Domotique avec Home Assistant

C'est ici que vous trouverez les animations autour de la domotique avec Home Assistant !

- Mon premier micro-controleur avec ESPHome (D1_mini)
- Mesurer la temperature / l'humidité avec ESPHome
- Utiliser son ESP comme un interrupteur !
- Configurer son module esp32Cam
- Créer une automatisation
- Guide pédagogique "Introduction à HA et arrosage du potager"

Mon premier micro-controleur avec ESPHome (D1_mini)

ESPHome c'est quoi ?

`Le projet ESPHome est un projet communautaire permettant de créer des automatismes intégrable dans un serveur Home Assistant à base de microcontrôleurs type ESP :`

Le lien du projet ES

https://esphome.io/

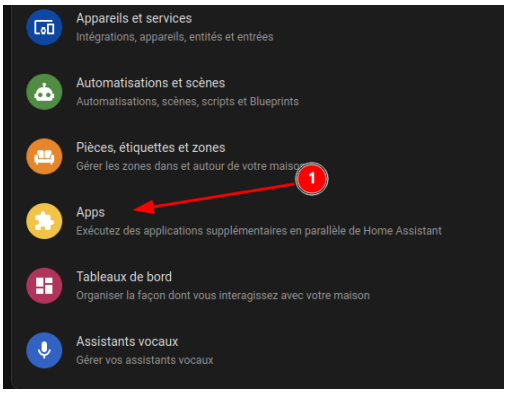
https://esphome.io/components/

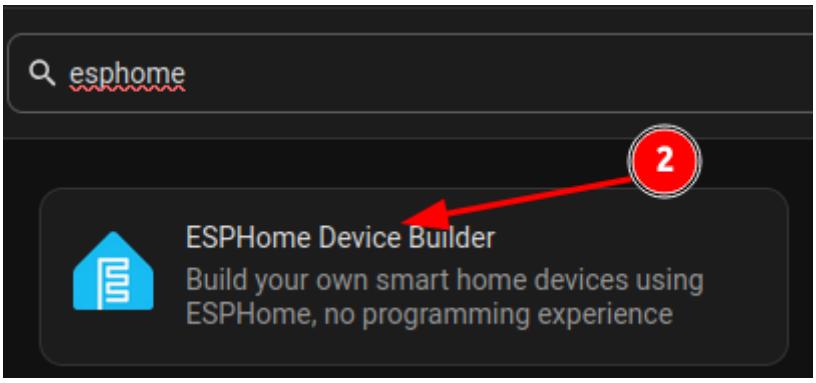
Quelques infos sur les microcontrôleurs type ESP :
https://fr.wikipedia.org/wiki/ESP32

https://fr.wikipedia.org/wiki/ESP8266

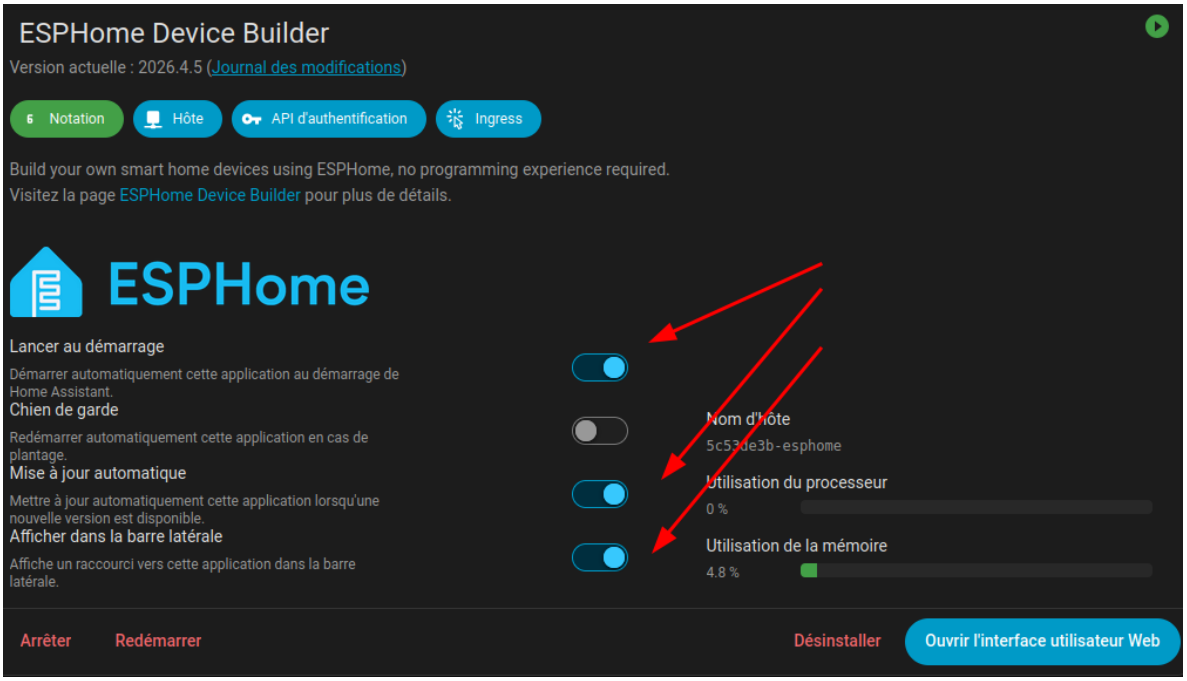
Comment ça s'installe ?

Une fois logué sur son serveur Home Assistant préféré, on va dans les **paramètres**, on clique sur **Apps** (qui nous sert à rajouter des applications tierces dans HA) et on trouve **ESPHome** dans la barre de recherche.



