LAB NOTE

Subject: Hardware/Software Interfacing

Lab 4: Timers

Student: Minh Quan Tran

Oct 03rd, 2024

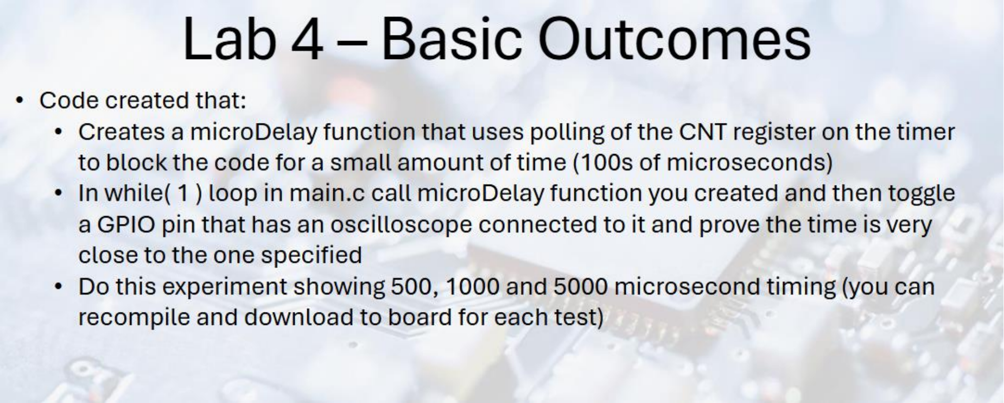
Table of Contents

TABLE OF FIGURES

1. Objectives

- Using STM32F411 board and STM32CubeIDE in Windows, create code to
  • Write code for the STM32F411 to configure and use Timers.

Basic:

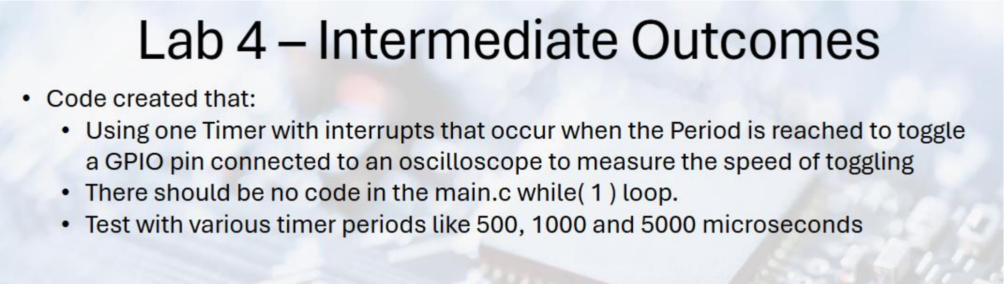


Lab 4 – Basic Outcomes

- Code created that:
  - Creates a microDelay function that uses polling of the CNT register on the timer to block the code for a small amount of time (100s of microseconds)
  - In while(1) loop in main.c call microDelay function you created and then toggle a GPIO pin that has an oscilloscope connected to it and prove the time is very close to the one specified
  - Do this experiment showing 500, 1000 and 5000 microsecond timing (you can recompile and download to board for each test)

Figure 1-1: Basic outcome

- Intermediate:

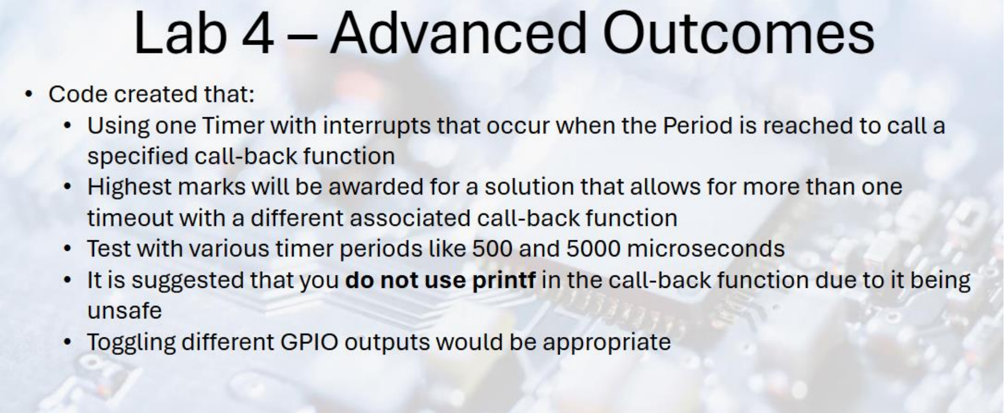


Lab 4 – Intermediate Outcomes

- Code created that:
  - Using one Timer with interrupts that occur when the Period is reached to toggle a GPIO pin connected to an oscilloscope to measure the speed of toggling
  - There should be no code in the main.c while(1) loop.
  - Test with various timer periods like 500, 1000 and 5000 microseconds

Figure 1-2: Intermediatet outcome

- Advance:



Lab 4 – Advanced Outcomes

- Code created that:
  - Using one Timer with interrupts that occur when the Period is reached to call a specified call-back function
  - Highest marks will be awarded for a solution that allows for more than one timeout with a different associated call-back function
  - Test with various timer periods like 500 and 5000 microseconds
  - It is suggested that you **do not use printf** in the call-back function due to it being unsafe
  - Toggling different GPIO outputs would be appropriate

Figure 1-3: Advance outcome

2. Problems and Solutions

2.1 Problems

- No problem

2.2 Solutions

3. Software Design

3.1 List of function

- This function is to create a delay in micro second by counting the number of ticks has passed using CNT registers.

```
void microDelay(uint32_t usDelay)
{
    // Get the current timer counter value
    uint32_t startTime = __HAL_TIM_GET_COUNTER(&htim1);
    uint32_t ticks = usDelay - 1; // 1 ticks = 1 us

    // Poll the CNT register until the specified number of ticks has passed
    while ((__HAL_TIM_GET_COUNTER(&htim1) - startTime) < ticks)
    {
        // Wait until the timer reaches the required ticks
    }
}
```

- This function is used to toggle LED whenever the timer overflow

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    if (htim->Instance == TIM1)
    {
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
    }
}
```

3.2 While loop

```
  while (1)
  {
#ifndef USE_INTERRUPT
        microDelay(PERIOD);
        HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_6);
#endif
    /* USER CODE END WHILE */

    /* USER CODE BEGIN 3 */
  }
  /* USER CODE END 3 */
}
```

REFERENCES