

**ECE 271 Microcomputer Architecture and Applications****Lab 7: Timer Input Capture in C****Instructor: Prof. Yifeng Zhu****Spring 2015****Goals**

1. Understand the basic concept of input capture function of a timer
2. Handle different events in the interrupt service routine
3. Handle timer counter overflow and underflow
4. Use a timer to measure the timestamp of a signal edge (rising or falling edge) external to the microprocessor

**Pre-Lab Assignment**

1. Read Chapter 15 of the Textbook.
2. Complete the pre-lab assignment (Part 1 and Part 2).
3. Write your answers to the pre-lab assignment in Readme.md and submit it.

**In-Lab Requirement**

1. Measure the period of a 1Hz square signal by using TIM4 Input Capture and calculate the accuracy
2. Measure distance using the ultrasonic sensor and calculate the accuracy
3. Something cool such as printing distance to the LCD display

**Post-Lab Assignment**

1. Complete the post-lab report
2. Write your report in Readme.md and submit it

**Timer Interrupt Status Register**

The free-run timer counter (CNT) of timers used in this lab is limited to 16 bits. There are two special events:

- When counting up, CNT restarts from 0 after it reaches 0xFFFF. This is called *counter overflow*.
- When counting down, CNT restarts from 0xFFFF after it reaches 0. This is called *counter underflow*.

When a counter overflow or underflow occurs, the timer can generate a timer interrupt if the Update Interrupt Enable (UIE) bit of the TIM DMA/Interrupt Enable Register (TIM\_DIER) is set.

In the timer interrupt service routine, you can check the TIM status register (TIM\_SR) to find out what events has generate the timer interrupt.

- If a counter overflow or underflow occurs, the Update Interrupt Flag (UIF) is set in TIM\_SR. The UIF flag is set by hardware.

- If a channel is configured as input, a valid transition of an external signal can trigger the timer interrupt. Take channel 1 as an example. If the channel 1 is configured as input capture and the Channel 1 Interrupt Flag (CC1IF) in TIM\_SR is set, the capture on Channel 1 has been triggered and the counter value has been copied to CCR1 register. The CC1IF is set by hardware.

The timer interrupt service routine must clear these flags of TIM\_SR to prevent it from being called again by the processor. The CC1IF flag is automatically cleared if the TIM\_CCR1 register is read. The UIF flag has to be explicitly cleared by software.

### *Part 1: Measure the period of 1 Hz square signal*

Pin	Available Alternative Functions
PB 6	I2C1_SCL/ <b>TIM4_CH1</b> /USART1_TX

Use an external function generator to generate a 1Hz square wave. Use your timer input capture function (specifically, PB 6/Timer 4 Channel 1) to measure the period of the square wave signal.

Make sure the output signal is limited to 3V to prevent damage to your board. It is highly recommended to connect the function generator to your board via a 10 kΩ resistor, limiting currents to a few milliamps.

In this lab, you are required to select High-Speed Internal Clock (HSI, 16MHz) as the processor clock. The following gives C codes to use HSI.

```
// Turn on HSI (16MHz)
RCC->CR |= RCC_CR_HSION;

// Wait until HSI is ready
while( (RCC->CR & RCC_CR_HSIRDY) == 0);

// Select HSI as system clock
RCC->CFGR &= ~RCC_CFGR_SW_HSI;
RCC->CFGR |= RCC_CFGR_SW_HSI;
while( (RCC->CFGR & RCC_CFGR_SWS)!=RCC_CFGR_SWS_HSI ); // Wait till HSI
```

### *Part 2: Interface with Ultrasonic distance sensor*

Part 2 involves capturing a square wave generated by an HC-SR04 Ultrasonic distance sensor. We will use PB.10 to trigger the ultrasonic sensor and PB.6 to capture the echo output of the sensor.

	Pin	Available Alternative Functions
Trigger	PB.10	I2C2_SCL/USART3_TX/ <b>TIM2_CH3</b> /LCD_SEG10
Echo	PB 6	I2C1_SCL/ <b>TIM4_CH1</b> /USART1_TX

Documentation for this device can be found online, what follows is a brief summary.

- The sensor is powered by 5V. Connect the Vcc line to EXT\_5V on the STM32L board, and GND to a ground connection on the board.
- While the board runs at 5V, it can be triggered by a 3.3V pulse. Its output is 5V, but many of the inputs on the STM32L board are five-volt tolerant and can handle a 5V input.
- As described in the documentation, to activate the sensor send a high pulse of at least 10us to the Trig input. An ultrasonic burst of 40kHz will be emitted, and then the device will return a square wave proportional to the distance to the nearest object.
- The return will be on the ECHO pin, a square wave ranging from 150us to 25ms (38ms if nothing is in range). To convert this value to inches, divide the time in us (microseconds) by 148.

