AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH

Faculty of Engineering

Lab Report

Experiment # 02

Title: Familiarization with an STM32, the study of blink test and implementation of a light-controlling system using microcontrollers.

Date of Perform	29 January 2023	**Date of Submission**	05 February 2023
Course Title	MICROPROCESSOR AND EMBEDDED SYSTEMS		
Course Code	COE3104	**Section**	C
Semester	Spring 2022-23	**Degree Program**	BSc in CSE & EEE
Course Teacher	**RICHARD VICTOR BISWAS**		

Declaration and Statement of Authorship:

1. I/we hold a copy of this Assignment/Case-Study, which can be produced if the original is lost/damaged.
2. This Assignment/Case-Study is my/our original work and no part of it has been copied from any other student's work or from any other source except where due acknowledgment is made.
3. No part of this Assignment/Case-Study has been written for me/us by any other person except where such collaboration has been authorized by the concerned teacher and is clearly acknowledged in the assignment.
4. I/we have not previously submitted or currently submitting this work for any other course/unit.
5. This work may be reproduced, communicated, compared, and archived for the purpose of detecting plagiarism.
6. I/we give permission for a copy of my/our marked work to be retained by the Faculty Member for review by any internal/external examiners.
7. I/we understand that Plagiarism is the presentation of the work, idea, or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offense that may lead to expulsion from the University. Plagiarized material can be drawn from, and presented in, written, graphic and visual forms, including electronic data, and oral presentations. Plagiarism occurs when the origin of the source is not appropriately cited.
8. I/we also understand that enabling plagiarism is the act of assisting or allowing another person to plagiarize or copy my/our work.

* Student(s) must complete all details except the faculty use part.
** Please submit all assignments to your course teacher or the office of the concerned teacher.

Group # 01

Sl No	Name	ID	Department
1	Zaid Amin **Rawfin**	20-42459-1	CSE

Faculty use only		
FACULTY COMMENTS	**Marks Obtained**	
	Total Marks	

Table of Contents

Experiment Title 3
Abstract 3
Introduction: 3
Theory and Methodology 3
Experimental Procedure 3
Equipment List 4
Circuit Diagram 4
Code & Explanation 6
Hardware Output Results 6
Simulation Output Result 8
Discussion & Conclusion 9
References 9

Experiment Title

Familiarization with an STM32, the study of blink test and implementation of a light-controlling system using microcontrollers.

Abstract

STM32CubeIDE is a comprehensive C/C++ development environment for STM32 microcontrollers and microprocessors that offers various advanced features such as peripheral configuration, code generation, code compilation, and debugging. The build and stack analyzer tools in STM32CubeIDE provide valuable information about the project's status and memory requirements. The debugging tools included are both standard and advanced, including views of core CPU registers, memories, peripheral registers, live variable watch, Serial Wire Viewer, and a fault analyzer.

Introduction:

The purpose of this experiment is to familiarize oneself with the STM32 Microcontroller through the blink test and implementing a traffic control system. The experiment uses the STM32 CUBE IDE and Proteus software for hardware setup and simulation, respectively. The objective is to gain an understanding of the microcontroller. The main objectives of this experiment are:

- Learning to make the LED blink using ST32.
- Implementation of a light control system using STM32.

Theory and Methodology

STM32CubeIDE is an advanced C/C++ development platform with peripheral configuration, code generation, code compilation, and debug features for STM32 microcontrollers and microprocessors. STM32CubeIDE includes build and stack analyzers that provide the user with useful information about project status and memory requirements.STM32CubeIDE also includes standard and advanced debugging features including views of CPU core registers, memories, and peripheral registers, as well as a live variable watch, Serial Wire Viewer interface, or fault analyzer.

Experimental Procedure

1. To begin, the process of blinking an LED light was studied and understood. This served as the foundation for the light control system implementation.
2. A circuit was constructed using the specified connection system between all elements, including the LED light, power source, and any additional components.
3. The STM32 Nucleo board was connected to the PC to serve as the central control system for the light control. This allowed communication between the PC and the circuit.
4. The connection between the STM32 Nucleo board and the PC enabled the PC to send commands to the circuit and control the LED light.
5. The knowledge gained from steps 1 and 2 was applied to implement the light control system, resulting in the desired blinking of the LED light.

