

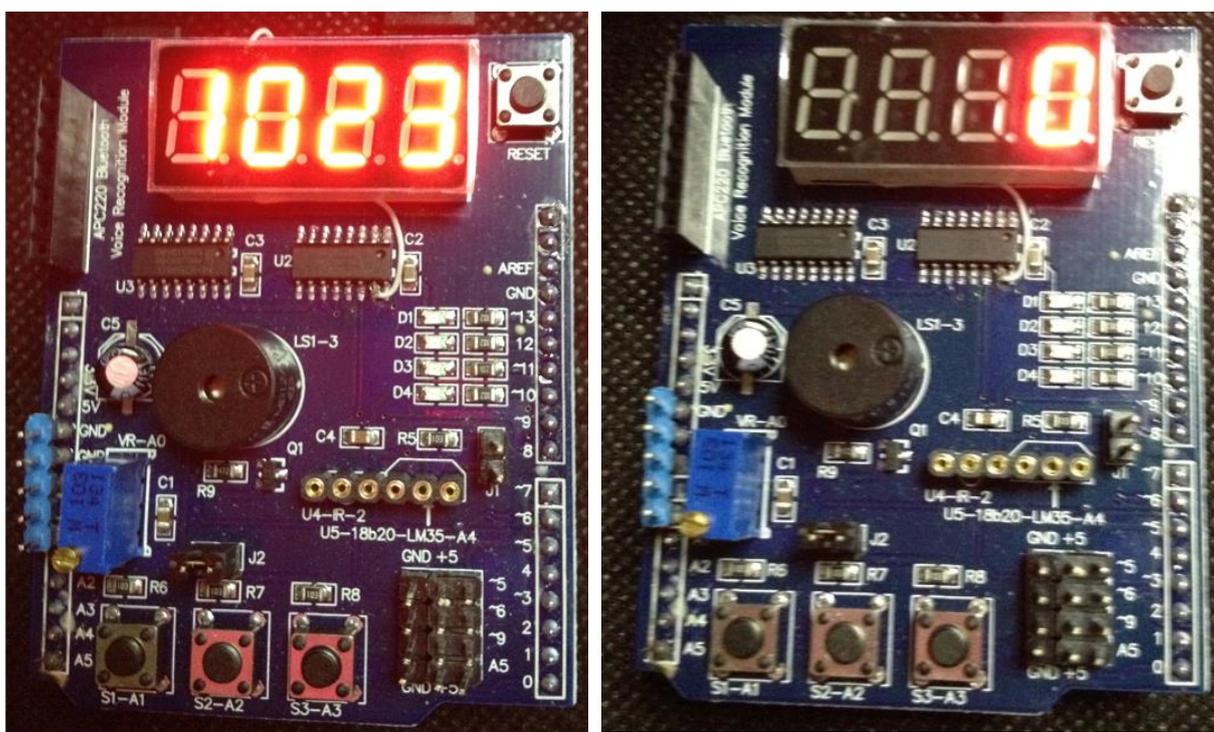
Multi function シールドの半固定抵抗器動作確認

Multi function シールドで確認していなかった半固定抵抗器ですが、やはり気になったので、動作確認するスケッチを作成してみました。

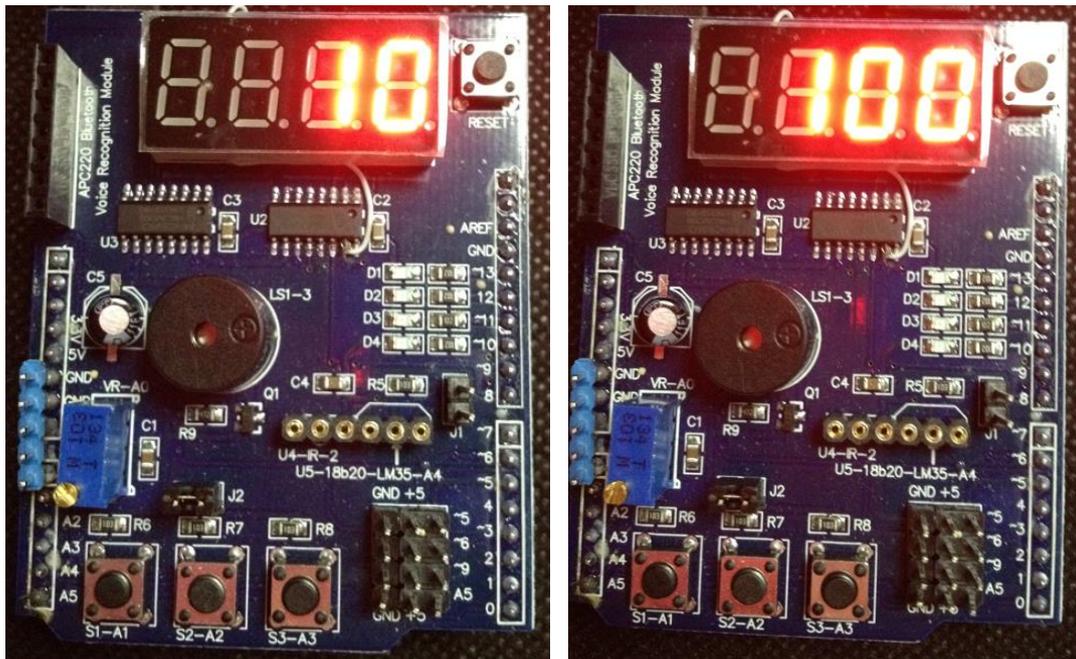
具体的には、半固定抵抗器からの電圧を読み出して、その結果を7セグ LED 部に表示することにします。これまでのスケッチでは、7セグ LED 部に時間を表示していましたが、そこを変更します。そのままでは、面白くないので、上位の桁の“0”は表示しない(0 サプレスする)ようにする機能も追加しました。

デフォルトでは 0 サプレス状態で、MFS の左側のスイッチを押すたびに 0 表示(0 パディング)と 0 サプレスが切り替わります。

下の左側の写真は最大値の写真です。この場合は、0 サプレスでも 0 パディングでも表示は変わりません。右側は最後の 0 以外は表示されていないので、0 サプレスが有効になっているのが、分かります。



表示が 2 桁や 3 桁の場合の例も示します。



なお、左側のブルーの5ピンヘッドが見えるのは、電圧をテストで確認するために立てたものです。

測定は100ミリ秒ごとに測定し、4回分の移動平均をとる筈にしています。このために、以下の変数が準備されています。(最後の2行分は表示の0サプレスのための変数です。)

```

43 int analog_data; // 半固定抵抗器からの電圧の読み出し値
44 int analog_work; // 作業用
45 int ad_cnt = 0; // 読み出し値の格納ポインタ
46 int analog_buff[4]; // 読み出し値の格納バッファ
47 int analog_sum; // 4回分の合計値
48 byte blank_flag = 1; // 0サプレッション・フラグ
49 byte blank_mode = 1; // 0サプレス モード

```

下に示すのが、初期設定部分です。最初に、analogRead関数で半固定抵抗器の値を読み出して、その値で4回分のバッファを埋めています。

```

66 // アナログ入力の初期設定
67
68 analogReference(DEFAULT);
69 analog_data = analogRead(VR_IN); // アナログ値を読みを出す
70 analog_buff[0] = analog_data; // バッファの値を設定
71 analog_buff[1] = analog_data;
72 analog_buff[2] = analog_data;
73 analog_buff[3] = analog_data;
74 analog_sum = ( analog_data << 2 ); // 積算値の初期化

```

下が、1000 の桁を表示する部分です。Blank flag が 1 でデータが 0 なら、“0”の表示の代わりに 10(空白を表示するデータ)を設定しています。

```
104         case 0x00:                                // 1000の桁
105             dt_wk = ( analog_data / 1000 ); // 1000の桁を求める
106             if ( ( 1 == blank_flag ) && (0x00 == dt_wk ) )
107             {
108                 dt_wk = 10;                        // 0表示を消す
109             }
110             else
111             {
112                 blank_flag = 0;                    // 0サプレスを解除
113             }
114             com_sel = LED_DIGIT[0];                // 桁選択信号をセット
115             break;
```

下は、1 の桁を表示する部分です。1 の桁はデータによらず表示します。また、147 行目では 0 サプレス・フラグを元に戻しています。これで、0 サプレスを制御しています。

```
145         default:                                // 1の桁
146             dt_wk = ( analog_data % 10 ); // 1の桁を求める
147             blank_flag = blank_mode;      // 0サプレスを設定
148             com_sel = LED_DIGIT[3];      // 桁選択信号をセット
```

下は、100 ミリ秒ごとに半固定抵抗器からのアナログ電圧を新しい値に更新する処理です。一番古い値を新しいデータに置き換えています。また、合計値から一番古い値を引いて、新しい値を加算して新しい合計値を求め、最新の合計値を右に 2 ビットシフトすることで平均値を求めています。

```
171 /*-----*/
172 *   最新の値を取り込む
173 -----*/
174     analog_data = analogRead(VR_IN); // アナログ値を読み出す
175     analog_sum -= analog_buff[ad_cnt]; // もっとも古いデータを削除
176     analog_sum += analog_data;      // 最新のデータを加算
177     analog_buff[ad_cnt] = analog_data; // 最新のデータを格納
178
179     analog_data = ( analog_sum >> 2); // 最新の平均値を設定
180     ++ad_cnt;                          // ポインタ更新
181     ad_cnt &= 0x03;                    // 折り返し
182
```

最後は、20 ミリ秒ごとに、スイッチの状態をチェックして、押されたら、0 サプレスのモードを切り替えています。

```

185 /*-----
186     switch data check timing
187 -----*/
188
189     if ( ++prescale4sw >= 5 )      // check SW check timing or not
190     {                               // 20milli seconds passing
191         prescale4sw = 0x00;        // clear timing counter
192         sw_data <<= 1;             // shift olddata left
193         sw_data += ( digitalRead(ex_swPin)&digitalRead(swPin) ); // read sw data
194         sw_work = (sw_data & 0x06); // extract check timing data
195         if ( 0x04 == sw_work )
196         {
197             blank_mode ^= 0x01;    // change blank mode flag
198         }
199     }

```

半固定抵抗器の動作確認でしたが、表示関係の部分が多かったかもしれません。

これで、Multi function シールドのボード上の機能は完了です。

以上