
Projet n°4 ICT: Mise en place d'un capteur de niveau d'eau d'une serre

le 19 janvier 2021
Encadrant: Mr FERRARI Jérôme

Table des matières

I- Etat de l'art: les capteurs mesure de niveau d'eau	3
II- Démarche du projet et problèmes rencontrés	4
Création du capteur	4
Serveur TTN	6
Stockage dans une base de données permanente INFLUXDB	7
Utilisation de PuTTY	7
Visualisation des données avec Grafana	8
III- Conclusion	9

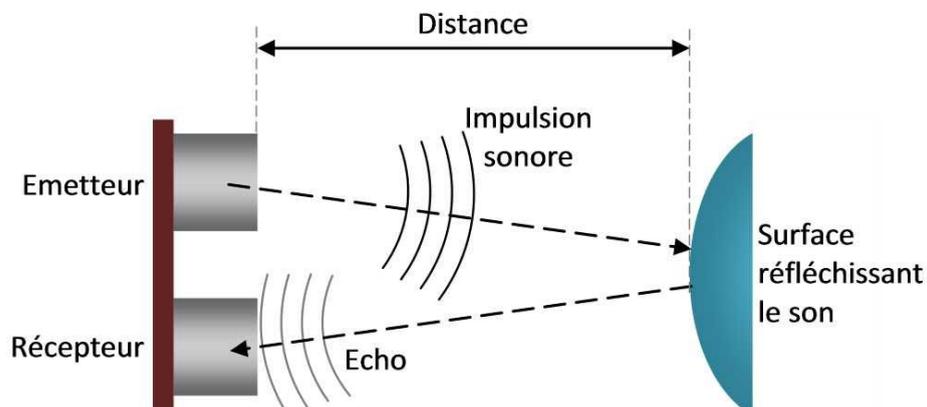
I- Etat de l'art: les capteurs mesure de niveau d'eau

Le tableau ci-dessous récapitule les différents capteurs existants pour mesurer un niveau d'eau.

Technologie	Étendue de mesure	Précision	Gamme de Température	Pression max (bar)
Flotteur	10 mm à 30 m	0,5 à 5 %	-180 à 450 °C	100
Plongeur	30 cm à 6 m	0,5 %	-150 à 450 °C	40
Palpeur électromécanique	50 max	1 mm	-160 à 180 °C	25
Capteur de pression	0 à 25 bar	0,5 à 1 %	-20 à 120 °C	40
Sonde capacitive	0 à 10 m	1 %	-20 à 85 °C	40
Radar	0 à 30 m	10 mm	-	-
Ultrasons	0 à 50 m	1 %	-190 à 250 °C	40

