Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Technische Informatik
und Kommunikationsnetze
Computer Engineering and Networks Laboratory

Embedded Systems Autumn 2020

Intermediate Test
Lothar Thiele
28.10.2020

Preliminary Comments

- The test will not be graded by us. The purpose of the test is to provide you with some means to evaluate the state of your knowledge.
- It is highly advised, that you solve the questions under exam conditions. You should print the intermediate test. The exam is “open book”, therefore, you can use some of the material from the lectures, exercises and labs. Please note that during the exam, you can only use printed material and a pocket calculator.
- The questions are taken from old exams. The time allocated for these questions would be about 45 minutes. To pass, you should achieve about 40%-50% of the maximal points.
- I will explain the solutions and answer questions during the exercise session on Wednesday, November 4.

Short Questions (8 points)

Answer the following questions for the MSP-432 processor:

(a) (3 points) A system uses UART to transmit data, with the following configuration: the baud rate is 115 200 bits/s, 1 start bit, 2 stop bits, 7 data bits and 1 parity bit. How much time is needed to transmit 240 KB of data (1 KB = 1024 bytes)?

(b) (2 points) How many bytes can be written in a block of memory accessed using byte-addresses `0x3000_0000` through `0x308F_FFFF`?

(c) (3 points) The following function returns the value that has been read from the appropriate GPIO port (pin 7 being the MSB).

```
uint8_t GPIO_getInputPortValue(uint_fast8_t selectedPort);
```

Pins 0 through 7 of GPIO port PORT1 are equipped with buttons with pull-up resistors (when the button is not pressed, the GPIO is connected to the supply voltage; when the button is pressed, the GPIO is connected to the ground). If only buttons connected to pins 2 and 7 of PORT1 are pressed, what value will variable `uint8_t kk` have after the following line of code is executed?

```
uint8_t kk = GPIO_getInputPortValue(PORT1) & 0x3C;
```

Hint: `&` is the logical AND operator.

Pulse Width Modulation (9 points)

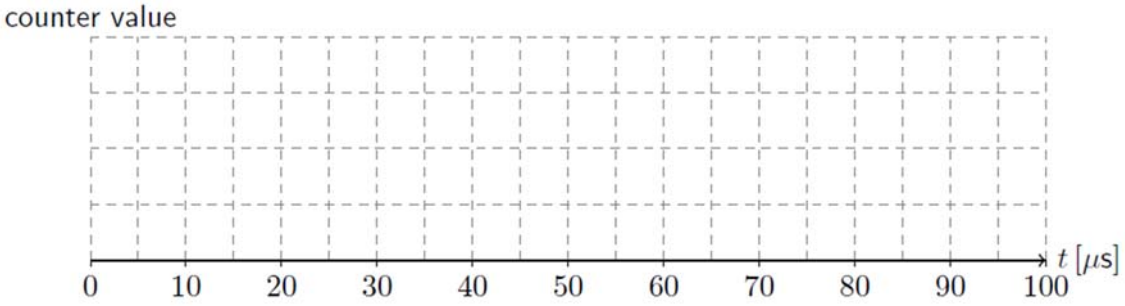
A code snippet is shown below, which shows the configuration of timer A1 for generating Pulse Width Modulation (PWM) signals on the MSP-432. Based on this code, answer the following questions.

