Atelier IoT 5 : Places de stationnement intelligentes 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 gestion des parcs de stationnement pour une municipalité peut représenter un réel défi. Il faut à la fois répondre aux besoins constatés tout en gardant une politique de l’occupation de l’espace public ouverte sur les autres moyens de transport. Cependant, une municipalité n’a aujourd’hui aucune visibilité sur l’occupation de son espace public, hormis les constats manuels et rapports d’utilisation des parcmètres.

Comment une municipalité peut obtenir des informations concrètes et en temps réel sur l’utilisation de son parc ? Comment la ville peut-elle anticiper le besoin de création de nouvelles places pour servir au mieux ses citoyens ? Enfin, comment la ville peut-elle réduire l’impact écologique des personnes recherchant une place pour y garer leur véhicule ? Pour commencer à répondre à ces questions, la municipalité de Springfield a décidé d’exploiter un capteur de proximité sur une place de stationnement pilote.

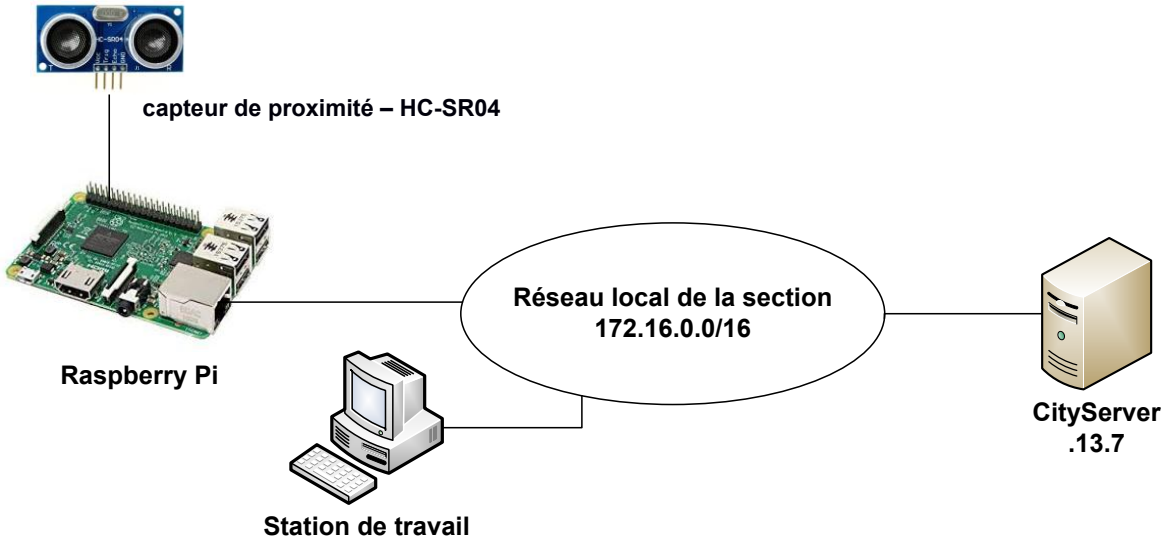
Plan de l’atelier

- Étape 1 : Vérification de l’inventaire
- Étape 2 : Préparation et connexion à l’équipement
- Étape 3 : Branchement du capteur de proximité
- Étape 4 : Collecte des premiers résultats
- Étape 5 : Envoi des informations
- Étape 6 : Vérification des résultats
- Étape 7 : Bonus – déclenchement d’une action automatisée

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

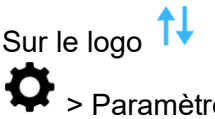
- ☐ Un Raspberry Pi 3
- ☐ Adaptateur secteur pour Raspberry Pi
- ☐ Un capteur de proximité – HC-SR04
- ☐ Une résistance 1k Ω – marron, noir, noir, marron, marron
- ☐ Une résistance 2k Ω – rouge, noir, rouge, marron ou 2 résistance de 1k Ω
- ☐ Un breadboard
- ☐ 4 câbles de raccordement mâle-femelle (bleu, jaune, rouge, noir)
- ☐ 2 câbles de raccordement mâle-mâle (noirs)
- ☐ 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 : raspberry`

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

```
pi@raspberry:~$
```

Se connecter en root :

Commande : « **sudo -s** »

```
root@raspberry:~#
```

Étape 3: Branchement du capteur de proximité

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

Avant de procéder, <u>ÉTEINDRE</u> 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	Vin	1
Noire	Ground	6	GND	4
Bleue	PWM - GPIO17	11	Trig	2
Jaune	PWM – GPIO27	13	Echo	3 **(Attention : voir annexe 2)**

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 proximité HC-SR04

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 le script **P5.py**

Editez le script **P5.py** :

- Remplacer la valeur "**`CityServer.lan`**" de la variable « **`cityserver`** » par son **adresse IP** (l'IP de votre serveur)
- Trouvez un moyen d'extraire les données du capteur, chaque seconde, à l'aide du script **P5.py** de votre création ou existant (le demander au professeur). La donnée doit être représentée en utilisant l'unité de mesure `cm` à deux décimales après la virgule maximum
- Testez le script :

```
root@raspberry:~#python P5.py
```

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 :**
  - `proximity_measure` – catégorie de l'enregistrement
  - `host` – nom d'hôte du Raspberry Pi
  - `value` – valeur du capteur en `cm`

Modifiez votre script pour envoyer toutes les secondes ces informations au serveur municipal.

Étape 6: Vérification des résultats

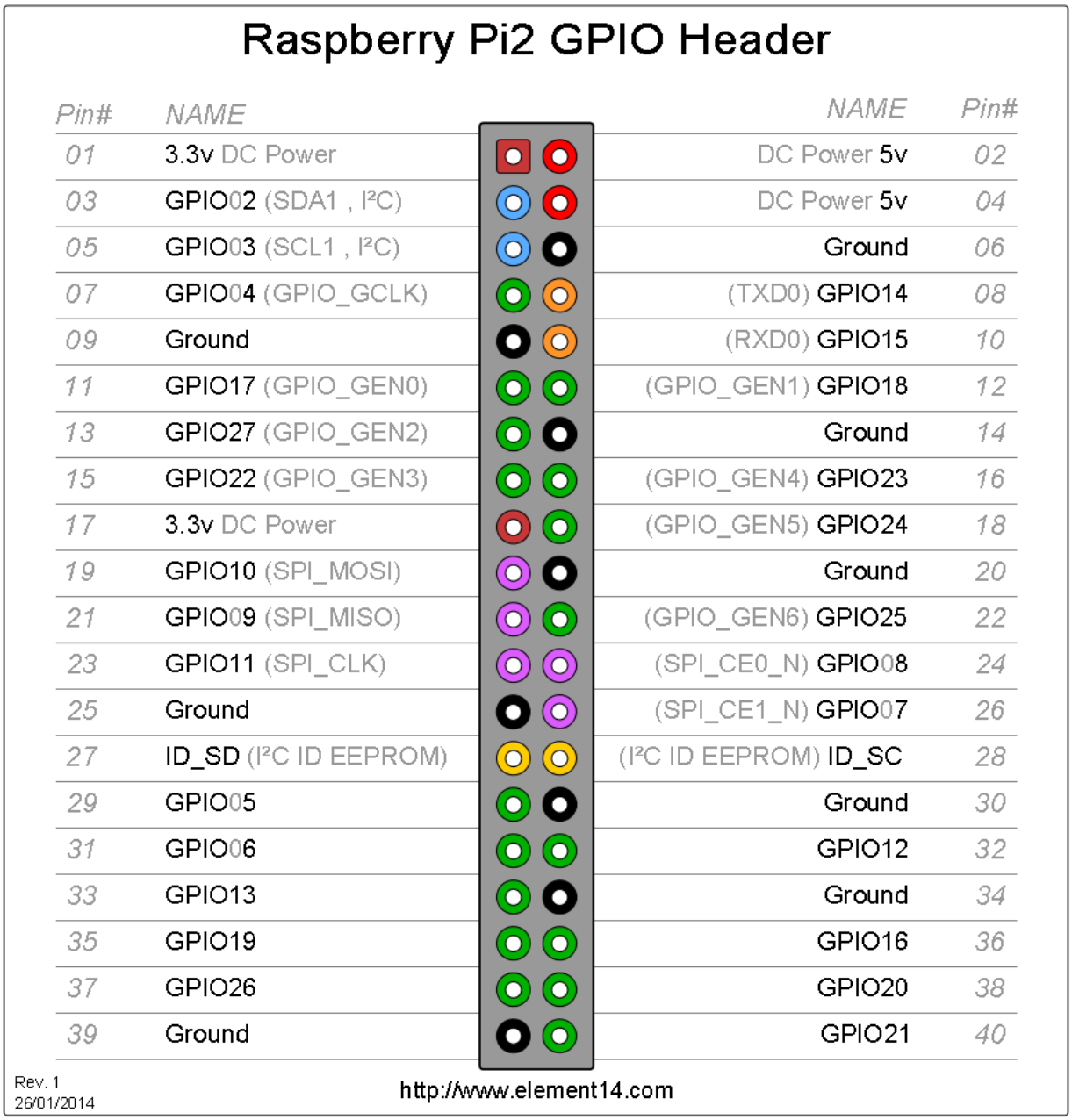
Lorsque vous soumettez une donnée au serveur avec succès, vous devriez recevoir le code de retour suivant : « `204 - No Content` ».

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

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



Annexe n°2

Représentation des connexions électroniques – Capteur de proximité HC-SR04

Attention ! Pont diviseur de tension à former sur la patte Echo, comme suit :