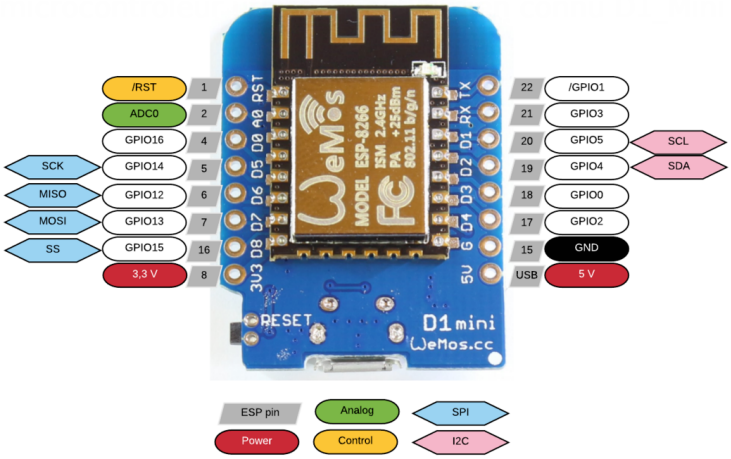


Une fois le logiciel installé, on n'oublie pas d'activer quelques paramètres recommandés comme le **démarrage automatique** d'ESPHome, les **mises à jours auto** ainsi que le r**accourcis dans la barre latérale** d'Home Assistant.



Le microcontrôleur

Le [illegible] basé sur un ESP8266)



Pour une description plus complète, on vous renvoi vers le wiki de nos copains brestois, les petits Débrouillards : https://www.wikidebrouillard.org/wiki/Item:D1_mini

Mon code YAML pour débuter

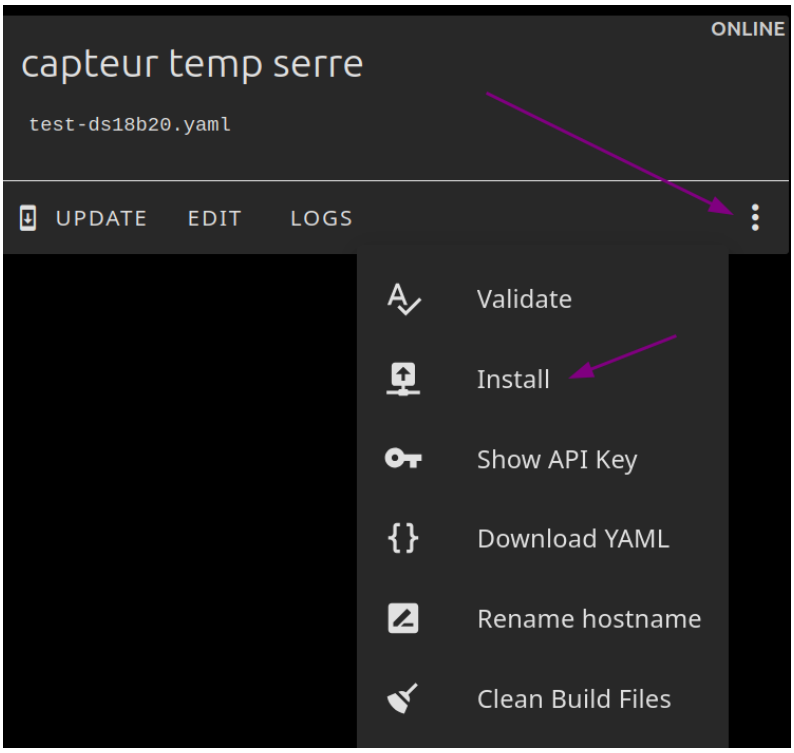
Voici un exemple de code en YAML pour permettre (avant tout autre rajout type capteur, sonde, interrupteur) à notre carte de se connecter à notre serveur Home assistant via ESPHome.

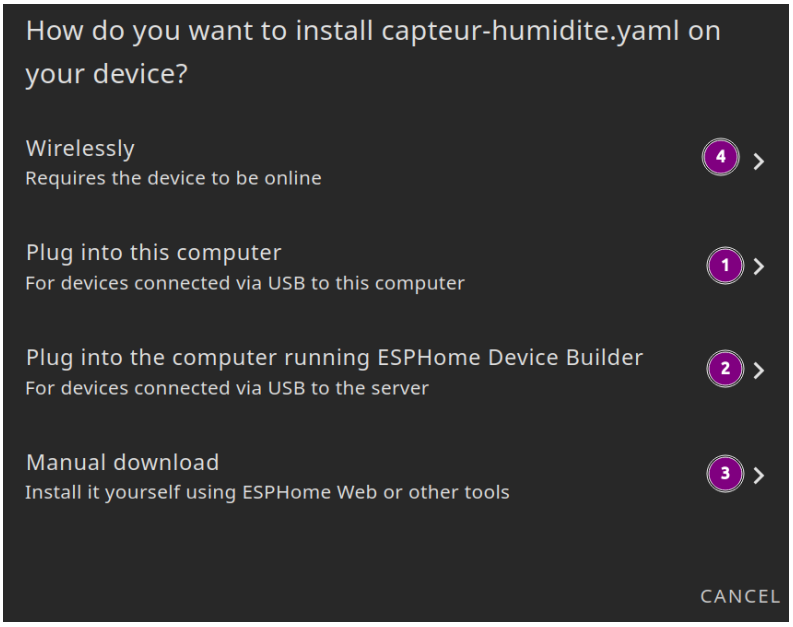
> Attention, la clé API est celle de votre ESPHome.
> Il ne faudra pas oublier non plus de saisir le nom de sa box (SSID) et le mot de passe correspondant dans l'onglet "secrets" en haut à droite de l'interface ESPHome

```
esphome:
  name: capteur-temp-serre
  friendly_name: capteur temp serre
```

```
esp8266:
  board:  d1_mini

# Active les journaux (logs) pour le débogage ()
logger:

# Permet à Home Assistant de découvrir l'ESP automatiquement
api:
  encryption:
    key: "mCFuxUyPOZ2HNTax72TKoCeG3Btb0j0qsG7pWzBBt+k="

# Permet de faire les futures mises à jour sans fil (OTA = Over The Air)
ota:
  - platform: esphome
    password: "0d9662546dc9759763b213a7aa5f3b7c"

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  # Génère un point d'accès de secours si le Wi-Fi coupe
  ap:
    ssid: "Capteur-Temp-Serre"
    password: "tregor22300"

captive_portal:
```

