**임베디드 소프트웨어**

***최종 프로젝트***

12141537 박성배

**목차**

**1. 프로젝트 개요**

1) 개활 환경 및 사용품

--------------------------------------------------------------------------------------------------------------

**2. 기능 및 설명**

--------------------------------------------------------------------------------------------------------------

**3. 실행화면 및 출력 화면**

--------------------------------------------------------------------------------------------------------------

**4. 소스 코드**

1) 변수

--------------------------------------------------------------------------------------------------------------

2) 모듈

--------------------------------------------------------------------------------------------------------------

3) 코드 설명

--------------------------------------------------------------------------------------------------------------

**프로젝트 개요**

**사용 모듈 및 개발 환경**

자유과제에서는 MicroC/OS-II에서 중요하게 다뤘던 Real\_Time 스케쥴링 및 MailBox,Messeage Queue, 세마포어, flag, ISR 기능들을 주로 사용 하였습니다 .

J-kit-128의 여러 센서와 레지스터중 FND,Led, Buzzer, Swicth와 INT4,INT5,TIMER2 모듈을 사용하였습니다.

여러 기능들이 동시에 사용될 수 있기 때문에 각 task 마다 세마포어나 event flag를 활용해

여러 task가 한번에 Critical section에 진입하는 것을 방지 하엿습니다.

Led, Test,buzzer,buzzer1,Fnd 의 5가지 task를 구현하였습니다.

**개발환경**: Window10, AvrStudio4, VisualStudio2019

**프로젝트 주제**

***<시간보다 빠르게 led 켜기>게임***

작동 순서 및 게임 Rule

1. 광센서를 이용하여 TEST 1,2,3 중 하나를 선택 할 수 있습니다. 밝기에 따라 test 의 난이도가 바뀝니다.

2. 원하는 test를 스위치버튼을 통해 시작합니다.

3. Test1->Test3순으로 난이도가 증가합니다. ( 주어진 시간이 짧아집니다)

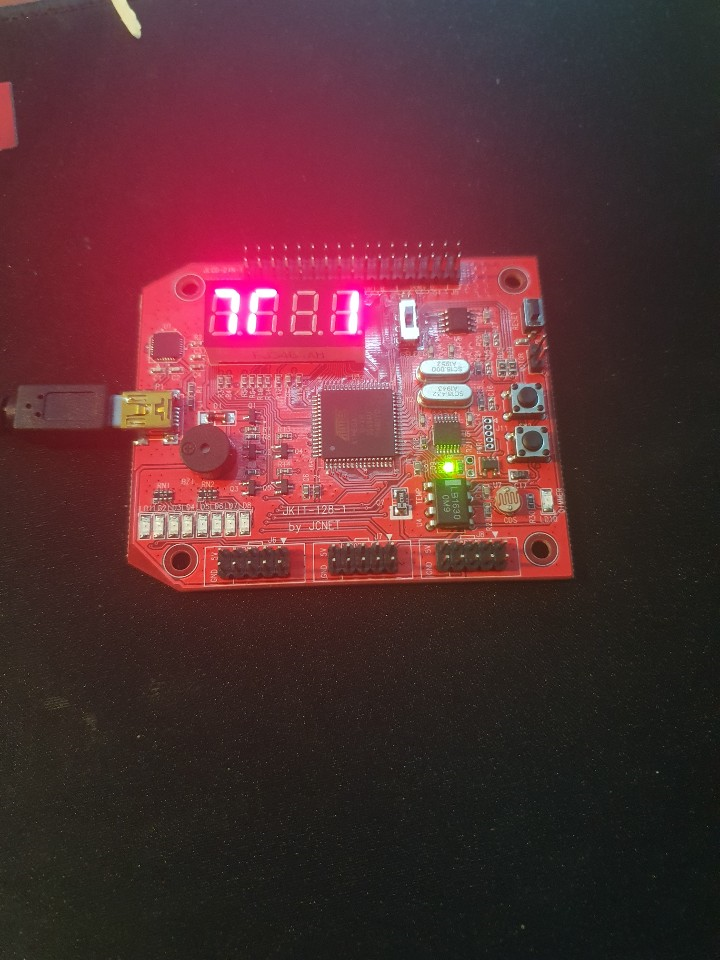
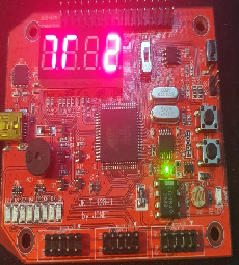
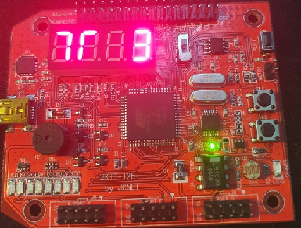
4. 스위치 버튼을 통해서 led를 점등 할 수 있고 8개 led 모두 점등시 승리 아니면 패배합니다.

