



# Raspberry Pi Pico Essentials

## voorbeeldhoofdstuk: WiFi met de Raspberry Pi Pico

Dogan Ibrahim (Verenigd Koninkrijk)

In deze aflevering van Elektor Boeken presenteren we ongeveer een half hoofdstuk uit Dogan Ibrahim's boek *Raspberry Pi Pico Essentials*, uitgegeven door Elektor in maart 2021 en inmiddels een bestseller. Destijds was het het eerste onafhankelijke boek waarin de omgang met de RP2040-microcontroller op de Raspberry Pi Pico-module in de praktijk werd beschreven, compleet met geteste software. Is 'de Pico' inderdaad zo toegankelijk als wordt beweerd, en maakt hij de verwachtingen waar van een 'drop-in' controllerboard? Dat zullen we in dit artikel ontdekken.

**Van de redactie:** dit artikel is een deel uit het boek *Raspberry Pi Pico Essentials*, enigszins aangepast en met een layout conform de redactionele normen en pagina-indeling van Elektor. Omdat het een gedeelte is uit een grotere publicatie, kunnen sommige termen in dit artikel verwijzen naar besprekingen elders in de oorspronkelijke boekpublicatie. Auteur en redactie hebben hun best gedaan om dit te vermijden, en zijn altijd bereid om te helpen met vragen – contactgegevens staan in het kader **Vragen of opmerkingen**

In dit artikel (oorspronkelijk ongeveer de helft van hoofdstuk 10 van het boek, *red.*) zullen we een project ontwikkelen waarin via een WiFi-verbinding communicatie tot stand wordt gebracht tussen de Raspberry Pi Pico en een smartphone.

### Een LED via WiFi aansturen vanaf een smartphone

**Beschrijving:** in dit project zenden we opdrachten via de WiFi-link van een mobiele telefoon om een LED aan te sturen (de LED kan worden vervangen door een relais, bijvoorbeeld om een apparaat in en uit te schakelen) die is aangesloten op de Raspberry Pi Pico. Commando's moeten worden beëindigd met een Return (CR/LF

of 'nieuwe regel'). Geldige opdrachten zijn onder meer:

**LON** Zet de LED AAN  
**LOFF** Zet de LED UIT

**Doel:** het doel van dit project is om het gebruik van WiFi-connectiviteit op de Raspberry Pi Pico te demonstreren.

**Pico WiFi-connectiviteit:** de Raspberry Pi Pico heeft geen ingebouwde WiFi-module en kan daarom niet worden verbonden met een WiFi-netwerk zonder koppeling met een externe WiFi-module. Waarschijnlijk de gemakkelijkste en goedkoopste manier de Pico van WiFi-mogelijkheden te voorzien,

is het gebruik van een ESP-01 processor-board. Dit is een kleine print (zie **figuur 1**) van slechts 2,7 x 1,2 cm op basis van de ESP8266-processor, en kost ongeveer vier dollar of vier tot vijf euro. De ESP-01 heeft de volgende interessante eigenschappen:

- > Voedingsspanning: +3,3 V
- > Interface: met eenvoudige AT-opdrachten via seriële poort/UART
- > Geïntegreerde TCP/IP-protocolstack
- > 802.11 b / g / n
- > Geen externe componenten nodig

De ESP-01 communiceert met de hostprocessor via zijn TX- en RX-seriële poort-pin. Het is een 8-pins board met deze pinbenamingen:

**VCC:** +3,3 V voeding

**GND:** voedingsmassa

**GPIO:** I/O-pin. Deze pin moet worden aangesloten op +3,3 V voor normaal bedrijf en op GND voor het uploaden van firmware naar de chip

**GPIO2:** I/O-pin voor algemeen gebruik  
**RST:** reset-pin. Moet worden aangesloten op +3,3 V voor normaal bedrijf  
**CH\_PD:** enable-pin inschakelen. Moet worden aangesloten op +3,3 V voor normaal bedrijf  
**TX:** seriële uitgang  
**RX:** seriële ingang

De pinnen van de ESP-01 zijn niet standaard breadboard-compatibel, dus een adapter is vereist als het board op een breadboard wordt gemonteerd (zie **figuur 2**).

**Blokschema:** **figuur 3** toont het blokschema van het project.

**Schema:** in **figuur 4** ziet u het schema van het project. De UART o TX- en RX-pinnen van de Raspberry Pi Pico worden gebruikt om met de ESP-01 te communiceren.

