Sat Cuve à eau

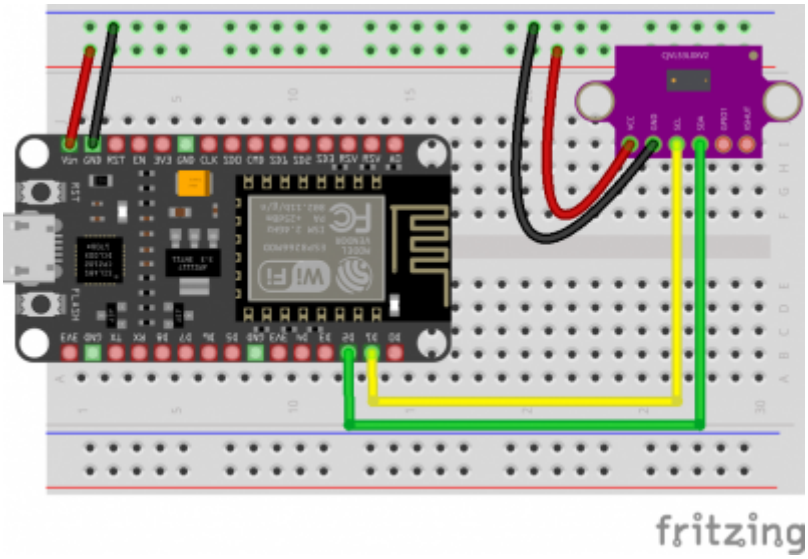
Objectif

Mesure la quantité d'eau disponible dans des citernes à eau.

Moyen : on utilise un capteur laser pour mesurer la distance entre le sommet de l'eau et le capteur de façon à extrapoler la hauteur de la colonne d'eau et en déduire le volume total disponible

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 : sat-cuve
- en téléchargement (PDF) : https://www.alldatasheet.com/datasheet-pdf/download/948120/STMICROELECTRONICS/VL53L0X.html

Code

Code basique

GYVL53L0XV2simple.yaml

```yaml
esphome:
  name: citerne
  friendly_name: Citerne

esp8266:
  board: esp01_1m

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
    key: "jKAUk7eaGd1hf4V1y9LIXGxA9nLUavgWrY2jFNmTQQ0="

ota:
  - platform: esphome
    password: "57e167ac1e278fd9f20970f9690f9ede"

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

  # Enable fallback hotspot (captive portal) in case wifi connection
fails
  ap:
    ssid: "Citerne Fallback Hotspot"
    password: "s6xKWSkMR6XG"

captive_portal:
# Activer le support i²c
i2c:
  sda: GPIO4    #D2  # à adapter selon ton câblage
  scl: GPIO5    #D1
  scan: true

# Capteur VL53L0X
sensor:
  - platform: vl53l0x
    name: "Distance Laser"
    update_interval: 60s
```

```
    long_range: false
    timeout: 200ms
```

Code avancé

GYVL53L0XV2adv.yaml

```
substitutions:
  devicename: citerne
  friendly_devicename: Citerne

esphome:
  name: ${devicename}
  friendly_name: ${friendly_devicename}

esp8266:
  board: esp01_1m

# Enable logging
logger:

# Enable Home Assistant API
api:
  encryption:
    key: "jKAUk7eaGd1hf4V1y9LIXGxA9nLUavgWrY2jFNmTQQ0="

ota:
  - platform: esphome
    password: "57e167ac1e278fd9f20970f9690f9ede"

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

  # Enable fallback hotspot (captive portal) in case wifi connection
fails
  ap:
    ssid: "${devicename} Fallback Hotspot"
    password: "s6xKWSkMR6XG"

captive_portal:
# Activer le support i²c
i2c:
  sda: GPIO4    #D2  # à adapter selon ton câblage
  scl: GPIO5    #D1
  scan: true

# Capteur VL53L0X
```

```
sensor:
  - platform: vl53l0x
    name: "Distance Laser"
    update_interval: 60s
    long_range: false
    timeout: 200ms

  - platform: wifi_signal
    name: "signal WiFi (${devicename})"
    update_interval: 60s

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

  - platform: template
    name: "Uptime (${devicename})"
    id: ${devicename}_uptime_human
    icon: mdi:clock-start

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