United International University
QUEST FOR EXCELLENCE

Department of Electrical and Electronic Engineering

Lab Report - 01

Submitted To

Mr. Raiyan Basher
Lecturer
United International University

Submitted By- Md. Yeahyea Hassan Siddique

Course Code: EEE 3404
Course Title: Microprocessor and Interfacing Laboratory
Section: A

Experiment 1: Blinking LEDs (Simulation in Proteus and practical implementation in the STM32 Blue Pill)

Introduction:

• Briefly introduce the concept of blinking LEDs, highlighting their applications in embedded systems and electronics projects.
• Mention the chosen simulation software (Proteus) and hardware platform (STM32 Blue Pill) and their suitability for this project.
• Provide an overview of the report's structure and key objectives.

Components:

1. Gather Hardware:

- STM32 Blue Pill development board
- LED (standard or RGB, depending on your preference)
- Resistor (220-ohm or adjusted based on LED specs)
- Breadboard or jumper wires
- USB cable
- Power supply (3.3V or 5V, depending on your board and
- LED)

2. Connect the Components:

- Mount the Blue Pill on the breadboard or connect it using jumper wires.
- Connect the LED

OUTCOME:

- When the switch is KEPT open, i.e. the input pin is CONTINUOUSLY HIGH, the LED with Pin 6 will glow continuously and that with Pin
- 7 will toggle at every 100 mS.
- When the switch is KEPT closed, i.e., the input pin is CONTINUOUSLY LOW, the LED with Pin 6 will toggle at every 100 mS and that with Pin 7 will glow continuously.

Configuration and connections:

- During configuration, High Speed Clock under RCC within System core should be set at "Crystal/ceramic resonator"
- In clock configuration, set the frequency at 72MHz.
- Pin 6 and Pin 7 of Port A will be set as Output and Pin 9 of the same port as input pin.
- Internal pull up should be chosen for the input pin. [from GPIO setting in System core]
- Two sets of a combination of one LED with a 330 ohms' resistor should be connected both with Pin 6 and Pin 7. One switch is connected between Pin 9 and ground.

Working process of blinking LEDs using Proteus simulation and practical implementation on the STM32 Blue Pill:

Proteus Simulation:

1. Component Selection:

- Open Proteus and add the STM32F4 Discovery board or a similar development board matching your Blue Pill's specifications.
- Place two LEDs and appropriate resistors (usually 220-330 ohms) in series with each LED to limit current and prevent damage. Connect the anode (longer leg) of each LED to a GPIO pin, and the cathode (shorter leg) to ground.

2. Schematic Creation:

- Connect the necessary power supplies (VDD and GND) to the board.
- Create connections between the board's components (MCU, LEDs, resistors) using wires.

3. Code Development:

- Write C code (using an IDE like Kiel MDK-ARM or STM32CubeIDE) to configure the GPIO pins as outputs.

4. Simulation:

- Compile and download the code to the simulated STM32 board.
- Run the simulation. You should see the LEDs blinking on and off in the Proteus environment.

Practical Implementation on STM32 Blue Pill:

1. Hardware Setup:

- Connect the Blue Pill to your computer using a USB cable and a suitable programmer/debugger (ST-LINK V2, Black Magic Probe, etc.).
- Connect the LEDs and resistors as in the simulation, matching the GPIO pin connections in your code. Ensure correct power supply connections.

