VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Electronics Engineering (SENSE)

B. Tech – Electronics & Communication Engineering

BECE403E – EMBEDDED SYSTEMS DESIGN

LAB RECORD

(lab slot L27+L28)

Submitted By

22BEC1205 – Jayakrishnan Menon

Submitted To

Dr. S.Muthulakshmi

DATE: 24/03/2025

Slot: L27+L28

Date: 24/03/2025

LAB – 10: Working with 7 Segment Display

AIM: To understand the interfacing of a 7 segment display with Nucleo64-STM32L152RE Board for various applications. We will also perform the following tasks:

Lab Task-1: Write a mbed C++ program to display decimal values from 0 to 9 on the 7-segment display interfaced with digital pins of Nucleo boards. Once the sequence completed from 0 to 9 then repeat the sequence again. Design and implement this logic on the STM 32 Nucleo L152RE board using Keil studio could platform.

Lab Task-2: Write a mbed C++ program to design a token display system which helps to serve the customer without the need of standing in a queue. This display system consist of one 7-segment display to display token numbers from 0 to 9, and token numbers are received into Nucleo board via 4x3 keypad. Design and implement this logic on the STM 32 Nucleo L152RE board using Keil studio could platform.

SOFTWARE REQUIRED: ARM Keil Studio (Mbed Online Compiler), Tera Term

HARDWARE REQUIRED: Micro USB cable, NUCLEO64-STM32L152 Board, Jumper Wires (M-F and M-M), Breadboard, 7 segment display, keypad

PROCEDURE:

1. Go to ARM Keil Studio (https://studio.keil.arm.com) and log in
2. Select File → New → Mbed Project
3. Click the Example project drop-down list and select "mbed2-example-blinky"
4. In Project name field, provide the name of the new project and click Add project
5. Double click on the "main.cpp" file from the newly created project folder
6. Modify the code in the editor window as per the logic of your application
7. Check for any errors in the program under the "Problems" tab of the panels window
8. If no errors, connect the Nucleo Board to the computer using Micro USB Cable
9. Click Play icon (Run project) to upload and start the code execution on the board.

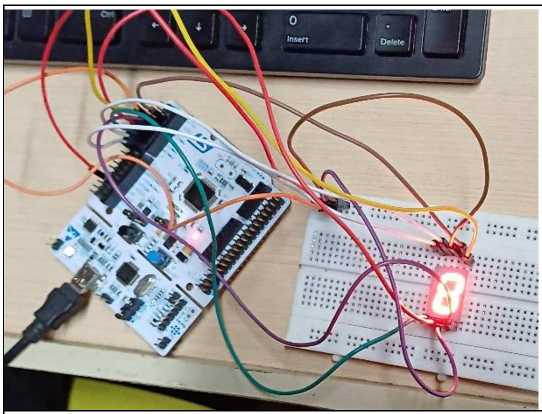
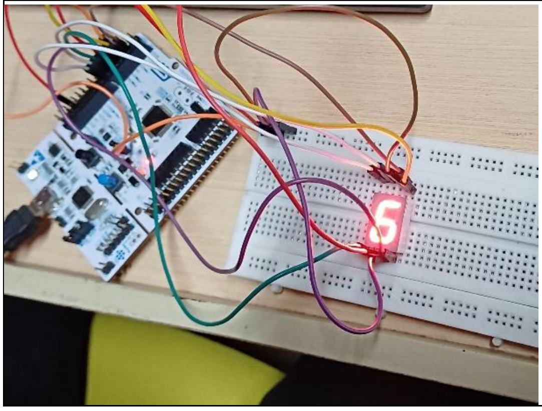
PROGRAMS:

Lab Task 1: Write a mbed C++ program to display decimal values from 0 to 9 on the 7-segment display interfaced with digital pins of Nucleo boards. Once the sequence completed from 0 to 9 then repeat the sequence again. Design and implement this logic on the STM 32 Nucleo L152RE board using Keil studio could platform.

Code:

```
#include "mbed.h"
BusOut display (PA_10,PB_3,PB_5,PB_4,PB_10,PA_8,PA_9,PC_7);

int main()
{
    while(1)
    {
        for(int i=0; i<=16; i++){
            switch(i)
            {
                case 0: display=0xc0; break;
                case 1: display=0xf9; break;
                case 2: display=0xa4; break;
                case 3: display=0xb0; break;
                case 4: display=0x99; break;
                case 5: display=0x92; break;
                case 6: display=0x82; break;
                case 7: display=0xf8; break;
                case 8: display=0x80; break;
                case 9: display=0x90; break;
                case 10: display=0x88; break;
                case 11: display=0x83; break;
                case 12: display=0xc6; break;
                case 13: display=0xa1; break;
                case 14: display=0x86; break;
                case 15: display=0x8e; break;
            }
            wait(0.5);
        }
    }
}
```

Output:

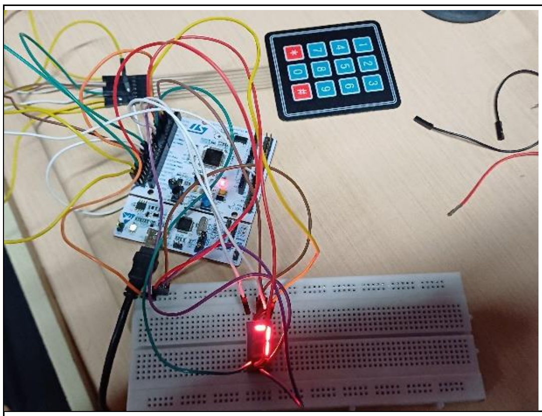
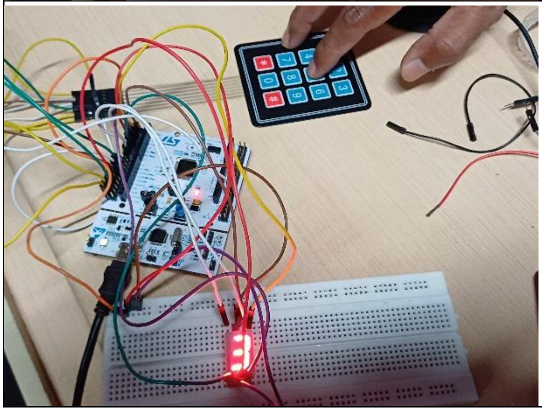
	**Fig. 1.1:** Number ‘8’ displayed on the 7 segment display.
	**Fig. 1.2:** Number ‘6’ displayed on the 7 segment display.

Lab Task 2: Write a mbed C++ program to design a token display system which helps to serve the customer without the need of standing in a queue. This display system consist of one 7-segment display to display token numbers from 0 to 9, and token numbers are received into Nucleo board via 4x3 keypad. Design and implement this logic on the STM 32 Nucleo L152RE board using Keil studio could platform.

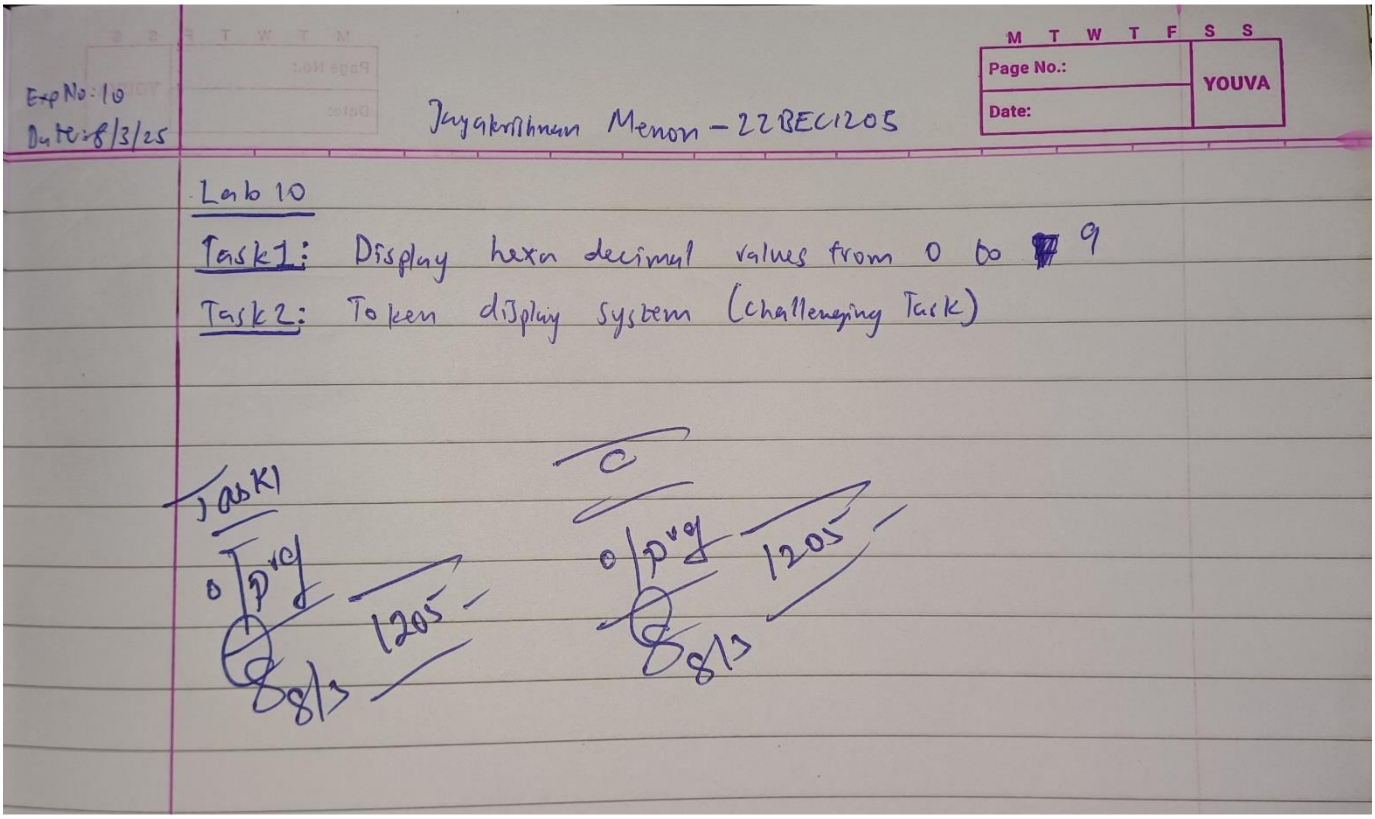
Code:

```
#include "mbed.h"
#include "keypad.h"
Keypad keypad (PC_0, PC_1, PB_0, PC_8, PA_4, PA_1, PA_0);
BusOut display (PA_10,PB_3,PB_5,PB_4,PB_10,PA_8,PA_9,PC_7);

char i;
int main()
{
    keypad.enablePullUp();
    while(1)
    {
        while (keypad.getKey() == '\0');
        i = keypad.getKey();
            switch(i)
            {
                case 0: display=0xc0; break;
                case 1: display=0xf9; break;
                case 2: display=0xa4; break;
                case 3: display=0xb0; break;
                case 4: display=0x99; break;
                case 5: display=0x92; break;
                case 6: display=0x82; break;
                case 7: display=0xf8; break;
                case 8: display=0x80; break;
                case 9: display=0x90; break;
                case 10: display=0x88; break;
                case 11: display=0x83; break;
                case 12: display=0xc6; break;
                case 13: display=0xa1; break;
                case 14: display=0x86; break;
                case 15: display=0x8e; break;
            }
            wait(0.5);

    }
}
```

Output:

	**Fig. 2.1:** Number '7' displayed on the 7 segment display when the '7' key is pressed.
	**Fig. 2.2:** Number '3' displayed on the 7 segment display when the '3' key is pressed.

OUTPUT VERIFICATION:



M T W T F S S

Page No.:

Date:

YOUVA

Exp No: 10
Date: 8/3/25

Jayakrishnan Menon - 22BEC1205

Lab 10

Task1: Display hexa decimal values from 0 to ~~F~~ 9

Task2: Token display system (challenging Task)

Task1

o/p vd

1205

8/3

C

o/p vd

1205

8/3

INFERENCE:

1. The Keypad.h library is designed to work with various keypad configurations, including 4x4, 3x4, and other matrix layouts, making it versatile for different applications.
2. The getKey() method is the core function for reading keypad input. It scans the keypad and returns the character corresponding to the pressed key.
3. void enablePullUp() Enables internal PullUp resistors on the columns pins.
4. Using the “BusOut” API, we can combine a number of “DigitalOut” pins to write values on all of them using a single statement. Its Syntax is: “BusOut Identifier(PinNames)”.
5. The “identifier” of a “BusOut” can be assigned values, inside the main function. Through this, the user can assign different output values to the Output Bus, based on the requirements. It is possible to assign hexadecimal values (eg: 0x08) or decimal values (eg: 8) to it.
6. Programmable Delays can be implemented through the use of the “wait()” function.
7. A program can be set to run indefinitely using the “while(1)” loop.

RESULT:

Thus, the interfacing of an Keypad with Nucleo64-STM32L152RE Board for applications involving the measure of distances and proximity was understood and the tasks were also performed successfully.