Equipment List

1. STM32 Nucleo-F401RE Board
2. Breadboard
3. LED lights (red, yellow, green)
4. Jumper wires
5. Computer

Circuit Diagram

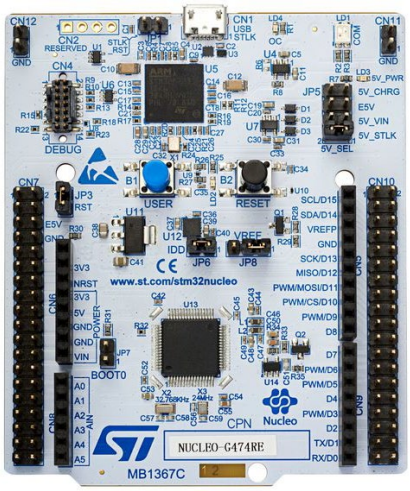

Fig. 1 STM32 Nucleo-F401RE Board

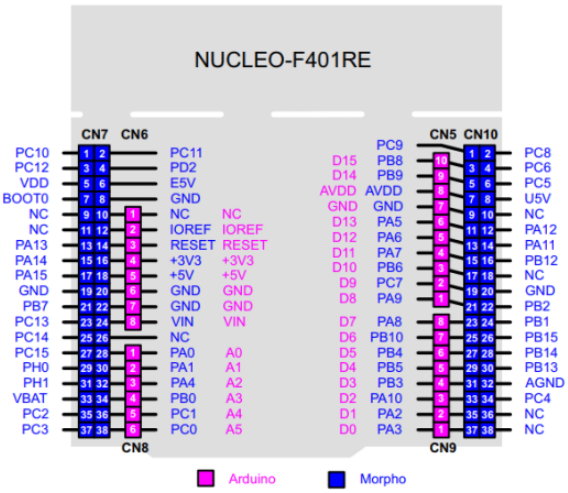

Fig 2. Arduino Uno (R3)

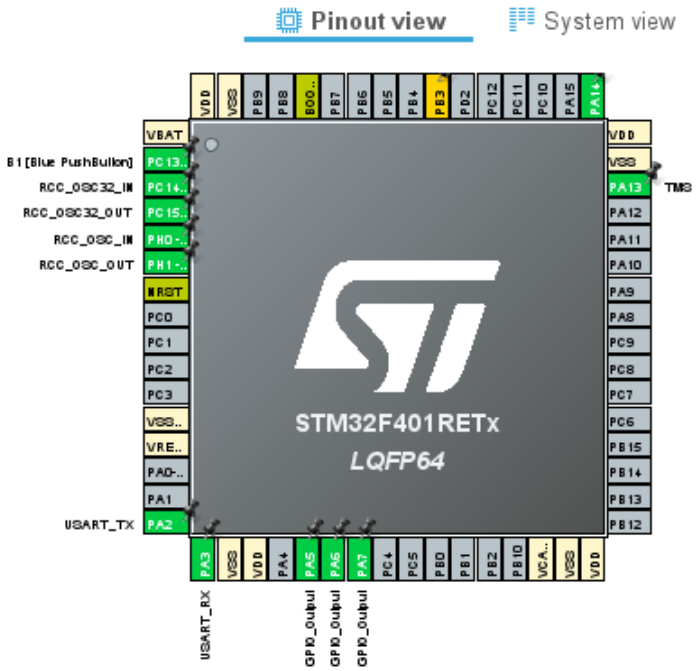

Fig 03: Pin setup

Here, the Red, Yellow, and Green LEDs are coupled to the STM32F401RET microcontroller's pins 5, 6, and 7, that are used and configured in this case as output pins.

Hardware Set-Up:

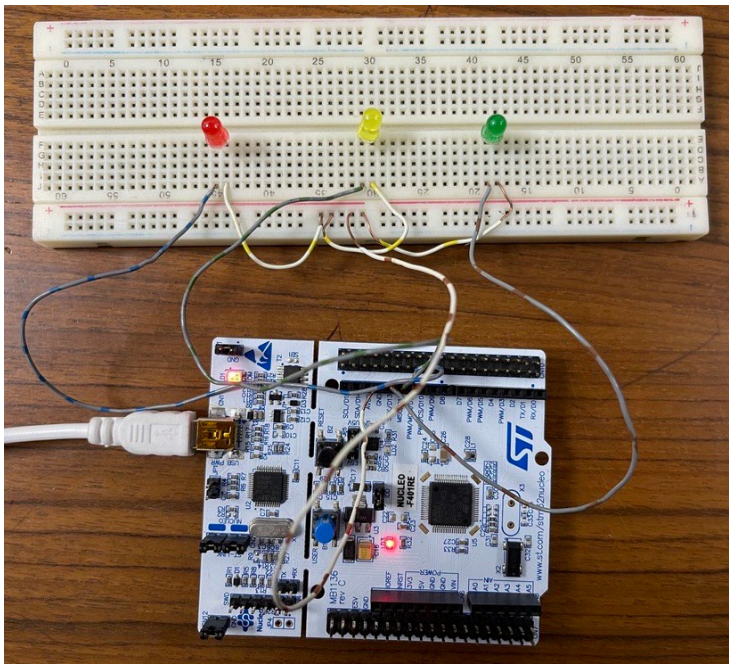

Fig 04. Hardware set-up for traffic light

Simulation Set-up:

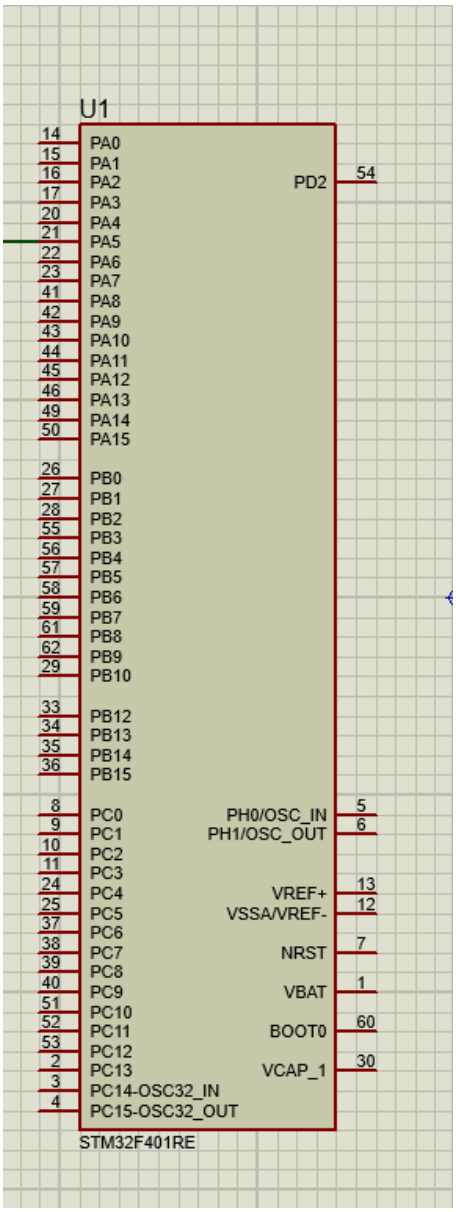

Fig 05: STM32 Simulation (Proteus)

Code & Explanation

```
int main(void) {
    HAL_Init();
    SystemClock_Config();

    MX_GPIO_Init();
    while (1) {
        /* USER CODE END WHILE */
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5); // Toggling the pin 5 output (ON)
        HAL_Delay(6000); // Delay of 6 seconds
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5); // Toggling the pin 5 output (OFF)

        // Loop for pin 6 or Yellow LED blinking
        int i = 0;
        while (i < 4) {
            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6); // Toggling the pin 6 output (ON)
            HAL_Delay(400); // Delay of 0.4 second
            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6); // Toggling the pin 6 output (OFF)
            HAL_Delay(400); // Delay of 0.4 second
            i++;
        }
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7); // Toggling the pin 7 output (ON)
        HAL_Delay(3000); // Delay of 3 second
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7); // Toggling the pin 7 output (OFF)

    }
}
```

Here, the output of the corresponding pin can be toggled using the procedure HAL_GPIO_TogglePin(). The HAL_Delay() function (which accepts time in milliseconds as input) determines the time for keeping the open or delaying the output on that pin. This function toggles the current open mode of that pin. In our instance, pin 5 was connected to the red LED. The yellow traffic control system LED, which is connected to pin 6 of the microcontroller, subsequently blinks four times using a loop. Finally, pin 7 is connected to the green LED, which in this instance will illuminate for three seconds. We can adapt it to any use case by adjusting the delay time.

