

Course: EEB2083/EDB2063: Microprocessor & Computer Architecture - May 2021

Project:  Kitchen Timer

Due Date:

Submitted to Lecturer:

Submitted By:

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **ID** | **Course** | **YEAR** |
| Tan Yu Ting | 19000505 | Computer Engineering | 2nd year 1st |
| Rayan Roy a/l Edward | 19000652 | Computer  Engineering | 2nd year 1st |

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| 1.0 | Introduction……………………………………………………. | 2 |
|  |  |  |
| 2.0 | Design Process…………………………….....………………….. | 3 |
|  | 2.1 Pseudo-Code……………………………………….………… | 3-7 |
|  | 2.2 Schematic Diagram of the Component……………….…………. | 8-10 |
|  | 2.3 Setup Component Configuration (Part by Part)……………….. | 10-13 |
|  | 2.4 Used of the Code…………………………………………………. | 13-18 |
|  | 2.5 Implementing Code in the Hardware……………………………… | 19-20 |
|  |  |  |
| 3.0 | Troubleshooting……………………………………………………... | 21 |
|  | 3.1 Having problem in the display LCD (00:00)……………………... | 21 |
|  | 3.2 Push button sensitivity ………..………………………………… | 21-22 |
|  | 3.3 Countdown problem because didn’t used return in the coding…... | 22-23 |
|  | 3.4  Countdown will be interrupted by the InterruptIn ………………. | 23 |
|  | 3.5  Having problem to reset the time………………………………. | 23 |
|  | 3.6:   Change the resistor for LED (adjust the brightness)…………… | 23-24 |
|  |  |  |
| 4.0 | Schematic Diagram…………………………………………………….. | 25 |
|  |  |  |
| 5.0 | Coding …………………………………………………………………. | 26-34 |
|  |  |  |
| 6.0 | Video Project…………………………………………………………. | 35 |

1.0 INTRODUCTION

This is project which kitchen timer. So basically this project is very important because this project is one of a real life application where is been used in a daily life. For example, where this kitchen timer is been used where in kitchen where it made to measure unit of time for cooking or for preparation of foods. So basically it helps to make sure that the food not to overcooked and able to serve to people. Thus, this report we will been explaining on how to design the hardware and coding for kitchen timer. Furthermore, we will show on how to debug or troubleshoot the problem that we have face during the project .

In general, this project about kitchen timer. Where the user set a timer and time will countdown. So, requires us to used 3 pushbutton , 1 LCD board and buzzer is optional for this project. In this project required us to used 3 pushbutton. So, these three pushbuttons have their own function where 2 pushbuttons will be used for setting the time for minutes and seconds. For example, where user wants to set the timer for 2:15. Thus it user need to press the pushbutton minutes for 2 times and pushbutton second for 15 times. While the third pushbutton for set/start. So, this pushbutton basically will help to set and start the timer. Next is the LCD board where the function is to display the countdown for the timer. Furthermore, in this project we apply buzzer in it where it will emit beep sound to indicate the timer have finished. Next, we used project requires that if the minutes or second is pressed until 59 after it will go back to 00. Thus, this is the requirement for this project.

We have added some additional features for the kitchen timer. For example, there will be a buzzer to buzz at the last 5 seconds of the count down and when the countdown is end. Besides that, we also added an extra function for our kitchen timer which will allow the timer to terminate and reset during the countdown process. This function will be useful if we set the wrong time for the timer so that we don’t need to wait for the timer to done countdown if we want to set a new time. Lastly, we have also added 3 LEDs that representing different states of the kitchen timer. The LEDs will be helped to check whether the timer and the coding work correctly at the same time will also improve the physical appearance of the kitchen timer.

2.0 DESIGN PROCESS

2.1 PSEUDO-CODE

1.0 Start.

2.0 Include the libraries that will be used in the coding

2.1 Include the “mbed.h” library for electrical components.

2.2 Include the “TextLCD.h” for the LCD output.

3.0 Declare the inputs for the coding.

3.1 Declare 3 push buttons as interrupt inputs using InterruptIn interface with specific pin connected.

4.0 Declare the outputs for the coding.

4.1 Declare a buzzer as digital outputs using DigitalOut interface with specific pin connected.

4.2 Declare 3 LEDs as digital outputs using DigitalOut interface with specific pin connected.

5.0 Create a LCD interface using TextLCD and specified pin connected to it.

6.0 Create an in-built timer and name it as variable debounce using Timer Interface.

7.0 Declare all the variables used in the code.

7.1 Declare integer variable named counter and initialized to 0.

7.2 Declare integer variable named seconds and initialized to 0.

7.3 Declare integer variable named minutes and initialized to 0.

8.0 Create a function that used to update the LCD screen display during countdown.

8.1 Locate the time display at specific position.

8.2 Display the time which is minutes and seconds.

9.0 Create a function that used to do the count down for the timer.

9.1 If variable seconds and variable minutes equals to zero

9.1.1 Return void to quit the function.

9.2 If variable seconds not equal to zero.

9.2.1 The variable seconds has decrement of 1.

9.2.2 Return void to quit the function.

9.3 If variable minutes not equal to zero.

9.3.1 The variable minutes has decrement of 1.

9.3.2 The variable seconds is assigned with value 59.

9.3.2 Return void to quit the function.

10.0 Create a function that used to detect the push button of Seconds.

10.1 If the time passed of the in-built timer in nucleo board (variable debounce) is more than 200 ms.

10.1.1 The variable seconds has increment of 1.

10.1.2 If the variable seconds is more than 59

10.1.2.1 Variable seconds is assigned with value 0.

10.2.3 Reset the timer named debounce.

11.0 Create a function that used to detect the push button of Minutes.

11.1 If the time passed of the in-built timer in nucleo board (variable debounce) is more than 200 ms.

11.1.1 The variable minutes has increment of 1.

11.1.2 If the variable minutes is more than 59.

11.1.2.1 Variable minutes is assigned with value 0.

11.2.3 Reset the timer named debounce.

12.0 Create a function that used to detect the push button of Set/Start.

12.1 If the time passed of the in-built timer in nucleo board (variable debounce) is more than 200 ms.

12.1.1 If the variable counter has increment of 1.

12.2.2 Reset the timer named debounce.

13.0 Create a function that used for the buzzer in last 5 seconds of countdown with parameter of float variable t.

13.1 Turn ON the buzzer by assigning logic value 1.

13.2 Wait for t seconds where t is the time delay.

13.3 Turn OFF the buzzer by assigning logic value 0.

14.0 Create a function that used for the buzzer when timer finish countdown with parameter integer variable i and float variable t.

