

Objectives: This lab test assesses the use of variables, expressions, selection, iteration, and functions in C.

Lab Test instructions

1. This lab test is for individuals working independently and must be completed during the lab session
2. Create a new sketches as requested in the lab test.
3. Before you leave the lab, you must demonstrate the final status of your programme.
4. Create a single plain text submission file (.txt) for the lab. Use a plain text editor (e.g. NotePad++) to edit the .txt file and do not use Microsoft word or similar. Copy all the sketches you write and any other answers required for the lab into the submission text file. *Name the file "108_LabTest1_firstname_surname.txt". Include your name and lab number at the top of the file and clearly label everything in the file. Submissions without names or unclear sections/sketches/answers will be marked down or (in the worst case) not marked at all.*

Marking for lab test

The lab behaviour is marked during the lab session based on demonstrated operation. It is essential that you get the running sketch checked before leaving the lab session.

The code will be reviewed and graded based on the submission document.

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.), inappropriate use of global variables, unnecessary code repetition, incorrect behaviour, or failure to follow requirements of the lab question.

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

1 Morse code counting — save sketch as “LT1_MorseCodeCounting”

Background: The objective in this lab test is to create a C programme running on arduino that will flash the LEDs in a morse code pattern of short and long blinks (known as dots and dashes respectively) in response to appropriate button clicks. The morse code patterns will be limited to those representing the letters A to E. The currently selected letter can be changed by means of the buttons also.

You will be expected to make use of the EE108 library which you should have already downloaded and set up as part of your previous labs/assignments.

A good solution to the lab test may make use of concepts from all the notes and accompanying examples up to 1.60 Programme Design (but excluding FSMs).

In general you should try and build up the programme incrementally. First get the minimum possible functionality working (usually serial printing and clicks). Then gradually add in all requirement functionality until the programme is done.

Requirements:

- Your programme will need to keep track of the currently selected letter to be communicated (using morse). This letter must be in the range A-E inclusive and is initially A.

Hint: You should keep track of a number rather than a letter. You will need to pick a number to represent A, a different number to represent B, etc. Your code should make the correspondence between numbers and letters obvious (e.g. by appropriately named constants).

- Whenever switch SW1 is normally clicked

- print information about the detected click (e.g. normal click SW1)
- Print the currently selected letter to serial
- flash the LEDs with the correct morse code pattern (see below) for the selected letter.

- Whenever SW2 is normally clicked

- print information about the detected click
- advance to the next letter (e.g. if the current letter is C, the next is D). Wrap as necessary to stay in the valid range (i.e. if the current letter is E, the next is A).
- Print the selected letter to serial
- flash the LEDs with the correct morse code pattern (see below) for the selected letter.

- Whenever SW2 is long clicked or very long clicked

- print information about the detected click
- back up to the previous letter (e.g. if the current letter is C, the previous is B). Wrap as necessary to stay in the valid range (i.e. if the current letter is A, the previous is E).
- Print the selected letter to serial
- flash the LEDs with the correct morse code pattern (see below) for the selected letter.

- To flash the LEDs in the morse code pattern read the following:


- Morse code allows numbers and text to be transmitted using a sequence of short and long signals referred to as *dots* and *dashes*. These long and short signals may be sounds or flashes of light. In this programme you will flash the LEDs. Each single character or digit is represented by a sequence of dots and dashes in morse code. The patterns (sequences) corresponding to the letters A to E are as follows:

Letter	pattern
A	• —
B	— • • •
C	— • — •
D	— • •
E	•

- In morse code, the duration of a dot is one time unit, T , and the duration of a dash is $3T$. The duration of the space between dots or dashes in the pattern is also T . A value of T somewhere between 150 and 300 ms is about right for ensuring the dots are short, but still long enough to be easily detected (by a human).
- Although we would normally use just one light source for dots and dashes, to facilitate debugging and demonstration you must use different LEDs when showing a dot or dash. Specifically, flash bar LEDs 1 and 2 together to represent a dot and flash bar LEDs 6-10 together to represent a dash.

Hint: you may find it quicker initially just to blink a single LED for dots and a different single LED for dashes. You can extend this to the final requirements afterwards.

Ensure that you use functions, loops, decision statements, and other C concepts as appropriate to give a good quality clean solution that avoids unnecessary redundancy and repetition of code. Ensure your indentation, naming style, and comments ensure clear communication and readability of your solution and meet EE108 guidelines.

 Copy the sketch into your answer document and get the behaviour checked by a demonstrator.