

# Practical 6

You are required to submit a makefile along with your code.

It should be called "*Makefile*" (no extension)

When run, the makefile should produce a file called "*main.elf*" (actually, anything with a .elf file extension will be fine)

If running Windows, you'll need to patch *make.exe* (the program which understands and runs makefiles) into your system. Download it from *Resources -> Software* and paste it into:

C:\Windows\

## Part 1:

A timer should generate an interrupt at a rate of 2 Hz. I.e: one IRQ every 0.5 seconds.

You should **calculate** the register values to get this frequency, and then **verify** on a scope.

No guessing here - we're engineers, damnit.

The Timer ISR should implement a binary counter: it should simply increment the value displayed on the LEDs by 1.

## Part 2:

In the event that SW3 is held down, the timer ISR should decrement the number on the LEDs by 1, instead of incrementing them.

## Part 3:

In the event that SW2 is held down, the timer frequency should be modulated by POT0.

When the pot is outputting 0 V, the IRQ frequency should be 10 Hz.

When the pot is outputting 3.3 V, the IRQ frequency should be 2 Hz.

Linear in between.

The suggested way to implement this is that every time an IRQ occurs, the POT0 value should be read and used to modify the timer frequency, thereby altering the time until the next interrupt.

## Notes:

This is the first time you're working with IRQs which can be difficult to debug if not working properly. The suggestion is to start with a very simple program. Forget about the ADC and push buttons. Just get the simplest part 1 working first: A timer generating an interrupt which adds 1 to the LEDs. Should be around 20 lines of code. Then you can know your interrupts are happy and move on.

## Marking:

Part 1: 4

Part 2: 1

Part 3: 3

Total: 8

