## Criterion 2a: Submission requirements

To obtain a mark out of 20 for Criterion 2b, your submission must meet the following:

1. Source code uploads and builds successfully to Gradescope, as indicated by the autograder output.
2. Your code does not link any precompiled code provided by the teaching team during learning activities throughout the semester (i.e., qutyio.o or qutyserial.o).

## Criterion 2b: Demonstrated microcontroller programming skills (20 marks)

This Criterion is assessed by Gradescope autograder.

1. Gameplay as specified on the Assignment task specification **Section A**:
    - Reading button presses.
    - Operation of the 7-segment display.
    - Validating the user entered sequence with Simon's.
    - For each sequence step, display of segments and sounding tone for half the **playback delay**, followed by the display off and the buzzer silent for half the playback delay.
    - Display of SUCCESS/FAIL patterns and the user's score.
2. Sequence generation as specified on the Assignment task specification **Section B**.
3. Playback delay as specified on the Assignment task specification **Section C**:
    - Playback delay time should be correct to within ±2%.
    - Playback delay adjustment via potentiometer functional.
4. Playback frequency as specified on the Assignment task specification **Section D**:
    - Playback frequency of four tones correct to within ±2%, and with 50% duty cycle accurate to ±2%.
5. UART functionality as specified on the Assignment task specification **Section E**:
    - Acceptance of commands through the UART.

- Gameplay using key presses instead of QUTy pushbuttons.
- Incrementing and decrementing the playback frequencies.
- Implementation of the RESET function.
- Display of strings in the serial monitor as specified.
- Provision of new SEED.
- Display, storage, and update of the high score table.

**Criterion 2c: Demonstrated conformance to good coding and embedded systems programming conventions and best practices (10 marks)**

This Criterion is marked by tutors.

1. Code, comments, and logical structure
   - Comments are included throughout your source code which document the intent of your code.
   - Variables scoped appropriately.
   - Tidy program layout with appropriate indenting
   - Your program is divided into logical functional modules using multiple source and header files.
2. Use of pre-defined bitmasks and group configurations
   - All predefined bitmasks, group configurations etc. (as provided by <avr/io.h>) used throughout your programme are appropriate within the context where they are used.
3. Integer types and appropriate arithmetic operations
   - All integers used throughout your program are declared using the type aliases defined in <stdint.h>.
   - No integer division.
   - No floating-point literals or operations are used within your program.
4. State machine usage
   - Functionality realised through the implementation of a state machine using enumerated type(s) and a switch-case statement(s).
   - Default cases are defined in all state machines to handle invalid states.
5. Efficient coding practices
   - Unnecessary reads from, or writes to, registers are eliminated (i.e., read-modify-write patterns are avoided where possible, redundant writes are eliminated).
   - No unnecessary loops or conditionals.
   - Sequence steps are generated from the Linear Feedback Shift Register on-the-fly as required, and not pre-generated and stored in an array.

| | | | | | | |
|---|---|---|---|---|---|---|
| Percent | 100 – 85 | 84 – 75 | 74 – 65 | 64 – 50 | 49 – 40 | 39 – 0 |
| Grade | 7 | 6 | 5 | 4 | 3 | 2/1 |

## 2b. Demonstrated microcontroller programming skills (20 marks)

| | Attribute demonstrated at the highest level of practice, with no or few minor errors. | Attribute demonstrated at an excellent level, but there may be several minor errors. | Attribute demonstrated at a good level, but there may be several minor and at least one significant error. | Attribute demonstrated at an acceptable level, but with several significant errors. | Inadequate demonstration of attribute. | Little to no attempt to demonstrate the attribute. |
|---|---|---|---|---|---|---|
| • Gameplay (5 marks)<br>○ Reading button presses<br>○ 7-seg display operation<br>○ Sequence validation | Debounced button presses | | | | | |
| • Sequence generation (3 marks) | | | | | | |
| • Playback delay (3 marks)<br>○ Delay correct<br>○ Delay update with potentiometer | Delay correct to ±2% | | | Delay correct to ±5% | | |
| • Playback frequency (3 marks)<br>○ Tone frequency correct<br>○ 50% duty cycle correct | Frequency and duty cycle correct to ±2% | | | Frequency and duty cycle correct to ±5% | | |
| • UART functionality (6 marks)<br>○ UART configuration and use<br>○ Gameplay using key presses<br>○ Inc / dec playback frequencies<br>○ Reset function<br>○ Serial display of strings<br>○ New seed<br>○ High score table | | | | | | |

| Percent | 100 – 85 | 84 – 75 | 74 – 65 | 64 – 50 | 49 – 40 | 39 – 0 |
|---|---|---|---|---|---|---|
| Grade | 7 | 6 | 5 | 4 | 3 | 2/1 |

| **2c. Demonstrated conformance to good coding and embedded systems programming conventions and best practices** | | | | | | **(10 marks)** |
|---|---|---|---|---|---|---|
| | Attribute demonstrated at the highest level of practice, with no or few minor errors. | Attribute demonstrated at an excellent level, but there may be several minor errors. | Attribute demonstrated at a good level, but there may be several minor and at least one significant error. | Attribute demonstrated at an acceptable level, but with several significant errors. | Inadequate demonstration of attribute. | Little to no attempt to demonstrate the attribute. |
| • Code, comments, and logical structure. (3 marks) | | | | | | |
| • Use of pre-defined bitmasks and group configurations. (1 mark) | | | | | | |
| • Integer types and appropriate arithmetic operations (2 marks) | | | | | | |
| • State machine usage (2 marks) | | | | | | |
| • Efficient coding practices (2 marks) | | | | | | |