

1 課題

1.1 課題内容の補足

クリア条件として, 岩をよけて画面から消えると +1 点・岩に当たると-5 点・10 点でゲームクリアという仕様にした.

1.2 プログラムリスト

課題で作成したプログラムをソースコード 1 に示す.

ソースコード 1 課題のソースコード

```
1  /*
2  *  memes_2012.c : 2012年度MEMEs 総合演習
3  *
4  */
5
6  #include <stdlib.h>
7  #include "iodefine.h"
8  #include "typedefine.h"
9
10 #define printf ((int (*)(const char *,...))0x00007c7c)
11
12 #define SW6 (PD.DR.BIT.B18)
13 #define SW5 (PD.DR.BIT.B17)
14 #define SW4 (PD.DR.BIT.B16)
15
16 #define LED6 (PE.DR.BIT.B11)
17 #define LED_ON (0)
18 #define LED_OFF (1)
19
20 #define DIG1 (PE.DR.BIT.B3)
21 #define DIG2 (PE.DR.BIT.B2)
22 #define DIG3 (PE.DR.BIT.B1)
23
24 #define SPK (PE.DR.BIT.B0)
25
26 #define LCD_RS (PA.DR.BIT.B22)
27 #define LCD_E (PA.DR.BIT.B23)
28 #define LCD_RW (PD.DR.BIT.B23)
29 #define LCD_DATA (PD.DR.BYTE.HH)
30
31 #define Boolean int
32 #define True 1
33 #define False 0
34
35 #define NMROF_ROCKS 6
```

```

36
37 struct position {
38     int x;
39     int y;
40     int active;
41 };
42 int point;
43
44 //当たり判定フラグ
45 int flag = 0;
46
47 void wait_us(_UINT);
48 void LCD_inst(_SBYTE);
49 void LCD_data(_SBYTE);
50 void LCD_cursor(_UINT, _UINT);
51 void LCD_putch(_SBYTE);
52 void LCD_putstr(_SBYTE *);
53 void LCD_cls(void);
54 void LCD_init(void);
55 void init_peior(void);
56 void init_paior(void);
57
58
59 // -----
60 // -- 使用する関数群 --
61 // -----
62 void wait_us(_UINT us) {
63     _UINT val;
64
65     val = us * 10 / 16;
66     if (val >= 0xffff)
67         val = 0xffff;
68
69     CMT0.CMCOR = val;
70     CMT0.CMCSR.BIT.CMF &= 0;
71     CMT.CMSTR.BIT.STR0 = 1;
72     while (!CMT0.CMCSR.BIT.CMF);
73     CMT0.CMCSR.BIT.CMF = 0;
74     CMT.CMSTR.BIT.STR0 = 0;
75 }
76
77 void LCD_inst(_SBYTE inst) {
78     LCD_E = 0;
79     LCD_RS = 0;
80     LCD_RW = 0;
81     LCD_E = 1;
82     LCD_DATA = inst;
83     wait_us(1);
84     LCD_E = 0;

```

```

85         wait_us(40);
86     }
87
88     void LCD_data(_SBYTE data) {
89         LCD_E = 0;
90         LCD_RS = 1;
91         LCD_RW = 0;
92         LCD_E = 1;
93         LCD_DATA = data;
94         wait_us(1);
95         LCD_E = 0;
96         wait_us(40);
97     }
98
99     void LCD_cursor(_UINT x, _UINT y) {
100         if (x > 15)
101             x = 15;
102         if (y > 1)
103             y = 1;
104         LCD_inst(0x80 | x | y << 6);
105     }
106
107     void LCD_putch(_SBYTE ch) {
108         LCD_data(ch);
109     }
110
111     void LCD_putstr(_SBYTE *str) {
112         _SBYTE ch;
113
114         while (ch = *str++)
115             LCD_putch(ch);
116     }
117
118     void LCD_cls(void) {
119         LCD_inst(0x01);
120         wait_us(1640);
121     }
122
123     void LCD_init(void) {
124         wait_us(45000);
125         LCD_inst(0x30);
126         wait_us(4100);
127         LCD_inst(0x30);
128         wait_us(100);
129         LCD_inst(0x30);
130
131         LCD_inst(0x38);
132         LCD_inst(0x08);
133         LCD_inst(0x01);

