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: 18/12/2024

Slot: L27+L28

Date: 13/12/2024

LAB – 01: LED Blinking

AIM: To understand the working of Nucleo64-STM32L152RE Board and to perform the following tasks:

Lab Task-1: Blinking of LED on Board
Lab Task-2: Blinking of LEDs alternatively in pairs (1010, 0101)
Lab Task-3: Blinking one LED at a time serially (1000, 0100, 0010, 0001)

SOFTWARE REQUIRED: ARM Keil Studio (Mbed Online Compiler)

HARDWARE REQUIRED: Micro USB cable, NUCLEO64-STM32L152 Board, LEDs, Jumper Wires (M-F and M-M), Breadboard

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.

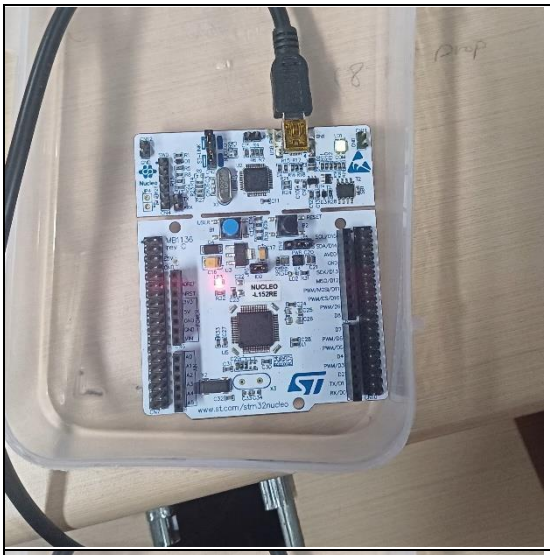
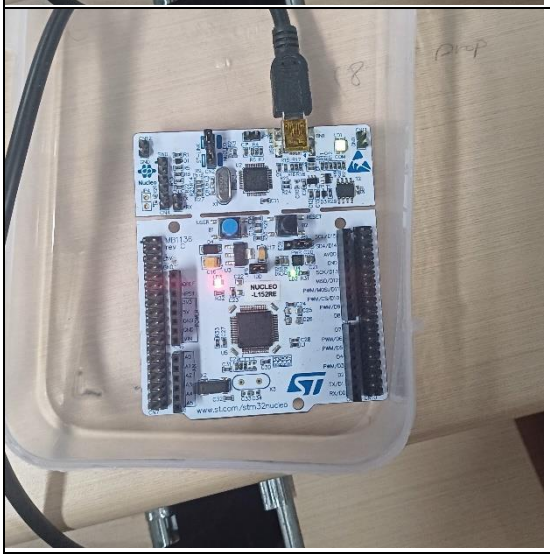
PROGRAMS:

Lab Task 1: Blinking of LED on Board

Code:

```
#include "mbed.h"

DigitalOut myled(LED1);

int main() {
  while(1) {
    myled = 1;
    wait(0.5);
    myled = 0;
    wait(0.5);
  }
}
```

Output:

	**Fig. 1.1:** LED on the Board is in the OFF state for 0.5 seconds
	**Fig. 1.2:** LED on the Board is in the ON state for 0.5 seconds

Lab Task 2: Blinking of LEDs alternatively in pairs (1010, 0101)

Code:

```
#include "mbed.h"

DigitalOut myled1(PC_4);
DigitalOut myled2(PB_13);
DigitalOut myled3(PB_14);
DigitalOut myled4(PB_15);

int main() {
  while(1) {
    myled1 = 1;
    myled2 = 0;
    myled3 = 1;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 1;
    myled3 = 0;
    myled4 = 1;
    wait(0.5);
  }
}
```

Output:

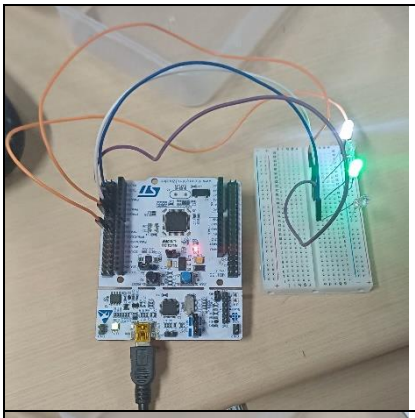


Fig. 2.1: 1st and 3rd LEDs from the top are in ON state and 2nd and 4th LEDs from the top are in OFF state for 0.5 seconds

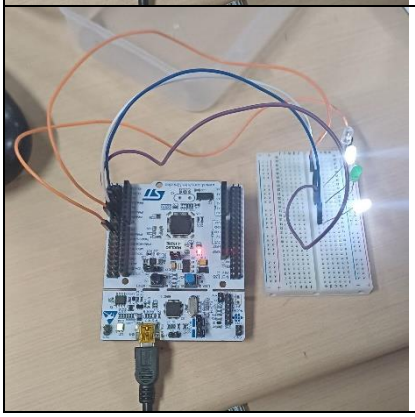


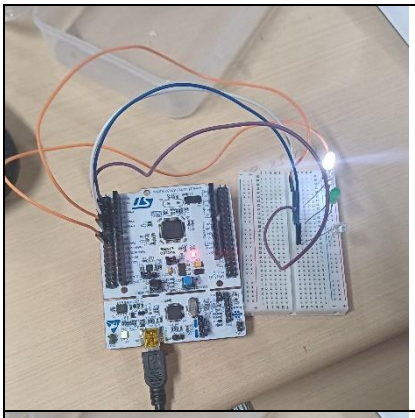
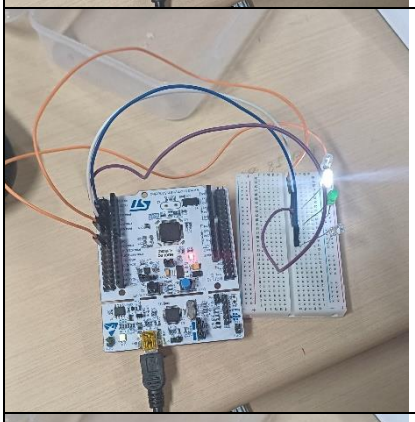
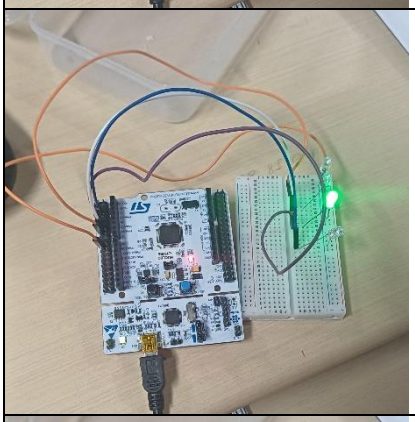
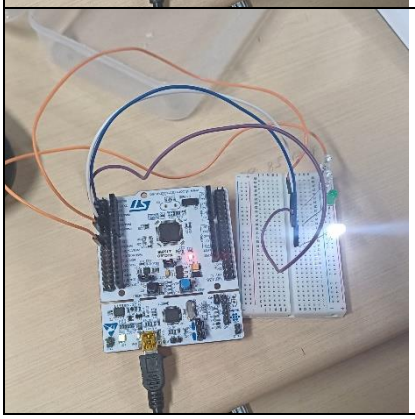
Fig. 2.2: 1st and 3rd LEDs from the top are in OFF state and 2nd and 4th LEDs from the top are in ON state for 0.5 seconds

Lab Task 3: Blinking one LED at a time serially (1000, 0100, 0010, 0001)

