

Projet MP3 : Mise en place capteur température eau serre

Objectif du TP : Monitoring d'une serre aquaponique en analysant les données d'un capteur de température dans l'eau

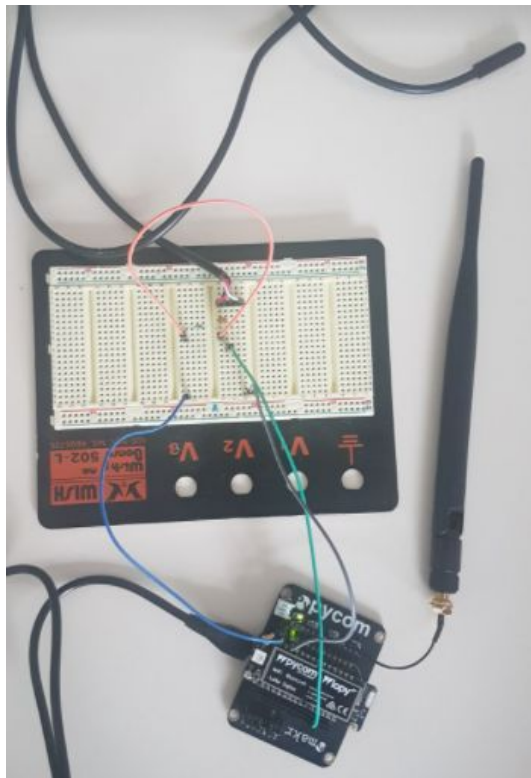
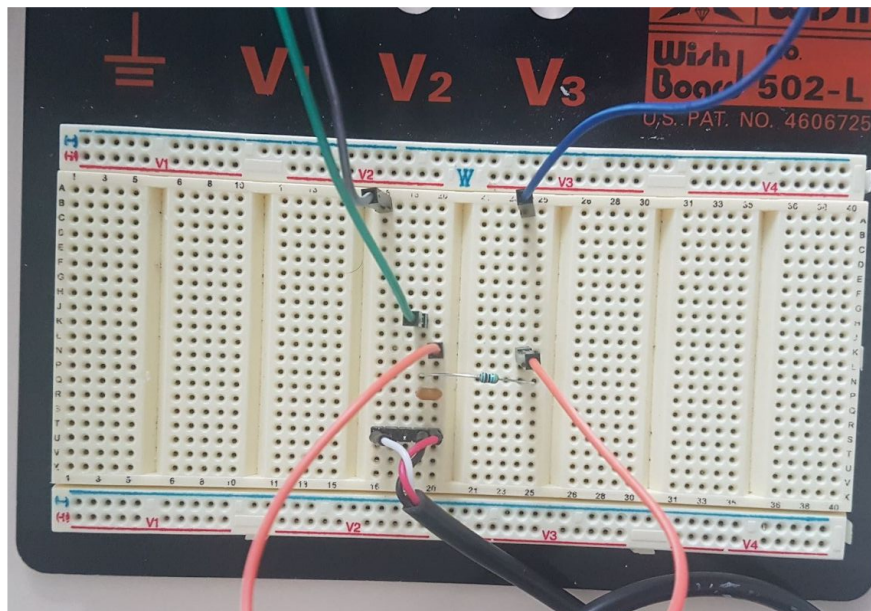
Tout au long de ce projet, nous nous sommes inspirés d'un travail effectué par Sylvain, un ancien camarade de l'ENSE3. Il explique toute sa démarche, afin d'obtenir des informations issues d'un capteur légèrement semblable au nôtre, dans la serre située sur le toit de l'école. L'article est à retrouver via [ce lien](#).

1) Construction du capteur de température.

La principale différence avec l'article de Sylvain a été le capteur lui-même. En effet, nous ne disposions pas du même type de capteur.

Nous avons alors rencontré quelques difficultés, car il nous manquait une partie du capteur pour l'assemblage, à savoir l'adaptateur d'impédance. Ce composant permet au signal provenant du capteur d'être compris par le microcontrôleur.

Pour remplacer l'adaptateur d'impédance, nous avons réalisé un branchement avec une résistance de $4,7\text{ k}\Omega$ et une capacité de 100 nF , comme montré ci-dessous sur les photos du montage global.



