Practical Part: Portable System Code Writing

Additional Hardware Connections:

- Buttons: PB0 to PB2 are connected to controller pins P1.0 to P1.2 respectively.
- Analog Input: An analog input is required to be connected to controller pin P1.3 (A3).
- LCD Display: Connect D4-D7 of the LCD to pins P1.4 to P1.7 respectively for 4-bit operation mode.
- LCD Control Lines: Code for the LCD is available; update it as needed for the control lines on pins P2.5, P2.6, and P2.7.
- Generator Output (Input Capture Mode): Connected to pin P2.4.
- Buzzer (Output Compare PWM Mode): The scope channel CH2 is connected to pin P2.2 and works in parallel.
- Vcc Voltage: The development kit operates between 3.5v-3.65v unlike the 3.3v in the lab system. The Vcc depends on which computer the development kit is connected to. To precisely check the output voltage level of the Vcc, measure the logical '1' voltage.

System Software Architecture

- Simple FSM Required: The system software must be based on a simple Finite State Machine (FSM) architecture as described in the preparation report. Upon pressing one of the three buttons, one of four actions should occur based on an external interrupt request.
- Portability Required: The system code must be divided into layers making it portable across families by merely replacing the BSP layer, specifically for MSP430x2xx and MSP430x4xx.
- Implementation Notes: Most development stages should be carried out on a personal development kit, with final testing and fitting done in the lab.
- Placement of Functions: While LCD driver functions need to be placed in the HAL, the API based on them should be located in the API layer.
- Before Writing the Code: A detailed software architecture diagram showing the states, nodes, and arcs representing transitions from one state to another due to interrupt requests must be drawn and attached to the preparation report.

System State Requirements:

- Idle (state=0): The controller, after a reset or upon completion of all actions, returns to sleep mode (Sleep Mode).
- Upon pressing button PB0 (state=1): Implement a frequency counter (Timer_A1 in MSP430x2xx family) to measure the frequency of an external clock signal fed from the signal generator to the controller's pin P2.4. Display the measured frequency value on the LCD screen according to the details provided, dynamically updating the measurement without refreshing the entire screen.
- Upon pressing button PB1 (state=2): Implement an Up/Down counting mode of a stopwatch on the LCD screen starting from 00:00 up to 01:00 for counting up and down to 00:00. The initial state upon entering this mode is 00:00. The counting precision and the positioning on the LCD screen should be as shown in the following illustration. This mode is defined to end upon reaching a counting value of 00:00 and can be interrupted by other buttons. The counting must be performed using the Timer_A0 module interrupts only within the MSP430x2xx family.
- Upon pressing button PB2 (state=3): Implement a tone generator based on an input signal vin with edge values 0v-3v from the generator producing square wave cycles. fin should be within the range [10Hz – 50Hz] and the output from the controller's pin P2.2, connected to a Buzzer, should emit tones in the range [1kHz – 2.5kHz]