**Programmalisting:** **listing 1** geeft het programma (programmaam: *Picowifi*). Het is opgenomen in de cumulatieve download die te vinden is onder Downloads op de Elektor-webpagina bij het boek [1]. Binnen de setup-routine wordt de seriële communicatiesnelheid ingesteld op 115200, wat de standaard

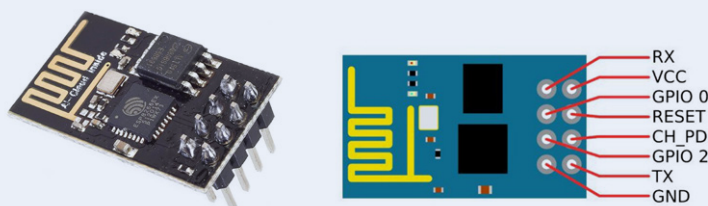
baudrate is voor ESP-01, en de LED is geconfigureerd als een uitgang en uitgeschakeld. De functie **ConnectToWiFi** wordt aangeroepen om verbinding te maken met de lokale WiFi-router. Commando's in AT-stijl worden gebruikt om de ESP-01 te configureren om verbinding te maken met de WiFi-router.

De rest van het programma loopt in een eindeloze lus die wordt gevormd met behulp van een *while*-instructie. Binnen deze lus worden gegevens ontvangen van de smartphone en wordt de LED navenant aangestuurd. De commando's **LON** en **LOFF** schakelen de LED respectievelijk AAN en UIT. Datapakketten worden van de smartphone ontvangen met behulp van de **readline**-functie. De functie **find** zoekt naar een substring in een string en retourneert een waarde die ongelijk nul is als de substring wordt gevonden. We gebruiken de functie **find** omdat de van de smartphone ontvangen gegevens dit formaat hebben: **+ ID0, n: data** (bijvoorbeeld **+ ID0,3: LON**) waarbij o de LinkID en n het aantal ontvangen tekens. Met de functie **find** kunnen we gemakkelijk zoeken naar de substrings **LON** of **LOFF** in het ontvangen datapakket.

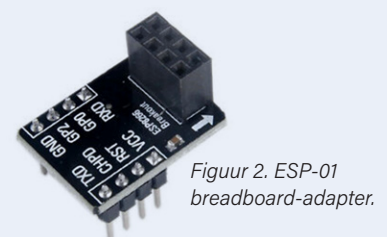
De functie **ConnectToWiFi** stuurt de volgende opdrachten naar de ESP-01 om verbinding te maken met het WiFi-netwerk:

**AT + RST** reset ESP-01  
**AT + CWMODE** stel de ESP-01-modus in (hier de Station-modus)  
**AT + CWJAP** stel W-Fi SSID-naam en wachtwoord in  
**AT + CPIMUX** stel de verbindingsmodus in (hier meerdere verbindingen)  
**AT + CIFS** retourneert het IP-adres (hier niet gebruikt)  
**AT + CIPSTART** stel de TCP- of UDP-verbindingsmodus, het IP-adres van de bestemming en het poortnummer in (hier wordt UDP gebruikt met het poortnummer ingesteld op 5000. Het IP-adres van de bestemming is ingesteld op "0.0.0.0" zodat elk apparaat gegevens kan zenden zolang poort 5000 wordt gebruikt (u kunt dit wijzigen in het IP-adres van uw smartphone om alleen gegevens van uw eigen telefoon te ontvangen).

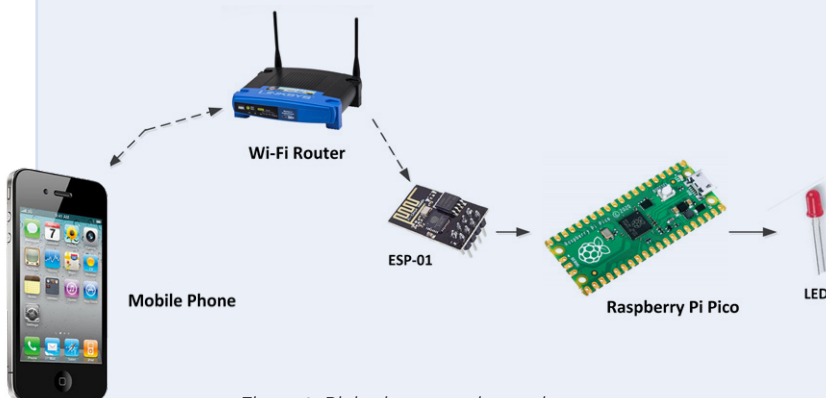
Merk op dat er na elke opdracht een kleine vertraging is ingevoegd. Het commando **AT + CWJAP** heeft een langere vertraging nodig. Het programma kan eenvoudig



