

آموزش برنامه ریزی و راه اندازی برد

آردوینو – Arduino

ارایه کننده: احسان شهنازی

درس: مکاترونیک ۱

استاد: آقای دکتر محلوجی

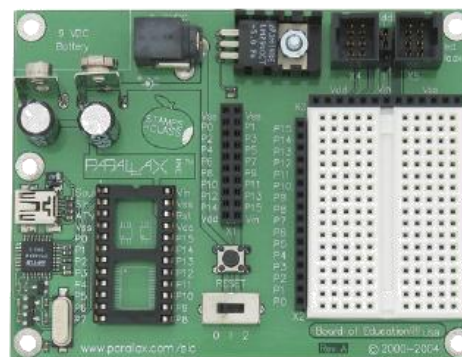
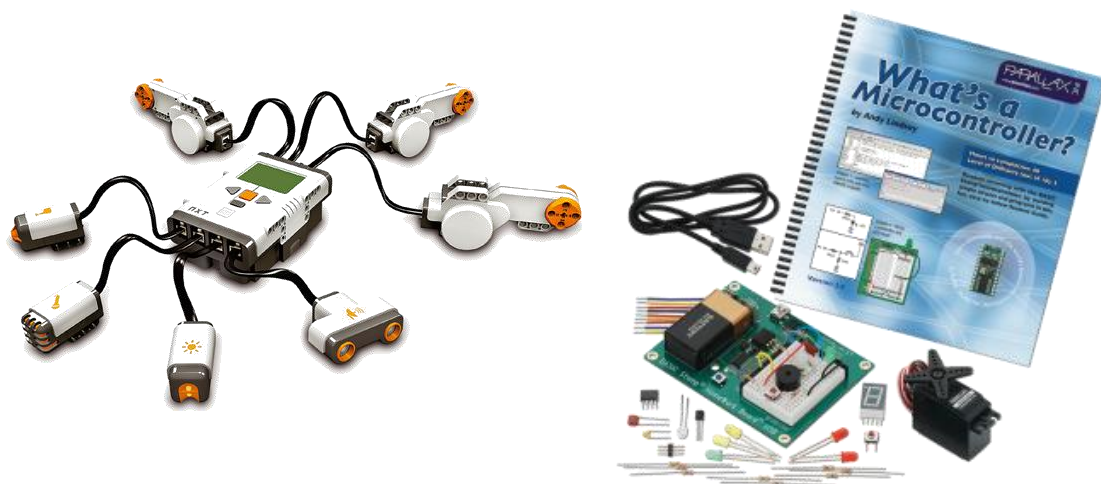
بهار ۱۳۹۴

فهرست موضوعی

- انواع بردهای میکروکنترلری
- آردوینو چیست؟
- آردوینو uno
- نرم افزار فریتزینگ
- نرم افزار آردوینو
- مثالها

انواع بردهای میکروکنترلی آماده

- [Arduino \(and Wiring\)](#)
[Barebone - custom solutions](#)
[BasicStamps - BasicX -](#)
[BasicAtom](#)
[Beagle Board](#)
[BUG](#)
[CUI \(CREATE USB Interface\)](#)
[Eobody2](#)
[Gainer](#)
[I-CubeX](#)
[LEGO Mindstorms](#)
[LittleBits](#)
[Make Controller](#)
[NADA Sketchtools](#)
[NerdKits](#)
[NETLab Toolkit](#)



انواع بردهای میکروکنترلی آماده

Phidgets

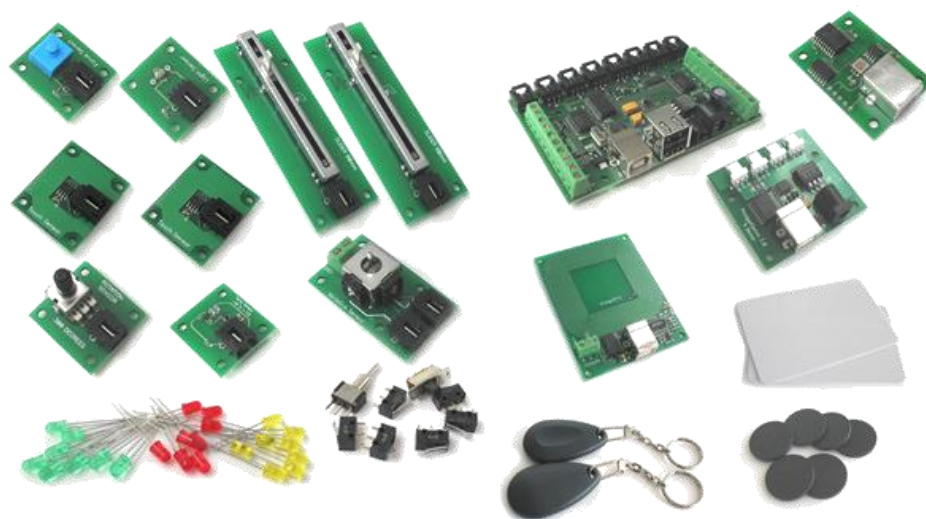
SeeedStudio Electronic Bricks

Snap Circuits

The Discover Electronics Kit

Vex Robotics

ZB₁



- و بسیاری بردهای دیگر، عملکرد مشابهی را ارائه می دهند.
- کلیه این ابزارها، جزئیات درهمی از برنامه نویسی میکروکنترلر را دربرداشته و آن را در یک پکیج easy-to-use جمع بندی نموده اند.

• بردهای میکروکنترلی Arduino



آردوینو چیست؟

- آردوینو ابزاری است برای تولید کامپیوترهایی که نسبت به کامپیوتر شخصی شما، مقدار بیشتری از دنیای فیزیکی را احساس و کنترل می کنند.
- آردوینو می تواند ورودی را از تعداد زیادی سویچ و حسگر بگیرد و سپس کنترل تنوعی از لامپ ها، موتورها، و سایر خروجی های فیزیکی را در اختیار بگیرد.
- برد آردوینو فرایند کار با میکروکنترلرها را تسهیل می کند.
- در مقایسه با سایر پلت فرم های میکروکنترلر، بردهای آردوینو نسبتاً ارزان هستند.
- نرم افزار آردوینو روی سیستم های عامل ویندوز، مکینتاش OS X و لینوکس اجرا می شود.

آردوینو UNO



مشخصات برد آردینو مدل آنو	
میکروکنترلر	ATmega328
ولتاژ عملیاتی	5 ولت
ولتاژ ورودی (پیشنهادی)	7-12 ولت
پین های دیجیتال ورودی/خروجی	۱۴ عدد که ۶ تای آن به عنوان خروجی PWM استفاده می شود.
پین های ورودی آنالوگ	6
جریان DC هر پین ورودی و خروجی	40 میلی آمپر
حافظه فلش	32 کیلوبایت (ATmega328) که 0.5 کیلوبایت از آن مورد استفاده BootLoader قرار می گیرد.
SRAM	2 کیلوبایت (ATmega328)
EEPROM	1 کیلوبایت (ATmega328)
سرعت ساعت	16 مگاهرتز

• نرم افزار فریتزینگ

Fritzing

f
Untitled Sketch 2.fzz - Fritzing - [Breadboard View]
File
Edit
Part
View
Window
Help

f
Welcome
Breadboard
Schematic
PCB

Recent Sketches

robot_fritzing.fzz

New Sketch
Open Sketch

Tip of the Day:

To delete a bendpoint from a wire, double-click it.

