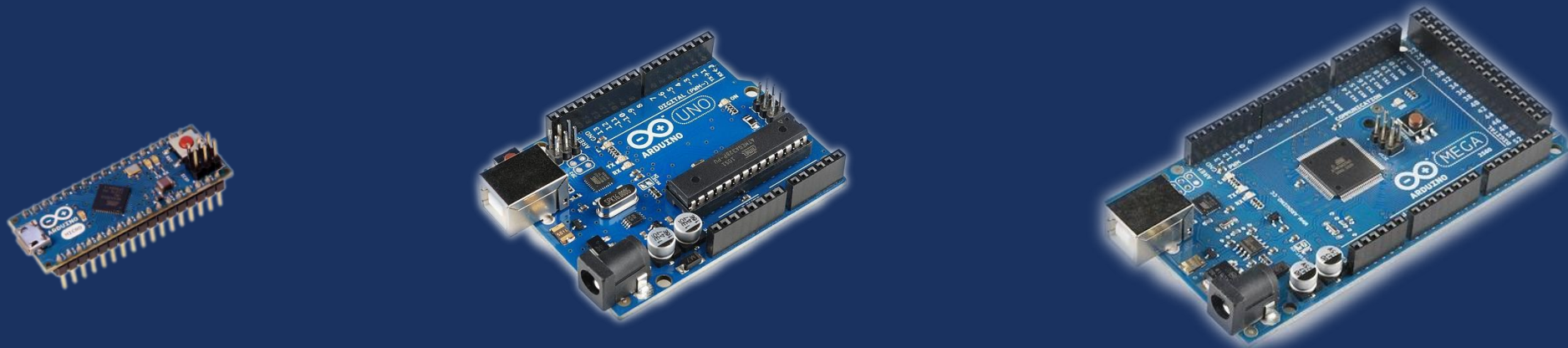


# مبانی کار با سخت افزار آردوینو

احمد رضا حیدری، دانشگاه کاشان - بهار ۱۴۰۴



# سر فصل های دوره و زمان بندی جلسات

- آشنایی مقدماتی با میکروکنترلرها و برد آردینو (جلسه ۱ پنج شنبه ۱۴۰۴/۲/۱۸ ساعت ۱۵ به صورت مجازی)
- کد نویسی آردینو با زبان C++ (جلسه ۲ جمعه ۱۴۰۴/۲/۱۹ ساعت ۱۵ به صورت مجازی)
- مبانی کار با سخت افزار آردوینو (جلسه ۳ یکشنبه ۱۴۰۴/۲/۲۱ ساعت ۲۰ به صورت حضوری)
- راه اندازی سنسورها و ماژولها (جلسه ۴ دوشنبه ۱۴۰۴/۲/۲۲ ساعت ۲۰ به صورت حضوری)
- راه اندازی و کنترل سرعت موتور DC (جلسه ۵ یکشنبه ۱۴۰۴/۲/۲۸ ساعت ۲۰ به صورت حضوری)
- راه اندازی سروو موتور و کنترل زاویه ای (جلسه ۵ یکشنبه ۱۴۰۴/۲/۲۸ ساعت ۲۰ به صورت حضوری)
- راه اندازی LCD کاراکتری برای نمایش اطلاعات (جلسه ۶ دوشنبه ۱۴۰۴/۲/۲۹ ساعت ۲۰ به صورت حضوری)
- راه اندازی و کنترل بازوی رباتیک با آردوینو (جلسه ۶ دوشنبه ۱۴۰۴/۲/۲۹ ساعت ۲۰ به صورت حضوری)

# شروع کار با آردوینو

**آردوینو** یک شرکت، پروژه و جامعه کاربری متن باز در زمینه سخت افزار و نرم افزار کامپیوتری است که کیت هایی مبتنی بر میکروکنترلر طراحی و تولید می کند. این کیت ها برای ساخت دستگاه های دیجیتال و اشیای تعاملی استفاده می شوند که توانایی حس کردن و کنترل اشیای موجود در دنیای فیزیکی را دارند. محصولات این پروژه به صورت متن باز هم در بخش سخت افزار و هم در بخش نرم افزار عرضه می شوند. ([ویکی پدیا](#))

منابع آردوینو در وبسایت رسمی آردوینو [arduino.cc](http://arduino.cc):

- [Getting started guide](#)

- [Arduino introduction](#)

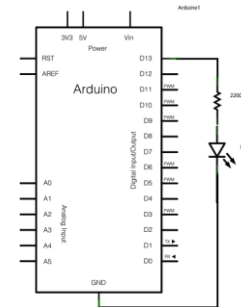
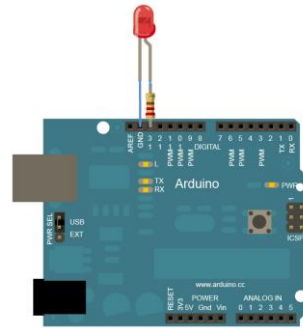
- [Arduino tutorials](#)

- How it works: [Foundations](#)

- [Language reference](#)

- [Arduino Hardware](#)

- [Arduino IDE](#)



```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000); // wait for a second
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
  delay(1000); // wait for a second
}
```

# موارد پیشنهادی برای شروع

برد آردوینو

■ UNO : پرتفدارترین برد چندمنظوره

■ Mega : دارای ورودی/خروجی های بیشتر و وقفه های بیشتر نسبت UNO

■ Nano : مشابه UNO ولی در قالبی کوچک تر

■ بردهای UNO و Leonardo از نظر شکل ظاهری یکسان هستند و با انواع شیلدهای شخص ثالث سازگارند

■ برد Nano ارزان و جمع و جور است، مناسب برای پروژه های دائمی. برای استفاده روی بردبرد، معمولاً نیاز به لحیم کاری پین هدر دارد.

■ هر سه برد UNO، Mega و Nano از سطح منطقی ۵ ولت استفاده می کنند.

سایر تجهیزات مورد نیاز:

کابل USB A to B (کابل پرینتری)

برد برد

سیم های جامپر

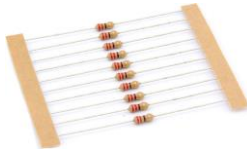
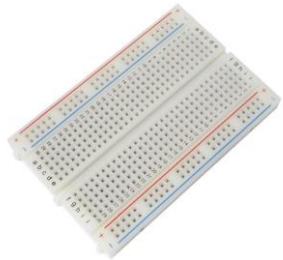
LED

مقاومت ها (۲۲۰ اهم، ۱۰ کیلو اهم)

پتانسیومتر (مقاومت متغیر، ۱۰ کیلو اهم)

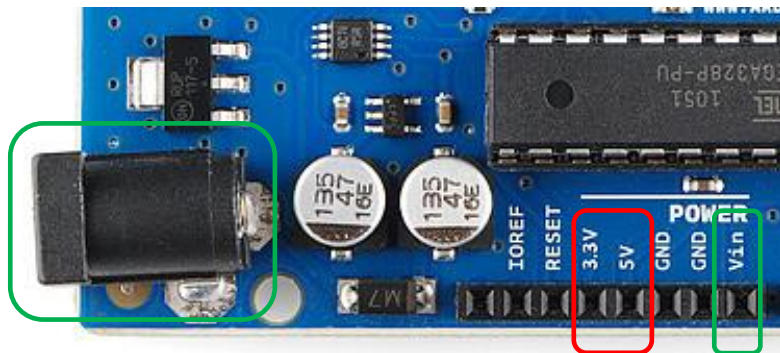
شستی (Pushbutton)

منبع تغذیه یا باتری



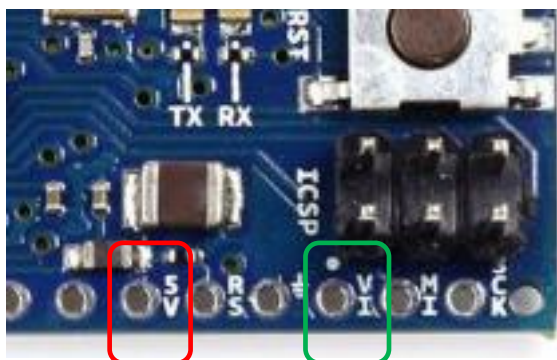
# تغذیه و توان آردوینو

Adruino Uno / Leonardo



In Out

Adruino Micro



تأمین توان برای آردوینو (بردهای ۵ ولتی):

- توان از طریق USB : توان کافی برای خود آردوینو و وسایل جانبی با جریان پایین (مثل یک LED ساده) را فراهم می‌کند.

نکته: هنگامی که آردوینو از طریق USB تغذیه می‌شود. نباید همزمان با آداپتور مجزا به آردوینو تغذیه داد!

- پین Vin / آداپتور نری (Barrel Plug) : ورودی ۷ تا ۱۲ ولت برای تغذیه رگولاتور ولتاژ داخلی برد.

- انتخاب منبع تغذیه به صورت خودکار انجام می‌شود.

تأمین توان برای وسایل جانبی:

- پین ۵ ولت: خروجی رگولاتور ولتاژ داخلی برد، حداکثر جریان حدود ۲۵۰ میلی‌آمپر.

- پین ۳٫۳ ولت: خروجی تنظیم‌شده، حداکثر ۵۰ میلی‌آمپر.

- پین‌های GPIO : حداکثر جریان قابل تحمل برای هر پین ۲۰ میلی‌آمپر است.

نکته: اگر وسایل جانبی در مجموع بیش از ۲۵۰ میلی‌آمپر جریان نیاز دارند، استفاده از منبع تغذیه خارجی توصیه می‌شود. موتورها و سرووها باید حتماً از منبع تغذیه خارجی (مثل آداپتور یا باتری) تغذیه شوند.

