リスト5-3-6　task4.cの追記部

```
40  // セマフォのハンドル定義
41  SemaphoreHandle_t  MQTTSemaphore;
42  // キューハンドル定義
43  QueueHandle_t  MQTTQueue;
44  // タスク1のハンドル
45  TaskHandle_t t1Handle;
46
47  /** キュー送信用バッファ定義 ***/
48  union {
49      uint8_t Buf[20];
50      struct {
51          uint8_t kind;
52          float data1;
53          float data2;
54          float data3;
55      }mes;
56  }mqtt;
57  /** 計測データ出力用メッセージバッファ定義 31 **/
58  char bme[]={0x30,0x1D,0x00,0x04,'T','e','s','t','0','0','0','0','.','0','0',',','0','0',
59              '.','0','0',',','0','0','.','0','0',',','E','N','D'};
60  int i;
61  /*** 関数プロト ***/
62  void SendStr(const char * str);
63  bool getResponse(char * word, uint8_t timeout);
64  void ftostring(int seisu, int shousu, float data, char *buffer);
65
```

ハンドル定義

task1のハンドル定義

キュー受信バッファの定義

MQTTパブリッシュ用メッセージの定義

```
127     switch ( task4Data.state )
128     {
129         /* Application's initial state. */
130         case TASK4_STATE_INIT:
131         {
132             bool appInitialized = true;
133
134
135             if (appInitialized)
136             {
137
138                 task4Data.state = TASK4_STATE_WAIT;
139             }
140             break;
141         }
142         case TASK4_STATE_WAIT:                                  // タスク1完了待ち
143             /*** セマフォTake ***/
144             xSemaphoreTake(MQTTSemaphore, portMAX_DELAY);       //永久待ち
145             task4Data.state = TASK4_STATE_SERVICE_TASKS;        // 次へ
146             break;
147         case TASK4_STATE_SERVICE_TASKS:
148         {
149             /** キューから取り出し **/
150             xQueueReceive(MQTTQueue, mqtt.Buf, portMAX_DELAY); // 永久待ち
151             /*** 文字列に変換 ***/
152             ftostring(4, 2, mqtt.mes.data1, &bme[8]);
153             ftostring(2, 2, mqtt.mes.data2, &bme[16]);
154             ftostring(2, 2, mqtt.mes.data3, &bme[22]);
155             // 計測データMQTTパブリッシュ送信実行
156             SendStr("AT+CIPSEND=31¥r¥n");                       // 送信
157             if(getResponse("OK", 1)==false)                     // エラーのとき
158             {
159                 SERCOM3_USART_Write("NG¥r¥n", 4);
160                 vTaskResume(t1Handle);                          // タスク1再実行
161                 task4Data.state = TASK4_STATE_WAIT;
162             }
163             for (i=0;i<31;i++)
164                 SERCOM5_USART_WriteByte(bme[i]);                // MQTTへパブリッシュ
165             if(getResponse("OK", 1)==false)                     // エラーのとき
166             {
167                 SERCOM3_USART_Write("NG¥r¥n", 4);
168                 vTaskResume(t1Handle);                          // タスク1再実行
169                 task4Data.state = TASK4_STATE_WAIT;
170             }
171             break;
172         }
```

WAITステートに変更

WAITステートを追加

キューからデータ取り出し

MQTTパブリッシュ実行

送信エラーの時はtask1をレジューム