2) Branchement du micro contrôleur Pycom

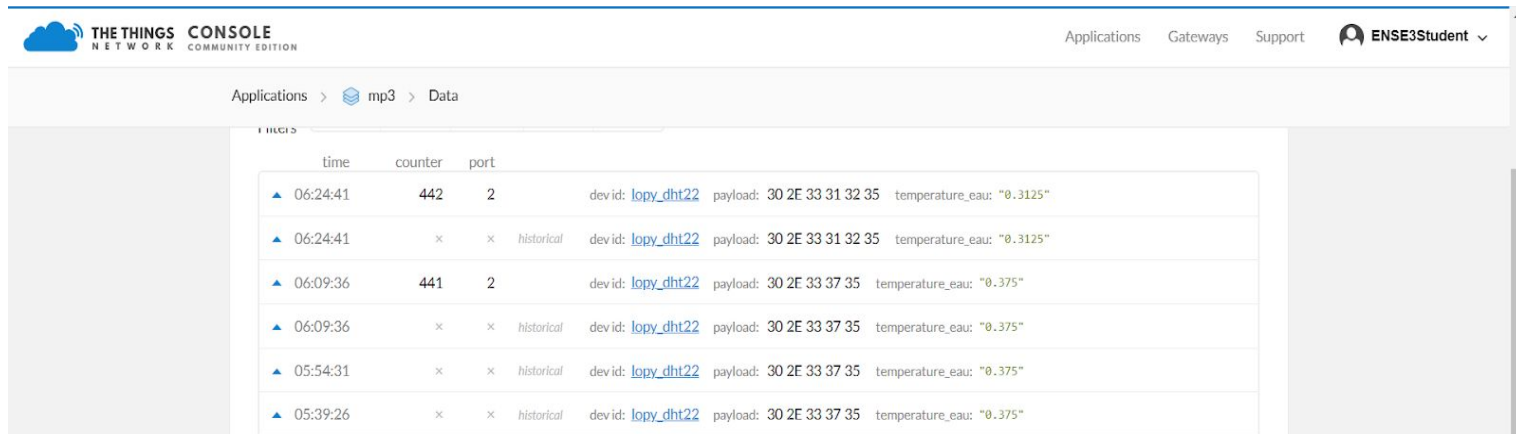
Par la suite, nous effectuons les branchements nécessaires pour connecter la Pycom à notre ordinateur. La Pycom est un microcontrôleur consommant très peu d'énergie et permettant de faire le lien entre les données du capteur et le serveur tampon via le protocole de télécommunication LoRa. Il s'agit d'une communication à bas débit pour des objets à faible consommation électrique et connectés à Internet. Il fallait donc faire le raccordement à

la prise USB et la connexion au serveur wi-fi adéquat. Ensuite, à l'aide du logiciel Atom, nous avons rédigé un code en python permettant l'acquisition et le traitement des données de températures renvoyées par le capteur à la Pycom. Une fois ce code finalisé, il faut le télécharger sur la Pycom.

Là encore, nous avons rencontré des difficultés. Nous avons essayé de nous inspirer du code proposé dans le projet de Sylvain, mais comme mentionné précédemment, le capteur utilisé n'était pas le même. Ne parvenant pas à adapter correctement le code, nous avons dû, pour la suite du projet, utiliser les données récupérées par un autre groupe.

3) Création d'un serveur tampon sur TheThingsWork

Une fois le Pycom réglé et paramétré pour notre projet, nous allons utiliser un serveur tampon pour récupérer les informations liées au capteur de température. Pour cela, nous avons créé notre propre serveur, avec l'appareil associé. En récupérant les caractéristiques du serveur créé, on modifie le programme proposé à l'intérieur du projet pour faire apparaître les données de température exclusivement. Initialement, le programme fournissait des données d'humidité, mais le capteur étant immergé dans un milieu aquatique, ces informations ne nous concernent pas. On modifie alors en conséquence le programme python. Quand le Pycom est branché, on obtient des données toutes les 5 minutes. Cela apparaît sous la forme montrée ci-dessous :



