```c
  1    /*------------------------------------------------------------------------
  2     * Name:     FSK.c
  3     * Purpose: Functions to provide frequency shift keying for an input waveform,
  4     *          by setting 2 different DDS frequencies appropiately.
  5     * Note(s): Example code taken from STMicroElectronics Application Teams,
  6     *          TIM_PWM_Input eample project.
  7     *------------------------------------------------------------------------
  8     *
  9     *------------------------------------------------------------------------*/
 10
 11    #include "STM32F4xx.h"
 12    #include "main_2.h"
 13    #include "FSK.h"
 14    #include "DDS.h"
 15
 16    void FSK_Init(void)
 17    {
 18      TIM_ICInitTypeDef  TIM_ICInitStructure;
 19
 20      GPIO_InitTypeDef GPIO_InitStructure;
 21      NVIC_InitTypeDef NVIC_InitStructure;
 22
 23      /* TIM4 clock enable */
 24      RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4, ENABLE);
 25
 26      /* GPIOB clock enable */
 27      RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOB, ENABLE);
 28
 29      /* TIM4 chennel2 configuration : PB.07 */
 30      GPIO_InitStructure.GPIO_Pin   = GPIO_Pin_7;
 31      GPIO_InitStructure.GPIO_Mode  = GPIO_Mode_AF;
 32      GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
 33      GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
 34      GPIO_InitStructure.GPIO_PuPd  = GPIO_PuPd_UP ;
 35      GPIO_Init(GPIOB, &GPIO_InitStructure);
 36
 37      /* Connect TIM pin to AF2 */
 38      GPIO_PinAFConfig(GPIOB, GPIO_PinSource7, GPIO_AF_TIM4);
 39
 40      /* Enable the TIM4 global Interrupt */
 41      NVIC_InitStructure.NVIC_IRQChannel = TIM4_IRQn;
 42      NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
 43      NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
 44      NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
 45      NVIC_Init(&NVIC_InitStructure);
 46
 47      /* TIM4 configuration: PWM Input mode */
 48      TIM_ICInitStructure.TIM_Channel = TIM_Channel_2;
 49      TIM_ICInitStructure.TIM_ICPolarity = TIM_ICPolarity_BothEdge;
 50      TIM_ICInitStructure.TIM_ICSelection = TIM_ICSelection_DirectTI;
 51      TIM_ICInitStructure.TIM_ICPrescaler = TIM_ICPSC_DIV1;
 52      TIM_ICInitStructure.TIM_ICFilter = 0x0;
 53
 54      TIM_PWMIConfig(TIM4, &TIM_ICInitStructure);
 55
 56      /* Select the TIM4 Input Trigger: TI2FP2 */
 57      TIM_SelectInputTrigger(TIM4, TIM_TS_TI2FP2);
 58
 59      /* Select the slave Mode: Reset Mode */
 60      TIM_SelectSlaveMode(TIM4, TIM_SlaveMode_Reset);
 61      TIM_SelectMasterSlaveMode(TIM4,TIM_MasterSlaveMode_Enable);
 62
 63      /* TIM enable counter */
 64      TIM_Cmd(TIM4, ENABLE);
 65
 66      /* Enable the CC2 Interrupt Request */
 67      TIM_ITConfig(TIM4, TIM_IT_CC2, ENABLE);
 68    }
```