

# REPORT



## 임베디드시스템 과제3

수강과목: 임베디드 시스템  
담당교수: 백윤주 교수님  
학 과: 전기컴퓨터공학부 정보컴퓨터공학전공  
이 름: 박윤형  
학 번: 201624481  
제출일자: 2021. 12. 10

## 1. bsp.c 파일 수정

```
static void BSP_LED_Init (void);  
static void USART_Config(void);  
static void BSP_PUSH_BUTTON_Init(void);
```

bsp.c의 BSP\_Init()에 USART와 USER button을 사용하기 위한 pin설정함수를 추가.

## 2. task 선언

```
static void AppTaskStart      (void      *p_arg);  
static void AppTaskCreate    (void);  
static void AppObjCreate     (void);  
  
static void AppTask_Button(void *p_arg);    //task determine if button is pressed  
static void AppTask_LED(void *p_arg);       //control LEDs  
static void AppTask_USART(void *p_arg);     //print string when button is pressed
```

버튼입력을 받는 task, LED를 조작하는 task, serial 통신을 하는 task를 각각 AppTask\_Button, AppTask\_LED, AppTask\_USART로 선언했다.

## 3. 전역변수 선언

```
112 unsigned short button_pressed = 0;    // 0: button isn't pressed, 1: button is pressed  
113  
114 OS_SEM MySem;                          // Semaphore  
115
```

Semaphore를 사용하기 위해 MySem이라는 semaphore 선언.

button\_pressed는 버튼을 눌렀는지 판단하기 위해 사용되는 변수다. AppTas\_Button에서 버튼이 눌렸음을 감지하면 button\_pressed가 1이 된다.

#### 4. Task 설명

##### -AppTask\_Button()

```
222 static void AppTask_Button(void *p_arg){
223     OS_ERR err;
224     CPU_TS ts;
225     BSP_LED_On(2);
226
227     int button = 0;
228
229     while (DEF_TRUE) {
230
231         while(!button){
232             button = GPIO_ReadInputDataBit(GPIOC, GPIO_Pin_13);
233             OSTimeDlyHMSM(0u, 0u, 0u, 50u, OS_OPT_TIME_HMSM_STRICT, &err);
234         }
235
236         OSSemPend(&MySem, 0, OS_OPT_PEND_BLOCKING, &ts, &err);
237
238         switch(err){
239             case OS_ERR_NONE:
240                 if(button){
241                     button_pressed = 1;
242                 }
243
244                 OSSemPost(&MySem, OS_OPT_POST_1, &err);
245
246             case OS_ERR_PEND_ABORT:
247                 break;
248             case OS_ERR_OBJ_DEL:
249                 break;
250         }
251
252         button = 0;
253
254         OSTimeDlyHMSM(0u, 0u, 0u, 50u, OS_OPT_TIME_HMSM_STRICT, &err);
255     }
256 }
```

231~234줄 : 버튼입력을 기다리다 입력이 들어오면 무한루프가 끝난다.

236~250줄 : 전역변수로 선언 되어있는 button\_pressed에 접근하기 전에 OSSemPend() 사용하고 정상적으로 semaphore에 에러가 없으면 button\_pressed를 1로 변경시키고 OSSemPost() 실행.

## -AppTask\_LED()

```
259 static void AppTask_LED(void *p_arg){
260
261     CPU_TS ts;
262     OS_ERR err;
263
264     int LED_mode = 0;        // 0: LED rolling, 1: show the result of button press
265     int LED_turn = 0;        //determine which LED to be turn on
266
267     while (DEF_TRUE) {
268         OSSemPend(&MySem, 0, OS_OPT_PEND_BLOCKING, &ts, &err);
269
270         switch(err){
271             case OS_ERR_NONE:
272                 if(button_pressed){
273                     LED_mode = 1;
274                 }
275
276                 OSSemPost(&MySem, OS_OPT_POST_1, &err);
277
278             case OS_ERR_PEND_ABORT:
279                 break;
280             case OS_ERR_OBJ_DEL:
281                 break;
282         }
283     }
```

268~282 줄 : 전역변수인 button\_pressed에 접근하기 전에 OSSemPend() 실행. semaphore에서 에러가 없으면 button\_pressed의 정보를 지역변수에 저장하고 OSSemPost() 실행.

그 외의 부분 : 버튼이 눌리지 않았을때는 LED1, LED2, LED3의 순서로 LED rolling을 하고, 버튼이 눌렸을 때 LED2가 켜져 있으면 모든 LED를 0.5초동안 켜고, LED1또는 LED3가 켜져있으면 모든 LED를 0.5초동안 끈다.

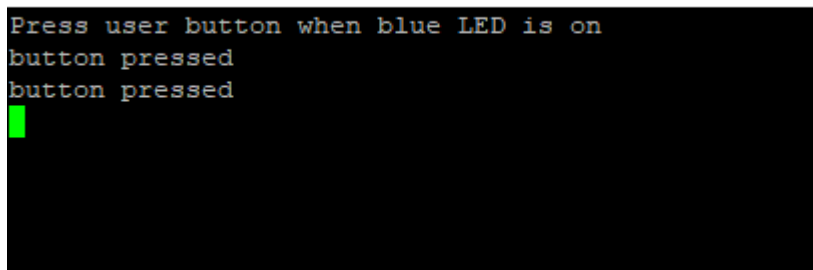
## -AppTask\_USART()

```
333 static void AppTask_USART(void *p_arg){
334
335     OS_ERR err;
336     CPU_TS ts;
337
338     send_string("Press user button when blue LED is on\n\r");
339
340     while (DEF_TRUE) {
341
342         OSSemPend(&MySem, 0, OS_OPT_PEND_BLOCKING, &ts, &err);
343
344
345         switch(err){
346             case OS_ERR_NONE:
347                 if(button_pressed){
348                     send_string("button pressed\n\r");
349                 }
350
351                 button_pressed = 0;
352                 OSSemPost(&MySem, OS_OPT_POST_1, &err);
353
354             case OS_ERR_PEND_ABORT:
355                 break;
356             case OS_ERR_OBJ_DEL:
357                 break;
358         }
359         OSTimeDlyHMSM(0u, 0u, 0u, 500u, OS_OPT_TIME_HMSM_STRICT, &err);
360     }
361 }
```

342줄에서 Semaphore의 신호를 기다리다 정상적으로 신호를 받으면 전역변수인 button\_pressed에 접근한다. 버튼이 눌렀다면 seial통신으로 버튼이 눌렸음을 출력한후 button\_pressed를 다시 0으로 초기화.

## 5. 결과

COM5 - PuTTY



```
Press user button when blue LED is on
button pressed
button pressed
█
```

1.보드 시작 시 putty로 파란색LED가 들어와 있을 때 버튼을 눌러야 함을 출력.

2. 버튼이 눌릴때마다 "button pressed"를 출력

3. LED1,2,3이 0.5초 간격으로 rolling blink

3. 파란색 LED가 켜있을 때 버튼을 누르면 모든 LED에 불이 들어오고, 다른 LED가 켜있을 때 버튼을 누르면 모든 LED가 꺼진다.