

リスト5-3-2 main.cに追加したサブ関数

```

87  □ /*****
88  * WiFi文字列送信関数
89  *****/
90  □ void SendStr(const char * str){
91      while(*str != 0){ // 文字列最後まで繰り返し
92          SERCOM5_USART_Write((char *)str, 1); // 送信実行
93          str++;
94      }
95  }
96  □ /*****
97  * ESP応答待ち
98  *****/
99  □ bool getResponse(char * word, uint8_t timeout){
100     char flag, a=0;
101     int j;
102
103     for(j=0;j<128;j++) // 受信バッファリセット
104         buff[j] = 0x20;
105     j = 0;
106     flag = 0; // 文字列発見フラグリセット
107     TC5_Timer16bitPeriodSet(3906*timeout); // タイマ3秒数セット
108     TC5_Timer16bitCounterSet(0);
109     TC5_TimerStart(); // タイマ5スタート
110     /*****/
111     while(TC5_TimerPeriodHasExpired() == false){ // タイムアップまで繰り返し
112         while(SERCOM5_USART_ReceiverIsReady() == true){ // 受信データありの場合
113             a = SERCOM5_USART_ReadByte(); // 受信データ取得
114             SERCOM3_USART_WriteByte(a);
115             if(a == '\0') continue; // 0x00は省く
116             buff[j] = a; // 受信バッファに追加
117             if(j<126) // 126文字以上は無視
118                 j++; // 次のバッファへ
119             buff[j] = 0; // 文字列終わりのフラグ
120         }
121         if(strstr(buff, word) != 0){ // 文字列検索
122             flag = 1; // 文字列発見フラグオン
123             TC5_TimerStop(); // タイマ1停止
124             break; // 強制終了
125         }
126     }
127     if(flag == 1)
128         return true; // 文字列が見つかった場合
129     else
130         return false; // タイムアップの場合
131 }

```

デバッグ支援
用UART出力

Wi-Fiの応答確認関数

タイムアウト検出用
にTC5を使う

指定文字列検出
でTrue、タイム
アウトでFalse

```

133 □ /*****
134 * Floatから文字列へ変換
135 * 合計有効桁は8桁以下とすること
136 *****/
137 void ftostring(int seisu, int shousu, float data, uint8_t *buffer)
138 {
139     int i;
140     long dummy;
141
142     if(shousu != 0) // 小数部桁ありか
143         buffer += seisu+shousu+1; // 全体桁数+小数点
144     else // 小数部桁なしのとき
145         buffer += seisu + shousu; // 全体桁数のみ
146     buffer--; // 配列ポインタ-1
147     for(i=0; i<shousu; i++) // 小数部を整数に変換
148         data = data * 10; // 10倍
149     /// dummyがオーバーフローすると変換不可(8桁が限界)
150     dummy = (long) (data + 0.5); // 四捨五入して整数に変換
151     for(i=shousu; i>0; i--) { // 小数桁数分繰り返し
152         *buffer = (unsigned char)(dummy % 10)+'0'; // 数値を文字に変換格納
153         buffer--; // 格納場所下位から上位へ
154         dummy /= 10; // 次の桁へ
155     }
156     if(shousu != 0) { // 小数桁0なら小数点なし
157         *buffer = '.'; // 小数点を格納
158         buffer--; // ポインタ-1
159     }
160     for(i=seisu; i>0; i--) { // 整数桁分繰り返し
161         *buffer = (unsigned char)(dummy % 10)+'0'; // 数値を文字に変換格納
162         buffer--; // ポインタ-1
163         dummy /= 10; // 次の桁へ
164     }
165 }

```