American International University- Bangladesh

Department of Computer Science

Lab Report Cover Sheet

Course Name	MICROPROCESSOR AND EMBEDDED SYSTEMS
Lab Report No.	02
Lecturer Name	**MD. ALI NOOR**
Semester	SUMMER 2021-22
Submission Date	23/06/2022
Section	O
Group No.	03

Student Name	**Student ID**	**Contribution (out of 100%)**
1. EMON, MEDEDI HASAN	19-41118-2	
2. ASIF AHAMMED	19-40020-1	
3. EASINUL ABEDIN SAYEM	19-40291-1	
4. MD REDOY SHEIKH	19-40101-1	
5. MD. NAZIM HASAN	19-40118-1	
6. MOHI UDDIN ANANDO	20-43585-1	

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

Abstract:

A microcontroller is a small integrated circuit that controls a single operation in an embedded system. On a single chip, a typical microcontroller contains a CPU, memory, and input/output (I/O) peripherals. Microcontrollers are utilized in products and equipment that are automatically operated, such as automotive engine control systems, implantable medical devices, remote controls, office machinery, appliances, power tools, toys, and other embedded systems. To complete the work with a microcontroller, C/C++ and assembly language are unquestionably required. For low power consumption, some microcontrollers may use four-bit words and run at frequencies as low as 4 kHz. A microcontroller is a compact microcomputer designed to control the functions of embedded systems in office equipment, robotics, home appliances, motor vehicles, and a variety of other devices. A microcontroller has components like - memory, peripherals and most importantly a processor.

Although a microcontroller is capable of doing a wide range of tasks, the following issues have arisen while dealing with and implementing the lab:

- To learn how to use Arduino and the delay functions to make an LED blink.
- To put it all together, I used Arduino and Proteus.

Objective:

The objectives of the experiment are given below:

- To get familiarized with Microcontroller.
- To make the LED blink using Arduino and the delay functions.
- To get introduce with the Implementation of a traffic control system using Arduino.
- To learn more about Arduino family.

Results:

1.1 Simulation (Using 'Proteous' software)

1. Here is the circuit we made accordingly as mentioned in the question:

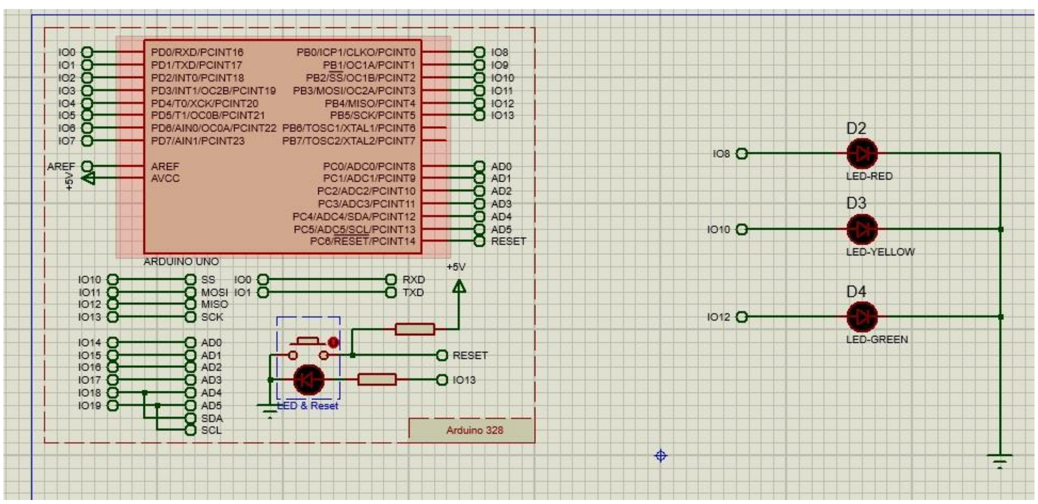

Figure 1: The basic circuit using Arduino UNO

2. Normal operation of a traffic control system: Basically in a traffic control system, some basic methodologies are used to create an outcome of the proper time delaying and also creating a signal flows. Using the microcontroller, we mainly manufacture the whole thing as a single machine which is used for the purpose of a traffic control system. We create a frequency like clock pulses by utilizing the codes and timing into the microchips. Also, this model includes 3 coloured (red, yellow, green) LED's, that works as the traffic lights on the road. The LED's get switched on and off by making the corresponding port pins enabled, using the code, we make sure of that. Thus, for a particular time the lights (red, yellow, green) turns on and off and spontaneously simulate as a working traffic signal.

3. Here is the code accordingly and the output operation:

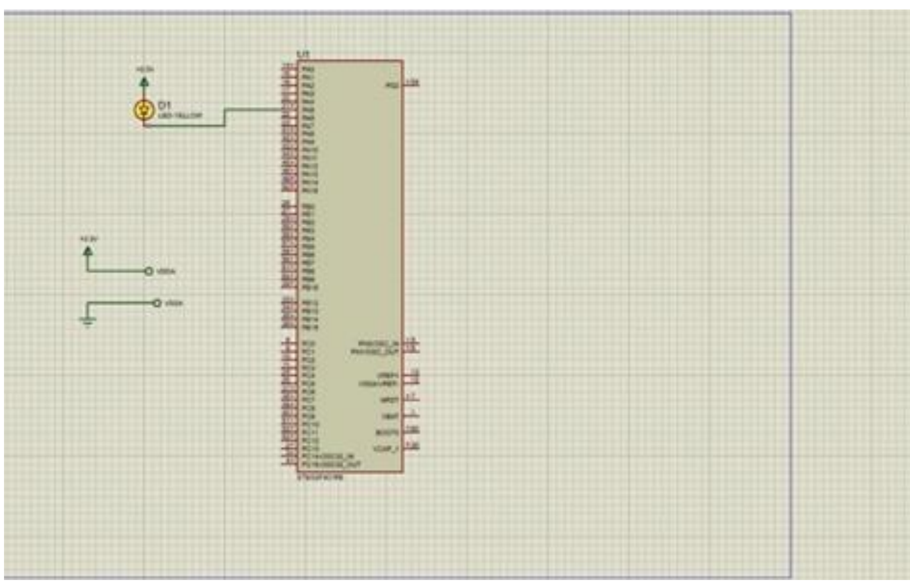

Figure 2: Yellow LED is ON.

