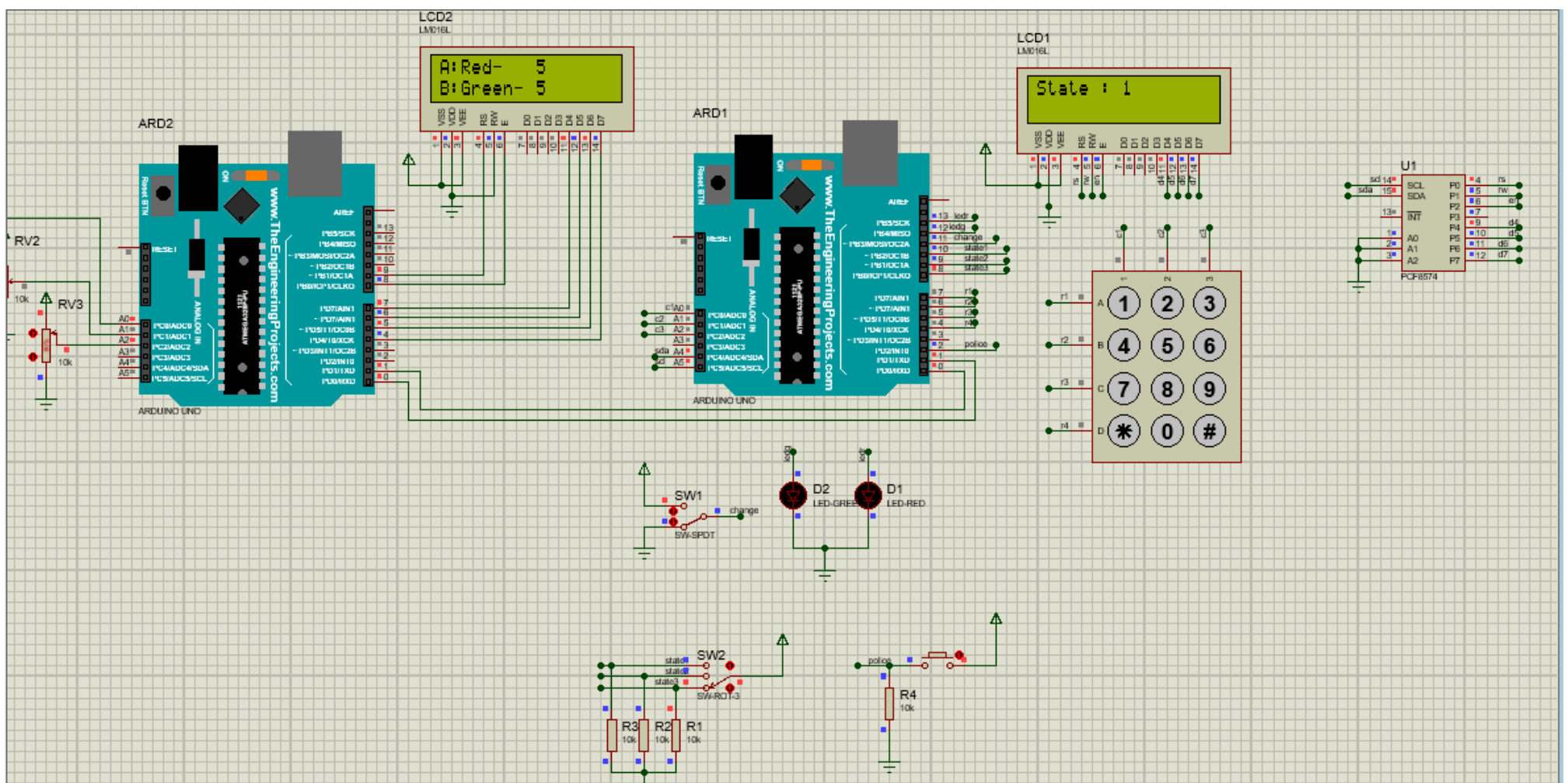


مدار ما از دو برد Arduino ساخته شده است، برد سمت راست برای تنظیم رمز ورود و تعیین mode عملکرد چراغ ها هست. به Arduino راست، ماژول های keypad و lcd وصل شده است. keypad برای وارد کردن رمز چهار رقمی و یا تعیین رمز جدید هست. روی نمایشگر lcd ، mode عملکرد چراغ ها نمایش داده می شود به علاوه عدد ورودی keypad، همچنان اعلان هایی مثل تغییر رمز، رمز اشتباه، وارد کردن رمز جدید و ... نیز روی آن نمایش داده می شود. به جهت محدود بودن تعداد پین های قابل استفاده در برد راست، lcd به وسیله یک واسطه spi به دو پین spi اردوینو وصل شده است.

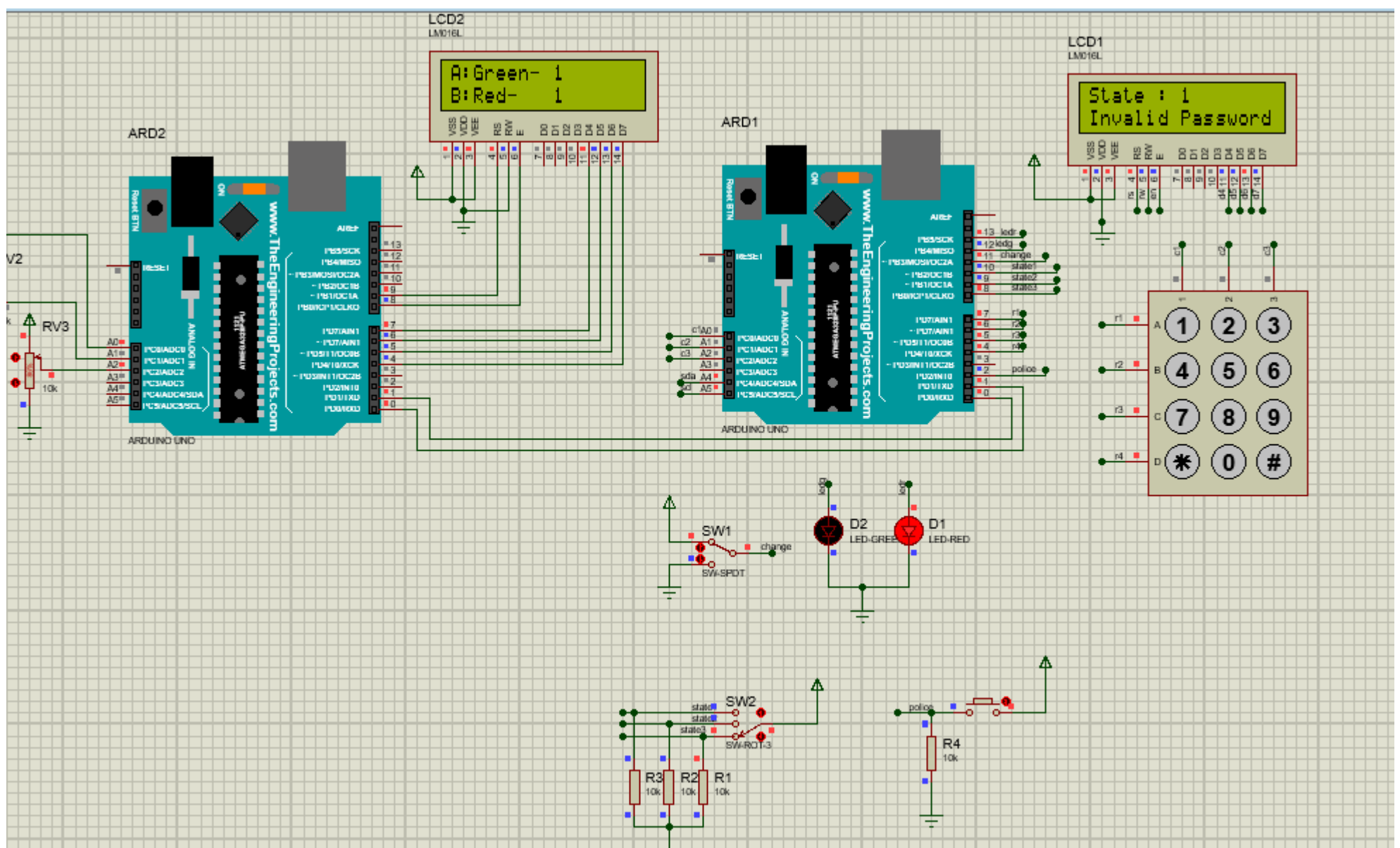
علاوه بر موارد بالا، دو led سبز و قرمز نیز به عنوان اعلان های نمایش دهنده ورود و یا تغییر رمز به برد متصل هست. ما یک کلید rot3 سه حالتی که نمایانگر تعیین سه mode کاری چراغ ها هست را نیز به سه پین برد وصل کرده ایم. هم چنین یک push button نیز به عنوان دکمه پلیس (مطابق توضیحات پروپوزال پروژه) به برد متصل هست. در نهایت یک سویچ ساده sw1 به عنوان قفل keypad به برد وصل شده است.

در سوی دیگر در سمت چپ برد Arduino دوم را داریم که یک lcd و سه ورودی ولتاژ متغیر به آن متصل شده اند. در روی lcd زمان باقی مانده هر چراغ و رنگش مشخص می شود. سه ورودی آنالوگ متغیر در واقع مدل شده ی سه سنسور تردد شمار هست که به صورت تغییرات ولتاژ آنالوگ وارد adc برد خواهد شد. لازم به ذکر است، دو برد راست و چپ از طریق پروتکل UART که به همدیگر متصل شده اند، با هم تبادل اطلاعات خواهند کرد.

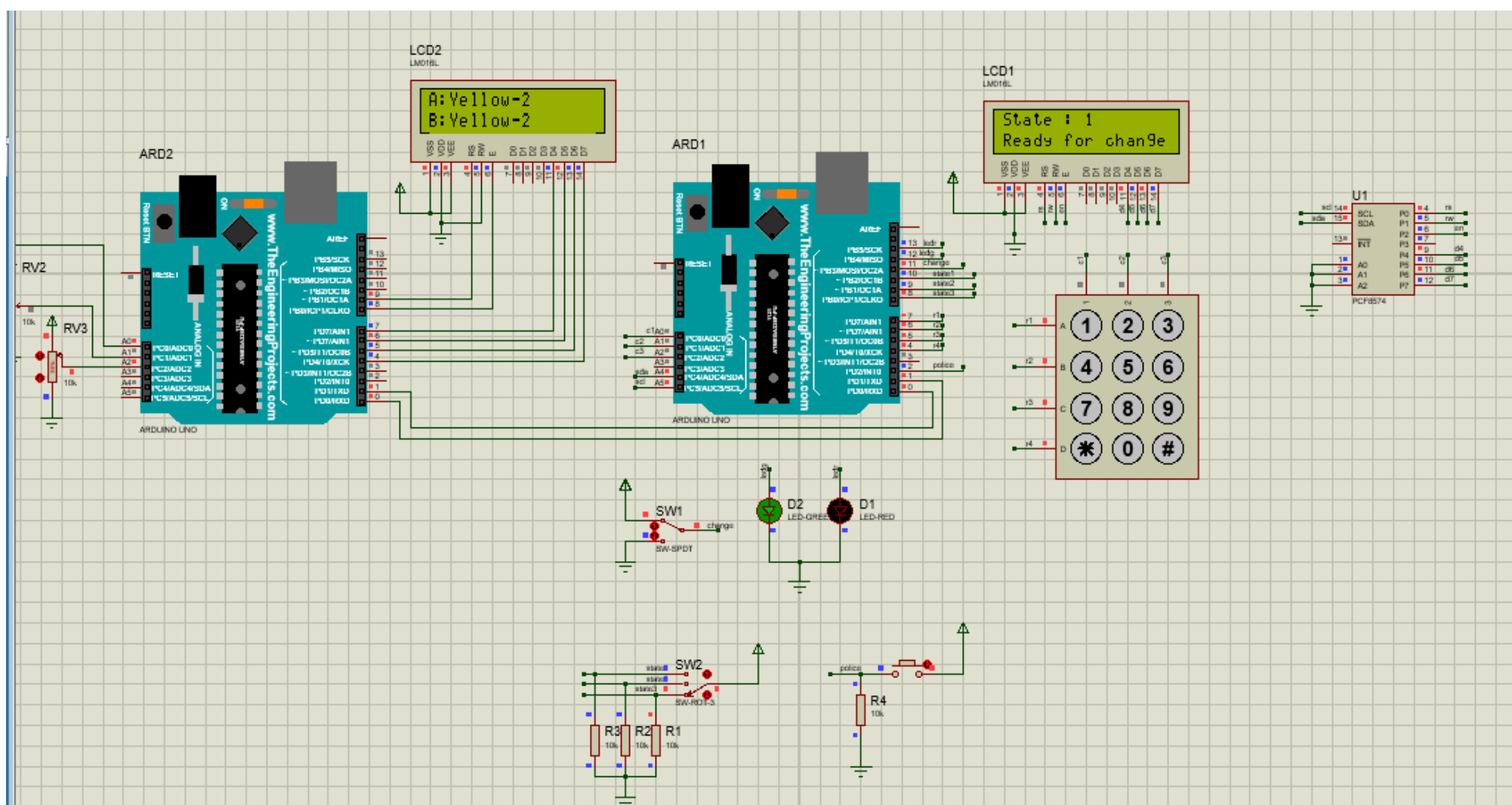
چند نمونه از اجرای واقعی برد را در ادامه می بینیم:



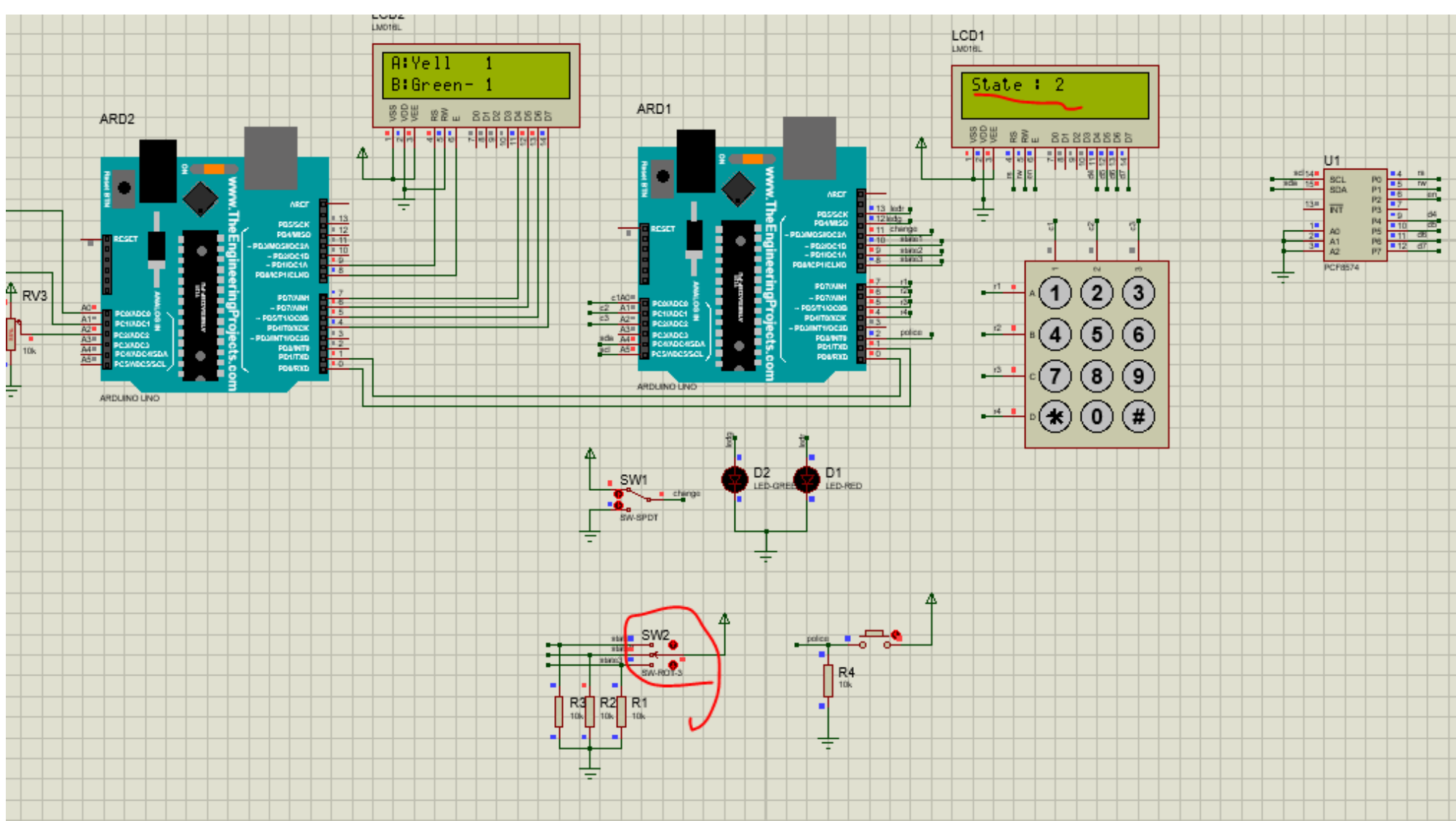
state کاری چراغ ها به صورت پیشفرض رو حالت یک یعنی شمارنده خودکار قرار دارد، در نمایشگر راست این را میبینیم، همچنین در lcd چپ زمان دو چراغ a و b و رنگ چراغ ها نشان داده شده است.



ما دکمه قفل keypad را به بالا می بریم، سپس یک رمز ۴ رقمی اشتباه را وارد می کنیم، مطابق تصویر، led قرمز ، سه بار چشمک میزند، و جمله پسوورد غلط نشان داده می شود.

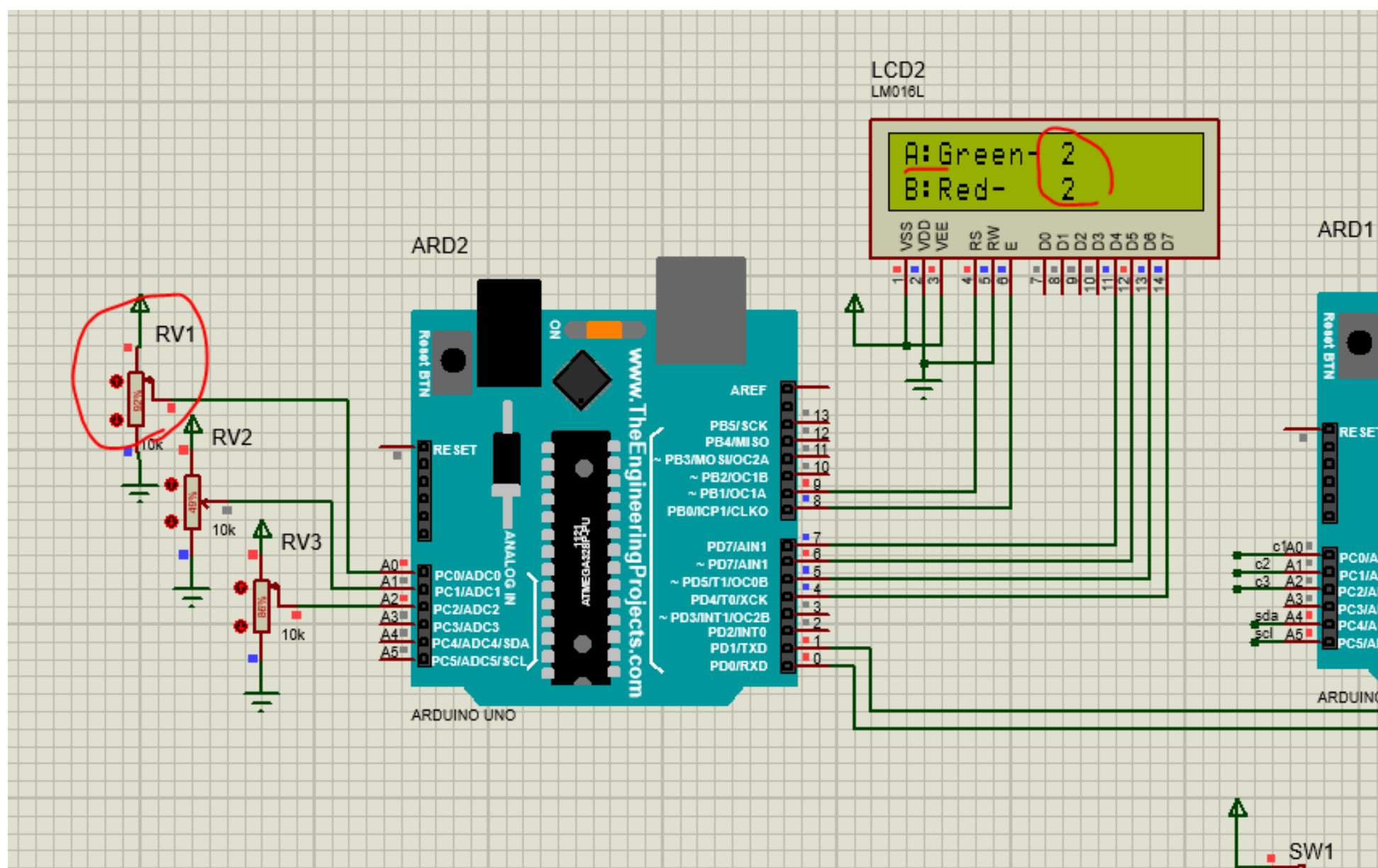


هنگامی که پسورد درست به صورت پیشفرض ۲۰۲۱ زده می شود، چراغ سبز سه بار چشمک میزند، و جمله آماده برای تغییر وضعیت نشان داده میشود. اکنون باتوجه به انتخاب دکمه state، یکی از سه وضعیت ۱ تا ۳ ذخیره می شود، مثلاً اگر دکمه روی state ۲ باشد، به شکل زیر می شود:



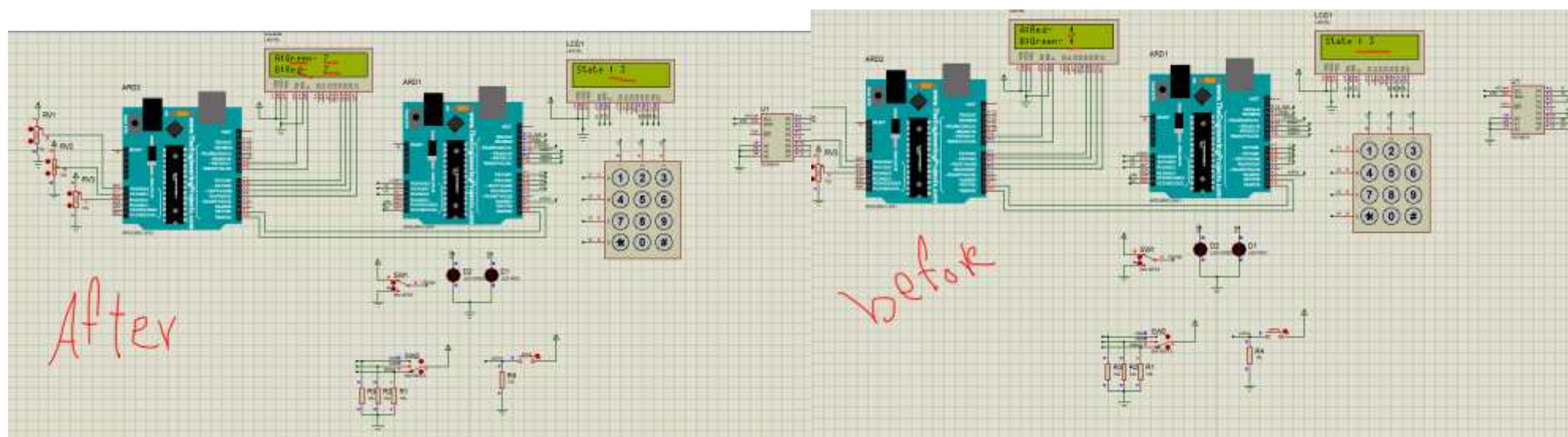
همان طور که مشاهده می شود، چون دکمه state روی ۲ بوده هست، به نمایشگر منتقل شده و نشان داده شده است. در این حالت باتوجه به وضعیت تردد سه جاده، جاده ای که در آن حجم تردد بیشتر باشد، ۲ ثانیه مانده به پایان چراغ سبزش، چراغ ها فریز می شوند، برای نمونه در عکس زیر مسیر A تردد بیشتری داشته است:





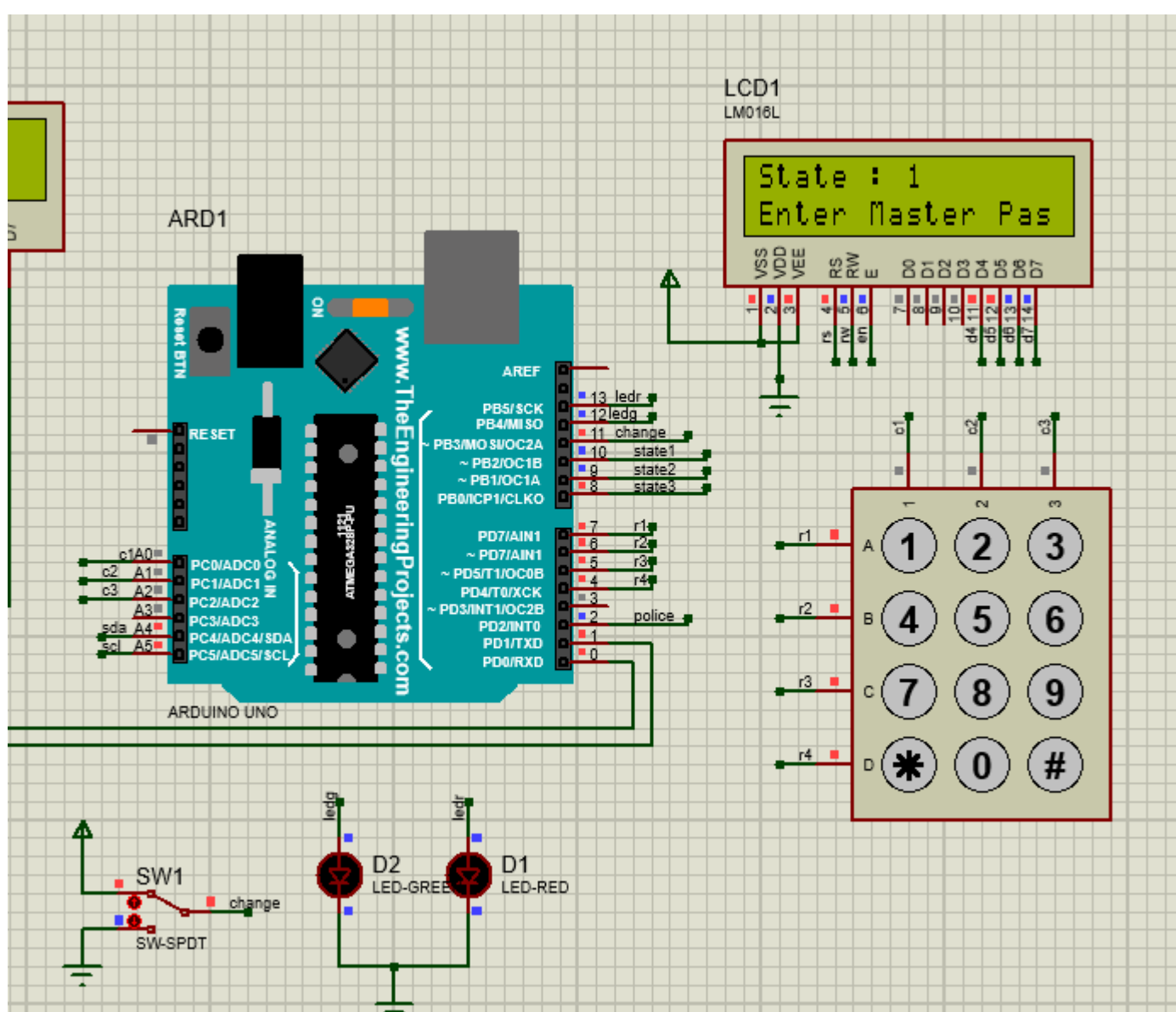
همان طور که میبینید، چون ورودی ولتاژ متغیر مسیر ۱، عدد بالاتری داشت، چراغ سبز A در دو ثانیه مانده به پایان فریز می شود.

اکنون state3 را امتحان می کنیم، در این حالت هرگاه دکمه police زده شود، باید بلافاصله رنگ چراغ ها ابتدا سریع زرد شده و جابجا شود. برای اینکار ابتدا کلید وضعیت را روی ۳ قرار داده و سپس می توانیم دکمه پلیس را امتحان کنیم:

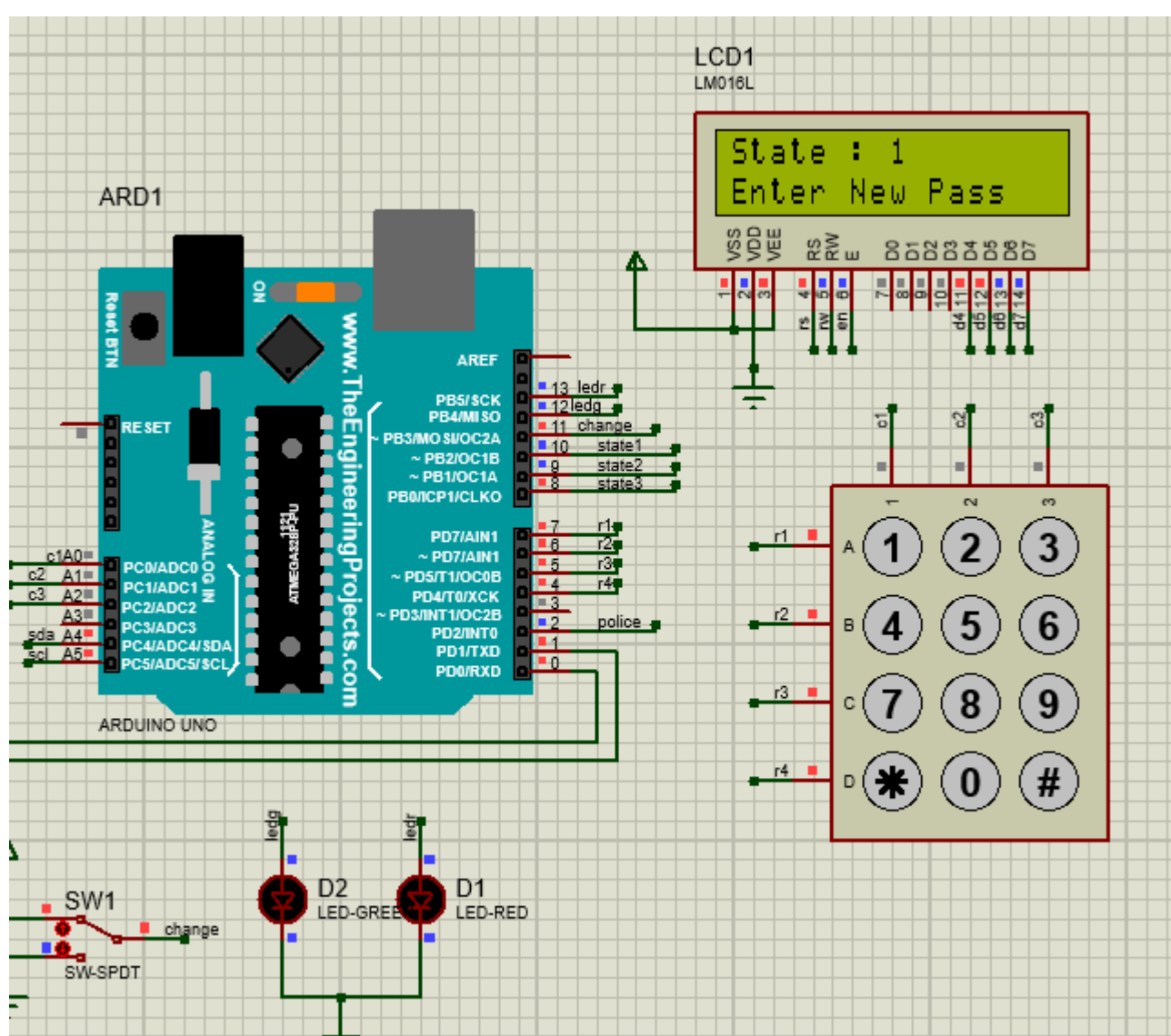


پس از زدن دکمه پلیس، مطابق دو عکس بالا، رنگ چراغ ها جابجا شده است.

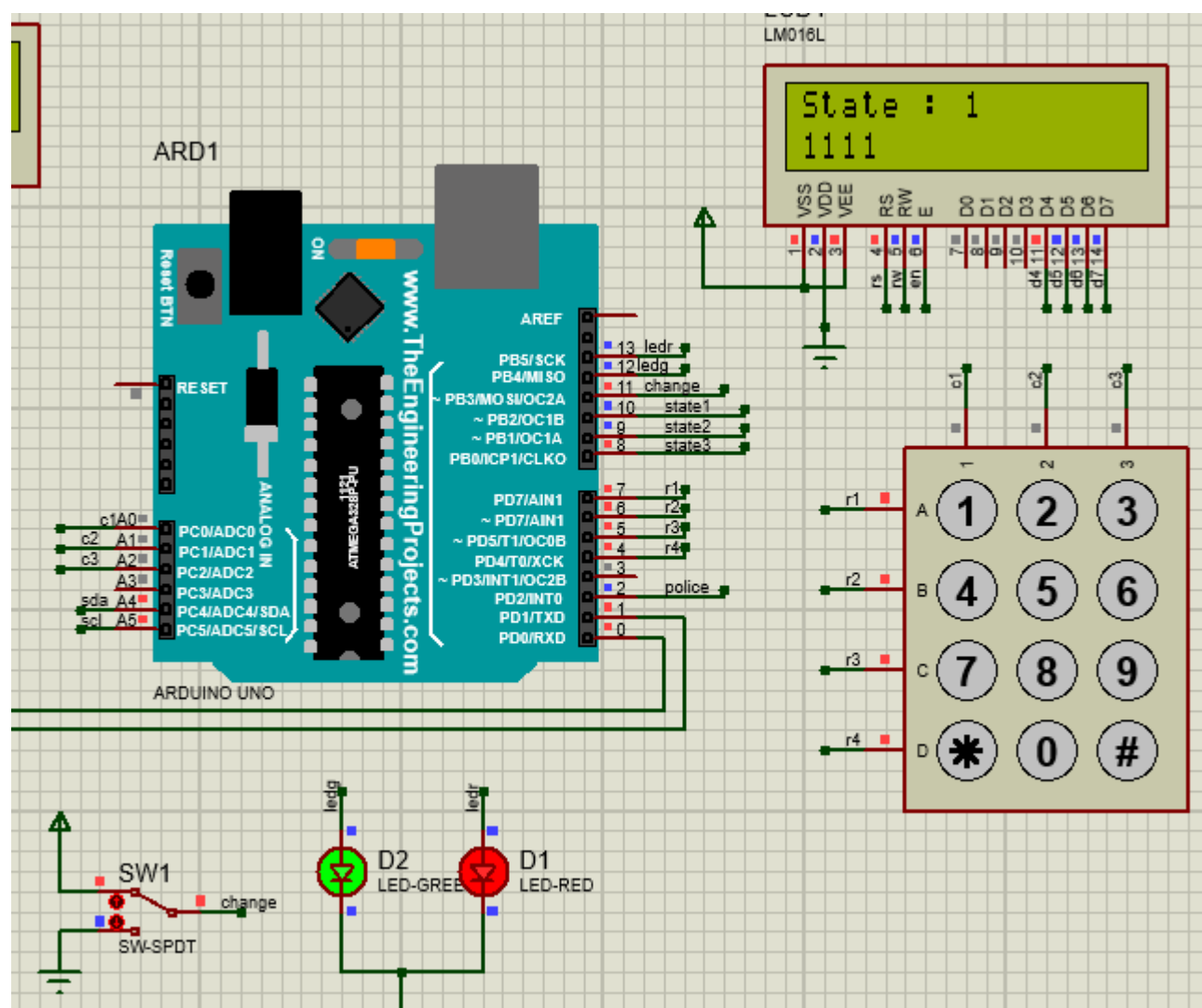
برای تغییر رمز keypad باید دکمه \* را بزنیم، سپس رمز اصلی را زده و پس از تایید صحت آن ، رمز جدید را وارد می کنیم تا ذخیره شود:



پس ازدن ستاره، متن رمز اصلی را وارد کنید ، به نمایش می آید، پس از وارد کردن رمز اصلی :



متن رمز جدید را وارد کنید به نمایش در می آید، اکنون یک رمز ۴ رقمی جدید مثلا ۱۱۱۱ را وارد می کنیم، در انتها شاهد عکس زیر خواهیم بود:



با ورود هر رقم، led قرمز چشمک میزند، در انتهای رقم چهارم، سه بار هردو led سبز و قرمز باهم چشمک خواهندزد که به معنی تعیین رمز جدید است.

## قسمت ۲ – بررسی کدها

چون دو برد آردوینو داریم، طبیعتاً دو کد برای آن‌ها خواهیم داشت. ابتدا کد آردوینو راست یعنی کنترل کننده keypad و وضعیت چراغ‌ها را نشان می‌دهیم:

```

2  #include <Keypad.h>
3  #include <string.h>
4  #include <stdio.h>
5  #include <Wire.h>
6  #include <LiquidCrystal_I2C.h>
7
8  LiquidCrystal_I2C lcd(0x20,16,2);
9
10 //const int rs = 13, en = 12, d4 = 11, d5 = 10, d6 = 9, d7 = 8;
11 //LiquidCrystal lcd(rs, en, d4, d5, d6, d7);
12
13 const byte ROWS = 4;
14 const byte COLS = 3;
15
16 char hexaKeys[ROWS][COLS] = {
17   {'1','2','3'},
18   {'4','5','6'},
19   {'7','8','9'},
20   {'*','0','#'}};
21 };
22 byte rowPins[ROWS] = {7, 6, 5, 4};
23 byte colPins[COLS] = {A0, A1, A2};
24

```

در خطوط ۲ تا ۶ کتابخانه‌های مورد نیاز برای keypad و lcd i2c و همچنین ارتباط سریال یعنی stdio، اضافه شده‌اند. در خط ۸ ماژول lcd i2c با کد 0x20 و ابعاد ۱۶\*۲ تعریف شده. در ادامه در

خطوط ۱۳ تا ۲۱، تعداد سطر و ستون keypad و آرایه ای از کاراکترهای آن تعیین شده است. محل پین های اتصال keypad نیز در خطوط ۲۱ و ۲۲ مشخص شده.

```
25 char inkey;
26 char pass[5];
27 char master[5]="2021";
28 int ledg = 12, ledr = 13, change_button = 11;
29 int change_en=0;
30 int state=1;
31 int i=0;
32 Keypad keypad34 = Keypad( makeKeymap(hexaKeys), rowPins, colPins, ROWS, COLS);
33 void setup() {
34     pinMode(ledg, OUTPUT);
35     pinMode(ledr, OUTPUT);
36     pinMode(change_button, INPUT);
37     pinMode(10, INPUT);
38     pinMode(9, INPUT);
39     pinMode(8, INPUT);
40     pinMode(2, INPUT);
41     attachInterrupt(digitalPinToInterrupt(2), police, RISING);
42     Serial.begin(9600);
43     lcd.init();
44     //lcd.begin(16, 2);
45     lcd.print("State : ");
46     lcd.setCursor(8, 0);
47     lcd.print(state);
48 }
```

خطوط ۲۵ تا ۳۱ متغیرهای لازم برنامه را در بردارد، مانند رشته پسورد، کاراکتر ورودی keypad، شماره led های قرمز و سبز و کلید وضعیت و شمارنده حلقه های for. در خط ۳۲ keypad دیفاین شده است. در خط ۳۴ تا ۴۰ وضعیت ورودی خروجی پین های برد راست را تعیین کرده ایم. همچنین در خط ۴۱ روی ورودی دکمه پلیس یک interrupt فعال کرده ایم تا اگر کامندی امد، بلافاصله اجرا شود. در خط ۴۲ ارتباط uart را با baudrate 9600 تعریف کردیم. خط ۴۳ تا ۴۷ تعیین وضعیت پیشفرض lcd راست است.

```
50 void loop()
51 {
52     change_en = 0;
53     if(digitalRead(change_button))
54         inputkey();
55
56     if(change_en)
57     {
58         state = digitalRead(10) + digitalRead(9)*2 + digitalRead(8)*3;
59         lcd.setCursor(8, 0);
60         lcd.print(state);
61         Serial.write(state);
62     }
63
64 }
```

لوپ اصلی برنامه در شکل بالا قرار دارد. در خط ۵۳ اگر دکمه فعال بودن keypad یک بود، تابع دریافت کلید از صفحه کلید را صدا می زنیم، حال نگاه می کنیم، اگر فلگ تغییر وضعیت روشن (این فلگ زمانی روشن هست که رمز ورودی درست وارد شده باشد) بود، مقدار state جدید را از سه ورودی کلید سه



حالت به دست می آوریم و آن را در خط های ۵۹ تا ۶۰ روی lcd چاپ می کنیم. در خط ۶۱ این وضعیت جدید را به اطلاع برد سمت چپ از طریق کابل سریال می رسانیم.

در ادامه تابع inputkey را داریم: وظیفه این تابع کنترل keypad دریافت ورودی های آن و تعیین و تکلیف حالت مدار است.

ابتدا چک می شود که کلید کنسل یا همان مربع زده نشده باشد. در ادامه در صورتی که کلید ستاره زده شده باشد، رویه ای که در صفحات قبل گفته شده بود طی می شود. سپس در صورتی که نه کلید ستاره بود و نه کلید مربع ، ۴ عدد ورودی را به منزله رمز نگاه می کند و چک می کنید که درست هست یا نه، اگر صحیح بود، فلگ تغییر وضعیت روشن خواهدشد.



```

97
98     for(i=0; i<4; i++)
99     {
100         inkey = keypad34.waitForKey();
101         if(inkey == '#')
102         {
103             lcd.setCursor(0, 1);
104             lcd.print("Cancelling...");
105             delay(100);
106             lcd.setCursor(0, 1);
107             lcd.print("                ");
108             return;
109         }
110         else pass[i] = inkey;
111         lcd.print(pass[i]);
112     }
113
114     if(!strcmp(pass, master))
115     {
116         lcd.setCursor(0, 1);
117         lcd.print("Enter New Pass");
118         delay(100);
119         lcd.setCursor(0, 1);
120         lcd.print("                ");
121         lcd.setCursor(0, 1);
122
123         for(i=0; i<4; i++)
124         {
125             inkey = keypad34.waitForKey();
126             if(inkey == '#')

```

```

66 void inputkey(void)
67 {
68     lcd.setCursor(0, 1);
69     for(i=0; i<4; i++)
70     {
71         inkey = keypad34.waitForKey();
72         if(inkey == '#')
73         {
74             lcd.setCursor(0, 1);
75             lcd.print("Cancelling...");
76             delay(50);
77             lcd.setCursor(0, 1);
78             lcd.print("                ");
79             return;
80         }
81     }
82     else if(inkey == '*')
83     {
84         digitalWrite(ledg, 1);
85         digitalWrite(ledr, 1);
86         delay(50);
87         digitalWrite(ledg, 0);
88         digitalWrite(ledr, 0);
89
90         lcd.setCursor(0, 1);
91         lcd.print("Enter Master Pas");
92         delay(100);
93         lcd.setCursor(0, 1);
94         lcd.print("                ");
95
96         lcd.setCursor(0, 1);

```

```

127 {
128     lcd.setCursor(0, 1);
129     lcd.print("Cancelling...");
130     delay(50);
131     lcd.setCursor(0, 1);
132     lcd.print("                ");
133     return;
134 }
135 else pass[i] = inkey;
136 lcd.print(pass[i]);
137 if(i==3)
138 {
139     digitalWrite(ledg, 1);
140     digitalWrite(ledr, 1);
141     delay(50);
142     digitalWrite(ledg, 0);
143     digitalWrite(ledr, 0);
144     delay(50);
145     digitalWrite(ledg, 1);
146     digitalWrite(ledr, 1);
147     delay(50);
148     digitalWrite(ledg, 0);
149     digitalWrite(ledr, 0);
150     delay(50);
151     digitalWrite(ledg, 1);
152     digitalWrite(ledr, 1);
153     delay(50);
154     digitalWrite(ledg, 0);
155     digitalWrite(ledr, 0);
156 }

```

```

156     }
157     else
158     {
159         digitalWrite(ledr, 1);
160         delay(50);
161         digitalWrite(ledr, 0);
162     }
163 }
164 strcpy(master, pass);
165 lcd.setCursor(0, 1);
166 lcd.print("Password Changed");
167 delay(50);
168 lcd.setCursor(0, 1);
169 lcd.print("                ");
170 return;
171 }
172 else
173 {
174     lcd.setCursor(0, 1);
175     lcd.print("Wrong Password");
176     delay(50);
177     lcd.setCursor(0, 1);
178     lcd.print("                ");
179     return;
180 }
181 }
182 else
183 {
184     pass[i] = inkey;
185     lcd.print("*");
186 }

```

```

186 }
187 }
188 if(!strcmp(master, pass))
189 {
190     lcd.setCursor(0, 1);
191     lcd.print("Ready for change");
192     digitalWrite(ledg, 1);
193     delay(50);
194     digitalWrite(ledg, 0);
195     delay(50);
196     digitalWrite(ledg, 1);
197     delay(50);
198     digitalWrite(ledg, 0);
199     delay(50);
200     digitalWrite(ledg, 1);
201     delay(50);
202     digitalWrite(ledg, 0);
203     delay(50);
204     lcd.setCursor(0, 1);
205     lcd.print("                ");
206     change_en = 1;
207     return;
208 }
209 else
210 {
211     lcd.setCursor(0, 1);
212     lcd.print("Invalid Password");
213     digitalWrite(ledr, 1);
214     delay(50);
215     digitalWrite(ledr, 0);
216     delay(50);

```

```

216     delay(50);
217     digitalWrite(ledr, 1);
218     delay(50);
219     digitalWrite(ledr, 0);
220     delay(50);
221     digitalWrite(ledr, 1);
222     delay(50);
223     digitalWrite(ledr, 0);
224     delay(50);
225     lcd.setCursor(0, 1);
226     lcd.print("                ");
227     change_en = 0;
228     return;
229 }
230 }
231
232 void police(void)
233 {
234     if(state == 3)
235     {
236         Serial.write(33);
237     }
238 }

```