

به نام خدا

دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر



آزمایشگاه ریزپردازنده

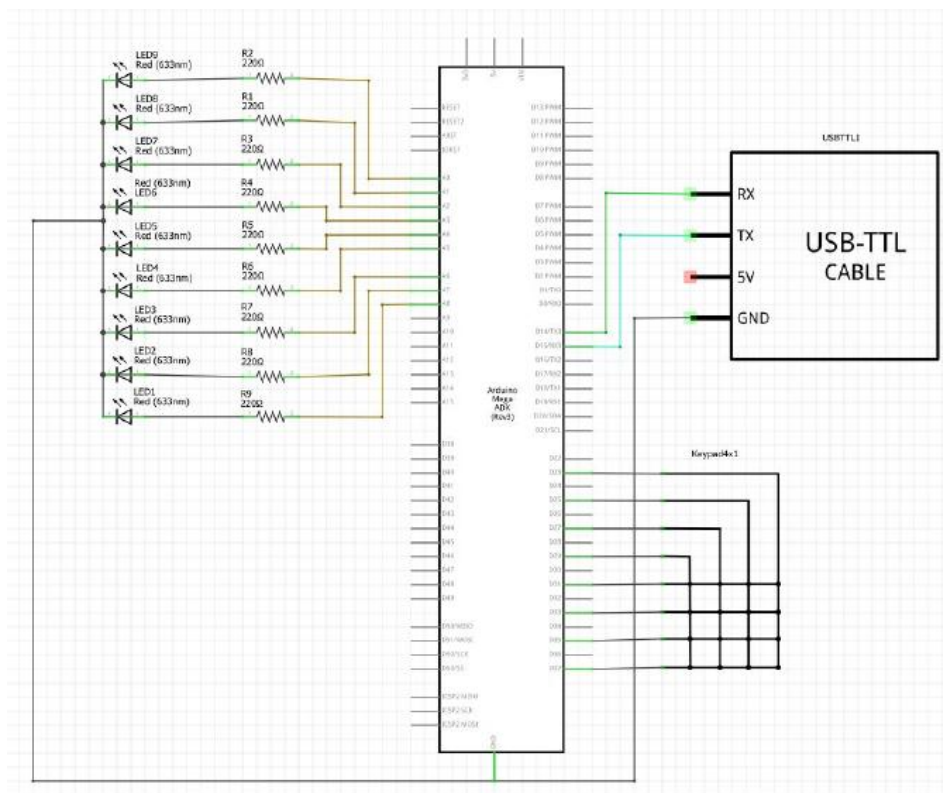
گزارش کار آزمایش ۲

استاد درس: مهندس معصوم زاده

مهدی رحمانی / ۹۷۳۱۷۰۱

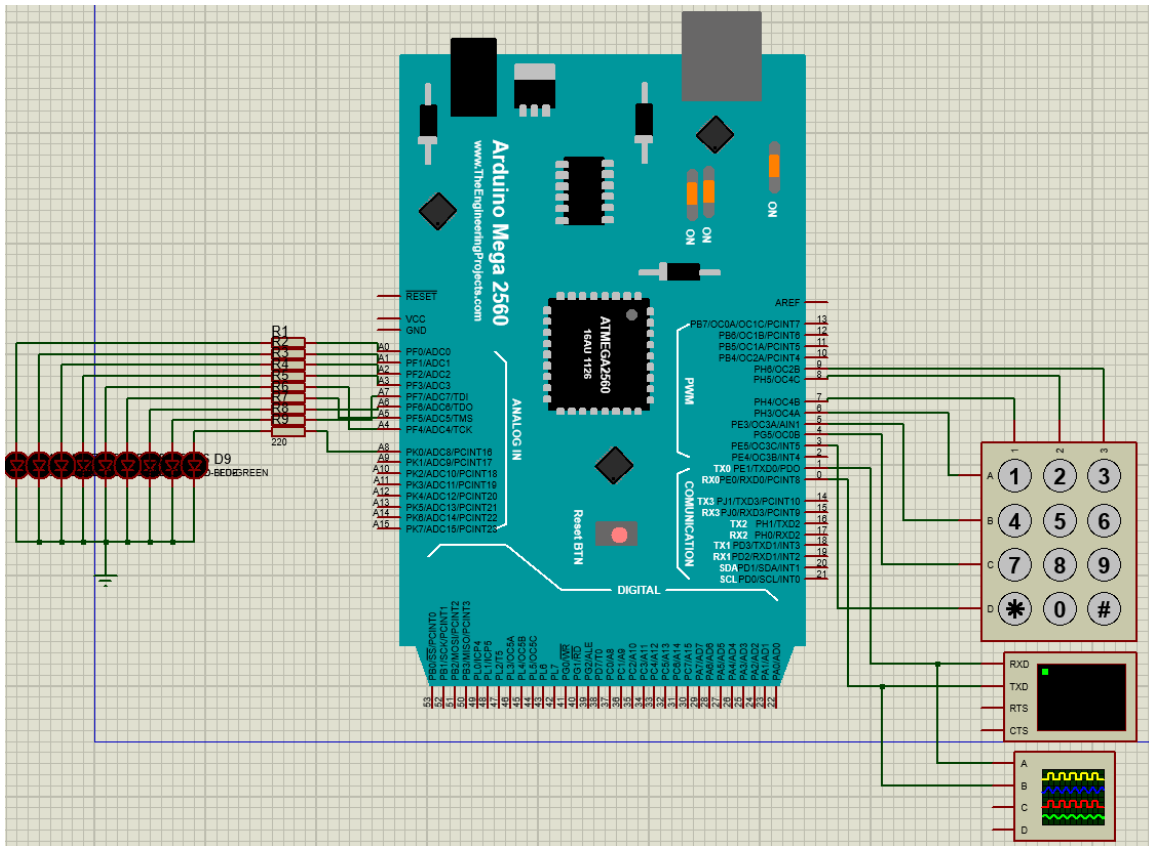
محمد امین رضائی / ۹۷۳۱۰۲۴

ابتدا مدار زیر را بررسی میکنیم:

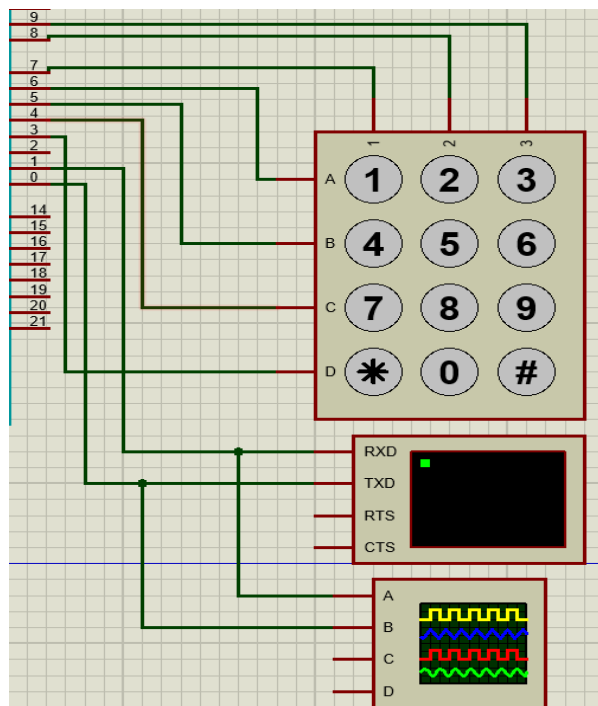


برای بستن مدار بالا در پروتئوس، به یک صفحه کلید keypad phone، یک برد Arduino2560، ۹ عدد مقاومت، ۹ عدد ال ای دی به رنگ های قرمز و سبز و آبی و یک اوسیلوسکپ و یک ترمینال مجازی احتیاج داریم. طبق دستور کار قطعات گفته شده را به پورت های مورد نظر در برد وصل می کنیم.

حال مداری که در پروتئوس کشیدیم را میبینیم:



حال نگاه دقیق تری به بخش سمت راست برد می اندازیم:



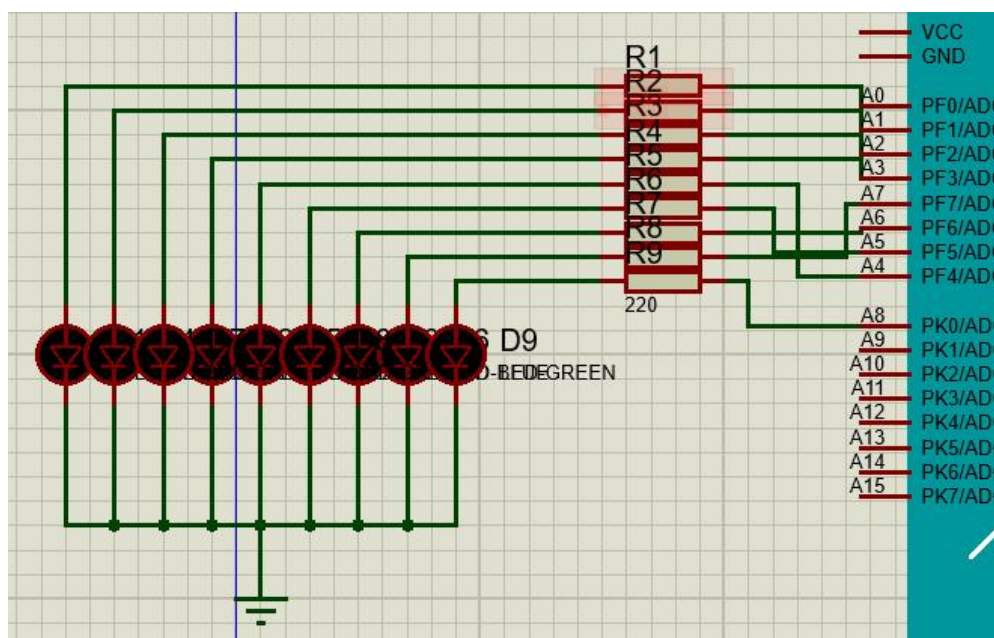
۳ خروجی یک و دو و سه، که مربوط به بالای صفحه کلید میباشد را به پورت ۹ و ۸ و ۷ قسمت

PWM مدار و خروجی A,B,C,D را به پورت های ۳ و ۴ و ۵ و ۶ وصل میکنیم.

دو خروجی RXD و TXD ترینال و خروجی های A,B اوسیلوسکوپ را هم به پورت های TX 0

و RX 0 وصل میکنیم.

حال نگاهی به سمت چپ مدار می اندازیم:



مشاهده میشود که هر پورت خروجی به کدام ال ای دی وصل میباشد.

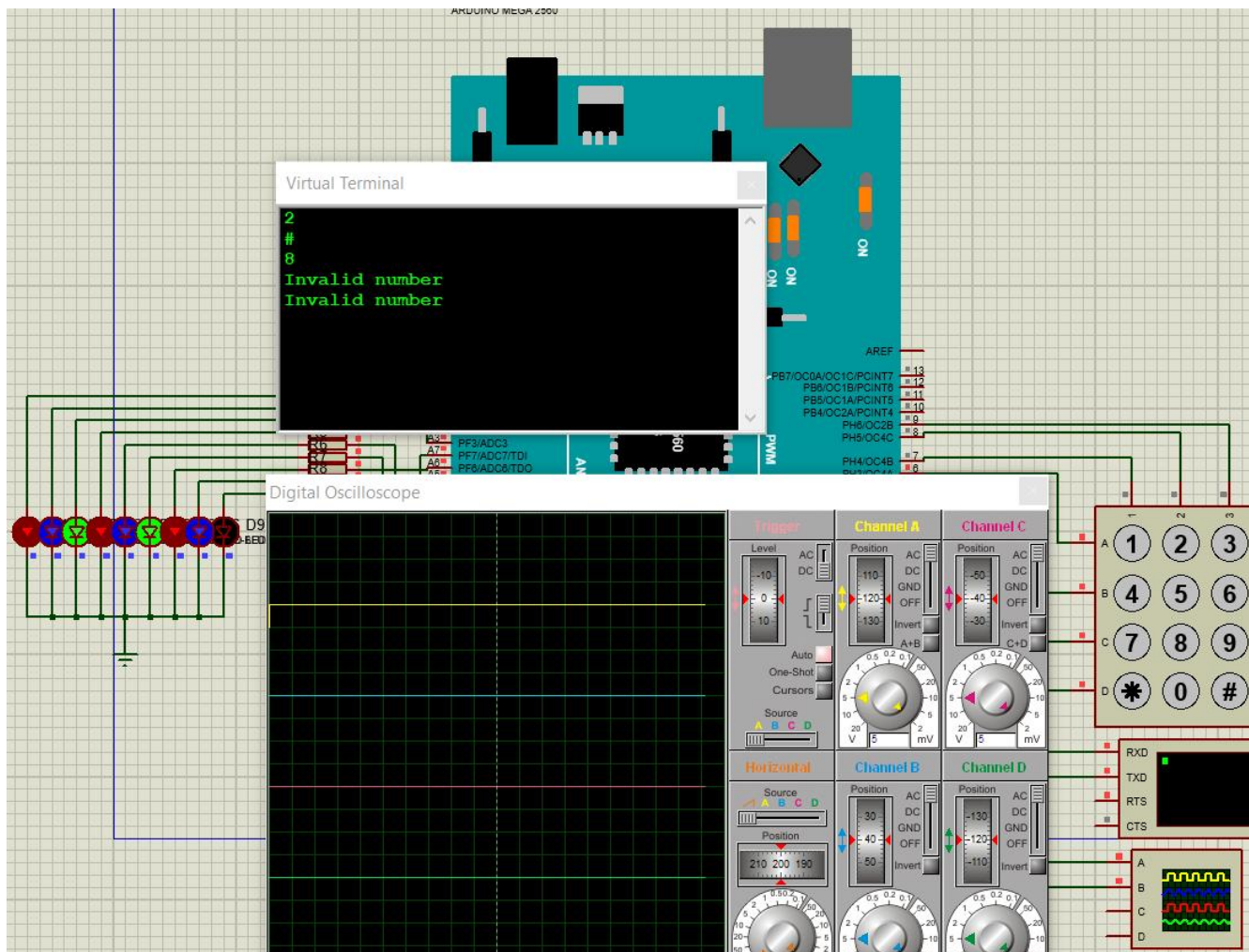
حال شبیه سازی را انجام داده و مشاهده میکنیم که با هر بار زدن کلید های ۱ تا ۹ و یا وارد کردن

عددی بین ۱ تا ۹ در ترینال، به همان تعداد از چپ به راست ال ای دی ها روشن میشوند.

اگر # را فشار دهیم یا در ترینال وارد کنیم، کلیه ال ای دی ها خاموش میشوند. به غیر از دو حالت

بالا، اگر چیز دیگری وارد ترینال شود، خروجی invalid number نمایش داده میشود.

حال نمونه ای از شبیه سازی میبینیم:



مشاهده شد که با هر بار وارد شدن یک عدد بین یک تا ۹، سیگنال هایی نوسان می کنند.
با هر بار زدن کلید یا وارد کردن عدد از طریق کیبورد، عدد در ترمینال نمایش داده می شود.

حال به سراغ کد برنامه میرویم:

```
1 #include <Keypad.h>
2
3 const int ROW_NUM = 4; //four rows
4 const int COLUMN_NUM = 3; //three columns
5 static const uint8_t ANALOG_PINS[] = {A0,A1,A2,A3,A4,A5,A6,A7,A8};
6 //DEFINE KEYPAD BUTTONS
7 char keys[ROW_NUM][COLUMN_NUM] = {
8     {'1','2','3'},
9     {'4','5','6'},
10    {'7','8','9'},
11    {'*','0','#'}
12 };
13 //DEFINE KEYPAD CONNECTED PORTS
14 byte pin_rows[ROW_NUM] = {6, 5, 4, 3}; //connect to the row pinouts of the keypad
15 byte pin_column[COLUMN_NUM] = {7, 8, 9}; //connect to the column pinouts of the keypad
16
17 Keypad keypad = Keypad( makeKeymap(keys), pin_rows, pin_column, ROW_NUM, COLUMN_NUM );
```

در این بخش از کد، ابتدا تعداد سطر ها و ستون های صفحه کلید ماتریسی ما مشخص شده و به صورت یک آرایه دوبعدی صفحه کلید ساخته میشود. در خط ۵ برای راحتی کار با خروجی ها، خروجی ها به شکل ۹ خانه یک آرایه در می آیند.

همچنین در خط ۱۴ و ۱۵، پورت های وصل شده به صفحه کلید ما، به صورت دو آرایه تک بعدی که حاوی شماره پین های وصل شده به صفحه کلید هستند داده میشود.

در خط ۱۷ هم یک شیء صفحه کلید به نام keypad با مشخصات ساخته شده در خطوط قبل ترش ساخته میشود.

```

18
19 void setup() {
20     //BEGIN SERIAL
21     Serial.begin(9600);
22     //DEFINE OUTPUT PINS
23     pinMode(A0, OUTPUT);
24     pinMode(A1, OUTPUT);
25     pinMode(A2, OUTPUT);
26     pinMode(A3, OUTPUT);
27     pinMode(A4, OUTPUT);
28     pinMode(A5, OUTPUT);
29     pinMode(A6, OUTPUT);
30     pinMode(A7, OUTPUT);
31     pinMode(A8, OUTPUT);
32 }

```

در این بخش هم پین های خروجی مشخص شده و در خط ۲۱ برای استفاده از ترمینال مجازی، از دستور بالا استفاده میکنیم و نرخ انتقال داده را ۹۶۰۰ بیت بر ثانیه تعیین میکنیم.

```

51 void check(char key) {
52     //Control EMPTY TERMINAL
53     if(!key)
54         return;
55     //PRINT INVALIDs
56     if((key<'1' || key>'9') && key!='#'){
57         Serial.println("Invalid number");
58         return;
59     }
60     //PRINT KEY NUMBER
61     Serial.println(key);
62     //TURN OFF LEDs
63     for (int i=0; i<9; i++)
64         digitalWrite(ANALOG_PINS[i], LOW);
65     //TURN ON LEDs
66     for (char i='1'; i<=key; i++){
67         digitalWrite(ANALOG_PINS[i-49], HIGH);
68         delay(100);
69     }
70 }

```

در این بخش یک تابع مینویسیم تا در بخش loop از آن استفاده کنیم. اسم تابع check میباشد.

ورودی این تابع کاراکتر کلید میباشد که در بخش لوپ گرفته میشود.

ابتدا در خطوط ۵۳ و ۵۴ چک میشود که اگر ورودی خالی بود تابع کاری نکند.

در خطوط ۵۶ تا ۵۹ چک میشود که اگر ورودی ما چیزی غیر از عدد ۱ تا ۹ یا # باشد، در

ترمینال پیغام invalid number چاپ شود و تابع کارش تمام شود.

خط ۶۰ هم عدد وارد شده بین ۱ تا ۹ و یا # را در ترمینال چاپ میکند. خط ۶۳ به بعد هم می

آید ال ای دی هارا خاموش کرده و به ترتیب از سمت چپ آنها را با تاخیر ۱۰۰ میلی ثانیه

روشن میکند.

```
34 void loop() {
35
36     //GET CHARACTER OF KEY PRESSED
37     char key = keypad.getKey();
38     //CHECK KEY AND TURN ON OR OFF LED
39     check(key);
40
41     //SERIAL AVAILABILITY CHECK
42     if(Serial.available()) {
43         //GET CHARACTER OF KEY PRESSED IN TERMINAL
44         key = Serial.read();
45         //CHECK KEY AND TURN ON OR OFF LED
46         check(key);
47     }
48     .....
49 }
```

این بخش از کد هم مربوط به loop می باشد که در آن ابتدا کاراکتر key از صفحه کلید گرفته

میشود و سپس وارد تابع check می شود. حال اگر سریال موجود بود، کاراکتر key از سریال یا

همان ترمینال خوانده شده و به تابع check داده می شود.

پس ما با کمک تابع check تمام کارهای لازم را ساماندهی کردیم.

یاعلی

پایان