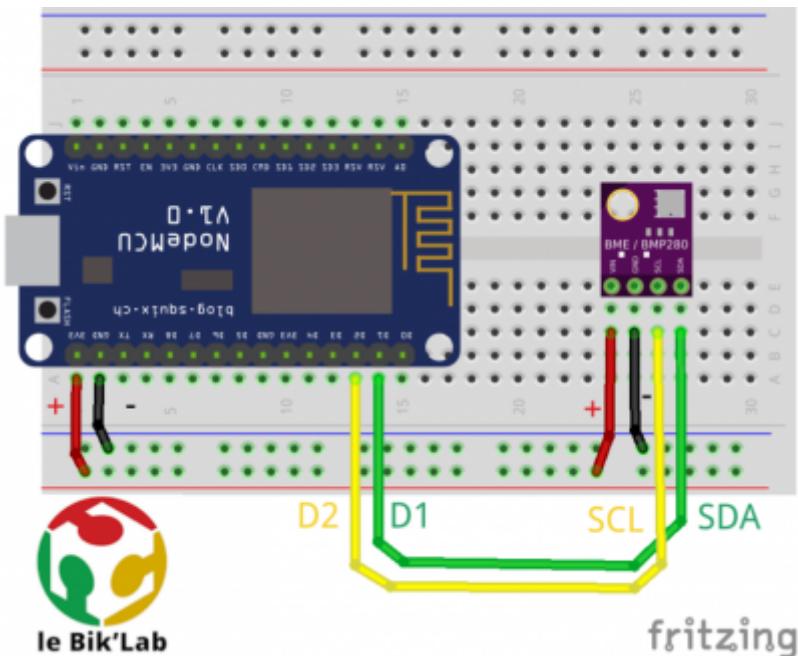


ESPhome / BME 280

Montage



Attention à la polarité ! (+/-)



Suivre le schéma de montage en respectant les conventions de couleur pour les fils.

Documentation technique

La documentation technique complète est disponible :

- en ligne : [Kit BME280](#)
- en téléchargement (PDF) :

fiche_kit_bme280.pdf

Toute cette documentation est diffusée sous [licence Creative Commons CC BY 4.0 Deed](#) pour en faciliter la réutilisation.

Code

Code basique

Ce code simple fonctionne directement avec nos kits. Si vous avez plusieurs nœuds, prenez soin de

changer "station1" dans le code pour reconnaître les nœuds.

bme280simple.yaml

```
esphome:  
    name: "station1" # le nom de l'objet connecté  
  
esp32:  
    board: nodemcu-32s # ajuster selon la plateforme, ok pour nos kits  
    framework:  
        type: arduino  
  
logger:  
  
# Enable Home Assistant API  
api:  
    encryption:  
        key: "JhwNLgVDiZLATksukQRb2//wYz/olZdI/mBx22uX9WA="  
        # voir  
        https://esphome.io/components/api.html#configuration-variables  
        # il y a sur la page un générateur de clé aléatoire  
        # changez et mettez une autre valeur pour votre noeud  
  
ota:  
    password: "secret_ota_password" # changer pour une valeur de votre  
choix  
  
wifi:  
    ssid: "wifi_ssid" # nom du réseau wifi  
    password: "wifi_password" # mot de passe du réseau wifi  
  
## on définit les GPIO du bus I2C  
i2c:  
    sda: D1 # à changer si carte différente (GPIO21 pour esp32)  
    scl: D2 # à changer si carte différente (GPIO22 pour esp32)  
    scan: True  
    id: bus_a  
  
sensor:  
    - platform: bme280 # on ajoute le capteur bme280  
    # Doc esphome: https://esphome.io/components/sensor/bme280.html  
        temperature: # définition du capteur de température du bme280  
            name: "Temperature BME280 (station1)"  
        pressure: # définition du capteur de pression atmo du bme280  
            name: "Pression atmosphérique (station1)"  
        humidity: # définition du capteur d'humidité dans l'air du bme280  
            name: "Hygrométrie (station1)"  
        address: 0x76 #sur certaines variantes du capteur, changer pour  
0x77  
        update_interval: 60s # changer ceci si vous souhaitez rafraîchir
```

```
plus
# ou moins souvent, par exemple 10s ou 5m.
```

Code avancé

Ce code reprend toutes les fonctionnalités du code précédent, mais en plus, déplace les codes wifi et autres codes secrets dans le fichiers secrets.yaml (qui doit exister, sinon ça ne fonctionne pas) et en plus ajoute des capteurs “techniques”, comme l'adresse IP, l'uptime ou encore la force du signal wifi reçu. Si vous devez changer le nom du noeud, il suffit de changer la ligne 2 et modifier “station1” par ce que vous souhaitez, les modifications seront appliquées dans le reste du code.

bme280adv.yaml

```
substitutions:
  devicename: station1 # nom de l'objet connecté, changer ici
  uniquement

esphome:
  name: ${devicename} # la valeur définie plus haut sera placée ici

esp32:
  board: nodemcu-32s # ajuster selon la plateforme, ok pour nos kits
  framework:
    type: arduino

logger:

# Enable Home Assistant API
api:
  encryption:
    key: "JhwNLgVDiZLATKsukQRb2//wYz/olZdI/mBx22uX9WA="
    # voir
    https://esphome.io/components/api.html#configuration-variables
    # il y a sur la page un générateur de clé aléatoire
    # changez et mettez une autre valeur pour votre noeud

ota:
  # le mot de passe OTA sera récupéré dans sectets.yaml
  password: !secret ota_password
  # décommenter la ligne ci après pour définir le mot de passe ici
  # password: "wifi_password" # mot de passe du réseau wifi

wifi:
  # le ssid et le password wifi seront récupérés dans sectets.yaml
  ssid: !secret wifi_ssid
  password: !secret wifi_password
  #décommenter pour définir le wifi dans ce fichier, supprimer au
  dessus
  # ssid: "wifi_ssid" # nom du réseau wifi
```

```
# password: "wifi_password" # mot de passe du réseau wifi

## on définit les GPIO du bus I2C
i2c:
  sda: D1 # à changer si carte différente (GPIO21 pour esp32)
  scl: D2 # à changer si carte différente (GPIO22 pour esp32)
  scan: True
  id: bus_a

sensor:
  - platform: bme280 # on ajoute le capteur bme280
    # Doc esphome: https://esphome.io/components/sensor/bme280.html
    temperature: # définition du capteur de température du bme280
      name: "Temperature BME280 (${devicename})"
    pressure: # définition du capteur de pression atmo du bme280
      name: "Pression atmosphérique (${devicename})"
    humidity: # définition du capteur d'humidité dans l'air du bme280
      name: "Hygrométrie (${devicename})"
    address: 0x76 #sur certaines variantes du capteur, changer pour 0x77
    update_interval: 60s

    # mesure la force du signal wifi reçu en dB
    # documentation : https://esphome.io/components/sensor/wifi_signal
    - platform: wifi_signal
      name: "signal WiFi (${devicename})"
      update_interval: 10s

    # donne l'utptime (depuis combien de temps l'objet connecté est il allumé)
    # documentation : https://esphome.io/components/sensor/uptime
    # ici on ajoute des calculs pour avoir un format lisible (j:h:m:s)
    - platform: uptime
      name: "${devicename} Uptime Sensor"
      id: ${devicename}_uptime_sensor
      update_interval: 60s
      on_raw_value:
        then:
          - text_sensor.template.publish:
              id: ${devicename}_uptime_human
              state: !lambda |-
                  int seconds =
round(id(${devicename}_uptime_sensor).raw_state);
                  int days = seconds / (24 * 3600);
                  seconds = seconds % (24 * 3600);
                  int hours = seconds / 3600;
                  seconds = seconds % 3600;
                  int minutes = seconds / 60;
                  seconds = seconds % 60;
```

```
        return (
            (days ? String(days) + "j " : "") +
            (hours ? String(hours) + "h " : "") +
            (minutes ? String(minutes) + "m " : "") +
            (String(seconds) + "s")
        ).c_str();
```

text_sensor:

on présente l'adresse IP du noeud
documentation
https://esphome.io/components/text_sensor/wifi_info.html

```
- platform: wifi_info
  ip_address:
    name: "adresse IP (${devicename})"
    id: ${devicename}_ip_address
```

```
- platform: template
# on présente l'uptime sous forme texte, voir plus haut
  name: "Uptime (${devicename})"
  id: ${devicename}_uptime_human
  icon: mdi:clock-start
```

version de espHome utilisée
https://esphome.io/components/text_sensor/version

```
- platform: version
  name: "Version d'ESPHome installée"
  id: ${devicename}_ESPHome_Version
```

From:
<https://wiki.lebiklab.fr/> - Wiki Le BIK'LAB



Permanent link:
<https://wiki.lebiklab.fr/doku.php?id=projets:home-assistant:esphome: noeud-basique-mesures-environnementales:bme280&rev=1698377963>

Last update: 04/04/2024 15:35