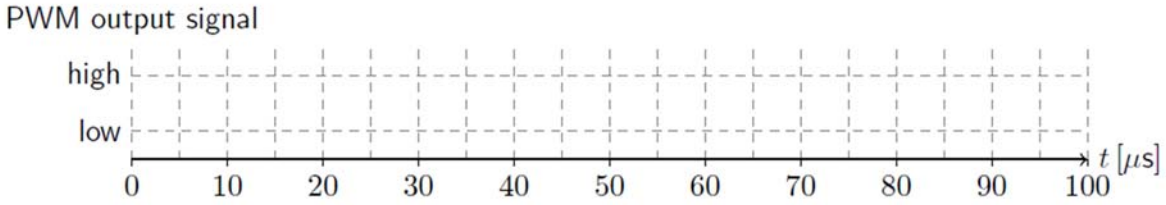
```
/* Timer_A1 up mode configuration parameters */
const Timer_A_UpModeConfig upConfigA1 =
{
  TIMER_A_CLOCKSOURCE_SMCLK,           // SMCLK clock source runs at 3MHz
  TIMER_A_CLOCKSOURCE_DIVIDER_3,       // SMCLK divider set to 3
  71,                                  // Value in CCR0
  TIMER_A_TAIE_INTERRUPT_DISABLE,      // Disable Timer interrupt
  TIMER_A_CCIE_CCR0_INTERRUPT_DISABLE, // Disable CCR0 interrupt
  TIMER_A_DO_CLEAR                     // Clear value
};

/* Timer_A1 compare configuration parameters */
Timer_A_CompareModeConfig compareConfigA1 =
{
TIMER_A_CAPTURECOMPARE_REGISTER_1,     // Use CCR1 as compare register
TIMER_A_CAPTURECOMPARE_INTERRUPT_DISABLE,  // Disable CCR interrupt
TIMER_A_OUTPUTMODE_TOGGLE_RESET,       // Toggle-reset output mode
60                                     // Compare value (CCR1)
};

/* Configure the compare functionality of Timer_A1 */
Timer_A_initCompare(TIMER_A1_BASE, &compareConfigA1);
/* Configure Timer_A1 for Up Mode */
Timer_A_configureUpMode(TIMER_A1_BASE, &upConfigA1);
/* Start counter of Timer_A1 */
Timer_A_startCounter(TIMER_A1_BASE, TIMER_A_UP_MODE);
```

(a) (3 points) Calculate the period of timer A1, in microseconds. Show the steps of your calculation.

(b) (3 points) Calculate the PWM duty-cycle (the ratio of time the PWM signal is high), in percent. Show the steps of your calculation.

(c) (3 points) Sketch the evolution of the counter value of timer A1, as well as the PWM output signal, from time 0 μs to time 10 μs. Indicate where the output signal toggles. Assume that the value of the counter is zero at time 0 μs.

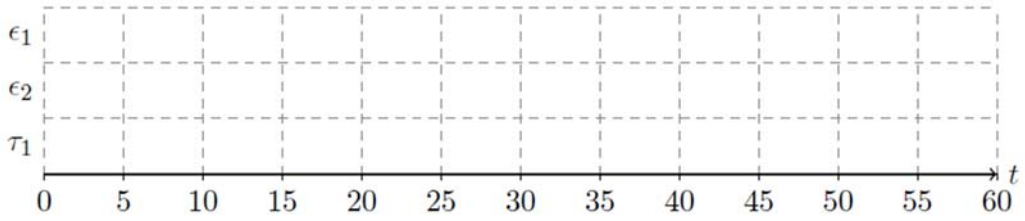




Interrupts (maximal 7 points)

A system consists of one task τ_1 and two interrupts ϵ_1 and ϵ_2, with the following parameters. Task τ_1 has period $P_{\tau_1} = 50$, execution time $C_{\tau_1} = 20$, initial phase $\phi_{\tau_1} = 0$, and relative deadline $D_{\tau_1} = 40$. Interrupt ϵ_1 has a minimal inter-arrival time $T_{\epsilon_1} = 20$, and execution time $C_{\epsilon_1} = 7$. Interrupt ϵ_2 has a minimal inter-arrival time $T_{\epsilon_2} = 15$, and execution time $C_{\epsilon_1} = 2$. For both interrupts, the overhead is $h = 3$. From the moment an instance of task τ_1 is released, until it finishes execution, interrupt ϵ_1 is ignored, while interrupt ϵ_2 is serviced immediately whenever it appears. For this system, answer the following questions.

(a) (3 points) Assuming interrupt ϵ_1 arrives at times 5 and 30, and interrupt ϵ_2 arrives at times 15 and 40, draw the execution of the task and interrupts from time 0 to time 60.



(b) (4 points) What is the minimal inter-arrival time for interrupt ϵ_2, such that task τ_1 can never miss its deadline?
Hint: Search for the worst-case scenario with regards to the arrival of interrupts.

