匠牛社区
jiang-niu.com

DSP 基础开发教程
V1.0

修定历史记录：

日期	版本	说明	作者
2018.10.25	Version<1.0>	文件创建 添加 SYS/BIOS 介绍章节 添加 Task 章节 添加 Semaphore 章节 添加 Clock 章节 添加 Mailboxs 章节 添加安装 RTOS_SDK 章节 添加 MessageQ 实例章节 添加 MathLib 实例章节	Wang

Note：任何修改操作请在上述文档中备注说明。

目录

一 快速入门

1.1 SYS/BIOS 介绍

SYS/BIOS 是一个可扩展的实时的操作系统。具有非常快速的响应时间（在中断和任务切换时达到较短的延迟），响应时间的确定性，强壮的抢占系统，优化的内存分配和堆栈管理（尽量少的消耗和碎片）。能够实现系统的模块化并可裁剪。

SYS / BIOS 不需要预付或运行时许可证费用。

SYS / BIOS 提供以下好处：

1 可以静态或动态配置所有 SYS / BIOS 对象。

2 线程模型为各种情况提供线程类型。 支持硬件中断，软件中断，任务，空闲功能和定期功能。您可以通过选择线程类型来控制线程的优先级和阻塞特性。

3 提供了支持线程之间的通信和同步的结构。这些包括 Semaphores、Mailboxes、Events、Gates。

4 为了提高性能，在主机上格式化了检测数据（例如日志和跟踪）。

1.2 Tasks

1 Task 介绍

SYS / BIOS 任务对象是由 Task 模块管理的线程， Task 具有高优先级高于低优先级，Task 优先级低于硬中断和软中断。

Task 模块根据任务的优先级和任务的当前执行状态动态调度和抢占任务，这可以确保给处理器提供的是最高优先级线程。 Task 最多可以有 32 个优先级，默认级别为 16，最低优先级为 0 用于保留运行空闲任务。

2 Task 使用

```
Task_Params taskParams;
```

```
Task_Handle task0;
Error_Block eb;
Error_init(&eb);
Task_Params_init(&taskParams);
taskParams.stackSize = 512;
taskParams.priority = 15;
task0 = Task_create((Task_FuncPtr)hiPriTask, &taskParams, &eb);
If ( task0 == NULL )
    System_abort("Task create failed");
```

1.3 Semaphores

1 Semaphores 介绍

SYS / BIOS 提供了一组基本的功能，用于基于信号量的任务间同步和通信。信号量通常用于协调对一组竞争任务中的共享资源的访问， Semaphore 模块提供了操作通过 Semaphore_Handle 类型的句柄访问的信号量对象的函数。

信号量对象可以声明为计数或二进制信号量，也可以声明为简单（FIFO）或优先级信号量，信号量可用于任务同步和互斥。

2 Semaphores 使用

```
Semaphore_Params SemParams;
Semaphore_Handle SemHandle;
Semaphore_Params_init(&SemParams);
SemParams.mode = Semaphore_Mode_COUNTING_PRIORITY;
SemHandle = Semaphore_create(1, &SemParams, NULL);
Semaphore_post(SemHandle);
```

```
Semaphore_pend(SemHandle, BIOS_WAIT_FOREVER);
```

1.4 Clock

1 Clock 介绍

所谓定时器本质上递减计数器，当计数器减到零时可以触发某种动作的执行。这种动作可以通过回调函数来实现，当定时器计时完成后，自定义的回调函数会立即被调用，需要注意的是一定要避免在回调函数中使用阻塞调用（例如调用任何可以阻塞或删除定时器任务的函数）。

定时器分为单次定时器和周期性定器。

2 Clock 使用

1 定义定时器时间到期处理函数

```
Void myClockFuncPtr(UArg arg)
{
}
```

2 初始化定时器参数以及设置

```
Clock_Params clockParams;
Clock_Handle myClock;
Clock_Params_init(&clockParams);
clockParams.period = Clock_tickPeriod;    //1ms
clockParams.startFlag = TRUE;
clockParams.arg    = (UArg)0x5555;
```

3 创建定时器以及启动定时器

```
myClock = Clock_create(myClockFuncPtr, 1000, &clockParams, &eb);
Clock_start(myClock);
```

1.5 Mailboxes

1 Mailboxes 介绍

Mailboxes 可以用在同一处理器上将缓冲区从一个任务传递到另一个任务。

2 MailBoxes 使用

Mailbox_create()和 Mailbox_delete()分别用于创建和删除邮箱。

```
Mailbox_Handle Mailbox_create(SizeT          bufsize,
                              UInt           numBufs,
                              Mailbox_Params *params,
                              Error_Block    *eb)
Void Mailbox_delete(Mailbox_Handle *handle);
```

Mailbox_pend()用于从邮箱中读取缓冲区，如果没有可用的缓冲区（即邮箱为空）则 Mailbox_pend()会阻塞。 timeout 参数允许任务等待超时，无限期等待 BIOS_WAIT_FOREVER 或完全不等待 BIOS_NO_WAIT。 时间单位是系统时钟滴答。

```
Bool    MailBox_pend(Mailbox_Handle  handle,
                     Ptr             buf,
                     UInt            timeout);
```

Mailbox_post()用于将缓冲区发布到邮箱。 如果没有可用的缓冲区插入（即邮箱已满）,则 Mailbox_post()会阻塞。 timeout 参数允许任务等待超时，无限期等待 BIOS_WAIT_FOREVER 或完全不等待 BIOS_NO_WAIT。

```
Bool    Mailbox_post(Mailbox_Handle      handle,
                     Ptr                 buf,
                     UInt                timeout);
```

二 RTOS_SDK 详解

2.1 RTOS SDK 安装

TI 提供了 Windows 和 Linux 版本的 RTOS SDK 开发包， 本文主要介绍如何配置 Linux 下的 RTOS SDK 开发环境。

1 从 ti 官网下载 AM5708 RTOS_SDK_4.3 软件包

http://software-dl.ti.com/processor-sdk-rtos/esd/AM57X/04_03_00_05/index_FDS.html

PROCESSOR-SDK-RTOS-AM57X Product downloads

Title	Description	Size
AM57xx RTOS SDK Essentials		
ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Windows-x86-Install.exe	AM57xx RTOS SDK installer for Windows Host	1399480K
ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86-Install.bin	AM57xx RTOS SDK installer for Linux Host	1436056K
ccs_setup_7.4.0.00015.exe	Code Composer Studio IDE for Windows Host	
CCS7.4.0.00015_web_linux-x64.tar.gz	Code Composer Studio IDE for Linux Host	
AM57xx RTOS SDK Optional Addons		
Download Pin Mux Tool	AM57xx Pin Mux Configuration Utility	
Download Clock Tree Tool	AM57xx Clock Tree Configuration Utility	
Related Software	Additional software elements not included in SDK	

注意：

请先参考《IndustriPi SDK 安装教程 V1.0》，安装 SDK_4.3 软件包。

2 安装 RTOS SDK_4.3 软件包

将 RTOS SDK 软件包拷贝到 ubuntu 家目录，运行如下命令安装

./ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86-Install.bin

3 下载 JN-IndustriPi RTOS 补丁包

http://www.jiang-niu.com/download.html

4 Ubuntu PC 运行如下命令，安装 JN-IndustriPi 补丁包

tar -xvf JN-IndustriPi_rtos_patch_4.3.tar.gz

cp JN-IndustriPi_rtos_patch_4.3 ~/ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86

2.2 RTOS SDK 组件介绍

RTOS SDK 组件如下图 2-2 所示：

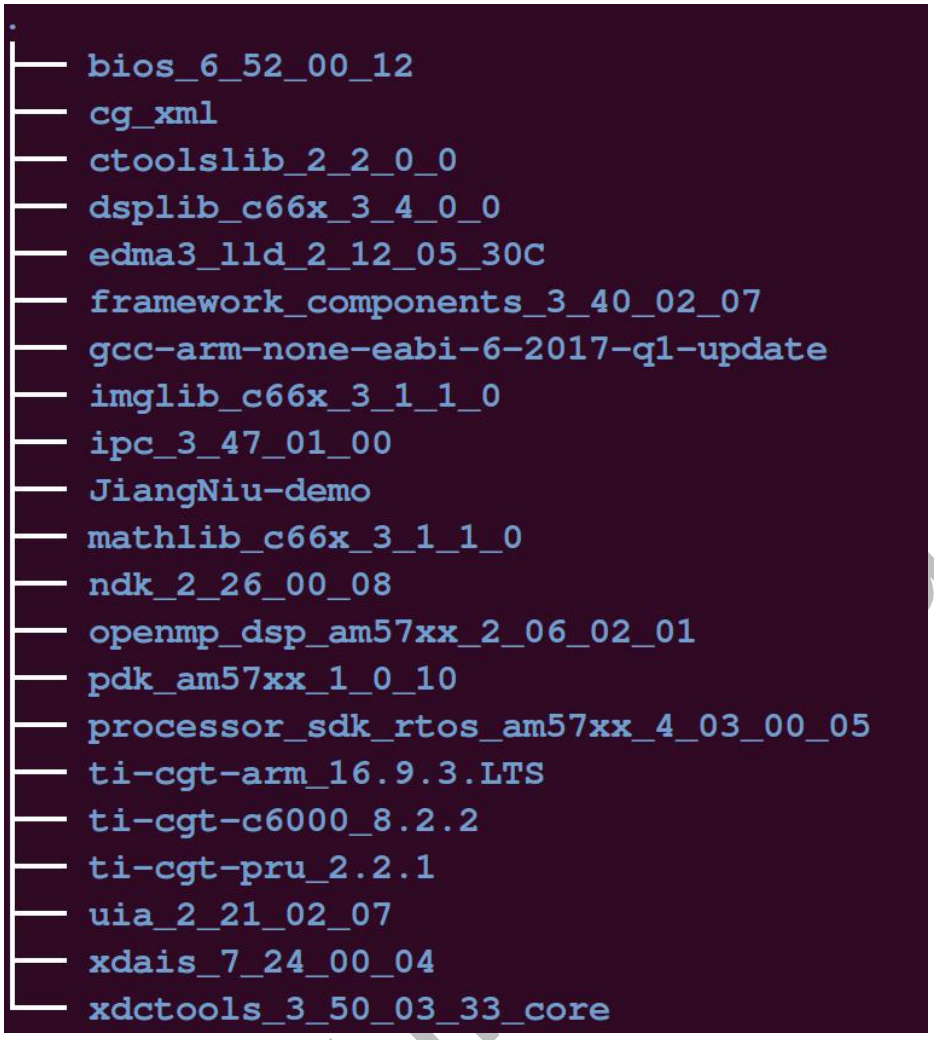

图 2-2

xdctools_3_50_03_33_core：TI DSP 编译工具

ti-cgt-c6000_8.2.2：TI DSP C66x 基本库

bios_6_52_00_12： TI SYS/BIOS 库

processor_sdk_rtos_am57xx_4_03_00_05： TI RTOS 库

JiangNiu-demo：匠牛社区官方实例

库名目录/docs：TI 官网提供该库帮助文档

库名目录/examples： TI 官网提供该库使用实例

三 实例详解

1 MessageQ 实例

实例源码位于~/ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86/JiangNiu-demo/JN-IndustriPi/MessageQ/

1.1.1 实例执行步骤

1 在 Ubuntu 运行如下命令，编译 MessageQ 实例。

```
cd ~/ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86/JiangNiu-demo/JN-IndustriPi/MessageQ/
make
```

需要拷贝 DSP 固件、ARM 程序到 IndustriPi 板卡

```
cp dsp1/bin/release/server_dsp1.xe66 /media/linux/sdb1/.
cp host/bin/release/app_host /media/linux/sdb1/.
```

2 启动 IndustriPi 板卡

```
root@am57xx-evm:~# cp /run/media/sdb1/app_host ~/.
root@am57xx-evm:~# cp /run/media/sdb1/server_dsp1.xe66 /lib/firmware/dra7-dsp1-fw.xe66
root@am57xx-evm:~# sync
```

重启开发板。

1.1.2 实例结果

在终端输入一串字符，经过 DSP 倒序输出该字符串。

运行结果如下图 3-1-1 所示：

```
root@am57xx-evm:~# ./app_host
--> main:
[  873.341427] omap-iommu 41501000.mmu: 41501000.mmu: version 3.0
[  873.348568] omap-iommu 41502000.mmu: 41502000.mmu: version 3.0
--> Main_main:
Please enter a string of characters,enter "exit" to close the program.
123456789
9 8 7 6 5 4 3 2 1
Please enter a string of characters,enter "exit" to close the program.
exit
root@am57xx-evm:~#
```

图 3-1-1

2 MathLib 实例

实例源码位于~/ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86/JiangNiu-demo/JN-IndustriPi/MathDemo/

2.1.1 实例执行步骤

1 在 Ubuntu 运行如下命令，编译 MathDemo 实例。

cd ~/ti-processor-sdk-rtos-am57xx-evm-04.03.00.05-Linux-x86/JiangNiu-demo/JN-IndustriPi/MathDemo/

make

需要拷贝 DSP 固件、ARM 程序到 IndustriPi 板卡

cp dsp1/bin/release/server_dsp1.xe66 /media/linux/sdb1/.

cp host/bin/release/app_host /media/linux/sdb1/.

2 启动 IndustriPi 板卡

root@am57xx-evm:~# cp /run/media/sdb1/app_host ~/.

root@am57xx-evm:~# cp /run/media/sdb1/server_dsp1.xe66 /lib/firmware/dra7-dsp1-fw.xe66

root@am57xx-evm:~# sync

重启开发板。

2.1.2 实例运行结果

root@am57xx-evm:~# ./app_host

运行结果如下图 3-2-2 所示：

```
root@am57xx-evm:~# ./app_host
--> main:
[23283.231707] omap-iommu 41501000.mmu: 41501000.mmu: version 3.0
[23283.238919] omap-iommu 41502000.mmu: 41502000.mmu: version 3.0

12.900000 / 93.100000 = 0.138561
sqrt 12.900000 = 3.591657
cos 12.900000 = 0.944860
sin 12.900000 = 0.327474
tan 12.900000 = 0.346585
root@am57xx-evm:~#
```

图 3-2-2