

Busan Software Meister High School

MICROPROCESSOR

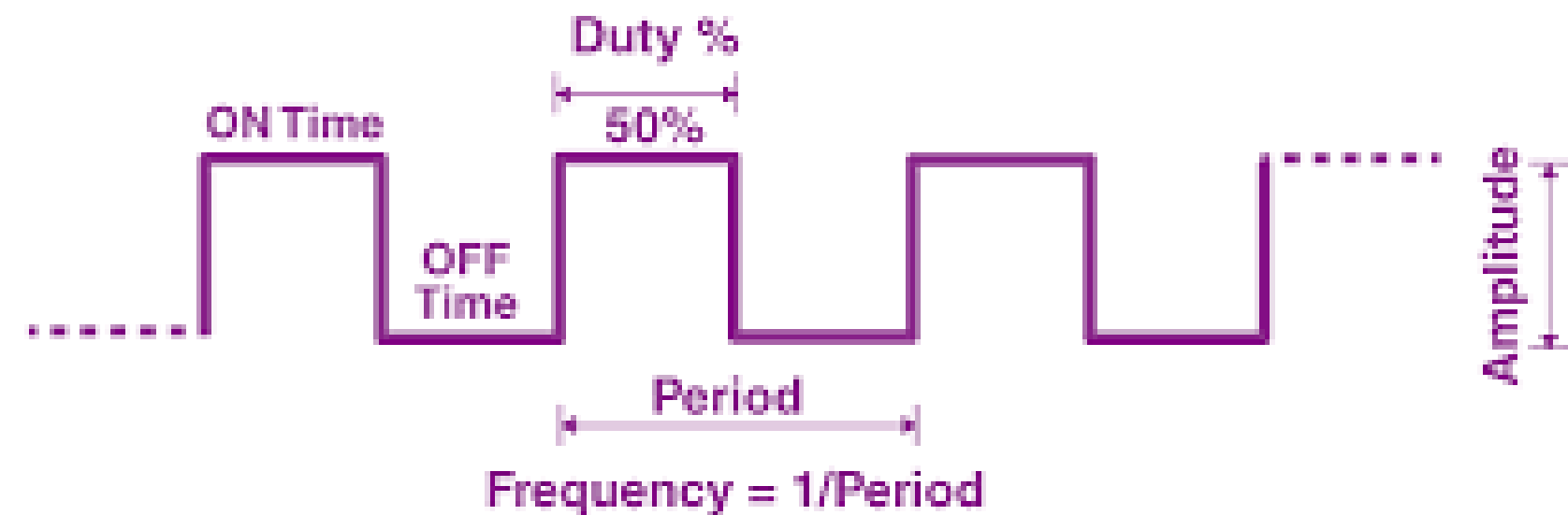
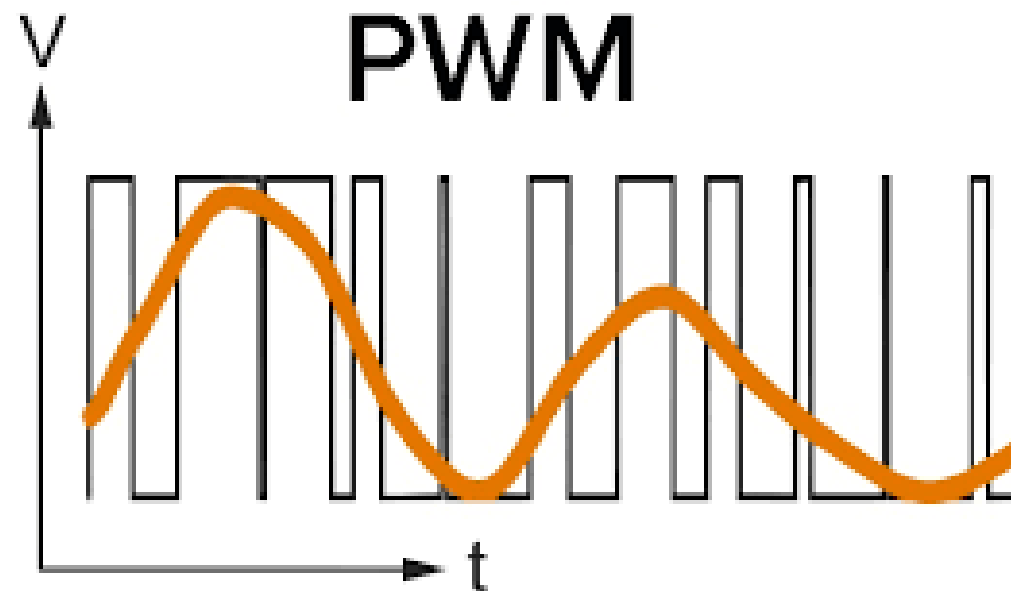
2309 양유빈

