

Timer

Timer

[Introduction to the results of routine experiments](#)

[Code Explanation](#)

[Complete code](#)

[Code structure](#)

Introduction to the results of routine experiments

In this section, we will learn how to use the K230 timer

K230 contains 6 Timer hardware modules, with the minimum timing period of 1 microsecond. These timers can be used to achieve precise timing and periodic tasks.

In this section, we set a one-shot timer and a periodic timer.

A single timer will only trigger the callback function once

The periodic timer will continuously execute the callback function according to the set period duration during the program.

We run the example code in this section and can see the following output in the console:

```
启动单次定时器 Starting one-shot timer...
单次定时器触发了! Single shot timer triggered!
启动周期定时器 Starting periodic timer...
周期定时器触发了! Periodic timer triggered!
周期定时器触发了! Periodic timer triggered!
周期定时器触发了! Periodic timer triggered!
Timer deinitialized
```

Code Explanation

The code file is located at: [Source code/02.Basic/09.timer.py]

Complete code

```
from machine import Timer
import time

def timer_callback_once(t):
    """
    单次定时器回调函数
    One-shot timer callback function
    """
    print("单次定时器触发了! Single shot timer triggered!")
```

```

def timer_callback_periodic(t):
    """
    周期性定时器回调函数
    Periodic timer callback function
    """
    print("周期定时器触发了! Periodic timer triggered!")

try:
    # 实例化一个软定时器
    # Initialize a virtual timer
    timer = Timer(-1)

    # 配置单次模式定时器, 周期为100ms
    # Configure one-shot timer with 100ms period
    print("启动单次定时器 Starting one-shot timer...")
    timer.init(period=100,
                mode=Timer.ONE_SHOT,
                callback=timer_callback_once)

    # 等待单次定时器触发完成
    # Wait for one-shot timer to complete
    time.sleep(0.2)

    # 配置周期模式定时器, 频率为1Hz (周期1秒)
    # Configure periodic timer with 1Hz frequency (1 second period)
    print("启动周期定时器 Starting periodic timer...")
    timer.init(freq=1,
                mode=Timer.PERIODIC,
                callback=timer_callback_periodic)

    # 让周期定时器运行4秒
    # Let periodic timer run for 4 seconds
    time.sleep(4)

except Exception as e:
    print(f"Error occurred: {e}")

finally:
    # 释放定时器资源
    # Release timer resources
    timer.deinit()
    print("Timer deinitialized")

```

Code structure

1. Import necessary modules `machine` and `time`.
2. Define two timer callback functions `timer_callback_once` and `timer_callback_periodic`.
3. In the try block:
 - Create a virtual timer instance.
 - Configure and start a one-shot timer, and set the callback function to `timer_callback_once`.
 - Waiting for the one-shot timer to fire to complete.
 - Configure and start the periodic timer and set the callback function to `timer_callback_periodic`.
 - Let the period timer run for 2 seconds.

4. If an exception occurs, catch it and print the error.

5. In the finally block:

- Release timer resources.
- The print timer has been released.

The complete flow chart is as follows:



