



# Le Banc connecté en toute sécurité

API et web services

Réunion AlpesView 2015  
Florian CHAPON & Emmanuel LANDRIVON

# SOMMAIRE

## **1. Fichier d'initialisation**

→ Librairie OpenG

## **2. Utilisation d'une API Matériel existante**

→ Cas de l'IPX800

## **3. Développement d'une API Logiciel**

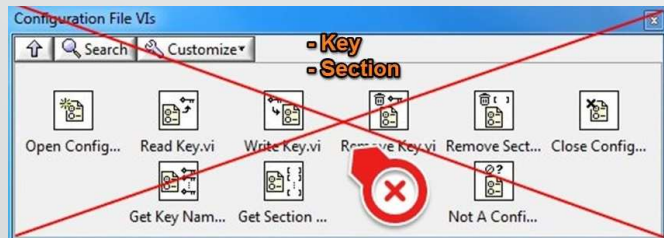
→ Pourquoi ?

→ Utilisation des web-services

## **4. Conclusion**

# 1. Fichier d'initialisation

**Problématique :** un fichier de configuration peut être compliqué à créer ou modifier s'il est important



**Solution :** Openg Variant



- Création de Key/section compliqué
- Lisibilité du code

- + Permet une sauvegarde facile de la configuration du programme à partir d'un cluster
- + Formatage automatisé (cluster)
- + Modularité du code

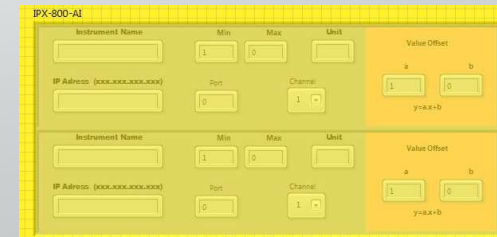
Save Cluster



Fichier de configuration texte → config.ini

```
[IPX-800-AI_0]
Instrument Name = "Test"
Min = "1.000000"
Max = "0.000000"
IP Address (xxx.xxx.xxx.xxx) = "192.68.0.2"
Port = "80.000000"
Channel = "0"
a = "1.000000"
b = "0.000000"
Unit = ""
```

Load Cluster



## 2. API ?

### Definitions



→ *Application Programming Interface*

→ *Interface de communication entre 2 systèmes (matériel et/ou logiciel)*



**m2m** : Interaction entre 2 machines (matériel + interface + logiciel) sans intervention humaine.



**Data push** : Sur un événement (cycle, déclencheur) envoi des données a un serveur de façon **unidirectionnel**

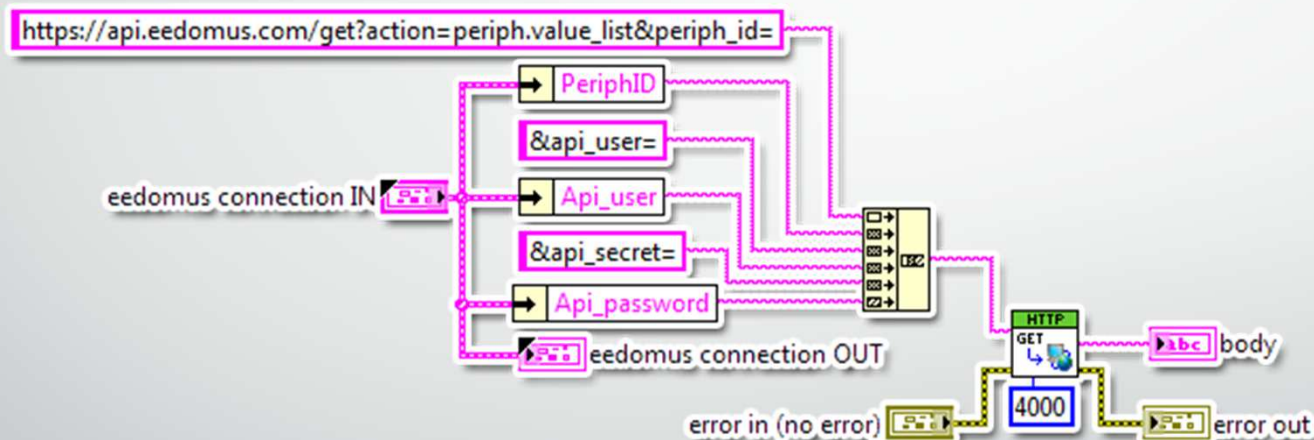
### Utilisation :

- Services d'une application mis à disposition ( accès aux variables, possibilités de modifications...)
- Bibliothèques, services web
- Système de monitoring
- Actions à distance sécurisé possible.

