CSE 321 Project 2 Report:

Alarm/Timer system using the Nucleo embedded platform

Prepared By: Chaktim Wong, Senior
Department of Computer Science and Engineering
School of Engineering and Applied Sciences
University at Buffalo

Introduction

This project involves implementing an all-in-one count-down alarm system that users can program. The project is implemented using the Nucleo embedded platform. The focus of this project is on utilizing the bare metal methodology of embedded operating system interaction.

Users will be able to interact with a 4 by 4 keypad with 16 buttons in total. There is an LCD which displays the timer value and any prompts included with each function. The included user callable functions are start timer, pause timer, and create timer through buttons 'A', 'B', and 'D', respectively. Number buttons 0-9 are used to set the timer up to 9 mins and 59 seconds after calling the create timer function. Once the timer is started, the user inputted timer value when decrease by 1 every second until the value hits 0 at which point the display shows "Time's Up" and all the LEDs light up. The timer will run forever (will be able to repeatedly create timers) until unplugged or stopped by the user.

Specifications

- Time is entered as m:ss
- Valid times can go up to 9 min and 59 sec
- User can press A to start the timer
- User can press B to stop/turn off
- User can press D to input the time
- Every time a value is entered, an LED lights up
- The LCD displays "Time Remaining: " followed by the current time
- When the specified time is reached the LCD will display Times Up and multiple LEDs will turn on
- Runs "forever"

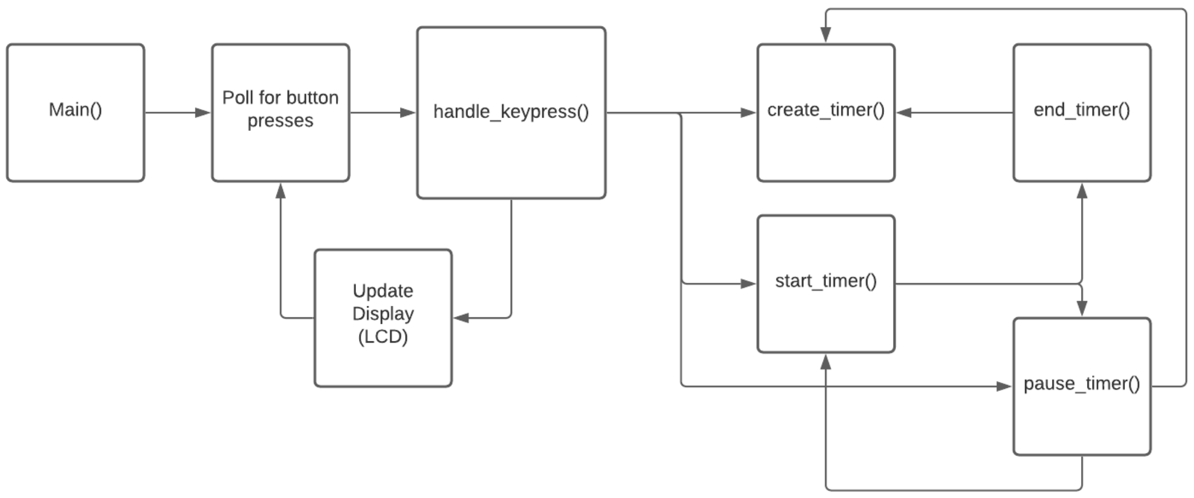
Features

- Timer
- Can handle up to 9 minutes and 59 seconds
- User programmable
- Alarm system
- LED light feedback for each button press
- All-in-one system
- Uses the Nucleo Embedded Platform
- Relatively Cheap
- Safe
- Quiet
- Hours of fun

