

## Exam TI2720-C Embedded Software

Friday April 19 2013 (14.00 - 17.00)

In order to avoid misunderstanding on the syntactical correctness of code fragments in this examination, we will always assume that we are dealing with pseudo-code, although we might have syntactically correct code in some cases. We assume that the required variables, semaphores, tasks, timers, etc. are always declared and initialized correctly.

Further, we assume the following abbreviations to be known:

- RR = Round Robin,
- RRI = Round Robin with Interrupts,
- FQS = Function Queue Scheduling,
- RTOS = Real-Time Operating System,
- IR = interrupt and
- ISR = interrupt service routing.

In this exam, we use the following definitions, unless stated otherwise:

```
void delay(int ms) {
    !! do some CPU computation to the amount of ms milliseconds
}
```

```
void putchar(char c) {
    while (!! UART tx buffer not empty);
    !! send c to UART tx buffer
}
```

```
void puts(char *s) {
    !! print string s using putchar
}
```

To pass this written exam, you need to answer correctly at least 18 questions. The relationship number of correctly answered questions and your mark is given below.

Nr. Correct Answers	18	19	20	21	22	23	24	25	26	27	28	29	30
Mark	6	6.5	6.5	7	7.5	7.5	8	8.5	8.5	9	9.5	10	10

<p><b>Question 1</b></p> <p>Which of the following statements is correct? An interrupt vector ...</p> <ul style="list-style-type: none"> <li>a. contains the address of an interrupt</li> <li><b>b. contains the address of an ISR</b></li> <li>c. points to a table with interrupt routines</li> <li>d. points to a table with interrupts</li> </ul>
<p><b>Question 2</b></p> <p>Which of the following statements is correct? Using interrupts improves ...</p> <ul style="list-style-type: none"> <li>a. task response time</li> <li>b. memory response time</li> <li><b>c. system response time</b></li> <li>d. processor response time</li> </ul>
<p><b>Question 3</b></p> <p>Which of the following statements is correct? An interrupt service routine is supposed to ...</p> <ul style="list-style-type: none"> <li>a. restore the lowest priority interrupt</li> <li><b>b. respond quickly</b></li> <li>c. decrease its priority</li> <li>d. disable the non-maskable interrupt</li> </ul>
<p><b>Question 4</b></p> <p>How can a low priority task prevent itself from being interrupted by a high priority task?</p> <ul style="list-style-type: none"> <li>a. By enabling interrupts</li> <li><b>b. By disabling the interrupts</b></li> <li>c. By using semaphores</li> <li>d. By avoiding critical sections</li> </ul>
<p><b>Question 5</b></p> <p>What happens when a hardware interrupt occurs in the middle of an instruction execution?</p> <ul style="list-style-type: none"> <li>a. the execution stops and the interrupt is handled at once</li> <li><b>b. the instruction is executed until completion and the interrupt is handled next</b></li> <li>c. the operating system checks the priority to determine whether or not the interrupt should be serviced</li> <li>d. an exception is generated and execution of the program halts</li> </ul>
<p><b>Question 6</b></p> <p>The primary shortcoming for RRI architecture is:</p> <ul style="list-style-type: none"> <li><b>a. all the tasks have the same priority</b></li> <li>b. it is more complex than RR</li> <li>c. critical sections must be used</li> <li>d. the memory footprint increases</li> </ul>
<p><b>Question 7</b></p> <p>Which architecture has interrupt routines that add function pointers to a queue for the main function to call?</p> <ul style="list-style-type: none"> <li>a. RRI</li> <li><b>b. FQS</b></li> <li>c. RR</li> <li>d. RTOS</li> </ul>
<p><b>Question 8</b></p> <p>Which architecture has the best response time for task code?</p> <ul style="list-style-type: none"> <li>a. RRI</li> <li>b. FQS</li> <li>c. RR</li> <li><b>d. RTOS</b></li> </ul>
<p><b>Question 9</b></p> <p>For which of the following requirements is the round-robin architecture the ideal choice?</p> <ul style="list-style-type: none"> <li><b>a. robustness</b></li> <li>b. responsiveness</li> <li>c. availability</li> <li>d. usability</li> </ul>
<p><b>Question 10</b></p> <p>Which of the following statements holds for the FQS architecture?</p> <ul style="list-style-type: none"> <li>a. the code length of the lower priority tasks does not affect the execution time of other tasks</li> <li>b. the code length of the higher priority tasks does not affect the execution time of other tasks</li> <li><b>c. to reduce the response time of higher priority tasks, the code for lower priority tasks should be small</b></li> <li>d. to reduce the response time of higher priority tasks, the code for lower priority tasks should be large</li> </ul>

