

QuecPython

Timer 定时器使用说明

LTE 系列

版本: Timer 定时器开发使用说明_V1.0

日期: 2020-12-29

状态: 临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市闵行区田林路 1016 号科技绿洲 3 期（B 区）5 号楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

<http://www.quectel.com/cn/support/sales.htm>

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

<http://www.quectel.com/cn/support/technical.htm>

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2020，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2020.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2020-12-29	唐大为	初始版本

目录

1	EC600S Timer 基本概述.....	- 5 -
2	硬件描述.....	- 5 -
3	软件设计.....	- 5 -
3.1	常量说明.....	- 5 -
3.2	创建 Timer 对象.....	- 5 -
3.3	启动定时器.....	- 5 -
3.4	关闭定时器.....	- 6 -
4	交互操作.....	- 6 -
5	下载验证（待写）.....	- 6 -
5.1	软件代码.....	- 6 -
5.2	硬件连接：.....	- 7 -
5.3	运行效果.....	- 7 -

1 EC600S Timer 基本概述

文档主要基于 EC600S 介绍如何使用 QuecPython_Timer，定时器最常用的就是定时与计数，即可以作为精准延时处理，也可以在接通一个时钟源的前提下，统计总共有多少脉冲。通过本文你将了解到 Timer 的所有设置参数及使用方法。

2 硬件描述

目前开放共 4 个定时器

3 软件设计

3.1 常量说明

常量	说明
Timer.Timer0	定时器 0
Timer.Timer1	定时器 1
Timer.Timer2	定时器 2
Timer.Timer3	定时器 3
Timer.ONE_SHOT	单次模式，定时器只执行一次
Timer.PERIODIC	周期模式，定时器循环执行

3.2 创建 Timer 对象

timer = Timer(Timern)。创建 Timer 对象接口参数介绍如下：

参数	类型	说明
Timern	int	定时器号。EC600S 支持定时器 Timer0~Timer3

3.3 启动定时器

timer.start(period, mode, callback)：启动对应的定时器，接口参数介绍如下：

参数	类型	说明
period	int	中断周期，单位毫秒
mode	int	运行模式 Timer.ONE_SHOT 单次模式，定时器只执行一次 Timer.PERIODIC 周期模式，循环执行
callback	function	定时器执行函数

