EXAM
(Tentamen)

TDDI11
Embedded Software

2016-06-03 08:00-12:00

On-call (jour):
Mikael Asplund, 013-282668

Admitted material:

- Dictionary from English to your native language

General instructions:

- The assignments are **not ordered** according to difficulty.
- You may answer in either English or Swedish.
- Read all assignments carefully and completely before you begin.
- Use a new sheet for each assignment and use only one side.
- Before you hand in, order the sheets according to assignment, number each sheet, and fill in AID-number, date, course code and exam code at the top of the page.
- Write clearly. Unreadable text will be ignored.
- Be precise in your statements.
- **Motivate** clearly all statements and reasoning.
- **Explain** calculations and solution procedures.
- If in doubt about the question, write down your interpretation and assumptions.
- Grading: U, 3, 4, 5. The preliminary grading thresholds for p points are:

$0 \leq p < 20$:	U
$20 \leq p < 27$:	3
$27 \leq p < 34$:	4
$34 \leq p \leq 40$:	5

Good Luck!

Question 1, multiple choice. (10 points)
Use the answer sheet at the end of the exam. Not motivation or explanation is required for this question.

1a)
Consider the following requirements for an embedded device:
R1: When the user presses button A, the device shall be turned on.
R2: The device shall respond to user input within 500ms.

Which of the following statements is correct?

1. R1 is an extra-functional requirement as it is critical that the device can be turned on.
2. R2 is a functional requirement as it describes how the device should function on user input.
3. R2 is an extra-functional requirement as it refers to timeliness properties of the device.

1b)
Which of the following statements are correct?
1. Embedded systems are often real-time systems.
2. A system cannot be both a real-time system and an embedded system.
3. A real-time system is by definition embedded.

1c)
What will be the output from the following C program?

```
#include <stdio.h>
int main() {
      unsigned long int a = 5;
      unsigned long int *b;
      b = &a;
      printf("%lu \n", (unsigned long int)&b);
}
```

1. The value of b
2. 5
3. The address of b

1d)
What will be the output from the following C program?

```
#include <stdio.h>
int main() {
      printf("%d \n", 1 && 2);
}
```

1. 1
2. 0
3. 3

1e)
What will be the output from the following C program?

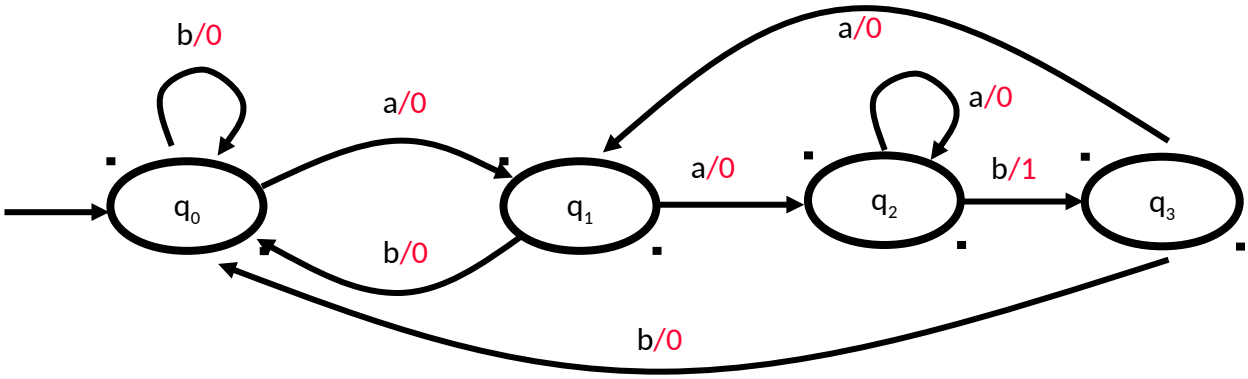
```
#include <stdio.h>
int main() {
      printf("%d \n", (1 | ~5)&13);
}
```

1. 1
2. 9
3. 13

1f)
Which of the following statements is not correct?

1. Direct memory access gives the programmer direct access to the memory.
2. Execution time of the interrupt service routine is a concern in real-time systems.
3. An *interrupt vector* is an address to an interrupt service routine.

