

La station météo

Anna MANI



Anna MANI – 1918-2001

Scientifique météorologue, elle a réalisé des travaux de recherche sur la couche d'ozone, inventé une sonde de mesure de l'ozone. Elle a travaillé également sur l'énergie solaire thermique et les parcs éoliens en Inde. Sa carrière scientifique débute dans le domaine de la physique, sur les propriétés optiques des diamants et des rubis.

Nous lui dédions cette station météorologique pédagogique, puisse-t-elle inspirer les filles comme les garçons dans le goût pour les sciences et les technologies.

Photo : Wikipédia, source inconnue, usage pédagogique - https://fr.wikipedia.org/wiki/Anna_Mani

1- Présentation : De quoi s'agit-il ?

Il s'agit d'une station météo connectée à votre réseau wifi (et même à internet !).

On peut interagir avec cette station météo grâce au logiciel SNAP (sorte de SCRATCH, pour les personnes qui connaissent).

La station météo est composée d'un ensemble de capteurs :

- **Baromètre** : Capteur de pression (pression et température).
- **Hygromètre** : Capteur d'humidité (humidité et température).
- **Thermomètre** : Capteur de température.
- **Photo-résistance** : Capteur de luminosité.

Plus qu'une station météo, c'est aussi un **objet connecté** qui interagir avec **internet** !

Elle s'appuie sur la carte électronique « **Wemos D1 mini** » qui est une sorte de carte « **Arduino** » équipé de wifi (elle peut soit se connecter à un réseau wifi, soit être elle-même un réseau wifi et faire office de serveur – ces deux modes peuvent fonctionner en même temps).

Pourquoi choisir le Wemos D1 mini ?

Essentiellement pour deux raisons :

1. C'est une carte électronique qui est compatible Arduino tout en ayant le wifi.
2. Son prix est très bas : vendue sur le site chinois de commerce en ligne Aliexpress, elle ne coûte que 2 à 3 euros.

En étant compatible Arduino, nous pouvons nous appuyer sur nos connaissances, sur la documentation qui est abondante et disponible y compris en français et sur toute la communauté Arduino qui a déjà développé de nombreux projets techniques ou pédagogiques.

Le prix très faible en fait une carte qu'on peut donner aux participants aux activités, leurs permettant ainsi de poursuivre leurs expérimentations à la maison.

2- Comment cela fonctionne-t-il ?

Il faut charger le programme « station météo Anna MANI » sur le Wemos. De cette manière le Wemos est prêt à fonctionner et interagir avec le logiciel SNAP.

Il n'est donc pas nécessaire de programmer le Wemos avec le logiciel Arduino.

L'utilisation pédagogique du kit station météo se fait :

1. En réalisant les montages sur la plaque de prototypage.
2. En utilisant ces montages avec le logiciel SNAP.

Ce livret est conçu pour vous accompagner dans la prise en main du Wemos D1 mini et de la station Météo Anna MANI !

Avant toute chose, en voici la table des matières :

Table des matières

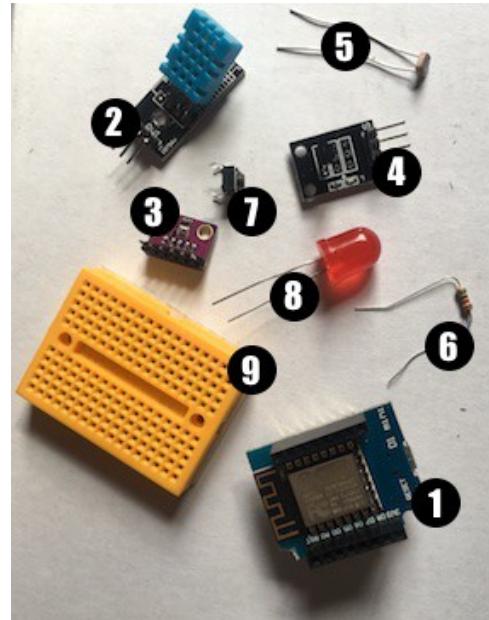
1-Présentation : De quoi s'agit-il ?.....	2
2-Comment cela fonctionne-t-il ?.....	2
3-Les composants de la « Station Météo Anna MANI» :	5
3.1-Liste des composants.....	5
3.2-Description des capteurs.....	5
4-Comment récupérer les données des capteurs ?.....	7
4.1-Première chose : Connectez votre station météo à votre réseau Wifi.....	7
4.2-Connaître l'adresse IP de sa Station Météo !.....	8
4.3-Premier essais : lire les données de l'Hygromètre.....	9
4.3.1-Montage des composants.....	9
4.3.2-Récupérer les données.....	9
4.4-La configuration de base du Wemos « Station météo Anna MANI».....	10
4.5-Utiliser SNAP! pour lire les données d'un capteur.....	11
5-Les fiches d'activité.....	13
5.1-Fonctionnement de base de la station météo Anna MANI.....	14
5.2-Un indicateur Jour/nuit.....	15
5.2.1-Le défi ingénieur : Comment réaliser un indicateur Jour/Nuit ?.....	15
5.2.2-Si vous voulez aller plus vite avec l'indicateur Jour/nuit.....	16
5.3-Un thermomètre.....	17
5.3.1-Le défi ingénieur : Comment réaliser un affichage de température à aiguille numérique ?.....	17
5.3.2-Pour aller plus vite avec le thermomètre.....	18
5.4-Un hygromètre.....	19
5.4.1-Le défi ingénieur : Comment réaliser un hygromètre qui calcule le point de rosée ?....	19
5.4.2-Pour aller plus vite avec l'hygromètre !.....	20
5.5-La station météo Anna MANI au complet !.....	21
5.5.1-Le défi ingénieur : Tout à fond !	21
5.6-Le bouton sans fil qui appelle la LED !.....	22
5.6.1-Le défi ingénieur : Comment allumer une LED et une ampoule SNAP!	22
5.6.2-Pour aller plus vite avec le bouton sans fil qui appelle la LED !.....	23
5.7-Inventer une activité.....	24
6-Annexes :	25
6.1-Le logiciel SNAP!.....	26
6.2-Utiliser la carte Wemos D1 mini.....	29
6.2.1-Une petite manip' pour utiliser le logiciel Arduino avec le Wemos D1 Mini :	29
6.3-En savoir plus sur le Wemos D1 mini.....	31
6.4-Charger le programme sur le Wemos avec le logiciel Arduino.....	32
6.4.1-Ajouter les bibliothèques nécessaires.....	32
6.4.2-Télécharger le programme sur le Wemos.....	33
6.5-Utiliser IFTTT.....	35
6.6-Le bouton qui twite avec IFTTT !.....	36
6.7-Ressources humaines et documentaires.....	41
6.8-L'aide mémoire de tous les blocs SNAP!.....	42
6.9-Thingspeak présentations : Un site web pour l'internet des objets.....	43
6.10-Comment ça marche ?.....	43

6.11-Comment créer et alimenter sa chaîne en données ?.....	45
6.11.1-Créer un compte.....	45
6.11.2-Créer sa chaîne.....	46
6.11.3-Comprendre votre chaîne.....	47
6.11.4-Configurer le programme Station Météo Anna MANI.....	47
6.11.5-Au sujet de l'onglet « API Keys ».....	48

3- Les composants de la « Station Météo Anna MANI » :

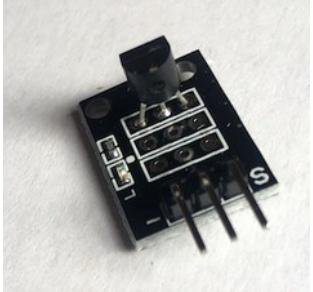
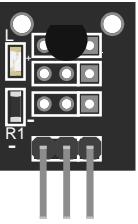
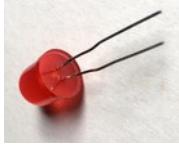
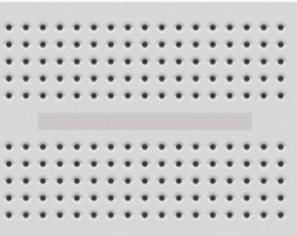
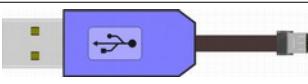
3.1- Liste des composants

1. Wemos D1 Mini
2. Capteur de température et humidité DHT11
3. Capteur de température et pression BMP180
4. Capteur de température DS18B20
5. Photorésistance
6. Résistance 1Kohms
7. Bouton poussoir
8. LED
9. Plaque de prototypage
10. Cable USB ↔ microUSB



3.2- Description des capteurs

nom	Photo	Pictogramme	description
Wemos D1 Mini			<p>Voir les détails sur http://www.wemos.cc</p> <p>nous avons choisi de lui souder ses longues pates avec broches pour ne pas perdre de possibilité de branchement lorsqu'il est installé sur la plaque de prototypage</p>
Capteur de température et humidité DHT11			<ul style="list-style-type: none"> - Alimentation : 3,3V - 5V - Consommation : 0.5 mA en nominal / 2.5 mA maximum - Etendue de mesure température : 0°C à 50°C ± 2°C - Etendue de mesure humidité : 20-90%RH ±5%RH
Capteur de température et pression BMP180			<ul style="list-style-type: none"> - Alimentation : 3,3V - 5V - Consommation : 2,8 uA - Etendue de mesure température : -40°C à 85°C ± 1°C - Etendue de mesure de pression : 300 – 1100 hPa ± 1 hPa - fonctionne en protocole I2C.

Capteur de température DS18B20			- Alimentation : 3,3V - 5V - Consommation : 2,8 uA - Etendue de mesure température : -55°C à 125°C ± 0,5°C fonctionne en protocole OneWire.
Photorésistance			résistance qui varie en fonction de la lumière.
Résistance 1K			Résistance de 1 Kohms (1000 ohms)
Bouton poussoir			Quand on appuie, le courant passe (il est dit « Normalement Ouvert – NO). Il en existe de plusieurs types : Normalement Fermé (NF), le contraire du premier
LED			Light Emitting Diode – Diode Électro-Luminescente en français soit DEL. Elle ne laisse passer le courant que dans un sens en émettant de la lumière
Plaque de prototypage			10X17 soient 170 trous. Les trous sont connectés ensemble en deux rangées parallèle de 17 lignes de 5 trou.
Cable USB ↔ microUSB			Permet soit d'alimenter le Wemos, soit de le connecter à un ordinateur

4- Comment récupérer les données des capteurs ?

4.1- Première chose : Connectez votre station météo à votre réseau Wifi.

Si vous avez déjà connecté votre station météo « Anna MANI » à votre réseau wifi, elle s'en souvient et se reconnecte automatiquement.

Sinon, voici la marche à suivre :

Recherchez le Point d'Accès wifi « **AutoConnectAP** » et connectez-vous (il faut attendre quelques minutes après l'allumage, le temps que toutes les fonctions s'activent).

Une fenêtre « **AutoConnectAP** » s'ouvre automatiquement (si ce n'est pas le cas ouvrez la fenêtre d'un navigateur et essayez de rejoindre une page web, le navigateur devrait vous rediriger automatiquement vers cette page).



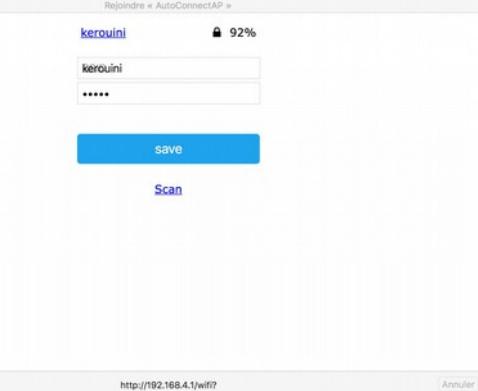
AutoConnectAP
WiFiManager
Configure WiFi
Configure WiFi (No Scan)
Info
Reset

Cliquez sur « **Configure WiFi** »

Cela déclenche un scan automatique des réseaux Wifi disponibles.



Vous cliquez sur le réseau de votre choix et vous entrez votre mot de passe.



Rejoindre « AutoConnectAP »
kerouini 92%
kerouini
.....
Save
Scan



Credentials Saved
Trying to connect ESP to network.
If it fails reconnect to AP to try again.

Dès que les mots de passe et nom du réseau sont enregistrés (« **Credentials saved** » en anglais).

Cela indique « **trying to connect ESP to network. If it fails reconnect to AP to try again.** »

Qui peut se traduire par : « *Essais de connexion de l'ESP(Wemos) au réseau. Si cela échoue, reconnectez-vous au Point d'Accès pour ré-essayer.* »

4.2- Connaitre l'adresse IP de sa Station Météo !

Ouvrez votre logiciel Arduino

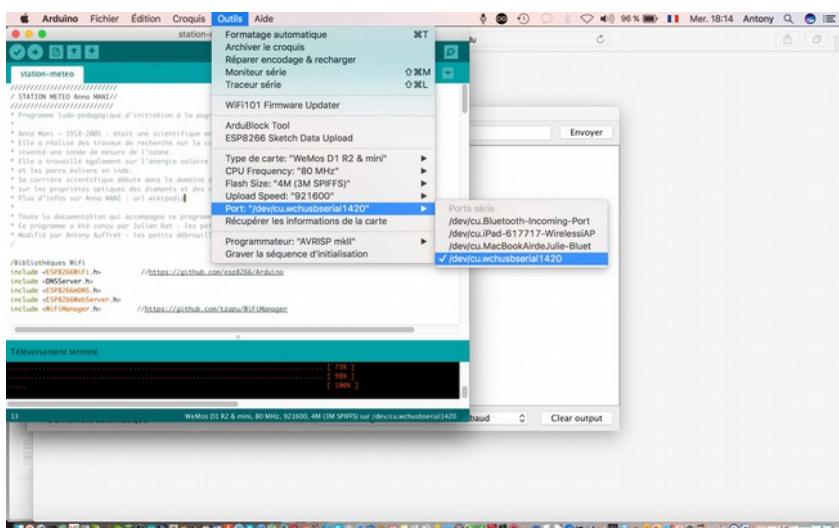
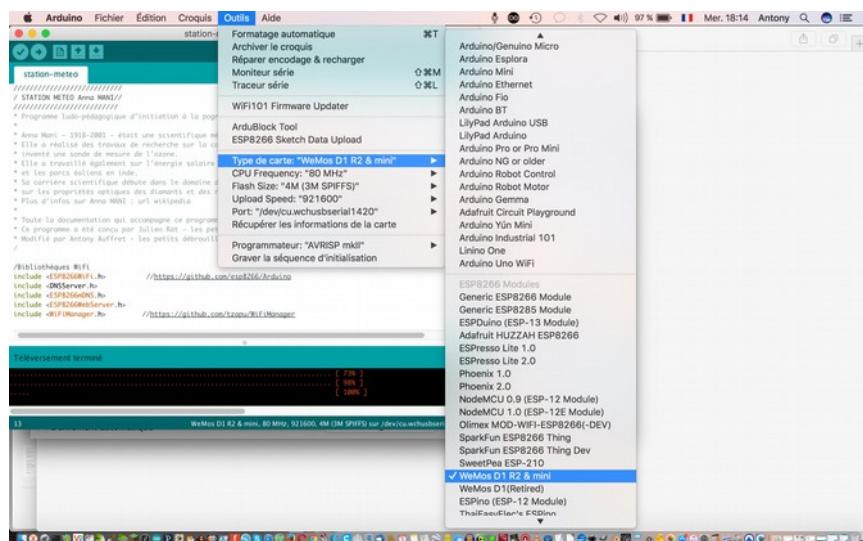
Branchez le Wemos.

Choisissez le bon type de carte.

Outil → Type de carte → Wemos D1 R2 & mini

ATTENTION !

Il faut que vous ayez configuré le logiciel Arduino pour qu'il puisse gérer les cartes Wemos (voir Annexe).

