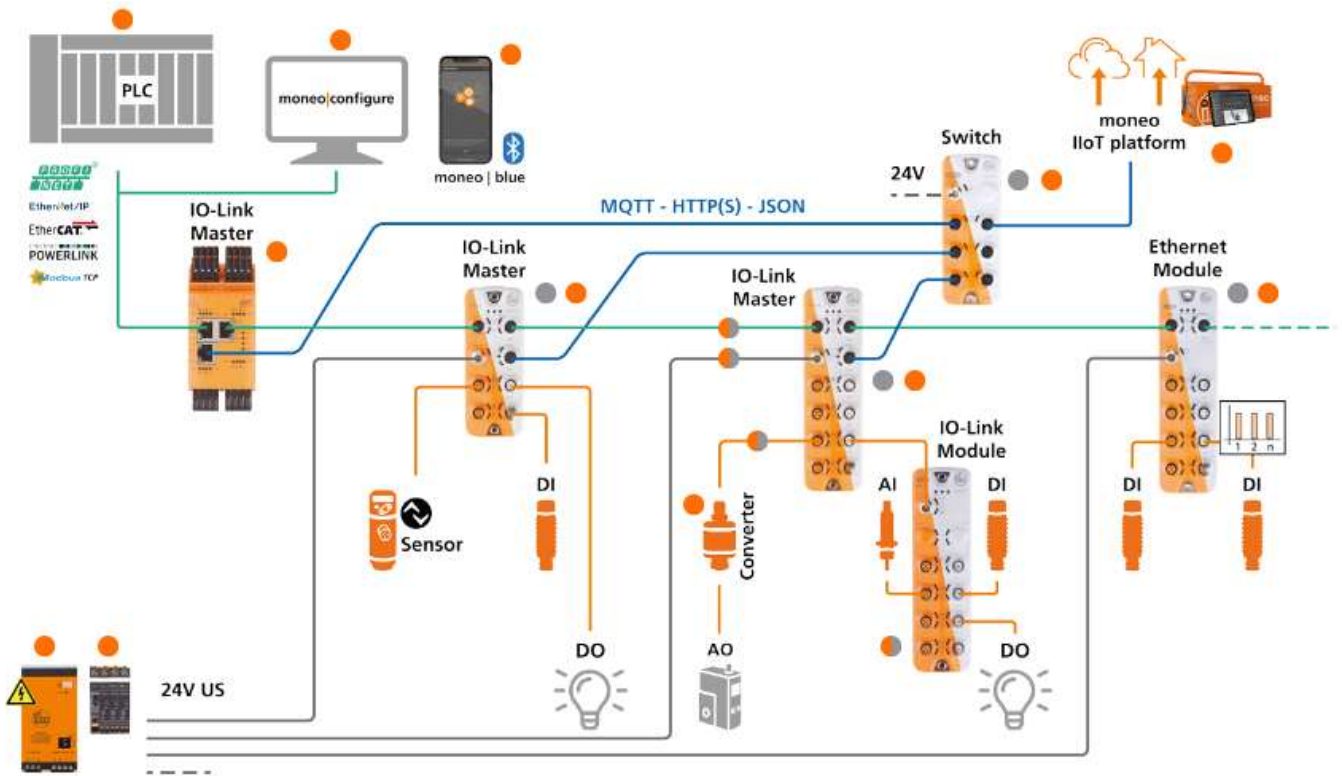


# Introduction aux maîtres IO LINK

COLIN.G

Les maîtres IO LINK disposent de connecteurs IO LINK sur lesquels on peut câbler des capteurs IO LINK, des connecteurs Ethernet pour le dialogue avec l'automate programmable suivant différents protocoles (Profinet – EtherNet/IP – Modbus TCP – EtherCAT – POWERLINK ...), des connecteurs Ethernet pour transmettre des grandeurs mesurées sur un serveur IoT (protocole MQTT – HTTP – JSON).



## Maître IO LINK - Introduction

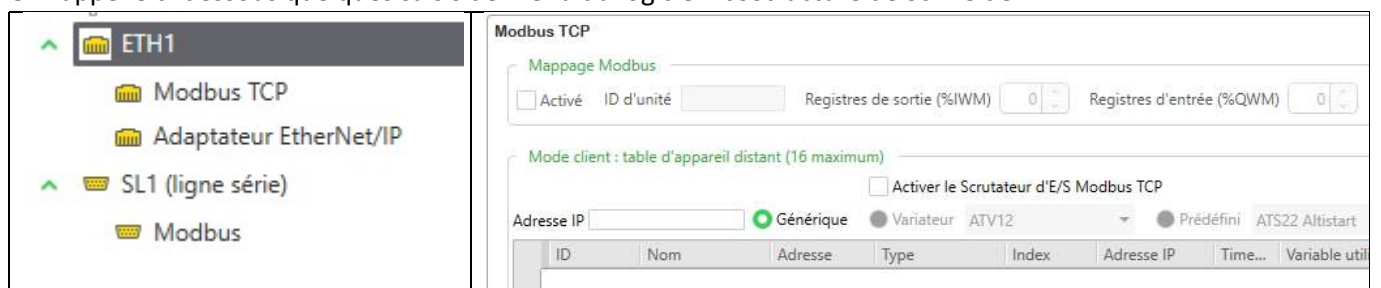
⇒ En consultant le site IFM (Produits\IO-Link\Maître IO-Link\Maîtres IO-Link pour applications de terrain dans l'automatisation industrielle), indiquer pour les 3 maîtres IO LINK proposés ci-dessous les protocoles supportés.

### Protocoles supportés (faire une croix) :

	IO-Link	PROFINET	EtherNet/IP	EtherCAT	MQTT JSON	Modbus TCP	POWERLINK	AS-i
AL1320								
AL1350								
AL1340								

IFM distingue les ports Ethernet IoT (pour objets connectés avec protocole MQTT JSON) et les ports Ethernet pour le dialogue avec automate (protocole MODBUS TCP, PROFINET etc...).

On rappelle ci-dessous quelques saisis de menu du logiciel EcoStructure de Schneider



<p><b>Paramètres de sécurité</b></p> <p><input type="checkbox"/> Protocole de programmation activé</p> <p><input type="checkbox"/> Protocole EtherNet/IP activé</p> <p><input type="checkbox"/> Serveur Modbus activé</p> <p><input type="checkbox"/> Protocole de recherche automatique activé</p>	<p><b>Adaptateur EtherNet/IP</b></p> <p><b>Paramètres</b></p> <p><input type="checkbox"/> Activé</p> <p>Assemblage d'entrée (cible --&gt; origine, %QWE)</p> <p>Instance <input type="text" value="0"/></p> <p>Taille (Mots) <input type="text" value="0"/></p> <p>Assemblage de sortie (origine --&gt; cible, %IWE)</p> <p>Instance <input type="text" value="0"/></p> <p>Taille (Mots) <input type="text" value="0"/></p>
---	---

⇒ Indiquer quels seraient (à priori) les maîtres IO LINK compatibles pour un dialogue avec cet automate Schneider

