Computing for Engineers (EE108)
Laboratory 3– Iteration

Objectives: This laboratory aims to explore the use of iteration (loops) in C.

Learning outcomes Write programmes using while-loops and for-loops including the use of the break and continue statements

Lab instructions

1. The lab is for <u>individuals</u> working independently and must be completed during the lab session
2. Create a new sketch for each major section in the lab.
3. Before you leave the lab, call the lab demonstrator to check what you have done for all the sections (this is why separate sketches are important). *Anything not checked during lab will have to be assigned zero marks.*
4. Create a <u>single</u> plain text submission file (.txt) for the lab. Use a text editor (e.g. NotePad++) and do not use Microsoft word. Copy all the sketches you write and any other answers required for the lab into the submission text file. *Name the file "108_Lab3_firstname_surname.txt", using your actual name. Include your name and lab number at the top of the submission file and clearly label everything in the file. Submissions without names or clear identification of sections/sketches/answers will have to be marked down or (in the worst case) not marked at all.*

Marking for lab/assignment

This lab will be marked during lab sessions based on demonstrated behaviour and brief code inspection. It is essential that you get behaviour demonstrated and the code checked during the lab session.

For all code sections, marks will be deducted for bad communication and style (e.g. missing or mismatching comments, poor variable names, bad indentation, etc.), incorrect behaviour, or failure to follow the requirements of the lab question.

General marks are also lost if the submission document instructions are not followed.

1   Bar LED scan using loops

(10 marks)

Background: In this lab you will ultimately develop a <u>single solution</u> but you will build it up gradually in multiple parts as specified in the sections below.

The final sketch will wait for a click of button SW1 before starting. Then it will repeat a scan of the bar LEDs up to 10 times. The user can temporarily skip the full scan by pressing SW2 or can stop the scans altogether by pressing SW1. Once the user is not pressing SW1 any more, the sketch repeats (i.e. the loop function exits and is immediately called again).

This lab will test the use of nested loops, the break and continue statements.

Refer to notes "1.40 Iteration (loops)" as necessary. Some tutorial files have been supplied with the lab – you will be referred to them at the appropriate place in text below.

1.1   *Outer (repetitions) loop with blink and print – save as "Lab3_BarLedScan_1"*

Background: In this section you will create the initial sketch which creates the outline code structure but does not yet implement a scan of the bar LEDs. The basic idea is to prompt the user to click SW1. Then wait until SW1 is clicked before proceeding. Once it has been clicked, do the following 10 times: blink BAR_LED_1 once and then, as temporary placeholder code, print to serial. (Later we will replace the placeholder code with the code to scan the bar LEDs).

Requirements:

- Download the starter sketch "**Lab3_BarLedScan_Starter**". This defines some useful constants, sets up the serial and bar LEDs as needed in the setup function, and finally waits for button SW1 to be clicked in the loop function. You can copy from this starter sketch into your own sketch.
- In the loop function, once SW1 has been clicked, print out that it has been clicked. Then use an appropriate loop (e.g. while-loop or for-loop) to perform NUM_REPS (see starter sketch) repetitions of the following operations in order. <u>We'll call this loop the **repetitions loop**</u> and you should print that you are about to start it:
    - Print the current repetition number to serial, e.g. "rep 1", "rep 2" etc.
    - Blink BAR_LED_1 once (switch on for 100 ms; use the appropriate constant already provided in the starter code) – *Hint: remember BAR_LED_1_PIN refers to the pin for BAR_LED_1*
    - Print "Placeholder – bar LED scan would go here"
    - Delay for 0.5 seconds (use the constant already provided in the starter code)
- When the repetitions loop has terminated, the starter code prints "end of repetitions loop". Therefore the overall output for a single execution of the loop() function, should look like the following:
```
loop function starting -- click SW1 to begin
SW1 clicked, starting repetitions loop
Rep 1, Placeholder – bar LED scan would go here
Rep 2, Placeholder – bar LED scan would go here
…
End of repetitions loop
```

🖉 Copy the sketch into your answer document – only demonstrate your final solution for the lab

1.2 *Using continue to skip to the next iteration – save as "Lab3_BarLedScan_2"*

Background: Here you will modify the sketch from part 1 to skip to next iteration if SW2 is pressed using a continue statement.

Requirements:

- Inside the loop function, we'll refer to the for-loop or while-loop you wrote in part 1 as the repetitions loop. Inside the repetitions loop, after blinking the LED but before printing the placeholder text, check to see if SW2 is pressed in any way. If it is pressed, then skip to the next iteration of the repetitions loop without printing the placeholder text. Use a continue statement (see notes) to achieve this.
- Ensure that the variable which counts the number of repetitions is only incremented if the placeholder text is printed – if it is not printed (due to skipping) then the number of repetitions <u>must not</u> increase.

Hint: refer to Lab3_ButtonTutorial_2 to see how to check if SW2 is pressed in any way. Ensure that the variable which counts the number of repetitions is only incremented at the bottom of the repetitions loop to ensure that only "full" repetitions are counted.

🖉 Copy the sketch into your answer document – only demonstrate your final solution for the lab

1.3 *Nested loop for bar LED scan – save as "Lab3_BarLedScan_3"*

Background: Now you will replace the placeholder print with the code to scan the bar LEDs. This will require a nested loop which we'll refer to as the **scan loop**.

Requirements:

- Inside the repetitions loop, replace the placeholder print code with a nested loop to scan over all the bar LEDs from 2 to 5 blinking each one on and off in succession.
- Each Bar LED should be on for 100 ms and off for 50 ms before the next LED is switched on. (There are already constants defined with these numbers in the starter sketch.)

Hint: to light a particular bar LED we would usually use the constant for the relevant bar LED pin, e.g. to light bar LED 5 we could write `digitalWrite(BAR_LED_5_PIN, HIGH)`. *However this doesn't allow us to iterate over the different bar LEDs in a loop. Instead we must make use of the knowledge that the bar LEDs are connected to consecutive pins, so if we know the first pin we can calculate the pin for any subsequent LED. In particular, to light LED 5 we could also write* `digitalWrite(BAR_LED_1_PIN + 4, HIGH)` *or to a light a generic LED numbered by the variable* `ledNum` *we would write* `digitalWrite(BAR_LED_1_PIN + (ledNum-1), HIGH)`.

🖉 Copy the sketch into your answer document – only demonstrate your final solution for the lab

1.4 *Using break to exit the loops – save as "Lab3_BarLedScan_4"*

Background: We want to modify the programme so that the user can stop the bar LED scans immediately when the user presses SW1 in any way. To do this we need to break out of both the scan loop (nested loop) and the repetitions loop (outer loop).

Requirements:

- Inside the nested loop which scans the bar LEDs, check to see if SW1 is pressed in any way. (Use the same approach as you used for SW2 in part 1.2) and break out of the loop immediately if it is.
- Immediately following the bar LED scan loop, check to see if the bar LED scan terminated normally or because of a break statement.

  Hint: check the variable you were using in the controlling expression of the bar LED scan loop.

- If the bar LED scan terminated due to a break, then immediately break out of the repetitions loop also, as the user wants to completely stop the current and future scans.
- After breaking out of the repetitions loop like this, the user is may be still pressing SW1. Check if the user is pressing SW1 in any way and if so print a prompt to "release SW1" and wait until SW1 is released (see **Lab3_ButtonTutorial_2** for how you might do this).

✎ Copy the sketch into your answer document – only demonstrate your final solution for the lab

1.5 Extra credit - Customizable LED at which to end scan – save as "Lab3_BarLedScan_5"

Note: this section is for extra credit which may be added to improve your CA mark.

Background: Rather than scan from Bar LED 2 to Bar LED 5 and back every time, we want to make the LED at which the scan finishes customizable rather than being always being Bar LED 5.

Requirements:

- Inside the repetitions loop, read the digitized analogue value from the potentiometer by calling `analogRead(POT_PIN)` and saving it's return value. This number ranges from 0-1023. Dividing by 100 gives a number from 0 to 10 inclusive.
- Use the potentiometer value to decide the bar led number at which to end the scan.
- To ensure that we can always see at least a minimal scan pattern, the lowest Bar LED number allowed for the end of scan is Bar LED 3.

✎ Copy the sketch into your answer document – only demonstrate your final solution for the lab