1) 승리시 -> Win 을 FND에 표기하고 (아빠힘내세요->동요 출력)

2) 패배시 -> LOSE 를 FND에 표기하고 ( 산토끼->동요 출력)

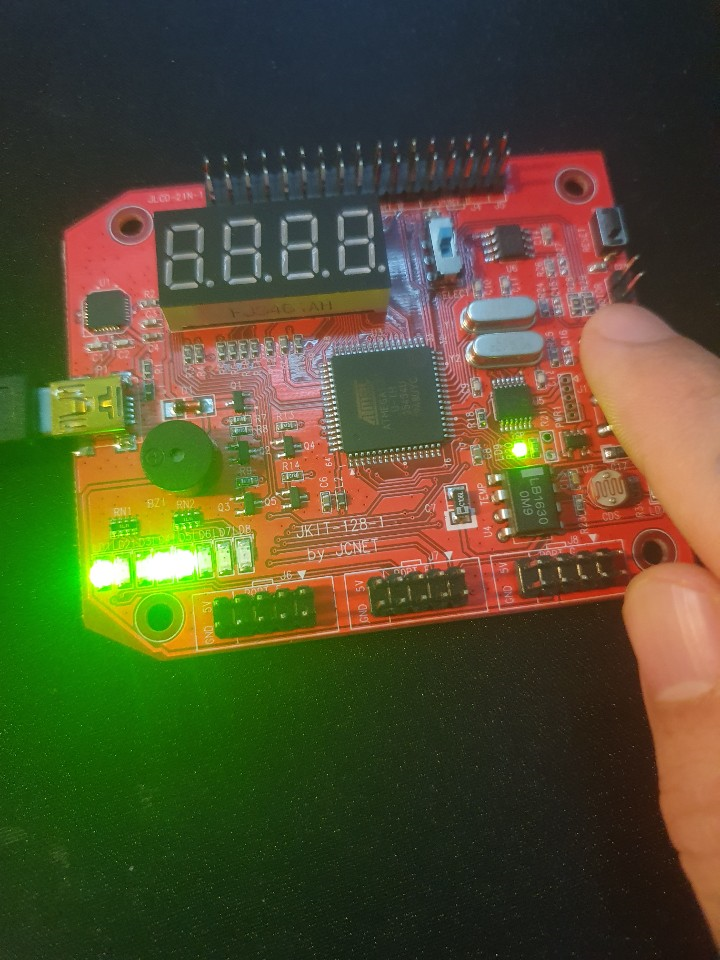
5.노래가 끝나면 다시 광센서를 통해 Test를 선택 할 수 있습니다.

