**REPORT**

**| 과목: 임베디드 소프트웨어 프로그래밍**

**| Deadline : 2019/12/15**

**| 컴퓨터공학과**

**| 12171598 김용준**

**| 12151375 김현석**

**| 임베디드 소프트웨어 프로그래밍 최종과제.**

# [0] : 프로젝트 설명

# [1] : 코드 분석

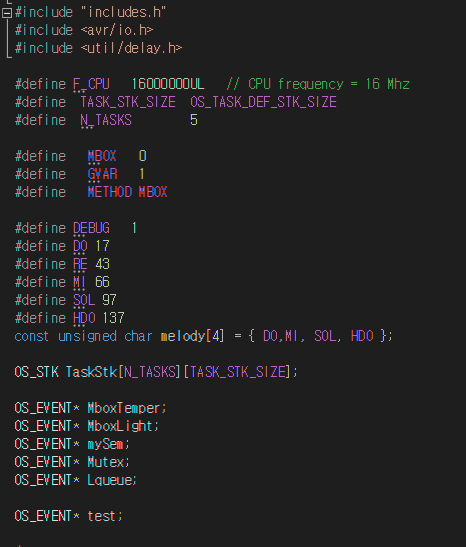
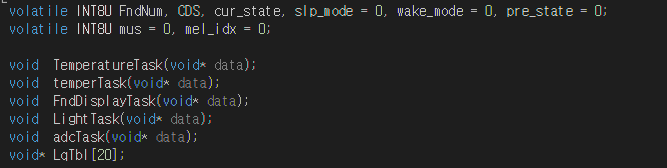
# [2] : 보완해야할 점

# 

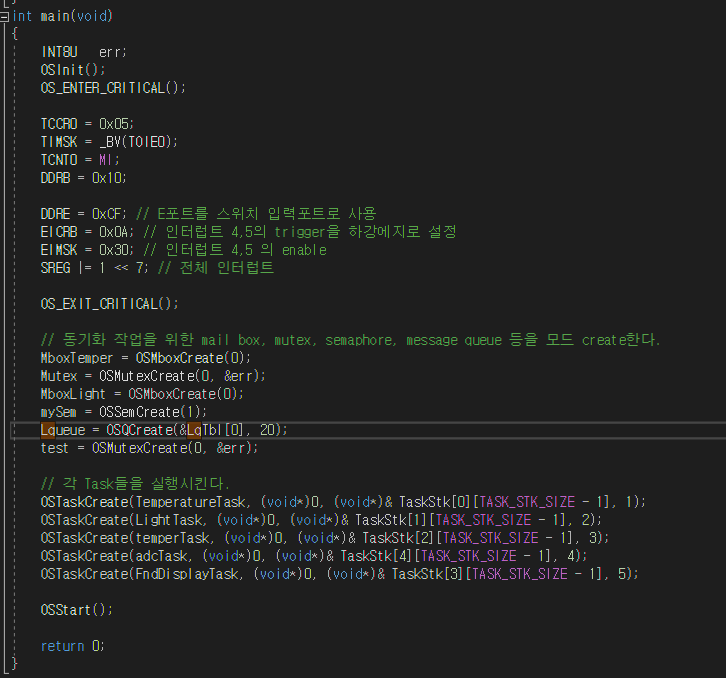
1. **프로젝트 설명**

해당 프로젝트는 자동으로 실내의 온도를 조절하고 내부의 광량에 따라서 커튼을 닫고 내부의 조명을 조절하는 임베디드를 구현하려는 의도로 만들어졌습니다. 해당 프로그램은 잠을 잘 때 내부의 온도를 조절할 수 있는 모드(온도가 일정온도 이상으로 상승 시 에어컨을 작동시키고 일정온도 밑으로 내려가면 자동으로 히터를 가동한다.)와 깨어 있을 시에는 외부의 광량에 따라서 커튼을 치고 닫으면서 내부의 조명을 밝히는 모드로 나뉜다. 처음에 스위치 버튼으로 on/off를 조절할 수 있고 off시에 on 스위치를 누르면 sleep 모드가 켜진다. Sleep 모드에서 위에서 설명한 것처럼 온도에 따라서 히터와 에어컨이 가동된다. 이 상태에서 Wakeup모드 스위치를 누르게 되면 광량에 따라 커튼이 자동으로 작동하는 모드로 이동한다. 낮 시간 동안 외부의 햇빛이 강하면 자동으로 커튼을 치게 되고 어둡게 된 내부에 약하게 조명을 밝힌다. 또 해가 진 이후의 시간에는 커튼을 걷고 내부의 조명을 밝힌다. 이때 내부의 조명은 밖이 어두운 정도에 따라서 조명의 세기를 다르게 나타낼 수 있다. 또한 커튼이 치고 걷힐 때는 부저가 울리게 되어 커튼의 변화를 알릴 수 있다.

