```
 1 /*---------------------------------------------------------------------------*/
 2 /*  ものつくりコンテスト山形県大会08 電子回路組立部門                       */
 3 /*                                                                           */
 4 /* 全工長協会HP掲示  制御プログラム課題(例)                                  */
 5 /*                                                                           */
 6 /* 課題6   物体の位置に相当するＬＥＤと７セグメントＬＥＤの両方を同時に点灯 */
 7 /*         させる。                                                          */
 8 /*                                                                           */
 9 /*   センサーの前に物体を置き、その位置のＬＥＤの点灯位置を７セグメントＬＥＤ */
10 /*   で数値を同時点灯させる。次に物体を移動させた時にＬＥＤの点灯位置、７セグ */
11 /*   メントＬＥＤの数値も変化させる。範囲、測定位置と点灯ＬＥＤは課題２と同じ */
12 /*   とする。７セグメントＬＥＤの数値パターンは課題３と同じとする。          */
13 /*                                                                           */
14 /*   初期状態                                                                */
15 /*     (1) センサは１ｍ以上前に何もおいていない。                            */
16 /*     (2) トグルスイッチはＬの位置にある。                                  */
17 /*     (3) タクトスイッチは押されていない。                                  */
18 /*     (4) 光センサは光を受光している。                                      */
19 /*     (5) タクトスイッチ動作による状態変化は、特に指定がない限り押した瞬間 */
20 /*         に行うものとする。                                                */
21 /*                                                                           */
22 /* ※今後確認                                                                */
23 /*   距離のデータはあらかじめ測定したものを使用した。                        */
24 /*   同時表示なので、LED,7segLED表示処理は割込処理とした。                   */
25 /*                                                                           */
26 /*                                山形電波工業高等学校 情報技術科 & 電子工学科 */
27 /*                                                Ver. 1.0 平成20年5月 6日(火) */
28 /*                                                Ver. 2.0 平成20年5月10日(土) */
29 /*                                                                           */
30 /*---------------------------------------------------------------------------*/
31 //#include "comment08.h"
32
33 /*---------------------------------*/
34 /* 組み込みファイル定義            */
35 /*---------------------------------*/
36   #include <3048.h>
37   #include "macro08.h"
38
39
40 /*-------------------------------------------------*/
41 /* グローバル変数宣言(どの関数からも参照、変更できる) */
42 /*-------------------------------------------------*/
43   unsigned char  psd_data;            /* 割り込みで表示処理する */
44
45   unsigned int  itu4_count = 0;       /* 10msで+1のカウンタ(iTU0割り込みでカウント) */
46
47   unsigned char  ledort7seg_flag = 0; /* 課題6 LED or 7segLEDどちらに表示するか */
48
49
50 /*-------------------------------------------------*/
51 /* このプログラム中で使用する関数の記述            */
52 /*-------------------------------------------------*/
53 //#include "dempalib08.h"
54
55
56 /*---------------------------------------------------------------------------*/
57 /*  モジュール名 Init_Port                                                   */
58 /*  処理概要     H8 ポート初期化                                             */
59 /*  引数         なし                                                        */
60 /*  戻り値       なし                                                        */
61 /*---------------------------------------------------------------------------*/
62 void Init_Port( void )
63 {
64       /* I/Oポート設定  0:入力  1:出力 */
65          P6.DDR = 0x00;             /* on bord ディップスイッチ       */
66                                     /* 0x00 = 0000 0000  全bit入力設定 */
67
68          P4.DDR = 0xff;             /* 事前配布基板(1) LED & 7segLED          */
69                                     /* 0xff = 1111 1111  全bit出力設定        */
70          P4.DR.BYTE = 0x00;
71
72          PA.DDR = 0xff;             /*                                        */
73                                     /* 0x00 = 0000 0000  全bit入力設定        */
74          PA.DR.BYTE = 0x00;
75
76          PB.DDR = 0xff;             /*                                        */
77                                     /* 0xff = 1111 1111  全bit出力設定        */
78          PB.DR.BYTE = 0x00;
79
