

اتوماسیون

دکتر پدرام پیوندی

آردوینو (Arduino) چیست ؟

- همانطور که قبلا گفتیم ما برای برنامه نویسی نیاز به یک سخت افزای به نام پردازشگر (CPU) داریم که فرامین ما را محاسبه و اجرا کند



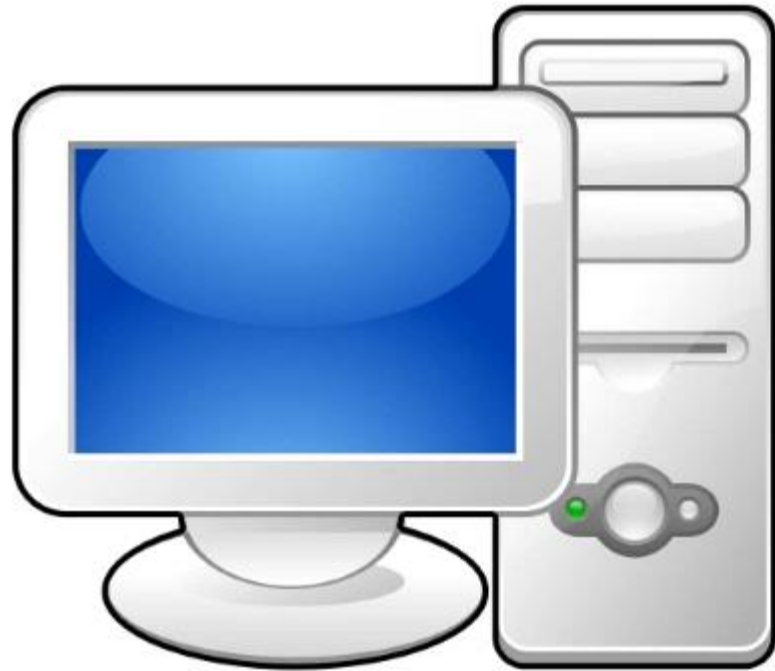
آردوینو (Arduino) چیست ؟

- حال فرض کنیم اگر بخواهیم یک ماشین لباس شویی را برنامه ریزی و کنترل کنیم، آیا منطقی است که از CPU کامپیوتر و امثال آن استفاده شود؟؟



آردوینو (Arduino) چیست ؟

- در این هنگام دانشمندان به فکر ساخت وسیله ای افتادند که همانند یک کامپیوتر، قابلیت های آن را داشته باشد اما قدرت پردازش آن کم باشد و در نتیجه قیمت آن بسیار ارزان و مصرف انرژی آن بسیار پایین خواهد بود و اینگونه شد که میکروکنترلرها، ساخته شد...



آردوینو (Arduino) چیست ؟

- آردوینو یک برد آزمایشی توسعه پذیر (Development Board) است، که در هسته آن یک میکروکنترلر مدل ATMEGA AVR Atmega328p قرار گرفته است.



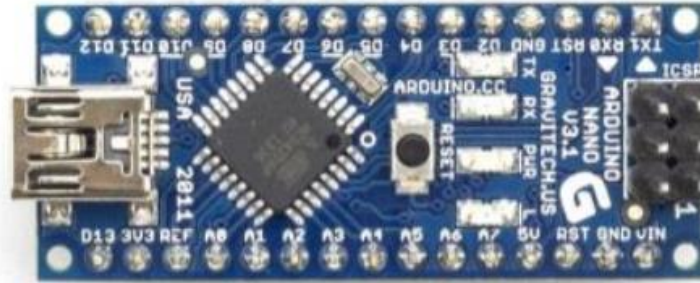
آردوینو (Arduino) چیست ؟

• آردوینو چه مزیت هایی دارد ؟

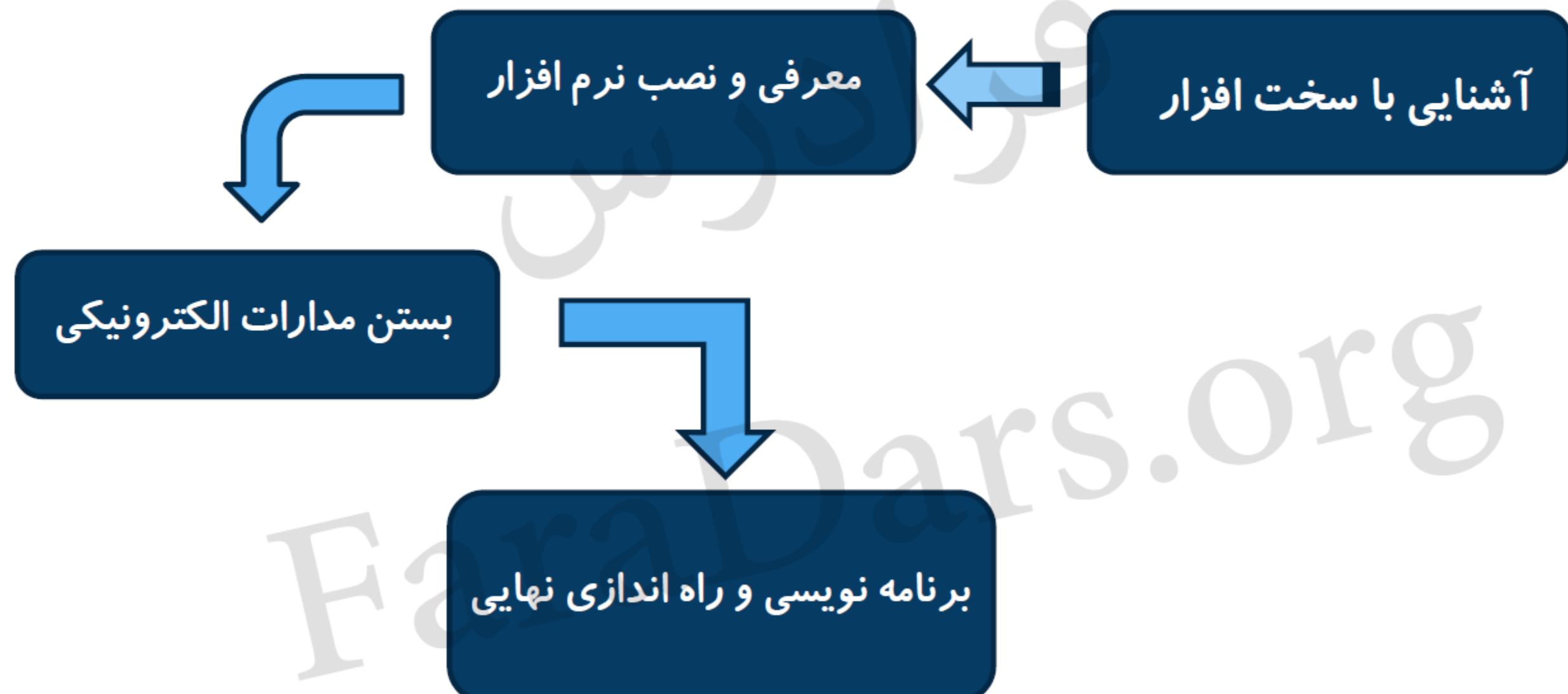
- ۱- برای استفاده از آن نیازی به لحیم کاری ندارید
 - ۲- یک نرم افزار کاملاً رایگان و متن باز (open Source) به همراه سخت افزار متن باز دارید که تمامی اطلاعات آن موجود و قابل اشتراک است
 - ۳- زبان برنامه نویسی در آن، به صورت سطح بالا (به زبان انسان نزدیک بوده) و برنامه نویسی آن با استفاده از نرم افزار مخصوص آردوینو (Arduino IDE) به مراتب ساده تر از دیگر کامپایلر های موجود است
 - ۴- نیاز به سخت افزار جانبی برای پروگرام کردن برد خود ندارید. (به دلیل وجود رابط USB)
 - ۵- دارا بودن کتابخانه های غنی و کدهای رایگان و آموزش های متنوع در سطح بین المللی
- و



انواع مختلف آردوینو



شرح انجام کار



معرفی ساختار زبان C/C++ برای برنامه نویسی

- Void setup() {

دستورات برنامه نویسی
}

- این تابع (Function) برای آماده سازی برنامه به کار می رود، و تمامی دستورات فقط یک بار در آن اجرا می شود
- برای مثال دستوراتی مانند pinMode و تنظیمات Serial در این قسمت است

- Void loop(){

دستورات برنامه نویسی
}

- این تابع تابع خطوط برنامه را یکی پس از دیگری اجرا می کند، و دائماً برنامه را تکرار می کند
- هسته ی اصلی برنامه در آردوینو، برنامه نویسی در این تابع می باشد!!

معرفی ساختار زبان C/C++ برای برنامه نویسی

- بله !!! ما می توانیم به تعداد دلخواه تابع در برنامه خود ایجاد کنیم

```
int sensor() {  
    int sensorValue = analogRead(A0);  
    sensorValue /= 4; // مقدار ۱۰۲۳-۰ را به ۲۵۵-۰ تبدیل می کند  
    return sensorValue; // مقدار نهایی sensorValue را به برنامه اصلی باز می گرداند  
}
```

معرفی ساختار زبان C/C++ برای برنامه نویسی

- گذاشتن چند خط توضیحات در برنامه

```
/*  
توضیحات  
*/
```

- گذاشتن توضیحات فقط در یک خط

```
// توضیح در یک خط
```

معرفی ساختار زبان C/C++ برای برنامه نویسی

• انواع متغیرها

<i>Keyword</i>	<i>Variable Type</i>	<i>Range</i>
char	Character (or string)	-128 to 127
int	Integer	-32,768 to 32,767
short short int	Short integer	-32,768 to 32,767
long	Long integer	-2,147,483,648 to 2,147,483,647
unsigned char	Unsigned character	0 to 255
unsigned int	Unsigned integer	0 to 65,535
unsigned short	Unsigned short integer	0 to 65,535
unsigned long	Unsigned long integer	0 to 4,294,967,295
float	Single-precision floating point (accurate to 7 digits)	$\pm 3.4 \times 10^{-38}$ to $\pm 3.4 \times 10^{38}$
double	Double-precision floating point (accurate to 15 digits)	$\pm 1.7 \times 10^{-308}$ to $\pm 1.7 \times 10^{308}$

معرفی ساختار زبان C/C++ برای برنامه نویسی

- انواع متغیرها
آرایه (Arrays):

```
int MyArray[] = {متغیر ۰, متغیر ۱, متغیر ۲, ... }
```

برای خواندن مقدار از یک آرایه به شکل زیر عمل می کنیم:

Example → Value = MyArray[2];

معرفی ساختار زبان C/C++ برای برنامه نویسی

• عملیات ریاضی

$X++ \rightarrow X=X+1$

$X-- \rightarrow X=X-1$

$X+=Y \rightarrow X=X+Y$

$X-=Y \rightarrow X=X-Y$

$X*=Y \rightarrow X=X*Y$

$X/=Y \rightarrow X=X/Y$

$X \% Y \rightarrow$ باقی مانده تقسیم

معرفی ساختار زبان C/C++ برای برنامه نویسی

- عملیات مقایسه ای

$X == Y \rightarrow$ مساوی

$X != Y \rightarrow$ نامساوی

$X < Y \rightarrow$ X کوچکتر از Y

$X > Y \rightarrow$ X بزرگتر از Y

$X <= Y \rightarrow$ X کوچکتر مساوی Y

$X >= Y \rightarrow$ X بزرگتر مساوی Y

$X \&\& Y \rightarrow$ AND منطقی

$X || Y \rightarrow$ Or منطقی

$!X \rightarrow$ NOT منطقی

معرفی دستورات زبان C/C++ ویژه آردوینو

- HIGH / LOW

این دستورات به طور کلی برای نشان دادن منطق ۱ یا صفر است. زمانی منطق یک برقرار است که سطح ولتاژ در ۵ ولت و زمانی منطق ۰ (صفر) برقرار است که سطح ولتاژ در صفر ولت باشد.

- Example → `digitalWrite(13, HIGH);` → یعنی پایه شماره ۱۳ را به سطح منطقی یک برسان (۵ ولت شود)

معرفی دستورات زبان C/C++ ویژه آردوینو

- INPUT / OUTPUT
- غالباً همراه با دستور pinMode() می آید، و برای نشان دادن ورودی یا خروجی بودن یک پایه مورد نظر است
- Example → pinMode(13, OUTPUT);

Dars.org

معرفی دستورات زبان C/C++ ویژه آردوینو

- `digitalWrite(پایه, HIGH یا LOW);`
 - تغییر وضعیت منطقی یک پایه، از یک به صفر و یا بالعکس
- `digitalRead(پایه)`
 - خواندن مقدار یک پایه به صورت منطقی (فقط صفر و یک)
- Example → `value = digitalRead(12);`

معرفی دستورات زبان C/C++ ویژه آردوینو

- `analogRead` (نام پایه آنالوگ)
 - خواندن یک مقدار به صورت آنالوگ از پایه های آنالوگ با دقت ۱۰ بیت (یعنی ۰ تا ۱۰۲۳)
 - نکته: پایه های آنالوگ برخلاف دیجیتال نیازی به تعریف کردن به صورت `pinMode()` ندارد

Example → `value = analogRead(A0);`

معرفی دستورات زبان C/C++ ویژه آردوینو

- `analogWrite( PWM نام پایه های ۲۵۵ , )`; مقدار صفر تا ۲۵۵
- تولید پالس های مربعی PWM با duty Cycle های مختلف

Example →

```
void loop(){  
  int value = analogRead(A0); // خواندن مقدار صفر تا ۱۰۲۳ از پتانسیومتر  
  value /=4 ; // تقسیم ۱۰۲۳-۰ به ۲۵۵-۰  
  analogWrite(13, value); // اعمال پالس به پایه شماره ۱۳  
}
```

معرفی دستورات زبان C/C++ ویژه آردوینو

- `delay()` (بر حسب میلی ثانیه);

معرفی دستورات زبان C/C++ ویژه آردوینو

- `Serial.begin()` (نرخ تبادل اطلاعات);
- 300, 600, 1200, 2400, 4800, **9600**, 14400, 19200, 28800, 38400, 57600, 115200
- نکته: دستور `Serial.begin()` باید در تابع `{ } void setup()` بیاید

معرفی دستورات زبان C/C++ ویژه آردوینو

- `Serial.println()` (اطلاعات جهت نمایش در پورت سریال);
 - این دستور جهت نمایش اطلاعات و نوشته های شما، بر روی پورت سریال مورد استفاده قرار می گیرد.
 - **Serial.print()** در آن است که دستورات را در خط جدید اجرا کرده و در **Serial Monitor** خروجی را بهتر می توان مشاهده کرد
- **Example** →
- `Serial.println("Faradars.org");` // چاپ و نمایش فقط نوشته
- `Serial.println(SensorValue);` // چاپ و نمایش مقدار سنسور

معرفی ساختارهای شرطی در C/C++ و آردوینو

- ساختار شرطی if-else

```
if ( شرط برقرار شد )  
دستوراتی که باید اجرا شود  
{
```

```
→ در غیر این صورت else{  
دستورات دیگری که باید اجرا شود  
}
```

معرفی ساختارهای شرطی در C/C++ و آردوینو

- ساختار شرطی if-else 2

```
{ ( شرط اول برقرار شد ) if  
دستوراتی که باید اجرا شود  
}
```

```
{ ( شرط دوم برقرار شد ) else if  
دستوراتی که باید اجرا شود  
}
```

```
else { → در غیر این صورت  
دستورات دیگری که باید اجرا شود  
}
```

