

Documentation technique mise à disposition :

- Notice d'utilisation du capteur VBB001
- Document technique du capteur VBB001
- La description IODD du capteur VBB001

Fichier JSON de programmation NODE RED mis à disposition. Il sera importé sous NODE RED.

La programmation NODERED est accessible sous l'explorateur internet avec le port 1880 :

- En local ⇒ localhost :1880
- Sur poste distant ⇒ IP_Poste :1880

L'interface utilisateur est accessible en ajoutant /ui ⇒ IP_Poste :1880/ui

Le capteur IOLINK VBB001 est branché à un maître IOLINK AL1320, sur le port 1.

Les logiciels mis à disposition : Ethernet Device Configuration – Dipiscan – IoT-Core_Assistant_V1.2 – LR DEVICE – MONEO Configure

La configuration du maître est accessible via la page : IP_Maître/web/subscribe.

Le serveur MQTT choisi sera celui installé sur une Raspberry dont l'adresse IP est : 172.16.6.61

1 - Paramétrage MQTT

⇒ Faire une recherche de l'adresse IP du Maître, et la noter ci-dessous.

⇒ Paramétrer la diffusion MQTT sur le maître pour avoir une diffusion des données process du capteur VBB001 toutes les 10 secondes. Préciser ci-dessous l'adresse du BROKER, le TOPIC, la cause de l'événement choisi et le paramétrage correspondant, les données et le port interrogé.

⇒ Paramétrer le client MQTT sous NODE RED pour recevoir les données.

⇒ Ajouter un NODE DEBUG pour vérifier la bonne réception du fichier JSON.

⇒ Faire valider par le prof .

2 – Câblage

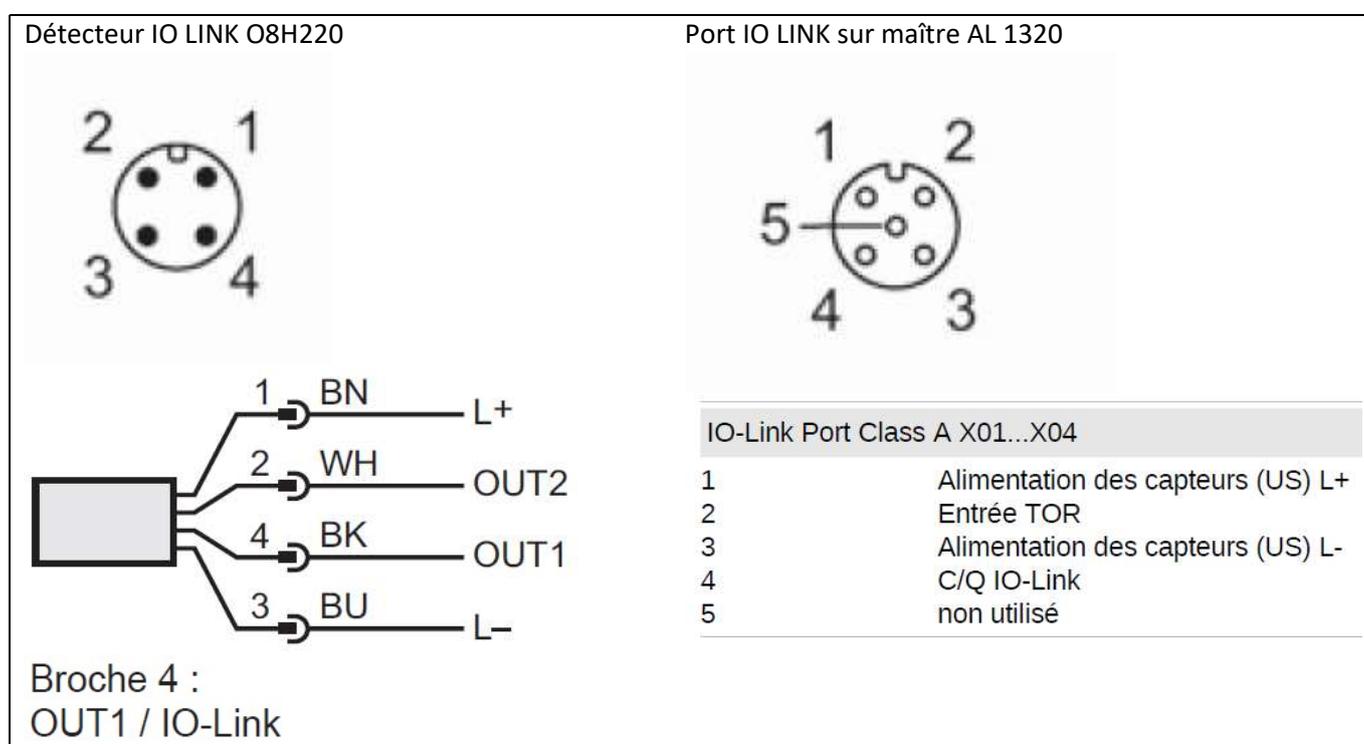
A partir des notices du capteur, répondre aux questions suivantes.

