

**STM32 CubeIDE, CubeMx for beginners with Examples**

Welcome to my STM32CubeIDE specifically designed for Beginners. In this, we will take a step-by-step approach to ensure that you gain a solid understanding of how to use **CudeIDE, CubeMx** for programming the **STM32** Microcontroller.

**Hardware requirements:**

1. STM32 blue pill board
2. ST-LINK/V2
3. Jumper wires
4. [SSD1306 Oled Display Module](https://amzn.to/3KaE0PL)
5. [HC-SR04 Ultrasonic Sensor](https://amzn.to/3Q8wsko)
6. [Push Button](https://amzn.to/43DRjz9)
7. Led, resistors

**Software requirements:**

1. Stm32 cubeide
2. Stm32 cubemx

Installation procedure: <https://www.youtube.com/watch?v=qe8ToTBv7jY>

I will be using the **STM32F103C6T6A** microcontroller board which is also known as Blue Pill, and for uploading the programming I will use the **ST-Link/V2**.

## **STM32F103C6T6A MICROCONTROLLER OVERVIEW**

## The STM32 Blue Pill development board is based on the STM32F103C6T6A microcontroller from the STM32F1 series by STMicroelectronics. It uses the ARM Cortex-M3 core and is widely used for embedded system learning and development due to its low cost and rich peripherals.

## **MICROCONTROLLER CORE**

## The STM32F103C6T6A is based on the ARM Cortex-M3 32-bit RISC processor. It provides a good balance between performance, power efficiency, and ease of programming, making it suitable for real-time embedded applications.

## **CLOCK SPEED**

## The microcontroller operates at a maximum clock frequency of 72 MHz, which allows fast execution of instructions and efficient peripheral handling.

## **MEMORY**

## The STM32F103C6T6A contains 32 KB of Flash memory used to store the program code and 10 KB of SRAM used for data storage during program execution.



## **OPERATING VOLTAGE**

## The operating voltage range of the STM32F103C6T6A is from 2.0 V to 3.6 V, with 3.3 V as the typical operating voltage. This makes it suitable for low-power applications.

## **GPIO PINS**

## The microcontroller provides approximately 32 GPIO pins. These pins are divided into Port A, Port B, and Port C and can be configured as input or output for interfacing with external devices such as LEDs, switches, sensors, and modules.

## **ANALOG PINS**

## The STM32F103C6T6A has 10 analog input pins connected to a built-in 12-bit Analog-to-Digital Converter (ADC). These pins are used to measure analog signals from sensors.

## **COMMUNICATION INTERFACES**

## The microcontroller supports multiple communication protocols. It has three USART interfaces for serial communication, two I2C interfaces for sensor and module communication, SPI for high-speed data transfer, and CAN for industrial communication.

## **PWM AND TIMERS**

## Multiple timers are available in the STM32F103C6T6A. These timers are used to generate PWM signals, perform time measurements, and control motors and other real-time operations.

## **POWER PINS ON BLUE PILL BOARD**

## The Blue Pill board provides two 3.3 V pins from the onboard voltage regulator. These pins can be used to power external sensors. The 5 V pin is used as an input power pin, and the onboard regulator converts it to 3.3 V. The board also has three common ground pins.

## **PROGRAMMING AND DEBUGGIN**

## The STM32F103C6T6A supports programming and debugging through SWD (Serial Wire Debug) pins. An ST-Link V2 programmer is commonly used for flashing and debugging the microcontroller.

## **VOLTAGE TOLERANCE**

## Most GPIO pins on the board are 5 V tolerant, allowing safe interfacing with 5 V devices. However, the PB5 pin is not 5 V tolerant and should only be used with 3.3 V signals.

## **DEVELOPMENT ENVIRONMENT**

## STM32CubeIDE is the official integrated development environment used for programming the STM32F103C6T6A. Programming is typically done using C or C++.

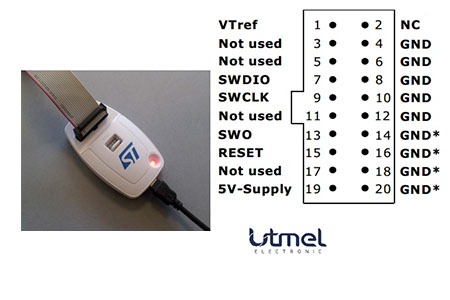
## **APPLICATIONS**

## The STM32F103C6T6A is widely used in industrial automation, robotics, IoT devices, motor control systems, and embedded system learning projects.



## **STM32 Blue Pill Board Pinout:**STM32 CubeIDE

## **ST-Link/V2 Pinout:**



**Connection:**

| **ST-Link/V2 Pin** | **Signal Name** | **STM32 Blue Pill Pin** |
| --- | --- | --- |
| Pin 7 | SWDIO | SWDIO |
| Pin 9 | SWCLK | SWCLK |
| Pin 20 | GND | GND |
| Pin 1 | VREF (3.3V) | 3.3V |



**Project-1: Blink on-board LED**

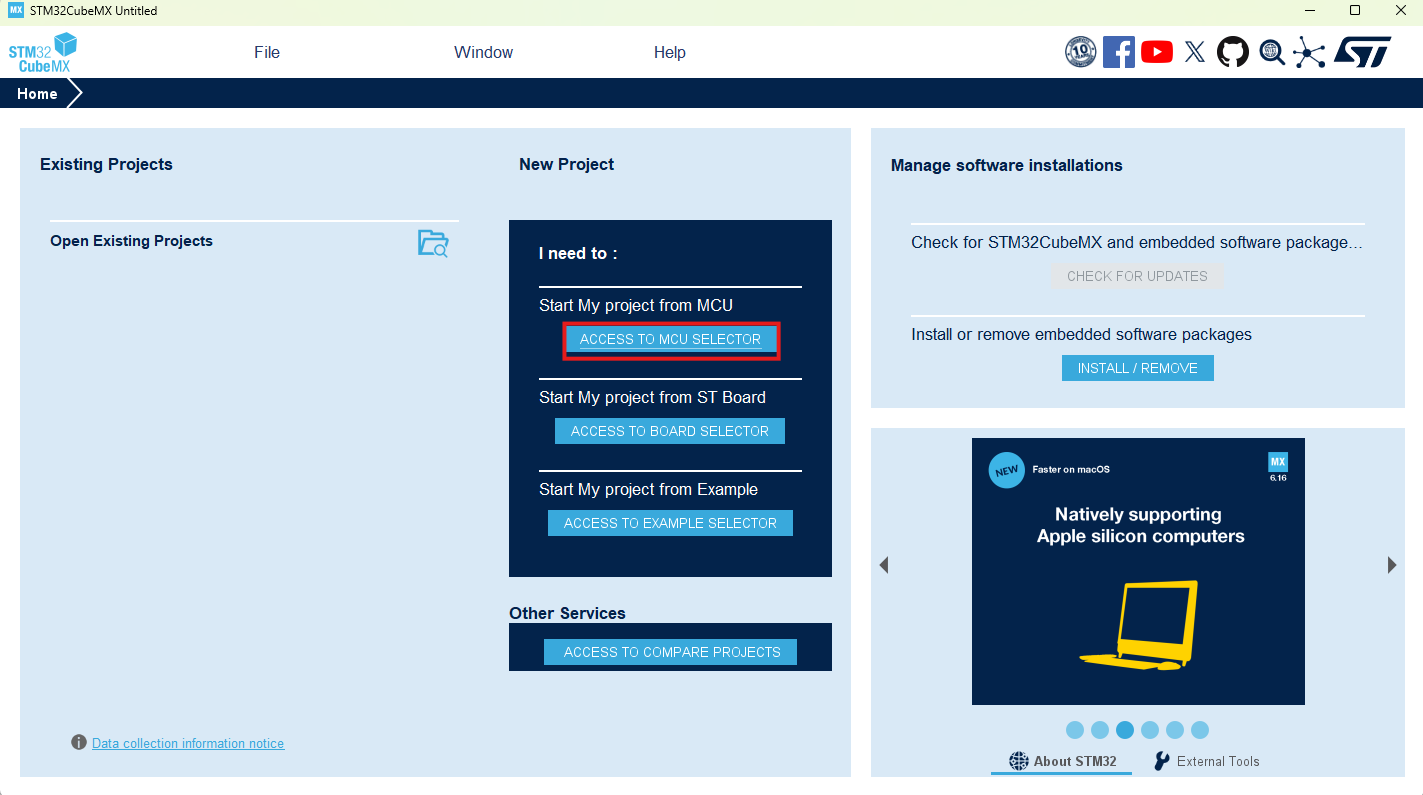
**Objective:** To understand basic GPIO configuration and control by blinking an on-board LED using the STM32 microcontroller.

**Learning Outcomes:** Configure GPIO pins as output using STM32CubeIDE. Generate delays and control LED ON/OFF through embedded C code

