

```

0:/* ----- */
1:/* */
2:/*   ものつくりコンテスト06 電子回路組立部門 */
3:/*   山形電波工高csc & 情報技術科 用プログラム */
4:/* */
5:/*                                     Ver 1.0 2005.05.29 */
6:/*                                     1.1 2006.04.18 */
7:/*                                     1.2 2006.05.10 */
8:/*                                     1.3 2006.05.29 */
9:/*                                     1.4 2006.05.31 */
10:/*                                    1.5 2006.06.03 */
11:/* ----- */
12:/* ----- */
13:/* */
14:/* [基本仕様] */
15:/* */
16:/*   C P U   H8/3048F (14.7456MHz) */
17:/* */
18:/*   事前配布基板(1) 7segLEDボード & DCモータ */
19:/* */
20:/*   事前配布基板(2) ステッピングモータ & LED */
21:/* */
22:/*   当日製作基板   トグルスイッチ & フォトインタラプタ */
23:/* ----- */
24:/* ----- */
25:
26:
27:/* ----- */
28:/*   標準 I/Oアサイン */
29:/* ----- */
30:/* ----- */
31:/* P6-0   on bord dipスイッチ 下位   IN */
32:/* P6-1   on bord dipスイッチ 上位   IN */
33:/* P6-2   未使用(使用できず)         OUT */
34:/* P6-3   未使用(使用できず)         OUT */
35:/* P6-4   未使用(使用できず)         OUT */
36:/* P6-5   未使用(使用できず)         OUT */
37:/* P6-6   未使用(使用できず)         OUT */
38:/* P6-7   未使用(使用できず)         OUT */
39:/* ----- */
40:/* ----- */
41:
42:
43:/* ----- */
44:/*   配布ボード(1) 7セグメントLEDボード */
45:/* ----- */
46:/* ----- */
47:/* P4-0   7seg LED a                   OUT */
48:/* P4-1   7seg LED b                   OUT */
49:/* P4-2   7seg LED c                   OUT */
50:/* P4-3   7seg LED d                   OUT */
51:/* P4-4   7seg LED e                   OUT */
52:/* P4-5   7seg LED f                   OUT */
53:/* P4-6   7seg LED g                   OUT */
54:/* P4-7   7seg LED Dp (注 使用せず)  OUT */
55:/* ----- */
56:/* ----- */
57:/* ----- */
58:/* P5-0   7seg LED 下位 表示指示     OUT */
59:/* P5-1   7seg LED 上位 表示指示     OUT */
60:/* P5-2   使用せず                   OUT */
61:/* P5-3   "                            OUT */
62:/* P5-4   "                            OUT */
63:/* P5-5   "                            OUT */
64:/* P5-6   "                            OUT */
65:/* P5-7   "                            OUT */
66:/* ----- */
67:/* ----- */
68:/* ----- */
69:/* PB-0   DC motor                     OUT(PWM) */
70:/* PB-1   使用せず                   OUT */
71:/* PB-2   "                            OUT */
72:/* PB-3   "                            OUT */
73:/* PB-4   "                            OUT */
74:/* PB-5   "                            OUT */
75:/* PB-6   "                            OUT */
76:/* PB-7   "                            OUT */
77:/* ----- */
78:/* ----- */
79:
80:

```

```

81:/* ----- */
82:/* 配布ボード(2) ステッピングモータ & LEDボード */
83:/* ----- */
84:/* ----- */
85:/* PA-0 ステッピングモータ A OUT */
86:/* PA-1 ステッピングモータ B OUT */
87:/* PA-2 ステッピングモータ C OUT */
88:/* PA-3 ステッピングモータ D OUT */
89:/* PA-4 LED 1 OUT */
90:/* PA-5 LED 2 OUT */
91:/* PA-6 LED 3 OUT */
92:/* PA-7 LED 4 OUT */
93:/* ----- */
94:/* ----- */
95:
96:
97:/* ----- */
98:/* 当日製作ボード トグルSW & フォトインタラプタ */
99:/* ----- */
100:/* P7-0 トグルスイッチ IN */
101:/* P7-1 フォトインタラプタ IN */
102:/* P7-2 使用せず IN */
103:/* P7-3 " IN */
104:/* P7-4 " IN */
105:/* P7-5 " IN */
106:/* P7-6 " IN */
107:/* P7-7 " IN */
108:/* ----- */
109:
110:
111:
112:/* ----- */
113:/* 組み込みファイル定義 */
114:/* ----- */
115:#include <3048.h>
116:
117:
118:/* ----- */
119:/* マクロ定義 */
120:/* ----- */
121: #define DIP_SW ( (_BYTE)( (~P6.DR.BYTE) & 0x03) ) /* on bord ディップスイッチ */
122: #define DIP_SW_HI ( (_BYTE)( (~P6.DR.BYTE) >> 1) & 0x01 ) /* on bord ディップスイッチ上位 */
123: #define DIP_SW_LO ( (_BYTE)( (~P6.DR.BYTE) & 0x01) ) /* on bord ディップスイッチ下位 */
124:
125: #define PWM_CYCLE 1843 /* 1ms モータPWM */
126:
127:
128:
129:
130: /* ----- */
131: /* 事前配布基板(1) 7セグメントLED に関する記述 */
132: /* ----- */
133: #define KIBAN1 P4.DR.BYTE /* この記述によりプログラム中では 基板(1)を KIBAN1
133:として扱うことができる
133:*/
134:
135: #define DCMOTOR PB.DR.BIT.B0 /* 06年追加(仮) */
136:
137: #define SEG_SELECT P5.DR.BYTE
138:
139: #define LO_SEG 1
140: #define HI_SEG 2
141: #define HI_LO_SEG 3
142:
143:
144:
145: /* ----- */
146: /* 事前配布基板(2) ステッピングモータ & LED に関する記述 */
147: /* ----- */
148: #define KIBAN2 PA.DR.BYTE /* この記述によりプログラム中では 基板(2)を KIBAN2
148:として扱うことができる
148:*/
149:
150: #define MOTOR PA.DR.BYTE /* 本当は下位4bit */
151: #define LED PA.DR.BYTE /* 本当は上位4bit */
152:
153: #define LED1 PA.DR.BIT.B4 /* 配布基板(2) LED1 */
154: #define LED2 PA.DR.BIT.B5 /* 配布基板(2) LED2 */
155: #define LED3 PA.DR.BIT.B6 /* 配布基板(2) LED3 */
156: #define LED4 PA.DR.BIT.B7 /* 配布基板(2) LED4 */
157:

```

```

158: #define MLED PA.DR.BIT.B7 /* 配布基板(2) LED4 */
159:
160:
161:
162: /*-----*/
163: /* 当日製作基板 トグルスイッチ & フォトインタラプタ */
164: /*-----*/
165: #define KIBAN3 P7.DR.BYTE /* この記述によりプログラム中では 基板(3)を KIBAN3
165:として扱うことができる
165:*/
166:
167: #define SW P7.DR.BIT.B0 /* トグルスイッチ */
168: #define PHOTO P7.DR.BIT.B1 /* フォトインタラプタ */
169:
170: //#define PUSH_SW ( PA.DR.BIT.B7 == 1 ) /* トグルスイッチの押下でTRUE */
171:
172:
173:
174: /*-----*/
175: /* その他定義 */
176: /*-----*/
177: #define ON 1
178: #define OFF 0
179:
180:
181:
182: /*-----*/
183: /* グローバル変数宣言(どの関数からも参照できるが変更はできない) */
184: /*-----*/
185: /* 7セグメントLED表示データ */
186:const unsigned char tendo[ 18 ] = {
187:
188:     0xc0, /* 0 1100 0000 */
189:     0xf9, /* 1 1111 1001 */
190:     0xa4, /* 2 1010 0100 */
191:     0xb0, /* 3 1011 0000 */
192:     0x99, /* 4 1001 1001 */
193:     0x92, /* 5 1001 0010 */
194:     0x82, /* 6 1000 0010 */
195:     0xf8, /* 7 1111 1000 */
196:     0x80, /* 8 1000 0000 */
197:     0x90, /* 9 1001 0000 */
198:     0x88, /* a 1000 1000 */
199:     0x83, /* b 1000 0011 */
200:     0xa7, /* c 1010 0111 */
201:     0xa1, /* d 1010 0001 */
202:     0x86, /* e 1110 0110 */
203:     0x8e, /* f 1000 1110 */
204:
205:     0x89, /* H 1000 1001 */
206:     0xc7 /* L 1100 0111 */
207:
208:};
209:
210:
211: /*-----*/
212: /* ステッピングモータ励磁信号(1相,2相,1-2相励磁対応) */
213: /*-----*/
214:const unsigned char RotateData1[ 16 ][ 4 ]={
215:     { 0x00, 0x00, 0x00, 0x00 },
216:     { 0x11, 0x22, 0x44, 0x88 },
217:     { 0x22, 0x44, 0x88, 0x11 },
218:     { 0x33, 0x66, 0xcc, 0x99 },
219:     { 0x44, 0x88, 0x11, 0x22 },
220:     { 0x55, 0xaa, 0x55, 0xaa },
221:     { 0x66, 0xcc, 0x99, 0x33 },
222:     { 0x77, 0xee, 0xdd, 0xbb },
223:     { 0x88, 0x11, 0x22, 0x44 },
224:     { 0x99, 0x33, 0x66, 0xcc },
225:     { 0xaa, 0x55, 0xaa, 0x55 },
226:     { 0xbb, 0x77, 0xee, 0xdd },
227:     { 0xcc, 0x99, 0x33, 0x66 },
228:     { 0xdd, 0xbb, 0x77, 0xee },
229:     { 0xee, 0xdd, 0xbb, 0x77 },
230:     { 0xff, 0xff, 0xff, 0xff }
231:
232:};
233:
234:const unsigned char RotateData[ 16 ][ 4 ]={
235:     { 0x00, 0x00, 0x00, 0x00 },
236:     { 0x81, 0x42, 0x24, 0x18 },