1. **코드 분석**

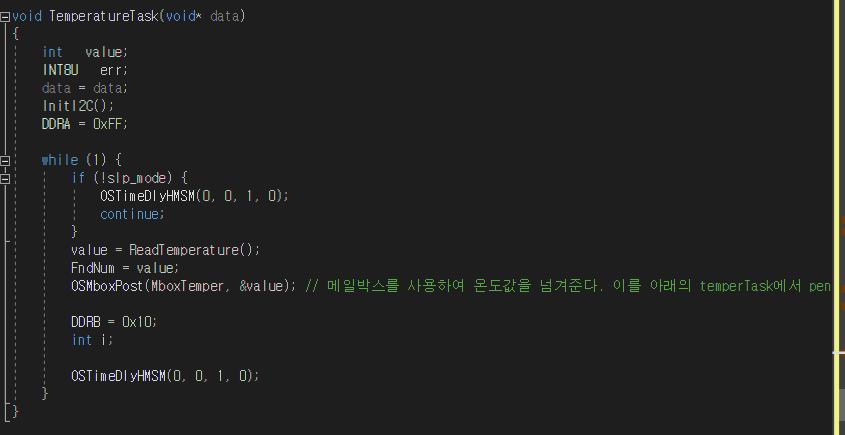
 

먼저 변수의 선언 내용은 위와 같다. **태스크를 5개 만들고 온도를 전달하는 메일박스, 공유변수를 보호하는 세마포어, 그리고 광량을 전달하는 메시지 큐** 를 생성하였다. **센서는 온도센서, 조도센서를 사용하고, 그리고 스위치2개와 부저를 사용하였다.**

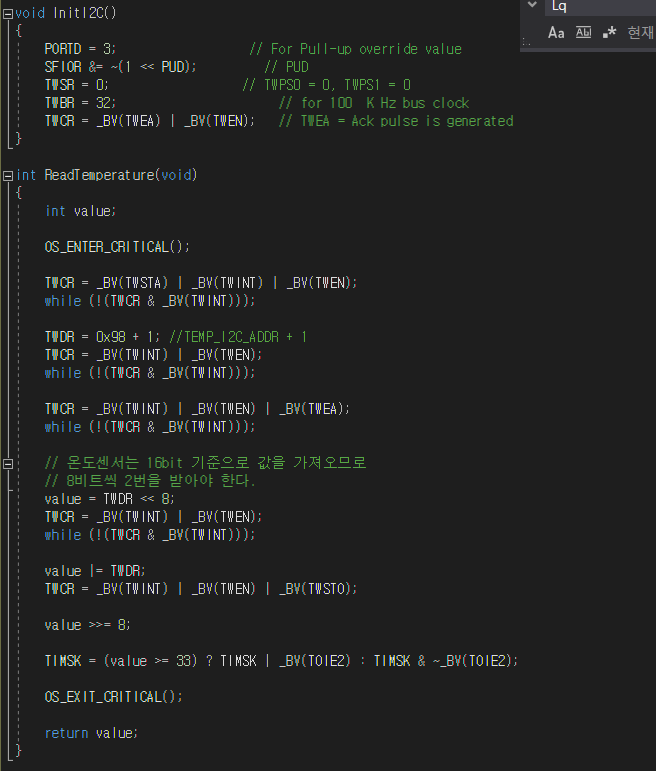


메인함수의 내용이다 타이머0의 분주비를 128로 설정하고 부저를 출력포트로 설정하고, 스위치를 입력포트로 설정, 인터럽트 트리거는 하강에지 트리거, 4번과 5번의 인터럽트를 enable 하였다. 그 이후의 내용은 위의 메일박스,메시지큐,뮤텍스,세마포어를 생성하였다.

