**임베디드시스템 설계 및 실험 보고서(6주차)**



**8조**

작성자 : 202055516 김명서

조원:

201602181 정진성

201824483 박진영

201824523 안혜준

201924660 한병정

1. **실험 목표 및 기본 개념**
   1. **실험 목표**
2. Interrupt 방식을 활용하여 GPIO 제어와 UART 통신 방법을 배운다.
3. 라이브러리 함수 사용법을 숙지한다.

**1.2 기본 개념**

이벤트가 발생하는지 지속적으로 감시하는 polling 방식과 달리 interrupt는 특정 이벤트가 발생하면 진행하던 작업을 멈추고 해당 인터럽트 서비스 루틴을 수행한 후 다시 이전 작업으로 돌아간다.

인터럽트는 하드웨어 인터럽트와 소프트웨어 인터럽트로 나뉜다. 하드웨어 인터럽트는 비동기식 event 처리로 주변장치의 요청에 의해 발생하는 interrupt이고 소프트웨어 인터럽트는 프로세서 자체에서 발생하는 인터럽트이다. 이번 실험에서는 외부 인터럽트인 EXTI를 다룬다.

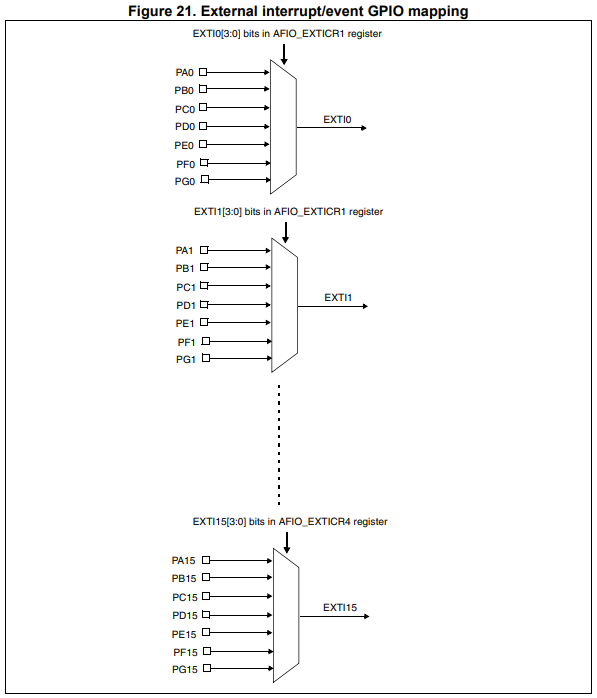
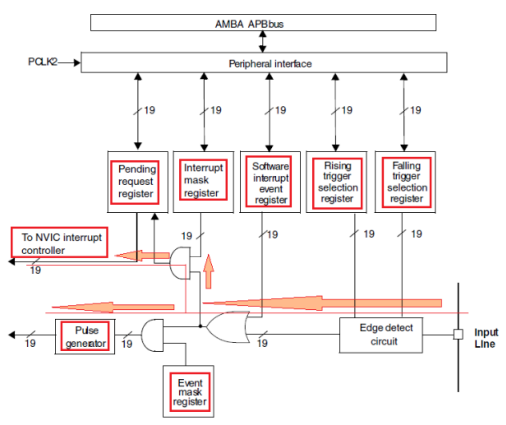
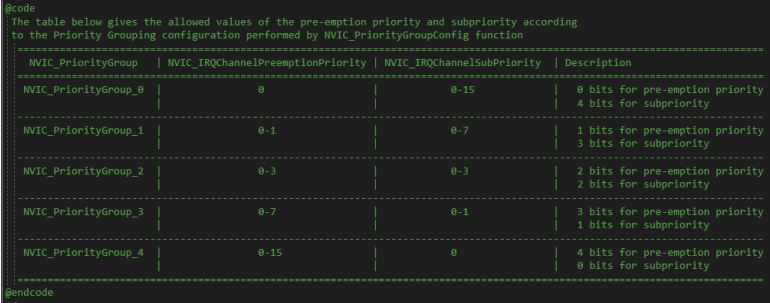


그림 1



EXTI는 External interrupt/event controller의 줄임말로 Rising-Edge 또는 Falling-Edge의 신호를 인식하여 interrupt를 발생시킨다. 각 port의 n번 pin은 EXTI n에 연결된다. EXTI는 Event Mode 와 Interrupt Mode 를 선택하여 설정 가능하나, 해당 실험에서는 interrupt Mode만을 사용한다. 인지된 interrupt는 pending register에 저장되어 그 중 priority가 가장 높은 interrupt부터 처리한다.