**Maîtres IO LINK compatibles pour un dialogue avec l'automate Schneider M221**

AL1320     AL1340     AL1350     AL1342     AL1302     AL1102

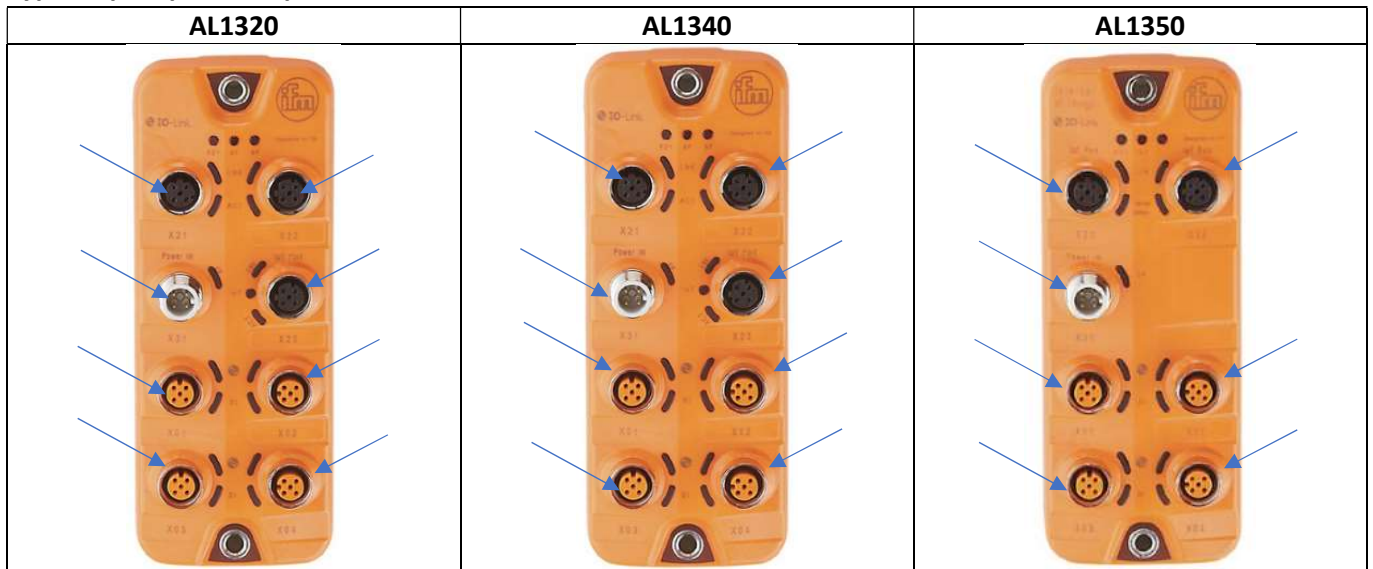
⇒ Indiquer quels seraient (à priori) les maîtres IO LINK compatibles pour un dialogue avec un automate Siemens

**Maîtres IO LINK compatibles pour un dialogue avec un automate Siemens**

AL1320     AL1340     AL1350     AL1342     AL1302     AL1102

⇒ Pour les maîtres IO LINK ci-dessous, préciser le type des différents ports parmi ceux-ci :  
IoT – EtheNet/IP – Profinet – IO LINK – MODBUS TCP – Power IN

**Type de port pour chaque maître**



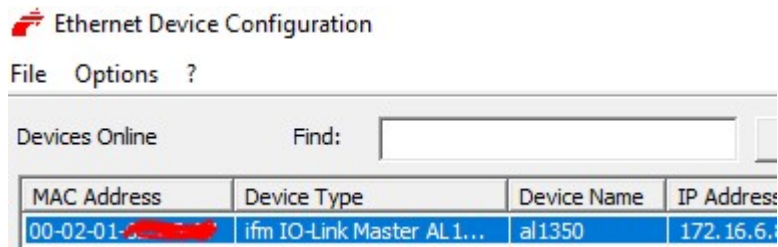
**Maître IO LINK – Paramétrage IoT**

Le maître IO LINK AL1350 est câblé sur le réseau du lycée. Un capteur de température IO LINK est branché sur ce maître.

⇒ En observant le montage, indiquer les ports utilisés dans ce cas.

<p><b>Ports utilisés:</b></p>
-------------------------------

⇒ A l'aide du logiciel **Ethernet Device Configuration** ou le logiciel **DipiScan**, faire une recherche pour connaître l'adresse IP du maître IO LINK.



⇒ Noter l'adresse MAC et l'adresse IP trouvée. Indiquer si l'adresse IP est compatible avec celle du réseau du lycée.

<b>MAC</b>	<b>Compatibilité</b>
<b>IP</b>	



⇒ A l'aide du logiciel LR DEVICE (menu IFM electronic), ajouter le maître IO LINK

⇒ Indiquer si l'attribution de l'adresse IP est réalisée de façon automatique (serveur DHCP) ou de façon fixe. Noter l'IP de la passerelle et le masque de réseau.

<input type="checkbox"/> DHCP ou <input type="checkbox"/> IP fixe	IP passerelle	Masque
---	---------------	--------

Dans le menu Cockpit de l'appareil AL1350, il est possible de récupérer des informations sur certains paramètres du maître IO LINK.

⇒ Noter la tension d'alimentation, la température et le courant consommé pour l'appareil AL1350

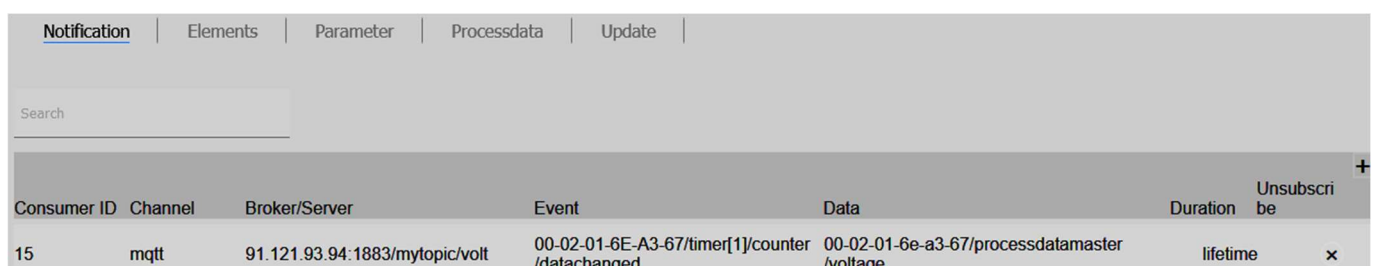
Tension	Température	Courant
---------	-------------	---------

Dans le menu Cockpit, P1 : TV7106 , il est possible de visualiser la température du capteur de température branché sur le port 1 IO LINK.

⇒ Noter la température mesurée

Température délivrée par le capteur
-------------------------------------

⇒ Dans la barre de l'explorateur internet (Firefox par exemple), taper l'adresse du maître IO LINK, suivi de /web/subscribe : exemple 172.16.6.88/web/subscribe

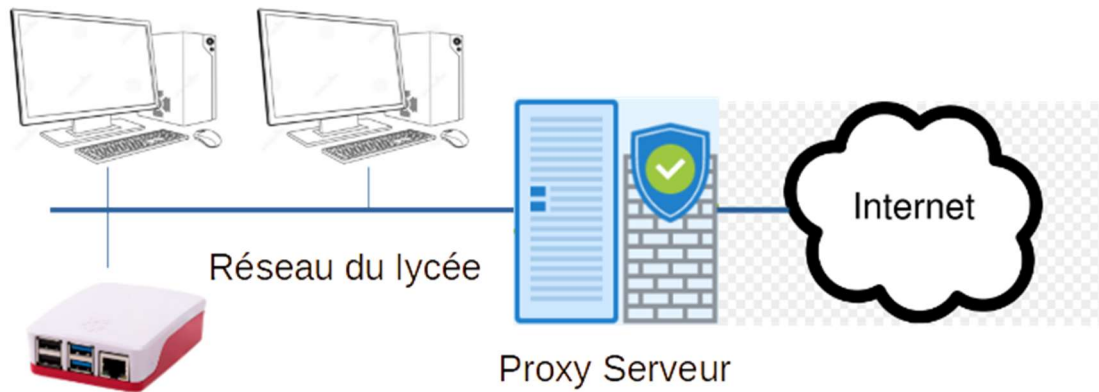


### Paramétrage du service MQTT

Pour utiliser le service MQTT, il faut un serveur MQTT, appelé BROKER.

Le logiciel le plus répandu pour rendre ce service est MOSQUITTO. Il peut être installé sur n'importe quel poste du lycée tournant sous Windows ou sous Linux. On peut également utiliser un serveur sur Internet, comme test.mosquitto.org.

Pour des raisons de sécurité informatique et éviter les problèmes avec des pare feux (firewall) ou proxy du lycée, on utilisera un serveur interne au lycée , ici une raspberry. Dans un cadre professionnel, il faudrait utiliser un serveur digne de ce nom, avec disque dur monté en RAID, alimentation de sécurité, onduleur, etc..



Dans l'exercice qui suit, on souhaite transmettre la valeur de la tension d'alimentation du maître IO LINK, toutes les 5 secondes.