⇒ Indiquer la tension d'alimentation du capteur et le courant maximal qu'il peut fournir

1^{er} Cas : Le capteur est relié à un maître IO LINK

Dans ce cas la sortie du détecteur fonctionne automatiquement en ligne de communication IO LINK.

⇒ Compléter le câblage ci-dessous entre les 2 connecteurs

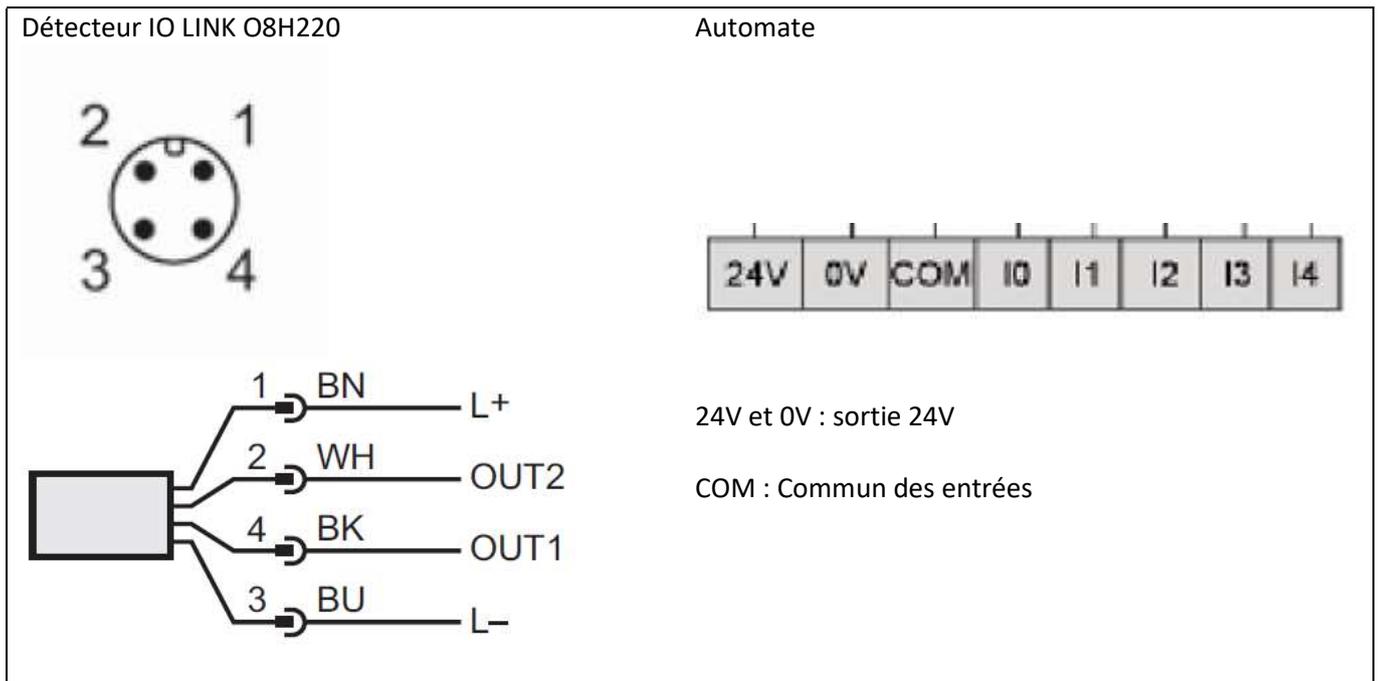


2^{ème} Cas : Le détecteur est relié à un automate programmable

Dans ce cas les sorties du capteur fonctionnent automatiquement en sortie TOR (s'il n'y a pas de communication série sur la ligne IO LINK).

⇒ Déterminer le(s) mode(s) de fonctionnement possible(s) sur les 2 sorties OUT1 et OUT2 du capteur (NPN ou PNP).

⇒ Compléter le câblage du détecteur sur l'entrée I0 de l'automate (COM également) pour une sortie PNP.



3 – Paramétrage du capteur

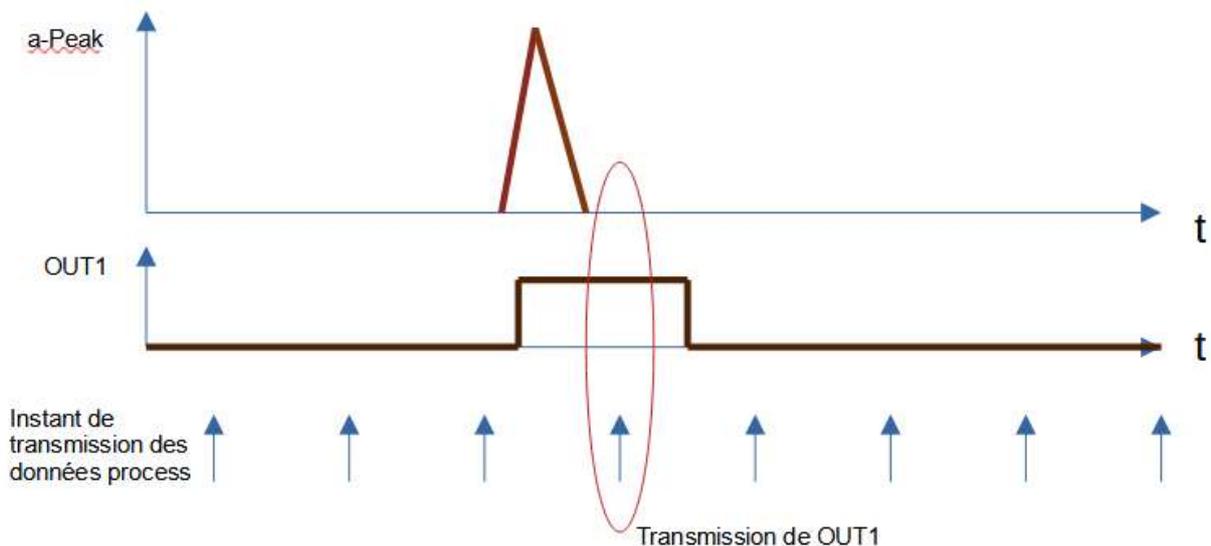
On souhaite détecter des chocs sur le capteur et transmettre une alarme (Mail ou Message Telegram) lorsqu'un seuil est dépassé.

Les réglages des seuils se font sur le capteur, et non sous NODE RED.

Les alarmes seront transmises en prenant en compte l'état des sorties OUT1 ou OUT2.

On veillera à mettre un délai au déclenchement supérieur à la valeur entre 2 instants de transmission des données process.

Dans l'exemple ci-dessous, un choc (sur a-Peak) est détecté par le capteur mais ne sera sans doute pas transmis s'il a lieu entre 2 instants de transmission. Par contre en mettant un délai au déclenchement de OUT1 supérieur à l'écart entre 2 transmissions, l'état de OUT1 sera obligatoirement transmis.



Problème posé : On souhaite détecter un choc supérieur à 50m/s^2 . Lors de cette détection, la sortie OUT1 passera à 1 pendant un temps supérieur au temps entre 2 transmissions des données process. Une alerte sera transmise dans ce cas.

⇒ Indiquer la plage de mesure de a-Peak (choc) en m/s^2

⇒ Indiquer la configuration de la sortie OUT1 (Fonction et normalement ouvert ou fermé)

⇒ Indiquer le paramètre à choisir pour activer la sortie OUT en cas de choc.

⇒ Indiquer le paramètre à valider pour activer la sortie OUT1

⇒ Indiquer le paramètre à modifier pour ajouter le retard au déclenchement nécessaire

⇒ Procéder aux réglages et aux tests. Vérifier la réception des alertes.

⇒ Faire valider par le prof .

Validation :

4 – Test des messages MQTT

On souhaite dans cette partie vérifier la transmission des messages MQTT.
On utilise pour cela le logiciel MQTT.fx

Réception des messages

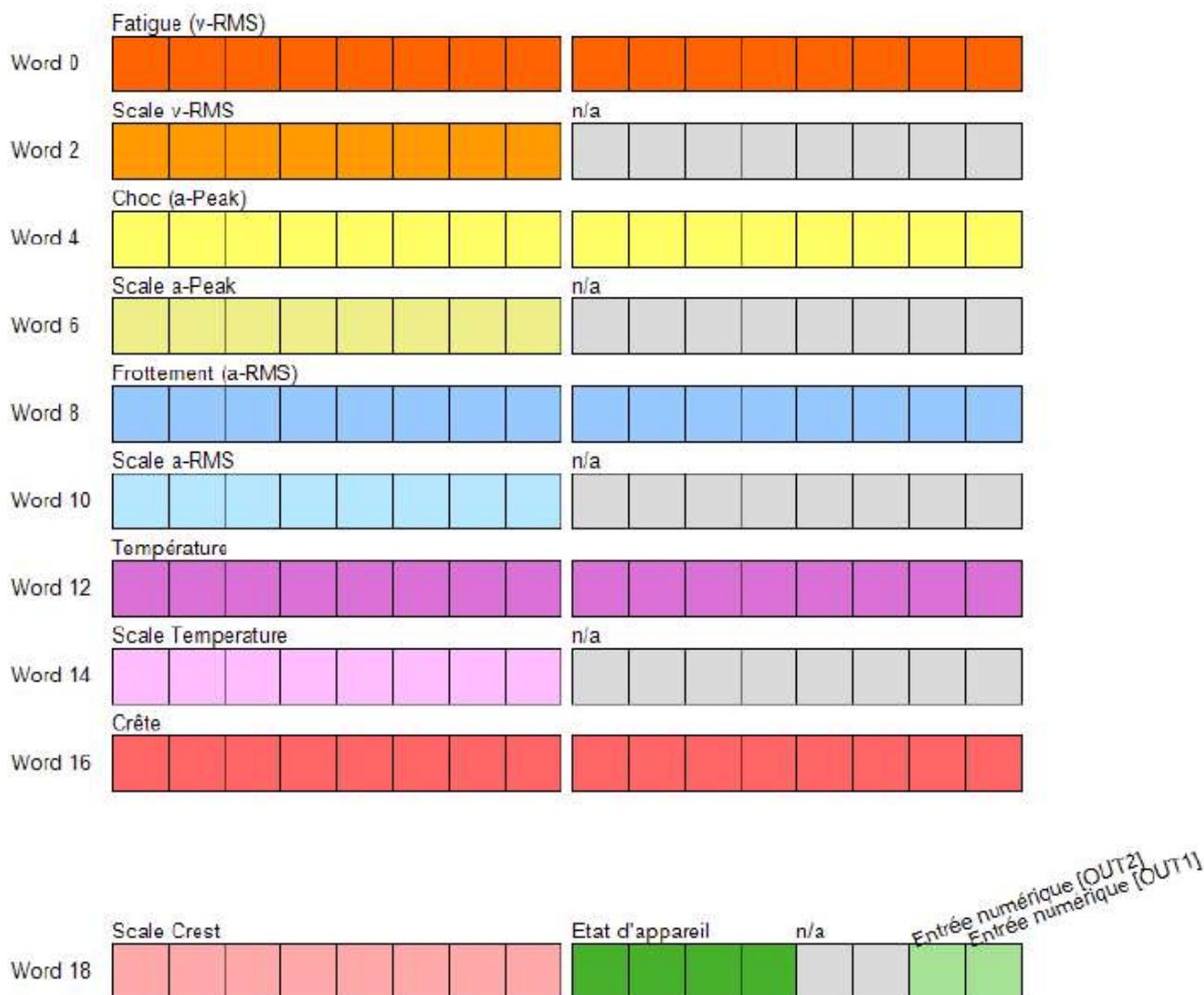
- ⇒ Lancer le logiciel MQTT.fx est le paramétrer pour recevoir les messages envoyés par le maître IOLINK
- ⇒ Sauvegarder un message reçu sous format JSON
- ⇒ Faire valider par le prof .

Validation :

Transmission de messages

Dans cette partie on souhaite simuler la transmission d'un message par le maître IO LINK pour vérifier la transmission d'alerte par exemple.
A l'aide du logiciel MQTT.fx on transmet une information OUT1 à 1.

Ci-dessous un extrait de la documentation IODD du capteur VBB001



⇒ Entourer le bit concerné par l'état de OUT1, sur la figure précédente.

Sur la figure ci-dessous est représentée la transmission du fichier JSON en MQTT du capteur VBB001.

```
{
  "code" : "event",
  "cid" : 557,
  "adr" : "/3axes/vbb001",
  "data" : {
    "eventno" : "2805",
    "srcurl" : "00-02-01-6E-FD-5E/timer[1]/counter/datachanged",
    "payload" : {
      "/timer[1]/counter" : {
        "code" : 200,
        "data" : 594
      },
      "/iolinkmaster/port[1]/iolinkdevice/pdin" : {
        "code" : 200,
        "data" : "0000FC000002FF000000FF000144FF000028FF00"
      }
    }
  }
}
```

⇒ Entourer le caractère qui permet de déterminer l'état de OUT1

⇒ Indiquer la valeur de ce caractère lorsque OUT1 est actif seul

⇒ Indiquer la valeur de ce caractère lorsque OUT1 et OUT2 sont actifs

⇒ Publier un message MQTT à l'aide du logiciel MQTT.fx pour simuler le passage de OUT1 à 1 et vérifier la réception d'une alerte. (Utiliser pour cela le fichier enregistré précédemment).

⇒ Faire valider par le prof .

Validation :