그림

이번 실험에서는 라이브러리에 정의된 IRQHandler의 정의된 이름을 그대로 사용하여 원하는 함수를 구현한다. 또한 인터럽트의 우선순위는 NVIC를 통해서 결정할 수 있다. 우선순위는 값이 작을수록 운선순위가 높다.

**2. 실험 과정**

* 1. **RCC 클럭 인가**

RCC\_APB2PeriphClockCmd 함수를 이용하여 UART, JoyStick, LED, USART, AFIO 클럭을 인가해준다.

RCC\_APB2PeriphClockCmd를 사용하는 방법은 다음과 같다.

*RCC\_APB2PeriphClockCmd(활성화하고자 하는\_포트, ENABLE);*

아래의 그림3을 통해서 USART는 GPIOA에, JoyStick은 GPIOC/GPIOB에 해당하고, LED는 GPIOD에 해당함을 알 수 있다.

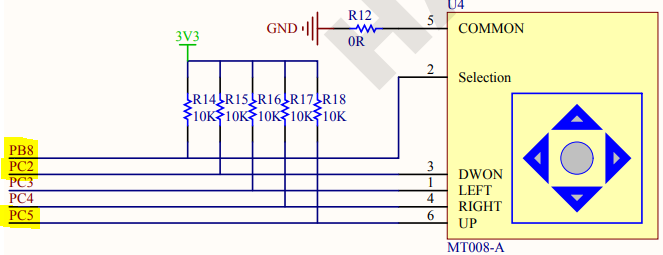
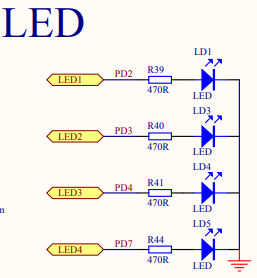
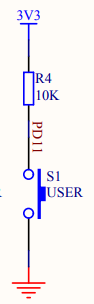


그림 3



따라서 최종 코드는 다음과 같다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* 1. **초기화**

2.2.1 GPIO 핀 초기화

핀을 사용하기위해서 GPIO를 초기화한다. GPIO\_InitTypeDef 에서 다음을 정의하여 사용한다.

*GPIO\_InitStructure.GPIO\_Pin = (초기화하고자하는 핀 번호들을 정의)*

*GPIO\_InitStructure.GPIO\_Mode = (input인지 output인지를 정의)*

* Button과 LED는 그림 3을 통해서 Pull Up임을 알 수 있다.

*GPIO\_InitStructure.GPIO\_Speed = (output의 경우 속도를 정의)*

* 이 실험에서는 기본적인 50MHz를 사용

텍스트이(가) 표시된 사진

자동 생성된 설명*GPIO\_Init(포트 식별자, &GPIO\_InitStructure);*

* + 1. EXTI 초기화

인터럽트 발생에 사용할 EXTI에 대한 설정을 초기화한다. 각 포트 번호에 맞게 EXTI 라인을 선택한다.

*EXTI\_InitStructure.EXTI\_Line = (초기화하고자하는 line 정의)*

*EXTI\_Mode* 🡪 이번 실험에서는 interrupt를 다룬다.

*EXTI\_Trigger = (인터럽트를 인식할 엣지 상승/하강)*

*EXTI\_LineCmd = ENABLE*

*EXTI\_Init (&EXTI\_InitStructure*

조이스틱 selection의 경우 이번 실험에서 사용하지 않으므로 작성하지 않았다.

