

Adafruit Feather esp8266

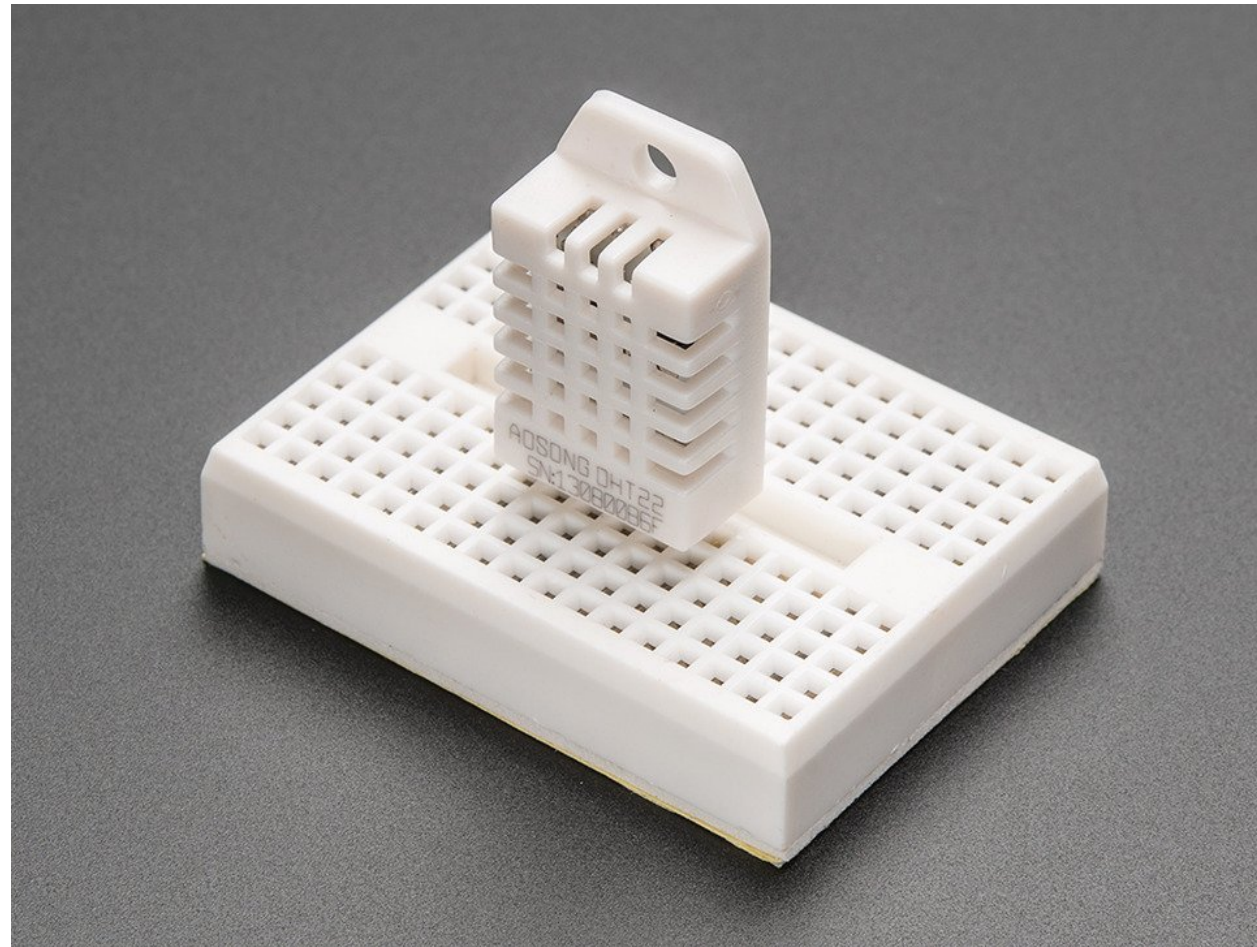
DHT22

Capteur d'humidité et de température

Matériel

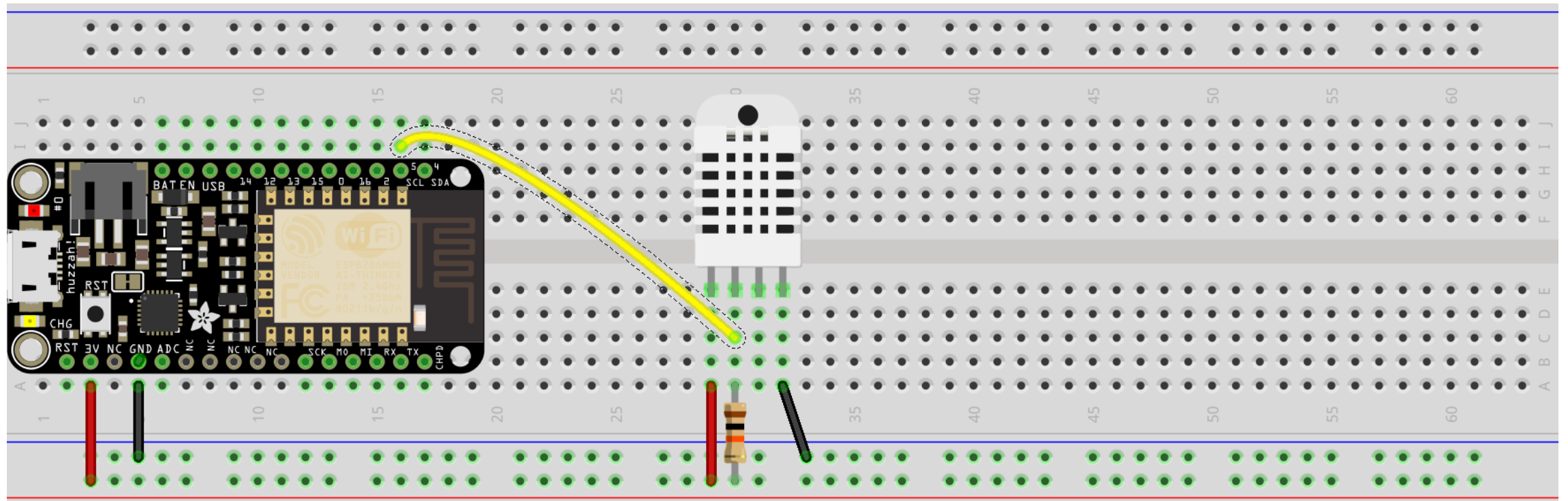
- Adafruit feather Esp8266
- DHT 22

DHT 22



- Doc : <https://learn.adafruit.com/dht>

Connecter le capteur



Résistance de pull-up (10k) entre data et 3.3v

fritzing

Bibliothèques

- https://github.com/adafruit/Adafruit_Sensor
- <https://github.com/adafruit/DHT-sensor-library>

Installation d'une bibliothèque dans l'IDE Arduino

- Ouvrir le gestionnaire de bibliothèques :

