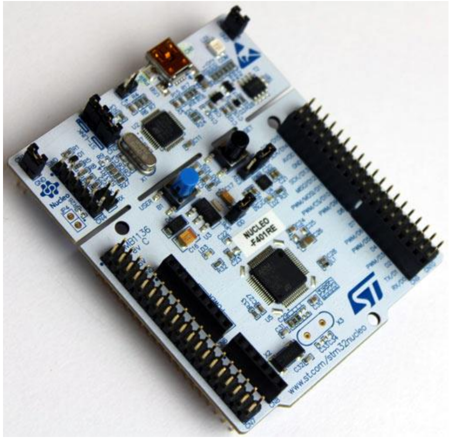
Title: Familiarization with an STM32, the study of blink test and implementation of a light-controlling system using microcontrollers.

Introduction:

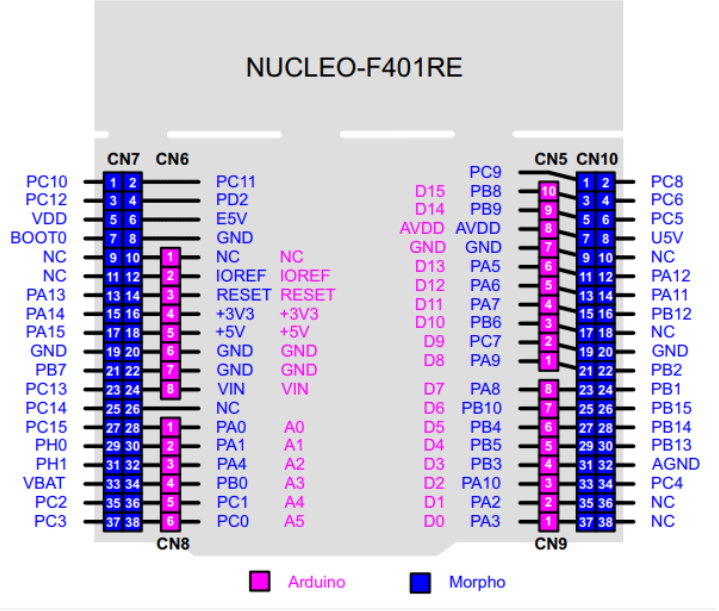
The objective of this experiment is to get familiarized with Microcontroller.
□ Learning to make the LED blink using ST32.
□ Implementation of a light control system using STM32.

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.

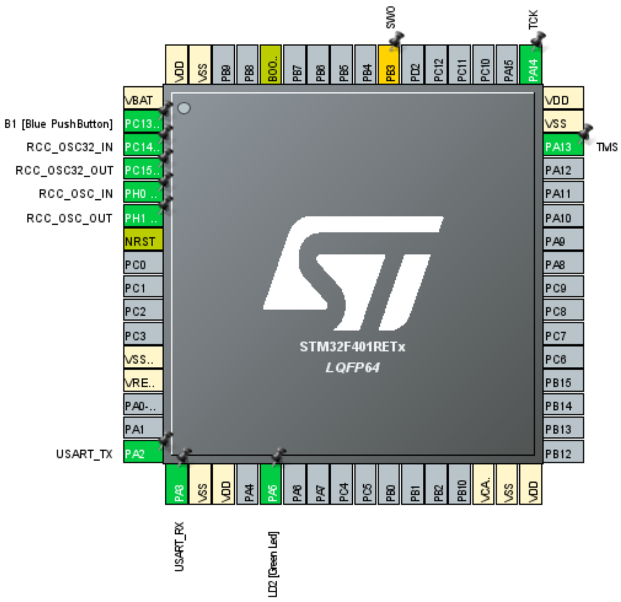
Overview of STM32 Nucleo-F401RE Board:



Pin Configuration:


NUCLEO-F401RE
CN7 CN6
CN5 CN10
CN8
CN9
PC10 PC12 VDD BOOT0 NC NC PA13 PA14 PA15 GND PB7 PC13 PC14 PC15 PH0 PH1 VBAT PC2 PC3
PC11 PD2 E5V GND NC IOREF RESET +3V3 +5V GND GND VIN NC PA0 PA1 PA4 PB0 PC1 PC0
NC IOREF RESET +3V3 +5V GND GND VIN A0 A1 A2 A3 A4 A5
D15 D14 AVDD GND D13 D12 D11 D10 D9 D8 D7 D6 D5 D4 D3 D2 D1 D0
PC9 PB8 PB9 AVDD GND PA5 PA6 PA7 PB6 PC7 PA9 PA8 PB10 PB4 PB5 PB3 PA10 PA2 PA3
PC8 PC6 PC5 U5V NC PA12 PA11 PB12 NC GND PB2 PB1 PB15 PB14 PB13 AGND PC4 NC NC
Arduino
Morpho

Pin configuration from STM32Cube IDE:


SWO
TCK
TMS
B1 [Blue PushButton]
RCC_OSC32_IN
RCC_OSC32_OUT
RCC_OSC_IN
RCC_OSC_OUT
USART_TX
USART_RX
LD2 [Green Led]
STM32F401RETx
LQFP64
VDD VSS PB9 PB8 BOO.. PB7 PB6 PB5 PB4 PB3 PD2 PC12 PC11 PC10 PA15 PA14
VBAT PC13.. PC14.. PC15.. PH0.. PH1.. NRST PC0 PC1 PC2 PC3 VSS.. VRE.. PA0-.. PA1 PA2
VDD VSS PA13 PA12 PA11 PA10 PA9 PA8 PC9 PC8 PC7 PC6 PB15 PB14 PB13 PB12
PA3 VSS VDD PA4 PA5 PA6 PA7 PC4 PC5 PB0 PB1 PB2 PB10 VCA.. VSS VDD

Apparatus:

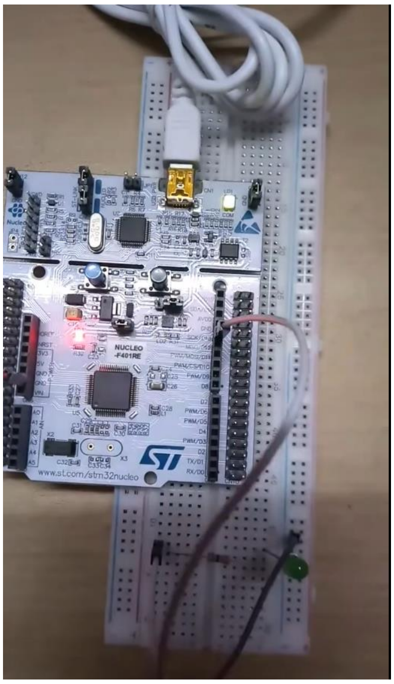
1) STM32 Cube IDE (1.0.1 or any recent version)

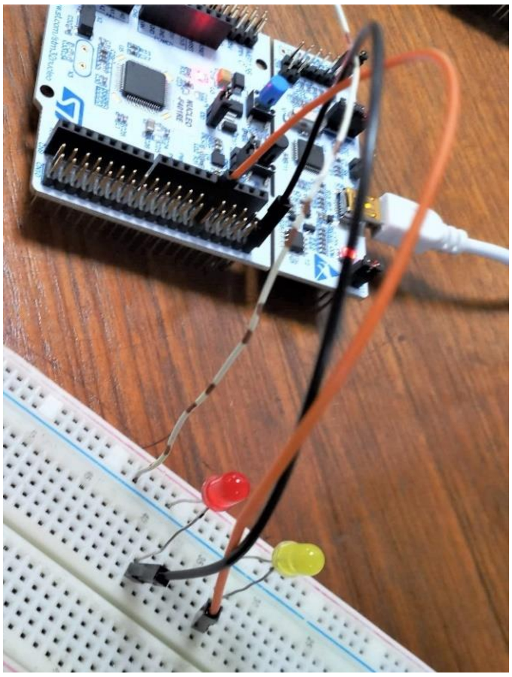
2) STM32 Cube IDE board

3) LED lights (RED, GREEN, or YELLOW) and three 200 ohms resistors and jumper wires

Experimental Procedure:

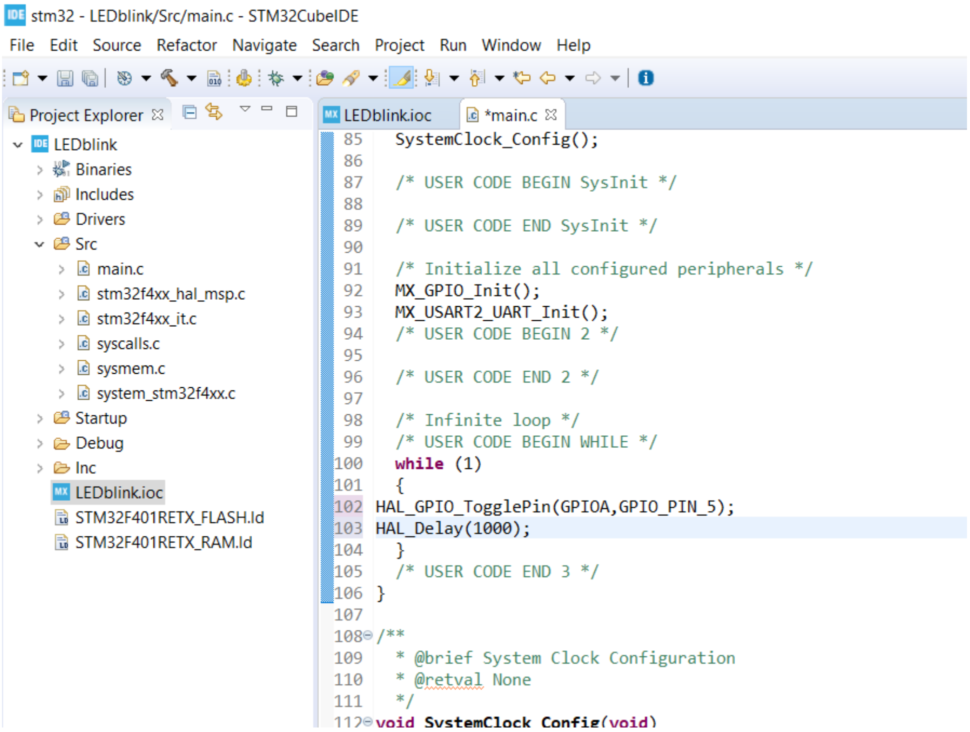
The main task of our lab is to understand and implement a light control system after understanding to blink a LED light. Make the circuits first using the following connection system between all the elements. Then plug in the STM32 Nucleo board to the PC.





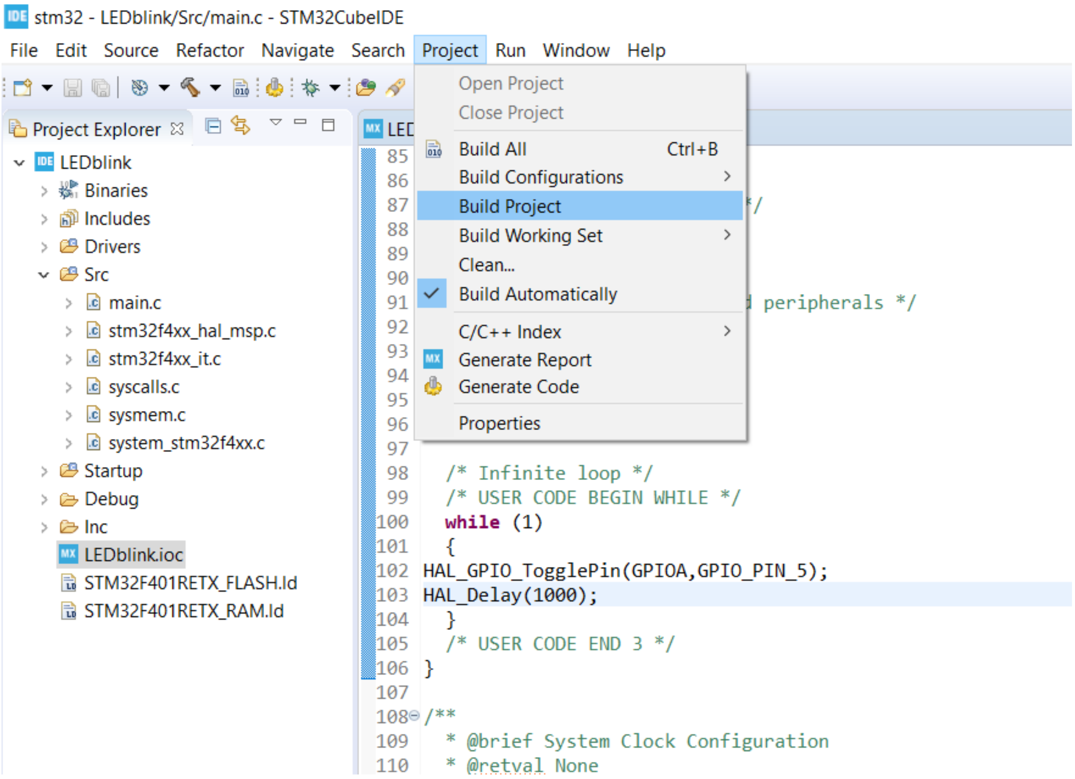
Using STM32Cube IDE to write code for a simple Blink program first:

1. Open the STM32Cube IDE and in the first while loop write the program as follows:



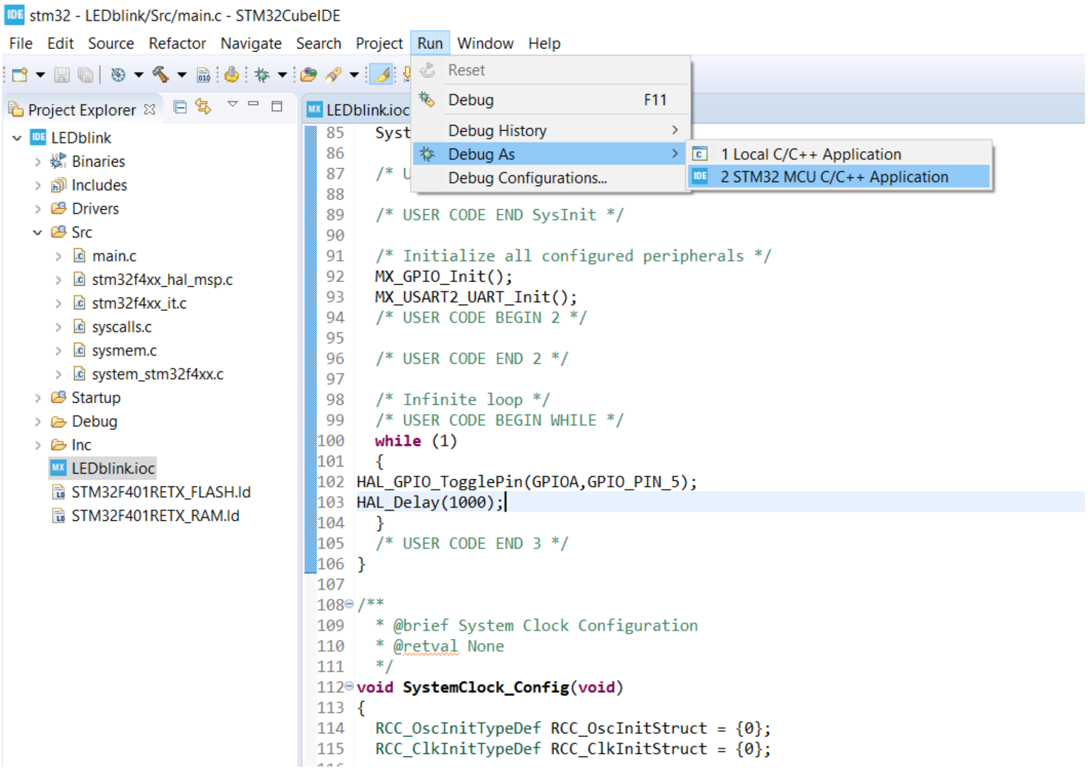
```
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART2_UART_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
  */
void SystemClock_Config(void)
```