**2. 실행과정**



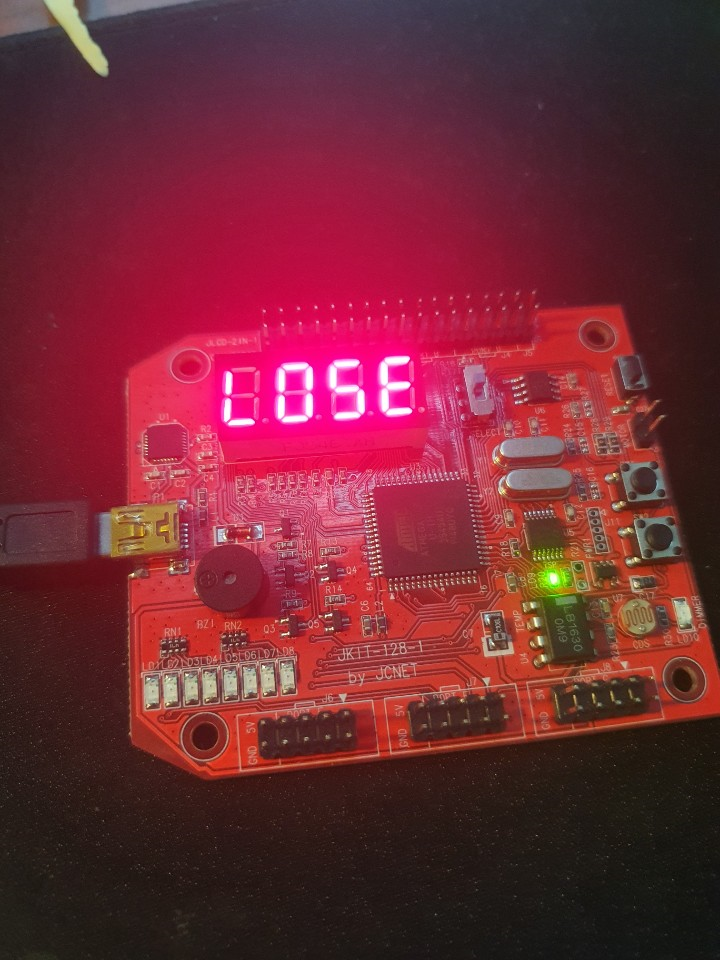
1.시작화면 -> 광센서에 의해서 T 1,2,3이 표기 됩니다.

Switch2버튼을 클릭해서 게임을 시작합니다 .



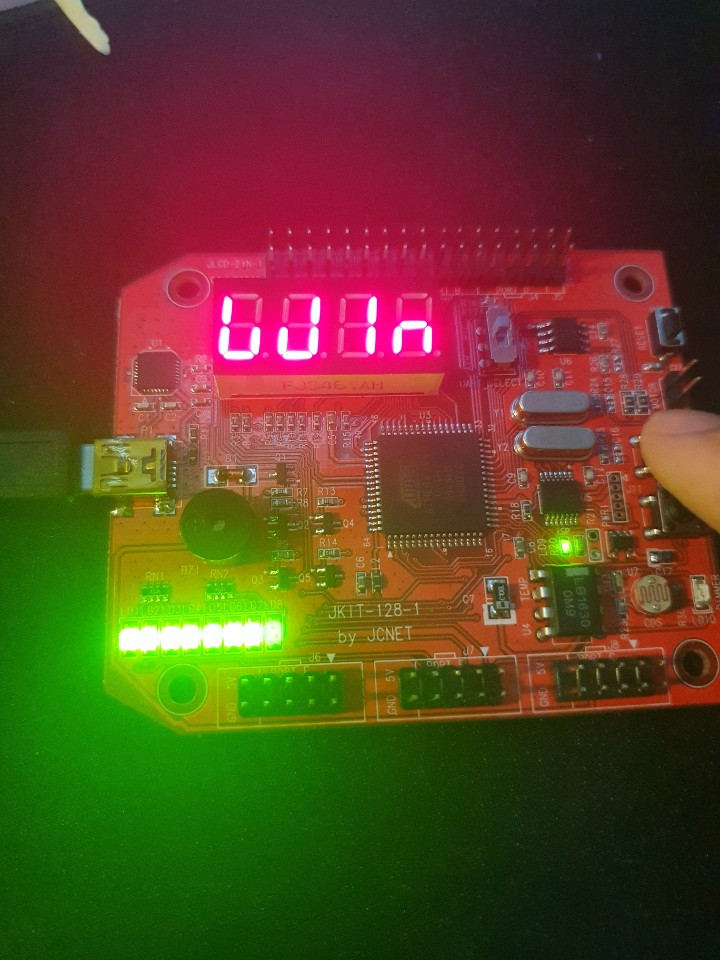
2 Switch 1을 클릭해서 led를 왼쪽부터 점등합니다.

시간안에 led 를 키지 못했다면

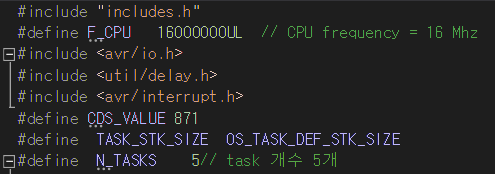
.

FND에 LOSE를 표기하고 아빠힘내세요~ 노래를 출력합니다.

3. 모두 점등 선고시 Win를 fnd에 표기 하고 산토끼~ 노래를 출력합니다



**코드 설명**



Includes 헤더파일

Avr/io.h -> 입출력 제어

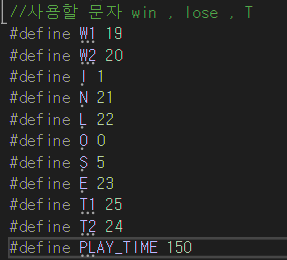
Util/delay.h-> task간 딜레이를 주어 우선순위가 작아도 실행가능 하게 해준다

Avr/interrupt.h-> ISR를 사용 할 수 있다.

N\_TASKS 5 -> 5개의 함수를 task를 사용할 것이기 때문에 5로 선언했습니다.

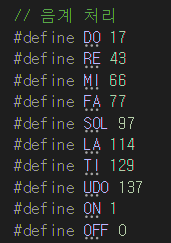
F\_CPU-> cpu진동수 16MHZ 설정

CDS\_Value ->광센서 인식하는 빛의 세기 871를 초기값으로 세팅하엿습니다.

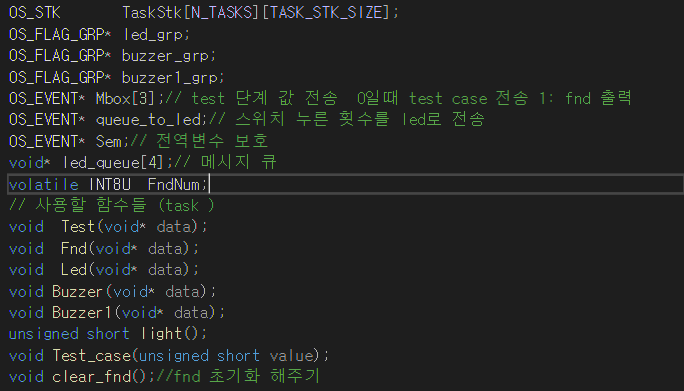


사용할 문자 LOSE, Win, T 를 세팅하였습니다

Play\_time -> Test마다 수행시간을 다르게 하기위해 초기값을 150으로 설정하였습니다.



사용할 음계와 ON/OFF값을 설정하엿습니다.



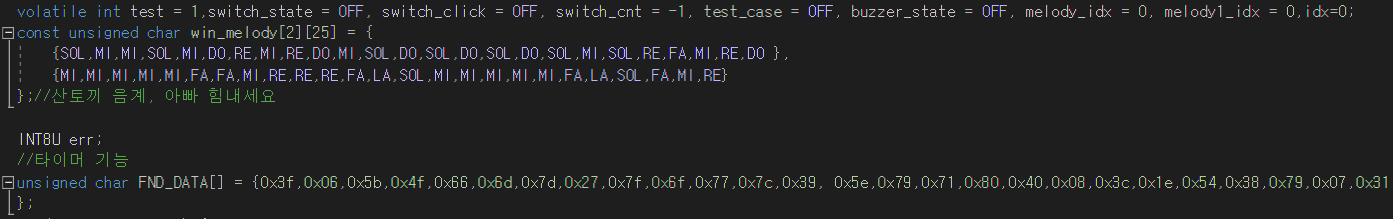
Flag로 사용할 led\_grp, buzzer\_grp,buzzer1\_grp

OS\_EVENT로 Mbox[3] -> T1,T2,T3 3가지 상황이있기 때문에 3으로 선언하였습니다.

Led\_queue[4]-> 메시지큐에서 사용할 배열

Test(), Fnd(),Led(),Buzzer,Buzzer1,light(),Test\_case(), clear\_fnd() 함수를 prototype으로 선언하였습니다.

각 함수들의 로직 및 구현 내용들은 아래에서 설명하겠습니다.



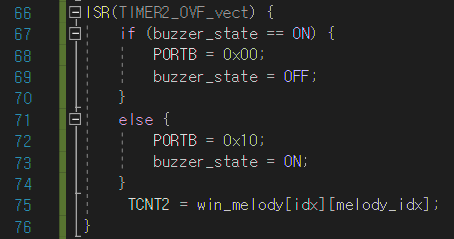
그 외 사용할 변수들을 초기값으로 설정하엿습니다. volatile를 사용한 이유는 컴파일러가

컴파일과정에서 멋대로 최적화를 시킬 수 있기 때문에 선언한 변수들을 건드리지 않도록 volatile로 선언하였습니다.

Win\_melody[2][25]-> 산토끼[25], 아빠힘내세요[25]의 배열을 2차원 배열로 선언하여 이겻을때와

졌을때 각각 다른 노래를 출력하도록 설정 하엿습니다.

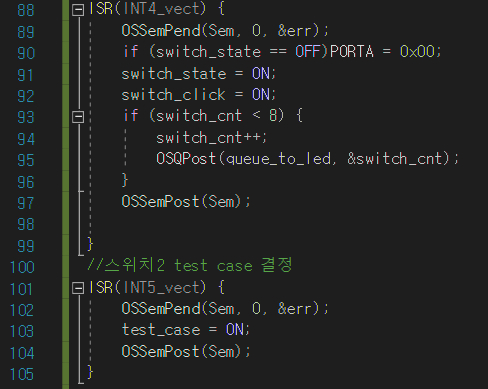
FND\_DATA[26]-> 위에서 선언한 Win,LOSE, T 와 1~9의 값들을 출력 할 수 있도록 배열로 FND출력값을 선언 하였습니다.



**TIMER ISR**

2번 타이머에 interrupt 발생시 buzzer=ON -> OFF OFF->ON으로 세팅하고

TCNT2의 값을 win\_melody[idx][melody\_idx]로 설정 하였습니다.



INT4는 SW1과 연결되어 있고 스위치 상태가 OFF(Test상태)라면 led를 전부 끄고 ON으로 바꿔주고 스위치를 몇 번 눌렀는지를 체크하고 cnt가 8보다 작다면 cnt를 증가 시켜줍니다.

그리고 메시지 큐를 통해서 값을 전달 합니다.

스위치 2를 통해서 test1,2,3 중 선택한 것을 시행합니다.



**Main()**

각 register 값들을 초기화 시켜줍니다 .

DDRA-> Led 출력 포트 , DDRB-> 버저 출력포트 DDRE-> SW1,SW2 스위치를 사용합니다.

