

Connecter Arduino à Internet

Utilisation de la carte MKR1000

Différents types de connectivité

- Ethernet
- WiFi
- GSM/Satellite
- Sigfox / LoRa
- Bluetooth / BLE

Internet

- Infrastructure physique
- Infrastructure logique : TCP/IP : Suite de protocoles permettant l'interconnection des machines en réseau
- Internet des objets :
 - Les objets du monde physique peuvent avoir leur existence propre sur le réseau (une adresse, la faculté d'envoyer et/ou recevoir des données)
 - agrégation des données collectées par des objets via des plateformes Cloud
- Cloud Computing permet l'exploitation de la puissance de calcul de machines distantes via le réseau

Internet et l'IOT

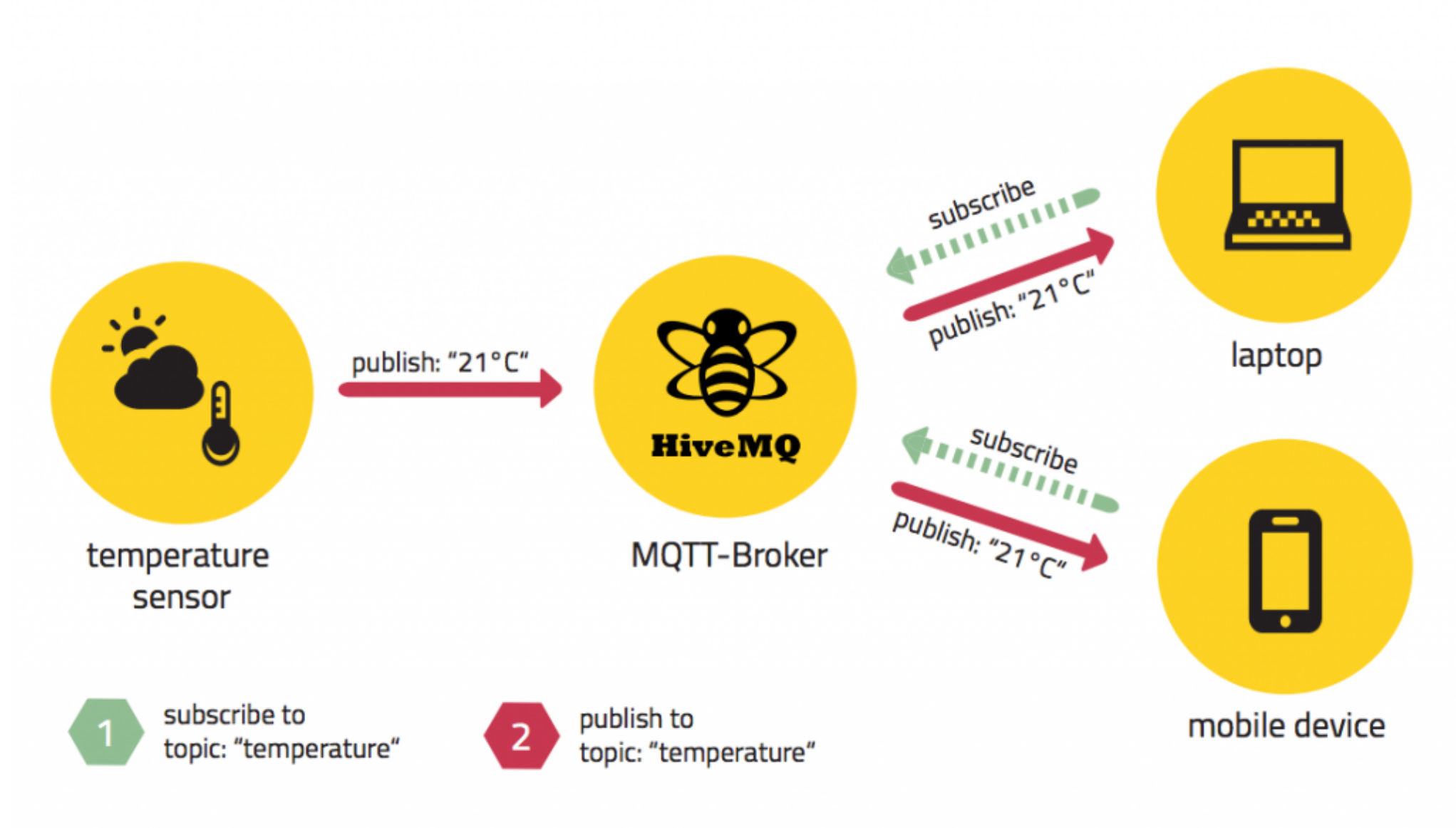
- Les plateformes Cloud
 - Aggrégateurs de données
 - Communication bi-directionnelle avec les devices
- Cas d'utilisation : adafruit.io

