Découverte des objets connectés

D'après https://www.electro-info.ovh/premiers-pas-avec-arduino-d1-mini

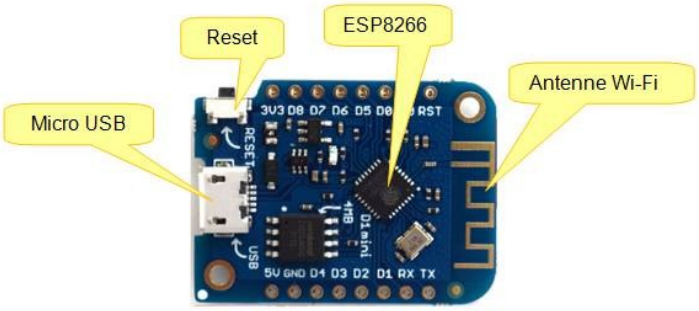
1. Présentation :

Ce TP permet de créer un bouton connecté WiFi à partir de deux cartes Arduino D1 mini (ESP8266). L'émetteur utilise un shield 1-bouton et le récepteur un shield relais. Dans la deuxième partie, on ajoute un shield de mesure de température afin de programmer le fonctionnement d'un thermostat.

La carte Arduino D1 Mini est basée sur le module ESP8266. Elle permet de réaliser des objets connectés grâce à sa connectivité Wi-Fi.

Comme pour la carte Arduino UNO il existe des shields destinés à la carte D1 Mini (ci-contre shields relais et bouton poussoir).

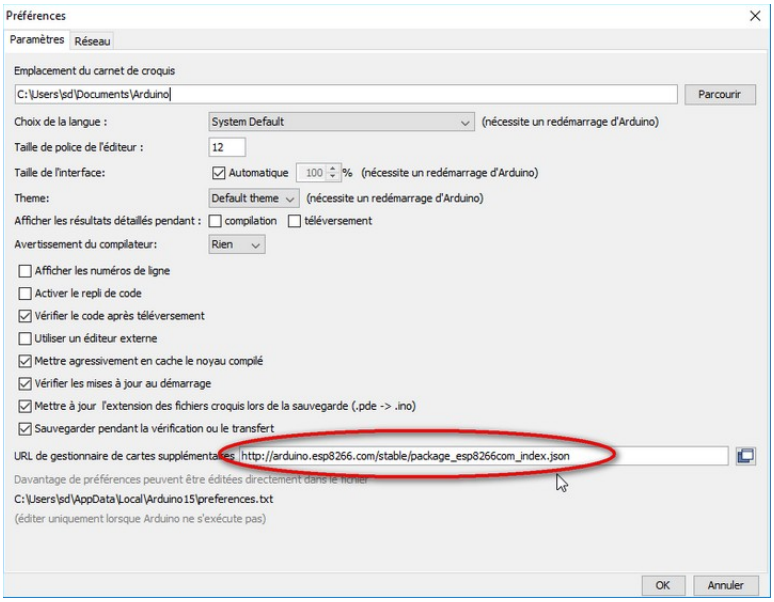
La "carte mère" est composée de deux parties principales : le microcontrôleur et le module de communication USB qui permet de programmer le microcontrôleur depuis un PC (pas besoin de programmateur). Le port micro USB permet également d'alimenter la carte (à condition que la consommation ne dépasse pas ce que peut fournir un port USB).



2. Configuration de l'EDI

Ouvrir l'EDI Arduino. Aller dans "Fichier => Préférences" :

Dans l'onglet Réseau, paramétrer tout d'abord le proxy du lycée :
172.16.0.254
Port 3128

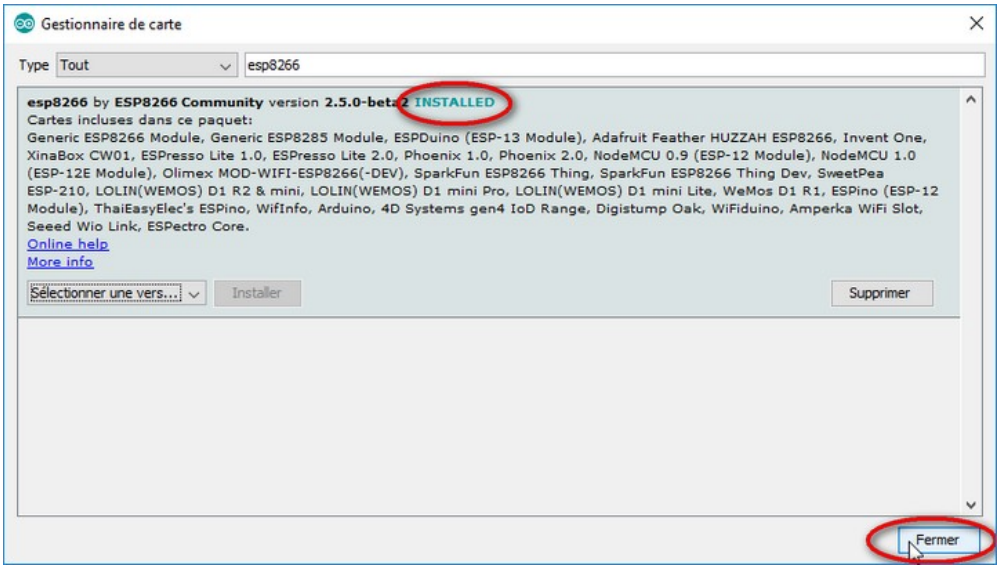


Dans "URL de gestionnaire de cartes supplémentaires", entrer :