20231019 마이크로프로세서

PWM

Pulse Width Modulation

- PWM(Pulse Width Modulation)은 디지털 신호를 사용하여 아날로그 신호를 효과적으로 모방하는 기술(펄스 폭 변환)
 - 주기적으로 HIGH상태와 LOW상태를 반복하는 디지털 신호 생성
 - 일정한 주기 내에서 ON 상태와 OFF 상태를 조절하여 아날로그 값에 대한 근사값 생성
 - 긴 ON 상태는 높은 아날로그 값, 짧은 ON 상태는 낮은 아날로그 값에 해당
 - PWM 신호의 주기가 빠를수록 아날로그 값을 정밀하게 근사화 가능
 - **듀티비**(Duty Cycle) : HIGH인 신호와 LOW인 신호의 비율
- 반복적으로 HIGH – LOW – HIGH – LOW … 신호를 생성해야 하는 경우 타이머 로직을 이용하면 편리(펄스의 주기와 펄스의 듀티사이클 조정 기능 내장)



PWM 관련 HAL 라이브러리

PWM related HAL library

형태	HAL_StatusTypeDef HAL_TIM_PWM_Start(TIM_HandleTypeDef *htim, uint32_t Channel)
설명	지정된 타이머의 지정된 채널에 대한 PWM 동작을 시작
파라미터	htim : PWM 동작을 시작할 타이머 구조체 포인터 값(&htim1(timer1) ~ &htim3(timer3)) Channel : 타이머 내에서 지정된 채널(TIM_Channel_1, TIM_Channel_2, TIM_Channel_3, TIM_Channel_4)
실행결과	실행 결과(HAL_OK, HAL_ERROR, HAL_BUSY, HAL_TIMEOUT)
예시	Timer1의 Channel1에 대한 PWM 동작을 시작하고 싶을 때의 코딩: HAL_TIM_PWM_Start(&htim1, TIM_Channel_1);

