

CSE 321

Project 2

**Count-Down
Alarm System**

Jesse Both

Fall 2021

November 4, 2021

Contents

Introduction	3
Specifications	3
Featrues	3
Applications	4
Block Diagram	4
Functionality	5
Bonus Functionality	5
Diagram	6
BOM	6
Schematic	7
Test Plan	8
Results	8
Recommendations	8

Introduction

The purpose of this project was to implement a count-down alarm system that utilizes bare metal methodology. The system allows input via a 4x4 keypad and displays pertinent information to the user. When the timer has run out of time, a series of alarm indicator LEDs light up.

Specifications

- Time is entered in the format 'm:ss'.
- 'A' starts the timer.
- 'B' stops and turns the timer off.
- 'C' changes the count direction.
- 'D' is pressed to initiate entering a time.
- Bounce needs to be handled.
- An LED lights up after every keypress.
- The system runs forever.

Features

The system features a 4x4 keypad where a series of interrupts are utilized to determine the key that was pressed. The keypad allows the user to manipulate the timer. The user is able to input a time, start and stop the timer.

An important feature that needed to be handled was to get rid of or reduce bounce. In this system this was handled by adding a quarter second delay between each keypress. In the main loop there is a thread delay of 10ms. This quarter second key press delay was handled by having a keypress flag and a time decrementer. When the keypress flag was set, the time decrementer would be a value between 1 and 25. This value would decrement until it is equal to 0 and the keypress flag is set to 0 to indicate that a new key can be pressed. This solution is efficient because it does not lock the thread for the entire 250 ms.

A 16x2 LCD is used to provide the user with a visual interface of the system. The LCD can display prompts for the user aswell as display the current time.

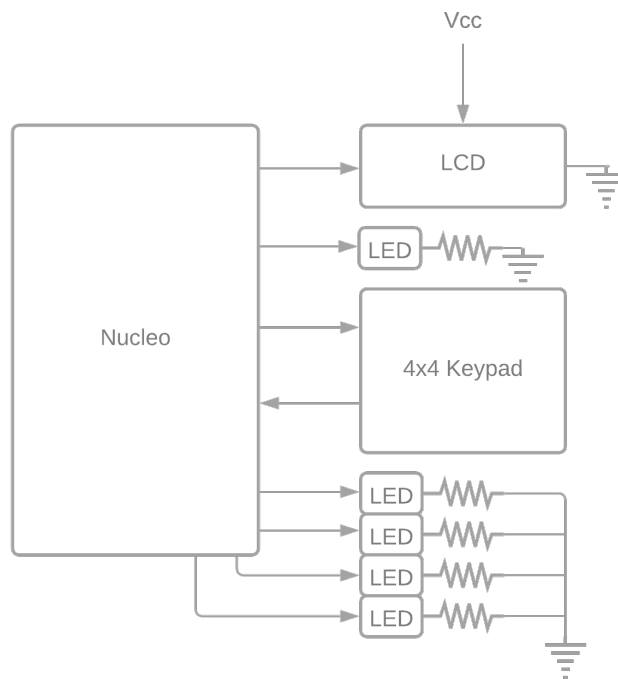
A blue LED is used to provide the user with a visual that their keypress has been noticed by the system.

A series of red and yellow LEDs are used to indicate when time is up. These LEDs are programmed to light up sequentially for an appealing visual indication.

Applications

- Egg Timer
- Procrastination Timer
 - Reduces the amount of time the user can procrastinate.