All Tips
Next Tip

Projects | Blog

Unable to reach blog.fritzing.org

Fritzing News.
fritzing
Blog

Fab | Shop



Fritzing CreatorKit

The Fritzing Creator Kit provides a fast, easy and entertaining introduction to the world of interactive electronics.

order now >>

Get your Creator Kit now.
fritzing
Shop

Parts

Core Parts

CORE
MINE
PA

Textile
ICs
CONTRIB

Inspector

Breadboard1

v. 4

Breadboard1

Placement

location
-0.052
0.000

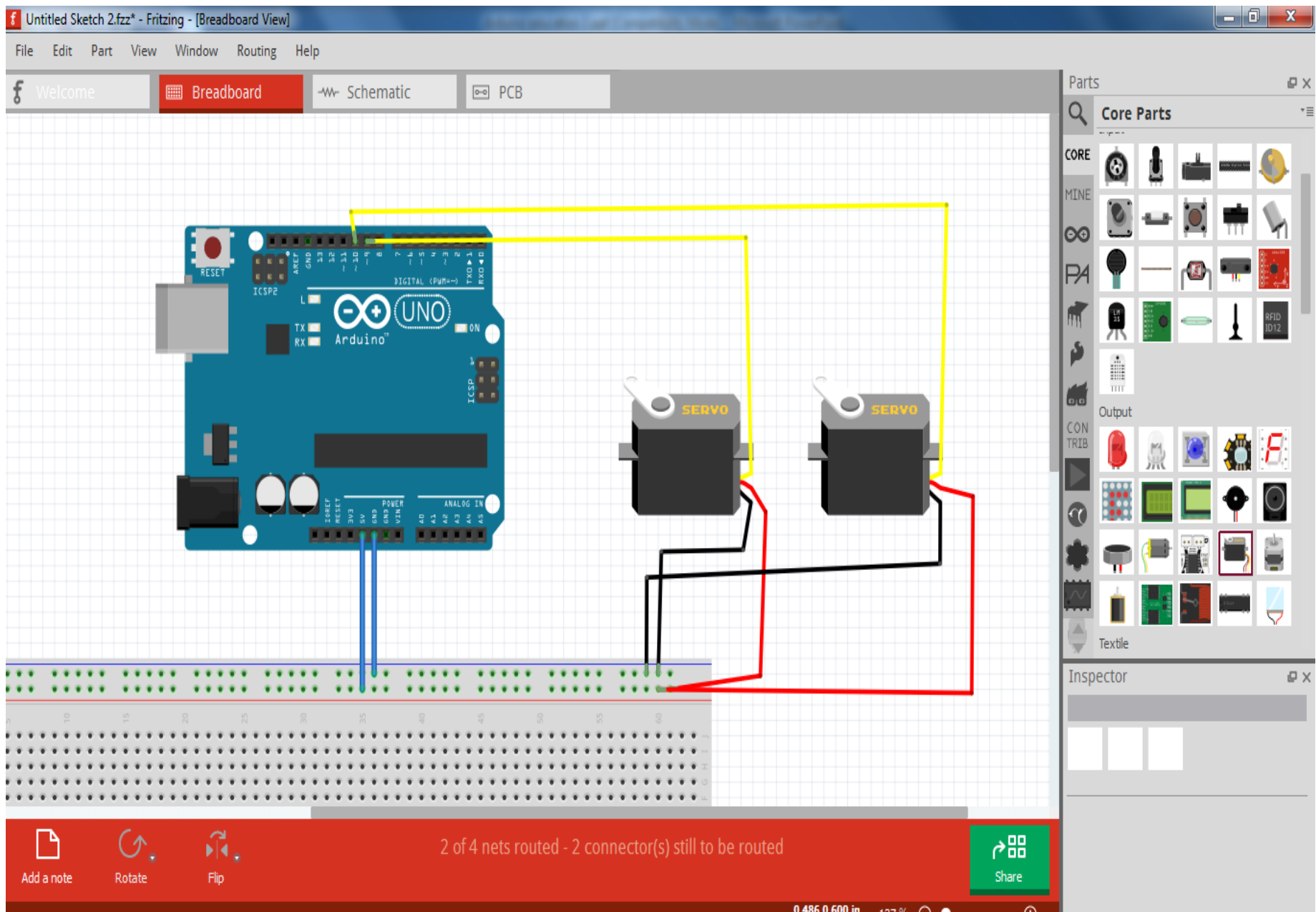
rotation
0.0

