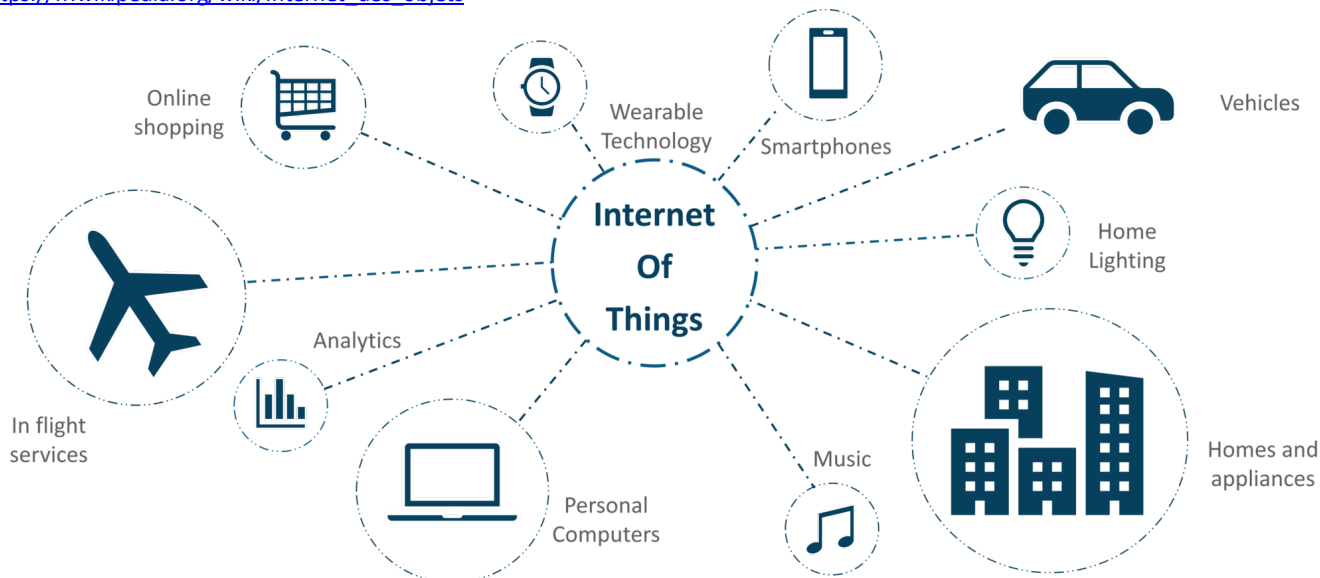
	Innovation et Développement Durable		1 ^{ère} STI2D
		Tutoriel : Internet Of Things (IoT)		
Séquence 1 : Les produits durables			Tutoriel	IT

1. Mise en situation

1.1. Introduction (d'après Wikipédia)

https://fr.wikipedia.org/wiki/Internet_des_objets



L'**Internet des objets**, ou IdO (en anglais Internet of Things, ou **IoT**) est l'interconnexion entre Internet et des objets, des lieux et des environnements physiques. L'appellation désigne un nombre croissant d'objets connectés à Internet permettant ainsi une communication entre nos biens dits physiques et leurs existences numériques. Ces formes de connexions permettent de rassembler de nouvelles masses de données sur le réseau et donc, de nouvelles connaissances et formes de savoirs.


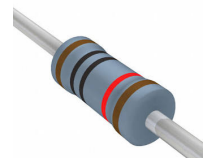
Considéré comme la troisième évolution de l'Internet, baptisé Web 3.0 (parfois perçu comme la généralisation du Web des objets mais aussi comme celle du Web sémantique) qui fait suite à l'ère du Web social, l'Internet des objets revêt un caractère universel pour désigner des objets connectés aux usages variés, dans le domaine de la e-santé, de la domotique ou du quantified self.


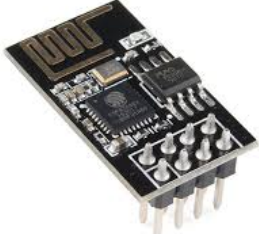


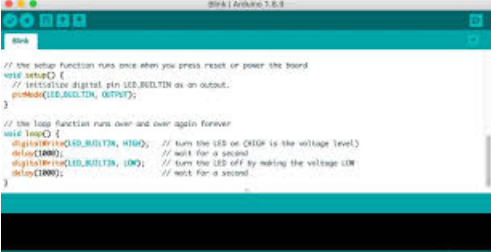

1.2. Objectif du tutoriel

On souhaite échanger des données depuis le laboratoire d'IT vers l'extérieur :

- **Objectif 1** : Afficher le niveau de luminosité du laboratoire sur un smartphone connecté à Internet
- **Objectif 2** : Commander l'allumage d'une LED depuis un smartphone connecté à Internet

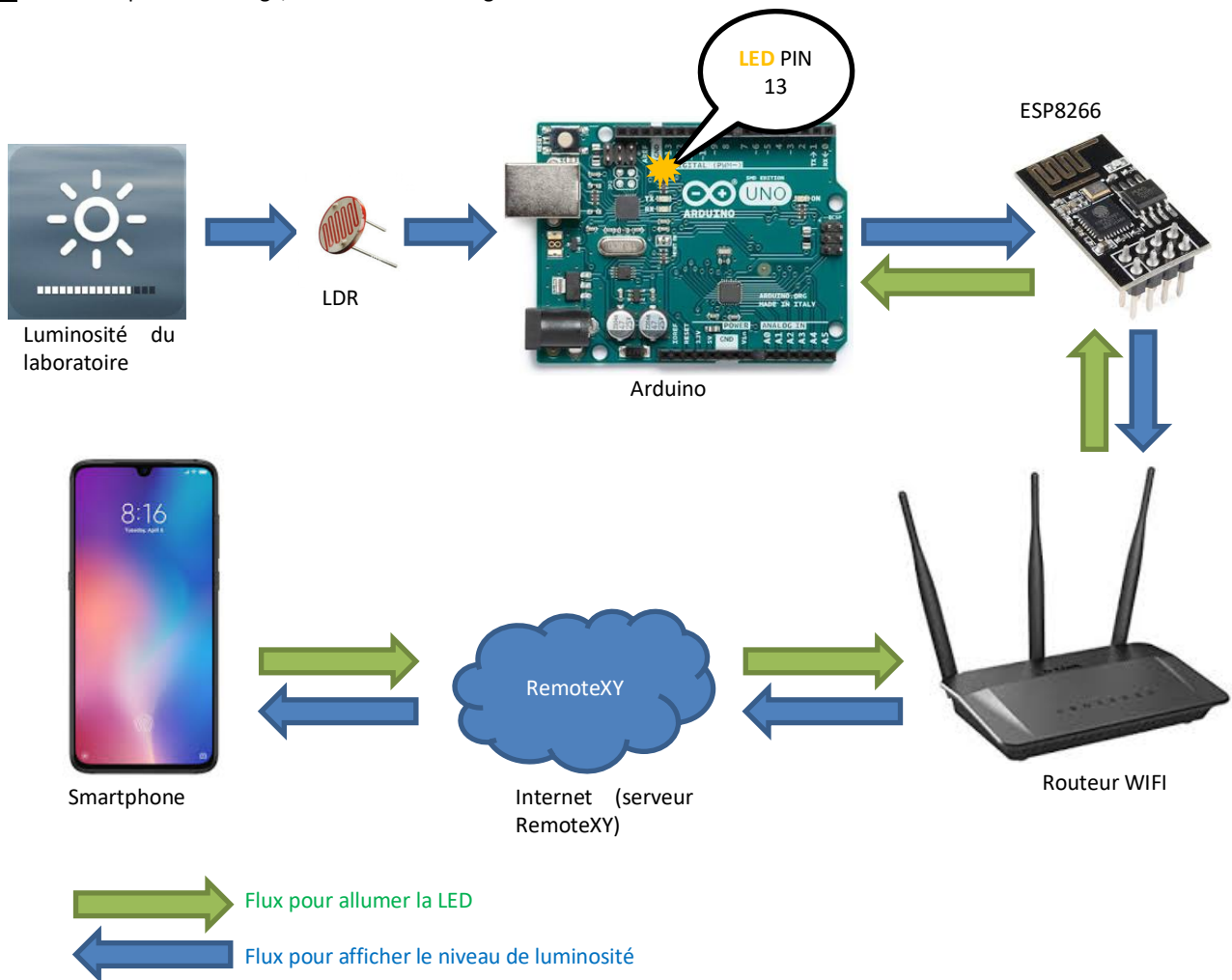
1.3. Matériel et logiciels utilisés

Désignation	Illustration	Fonction
LDR (Light Dependant Resistor)		Convertir un niveau de luminosité en information analogique (Résistance)
Résistance 10 kΩ		Réaliser un pont diviseur de tension avec la LDR afin de convertir le niveau de résistance en tension

<p>Carte Arduino UNO</p>		<p>Exécuter le programme afin réaliser les interactions entre les différents équipement</p>
<p>Module ESP8266</p>		<p>Connecter l'Arduino à Internet en WIFI</p>
