

inst.c

```
1 #include <msp430g2553.h>
2 #include <stdio.h>
3 #include <stdlib.h> //乱数関数 PC内のある規則に従って、疑似乱数を発生させる
4 #include <time.h> //現在の時刻を秒数で返す関数
5
6 #define DATA BIT0 // DATA P2.0
7 #define CLOCK BIT1 // CLOCK P2.1
8 #define LATCH BIT2 // ST_CP P2.2
9 #define SORENOIDO BIT4 // ソレノイド P2.4
10 #define LED BIT0 // LED青 P1.0
11 #define OE BIT1 // OE P1.1
12 #define SW1 BIT2 // stopボタン左 P1.2
13 #define SW2 BIT3 // stopボタン中央 P1.3
14 #define SW3 BIT4 // stopボタン右 P1.4
15 #define SW4 BIT5 // startボタン P1.5
16 #define LED2 BIT6 // 枚数MAX LED赤 P1.6
17 #define SENSEA BIT7 // フォトインタラプタ P1.7
18
19
20
21 void led(){
22     P1DIR |= LED; // P1.0を出力設定
23     P1OUT |= LED; // P1.0を1出力(LED青 点灯)
24 }
25
26 void led2(){
27     P1DIR |= LED2; // P1.6を出力設定
28     P1OUT &= ~LED2; // P1.6を0出力(LED赤 点灯)
29 }
30
31
32 void sensa(){
33     P1OUT |= SENSEA; // P1.7に1を設定
34     P1IE |= SENSEA; // センサの割り込みイネーブル
35     P1IES |= SENSEA; // 立ち下がりで割り込み発生
36     P1IFG &= ~SENSEA; // 割り込みクリア
37 }
38
39 void sorenoido(){
40     P2DIR |= SORENOIDO; // P2.4を出力設定
41     P2OUT |= SORENOIDO; // P2.4を1出力
42 }
43
44 void sw1(){
45     P1OUT |= SW1; // P1.2に1を設定
46     P1IE |= SW1; // ストップ左の割り込みイネーブル
47     P1IES |= SW1; // 立ち下がりで割り込み発生
48     P1IFG &= ~SW1; // 割り込みクリア
49 }
50
51 void sw2(){
52     P1OUT |= SW2; // P1.3に1を設定
53     P1IE |= SW2; // ストップ中央の割り込みイネーブル
54     P1IES |= SW2; // 立ち下がりで割り込み発生
55     P1IFG &= ~SW2; // 割り込みクリア
56 }
57
58 void sw3(){
59
60     P1OUT |= SW3; // P1.4に1を設定
61     P1IE |= SW3; // ストップ右の割り込みイネーブル
62     P1IES |= SW3; // 立ち下がりで割り込み発生
63     P1IFG &= ~SW3; // 割り込みクリア
64 }
65
66 void sw4(){
67
68     P1OUT |= SW4; // P1.5に1を設定
69     P1IE |= SW4; // スタートの割り込みイネーブル
70     P1IES |= SW4; // 立ち下がりで割り込み発生
71     P1IFG &= ~SW4; // 割り込みクリア
72 }
73
74
75
76 void delay(int ms){
```

```

77     while(ms--){
78
79
80         __delay_cycles(10);
81     }
82 }
83
84 void myclock(){
85     P2OUT |= CLOCK; // P2.1に1を設定
86     P2OUT ^= CLOCK; // P2.1を反転
87 }
88
89 void ratch(){
90     P2OUT &= ~LATCH; // P2.2に0を設定
91     P2OUT |= LATCH; // P2.2に1を設定
92     P2OUT &= ~LATCH; // P2.2に0を設定
93 }
94
95 void oe(){
96     P1DIR |= OE; //P1.1を出力設定
97     P1OUT &= ~OE; //P1.1を0出力
98 }
99
100 void data(int bit){
101     if(bit){ //もしbitならP2.0に0を設定
102         P2OUT &= ~DATA;
103     }
104     else{
105         P2OUT |= DATA; // それ以外ならP2.0に1を設定
106     }
107 }
108
109 void shiftout(int dat){ //datの呼び起こし
110     int i; // 変数i定義
111     for(i=7;i>=0;i--){ // iを7~0まで繰り返す
112         data((dat>>i)&1); //datを右にシフトしたものと1のand
113         myclock(); //Clock関数呼び出し
114     }
115
116 }
117
118 int stop1;
119 int stop2;
120 int stop3;
121 int start;
122 int sen;
123 int SEG_NUMBER[10]={
124     0x3F,//0
125     0x06,//1
126     0x5B,//2
127     0x4F,//3
128     0x66,//4
129     0x6D,//5
130     0x7D,//6
131     0x27,//7
132     0x7F,//8
133     0x6F,//9
134
135
136
137     };
138
139
140 int main( void ){
141
142
143     // Stop watchdog timer to prevent time out reset
144
145     WDTCTL = WDTPW + WDTHOLD;
146     P2DIR |= (DATA + CLOCK + LATCH );// P2.0、P2.1、P2.2を出力設定
147
148     srand(time(NULL)); //乱数の初期化 時刻情報を利用
149
150
151     sw1();
152     sw2();

```

```

153 sw3();
154 sw4();
155 sensa();
156 oe();
157
158 _BIS_SR(GIE); //割り込み関数 許可
159 P1OUT |= LED2; //P1.6に1を設定(LED赤消灯)
160 P2OUT &= ~SORENOIDO; //P2.4に0を設定(ソレノイドoff)
161
162 int g = 0; //7segカウント10の位
163 int h = 0; //7segカウント1の位
164 int i = 8; //7seg左
165 int j = 8; //7seg右
166 int k = 8; //7seg中央
167 int l = 0; //センサに1が入ったらスタート可能
168
169
170 shiftout(SEG_NUMBER[0]); //7seg初期状態
171 shiftout(SEG_NUMBER[0]);
172 shiftout(SEG_NUMBER[8]);
173 shiftout(SEG_NUMBER[8]);
174 shiftout(SEG_NUMBER[8]);
175 ratch(); //ラッチ起動
176
177 while(1){
178
179     if(sen == 1){ //センサを通過したら
180         h++; //カウント7seg 1の位を+1
181         P1OUT |= LED2; //LED赤 消灯
182         P2OUT &= ~SORENOIDO; //ソレノイドoff
183
184         if(h<=9){ //カウント7seg 1のくらいが9以下なら何もしない
185             }
186         else{ //9を超えたら1の位は0に戻る
187             h=0;
188         }
189         if(h == 0){ //もし1の位が0になったら、10の位を+1
190             g++;
191             if(g == 10){ //もし10の位が9になったら
192                 g=0; //10の位は0へ
193                 led2(); //LED赤 点灯
194                 sorenoido(); //ソレノイド オン
195             }
196         }
197     else{
198
199     }
200
201     sen=0; //センサを通過したら、センサは0に戻る
202     l++; //スタート可能回数+1
203     start=0; //スタート、ストップは押されていない状態へ
204     stop1=0;
205     stop2=0;
206     stop3=0;
207
208     i=8; // i,j,kは8に戻る
209     j=8;
210     k=8;
211
212
213     }
214     else{
215
216     }
217
218
219
220     shiftout(SEG_NUMBER[g]); //7seg 5つ目
221     shiftout(SEG_NUMBER[h]); //7seg 4つ目
222     shiftout(SEG_NUMBER[i]); //7seg 3つ目
223     shiftout(SEG_NUMBER[j]); //7seg 2つ目
224     shiftout(SEG_NUMBER[k]); //7seg 1つ目
225     ratch();
226
227     delay(10000); //ルーレットの速度
228     if(l == 0){ //センサを通過していなかったら戻る

```

```

229         continue;
230     }
231     else{
232     }
233     }
234     if(start == 1){ //スタートが押されたら次のコードへ
235     }
236     }
237     else{ //押されていないかったら戻る
238         continue;
239     }
240
241     if(1 <= stop1 && stop1<= 55){ //stop1が押されたらフラグ45まで乱数を表示
242         switch(stop1){
243             case 1:
244             case 3:
245             case 6:
246             case 10:
247             case 15:
248             case 21:
249             case 28:
250             case 36:
251             case 45:
252
253                 i = rand() % 10; // 10で割った余りを取得すれば、0~9の値が得られる
254                 break;
255             case 54://フラグ54になりhが2になったら、i=7
256                 if(h == 2){
257                     i = 7;
258                 }
259                 else{
260                     i = rand()%10;//それ以外は乱数表示
261                 }
262                 break;
263             }
264
265             stop1++;
266
267         }
268     }
269     else if(stop1 == 56){
270     }
271     }
272
273     else{
274         i++;
275     }
276     if(1 <= stop2 && stop2<= 55){
277         switch(stop2){
278             case 1:
279             case 3:
280             case 6:
281             case 10:
282             case 15:
283
284                 case 21:
285                 case 28:
286                 case 36:
287                 case 45:
288
289                 j = rand() % 10;
290                 break;
291             case 54:
292                 if(h == 2){
293                     j = 7;
294                 }
295                 else{
296                     j=rand()%10;
297                 }
298                 break;
299             }
300
301             stop2++;
302         }
303     }
304     else if(stop2 == 56){

