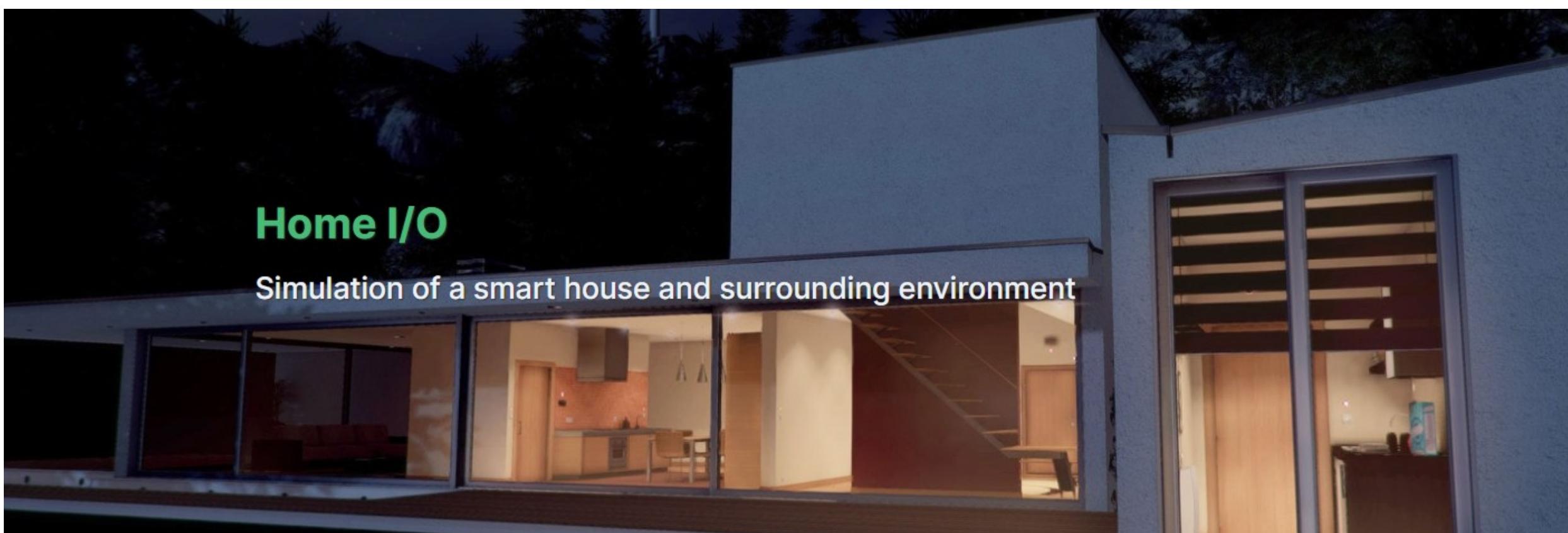
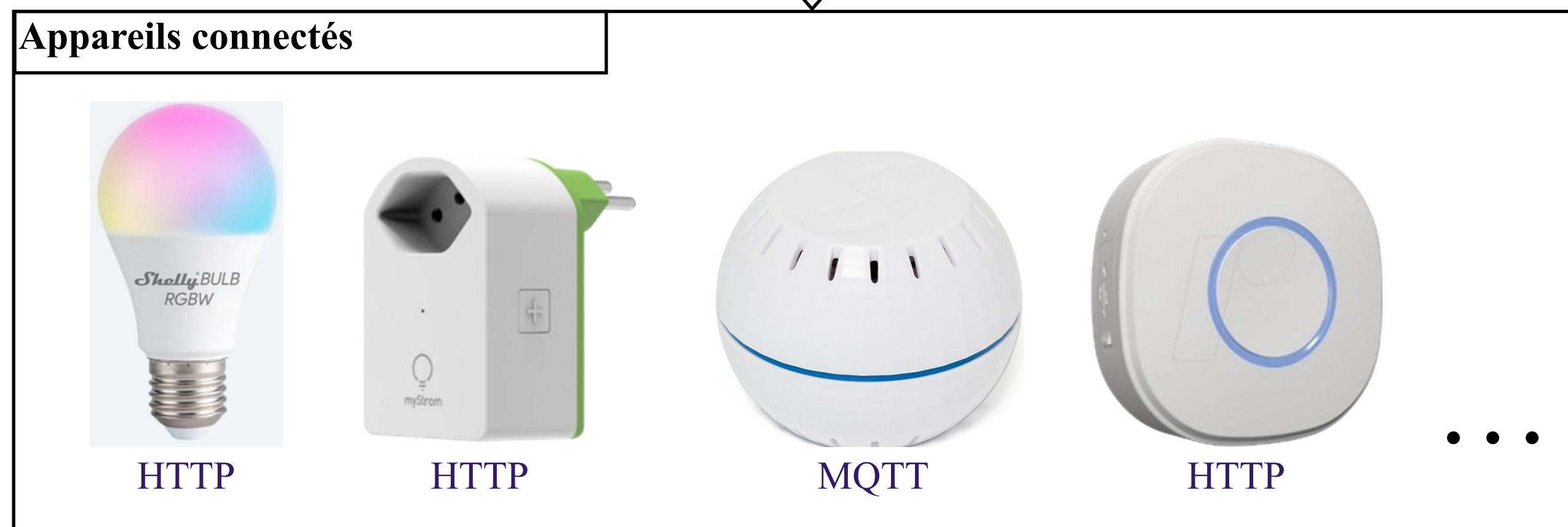