THE THINGS CONSOLE
NETWORK COMMUNITY EDITION

Applications Gateways Support ENSE3Student

Applications > mp3 > Data

time	counter	port	dev id	payload	temperature_eau	
06:24:41	442	2	lopy_dht22	30 2E 33 31 32 35	"0.3125"	
06:24:41	x	x	historical	dev id: lopy_dht22	payload: 30 2E 33 31 32 35	temperature_eau: "0.3125"
06:09:36	441	2	lopy_dht22	30 2E 33 37 35	temperature_eau: "0.375"	
06:09:36	x	x	historical	dev id: lopy_dht22	payload: 30 2E 33 37 35	temperature_eau: "0.375"
05:54:31	x	x	historical	dev id: lopy_dht22	payload: 30 2E 33 37 35	temperature_eau: "0.375"
05:39:26	x	x	historical	dev id: lopy_dht22	payload: 30 2E 33 37 35	temperature_eau: "0.375"

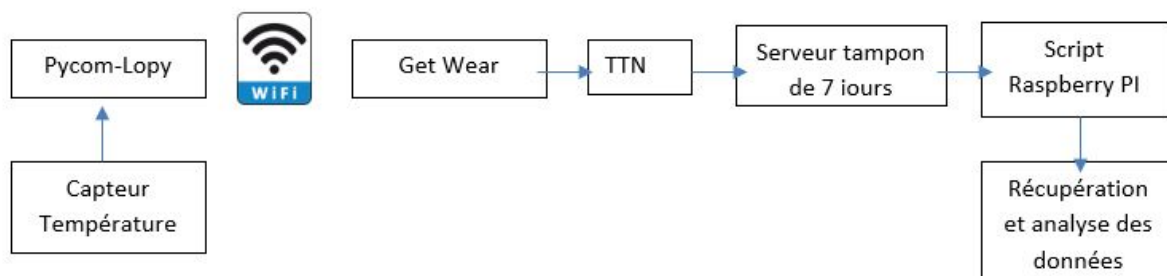
Ces données sont stockées sur ce serveur pour une durée de 7 jours maximum. Si on veut obtenir un stockage sur une durée plus grande, il faut utiliser un autre moyen, en stockant les résultats dans une base de données InfluxDB par exemple. Ce travail a été effectué par un autre groupe d'étudiants.

4) Connection à la machine virtuelle (équivalent à un Raspberry) afin de récupérer les données stockées sur le serveur tampon TTN

Plutôt que de passer par un outil de gestion de base de données comme InfluxDB, nous allons utiliser une Raspberry PI afin de stocker les informations de façon permanente issues du capteur. Cela permettra, dans le cas où il y a un souci avec l'ordinateur ou le script, de récupérer tout de même les informations.

Un Raspberry est un nano-ordinateur à tout faire et très peu cher (moins de 50€ selon les modèles). Dans notre projet, il va nous servir à récupérer les données stockées sur le serveur tampon.

Voici un schéma global des interconnexions entre les différents outils utilisés :



Afin d'insérer un script sur la Raspberry PI, nous allons utiliser une connexion virtuelle, via le logiciel Putty.