Pour téléverser ce premier programme dans notre microcontrôleur, on clique sur "Install" depuis l'item crée :





1. En branchant son ESP en USB sur son **PC** (marche bien sous Windows, moins sous Linux)
2. En branchant son ESP en USB sur le directement sur le **serveur** Home Assistant
3. En téléchargeant le programme (.bin) sur son ordinateur et en le televersant avec un locigiciel comme ESPtool.
   Dans ce cas, on ouvrira le terminal depuis le dossier contenant le .bin et on lancera ces 3 commandes :

```
sudo apt install esptool    #pour installer l'application
sudo dmesg | grep tty       #pour vérifier le port utilisé par le périphérique (souvent ttyUSB0)
sudo esptool --port /dev/ttyUSB0 --baud 460800 write_flash 0x00000 capteur-humidite.bin  #la commande de televersement en precisant bien le bon port (ttyUSB0 dans notre cas) ainsi que le
```

```
nom du fichier.bin (capteur-humidite.bin)
```

Par la suite, quand on modifiera/rajoutera des choses dans notre programme, on pourra le faire directement en wifi (4) !

> On ne peut téléverser en wifi que si on a un paragraphe OTA défini dans notre programme !

Mesurer la temperature / l'humidité avec ESPHome

WORK IN PROGRESS/

Mettre en place une mesure de température en passant par un capteur type **DS18B20**

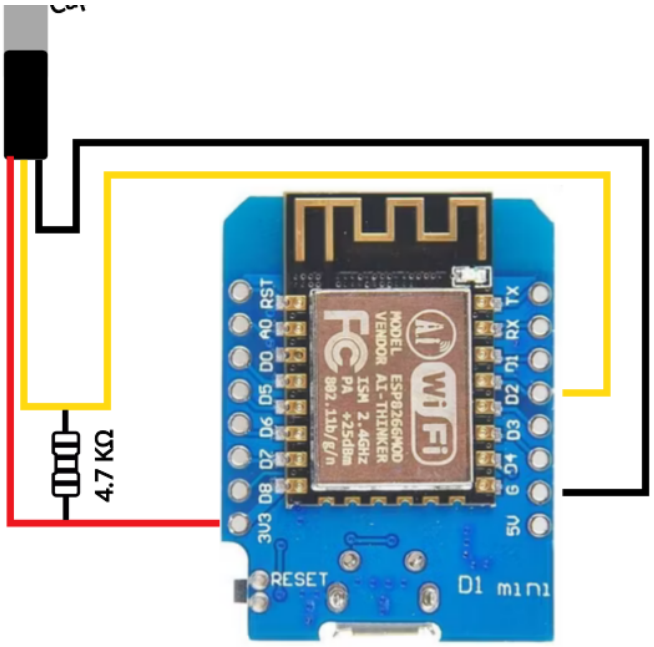
De quoi on parle ?

Le capteur de température DS18B20 est un capteur très utilisé dans les montages électronique. Il s'alimente de **3 à 5 Volts** et a une plage de mesure de **-55° à +125°**
Même dans la serre, à plus de 50° : il résiste !
De plus, dans sa forme encapsulée (comme sur la photo), il est **étanche**.

Notre objectif sera donc de l'interfacer avec un micro controleur ESP (D1_mini dans notre exemple) pour remonter ses données vers notre serveur Home Assistant, en passant par ESP Home.

Voir Page "Mon premier micro-controleur avec ESP Home"

La soudure :



ESP8266 D1_Mini

Une fois équipé de son fer à souder préféré, on soudera le fil rouge (la phase du capteur) sur la broche 3V3 de l'ESP, le fil de donnée (le jaune en général) sur, par exemple, la PIN D2. Le fil noir (la masse du capteur) sera quand à lui soudé sur une des PIns GND (Ground)

On soudera également, entre le fil de donnée et la phase, une résistance de 4.7Kohm (pour stabiliser la tension/le passage des données.

