

Busan Software Meister High School

MICROPROCESSOR

2309 양유빈

20230525 마이크로프로세서

인터럽트

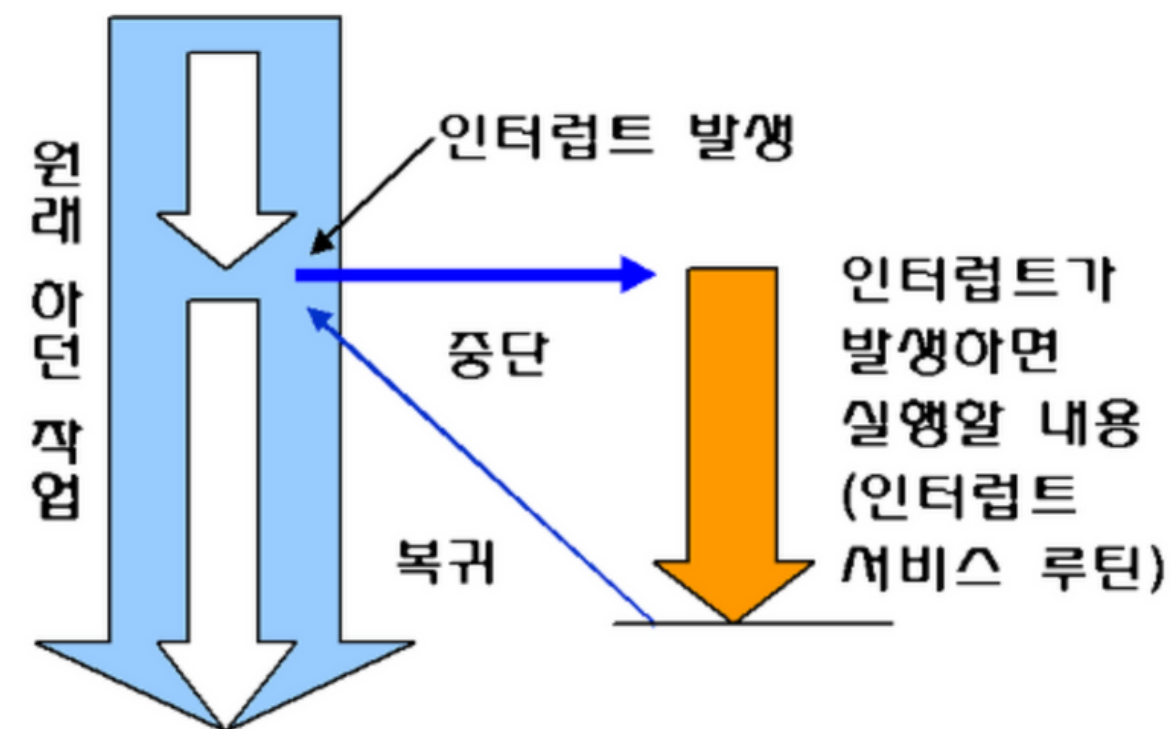
interrupt

폴링(polling) vs 인터럽트(interrupt)

폴링 : 주기적으로 상태를 체크하는 것

인터럽트 : 끼어듦 또는 가로막기라고도 하며, 마이크로프로세서(CPU)가 프로그램을 실행하고 있을 때, 입출력 하드웨어 등의 장치에 예외상황 발생시 마이크로프로세서에게 알려 처리할 수 있도록 하는 메커니즘

- 실행 중인 프로그램(메인 프로그램) 실행 중지
- 인터럽트 처리 프로그램(인터럽트 서비스 루틴) 실행
- 중지되었던 프로그램(메인 프로그램) 다시 실행



GPIO 인터럽트 관련 HAL 라이브러리

HAL library for GPIO interrupts

 형태: void HAL_GPIO_EXTI_IRQHandler(uint16_t GPIO_Pin)

설명	GPIO에서 발생하는 EXTI 인터럽트 처리를 위한 콜백함수 (stm32f10x_it.c 파일 내의 인터럽트 핸들러 함수 [EXTIx_IRQHandler()])에서 호출되어 사용)
파라미터	GPIO_Pin : GPIO pin을 지정하며, GPIO_PIN_0 ~ GPIO_PIN_15 사이의 값을 가짐
리턴값	없음

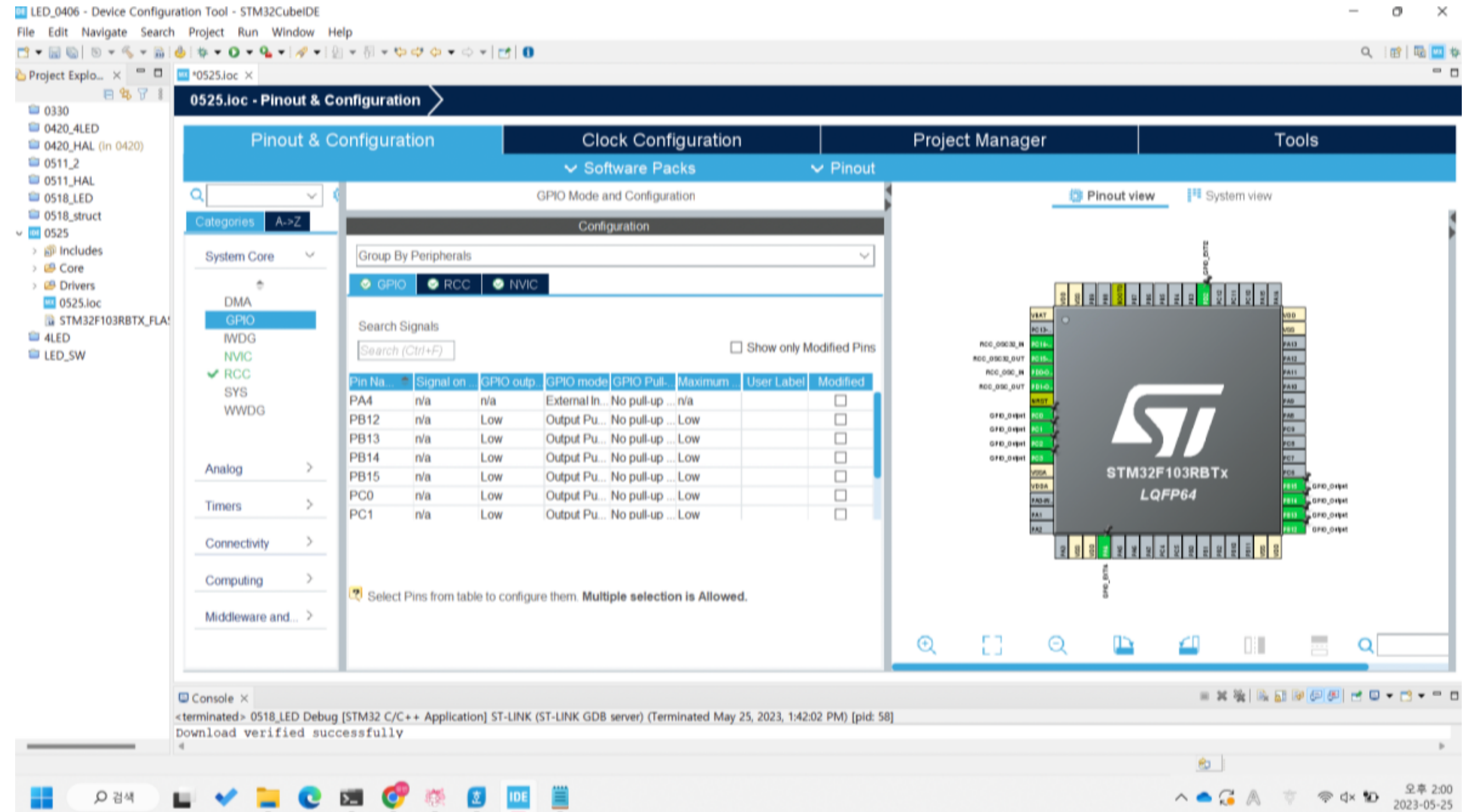
 형태: void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin)

설명	EXTI 인터럽트를 처리하기 위한 콜백 함수(main.c 파일 내에서 구현하여 사용)
파라미터	GPIO_Pin : GPIO pin을 지정하며, GPIO_PIN_0 ~ GPIO_PIN_15 사이의 값을 가짐
리턴값	없음