Croquis > inclure une bibliothèque > Gérer les bibliothèques



Code de test

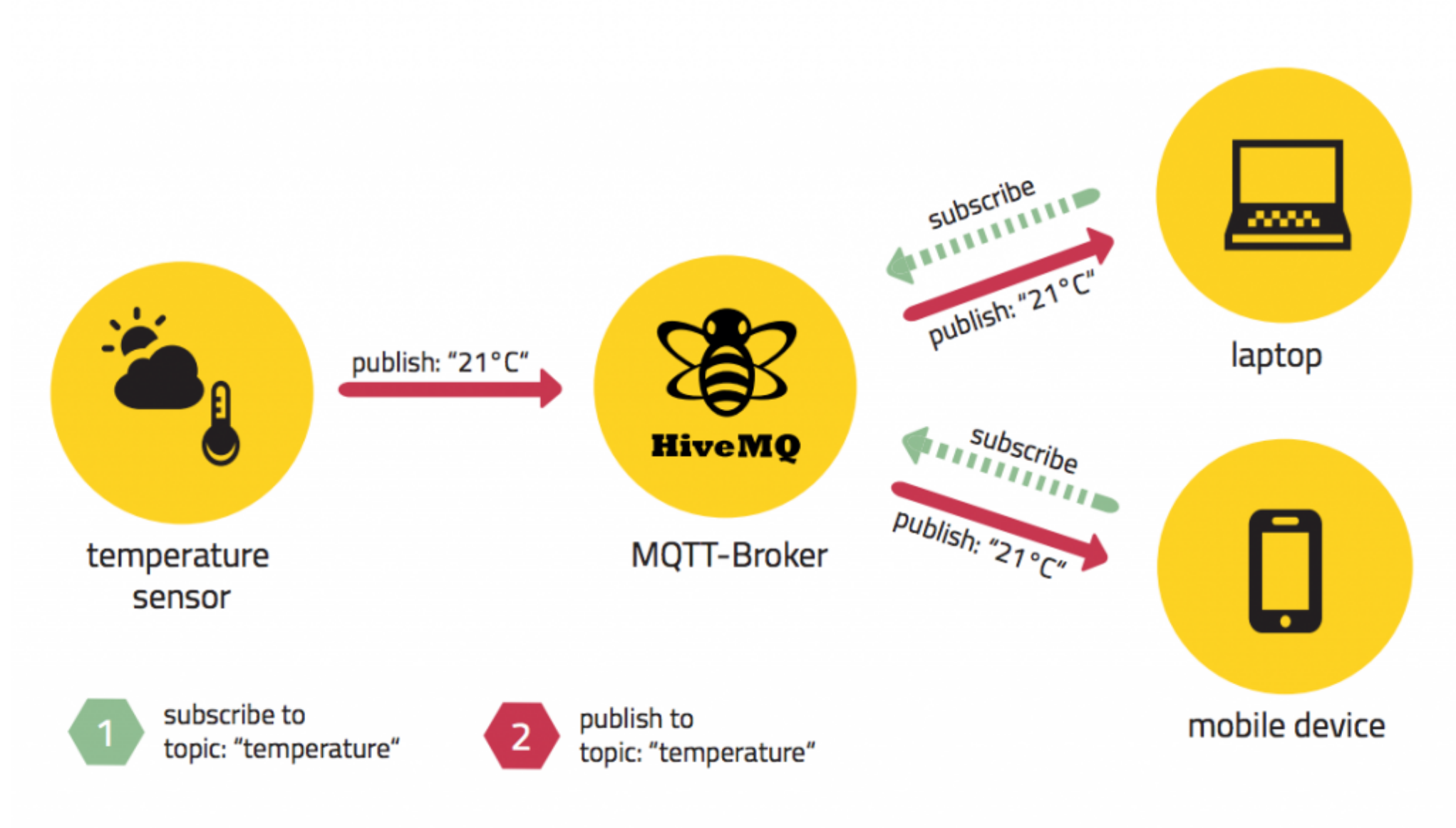
V. exemple : DHT22 test

Internet et l'IOT

- Le protocole HTTP
 - Les API REST (exemple : openweathermap.org)
 - Le format JSON
- Le protocole MQTT
 - Le Pattern Publish/Subscribe (exemple sur la plateforme adafruit.io)

Internet et l'IOT

Le Pattern Publish/Subscribe



Connexion à un broker Mqtt

- Créer un compte sur adafruit.io

adafruit.io

- **Présentation**
- **concepts de base: feeds, dashboards**
- **créer un compte**
 - **créer groupe, feeds et dashboard**
 - **recupérer la clé API**
- **installer la librairie arduino**
- **se connecter à la plate-forme**
- **communiquer avec la plateforme**

adafruit.io

Présentation

- Plateforme cloud d'agrégation de données
- Destinée aux objets connectés (la librairie est disponible pour plusieurs plateformes : la gamme Adafruit, Arduino, Raspberry Pi...)
- Permet de se connecter à d'autres services web (IFTTT, Zappier)
- Gratuit pour une utilisation limitée
- Http (REST) et Mqtt

adafruit.io

Concepts

- Deux concepts principaux :
 - Les feeds (pouvant être groupés) : permettent de définir des “points d’accès” pour publier ou retrouver les données d’un capteur
 - Les dashboards : fournissent une interface graphique permettant de lire les données des feeds ou d’agir sur un actuateur

