**Instituto Superior Técnico**

**Master's Degree in Information Systems and Computer Engineering**

**Software for Embedded Systems**

# 2nd Lab work: Sensing the Real World

| Group: | | |
|---|---|---|
| Student 1: | | |
| Student 2: | | |
| Student 3: | | |

## Goal:

The goal of this laboratory work is to sense physical quantities and to control actuators according to the measured environment.

## Description:

Build an embedded system using Arduino UNO board to control 3 LEDs depending on the state of 3 different sensors (temperature, potentiometer and light intensity).

- The LED associated with the temperature sensor must be turned on when the temperature read is greater than 26 ºC (to be eventually redefined at the laboratory).
- The LED controlled by the potentiometer must blink with a period of time between 0.2 and 2 seconds, depending of the rotation applied to the potentiometer.
- The LED for the light intensity function must change its own light intensity based on the light intensity sensed in the environment (see Reference 5 about Pulse Width Modulation).

A diagram of the circuit is represented in the figure.

## References:


1. https://www.arduino.cc/en/Reference/digitalWrite
2. https://www.arduino.cc/en/Reference/AnalogRead
3. https://www.arduino.cc/en/Reference/Serial
4. https://www.arduino.cc/en/Tutorial/Calibration
5. https://www.arduino.cc/en/Tutorial/PWM
6. https://www.arduino.cc/en/Reference/Delay

## Recommendations:

In order to fulfill your work with security and not 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 with the circuit disconnect from the source. | |
| Call the professor or responsible for the laboratory, before you connect the circuit to the source. | |
| Make sure the circuit is well connected (resistors, capacitors, etc.) to prevent a short circuit, or damage the hardware. | |

## Mapping analog measurements:

Usually the digital readings retrieved from sensors do not correspond directly to the value of the physical quantity, but rather to values between 0 and a maximum binary value (say 1023 for a 10 bits word).
Therefore some mapping may be required. When the value is relative to an offset a simple mapping is adequate, but in general a more complex mathematical conversion will be needed. For example, when rotating a Servo motor with input from a potentiometer we know that when the value of the potentiometer is 0 then the servo angle must be 0° as well. Logically, if the value of the potentiometer is 1023 (its maximum reading) then the servo angle must be 180° (its maximum position):

$$map(value, 0, 1023, 0, 180)$$

Besides this some sensors require a linearization of its transfer function (physical quantity → voltage). For example, to convert the reading from a temperature sensor to the real temperature the following transformation is required:

(Based on TMP35) $T = (((sensor\ value\ /\ 1024.0)\ *\ 5.0\ )\ -\ 0.5\ )\ *\ 100$

## Programming with analog sensors:

To control an external analog sensor you must attach it to an Arduino analog pin. The software allocation of a sensor to an analog pin is done using the following code:

$$int\ const\ tempSensor\ =\ A0;$$

where A0 is the physical pin where the sensor is attached.

To read the value assigned to a specific pin use Arduino function *analogRead(PIN)*, as follows:

$$int\ temperatureValue\ =\ analogRead(tempSensor);$$

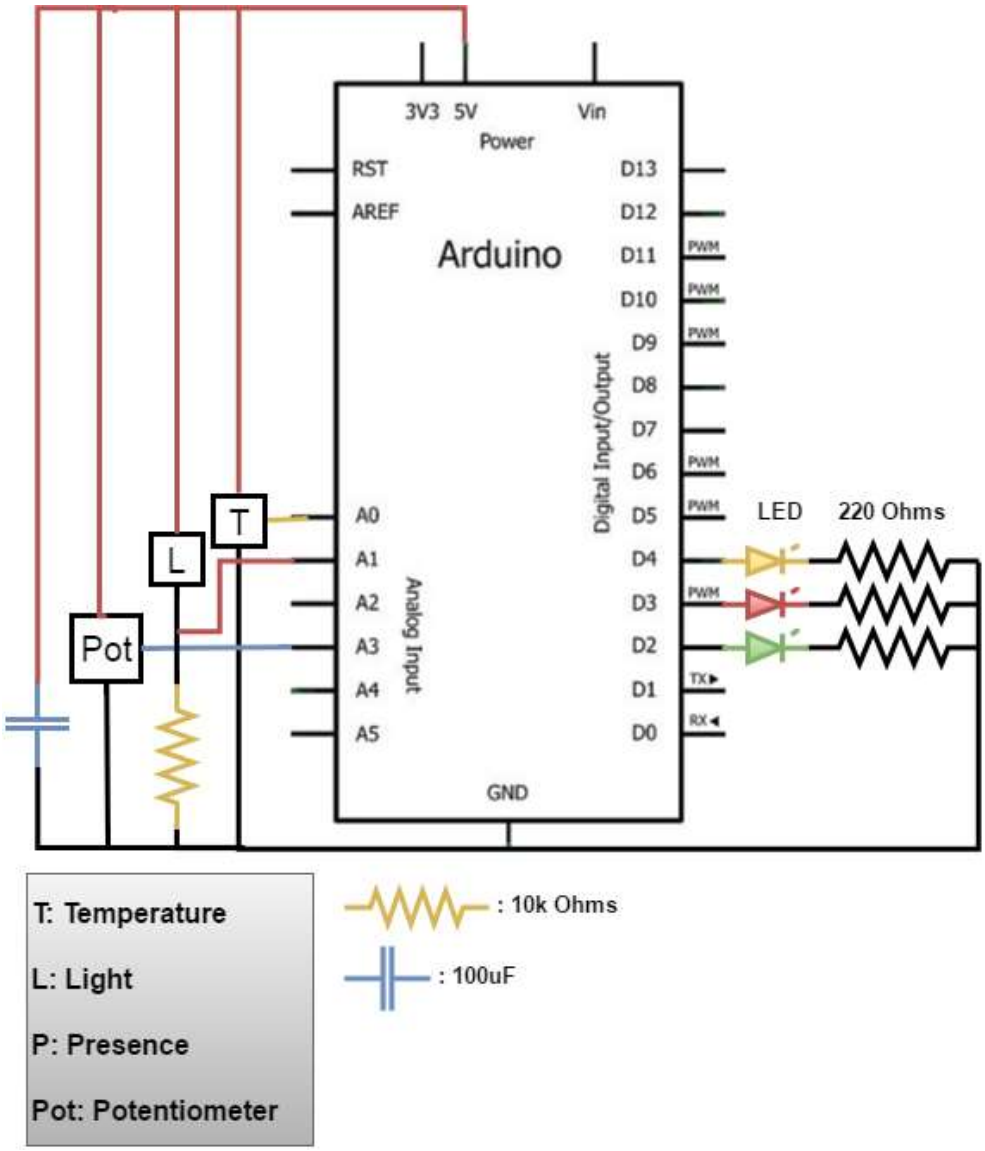
## Debug:

In order to control some variables and *debug* your program it is possible to print them to the Serial Monitor present in Arduino IDE. This kind of mechanisms is useful, for example, when you want to keep track of the temperature variation but you don´t know the exact temperature in the room.
First start by starting a serial communication with the PC:

$$void\ setup()\{\ \ldots;\ \ Serial.begin(9600);\ \ \ldots;\}$$

Then, just *Serial.print* your variables and/or strings:

$$Serial.println(temperatureValue);$$



## Program the application:

- Code:

```
void setup() {




}

void loop() {









}
```

- Questions:

Question:

For each of the three pairs sensor-actuator describe:

1. The mapping process implemented;
2. the process, or technique, used to modulate the behavior of the actuator.

Answer:

Question:

What is the system software pattern of the application?

Answer: