1DT301 – Lab assignment 4:
C-programming and interrupts

Goal for this lab:

- Gain some experience in using the C language to program the Pico both on a low level and a higher level.
- Learn to use timer interrupts to make a counter.
- Learn to use GPIO interrupts to check for inputs.

Presentation rules

1) You must submit a report for assignment. The source code and makefiles should be uploaded to Gitlab, or a similar service. The Report and source code must be submitted on Moodle before the deadline.

2) You have two options for the report:
    a. Write a report in LaTeX as in Lab1, submit the report as .pdf and give a link to Gitlab, either in the report or on Moodle.
    b. Write the report in Markdown, as an .MD file included in your gitlab repo and submit the Gitlab link on Moodle.

3) You must present the lab to your lab techer during week 44 (27 oct – 30 oct). All group members must be present during the presentation, and you need to show that the hardware set up and code works. Both group members must be prepared to answer questions about the hardware and source code during the presentation!

4) To pass the assignment, you must solve all tasks, pass the presentation, submit working code as a Gitlab link and write a well-structured report. Only submitting code is not enough to pass the assignment!

Important note

You must use interrupts to control the buttons on task 2 and on task 3 both the counter and buttons must be controlled by using interrupts!

Tasks

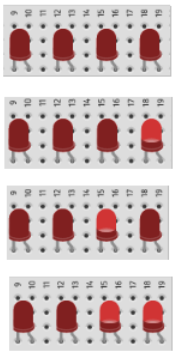
Task 1: Input and output in C

Use the same setup as in Lab3 Tasks 4 and 5, that is, an 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.

a) Write a C program to implement the functions. To read buttons and control LED, use the C function gpio_put(…) and gpio_get().
b) Re-write the C program so that is 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!
c) 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 previos 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 must use GPIO interrupts to handle the signals from the buttons!