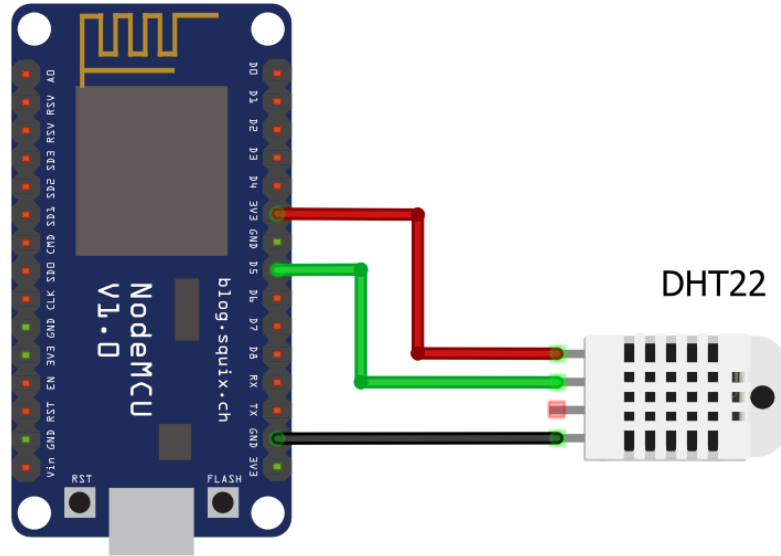


Station météo

Sonde



ESP8266 : wifi
ESP32 : wifi + Bluetooth



Capteur d'Humidité et de Température - DHT22

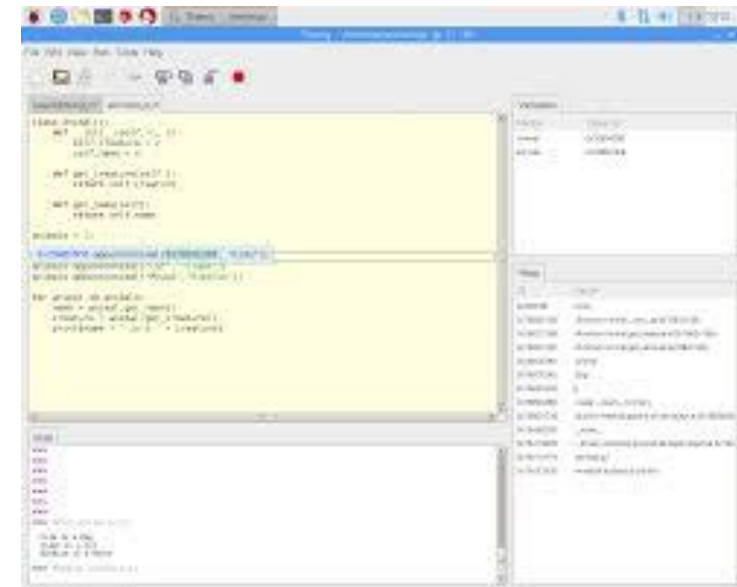
- Économique grâce à une interface numérique à câble unique
- Calibré et ne nécessitant pas de composants complémentaires
- Entrée : 3,3 à 6 V
- Courant de mesure : 1 à 1,5 mA
- Courant de veille : 40 à 50 μ A
- Humidité : 0 à 100 % d'humidité relative
- Température : -40 ° à 80 °C (+0,5 °C)
- Précision de l'humidité relative : +-2%
- Conforme RoHS

```
Blink

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

Arduino



Python

Architectures matérielles et systèmes d'exploitation

Contenus	Capacités attendues	Commentaires
Périphériques d'entrée et de sortie Interface Homme-Machine (IHM)	Identifier le rôle des capteurs et actionneurs. Réaliser par programmation une IHM répondant à un cahier des charges donné.	Les activités peuvent être développées sur des objets connectés, des systèmes embarqués ou robots.

Représentation des données : types et valeurs de base

Représentation d'un texte en machine. Exemples des encodages ASCII, ISO-8859-1, Unicode	Identifier l'intérêt des différents systèmes d'encodage. Convertir un fichier texte dans différents formats d'encodage.	Aucune connaissance précise des normes d'encodage n'est exigible.
--	--	---

Traitement de données en tables

Indexation de tables	Importer une table depuis un fichier texte tabulé ou un fichier CSV.	Est utilisé un tableau doublement indexé ou un tableau de p-uplets qui partagent les mêmes descripteurs.
----------------------	--	--

Th Thonny

Thonny - C:\Users\Philippe\Desktop\NodeMcu\workspace\ESP8266\dht22.py @ 19:1

File Edit View Run Device Tools Help

```
1 #hardware platform: ESP8266
2 #Capeur dht22 (boitier blanc), mesure temperature et taux d'humidite
3 #DHT 22 sur D14
4
5 from machine import Pin
6 import dht
7 import time
8
9 dht22 = dht.DHT22(Pin(14))
10
11 while True:
12     dht22.measure()
13     print(str(dht22.temperature())+"",str(dht22.humidity())+"")
14
15 print("Humidite\t:",dht22.humidity()," %")
16 _print("Temperature\t:",dht22.temperature()," °C")
17
18 |
19
20 time.sleep(1)
21
22
23
24
25
26
```

Shell

```
temperature : 23.0 °C
rh : 70.3 %
b'23.0,70.3\r\n'
temperature : 23.0 °C
rh : 70.3 %
b'23.0,70.3\r\n'
temperature : 23.0 °C
rh : 70.3 %
b'23.0,70.3\r\n'
temperature : 23.0 °C
rh : 70.3 %
b'23.0,70.3\r\n'
temperature : 23.0 °C
rh : 70.3 %
b'23.0,70.3\r\n'
```

Thonny options

Assistant Debugger Editor General Interpreter Terminal Theme & Font

What should Thonny use for running your code?

The same interpreter which runs Thonny (default)

Details

C:\Users\Philippe\AppData\Local\Programs\Thonny\python.exe

OK Cancel

Program arguments:

Name	Value
ACTIVE	'active'
ALL	'all'
ANCHOR	'anchor'

application has been destroyed



IHM → PC



Acquisition données → ESP8266

Thonny - C:\Users\Philippe\Desktop\NodeMcu\workspace\ESP8266\dht22.py @ 15:1

File Edit View Run Device Tools Help

ihm.py x dht22.py x

```
1 #hardware platform: ESP8266
2 #Capteur dht22 (boitier blanc), mesure temperature et taux d'humidite
3 #DHT 22 sur D14
4
5 from machine import Pin
6 import dht
7 import time
8
9 dht22 = dht.DHT22(Pin(14))
10
11 while True:
12     dht22.measure()
13     print(str(dht22.temperature())+", "+str(dht22.humidity()))
14     time.sleep(1)
15
16 '''
17 print("Humidite\t:",dht22.humidity()," %")
18     print("Temperature\t:",dht22.temperature()," C")
19 '''
20
21
22
23
24
25
```

Thonny options

Assistant Debugger Editor General Interpreter Terminal Theme & Font

What should Thonny use for running your code?

MicroPython on a generic device

Details

Port (look for your device name, "USB Serial" or "UART")

USB-SERIAL CH340 (COM15)

OK Cancel

Variable:

Name

bdev

gc

Shell x

```
26.8, 65.4
26.8, 65.4
26.8, 65.4
26.8, 65.4
```

Sonde



Trame au format csv

`print(str(dht22.temperature())+", "+str(dht22.humidity()))`

- Données séparées par une virgule
- Carriage return « CR » + line feed « LF »



```

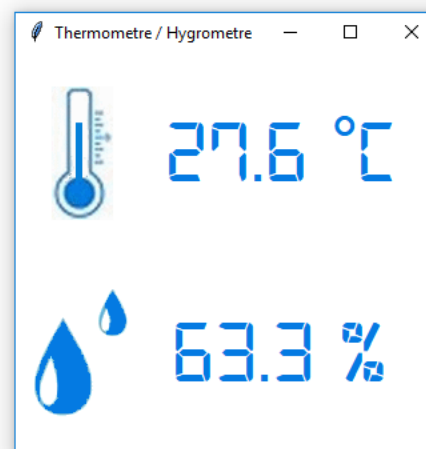
1  # -*- coding:utf-8 -*-
2  # Python 3.7.2 - Thermometre-Hygrometre
3
4  from tkinter import *
5  import time
6  import serial
7
8  Port = serial.Serial("COM15", 115200)
9  print(Port.name)
10 trame=[] #Trame envoyee par ESP
11
12 def on_closing():
13     print("exit")
14     fenetre.destroy()
15 #     Port.close()
16     print("Port ferme")
17
18
19 fenetre = Tk()
20
21 fenetre.title("Thermometre / Hygrometre") #Titre de l'application
22 fenetre.geometry("320x300+100+100") #Largeur x hauteur + position x + position y
23 fenetre.configure(bg='white')
24 fenetre.protocol("WM_DELETE_WINDOW", on_closing)
25
26 C1 = Canvas(fenetre,width=100,height=150,bg='white',highlightthickness="0")
27 Img0 = PhotoImage(file="thermometre.gif")
28 C1.create_image(50,75, image=Img0)
29 C1.grid(row=0,column=0)
30
31 C2 = Canvas(fenetre,width=100,height=150,bg='white',highlightthickness="0")

```

