

基于at89c51的猜数字小游戏

- 1 @author:q
- 2 实验环境:keil5,proteus 8.6
- 3 项目git位置:<https://github.com/9cij/c51>

游戏流程

电脑随机产生一个数字,玩家输入,如果大了,就输出大了,如果小了,就输出小了,然后继续猜,直到猜对为止.

按下enter键,游戏开始

按加减键修改数字,切换用来控制修改十位数或是个位数

按下enter提交输入,猜错会提示大了或者小了,可以继续猜

对了会提示win

win之后等5s会自动返回首页,也可以随时通过restart返回首页

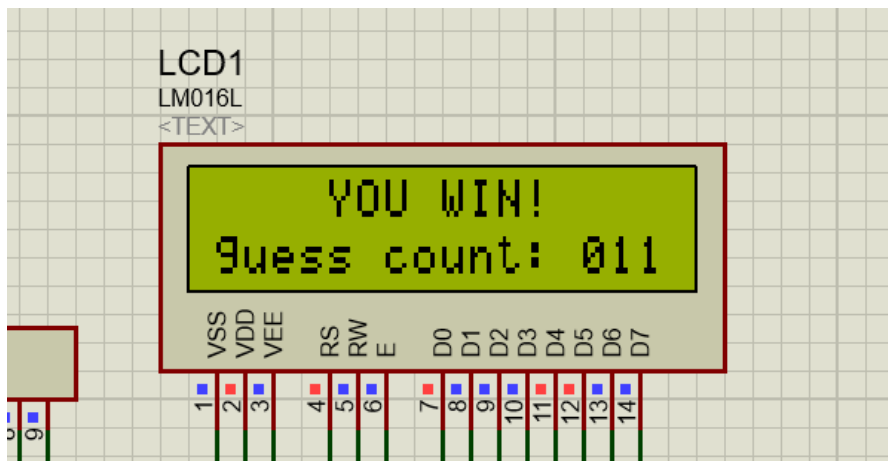
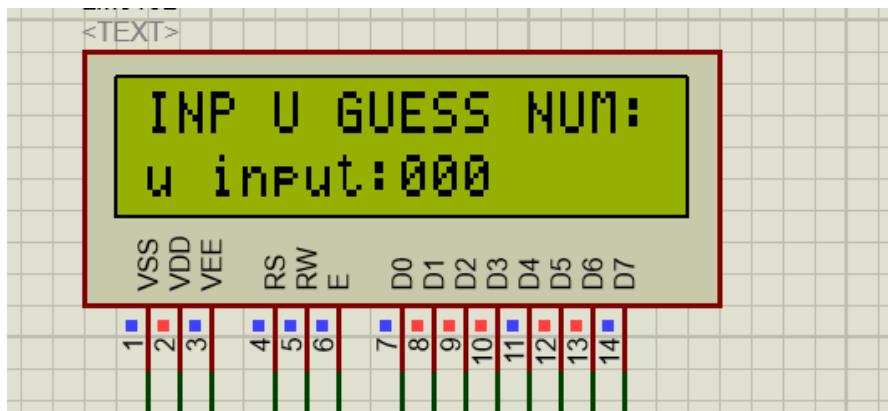
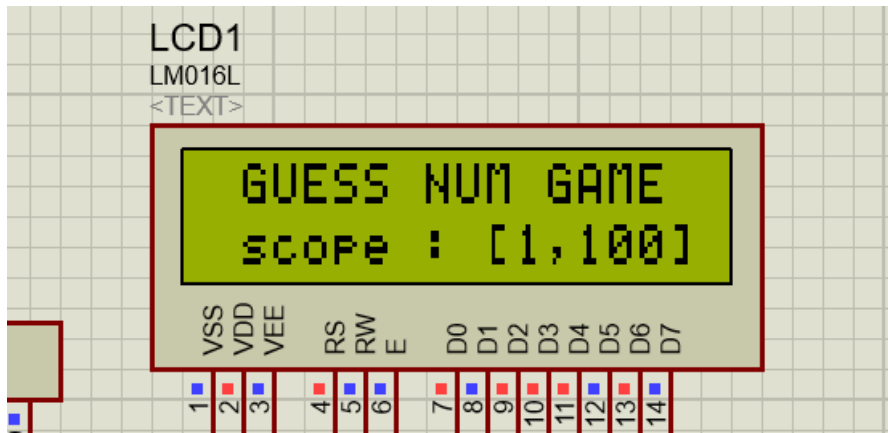
设计思路

首先需要有三个页面

首页,显示游戏的名字(guess num game)和基本信息(随机值产生的范围,1到100的左闭右闭区间),响应游戏开始的按键

游戏开始的页面,第一行为input u guess num,当点击确认时,第一行会变成提示(big,small,win),第二行为用户的输入值,每次对输入值的修改,都会实时显示到第二行

游戏胜利的界面(只能胜利或者重置),第一行固定(you win),第二行显示用户尝试的次数,该页面等待5s后会自动返回首页



然后需要调节数字的五个按键

按键一:加

按键二:减

按键三:切换修改的位数,初始为个位,点击一次则会切到十位,思路来自于可调时钟,十位数的调节直接+10或-10即可

按键四:start/commit 第一次点击是告诉程序,可以开始游戏了,第二次是提交所猜测的数字

按键五:restart 随时都能点击,重置游戏状态,回到首页

程序逻辑

1.初始化,程序通电,初始化液晶显示器,简易的锁逻辑,一些必要的全局变量

2.获取一个随机值

3.进入首页

4.开始监听按键事件

4.1监听enter键

4.1.1第一次按下enter的时候,才会进入游戏,第二次是提交输入,所以这里有个enter_flag来记录当前是第几次按下,第一次按下后,enter_flag自增,然后进入到游戏页

4.1.2然后显示用户输入的值在显示器第二行,初始值是0

4.1.3将key设为1(unlock),key的初值是0,监听按键事件中,仅当key非0的时候才会监听(响应)切换和加减按键

4.2第一次按下enter后,key=1,就解锁了,所以程序开始监听k2(切换)键

4.2.1key=1时,程序可以对个位数进行加减操作,按下k2(切换键),开始操作十位数

4.2.2程序在每次操作个位数和十位数后,都会直接展示出来

4.3程序再次按下enter键,enter_flag自增,变成2

4.3.1guess_count++记录提交次数

4.3.2开始对比用户输入和系统产生的随机值

4.3.2.1如果相同,赢,将enter_flag设为0,结束监听提交事件,key设为0,锁住切换和加减按键

4.3.2.2大了,输出大了,小了,输出小了,enter_flag设为1,继续监听提交事件

5.监听restart按键

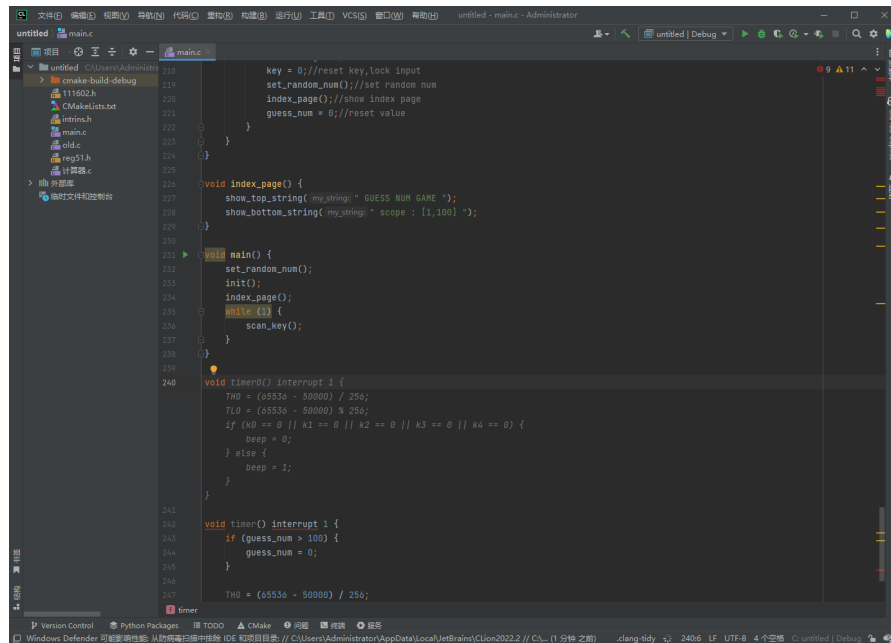
5.1检测到restart按下,key置为0,随机数也重新生成,跳转到首页,尝试猜测次数设为0,enter_flag设为0

小技巧

clion编写代码,方便格式化,

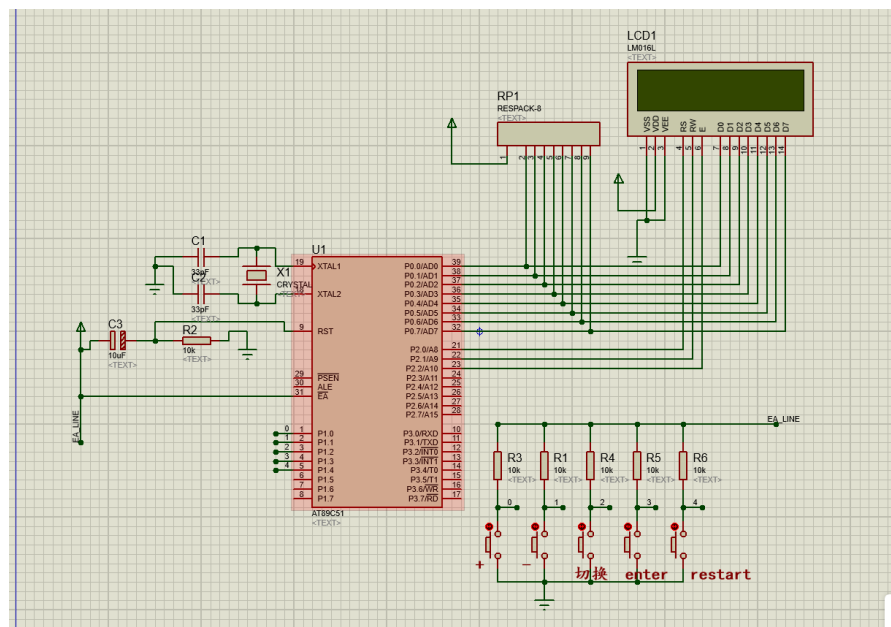
配合github copilot自动补全,写个函数名和注释,ai帮你写代码

边写边调试,新增功能就直接复制到keil5编译输出hex文件,dsn图直接仿真测试



```
120      key = 0; //reset key, lock input
121      set_random_num(); //set random num
122      index_page(); //show index page
123      guess_num = 0; //reset value
124  }
125  }
126  }
127  }
128  }
129  }
130  }
131  }
132  }
133  }
134  }
135  }
136  }
137  }
138  }
139  }
140  }
141  }
142  }
143  }
144  }
145  }
146  }
147  }
148  }
149  }
150  }
151  }
152  }
153  }
154  }
155  }
156  }
157  }
158  }
159  }
160  }
161  }
162  }
163  }
164  }
165  }
166  }
167  }
168  }
169  }
170  }
171  }
172  }
173  }
174  }
175  }
176  }
177  }
178  }
179  }
180  }
181  }
182  }
183  }
184  }
185  }
186  }
187  }
188  }
189  }
190  }
191  }
192  }
193  }
194  }
195  }
196  }
197  }
198  }
199  }
200  }
201  }
202  }
203  }
204  }
205  }
206  }
207  }
208  }
209  }
210  }
211  }
212  }
213  }
214  }
215  }
216  }
217  }
218  }
219  }
220  }
221  }
222  }
223  }
224  }
225  }
226  }
227  }
228  }
229  }
230  }
231  }
232  }
233  }
234  }
235  }
236  }
237  }
238  }
239  }
240  }
241  }
242  }
243  }
244  }
245  }
246  }
247  }
248  }
249  }
250  }
251  }
252  }
253  }
254  }
255  }
256  }
257  }
258  }
259  }
260  }
261  }
262  }
263  }
264  }
265  }
266  }
267  }
268  }
269  }
270  }
271  }
272  }
273  }
274  }
275  }
276  }
277  }
278  }
279  }
280  }
281  }
282  }
283  }
284  }
285  }
286  }
287  }
288  }
289  }
290  }
291  }
292  }
293  }
294  }
295  }
296  }
297  }
298  }
299  }
300  }
301  }
302  }
303  }
304  }
305  }
306  }
307  }
308  }
309  }
310  }
311  }
312  }
313  }
314  }
315  }
316  }
317  }
318  }
319  }
320  }
321  }
322  }
323  }
324  }
325  }
326  }
327  }
328  }
329  }
330  }
331  }
332  }
333  }
334  }
335  }
336  }
337  }
338  }
339  }
340  }
341  }
342  }
343  }
344  }
345  }
346  }
347  }
348  }
349  }
350  }
351  }
352  }
353  }
354  }
355  }
356  }
357  }
358  }
359  }
360  }
361  }
362  }
363  }
364  }
365  }
366  }
367  }
368  }
369  }
370  }
371  }
372  }
373  }
374  }
375  }
376  }
377  }
378  }
379  }
380  }
381  }
382  }
383  }
384  }
385  }
386  }
387  }
388  }
389  }
390  }
391  }
392  }
393  }
394  }
395  }
396  }
397  }
398  }
399  }
400  }
401  }
402  }
403  }
404  }
405  }
406  }
407  }
408  }
409  }
410  }
411  }
412  }
413  }
414  }
415  }
416  }
417  }
418  }
419  }
420  }
421  }
422  }
423  }
424  }
425  }
426  }
427  }
428  }
429  }
430  }
431  }
432  }
433  }
434  }
435  }
436  }
437  }
438  }
439  }
440  }
441  }
442  }
443  }
444  }
445  }
446  }
447  }
448  }
449  }
450  }
451  }
452  }
453  }
454  }
455  }
456  }
457  }
458  }
459  }
460  }
461  }
462  }
463  }
464  }
465  }
466  }
467  }
468  }
469  }
470  }
471  }
472  }
473  }
474  }
475  }
476  }
477  }
478  }
479  }
480  }
481  }
482  }
483  }
484  }
485  }
486  }
487  }
488  }
489  }
490  }
491  }
492  }
493  }
494  }
495  }
496  }
497  }
498  }
499  }
500  }
```

原理图:



代码

简单的定义一下需要用到的东西:

```

1 #include <reg51.h>
2 #include <111602.h>
3 #include <stdio.h>
4 #include <stdlib.h>
5
6 sbit k0 = P1 ^ 0; //add
7 sbit k1 = P1 ^ 1; //sub
8 sbit k2 = P1 ^ 2; //change
9 sbit k3 = P1 ^ 3; //enter
10 sbit k4 = P1 ^ 4; //restart
11
12 #define uchar unsigned char
13 #define uint unsigned int
14
15 void init(); //初始化,通电,初始化111602.h,设置
16 void set_random_num(); //设置随机数,且随机需要time.h,这里用rand()函数,属于假随机
17 void re_delay(int n); //重写了111602.h中的delay函数,用来计时5s,uchar最大是244,将uchar改写成了int,
18 void my_delay(); //清屏函数
19 void show_top_string(char my_string[]); //显示顶部字符串,第一行,0x00
20 void show_bottom_string(char my_string[]); //显示底部字符串,第二行,0x40
21 void show_value(); //显示用户输入的当前值
22 void show_win(); //显示胜利的页面,专门写一个函数,因为要显示胜利的所尝试的次数
23 void set_ten(); //调教函数,这是调十位数的
24 void set_unit(); //调教函数,这是调个位数的
25 void compare(); //比较函数,用来比较用户输入的值和随机数的大小
26 void scan_key(); //扫描按键函数
27 void index_page(); //首页函数,显示欢迎页面
28
29 int random_num = 0;
30 /*猜数字小游戏,电脑随机产生一个数字,玩家输入,如果大了,就输出大了,如果小了,就输出小了,然后继续猜,直到猜对为止*/
31
32 const uint len_line = 16; //一行的最大字符数
33 const uchar my_nums[] = "0123456789"; //用来显示数字的字符数组
34 const uchar win_string[] = "guess count:"; //胜利页面的基础字符串
35 uint i; //循环用,每次使用前会重新赋值
36 uint guess_num = 0; //用户输入的数字
37 uint key = 0; // 0:锁定输入 1:修教十位 2:修教个位
38 uint enter_flag = 0; //是否按下了enter键,第一次按下enter键:开始游戏,第二次按下enter键:调用比较函数,检查输入是否和随机数相等
39 uint guess_count = 0; //尝试的次数
40

```

工具类:

```

41 void init() { //must, only run once
42     EA = 1;
43     ET0 = 1;
44     TMOD = 0X11;
45     TH0 = (65536 - 50000) / 256;
46     TL0 = (65536 - 50000) % 256;
47     TR0 = 1;
48     /*-----*/
49     LcdInitiate(); //init 111602.h
50     WriteAddress(0x00);
51     delay(5); //must, only run once
52 }
53
54 void set_random_num() {
55     //random get a num from 0 to 99
56     random_num = rand() % 100 + 1;
57 }
58
59 void re_delay(int n)
60 {
61     i=0;
62     for(i<n;i++)
63         delay1ms();
64 }
65
66 void my_delay() {
67     //delay
68     uint j = 0;
69     for (j; j < 26000; j++);
70 }
71

```

```

70 }
71
72 void show_top_string(char my_string[]) {
73     //print specify string as top line
74     WriteAddress(x: 0x00);
75
76     i = 0;
77     for (i; i < len_line; i++) {
78         WriteData(y: my_string[i]);
79     }
80 }
81
82 void show_bottom_string(char my_string[]) {
83     //print specify string as bottom line
84     WriteAddress(x: 0x40);
85     i = 0;
86     for (i; i < len_line; i++) {
87         WriteData(y: my_string[i]);
88     }
89 }
90
91 void show_value() {
92     WriteAddress(x: 0x40);
93     WriteData(y: 'u');
94     WriteData(y: ' ');
95     WriteData(y: 'i');
96     WriteData(y: '\n');
97     WriteData(y: 'p');
98     WriteData(y: 'u');
99     WriteData(y: 't');
100    WriteData(y: ':');
101    WriteData(y: my_nums[guess_num / 100]);
102    WriteData(y: my_nums[guess_num % 100 / 10]);
103    WriteData(y: my_nums[guess_num % 10]);
104
105    // debug, show random num
106    // WriteData(my_nums[random_num/100]);
107    // WriteData(my_nums[random_num%100 / 10]);
108    // WriteData(my_nums[random_num%10]);
109 }
110

```

```

110
111 void show_win() {
112     show_top_string(my_string: " YOU WIN! ");
113     WriteAddress(x: 0x40);
114     i = 0;
115     for (i; i < 13; i++) {
116         WriteData(y: win_string[i]);
117     }
118
119     WriteData(y: my_nums[guess_count / 100]);
120     WriteData(y: my_nums[guess_count % 100 / 10]);
121     WriteData(y: my_nums[guess_count % 10]);
122
123     re_delay(n: 5000); //delay 5s
124     k4=0; //restart
125 }
126

```

```

126
127 void set_ten() {
128     //set ten
129     if (k0 == 0) {
130         my_delay();
131         if (k0 == 0) { guess_num = guess_num + 10; }
132     }
133
134     if (k1 == 0) {
135         my_delay();
136         if (k1 == 0) {
137             if (guess_num < 10) { guess_num = 99; }
138             else { guess_num = guess_num - 10; }
139         }
140     }
141 }
142
143 void set_unit() {
144     //set unit
145     if (k0 == 0) {
146         my_delay();
147         if (k0 == 0) { guess_num++; }
148     }
149     if (k1 == 0) {
150         my_delay();
151         if (k1 == 0) {
152             if (guess_num == 0) { guess_num = 99; }
153             else { guess_num--; }
154         }
155     }
156 }
157

```

```

157
158 void compare() {
159     //compare user guess num and random num
160     // show_bottom_string();
161     if (guess_num == random_num) {
162         enter_flag = 0;
163         key = 0; //lock input
164         show_win();
165     } else if (guess_num > random_num) {
166         show_top_string( my_string: "    TOO BIG!    ");
167         show_bottom_string( my_string: "                ");
168         enter_flag = 1;
169     } else if (guess_num < random_num) {
170         show_top_string( my_string: "    TOO SMALL!  ");
171         show_bottom_string( my_string: "                ");
172         enter_flag = 1;
173     }
174 }
175

```

```

176 void scan_key() {
177     //hook key press event
178     if (k3 == 0) { //enter
179         my_delay();
180         if (k3 == 0) {
181             enter_flag++;
182             if (enter_flag == 1) { //first press enter, start game
183                 show_top_string( my_string: "INP U GUESS NUM:");
184                 show_bottom_string( my_string: " ");
185                 show_value();
186                 key = 1; //press enter, unlock input
187             }
188         }
189     }
190
191     if (enter_flag == 2) { //second press enter, compare user input num and random num
192         guess_count++; compare();
193     }
194
195     /*****-----*****/
196     if (key != 0) {
197         if (k2 == 0) { //change set
198             my_delay();
199             if (k2 == 0) {
200                 key++;
201                 if (key > 2) { key = 1; }
202             }
203         }
204         if (key == 1) { //first set unit
205             set_unit();
206             show_value();
207         } else if (key == 2) { //second set ten
208             set_ten();
209             show_value();
210         }
211     }
212
213     /*****-----*****/
214     if (k4 == 0) { //restart

```

```

212
213     /*****-----*****/
214     if (k4 == 0) { //restart
215         my_delay();
216         if (k4 == 0) {
217             key = 0; //reset key, lock input
218             set_random_num(); //set random num
219             index_page(); //show index page
220             guess_num = 0; //reset value
221         }
222     }
223 }
224

```

```

224
225 void index_page() {
226     show_top_string( my_string: " GUESS NUM GAME ");
227     show_bottom_string( my_string: " scope : [1,100] ");
228 }
229

```



```

230 void main() {
231     set_random_num();
232     init();
233     index_page();
234     while (1) {
235         scan_key();
236     }
237 }
238
239 void timer() interrupt 1 {
240     if (guess_num > 100) {
241         guess_num = 0;
242     }
243
244     TH0 = (65536 - 50000) / 256;
245     TL0 = (65536 - 50000) % 256;
246 }
247
248

```

感想

一开始的想法很多,想过做井字棋,贪吃蛇,四则运算计算器,二八十六进制转换器,但是嫌麻烦,最后选了一个猜数字的小游戏来做,总体实现起来没什么难度,偶尔出点小bug,也是因为加锁后忘记释放,有时候hex文件要跟dsn文件放到同一目录才能正常仿真,单片机debug的时候非常折磨人心态,所以初学还是适合做一个效果就编译仿真一次,单片机还是挺有意思的