이제 태스크의 내용을 보면 다음과같다. 먼저 온도를받아오는 태스크이다.

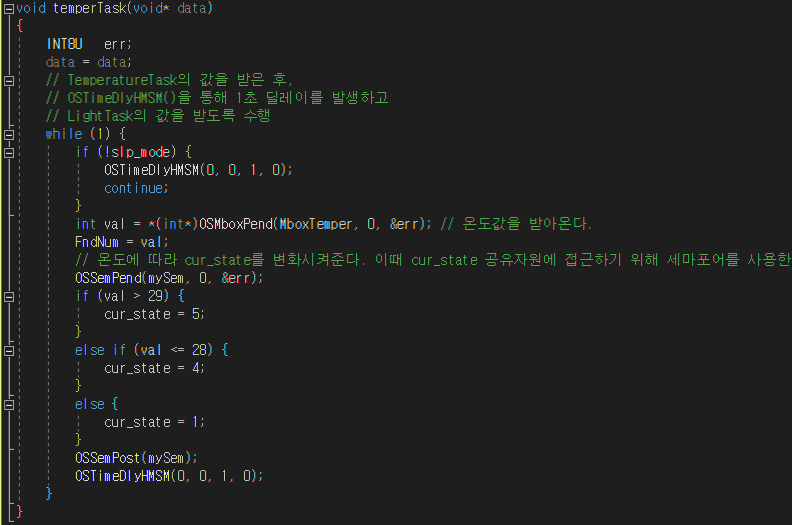


온도 태스크는 wakeup모드에서는 작동할 필요가없으므로 OSTIMEDLYHMSM를 주어서 우선순위가 높음에도 불구하고 실행이 되지않도록 하였다. readtemperature에서 읽어온 현재 온도값을 value에 저장하고 메일박스에 보낸다.



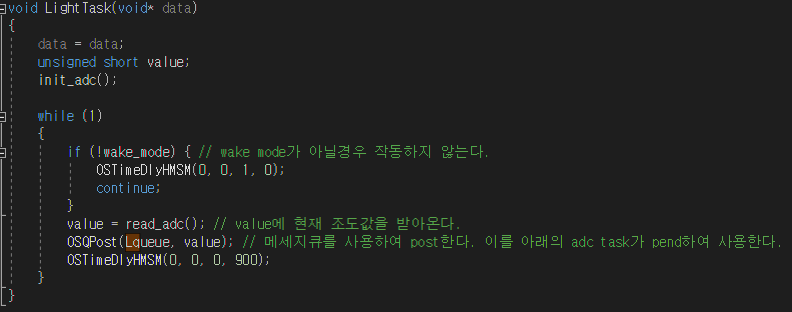
온도를 읽는 포트를 초기화하고 온도값을 받아오는 함수는 위와 같다.

동기화는 다음 태스크에서 일어난다.

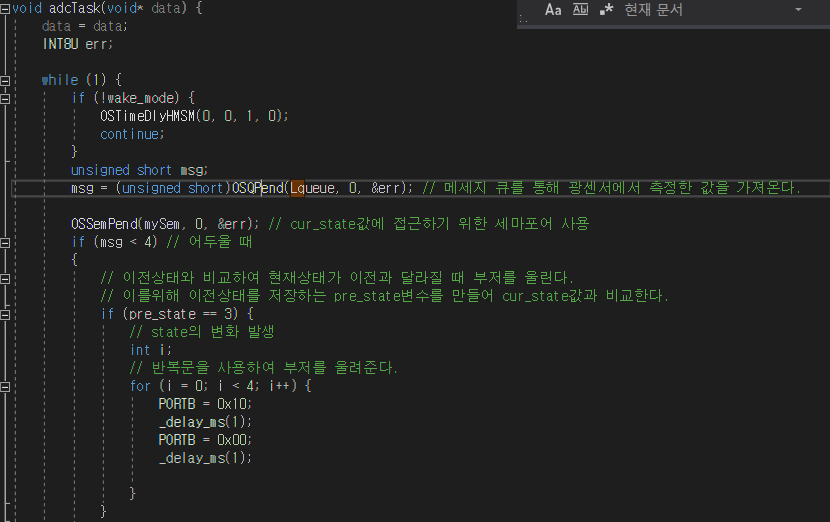
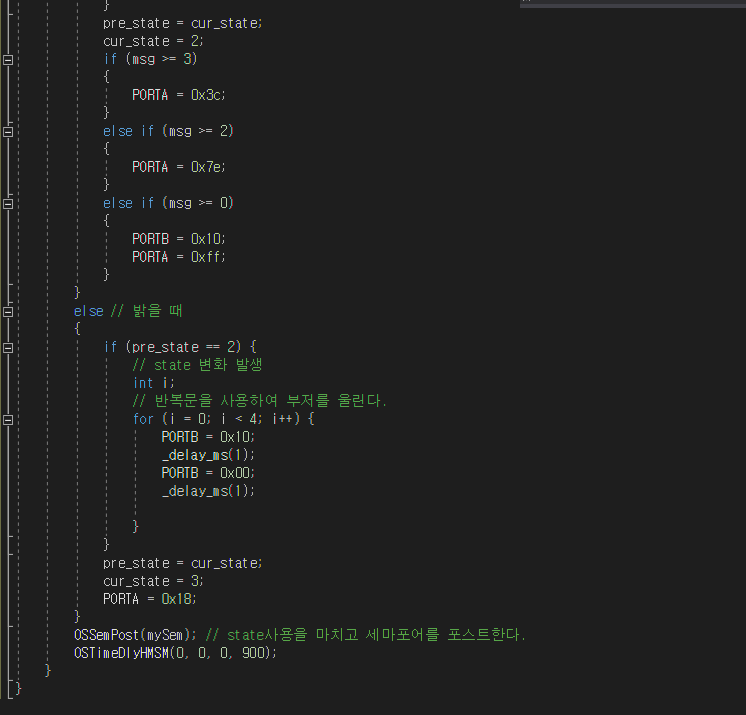


이 태스크는 역시 wakeup모드에서는 실행이 되지않는다. 메일박스에서 temperatureTask가 보낸 값을 꺼내고 온도에 따라서 state 변수를 초기화한다. State 변수는 출력시 비교하는 값이므로 이후에 보도록하자.

다음은 광량을 받아오는 태스크이다. 해당 태스크는 빛을 받아오는 태스크와 비슷하지만 이번엔 메시지큐로 동기화를 실행하였다. 코드는 다음과 같다.

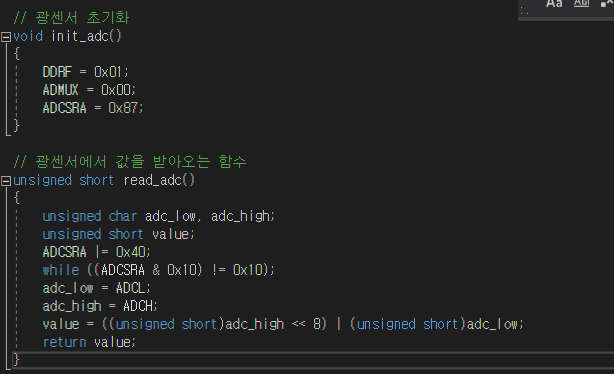


역시 sleep모드에서는 태스크는 수행되지않고 우선순위를 뺏기게 된다. Value에 현재 광량을 저장하고 이번에 메시지 큐에 저장하게 된다. 동기화는 다음 태스크에서 이루어진다.

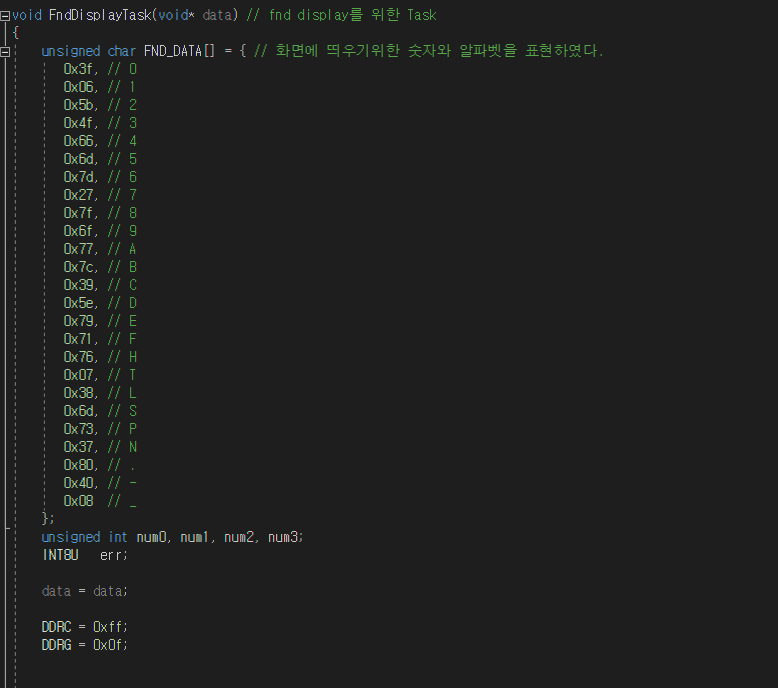
 

해당 태스크는 메시지 큐에서 받아온다. 여기서 온도를 받아오는 태스크와 다른점은 이 태스크는 커튼이 걷히거나 닫힐 때 부저를 울리게 되어있으므로 이전의 상태를 체크해야한다. **이 부분에서 cur\_state는 전역변수이므로 세마포어를 이용해 보호해야한다.** 전의 상태를 검사한후에 바뀌었을시에 반복문을 이용해 부저를 울리게 된다. 그렇지 않으면 cur\_state을 저장하고 불을 밝히게 되는데 광량의 값에 따라 LED의 점등하는 빛의 개수가 다르게 된다.

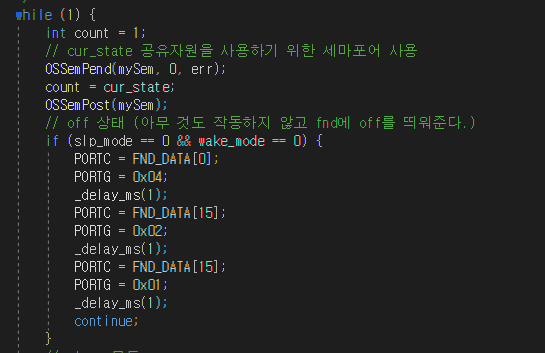
광센서에서 광량을 리턴하는 함수는 다음과 같다.



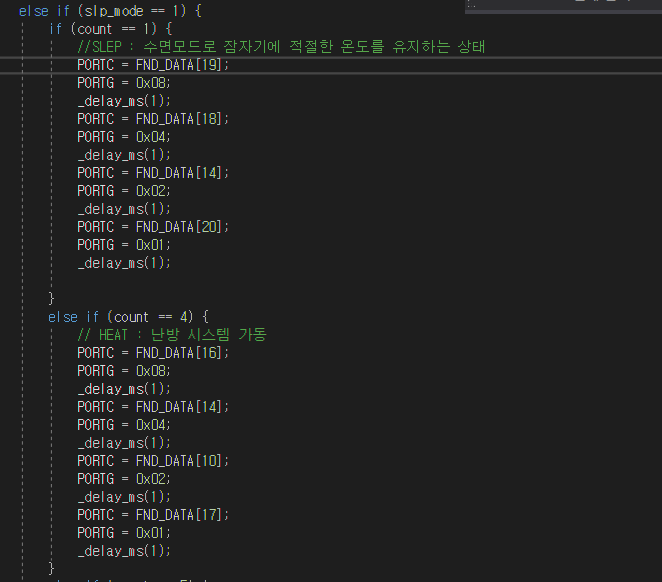
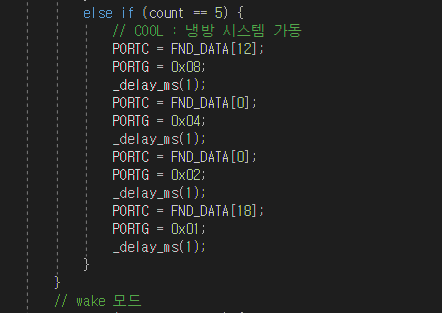
마지막으로 FND에 디스플레이 하는 태스크를 확인해보자

