

Andrew Schick

CSE 321 – Embedded Systems

Project 2 – Planning

10/15/2021

## Ask

Purpose:

Countdown alarm displayed on an LCD where users can input the time and control the timers' behavior.

Inputs: keypad

- A – start timer
- B – pause timer
- D – input time
- Numbers 0-9 – set digit

Outputs: LEDs, LCD,

- Specific LEDs light up when a value is entered.
- multiple LEDs will light up when the countdown reaches 0.
- LCD will display 'enter prompt 0:00' when D is pressed
- LCD will display 'Time remaining countdown\_number' when A or B have been pressed
- LCD will display 'Times up' when the countdown\_number reaches 0.

Constraints

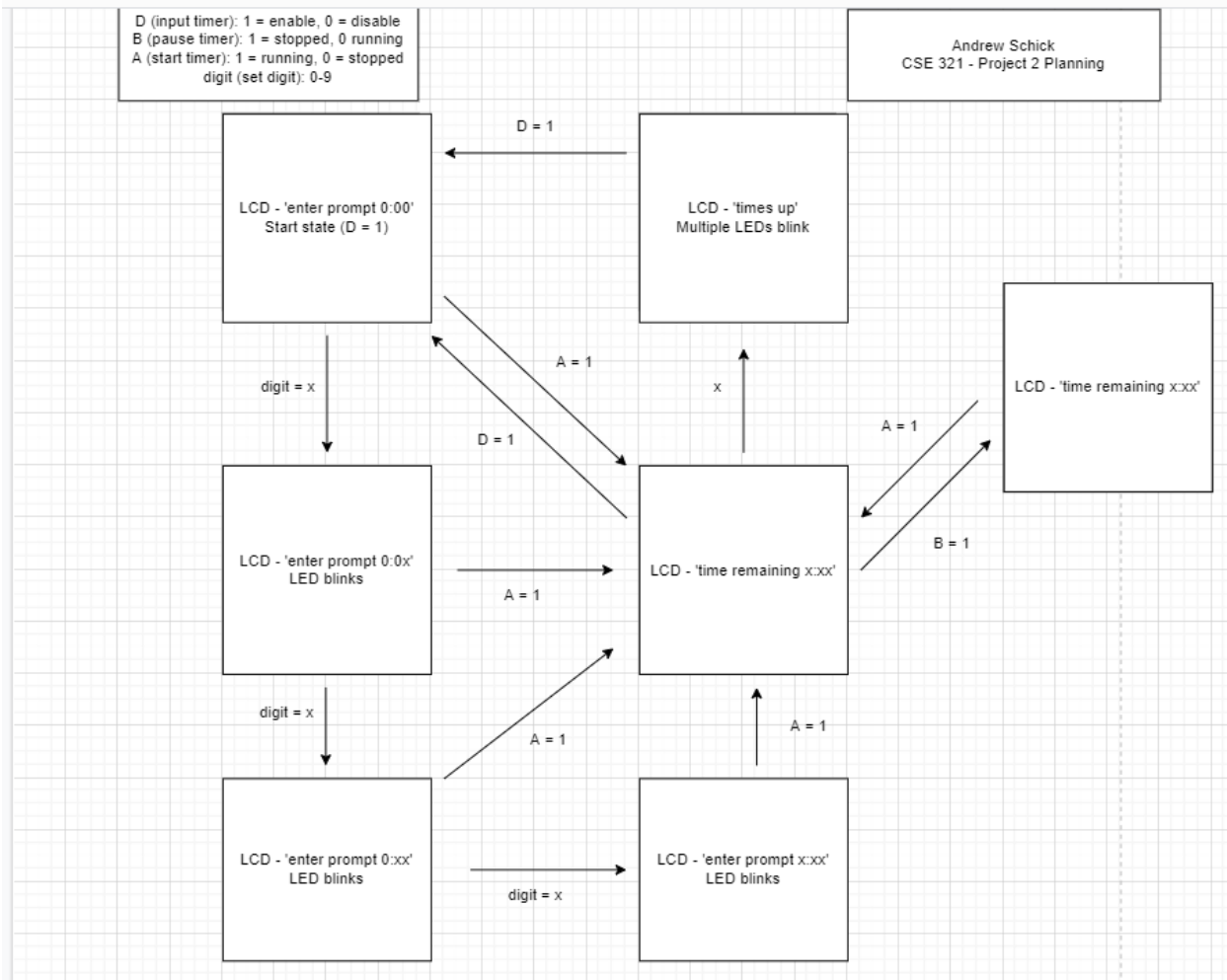
- Time entered can range from [0:00, 9:59] (m:ss)
- Must be able to display numbers and letters on LCD at the same time
- Timer will run forever meaning the system will continuously take in inputs and display on LCD
- Only Mbed OS API allowed
- Must use interrupts and ISRs

Peripherals:

- Keypad
- LCD
- LEDs

## Research

Looking at the 3 control inputs (A, B, D) and the LCD display output states a state diagram would be highly effective in understanding the system better.



After developing the state diagram, it seems there are four 'main' states where interrupts and ISRs can disseminate them. The four main states being:

- Setting the timer (activated by D)
- Timer counting down (activated by A)
- Timer paused (activated by B)
- Times up (activated by LCD screen showing 'time remaining 0:00')

## **Planning**

Being four main stages there will be four interrupts and four ISRs.

- D interrupt, setting\_timer ISR.
- A interrupt, counting\_down ISR
- B interrupt, timer\_paused ISR
- 0 interrupt, timer\_finished ISR

In the state diagram there is a state marked 'Start state', this state will instantly be entered by triggering the D interrupt in main. Once the program has entered setting\_timer it will never come back to the main method. The program will stay within the four ISR methods changing only when an interrupt is called.

Not sure if there needs to be four interrupts and instead just use polling with conditions and have set variables for some of the current interrupts.

Each interrupt will be associated with the activation of a single pin

When the program is in the setting\_timer state the system will constantly be polling for number inputs and A. When there have been 3 numbers entered the system will then only check for the A interrupt. The digits will be entered from least significant to most significant. There will be a global variable used to track the current time and that will be decremented by 1 each second the program is in the counting\_down. The time counter will only decrement when counting\_down, the only other time it can be changed is in setting\_timer. The three standard prompt messages will be stored in global values as strings.