**Procedure/Setup:**

**Step-1:** Open stm32Cubemx

**Step-2:**

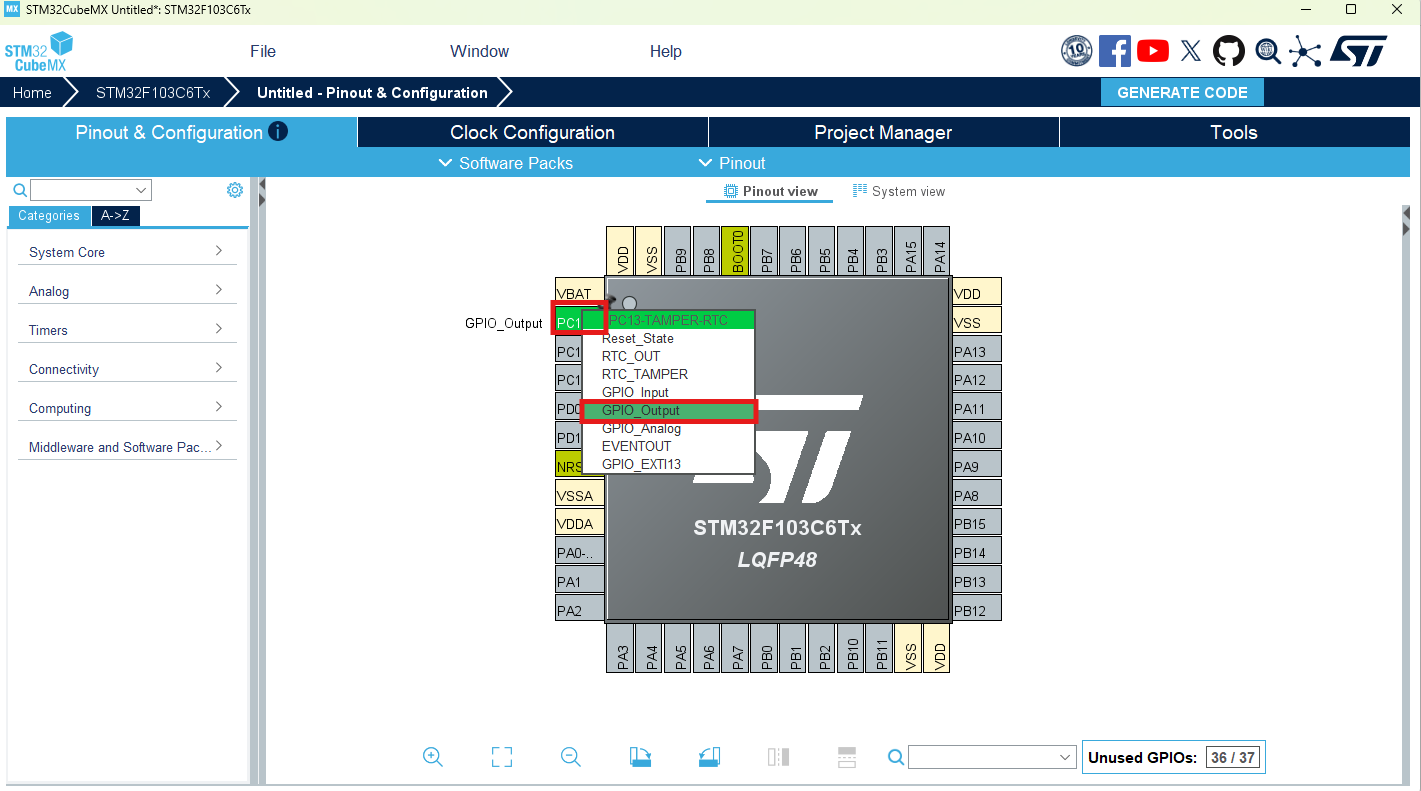


**Step-3:**

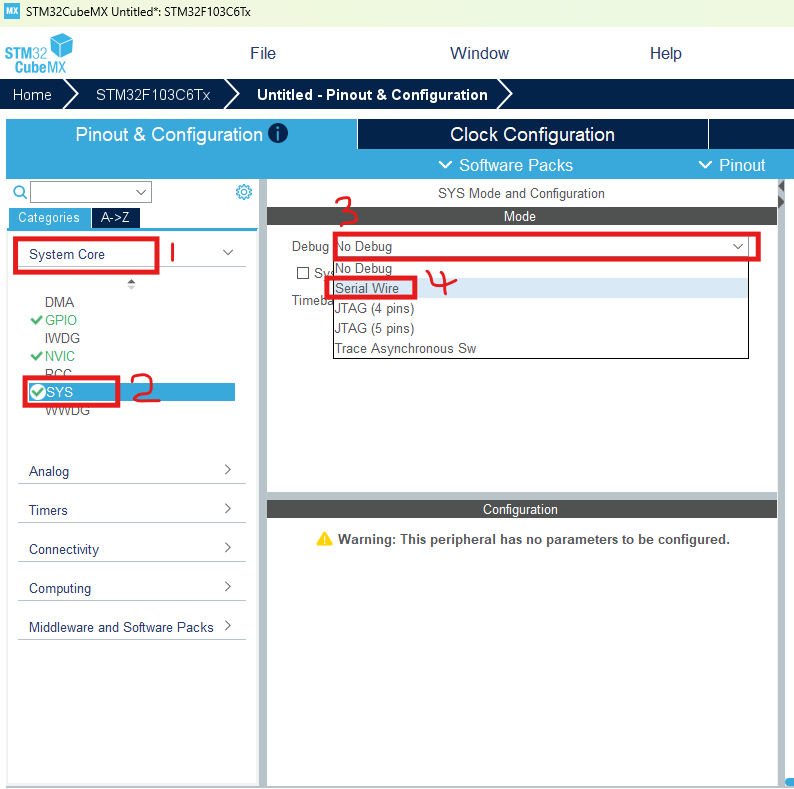




**Step-4:**

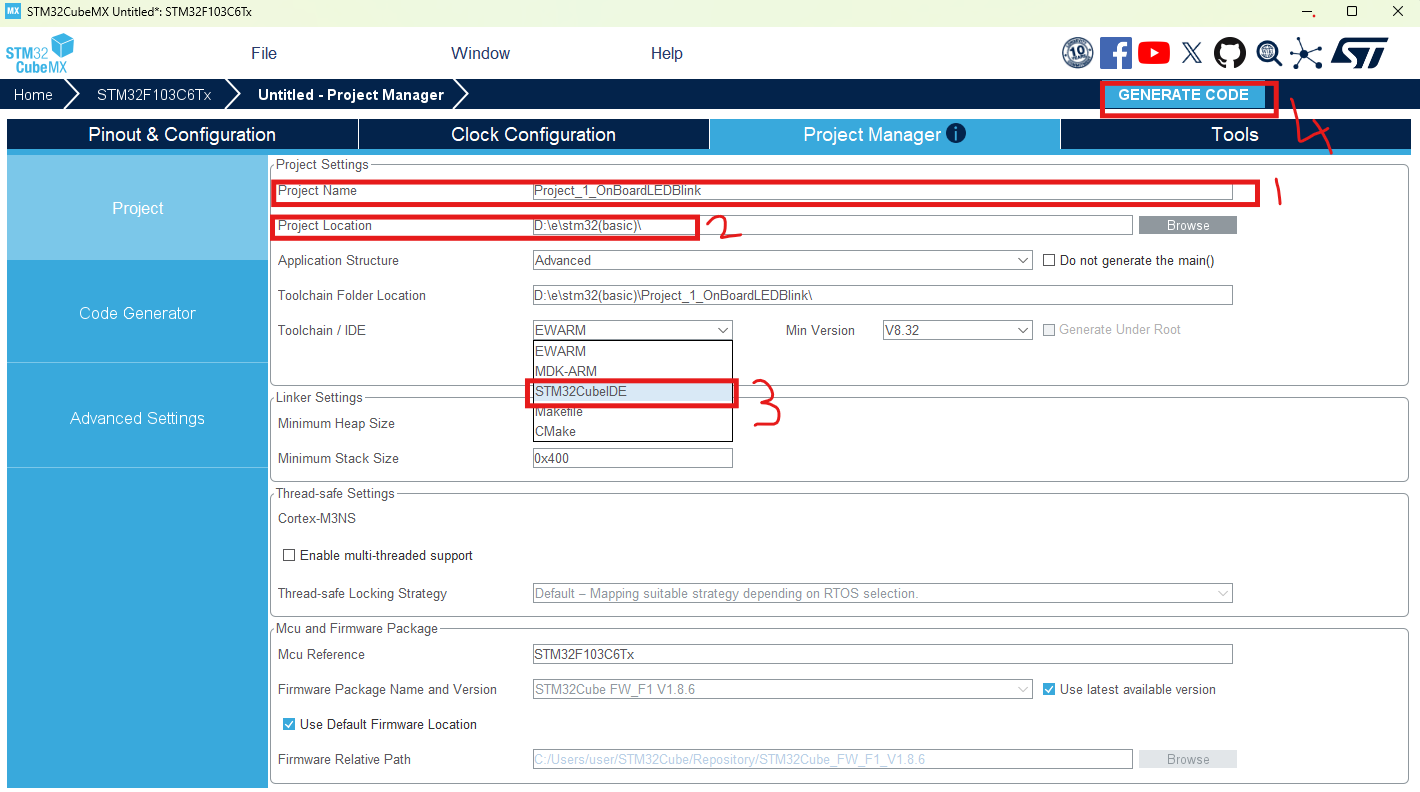


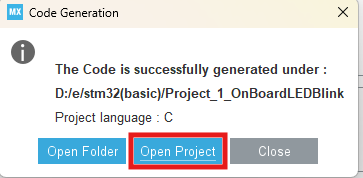
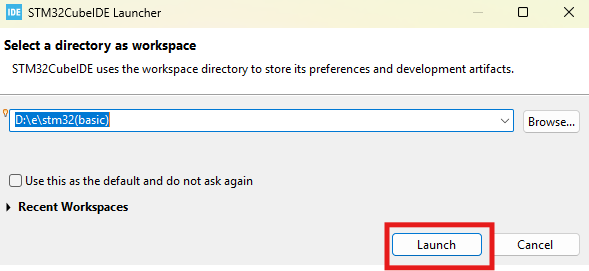
**Step-5:**



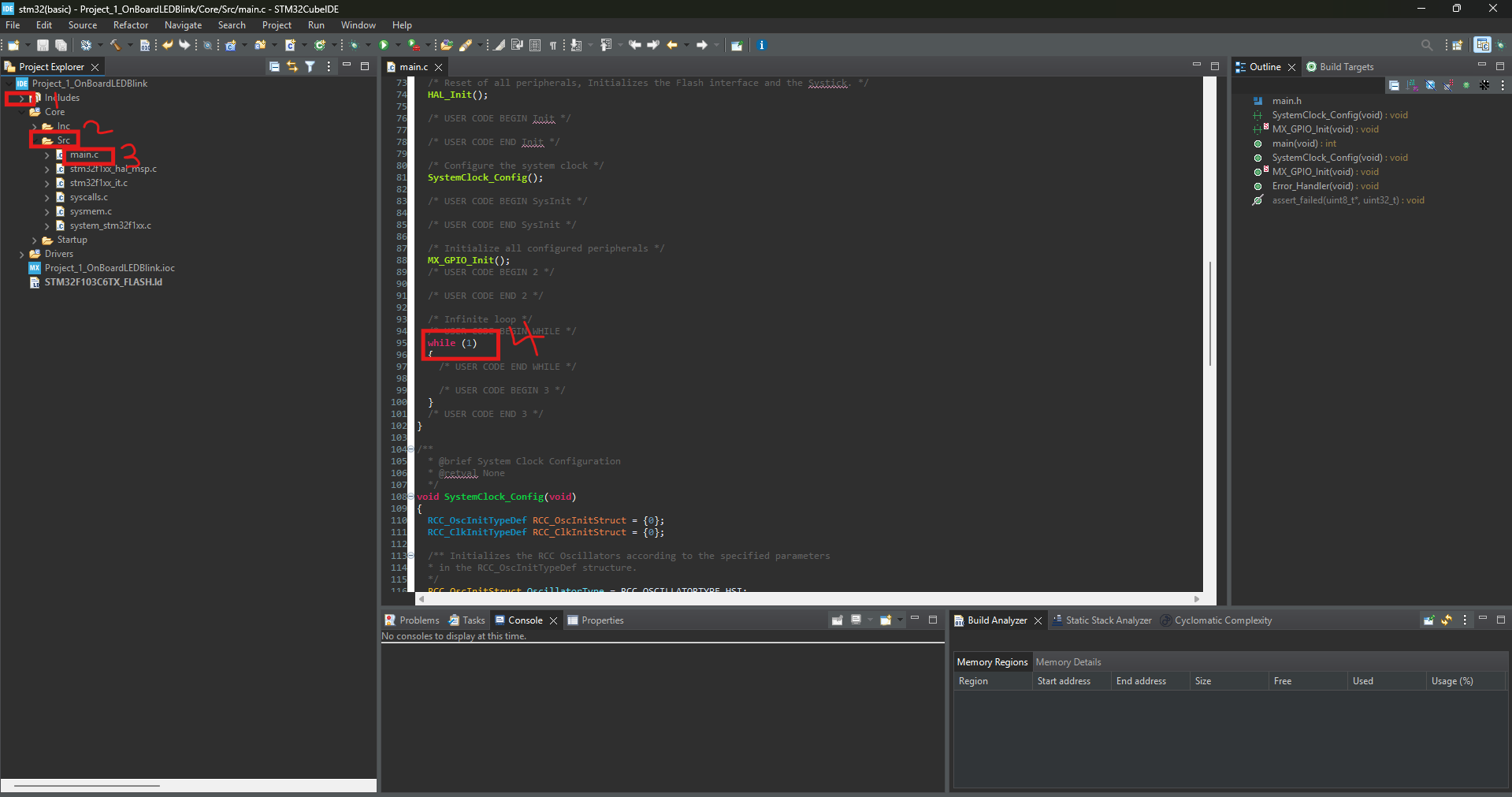


**Step-6:**



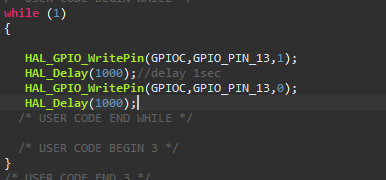
 

**Step-7:** Expand **Core 🡪** Expand **src 🡪 main.c 🡪** Scroll to **while(1)**

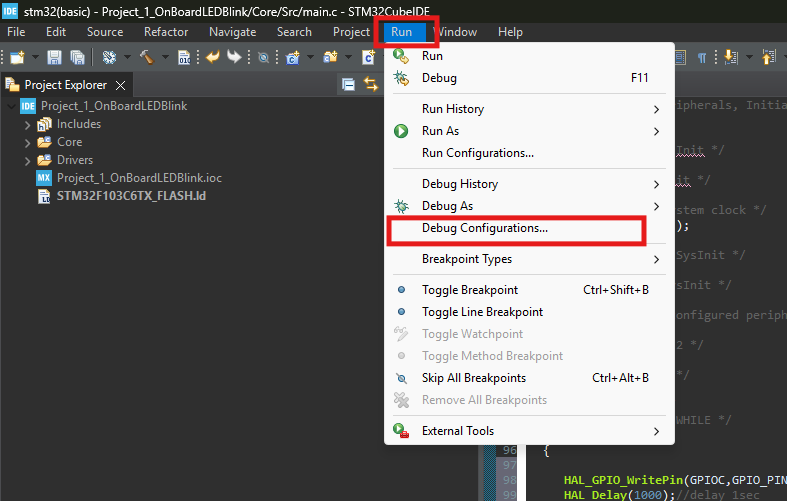
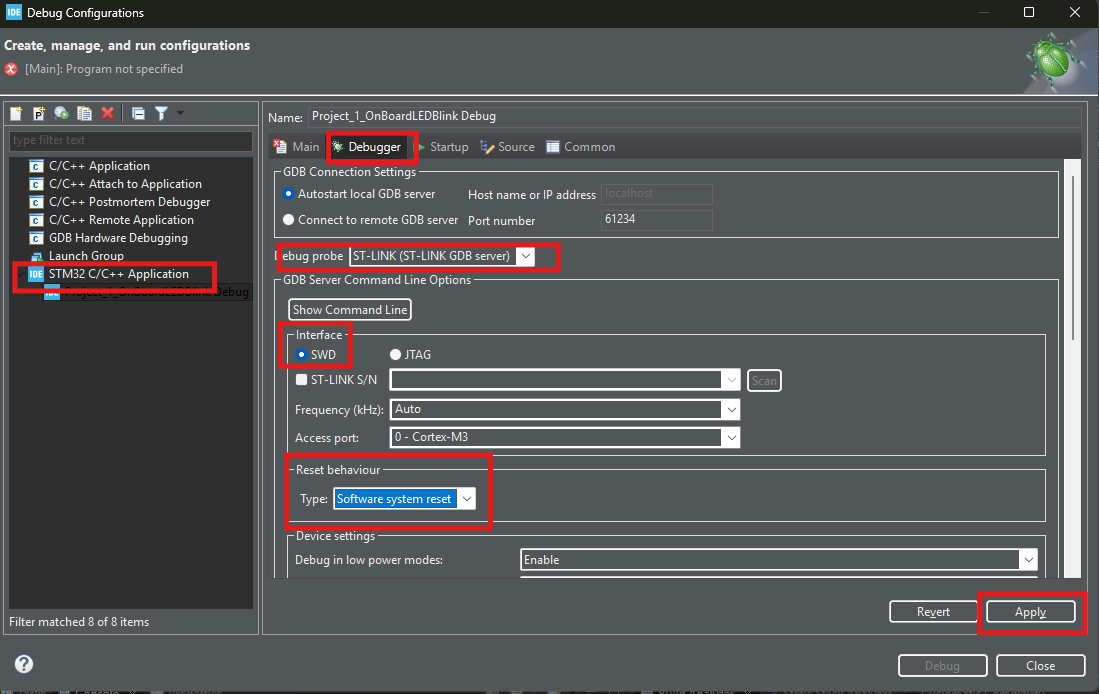




**Code:**

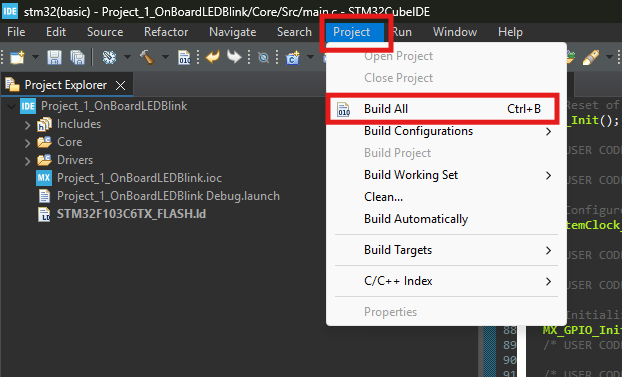


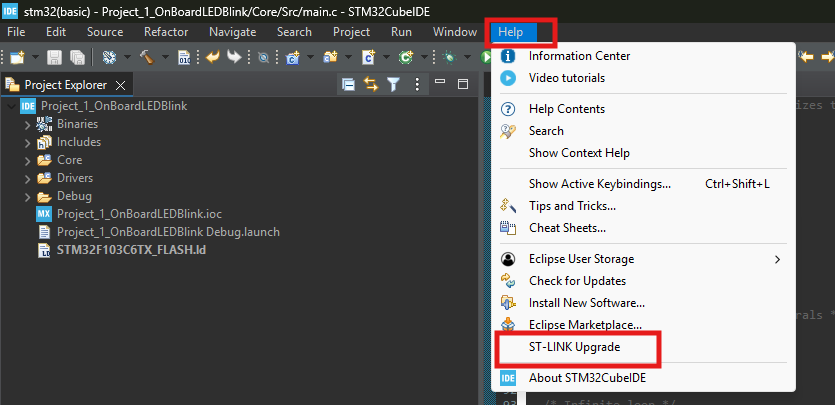
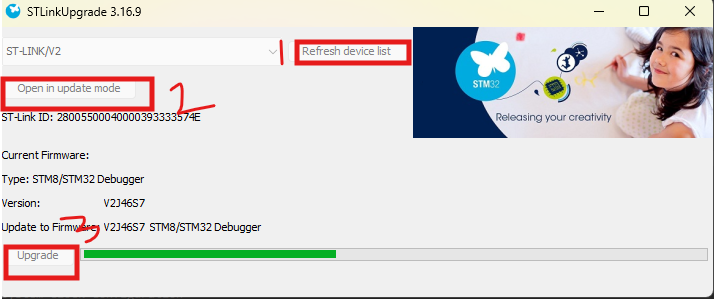
**Step-8:**

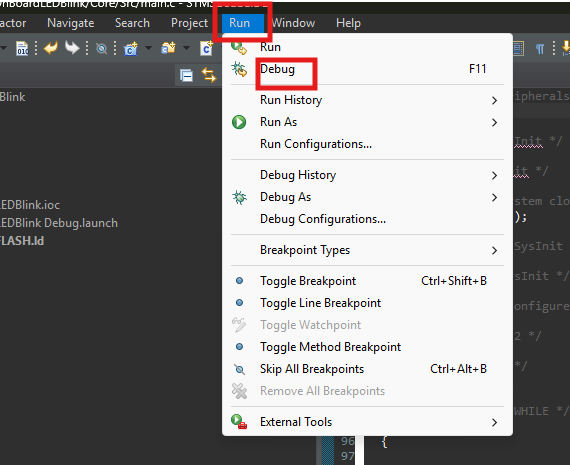
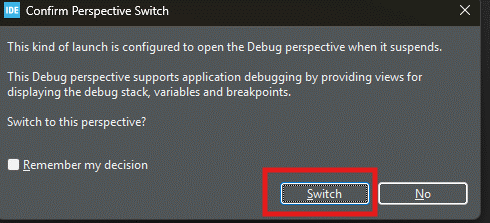


**Step-9:**



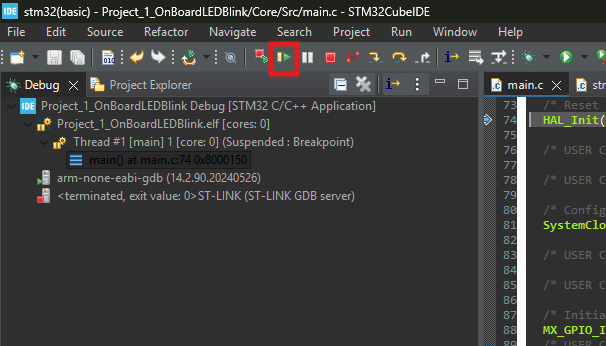
 

**Step-10:**

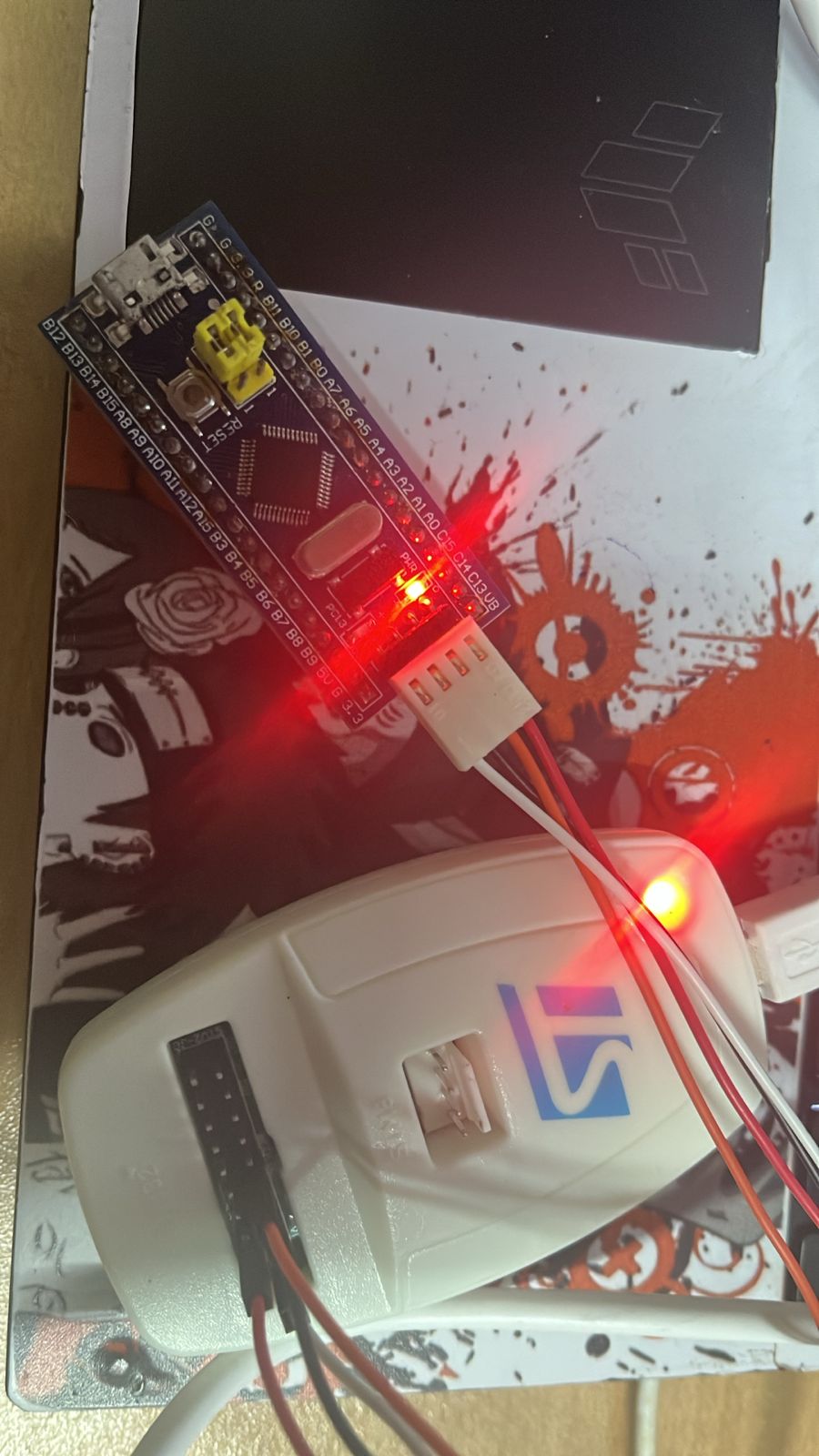
 



**Step-11:**



**Output:**



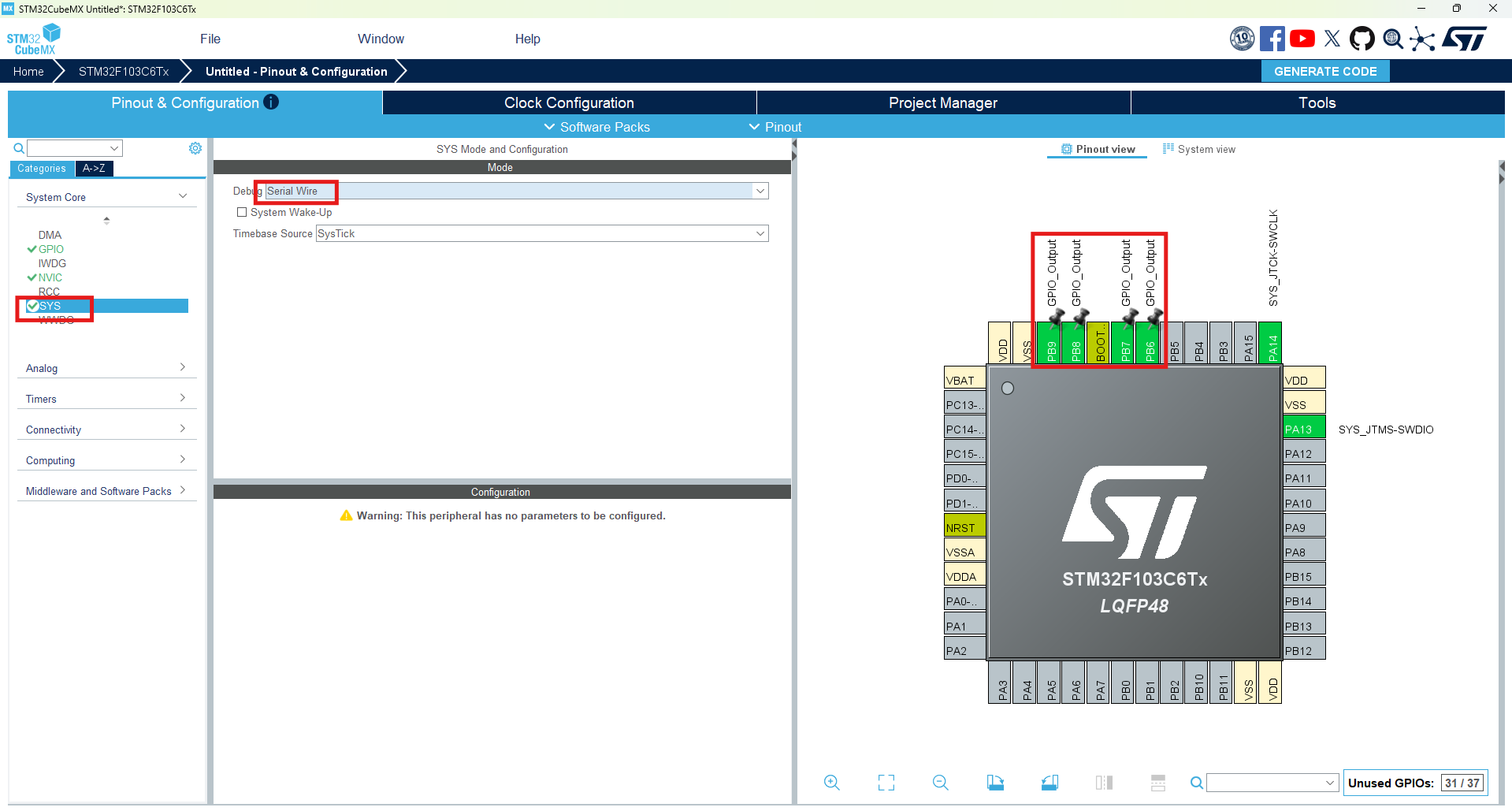
**Project-2: Blink External LED**

**Objective:** To interface and control external LEDs using GPIO pins of the STM32 microcontroller and generate a blinking pattern.

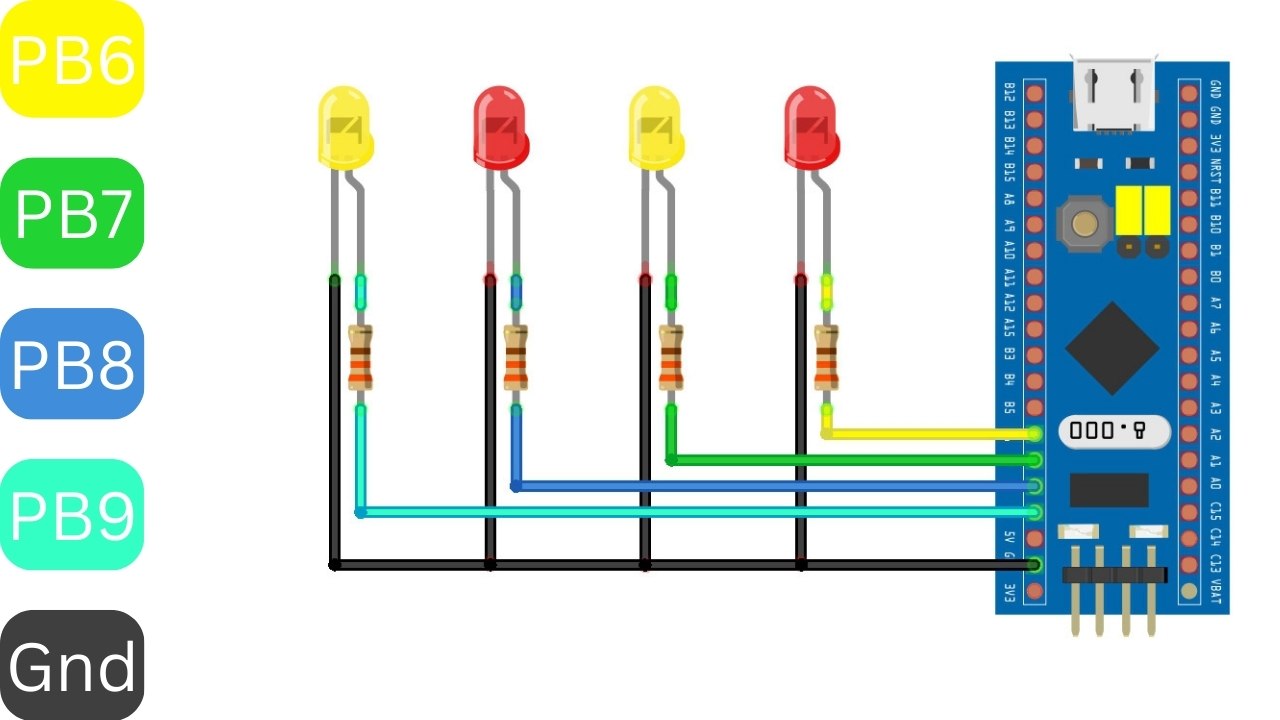
**Learning Outcomes:** Interface external LEDs with proper current-limiting resistors. Configure GPIO pins as output and control external LEDs using STM32CubeIDE

**Procedure/Setup:**

**Step-1:**



**Circuit diagram:**

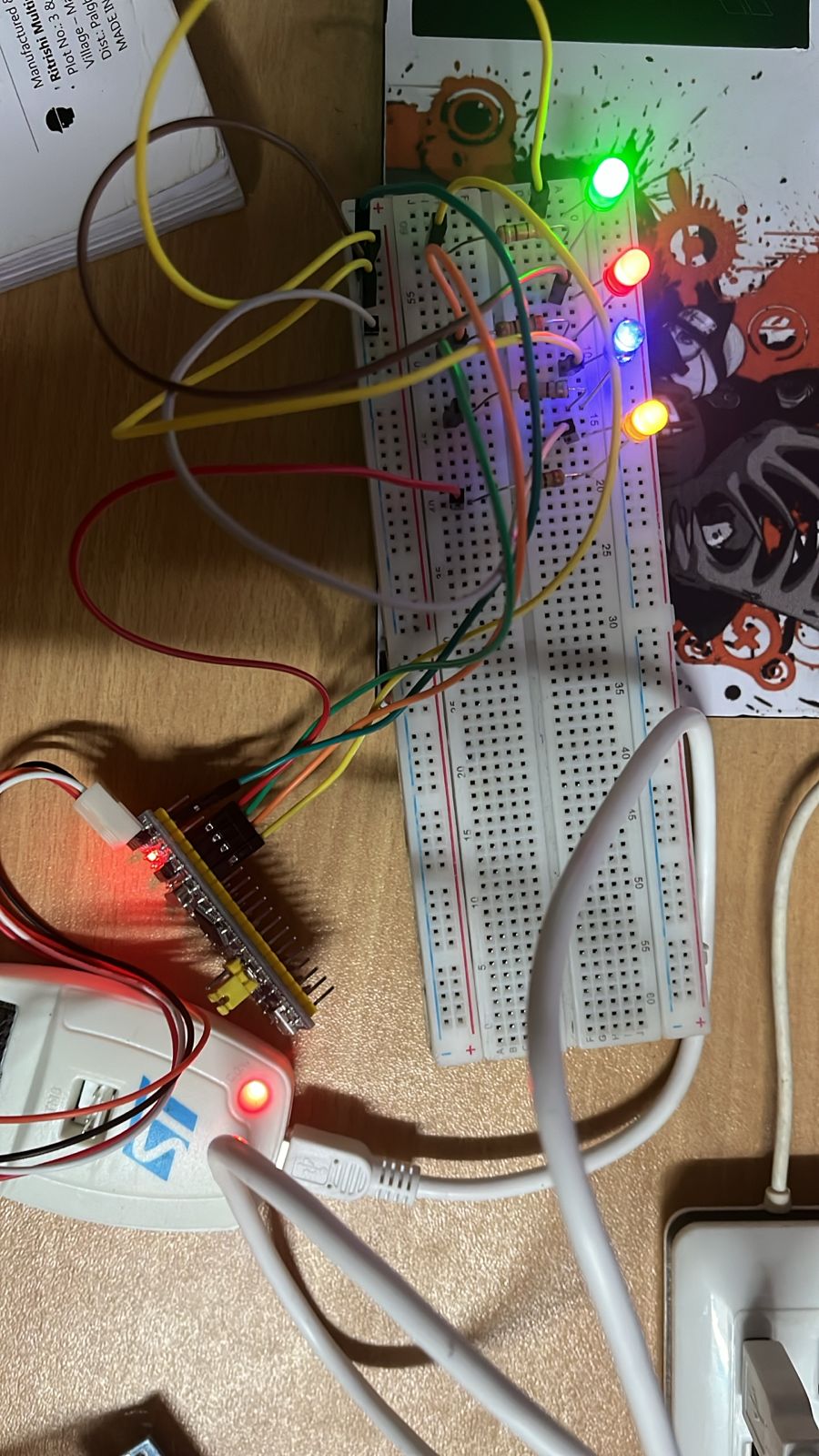
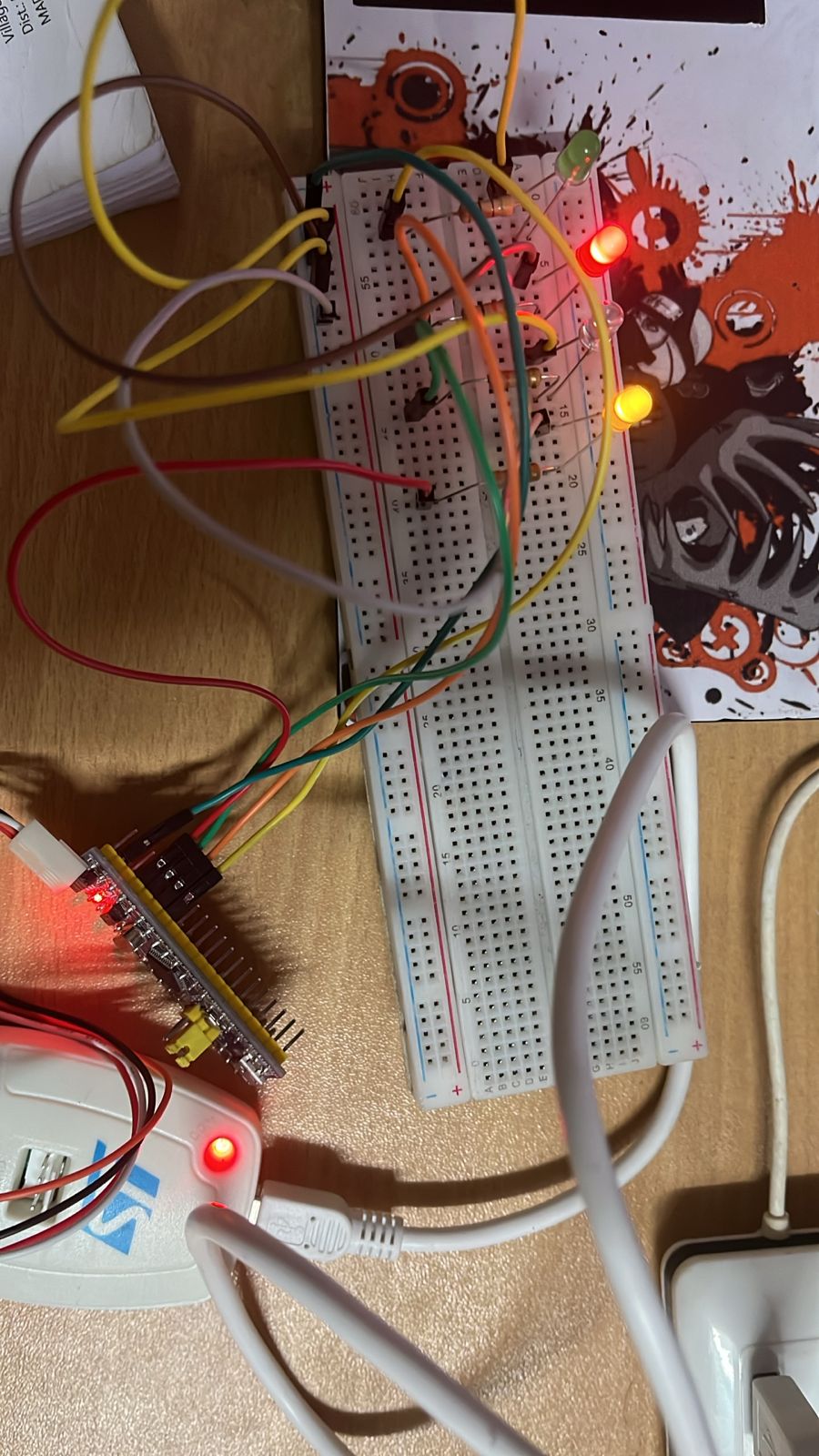




**Code:**



**Output:**



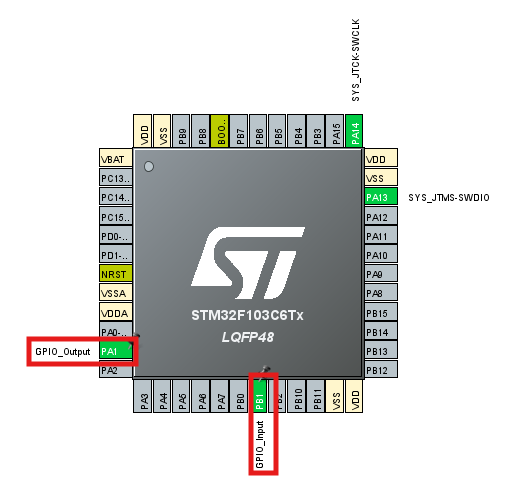
**Project-3: Button controlled LED**

**Objective:** To interface a push button with the STM32 microcontroller and control an external LED based on button input.

