APP MIT Inventor:



[**I. Introduction à MIT App inventor: 3**](#_k83bo464r1rc)

[**II. Mise en ligne d’une application MIT app inventor: 3**](#_4wvxpq67vvpf)

[**III. Page d'accueil 3**](#_hx3i7butz7m1)

[A. Screen 3](#_obtfy7411icg)

[B. Blocs 4](#_m5fui7yrp7fu)

[**IV. Télécommande 4**](#_imc616ut9cms)

[A. Screen: 4](#_opfqcugpv1hl)

[B. Blocs: 5](#_x6ua7iqmnl7o)

[1. Bloc d’initialisation: 5](#_uupurx5va7u7)

[2. Changement de l’adresse IP de la bouée: 6](#_r81oh56xmw5x)

[3. Ajout du numéro de bouée à la fin de l’adresse IP, en fonction du choix de bouée 7](#_vkam9ft9df0f)

[4. Contrôler l’envoie des données: 7](#_2s2941490gu1)

[5. Envoie des blocs de caractères correspondant aux voies de la bouée: 7](#_oq1msvhkyqu)

[6. Envoie des blocs de caractères correspondant au “Bouton ON” de la bouée: 8](#_prdupepjmofu)

[7. Récupération des données du joystick et mise à l’échelle des valeurs: 9](#_ycj2ubd8xpz1)

[8. Modélisation des switchs de la télécommande: 9](#_xrq7dn8of57n)

[9. Récupération des données de la voies 1 à 4: 10](#_1lrezio5iqdo)

[**V. Map 12**](#_h7ufqe2e06rr)

[A. Screen: 12](#_19l7fbxwiab0)

[B. Blocs: 12](#_er9bjtgnonew)

[1. Initialisation des paramètres 13](#_4ssd5yfkd7sf)

[2. Création d’un point GPS cible sur la carte: 13](#_al6r8enomttr)

[3. Suppression d’un marqueur cible: 14](#_lvovndniadrt)

[4. Gestion de l’adressage IP de la bouée: 15](#_h9nbin2aqkd6)

[5. Envoi du train de caractère pour récupérer le numéro de bouée et les coordonnées GPS de la bouée: 16](#_tzic518qpy)

# Introduction à MIT App inventor:

Afin d’avoir une interface homme-machine, nous avons choisi de travailler avec MIT APP inventor, qui permet de créer de façon intuitive et efficace une application.

Voici le lien permettant de créer une application Androïd gratuitement et en ligne:

<https://appinventor.mit.edu/>

Cette application à un rôle de capteur. C’est à dire que lorsqu'un ordre à la bouée est donnée sous la forme d’une interaction avec l’application, l’application envoie à la bouée des “Blockchain” ou un train de chaîne de caractère à la bouée. Le traitement de ces informations se fait ensuite directement à travers la carte Arduino afin de transformer les ordres en action réalisé par la bouée.

Pour reconnaître quel type d’ordre provient de la bouée, nous avons mis des “mots clés”, en début de chaîne, ce qui nous permettra de repérer le type de chaîne envoyée.

# Mise en ligne d’une application MIT app inventor:

Vous retrouverez l'ensemble de la procédure de mise en ligne de l’application Androïd sur le lien suivant: <https://fr.yeeply.com/blog/publier-application-google-play-store/>

Pour résumer les étapes, il faut:

1. Créer une compte développeur sur google play (coût unique de 25 USD)
2. cliquer sur “toutes les applications” puis “créer une application”
3. Créer et remplir une liste Play Stores (avec les informations que les utilisateurs verront)
4. Télécharger le document apk de l’application et choisir le type de publication (test interne, fermé, ou ouvert)
5. Définir la classification du contenu de l’application (primordial car si non classifié, l’application pourrait être retiré)
6. Définir le prix et la distribution de votre application
7. Soumettre l’application à un examen (il faut attendre un contrôle vert sur chaque section)
8. Attendre que la version de déploiement soit disponible (cela peut prendre quelques jours)

# Page d'accueil

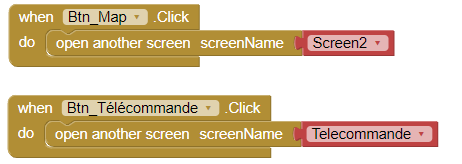
La page d’accueil a plus une fonction graphique que de commande de la bouée. Elle permet de présenter l’application et les personnes partenaires de l’application. Elle permet également de faire le lien avec la page où se trouve la télécommande et la page où se trouve la carte.