80          P5.DDR = 0xff;             /* 事前配布基板(1) LED or 7segLED指示端子 */
81                                     /* 0xff = 1111 1111  全bit出力設定        */
82          P5.DR.BYTE = 0x00;
83
84                                     /* 当日作成基板はP7ポート(P7は入力専用なので設定なし) */
85 }
86
```

```
 87
 88 /*------------------------------------------------------------------------------*/
 89 /*  モジュール名 Init_H8                                                        */
 90 /*  処理概要     H8 タイマー初期化                                              */
 91 /*  引数         なし                                                           */
 92 /*  戻り値       なし                                                           */
 93 /*------------------------------------------------------------------------------*/
 94 /* 参考 H8/3048f(旧cpu) 14.7854MHz      */
 95 /*       16ms   29490                   */
 96 /*       10ms   18432                   */
 97 /*        8ms   14745                   */
 98 /*        5ms    9216                   */
 99 /*        1ms    1843                   */
100 /*--------------------------------------*/
101 void Init_H8( void )
102 {
103     EI; /* 割り込み許可 */
104
105       /* タイマ設定      */
106          ITU.TMDR.BIT.PWM1 = 0;    /* ITU1:通常動作        */
107          ITU.TMDR.BIT.PWM2 = 0;    /* ITU2:通常動作        */
108          ITU.TMDR.BIT.PWM3 = 1;    /* ITU3:PWMモード       */
109          ITU.TMDR.BIT.PWM4 = 0;    /* ITU4:通常動作        */
110
111
112       /* iTu1 タイマ設定 */
113          ITU1.TCR.BIT.CCLR = 1;    /* カウンタクリア要因                          */
114          ITU1.TCR.BIT.CKEG = 0;    /* クロックエッジ                              */
115          ITU1.TCR.BIT.TPSC = 3;    /* タイマプリスケーラ 1.8432MHz                */
116          ITU1.GRA = 18432;         /* 割り込みインターバル 1.8432MHz/18432=100Hz */
117          ITU1.TIER.BIT.IMIEA = 1;  /* IMFAフラグによる割り込み許可                */
118
119       /* iTu2 タイマ設定 */
120          ITU2.TCR.BIT.CCLR = 1;    /* カウンタクリア要因                          */
121          ITU2.TCR.BIT.CKEG = 0;    /* クロックエッジ                              */
122          ITU2.TCR.BIT.TPSC = 3;    /* タイマプリスケーラ 1.8432MHz                */
123          ITU2.GRA = 18432;         /* 割り込みインターバル 1.8432MHz/18432=100Hz */
124          ITU2.TIER.BIT.IMIEA = 1;  /* IMFAフラグによる割り込み許可                */
125
126       /* iTu3 モータタイマ設定 */
127          ITU3.TCR.BIT.CCLR = 1;    /* カウンタクリア要因                          */
128          ITU3.TCR.BIT.CKEG = 0;    /* クロックエッジ                              */
129          ITU3.TCR.BIT.TPSC = 3;    /* タイマプリスケーラ 1.8432MHz                */
130          ITU3.GRA = PWM_CYCLE;     /* モータPWM周期 1.8432MHz/1843=1kHz           */
131          ITU3.GRB = 0;             /* モータPWM DUTY 0%                           */
132
133       /* タイマ設定 */
134          ITU4.TCR.BIT.CCLR = 1;    /* カウンタクリア要因                          */
135          ITU4.TCR.BIT.CKEG = 0;    /* クロックエッジ                              */
136          ITU4.TCR.BIT.TPSC = 3;    /* タイマプリスケーラ 1.8432MHz                */
137          ITU4.GRA = 18432;         /* 割り込みインターバル 1.8432MHz/18432=100Hz */
138          ITU4.TIER.BIT.IMIEA = 1;  /* IMFAフラグによる割り込み許可                */
139
140
141       /* タイマスタート */
142          ITU.TSTR.BIT.STR0 = 0;    /* まだスタートしない */
143          ITU.TSTR.BIT.STR1 = 0;    /* まだスタートしない */
144          ITU.TSTR.BIT.STR2 = 0;    /* まだスタートしない */
145          ITU.TSTR.BIT.STR3 = 0;    /* まだスタートしない */
146          ITU.TSTR.BIT.STR4 = 0;    /* まだスタートしない */
147 }
148
149
150
151 /*------------------------------------------------------------------------------*/
152 /*  モジュール名  start_itu0                                                    */
153 /*  処理概要      タイマーitu4のスタート                                        */
154 /*  引数          なし                                                          */
155 /*  戻り値        なし                                                          */
156 /*------------------------------------------------------------------------------*/
157 void start_warikomi( void )
158 {
159         ITU.TSTR.BIT.STR4 = 1; /* itu4スタート */
160 }
161
162
163 /*------------------------------------------------------------------------------*/
