✅ Objective

Toggle an LED every 1 second using **TIM2 interrupt**.

🧠 Key Concepts

1. **Timer Interrupts**

- TIM2 can be configured to trigger an **interrupt** at a specific interval (e.g., every 1 second).
- When that happens, a special **callback function** gets called automatically by the HAL (Hardware Abstraction Layer).

---

2. **Callback Function**

- In STM32 HAL, when a timer reaches the configured period, HAL executes:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
```

- This is called from the HAL's **IRQ Handler** internally.
- You **don't call it yourself** — it's triggered automatically when the interrupt occurs.

🛠️ Step-by-Step Configuration in STM32CubeIDE

Step 1: Create a new project

- Use **STM32CubeIDE**.
- Select your STM32 chip or board (e.g., STM32F103C8T6 or Nucleo).

PROF

Step 2: Configure TIM2

1. Open **.ioc file**
2. Go to **Timers → TIM2**
3. Set Mode to **Internal Clock**
4. Click on **NVIC Settings** and enable **TIM2 global interrupt**
5. Set Prescaler and Period to get 1 second:

Example (for 72 MHz clock):

```
Prescaler = 7200 - 1    // Divides 72 MHz to 10 kHz
Period    = 10000 - 1   // Overflows every 1 second
```

Step 3: Configure LED pin

- Go to **GPIO**
- Set one pin (e.g., PA5) to **GPIO Output** (for toggling LED)

Step 4: Generate Code

Click **Project → Generate Code**, then open `main.c`.

Step 5: Start the timer in interrupt mode

In `main.c`, inside `main()` after HAL_Init():

```
HAL_TIM_Base_Start_IT(&htim2);  // Start TIM2 in interrupt mode
```

Step 6: Implement the Callback

In `stm32f1xx_it.c` or `main.c`, add:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Instance == TIM2)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);  // Toggle the LED
    }
}
```

PROF

🔁 What Happens at Runtime

1. TIM2 counts based on the clock and prescaler.
2. After reaching the `Period` value, it **overflows**.
3. An **interrupt** is generated.
4. HAL calls:

   ```
   HAL_TIM_IRQHandler(&htim2);
   ```

5. Inside this handler, `HAL_TIM_PeriodElapsedCallback()` is executed **automatically**.

6. Your code inside the callback toggles the LED.

🔍 Summary

Step	What It Does
Timer Setup	Configures time base and interrupt enable
`HAL_TIM_Base_Start_IT()`	Starts the timer in interrupt mode
Callback	Your function that executes on overflow
LED Toggle	Done inside callback every 1 second

✅ Optional Debug Tip

Add a breakpoint inside `HAL_TIM_PeriodElapsedCallback()` to see when it's triggered.
