

workflow node

1. 종류 :

- action : IoT 기기에 내리는 명령
- condition : 조건문 (해당 조건을 만족하면 내부 내용 실행, 아니면 바로 다음 노드로 이동)
- loop : 반복문
- cwork : 해당 노드의 자식 노드들을 동시 실행
- trigger : workflow가 실행될 조건 (반드시 <flow>의 첫번째 자식노드이어야 함.)

2. 구조

```
struct node {
    enum node_t type; //노드의 종류

    int appid; //기기 아이디
    uint command, option; //기기에게 내릴 명령 및 옵션

    struct node *next //다음 노드 주소
    struct node *childe //자식 노드 주소
};
```

3. 종류별 내용 할당

variable	unsigned int command	unsigned int option	
action	명령	알림	
condition	명령 (반드시 기기로부터 return 값이 있음.)	비교식 (1 byte)	비교값 (3 bytes)
loop			
co-work	사용하지 않음.		

* 비교식 : ==, >= 등

* condition & loop : 명령을 기기에 내리고 기기로부터 받은 리턴값과 비교값을 비교식으로 비교

* trigger는 condition과 동일하게 내용이 할당됨.

* 명령과 알림, 비교값의 각 비트를 어떻게 사용할지는 제조사 마음대로.

ex) 시스템 내장 시간을 비교하는 경우

비교값 (3 bytes)			
요일 (8 bits) (첫비트 외 7개 각각 : 일 ~ 토요일)	시 (5 bits) (0~23)	분 (6 bits) (0~60)	초 (6 bits) (0~60)

ex) 시스템 내장 카운터를 비교하는 경우

비교값 (3 bytes)
값

* 저번 회의 내용 때, notification 노드를 삭제하기로 하였고, 원래 노드의 내용을 명령부에 추가하여야 하지만, 일단 기본 기능 구현이 우선이 되어야하므로 알림 기능을 제외하고 하기로 하였기 때문에 실제로 <action>노드의 option 변수는 항상 0

* 기본 기능이 제대로 동작하는지를 확인하는게 우선이므로, condition 노드에는 시간만을, loop 노드에는 시간과 카운트만을 넣도록 함.

* 시스템 내장 시계의 appid 는 1, 시스템 내장 카운터의 appid는 2로 일단 배정시킴. 또한 현재시간과 현재 카운터값을 구해오는 명령을 1로 배정.