

## 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 0xAA to LEDs
- infinite while loop doing nothing.

### Part 2: (3)

Replace the empty infinite while loop with some logic which increments the value on the LEDs after a 1 second delay in the following pattern (AA -> FF -> 55 -> 00) and repeat 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 appropriate pattern on the LEDs.

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 cycling through each pattern every 1 second.

### Part 3: (2)

If SW1 is held, freeze the patterns cycle. When SW1 is released, resume with the patterns cycle.

Marked out of: 7