```

```

134     wait_us(1640);
135     LCD_inst(0x06);
136     LCD_inst(0x0c);
137 }
138
139 // -----
140 // -- ゲーム用の関数群 --
141
142 // -- 自分を移動 --
143 void move_me(struct position *me)
144 {
145     struct position old_position;
146
147     old_position.x = me->x;
148     old_position.y = me->y;
149
150     if (AD0.ADDR0 < 0x4000) {
151         // -- ジョイスティック上 --
152         me->y = 0;
153     } else if (AD0.ADDR0 > 0xc000) {
154         // -- ジョイスティック下 --
155         me->y = 1;
156     }
157
158     if (AD0.ADDR1 < 0x4000) {
159         // -- ジョイスティック右 --
160         if(me->x >= 15){
161             me->x = 15;
162         } else {
163             me->x += 1;
164         }
165     } else if (AD0.ADDR1 > 0xc000) {
166         // -- ジョイスティック左 --
167         if(me->x <= 0){
168             me->x = 0;
169         } else {
170             me->x -= 1;
171         }
172     }
173
174     if (old_position.y != me->y || old_position.x != me->x) {
175         // -- 移動したとき .. 古い表示を消す --
176         flag = False;
177         LCD_cursor(old_position.x, old_position.y);
178         LCD_putch(' ');
179     }
180     LCD_cursor(me->x, me->y);
181     LCD_putch('>');
182 }

```



```

232 }
233
234 void init_peior(void) {
235     PFC.PEIORL.BIT.B3 = 1;
236 }
237 void init_paior(void) {
238     PFC.PAIORH.BYTE.L |= 0x0F;
239 }
240
241
242 // -----
243 // -- メイン関数 --
244 void main(){
245     while(1){
246         struct position me;    // 自分の車の座標
247         struct position rock[NMROF_ROCKS]; // 岩の座標
248         int move_timing, new_timing;
249         int ad, i;
250         point = 0;
251         flag = False;
252
253         STB.CR4.BIT._AD0 = 0;
254         STB.CR4.BIT._CMT = 0;
255         STB.CR4.BIT._MTU2 = 0;
256
257         CMT0.CMCSR.BIT.CKS = 1;
258
259         // MTU2 ch0
260         MTU20.TCR.BIT.TPSC = 3; // 1/64選択
261         MTU20.TCR.BIT.CCLR = 1; // TGRA のコンペアマッチでクリア
262         MTU20.TGRA = 31250 - 1; // 100ms
263         MTU20.TIER.BIT.TTGE = 1; // A/D 変換開始要求を許可
264
265         // AD0
266         AD0.ADCSR.BIT.ADM = 3; // 2チャンネルスキャンモード
267         AD0.ADCSR.BIT.CH = 1; // AN0
268         AD0.ADCSR.BIT.TRGE = 1; // MTU2 からのトリガ有効
269         AD0.ADTSR.BIT.TRG0S = 1; // TGRA コンペアマッチでトリガ
270
271         // MTU2 ch1
272         MTU21.TCR.BIT.TPSC = 3; // 1/64選択
273         MTU21.TCR.BIT.CCLR = 1; // TGRA のコンペアマッチでクリア
274         MTU21.TGRA = 31250 - 1; // 100ms
275
276         LCD.init();
277
278         MTU2.TSTR.BIT.CST0 = 1; // MTU2 CH0 スタート
279         MTU2.TSTR.BIT.CST1 = 1; // MTU2 CH1 スタート
280

```



```

329         wait_us(5000);
330         printf("\nSEE_YA!\n");
331         break;
332     }
333     if(point > 9){
334         //clær
335         printf("\nCLEAR!\n"); //debug
336         break;
337     } else if (point < 0){
338         printf("\nGAME_OVER!\n"); //debug
339         break;
340     }
341     if (MTU21.TSR.BIT.TGFA) {
342         // MTU2 ch1 コンペアマッチ発生 (100ms 毎)
343         MTU21.TSR.BIT.TGFA = 0; // フラグクリア
344
345         move_me(&me); // 自分移動
346         if (move_timing++ >= 2) {
347             move_timing = 0;
348             move_rock(rock, me); // 岩を移動
349             if (new_timing-- <= 0) {
350                 new_timing = rand() * 5 / (RAND_MAX + 1);
351                 new_rock(rock); // 新しい岩が出現
352             }
353         }
354     }
355 }
356 }
357 }
358 }

```

1.3 使用した機能の説明

今回使用した機能について、技術的な説明を記す。MEMES サポートページ http://memes.sakura.ne.jp/memes/?page_id=295 も合わせて参照されたし。

1.3.1 スイッチ

ゲームの開始・一時停止・リセットを実現するために、スイッチを SW4～SW6 まで使用した。それぞれのスイッチが押されることにより PD.DR.BIT.B16 PD.DR.BIT.B18 の値が自動的に変化することを用いて制作を行った。

1.3.2 ジョイスティック

上下左右にカーソルを移動するために使用した。A/D 変換を行ってアナログジョイスティックの値をデジタル値として取得している。AD0 を 2 チャンネルスキャンモードにし、AN0 を上下の動きに、AN1 を左右の動きに対応させた。

1.3.3 7セグメントLED

仕様上1桁使うことができれば良いので, シンプルに `PA.DR.BYTE.HL |= point` といった形で表示させた.

2 オリジナル機能

実装したオリジナル機能について示す.

- 起動時にメッセージを表示
- リセットしても、またゲームを一から開始できるようにした