# آموزش نصب ARDUINO IDE

## Downloads

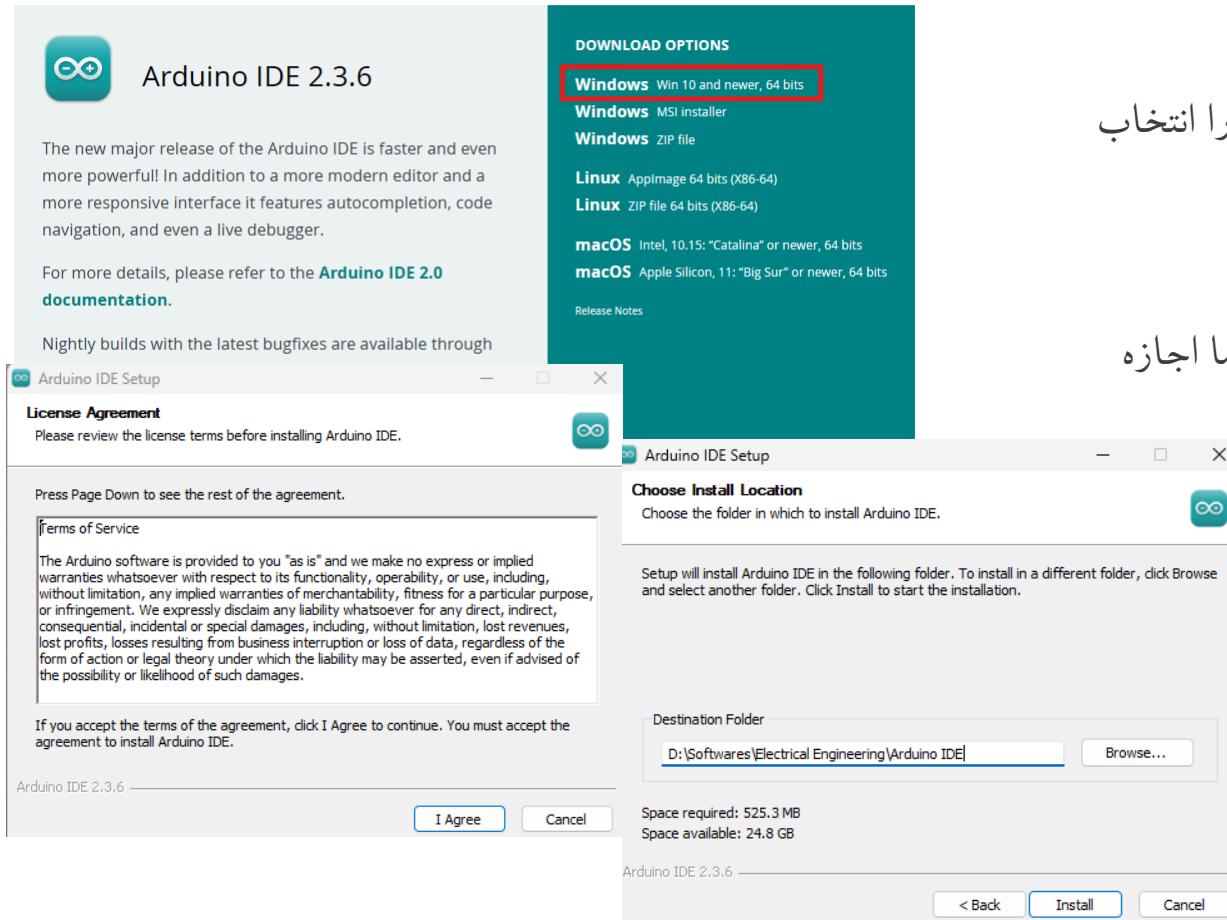
وارد سایت رسمی آردوینو شوید:

<https://www.arduino.cc/en/software>

بسته به سیستم عامل خود یکی از گزینه ها را انتخاب کنید در اینجا ما ویندوز را انتخاب میکنیم.

نصب در ویندوز فایل exe را اجرا کنید.

گزینه های پیش فرض را بپذیرید (Next → Install) در حین نصب، اگر از شما اجازه نصب درایورها خواسته شد، Yes بزنید.





# دیاگرام پین‌های آردوینو (PIN MAPPING / PINOUT)

دیاگرام پین‌های آردوینو (PINOUT) اطلاعات مربوط به عملکردهای مختلف هر پین از میکروکنترلر را ارائه می‌دهند.

■ مبدل آنالوگ به دیجیتال (ADC): مشخص‌کننده‌ی پین‌های آنالوگ.

■ پین‌های دارای قابلیت (PWM): برای تولید سیگنال‌های مدولاسیون پهنای پالس.

■ پین‌های ارتباط سریال: شامل (TX/RX) ارتباط SDA/SCL و I<sup>2</sup>C و SPI.

■ وقفه‌ها و وقفه‌های تغییر پین (INT / PCINT): برای پاسخ سریع به رویدادهای خارجی.

■ شماره پورت و بیت در ATmega برای دسترسی مستقیم و سطح پایین‌تر به پین‌ها.

نقشه‌های پین برای هر مدل برد در صفحه محصولات آردوینو و یا اگر از میکروکنترلر

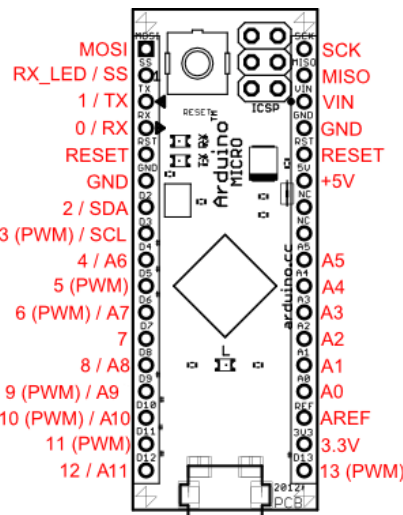
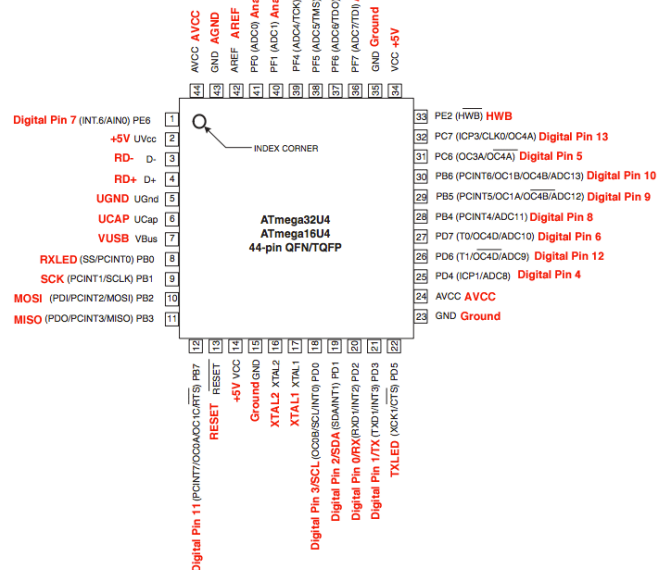
دیگری استفاده می‌کنید در صفحات محصولات آن شرکت یافت می‌شود.

**Atmega168 Pin Mapping**

Arduino function		Atmega168 Pin	Atmega168 Pin	Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22	GND
GND	GND	8	21	AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3) digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MOSI, MISO, SCK connections (Atmega168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Arduino functions in red



# پین اوت دیاگرام چیست؟ (PINOUT DIAGRAM)

پین اوت دیاگرام، نمایشی گرافیکی از پایه‌ها (پین‌ها) در یک قطعه الکترونیکی (مانند میکروکنترلر، آی‌سی یا کانکتور) است که نحوه اتصال و کارکرد هر پایه را مشخص می‌کند.

کاربردها:

- طراحی مدارهای الکترونیکی

- سیم‌کشی و عیب‌یابی

- برنامه‌نویسی میکروکنترلرها

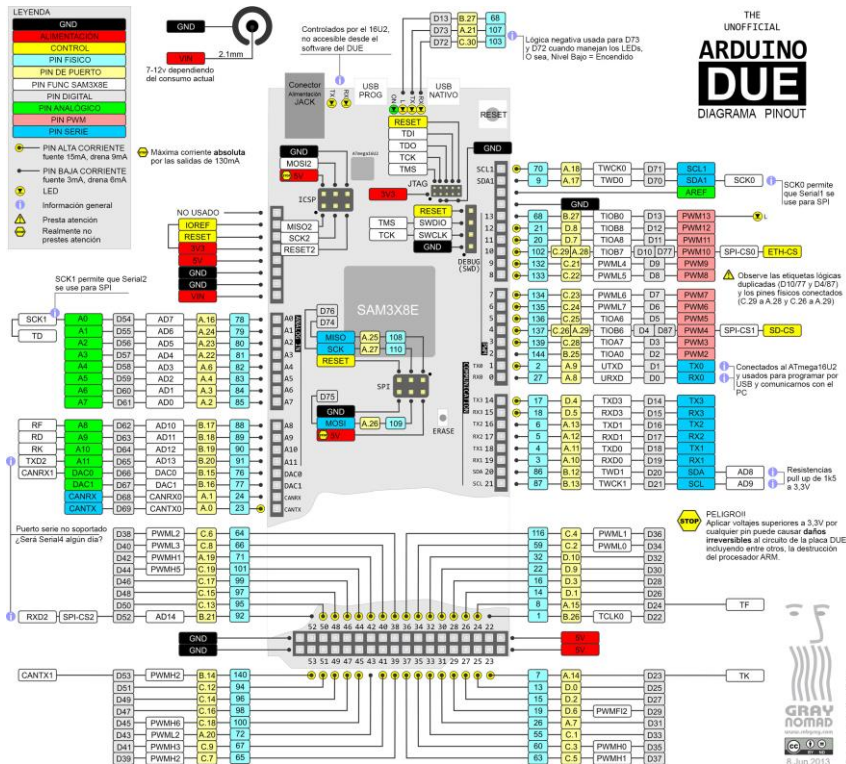
- اتصال به سنسورها و تجهیزات جانبی

- اجزای دیاگرام:

- شماره پین

نام پین (مانند VCC, GND, TX, RX, GPIO)

نوع عملکرد (ورودی، خروجی، آنالوگ، دیجیتال و...)



<https://docs.arduino.cc/hardware>



# خلاصه توضیحات پین‌اوت دیاگرام برد های آردوینو

پین‌اوت دیاگرام آردوینو، نقشه‌ای است که نشان می‌دهد هر پایه (Pin) از برد آردوینو چه عملکردی دارد. این دیاگرام برای اتصال سنسورها، موتورها، نمایشگرها و دیگر اجزا به برد ضروری است. دسته‌بندی پایه‌ها:

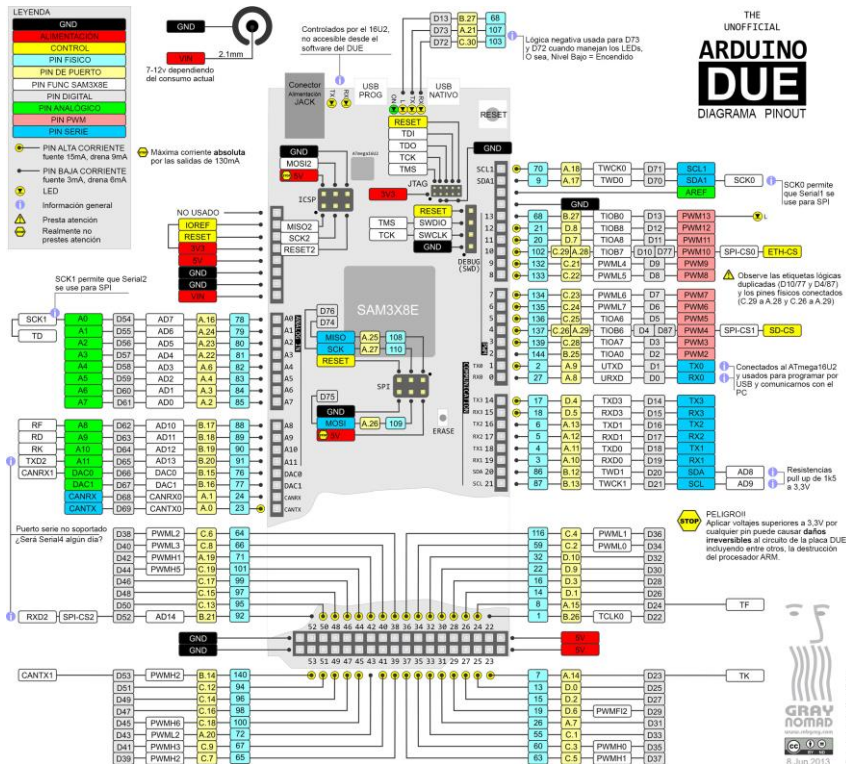
■ **Digital Pins** (پین‌های دیجیتال): پایه‌های شماره‌دار D0 تا D13 برای ورودی یا خروجی دیجیتال (روشن/خاموش). برخی از آنها از PWM پشتیبانی می‌کنند.