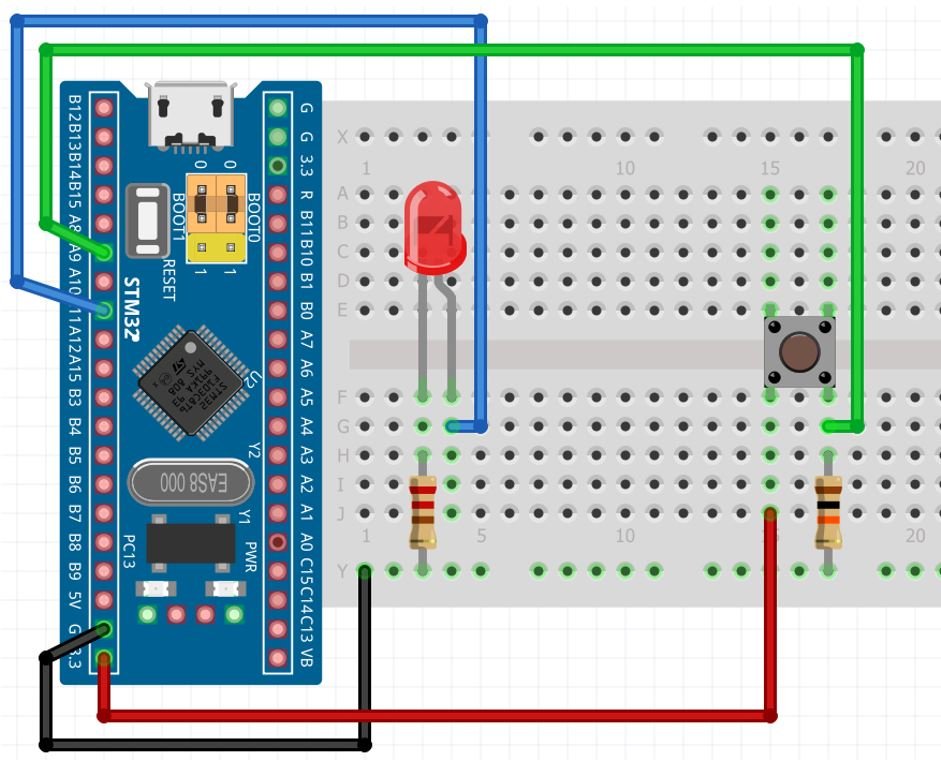
**Learning Outcomes:** Configure GPIO pins as input (button) and output (LED). Read button status and control an LED using embedded C in STM32CubeIDE

**Procedure/Setup:**

**Step-1:**

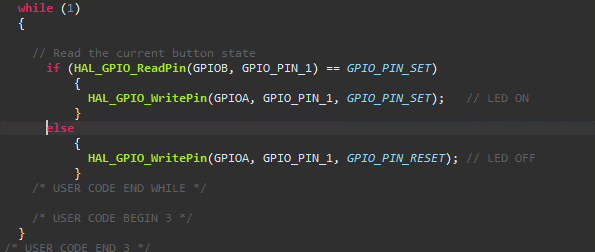


**Circuit Diagram:**

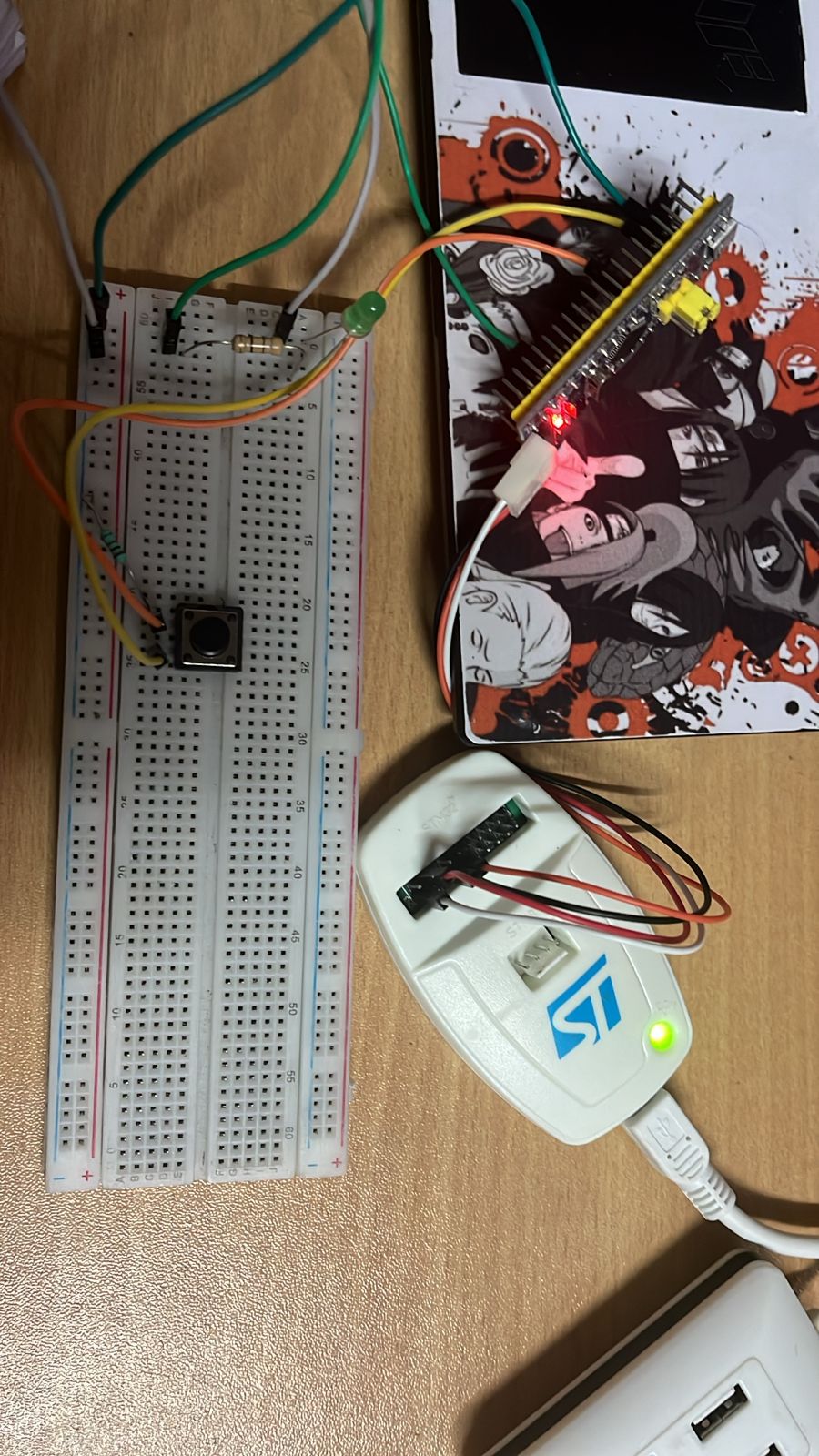
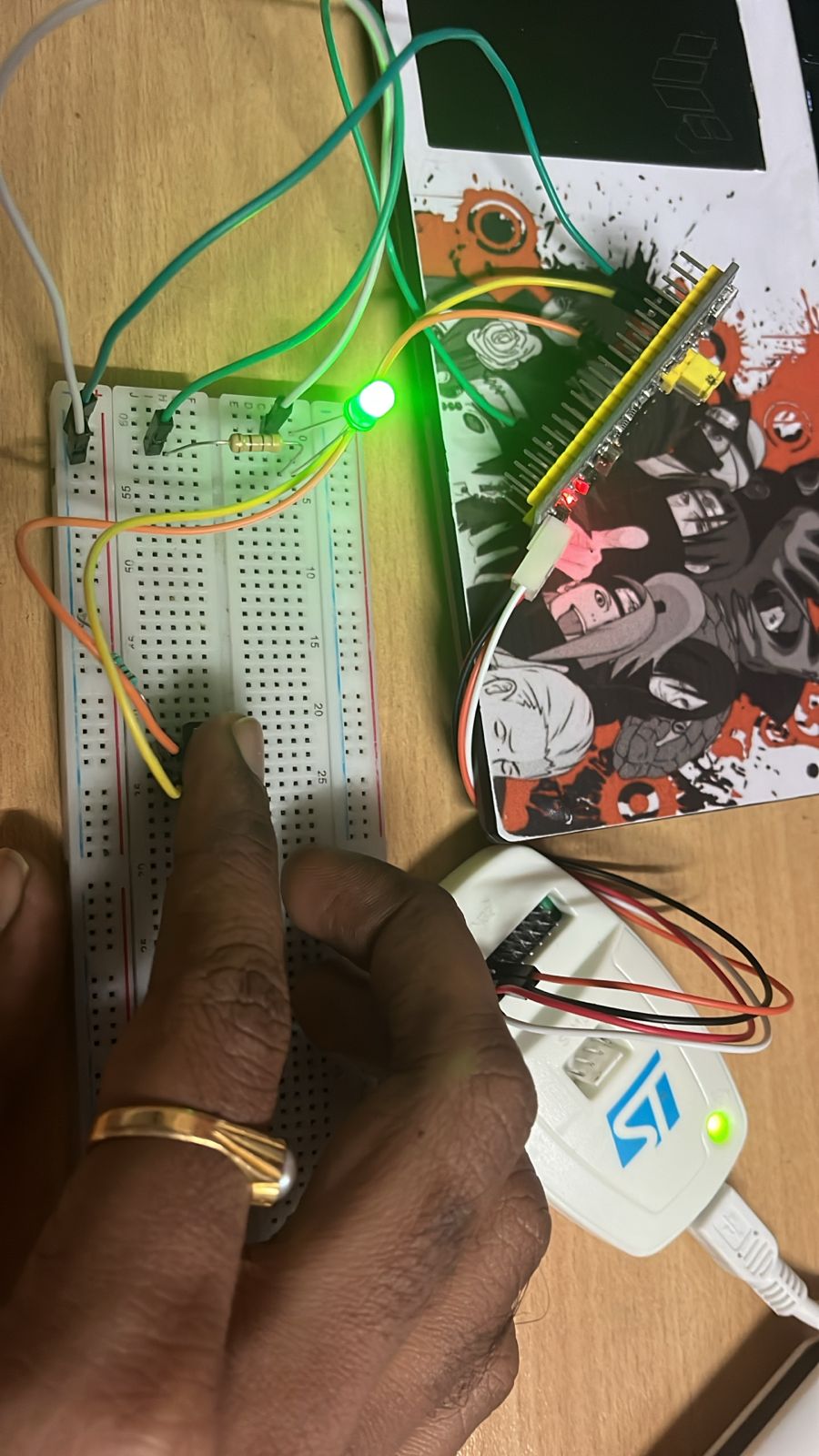