معرفی ساختارهای شرطی در C/C++ و آردوینو

- ساختار حلقه For

For ((میزان گام ; شرط ; مقدار اولیه) {
 دستوراتی که باید اجرا شود
}

Example →

```
for (int i=0; i<10; i++){  
digitalWrite(led,HIGH);  
delay(1000);  
digitalWrite(led,LOW);  
Delay(1000);};
```

معرفی ساختار های شرطی در C/C++ و آردوینو

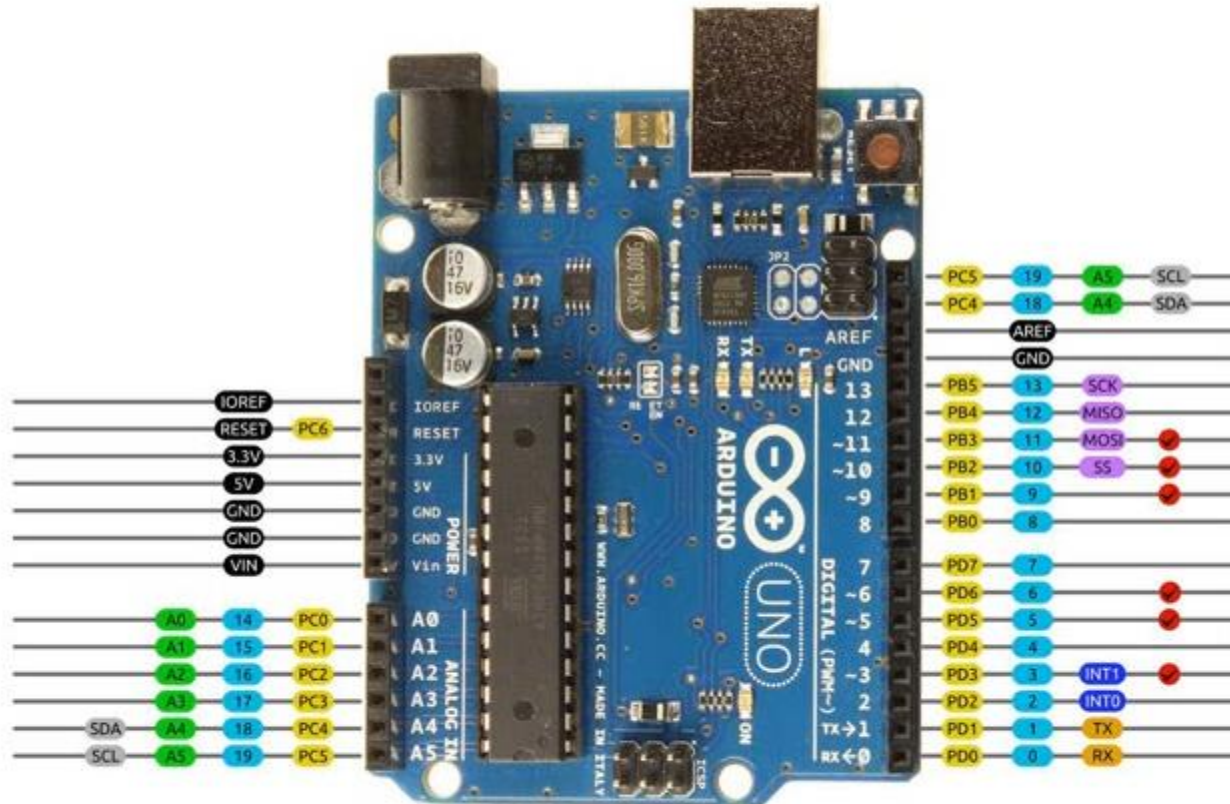
- ساختار حلقه While

```
{ (مادامی که شرط برقرار است) while  
    دستوراتی که باید اجرا شود  
}
```

این دستور برای آن به کار می رود که ما یک شرط را به عنوان ابزار کنترلی برای یک حلقه در نظر بگیریم، و مادامی که این شرط برقرار است، دستورات داخل حلقه همواره اجرا می شود

نکته: اگر در برنامه ای `{ while(1)` وجود داشت، برای ساخت یک حلقه بی نهایت تکرار به کار می رود. چون همواره شرط داخل پرانتز صحیح (**True**) و برقرار است. (صفر نیست)

واحد های داخلی آردوینو



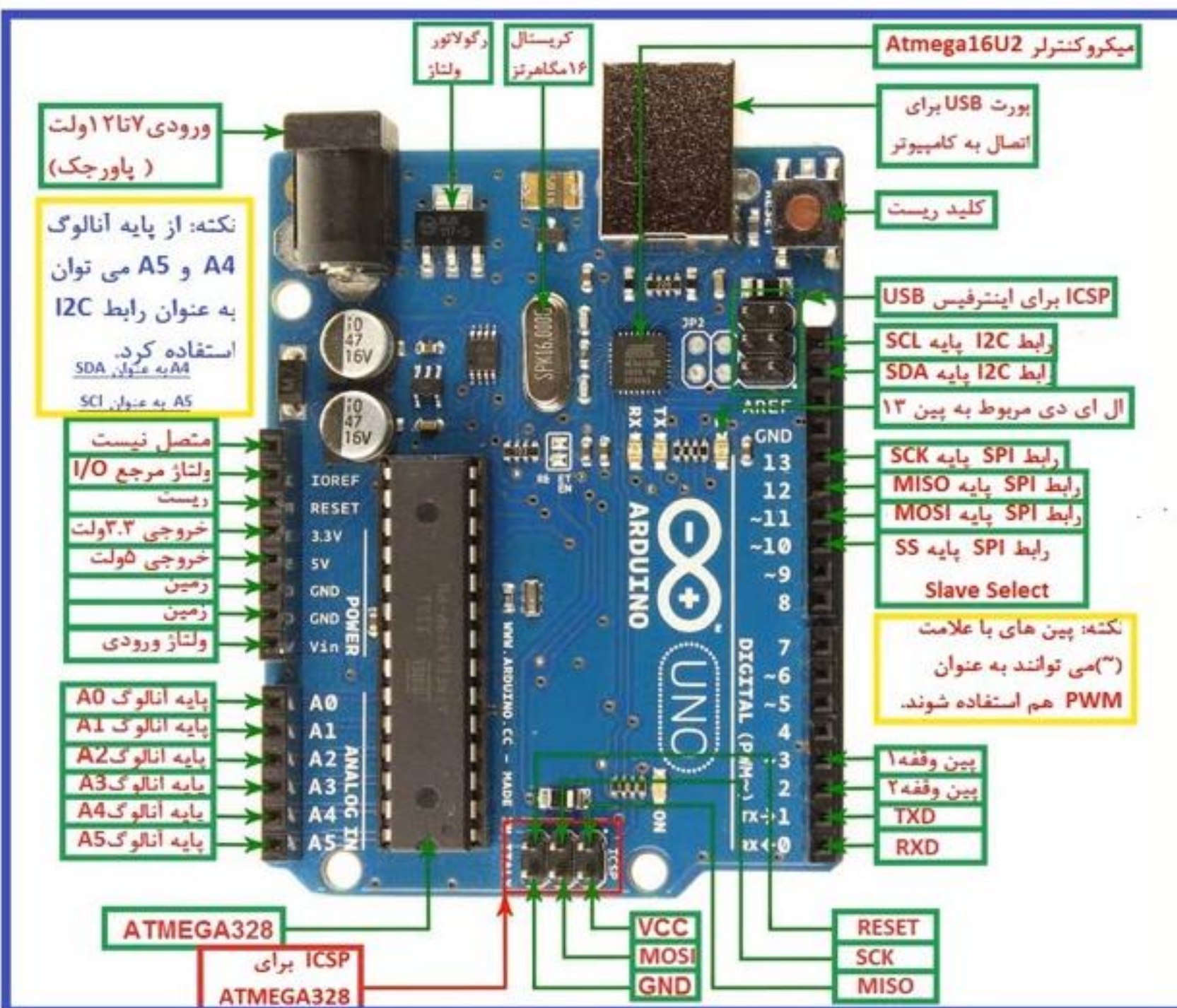
• واحد PWM

• واحد ارتباط سریال

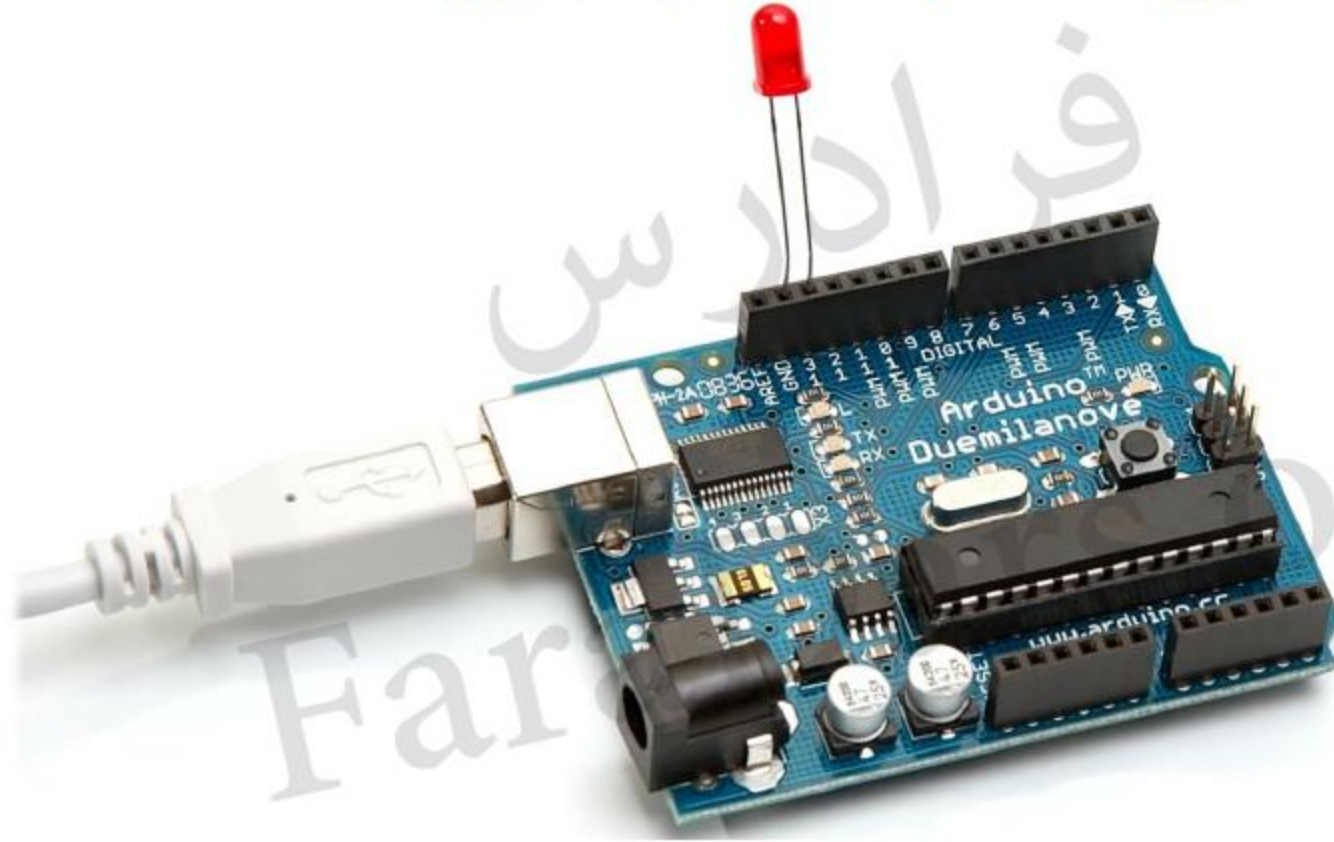
• واحد SPI

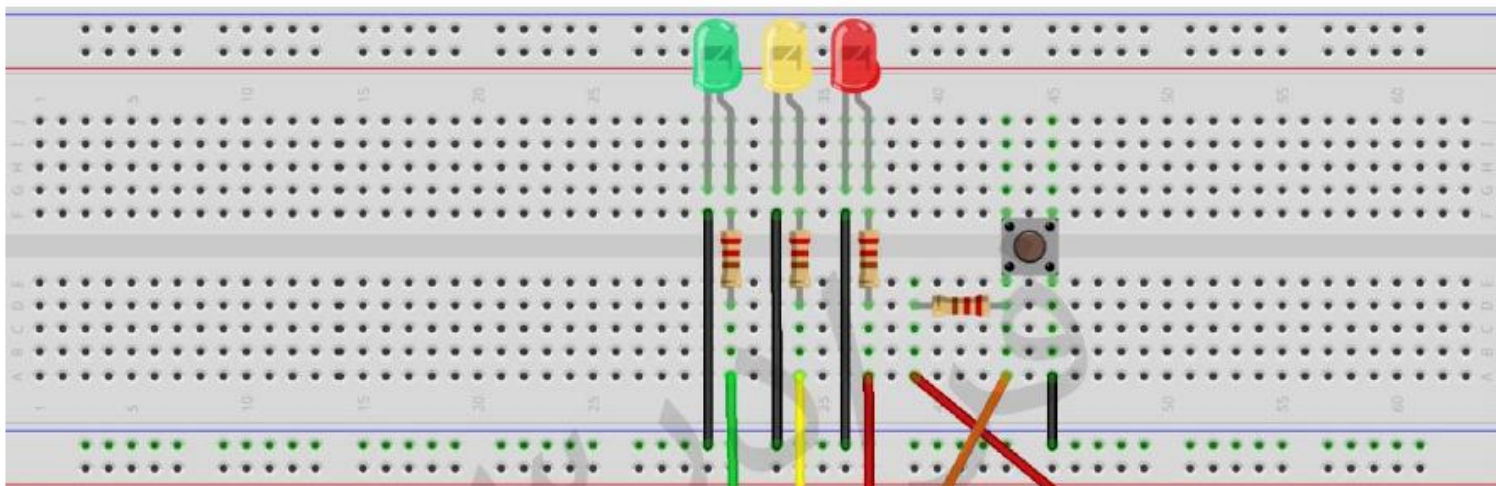
• واحد I2C و TWI

• واحد وقفه (Interrupt)