Block Diagram



Functionality

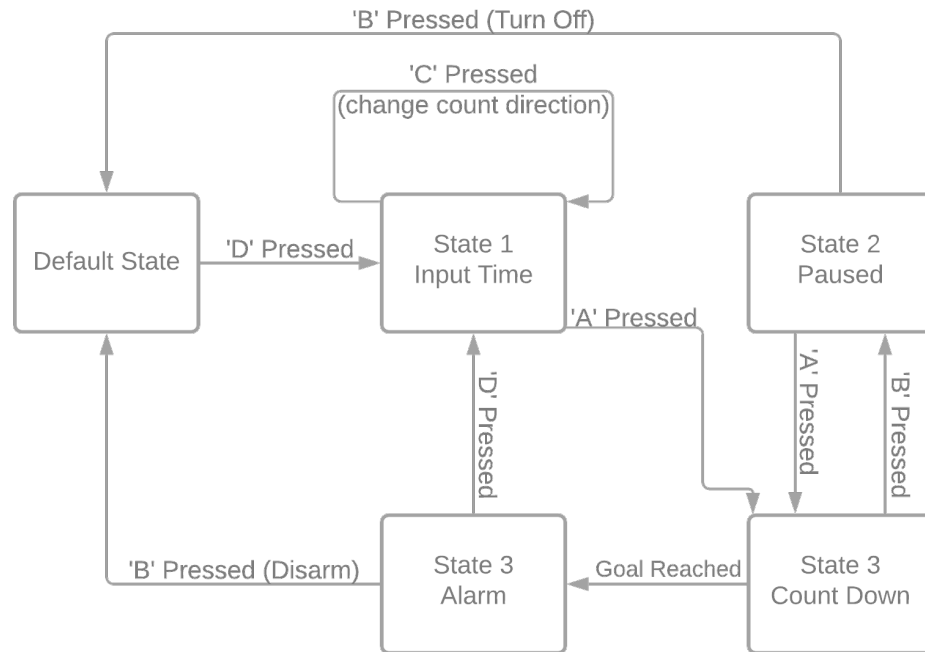
- Press 'D' to input a time.
- Input a time in the format m:ss.
- Press 'A' to start the timer with the input time.
- Press 'B' to pause the timer.
- Press 'B' to reset or 'A' to start after pause.
- When the goal is reached, an LED alarm will blink.
- Everytime a key is pressed an indicator LED illuminates.

Bonus Functionality

The Bonus functionality includes an additional button input 'C'. When C is pressed the counter direction changes and an indicator is displayed on the LCD to show whether the alarm will be incrementing or decrementing. 'C' can be pressed when in the 'enter time' mode. 'C' can be pressed as many times as the user want in the enter time mode. After each press, the timer direction is reversed. The new functionality, including 'C' is as follows:

- Press 'D' to input a time.
- Input a time in the format m:ss. Press 'C' to change count direction.
- Press 'A' to start the timer with the input time.
- Press 'B' to pause the timer.
- Press 'B' to reset or 'A' to start after pause.
- When the goal is reached, an LED alarm will blink.
- Everytime a key is pressed an indicator LED illuminates.

Diagram



BOM

- 4x4 keypad
- 16x2 LCD (1802)
- 5 LEDs
 - x1 Blue
 - x2 Yellow
 - x2 Red
- Resistors
 - x4 1k Ω (for keypad pulldown)
 - x5 220 Ω



Test Plan

The test plan for this system mostly involved trial and error. The first step to verify the system was to make sure the count down timer functionality was working. This was done by hard coding a specified time and using the ticker isr to decrement the timer. In this process, the functionality of the timer string to minutes and second integers was verified as well as converting the integer minutes and seconds back into a string. This was necessary because after the user inputs a string in the format m:ss, this needs to be decoded into 2 integers. While the count down process is happening these integers are decremented and converted to a string that is output to the LCD.

The next step was to verify the keypad. This was verified by using print statements after each key was pressed. If the corresponding key was displayed the system is working. One point of concern in this implementation is if the user were to manage to press a key between the time that the new row value is set and before the gpio pin is activated, the wrong key would be displayed. The chance for this is almost 0, but it is an edge case that should be mentioned.

The last step for testing was to make sure the LED functionality was correct. Based on the specifications of the system, an indicator LED should illuminate after each key press. This can be tested by pressing a key and verifying that it does indeed light up. The other LED functionality includes an LED alarm indicator. These LEDs should emit light when time is up or time has been reached. These LEDs can be verified by setting a time with the keypad and verifying that when the time ends that the LEDs light up.

Results

Based on the testing that took place, the system appears to work as it was designed. The LCD displays prompts and outputs the timer in the correct format as expected. The keypad was properly debounced and allows the user to input characters depending on the systems state. The LED alarm acts turns on and off as it should. The keypress indicator LED turns on briefly when a key is pressed.

Recommendations

It is recommended to expand the timer capability. In its current state, the timer can only allow for input that is in the format m:ss. In order to make the

timer more capable, it should be able to handle atleast hh:mm:ss. Inorder to implement this new version of the system, some adjustments would have to be made in the timer code file. The function that need adjustments are `int_timer()`, `set_press_timer()` and `output_press_timer()`.