

گزارش شبیه سازی روی برد Arduino Uno

هلیا قربانی - ۹۸۲۴۳۵۳

پیاده سازی نرم افزاری:

```
const int BUTTON1 = 7;  
const int BUTTON2 = 6;  
const int LED1 = 13;  
const int LED2 = 12;  
const int LED3 = 11;  
const int LED4 = 10;  
const int LED5 = 9;  
const int LED6 = 8;
```

```
int BUTTONstate1 = 0;  
int BUTTONstate2 = 0;
```

```
const int LEDs[6] = {LED1, LED2, LED3, LED4, LED5, LED6};
```

در ابتدای برنامه متغیر ها و ثابت ها تعریف شده اند که شامل دکمه ها، LED ها، وضعیت دکمه ها و آرایه از LED ها هستند و برای هر یک از دکمه ها و LED ها یک پین از برد در نظر گرفته شده است.

```
void led_off() {  
    for (int i = 0; i < 6; i++){  
        digitalWrite(LEDs[i], LOW);  
    }  
}
```

سپس یک تابع تعریف شده است که در صورتی که کلید رها شد برای خاموش کردن همه LED ها از آن استفاده شود.

```
void setup() {  
    pinMode(BUTTON1, INPUT);  
    pinMode(BUTTON2, INPUT);  
    for (int i = 0; i < 6; i++){  
        pinMode(LEDs[i], OUTPUT);  
    }  
}
```

در تابع setup تنظیمات مورد نیاز مانند ورودی کردن ۲ دکمه و خروجی کردن LED ها انجام شده

است.

در حلقه loop که بخش اجرایی برنامه است دو بخش اصلی وجود دارد که به شرح زیر است:

```
BUTTONstate1 = digitalRead(BUTTON1);
if (BUTTONstate1 == HIGH){
    for (int i = 0; i < 6; i++){
        if (BUTTONstate1 == HIGH){
            digitalWrite(LEDs[i], HIGH);
            digitalWrite(LEDs[(i+5) % 6], LOW);
            delay(1000);
            BUTTONstate1 = digitalRead(BUTTON1);
        }
        else{
            break;
        }
    }
}

else{
    led_off();
}
```

در ابتدا حالت اول شبیه سازی شده است که وضعیت فشردن کلید ۱ بررسی می شود و در صورتی که فشرده شده بود به ترتیب LED ها از چپ به راست روشن می شوند و هر LED که روشن می شود LED قبلی که روشن بود، خاموش می شود و این روند تا رها شدن کلید ۱ ادامه دارد. همچنین در صورتی که کلید ۱ رها شود تابع اشاره شده برای خاموش کردن LED ها فراخوانی می شود. به طور دقیق تر رونر روشن و خاموش شدن LED ها به این شکل است که ابتدا LED1 روشن می شود و LED6 خاموش می شود که در ابتدا LED6 خاموش است و خاموش باقی می ماند در چرخش های بعدی این LED روشن است، در این مرحله خاموش خواهد شد. سپس LED1 خاموش شده و LED2 روشن می شود و این روند تا رسیدن به مرحله ای که LED6 روشن می شود و LED5 خاموش می شود ادامه دارد و پس از این تا رها شدن کلید این روند تکرار خواهد شد. تاخیر در نظر گرفته شده برای روشن و خاموش شدن LED ها ۱s است.

```
BUTTONstate2 = digitalRead(BUTTON2);
int status = HIGH;
if(BUTTONstate2 == HIGH){
    for(int i = 0; i < 6; i++){
```

```

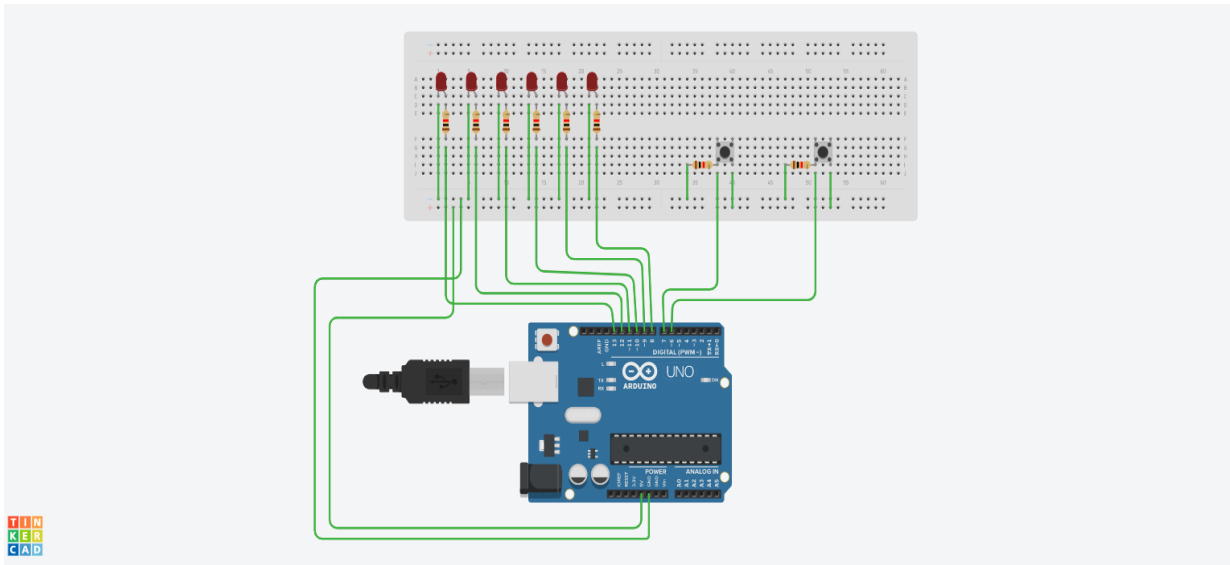
        if(i % 2 == 0){
            digitalWrite(LEDs[i], status);
        }
        else{
            digitalWrite(LEDs[i], !status);
        }
        if (i == 5 && digitalRead(BUTTON2)){
            delay(1000);
            status = !status;
            i = -1;
        }
    }
}
else{
    led_off();
}
}

```

در حالت دوم وضعیت فشرده شدن کلید ۲ بررسی می شود و در صورتی که فشرده شده بود ابتدا LED های فرد به مدت ۱s روشن می شوند و سپس LED های فرد خاموش شده و LED های زوج به مدت ۱s روشن می شوند و این روند تا رها شدن کلید ۲ ادامه دارد. همچنین در صورتی که کلید ۲ رها شود تابع اشاره شده برای خاموش کردن LED ها فراخوانی می شود. (LED ها از چپ به راست در نظر گرفته شده اند)

چون آرایه ای از LED ها در نظر گرفته شده و در حلقه index ها در نظر گرفته شده اند و به طور مثال index=0 همان LED1 است یعنی LED های فرد در index ها زوج قرار دارند پس ابتدا LED های فرد که index زوج دارند را ۱s روشن و LED های زوج را خاموش می کنیم. برای این کار متغیر status ابتدا برابر HIGH است که با آن LED های فرد را روشن و با not کردن آن LED های زوج را خاموش می کنیم. البته پس از اینکه وضعیت خاموش یا روشن بودن همه LED ها تعیین شد مجدد وضعیت فشرده شده کلید ۲ بررسی می شود و اگر کلید در حال فشرده شدن بود متغیر status تغییر وضعیت می دهد و not می شود تا این بار LED های فرد را خاموش و LED های زوج را روشن کند. این روند تا زمانی که کلید ۲ فشرده می شود ادامه می یابد و سپس متوقف می شود.

شبیه سازی سخت افزاری:



در پیاده سازی سخت افزاری از برد Arduino Uno، دو Push Button، هشت مقاومت به ازای مجموع تعداد LED ها و Push Button ها، شش LED و یک Breadboard استفاده شده است. خطوط تغذیه مثبت و منفی بردبرد را به ترتیب به ولتاژ 5v و GND به برد Arduino Uno وصل می کنیم. کاتد LED ها به تغذیه منفی بردبرد و بخش آند را با یک مقاومت به پین هایی که در پیاده سازی نرم افزاری مشخص کرده بودیم متصل می کنیم. برای کلید ها هم به شکل مشابه عمل می کنیم. یک سر آن ها را به تغذیه مثبت و سر دیگر را به پین مشخص شده وصل می کنیم. به این ترتیب زمانی که برنامه اجرا شود با فشردن کلید ۱ LED ها به ترتیب روشن می شوند و با روشن شدن هر LED، LED قبلی خاموش می شود و این روند تا رها کردن کلید ۱ ادامه می یابد. همچنین زمانی که کلید ۲ فشرده شود ابتدا LED های فرد 1s روشن می شوند و سپس LED های فرد خاموش شده و LED های زوج ۱s روشن می شوند و LED های فرد خاموش می شوند و این روند تا رها کردن کلید ۲ ادامه دارد.

شبیه سازی در Tinkercad انجام شده است و در لینک زیر قابل مشاهده است (نیاز به vpn و لاگین در سایت Tinkercad دارد):

<https://www.tinkercad.com/things/kXfbr1vYoUp-smashing-luulia/editel?sharecode=6C6Ybf3AS7CIJla6leK2luouetNGz-tmuemgqosy1s>

همچنین کد شبیه سازی انجام شده در فایل code.ino قرار دارد و خروجی آن در ویدیو output.mp4 قابل مشاهده است.