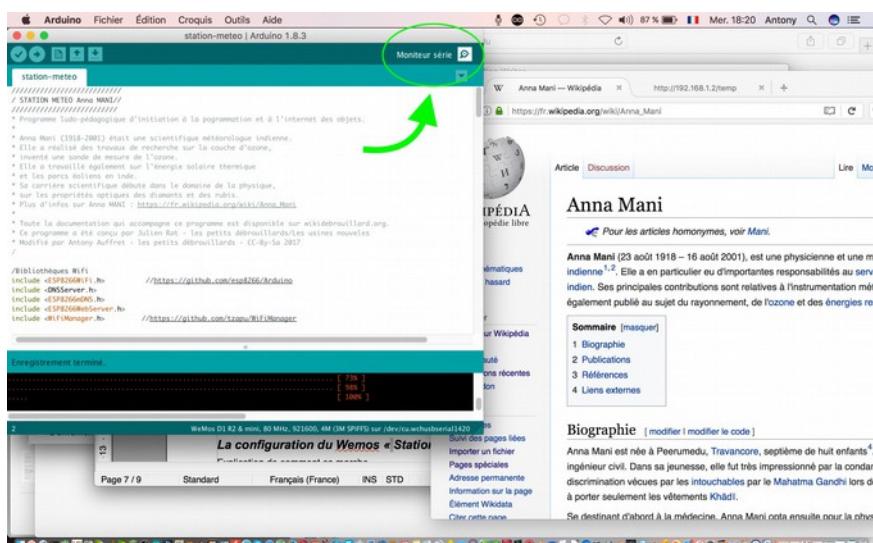


Choisissez le port.

Outils → Port : → wchusbserial1420

Ouvrez le moniteur série.

Pour cela cliquez sur
« moniteur série » sur une fenêtre du logiciel Arduino.



Redémarrez le Wemos : au démarrage, il se connecte au réseau configuré. Dès que la connexion est réussie la led bleue du Wemos s'allume.

Le moniteur série vous indique l'adresse IP de votre Wemos connecté à votre réseau Wifi.

Ici c'est **192.168.1.2**

Chez vous cela peut-être différent, cela dépend de votre réseau wifi.



```

/dev/cu.wchusbserial1420
Envoyer
Epif@110ex:~$ *WM:
*WM: AutoConnect
*WM: Connecting as wifi client...
*WM: Using last saved values, should be faster
*WM: Connection result:
*WM: 3
*WM: IP Address:
*WM: 192.168.1.2
WiFi connecté
adresse IP :
192.168.1.2

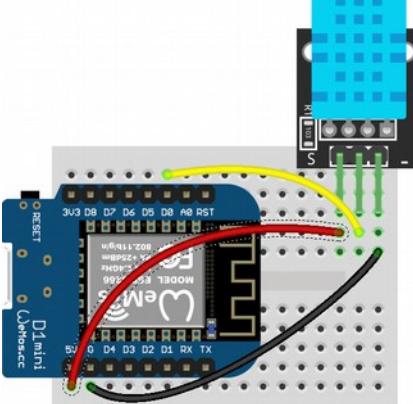
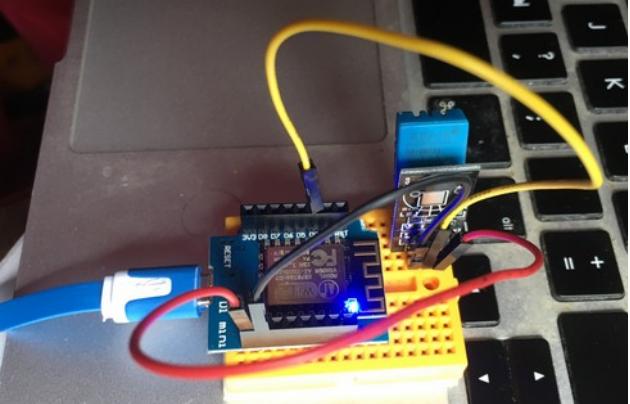
Nouvelle ligne 9600 baud Clear output

```

4.3- Premier essais : lire les données de l'Hygromètre.

4.3.1- Montage des composants

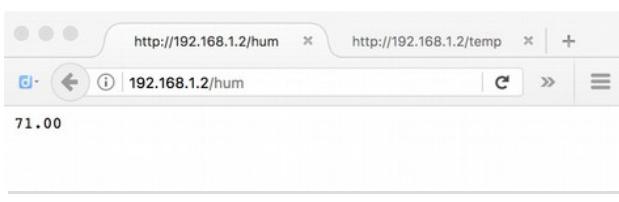
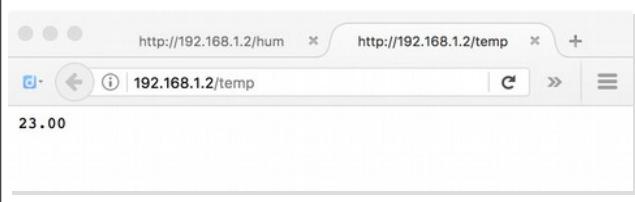
Ici le schéma de montage + photo du montage.

Schéma de montage	Réalisation
	

4.3.2- Récupérer les données

Utiliser un navigateur en mettant dans la barre d'adresse votre adresse ip suivie de « /hum » pour l'humidité et « /temp » pour la température du capteur DHT11.

Rafraîchir la page.

Humidité.	Température.
	

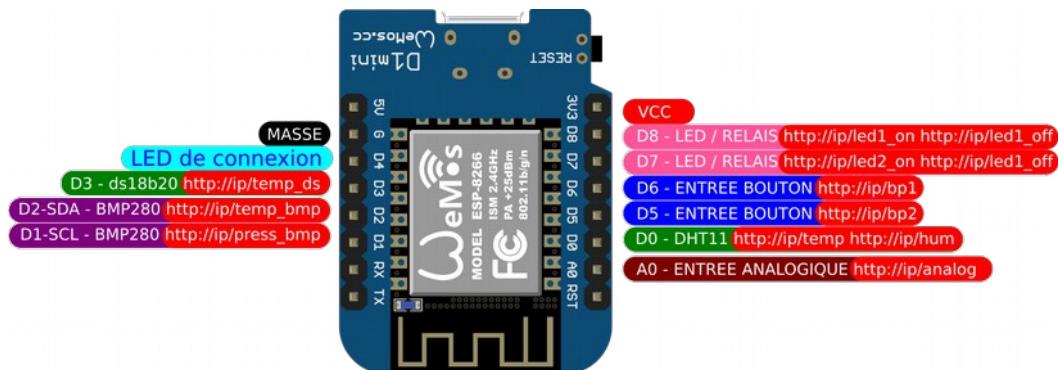
4.4- La configuration de base du Wemos « Station météo Anna MANI ».

Le cœur de la station météo Anna MANI est une carte électronique compatible Arduino qui s'appelle « **Wemos D1 mini** ».

Elle est équipée du composant « **ESP8266** » qui lui permet de créer un Point d'Accès Wifi ou de se connecter à un réseau existant en mode « Station ». Le Wemos est également utilisé comme serveur web, capable de générer des pages web.

Le code Arduino « **Station Météo Anna MANI** » doit être chargé sur le Wemos afin qu'on puisse l'utiliser.

Les différents capteurs et actionneurs de la station météo sont connectés aux différentes broches du Wemos selon ce schéma :



Les données issues des capteurs sont publiées sur des pages web délivrées par le serveur du Wemos. L'adresse web est composée de l'adresse IP de votre Wemos <http://---IP---> et de l'adresse de la donnée. Dans l'exemple précédent pour avoir la température, on utilise l'adresse suivante : <http://192.168.1.2/temp>

Capteur ou Actionneur	URL	Résultat	Si le capteur n'est pas connecté
DS18B20 - Thermomètre	http://---IP---/temp_ds	Renvoie la valeur (21 par ex.)	-127
BMP280 - Thermomètre	http://---IP---/temp_bmp	Renvoie la valeur (21 par ex.)	-144
BMP280 - Pression	http://---IP---/press_bmp	Renvoie la valeur (1013 par ex.)	1314.15
DHT11 - Thermomètre	http://---IP---/temp	Renvoie la valeur (21 par ex.)	nan
DHT11 - Hygrométrie	http://---IP---/hum	Renvoie la valeur (75 par ex.)	nan
LED1	Allumer : http://---IP---/led1_on Eteindre : http://---IP---/led1_off	Met la broche D8 en HIGH Met la broche D8 en LOW	Non concerné, ce n'est pas un capteur
LED2	Allumer : http://---IP---/led2_on Eteindre : http://---IP---/led2_off	Met la broche D7 en HIGH Met la broche D7 en LOW	Non concerné, ce n'est pas un capteur
Bouton Poussoir 1	http://---IP---/bp1	Renvoie 0 ou 1 si appui	0
Bouton Poussoir 2	http://---IP---/bp2	Renvoie 0 ou 1 si appui	0
Photo-résistance	http://---IP---/analog	Renvoie la valeur (99 par ex.)	chiffres parasites

Ainsi, il n'y a plus qu'à utiliser un navigateur pour récupérer les données.

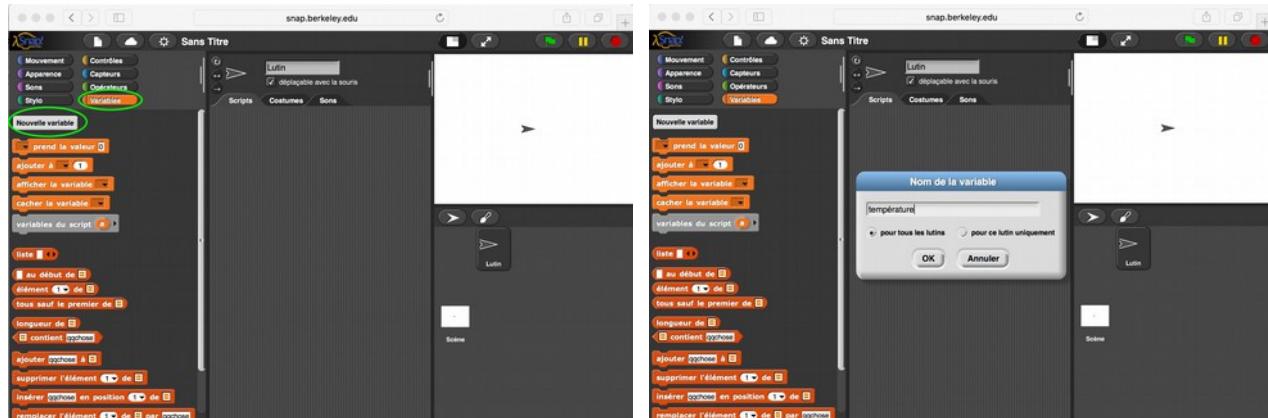
Si vous avez une grande plaque de prototypage vous pouvez tout brancher, sinon, vous pouvez effectuer les montages les uns après les autres.

L'utilisation du logiciel en ligne SNAP! permet de jouer avec les données issues des capteurs. Et comme votre Wemos peut se connecter à Internet, c'est le monde de l'internet des objets qui s'ouvre à vous !

4.5- Utiliser SNAP! pour lire les données d'un capteur

Les données des capteurs de la station météo Anna MANI sont accessibles via une URL.

Créer une variable pour stocker la donnée : il vaut toujours mieux créer des variables dont le nom est explicite, cela permet à tous les monde de comprendre rapidement à quoi elles servent : « Température » pour la température, « Pression » pour la pression, etc.



Pour créer votre variable, allez dans les blocs « **variables** », cliquez sur « **nouvelle variable** », donnez-lui un nom explicite.

Dans la colonne des blocs, la coche à gauche du nom de la variable affiche la variable dans la scène.

Pour donner à votre variable, la valeur de la température, choisissez le bloc « **prend la valeur** ».

http:// 192.168.0.30/temp

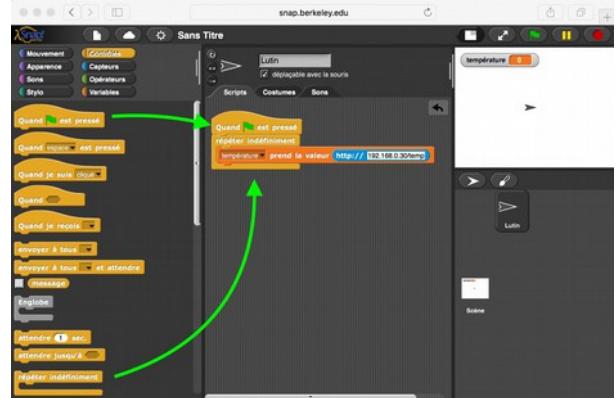
Dans les **capteurs**, le bloc « **http://** » permet d'appeler une page web. Il suffit d'y mettre l'URL du capteur qu'on souhaite (ici c'est mon adresse IP, cela est sûrement différent pour vous).



La variable « **température** » prend la valeur renvoyée par l'URL « **http://192.168.0.30/temp** ».

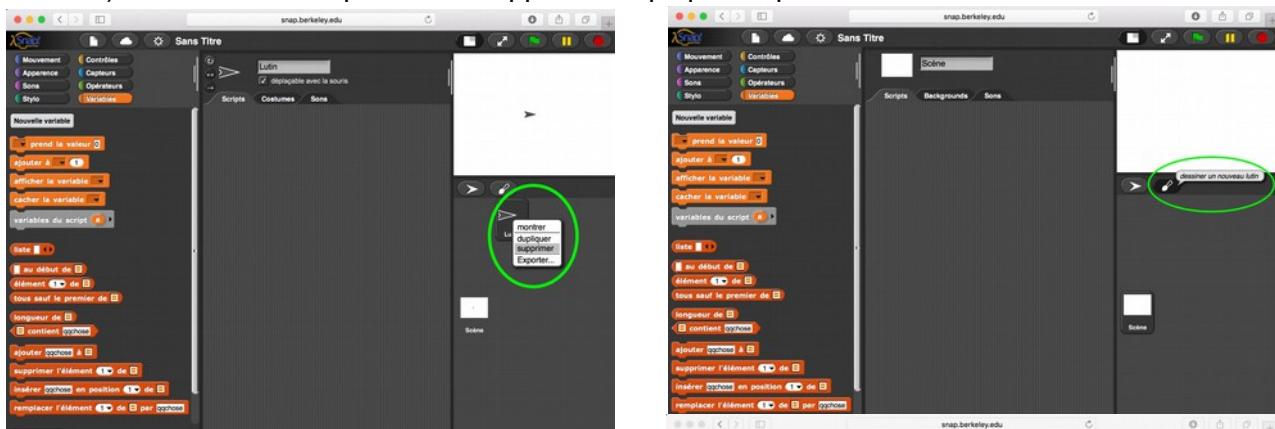
Cette valeur est actualisée en permanence grâce à la boucle « **répéter indéfiniment** ».

Pour actualiser la valeur de la température, on utilise le bloc « **répéter indéfiniment** ».



Mettre en forme les données récupérées

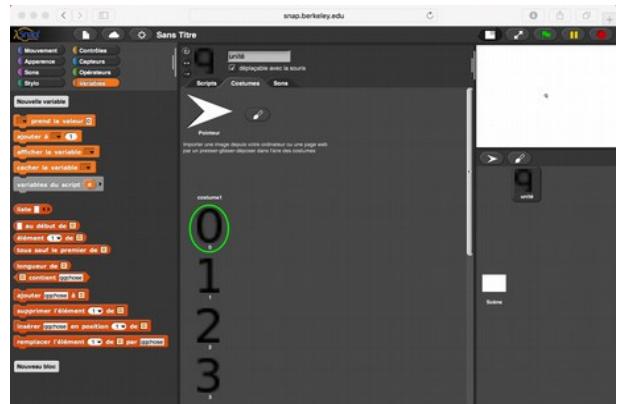
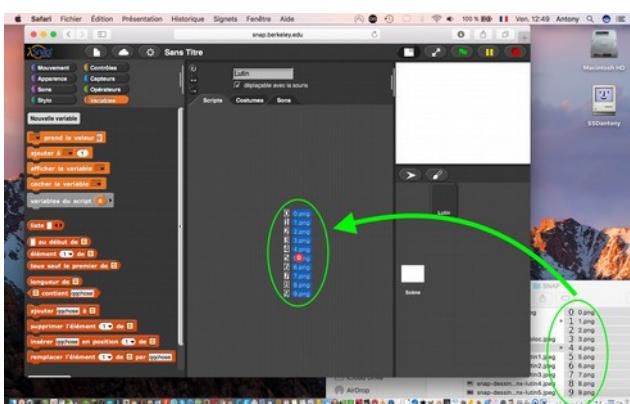
Il faut commencer par dessiner un nouveau lutin. Mais avant, si vous faites un clic-droit (ctrl-clic sur mac) sur le lutin, vous pouvez le supprimer/dupliquer/exporter/montrer.



Il y a deux types de lutins (lutin et lutin tortue), à ce stade, je ne connais pas la différence.

Lorsque vous cliquez sur l'icône en forme de pinceau « **dessiner un nouveau lutin** », une fenêtre « **Paint éditeur** » apparaît. Pour notre affichage de données nous avons simplement besoin d'un lutin qui a 10 « **costumes** » : les chiffres de 0 à 9.

Pour cela, nous allons glisser-déposer les images nécessaires. Mais avant, il faut fermer la fenêtre « **paint editor** » en cliquant sur « **OK** ».

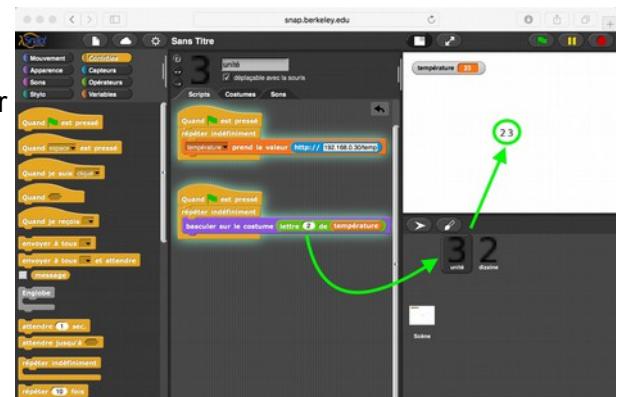


Les « **costumes** » sont nommés par leur chiffre.

Dans la gamme de température ordinaire, la valeur de la température ne comporte que 2 chiffres.

On crée 2 lutins : un pour les dizaines, un pour les unités. Le premier chiffre de la température c'est une dizaine, le second, une unité.

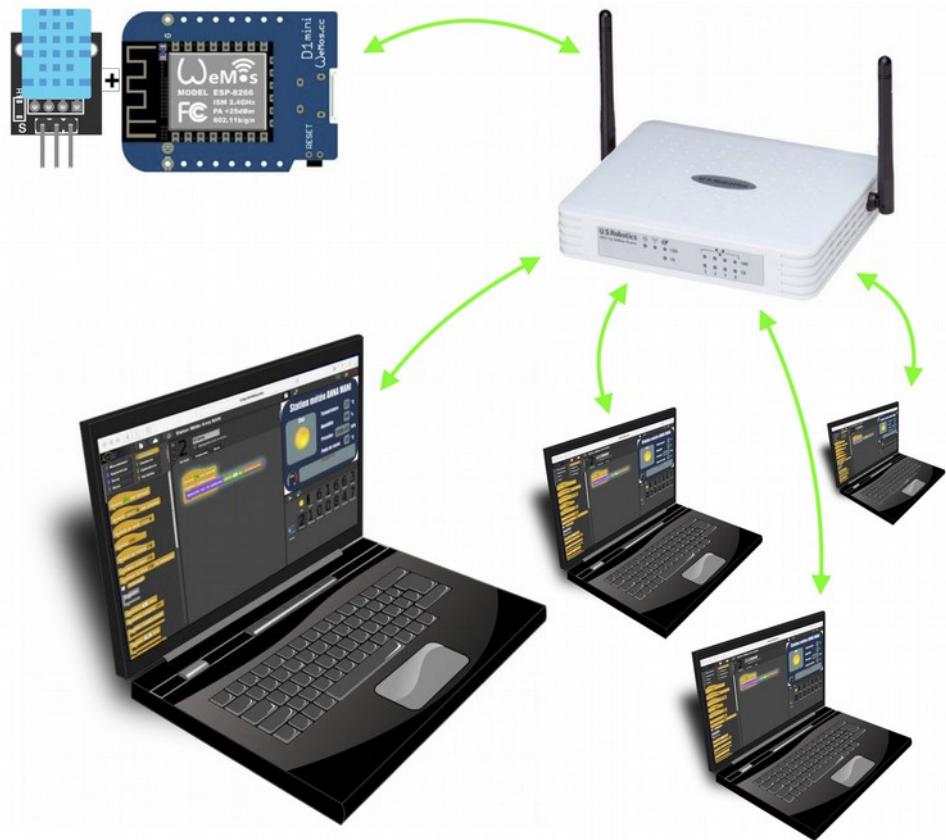
Si il fait 23°C, la « **lettre 2** » de « **température** » est « **3** » et la « **lettre 1** » est « **2** ». On fait basculer sur le costume correspondant.



5- Les fiches d'activité

- **Fonctionnement de base de la station météo Anna MANI.**
- **Un indicateur Jour/nuit**
- **Un thermomètre**
- **Un hygromètre**
- **La station Anna MANI au complet !**
- **Le bouton sans fil qui appelle la LED !**
- **Inventer une activité**

5.1- Fonctionnement de base de la station météo Anna MANI



Le Wemos D1 mini fait fonctionner un serveur web qui renvoie des pages web avec les données qu'il récupère des capteurs.

Il faut le connecter à un routeur wifi (une « box » comme on dit sur les publicités).

Tous les ordinateurs connectés à ce routeur wifi peuvent appeler des données sur la station météo.

On peut y brancher un capteur analogique. Par défaut c'est une photo-résistance, mais ça peut être n'importe quel capteur analogique (potentiomètre par exemple).

Il est possible de contrôler 2 sorties du Wemos via le web pour allumer une led ou activer un relais.

Il y a également deux entrées pour un bouton poussoir.

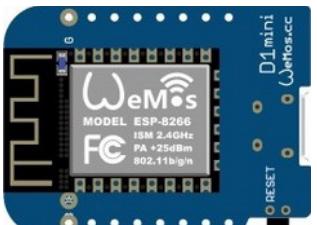
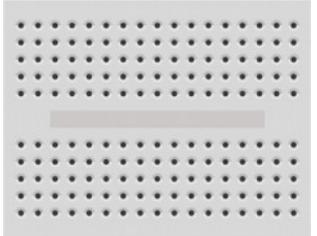
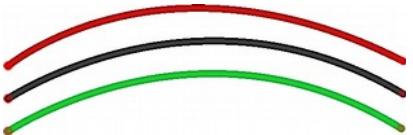
L'utilisation de **SNAP!** permet d'exploiter la station météo Anna MANI de manière créative et ludique, en s'appuyant sur la programmation par bloc.



5.2- Un indicateur Jour/nuit

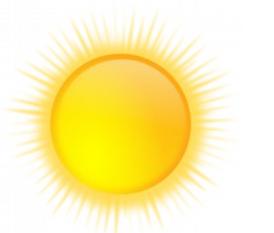
5.2.1- Le défi ingénieur : Comment réaliser un indicateur Jour/Nuit ?

Le matériel à disposition :

Nom et propriété	Pictogramme	Précision
Wemos D1 mini		La base
Photo-résistance		La valeur dépend de la lumière. Plus il y a de lumière, moins il y a de résistance. http://---IP---/analog
Résistance		Elle sert à limiter le courant qui passe dans la photo-résistance
Plaque de prototypage		Pour nos branchements
3 Fils Dupont		Pour connecter nos composants

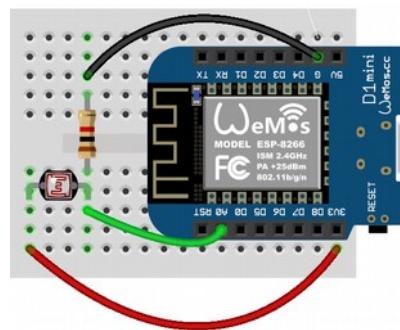
Pour le programme :

une image de soleil et une de lune pour symboliser le jour et la nuit.



5.2.2- Si vous voulez aller plus vite avec l'indicateur Jour/nuit

Schéma de brochage :



Proposition de programme :

Créez un lutin « jour-nuit » ayant deux costumes : un costume « soleil » pour symboliser le jour, un costume « lune » pour symboliser la nuit (par exemple).

Stockez la valeur de la photo-résistance dans une variable (ici on l'a nommée « analog »).



Résultat :

Valeur de « analog » supérieure à 40	Valeur de « analog » inférieure à 40
 A yellow sun icon with radiating rays, representing a value above 40. analog 51	 A grey moon icon with craters, representing a value below 40. analog 13

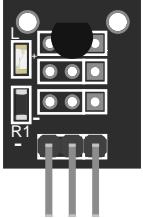
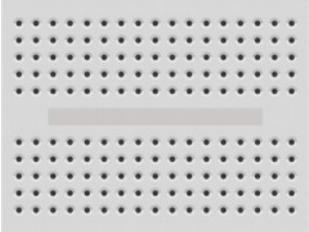
D'autre défi ?

- Allumer une led quand la lumière est trop basse !
 - Une deuxième lorsqu'elle la lumière est trop forte ?

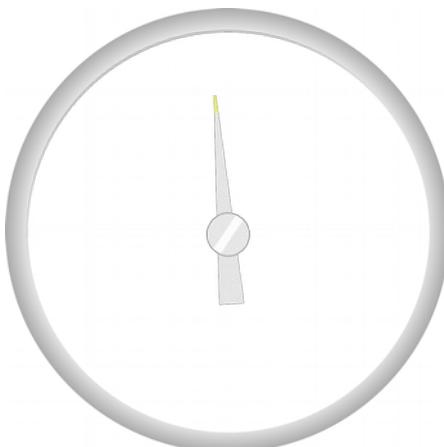
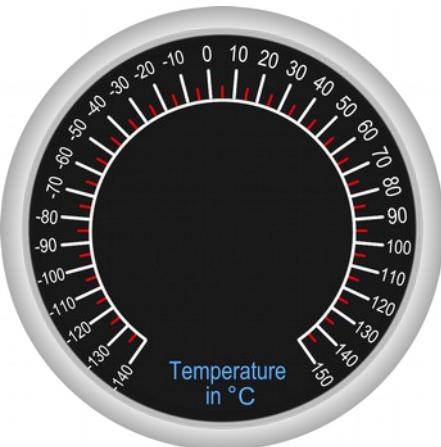
5.3- Un thermomètre

5.3.1- Le défi ingénieur : Comment réaliser un affichage de température à aiguille numérique ?

Matériel à disposition :

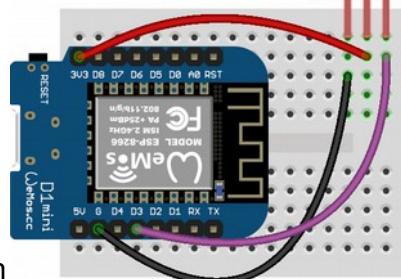
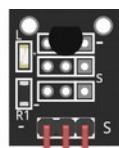
Nom et propriété	Pictogramme	Précision
Wemos D1 mini		La base
Capteur de température DS18B20		Il mesure la température de -55°C à 125°C avec une précision de 0.5°C http://---IP---/temp_ds
Plaque de prototypage		Pour nos branchements
3 Fils Dupont		Pour connecter nos composants

Pour le programme : Sur le site <https://openclipart.org/> vous trouverez des images pour vos programmes. Voici, par exemple une image de cadran et une autre d'aiguille.



5.3.2- Pour aller plus vite avec le thermomètre

Schéma de brochage :



Proposition de programme :

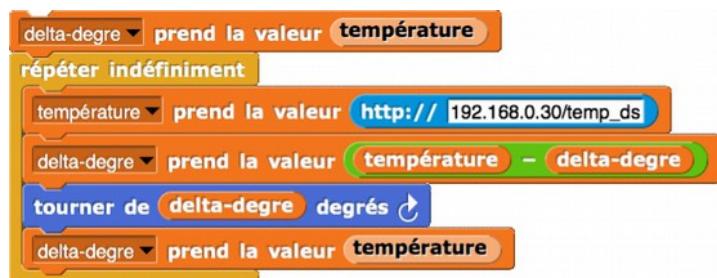
Pas facile de faire un thermomètre rotatif ! Dans SNAP!, la rotation faudra donc calibrer le thermomètre. Ensuite, mettre l'aiguille en position puis la déplacer suivant les variations de température.



Pour calibrer on fait tourner l'aiguille avec les touches « flèche haut » et « flèche bas ».

C'est une boucle conditionnelle qui permet de calibrer l'affichage. Une fois la calibration effectuée, on appuie sur la touche « espace ».

La variable « **température** » stocke la valeur renvoyée par le capteur. La variable « **delta-degre** » sert à stocker la variation de température pour faire tourner l'aiguille d'autant.



Il ne faut pas oublier de positionner l'aiguille à la première température mesurée



Pour plus de convivialité on peut afficher des consignes explicatives au lancement du programme.



Et voici le résultat !

Si l'aiguille et le cadran ne sont pas alignés, vous pouvez utiliser l'instruction « aller à X : 0 Y : 0 » sur chacun des lutins.



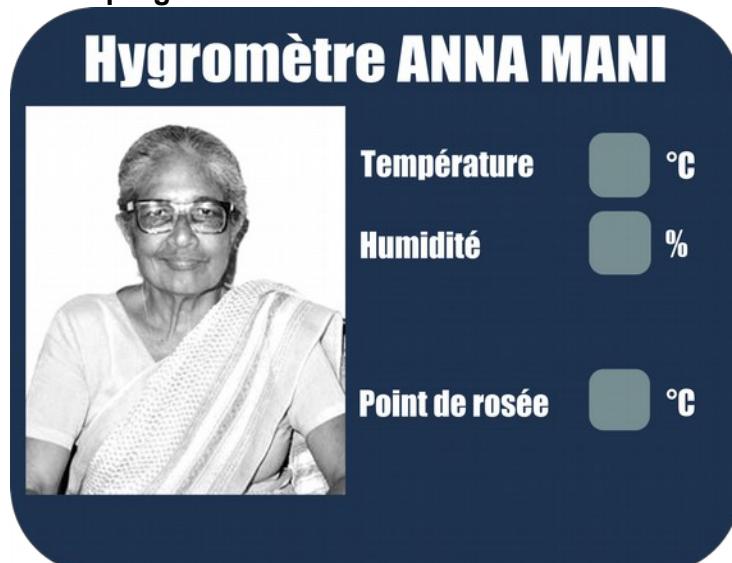
5.4- Un hygromètre

5.4.1- Le défi ingénieur : Comment réaliser un hygromètre qui calcule le point de rosée ?

Matériel à disposition :

Nom et propriété	Pictogramme	Précision
Wemos D1 mini		La base
Capteur de température et d'humidité DHT1		<ul style="list-style-type: none"> - Alimentation : 3,3V - 5V - Etendue de mesure température : 0°C à 50°C ± 2°C - Etendue de mesure humidité : 20-90%RH ±5%RH <p>http://---IP---/temp http://---IP---/hum</p> <p>l'URL renvoie « nan » si le capteur n'est pas branché.</p>
Plaque de prototypage		Pour nos branchements
3 Fils Dupont		Pour connecter nos composants

Pour le programme :



utilisez des images pour l'affichage des chiffres et un fond de scène adapté comme zone d'affichage.

6 7 8 9
 2 3 4 5
 0 1

5.4.2- Pour aller plus vite avec l'hygromètre !

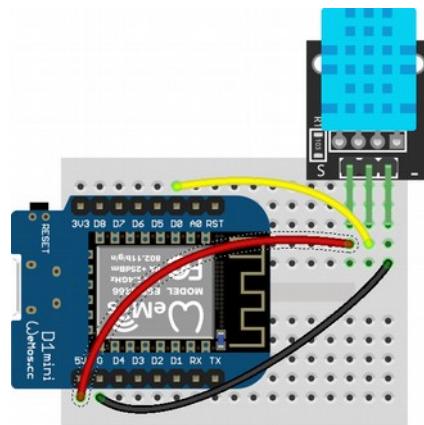
Schéma de brochage :

Pour le programme :

On trouve sur Wikipédia la formule de calcul du point de rosée :

https://fr.wikipedia.org/wiki/Point_de_ros%C3%A9e

- T, température mesurée : $0^{\circ}\text{C} < T < 60^{\circ}\text{C}$
- RH, humidité relative : $0,01 \text{ (1\%)} < RH < 1,00 \text{ (100\%)}$
- Tr, point de rosée : $0^{\circ}\text{C} < Tr < 50^{\circ}\text{C}$



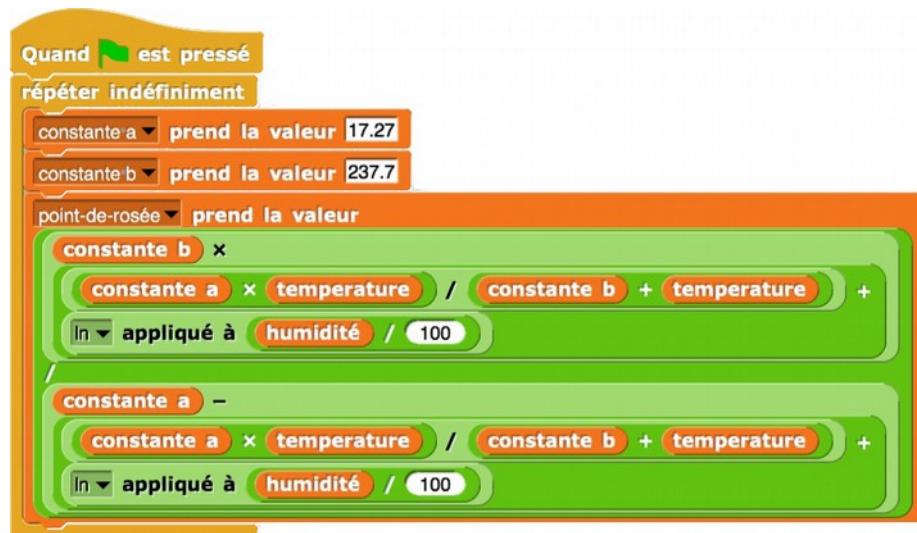
$$Tr = \frac{bF}{a - F} \quad F \text{ est une fonction utilisant les variables } T \text{ et } RH \text{ comme ceci } F = \frac{aT}{b + T} + \ln RH$$

avec a = 17,27 et b=237,7 en °C.

Pour stocker les valeurs de a et b on crée les variables :

- **constante a**, on y met la valeur de a
- **constante b**, on y met la valeur de b

Ensuite, on utilise les opérateurs pour effectuer le calcul :



Référez-vous à la fiche « **mettre en forme les données récupérées** » pour afficher les valeurs de la température, de l'humidité et du point de rosée.

Encore plus loin :

- Sauriez-vous créer un autre fond d'écran ?
- Sauriez-vous utiliser d'autre chiffres (en en prenant sur le site web <https://openclipart.org/> par exemple) ou à partir d'une typographie de votre ordinateur.

5.5- La station météo Anna MANI au complet !

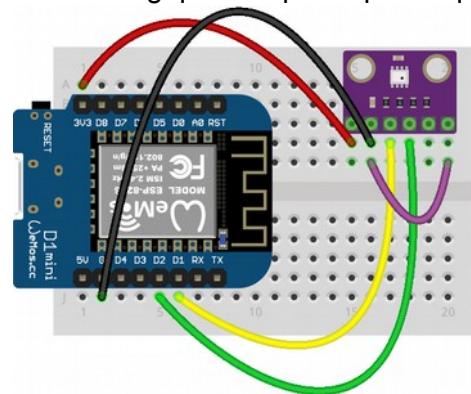
5.5.1- Le défi ingénieur : Tout à fond !

Choisissez parmi les composants pour réaliser une station météorologique complète qui indique la température, l'humidité, le jour ou la nuit, le point de rosée et qui, pourquoi pas, fait des prévisions météo.

Matériel : Vous avez tout le matériel à disposition !

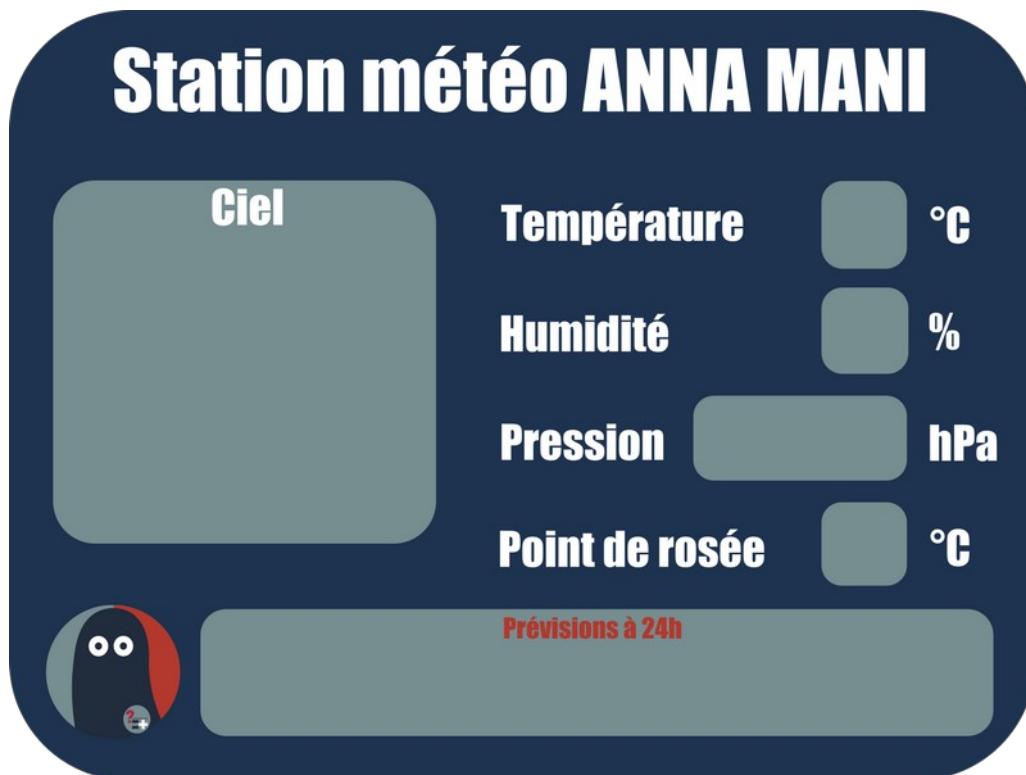
Pour faciliter la réalisation, voici le brochage du capteur de pression Bmp280

→



Pour le programme :

Voici une « scène » pour réaliser une belle station météo

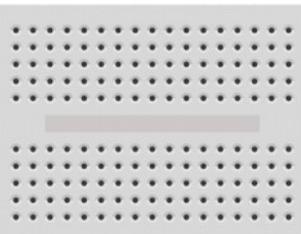


Pour les prévisions météo, il existe la méthode dite « Moreux ». L'abbé Moreux a mis en place dans les années 1930 une méthode permettant de prévoir le temps des prochaines 24 heures avec une précision de l'ordre de 80%. Dans cette méthode, deux paramètres viennent s'ajouter à la pression atmosphérique : la **saisonnalité** et la **direction du vent**. Découvrir tous les détail ici : <http://leguidemeteo.com/prevoir-le-temps-a-partir-de-son-barometre/>

5.6- Le bouton sans fil qui appelle la LED !

5.6.1- Le défi ingénieur : Comment allumer une LED et une ampoule SNAP! à distance avec un bouton poussoir !

Matériel à disposition :

Nom et propriété	Pictogramme	Précision
Wemos D1 mini		La base
Bouton poussoir		Renvoie 0 ou 1 si appui (branché en D6). http://---IP---/bp1
LED		Branchée en D8, elle est gérée par ces URL : Allumer : http://---IP---/led2_on Eteindre : http://---IP---/led2_off
Plaque de prototypage		Pour nos branchements
4 Fils Dupont		Pour connecter nos composants

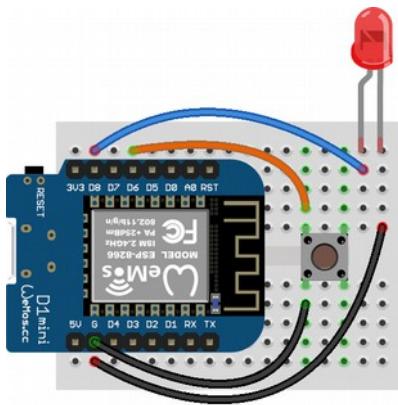
Pour le programme :

Voici deux costumes (allumé et éteint) pour un lutin « ampoule ».



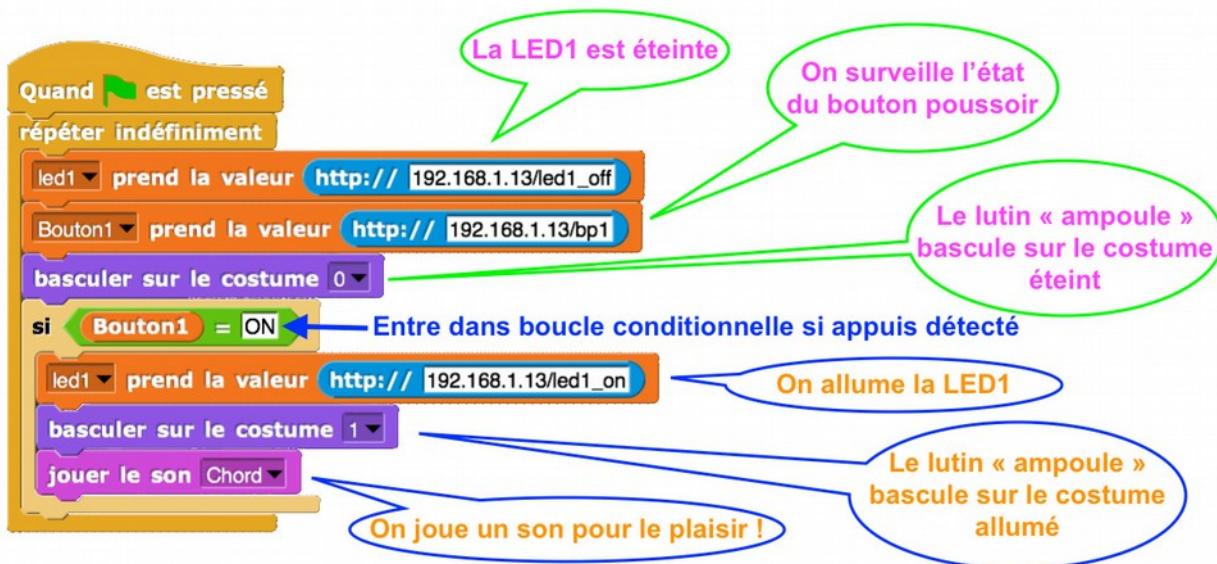
5.6.2- Pour aller plus vite avec le bouton sans fil qui appelle la LED !

Schéma pour le brochage :



Pour le programme SNAP! :

On a commenté le programme pour aider à sa compréhension.

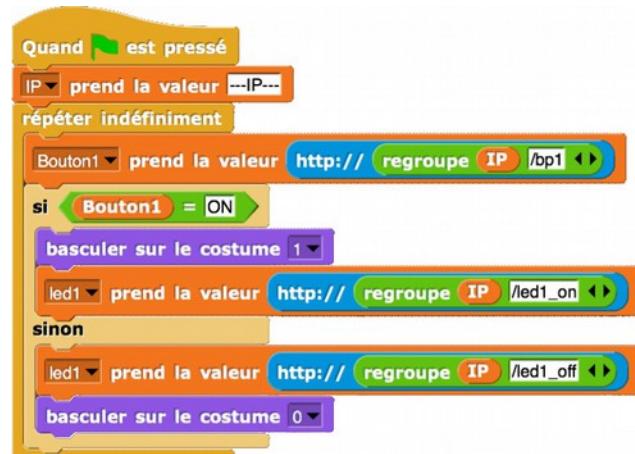


C'est un manque de SNAP!, tout comme SCRATCH d'ailleurs, il n'est pas possible de commenter les programmes. Il faut insister sur le fait que commenter ses programmes est une bonne pratique pour les autres, mais aussi pour soi.

ASTUCE !

Comme il parfois laborieux de re-taper les adresses IP, l'astuce consiste à la mettre dans une variable une bonne fois pour toute – ici elle est nommée « IP ». Pour faire une requête on utilise alors l'opérateur « regroupe ».

PS : j'ai ôté le son, c'était relou !



Allons plus loin dans les défis :

- et si vous programmiez un « télécran » en SNAP! avec deux boutons ?

5.7- Inventer une activité

La station météo Anna MANI est conçue pour l'apprentissage du code et de l'internet des objets.

Le microcontrôleur Wemos D1 mini est préprogrammé, il est prêt à recevoir des capteurs, à répondre au logiciel **SNAP!** et à communiquer avec le monde entier avec internet.

Cette station météo est donc faite pour servir d'outil pédagogique. Au pédagogues de s'en servir pour imaginer et mettre en oeuvre des activités pédagogiques.

Dans ce document nous proposons des fiches simples, inspirées des fiches auto-correctives de la pédagogie Freinet. Elle ont deux faces :

- Une face qui pose un « défi ingénieur » et donne quelques ressources.
- Une autre face qui présente le schéma de brochage des composants et des pistes pour écrire le code **SNAP!**.

Ces fiches sont conçues pour être données aux apprenants qui peuvent avancer à leur rythme, se révéler plus ou moins d'indices et en tout cas, découvrir une solution au problème posé.

Quelques idées :

- Un détecteur de passage avec la photo-résistance ?
- Un détecteur de café chaud avec le thermomètre ?

Comment avoir des idées ?

Capteur	A quoi pourrait-il servir ?
photorésistance	<ul style="list-style-type: none">• Déetecter le passage de quelqu'un• Une porte ouverte• L'allumage d'une lampe• L'allumage d'un écran• ...
pression	<ul style="list-style-type: none">• Mesurer la pression dans des expériences scientifiques• Jeu de celui qui souffle le plus fort• ...
humidité	<ul style="list-style-type: none">• Trouver le coin le plus humide
température	<ul style="list-style-type: none">• Bouilloire électrique• Cafetière électrique• quelqu'un assis sur sa chaise• ...

6- Annexes :

- 1. Le logiciel SNAP!.**
- 2. La carte Wemos D1 Mini.**
- 3. Charger le programme sur le Wemos avec le logiciel Arduino**
- 4. Utiliser IFTTT**
- 5. Le bouton qui twite avec IFTTT**
- 6. Ressources humaines et documentaires**

6.1- Le logiciel SNAP!.

SNAP! (qui s'appelait anciennement BYOB – Build Your Own Bloc – construisez vos propres blocs) est un langage de programmation visuel par glisser-déposer. Il s'agit d'un prolongement de **SCRATCH** (Logiciel du MIT) qui vous permet de créer vos propres blocs. Il comporte également des fonctionnalités supplémentaires qui en font un outil sérieux pour une introduction à l'informatique pour les étudiants du secondaire ou des collèges.



SNAP! est un logiciel qui fonctionne dans presque tous les navigateurs (basé sur HTML5 et javascript) et sur toutes les plateformes (Linux, Mac, Windows).

Il est porté par une équipe de l'**Université de Californie à Berkeley** sous licence libre AGPL.

Recommandations :

Créez-vous un compte pour pouvoir enregistrer vos programmes (soit dans votre navigateur, sur le Cloud de SNAP!).

Exportez vos programmes en XML :

C'est très utile pour pouvoir les partager ou les ouvrir dans d'autres navigateurs.

Importez des programmes en XML :

En quelques clics, vous importez et testez les créations des autres.

L'interface de SNAP! :

The screenshot shows the Snap! interface. On the left is the block palette with categories like Mouvement, Contrôles, Apparence, Capteurs, Sons, Stylo, and Catégories de blocs. The main area shows a script editor with various scripts (e.g., 'Quand vert est pressé', 'répéter indéfiniment') containing blocks for movement, control, and sensors. To the right is a preview window titled 'Station météo ANNA MANI' showing a sun icon and weather data (Température: 23 °C, Humidité: 71 %, Pression: 1008.11 hPa, Point de rosée: 17 °C). Below the preview is a 'Scène' panel with a grid for costumes and a 'Scène' tab.

Où trouver SNAP! ?

Sur le site de l'Université de Californie à Berkeley :

<http://snap.berkeley.edu/snapsource/snap.html>

Si SNAP! n'est pas accessible via l'URL principale, il existe des sites miroirs :

- <http://bjc.edc.org/snapsource/snap.html>
- <https://snap.apps.miosoft.com/>
- <http://media.mit.edu/~harveyb/snap>
- <http://cs10.org/snap>

Comment apprendre à utiliser SNAP! ?

Les ressources de **SCRATCH** sont innombrables et très bien faites. Les compétences acquises sur **SCRATCH** sont transposable à l'identique sur SNAP!.

Vous trouverez un ensemble de ressources en français sur Wikidébrouillard, à la page « **Trousse à outils pour animer avec SCRATCH** » : http://wikidebrouillard.org/index.php?title=Trousse_%C3%A0_outils_pour_animer_avec_Scratch

En anglais, le manuel de référence de **SNAP!** : <http://snap.berkeley.edu/SnapManual.pdf>

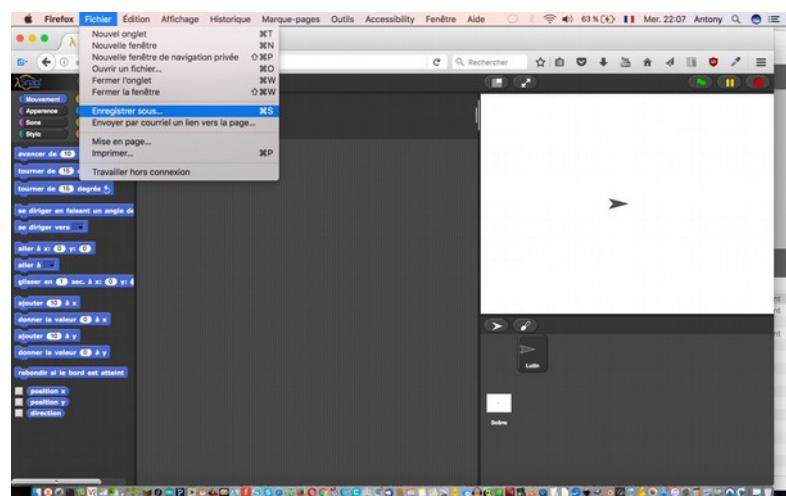
ASTUCE : Comment avoir SNAP! sur son ordinateur hors connexion ?

Avec le navigateur **Firefox** :

Allez sur la page d'accueil de **SNAP!**

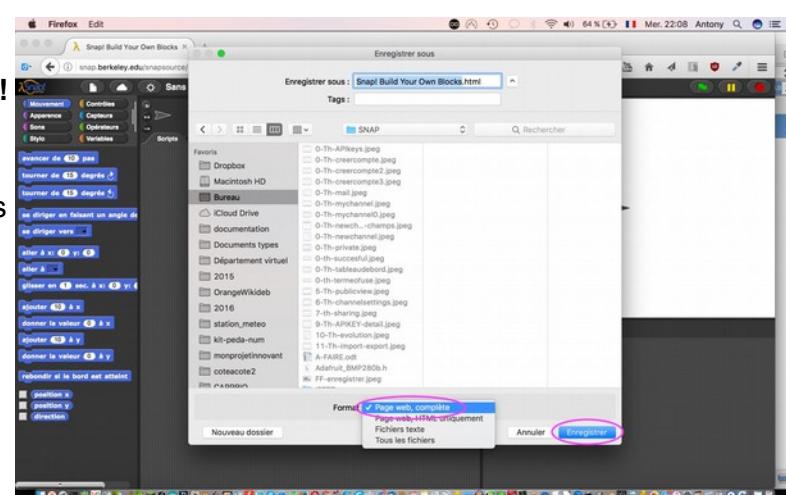
Puis, cliquez sur **Fichier** →

Enregistrer sous

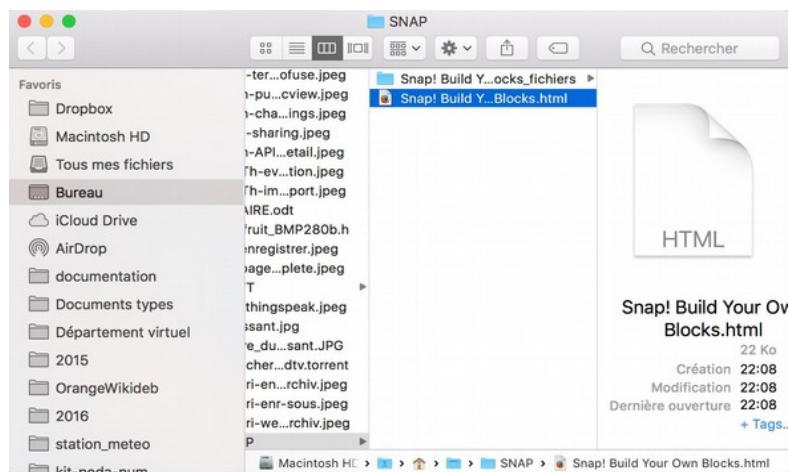


Dans la fenêtre qui s'ouvre choisissez comme format : **Page web complète** !

Puis cliquez sur « **enregistrer** »
(choisissez ou créez un dossier dans lequel vous retrouverez facilement ces fichiers).

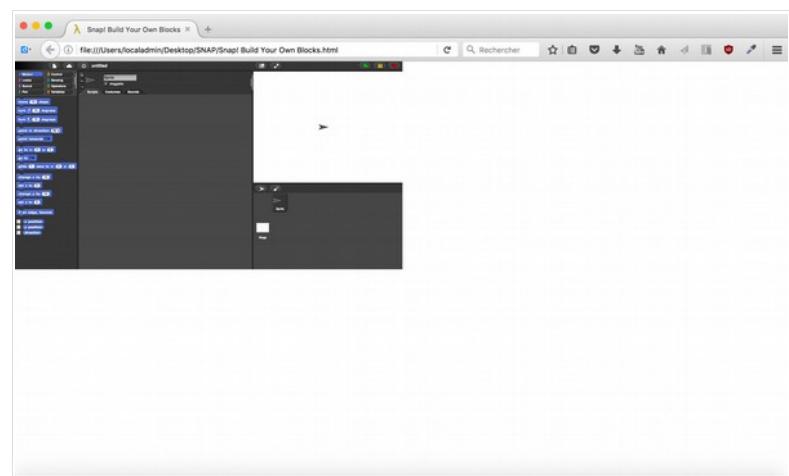


Pour avoir accès à Snap! hors connexion, allez dans le dossier dans lequel vous avez enregistré SNAP!
Puis cliquez sur la page « **Snap!**
Build Your Own Blocks.html ».

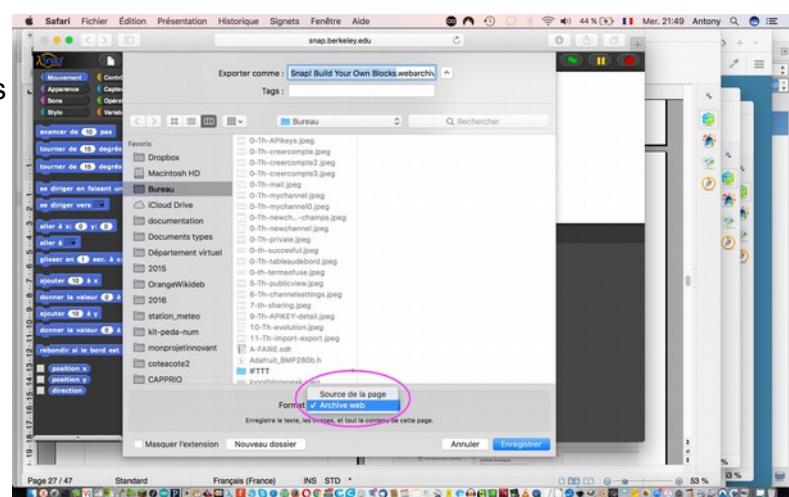


Etonnamment **SNAP!** Apparaît en petit mas si redimensionnez la taille de la fenêtre, il s'y adapte.

En revanche, vous ne pouvez pas changer la langue.



Pour les autres navigateurs, faites vos tests ! Par exemple avec **Safari** c'est dans le format « **webarchive** » que cela fonctionne.



Amusez-vous bien !

6.2- Utiliser la carte Wemos D1 mini

Cette petite carte est livrée en kit pour une utilisation variée :

- Souder fils et composants directement sur la carte, pour des montage définitifs.
- Souder des broches femelles simples.
- Souder des broches malles simples, pour brancher la carte sur une plaque de prototypage.
- Souder des broches femelles traversantes pour allier le coté pratique des deux.

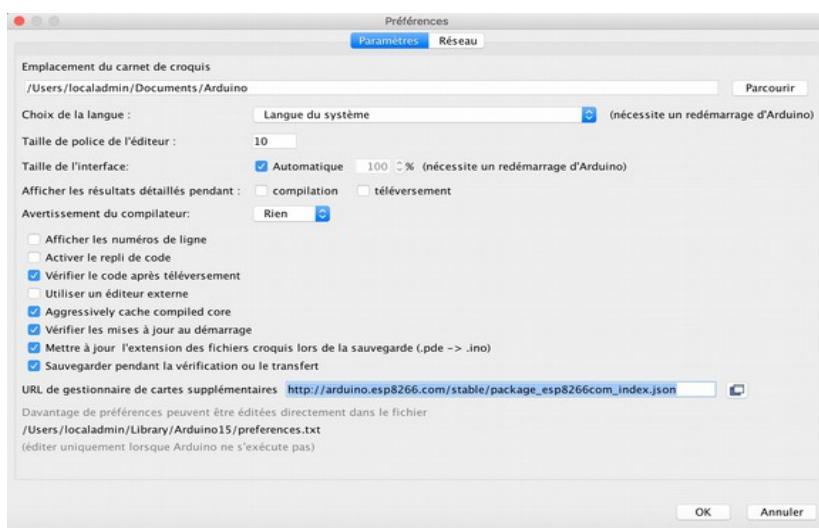


6.2.1- Une petite manip' pour utiliser le logiciel Arduino avec le Wemos D1 Mini :

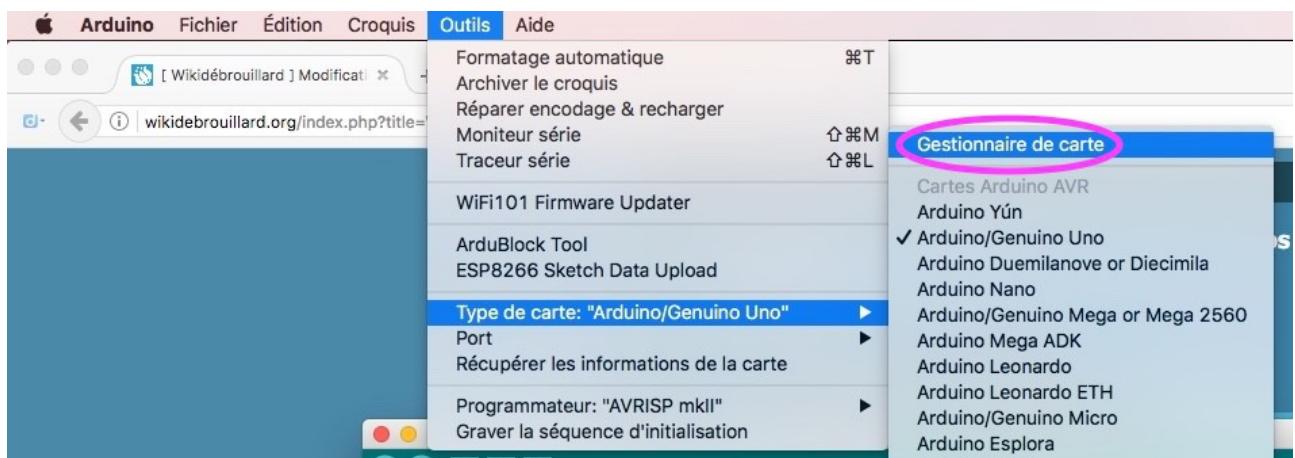
1 - Pour mac et windows, il faut installer les pilotes (drivers) téléchargeable ici :
<https://wiki.wemos.cc/downloads>

2 – Ensuite, il faut ajouter la carte au logiciel Arduino. Pour cela allez dans les « Préférences » d'Arduino, ajoutez l'URL suivante dans « URL de gestion de cartes supplémentaires » :

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



Puis, cliquez sur **outils** → **Type de carte** → **Gestionnaire de carte**.



La, l'interface de gestion des cartes s'ouvre, tapez dans la recherche « esp8266 ».

Cliquez sur « **esp8266 By ESP8266 Community** » puis sur « **Installer** ».



Le logiciel Arduino est prêt à gérer votre carte Wemos D1 mini !

Vous retrouvez ces consignes sur le Wiki du Wemos : https://wiki.wemos.cc/products:d1:d1_mini

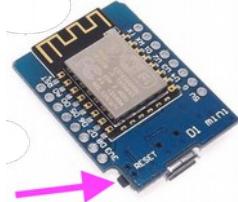
AVERTISSEMENT :

Les sources de dysfonctionnement sont nombreuses. Mais ne vous découragez pas. Il faut être persévérent et vous apprendrez à résoudre les problèmes rencontrés.

La bidouille informatique n'est ni évidente, ni facile. Elle est exigeante en terme de rigueur et de complexité.

En prenant le temps d'apprendre et pratiquer, on acquière une bonne autonomie assez rapidement.

SOURCES D'ERREUR

Symptôme	Idée de solution
Votre ordinateur plante quand vous branchez le Wemos (Kernel Panic sous mac par exemple).	Il faut installer les pilotes → https://wiki.wemos.cc/downloads
Le logiciel Arduino vous renvoie l'erreur : <code>error: espcomm_open failed</code> <code>error: espcomm_upload_mem failed</code>	C'est un problème de communication entre l'ordinateur et le wemos. Les causes peuvent être multiples : <ul style="list-style-type: none">• Vérifiez le branchement de votre wemos en USB à l'ordinateur.• Vérifiez que votre câble USB → MicroUSB est bien un câble qui transfère les données (certains servent juste à charger les téléphones).• Changez de câble USB → MicroUSB.
Impossible de charger le programme sur le wemos.	 <p>Pressez le bouton « reset » pendant quelques secondes lorsque le wemos est branché à votre ordinateur.</p>

6.3- En savoir plus sur le Wemos D1 mini

Le **Wemos D1 mini** est basé sur l'**esp8266**. Il s'agit, comme l'**Arduino**, d'un **microcontrôleur**. Il est doté en plus du **wifi**.

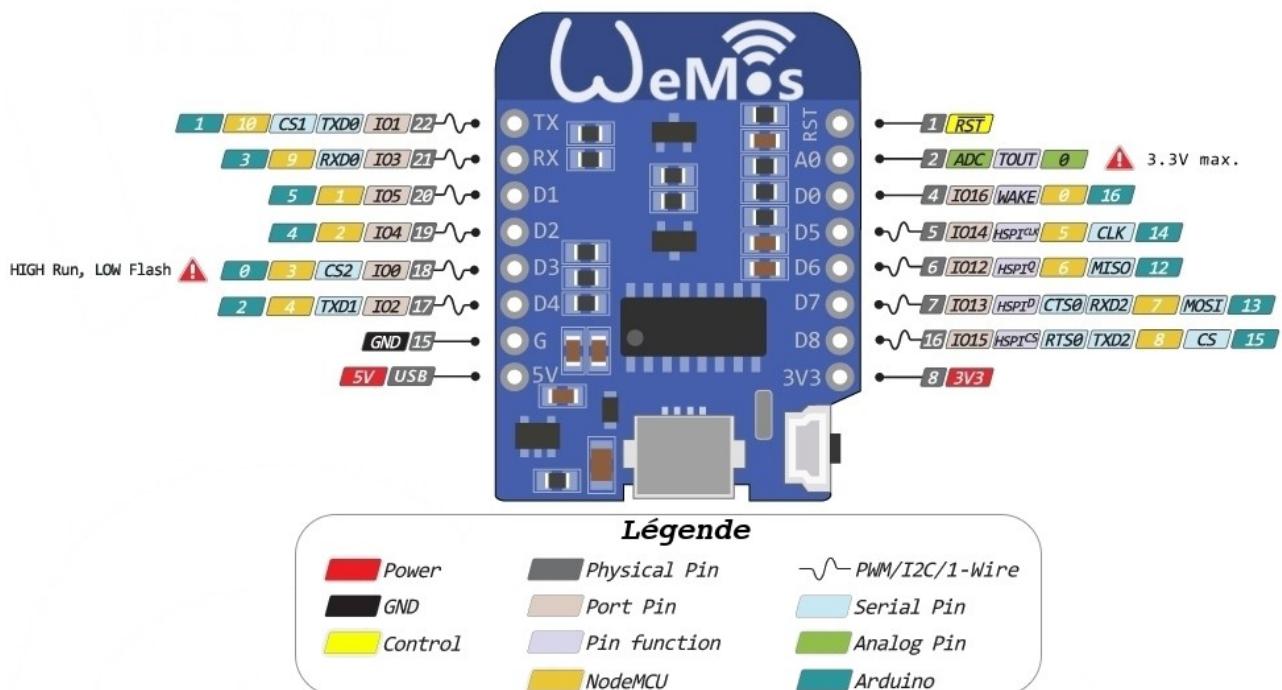
Il existe beaucoup d'autres cartes basées sur l'esp8266.

