```
/*******************************************************************************
 * Name:    PWM.c
 * Description: STM32 Pluse Width Modulation
 * Version: V1.00
 * Authors: Li Pan
 *
 ******************************************************************************/
#include "stm32f10x.h"
#include "PWM.h"
#include "GPIO.h"
#include "CLOCK.h"

void PWM_INIT( void )
{
      //Enable the TIM1,PORT A, Alternate function clock
      RCC->APB2ENR |=RCC_APB2ENR_TIM1EN | RCC_APB2ENR_IOPAEN |
RCC_APB2ENR_AFIOEN;


      //Set PA8 as AFIO Push-Pull output
      GPIOA->CRH |= GPIO_CRH_CNF8_1 | GPIO_CRH_MODE8;
      GPIOA->CRH &= ~GPIO_CRH_CNF8_0;


      //Initialize TIM1 flags for PWM
      TIM1->CR1 |= TIM_CR1_CEN;    //Enable the timer
      TIM1->CR2 |= TIM_CR2_OIS1;  //Set idle states 'high'
      TIM1->EGR |= TIM_EGR_UG;          //Reset the counter when it has
completed counting


      TIM1->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1PE |
TIM_CCMR1_OC1FE;

      TIM1->CCER |= TIM_CCER_CC1E;

      //Set 10ms period, which is  frequency is 10KHz
      TIM1->PSC = 2399;

      //Initial PERIOD - frequency :  Since we make the Resolution is 1% which
is 0.1 ms
      TIM1->ARR = 100;

      //Initial PULSE WIDTH 50%
      TIM1->CCR1 = 50  ;


      //
      TIM1->BDTR |= TIM_BDTR_MOE | TIM_BDTR_OSSI;

      //enable the counter again
      TIM1->CR1 |= TIM_CR1_ARPE | TIM_CR1_CEN;
```

```
}


/*
 Changes the duty cycle.
 Valid values: 1 to 99
*/
void SetDutyCycle( int value )
{
      //int value  = read_SW();
      //Set new duty cycle value
      TIM1->CCR1 = value;

      //Transfer new value to register
      TIM1->EGR |= TIM_EGR_UG;

      //delay(600000);
}

void PWM2_INIT( void )
{
      //Enable the TIM1,PORT A, Alternate function clock
      RCC->APB2ENR |=RCC_APB2ENR_TIM1EN| RCC_APB2ENR_IOPAEN |
RCC_APB2ENR_AFIOEN;


      //Set PA9 as AFIO Push-Pull output
      GPIOA->CRH |= GPIO_CRH_MODE9_0 | GPIO_CRH_MODE9_1;
      GPIOA->CRH |= GPIO_CRH_CNF9_1;
      GPIOA->CRH &= ~GPIO_CRH_CNF9_0;


      //Initialize TIM1 flags for PWM
      TIM1->CR1 |= TIM_CR1_CEN;    //Enable the timer
      TIM1->CR2 |= TIM_CR2_OIS2;  //Set idle states 'high'
      TIM1->EGR |= TIM_EGR_UG;           //Reset the counter when it has
completed counting


      TIM1->CCMR1 |= TIM_CCMR1_OC2M_2 | TIM_CCMR1_OC2M_1 | TIM_CCMR1_OC2PE |
TIM_CCMR1_OC2FE;

      /////////////////////////////////
      TIM1->CCER |= TIM_CCER_CC2E;

      //Set 10ms period, which is  frequency is 10KHz
      TIM1->PSC = 2399;

      //Initial PERIOD - frequency :  Since we make the Resolution is 1% which
is 0.1 ms
      TIM1->ARR = 100;

      //Initial PULSE WIDTH 50%
      ////////////////////////////////
      TIM1->CCR2 = 50  ;


      //
      TIM1->BDTR |= TIM_BDTR_MOE | TIM_BDTR_OSSI;

      //enable the counter again
      TIM1->CR1 |= TIM_CR1_ARPE | TIM_CR1_CEN;


}


/*
 Changes the duty cycle.
 Valid values: 1 to 99
*/
void SetDutyCycle2( int value )
{
      //int value  = read_SW();
      //Set new duty cycle value
      TIM1->CCR2 = value;

      //Transfer new value to register
      TIM1->EGR |= TIM_EGR_UG;

      //delay(600000);
}
```