

Typical Questions for Experiment 1

Checkpoint 1

- Where is the microcontroller on the DSLMU Microcontroller Board?
- Where is the crystal oscillator? What does it do?
- What is the purpose of the static RAM components? Where are they on the board?
- What is the Flash ROM component used for? Where is it located?
- What binary bit pattern will turn the eight LEDs on? What value will turn them off? How did you come up with these values?
- What is the address of the eight LEDs on the DSLMU Microcontroller Board?

Checkpoint 2

- What is the value of LED_port? Which line in the program sets this value?
- What does the directive `.equ Value1, 0b11111111` do?
- What do the lines starting with `do {` and ending with `} forever` do? What use are these lines?
- What is the meaning of `strb r5, [r4]`? What is the memory address used by this instruction?
- When single-stepping through your program, what registers change their value and where?

Checkpoint 3

- What is the difference between the code in Figure 5 and Figure 10? What is the purpose of this difference? In other words, why is this change needed?
- What is the address of main_loop?
- What problem occurs when trying to step through a full cycle of code (ie, so that the LEDs turn on, then turn off and finally turn on again)?
- How many steps are needed to complete a single cycle (ie, starting at main_loop and returning to that point)?
- How would you use the Multi-Step button to go through a single cycle of the program?
- What is the actual instruction used instead of `ldr r4, =LED_port`?
- What would happen if you changed the line with `.equ WaitVal` to `.equ WaitVal, 0`? Why?

Checkpoint 4

- What are the steps you need to take to convert a program written in C into an executable?
- What would happen if you changed the value of WaitVal to zero? Why?
- What is the starting address of your program? What is the address of the function main?

Checkpoint 5

- What assembly language statements are used to implement the C `for (i = 0; i < delayval; i++)` statement?
- What register is used to store the address LED_port?
- What register is used to hold the value of Value1? What line initialises this register?
- Where is the value WaitVal used in the assembly language code? Why do some statements appear to use the value 9984?
- What similarities and differences do you note between the hand-written code in Figure 10 and the code generated by the C compiler?