On implémente alors le script affiché ci-dessous sur la carte. Cela devrait nous permettre de récupérer les données d'un autre capteur, qui est lui situé dans un garage et mesure à la fois une température et un pH, mais le raisonnement serait le même si on accédait aux données de n'importe quel capteur, il faudrait seulement ajuster le code. Ce dernier effectue ici des mesures toutes les 15 minutes.

```

pi@raspberrypi ~
GNU nano 3.2 Groupe2.py
import json
import requests
import paho.mqtt.client as mqtt #import the client
import time
broker_address="localhost"
client = mqtt.Client("F1") #create new instance
client.connect(broker_address) #connect to broker

while True:

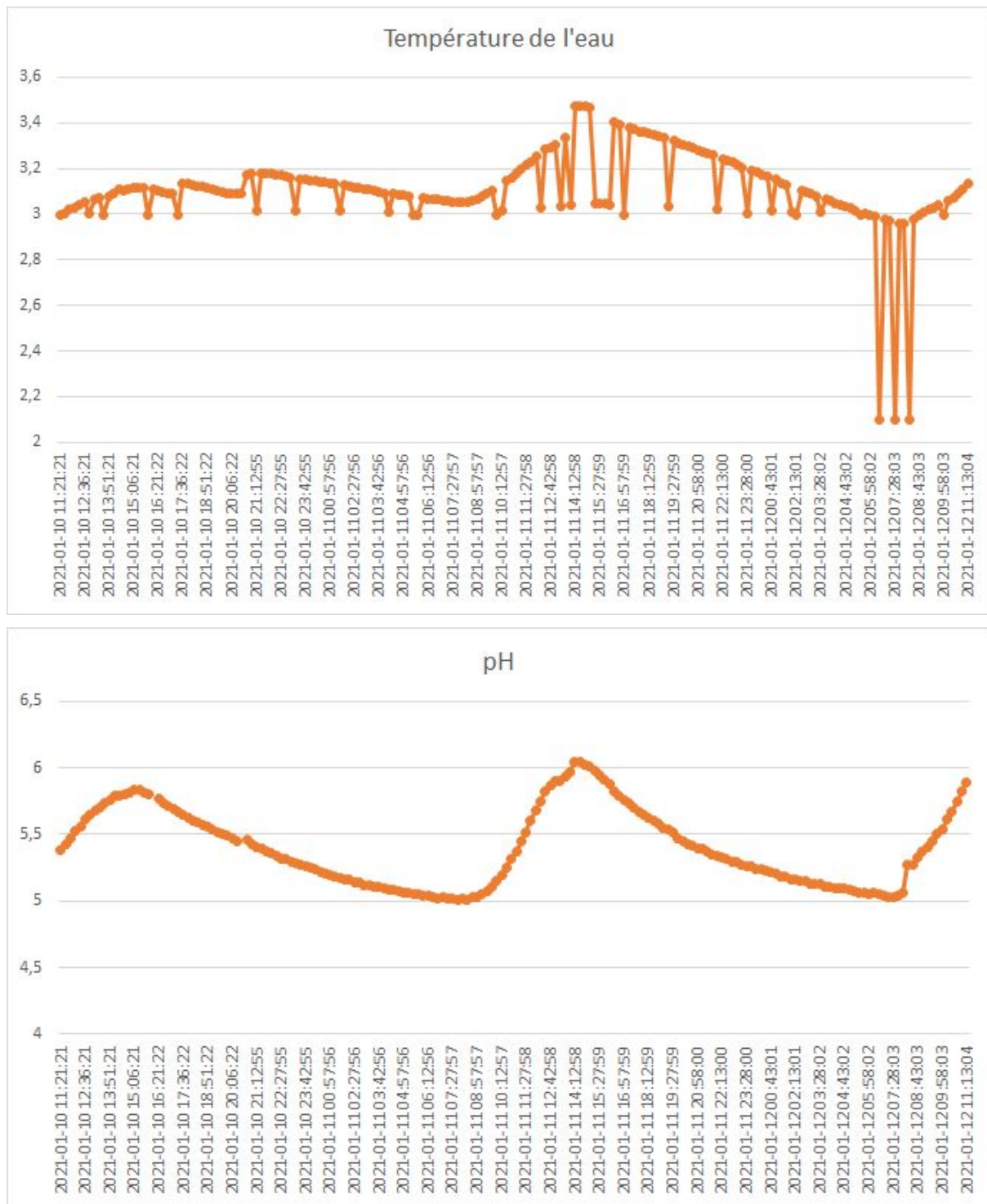
    headers = {'Authorization': 'key ttn-account-v2.2g6_wlwybeIJ_xhOIMsIUqR_Sip3_RC6IMgaEyuebg',
              'Content-type': 'application/json'}
    response = requests.get('https://capteurs_garage.data.thethingsnetwork.org/api/v2/query?last=2d', headers=headers).json()
    print("0")
    temp=[]
    ph=[]
    date=[]
    for i in range(len(response)):
        temp.append(response[i]['temperature_eau'])
        ph.append(response[i]['ph_eau'])
        date.append(response[i]['time'])
    response = str(response)
    print(temp,ph,date)
    print("1")
    f=open('Groupe2mp3.csv','w')
    f.write(date)
    f.write(temp)
    f.write(ph)
    f.close()
    response = response.replace("\n", "")

    obj = json.loads(response)
    dict=json.dumps(obj)
    print("2")
    dict_json=json.loads(dict)
    print("2b")
    id_json=json.dumps(dict_json[0])
  