<p>Question 11</p> <p>Which of the following statements is correct? Good RTS design means ...</p> <ul style="list-style-type: none"> <li>a. dynamic allocation of one task for every functionality</li> <li><b>b. short disabling of interrupts in case of data sharing</b></li> <li>c. deal with much of the functionality in the ISR</li> <li>d. allocation of tasks with long periods</li> </ul>
<p>Question 12</p> <p>Who handles the signaling between the interrupt routines and the task code in RTOS?</p> <ul style="list-style-type: none"> <li>a. the user</li> <li><b>b. the RTOS</b></li> <li>c. the ISR</li> <li>d. it is done automatically</li> </ul>
<p>Question 13</p> <p>Which factor determines the best response time of an RTOS reacting on an interrupt? The shortest period of time ...</p> <ul style="list-style-type: none"> <li>a. required for executing lower-priority interrupt routines</li> <li><b>b. required for saving all data of the current task</b></li> <li>c. that the interrupt is enabled</li> <li>d. required for another high-priority task to finish execution</li> </ul>
<p>Question 14</p> <p>In an RTOS, tasks can be in state BLOCKED, READY or RUNNING. Which of the following statements is true?</p> <ul style="list-style-type: none"> <li>a. A task can transition directly from BLOCKED to RUNNING</li> <li>b. A task can transition directly from READY to BLOCKED</li> <li><b>c. A task can transition directly from READY to RUNNING</b></li> <li>d. When in state BLOCKED it stays there forever</li> </ul>
<p>Question 15</p> <p>Which of the following statements is correct? A reentrant function</p> <ul style="list-style-type: none"> <li><b>a. may only call other reentrant functions</b></li> <li>b. can create a shared data problem</li> <li>c. may not be called by different tasks</li> <li>d. must use hardware in a non-atomic way</li> </ul>
<p>Question 16</p> <p>Which of the following statements is <b>FALSE</b>:</p> <ul style="list-style-type: none"> <li>a. semaphores can be used for solving shared data problems</li> <li>b. interrupts can be disabled in order to protect a critical section</li> <li>c. semaphores can be used for signaling between tasks</li> <li><b>d. interrupts can be used for signaling between tasks</b></li> </ul>
<p>Question 17</p> <p>A "heartbeat timer" refers to:</p> <ul style="list-style-type: none"> <li>a. a task-related function provided by an RTOS</li> <li>b. a user-defined function for keeping track of application time</li> <li><b>c. a single hardware timer an RTOS is using as base for all timings</b></li> <li>d. a function that resets a device when detecting a deadlock</li> </ul>
<p>Question 18</p> <p>Tasks in an RTOS are often structured as:</p> <ul style="list-style-type: none"> <li>a. state machines with states stored in global variables and messages in their queues don't have any role</li> <li><b>b. state machines with states stored in private variables and messages in their queues acting as events</b></li> <li>c. RR architectures with messages triggering state changes</li> <li>d. RRI architectures with interrupts advancing the state machine</li> </ul>
<p>Question 19</p> <p>Which of the following statements is correct? The advantage of more tasks is that...</p> <ul style="list-style-type: none"> <li>a. it leads to more coupling</li> <li><b>b. it leads to better encapsulation</b></li> <li>c. it leads to higher system throughput</li> <li>d. it leads to bigger system footprint</li> </ul>
<p>Question 20</p> <p>Why should time-slicing not be used in an RTOS?</p> <ul style="list-style-type: none"> <li>a. Equally important tasks require equal processor attention</li> <li>b. It creates too many data sharing problems</li> <li><b>c. Alternatively, all tasks could become one task</b></li> <li>d. It should be used; time slicing is essential in an RTOS</li> </ul>

<p><b>Question 21</b></p> <p>Which of the following statements is correct? Code memory can be saved through...</p> <ul style="list-style-type: none"> <li>a. packing/compression of data structures</li> <li>b. using local variables</li> <li>c. using function pointers</li> <li>d. <b>the static keyword</b></li> </ul>
--

<p><b>Question 22 - 23</b></p> <p>Given is the following RTOS (pseudo) code.</p> <p>...</p> <pre> void T1(void) {     while (1) {         OSSemPend(sem1); // may unblock any time         f(1);     } }  void T2(void) {     while (1) {         OSSemPend(sem2); // may unblock any time         f(-1);     } }  void f(int delta) {     OSSemPend(mutex);     counter = counter + delta; // adjust some global counter     OSSemPost(mutex); } </pre>
--

<p><b>Question 22</b></p> <p>Which of the following statements is correct?</p> <ul style="list-style-type: none"> <li>a. The function f( ) is not reentrant</li> <li>b. <b>There is no shared data problem</b></li> <li>c. There is a shared data problem between T1 and T2</li> <li>d. This is a typical case for priority inversion</li> </ul>
--