```

Shell
temperature : 27.6 °C
rh : 63.3 %
b'27.6,63.3\r\n'
temperature : 27.6 °C
rh : 63.3 %
b'27.6,63.3\r\n'
temperature : 27.6 °C
rh : 63.3 %
b'27.6,63.3\r\n'
temperature : 27.6 °C
rh : 63.3 %
b'27.6,63.3\r\n'
temperature : 27.6 °C
rh : 63.3 %

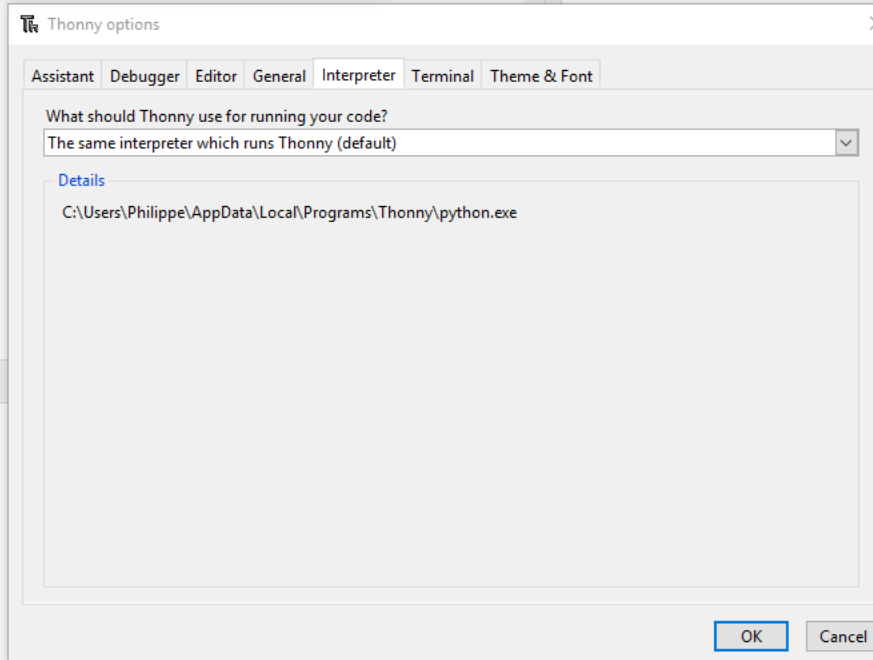
```

Program arguments:

Variables ×

Name	Value

Assistant ×

IHM

▶ Run

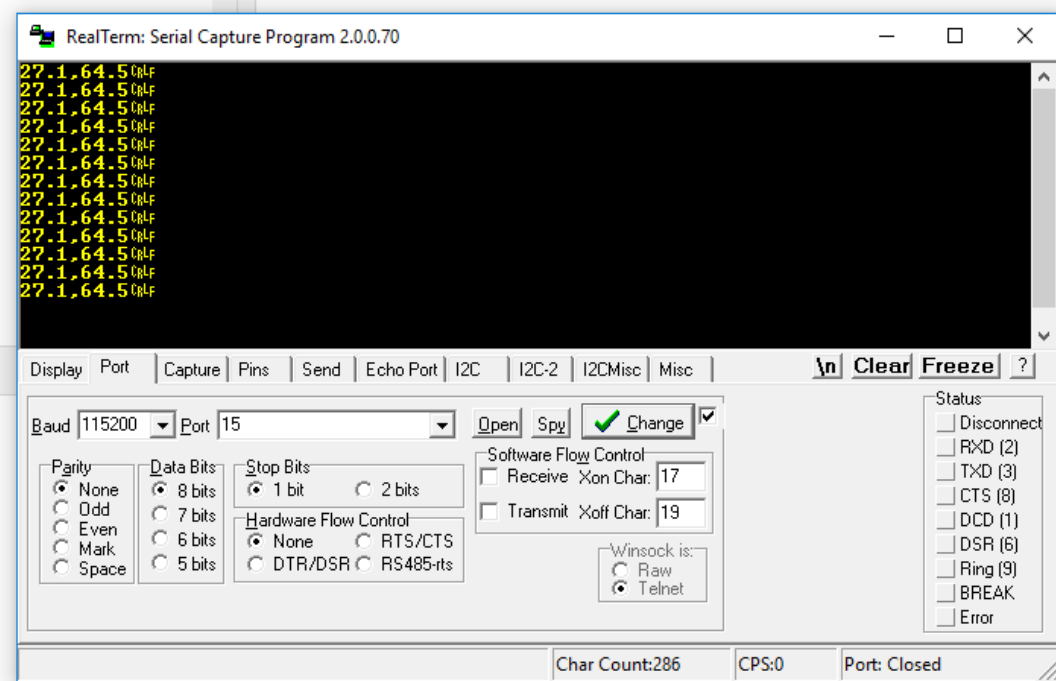
Traitement de la trame

b'27.7,63.0\n → Liste
Codage ASCII
1 octet par caractère

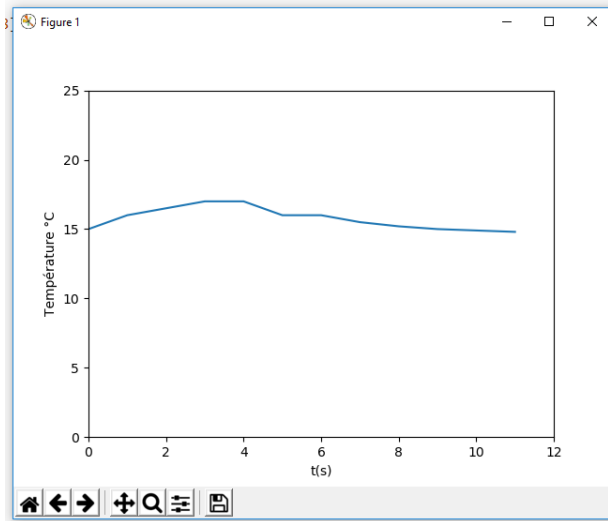
Encodage format csv