```

Plus précisément, le programme implémenté permet de créer un fichier CSV sur la Raspberry, appelé “Groupe2mp3.csv”, avec les données de température et de pH associées à la date, pendant 2 jours. On vérifie ci-dessous que le fichier est bien créé avec 3 listes de données : la date, la température et le pH.

```

pi@raspberrypi ~
GNU nano 3.2 Groupe2mp3.csv
[{"2021-01-10T11:21:21.0171965298", "2021-01-10T11:36:21.0895001616", "2021-01-10T11:51:21.1124371448", "2021-01-10T12:06:21.1137399328", "2021-01-10T12:21:21.114592166", "2021-01-10T12:36:21.11539999999999998", "2021-01-10T12:51:21.11620000000000002", "2021-01-10T13:06:21.11700000000000004", "2021-01-10T13:21:21.11780000000000006", "2021-01-10T13:36:21.11860000000000008", "2021-01-10T13:51:21.11940000000000010", "2021-01-10T14:06:21.12020000000000012", "2021-01-10T14:21:21.12100000000000014", "2021-01-10T14:36:21.12180000000000016", "2021-01-10T14:51:21.12260000000000018", "2021-01-10T15:06:21.12340000000000020", "2021-01-10T15:16:21.12420000000000022", "2021-01-10T15:31:21.12500000000000024", "2021-01-10T15:46:21.12580000000000026", "2021-01-10T16:01:21.12660000000000028", "2021-01-10T16:16:21.12740000000000030", "2021-01-10T16:31:21.12820000000000032", "2021-01-10T16:46:21.12900000000000034", "2021-01-10T17:01:21.12980000000000036", "2021-01-10T17:16:21.13060000000000038", "2021-01-10T17:31:21.13140000000000040", "2021-01-10T17:46:21.13220000000000042", "2021-01-10T18:01:21.13300000000000044", "2021-01-10T18:16:21.13380000000000046", "2021-01-10T18:31:21.13460000000000048", "2021-01-10T18:46:21.13540000000000050", "2021-01-10T19:01:21.13620000000000052", "2021-01-10T19:16:21.13700000000000054", "2021-01-10T19:31:21.13780000000000056", "2021-01-10T19:46:21.13860000000000058", "2021-01-10T20:01:21.13940000000000060", "2021-01-10T20:16:21.14020000000000062", "2021-01-10T20:31:21.14100000000000064", "2021-01-10T20:46:21.14180000000000066", "2021-01-10T21:01:21.14260000000000068", "2021-01-10T21:16:21.14340000000000070", "2021-01-10T21:31:21.14420000000000072", "2021-01-10T21:46:21.14500000000000074", "2021-01-10T22:01:21.14580000000000076", "2021-01-10T22:16:21.14660000000000078", "2021-01-10T22:31:21.14740000000000080", "2021-01-10T22:46:21.14820000000000082", "2021-01-10T23:01:21.14900000000000084", "2021-01-10T23:16:21.14980000000000086", "2021-01-10T23:31:21.15060000000000088", "2021-01-10T23:46:21.15140000000000090", "2021-01-11T00:01:21.15220000000000092", "2021-01-11T00:16:21.15300000000000094", "2021-01-11T00:31:21.15380000000000096", "2021-01-11T00:46:21.15460000000000098", "2021-01-11T01:01:21.15540000000000100", "2021-01-11T01:16:21.15620000000000102", "2021-01-11T01:31:21.15700000000000104", "2021-01-11T01:46:21.15780000000000106", "2021-01-11T02:01:21.15860000000000108", "2021-01-11T02:16:21.15940000000000110", "2021-01-11T02:31:21.16020000000000112", "2021-01-11T02:46:21.16100000000000114", "2021-01-11T03:01:21.16180000000000116", "2021-01-11T03:16:21.16260000000000118", "2021-01-11T03:31:21.16340000000000120", "2021-01-11T03:46:21.16420000000000122", "2021-01-11T04:01:21.16500000000000124", "2021-01-11T04:16:21.16580000000000126", "2021-01-11T04:31:21.16660000000000128", "2021-01-11T04:46:21.16740000000000130", "2021-01-11T05:01:21.16820000000000132", "2021-01-11T05:16:21.16900000000000134", "2021-01-11T05:31:21.16980000000000136", "2021-01-11T05:46:21.17060000000000138", "2021-01-11T06:01:21.17140000000000140", "2021-01-11T06:16:21.17220000000000142", "2021-01-11T06:31:21.17300000000000144", "2021-01-11T06:46:21.17380000000000146", "2021-01-11T07:01:21.17460000000000148", "2021-01-11T07:16:21.17540000000000150", "2021-01-11T07:31:21.17620000000000152", "2021-01-11T07:46:21.17700000000000154", "2021-01-11T08:01:21.17780000000000156", "2021-01-11T08:16:21.17860000000000158", "2021-01-11T08:31:21.17940000000000160", "2021-01-11T08:46:21.18020000000000162", "2021-01-11T09:01:21.18100000000000164", "2021-01-11T09:16:21.18180000000000166", "2021-01-11T09:31:21.18260000000000168", "2021-01-11T09:46:21.18340000000000170", "2021-01-11T10:01:21.18420000000000172", "2021-01-11T10:16:21.18500000000000174", "2021-01-11T10:31:21.18580000000000176", "2021-01-11T10:46:21.18660000000000178", "2021-01-11T11:01:21