**Submission Questions:**

18.

a) What operating system (including revision) did you use for your code development?

Ans: Windows 11

b) What compiler (including revision) did you use?

Ans: SDCC

c) What exactly (include name/revision if appropriate) did you use to build your code (what IDE,

make/makefile, or command line)?

Ans:STM32CubeIDE 1.13.2 and CodeBlocks:SDCC

d) Did you install and use any other software tools to complete your lab assignment?

Ans No.

e) Did you experience any problems with any of the software tools? If so, describe the problems.

Ans: No

**Pictures/Screenshots:**

1. Part 1: Terminal output after filling the data segment with a “U” character (55).

Data segment:

A screenshot of a computer

Description automatically generated

Downloading a file using “D” command in PAULMON2.

A screenshot of a computer

Description automatically generated

2.**Part2:** Creating buffers.

Creating buffers 0 and 1 with equal size with the input number which is between 32 and 4800 both inclusive.

The input number must be divisible by 16.

A screenshot of a computer

Description automatically generated

Terminal output when the input is a storage character.

A screenshot of a computer

Description automatically generated

Terminal output when the input is “+”, creating new buffers whose size must be between 20 and 400.

A screenshot of a computer

Description automatically generated

Terminal output depicting the Heap Report when input is “?”

A screenshot of a computer

Description automatically generated

Terminal output depicting the Heap Report when input is “=”

A screenshot of a computer

Description automatically generated

Terminal output 2 depicting the Heap Report when input is “@”

A screenshot of a computer

Description automatically generated

Terminal output 2 when invalid inputs are given.

A screenshot of a computer

Description automatically generated

3.Logic analyzer pictures of **Virtual Debug port**.

For a storage Character in Buffer 0:

A screenshot of a computer

Description automatically generated

For a “+” input:

A screenshot of a computer

Description automatically generated

For a “-“ input :

A screenshot of a computer

Description automatically generated

1. **Part3:** Terminal Pictures

For the required part, one GPIO pin is used to generate the PWM signal with a default 60% duty cycle.  
The duty cycle increases by 5% every time A is given as input.

The duty cycle decreases by 5% every time A is given as input.

The duty cycle increases/decreases by 10% every time the button is pressed.

When P is given as input, the current duty cycle is displayed.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Oscilloscope pictures for different duty cycles varied by giving “A”,”B” and button.

Duty cycle=10%

A screen shot of a graph

Description automatically generated

Duty cycle= 45%

A screen shot of a computer

Description automatically generated

Duty cycle=95%

A screen shot of a graph

Description automatically generated

5. Terminal Pictures:

For supplemental elements, demonstrating the PCA modes.

-Pulse Width Modulation

-High-Speed output

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Oscilloscope pictures:

For the supplementary part, one GPIO pin is configured for PWM duty cycle of 33%.

A screen shot of a graph

Description automatically generated

For the supplementary part, one pin is set for High-Speed Output.

A screen shot of a graph

Description automatically generated

For the supplementary part, when PWM is ON for one pin.

A screen shot of a graph

Description automatically generated

For the supplementary part, when PWM is OFF for one pin.

A screen shot of a graph

Description automatically generated

the supplementary part, output at ALE when one pin is set to the minimum peripheral clock frequency supported by the CKRL register.

A screen shot of a graph

Description automatically generated

For the supplementary part, output at ALE when one pin is set to the maximum peripheral clock frequency supported by the CKRL register.

A screen shot of a graph

Description automatically generated

For the optional challenges part, where the heap is created with 5600 bytes. Heap report after creating a buffer of 100 bytes.

A screenshot of a computer

Description automatically generated

For the optional challenges part, where the heap is created with 5600 bytes. Heap report after creating a buffer of 210 bytes.

A screenshot of a computer

Description automatically generated

For the optional challenges part, where the heap is created with 5600 bytes. Heap report after creating a buffer of 800 bytes.

A screenshot of a computer

Description automatically generated

Code snippet which I used to avoid button debouncing.

A computer screen shot of a computer code

Description automatically generated

**SIGNIFICANT LEARNINGS:**

* I have learned how to use Paulmon2 commands to modify the data segment and navigate through the program segment.
* I have learned how to run a specific program by jumping to its location using Paulmon2 commands.
* I have learned how to write a User Interface for dynamic memory allocation of buffers in heap memory and perform various functions on it.
* I have learned how to implement PWM using a GPIO pin and change the LED intensity using PWM.
* I have learned about different PCA modes while implementing the supplementary and challenge elements.