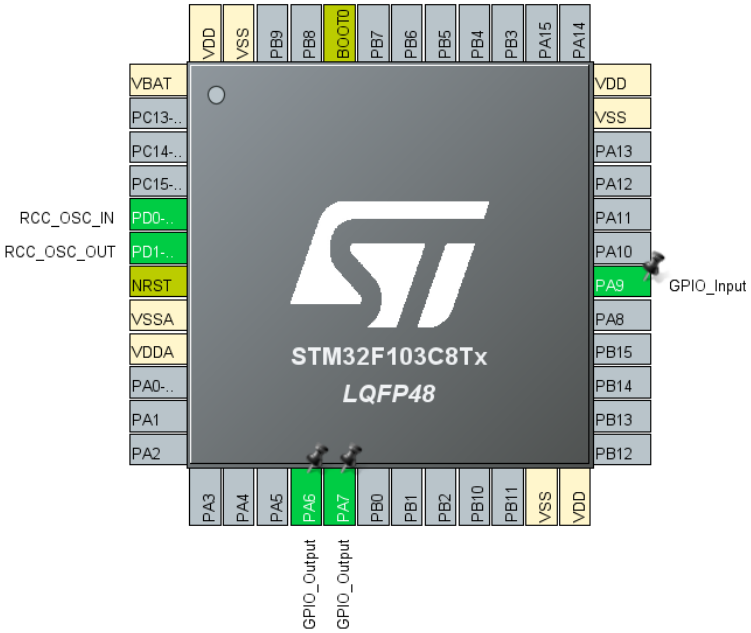
2. Code Upload:

- Use your chosen IDE to compile and upload the code to the Blue Pill.

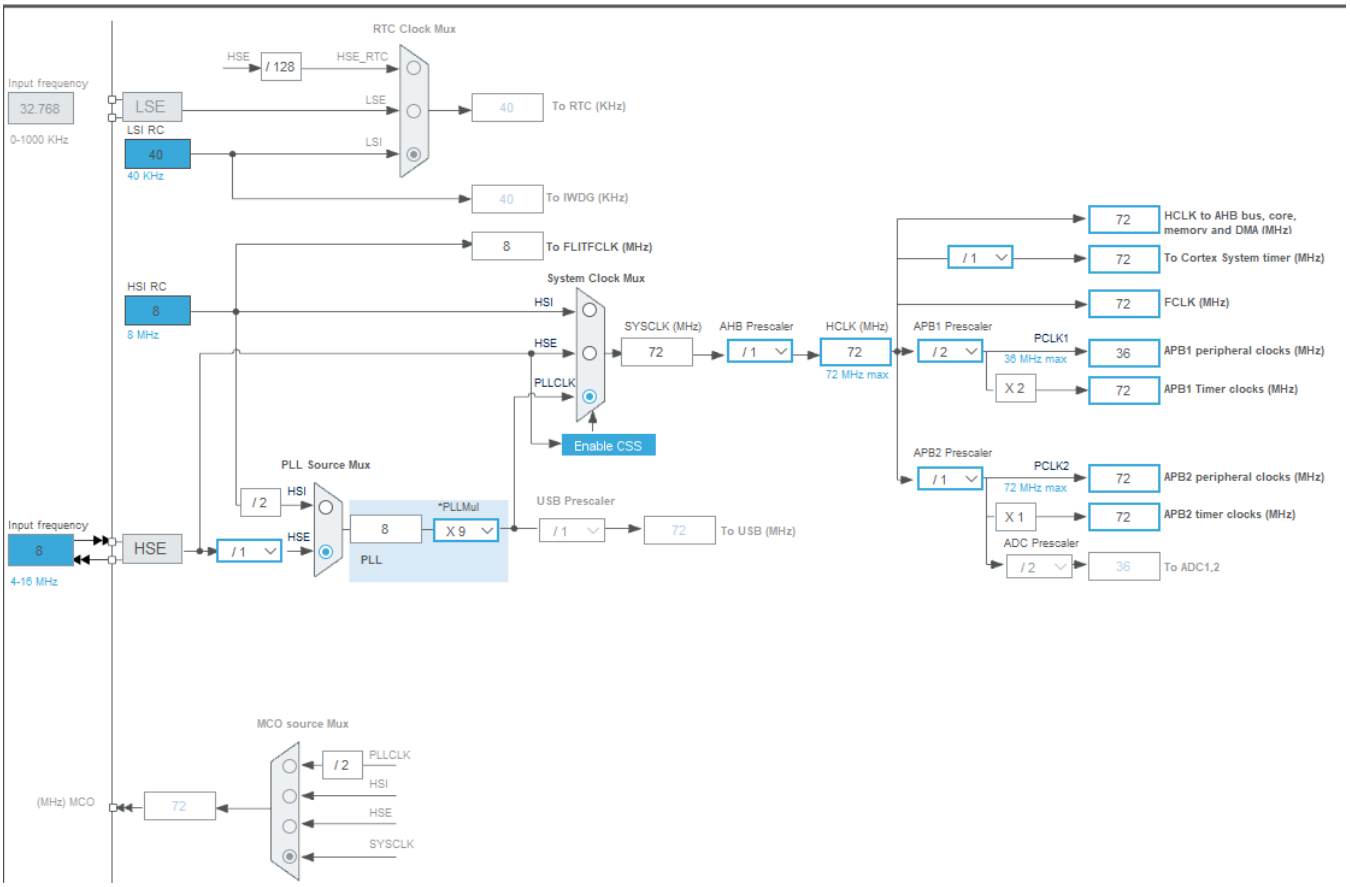
3. Observation:

- If everything is connected and programmed correctly, you should see the LEDs blinking on the physical Blue Pill board.

Pinout & Configuration:



Clock Configuration:



Code to be added:

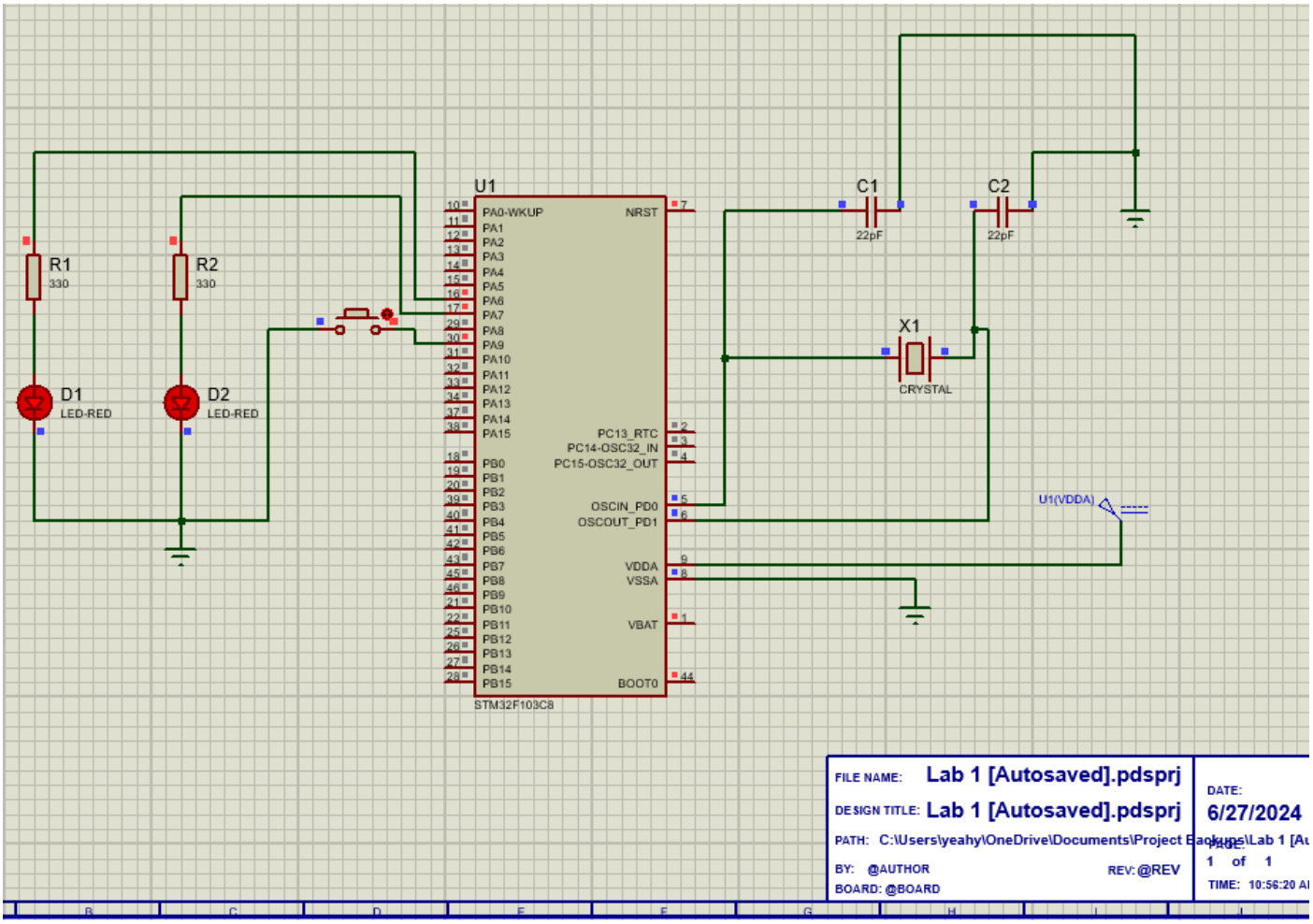
```
  */
/* USER CODE END Header */
/* Includes ------------------------------------------------------------------*/
#include "main.h"
#include <stdbool.h>

/* Private includes ----------------------------------------------------------*/
/* USER CODE BEGIN Includes */

/* USER CODE END Includes */

/* Private typedef -----------------------------------------------------------*/
/* USER CODE BEGIN PTD */
```

```
  /* Configure the system clock */
  SystemClock_Config();
  /* USER CODE BEGIN SysInit */
  /* USER CODE END SysInit */
  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  /* USER CODE BEGIN 2 */
bool b;
HAL_GPIO_WritePin(GPIOA,GPIO_PIN_6,0);
HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7,0);
  /* USER CODE END 2 */
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {b=HAL_GPIO_ReadPin(GPIOA,GPIO_PIN_9);
  if(b)
  {HAL_GPIO_WritePin(GPIOA,GPIO_PIN_6,1);
  HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_7);
  HAL_Delay(100);
  }
  else
  {HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_6);
  HAL_GPIO_WritePin(GPIOA,GPIO_PIN_7,1);
  HAL_Delay(100);
  }
    /* USER CODE END WHILE */
```

