

제어용통신 수업자료

NUCLEO-F429 보드를 이용한 모터 엔코더값 읽기

2019. 11 (Ver1.2)

한국산업기술대학교
메카트로닉스공학과

목 차

1. Nucleo-F429보드와 모터 엔코더의 결선
2. 예제 프로그램 다운로드 및 실행
3. 주요 소스코드 설명

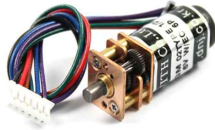
1. Nucleo-F429보드와 모터 엔코더의 결선

1.1. 모터 사양

제조사 : dnj (www.dnj.co.kr)

모델 : RA-12WGM 02TYPE (6V) with 2channel Encoder, 감속비 1/60

- 6V로 구동되는 DC Motor, 1/60 감속비 부착, 엔코더 부착



모터 사양

- Reduction ratio : 1/10 ~ 1/298 (10 kinds)
- Rated torque : 30gf-cm ~ 700gf-cm
- Rated speed : 1,500 rpm ~ 50 rpm
- Install motor : DC 6v / 18,000 rpm / 0.61W

엔코더 사양 : 2 channel Hall effect encdoer

- Encoder : 6P/R

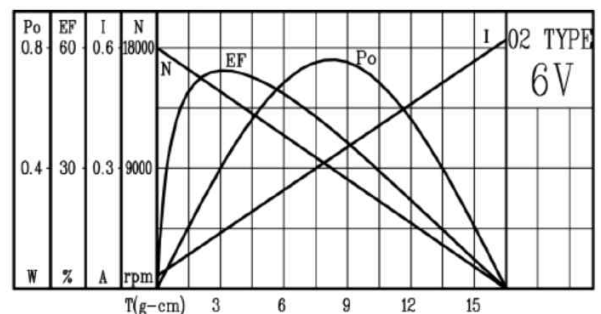
감속모터사양 / Geared Motor Specification

감속기길이(mm) Gear Head L		9.15									
중량(g)		9.5									
감속비 Reduction ratio		1/10	1/19	1/29	1/60	1/76	1/102	1/134	1/197	1/235	1/298
02 TYPE 6V	정격토크(gf-cm) Rated torque	24	40	60	115	150	200	235	345	410	520
	정격 회전수(RPM) Rated speed	1450	770	510	245	195	145	110	75	63	50
	무부하 회전수(RPM) No Load speed	1800	947	620	300	237	176	134	91	76	60

모터사양 / Installed Motor Specification

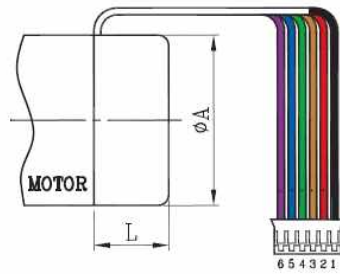
02 TYPE MOTOR (DC6V)	
정격 토크 Rated torque	30(gf-cm)
정격 회전수 Rated speed	14,900(RPM)
정격 전류 Rated current	≤170(mA)
무부하 회전수 No load speed	18,000(RPM)
무부하 전류 No load current	≤50(mA)
정격 출력 Rated output	0.46(W)

그래프 / Graph



1.2 엔코더 사양

Two channel Hall Effect Encoder



엔코더 컨넥터 핀별 내용 :

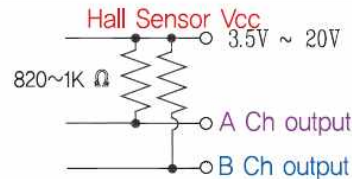
Two Channel Encoder Connections :

1. Black : -MOTOR
2. Red : +MOTOR
3. Brown : HALL SENSOR Vcc
4. Green : HALL SENSOR GND
5. Blue : HALL SENSOR B Vout
6. Purple : HALL SENSOR A Vout

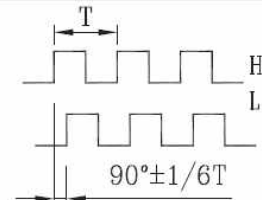
★ WITHOUT CAP

Motor φA	CAP L	COUNTS POLES OF PER TURN(PPR)		Wire Type Length	Connector Type	model
		current	limit.			
φ12	★ 6.5	2, 6 (1, 3)	6 (3)	UL1061 AWG26 100mm	JST ZHR-6 P=1.5-6P	IG12, IG16, RA12W, RA20
φ15.4	★ 6.5	2, 6 (1, 3)	6 (3)			
φ20.3	★ 8.5	2, 6 (1, 3)	6 (3)			
φ30.0	12.6	26 (13)	26 (13)	UL1007 AWG24 100mm	JST PHR-6 P=2.0-6P	IG22, IG22C, IG28, IG30, IG32, IG32P, IG32R, IG36P, IG43, RA35, RB30, RB35, RB40
φ32	14.3	26 (13)	26 (13)			
φ36	13.5	26 (13)	26 (13)			
φ42.5	15.5	38 (19)	38 (19)	UL1007 AWG24 UL1007 AWG18 100mm	JST PHR-4 P=2.0-4P Molex 09-50-3021 P=3.96-2P	IG42, IG52
φ52	18.0	38 (19)	38 (19)			
φ54						

출력회로 구성
Output Circuit :



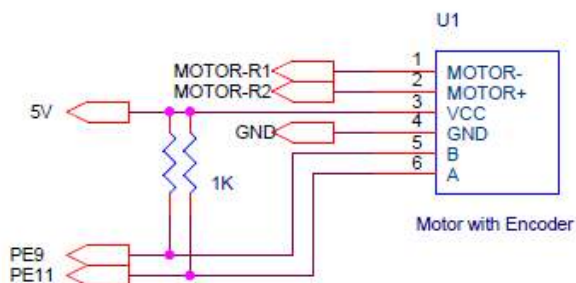
출력 파형
Output Wave :



[출처 : http://www.dnj.co.kr/bbs/board.php?bo_table=spur_en&wr_id=21]

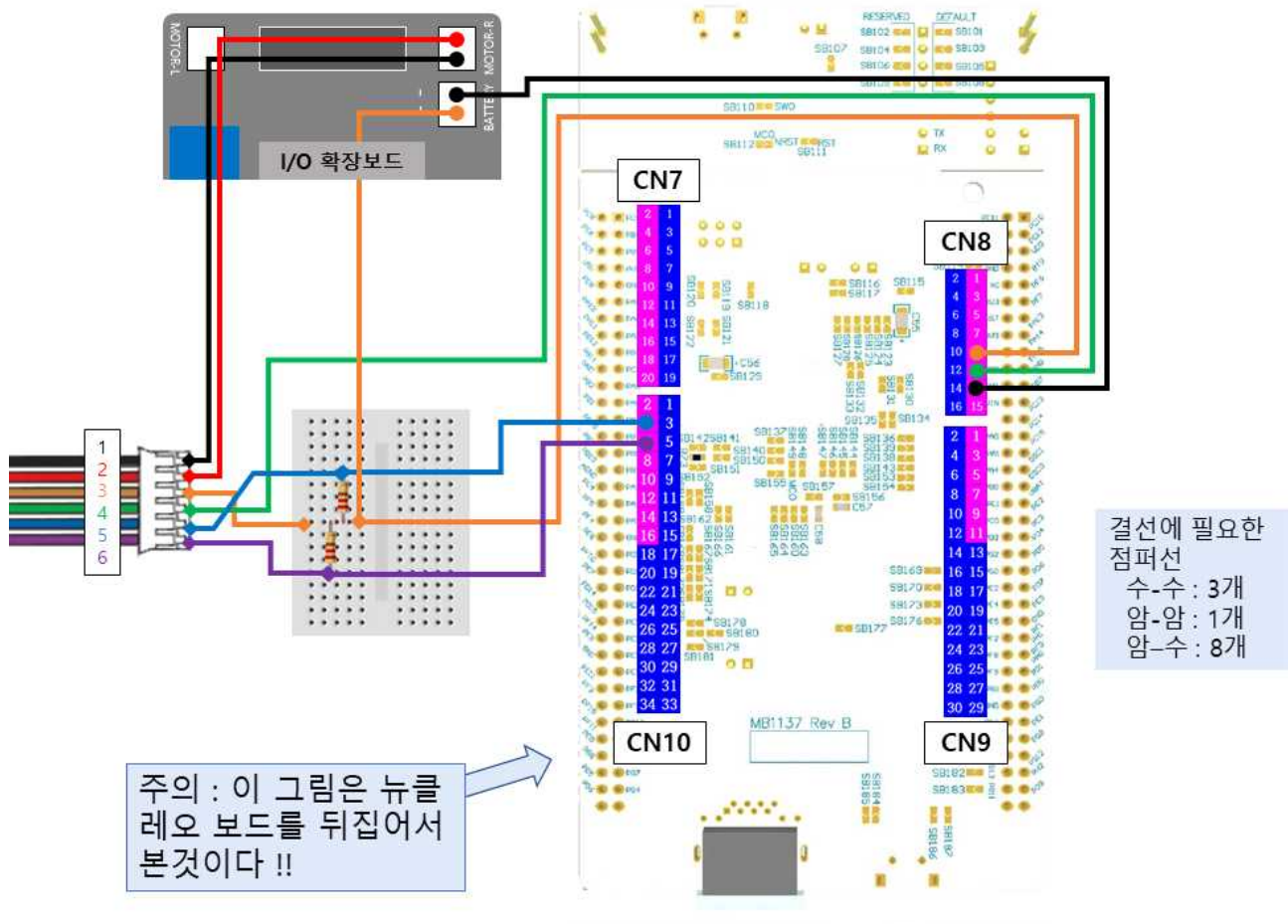
1.3 엔코더 결선도

1) 결선 회로도

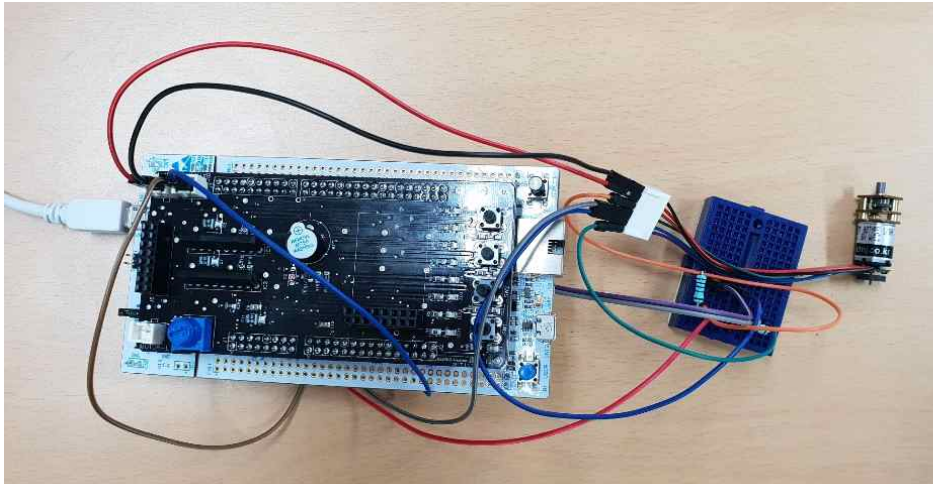


	NUCLEO-F429 보드
5V	CN8의 9번핀
GND	CN8의 11, 13번핀
PE9	CN10의 4번핀
PE11	CN10의 6번핀

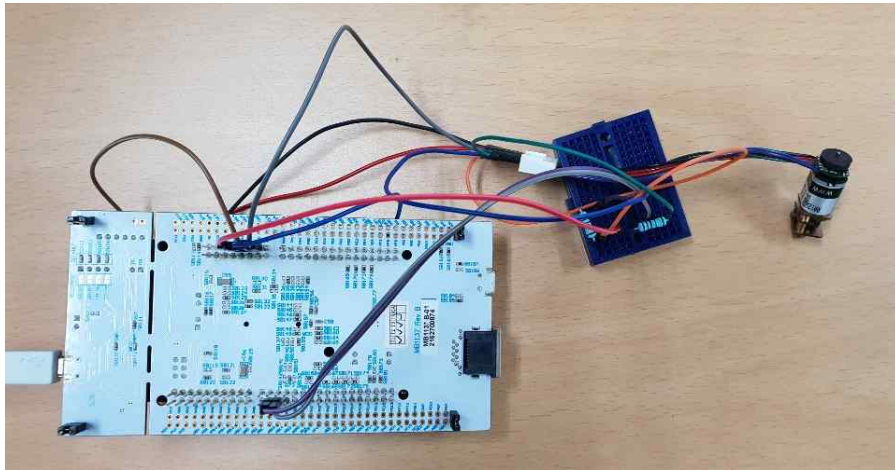
2) 점퍼선을 이용한 결선 예



- 보드의 뒷면 핀 중 5V와 GND핀을 IO 확장보드의 'BATTERY' 커넥터에 연결
- IO 확장보드의 'MOTOR-R' 커넥터에 모터의 MOTOR+- 핀을 연결
- 엔코더의 VCC는 MCU 5V 또는 MCU 3V3를 사용
- 엔코더의 GND는 MCU GND에 연결
- 나머지 부분은 '실제 결선 회로도'와 같이 연결

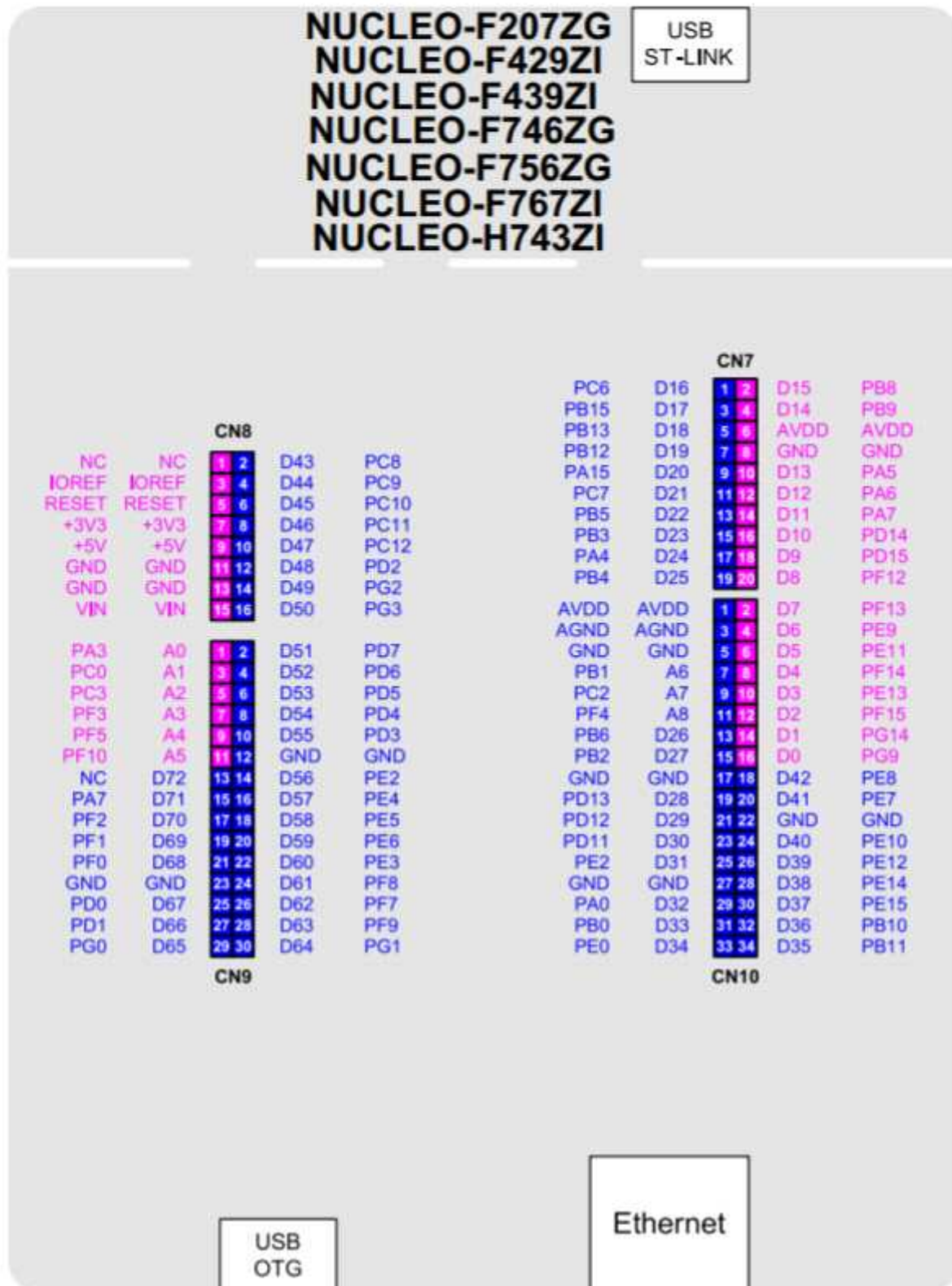


[결선 예 : 윗면에서 본 모습]



[결선 예 : 아랫면에서 본 모습]

[참고] 뉴클레오 보드의 위쪽에서 본 커넥터의 핀 맵



2. 예제 프로그램 다운로드 및 실행

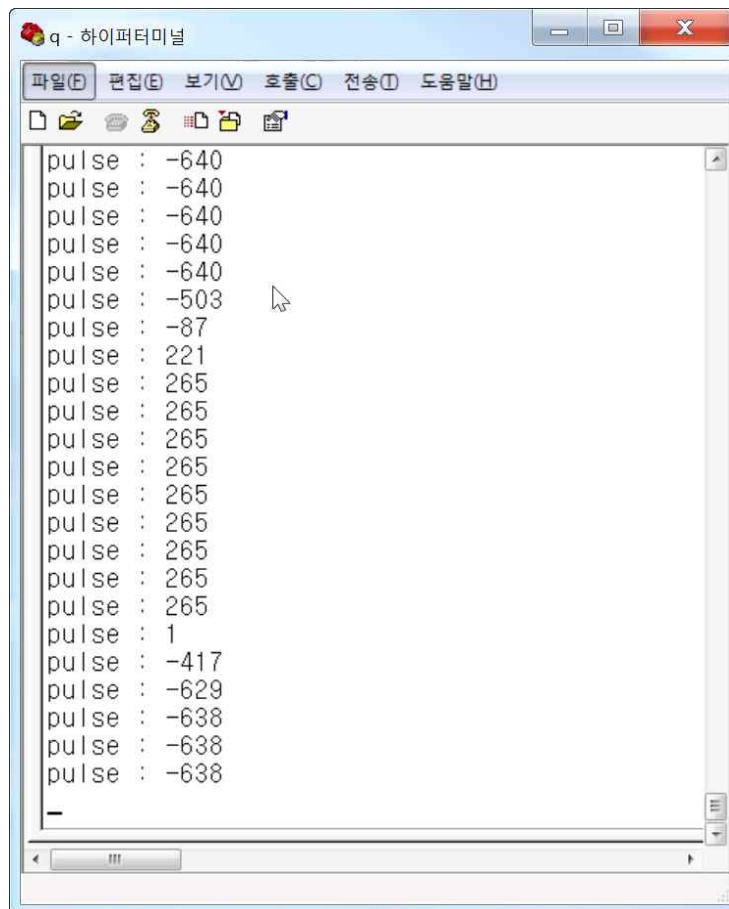
가. 네이버 카페 (Cortexworld)에서 예제를 다운받는다.

