

```

#include "main.h"
#include "stm32f3xx_hal_tim.h"
void SystemClock_Config(void);

void TIM1_CC_IRQHandler(void)
{
    if (TIM1->SR & TIM_SR_CC1IF) {
        // CC1割り込みフラグがセットされたときに実行される処理
        // LEDの状態を反転させる
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_5);

        // CC1割り込みフラグをクリアする
        TIM1->SR &= ~TIM_SR_CC1IF;
    }
}

int main(void)
{
    HAL_Init();

    SystemClock_Config();

    __HAL_RCC_GPIOA_CLK_ENABLE();

    // GPIOAのクロックを有効化する
    RCC->AHBENR |= RCC_AHBENR_GPIOAEN;

    // GPIOAのピン5を出力に設定する
    GPIOA->MODER |= GPIO_MODER_MODER5_0;

    // TIM1のクロックを有効化する
    RCC->APB2ENR |= RCC_APB2ENR_TIM1EN;

    // TIM1を設定する
    TIM1->PSC = 16000 - 1; // プリスケアラの設定
    TIM1->ARR = 1000 - 1; // カウンタの最大値の設定
    TIM1->DIER |= TIM_DIER_CC1IE; // CC1割り込みを有効化

    // TIM1の割り込みを設定する
    NVIC_EnableIRQ(TIM1_CC_IRQn);
    NVIC_SetPriority(TIM1_CC_IRQn, 0);

    // TIM1を有効化する
    TIM1->CR1 |= TIM_CR1_CEN;

    while (1) {
        // HAL_GPIO_TogglePin(GPIOA,GPIO_PIN_5);
        // HAL_Delay(1000);
        // メインループで何か他の処理を行う
    }
}

```

```

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

    /** Initializes the RCC Oscillators according to the specified parameters
     * in the RCC_OscInitTypeDef structure.
     */
    RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSI;
    RCC_OscInitStruct.HSISState = RCC_HSI_ON;
    RCC_OscInitStruct.HSICalibrationValue = RCC_HSICALIBRATION_DEFAULT;
    RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
    RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSI;
    RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL16;

    /** Initializes the CPU, AHB and APB buses clocks
     */
    RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_HCLK|RCC_CLOCKTYPE_SYSCLK
        |RCC_CLOCKTYPE_PCLK1|RCC_CLOCKTYPE_PCLK2;
    RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
    RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
    RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV2;
    RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
}

```