Zachary Wolf

CSE321

Project 2

Fall semester, 2021

Purpose: To create an implementation of a timer in an embedded system that can be set and display its time.

Introduction

The following document provides details for the specifics of a timer project that was developed for use on the L4R5ZI board. Why, how, and it's components are all detailed in the following sections.

Specifications

This implementation is designed for usage with any PC capable of running MBED studio, and that has the IO required to connect to a micro USB B port.

Pins PB3 PB4 and PB5 triggers when powered.

All resistors used are 1kOhm

LEDs should not exceed 3 volts at any time. If an LED has blown, it it likely the result of a short circuit, or improper wiring that circumvents the resistors.

Input power should function for both 3v3 or 5v for the LEDs, and 5v for the LCD.

Features

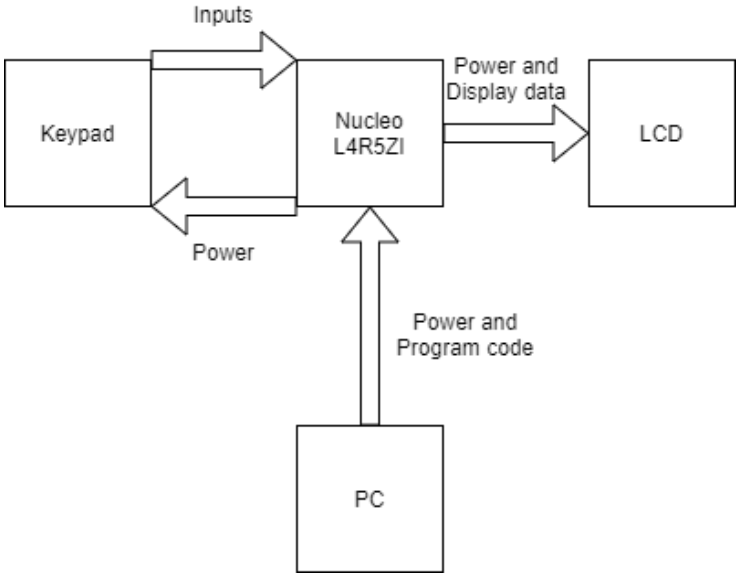
- An external LCD display which will display the time remaining or being set, as well as a short sentence explaining what the time on the screen signifies.
- Keypad for starting, stopping, setting and resetting the timer
- 3 LEDs to denote when A, B and D have been pressed on the keypad

Applications

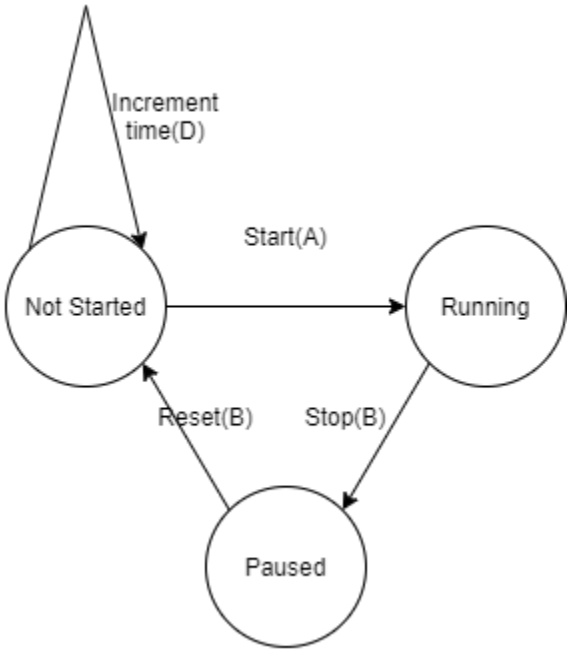
The device that was designed is a timer that can be set for times up to just below 10 minutes and therefore is useful in any scenario that calls for a timer.