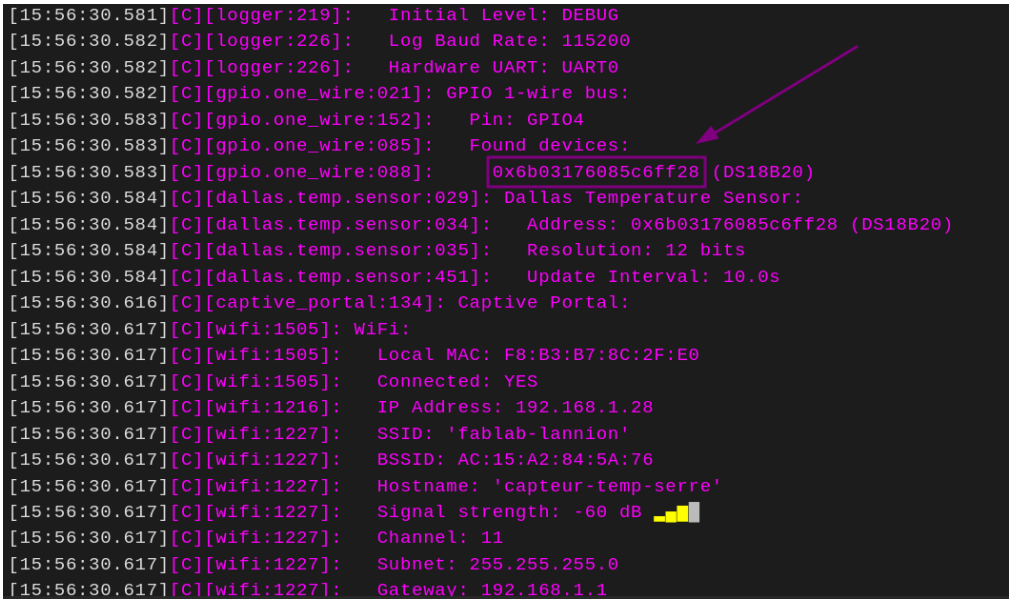
plan_soudure_DS18B20.svg

Programmer l'ESP et remonter les Données

Une fois les composants soudés, on va passer par l'interface d'ESPHome pour charger du code dans le micro-contrôleur.
Si notre micro-contrôleur est déjà reconnu et en capacité de communiquer en Wifi (voir module précédent) , on aura qu'a rajouter un bloc de code dans le fichier YAML pour communiquer avec le capteur de température.

```
one_wire:               #précise qu'on utilise le protocole OneWire
  - platform: gpio
    pin: GPIO4          #La pin D2 du micro-controleur, correspondant à GPIO04 : voir plan du
premier module


sensor:
  - platform: dallas_temp                 #  On précise utiliser un capteur type
dallas_temp
    name: "Temperature DS18B20"           # On lui choisi un nom
    update_interval: 10s                  # On lui demande de remonter la temperature vers le
serveur toutes les 10 secondes
```

Plusieurs capteurs de température sur une carte?



Le protocole 1-Wire (https://fr.wikipedia.org/wiki/1-Wire) permettant de connecter plusieurs capteurs en même temps, il conviendra de récupérer l'adresse de chaque capteur connecté (dans les Logs) pour ne pas les mélanger.

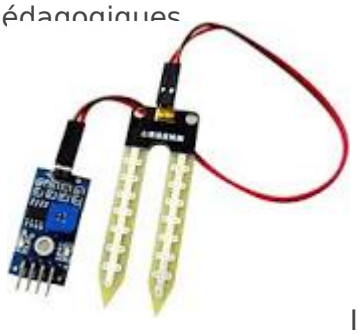
Il n'y aura plus qu'à ajouter un bloc de code par capteur dans le YAML correspondant

```
sensor:
  - platform: dallas_temp
    address: 0x123456789ABCDEF  #adresse numérique que l'on a récupéré dans les Logs
    name: "Temperature Salon"
```

Utiliser un capteur d'humidité type HW-080

De quoi on parle ?

Le capteur HW-080 est un grand classique dans la famille des capteurs utilisés dans les kits pédagogiques.



Il s'agit d'une résistance qui varie en fonction de la conductivité du sol.

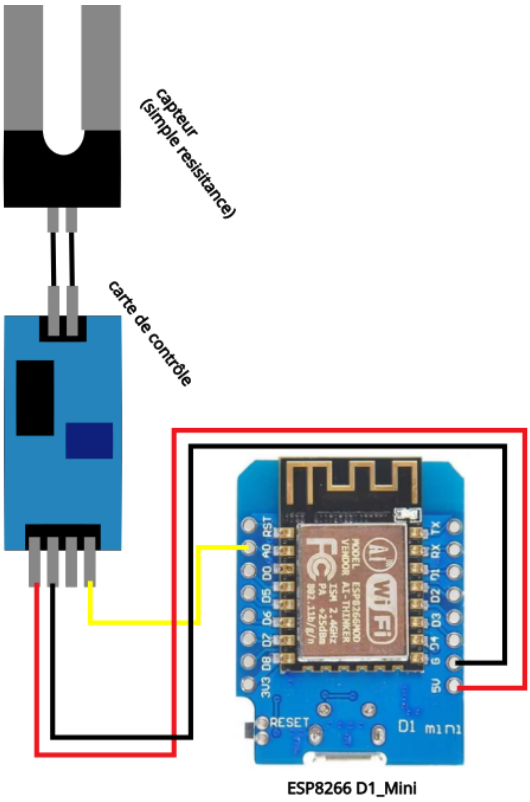
Le HW-080 possède une sortie analogique qui va lire une tension entre 0V et 3.3V et la convertir en valeur numérique.

- **Plus la terre est sèche**, plus la résistance est forte, plus la tension de sortie (**AO**) est **proche de VCC** (tension max).
- **Plus la terre est mouillée**, plus l'eau conduit l'électricité, plus la résistance baisse, et plus la tension **descend vers 0V**.

La soudure

La partie soudure est assez classique : un +, un - et fil de données !

=> Le fil rouge (couleur par conv ur aller au VCC
=> Le fil noir raccordera lui les G
=> Le fil jaune quand a lui, servir



ESP8266 D1_Mini

une entrée analogique de l'ESP, A0 dans notre cas.

> Ne pas hésiter à se référer au Pin-out de l'ESP !

plan_soudure_HW_080.svg

Le code YAML

L'étape suivante va donc être de rajouter du code à un ESP communiquant déjà en wifi avec notre interface ESPHome.

```
sensor:
  - platform: adc
    pin: A0
    name: "Humidité du sol - Valeur brute"
    id: humidite_sol_brute
    unit_of_measurement: "" # On enlève le "V" qui induit en erreur
    accuracy_decimals: 0
    update_interval: 5s     # 5 secondes pour que ce soit réactif en atelier

    filters:
      # Étape indispensable sur ESP8266 pour récupérer la valeur brute de 0 à 1023
      - multiply: 1023.0

      # Calibration "Tout ou rien" à ajuster avec vos tests :
      - calibrate_linear:
          # Format : valeur_brute -> pourcentage_voulu
          - 850 -> 0.0    # À l'air libre (Sec)
          - 250 -> 100.0  # Dans un verre d'eau (Humide)

  - platform: template
    name: "Humidité du sol"
    unit_of_measurement: "%"
    icon: "mdi:water-percent"
    accuracy_decimals: 1
    lambda: |-
      return id(humidite_sol_brute).state;
```

```
sensor:
  - platform: adc
    pin: A0
    name: "Humidité du sol - Valeur brute"
    id: humidite_sol_brute
```

```
    unit_of_measurement: "" # On enlève le "V" qui induit en erreur
    accuracy_decimals: 0
    update_interval: 5s     # 5 secondes pour que ce soit réactif en atelier

    filters:
      # Étape indispensable sur ESP8266 pour récupérer la valeur brute de 0 à 1023
      - multiply: 1023.0

      # Calibration "Tout ou rien" à ajuster avec vos tests :
      - calibrate_linear:
          # Format : valeur_brute -> pourcentage_voulu
          - 850 -> 0.0    # À l'air libre (Sec)
          - 250 -> 100.0  # Dans un verre d'eau (Humide)

  - platform: template
    name: "Humidité du sol"
    unit_of_measurement: "%"
    icon: "mdi:water-percent"
    accuracy_decimals: 1
    lambda: |-
      return id(humidite_sol_brute).state;
```

Utiliser son ESP comme un interrupteur !

WORK IN PROGRESS/

Utiliser son ESP comme un interrupteur

De quoi on parle ?

L'idée ici sera de rendre l'appareil (une pompe, un volet roulant, un moteur) **actionnable à distance**.
A partir de là, il sera possible :

- Que l'utilisateur déclenche son appareil à partir de son ordinateur ou de son smartphone.
- Bien plus puissant, que l'appareil soit allumé/éteint automatiquement à partir d'une condition, par exemple :
  - en fonction de la température
  - de la production électrique
  - de l'hygrométrie...

Faisons allumer une LED à distance

La soudure

Le code YAML

Voici un exemple de bloc que l'on peut rajouter pour utilisation d'un actionneur

Extrait de la documentation https://esphome.io/components/light/

```
output:
  - platform: esp8266_pwm
    pin: GPIO4          #Pin GPIO utilisée (Voir Pinout de la carte)
    id: led_d1mini     #Nom

light:
  - platform: monochromatic         #Type d'éclairage  utilisé (LED simples, bande de led,
bande adressable...
    name: "LED D1 Mini"
    output: led_d1mini
```

Configurer son module esp32Cam

```
esphome:
  name: esp32cam-fablab
  friendly_name: ESP32 CAM Fablab

esp32:
  board: esp32cam
  framework:
    type: arduino
psram:


# Logs
logger:

# API Home Assistant
api:
  encryption:
    key: "Votre clé API qui sert à communiquer avec le logiciel ESPHome"

# OTA
ota:
  - platform: esphome
    password: "motdepasseota"

wifi:
  ssid: !secret wifi_ssid
  password: !secret wifi_password

  ap:
    ssid: "ESP32CAM-Secours"
```

```
    password: "12345678"

captive_portal:

# Web interface optionnelle
web_server:
  port: 80

# Configure les réglages de la caméra pour l'interface web
#esp32_camera_web_server:
#  - port: 8080
#    mode: stream



# Flash LED
output:
  - platform: gpio
    pin: GPIO4
    id: flash_led

light:
  - platform: binary
    name: "ESP32 CAM Flash"
    output: flash_led

# Caméra
esp32_camera:
  name: "Camera Fablab"

  external_clock:
    pin: GPIO0
    frequency: 20MHz

  i2c_pins:
    sda: GPIO26
    scl: GPIO27

  data_pins:
    - GPIO5
    - GPIO18
```

```
    - GPIO19
    - GPIO21
    - GPIO36
    - GPIO39
    - GPIO34
    - GPIO35

  vsync_pin: GPIO25
  href_pin: GPIO23
  pixel_clock_pin: GPIO22

  power_down_pin: GPIO32

  resolution: 1280x1024
  jpeg_quality: 10

  max_framerate: 15 fps
  idle_framerate: 0.2 fps

  vertical_flip: false
  horizontal_mirror: false

# Signal WiFi
sensor:
  - platform: wifi_signal
    name: "ESP32 CAM WiFi Signal"
    update_interval: 60s

# Redémarrage
button:
  - platform: restart
    name: "ESP32 CAM Restart"
```

Créer une automatisation

De quoi parle ?

