The main objective of this Lab assignment (in three parts) is to help you reach some of the expected outcomes of this course – to be capable of designing, programming and debugging a digital circuit based on a microcontroller. In addition, the report you will have to submit at the end will be an opportunity to improve your skills in schematic capture and technical writing.
Make a serious effort to do everything needed before the lab classes and if you feel a bit lost on this first part, don't panic – instead come to see us and ask for guidance. We are here to help you.

Problem:
To manage the traffic flow at a given crossing there is a set of traffic lights controlled by an ATmega328p microcontroller. The traffic should alternate according to the following pattern: 50 seconds green NS (North-South) followed by 5 seconds yellow and 5 seconds red. Then 50 seconds green EW (East-West), followed by 5 seconds yellow a 5 seconds red.
At any moment, if an emergency sensor is activated, all traffic should be properly stopped and resumed 10 seconds later in the opposite direction.

1. Kick-start discussion (before class)
Make sure you understand the problem well and discuss with your group-mate and other colleagues different strategies to tackle it.

2. Preparation and submission (before class):
Fill-in the answer sheet with a solution of your own and submit it, individually, for evaluation. Pay very close attention to the submission instructions and deadlines. For the first lab class try at least to accomplish points 2.1 and 2.2 of the preparation sheet.

3. Programming (before class):
There isn't enough time to code extensively in class. Please use that precious time to discuss whatever you need with the instructor and get answers to your questions, not to code. Prepare your code at home and bring it to the lab to debug and optimize.

a) **Basic controller**. Implement, in standard C, the first state diagram of the traffic light (the one ignoring emergency situations). Structure your code in a consistent way: hardware initialization function, state dispatcher (switch) and independent functions for each state. Always use symbolic names, never *out of the blue* numeric constants. You can use the template at the end of this guide, if you wish.

b) **Graceful degradation**. Add to your code the states required to implement the following *graceful degradation* behavior: if, for whatever reason, an illegal state is reached the yellow lights should start toggling at a rate of once per second until someone resets the system. This will probably add two more states to your code.

c) **Emergency**. Add to your code the states corresponding to emergency situations, according to the second state diagram submitted. For the time being there is no way to move from normal to emergency situations and vice versa. To test your code you just need to properly initialize the state variable to enter and demonstrate the normal or the emergency behavior.

For the first lab class try to accomplish at home as much as you can but, at least, points a) and b) above.

4. Execution and demonstration (in class):
At the beginning of the lab class show the instructor the schematic of your solution for approval. Assemble the circuit, according to the approved schematic, on a prototyping board and use your Arduino to control the LEDs. You should not take more than 15 minutes to assemble the complete circuit. Compile your software and upload it to your board. Explain what happens. Call the instructor if you need help.
To demonstrate this part of the assignment reduce 10x all timings and measure them using a digital oscilloscope. Show to your instructor the three waveforms of the NS traffic flow and save them for later use in the report.

2.1. Draw by hand the schematic of a possible solution for this problem using the ATmega328p microcontroller and LEDs Kingbright 7104SGD, 7104SRD and 7104SYD to simulate the lamps. Use the lecture's examples as a starting point, consider Id=5mA and E12 resistors. Use the space below for the calculations and the provided drawing frame for the schematic. Fill-in at least the mandatory information of the title block according to ISO EN 7200:2004.

(E12: 10, 12, 15, 18, 22, 27, 33, 39, 47, 56, 68, 82)

2.2. Draw a state diagram that models the behavior in normal situations (ignoring emergencies). Use the following outputs: **GNS** (Green North-South), **YNS**, **RNS**, **GEW**, **YEW**, **REW** (Red East-West).

2.3. Draw another state diagram, independent from the first, that models the behavior in emergency situations. Use the same outputs. There should be no connection between the two diagrams.

2.4. (optional) Suggest a way to link both diagrams

I hereby declare to be the author of this work
Signature:

Section/Group:

Responsible dept.	Technical reference	Document type	Document status			
DEEC–MIEEC–SBMI						
U.PORTO Faculdade de Engenharia	Created by	Title, Supplementary title				
	Approved by		Rev.	Date of issue	Lang.	Sheet