Instituto Superior Técnico

Applications and Computation for the Internet of Things

1st Lab work: Building an embedded system [1]

Group:		
Student 1:		
Student 2:		
Student 3:		

Goal:

The goal of this work is to put students for the first time in touch with the Arduino environment – Starter Kit and IDE – to drive simple actuators (in the case LEDs).

Description:

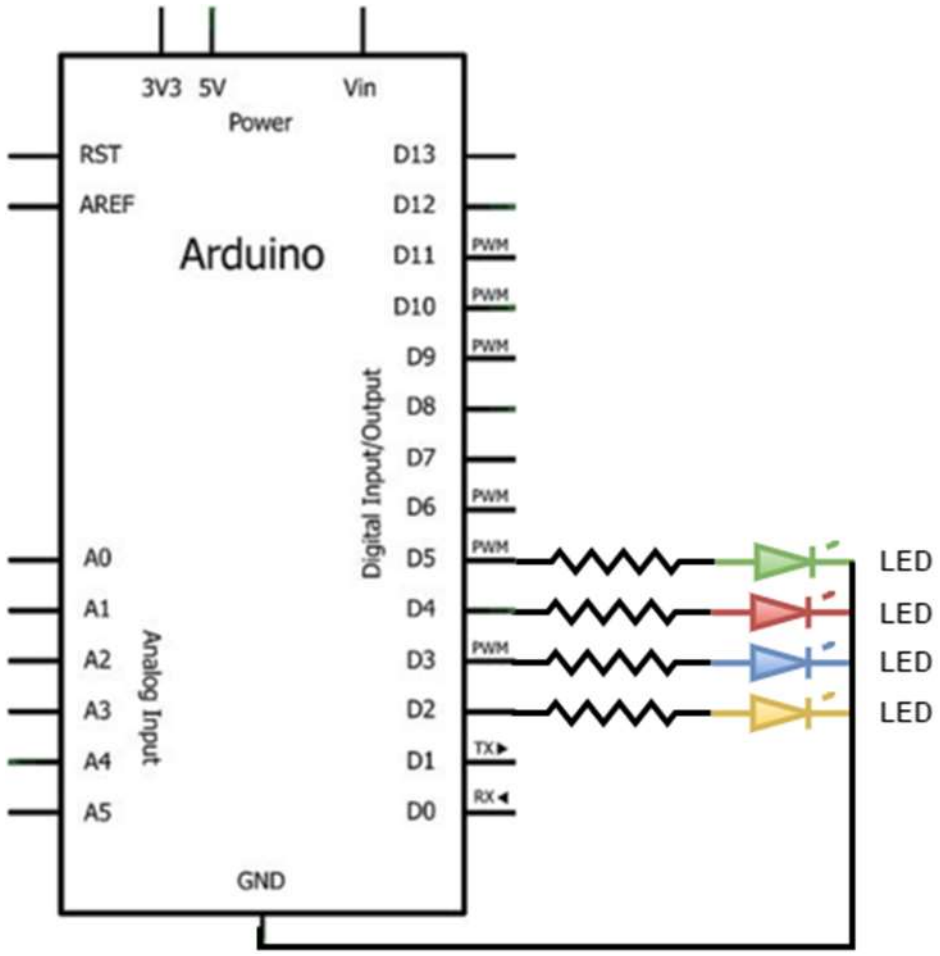
Build an embedded system using the Arduino UNO board to control 4 LEDs with different colors. In normal operation, in each 5 seconds period the system will have the the following pattern of activity, with, at most, only one LED active at a time (1 second slots):

1 – Red LED ON
2 – Green LED ON
3 – Blue LED ON
4 – Yellow LED ON
5 – All LEDs OFF

This behavior is then repeated.

The figure represents the circuit to drive the LEDs to be assembled.

[1] V1.5, Nov 2022; V1.4, Dec 2021; V1.4, Oct 2020; V1.3, Sep 2019; V1.2, Oct. 2018; V1.1, Oct. 2017.



References:

1. https://www.arduino.cc/reference/en/language/functions/digital-io/digitalwrite/
2. https://www.arduino.cc/reference/en/language/functions/time/delay/ (or https://www.arduino.cc/reference/en/language/functions/time/millis/)

Recommendations:

In order to develop your work safely, and to avoid damaging the hardware involved, remember to carry out the recommendations below. As you are working fill the boxes to be certain that you fulfill all security measures.

Always work – insert, connect, disconnect devices, etc. – with the circuit disconnect from the source (the power supply, or the PC).	
Call the teacher, or the responsible for the laboratory, before you connect the circuit to the source and power the circuit.	
Make sure the circuit is well connected (resistors, capacitors, etc.) to prevent a short circuit, or to damage the hardware. (E. g. never connect a LED directly to a terminal of	

the controller and GND, or VCC.)	
Avoid bending the component terminals as much as possible. If necessary (e.g. resistors) bend the terminal approximately 5 mm from the body of the component.	

Design the interface:

Calculate the values of resistors associated with the LEDs. (Avoid using resistors with less than 200 Ω. The lower the resistance value, the higher the power consumed and the "stress" of the components.)

R_red

R_green

R_blue

R_yellow

Interface the circuit to a press button.

Whenever the button is pressed (OFF → ON → OFF) the LED activated at the moment must remain ON. (In stage 5 the system stops with all LEDs OFF.) With the system stopped it is easier to read the voltage drop on each LED. When the button is pressed again the system will continue its normal operation sequence.

Draw and design the press button interface to the controller.

Measure (*) the voltage drops on the LEDs:

V_red

V_green

V_blue

V_yellow

(*) Use a multimeter, or read an analog input of the Arduino and transfer the value to the PC. Describe the method used.

Estimate the power consumption of the interface (the circuit with resistors and LEDs in the figure) in normal operation.

Program the application:

Add your program listing (adequately structured and commented).
Organize your program in “basic blocks”, possibly each block controlling an actuator, or processing a sensor. A basic block is a sequence of instructions with no embedded branches (except at end), and no branch targets (except at beginning).

```
void setup() {

…


}

void loop() {

…


}
```