

Last name:

.....

First name:

.....

Section:

.....

Cours de 3ème année,

Section d'Electricité

Date and place: December 8th, 2017; MED 2 2524

Duration: 1h45

Systèmes Embarqués Microprogrammés

Grade:

.....

Mid-term Exam

IMPORTANT NOTES:

- The skeleton project for the exam can be downloaded from the Moodle Site under the link **“midterm_code”**. This project is similar to the ones provided during the practical sessions. In the source files there are already placeholders to implement each exercise of the exam. The exercises must be implemented in the skeleton project.
- Follow carefully all the instructions given in the current document and in the comments of the source code files of the skeleton project.
- A label “//...TO COMPLETE EXERCISE X” indicates where to write code for exercise X, X=1...n.
- The implemented exercises must work correctly **in the NDS simulator** to be considered as correctly done. However, the source code will be also evaluated after the exam.
- A compressed file of the project including the implemented code must be submitted by means of the different forms available on the Moodle Site **after finishing each exercise to be considered as completed.**

PROJECT DEFINITION:

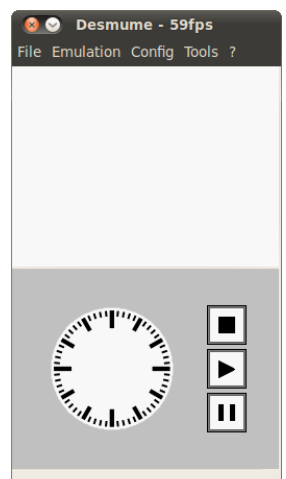
The project consists in 5 compulsory exercises to implement a simple countdown timer. When the program starts, the countdown timer is in STOP mode. The user can set the timer duration with the UP and DOWN arrows by steps of +/-30 seconds. The timer can be launched with the PLAY button (triangle). When the time is elapsed, the main screen becomes red. At any moment, the user can stop the timer by pressing the PAUSE button (double bars), and relaunch it with the PLAY button. However, if the user presses the STOP button (square), the timer is reset. In parallel to the countdown timer's execution, the red hand of the clock rotates. It always starts and finishes at the vertical position. Its angular rotation speed depends on the initial time set by the user. Note that many functions of the project are already implemented and **only the placeholders** need to be completed.

EXERCISE 1 1 point		Works?	Check?	Sign.:
-----------------------	---	--------	--------	--------

The bottom screen will be used to show the three buttons and the dial of the countdown timer. This will be done by means of an image (**plate.png**) that can be downloaded from the Moodle site of the course. For this exercise the missing parts of the function in the file *graphics_sub.c*, which is prototyped in the header *graphics_sub.h*, must be completed following the next steps:

NOTE: the macros related to the SUB engine are followed by the suffix “_SUB” (i.e. BG_PALETTE_SUB, BGCTRL_SUB etc...).

- Uncomment the line related to Exercise 1 in the **main()** function of the file *main.c*.
- Complete the function **configureGraphics_Sub_Plate()** in the file *graphics_sub.c* following the given comments to configure the SUB engine in mode 5, activate/configure the background BG3 and transfer the image information to the corresponding locations in memory.



- Download the image **plate.png** into the “data” folder of the project and create the configuration grit file in order to obtain the bitmap, and the corresponding 256 colors palette (therefore using pixels represented with 8 bits).
- Compile the project and correct the possible errors. The bottom screen should display the image shown in the figure included at the beginning of this exercise.

EXERCISE 2 1 point		Works?	Check?	Sign.:
-----------------------	---	--------	--------	--------

The upper screen will be used to display the time value of the countdown timer. This value will be decremented every 1 ms, thanks to interrupts of the hardware timer. The graphical configuration of the top screen is already implemented. However, the interrupts of the timer and screen vertical blank have to be configured. The functions inside the file *irq_management.c*, which are prototyped in the header file *irq_management.h* must be completed. In order to complete the exercise, follow the next steps:

- Uncomment the line related to Exercise 2 in the **main()** function of the file *main.c*.
- Complete the function **IRQ_initialize()** in the file *irq_management.c* following the given comments to configure the timer 0 and vertical blank.
NOTE: Do not call **irqInit()** since the touchscreen will be used in one of the next exercises.
- Compile the project and correct the possible errors. The upper screen of the simulator must display a time value decremented every 1 ms. This value is decremented when the program starts its execution.



EXERCISE 3 1 point		Works?	Check?	Sign.:
-----------------------	---	--------	--------	--------

During the execution of the countdown timer, the red hand of the clock must be rotating (clockwise). It always starts and finishes at the vertical position. In order to obtain the picture on the right, the background 2 of the bottom screen has be configured, by using affine matrix transformations.

- Uncomment the line related to Exercise 3 in the **main()** function of the file *main.c*.
- Complete the function **rotatImage_Sub_BG2()** in the file *clk_hand.c* following the given comments.
NOTE: Do not forget to use the constants: **CLK_HAND_MARGIN_X_PIX** and **CLK_HAND_MARGIN_Y_PIX**, defined inside the header file *clk_hand.h*, in order to center the hand inside the dial of the clock.
- Compile the project and correct the possible errors.



EXERCISE 4 1 point		Works?	Check?	Sign.:
-----------------------	---	--------	--------	--------

The two vertical arrows of the cross pad (UP and DOWN keys) of the NDS will be used to increment or decrement, respectively, the initial value of the countdown timer selected by the user. This increase or decrease has to be performed on the minutes and seconds with a step of +/-30 seconds, when the countdown timer is in STOP mode (**only**). Moreover, the possible initial value of the timer belongs to the range [00:00.000 ; 59:30.000]. Follow the next steps to validate this exercise:

- Complete the function **handleInput()**, in the file *irq_management.c* following the given comments. The global variable “*countdown_state*” can be used to track the state of the countdown timer. The key pressed detection on the touchpad has to be also implemented (and will be used in the next exercise).
- Compile the project and correct the possible errors. When the program starts, the value of the countdown timer on the top screen can be modified with a step of +/-30 seconds, each time we press the UP or DOWN arrows.

EXERCISE 5 1 point		Works?	Check?	Sign.:
-----------------------	---	--------	--------	--------

The behavior of the countdown timer is controlled thanks to the three buttons (STOP, PLAY and PAUSE), pressed with the touchscreen. In Exercise 1, an image showing the three buttons and the dial of the clock was loaded and displayed on the SUB engine (bottom screen). In this exercise the touchscreen will be used to implement the functionality of these three buttons. To complete the exercise, follow the next steps:

- Complete the function **handleTouchPad()** in the file *irq_management.c* following the given comments to read the touchscreen and map the touched coordinates to the corresponding control (STOP, PLAY, PAUSE). The diagram below shows the sensitivity area for each of the controls. The states (tracked with “*countdown_state*”) of the Finite State Machine (FSM), which implement the behavior of the countdown timer, have to be updated each time one of the three buttons has been pressed. These states (STOP, PLAY, PAUSE) are listed inside the file *irq_management.c*.
- Compile the project, correct the possible errors and check that the countdown timer is working as expected.

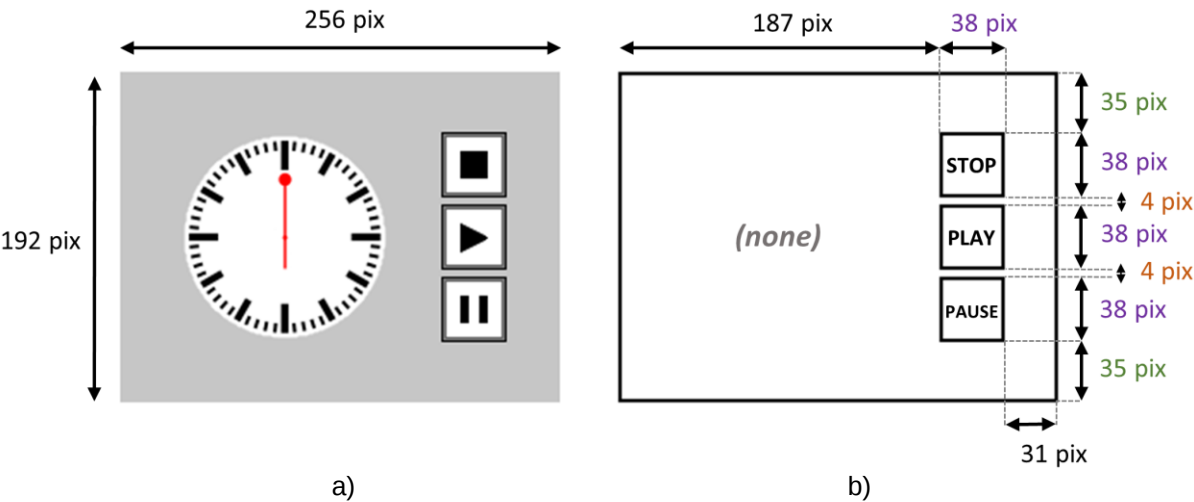


Figure 1: a) Image displayed in the bottom screen.
b) Diagram of the sensitivity areas of the touchscreen.

FINAL TIME:	Sign.:
-------------	--------