

# 操作系统开发指导手册

版本：V1.0.2

发布日期：2022/10/14

# 服务与支持

如果您有任何关于模组产品及产品手册的评论、疑问、想法，或者任何无法从本手册中找到答案的疑问，请通过以下方式联系我们。



## 中移物联网有限公司

OneMO官网: [onemo10086.com](http://onemo10086.com)

邮箱: [SmartModule@cmiot.chinamobile.com](mailto:SmartModule@cmiot.chinamobile.com)

客户服务热线: 400-110-0866

微信公众号: CMOneMO



中国移动  
China Mobile

# 文档声明

## 注意

本手册描述的产品及其附件特性和功能，取决于当地网络设计或网络性能，同时也取决于用户预先安装的各种软件。由于当地网络运营商、ISP，或当地网络设置等原因，可能也会造成本手册中描述的全部或部分产品及其附件特性和功能未包含在您的购买或使用范围之内。

## 责任限制

除非合同另有约定，中移物联网有限公司对本文档内容不做任何明示或暗示的声明或保证，并且不对特定目的适销性及适用性或者任何间接的、特殊的或连带的损失承担任何责任。

在适用法律允许的范围内，在任何情况下，中移物联网有限公司均不对用户因使用本手册内容和本手册中描述的产品而引起的任何特殊的、间接的、附带的或后果性的损坏、利润损失、数据丢失、声誉和预期的节省而负责。

因使用本手册中所述的产品而引起的中移物联网有限公司对用户的最大赔偿（除在涉及人身伤害的情况中根据适用法律规定的损害赔偿外），不应超过用户为购买此产品而支付的金额。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。公司保留随时修改本手册中任何信息的权利，无需进行提前通知且不承担任何责任。

## 商标声明



为中国移动注册商标。

本手册和本手册描述的产品中出现的其他商标、产品名称、服务名称和公司名称，均为其各自所有者的财产。

## 进出口法规

出口、转口或进口本手册中描述的产品（包括但不限于产品软件和技术数据），用户应遵守相关进出口法律和法规。

## 隐私保护

关于我们如何保护用户的个人信息等隐私情况，请查看相关隐私政策。

## 操作系统更新声明

操作系统仅支持官方升级；如用户自己刷非官方系统，导致安全风险和损失由用户负责。

## 固件包完整性风险声明

固件仅支持官方升级；如用户自己刷非官方固件，导致安全风险和损失由用户负责。

## 版权所有©中移物联网有限公司。保留一切权利。

本手册中描述的产品，可能包含中移物联网有限公司及其存在的许可人享有版权的软件，除非获得相关权利人的许可，否则，非经本公司书面同意，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并以任何形式传播。



中国移动  
China Mobile

# 关于文档

## 修订记录

版本	描述
V1.0.0	初版
V1.0.1	更新手册适用范围
V1.0.2	更新手册适用范围



中国移动  
China Mobile

# 目录

服务与支持..... ii

文档声明..... iii

关于文档..... v

1. 概述..... 7

    1.1. 适用范围..... 7

    1.2. 写作目的..... 7

2. 功能说明..... 8

3. API说明..... 9

4. 应用指导..... 10

    4.1. 线程..... 11

    4.2. 消息队列..... 13

    4.3. 互斥锁..... 14

    4.4. 信号量..... 15

    4.5. 延时..... 16

    4.6. 软件定时器..... 17

5. 编程设计注意..... 18



中国移动  
China Mobile

# 1. 概述

本文档对OpenCPU操作系统进行使用说明。

## 1.1. 适用范围

Table 1. 适用模组

模组系列	模组子型号
MN316	MN316-DBRS/MN316-DLVS
MN316-S	MN316-S-DLVS
M5310-E	M5310-E-BR/M5310-E-LR
ML302	ML302-ANLM/ML302-CNLM/ML302-DNLM/ML302-ENLM/ML302-FNLM/ML302-GNLM/ML302-HNLM/ML302-INLM/ML302-JNLM/ML302-KNLM/ML302-LNLM/ML302-MNLM/ML302-PNLM/ML302-QNLM
ML305	ML305-DNLM/ML305-RNLM/ML305-SNLM/ML305-TNLM
ML307A	ML307A-DCLN/ML307A-DSLN

## 1.2. 写作目的

本文档主要介绍OpenCPU使用的操作系统类型，各主要功能的使用方法，以及编程中的注意事项，帮助用户快速熟悉操作系统的使用。

## 2. 功能说明

OpenCPU操作系统提供符合cmsis v2标准的API。

Table 2. 系统心跳周期

模组	系统心跳周期
ML302/ML305	20ms
MN316/MN316-S/M5310-E	1ms
ML307A	5ms

操作系统功能主要包含线程、消息队列、互斥锁、信号量、延时、软件定时器。



中国移动  
China Mobile



## 3. API说明

OpenCPU操作系统API数量、名称及参数的详细说明请参考对应版本OpenCPU SDK内的API手册。



中国移动  
China Mobile

## 4. 应用指导

本章对线程、消息队列、互斥锁、信号量、延时、软件定时器提供示例代码说明。



中国移动  
China Mobile

## 4.1. 线程

本节主要对OpenCPU线程的用法和注意事项进行说明。

### 创建线程

```
osThreadAttr_t attr = {
    .name = "OpenCPU-DEMO",
    .priority = osPriorityNormal,
    .stack_size = 1024, // size = 1024*4
};

g_opencpu_tsk_handle = osThreadNew(cm_task_loop, NULL, (const osThreadAttr_t *)&attr);
```

#### Important:

ML302/ML305模组用户线程优先级必须低于或等于osPriorityNormal;  
ML307A模组用户线程优先级必须低于或等于osPriorityNormal，线程名称长度不超过5个字符;  
M5310-E模组用户线程优先级必须低于或等于osPriorityLow3;  
MN316/MN316-S用户线程优先级必须低于或等于osPriorityNormal3。

### 获取线程状态

```
osThreadState_t stat = osThreadGetState(thread_id);
```

### 获取线程剩余栈空间大小

```
uint32_t stack_space = osThreadGetStackSize(thread_id);
```

### 让出CPU,调度相同优先级的其他线程

```
osStatus_t status = osThreadYield();
```

### 强制退出指定线程

```
osStatus_t status = osThreadTerminate(thread_id);
```

#### Important: ML307A强制退出后资源不释放，建议线程创建后不退出。

### 退出当前线程

```
void demo_task(void* arg)
{
    do_something();
    osThreadExit();
}
```

 **Important:** ML307A不支持此功能。



中国移动  
China Mobile

## 4.2. 消息队列

本节主要以示例对消息队列功能做介绍。

### 创建消息队列

```
osMessageQueueId_t g_test_queue = NULL;  
g_test_queue = osMessageQueueNew(10, sizeof(void*), NULL);
```

### 发送数据

```
void* msg = cm_malloc(128);  
osMessageQueuePut(g_test_queue, &msg, 0, osNoWait);
```

### 接收数据

```
void* msg = NULL;  
if(osMessageQueueGet(g_test_queue, &msg, NULL, osWaitForever) == osOK){  
    do_something();  
}  
cm_free(msg);
```

### 删除队列

```
osMessageQueueDelete(g_test_queue);  
g_test_queue = NULL;
```

## 4.3. 互斥锁

本节主要以代码示例对互斥锁进行介绍。

### 创建互斥锁

```
osMutexId_t g_test_mutex = NULL;  
g_test_mutex = osMutexNew(NULL);
```

### 持有锁

```
if(g_test_mutex != NULL)  
{  
    osMutexAcquire(g_test_mutex, osWaitForever);  
}  
  
do_something();  
osMutexRelease(g_test_mutex);
```

### 释放锁

```
if(g_test_mutex != NULL)  
{  
    osMutexAcquire(g_test_mutex, osWaitForever); //持有锁  
}  
  
do_something();  
osMutexRelease(g_test_mutex); //释放锁
```

### 删除互斥锁


```
if(mutex != NULL)  
{  
    osMutexDelete(mutex);  
    mutex = NULL;  
}
```

## 4.4. 信号量

本节主要以代码示例对信号量进行说明。

### 创建信号量

```
osSemaphoreId_t g_test_semap = NULL;  
g_test_seamp = osSemaphoreNew(1, 0, NULL);
```

 **Important:** ML307A仅支持二值信号量。

### 等待信号

```
if(g_test_semap != NULL)  
{  
    osSemaphoreAcquire(g_test_seamp, osWaitForever);  
}  
  
do_something();
```

### 发出信号

```
do_something();  
if(g_test_semap != NULL)  
{  
    osSemaphoreRelease(g_test_seamp);  
}
```

### 删除信号量

```
if(g_test_seamp != NULL)  
{  
    osSemaphoreDelete(g_test_seamp);  
    g_test_seamp = NULL;  
}
```

## 4.5. 延时

本节主要以代码示例介绍延时功能。

### 延时指定心跳周期数

```
osDelay(100); //延时100个心跳周期
```



中国移动  
China Mobile



## 4.6. 软件定时器

本节主要以代码示例对软件定时器进行介绍。

### 创建定时器

```
void test_timer_callback(void *param)
{
    do_something();
}

osTimerId_t g_test_timer;

osTimerAttr_t timer_attr = { .name = "demo_timer" };
g_test_timer = osTimerNew(test_timer_callback, osTimerPeriodic, NULL, &timer_attr);
```

 **Important:** ML307A定时器回调中禁止打印日志或做串口输出等耗时操作。

### 启动定时器

```
osTimerStart(g_test_timer, 5000); //启动定时器，定时5000个心跳周期。
```

### 检查定时器是否正在运行

```
uint32_t is_running = osTimerIsRunning(g_test_timer);
```

### 停止定时器

```
osTimerStop(g_test_timer);
```

### 删除定时器

```
osTimerDelete(g_test_timer);
```

## 5. 编程设计注意

应用操作系统时应注意以下事项：

- 线程内应避免长时间死循环，否则可能造成系统实时性下降，影响模组正常功能。
- 定时器回调函数不可阻塞，否则会引起功能紊乱。



中国移动  
China Mobile