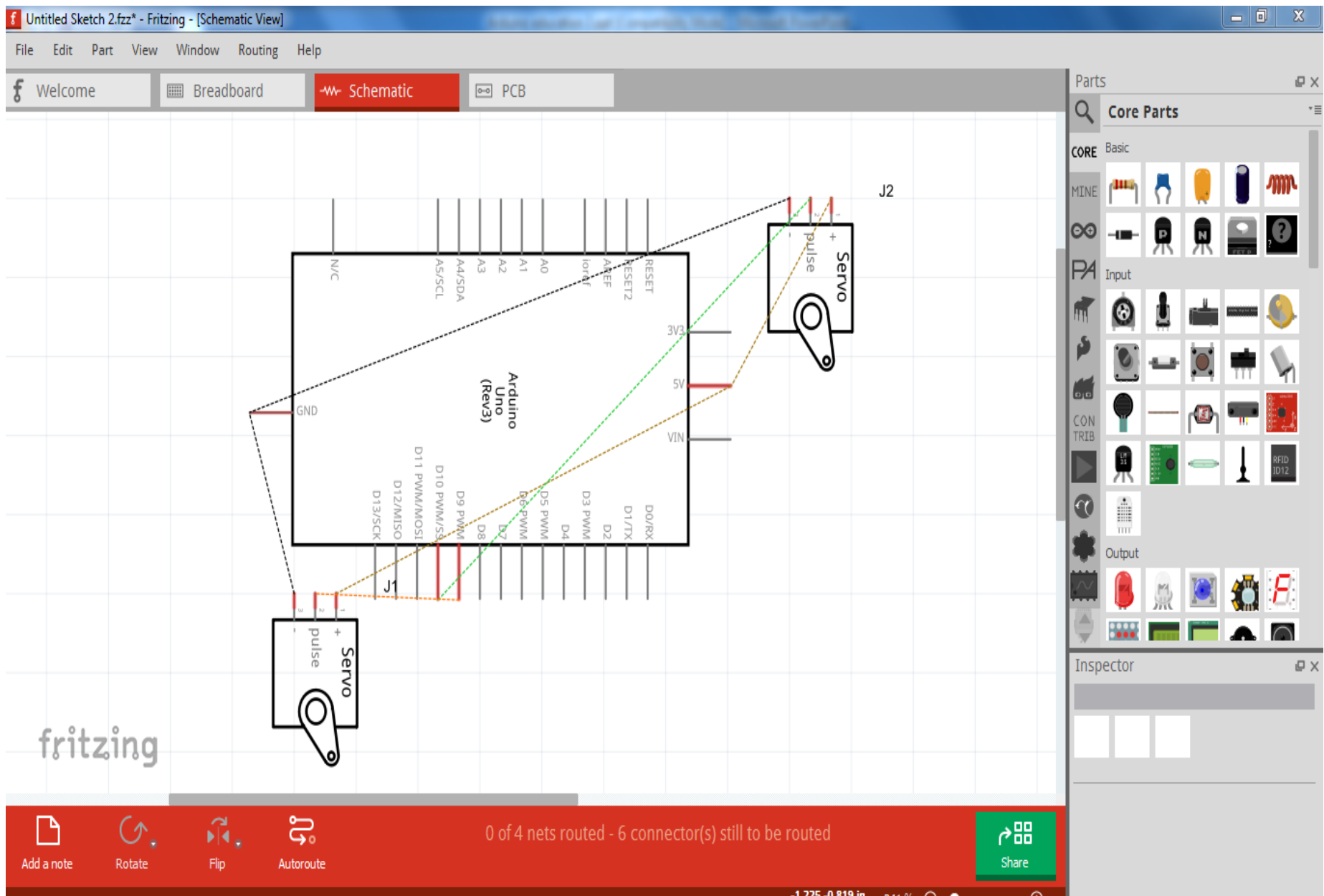
Locked

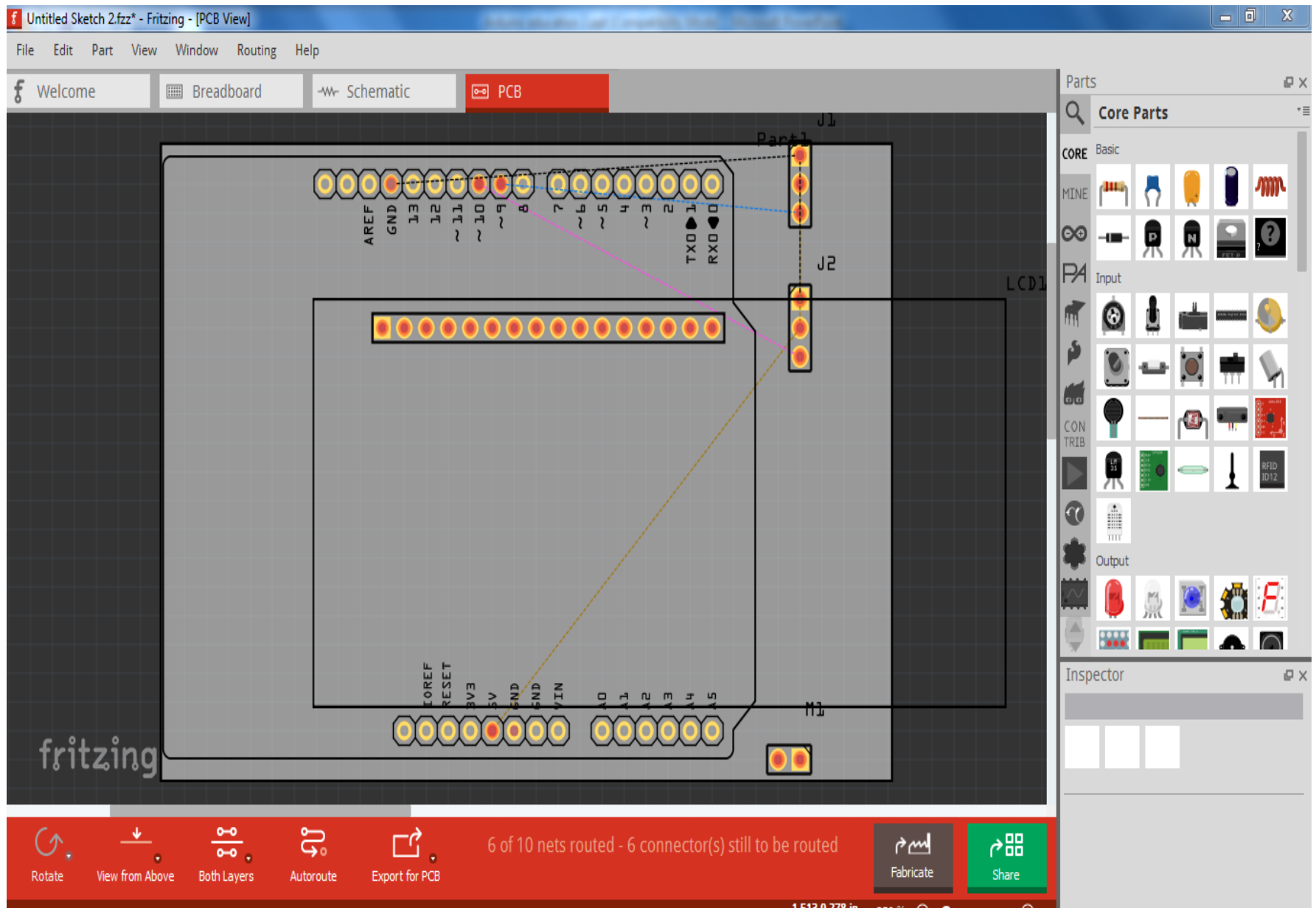
Properties

۱۰

www.mecha.blog.ir





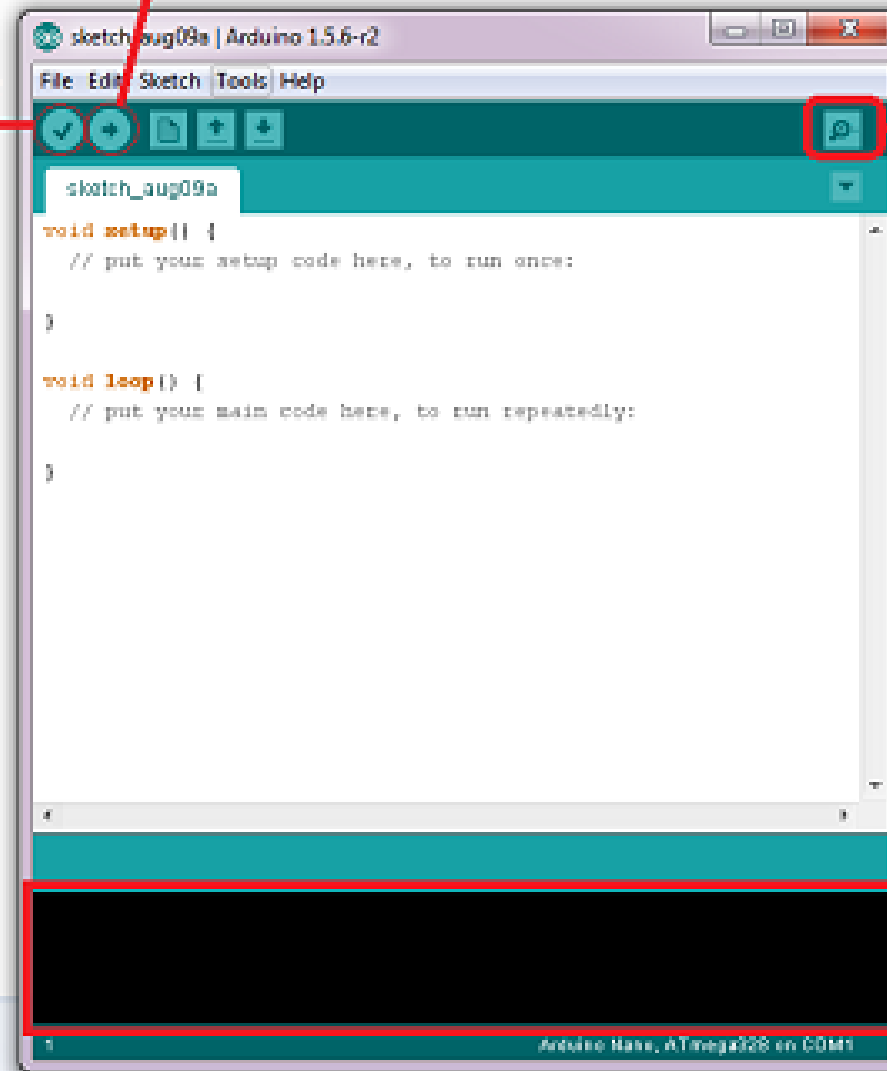


• محیط نرم افزار آردینو

Arduino

وقتی برنامه را نوشتید
با این برنامه را روی
اردوینو آپلود می کنیم

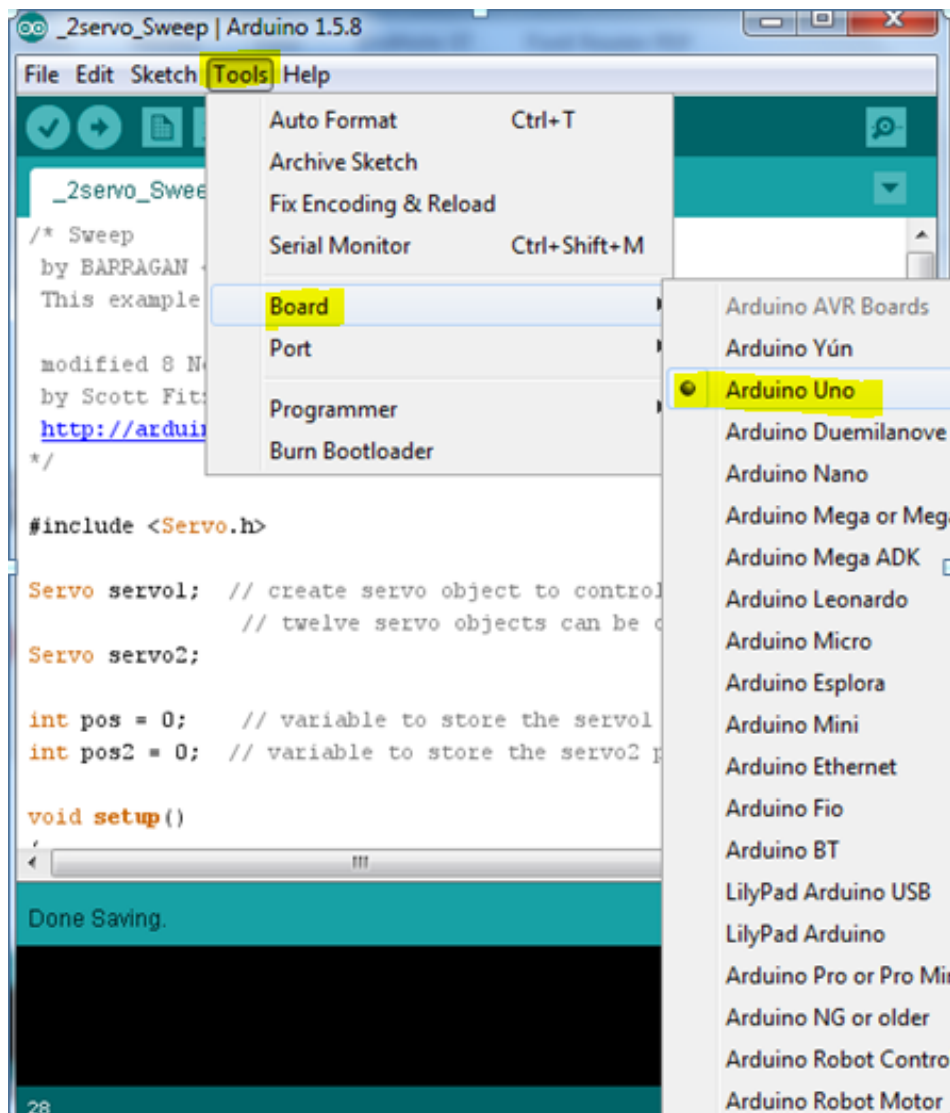
کامپایلر کردن و اشکال
زدایی برنامه



با زدن این دکمه نرمینال برای
ارتباط سریال ظاهر می شود

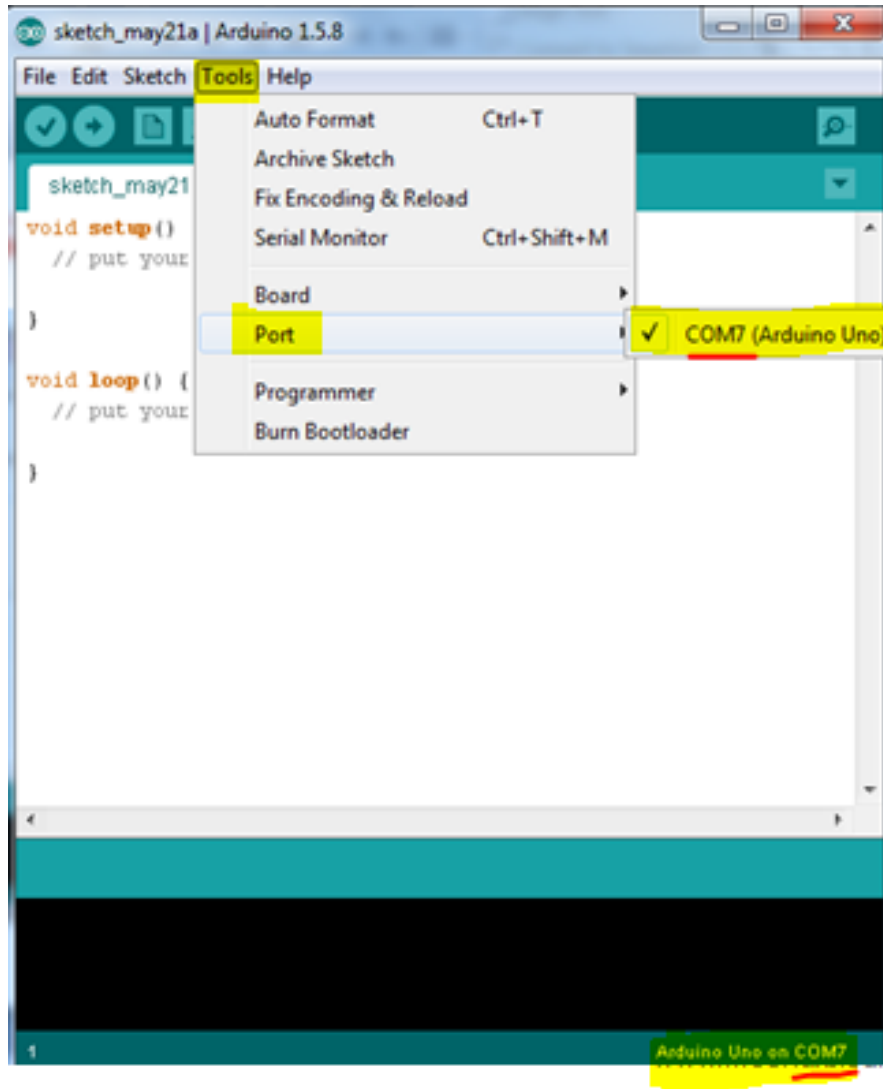
در این قسمت خطا
ها و پیام ها نمایش
داده می شوند