<p>Routeur WIFI</p>		<p>Mettre à disposition Internet en WIFI</p>
<p>Smartphone Android</p>		<p>Interface Utilisateur</p>
<p>Arduino IDE</p>	 <pre> // the setup function runs once when you press reset or power the board void setup() { // initialize digital pin LED_BUILTIN as an output. pinMode(LED_BUILTIN, OUTPUT); } // the loop function runs over and over again forever void loop() { digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW delay(1000); // wait for a second } </pre>	<p>Programmation de l'Arduino</p>
<p>RemoteXY</p>		<p>Serveur Cloud IoT - Permet le stockage et la consultation des données sur Internet</p>

1.4. Synoptique de l'installation

Note : Afin de simplifier le câblage, on utilise la LED intégrée de la carte Arduino



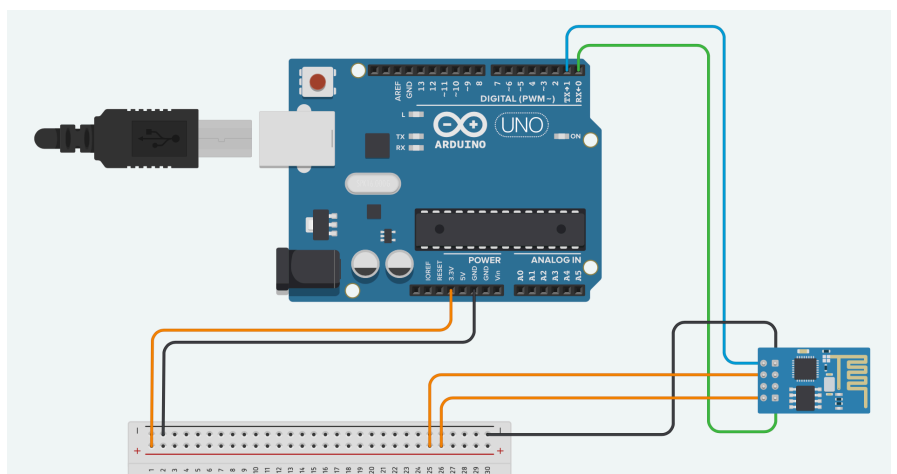
2. Préparation du module ESP8266 (A faire par le professeur)

Par défaut, le module ESP8266 communique avec la carte Arduino via UART avec une vitesse de **115200 Bauds**.

Cette vitesse est **trop rapide** pour la communication série virtuelle que nous allons utiliser. **Il faut alors abaisser la vitesse du module ESP8266 à 19200 Bauds.**

Procédure :

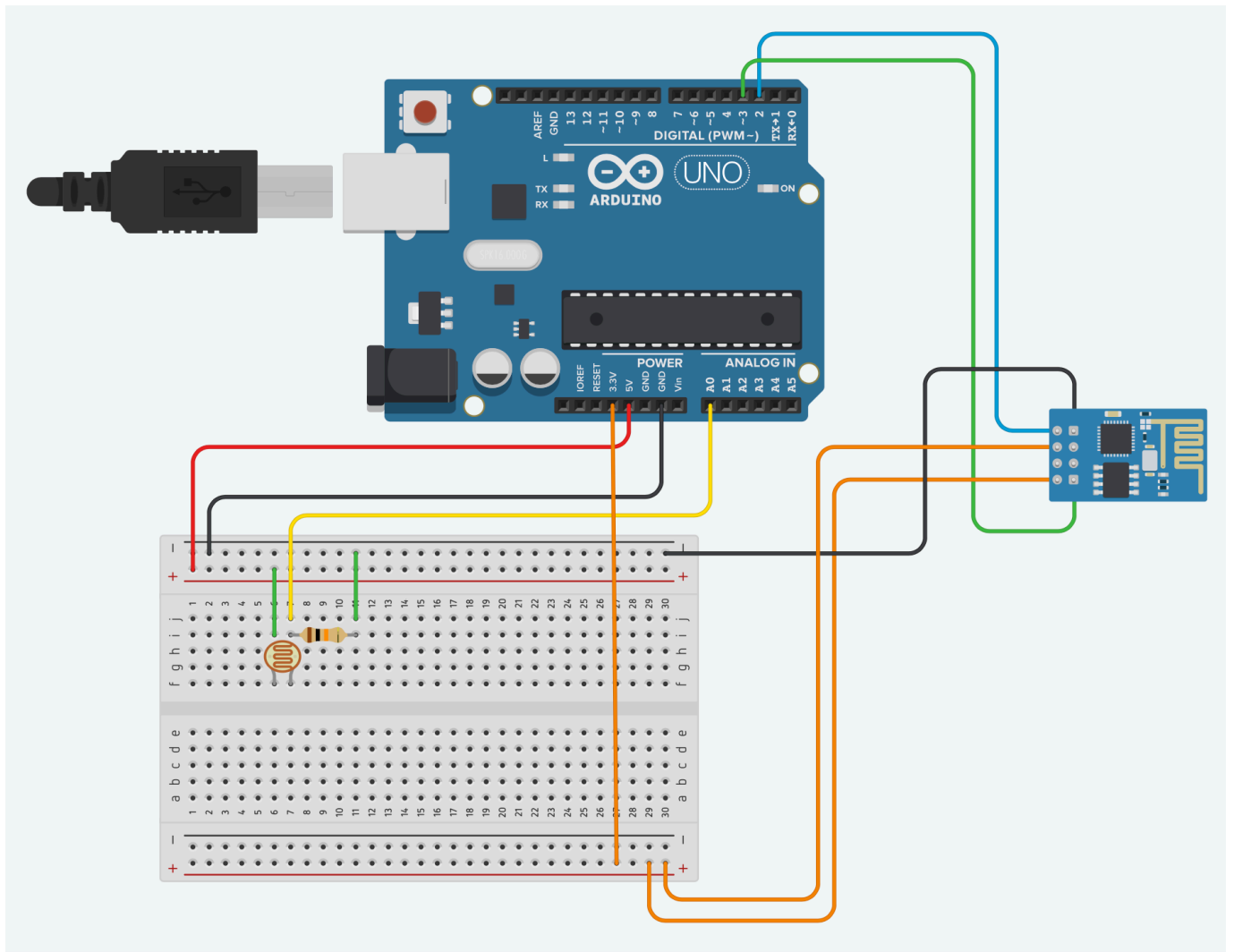
- Débrancher le module ESP8266 de l'Arduino
- Ouvrir l'IDE Arduino, charger le programme « BareMinimum.ino » (Fichier/Exemples/01.Basics)
- Téléverser le programme
- Débrancher l'Arduino de l'USB
- Connecter le module ESP8266
- Rebrancher l'Arduino
- Ouvrir le moniteur série (Baud = 115200, « Les 2, NL et CR »)
- Entrer la commande « AT » > la console doit répondre « OK »
- Entrer la commande « AT+UART_DEF=19200,8,1,0,0 », la console doit répondre « OK ».