**Code:**



**Output:**



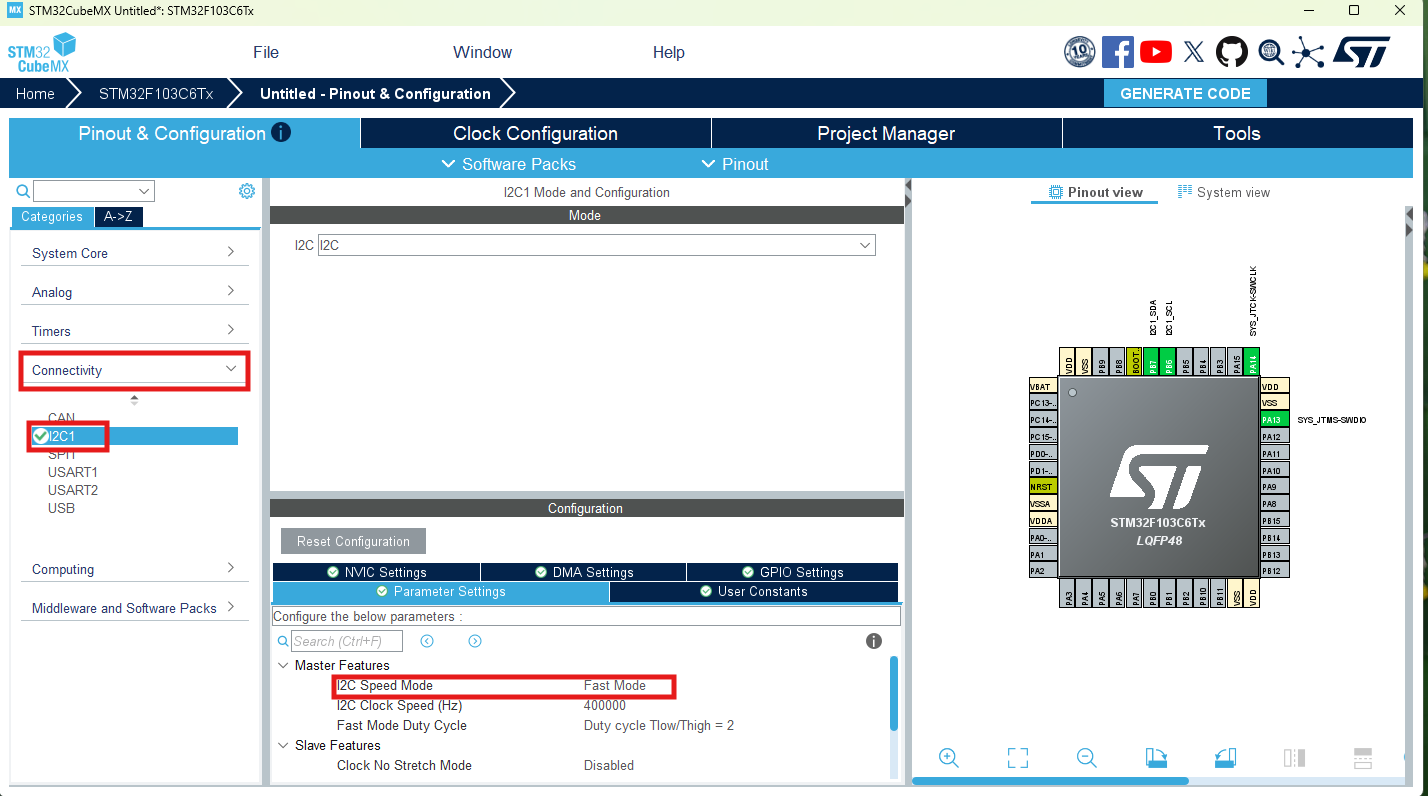
**Project-4: Interfacing OLED**

**Objective:** To interface an OLED display with the STM32 microcontroller and display basic text or graphics using a communication protocol.

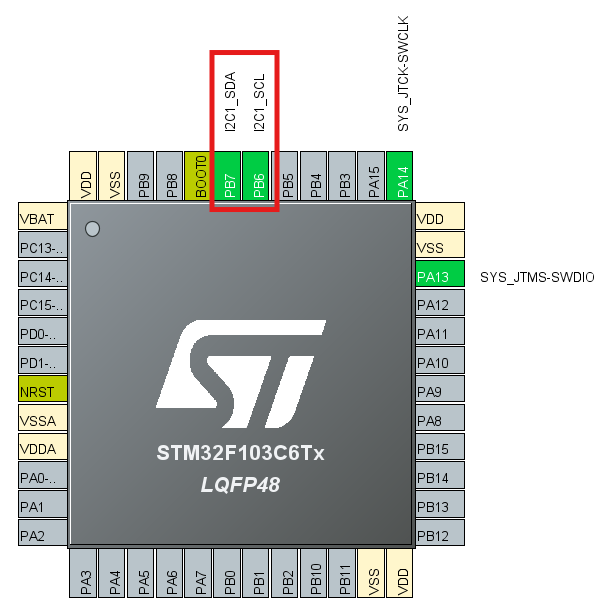
**Learning Outcomes:** Interface an OLED display using I2C protocol. Initialize the display and send data/commands using STM32CubeIDE.

**Setup/Procedure:**

**Step-1:**

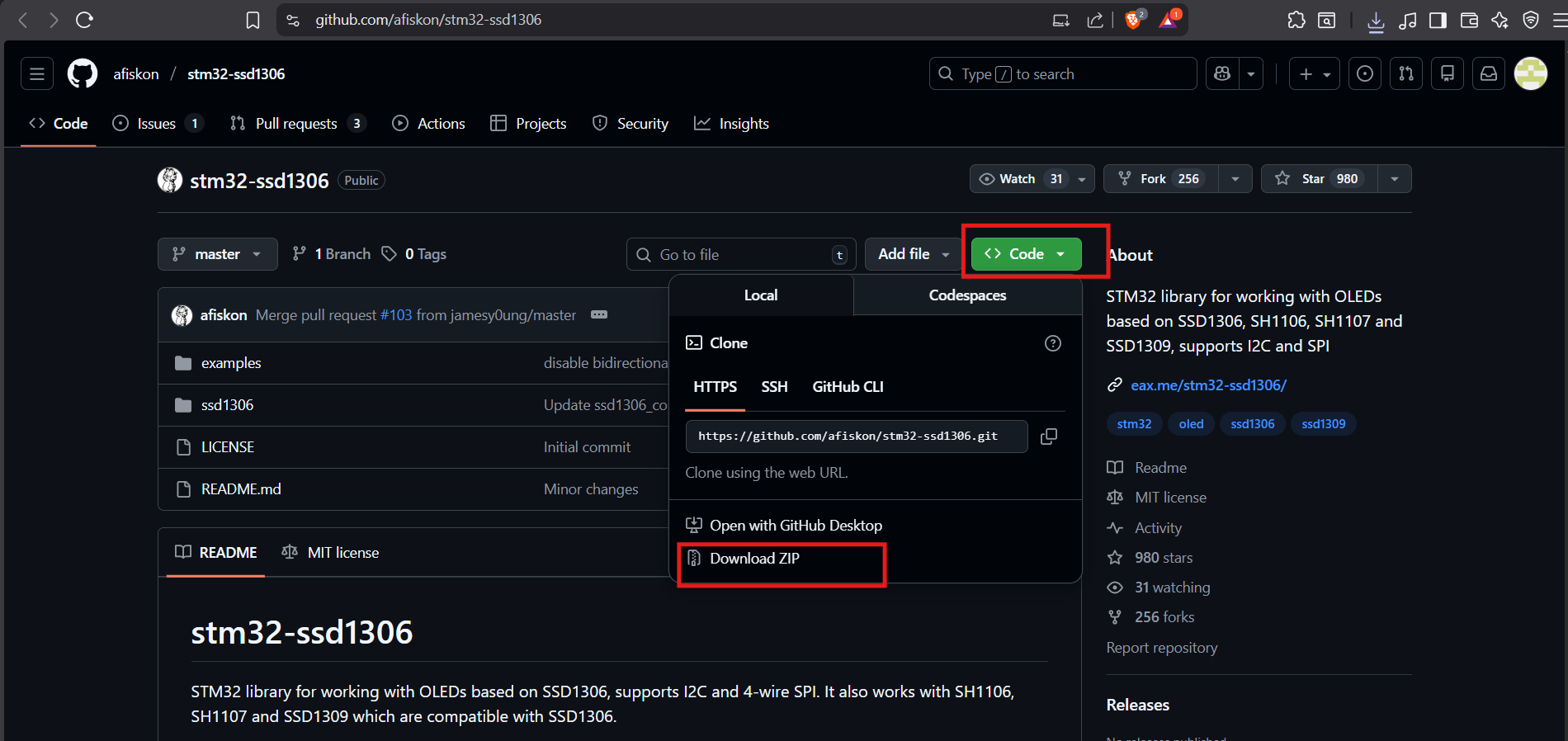


**Step-2:**

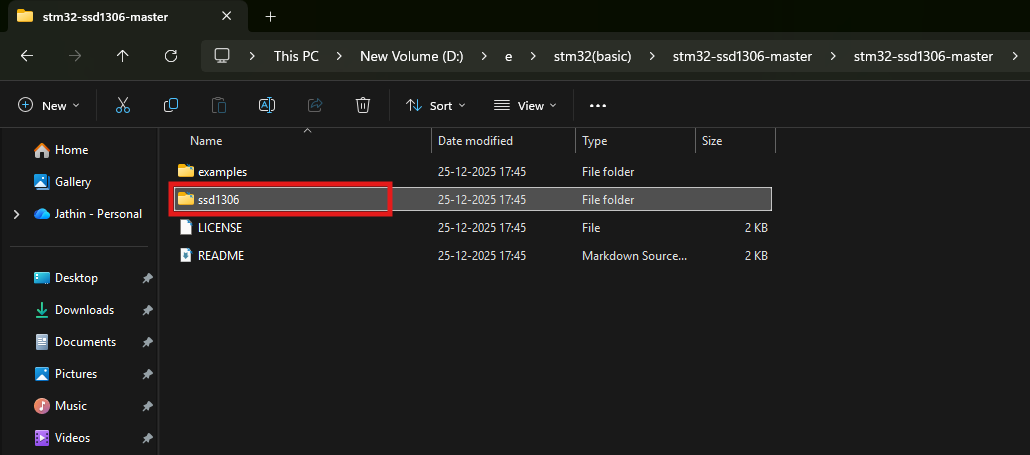
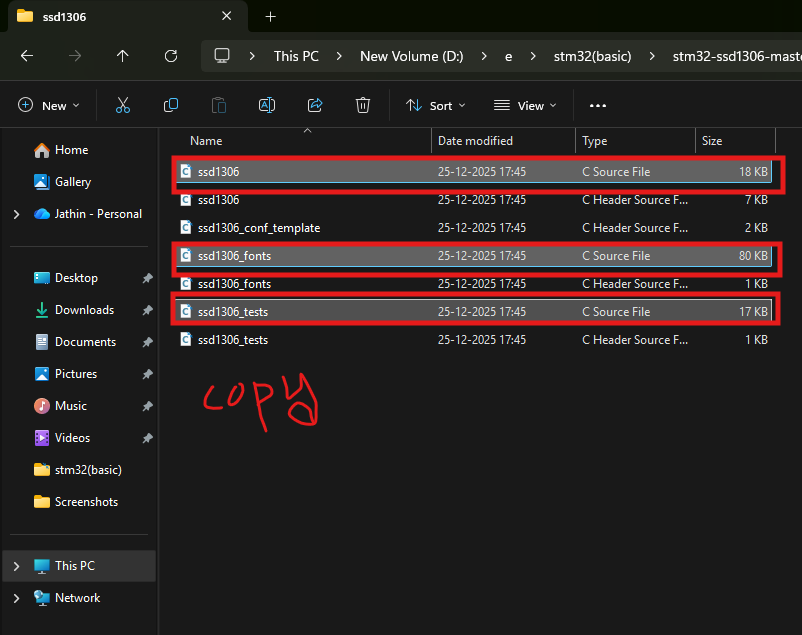




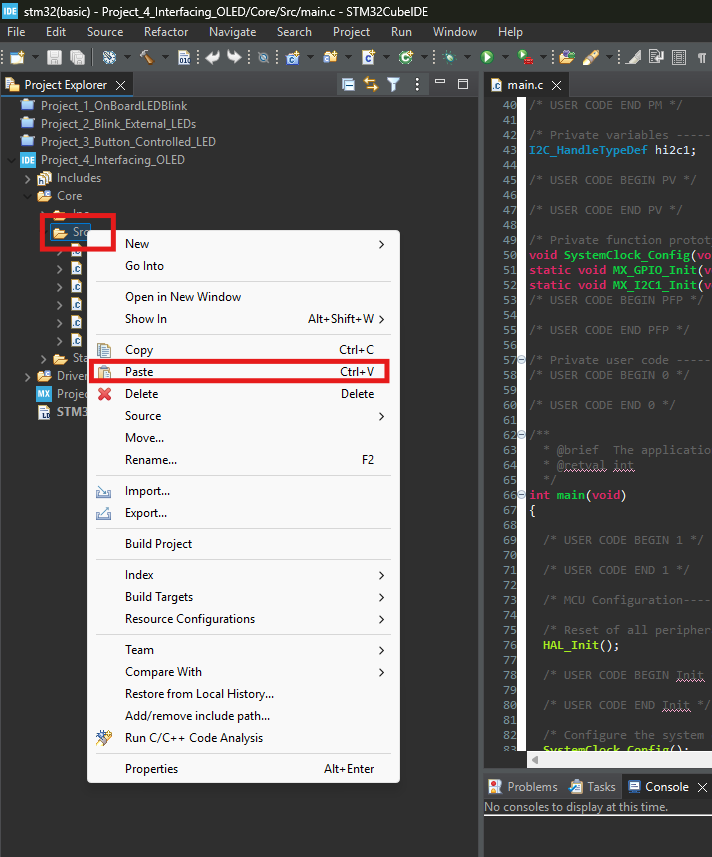
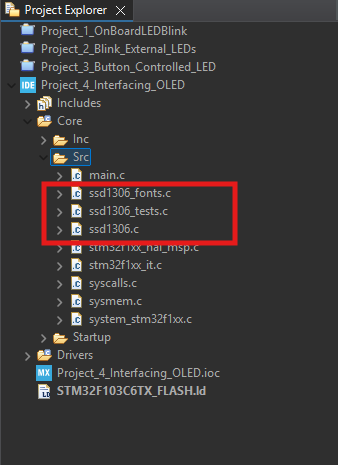
**Step-3:** visit <https://github.com/afiskon/stm32-ssd1306>



