

# 动态定时器

RealTouch 评估板 RT-Thread 入门文档

版本号：1.0.0

日期：2012/8/14

修订记录

日期	作者	修订历史
2012/8/14	bloom5	创建文档

# 实验目的

---

- ❑ 快速熟悉了解动态定时器相关的 API

# 硬件说明

---

本实验使用 RT-Thread 官方的 Realtouch 开发板作为实验平台。涉及到的硬件主要为

- ❑ 串口 3，作为 rt\_kprintf 输出，需要连接 JTAG 扩展板  
具体请参见《Realtouch 开发板使用手册》

# 实验原理及程序结构

---

## 实验设计

本实验的主要设计目的是帮助读者了解动态定时器的创建、删除过程熟悉相关。请读者注意，本实验本身不具有实际的工程参考价值，只是帮助读者快速了解相关 API 的用法。

## 源程序说明

本实验对应 l\_kernel\_timer\_dynamic

### 系统依赖

在 rtconfig.h 中需要开启

- ❑ #define RT\_USING\_HEAP

此项可选，开启此项可以创建动态线程和动态信号量，如果使用静态线程和静态信号量，则此项不是必要的

- ❑ #define RT\_USING\_CONSOLE

此项必须，本实验使用 rt\_kprintf 向串口打印按键信息，因此需要开启此项

- ❑ #define RT\_USING\_TIMER\_SOFT

此项必须，本实验使用软件定时器，因此必须开启此项

### 主程序说明

这个例程会创建两个动态定时器对象，一个是单次定时，一个是周期性的定时。

### 定时器创建

```
timer1 = rt_timer_create("timer1", /* 定时器名字是 timer1 */
    timeout1, /* 超时回调的处理函数 */
    RT_NULL, /* 超时函数的入口参数 */
    1000, /* 定时长度，以 OS Tick 为单位，即 10 个 OS Tick */
    RT_TIMER_FLAG_PERIODIC); /* 周期性定时器 */

/* 启动定时器 */
if (timer1 != RT_NULL)
    rt_timer_start(timer1);

/* 创建定时器 2 */
timer2 = rt_timer_create("timer2", /* 定时器名字是 timer2 */
    timeout2, /* 超时回调的处理函数 */
    RT_NULL, /* 超时函数的入口参数 */
    3000, /* 定时长度为 30 个 OS Tick */
    RT_TIMER_FLAG_ONE_SHOT); /* 单次定时器 */

/* 启动定时器 */
if (timer2 != RT_NULL)
    rt_timer_start(timer2);
```

### 两个定时器的超时函数

```
static void timeout1(void* parameter)
{
    rt_kprintf("periodic timer is timeout\n");
}

/* 定时器 2 超时函数 */
static void timeout2(void* parameter)
{
    rt_kprintf("one shot timer is timeout\n");
}
```

## 编译调试及观察输出信息

编译请参见《RT-Thread 配置开发环境指南》完成编译烧录，参考《Realtouch 开发板使用手册》完成硬件连接，连接扩展板上的串口和jlink。

运行后可以看到如下信息：

[illegible]

## 结果分析

本例程主要展示的就是两种不同定时器的工作方式，周期定时器每隔一秒触发一次超时函数，单次定时器则是在第 3 秒是触发一次超时函数。这 1 秒和 3 秒是如何确定的？可以看一下 `rtconfig.h` 文件，里面 `RT TICK PER SECOND` 目前被定为 1000，也就是说 1000 `os tick` 为一秒。