

HTTP-Client

HTTP-Client

[Routine explanation](#)

[Introduction](#)

[Complete code](#)

[flow chart](#)

[HTTP Client](#)

[What it can be used for](#)

Routine explanation

Introduction

In this section, we use K230 as an HTTP client. We use K230 to send HTTP message format requests to the server and receive messages sent back by the server. This can be understood in a simple way: let K230 do the work that a normal browser does.

In order to facilitate everyone's testing, in this section we will use k230 to access

www.baidu.com

Because this is not a local server, the WIFI that the K230 is connected to needs to be able to access the external network. That is to say, if your mobile phone or computer is connected to this WIFI, your mobile phone or computer needs to be able to open the URL

www.baidu.com.

We open the example code and modify the parameters of the Connect_WIFI function in the main function to SSID (the name of the WIFI) and KEY (the password of the WIFI)

```
Connect_WIFI("Yahboom", "yahboom890729")
```

After modification, we click Run in the lower left corner and can see the following output:

```
Connection address: ('183.2.172.42', 80)
```

```
b'HTTP/1.0 200 OK\r\nAccept-Ranges: bytes\r\nCache-Control: no-cache\r\nContent-Length: 29506\r\n .... The following part is too long and omitted'
```

The first line indicates the IP address that the domain name www.baidu.com actually points to.

The second line is the HTTP response containing the HTML page code returned by the server when we visit the page www.baidu.com/index.html. If we use a browser to visit www.baidu.com/index.html, when the browser gets the returned HTML code, it will parse the returned value into the search interface that we often see.

Complete code

The complete code file is in [Source Code/11.Network/05.http/http_client.py]

```
import network
import socket
import os, time

HOST = "www.baidu.com"

def Connect_WIFI(ssid, key):
    """
    使用 WLAN 连接网络。
    Connect to a WLAN network.

    :param ssid: WLAN 网络的 SSID
    :param key: WLAN 网络的密钥
    :return: 网络接口的 IP 地址
    """
    sta = network.WLAN(0)
    sta.connect(ssid, key)
    while not sta.isconnected():
        time.sleep(1)
    return sta.ifconfig()[0]

def main(use_stream=True):
    """
    主要函数，完成以下步骤：
    1. 连接 WLAN 网络
    2. 创建 socket
    3. 获取主机地址和端口号
    4. 连接到主机
    5. 发送 HTTP GET 请求并打印响应

    The main function that does the following:
    1. Connect to a WLAN network
    2. Create a socket
    3. Get the host address and port number
    4. Connect to the host
    5. Send an HTTP GET request and print the response

    :param use_stream: 是否使用流式 socket 进行读取
    """
    global HOST
    Connect_WIFI("Yahboom", "yahboom890729")

    s = socket.socket()

    ai = []
    for attempt in range(0, 3):
        try:
            ai = socket.getaddrinfo(HOST, 80)
            break
        except Exception as e:
            print("getaddrinfo again", e)

    if ai == []:
```

```
    print("连接错误 Connect error")
    s.close()
    return

addr = ai[0][-1]
print("连接地址 address:", addr)

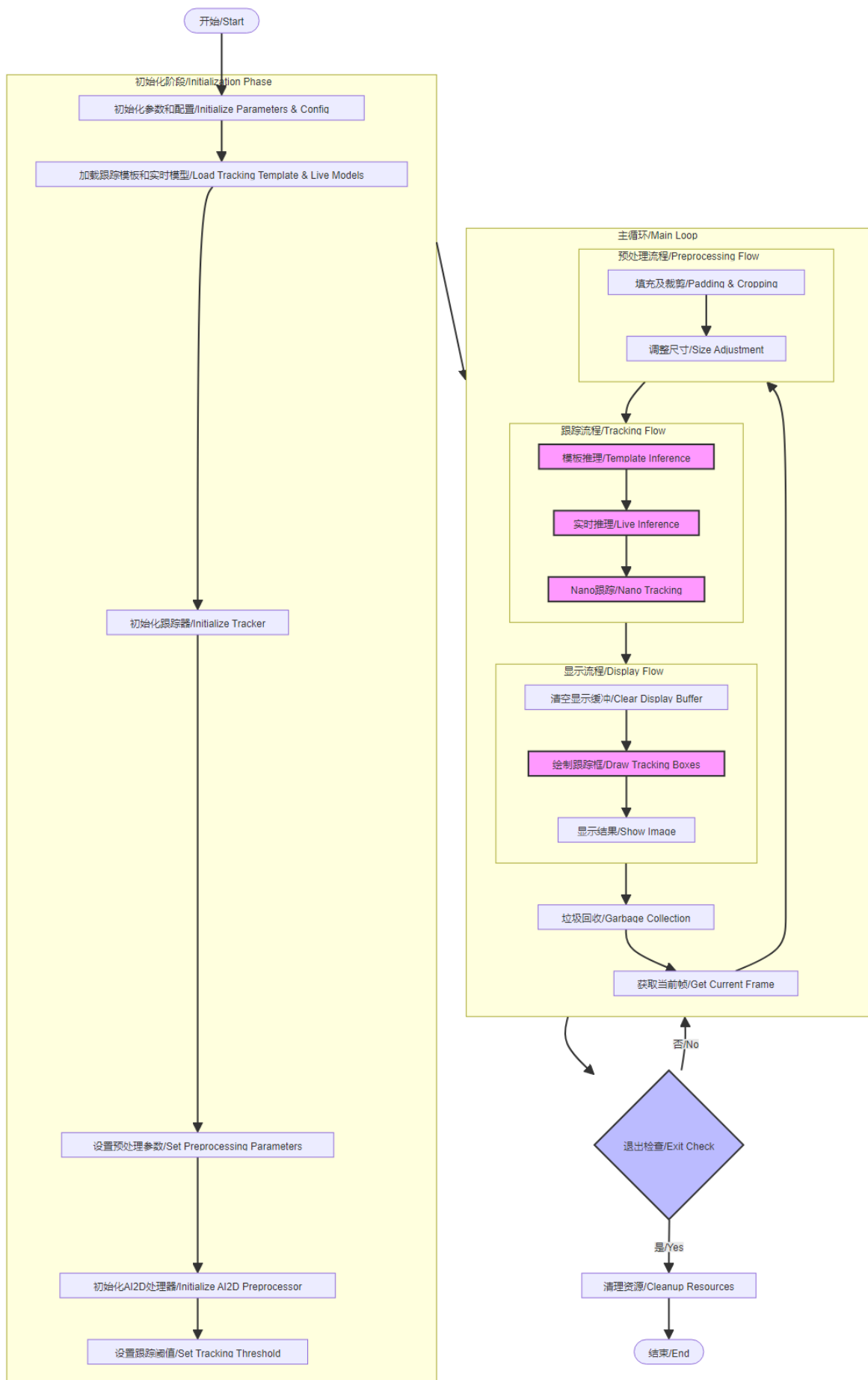
s.connect(addr)

if use_stream:
    s = s.makefile("rwb", 0)
    s.write(b"GET /index.html HTTP/1.0\r\n\r\n")
    print(s.read())
else:
    s.send(b"GET /index.html HTTP/1.0\r\n\r\n")
    print(s.recv(4096))

s.close()

main()
```

flow chart



What it can be used for

HTTP is a very common protocol. In many IoT scenarios, HTTP is often used for communication between embedded devices and servers.

At the same time, if you have access to the Internet, there are many feature-rich API interfaces available for us to call.

Extension: Request module call