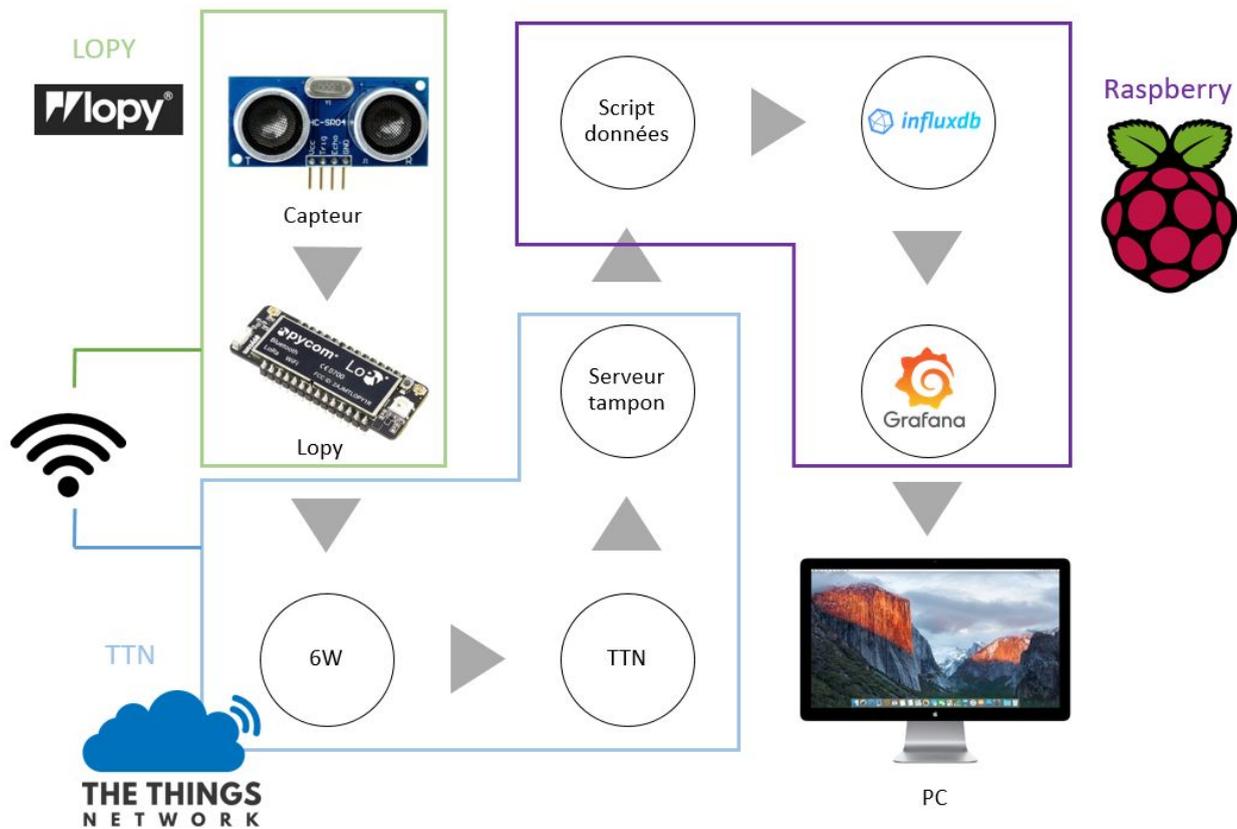
Le capteur à ultrasons semble être adapté à beaucoup d'utilisations mais nous aurions pu ici nous contenter d'un capteur de type flotteur ou de type plongeur car il s'agit d'une serre aquaponique dont la hauteur d'eau du bassin est de 55cm de plus on cherche à mesurer une différence de hauteur de l'ordre du mm.

La figure ci-dessous explicite le fonctionnement d'un capteur ultrasons:



II- Démarche du projet et problèmes rencontrés

L'objectif de ce projet est de pouvoir mesurer le niveau d'eau dans une serre aquaponique et de pouvoir afficher les données récupérées sur un logiciel de visualisation de données. Le schéma présenté ci-dessous représente tout le processus, de la récupération des données par le capteur à la visualisation sur un PC.



1) Création du capteur

Nous avons à notre disposition une carte Lopy 4 (nom du wifi: Lopy wLan 6880) ainsi qu'un capteur Hc-SR04 ultrasonic.

La première étape est la création du capteur avec les branchements ci-dessous:

Lopy <--> HC-SR04

Vin <--> Vcc

Gnd <--> Gnd

Echo <--> G7

Trigger <--> G8

Notre démarche a été la suivante: nous avons dans un premier temps entrepris de communiquer via Lora avec Lopy en suivant un tutoriel. Nous avons rencontré quelques difficultés pour établir la connexion avec la Lopy mais nous avons finalement réussi. Nous avons dû à plusieurs reprises changer de carte. Cela est dû au fait que c'est une carte de développement dont le firmware est encore en cours d'amélioration. Ce qui amène à des instabilités dans son fonctionnement

Lorsque cela était réglé, nous avons entré le code afin de lire les données envoyées par le capteur sur le logiciel Atom. Nous avons obtenu :

```
Connecting to 192.168.4.1...
  Out of range
Distance 259 cm
Distance 344 cm
Distance 332 cm
Distance 361 cm
```

L'étape suivante fut de connecter la Lopy à la console The Things Network. Nous avons de nouveau rencontré des difficultés à nous connecter, la carte Lopy ne se joignant pas. Nous avons dû revoir le branchement de l'antenne sur la carte Lopy. Sans vraiment comprendre pourquoi la carte Lopy se joignait ou non à la console, au bout d'un temps plus ou moins long. Mais nous avons finalement pu nous connecter.

Après s'être connecté, il faut ensuite envoyer les données à la console, pour cela nous avons le code suivant :

```
##### Main program #####
def sonar():
    echo = Pin(Pin.exp_board.G7, mode=Pin.IN)
    trigger = Pin(Pin.exp_board.G8, mode=Pin.OUT)
    trigger(0)
    chrono = Timer.Chrono()
    while True:
        chrono.reset()
        trigger(1)
        time.sleep_us(10)
        trigger(0)
        while echo() == 0:
            pass
        chrono.start()
        while echo() == 1:
            pass
        chrono.stop()
        distance = chrono.read_us() / 13.51
        pkt_2=str(distance).encode()
        s.send(pkt_2)
        time.sleep(2)

pycom.heartbeat(False)

sonar()
##### End of Main program #####
```

Nous avons adapté le code effectuant la distance dans l'air à un code calculant la distance dans l'eau.

2) Serveur TTN

Application ID: mp4capteurwater

Device ID: mp4capteur2lo

Un serveur TTN relie les objets à internet, on va donc se servir de TTN pour créer un serveur tampon pour stocker temporairement nos données issues de la Lopy.

Les données de distance enregistrées par le sonar, ont été envoyées sur la console, il a fallu ensuite les décoder grâce au code suivant :

```
1 function Decoder(bytes, port) {
2   var decoded = {};
3   decoded.distance_cm = "";
4   for(i=0; i<bytes.Length; i++){
5     decoded.distance_cm = decoded.distance_cm + String.fromCharCode(bytes[i])
6   }
7   return decoded;
8 }
```

Finalement, la console affichant la valeur de la distance calculée par le capteur toutes les 5 secondes :

▲ 10:36:20	140	2	dev id: lgy_sonar	payload: 33 31 2E 30 35 30 38 36	distance_cm: "31.05086"
▲ 10:36:15	139	2	dev id: lgy_sonar	payload: 33 31 2E 31 35 39 30 35	distance_cm: "31.15905"

3) Stockage dans une base de données permanente INFLUXDB

Tout d'abord, expliquons ce qu'est InfluxDB. Il s'agit d'une TSDB (Time Serie Data Base) qui est faite pour stocker une grande quantité de données chronologiques ou de séries temporelles venant de sources multiples. Elle est utilisée pour collecter des données de l'internet des objets.

Les bases de données de type InfluxDB sont plutôt utilisées sous Linux or nous n'avions pas accès à un système Linux lors des manipulations. Nous avons essayé d'adapter la démarche sous Windows en suivant les instructions du lien suivant: <http://richardn.ca/2019/01/04/installing-influxdb-on-windows/> mais nous avons rencontré de nombreux problèmes techniques dont nous n'avions pas la solution ni même sur Internet. Ces problèmes faisaient appel à des connaissances en dehors de nos compétences car nous ne sommes pas spécialisés précisément dans ce domaine.

4) Utilisation de PuTTY

Nous avons donc trouvé une alternative pour pouvoir avoir accès à InfluxDB. Nous avons installé le logiciel PuTTY sur notre ordinateur personnel afin d'être connecté à un micro-ordinateur du type Raspberry Pi et d'avoir accès à un environnement Linux. Il a donc fallu réaliser un script à envoyer sur cette Raspberry Pi pour pouvoir récupérer les données sur un serveur tampon TTN. A partir d'une base de script adaptée à notre utilisation, nous avons essayé de récupérer les données issues du capteur.

Dans un premier temps un script bien abouti nous a été donné par notre professeur. Lancer la commande "sudo nano Groupe4.py" nous permet d'ouvrir le programme Groupe4.py dans l'éditeur nano, le complément sudo permet de nous considérer comme "Super Utilisateur". Nous avons dans un premier temps éprouvé des difficultés car nous n'arrivons pas à récupérer toutes les données mais seulement la dernière, sachant que le capteur envoie des données toutes les 15 minutes.