⇒ Cliquer sur + pour ajouter une nouvelle publication MQTT	
⇒ Pour l'événement qui va provoquer une publication, choisir timer[1]/counter/datachanged On réglera le timer 1 (le temps entre 2 transmissions)	
⇒ Sélectionner les données à transmettre, ici uniquement la valeur de la tension d'alimentation	
⇒ Entrer un numéro ID (prendre jour et mois d'anniversaire par exemple)	
⇒ Entrer l'adresse IP du BROKER : <b>172.16.6.61</b>	
⇒ Entrer le numéro du port MQTT 1883	
⇒ Entrer le sujet (ou Topic) : entrer votre nom suivi de volt : /dupont/volt	
⇒ Dans le menu Parameter/Timer[1], vérifier que l'intervalle de temps est réglé sur 5000 ms	

### Vérification des publications avec le logiciel MQTT.fx

- ⇒ Lancer le logiciel MQTT.fx
- ⇒ Dans le paramétrage du BROKER, donner un nom, entrer l'adresse du Broker et le numéro du port. Entrer un Client ID quelconque. Ne pas déclarer de proxy puisqu'on reste sur le réseau interne.
- ⇒ Se connecter au Broker
- ⇒ Dans le menu Subscribe, souscrire au Topic enregistré précédemment.
- ⇒ Visualiser le contenu du fichier JSON envoyé par le maître IO LINK.

```
colin/volt
07-12-2022 19:31:04.70264838
{"code": "event", "cid": 1403, "adr": "/colin/volt", "data": {"eventno": "6965", "srcurl": "00-02-01-6E-A3-67/timer[1]/counter/datachanged", "payload": {"timer[1]/counter": {"code": 200, "data": 6965}, "/processdatamaster/voltage": {"code": 200, "data": 24034}}}}
```

Dans ce fichier JSON, la valeur de la tension d'alimentation est de 24034 mV, soit pratiquement 24V.

## Récupération des données sous NODE RED

En toute logique on doit utiliser NODE RED installé sur un serveur digne de ce nom. Il est possible d'installer NODE RED sur un PC sous Windows (son propre PC par exemple), mais si le PC est éteint, le service est perdu.

On utilisera Node Red installé sur le PC.

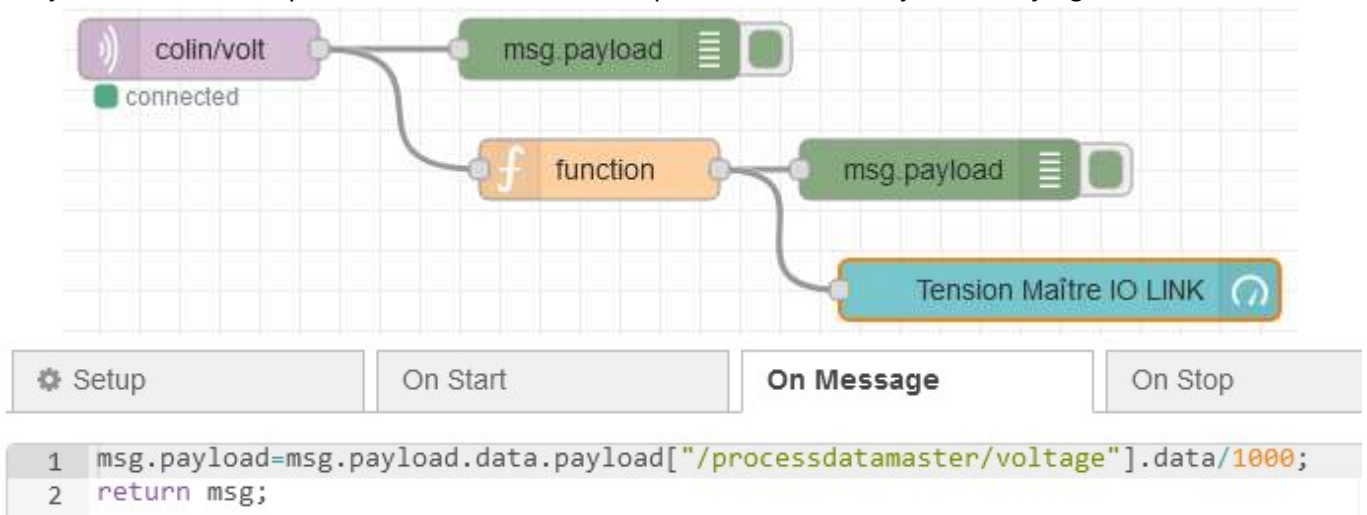
⇒ Sous Node Red réaliser le flow ci-dessous en paramétrant correctement le Broker (172.16.6.61) et le Topic.



