```

1 /*-----*/
2 /*  ものづ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4 物体の位置に相当するLEDか7セグメントLEDを光センサで切り替え */
7 /*      て点灯させる。 */
8 /*-----*/
9 /*  センサーの前に物体を置き、その位置を光センサに光が当たっているのなら */
10 /*  LEDの点灯位置を、光が当たっていないなら7セグメントLEDで数値点灯 */
11 /*  させる。次に物体を移動させたときにLEDの点灯位置、7セグメントLEDの */
12 /*  数値も変化させる。範囲、測定位置と点灯LEDは課題2と同じとする。 */
13 /*  7セグメントLEDの数値パターンは課題3と同じとする。 */
14 /*-----*/
15 /* 初期状態 */
16 /*      (1) センサは1m以上前に何もおいていない。 */
17 /*      (2) トグルスイッチはLの位置にある。 */
18 /*      (3) タクトスイッチは押されていない。 */
19 /*      (4) 光センサは光を受光している。 */
20 /*      (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した瞬間 */
21 /*      に行うものとする。 */
22 /*-----*/
23 /* 今後確認 */
24 /*  距離のデータはあらかじめ測定したものを使用した。 */
25 /*  LED,7segLED点灯処理は割込処理とした。 */
26 /*-----*/
27 /* 山形電波工業高等学校 情報技術科 & 電子工学科 */
28 /*  Ver. 1.0 平成20年5月 6日(火) */
29 /*  Ver. 2.0 平成20年5月10日(土) */
30 /*-----*/
31 /*-----*/
32 /*#include "comment08.h"
33
34 /*-----*/
35 /* 組み込みファイル定義 */
36 /*-----*/
37 #include <3048.h>
38 #include "macro08.h"
39
40
41 /*-----*/
42 /* グローバル変数宣言(どの関数からも参照、変更できる) */
43 /*-----*/
44 unsigned char psd_data; /* 割り込みで表示処理する */
45
46 unsigned char ledort7seg_flag = 0; /* 課題4,6 LED or 7segLEDどちらに表示するか */
47
48
49 /*-----*/
50 /* このプログラム中で使用する関数の記述 */
51 /*-----*/
52 /*#include "dempalib08.h"
53
54
55 /*-----*/
56 /* モジュール名 Init_Port */
57 /* 処理概要 H8 ポート初期化 */
58 /* 引数 なし */
59 /* 戻り値 なし */
60 /*-----*/
61 void Init_Port( void )
62 {
63     /* I/Oポート設定 0:入力 1:出力 */
64     P6.DDR = 0x00; /* on bord ディップスイッチ */
65                  /* 0x00 = 0000 0000 全bit入力設定 */
66
67     P4.DDR = 0xff; /* 事前配布基板(1) LED & 7segLED */
68                  /* 0xff = 1111 1111 全bit出力設定 */
69
70     P4.DR.BYTE = 0x00;
71
72     PA.DDR = 0xff; /*
73                  /* 0x00 = 0000 0000 全bit入力設定 */
74
75     PA.DR.BYTE = 0x00;
76
77     PB.DDR = 0xff; /*
78                  /* 0xff = 1111 1111 全bit出力設定 */
79
80     PB.DR.BYTE = 0x00;
81
82     P5.DDR = 0xff; /* 事前配布基板(1) LED or 7segLED指示端子 */
83                  /* 0xff = 1111 1111 全bit出力設定 */
84
85     P5.DR.BYTE = 0x00;
86
87     /* 当日作成基板はP7ポート(P7は入力専用なので設定なし) */
88 }

