



RevolvAir.org

TP1

Objets connectés au service des citoyens

Science citoyenne



Plate-forme Web et IoT



Assembler et codifier un objet connecté
afin de suivre l'évolution de la qualité de l'air
Arduino, Wifi, capteur de particules fines, température et humidité



Table des matières

Introduction.....	3
Objectif du travail.....	3
Critères d'acceptation de la logique applicative.....	4
Bonus.....	5
Tests unitaires.....	7
Tests d'intégration – preuve vidéo.....	8
Architecture.....	9
Fonctionnement de l'API REST.....	9
Contraintes techniques.....	10
Technologies.....	10
Évaluation.....	10
Remise.....	11
Énoncé de la compétence.....	11





Introduction

Nous respirons environ 15 000 litres d'air par jour! La mauvaise qualité de l'air de nos villes peut avoir un impact négatif sur notre santé. Elle peut provoquer des **allergies**, de **l'asthme**, des **difficultés de concentration** et une **mauvaise qualité de sommeil** et des décès. Dans la ville de Québec, près de 300 décès annuels prématurés sont dus à la mauvaise qualité de l'air. Au Québec, ce sont 4 000 morts prématurées. D'où l'importance de bien connaître ce qui compose notre air.

L'air de nos maisons peut être 5 fois plus pollué que celui extérieur à différents moments de la journée. À diverses occasions, elle peut se détériorer. Notamment, lorsque nous cuisinons, des microparticules et de la fumée sont projetées dans l'air. Les nettoyants que nous utilisons dégagent des composés organiques volatils qui sont fortement liés à l'asthme. Ces COV ont un impact direct sur les poumons, le cœur, le cerveau et nos hormones. Finalement, le radon, un gaz radioactif cancérogène, est présent dans plus de 7% des propriétés canadiennes.

L'objectif du projet **RevolvAir**, est de fournir un produit qui analyse intelligemment la qualité de l'air et présente les données reçues afin de proposer des recommandations pour favoriser votre sécurité et votre santé.

Objectif du travail

L'objectif de ce travail pratique est d'assembler le matériel afin de construire une station d'analyse de qualité de l'air pour mesurer les particules fines de 2.5 μm . Une fois le matériel assemblé, vous serez en mesure de programmer la logique applicative de l'objet connecté et de développer les récits utilisateurs présentés dans la prochaine section.

Ce travail sera réalisé en équipe de 2.



Critères d'acceptation de la logique applicative



Voici la liste des critères d'acceptation pour le travail à réaliser :

- L'objet doit **lire les valeurs des capteurs de particules fines** aux 2 minutes.
- L'objet doit **lire les valeurs du capteur de température et humidité** aux 2 minutes.
- L'objet doit **envoyer les valeurs des capteurs à l'API REST** afin de sauvegarder les données aux 2 minutes.
 - PRÉCONDITION : L'objet doit se connecter et demeurer connecté au réseau Wifi afin de pouvoir envoyer les données à la plate-forme infonuagique.
 - L'objet tente de se reconnecter continuellement toutes les 20 secondes si le réseau Wifi est déconnecté.
- La lumière **DEL change de couleur** en fonction de l'état du capteur
 - Si la connexion Wifi est impossible, la lumière DEL clignote rapidement en **bleu**.
 - Si la valeur PM 2.5 est entre 0 et 11, la DEL est **verte**.
 - Si la valeur PM 2.5 est entre 12 et 34, la DEL est **jaune**.
 - Si la valeur PM 2.5 est entre 35 et 54, la DEL est **orange**.
 - Si la valeur PM 2.5 est entre 55 et plus, la DEL est **rouge et clignote**.
- L'objet doit offrir un **Hot Spot et un serveur Web** afin de permettre à un appareil mobile de se connecter directement à l'objet connecté pour visualiser les données des capteurs.

Voici les pages Web à développer :

- Page des **Particules fines** :
 - Afficher les données ponctuelles de PM 2.5 à intervalle de 5 secondes.
 - Unités des données affichées : $\mu\text{g}/\text{m}^3$.
 - Afficher la description qui représente la qualité de l'air (<https://revolvair.org/indice-de-qualite-de-lair-iga/> - Échelle pour les données brutes PM 2.5 en microgrammes par mètre cube ($\mu\text{g}/\text{m}^3$))
- Page **Température et humidité relative** :
 - Afficher les données température et d'humidité à intervalle de 5 secondes.
 - Unités degrés Celsius ($^{\circ}\text{C}$) et Humidité relative (%HR)



- Page **Wifi** :
 - Afficher les données de connexion Wifi (SSID et force du wifi).
- **Menu**
 - Afin de naviguer entre les pages, un menu est disponible tout en haut.
 - PM25, température et humidité
 - WiFi
 - À propos du logiciel (information sur les auteurs)
- **Débogage**
 - Le moniteur série doit travailler à 115200 bauds.
 - Le moniteur série affiche les données suivantes :
 - La **Mac ID** et le **Device Id**.
 - Le SSID du réseau Wifi utilisé.
 - En boucle :
 - Valeurs suivantes :
 - PM 1, PM 2.5, PM10
 - DCL 2.5
 - IQA de l'air
 - Température
 - Humidité
 - Suite à une requête POST à l'API REST
 - L'URL utilisée pour envoyer les données
 - Le code de retour (200, 401)

Bonus

- 5 points : Intégrer le Wifi Manager afin de changer la configuration du réseau sans-fil.
- 2 points : Intégrer le BUZZER afin de lancer un signal sonore lorsque les PM25 dépassent 50.
- 2 points : Afficher les données sous forme de graphique pour la page Page des Particules fines



**Avant de lancer le travail sur un
BONUS,
vérifier votre bonne compréhension
avec le chargé de projet.**

**Certaines modifications pourront être
apportées tout au long du projet.**

**Toute proposition pour améliorer le
produit sera considérée et évaluée.**

**Réservez une revue de code avec
votre enseignant au moins 3 jours
avant la remise**





Tests unitaires

Les méthodes suivantes devront être accompagnées de tests unitaires.

String getAQILabelFromPM25(float pm25Value)

Cette méthode retourne le nom du niveau de pollution en fonction de la valeur de PM 2.5.

- Bon, acceptable, mauvais, ...

Utiliser le tableau présenté à cette adresse :

<https://revolvair.org/indice-de-qualite-de-lair-iqua/#echelle-ugm3>

Dans la section intitulée

« Échelle pour les données brutes PM 2.5 en microgrammes par mètre cube ($\mu\text{g}/\text{m}^3$) »

rgbData getColorFromPM25(float pm25Value)

Cette méthode retourne une structure définissant la couleur RGB de la DEL en fonction de la valeur de PM 2.5.

La fonction getColorFromPM25(float pmValue) devra être accompagnée de tests unitaires.

- getColorFromPM25(0)
- getColorFromPM25(15)
- getColorFromPM25(16)
- getColorFromPM25(30)
- getColorFromPM25(150)

test_DEL_color_changes_when_pm25_value_increment_from_0_to_300()

Ce test permet de faire varier la couleur de la DEL en fonction des PM détectées.

Celui qui lance les tests doit pouvoir visualiser le changement de couleur.

Penser à ajouter un délai de 10 ms.



Tests d'intégration – preuve vidéo

Fournissez la preuve que votre objet connecté répond aux demandes du client.

Créez une courte vidéo (3 minutes maximum) du bon fonctionnement de votre système.

Voici les éléments à inclure :

- Connexions de votre appareil
- Affichage des données de la console.
- Affichage des données de la page Web du Hot spot.
- Affichage des données de la station sur la plate-forme Web.

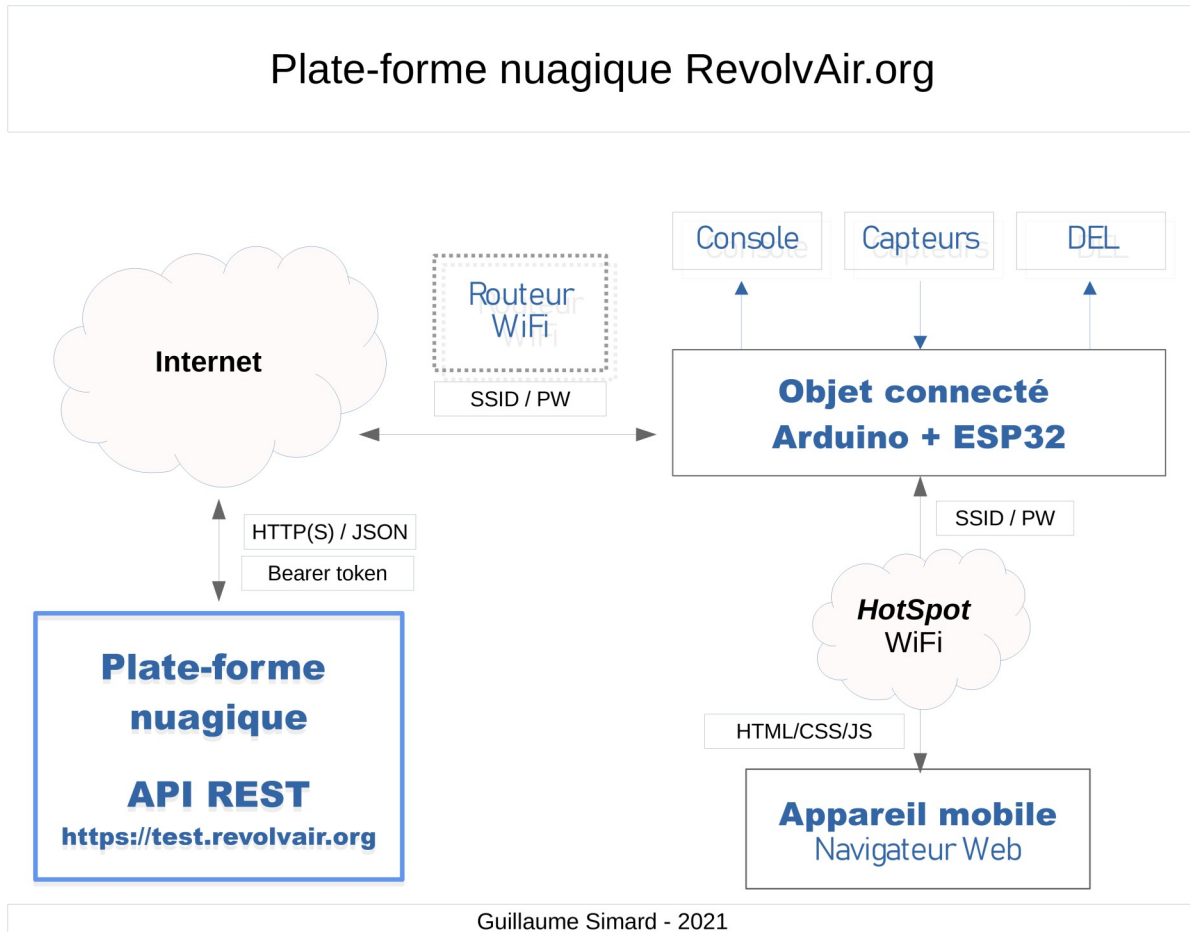
Conseils :

- Éviter les mouvements de caméra trop rapides.
- Assurez-vous que le sujet est en focus.
- Faire un cadrage précis pour faciliter la lecture ou la compréhension.
- Décrivez le bon fonctionnement en parlant doucement et clairement.
- Préparer un script et y intégrer tous les éléments importants.



Architecture

Le schéma d'architecture suivant présente les éléments importants de la plate-forme :



Fonctionnement de l'API REST

Le document **routes-api-revolvair-mobile-iot.pdf** contient la liste des routes à utiliser afin d'envoyer les données à la plate-forme infonuagique. De façon générale, afin d'échanger des données avec l'API, il faut s'authentifier et lancer une requête de type GET ou POST. GET permet de demander les données et POST est utiliser pour envoyer des données.

En résumé, il faudra lancer une requête POST afin d'envoyer de façon régulière les dernières valeurs des capteurs.



Contraintes techniques

- Le code devra être hébergé sur **GitLab** (repo privé et inviter l'utilisateur @ifocus22)
- Les **tests automatisés** doivent être maintenus.
- **BONUS** : Une **intégration continue** doit être mise en place avec les branches usuelles.
 - <https://docs.platformio.org/en/latest/integration/ci/gitlab.html>

Technologies

- Programmation Arduino – Langage C et C++
- Client : HTML / CSS / JavaScript
- Échange avec une API REST : protocole HTTP (méthodes GET et POST)
- Base de code existante

Évaluation

Ce travail vaut pour 35% de la note finale.

Énoncé		Valeur
Respect des critères d'acceptation	(00SX 4)	60
Qualité de la codification (OO et SOLID)	(00SX 4)	12
Qualité des tests	(00SS 6)	8
Respect du devis pour l'assemblage matériel	(00SX 6)	10
Qualité de l'interface utilisateur	(00SX 3)	5
Documentation de l'utilisateur	(00SX 6)	5
	Total	100



Remise

Remettre le contenu zippé sur **Omnivox** au plus tard le 27 septembre 2022 à minuit.

Code

- Dernière version du code fonctionnel (branche *master*) avec les tests.

Lisez-moi.txt

- Noms des développeurs.
- Lien vers le projet GitLab (la branche *master* sera évaluée).
- Lien vers la page Web des données de la station.
- Bogues connus.

Documentation

- Compléter le gabarit (documentation-revolver.docx).
- Assemblage du matériel, compilation du code et mise en marche.

Vidéo

- Preuve du bon fonctionnement de votre projet.

Énoncé de la compétence

- **00SX** - Effectuer le développement d'applications pour des objets connectés.
 - **00SS 6** - Contrôler la qualité de l'application.
 - **00SX 4** - Programmer la logique applicative de l'objet et la logique applicative de contrôle ou de surveillance.
 - **00SX 6** - Participer à la mise en service de l'application.
 - **00SX 3** - Générer ou programmer l'interface utilisateur.
- Installer les micrologiciels des objets et établir la connectivité pour la communication des objets;
- Programmer les objets connectés pour collecter les données des capteurs, les prétraiter et les transmettre à un serveur pour colliger et publier les données;
- Développer une application pour analyser les données, en utilisant des services d'échange de données; Développer une interface pour afficher les données et les résultats de l'analyse.