انتخاب نوع برد آردینو



- ۱ - ابتدا برد آردوینو را با کابل USB به لب تاب یا کامپیوتر وصل کنید.
- ۲ - از گزینه های روبرو Tools و سپس board را انتخاب کنید ، لیستی از برد های آردوینو ظاهر می شود ، مدل UNO را انتخاب کنید .

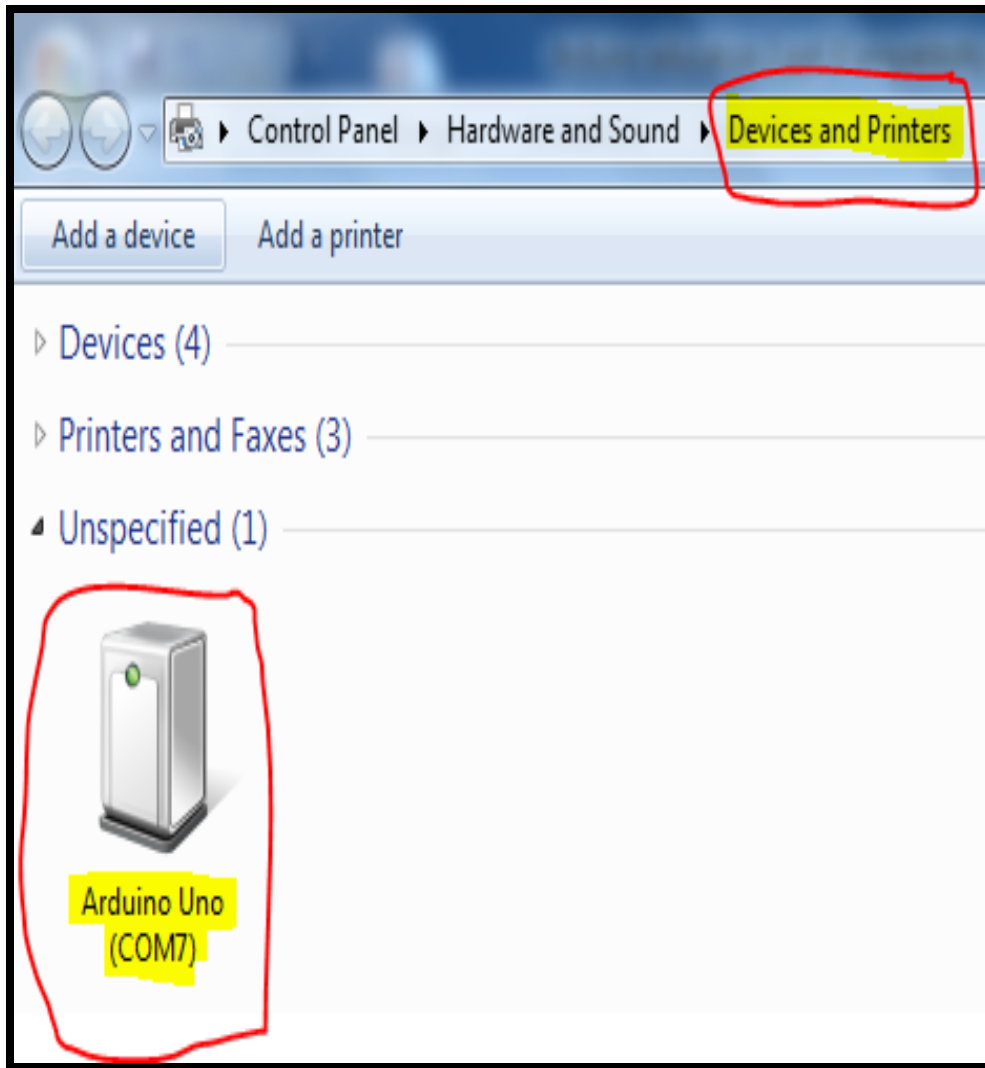
انتخاب پورتی که آردوینو به آن متصل شده است



• بعد از اتصال آردوینو به کامپیوتر از مسیر مشخص شده در شکل روبرو، پورتی را که کامپیوتر به آردوینو اختصاص داده را مشخص می کنیم.

به این ترتیب برد آردوینو از طریق این پورت برنامه ریزی می شود.