```

```

237:     { 0x42, 0x24, 0x18, 0x81 },
238:     { 0x93, 0xc6, 0x6c, 0x39 },
239:     { 0x24, 0x18, 0x81, 0x42 },
240:     { 0xa5, 0x5a, 0xa5, 0x5a },
241:     { 0x36, 0x9c, 0xc9, 0x63 },
242:     { 0xb7, 0xde, 0xed, 0x7b },
243:     { 0x18, 0x81, 0x42, 0x24 },
244:     { 0xc9, 0x63, 0x36, 0x9c },
245:     { 0x5a, 0xa5, 0x5a, 0xa5 },
246:     { 0xdb, 0xe7, 0x7e, 0xbd },
247:     { 0x6c, 0x39, 0x93, 0xc6 },
248:     { 0xed, 0x7b, 0xb7, 0xde },
249:     { 0x7e, 0xbd, 0xdb, 0xe7 },
250:     { 0xff, 0xff, 0xff, 0xff }
251: };
252: };
253:
254:
255: /*-----*/
256: /* グローバル変数宣言(どの関数からも参照、変更できる) */
257: /*-----*/
258: unsigned int shadan_kaisu = 0;          /* 課題1 フォトインタラプタ遮断回数 初期値 0 */
259:                                         /* 課題3          "          "          */
260:
261:
262: unsigned int w_shadan_kaisu = 0;       /* 06課題 ストップウオッチカウンタ 初期値 0 */
263:
264: unsigned int seg = 0;                  /* 06課題 ストップウオッチカウンタ 初期値 0 */
265:
266:
267: /*-----*/
268: /* このプログラム中で使用する関数の宣言 */
269: /*-----*/
270:
271: void Init_Port( void );                /* H8 ポート初期化 */
272: void Init_H8( void );                  /* H8 内部I/O初期化 */
273:
274: void wait( int iTimer );               /* ウェイト */
275:
276: void opening( void );                 /* オープニング */
277: void seg_cls( void );                 /* 7セグメントLEDボード消去 */
278:
279:
280:
281: void kadai1( void );                  /* 事前公開課題1 */
282: void brink( int brink_kaisu );        /* 7セグメントLED 点滅 */
283:
284:
285: void kadai2( void );                  /* 事前公開課題2 */
286: unsigned char rrotate(unsigned char Data, unsigned char shadan_kaisu);
287: unsigned char lrotate(unsigned char Data, unsigned char shadan_kaisu);
288:
289:
290: void kadai3( void );                  /* 事前公開課題3 */
291:
292: void kadai61( void );                 /* 事前課題06年(1) */
293:
294:
295:
296: /*-----*/
297: /* このプログラム中で使用する関数の記述 */
298: /*-----*/
299:
300:
301: /*-----*/
302: /* void Init_Port( void ) */
303: /* 機能: H8 ポート初期化 */
304: /*-----*/
305: /* I/Oポートの初期設定を行う。 */
306: /*-----*/
307: void Init_Port( void )
308: {
309:     /* I/Oポート設定 0:入力 1:出力 */
310:     P6.DDR = 0x00;                    /* on bord ディップスイッチ */
311:                                         /* 0x00 = 0000 0000 全bit入力設定 */
312:
313:
314:     PA.DDR = 0xff;                    /* 事前配布基板(2) ステッピングモータ & LED */
315:                                         /* 0xff = 1111 1111 全bit出力設定 */
316:     PA.DR.BYTE = 0x00;
317:

```

```

318:
319:     P4.DDR = 0xff;          /* 事前配布基板(1) 7セグメントLED */
320:                               /* 0xff = 1111 1111 全bit出力設定 */
321:     P4.DR.BYTE = 0x00;
322:
323:
324:     PB.DDR = 0xff;          /* 事前配布基板(1) DCモータ */
325:                               /* 0xff = 1111 1111 全bit出力設定 */
326:     PB.DR.BYTE = 0x00;
327:
328:
329:     P5.DDR = 0xff;          /* 事前配布基板(1) 7セグメントLED指示端子 */
330:                               /* 0xff = 1111 1111 全bit出力設定 */
331:     P5.DR.BYTE = 0x00;
332:
333: }
334:
335:
336:
337: /*-----*/
338: void Init_H8( void )      /*
339: /*-----*/
340: /* 機能 : H8 内部I/O初期化 */
341: /*-----*/
342: /* I/Oポートとタイマの初期設定を行う。 */
343: /*
344: /* 参考 */
345: /*      16ms   29490 */
346: /*      10ms   18432 */
347: /*      8ms    14745 */
348: /*      5ms    9216  */
349: /*      1ms    1843  */
350: /*-----*/
351: void Init_H8( void )
352: {
353:     EI; /* 割り込み許可 */
354:
355:
356:
357: /* タイマ設定 */
358: ITU.TMDR.BIT.PWMO = 0; /* ITU0:通常動作 */
359: ITU.TMDR.BIT.PWM1 = 0; /* ITU1:通常動作 */
360: ITU.TMDR.BIT.PWM2 = 0; /* ITU2:通常動作 */
361: ITU.TMDR.BIT.PWM3 = 1; /* ITU3:PWMモード */
362: ITU.TMDR.BIT.PWM4 = 0; /* ITU4:通常動作 */
363:
364:
365: /* iTu0 タイマ設定 */
366: ITU0.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
367: ITU0.TCR.BIT.CKEG = 0; /* クロックエッジ */
368: ITU0.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
369: ITU0.GRA = 1843; /* 割り込みインターバル 1.8432MHz/1843=100Hz */
370: ITU0.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
371:
372:
373: /* iTu1 タイマ設定 */
374: ITU1.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
375: ITU1.TCR.BIT.CKEG = 0; /* クロックエッジ */
376: ITU1.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
377: ITU1.GRA = 18432; /* 割り込みインターバル 1.8432MHz/18432=100Hz */
378: ITU1.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
379:
380:
381: /* iTu2 タイマ設定 */
382: ITU2.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
383: ITU2.TCR.BIT.CKEG = 0; /* クロックエッジ */
384: ITU2.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
385: ITU2.GRA = 18432; /* 割り込みインターバル 1.8432MHz/18432=100Hz */
386: ITU2.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
387:
388:
389: /* iTu3 モータタイマ設定 */
390: ITU3.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */
391: ITU3.TCR.BIT.CKEG = 0; /* クロックエッジ */
392: ITU3.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
393: ITU3.GRA = PWM_CYCLE; /* 左モータPWM周期 1.8432MHz/1843=1kHz */
394: ITU3.GRB = 0; /* 左モータPWM DUTY 0% */
395:
396:
397: /* タイマ設定 */
398: ITU4.TCR.BIT.CCLR = 1; /* カウンタクリア要因 */