Internet et l'IOT

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

Internet et l'IOT

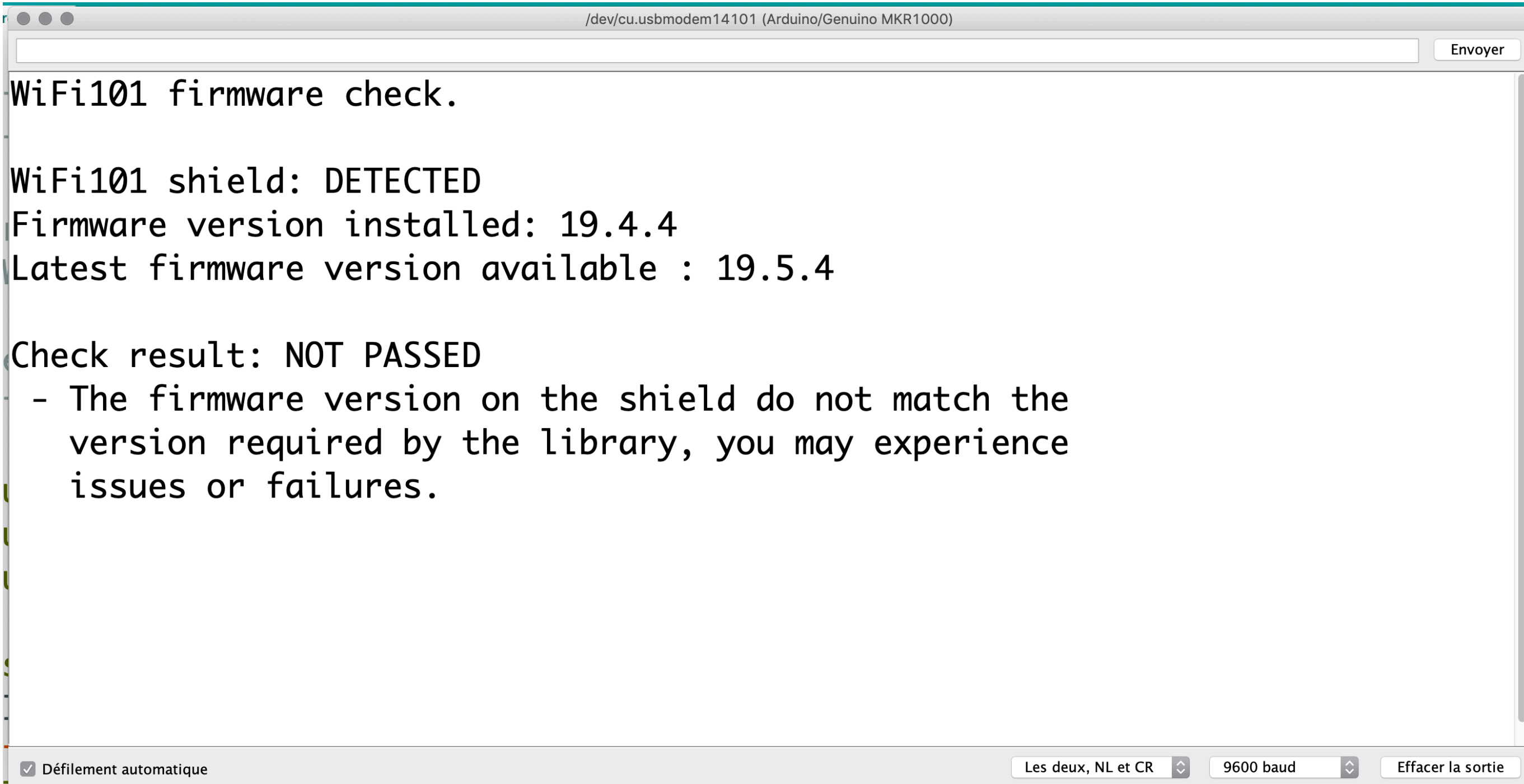
Le Pattern Publish/Subscribe



Connecter le MKR1000 au WiFi

- Installer La librairie WiFi101 via le gestionnaire de bibliothèques de l'IDE Arduino ou via son fichier .zip
- Vérifier que le firmware de la carte est bien la dernière version :
 - Fichier > Exemples > WiFi101 > CheckWiFi101FirmwareVersion