**Step-4:**

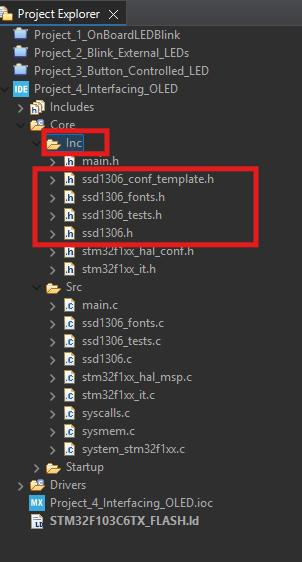
 

**Step-5:**

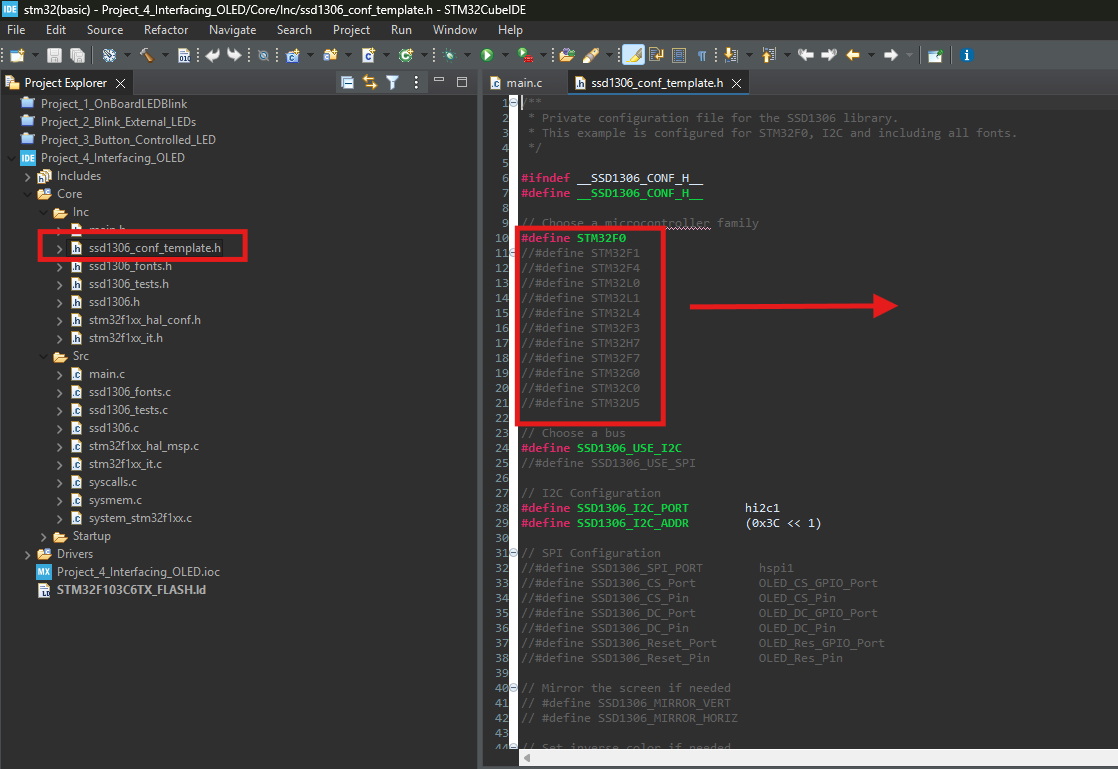
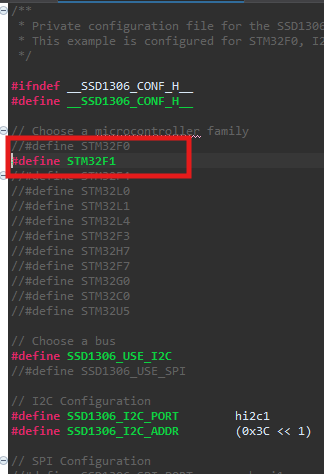
 



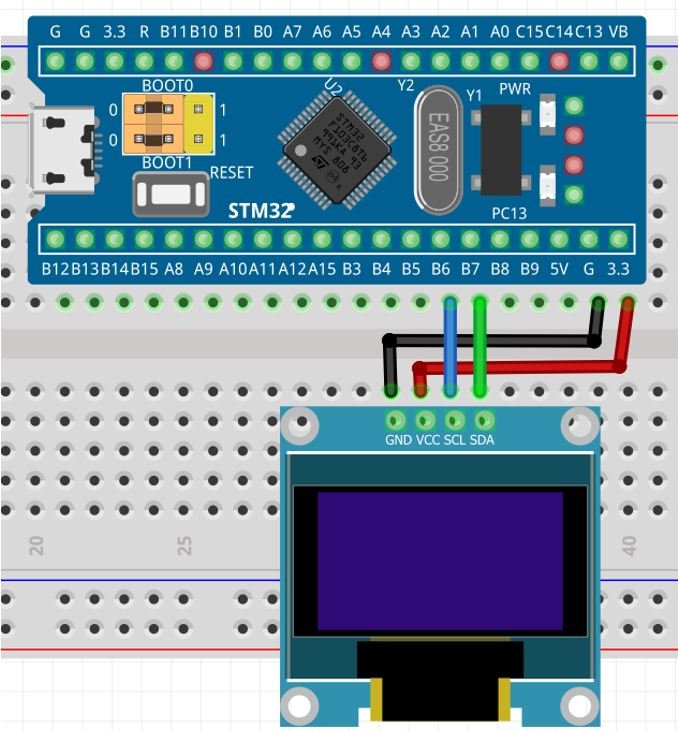
**Step-6:** Repeat the same process with header source file and paste them in Inc.



**Step-7:**

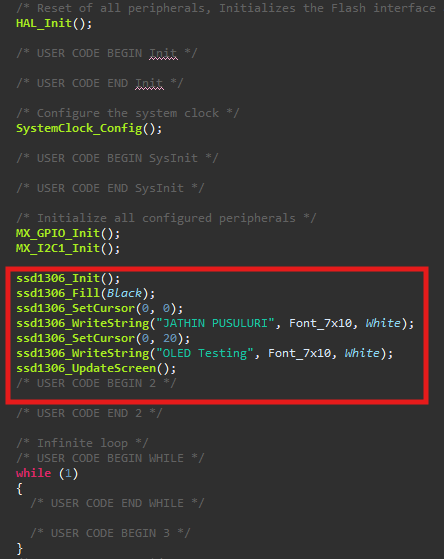
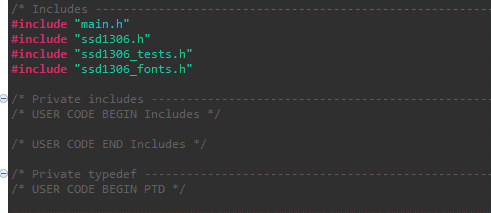
 

**Circuit diagram:**

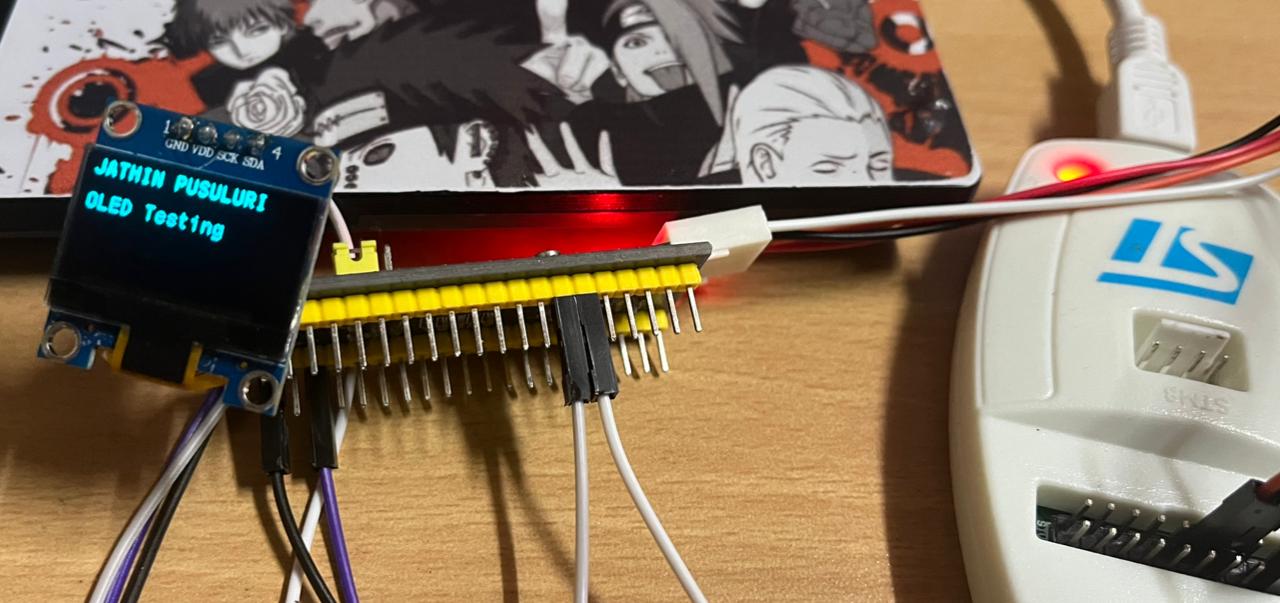




**Code:**



**Output:**





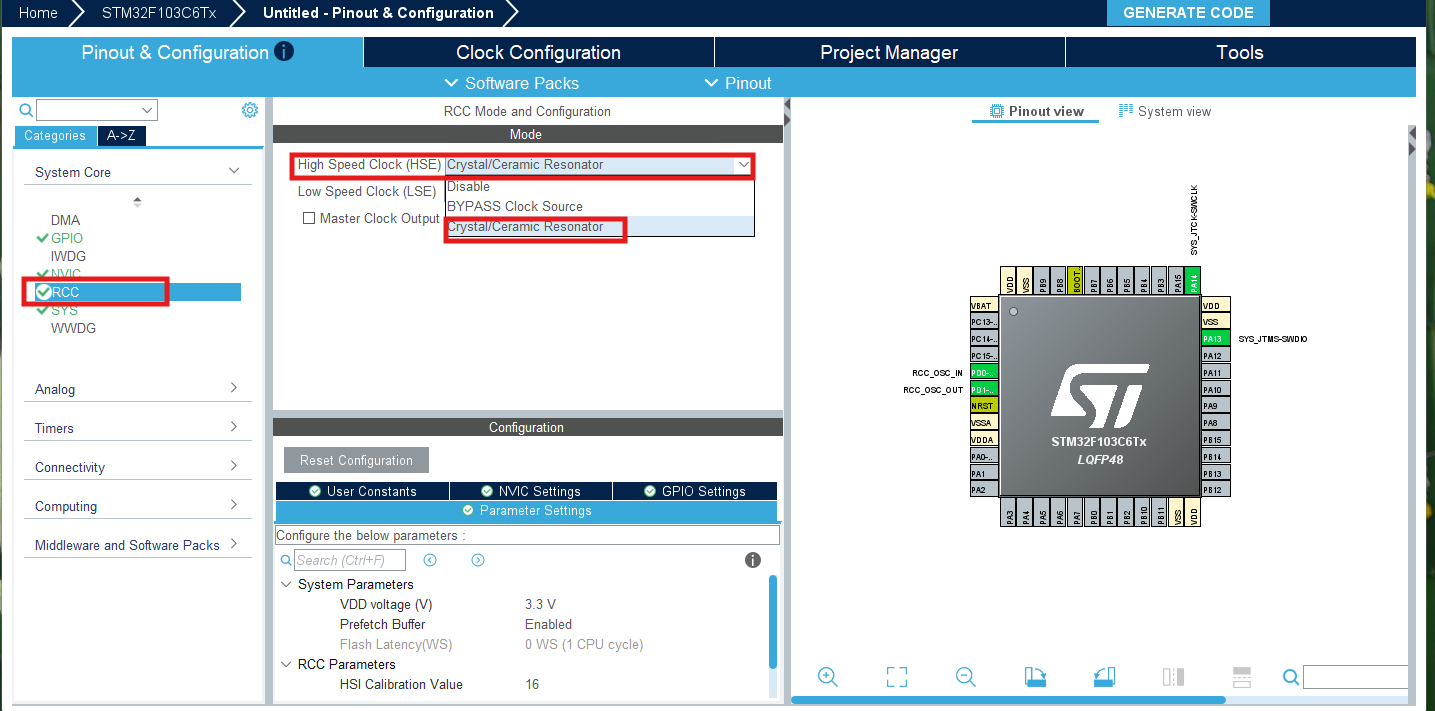
**Project-5: Distance Measurement using Ultrasonic sensor**

**Objective:** To measure distance using an ultrasonic sensor interfaced with the STM32 microcontroller by calculating time of echo signal.

