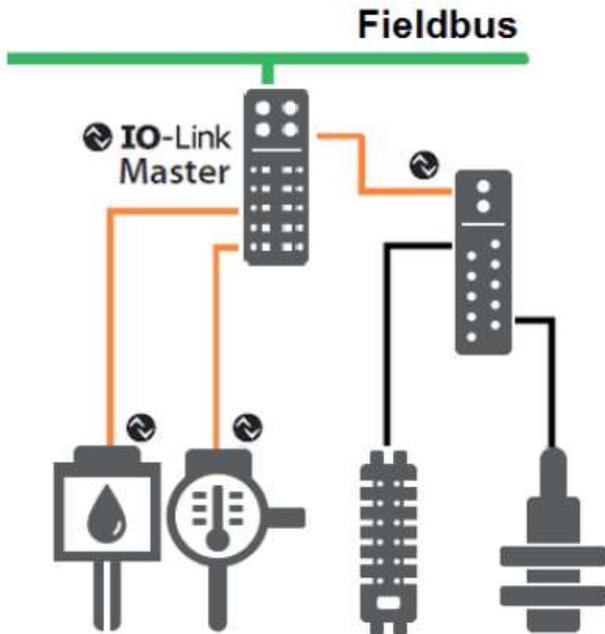


Présentation des capteurs IO LINK

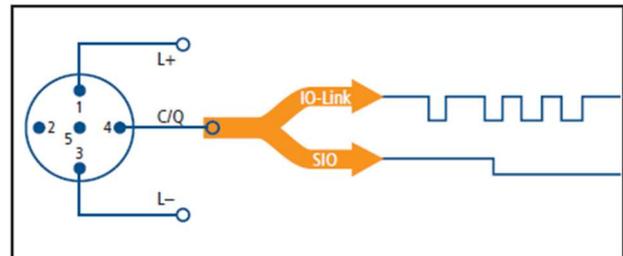
Les capteurs IO LINK sont destinés à être connectés à un maître IO LINK.



En standard, la connectique a 3 bornes. La borne 4 (C/Q) est une sortie de transmission série de données issues du capteur. Elle peut également fonctionner en entrée/sortie tout ou rien (Standard Input Output).

- Pin 1: 24 V
- Pin 3: 0 V
- Pin 4: Switching and communication line (C/Q)

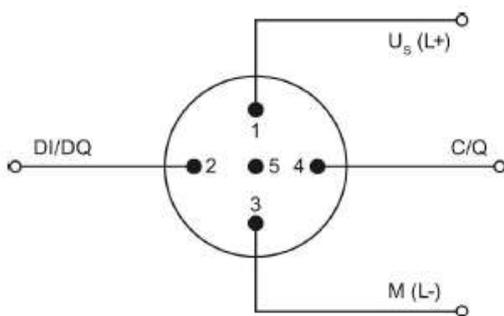
Besides the IO-Link communication, these three pins are also used to supply the device with at least 200 mA



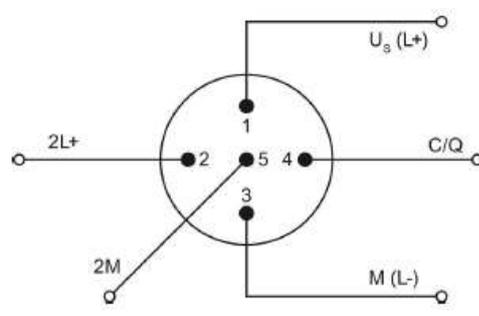
Les vitesses de transmission sur la liaison IO LINK

- COM 1 = 4,8 kbauds
- COM 2 = 38,4 kbauds
- COM 3 = 230,4 kbauds (en option)

Port Classe A : Possibilité d'avoir une entrée/sortie TOR supplémentaire sur la broche 2



Port Classe B : Possibilité d'avoir une alimentation supplémentaire.



Analyse des paramètres d'un capteur de température TV7105

Les capteurs IO LINK possèdent un ensemble de paramètres qui décrivent son fonctionnement. Un fichier IODD (IO Device Description) de paramétrage est associé à chaque capteur.

Exemples de paramétrage pour le capteur TV7105

Pour répondre aux questions suivantes, il faut rechercher les informations dans le fichier de description du capteur TV7105 joint en annexe.

Le capteur dispose de 2 bornes d'alimentation et de 2 sorties (OUT1 et OUT2).

⇒ Préciser les bornes pour l'alimentation et les sorties OUT1 et OUT2. Préciser le numéro de la ligne IO LINK.

L+ ⇒	OUT1 ⇒	IO LINK ⇒
L- ⇒	OUT2 ⇒	

Données process: La valeur de la température fournie par le capteur est codée sur 16 bits.

⇒ Indiquer la valeur retournée pour une température de -50°C et la valeur retournée pour 150°C. En déduire la résolution en °C du capteur.

N pour -50°C ⇒ Npour 150 °C ⇒	Résolution en °C
----------------------------------	------------------

⇒ Indiquer à quoi correspond les variables **uni** , **Hi** et **Lo**

uni ⇒	Hi ⇒ Lo ⇒
-------	--------------

Le capteur de température est branché sur un maître IO LINK, câblé sur le réseau du lycée.

