Ask

Objective: Design an all-in-one count-down alarm system that users can program, utilizing the bare metal methodology of embedded operating system interaction.

Inputs: Keypad buttons (numbers 0-9; and letters A,B, and D). Optional: Keypad button C.

Outputs: LEDs and LCD.

Constraints/Relationships:

- Time is entered as m:ss
- Valid times can go up to 9 min and 59 sec
- User can press A to start the timer
- User can press B to stop/turn off
- User can press D to input the time
- User must use a prompt to enter time
- Every time a value is entered, an LED lights up
- The LCD will display Time Remaining: followed by the current time
- When the specified time is reached the LCD will display Times Up and multiple LEDs will turn on
- Must run "forever"
- [OPTIONAL]: Expand the options
    - User can press C to control counting (up or down)
    - The LCD will display Time Passed: followed by the current time when in count up mode
    - When the time has been reached the LCD will display Time Reached, and multiple LEDs will turn on

Research/Imagine

Research Tasks:

- Keypad and LCD configuration
    - How they work (look through
    - Ports used
    - How the ports should connect
    - How to receive data from the keypad
    - How to send data to LCD
- Determine how to utilize interrupts and ISR for keypad presses.
- Determine how to utilize SysTick for the timer feature
- Figure out a method to deal with bounce

Possible Solution:

- Create interrupts for each button press:
    - D: Starts a timer setup function loop which accepts number presses
        - Allows num 0-9 button press interrupts
        - Presses add to the timer (with a limit of 2 digits for both minutes and seconds)
    - A: Ends the timer setup function loop, and starts a countdown loop
    - B: Ends the countdown loop, saves the current timer to a global var, and starts a standby loop which waits for either button “A” or button “D” to be pressed.
- The main while(true) loop handles which mode the timer is in and which button interrupts should be accepted based on the mode.

Plan

Flowchart:

Test Plan:

- Display
    - Verify that the display shows “Time Remaining: 0:00”
- Timer Setup
    - Verify that values can only be inputted into the timer after pressing button “D”.
    - Verify that every button press that inputs a value lights up an LED.
    - Verify that the inputted values are displayed on the LCD.
        - Make sure that the value cannot exceed 9min and 59secs by pressing “1”, “0”, “0”, “0”.
- Timer Start
    - Verify that pressing button “A” starts the timer (numbers on LCD begin decreasing)
    - Verify that for every second that passes, the timer also decreases by 1 second.
        - Make sure that minutes also decreases by 1 each minute
- Timer Pause/Stop
    - Check that pressing button “B” stops the timer.
    - After a pause has been initiated (from pressing “B”), verify that pressing “A” starts the timer again from where it left off when it paused.
    - During the pause state, verify that pressing button “D” resets the timer, and allows for new values to for the timer.
- End of timer

    - Verify that when the timer hits 0 mins and 0 secs, all the LEDs light up and the LCD displays “Times Up”
- Forever Loop
    - Run the timer multiple times by setting the timer to a low value such as 5 seconds, and verify that the timer works each time.