



دانشگاه خوارزمی
دانشکده فنی و مهندسی
رشته مهندسی کامپیوتر

آزمایشگاه شماره ۲: آشنایی با Arduino

نگارندگان: دانیال صابر، سعید وجدی
استاد درس: محمد لالی

آبان ۱۴۰۱

چکیده

آردوینو در اصل یک سخت‌افزار متشکل از یک میکروکنترلر مثل ATmega328 است که یک برنامه خاص تحت عنوان bootloader در آن نصب شده است [۱]. یکی از مشکلاتی که میکروکنترلرها دارند، حجم زیاد مطالب و پیچیدگی زیاد آن‌ها برای یک فرد مبتدی است. کدنویسی آن‌ها در ابتدای کار اغلب دشوار و طاقت فرسا بوده و باعث ناامیدی زیادی برای نوآموزان می‌شوند. آردوینو این مشکل را تا حد زیادی رفع کرده است و افراد مبتدی می‌توانند در شروع کار بدون درگیر شدن با مفاهیم پیچیده با میکروکنترلر کار کنند [۲]. هدف ما در این آزمایش، پیاده‌سازی برنامه‌هایی از قبیل LED چشمک‌زن، نحوه کنترل LEDها با کلید و همینطور آشنایی با تابع‌های Delay و Millis و مفاهیم Multithreading می‌باشد. همچنین، ما تمامی برنامه‌ها بر روی Arduino Uno پیاده می‌کنیم.

۱. مقدمه

برد آردوینو در مؤسسه ی Ivrea Interaction Design برای دانشجویانی که هیچ پیش زمینه ای در مورد مفاهیم الکترونیکی و برنامه ریزی ندارند، طراحی شد. با استفاده از آردوینو می توان کد های کامپیوتری را نوشت و آن ها را بر روی برد فیزیکی بارگذاری کرد. برد آردوینو برای کسانی که تازه شروع به انجام کارها و پروژه های الکترونیکی کرده‌اند بسیار محبوب است. برد آردوینو یک میکروکنترلر نیست، بلکه یک پلتفرم الکترونیکی open-source است. در حقیقت، آردوینو یک PCB است که دارای میکروکنترلرها، LED ها، و تعداد زیادی قطعات و لوازم جانبی دیگر است. معمولاً برای عملیات های دارای ورودی و خروجی مانند کنترل یک موتور، خواندن پیام های سنسورها، و انجام محاسبات کوچک از آردوینو استفاده می شود [۳]. ما در این آزمایش ابتدا یک برنامه چشمک‌زن را در Arduino IDE مینوسیم، سپس با استفاده از یک برد Arduino Uno برنامه نوشته شده را اجرا خواهیم کرد. همچنین، تمرین‌های مرتبط با کنترل چندین LED با استفاده از توابع Delay و Millis را بررسی خواهیم کرد.

۲. روش‌ها و تجهیزات مورد استفاده

برای انجام تمرین، از برنامه Arduino IDE، برنامه Proteus، برد Arduino Uno، LED، و مقاومت ۳۳۰ اهم استفاده شده است.

۳. تمارین

۳/۱. تابع Delay و روش‌های مشابه

نحوه کار تابع delay بسیار ساده است. این تابع یک عدد صحیح را به عنوان ورودی میپذیرد و این عدد میزان تاخیر را در مقیاس میلی ثانیه مشخص میکند. برای مثال وقتی دستور delay(1000) داده میشود، برنامه برای یک ثانیه می‌ایستد. تابع delay() یک تابع مسدود کننده است. توابع مسدود کننده، برنامه را از انجام هر کار دیگری تا زمانی که آن کار خاص کامل نشده است، باز می‌دارد. اگر نیاز باشد که چندین کار همزمان انجام شوند، نمی‌توان از تابع delay() استفاده کرد. برای رفع این مشکل، میتوان از تابع millis() استفاده کرد [۴]. با استفاده از عملیات ریاضی، می‌توان به راحتی بررسی کرد که چقدر زمان گذشته است بدون اینکه کد مسدود شود. در شکل زیر، نحوه کار با این تابع آمده است:

```

1  const int ledPin = 13;
2  int ledState = LOW;
3
4  unsigned long previousMillis = 0;
5
6  const long interval = 1000;
7
8  void setup() {
9      pinMode(ledPin, OUTPUT);
10 }
11
12 void loop() {
13     unsigned long currentMillis = millis();
14
15     if (currentMillis - previousMillis >= interval) {
16         previousMillis = currentMillis;
17
18         if (ledState == LOW) {
19             ledState = HIGH;
20         } else {
21             ledState = LOW;
22         }
23         digitalWrite(ledPin, ledState);
24     }
25 }
```

شکل ۱: نحوه کار تابع millis

این برنامه با کم کردن زمان ثبت شده قبلی از زمان کنونی آغاز میشود. اگر باقی‌مانده عددی بزرگتر از interval باشد، مقدار زمان ثبت شده قبلی را با زمان کنونی به‌روز میکند و LED را روشن و یا خاموش میکند.

۳/۲. برنامه چشمک‌زن

برای انجام این تمرین بایستی هر نیم ثانیه وضعیت خروجی صفر را برعکس کنیم. برنامه در شکل زیر آورده شده است و شبیه‌سازی در Proteus نیز ضمیمه شده است.

```

Reversing_state_0.ino
1  int stateLED0 = LOW;
2
3  void setup() {
4
5      pinMode(13, OUTPUT);
6
7  }
8
9  void loop() {
10
11      digitalWrite(13, stateLED0);
12      delay(150);
13      digitalWrite(13, !stateLED0);
14      delay(150);
15
16  }

```

شکل ۲. عوض کردن خروجی 0

۳/۳. هشت LED چشمک‌زن با کلید

در این تمرین، هشت LED که هر کدام به مقاومت ۳۳۰ اهم متصل هستند در فاصله زمانی نیم ثانیه با استفاده از کلید روشن و خاموش میشوند. کد این برنامه در شکل زیر آمده است. همچنین شبیه‌سازی این تمرین در Proteus ضمیمه شده است.

```

1  void setup() {
2
3      for (int pin = 6; pin < 14; pin++) {
4          pinMode(pin, OUTPUT);
5      }
6
7      pinMode(1, INPUT);
8
9  }
10
11 void loop() {
12
13     int BUTTONstate = digitalRead(1);
14
15     if (BUTTONstate == LOW){
16
17         for (int pin = 6; pin < 14; pin++){
18             digitalWrite(pin, HIGH);
19         }
20
21         delay(500);
22
23         for (int pin = 6; pin < 14; pin++){
24             digitalWrite(pin, LOW);
25         }
26
27         delay(500);
28     }
29     else{
30
31         for (int pin = 6; pin < 14; pin++){
32             digitalWrite(pin, LOW);
33         }
34

```

شکل ۴: هشت LED چشمک‌زن

۳/۴. هشت LED چشمک‌زن یکی در میان

این تمرین نیز شباهت‌هایی با تمرین قبل دارد، با این تفاوت که این هشت LED به ترتیب روشن شوند و بعد خاموش شوند. به این طریق که ابتدا LED اول روشن می‌شود. سپس LED دوم روشن شده و LED اول خاموش می‌شود. بعد از آن LED سوم روشن می‌شود و LED دوم خاموش می‌شود. این روند تا انتها ادامه می‌یابد. کد این برنامه و همچنین شبیه‌سازی این تمرین در Proteus ضمیمه شده است.

۳/۵. Multithreading در Arduino

مولتی‌تسکینگ یا توانایی انجام چند کار به صورت همزمان، موجب رخ دادن انقلابی بزرگ در بهره‌وری، انعطاف‌پذیری، تطابق‌پذیری و کارآمدی کامپیوترها شد. امروزه در سیستم‌های نهفته هم میکروکنترلرها به این توانایی مجهز هستند و می‌توانند با استفاده از مولتی‌تسکینگ دو یا چند وظیفه را به صورت همزمان انجام دهند، بدون آنکه در انجام هر کدام از آن‌ها وقفه ایجاد شود. خوشبختانه آردوینو هم قابلیت مولتی‌تسکینگ را دارد [۵]. با استفاده از تابع millis که در تمرین اول شرح داده شد، میتوان چند کار به طور هم زمان انجام داد. برای شبیه‌سازی، ما در این تمرین دو LED را با فواصل زمانی ۲۰۰ و ۲۰۰۰ میلی‌ثانیه به طور همزمان روشن و خاموش میکنیم. کد این برنامه در شکل زیر آمده است. همچنین شبیه‌سازی این تمرین در Proteus ضمیمه شده است.

```

Blinking_2_LEDs_with_Tread.ino  test.ino
1  unsigned long prev_Time_T1 = millis();
2  unsigned long prev_Time_T2 = millis();
3
4  long interval_T1 = 200;
5  long interval_T2 = 2000;
6
7  int LED1_state = LOW;
8  int LED2_state = LOW;
9
10 void setup() {
11
12     pinMode(13, OUTPUT);
13     pinMode(12, OUTPUT);
14
15 }
16 void loop() {
17     unsigned long currentTime = millis();
18     //Task1
19
20     if (currentTime - prev_Time_T1 > interval_T1){
21         LED1_state = !LED1_state;
22         digitalWrite(13, LED1_state);
23
24         prev_Time_T1 = currentTime;
25     }
26     //Task2
27
28     if (currentTime - prev_Time_T2 > interval_T2){
29         LED2_state = !LED2_state;
30         digitalWrite(12, LED2_state);
31
32         prev_Time_T2 = currentTime;
33     }
34

```

شکل ۵: مولتی‌تسکینگ در Arduino

۴. نتیجه‌گیری

در این آزمایش با نحوه کار با Arduino و برنامه‌نویسی در Arduino IDE آشنا شدیم. در تمرین اول دیدیم که چه تفاوت‌هایی بین توابع delay و millis وجود دارد. سپس، نحوه پیاده‌سازی برنامه LED چشمک‌زن را مشاهده کردیم. همچنین، در تمرین بعد برنامه چشمک‌زن را به هشت LED گسترش دادیم و آزمایش کردیم که چگونه میتوان هشت LED را به طور همزمان و یا به طور یکی در میان روشن و خاموش کرد. در نهایت، در تمرین آخر نحوه انجام چند کار با Arduino را بررسی کردیم و شرح دادیم که چگونه میتوان دو LED را با فواصل زمانی متفاوت روشن و خاموش کرد.

۵. مراجع

- [1]- <https://parselectronic.com/arduino-board-parts-and-application/>
- [2]- <https://microlearn.ir/9480/what-is-arduino.html>
- [3]- <https://fanamoozan.com/arduino/>
- [4]- <https://randomnerdtutorials.com/why-you-shouldnt-always-use-the-arduino-delay-function/>
- [5]- <https://melec.ir/%D8%A2%D9%85%D9%88%D8%B2%D8%B4-%D9%85%D9%88%D9%84%D8%AA%DB%8C-%E2%80%8C%D8%AA%D8%B3%DA%A9%DB%8C%D9%86%DA%AF-%D8%AF%D8%B1-%D8%A2%D8%B1%D8%AF%D9%88%DB%8C%D9%86%D9%88/>