

Development of a unified interface for monitoring building heating systems

In-depth Project Spring 22

Guillaume RIONDET

HES-SO Master MSE

6th, July 2022

Hes·SO

MSE

Context

- ▶ Heating installations must fully work all the time
- ▶ Caretakers and technicians perform periodic check-up
- ▶ Issue detected : technicians will resolve it as quick as possible
- ▶ Technicians cannot be everywhere at any time

Problematic

How to monitor and perform remote maintenance on heating stations ?

Summary

Introduction

Technologies

Objectives

Communication protocols

Deployments

Conclusion

Main idea

- ▶ SG-Energies : STEP project
- ▶ Reduce the frequency and duration of breakdowns
- ▶ Limit the unnecessary on-site visit of technicians
- ▶ Add a device to the heating station : SG-Box

Main idea



Edge Device : SG-Box

- ▶ Based on Raspberry Compute Module 4
- ▶ Core component of the system
- ▶ Query the main control unit



Node-RED

► Flow-based development tool

The screenshot displays the Node-RED web interface. At the top, a notification reads "Successfully activated: debug". The interface is divided into several sections:

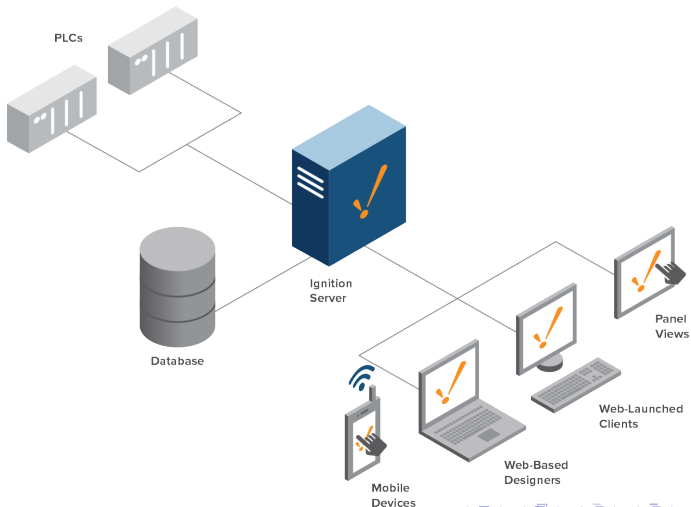
- Left Sidebar:** Contains a search bar "filter nodes" and a list of node categories: "subflows" (Puck Remote), "input" (inject, catch, status, link, mqtt, http, websocket, tcp, udp, serial, Watson IoT), and "output" (screen).
- Central Workspace:** A grid-based area where nodes are connected to form a flow. The flow includes:
 - A "Puck Remote" node connected to a "switch" node.
 - The "switch" node connected to another "switch" node.
 - The second "switch" node connected to four "Off", "Red", "Green", and "Blue" nodes.
 - All four color nodes connected to a "Sense HAT" node.
 - A "Sense HAT" node connected to an "X-axis" node.
 - The "X-axis" node connected to a "set msg.payload" node.
 - The "set msg.payload" node connected to a "trigger 2s" node.
 - A "Sense HAT" node connected to an "X-axis" node.
 - The "X-axis" node connected to a "set msg.payload" node.
 - The "set msg.payload" node connected to a "limit 1 msg/2s" node.
 - The "limit 1 msg/2s" node connected to an "event" node.
 - A "screen" node connected to a "set msg.payload" node.
 - The "set msg.payload" node connected to a "Sense HAT" node.
- Right Sidebar:** A "debug" console showing log messages. The messages are:
 - 0.832: 20/10/2019, 16:55:05 node: 59d5x394.52997c motion : msg.payload.acceleration.y : number
 - 0.8124: 20/10/2019, 16:55:05 node: 59d5x394.52997c motion : msg.payload.acceleration.y : number
 - 0.3244: 20/10/2019, 16:55:06 node: 59d5x394.52997c motion : msg.payload.acceleration.y : number
 - 0.2953: 20/10/2019, 16:55:06 node: 59d5x394.52997c motion : msg.payload.acceleration.y : number
 - 0.8537: 20/10/2019, 16:55:06 node: 59d5x394.52997c motion : msg.payload.acceleration.y : number
 - 1.5159: 20/10/2019, 16:55:06 node: 59d5x394.52997c motion : msg.payload.acceleration.y : number

