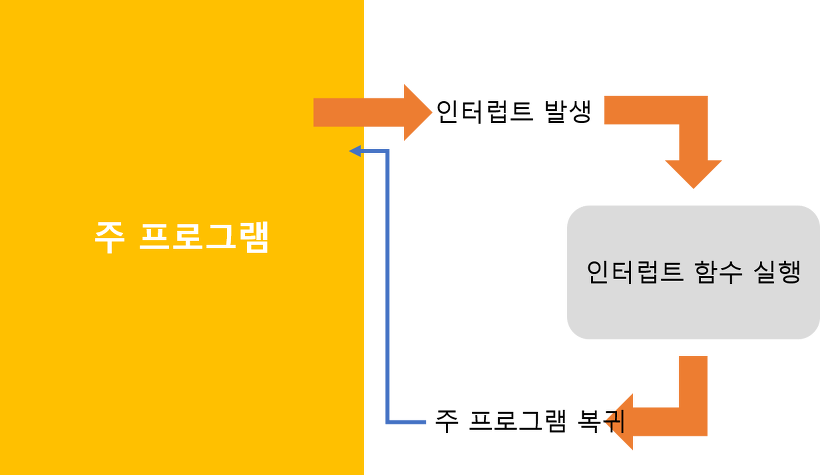
## **1. 인터럽트의 개념**

인터럽트는 지정된 핀의 Input 상태가 원하는 **조건**과 일치 하면 미리 등록한 인터럽트 callback 함수(ISR, Interrupt Service Reoutines)를 자동으로 호출해주는 기능입니다. 인터럽트가 발생하면 실행중이던 주 프로그램을 잠시 중단하고, callback 함수가 끝날때까지 잠시 대기 상태가 됩니다.



이러한 방식의 인터럽트 기능을 "하드웨어 인터럽트" 라고 합니다.

(참고 / 인터럽트 방식중에 timer(타이머) 인터럽트 라는 기능도 있음.)

다시한번 설명 해보자면,

**1. 주 프로그램이 작동하던 중에 인터럽트 핀에 신호가 발생하면,**

**2. 주 프로그램의 작동을 일시 중단하고,**

**3. 인터럽트 함수를 실행,**

**4. 인터럽트 함수 실행을 마치면,**

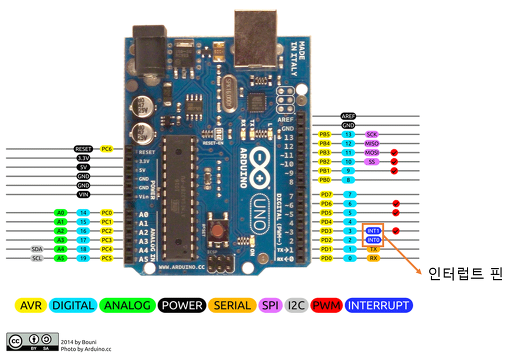
**5. 다시 주 프로그램의 중단지점으로 돌아가 아무일도 없었다는 듯이 이어서 실행**

**(다만 인터럽트 함수에서 실행된 것은 유지(ex. 변수값을 true에서 false로 바꾸었다면 false인 상태로 실행))**

인터럽트 기능은 signal 상태에 따라서 자동으로 특정 동작을 수행하게 해주기 때문에 프로그램 실행 타이밍 문제를 해결하는 강력한 도구입니다.

주로 Rotary encoder, User input 등을 처리하거나 할때 사용되곤합니다.

## **2. 아두이노 인터럽트 핀 연결**

****

아두이노 UNO를 기준으로 설명드리겠습니다. UNO는 디지털 2번, 3번 핀이 인터럽트핀으로 할당되어있습니다.

| Number **0** | D2 |
| --- | --- |
| Number **1** | D3 |

D2 핀은 인터럽트 0번 / D3 핀은 인터럽트 1번으로 지정이 되어있습니다.

추가로 다른 보드들에 대해서도 설명하자면,

| **보드 종류** | **int.0** | **int.1** | **int.2** | **int.3** | **int.4** | **int.5** |
| --- | --- | --- | --- | --- | --- | --- |
| **Uno** | 2 | 3 |  |  |  |  |
| **Mega2560** | 2 | 3 | 21 | 20 | 19 | 18 |
| **Leonardo** | 3 | 2 | 0 | 1 | 7 |  |
| **Due** | 모 | 든 | 핀 | 이 | 지 | 원 |

이러한 인터럽트 넘버를 기억해야 하는 이유는 인터럽트 기능을 사용할 때에는 핀번호를 사용하지 않고 **인터럽트 넘버**를 사용하기 때문입니다.

