Ask

Objective: Create a timer that counts down from a maximum of 9 minutes and 59 seconds. The time will be displayed on an LCD display. A user will be able to use a keypad to start the timer with **A**, stop the timer with **B,** and enter the time with **D.** When the timer has completed its count down, the LCD will display **Times Up,** and turn on multiple LEDs.

Inputs:

1. User presses A
2. User presses B
3. User presses D
4. User presses numbers.

Outputs:

1. Timer counts down
    a. Multiple LEDs turn on
    b. LCD displays "Times Up"
2. Timer Stops
3. User prompted to enter the time
4. LCD will display the time entered

Required Materials:

1.) LCD
2.) Nucleo
3.) Solderless Breadboard
4.) Jumper Wires
5.) LEDs

Constraints:

1. A starts the timer.
2. B stops/turns off the timer.
3. D will allow user to input the time.
4. User must use a prompt to enter time.
5. A LED lights up every time a value is entered.
6. The LCD will display **Time Remaining:** followed by the current time
7. When timer finishes:
    a. LCD displays **Times Up**
    b. Multiple LEDS will turn on
8. Must run forever.
9. Must have one interrupt and ISR.
10. Bounce must be addressed.
11. Proper commenting must be used
12. Registers must be controlled bitwise, except for the LCD or an interrupt.

Research/Imagine

A solution to this problem requires several hardware components. The first step would be to read user input using the keypad. The keypad is a collection of normally open switches, when there is no compression, it remains in the off state. It will be required then, that an interrupt be detected on a rising edge. There are four wires that supply power to the rows, and 4 wires that connect the columns to the GPIO.

The LCD used in this system is the 1802 Mouser. The API has functions that will be needed for the display to function correctly. Begin() will initialize the LCD, clear() will clear the display for a current cursor row. The print function will print a string onto the LED. The setCursor(col, row) function will put the cursor in the correspond column and row.

The behavior of interrupts must be understood to complete this assignment. An interrupt is a hardware triggered event; in this case a button is pressed on the keypad. When an interrupt occurs through a button press, we must invoke an ISR depending on the state the timer is in, and the button that was pressed. The five conditions to allow interrupts are:

1.) Arm Device.
2.) NVIC enabled.
3.) Global enable.
4.) Priority must be higher than current level, a thread control.
5.) Trigger occurs.

When these steps are done, enable the pins, and associate them with a specific interrupt. Create a detection mechanism for a rising edge and call the appropriate ISR.

Plan

The first task will be to enable the ports for the pins we are going to use. For the SDA and SCL line PB13 and PB12 will be used respectively. For the Keypad, all four rows and columns need to be connected. The rows will be supplied, while the columns will be connected to a GPIO. For this implementation, the columns 1-4 starting from the left will be mapped to PB8-PB11 respectively. There will be four external LEDs connected to the breadboard. When a value is entered, a LED will light up, depending on the column the button is in.

To connect the LCD, ground and VCC must be supplied. SDA and SCL must be connected to their corresponding pins. For this project SDA will be connected to PB13, and SLC will connect to PB12.