یافتن پورت اختصاص داده شده به آردوینو



اگر نتوانستیم به درستی پورتی را که به آردوینو متصل شده است را تشخیص دهیم می توانیم؛
از مسیر

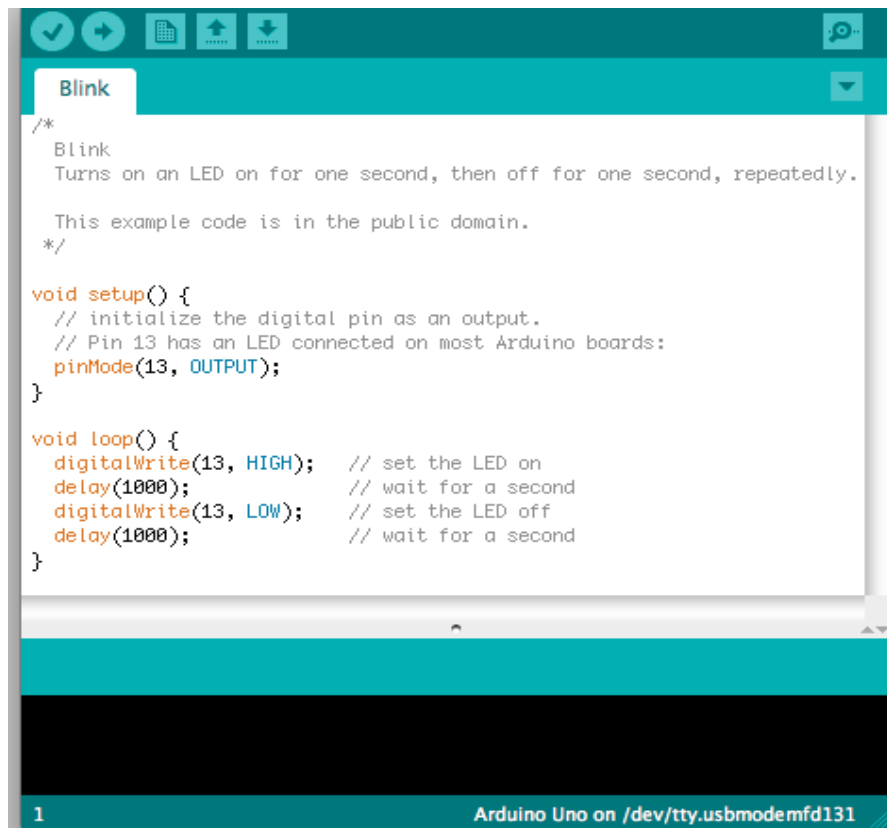
Start / Device and Printers

پورتی را که به آردوینو اختصاص داده شده است را ببینیم.

مثال blink را باز کنید

• نمونه برنامه LED blink را باز کنید.

• File / Examples / Basics / Blink.

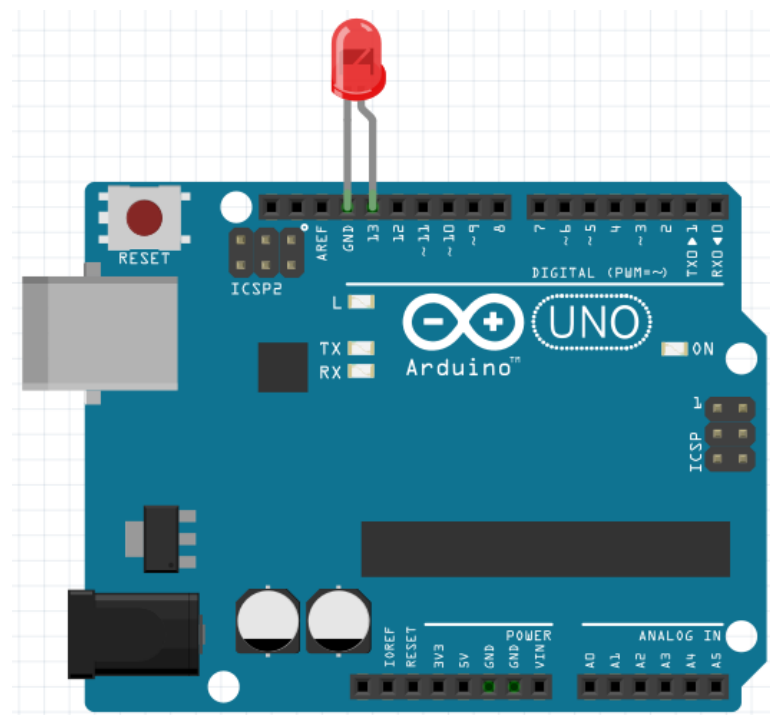


```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repeatedly.
 *
 * This example code is in the public domain.
 */

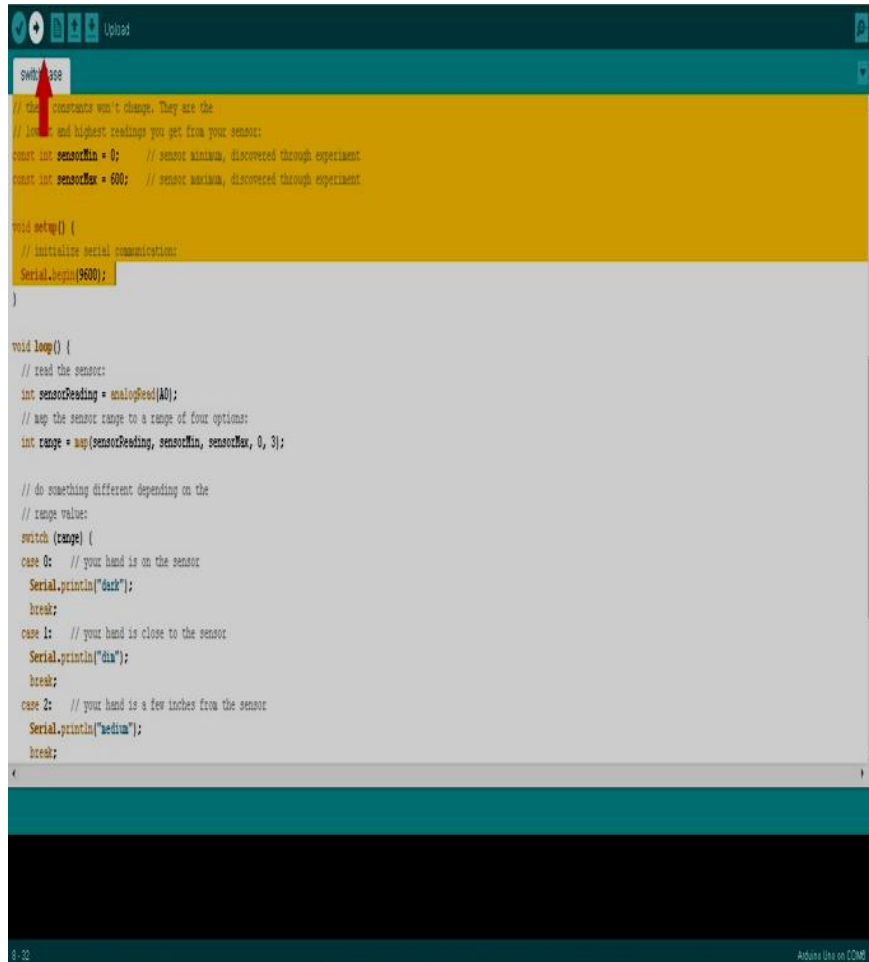
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);             // wait for a second
  digitalWrite(13, LOW);  // set the LED off
  delay(1000);             // wait for a second
}
```

1 Arduino Uno on /dev/tty.usbmodemfd131



برنامه را آپلود کنید



```
// these constants won't change. They are the
// lowest and highest readings you get from your sensor:
const int sensorMin = 0;    // sensor minimum, discovered through experiment
const int sensorMax = 600;  // sensor maximum, discovered through experiment

void setup() {
  // initialize serial communications:
  Serial.begin(9600);
}

void loop() {
  // read the sensor:
  int sensorReading = analogRead(A0);
  // map the sensor range to a range of four options:
  int range = map(sensorReading, sensorMin, sensorMax, 0, 3);

  // do something different depending on the
  // range value:
  switch (range) {
    case 0: // your hand is on the sensor
      Serial.println("dark");
      break;
    case 1: // your hand is close to the sensor
      Serial.println("dim");
      break;
    case 2: // your hand is a few inches from the sensor
      Serial.println("medium");
      break;
  }
```

- اکنون، خیلی ساده، روی دکمه Upload در محیط آردوینو کلیک کنید. چند ثانیه صبر کنید- باید چراغ‌های RX و TX روی برد را به حالت چشمک زن ببینید.