Connecter le MKR1000 au WiFi



The screenshot shows the Arduino IDE serial monitor window. The title bar indicates the connection is to `/dev/cu.usbmodem14101 (Arduino/Genuino MKR1000)`. The serial output text is as follows:

```
WiFi101 firmware check.  
  
WiFi101 shield: DETECTED  
Firmware version installed: 19.4.4  
Latest firmware version available : 19.5.4  
  
Check result: NOT PASSED  
- The firmware version on the shield do not match the  
  version required by the library, you may experience  
  issues or failures.
```

At the bottom of the window, there are several controls: a checkbox for "Défilement automatique" (checked), a dropdown menu for "Les deux, NL et CR", a dropdown menu for "9600 baud", and a button labeled "Effacer la sortie".

Connecter le MKR1000 au WiFi

- Au besoin, mettre à jour le firmware du mkr1000
 1. Charger le sketch FirmwareUpdater (Fichier > Exemples > WiFi101 > FirmwareUpdater)
 2. Utiliser l'outil WiFi101 Firmware Updater (Outils > WiFi101 Firmware Updater)

Installer les librairies

- `ArduinoHttpClient`
- `PubSubClient`

Exemple 1 : Publier des données sur dweet.io

Dweet

- <https://dweet.io> service de messagerie pour l'internet des objets
- Dweet => message
- Thing => identifiant de l'objet connecté

Exemple 1 : Publier des données sur dweet.io

Dweet

- Fonctionne en HTTP
- Publier une info => effectuer une requête vers une url <https://dweet.io/dweet/for/my-thing-name?hello=world>
- Récupérer une information (JSON)
 - derniers messages <https://dweet.io/get/dweets/for/my-thing-name>
 - dernier message <https://dweet.io/get/latest/dweet/for/my-thing-name>

Exemple 1 : Publier des données sur dweet.io

- Connecter un potentiomètre à la carte MKR1000
- Dupliquer le code dans

Exemples > ArduinoHttpClient > DweetPost

Exemple 1 : Publier des données sur dweet.io

- Dans le code d'exemple :
 - ajouter SSID et mot de passe du Wifi dans le fichier **arduino_secrets.h**
 - **ATTENTION** : dans la fonction **setup()**, la ligne
`while(!Serial);`
Attend que le moniteur série soit ouvert
 - ligne 60 : définir un nom unique pour la variable **dweetName**

Exemple 2 : Publier des données sur un broker MQTT

- Connecter un potentiomètre à la carte MKR1000
- Utiliser la librairie PubSubClient (voir exemple mqtt_pub_sub)
- Utiliser l'adresse IP de mon broker en local (mosquitto + Node Red)
 - xxx.xxx.xxx.xxx sur le port 1883

Exemple 2 : Publier des données sur un broker MQTT

Node Red : “Flow-based programming for the Internet of Things”

Exemple 2 : Publier des données sur un broker MQTT

The screenshot displays the Node-RED web interface. The top bar includes the Node-RED logo, a search bar, and buttons for 'Deploy', 'info', 'debug', and 'dashboard'. The left sidebar is divided into three sections: 'input', 'output', and 'function'. The 'input' section contains nodes like inject, catch, status, link, mqtt, http, websocket, tcp, and udp. The 'output' section contains debug, link, mqtt, http response, websocket, tcp, and udp. The 'function' section contains a function node. The main workspace shows a flow named 'Flow 1' with two parallel paths. The top path consists of a 'timestamp' node connected to an 'mqtt' node with the topic '/test'. The bottom path consists of an 'mqtt' node with the topic '/test' connected to a 'msg.payload' node. Both MQTT nodes show a 'connected' status. The right sidebar is currently empty.

```
graph LR; timestamp[timestamp] --> mqtt1[mqtt /test]; mqtt1 --> mqtt2[mqtt /test]; mqtt2 --> payload[msg.payload];
```