/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Release Notes for RTX51 TINY Version 2 \*/

/\* PK51 PROFESSIONAL DEVELOPERS KIT \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

This file contains release notes and last minute changes relating to the

RTX51 TINY Real Time Operating System Version 2.

Information in this file, the accompanying manual, and software is

Copyright (c) 2002 Keil Software and Keil Elektronik. All rights reserved.

Contents

========

1. What's New in RTX51 TINY Version 2

1.1. Code Banking Support

1.2. Explicit Task Switch (Function: os\_switch\_task)

1.3. RUN Status Flag for Tasks

1.4. CPU IDLE Mode for Power-Reduction

1.5. Add User Code to the 8051 CPU Hardware Timer Routine

1.6. Handling for Interval Events (Function: os\_reset\_interval)

2. Reduced Code/Data Size and Improved Performance

3. Problems solved in RTX51 TINY Version 2.02

4. Example Programs

1. What's New in RTX51 TINY Version 2

RTX51 TINY Version 2 has been completly restructured to gain flexiblity,

accelarate performance, and reduce code/data space requirements. Also

several new features are added to this popular Real-Time Kernal.

The following section describes the new features in RTX51 TINY Version 2:

1.1. Code Banking Support

You can now use RTX51 TINY together with code banking application. Code

banking support is enabled in the file CONF\_TNY.A51 with the CODE\_BANKING

configuration variable. As with all other code banking applications you

need to configure the file L51\_BANK.A51. Note: The banking configuration

file L51\_BANK.A51 V2.12 or higher is required.

1.2. Explicit Task Switch (Function: os\_switch\_task)

The function os\_switch\_task lets a task give up the CPU and let another

task execute. If the task calling os\_switch\_task is the only task that is

ready for execution, it will resume running immediately.

Function prototype:

void os\_switch\_task (void);

Program Example:

void long\_job (void) \_task\_ 1 {

float f1, f2;

f1 = 0;

while (1) {

f2 = log (f1);

f1 += 0.0001;

os\_switch\_task (); // execute other tasks

}

}

1.3. READY Status Flag

In addition to the signal flag, each task has also a READY status flag.

A new function allows you to set the READY status flag. If the READY

status flag of a tasks is set, the task will always stay in the Ready

state. The os\_wait function will immidiatly continue to the execute

when the READY status flag is set. When a task becomes active, the status

of the READY status flag is cleared and the previous status is signaled as

retrun value of the os\_wait and os\_switch\_task function.

The os\_wait function requires no specific argument (like K\_SIG for the

signal flag) to wait for the READY status flag.

Compared to the os\_send\_signal function, the os\_set\_ready function executes

faster. os\_set\_ready does not check for valid taskid arguments.

Function Prototypes

void os\_set\_ready (unsigned char taskid);

void isr\_set\_ready (unsigned char taskid);

Set the READY status flag of the task specified by taskid.

The value of the READY status flag is indicated by the return value RDY\_EVENT of the

os\_wait and os\_switch\_task function. When a task becomes ready, the READY status flag

of these task is automatically cleared.

1.4. CPU IDLE Mode for Power-Reduction

Many 8051 variants can be switched into an IDLE mode where the CPU is gated

off the clock signal. In IDLE mode, the CPU stops instruction execution.

However peripherals including the interrupt system still continue their

operation. RTX51 Tiny invokes the IDLE mode when no task is in the status

READY. Typcially the IDLE mode is terminated by an CPU interrupt, i.e. the

RTX51 Tiny System Clock Interrupt.

On most 8051 variants the IDLE mode is started by setting bit 0 (IDL) in

the PCON register. However, the configuration file CONF\_TNY.A51 provides

the macro CPU\_IDLE for the configuration of the IDLE mode activation code.

1.5. Add User Code to the 8051 CPU Hardware Timer Routine

In RTX51 Tiny Version 2 it is possible to add user code for the RTX51

System Timer Interrupt routine. In this way you may use the fixed clock

rate of the RTX51 System Timer Interrupt to execute own routines. The user

code can be configured with macro HW\_TIMER\_CODE in the configuration file

CONF\_TNY.A51.

