EEET2481

Assessment 3 (40 marks)

Instructions and Notes:

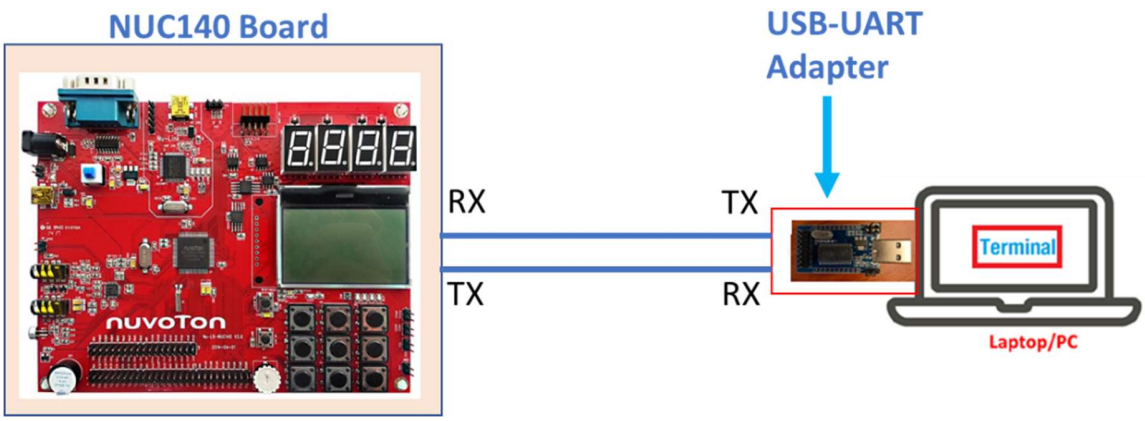
1. This assessment must be completed by a group of two students. Both students are expected to collaborate and contribute equally to successfully complete the three exercises. Contributions will be assessed during the demonstration. Students who do not contribute meaningfully will receive a lower mark, and no contribution will result in zero marks.
2. Each team will participate in a demonstration session to present their solutions to the two exercises to the lecturer. During this session, individual students will be required to:
    - Questions about the two exercises, including explaining how they solved them.
    - Complete an additional exercise related to the studied topics.
    - Perform a measurement task involving OSC or VB.

Submission

1. Each exercise requires a complete Keil MDK project that can be rebuilt and executed.
2. Capture 4–5 waveforms depicting the transmission of 4–5 characters through UART and annotate them to verify accurate transmission of the intended characters (e.g., ‘H’, ‘S’, ‘3’) in accordance with the configured UART data framework and baud rate.
3. In exercise 2, you are required to capture a waveform that shows the 1/10 second delay generated by a Timer interrupt of the NUC140 MCU.
4. Each group may provide supporting documents, including a state diagram and additional materials that clarify their design and demonstrate their understanding.

Exercise 1 – UART (8 marks)

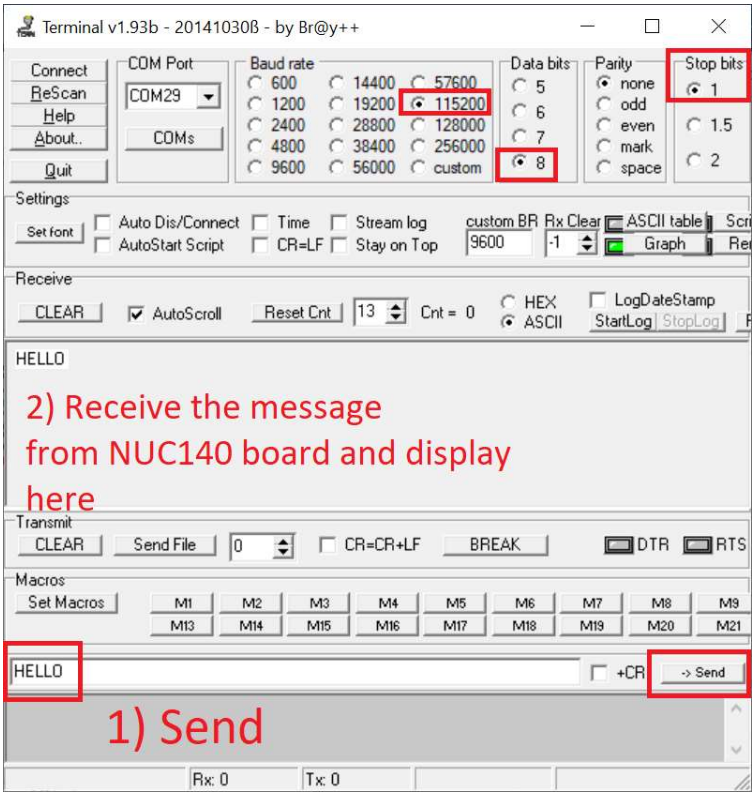
In this exercise you will be develop a program that will run on the NUC140 board, to send and receive data between the NUC140 Board and your laptop. From laptop side, you need an additional USB-UART adapter together with Terminal software to facilitate the communication. The diagram of the system can be simplified as follows:



The program has the following specifications:

a. Clock frequency of the CPU is at 50 MHz
b. UART channel 0 specification:
    - UART clock source is 22.1184 MHz
    - UART0 channel is used to transmit data and receive data.
    - Data packet: 1 start bit + 8 data bit + no parity bit + 1 stop bit
    - Baud rate: 115200 bps

After the reset, the board is standing by to wait for data to send from laptop. You will send the message "**HELLO, THIS IS <*student 1 name*>, <*student 2 name*>, <*student 3 name*>, FROM TEAM <*team name*>**" from Terminal to the NUC140 board. The board will receive the message and send it back to the laptop. If the setting is successful implemented, you can see the message re-appear in the RX box of Terminal.



For this exercise 1, you must use UART Interrupts for both TX and RX.

For some computers you might need to install the CH341 driver:

https://www.driverscape.com/download/usb-serial-ch340

Exercise 2 – Stopwatch (17 marks)

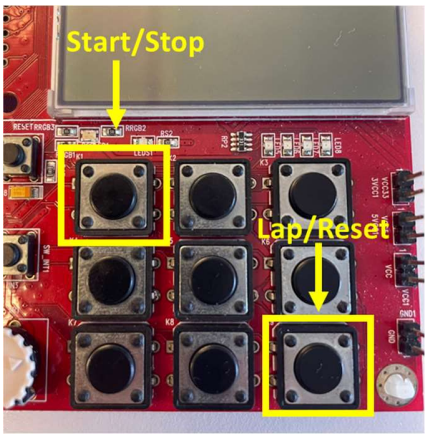
2.1 Description

You will design a **stopwatch system** and implement it on the NUC140 MCU board. If you are not familiar with the stopwatch system, you can try the stopwatch app on your mobile phone to get a sense of how it should work. The system functions as follows:

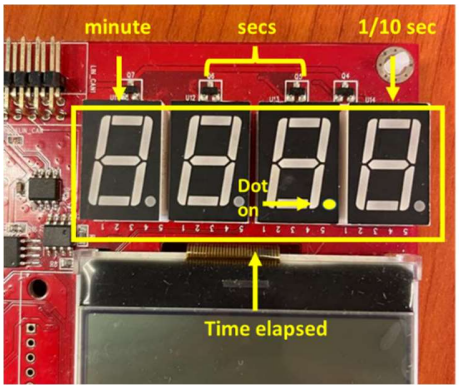
a. After reset the system is in *Idle* mode, you will indicate this by turning on only **LED5** on the board.

b. When we press Key **K1 (Start/Stop** key**)** from the board, the stopwatch will start counting (we call this *Count* mode):

- **LED5** is off.
- **LED6** is on.
- The precision of the timer is **1/10 second.** To implement the counter, you must use **Timer0 interrupt**.