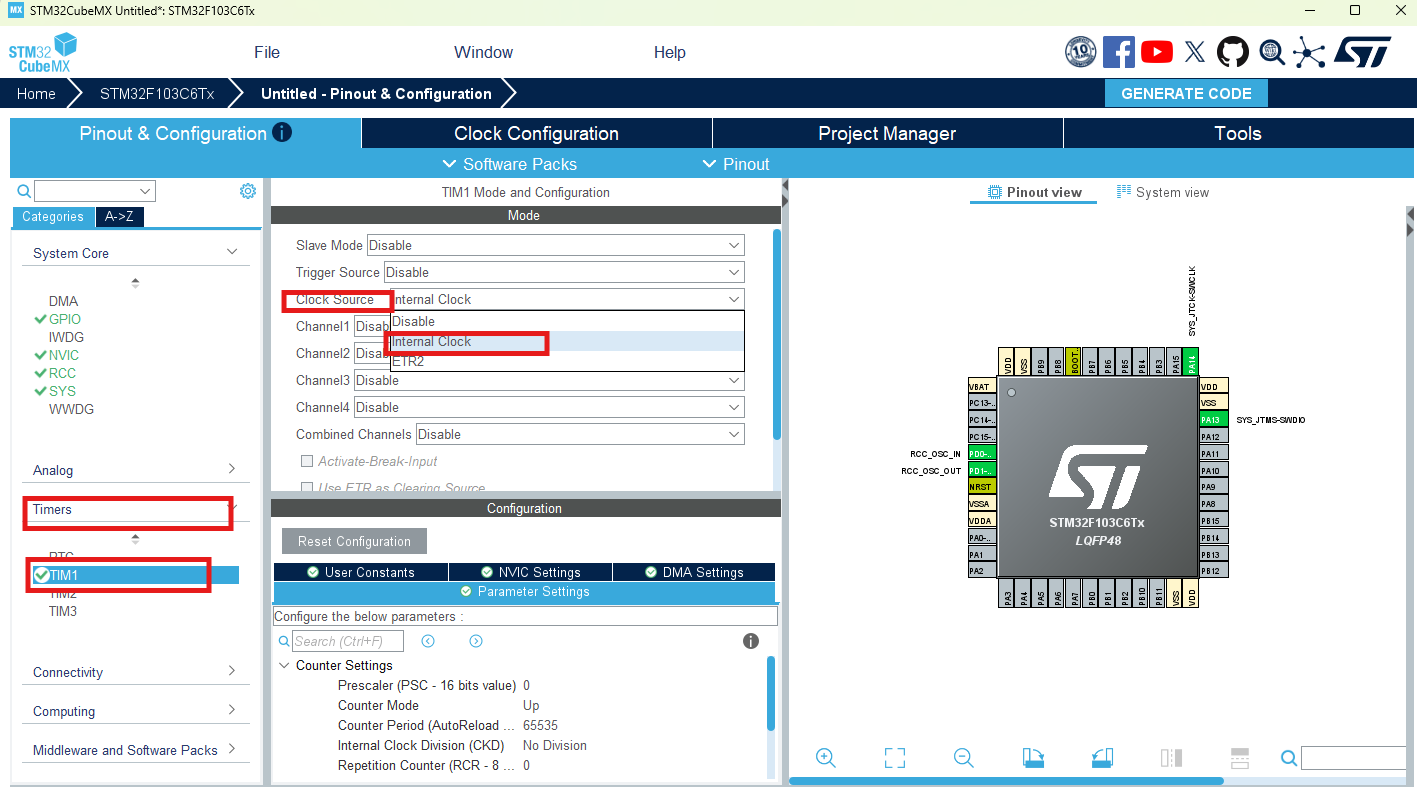
**Learning Outcomes:** Interface an ultrasonic sensor using GPIO and timer/delay functions. Calculate distance based on ultrasonic pulse timing using embedded C.

**Setup/Procedure:**

**Step-1:**

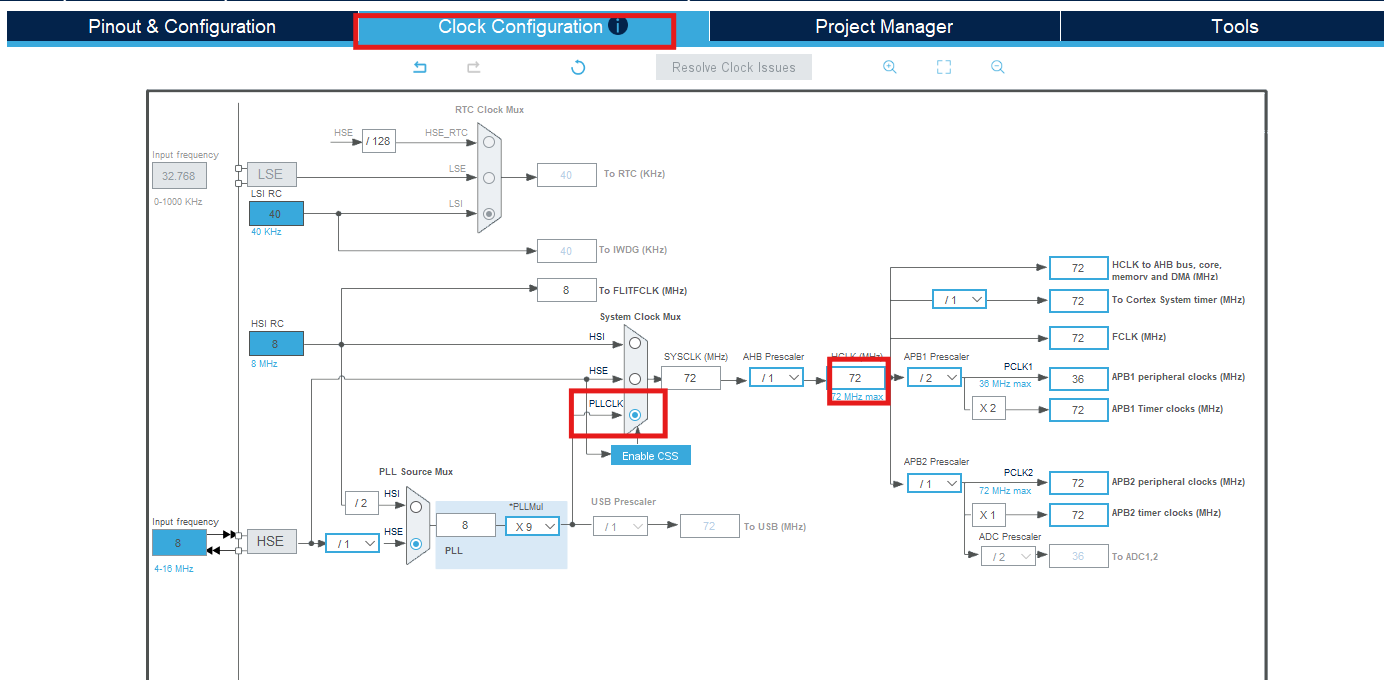


**Step-2:**

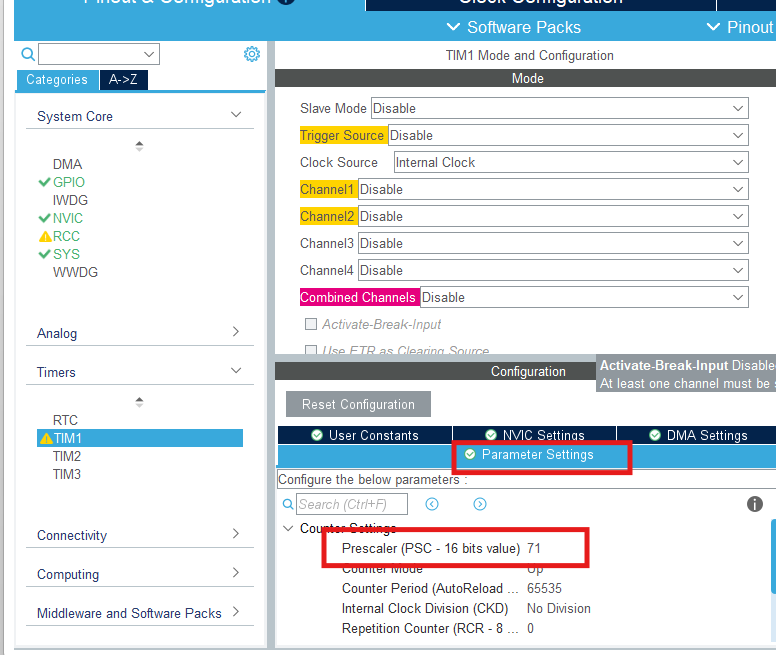




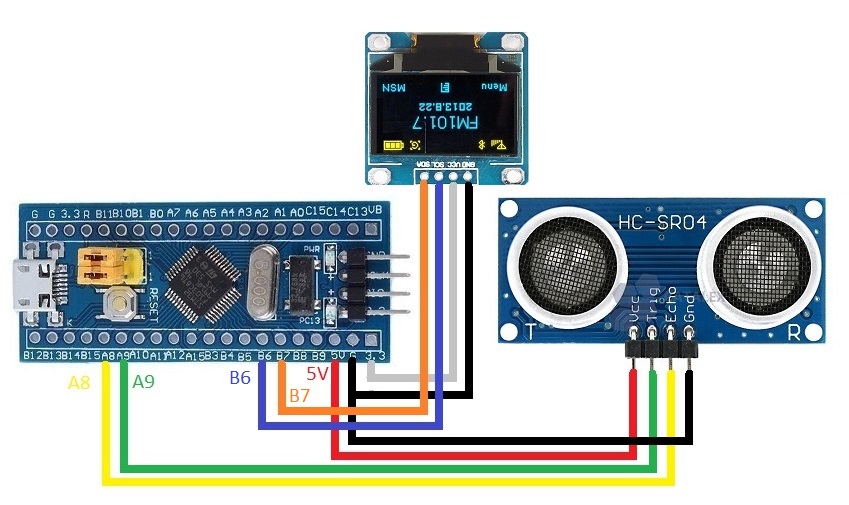
**Step-3:**



**Step-4:**

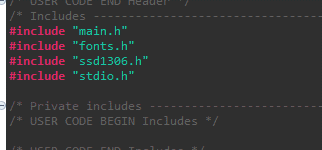


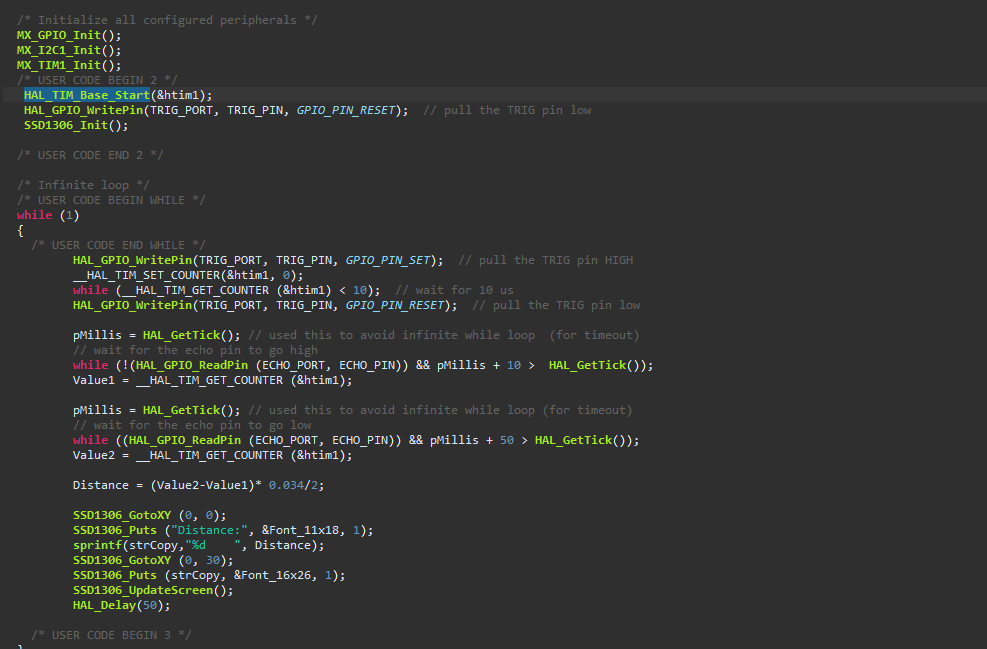
**Circuit diagram:**





**Code:**







**Output:**

