

4.) b.)

```
void Timer0A_Init(void){
    SYSCTL->RCGCTIMER |= 0x01; // activate timer0
    SYSCTL->RCGCGPIO |= 0x02; // activate port B
    GPIOB->DIR &= ~0x40; // make PB6 input
    GPIOB->AFSEL |= 0x40; // enable alt funct on PB6
    GPIOB->DEN |= 0x40; // configure PB6 as T0CCP0
    GPIOB->PCTL = (GPIOB->PCTL_R&0xF0FFFFFF)+0x07000000;
    TIMER0->CTL &= ~0x00000001; // disable timer0A during setup
    TIMER0->CFG = 0x00000004; // configure for 16-bit capture mode
    TIMER0->TAMR = 0x00000007; // configure for rising edge event
    TIMER0->CTL &= ~0x0000000C; // rising edge
    TIMER0->TAILR = 0x0000FFFF; // start value
    TIMER0->TAPR = 0xFF; // activate prescale, creating 24-bit
    TIMER0->IMR |= 0x00000004; // enable capture match interrupt
    TIMER0->ICR = 0x00000004; // clear timer0A capture match flag
    TIMER0->CTL |= 0x00000001; // timer0A 24-b, +edge, interrupts
    NVIC->IP[19] = 2 << 5; // 19: IRQ number; 2: priority; 5: fixed-shift 5 position
    NVIC->ISER[0] = 1<<19; // enable interrupt 19 in NVIC
    __enable_irq(); /* global enable IRQs */
}
```

```
void Timer1A_Init(void){
    SYSCTL->RCGCTIMER |= 0x02; // activate timer1
    SYSCTL->RCGCGPIO |= 0x02; // activate port B
    GPIOB->DIR &= ~0x10; // make PB4 input
    GPIOB->AFSEL |= 0x10; // enable alt funct on PB4
    GPIOB->DEN |= 0x10; // configure PB4 as T0CCP0
    GPIOB->PCTL = (GPIOB->PCTL_R&0xF0FFFFFF)+0x07000000;
    TIMER1->CTL &= ~0x00000001; // disable timer0A during setup
    TIMER1->CFG = 0x00000004; // configure for 16-bit capture mode
    TIMER1->TAMR = 0x00000007; // configure for rising edge event
    TIMER1->CTL &= ~0x0000000C; // rising edge
    TIMER1->TAILR = 0x0000FFFF; // start value
    TIMER1->TAPR = 0xFF; // activate prescale, creating 24-bit
    TIMER1->IMR |= 0x00000004; // enable capture match interrupt
    TIMER1->ICR = 0x00000004; // clear timer1A capture match flag
    TIMER1->CTL |= 0x00000001; // timer1A 24-b, +edge, interrupts
    NVIC->IP[19] = 2 << 5; // 19: IRQ number; 2: priority; 5: fixed-shift 5 position
    NVIC->ISER[0] = 1<<19; // enable interrupt 19 in NVIC
    __enable_irq(); /* global enable IRQs */
}
```