텍스트이(가) 표시된 사진

자동 생성된 설명

* + 1. USART 초기화

시리얼 통신을 위해 USART를 설정한다. USART\_InitTypeDef 에서 다음을 정의한다. Parity와 HardwareFlowControl은 사용하지 않으므로 None으로 설정한다.

*USART\_BaudRate* 🡪 9600으로 사용

*USART\_StopBits = (전달되는 stop bit의 수)*

*USART\_WordLength = (전달하고자하는 글의 길이)*

*USART\_Mode = (데이터를 송신-RX할지, 수신-TX할지를 정의)*

*USART\_Init(USART1, &USART\_InitStructure);*

텍스트이(가) 표시된 사진

자동 생성된 설명*USART\_ITConfig(활성화할\_USARTx, 송수신 상태 ,ENABLE);*

* + 1. NVIC 초기화

여러 인터럽트를 관리하기 위해 우선순위를 지정하여 초기화한다. 이번 실험에서는 조이스틱, s1버튼, UART 순서대로 우선순위를 부여했다. 조이스틱의 경우 preemptionPriority는 같으나 두 동작을 구분하기 위해 SubPriority에 차등을 둔다. 숫자가 작을수록 우선순위가 높다

*NVIC\_IRQChannel = (사용하고자하는 포트의 IRQ)*

*NVIC\_IRQChannelPreemptionPriority = (우선순위 번호 부여)*

*NVIC\_IRQChannelSubPriority = (sub번호 부여)*

*NVIC\_IRQChannelCmd = ENABLE;*

텍스트이(가) 표시된 사진

자동 생성된 설명*NVIC\_Init(&NVIC\_InitStructure);*

.

* 1. **IRQHandler**

2.2.4에서 작성한 각각의 IRQ에 대해 처리할 IRQHandler를 작성한다. USART1의 경우 입력받은 값(a 또는 b)에 따라 led의 불빛 이동 방향을 바꿔줘야하며, EXTI15\_10은 팀이름 메시지를 출력해야한다. 또한 EXTI9\_5는 led방향을 순방향으로, EXTI2는 led방향을 역방향으로 재생해야한다. Led의 방향은 direction이라는 전역변수를 통해서 제어하였다. 한번 인식된 인터럽트가 무한히 반복 입력되지 않도록 각 Handler함수 수행 후 ‘Read data register not empty' 플래그를 clear해준다.

텍스트이(가) 표시된 사진

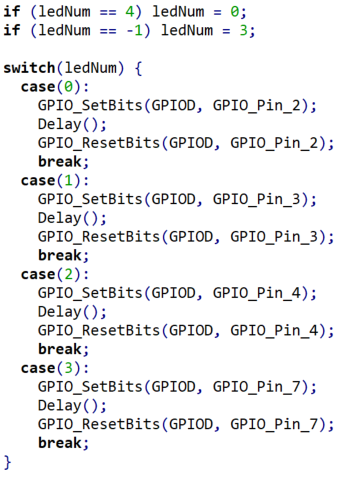
자동 생성된 설명

텍스트, 실내, 스크린샷이(가) 표시된 사진

자동 생성된 설명

* 1. **main문 작성**

led의 순서를 나타내는 ledNum에 1또는 -1을 나타내는 전역변수, direction만큼 더해 순방향 혹은 역방향으로 순회한다.



또한 s1의 버튼을 클릭하면 전역 변수인 message의 값이 1로 변하면서 팀 이름을 출력한다.

텍스트이(가) 표시된 사진

자동 생성된 설명

**s**

1. **실험 결과**

보드를 켜면 LED 물결 기능 유지 (LED 1->2->3->4->1->2->3->4->1->… 반복)

A (=조이스틱 up 또는 a입력) LED 물결 방향 변경 - 1->2->3->4

B (=조이스틱 down 또는 b입력) LED 물결 방향 변경 - 4->3->2->1

S1 버튼을 누를 경우 Putty로 “TEAM08.\r\n“ 출력

실험 결과 코드와 동영상은 별도로 첨부하였다. 아래의 사진은 해당 동영상을 캡쳐하여 사용하였다.

1. 시리얼 통신창에 a 또는 b를 입력하면 led의 순회 방향이 바뀐다.