http://arduino.esp8266.com/stable/package_esp8266com_index.json

Cliquer sur "OK".

Aller ensuite dans "Outils => Type de carte => Gestionnaire de carte" et installer "esp8266" :



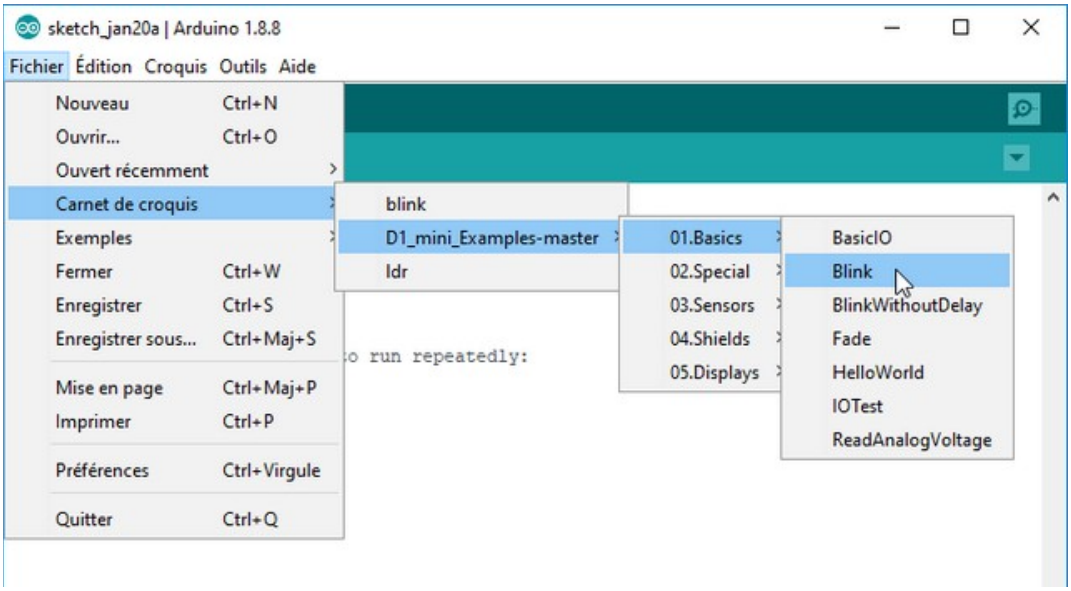
Télécharger les exemples de la D1 mini :
https://github.com/wemos/D1_mini_Examples/archive/master.zip
Décompresser l'archive dans le dossier : "documents/Arduino

Redémarrer l'EDI Arduino.

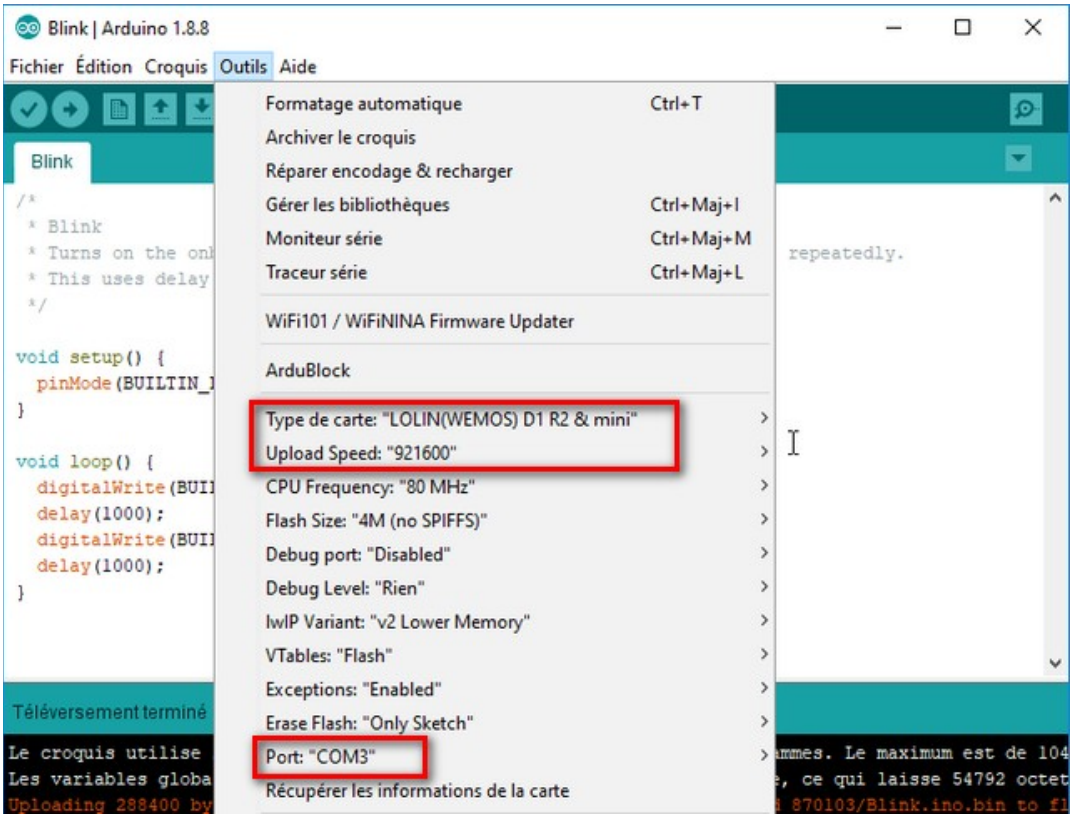
Connecter la carte D1 Mini à l'ordinateur avec un câble micro USB. Vérifier la bonne installation des drivers et repérer le numéro de port COM associé à la carte D1 mini (si besoin dans gestionnaire de périphériques).