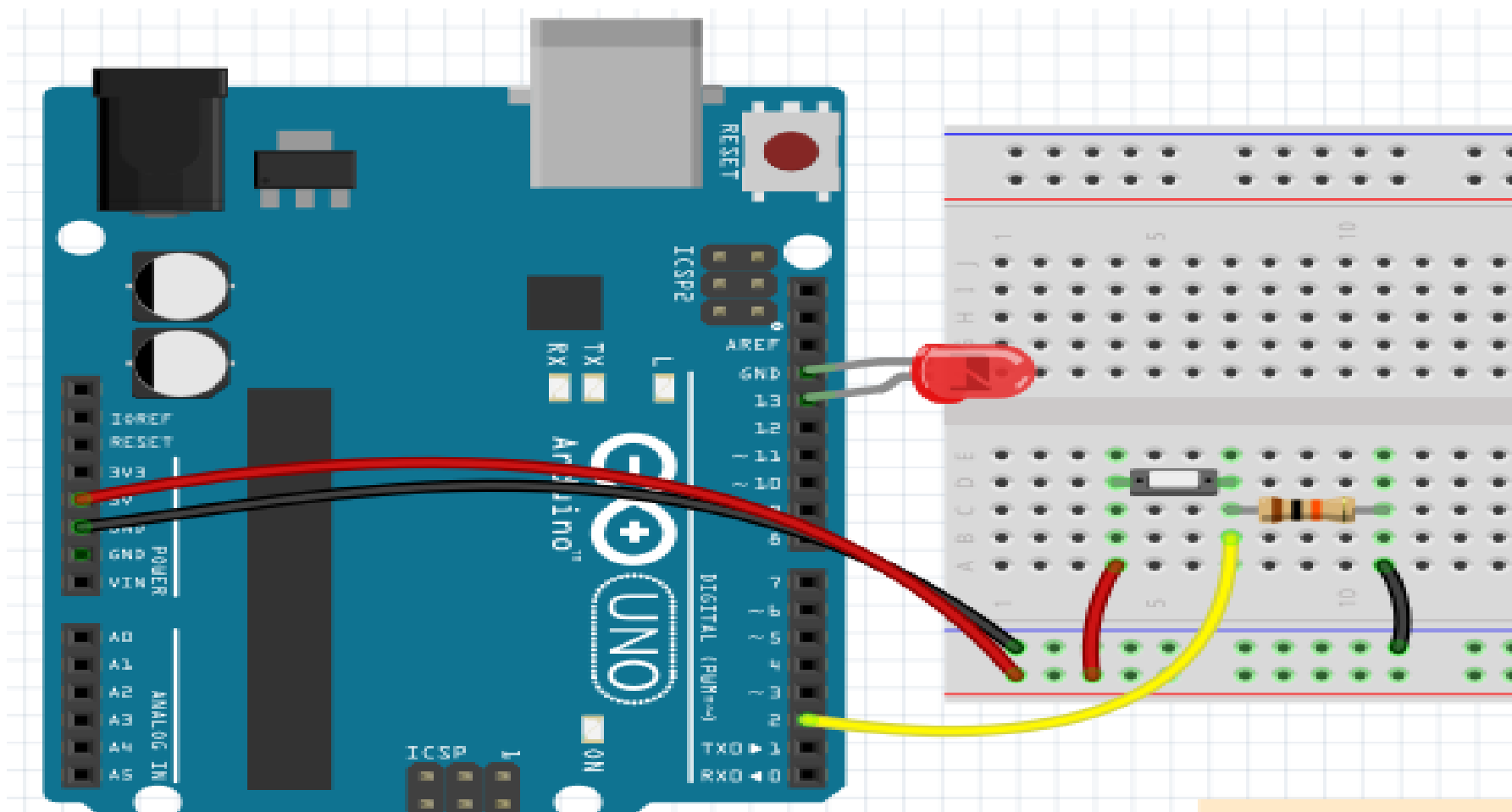
- اگر عملیات آپلود با موفقیت انجام شود، پیغام Done uploading در نوار وضعیت ظاهر می شود.

راه اندازی مدار

- چند ثانیه پس از اینکه آپلود تمام شد، LED که به پین ۱۳ برد متصل است (L)، باید به حالت چشمک زن درآید.
- اگر این اتفاق افتاد، به شما تبریک می گوئیم !
- شما آپلود و اجرای برنامه آردوینو را یاد گرفته اید.