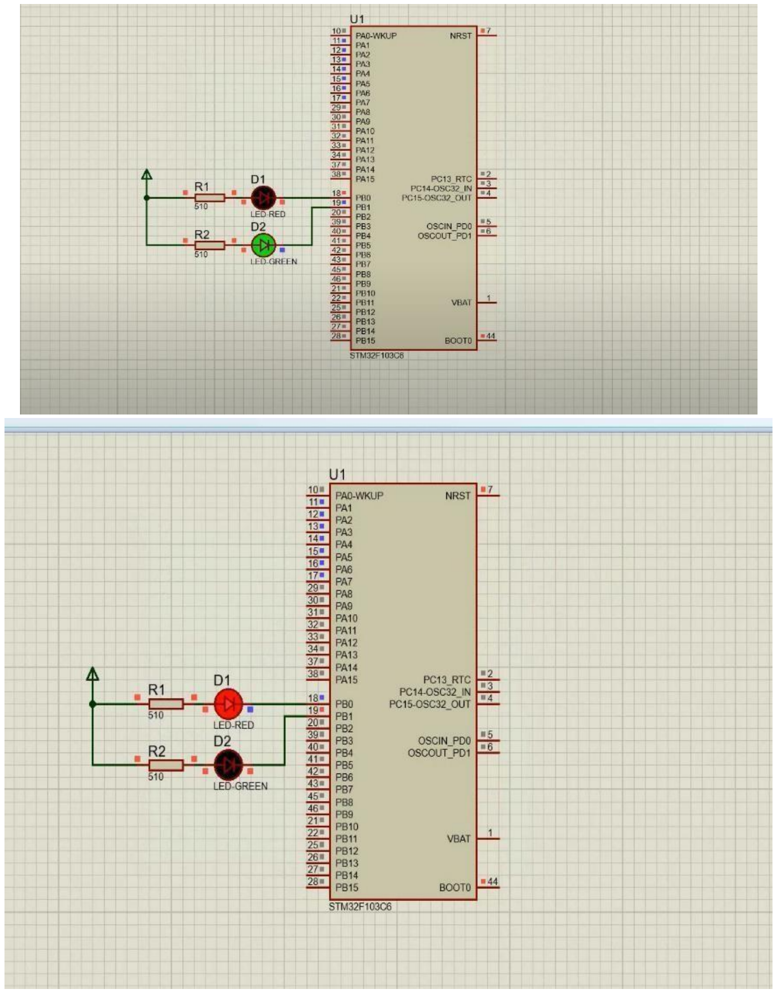

Figure 3: For Red &Green LED

Simulation code for traffic control system:

4. After rewriting the code for operation and same circuit, the simulation had some changes because there were certain delays.

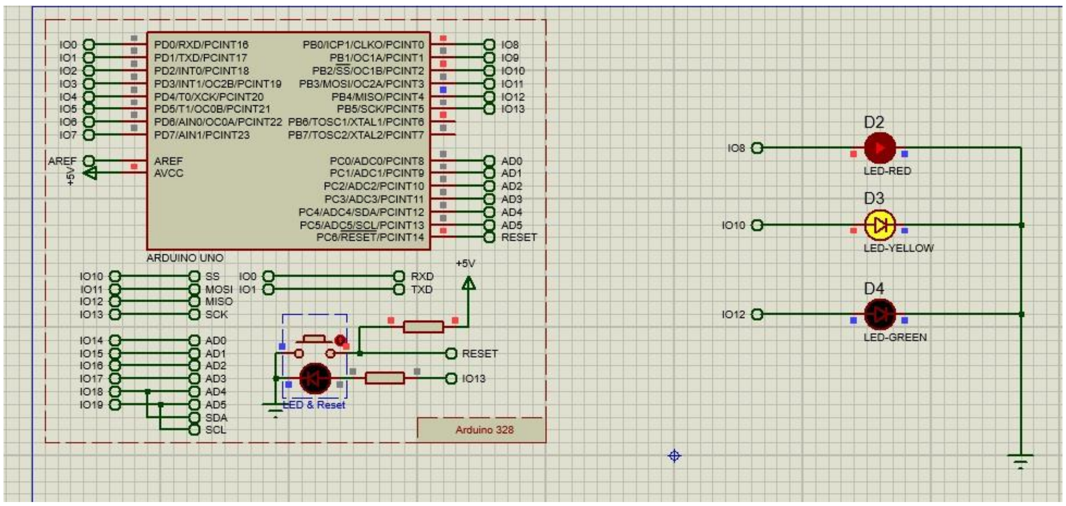

Figure 3: Delayed situation of turning ON.