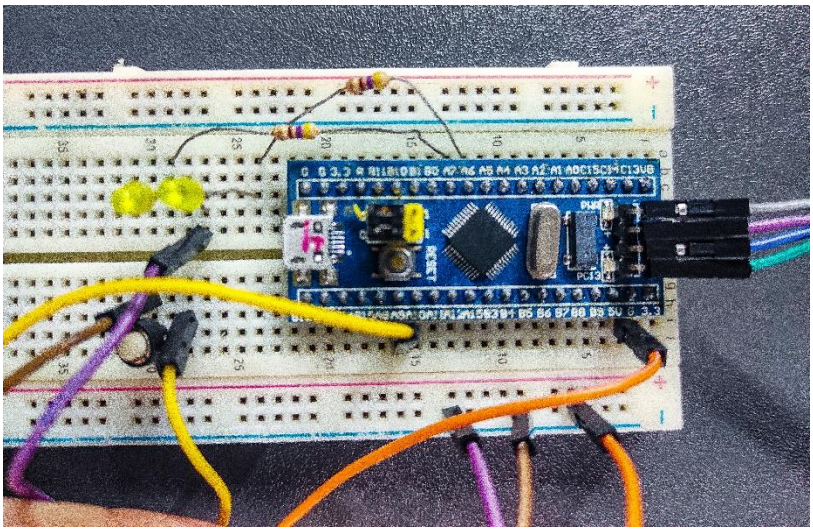
Output:

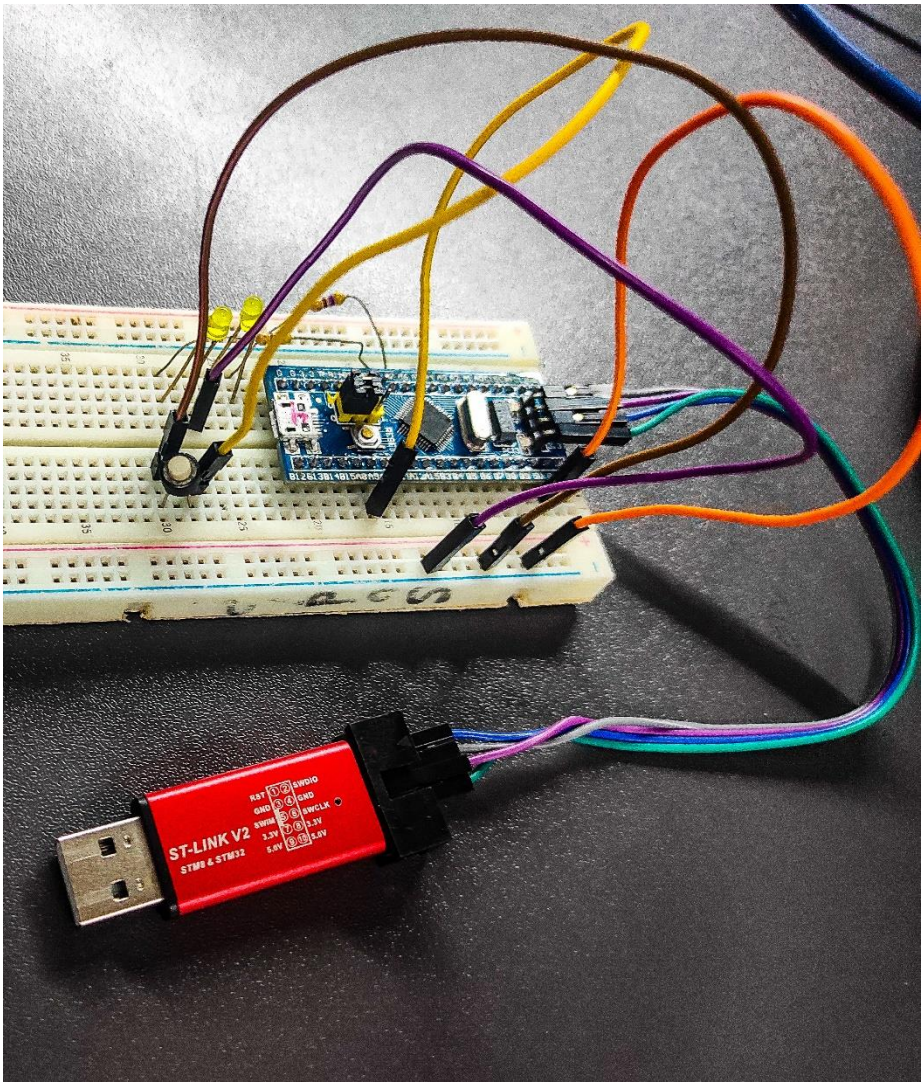
```
Problems  Tasks  Console  Properties
CDT Build Console [Lab 1]
make -j12 all
arm-none-eabi-size  Lab\ 1.elf
   text    data     bss     dec     hex filename
   4796      20    1572    6388    18f4 Lab 1.elf
Finished building: default.size.stdout


20:05:06 Build Finished. 0 errors, 0 warnings. (took 313ms)
```

PROTEUS SIMULATION:



Hardware Implementation:





Conclusion:

- Summarize the key findings and achievements of the project.
- Discuss any challenges encountered and solutions implemented.
- Suggest potential improvements or extensions for future work.
- Express your overall takeaways and learnings from the project.