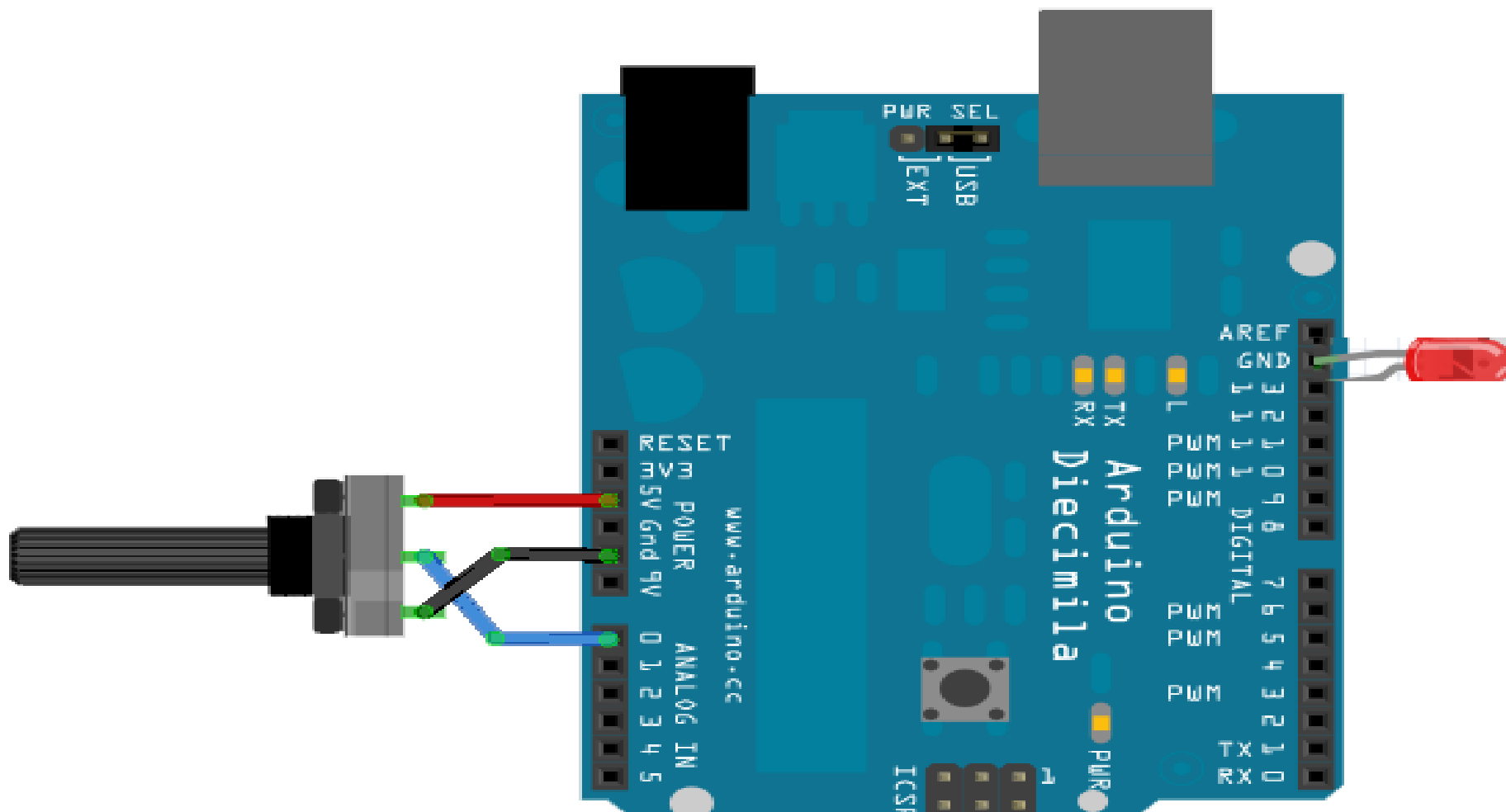
خاموش و روشن کردن ال ای دی توسط شستی

Example / Digital / Button

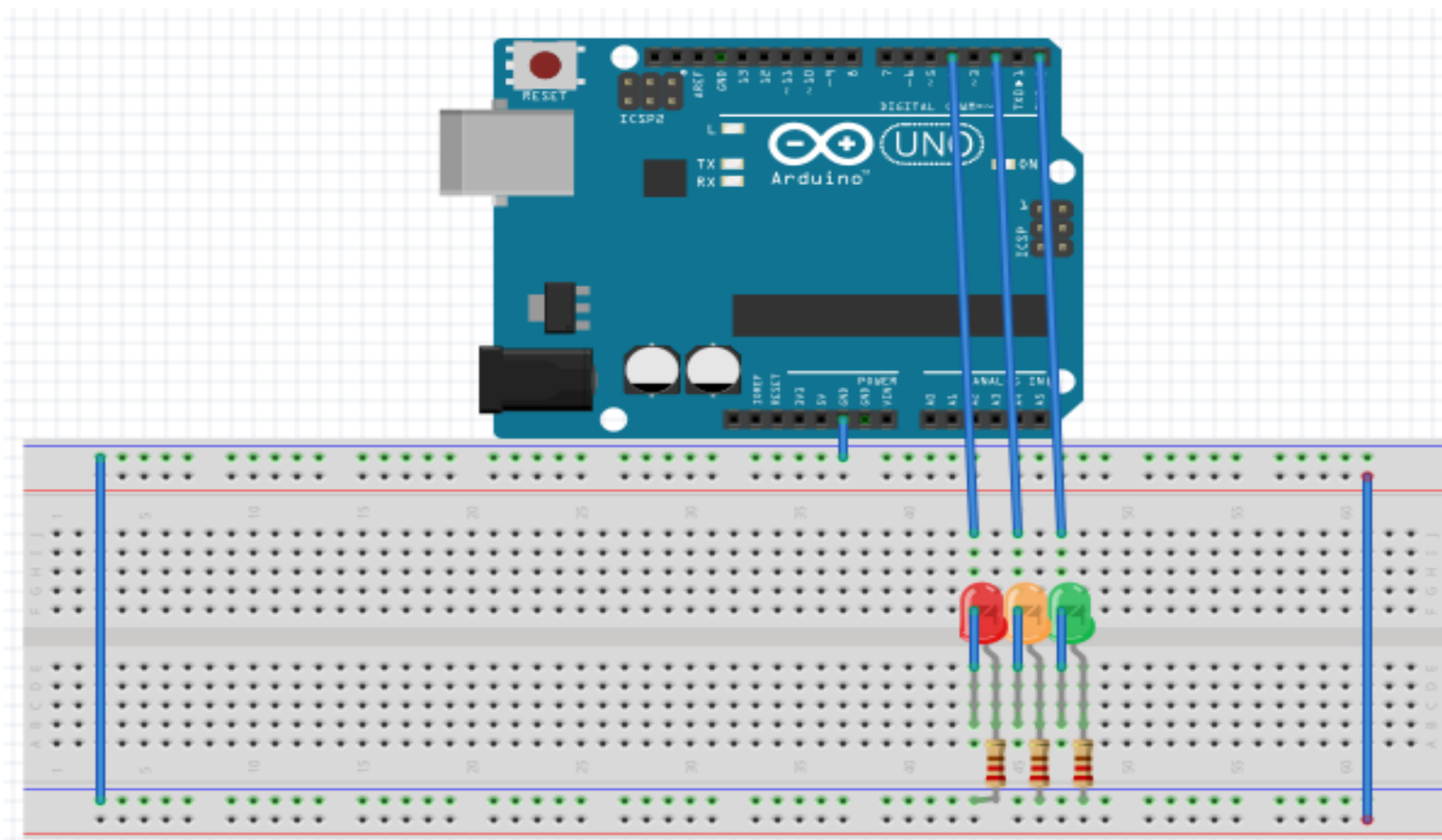


کنترل سرعت چشمک زدن ای دی

Example / Analog / AnalogInput



چراغ راهنمایی و رانندگی



برنامه چراغ راهنمایی و رانندگی

```
/* LED Traffic*/
```

```
void setup() {  
  pinMode(0, OUTPUT);  
  pinMode(2, OUTPUT);  
  pinMode(4, OUTPUT);  
}
```

```
// the loop function runs over and over again forever
```

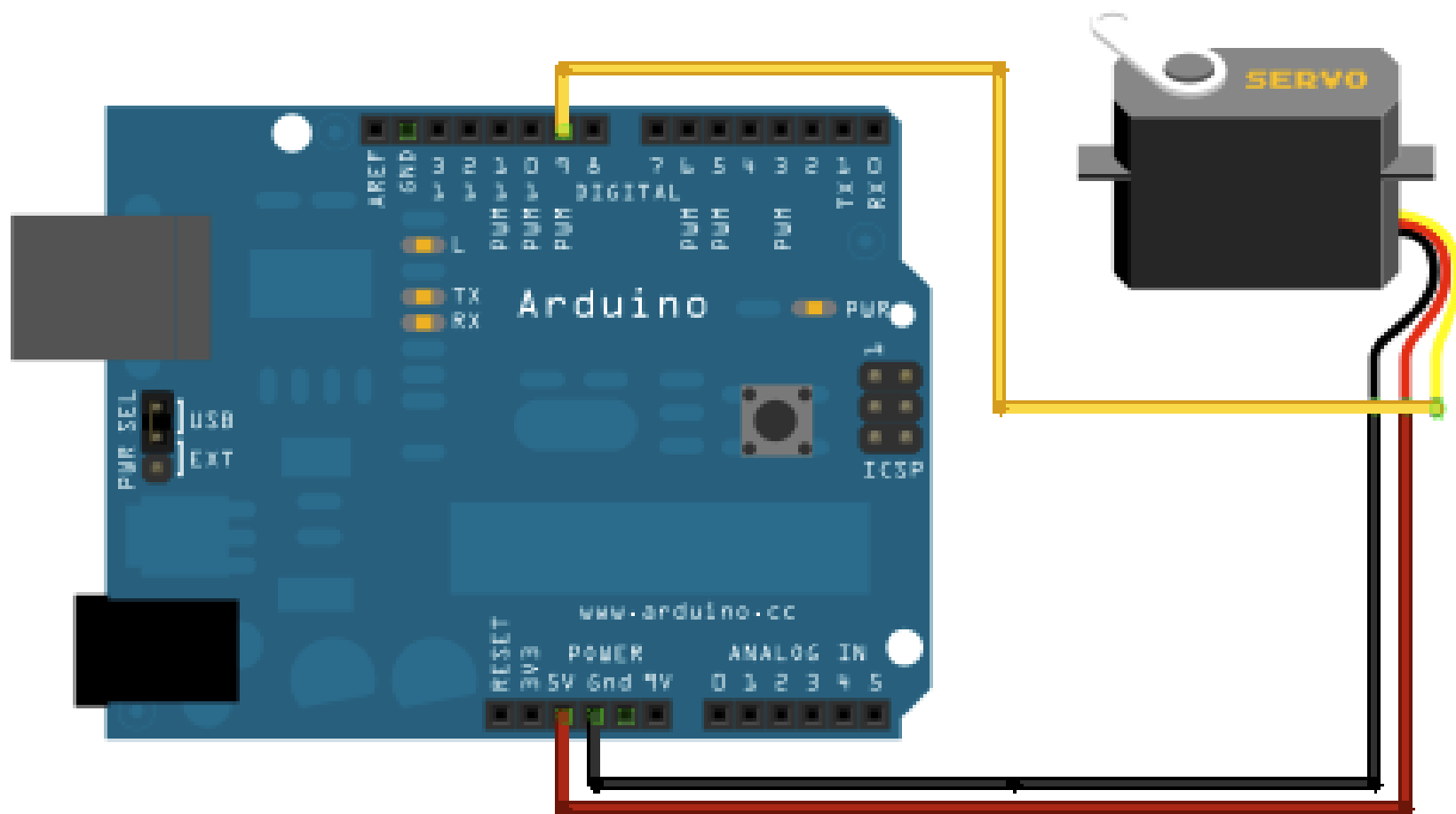
```
void loop() {  
  digitalWrite(0, HIGH); // turn the LED on (HIGH is the  
  voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(0, LOW);  // turn the LED off by making  
  the voltage LOW
```

```
  
  digitalWrite(2, HIGH); // turn the LED on (HIGH is the  
  voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(2, LOW);  // turn the LED off by making  
  the voltage LOW
```

```
  
  digitalWrite(4, HIGH); // turn the LED on (HIGH is the  
  voltage level)  
  delay(1000);           // wait for a second  
  digitalWrite(4, LOW);  // turn the LED off by making  
  the voltage LOW  
}
```

