1DT301 – Lab assignment 3

Goal for this lab:
- Learn to access data stored in .data section by using PC relative addressing
- Learn to write simple subroutines
- Learn how to connect a pushbutton and read input values

Presentation of results:

After each task has been solved, contact a teacher for grading. To complete the lab, you also need to submit the code on Moodle.

Each group must submit solutions to all lab tasks, and **the names of all group members must be in the header of every file**. **Before sending in the code, your group must show the results to the teacher and all group members must be present during the presentation.**

Tasks

Task 1:

Write a program to calculate the average value of 8 numbers defined in the .data section. Then, show the result in the terminal (i.e. Minicom). You can use the following code as a template:

```
.thumb_func          @ Necessary because sdk uses BLX
.global main          @ Provide program starting address to linker

main:
    BL  stdio_init_all  @ initialize uart or usb
loop:
    LDR R0, =my_array
    MOV R1, #8          @ 8 elements in the array
    BL average          @ Call the subroutine average, with parameters R0 and R1

    @Print string and average value
    MOV R1, R0          @ Move average value to printf parameter R1
    LDR R0, =message_str    @ load address of helloworld string
    BL  printf       @ Call pico_printf
    B   loop         @ loop forever

@Subroutine average takes the parameters:
@R0 - Memory address to first element of integer array
@R1 - Number of integers in the array
@R0 - Return value (integer average value)
average:
    @Your task is to implement this subroutine


.data
    .align  4   @ necessary alignment
    message_str: .asciz   "Average value %d\n"
    my_array: .word 10, 20, 30, 40, 50, 60, 70, 80
```
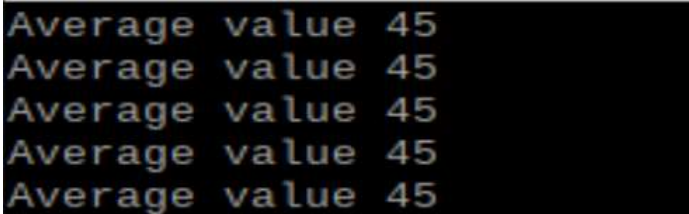
To show the result, the easiest way is probably to write the result continuously in an infinte loop, as in the Hello World program:

```
Average value 45
Average value 45
Average value 45
Average value 45
Average value 45
```

Remark: There is no division operation in the ARMv6m, but you can assume that the number of integers in the array is a power of two!

You **must** implement the average calculation as a subroutine!

Task 2:

Connect an LED to GP0 and two pushbuttons: One connected to GP1 and one to GP2.

The pushbuttons look like this:

First, you have to find out how to connect the pushbutton and how to setup the port to read if it is pushed down or not!

Then, write a program with the following functionality:

If pushbutton on GP1 is down, turn on the LED. If pushbutton on GP2 is pushed down, turn off the LED.

Task 3:

Extend the circuit you made in Task 2 by connecting one pushbutton to GP3 and one to GP4.

Then, write a program with a blinking LED and program the pushbuttons to control the blinking speed in the following way:

- When pushbutton on GP3 is pushed down, increase the blinking speed, so that the LED blinks faster.
- When pushbutton on GP4 is pushed down, decrease the blinking speed, so that the LED blinks slower.

The program should start with the LED blinking with a 1s ON, 1s OFF and increse/decrese speed by increments/decrements of 100ms at the time.

Be sure to limit the blinking to actual corresponding visible values.

Therefore, not too fast or it will appear always ON, not too slow to appear more than 2s on/off (10 increments and 10 decrements in total from the starting values)