Wifi Analyzer Software

Aug 2026

Back to Pico Wifi Analyzer Phase 1

Introduction

When you did the section where you connected up the LCD to the Pico, there was some simple test Python scripts to see that the LCD works. Also, you have tested your Wifi connection to see that you can talk to your router. Remember, the Pico only supports 2.4Ghz Wifi.

Here is the Python code you need to save to your Pico. To start, I would just copy this code in to Thonny and run it from there.

| download

```
# ------------------------------------------------------
# Wifi Analyzer
# Scans Wifi SSIDs every 15 minutes to keep track of
# current Wifi status.
# Alan Walker & CoPilot
# 19/08/2026
# ------------------------------------------------------

from machine import Pin, I2C
import ssd1306
import time
import network
import socket

# ------------------------------------------------------
# OLED Setup
# ------------------------------------------------------
i2c = I2C(0, scl=Pin(5), sda=Pin(4), freq=200000)
lcd = ssd1306.SSD1306_I2C(128, 64, i2c)

# ------------------------------------------------------
# Manual Button (GP16)
# ------------------------------------------------------
button = Pin(16, Pin.IN, Pin.PULL_UP)   # active-low

# ------------------------------------------------------
# Onboard LED (Pico 2 W)
# ------------------------------------------------------
led = Pin("LED", Pin.OUT)

# Global WiFi status (updated after each scan)
wifi_ok = True

# ------------------------------------------------------
# LED Controller (continuous blink if ANY SSID down)
# ------------------------------------------------------
def update_led():
    global wifi_ok

    if wifi_ok:
        led.value(1)   # solid ON
    else:
        # continuous blink (non-blocking)
```

```
        led.value(1)
        time.sleep(0.15)
        led.value(0)
        time.sleep(0.15)

# ------------------------------------------------------
# UI Renderer
# ------------------------------------------------------
def draw_ui(header="WiFi Analyzer", last="--:--",
            linksys="------", draytek="------",
            status="Idle", next_scan="--m"):

    lcd.fill(0)

    lcd.text(header, 0, 0)
    lcd.text("Last: " + last, 0, 8)

    lcd.text("Linksys: " + linksys, 0, 24)
    lcd.text("Draytek: " + draytek, 0, 32)

    lcd.text("Status: " + status, 0, 48)
    lcd.text("Next: " + next_scan, 0, 56)

    lcd.show()

# ------------------------------------------------------
# Scanning Animation
# ------------------------------------------------------
def animate_scanning():
    frames = ["Scanning.", "Scanning..", "Scanning..."]
    for frame in frames:
        draw_ui(status=frame)
        update_led()
        time.sleep(0.3)

# ------------------------------------------------------
# WiFi Test Engine
# ------------------------------------------------------
def test_network(ssid, password):
    print("Trying:", ssid)

    wlan = network.WLAN(network.STA_IF)

    wlan.active(False)
    time.sleep(0.2)
    wlan.active(True)

    wlan.connect(ssid, password)

    for _ in range(20):
        if wlan.isconnected():
            break
        time.sleep(0.5)

    if not wlan.isconnected():
        return False

    try:
        socket.getaddrinfo("google.com", 80)
        return True
    except:
        return False

# ------------------------------------------------------
# Test SSIDs
# ------------------------------------------------------
networks = [
    ("SSID1", "YOUR-SSID-PASSWORD"),
    ("SSID2", "YOUR-SSID-PASSWORD")
]
```

```
# ------------------------------------------------------
# Rolling History
# ------------------------------------------------------
history = {
    "Linksys": [],
    "Draytek": []
}

# ------------------------------------------------------
# Startup Scan
# ------------------------------------------------------
animate_scanning()
print("Starting WiFi scan...")

time.sleep(1)

results = []
for ssid, pw in networks:
    ok = test_network(ssid, pw)
    results.append(ok)

# Update global WiFi status
wifi_ok = all(results)

linksys_status = "1" if results[0] else "x"
draytek_status = "1" if results[1] else "x"

history["Linksys"].append(linksys_status)
history["Draytek"].append(draytek_status)

if len(history["Linksys"]) > 5:
    history["Linksys"].pop(0)
if len(history["Draytek"]) > 5:
    history["Draytek"].pop(0)