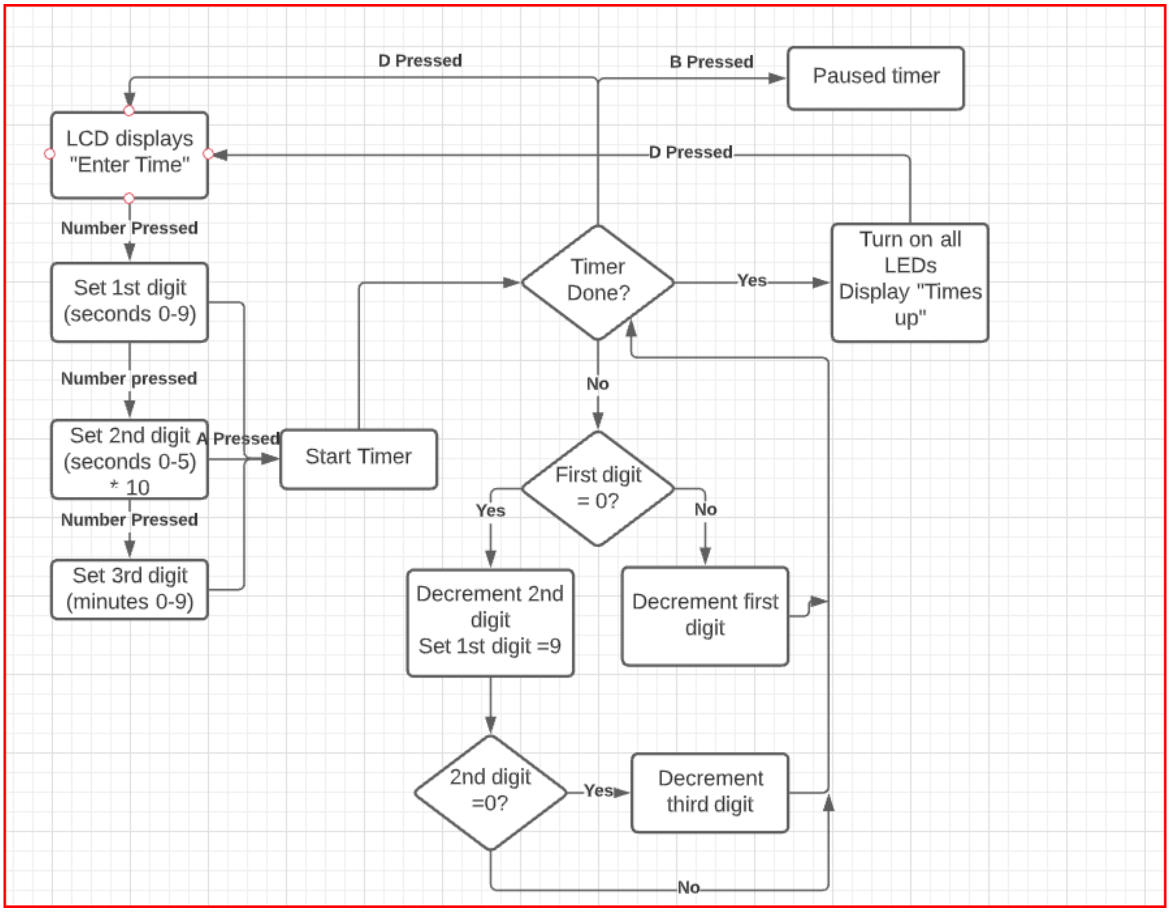
Each button on the keypad will have different behaviors based on the state. The three states of the timer are:

1. Inputting Time
   a. A: Starts timer, if there is no time inputted, does nothing.
   b. Numbers: Displays current digit, and lights up an LED
   c. B, C, D: Does nothing.
2. Timer Started
   a. A & C: Does nothing.
   b. B: Pauses timer

c. D: Change state to “Inputting time”.
d. Numbers: Does nothing.

3. Timer Complete
   a. A: Does nothing.
   b. B: Does nothing.
   c. D: Change state to “Inputting time”

The timer will be initialized with the “Inputting Time” state, where users will only be able to input the time, as well as start the timer. If the counter begins, the user may pause the timer, or renter a time. If the timer is allowed to finish, the LCD will display “Times Up” and turn on all four LEDs. The user may press D in this state to reenter a new time. The flowchart below details at a high level of abstraction the steps required to create the timer.


D Pressed
B Pressed
Paused timer
LCD displays "Enter Time"
D Pressed
Number Pressed
Turn on all LEDs Display "Times up"
Set 1st digit (seconds 0-9)
Timer Done?
Yes
No
Number pressed
Set 2nd digit (seconds 0-5) * 10
A Pressed
Start Timer
First digit = 0?
Yes
No
Number Pressed
Set 3rd digit (minutes 0-9)
Decrement 2nd digit Set 1st digit =9
Decrement first digit
2nd digit =0?
Yes
Decrement third digit
No