```
#include "servo.h"

// configuring the cpower perhiperhials
void clockInit(void){
    RCC->CFGR = RCC_CFGR_SW_HSE;
    RCC->CR &= ~RCC_CR_PLLON;
    RCC->CFGR = 0x00050000;// 24 MHz
    RCC->CR |= RCC_CR_PLLON;
    RCC->CFGR |= RCC_CFGR_SW_PLL;// 0x00000002;
    while (((RCC->CFGR) & (RCC_CFGR_SW_PLL | RCC_CFGR_SWS_PLL)) == 0);
    RCC->APB2ENR |= RCC_APB2ENR_IOPAEN |RCC_APB2ENR_IOPBEN |
RCC_APB2ENR_AFIOEN | RCC_APB2ENR_ADC1EN |RCC_APB2ENR_IOPCEN ;
}

void tim3_IO_init(void){
      RCC->APB1ENR |= RCC_APB1ENR_TIM3EN | RCC_APB2ENR_IOPAEN |
RCC_APB2ENR_IOPBEN | RCC_APB2ENR_IOPCEN;
      GPIOA->CRL |= GPIO_CRL_CNF6_1 | GPIO_CRL_MODE6;
      GPIOA->CRL &= ~GPIO_CRL_CNF6_0;
}

void tim3GPIOSetup ()
{
      TIM3 ->CR1 |= TIM_CR1_CEN;
      TIM3 -> EGR |= TIM_EGR_UG;

      //PWM Mode 1
      TIM3->CCMR1 |= TIM_CCMR1_OC1M_2 | TIM_CCMR1_OC1M_1;

      //Preload Enable, Fast Enable
      TIM3->CCMR1 |= TIM_CCMR1_OC1PE | TIM_CCMR1_OC1FE;

      //Enable CH1
      TIM3->CCER |= TIM_CCER_CC1E;

      // Divide 24 MHz by 2400 (PSC+1), PSC_CLK= 10000 Hz, 1 count = 0.1 ms
      TIM3->PSC = 0xEF;

      // 100 counts = 10 ms or 100 Hz
      TIM3->ARR = 2000;

      // 50 counts = 5 ms = 50% duty cycle
      TIM3->CCR1 = 150;

      // Enable Timer3
      TIM3->CR1 |= TIM_CR1_ARPE | TIM_CR1_CEN;
}
float converter(uint16_t test)
{
            float result;
            result = test * 1.11111111111;
            result = result + 50;
            return result;
}
```

```
void updateservo(uint16_t test)
{

           TIM3->CCR1 = converter(test);
    TIM3 -> EGR |= TIM_EGR_UG;
}
void moveservo(int angle )
{
        int time1 = 3400000;
        int time2 = 2000000;
        updateservo(angle);
                delay(time2);
}
```