Node-RED

- ▶ Flow-based development tool
- ▶ Deployed on SG-Box
- ▶ Responsible to query main control units
- ▶ Send retrieved data to Ignition Server

Ignition

► Supervisory control and data acquisition software



Ignition

- ▶ Supervisory control and data acquisition software
- ▶ Store the SG-Box data
- ▶ Later : provide interface for technicians for monitoring

Objectives

1. Discover communication protocols : ModBUS - BACnet
2. Interface main control units with SG-Box device
3. Read retrieved data on Ignition platform
4. Write data through Ignition platform on the main control unit

ModBUS vs BACnet

Criterion	ModBUS	BACnet
Year of creation	1979	1995
Continuous maintenance	No	Yes
Area of use	Programmable logic controllers of any type	Building automation and control systems
Typing	Boolean or Integer 16 bits	Many types available
Data availability	Value only	Value + many optional fields
Command prioritization	No	Yes

Deployments

▶ ModBUS

1. Vernier : Viessmann
2. Chancy (Bellegarde road) : Froeling

▶ BACnet

1. Geneva (Yvoi Boulevard) : Sauter

Step by step methodology

- ▶ Read one data point
- ▶ Read multiple data points in one time
- ▶ Write one data point
- ▶ Write multiple data points

Step by step methodology

1. Partial workflow : Main control unit - NodeRED
2. Full workflow : Main control unit - NodeRED - Ignition server

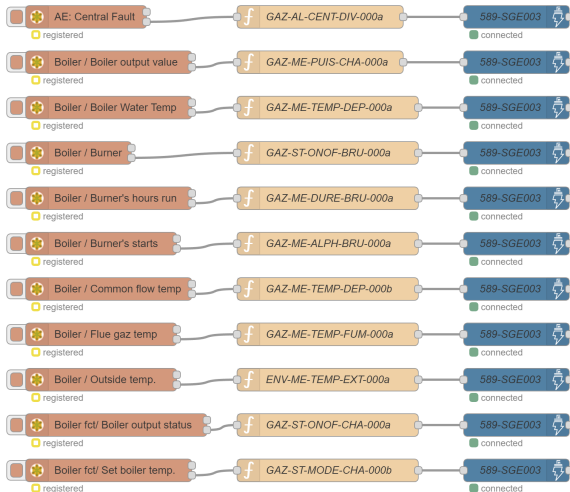
Step by step methodology : Read multiple data points

- ▶ Naive method : Duplicate the read of one data point multiple times



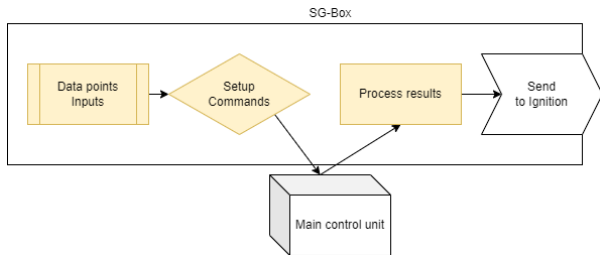
Step by step methodology : Read multiple data points

- ▶ Naive method : Duplicate the read of one data point multiple times



Step by step methodology : Read multiple data points

- ▶ Generic method
 - ▶ Same principles for both protocols : only tools are adjusted



Deployments progress

- ▶ ModBUS : Vernier and Chancy
 - ▶ Data points reading through Ignition : Ok
 - ▶ Data points writing through Ignition : Partially ok
- ▶ BACnet : Geneva (Yvoi boulevard)
 - ▶ Data points reading through Ignition : Ok
 - ▶ Data points writing through Ignition : Partially ok

Issues encountered

- ▶ Instability of the 4G network and dongle
- ▶ Network : Conflict between same subnets
- ▶ Network : Prioritization between 4G network interface and Ethernet interface

Common solution added

- ▶ VPN on SG-Box devices
 - ▶ Remote access and update
 - ▶ Unique credentials for each SG-Box
 - ▶ Setup procedure for further installation



Conclusion

- ▶ Proof-of-concept of the whole system
- ▶ Generic viable solution
- ▶ One of each type of deployment are currently running

Future perspectives

- ▶ Improve input method in NodeRED flow
- ▶ Multiple writing
- ▶ Development of the interface to visualize retrieved data and monitor remotely

Auto-Evaluation

- ▶ Discovery of a new world
- ▶ Concrete project
- ▶ Be more productive on documentation writing
- ▶ Having more time could lead to a better optimization of the multiple read solution