GPIO 인터럽트

GPIO interrupt //채터링 해결 방안

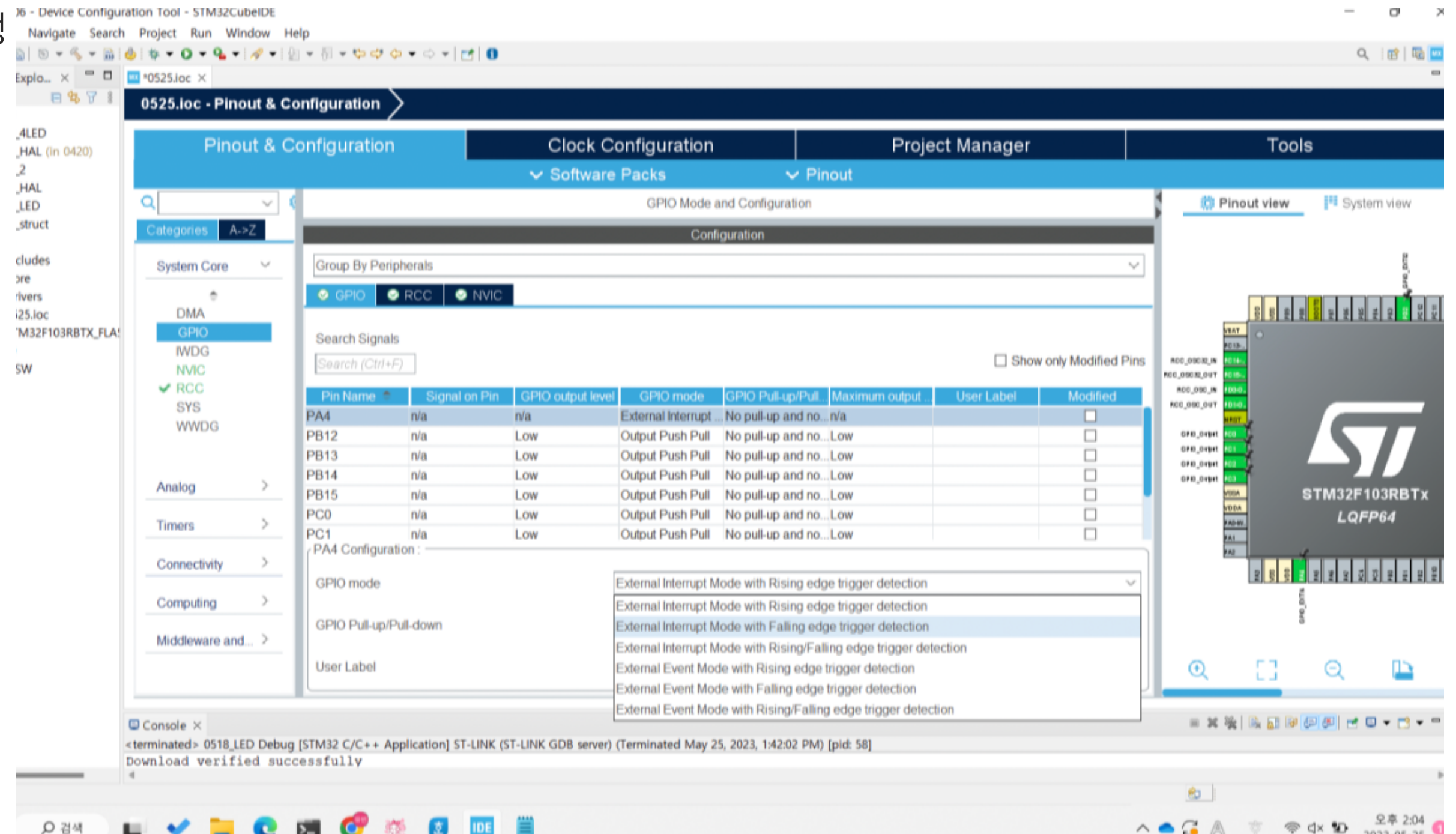
// 초기 세팅



GPIO 인터럽트

GPIO interrupt

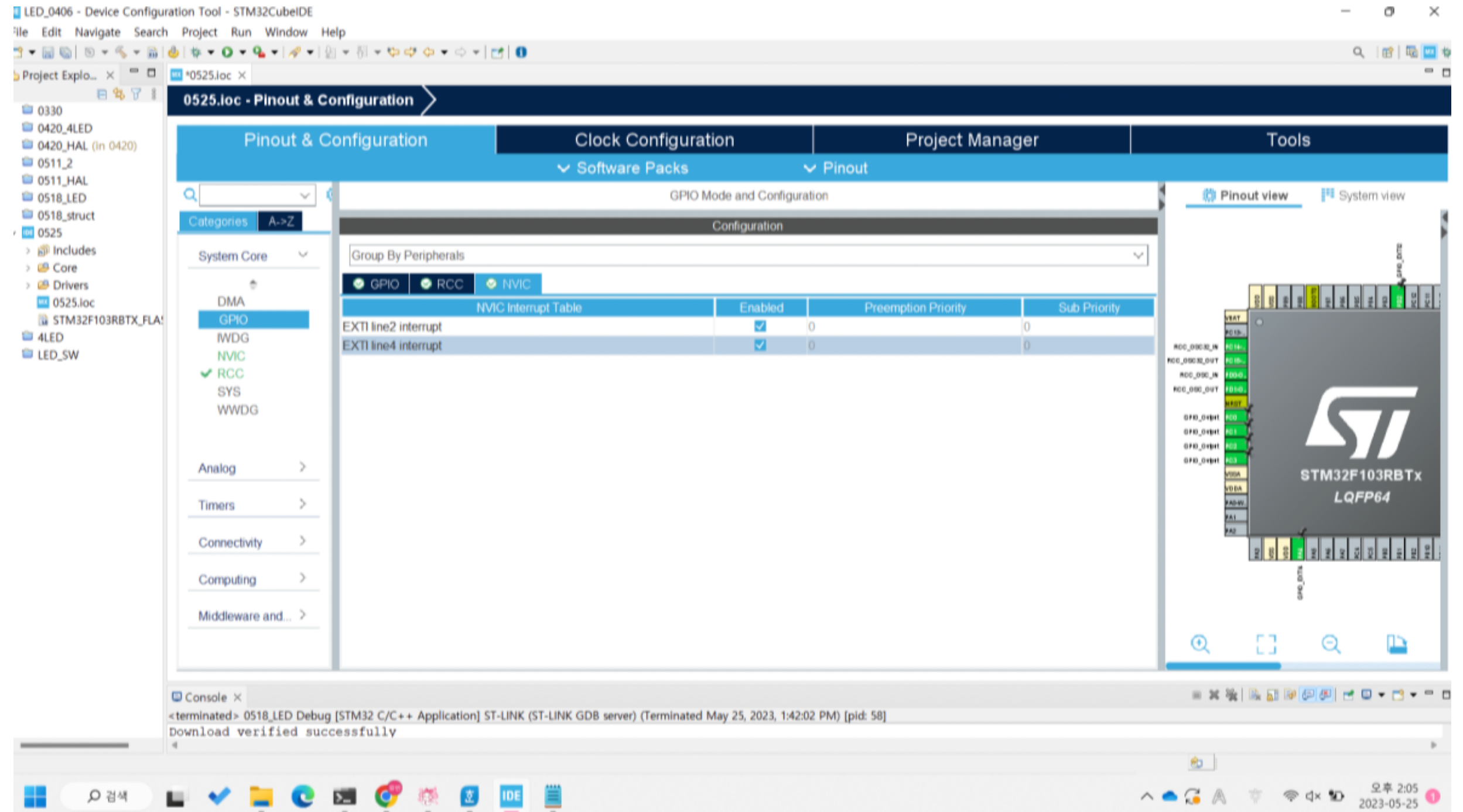
// 인터럽트 사용하려면 Falling으로 수정



GPIO 인터럽트

GPIO interrupt

// 인터럽트 사용할건지 확인



LED ON/OFF 프로그래밍-인터럽트

LED ON/OFF Programming - Interrupt

📌 채터링 발생 X

```
197 /* USER CODE BEGIN 4 */
198 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
199     if (!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)) {
200         if (led_state == LED_ON) {
201             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 0); //control+space
202             HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 0);
203             led_state = LED_OFF;
204         } else {
205             HAL_GPIO_WritePin(GPIOB, GPIO_PIN_15, 1);
206             HAL_GPIO_WritePin(GPIOC, GPIO_PIN_0, 1);
207             led_state = LED_ON;
208         }
209     }
210 }
211
212 /* USER CODE END 4 */
```

