CSE321: Real-Time and Embedded Systems

Project 2

Katherine Stock
University at Buffalo
Fall 2021

Table of Contents

Introduction

The objective of this project is to create a bare-metal timer system that the user can provide input to. The user can swap between two modes, count-down and count-up, before inputting their time and starting the timer via a matrix keypad. When the time is up or has been reached, an LCD and many LEDs will update to alert the user. The following sections will explain the specifications, features, applications, and other details of the timer system.

Specifications

For the timer, specifications of each part are as follows:

- Keypad - Only appropriate buttons may be interacted with at any given time. In the starting state, the user can only cause a response on the LCD by pressing ‘C’, to switch into count-up mode, or by pressing ‘D’ to begin inputting time. Number keys do nothing at this point. Once in the time state, the user can only interact with the number keys -- provided that they are valid, the software will not let the user enter a minute value greater than 9 or a seconds value greater than 59 -- ‘A’ to start the timer, or ‘B’ to reset.
- LCD - The LCD updates are triggered by state changes caused by user input on the keypad. Specifics of these are defined in the code comments and README but there are 8 possible states that all trigger different responses and prompts to be displayed on the LCD. In the timer state, the LCD updates on each second.
- LEDs - One LED for the specified column will flash when a keypress from that column has been read. Once the timer has finished running, 3 different LEDs will light up and remain lit until the user presses ‘B’ and resets the timer.

Features

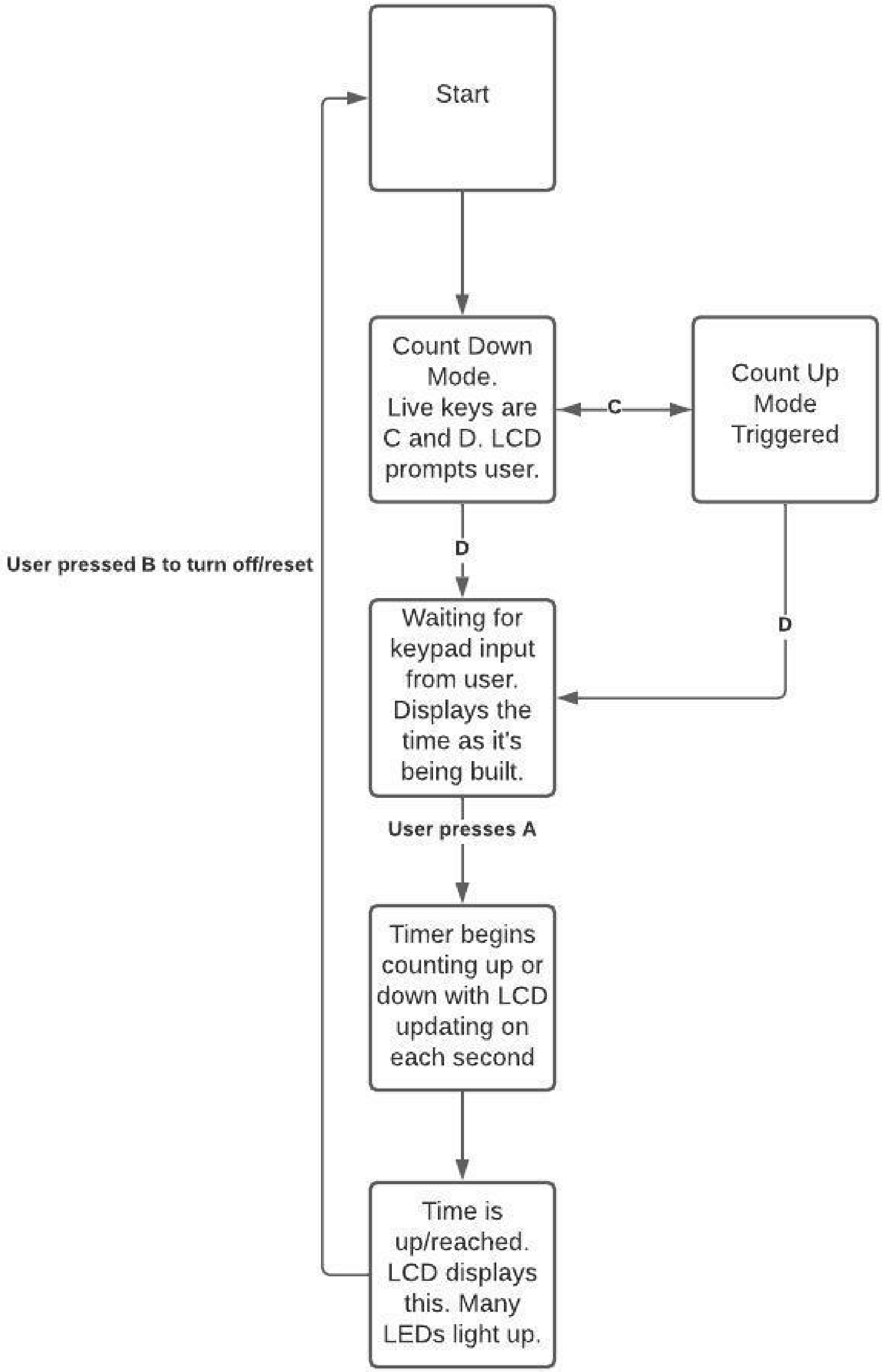
The features of this timer include the following:

- Ability to switch between counting down and counting up
- LEDs that light up on each key press
- An LCD to update the user on the current status of the timer at each second and prompt the user when necessary
- Many LEDs that light up once the timer is done
- A 4x4 matrix keypad for the user to interact with the timer

Applications

Real world applications of the project are in general use as a timer. The timer can only be used for less than 10 minutes so it is ideal for timing something like steeping tea, journaling, doing a short workout, or breaking up a study session into 9 minute 59 second blocks.

Block Diagram


Start
Count Down Mode.
Live keys are C and D. LCD prompts user.
C
Count Up Mode Triggered
D
User pressed B to turn off/reset
Waiting for keypad input from user. Displays the time as it's being built.
D
User presses A
Timer begins counting up or down with LCD updating on each second
Time is up/reached. LCD displays this. Many LEDs light up.

Functionality

Upon start, the timer is in a count-down mode and prompts the user to press D to input time. At this point only can the user press ‘C’ to toggle back and forth between count-down and count-up modes.

After pressing ‘D’, the LCD updates to display the input time as it is being built. The user must enter the time in the form m:ss, only the first 3 valid keypresses are accepted and the user must press ‘B’ to turn off/reset the timer if they make a mistake. Users are not able to input a two-digit minute section or a seconds value greater than 59, this makes our maximum possible time 9:59.