```

```

399: ITU4.TCR.BIT.CKEG = 0; /* クロックエッジ */
400: ITU4.TCR.BIT.TPSC = 3; /* タイマプリスケラ 1.8432MHz */
401: ITU4.GRA = 18432; /* 割り込みインターバル 1.8432MHz/18432=100Hz */
402: ITU4.TIER.BIT.IMIEA = 1; /* IMFAフラグによる割り込み許可 */
403:
404:
405: /* タイマスタート */
406: ITU.TSTR.BIT.STRO = 0; /* まだスタートしない */
407: ITU.TSTR.BIT.STR1 = 0; /* まだスタートしない */
408: ITU.TSTR.BIT.STR2 = 0; /* まだスタートしない */
409: ITU.TSTR.BIT.STR3 = 1; /* itu3スタート */
410: ITU.TSTR.BIT.STR4 = 0; /* まだスタートしない */
411:
412:}
413:
414:
415:
416:/*-----*/
417:/* モジュール名 wait */
418:/* 処理概要 タイマー */
419:/* 引数 タイマー値 */
420:/* 1: 10[ms] */
421:/* 10: 100[ms] = 0.1[s] */
422:/* 50: 500[ms] = 0.5[s] */
423:/* 100:1000[ms] = 1.0[s] */
424:/*-----*/
425:void wait( int iTimer )
426:{
427:    int i;
428:
429:    while( iTimer ) {
430:
431:        for( i = 0; i < 5000; i++ ); /* H8/3048f */
432:        /*for( i = 0; i < 8333; i++ );*/ /* H8/3048f-one */
433:        iTimer--;
434:
435:    }
436:
437:}
438:
439:
440:
441:/*-----*/
442:/* void speed( int accele ) */
443:/*-----*/
444:/* 機能 : 速度制御 */
445:/* 引数 : accele = モータ速度 0 ~ 100 */
446:/*-----*/
447:void speed( int accele )
448:{
449:    unsigned int speed_max = 29490;
450:
451:    /* モータ */
452:    ITU3.GRB = speed_max / 100 * accele;
453:
454:}
455:
456:
457:
458:/*-----*/
459:/* モジュール名 k1_test */
460:/* 処理概要 事前配布ボード1 (7segLEDボード)テスト(1) */
461:/* 7segLED単純表示 */
462:/* 引数 なし */
463:/* 戻り値 なし */
464:/*-----*/
465:void k1_test1( void )
466:{
467:    unsigned int i;
468:
469:
470:    KIBAN1 = 0xff; /* 7セグメントLED 消去 */
471:
472:
473:    /*-----*/
474:    /* 7segLED1(下位) 表示 */
475:    /*-----*/
476:    SEG_SELECT = LO_SEG; /* 表示は下位のみ 0x01:0000 0001 */
477:    for( i = 0; i < 16; i++ ) {
478:        KIBAN1 = tendo[ i ];
479:        wait( 50 );

```

```

480:     }
481:     KIBAN1 = 0x7f; /* 7セグメントLED 消去 */
482:     wait( 50 );
483:
484:
485:     /*-----*/
486:     /* 7segLED2(上位) 表示 */
487:     /*-----*/
488:     SEG_SELECT = HI_SEG; /* 表示は下位のみ 0x02:0000 0010 */
489:     for( i = 0; i < 16; i++ ) {
490:         KIBAN1 = tendo[ i ];
491:         wait( 50 );
492:     }
493:     KIBAN1 = 0x7f; /* 7セグメントLED 消去 */
494:     wait( 50 );
495:
496:
497:     /*-----*/
498:     /* 7segLED2(上位) 表示 */
499:     /*-----*/
500:     SEG_SELECT = HI_LO_SEG; /* 表示は上位下位両方向 0x03:0000 0011 */
501:     wait( 100 );
502:
503:     for( i=0; i<16; i++ ) {
504:         KIBAN1 = tendo[ i ];
505:         wait( 50 );
506:     }
507:     KIBAN1 = 0x7f;
508:
509:     while(1){}
510:
511: }
512:
513:
514:
515: /*-----*/
516: /* モジュール名 k1_test2 */
517: /* 処理概要 事前配布ボード1 (7segLEDボード)テスト(2) */
518: /* 2桁の10進数表示 00~99まで */
519: /* 引数 なし */
520: /* 戻り値 なし */
521: /*-----*/
522: void k1_test2( void )
523: {
524:     unsigned int i, j;
525:
526:
527:     for( i = 0; i < 10; i++ ) {
528:
529:         for( j = 0; j < 10; j++ ) {
530:
531:             /*-----*/
532:             /* 7segLED1(下位) 表示 */
533:             /*-----*/
534:             SEG_SELECT = LO_SEG; /* 表示は下位のみ 0x01:0000 0001 */
535:             KIBAN1 = tendo[ i ];
536:             wait( 3 );
537:
538:             /*-----*/
539:             /* 7segLED2(上位) 表示 */
540:             /*-----*/
541:             if( i != 0 ) {
542:                 SEG_SELECT = HI_SEG; /* 表示は上位のみ 0x02:0000 0010 */
543:                 KIBAN1 = tendo[ i ];
544:                 wait( 3 );
545:             } /* end of if */
546:
547:         } /* end of for( j ) */
548:
549:     } /* end of for( i ) */
550:
551: }
552:
553:
554:
555: /*-----*/
556: /* モジュール名 k1_test3 */
557: /* 処理概要 事前配布ボード1 (7segLEDボード)テスト(3) */
558: /* 2桁の10進数表示 99~00まで */
559: /* 引数 なし */
560: /* 戻り値 なし */

```

```

561:/*-----*/
562:void k1_test3( void )
563:{
564:    int i, j, k;
565:
566:
567:    for( i = 99; i >= 0; i-- ) {
568:
569:
570:        j= i / 10;      /* 10の位を求める */
571:        k = i - j * 10; /* 1の位を求める */
572:
573:
574:        /*-----*/
575:        /* 7segLED1(下位)(1の位)表示 */
576:        /*-----*/
577:        SEG_SELECT = LO_SEG; /* 表示は下位のみ 0x01:0000 0001 */
578:        KIBAN1 = tendo[ k ];
579:        wait( 3 );
580:
581:
582:        /*-----*/
583:        /* 7segLED1(上位)(10の位)表示 */
584:        /*-----*/
585:        if( i != 0 ) {
586:            SEG_SELECT = HI_SEG; /* 表示は上位のみ 0x02:0000 0010 */
587:            KIBAN1 = tendo[ i ];
588:            wait( 3 );
589:        } /* end of if */
590:
591:    } /* end of for( i ) */
592:
593:}
594:
595:
596:
597:/*-----*/
598:/* モジュール名 k1_test4 */
599:/* 処理概要     事前配布ボード1 (7segLEDボード)テスト(1) */
600:/*             7segLED単純表示 & DCモータ単体テスト */
601:/* 引数         なし */
602:/* 戻り値       なし */
603:/*-----*/
604:void k1_test4( void )
605:{
606:    unsigned int i;
607:
608:
609:    //KIBAN1 = 0xff; /* 0xff : 1111 1111 */
610:    KIBAN1 = 0x7f; /* 0x7f : 0111 1111 */
611:    wait( 90 );
612:
613:
614:    speed(100); /* DCモータ高速回転 */
615:
616:
617:    /*-----*/
618:    /* 7segLED1(下位) 表示 */
619:    /*-----*/
620:    SEG_SELECT = LO_SEG; /* 表示は下位のみ 0x01:0000 0001 */
621:    for( i = 0; i < 10; i++ ) {
622:        KIBAN1 = tendo[ i ];
623:        wait( 50 );
624:    }
625:    KIBAN1 = 0x7f;
626:
627:
628:    speed(1); /* DCモータ低速回転 */
629:
630:
631:    /*-----*/
632:    /* 7segLED2(上位) 表示 */
633:    /*-----*/
634:    SEG_SELECT = HI_SEG; /* 表示は上位のみ 0x02:0000 0010 */
635:    for( i = 0; i < 10; i++ ) {
636:        KIBAN1 = tendo[ i ];
637:        wait( 50 );
638:    }
639:    KIBAN1 = 0x7f;
640:
641:

```

```

642:         speed(2); /* DCモータ中速 */
643:
644:
645:         /*-----*/
646:         /* 7segLED2(上位) 表示 */
647:         /*-----*/
648:         SEG_SELECT = HI_LO_SEG; /* 表示は上位下位両方向じ 0x03:0000 0011 */
649:         wait( 100 );
650:
651:         for( i=0; i<10; i++ ) {
652:             KIBAN1 = tendo[ i ];
653:             wait( 50 );
654:         }
655:         KIBAN1 = 0x7f;
656:
657:
658:         speed(100); /* DCモータ高速回転 */
659:
660:
661:         while(1){}
662:
663:
664: }
665:
666:
667:
668: /*-----*/
669: /* モジュール名 k2_test1 */
670: /* 処理概要     事前配布基板2 テスト(1) */
671: /*             ステッピングモータ右回転 */
672: /*             1相励磁 無限回転 LED同期 */
673: /* 引数         なし */
674: /* 戻り値       なし */
675: /*-----*/
676: void k2_test1( void )
677: {
678:
679:     while(1)
680:     {
681:
682:         KIBAN2 = 0x88;          /* 1000 1000 */
683:         wait( 10 );
684:         KIBAN2 = 0x44;          /* 0100 0100 */
685:         wait( 10 );
686:         KIBAN2 = 0x22;          /* 0010 0010 */
687:         wait( 10 );
688:         KIBAN2 = 0x11;          /* 0001 0001 */
689:         wait( 10 );
690:
691:     }
692:
693: }
694:
695:
696:
697: /*-----*/
698: /* モジュール名 k2_test2 */
699: /* 処理概要     事前配布基板2 テスト(1) */
700: /*             ステッピングモータ右回転 */
701: /*             2相励磁 無限回転 LED同期 */
702: /* 引数         なし */
703: /* 戻り値       なし */
704: /*-----*/
705: void k2_test2( void )
706: {
707:
708:     while(1)
709:     {
710:
711:         KIBAN2 = 0xcc;          /* 1100 1100 */
712:         wait( 10 );
713:         KIBAN2 = 0x66;          /* 0110 0110 */
714:         wait( 10 );
715:         KIBAN2 = 0x33;          /* 0011 0011 */
716:         wait( 10 );
717:         KIBAN2 = 0x99;          /* 1001 1001 */
718:         wait( 10 );
719:
720:     }
721:
722: }

