

임베디드 시스템

Term Project 보고서

4조 스마트 화분

201924403 강수빈

201924539 이영인

201924490 서경우

목차

1. 임베디드시스템 / RTOS

2. 개발 목적

3. 세부 기능

4. 사용 센서

5. 기능 구현 설명

6. 역할 분담

7. 결론 및 기대효과

1. 임베디드시스템 / RTOS

: 임베디드시스템이란 프로그램 할 수 있는 컴퓨터가 큰 시스템의 부품으로 사용되는 것을 말하며 비행기, 기차, 다양한 디지털 기기 등 여러 분야에 사용되고 있다. RTOS는 Real Time Operating System의 줄임말로 실시간 운영체제를 뜻하며, 주어진 시간 내에 목적을 달성하는 것을 보장하는 운영체제를 말한다. RTOS는 Hard RTOS와 Soft RTOS 두가지로 나뉘는데 Hard RTOS는 어떤 작업을 일정 시간 안에 반드시 처리해야 하며, 그 이후의 결과 값은 소용이 없는 것을 뜻한다. Soft RTOS는 주어진 시간 내에 작업을 완료하지 못하더라도 그 시간에서 약간 후의 결과 값도 인정하는 것을 말한다.

2. 개발 목적

: 토양 습도센서를 활용하여 화분의 상태를 측정하고 LED와 부저를 활용해 토양 습도 상태 이상을 알 수 있는 스마트 화분을 개발한다. 이 시스템을 이용하면 매번 식물의 토양 습도에 신경을 쓰지 않고 습도에 따른 부저의 알람을 통해 토양의 습도를 조절해 주면 되기 때문에 식물을 키우는 사람이라면 유용하게 활용할 수 있다.

3. 세부 기능

- a. 토양 습도센서를 이용해 현재 토양의 습도를 측정
- b. 시리얼 터미널에서 측정된 습도 값 확인
- c. 토양의 습도가 건조하면 3번 LED(RED)가 ON / 부저 ON
- d. 토양의 습도가 습하면 2번 LED(BLUE)가 ON

4. 사용 센서

- a. 토양습도센서



→ 토양의 습도를 측정

- b. 피에조 부저



→ 측정한 토양의 습도가 낮을 경우(건조한 경우) 동작

5. 기능 구현 설명

```
static void AppTask_usart(void *p_arg)
{
    OS_ERR err;
    CPU_INT32U adc_value;
    OS_MSG_SIZE size;
    CPU_TS ts;

    char temp[10];
    while(DEF_TRUE){
        adc_value = (CPU_INT08U)OSQPend((OS_Q *)&ADC_Q,
                                         (OS_TICK)0,
                                         (OS_OPT)OS_OPT_PEND_BLOCKING,
                                         (OS_MSG_SIZE *)&size,
                                         (CPU_TS *)&ts,
                                         (OS_ERR *)&err);

        send_string("\n\r");
        send_string("humidity: ");
        send_string(itoa(adc_value, temp, 10));
        send_string("\n\r");

        OSTimeDlyHMSM(0u, 0u, 0u, 10u,
                      OS_OPT_TIME_HMSM_STRICT,
                      &err);
    }
    send_string("\n\rReceive end \n\r");
}
```

→Usart task

: usart task를 통해 adc_value

(토양습도값)을 serial 모니터에 출력해준다.

(이때 adc_value값은 MessageQueue의

Pend를 사용하여 받아온다.)

```
static void AppTask_ADC(void *p_arg)
{
    OS_ERR err;

    while(DEF_TRUE){
        value = Get_ADC_Converted_Value();

        OSQPost((OS_Q *)&ADC_Q,
                (void *)value,
                (OS_MSG_SIZE)sizeof(void *),
                (OS_OPT)OS_OPT_POST_ALL,
                (OS_ERR *)&err);

        OSTimeDlyHMSM(0u, 0u, 2u, 0u,
                      OS_OPT_TIME_HMSM_STRICT,
                      &err);
    }
}
```

→ADC task

: adc_value값 (토양습도값)을

MessageQueue를 사용하여 Post해준다.

```

static void AppTask_LED(void *p_arg)
{
    OS_ERR err;
    BSP_LED_On(3);
    BSP_LED_On(2);
    CPU_INT08U adc_value;
    OS_MSG_SIZE size;
    CPU_TS ts;

    while (DEF_TRUE){
        adc_value = (CPU_INT08U)OSQPend((OS_Q *)&ADC_Q,
            (OS_TICK)0,
            (OS_OPT)OS_OPT_PEND_BLOCKING,
            (OS_MSG_SIZE *)&size,
            (CPU_TS *)&ts,
            (OS_ERR *)&err);

        if(adc_value > 200){
            BSP_LED_On(3);
            BSP_LED_Off(2);
        }
        else{
            BSP_LED_Off(3);
            BSP_LED_On(2);
        }

        OSTimeDlyHMSM(0u, 0u, 0u, 10u,
            OS_OPT_TIME_HMSM_STRICT,
            &err);
    }
}

static void AppTask_buzzer(void *p_arg)
{
    OS_ERR err;
    CPU_INT08U adc_value;
    OS_MSG_SIZE size;
    CPU_TS ts;
    Buzzer_off();

    while (DEF_TRUE){
        Buzzer_off();
        adc_value = (CPU_INT08U)OSQPend((OS_Q *)&ADC_Q,
            (OS_TICK)0,
            (OS_OPT)OS_OPT_PEND_BLOCKING,
            (OS_MSG_SIZE *)&size,
            (CPU_TS *)&ts,
            (OS_ERR *)&err);

        if(adc_value > 200){
            Buzzer_On();
            //Buzzer_off();
        }
        else{
            Buzzer_off();
        }

        OSTimeDlyHMSM(0u, 0u, 1u, 0u,
            OS_OPT_TIME_HMSM_STRICT,
            &err);
    }
}

```

→LED task & Buzzer task

: MessageQueue의 Pend를 이용하여 adc_value(토양습도값)을 받아와 이 값이 200 이상인 경우(건조함) LED3(RED)과 부저를 동작시킨다. adc_value의 값이 그 이하일 경우(습함) LED2(BLUE)를 동작시킨다.

6. 역할분담

: 텀프로젝트 진행 시 마다 다같이 모여 Task 및 전체코드에 대해 토론하고 조사하여 다같이 코드를 구성하였다. 또한 보고서는 공유문서를 통해 같이 작성하였다.

7. 결론 및 기대효과

: 텀프로젝트를 통해 한 학기동안 배운 임베디드와 관련된 지식을 활용하여 실생활에 이용가능한 시스템을 설계/구현하였다. 이를 통해 임베디드 시스템의 필요성을 느꼈고, 내용에 대한 이해도 또한 더 높아졌다.

프로젝트를 진행하면서 토양 습도 센서의 값이 일정하지 못하고 중간중간 급격히 값이 튀는 경우가 있었다. 이러한 센서의 민감도가 개선되면 실생활에서 유용하게 사용될 수 있을 것이다. 그에 따라 식물을 잘 키우지 못해 포기했던 사람들도 쉽게 식물을 키울 수 있게 되는 효과를 기대할 수 있다.