EICRB,EIMSK-> 인터럽트 설정

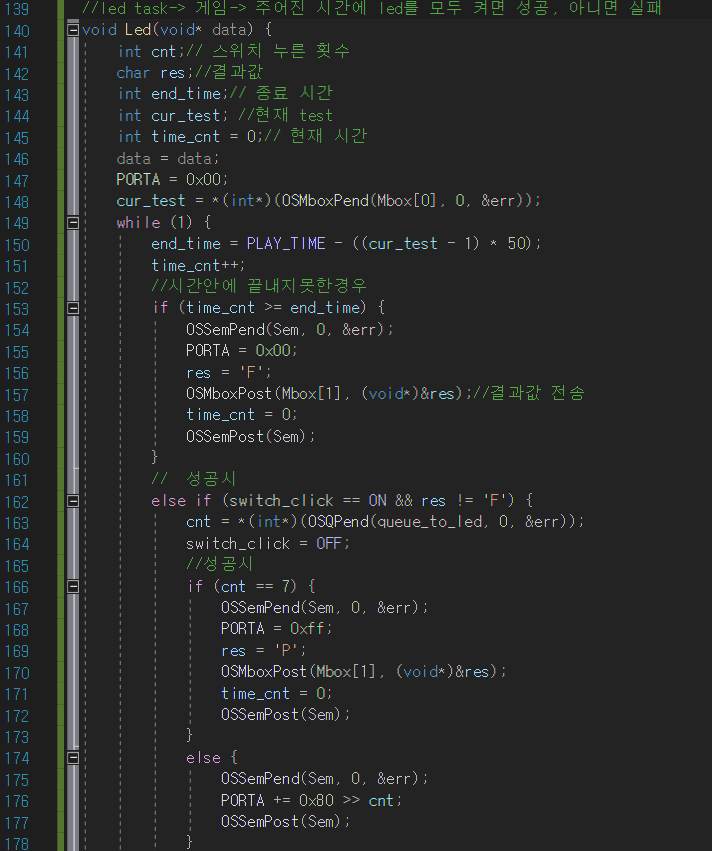
SREG->전체 인터럽트 허용

ADMUX-> 5V전압 사용, ADC0 사용

TCNT2: 타이머/count 기능입니다 (그중에서 2번 타이머 설정)

Mbox, 메시지 큐, 이벤트 Flag, 세마포어 선언

5가지 Task 생성



**Led()** 함수입니다.

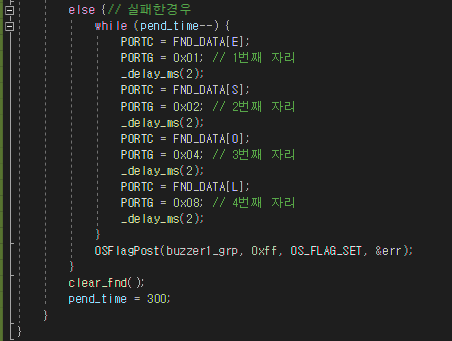
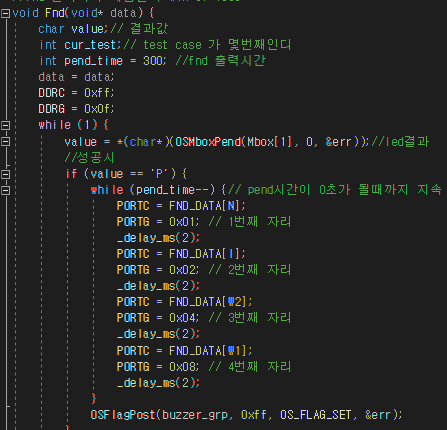
Mbox로부터 값을 받을 때까지 Pend 후 값을 받으면

Test1,2,3에 따라 Play\_time 값을 변환 시킵니다 150->100->50 순서로 시간이 짧아지게 됩니다.

Time\_cnt>=end\_time 즉 주어진 시간안에 끝내지 못한 경우 led를 모두 끄고 res-> F 값을 넘겨줍니다

Res 값이 L 이 아니고 스위치가 ON일 경우 큐에 led를 누른 횟수를 받을때까지 Pend합니다

받고나면 P값을 Mbox를 통해서 post 합니다 .



**FND()** 함수입니다.

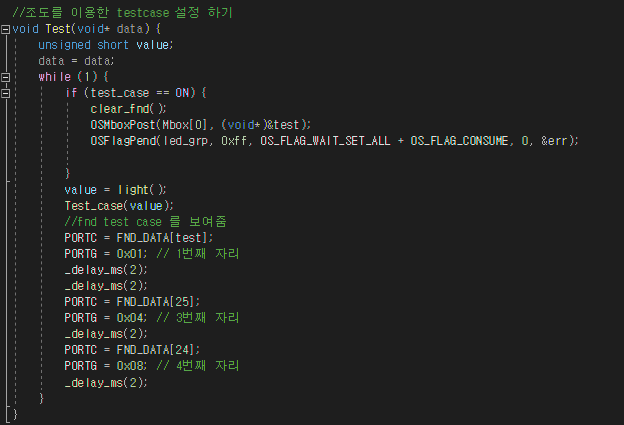
OSMboxPend()를 통해서 Mbox로부터 P, F의 결과 값을 받아 옵니다.