Température + séparateur « , » + humidité + fin ligne « \n »

Capture de la trame dans un terminal



Evolutions du projet

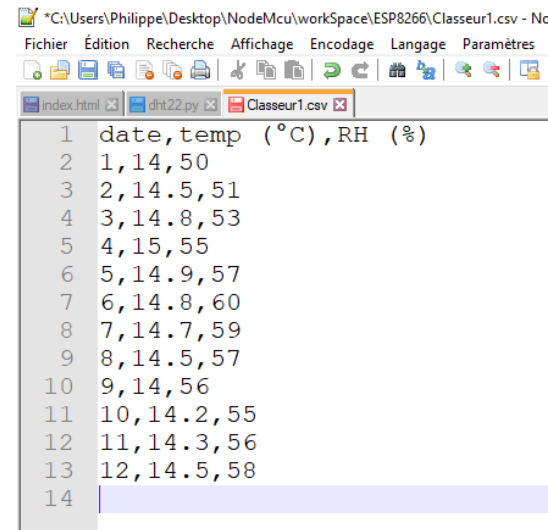


Affichage des données en temps réel

Affichage température en °C / °F
pour les 12 dernières heures

Valeurs min. et max.

Calendrier, horloge et alarme...



1	date	temp (°C)	RH (%)
2	1	14	50
3	2	14.5	51
4	3	14.8	53
5	4	15	55
6	5	14.9	57
7	6	14.8	60
8	7	14.7	59
9	8	14.5	57
10	9	14	56
11	10	14.2	55
12	11	14.3	56
13	12	14.5	58
14			

Enregistrement des données
au format csv

Radio 433 Mhz



Communication