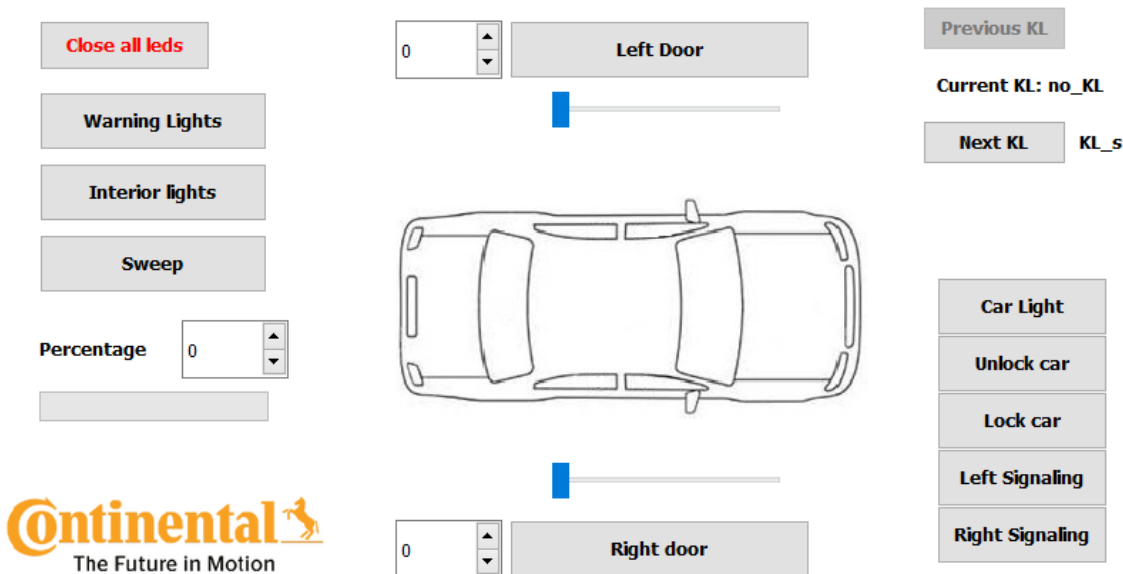# Laboratory 1 – Interior lights application

Interior lights interface

Exercise 1: Open and close one led

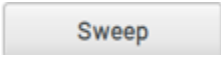**Close all leds** button must clear all the leds when is pressed.

Complete the specific function to do this action.

Interior lights button must open and close 1 LED when is pressed using **set_interior_lights** function.

Complete the specific function to do this action.
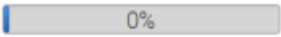
Exercise 2: Sweep all leds

This exercise uses all the leds.

Sweep button must give "One led at a time" effect when is pressed, start with the 0 led and open each led till the 3'th led.

HINT: Use **set4leds** and modify it.

Exercise 3: Control led brightness

This exercise uses the 6'th led.

Percentage spinbox is scaled between 0-100, this means the led brightness percentage. The progress bar is also scaled between 0-100 and must go through all the values untill his value is equal with led brightness percentage, creating the fade event.

Progress bar value is stored in a variable and in the **valuechange** function is checked if it is bigger or less than spinbox value.

If progress bar value is **less** than spinbox value you should **fade out** from the actual brightness (meaning the progress bar percentage) to the next brightness (meaning the spin box percentage).

Change "**change_pb_down_value**" function to make this action work.

If progress bar value is bigger than spinbox value you should **fade in** from the actual brightness (meaning the progress bar percentage) to the next brightness (meaning the spin box percentage).

Change "**change_pb_up_value**" function to make this action work.

HINT: Use **ENTER key** after setting the value you want for the spinbox.

Exercise 4: KL control

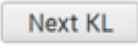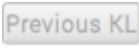Kl is the abbreviation from 'klemme', which is the German term for connector/connection.

no_KL – 5,4,3,2 leds closed

KL_S – 5'th led, white color (is the ignition switch position #1 - accessory)

KL_15 – 4'th led, green color (ignition switch position #2 – ON)

KL_50 – 3'rd led, red color (ignition switch position #3 – start, CRANKING)

KL_75 – 2'nd led, blue color (ENGINE RUNNING)

KL_list is a list that contain all the KL's. Using [Next KL] and [Previous KL] you have to go through all this list and set the following status for current KL:

no_KL = all 4 leds closed

KL_S = just KL_S led open

KL_15 = KL_s, KL_15 leds open

KL_50 = KL_s, KL_15, KL_50 leds open
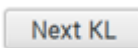
KL_75 = KL_s, KL_15, KL_50, KL_75 leds open
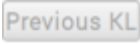
Complete **KL_lights**, **prev_kl_function**, **next_kl_function** functions to make this application work.

