

BLDC Motor 를 전진과 정지만 구동되는것을 위해 문제점과 원인을 찾아보았다.
예전에 했던 코드를 사용하여 ESC설명서를 다시 확인해보았다.

[ESC프로그램하는 법]

1. 프로그램 가능한 리스트 (굵은 체는 처음 설정 값입니다.)

프로그램 가능한 품목	프로그램 값								
	1	2	3	4	5	6	7	8	9
기본 품목									
1. 주행 모드	전진과 정지	전진,후진, 정지	전진,후진 (락크라울링)						
2. 중립 브레이크	0%	5%	10%	20%	40%	60%	80%	100%	
3. 저전압 차단	보호 안함	2.6V/Cell	2.8V/Cell	3.0V/Cell	3.2V/Cell	3.4V/Cell			
4. 편지	Level 1	Level 2	Level 3	Level 4	Level 5	Level 6	Level 7	Level 8	Level 9
고급 옵션									

주행모드를 확인하면 1.전진과 정지, 2.전진, 후진, 정지 모드, 3.전진,후진(락크라울링)모드가 있는데 이것을 설정해야 한다.

1. 전진과 정지 모드인 경우에 켈레브레이션할때 스로틀 범위를 설정하는데 전진과 정지만 설정이 되서 1000~2000 의 신호를 주었을때 작동했다.
2. 전진,후진,정지 모드인 경우에 중립스로틀을 1500으로 주고 전진 최고치를 2000 backward zone을 1000으로 지정해서 1000~1400정도면 후진이 되는 줄 알고 설정을 하였는데 브레이크의 강도만 강해지는 느낌이었다.
3. 전진, 후진(락크라울링)모드에서 브레이크가 없는 모드에서 1500을 중심으로 잡고 1000이 후진의 최고 속도 2000이 전진의 최고속도로 설정하여 하니 성공하였다.

우리 조의 같은경우 BLDC모터 2개를 사용하는데 하나는 역회전으로 하기때문에 설정할때 2000을 후진으로 주고 1000을 전지으로 주면 될 것 같다.

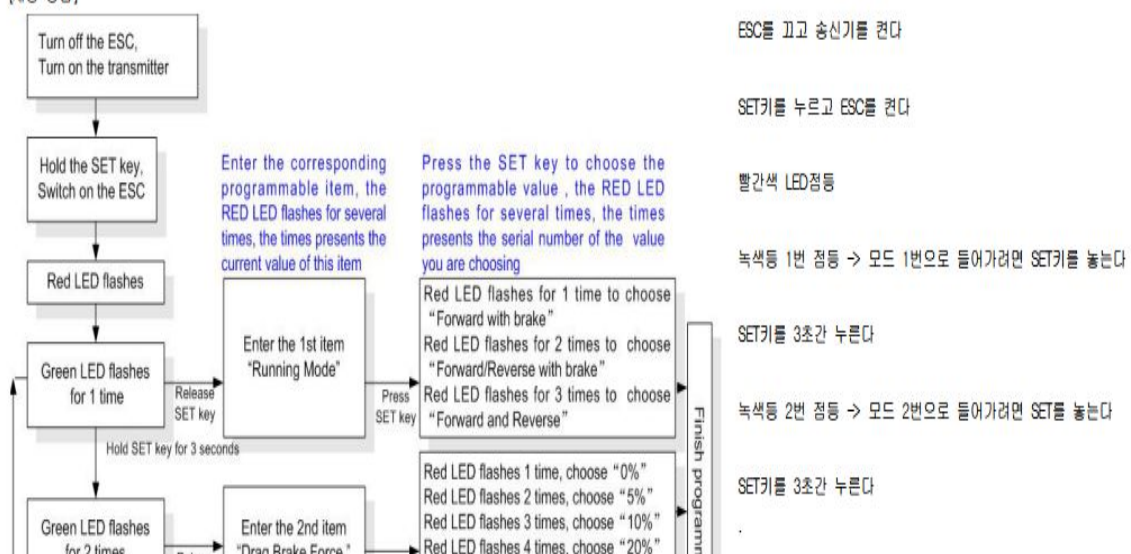
코드는 형준이가 짰다. HAL CoGen 은 전과 같이 하면된다.

아래에는 ESC 주행모드 설정 방법이다.

HOBBYWING 브러시리스 스피드 컨트롤러 XERUN-120A-1S 매뉴얼

페이지 6 / 6 HOBBYWING

[세팅 방법]



```
/* USER CODE BEGIN (0) */
/* USER CODE END */
```

```
/* Include Files */
```

```
#include "HL_sys_common.h"
#include "HL_system.h"
#include "HL_etpwm.h"
#include "HL_sci.h"
#include <string.h>
#include <stdio.h>
#include <math.h>
```

```
#define TSIZE1 10
#define TSIZE2 5
#define TSIZE3 4
```

```
uint8 TEXT1[TSIZE1] = {'H', 'e', 'l', 'l', 'o', ' ', 'T', 'h', 'e', 'C', 'o', 'r', 't', 'e', 'x', '-', 'R', '5', 'F', ' ', 'M',  
'C', 'U'};
```

```
unsigned int ePWM1B = 90;
```

```
void sciDisplayText(sciBASE_t *sci, uint32 *text, uint32 length);
void wait(uint32 time);
```

```
#define UART sciREG1
```

```
uint32 receiveData = 0;
```

```

uint32 temp = 0;
float flat = 1;
int ten = 10;
int squire = 3;

int main(void)
{
    scilnit(); /* SCI/SCI-Lin 초기화, 짝수 패리티, Stop Bits : 2 */
    etpwmInit();

    while(1)
    {
        etpwmStartTBCLK();

        squire = 3; // 10 의 제곱
        receiveData = 0;
        while(1)
        {
            temp = sciReceiveByte(UART);
            sciDisplayText(UART, &temp, TSIZE1); /* Text 전송 */

            temp = temp-48;
            if(temp > 9 || temp < 0)
                break;
            receiveData += temp * pow(ten, squire);
            printf("receiveData = %d\n", receiveData);
            squire--;
        }

        if(receiveData > 0){
            ePWM1B = receiveData;
            etpwmSetCmpB(etpwmREG1, ePWM1B);
            printf("11 ePWM1B = %d\n", ePWM1B);
            wait(4000);
        }else{
            printf("-----\n");
        }

        wait(4000);
    }
}

```

```
void sciDisplayText(sciBASE_t *sci, uint32 *text, uint32 length)
{
    while(length--)
    {
        while((UART->FLR & 0x4) == 4)
            ; /* wait until busy */
        sciSendByte(UART, *text++); /* send out text */
    }
}
```

```
void wait(uint32 time)
{
    time--;
}
/* USER CODE BEGIN (4) */
/* USER CODE END */
```