## Screen



## Blocs

Cette fenêtre n’ayant pas beaucoup de fonctionnalités, peut de blocs y sont associés. Seuls les blocs de changement de screen existent. La screen 2 correspond à la page map.



# Télécommande

## Screen:

Voici à quoi ressemble l’interface de l’application sur l'écran télécommande. En légende sont inscrites toutes les appellations utilisées dans la partie des blocs.

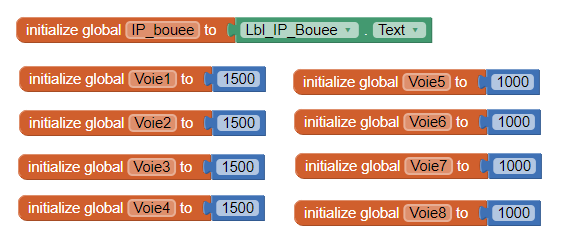


## Blocs:

### Bloc d’initialisation:

Ce bloc à pour objectif d’initialiser toutes les variables lors du lancement de l’application, ou lors de l’ouverture de nouvelle fenêtre.

Tout d’abord nous initialisons toutes les variables globales:



Chaque voie correspond à une voie physique d’une télécommande classique.

Puis nous initialisons les variables lié à la télécommande:



### Changement de l’adresse IP de la bouée:





### Ajout du numéro de bouée à la fin de l’adresse IP, en fonction du choix de bouée

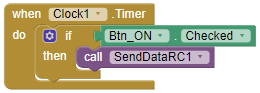


!! Pour le moment les chiffres ne viennent pas se remplacer mais s’ajouter. Ainsi pour changer de numéro de bouée, il faut d’abord changer l’IP grâce au bouton, puis choisir le numéro de bouée.

### Contrôler l’envoie des données:

Afin de contrôler l’envoie des données, et éviter la saturation des informations envoyées à la carte de programmation lors de l’utilisation de la télécommande présente sur l’application, nous avons utilisé une horloge (ou clock)

Voici le bloc associé à cette fonction:



### Envoie des blocs de caractères correspondant aux voies de la bouée:

Pour pouvoir faire le lien entre l’application et la bouée, nous envoyons des trains de données grâce au protocole UDP.

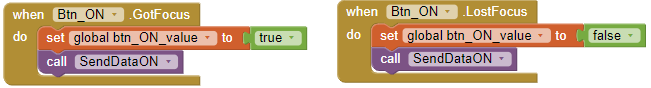
Pour récupérer les données concernant RC1 (le mode 4 voies), voici le bloc utilisé:



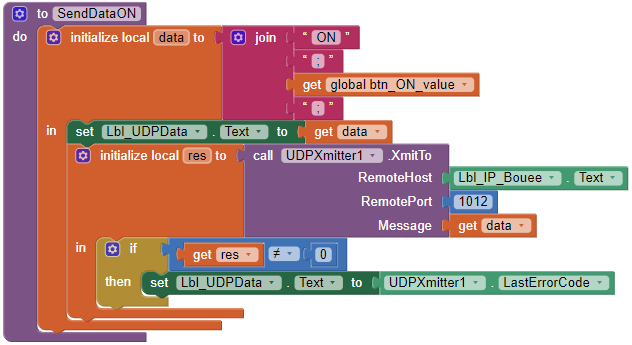
De même, nous avons construit le même type de bloc pour RC2, mais cette fois-ci avec non pas 4 voies dans le haut du bloc, mais 8 voies.

### Envoie des blocs de caractères correspondant au “Bouton ON” de la bouée:

Ces blocs correspondent à l’appelle de la fonction “SendDataON” lorsque le bouton est coché et décoché. Lorsque celui-ci est cohé, on attribue la valeur “true” à la variable globale “btn\_ON\_value”, et lorsque celui-ci est décocher, la valeur false.

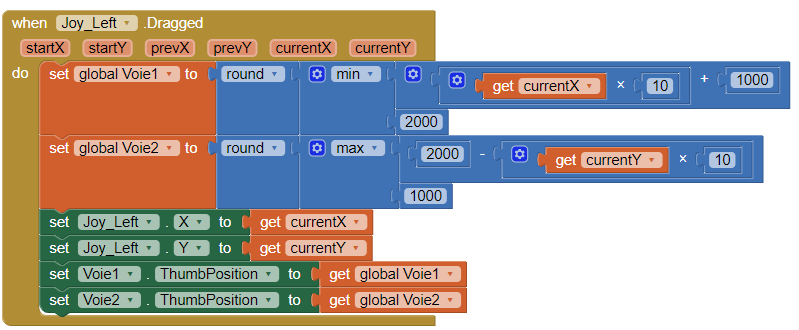


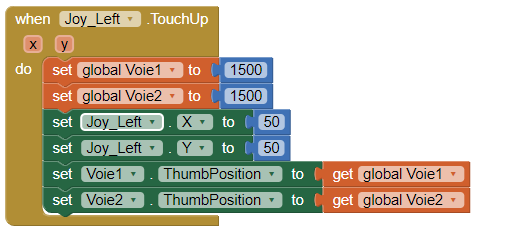
La fonction “SendDataON” correspond à l’envoi du train de caractère à la carte programmable:



### Récupération des données du joystick et mise à l’échelle des valeurs:

Les blocs ci-dessous correspondent aux blocs permettant de modéliser les joysticks de la télécommande. Vous ne trouverez que ceux pour le joystick de gauche. Cependant il existe exactement le même bloc pour le joystick droit.





### Modélisation des switchs de la télécommande:

Afin de modéliser les huits voies le la télécommande, nous avons rajouté des boutons “Action L” et “Action R” pour modéliser les switchs 2 positions, ainsi que des spinners “Mode L” et “Mode R” pour modéliser les switchs trois positions.





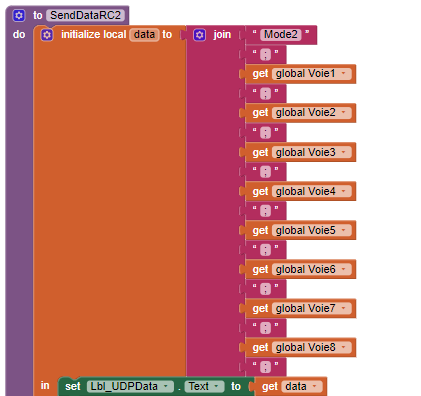
### Récupération des données de la voies 1 à 4:

Le rôle de l’application est de capter les ordres réalisés sur l'écran et de les transmettre à la bouée. Pour cela nous utilisons un protocole UDP, et envoyons les valeurs récupérées sur chaque vois sous forme d’un train de caractère. Voici le bloc permettant cet envoi.



1. Récupération des données de la voie 1 à 8:

Pour la récupération des 8 voies d’un seul coup, nous avons réalisé le même type de bloc que dans la partie ci-dessus, en ajoutant simplement les voies de 5 à 8 dans la première partie du bloc. Voici la partie concernée par le changement dans le bloc.



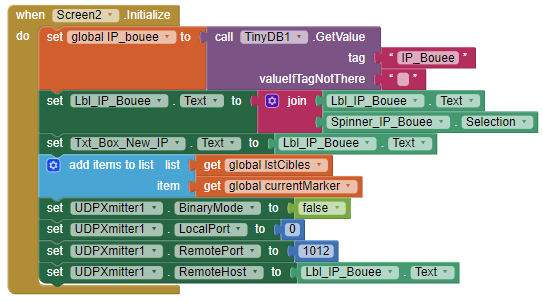
# Map

## Screen:



## Blocs:

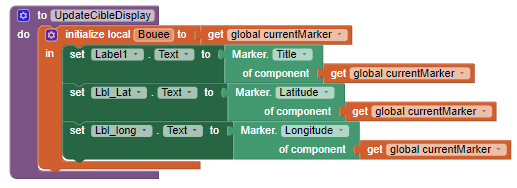
### Initialisation des paramètres



### Création d’un point GPS cible sur la carte:

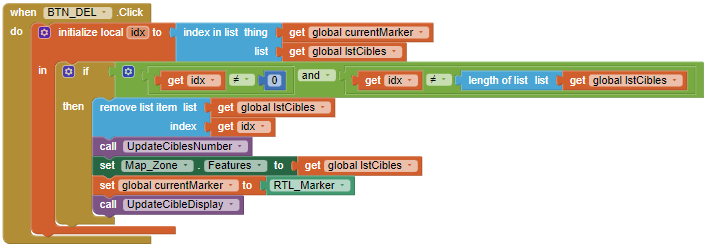


Ce bloc fait appel, à la fin, à la fonction “UpdateCibleDisplay”. La fonction est détaillée ci-dessous. Cette fonction permet de mettre à jour l’affichage des données concernant le numéro de cible, et les coordonnées GPS du marquer créé.



### Suppression d’un marqueur cible:

Ce bloc permet que lorsque l’on appuis sur le bouton “DEL”, le marqueur sélectionné à ce moment-là est supprimé, et les données associées également.



Ce bloc fait appelle à deux fonctions:

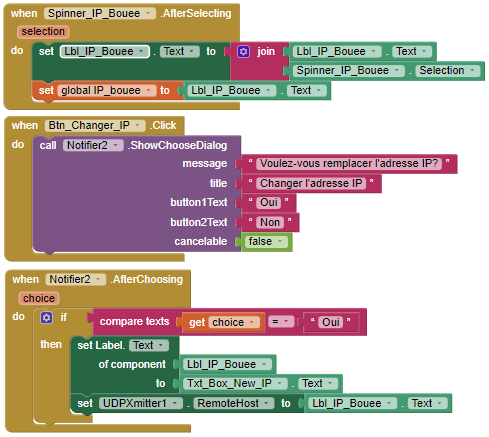
* UpdateCibleDisplay (déjà présenté ci-dessus)
* UdateCiblesNumber

Cette deuxième fonction permet d’ajouter un numéro de bouée, et donc une nouvelle cible dans la liste des bouées:



### Gestion de l’adressage IP de la bouée:

Pour le changement d’adresse IP, nous avons procédé de la même manière que pour la télécommande, voici donc les blocs concernés:



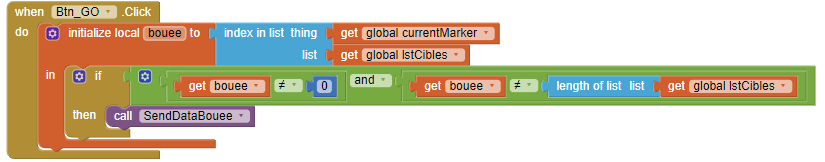
### Envoi du train de caractère pour récupérer le numéro de bouée et les coordonnées GPS de la bouée:

Pour la réalisation la de l’envoi de train de caractère permettant de récupérer pour une bouée donnée les coordonnées (longitude et latitude) visés, nous avons utilisé deux blocs différents.

Tout d’abord nous avons fabriqué la fonction permettant de créer le train de caractère que nous souhaitons envoyer. Nous avons choisi le mot “GPS” comme mot repère du bloc, qui nous permettra ainsi d’identifier la réception d’un bloc possédant les coordonnées visées de la bouée lors du traitement des blocs dans la partie Arduino.



Puis nous avons réalisé les conditions d’appel de la fonction:



Lorsque le Bouton Go est cliqué, alors les coordonnées de la cible sélectionnée sont envoyées au récepteur wifi arduino.