```

```

87 /*-----*/
88 /* モジュール名 Init_H8 */
89 /* 処理概要 H8 タイマー初期化 */
90 /* 引数 なし */
91 /* 戻り値 なし */
92 /*-----*/
93 /* 参考 H8/3048f(旧cpu) 14.7854MHz */
94 /* 16ms 29490 */
95 /* 10ms 18432 */
96 /* 8ms 14745 */
97 /* 5ms 9216 */
98 /* 1ms 1843 */
99 /*-----*/
100 void Init_H8( void )
101 {
102     EI; /* 割り込み許可 */
103
104     /* タイマ設定 */
105     ITU.TMDR.BIT.PWM1 = 0; /* ITU1:通常動作 */
106     ITU.TMDR.BIT.PWM2 = 0; /* ITU2:通常動作 */
107     ITU.TMDR.BIT.PWM3 = 1; /* ITU3:PWMモード */
108     ITU.TMDR.BIT.PWM4 = 0; /* ITU4:通常動作 */
109
110     /* iTu1 タイマ設定 */
111     ITU1.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
112     ITU1.TCR.BIT.CKEG = 0; /* クロックエッジ */
113     ITU1.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
114     ITU1.GRA = 18432; /* 割り込みインターバル 1.8432MHz/18432=100Hz */
115     ITU1.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
116
117     /* iTu2 タイマ設定 */
118     ITU2.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
119     ITU2.TCR.BIT.CKEG = 0; /* クロックエッジ */
120     ITU2.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
121     ITU2.GRA = 18432; /* 割り込みインターバル 1.8432MHz/18432=100Hz */
122     ITU2.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
123
124     /* iTu3 モータタイマ設定 */
125     ITU3.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
126     ITU3.TCR.BIT.CKEG = 0; /* クロックエッジ */
127     ITU3.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
128     ITU3.GRA = PWM_CYCLE; /* モータPWM周期 1.8432MHz/1843=1kHz */
129     ITU3.GRB = 0; /* モータPWM DUTY 0% */
130
131     /* タイマ設定 */
132     ITU4.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
133     ITU4.TCR.BIT.CKEG = 0; /* クロックエッジ */
134     ITU4.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
135     ITU4.GRA = 18432; /* 割り込みインターバル 1.8432MHz/18432=100Hz */
136     ITU4.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
137
138     /* タイマスタート */
139     ITU.TSTR.BIT.STRO = 0; /* まだスタートしない */
140     ITU.TSTR.BIT.STR1 = 0; /* まだスタートしない */
141     ITU.TSTR.BIT.STR2 = 0; /* まだスタートしない */
142     ITU.TSTR.BIT.STR3 = 0; /* まだスタートしない */
143     ITU.TSTR.BIT.STR4 = 0; /* まだスタートしない */
144 }
145
146 /*-----*/
147 /* モジュール名 start_itu0 */
148 /* 処理概要 タイマーitu4のスタート */
149 /* 引数 なし */
150 /* 戻り値 なし */
151 /*-----*/
152 void start_warikomi( void )
153 {
154     ITU.TSTR.BIT.STR4 = 1; /* itu4スタート */
155 }
156
157 /*-----*/
158 /* モジュール名 課題4用 int_imia4 */
159 /* 処理概要 */
160 /* 引数 なし */
161 /* 戻り値 なし */
162 /*-----*/
163 void int_imia4( void )
164 {
165     unsigned char w_data; /* コンパイラ不良? warning対策 */
166 }
167
168
169
170
171
172

```

```

173         if( ledort7seg_flag == T7SEG ) {
174
175             /*-----*/
176             /* 7セグメントLED出力 */
177             /*-----*/
178             SEG_SELECT = T7SEG;
179
180             if( psd_data >= 117 ) {
181                 KIBAN081 = tendo[ 1 ];
182
183             } else if( psd_data >= 80 ) {
184                 KIBAN081 = tendo[ 2 ];
185
186             } else if( psd_data >= 54 ) {
187                 KIBAN081 = tendo[ 3 ];
188
189             } else if( psd_data >= 41 ) {
190                 KIBAN081 = tendo[ 4 ];
191
192             } else if( psd_data >= 30 ) {
193                 KIBAN081 = tendo[ 5 ];
194
195             } else if( psd_data >= 26 ) {
196                 KIBAN081 = tendo[ 6 ];
197
198             } else if( psd_data >= 23 ) {
199                 KIBAN081 = tendo[ 7 ];
200
201             } else if( psd_data >= 20 ) {
202                 KIBAN081 = tendo[ 8 ];
203
204             } else {
205                 KIBAN081 = -0x00;
206
207             }
208
209     } else {
210         /*-----*/
211         /* LED出力 */
212         /*-----*/
213         SEG_SELECT = LED;
214
215         if( psd_data >= 117 ) {
216             KIBAN081 = -0x01;
217
218         } else if( psd_data >= 80 ) {
219             KIBAN081 = -0x02;
220
221         } else if( psd_data >= 54 ) {
222             KIBAN081 = -0x04;
223
224         } else if( psd_data >= 41 ) {
225             KIBAN081 = -0x08;
226
227         } else if( psd_data >= 30 ) {
228             KIBAN081 = -0x10;
229
230         } else if( psd_data >= 26 ) {
231             KIBAN081 = -0x20;
232
233         } else if( psd_data >= 23 ) {
234             KIBAN081 = -0x40;
235
236         } else if( psd_data >= 20 ) {
237             /*KIBAN081 = -0x80; warning */
238             w_data = 0x80;
239             KIBAN081 = -w_data;
240
241         } else {
242             KIBAN081 = -0x00;
243
244         }
245
246     }
247
248
249     /*-----*/
250     /* reset interrupt request */
251     /*-----*/
252     ITU4.TSR.BIT.IMFA = 0;
253 }
254 }
255
256
257 /*-----*/
258 /* モジュール名 wait */