3. Premier programme : le traditionnel Blink

Ouvrir le programme d'exemple "Blink" :



Configurer le type de carte, la vitesse de communication et le numéro de port COM comme indiqué ci-dessous :



Téléverser le programme, vérifier le bon fonctionnement.

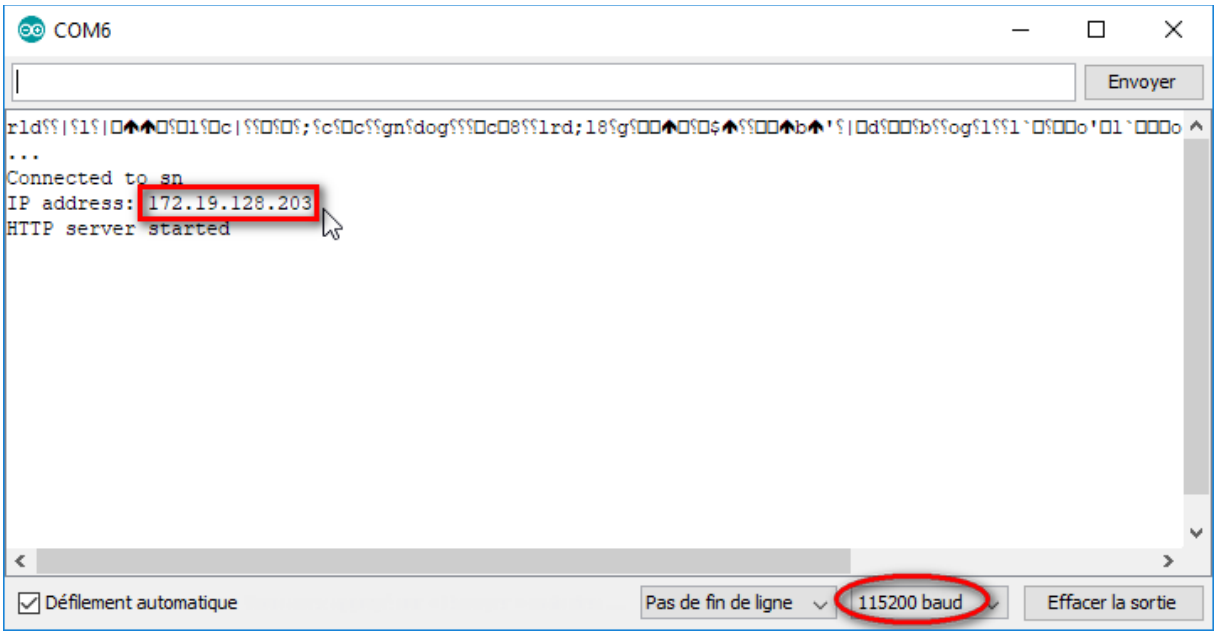
4. Deuxième programme : serveur WEB

Ouvrir le programme fourni : *"ServerWebRelais.ino"*.

Le relais peut être remplacé dans un premier temps par une LED.

Modifier, si nécessaire, les lignes 6 et 7 (SSID et clé de votre point d'accès WiFi).