.18740000000000180", "2021-01-11T11:16:21.18820000000000182", "2021-01-11T11:31:21.18900000000000184", "2021-01-11T11:46:21.18980000000000186", "2021-01-11T12:01:21.19060000000000188", "2021-01-11T12:16:21.19140000000000190", "2021-01-11T12:31:21.19220000000000192", "2021-01-11T12:46:21.19300000000000194", "2021-01-11T13:01:21.19380000000000196", "2021-01-11T13:16:21.19460000000000198", "2021-01-11T13:31:21.19540000000000200", "2021-01-11T13:46:21.19620000000000202", "2021-01-11T14:01:21.19700000000000204", "2021-01-11T14:16:21.19780000000000206", "2021-01-11T14:31:21.19860000000000208", "2021-01-11T14:46:21.19940000000000210", "2021-01-11T15:01:21.20020000000000212", "2021-01-11T15:16:21.20100000000000214", "2021-01-11T15:31:21.20180000000000216", "2021-01-11T15:46:21.20260000000000218", "2021-01-11T16:01:21.20340000000000220", "2021-01-11T16:16:21.20420000000000222", "2021-01-11T16:31:21.20500000000000224", "2021-01-11T16:46:21.20580000000000226", "2021-01-11T17:01:21.20660000000000228", "2021-01-11T17:16:21.20740000000000230", "2021-01-11T17:31:21.20820000000000232", "2021-01-11T17:46:21.20900000000000234", "2021-01-11T18:01:21.20980000000000236", "2021-01-11T18:16:21.21060000000000238", "2021-01-11T18:31:21.21140000000000240", "2021-01-11T18:46:21.21220000000000242", "2021-01-11T19:01:21.21300000000000244", "2021-01-11T19:16:21.21380000000000246", "2021-01-11T19:31:21.21460000000000248", "2021-01-11T19:46:21.21540000000000250", "2021-01-11T20:01:21.21620000000000252", "2021-01-11T20:16:21.21700000000000254", "2021-01-11T20:31:21.21780000000000256", "2021-01-11T20:46:21.21860000000000258", "2021-01-11T21:01:21.21940000000000260", "2021-01-11T21:16:21.22020000000000262", "2021-01-11T21:31:21.22100000000000264", "2021-01-11T21:46:21.22180000000000266", "2021-01-11T22:01:21.22260000000000268", "2021-01-11T22:16:21.22340000000000270", "2021-01-11T22:31:21.22420000000000272", "2021-01-11T22:46:21.22500000000000274", "2021-01-11T23:01:21.22580000000000276", "2021-01-11T23:16:21.22660000000000278", "2021-01-11T23:31:21.22740000000000280", "2021-01-11T23:46:21.22820000000000282", "2021-01-12T00:01:21.22900000000000284", "2021-01-12T00:16:21.22980000000000286", "2021-01-12T00:31:21.23060000000000288", "2021-01-12T00:46:21.23140000000000290", "2021-01-12T01:01:21.23220000000000292", "2021-01-12T01:16:21.23300000000000294", "2021-01-12T01:31:21.23380000000000296", "2021-01-12T01:46:21.23460000000000298", "2021-01-12T02:01:21.23540000000000300", "2021-01-12T02:16:21.23620000000000302", "2021-01-12T02:31:21.23700000000000304", "2021-01-12T02:46:21.23780000000000306", "2021-01-12T03:01:21.23860000000000308", "2021-01-12T03:16:21.23940000000000310", "2021-01-12T03:31:21.24020000000000312", "2021-01-12T03:46:21.24100000000000314", "2021-01-12T04:01:21.24180000000000316", "2021-01-12T04:16:21.24260000000000318", "2021-01-12T04:31:21.24340000000000320", "2021-01-12T04:46:21.24420000000000322", "2021-01-12T05:01:21.24500000000000324", "2021-01-12T05:16:21.24580000000000326", "2021-01-12T05:31:21.24660000000000328", "2021-01-12T05:46:21.24740000000000330", "2021-01-12T06:01:21.24820000000000332", "2021-01-12T06:16:21.24900000000000334", "2021-01-12T06:31:21.24980000000000336", "2021-01-12T06:46:21.25060000000000338", "2021-01-12T07:01:21.25140000000000340", "2021-01-12T07:16:21.25220000000000342", "2021-01-12T07:31:21.25300000000000344", "2021-01-12T07:46:21.25380000000000346", "2021-01-12T08:01:21.25460000000000348", "2021-01-12T08:16:21.25540000000000350", "2021-01-12T08:31:21.25620000000000352", "2021-01-12T08:46:21.25700000000000354", "2021-01-12T09:01:21.25780000000000356", "2021-01-12T09:16:21.25860000000000358", "2021-01-12T09:31:21.25940000000000360", "2021-01-12T09:46:21.26020000000000362", "2021-01-12T10:01:21.26100000000000364", "2021-01-12T10:16:21.26180000000000366", "2021-01-12T10:31:21.26260000000000368", "2021-01-12T10:46:21.26340000000000370", "2021-01-12T11:01:21.26420000000000372", "2021-01-12T11:16:21.26500000000000374", "2021-01-12T11:31:21.26580000000000376", "2021-01-12T11:46:21.26660000000000378", "2021-01-12T12:01:21.26740000000000380", "2021-01-12T12:16:21.26820000000000382", "2021-01-12T12:31:21.26900000000000384", "2021-01-12T12:46:21.26980000000000386", "2021-01-12T13:01:21.27060000000000388", "2021-01-12T13:16:21.27140000000000390", "2021-01-12T13:31:21.27220000000000392", "2021-01-12T13:46:21.27300000000000394", "2021-01-12T14:01:21.27380000000000396", "2021-01-12T14:16:21.27460000000000398", "2021-01-12T14:31:21.27540000000000400", "2021-01-12T14:46:21.27620000000000402", "2021-01-12T15:01:21.27700000000000404", "2021-01-12T15:16:21.27780000000000406", "2021-01-12T15:31:21.27860000000000408", "2021-01-12T15:46:21.27940000000000410", "2021-01-12T16:01:21.28020000000000412", "2021-01-12T16:16:21.28100000000000414", "2021-01-12T16:31:21.28180000000000416", "2021-01-12T16:46:21.28260000000000418", "2021-01-12T17:01:21.28340000000000420", "2021-01-12T17:16:21.28420000000000422", "2021-01-12T17:31:21.28500000000000424", "2021-01-12T17:46:21.28580000000000426", "2021-01-12T18:01:21.28660000000000428", "2021-01-12T18:16:21.28740000000000430", "2021-01-12T18:31:21.28820000000000432", "2021-01-12T18:46:21.28900000000000434", "2021-01-12T19:01:21.28980000000000436", "2021-01-12T19:16:21.29060000000000438", "2021-01-12T19:31:21.29140000000000440", "2021-01-12T19:46:21.29220000000000442", "2021-01-12T20:01:21.29300000000000444", "2021-01-12T20:16:21.29380000000000446", "2021-01-12T20:31:21.29460000000000448", "2021-01-12T20:46:21.29540000000000450", "2021-01-12T21:01:21.29620000000000452", "2021-01-12T21:16:21.29700000000000454", "2021-01-12T21:31:21.29780000000000456", "2021-01-12T21:46:21.29860000000000458", "2021-01-12T22:01:21.29940000000000460", "2021-01-12T22:16:21.30020000000000462", "2021-01-12T22:31:21.30100000000000464", "2021-01-12T22:46:21.30180000000000466", "2021-01-12T23:01:21.30260000000000468", "2021-01-12T23:16:21.30340000000000470", "2021-01-12T23:31:21.30420000000000472", "2021-01-12T23:46:21.30500000000000474", "2021-01-13T00:01:21.30580000000000476", "2021-01-13T00:16:21.30660000000000478", "2021-01-13T00:31:21.30740000000000480", "2021-01-13T00:46:21.30820000000000482", "2021-01-13T01:01:21.30900000000000484", "2021-01-13T01:16:21.30980000000000486", "2021-01-13T01:31:21.31060000000000488", "2021-01-13T01:46:21.31140000000000490", "2021-01-13T02:01:21.31220000000000492", "2021-01-13T02:16:21.31300000000000494", "2021-01-13T02:31:21.31380000000000496", "2021-01-13T02:46:21.31460000000000498", "2021-01-13T03:01:21.31540000000000500", "2021-01-13T03:16:21.31620000000000502", "2021-01-13T03:31:21.31700000000000504", "2021-01-13T03:46:21.31780000000000506", "2021-01-13T04:01:21.31860000000000508", "2021-01-13T04:16:21.31940000000000510", "2021-01-13T04:31:21.32020000000000512", "2021-01-13T04:46:21.32100000000000514", "2021-01-13T05:01:21.32180000000000516", "2021-01-13T05:16:21.32260000000000518", "2021-01-13T05:31:21.32340000000000520", "2021-01-13T05:46:21.32420000000000522", "2021-01-13T06:01:21.32500000000000524", "2021-01-13T06:16:21.32580000000000526", "2021-01-13T06:31:21.32660000000000528", "2021-01-13T06:46:21.32740000000000530", "2021-01-13T07:01:21.32820000000000532", "2021-01-13T07:16:21.32900000000000534", "2021-01-13T07:31:21.32980000000000536", "2021-01-13T07:46:21.33060000000000538", "2021-01-13T08:01:21.33140000000000540", "2021-01-13T08:16:21.33220000000000542", "2021-01-13T08:31:21.33300000000000544", "2021-01-13T08:46:21.33380000000000546", "2021-01-13T09:01:21.33460000000000548", "2021-01-13T09:16:21.33540000000000550", "2021-01-13T09:31:21.33620000000000552", "2021-01-13T09:46:21.33700000000000554", "2021-01-13T10:01:21.33780000000000556", "2021-01-13T10:16:21.33860000000000558", "2021-01-13T10:31:21.33940000000000560", "2021-01-13T10:46:21.34020000000000562", "2021-01-13T11:01:21.34100000000000564", "2021-01-13T11:16:21.34180000000000566", "2021-01-13T11:31:21.34260000000000568", "2021-01-13T11:46:21.34340000000000570", "2021-01-13T12:01:21.34420000000000572", "2021-01-13T12:16:21.34500000000000574", "2021-01-13T12:31:21.34580000000000576", "2021-01-13T12:46:21.34660000000000578", "2021-01-13T13:01:21.34740000000000580", "2021-01-13T13:16:21.34820000000000582", "2021-01-13T13:31:21.34900000000000584", "2021-01-13T13:46:21.34980000000000586", "2021-01-13T14:01:21.35060000000000588", "2021-01-13T14:16:21.35140000000000590", "2021-01-13T14:31:21.35220000000000592", "2021-01-13T
```

Nous avons observé plusieurs valeurs aberrantes que nous avons décidé de retirer ou de corriger. On note cependant sur la courbe de la température que certains points se dégagent toujours grandement de la tendance générale suivie, malgré le fait que les valeurs ne soient pas aberrantes.

Conclusion

Dans un premier temps, nous avons eu quelques difficultés de compréhension des outils à utiliser entre eux car nous n'avions jamais eu l'occasion de manipuler et programmer afin de récupérer à distance des données d'un capteur. Ce projet était donc intéressant pour apprendre de nouvelles choses sur les ICT, et notamment la transmission et l'exploitation de données, qui sont des sujets amenés à jouer un rôle important dans les prochaines années. Nous avons également eu beaucoup de problèmes pour se connecter au réseau de la Raspberry PI, ce qui fait que nous n'avons pas eu tout le temps escompté pour manipuler les données récupérées.

Bibliographie

- <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/>
- article de Sylvain lors de son stage à l'ENSE3
<http://miniprojets.net/index.php/2019/07/23/monitoring-dune-serre-du-capteur-jusquau-serveur/>
- <https://www.carnetdumaker.net/articles/utiliser-un-capteur-de-temperature-et-dhumidite-dht11-dht22-avec-une-carte-arduino-genuino/#:~:text=Un%20capteur%20DHT11%20ou%20DHT22,files%20pour%20c%C3%A2bler%20notre%20montage.>