

## 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, as a .pdf file. Each group must submit solutions to all lab tasks, and **the names of all group members must be in the header of every file**. Use the following header:

*Course 1dt301: Computer Technology 1*

*Lab Assignment 3, 2023*

*Names of BOTH members in the lab group*

# 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

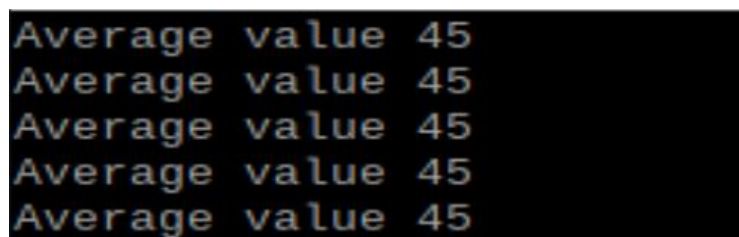
main:
    BL  stdio_init_all @ initialize uart or usb
loop:
    LDR R0, =my_array
    MOV R1, #8         @ 8 elements in the array
    BL  average        @ Call the subroutine average, with parameters R0 and R1

    @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"
    .align 4 @ necessary alignment
    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 infinite 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 pin 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.

In this task, **you must use a C function** to read the pin!

**Hint:** You can use a C function in the RPi Pico SDK to read the input that works almost like the `gpio_put()` function. Search the SDK documentation to find out about this function. Remember to add it to the `sdlink.c` file because the function is an inline function!

## Task 3:

Connect a LED to GP0 and two pushbuttons: One connected to GP1 and one to GP2. This is the same setup as Task2.

This time, **you are not allowed to use C functions** to read from the input pin or to turn the LED on/off! You have to use reads and writes to and from hardware registers. You can use the example from the book, chapter 9.

You **are allowed to use C functions to initialize and set direction**, like in the program listing 8-1 in the book. Alternatively you can use the `gpioinit` function in Listing 9-5 to initialize the pins.