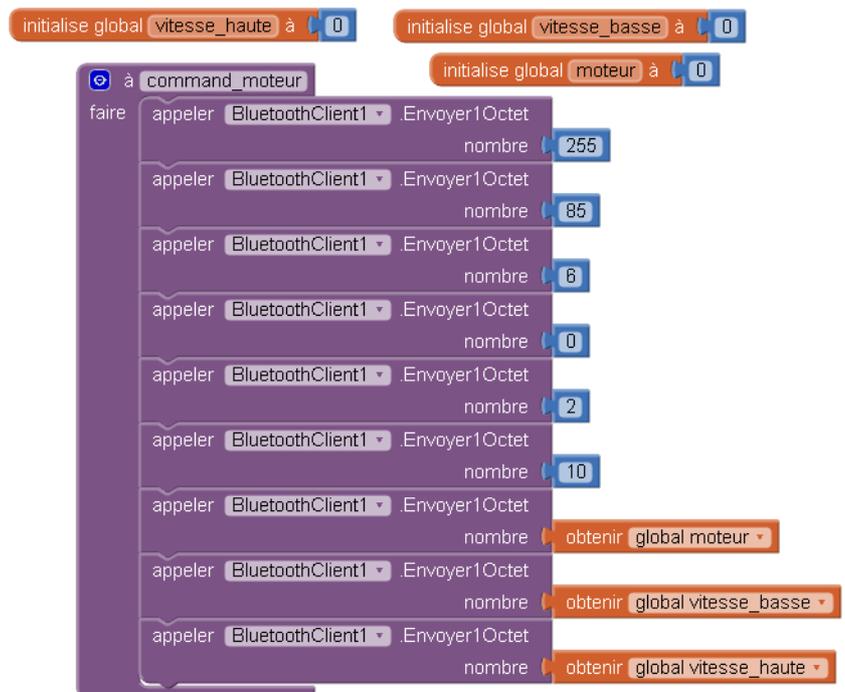
Créer une procédure ou sous-programme pour envoyer les 9 octets qui permettent de commander les moteurs du robot.

Pour cela créer 3 variables:

- Moteur qui indique quel moteur je veux piloter  
9 c'est le gauche, 10 c'est le droit

Vitesse basse pour indiquer la vitesse basse  
vitesse haute pour la vitesse haute

ces 2 dernières variables vont permettre de définir le sens (voir le tableau au chapitre 6)



Pour éviter d'écrire les mêmes programmes pour chaque commande, créer des procédures sur lesquelles on va jouer avec les variables.

Description		Moteur	Vitesse basse	Vitesse haute
Avancer	moteur gauche ↺	9	0	255
	moteur droit ↻	10	255	0
reculer	moteur gauche ↻	9	255	0
	moteur droit ↺	10	0	255
STOP	Moteurs gauche 9 et droit 10	9 et 10	0	0

```

à Avant_mot_gauche
faire
mettre global moteur à 9
mettre global vitesse_basse à 0
mettre global vitesse_haute à 255

à Avant_mot_droit
faire
mettre global moteur à 10
mettre global vitesse_basse à 255
mettre global vitesse_haute à 0

à STOP
faire
mettre global vitesse_basse à 0
mettre global vitesse_haute à 0

à Recul_mot_droit
faire
mettre global moteur à 10
mettre global vitesse_basse à 0
mettre global vitesse_haute à 255

à Recul_mot_gauche
faire
mettre global moteur à 9
mettre global vitesse_basse à 255
mettre global vitesse_haute à 0
    
```

Définir des actions pour chaque bouton du pupitre de commande.

```

quand droite Clic
faire
appeler STOP
mettre global moteur à 10
appeler command_moteur
appeler Avant_mot_gauche
appeler command_moteur

quand gauche Clic
faire
appeler STOP
mettre global moteur à 9
appeler command_moteur
appeler Avant_mot_droit
appeler command_moteur

quand stop Clic
faire
appeler STOP
mettre global moteur à 9
appeler command_moteur
mettre global moteur à 10
appeler command_moteur

quand avancer Clic
faire
appeler Avant_mot_gauche
appeler command_moteur
appeler Avant_mot_droit
appeler command_moteur

quand reculer Clic
faire
appeler Recul_mot_gauche
appeler command_moteur
appeler Recul_mot_droit
appeler command_moteur

quand avancer_droite Clic
faire
appeler Avant_mot_droit
mettre global vitesse_basse à 125
appeler command_moteur
appeler Avant_mot_gauche
appeler command_moteur

quand avancer_gauche Clic
faire
appeler Avant_mot_droit
appeler command_moteur
appeler Avant_mot_gauche
mettre global vitesse_basse à 125
appeler command_moteur

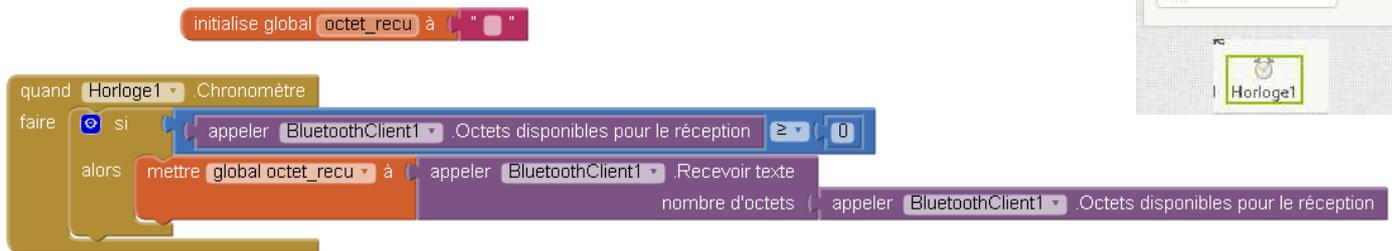
quand reculer_droite Clic
faire
appeler Recul_mot_droit
mettre global vitesse_basse à 125
appeler command_moteur
appeler Recul_mot_gauche
appeler command_moteur

quand reculer_gauche Clic
faire
appeler Recul_mot_droit
appeler command_moteur
appeler Recul_mot_gauche
mettre global vitesse_basse à 125
appeler command_moteur
    
```

En réalisant des tests, je me suis aperçu que lorsque l'on envoie une commande au Mbot, celui-ci renvoie 2 octets (C'est un constat). Cela peut engendrer des erreurs surtout si on demande des informations sur les capteurs. J'ai trouvé une solution (Détails plus loin) qui permet de gérer l'erreur qui en résulte.

On utilise la fonction chronomètre.  
(c'est paramétré dans le designer).

On observe toutes les 500ms si on reçoit une donnée sur le port Bluetooth  
Si une donnée est reçue on la stocke dans une variable.



16. Lire les valeurs des capteurs du Mbot avec une application pour les afficher

Afin de recevoir les informations d'un capteur du Mbot, je dois envoyer une requête au Mbot pour qu'il puisse m'envoyer les informations de la valeur du capteur.



1/ Requête pour lire un capteur (7 octets)

2/ envoie la valeur du capteur ( 10 octets )



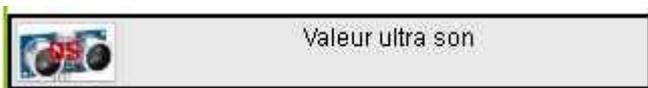
Un bouton INIT permet de remettre les variables à 0.



```

quand init .Clic
faire
mettre global octet à 0
mettre Val_US Texte à 0
mettre Val_lum Texte à 0
mettre val_SL Texte à 0
    
```

17. Afficher les valeurs du capteur ultra-son



Ici, un bouton qui lorsque j'appuie dessus demande au mbot d'envoyer la valeur du capteur ultrason.

```

à Appel_val_ultrason
faire
appeler BluetoothClient1 Envoyer1Octet nombre 255
appeler BluetoothClient1 Envoyer1Octet nombre 85
appeler BluetoothClient1 Envoyer1Octet nombre 4
appeler BluetoothClient1 Envoyer1Octet nombre 2
appeler BluetoothClient1 Envoyer1Octet nombre 1
appeler BluetoothClient1 Envoyer1Octet nombre 1
appeler BluetoothClient1 Envoyer1Octet nombre 3
    
```

```

initialise global bt_us à ""
initialise global US_1 à 0
initialise global US_2 à 0
initialise global US_4 à 0
initialise global US_3 à 0
initialise global US à 0

quand Ultrason .Clic
faire
appeler Appel_val_ultrason
si BluetoothClient1 Disponible
alors
mettre global octet à appeler BluetoothClient1 Recevoir octet non signé nombre d'octets 4
mettre global US_1 à appeler BluetoothClient1 Recevoir octet non signé nombre d'octets 1
mettre global US_2 à appeler BluetoothClient1 Recevoir octet non signé nombre d'octets 1
mettre global US_3 à appeler BluetoothClient1 Recevoir octet non signé nombre d'octets 1
mettre global US_4 à appeler BluetoothClient1 Recevoir octet non signé nombre d'octets 1
mettre global octet à appeler BluetoothClient1 Recevoir octet non signé nombre d'octets 2
appeler convertir_val_US
    
```

J'envoie ce protocole, c'est à dire 7 octets pour lui demander qu'il me renvoie la valeur du capteur ultrason

Quand j'appuie sur le bouton alors

- J'appelle la procédure appel\_val\_ultrason
- Si une valeur est disponible sur le port Bluetooth
- Alors

- Mettre les 4 premiers octets dans une variable
- mettre octet (5) dans US1
- mettre octet (6) dans US2
- mettre octet (7) dans US3
- mettre octet (8) dans US4
- Mettre les deux octets dans une variable.
- Appeler convertir\_val\_US (ici on va simplement afficher les valeurs à l'écran)

```

à convertir_val_US
faire
mettre global bt_us à joint obtenir global US_1
obtenir global US_4
obtenir global US_2
obtenir global US_3
mettre Val_US Texte à obtenir global bt_us
    
```

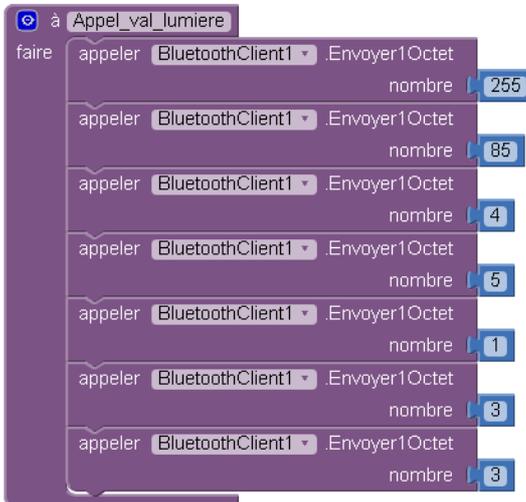
Ensuite le calcul est le suivant: Résultat = (0xVAL1<<24) + (0xVAL4<<16)+(0xVAL2<<8)+0xVAL3  
 Ox, c'est la valeur hexadécimal, et <<24 correspond à la place de l'octet car c'est une valeur sur 32 bits

## 18. Afficher les valeurs du capteur de lumière

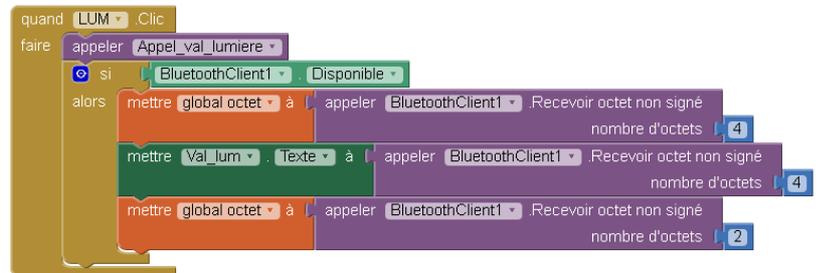
C'est le même principe que pour le capteur ultrason mais avec une autre valeur pour le protocole



Ici, un bouton qui lorsque j'appuie dessus demande au mbot d'envoyer la valeur du capteur de lumière



J'envoie ce protocole, c'est à dire 7 octets pour lui demander qu'il me renvoie la valeur du capteur de lumière



Quand j'appuie sur le bouton lum alors

- J'appelle la procédure appel\_val\_lumiere

Si une valeur est disponible sur le port Bluetooth

Alors

- Mettre les 4 premiers octets dans une variable
- J'affiche dans la case Val\_lum, les 4 octets reçus.
- Mettre les deux octets dans une variable.

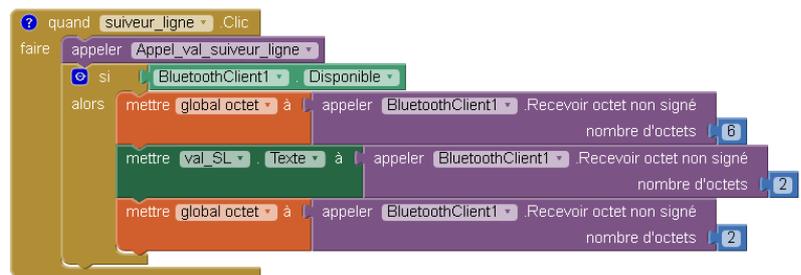
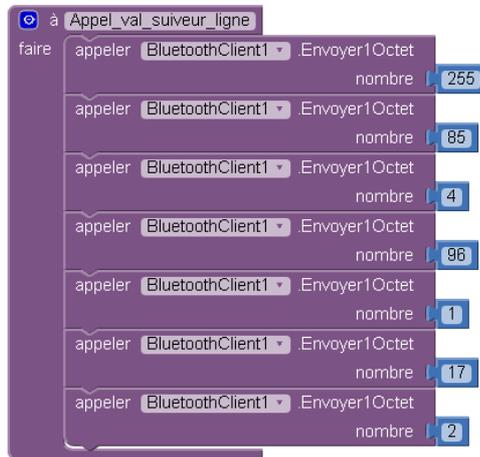
Ensuite le calcul est le suivant: Résultat = (0xVAL1<<24) + (0xVAL4<<16)+(0xVAL2<<8)+0xVAL3  
Ox, c'est la valeur hexadécimal, et <<24 correspond à la place de l'octet car c'est une valeur sur 32 bits

## 19. Afficher les valeurs du capteur de suiveur de ligne



Valeur suiveur de ligne

Ici, un bouton qui lorsque j'appuie dessus demande au mbot d'envoyer la valeur du capteur



J'envoie ce protocole, c'est à dire 7 octets pour lui demander qu'il me renvoie la valeur du capteur suiveur de ligne

Quand j'appuie sur le bouton suiveur\_ligne alors  
- J'appelle la procédure appel\_val\_suiveur\_ligne  
Si une valeur est disponible sur le port Bluetooth  
Alors

- Mettre les 6 premiers octets dans une variable
- J'affiche dans la case Val\_SL, les 2 octets reçus.
- Mettre les deux octets dans une variable.

## Recevoir les informations du capteur

Le capteur de ligne renvoie 4 informations différentes.

Soit les 2 cellules renvoient une information pour dire que la ligne est blanche (64,64)

Soit la cellule droite renvoie une ligne noire et celle de gauche une ligne blanche (0,64)

Soit la cellule droite renvoie une ligne blanche et celle de gauche une ligne noire (128,63)

Soit les 2 cellules renvoient une information pour dire que la ligne est noire. (0,0)

## 20. L'application MbotLED

Vous retrouverez plusieurs applications.

L'application mbotLED qui permet uniquement de piloter les LED

Inspirer fortement de <http://webtoolsreview.blogspot.fr/2016/04/programming-mbot-with-app-inventor-2.html>

L'application permet de piloter les 2 LEDS du Mbot en déplaçant des curseurs RGB sur l'écran.

On envoie une requête sur le Mbot, même principe que pour les moteurs.

Quand j'appuie sur le bouton rouge  
Alors J'appelle la procedure command  
J'envoie sur les 3 derniers octets  
Les 3 valeurs de mes curseurs qui définissent  
Les valeurs RGB

Si Led =0 les deux vont s'allumer  
Si LED=1 Gauche  
Si LED = 2 Droite

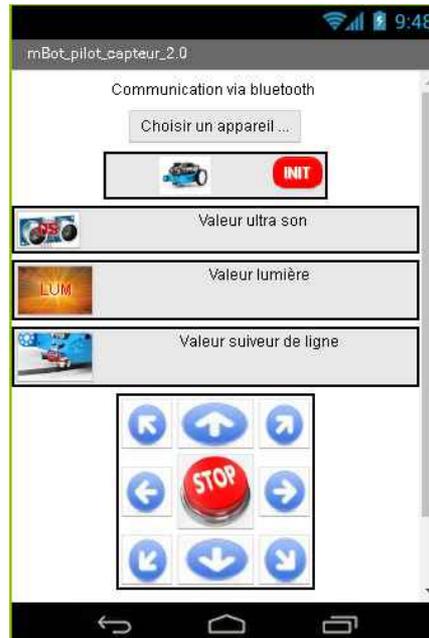
## 21. L'application Mbot Pilot 1.0

Cette application permet de piloter les 3 actionneurs du Mbot (Moteurs, LED et avertisseur)

Un curseur permet de changer la  
Valeur de la tonalité.

## 22. L'application Mbot Pilot\_capteur 2.0

Cette application permet de piloter le robot et de recevoir les informations des 3 capteurs.



## 23. Conclusion

Maintenant à vous de jouer. Certaines fonctions sont trop compliquées pour des élèves de 3ème, il suffit dans ce cas de donner des procédures déjà faites. On peut dans ce cas utiliser les procédures de pilotage déjà faites pour les moteurs.

Il faudra aussi réaliser des procédures pour convertir les 4 octets des capteurs dans une seule variable.