



# گزارش آزمایش شماره ۱

## مشخصات برد های آردوئینو

		2560	due	uno
Frequency		16 MHz	84 MHz	16 MHz
I/O inputs	digital	54	54	6
	analog	16	12	6
voltage	Operating voltage	5V	3.3V	5V
	Input Voltage (recommended)	7-12V	7-12V	7-12V
	Input Voltage (limited)	6-20V	6-16V	6-20V

## قانون اهم و روابط بین اختلاف پتانسیل و شدت جریان و مقاومت

$$R = \frac{V}{I} \quad (\Omega = \frac{V}{A})$$

نسبت اختلاف ولتاژ دو سر یک جسم (مثل سیم) به جریانی که از آن میگذرد مقدار ثابتی است که به آن مقاومت میگویند.

واحد آن ولت بر آمپر یا همان اهم میباشد.

با توجه به رابطه ی بالا میتوان گفت مقاومت با ولتاژ دو سر آن رابطه مستقیم و همچنین با جریان گذرنده از آن رابطه عکس دارد.



## نحوه خواندن کد رنگی مقاومت

همانطور که در شکل پایین مشاهده میشود باید رنگ ها را از سمت چپ به راست بررسی کنیم. دو نوار سمت چپ(نوار مربوط به تلورانس باید در سمت راست قرار داشته باشد) یک عدد دورقمی را میسازند. رقم سوم عددیست که باید عدد ۱۰ به توان آن برسد و در عدد ساخته شده توسط دو نوار سمت چپ ضرب شود تا مقاومت کل بدست آید.

تمام رنگ های موجود در جدول زیر نمایش داده شده اند:

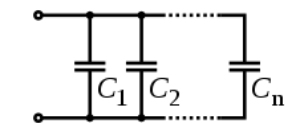
$$R = AB \times 10^C \pm D\%$$

4-Band-Code					
2%, 5%, 10%			560kΩ ± 5%		
COLOR	1st BAND	2nd BAND	3rd BAND	MULTIPLIER	TOLERANCE
Black	0	0	0	1Ω	
Brown	1	1	1	10Ω	± 1% (F)
Red	2	2	2	100Ω	± 2% (G)
Orange	3	3	3	1KΩ	
Yellow	4	4	4	10KΩ	
Green	5	5	5	100KΩ	± 0.5% (D)
Blue	6	6	6	1MΩ	± 0.25% (C)
Violet	7	7	7	10MΩ	± 0.10% (B)
Grey	8	8	8		± 0.05%
White	9	9	9		
Gold				0.1	± 5% (J)
Silver				0.01	± 10% (K)

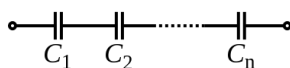


# گزارش آزمایش شماره ۱

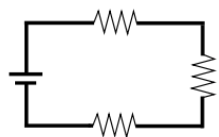
## نحوه سری و موازی بستن مقاومت و خازن



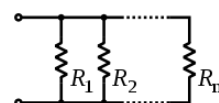
$$C_{eq} = \sum_{i=0}^n C_i$$



$$C_{eq} = \sum_{i=0}^n \frac{1}{C_i}$$



$$R_{eq} = \sum_{i=0}^n R_i$$



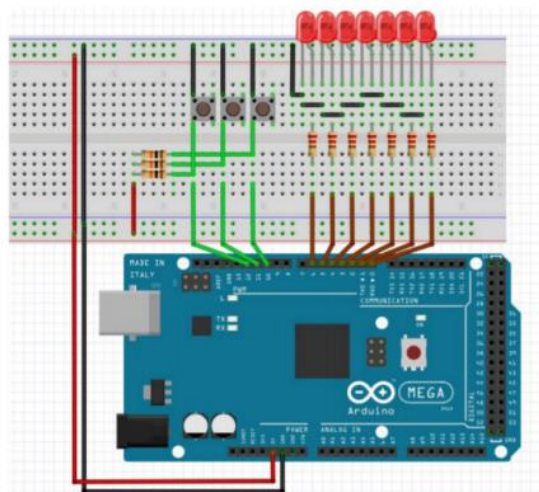
$$R_{eq} = \sum_{i=0}^n \frac{1}{R_i}$$

## نحوه استفاده و عملکرد دستورات

- `digitalWrite()`: برای مقداردهی به یک پین دیجیتال است. (HIGH/LOW)
- `digitalRead()`: برای خواندن مقدار یک پین دیجیتال استفاده میشود. (HIGH/LOW)
- `analogWrite()`: برای مقداردهی پیوسته به یک پین آنالوگ است.
- `analogRead()`: برای خواندن مقدار یک پین آنالوگ استفاده میگردد.

## آزمایش شماره یک (۷ چراغ چشمک زن و ۳ دکمه)



هدف ما پیاده سازی مدار زیر در نرم افزار proteus میباشد.



همینطور که واضح است ما به ۷ عدد LED (output)، ۳ عدد دکمه (input) و ۱۰ عدد مقاومت نیاز داریم. دو نوع مقاومت نیاز خواهیم داشت:

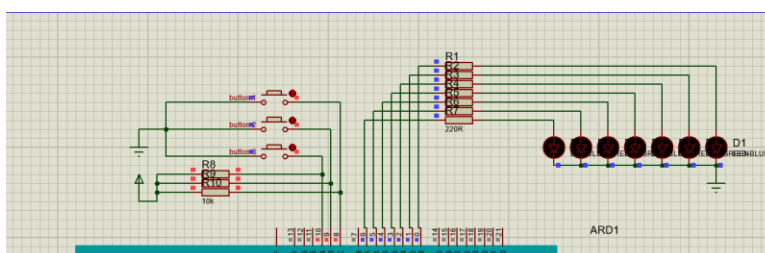


# گزارش آزمایش شماره ۱

مربوط به دکمه ها	مربوط به LEDها	
		شکل
10x10 <sup>3</sup> kΩ	22x10 <sup>1</sup> Ω	مقاومت
3	7	تعداد

پورت های ۰ تا ۶ را به ۷ پورت ورودی که به قرار است به دیود ها وصل کنیم و همچنین پورت های ۸ تا ۱۰ را به کلید ها متصل شوند اختصاص می‌دهیم.

به همین ترتیب در proteus مدار شکل موجود در دستور کار را پیاده سازی می‌کنیم:



برای برنامه نویسی روی برد مورد نظر، کد زیر را داخل Arduino می‌نویسیم:

```
int button_1 = 1; //the value of button#1
int button_2 = 1; //the value of button#2
int button_3 = 1; //the value of button#3
int delay_period = 100; //delay period (ms)
void setup() {
    // initialize LEDs
    pinMode(0, OUTPUT); //LED0
    pinMode(1, OUTPUT); //LED1
    pinMode(2, OUTPUT); //LED2
    pinMode(3, OUTPUT); //LED3
    pinMode(4, OUTPUT); //LED4
    pinMode(5, OUTPUT); //LED5
    pinMode(6, OUTPUT); //LED6
    // initialize Buttons
    pinMode(8, INPUT); // key #1
    pinMode(9, INPUT); // key #2
    pinMode(10, INPUT); // key #3
}
void loop() {
    button_1 = digitalRead(8);
    button_2 = digitalRead(9);
    button_3 = digitalRead(10);
    if (button_3 == LOW) { // button#3 has been pressed
        digitalWrite(0, LOW);
        digitalWrite(1, LOW);
        digitalWrite(2, LOW);
        digitalWrite(3, LOW);
        digitalWrite(4, LOW);
        digitalWrite(5, LOW);
        digitalWrite(6, LOW);
    } else if (button_2 == LOW) { // button#2 has been pressed
        // activate LED #1
        digitalWrite(0, HIGH);
        delay(delay_period);
    }
}
```



# گزارش آزمایش شماره ۱

```
//activate LED #2
digitalWrite(1, HIGH);
delay(delay_period);
.
.
.

//activate LED #7
digitalWrite(6, HIGH);
delay(delay_period);
}else if ( button_1 == LOW){//button#1 has been pressed
//activate LED #7
digitalWrite(6, HIGH);
delay(delay_period);
//activate LED #6
digitalWrite(5, HIGH);
delay(delay_period);
.
.
.

//activate LED #1
digitalWrite(0, HIGH);
delay(delay_period);
}
}
```

در تابع setup تمام پورت‌ها مشخص میشوند. بدنه تابع loop دارای ۳ if می‌باشد که هر کدام مربوط به کارهایی است که هریک از دکمه‌ها انجام می‌دهند. فرم if بندی‌ها بطوری است که اولویت ابتدا با دکمه سوم سپس با دکمه دوم و در آخر با اولین دکمه باشد؛ یعنی اگر مثلاً تمام دکمه‌ها همزمان فشار داده شوند، برنامه اولویت را به دکمه ۳ می‌دهد (تمام دیودها خاموش باقی می‌مانند).

- در دستورات مربوط به دکمه سوم تمامی ۷ پورت را در حالت LOW قرار می‌دهیم که دیودها خاموش شوند.
- برای دکمه دوم از سمت راست دکمه‌ها را یکی پس از دیگری روشن می‌کنیم تا در پایان تمامی دیودها روشن باشند، بعد از روشن کردن هر کدام از دیودها باید زمان مشخصی صبر کنیم (بصورت پیشفرض این مقدار برابر ۱۰۰ میلی ثانیه در متغیر delay\_period ذخیره شده است که میتوان در ابتدای برنامه مقدار آن را تغییر داد).
- نحوه کار اولین دکمه بسیار شبیه دکمه دوم می‌باشد با این تفاوت که ترتیب خاموش-روشن‌ها بر خلاف حالت قبل از چپ به راست می‌باشد.

بعد از استفاده از فایل hex تولید شده توسط کد نوشته شده در برنامه proteus، میتوان خروجی‌های دلخواه را با فشردن کلیدها مشاهده کرد.