Description of existing functions:

set_bg_colors – make the application running like the real leds, opening interface leds.

**prev_kl_function** - when [Previous KL] button is pressed it changes the current KL to the previous KL.

next_kl_function – when [Next KL] button is pressed it changes the current KL to the next KL.

set_enable – set Previous KL button to disable when current KL is no_KL and set the "Next KL" button to disable when current KL is KL_75.

Exercise 5: Obstacle detection (Optional, BONUS)

This exercise uses the 0 and 1'st led.

Let's start talking about the widgets that we use to make this exercise running. In left of the "Left Door"/ "Right door" button there is a spin box. This spinbox is actually the **obstacle**. The spinbox is scaled between 0-100 (meaning the distance in cm that a door can open). Setting a value between 0-100 we fix a obstacle at the given value.
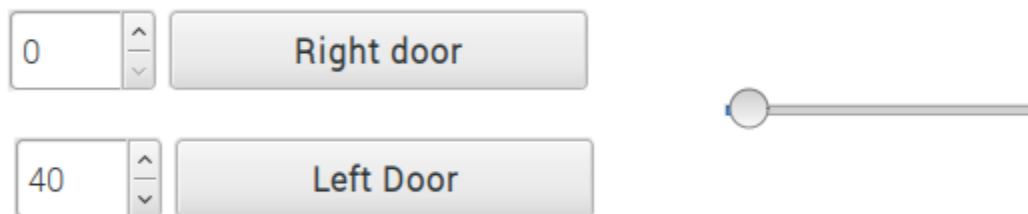
There are also 2 sliders, those sliders are also scaled between 0-100.

The "Left Door"/ "Right door" button are making all the magic, because **after** we set the obstacle we just press one of these buttons to simulate an open door. Then a led has to **fade in** from 0 to the brightness equal with the **obstacle** value. When the obstacle value is reached, the led should change in a red colour and **stop fading in**.

In order to simulate closing of the door, you will use the sliders and **fade out** the leds.

 HINT: We can only fade out from the sliders, the buttons are used to fade in.

Complete specific function to make the obstacle detection work.

Attention! exercises 6,7,8,10,11 use **setWarningLights** to set the color of the LEDs

Exercise 6: Warning Lights

> **Warning Lights**

When the button is pressed for the first time 4 LEDs will flash until the button is pressed again.

If **Right Signaling** or **Left Signaling** are on, they must be switched off during the operation of the **Warning Lights**, after which they must resume their functionality.

Complete the specific function to do this action.

Exercise 7: Left Signaling

> **Left Signaling**

When the button is pressed for the first time 2 LEDs will flash until the button is pressed again.

If **Warning Lights** or **Right Signaling** are on, they must be switched off during the operation of the **Left Signaling**, after which they must resume their functionality.

Complete the specific function to do this action.

Exercise 8: Right Signaling

> **Right Signaling**

When the button is pressed for the first time 2 LEDs will flash until the button is pressed again.

If **Warning Lights** or **Left Signaling** are on, they must be switched off during the operation of the **Right Signaling**, after which they must resume their functionality.

Complete the specific function to do this action.

Exercise 9:

> **Car Light**

The button must turn on and off an LED. Complete **carLight** function to do this.

Exercise 10:

Unlock car

When the button is pressed **Warning Lights** should flash for 2 times and the **Car Light** must be switched on if is off.

Exercise 11:

Lock car

When the buttons is pressed Warning Lights should flash 1 time and **Warning Lights**, **Left Signaling**, **Right Signaling** and **Car Light** must be switched off if they are on.

BONUS:

Open the LedMatrix file from the same directory. In this file you have an 8x8 matrix and 3 buttons at your disposal. You will use the code from function1 to light up certain LEDs.

Ex1   Ex2   Ex3