پروژه LED چشمک زن با آردوینو

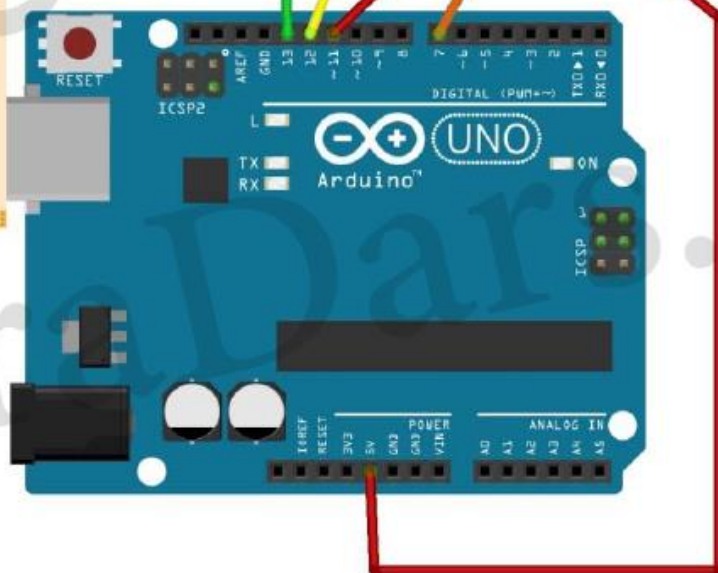




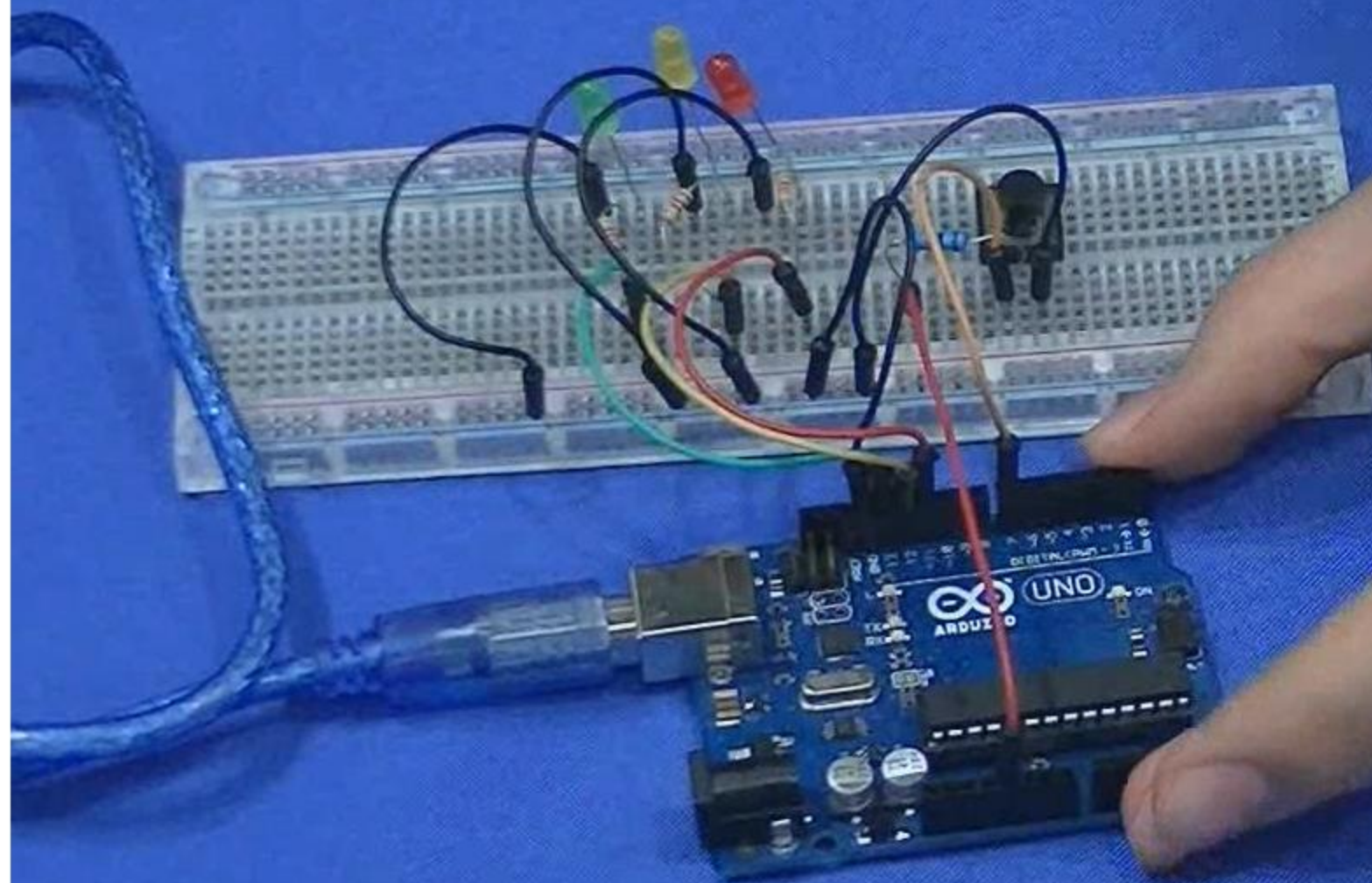
introduction to Arduino

session2 Section4
Example

==>FaraDars.org <==



fritzing

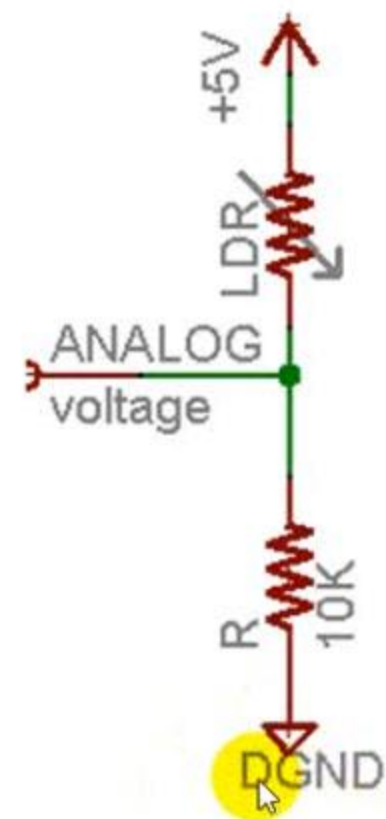
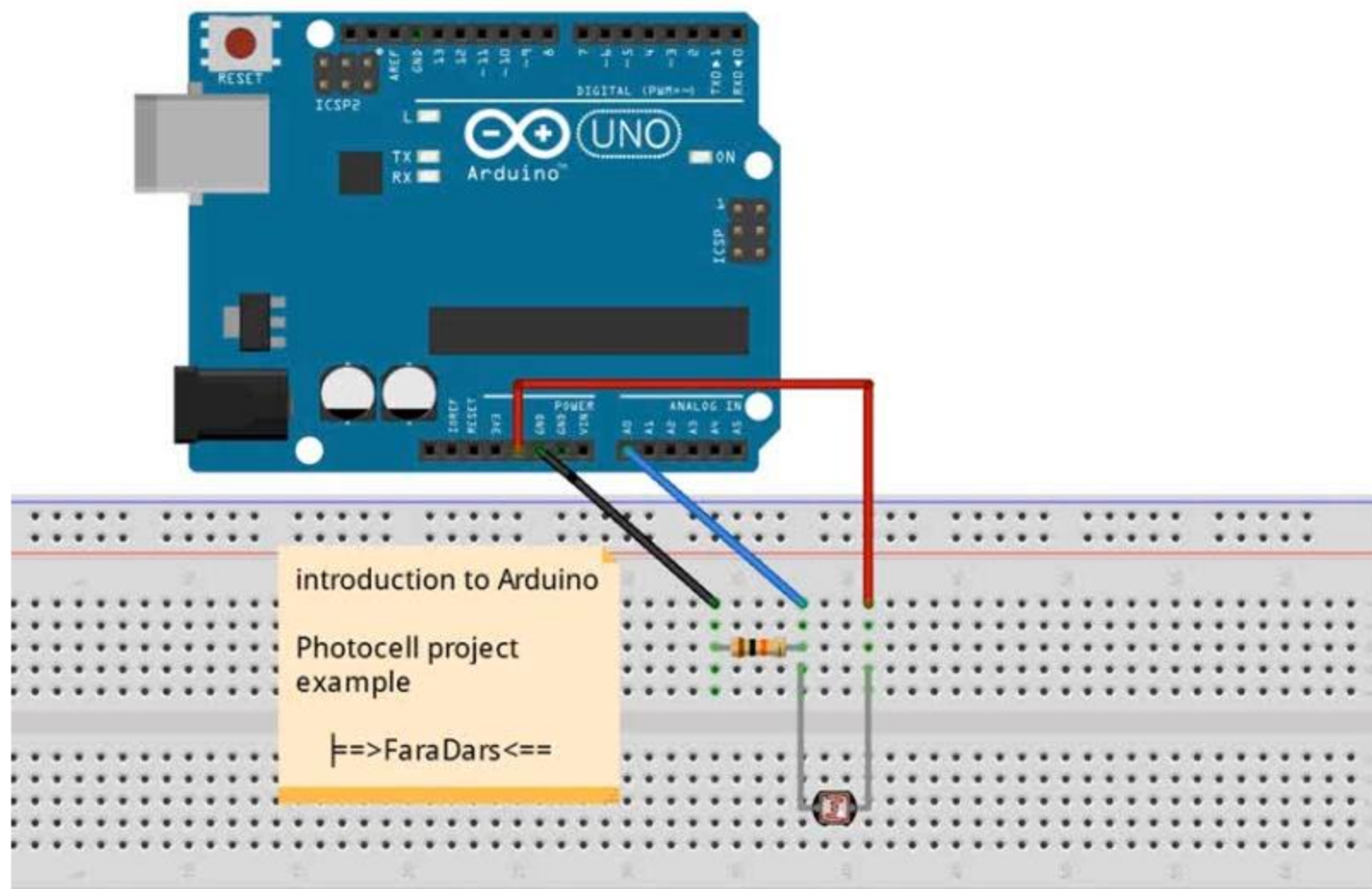


فتوسل چیست ؟

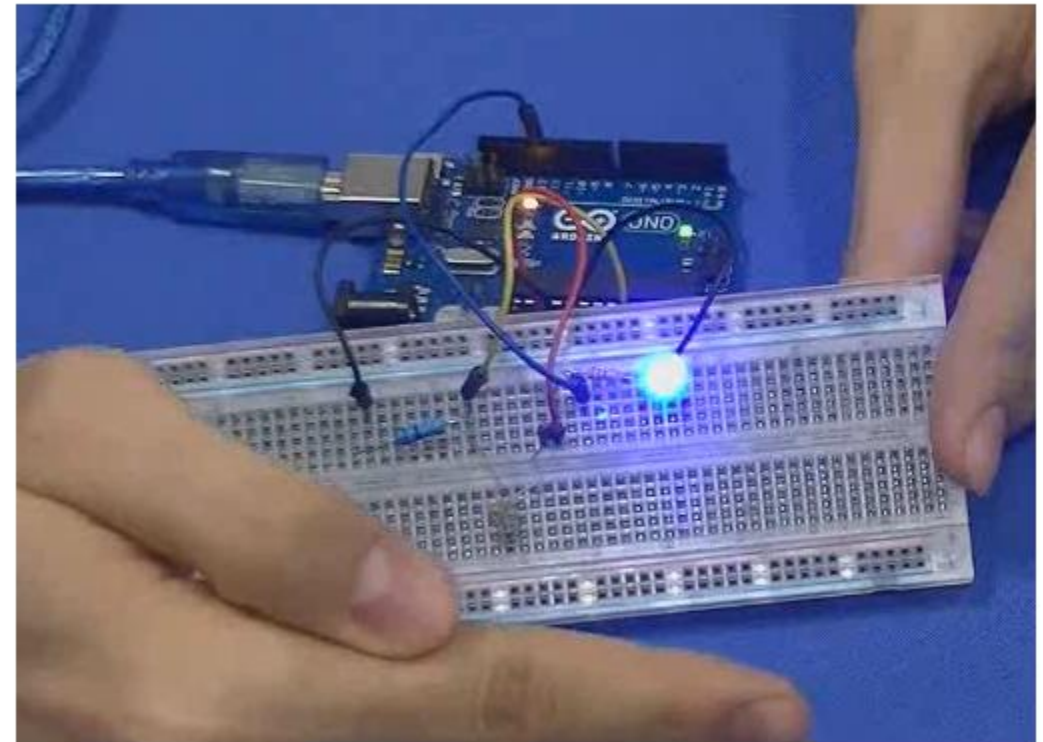
- فتوسل یک مقاومت حساس به نور است، که با تغییر میزان شدت نور، مقاومت خروجی آن تغییر می کند
- نام دیگر آن (LDR (Light dependent Resistor که مقاومت آن در نور کم و تاریک زیاد بوده و اگر نور بیشتری به آن بتابد مقاومت آن کمتر خواهد شد



اتصال فتوسل به آردوینو



```
int sensor;  
int led=13;  
  
void setup(){  
  Serial.begin(9600);  
  pinMode(led,OUTPUT);  
}  
  
void loop(){  
  sensor = analogRead(A0); // Khandane Meghdare Sensor Photocell  
  
  Serial.print("Sensor Value= ");  
  Serial.println(sensor);  
  delay(300);  
  
  if( sensor < 100){  
    digitalWrite(led,LOW);  
  }  
  else{  
    digitalWrite(led, HIGH);  
  }  
}
```



سنسور فراصوت (UltraSonic) چیست ؟

- سنسور Ultrasonic، با استفاده از امواج فراصوت (که گوش انسان قادر به شنیدن آن نیست)، به وسیله ی قطعه ی پیزوالکتریکی که بر روی آن تعبیه شده است، تولید شده و به ما جهت تشخیص مسافت ها با دقت تقریباً بالایی کمک می کند.
- جالب است بدانید این سنسورها از الگوی مسیریابی خفاش ها، الهام گرفته شده است که همانطور که می دانید این پرندگان قدر به دیدن محیط اطراف خود نیستند و با شنیدن بازخورد (پژواک) صدایی که از محیط دریافت می شود موانع را تشخیص می دهند



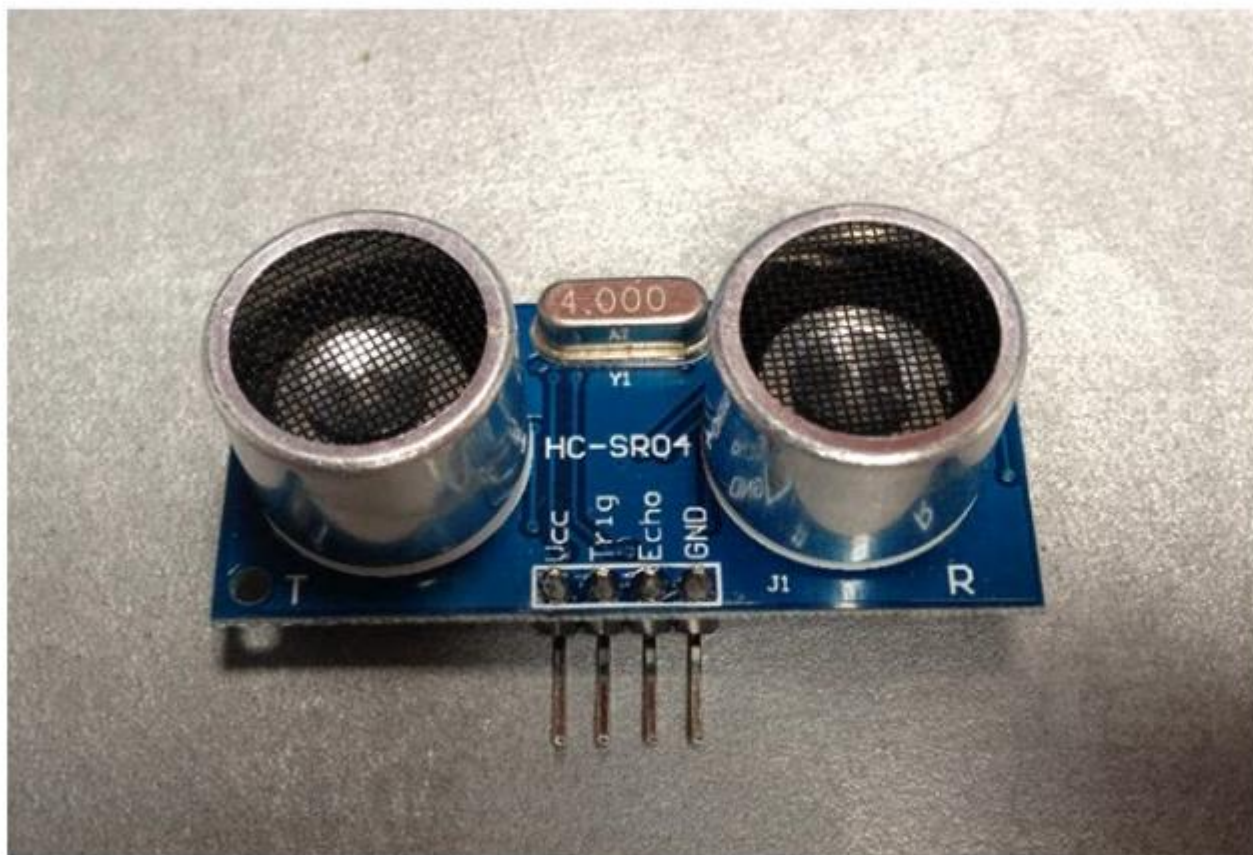
سنسور فراصوت (UltraSonic) چیست ؟

- نحوه ی عملکرد سنسور اولتراسونیک به این شکل است که یک صوت در فرکانس بالا را تولید و ارسال می کند، پس از برخورد این صوت با موانع محیطی، پژواک آن بازگشت داده می شود، و در گیرنده با استفاده از سنسور پیزو الکتریکی که در حالت معکوس فرستنده تنظیم شده است زمان طی شده پس از ارسال صوت و بازگشت آن را محاسبه می کنیم.
و از طریق محاسبه سرعت صوت (۳۴۰ متر بر ثانیه) می توان فاصله تا جسم مورد نظر را تخمین زد !

- $$\frac{1 \text{ ثانیه}}{340 \text{ متر}} \times \frac{1000000 \text{ میکروثانیه}}{1 \text{ ثانیه}} \times \frac{1 \text{ متر}}{100 \text{ سانتی متر}} \approx 29 \frac{\text{میکروثانیه}}{\text{سانتی متر}}$$

معرفی یک سنسور فراصوت UltraSonic

- ماژول SRF04 یک ماژول ارزان قیمت و مناسب جهت استفاده در پروژه ما می باشد



```
int PingPin = 13;
int EchoPin = 12;

void setup() {

  pinMode(PingPin, OUTPUT);
  pinMode(EchoPin, INPUT);

  Serial.begin(9600);
}

void loop() {
  unsigned long duration, cm;

  digitalWrite(PingPin, LOW);
  delayMicroseconds(2);
  digitalWrite(PingPin, HIGH);
  delayMicroseconds(5);
  digitalWrite(PingPin, LOW);

  duration = pulseIn(EchoPin, HIGH); // Daryaft Zamane bazgasht Sot bar Habs Micro Saniye

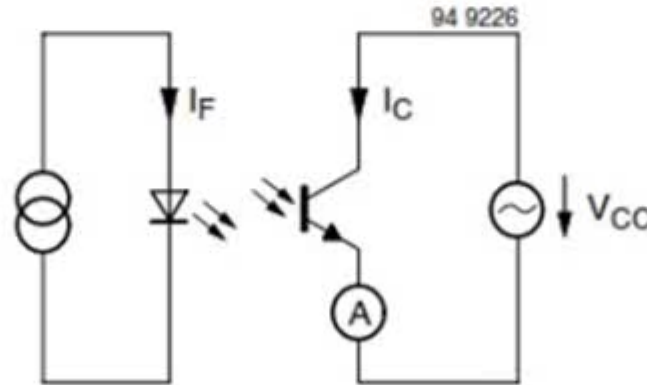
  cm = (duration/29)/2;;

  Serial.print(cm);
  Serial.println("cm");
  delay(100);
}
```



سنسور مادون قرمز چیست ؟

- سنسور مادون قرمز (Infara Red) یا فروسرخ، نوری نامرئی (از دید ما انسان ها) است و به دلیل ویژگی های خوب ذاتی و طول موج مناسب، جهت انتقال اطلاعات، و فاصله یابی مورد استفاده قرار می گیرد.

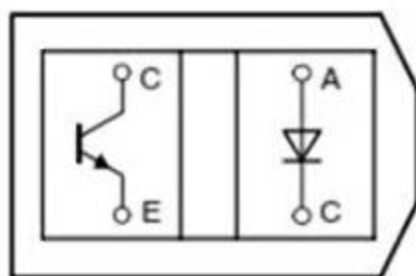


سنسور مادون قرمز چیست ؟

- سنسور TCRT5000 یک نمونه سنسور ارزان قیمت و پکیج شده جهت استفاده در تعیین مسافت، سنجش دور در انکودرها و همچنین در ربات های مسیر یاب نیز کاربرد دارد.

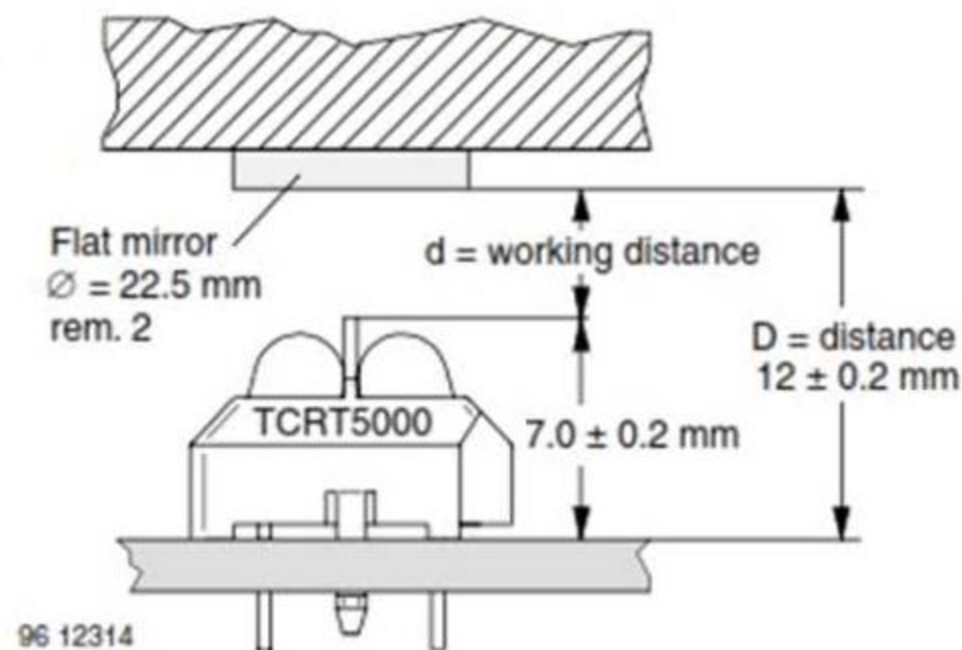


19156_2

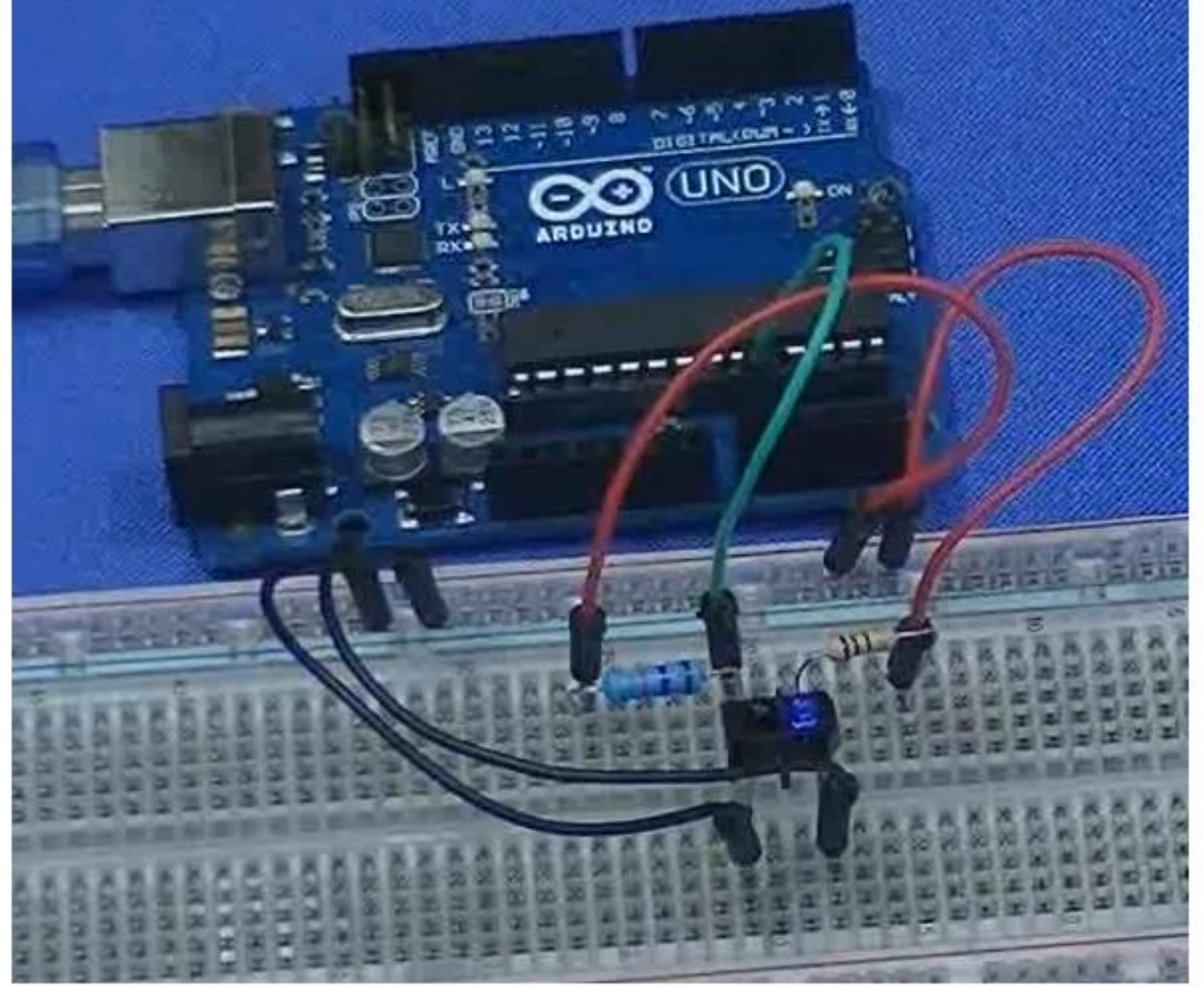
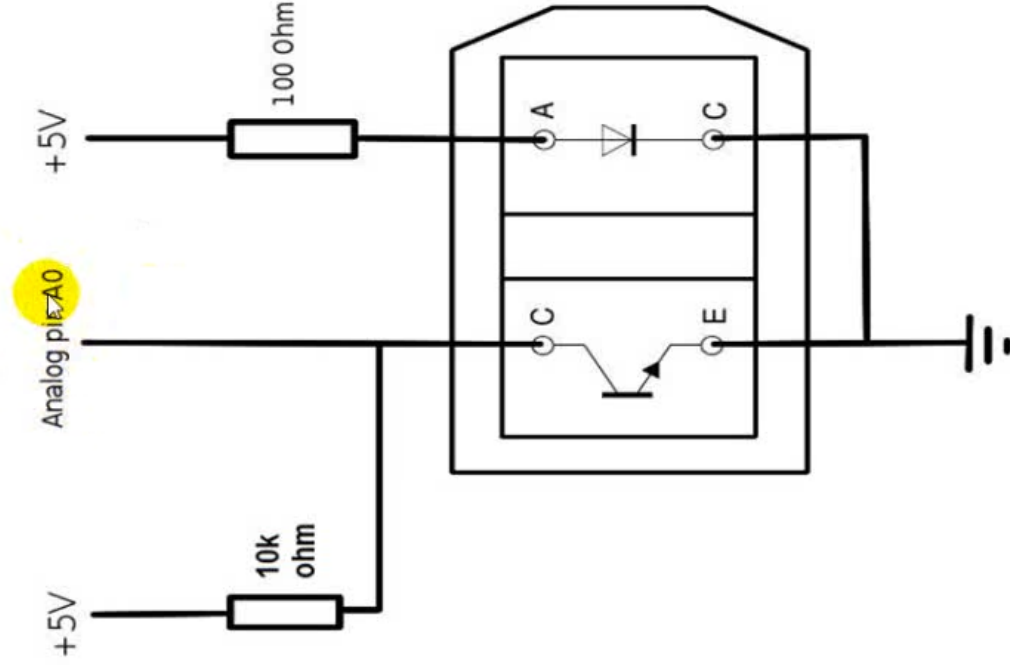


Top view

19156_1



اتصال سنسور TCRT5000 به آردوینو



```
int sensor;

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

void loop() {
    sensor = analogRead(12);

    Serial.print("Sensor Value= ");
    Serial.println(sensor);
    delay(300);
}
```

```
int sensor;

void setup() {
    Serial.begin(9600);
    pinMode(12, INPUT);
}

void loop() {
    sensor = digitalRead(12);

    Serial.print("Sensor Value= ");
    Serial.println(sensor);
    delay(300);

}
```

راه اندازی درایور موتور DC

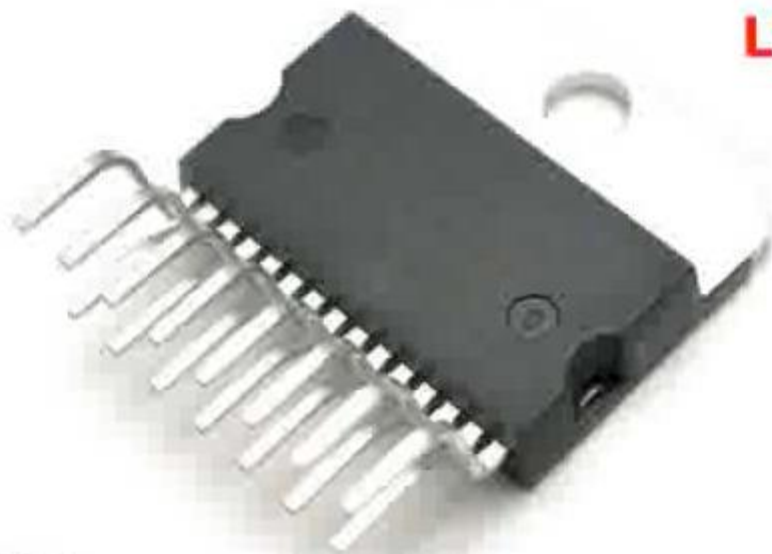
Summary

- درایور (**Driver**) یا راه انداز چیست ؟

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader

راه اندازی درایور موتور DC با تراشه L298

آشنایی با مشخصات فنی تراشه L298



Summary

Operating Voltage 4V to 35V

Motor controller L298N, Drives 2 DC motors or 1 stepper motor

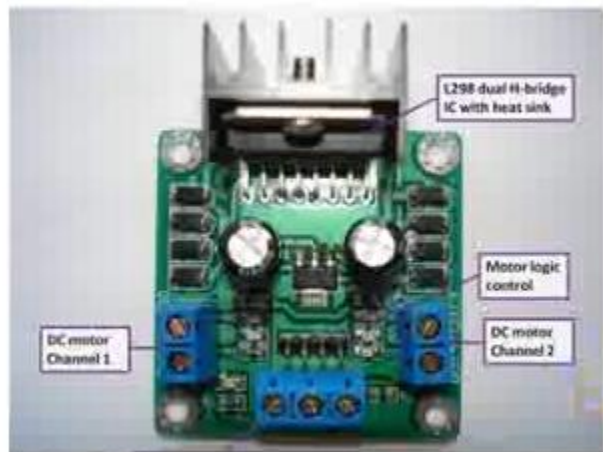
Max current 2A per channel or 4A max

Free running stop and brake function

Chip: ST L298N

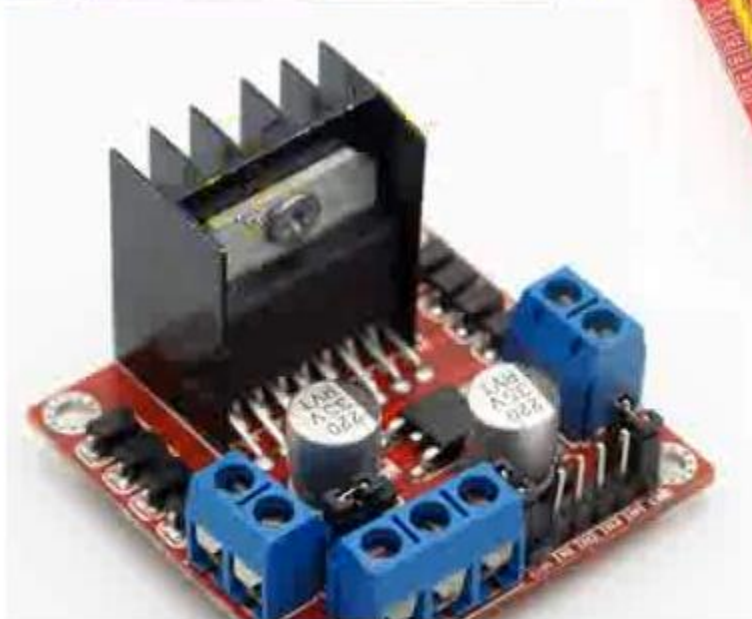
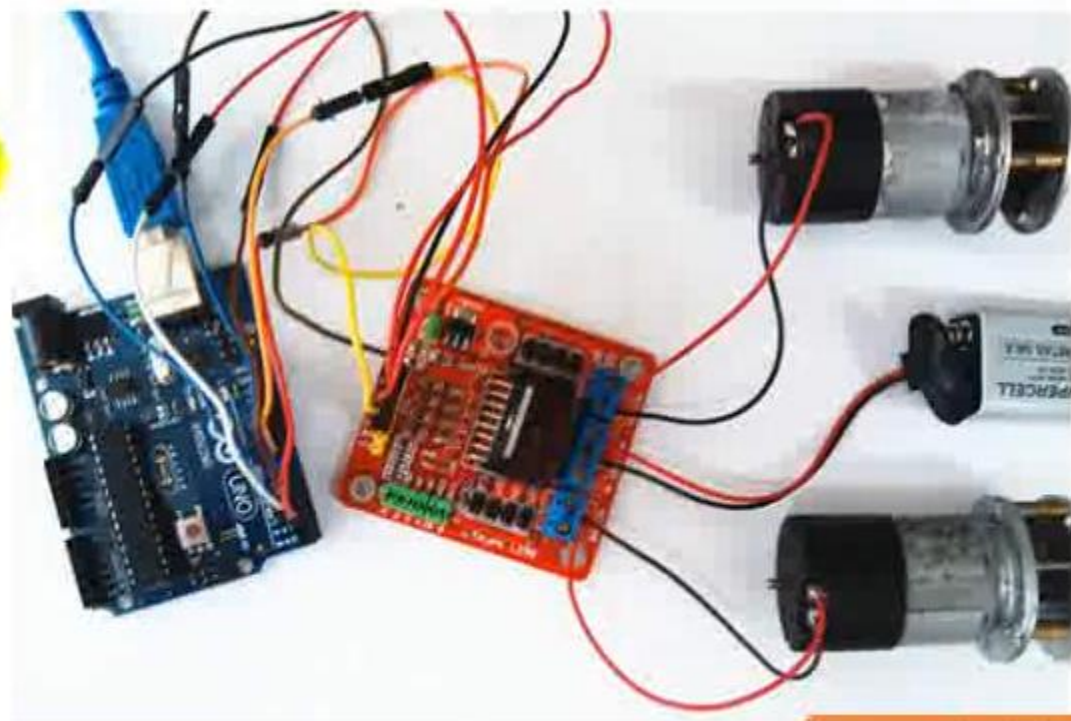
Logic power supply: 5V

Max power: 25W

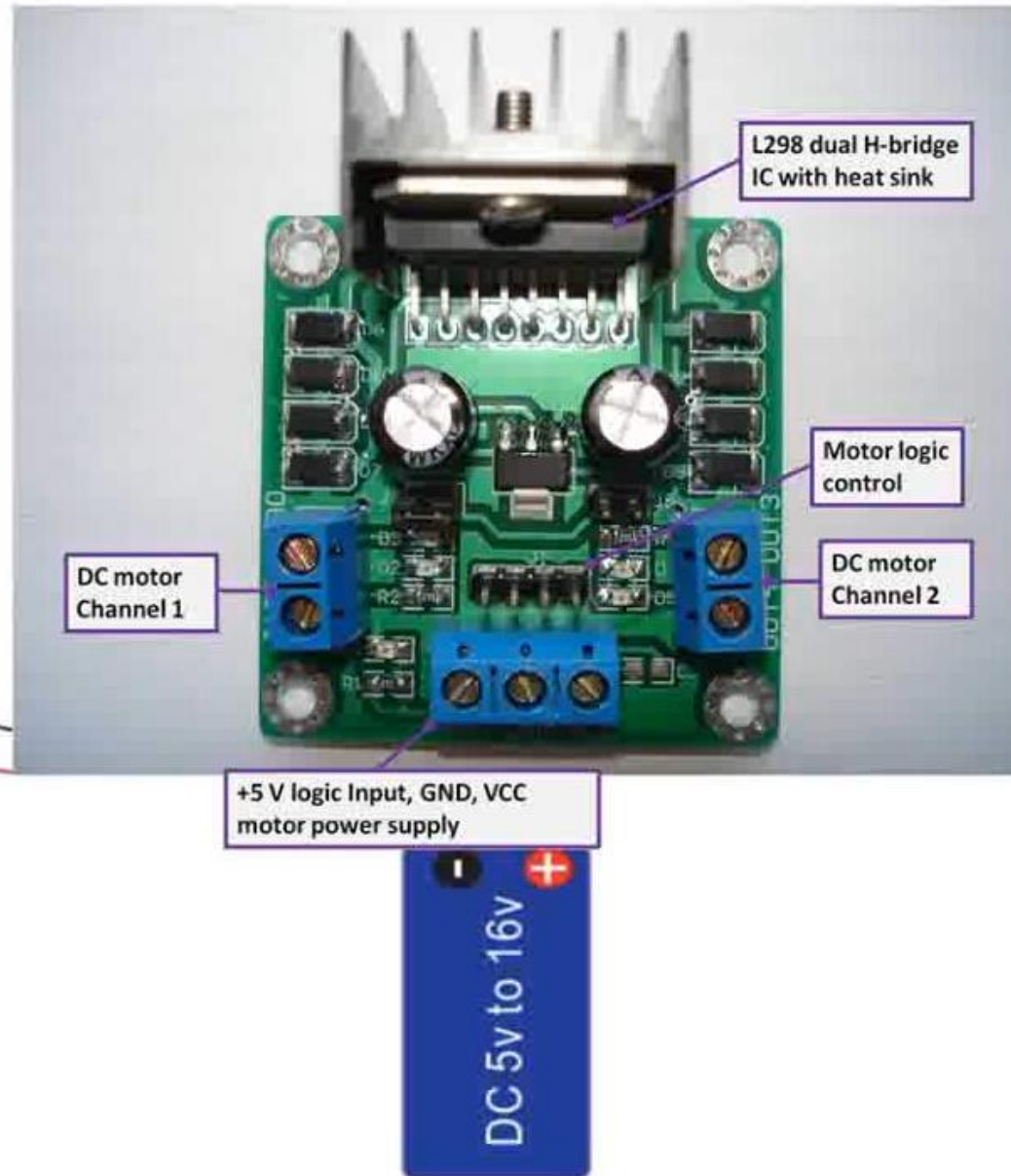
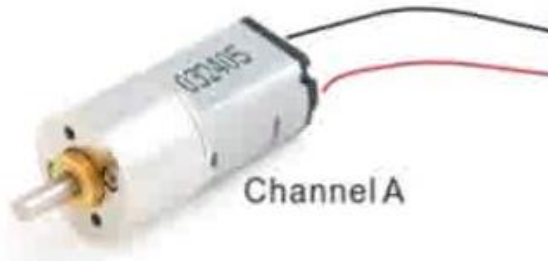


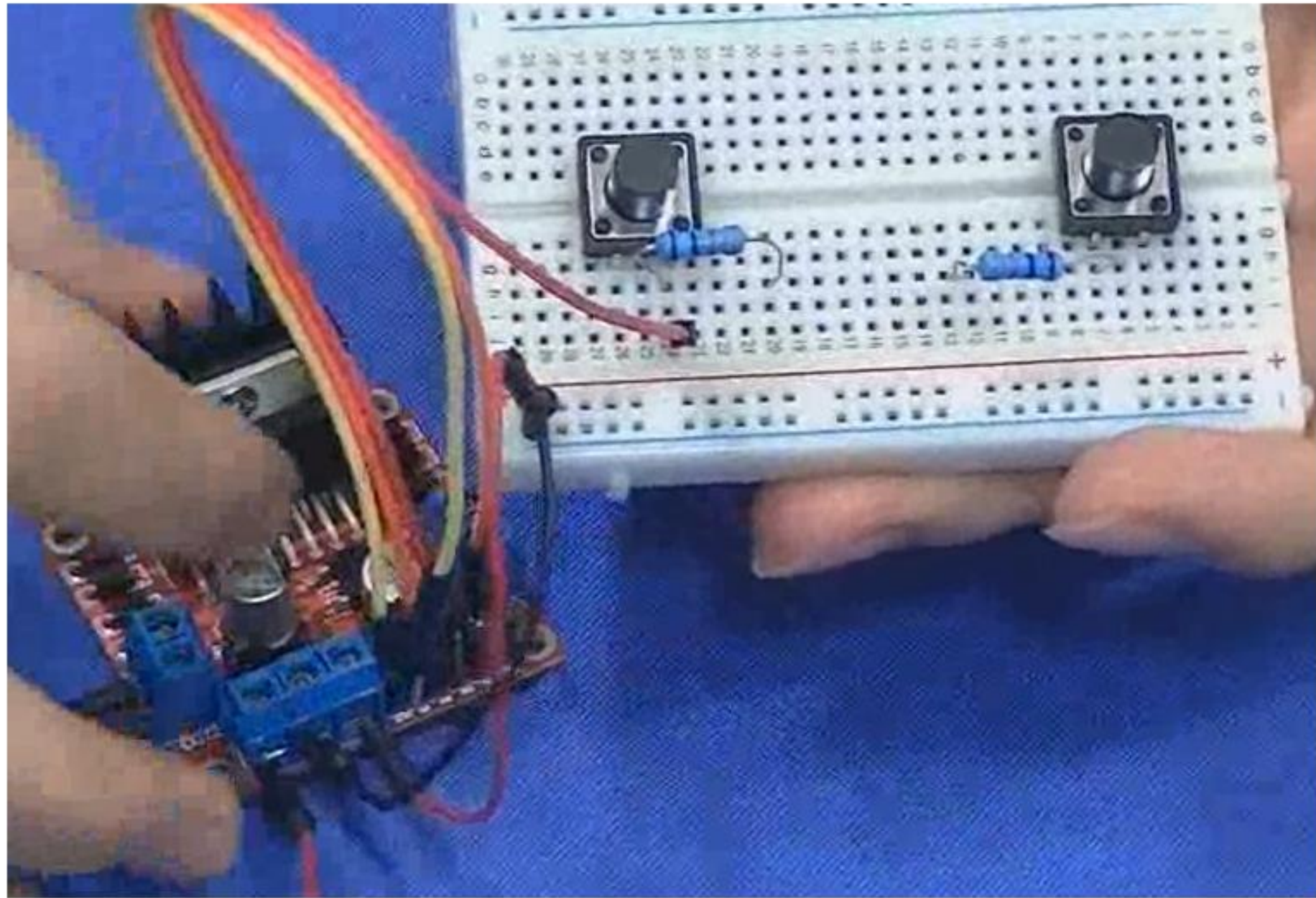
راه اندازی درایور موتور DC با تراشه L298

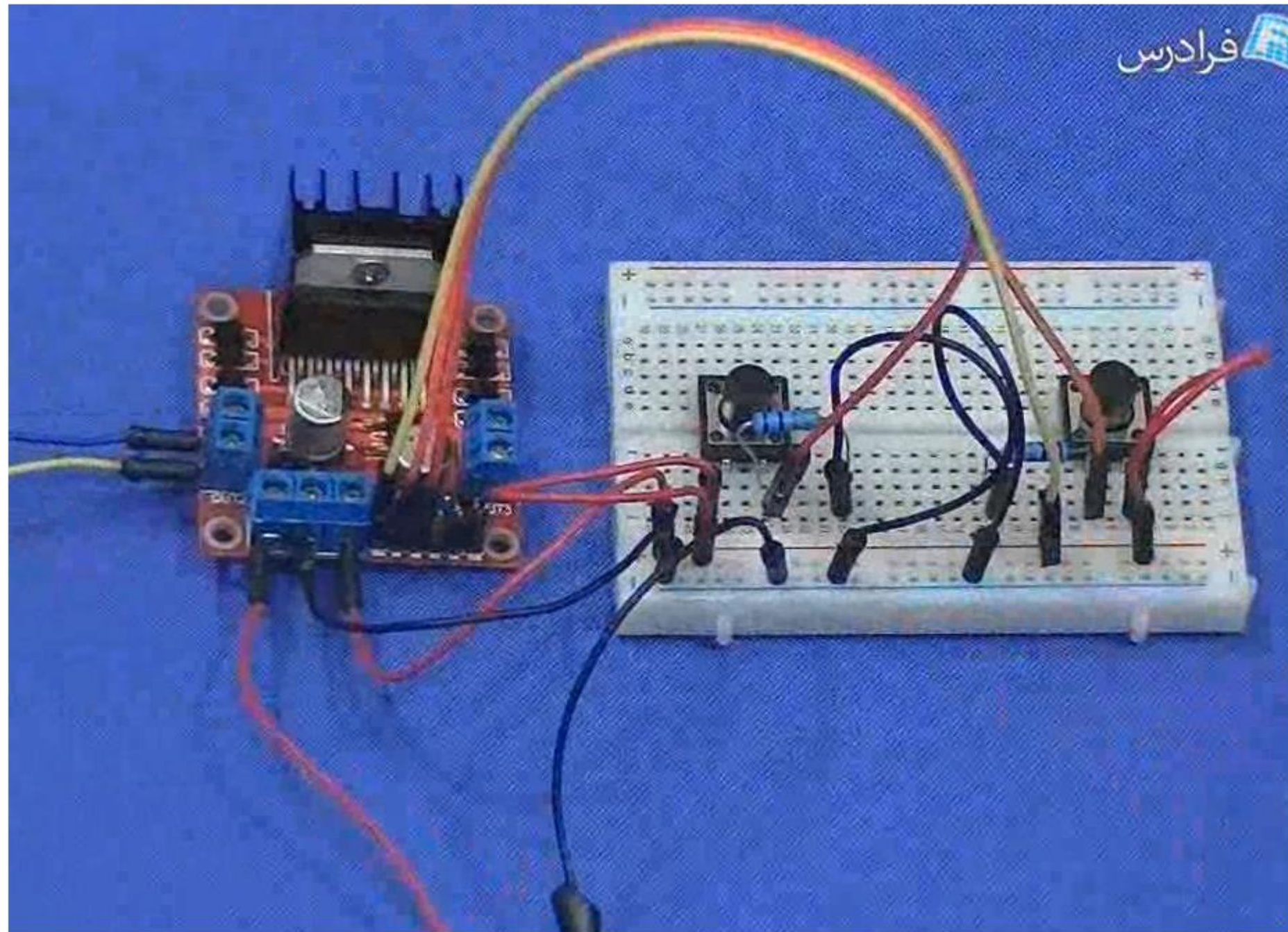
اشکال رایج در بازار (به صورت ماژول درایور موتور DC)

