Försättsblad till skriftlig tentamen vid Linköpings universitet

Datum för tentamen	2019-10-31
Sal (1)	TER1(3)
Tid	14-18
Utb. kod	TDDI11
Modul	TEN2
Utb. kodnamn/benämning **Modulnamn/benämning**	Programmering av inbyggda system Skriftlig tentamen
Institution	IDA
Antal uppgifter som ingår i tentamen	8
Jour/Kursansvarig Ange vem som besöker salen	Ahmed Rezine
Telefon under skrivtiden	013 28 1938
Besöker salen ca klockan	
Kursadministratör/kontaktperson (namn + tfnr + mailaddress)	Veronica Kindeland Gunnarsson veronica.kindeland.gunnarsson@liu.se 5634
Tillåtna hjälpmedel	Ordbok engelska-annat språk.
Övrigt	
Antal exemplar i påsen	

EXAM
(Tentamen)

TDDI11
Embedded Software

2019-10-31 kl: 14-18

On-call (jour):
Ahmed Rezine, 013 - 28 1938

Admitted material:
- Dictionary from English to another 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
$20 \leq p \leq 30$:	3
$31 \leq p \leq 35$:	4
$36 \leq p \leq 40$:	5

Question 1, multiple choice. (10 points)
Use the answer sheet at the end of the exam. No motivation or explanation is required for this 10-points question. **Zero or more statements may be correct for each question**. Tick each statement if and only if it is correct. Ticking a wrong statement or missing to tick a correct statement gives 0 points for that question.

1a) Assume you use a single bit in each transmitted byte as a parity bit.

1. The parity bit allows you to detect all transmission errors.
2. The parity bit allows you to detect and repair any single bit flip.
3. This is only possible in UARTs based transmission.

1b) A soft-real-time embedded systems

1. That may compute wrong values is incorrect.
2. That misses some deadlines results in degraded performance.
3. Requires no deadlines are missed by the software.

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

```
#include <stdio.h>
int main() {
  int a[5];
  a[0]=0;
  a[1]=1;
  for(int i=0; i < 3; ++i)
    *(a + i + 2)= *(a + i) + *(a + i + 1);
  printf("%d \n", a[3]);
}
```

1. 1
2. 2
3. 3

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

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

1. 0
2. 1
3. 2

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

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

1. 1
2. 2
3. 3

1f) Which of the following statements are correct?

1. The foreground/background model does not allow interrupt service routines.
2. The foreground/background model may wait that several unrelated background tasks are executed before processing a foreground input.
3. The foreground/background model has problems with scalability and maintainability.

1g) A finite state machine capturing the program controlling an embedded system…

1. Can be implemented using a sequential language like C
2. Requires a special graphic based language
3. Cannot be built as embedded systems need to be programmed in a sequential language like C

1h) Some advantages of using a message passing …

1. Can be used in a shared-memory based system
2. Can result in deadlocks
3. Can result in variables being modified by other threads and therefore requires mutexes to protect individual reads and writes

1i) Requirements and specification

1. Both describe the system behavior desired by the customer
2. Can be described using state machines
3. Should allow for tracebility

1j) Assume int requires 4 bytes. What is the output of the following program:

```
#include <stdio.h>

  union Data{
    int i;
    int arr[2];
  } data;

int main() {

  data.i = 2;

  printf("%d \n", sizeof(data));

}
```

1. 2
2. 4
3. 8

Question 2. (4 points)

Introduce the I2C bus protocol and briefly explain how it works.
Mention one advantage over UARTs based I/O.

Question 3. (5 points)
Explain the steps involved when I/O programming using interrupt and when using polling. What is the difference between the two of them in terms of required CPU cycles and hardware support? Explain.

Question 4. (6 points)
Consider a task set with three periodic tasks: Task 1 with period T1= 12 and execution time C1= 2, Task 2 with period T2= 8 and execution time C2= 4, and task 3 with period T3=6 and execution time C3=2. All three tasks are to run on the same processor using some scheduling algorithm.

1. Give the processor utilization ratio in case the tasks are scheduled (1pt)
2. Which task would get the highest priority if Rate Monotonic Scheduling (RMS) is used (1pt)
3. Can the tasks be scheduled using preemptive RMS? Explain with a diagram (2pt).
4. Can the tasks be scheduled using preemptive Earliest Deadline First (EDF)? Explain using a diagram (2pt).

Question 5. (5 points)
Give a Mealy machine (outputs associated to transitions, not states) that takes sequences of 0s and 1s as input. The machine should output 1 when it finished reading a non-empty sequence of ones that is of an even length (i.e., is divisible by 2). It should output 0 otherwise.

Question 6. (5 points)
Describe the sequence of events that occur when a CPU requests a DMA (Direct Memory Access) controller to buffer data, e.g., disk data, to memory. Describe the main advantages compared to interrupt based I/O.

Question 7. (2 points)
The following code does not compile because of the definition of the constant PI. Repair the definition.

```
#include <stdio.h>

#define PI = 3.14

int main() {
  printf("With diameter 10, a circle area is: %f\n", (10/2)*PI*PI);
}
```

Question 8. (3 points)
In this question, you can use bit-level operators: e.g., "&" (and), "|" (or), "<<" (shift left), ">>" (shift right), etc. You are not allowed to use loops, additions, divisions, subtractions, multiplications or "modulo" "%" operations. Write a C function "int remainder_8(unsigned short x)" that returns the remainder of dividing the 16-bits unsigned x by 8. For instance, remainder_8(10) should return 2 and remainder_8(16) should return 0.

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