<p><b>Question 23</b></p> <p>Which of the following statements is correct? For the program to work correctly mutex must be initialized with ...</p> <ul style="list-style-type: none"> <li>a. -1</li> <li>b. 0</li> <li>c. <b>1</b></li> <li>d. 2</li> </ul>
--

<p><b>Question 24</b></p> <p>Given is the following RTOS (pseudo) code.</p> <pre> void isr_buttons(void) // arrive here if a button is pressed {     x = X32_PERIPHERALS[PERIPHERAL_TEMP1];     y = X32_PERIPHERALS[PERIPHERAL_TEMP2];     ... }  ...  while (! program_done) {     temp1 = x; temp2 = y;     X32_display = ((temp1 &amp; 0xff) &lt;&lt; 8)   (temp2 &amp; 0xff);     if (temp1 != temp2) {         !! shutdown plant     } } </pre>
<p>Which of the following statements is correct?</p> <p>a. There is a potential deadlock problem in this code</p> <p><b>b. There is a potential shared data problem in this code</b></p> <p>c. There is a potential problem with respect to reentrancy in this code</p> <p>d. There is no potential problem in this code</p>

<p><b>Question 25 - 27</b></p> <p>Given is the following RTOS (pseudo) code, which reads the current values of 3 different buttons and acts accordingly. The 3 buttons are all mapped to bits 0..2 of the register buttons. The buttons are already debounced; execution time of ISR is negligible.</p> <pre> void f1(void) { delay(1000); } void f2(void) { delay(2000); } void f3(void) { delay(3000); }  isr_button1() {b1=1}; // arrive here if button 1 is pressed isr_button2() {b2=1}; // arrive here if button 2 is pressed isr_button3() {b3=1}; // arrive here if button 3 is pressed  void main (void) {     while (1) {         if (b1) { f1(); b1=0; }         if (b2) { f2(); b2=0; }         if (b3) { f3(); b3=0; }     } } </pre>
--

<p><b>Question 25</b></p> <p>None of the buttons has been pressed. Which of the following statements is correct? The longest time it takes to complete f3( ) once is:</p> <p>a. &lt; 1 second</p> <p>b. 1 second</p> <p>c. 2 seconds</p> <p><b>d. 3 seconds</b></p>
---

<p><b>Question 26</b></p> <p>b1 to b3 are in arbitrary states. The longest execution time (worst-case execution time - wcet) of f1( ) is:</p> <p><b>a. 1 second</b></p> <p>b. 3 seconds</p> <p>c. 5 seconds</p> <p>d. 6 seconds</p>
---

<p><b>Question 27</b></p> <p>Which of the following statements is correct? This source code is an example of a ...</p> <p>a. RR architecture</p> <p><b>b. RRI architecture</b></p> <p>c. FQS architecture</p> <p>d. RTOS architecture</p>
---

Question 28
<p>Given is the following RTOS (pseudo) code.                      ...</p> <pre> void isr_button(void) { // arrive here when button pressed     delay(20); // wait for debounce     // do something -- takes 10 ms }  void task(void) {     while (1) {         delay(100); // synchronize         f();        // perform function that takes 100 ms     } }                     </pre>
<p>Pressing the button during synchronization will ...</p> <ol style="list-style-type: none"> <li>extend the duration of synchronization by 10 ms</li> <li>extend the duration of the synchronization by 20 ms</li> <li><b>extend the duration of the synchronization by 30 ms</b></li> <li>not have any delaying effect</li> </ol>

Question 29 - 30
<p>Given is the following RTOS (pseudo) code. T1 has the highest priority, the time for puts and context switching is negligible:                      ...</p> <pre> void T1() {     while (1) {         puts("1 ");         OSTimeDly(10);     } }  void T2() {     while (1) {         puts("2 ");         OSTimeDly(10);     } }                     </pre>

Question 29
<p>Which of the following statements is correct? The display shows:</p> <ol style="list-style-type: none"> <li>2 1 2 1 2 1 2 1 ...</li> <li><b>1 2 1 2 1 2 1 2 ...</b></li> <li>2 2 2 2 2 2 2 ...</li> <li>1 1 1 1 1 1 1 1 ...</li> </ol>

Question 30
<p>The <i>OSTimeDly</i>(10) call is replaced by a <i>delay</i>(10) call.                      Which of the following statements is correct? The display shows:</p> <ol style="list-style-type: none"> <li>2 1 2 1 2 1 2 1 ...</li> <li>1 2 1 2 1 2 1 2 ...</li> <li>2 2 2 2 2 2 2 ...</li> <li><b>1 1 1 1 1 1 1 1 ...</b></li> </ol>