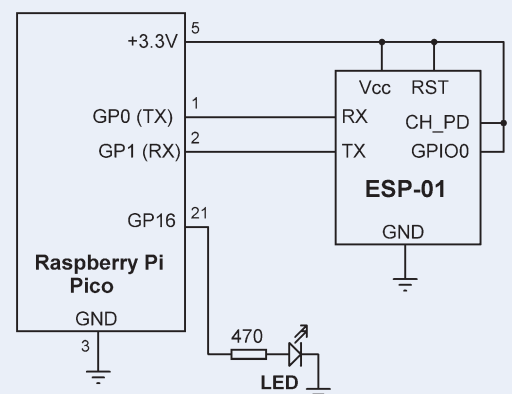
Figuur 1. Het ESP-01 processorprintje.



Figuur 2. ESP-01 breadboard-adaptor.



Figuur 3. Blokschema van het project.



Figuur 4. Schema van het project.



### Listing 1. Python-programma om een ESP-01 als WiFi-module te gebruiken

```
#-----
#           USING WI-FI
#           =====
#
# In this project a ESP-01 chip is connected to the Raspberry
# Pi Pico. This chip is used to connect the Pico to the Wi-Fi
#
# Author: Dogan Ibrahim
# File  : Picowifi.py
# Date  : February 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)

#
# Send AT commands to ESP-01 to connect to local WI-Fi
#
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()

#
# Main program loop
#
while True:
    buf = uart.readline()           # Read data
    dat = buf.decode('UTF-8')       # Decode
    n = dat.find("LON")             # Includes LON?
    if n > 0:
        LED.value(1)               # LED ON
    n = dat.find("LOFF")            # Includes OFF?
    if n > 0:
        LED.value(0)               # LED OFF
```

worden aangepast zodat de vertragingen kunnen worden verwijderd en de reacties van de ESP-01 kunnen worden gecontroleerd. Op deze manier kan het programma, zodra de juiste reactie is ontvangen, verder gaan. Mogelijk moet u de ESP-01-hardware resetten (uit- en weer inschakelen) voordat u het programma start.

### Test van het programma

Het programma kan eenvoudig worden getest met het programma *PacketSender* (zie **figuur 5**) op de PC of met een smartphone na installatie van een UDP-app.

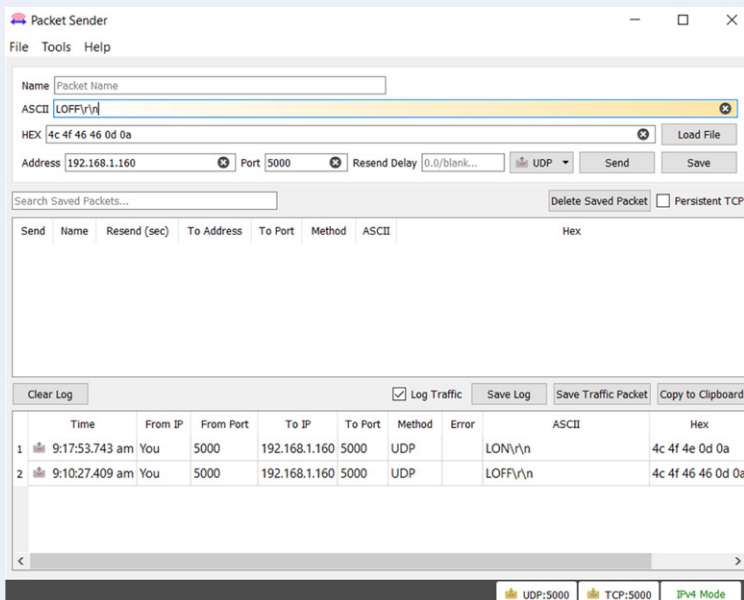
U moet een UDP Server-app op uw mobiele Android-telefoon installeren voordat u de test met de smartphone start. Er zijn veel gratis beschikbare UDP-apps in de Play Store. Die welke in dit project is geïnstalleerd en gebruikt, wordt door KJM de UDP/TCP-widget genoemd, zoals te zien in **figuur 6**.