164 /*  モジュール名 課題6用 int_imia4                                              */
165 /*  処理概要                                                                    */
166 /*  引数         なし                                                           */
167 /*  戻り値       なし                                                           */
168 /*------------------------------------------------------------------------------*/
169 void int_imia4( void )
170 {
171
172     unsigned char  w_data;        /* コンパイラ不良? worning対策 */
```

```
173
174           itu4_count++;
175           if( itu4_count == 1 ) {   /* 10ms毎切り替える */
176
177                if( ledort7seg_flag == LED ) {
178
179                       /*--------------------------*/
180                       /* 7セグメントLED出力       */
181                       /*--------------------------*/
182                         SEG_SELECT = T7SEG;
183                         ledort7seg_flag = T7SEG;
184                                   if( psd_data >= 117 ) {
185                                             KIBAN081 = tendo[ 1 ];
186
187                                   } else if( psd_data >= 80 ) {
188                                             KIBAN081 = tendo[ 2 ];
189
190                                   } else if( psd_data >= 54 ) {
191                                             KIBAN081 = tendo[ 3 ];
192
193                                   } else if( psd_data >= 41 ) {
194                                             KIBAN081 = tendo[ 4 ];
195
196                                   } else if( psd_data >= 30 ) {
197                                             KIBAN081 = tendo[ 5 ];
198
199                                   } else if( psd_data >= 26 ) {
200                                             KIBAN081 = tendo[ 6 ];
201
202                                   } else if( psd_data >= 23 ) {
203                                             KIBAN081 = tendo[ 7 ];
204
205                                   } else if( psd_data >= 20 ) {
206                                             KIBAN081 = tendo[ 8 ];
207
208                                   } else {
209                                             KIBAN081 = ~0x00;
210
211                                   }
212
213                } else {
214                       /*--------------------------*/
215                       /* LED出力                  */
216                       /*--------------------------*/
217                         SEG_SELECT = LED;
218                         ledort7seg_flag = LED;
219
220
221                          if( psd_data >= 117 ) {
222                                             KIBAN081 = ~0x01;
223
224                          } else if( psd_data >= 80 ) {
225                                             KIBAN081 = ~0x02;
226
227                          } else if( psd_data >= 54 ) {
228                                             KIBAN081 = ~0x04;
229
230                          } else if( psd_data >= 41 ) {
231                                             KIBAN081 = ~0x08;
232
233                          } else if( psd_data >= 30 ) {
234                                             KIBAN081 = ~0x10;
235
236                          } else if( psd_data >= 26 ) {
237                                             KIBAN081 = ~0x20;
238
239                          } else if( psd_data >= 23 ) {
240                                             KIBAN081 = ~0x40;
241
242                          } else if( psd_data >= 20 ) {
243                                            /*KIBAN081 = ~0x80;    warning  */
244                                             w_data = 0x80;
245                                             KIBAN081 = ~w_data;
246
247                          } else {
248                                             KIBAN081 = ~0x00;
249                          }
250
251
252
253                }
254
255             itu4_count = 0;
256
257           }
258
```

```
259         /*------------------------*/
260         /* reset interrupt request */
261         /*------------------------*/
262            ITU4.TSR.BIT.IMFA = 0;
263 }
264
265
266
267 /*--------------------------------------------------------------------------*/
268 /*  モジュール名 wait                                                       */
269 /*  処理概要     ソフトウェアタイマー                                       */
270 /*  引数         タイマー値    1:  10[ms]                                   */
271 /*                            10: 100[ms] = 0.1[s]                        */
272 /*                            50: 500[ms] = 0.5[s]                        */
273 /*                           100:1000[ms] = 1.0[s]                        */
274 /*  戻り値       なし                                                       */
275 /*--------------------------------------------------------------------------*/
276 void wait( int iTimer )
277 {
278      int i;
279           while( iTimer ) {
280                for( i = 0; i < 5000; i++ );    /* H8/3048f */
281           /*for( i = 0; i < 8333; i++ );*/  /* H8/3048f-one */
282                iTimer--;
283           }
284 }
285
286
287 /*--------------------------------------------------------------------------*/
288 /*  モジュール名  seg_cls                                                   */
289 /*  処理概要      7セグメントLED消去                                        */
290 /*  引数          なし                                                      */
291 /*  戻り値        なし                                                      */
292 /*--------------------------------------------------------------------------*/
293 void seg_cls( void )
294 {
295           KIBAN081= 0xff;   /* KIBAN081 = ~0x00; */
296
297 }
298
299
300 /*--------------------------------------------------------------------------*/
301 /*  モジュール名  openning                                                  */
302 /*  処理概要      オープニング(山形電波工高固有)                            */
303 /*  引数          なし                                                      */
304 /*  戻り値        なし                                                      */
305 /*--------------------------------------------------------------------------*/
306 void openning08( void )
307 {
308      signed char i;                    /* -128 ～ +127 */
309
310           SEG_SELECT = T7SEG;          /* 表示は7セグメントLED */
311           for( i = 3; i >= 0; i-- ) {
312                KIBAN081= tendo[ i ];   /* 表示 */
313                   wait( 50 );
314                seg_cls();              /* 消去 */
315                   wait( 50 );
316           }
317 }
318
319
320 /*--------------------------------------------------------------------------*/
321 /*  モジュール名 Init_ad                                                    */
322 /*  処理概要     ＡＤ初期化                                                 */
323 /*  引数         なし                                                       */
324 /*  戻り値       なし                                                       */
325 /*--------------------------------------------------------------------------*/
326 void Init_ad( void )
327 {
328        /* ADコントロールレジスタ設定 */
329           AD.CSR.BYTE = 0x00;     /* ch0(AN0 (P70) ) 単一モード */
330         /*AD.CSR.BYTE = 0x01;*/   /* ch1(AN1 (P71) ) 単一モード */
331         /*AD.CSR.BYTE = 0x02;*/   /* ch2(AN2 (P72) ) 単一モード */
332         /*AD.CSR.BYTE = 0x03;*/   /* ch3(AN3 (P73) ) 単一モード */
333         /*AD.CSR.BYTE = 0x04;*/   /* ch4(AN4 (P74) ) 単一モード */
334         /*AD.CSR.BYTE = 0x05;*/   /* ch5(AN5 (P75) ) 単一モード */
335         /*AD.CSR.BYTE = 0x06;*/   /* ch6(AN6 (P76) ) 単一モード */
336         /*AD.CSR.BYTE = 0x07;*/   /* ch7(AN7 (P77) ) 単一モード */
337
338 }
339
340
341 /*--------------------------------------------------------------------------*/
342 /* モジュール名 start_ad                                                    */
343 /* 処理概要     ＡＤ変換スタート                                            */
344 /* 引数         なし                                                        */
```

```
345 /* 戻り値       なし                                                          */
346 /*--------------------------------------------------------------------------*/
347 void start_ad( void )
348 {
349           AD.CSR.BIT.ADST = 1; /* AD変換開始 */
350
351 }
352
353
354 /*--------------------------------------------------------------------------*/