adafruit.io

Mise en oeuvre 1/4

Dans l'interface web

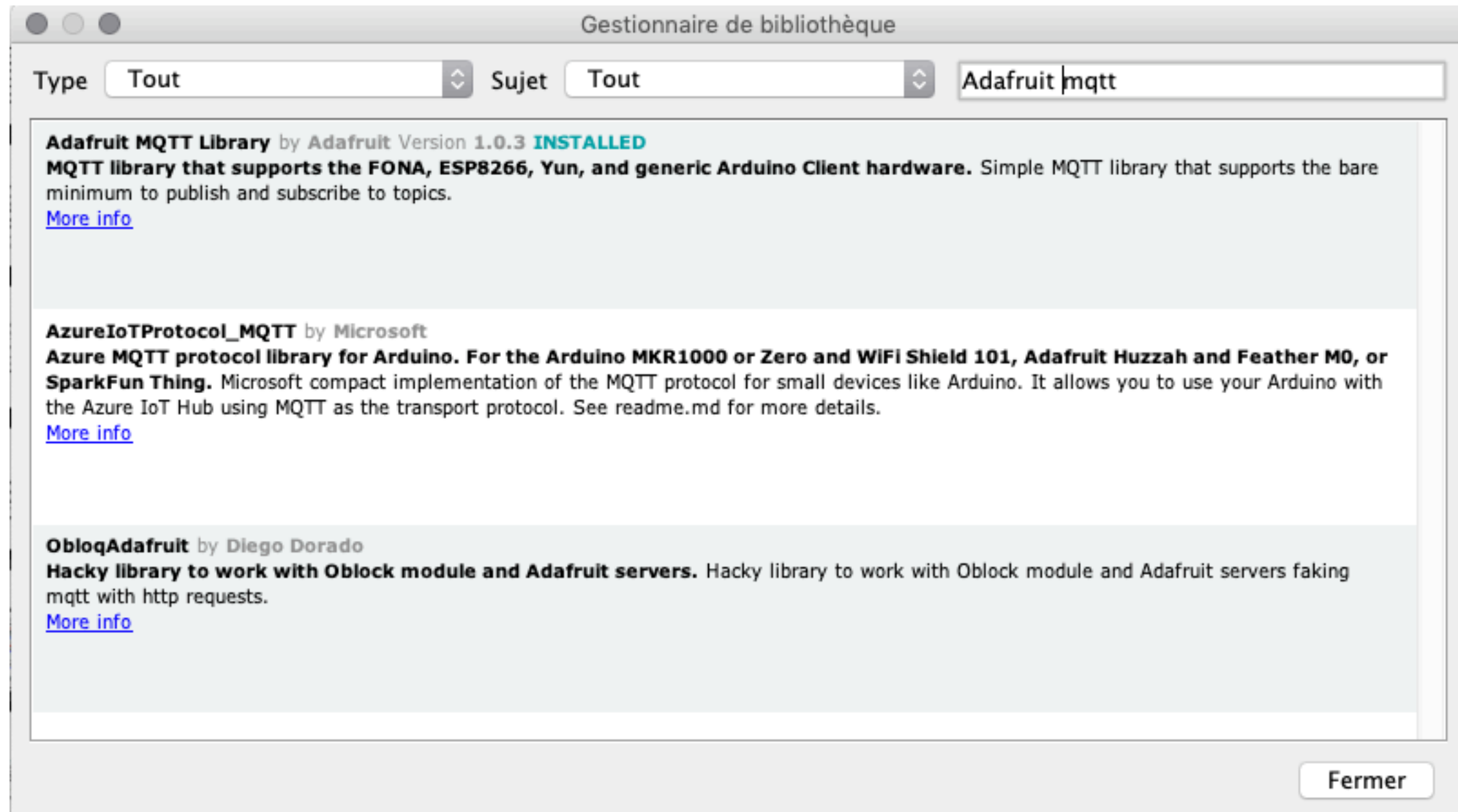
1. Créer un compte sur <https://adafruit.io>
2. Créer son premier feed
3. Créer un dashboard permettant d'afficher les données du feed
4. récupérer sa clé d'API (AIO key)

adafruit.io

Mise en oeuvre 2/4

Dans l'IDE Arduino

1. Installer la librairie **Arduino Adafruit MQTT**



adafruit.io

Mise en oeuvre 3/4

Dans l'IDE Arduino



The screenshot shows the Arduino IDE interface. The title bar at the top reads "mqtt_adafruit.io - settings.h | Arduino 1.8.9". Below the title bar is a toolbar with icons for saving, running, and other IDE functions. The main editor area displays the contents of the "settings.h" file, which is a header file for MQTT configuration. The code is as follows:

```
1 #define WLAN_SSID "ssid"
2 #define WLAN_PASS "pass"
3
4 #define IO_SERVER "io.adafruit.com"
5 #define IO_SERVERPORT 1883
6 #define IO_USERNAME "username"
7 #define IO_KEY "key"
```


adafruit.io

Mise en oeuvre 4/4

Dans l'IDE Arduino

1. Définir les différentes variables destinées à la connexion (dans le cas où on se base sur un exemple existant, ces variables peuvent se trouver dans le fichier settings.h)
2. Inclure la bibliothèque **ESP8266WiFi**
3. Initialiser la connexion en instantiant un objet

adafruit.io

Exemple: publier des données

Exemple: recevoir des données

adafruit.io

Exercice : publier les données issues du
DHT22