1g)
Consider the statemachine below:



Which of the following outputs can be produced by this statemachine?

1. 00100000100
2. 00010100000
3. 01000000000

1h)
Consider a system that is composed of three components. The three components can each be described with state machines that have 4, 5 and 10 reachable states respectively. How many reachable states are there in the full system?

1. Exactly 19.
2. Somewhere between 19 and 200.
3. Exactly 200.

1i)
Context switching refers to:

1. Storing and restoring the process state (i.e., program counter, registers, etc) when switching between processes in a multitasking system.
2. Changing the context of a process by moving it from one core to another in multi-core system.
3. Moving the data and program memory of a process from one location in memory to another in order to reduce fragmentation.

1j)
Which of the following statements is correct?

1. The Earliest Deadline First (EDF) scheduling policy works by giving the highest priority to the process with the shortest time from the release of the task instance to its deadline.
2. The Rate Monotonic Scheduling (RMS) policy is optimal among all static and dynamic priority schedulers.
3. The Controller Area Network (CAN) bus protocol uses static priorities among the frames to determine access to the bus.

Question 2. (5 points)
Consider a system that periodically reads the input from an accelerometer unit. The data from the accelerometer comes in the following format: X acceleration (2 bytes), Y acceleration (2 bytes), Z acceleration (2 bytes). The data is provided in big-endian order whereas the processor uses little-endian. Provide C-like pseudocode for a program that prints out the acceleration of the three dimensions, read from a 6-byte array *buffer*.

Question 3. (5 points)
Describe the sequence of events that occur when a key is pressed and read into memory when using interrupt-driven I/O.

Question 4. (5 points)
Consider the following task set for small robot system:

Task	**Period**	**Worst-case execution time (WCET)**
Control task	50ms	30ms
Planning task	200ms	20ms
Communication task	600ms	60ms

4a) What is the utilisation of the task set? (1p)

4b) What would the priorities of the tasks be if the RMS policy is used? (1p)

4c) Assume that tasks are preemptive. Can the tasks be scheduled using the EDF scheduling policy? (1p)

4d) Explain what happens if the tasks are non-preemptive. (2p)

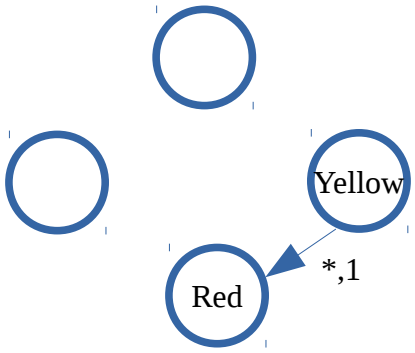
Question 5. (5 points)
Explain the foreground/background system. Draw a figure, if necessary. Mention the advantages and disadvantages of this system. Finally, describe an alternative approach that does not have the same disadvantages.

Question 6. (5 points)
Give one example of an embedded system and describe what you consider to be the five most important design metrics for this system. Remember to motivate your answers.

Question 7. (5 points)

Consider a system for controlling a traffic light at a pedestrian crossing. There are three possible outputs, red, yellow and green, which refer to the colour shown to the cars. The system has two inputs, one for whether the button has been pressed (this input is automatically set to 0 when the light is yellow or red), and one for a timer.

The figure below shows all the necessary states for a Moore state machine (i.e., the output depends only on the state), but you need to complete the state machine with labels, transitions, inputs, outputs, and indicate the initial state. Write the input as two binary digits representing the button and the timer, using the *-symbol to denote if the value does not matter (e.g., the transition from Yellow to Red is enabled when the timer input is 1, and the value of the button does not matter).



Answer sheet for question 1. Please hand this paper in together with the answers for the other questions (numbered and with AID number).

1a)	( ) 1	( ) 2	( ) 3
1b)	( ) 1	( ) 2	( ) 3
1c)	( ) 1	( ) 2	( ) 3
1d)	( ) 1	( ) 2	( ) 3
1e)	( ) 1	( ) 2	( ) 3
1f)	( ) 1	( ) 2	( ) 3
1g)	( ) 1	( ) 2	( ) 3
1h)	( ) 1	( ) 2	( ) 3
1i)	( ) 1	( ) 2	( ) 3
1j)	( ) 1	( ) 2	( ) 3