14.1 Create a while loop with condition decrement of i by 1 (repeat i times).

14.1.1 Clear the LCD screen display.

14.1.2 Display a message “TIME'S UP !!” on LCD.

14.1.3 Turn ON the buzzer by assigning logic value 1.

14.1.4 Wait for t seconds.

14.1.5 Turn OFF the buzzer by assigning logic value 0.

14.1.6 Wait for t seconds.

15.0 Design the main function that will be executed in the program.

15.1 Initialize all three LEDS (red, blue and green LED) to be OFF by assigning logic 0.

15.2 Create a endless while loop

15.2.1 Clear the LCD screen display.

15.2.2 Display a message “PRESS TO SET” on LCD.

15.2.3 Create a while loop with condition variable counter equals to 0.

15.2.3.1 Turn ON red LED.

15.2.3.2 Assign variable seconds with value 0.

15.2.3.3 Assign variable minutes with value 0.

15.2.3.4 Start the timer named debounce.

15.2.3.5 Call the function that detect push button for Set/Start.

15.2.4 Turn OFF the red LED.

15.2.5 Clear the LCD screen display.

15.2.6 Display a message “SET YOUR TIMER” on LCD.

15.2.7 Call the function to update the LCD screen display.

15.2.8 Create a while loop with condition variable counter equals to 1.

15.2.8.1 Turn ON blue LED.

15.2.8.2 Enable the interrupt input of push button for Second.

15.2.8.3 Enable the interrupt input of push button for Minute.

15.2.8.4 Start the timer named debounce.

15.2.8.5 Call the function that detect push button for Set/Start.

15.2.8.6 Start the timer named debounce.

15.2.8.7 Call the function that detect push button for Minute.

15.2.8.8 Start the timer named debounce.

15.2.8.9 Call the function that detect push button for Second.

15.2.8.10 Call the function to update the LCD screen display.

15.2.9 Turn OFF blue LED.

15.2.10 Create a while loop with condition variable counter equals to 2.

15.2.10.1 Turn ON green LED.

15.2.10.2 Clear the LCD screen display.

15.2.10.3 Display a message “COUNTDOWN START” on LCD.

15.2.10.4 Call the function to update the LCD screen display.

15.2.10.5 Disable the interrupt input of push button for Second.

15.2.10.6 Disable the interrupt input of push button for Minute.

15.2.10.7 If variable counter equals to 2

15.2.10.7.1 Call the function to count down.

15.2.10.7.2 Wait for 1 second.

15.2.10.7.3 Call the function to update the LCD display.

15.2.10.8 If variable counter more than or equals to 3

15.2.10.8.1 Turn OFF green LED.

15.2.10.8.2 Assign variable counter with value 0.

15.2.10.8.3 Assign variable seconds with value 0.

15.2.10.8.4 Assign variable minutes with value 0.

15.2.10.8.5 Clear the LCD screen display.

15.2.10.8.6 Display a message “PRESS TO SET” on LCD.

15.2.10.8.7 Break the outer while loop which is loop with statement counter equals to 2.

15.2.10.9 If variable minutes equals to 0, variable seconds less than or equal to 5 and variable counter equals to 2.

15.2.10.9.1 Assign the red LED to opposite state.

15.2.10.9.2 Call the function for buzzer at last 5 seconds with argument 0.2 (each beep 0.2s).

15.2.10.10 If variable seconds equals to 0, variables minutes equals to 0 and variable counter equals to 2 (countdown end).

15.2.10.10.1 Turn OFF green LED.

15.2.10.10.2 Call the function for buzzer when countdown end with argument 5 and 0.2.

15.2.10.10.3 Assign variable counter to zero.

15.2.10.10.4 Wait for 1 second.

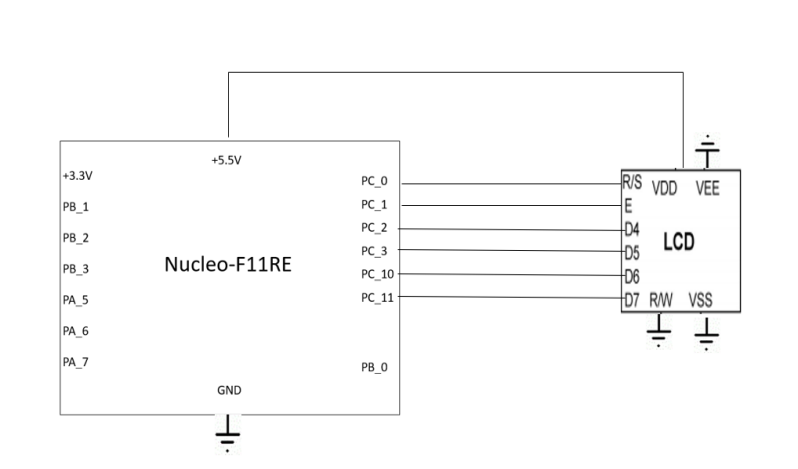
15.2.10.10.5 Break the outer while loop which is loop with statement counter equals to 2.

16.0 End.

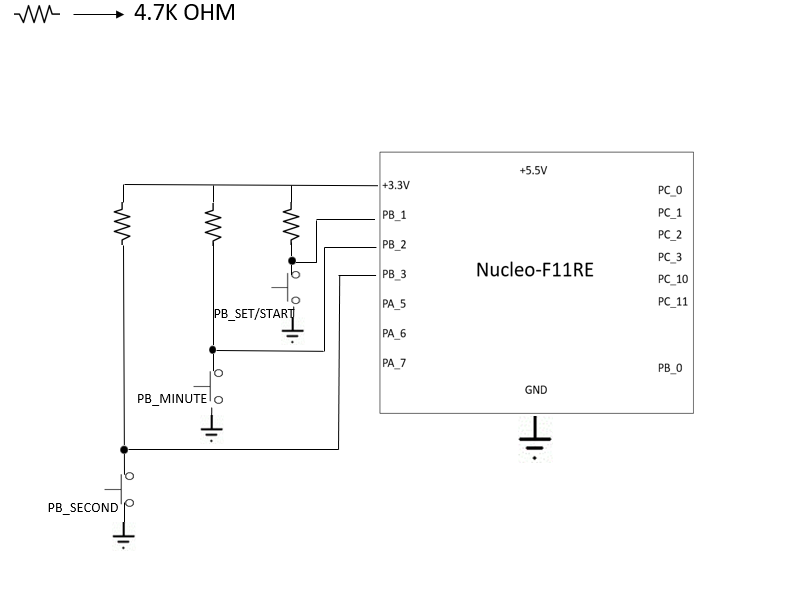
2.2 Schematic Diagram of the Component

So, before we start to do anything on the hardware. We decided to do a simple schematic for 4 type of configuration setup. Where led configuration, LCD board configuration, pushbutton configuration and buzzer configuration. This is because it will be easy to for us to check whether the setup configuration is place in proper way or not . we can solve the particular component configuration by referencing the schematic diagram and check whether the pin or component placement is follow the order of the schematic diagram or not. Basically of the schematic diagram will not problem because we referring back our Lab 6,7,8 for this project. Thus there is least chances of schematic diagram is wrongly done.

1. Schematic Diagram of LCD Board

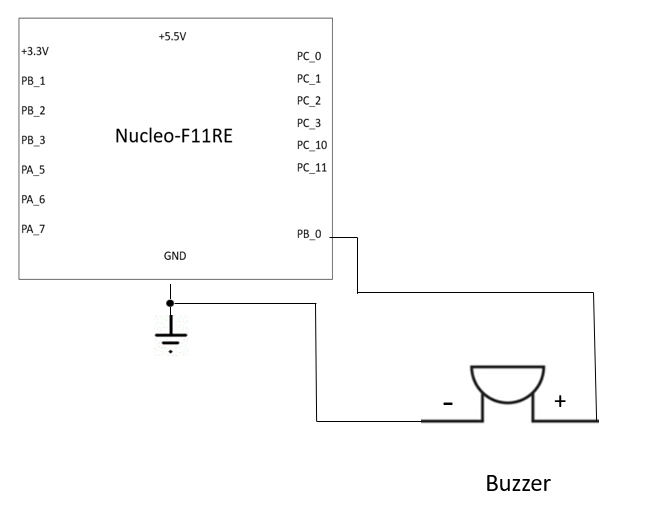


Above figure is LCD board configuration setup where it have mention which pin will be used and then which pin is connected with. So from the figure above we can see that PC 0,1,2,3,10,11 are connected with R/S,E,D4,D5,D6,D7 of the lCD board. Furthermore, we take note that power need to supply and area need to be connected to the LCD board is VDD where the amount of power supply form the nucleo board to LCD board is 5.5V. While the rest which are VEE,R/W,VSS are connected to the ground. Thus, by following this schematic diagram there will be display in the LCD board.

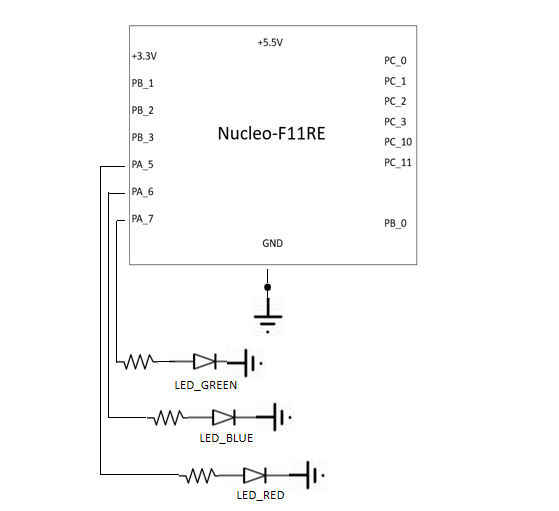
1. Schematic Diagram of Pushbutton

In this project we are required to used 3 type of pushbutton where first pushbutton is set/start the timer.Second and third pushbutton used for set for timer in seconds and minutes.Thus PB 1,2 and 3 will be connected with the pushbutton as the figure above and supply 3.3V for the pushbutton . Beside that we used resistor of 4.7k ohm to avoid overflow current. Then , one leg of the pushbutton will be connected with the ground terminal.

1. Schematic Diagram of Buzzer



So basically the schematic diagram of buzzer. This is additional component in this project where. The function of these buzzer is to emit sound . So basically the connection for this is where GDN need to one negative terminal of the buzzer leg and PB\_0 is connected with positive leg. Thus, if follow this schematic diagram the buzzer will work perfectly fine. Furthermore, we study did extra research on how to connect the the buzzer by referring to youtube.

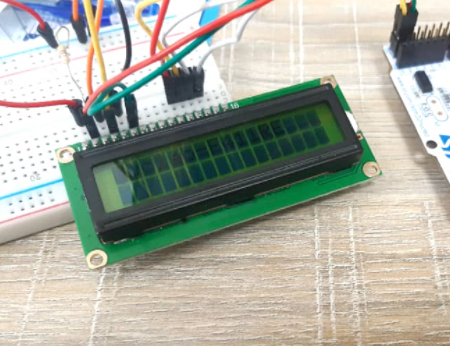
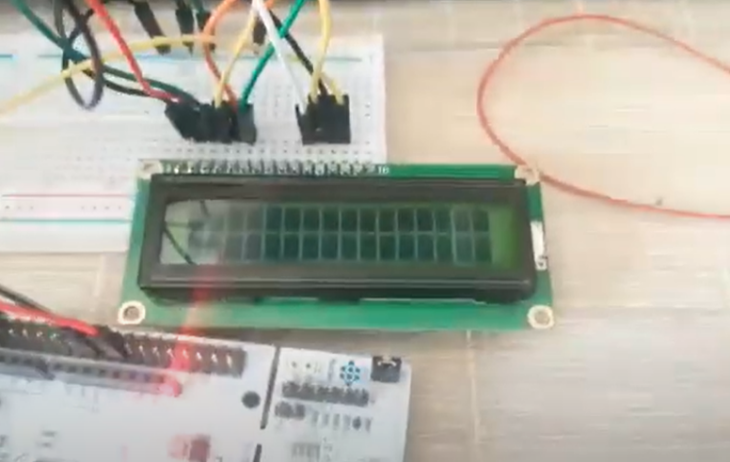
1. Schematic Diagram of LED lights

This is schematic diagram of led configuration. So basically as we can see figure above. We need 3 type of LED light which is blue,red and green led light. It will be connected between resistor and connected with the ground. While pin PA\_5 ,6 ,7 will used in these LED configuration.

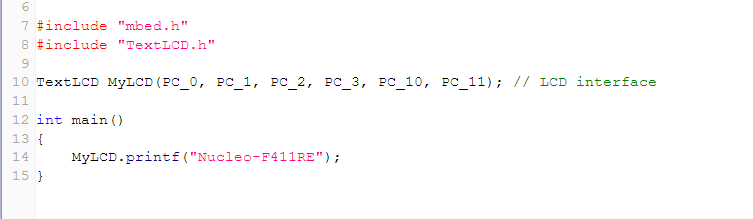
2.3: Setup Component Configuration (Part by Part)

So first we follow the schematic diagram to setup the component parts by parts. This is because we want to test the all the component is working or not and the configuration of the of component is setup in proper way or not.

1. LCD board configuration

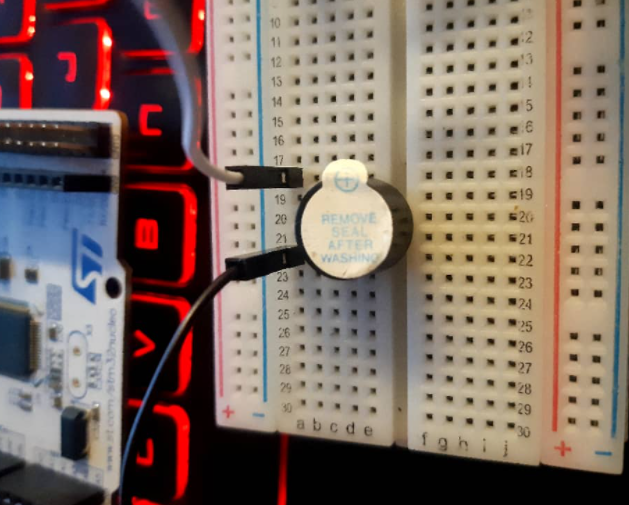


After following the schematic diagram of LCD board we will obtain the it as diagram above and setup the diagram we apply the coding of lab 8. Where lab 8 is about display Output in the LCD board. But using the code in lab 8 we can determine whether the component is working or not. Thus by using the code we obtain that the LCD board can display and it works perfectly fines where can refer figure 2 it display output where “Nucleo-F11RE”. Thus for here we can notice schematic diagram and configuration does not have any problem.

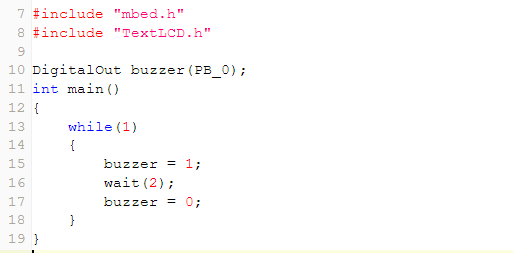


This is coding of lab 8 is been test with the LCD board configuration . So as we can see that figure is about display “Nucleo-F411RE” in the LCD board. We also have declare the pin that we are using in the hardware where PC (0,1,2,3,10 and 11). Thus bu using this code and we can obtain the output as figure 2.

