

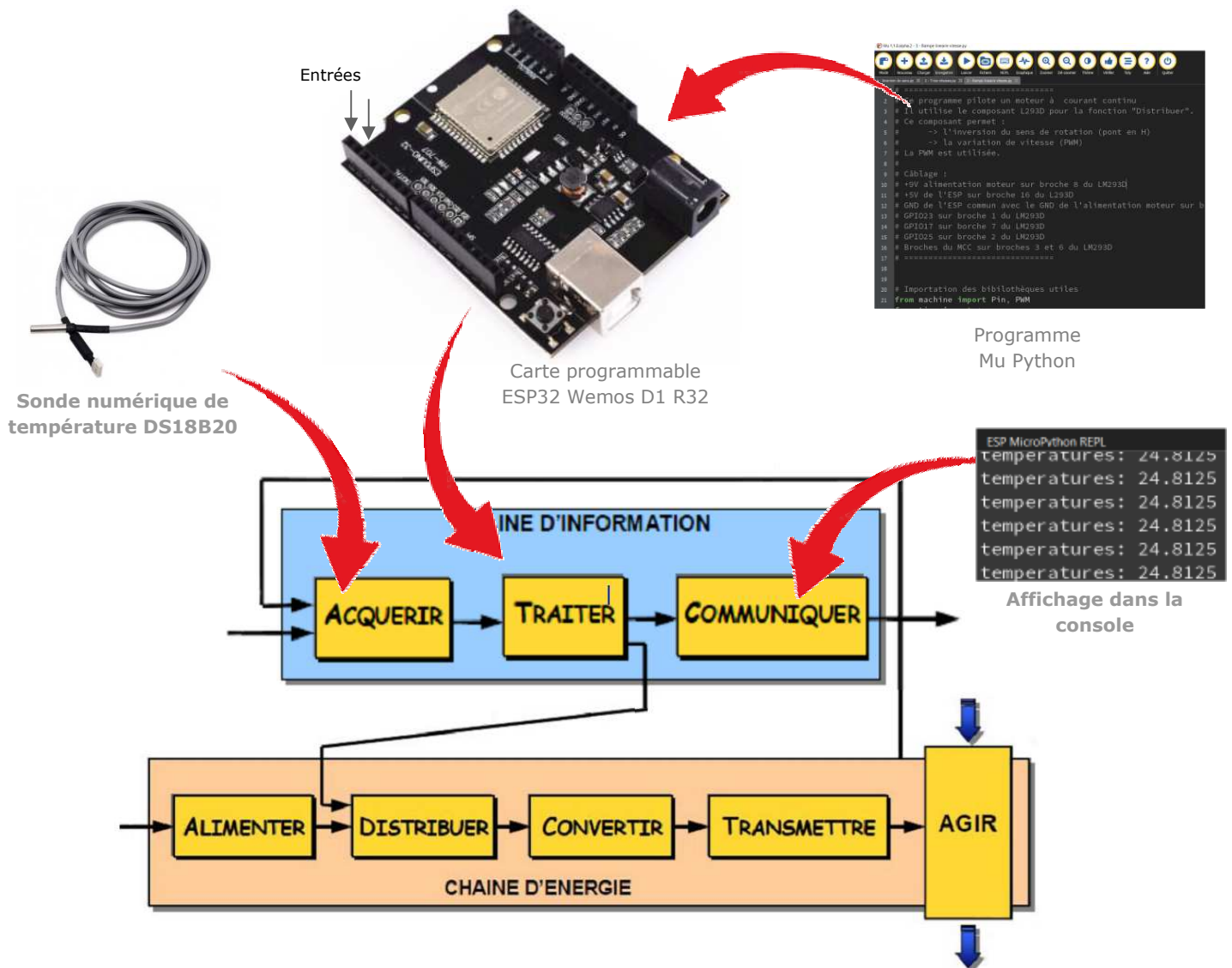


# MISE EN ŒUVRE

→ **TRAITER** : ESP32 WEMOS (EDI MU)

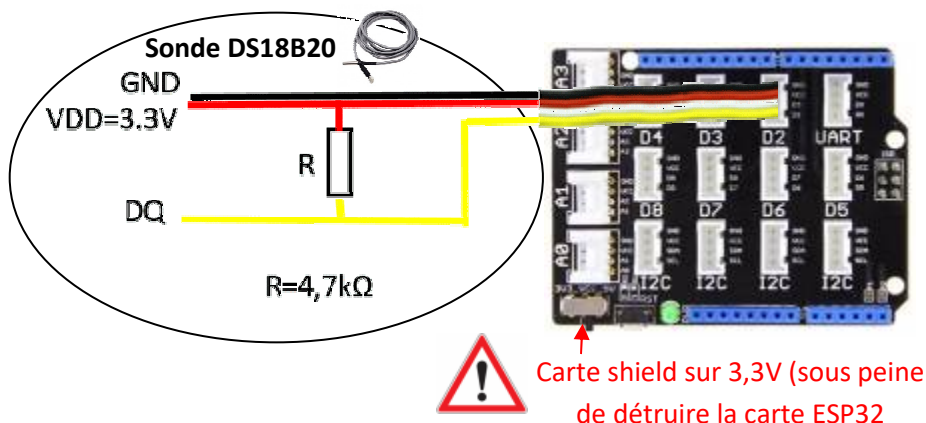
→ **ACQUERIR** : Sonde numérique de température DS18B20

## 1 – Mise en situation



## 2 – Plan de câblage / Montage

**Raccordement :** DQ sur D2 shield base 1 (broche 26)



## 3 – Explications

### Principe de fonctionnement :

Le DS18B20 est composé d'un capteur de température, d'un convertisseur analogique - numérique, d'une mémoire de 8 octets et d'une EEPROM de 3 octets.

Il communique ses données numériques grâce à un seul fil qui utilise le protocole OneWire.

Le pilote OneWire est implémenté dans MicroPython et fonctionne sur toutes les broches de l'ESP32.

Il nécessite l'utilisation d'une bibliothèque pour le contrôler.

- **Télécharger** la bibliothèque : [esp\\_18x20.zip \(fournie\)](#) et l'**installer** (voir le [tutoriel fourni avec le logiciel Mu Python](#))

### Valeurs caractéristiques : (voir le Datasheet DS18b20 fourni)

- Alimentation : 3 à 5,5VDC
- Plage de température mesurable : -55 à +125 °C
- Précision : +0.0625°C en 12 bits, +0.125°C en 11 bits, +0.25°C en 10 bits et +0.5°C en 9 bits
- Temps de conversion : minimum (9 bits) : 93.75 ms, maximum (12 bits) : 750 ms
- Communication via bus OneWire

## 4 – Programme

### ESP32 Micropython programme qui affiche la température

```
import time, machine, onewire, esp_ds18x20

# DQ = broche 26 (D2 shield base 1)
data = machine.Pin(26)

# creation objet onewire
ds = esp_ds18x20.DS18X20(owewire.OneWire(data))

# Scan le bus OneWire et recupere l'ID de chaque sonde
adresses_cpt = ds.scan()
print('Adresses capteurs trouvées:', adresses_cpt)

# Lecture et affichage température de chaque sonde
while True:
    print('temperatures:', end=' ')
    ds.convert_temp()
    for adresse in adresses_cpt:
        print(ds.read_temp(adresse), end=' ')
    print()
    time.sleep_ms(1000)
```