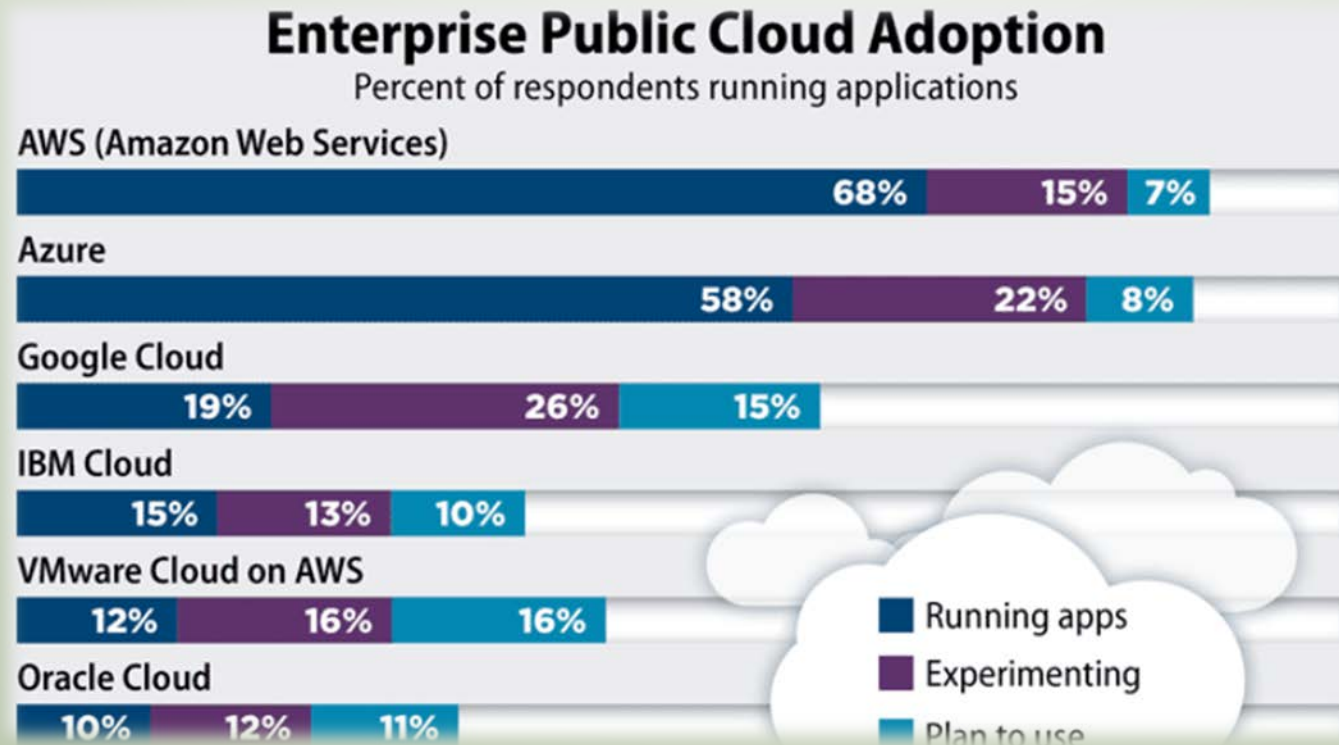


Mise en œuvre de l'IOT

Solutions matériels et logicielles



Interopérabilité



Quelles sont vos attentes ?

Pourquoi voulez vous mettre en place une solution IOT ?

Quelles sont vos principales difficultés ?

Qu'est ce que pour vous, un bon système IOT ?

Contexte / environnement travail (1/2)



- Laboratoire chimie #200 personnes / catalyse
- **Unité fonctionnelle : Equipe Ingénierie**
- Responsable instrumentation de l'équipe
 - # 30 personnes / 20 étudiants ou CDD
- Changement étudiants / Projets tous les 3 ans

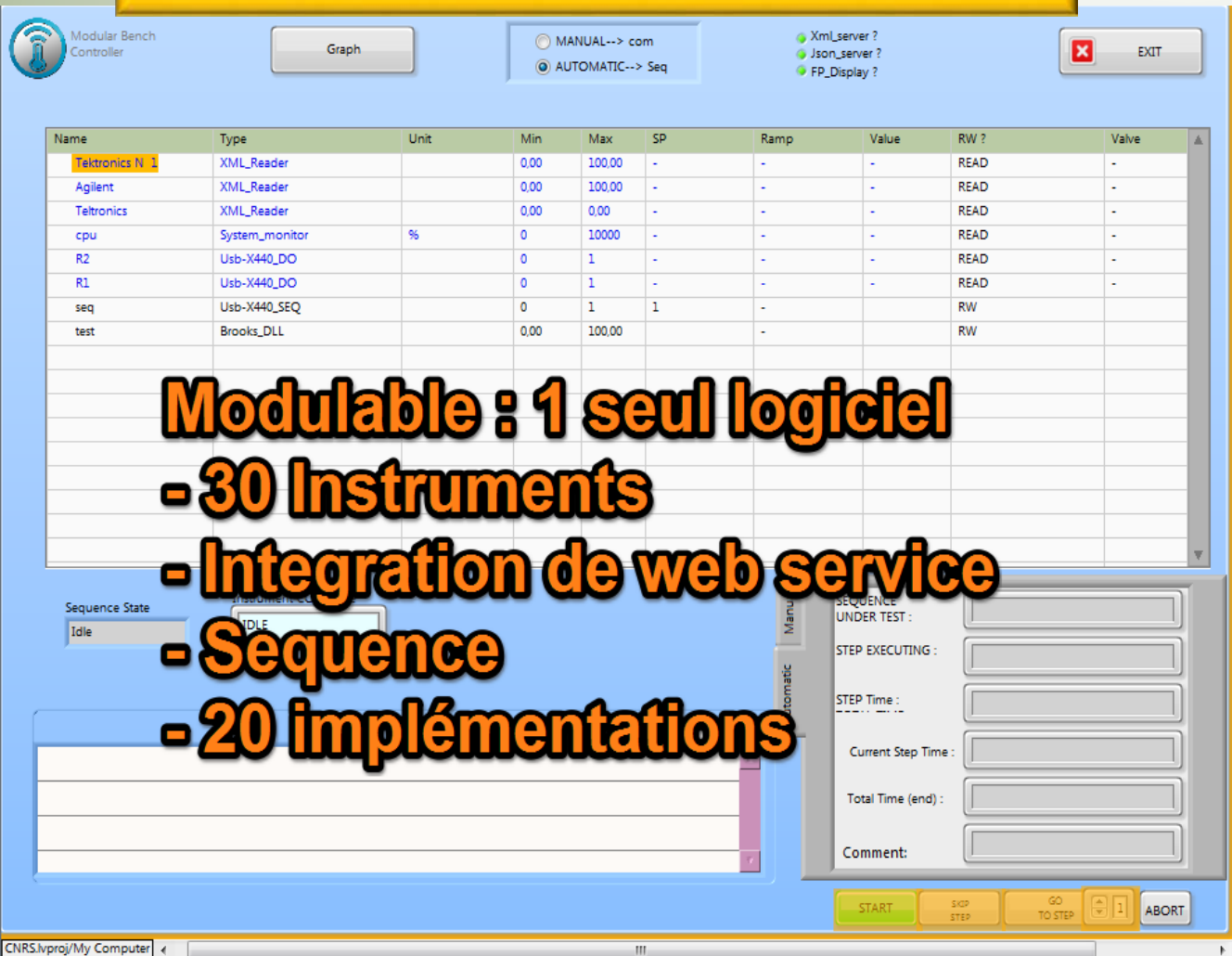
Conception modulaire pour le **matériel** et les **logiciels**

→ **Simplification + optimisation**

IOT : qualité , sécurité , Cout.

Contexte / environnement travail (2/2)

1 logiciel



Modular Bench Controller

Graph

☐ MANUAL--> com
☒ AUTOMATIC--> Seq

Xml_server ?
Json_server ?
FP_Display ?

EXIT

Name	Type	Unit	Min	Max	SP	Ramp	Value	RW ?	Valve
Tektronics N 1	XML_Reader		0.00	100.00	-	-	-	READ	-
Agilent	XML_Reader		0.00	100.00	-	-	-	READ	-
Teltronics	XML_Reader		0.00	0.00	-	-	-	READ	-
cpu	System_monitor	%	0	10000	-	-	-	READ	-
R2	Usb-X440_DO		0	1	-	-	-	READ	-
R1	Usb-X440_DO		0	1	-	-	-	READ	-
seq	Usb-X440_SEQ		0	1	1	-	-	RW	-
test	Brooks_DLL		0.00	100.00	-	-	-	RW	-

Sequence State: Idle

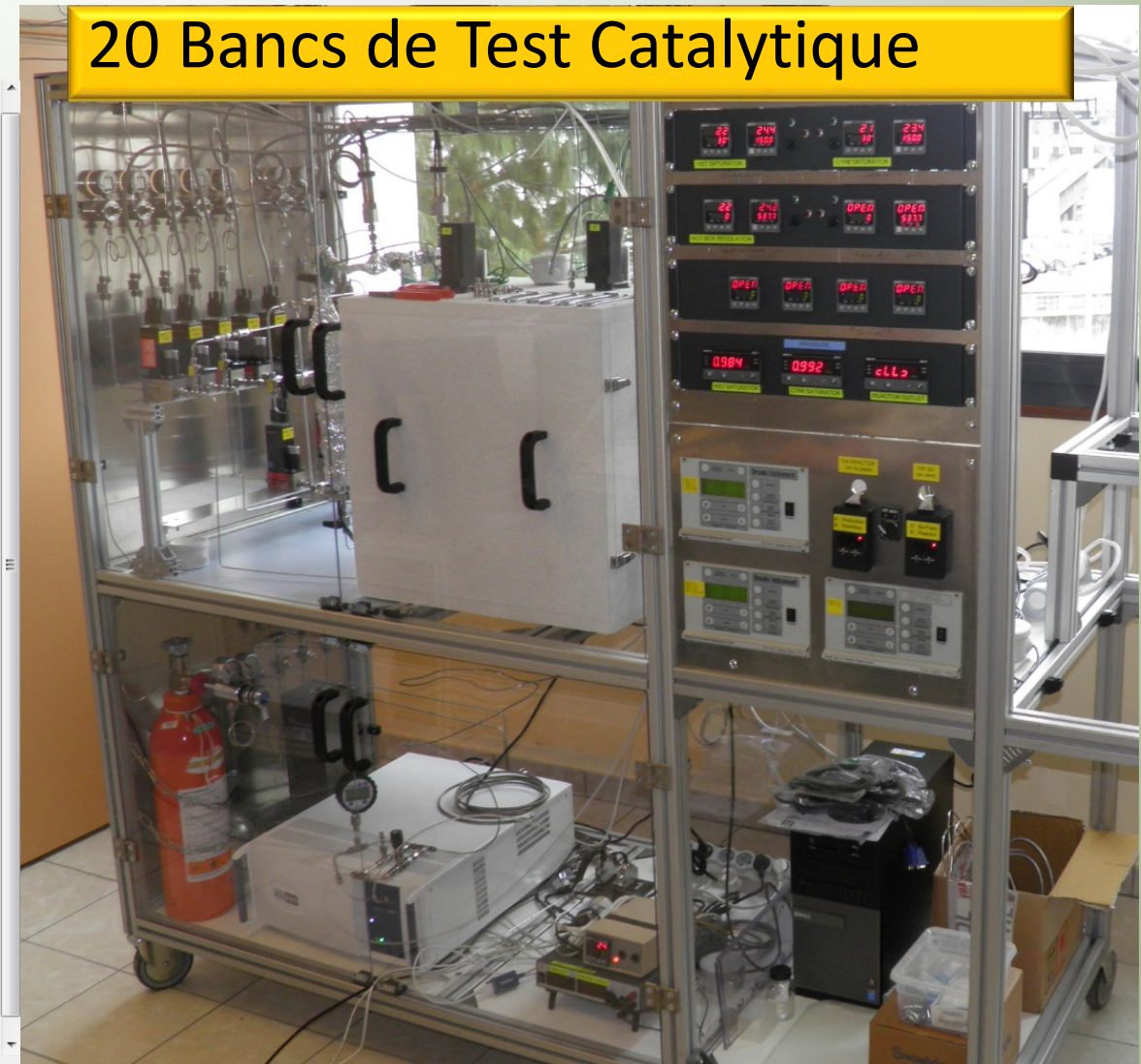
SEQUENCE UNDER TEST :
STEP EXECUTING :
STEP Time :
Current Step Time :
Total Time (end) :
Comment :

START SKIP STEP GO TO STEP 1 ABORT

Modulable : 1 seul logiciel

- 30 Instruments
- Integration de web service
- Sequence
- 20 implémentations

20 Bancs de Test Catalytique



Intro : Idée reçue N°2

→ Ce n'est pas assez sécurisé



FAUX

→ On peut lire uniquement les données → aucunes actions



→ On **choisit** les données que l'on veut envoyer

→ On accède pas aux données directement : méthode **encapsulation**

→ On peut **limiter** l'accès à certains utilisateurs (root/admin/user)

Intro : Idée reçue N°3

→ Cela ne sert à pas grand-chose d'envoyer sur internet...

FAUX

→ Alerte des utilisateurs (mail/sms...) 

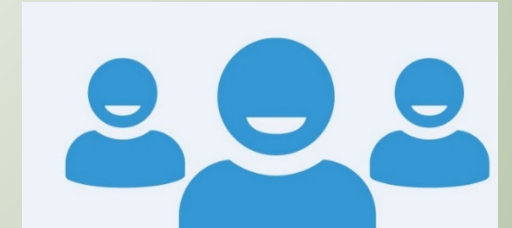
→ Watchdog avancé.



→ Historique évènements / log



→ Visualisation des données.



Objectif → Satisfaction utilisateurs

Interface logiciel : Intégration (1/2)

Objectif : Processus découplé

Pourquoi



RT \neq IOT \rightarrow Choix et traitement des données à envoyées

Comment

- Logiciel \rightarrow serveur de données (xml/json)

Gain

- Faciliter le debug.
- Sécurité
- Robustesse

NE JAMAIS MANQUER LES MISES À JOUR DE SÉCURITÉ !



Interface Matériel : choix (2/2)

Device with I/O (Server)

Client Machine



HTTP Request

{ REST }

XML Response

```
- <Response>
- <Terminal>
  <Name>Sum</Name>
  <Value>9.000000</Value>
</Terminal>
</Response>
```



Interface Matériel : intégration (1/2)

- Quels sont les valeurs les plus **importantes** ?
 - Quels sont les **objectifs** :
sécurité / dérive / validation / sauvegarde de données / proof of concept
- Découplage du système existant :**
Matériel → utiliser des I/O qui ne perturberons pas le système en place



Ne Jamais modifier un système sûr en fonctionnement

Interface Matériel : choix (2/2)



- **Intelligence** aux niveau du **matériel** → simplification et robustesse
 - Régulateur de température (autonome, rampe, paliers)
 - Détecteur de gaz (avec afficheur et relais)



- Haute interopérabilité → facilité et souplesse d'intégration
 - **API RESTFUL** → commandes http
 - **Serveur web** → XML ou Json
 - **I/O** : Pilotage sans com possible / découplage du système
 - Port communication Industriel : modbus, OPC



Systemes simple, configurable et auto-documenté



Eviter les solutions propriétaires (pérennité et interopérabilité réduite)

