

Practical 6

You should write and submit four files:

- Makefile
- Linker script
- C source file
- Assembly startup file

These files should be submitted in a zip archive. The files must be in the root of the zip file. In other words, when I open the .zip, the files must be immediately visible - you mustn't have to go into a folder in the zip to find the files. Ask a tutor if you're unsure of how to do this.

Running 'make' in the root of your .zip should produce an elf file.

Part 1: (2)

Write a main() function which does the following:

- enable GPIOB in RCC
- set lower byte of port B to outputs
- write 0xA0 to LEDs
- infinite while loop doing nothing.

I know that it's bad practice, but for now you can just write a value directly to the RCC AHBENR rather than doing bit masking to set the IOPBEN bit high. Bit masking in C will be taught shortly.

Part 2: (2)

Replace the empty infinite while loop with some logic which increments the value on the LEDs after a 1 second delay.

In an infinite loop:

- set the contents of address 0x2000 00F0 to some very large value.
- decrement the contents of 0x2000 00F0 by 1 until it gets to 0.
- set the value on the LEDs to the current value on the LEDs plus 1.

This should result in you having a finite while loop within an infinite while loop.

You will need to find a suitable starting value for your counter memory address by trial and error. Start at 1 000 000 and work from there.

The resulting behaviour should be the LEDs counting up from 0xA0 by 1 every 1 second.

Marked out of: 4