La ligne suivante permet d'aller sur le site et de demander des données:

```
response=requests.get('https://capteurs_garage.data.thethingsnetwork.org/api/v2/query?last=15m',  
headers=headers).json()
```

Ici nous récupérons les données du capteur (ph et température) des 15 dernières minutes (puisque le capteur récupère environ une donnée tous les quart d'heure, nous en récupérons donc qu'une seule aussi). Nous allons donc modifier last=15m en last=2d pour obtenir toutes les valeurs captées les deux derniers jours.

La commande "sudo python3 Groupe4.py" nous permet de lancer notre programme dans l'éditeur Python3. Nous avons ainsi toutes les valeurs des deux derniers jours.

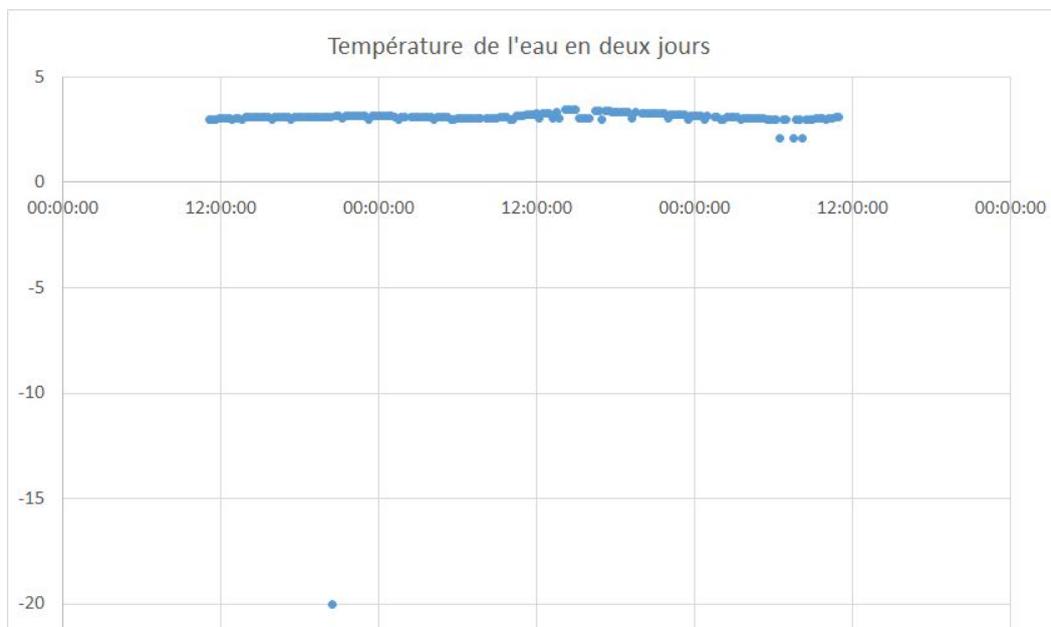
Avec les autres groupes travaillant sur des projets similaires, nous nous sommes répartis les tâches et un des autre groupe a créé la base de données permanente sous InfluxDB.

Puisque nous dépendions d'un autre groupe, nous avons récupéré les données obtenus, filtré avec un peu de python puis rangé les valeurs dans excel. Les graphiques obtenus dans la partie suivante proviennent d'un tableur EXCEL.

