

基本内核工程 for Nios II

目录

1. 创建一个新的工程.
2. 为工程添加头 RT-Thread 的内核源文件.
3. 添加 BSP 及应用程序源文件.
4. 配置 RT-Thread 的头文件搜索路径.
5. 配置 RT-Thread.
6. 编译, 调试, 及运行.

1. 创建一个新的工程

与常规创建工程一样,选择工程模板时选择空白工程即可.

2. 为工程添加 RT-Thread 的内核源文件

需要添加的文件如下:

rt-thread 目录文件列表

├──include	RT-Thread 头文件目录
rtdef.h	
rthw.h	
rtm.h	
rtthread.h	
├──libcpu	存放与 CPU 内核有关的文件的目录
├──nios	
├──nios_ii	
context_gcc.S	线程切换
interrupt.c	保存中断切换中上下文的栈地址
stack.c	栈初始化
vector.S	中断处理(暂只接管中断退出部分)
└──src	RT-Thread 内核源文件目录
clock.c	
device.c	
idle.c	
ipc.c	
irq.c	
kservice.c	
kservice.h	
mem.c	
mempool.c	
module.c	
module.h	
object.c	
rtm.c	
scheduler.c	
SConscript	
slab.c	
thread.c	
timer.c	

3. 添加 BSP 及应用程序源文件

主要添加用于启动系统的程序, 及驱动程序, 应用程序.

参考示例如下:

application.c

board.c

startup.c

uart.c

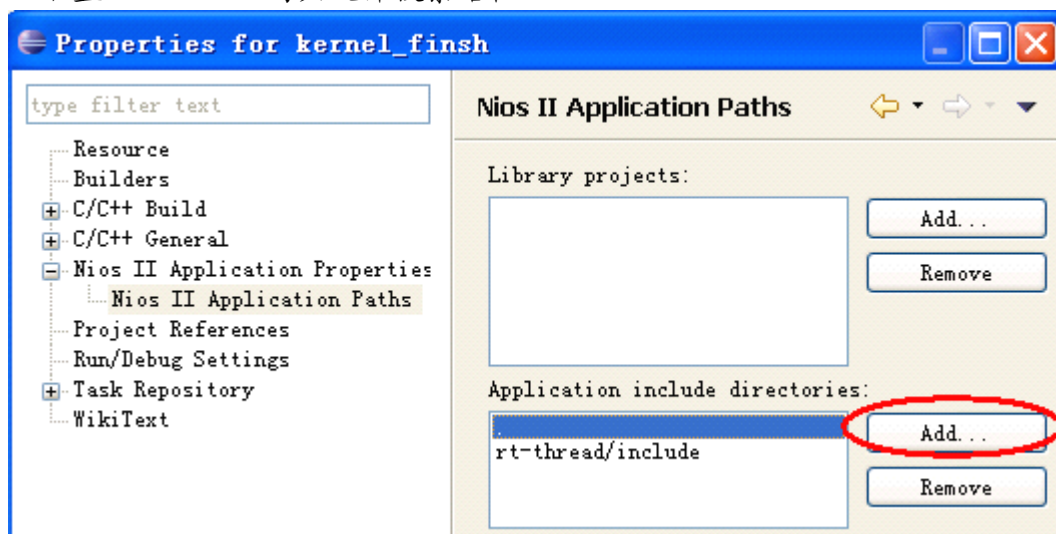
board.h

rtconfig.h

sopc.h

Uart.h

4. 配置 RT-Thread 的头文件搜索路径



注意:

1. 不同版本的 Nios II IDE 的头文件设置位置不同 (如图版本为 10.0)
2. 在添加 RT-Thread 头文件目录时, 还应该添加应用程序的头文件所在目录.

5. 配置 RT-Thread

配置 RT-Thread 主要是修改 rtconfig.h 的配置.

Nios II 主要的配置参数如下 (基本内核):

```
/* RT_ALIGN_SIZE*/
#define RT_ALIGN_SIZE 4

/* config stack for idle */
#define IDLE_THREAD_STACK_SIZE 1024

/* use newlib */
#define RT_USING_NEWLIB
```

6. 编译, 调试, 及运行

编译及运行与普通应用程序并无大的区别.

不过注意: Nios II IDE 9.1 版本请把汇编文件名由大写的 .S 改为小写的 .s
系统启动后, 在串口输出如下:

```
----- board info -----  
ALT_DEVICE_FAMILY:    CYCLONEII  
ALT_CPU_ARCHITECTURE: altera-nios2  
ALT_CPU_CPU_FREQ:    100000000  
memory size:         at 0x01000000 0x00800000 byte
```

```
\ | /  
- RT -   Thread Operating System  
/ | \ 0.4.0 build Mar  4 2011  
2006 - 2011 Copyright by rt-thread team  
led1 on, count : 0  
led2 on, count : 0  
led1 off  
led1 on, count : 1  
led2 off  
led1 off  
led1 on, count : 2  
led2 on, count : 1  
led1 off  
led1 on, count : 3
```