CSSE3010 Project Milestone (External only)

Autonomous Remote Control of a Sorting Machine

(5% – 10 Marks)

Due: Your CSSE3010 Zoom Session in week 11, 2021

Last Updated: May 4, 2021

ECP Hurdle: Must be submitted, to pass.

(Closed Shoes MUST BE Worn in the labs)

1 Introduction

The project milestone is based on your code development progress for the project. The project milestone is intended for you to show how you will plan your implementation for the project and what signs of progress that you have made.

You must attend your CSSE3010 Zoom session in Week 11 and explain your code or demo to an assessor.

The project milestone is also an opportunity for you to get feedback on your current progress.

2 Code Development – 10 marks

Your code will be examined by a tutor. **The committed code does not need to function, be compiled or be complete.** The code presented must show what signs of progress that you have made. This may include partially filled in functions or pseudo code.

Your code must be in git by the due date and there must be atleast 3 commits, with descriptive commit messages by the due date.

2.1 CAG Simulator Progress – 2 marks

You must show what progress you have made on the CAG Simulator. This should involve showing what progress you have made on the `CAG_simulator .c/.h` files. You must demonstrate that you have sufficient knowledge of the Conway Game of Life (CGoL) algorithm.

2.2 CAG Display Progress – 2 marks

You must show what progress you have made on the CAG Display. This should involve showing what progress you have made on the `CAG_display .c/.h` files.

2.3 CAG Input Progress – 2 marks

You must show what progress you have made on either of the CAG Inputs: Keypad Grid or CAG Keypad Mnemonic or CAG joystick. This should involve showing what progress you have made on the related `.c/.h` files.

2.4 Code Quality – 2 marks

Any codd shown must conform the following:

- descriptive, readable naming of variables, functions, parameters, constants, macros, types
- constants and defines used instead of “magic values”
- modularity and functional decomposition instead of repeated code
- consistent white space (horizontal and vertical)
- conformation with the CSSE3010 Style Guide (on Blackboard)
- debugging and testing code neatly excluded from release compile for submission
- design justifications where appropriate
- documentation for functions
- comments for global variables, constants, macros, etc where appropriate

Note that you are encouraged to have debug/test code and extra functionality where appropriate. #define and build configurations should be used to enable or disable the debug/test code. Commented out code is not permitted.

2.5 Code Structure – 2 marks

You are required to reuse and expand upon your `mylib` library in this project. You should include additional files to your library. You must have project main.c code in your `stages/pf` folder. You must follow the mylib task and register guidelines. Peripherals used should have corresponding task and register files. Register files are to be used by the task files.

2.6 Deductions

Deductions to your Milestone total mark will be applied if:

1. You do not have sufficient or incoherent git messages.

2. Use a folder structure that does not adhere to the structure specified in the project specification.

3 Criterion

The milestone are marked according to the criterion outlined in the table below. If you fail to demonstrate sufficient understanding and functionality in the specified marking time you will not be allowed to repeat. You must pass the pre-marking checks before you are allowed to be marked. **All code assessed for the Milestone must be your own work.**

Pre Marking Checks

The following criteria **must** be met **before** you are allowed to demo.

Check	**P/F**
Your latest project and mylib code **must** be in git.	
At least 3 git commits have been made by the git due date.	
Your mylib and top comments are correctly filled out.	

Failure to meet pre-marking checks will mean that you are not allowed to be marked.

Code Development Criterion

You must be able to combine all design tasks, into the same file and demo all design tasks, without reprogramming your Nucleo. Note: You may be asked to make minor modifications to your code by the assessor, during your demo, which must be passed.

CAG Simulator Progress	
0	No or little progress made in the associated code files.
1	Some progress has been made in the associated code files or the CGoL algorithm is not understood.
2	Good progress has been made in the associated code files and the CGoL algorithm is well understood.
CAG Display Progress	
0	No or little progress made in the associated code files.
1	Some progress has been made in the associated code files.
2	Good progress has been made in the associated code files.
CAG Input Progress	
0	No or little progress made in the associated code files.
1	Some progress has been made in the associated code files.
2	Good progress has been made in the associated code files.
Code Quality	
0	Code does not show sufficient adherence to code style/quality guidelines.
1	Code does show some adherence to code style/quality guidelines but critical comments (e.g. top comment) are missing.
2	Code adheres well to code style/quality Guidelines.
Code Structure	
0	No or little adherance to code structure and mylib guidelines.
1	Some adherence to code structure and mylib guidelines.
2	Full adherence to code structure and mylib guidelines.
Deductions – Applied to Total Mark	
-2	No or insufficient or incoherent descriptive git commit messages.
-5	Code in folders not specified by the CAG project specification are used.

Student Name:

Student Number	**Mark (/10)**	**Marker**	**Date**