# Binary Game Using µC/OS-II

**Objective:** The objective of this lab is to become further familiar with µC/OS-II through design and implementation of a binary game. Multiple periodic tasks will be created for core functions and I/O control. Semaphore, event flag, and mailbox (or message queue) should be used for inter-task communication and synchronization.

**Procedure:**
Unlike the previous labs, step-by-step instructions will not be provided for this design lab. Instead, a number of system requirements will be given below. In addition to the contents of lab report and in-lab work, your mark will be predominantly evaluated based on the satisfaction of the system requirements. Read through all the requirements carefully prior to design and implementation.

**Design Constraints:**

**1.** The binary game should meet the following key requirements.
- 1.1. The binary game must be implemented using µC/OS-II.
- 1.2. The binary game must run on a DE2-series board.

**2.** General design constraints of the binary game are listed below
- 2.1. Each round of the game consists of 10 questions.
- 2.2. Each question is to find the equivalent binary value of a random decimal number between 1 and 255.
- 2.3. Each question has 30 seconds to answer.
- 2.4. Each correct answer (i.e. equivalent binary value) wins 10 points.
- 2.5. (Optional) The game should have multiple levels and speeds; i.e. shorter time to answer a question at a higher level.

**3.** Constraints on the implementation of the binary game are listed below.
- 3.1. After power on, the game should be in IDLE state.
- 3.2. While the game is in IDLE state, pressing KEY1 should start a new game.
- 3.3. During the game, pressing KEY1 should toggle between PLAY and PAUSE states.
- 3.4. When the game is in PAUSE state, pressing KEY0 should reset a game and enter the IDLE state.
- 3.5. SW7-0 should be used to set the equivalent binary value.
- 3.6. Pressing KEY2 to enter an answer. Note: Also reset the remaining time to 30 seconds.
- 3.7. Remaining time for each question should be displayed in seconds on the top row on the LCD display.
- 3.8. Current score should also be displayed on the top row of the LCD display.
- 3.9. The randomly generated decimal number should be displayed on the bottom row of the LCD display.
- 3.10. After the end of each game, both the total score and the total elapsed game time should be displayed on the top row of the LCD display.
- 3.11. The total elapsed game time should only count the time that the game is in PLAY state.
- 3.12. The total elapsed game time should be displayed in MM:SS format.
- 3.13. The total elapsed game time should be with the accuracy of no less than 0.25 seconds.

3.14. (Optional) While the game is in IDLE state, the bottom row of the LCD display displays "Press KEY1 to start……".
3.15. (Optional) While the game is in PAUSE state, the bottom row of the LCD display displays "Press KEY2 to resume……".

**4.** For learning purposes, the following constraints on μC/OS-II must also be satisfied.
4.1. At least three periodic tasks should be used for implementing the binary game.
4.2. At least one mailbox should be used for inter-task communication or synchronization.
4.3. At least one semaphore should be used for inter-task communication or synchronization.
4.4. At least one event flag should be used for resource or activity synchronization.

**Deliverables:**

**1.** Lab report: Be complete in your lab report. The lab report should include detailed description of your design and implementation, as well as the test procedures and results that validate your design.
**2.** Design files: Zip the folder which contains your source code and the document. Hand in the zip file to Blackboard before the due date specified.
**3.** Demonstration: **You must also demonstrate your final design to the lab instructor(s)**.