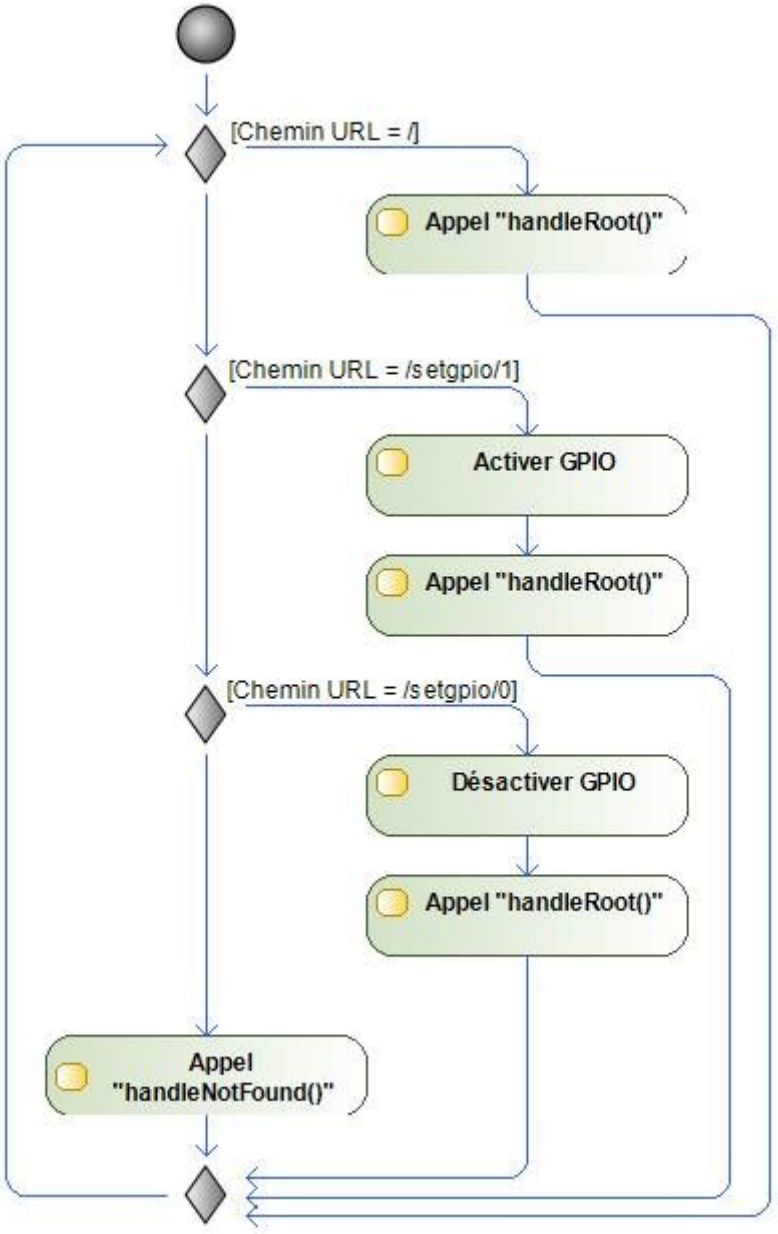
Le téléverser et ouvrir le moniteur série (CTRL + SHIFT + M), leconfigurer à 115 200 bauds. Appuyer sur le reset de la D1 Mini et relever l'adresse IP obtenue par la carte :



Taper cette adresse IP dans la barre d'adresse d'un navigateur et tester le bon fonctionnement :

Analyser le fonctionnement du programme à l'aide du diagramme d'activité ci-dessous :

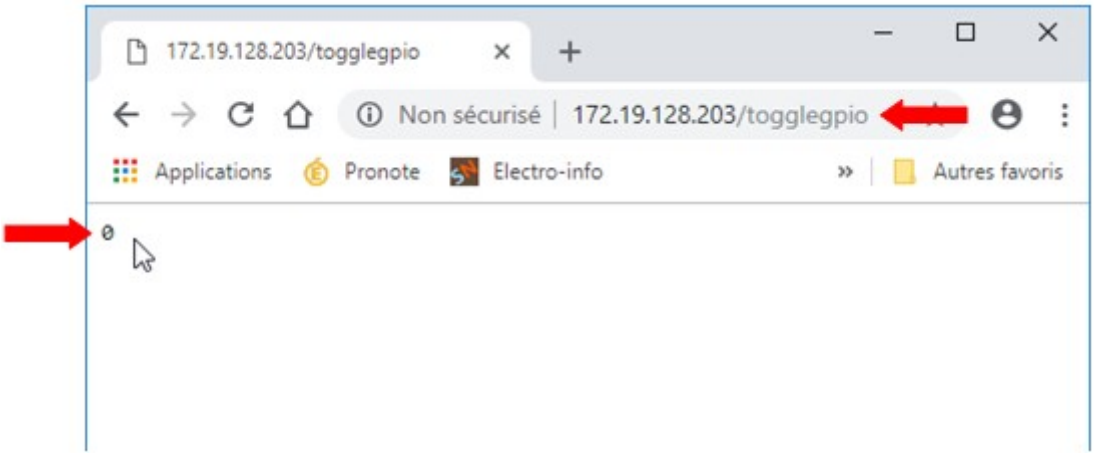
Diagramme d'activité :



5. Modification du programme

Créer une nouvelle fonction qui inversera l'état du relais et renverra une page Web en mode texte (utiliser le paramètre "text/plain" à la place de "text/html") contenant seulement "0" ou "1" selon l'état du relais. Cette fonction sera appelée lorsque le chemin de l'URL sera : "/togglegpio".

