

## به نام خداوند جان و خرد

نام و نام خانوادگی	تاریخ آزمایش	شماره آزمایش
مینا زواری	چهارشنبه 24 اردیبهشت	آزمایش شماره 33

### عنوان :

پیاده‌سازی وقفه (Interrupt) در آردوینو برای پایش تغییرات پایه دیجیتال

### هدف آزمایش:

هدف اصلی این آزمایش، طراحی و پیاده‌سازی یک مدار با استفاده از برد آردوینو UNO و یک کلید فشاری است که قادر به تشخیص تغییر وضعیت یک پایه دیجیتال (پایه 2) از طریق مکانیزم وقفه (Interrupt) باشد. هدف دیگر، نمایش نحوه استفاده از Interrupt Service Routine (ISR) برای اجرای یک تابع خاص در پاسخ به رویداد وقفه و همچنین ارسال پیام از طریق ارتباط سریال به کامپیوتر در هنگام وقوع وقفه است. این آزمایش همچنین بر درک تفاوت‌های بین پولینگ (Polling) و وقفه و مزایای استفاده از وقفه در برنامه‌نویسی آردوینو تأکید دارد.

### تئوری آزمایش:

- **برد آردوینو: UNO** به عنوان میکروکنترلر اصلی، وظیفه‌ی تشخیص رویداد وقفه، اجرای تابع سرویس وقفه (ISR) و ارسال داده از طریق پورت سریال را بر عهده دارد. آردوینو Uno دارای دو پایه مخصوص وقفه خارجی (External Interrupt) است: پین 2 (INT0) و پین 3 (INT1).
- **وقفه (Interrupt):** یک قابلیت در میکروکنترلرها است که به CPU اجازه می‌دهد تا به صورت موقت اجرای برنامه اصلی خود را متوقف کرده و به یک رویداد خاص (مانند تغییر وضعیت یک پایه دیجیتال) پاسخ دهد. این مکانیزم به سیستم اجازه می‌دهد تا به رویدادهای مهم به سرعت و با کارایی بالا پاسخ دهد، بدون نیاز به بررسی مداوم (Polling) وضعیت پین‌ها. وقفه می‌تواند توسط لبه بالارونده (RISING)، لبه پایین‌رونده (FALLING)، تغییر وضعیت (CHANGE) یا وضعیت LOW یک پین فعال شود.
- **پایه‌های وقفه در آردوینو: Uno** همانطور که در تصاویر مشخص شده است، پایه‌های دیجیتال 2 و 3 در برد آردوینو Uno به ترتیب به عنوان INT0 و INT1 برای وقفه‌های خارجی اختصاص داده شده‌اند. در این آزمایش، از پایه 2 (INT0) برای تشخیص تغییر وضعیت استفاده می‌شود.

- **پین‌های دیجیتال آردوینو:** پین‌های 0 تا 13 در آردوینو UNO می‌توانند به عنوان ورودی یا خروجی دیجیتال پیکربندی شوند. در این آزمایش، از پین دیجیتال 2 برای خواندن وضعیت کلید فشاری و از پین 13 (LED) داخلی آردوینو (برای نمایش بصری تغییر وضعیت استفاده می‌شود).
- **ارتباط سریال:** برای ارسال پیام‌ها و وضعیت متغیر state از آردوینو به کامپیوتر استفاده می‌شود. از Serial Monitor در محیط Arduino IDE می‌توان برای مشاهده‌ی این داده‌ها استفاده کرد.
- **تابع attachInterrupt () :** این تابع در آردوینو برای اتصال یک رویداد وقفه به یک ISR خاص استفاده می‌شود. فرم کلی آن `attachInterrupt(interrupt, function, mode)` است:

  - **Interrupt :** شماره وقفه مربوط به پایه (برای پین 2، این مقدار 0 است، زیرا مربوط به INT0 است)
  - **Function :** نام ISR که هنگام وقوع وقفه اجرا می‌شود.
  - **mode :** حالت فعال‌سازی وقفه (مثلاً **CHANGE** که در این آزمایش استفاده شده و در هر تغییر وضعیت پایه وقفه را فعال می‌کند)

#### شرح مدار و قطعات مورد استفاده:

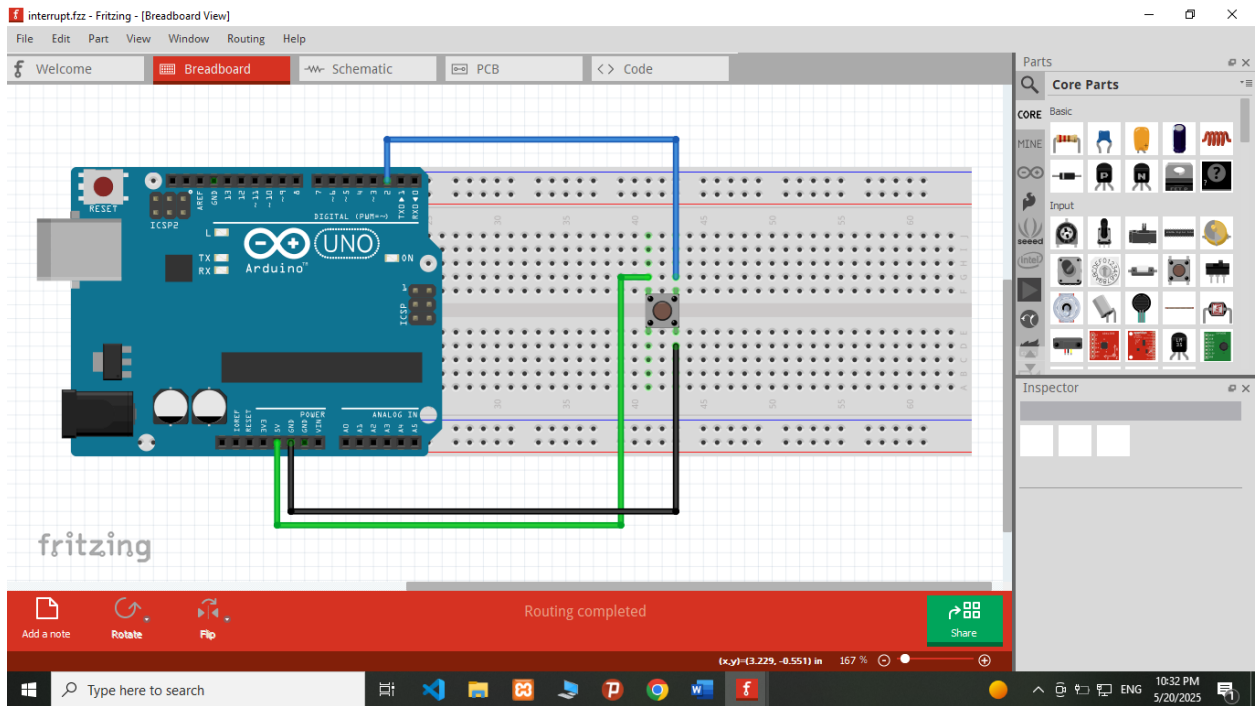
- برد آردوینو UNO
- کلید فشاری Push Button : 1 عدد
- بردبورد
- سیم‌های مخابراتی

#### روش انجام آزمایش:

##### ش انجام آزمایش:

##### 1. اتصالات سخت‌افزاری:

- پایه C پوش باتن را به پایه دیجیتال 2 آردوینو متصل می‌کنیم. (سیم آبی)
- پایه D پوش باتن را به زمین (GND) آردوینو متصل می‌کنیم. (سیم مشکی)
- پایه A پوش باتن را به پایه 5V آردوینو متصل می‌کنیم. (سیم سبز)



## 2. برنامه نویسی آردوینو :

- برنامه آردوینو IDE را باز کنید.
- کدهای زیر را در آن وارد کنید:

```
int pin = 2; // interrupt pin
volatile int state = LOW;

void setup() {
    pinMode(13, OUTPUT);

    pinMode(pin, INPUT_PULLUP); //
    داخلی pull-up استفاده از مقاومت
    Serial.begin(9600); // شروع ارتباط
    سریال

    attachInterrupt(digitalPinToInterrupt
    (pin), blink, CHANGE);
}

void loop() {
    digitalWrite(13, state);
}

void blink() {
    state = !state;

    Serial.print("Interrupt! State is
    now: ");

    Serial.println(state == HIGH ?
    "HIGH" : "LOW");
}
```

## نتیجه گیری:

در این آزمایش، یک مدار و برنامه برای پایش تغییرات وضعیت یک پایه دیجیتال با استفاده از مکانیزم وقفه (Interrupt) در برد آردوینو UNO با موفقیت طراحی و پیاده‌سازی شد. با هر بار فشردن یا رها کردن کلید متصل به پایه 2 آردوینو، وقفه فعال شده و تابع `blink()` به عنوان Interrupt Service Routine (ISR) فراخوانی می‌شود. این تابع، وضعیت یک متغیر `volatile` به نام `state` را تغییر داده و LED داخلی آردوینو (پین 13) را خاموش یا روشن می‌کند. همچنین، پیامی حاوی وضعیت جدید به Serial Monitor ارسال می‌شود.

این آزمایش به وضوح نشان داد که چگونه می‌توان از وقفه‌ها برای پاسخگویی سریع و کارآمد به رویدادهای خارجی استفاده کرد، بدون اینکه نیاز باشد به صورت مداوم وضعیت پین‌ها را در حلقه `loop()` بررسی کنیم. این رویکرد (Interrupt-driven) نسبت به پولینگ (Polling) در سناریوهایی که نیاز به پاسخگویی فوری به رویدادها وجود دارد (مانند خواندن ورودی‌های کاربر، دریافت داده از سنسورها یا ارتباطات سریال)، کارایی و پایداری سیستم را به طور قابل توجهی افزایش می‌دهد. استفاده از متغیر `volatile` نیز اهمیت مدیریت صحیح حافظه در هنگام تعامل بین ISR و قسمت اصلی برنامه را برجسته کرد. این تکنیک می‌تواند در طیف وسیعی از پروژه‌های آردوینو، از جمله سیستم‌های امنیتی، رباتیک و اتوماسیون خانگی، کاربرد داشته باشد.