

リスト5-3-9 task6.cの追記部

```

38 // キューハンドル定義
39 QueueHandle_t LCDQueue;
40 QueueHandle_t LOGQueue;
41 // セマフォ定義
42 SemaphoreHandle_t ADCSemaphore;
43
44 /** キュー送信用バッファ定義 **/
45 union {
46     uint8_t Buf[20];
47     struct {
48         uint8_t kind;
49         float pot1;
50         float pot2;
51     } mes;
52 }Potent;
53 /** グローバル変数定義 **/
54 uint16_t result;
55 extern uint8_t LogFlag; // ログ中フラグ

```

ハンドル定義

セマフォ定義

キュー送信バッファの定義

```

90 void TASK6_Initialize ( void )
91 {
92     /* Place the App state machine in its initial state. */
93     task6Data.state = TASK6_STATE_INIT;
94
95
96
97     /* TODO: Initialize your application's state machine and
98      * parameters.
99      */
100     ADCO_Enable();
101     Potent.mes.kind = 'V'; // データ区別セット
102 }
103

```

ADC有効化

```

133 case TASK6_STATE_SERVICE_TASKS:
134 {
135     /** TC4セマフォTake **/
136     xSemaphoreTake(ADCSemaphore, portMAX_DELAY); //永久待ち
137     /** POT1のAD変換 **/
138     ADCO_ChannelSelect(ADC_POSINPUT_AIN14, ADC_NEGINPUT_GND);
139     ADCO_ConversionStart();
140     while(!ADCO_ConversionStatusGet());
141     result = ADCO_ConversionResultGet();
142     Potent.mes.pot1 = (3.3 * result) / 4095;
143     /** POT2のAD変換 **/
144     ADCO_ChannelSelect(ADC_POSINPUT_AIN15, ADC_NEGINPUT_GND);
145     ADCO_ConversionStart();
146     while(!ADCO_ConversionStatusGet());
147     result = ADCO_ConversionResultGet();
148     Potent.mes.pot2 = (3.3 * result) / 4095;
149     /** キューへ送信 **/
150     xQueueSend(LCDQueue, Potent.Buf, 0);
151     if(LogFlag == 1)
152         xQueueSend(LOGQueue, Potent.Buf, 0);
153     break;
154 }

```

電圧を計測

表示とログ
キューに送信

// LCDキューに送信
// ログ中の場合
// ログキューに送信