

リスト5-3-5 task3.cの追記部

```

40 // キューハンドル定義
41 QueueHandle_t LCDQueue;
42 /** キュー受信用バッファ定義 **/
43 union {
44     uint8_t Buf[20];
45     struct {
46         uint8_t kind;
47         float data1;
48         float data2;
49         float data3;
50     } mes;
51 }Disp;
52 /** 計測データ表示用メッセージバッファ定義 **/
53 const uint8_t Room1[] = {0x81,0x9A,0x81,0x40,0x8E,0xBA,0x93,0xE0,0x8A,0xC2,
54     0x8B,0xAB,0x82,0xCC,0x8F,0xEE,0x95,0xF1,0x95,0x5C,0x8E,0xA6,0x81,0x40,
55     0x81,0x9A,0x00,0x00}; // **室内環境の情報表示**
56 const uint8_t Room2[] = {0x8B,0x43,0x88,0xB3,0x81,0x46,0x00,0x00}; // 気圧:
57 const uint8_t Room3[] = {0x89,0xB7,0x93,0x78,0x81,0x46,0x00,0x00}; // 温度:
58 const uint8_t Room4[] = {0x8E,0xBC,0x93,0x78,0x81,0x46,0x00,0x00}; // 湿度:
59 const uint8_t Room5[] = {0x81,0x9F,0x81,0x40,0x89,0xC2,0x95,0xCF,0x92,0xEF,
60     0x8D,0x52,0x82,0xCC,0x93,0x64,0x88,0xB3,0x8C,0x76,0x91,0xAA,0x81,0x40,
61     0x81,0x9F,0x00,0x00}; // **可変抵抗の電圧計測**
62 uint8_t Msg1[] = "P=1013 hPa ";
63 uint8_t Msg2[] = "T=23.1 DegC";
64 uint8_t Msg3[] = "H=55.6 %RH ";
65 uint8_t Msg4[] = "V1=x.xx volt";
66 uint8_t Msg5[] = "V2=x.xx volt";
67 /** 関数プロト **/
68 void ftostring(int seisu, int shousu, float data, uint8_t *buffer);

```

ハンドル定義

キュー受信バッファの定義

LCD表示メッセージの定義

```

104 void TASK3_Initialize ( void )
105 {
106     /* Place the App state machine in its initial state. */
107     task3Data.state = TASK3_STATE_INIT;
108
109
110     /* TODO: Initialize your application's state machine and other
111     * parameters.
112     */
113     /** LCD初期化 **/
114     lcd_Init();
115     lcd_Clear(BLACK);
116 }

```

LCD初期化

```

147
148
149
150 case TASK3_STATE_SERVICE_TASKS:
151 {
152     /** キューから取り出し **/
153     xQueueReceive(LCDQueue, Disp.Buf, portMAX_DELAY);
154     /** 環境データの場合 **/
155     if(Disp.mes.kind == 'M'){
156         /** 文字列に変換 **/
157         ftostring(4, 0, Disp.mes.data1, Msg1+2);
158         ftostring(2, 1, Disp.mes.data2, Msg2+2);
159         ftostring(2, 1, Disp.mes.data3, Msg3+2);
160         /** 表示実行 **/
161         Kanji_Str(0, 0, Room1, WHITE, BLUE);
162         Kanji_Str(1, 1, Room2, WHITE, BLACK);
163         lcd_Str(4, 1, Msg1+2, WHITE, BLACK);
164         Kanji_Str(1, 2, Room3, GREEN, BLACK);
165         lcd_Str(4, 2, Msg2+2, GREEN, BLACK);
166         Kanji_Str(1, 3, Room3, CYAN, BLACK);
167         lcd_Str(4, 3, Msg3+2, CYAN, BLACK);
168     }
169     /** 電圧計測の場合 **/
170     else if(Disp.mes.kind == 'V'){
171         /** 文字列に変換 **/
172         ftostring(1, 2, Disp.mes.data1, Msg4+3);
173         ftostring(1, 2, Disp.mes.data2, Msg5+3);
174         /** LCD表示実行 **/
175         Kanji_Str(0, 4, Room5, WHITE, RED);
176         lcd_Str(1, 5, Msg4, YELLOW, BLACK);
177         lcd_Str(1, 6, Msg5, YELLOW, BLACK);
178     }
179     /** SDカードログメッセージの場合 **/
180     else if(Disp.mes.kind == 'S'){
181         Kanji_Str(2, 8, &Disp.Buf[1], BLACK, YELLOW);
182     }
183     else if(Disp.mes.kind == 'E'){
184         Kanji_Str(2, 8, &Disp.Buf[1], BLACK, RED);
185     }
186     break;
187 }

```

キューからデータ取り出し

データ種別ごとに編集して表示