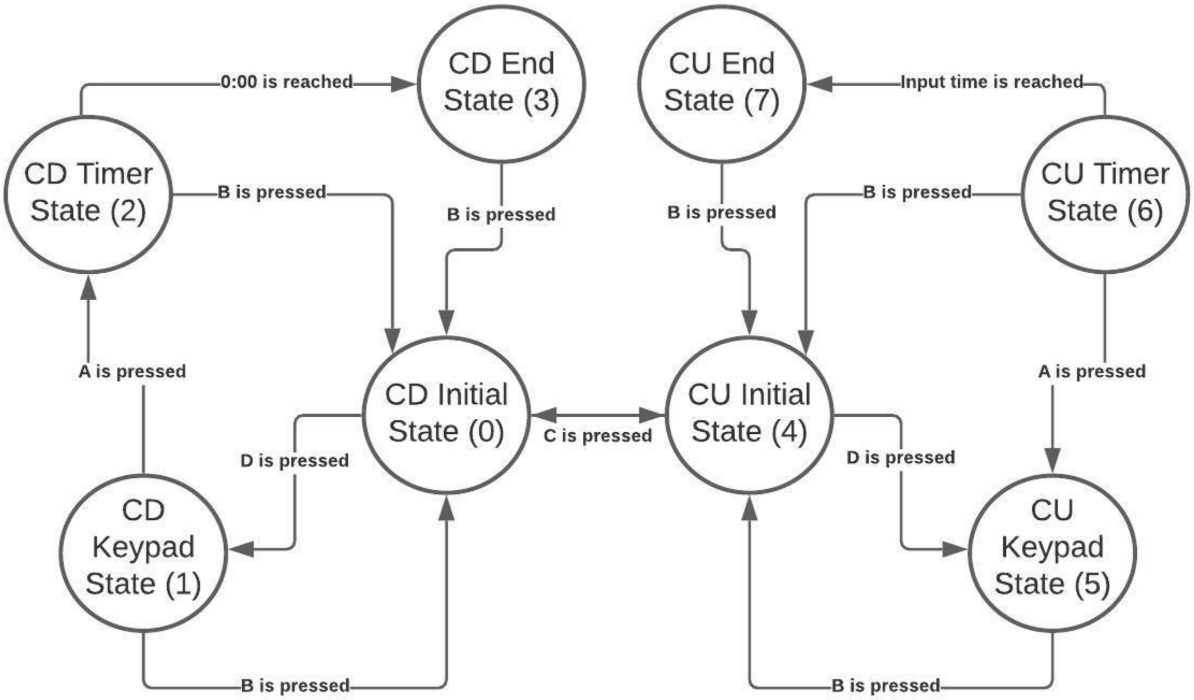
After the time is input, the user can press ‘A’ to start the timer. The LCD will display the time remaining or time passed and update every second. Once time is up or the goal time is reached, 3 LEDs will light up and display that the time is reached/up. It will stay in this state until the timer is reset by the user.

At any point, the user can press ‘B’ to reset the timer and they can then repeat the above process to use the timer again and again. An LED will also light up on each keypress to confirm that the key was read.

Bonus Functionality

Additional functionality includes using the ‘C’ button on the 4x4 keypad to control counting up or down. The default state of the timer is counting down, but this can be changed before ‘D’ is pressed to input the time. Because the base functionality already followed a state diagram, additional states and conditions at each step were the only necessary additions. The model could easily be extended to include an even wider range of user options.

Diagram (FSM State Diagram)