Het programma kan als volgt worden getest:

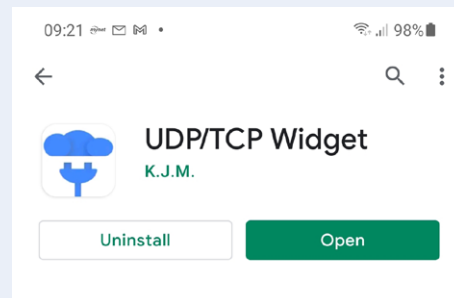
- Bouw de schakeling.
- Download het programma naar uw Raspberry Pi Pico.
- Start de *UDP/TCP Widget*-app op uw mobiele telefoon.
- Klik op het tandwielssymbool en stel het protocol in op *UDP*, het IP-adres op het *IP-adres* van uw Raspberry Pi Pico (192.168.1.160 in de Pico van de auteur), en stel *Port* in op 5000, zoals te zien in **figuur 7**.
- Klik op het menu-item *MESSAGE* en selecteer *Text (UTF-8)* als formaat, en voer de opdracht **LON** in om de LED in te schakelen. Gebruik *LF\r\n* als terminator en klik op het OK-symbool (vinkje), zoals in **figuur 8**.
- Klik nu op de *SEND*-knop (**figuur 9**) om het commando naar de Raspberry Pi Pico te sturen. U zou het bericht *Packet Sent* kort bovenaan uw Android-scherm moeten zien.

Merk op dat het IP-adres van de ESP-01 kan worden verkregen door alle apparaten op de lokale WiFi-router te scannen. De Android-app *Who Uses My WiFi – Network Scanner* van Phuongpn kan bijvoorbeeld worden gebruikt om de IP-adressen te zien van alle apparaten die met uw router zijn verbonden. De ESP-01 wordt weergegeven zoals in **figuur 10** (IP: 192.168.1.160), met de naam *Espressif*. ◀

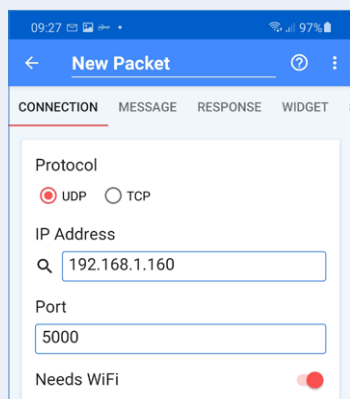
210198-03



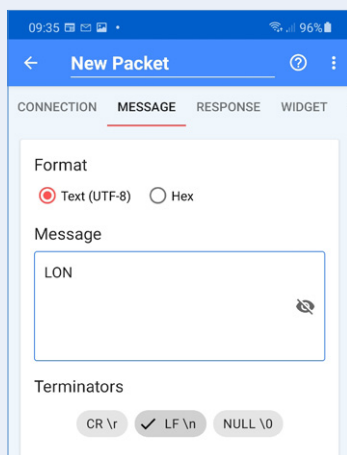
Figuur 5. De PacketSender om het programma te testen.



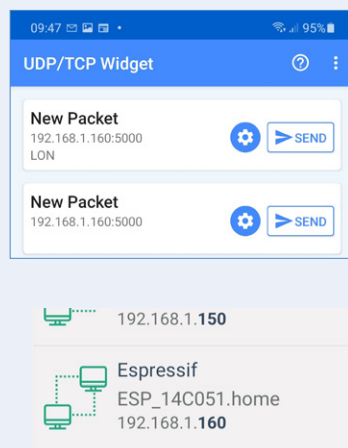
Figuur 6. UDP/TCP Widget-apps voor Android.



Figuur 7. Configuratie van de app.



Figuur 8. Commando om de LED in te schakelen.



Figuur 9. Klik op SEND om het commando te verzenden.

Figuur 10. Het IP-adres van de ESP-01 zoeken.



## Gerelateerde producten



- **Boek: "Raspberry Pi Pico Essentials"**  
[www.elektor.nl/raspberry-pi-pico-essentials](http://www.elektor.nl/raspberry-pi-pico-essentials)
- **E-boek: "Raspberry Pi Pico Essentials"**  
[www.elektor.nl/raspberry-pi-pico-essentials-e-book](http://www.elektor.nl/raspberry-pi-pico-essentials-e-book)



- **Raspberry Pi Pico Microcontroller Board**  
[www.elektor.nl/raspberry-pi-pico-microcontroller-board](http://www.elektor.nl/raspberry-pi-pico-microcontroller-board)

## Vragen of opmerkingen?

Hebt u technische vragen of opmerkingen naar aanleiding van dit artikel? Stuur een e-mail naar de auteur via [d.brahim@btinternet.com](mailto:d.brahim@btinternet.com) of naar de redactie van Elektor via [redactie@elektor.com](mailto:redactie@elektor.com).

## Een bijdrage van

Tekst: Dogan Ibrahim  
Redactie: Jan Buiting  
Vertaling: Eric Bogers  
Layout: Harmen Heida

## WEBLINK

- [1] **Voorbeeld-programmabestanden:**  
[www.elektor.com/raspberry-pi-pico-essentials](http://www.elektor.com/raspberry-pi-pico-essentials)