Value=P 라면 Win을 출력하고 Flag로 Buzzer\_grp를 보냅니다 .

Value=F 라면 LOSE를 출력하고 Flag로 Buzzer1\_grp를 보냅니다 .

Buzzer\_grp의 경우 산토끼~ 노래를 출력하는 플래그가 됩니다.

Buzzer1\_grp의 경우 아빠힘내세요~노래를 출력하는 플래그가 됩니다.



Test()

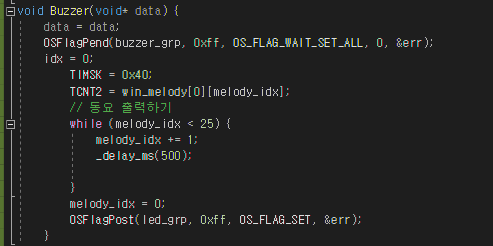
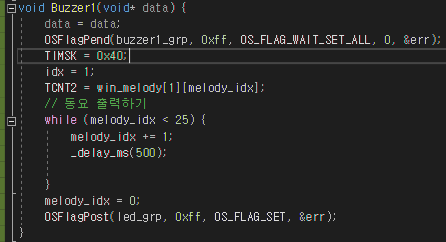
1,2,3 중에 어떤 Test를 할지 결정 합니다.

결정된 test를 mbox로 post 해줍니다.

OS\_Event\_Flag인 Led\_grp에서 flag가 올때까지 pend 합니다.

레벨이 결정되었다면 light() 함수를 호출 하게 됩니다.

FND에 T1, T2,T3를 출력합니다



Buzzer1 과 Buzzer 함수입니다 .

버저1은 게임에서 실패했을시 즉 F 일때 flag를 보내고 버저1에서 flagpend를 통해

대기합니다.

TCNT2=melody[1][melody1\_idx]로 하고 아빠 힘내세요~ 를 melody1\_idx 값을 하나씩 더해가면서

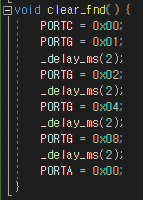
25번째 배열까지 출력합니다.

노래가 끝나면 led\_grp에 event를 post 합니다

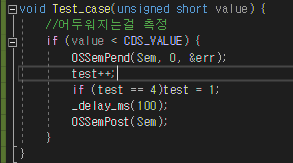
버저는 승리 했을떄 즉 P 일때 flag 를 받을 수 있습니다 .

이번엔 TCNT2=melody[0][melody\_idx]로 배열을 설정하고 산토끼~를 출력합니다 . 방식은 버저1과 같습니다.

노래가 끝나면 led-grp에 event를 post 합니다.

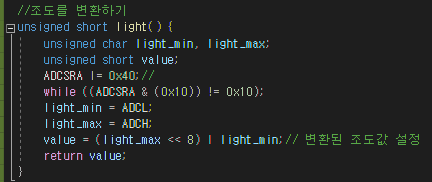


FND 화면에 출력된 값들을 초기화 해줍니다.



광센서를 이용하여 밝기를 측정합니다 가장 어두울때가 3이며 3보다 커지면 다시 1로 초기화

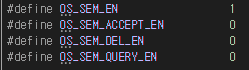
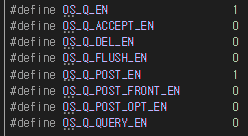
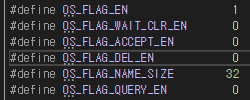
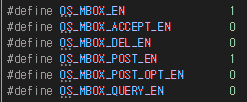
합니다.



조도값을 digital 값으로 변환시켜주는 역할을 합니다.

그 외 수정 내용

OS\_cfg.h



Event,Flag, Mbox,Sem,Q 의 값들을 수정해주었습니다.

PS. 그 외 영상파일은 따로 첨부해두었습니다