## API WEB: Qu'est ce que c'est ?

- Fonctionne sur des requêtes web (HTTP GET)
- Informations / confirmations = réponses serveurs (bonne prise en compte de l'information)
- Intégration dans labview sous forme de driver?

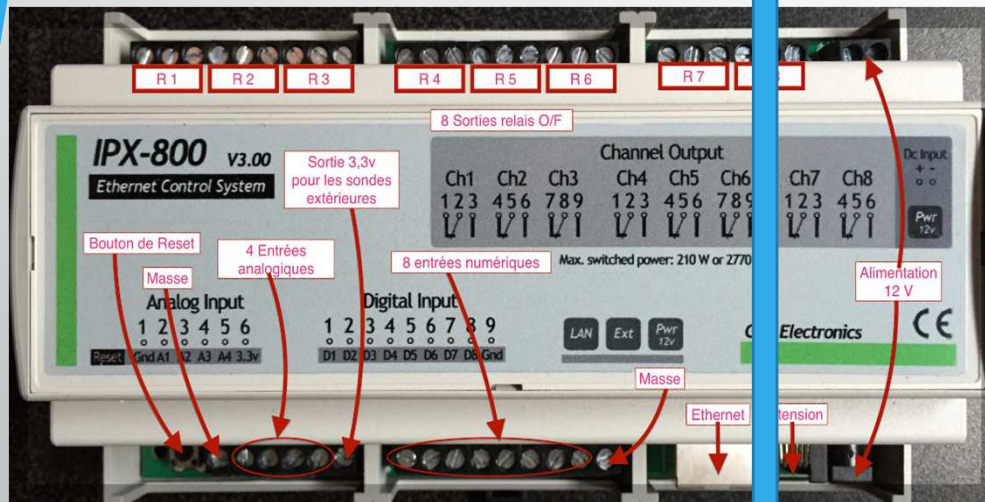
`https://base_url:port ?parametre1=value &parametre2=value &parametre3=value`



IP `https://api.service.com/get?action=periph.value&periph=value&user=user&admin=admin`

## IPX800 V3.0

- Carte web relais
- AI, DI, DO au service de l'instrumentation
- Peu coûteux ( < 200€ )
- Configurable sans programmation
- Fonctionnement autonome.
- API, connectivité (serveur WEB)



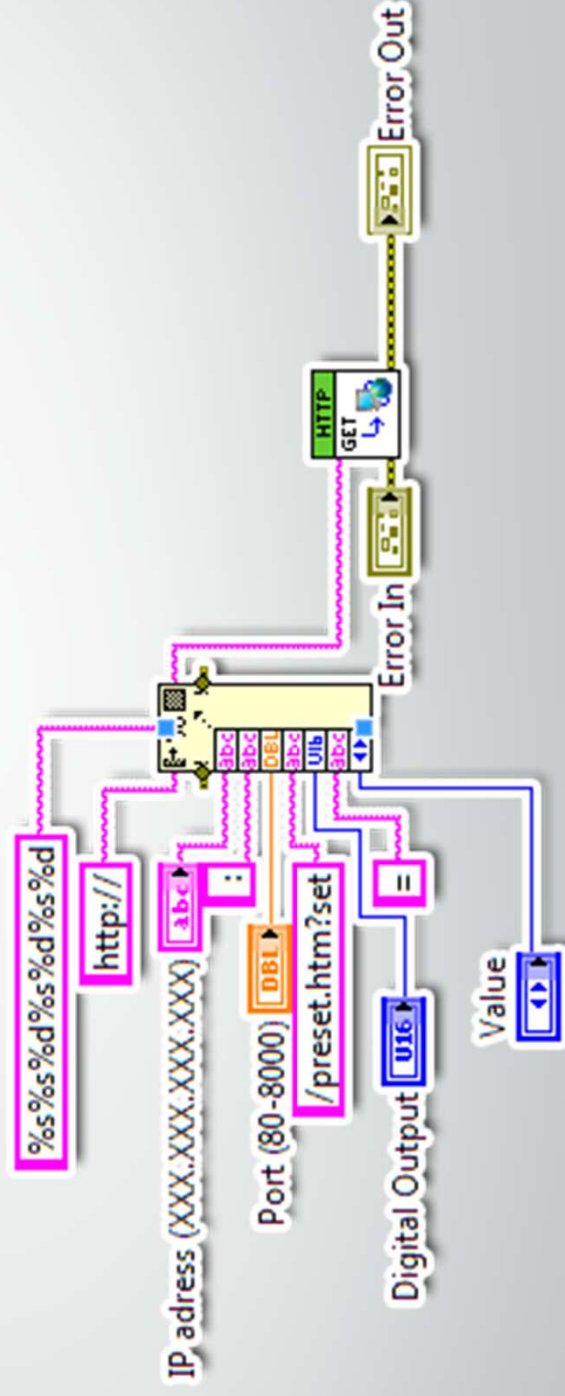
## Développement de drivers

- Récupération de donnée depuis un XML (heure, AI, DI, DO, compteurs...)
- Commande des relais

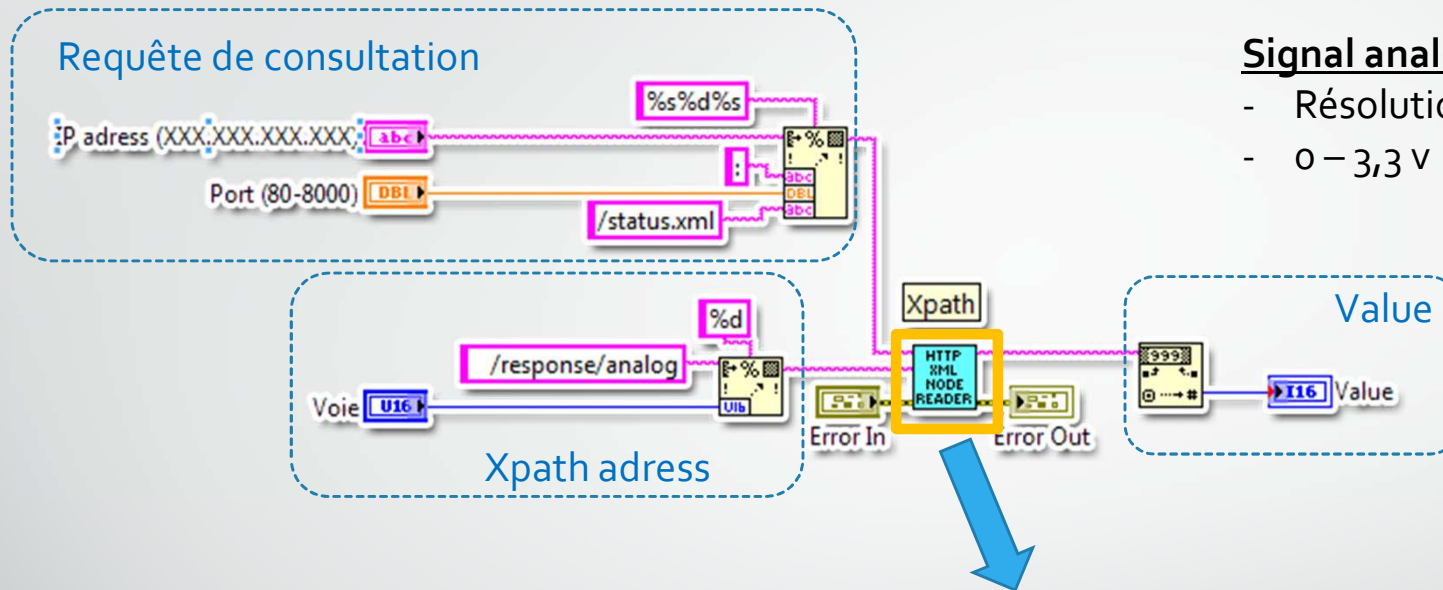


## Ecriture des DO de la carte IPX

IP address (XXX.XXX.XXX.XXX)		Port (80-8000)	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Digital Output		Value	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Error In		Error Out	
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>

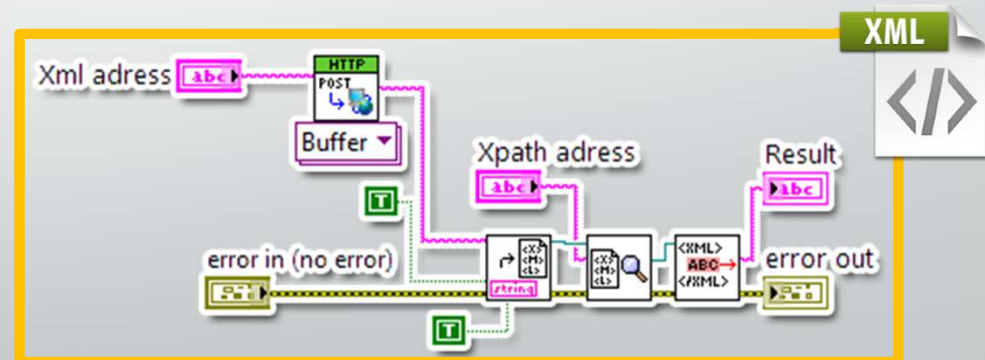


## Récupération de la valeur d'une sortie analogique à partir du fichier XML de la carte IPX : status.xml



### Passage de la réponse :

- String to XML
- Search XML node
- XML value





### 3. Pourquoi développer une API logiciel ?

#### Objectif :

- Faciliter l'accès aux données
- Mise en forme des données (visualisation / création rapport)
- Alerte sur des événements (mail/sms/push)

#### Problématiques :

- Comment rendre disponible les données et sous quelle forme ?
- Comment y accéder de façon sécurisé ?

### 3. Développement d'une API via les web services

#### Objectifs :

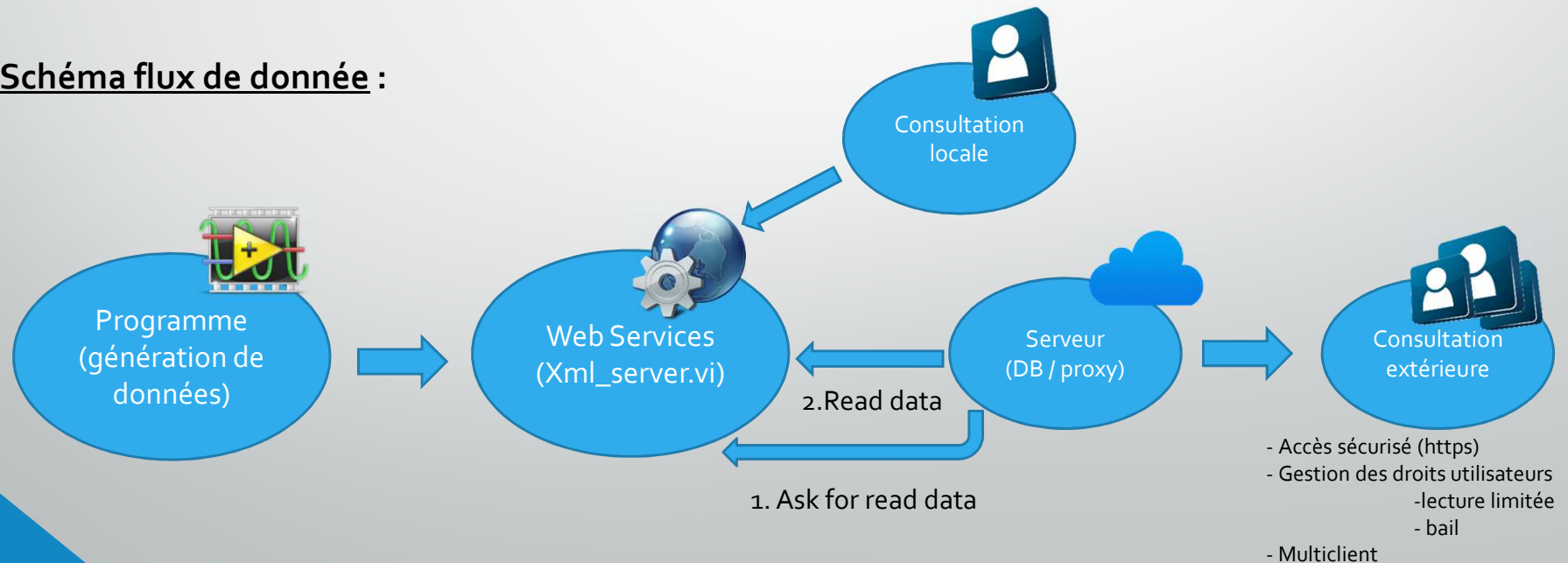
- Accéder aux valeurs générées par le programme
- Consultation uniquement
- En local ainsi qu'à distance (passerelle)
- Haute Interopérabilité avec l'XML

