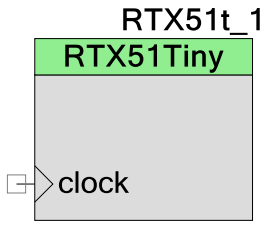
Keil RTX51 Tiny Real-Time Operating System (RTOS)

1.0

Features



- Cooperative and Round Robin multitasking supported
- Total of 16 tasks supported
- Signals are supported
- Configurable Round Robin timeout
- Stack error check supported
- Configurable free stack limit
- Long user interrupts support

General Description

RTX51 Tiny is a real-time operating system (RTOS) which allows users to create applications that simultaneously perform multiple functions or tasks. RTX51 Tiny component can be configured to use Cooperative or Round Robin scheduling with support to modify the Round Robin timeout value.

When to use a RTX51 Tiny

RTX51 Tiny component can be used in a PSoC 3 application which requires simultaneous execution of several tasks performing different jobs. For example it can be used in a system where monitoring of input from various sources is needed in conjunction with displaying data on an LCD without missing any possible input.

Input/Output Connections

This section describes the various input and output connections for <Component>. An asterisk (*) in the list of I/Os indicates that the I/O may be hidden on the symbol under the conditions listed in the description of that I/O.

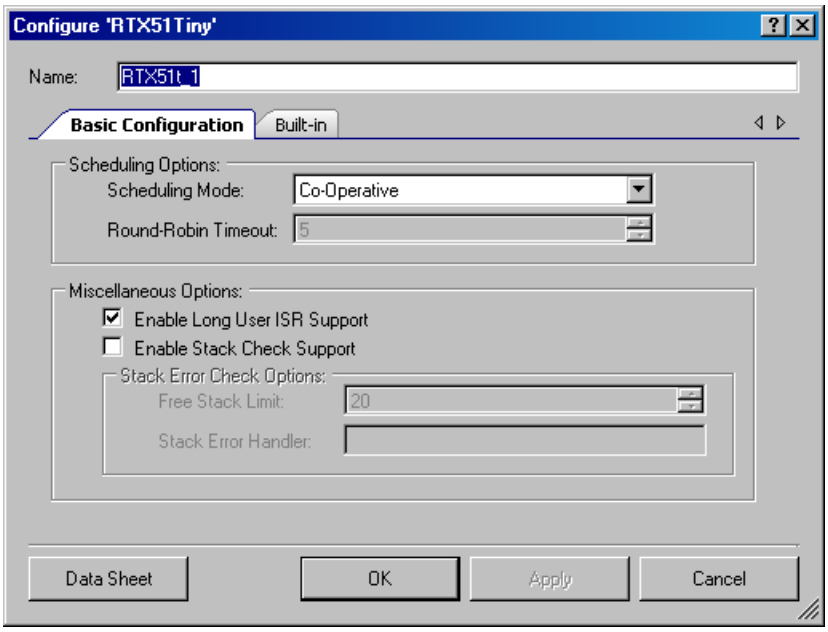
clock – Input

The clock input defines the rate at which the periodic interrupt is generated for the RTX51 Tiny RTOS. This interrupt is the RTX51 Tiny timer tick. Timeout and interval values specified for the

RTX51 Tiny RTOS routines are measured using the RTX51 Tiny Timer Tick. This input is always visible and must be connected.

Component Parameters

Drag an RTX51 Tiny RTOS component onto your design and double-click it to open the Configure dialog.



The RTX51 Tiny component provides the following parameters.

Scheduling Options

Scheduling Mode

Select the scheduling mode required. Round Robin or Cooperative scheduling option can be selected.

Round Robin Timeout

Specifies the number of RTX51 Tiny timer ticks each task runs before a RoundRobin task switch. This option is enabled when scheduling mode is set to RoundRobin.

Miscellaneous Options

Enable Long User ISR Support

This option specifies whether the application has interrupts (other than the RTX51 Tiny Timer Tick Interrupt) that execute longer than a timer tick interval. When this configuration option is

checked, RTX51 Tiny includes code to protect the RTX51 Tiny Timer Tick Interrupt from reentrancy.

Enable Stack Check Support

Enable the stack overflow checking in RTX51 Tiny RTOS. When this option is checked, Free Stack and Stack Error Handler option are enabled.

Stack Error Check Options

- Free Stack Limit - This option defines the minimum number of bytes available on the stack. When switching to a task, if RTX51 Tiny detects fewer than the value specified, the function specified by “Stack Error Handler” is executes. The default setting is 20 bytes.
- Stack Error Handler - This option specifies the instructions to execute in the event of a stack error (fewer than “Free Stack” bytes available). The default code disables interrupts and enters an endless loop.

Placement

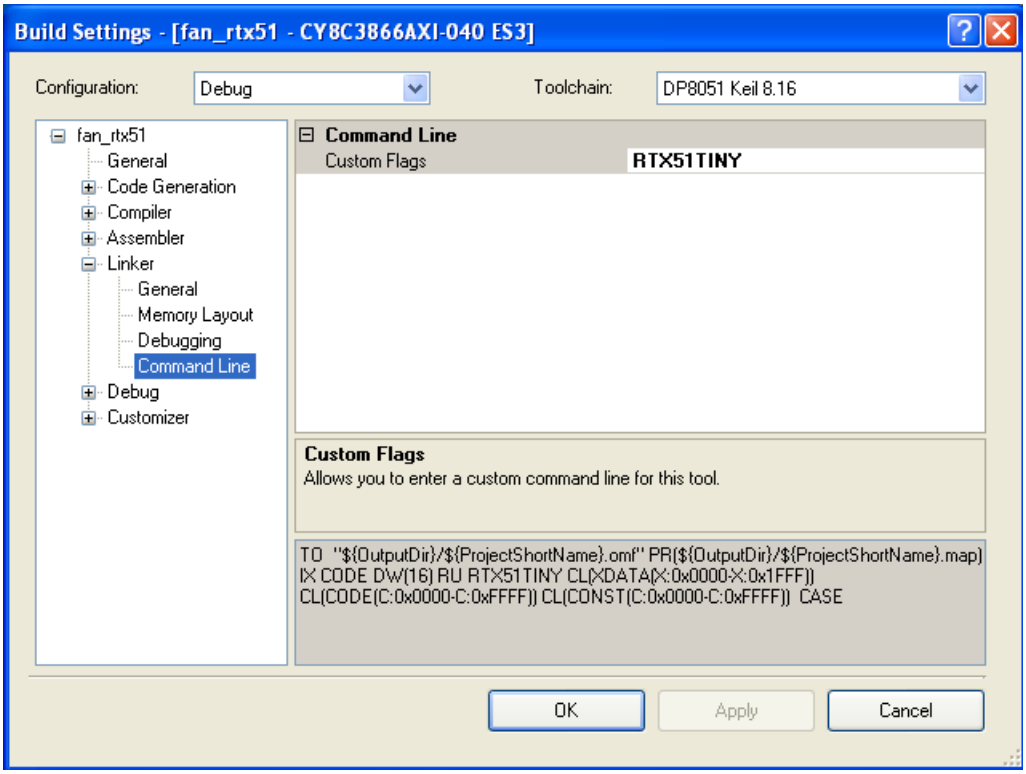
RTX51Tiny component is based on the interrupt component and it consumes one entry in the devices interrupt vector.

Resources

None

Build Settings

Include the linker directive RTX51TINY in the linker command line. This directive specifies that the program being linked is a real-time application that uses the RTX51 Tiny Real-Time Operating system. RTX51 Tiny libraries are included in the linkage and linker support for task functions is enabled.



Application Programming Interface

Application Programming Interface (API) routines allow you to configure the component using software. The following table lists and describes the interface to each function. The subsequent sections cover each function in more detail.

By default, PSoC Creator assigns the instance name RTX51t_1" to the first instance of a component in a given design. You can rename it to any unique value that follows the syntactic rules for identifiers. The instance name becomes the prefix of every global function name, variable, and constant symbol. For readability, the instance name used in the following table is "RTX51".

Function	Description
RTX51_Start	Initialize the module.
RTX51_Stop	Disable the module.

void RTX51_Start(void)

Description: This function initialize the RTX51 Tiny as follows:

- Enable the clock.
- Disable the interrupts.
- Change the ISR vector for the Interrupt to point to RTX51 Tiny RTOS timer ISR.
- Set the priority of interrupt.
- Enable the interrupt.

Parameters: None

Return Value: None

Side Effects: None

void RTX51_Stop(void)

Description: This function disables the interrupt

Parameters: None

Return Value: None

Side Effects: None

Sample Firmware Source Code

The following is a C language example demonstrating the basic functionality of the RTX51 Tiny component. The name of the component has been changed to RTX51 after placement in the design.

Note If you rename your component you must also edit the example code as appropriate to match the component name you specify.

```
#include <device.h>
#include <rtx51tny.h>

uint8 counter0 = 0;
uint8 counter1 = 0;
uint8 counter2 = 0;

/*****************************************************************************/
/* Task 0 'job0': RTX51 tiny starts execution with task 0                    */
/*****************************************************************************/
job0 () _task_ 0
{
    /** Enable the RTX51Tiny */
    RTX51_Start();

    os_create_task (1);      /* start task 1 */
    os_create_task (2);      /* start task 2 */
```

```
    while (1)
    {
        counter0++;
    }
}

/*****************************************************************************/
/* Task 1 'job1': RTX51 tiny starts this task with os_create_task (1)        */
/*****************************************************************************/
job1 () _task_ 1
{
    while (1)
    {
        counter1++;
    }
}

/*****************************************************************************/
/* Task 2 'job2': RTX51 tiny starts this task with os_create_task (2)        */
/*****************************************************************************/
job2 () _task_ 2
{
    while (1)
    {
        counter2++;
    }
}
```

Interrupt Service Routine

Not applicable.

Functional Description

RTX51 Tiny is a real-time operating system (RTOS) which allows you to create applications that simultaneously perform multiple functions or tasks. This is often required in an embedded application. While it is certainly possible to create realtime programs without an RTOS (by executing one or more functions or tasks in a loop), there are numerous scheduling, maintenance, and timing issues that an RTOS like RTX51 Tiny can solve for you.

A realtime operating system (RTOS) allows flexible scheduling of system resources, like the CPU and memory, and offers communication between tasks.