- The LED display will display the time elapsed on the 4 x 7-segment LEDs of the board as follows:

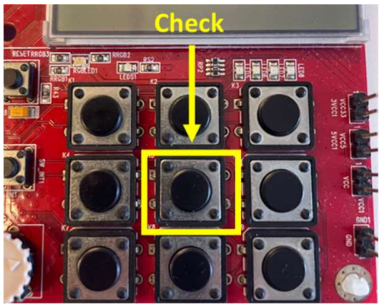


- While the counter runs, if you press **Start/Stop** key, the counter will be paused (we call this *Pause* mode) and resumed to *Count* mode if you press **Start/Stop** key one more time. During pause, the 7-segment will freeze the display time.
  - When the system is in *Pause* mode, the **LED6** is off, **LED7** is ON

c. During *Count* mode, every time you press **Lap/Reset** key (**K9**), the lap time is recorded. You will be record **maximum 5 lap times** in your system. If you press *Lap/Reset* key more than 5 times, then the new lap times will overwrite the previous ones. These recorded lap times can be displayed later (see e)

d. During the *Pause* mode, if you press **Lap/Reset** key, the counter will be reset, and the system will go back to *Idle* mode

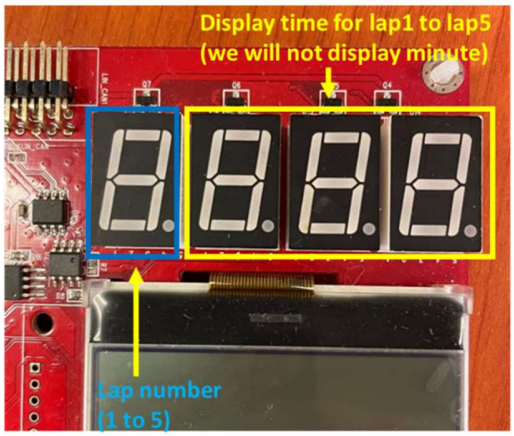
e. During the *Pause* mode, if you press **Check** key (**K5**), the system will go to *Check* mode to display all the lap times that are recorded, starting with the Lap 1



The display will look like following:

- The lap time will be displayed with 3 digits only (we will ignore minute in this mode)
- The first digit will display what lap number you are checking.

- If you press the **Rotate** key (GPB15), it will rotate between the laps as in Lap1->2->3->4->5->1. You must use external interrupt to control the Rotate key.



If we press the **Check** key again, it will exit *Check* mode and go back to *Pause* mode

2.2 Requirements and notes

Your final prototype must meet all the functions described in section 2.1. In addition:

- Pay attention to the compulsory requirements for each component.
- You must apply the Finite State Machine (FSM) approach in this exercise.
- There are additional tutorial notes on how to use Key Matrix and how to design system using FSM. You can find their notes and code on Canvas page of the Assessment 3.
- You can add debouncing code to your keys if the performance is not stable.
- You are permitted to utilise other functions covered in class or used in tutorials, such as CLK_SysTickDelay and GPIO_SetMode.
- It is strongly recommended to modularize your final code by utilizing functions whenever possible.
- For key functions (Timers, Interrupts, etc.), you are required to write your own functions using the register-based approach. Using built-in functions for these functions is not permitted.

Demonstration (15 marks)

Each group will be assigned a demonstration session during Week 12–Week 13 (exact schedule to be announced later) to discuss their work. The session will have a maximum duration of 20–30 minutes and include the following tasks:

- Present Exercise 1: One student will present the function of the system described in Exercise 1 using the NUC140 board.
- Demonstrate Exercise 2: Both team members must demonstrate the operation of the system described in Exercise 2 on the NUC140 board. Coordination and fair distribution of responsibilities are essential; insufficient contribution from a team member may result in a lower mark. The flow
    - Setting up the oscilloscope for measurements.
    - Briefly introducing your setup.
    - Explaining your algorithm and code.
    - Comparing your code's functionality with the project requirements.
    - Allocating the majority of the demonstration time to Q&A, where each student may be asked about any aspect of the project.
- Answer Course Questions: Respond to one question related to a course topic, which may involve coding.
- Complete a Measurement Task: Perform a measurement task linked to one of the exercises (Exercise 1 or Exercise 2) or an independent measurement task. Coding is not required for this task.