On peut programmer le Wemos D1 mini de différente manière. Il nous intéresse car on peut l'utiliser comme un Arduino, avec le même logiciel.

Ce que sait faire le Wemos est remarquable :

- C'est une sorte d'Arduino.
- Il peut se connecter en wifi à une box, à un ordinateur, à l'internet, à un autre Wemos.
- Il peut être un simple « client », c'est à dire consulter des informations sur internet, interroger des serveurs,...
- Il peut être serveur, c'est à dire envoyer des informations, répondre à des requêtes, servir des pages web. Et on peut interagir avec le Wemos par ce biais.
- Il peut être point d'accès wifi.

Pour un utilisateur débutant, la principale différence entre l'Arduino et le Wemos ce sont les broches. Il n'y a qu'une broche analogique sur le Wemos (6 pour Arduino), 11 broches numériques sur le wemos (14 pour Arduino). Le Wemos fonctionne en 3,3V. Il faut veiller à ne jamais envoyer plus de 3,3V dans ses broches au risque de le détruire.



On peut l'acheter pour moins de 3€ sur le site chinois Aliexpress.

En savoir plus :

Wikipédia : <https://fr.wikipedia.org/wiki/ESP8266>

En anglais, le wiki du fabricant : https://wiki.wemos.cc/products:d1:d1_mini

6.4- Charger le programme sur le Wemos avec le logiciel Arduino

Pour programmer le wemos, nous utilisons le logiciel Arduino.

Nous devons ajouter quels outils complémentaires à ce logiciel pour deux raisons :

- Lui donner les capacités de gérer la carte Wemos D1 mini.
- Le doter de « bibliothèques » pour gérer le Wifi et les capteurs.

Attention en anglais le terme « librairies » est un faux-amis. Il se traduit par bibliothèque. Les bibliothèques sont des fichiers de code qu'il serait laborieux de retaper à chaque fois qu'on en a besoin.

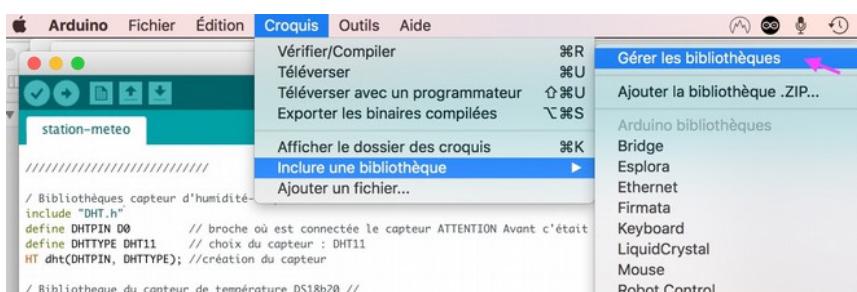
6.4.1- Ajouter les bibliothèques nécessaires

Pour la station météo Anna MANI nous avons besoin des bibliothèques suivantes :

- **wifimanager** - <https://github.com/tzapu/WiFiManager/archive/master.zip>
- **adafruit BMP280** - https://github.com/adafruit/Adafruit_BMP280_Library/archive/master.zip
- **dallas ds18b20** - <https://github.com/milesburton/Arduino-Temperature-Control-Library/archive/master.zip>
- **dht11** - <https://github.com/adafruit/DHT-sensor-library/archive/master.zip>

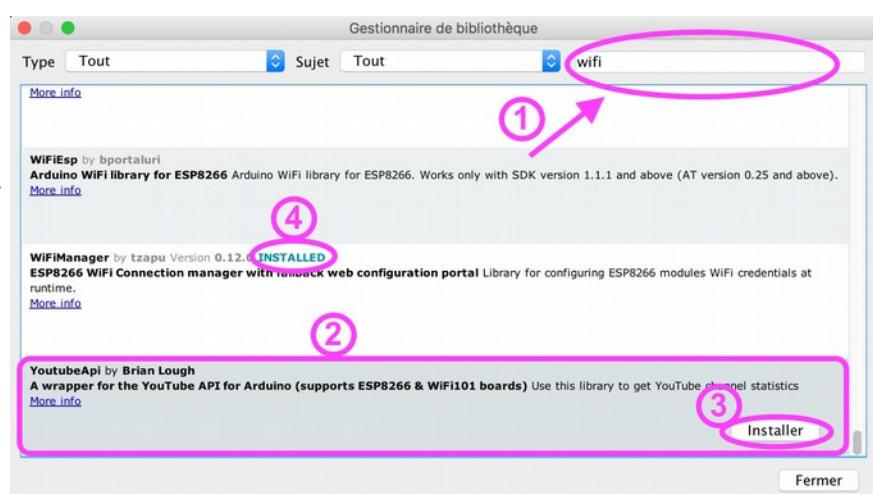
Pour installer les bibliothèques, il existe plusieurs manières. Le logiciel vous aide directement à installer les bibliothèques. Pour le cas où certaines ne seraient pas disponibles, essayez la seconde méthode.

Commencez par celle-ci :
ouvez le logiciel Arduino puis
allez dans **Croquis** → **include une bibliothèque** → **gérer les bibliothèques**



La fenêtre de gestion des bibliothèques s'ouvre :

1. Recherchez votre bibliothèque.
2. Cliquez la bibliothèque recherchée, « installer » apparaît.
3. Cliquez sur « installer »
4. Votre bibliothèque apparaît comme installée (il faut probablement redémarrer Arduino).

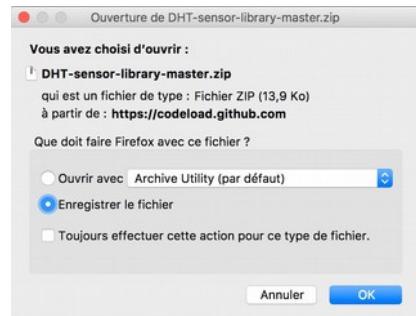


La seconde méthode :

Cliquez sur le lien donné plus haut pour télécharger la bibliothèque. Par exemple la bibliothèque qui permet de gérer les capteurs DHT.

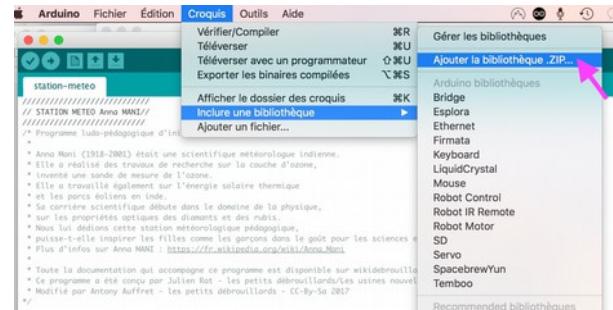
<https://github.com/adafruit/DHT-sensor-library/archive/master.zip>

Votre navigateur vous propose de l'enregistrer, vous le faites, sur votre bureau par exemple.

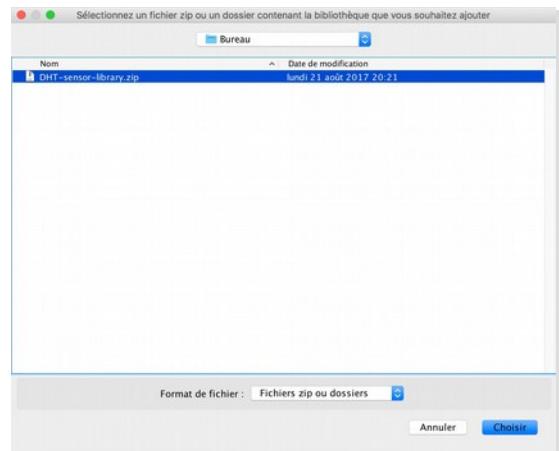


Ensuite, vous allez ajouter cette bibliothèque dans le logiciel Arduino.

Cliquez sur Croquis → inclure une bibliothèque
→ Ajouter la bibliothèque.ZIP



Une fenêtre de recherche s'ouvre.



Sélectionnez la bibliothèque, cliquez sur Choisir

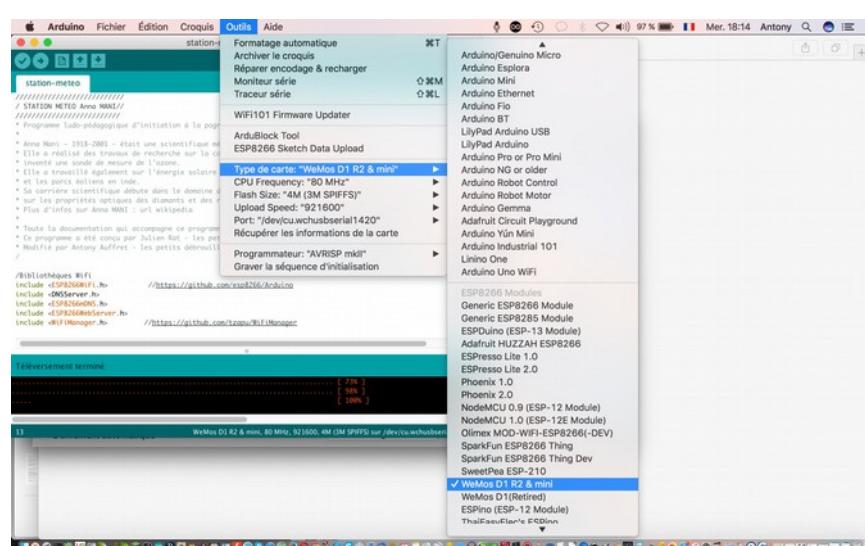
C'est fait !

6.4.2- Télécharger le programme sur le Wemos

Ouvrez votre logiciel Arduino

Branchez le Wemos.

Choisissez le bon type de carte.



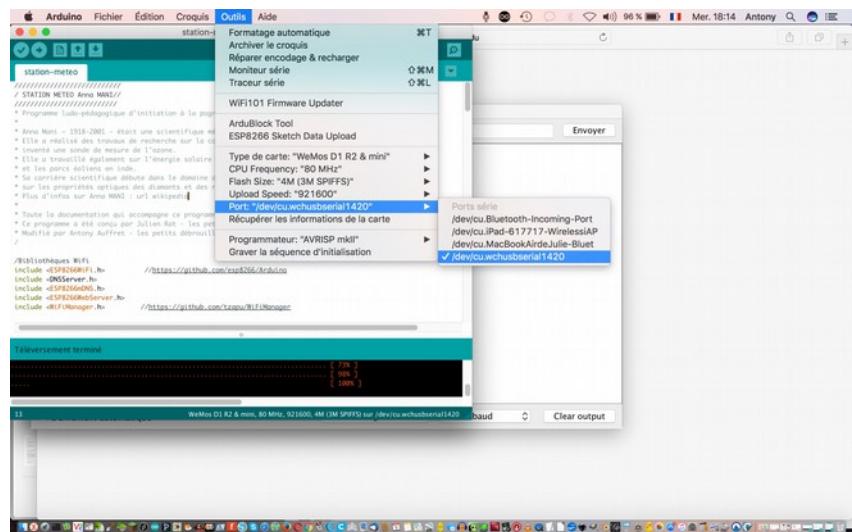
ATTENTION !

Il faut que vous ayez configuré le logiciel Arduino pour qu'il puisse gérer les cartes Wemos (vu en annexe).

Choisissez le port.

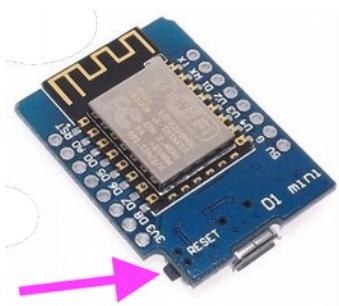
Si le port du Wemos n'apparaît pas, il y a plusieurs causes possibles :

- Les pilotes ne sont pas installé.
- Le câble utilisé n'est pas bon (il existe des câbles USB-Micro USB qui ne font pas circuler les données (ceux des chargeurs simples par exemple)).

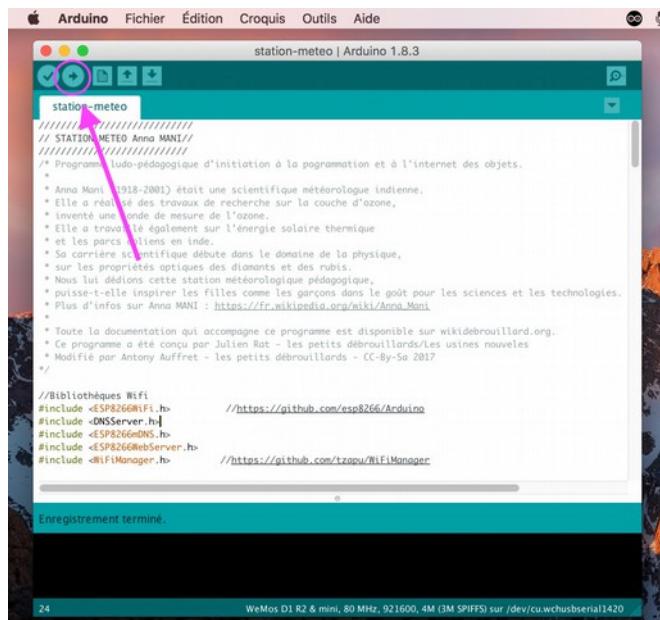


Téléversez le programme « Station-météo » dans votre Wemos.

Si le téléchargement ne fonctionne pas essayez d'appuyer sur le bouton « reset » du Wemos.



Il est maintenant prêt à servir.



6.5- Utiliser IFTTT

Le site web ifttt.com propose un service très simple de bidouille numérique. Un service web (site, objet connecté, autre) sert de « déclencheur » (Trigger) et cela provoque une « action » (telle action sur tel site web ou autre).

Si il se passe ça, alors ça déclenche cela !

if + this then that



Si Je reçois un mail attendu, alors je suis prévenu par SMS.

Si mon vidéaste préféré publie une vidéo, alors ça la poste sur Facebook.

Si la station spatiale internationale passe au-dessus de ma position, alors je reçois un SMS.

Si il pleut, alors mon compte twitter poste un twit de nuage.

Si mon frigo givre, alors envoie une cyberlettre à tous mes amis

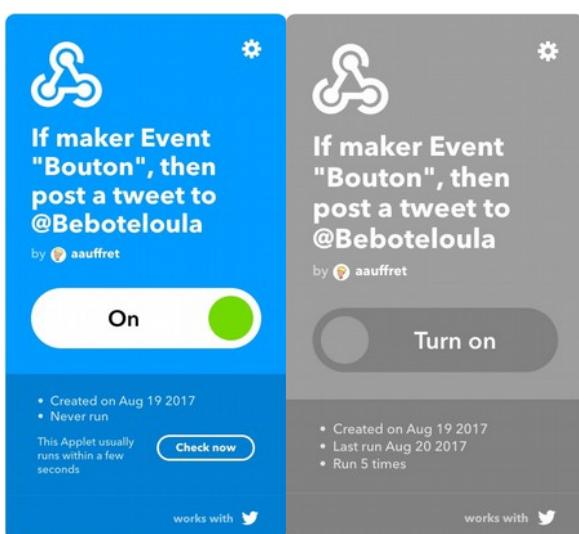
Du plus utile, au moins indispensable, mais de quoi s'amuser à inventer des robots logiciels interactifs !

