**임베디드시스템 설계 및 실험 보고서**

**[002분반 - 2조 - 11주차]**

****

|  |  |
| --- | --- |
| **조원** | 202055531 김후겸  202055584 이태경  202155540 김채현  202255535 김진우 |
| **실험날짜** | 2024-11-13 |

**1. 실험 주제**

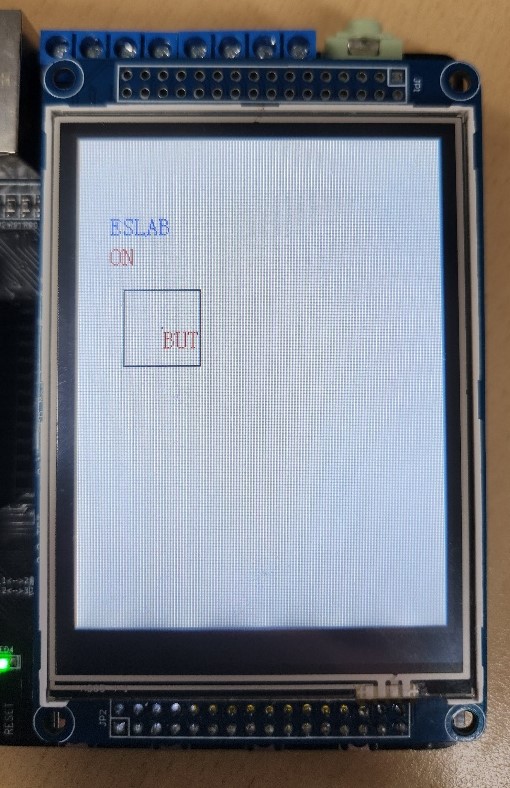
- Timer와 PWM

**2. 실험 목적**

* Timer의 종류와 특징의 이해
* PWM 신호의 이해
* 분주 계산 방법

**3. 세부 실험 목적**

1. TFT LCD에 team 이름, LED토글 ON/OFF 상태, LED ON/OFF 버튼 생성
2. LED ON 버튼 터치 시 TIM2 interrupt, TIM3 PWM을 활용하여 LED 2개와 서보모터 제어
3. LED OFF 버튼 터치 시 LED Toggle 동작 해제 및 서보모터 동작 반전

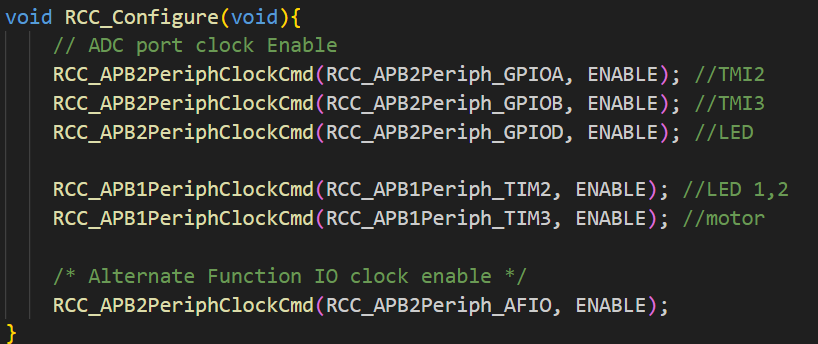


**4. 실험 장비**

* STM32F107VCT6
* TFT-LCD

**5. 실험 과정**

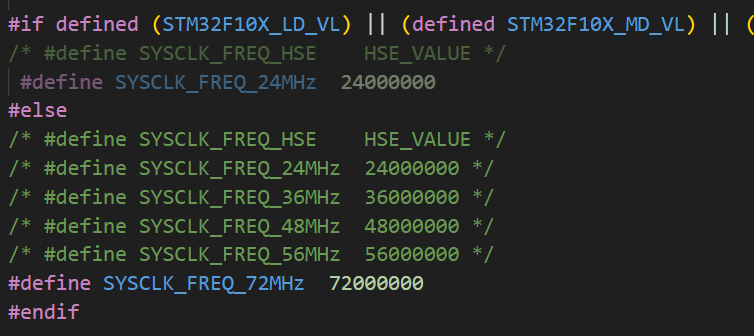
1. RCC & GPIO Configuration



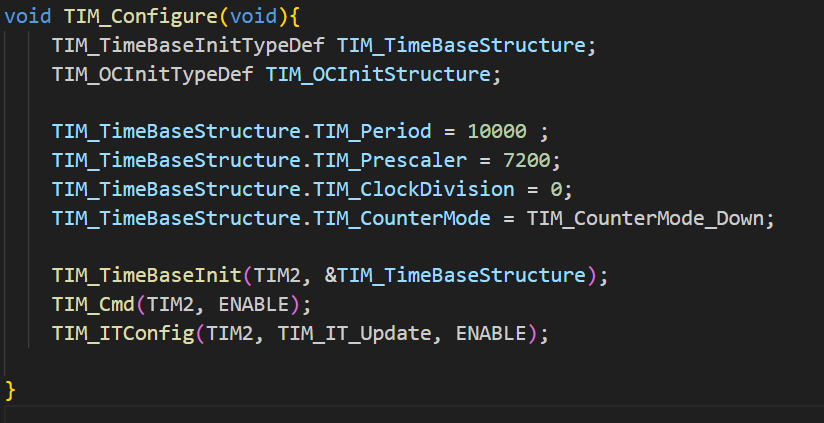


* LED 제어를 위한 타이머로 TIM2를 사용한다.
* 서보모터 제어를 위한 타이머로 TIM3를 사용한다.
* 타이머를 위한 GPIOA, GPIOB를 enable한다.
* LED 사용을 위해 GPIOD를 enable한다.
* 타이머는 APB1에 속하므로 APB1 주변장치로 타이머를 Enable한다.

2. Timer 설정



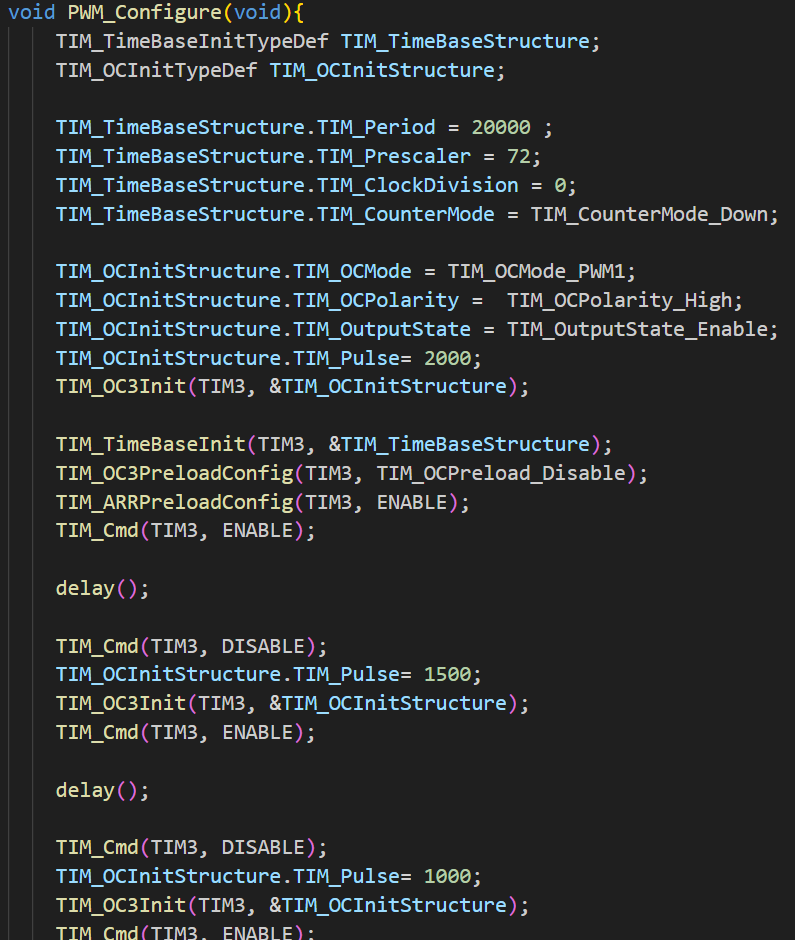
system\_stm32f10x.c파일을 보면 우리가 사용하는 주파수가 무엇인지 알 수 있다. 위 사진을 보면 72MHz를 사용한다.



타이머의 클락을 우리가 사용하기 쉬운 값으로 맞춰주기 위해 분주를 계산한다. 시스템 클락이 72MHz이므로 Period를 10000, Prescaler = 7200으로 분주하여 1초로 설정한다. 이렇게 하면 타이머는 1초마다 한번씩 cnt값을 변화시킬 수 있다.

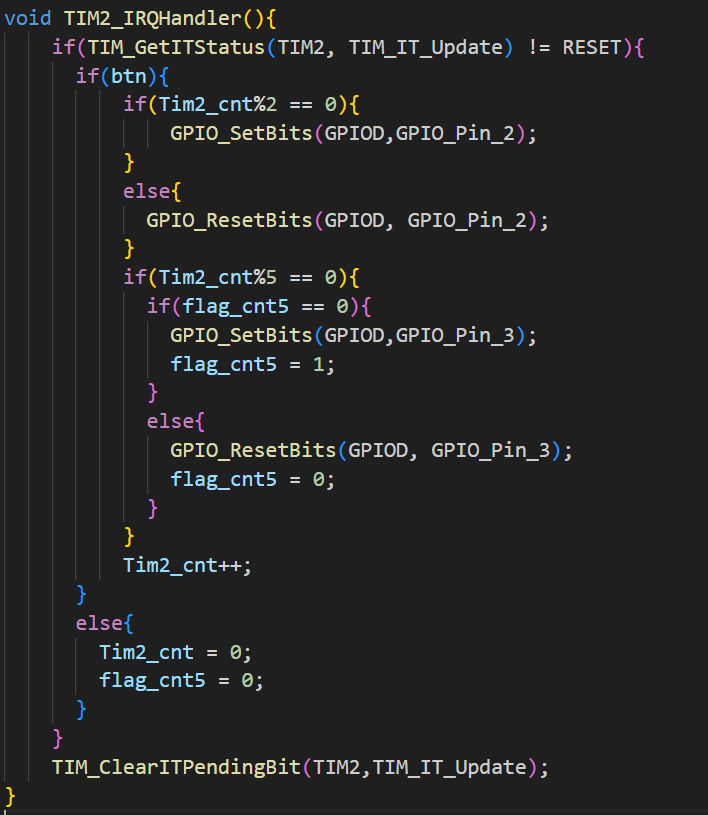
설정이 끝났으면 Init, cnt를 통해 타이머를 활성화 시키고 ITConfig를 통해 TIM2 인터럽트를 활성화시킨다.

3. PWM 설정



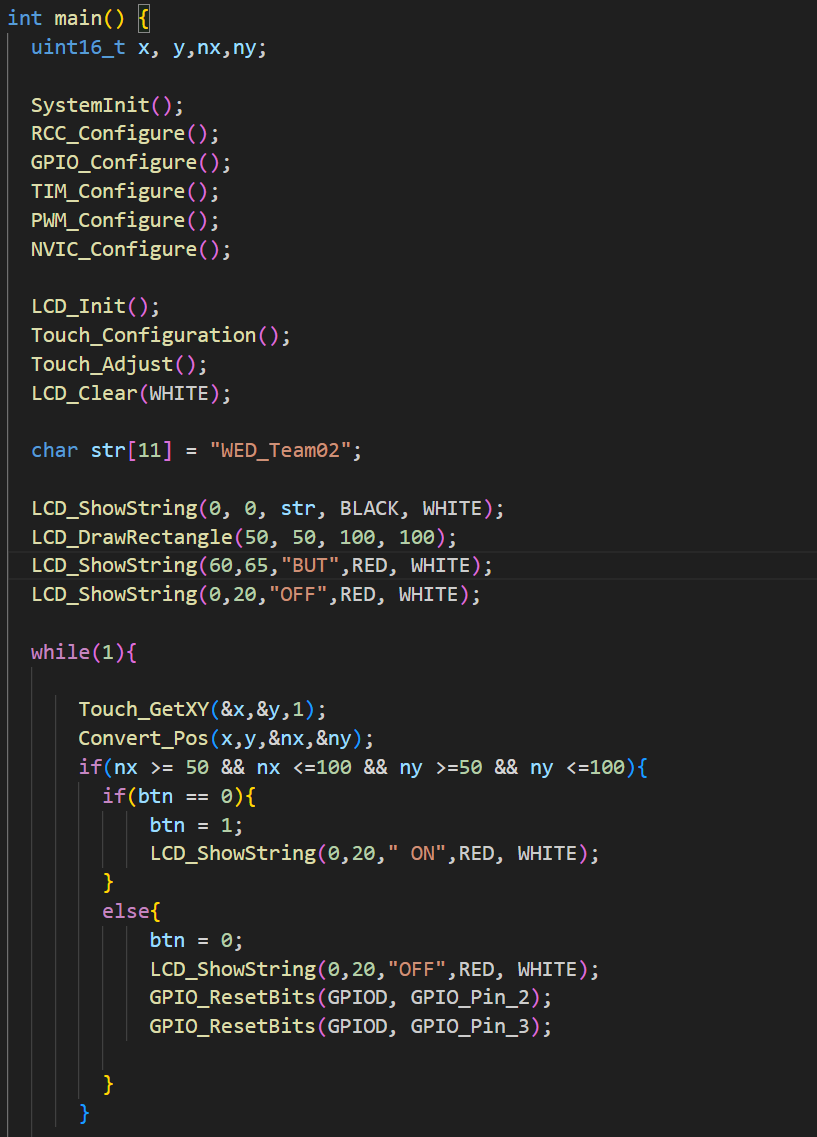
* PWM은 ON과 OFF 시간의 비율을 변화시켜 전체적인 평균값을 조절해 모터의 속도를 제어할 수 있다. ARR에 의해 주파수가 결정되고, CRR에 의해 duty cycle이 결정된다.
* ARR은 Auto-reload register로 상한선을 지정하는 레지스터이다. ARR이 100이면 100까지 카운터가 진행되고 0으로 돌아가서 자동 반복한다.
* CRR은 x번째 타이머의 y번째 채널 출력을 조절할 수 있는 레지스터이다. CCR 값이 ARR보다 큰 경우 또는 0인 경우에는 PWM 기능이 동작하지 않는다.
* CNT는 타이머의 카운터 값이 저장되어 있다. 설정된 주기에 맞게 1씩 감소하다 0을 만나면 인터럽트를 발생시킨다.
* Ouput Compare는 TIMx\_CNT의 값과 TIMx\_CCRx 레지스터 값이 일치할 때 Output Pin 상태를 제어할 수 있는 모드이다. TIM\_OCInitTypeDef 구조체로 설정한다.
* PWM1 모드는 CNT가 CCR 보다 작을 경우 Low, CNT가 CCR보다 큰 경우에 High 상태가 된다.
* Polarity는 출력의 극성(Low, High)를 결정한다.
* OutputState는 Output 채널의 enable/disable 설정을 담당한다.
* Pulse는 CCR 레지스터의 값으로 CNT와 비교하여 일치하면 인터럽트가 발생한다. 즉, Output이 있는 경우에 해당 Output 상태를 변경 시킨다. 우리는 2us, 1.5us, 1us로 설정하여 각각 90, 0, -90의 회전각으로 딜레이를 두고 서보모터를 작동시켰다.
* OC3Preload 설정에서 OC3의 전처리를 disable 하였다.
* ARRPreload 설정에서 ARR의 전처리를 enable하여 실행시 ARR을 우선적으로 load하게 하였다.
* Cmd를 통해 TIM3 블록을 구동시켰다.
* 이 후, 서보모터 동작을 위해 이미 작동하던 TIM3를 중단시키고 Pulse값을 바꾼 후 서보모터가 동하게 하였다.

4. TIM2\_IRQHandler 설정



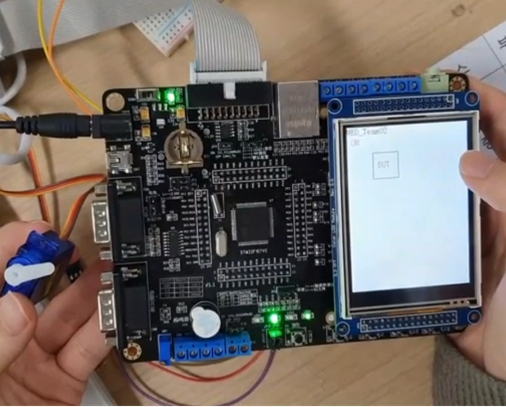
* LCD 버튼이 눌렸을 경우 1초마다 LED를 토글 시키기 위해서 Tim2 cnt를 2로 Modulation 연산을 하여 0,2,4... 초마다 LED1이 켜지고 1,3,5 초마다 LED1이 꺼짐으로 toggle동작을 하도록 하였다.
* LCD 버튼이 눌렸을 경우 5초마다 LED를 toggle 시키기 위해서 Tim2\_cnt를 5로 Modulation 연산을 하여 0,5,10,15...초마다 flag\_cnt5 값이0이면5초동안LED2를키고, flag\_cnt 값이 1이면 5초동안 LED를 끈다. 그다음 onoff동작을 위해 다시if문에들어오면flag값을반전시킨다.
* 위의 동작들을 한 번 진행하면, Tim2\_cnt값을1증가시킨다.
* 만약 버튼이 눌리지 않았을 때는 모든LED를off하기위해서Tim2\_cnt 값과flag\_cnt5 값을 모두 0으로 설정한다

5. main



* 지난주 실습에서 진행한 LCD를 활용하여 LCD에 표시될 문자들을 설정하였다.(팀명, ON/OFF, 버튼)
* 버튼이 눌러졌을 때, 전역변수인 bnt 값을 1로 바꾸어 핸들러가 이를 감지하도록 하였다. 또한 on,off 상태를 on으로 바꿔준다.
* 버튼 이외의 영역이 눌러졌을 때는 아무 동작도 하지 않고, OFF를 표시한다.

**6. 실험 결과**

****

**7. 분석 및 결론**

이번 실험에서는PWM 신호를 이용한 서보 모터제어와 타이머 인터럽트를 활용한 LED 제어를 배웠다. 서보 모터의 각도를 조정하기 위해 PWM 신호의 듀티 사이클을 조절하는 과정을 실습하면서, 하드웨어 장치가 어떻게 신호를 인식하고 동작하는지에 대해 이해할 수 있었다. 또한, 타이머 인터럽트를 활용하여 LED의 점멸 주기를 설정함으로써, 타이머가 보드 내에서 어떤 방식으로 동작하며, 이를 통해 정확한 타이밍으로 장치를 제어할 수 있다는 것을 확인하였다.