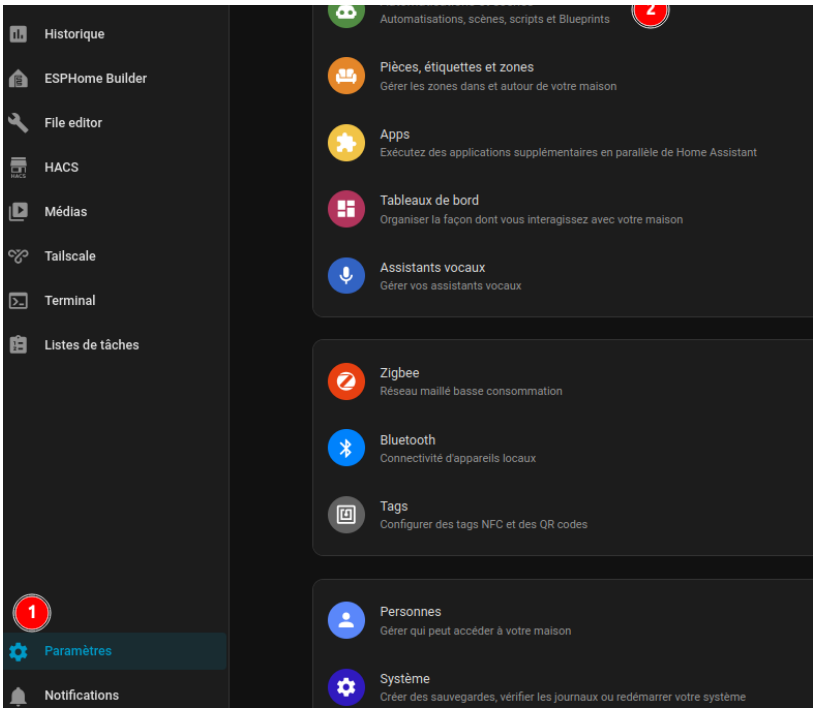
Maintenant que l'on monitore différentes données (température, humidité,...) et que l'on a des actionneurs, on va pouvoir créer des automatismes.
Insérer ici, des exemples d'automatisations réalisées chez vous ou ailleurs:

- Déclencher l'arrosage de la serre (avec un relais) pendant 30secondes toutes les 24heures si la température est inférieure à 30 °
- Si j'actionne tel bouton, alors ouvrir mes volets roulants
- SI la production de mes panneaux solaires est supérieures à 600 watts, alors déclencher la résistance du chauffe-eau
- ....

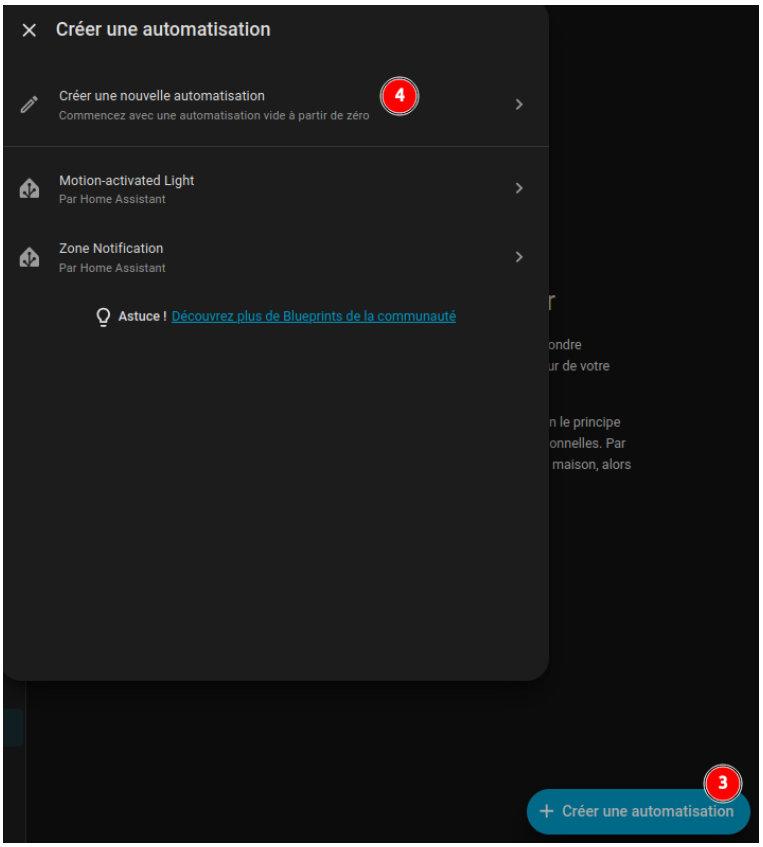
L'automatisme

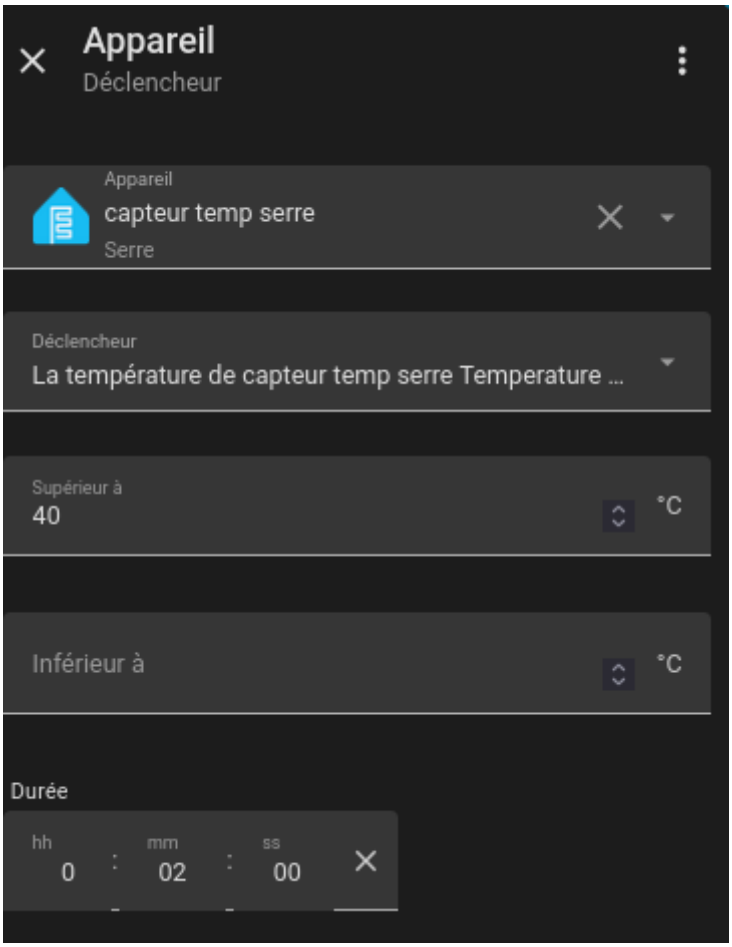
Pour l'exemple, prenons un besoin :
"je souhaite allumer une LED rouge quand la température dépasse les 40° dans ma serre !"
Rien qu'avec ce besoin exprimé, on a déjà la logique du programme à construire :