Hardware Output Results

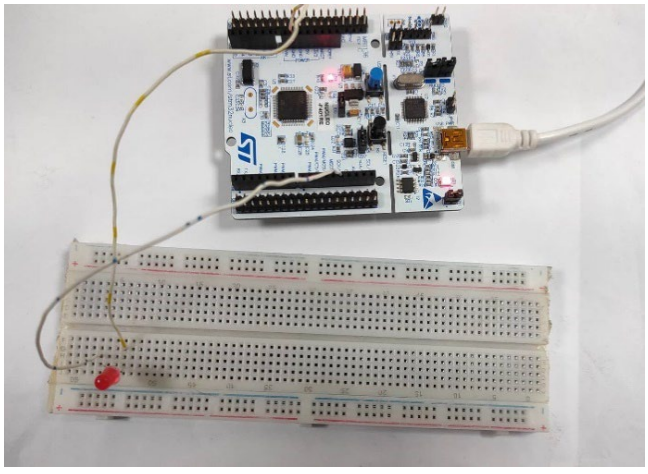

Fig 05: Hardware output with single LED light OFF

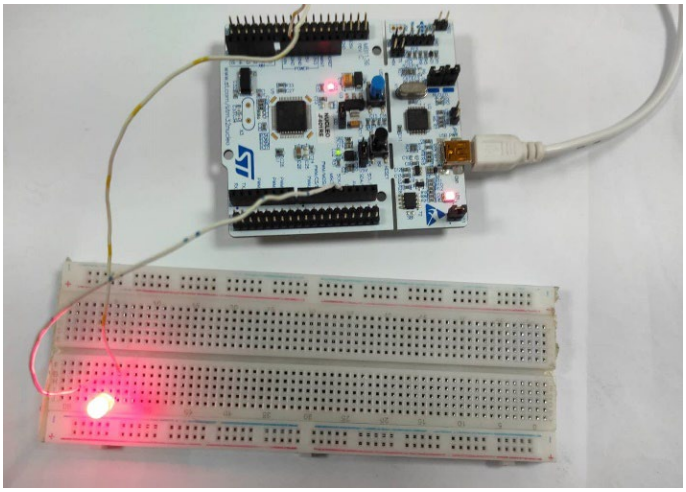

Fig 06: Hardware output with single LED light ON

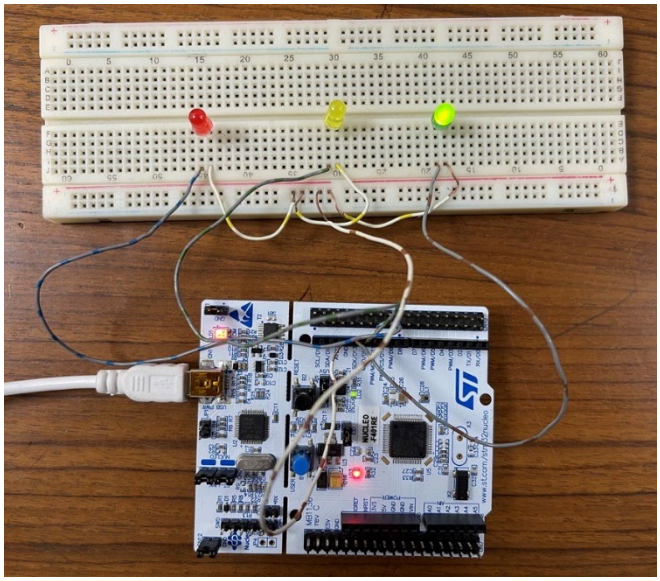

Fig 07: Traffic system; (Green light)

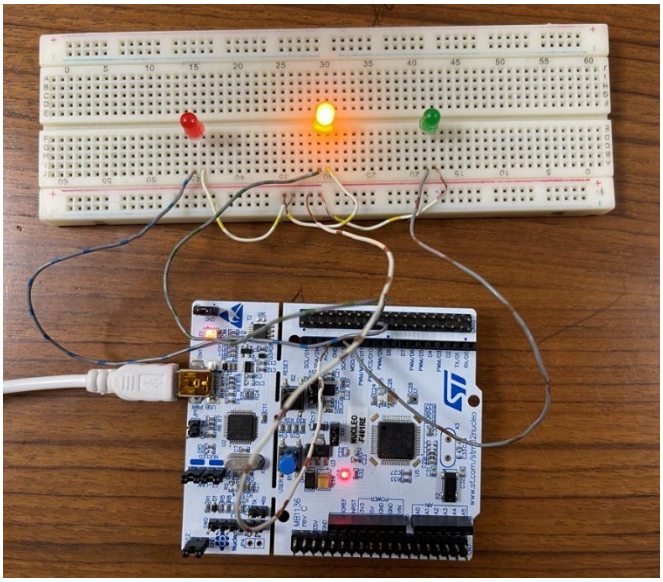

Fig 08: Traffic system; (Yellow light)

At figure 07, the green LED is turned on for 3000ms or 3 seconds and then it will turn off, which is connected to pin-07 of the microprocessor. After this green LED is turned off, the system will be in the first state, which is the red LED turned on.