Tester le bon fonctionnement dans un navigateur :



6. Configuration de la deuxième carte :

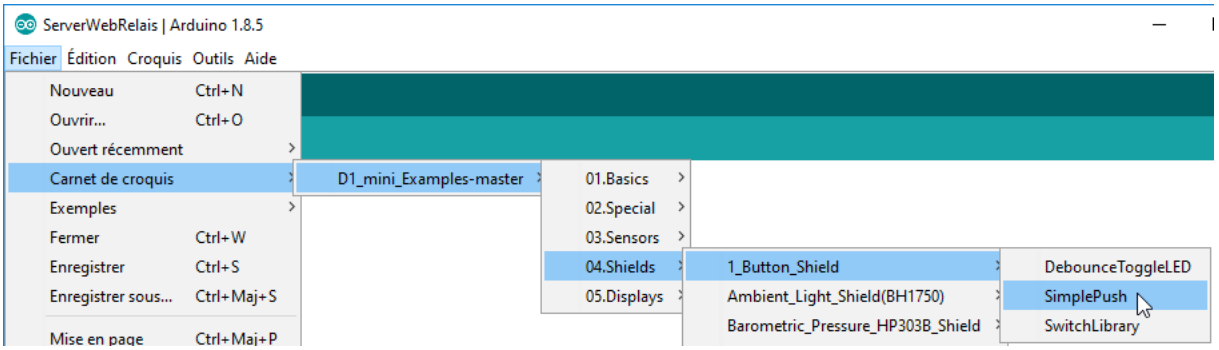
Nous allons utiliser une deuxième carte équipée d'un shield "1 bouton". Le but final étant que l'appui sur le bouton permette de faire commuter le relais d'un état à l'autre.

Le bouton est connecté sur l'entrée D3 de la mini D1 et est actif au niveau bas.

Si le shield n'est pas disponible, utiliser un bouton poussoir classique branché en filaire.

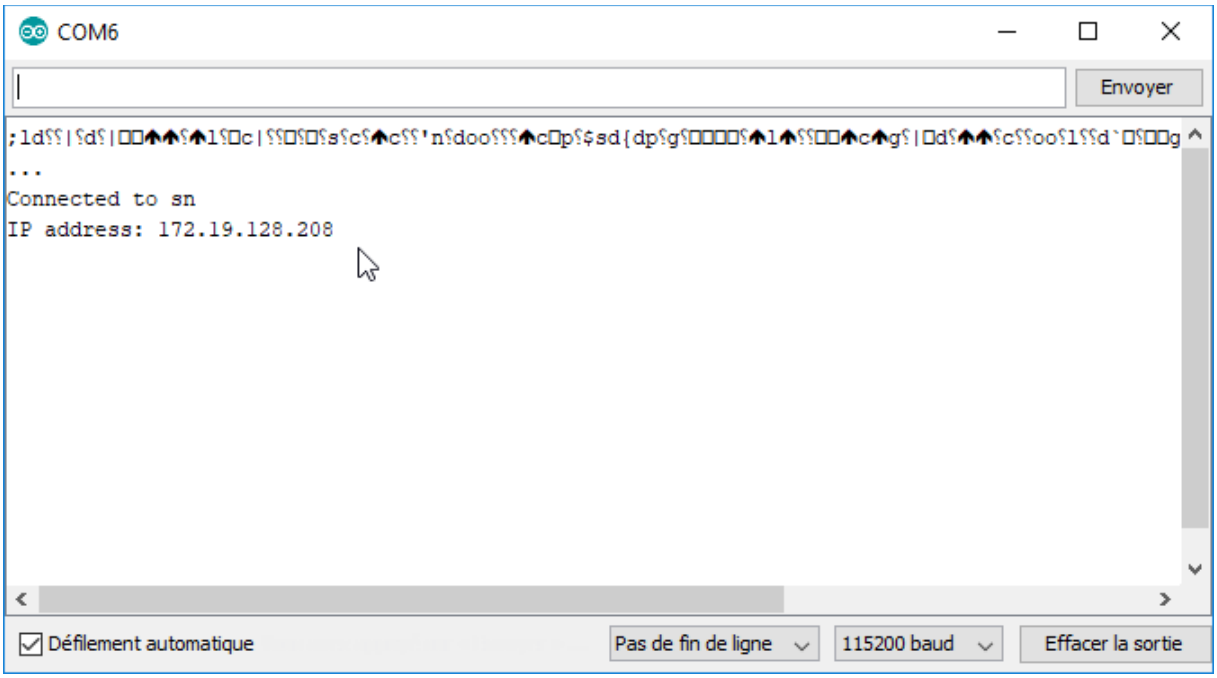
Ouvrir l'exemple "SimplePush" :

Tester.



Modifier l'exemple de telle sorte que la carte équipée du shield bouton se connecte au même point d'accès Wi-Fi que celle équipée du shield relais.

Tester (vérifier le bon fonctionnement avec la console) :



A partir de l'exemple d'envoi de trame http donné ci-dessous, modifier le programme de telle sorte que le relais de la première carte commute à chaque appui sur le bouton.

Exemple d'envoi d'une trame HTTP GET :

```
WiFiClient client ;
        // Etablissement de la connexion au serveur
if (client.connect("172.19.128.203", 80)) {
        // Envoi la requete au serveur
        client.print(String("GET /togglegpio HTTP/1.1\r\n") +
        "Host: 172.19.128.203\r\n" +
        "Connection: close\r\n\r\n");
}
```

Pour plus d'explications : https://projetsdiy.fr/esp8266-client-web-exemples-communication-tcp-ip-esp8266wifi-esp8266httpclient/

Tester.

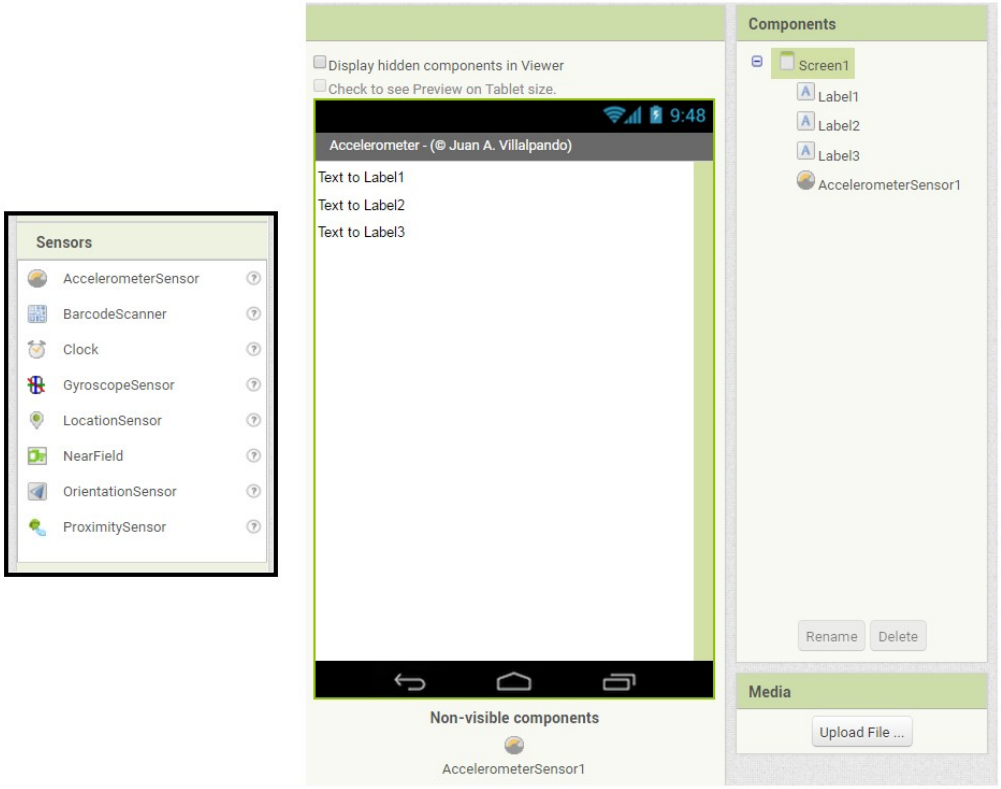
Commande BT – BTS SN

Lycée Blaise Pascal Longuenesse

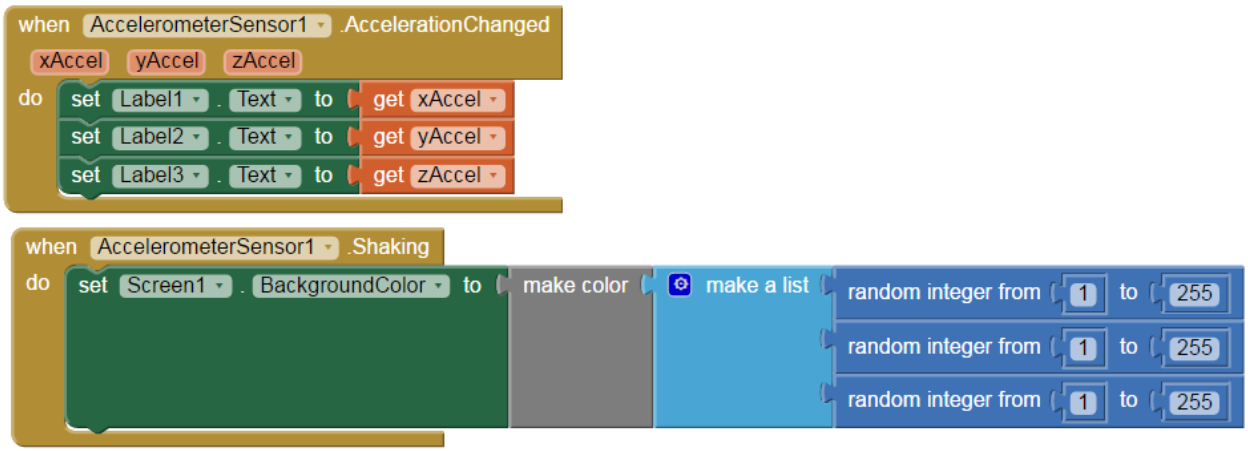
Pilotage via capteurs d'un smartphone

Utilisation de l'accéléromètre :

Saisir et tester le programme suivant. Visionner la vidéo sur les capteurs MEMS.



Blocks



- Additive color mixing. By adding Red, Green and Blue, obtain any other color
- Each color RVA, has 255 different shades, from 1 to 255. (this example 1-255)

- In total you can get 255 x 255 x 255 = 16,581,375 colors

Modification du programme LED :

Utiliser le programme 1 du TP précédent (allumage d'une LED via BT) pour allumer la LED lorsque le téléphone est penché vers l'avant et l'éteindre lorsqu'il est penché vers l'arrière.

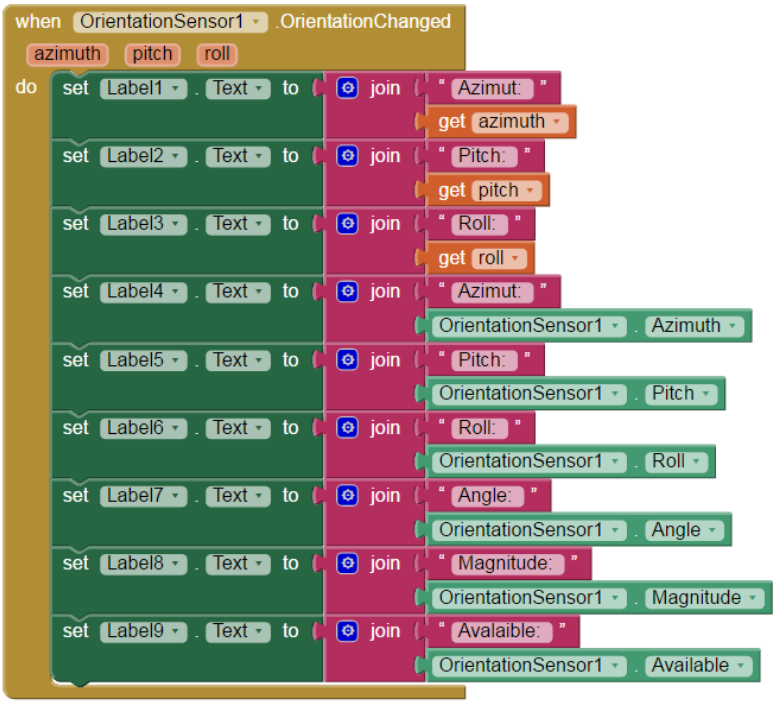
Modification du programme curseur :

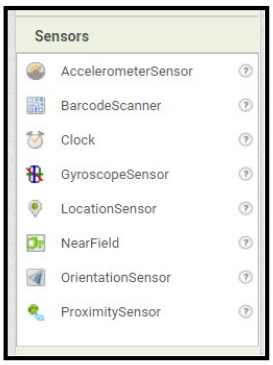
Utiliser le programme 2 du TP précédent (gérer la luminosité d'une LED via curseur) pour faire tourner le servomoteur sur la gauche lorsque le téléphone se déplace vers la gauche

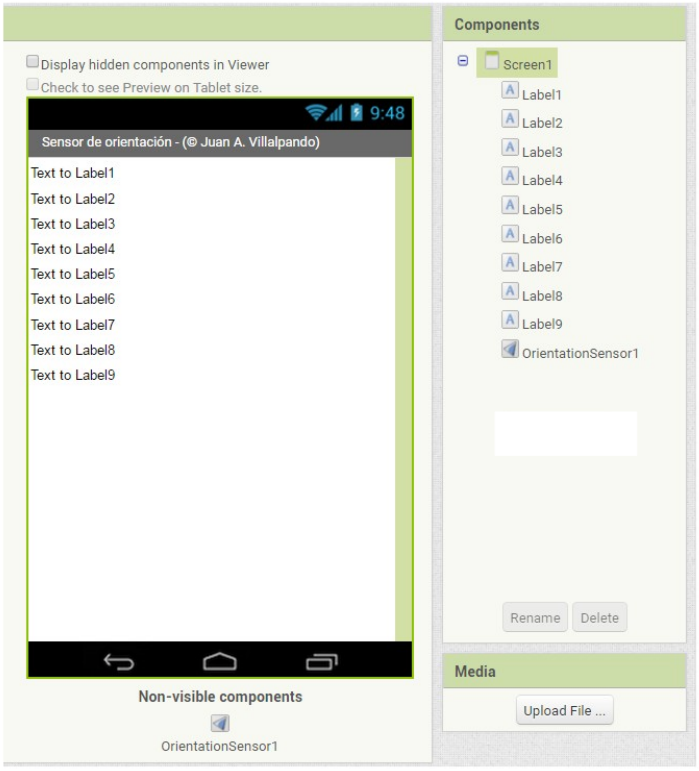
...et sur la droite lorsque le téléphone se déplace vers la droite.

Utilisation du capteur d'orientation :

Saisir et tester le programme suivant.





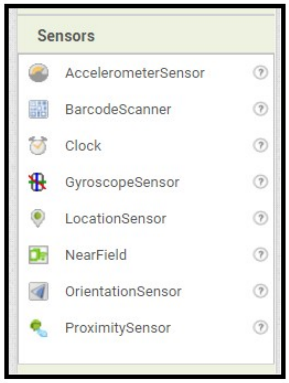


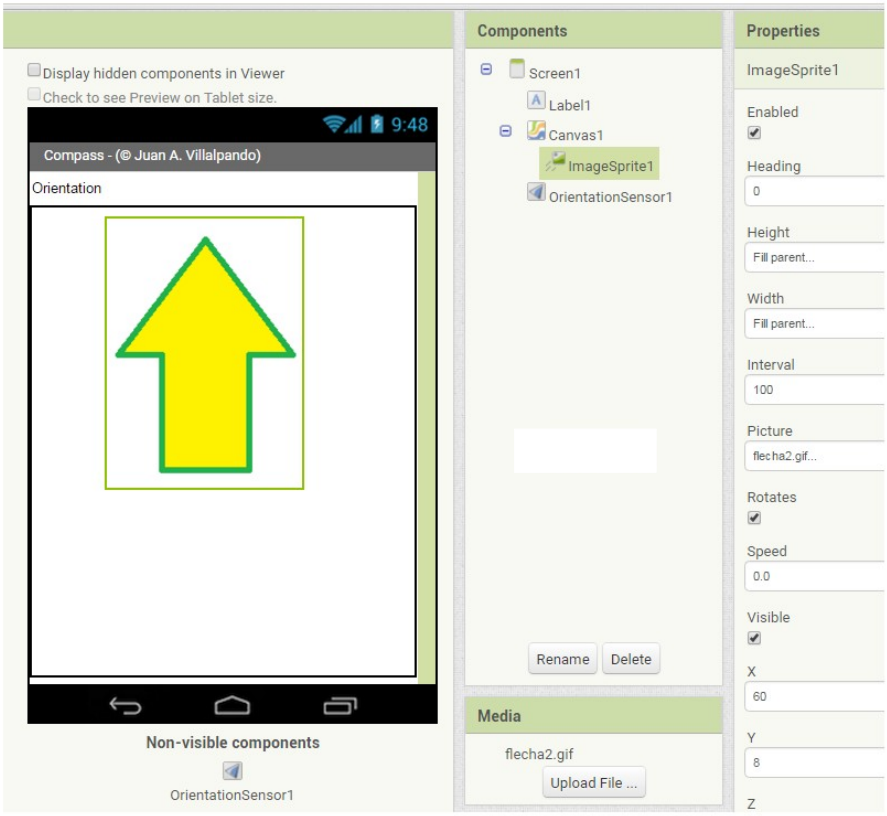
Utilisation du Canvas :

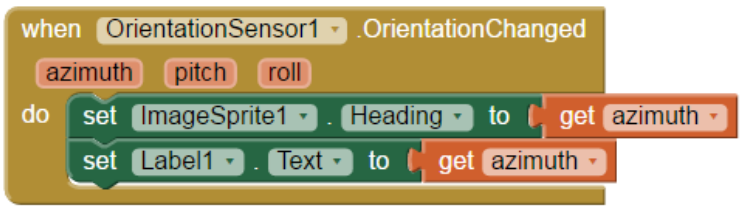
Saisir et tester le programme suivant.

- **Canvas** is an area where you can draw, move cartoon-sprites, ball,...
- **ImageSprote** is a drawing that can be moved around the canvas.

- We put a **Canvas.**
- Within the Canvas a **ImageSprite.**
- **We upload** this image arrow: (flecha2.gif)
- In the ImageSprite establish the **Property Photo** with flecha2.gif







When you change the orientation of the mobile, the top (Heading) of **ImageSprite** is drawn on the angle **azimuth.**
In this way the top of the **ImageSprite** always marks the North. That is, it acts as a compass.

You can also put a picture of a compass. **Brujula.gif**