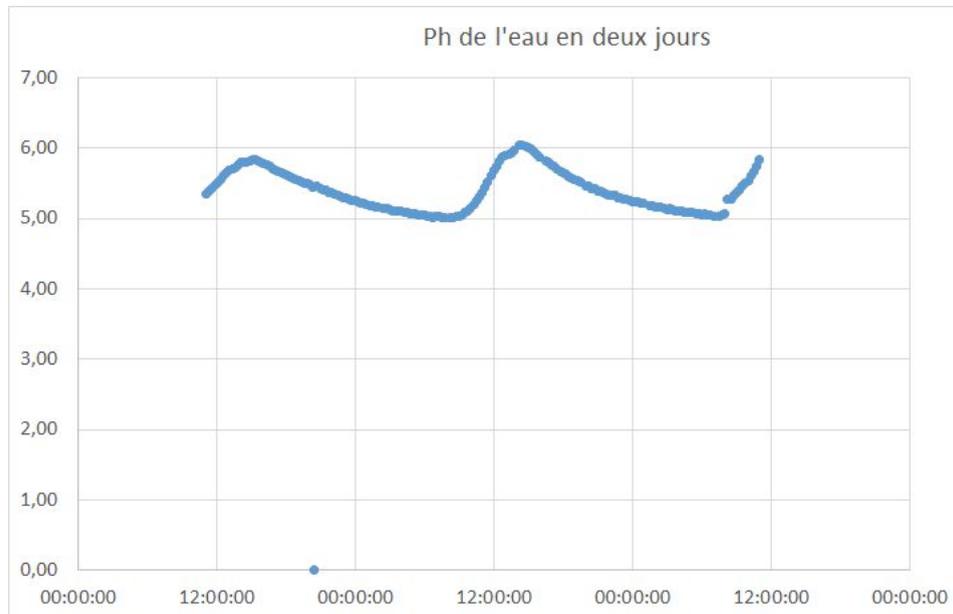
5) Visualisation des données avec Grafana

Grafana est un logiciel qui permet de réaliser des tableaux de bord et des graphiques depuis plusieurs sources dont des bases de données temporelles comme Graphite, InfluxDB et OpenTSDB. Dans notre cas, la base de données temporelle sera InfluxDB.

Nous n'avons pas eu le temps de visualiser les données car installer InfluxDB a déjà pris beaucoup de temps. De plus, durant l'hiver, quand nous avons fait nos expériences, les variations d'eau dans le bassin sont trop faibles (-2mm par jour) pour le sonar. Nous nous sommes donc intéressé aux traitement des données de température/PH. Ainsi nous avons créé des graphiques, à partir des données brutes pour pouvoir tout de même effectuer une visualisation de données. Les deux graphiques ci-dessous représentent la température de l'eau et le ph de l'eau.



La température stagne aux alentours de 3 degrés. On s'attendait à percevoir une augmentation de température le jour et une diminution la nuit mais ce n'est pas le cas. Peut-être que le bassin contient beaucoup d'eau et que la serre isole bien le bassin, ce qui réduit fortement les fluctuations de température de l'eau.



Le Ph oscille plutôt régulièrement entre 5 et 6. Nous ne savons pas si des plantes sont dans la serre, si c'est le cas, cela expliquerait les variations de pH, la journée les plantes génèrent du dioxygène grâce à la photosynthèse, il y a donc moins de CO₂ dans la serre, puis moins de CO₂ absorbé par l'eau, la rendant moins acide. La nuit c'est l'inverse, l'augmentation du taux de CO₂ de la serre fait augmenter l'acidité de l'eau.

On constate un point aberrant à 20h27, on obtient une pH de 0 et une température de -20°C. Evidemment cela ne correspond pas aux conditions réelles, il s'agit des valeurs extrêmes que le capteur peut envoyer. En effet cela vient du fonctionnement du capteur, en général il envoie une tension, entre 0 et 5 Volt par exemple, qui correspond à une donnée réelle en fonction de sa capacité. Pour le pH, il varie entre 0 et 14, donc pour signaler un pH de 7, le capteur envoie une tension de 2,5 V. Pour conclure, la valeur de 0 pH et la température de -20°C signifie que l'information n'est pas passée, soit le capteur n'a pas émis de valeur (donc 0V) soit le récepteur n'a pas reçu de signal et l'a interprété comme 0.

III- Conclusion

Nous avons rencontré beaucoup de difficultés lors de ce projet mais il a permis de se familiariser avec des technologies que nous n'avions pas forcément rencontrées avant. Ce sont des technologies faciles d'accès et peu chères mais très peu fiables, ce qui peut poser problèmes pour des projets qui dépendent intégralement de la connexion LoRA. Nous n'avons malheureusement pas pu mener à bien le projet jusqu'à la fin mais avons compris la démarche et nous savons qu'avec plus de temps nous aurions pu mener à bien le projet jusqu'au bout.