355 /* モジュール名 end_ad                                                      */
356 /* 処理概要     ＡＤ変換終了                                                */
357 /* 引数         なし                                                        */
358 /* 戻り値       なし                                                        */
359 /*--------------------------------------------------------------------------*/
360 void end_ad( void )
361 {
362           while( AD.CSR.BIT.ADF != 1 ){}  /* AD変換終了チェック */
363
364 }
365
366
367 /*--------------------------------------------------------------------------*/
368 /* モジュール名 get_ad8                                                     */
369 /* 処理概要     データ読み込み                                              */
370 /*              上位10bitにデータが格納されるので、下位8bitにシフトし(右に8bit) */
371 /*              変換データを確定する。（下位2bitゴミとしてカット）          */
372 /* 引数         なし                                                        */
373 /* 戻り値       8bit data                                                   */
374 /*              1111 1111 11-- ---- : 変換後データ                          */
375 /*              ---- ---- 1111 1111 : 8bitシフト後データ                    */
376 /*                                                                          */
377 /*              ---- ---- 0000 0000 : 0x00      0                           */
378 /*                        :                     :                           */
379 /*              ---- ---- 1111 1111 : 0xff     255  0～255 の 256通りのデータ  */
380 /*--------------------------------------------------------------------------*/
381 unsigned char get_ad8( void )
382 {
383     unsigned char indata;        /* unsigned intで 0 ～ 255 まで扱える */
384
385           indata = AD.DRA >>8;    /* 上位10bitを右に8bitシフトする。下位2bit無視 */
386                                   /* 1111 1111 1100 0000           */
387                                   /* ---- ---- 1111 1111           */
388
389         /*indata = AD.DRB >>8;*/  /* 上位10bitを右に8bitシフトする。*/
390         /*indata = AD.DRC >>8;*/  /* 上位10bitを右に8bitシフトする。*/
391         /*indata = AD.DRD >>8;*/  /* 上位10bitを右に8bitシフトする。*/
392         //indata = indata & 0xff;
393
394           return indata;
395 }
396
397
398 /*--------------------------------------------------------------------------*/
399 /* モジュール名 main                                                        */
400 /* 処理概要     メイン処理                                                  */
401 /* 引数         なし                                                        */
402 /* 戻り値       なし                                                        */
403 /*--------------------------------------------------------------------------*/
404 int main( void )
405 {
406
407       /*-----------------*/
408       /* H8のポート初期化 */
409       /*-----------------*/
410          Init_Port();
411
412
413       /*-----------------*/
414       /* H8タイマー初期化 */
415       /*-----------------*/
416          Init_H8();
417
418
419       /*--------------------------*/
420       /* オープニング(山形電波固有) */
421       /*--------------------------*/
422          openning08(); /* プログラムスタートの知らせ */
423
424
425       /*------------------------------------------------------*/
426       /* 課題処理プログラム                                   */
427       /*------------------------------------------------------*/
428
429            /*-----------------*/
430            /* AD初期化        */
```

```
431             /*-----------------*/
432                Init_ad();
433
434             /*-----------------*/
435             /* 表示先設定      */
436             /*-----------------*/
437                SEG_SELECT = LED;                /* 表示先はLED */
438
439             /*-------------------------*/
440             /* 割り込みにて表示スタート */
441             /*-------------------------*/
442                start_warikomi();
443
444
445                while(1) {
446
447                      start_ad();                /* AD変換スタート     */
448                      end_ad();                  /* AD変換終了チェック */
449                      psd_data = get_ad8();      /* psd読み込み */
450
451                }/* end of while(1) */
452
453 }
```