## **3. 인터럽트 함수**

| 1 | attachInterrupt( **interrupt\_number**, **ISR**, **mode** ); | [cs](http://colorscripter.com/info#e) |
| --- | --- | --- |

| **interrupt\_number** | 인터럽트 번호를 입력 (핀번호 아님주의 / 단, due 만 핀번호 사용) |
| --- | --- |
| **ISR** | 인터럽트가 발생할 때 실행 할 callback 함수 |
| **mode** | 인터럽트가 수행될 조건 |

여기서 잠깐? mode에서 인터럽트가 수행될 조건은 또 뭔가요? 라고 하실 수 있는데 다음 챕터에서 이어서 설명하도록 하겠습니다.

## **4. 하드웨어 인터럽트 모드(mode)**

인터럽트를 실행할때에는 특정 signal이 발생했을 때! 라고 앞서 이야기 했습니다. 이러한 특.정.시.그.널 을 설정하는 것이 바로 mode 입니다.

| **LOW** | pin 이 LOW 상태 일때 |
| --- | --- |
| **CHANGE** | pin 입력 값이 변결 될 때 ( H -> L 또는 L -> H ) |
| **RISING** | LOW -> HIGH 로 변경 될 때 |
| **FALLING** | HIGH -> LOW로 변경될 때 |

(추가) HIGH : pin이 HIGH 상태 일 때 (DUE 에서만 지원)

## **5. 주의 사항**

1. 인터럽트 callback 함수(ISR)는 파라미터를 전달하거나 리턴할 수 없습니다. void xxx\_callback() 형태가 됩니다.
2. ISR 안에서는 delay() 함수를 사용할 수 없습니다.
3. ISR이 실행될 때 millis() 를 사용하더라도 값이 증가하지 않습니다.  
   delayMicroseconds() 의 경우에는 인터럽트에 독립적이므로 정상 동작합니다.
4. ISR 안에서 Serial data를 읽을 경우 값이 소실됩니다.
5. ISR 안에서 수정되는 **전역 변수는 volatile** 로 선언되어야 합니다.  
   (항상 메모리에 상주 할 수 있게 한다.)
6. ISR은 최대한 짧고 빠르게 수행되도록 작성해야 합니다.
7. 여러개의 ISR이 등록되어 있더라도 동시에 수행되지는 않습니다.
8. 블루투스 모듈과의 통신을 위해 주로 사용되는 SoftwareSerial 라이브러리의 경우 내부적으로 인터럽트를 사용하는 것으로 알려져 있습니다. 따라서 UNO 보드의 경우 D2, D3 에 연결해야만 정상동작 합니다.  
   따라서, 블루투스 모듈과의 통신을 사용할 경우 인터럽트 기능을 사용하지 않고 읽고자 할때에만 정보를 수신하고 싶다면, D2 D3 핀을 제외한 다른 핀에 연결해 주는것이 좋습니다.

## **6. 예제 코드**

| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | int pin = 13;  volatile int state = LOW;    void setup(){  pinMode(pin, OUTPUT);  attachInterrupt(0, blink, CHANGE);  }    void loop(){  digitalWrite(pin, state);  }    void blink(){  state = !state;  }  [*Colored by Color Scripter*](http://colorscripter.com/info#e) | [cs](http://colorscripter.com/info#e) |
| --- | --- | --- |

(출처 : 하드카피월드)

앞서 인터럽트에서 수정하는 전역변수는 volatile로 선언해야 한다고 했습니다. state 변수를 전역으로 선언한 뒤, ISR 에서 수정하도록 하겠습니다. 따라서 아래와 같이 선언해줘야 합니다.

| 2 | volatile int state = LOW; | [cs](http://colorscripter.com/info#e) |
| --- | --- | --- |

인터럽트를 사용하려면 미리 인터럽트를 사용한다고 아두이노에 알려줘야 겠죠. setup() 초기화 함수 안에서 attachInterrupt() 함수를 이용하면 됩니다. 여기서는 핀 입력값이 변경될 때 마다 실행되도록 하기 위해 CHANGE 모드를 사용했습니다.

| 6 | attachInterrupta (0, blink, CHANGE); | [cs](http://colorscripter.com/info#e) |
| --- | --- | --- |

attachInterrupt() 함수에 사용된 blink 는 함수를 나타냅니다. 즉, 스케치 어딘가에 blink() 함수의 동작을 정의 해두어야 합니다.

| 13  14  15 | void blink(){  state = !state;  } | [cs](http://colorscripter.com/info#e) |
| --- | --- | --- |

blink() 함수는 단순하게 입력값이 변경되었음을 알려주는 변수 state의 값을 HIGH->LOW 또는 LOW->HIGH로 바꿔줍니다.

blink() 함수는 ISR callback() 함수이고, 이 안에서 수정되는 전역 변수가 state이기 때문에 앞서 state를 volatile로 선언했음을 상기하세요.