1.6. Handling for Interval Events (Function os\_reset\_interval)

RTX51 Tiny maintains a software timer for each task. This software timer is

used as interval timer when os\_wait is called with a K\_IVL event argument.

The interval time specified is substracted from the software timer on the

entry of the os\_wait function.

When two event arguments are used (signal and interval might be combined with

K\_SIG | K\_IVL) with os\_wait and the signal event occurs, the software timer might

not be elapsed and this may result in incorrect interval times on further os\_wait

calls with the K\_IVL event argument. The function os\_reset\_interval overcomes this

problem and resets the interval time.

Example:

void maintask (void) \_task\_ 0 {

os\_create\_task (1);

while (1) {

event = os\_wait2 (K\_SIG | K\_IVL, 100);

switch (event) {

case TMO\_EVENT:

a = event;

break;

case SIG\_EVENT:

b = event;

os\_reset\_inverval (100); // correct interval time on a signal event

break;

}

}

}

2. Reduced Code/Data Size and Improved Performance

RTX51 Tiny Version 2 is fully scalable and shows significant code size

reductions when:

a) Round Robin Task Switching is disabled with the setting TIMESHARING

EQU 0 in the configuation file CONF\_TNY.A51.

b) not all RTX51 Tiny system functions are used.

c) Stack Checking is disabled with the setting FREE\_STACK EQU 0 in the

configuration file CONF\_TNY.A51.

Compared to RTX51 Tiny Version 1 the code size requirements are reduced.

However also the overall system performance is improved. When you disable

Round Robin Task Switching even the data requirements are reduced.

3. Problems solved in RTX51 TINY Version 2.02

The following known problems of RTX51 Tiny are corrected in Version 2.02:

a) RTX51 Tiny V1.06: isr\_send\_signal function may destroy the READY status

when an interrupt occurs during os\_wait, whereby the current task is

waiting for the signal from this interrupt.

b) RTX51 Tiny V1.06: K\_IVL and K\_SIG event cannot be combined in os\_wait

since it is not possible to correct the interval timer. In RTX51 Tiny

V2.0 the function os\_reset\_interval allows you to correct the interval

timer.

c) RTX51 Tiny V1.06: TIMESHARING (round-robin time) cannot be set to 1.

In in this case it can happen (when the interrupt appears before the

robin-time is set) that the timeout is delayed 256 system clock ticks.

d) RTX51 Tiny V1.06: When user interrupts are longer than a RTX51 Tiny

System Clock Timer Tick, the RTX51 Tiny System Clock Timer is called

recursively. This overwrites the values SAVEPSW and SAVEACC with wrong

values and causes typically a system crash.

In RTX51 Tiny V2.0 it is recommended that you set LONG\_USR\_INTR EQU 1

in the configuration file CONF\_TNY.A51 if your application contains an

interrupt function that may take longer for execution than the RTX51

Tiny System Clock Timer Interval time. It might be also required to

set this flag when your code executes most of the time interrupt code

with high interrupt priority.

e) RTX51 Tiny V2.00 BETA: os\_wait did not reset the SIGNAL event, when

the signal was already set before the os\_wait call.

f) RTX51 Tiny V2.00 BETA: os\_wait with K\_IVL argument did not work.

g) RTX51 Tiny V2.01: os\_wait returned wrong event codes when K\_SIG and K\_TMO

was used in different combinations within the same task.

h) RTX51 Tiny V2.01: K\_IVL may caused 256 additional timer tick delays if the

interval time is lower than the timer underflow value of that task.

4. Example Programs

The following example programs are provided with RTX51 Tiny Version2.

Examples\Ex1 Demonstrates Round-Robin Multitasking.

Examples\Ex2 Shows how to use time delays and signals with the

os\_wait function.

Examples\Traffic A time-controlled Traffic Light Controller.

Examples\Banking Uses code banking a shows several new features of

RTX Tiny Version 2.