Source Code: In the STM32 Cube IDE, Using C programming language to configure and debug features for the STM32 microcontroller, shown below.

```c
#include "main.h"
 void SystemClock_Config(void);
static void MX_GPIO_Init(void);


int main(void)
{


  HAL_Init();
  SystemClock_Config();


  MX_GPIO_Init();


  while (1)

            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);
            HAL_Delay(3000);
            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
            HAL_Delay(1000);
            HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_7);
            HAL_Delay(3000);
  {






  }
}


/**
* @brief System Clock Configuration
* @retval None
  */
void SystemClock_Config(void)
{
  RCC_OscInitTypeDef RCC_OscInitStruct = {0};
  RCC_ClkInitTypeDef RCC_ClkInitStruct = {0};


  /** Configure the main internal regulator output voltage    */
  __HAL_RCC_PWR_CLK_ENABLE();
  __HAL_PWR_VOLTAGESCALING_CONFIG(PWR_REGULATOR_VOLTAGE_SCALE2);
/** Initializes the CPU, AHB and APB busses clocks
  */
  RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
  RCC_OscInitStruct.HSIState = RCC_HSI_ON;
  RCC_OscInitStruct.HSICalibrationValue =
RCC_HSICALIBRATION_DEFAULT;
  RCC_OscInitStruct.PLL.PLLState = RCC_PLL_NONE;    if
(HAL_RCC_OscConfig(&RCC_OscInitStruct) != HAL_OK)
  {
    Error_Handler();
  }
  /** Initializes the CPU, AHB and APB busses clocks
  */
  RCC_ClkInitStruct.ClockType =
RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK


|RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
  RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_HSI;
  RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
  RCC_ClkInitStruct.APB1CLKDivider         =         RCC_HCLK_DIV1;
 RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
```

```c
static void MX_GPIO_Init(void)
{
  GPIO_InitTypeDef GPIO_InitStruct = {0};
  __HAL_RCC_GPIOH_CLK_ENABLE();
  __HAL_RCC_GPIOA_CLK_ENABLE();
  HAL_GPIO_WritePin(GPIOA,
GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7, GPIO_PIN_RESET);
  GPIO_InitStruct.Pin =
GPIO_PIN_5|GPIO_PIN_6|GPIO_PIN_7;
  GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
  GPIO_InitStruct.Pull = GPIO_NOPULL;
  GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_LOW;
  HAL_GPIO_Init(GPIOA, &GPIO_InitStruct);
 }
void Error_Handler(void)
{


}


#ifdef USE_FULL_ASSERT
Void assert_failed(uint8_t *file, uint32_t line)
{
   }
#endif
```

<pre> if (HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0)
!= HAL_OK)
  {
    Error_Handler();
  }
}</pre>	

Hardware Set-Up:

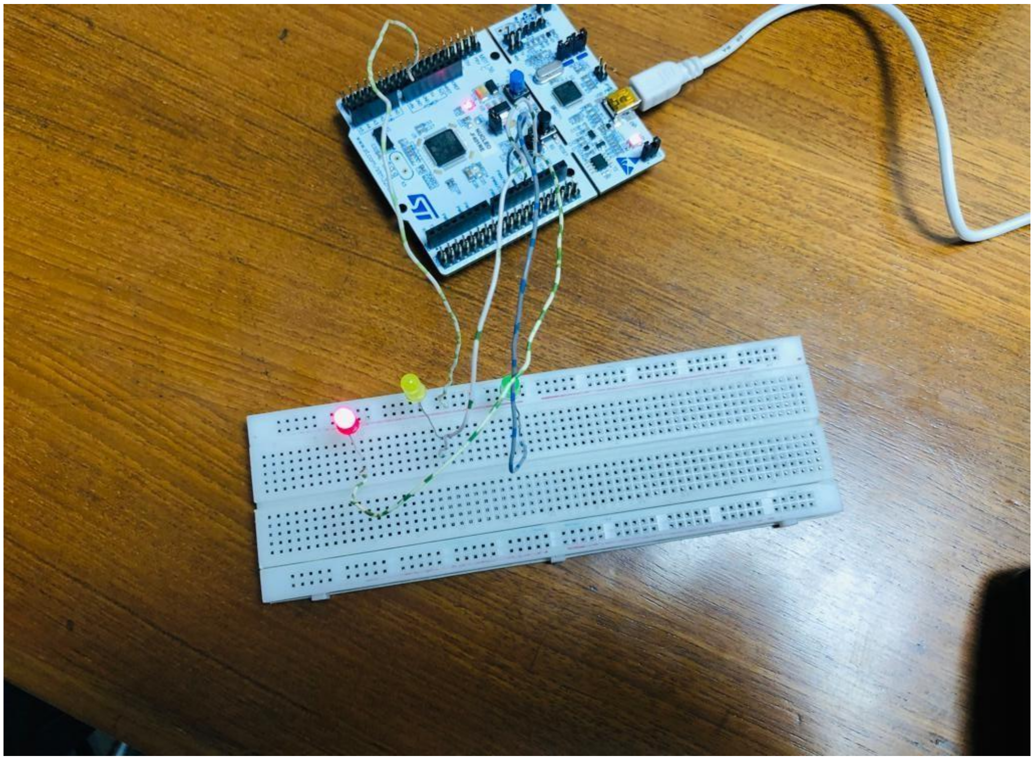

Fig: Red light turned on

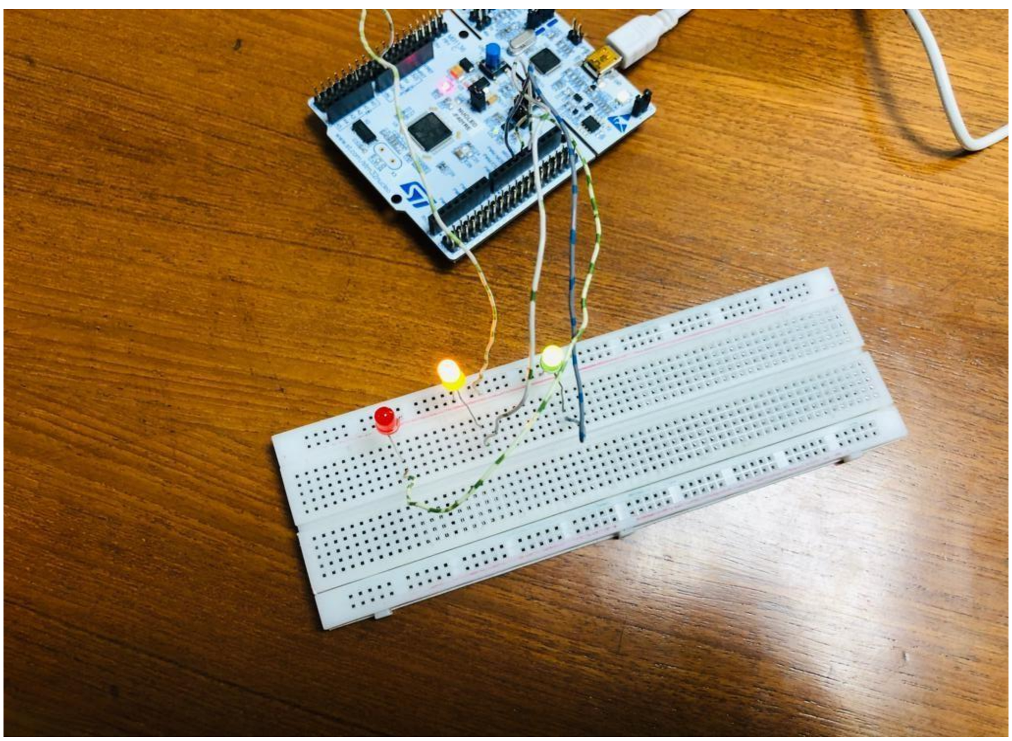

Fig: Yellow light turned on

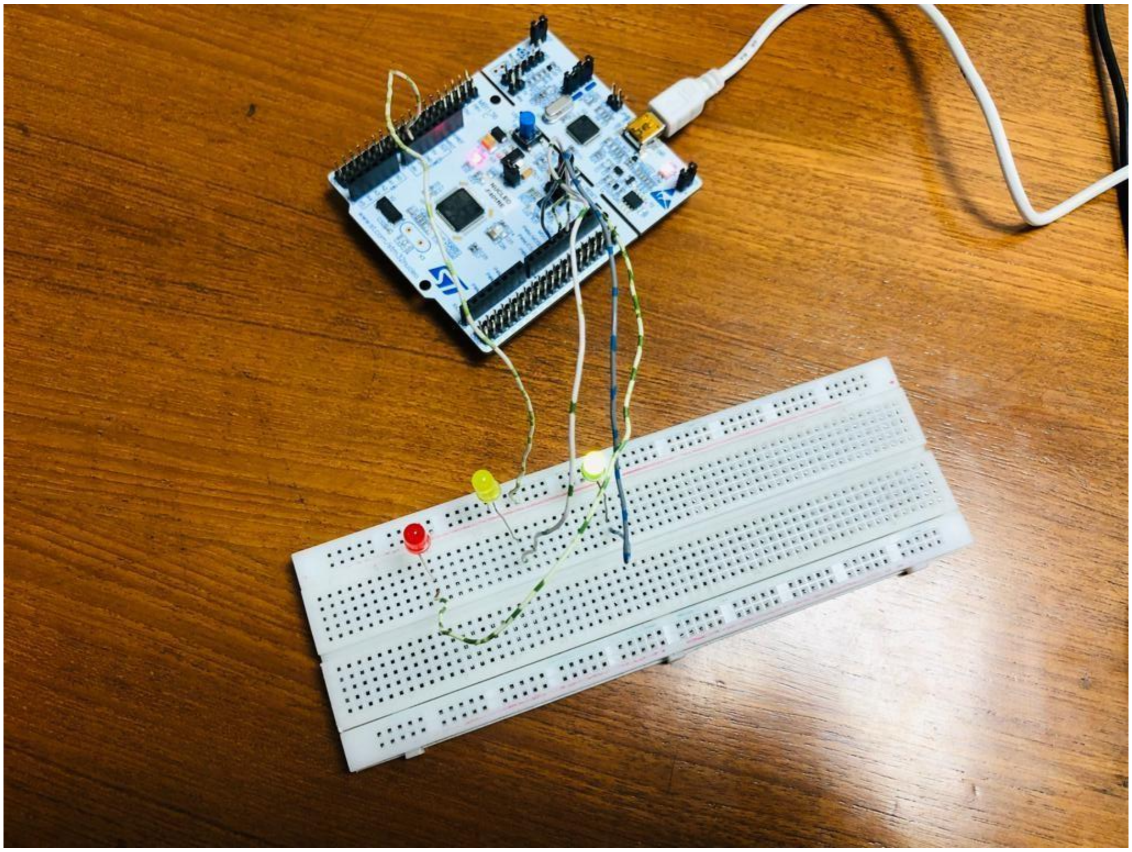

Fig: Green light turned on

Question Answer:

1) Include all codes and scripts into lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 1.

Answer: All code and scripts are given in Result and Discussion section.

Discussion:

In the following experiment, the primary target was to acquire knowledge on a microcontroller STM32 and assemble a demo light control system. The experiment with three LED lights (RED, Green, and Yellow) and STM32 board, and some wires to prepare a light system. After connecting the LEDs and wires to the board as shown in the image, it was time to connect them with the STM32 microcontroller. STM32 had 28 pins. Wires are gained in PA5, PA6, and PA7 at the end.

Once the circuit construction has been done, the USB should be inserted into the computer to develop program commands on the STM32 board. The required code was written in "while loop" which was 4 to 5 lines as shown in the "Source code" section. After building and running the program from STM32 cube IDE, the board showed results to blink LED lights.

Conclusion:

The experiment's main goal was to familiarize participants with microcontrollers. We learned about microcontrollers while working on this experiment and built a traffic control system using an Arduino Uno. Our course instructor walked us through the entire procedure so that we could successfully run this code. Working on Arduino was difficult at first, but it progressively became easier to manage. However, with the assistance of our teacher, the code was eventually run successfully. As a result, we concluded our report.

References:

1) Lab manual

2) STM32 Cube IDE https://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-armhttps://www.st.com/en/microcontrollers-microprocessors/stm32-32-bit-arm-cortex-mcus.htmlcortex-mcus.html

3) Proteus 8 Professional v8.11