3. Câblage Arduino

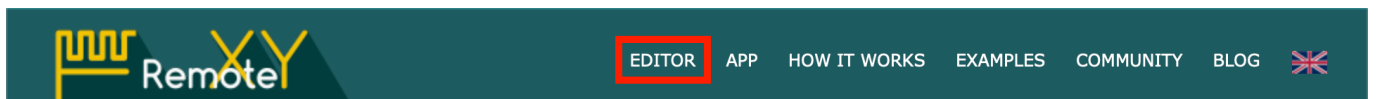
Note : Le câblage doit se faire HORS TENSION (Arduino débranché).

FAIRE VERIFIER VOTRE MONTAGE AVANT LA MISE SOUS TENSION !!



4. RemoteXY

- Aller sur le site <http://remotexy.com/en/> et créer un compte.
- Cliquer sur « Editor » afin de lancer l'interface de conception :



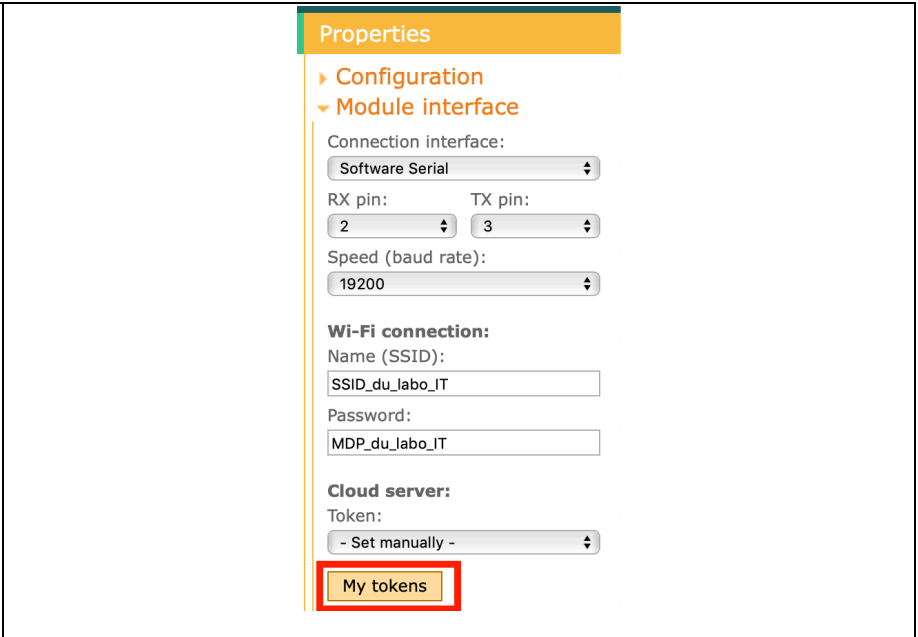
- Glisser / Déplacer les éléments de l'interface (au besoin, redimensionner) :

- Dans le volet « Properties », cliquer sur « Configuration », puis sur « Arduino UNO » :

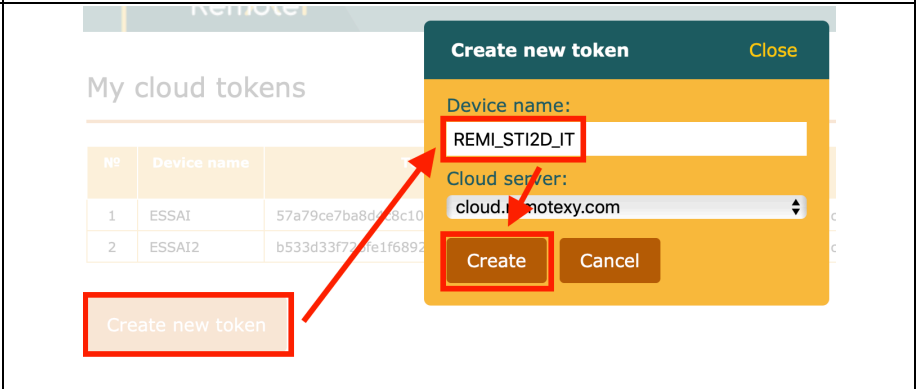
- Régler les paramètres afin d'obtenir la configuration suivante, puis cliquer sur « Apply » :

- Dans le volet « Properties », cliquer sur « Module Interface », et régler la vitesse ainsi que les paramètres de connexion au WIFI du laboratoire d'IT :

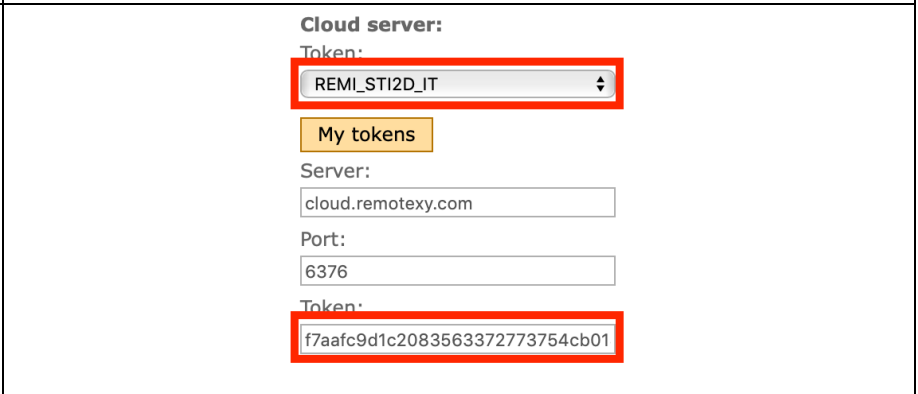
- Il faut à présent configurer un serveur internet RemoteXY afin de s'y connecter via le module WIFI. Cliquer sur « My Tokens » :



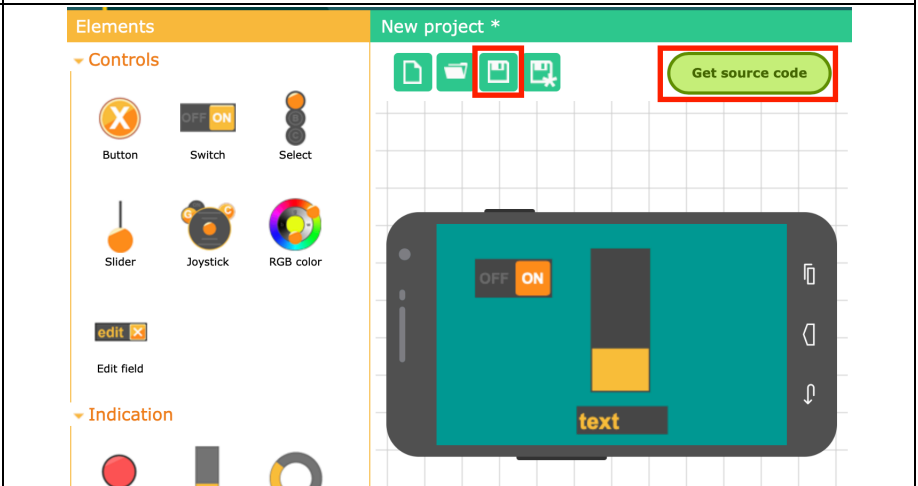
- Cliquer sur « Create New Token », saisir un nom et cliquer sur « Create » :

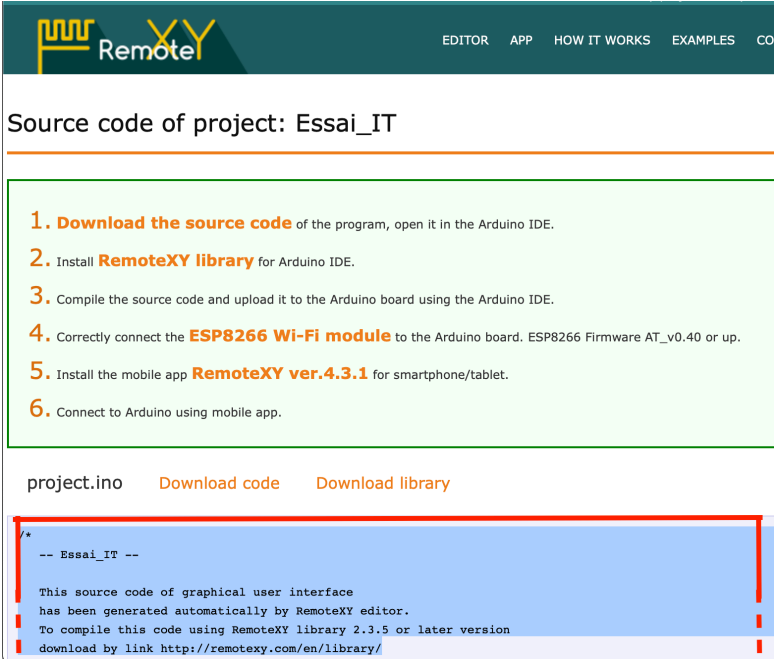


- Sélectionner le serveur nouvellement créé dans la liste, et copier/coller la clé afin de se l'envoyer par e-mail (sur le smartphone de contrôle) :



- Sauvegarder l'interface, puis cliquer sur « Get source code » :



<ul style="list-style-type: none"> Copier le code généré et le coller dans un nouveau Sketch de l'IDE Arduino : 	 <p>The screenshot shows the RemoteXY web editor interface. At the top, there's a navigation bar with 'EDITOR', 'APP', 'HOW IT WORKS', 'EXAMPLES', and 'CO'. Below that, the title 'Source code of project: Essai_IT' is displayed. A green box contains a numbered list of 6 steps: 1. Download the source code, 2. Install RemoteXY library, 3. Compile and upload, 4. Connect ESP8266 module, 5. Install mobile app, 6. Connect to Arduino. Below the list are links for 'project.ino', 'Download code', and 'Download library'. A code block is shown with a red border, containing a comment and instructions about the GUI code generation.</p>
--	---

5. Programmation Arduino

Le code généré est presque utilisable en l'état, il faut cependant rajouter quelques lignes dans le **loop** afin de faire fonctionner l'afficheur à barre et le texte :

```
void loop()
{
  RemoteXY_Handler ();

  digitalWrite(PIN_SWITCH_1, (RemoteXY.switch_1==0)?LOW:HIGH);

  int lum = analogRead(A0) / 10.23; // Lire la pin A0 pour déterminer la luminosité
  // (on divise par 10.23 pour avoir une échelle de 0 à 100, nécessaire pour le graphe à barre)

  RemoteXY.level_1 = lum; // Afficher la luminosité sur le graphe à barre

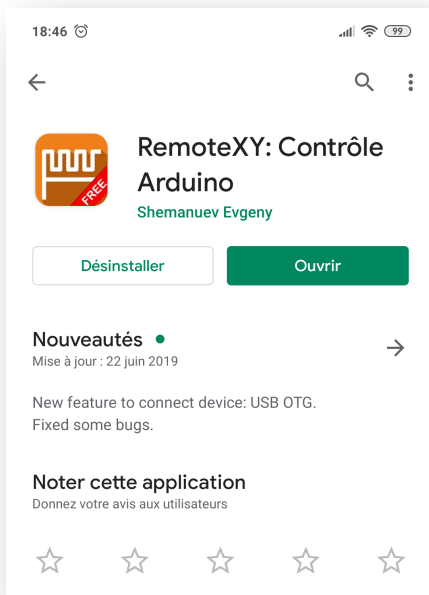
  dtostrf(lum, 4, 0, RemoteXY.text_1); // Afficher la luminosité sur le texte
  //(fonction dtostrf permet de convertir un nombre en chaîne de caractères)

  // TODO you loop code
  // use the RemoteXY structure for data transfer
}
```

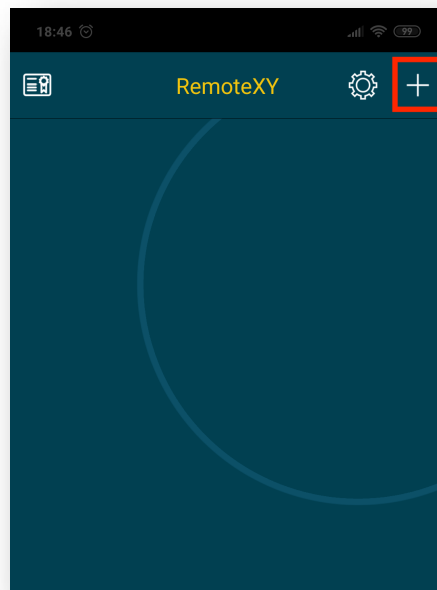
Téléverser le programme sur la carte Arduino.

6. Accès à l'interface (UI) sur le smartphone Android

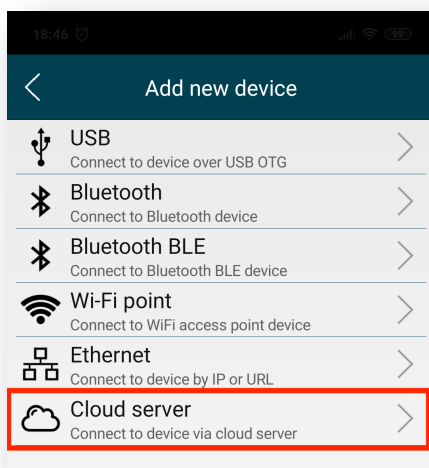
1. Sur le Playstore, installer l'application « RemoteXY » :



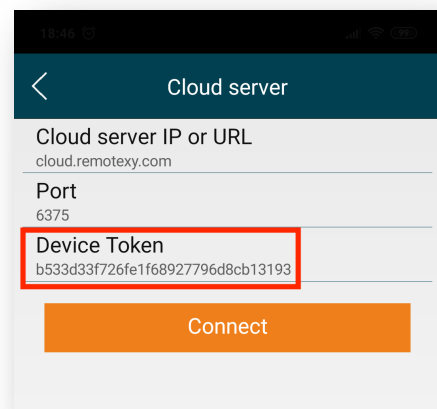
2. Une fois l'application démarrée, ajouter un nouveau serveur :



3. Choisir un serveur « Cloud » :



4. Entrer la clé précédemment copiée au paragraphe 4, puis cliquer sur « Connect » :



5. Si tout va bien, l'interface apparaît, on peut alors interagir avec le montage du laboratoire d'IT **depuis n'importe quel lieu** (à condition de disposer d'un accès internet sur le Smartphone ;-)

