

عنوان: تمرین 1 (مدار چشمک زن در آردوینو)

چکیده:

در این تمرین یک مدار چشمک زن ساده با کمک نرم افزار پروتئوس و ide آردوینو پیاده سازی خواهیم کرد. قسمت کد در ide و قسمت نمایش در پروتئوس انجام خواهد شد. این کار بسیار ساده خواهد بود ولی چالشی که در این تمرین با آن مواجه بودیم نصب ابزار های مختلف بود که در قسمت "روش ها و ابزار های مورد نیاز" برای شما آورده ایم.

مقدمه:

در دنیای برنامه نویسی هنگامی که شما میخواهید با زبان برنامه نویسی آشنا شوید معمولاً اولین برنامه ای که مینویسید یک "hello world" ساده است. در اینجا مدار چشمک زن نیز برای ما حکم "hello world" را دارد. پس هدف در اینجا بیشتر آشنایی با قسمت های مختلف ابزار های به کار گرفته شده و اجرا یک برنامه ساده است. برای این آزمایش شما کمی به دانش قبلی در حوزه برنامه نویسی نیاز خواهید داشت.

روش ها و تجهیزات مورد استفاده:

منابع اینترنتی

نرم افزار پروتئوس

Arduino IDE

کتاب خانه آردوینو برای پروتئوس

روش آزمایش:

در این آزمایش ابتدا کدی که با استفاده از تابع delay زده ایم و منجر به چشمک زدن led میشود را در ide نوشته export آن را گرفته و در پروتئوس روی یک Arduino UNO اجرا میکنیم. برای اینکار پس از insert کردن آردوینو روی آن دو بار کلیک چپ کرده و از منوی باز شده program file را همان فایلی که export کردیم قرار میدهیم.

نتایج:

وقتی که برنامه را روی پروتئوس اجرا کردیم مشاهده خواهیم کرد که کلید روشن و خاموش خواهد شد. این همان هدف ما بود.

بحث درمورد نتایج:

از روی برنامه ای که نوشته ایم میتوان یک چیز را برداشت کرد اینکه مدار به ازای مقدار تاخیری که به او داده ایم منتظر و است و هیچ کاری نمیکند که این برای مدار های کاربردی بسیار بد است. به همین خاطر است که ما به سراغ برنامه نویسی threading میرویم که در اینجا همه کار ها باهم به صورت موازی اجرا میشوند.

نتیجه گیری:

به وضوح این برنامه بهترین برنامه برای یک مدار چشمک زن نیست و برنامه های متعدد زیادی میتوان زد که یکی از آنها با استفاده از threading میباشد.

مراجع:

<https://www.youtube.com/watch?v=mz1TsvJzB1c>

<https://www.youtube.com/watch?v=eyuflOLmjJA>

منابع برای مطالعه بیشتر :

<https://medium.com/@randikamadu1995/arduino-led-blink-simulation-on-proteus-d8ba143b5aac>

<https://www.theengineeringprojects.com/2017/01/simple-arduino-led-example-proteus.html>

## عنوان : تمرین 2 (بررسی مدار reset)

چکیده:

در این تمرین قصد داریم بررسی کنیم که مدار reset چطور کار میکند. با فرض اینکه مدار reset، active-low هست به یکسری نتایجی میرسیم که شاید برخلاف باور ذهنی ما باشد. این مدار به کمک خازن و کلید پیاده سازی میشود.

مقدمه:

در دنیای الکترونیک کمتر مداری پیدا میشود که فاقد reset باشد. نقش پررنگ reset سبب شد که به سراغ بررسی این مدار برویم. این آزمایش به ما دقیقاً نشان میدهد که مدار reset چطور کار میکند. کار اینجا بسیار ساده است فقط دو حالت را بررسی میکنیم زمانی که کلید باز است و میخواهد بسته شود و زمانی که کلید بسته است و میخواهد باز شود.

روش ها و تجهیزات مورد استفاده:

منابع اینترنتی

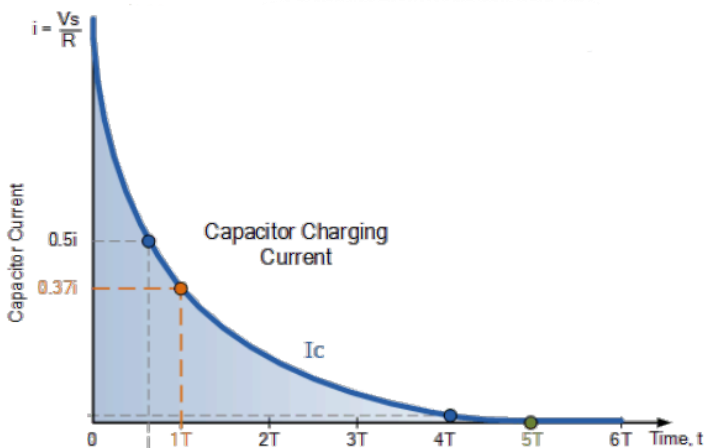
روش آزمایش:

در این آزمایش ما حالات ممکن را بررسی کردیم. یعنی ابتدا گفتیم زمانی که کلید باز است و میخواهد بسته شود و زمانی که کلید بسته است و میخواهد باز شود چه اتفاقی رخ خواهد داد و این را به عنوان گزارش ثبت کردیم.

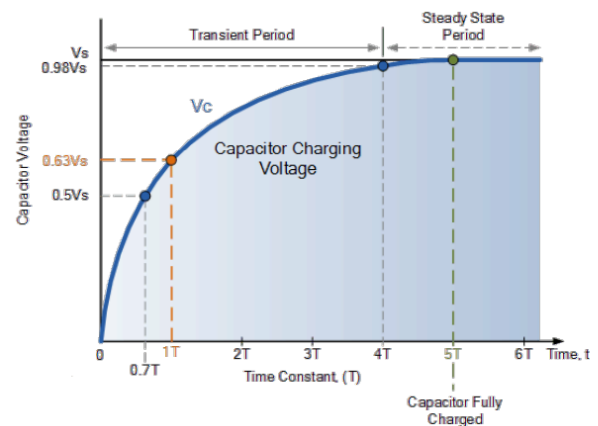
نتایج:

مراحل کار در این مدار به صورتی است که اگر در ابتدا کلید بسته باشد جریان به طور مستقیم به زمین میرود و چون کلید باعث اتصال کوتاه می شود، همچنین ولتاژ vcc به reset میرود. حال اگر کلید را باز کنیم جریان ابتدا به خازن میرود و مقدار reset برای مدت کوتاهی صفر میشود که چون reset در این مدارات active-low هست، reset فعال میشود. بعد از مدت کوتاهی خازن پر شده و مقدار ولتاژی که به reset میرسد به عنوان یک منطقی شناخته شده و reset غیر فعال میشود.

همانطور که در figure 1,2 مشاهده میکنید بعد از T1 که سپری شود مقدار Vcc0.63 به reset منتقل میشود که به عنوان یک منطقی تلقی میشود.



2 Figure



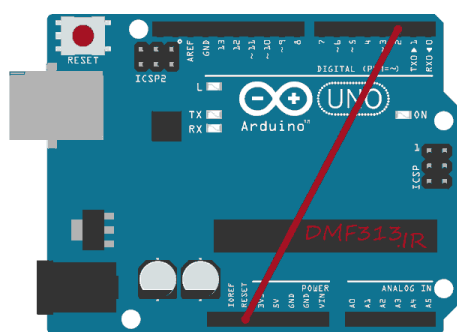
1 Figure

## ریست در AVR:

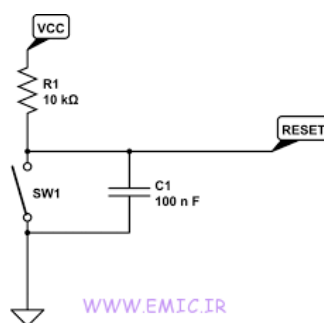
خود ریست یک وقفه ای هست که اگر رخ دهد همه تنظیمات به حالت اول و همه رجیستر ها به مقدار اولیه خود بازمیگردند. مقدار زمانی که RESET فعال هست را start-up گویند. بعد از این زمان مدار شروع به کار کردن میکند و مدار reset غیرفعال میشود. این بین سخت افزاری است. برای عملکرد عادی آی سی نباید به جایی وصل شود و برای ریست کردن میکرو باید به زمین وصل شود.

در avr می توان به چندین روش ، میکرو کنترلر را Reset کرد. با برنامه ریزی یا توسط شرایط خاص که پیش آید. سه راه برای این کار در زیر آمده است:

1. هنگامی که ولتاژ تغذیه متصل به پایه VCC میکرو کنترلر کمتر از مقدار ولتاژ آستانه شود. میکرو کنترلر reset می شود. (مانند figure 4)
2. با استفاده از یک کلید مخصوص reset ( این همان روشی است که در شکل دستور کار مدارش آمده است. (figure 3)
3. با استفاده از watchdog timer، به اینصورت که هر موقع فعال شد، مدار reset هم فعال میشود.



4 Figure



3 Figure

برای جلوگیری از نویز یک مقاومت در حدود 5k تا 10k به vcc وصل می که همان pull-up می باشد. خازن به کار برده شده در مدار reset برای شدن میکرو در لحظه ی قطع وصل شدن برق dc میکرو می باشد. مقدار این خازن معمولاً بین 10u الی 100u میباشد.

بحث در مورد نتایج:

نتایج به دست آمده به وضوح کار یک مدار reset را نشان میدهد. خازن در واقع حکم یک داور را بازی میکند تا زمانی که سوت را نزنده reset فعال بوده و دارد کارش را انجام میدهد زمانی که به مقدار معینی برسد reset غیر فعال میشود. با این وجود یکی از مهم ترین پارامتر ها که باید به خاطر داشت انتخاب درست خازن است که چه مقدار ظرفیت داشته باشد.

نتیجه گیری:

اگر مداری نیاز به reset داشته باشد و reset active-low باشد یکی از راه های ایجاد مدار برای آن، مدار ذکر شده میباشد.

مراجع:

<https://emadars.ir/how-to-connect-the-reset-key-to-the-microcontroller/>

<http://electrolab.ir/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D9%85%DB%8C%D8%A9%D8%B1%D9%88%D8%A9%D9%86%D8%AA%D8%B1%D9%84%D8%B1-avr-%D9%82%D8%B3%D9%85%D8%AA-%D8%A6%D9%87%D8%A7%D8%B1%D9%85-%D8%AD%D8%AF%D8%A7%D9%82%D9%84-%D9%85/>

<https://emic.ir/1396/12/04/%d8%a7%d8%b4%d9%86%d8%a7%db%8c%db%8c-%d8%a8%d8%a7-%d9%be%db%8c%d9%86-%d9%87%d8%a7%db%8c-%d9%85%db%8c%da%a9%d8%b1%d9%88%da%a9%d9%86%d8%aa%d8%b1%d9%84%d8%b1-atmega16-32/>

<https://dmf313.ir/%D8%A7%D9%86%D9%88%D8%A7%D8%B9-%D8%B1%D9%88%D8%B4-%D8%B1%DB%8C%D8%B3%D8%AA-%D9%86%D8%B1%D9%85-%D8%A7%D9%81%D8%B2%D8%A7%D8%B1%DB%8C-%D9%85%DB%8C%DA%A9%D8%B1%D9%88/>

منابع بیشتر برای مطالعه:

[https://octavosystems.com/app\\_notes/osd335x-design-tutorial/bare-minimum-boot/reset-circuitry/](https://octavosystems.com/app_notes/osd335x-design-tutorial/bare-minimum-boot/reset-circuitry/)

<https://www.maximintegrated.com/en/design/technical-documents/app-notes/1/1050.html>

<https://www.quora.com/What-is-reset-circuit-in-embedded-system>

### عنوان: تمرین 3 ( مقاومت pull-up):

چکیده:

هدف از این آزمایش کار با مقاومت pull-up است. در اینجا مداری از این مقاومت خواهیم دید و برای آن مدار مقدار مقاومت pull-up را محاسبه خواهیم کرد. اهمیت این آزمایش از آن جهت است که آزمایش کننده با مقاومت pull-up آشنا شده و متوجه میشود در کجا ها باید از آن استفاده کند. نتیجه اصلی این آزمایش این است که بتوانیم یک مقاومت pull-up را در مدار به کار ببندیم.

مقدمه:

موضوع این آزمایش مقاومت pull-up میباشد و هدف از این آزمایش تحقیق در مورد مقاومت pull-up و محاسبه آن بوده است. تجهیزات و روش های خاصی در این آزمایش به کار گرفته نشده و این آزمایش بیشتر جنبه تئوریک دارد تا عملی. در ادامه خواهیم دید که مقاومت pull-up چقدر کاربردی است و وجود آن برای برخی از مدار ها حیاتی است.

روش ها و تجهیزات مورد استفاده:

منابع اینترنتی

روش آزمایش:

در این آزمایش ابتدا باید در مورد مقاومت pull-up اطلاعات کسب کرد. بعد از آن به سراغ آن مداری میرویم که مقاومت pull-up در آن استفاده شده است. بعد از آن به سراغ فرمول محاسبه مقاومت pull-up میرویم و از روی تقسیم  $V_{cc}$  بر جریانی که از مقاومت pull-up گذر میکند مقدار مقاومت را تعیین میکنیم.

نتایج:

در مدارات الکترونیکی بسیار از pull-up استفاده میشود. کاربرد این مقاومت زمانی است که ما یک پایه را به عنوان ورودی داشته باشیم. حال اگر این پایه به زمین یا منبع تغذیه وصل نباشد آنگاه نمی توان مقدار دقیق آن را تعیین کرد (حالت float). علاوه بر این امکان رسوخ نویز به مدار هم وجود دارد. برای این منظور از مقاومت pull-up استفاده میکنند. این مقاومت بین تغذیه مدار و پایه میکروکنترلر وصل میشود.

این نکته نیز شایان ذکر است که مقاومت pull-up سبب افزایش توان مصرفی می شود. مقدار این مقاومت نباید بیش از حد بزرگ باشد به طوری که تشخیص یک منطقی برای پایه دشوار شود و از طرفی هم نباید آنقدر کم باشد که جریان و توان زیادی را به خود جذب کند.

در این جا اگر مقدار ولتاژ مصرفی led را 2 فرض کنیم با توجه به فرمول داریم: (جریان را 0.001 آمپر فرض میکنیم.)

$$R = \frac{V_{cc}}{I_R} = \frac{5}{0.001} = 5 \text{ k}\Omega$$

بحث در مورد نتایج:

در اینجا چیزی که نمیتوان به طور قطع تعیین نمود جریان است. درست است که ما در اینجا مقدار آن را برابر 0.001A در نظر گرفته ایم اما ممکن است برای مدار های مختلف متفاوت باشد پس نمیتوان گفت که برای  $V_{cc}$  که 5v باشد مقاومت pull-up هم 5 kΩ است.

نتیجه گیری:

مقاومت pull-up شاید به نظر یک مقاومت ساده بیايد ولی کار مهم حذف نویز را برعهده دارد. این مقاومت هم چنین از اتصال کوتاه شدن منبع تغذیه و زمین جلوگیری میکند.

راجع:

<https://melec.ir/%D9%85%D9%82%D8%A7%D9%88%D9%85%D8%AA-%D9%87%D8%A7%DB%8C-pull-up-%D8%8C-%DA%A9%D8%A7%D8%B1%D8%A8%D8%B1%D8%AF-%D9%88-%D9%85%D8%AD%D8%A7%D8%B3%D8%A8%D9%87/>

<https://blog.faradars.org/%D9%85%D9%82%D8%A7%D9%88%D9%85%D8%AA-%D9%BE%D9%88%D9%84-%D8%A2%D9%BE/>

منابع بیشتر برای مطالعه:

[https://en.wikipedia.org/wiki/Pull-up\\_resistor](https://en.wikipedia.org/wiki/Pull-up_resistor)

<https://learn.sparkfun.com/tutorials/pull-up-resistors/all>

<https://eepower.com/resistor-guide/resistor-applications/pull-up-resistor-pull-down-resistor/>

#### عنوان : تمرین 4 (ارسال و دریافت کاراکتر های تصادفی در اردوینو)

چکیده :

هدف از این آزمایش آشنایی با نحوه رد و بدل کردن داده بین دو اردیونو بوده است. چیزی که در این آزمایش به آن رسیدیم کار با serializer بود که یکی از مهم ترین مباحث آموزشی در اردوینو به حساب می آید. اهمیت این آزمایش از آن جهت است که برای برنامه های بزرگ تر که قرار است در آینده با اردوینو بنویسید این آزمایش میتواند مقدمه خیلی خوبی باشد تا بسیاری از مسائل را بتوانید با توابع و روش هایی که در این آزمایش با آنها برخورد خواهید کرد را بیاموزید. پس نتیجه اصلی که از این آزمایش میتوان گرفت کار با توابع مطرح شده در آزمایش میباشد.

مقدمه:

موضوع گزارش نوشتن دو برنامه است یکی برای تولید کاراکتر تصادفی و ارسال آن، یکی هم برای دریافت این کاراکتر ها. شما در این آزمایش نیاز دارید که کمی با ساختار ide اردوینو آشنا باشید. چه بسا این آزمایش پایه بر بسیاری از برنامه های بزرگ تر شما باشد.

روش ها و تجهیزات مورد استفاده:

منابع اینترنتی

Ide اردوینو

روش آزمایش:

ابتدا به سراغ برنامه ای میرویم که قرار است رشته ی تصادفی را ارسال کند. در این برنامه نیز ابتدا به سراغ تولید رشته تصادفی میرویم و آن را پیاده سازی میکنیم سپس به کمک توابع پیاده سازی شده در اردوینو رشته تصادفی را به اردوینو دیگر ارسال میکنیم.

در سمت دیگر که ما قرار است یک برنامه برای دریافت رشته ی تصادفی را بنویسیم ابتدا صبر میکنیم که داده ارسال شود تا آن موقع برنامه در یک حالت انتظار به سر میبرد. بعد از این که داده دیدیم به سراغ دریافت آن میرویم و سپس آن را چاپ میکنیم.

نتایج:

برنامه برای ارسال داده

```
int i = 0;
String random_char[40] = { "a", "b", "c", "d", "e", "f", "g", "h", "i", "j", "k",
"l", "m", "n", "o", "p", "q", "r", "s", "t", "u", "v", "w", "x", "y", "z", "1",
"2", "3", "4", "5", "6", "7", "8", "9", "0" };
String randString = "";

void setup() {
  Serial.begin(230400);
}

void loop() {
  for (i = 0; i < 200; i++) {
    randString = randString + random_char[random(0, 40)];
  }
  delay(1000);
  Serial.write(randString);
}
```

Random\_char یک ارایه هست که تعدادی کاراکتر در خود دارد سپس با استفاده از rand عددی بین 0 تا 39 تولید میکنیم و طبق آن عدد خانه ای را از ارایه انتخاب کرده و در رشته randString میریزیم. این کار را 200 مرتبه دیگر تکرار میکنیم.

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

void loop() {
  Serial.println("waiting for data...");
  while (Serial.available() == 0) {} //waiting
  String rand_string = Serial.readString();
  rand_string.trim();
  Serial.println("Data received your data is:");
  Serial.println(rand_string);
}
```

برنامه بالا برای دریافت داده است. در این برنامه ابتدا منتظر داده میمانیم هنگامی که داده ای مشاهده کردیم آن را با استفاده از readstring میخوانیم و در رشته ای ذخیره میکنیم کار trim حذف چیز هایی مانند /n, whitespace, /r در انتهای رشته هست.

بحث در مورد نتایج:

این برنامه یکی از روش هایی است که میتوان این مسئله را حل کرد و در کنار اینکه ساده به نظر میرسد دارای یک ایراد است. آن ایراد در اردوینو ایست که در حال انتظار است و تا موقعی که داده ای دریافت نکند بیکار خواهد ماند حتی عاجز از چشمک زدن یک led است. برای حل این مشکل میتوان برنامه را به صورت thread نوشت.

نتیجه گیری:

با استفاده از اردوینو میتوان یک رشته ای از کاراکتر های دلخواه خواست و آن را به اردوینو دیگر منتقل کرد.

مراجع:

<https://forum.arduino.cc/t/random-alphanumeric-generator/330466>

<https://www.arduino.cc/reference/en/language/functions/communication/serial/write/>

<https://www.arduino.cc/reference/en/language/functions/communication/serial/readstring/>



منابع جهت مطالعه بیشتر:

<https://create.arduino.cc/projecthub/BitteristSquash/how-to-get-string-data-from-computer-via-serial-port-2ca228>

<https://forum.arduino.cc/t/serial-write-to-send-a-string/87226>