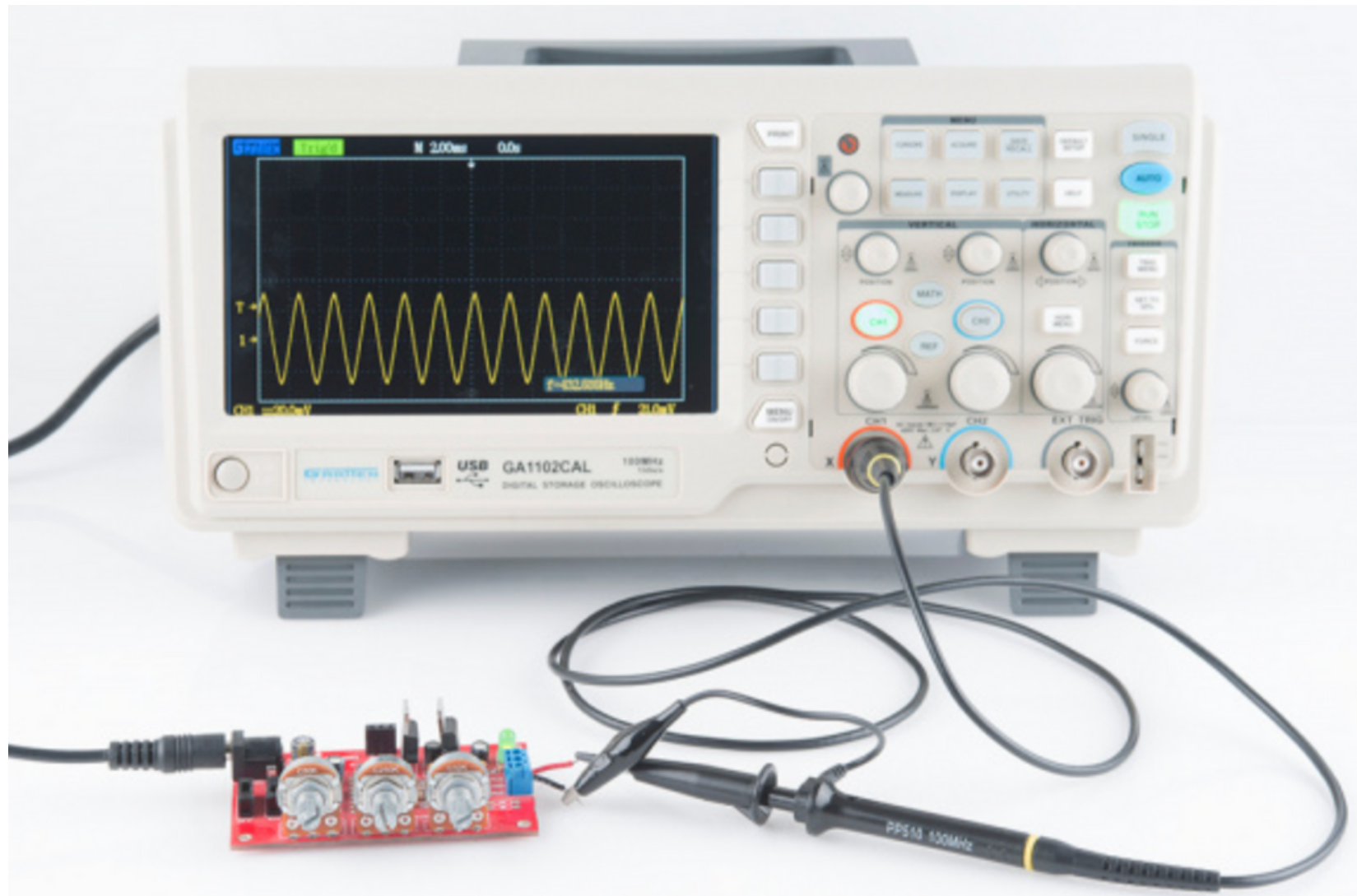
PWM 관련 HAL 라이브러리

PWM related HAL library

형태	HAL_StatusTypeDef HAL_TIM_PWM_Stop(TIM_HandleTypeDef *htim, uint32_t Channel)
설명	지정된 타이머의 지정된 채널에 대한 PWM 동작을 정지
파라미터	htim : PWM 동작을 시작할 타이머 구조체 포인터 값(&htim1(timer1) ~ &htim3(timer3)) Channel : 타이머 내에서 지정된 채널(TIM_Channel_1, TIM_Channel_2, TIM_Channel_3, TIM_Channel_4)
실행결과	실행 결과(HAL_OK, HAL_ERROR, HAL_BUSY, HAL_TIMEOUT)
예시	Timer1의 Channel1에 대한 PWM 동작을 정지하고 싶을 때의 코딩: HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);

오실로스코프

oscilloscope



- 오실로스코프(oscilloscope)는 특정 시간 간격(대역)의 전압 변화를 볼 수 있는 장치
- 주로 주기적으로 반복되는 전자 신호를 표시하는데 사용
- 시간에 따라 변화하는 신호를 주기적이고 반복적인 하나의 전압 형태로 파악 가능

LED 밝기 연속적 변화

LED brightness continuous change

```
int main(void) {  
    /* USER CODE BEGIN 2 */  
    HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);  
    /* USER CODE END 2 */
```

```
__HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, 0);  
/* USER CODE BEGIN 2 */  
HAL_TIM_PWM_Start(&htim3, TIM_CHANNEL_1);  
/* USER CODE END 2 */  
  
/* Infinite loop */  
/* USER CODE BEGIN WHILE */  
while (1)  
{  
    for(int i = 0; i < 1000; i+=10){  
        __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, i);  
        HAL_Delay(100);  
    }  
    for(int i = 999; i > 0; i-=10){  
        __HAL_TIM_SET_COMPARE(&htim3, TIM_CHANNEL_1, i);  
        HAL_Delay(100);  
    }  
    /* USER CODE END WHILE */  
  
    /* USER CODE BEGIN 3 */  
}  
/* USER CODE END 3 */
```