```

```

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

```

```

345 /* モジュール名 end_ad */
346 /* 処理概要 A/D変換終了 */
347 /* 引数 なし */
348 /* 戻り値 なし */
349 /*-----*/
350 void end_ad( void )
351 {
352     while( AD.CSR.BIT.ADF != 1 ){ /* AD変換終了チェック */
353
354 }
355
356
357 /*-----*/
358 /* モジュール名 get_ad8 */
359 /* 処理概要 データ読み込み */
360 /* 引数 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) */
361 /* 戻り値 変換データを確定する。(下位2bitゴミとしてカット) */
362 /* 引数 なし */
363 /* 戻り値 8bit data */
364 /* 1111 1111 11-- ---- : 変換後データ */
365 /* ---- ---- 1111 1111 : 8bitシフト後データ */
366 /*
367 /* ---- ---- 0000 0000 : 0x00 0
368 /* :
369 /* ---- ---- 1111 1111 : 0xff 255 0~255 の 256通りのデータ */
370 /*-----*/
371 unsigned char get_ad8( void )
372 {
373     unsigned char indata; /* unsigned intで 0 ~ 255 まで扱える */
374
375     indata = AD.DRA >>8; /* 上位10bitを右に8bitシフトする。下位2bit無視 */
376 /* 1111 1111 1100 0000 */
377 /* ---- ---- 1111 1111 */
378
379     /* indata = AD.DRB >>8; */ /* 上位10bitを右に8bitシフトする。 */
380     /* indata = AD.DRC >>8; */ /* 上位10bitを右に8bitシフトする。 */
381     /* indata = AD.DRD >>8; */ /* 上位10bitを右に8bitシフトする。 */
382     // indata = indata & 0xff;
383
384     return indata;
385 }
386
387
388 /*-----*/
389 /* モジュール名 main */
390 /* 処理概要 メイン処理 */
391 /* 引数 なし */
392 /* 戻り値 なし */
393 /*-----*/
394 int main( void )
395 {
396
397     /*-----*/
398     /* H8のポート初期化 */
399     /*-----*/
400     Init_Port();
401
402
403     /*-----*/
404     /* H8タイマー初期化 */
405     /*-----*/
406     Init_H8();
407
408
409     /*-----*/
410     /* オープニング(山形電波固有) */
411     /*-----*/
412     opening08(); /* プログラムスタートの知らせ */
413
414
415     /*-----*/
416     /* 課題処理プログラム */
417     /*-----*/
418
419     /*-----*/
420     /* AD初期化 */
421     /*-----*/
422     Init_ad();
423
424     /*-----*/
425     /* 表示先設定 */
426     /*-----*/
427     SEG_SELECT = LED; /* 表示先はLED */
428
429     /*-----*/
430     /* 割り込みにて表示スタート */

```

```
431     /*-----*/
432     start_warikomi();
433
434     while(1) {
435
436         ledort7seg_flag = DIP_SW_HI;
437
438         start_ad();           /* AD変換スタート */
439         end_ad();           /* AD変換終了チェック */
440         psd_data = get_ad8(); /* psd読み込み */
441
442     } /* end of while(1) */
443
444 }
445 }
```