1. Buzzer configuration

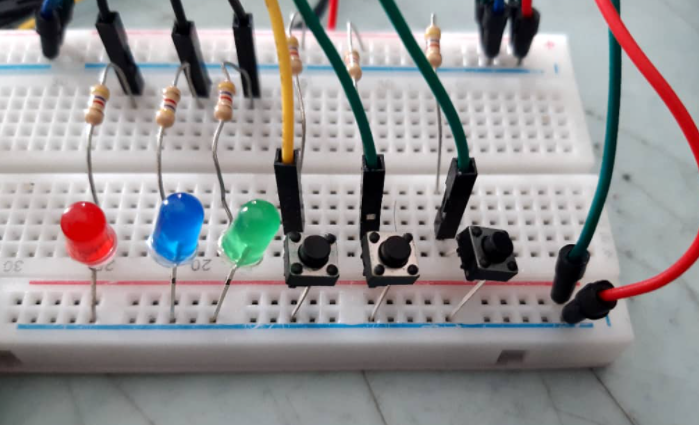


So this how the setup of buzzer after referring the schematic diagram. So basically the left ground pin is connected with negative pin (short leg) while PB\_0 is connected at the right side of the positive leg ( long leg). The function of buzzer is to emit sound . In this project buzzer is just additional component where it help to emit sound when specific program. For example , in our project when the timer reached to zero. Thus the buzzer will emit the sound.

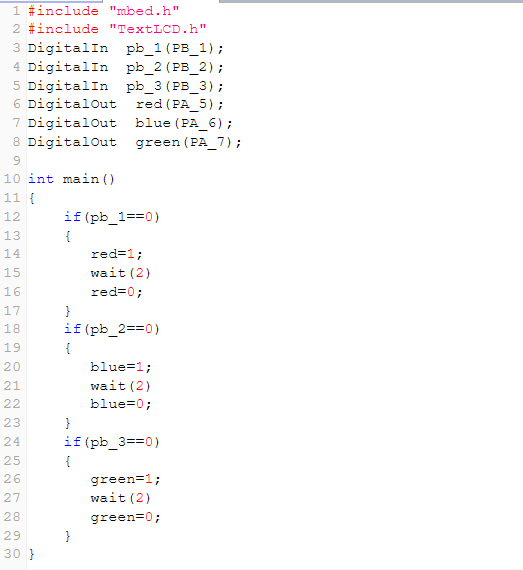


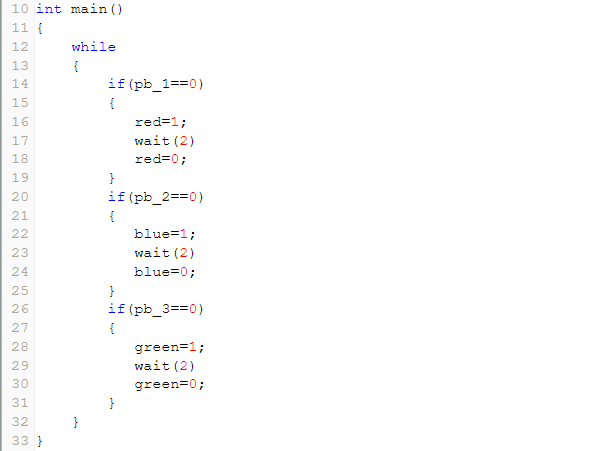
This is coding for the buzzer to emit sounds. So basically we refer this coding in youtube and found out that buzzer component is same with the LED lights. Where is basically is logic 0 and 1. So if is logic 1 which means that the buzzer will emit sound else if logic is 0 it will not emit any sound .So as we refer the coding this coding basically will emit sound for 2 second then turn off the sound and continue the repetition.Thus apply this code in the hardware, we found out that the buzzer is working perfectly fine.

1. LED and switch configuration

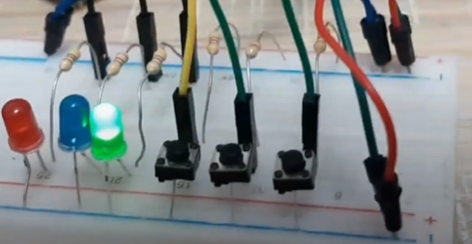


So we have combine switch and Led light configuration . Basically we follow the schematic diagram for the led and switch where combine in one breadboard. Furthermore, we connected the pin according to the schematic diagram where pin of PA\_ 5,6 and 7 which is connected for led lights and PB\_ 1,2 and 3 is connected with the pushbutton. In addition we used resistor of 4.7 K ohm.





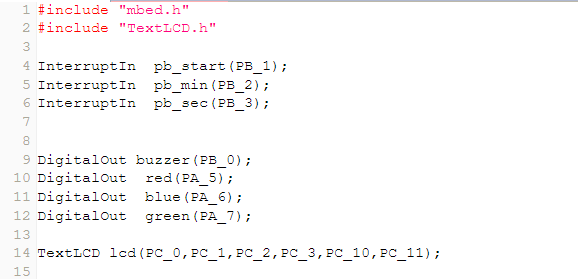
This is coding that we apply for this hardware to check whether the pushbutton and led light is working probably or not and configuration of the is it in proper or not. So from the coding , when pushbutton is pressed thus the logic will 0 this because we are using pull up resistor. Which means that when pushbutton is not pressed thus the logic will be 1 when pressed it will 0. So basically when one of the button is pressed the the particular led which have been assign will turn on for 2 second and then it will one. Then it will wait for next pushbutton pressed.This because it I endless while loop and it will wait next input from the user. We have apply this in the bread board and found that the component and configuration is working perfectly fine which can refer figure below.



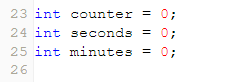
2.4: Used of the Code

1. Function()

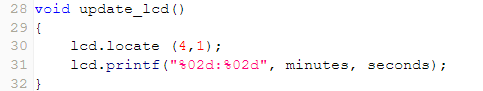
After we done testing all the component configuration , and pseudocode. Now we do the coding for this project which kitchen timer which the project requires us 3 pushbutton, 1 lcd board , timer countdown and increment of time is by 1. Fro example if the user want to set the second for 15 second thus he need to pressbutton second for 15 times. However we will explain the coding function for this kitchen timer project.



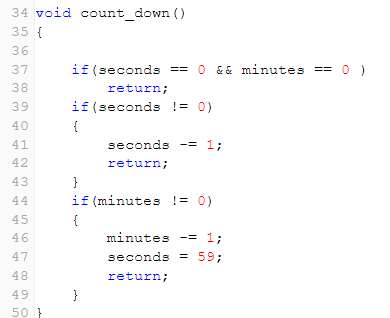
So first thing is we need to include the libraries which is “mbed.h” for electrical components and “TextLCD.h” for the LCD output. Then we declare 3 pushbutton as InterruptIn which known as ( pb\_start,pb\_min,pb\_sec). Furthermore declare the buzzer and led lights (red,blue and green) as DigitalOut . Lastly, declare lcd interface as TextLCD and declares it pin which as figure above.



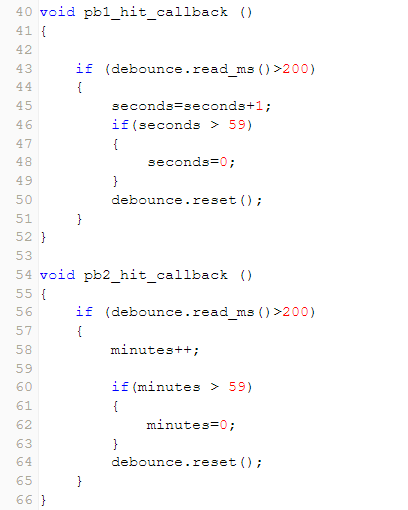
Basically we declare the variable as integer as put as global variable which means that it can be access from any function. Beside that initialize the variable as 0 .



This is update\_lcd function basically will display the and update tge timer clock of the LCD. So lcd.locate (4,1) basically it will display start after 4 boxes in the LCD and lcd.printf basically just display the timer for minutes and seconds.



So figure above is above count down function where the function is to count down the time. So basically we need to used conditional statement for this because there will be 3 type of scenarios. The first if statement is when both minutes and second is 0 thus it will return to the call function . Second if statement is when second is not equal to 0 thus it will be minus by 1. The last statement if minutes if not equal to zero thus it will be minus 1 and second which is 0 will become 59 seconds . Thus this how the count down works.



This 2 function basically same purpose where pb1\_hit\_callback for pushbutton for second. So, what it does is when user pressed the pushbutton for second thus there will increment of the second by 1. While pb2\_hit\_callback for pushbutton of minutes. So basically the function is same with second where user pressed the pushbutton thus minute will increase by 1.

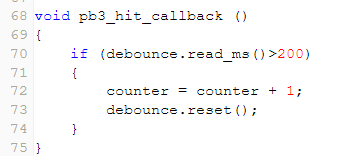
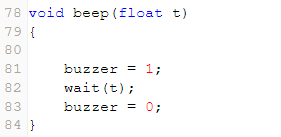
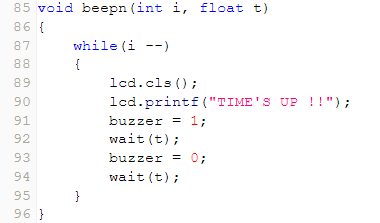


Figure above is about pb3\_hit\_callback function where this function is for set/start pushbutton. So basically of this function is when user pressed pushbutton set/start thus counter will increase by 1. Where this counter have very important role in the int main.

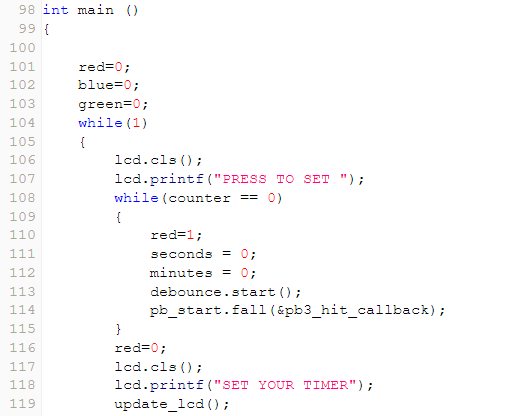


This void beep function where inside the parenthesis has a parameter which float t where it will be used in this function. So basically this function when buzzer is 1 which means that the buzzer will emit sound t second then turn off the buzzer by assign 0.

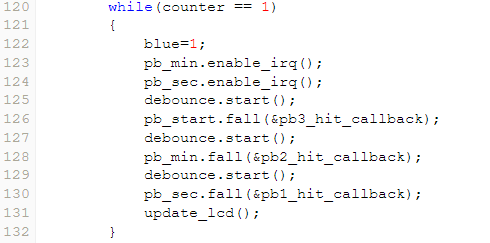


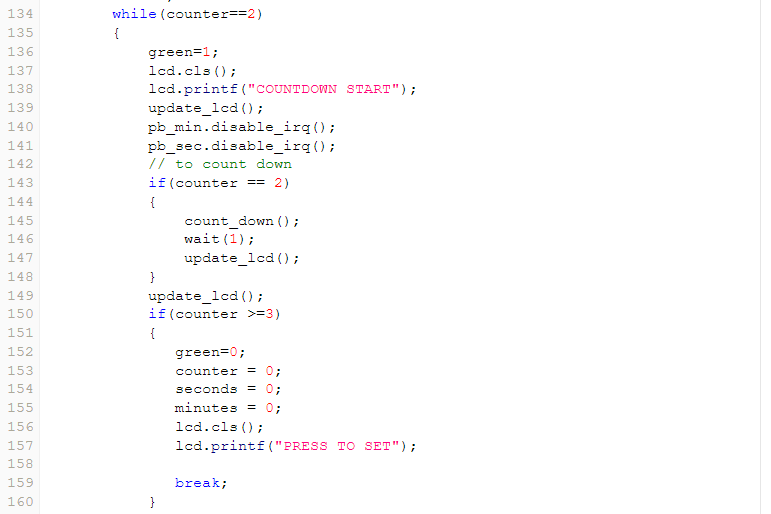
This for beepn function which has 2 parameters in the parenthesis which is int I and float t which will be use in the function . So basically this function will let the buzzer to emit sound for few times which will base of int I . For example, if the int I is 5 thus it will beep for 5 times and LCD will display times up.

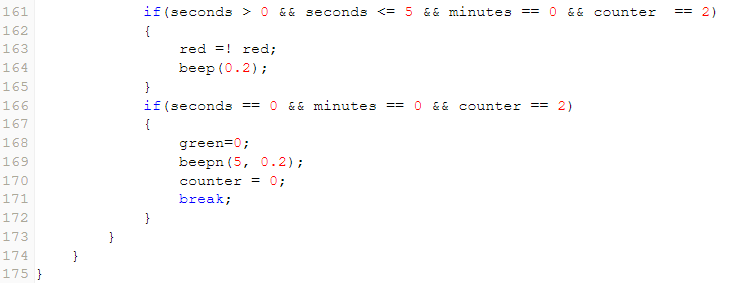
1. Int Main ()



Now comes to the integer main function where all the execution will start from here and will call the rest of the function. So from the figure above line 101 to 103 we assign the led light to 0 which means that to turn off . Next, in the while loop it will display “set the timer” and enter to inner loop where counter ==0 . Thus when it enter the inner loop red led will turn on and minutes and seconds is assign to 0. Next pb\_start.fall () will wait for the user to press the pushutton of set/start. Thus it will keep executing this loop until the user press pushbutton set/start. Once the pushbutton set/start is been pressed thus counter will increase by 1 where it will terminate the while loop of counter ==0. Therefore it will continues executing the next line which it will start from line 116 where red led will turn off and LCD will display “SET YOUR TIMER”.



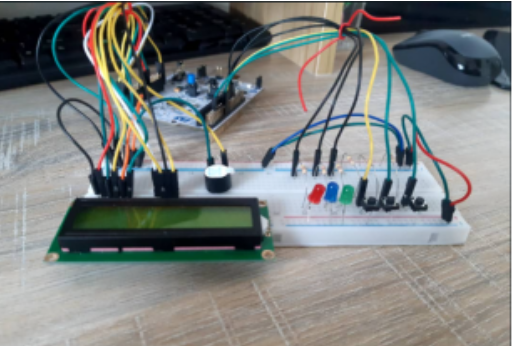
So next it will go to the while loop of counter == 1. So basically in the while loop will set the timer for the countdown. Pushbutton of minute and second change to enable where it can be use in this while loop. While setting the timer there will be indicator where blue led light which will turn on. So basically this blue led will help to identify that the program is asking us to set the timer. So user can keep pressing pushbutton of minute or second according desired of the user. So after set the timer. The user need to press pushbutton of set/start to break the while loop because the counter will be 2 currently. Where counter ==2 for countdown and before it goes to next while loop the blue led will turn of where it assign 0.



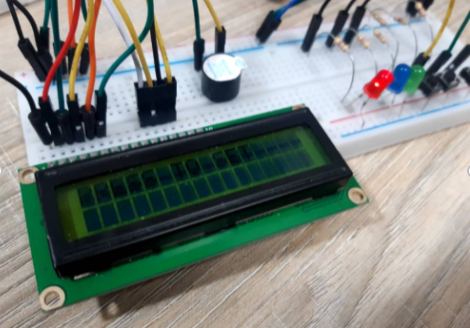
After setting up the time, it will go next while loop where it will start the countdown. So first thing green led will turn on and LCD board will display countdown time . So in this while loop we have disable the pushbutton of minute and second to avoid any interruption during the countdown. In this while loop we used 4 if statement where each if statement have their important function.

So the first if statement where counter ==2. Inside this if statement basically will call the function of count\_down () to countdown the time which we have elaborate the functionality of it. Second if statement where counter>=3. So if the user want to reset the timer thus the user need to pressed the pushbutton of set/start to increase the counter. So basically this state will will make all the re-declare variable value to zero and got back for while of counter 0 and break the while loop of counter ==2. Thus it will go back to intial while loop of counter ==0. Third if statement basically where left 5 more second. Thus this 5 more second will emit beep sound from the buzzer and used red led light to indicate the the countdown is going to finish.

Last if statement where it will beep for 5 times by calling the function of beepn and passing with two parameter which is 5 and 0.2. After executing beepn function the counter will assign as 0 and break the loop. Thus it will the go back from the top of the main. Which means that these program can used for multiple times. So to end the program just unplug the power source.Now we will apply the code in the hardware.So basically we combine all the 4 schematic diagram into 1. which can refer figure below.



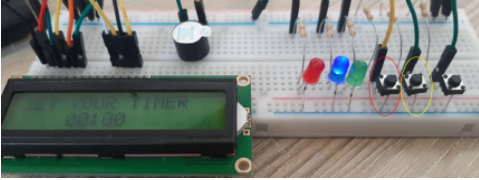
2.5 Implementing Code in the Hardware



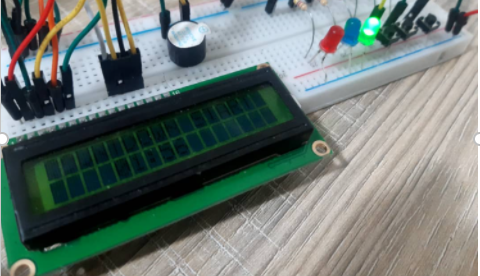
After done doing the coding for kitchen timer. We import the code into a binary file and transfer it to nucleo board.As we can see figure above where it shows red led is turn on. So, the definition of this is to indicate that the program is on and there is nothing happening and waiting for user to set the timer. In same time, the LCD board will display a phrase where “PRESS TO SET” which means that to instruct the user to press pushbutton of set/start to execute the next part. So far as we checked with our coding it same until now. Thus, in our coding when user press set/start button to set the timer the red led will turn off and blue led will turn on and same time LCD display will also change.



When user pressed the set/start button to set the timer. In the circuit there will be changes in term of the led lights and LCD boards. So, the changes are the display of LCD board where it will the countdown “00:00” and phrase where” SET YOUR TIMER”. Besides that, blue led light is turn on while red led light is turn off this because to indicate that the program is now at set timer where it will get the input from the user. So now the user can set their timer by pressing the 2-push button of minutes and second



From the figure above, the red circle is minute pushbutton and yellow circle for second pushbutton so from here user can press the button according to time that user want to set. So, the set timer will increase by 1 for second and minute when every time when user pressed the pushbutton. For example, if the user wan to set 2 minutes and 15 seconds thus the user needs to press the pushbutton 2 times for minutes and 15 times for pushbutton seconds. Thus, it will help to check whether our pushbutton working probably and check the sensitivity of the pushbutton. After set the timer, pressed the start pushbutton to start the countdown of the timer. Thus, there will be changes in term of the hardware where the LCD board will display the countdown timer and green LED light will turn on which indicate as the countdown has started.



So basically, it will continue to do the countdown until it reaches to 00:00. Once it reaches to zero the buzzer will emit a sound that indicates that the countdown is finish or reach to 00:00. These will help for user to be aware where the countdown has finish. Once the countdown done it will go back to “press to set”. Thus, user can set the timer and start countdown again which it will be the same process

In nutshell, the code and hardware is working fine. The order of code in mbed is same with hardware. Where in code press to set and user input the timer and pressed start pushbutton to start the countdown. Where we have try it implementing the hardware and found out that it is in sequence as the code.

3.0 Troubleshooting

3.1 Having problem in the display LCD (00:00)

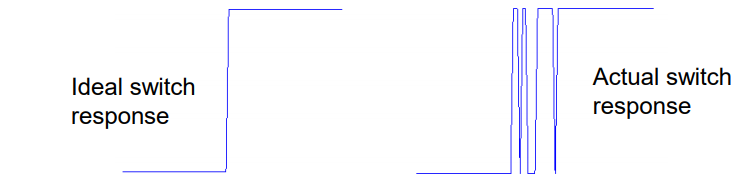
When we first design the code and we want to display the timer on the LCD screen, we used format specifier of %d to display the two integer variables which is minutes and seconds on the LCD screen ("%d:%d", minutes, seconds). However, when we try to run the code, we soon realize that suppose 1 minute 5 seconds will be displayed as 01:05 be it will eventually turn out to be 10:50 on the LCD screen.

After we did some researches on the Internet, we found out that the problem is at the format specifier where we have to change from %d to %02d because the 02d format specifier will make sure the value displayed left padded with zeros up to 2 digits. After I change the line of code from <lcd.printf("%d:%d", minutes, seconds);> to < lcd.printf("%02d:%02d", minutes, seconds);>, now that the time of 1 minute 5 seconds can be displayed on the LCD screen as 01:05 and the problem is solved.

3.2  Push button sensitivity

When we are checking with the component configuration of the push buttons using Lab7. We realize that the push buttons are too sensitive where they will react several times even we just pressed it once. A simple code to detect this problem is <lcd.printf(“%d,Count”);> where we expect the integer variable Count is added by 1 and the value is display on the screen. However, due to the sensitivity of the push buttons, we realize that the value of variable Count did not have increment of 1 only for each press as what we expected. The increment is more than one and varies when we tested for the push button.

After we did some researches from the Internet, we found out that this kind of condition is known as bouncing of push buttons. Below is a diagram showing the condition of bouncing. Bouncing is the condition where the switches/push buttons will actually “bounces” before it change from logic 0 (low) to logic 1 (high). This is the reason why the pushbuttons will increment of have more 1 even we just pressed it once. Therefore, to overcome this problem, we need to debounce it.



The method we used to debounce the push buttons are through software where we edit on the C++ coding to escape from the bouncing period. We make use of the Timer interface in the C++ coding where the Timer interface representing the in-built timer of the nucleo board. Before we pressed the push buttons, the in-built timer will be started. Then, after we pressed the push buttons, it will not react immediately where we designed an if statement where the push buttons will only react when the time passed of the in-built timer is more than 200ms.



Therefore, after we ignore the first 200ms of the responses of the push buttons, we can escape from the bouncing period since we are not reacting with the push buttons during the first 200ms. The push buttons will only response when the time passed of the in-built timer is more than 200ms which is when the switch responses went back normal. The period of bouncing of 200ms is obtained through the method of trial and error. We have tried the of time from 10ms and keep increase the value until we can obtain the exact increment of only one on the variable Count when we were testing with the push buttons. At the end, we found out that 200ms is the best value to choose and the sensitivity of push buttons are now back to normal since we ignored the bouncing period of the push buttons.After each response from the push buttons, we will then reset the timer and the timer will be started again when any of the three push buttons are pressed again.

3.3 Countdown problem because didn’t used return in the coding

In the function that used to count down for the timer, we designed for the decrement of the seconds and minutes and return when both integer variables seconds and minutes are equals to zero. Therefore, there are three if statements in the function where two are for the decrement for variable seconds and variable minutes and the last statement is to return when the count down end. However, we soon realized that the timer did not work like a normal timer. For example, after displayed 02:28, the timer in LCD then display 02:27, 01:59, 01:58, 00:59, 00:58 and so on. Then, we found out that the cause of this problem is at the two if statements for the decrement of seconds and minutes.

This is because we did not return and exit the function after each if statement is executed. Without the return at each if statements, the seconds in timer will be decreased by 1 at first, then followed by the minutes will be decreased by 1 also and seconds will be assigned with value 59. After realized this problem, we then add return void in each if statements so that it quit the function after each statements is executed. The count down of the timer is then back to normal where the seconds will decreases until zero and minutes will be decreased by 1.

3.4 Countdown will be interrupted by the InterruptIn

During the processing of keep testing with our timer, we then realized another problem which is the push buttons for minutes and seconds will interrupt the timer even when the timer is countdown. For example, when the timer is counting, the minutes or seconds will be increased by 1 if I accidentally pressed the push buttons for minutes and seconds since the push buttons are declared as interrupt inputs and they can interrupt the program at any time.

To avoid the timer so we disturbed by the two push buttons for minutes and seconds, we then disable the two push buttons during the process of count down. The function disable\_irq() in the InterruptIn interface is used to disable the push button. Then, we can enable back the push buttons when it comes to the time where we have to set the timer using the function enable\_irq().

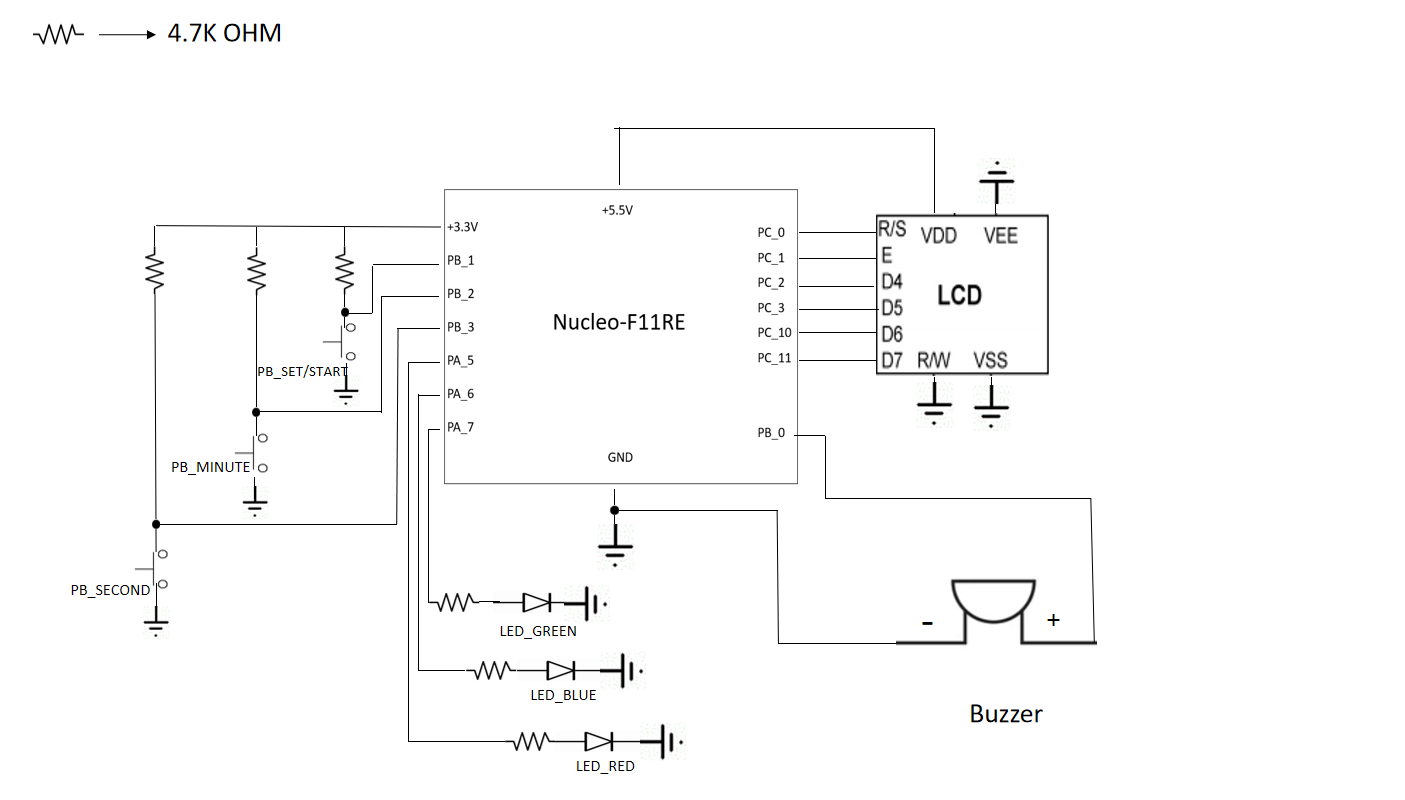
3.5:  Having problem to reset the timer

We also realized that the timer cannot be stopped or reset once we started the count down of the timer. This might be inconvenient especially when we set the wrong time for the timer and accidentally pressed the start button since we cannot reset the timer. In order to overcome this problem, we added an additional features for out kitchen timer. The first pressed on push buttons for Set/Start is to set the timer and the second pressed is to start the timer. Then, we designed the code so that it will allowed the timer to be reset if the push button are pressed more than 2 times. Therefore, if variable counter that will have increment of 1 each time we pressed the push button for Set/Start is more than or equal to 3, we will clear the LCD screen display. All three integer variables which are the seconds, minutes and counter will also be set back to zero. A message “PRESS TO SET” will be displayed on the LCD screen where the user can press the push button to set the timer again.

3.6:   Change the resistor for LED (adjust the brightness)

We have added an additional features for our kitchen timer where we added three colour of LEDs (red, blue, green) that representing different state of the kitchen timer. For example, red LED represents that the timer is not working, blue LED represents the timer is ready to be set and green LED represents the count down of the timer. At first, we used 220ohm resistors and we found out that the LEDs are too bright until it will affect the user’s vision to look at the LCD screen display. Besides that, it will also cause the user to feel uncomfortable with their eyes if the LEDs are too bright. To overcome this problem, we then change the 220ohm to other resistor with higher resistance which is 4.7k ohm resistor. Due to the increase in resistance, the current flow through the LEDs become lower and the level of brightness decreased as well. With this, we can see both LCD screen display and also the LEDs light clearly and comfortably.

4.0 Schematic Diagram



5.0 Coding

#include "mbed.h"

#include "TextLCD.h"

// INPUTS

// Connect three buttons input on pin PB1,PB2,PB3

// Use InterruptIn so that we can interrupt at any time when the program is executing

InterruptIn pb\_start(PB\_1);

InterruptIn pb\_min(PB\_2);

InterruptIn pb\_sec(PB\_3);

// OUTPUTS

// Connect a buzzer as output on pin PB0

DigitalOut buzzer(PB\_0);

// Connect three LEDs as output on pin PA5,PA6,PA7

DigitalOut red(PA\_5); // When the timer is not using

DigitalOut blue(PA\_6); // When setting the timer

DigitalOut green(PA\_7); // When counting down

// Pin connection from nucleo to LCD

TextLCD lcd(PC\_0,PC\_1,PC\_2,PC\_3,PC\_10,PC\_11);

// The Timer interface is used to create, start, stop and read a timer for measuring small times

// >> built-in timer for real time interrupt

// >> implementing a debounce counter to avoid multiple interrupts

Timer debounce;

// Declare all the integer variables that used in this project & initialize to 0

int counter = 0;

int seconds = 0;

int minutes = 0;

// Create a funtion to display & update the timer clock in LCD

void update\_lcd()

{

lcd.locate (4,1);

// the display will start after 4 boxes in the LCD （\_ \_ \_ \_ 00:00)

// display down (second line) on LCD

lcd.printf("%02d:%02d", minutes, seconds);

// 2d so that 09,08,07,.. can be displayed >> >> left padded with zeros up to 2 digits.

// if only %d, it will become 90,80,70,...

}

// Create a function that will do the count down for the timer

void count\_down()

{

// First, check whether the timer end

// Second, check whether the second is 0

// Last, check whether the minute is 0

// if any of these is executed, it will return and will not go to other if statement

// if both sec and min reach 0 >> timer end and return

if(seconds == 0 && minutes == 0 )

return;

// if both sec not 0 >> second -1

if(seconds != 0)

{

seconds -= 1;

return;

}

// if both sec reach 0 and minute not 0 >> min -1, sec=59

// for example, 02:00 -> 01:59

if(minutes != 0)

{

minutes -= 1;

seconds = 59;

return;

}

}

// pb1 = pushbutton for seconds

// This function will be used when setting the timer for seconds

// for each time this function executed, vairbale seconds will be increased by 1

void pb1\_hit\_callback ()

{

// .read\_ms() is to get the time passed in ms

// this is used becuase the pushbutton will "bounces" in a very short time after we click the pushbutton

// so >200 is to escape from the period of bouncing

if (debounce.read\_ms()>200)

{

seconds=seconds+1;

if(seconds > 59)

{

seconds=0;

}

// reset the built-in timer to zero

debounce.reset();

}

}

// pb2= pushbutton for minutes

// This function will be used when setting the timer for minutes

// for each time this function executed, vairbale minutes will be increased by 1

void pb2\_hit\_callback ()

{

if (debounce.read\_ms()>200)

{

minutes++;

if(minutes > 59)

{

minutes=0;

}

debounce.reset();

}

}

// pb3 pushbutton for Set/Start

// This function will be used to run/control the timer

// for each time this function executed, vairbale counter will be increased by 1

void pb3\_hit\_callback ()

{

if (debounce.read\_ms()>200)

{

counter = counter + 1;

debounce.reset();

}

}

// this function is used for the buzzer in the last 5 seconds of timer count down

void beep(float t)

{

// Buzzer buzz for every second follow the timer

buzzer = 1;

wait(t);

buzzer = 0;

}

// this function is used for the buzzer when the timer finish countdown 00:00

// i is the number of times to buzz

// t is interval of time

void beepn(int i, float t)

{

// Buzzer buzz for t seconds

while(i --)

{

lcd.cls();

lcd.printf("TIME'S UP !!");

buzzer = 1;

wait(t);

buzzer = 0;

wait(t);

}

}

int main ()

{

// Initialize all the LEDs to be OFF

red=0;

blue=0;

green=0;

// endless loop

while(1)

{

// clear the LCD screen

lcd.cls();

lcd.printf("PRESS TO SET ");

// counter 0 >> BEFORE pressed for the timer to set

while(counter == 0)

{

red=1;

seconds = 0;

minutes = 0;

// start the timer

debounce.start();

// only pb\_start is avaible here

// therefore, pressing pb for min and sec will not respond

pb\_start.fall(&pb3\_hit\_callback);

// fall() is to attach a function to call when a falling edge occurs on the input

}

red=0;

lcd.cls();

lcd.printf("SET YOUR TIMER");

update\_lcd();

// counter 1 >> AFTER pressed to set the timer

while(counter == 1)

{

blue=1;

// enable\_irq is the function for InterruptIn to enable the electronics

// enable pb for min and sec

pb\_min.enable\_irq();

pb\_sec.enable\_irq();

debounce.start();

pb\_start.fall(&pb3\_hit\_callback); // start the timer

debounce.start();

pb\_min.fall(&pb2\_hit\_callback); // increase 1 for min

debounce.start();

pb\_sec.fall(&pb1\_hit\_callback); // increase 1 for sec

update\_lcd();

// After every click on pb, update the LCD

}

blue=0;

// counter 2 >> the timer START count down

while(counter==2)

{

green=1;

lcd.cls();

lcd.printf("COUNTDOWN START");

update\_lcd();

// disable\_irq is the function in InterruptIn

// disable the pb for min and sec so that it will not interrupt the timer during count down

pb\_min.disable\_irq();

pb\_sec.disable\_irq();

// to count down

if(counter == 2)

{

count\_down();

wait(1);

update\_lcd();

}

update\_lcd();

// to reset (in case user want to stop the timer during count down)

if(counter >=3)

{

green=0;

counter = 0; // set back the counter to 0 (before SET timer)

seconds = 0;

minutes = 0;

lcd.cls();

lcd.printf("PRESS TO SET");

break; // break the outer while loop so that the program can go back to while(counter == 0)

}

// for buzzer during the last 5 seconds (buzz for 5 times)

if(seconds > 0 && seconds <= 5 && minutes == 0 && counter == 2)

{

red =! red; // blinking for red led means last 5 second

beep(0.2); // 0.2s is the interval

}

// for buzzer when the timer finish

if(seconds == 0 && minutes == 0 && counter == 2)

{

green=0; // done counting

beepn(5, 0.2); //5 is buzz for 5 times, interval of 0.2s

counter = 0;

wait(1.0);

break;

}

}

}

}

6.0 Video Project

<https://www.youtube.com/watch?v=9L__K-9A0UM>

<https://www.youtube.com/watch?v=wZozc8o0g7c>

<https://www.youtube.com/watch?v=ENDs9Dbf0a0>