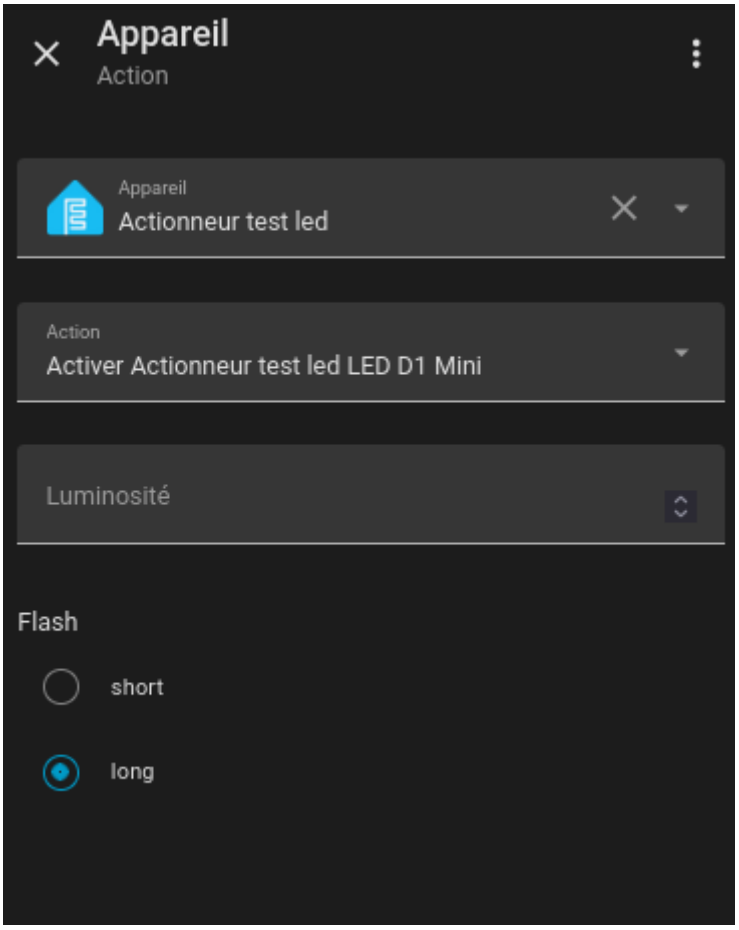
```
==> Si (température de la serre) >= 40° :
          Alors : allumer (LED)
```



Avec Home assistant, point besoin d'écrire des lignes de codes ; dans notre interface Home Assistant, on se rend dans le panneau "**Paramètres**" puis "**Automatisations et scènes**". Ensuite, on clique sur "**Créer une automatisation**" et "**Créer une nouvelle automatisation**".




Appareil
Déclencheur
Appareil
capteur temp serre
Serre
Déclencheur
La température de capteur temp serre Temperature ...
Supérieur à
40
°C
Inférieur à
°C
Durée
hh
0
mm
02
ss
00


Appareil
Action
Appareil
Actionneur test led
Action
Activer Actionneur test led LED D1 Mini
Luminosité
Flash
short
long

Guide pédagogique "Introduction à HA et arrosage du potager"

Guide pédagogique "Introduction à HA et arrosage du potager"

□□ Objectifs principaux :

- Introduire le rôle d'un serveur domotique Home Assistant
- Introduire le fonctionnement d'un microcontrôleur et de ESP-Home
- Être en capacité de Nommer quelques automatismes possibles

□□ Public cible :

- Tous publics

□□ Pré-requis :

- Savoir utiliser clavier/souris
- Avoir quelques notion du fonctionnement d'un réseau local

□□ Durée de l'atelier : 2h

□□ Matériel à préparer

Matériel	Consommables
Projecteur	
Ordinateurs x participants (si pas de pcs perso)	
connexion internet	
Serveur HA connecté	
Module ESP_température	
Module ESP_hygrométrie	
Module ESP_LED	
scanner d'IP (Angry IP scanner)	

Retrouvez ici les .bin des 3 capteurs (partie wifi seulement) :

actionneur_led.bin

capteur-humidite.bin

capteur-temp-serre.bin

□□ Déroulé de l'animation

Introduction *20mn*

*L'animateur **propose** un tour du Fablab. Il **introduit** le sujet en **provoquant de l'échange** sur les connaissances et essais de chacun.*

