

main.cの追記部

```
23 // キューハンドル定義
24 QueueHandle_t LCDQueue;
25 QueueHandle_t MQTTQueue;
26 QueueHandle_t LOGQueue;
27 // セマフォハンドル定義
28 SemaphoreHandle_t ADCSemaphore;
29 SemaphoreHandle_t MQTTSemaphore;
30 // グローバル変数定義
31 char buff[128];
32
33 /** 関数プロトタイプ */
34 void SendStr(const char * str);
35 bool getResponse(char * word, uint8_t timeout);
36 void ftostring(int seisu, int shousu, float data, uint8_t *buffer);
37
38 /***** TC4割り込み処理関数 *****/
39 * TC4割り込み処理関数
40 * 0.5秒周期でセマフォをGive
41 *****/
42 void TC4_ISR(TC_TIMER_STATUS status, uintptr_t context){
43     static portBASE_TYPE HigherTask;
44
45     /** セマフォGive **/
46     HigherTask = pdFALSE;
47     xSemaphoreGiveFromISR(ADCSemaphore, &HigherTask);
48 }
49
```

ハンドル変数の定義

TC4の割り込み処理関数

task6のセマフォGive

```
56 int main ( void )
57 {
58     /* Initialize all modules */
59     SYS_Initialize ( NULL );
60
61     /** キュー定義 **/
62     LCDQueue = xQueueCreate(10, 20); // 20byte x 10
63     MQTTQueue = xQueueCreate(10, 20); // 20byte x 10
64     LOGQueue = xQueueCreate(10, 20); // 20byte x 10
65     /** タイマ4割り込み処理関数定義 **/
66     TC4_TimerCallbackRegister(TC4_ISR, (uintptr_t)NULL);
67     /** セマフォ生成 **/
68     ADCSemaphore = xSemaphoreCreateBinary();
69     MQTTSemaphore = xSemaphoreCreateBinary();
70     /** センサBME280初期化 **/
71     bme_init(); // センサ初期設定
72     bme_gettrim(); // センサ較正值読み出し
73
74     TC4_TimerStart(); // 電圧計測開始
75
76     /***** メインループ *****/
77     while ( true )
78     {
79         /* Maintain state machines of all polled MPLAB Harmony modules. */
80         SYS_Tasks ( );
81     }
82
83     /* Execution should not come here during normal operation */
84
85     return ( EXIT_FAILURE );
86 }
```

キューの生成

セマフォの生成