IOT MOXA : Modbus Gateway

MB : Armv7 / 512 Mo / 1Ghz / 8Gb eMMC

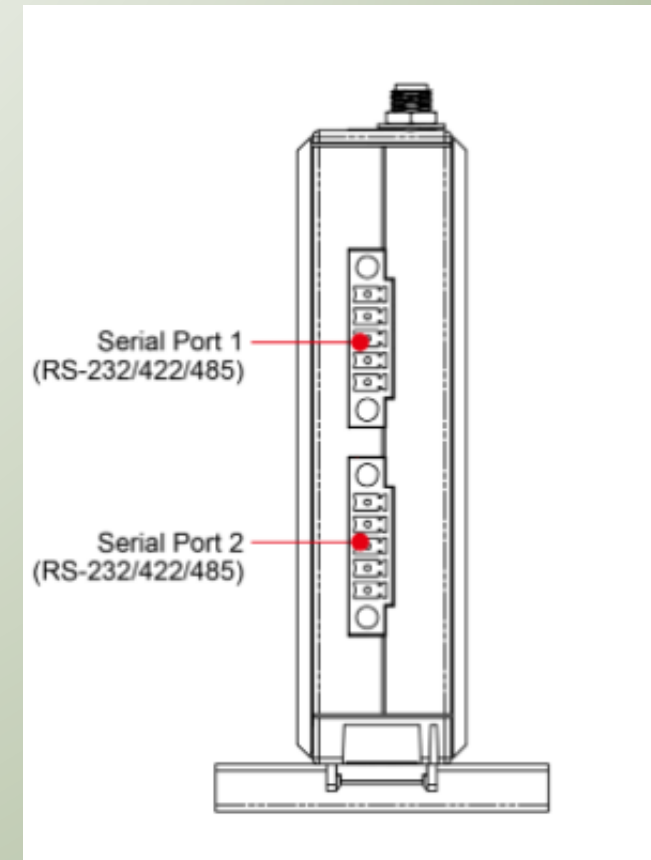
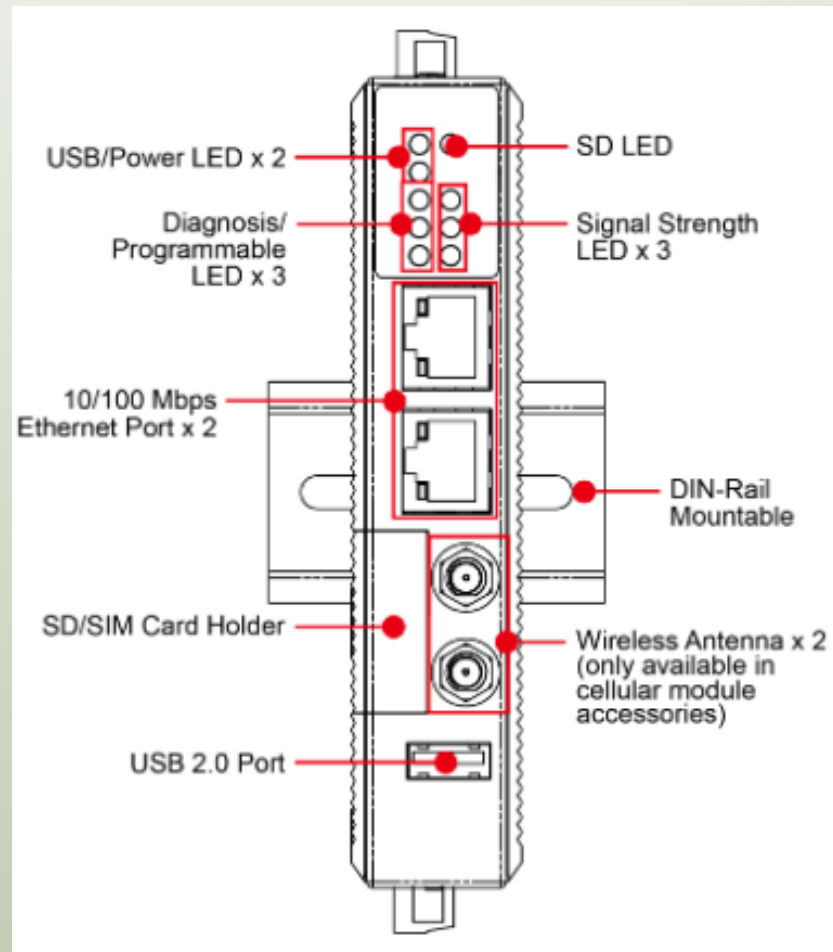
Pre-installed OS : Linux Debian 9 (Linux kernel v4.4)

