EEET2505 - Introduction to Embedded Systems

Assignment (35%)

# Objectives

For this assignment:

* You will work on exercises on Timer and Interrupts to generate certain waveforms.
* Then, you will work on a mini project on building a Digital Clock system on Arduino and other components. This mini project requires you to:
  + Analyse requirements and go through a design process of an embedded system, both in software and hardware using AVR microcontroller, including General Purpose Input/output (GPIO), Timers/Counters, Interrupts - Timer Interrupt and External Interrupt
  + Demonstrate and discuss the design and operation of a complete system.

# Exercise 1 (**2 marks**)

Use CTC mode of Timers to generate the following waveform on the OCnA (with n is 0,1,2 depending on the Timer you use).

Diagram

Description automatically generated

This figure is adapted from the definition of the CTC mode from datasheet (Page 100 in the attached datasheet on Canvas). You are freely to choose the type of Timer, the setting, and Interrupts if any. The generated delay time can be a bit off from the requirement (could be due to rounding the Timer Count from calculation).

# Exercise 2 (**4 marks**)

Write a program to generate the following Timing Diagram with the order:

* Step 1 - A button is pressed (falling edge is generated)
* Step 2 - A GPIO pin number 1 will be raised to 1 right away
* Step 3 - After 1 second, a GPIO pin number 2 will start generating 0.5Hz clock (1 second ON, 1 second OFF)
* Step 4 – At the **rising edge** of the clock om GPIO2 pin 2, a data bit will be written out on the GPIO pin number 3. **The data is read from an array of 16 elements, that is hardcoded in your program.** 2 15 cycles (16 cycles in total)
* Step 6 - GPIO pin 1 will be pull down to zero

Timeline

Description automatically generated with low confidence

In this program, you are required to use Timer to generate the delay time. Other than that, you are freely to choose your other approaches (use Interrupts or not, use External Interrupts or not).

To verify the program, it is recommended that you connect the GPIOs to button and LEDs correspondingly.

# Mini Project - Basic Digital Clock with AVR Microcontroller (**24 marks**)

## Function Description

You will design a basic clock having the following functions:

* The clock can count real time and display the time on 7-segment LEDs in 24-hour format.
* The clock has an alarm that goes off after some pre-set time. This pre-set time can be fixed in the embedded code (for example you can set 3 minutes inside your code).
* The clock can be controlled and configured using external buttons with the following control scheme:

1. Before it runs, we must set the *Initial time* (this is the **Set** **mode**). After reset or power-up, the system will go into this mode by default. To set the time, we need:
   * A button to choose different digits of the 7-segment for minute and hour settings by shifting the selection to either left or right. We call this **Digit** button
   * Another button to increase (or decrease) the value of digits. We call this **Value** button
2. To start count time, we need to press another button to change to **Time mode** (every time you press this button, it will swap between Time mode and Set mode). In this Time mode, the clock simply counts and display time starting from the *Initial time* on 7-segment.
   * Due to the pin limitation, we will use **only 2 of 7-segment LEDs** via GPIOs of the AVR Controller in this project.
   * Seconds will not be shown.
   * At one time, only Hours HH or Minutes MM will be shown on the 2 LEDs.
   * A button will be used to switch the display from HH to MM. If we press it, it will change from HH to MM and vice versa. We call this **Display** button
3. During the Time mode, as time elapses after the pre-set time, an LED/buzzer will blink/sound with the frequency of 10Hz for 5 seconds to indicate this alarm.

## Other essential design requirements

Other requirements of this system:

* The entire system is constructed properly on breadboard.
* Buttons should be debounced properly.
* You can use delay C functions (such as delay\_ms) for other purposes **however you must use Timer to keep the time for the clock.**
* Timer for this clock should be precise – You should measure the time of this clock by testing it versus your laptop’s clock for at least 8 hours. Document the number of seconds difference between the two and report.
* Pay attention to the pin assignment – since we got many peripherals now and your MCU has limited number of pins, you might need to plan your pin allocations wisely and come up with control strategy. *The number of buttons as suggested above is your information, if you do come up with different number of buttons but still deliver the functions of the system, that is still acceptable in this work.*

## Other optional functions

Your groups can consider implementing the following optional features:

* An indicator LED to tell what mode the clock is in – Time or Set mode.
* Some button or mechanism to turn on and turn off Alarm before 5 seconds.
* Blinking effect for LEDs to indicate which one is active. For example, when you set the Minute, that LED can blink so we know we are setting that digit.
* Other useful functions are welcome.

# Demonstration for the mini-Project (**5 marks**)

For the mini project, your team will be asked to join a short demo session where I will ask questions to both members.

# Teamwork

The contribution of each student will be assessed via peer assessment and teacher’s evaluation.

# Submission

Each pair is required to submit the following:

1. You need to submit some pages of supporting document (I will provide template). The purpose of this is to explain your pin assignment and your design choice. Maximum page is 6 pages.
2. Folders to store Arduino projects. These projects should be ready to re-run (*I will verify your work when I mark)*.
3. Videos to show your systems working
   * One video for Exe1
   * One video for Ex2
   * One video for Mini project

Use the same guideline for Lab2’s video for making these videos.

1. Finally, you can upload all of these into a OneDrive folder, name it **EEET2505\_Assignment\_Team\_Number\_XXX**, and then share the access with me and your teammates.

The link to the folder MUST BE PASTED to the Canvas

# Assessment

* Students will be marked based on their work, their report and demonstration with the lecturer.
* During the demonstration, students need to demonstrate their systems and prepared to answer some questions.
* Peer assessments and extra questions will be used to assess contributions of each student.

Marking Rubric is found on Canvas