ENGD2103 SUMMATIVE ASSIGNMENT
CONCURRENT PROCESSING AND BARE-METAL PROJECT
2022-2023

Author: M A Oliver and T Allidina

Title: ***Summative Assignment***

1 Aims and objectives

This document provides the specification for the ENG2103 Summative coursework.

The purpose of this assessment is to give you experience of:-

- Controlling and running multiple processes on a single microcontroller.
- Controlling the digital pins of the ATMEGA328P microcontroller in a "bare-metal" fashion.

2 Academic Practice Offences

Any attempt to obtain a pass by unfair means is a very serious offence. Submissions will be checked for plagiarism, cases of copying the work of another student, supplying solutions to fellow students (whether intentional or not), or the use of other unacknowledged material will normally result in a fail grade of zero marks for the entire assignment. You should be aware that formal disciplinary action, as described in the University Regulations, may also be taken against individuals who fall foul. The disciplinary panel has the power to inflict serious penalties including expulsion from the university.

The bottom line is that you will need to submit your own work, and present your own work. You should know your own work intimately and be able to answer any question about it during the presentation.

3 The Task

In this assessed task, you will be required to write a number of code modules that are capable of running concurrently. In other words, they must each be capable of running in a non-blocking fashion. These modules will be:-

1. A “Heartbeat” indicator
2. A Traffic Lights controller
3. A Counter (using a shift-register and a 7-segment digital display)
4. An Orientation indicator based on an accelerometer
5. A switch-debounce module for detecting keypresses
6. A switch-debounce module for changing modes
7. A task scheduler

From a hardware perspective, you will have already constructed a circuit on breadboard. Using the parts from the three formative exercises, the circuit will contain:-

- Arduino Nano
- 6 LEDs (2 x red, 2 x yellow and 2 x green) for the Traffic Lights controller (with appropriate series resistors)
- 1 Seven Segment Common-Cathode digital display (with appropriate series resistors).
- 1 74HC595 Shift Register
- 2 Push-button Switches
- 1 GY-521 Module with MPU6050 Accelerometer / Gyroscope

You are advised to read this document carefully before commencing.

4 Traffic Lights Module

The Traffic Lights controller will control a simple 2-way crossroads.

- Traffic Lights Set 1 control the North-South lanes.
- Traffic Lights Set 2 control the East-West lanes.

On reset, the Traffic Lights will start with both sets of lights having equal priority (i.e. both sets get green for the same duration). It will be possible to configure the priority with the following three options:

- Equal Priority,
- SET 1 Priority (i.e. Set 1 has a longer green period than Set 2), or
- SET 2 Priority (i.e. Set 2 has a longer green period than Set 1).

The table below shows the timings for the traffic lights.

STATE	**LIGHTS SET 1**	**LIGHTS SET 2**	**TIME (s) Equal Priority**	**TIME (s) SET 1 Priority**	**TIME (s) SET 2 Priority**
0	🟡	🟡	2	2	2
1	🔴	🔴	1	1	1
2	🔴 🟡	🔴	1	1	1
3	🟢	🔴	**6**	**8**	**4**
4	🟡	🔴	2	2	2
5	🔴	🔴	1	1	1
6	🔴	🔴 🟡	1	1	1
7	🔴	🟢	**6**	**4**	**8**
8	🔴	🟡	2	2	2

The sequence is slightly more involved than the traffic lights controller encountered in the formative assignment. Also note the differences in timings.

State 0 is a safe state that will bring all traffic to order. This state will be run just once following each time the Traffic Lights controller is started. Once State 0 has been executed, the system will continually cycle-through States 1 to 8.

If the Traffic Lights priority gets changed, it should be applied during the Amber-to-Green transition, i.e. at the very start of STATE 3 or STATE 7. (The mechanism for changing the state will be discussed later).

Whenever the Traffic Lights controller is not running, its LEDs will need to be switched off.

IMPORTANT
The LEDs representing the traffic lights will need to be driven via a bare-metal approach. **Therefore the use of `digitalWrite()` and `pinMode()` is prohibited for this exercise.**

5 Heartbeat Module

The Decimal Point (DP) segment of the 7-segment digital display needs to be programmed to blink on and off as a 'heartbeat' to indicate the system is active. The heartbeat requirement is that the DP will be repeatedly switched on for 270ms and switched off for 270ms.

Note: The 7-segment digital display will be driven via the shift-register, which in turn will be driven in a 'bare-metal' fashion by the Arduino. You will need to drive, i.e. bit-bang, the shift-register's LATCH, CLOCK and DATA lines yourself in a 'bare-metal' fashion without the need of external libraries, etc. Therefore the use of `shiftOut()`, `digitalWrite()` and `pinMode()` is prohibited for this exercise.

6 The Pushbutton Switch Modules

In this exercise, there will be two pushbutton switches, SW1 and SW2:-

- SW1 will be used for counting keypresses
- SW2 will be used for changing mode

In both cases a non-blocking debounce routine is required. For this exercise, a slightly less aggressive debounce of 300ms is required.

The pushbutton switch will be set-up and interrogated in a 'bare-metal' fashion by the Arduino. Therefore the use of `digitalWrite()`, `pinMode()` and `digitalRead()` are prohibited for this exercise.

7 Counter Module

When the Counter module is running, a 300ms debounced keypress of SW1 will cause the system to increment a counter variable; only the least significant nibble of the counter variable will be displayed on the 7-segment display as a hexadecimal digit. During initialization, the counter shall be initialized to 0. Whenever the Counter module is restarted, the counter shall resume counting from where it left off.

8 Accelerometer Module

When the Accelerometer module is running, the 7-segment LED display should display the orientation of the board, as per the previous exercise, namely:

- 'F' – for flat
- 'b' – for flat, base-up
- 'L' – for landscape
- 'U' – for upside-down
- 'l' – (lower-case 'L') for portrait (rotated to the left)
- 'r' – (lower case 'R') for portrait (rotated to the right)

All these characters need to be displayed so they can be read normally according to the current orientation of the breadboard.

This will also control the priority Traffic Lights system (whether or not the Traffic Lights module is running).

If the board is oriented **Flat**, **Flat Base-up**, **Landscape**, or **Landscape Upside-down**, both sets of the Traffic Lights have **Equal Priority**.

If the board is oriented **Portrait-Left**, then the Traffic Lights **Set 1** gets priority.

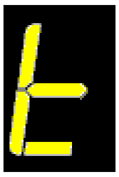
If the board is oriented **Portrait-Right**, then the Traffic Lights **Set 2** gets priority.

9 Modes

As mentioned earlier, switch SW2 is used to cycle between modes. The software will need to be designed to start in Mode 1. A press of SW2 advances to the next mode. When in Mode 5, a press of SW2 returns the mode back to Mode 1.

MODE	Traffic Lights Running?	Counter Module Running?	Accelerometer Module Running?	Heartbeat Module Running?	SW1 Module Running?	SW2 Module Running?
1(*)	YES	NO	NO	YES	NO	YES
2	NO	YES	NO	YES	YES	YES
3	NO	NO	YES	YES	NO	YES
4	YES	YES	NO	YES	YES	YES
5	YES	NO	YES	YES	NO	YES

(*) In Mode 1, (when the Counter AND Accelerometer modules are NOT running, the 7-segment LED display should display the lower-case letter 't'.



10 Submission

The deadline for submission is 14:00 on Friday 16th December (Week 11).

For the submission process, you are required to upload two items to a repository on Blackboard:-

- Photograph of your breadboard in .JPG or .PNG format. **Proprietary formats such as Apple's .HEIC format will be marked as zero.**
- Zipped archive of your project. **This must be a ZIP file having a .zip extension.** (.RAR format files cannot be opened on University computers and will attract a mark of zero.)

Demonstrations of your work will take place during the scheduled laboratory sessions in Week 15. You may present your demonstration earlier in Week 14, should you wish.

- **Attending your timetabled laboratory session is an absolute requirement not an option. Cherry-picking an alternative session will result in a mark of zero.**
- During the assessment, you will be observed downloading your code from the Blackboard repository and uploading it to your board. This promotes fairness as this eliminates any advantage of presenting at a later session.

This laboratory exercise will contribute towards 100% of your final module mark. You must work individually. Do not attempt to present somebody else’s work as yours as this could be considered as an Academic Practice Offence.

11 Marking Scheme

Submission Marking Scheme

CATEGORY	SUB-CATEGORY	SCORE
Documentation (10)	Breadboard Layout / 5 Code commenting / 5	/ 10
Structure (20)	Modularized code / 5 Use of HAL / 5 Encapsulated timings / 5 Inherited encapsulation / 5	/ 20
Bare Metal (10)	GPIO with D0 and D1 properly taken care of / 6 Shift register driver / 4	/ 10
Concurrency (20)	Demonstrable (0 for severe use of `delay()`) / 7 Scheduler / 8 Robustness / 5	/ 20
Mode 1 (5)	Correct sequencing / 3 Correct timing / 2	/ 5
Mode 2 (5)	Correct sequencing / 3 Correct character set / 2	/ 5
Mode 3 (5)	Correct orientation / 3 Correct character set / 2	/ 5
Mode 4 (8)	Mode 1 & 2 together / 4 No race-conditions / 4	/ 8
Mode 5 (8)	Mode 1 & 3 together / 3 Priority timings / 3 No race-conditions / 2	/ 8
Debouncers (5)	SW1 debounced / 1 SW2 debounced / 1 Compliance / 1 Robust timing / 2	/ 5
Heartbeat (4)	Functional / 2 Timing / 1 No race-conditions / 1	/ 4

Demo Mark Scheme

CATEGORY	**SUB-CATEGORY**	**SCORE**
Demo (30)	Organization / 5 Q&A and ability to show ownership of code. / 25	/ 30
Traffic Lights (10)	Correct sequencing and timing / 5 Bare metal implementation / 5	/ 10
Counter (10)	Correct sequencing and character set / 5 Bare metal implementation / 5	/ 10
Accelerometer (10)	Correct orientation and character set / 5 Boundary conditions / 5	/ 10
Mode 1 (5)	Traffic lights and heartbeat working together / 5	/ 5
Mode 2 (5)	Counter and heartbeat working together / 5	/ 5
Mode 3 (5)	Accelerometer and heartbeat working together / 5	/ 5
Mode 4 (5)	Traffic lights, counter & heartbeat working together / 5	/ 5
Mode 5 (10)	Traffic lights, accelerometer & heartbeat working together / 5 Priority timings / 5	/ 10
Debouncers (5)	Both switches debounced / 5	/ 5
Heartbeat (5)	To specification / 3 No race conditions / 2	/ 5

Faculty of Computing, Engineering & Media (CEM)
Coursework Brief 2022/23

Module name:	**Embedded Systems Fundamentals**	
Module code:	**ENGD2103**	
Title of the Assessment:	**Concurrent Processing and Bare Metal Project**	
This coursework item is: *(delete as appropriate)*		Summative
This summative coursework will be marked anonymously:		No

The learning outcomes that are assessed by this coursework are:

1. Bare metal operation
2. Concurrency
3. Structured programming

This coursework is:	Individual	

This coursework constitutes 100%of the overall module mark.

Date Set: 07/11/2022	
Date & Time Due (the deadline):	**16/12/2022 at 14.00**
In accordance with the University Assessment and Feedback Policy, your marked coursework and feedback will be available to you on:	**16/01/2022**

You should normally receive feedback on your coursework **no later than 15 University working days after the formal hand-in date,** provided that you have met the submission deadline

If for any reason this is not forthcoming by the due date your module leader will let you know why and when it can be expected. The Associate Professor Student Experience (CEMstudentexperience@dmu.ac.uk) should be informed of any issues relating to the return of marked coursework and feedback.

When completed you are required to submit your coursework via:

1. Blackboard submission journal

If you need any support or advice on completing this coursework please visit the Student Matters tab on the CEM Blackboard shell.

Late submission of coursework policy:

Late submissions will be processed in accordance with current University regulations.

Please check the regulations carefully to determine what late submission period is allowed for your programme.

Academic Offences and Bad Academic Practices:

Please ensure you read the section entitled "Academic Offences and Bad Academic Practice" in the module handbook or the relevant sections in this link: BaseCamp Link: Overview: Assessment and Good Academic Practices

Tasks to be undertaken: Write a single well-structured code project that implements the specified five modes. Code must be capable of concurrent operation. I/O access must be bare-metal.
Deliverables to be submitted for assessment: • **Photograph of breadboard. Must be .PNG or .JPG (Other formats such as Apple's .HEIC format *MUST NOT BE USED* and will attract a mark of zero).** • **ZIP file containing the code environment. Must be a .ZIP file. (Other formats such as .RAR *MUST NOT BE USED* as they cannot be opened on university computers and will attract a mark of zero).**
How the work will be marked: • **Blackboard submission marked offline.** • **Demo will be marked face-to-face in timetabled lab sessions.**

Module leader/tutor name:	**Dr Michael Oliver**
Contact details:	**michael.oliver@dmu.ac.uk**

Should you need any further information or advice please email
cemadvicecentre@dmu.ac.uk