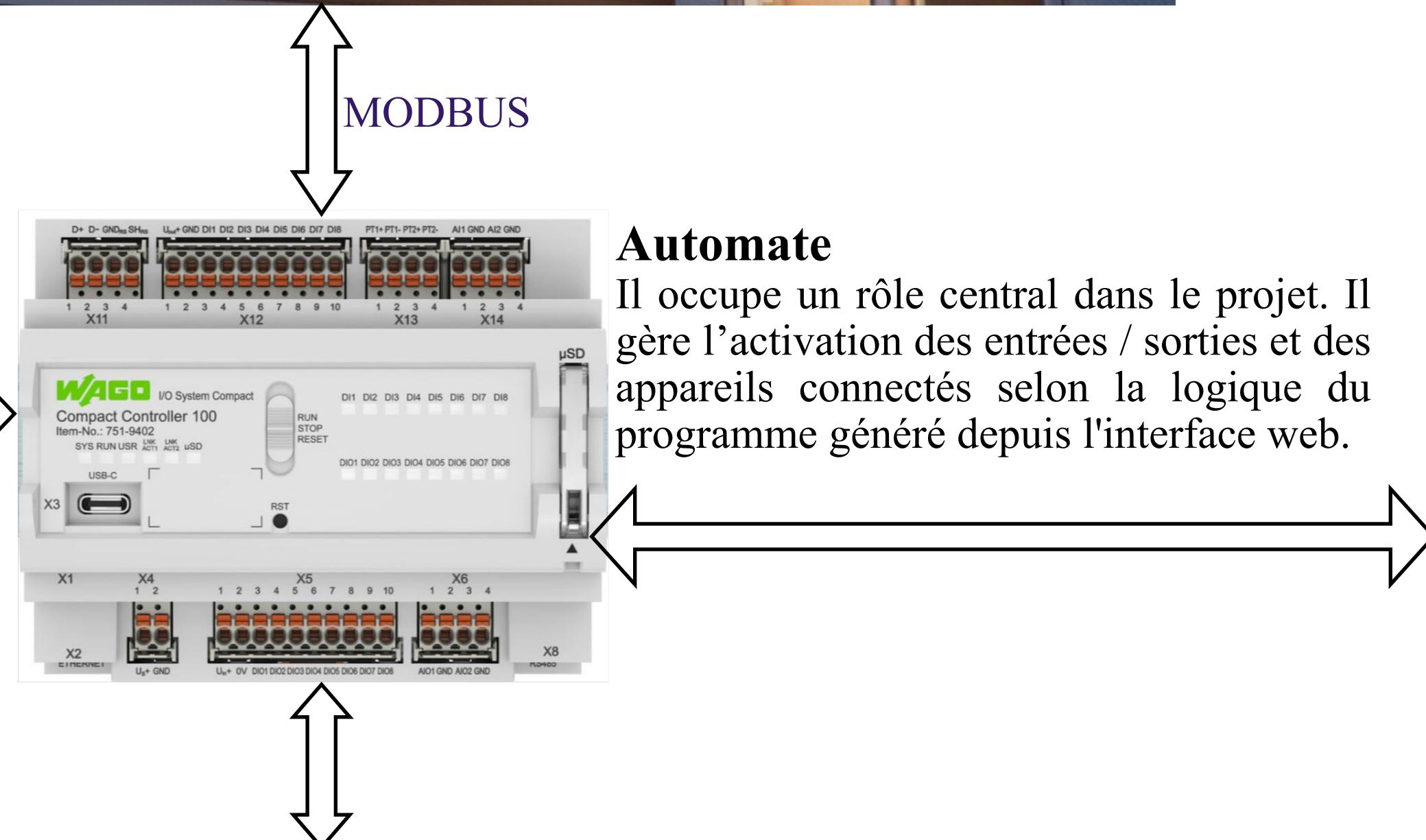