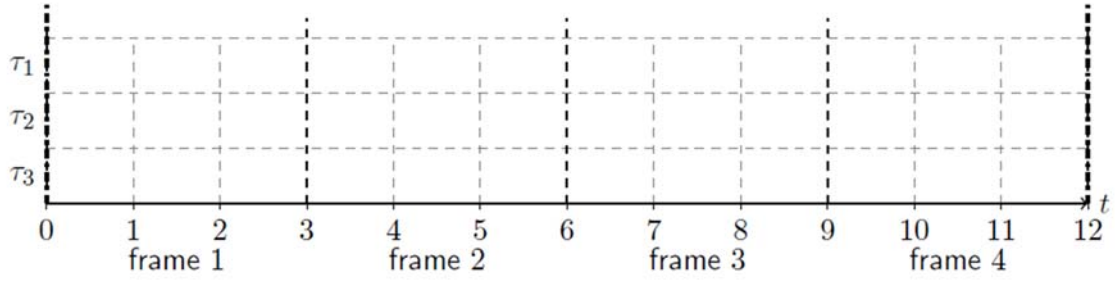
Cyclic-Executive Scheduling (8 points)

Cyclic-executive scheduling with a period $P = 12$ and a frame length $f = 3$ is used to schedule the task-set given in Table 1. Note that "frame 1" is the first frame of each period.

Table 1: A task set

Task	Period	Deadline	Phase	Execution Time	Frames
τ_1		4		1	2, 4
τ_2	12	10	2	2	
τ_3	6	5	1	1.5	

(a) (4 points) Determine one feasible assignment of tasks τ_2 and τ_3 to frames, construct the schedule for one period P and illustrate it graphically.



(b) (4 points) Determine the period of task τ_1 and its minimal possible phase, if the task is executed in frames 2 and 4.

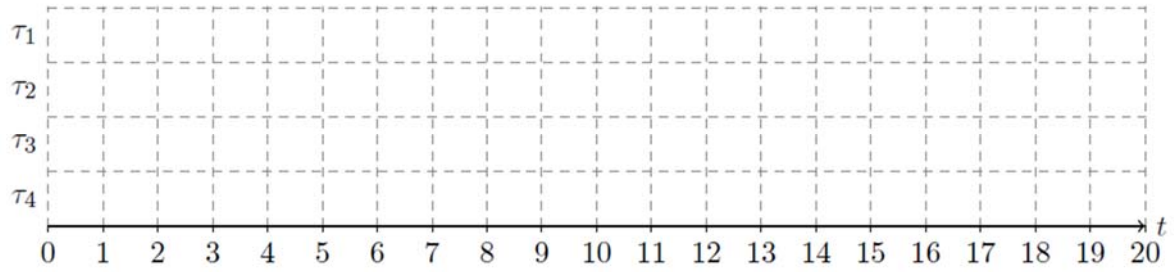
Earliest Deadline First, Online (5 points)

We have a task-set with four tasks whose jobs arrive aperiodically, and the system schedules them with Earliest Deadline First. Details on the task set, and the release times of tasks are given in Table 4.

Table 4: A task set

	τ_1	τ_2	τ_3	τ_4
Release Times	0, 4, 8, 12, 16	2, 10	0, 12	3
Relative Deadline	4	5	5	10
Execution Time	1	2	3	5

(a) (4 points) Construct a schedule according to EDF, and illustrate it graphically from time 0 to time 20. Indicate any deadline misses.



(b) (1 point) Based on your result in part (a), in the online scheduling case, at what time can the system detect that a deadline violation will happen? Online scheduling assumes that the system does not know the arrival time of jobs, until they actually arrive.

Earliest Deadline First (EDF*) (maximal 6 points)

Four jobs A, B, C, and D, execute with the following precedence constraints: $A \rightarrow B$, $A \rightarrow C$, $B \rightarrow D$, and $C \rightarrow D$. More details about the jobs are given in Table 2

Table 2: A task set

	A	B	C	D
Release Time	0	0	5	0
Deadline	12	12	8	12
Execution Time	4	1	4	3

(a) (4 points) Modify the release times and deadlines of jobs according to the EDF* algorithm, such that EDF can be used for scheduling without breaking any precedence constrains.

	A	B	C	D
Modified Release Time				
Modified Deadline				

(b) (2 points) Construct a schedule according to EDF*, and illustrate it graphically. Note any jobs that miss their deadline.