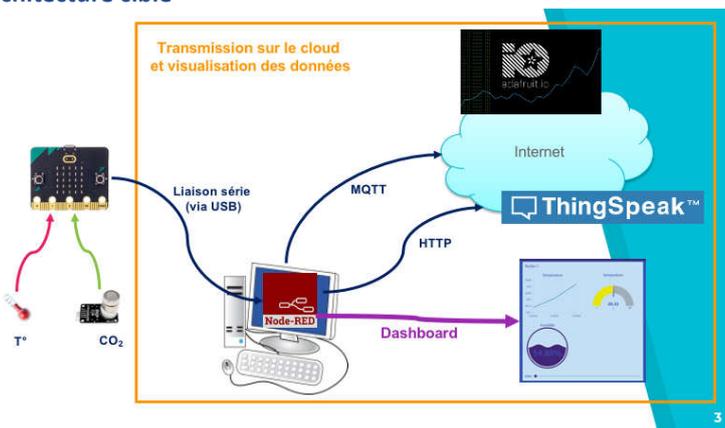


Objectifs

S'approprier les concepts d'internet des objets

Mettre en place une architecture IoT compatible avec du matériel / logiciel utilisables avec les élèves dans le cadre d'un enseignement au lycée.

Architecture cible



Objet = carte Micro-bit avec capteurs connecté via une liaison série (émulée par le câble USB) à un PC

Passerelle IoT = PC avec Node-Red

Collecte et visualisation des données : Dashboard Node-Red en local, clouds IO.adafruit.com ou Thingspeak sur Internet

Définitions

IoT (Internet of Things / Internet des objets) : écosystème d'appareils interconnectés, possédant un identifiant unique, pouvant transmettre des données sur un réseau sans l'intervention d'un humain

Un objet peut être : capteur embarqué dans un véhicule, puce communicante sur un animal, ...

Un objet peut être utilisé de trois façons :

- Transmission d'informations (capteurs connectés, capteurs intelligents)
- Réception d'informations (actionneurs, valves connectés, alarmes, caméras...)
- Transmission ET réception d'informations (objets embarquant un capteur et un actionneur)

Les outils et support

Carte programmable Micro :bit

Le code pour la Micro :bit est fourni, mais il est possible d'utiliser une autre carte programmable qui envoie les informations sur une liaison série via USB. Avec une carte Arduino le code sera adapté avec l'utilisation de `Serial.println()`.

Déroulé

Un powerpoint + des vidéos sur Youtube accompagnent et décrivent les différentes étapes.

Playlist :

<https://www.youtube.com/playlist?list=PLCixRgM2bXnX-glvCNSwPqBWnN4ZJ57Mq>

Pré-requis : installer Node-Red (Windows ou Raspberry Pi)

<https://youtu.be/Y6uKy5OCDgA>

Point d'attention :

Système Debian / Raspberry / Ubuntu : utiliser de préférence le script curl du site Node-Red qui gère toute l'installation y compris de Node.js, npm dans les versions compatibles avec Node-Red. Cette procédure est décrite dans la vidéo.

<https://nodered.org/docs/getting-started/raspberrypi#running-locally>

Étape 1

Programmer la carte Micro :bit pour qu'elle envoie des données sur la liaison série, récupérer les données sur Node-Red

<https://youtu.be/J2-RCy4Tww>

Code source pour l'envoi de la température :

```
import os
import sys

from microbit import *

# liaison série
# ne pas mettre les broches tx et rx pour que le serial soit disponible
pour les données
uart.init(9600)

# boucle infinie
while 1:
    # temporisation 1s
    sleep(2000)
    # lecture de la temperature interne du microbit
    dataTemp = temperature()
    print(dataTemp)
    # creation du buffer format string
    buffer = str(dataTemp)+ "\n"
    # ecriture sur la liaison série
    uart.write(buffer)

sys.exit()
```

