ITU Control & Automation Eng. Dept.
KON309E Microcontroller Systems
Experiment 2

Aim: Controlling the brightness of the LEDs using PWM.

For this experiment, you will need to consult the following reference documents:

- LPC824 user manual
- LPC82x datasheet
- Alakart schematic diagram

Write the code **using the peripheral support libraries (Xpressso SDK)**.

PART I

We will connect three LEDs. We will select each LED by pressing one button, and adjust its brightness using the two other buttons for brightness up and brightness down.

1. Construct a circuit consisting of 3 LEDs (**1 red, 1 green, 1 yellow**) and 3 buttons.
   Button 1 is for selecting one LED and **Button 2** and **Button 3** are for controlling its brightness.
2. At power ON, only red LED is ON with a brightness level of 15%. The other two LEDs must be OFF. However, their brightness level default value must also be 15%.
3. Use Button 1 to select the next LED.
   a. When Button 1 is pressed the current LED will turn OFF and the next LED will turn ON in the following order:
   Red -> green -> yellow -> red ...
   b. Only the selected LED must be ON at any given time.
   c. The brightness level of each LED will be saved, so that the next time that LED is selected, its brightness will be restored to its previous value.
4. Use Button 2 and Button 3 to change the brightness of the selected LED.
   a. Brightness value must range from 5% to 60%, in 5% increments.
   b. Button 2 is used to increase the brightness and Button 3 to decrease it.
   c. Brightness must change by one step at each button press and stay the same until the button is released pressed again.
   d. If the brightness limit is reached, pressing the same button must not have any effect.

Note: Please pay attention to the folowing:

- **Insert Alakart into the breadboard also** when preparing your circuit.
- Use PWM generated directly by a timer peripheral to adjust the brightness levels.
- Use a **proper debouncing method** to read switches, so that the user interface is steady and stable.
- Use 10kΩ pull up resistors for buttons and 220Ω series resistors for LEDs.
- The long pins of the LEDs are anodes.
- Connect V_{DD} (3.3V) and **Ground** pins of the microcontroller to the breadboard's (**+**) and (**–**) sockets via jumpers.
- The processor pins where the buttons are connected are set as digital inputs, and those where the LEDs are connected are set as digital outputs.
- Use **switch case** structure for coding the finite state machine (FSM).

Bonus:

Can you eliminate the physical pull-up resistors but configure the GPIO using IOCON peripheral so that the internal strong pull-up function is enabled?

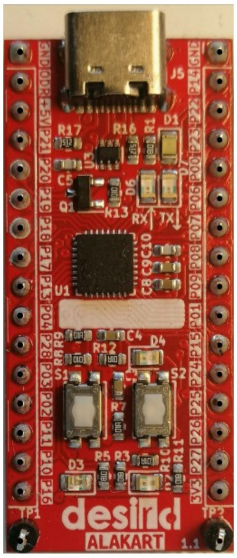
Report Content

Your report must be a formal account of what you did in the experiment.
Make sure that it includes the following items:

- How you have configured the processor i.e.:
  - What pins are inputs, what pins are outputs, etc.
  - Which peripheral devices were explicitly powered up,
  - How the GPIO peripheral was configured;
  - What values were written to which registers,
  - Which default settings of the peripheral were used.
    These must appear in all of your reports from now on.
- Propose a PWM frequency and **show how you calculate the prescaler and match register value for that frequency**.
- The finite state machine diagram that you have designed. Use **draw.io** to draw it. Make sure that you **clearly mark the state names, transition conditions and outputs**.
- Include the code that implements the finite state machine (The switch-case part of the code only) as formatted text.
  (Photographs or screenshots are not accepted!)
- What problems you have encountered and how you solved them.

Appendix: Pin connections of components on Alakart

- LEDs:
  - RED: D4 on GPIO PIN12
  - Blue: D3 on GPIO PIN16
  - Green: D2 on GPIO PIN27
  - White: D1 Power on
  - Green: D6 Transmit to PC
  - Red: D7 Transmit from PC
- Buttons:
  - S1: Reset
  - S2: ISP (enter boot mode)
    Also: User button.
- Red Pins: Test
  - TP1: GPIO PIN16
  - TP2: GPIO PIN27
- Purple Pin: Open drain FET
  - ODR: GPIO PIN21



• GND	GND
• ODR	P14
• +5V	P22
• P21	P23
• P20	P00
• P19	P06
• P18	P07
• P17	P08
• P13	P09
• P04	P01
• P28	P15
• P03	P24
• P02	P25
• P11	P26
• P10	P27
• P16	+3V3
• TP1	TP2

Note:

- PIO0_10, PIO0_11 are open drain pins. Research what this means for the experiment.
- PIO0_2, and PIO0_3 are debugger pins by default. PinPIO0_5 is Reset pin by default. You can use them if you wish, but will **need to disable their default functionality in PINENABLE0 register** first.