Applications

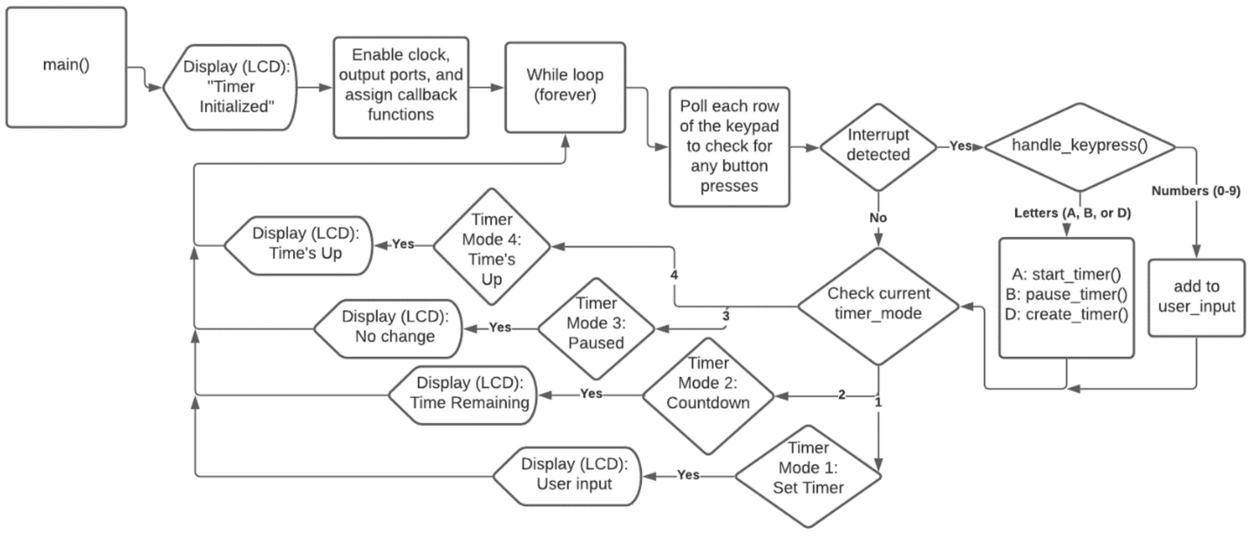
- Timer (for cooking, exercise, breaks, and more)
- Toy
- Practice tool for Nucleo programming
- Experiment/learning platform for learning how circuits work