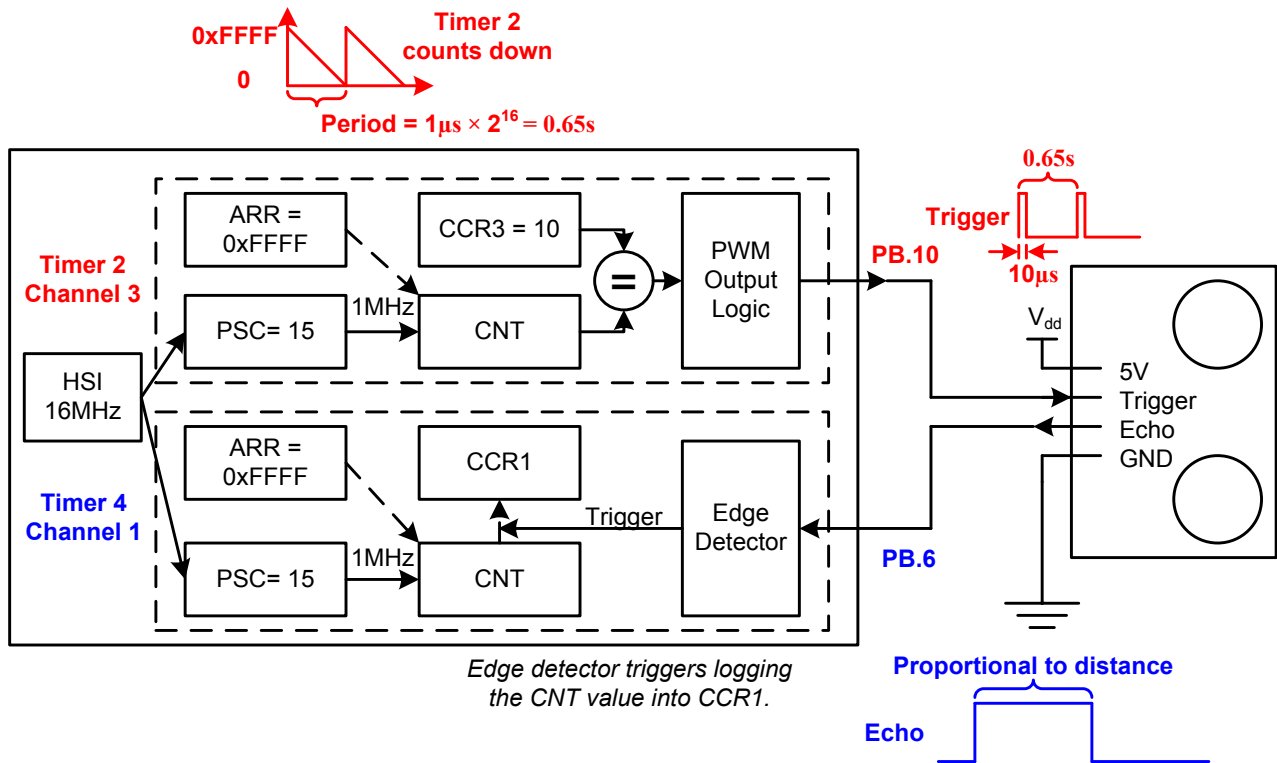


Figure 1. Connection and configuration diagram for interfacing ultrasonic distance sensor

**ECE 271 Microcomputer Architecture and Applications****Lab 7 Timer Input Capture  
Pre-Lab Assignment*****Part 1: Measure the period of 1 Hz square signal***

Pre-Lab Assignment:

$$f_{\text{Timer clock}} = \frac{f_{\text{HSI}}}{1 + \text{PSC}}$$

For Timer 4, the clock pre-scalar register (PSC), the compare/capture register (CCR), and the timer free-run counter has 16 bits. Thus their value is limited to  $2^{16} - 1 = 65535$ .

**Write your answers to Readme.md and submit it to Gitlab server:**

Timer Input Clock Frequency: 16MHz

**1. Select your timer prescaler (PSC)**

PSC = \_\_\_\_\_

- 2. If both rising and fall edges of an external signal are selected as active transition, the time interval between two consecutive timer interrupts is 0.5 second. What should be the difference of CCR values in two consecutive timer interrupts? Show your calculation.**
- 3. For the prescaler you have selected, how long (in terms of seconds) after reset a counter overflow or underflow event occurs? Show your calculation.**
- 4. If a counter overflow/underflow occurs, the difference of two CCR readings may not correctly measure the time interval. Explain why and how to solve it? (Hints: In the timer interrupt service routine, check the UIF flag)**

**Part 2: Interface with Ultrasonic distance sensor**

As shown in Figure 1, PSC is selected as 15.

$$f_{Timer\ Clock} = \frac{f_{HSI}}{1 + PSC} = \frac{16MHz}{1 + 15} = 1MHz$$

For Timer 4, the clock pre-scalar register (PSC), the compare/capture register (CCR), and the timer free-run counter has 16 bits. Thus their value is limited to  $2^{16} - 1 = 65535$ .

**Write your answers to Readme.md and submit it to Gitlab server:**

Timer Input Clock Frequency: 16MHz

Timer Prescaler: 15

- 1. What is time resolution (minimum time unit) of the input capture function?**
- 2. If the echo signal is high for time duration from 150us to 38ms, what is the min and max value difference of CCR values in two consecutive interrupts?**
- 3. For the prescaler you have selected, how long (in terms of seconds) after reset a counter overflow or underflow event occurs? Show your calculation.**

## ECE 271 Microcomputer Architecture and Applications

### Lab 7 Timer Input Capture In-Lab Demo

#### *Part 1: Measure the period of 1 Hz square signal*

- Basic requirement: In the debug environment, show the period you have measured. Record the accuracy for the post-lab assignment.
- Something cool: for example, show time measurements on LCD.

#### *Part 2: Interface with Ultrasonic distance sensor*

- Basic requirement: In the debug environment, show the distance you have measured. Record the accuracy for the post-lab assignment.
- Something cool: for example, show distance measurements on LCD.

## ECE 271 Microcomputer Architecture and Applications

### Lab 7 Timer Input Capture Post-Lab Assignment

**Write your answers to Readme.md and submit it to Gitlab server:**

1. Does the timer counter count up or down in your lab? If counting up, how did you handle the counter overflow? (If counting down, how did you handle the counter underflow?)
2. What is the accuracy when measuring the period of 1Hz square wave?
3. What is the accuracy of the distance you have measured?
4. What is the most challenge issue you had in this lab?
5. Do you have any suggestions or comments of this lab?
6. Do you have any comments of the textbook?