ECE271: Microcomputer Architecture and Applications — University of Maine

Prelab for Lab #9: Input Capture and Distance Sensor

Week of 4 April 2022

Part A – Textbook Readings / Videos

1. Read Textbook `Chapter 15.4` to review input capture.

Part B – Prelab assignment

In this lab we will set up the TIM4 timer for input capture on pin PB6 and the TIM1 timer to generate a trigger pulse on pin PE11.

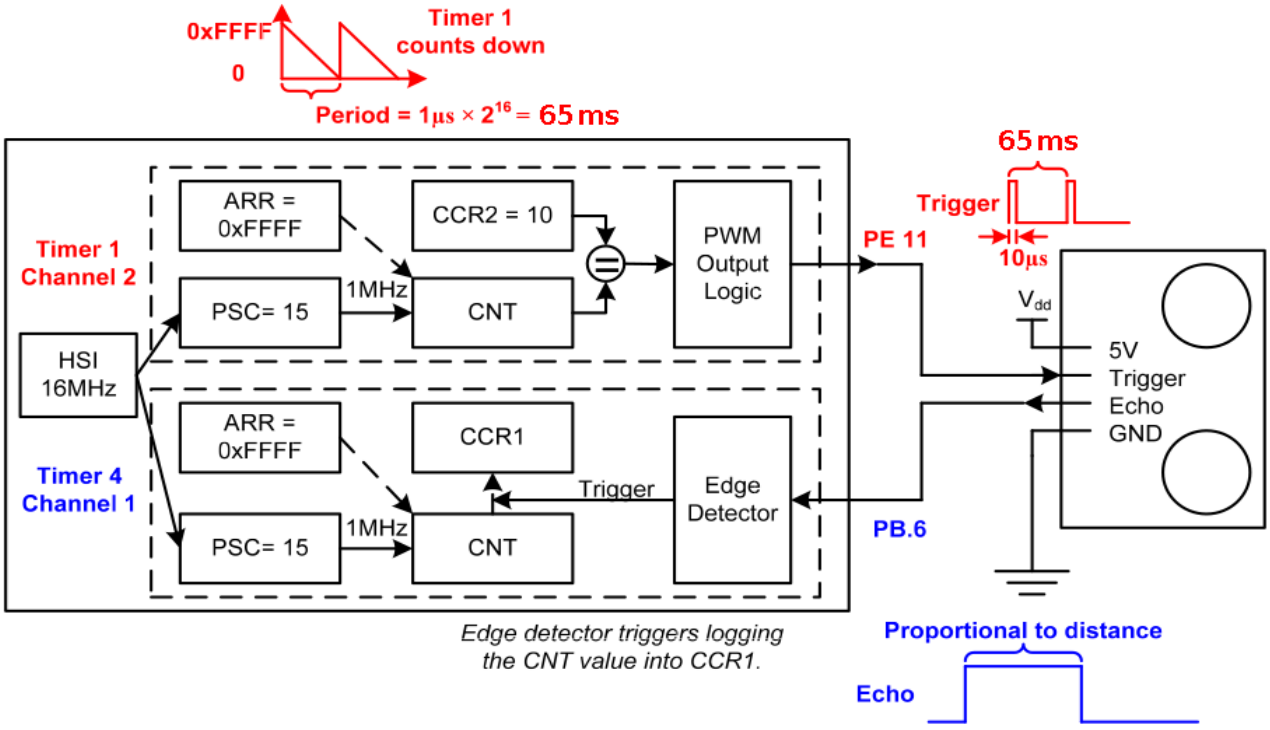

Figure 1: Timer setup for lab.