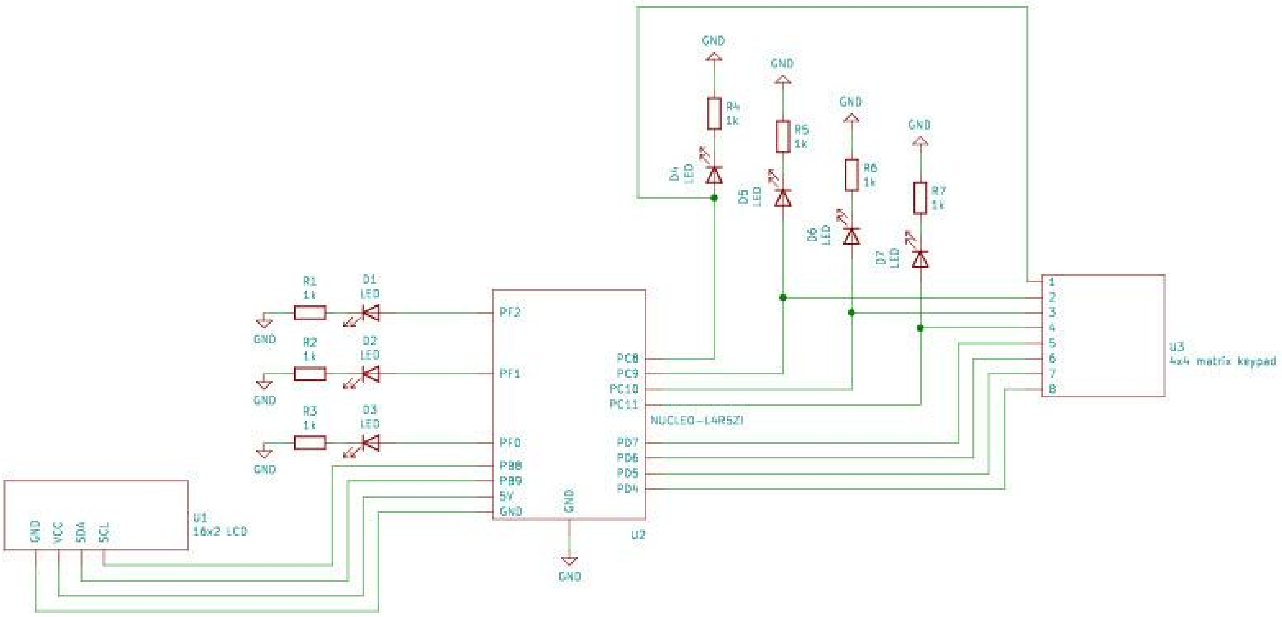


Bill of Materials

	Item Name	Quantity	Price
1	Nucleo-L4R5ZI	1	$28.14
2	Breadboard	1	$15.40
3	USB a to Micro USB B cable	1	$2.68
4	16x2 1802 model LCD (comes with male/male jumper adaptor)	1	$4.88
5	Male/Male Jumpers	16	$3.90/140 pack
6	LEDs	7	$3.95/25 pack
7	Resistors (1kOhm)	7	$0.95/20 pack
8	4x4 Matrix Array 16 key membrane switch keypad	1	$6.99/4 pack
	Total		**$66.89**

Technology requirements only include access to a computer with internet and MBED Studio to program the Nucleo.

Schematic


GND
R4
1k
D4
LED
GND
R5
1k
D5
LED
GND
R6
1k
D6
LED
GND
R7
1k
D7
LED
R1
1k
D1
LED
GND
R2
1k
D2
LED
GND
R3
1k
D3
LED
GND
PF2
PF1
PF0
PB8
PB9
5V
GND
GND
PC8
PC9
PC10
PC11
NUCLEO-L4R5ZI
PD7
PD6
PD5
PD4
U2
GND
1
2
3
4
5
6
7
8
U3
4x4 matrix keypad
GND
VCC
SDA
SCL
U1
16x2 LCD

Test Plan

Basic testing was done as each peripheral was added to the project and more in-depth testing was done once implementation was complete.

To test the keypad polling and interrupts, a character value was set to have the value expected in each column and row. The character was printed for visual debugging of the software and hardware. LEDs for

To test the LCD hardware, declaration, and its associated functions; basic prints, clears, and cursor sets were run.

To test the ticker and other software independently of the keypad, the variables associated with the user-built time and state were assigned set values and run to simulate the timer running after the user has input a time and pressed ‘A’. The ticker was tested side-by-side with the LCD to make sure the function attached to the ticker was appropriately changing states which would update the LCD output.

Once all aspects were tested separately, they were put together and tested for basic cases and edge cases.

Results

The project implementation was successful. The countdown/count-up timer meets the parameters and constraints specified in the handout as well as meeting the specifications decided for my implementation.

The test plan was also successful. Testing in pieces and then all together allowed for the isolation of variables and made debugging software vs hardware much easier. Keypad testing caught hardware bugs with jumpers going to the wrong pin and subsequently being read as the wrong value. Edge case tests also caught previously undefined behavior when specific buttons were pressed when in a state where they should not have caused a response.

Recommendations for Improvement

To improve upon the current model and code, the code can be cleaned up for clarity and more states can be added for a better user experience. Switch cases may also help to clean up the transition and reading of states which are currently wrapped in clunky if statements.

Finding a way for the user to change their input value without needing to restart the entire timer would also be an improvement on the current implementation, especially if the timer is being used regularly.

A final recommendation is extending functionality to include sound so the timer can alert even if the user is not watching the LCD and LEDs.