

## 1. 생성한 커널 객체

### (1) ADC mutex

온도 센서, 습도 센서, 조도 센서가 하나의 ADC를 공유하기 위해 사용함. 어떤 센서가 ADC를 설정하고, ADC로 부터 오는 인터럽트를 기다리고 있을 때. 다른 센서가 ADC 설정값을 건드릴 수 없도록 하기 위해서 필요하다.

### (2) TempQ message queue

통계 task에게 측정한 온도 값을 보내기 위해 필요

### (3) HumidQ message queue

통계 task에게 측정한 습도 값을 보내기 위해 필요

### (4) PhotoQ message queue

통계 task에게 측정한 조도 센서값을 보내기 위해 필요

### (5) OS\_MEM

통계 task에게 실수 값을 보낼 때, 블루투스 task에게 전송할 문자열을 보낼 때 필요하다. Message queue는 포인터밖에 전달할 수 없으므로, OS\_MEM에서 메모리를 할당 받아 해당 메모리에 값을 저장한 후, 포인터를 전달한다.

이 OS\_MEM 객체에 할당되는 메모리 파티션은 CPU\_INT08U[60][200], 즉 한 블록의 크기는 200바이트, 총 블록의 개수는 60개이다.

## 2. 생성한 Task

### (1) START task

우선순위: 2

스택 크기: 128\*4 바이트

Task Message Queue: 5개

Start task는 main에서 처음 생성되는 task이다. UART를 통해 온도 임계값, 습도 임계값, 조도센서 임계값을 받아 전역변수에 저장하고, 다른 task와 커널 객체를 만든다 (OS\_MEM 제외.)

이때  $\mu$ C/OS-III에서 제공하는 Board Support Package에 정의되어 있는 BSP\_Ser\_RdStr()이라는 함수를 사용하는데 이 함수는 세마포어를 획득하여 다른 task가 문자열을 받을 수 없도록 한 후, UART에서 RXNE(RX Not Empty) 인터럽트가 발생하기를 기다리고, 발생하여 ISR이 문자를 보내주면 문자를 지정한 배열에 저장해주는

역할을 한다( $W_n$  또는  $W_r$ 을 만나면 종료로 가정)

## (2) Temper task

우선순위: 3

스택 크기:  $128 \times 4$  바이트

Task Message queue: 10개

측정한 온도값을 처리하는 task이다. 먼저 ADC mutex로 부터 ADC 접근을 허가받으면, 온도센서가 연결되어 있는 핀에 맞추어 ADC를 설정한 후, ADC ISR로부터 Task message queue를 통해 값이 오기를 기다린다. 값이 오면, mutex를 post하고, 받은 값을 섭씨 온도로 변환시킨다. 그리고 온도값 정보를 온도값이라는 표시와 함께 문자열로 변환시켜 bluetooth task에 보낸다. 그리고 온도값 정보를 TempQ를 통해 통계 task로 보낸다. 마지막으로, 온도가 임계값보다 작으면, bluetooth task에게 온도가 낮다는 경고 문자열을 긴급하게 보내고(LIFO), Heater task에게 신호를 보낸다(만약 Heater task가 이미 작동중이라면, 신호를 보내지 않는다. Heater task는 작동이 끝나면 Temper task에 신호를 보내 더이상 작동되지 않음을 알린다).

## (3) SoilHumid Task

우선순위: 4

스택 크기:  $128 \times 4$  바이트

Task Message queue: 10개

측정한 토양습도값을 처리하는 task이다. 먼저 ADC mutex로 부터 ADC 접근을 허가받으면, 토양습도센서가 연결되어 있는 핀에 맞추어 ADC를 설정한 후, ADC ISR로부터 Task message queue를 통해 값이 오기를 기다린다. 값이 오면, mutex를 post하고, 받은 값을 습도값으로 변환시킨다. 그리고 습도값 정보를 습도값이라는 표시와 함께 문자열로 변환시켜 bluetooth task에 보낸다. 그리고 습도값 정보를 HumidQ를 통해 통계 task로 보낸다. 마지막으로, 습도가 임계값보다 작으면, bluetooth task에게 습도가 낮다는 경고 문자열을 긴급하게 보내고(LIFO), pump task에게 신호를 보낸다(만약 pump task가 이미 작동중이라면 신호를 보내지 않는다. Pump task는 작동이 끝나면 SoilHumid task에 신호를 보내 더이상 작동되지 않음을 알린다).

## (4) Photoresistor Task

우선순위: 5

스택 크기:  $128 \times 4$  바이트

Task Message queue: 10개

측정한 조도센서값을 처리하는 task이다. 먼저 ADC mutex로 부터 ADC 접근을 허가 받으면, 조도센서가 연결되어 있는 핀에 맞추어 ADC를 설정한 후, ADC ISR로부터 Task message queue를 통해 값이 오기를 기다린다. 값이 오면, mutex를 post하고, 받은 조도값 정보를 조도값이라는 표시와 함께 문자열로 변환시켜 bluetooth task에 보낸다. 그리고 조도값 정보를 PhotoQ를 통해 통계 task로 보낸다. 마지막으로, 조도값이 임계값보다 크면, bluetooth task에게 조도가 낮다는 경고 문자열을 긴급하게 보내고(LIFO), light task에게 신호를 보낸다(만약 light task가 이미 작동중이라면 신호를 보내지 않는다. light task는 작동이 끝나면 Photoresistor task에 신호를 보내 더이상 작동되지 않음을 알린다).

#### (5) 통계 task

우선순위: 6

스택 크기: 128\*4 바이트

Task Message queue: 0개

센서값 통계 task이다. TempQ, HumidQ, PhotoQ를 OS\_OPT\_PEND\_NON\_BLOCKING으로 기다려 온도, 습도, 조도 값이 왔는지 확인한다. 만약 도착했다면 지역변수 ???sum에 더하고, 카운트를 증가시킨다. 온도, 습도, 조도 값이 모두 추가로 5개씩 모일 때마다 평균을 내어 평균값 정보를 bluetooth task로 보낸다.

#### (6) Bluetooth task

우선순위: 2

스택 크기: 128\*4 바이트

Task Message queue: 30개

각종 task에서 받은 메시지를 bluetooth로 보내는 task이다. 이때  $\mu$ C/OS-III 에서 제공하는 Board Support Package에 정의되어 있는 BSP\_Ser\_WrStr()를 이용한다. 이 함수는 세마포어를 획득하여 다른 task가 UART로 메시지를 보낼 수 없도록한 후 한 문자씩 보내며 UART에서 TC(Transmit Complete) 인터럽트가 발생할 때까지 기다리고, 발생하면 다음 문자를 보내며, 다 보내면 세마포어를 post하는 함수이다.

#### (7) Heater Task

우선순위: 2

스택 크기: 128\*4 바이트

Task Message queue: 0개

Heater task가 시작되면, heater task가 실행되고 있지 않다고 Temper Task에 신호를 보낸다. 그리고 Temper Task로부터 신호가 오기를 기다리고, 신호가 오면, 히터와 연결되어있는 릴레이 모듈에 신호를 보내 히터를 동작시키게 한다. 약 10초 후에 히터를 끄고 히터가 꺼졌다고 Temper Task에 신호를 보내게 된다.

#### (8) Pump Task

우선순위: 2

스택 크기: 128\*4 바이트

Task Message queue: 0개

Pump task가 시작되면, pump task가 실행되고 있지 않다고 SoilHumid task에 신호를 보낸다. 그리고 soilhumid Task로부터 신호가 오기를 기다리고, 신호가 오면, 펌프와 연결되어있는 릴레이 모듈에 신호를 보내 펌프를 동작시키게 한다. 약 10초 후에 히터를 끄고 펌프가 꺼졌다고 SoilHumid Task에 신호를 보내게 된다.

#### (9) Light Task

우선순위: 2

스택 크기: 128\*4 바이트

Task Message queue: 0개

Light Task가 시작되면, Light Task가 실행되고 있지 않다고 Photoresistor task에 신호를 보낸다. 그리고 photoresistor Task로부터 신호가 오기를 기다리고, 신호가 오면, LED를 켜다. 약 10초 후에 LED를 끄고 LED가 꺼졌다고 photoresistor task에 신호를 보내게 된다.

### 3. ISR handler

#### (1) ADC ISR handler

ADC로 부터 인터럽트가 왔을 때 실행할 ISR handler이다. 온도센서로부터 왔다면 온도센서값을 Temper Task로 보내고, 토양습도 센서로부터 왔다면 토양습도 값을 SoilHumid Task로 보내고, 조도센서로부터 왔다면 조도 센서값을 PhotoResistor Task로 보낸다.

### 4. 사용한 OS 기능

#### (1) Task Management

센서값 측정, 히터 작동, 통계 등 여러 task를 만들어 multitasking을 하도록 하여 CPU 사용량을 극대화하고, 복잡한 하나의 task를 여러 개의 task로 분산시켜 설계 및 유지보수에 용이했다.

## (2) Interrupt Management

ADC를 통해 센서값 변환이 완료되었을 때, 전송이 완료되었을 때, 블루투스로부터 값을 받았을 때, 버튼이 눌렸을 때 인터럽트를 발생시켜 이벤트에 실시간으로 대응할 수 있게 한다.

## (3) Resource management

하나의 ADC를 3개의 센서가 나누어 쓰기 위하여 mutex를 사용하였다.

## (4) Synchronization

온도, 습도, 조도 값에 문제가 생겼을 때 해당하는 task에 신호를 보내어 히터를 켜거나, 펌프를 작동시키거나, LED를 켜는 등의 대응을 하게 한다.

## (5) Message Passing

Bluetooth task에게 블루투스로 전송할 메시지를, 통계 task에게 센서 값을 전달할 때 필요하다.

## (6) Memory Management

메시지를 전달할 때, 메시지를 담을 메모리를 얻기 위해서 필요하다.