

Atelier IoT 1 : Analyse de la pollution d'une ville



Introduction

La ville intelligente représente désormais une réalité pour les citoyens de plusieurs villes dans le monde. Cependant, la plupart des habitants n'en connaissent pas les principes ou n'en mesurent pas l'intérêt, ni pour eux ni pour la communauté. Et si la ville connectée constituait l'un des meilleurs moyens d'améliorer leur quotidien et leur bien-être ?

La notation CO_2 est la formulation chimique qui désigne le dioxyde de carbone, un gaz naturellement présent dans l'air. Les sources d'émissions de ce gaz sont aujourd'hui multiples, notamment par les transports ou la production d'électricité. L'accumulation de CO_2 dans l'atmosphère aggrave l'effet de serre naturel, entraînant un réchauffement climatique de la planète et avec lui des conséquences à court et long terme graves. La déforestation contribue à amplifier cet effet. Au-delà de ces problématiques, il s'agit également d'un enjeu de santé publique pour les citoyens du monde entier.

Comment pouvons-nous assurer que notre ville et ses industries ne contribuent pas au réchauffement climatique ? Comment la ville peut-elle estimer son empreinte carbone pour mieux la compenser ? C'est ce que la municipalité de Springfield a décidé d'étudier à l'aide d'un capteur de CO_2 .

Plan de l'atelier

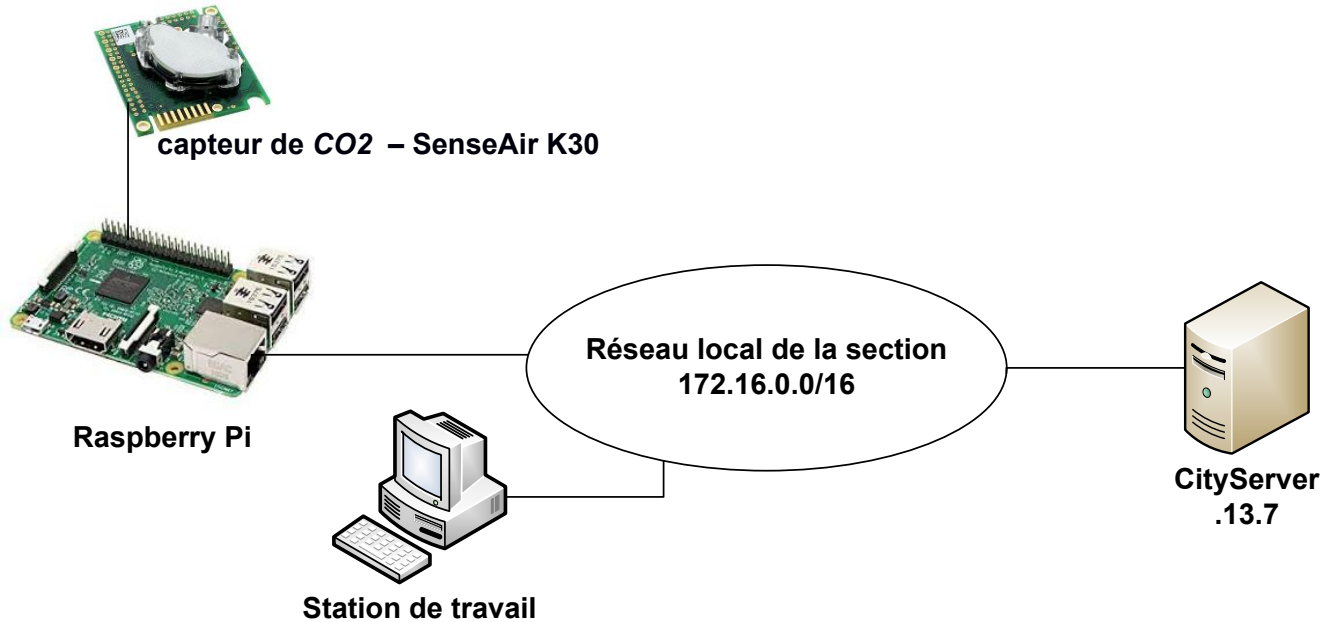
- Étape 1 : Vérification de l'inventaire
- Étape 2 : Préparation et connexion à l'équipement
- Étape 3 : Branchement du capteur de CO_2
- Étape 4 : Collecte des premiers résultats
- Étape 5 : Envoi des informations
- Étape 6 : Vérification des résultats

Atelier IoT 1 : Analyse de la pollution d'une ville

Durée de l'atelier

4 heures

Schéma de topologie



Adressage IP

Équipement	Interface	Adresse IP	Masque de sous-réseau	Passerelle par défaut
Raspberry Pi	Eth0	172.16.X.Y0	255.255.0.0	172.16.13.254
Station de travail	Eth	172.16.X.Y1 ou Y2	255.255.0.0	172.16.13.254

X : numéro de la salle

Y : numéro du sous-répartiteur

Outils nécessaires



- Un Raspberry Pi 3
- Adaptateur secteur pour Raspberry Pi
- Un capteur de CO₂ – SenseAir K30
- 4 câbles de raccordement femelle-femelle (bleu, jaune, rouge, noir)
- Une station de travail
- Un client SSH, type PuTTY

Étape 1: Vérification de l'inventaire

- Vérifiez au préalable que vous disposez des outils nécessaires listés ci-dessus.

Étape 2: Préparation et connexion à l'équipement

Tout d'abord, vous devez mettre en place le nano-ordinateur qui va nous servir lors de cette expérimentation, le Raspberry Pi :

1. Installez le système Raspbian sur la carte micro-SD,
2. Connectez la Raspberry Pi à un écran,
3. Alimentez la Raspberry Pi à l'aide de l'adaptateur secteur fourni,
4. Configurez les paramètres IP
 - a. Sur le logo  > Advanced Options > Modifier les connexions > Connexions filaire 1 >  > Paramètre IPV4 > Add (Mettre votre configuration IPV4) et enregistrer.

5. Activez le SSH par l'interface graphique

- a. Sur le logo  > Préférences > Configuration de Raspberry Pi > Interfaces > SSH

6. Connectez-vous en SSH au Raspberry avec Putty

Login : **pi**

Mot de passe : **raspberrypi**

Une fois ces informations saisies, vous devriez voir apparaître le prompt de l'équipement, sous la forme suivante :

```
pi@raspberrypi:~$
```

Se connecter en root :

Commande : « **sudo -s** »

```
root@raspberrypi:~#
```

Désactivez l'affichage de la console sur le port série à l'aide de la procédure suivante :

- Exécutez la commande « **raspi-config** »
- Sélectionnez l'option « **3 Interfacing Options** »
- Sélectionnez l'option « **I6 Serial Port** »
- Répondez « **No** » à la question « **Would you like a login shell to be accessible over serial ?** », puis « **Yes** » à la question « **Would you like the serial port hardware to be enabled** », puis quittez l'assistant sans redémarrer.

Saisissez la commande suivante pour désactiver le service getty sur le port série :

```
...#systemctl mask serial-getty@ttyAMA0.service
```

Atelier IoT 1 : Analyse de la pollution d'une ville

Assurez-vous que « `console=serial0,115200` » est supprimé dans le fichier `/boot/cmdline.txt`,

Ajouter dans le fichier `/boot/config.txt` la ligne suivante :

```
dtoverlay=pi3-disable-bt
```

Redémarrez l'équipement à l'aide de la commande « `reboot` ».

Étape 3: Branchement du capteur de CO_2

Il s'agit dans cette étape de connecter le capteur de CO_2 au Raspberry Pi.



Avant de procéder, **ÉTEINDRE** le Raspberry Pi pour éviter tout dommage.

