

General Assessment Guidelines for Engineering Software 3 - Assessment 2

Basic Guideline

1. Oral demonstration	20%
a. Demonstration of the code working with its full features.	10%
b. Questions/Answers	10%
2. Code submission.	80%
a. Code Quality: Reusability - Can it be used by others on its own? Adoption of modular approach, Good use of comments, proper alignment, etc.	40%
b. User and Programmer's Guide (Max. 6 pages): The user and Programmer's Guide should include: <ul style="list-style-type: none">1-2-page easy-to-read instructions on how to operate the traffic Light.Maximum 4 pages programmer's guide. It should include explanations of key code-sections/variables/methods and what they do.	20%
c. Lab book : Lab book for Lab 5 , Lab 6, and Assignments 2.	10%
d. Extra features : Any useful feature(s) added.	10%

Further Notes on Assessment

- Oral assessments are on Tuesday and Thursday sessions in week 11 (30th November and 02nd December 2021) as per your group.
- Each student will be given 15 minutes for the assessment. Please ensure you are ready for the assessment on your allocated time slot, otherwise, late penalty applies.
- The exact time schedules for your assessment will be put on Learn by Week 10
- For submission, the following must be put in a zipped folder and uploaded to Learn by 5pm on Monday, 29th November 2021 (Week 11): Please note that there is a separate submission box for report (user and programmer's guide).
 - All application source and header files you authored
 - All the other source and header files you have used, including the ones given to you
 - Basically, the **entire content** of the **src folder in SDK**
 - The hardware design (the **es3_hw_assmnt_2 folder**, only if you have changed anything in it). Note that if you have changed anything in hardware this has to be made clear in your user guide.

Extended Guidelines

Oral Demonstration

Please make sure you have your application already downloaded on the FPGA ready for demonstration to your assessor. Please also have your code displayed on a computer screen. The oral assessor will need to see a demonstration of both the basic functionalities and the extra features (if any) that you have developed. The assessor will then proceed to ask you questions about your implementation with reference to your code to check your understanding. The oral assessment marking criteria are as follows: The oral assessment marking criteria are as follows:

- Demonstration of the code working with different features (10%)
 - Demonstrating the basic functionalities specified in the assignment requirements.
 - Mention any specific Basys3 board features used and any functions written, or libraries used. Explain how these were used to build the application.
 - Demonstrating any distinct added features that set the application aside from the ones developed by others.

2. Explanation of code and Answer to questions (10%)

- Mention any specific Basys3 board features used and any functions written, or libraries used.
- Explain how these were used to build the application.
- Mention and explain any features that distinguish your system and set it aside.
- Answer any question(s) regarding the design, implementation/Code.

Online Submission Assessment Component

3. Code quality (40%)

a. Modularity

- Adoption of modular approach. The code should be divided into smaller and more manageable files.
- Good application folder structure. Good code structure should consider the functions, naming, spacing, etc. For example, using a name such as T.c for traffic light is not meaningful.

b. Code Commenting

- Good use of comments for improved code reusability and understanding by other Engineers for possible integration into larger applications. This enhances possible integration into larger applications. For assessment 2, we expect the following at the very least:

1. A general body of comments at the top of each source and header file detailing File Name, Project Name, Target Device/Platform, Tool Version, Your Name, Your Company, File Creation Date and Time, Description, and so on. See an example below. You can add more to these if it will aid the understanding of your code.

File Name: seg7_display.c

Project Name: 7 Segment Application

Target Device/Platform: Basys3 Board (with Microblaze processor on the Artix-7 FPGA)

Tool Version: Xilinx SDK 2015.2

Name: Your Name

Company: University of Edinburgh

Creation Date and Time: 08/11/2021; 03:15

Description: This source file contains functions used to drive the 7 segment display. There are three functions in this file. The **displayNumber()** function receives an unsigned 16-bit integer. It is used to assign the digit number and the value to be displayed per digit when the timer interrupt occurs. The **calculateDigits()** function is used to extract the digits (of which a maximum of four can be displayed) from the number to be displayed. The **hwTimerISR()** (which is defined in the xinterruptES3.c file) method is used to call the **displayDigit()** method, which selects the segments and displays the digits on the 7-segment display. All these functions are declared in the seg7_display.h header file.

2. It is not enough to simply add a comment to a code line or section, it is very important that comments are descriptive of the functionality implemented by the code. For example, comment 1 below is not good enough since that comment is already obvious from the code. However, comment 2 is better since it describes the functionality of the code.

// Check if pushBtnLeft is equal to 1	-	This is not good enough
// Check if the left push button has been pressed	-	This is better
If (pushBtnLeft == 1) {		
...		
}		

3. It is also a good practice to have sectional comments at the beginning of function definitions. If you do this properly, you do not necessarily have to comment each line of code in the body of the function.

4. User guide and Programmer's guide (20%)

Your user guide needs to include enough information for someone to understand the main functionality of your submitted code. Please note that your code submission needs to include any files you have authored including those provided to you and you have later changed. Where you have made changes, this has to be clearly stated in your user guide. Your user guide needs to include enough information for another engineer to load your code onto an FPGA and then run it. So you need to include clear instruction on how to use your application on the FPGA including key input and output definition. The following additional points should be noted for your user guide:

- a. Introduction and description of the application and its functionalities, including board inputs and outputs.
- b. Mention and explain any features that distinguish your system and set it aside.
- c. User guide to be contained on 1-2 pages of A4, with flexible layout, within reason.

The remaining 4 pages of the user's and programmer's guide should contain:

- d. Description of files, functions, and variables used.
- e. Functions used and their implementation.
- f. Extra features (if applicable) and their realization should be clearly highlighted and explained.
- g. Please add references if you have used external sources.
- h. 4 pages of A4 cannot be exceeded, anything beyond will be discarded and not marked.
- i. PDF version of Word-processed documents most welcome, equivalent word-processed forms acceptable.
- j. Any number of words, but font size not smaller than Times size 10 and line spacing not less than 1.5.
- k. You can include any of the following information to improve the readability of the guide, especially if it will make it easier for another engineer to understand, run, and use the submitted code(if you have included any of these, please, refer to it in the programmer guide):
 - o Program flow charts.
 - o Diagrams.
 - o Tables.
 - o Any other information that you think will make it easier for a user to understand, use and run your application.

5. Extra Features (10%)

- a. As Engineering Software 3 is more software-oriented, it is important that your extra features have good software components for them to be considered. Keep this in mind especially if you have added extra functionalities to the hardware platform.
- b. How realistic are the added features? Do they have any practical usefulness?

Grade Guide

A1	Excellent	90-100%
A2	Excellent	80-89%
A3	Excellent	70-79%
B	Very good	60-69%
C	Good	50-59%
D	Pass	40-49%
E	Marginal fail	30-39%
F	Clear fail	20-29%
G	Bad fail	10-19%
H	Bad fail	0-9%