# SoftPLC pour IoT

L'entreprise WAGO, qui commercialise des automates, a mandaté la HES-SO afin de réaliser un nouveau HAL (Hardware Abstraction Layer) pour ses nouvelles interfaces des PLC WAGO CC100 (751-9401 et 751-9402). L'objectif est de permettre aux automaticiens de programmer de manière simple via une page web, tout en leur donnant la possibilité de réaliser des tâches complexes telles que la communication HTTP, MQTT, MODBUS, ainsi que d'autres fonctions avancées. Cela permettra l'intégration de systèmes IoT, en facilitant la mise en œuvre de communications et de fonctions connectées directement depuis l'interface web.



- Appareils câblés**
- Boutons
  - Lampes
  - Potentiomètres
  - Electrovanne proportionnelle
  - Etc.



**Interfaces : Programmation (1) / Débogage (2) / Utilisateur (3)**

**(1)**

The logic diagram consists of several blocks: Input (DI1), RF\_trig, AND (&), Input0, Input1, Input2, Constant value Input (1000), TOF, and Output (DO1). The logic flow starts with DI1 connected to Input0. Input0 is connected to Input1 through an AND gate. Input1 is connected to Input2. Input2 is connected to a TOF block. The TOF block has a Time [ms] input and an Output. The Output of the TOF block is connected to a Constant value Input (1000). The output of the Constant value Input is connected to Input1. The output of Input1 is connected to Input2. The output of Input2 is connected to the TOF block. The output of the TOF block is connected to the Output (DO1).

**(2)**

This screenshot shows a more complex logic diagram involving an HTTP Client node. It includes blocks like Rtrig, Ftrig, Not, bool to string, Find, SR, and various inputs and outputs related to a heater's relay state and temperature.

**(3)**

The user interface includes a "Enter message" field and a "Send" button. Below it, there are sections for "Main" (DI1 confirmation), "Heater" (Relay On/Off), and "States". The "States" section lists "Relay state : ON" and "Relay response : {power:0,Ws:0,relay:true,temperature:25.9}".

**Extrait code derrière (golang)**

```

38 func (n *SRNode) ProcessLogic() {
39     if n.input == nil {
40         n.output[0].output = "0"
41         return
42     }
43     if n.input[0].Input == nil || n.input[1].Input == nil {
44         n.output[0].output = "0"
45         return
46     }
47     if *n.input[1].Input == "1" {
48         n.output[0].output = "0"
49     } else if *n.input[0].Input == "1" {
50         n.output[0].output = "1"
51     }
52 }
53

```