



Raspberry Pi Pico Essentials

Extrait : Wi-Fi avec le Raspberry Pi Pico

Dogan Ibrahim (Royaume-Uni)

Cet article reproduit presque la moitié d'un chapitre du livre en anglais, *Raspberry Pi Pico Essentials*, de Dogan Ibrahim, publié par Elektor en mars 2021 et devenu un best-seller. Il s'agissait alors du premier ouvrage indépendant présentant des aspects concrets et éprouvés du microcontrôleur RP2040 embarqué sur le module Raspberry Pi Pico, ainsi que les logiciels correspondants. Le Pico est-il aussi accessible qu'il le prétend et est-il à la hauteur de ce que l'on attend d'une carte à contrôleur « prête à l'emploi » ? C'est ce que nous allons découvrir dans cet article.

Note de l'éditeur : cet article est un extrait du livre *Raspberry Pi Pico Essentials* formaté et légèrement modifié pour correspondre aux normes éditoriales et à la mise en page du magazine *Elektor*. Puisque cet article est extrait d'une publication plus vaste, certains termes peuvent faire référence à des passages du livre d'origine situés ailleurs. L'auteur et l'éditeur ont fait de leur mieux pour l'éviter et seront heureux de répondre aux questions – Pour les contacter, voir l'encadré « Des questions, des commentaires ? ».

Dans cet article (à l'origine la moitié du chapitre 10 du livre, *Ndlr*), nous réaliserons un projet qui utilise une liaison Wi-Fi pour établir la communication entre le RPi Pico et un smartphone.

Commande d'une LED à partir d'un smartphone en utilisant le Wi-Fi

Description : dans ce projet, nous piloterons une LED connectée au RPi Pico à partir d'un téléphone portable via la liaison Wi-Fi (pour commander un appareil, la LED peut être remplacée par ex. par un relais). Les commandes doivent être terminées par un retour chariot (CR/LF ou « nouvelle ligne »). Les commandes valides sont les suivantes :

LON Allumer la LED
LOFF Éteindre la LED

Objectif : présenter l'utilisation de la connexion Wi-Fi sur le RPi Pico.

Connexion Wi-Fi du Pico : le RPi Pico n'a pas de module Wi-Fi intégré. Il ne peut donc pas être connecté à un réseau Wi-Fi sans être relié à un module Wi-Fi externe. Le moyen le plus simple et le plus économique de le doter d'une fonction Wi-Fi est d'utiliser une carte processeur ESP-01. Il s'agit d'une carte minuscule (voir **fig. 1**), mesurant seulement 2,7 cm × 1,2 cm, animée par un processeur ESP8266 et coûtant environ 4 à 5 €. Voici les caractéristiques prometteuses de l'ESP-01 :

- Tension de fonctionnement : +3,3 V
- Interface : utilisation de commandes AT simples sur le port série/UART
- Pile de protocoles TCP/IP intégrée
- 802.11 b/g/n
- Aucun composant externe requis

L'ESP-01 communique avec le processeur hôte par l'intermédiaire de ses broches de port série TX et RX. Il s'agit d'une carte avec les huit broches suivantes :

- | | |
|---------|---|
| VCC : | broche d'alimentation +3,3 V |
| GND : | masse de l'alimentation électrique |
| GPIO0 : | broche d'entrée/sortie. Cette broche doit être connectée à +3,3 V pour un fonctionnement normal, et à GND pour télécharger le micrologiciel sur la puce |
| GPIO2 : | broche d'E/S à usage général |
| RST : | broche de réinitialisation. Doit être connectée à +3,3 V pour un fonctionnement normal. |

CH_PD : broche d'activation. Doit être connectée à +3,3 V pour un fonctionnement normal.

TX : broche de sortie série
RX : broche d'entrée série

Les broches de l'ESP-01 ne sont pas compatibles avec une platine d'essai standard, un adaptateur est donc nécessaire pour installer la carte sur une platine d'essai (fig. 2).

Schéma fonctionnel : la figure 3 montre le schéma fonctionnel du projet.

Schéma de circuit : la figure 4 montre le schéma du circuit du projet. On utilise les broches UART TX et RX du RPi Pico pour communiquer avec l'ESP-01.

Listage des programmes : le listage 1 montre le programme (Picowifi). Il est inclus dans la compilation de fichiers située dans la section Téléchargements de la page web Elektor du livre [1]. Au début du programme, la vitesse de transmission série est réglée sur 115200, c'est le débit en bauds par défaut pour l'ESP-01, et la LED est configurée en sortie et est éteinte. La fonction `ConnectToWiFi` est appelée pour se connecter au routeur Wi-Fi local. Des commandes de type AT sont utilisées pour configurer l'ESP-01 afin qu'il se connecte au routeur Wi-Fi.

Le reste du programme s'exécute dans une boucle sans fin formée à l'aide d'une instruction `while`. À l'intérieur de cette boucle, on reçoit des données du smartphone et on commande la LED en conséquence. Les commandes `LONG` et `LOFF` allument et éteignent la LED, respectivement. Les paquets de données sont reçus du smartphone à l'aide de la fonction `readline`. La fonction `find` recherche une sous-chaîne dans une chaîne de caractères et renvoie une valeur non nulle si la sous-chaîne est trouvée. On utilise la fonction `find` car les données reçues de l'appareil mobile ont le format suivant : `+ID0, n: données` (par ex. `+ID0, 3: LONG`) où `0` est l'ID de la liaison et `n` est le nombre de caractères reçus. En utilisant la fonction `find`, nous pouvons facilement rechercher les chaînes de caractères `LONG` ou `LOFF` dans le paquet de données reçu.

La fonction `ConnectToWiFi` envoie les commandes suivantes à l'ESP-01 pour se connecter au réseau Wi-Fi :

AT+RST réinitialise l'ESP-01
AT+CWMODE définit le mode de l'ESP-01 (ici, il est réglé sur le mode Station).
AT+CWJAP définit le SSID et le mot

de passe du Wi-Fi.
AT+CPIMUX définit le mode de connexion (ici, il s'agit de connexions multiples).

AT+CIFSR renvoie l'adresse IP (inutilisé ici).

AT+CIPSTART définit le mode de connexion TCP ou UDP, l'adresse IP de destination et le numéro de port (on utilise ici UDP avec le numéro de port 5000). L'adresse IP de destination est réglée sur « 0.0.0.0 » afin que n'importe quel appareil puisse envoyer des données du moment que le port 5000 est utilisé (vous pouvez remplacer cette valeur par l'adresse IP de votre smartphone pour ne recevoir des données que de votre téléphone).

Notez que chaque commande est suivie d'un petit délai. La commande `AT+CWJAP` nécessite un délai plus long. On peut facilement modifier le programme pour supprimer les délais en vérifiant les

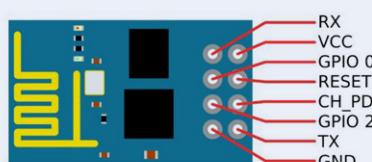
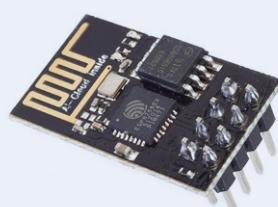


Figure 2 : Adaptateur ESP-01 pour platine d'essai.

Figure 1 : Carte à processeur ESP-01.

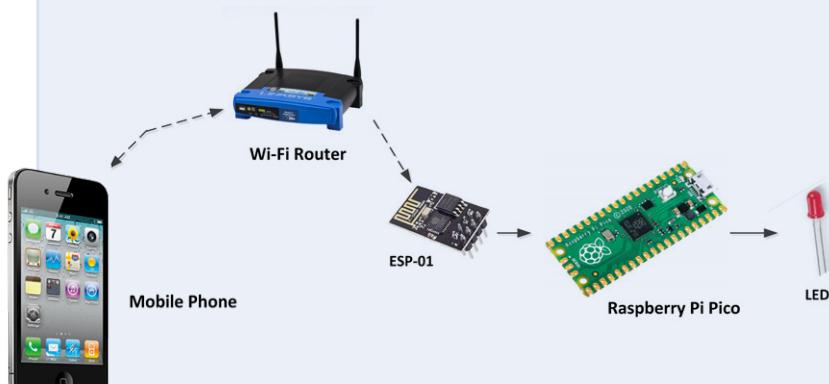


Figure 3 : Schéma fonctionnel du projet.

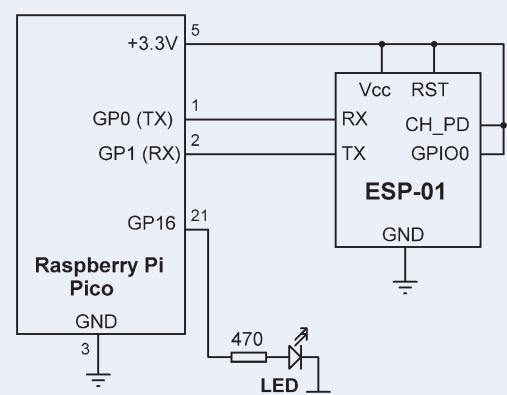


Figure 4 : Schéma du circuit du projet.



Listage 1. Programme Python pour commander un ESP-01 comme module Wi-fi.

```
# -----
#           UTILISER LE WI-FI
#           =====
#
# Dans ce projet, une puce ESP-01 est connectée
# au Raspberry Pi Pico. Cette puce est utilisée
# pour connecter le Pico au réseau Wi-Fi.

# Auteur : Dogan Ibrahim
# Fichier : Picowifi.py
# Date : février 2021
# -----
from machine import Pin, UART
import utime
uart = UART(0, baudrate=115200,rx=Pin(1),tx=Pin(0))

LED = Pin(16, Pin.OUT)
LED.value(0)

#
# Envoi des commandes AT pour la connexion de l'ESP01 au Wi-Fi local
#
def ConnectToWiFi():
    uart.write("AT+RST\r\n")
    utime.sleep(5)

    uart.write("AT+CWMODE=1\r\n")
    utime.sleep(1)

    uart.write(''AT+CWJAP="BTHomeSpot-XNH","49345xyzpq"\r\n''')
    utime.sleep(5)

    uart.write("AT+CPIMUX=0\r\n")
    utime.sleep(3)

    uart.write(''AT+CIPSTART="UDP","0.0.0.0",5000,5000,2\r\n''')
    utime.sleep(3)

ConnectToWiFi()

#
# Boucle principale
#
while True:
    buf = uart.readline()          # Lecture des données
    dat = buf.decode('UTF-8')       # Décodage
    n = dat.find("LON")            # Comporte LON?
    if n > 0:
        LED.value(1)              # LED ON
    n = dat.find("LOFF")           # Comporte LOFF?
    if n > 0:
        LED.value(0)              # LED OFF
```

réponses de l'ESP-01. De cette façon, dès que la réponse correcte est reçue, le programme peut continuer. Il se peut que vous deviez réinitialiser matériellement l'ESP-01 en l'éteignant et en le rallumant avant de lancer le programme.

Test du programme

L'utilitaire **PacketSender** (**fig. 5**) sur PC ou smartphone (après installation d'une appli UDP) permet de tester sans peine le programme.

Vous devez installer une appli de serveur UDP sur votre téléphone mobile Android avant de commencer le test avec le smartphone. Il existe de nombreuses applis UDP disponibles gratuitement dans le Play Store. Pour ce projet, nous avons installé et utilisé **UDP/TCP Widget** de K.J.M, comme le montre la **figure 6**.

Les étapes pour tester le programme sont les suivantes :

- Construisez le circuit.
- Téléchargez le programme sur votre RPi Pico.
- Lancez l'application **UDP/TCP Widget** sur votre téléphone portable.
- Cliquez sur le symbole de l'engrenage et réglez le protocole sur **UDP**, l'adresse IP sur l'adresse IP de votre RPi Pico (192.168.1.160 sur le Pico de l'auteur) et le port sur 5000, comme illustré à la **figure 7**.
- Cliquez sur l'élément de menu **MESSAGE** et sélectionnez **Texte (UTF-8)** comme Format, et entrez la commande **LON** pour allumer la LED. Sélectionnez **LF\n** comme Terminator et cliquez sur le symbole **OK** (symbole de vérification), comme le montre la **figure 8**.
- Maintenant, cliquez sur le bouton **SEND** (**fig. 9**) pour envoyer la commande au RPi Pico. Vous devriez voir le message « **Packet Sent** » s'afficher temporairement en haut de votre écran Android.

On peut obtenir l'adresse IP de l'ESP-01 en scannant tous les appareils connectés au routeur Wi-Fi local. Vous pouvez vous servir par ex. de l'appli Android **Who Uses My WiFi - Network Scanner** de Phuongpn pour voir les adresses IP de tous les appareils connectés à votre routeur. L'ESP-01 est listé comme indiqué dans la **figure 10** (IP : 192.168.1.160), avec le nom **Espressif**.