■ **Analog Pins** (آنالوگ): پایه‌های A0 تا A5 برای خواندن سیگنال‌های آنالوگ (ولتاژ متغیر).

■ **Power Pins** (تغذیه): پایه‌هایی مثل 5V، 3.3V، VIN و GND برای تأمین برق مدار.

■ **Communication Pins** (پین‌های ارتباطی): شامل TX/RX برای سریال، SDA/SCL برای I<sup>2</sup>C، و MOSI/MISO/SCK برای SPI.

■ **Reset Pin**: برای راه‌اندازی مجدد (ریست) دستی برد.



<https://docs.arduino.cc/hardware>

# وبسایت مستندات سخت‌افزاری آردوینو

وبسایت رسمی مستندات سخت‌افزار آردوینو، منبعی جامع برای اطلاعات دقیق درباره تمامی بردهای آردوینو و شیلدهای رسمی آن است. این سایت توسط تیم رسمی Arduino ارائه شده و به‌روزرسانی می‌شود.

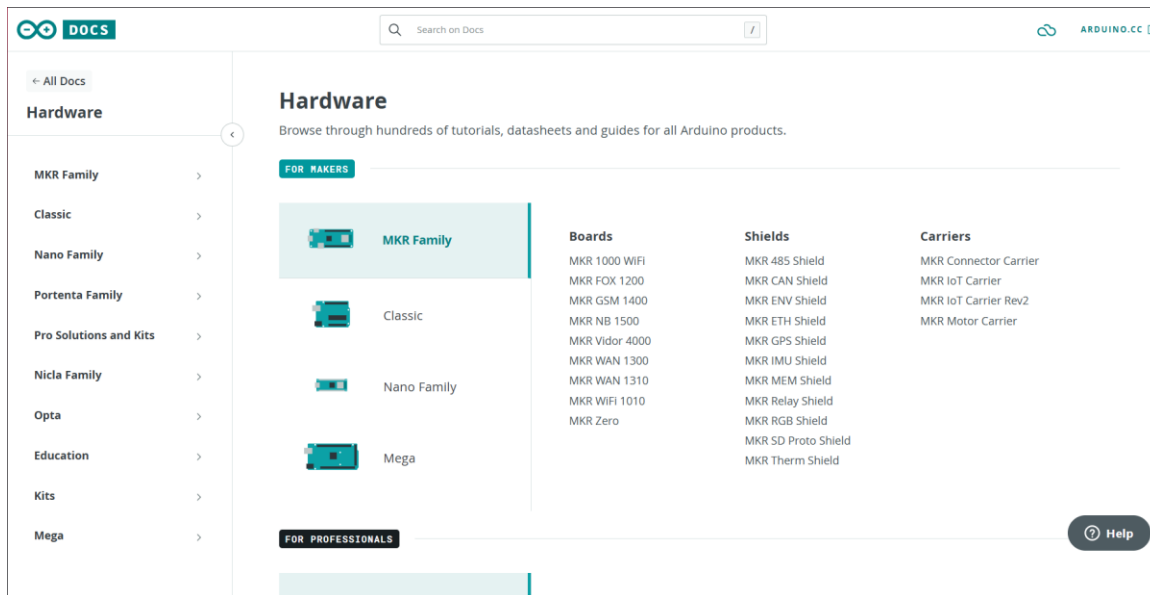
در این صفحه می‌توانید به موارد زیر دسترسی داشته باشید.

■ معرفی کامل بردهای مختلف مانند Leonardo، Mega، Nano، Uno و ...

■ دیاگرام‌ها و پین‌اوت‌های دقیق هر برد مشخصات فنی (ولتاژ، حافظه، پورت‌ها، نوع میکروکنترلر)

■ راهنمای اتصال و استفاده از شیلدها و سنسورها

■ دسترسی به فایل‌های طراحی سخت‌افزار (شماتیک، PCB، CAD)



<https://docs.arduino.cc/hardware>

# صفحه سخت افزار برد 3 ARDUINO UNO REV

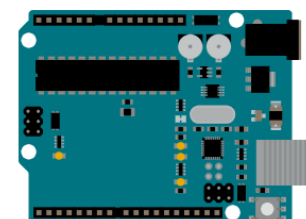
برد Arduino UNO یکی از محبوب ترین و پرکاربردترین بردهای آردوینو است که بر پایه میکروکنترلر ATmega328P ساخته شده. این برد برای یادگیری، پروژه های دانشجویی و نمونه سازی بسیار مناسب است.

در تصویر زیر صفحه رسمی سخت افزار برد Arduino UNO را میبینید که میتوان از اینجا Pinout و دیتاشیت و شماتیک PCB و فایل CAD برد را دانلود کنید.

## UNO R3

The Arduino UNO is the best board to get started with electronics and coding. If this is your first experience tinkering with the platform, the UNO is the most robust board you can start playing with. The UNO is the most used and documented board of the whole Arduino family.

GET STARTED



Buy now

### DOWNLOADABLE RESOURCES

Pinout (PDF)

Datasheet

Schematics

CAD Files

Features

Tutorials

Tech Specs

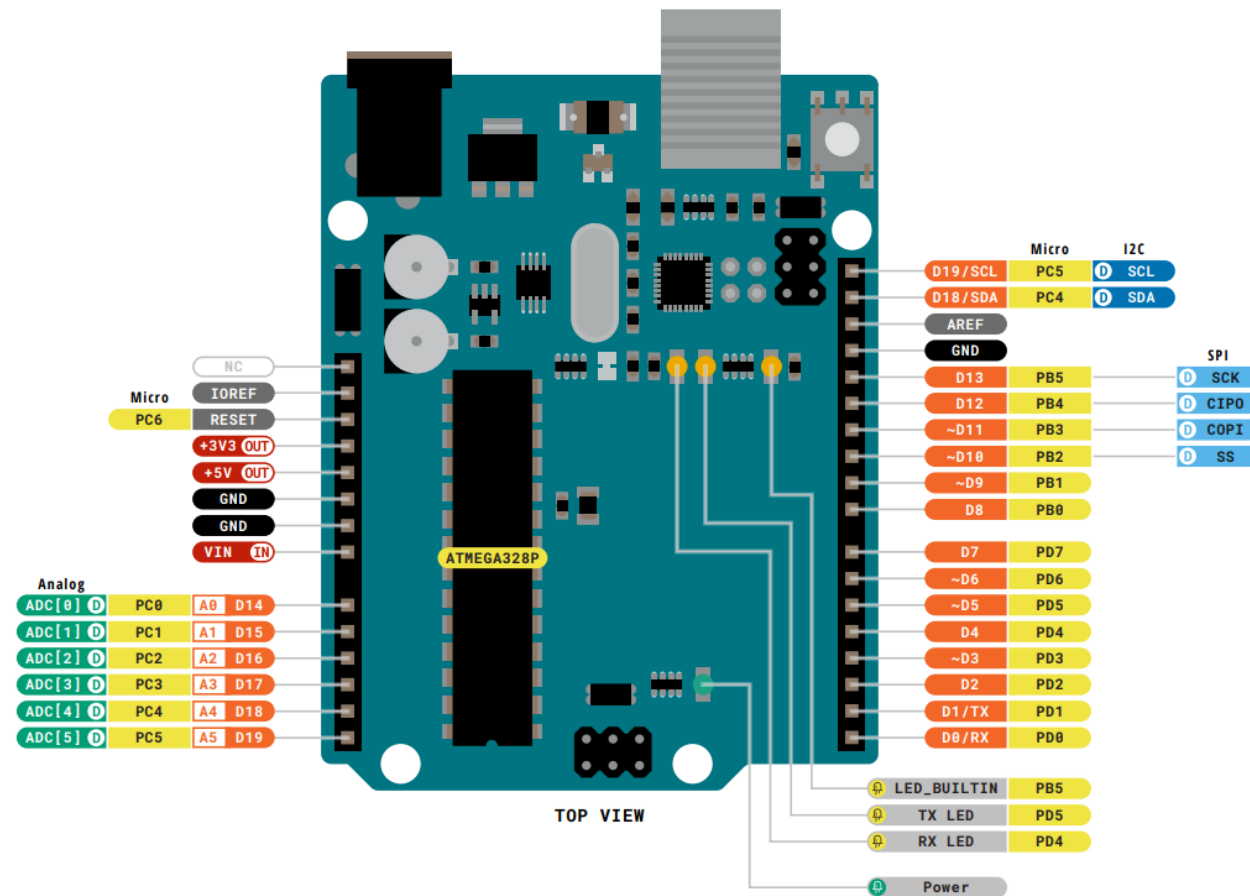
Compatibility

Suggested Libraries

Arduino UNO is a microcontroller board based on the **ATmega328P**. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. You can tinker with your UNO without worrying too much about doing something wrong, worst case scenario you can replace the chip for a few dollars and start over again.

<https://docs.arduino.cc/hardware/uno-rev3/>

# پین اوت دیاگرام ARDUINO UNO REV 3



<https://docs.arduino.cc/hardware/uno-rev3/>

# صفحه سخت افزار برد ARDUINO NANO

برد Arduino Nano نسخه‌ای کوچک‌تر از UNO است، اما با همان عملکرد. مناسب برای پروژه‌هایی با فضای محدود و مدارهای قابل حمل. مبتنی بر میکروکنترلر ATmega328P است.

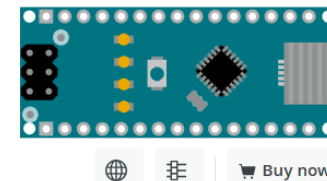
## Nano

The Arduino Nano is Arduino's classic breadboard friendly designed board with the smallest dimensions. The Arduino Nano comes with pin headers that allow for an easy attachment onto a breadboard and features a Mini-B USB connector.

GET STARTED

### DOWNLOADABLE RESOURCES

 Pinout (PDF)  Datasheet  Schematics



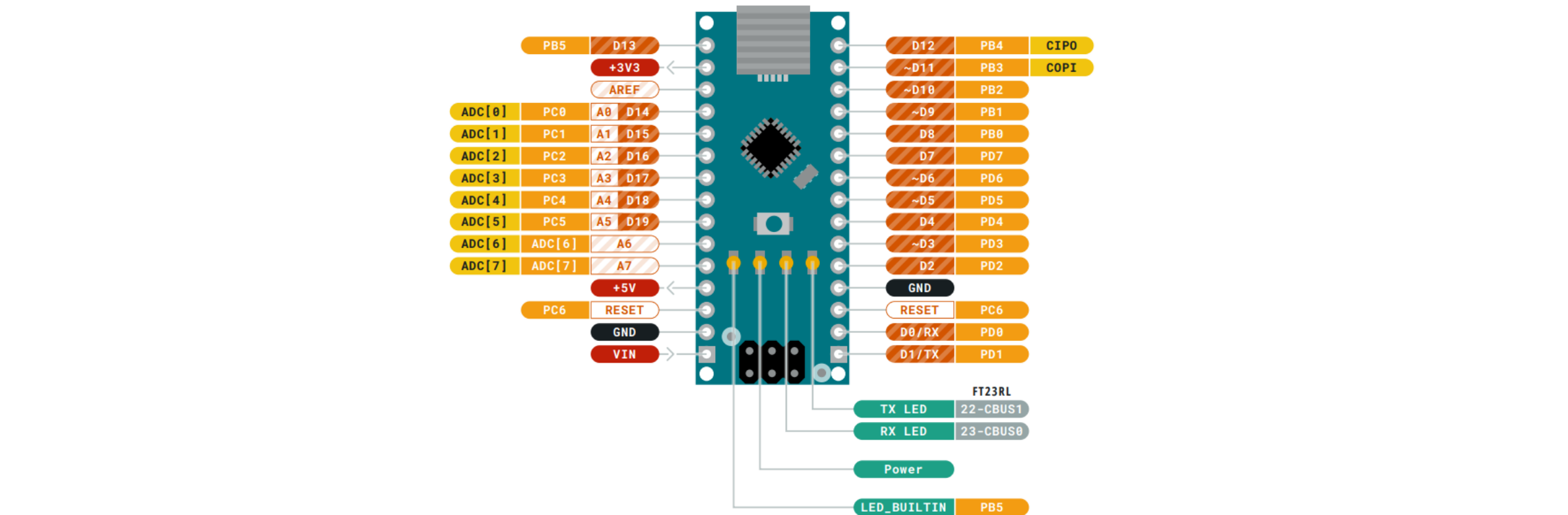
[Features](#) [Tutorials](#) [Tech Specs](#) [Compatibility](#) [Suggested Libraries](#)

The classic Nano is the oldest member of the Arduino Nano family boards. It is similar to the Arduino Duemilanove but made for the use of a breadboard and has no dedicated power jack. Successors of the classic Nano are for example the Nano 33 IoT featuring a WIFI module or the Nano 33 BLE Sense featuring Bluetooth® Low Energy and several environment sensors.

<https://docs.arduino.cc/hardware/nano/>

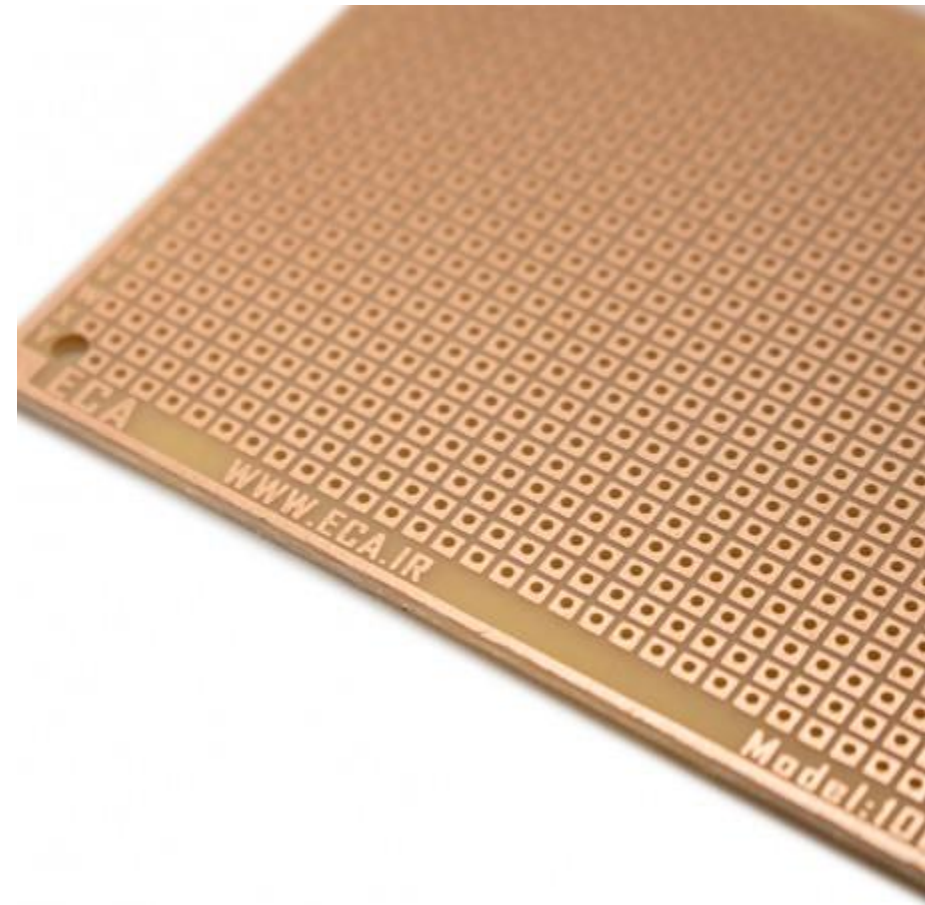
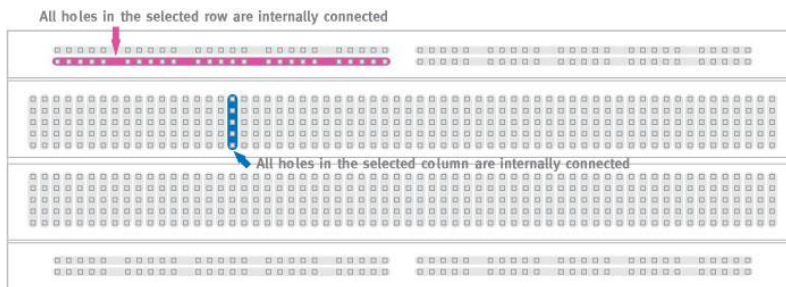


پیناوت دیاگرام ARDUINO NANO



<https://docs.arduino.cc/hardware/nano/>

# BREADBOARD & PERFBOARD



# محیط نرم افزار ARDUINO IDE

## VERIFY/UPLOAD

- دکمه‌ی بررسی (✓) برای چک کردن خطاهای کد.

- دکمه‌ی آپلود (→) برای بارگذاری برنامه روی برد آردوینو.

- **SELECT BOARD & PORT** (انتخاب نوع برد آردوینو و پورته‌ی که دستگاه به آن متصل است).

- **OPEN SERIAL MONITOR** (باز کردن Serial Monitor برای مشاهده داده‌هایی که از طریق پورت سریال ارسال می‌شوند).

- **OPEN SERIAL PLOTTER** (باز کردن نمودار سریال برای نمایش گرافیکی مقادیر دریافتی از برد).

- **SKETCHBOOK** (دسترسی به پروژه‌های ذخیره‌شده‌ی قبلی یا ایجاد یک پروژه‌ی جدید).

- **BOARD MANAGER** (مدیریت و نصب بردهای مختلف پشتیبانی‌شده در آردوینو).

- **LIBRARY MANAGER** (اضافه یا حذف کردن کتابخانه‌ها (Libraries) که کدها به آن‌ها نیاز دارند).

- **DEBUGGER** (ابزار دیباگ برای بررسی گام‌به‌گام عملکرد برنامه (در برخی بردها قابل استفاده است. متأسفانه برای آردوینو Uno و Nano این امکان وجود ندارد)

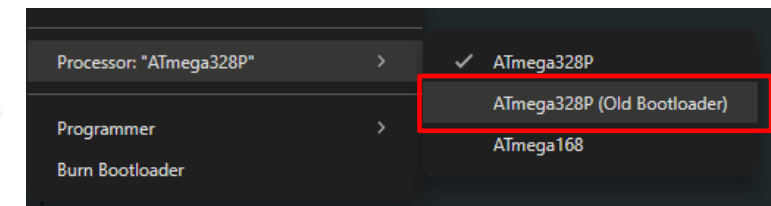
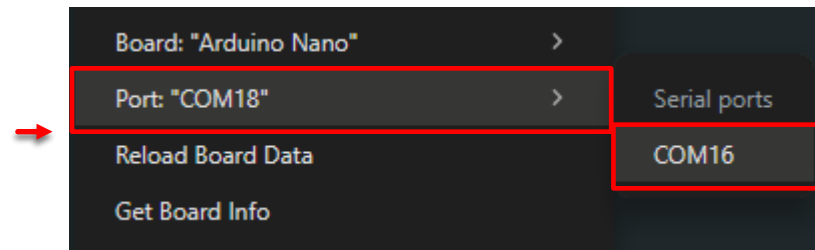
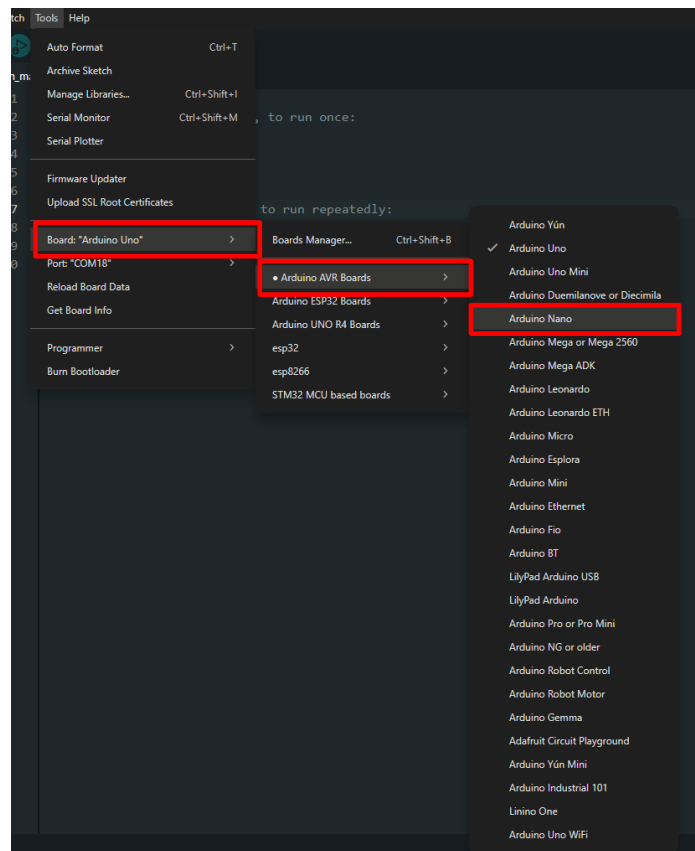
- **SEARCH** (جستجو در کدها یا در اسناد پروژه برای یافتن عبارات مشخص).



<https://www.arduino.cc/en/software/>

# نکته ای درباره پروگرام کردن ARDUINO NANO و UNO با نرم افزار ARDUINO IDE

برای پروگرام کردن آردوینو وقتی که با کابل USB به برد متصل شدید، کامپیوتر یک پورت COM مجازی برای برد شما در نظر میگیرد و شما با تعیین نوع برد (در بخش Tools → Arduino AVR Boards) و پورت آن (COM18 در این مثال)

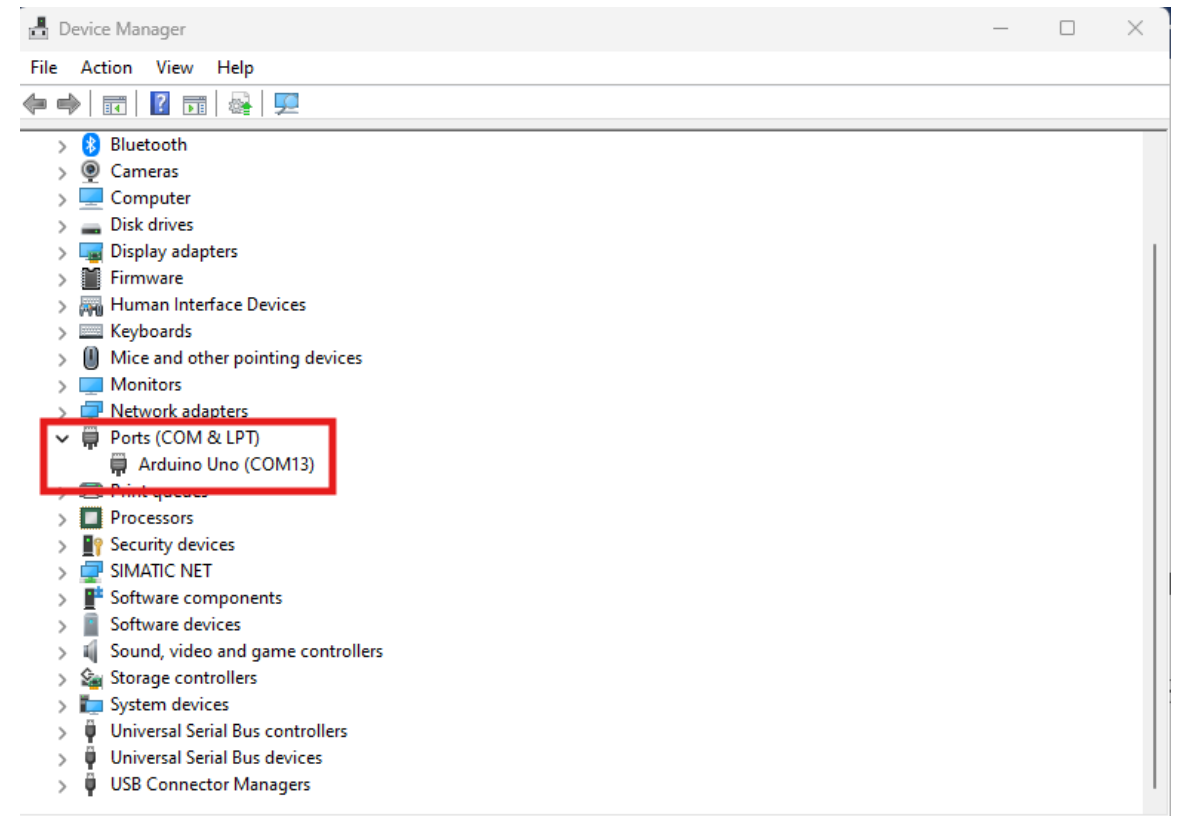
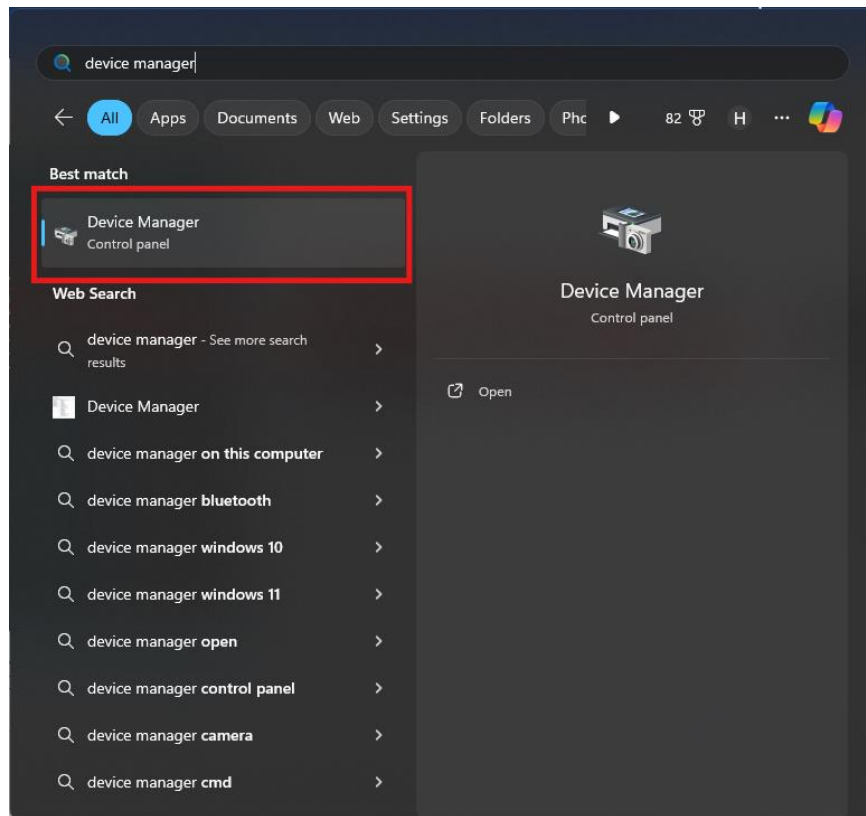


نکته این مرحله معمولاً برای آردوینو UNO نیاز نمی شود. اما به دلیل واردات برد های غیر رسمی آردوینو به بازار ایران ممکن است آردوینو نانو شما از آپشن زیر استفاده کنید.

Processor:"ATmega328P" → ATmega328p (Old Bootloader)

# یافتن عدد ست شده برای پورت آردوینو

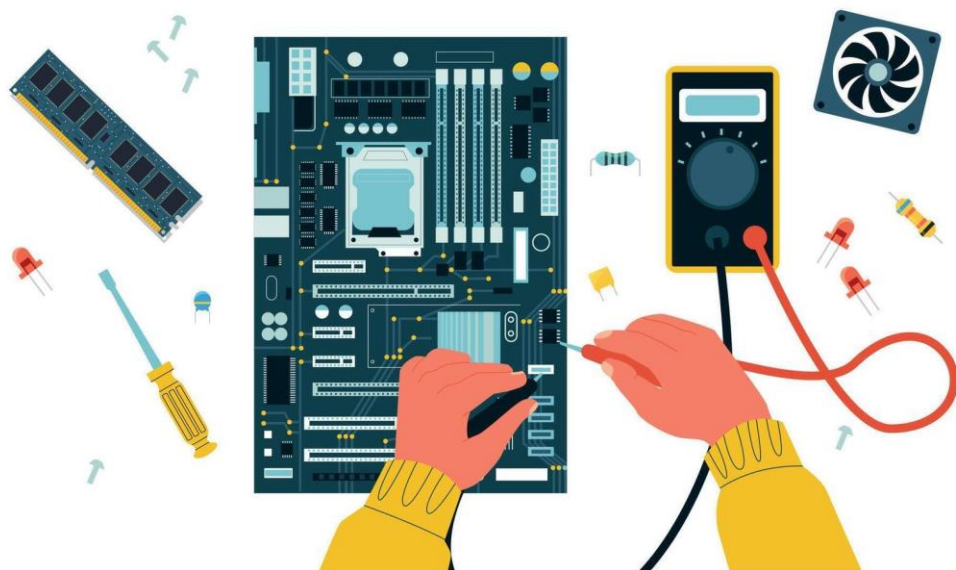
برای یافتن عدد ست شده برای پورت آردوینو شما با سرچ Device Manager در سرچ ویندوز و رفتن در بخش Ports (COM & LPT) با قطع و وصل کردن اتصال آردوینو به کامپیوتر عدد شماره پورت آردوینو خود را پیدا کنید.



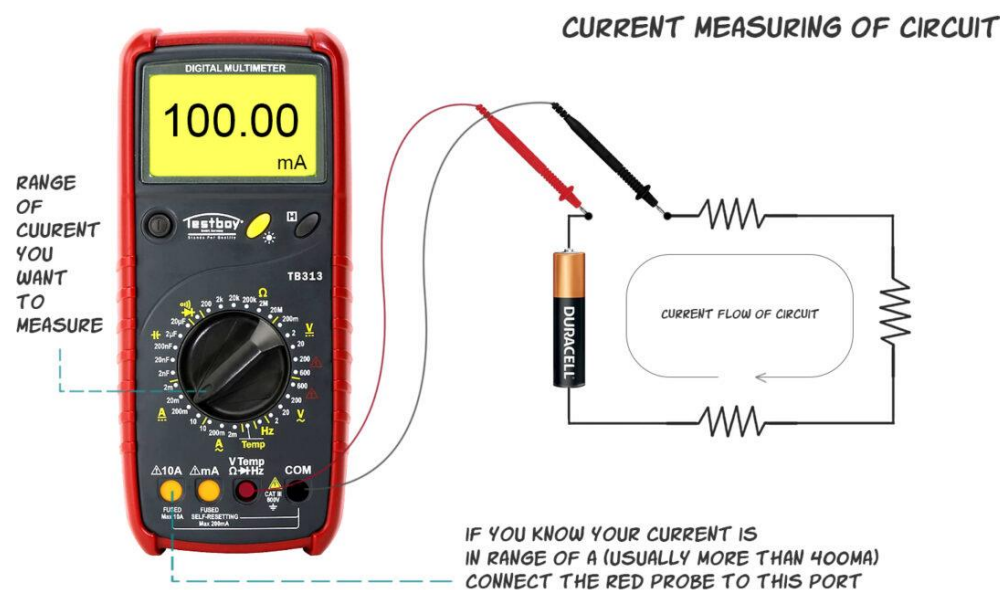


## چگونه یک مدار را دیباگ (عیب‌یابی) کنیم

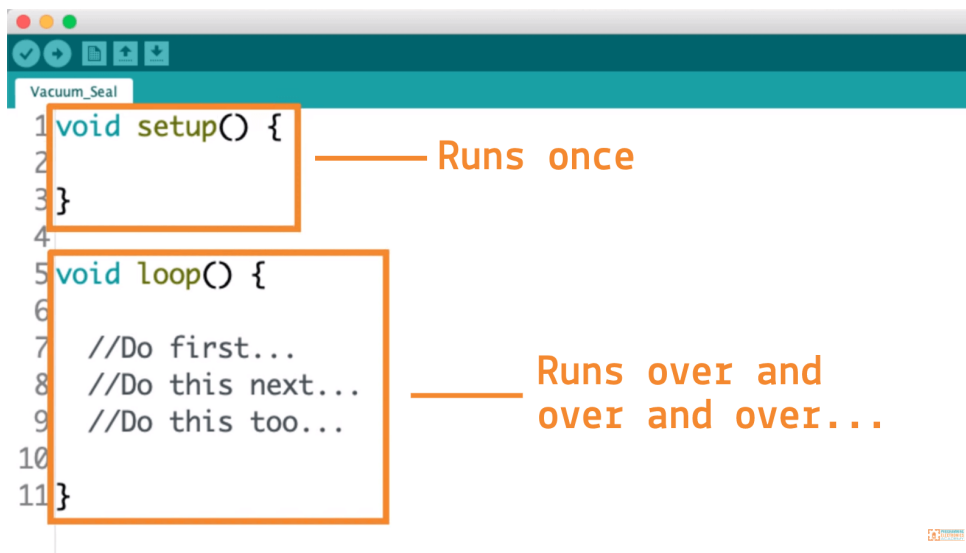
- مطمئن شوید که آردوینو برق دارد.
- مدار را از نظر باز بودن یا قطعی بررسی کنید.
- مطمئن شوید از پایه‌ی درست آردوینو استفاده می‌کنید.
- LED را برعکس کنید (چون قطبیت دارد).
- یک LED دیگر امتحان کنید.
- یک پایه‌ی دیگر آردوینو را امتحان کنید (فراموش نکنید که کدتان را هم تغییر دهید).
- مراقب اتصال کوتاه باشید (اگر مدار داغ شد یا بوی سوختگی آمد، برق را قطع کنید).
- اگر مدار به‌صورت متناوب کار می‌کند، سیم‌های شل را بررسی کنید.



## اشاره ای به مولتی متر



# آشنایی با توابع SETUP و LOOP در برنامه‌نویسی آردوینو



بیشتر بردهای آردوینو به گونه‌ای طراحی شده‌اند که تنها یک برنامه روی میکروکنترلر آنها اجرا شود. این برنامه می‌تواند برای انجام یک عمل ساده مانند چشمک زدن یک LED نوشته شده باشد، یا می‌تواند صدها عمل را به صورت چرخه‌ای اجرا کند. دامنه عملکرد برنامه‌ها بسته به هدف آنها متفاوت است.

برنامه‌ای که روی میکروکنترلر بارگذاری می‌شود، به محض روشن شدن، شروع به اجرا خواهد کرد. هر برنامه دارای تابعی به نام `void setup` و `void loop` است.

تابع `setup` تنها یک بار، در ابتدای اجرای برنامه و پس از روشن شدن یا ریست شدن میکروکنترلر اجرا می‌شود. این تابع برای انجام تنظیمات اولیه استفاده می‌شود.

```
// the setup function runs once when you press reset or power the board  
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
// the loop function runs over and over again forever  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)  
  delay(1000); // wait for a second  
  digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW  
  delay(1000); // wait for a second  
}
```

پس از اجرای کامل تابع `setup`، تابع `loop` به طور مداوم و پی‌درپی اجرا می‌شود و منطق اصلی برنامه در آن قرار دارد.

# آشنایی با توابع SETUP و LOOP در برنامه‌نویسی آردوینو

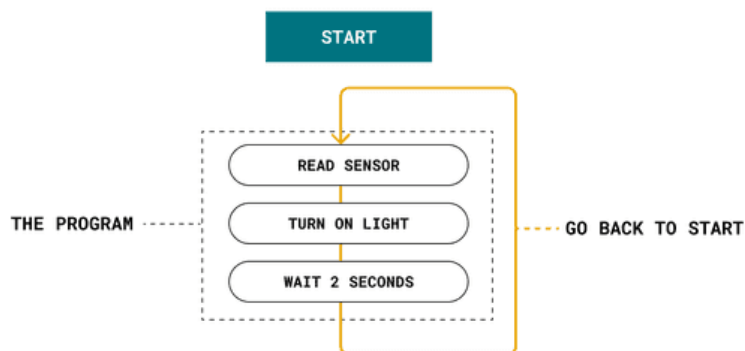
درون تابع **setup**، می‌توانید برای مثال:

- تعیین پین‌ها به عنوان ورودی یا خروجی
- مقداردهی اولیه به متغیرها یا کتابخانه‌ها
- راه‌اندازی ارتباط سریال یا سنسورها

درون تابع **loop**، می‌توانید برای مثال:

- یک سنسور را بخوانید.
- یک چراغ را روشن کنید.
- بررسی کنید که آیا یک شرط برقرار است یا نه.
- یا همه موارد بالا را انجام دهید.

سرعت اجرای برنامه بسیار بالاست، مگر اینکه به آن بگوییم که آهسته‌تر اجرا شود. سرعت اجرا به اندازه برنامه و زمانی که اجرای آن توسط میکروکنترلر طول می‌کشد بستگی دارد، اما معمولاً در حد میکروثانیه (یک میلیونم ثانیه) است.



# ورودی/خروجی دیجیتال DIGITAL I/O چیست؟

پایه‌های دیجیتال در آردوینو فقط دو حالت دارند:

روشن (HIGH) یا خاموش (LOW)

کاربردها:

- روشن/خاموش کردن LED

- خواندن وضعیت یک دکمه

دستورات رایج:

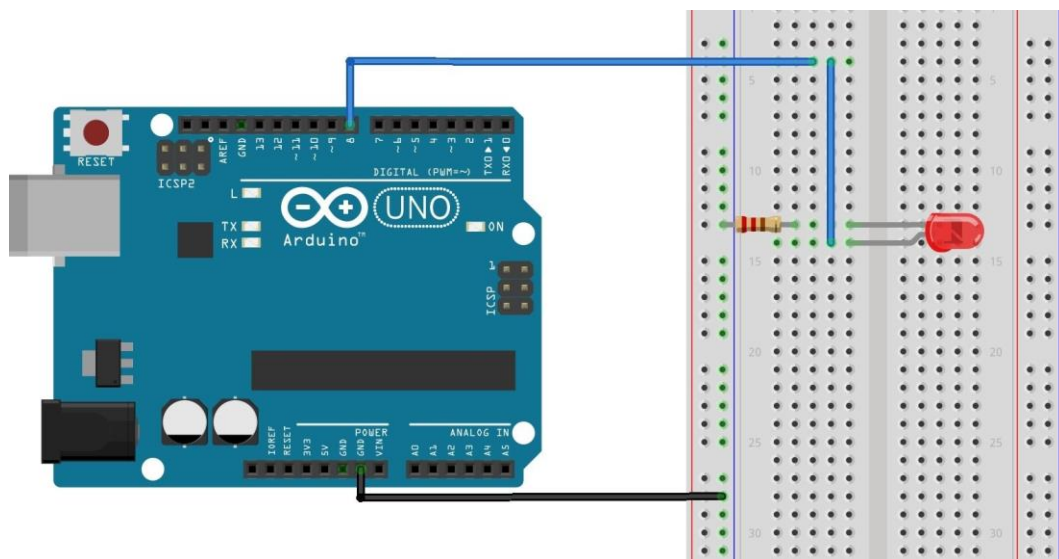
- تنظیم پین به صورت خروجی ← `pinMode(pin, OUTPUT);`

- تنظیم پین به صورت ورودی ← `pinMode(pin, INPUT);`

- ارسال سیگنال ۵ ولت یا روشن کردن ← `digitalWrite(pin, HIGH);`

- ارسال سیگنال ۰ ولت یا خاموش کردن ← `digitalWrite(pin, LOW);`

- خواندن وضعیت از ورودی دیجیتال ← `digitalRead(pin);`





# ورودی/خروجی دیجیتال DIGITAL I/O چیست؟

پایه‌های دیجیتال در آردوینو فقط دو حالت دارند:

روشن (HIGH) یا خاموش (LOW)

کاربردها:

- روشن/خاموش کردن LED

- خواندن وضعیت یک دکمه

دستورات رایج:

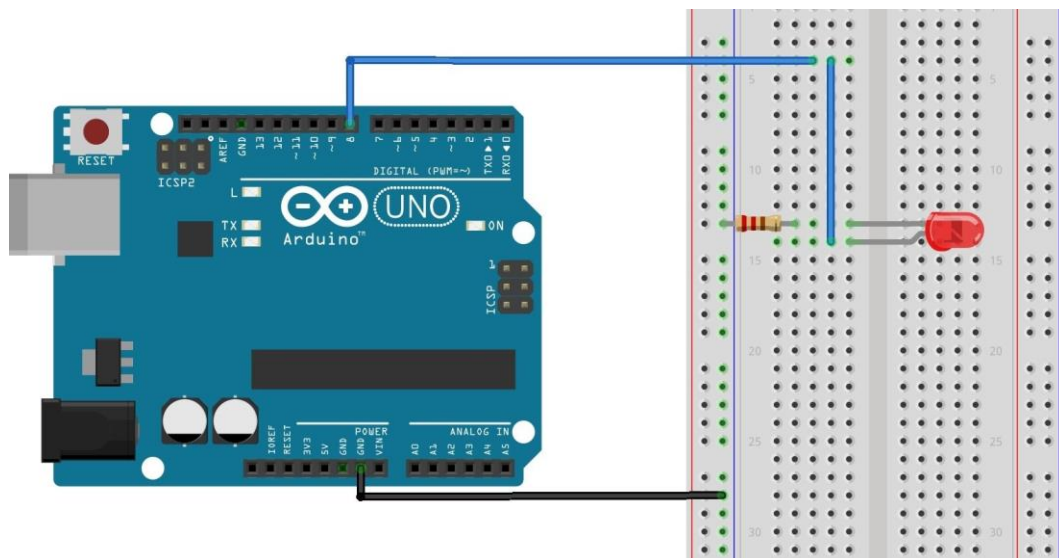
- تنظیم پین به صورت خروجی ← `pinMode(pin, OUTPUT);`

- تنظیم پین به صورت ورودی ← `pinMode(pin, INPUT);`

- ارسال سیگنال ۵ ولت یا روشن کردن ← `digitalWrite(pin, HIGH);`

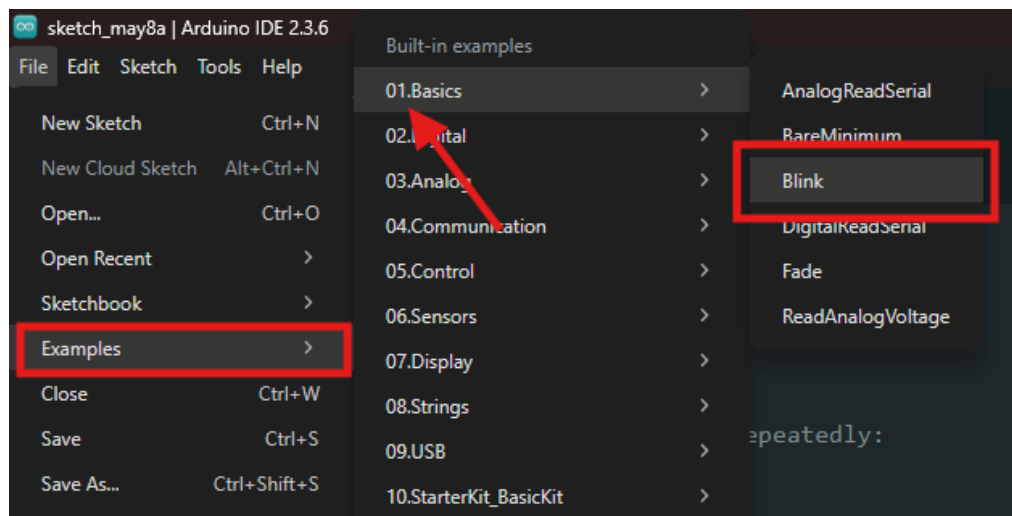
- ارسال سیگنال ۰ ولت یا خاموش کردن ← `digitalWrite(pin, LOW);`

- خواندن وضعیت از ورودی دیجیتال ← `digitalRead(pin);`



## مثال ساده DIGITAL I/O (اولین مثال آردوینو، مثال چشمک زن)

نرم افزار آردوینو در خود مثال های متعددی دارد که در اینجا مثال Blink یا چشمک زن که یکی از ساده ترین مثال های آردوینو هست را مشاهده میکنید.



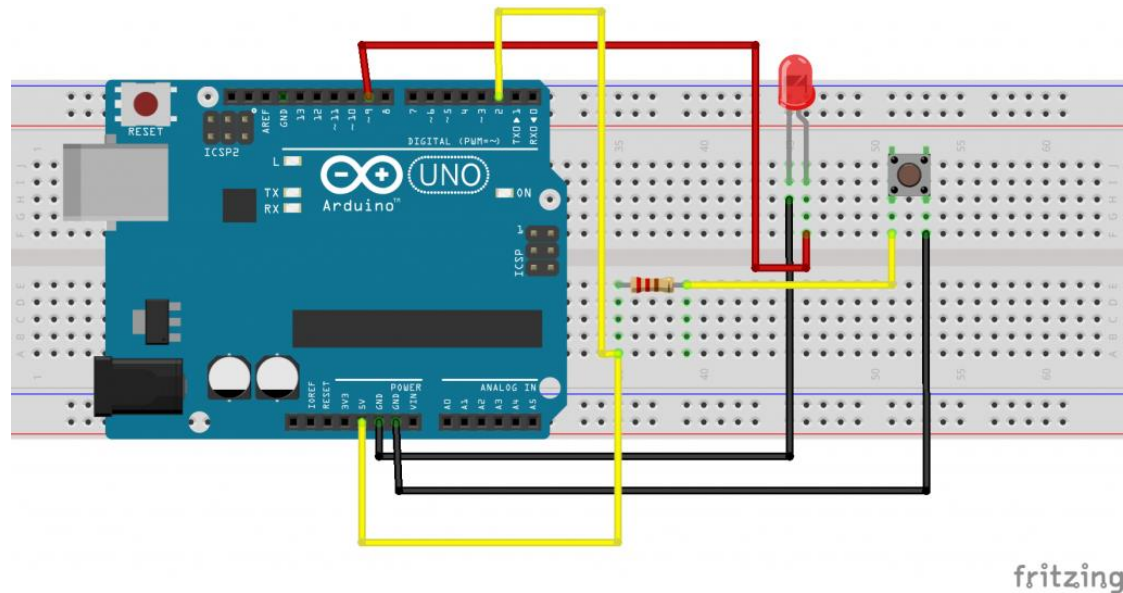
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);                     // wait for a second
  digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);                     // wait for a second
}
```

تمرین: برنامه ای بنویسید که ۲ ثانیه LED روشن باشد و ۰.۵ ثانیه خاموش

## مثال ساده DIGITAL I/O (روشن کردن چراغ با استفاده از دکمه)

هدف: روشن کردن یک LED با فشار دادن یک دکمه



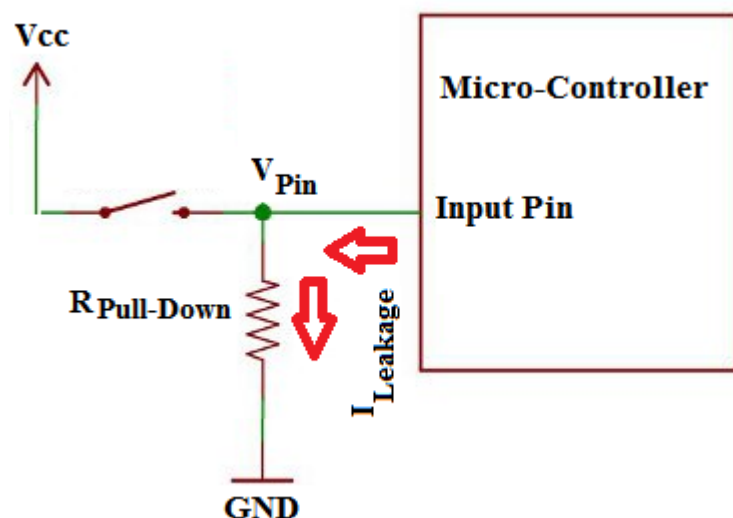
```
void setup() {  
  pinMode(2, INPUT);  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  if (digitalRead(2) == HIGH) {  
    digitalWrite(13, HIGH);  
  } else {  
    digitalWrite(13, LOW);  
  }  
}
```

تمرین:

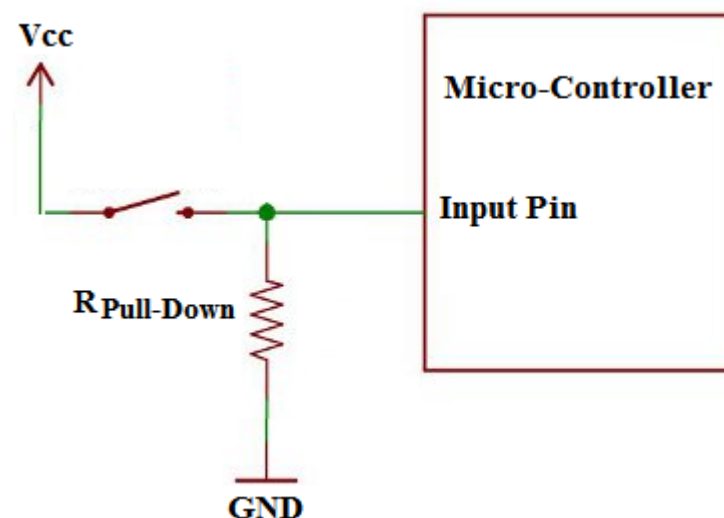
- برنامه ای بنویسید که در حالت عادی LED روشن باشد و وقتی دکمه فشرده شد LED خاموش شود.
- دو LED را به آردوینو متصل کنید و برنامه ای بنویسید که با زدن دکمه یکی از وضعیت روشن به خاموش و دیگری از خاموش به روشن شود.

## مقاومت های PULL UP و PULL DOWN

فرض کنید که در حال استفاده از یک میکروکنترلر هستید و یک پین خاص آن را به صورت ورودی تعریف کرده باشید. در صورتی که هیچ مقاومت Pull Up یا Pull Down بر روی پین ورودی وجود نداشته باشد و پین توسط شما به GND و یا Vcc متصل نباشد، مقدار دیجیتال این پین که توسط میکروکنترلر خوانده می شود تحت تاثیر نویز و وضعیت پایه های کناری و غیره قرار می گیرد و به طور کلی مقدار خوانده شده قابل پیش بینی نیست. معمولاً میکروکنترلرها امکان اتصال مقاومت Pull Up و یا Pull Down به صورت داخلی را برای کاربر فراهم می کنند (در آردوینو Pull Up) که حجم مدار خارجی را کم و مشکلی که در بالا ذکر شد را بر طرف می نمایند. اما فرض کنید که پین به صورت ورودی تعریف شده و هیچ مقاومت بالاکش و یا پایین کش به آن وصل نشده و اصطلاحاً به صورت شناور (Floating) است.



اثر جریان نشتی پین و مقاومت Pull Down بسیار بزرگ



پین ورودی با مقاومت Pull Down خارجی

<https://roboeq.ir/blog/pull-up-and-pull-down-resistors/>

# PSEUDOCODE & FLOWCHART

## ■ Pseudocode:

Start program

Set LED pin as output

Set button pin as input

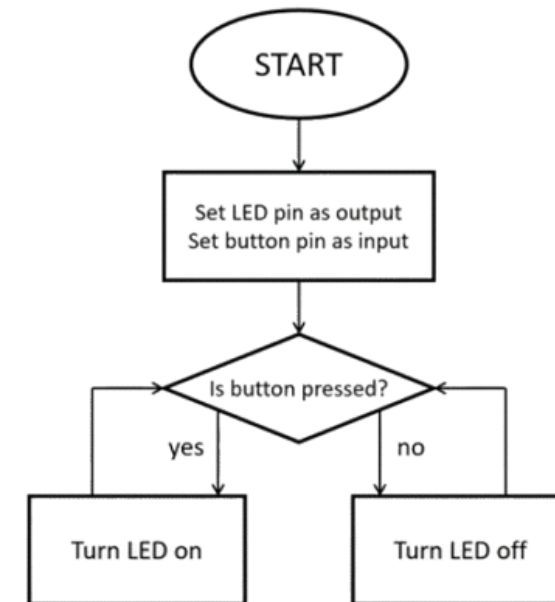
Infinite loop:

    Is the button pressed?

        If yes, turn on the LED

        If no, turn off the LED

Flowchart:

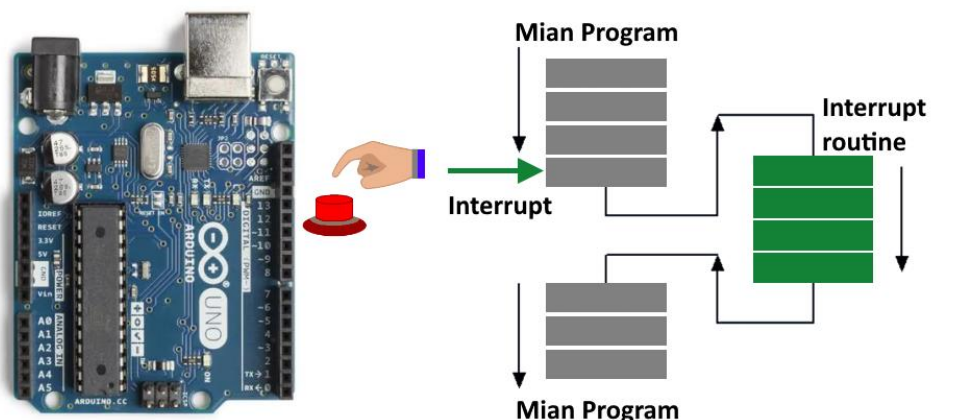




## مثال اضافه!!

مثال اضافه! دو LED را به آردوینو متصل کنید و یکی را بلینکر بنویسید و دیگری به نحوی باشد که با کلید روشن شود. (اشاره ای به وقفه در آردوینو)

این مثال جز سر فصل این دوره نیست اما خوب هست خودتان درمورد وقفه (Interrupt Service Routine) و نوشتن تابعی غیر از توابع `loop` و `setup` مطالعه کنید.



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

  DashedLine(); ← Function is called here
  Serial.println("| Program Menu |");
  DashedLine(); ← Function is called again
}

void loop() {
}

void DashedLine()
{
  Serial.println("- - - - -"); } Function is
                                     created here
```

```
bool led3State = false;

void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, INPUT_PULLUP);
  attachInterrupt(digitalPinToInterrupt(4), toggleLED3, FALLING);
}

void loop() {
  digitalWrite(2, HIGH);
  delay(500);
  digitalWrite(2, LOW);
  delay(500);

  digitalWrite(3, led3State);
}

// Interrupt Service Routine (ISR)
void toggleLED3() {
  led3State = !led3State;
}
```

# ورودی آنالوگ ANALOG INPUT

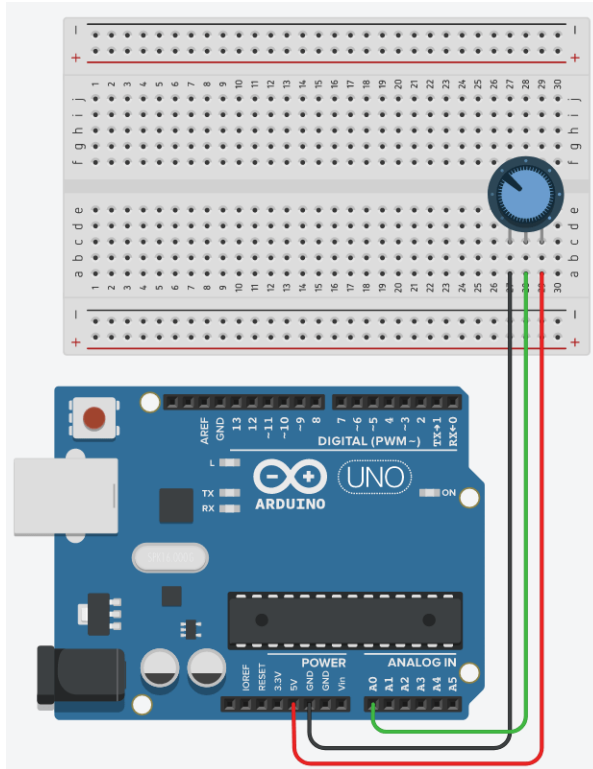
بعضی حسگرها مثل پتانسیومتر یا سنسور دما، خروجی پیوسته دارند

بین‌های A0 تا A5 در آردوینو برای ورودی آنالوگ هستند.

مقدار ورودی بین ۰ تا ۵ ولت خوانده می‌شود و به عددی بین ۰ تا ۱۰۲۳ تبدیل می‌شود (توسط ADC - مبدل آنالوگ به دیجیتال)

دستور لازم برای ورودی آنالوگ:

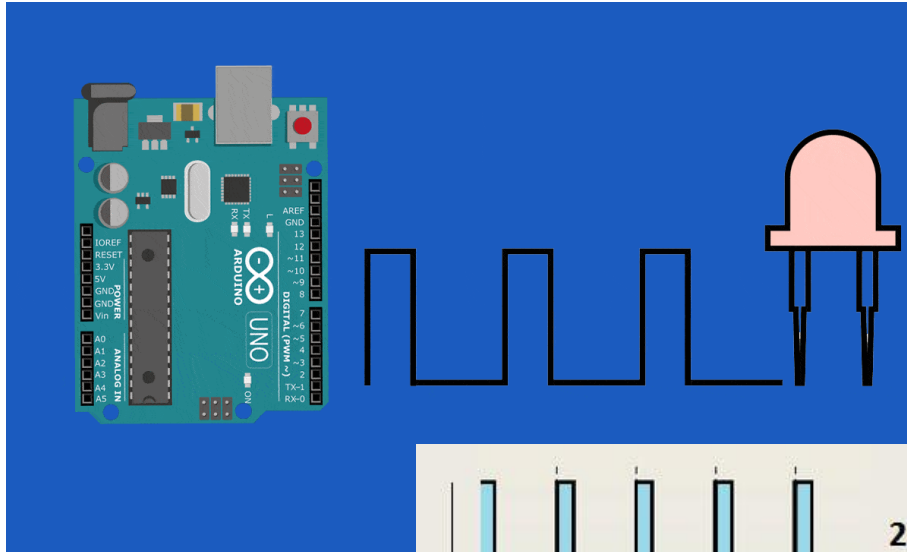
■ خواندن بین A0 آردوینو ← `analogRead(A0);`



```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  delay(200);  
}
```

مثال خواندن یک ورودی آنالوگ (توسط پتانسیومتر)

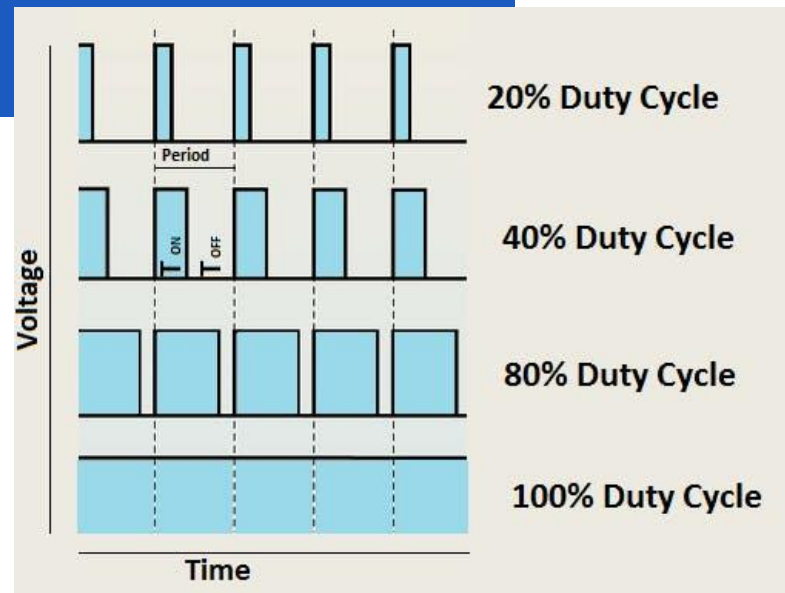
# خروجی شبه آنالوگ (PWM) ANALOG OUTPUT



آردوینو خروجی آنالوگ واقعی ندارد، ولی می‌تواند با PWM خروجی شبه آنالوگ تولید کند کاربرد: کنترل روشنایی LED، سرعت موتور

توجه شود در آردوینو UNO فقط پین های ۳، ۵، ۶، ۹، ۱۰ و ۱۱ (پین های دارای علامت ~ در کنار شماره) دیجیتال چنین قابلیت دارند  
دستورات لازم برای خروجی آنالوگ:

- تنظیم پین به صورت خروجی ← `pinMode(9, OUTPUT);`
- نوشتن مقدار بر روی پین ← `analogWrite(9, pwmValue);`



## مثال کنترل روشنایی LED با PWM (دایمر)

در این مثال، روشنایی یک LED با چرخاندن پتانسیومتر تغییر می‌کند. مقدار خوانده‌شده از A0 به مقدار PWM بین ۰ تا ۲۵۵ تبدیل می‌شود و به پین خروجی داده می‌شود.

```
void setup() {  
  pinMode(9, OUTPUT);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  int pwmValue = map(sensorValue, 0, 1023, 0, 255);  
  analogWrite(9, pwmValue);  
  delay(10);  
}
```

تابع map تبدیل یک عدد از یک بازه (range) به عددی متناظر در بازه‌ای دیگر

به فرض در این مثال: تبدیل مقدار خوانده‌شده از سنسور (۰ تا ۱۰۲۳) به مقدار PWM (۰ تا ۲۵۵)

# راه های ارتباطی و لینک های مربوط به دوره

در صورت وجود هرگونه ابهام و مشکل در حین دوره میتوانید از راه های زیر با بنده در ارتباط باشید:

[theheidari@gmail.com](mailto:theheidari@gmail.com)

در تلگرام و ایتا [@xheidari](https://t.me/xheidari)

<https://www.linkedin.com/in/xheidari/>

در ضمن تمامی محتوای ارائه شده دوره به تدریج در گروه تلگرامی و لینک گیت هاب دوره آپلود میشود:

<https://github.com/xheidari/ArduinoCourse>

شبیه سازی ها در سایت Tinkercad انجام خواهد شد و در لینک زیر شبیه سازی های انجام شده در کلاس قرار میگیرد:

<https://www.tinkercad.com/joinclass/KD54P7ADW>

با تشکر از توجه شما!