```

```

723:
724:
725:
726:/*-----*/
727:/* モジュール名 k2_test3 */
728:/* 処理概要 事前配布基板2 テスト(3) */
729:/* ステッピングモータ右回転 */
730:/* 1相励磁 1回転 LED同期 */
731:/* 引数 なし */
732:/* 戻り値 なし */
733:/*-----*/
734:void k2_test3( void )
735:{
736:    unsigned int i;
737:
738:    // PA.DR.BYTE = 0x00; /* 励磁off */ /* ない方が良さそうだ */
739:    // wait(20);
740:
741:    for( i = 0; i < 5; i++ ) {
742:
743:        KIBAN2 = 0x88; /* 1000 1000 */
744:        wait( 10 );
745:        KIBAN2 = 0x44; /* 0100 0100 */
746:        wait( 10 );
747:        KIBAN2 = 0x22; /* 0010 0010 */
748:        wait( 10 );
749:        KIBAN2 = 0x11; /* 0001 0001 */
750:        wait( 10 );
751:    }/* end of for */
752:
753:    while(1){}
754:
755:}
756:
757:
758:
759:/*-----*/
760:/* モジュール名 k2_test4 */
761:/* 処理概要 事前配布基板2 テスト(4) */
762:/* ステッピングモータ 左回転 & 右回転 */
763:/* 1相励磁 1回転 LED同期 */
764:/* 引数 なし */
765:/* 戻り値 なし */
766:/*-----*/
767:void k2_test4( void )
768:{
769:    unsigned int i,j;
770:
771:    // PA.DR.BYTE = 0x00; /* 励磁off */ /* ない方が良さそうだ */
772:    // wait(20);
773:
774:    for( i = 0; i < 5; i++ ) {
775:
776:        for( j = 0; j < 5; j++ ) {
777:
778:            KIBAN2 = 0x88; /* 1000 1000 */
779:            wait( 10 );
780:            KIBAN2 = 0x44; /* 0100 0100 */
781:            wait( 10 );
782:            KIBAN2 = 0x22; /* 0010 0010 */
783:            wait( 10 );
784:            KIBAN2 = 0x11; /* 0001 0001 */
785:            wait( 10 );
786:
787:        }/* end of for(j) */
788:
789:    }/* end of for(i) */
790:
791:
792:    wait( 50 );
793:
794:
795:    for( i = 0; i < 5; i++ ) {
796:
797:        for( j = 0; j < 5; j++ ) {
798:
799:            KIBAN2 = 0x22; /* 0010 0010 */
800:            wait( 10 );
801:            KIBAN2 = 0x44; /* 0100 0100 */
802:            wait( 10 );
803:            KIBAN2 = 0x88; /* 1000 1000 */

```

```

804:         wait( 10 );
805:         KIBAN2 = 0x11;          /* 0001 0001 */
806:         wait( 10 );
807:
808:     } /* end of for(j) */
809:
810: } /* end of for(i) */
811:
812:
813:     while(1){
814:
815: }
816:
817:
818:
819: /*-----*/
820: /* モジュール名 k3_test */
821: /* 処理概要 当日製作基板テスト */
822: /* 引数 なし */
823: /* 戻り値 なし */
824: /*-----*/
825: void k3_test1( void )
826: {
827:     unsigned int  indata;
828:
829:     while(1) {
830:
831:         indata = KIBAN3 & 0x03; /* 0x03: 0000 0011 */
832:         indata = indata << 4; /* 下位2bitを左に4bitシフト */
833:         KIBAN2 = indata;
834:
835:     } /* end of while(1) */
836:
837: }
838:
839:
840:
841: /*-----*/
842: /* void int_imia0( void ) */
843: /*-----*/
844: /* 機能 : 7セグメントLED表示 */
845: /*-----*/
846: /* この関数はITU2のインターバル割込にて5ms毎に自動的に励起される。 */
847: /*-----*/
848: void int_imia0( void ) /* 10msインターバル割り込み */
849: {
850:
851:     int i, j, k;
852:
853:
854:     /*-----*/
855:     /* 10進数表示 */
856:     /*-----*/
857:     shadan_kaisu = w_shadan_kaisu / 10;
858:     j = shadan_kaisu / 10; /* 10の位を求める */
859:     k = shadan_kaisu - j * 10; /* 1の位を求める */
860:
861:
862:     /*-----*/
863:     /* 7segLED出力 */
864:     /*-----*/
865:
866:     if ( seg == 0 ) {
867:
868:         /*-----*/
869:         /* 7segLED1(下位)(1の位)表示 */
870:         /*-----*/
871:         SEG_SELECT = LO_SEG; /* 表示は下位(1の位) */
872:         KIBAN1 = tendo[ k ]; /* 遮断時間表示 */
873:         seg = 1;
874:
875:     } else {
876:         /*-----*/
877:         /* 7segLED1(上位)(10の位)表示 */
878:         /*-----*/
879:         SEG_SELECT = HI_SEG; /* 表示は上位(10の位) */
880:         KIBAN1 = tendo[ i ]; /* 遮断時間表示 */
881:         seg = 0;
882:     }
883:
884:     /*-----*/

```

```

885:      /* reset interrupt request */
886:      /*-----*/
887:      ITU0.TSR.BIT.IMFA = 0;
888:
889: }
890:
891:
892:
893: /*-----*/
894: /* void int_imia1( void ) */
895: /*-----*/
896: /* 機能：フォトインタラプタ遮断時間のカウント */
897: /*-----*/
898: /* この関数はITU1のインターバル割込にて10ms毎に自動的に励起される。*/
899: /*-----*/
900: void int_imia1( void ) /* 10msインターバル割込み */
901: {
902:
903:
904:     if ( SW ) { /* スイッチON */
905:
906:         if ( !PHOTO ) { /* フォトインタラプタが遮断されていれば */
907:
908:             w_shadan_kaisu++; /* カウント */
909:             if ( w_shadan_kaisu > 999 ) w_shadan_kaisu = 0;
910:
911:         } else {
912:
913:             /* 透過してればカウントはそのまま */
914:
915:         }
916:
917:     } else { /* スイッチoffならストップウォッチリセット */
918:
919:         w_shadan_kaisu = 0; /* リセット */
920:
921:     }
922:
923:
924:
925: /*-----*/
926: /* reset interrupt request */
927: /*-----*/
928:     ITU1.TSR.BIT.IMFA = 0;
929:
930: }
931:
932:
933:
934: /*-----*/
935: /* void int_imia2( void ) */
936: /*-----*/
937: /* 機能：事前公開課題3用 7セグメントLED表示 */
938: /*-----*/
939: /* この関数はITU2のインターバル割込にて10ms毎に自動的に励起される。*/
940: /*-----*/
941: void int_imia2( void ) /* 10msインターバル割込み */
942: {
943:
944:     int k;
945:
946:
947:     /* (注意) */
948:     /* 遮断回数表示は下1桁 */
949:     //if ( shadan_kaisu > 10 ) shadan_kaisu = 0; /* */
950:     if ( shadan_kaisu > 15 ) shadan_kaisu = 0; /* */
951:
952:
953:     k = shadan_kaisu; /* 1の位を求める */
954:
955:
956: /*-----*/
957: /* 7segLED ダイナミック点灯 (上位と下位を交互に表示) */
958: /*-----*/
959:
960:     if ( seg == 0 ) {
961:
962:         /*-----*/
963:         /* 7segLED1(下位)(1の位)表示 */
964:         /*-----*/
965:         SEG_SELECT = LO_SEG; /* 表示は下位(1の位) */

```

```

966:         KIBAN1 = tendo[ k ];
967:         seg = 1;          /* 次は上位桁表示  */
968:
969:     } else {
970:
971:         /*-----*/
972:         /* 7segLED1(上位)(10の位)表示 */
973:         /*-----*/
974:         SEG_SELECT = HI_SEG;    /* 表示は上位(10の位) */
975:         KIBAN1 = tendo[ 16 ];   /* H表示 */
976:         seg = 0;               /* 次は下位表示 */
977:
978:     }
979:
980:
981:     /*-----*/
982:     /* reset interrupt request */
983:     /*-----*/
984:     ITU2.TSR.BIT.IMFA = 0;
985:
986: }
987:
988:
989:
990: /*-----*/
991: /* void int_imia4( void ) */
992: /*-----*/
993: /* 機能：7セグメントLED表示 */
994: /* 事前公開課題1 , , */
995: /*-----*/
996: /* この関数はITU4のインターバル割込にて10ms毎に自動的に励起される。 */
997: /*-----*/
998: void int_imia4_124( void ) /* 10msインターバル割り込み */
999: {
1000:
1001:     int i, j, k;
1002:
1003:
1004:     /* (注意) 3桁以上の数字ならば下2桁を有効にする */
1005:     /* 7セグメントLEDが2桁しかないため */
1006:     if ( shadan_kaisu >= 100 ) shadan_kaisu = 0; /* この方法ではちょっとまずいかも? */
1007:                                           /* とりあえずはこれでOK */
1008:
1009:
1010:     j = shadan_kaisu / 10;    /* 10の位を求める */
1011:     k = shadan_kaisu - j * 10; /* 1の位を求める */
1012:
1013:
1014:     /*-----*/
1015:     /* 7segLED ダイナミック点灯 (上位と下位を交互に表示) */
1016:     /*-----*/
1017:
1018:     if ( seg == 0 ) {
1019:
1020:         /*-----*/
1021:         /* 7segLED1(下位)(1の位)表示 */
1022:         /*-----*/
1023:         SEG_SELECT = LO_SEG;    /* 表示は下位(1の位) */
1024:         KIBAN1 = tendo[ k ];
1025:         seg = 1;               /* 次は上位桁表示  */
1026:
1027:     } else {
1028:
1029:         /*-----*/
1030:         /* 7segLED1(上位)(10の位)表示 */
1031:         /*-----*/
1032:         //if( j != 0 ) { /* 0も表示する? */
1033:         //    SEG_SELECT = HI_SEG; /* 表示は上位(10の位) */
1034:         //    KIBAN1 = tendo[ i ];
1035:         //} /* end of if */
1036:
1037:
1038:         /* 0も表示する */
1039:         SEG_SELECT = HI_SEG;    /* 表示は上位(10の位) */
1040:         KIBAN1 = tendo[ i ];
1041:         seg = 0;               /* 次は下位表示 */
1042:
1043:     }
1044:
1045:
1046:     /*-----*/

```

```

1047:      /* reset interrupt request */
1048:      /*-----*/
1049:      ITU4.TSR.BIT.IMFA = 0;
1050:
1051: }
1052:
1053:
1054:
1055: /*-----*/
1056: /* void int_imia4( void ) */
1057: /*-----*/
1058: /* 機能：7セグメントLED表示 */
1059: /* 事前公開課題1 , , , */
1060: /*-----*/
1061: /* この関数はITU4のインターバル割込にて10ms毎に自動的に励起される。 */
1062: /*-----*/
1063: void int_imia4_1234( void ) /* 10msインターバル割り込み */
1064: {
1065:
1066:     int i, j, k;
1067:
1068:     if ( !SW ) { /* トグルスイッチがLである */
1069:
1070:         /* (注意) 3桁以上の数字ならば下2桁を有効にする */
1071:         if ( shadan_kaisu >= 100 ) shadan_kaisu = 0; /* この方法ではちょっとまずいかも? */
1072:         /* とりあえずはこれでOK */
1073:
1074:
1075:         j = shadan_kaisu / 10; /* 10の位を求める */
1076:         k = shadan_kaisu - j * 10; /* 1の位を求める */
1077:
1078:
1079:     } else { /* トグルスイッチをHにするとスタートに戻す。 */
1080:
1081:
1082:         shadan_kaisu = 0; /* 遮断回数ゼロクリア */
1083:         j = k = 0; /* 各桁をゼロクリア */
1084:
1085:
1086:     } /* end of if */
1087:
1088:
1089: /*-----*/
1090: /* 7segLED ダイナミック点灯 (上位と下位を交互に表示) */
1091: /*-----*/
1092:
1093:     if ( seg == 0 ) {
1094:
1095:         /*-----*/
1096:         /* 7segLED1(下位)(1の位)表示 */
1097:         /*-----*/
1098:         SEG_SELECT = LO_SEG; /* 表示は下位(1の位) */
1099:         KIBAN1 = tendo[ k ];
1100:         seg = 1; /* 次は上位表示 */
1101:
1102:     } else {
1103:
1104:         /*-----*/
1105:         /* 7segLED1(上位)(10の位)表示 */
1106:         /*-----*/
1107:         //if( j != 0 ) { /* 0も表示する? */
1108:         //    SEG_SELECT = HI_SEG; /* 表示は上位(10の位) */
1109:         //    KIBAN1 = tendo[ i ];
1110:         //} /* end of if */
1111:
1112:
1113:         /* 0も表示する */
1114:         SEG_SELECT = HI_SEG; /* 表示は上位(10の位) */
1115:         KIBAN1 = tendo[ i ];
1116:         seg = 0; /* 次は下位表示 */
1117:
1118:     }
1119:
1120:
1121: /*-----*/
1122: /* reset interrupt request */
1123: /*-----*/
1124:     ITU4.TSR.BIT.IMFA = 0;
1125:
1126: }
1127:

```

```

1128:
1129:
1130:/*-----*/
1131:/* void int_imia4( void ) */
1132:/*-----*/
1133:/* 機能：7セグメントLED表示 */
1134:/* 事前公開課題1 , , , , */
1135:/*-----*/
1136:/* この関数はITU4のインターバル割込にて10ms毎に自動的に励起される。*/
1137:/*-----*/
1138:void int_imia4( void ) /* 10msインターバル割り込み */
1139:{
1140:
1141:    int i, j, k;
1142:
1143:    if ( !SW ) { /* トグルスイッチがLである */
1144:
1145:        /* (注意) 3桁以上の数字ならば下2桁を有効にする */
1146:        if ( shadan_kaisu >= 100 ) shadan_kaisu = 0; /* この方法ではちょっとまずいかも? */
1147:                                                /* とりあえずはこれでOK */
1148:
1149:
1150:        j = shadan_kaisu / 10; /* 10の位を求める */
1151:        k = shadan_kaisu - j * 10; /* 1の位を求める */
1152:
1153:
1154:
1155:    } else { /* トグルスイッチをHにするとスタートに戻す。 */
1156:
1157:
1158:        shadan_kaisu = 0; /* 遮断回数ゼロクリア */
1159:        j = k = 0; /* 各桁をゼロクリア */
1160:
1161:
1162:    } /* end of if */
1163:
1164:
1165:
1166:/*-----*/
1167:/* 7segLED ダイナミック点灯 (上位と下位を交互に表示) */
1168:/*-----*/
1169:
1170:    if ( seg == 0 ) {
1171:
1172:        /*-----*/
1173:        /* 7segLED1(下位)(1の位)表示 */
1174:        /*-----*/
1175:        SEG_SELECT = LO_SEG; /* 表示は下位(1の位) */
1176:        KIBAN1 = tendo[ k ];
1177:        seg = 1; /* 次は上位表示 */
1178:
1179:    } else {
1180:
1181:        /*-----*/
1182:        /* 7segLED1(上位)(10の位)表示 */
1183:        /*-----*/
1184:        //if( j != 0 ) { /* 0も表示する? */
1185:        //    SEG_SELECT = HI_SEG; /* 表示は上位(10の位) */
1186:        //    KIBAN1 = tendo[ i ];
1187:        //    wait(1);
1188:        //} /* end of if */
1189:
1190:        /* 0も表示する */
1191:        SEG_SELECT = HI_SEG; /* 表示は上位(10の位) */
1192:        KIBAN1 = tendo[ i ];
1193:        seg = 0; /* 次は下位表示 */
1194:
1195:    }
1196:
1197:    /*-----*/
1198:    /* reset interrupt request */
1199:    /*-----*/
1200:    ITU4.TSR.BIT.IMFA = 0;
1201:
1202:}
1203:
1204:
1205:
1206:/*-----*/
1207:/* モジュール名 opening */
1208:/* 処理概要 オープニング */

```

```

1209:/* 引数      なし      */
1210:/* 戻り値     なし      */
1211:/*-----*/
1212:void opening( void )
1213:{
1214:    unsigned int i;
1215:
1216:    SEG_SELECT = HI_LO_SEG; /* 表示は上位下位両方同じ 0x03:0000 0011 */
1217:    KIBAN1 = 0xff; /* 全消灯 */
1218:
1219:
1220:    SEG_SELECT = HI_LO_SEG; /* 表示は上位(10の位)と下位(1の位)の両方 */
1221:    for( i = 0; i < 3; i++ ){
1222:
1223:        KIBAN1 = tendo[ 16 ]; /* H表示 */
1224:        wait( 30 );
1225:
1226:        KIBAN1 = tendo[ 17 ]; /* L表示 */
1227:        wait( 30 );
1228:
1229:    }
1230:
1231:}
1232:
1233:
1234:
1235:/*-----*/
1236:/* モジュール名  seq_cls      */
1237:/* 処理概要      7セグメントLED消去      */
1238:/* 引数          なし      */
1239:/* 戻り値       なし      */
1240:/*-----*/
1241:void seq_cls( void )
1242:{
1243:    unsigned int i;
1244:
1245:    SEG_SELECT = HI_LO_SEG; /* 表示は上位下位両方同じ 0x03:0000 0011 */
1246:    KIBAN1 = 0xff; /* 全消灯 */
1247:
1248:}
1249:
1250:
1251:
1252:/*-----*/
1253:/* モジュール名  brink      */
1254:/* 処理概要      7セグメントLED点滅処理      */
1255:/* 引数          点滅回数      */
1256:/* 戻り値       なし      */
1257:/*-----*/
1258:void brink( int brink_kaisu )
1259:{
1260:    unsigned int i;
1261:
1262:
1263:    for ( i = 0; i < brink_kaisu; i++ ) { /* 点滅 */
1264:
1265:        /*-----*/
1266:        /* 消灯 */
1267:        /* 点滅は消灯を先にした */
1268:        /*-----*/
1269:        ITU.TSTR.BIT.STR4 = 0; /* itu4停止(割り込み処理 停止 ) */
1270:        seq_cls(); /* 7セグメントLED 消去 */
1271:        wait( 50 );
1272:
1273:
1274:        /*-----*/
1275:        /* 点灯 */
1276:        /*-----*/
1277:        ITU.TSTR.BIT.STR4 = 1; /* まだスタートしない */
1278:        wait( 50 );
1279:
1280:    } /* end of for */
1281:
1282:
1283:    ITU.TSTR.BIT.STR4 = 1; /* itu4再スタート */
1284:
1285:}
1286:
1287:
1288:
1289:/*-----*/

```

```

1290:/* モジュール名 RightRotate */
1291:/* 処理概要 Data を shadan_kaisuビット 右ローテートする(8ビット) */
1292:/* 引数 被操作データ(下位4bitのみ有効) unsigned char Data */
1293:/* ローテート回数(0..3) unsigned char count */
1294:/* 戻り値 変換値 unsigned char */
1295:/*-----*/
1296:unsigned char rrotate(unsigned char Data, unsigned char count )
1297:{
1298:    return( RotateData[ Data & 0xf ][ ( 4 - count ) & 3 ] );
1299:
1300:}
1301:
1302:
1303:
1304:/*-----*/
1305:/* モジュール名 LeftRotate */
1306:/* 処理概要 Data を shadan_kaisuビット 左ローテートする(8ビット) */
1307:/* 引数 被操作データ(下位4bitのみ有効) unsigned char Data */
1308:/* ローテート回数(0..3) unsigned char count */
1309:/* 戻り値 変換値 unsigned char */
1310:/*-----*/
1311:unsigned char lrotate( unsigned char Data, unsigned char count )
1312:{
1313:    return( RotateData[ Data & 0xf ][ count & 3 ] );
1314:
1315:}
1316:
1317:
1318:/*-----*/
1319:/* モジュール名 kadai1 */
1320:/* 処理概要 平成17年度全工長協会HP掲載 事前公表課題1 */
1321:/* このルーチンではフォトインタラプタの遮断回数のみカウントする。 */
1322:/* 7セグメントLEDの表示は全て割り込み処理(int_imia4)で行う。 */
1323:/* 引数 なし */
1324:/* 戻り値 なし */
1325:/*-----*/
1326:void kadai1( void )
1327:{
1328:
1329:    /*-----*/
1330:    /* スタート */
1331:    /*-----*/
1332:
1333:    shadan_kaisu = 0; /* フォトインタラプタ遮断回数初期化(ゼロクリア) */
1334:
1335:
1336:    /*-----*/
1337:    /* 割り込み処理開始 */
1338:    /*-----*/
1339:    ITU.TSTR.BIT.STR4 = 1;
1340:
1341:
1342:    while( 1 ) {
1343:
1344:        /*-----*/
1345:        /* フォトインタラプタが反応するまで */
1346:        /*-----*/
1347:        while( PHOTO ) { } /* フォトインタラプタの反応がない(透過中) 待ち状態 */
1348:        wait( 2 ); /* チャタリング防止 */
1349:        while( PHOTO ) { } /* 再確認 */
1350:
1351:        /*-----*/
1352:        /* フォトインタラプタが反応しているうち */
1353:        /*-----*/
1354:        while( !PHOTO ) { } /* フォトインタラプタの反応のある(遮断中) 待ち状態 */
1355:        wait( 2 ); /* チャタリング防止 */
1356:        while( !PHOTO ) { } /* 再確認 */
1357:
1358:
1359:        /*-----*/
1360:        /* 遮断回数カウント */
1361:        /*-----*/
1362:        shadan_kaisu++;
1363:
1364:        if( ( shadan_kaisu > 0 ) && ( ( shadan_kaisu % 10 ) == 0 ) ) {
1365:
1366:            brink( 5 ); /* 5回点滅 */
1367:
1368:        } /* end of if */
1369:
1370:}

```

```

1371:
1372:         }/* end of while(1) */
1373:
1374: }
1375:
1376:
1377:
1378: /*-----*/
1379: /* モジュール名   kadai2 */
1380: /* 処理概要     平成17年度全工長協会HP掲載  事前公表課題2 */
1381: /* 引数         なし */
1382: /* 戻り値       なし */
1383: /*-----*/
1384: void kadai2( void )
1385: {
1386:
1387:     unsigned char i, j, k;
1388:     unsigned int  kaiten_su;
1389:     unsigned char smotor, smotor2; /* motor励磁信号 */
1390:
1391:
1392:     /*-----*/
1393:     /* スタート */
1394:     /*-----*/
1395:
1396:     seg_cls(); /* 7セグメントLEDは使用しないので消しておく */
1397:
1398:
1399:     kaiten_su = 0; /* ステッピングモータ回転数期化(ゼロクリア) */
1400:
1401:
1402:     KIBAN2 = 0; /* ステッピングモータoff LED全消灯 */
1403:
1404:
1405:     while(1) {
1406:
1407:
1408:         while( !SW ) { /* SWがL */
1409:
1410:
1411:             /*-----*/
1412:             /* フォトインタラプタが反応しているうち */
1413:             /* ステッピングモータ時計方向一定回転, LED連動 */
1414:             /*-----*/
1415:             //smotor = 0x08; /* モータのみ */
1416:             smotor = 0x88; /* 1相励磁 LED連動 */ /* */
1417:             //smotor = 0xcc; /* 2相励磁 LED連動 */
1418:
1419:             while( !PHOTO ) { /* フォトインタラプタの反応のある間(遮断中)実行 */
1420:
1421:                 KIBAN2 = smotor;
1422:                 wait( 10 );
1423:                 smotor = lrotate( smotor, 1 );
1424:
1425:             }/* end of while */
1426:
1427:         }/* end of while */
1428:
1429:
1430:         /*-----*/
1431:         /* SWがH(on)になると */
1432:         /*-----*/
1433:         if ( SW ) {
1434:
1435:             /*-----*/
1436:             /* 励磁信号は継続 */
1437:             /*-----*/
1438:             /* -1) 反時計方向に5回転 20pulse x 5 = 100pulse */
1439:             /*-----*/
1440:             for( i = 0; i < 5; i++ ) {
1441:
1442:                 kaiten_su = ( (i + 1) << 4 ) & 0xf0;
1443:
1444:                 for( j = 0; j < 20; j++ ) {
1445:
1446:                     smotor2 = kaiten_su | ( smotor & 0x0f );
1447:                     KIBAN2 = smotor2;
1448:                     wait( 10 );
1449:                     smotor = rrotate( smotor, 1 );
1450:
1451:                 }/* end of for(j) */

```

```

1452:
1453:         }/* end of for(i) */
1454:
1455:
1456:
1457:         /*-----*/
1458:         /* -1) LEDを5回点滅 */
1459:         /*-----*/
1460:
1461:         for ( i = 0; i < 5; i++ ) {
1462:
1463:             KIBAN2 = 0xf0 | smotor; /* 下位4bit(励磁信号)は残す */
1464:             wait( 50 ); /* 少し待つ */
1465:
1466:
1467:             KIBAN2 = 0x0f & smotor; /* 下位4bit(励磁信号)は残す */
1468:             wait( 50 ); /* 少し待つ */
1469:
1470:         }/* end of for */
1471:
1472:
1473:
1474:         /*-----*/
1475:         /* -1) 反時計方向に 10回転 20pulse x 5 = 100pulse */
1476:         /*-----*/
1477:
1478:         for( i = 0; i < 10; i++ ) {
1479:
1480:             kaiten_su = ( ( i + 1 ) << 4 ) & 0xf0;
1481:
1482:             for( j = 0; j < 20; j++ ) {
1483:
1484:                 smotor2 = kaiten_su | ( smotor & 0x0f );
1485:                 KIBAN2 = smotor2;
1486:                 wait( 10 );
1487:                 smotor = lrotate( smotor, 1 );
1488:
1489:             }/* end of for(j) */
1490:
1491:         }/* end of for(i) */
1492:
1493:
1494:
1495:         /*-----*/
1496:         /* -2) LEDを10回点滅 */
1497:         /*-----*/
1498:
1499:         for ( i = 0; i < 10; i++ ) {
1500:
1501:             KIBAN2 = 0xf0 | smotor; /* 下位4bit(励磁信号)は残す */
1502:             wait( 50 ); /* 少し待つ */
1503:
1504:
1505:             KIBAN2 = 0x0f & smotor; /* 下位4bit(励磁信号)は残す */
1506:             wait( 50 ); /* 少し待つ */
1507:
1508:         }/* end of for */
1509:
1510:
1511:
1512:     }/* end of if */
1513:
1514:
1515: }/* end of while(1) */
1516:
1517:
1518: }
1519:
1520:
1521:
1522: /*-----*/
1523: /* モジュール名 kadai3_1 */
1524: /* 処理概要 事前公表課題3 */
1525: /* 05年山形県大会は事前配布基板3の回路からDCモータ回路を除いて */
1526: /* 行われた。したがって公開課題3はDCモータをステッピングモータ */
1527: /* に置き換えてプログラム作成に取り組んだ。 */
1528: /* 引数 なし */
1529: /* 戻り値 なし */
1530: /*-----*/
1531: void kadai3_1( void )
1532: {

```

```

1533:
1534:     unsigned char i, j, k;
1535:     unsigned int  kaiten_su;
1536:     unsigned char smotor, smotor2; /* motor励磁信号 */
1537:
1538:
1539:     /*-----*/
1540:     /* スタート */
1541:     /*-----*/
1542:
1543:     KIBAN2 = 0; /* ステッピングモータoff LED全消灯 */
1544:
1545:
1546:     //smotor = 0x08; /* モータのみ */
1547:     smotor = 0x88; /* 1相励磁 LED連動 */
1548:     //smotor = 0xcc; /* 2相励磁 LED連動 */
1549:
1550:
1551:     while(1) {
1552:
1553:         if ( SW ) { /* SWの出力を"H"にすると */
1554:
1555:             /*----*/
1556:             /*      */
1557:             /*----*/
1558:             SEG_SELECT = HI_SEG;
1559:             KIBAN1 = tendo[ 16 ]; /* H表示 */
1560:
1561:
1562:             KIBAN2 = smotor;
1563:             wait( 5 ); /* 高速回転 */
1564:             smotor = lrotate( smotor, 1 );
1565:
1566:
1567:         } else { /* SWの出力を"L"にすると */
1568:
1569:             /*----*/
1570:             /*      */
1571:             /*----*/
1572:             SEG_SELECT = HI_SEG;
1573:             KIBAN1 = tendo[ 17 ]; /* L表示 */
1574:
1575:
1576:         } /* end of if */
1577:
1578:     } /* end of while(1) */
1579: }
1580:
1581:
1582:
1583:
1584: /*-----*/
1585: /* モジュール名   kadai3_12 */
1586: /* 処理概要       事前公表課題3 */
1587: /*               05年山形県大会は事前配布基板3の回路からDCモータ回路を除いて */
1588: /*               行われた。したがって公開課題3はDCモータをステッピングモータ */
1589: /*               に置き換えてプログラム作成に取り組んだ。 */
1590: /* 引数           なし */
1591: /* 戻り値         なし */
1592: /*-----*/
1593: void kadai3_12( oid )
1594: {
1595:
1596:     unsigned char i, j, k;
1597:     unsigned int  kaiten_su;
1598:     unsigned char smotor, smotor2; /* motor励磁信号 */
1599:
1600:
1601:     /*-----*/
1602:     /* スタート */
1603:     /*-----*/
1604:
1605:     KIBAN2 = 0; /* ステッピングモータoff LED全消灯 */
1606:
1607:
1608:     //smotor = 0x08; /* モータのみ */
1609:     smotor = 0x88; /* 1相励磁 LED連動 */
1610:     //smotor = 0xcc; /* 2相励磁 LED連動 */
1611:
1612:
1613:     while(1) {

```

```

1614:
1615:
1616:         if ( SW ) { /* SWの出力を"H"にすると */
1617:
1618:             /*----*/
1619:             /*      */
1620:             /*----*/
1621:             SEG_SELECT = HI_SEG;
1622:             KIBAN1 = tendo[ 16 ]; /* H表示 */
1623:
1624:
1625:             if ( !PHOTO ) { /* フォトインタラプタを遮断すると */
1626:
1627:
1628:                 KIBAN2 = smotor;
1629:                 wait( 10 ); /* 低速回転 */
1630:                 smotor = lrotate( smotor, 1 );
1631:
1632:
1633:             } else { /* フォトインタラプタ透過 */
1634:
1635:                 KIBAN2 = smotor;
1636:                 wait( 5 ); /* 高速回転 */
1637:                 smotor = lrotate( smotor, 1 );
1638:
1639:
1640:             } /* end of if */
1641:
1642:
1643:
1644:         } else { /* SWの出力を"L"にすると */
1645:
1646:
1647:             /*----*/
1648:             /*      */
1649:             /*----*/
1650:             SEG_SELECT = HI_SEG;
1651:             KIBAN1 = tendo[ 17 ]; /* L表示 */
1652:
1653:
1654:         } /* end of if */
1655:
1656:     } /* end of while(1) */
1657:
1658: }
1659:
1660:
1661:
1662: /*-----*/
1663: /* モジュール名 kadai3_1234 */
1664: /* 処理概要 事前公表課題3 */
1665: /* 05年山形県大会は事前配布基板3の回路からDCモータ回路を除いて */
1666: /* 行われた。したがって公開課題3はDCモータをステッピングモータ */
1667: /* に置き換えてプログラム作成に取り組んだ。 */
1668: /* 引数 なし */
1669: /* 戻り値 なし */
1670: /*-----*/
1671: void kadai3_1234( void )
1672: {
1673:
1674:     unsigned char i, j, k;
1675:     unsigned int kaiten_su;
1676:     unsigned char smotor, smotor2; /* motor励磁信号 */
1677:
1678:
1679:     /*-----*/
1680:     /* スタート */
1681:     /*-----*/
1682:
1683:     seg_cls(); /* 7セグメントLED消去 */
1684:
1685:     KIBAN2 = 0; /* ステッピングモータoff LED全消灯 */
1686:
1687:     shadan_kaisu = 0;
1688:
1689:     //smotor = 0x08; /* モータのみ */
1690:     smotor = 0x88; /* 1相励磁 LED連動(動作確認の為) */
1691:     //smotor = 0xcc; /* 2相励磁 LED連動(動作確認の為) */
1692:
1693:
1694:     while(1) {

```

```

1695:
1696:
1697:     if ( SW ) { /* SWの出力を"H"にすると */
1698:
1699:         /*-----*/
1700:         /* 割り込み処理開始 */
1701:         /*-----*/
1702:         ITU.TSTR.BIT.STR2 = 1;
1703:
1704:
1705:         /*----*/
1706:         /* */
1707:         /*----*/
1708:         // SEG_SELECT = HI_SEG;
1709:         // KIBAN1 = tendo[ 16 ]; /* H表示 */
1710:
1711:
1712:         /*-----*/
1713:         /* フォトインタラプタ透過状態 (遮断待ち) */
1714:         /*-----*/
1715:         while( PHOTO ) { /* フォトインタラプタの反応がない(透過中) 待ち状態 */
1716:
1717:             KIBAN2 = smotor;
1718:             wait( 5 ); /* 高速回転 */
1719:             smotor = lrotate( smotor, 1 );
1720:         }
1721:
1722:         wait( 2 ); /* チャタリング防止 */
1723:
1724:         while( PHOTO ) { /* 再確認 */
1725:
1726:             KIBAN2 = smotor;
1727:             wait( 5 ); /* 高速回転 */
1728:             smotor = lrotate( smotor, 1 );
1729:         }
1730:
1731:
1732:         /*-----*/
1733:         /* フォトインタラプタが遮断中 (透過待ち) */
1734:         /*-----*/
1735:         while( !PHOTO ) { /* フォトインタラプタの反応のある(遮断中) 待ち状態 */
1736:
1737:             KIBAN2 = smotor;
1738:             wait( 10 ); /* 低速回転 */
1739:             smotor = lrotate( smotor, 1 );
1740:         }
1741:
1742:         wait( 2 ); /* チャタリング防止 */
1743:
1744:         while( !PHOTO ) { /* 再確認 */
1745:             KIBAN2 = smotor;
1746:             wait( 10 ); /* 低速回転 */
1747:             smotor = lrotate( smotor, 1 );
1748:         }
1749:
1750:
1751:         /*-----*/
1752:         /* 遮断回数カウンタ */
1753:         /*-----*/
1754:         shadan_kaisu++;
1755:
1756:
1757:     } else { /* SWの出力を"L"にすると */
1758:
1759:
1760:         /*-----*/
1761:         /* 割り込み処理停止 */
1762:         /*-----*/
1763:         ITU.TSTR.BIT.STR2 = 0;
1764:
1765:
1766:         /*----*/
1767:         /* */
1768:         /*----*/
1769:         SEG_SELECT = HI_SEG;
1770:         KIBAN1 = tendo[ 17 ]; /* L表示 */
1771:
1772:
1773:     } /* end of if */
1774:
1775:

```

```

1776:         }/* end of while(1) */
1777:
1778: }
1779:
1780:
1781:
1782: /*-----*/
1783: /* モジュール名   kadai3_1234 */
1784: /* 処理概要       事前公表課題3 */
1785: /*               05年山形県大会は事前配布基板3の回路からDCモータ回路を除いて */
1786: /*               行われた。したがって公開課題3はDCモータをステッピングモータ */
1787: /*               に置き換えてプログラム作成に取り組んだ。 */
1788: /*               */
1789: /*               時スイッチの出力"L"の場合スタートに戻す。 */
1790: /*               */
1791: /* 引数           なし */
1792: /* 戻り値         なし */
1793: /*-----*/
1794: void kadai3( void )
1795: {
1796:
1797:     unsigned char i, j, k;
1798:     unsigned int  kaiten_su;
1799:     unsigned char smotor, smotor2; /* motor励磁信号 */
1800:
1801:
1802:     /*-----*/
1803:     /* スタート */
1804:     /*-----*/
1805:
1806:     seg_cls(); /* 7セグメントLED消去 */
1807:
1808:     KIBAN2 = 0; /* ステッピングモータoff LED全消灯 */
1809:
1810:     shadan_kaisu = 0;
1811:
1812:     //smotor = 0x08; /* モータのみ */
1813:     smotor = 0x88; /* 1相励磁 LED連動(動作確認の為) */
1814:     //smotor = 0xcc; /* 2相励磁 LED連動(動作確認の為) */
1815:
1816:
1817:     while(1) {
1818:
1819:
1820:         if ( SW ) { /* SWの出力を"H"にすると */
1821:
1822:             /*-----*/
1823:             /* 割り込み処理開始 */
1824:             /*-----*/
1825:             ITU.TSTR.BIT.STR2 = 1;
1826:
1827:
1828:             /*----*/
1829:             /* */
1830:             /*----*/
1831:             // SEG_SELECT = HI_SEG;
1832:             // KIBAN1 = tendo[ 16 ]; /* H表示 */
1833:
1834:
1835:             /*-----*/
1836:             /* フォトインタラプタ透過状態 (遮断待ち) */
1837:             /*-----*/
1838:             while( PHOTO && SW ) { /* フォトインタラプタの反応がない(透過中) 待ち状態 */
1839:
1840:                 KIBAN2 = smotor;
1841:                 wait( 5 ); /* 高速回転 */
1842:                 smotor = lrotate( smotor, 1 );
1843:             }
1844:
1845:             wait( 2 ); /* チャタリング防止 */
1846:
1847:             while( PHOTO && SW ) { /* 再確認 */
1848:
1849:                 KIBAN2 = smotor;
1850:                 wait( 5 ); /* 高速回転 */
1851:                 smotor = lrotate( smotor, 1 );
1852:             }
1853:
1854:
1855:             /*-----*/
1856:             /* フォトインタラプタが遮断中 (透過待ち) */

```

```

1857:          /*-----*/
1858:          while( !PHOTO && SW ) { /* フォトインタラプタの反応のある(遮断中) 待ち状態 */
1859:
1860:              KIBAN2 = smotor;
1861:              wait( 10 ); /* 低速回転 */
1862:              smotor = lrotate( smotor, 1 );
1863:          }
1864:
1865:          wait( 2 ); /* チャタリング防止 */
1866:
1867:          while( !PHOTO && SW ) { /* 再確認 */
1868:              KIBAN2 = smotor;
1869:              wait( 10 ); /* 低速回転 */
1870:              smotor = lrotate( smotor, 1 );
1871:          }
1872:
1873:
1874:          /*-----*/
1875:          /* 遮断回数カウンタ */
1876:          /*-----*/
1877:          shadan_kaisu++;
1878:
1879:
1880:      } else { /* SWの出力を"L"にすると */
1881:
1882:
1883:          /*-----*/
1884:          /* 割り込み処理停止 */
1885:          /*-----*/
1886:          ITU.TSTR.BIT.STR2 = 0;
1887:
1888:
1889:          /*----*/
1890:          /* */
1891:          /*----*/
1892:          SEG_SELECT = HI_SEG;
1893:          KIBAN1 = tendo[ 17 ]; /* L表示 */
1894:
1895:      } /* end of if */
1896:
1897:
1898:
1899:   } /* end of while(1) */
1900:
1901: }
1902:
1903:
1904:
1905: /*-----*/
1906: /* モジュール名 kadai061 */
1907: /* 処理概要 平成18年度全工長協会HP掲載 事前公表課題 ストップウォッチ */
1908: /* トグルスイッチの出力が"H"の場合、フォトインタラプタの遮断時間 */
1909: /* を7セグメントLEDに表示する。表示は0.0秒~9.9秒まで */
1910: /* トグルスイッチの出力が"L"の場合、遮断時間をゼロに戻す。 */
1911: /* 引数 なし */
1912: /* 戻り値 なし */
1913: /*-----*/
1914: void kadai061( void )
1915: {
1916:
1917:     KIBAN1 = 0xff; /* 7segLED cls */
1918:
1919:     ITU.TSTR.BIT.STRO = 1;
1920:     ITU.TSTR.BIT.STR1 = 1;
1921:
1922:
1923:     /* スタート */
1924:     while(1) {
1925:
1926:         /* すべて割り込み処理 */
1927:
1928:     } /* end of while(1) */
1929:
1930: }
1931:
1932:
1933:
1934: /*-----*/
1935: /* モジュール名 main */
1936: /* 処理概要 メイン処理 */
1937: /* 引数 なし */

```

```

1938: /* 戻り値      なし                                     */
1939: /*-----*/
1940: int main( void )
1941: {
1942:
1943:
1944:     /*-----*/
1945:     /* H8のポート初期化 */
1946:     /*-----*/
1947:     Init_Port();
1948:
1949:
1950:     /*-----*/
1951:     /* H8のタイマー初期化 */
1952:     /*-----*/
1953:     Init_H8();
1954:
1955:     speed ( 0 );
1956:
1957:     /*-----*/
1958:     /* オープニング */
1959:     /*-----*/
1960:     opening(); /* プログラムスタート知らせ */
1961:
1962:
1963:
1964:     /*-----*/
1965:     /* 準備完了 処理スタート */
1966:     /*-----*/
1967:     switch ( DIP_SW ) { /* on bord dip_switchにより処理変更 */
1968:
1969:         case 0: /* 大会本番 */
1970:
1971:             kadai061(); /* 事前公開課題06-1 */
1972:
1973:             break;
1974:
1975:
1976:         case 1: /* 事前公開課題 & 配布基板1テスト*/
1977:
1978:             kadai1(); /* 事前公開課題1 */
1979:
1980:             //k1_test1(); /* 7seqLEDテスト */
1981:             //k1_test2(); /* 7seqLEDテスト2 */
1982:             //k1_test3(); /* 7seqLEDテスト3 */
1983:             //k1_test4(); /* 7seqLED & DCモータテスト */
1984:             break;
1985:
1986:
1987:         case 2: /* 事前公開課題2 & 配布基板2テスト */
1988:
1989:             kadai2(); /* 事前公開課題2 */
1990:
1991:             //k2_test1();
1992:             //k2_test2();
1993:             //k2_test3();
1994:             //k2_test4();
1995:             break;
1996:
1997:
1998:         case 3: /* 当日製作基板 テスト*/
1999:
2000:             kadai3(); /* 事前公開課題3 */
2001:
2002:             //k3_test1();
2003:             break;
2004:
2005:
2006:         default:
2007:             break;
2008:
2009:
2010:     } /* end of switch case */
2011:
2012: } /* end of main */

```