```
/*------------------------------------------------------------------------*/
/* (第21回全国高等学校ロボット競技会「金シャッチー競技」)                  */
/* 第22回メカトロアイディアコンテスト山形県大会                           */
/* 自立型マシン用プログラム                          Ver.1.0 H25.06.22～   */
/*                                                      山形電波工高csc   */
/*------------------------------------------------------------------------*/
/* ハード仕様                                                             */
/*  ＣＰＵ          マイコンカーラリー用 H8/3048f-one 24.576MHz            */
/*                        リセットスイッチ×1(on board)                    */
/*                        ディップスイッチ×1(on board)                    */
/*  マザーボード    自作   スタートスイッチ×1                              */
/*                        作戦スイッチ ×1                                 */
/*                        確認用ＬＥＤ回路                                */
/*  センサ          ライントレース用(マイコンカーラリーVer4.1)             */
/*                  前壁検出距離センサ(PSD SHARP社製 GP2D12×1)             */
/*                  リモコンロボット乗り込み検出センサ(対物センサ×1)       */
/*  モータドライブ  マイコンカーラリーVer3.0                               */
/*------------------------------------------------------------------------*/
/* ポート仕様概要                                                         */
/*      ポート9 P90       スタートスイッチ                                */
/*      ポート6 P60～P63  on board ディップスイッチ                       */
/*      ポート3 P30～P33  増設ディップスイッチ                            */
/*      ポート4 P40～P47  ＬＥＤ                                          */
/*      ポート1 P10～P17  ラインセンサ                                    */
/*      ポート7 P70～P77  PSD,対物センサ                                  */
/*      ポートA PA0～PA7  使用せず                                        */
/*      ポートB PB0～PB7  モータドライブボード                            */
/*------------------------------------------------------------------------*/
/*      P9 0  スタートスイッチ  ;;                                        */
/*         1          未        ;;                                        */
/*         2          〃        ;;                                        */
/*         3          〃        ;;                                        */
/*         4          〃        ;;                                        */
/*         5          〃        ;;                                        */
/*         6          〃        ;;                                        */
/*         7          〃        ;;                                        */
/*------------------------------------------------------------------------*/
/*      P6 0  on board ディップスイッチ;;上位用                           */
/*         1          〃        ;;                                        */
/*         2          〃        ;;                                        */
/*         3          〃        ;;                                        */
/*         4       未使用       ;;                                        */
/*         5          〃        ;;                                        */
/*         6          〃        ;;                                        */
/*         7          〃        ;;                                        */
/*------------------------------------------------------------------------*/
/*      P3 0  増設ディップスイッチ;;下位用                                */
/*         1          〃        ;;                                        */
/*         2          〃        ;;                                        */
/*         3          〃        ;;                                        */
/*         4       未使用       ;;                                        */
/*         5          〃        ;;                                        */
/*         6          〃        ;;                                        */
/*         7          〃        ;;                                        */
/*------------------------------------------------------------------------*/
/*      P4 0    ＬＥＤ(1)      ;;右                                       */
/*         1    ＬＥＤ(2)      ;;                                         */
/*         2    ＬＥＤ(3)      ;;                                         */
/*         3    ＬＥＤ(4)      ;;                                         */
/*         4    ＬＥＤ(5)      ;;                                         */
/*         5    ＬＥＤ(6)      ;;                                         */
/*         6    ＬＥＤ(7)      ;;                                         */
/*         7    ＬＥＤ(8)      ;;左                                       */
/*------------------------------------------------------------------------*/
/*      P1 0    床センサ(1)    ;;右端                                     */
/*         1      〃    (2)    ;;                                         */
/*         2      〃    (3)    ;;                                         */
/*         3      〃    (4)    ;;                                         */
/*         4      〃    (5)    ;;                                         */
/*         5      〃    (6)    ;;                                         */
/*         6      〃    (7)    ;;                                         */
/*         7      〃    (8)    ;;左端                                     */
/*------------------------------------------------------------------------*/
/*      P7 0    PSDセンサ      ;;                                         */
/*         1     未使用        ;;                                         */
/*         2       〃          ;;                                         */
/*         3       〃          ;;                                         */
/*         4    乗り込みセンサ ;;                                         */
/*         5     未使用        ;;                                         */
/*         6       〃          ;;                                         */
/*         7       〃          ;;                                         */
/*------------------------------------------------------------------------*/
/*      PA 0     未使用        ;;                                         */
/*         1       〃          ;;                                         */
/*         2       〃          ;;                                         */
```

```
 86 /*         3       〃          ;;                                          */
 87 /*         4       〃          ;;                                          */
 88 /*         5       〃          ;;                                          */
 89 /*         6       〃          ;;                                          */
 90 /*         7       〃          ;;                                          */
 91 /*-------------------------------------------------------------------------*/
 92 /*      PB 0                   ;;                                          */
 93 /*         1                   ;;                                          */
 94 /*         2                   ;;                                          */
 95 /*         3                   ;;                                          */
 96 /*         4                   ;;                                          */
 97 /*         5                   ;;                                          */
 98 /*         6                   ;;                                          */
 99 /*         7                   ;;                                          */
100 /*-------------------------------------------------------------------------*/
101 /*      P5 0     未使用         ;;                                          */
102 /*         1       〃          ;;                                          */
103 /*         2       〃          ;;                                          */
104 /*         3       〃          ;;                                          */
105 /*         4       〃          ;;                                          */
106 /*         5       〃          ;;                                          */
107 /*         6       〃          ;;                                          */
108 /*         7       〃          ;;                                          */
109 /*-------------------------------------------------------------------------*/
110
111 /*====================================*/
112 /* インクルード                       */
113 /*====================================*/
114 #include    <no_float.h> /* stdioの簡略化 最初に置く*/
115 #include    <stdio.h>
116 #include    <machine.h>
117 #include    "h8_3048.h"
118
119 /*====================================*/
120 /* シンボル定義                       */
121 /*====================================*/
122 /* 定数設定 */
123 #define         TIMER_CYCLE     3071    /* タイマのサイクル 1ms     */
124                                         /* Φ/8で使用する場合、      */
125                                         /* Φ/8 = 325.5[ns]          */
126                                         /* ∴TIMER_CYCLE =           */
127                                         /*      1[ms] / 325.5[ns]   */
128                                         /*              = 3072      */
129 #define         PWM_CYCLE       49151   /* PWMのサイクル 16ms       */
130                                         /* ∴PWM_CYCLE =             */
131                                         /*      16[ms] / 325.5[ns]  */
132                                         /*              = 49152     */
133 #define         SERVO_CENTER    5000    /* サーボのセンタ値         */
134 #define         HANDLE_STEP     26      /* 1°分の値                */
135
136 /* マスク値設定 ×：マスクあり(無効)  ○：マスク無し(有効) */
137 #define         MASK2_2         0x66    /* ××○○××○○×         */
138 #define         MASK2_0         0x60    /* ××○○×××××         */
139 #define         MASK0_2         0x06    /* ××××××○○×         */
140 #define         MASK3_3         0xe7    /* ○○○××○○○         */
141 #define         MASK0_3         0x07    /* ××××××○○○         */
142 #define         MASK3_0         0xe0    /* ○○○××××××         */
143 #define         MASK4_0         0xf0    /* ○○○○××××         */
144 #define         MASK0_4         0x0f    /* ××××○○○○         */
145 #define         MASK4_4         0xff    /* ○○○○○○○○         */
146 #define         MASK1_8         0x18    /* ×××○○×××         */
147
148 #define PUSH_SW ((P9DR & 0x01)==1)          /* スタートスイッチ押下でTRUE       */
149 #define LED     P4DR                        /* 動作確認用LED 8個                */
150 #define DIP_HI  ((_BYTE)( ~P6DR & 0x0f )    /* ディップスイッチ1 on bord (4bit) */
151 #define DIP_LO  ((_BYTE)( P3DR & 0x0f )     /* ディップスイッチ2 増設(4bit)     */
152
153 /*====================================*/
154 /* プロトタイプ宣言                   */
155 /*====================================*/
156 void init( void );
157 void timer( unsigned long timer_set );
158
159 int check_crank( void );
160
161 unsigned char sensor_inp( unsigned char mask );
162 unsigned char dipsw_get( void );
163 unsigned char pushsw_get( void );
164
165 unsigned char norikomi_get( void );
166
167 void led_out( unsigned char led );
168 void speed( int accele_l, int accele_r );
169 void handle( int angle );
170
```

```
171 void led_on( void );       /* LED 全点灯 */
172 void led_off( void );      /* LED 全消灯 */
173 void led_on_0f( void );    /* LED 点灯   */
174 void led_on_f0( void );    /* LED 点灯   */
175 void led_on_aa( void );    /* LED 点灯   */
176 void led_on_55( void );    /* LED 点灯   */
177
178 int get_ad0_single( void );
179 int get_ad0_scan( void );
180 int get_ad1_scan( void );
181
182 void ad0_single_test( void );
183 void ad0_scan_test( void );
184 void ad1_scan_test( void );
185
186 void trace( void );
187
188 /*====================================*/
189 /* グローバル変数の宣言               */
190 /*====================================*/
191 unsigned long   cnt0;    /* timer関数用  */
192 unsigned long   cnt1;    /* main内で使用 */
193 int             pattern; /* パターン番号 */
194
195
196 /*--------------------------------------------------------------------*/
197 /* 処理概要      H8/3048F-ONE 内蔵周辺機能 初期化                     */
198 /* 引  数        なし                                                 */
199 /* 戻り値        なし                                                 */
200 /*               1出力設定 0入力設定                                  */
201 /*--------------------------------------------------------------------*/
202 void init( void )
203 {
204       /* I/Oポートの入出力設定 */
205          P1DDR = 0x00;   /* ポート1 P10～P17 ラインセンサ              */
206          P2DDR = 0xff;
207          P3DDR = 0xf0;   /* ポート3 P30～P33 増設ディップスイッチ      */
208          P4DDR = 0xff;   /* ポート4 P40～P47 ＬＥＤ                    */
209          P5DDR = 0xff;
210          P6DDR = 0xf0;   /* ポート6 P60～P63 on board ディップスイッチ */
211          P8DDR = 0xff;
212        //P9DDR = 0xf7;   /* 通信ポート                                 */
213          P9DDR = 0xf6;   /* 通信ポート, ポート9 P90 スタートスイッチ   */
214          PADDR = 0xff;   /* 使用せず */
215          PBDR  = 0xc0;
216          PBDDR = 0xfe;   /* モータドライブ基板Vol.3                    */
217
218     /* ※センサ基板のP7は、入力専用なので入出力設定はありません       */
219                         /*  ポート7 P70～P77  P70 PSD, P74対物センサ  */
220
221            /* A/Dの初期設定 */     /* シングルモード             */
222             //AD_CSR = 0x00;       /* AN0使用                    */
223
224            /* A/Dの初期設定 */
225               AD_CSR = 0x11;       /* スキャンモード使用AN0-AN1 */
226               AD_CSR |= 0x20;      /* ADスタート                 */
227
228       /* ITU0 1msごとの割り込み */
229          ITU0_TCR = 0x23;
230          ITU0_GRA = TIMER_CYCLE;
231          ITU0_IER = 0x01;
232
233       /* ITU3,4 リセット同期PWMモード 左右モータ、サーボ用 */
234          ITU3_TCR = 0x23;
235          ITU_FCR  = 0x3e;
236          ITU3_GRA = PWM_CYCLE;               /* 周期の設定        */
237          ITU3_GRB = ITU3_BRB = 0;            /* 左モータのPWM設定 */
238          ITU4_GRA = ITU4_BRA = 0;            /* 右モータのPWM設定 */
239          ITU4_GRB = ITU4_BRB = SERVO_CENTER; /* サーボのPWM設定   */
240          ITU_TOER = 0x38;
241
242       /* ITUのカウントスタート */
243          ITU_STR = 0x09;
244 }
245
246 /*--------------------------------------------------------------------*/
247 /* 処理概要      ITU0 割り込み処理                                    */
248 /* 引  数        なし                                                 */
249 /* 戻り値        なし                                                 */
250 /*--------------------------------------------------------------------*/
251 #pragma interrupt( interrupt_timer0 )
252 void interrupt_timer0( void )
253 {
254          ITU0_TSR &= 0xfe; /* フラグクリア */
255          cnt0++;
```

```
256         cnt1++;
257 }
258
259 /*---------------------------------------------------------------------*/
260 /* 処理概要      タイマ本体                                            */
261 /* 引  数        タイマ値 1=1ms                                        */
262 /* 戻り値        なし                                                  */
263 /*---------------------------------------------------------------------*/
264 void timer( unsigned long timer_set )
265 {
266         cnt0 = 0;
267         while( cnt0 < timer_set );
268 }
269
270 /*---------------------------------------------------------------------*/
271 /* 処理概要      センサ状態検出                                        */
272 /* 引  数        マスク値                                              */
273 /* 戻り値        センサ値                                              */
274 /*---------------------------------------------------------------------*/
275 unsigned char sensor_inp( unsigned char mask )
276 {
277     unsigned char sensor;
278         sensor  = ~P1DR; /* 校内 */ /* 黒字に白テープ */
279        //sensor  =  P1DR; /* 本番 */ /* 白地に黒テープ */
280
281         sensor  &= 0xef;     /* sensor = sensor & 0xef :    0xef: 1110 1111 */
282                              /* 第4bit 無効                                 */
283
284         if( sensor & 0x08 ) sensor |= 0x10;
285           /* 0x08: 0000 1000 第3bitは中心センサが入力 センサ反応ありは1になる */
286           /*                                          センサ反応ありは0になる */
287           /* もし第3bitが1ならば、sensor |=0x10;を実行する */
288           /* 中心センサが反応していたら第4bitも1にする     */
289         sensor &= mask;
290         return sensor;
291 }
292
293 /*---------------------------------------------------------------------*/
294 /* 処理概要      右クランク検出処理                                    */
295 /* 引  数        なし                                                  */
296 /* 戻り値        0:なし 1:あり                                         */
297 /*---------------------------------------------------------------------*/
298 int check_rcrank( void )
299 {
300     unsigned char b;
301     int ret;
302         ret = 0;
303         b = sensor_inp(MASK4_4);
304         if( b==0x1f ) { /* ○○○  ●  ●●●  : 000 1 111 */
305                         /* ○○○●  ●●●●  : 0001 1111 */
306             ret = 1;
307         }
308         return ret;
309 }
310
311 /*---------------------------------------------------------------------*/
312 /* 処理概要      左クランク検出処理                                    */
313 /* 引  数        なし                                                  */
314 /* 戻り値        0:なし 1:あり                                         */
315 /*---------------------------------------------------------------------*/
316 int check_lcrank( void )
317 {
318     unsigned char b;
319     int ret;
320         ret = 0;
321         b = sensor_inp(MASK4_4);
322         if( b==0xf8 ) { /* ●●●  ●  ○○○  : 111 1 000 */
323                         /* ●●●●  ●○○○  : 1111 1000 */
324             ret = 1;
325         }
326         return ret;
327 }
328
329 /*---------------------------------------------------------------------*/
330 /* 処理概要      クロスライン検出処理                                  */
331 /* 引  数        なし                                                  */
332 /* 戻り値        0:なし 1:あり                                         */
333 /*---------------------------------------------------------------------*/
334 int check_crossline( void )
335 {
336     unsigned char b;
337     int ret;
338         ret = 0;
339         b = sensor_inp(MASK4_4);
340         if( b==0xff ) { /* ●●●  ●  ●●●  : 111 1 111 */
```

```
341                         /* ●●●●  ●●●●  : 1111 1111 */
342             ret = 1;
343         }
344         return ret;
345 }
346
347 /*---------------------------------------------------------------------*/
348 /* 処理概要      ディップスイッチ値読み込み                            */
349 /* 引　数        なし                                                  */
350 /* 戻り値        スイッチ値 0～15                                      */
351 /*---------------------------------------------------------------------*/
352 unsigned char dipsw_get( void )
353 {
354     unsigned char sw;
355         sw = ~P6DR; /* ディップスイッチ読み込み */
356         sw &= 0x0f;
357         return  sw;
358 }
359
360 /*---------------------------------------------------------------------*/
361 /* 処理概要      プッシュスイッチ値読み込                              */
362 /* 引　数        なし                                                  */
363 /* 戻り値        スイッチ値 ON:1 OFF:0                                 */
364 /*---------------------------------------------------------------------*/
365 unsigned char pushsw_get( void )
366 {
367     unsigned char sw;
368         sw = ~PBDR; /* スイッチのあるポート読み込み */
369         sw &= 0x01;
370         return  sw;
371 }
372
373 /*---------------------------------------------------------------------*/
374 /* 処理概要      乗り込みセンサ読み込み                                */
375 /* 引　数        なし                                                  */
376 /* 戻り値        センサ値 ON(乗り込んだ):1 OFF(乗り込んでいない):0     */
377 /*---------------------------------------------------------------------*/
378 unsigned char norikomi_get( void )
379 {
380     unsigned char b;
381         b = ~P7DR;
382         b &= 0x10;
383         b >>= 4;
384         return  b;
385 }
386
387 /*---------------------------------------------------------------------*/
388 /* 処理概要      LED制御                                               */
389 /* 引　数        スイッチ値 LED0:bit0 LED1:bit1  "0":消灯 "1":点灯     */
390 /*               例)0x3→LED1:ON LED0:ON  0x2→LED1:ON LED0:OFF         */
391 /* 戻り値        なし                                                  */
392 /*---------------------------------------------------------------------*/
393 void led_out( unsigned char led )
394 {
395     unsigned char data;
396
397         led = ~led;
398         led <<= 6;
399         data = PBDR & 0x3f;
400         PBDR = data | led;
401 }
402
403
404 /*---------------------------------------------------------------------*/
405 /* 処理概要      速度制御                                              */
406 /* 引　数        左モータ:-100～100 , 右モータ:-100～100               */
407 /*               0で停止、100で正転100%、-100で逆転100%                */
408 /* 戻り値        なし                                                  */
409 /*---------------------------------------------------------------------*/
410 void speed( int accele_l, int accele_r )
411 {
412     unsigned char   sw_data;
413     unsigned long   speed_max;
414
415         sw_data  = dipsw_get() + 5;         /* ディップスイッチ読み込み */
416         speed_max = (unsigned long)(PWM_CYCLE-1) * sw_data / 20;
417
418       /* 左モータ */
419         if( accele_l >= 0 ) {
420             PBDR &= 0xfb;
421             ITU3_BRB = speed_max * accele_l / 100;
422         } else {
423             PBDR |= 0x04;
424             accele_l = -accele_l;
425             ITU3_BRB = speed_max * accele_l / 100;
```

```
426         }
427
428       /* 右モータ */
429         if( accele_r >= 0 ) {
430             PBDR &= 0xf7;
431             ITU4_BRA = speed_max * accele_r / 100;
432         } else {
433             PBDR |= 0x08;
434             accele_r = -accele_r;
435             ITU4_BRA = speed_max * accele_r / 100;
436         }
437 }
438
439 /*-----------------------------------------------------------------------*/
440 /* 処理概要      サーボハンドル操作                                      */
441 /* 引　数        サーボ操作角度：-90～90                                 */
442 /*               -90で左へ90度、0でまっすぐ、90で右へ90度回転             */
443 /* 戻り値        なし                                                    */
444 /*-----------------------------------------------------------------------*/
445 void handle( int angle )
446 {
447         ITU4_BRB = SERVO_CENTER - angle * HANDLE_STEP;
448 }
449
450 /*-----------------------------------------------------------------------*/
451 /* モジュール名  led_off                                                 */
452 /* 処理概要      LED全消灯                                               */
453 /* 引　数        なし                                                    */
454 /* 戻り値        なし                                                    */
455 /*-----------------------------------------------------------------------*/
456 void led_off(void)
457 {
458         LED = 0x00; /* LED off */
459 }
460
461 /*-----------------------------------------------------------------------*/
462 /* 処理概要      LED全点灯                                               */
463 /* 引　数        なし                                                    */
464 /* 戻り値        なし                                                    */
465 /*-----------------------------------------------------------------------*/
466 void led_on(void)
467 {
468         LED = 0xff; /* LED on */
469 }
470
471 /*-----------------------------------------------------------------------*/
472 /* 処理概要      LED半点灯                                               */
473 /* 引　数        なし                                                    */
474 /* 戻り値        なし                                                    */
475 /*-----------------------------------------------------------------------*/
476 void led_on_f0(void)
477 {
478         LED = 0xf0; /* LED on */
479 }
480
481 /*-----------------------------------------------------------------------*/
482 /* 処理概要      LED半点灯                                               */
483 /* 引　数        なし                                                    */
484 /* 戻り値        なし                                                    */
485 /*-----------------------------------------------------------------------*/
486 void led_on_0f(void)
487 {
488         LED = 0x0f; /* LED on */
489 }
490
491 /*-----------------------------------------------------------------------*/
492 /* 処理概要      LED半点灯                                               */
493 /* 引　数        なし                                                    */
494 /* 戻り値        なし                                                    */
495 /*-----------------------------------------------------------------------*/
496 void led_on_aa(void)
497 {
498         LED = 0xaa; /* LED on 1010 1010 */
499 }
500
501 /*-----------------------------------------------------------------------*/
502 /* 処理概要      LED半点灯                                               */
503 /* 引　数        なし                                                    */
504 /* 戻り値        なし                                                    */
505 /*-----------------------------------------------------------------------*/
506 void led_on_55(void)
507 {
508         LED = 0x55; /* LED on 0101 0101 */
509 }
510
```

```
511 /*----------------------------------------------------------------------*/
512 /* 処理概要      A/D値読み込み(AN0)                                     */
513 /* 引　数        なし                                                   */
514 /* 戻り値        A/D値 0～1023                                          */
515 /*----------------------------------------------------------------------*/
516 int get_ad0_single( void )
517 {
518     int i;
519 
520         AD_CSR |= 0x20;                     /* ADスタート               */
521         while( !(AD_CSR & 0x80) );          /* エンドフラグをチェック   */
522         AD_CSR &= 0x7f;                     /* エンドフラグクリア       */
523         i = AD_DRA >> 6;                    /* 代入                     */
524 
525         return i;
526 }
527 
528 /*----------------------------------------------------------------------*/
529 /* 処理概要      A/D値読み込み(AN0)                                     */
530 /* 引　数        なし                                                   */
531 /* 戻り値        A/D値 0～255                                           */
532 /*               本当は0～1023だが下位2bit無視(ノイズ対策)              */
533 /*----------------------------------------------------------------------*/
534 int get_ad0_scan( void )
535 {
536         return AD_DRA >> 6;
537 }
538 
539 /*----------------------------------------------------------------------*/
540 /* 処理概要      A/D値読み込み(AN1)                                     */
541 /* 引　数        なし                                                   */
542 /* 戻り値        A/D値 0～255                                           */
543 /*               本当は0～1023だが下位2bit無視(ノイズ対策)              */
544 /*----------------------------------------------------------------------*/
545 int get_ad1_scan( void )
546 {
547         return AD_DRB >> 6;
548 }
549 
550 /*----------------------------------------------------------------------*/
551 /* 処理概要      A/D値読み込み(AN0)シングルモード                       */
552 /* 引　数        なし                                                   */
553 /* 戻り値        A/D値 0～255(本当は0～1023だが下位2bit無視             */
554 /*----------------------------------------------------------------------*/
555 void ad0_single_test( void )
556 {
557     int ad;
558         while( 1 ) {
559               ad = get_ad0_single();
560               P4DR = ad;
561               printf( "%05d\n", ad );
562         }
563 }
564 
565 /*----------------------------------------------------------------------*/
566 /* 処理概要      A/D値読み込み(AN0)スキャンモード                       */
567 /* 引　数        なし                                                   */
568 /* 戻り値        A/D値 0～255(本当は0～1023だが下位2bit無視             */
569 /*----------------------------------------------------------------------*/
570 void ad0_scan_test( void )
571 {
572     int ad;
573         while( 1 ) {
574               ad = get_ad0_scan();
575               P4DR = ad;
576               printf( "%05d\n", ad );
577               timer(100);
578         }
579 }
580 
581 /*----------------------------------------------------------------------*/
582 /* 処理概要      A/D値読み込み(AN1)スキャンモード                       */
583 /* 引　数        なし                                                   */
584 /* 戻り値        A/D値 0～255(本当は0～1023だが下位2bit無視             */
585 /*----------------------------------------------------------------------*/
586 void ad1_scan_test( void )
587 {
588     int ad;
589         while( 1 ) {
590               ad = get_ad1_scan();
591             //PADR = ad;
592               P4DR = ad;
593               printf( "%05d\n", ad );
594         }
595 }
```

```
596
597 /*-----------------------------------------------------------------*/
598 /* 処理概要      ライントレース                                    */
599 /* 引　数        なし                                              */
600 /* 戻り値        なし                                              */
601 /*-----------------------------------------------------------------*/
602 void trace( void )
603 {
604                                         /* ○○○　－　○○○ */
605         switch( sensor_inp(MASK3_3) ) { /* ○○○－　－○○○ */
606
607             case 0x00: /* 中心のみ反応 or コースアウト */
608                  speed( 100 , 100 );
609                  break;
610
611             case 0x04:  /* ○○○－　－●○○ */ /* 000- -100 */
612                  speed( 100 ,  80 );
613                  break;
614
615             case 0x06:  /* ○○○－　－●●○ */ /* 000- -110 */
616                  speed( 100,   60 );
617                  break;
618
619             case 0x03:  /* ○○○－　－○●● */ /* 000- -011 */
620                  speed( 100,   40 );
621                  break;
622
623             case 0x20:  /* ○○●－　－○○○ */ /* 001- -000 */
624                  speed(  80 , 100 );
625                  break;
626
627             case 0x60:  /* ○●●－　－○○○ */ /* 011- -000 */
628                  speed(  60 , 100 );
629                  break;
630
631             case 0xc0:  /* ●●○－　－○○○ */ /* 011- -000 */
632                  speed(  40 , 100 );
633                  break;
634
635             default:
636                  break;
637
638         }/* end of switch( sensor_inp(MASK3_3) ) */
639 }
640
641 /*-----------------------------------------------------------------------*/
642 /* メインプログラム                                                      */
643 /*-----------------------------------------------------------------------*/
644 void main( void )
645 {
646     int     i;
647
648       /* マイコン機能の初期化 */
649          init();                 /* 初期化           */
650          init_sci1( 0x00, 79 ); /* SCI1初期化       */
651          set_ccr( 0x00 );       /* 全体割り込み許可 */
652
653       /* マイコンカーの状態初期化 */
654          handle( 0 );
655          speed( 0, 0 ); /* speed( 左モータの速度(0～100) , 右モータの速度(0～100) ) */
656                         /* ロボット停止 */
657
658                 /*----------------------*/
659                 /* アナログセンサテスト */
660                 /*----------------------*/
661                  //ad0_single_test();
662                  /*ad0_scan_test();*/
663
664          pattern = 0;
665          while( 1 ) {
666                 switch( pattern ) {
667             /*---------------------------------*/
668             /* 電源ON後スタートスイッチ入力待ち */
669             /*---------------------------------*/
670                    case 0:
671                         if( pushsw_get() || PUSH_SW ) {
672                             pattern = 1;
673                             cnt1 = 0;
674                             break;
675                         }
676                              if( cnt1 < 100 ) { /* LED点滅処理 */
677                                  led_out( 0x1 );
678                                  led_on_0f();
679                              } else if( cnt1 < 200 ) {
680                                  led_out( 0x2 );
```

```
                                led_on_f0();
                            } else {
                                cnt1 = 0;
                            }
                        break;/* end of case 0: */


                /*-------------------------------------------------------------*/
                /* スタートスイッチが押されてのでLEDを4回点滅しスタートする。 */
                /*-------------------------------------------------------------*/
                case 1:
                    for( i=0; i<4; i++) {
                        led_out( 0x03 );
                        led_on();
                        timer(500);
                        led_out( 0x00 );
                        led_off();
                        timer(500);
                    }
                    pattern = 11;
                    cnt1 = 0;
                    break;/* end of case 1: */


                /*--------------------------------------*/
                /* 往路走行(1/2) 右クランクまで走行する。 */
                /* 障害物あり                           */
                /*--------------------------------------*/
                case 11:
                    if( check_rcrank() ) { /* 右クランクか? */
                        pattern = 21;
                        cnt1 = 0;
                        break;
                    }
                  /*-----------------------------------------*/
                  /* 右クランク見つるまでライントレースする。 */
                  /* 障害物あるが現状は何も対策していない     */
                  /*-----------------------------------------*/
                    trace();
                    break;/* break of case 11: */


                /*-------------------*/
                /* 右クランク発見した */
                /*-------------------*/
                case 21: /* まず減速 */
                    led_out( 0x03 );
                    led_on_0f();
                    speed( 50 , 50 );
                    pattern = 22;
                    cnt1 = 0;
                    break;/* end of case 21: */

                case 22: /* そのまましばらく前進する */
                    if( cnt1 >  800 ) { /* cnt1は自動カウントアップ cnt1>1000 で約1秒位 */
                                        /* 要調整!! */
                        pattern = 23;
                        cnt1 = 0;
                    }
                    break;/* end of case 22: */

                case 23: /* ロボット停止する */
                    speed(  0 ,  0 );
                    pattern = 24;
                    cnt1 = 0;
                    break;/* end of case 23: */

                case 24: /* 姿勢安定まで待つ */
                    if( cnt1 > 1000 ) { /* 約1秒wait */
                        pattern = 25;
                        cnt1 = 0;
                    }
                    break;/* end of case 24: */

                case 25: /* 右信地旋回する */
                    speed( 100 , -100 ); /* 左モータ正転100% 右モータ逆転100% ∴右信地旋回 */
                    pattern = 26;
                    cnt1 = 0;
                    break;/* end of case 25: */

                case 26: /* センサが反応するまで旋回する。(要調整!!) */
                   //if( sensor_inp(MASK4_0) != 0) { /* ○○○○　－－－－*/
                    if( sensor_inp(MASK1_8) != 0) { /* －－－○　○－－－*/
                                                    /* 中心センサ反応するまで */
                   //if( sensor_inp(MASK0_4) != 0) { /* －－－－　○○○○*/
```

```
766                         pattern = 27;
767                         cnt1 = 0;
768                    }
769                    break;/* end of case 26: */
770
771               case 27: /* ロボット停止する */
772                    led_out( 0x0 );
773                    speed( 0 , 0 );
774                    pattern = 28;
775                    cnt1 = 0;
776                    break;/* end of case 27: */
777
778               case 28: /* 姿勢安定まで待つ */
779                    if( cnt1 > 1000 ) {
780                         pattern = 31;
781                         cnt1 = 0;
782                    }
783                    break;/* end of case 28: */
784
785
786               /*-------------------------------------*/
787               /* 往路走行(2/2) 競技台まで走行する。    */
788               /* 障害物なし                          */
789               /* (リモコンロボットを迎えに行く)       */
790               /*-------------------------------------*/
791               case 31:
792                    if( get_ad0_scan()>400 ) { /* 競技台手前10cm程度 */
793                      //speed( 30 , 30 ); /* 低速で競技台に体当たりする */
794                         speed( 10 , 10 ); /* 競技台押しつけ不具合あり(要改善) */
795                         led_out( 0x0 );
796                         pattern = 32;
797                         cnt1 = 0;
798                         break;
799                    }
800                 /*-------------------------------------*/
801                 /* 競技台見つるまでライントレースする。  */
802                 /* 障害物なし                          */
803                 /*-------------------------------------*/
804                    trace();
805                    break;/* end of case 31: */
806
807              /*-------------------------------------*/
808              /* 競技台と密着させるためにそのまま前進 */
809              /* その後ロボット停止                   */
810              /*-------------------------------------*/
811                 case 32:
812                    if( cnt1 > 500 ) {
813                         speed( 0 , 0 );
814                         pattern = 33;
815                         cnt1 = 0;
816                    }
817                    break;/* end of case 32: */
818
819               /*---------------------------------*/
820               /* リモコンロボット乗り込むまで待機 */
821               /*---------------------------------*/
822               case 33:
823                    if( norikomi_get() ) { /* リモコンロボット乗り込んだか? */
824                     /* スタート！！ */
825                         led_out( 0x0 );
826                         led_off();
827                         pattern = 34;
828                         cnt1 = 0;
829                         break;
830                    }
831                         if( cnt1 < 100 ) { /* LED点滅処理 */
832                              led_out( 0x1 );
833                              led_on_aa();
834                         } else if( cnt1 < 200 ) {
835                              led_out( 0x2 );
836                              led_on_55();
837                         } else {
838                              cnt1 = 0;
839                         }
840                    break;/* end of case 33: */
841
842
843               /*-----------------------------------------------------------*/
844               /* リモコンロボットが乗り込んだので、LEDを4回点滅し再スタートする。 */
845               /*-----------------------------------------------------------*/
846               case 34: /* 少しバックする */
847                    for( i=0; i<3; i++) {
848                         led_out( 0x03 );
849                         led_on();
850                         timer(500);
```

```
                    led_out( 0x00 );
                    led_on();
                    timer(500);
                }
                speed( -80 , -80);
                pattern = 35;
                cnt1 = 0;
                break;/* end of case 34: */

        case 35: /* ロボット停止 */
                if( cnt1 > 500 ) {
                    speed( 0 , 0 );
                    pattern = 36;
                    cnt1 = 0;
                }
                break;/* end of case 35: */

        case 36: /* 姿勢安定まで待つ */
                if( cnt1 > 1000 ) { /* 約1秒wait */
                    pattern = 37;
                    cnt1 = 0;
                }
                break;/* end of case 36: */

        case 37: /* 右180deg信地旋回する */
                speed( 100 , -100 );   /* 左モータ正転100% 右モータ逆転100% ∴右信地旋回 */
                pattern = 38;
                cnt1 = 0;
                break;/* end of case 37: */

        case 38: /* センサがラインをはずすまでで待つ */
                if( cnt1 > 800 ) {
                    pattern = 39;
                    cnt1 = 0;
                }
                break;/* end of case 38: */

        case 39: /* センサが反応するまで旋回する。 */
                 /* ラインが見つかったら停止する。 */
           //if( sensor_inp(MASK4_0) != 0) { /* ○○○○  ----*/
             if( sensor_inp(MASK1_8) != 0) { /* ---○  ○---*/
           //if( sensor_inp(MASK0_4) != 0) { /* ----  ○○○○*/
                    speed( 0 , 0 );
                    pattern = 40;
                    cnt1 = 0;
                }
                break;/* end of case 39: */

        case 40: /* 姿勢安定まで待つ */
                if( cnt1 > 1000 ) {
                    pattern = 41;
                    cnt1 = 0;
                }
                break;/* end of case 40: */


    /*--------------------------------------*/
    /* 往路走行(1/2) 左クランクまで走行する。*/
    /* 障害物なし                           */
    /* (リモコンロボット搬送)               */
    /*--------------------------------------*/
        case 41:
                if( check_lcrank() ) { /* 左クランクか? */
                    pattern = 51;
                    cnt1 = 0;
                    break;
                }
          /*-----------------------------------------*/
          /* 左クランク見つるまでライントレースする。 */
          /* 障害物なし                              */
          /*-----------------------------------------*/
             trace();
             break;/* break of case 41: */


    /*-------------------------------*/
    /* 左クランクまで走行した時の処理 */
    /*-------------------------------*/
        case 51: /* まず減速 */
                led_out( 0x03 );
                speed( 50 , 50 );
                pattern = 52;
                cnt1 = 0;
                break;/* end of case 51: */

```

```
936                     case 52: /* そのまましばらく前進する */
937                          if( cnt1 > 800 ) { /* cnt1は自動カウントアップ cnt1>1000 で約1秒位 */
938                              pattern = 53;
939                              cnt1 = 0;
940                          }
941                          break;/* end of case 52: */
942
943                     case 53: /* ロボット停止する */
944                          speed(  0 ,  0 );
945                          pattern = 54;
946                          cnt1 = 0;
947                          break;/* end of case 53: */
948
949                     case 54: /* 姿勢安定まで待つ */
950                          if( cnt1 > 1000 ) { /* 約1秒wait */
951                              pattern = 55;
952                              cnt1 = 0;
953                          }
954                          break;/* end of case 54: */
955
956                     case 55: /* 左信地旋回する */
957                          speed( -100 , 100 ); /* 左モータ逆転100% 右モータ正転100% ∴左信地旋回 */
958                          pattern = 56;
959                          cnt1 = 0;
960                          break;/* end of case 55: */
961
962
963                     case 56: /* センサが反応するまで旋回する。 */
964                        //if( sensor_inp(MASK4_0) != 0) { /* ○○○○  ーーーー*/
965                          if( sensor_inp(MASK1_8) != 0) { /* ーーー○  ○ーーー*/
966                        //if( sensor_inp(MASK0_4) != 0) { /* ーーーー  ○○○○*/
967                              pattern = 57;
968                              cnt1 = 0;
969                          }
970                          break;/* end of case 56: */
971
972                     case 57: /* ロボット停止する */
973                          led_out( 0x3 );
974                          speed( 0 , 0 );
975                          pattern = 58;
976                          cnt1 = 0;
977                          break;/* end of case 57: */
978
979                     case 58: /* 姿勢安定まで待つ */
980                          if( cnt1 > 1000 ) {
981                              pattern = 61;
982                              cnt1 = 0;
983                          }
984                          break;/* end of case 58: */
985
986
987                /*-----------------------------*/
988                /* 往路走行(2/2) 障害物あり    */
989                /* 競技台まで走行する。        */
990                /* リモコンロボット搬送        */
991                /*-----------------------------*/
992                     case 61:  /* トレース4 リモコンロボットをスタートゾーンに戻る */
993                          if( get_ad0_scan()>400 ) { /* 競技台手前10cm程度 */
994                            //speed( 30 , 30 ); /* 低速で前進し競技台にぶつかる */
995                              speed( 10 , 10 ); /* 競技台押しつけ、不具合あり */
996                              led_out( 0x0 );
997                              pattern = 62;
998                              cnt1 = 0;
999                              break;
1000                         }
1001                       /*--------------------------------------*/
1002                       /* 競技台見つるまでライントレースする。 */
1003                       /* 障害物あり                           */
1004                       /* とりあえず何も対策しない             */
1005                       /*--------------------------------------*/
1006                          trace();
1007                          break;/* end of case 61: */
1008
1009
1010               /*-----------------------------*/
1011               /* 競技終了                    */
1012               /*-----------------------------*/
1013                     case 62: /* リモコンロボット降りたか */
1014                          if( norikomi_get() ) {
1015                              led_out( 0x0 );
1016                              speed( 0 , 0 );
1017                              pattern = 63;
1018                              cnt1 = 0;
1019                              break;
1020                          }
```

```
1021                         break;/* end of case 62: */
1022
1023                    /*---------------------------------------*/
1024                    /* リモコンロボットが降りたのでロボット停止 */
1025                    /*---------------------------------------*/
1026                       case 63:
1027                            for( i=0; i<2; i++) {
1028                                 led_out( 0x03 );
1029                                 led_on();
1030                                 timer(500);
1031                                 led_out( 0x00 );
1032                                 led_off();
1033                                 timer(500);
1034                            }
1035                            pattern = 64;
1036                            cnt1 = 0;
1037                            break;/* end of case 63: */
1038
1039
1040                       case 64: /* 競技終了 */
1041                            pattern = 0;
1042                            break;/* end of case 64: */
1043
1044
1045                       default: /* どれでもない場合は待機状態に戻す */
1046                            pattern = 0;
1047                            break;
1048
1049                  }/* end of switch( pattern ) */
1050
1051             }/* while( 1 ) */
1052
1053 }/* end of main */
1054 /*------------------------------------------------------------------------*/
1055 /* end of file                                                            */
1056 /*------------------------------------------------------------------------*/
```