Créez-vous un compte. Pour cela, il vous faut une adresse mail et retenir votre mot de passe !

Vocabulaire de IFTTT :

Vous y créez des « **RECETTES** » (**Recipe**) avec des « **DECLENCHEURS** » (**trigger**) qui lancent des « **ACTIONS** » (**Action** aussi en anglais).

Lorsque la recette est réalisée, c'est une « **Application** » (**Applet**), un mini logiciel, quoi !



L'application peut-être **activée (ON)** ou éteinte.

6.6- Le bouton qui twite avec IFTTT !

Ce que fera l'application que nous allons développer :

Nous souhaitons que lorsqu'on appuie sur le bouton de la station météo Anna MANI, cela poste un twit avec le taux d'humidité et l'heure.

La recette :		if this then that	
Déclencheur	appuyer sur le bouton de la station météo Anna MANI.		Webhooks (Les hameçons web) permet de recevoir un déclencheur avec 3 valeurs max (l'humidité et l'heure dans notre cas). Le déclencheur est un simple appel d'URL
Action	poster un twit avec le taux d'humidité et l'heure.		Indispensable pour pouvoir twitter ! Il vous faudra donc un compte twitter.

Après l'inscription sur le site, cliquez sur votre **nom**, puis sur « **New Applet** ». L'écran suivant apparaît.

On choisit le service « **Webhooks** » (anciennement appelé « **Maker** »).



Choose a service
Step 1 of 6

Q web

 Webhooks

On choisit ensuite le « déclencheur », facile ici il n'y en a qu'un : « **Receive a web request** » - recevoir une requête web.

Choose trigger
Step 2 of 6

 Receive a web request

This trigger fires every time the user makes a POST or a web request to notify it of an event. For information on triggering events, go to your webhook settings and then the listed URL (web) or tap your username (mobile).

On donne un nom à notre déclencheur (compléter le « champ » (field) « Event Name » – nom de l'événement).

Complete trigger fields
Step 2 of 6

Event Name
Bouton

The name of the event, like "button_pressed" or "front_door_opened"

Create trigger

on clique sur « **create trigger** » pour créer notre déclencheur « **Bouton** ».

ça y est, la première partie est créée, nous avons fait le « This » de « If This Then That ».

Cliquez sur « That ».

Maintenant, recherchez le service « Twitter », puis cliquez sur l'icône twitter.

Il faut connecter twitter à IFTTT, vous allez devoir entrer vos login et mot de passe twitter et donner votre autorisation.



Plusieurs actions vous sont proposées, on va choisir « Post a tweet » (poster un twit) !

Comme pour le déclencheur, vous devez compléter les champs →

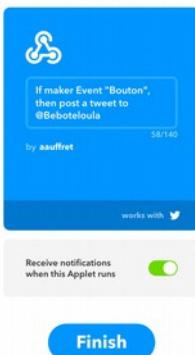
Ajouter des valeurs par exemple.

Et voilà le travail : Avec **EventName** = Bouton (C'est le nom que nous avons donné à notre déclencheur Webhooks plus tôt), **value1** qui sera la valeur de l'humidité et **value2** qui sera l'heure.

Le texte du twit est personnalisé, il ne reste plus qu'à cliquer sur « Create action ». →

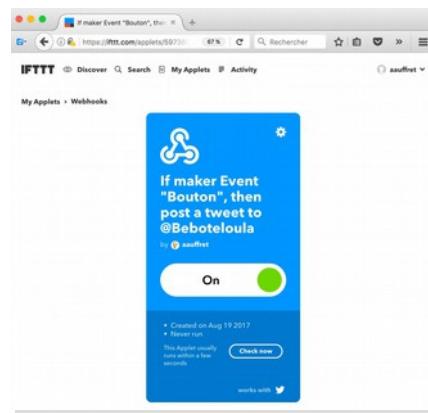
Review and finish

Step 6 of 6

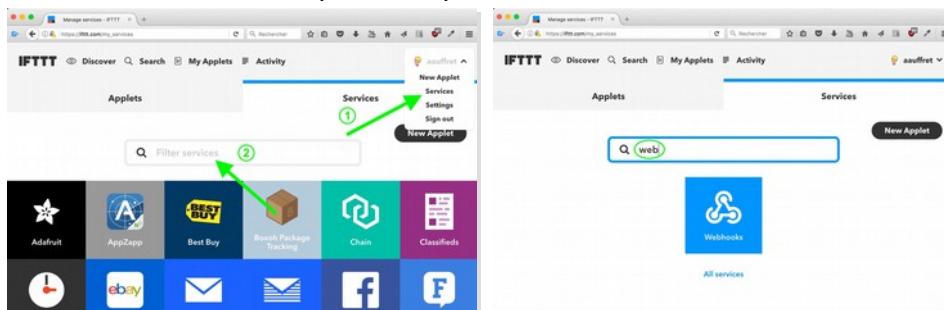


On passe en revue et on clique « **Finish** ».

Il ne reste qu'à récupérer les informations de « **Webhooks** » pour les utiliser dans **SNAP!**

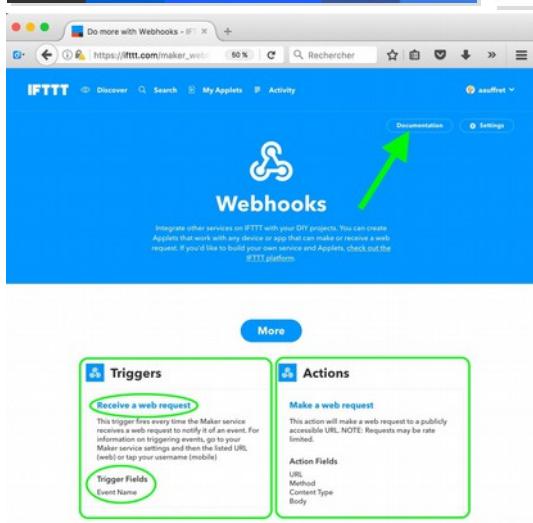


Voici la marche à suivre pour récupérer les informations de « **Webhooks** » :

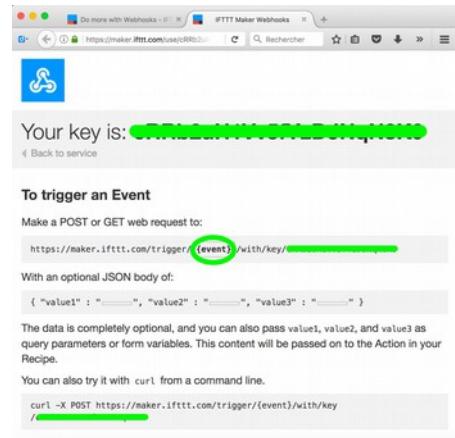


Cliquez sur **votre nom** (en haut à droite), puis sur « **services** » 1.

Dans le recherche, cherchez « **Webhooks** » 2, puis cliquez sur l'icône bleue de « **Webhooks** »



Cette page vous explique le fonctionnement de « **Webhooks** ». Pour avoir toutes les infos nécessaires au projet, jetez un œil à la **documentation** :



Vous y récupérez plusieurs infos utiles :

- Votre **clé (Your Key)**.
- L'**URL** à laquelle faire la requête et comment l'utiliser :
<https://maker.ifttt.com/trigger/{event}/with/key/XXX-votre-clé-XXX>

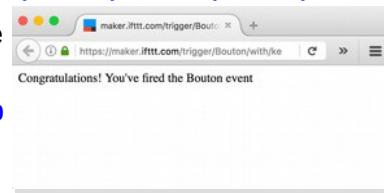
Pour ajouter des valeurs, cela se fait de cette manière :

<https://maker.ifttt.com/trigger/{event}/with/key/XXX-votre-clé-XXX?value1={VALUE1}&value2={VALUE2}&value3={VALUE3}>

Par exemple vous pouvez taper dans un navigateur directement une URL pour tester :

<https://maker.ifttt.com/trigger/Bouton/with/key/XXX-votre-clé-XXX?value1=75&value2=12h10>

Vous recevez même les félicitations de **Webhooks** !



Passons au code de SNAP!

Pour faire l'**heure**, nous créons la **variable heure** dans laquelle nous allons mettre l'heure sous le format suivant : 12h20mn. Il y a tout ce qu'il faut dans **SNAP!** pour cela.



Pour récupérer l'**état du bouton** et l'**humidité** c'est classique (Attention ! Remplacez les IP par les vôtres).



Pour construire l'URL de la requête à faire à IFTTT, j'ai choisi de créer plusieurs variables :

- L'URL de IFTTT : **IFTTT**
- Le nom du déclencheur : **déclencheur**.
- La clé « Webhooks » : **clé-Webhooks**.
- Les valeurs : **value1, value2, value3**.



Pour l'appeler, on affecte cette URL à la variable **Requête**.



On construit la boucle conditionnelle du bouton avec différents éléments :



- La boucle « si ».
- La comparaison de l'état du bouton1.
- La requête qui sera lancée.



Et voici le résultat :



Il ne reste plus qu'à tout assembler !

Quand  est pressé

Récupération des données de la station et de l'heure



Définition des éléments qui constitueront l'URL



Boucle conditionnelle : en cas d'appui, l'URL est appelée.



Et n'oubliez pas de tout répéter indéfiniment !

Le programme final : et ça twite !

The image shows a Scratch script and a screenshot of a Twitter profile. The Scratch script (left) is titled "Quand bouton1 est pressé" and has a "répéter indéfiniment" loop. It triggers an IFTTT webhook with URL `http://192.168.0.23/bp1` and sends the values of "humidité" and "heure". The IFTTT webhook then triggers another IFTTT webhook with URL `maker.ifttt.com/trigger/`. The final action is to tweet the message "Quelqu'un a appuyé sur le Bouton ! L'humidité est de %5 hum et l'heure est %heure". The Twitter screenshot (right) shows the profile of "Antoine de st-exhube" (@Beboteloula) with 35 tweets, 179 followers, and 14 following. A green circle highlights the tweet from 51 seconds ago: "Quelqu'un a appuyé sur le Bouton ! L'humidité est de 75% à 11h43mn".

Amusez-vous bien avec IFTTT !

if + this then that



6.7- Ressources humaines et documentaires

Julien Rat, Les petits débrouillards & Les usines nouvelles.
Mail : j.rat@lespetitsdebrouillards.org

Antony Auffret, Les petits débrouillards & Université Rennes 2.
Mail : aauffret@lespetitsdebrouillards.org



<http://www.lespetitsdebrouillards.org>



<http://www.lesusinesnouvelles.com>

Les images de ce document ont été réalisées et/ou traitées avec des logiciels libres :

- Rédaction sous OpenOffice
- Dessin vectoriel sous Inkscape
- Traitement des images sous GIMP



Le programme du Wemos D1 mini à été réalisé avec Arduino.

Les schémas de brochage avec Fritzing.

Les programmes ludo-pédagogiques avec SNAP!



6.8- L'aide mémoire de tous les blocs SNAP!



Aide-mémoire de tous les blocs de SNAP!

Snap! est un logiciel de programmation visuelle par glisser-déposer de blocs d'instructions inspiré de SCRATCH du MIT. Il vous permet de créer vos propres blocs et possède des fonctionnalités supplémentaires. Il est porté par l'université de Californie à Berkeley.

<http://snap.berkeley.edu>

Sites miroirs :
<http://bjc.edc.org/snapsource/snap.html>
<https://snap.apps.miosoft.com/>
<http://media.mit.edu/~harveyb/snap>
<http://cs10.org/snap>

Document réalisé par Anthy Auffret
 Les petits débrouillards
 sous licence CC-By-SA

Le document aide-mémoire de tous les blocs de SNAP! est divisé en deux parties principales :

- Top Row:** Affiche les catégories de blocs : Mouvement, Contrôles, Apparence, Capteurs, Sons, Opérateurs, Stylo, Variables.
- Bottom Row:** Affiche les blocs détaillés pour chaque catégorie.

Mouvement :

- avancer de 10 pas
- tourner de 15 degrés ⌂
- tourner de 15 degrés ⌄
- se diriger en faisant un angle de 90°
- se diriger vers []
- aller à x: 0 y: 0
- aller à []
- glisser en 1 sec. à x: 0 y: 0
- ajouter 10 à x
- donner la valeur 0 à x
- ajouter 10 à y
- donner la valeur 0 à y
- rebondir si le bord est atteint
- [] position x
- [] position y
- [] direction

Contrôles :

- basculer sur le costume ▾
- costume suivant
- [] costume n°
- dire Salut! pendant 2 sec.
- dire Salut!
- penser Mmmh... pendant 2 sec.
- penser Mmmh...
- ajouter à l'effet transparence 25
- mettre l'effet transparence à 0
- annuler les effets graphiques
- ajouter 10 à la taille
- choisir 100 % de la taille initiale
- [] taille
- montrer
- cacher
- envoyer au premier plan
- déplacer de 1 plan arrière

Apparence :

- jouer le son ▾
- jouer le son ▾ jusqu'au bout
- arrêter tous les sons
- faire une pause pour 0.2 temps
- jouer la note 60 pour 0.5 temps
- ajouter 20 au tempo
- choisir le tempo à 60 bpm
- [] tempo

Capteurs :

- effacer tout
- stylo en position d'écriture
- relever le stylo
- mettre la couleur [] pour le stylo
- ajouter 10 à la couleur du stylo
- choisir la couleur 0 pour le stylo
- ajouter 10 à l'intensité du stylo
- choisir l'intensité 100 pour le stylo
- ajouter 1 à la taille du stylo
- choisir la taille 1 pour le stylo
- estampiller
- remplir

Sons :

- [] Nouvelle variable
- [] prend la valeur 0
- ajouter à [] 1
- afficher la variable []
- cacher la variable []
- variables du script []
- [] liste []
- [] au début de []
- élément 1 de []
- tous sauf le premier de []
- longueur de []
- [] contient qqchose
- ajouter qqchose à []
- supprimer l'élément 1 de []
- insérer qqchose en position 1 de []
- remplacer l'élément 1 de [] par qqchose
- Nouveau bloc

Opérateurs :

- [] < []
- [] = []
- [] > []
- [] et []
- [] ou []
- [] non []
- [] vrai []
- [] regroupe Bonjour Monde []
- [] découpe Bonjour-Monde entre les []
- [] lettre 1 de Monde
- [] longueur de Monde
- [] valeur unicode de a
- [] unicode 65 comme lettre
- [] 5 est un(e) nombre [] ?
- [] est identique à [] ?
- [] fonction JavaScript ([]) { [] }

Stylo :

- [] attendre 1 sec.
- [] attendre jusqu'à []
- [] répéter indéfiniment
- [] Quand [] est pressé
- [] Quand espace est pressé
- [] Quand je suis cliqué
- [] Quand []
- [] Quand je reçois []
- [] envoyer à tous
- [] envoyer à tous et attendre
- [] message
- [] Englobe
- [] attendre 1 sec.
- [] attendre jusqu'à []
- [] répéter indéfiniment
- [] Quand je commence comme clone
- [] Clone
- [] supprime ce clone
- [] mette en pause []
- [] répéter 10 fois
- [] répéter jusqu'à []
- [] si []
- [] si []
- [] sinon
- [] rapporte []
- [] arrête tout
- [] arrête tout sauf ce script
- [] exécute []
- [] lance []
- [] appelle []
- [] exécute [] avec continuation
- [] appelle [] avec continuation
- [] et []
- [] ou []
- [] non []
- [] vrai []
- [] distance de []
- [] réinitialiser le chronomètre
- [] chronomètre
- [] costume n° de []
- [] attribut neighbors []
- [] http://snap.berkeley.edu
- [] turbo mode activé ?
- [] turbo mode prend la valeur []
- [] date courante jour []
- [] et []
- [] ou []
- [] non []
- [] vrai []
- [] regroupe Bonjour Monde []
- [] découpe Bonjour-Monde entre les []
- [] lettre 1 de Monde
- [] longueur de Monde
- [] valeur unicode de a
- [] unicode 65 comme lettre
- [] 5 est un(e) nombre [] ?
- [] est identique à [] ?
- [] fonction JavaScript ([]) { [] }