1. Doing input capture with timer TIM4

For this lab we will be using the 16MHz HSI timer.
The formula is $f_{timer\_clock} = \frac{f_{HSI}}{1+PSC}$

$1MHz = \frac{16MHz}{1+PSC}$ $\quad \frac{1}{1\mu} = 1MHz$ $\quad PSC = 15$

1. What value should you use for the PSC to get a counting increment of $1\mu s$? (remember, $period = \frac{1}{f}$)
2. For the prescaler value in PSC, with the 16-bit TIM4 timer, how long (in terms of seconds) does it take for a counter overflow or underflow event to occur?

$\frac{1\mu s}{65535} = 65.535\,ms$

2. Settings needed to have pin PE11 connected to timer TIM1 Channel 2

You will need to set the following fields. Write the values to mask/set. If no mask is needed you can let that blank. You can use pre-defined names for the bits rather than raw hex values.

- Set `GPIOE->MODER` for pin 11 to be "alternate" mode.
  MASK MODER=$\sim$(GPIO_MODER_MODER11)
  VALUE MODER=GPIO_MODER_MODER11_1

- Set `GPIOE->AFR[0]` and `GPIOE->AFR[1]` for alternate function of Pin 11 to be `TIM1_CH2`. You can look in Appendix I of the book to see which one this is. This should be in one of the document pdfs too but I wasn't able to find which one.
  MASK AFR[0]=
  VALUE AFR[0]=
  MASK AFR[1]=0xffff fff0
  VALUE AFR[1]=0x0000 0001

- Set `GPIOE->OTYPER` for Pin 11 to be push-pull
  MASK OTYPER=$\sim$(GPIO_OTYPER_OT_11)
  VALUE OTYPER=_____

- Set `GPIOE->PUPDR` for Pin 11 for no pull-up/pull-down
  MASK PUPDR=$\sim$(GPIO_PUPDR_PUPDR11)
  VALUE PUPDR=NA

3. Settings needed to generate a 10μs pulse on PE11

You will need to set the following fields. Write the values to mask/set. If no mask is needed you can let that blank. You can use pre-defined names for the bits rather than raw hex values.

- Set `TIM1->CR1` for the counting direction to be up.
  MASK CR1=$\sim$(TIM_CR1_DIR)
  VALUE CR1=_____

- We will use a 16MHz HSI clock for this lab. Set the prescalar `TIM1->PSC` to count at 1MHz.
  MASK PSC=_____
  VALUE PSC=_____ TIM1->PSC = 15 $f_{timer\_clock} = \frac{f_{HSI}}{1+PSC}$

- Set the `TIM1->ARR` register to give the maximum possible period
  MASK ARR=_____
  VALUE ARR=_____ PWM Period = $\frac{(ARR+1)}{1MHz}$ TIM1->ARR = 65535

- Clear the `OC2M` field in the `TIM1->CCMR1` register and select PWM Mode 1 (`OC2M` = 110):
  MASK CCMR1= ~(TIM_CCMR1_OC2M)
  VALUE CCMR1= TIM_CCMR1_OC2M_1 & TIM_CCMR1_OC2M_2

- Enable the Output 2 preload enable in `TIM1->CCMR1` (`OC2PE`): MASK CCMR1=_____
  VALUE CCMR1= TIM_CCMR1_OC2PE ~(TIM_CCMR1_OC2PE)

- Select the output polarity by clearing the `CC2P` field in the `TIM1->CCER` register:
  MASK CCER= ~(TIM_CCER_CC2P)
  VALUE CCER= NA

- Enable output of Channel 2 (CH2) by setting the `CC2E` bit in `TIM1->CCER`:
  MASK CCER= ~(TIM_CCER_CC2E)
  VALUE CCER= TIM_CCER_CC2E

- Set the main output enable (MOE) in `TIM1->BDTR` :
  MASK BDTR= ~(TIM_BDTR_MOE)
  VALUE BDTR= TIM_BDTR_MOE

- Set the output compare register for channel 2 `TIM1->CCR2` to be on for $10\mu s$ each period: MASK CCR2=_____
  VALUE CCR2=_____ TIM1 → CCR2 = 10