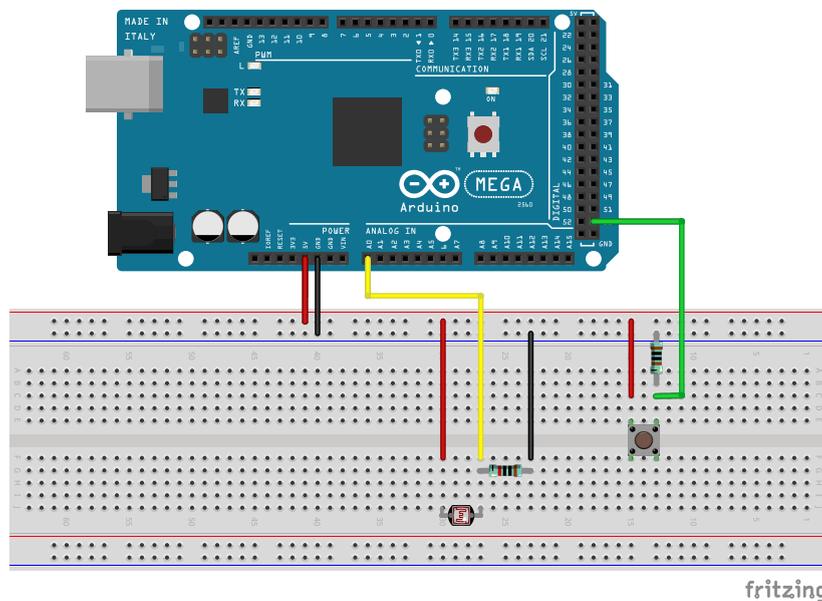


<b>Deadline</b>	<b>8th December 12 p.m.</b>
	Send as email to your tutor
<b>Mail Subject</b>	<i>[PFE15][Assignment3]\$matriculationnumber</i>
<b>Accepted Formats</b>	PDF, txt
<b>File Name Convention</b>	<i>PFE15_Assignment3_\$matriculationnumber.\$format</i>
<b>Remark</b>	You are supposed to include all code examples into your document

## 1 Measurement Series [15 Points]

In the past lectures you have been introduced to *loops* and *arrays*. On this exercise sheet you will have to implement programs using these constructs.

This exercise is about reading data from a light sensor and storing them into an array. Start by assembling the circuit depicted below.



### Requirements:

1. On the first press of a pushbutton a measurement starts.
2. On the second press of a pushbutton, or after at most 30 seconds, the measurement is complete.
3. During a measurement a measure point should be created every 10 milliseconds. In order to accomplish this your program has to query the sensor value and store it into an array.
4. When the measurement is complete, the minimum, the maximum and the average value of all measure points should be printed out on the serial console.

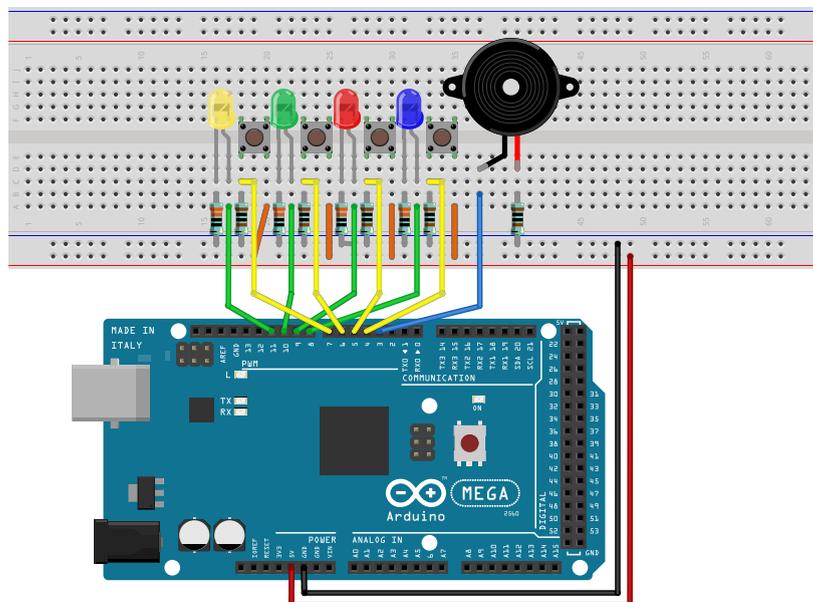
## 2 Simon [35 Points]

In this exercise you have to implement a variant of the popular game *Simon*. Here, a player has to repeat the random sound and light signals that a game automaton generates. Every new round the sequence will be extended by one additional element (sound and color) and the generation sequence increases. As soon as the player fails to replay the sequence correctly, he loses the game.

However, you will not implement the game itself, but a slightly modified variant of it. The main difference to the original is, that you and the game automaton swap places: You can specify the sequence being replayed using the pushbuttons and your Arduino has to replay and animated it using sounds and lights.

### Requirements:

1. Connect the LEDs, the pushbuttons and the piezo speaker as shown in the figure. (Hint: The piezo speaker looks differently in the figure. You can recognize it by the blue cable it is connected to.)



fritzing

2. Implement the workflow as described below.
3. The first pushbutton shall be assigned to the first LED (blue) and so forth. The blue color can be replaced by any color available to you.
4. The first pushbutton shall generate a tone with a frequency of 220Hz. The remaining pushbuttons should use 440, 660 and 880Hz respectively.
5. Generating tones is possible using the `tone` and `noTone` instructions which are described on the Arduino web page<sup>1</sup>.

### Workflow:

1. Wait for a push on one of the four connected pushbuttons.
2. Play a sound and flash a LED as assigned above.
3. Playing and flashing should happen in parallel and take about 500ms respectively.
4. Record the pressed pushbutton into an array.
5. Play the already recorded sequence. Every tone or flash of the respective LED should take about 250ms.
6. Finally, start over at step one.

<sup>1</sup><http://www.arduino.cc/en/Reference/HomePage>

### 3 Bonus: Circular Buffer [10 Points]

During the last assignment you should have noticed that you had to choose a fixed size to the array. Therefore, you could either choose this size too small –i.e. the provided space would not be sufficient to store all elements. Otherwise, you could choose the size too large as well – ultimately causing an unnecessary memory overhead.

In this assignment you have to use an array with a fixed size of 10 elements. Additionally, the program should be modified in a way that allows the “deletion” of the array’s oldest element if all slots in the array have been used – thus effectively freeing a slot.

According to this, only the last 10 keystrokes can be repeated at most.