```c
#include "PWM.h"

void initPWM()
{
 TIM1->CR1 |= TIM_CR1_CEN; // Enable Timer1
 TIM1->CR2 |= TIM_CR2_OIS1; // Output Idle State for Channel 1 OC1=1 when MOE=0
 TIM1->EGR |= TIM_EGR_UG; // Reinitialize the counter
 TIM1->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1 | TIM_CCMR1_OC1PE |
 TIM_CCMR1_OC1FE; //PWM mode 1, Preload Enable, Fast Enable
 //TIM1->CCMR2 not used for this application
 TIM1->CCER |= TIM_CCER_CC1E; //Enable CH1 output on PA8
 TIM1->PSC = 0x095F; //Divide 24 MHz by 2400 , PSC_CLK = 10000 Hz, 1 count = 0.1 ms
 TIM1->ARR = 100; // 100 counts = 10 ms
 TIM1->CCR1 = 10; // 10 counts = 1 ms = 10% duty cycle
 TIM1->BDTR |= TIM_BDTR_MOE | TIM_BDTR_OSSI; //Main Output Enable, Force Idle Level First
 TIM1->CR1 |= TIM_CR1_ARPE | TIM_CR1_CEN; // Enable Timer1
}

void initRegisterForPWM()
{
  //Enable Timer1 AFIOenabled and IOA enabled
  RCC->APB2ENR |= RCC_APB2ENR_IOPAEN | RCC_APB2ENR_AFIOEN | RCC_APB2ENR_TIM1EN;

  //PA8 alternate function output 50Mhz
  GPIOA->CRH |= GPIO_CRH_MODE8 | GPIO_CRH_CNF8_1;
  GPIOA->CRH &= ~GPIO_CRH_CNF8_0;

  //PA1 analog voltage to change duty cycle
  GPIOA->CRL |= GPIO_CRL_MODE1;
  GPIOA->CRL &= ~GPIO_CRL_CNF1;
}

void changeDutyCycle(uint32_t ADCVoltage)
{
  //ADCVoltage max is 0xFFF
  //ADCVoltage/0xFFF goes from 0 to 1
  //(ADCVoltage * 100) / 0xFFF goes from 0 to 100
  TIM1->CCR1 = (ADCVoltage * 100) / 0xFFF;
  //Need to update register
  TIM1->EGR |= TIM_EGR_UG;
}
```