```

```

305     }
306     else{
307         j++;
308     }
309     if(1 <= stop3 && stop3<= 55){
310         switch(stop3){
311             case 1:
312             case 3:
313             case 6:
314             case 10:
315             case 15:
316
317                 case 21:
318                 case 28:
319                 case 36:
320                 case 45:
321
322             k = rand() % 10;
323             break;
324             case 54:
325                 if(h == 2){
326                     k = 7;
327                 }
328                 else{
329                     k=rand()%10;
330                 }
331             break;
332         }
333     stop3++;
334 }
335     }
336     else if(stop3 == 56){
337     }
338 }
339 else{
340     k++;
341 }
342 }
343
344 if(stop1 == 56 && stop2 == 56 && stop3 == 56){//stopが全て止まったら
345     l--; //スタート可能-1
346     start=0; //start stop押されていない状態へ
347     stop1=0;
348     stop2=0;
349     stop3=0;
350
351
352     if(i == 7 && j == 7 && k == 7){
353         led();
354         sorenoido();
355         g=0;
356         h=0;
357
358         P1OUT |= OE;
359         delay(10000);
360         P1OUT &= ~OE;
361         delay(10000);
362         P1OUT |= OE;
363         delay(10000);
364         P1OUT &= ~OE;
365         delay(10000);
366         P1OUT |= OE;
367         delay(10000);
368         P1OUT &= ~OE;
369         delay(10000);
370         P1OUT |= OE;
371         delay(10000);
372         P1OUT &= ~OE;
373         delay(10000);
374         P1OUT |= OE;
375         delay(10000);
376         P1OUT &= ~OE;
377         delay(10000);
378         P1OUT |= OE;
379         delay(10000);
380         P1OUT &= ~OE;
381         delay(10000);

```

```

382         P1OUT |= OE;
383         delay(10000);
384         P1OUT &= ~OE;
385         delay(10000);
386         P1OUT |= OE;
387         delay(10000);
388         P1OUT &= ~OE;
389         delay(10000);
390         P1OUT |= OE;
391         delay(10000);
392         P1OUT &= ~OE;
393         delay(10000);
394         P1OUT |= OE;
395         delay(10000);
396         P1OUT &= ~OE;
397         delay(10000);
398
399
400
401     }
402
403
404
405     }
406     else{
407         P1OUT &= ~LED;
408         P2OUT &= ~SORENOIDO;
409
410     }
411
412
413     if(i>=10){
414         i=0;
415     }
416     if(j>=10){
417         j=0;
418     }
419     if(k>=10){
420         k=0;
421     }
422
423
424 }
425
426 }
427 #pragma vector=PORT1_VECTOR
428 __interrupt void Port1(void){
429
430
431 if(P1IFG & BIT3){
432 if(start == 1 && stop1 == 0){
433     stop1=1;
434 }
435 else{
436
437 }
438 }
439 P1IFG &= ~BIT3;
440
441 if(P1IFG & BIT4){
442 if(start == 1 && stop2 == 0){
443     stop2=1;
444 }
445 else{
446
447 }
448 }
449 P1IFG &= ~BIT4;
450
451 if(P1IFG & BIT5){
452     if(start == 1 && stop3 == 0){
453         stop3=1;
454     }
455     else{
456
457     }
458 }

```

inst.c

```
459
460 P1IFG &= ~BIT5;
461
462
463 if(P1IFG & BIT2){
464     if(start == 0){
465         start=1;
466     }
467     else{
468     }
469 }
470 }
471 P1IFG &= ~BIT2;
472
473 if(P1IFG & BIT7){
474
475     sen=1;
476 }
477
478 P1IFG &= ~BIT7;
479 }
480
481
```