1DT301 – Lab assignment 2

Goal for this lab:

- Getting started using the RPi Pico microcontroller and the development environment.
- Learn to program in the ARMv6-M Thumb instruction set.
- Learn to use the printf instruction to show output strings in a terminal.
- Learn to program the output pins and use them to control LEDs and a 7-segment display

Presentation of results:

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 41 (6 oct – 9 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 pass 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!

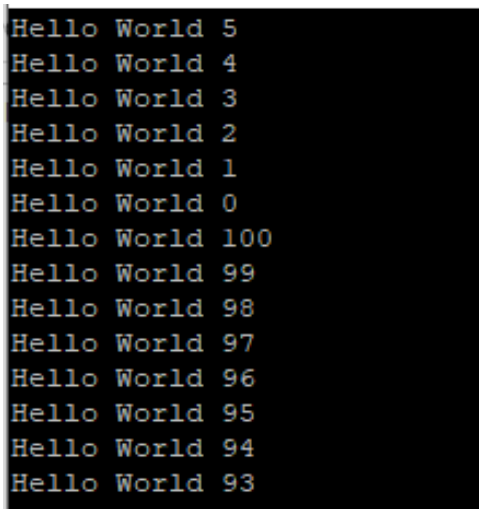
Important note

All programming tasks must be done in Assembly code! You are not allowed to use a high-level language like C och Python. However, you are allowed to call C functions from your Assembly programs!

Tasks

Task 1:

- Download the source code for the Hello World program presented on page 24 in Stephen Smith's book.
- Modify the program so that the counter starts at 100, decreases by one each time the loop is repeated. Print the text Hello World followed by the counter number. When the counter reaches 0, it shall be reset to 100 and start decreasing again, in an infinite loop.
- Compile and upload the program to the Pico. Open minicom, putty or a similar terminal to show the result. Example:



Task 2:

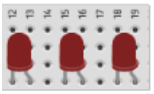
Connect one green, one yellow and one red light to three of the pins, in the same order as a traffic light. Then, write a program to make them flash like a traffic light.

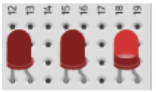
Use pins GP0 for green, GP1 for yellow and GP2 for green LED! Also, make sure you understand how to connect an LED and remember to ALWAYS connect it in series to a resistor!

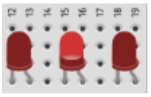
You can use the code in Listing 8-1 (page 150). Copy everything from folder Chapter 8/1/ and edit the .S file to write your own code!
Note: The functions link_gpio_put and link_gpio_set_dir are C functions, not assembly, but you don't need to care much about this, just **remember to copy the file sdklink.c**!

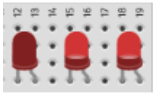
Task 3:

Use the same setup as in Task 2 and create a binary conter that counts from 000 up to 111. The picture below shows the first 4 steps. When the counter reaches its final value 111, count back to 000 one step at a time, repeate infinitely. Make one second delay between the counter values.



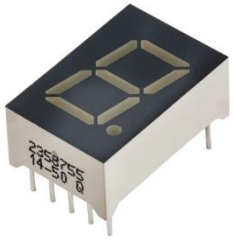






Task 4:

Connect a 7-segment display to the Pico. Implement a counter that counts from 0 up to 9, then count back to 0, and repeats infinitely. Make a delay of 1 second between the increments/decrements of the counter!



Use pins GP0, GP1, …, GP6 for segments A, B, …, G, respectively!
Hint: If you want to set several GPIO ports using only one instruction, a C function called `gpio_put_all(…)` can be useful!