# ------------------------------------------------------
# Time Sync
# ------------------------------------------------------
if any(results):
    try:
        import ntptime
        ntptime.settime()
        now = time.localtime(time.time() + 3600)
    except:
        now = time.localtime()
else:
    now = time.localtime()

network.WLAN(network.STA_IF).disconnect()

timestamp = "{:02d}:{:02d}".format(now[3], now[4])

linksys_hist = "".join(history["Linksys"])
draytek_hist = "".join(history["Draytek"])

draw_ui(
    last=timestamp,
    linksys=linksys_hist,
    draytek=draytek_hist,
    status="Done",
    next_scan="15m"
)

# ------------------------------------------------------
# Auto-scan loop
# ------------------------------------------------------
next_scan_seconds = 900
scanning_in_progress = False
```

```
while True:

    for _ in range(600):   # 60 seconds

        update_led()  # LED continues blinking if needed

        if not scanning_in_progress and not button.value():
            scanning_in_progress = True
            animate_scanning()
            print("Manual scan starting...")

            results = []
            for ssid, pw in networks:
                ok = test_network(ssid, pw)
                results.append(ok)

            wifi_ok = all(results)

            linksys_status = "1" if results[0] else "x"
            draytek_status = "1" if results[1] else "x"

            history["Linksys"].append(linksys_status)
            history["Draytek"].append(draytek_status)

            if len(history["Linksys"]) > 5:
                history["Linksys"].pop(0)
            if len(history["Draytek"]) > 5:
                history["Draytek"].pop(0)

            if any(results):
                try:
                    import ntptime
                    ntptime.settime()
                    now = time.localtime(time.time() + 3600)
                except:
                    now = time.localtime()
            else:
                now = time.localtime()

            network.WLAN(network.STA_IF).disconnect()

            timestamp = "{:02d}:{:02d}".format(now[3], now[4])

            linksys_hist = "".join(history["Linksys"])
            draytek_hist = "".join(history["Draytek"])

            draw_ui(
                last=timestamp,
                linksys=linksys_hist,
                draytek=draytek_hist,
                status="Done",
                next_scan=f"{next_scan_seconds // 60}m"
            )

            scanning_in_progress = False
            time.sleep(0.3)

        time.sleep(0.1)

    next_scan_seconds -= 60

    update_led()

    mins = next_scan_seconds // 60

    draw_ui(
        last=timestamp,
        linksys=linksys_hist,
        draytek=draytek_hist,
        status="Done",
```

```
            next_scan=f"{mins}m"
        )

        if next_scan_seconds <= 0:

            scanning_in_progress = True
            animate_scanning()
            print("Auto-scan starting...")

            results = []
            for ssid, pw in networks:
                ok = test_network(ssid, pw)
                results.append(ok)

            wifi_ok = all(results)

            linksys_status = "1" if results[0] else "x"
            draytek_status = "1" if results[1] else "x"

            history["Linksys"].append(linksys_status)
            history["Draytek"].append(draytek_status)

            if len(history["Linksys"]) > 5:
                history["Linksys"].pop(0)
            if len(history["Draytek"]) > 5:
                history["Draytek"].pop(0)

            if any(results):
                try:
                    import ntptime
                    ntptime.settime()
                    now = time.localtime(time.time() + 3600)
                except:
                    now = time.localtime()
            else:
                now = time.localtime()

            network.WLAN(network.STA_IF).disconnect()

            timestamp = "{:02d}:{:02d}".format(now[3], now[4])

            linksys_hist = "".join(history["Linksys"])
            draytek_hist = "".join(history["Draytek"])

            draw_ui(
                last=timestamp,
                linksys=linksys_hist,
                draytek=draytek_hist,
                status="Done",
                next_scan="15m"
            )

            next_scan_seconds = 900
            scanning_in_progress = False
```