⇒ Faire une recherche d'adresse du maître à l'aide du logiciel Ethernet Device Configuration

Adresse IP trouvée :

⇒ A l'aide du logiciel LR Device, ajouter le maître IO LINK et en cliquant sur P1 : TV7105. Retrouver les paramètres du capteur et noter les valeurs enregistrées pour uni, Hi et Lo

uni ⇒	Hi ⇒ Lo ⇒
-------	--------------

Traitement des données avec NODE RED

Nous allons paramétrer le maître IO LINK de la même manière que dans le TP précédent (Introduction aux maîtres IO LINK).

Remarque : Le serveur du maître IO LINK accepte à un nombre limité de connexion. L'accès à la page suivante n'est pas forcément accessible pour tout le monde en même temps.

⇒ Sous l'explorateur Internet, essayer d'accéder au menu de paramétrage du maître IO LINK : IP_Master/web/subscribe.

⇒ Une publication MQTT sera ajoutée par le prof avec les paramètres suivants :

- Adresse du BROKER : 172.16.6.61 (Raspberry sur réseau lycée).
- Événement provoqué par le Timer 1 toutes les 5 s.
- Topic : colin/tempTV7105
- MQTT port 1883
- La donnée à transmettre : port[1]/iolinkdevice/pdin



Exemple :

Consumer ID	Channel	Broker/Server	Event	Data	Duration
363	mqtt	172.16.6.167:1883/colin/tempTV7105	00-02-01-6E-A3-67/timer[1]/counter/datachanged	00-02-01-6e-a3-67/iolinkmaster/port[1]/iolinkdevice/pdin	lifetime

⇒ Sous NODE RED, réaliser le flux suivant et paramétrer le NODE mqtt in avec le même BROKER et le même TOPIC que pour le maître IO LINK

```
colin/tempTV7105 : msg.payload : Object
▼ object
  code: "event"
  cid: 363
  adr: "/colin/tempTV7105"
  data: object
    eventno: "432"
    srcurl: "00-02-01-6E-A3-67/timer[1]/counter/datachanged"
    payload: object
      /timer[1]/counter: object
        /iolinkmaster/port[1]/iolinkdevice/pdin: object
          code: 200
          data: "00F2FF00"
```

⇒ Justifier que la valeur de data affichée ci-dessus (00F2FF00), et à l'aide de la documentation du capteur, correspond à une température de 24,2°C.

Traitement de la donnée :

Problème : la donnée renvoyée est ici sous forme d'une chaîne de caractères (« 00F2FF00 ») représentant la valeur en hexadécimale.

La valeur de la température est alors ici $0x00F2 = 242$ dixièmes de degré.

En javascript il existe une fonction `parseInt` qui transforme une chaîne de caractères en entier

Syntax

```
parseInt(string, radix)
```

Definition and Usage

The `parseInt` method parses a value as a string and returns the first integer.

A radix parameter specifies the number system to use:

2 = binary, 8 = octal, 10 = decimal, 16 = hexadecimal.

If radix is omitted, JavaScript assumes radix 10. If the value begins with "0x", JavaScript assumes radix 16.

⇒ Ajouter la fonction suivante et observer le résultat obtenu (prendre la remarque ci-après).

```
1 msg.payload=parseInt(msg.payload.data.payload["/iolinkmaster/port[1]/iolinkdevice/pdin"].data,16)/655360;
2 return msg;
```

Remarque : Pour ne pas retaper la ligne complète, utiliser l'icône « Copier le chemin » comme dans l'exemple ci-dessous :



Lorsqu'on divise un nombre binaire par 2 on décale le mot binaire d'un rang à droite

/2 ⇒ décalage d'un bit à droite (équivalent à $\gg 1$)

/4 ⇒ décalage de 2 bits à droite ($\gg 2$)

/256 ⇒ $\gg 8$

⇒ Justifier alors la division par 655360 après avoir transformé la chaîne de caractères représentant un nombre hexadécimal en un entier.

⇒ Comme pour le TP précédent (introduction aux maîtres IO Link) , compléter le flow Node Red pour obtenir une jauge affichant la température, puis transformer l'information en un fichier JSON.

⇒ Compléter le flow Node Red pour enregistrer les données de la température dans une base de données dont les paramètres sont les suivants :

- Base de données hébergée sur une Raspberry dont l'adresse IP est : 172.16.6.68
- Nom de la base de données TPTSMI
- Nom de la table (Measurement) : nom_etudiant
- Pas de mot de passe

⇒ A l'aide du logiciel Putty, se connecter en SSH sur la Raspberry (login : pi mot de passe : raspi4\$)

⇒ A l'aide des commandes suivantes, vérifier que les données sont enregistrées dans la base de données :

- Exécuter InfluxDB ⇒ `influx -precision rfc3339`
- Visualiser les bases de données ⇒ `show databases`
- Utiliser la base de données TPTSMI ⇒ `use TPTSMI`
- Visualiser les données enregistrées ⇒ `select * from nom_etudiant`

⇒ Faire une démo au prof