```c
/*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);

#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;
  else
    TIMER0->TAILR=LOW;
  TIMER0->ICR |=0x01;
  return;
}
int main()
{
  pulse_init();
  while(1){}

}
```