DESN2000 (Computer Engineering) 2024 T2
Project Brief: Custom Clock with Laboratory Timer Functions

1. Project Description

In a biochemistry laboratory, technicians need to manage various processes, such as chemical reactions, and keep track of their durations. Each laboratory technician deals with a consistent set of tasks (one to four tasks) over several months. Currently, each technician uses multiple individual clocks for this, which is not very productive. They have asked us to design a custom clock that integrates **both** specialised laboratory timer functions and standard clock features.

img: https://pxhere.com/en/photo/1293256
(CC0 Public Domain)

Below is what the lab technicians have so far provided:

Laboratory Timer Functionality:

Laboratory technicians should be able to configure the clock to start up to four processes either in parallel or sequentially. Each process has a different duration (e.g., PCR reaction for 5 minutes, shearing for 10 minutes, adaptor ligation for 3 minutes). Identical tasks are never done simultaneously. After completing a project, the clock should allow reconfiguration for a different set of tasks. The user should be able to configure 1-4 timers, each associated with a label and countdown time. Each timer should be associated with a user-defined label during configuration (e.g., PCR reaction). Settings should be remembered in memory/saved for future use.

Requirements:

- Users should be able to initiate a timer by pressing a designated button.
- The timers should be clearly displayed on the LCD screen appropriately.

- When the timer reaches its end a sound notification must be provided.
- Ensure each timer has a unique notification (e.g., different sounds) to distinguish it from others.

Standard clock:

There should be an easy way to switch the clock between the standard clock and the laboratory timers. Note that switching between the two should not clear any existing timers. The standard clock features that must be supported are the time of the day, alarm, countdown timer and stopwatch (one instance of each is sufficient).

This is all the information provided by the laboratory technicians. While it may not be entirely clear, such is the nature of a design project. If you have any doubts, please post a question on the ED forum for the project, so that we can reach out to the laboratory personnel for clarification. If there are reasonable assumptions you can make, feel free to do so - but you should **state** these assumptions when presenting/marking.

The clock firmware is to be written in C and/or assembly language. You will be given a development board with an STM32 ARM microcontroller during the course. To build the clock, you can decide which components of the development board will map to which components in the clock.

Once you've covered the basic functions of the clock described above, you must add features to make it **user-friendly** and **attractive**. For instance, you can use the stepper motor on the board as a clock hand; make the user interface (e.g., the interaction with the menu) determine less than optimal lighting using the LDR and illuminate some LEDs; and provide the capability to interface with a computer through the USART. You do not have to limit yourself to these examples, it is totally up to you to use your creativity, as this is a design project.

Note that the above description is **deliberately made open-ended** as this is a design project which is expected to stimulate your thinking and creativity. After you spend the first few weeks drafting and brain-storming your design, we can provide a detailed specification for you to adhere to (only for those who want to, if many of you request).

2. Teamwork

This is a group project, and you are required to work in a group of **three** students. Your tutors will facilitate weekly sessions, where your team will work on various aspects of the project. It is important to work efficiently together as a team to achieve timely deliverables and to successfully complete this project. Each student will have access to their own STM32 ARM development board to ease collaboration and testing. Students are free to collaborate via any means they prefer but are expected to meet with tutors during face-to-face weekly labs/workshops.

3. Deliverables

The whole course and its assessments revolve around the project. The course outline contains how the marks will be distributed, and you will be provided with assessment guides in due course.

The final product is the clock and includes the source code, user guide and developer guide. Your tutor should be able to compile your source code and flash it onto a board and use it as instructed on the user manual. Your tutor will also go through your code with the help of developer guide and attempt to understand the implementation.

User guide: The user guide is a critical component of any project, as it provides instructions for the user to set up, operate, and troubleshoot independently, without assistance from the development team. The guide should be comprehensive, clear, and easy to follow, and should include step-by-step instructions for all key functions. The guide should be presented in PDF format, but you may also include additional resources, such as videos or diagrams, to enhance clarity and ease of use.

Source code: To ensure that your program is easily maintainable and extendable, it's crucial to structure it well and provide adequate comments. During development, you are expected to use a GitHub repository, which allows version control and collaboration. You should follow standard git practices like descriptive and meaningful commit messages, branch to isolate changes etc. If you need help setting up your own GitHub repository, feel free to ask a tutor either in person or on the ED forum.

Developer guide: The developer guide is an essential document that describes the design of your code for future developers who may need to extend, optimise, or fix bugs in the codebase. It should clearly indicate the mapping of components in the source code to the design components. You are expected to create the developer guide in markdown format and store it in the same GitHub repository as your code.