

[임베디드 시스템]

팀프로젝트 보고서

:SMOKE DETECTOR

강의명 : 임베디드시스템[059]

담당교수 : 백윤주

조원

정보컴퓨터공학부 201824523 안혜준

물리학과 201812114 김유빈

[목차]

I. 개요

- 사전지식
- 사용 디바이스
- RTOS개요

II. 프로젝트 구현

- 사전작업
- task
 1. AppTask_smoke
 2. AppTask_process
 3. AppTask_action

III. 결론 및 토의

- 역할분담
- 참고(시연영상 링크포함)

I. 개요

현대 사회에서는 많은 요인들이 사회 갈등의 원인이 되고 있다. 그 중에서도 꾸준히 논란의 불씨가 되어온 흡연자와 비흡연자의 갈등은 제도적 노력에도 불구하고 여전히 사그라들지 않고 있다. 최근에도 금연구역으로 지정된 공공장소에서의 흡연으로 생기는 이웃 간 불화가 언론에 언급되고 있다. 또한 무분별한 흡연은 종종 산불 및 화재의 원인이 되기도 한다.

이를 해결하기 위해 정부 및 지자체 등은 흡연 구역을 만들어 이러한 문제를 방지하려고 노력했지만 흡연 장소의 특성상 지속적인 단속이 시행되기 어렵기 때문에 이러한 갈등은 좀처럼 잦아들 기미가 보이지 않고 있다.

이번 프로젝트인 'smoke detector' 이러한 문제점을 보완하는 데에 주력한다.

smoke detector는 흡연을 감지하고 경보음을 울림으로써 허가되지 않은 구역에서의 흡연을 지양시키고 흡연자가 자발적인 흡연구역을 준수하도록 하여 이웃 간의 갈등 해소와 담배와 관련된 화재 문제에 예방책을 제시하는 것을 목적으로 제작되었다.

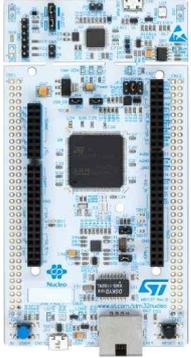
<사전지식>

-RTOS(RealTime Operating System)

실시간 응용 프로그램을 위해 개발된 운영체제로 높은 처리율과 실시간 성능보장을 목적으로 사용되며 이와 같은 특성 때문에 real time 반응이 필요한 임베디드 프로젝트에 적합한 os이다.

이번 과제는 이러한 RTOS 중 μ C/OS-3를 사용하여 그동안 학습한 내용을 종합하고 응용 및 점검하며 프로젝트를 제작한다.

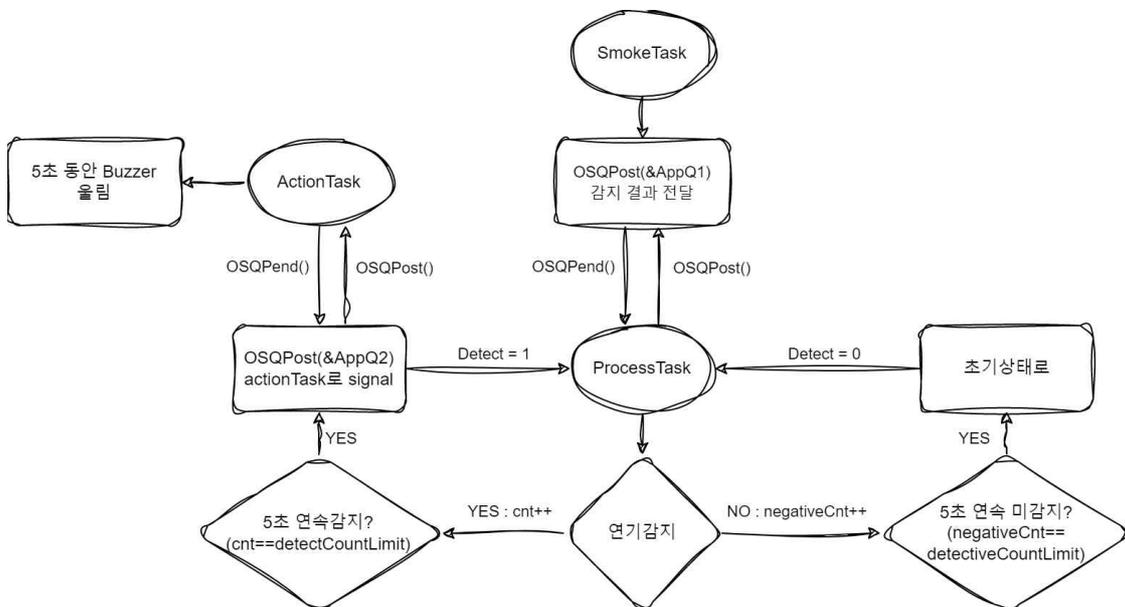
<사용 디바이스>

디바이스	설명 및 사용법	사진
NUCLEO-F429ZI	<ul style="list-style-type: none"> : μC/OS-3를 탑재한 보드 : usb cable로 전원인가 + J-link연결 : true studio로 코딩한 프로젝트 동작 	
MQ-2 센서	<ul style="list-style-type: none"> : 접촉 연소식 방식을 통해 연기, 가스 (메탄, 부탄) 검출 -> 가스 유입 시 저항값 감소로 전류량 증가 : 담배 연기에 포함된 일산화탄소 감지 <p>vcc : 5v gnd : gnd</p>	

	DO : PG9(D0) : 전압 인가시 경보음 출력	
능동부저모듈 (mh-fmd)	vcc : 5v gnd : gnd I/O : PG3(D50)	

기타 부속품 : 점퍼선, bread board, 5pin cable

II. 프로젝트 구현



이번 프로젝트는 위의 흐름으로 동작하며, 총 3개의 Task와 2개의 message Queue로 이루어져있다. Task는 역할에 따라 흡연을 감지하는 smokeTask, 센싱한 값을 처리하는 processTask, 처리 결과를 출력하는 actionTask로 구성된다.

메시지큐는 Task간 interaction을 위해 사용되며 센싱한 값을 smokeTask->processTask로 넘길때는 AppQ1을, 처리결과를 processTask->actionTask로 넘길때는 AppQ2를 사용한다.

각 작업에 대한 세부 사항은 아래와 같다.

[사전 작업]

GPIO : 외부 I/O 장치를 사용하기 위한 설정

메세지 큐 : Task간 interaction을 위한 메세지 큐(AppQ1 , AppQ2) 생성 및 설정

[TASK]

1. AppTask_smoke

sensingMs(500ms) 간격으로 mq센서를 읽고 그 값을 messageQueue(AppQ1)를 사용해 post한다(OSQPost()).

```
230=static void AppTask_smoke(void *p_arg)
231 {
232     OS_ERR err;
233
234     while (DEF_TRUE) {
235         uint8_t input = ReadMQ2();
236         OSQPost((OS_Q *)&AppQ1,
237             (void*)&input,
238             (OS_MSG_SIZE)sizeof(void *),
239             (OS_OPT )OS_OPT_POST_FIFO,
240             (OS_ERR *)&err);
241
242         OSTimeDlyHMSM(0u, 0u, 0u, sensingMs,
243             OS_OPT_TIME_HMSM_STRICT,
244             &err);
245
246     }
247 }
```

post한 메시지는 AppTask_Process의 pend를 해소시켜 동작을 trigger 한다.

2. AppTask_process

센싱한 값을 처리하는 task이며 AppTask_smoke에서 post한 메시지를 받으면 pend가 해소되어 동작을 시작한다.

```
336     uint8_t detect = 0;
337     while (DEF_TRUE) {
338
339         p_msg = OSQPend((OS_Q *)&AppQ1,
340             (OS_TICK )0,
341             (OS_OPT )OS_OPT_PEND_BLOCKING,
```

MQ-2 센서는 연기를 감지했을 때 0을 출력하므로 p_msg가 0이라면 count값을 증가시켜 count가 detectCountLimit에 도달하면 (5초 동안 감지되면) messageQueue (AppQ2)를 사용하여 AppTask_Action에 메시지를 post하고 동작을 trigger한다.

```
374         if (cnt == detectCountLimit) {
375             detect = 1;
376             cnt = 0;
377             OSQPost((OS_Q *)&AppQ2,
378                 (void*)&detect,
```

5초가 되기 전에 연기가 더 이상 감지되지 않는다면 흡연 중이 아닌 것으로 판단하여 count 값을 초기화하고 다시 연기가 감지되기를 기다린다.

3. AppTask_action

부저 출력을 관리하는 task이다. AppTask_process와 AppQ2로 연결되어 AppTask_process의 post를 통해 동작한다.

pend가 해소되면 부저를 on상태로 바꾼후(BuzzerOn()) 5초간 delay하고 5초 뒤에 부저를 off 상태로 변경(BuzzerOff())하여 5초 동안 부저를 울리도록 설계하였다.

```

262= static void AppTask_action(void *p_arg)
263 {
264     OS_ERR err;
265     void *p_msg;
266     OS_MSG_SIZE msg_size;
267     CPU_TS ts;
268
269     while (DEF_TRUE) {
270         p_msg = OSQPend((OS_Q *)&AppQ2,
271                        (OS_TICK )0,
272                        (OS_OPT )OS_OPT_PEND_BLOCKING,
273                        (OS_MSG_SIZE *)&msg_size,
274                        (CPU_TS *)&ts,
275                        (OS_ERR *)&err);
276         send_string("\n\r-----BIIPPPPPPPP!!!-
277         BuzzerOn();
278         OSTimeDlyHMSM(0u, 0u, 5u, 0u,
279                      OS_OPT_TIME_HMSM_STRICT,
280                      &err);
281         BuzzerOff();
282     }
283 }

```

4. 기타 함수

ReadMQ2 : mq센서 값을 읽어서 반환

BuzzerOn/Off : buzzer on/off기능

III. 결론 및 토의

이번 프로젝트(smoke detector)에서는 금연구역에서 흡연을 저지하기 위해 담배 연기를 감지하고 경보음을 울리는 담배연기 감지기를 제작하였다.

사용한 task는 총 3개로 각각 input(AppTask_smoke), data처리(AppTask_process), output(AppTask_action)의 역할을 수행한다.

AppTask_smoke에서 0.5초 간격으로 연기를 센싱해 AppTask_process로 보내 데이터를 처리하고 5초 동안 연기가 감지되었다면 AppTask_process에서 AppTask_action으로 신호를 보내 부저가 울리게 된다.

담배 연기에 대한 sensing은 mq2 센서를 사용하였고 출력은 USART와 능동 부저를 사용하였다.

이번 프로젝트는 실내에서는 예상한대로 잘 동작했지만, 야외에서 동작시키면 날씨의 영향(특히 바람)을 크게 받았다. 시연영상에서도 나타나는 것과 같이 바람이 많이 부는 경우에는 연기가 흩어져 잘 감지되지 않았다. 따라서 트인 야외보다는 실내나 바람이 불지 않는 야외에서 사용하는 것이 적절할 것이다. 또 먼 거리에서는 센싱이 잘 되지 않았고 정밀한 센싱이 어려웠는데 센서를 일정 간격으로 여러개 달거나 일산화탄소 감지에 더 뛰어난 mq7등을 사용하면으로써 해결할 수 있을 것이다. 공기보다 가볍고 퍼지는 일산화탄소의 특징을 이용하여 센서를 사람의 신장보다 약간 높은 곳에 다는 것도 도움이 될 것이라고 생각한다.

이번 프로젝트를 통해 금연구역에서 흡연자 감소, 흡연 구역의 활성화, 이웃 간의 갈등 해소 등을 기대하고 있으며 와이파이 모듈 등을 더 달아 흡연 정보 수집하여 해당 구역의 흡연 데

이터를 얻는 방향으로 발전시킬 수도 있을 것이라 생각한다.

[역할분담]

소속	학번	이름	역할
정보컴퓨터공학부	201824523	안혜준	appTask_action appTask_process report작성
물리학과	201812114	김유빈	gpio init 및 sensor sensing appTask_smoke report작성

[user-Interface(SERIAL 통신)]

```
##### # # ##### # # #####
# ## ## # # # #
# # # # # # # # #
##### # # # # # ## #####
# # # # # # # #
# # # # # # # #
##### # # ##### # # #####

##### ##### ##### ##### ##### ##### #####
# # # # # # # # # # #
# # # # # # # # # # #
# # ##### # ##### # ##### #####
# # # # # # # # # # #
# # # # # # # # # # #
##### ##### # ##### ##### # ##### # #

* If smoke is detected for 5 seconds, BIIPPPPPPPP!!! *

Now state: not smoking (0/10)█
```

[참고]

project git : <https://github.com/hyejunn/smoke-detector>

시연영상 : <https://youtu.be/fxReEu6GE9A>