2. Next save the project file then build the project.



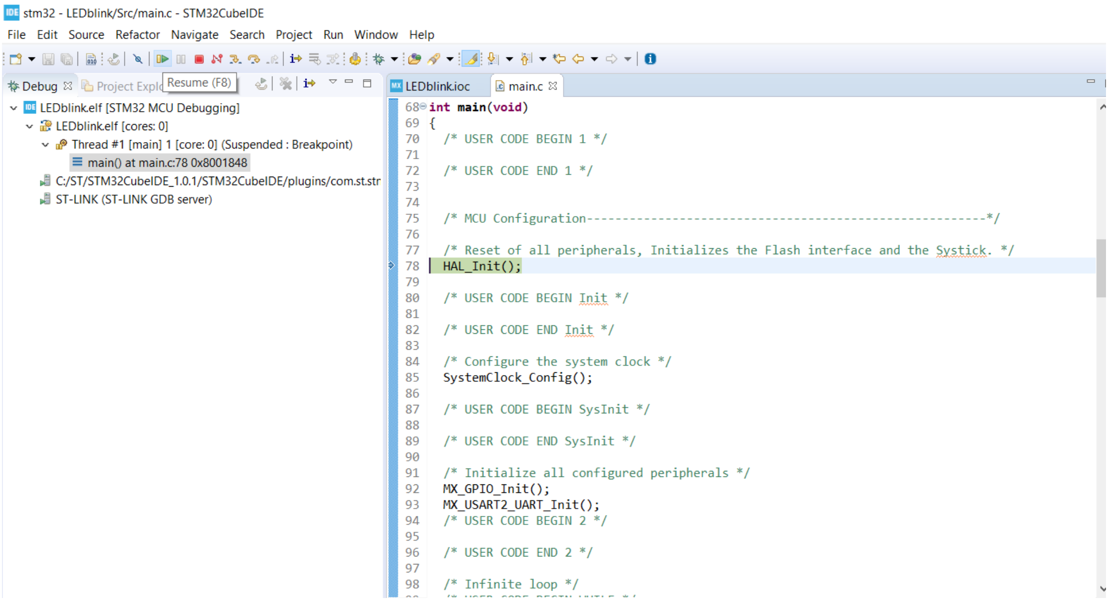
```
  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}

/**
  * @brief System Clock Configuration
  * @retval None
```

3. Next, run the program to debug as STM32 MCU C/C++ Application.



```
  Syst

  /* U

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART2_UART_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
  /* USER CODE BEGIN WHILE */
  while (1)
  {
HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
HAL_Delay(1000);
  }
  /* USER CODE END 3 */
}

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

4. Select the resume option to see the board showing results to blink the LED lights.


stm32 - LEDblink/Src/main.c - STM32CubeIDE
File Edit Source Refactor Navigate Search Project Run Window Help
Debug
Project Explo
Resume (F8)
LEDblink.ioc
main.c

LEDblink.elf [STM32 MCU Debugging]
LEDblink.elf [cores: 0]
Thread #1 [main] 1 [core: 0] (Suspended : Breakpoint)
main() at main.c:78 0x8001848
C:/ST/STM32CubeIDE_1.0.1/STM32CubeIDE/plugins/com.st.str
ST-LINK (ST-LINK GDB server)

```
int main(void)
{
  /* USER CODE BEGIN 1 */

  /* USER CODE END 1 */


  /* MCU Configuration--------------------------------------------------------*/

  /* Reset of all peripherals, Initializes the Flash interface and the Systick. */
  HAL_Init();

  /* USER CODE BEGIN Init */

  /* USER CODE END Init */

  /* Configure the system clock */
  SystemClock_Config();

  /* USER CODE BEGIN SysInit */

  /* USER CODE END SysInit */

  /* Initialize all configured peripherals */
  MX_GPIO_Init();
  MX_USART2_UART_Init();
  /* USER CODE BEGIN 2 */

  /* USER CODE END 2 */

  /* Infinite loop */
```

Try to rewrite the code again for the two LED lights

Questions for report writing:

1) Include all codes and scripts into the lab report following the writing template mentioned in appendix A of Laboratory Sheet Experiment 2.
2) Include the proteus simulation of the blink program and light control system. you can learn the simulation from the following link: https://www.youtube.com/watch?v=MDsoLQicdAk

3) Design a simulation for a traffic system using the Proteus simulation tool and STM32Cube IDE. HAL_GPIO_WritePin()

Example: HAL_GPIO_WritePin(GPIOB, GPIO_PIN_0, GPIO_PIN_RESET);

Reference(s):

- https://www.st.com/en/evaluation-tools/nucleo-f401re.html for STM32F401RE,datasheet
- www.st.com
- https://www.st.com/resource/en/user_manual/dm00105879-description-of-stm32f4-hal-and-ll-drivers-stmicroelectronics.pdf
- www.st.com/en/development-tools/stm32cubeide.html