Variables :

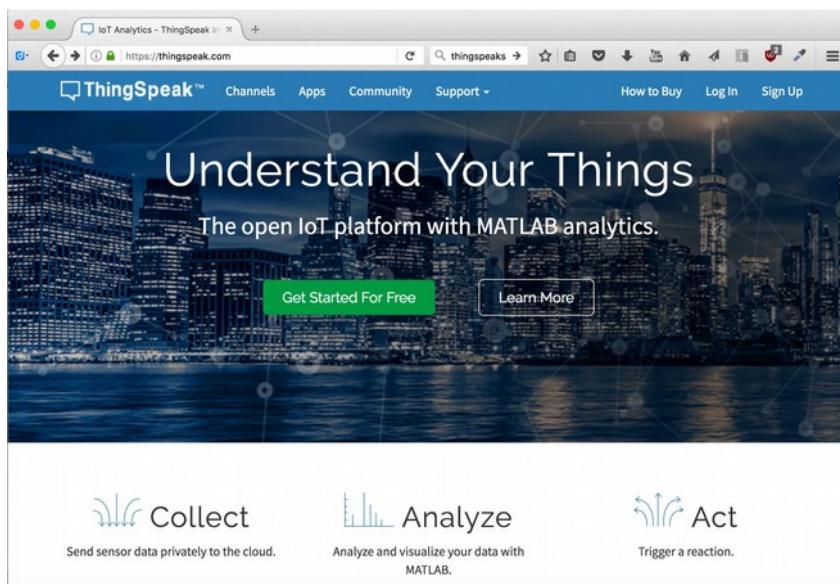
- [] Nouvelle variable
- [] prend la valeur 0
- ajouter à [] 1
- afficher la variable []
- cacher la variable []
- variables du script []
- [] liste []
- [] au début de []
- élément 1 de []
- tous sauf le premier de []
- longueur de []
- [] contient qqchose
- ajouter qqchose à []
- supprimer l'élément 1 de []
- insérer qqchose en position 1 de []
- remplacer l'élément 1 de [] par qqchose
- Nouveau bloc



6.9- Thingspeak présentations : Un site web pour l'internet des objets

Selon les propres termes des développeurs de Thingspeak, « c'est un logiciel open-source et une API (API est l'acronyme de **Application Programming Interface** – Interface de programmation – C'est l'interface qui sert à connecter un logiciel à des services), pour l'internet des objets. ». C'est à la fois un logiciel que chacun peut installer et un site web.
Il sert à collecter, stocker, mettre en forme et analyser des données.
Il peut également servir à déclencher des actions en fonction des données qui lui sont transmises.
Il utilise le protocole « **http** » ce qui en fait un service relativement simple à utiliser.

Thingspeak sert donc à :



The screenshot shows the ThingSpeak homepage. At the top, there's a navigation bar with links for "Channels", "Apps", "Community", "Support", "How to Buy", "Log In", and "Sign Up". The main banner features a city skyline at night with the text "Understand Your Things" and "The open IoT platform with MATLAB analytics." Below the banner are three main sections: "Collect" (Send sensor data privately to the cloud), "Analyze" (Analyze and visualize your data with MATLAB), and "Act" (Trigger a reaction).

- **Collect** : collecter des données de capteurs, en toute sécurité sur les serveurs de

Thingspeak.

- **Analyse** : analyser et visualiser ces données.
- **Act** : déclencher des actions.

Dans ce tutoriel nous allons nous intéresser à la collecte et l'affichage des données.

Avez-vous remarqué que « **Thing Speak** » signifie « **chose parle** »

6.10- Comment ça marche ?

Passons par une métaphore : Imaginez un groupe de personnes. Ils font partie du club des météophiles de l'île de Ouessant.

Ce sont Géraldine, Gontran, Prisca et Odilon qui assurent les relevés et donnent ces relevés à Viviane qui est chargée de l'écriture des données.

Viviane écrit chaque jour des relevés météo, dans des tableaux de suivi (un tableau pour la température, un pour l'hygrométrie, etc.).

En résumé :

- Géraldine donne la Température à Viviane qui l'inscrit dans le tableau des températures.
- Gontran donne l'Hygrométrie pour le tableau de suivi de l'Hygrométrie

- Prisca donne la Luminosité pour le tableau adapté
- Odilon donne la Pression etc.

C'est comme cela que fonctionne Thingpeak !

Ce que c'est	Pour Notre groupe	Pour Thingspeak
L'adresse	Le club de météophiles de l'île d'Ouessant	Http://www.thinspeak.com
La mission de mise à jour des données	Géraldine, Gontran, Prisca et Odilon	update
La clé pour l'écriture des relevés météo	Viviane	La clé d'API pour l'écriture
Les tableaux	Température, Pression, ...	Field1, Field2,...

La mise à jour d'un tableau se fait comme ceci :

Au club de météophiles de l'île d'Ouessant	Géraldine	Demande à Viviane	D'écrire dans le tableau des températures la température 22°C
https://api.thingspeak.com	/update	?api_key=XXXXXXXXXXXXXX	&field1=22

Pour mettre à jour un tableau sur Thingspeak, il suffit juste d'appeler l'URL concernée. Faites le test ici, je vous propose un bac à sable : cliquez sur cette URL (ou copiez-collez-là dans la barre d'adresse d'un navigateur).

https://api.thingspeak.com/update?api_key=GBAJRPBIS940YBL6&field1=22

Jouez en changeant la valeur qu'il y a après « **field1=** ».

Puis allez voir la chaîne qui reçoit l'info : <https://thingspeak.com/channels/320552>

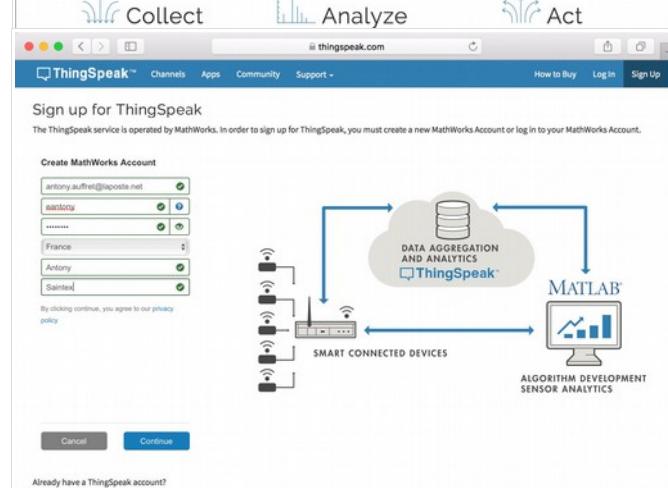


6.11- Comment créer et alimenter sa chaîne en données ?

6.11.1- Créez un compte

Sur la page d'accueil, cliquez en haut à droite sur « Sign up ».

Thingspeak est un service porté par MathWorks. Le compte que vous créez est donc un compte Mathwork.

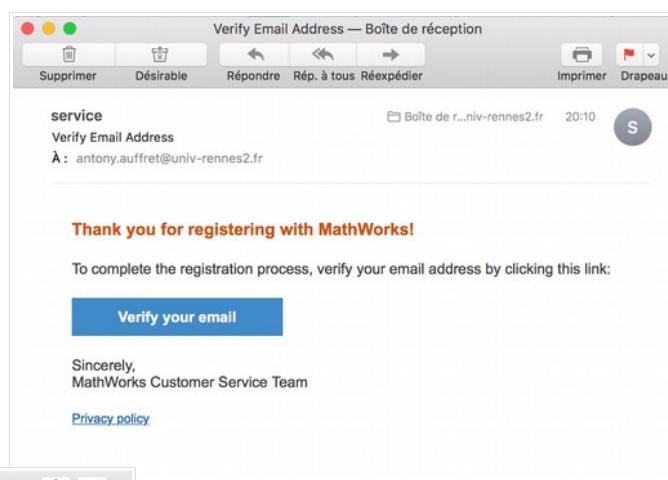


C'est un remplissage classique :

- Votre e-mail
- Votre pseudo
- Votre mot de passe
- Votre nom
- Votre prénom

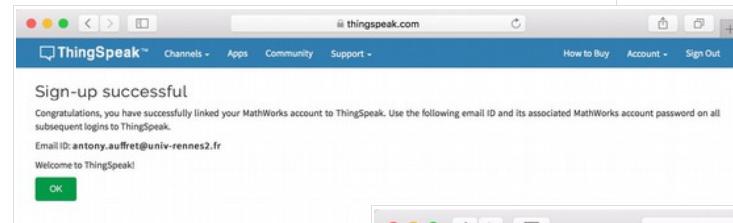
Cliquez sur « Continue »

Vous allez recevoir un e-mail de MathWork quelques minutes après la création de votre compte. Il faut cliquer sur le lien du mail pour finaliser la création de votre compte.

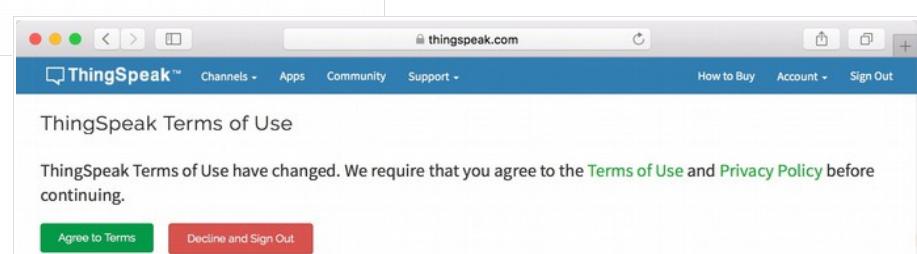


Après cela, retournez sur le site **Thingspeak** et cliquez en haut à droite sur « Log In ».

Vous avez réussis votre création de compte !



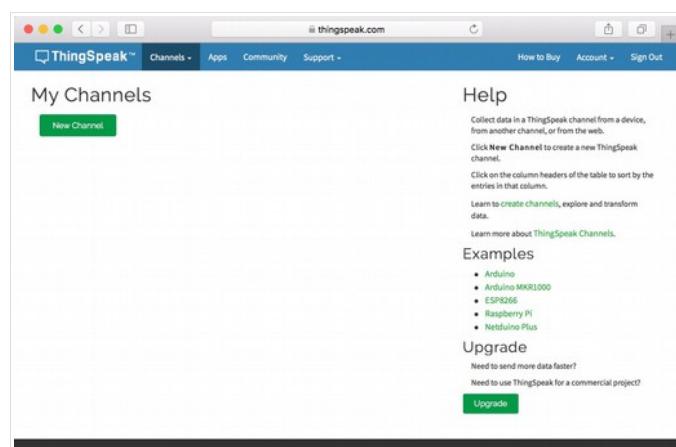
Il n'y a plus qu'à accepter les conditions d'utilisation « Agree to Terms ».



6.11.2- Créez votre chaîne

Cliquez, en haut, sur « **Channels** » puis « **My Channel** ».

Cliquez sur « **New Channel** »



Un formulaire s'ouvre. Il ne vous reste plus qu'à remplir les différents champs.

Name : pour le nom de votre chaîne

Description : c'est ce qui sera affiché sur votre chaîne

Field 1 : le premier tableau qui recevra vos données.

Field 2 : etc.

Metadata : métadonnées d'informations sur votre chaîne.

Tags : des mots-clés pour identifier votre chaîne.

Link to external Site : un lien vers votre site.

Show location : pour afficher la géolocalisation de votre chaîne avec la latitude et longitude.

Show Video : pour inclure une vidéo YouTube ou Vimeo.

Cliquez sur **Save Channel** C'est prêt !

Exemple de configuration pour la station météo :

6.11.3- Comprendre votre chaîne

Private view : pour visualiser vos données en privé

Public view : la vue publique de votre chaîne si c'est activé dans « **Sharing** »

Channel Settings : c'est le retour sur le formulaire de création de votre chaîne pour ajuster la configuration

Sharing : pour activer le partage soit avec des contact mail précis, soit en public.

APIKey : C'est la clé que vous trouvez les infos pour mettre à jour votre chaîne.

Data Import/export : pour exporter ou importer des données.

6.11.4- Configurer le programme Station Météo Anna MANI

Dans l'onglet « **API Keys** » copiez la clé « **Write API Key** » et collez-là à la place indiquée dans le programme Arduino.

```

station-meteo-anna-mani-v2 | Arduino 1.8.3

//Publication sur Thingspeak //
String apiKey = "XXXXXXXXXXXXXX"; //c'est ici qu'on place la clé d'API de Thingspeak
const char* serverpub = "api.thingspeak.com"; //adresse du serveur thingspeak pour publier
WiFiClient client; //démarrage du client WiFi
int tempo = 0; //création de la variable tempo, //servira à temporiser l'envoie de donnée à Thingspeak

//////////Broches/////////
const int led1 = D8;
const int led2 = D7;
const int bp1 = D6;
const int bp2 = D5;

void setup() {
  //Prepare la broche GPIO2 (marquée D4 sur le Wemos)
  pinMode(2, OUTPUT);
  digitalWrite(2, 1);

  Serial.begin(9600); // démarrage de la connexion série
}

Enregistrement terminé.

[ 73% ]
[ 97% ]
[ 100% ]

```

6.11.5- Au sujet de l'onglet « API Keys »

C'est grâce à elle qu'on pouvoir envoyer et recevoir des données avec le site Thinspeak.

Aide

Les clés d'API (API signifie Application Programming Interface – Interface de programmation – C'est l'interface qui sert à connecter un logiciel à des services) vous permettent d'écrire des données sur une chaîne ou de lire des données à partir d'une chaîne privée. Les clés d'API sont générées automatiquement lorsque vous créez une nouvelle chaîne.

Les clés de l'API :

Write API Key - Clé d'API pour l'écriture : utilisez cette clé pour écrire des données sur une chaîne. Si vous pensez que votre clé a été compromise, cliquez sur Générer une nouvelle clé d'écriture.

Read API Keys - Clés d'API pour la lecture : utilisez cette clé pour permettre à d'autres personnes d'afficher les flux et les graphiques de votre chaîne privé. Cliquez sur « **Générer une nouvelle clé d'API de lecture** » pour générer une clé de lecture supplémentaire pour la chaîne.

Note : Utilisez ce champ pour entrer des informations sur les clés de lecture de canal. Par exemple, ajoutez des notes pour suivre l'accès des utilisateurs à votre chaîne.

Utilisation de la clé d'API pour l'écriture sur notre chaine :

Update Channel Feed – Mise à jour de la chaine. C'est une URL qui met à jour chaque champ de la chaine.

https://api.thingspeak.com/update?api_key=XXXXXXXXXXXXXXXXXX&field1=0

URL Thingspeak pour la mise à jour	Clé d'API	Champ à mettre à jour avec la donnée
https://api.thingspeak.com/update	?api_key=XXXXXXXXXXXXXX	&field1=0
Cette URL est commune à toute les chaines.	La clé d'API permet à Thingspeak de reconnaître votre chaine. C'est un identifiant	Ici c'est 0 qui est envoyé

Pour en savoir plus :

<https://de.mathworks.com/help/thingspeak/update-channel-feed.html>

Crédits :

Image de routeur wifi : sous licence FAL - Free Art Licence – Auteur VASCER -
<https://fr.wikipedia.org/wiki/Routeur#/media/File:Routeur-wifi.jpg>

Image de Ouessant – Phare du Stiff : CC-By-Sa – Guy Bernard
https://commons.wikimedia.org/wiki/File:Phare_du_Stiff-Ouessant.JPG

Image d'ordinateur portable : CC0.

SNAP! : Logo, blocs, copie d'écran propriété de SNAP! - Université de Californie à Berkeley sous licence libre AGPL.

Photo des composants : CC0.

Pictogrammes des composants : Fritzing sous licence CC-By-Sa.

Iconographie pour les programmes : <https://openclipart.org/> sous licence CC0

Photo de Wemos : Wemos

Copies d'écrans : CC0

Des copies sont tirées des logiciels Arduino et du site web Thingspeak.

