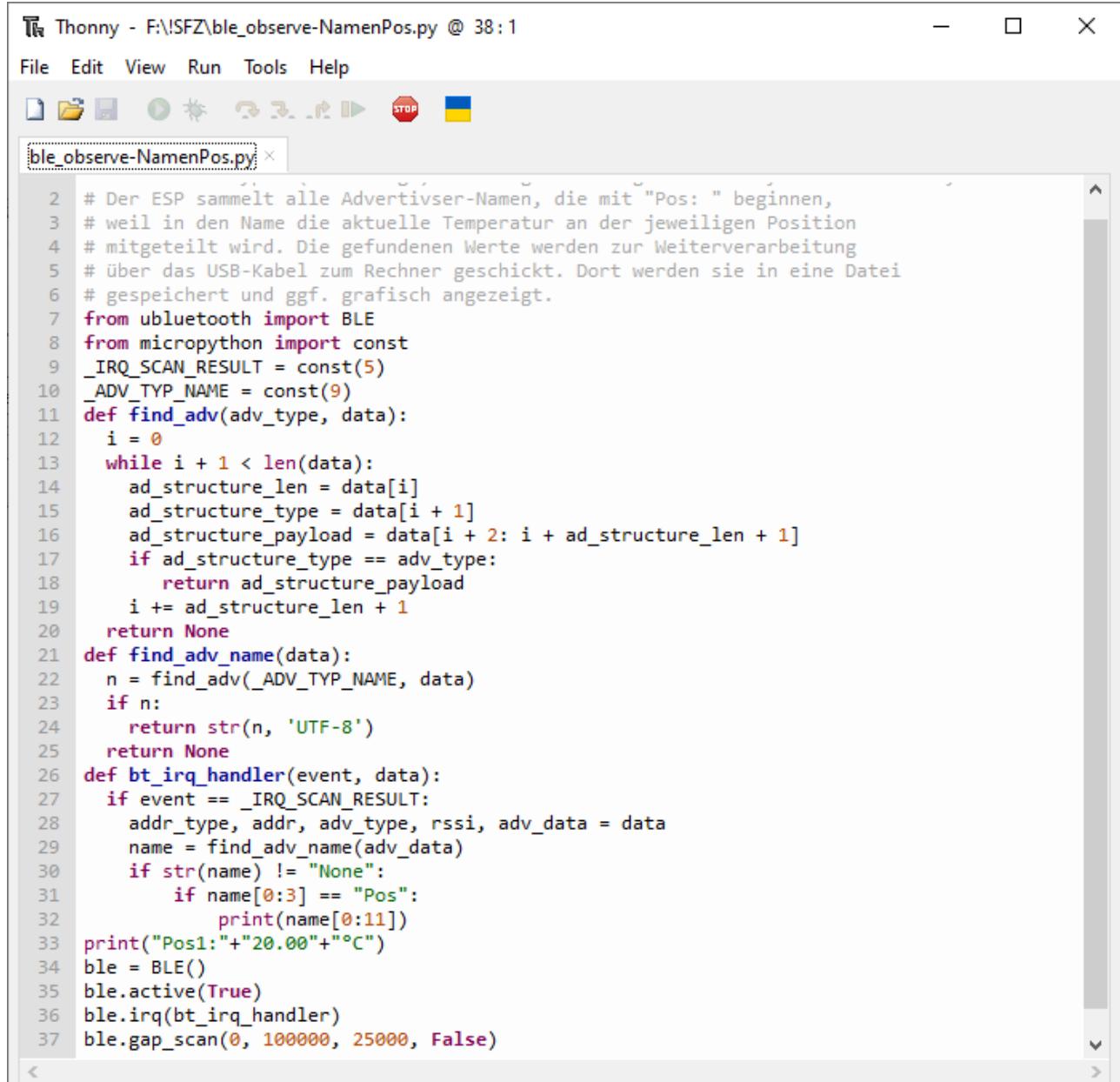


## Einsammeln der von den ESPs gesendeten Temperaturwerte mit "ble\_observe-NamenPos.py"

Einsammeln der von den ESPs gesendeten Temperaturwerte, die mit "ble\_adv\_DS1820\_Pos1.py", "ble\_adv\_DS1820\_Pos2.py" etc. programmiert wurden. Deren Übertragung auf einen Rechner kann mit dem folgenden Programm "ble\_observe-NamenPos.py" übernommen werden. Dabei ist Voraussetzung, das die BLE-Thermometer senden und der Rechner über USB-Kabel an den ESP32 angeschlossen ist und auf dem Rechner muss z.B. das Programm "BLE-Seriell-auf-W10-inDatei.py" laufen.



The screenshot shows the Thonny Python IDE interface with the script `ble_observe-NamenPos.py` open. The code is written in Python and uses the `ubluetooth` and `micropython` modules. It defines functions to find specific advertisement types and names, and a handler for BLE events to print names starting with "Pos:" followed by a temperature value in Celsius.

```
2 # Der ESP sammelt alle Advertivser-Namen, die mit "Pos: " beginnen,
3 # weil in den Name die aktuelle Temperatur an der jeweiligen Position
4 # mitgeteilt wird. Die gefundenen Werte werden zur Weiterverarbeitung
5 # über das USB-Kabel zum Rechner geschickt. Dort werden sie in eine Datei
6 # gespeichert und ggf. grafisch angezeigt.
7 from ubluetooth import BLE
8 from micropython import const
9 _IRQ_SCAN_RESULT = const(5)
10 _ADV_TYP_NAME = const(9)
11 def find_adv(adv_type, data):
12     i = 0
13     while i + 1 < len(data):
14         ad_structure_len = data[i]
15         ad_structure_type = data[i + 1]
16         ad_structure_payload = data[i + 2: i + ad_structure_len + 1]
17         if ad_structure_type == adv_type:
18             return ad_structure_payload
19         i += ad_structure_len + 1
20     return None
21 def find_adv_name(data):
22     n = find_adv(_ADV_TYP_NAME, data)
23     if n:
24         return str(n, 'UTF-8')
25     return None
26 def bt_irq_handler(event, data):
27     if event == _IRQ_SCAN_RESULT:
28         addr_type, addr, adv_type, rssi, adv_data = data
29         name = find_adv_name(adv_data)
30         if str(name) != "None":
31             if name[0:3] == "Pos":
32                 print(name[0:11])
33     print("Pos1:"+str(rssi)+"°C")
34 ble = BLE()
35 ble.active(True)
36 ble.irq(bt_irq_handler)
37 ble.gap_scan(0, 100000, 25000, False)
```