⇒ Retrouver dans la fenêtre DEBUG, le fichier JSON transmis toutes les 5 s.

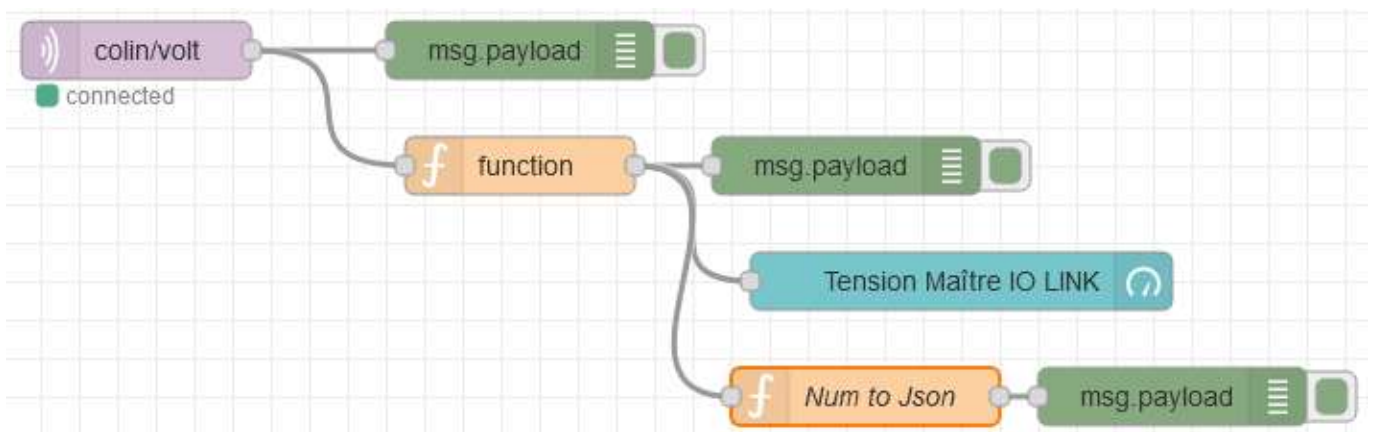
```
08/12/2022 15:14:00 node: bb2c2aab.1343d
colin/volt : msg.payload : Object
  ▼ object
    code: "event"
    cid: 1403
    adr: "/colin/volt"
  ▼ data: object
    eventno: "21160"
    srcurl: "00-02-01-6E-A3-67/timer[1]/counter/datachanged"
  ▼ payload: object
    ▼ /timer[1]/counter: object
      code: 200
      data: 21160
    ▼ /processdatamaster/voltage: object
      code: 200
      data: 24026
```

⇒ Ajouter une fonction pour extraire la valeur numérique de la tension et ajouter une jauge.



<pre>08/12/2022 15:40:40 node: e0d00d69.3bec2 colin/volt : msg.payload : number 24.034</pre>	<h3 style="text-align: center;">Tension Maître IO LINK</h3>  <p style="text-align: center;"><b>24.036</b> Volts</p>
--	---

⇒ Ajouter une fonction pour obtenir un fichier JSON pour la tension (pour enregistrement dans une base de données par exemple)



⚙ Setup	On Start	<b>On Message</b>	<pre>08/12/2022 15:55:20 node: e553918c.82353 colin/volt : msg.payload : Object   ▶ { Voltage: 24.039 }</pre>
<pre>1 msg.payload={"Voltage":msg.payload}; 2 return msg;</pre>			