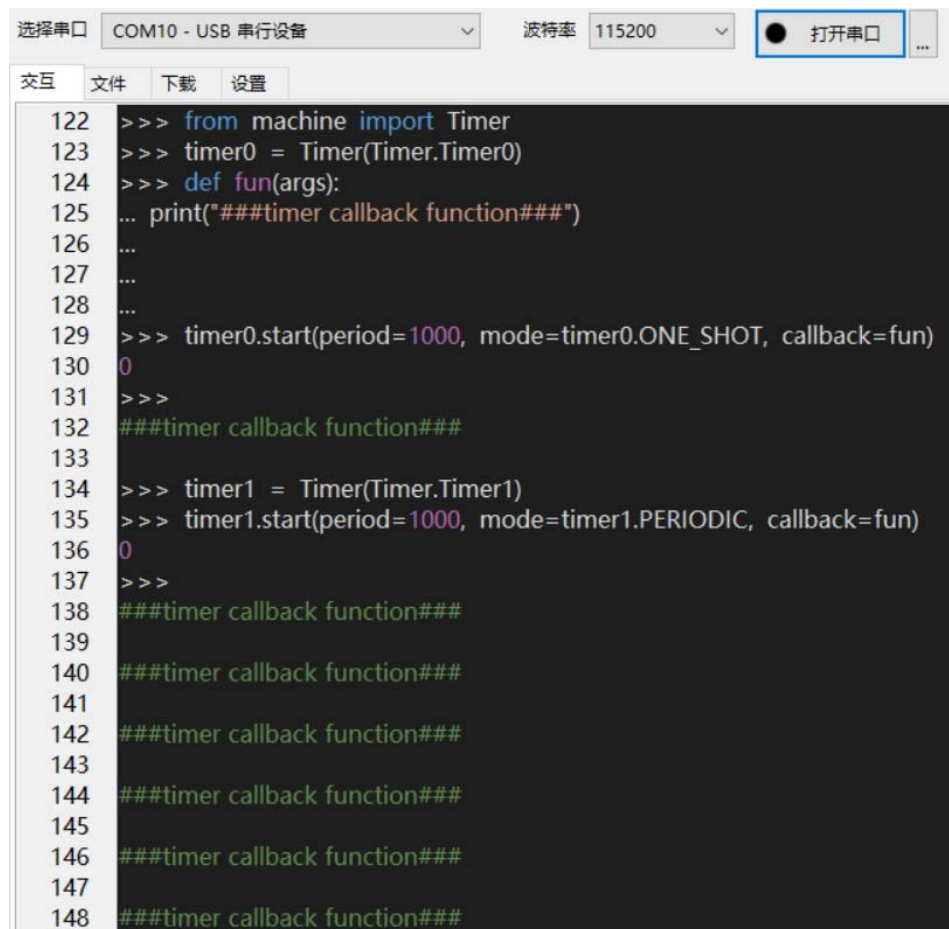
返回值：启动成功返回整型值 0，失败返回整型值-1。

3.4 关闭定时器

timer.stop(): 关闭对应的定时器，无参数。
返回值：成功返回整型 0，失败返回整型-1

4 交互操作

使用 QPYcom 工具和模组进行交互，下面实例是基于 Timer0，Timer1。Timer2 和 Timer3 配置类似。



```

122 >>> from machine import Timer
123 >>> timer0 = Timer(Timer.Timer0)
124 >>> def fun(args):
125 ... print("###timer callback function###")
126 ...
127 ...
128 ...
129 >>> timer0.start(period=1000, mode=timer0.ONE_SHOT, callback=fun)
130 0
131 >>>
132 ###timer callback function###
133
134 >>> timer1 = Timer(Timer.Timer1)
135 >>> timer1.start(period=1000, mode=timer1.PERIODIC, callback=fun)
136 0
137 >>>
138 ###timer callback function###
139
140 ###timer callback function###
141
142 ###timer callback function###
143
144 ###timer callback function###
145
146 ###timer callback function###
147
148 ###timer callback function###

```

注意：

1. from machine import Timer 即为让 Timer 模块在当前空间可见。
2. 只有 from machine import Timer 模块，才能使用 Timer 内的函数和变量。
3. 上述操作没有连接任何外设，仅作为熟悉指令参考

5 下载验证（待写）

5.1 软件代码

配套 demo 的参考代码为文档同目录下的 timer_file.py 文件。下载.py 文件到模组运行，代码如下：

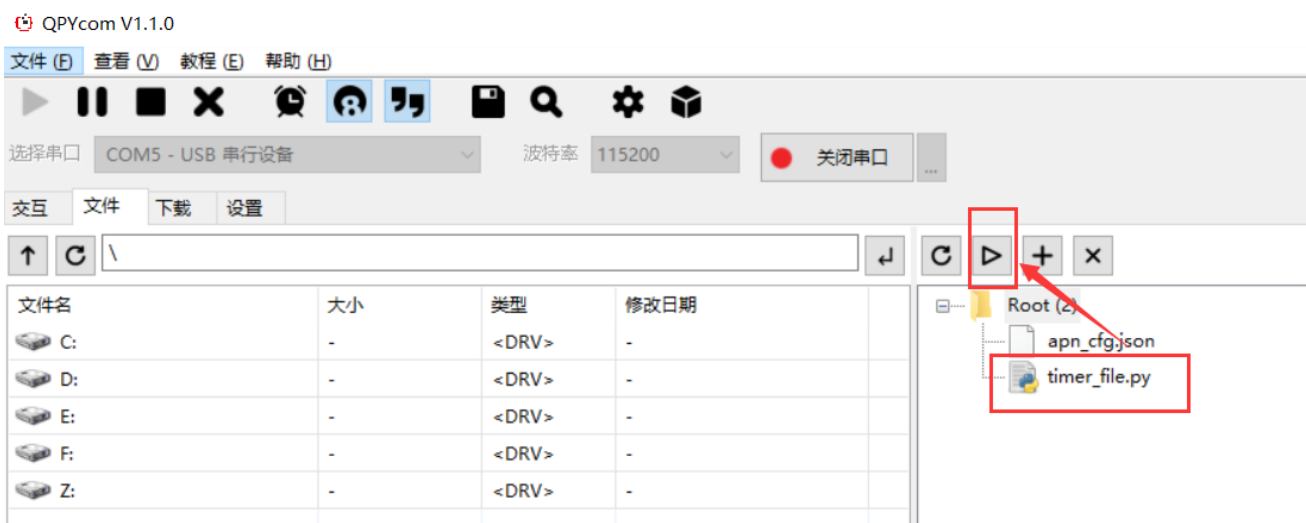
```
import log
from machine import Timer
log.basicConfig(level=log.INFO) # 设置日志输出级别
Timer_Log = log.getLogger("Quectel") # 获取 logger 对象
log_print_num = 5
state = 1
timer0 = Timer(Timer.Timer1)
# 创建一个执行函数，并将 timer 实例传入
def timer_test(t):
    global log_print_num
    global state
    Timer_Log.info('log_print_num is %d' % log_print_num)
    log_print_num -= 1
    if log_print_num <= 0:
        Timer_Log.info('timer exit')
        state = 0
        timer0.stop() # 结束该定时器实例
timer0.start(period=1000, mode=timer0.PERIODIC, callback=timer_test) # 启动定时器
while state:
    pass
```

5.2 硬件连接:

本文验证仅对 timer，无需外接外设。

5.3 运行效果

(1) 打开 QPYcom 运行 timer_file.py，如下图:



(2) 在 QPYcom 交互界面查看输出结果如下:

