EXAM
(Tentamen)

TDDI11
Embedded Software

2020-08-19 kl: 08-12

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

Admitted material:
- You can access your individual notes, books, and even to search the internet.
- No contacts, whether physical or virtual, are allowed during the duration of the exam with any person, whether the person is related to the course or not, except for contacting the examiner via email for questions if any.
- Any **suspected breach** will be **systematically reported to the disciplinary board**. We will use Urkund (an automatic tool to combat plagiarism).

General instructions:
- **The questions will refer to your "D1 D2" digits.** These are the last two digits of the "anonymous-ID" you are assigned during the exam. You find the "anonym-ID" on the Lisam page you have retrieved this pdf from (i.e., the page you got the exam questions from). For instance, if your "anonymous-ID" is A-2709 then your D1 is 0 and your D2 is 9 (all in base 10). If your "anonymous-ID" is A-123 then your D1 is 2 and your D2 is 3. Ask the examiner if this is unclear.
- The assignments are not ordered according to difficulty.
- You may answer in either English or Swedish.
- Do **not take pictures or draw**. We only accept PDF files obtained from a text editor or a word processor. You are free to use any text editor (examples include notepad, vim, gedit, emacs, etc) or other text-based office programs (Microsoft Word or Open/Libre Office or LaTeX). We expect you to generate and to submit a single PDF.
- Be **precise** in your statements.
- **Clearly motivate** 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. (10 points)

a. Explain at a high level the steps involved when sending the byte “D2” using a UART. For example, if your D2 is 3, then the question is about the steps involved in sending the byte: 0000 0011. You do not need to consider a parity bit. (2pts)

b. Give an example of an embedded system for which you would expect hard real time guarantees and an example of an embedded system for which you would expect soft-real time guarantees. Explain and justify your choices. (2pts)

c. Assume a 32-bit variable “**x**” to which you have saved the value generated by some peripheral. Write a C function “int myCheck(int x) ” that returns true exactly when the (D1 + 2)th most significant bit equals the D1:th most significant bit. (2pts)

d. Explain the difference between “concurrent” and “over-the wall” approaches to engineering projects. Which approach seems better to you? Explain. (2pts)

e. Assume a memory mapped display at address 0xB2000. The sequence:

0x47,0x46,0x6f,0x73,0x6f,0x7f,0x64,0x6c,0x20,0x4d,
0x6c,0x2b,0x75,0x7b,0x63,0x46,0x6b,0x42,0x21,0x78,
0x2a,0x68,0x2a,0x1d,0x2a,0x5a,0x2a,0x63,0x2a,0x1f,
0x47,0x46,0x6f,0x73,0x6f,0x7f,0x64,0x6c,0x20,0x4d,
0x6c,0x2b,0x75,0x7b,0x63,0x46,0x6b,0x42,0x21,0x78,
0x2a,0x68,0x2a,0x1d,0x2a,0x5a,0x2a,0x63,0x2a,0x1f,
0x47,0x46,0x6f,0x73,0x6f,0x7f,0x64,0x6c,0x20,0x4d,
0x6c,0x2b,0x75,0x7b,0x63,0x46,0x6b,0x42,0x21,0x78,
0x2a,0x68,0x2a,0x1d,0x2a,0x5a,0x2a,0x63,0x2a,0x1f,
0x47,0x46,0x6f,0x73,0x6f,0x7f,0x64,0x6c,0x20,0x4d

Consists of 100 bytes and is stored sequentially starting with 0x47 at byte 0xB2000 encoding the first character of the first row and 0x46 at byte 0xB2001 encoding the second character of the first row. The display has 10 rows with 10 characters on each row. Each character is encoded with one byte.

1. What is the address of the byte at line D1 and at column D2?

2. Write a function char* addressOf(int row, int col) that returns the address of the byte at line row and column col. (you can assume $0 \leq row \leq 9$ and $0 \leq col \leq 9$). (2pts)

```
1          SECTION .data
2          EXTERN  list
3  char    DB  0
4
5          SECTION .text
6          ALIGN   16
7          BITS    32
8
9  BASE    EQU 2e8h
10
11 LSR     EQU BASE+5
12 RBR     EQU BASE
13 THR     EQU BASE
14 ; -------------------------------
15         GLOBAL  foo
16         EXTERN  list_insert
17 foo:
18         STI
19         PUSHA
20
21         MOV  DX, LSR
22         IN   AL, DX
23         TEST AL, 00000001B
24         JZ   _Eoi
25
26         MOV  DX, RBR
27         IN   AL, DX
28
29         MOV  [char], AL
30
31         PUSH DWORD char
32         PUSH DWORD [list]
33
34         CALL    list_insert
35         ADD ESP,8
36
37 _Eoi:
38         MOV  AL, 00100000B
39         OUT  20H, AL
40
41         POPA
42         IRET
43 ; -------------------------------
44         GLOBAL  bar
45 bar:
46         MOV  DX, LSR
47 label0:     IN   AL, DX
48         TEST AL, 00100000B
49         JZ   label0
50
51         MOV  DX, THR
52         MOV  AL, [ESP + 4]
53
54         OUT  DX, AL
55
56         RET
57 ; -------------------------------
```