SW로 비상등 모드 변경 프로그램

Emergency light mode change program with SW

구조체 코드

```
59 /* USER CODE BEGIN 0 */
60 typedef struct led {
61     GPIO_TypeDef * port;
62     uint16_t pin;
63 }LED;
64 LED led[8] = {
65     {GPIOC, GPIO_PIN_3}, {GPIOC, GPIO_PIN_2},
66     {GPIOC, GPIO_PIN_1}, {GPIOC, GPIO_PIN_0},
67     {GPIOB, GPIO_PIN_15}, {GPIOB, GPIO_PIN_14},
68     {GPIOB, GPIO_PIN_13}, {GPIOB, GPIO_PIN_12}
69 };
70
71 void ambulance_a() {
72     for (uint8_t i = 0; i < 4; i++) {
73         HAL_GPIO_WritePin(led[i].port, led[i].pin, 1);
74         HAL_GPIO_WritePin(led[i + 4].port, led[i + 4].pin, 0);
75     }
76     HAL_Delay(300);
77     for (uint8_t i = 0; i < 4; i++) {
78         HAL_GPIO_WritePin(led[i].port, led[i].pin, 0);
79         HAL_GPIO_WritePin(led[i + 4].port, led[i + 4].pin, 1);
80     }
81     HAL_Delay(300);
82 }
83
84 void ambulance_b() {
85     for (uint8_t i = 0; i < 8; i++) {
86         HAL_GPIO_WritePin(led[i].port, led[i].pin, 1);
87         HAL_Delay(100);
88         HAL_GPIO_WritePin(led[i].port, led[i].pin, 0);
89         HAL_Delay(100);
90     }
91     for (uint8_t j = 6; j > 0; j--) {
92         HAL_GPIO_WritePin(led[j].port, led[j].pin, 1);
93         HAL_Delay(100);
94         HAL_GPIO_WritePin(led[j].port, led[j].pin, 0);
95         HAL_Delay(100);
96     }
97 }
98 /* USER CODE END 0 */
```

코드

```
133     while (1) {
134         if(ambulance_mode == MODE_A) ambulance_a();
135         else ambulance_b();
136         /* USER CODE END WHILE */
137
138         /* USER CODE BEGIN 3 */
139     }
```

```
239 #if 1
240 void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin) {
241     if(!HAL_GPIO_ReadPin(GPIOD, GPIO_PIN_2)) {
242         if(ambulance_mode == MODE_A) {
243             ambulance_mode = MODE_B;
244         } else {
245             ambulance_mode = MODE_A;
246         }
247     }
248 }
249
250 #endif
```

```
32 /* Private define -----
33 /* USER CODE BEGIN PD */
34 // #define LED_OFF 0
35 // #define LED_ON 1
36 #define MODE_A 0
37 #define MODE_B 1
38 /* USER CODE END PD */
39
40 /* Private macro -----
41 /* USER CODE BEGIN PM */
```


SW로 비상등 모드 변경 프로그램

Emergency light mode change program with SW

+에러해결-코드수정, led_all_off 추가

```
99 void led_all_off() {  
100     for(uint8_t k=0; k<8; k++){  
101         HAL_GPIO_WritePin(led[k].port, led[k].pin, 0);  
102     }  
103 }
```

```
77 void ambulance_a() {  
78     for (uint8_t i = 0; i < 4; i++) {  
79         HAL_GPIO_WritePin(led[i].port, led[i].pin, 1);  
80         HAL_GPIO_WritePin(led[i + 4].port, led[i + 4].pin, 0);  
81     }  
82     HAL_Delay(300);  
83     if(ambulance_mode == MODE_B) {  
84         led_all_off();  
85         return; //??? return!!  
86     }  
87     for (uint8_t i = 0; i < 4; i++) {  
88         HAL_GPIO_WritePin(led[i].port, led[i].pin, 0);  
89         HAL_GPIO_WritePin(led[i + 4].port, led[i + 4].pin, 1);  
90     }  
91     HAL_Delay(300);  
92     if(ambulance_mode == MODE_B) {  
93         led_all_off();  
94         return;  
95     }  
96 }  
97  
98 void ambulance_b() {  
99     for (uint8_t i = 0; i < 8; i++) {  
100         HAL_GPIO_WritePin(led[i].port, led[i].pin, 1);  
101         HAL_Delay(100);  
102         HAL_GPIO_WritePin(led[i].port, led[i].pin, 0);  
103         HAL_Delay(100);  
104         if(ambulance_mode == MODE_A) {  
105             led_all_off();  
106             return;  
107         }  
108     }  
109     for (uint8_t j = 6; j > 0; j--) {  
110         HAL_GPIO_WritePin(led[j].port, led[j].pin, 1);  
111         HAL_Delay(100);  
112         HAL_GPIO_WritePin(led[j].port, led[j].pin, 0);  
113         HAL_Delay(100);  
114         if(ambulance_mode == MODE_A) {  
115             led_all_off();  
116             return;  
117         }  
118     }  
119 }
```