

Capteur de niveau dans une cuve - Paramétrage LORAWAN

Présentation : On dispose d'un capteur à ultrason pour mesurer la hauteur d'un liquide dans une cuve. Celle cuve se situe à plusieurs centaines de mètre des premiers bâtiments. Il n'y a pas d'accès Wifi ou de liaison filaire.

On choisit une liaison sans fil, de type LORAWAN, dont la distance de transmission peut atteindre 15km en milieu dégagé.

Le capteur a les caractéristiques suivantes :



Specification

- Operating Voltage: 3.3V-5v
- Average Current: $\leq 15\text{mA}$
- Blind Zone Distance: 0-28cm
- Ranging Distance for Flat Object: 28-750cm
- Output: UART
- Response Time: 100ms
- Probe Center Frequency: $40\text{K} \pm 1.0\text{K}$
- Operating Temperature: $-15 \sim 60^\circ\text{C}$
- Storage Temperature: $-25 \sim 80^\circ\text{C}$
- Protection Rate: IP67

Ce capteur a une plage de fonctionnement de 28 à 750 cm. La sortie est de type UART (série asynchrone). Ce capteur est associé à un module LORAWAN. Les données sont transmises au serveur TTN.

1 – Paramétrage du capteur sur LORAWAN

On demande de rédiger un compte rendu, avec les réponses aux questions, des copies d'écran etc...

Le paramétrage du capteur a été réalisé au préalable sur le serveur TTN.

- ⇒ Se rendre sur le site de TTN : <https://www.thethingsnetwork.org>
- ⇒ Se loguer avec le compte : guy.colin@ac-lille.fr et mot de passe : ●●●●●● (à demander au prof).
- ⇒ Se rendre dans le menu « Console ».
- ⇒ Choisir la zone Europe
- ⇒ Dans le menu Gateways, vérifier que la passerelle carnot est connectée et relever son numéro EUI.
- ⇒ Dans le menu Applications, vérifier la présence de l'application ultra-son (ID : ultra-son-carnot).
- ⇒ Dans l'application ultra-son, relever le nombre de « End devices » déclarés
 - l'ID
 - le DevEUI
 - le JoinEUI

- l'AppKey
- le Device address

⇒ Dans le menu Payload formatters / Uplink, faire une copie d'écran du format (Formatter type et Formatter code).

⇒ Dans le menu Live data, vérifier la réception de données en provenance du capteur.

⇒ Indiquer la périodicité de transmission des données (faible ici pour le TP).

Le capteur transmet la distance en millimètres. L'information est transmise sur 2 octets : MSB en premier, puis LSB.

Exemple : si le capteur transmet les 2 octets [7,67] (codage en décimal), la valeur correspondante en mm sera : $7 * 256 + 67 = 1859$ mm.

⇒ Cliquer sur une trame transmise au serveur TTN, et observer le fichier au format JSON pour chaque trame

⇒ Dans le menu Integrations/MQTT , copier l'adresse du serveur public (port 1883), le Username. Le Password sera communiqué par l'ENT.

2 – Récupération des données dans Node Red

Sous Node Red, créer le Flow suivant :



Edit mqtt in node

Delete Cancel

Properties

Server: eu1.cloud.thethings.network:1883

Topic: v3/ultra-son-carnot@ttn/devices/eui-c0ee40ffff391923/up

QoS: 0

Output: a parsed JSON object

Name: Name

Edit mqtt in node > Edit mqtt-broker node

Delete Cancel

Properties

Name: Name

Connection

Server: eu1.cloud.thethings.network Port: 1883

Use TLS

Protocol: MQTT V3.1.1

Client ID: Leave blank for auto generated

Keep Alive: 60

Session Use clean session

Edit mqtt in node > **Edit mqtt-broker node**

Delete

Properties

Name

Connection Security Mess

Username ultra-son-carnot@ttn

Password

Edition de la fonction est la suivante (donnée ici en partie, à compléter)

Remarque : Il est possible de récupérer le chemin des données JSON avec la fonction Copy path .

```

c0ee40ffff391923/up : msg.payload : Object
  object
    end_device_ids: object
    correlation_ids: array[7]
    received_at:
      "2023-03-10T10:04:06.067788614Z"
    uplink_message: object
      session_key_id:
        "AYbKr/9SUU46mUSB5g5/kQ=="
      f_port: 1
      f_cnt: 214
      frm_payload: "B3c="
      decoded_payload: object
        bytes: array[2]
          0: 7
          1: 119
      rx_metadata: array[1]
      settings: object
  
```

Copy path

Edit function node

Delete Cancel

Properties

Name

Setup On Start **On Message** On Stop

```

1
2 var distance = msg.payload.uplink_message.decoded_payload.bytes[0] * 256 + msg.payload.uplink_message.decoded_payload.bytes[1];
3 msg.payload={"distance":distance};
4 return msg;
  
```

La fonction doit retourner la distance en mm, sous un format JSON comme ci-dessous :

```

28/11/2022 16:41:07 node: 9df8e0ca.2ad76
v3/ultra-son-carnot@ttn/devices/eui-c0ee40ffff391923/up :
  { distance: 1863 }
  
```

⇒ Ajouter un node Influxdb pour enregistrer la distance dans la base de données TPTSMI, créée lors d'un précédent TP.

⇒ Ajouter une fonction sous Node Red qui puisse extraire du fichier JSON, l'adresse (dev_addr) et l'afficher sous la forme suivante (à faire constater au prof).

```
28/11/2022 17:13:56 node: 9264efe3.d0706
```

```
v3/ultra-son-carnot@ttn/devices/eui-c0ee40fff391923/up :
```

```
▶ { dev_addr: "260B493F" }
```