Présentation de Home Assistant *20mn*

L'animateur **interroge** autour de la domotique :
"*Est que vous connaissez ?*"
"*Des retours d'expériences ?*"

L'animateur **projette l'interface** de Home Assistant, il **rappelle** le fonctionnement d'un **serveur**
L'animateur **présente** l'interface :
-Le Desk
-Les Apps
-Appareils et services

-Automatisation

les **capteurs du Fablab**, ainsi que différentes possibilités.
Projets concrets présentables :

mesure de conso video conso Linky

Mesure de production Routeur solaire

TP : trouver mon serveur HA sur le réseau
L'animateur rappelle le fonctionnement d'un réseau. Les apprenants scannent le réseau du fablab pour trouver différents périphériques.
Dans le navigateur :
==> adresseIP: 8123
==> (Sinon DNS) homeassistant.local:8123

Potager

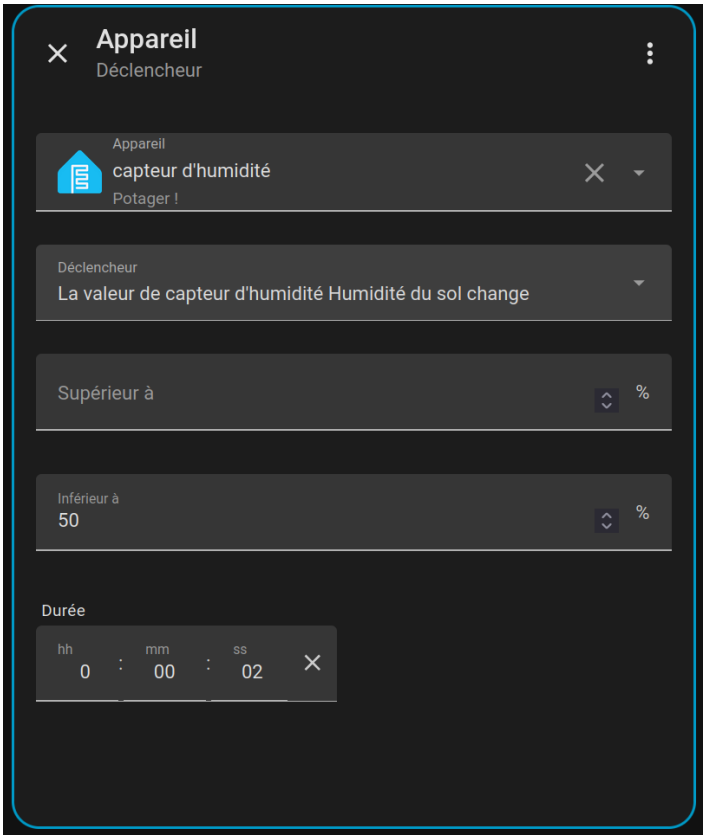
Présentation des microcontrôleurs et d'ESP Home *20mn*

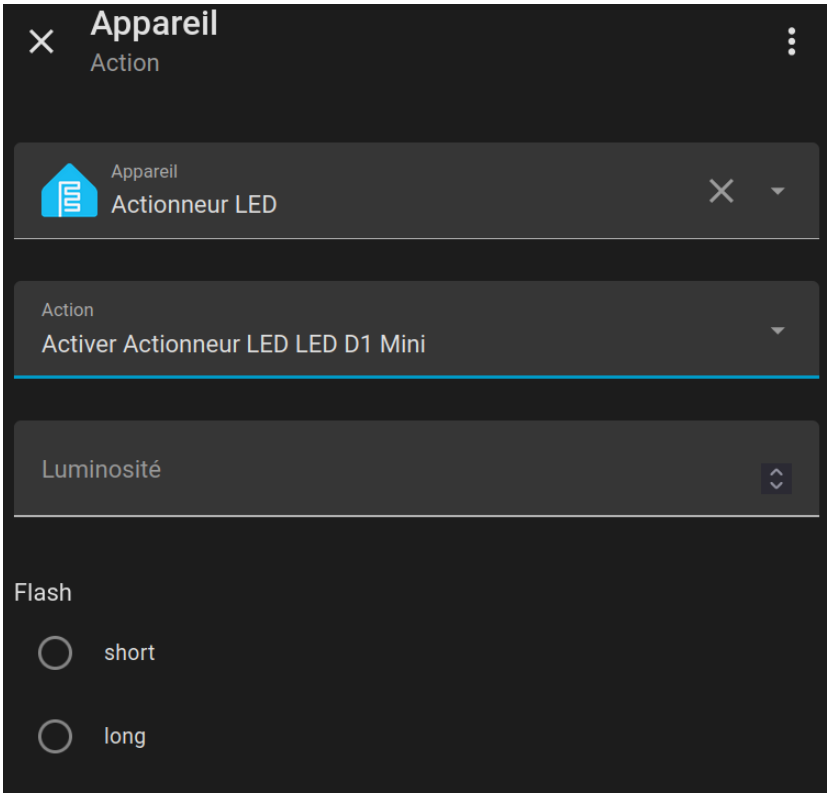
L'animateur **présente** les kits ESP du Fablab en interrogeant sur le rôle des **micro-contrôleurs** ainsi que sur les **capteurs** qui y sont connectés.
L'animateur présente le ESP home à travers plusieurs projets concrets présentables

Station météo Dashboard

TP utilisation au potager *30mn*

Les apprenants créent une scène en utilisant un capteurs et un actionneur exemple :





L'animateur fait déclencher et fait constater la nécessité d'une condition inverse (Sinon la LED s'allume et ne s’éteint plus) :

> Attention aux hystérésis 49-50-49-50-... La LED clignote sans arrêt