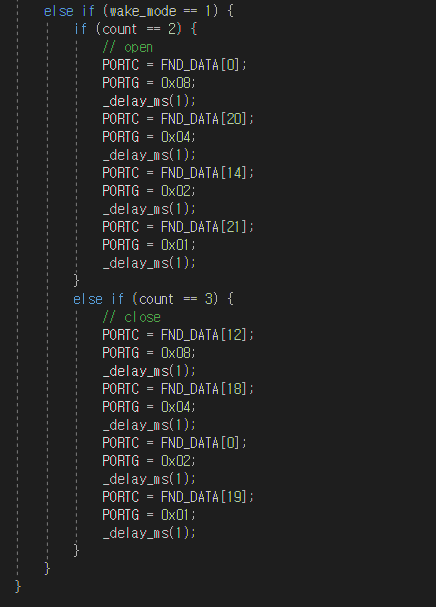


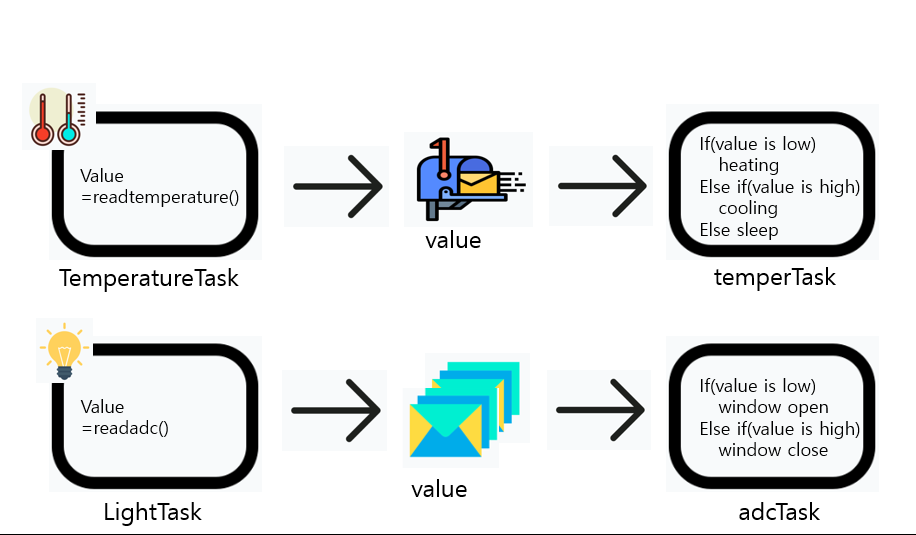
먼저 FND에 출력해야할 문자를 설정하였다. 그리고 FND선택 포트와 FND포트를 출력 포트로 설정하였다.



while문에 들어가게 되면 cur\_state을 받아와야 한다. **그래서 세마포어를 이용해 cur\_state을 보호한다**. 그리고 off상태에서는 FND로 off를 띄우게 된다. 다음 스크린샷은 해당 모드에 따른 FND에 출력하는 내용이므로 설명은 생략한다.





해당 프로그램이 진행되는 과정을 간단하게 그림으로 표현하면 위와 같다. 2개의 태스크가 온도를 읽고 온도를 변화시킬지 아닐지를 결정한다. 나머지 2개의 태스크는 광량을 읽은후에 커튼을 치고 조명을 밝힐것인지 아닌지를 결정하고 해당 결정에 따라서 부저를 울리게 된다.. 마지막으로 출력하는 태스크는 위 태스크들이 결정한 내용에 따라서 출력을 하게된다.

1. **보완해야 할 점**

첨부된 동영상을 확인하게 되면 출력값을 읽은 후 에 FND에 표현하게 되는데 이 과정에서 민감한 센서와 FND 사이에서 절충할 수 있는 분주비를 찾기가 어려웠다. 그래서 커튼을 치고 걷을시에 원래 계획은 간단한 음악이 울리게 하려했지만 순간적으로 계속 바뀌는 온도로 인해서 제대로 된 음악을 출력할수 없어 간단한 음만 출력하게 하였다. 이 문제를 해결하여서 음이 울릴 때 올바르게 울릴수있도록 해야하는 것이 이 프로젝트에서 보완해야할 점이다.