Effectuez les branchements électroniques comme mentionné dans le tableau ci-dessous :

Couleur de fil	Référence de pin physique (RPI)	Numéro de pin physique (RPI)	Référence de pin physique (capteur)	Numéro de pin physique (capteur)
Rouge	+5V	2	G+	3
Noire	Ground	6	G0	4
Bleue	TXD0	8	RxD (UART)	2
Jaune	RXD0	10	TxD (UART)	1

Vous trouverez en annexe les descriptifs des branchements côté Raspberry Pi et du capteur :

- **Annexe n°1** : Représentation des connexions électroniques – Raspberry Pi 3
- **Annexe n°2** : Représentation des connexions électroniques – Capteur de CO_2 K30

Une fois ces branchements effectués, **VALIDER** vos branchements électroniques auprès du professeur avant d'alimenter à nouveau le Raspberry Pi.

Étape 4: Collecte des premiers résultats

Une fois le Raspberry Pi redémarré, vous allez devoir communiquer avec le capteur pour en extraire des données :

Copiez sur le Raspberry les deux scripts **P1.py** et **co2_serial.py**

Editez le script **P1.py** :

- Remplacer la valeur "`CityServer.lan`" de la variable « `cityserver` » par son **adresse IP** (l'IP de votre serveur)

Atelier IoT 1 : Analyse de la pollution d'une ville

- Ajouter la ligne « `print(f"La valeur mesurée est de {co2_rounded} ppm")` » en dessous de la ligne « `co2_rounded = round(co2, 2)` » afin de faire apparaître à l'écran les mesures.
- *Remarque : le paramètre « `time.sleep` » (valeur 1 par défaut) indique le temps de pause avant une nouvelle mesure : modifier cette valeur si besoin.*

Testez le script :

```
root@raspberrypi:~#python P1.py
```

Doit apparaître à l'écran les valeurs mesurées en unité de mesure **ppm** (Nombre de molécules du gaz à effet de serre considéré par million de molécules d'air).

Étape 5: Envoi des informations

Vous arrivez désormais à collecter les données venant du capteur. Il vous reste alors à envoyer ces résultats au serveur municipal.

Pour information, le script utilise les paramètres techniques suivants :

- **Type de serveur :** InfluxDB
- **Type de requête :** HTTP POST
- **URI :** `http://172.16.13.7:10000`
- **Identifiant / mot de passe :** `influx_user / w1GYhmBvo`
- **Contenu à envoyer :**
 - `co2_measurement` – catégorie de l'enregistrement
 - `host` – nom d'hôte du Raspberry Pi
 - `value` – valeur du capteur en ppm

Étape 6: Vérification des résultats

Lorsque vous soumettez une donnée au serveur avec succès, vous devriez recevoir le code de retour suivant : « **La valeur mesurée est de 658 ppm** ».

Vérifiez que vos mesures s'affichent sur le graphe du CityServer affiché à l'écran.

Testez le bon fonctionnement de votre capteur en soufflant dessus (émission de Co2) et vérifiez visuellement l'augmentation du taux de Co2 affiché sur le graphe.

Félicitations ! Vous avez désormais un prototype qui fonctionne et qui communique avec le serveur municipal, via le Raspberry Pi.

Annexe n°1

Représentation des connexions électroniques – Raspberry Pi 3

Raspberry Pi 3 GPIO Header			
Pin#	NAME		NAME Pin#
01	3.3v DC Power		DC Power 5v 02
03	GPIO02 (SDA1 , I ² C)		DC Power 5v 04
05	GPIO03 (SCL1 , I ² C)		Ground 06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14 08
09	Ground		(RXD0) GPIO15 10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18 12
13	GPIO27 (GPIO_GEN2)		Ground 14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23 16
17	3.3v DC Power		(GPIO_GEN5) GPIO24 18
19	GPIO10 (SPI_MOSI)		Ground 20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25 22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08 24
25	Ground		(SPI_CE1_N) GPIO07 26
27	ID_SD (I ² C ID EEPROM)		(I ² C ID EEPROM) ID_SC 28
29	GPIO05		Ground 30
31	GPIO06		GPIO12 32
33	GPIO13		Ground 34
35	GPIO19		GPIO16 36
37	GPIO26		GPIO20 38
39	Ground		GPIO21 40

Rev. 2
29/02/2016

www.element14.com/RaspberryPi

Annexe n°2

Représentation des connexions électroniques – Capteur de CO₂ K30

