

Computer Architectures

January xx, 2020

Programming

Please read accurately:

- 1) The ARM programming part of the exam has a duration of 2 hours
- 2) You have to develop an ARM project using the KEIL μ Vision IDE
- 3) Login in your LABINF area and use the available installation (v4.74) to edit, compile and SW debug your code
- 4) Use the provided LANDTIGER board and HW debugger to prototype your project
- 5) You are allowed to access the teaching portal page; this access will be granted by the LABINF infrastructure and any other web page access will be denied and all attempts will provoke the immediate ejection from the exam: LABINF personnel will monitor the network usage along the exam.
- 6) You can bring a single USB key and use your personal projects and notes.
- 7) Before the exam time ends you MUST upload a zipped folder of the developed project called 201901xx.zip of your project including your project in the “elaborates” section of your Computer Architecture account, in the POLITO teaching portal. Late delivery will not be considered valid and always lead rejection.
- 8) The professors will reject delivered projects that produce errors during the compile phase; make sure your project compilation is free of errors.

Exercise 1 (max 30 points)

You are required to implement the following functionalities on the LANDTIGER board equipped with the LPC1768 chip.

- 1) After the pressure of the INT0 button, the system starts a single period of 6.541 seconds using TIMER2.
 - When the timer count elapses, an interruption has to be triggered
 - If INT0 is pressed during the count, it resets the timer count
- 2) During this time, the user can push EINT1 and EINT2 buttons. The system has to separately count the number of pressures for both buttons and store these values into 2 variables called **e1** and **e2**. At the end of the timer count,
 - The 4 lower LEDS have to show the binary codification of **e1**
 - The following function need to be invoked

```
int divisible(int e1, int e2);
```

which returns the value 0 if **e1** is divisible by **e2**, or the module of the division.
 - The 4 upper LEDS have to report the value returned by function **divisible**, that must be written in assembler.
- 3) Priority settings: the following priority of the interrupts needs to be setup (higher to lower priority):
 - TIMER2
 - INT0
 - EINT1
 - EINT2