At figure 08. the yellow LED is blinking four times with a 400ms delay, which is connected to pin-06 of the microcontroller, and the pin is configured in the code part to blink four times with a 0.4s delay. as a traffic warning signal.

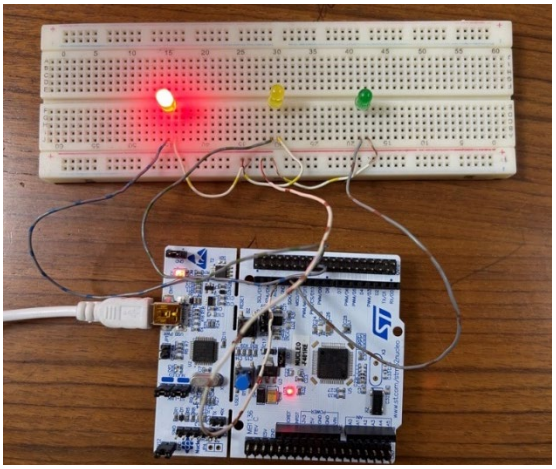

Fig 09: Traffic system; (Red light)

At figure 09 the red LED is connected to pin-05 of the STM32F401RET microcontroller. In the code part, the 5th pin is setup for the red LED. which will be turned on for 6000ms, or 6 seconds, to show the red sign of the traffic light.

Simulation Output Result

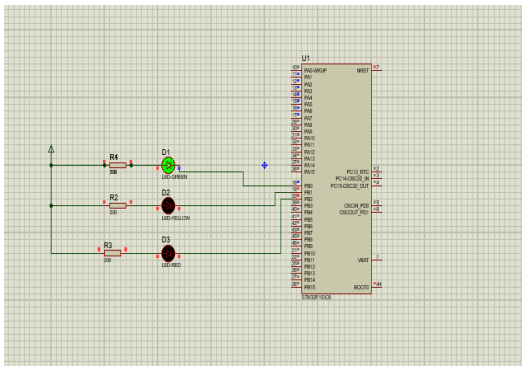

Fig 10: Simulation output Green LED light ON

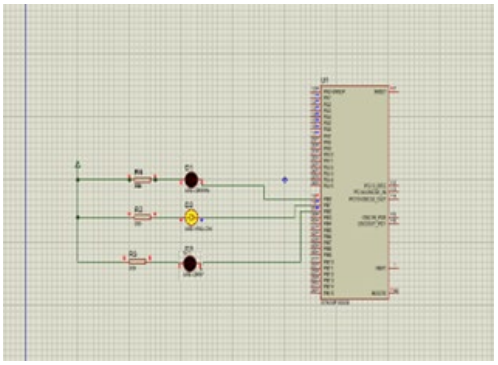

Fig 11: Simulation output Yellow LED light ON

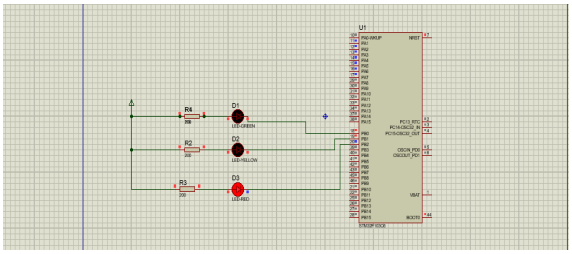

Fig 12: Simulation output Red LED light ON

Discussion & Conclusion

In this experiment, an STM32 is introduced, blink tests are studied, and a microcontroller-based light-control system is implemented. STM32 is also used to construct a traffic control system. A circuit has been created after that. The red, yellow, and green LED is connected to a breadboard with the help of two resistors. They have placed the resistors on the ground. The code was put into use once the circuit was designed. The STM32CUBE IDE was used to write the codes, which included delay and LED blink functions. The code was successfully executed after being written and simulated, and the outcome was reported. After then, a similar traffic control system was put into place. The experiment is carried out in the lab using the STM32CUBE IDE to write the code and the STM32F401RET microcontroller to execute it.

References

- AIUB Lab Manual
- https://www.st.com/en/evaluation-tools/nucleo-f401re.html
- https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf