

Documentation technique mise à disposition :

- Notice d'utilisation du capteur TV7105
- Document technique du capteur TV7105
- La description IODD du capteur TV7105

Fichier JSON de programmation NODE RED est mis à disposition. Il sera importé sous NODE RED. La palette « node-red-node-email » devra être installée.

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 TV7105 est branché à un maître IOLINK AL1320.

Un fer à souder mis à disposition permettra de faire monter le capteur en température. On veillera à ne pas dépasser les 100°C.

Les logiciels mis à disposition : Ethernet Device Configuration – Dipiscan – MQTT.fx

Le logiciel ifm monéo configure est installé sur quelques postes. Le logiciel LR DEVICE, en version démo, est installé sur l'ensemble des postes.

La configuration des notifications MQTT du maître est accessible via la page : IP_Maître/web/subscribe. Elle est donnée ci-dessous

Consumer ID	Channel	Broker/Server	Event	Data	Duration	Unsubscribe
555	mqtt	172.16.6.61:1883/master1/tv7105	00-02-01-6E-FD-5E/timer[1]/counter/datachanged	00-02-01-6e-fd-5e/iolinkmaster/port[1]/iolinkdevice/pdin	lifetime	x

Parameter	Value	Type	Namespace	Encoding	Valuation
interval	5000	number	json	integer	off: 0, min: 50, max: 2147483647
unit	ms	string	json	utf-8	

Le serveur MQTT choisi sera celui installé sur une Raspberry connectée sur le réseau du lycée.

1 - Paramétrage MQTT

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

⇒ A l'aide des paramètres de configuration des notifications MQTT, préciser ci-dessous l'adresse du BROKER et le port utilisé, le TOPIC, la cause de l'événement choisi et le paramétrage correspondant, les données et le port interrogé.

Adresse du broker :
Port du broker :
Topic :
Elément qui provoque l'envoi de la notification et paramétrage :
Données transmises et port IO LINK correspondant :

⇒ A l'aide du logiciel MQTT.fx, après paramétrage de celui-ci, vérifier la réception des notifications transmises par le broker.

⇒ Faire valider par le prof.

⇒ 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

On souhaite, en plus de la transmission des données process du capteur, allumer un voyant 24V lorsque la température dépasse 70°C et l'éteindre lorsque la température redescend en dessous de 60°C.

Le capteur TV7105 dispose de 2 sorties (OUT1 et OUT2)

⇒ Consulter la notice d'utilisation du capteur et préciser la sortie qui est utilisée pour transmettre les données IOLINK.

On utilisera la sortie OUT2 pour allumer le voyant.

Le voyant choisi affiche une consommation électrique de 18mA.

⇒ Consulter la documentation technique du capteur (TV7105-00-FR) et vérifier que la sortie du capteur est capable d'alimenter le voyant sans passer par un relais intermédiaire.

Les sorties du capteur peuvent être configurées en mode NPN ou PNP.

⇒ En s'inspirant des exemples de circuits de la notice d'utilisation du capteur, tracer les 2 schémas de câble possible dans notre cas et préciser le mode de configuration des sorties pour les 2 cas (NPN ou PNP).

On retiendra la version sortie PNP.

3 – Essais

Le capteur dispose de plusieurs fonctions de commutation possible (voir notice d'utilisation, fonction commutation).

⇒ Indiquer la fonction de commutation à choisir, les seuils et le mode (no ou nc) à programmer.

Les variables FOU1 et FOU2 permettent de définir le comportement des sorties OUT1 et OUT2.

⇒ A l'aide de la description IODD du capteur TV7105, indiquer les valeurs à appliquer sur ces 2 paramètres pour valider la sortie OUT2 uniquement.

Plage de fonctionnement et résolution

⇒ A l'aide de la description IODD du capteur TV7105, indiquer la plage de fonction du capteur et la résolution de la mesure (en °C).

⇒ A l'aide de la description IODD du capteur TV7105, indiquer la plage de réglage des seuils d'enclenchement et de déclenchement. Préciser la résolution de ces seuils.

Paramétrage et essais

⇒ Procéder au paramétrage du capteur et réaliser les essais avec un fer à souder.

⇒ Vérifier les seuils de basculement de la sortie OUT2.

⇒ Modifier, sous NODERED, le destinataire du mail d'alerte et tester la transmission et la réception du mail.

On souhaite mettre un délai au déclenchement (lorsque que la température descend en dessous du seuil bas).

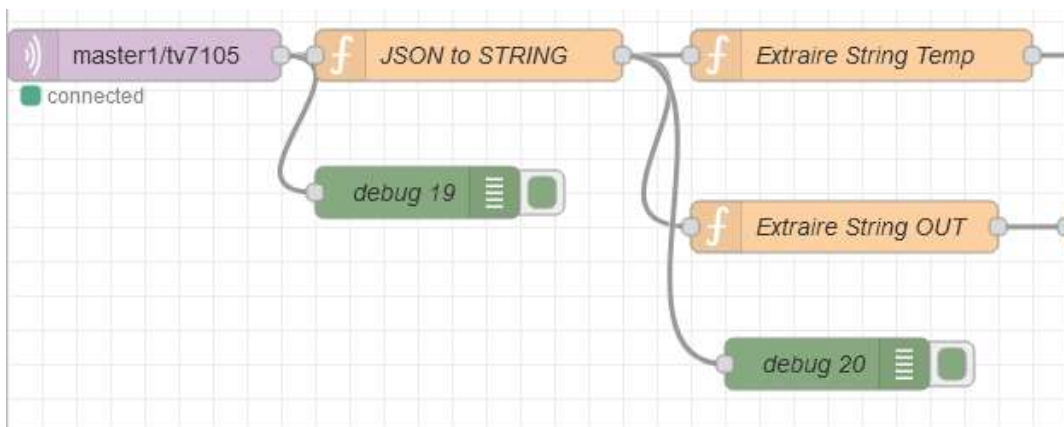
⇒ A partir des documentations du capteur, déterminer le paramètre qui permet d'ajouter un retard au déclenchement de la sortie OUT2.

⇒ Déterminer la plage de réglage de ce paramètre.

⇒ Régler une temporisation au déclenchement de 15 secondes et tester par un essai.

4 – Justifications des traitements sous NodeRed

⇒ Ajouter 2 NODE DEBUG comme ci-dessous et observer les données reçues dans la fenêtre DEBUG.



⇒ En déduire le traitement réalisé par la fonction JSON to STRING

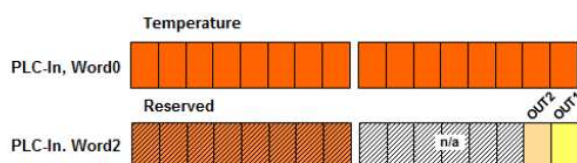
En sortie de la fonction JSON to STRING, on obtient une chaîne de caractères représentant les données process sous forme hexadécimale.

Dans le cas où en sortie de cette fonction on obtient la chaîne de caractère suivante : **010BFF00**,

⇒ Déterminer la valeur de la température, et les états des sorties OUT1 et OUT2 dans ce cas.

Données process Longueur totale en bits = 32
(Données process entrée)

Nom	Description	Type de données	Longueur en bits	Plage de valeurs	Gradient	Offset	Unité
Température	Température actuelle	IntegerT	16	-500 to 1500 (-32760) UL (32760) OL (32764) NoData	0.1	0	°C
OUT2	Etat dépend de [OU2]	BooleanT		(false) inactif (true) actif			
OUT1	Etat dépend de [OU1]	BooleanT		(false) inactif (true) actif			



⇒ Indiquer quelle serait la chaîne de caractère envoyée dans le cas où la température est de 28,5°C et les 2 sorties OUT1 et OUT2 à 1 logique.

⇒ De la même manière, placer un NODE DEBUG en sortie de la fonction « Extraire String Temp » et en déduire le traitement réalisé par cette fonction.

⇒ De la même manière, placer un NODE DEBUG en sortie de la fonction « String to Temp » et en déduire le traitement réalisé par cette fonction. Préciser quelle fonction javascript permet un tel traitement.