Storage Slot : SD / USB

Serial Ports : 2 x RS-232/422/485

Wireless (option) : Wifi or 4g

Ethernet : Wan/Lan



Things Pro : surcouche logiciel

Simplification :  → IOT

Protocoles IOT : Mqtt / Azure / Aws / API { REST } ❤️

Protocoles routeur : DHCP / Firewall / route

OS linux : node-red / influxdb....



Developer Administrator (Root)

- Token management for device identification
- All Administrator tasks



Administrator

- Manage devices
- Schedule batch jobs
- Manage user accounts



User

- Read only

Remote I/O
and Sensors



Local SCADA



Modbus Slave

Public / Private Cloud



MQTT / RESTful APIs
Cloud Clients

IIoT Gateway



Modbus RTU / TCP

ThingsPro Gateway

A ready-to-run data acquisition
and device management software



Calamp : Modem 4G

Objectif : accès au réseau 4G internet + I/O (4ai, 8di, 8do)

OS : OpenWrt

Protocoles : SNMP / DHCP / firewall / VPN

Wireless : Wifi / GPS

Com port : X2



Command SMS-CLI (sans 4g)
/output pw=[password] rn=v...
/iostatus pw=[password]
/status pw=[password]



Eedomus : Box domotique

Double API local/distante { REST } ❤️

Sauvegarde illimité

Pas de maintenance et Maj régulière ❤️

Market, communauté



Jeedom : Box domotique

Base de donnée + dashboard ❤️

Opensource ❤️

Local (+sauvegarde config Cloud)

API REST, Market



Moxa ioLogik E1200 Series : I/O



Modbus TCP → Scada / contrôle commande

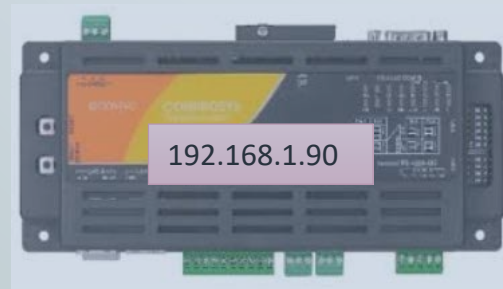
API { REST } → IOT ❤️

MX-AOPC UA Server (push)

Passerelle avec I/O



- Built-in **OPC UA Server**
- Promoting a multi-vendor system supporting a varied lineup of PLCs and **Modbus** devices
- Connect up to 10 links and data collection from 100 group registers
- Measurement and Upload
- **Web Monitoring build-in** ❤️
- **Web Task Script**



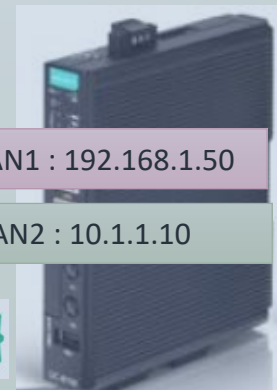
Routeur 4G

API
{ REST }

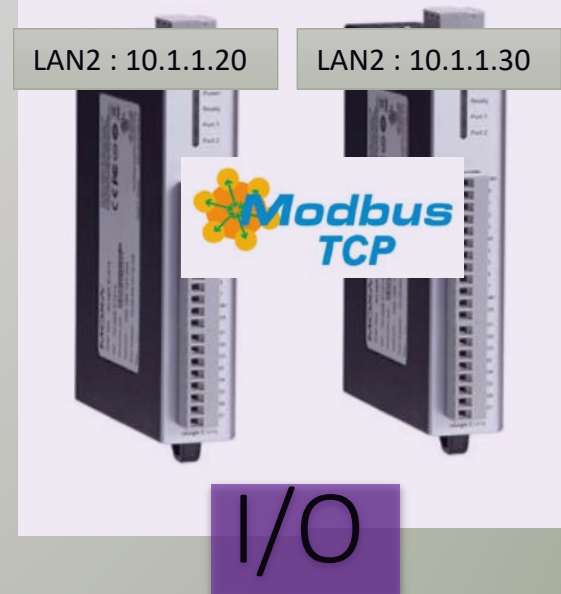


Passerelles

API
{ REST }



API
{ REST }



Analog output Analog input



0-10 V // 12 bit
Modbus TCP
API REST : JSON



0-10 V // 14 bit
Modbus TCP
API REST : JSON



0-3,3 V // 16 bit
API REST : XML



0-10 V // 10 bit
ZWAVE
Polling



0-80 mv // 12 bit
Modbus RTU
API REST

API RESTFUL

Low level : I/O moxa / IPX

Middleware : Moxa Things pro

Admin local : Eedomus php script

End-user Cloud: API eedomus

ThingsPro API Reference

modbus : Handles Modbus framework configuration

DELETE	/mxc/modbus/equipment-templates
GET	/mxc/modbus/equipment-templates
POST	/mxc/modbus/equipment-templates
DELETE	/mxc/modbus/equipment-templates/{id}
GET	/mxc/modbus/equipment-templates/{id}
PUT	/mxc/modbus/equipment-templates/{id}
GET	/mxc/modbus/interfaces
POST	/mxc/modbus/interfaces
DELETE	/mxc/modbus/interfaces/{id}
GET	/mxc/modbus/interfaces/{id}
PUT	/mxc/modbus/interfaces/{id}
GET	/mxc/modbus/equipments
GET	/mxc/modbus/access/{equipment}/{tag}
PUT	/mxc/modbus/access/{equipment}/{tag}

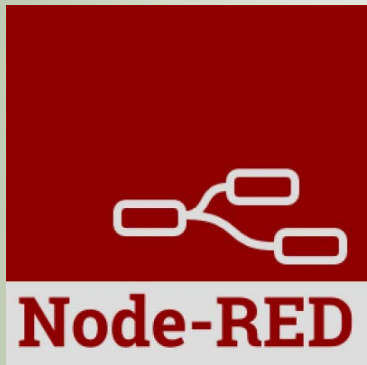
AVOUS DE JOUER

Mise en œuvre

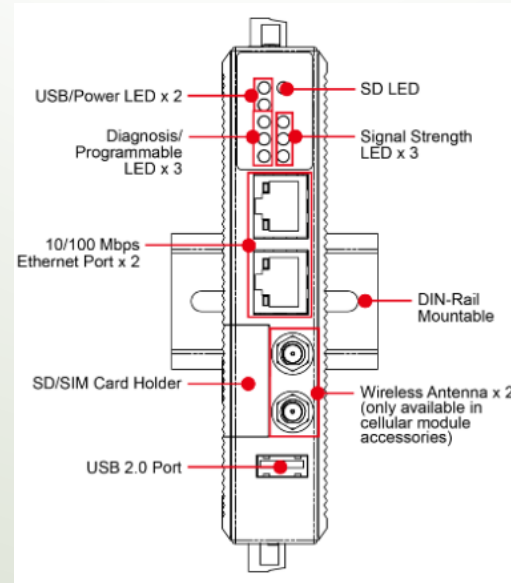
Changement Sortie analogique

- Appui bouton → 3V
- Alerte si > 2 Volt
 - Appel vocal
 - Sms
 - Push

Evènement Capteur de niveau



<http://192.168.1.60:1880/>



eedomus_local_API

