ساخت اعداد واقعا تصادفی

تاریخ انتشار۲۳ اسفند, ۱۳۹۹ توسط محمد جواد رشیدیانفر

سلام به همه شما مایکروالکامی های عزیز. امروز قصد داریم با هم دیگه روش تولید اعداد تصادفی رو بررسی کنیم. شاید برای شما هم پیش اومده باشه در یک پروژه یا کاری نیاز به رمز گذاری و تولید یک الگوریتم باشید. اون موقع هست که گذرتون به اعداد تصادفی خواهد افتاد. اما یک سوال مهم مطرح میشه! اینکه آیا این اعداد واقعا تصادفی هستند؟ یا اگر شما روش عملکرد تابع ساخت اعداد تصادفی رو بلد باشید میتونید اعداد بعدی و قبلی رو حدس بزنید؟ پس تا انتهای مطلب با ما همراه باشید.

برای بررسی و ساخت اعداد تصادفی در این مطلب از محیط برنامه نویسی Arduino IDE و یک بورد آردوینو استفاده شده است. همچنین برای نمایش اعداد بوجود اومده از سریال مانیتور آردوینو استفاده خواهیم کرد.

تابع random()

همانطور که از اسم تابع هم مشخص است، از این تابع برای ساخت اعداد تصادفی بین یک بازه مشخص در آردوینو استفاده می‌شود. منظور از بازه مشخص این است که ما خودمون اون رو در آرگومان تابع مشخص خواهیم کرد. از این تابع به 2 صورت زیر استفاده می‌شود:

حالت اول:

```
random(min, max);
```

با استفاده از مقادیر min و max تعیین میکنیم که مقدار خروجی تابع رندوم حداقل بین چند و حداکثر چند باشد. اعداد جایگزین min و max باید از نوع صحیح یا int باشند.

مثال:

```
long Random_Number;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Random_Number = random(0, 300);
  Serial.println(Random_Number);
```

```
  delay(1000);
}
```

مشاهده خروجی:

چون تابع random() در حلقه loop استفاده شده است، هر 1 ثانیه یک عدد تصادفی تولید خواهد شد. که تعدادی از کد های تولید شده را زیر مشاهده خواهید کرد. در صورتی که آردوینو را ریست کنید و مجدد راه اندازی شود، اعداد تولید شده مجددا تکرار خواهد شد.

```
7
49
173
158
130
272
144
278
23
109
240
65
192
242
87
203
.
.
.
```

حالت دوم:

```
random(max);
```

با استفاده از مقدار max تعیین میکنیم که مقدار خروجی تابع رندوم حداکثر چند باشد. در واقع عدد تولید شده بین صفر تا max-1 خواهد بود. مثلا اگر مقدار max رو 200 انتخاب کردیم، عدد تولید شده تصادفی یک عدد بین 0 تا 199 خواهد بود.

مثال:

```
long Random_Number;

void setup()
{
  Serial.begin(9600);
}

void loop()
{
  Random_Number = random(300);
  Serial.println(Random_Number);
  delay(1000);
}
```

مشاهده خروجی:

چون تابع random() در حلقه loop استفاده شده است، هر 1 ثانیه یک عدد تصادفی تولید خواهد شد. که تعدادی از کد های تولید شده را زیر مشاهده خواهید کرد. در صورتی که آردوینو را ریست کنید و مجدد راه اندازی شود، اعداد تولید شده مجددا تکرار خواهد شد.

```
7
49
173
158
130
272
144
278
23
109
240
65
192
242
87
203
.
.
.
```

همانطور که مشاهده کردید، خروجی تولید شده در هر دو حالت اول و دوم مشابه هم می‌باشد. این نکته را در نظر داشته باشید که نوع متغییر تعریف شده برای ذخیره عدد تصادفی تولید شده، باید متناسب به مقدار عدد max تعریف شده در آرگومان تابع باشد. بهتر است از متغییر نوع long استفاده شود.

تابع randomSeed()

تابع random()، در واقع تولید کننده اعداد تصادفی بوده و تابع randomSeed() تعیین کننده یا تنظیم کننده مقدار اولیه عدد تصادفی می‌باشد. با استفاده از این تابع میتوانیم اعداد تصادفی واقعی تری را داشته باشیم. یعنی تکراری نباشند و به نوعی کاملا تصادفی باشند. در آرگومان این تابع شما میتوانید یک عدد را مشخص کنید. مثلا با تابع random() یک عدد

تصادفی بسازید و بعنوان ورودی تابع randomSeed() در نظر بگیریم. یا شما میتوانید بجای این کار که مشکلات قبلی را در بر داشت (بعد از ریست شدن، اعداد تولید شده مشابه قبل از ریست شدن بودند) از پورت ADC میکرو استفاده کنید.

تابع randomSeed() مقدار اولیه عدد تصادفی تولید شده توسط تابع random() را ریست میکند. این مورد درست است که عدد تولید شده توسط تابع random() تصادفی است، اما دامنه اعداد تولید شده قابل حدس و پیش بینی است. لذا این مورد یک عیب میتواند حساب شود. پس چاره چیست؟

برای ساخت یک عدد تصادفی غیر قابل حدس، باید در هر بار اجرای برنامه مقدار اولیه تابع را با یک مقدار کاملا تصادفی، ریست کنیم. برای ساخت این مقدار کاملا تصادفی میتوانیم از پورت ADC میکروکنترلر استفاده کنیم. اگر یک پایه از پورت را بصورت آزاد رها کنیم، نویز ها و عوامل محیطی اطراف روی آن تاثیر خواهند گذاشت. این عوامل یا نویز ها میتوانند نویز ناشی از تریگ شدن یک موتور، روشن شدن یک لامپ، نویز ناشی از تلفن همراه و... باشد.

برای استفاده از این تابع باید بصورت زیر عمل کرد:

```
randomSeed(seed);
```

در این مثال از پایه A0 آردوینو برای خواندن مقدار آنالوگ استفاده شده است. همچنین این پایه بصورت آزاد می‌باشد و به جایی وصل نیست. در ابتدای برنامه اصلی، از دستور بالا استفاده خواهیم کرد. و مقدار خوانده شده توسط پایه A0، را بعنوان ورودی این تابع در نظر خواهیم گرفت.

```
long Random_Number;

void setup()
{
  Serial.begin(9600);
  randomSeed(analogRead(A0));
}

void loop()
{
  Random_Number = random(0, 300);
```

```
  Serial.println(Random_Number);
  delay(1000);
}
```

مشاهده نتیجه:

```
48
58
150
186
120
27
94
38
271
7
108
236
192
95
258
86
.
.
.
```

از آنجا که مقدار min و max تابع random() در این کد مشابه بخش اول مطلب است (random(0, 300))، اگر نتایج بدست آمده در حالت فعلی را با نتایج بدست آمده توسط حالت اول یا دوم بخش ابتدایی مطلب، جایی که تابع random() مطرح شد، را بررسی کنید مشاهده خواهید کرد که نتایج بدست آمده در این حالت با هم برابر نیستند. پس نتیجه میگیریم نتایج بدست آمده در این حالت تصادفی تر و غیر قابل حدس می‌باشند.

حال اگر آردوینو ریست شود و دوباره راه اندازی شود، باز هم نتایج بعد از ریست شدن با قبل از ریست شدن با هم تفاوت خواهند داشت؟

مشاهده نتیجه قبل از ریست کردن آردوینو:

```
48
58
150
186
120
27
94
38
271
7
108
236
192
95
258
86
 .
 .
 .
```

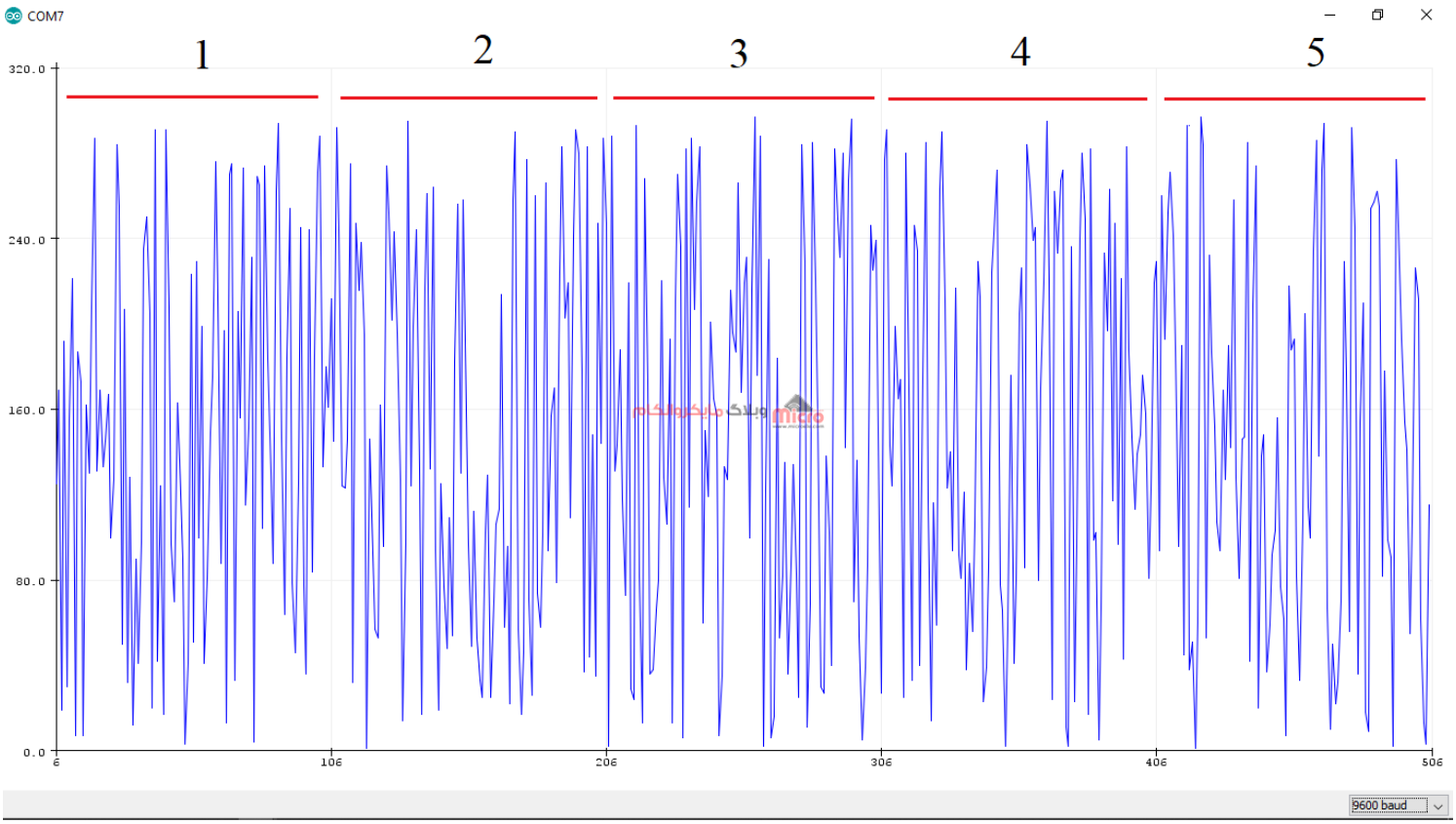
مشاهده نتیجه بعد از ریست کردن آردوینو:

```
237
40
181
41
```

```
172
189
135
151
46
75
218
33
295
82
260
62
45
251
.
.
.
```

پس جواب سوال بالا به روشنی مشخص خواهد شد. بله، بعد از راه اندازی مجدد، اعداد بوجود آمده با حالت قبل از راه اندازی مجدد برابر نخواهند بود. چرا که در ابتدای راه اندازی مقدار پایه A0 خوانده می‌شود و بعنوان ورودی تابع randomSeed() در نظر گرفته می‌شود. و این مقدار در هر لحظه متفاوت خواهد بود.

در تصویر زیر طی 5 مرحله آردوینو ریست شده و در هر مرحله میتوانید خروجی را مشاهده نمایید.



مشاهده خروجی اعداد تولید شده

امیدوارم از این آموزش کمال بهره را برده باشید. در صورتی که هرگونه نظر یا سوالی درباره این آموزش داشتید، لطفا اون رو در انتهای همین صفحه در قسمت دیدگاه ها قرار بدید. در کوتاه ترین زمان ممکن به اون ها پاسخ خواهم داد. اگر این مطلب براتون مفید بود، اون رو حتما با دوستانتون به اشتراک بگذارید. همینطور میتونید اون رو پس از اجرای عملی توی اینستاگرام با هشتگ #microelecom به اشتراک بگذارید و پیج مایکروالکام (@microelecom) رو هم منشن کنید.