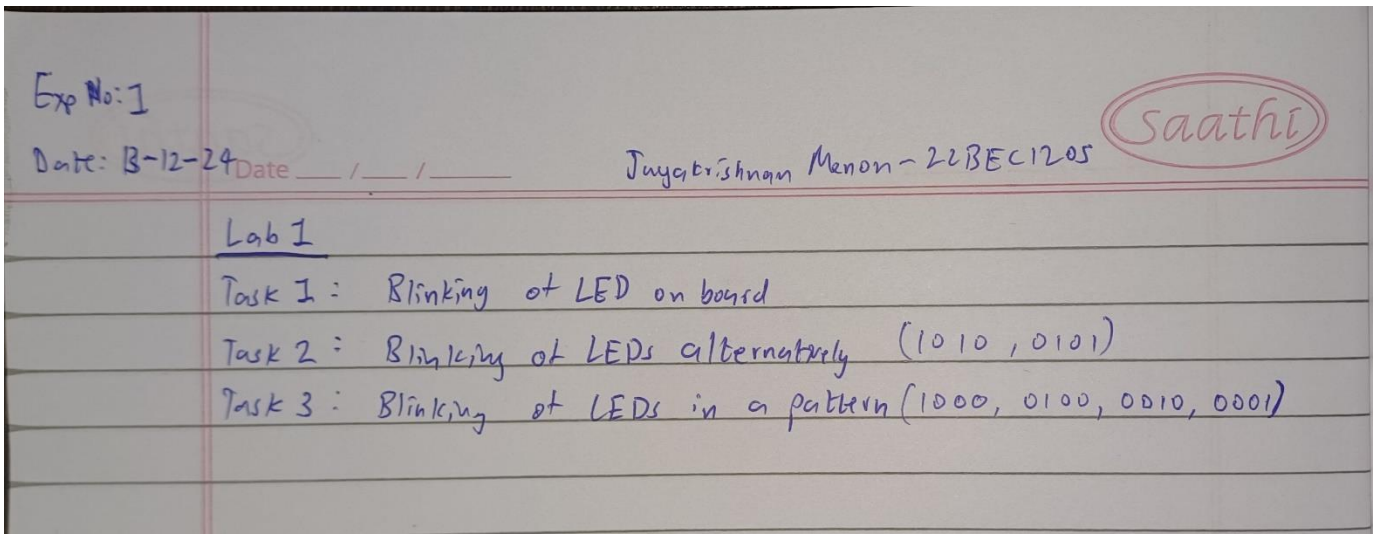
Code:

```
#include "mbed.h"

DigitalOut myled1(PC_4);
DigitalOut myled2(PB_13);
DigitalOut myled3(PB_14);
DigitalOut myled4(PB_15);

int main() {
  while(1) {
    myled1 = 1;
    myled2 = 0;
    myled3 = 0;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 1;
    myled3 = 0;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 0;
    myled3 = 1;
    myled4 = 0;
    wait(0.5);
    myled1 = 0;
    myled2 = 0;
    myled3 = 0;
    myled4 = 1;
    wait(0.5);
  }
}
```

Output:

	**Fig. 3.1:** 1st LED from the top is in ON state for 0.5 seconds, while the rest are in OFF state
	**Fig. 3.2:** 2nd LED from the top is in ON state for 0.5 seconds, while the rest are in OFF state
	**Fig. 3.3:** 3rd LED from the top is in ON state for 0.5 seconds, while the rest are in OFF state
	**Fig. 3.4:** 4th LED from the top is in ON state for 0.5 seconds, while the rest are in OFF state

OUTPUT VERIFICATION:



INFERENCE:

1. The header file “mbed.h” must be included while programming the Nucleo64 Board.
2. Pins and the on-board LED can be declared as Output through the use of the “DigitalOut” class, by passing the name of the pin (or LED) through an object of the same class. Here, “myled”, “myled1”, “myled2”, “myled3”, “myled4” objects were used in programming.
3. The main program to be executed is written inside the main function and it returns some value of the integer datatype.
4. If we want our program to run indefinitely on our board, we need to write it inside a “while(1)” loop. A while loop executes as long as the argument provided within its parenthesis is true. In while(1), the argument provided is ‘1’ which is always ‘true’ as it is not a ‘zero’ (‘0’ means false)
5. Inside the main function we can control the state of the previously declared Digital Output pins by equating the object corresponding to it as either ‘1’ or ‘0’. This feature is available only due to abstraction from regular/standard object-oriented programming. “myled” is not an ordinary variable, it is an object of the class “DigitalOut”.
6. Programmable Delays can be implemented through the use of the “wait()” function.

RESULT:

Thus, the working of the Nucleo64-STM32L152RE Board was understood and the tasks were performed successfully.