Question 2. (6 points)

Consider the code depicted in the Figure to the left with the two methods foo (line 17) and bar (line 45). Answer the following questions:

1. Which of the methods corresponds to polling-based communication, which corresponds to Interrupt based communication? Explain and justify. (2pts)

2. What are the advantages and disadvantages of each approach? (2pts)

3. In some calling conventions, the callee (the method being called) need not save certain registers. Do these conventions apply when calling an interrupt service routine? Explain. (2pts)

Question 3. (6 points)

Consider a task set with three periodic tasks: Task 1 with execution time C1= (D1 % 3) + 1 and period T1 = 3 C1; Task 2 with execution time C2= C1 + 1 and period T2= 3 C2; and Task 3 with execution time C3= C2 + 1 and period T3= 3 C2. For instance, if your D1= 7 then C1= D1 % 3 + 1= 2 (with T1= 6) and C2 = 3 (with T2= 9) and C3 = 4 (with T3= 12). 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. (2pts)
4. Can the tasks be scheduled using preemptive Earliest Deadline First (EDF)? Explain using a diagram. (2pts)

Question 4. (6 points)

The Mealy machine described in the Table below has 4 states $\{s_0, s_1, s_2, s_3\}$ where s_0 is the initial state. The machine takes sequences of 0s and 1s as input. It outputs 1 exactly when the last three inputs (including overlap) build the sequence 101. For instance, the machine outputs the sequence "00101010001" when it reads "10101011101".

	in:0	in:1
Initial: s_0	out: 0 / goto: s_0	out: 0 / goto: s_1
s_1	out: 0 / goto: s_2	out: 0 / goto: s_1
s_2	out: 0 / goto: s_0	out: 1 / goto: s_3
s_3	out: 0 / goto: s_2	out: 0 / goto: s_1

Let "d" be "(D2 % 3) + 3". If your D2 is 9, then your d is 3. If your D2 is 5 then your d is 5. Give a Mealy machine that reads sequences of 0s and 1s as input. The machine should output 1 exactly when it reads a zero after reading a non-empty sequence of ones where the number of ones is divisible by d.

For instance, if your d is 3, then the machine should output 1 each time it reads a zero after a consecutive sequence of n ones where n is divisible by 3, i.e., n is in {3, 6, 9, 12, 15, …}. It should output 0 otherwise.

Follow some examples:

- If your d = 3 and the machine reads "0010111100011100 …" then it should output "0000000000000010…"
- If your d = 4 and the machine reads "0010111100011100 …" then it should output "0000000010000000 …"
- If your d = 5 and the machine reads "00111111111100 …" then it should output "00000000000010 …"

Question 5. (5 points)

As it is often the case, newly pushed stack elements get smaller addresses. A function with one argument has just been called. The stack looks as follows:

Answer the following questions:

1. The 4 bytes "`0x3fffffe8, 0x3fffffe9, 0x3fffffea, 0x3fffffeb`" contain an address. Explain what this address represents and why is it stored at the top of the stack? (2pts)

2. What role is usually played by the register ebp? Why is it inconvenient to only use esp? (2pts)

3. Assume a big-endian system. Suppose the argument to the function is the 32 bits integer corresponding to the sequence (in decreasing order of significance) of 4 bytes with the most significant byte containing the value 3 (written 0x3), followed by a byte containing your value $D1$ (if your $D1$ is 9 then the second byte is 0x9), then a byte containing the value 4 and finally (the least significant byte) containing your value $D2$. Give the address in the stack of each one of these 4 bytes. (3pts)

```
address
--------------
0x3fffffdc
0x3fffffdd
0x3fffffde
0x3fffffdf
0x3fffffe0
0x3fffffe1
0x3fffffe2
0x3fffffe3
0x3fffffe4
0x3fffffe5
0x3fffffe6
0x3fffffe7
0x3fffffe8 <-- esp
0x3fffffe9
0x3fffffea
0x3fffffeb
0x3fffffec <-- (arg)
0x3fffffed
0x3fffffee
0x3fffffef
0x3ffffff0 <-- (caller saved register)
0x3ffffff1
0x3ffffff2
0x3ffffff3
0x3ffffff4 <-- (caller saved register)
0x3ffffff5
0x3ffffff6
0x3ffffff7
0x3ffffff8 <-- ebp
0x3ffffff9
0x3ffffffa
0x3ffffffb
```