

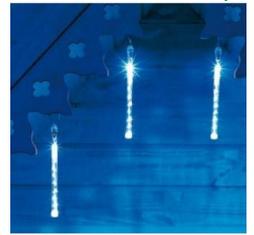
Programmation



La programmation Python avec ChatGPT

Remue-méninges 08-1

Quelle contribution peut-on attendre d'outils tels que ChatGPT ?



Bandeau WS2812B de 20 LED

Section de Technicien
Supérieur

option

**Electronique
&
Communication**



A. ChatGPT : Quèsaco¹ ?

Question posée à ChatGPT : *C'est quoi ChatGPT et à quoi cela sert ?*

Réponse de ChatGPT :

"ChatGPT est un modèle de langage développé par OpenAI qui utilise une architecture de traitement du langage naturel appelée GPT (*Generative Pre-trained Transformer*) pour générer des réponses à des questions ou pour interagir avec les utilisateurs par le biais de conversations en langage naturel.

Son objectif principal est d'offrir aux utilisateurs une expérience conversationnelle fluide et naturelle. L'outil est entraîné sur de vastes ensembles de données de texte afin d'apprendre à comprendre et à générer des réponses précises et pertinentes aux questions qui sont posées.

ChatGPT est souvent utilisé comme assistant personnel virtuel dans de nombreux domaines tels que la recherche d'informations, l'assistance à la clientèle, l'éducation et le divertissement. En résumé, C'est un outil de traitement de langage naturel sophistiqué qui peut aider les utilisateurs à communiquer et à interagir avec les ordinateurs de manière plus naturelle et efficace."

B. ChatGPT et l'enseignement des Sciences Numériques en STS

Un des écueils des activités en sciences numériques niveau Section de Technicien Supérieur est la nécessité d'utiliser des microcontrôleurs dans les systèmes électroniques. Ces composants nécessitent une programmation dont la complexité dépend des interactions du composant avec son environnement. Les difficultés peuvent être liées aux spécificités des capteurs souvent programmables, à la quantité de périphériques, à la nécessité de rendre rapide et fluide les réactions du programme...

Question posée à ChatGPT : *Que peut apporter ChatGPT à l'enseignement des Sciences Numériques pour une classe de Section de Techniciens Supérieurs ?*

Réponse de ChatGPT :

"ChatGPT peut être un outil utile pour l'enseignement des Sciences Numériques pour une classe de techniciens supérieurs en fournissant des réponses rapides et précises aux questions, des exemples pratiques, en encourageant l'auto-apprentissage, en économisant du temps et en offrant une expérience d'apprentissage interactive."

C. Cas d'utilisation : PROGRAMMER en MicroPython sur ESP32

1. Cas d'utilisation n°1 : Contrôle du fonctionnement de la carte Lolin Lite32

Ce premier exemple doit permettre de vérifier le bon fonctionnement de la carte *LoLin lite32*. Cette carte de développement dispose d'une LED connectée sur la broche n°22.



Question posée à ChatGPT : _____

¹ Locution provençale signifiant qu'est cela ?

Réponse de ChatGPT :

"Oui, je peux vous aider à écrire un programme en MicroPython pour un microcontrôleur ESP32 qui fera clignoter une LED connectée sur la broche 22. Voici un exemple de code qui fera clignoter la LED toutes les 500 millisecondes :

```
css Copy code  
  
import machine  
import time  
  
led = machine.Pin(22, machine.Pin.OUT)  
  
while True:  
    led.value(not led.value())  
    time.sleep_ms(500)
```

Ce code commence par importer les modules nécessaires *machine* et *time*.

Ensuite, la broche 22 est configurée en mode sortie avec la commande *machine.Pin(22, machine.Pin.OUT)*.

Enfin, une boucle infinie est utilisée pour faire clignoter la LED toutes les 500 millisecondes en inversant la valeur de la broche avec la commande *led.value(not led.value())* et en attendant 500 millisecondes avec *time.sleep_ms(500)*."

Conclusion :



La proposition de ChatGPT ne diffère pas sur le principe du programme original imaginé SANS ChatGPT :

```
from machine import Pin  
from time import sleep  
led = Pin(22, Pin.OUT)  
while True:  
    led.value(not led.value())  
    sleep(0.5)
```

2. Cas d'utilisation n°2 : Projet n°1 - Stalactites lumineuses

Le programme MicroPython permettant un effet visuel de type "stalactite lumineuse" sur un ruban de 20 LEDs a été réalisé. On en rappelle son contenu :

```

#importation des librairies
import machine, neopixel, time, random
#Nombre de LED sur le ruban
n = 20
#Numéro de la broche
p = 5
#Nombre de LED dans la stalactite
nb = 6
#Création d'un objet néopixel sur la broche p
np = neopixel.NeoPixel(machine.Pin(p),n)
#Effet stalactite sur nb LED
while True : #boucle infinie
    r = random.randint(0,255) #Nbre aléatoire composante rouge
    g = random.randint(0,255) #Nbr aléatoire composant verte
    b = random.randint(0,255) #Nbr aléatoire composante bleue
    for i in range(nb):
        for lum in range(nb): #nb LED (de 0 à nb-1) de lumineusité décroissante
            np[i-lum] = (round(r // 2 ** lum),round(g // 2 ** lum),round(b // 2 ** lum))
        np[i-nb] = (0, 0, 0) #La dernière LED est éteinte

    np.write() #Envoi sur le ruban
    time.sleep_ms(100) #Déterminer la vitesse de descente de la stalactite

```



Question [1] posée à ChatGPT : _____

Réponse de ChatGPT :

"Voici un exemple de programme en MicroPython pour contrôler un bandeau de 20 LED sur la broche 5 d'un ESP32. Ce programme utilise la bibliothèque NeoPixel pour contrôler les LED.

```

python
import machine
import neopixel
import time

# Définition des paramètres de la LED
NUM_LEDS = 20
PIN = 5

# Initialisation de la LED
led = neopixel.NeoPixel(machine.Pin(PIN), NUM_LEDS)

# Fonction pour dessiner une stalactite lumineuse
def draw_stalactite(color):
    for i in range(NUM_LEDS):
        led[i] = color
        led.write()
        time.sleep(0.1)
    for i in range(NUM_LEDS-1, -1, -1):
        led[i] = (0, 0, 0)
        led.write()
        time.sleep(0.1)

# Boucle principale pour dessiner la stalactite en boucle
while True:
    draw_stalactite((255, 0, 0)) # Rouge
    draw_stalactite((0, 255, 0)) # Vert
    draw_stalactite((0, 0, 255)) # Bleu

```

20 LED et broche n°5

Création d'un objet "NeoPixel"

Eclairage une à une des LED

Extinction une à une des LED

Succession de 3 couleurs

Première étape: _____



_____. Le programme est malgré tout fonctionnel.

La proposition de ChatGPT doit donc être amendée en reformulant ou en précisant la demande.

Des demandes formulées par l'utilisateur, l'intelligence artificielle va modifier ses éléments de réponse. Le rôle du prescripteur est de formuler le plus explicitement possible ses souhaits : c'est la difficulté !

Plusieurs étapes peuvent être nécessaires.



Question [2] posée à ChatGPT : _____
_____. *Peux tu modifier le programme précédent ?*

Réponse de ChatGPT :

"Bien sûr, voici une version modifiée du programme qui utilise l'effet stalactite pour éclairer une série de 6 LED avec une luminosité décroissante et parcourir le bandeau de 20 LED :"



Question [3] posée à ChatGPT : _____
_____.
_____.

Réponse de ChatGPT :

"Je suis désolé pour la confusion, voici une version modifiée du programme qui utilise l'effet stalactite tel que décrit :"



Question [4] posée à ChatGPT : _____

_____.

Réponse de ChatGPT :

"D'accord, voici une version modifiée du programme qui utilise l'effet stalactite tel que décrit, avec une luminosité décroissante linéairement de 100% à 10% sur les 6 LED allumées :"



Question [5] posée à ChatGPT : _____



Question [6] posée à ChatGPT : _____
_____.

```
python Copy code

import machine
import neopixel
import time
import urandom

# Définition des paramètres de la LED
NUM_LEDS = 20
PIN = 5

# Initialisation de la LED
led = neopixel.NeoPixel(machine.Pin(PIN), NUM_LEDS)

# Fonction pour dessiner l'effet stalactite avec une couleur aléatoire
def draw_stalactite():
    color = (urandom.getrandbits(8), urandom.getrandbits(8), urandom.getrandbits(8))
    for i in range(NUM_LEDS - 5):
        for j in range(6):
            brightness = int(10 + (j * (90 / 5)))
            led[i + j] = tuple(map(lambda x: int(x * brightness / 100), color))
        for k in range(NUM_LEDS):
            if k < i or k > i + 5:
                led[k] = (0, 0, 0)
        led.write()
        time.sleep(0.1)

# Boucle principale pour dessiner la stalactite en boucle avec des couleurs aléatoires
while True:
    draw_stalactite()
```



Question [7] posée à ChatGPT : _____

Réponse de ChatGPT :

"Bien sûr, voici un résumé des différentes étapes pour réaliser cet effet stalactite lumineux sur un bandeau de 20 LED avec une luminosité décroissante :

1. Initialiser les paramètres de la LED : le nombre de LED et le numéro de broche.
2. Initialiser la LED avec la bibliothèque "*neopixel*".
3. Écrire une fonction pour dessiner l'effet stalactite avec une couleur aléatoire : pour chaque position de départ, allumer une série de 6 LED avec une luminosité décroissante jusqu'à la dernière LED, puis éteindre les autres LED. La couleur de chaque LED est déterminée par une couleur aléatoire générée à chaque itération.

4. Dans la boucle principale, appeler la fonction de dessin de l'effet stalactite avec une couleur aléatoire en boucle.



Conclusion : La contribution de ChatGPT, _____, permet de répondre parfaitement à la demande de l'utilisateur. Un programme fonctionnel est proposé suivi d'explications. Deux possibilités à ce niveau :

1. Le prescripteur n'a pour seul objectif que d'obtenir un résultat. Il sait juger si le résultat correspond à son besoin et est capable de stopper les itérations de ChatGPT
2. Le prescripteur est aussi un apprenant et à partir d'un résultat qu'il qualifie de satisfaisant, il va pouvoir approfondir ses connaissances en programmation MicroPython par une analyse complémentaire du code.

C'est dans ce contexte que l'on doit se placer si on veut que ChatGPT nous apporte du savoir-faire.

3. Cas d'utilisation n°3 : Projet n°1 - Stalactites lumineuses AVEC ÉVOLUTION ALÉATOIRE

L'organisation des bandes lumineuses a été définie pour permettre un chaînage des LEDs. Il est essentiel de faire comprendre à ChatGPT ce principe de chaînage.

