

2DT901 – Lab assignment 4: C-programming and interrupts

Goal for this lab:

- Gain some experience in using the C language to program the Pico.
- Learn to use timer interrupts to make a counter.
- Learn to use GPIO interrupts to register inputs.

Presentation of results:

This is the final lab in the course and there will be no more lab sessions in the course. Because of this, you have the following two options to present the functionality:

1. If you are finished by 28 May, you can present it to the lab teacher at the tutoring session.
2. If you need more time, you can present it by making a short video and including a link to the video when you submit the lab.

Requirements:

- Start by presenting all group members with group number, and all group members present themselves by name.
- Show both the hardware configuration, the function and explain some of the code in each task. **All group members must be active during the presentation of the hardware and the code!**
- The video can be uploaded on MyMoodle together with the report, if the file is not too large. You can also upload the video to an online service, such as YouTube, and submit the link on MyMoodle.

he 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 2dt901: Computer Organization

Lab Assignment 4, 2024

Names of BOTH members in the lab group

Tasks

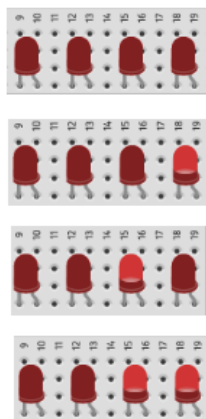
Task 1: Input and output in C

Use the same setup as in Lab3 Tasks 4 and 5, that is, a LED connected to GP0 and buttons connected to GP1 and GP2. Use button on GP1 to turn on LED, button on GP2 to turn it off.

- Write a C program to implement the functions. To read buttons and control LED, use the C function `gpio_put(...)` and `gpio_get()`.
- Re-write the C program so that it does not use the gpio functions, but instead hardware addresses of the SIO. However, you ARE allowed to use C functions to initialize the GPIO pins and set their directions!
- Connect one more LED to GP6. Extend the program from b) so that it turns on or off both LEDs simultaneously.

Task 2: Binary counter

Connect four LEDs in a row to make a binary counter. The counter should count from 0000 to 1111. The picture below shows the counter counting from 0000 to 0011.



The LEDs should be connected to ports GP1, GP2, GP3 and GP4.

Connect one button to GP5 and one button to GP6 with the following functions:

- Let the button on GP5 increment the counter (increase one step). If increase button is pressed when counter value is 15, nothing should happen!
- Let the button on GP6 decrement the counter (decrease one step). If decrease button is pressed when counter value is 0, nothing should happen!

Let the counter start at value 0. You must use interrupts to handle the inputs from the buttons! There will probably be problems with bouncing buttons (one button press counts as many) but you can ignore this problem.

Task 3: Binary counter with reset button

Use the same counter setup as in the previous task, but this time, let the counter increase automatically using a timer interrupt. Also, connect a button to GP0 to reset the counter. You don't need to use the buttons at GP5 and GP6 in this task.

Requirements:

- There should be 1 second time interval between the counter values.
- Stop the count when the counter reaches its maximum value 1111.
- At any time, the Reset button should reset the counter to 0000 and after that, the counter shall resume its counting.
- The counting must be implemented with a timer interrupt and you have to use GPIO interrupts to handle the signals from the buttons!