Étape 2

Créer un tableau de bord en local sur le PC (ou Raspberry) qui héberge Node-Red

https://youtu.be/MiYBG7_QQW4



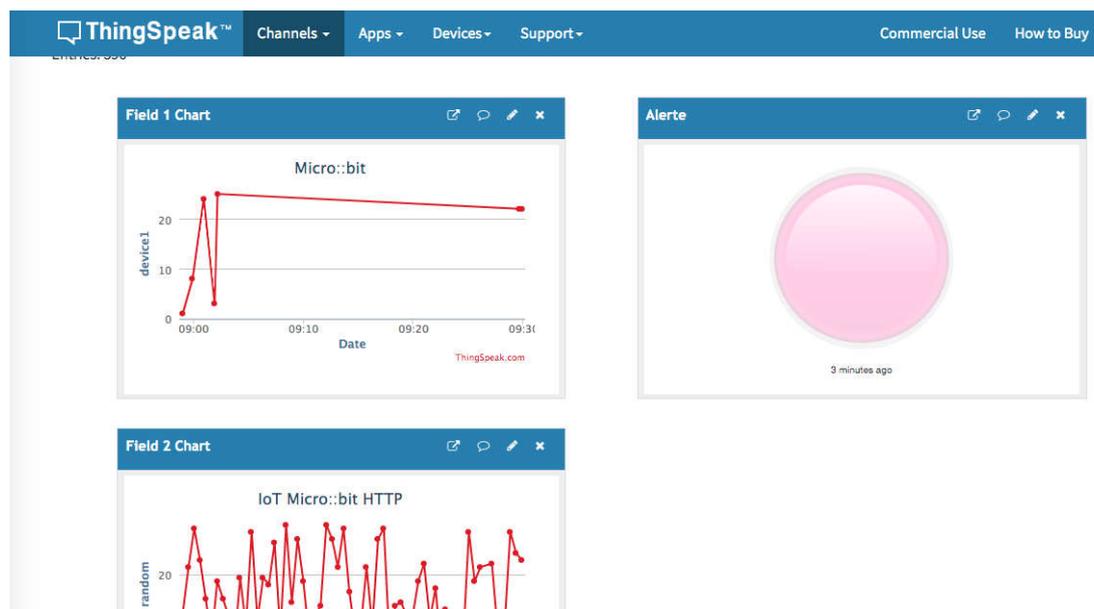
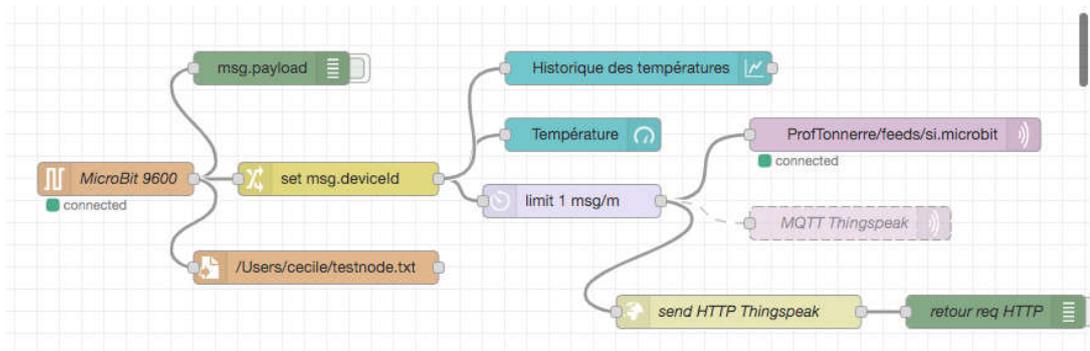
Etape 3

Transmettre les données sur un cloud (io.adafruit.com ou Thingspeak).

Transmission par protocole MQTT : https://youtu.be/c5P79QtO_z8

Attention : la transmission par **protocole MQTT** nécessite que le port 1883 soit ouvert

Solution alternative → Transmettre les données par **protocole HTTP** : <https://youtu.be/rwDsE9auKRI>



Vocabulaire

m2m : machine-to-machine → se dit des communications se faisant d'un appareil à un autre directement

MQTT : Message Queuing Telemetry Transport

Protocole particulièrement adapté au m2m, car il est léger et est prévu pour des trames très petites. Il nécessite peu de ressources et peu de bande passante, et l'implémentation la plus utilisée est basée sur TCP/IP. Il est donc devenu un standard de fait de l'internet des objets.

L'architecture MQTT n'est pas sur le modèle client/serveur mais broker/client. Avec MQTT, un « **broker** » reçoit les demandes de clients : abonnement (**subscribe**) et publication (**publish**) sur des sujets (**topics**), et rediriger les messages à chaque client abonné.

Particularités : Un client qui publie ne sait pas QUI reçoit le message et le broker ne stocke pas les messages. Il y a une hiérarchie arborescente des topics, avec un caractère joker #.

Exemples :

Client 1 publie '12,45' sur le topic 'etage02/bureau203/temperature'.

Client2 abonné à 'etage02/bureau203/temperature' reçoit l'information.

Client3 abonné à 'etage02/bureau203/co2' ne reçoit pas l'information.

Client4 abonné à 'etage02/bureau203/#' reçoit l'information.

Node-Red : Outil graphique de gestion de flux. Sous Node-Red, la donnée circule sous forme de messages. Le traitement des données se fait en connectant des « nodes » d'entrée à des « nodes » de sortie, avec autant de « nodes » intermédiaires que nécessaire.

Node-Red est un logiciel serveur accessible depuis les ordinateurs connectés depuis un navigateur, sur le port 1880 (par défaut).

Node d'entrée : entrée manuelle (au clic ou à intervalles réguliers), lignes ajoutés à un fichier, lecture port série

Node de sortie : requête http, écriture dans une base de données, envoi d'une requête MQTT...

Dashboard : tableau de bord / tableau de visualisation. Écran regroupant des informations de synthèse majoritairement sous formes visuelles. Un dashboard peut aussi comporter des éléments de contrôle (bouton on/off par exemple).

Cloud computing : infrastructure informatique décentralisée, où les données et/ou les applications sont externalisées sur des serveurs distants, le plus souvent chez des hébergeurs spécialisés. Dans l'informatique en nuage, l'utilisateur ne sait pas exactement où sont les serveurs où il se connecte, les données / applications peuvent être réparties sur plusieurs serveurs. Le cloud computing offre la possibilités d'accéder à ses données quelque soit l'endroit et le périphérique (PC, smartphone..).

Il permet aux entreprises de s'affranchir de la gestion des serveurs (disponibilité, mise à jour, sécurité) et de s'adapter facilement aux pics et baisses de charge rapides.

Bibliographie

Node-Red : <https://nodered.org/>

Thingspeak : <https://thingspeak.com/>

Cloud Adafruit : <https://io.adafruit.com/>

Chaîne Youtube des tutoriels : <https://www.youtube.com/@proftonnerre>

Micro : bit : liaison série <https://microbit-micropython.readthedocs.io/en/latest/uart.html>