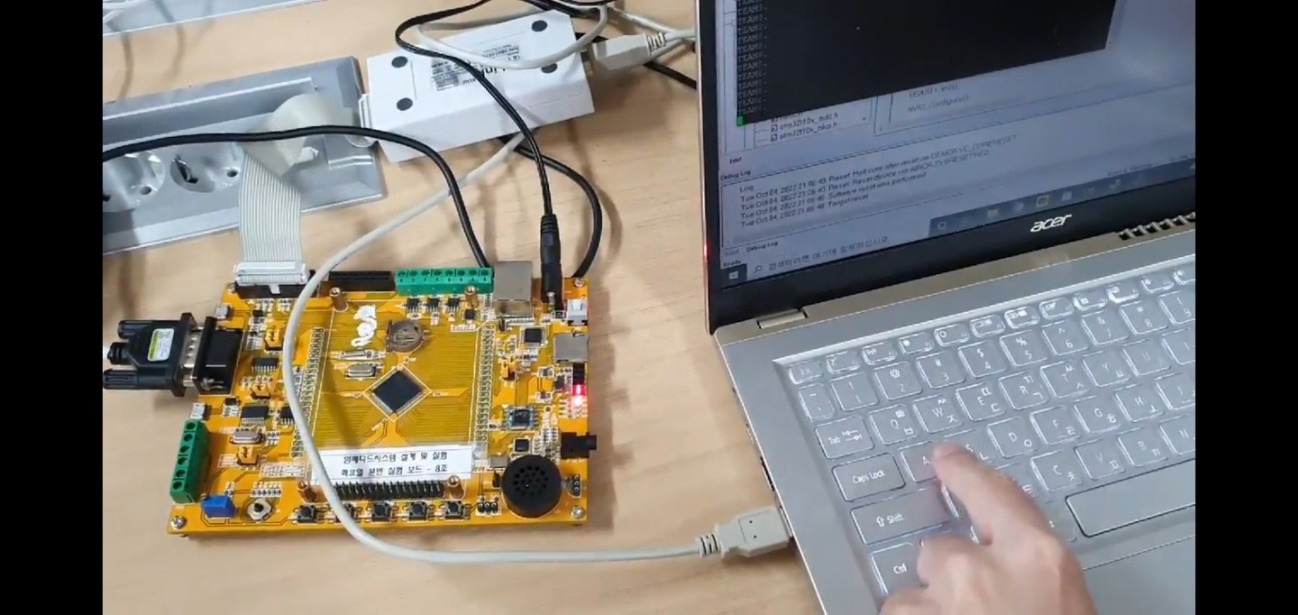
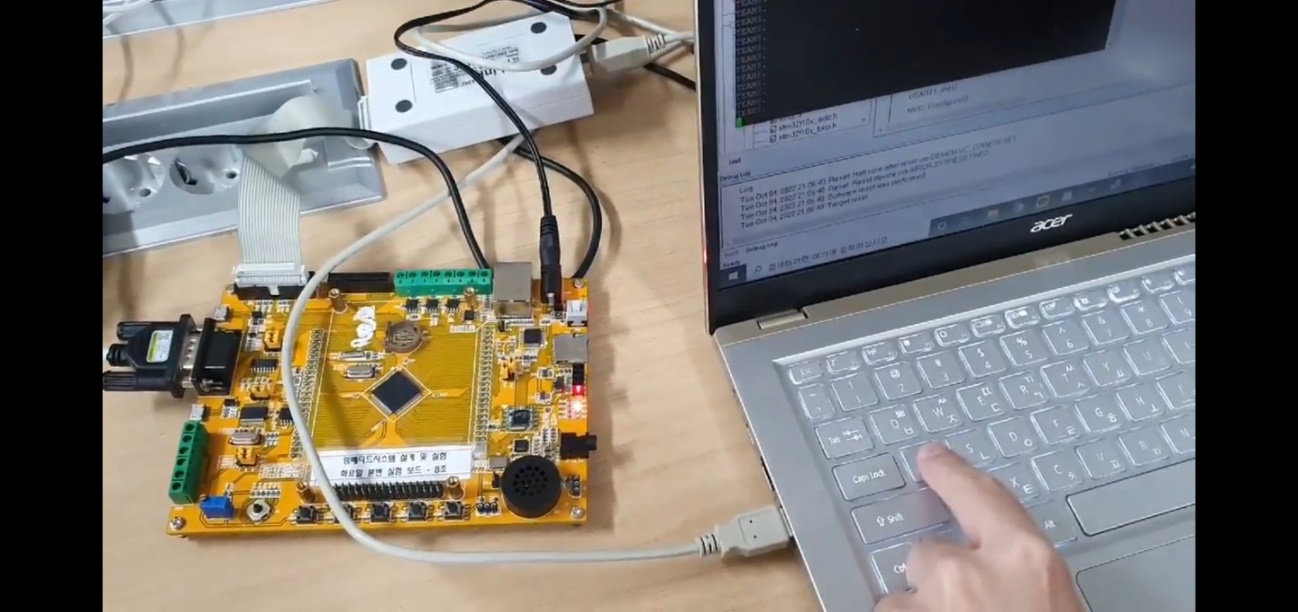


그림3. a 입력 시



1. 조이스틱을 위 또는 아래로 움직여서 led의 순회 방향을 바꾼다.

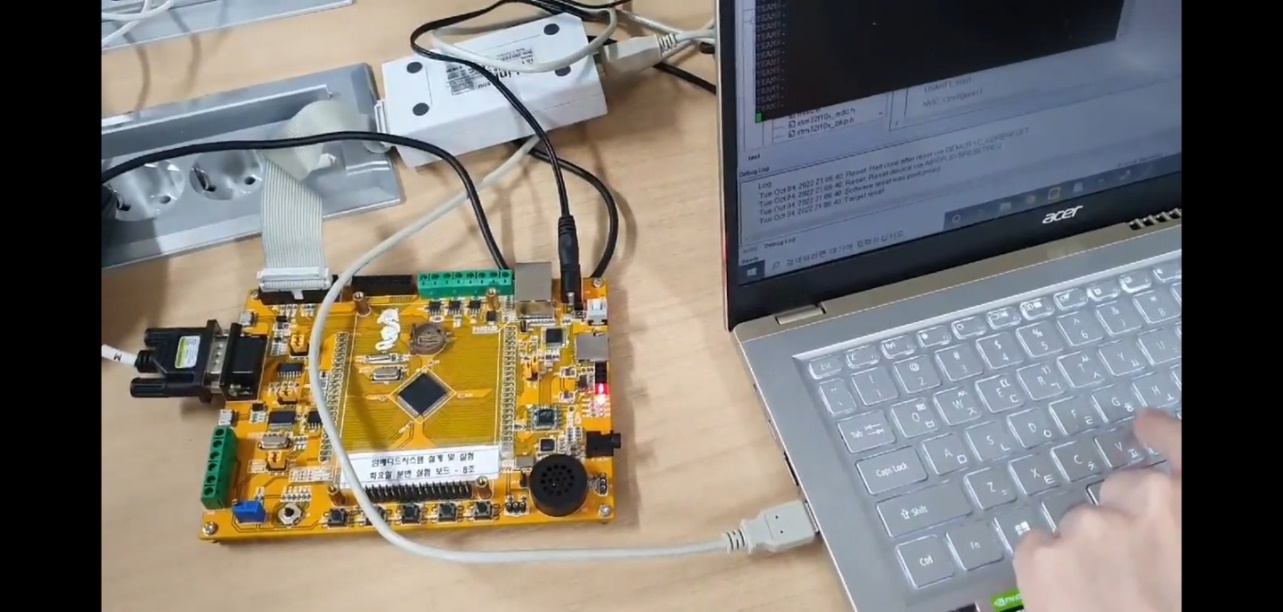


그림 5. b 입력 시

1. 텍스트, 전자기기, 컴퓨터, 디스플레이이(가) 표시된 사진

   자동 생성된 설명S1버튼을 누르면 team8이 출력된다.