Otto-Friedrich-Universität Bamberg
Fakultät WIAI

Lehrstuhl für Praktische Informatik, insbes.
Systemnahe Programmierung – Prof. Dr. Michael Engel
https://www.uni-bamberg.de/sysnap
michael.engel@uni-bamberg.de

Exercise 1

Bare Metal Programming

1.1 Bare Metal Programming

Install the RISC-V 64 cross compilation toolchain and the qemu emulator on your computer, then run the "Hello world" binary executable program from the gitlab repository in the qemu emulator:
https://gitlab.rz.uni-bamberg.de/sysnap/teaching/ose2025code/-/blob/main/exercise1/c/binary/hello
Check out the example C source code template from the github repository (the README there also gives links to the required tools)
https://gitlab.rz.uni-bamberg.de/sysnap/teaching/ose2025code/-/tree/main/exercise1
This program is a skeleton in which you need to implement missing code.

- Implement a function `void putachar(char c);` that outputs a single character to the UART as described in lecture 2. Remember to check the UART so that you wait for the transmitter to be available before sending a new character.
- Test your `putchar` function by adding some calls to the function to `main`.
- Add a function `void printastring(char *s);` that prints a C string passed as parameter `s` using your function `putachar`. Remember that a C string is terminated by the `NULL` character (binary value of the character = 0).
- Extend main to print the text "Hello, world!" to the emulated serial interface (UART).
- Extend your example program to read a string from the serial line ("keyboard" input in qemu), convert it to uppercase and output the converted string.