It could be used to keep track of the intended cooking time of a meal, taking a timed break, or

Block diagram



Functionality diagram - FSM diagram



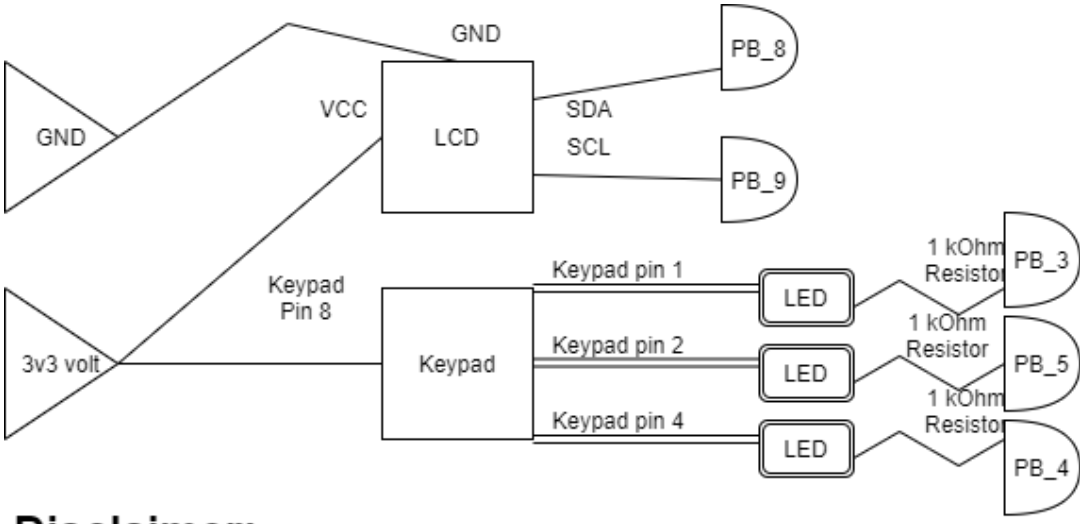
Note:

This implementation does not have a separate time setting state. D was supposed to increment the start time by a given amount.

Bill of Materials

- 3 LEDs
- Various assorted jumpers/wires
- NuceloL4R5ZI
- 3 1kOhm resistors
- C++
- MBED studio (macOS version)
- breadboard
- 4 x 4 Matrix Array 16 key membrane switch keypad
- USB a to Micro USB B cable
- USB A to USB C adaptor (only required if laptop lacks USB A ports and has USB C)
- Grove-16x2 LCD - model JHD1802M1

Schematic



Disclaimer:

This keypad configuration only works to be able to output A, B and D. It does not function for the rest of the buttons, but the design, had it worked as intended would have been able to function with just those 3 buttons.

Test plan

The plan for building and testing the implementation composed of multiple stages

Stage 1: Create a skeleton for the code
- Make dedicated functions for specific inputs

Stage 2: Figure out how to connect keypad
- Use print statement to test that keypad inputs are acting as intended

Stage 3: Implement interrupts for keypad input

Stage 4: Compensate for bounce
- test for bounce using print statement from keypad, check said outputs against expected button presses
- Wait commands should be able to negate bounce

Stage 5: Research and implement the timer
- Timer should be integrated with the existing dedicated input functions
- Should now be able to test the timer using the keypad inputs.

Stage 6: Figure out printing to LCD
- LCD text should change between running and input
- LCD text should change when timer is done
- Displayed time needs to be constantly updated

Results

Unfortunately the design was unable to be implemented correctly in the time provided. A misunderstanding early on related to how the keypad should be implemented, and various bugs from MBED in regards to various files not being found despite being where they should be, made it so code could not be tested for a substantial amount of time. This early on misunderstanding and an inability to check the code that was being written lead to the Keypad not being able to be implemented correctly. Since later steps required the keypad to test and program around, this caused the rest of the steps to be unable to be completed.

Recommendations for improvement:

More elaboration in the instructions would be beneficial. Elaboration on various aspects of the instructions for this report and the readme such as a sentence or two to briefly explain each section would be very useful. The rubrik can provide some insight in this regard, but often the title is the only explanation which results in a lot of guesswork.

In the LCD instructions, mentioning that the LCD functions should be appended to the end of the LCD object name would be useful.

Finally a clearer division between what's needed for stage 2 and stage 3 seems necessary. In its current format, having the label "part 2" right after "stage 2" can cause confusion as to whether it's "stage 2" or "part 2" which needs to be submitted by the midpoint. Also in regards to stage 2, having "project 2 readme" be in one grading rubric, then having an entire separate rubrik for the same readme makes it seem like they're two separate things. This can lead to confusion as whether or not a basic description readme, or a more in depth read me that corresponds to the rubrik is needed.