#### Web services :



- Accèdent aux variables grâce aux VG
- Exécution sur une requête

#### Schéma flux de donnée :





## 4. Perspectives / conclusions

- Accessibilité aux API existantes via labview.
- Serveur XML via les web services

### Perspectives (grâce a l'API developpée) !

- Création logiciel de visualisation (Diagramme)
- Création logiciel rapport automatisé.

MERCI POUR  
VOTRE ATTENTION

---



- Drivers IPX
- Serveur XML



## Consultation de donnée et centre alerte

### Interface ordinateur



### Interface mobile (IOS)

IPX800-V3

Ethernet Control System

Output

Input

01/01/2012

00:49:50

Control

1

2

3

4

Out

In

1. Counter Input 1

61

Edit

2. Counter Input 2

0

Edit

Output

1-8

Close

R1 | Commande pompe

R2 | not used

R3 | Relay EV NF

Led | W sensor

Led | Pompe ON/OFF

Led | Secu Oxy

Led | Watchdog | ON | vert

Commande ARU NO

Cam 1

Cam 2

Cam 3

Cam 4

Cam 5

Info.

118-MAT

---

Edit

An.1

Reaction Oven

28

Edit

An.2

Regeneration Oven

25

Edit

An.3

Temp

24.5°C

Edit

An.4

Analog 4

0

Edit

Preset

Timers

DynDNS

LAN settings

Advanced settings

M2M

Upload new site

Copyright : 2014 GCE.ELECTRONICS FRANCE

Select Input channel

Enter channel:1-32

Push & Email configuration

Push settings

Email settings

Input Name:

Assign digital Input to Output

Mode:
☒ On
☐ Off
☐ On/Off
☐ Switch
☐ VR
☐ Inv
 Assign to Counter 1 - 8

Link to Relay

☒ 01
☐ 02
☐ 03
☐ 04
☐ 05
☐ 06
☐ 07
☐ 08
☐ 09
☐ 10
☐ 11
☐ 12
☐ 13
☐ 14
☐ 15
☐ 16
☐ 17
☐ 18
☐ 19
☐ 20
☐ 21
☐ 22
☐ 23
☐ 24
☐ 25
☐ 26
☐ 27
☐ 28
☐ 29
☐ 30
☐ 31
☐ 32

## Input Push Notification

### Select input channel

Enter channel: 1-32

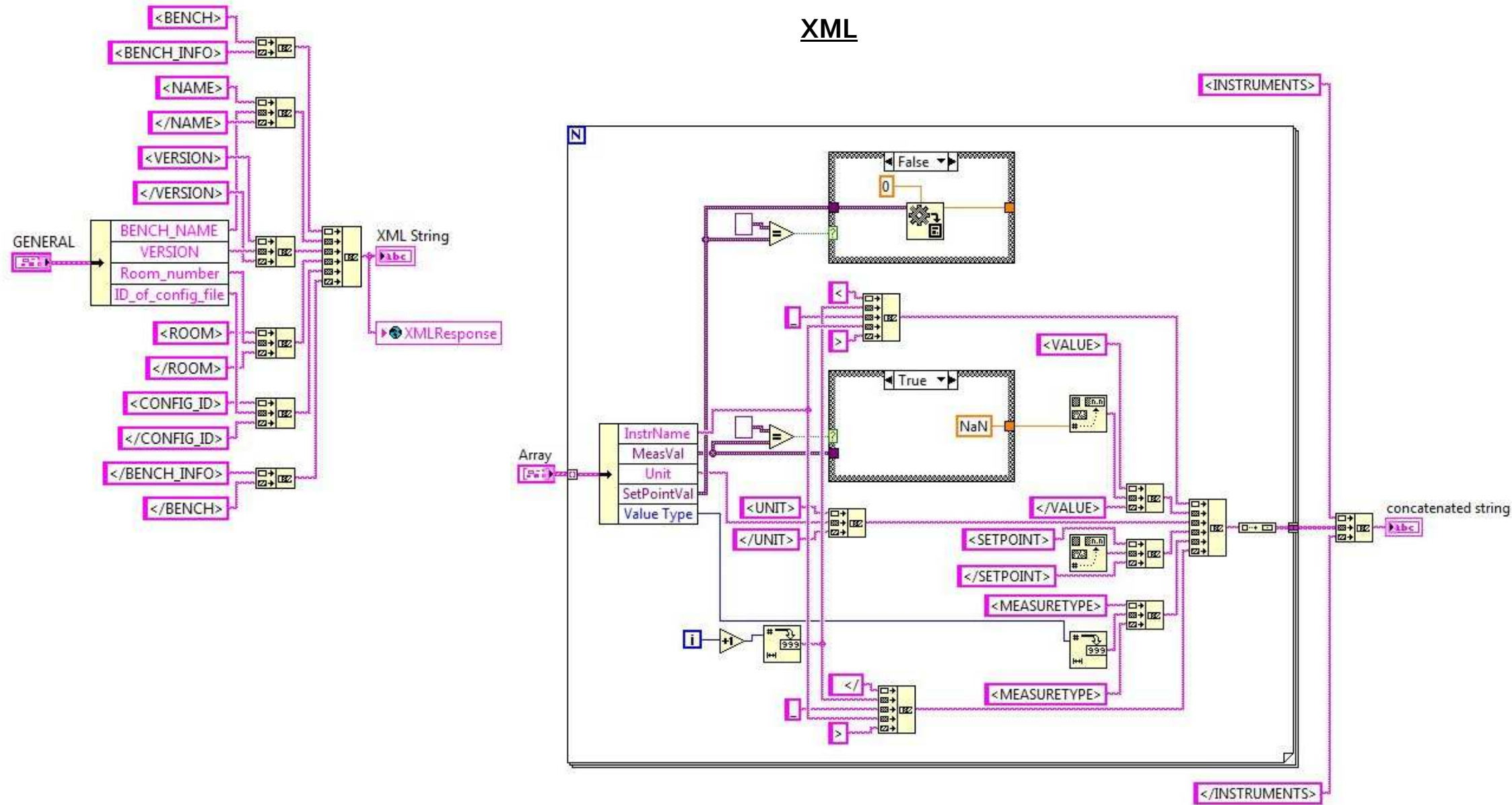
### Push settings

Server: 
 Port: 
 Login: 
☐ Enable

The message 1 is sent when input is closed. The message 2 is sent when input is open.  
 Enter empty if you don't want to send message.

Push 1:	<input style="width: 100%;" type="text" value="/api/set?action=periph.value&amp;periph_id=1&amp;isvalue=100&amp;api_user=1&amp;api_secret=1&amp;update_"/>	<input type="button" value="Save"/>
Push 2:	<input style="width: 100%;" type="text" value="/api/set?action=periph.value&amp;periph_id=1&amp;isvalue=0&amp;api_user=1&amp;api_secret=1&amp;update_"/>	<input type="button" value="Save"/>

## XML





## Exemple interface PID