Block Diagram



Functionality Diagram

- Flow Chart

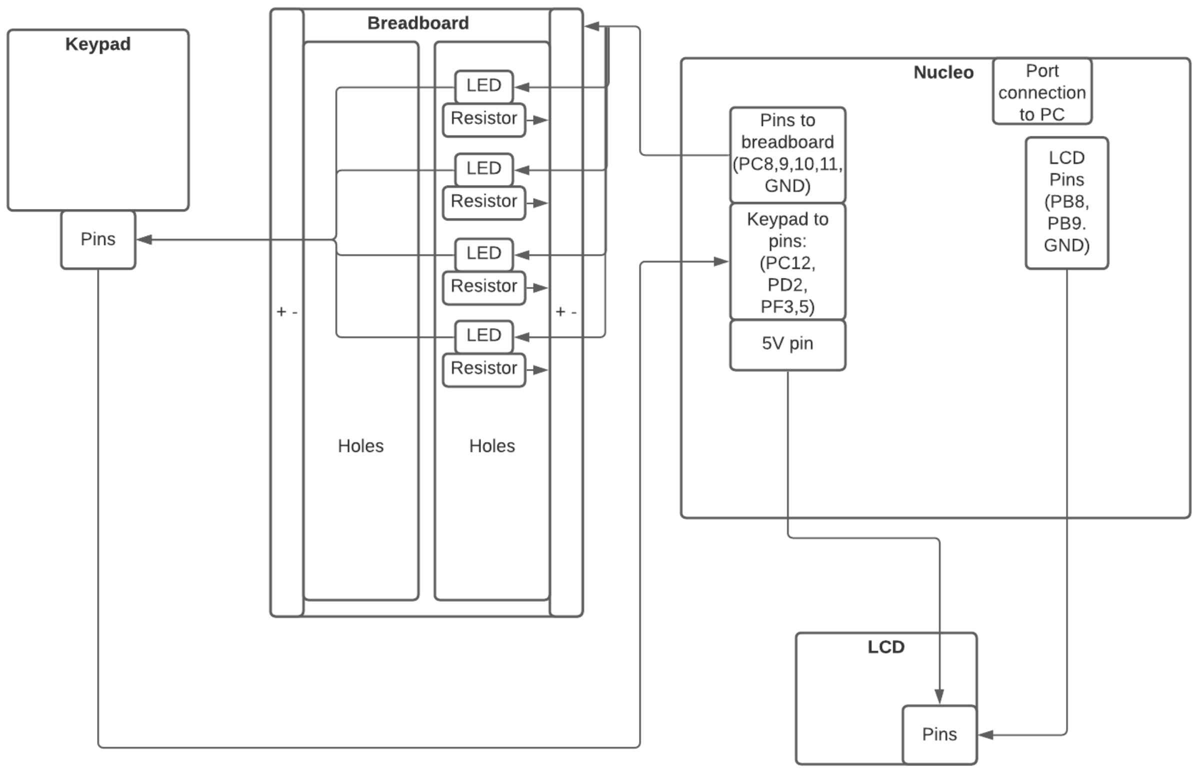


BOM

Materials needed for this project include the following:

- Nucleo-L4R5ZIKeypad
- Solderless Breadboard
- Jumper Wires (at least 4 Standard male/male wires)
- LEDs
- Good mental health
- Adequate sleep
- Electricity
- Resistors (1kOhm)
- A computer to program the Nucleo
- USB a to Micro USB B cable
- 16x2 LCDs with I2C for Arduinos and Raspberry PI, 1602 or 1802 model with the I2C element soldered on
- 4 x 4 Matrix Array 16 key membrane switch keypad

Schematic



Test Plan

- Check Display
  - Verify that the display shows “Timer initialized” on startup.
- Timer Setup
  - Verify that LCD displays “Set Timer: m:ss” in timer setup mode.
  - Verify that values can only be inputted into the timer after pressing button “D”.
  - Verify that every button press that inputs a value lights up an LED.
  - Verify that the inputted values are displayed on the LCD.
    - Make sure that the value cannot exceed 9 min and 59 secs by pressing “1”, “0”, “0”, “0”.
  - Verify that pressing D again resets the values displayed on LCD.
- Timer Start
  - Verify that pressing button “A” starts the timer (numbers on LCD begin decreasing)
  - Verify that for every second that passes, the timer also decreases by 1 second.
    - Make sure that minutes also decreases by 1 after seconds hit 0.
- Timer Pause/Stop
  - Check that pressing button “B” stops the timer.
  - After a pause has been initiated (from pressing “B”), verify that pressing “A” starts the timer again from where it left off when it paused.
  - During the pause state, verify that pressing button “D” resets the timer, and allows for new values to for the timer.
- End of timer
  - Verify that when the timer hits 0 mins and 0 secs, all the LEDs are lit and the LCD displays “Times Up”.
  - Verify that timer can be set to a new value from this state.
- Forever Loop
  - Run the timer multiple times by setting the timer to a low value such as 5 seconds, and verify that the timer works each time.

Results

The project was completed after many hours of work. It was a bittersweet experience overall. There were many stressful moments, but also a few dopamine filled moments when a bug has been removed.

Some roadblocks that appeared included setting up the keypad and getting rid of the bounce that occurs from each keypress. Unfortunately, bounce has not been completely resolved (it may not even be a bounce issue); currently, there is a chance that a keypress may be accepted twice even with interrupt delay and polling delay.

The rest of the project was completed relatively quickly, including the LCD configuration.

Recommendations for Improvement

A major point of potential improvement is dealing with the occasional double input from one keypress. This issue may or may not be related to bounce. Currently, there are no leads to why it occurs.

Another point where improvement can be made is adding additional functionality, starting with the keypad button “C”. Currently, no functionality exists for this button.

A small improvement can be made in wire length and wire placement. The LCD and keypad have a very limited range of movement and cannot be moved very from the breadboard and Nucleo. There are limitations to this, but a better setup may improve user experience.

The LEDS can be configured to a pattern when the timer ends to improve the visibility in the alarm feature of the timer.

Finally, the materials themselves can also be improved. A better keypad may stop the double input, but this is untested. A more integrated or modular setup may be possible with different materials. The LCD can be changed to display more data or to give the setup a better overall look. The durability of the setup may be improved with better wires, resistors, and LEDs.

Although the core features of this project has been completed, there are still many points of possible improvements. However, for this class, most of them are not that important, except for the double input issue that has yet to be resolved.