◀ (210198-04)

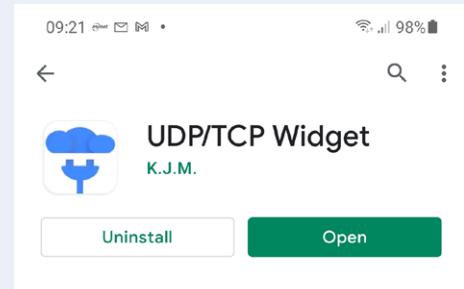
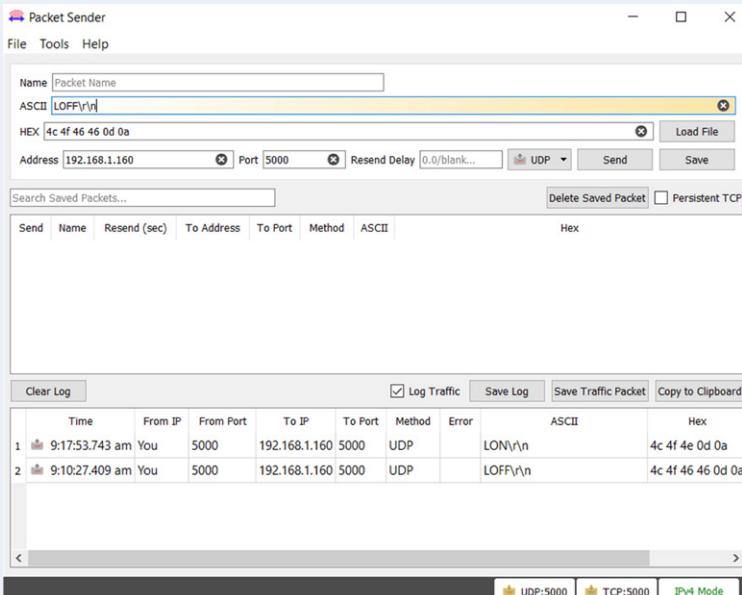


Figure 5 : Utilisation de PacketSender pour tester le programme.

Figure 6 : Appli UDP/TCP Widget pour Android.

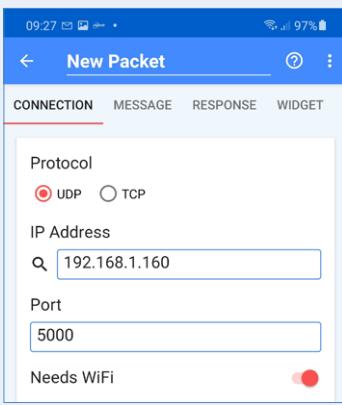


Figure 7 : Configuration de l'appli.

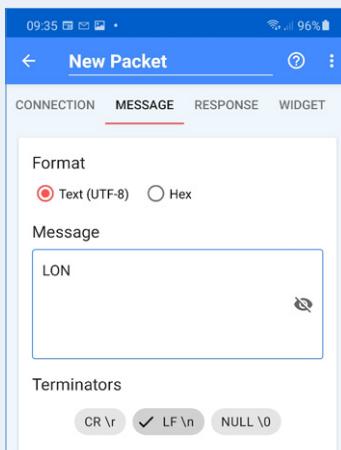


Figure 8 : Commande pour allumer la LED.

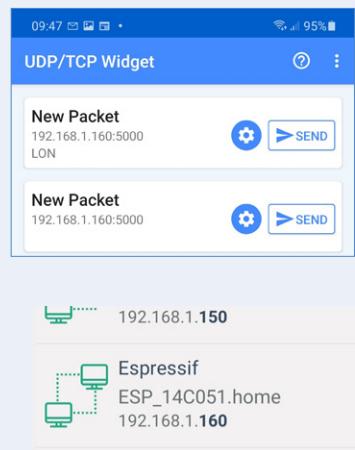


Figure 9 : Cliquez sur SEND pour transmettre la commande.

Figure 10 : Trouver l'adresse IP de l'ESP-01.



Produits



- **Livre en anglais « Raspberry Pi Pico Essentials »**
Version papier : www.elektor.fr/raspberry-pi-pico-essentials
Version numérique : www.elektor.fr/raspberry-pi-pico-essentials-e-book



- **Carte à microcontrôleur Raspberry Pi Pico**
www.elektor.fr/raspberry-pi-pico-microcontroller-board

Des questions, des commentaires ?
Envoyez un courriel à l'auteur (d.librahim@btinternet.com) ou contactez Elektor (redaction@elektor.fr).

Contributions

Auteur : Dogan Ibrahim
Rédaction : Jan Buiting
Mise en page : Harmen Heida
Traduction : Denis Lafourcade

LIEN

- [1] **Fichiers du programme d'exemple :** www.elektor.fr/raspberry-pi-pico-essentials