(게시글 1036 : F429보드를 이용한 엔코더 읽기)

나. 다운받은 파일의 압축을 푼다.

다. 압축을 푼 파일 중에서 F429ZI_Encoder_Example\MDK-ARM\STM32F429ZI_Example.uvprojx
파일 실행해서 빌드 후 보드에 다운로드

라. BaudRate 115200으로 설정 후 하이퍼터미널에서 동작 확인



3. 주요 소스코드 설명

[main.c]

```
int main(void)
{
    /* -- <1> 초기 설정 -- */
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    MX_USART3_UART_Init();

    MX_TIM1_Init(); // TIM1을 엔코더 모드(외부 카운터 모드)로 설정 (엔코더 펄스를 카운팅)
    MX_TIM2_Init(); // TIM2를 0.1초 마다 인터럽트를 발생하도록 설정
    MX_TIM4_Init(); // TIM4를 PWM 출력모드로 설정 (모터 구동용)
    MX_NVIC_Init();

    printf("Encoder Example\r\n");

    /* -- <2> Timer1을 엔코더 모드(외부 카운터 모드)로 시작, 채널1과 2사용-- */
    HAL_TIM_Encoder_Start(&htim1, TIM_CHANNEL_1 | TIM_CHANNEL_2);
    TIM1->CNT = 0;

    /* -- <3> Timer4를 PWM 모드로 시작, 채널1과 2사용-- */
    HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_1);
    HAL_TIM_PWM_Start(&htim4, TIM_CHANNEL_2);

    while (1)
    {
        /* -- <4> 버튼 입력이 있을 때 까지 대기-- */
        if(!flag.button)
            continue;

        /* -- <5> Timer4 채널1과 2의 Capture/Compare 레지스터 설정(모터 속도 설정) -- */
        TIM4->CCR1 = 299;
        TIM4->CCR2 = 0;

        /* -- <6> 360Pulse(1/2회전)이 넘을 때 까지 회전, 그 후 정지-- */
        while(1) {
            signed short count = TIM1->CNT;
            if(ABS(count) >= 360) {
                TIM4->CCR1 = 0;
                HAL_Delay(1000);
                break;
            }
        }

        HAL_Delay(1000);
    }
}
```

```

/* -- <7> 모터 속도 설정 : (5)와 반대 방향으로 회전 -- */
TIM4->CCR1 = 0;
TIM4->CCR2 = 299;

/* -- <8> 360Pulse(1/2회전)이 넘을 때 까지 회전, 그 후 정지-- */
while(1) {
    signed short count = TIM1->CNT;
    if(ABS(count) >= 360) {
        TIM4->CCR2 = 0;
        HAL_Delay(1000);
        break;
    }
    HAL_Delay(1000);
}
}
}

```

[interrupt.c]

```

#include "interrupt.h"

void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
    /* -- <1> 0.1sec 주기로 PC로 엔코더 펄스 값 전송-- */
    if(htim->Instance == TIM2){
        sprintf(str, "pulse : %dWrWn", (signed short)TIM1->CNT);
        HAL_UART_Transmit_IT(&huart3, (uint8_t*)str, (uint16_t)(strlen(str)));
    }
}

void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)
{
    if(GPIO_Pin == USER_Btn_Pin){
        /* -- <2> 이중 클릭 방지 -- */
        if(flag.button)
            return;
        /* -- <3> NUCLEO 보드의 파란색 버튼(USER B1 Button)을 누르면 동작 시작-- */
        flag.button = 1;
        HAL_TIM_Base_Start_IT(&htim2);
    }
}

```

[참고] 이 예제 프로그램은 엔코더 펄스를 2채배 하여 카운팅하도록 설정되어있다.
즉 모터가 1바퀴 회전하면 카운팅되는 펄스수가 720펄스로 된다.

[참고] 이 예제 프로그램은 CubeMX를 이용하여 만들어진 것이다.