```
/*Pulse_init.h file
Function for creating a pulse train using interrupts
Uses Channel 0, and a 1Mhz Timer clock (_TAPR = 15)
Uses Timer0A to create pulse train on PF2
*/

#include "TM4C123GH6PM.h"
void pulse_init(void);
void TIMER0A_Handler (void);
void detect_init (void);
void timer1A_delaySec(void);

#define LOW   60
#define HIGH  15

void pulse_init(void){
  volatile int *NVIC_EN0 = (volatile int*) 0xE000E100;
  volatile int *NVIC_PRI4 = (volatile int*) 0xE000E410;
  SYSCTL->RCGCGPIO |= 0x20; // turn on bus clock for GPIOF
  __ASM("NOP");
  __ASM("NOP");
  __ASM("NOP");

  GPIOF->DIR      |= 0x04; //set PF2 as output
  GPIOF->AFSEL    &= (0xFFFFFFFB);  // Regular port function
  GPIOF->PCTL     &= 0xFFFFF0FF;  // No alternate function
  GPIOF->AMSEL    =0; //Disable analog
  GPIOF->DEN      |=0x04; // Enable port digital

  //GPIOF->DIR       |= 0x08; //set GREEN pin as a digital output pin
  //GPIOF->DEN       |= 0x08;  // Enable PF3 pin as a digital pin

  SYSCTL->RCGCTIMER |=0x01; // Start timer0
  __ASM("NOP");
  __ASM("NOP");
  __ASM("NOP");
  TIMER0->CTL     &=0xFFFFFFFE; //Disable timer during setup
  TIMER0->CFG     =0x04;  //Set 16 bit mode
  TIMER0->TAMR    =0x02; // set to periodic, count down
  TIMER0->TAILR   =LOW; //Set interval load as LOW
  TIMER0->TAPR    =15; // Divide the clock by 16 to get 1us
  TIMER0->IMR     =0x01; //Enable timeout intrrupt

  //Timer0A is interrupt 19
  //Interrupt 16-19 are handled by NVIC register PRI4
  //Interrupt 19 is controlled by bits 31:29 of PRI4
  *NVIC_PRI4 &=0x00FFFFFF; //Clear interrupt 19 priority
  *NVIC_PRI4 |=0x40000000; //Set interrupt 19 priority to 2

  //NVIC has to be neabled
  //Interrupts 0-31 are handled by NVIC register EN0
  //Interrupt 19 is controlled by bit 19
  *NVIC_EN0 |=0x00080000;

  //Enable timer
  TIMER0->CTL      |=0x03; // bit0 to enable and bit 1 to stall on debug
  return;
}

void TIMER0A_Handler (void){
  GPIOF->DATA  ^= 4;  //toggle PF2 pin

  if(TIMER0->TAILR==LOW)
  {
    TIMER0->TAILR=HIGH;
    TIMER0->ICR |=0x01;
  }
  else
    TIMER0->TAILR=LOW;

  return;
}
void detect_init (void){
    SYSCTL->RCGCTIMER |= (1<<2); /* enable clock to Timer Block 3 */
    SYSCTL->RCGCGPIO |= (1<<1); /* enable clock to PORTB */

    __ASM("NOP");
```

```
78        __ASM("NOP");
79        __ASM("NOP");
80
81        GPIOB->DIR |= 0xEF; /* set PB4 an input pin */
82        GPIOB->DEN |= 0x10; /* set PB4 a digital pin */
83        GPIOB->AFSEL |= 0x10; /* enable alternate function on PB4 */
84        GPIOB->PCTL |= 0x00070000; //to enable T1CCP0 on PB4
85
86        SYSCTL->RCGCTIMER |= 0x02; /* enable clock to Timer Block 1 */
87        TIMER1->CTL &=0xFFFFFFFE; /* disable TIMER1 during setup */
88        TIMER1->CFG |= 0x04; /* configure as 16-bit timer mode */
89
90        TIMER1->TAMR |= 0x07; /* down-counter, edge time, capture mode */
91
92        TIMER1->CTL |= 0x0C; //set bits 3:2 to 0x03
93        TIMER1->TAILR =12345;//max value is 65535,60000 is choosen not to fail the operation
94        TIMER1->TAPR =15;
95
96        TIMER1->CTL |=0x03;   // Enable timer
97
98    }
99    void timer1A_delaySec(void)
100   {
101       SYSCTL->RCGCTIMER |= 8; /* enable clock to Timer Block 3 */
102       TIMER3->CTL = 0; /* disable Timer before initialization */
103       TIMER3->CFG = 0x04; /* 16-bit option */
104       TIMER3->TAMR = 0x02; /* periodic mode and down-counter */
105       TIMER3->TAILR = (64000 -1)/10; /* TimerAinterval load value reg*/
106       TIMER3->TAPR = 250 -1; /* TimerAPrescaler16MHz/250=64000Hz */
107       TIMER3->ICR = 0x1; /* clear the TimerAtimeout flag */
108       TIMER3->CTL |= 0x01; /* enable Timer A after initialization */
109       while ((TIMER3->RIS & 0x1) == 0)
110       { }; /* wait for TimerAtimeout flag */
111   }
```