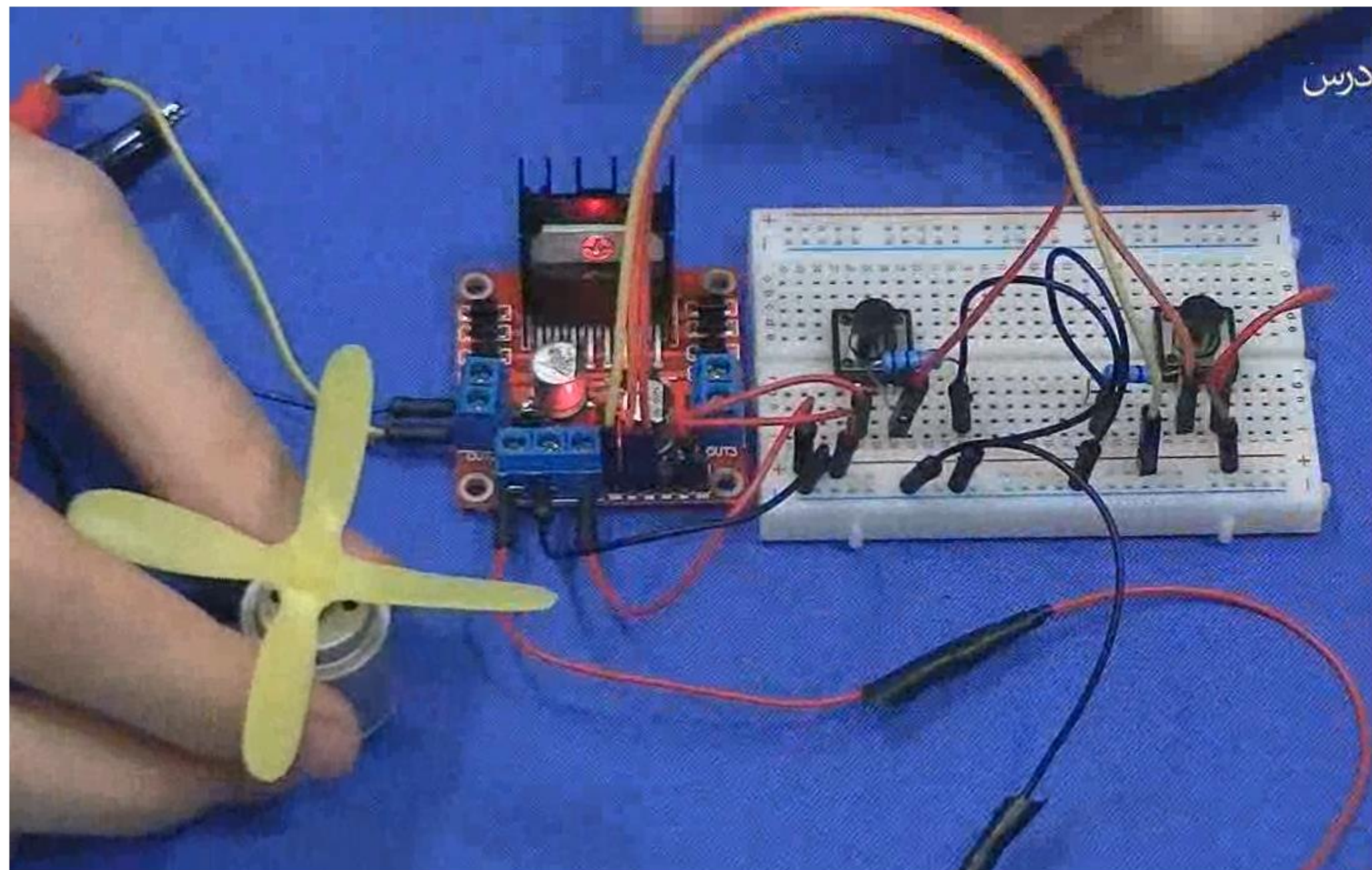


Channel A:
IN1---5V IN2---GND Forward
IN1---GND IN2--5V Reverse
ENA---5V channel A enable
ENA---GND channel A disable
ENA---PWM adjust speed







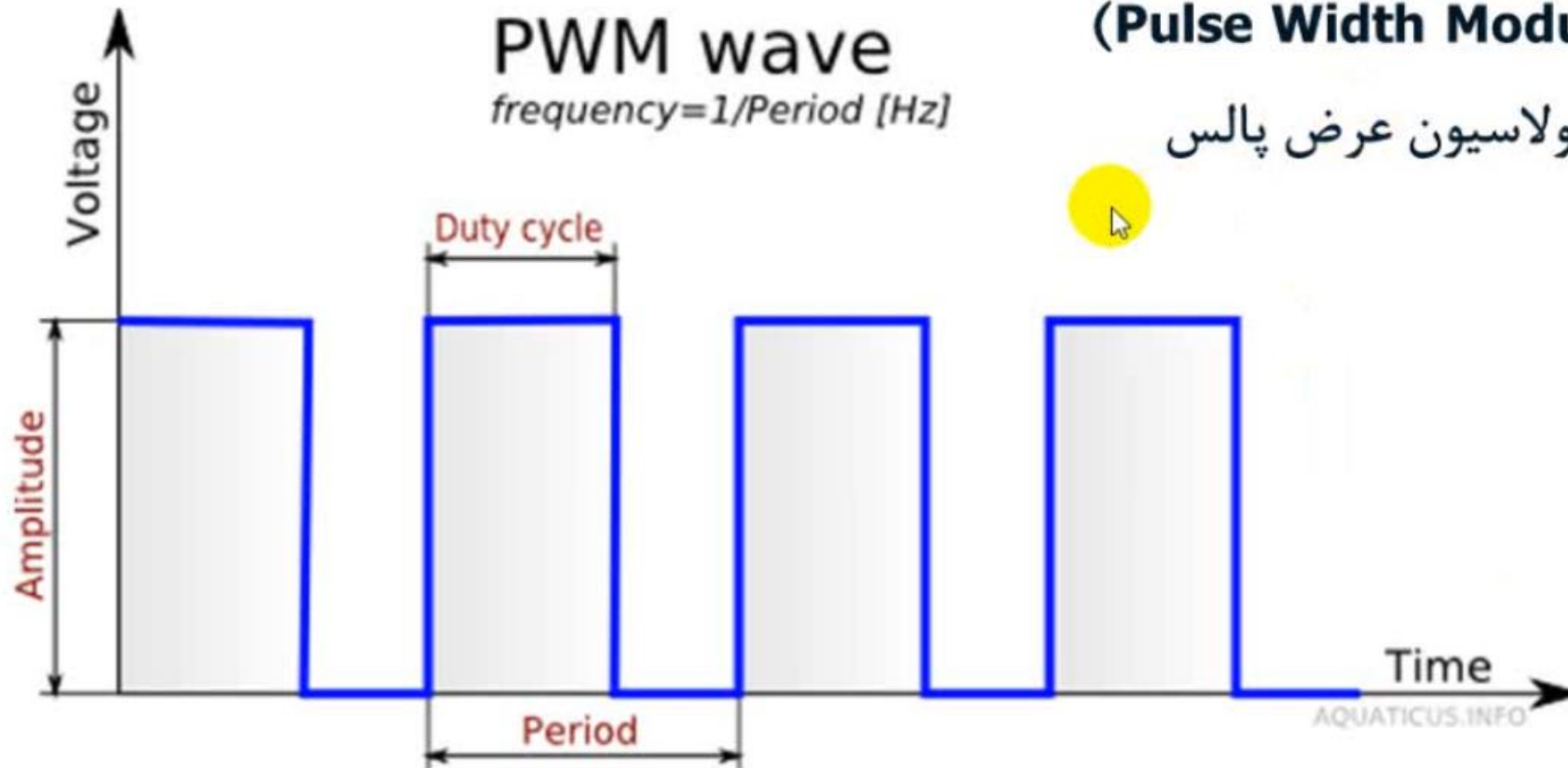


راه اندازی درایور موتور DC از طریق PWM

- PWM چیست ؟

- PWM مخفف (Pulse Width Modulation)

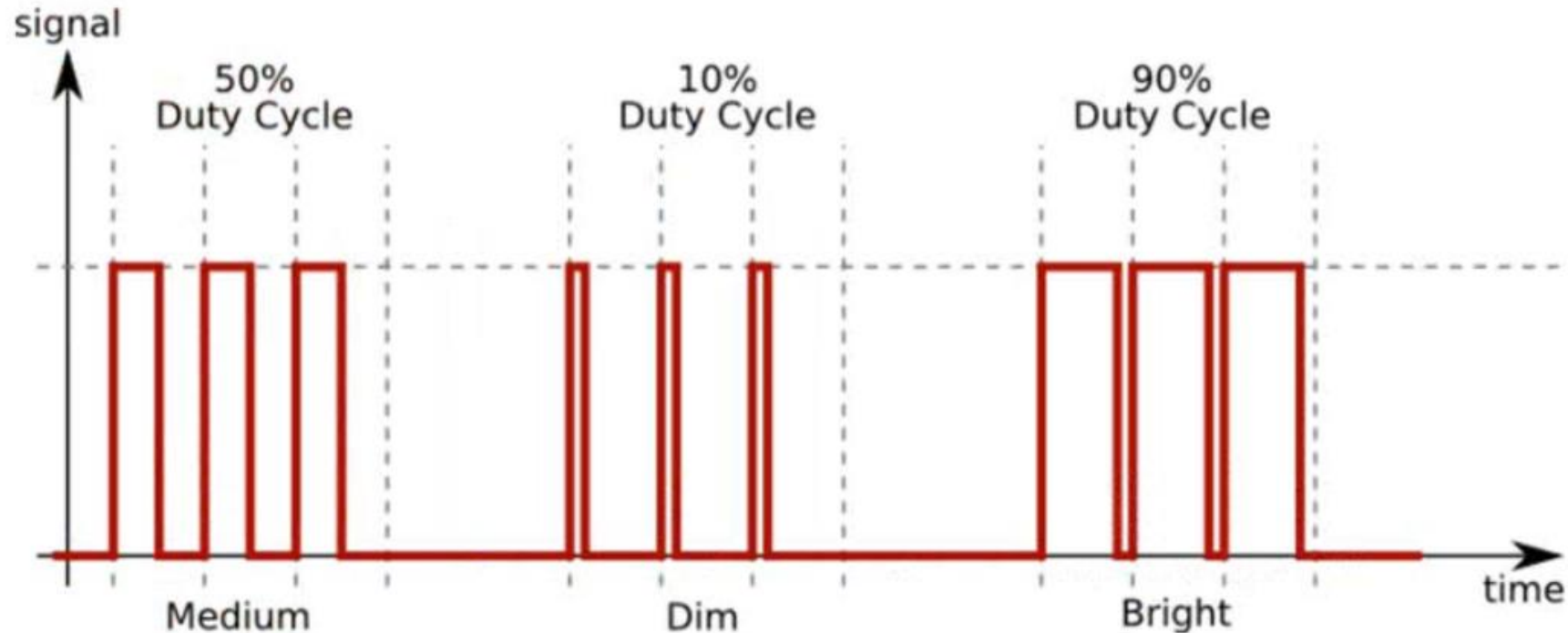
یعنی مدولاسیون عرض پالس



راه اندازی درایور موتور DC از طریق PWM

- کمی بیشتر از PWM بدانیم ...

- $$\text{Duty cycle} = \frac{\text{Time ON}}{\text{Time ON} + \text{Time OFF}} \times 100$$



راه اندازی درایور موتور DC از طریق PWM

دستور ایجاد پالس PWM در آردوینو

analogWrite(عدد از صفر تا ۲۵۵ , نام پایه);

- نکته: در **Arduino Uno** فقط پایه های (۳، ۵، ۶، ۱۰، ۱۱) امکان پیاده سازی PWM را به ما می دهد

Pulse Width Modulation

0% Duty Cycle - analogWrite(0)



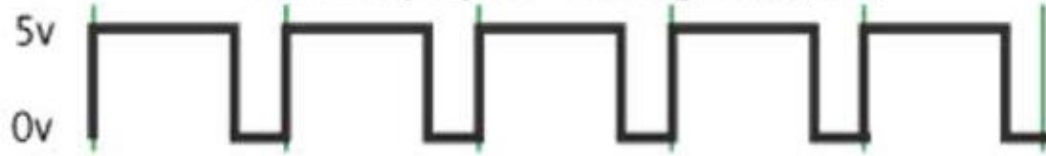
25% Duty Cycle - analogWrite(64)



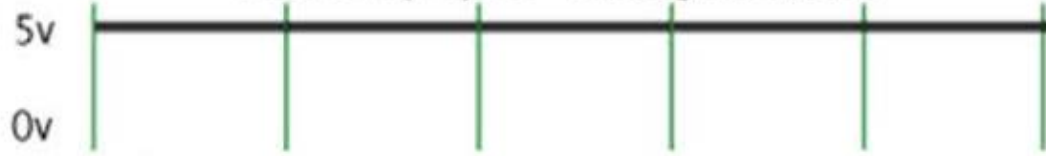
50% Duty Cycle - analogWrite(127)



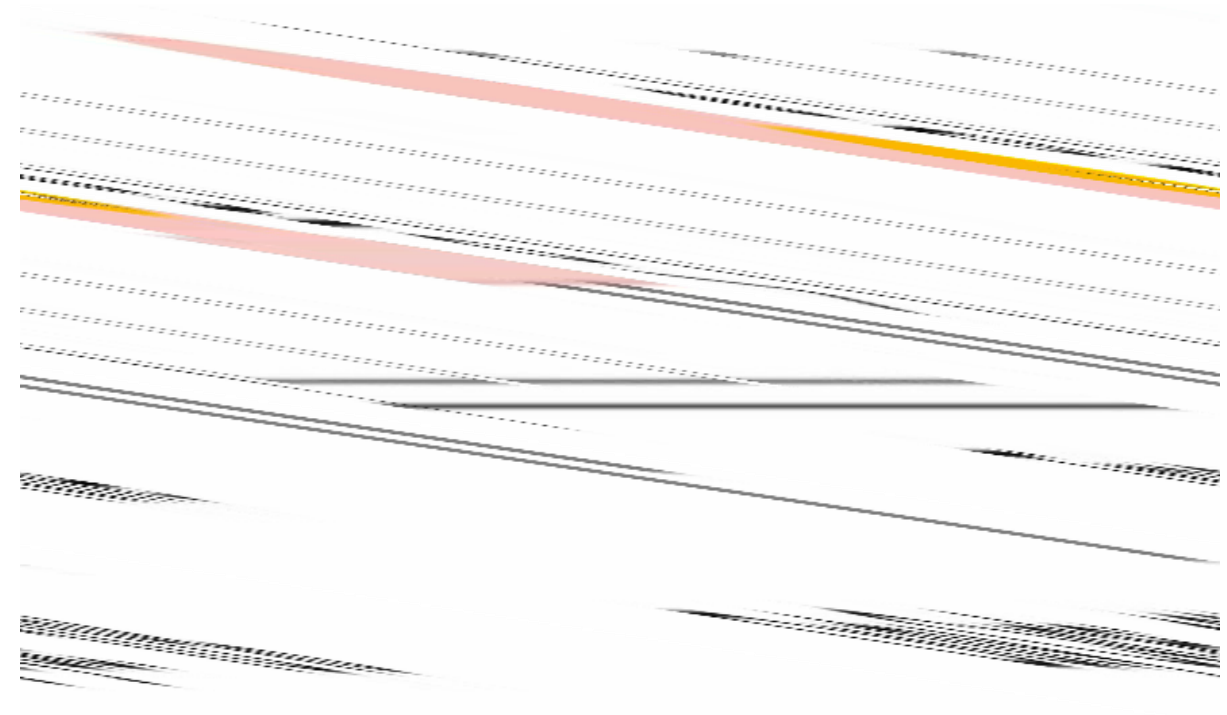
75% Duty Cycle - analogWrite(191)

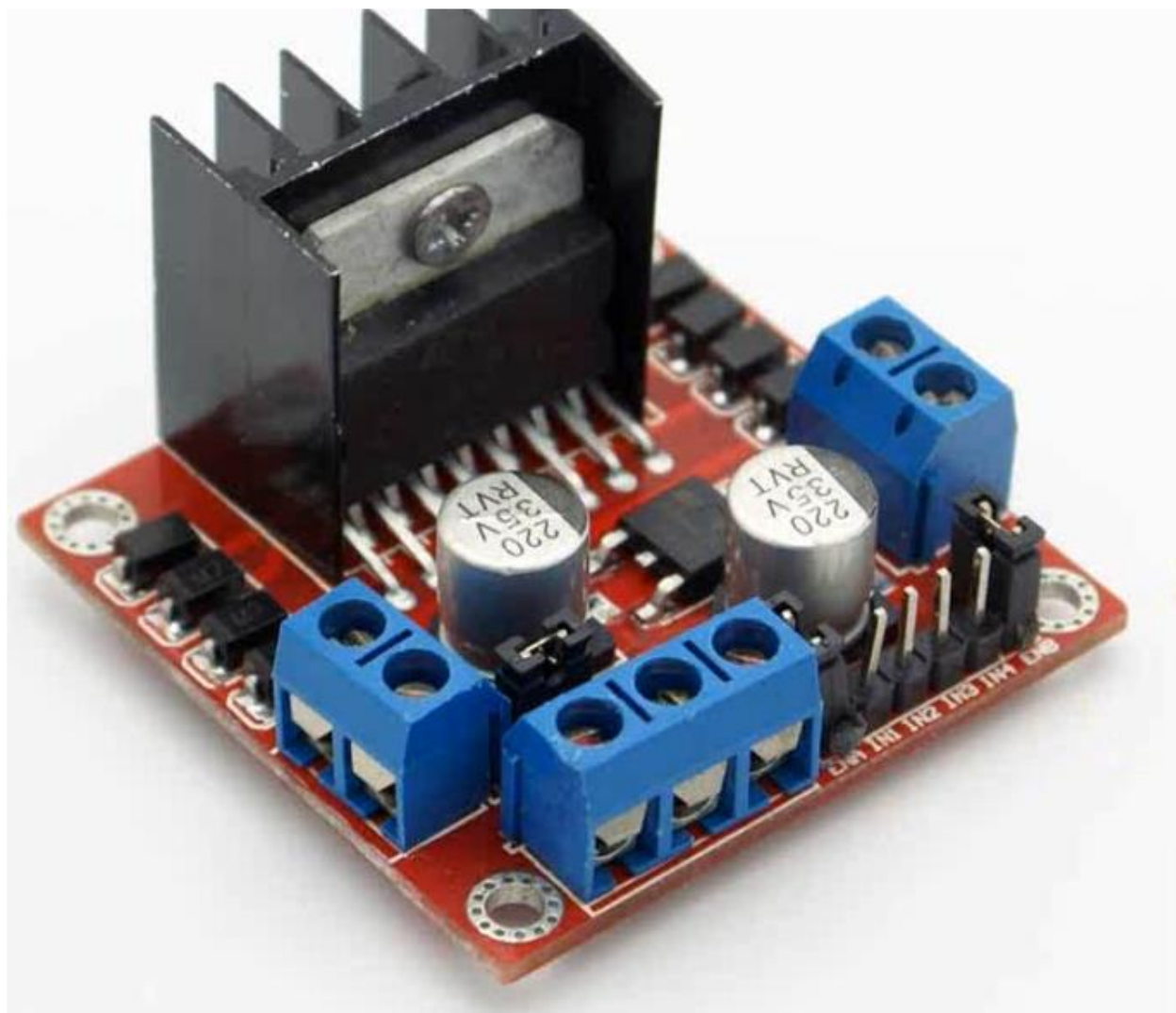


100% Duty Cycle - analogWrite(255)



- **analogWrite**(عدد از صفر تا ۲۵۵ , نام پایه);

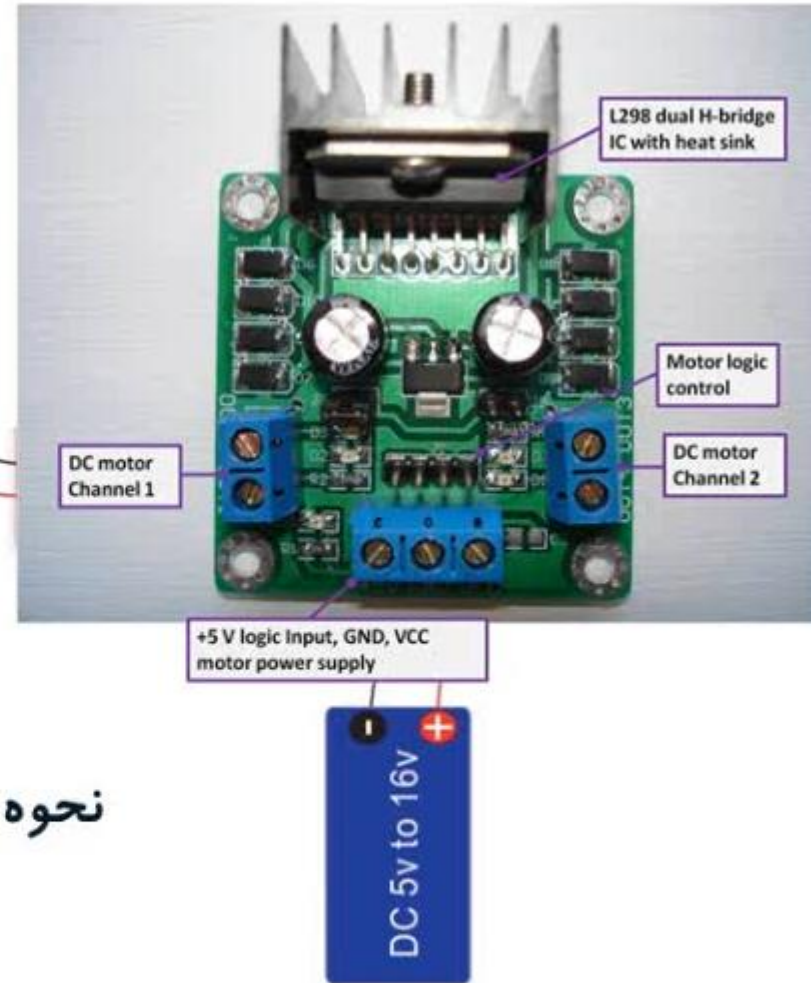




راه اندازی پیشرفته

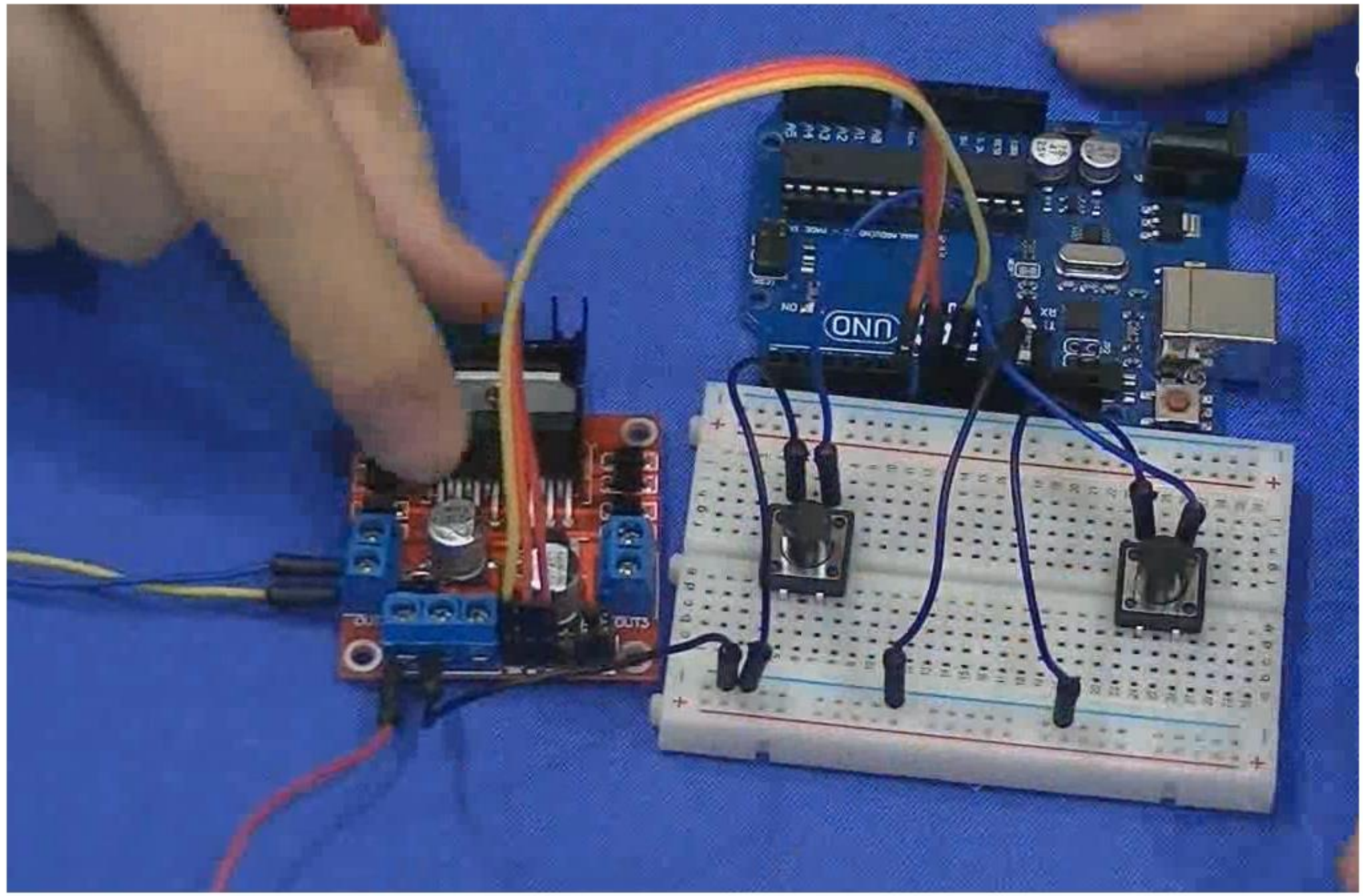
در این حالت ما می خواهیم یک موتور را به کانال **A** متصل کنیم و از طریق کلید فشاری (**Push Button**) جهت آن را کنترل کنیم. بنحوی که با فشردن کلید، سرعت موتور به تدریج افزایش یافته تا به ماکزیمم سرعت خود برسد و بالعکس.

Channel A:
IN1---5V IN2---GND Forward
IN1---GND IN2--5V Reverse
ENA---5V channel A enable
ENA---GND channel A disable
ENA---PWM adjust speed



نحوه ی اتصال به ماژول L298

IN1 ➔ Pin7 Arduino
IN2 ➔ Pin8 Arduino
ENA ➔ Pin9 Arduino



```

int IN1 = 7; // IN1 L298 ==> pin 7 Arduino
int IN2 = 8; // IN2 ==> Pin8
int ENA = 9; // Enable A ==> Pin9 PWM

int PB1 = 11; // Push button 1 ==> Pin 11 Arduino
int PB2 = 12; // Push button 2 ==> pin 12

int MotorSpeed = 0;

void setup(){
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(ENA, OUTPUT);

  pinMode(PB1, INPUT_PULLUP); // Default = HIGH
  pinMode(PB2, INPUT_PULLUP);

  analogWrite(ENA, 0);
}

void loop(){
  while ( (digitalRead(PB1) == HIGH) && (digitalRead(PB2) == HIGH) ){
    MotorSpeed = 0;
  }
  boolean PB1P = digitalRead(PB1); // PB1P = Push button 1 Pressed
  boolean PB2P = digitalRead(PB2);
  if( PB1P == LOW){
    // Forward
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);

```

```

    if( MotorSpeed < 255){
      MotorSpeed +=5;
      analogWrite(ENA, MotorSpeed);
      delay(100);
    }
    else {
      MotorSpeed = 255;
      analogWrite(ENA, MotorSpeed);
    }
  }

  else if( PB2P == LOW){
    // Reverse
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, HIGH);

    if( MotorSpeed < 255){
      MotorSpeed +=5;
      analogWrite(ENA, MotorSpeed);
      delay(100);
    }
    else {
      MotorSpeed = 255;
      analogWrite(ENA, MotorSpeed);
    }
  }
}

```



❖ آشنایی با موتورهای الکتریکی - سروو

- کاربرد سروو ها؛ وسایلی که نیاز به حرکت دقیق دارند.

- ربات ها

- هواپیماهای مدل

- دستگاه های تراشکاری (CNC)

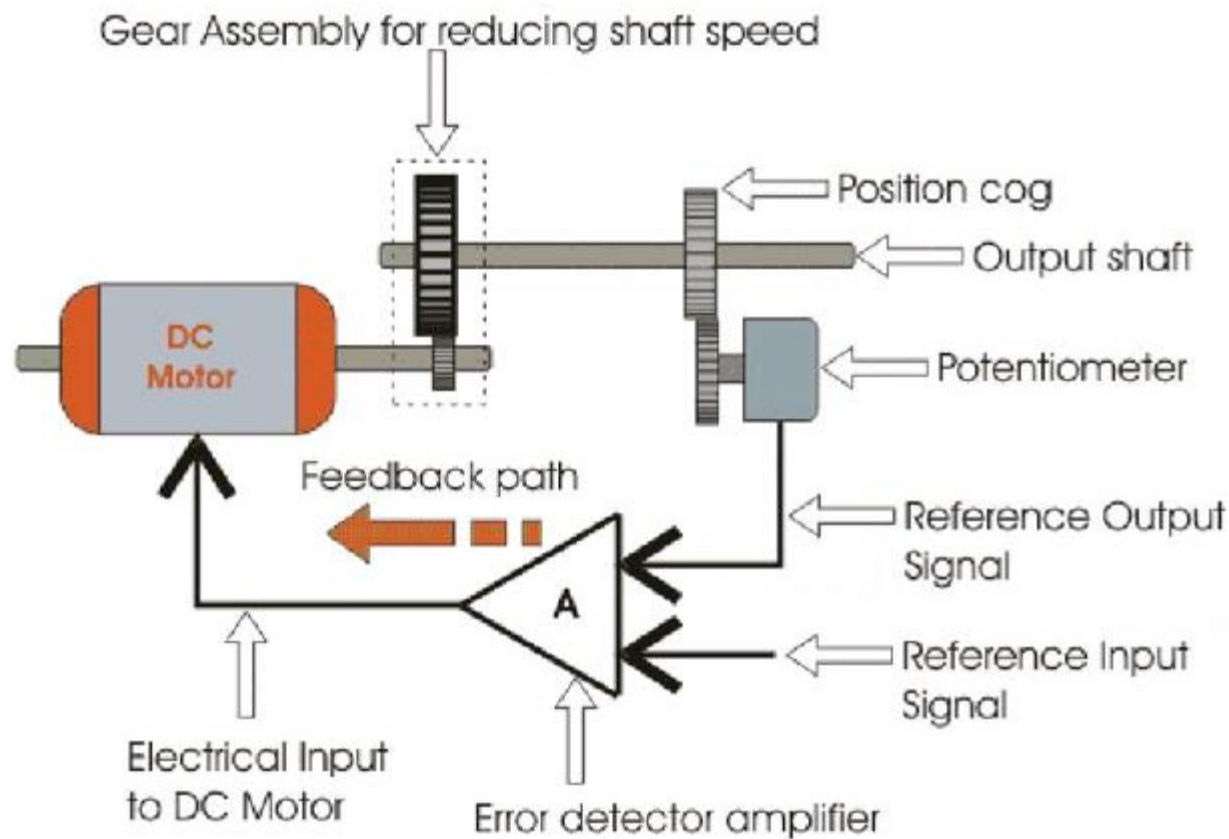
❖ آشنایی با موتورهای الکتریکی - سروو

• موتور سروو = موتور جریان مستقیم + مدار کنترل و چرخ دنده



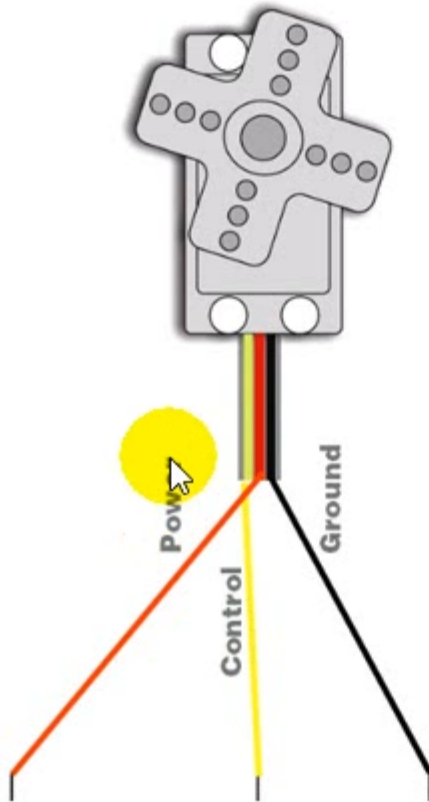
❖ آشنایی با موتورهای الکتریکی - سروو

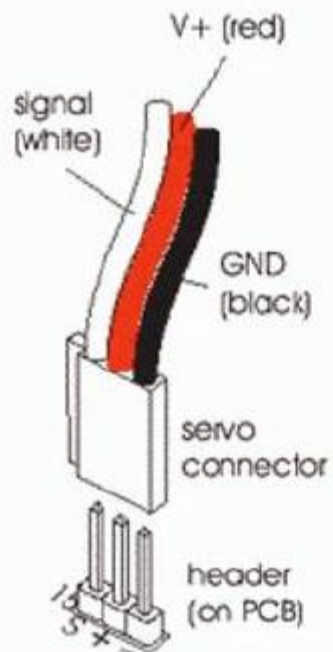
• بلوک دیاگرام



❖ آشنایی با موتورهای الکتریکی - سروو

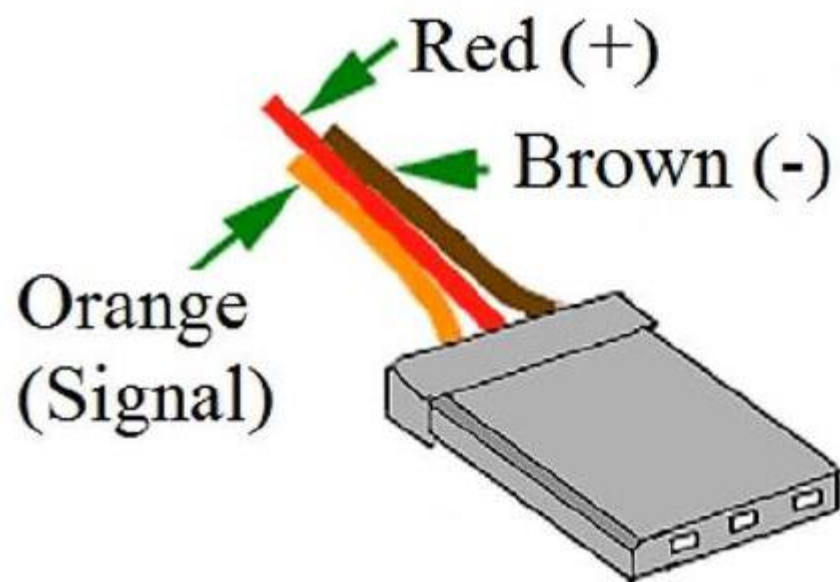
- کنترل موقعیت سروو موتور





معرفی پایه های Servo:

- اتصال به ۵ ولت (رنگ قرمز) $VCC \rightarrow$
- اتصال به زمین (غالباً به رنگهای تیره مشکی یا قهوه ای) $GND \rightarrow$
- اتصال به پایه های دیجیتال (PWM) (غالباً به رنگهای روشن تر نظیر سفید یا نارنجی) $Signal \rightarrow$

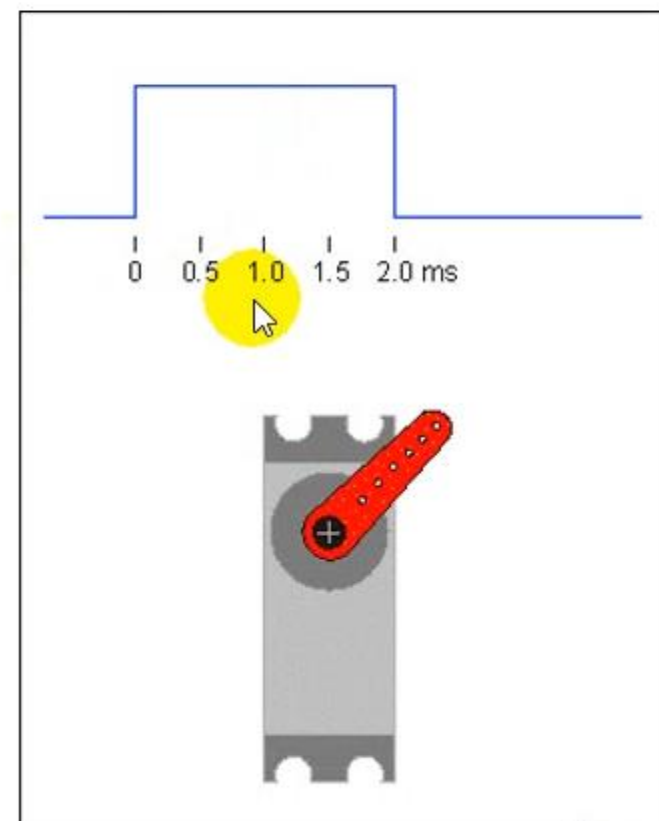
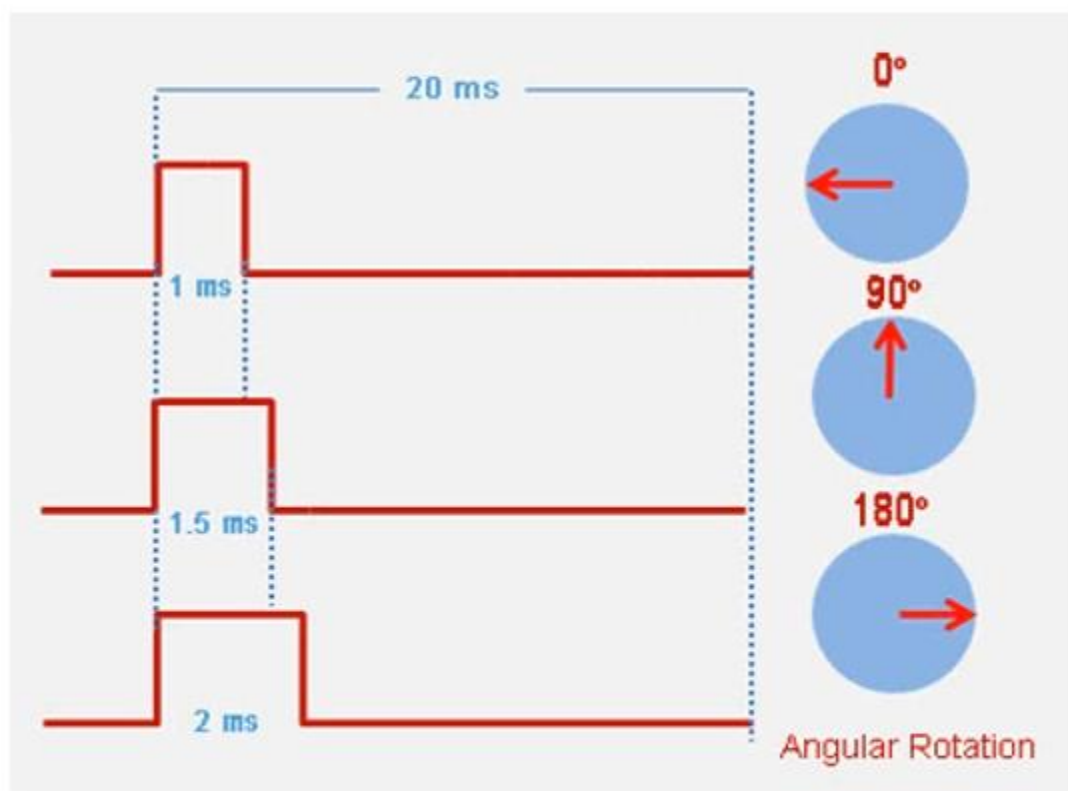
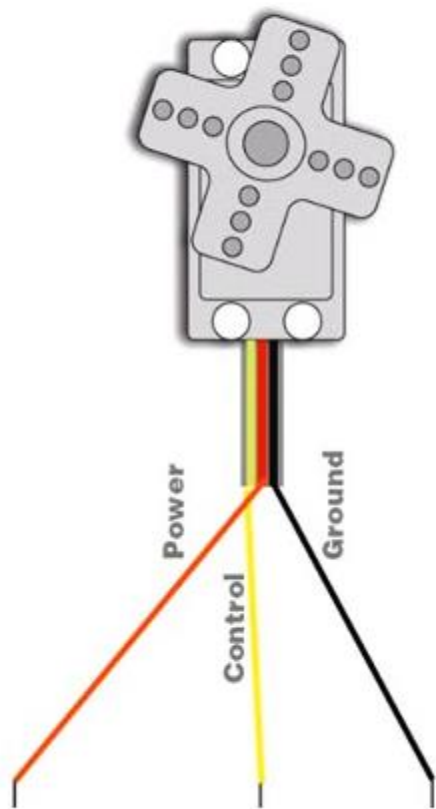


نکته!

- ولتاژ ورودی به VCC نباید از ۵ ولت بیشتر باشد!
- همچنین با توجه به ابعاد و گشتاور موتور باید منبع مناسب تغذیه برای فراهم کردن جریان مصرفی موتور فراهم گردد.

❖ آشنایی با موتورهای الکتریکی - سروو

- کنترل موقعیت سروو موتور



❖ آشنایی با موتورهای الکتریکی - پله ای

• کاربرد در

- حرکت محورهای ظریف دستگاه های CNC
- پرینترهای سه بعدی
- پرینترهای جوهر افشان
- اسکنرها
- پلاترها
- دی وی دی درایوها
- ...

❖ آشنایی با موتورهای الکتریکی - پله ای

• کاربرد در

• حرکت محورهای ظریف دستگاه های CNC

• پرینترهای سه بعدی

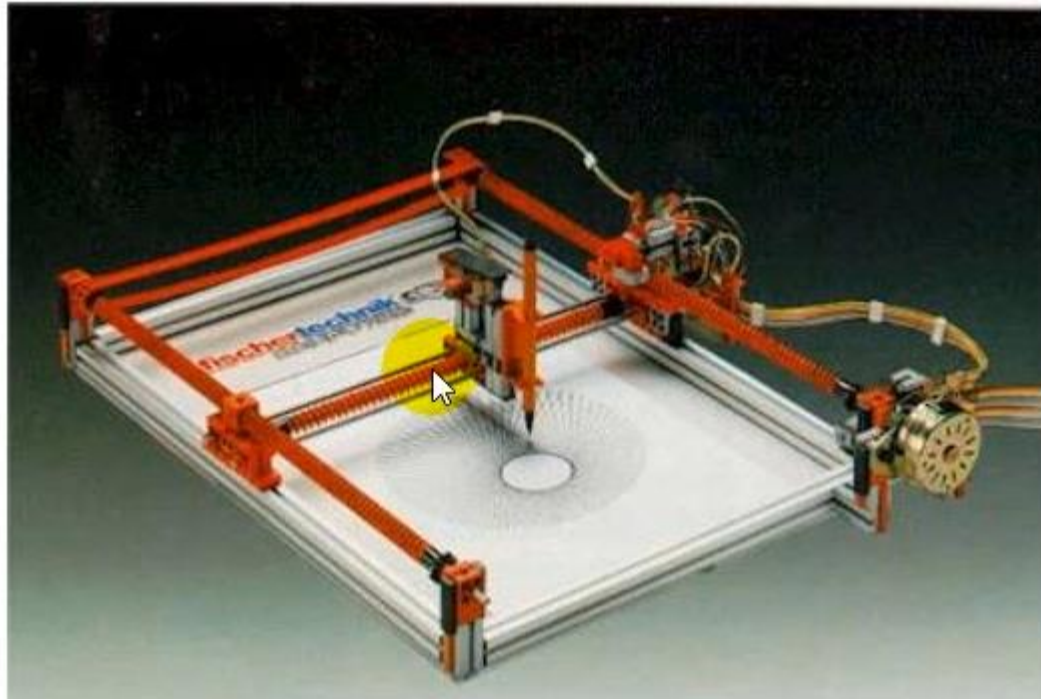
• پرینترهای جوهر افشان

• اسکنرها

• پلاترها

• دی وی دی درایوها

• ...



❖ آشنایی با موتورهای الکتریکی - پله ای

- موتورهای پله ای، 360° درجه را به تعدادی گام های مساوی تقسیم می کند و با هر بار اعمال فرمان به موتور، یک گام برخواهد داشت.
- این موتورها جاروبک ندارند و کنترل موقعیت آن ها بدون فیدبک می باشد.

