

**CMPE1250 – Practical Assessment #1**

**Embedded Systems Fundamentals**

**Computer Engineering Technology**

Date: October 2024 (1241)

Time allowed: 1 Hour, 50 Minutes

Materials Permitted: Unlimited Code, Electronic Files, Paper

Student Name

The following policy is an addition to the Exam and Coursework Regulations and Procedures as outlined in the NAIT Electronics/Computer Engineering handbook. *It is the student’s responsibility to be aware of the information contained in the handbook in addition to the following policy.*

**Use of Electronic Devices during Exams**

Use of electronic devices other than the approved calculator as outlined in the NAIT Electronics/Computer Engineering Student Guide or necessary medical devices will not be permitted during exams. The use of a calculator will be permitted only in exams where such use is considered appropriate to the content of the exam. For the purpose of this policy, exams are defined as any measurement tool used to determine learning taking place in classrooms, labs or in the field. These include but are not limited to lab exams, quizzes, midterm exams, final exams, etc.

*Definition of electronic devices: Including but not limited to tablets, MP3 players, pagers, cell phones, smart watches, radios, electronic daytimers, and laptop computers.*

The exam invigilator will ensure proper implementation of this policy and any decisions made will be final. Any challenges to a decision will need to be addressed in writing to the Assistant Program Head of the appropriate program.

Maximum Mark: 100

***You will create a project from the template repository for this exam. You will push the project and libraries, if applicable, at the conclusion of the exam. However, you can make partial commits for convenience.***

# Preparation [10]

Follow the procedure discussed in class by your instructor to create a project. Include initialization and support for your switches and indicator LEDs. You do not need to use the PLL library (but you can if want to do it). You will get marks deducted if you forget to add your name and /or date to the comments header section in *main.c*.

# Part A [40]

Create suitable code to create a modulus 8 counter (0-7) that increments approximately every 400[ms] using a blocking delay as described in class. Display the counter in binary using the LEDs. For instance, when the counter is zero, all LEDS are OFF and when it is 7, all LEDs are ON. The counter should reload automatically (run always). The table that describes the counter in decimal and binary is as follows.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DECIMAL** | **BINARY** | **RED** | **YELLOW** | **GREEN** |
| 0 | 000 | OFF | OFF | OFF |
| 1 | 001 | OFF | OFF | ON |
| 2 | 010 | OFF | ON | OFF |
| 3 | 011 | OFF | ON | ON |
| 4 | 100 | ON | OFF | OFF |
| 5 | 101 | ON | OFF | ON |
| 6 | 110 | ON | ON | OFF |
| 7 | 111 | ON | ON | ON |

# Part B [30]

Add suitable code such that when the **UP** switch is pushed, the counter update time changes to **every 200[ms].** You should now see the counter updating at double the speed. Pressing the **DOWN** switch should set the update time back to every **400[ms]**. Please note that since a blocking delay is being used, the button should be held pressed for about half a second for the system to react. This is normal due to the implementation used (delay).

# Part C [20]

# Add a feature such that when any two (and ONLY two) of the switches are pushed, the counter update stops. Pressing any three (and ONLY three) at this point should resume the counter update again.

# Ensure the project is committed and pushed to the repository as well as your libraries, if used.