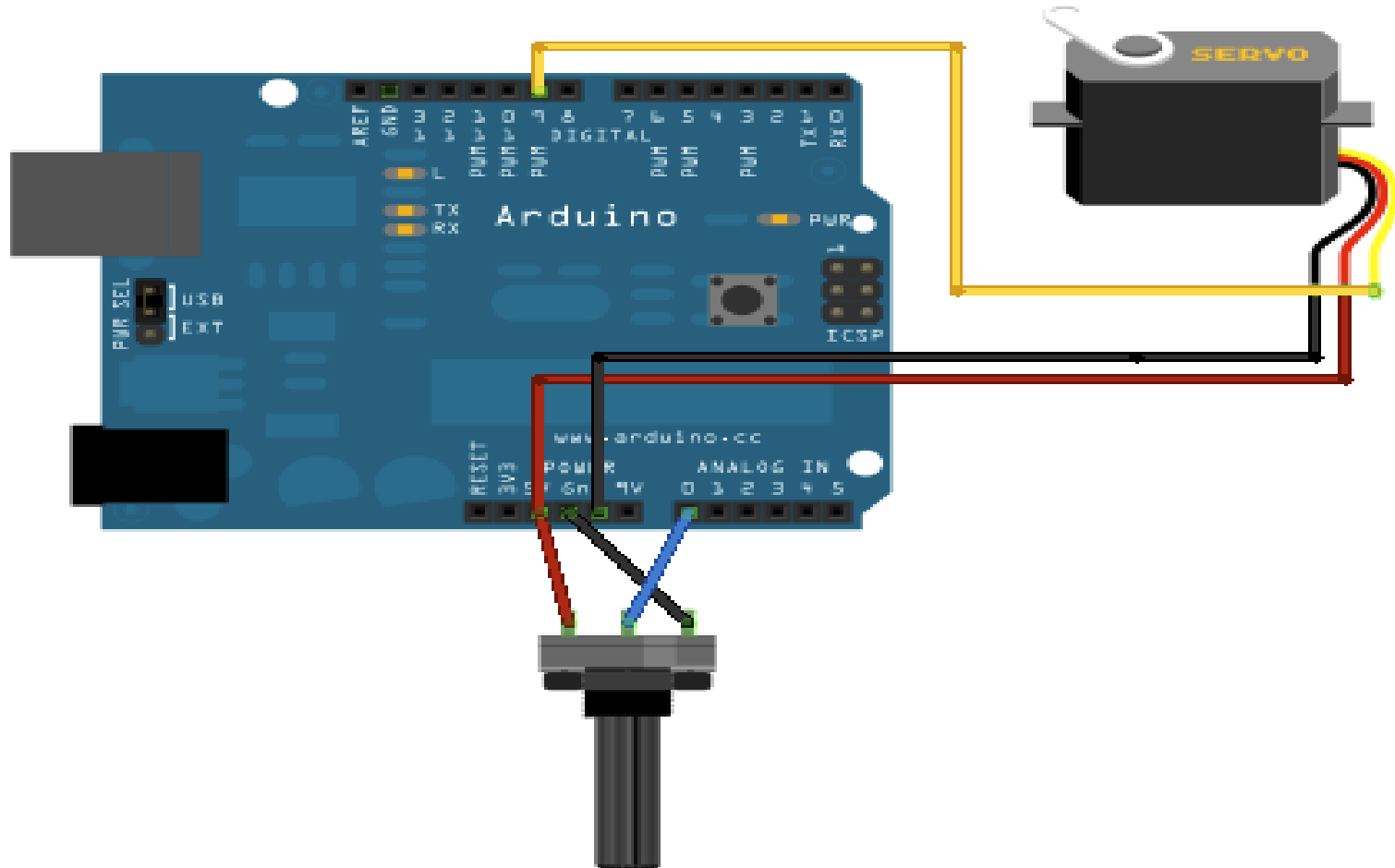
کنترل سروو از طریق برنامه

Example / Servo / Sweep

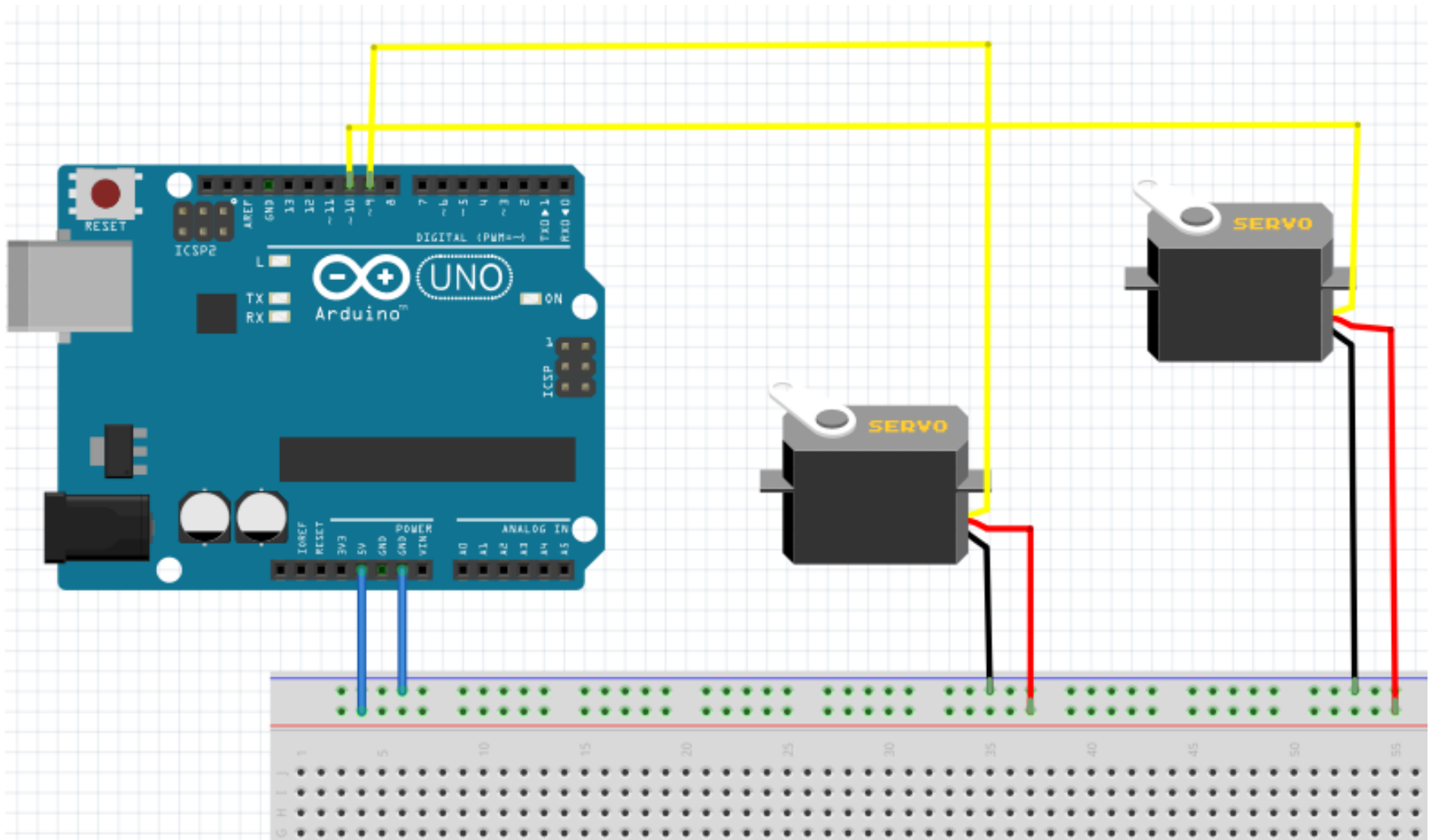


کنترل سروو موتور با ولوم

Example / Servo / Knob



کنترل دو عدد سرو موتور



برنامه کنترل دو عدد سرو موتور

```
/*Drive 2 servo motor*/

#include <Servo.h>

Servo servo1; // create servo object to control a servo
               // twelve servo objects can be created on most
               boards
Servo servo2;

int pos = 0; // variable to store the servo1 position
int pos2 = 0; // variable to store the servo2 position

void setup()
{
  servo1.attach(9); // attaches the servo on pin 9 to the
                    servo object
  servo2.attach(10); // attaches the servo on pin 10 to
                     the servo object
}
```

```
void loop()
{
  for(pos = 0; pos <= 180; pos += 50) // goes from 0 degrees to 180 degrees
  {
    // in steps of 1 degree
    servo1.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }
  for(pos = 180; pos >= 0; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    servo1.write(pos); // tell servo to go to position in variable 'pos'
    delay(15); // waits 15ms for the servo to reach the position
  }

  for(pos2 = 0; pos2 <= 180; pos2 += 1) // goes from 0 degrees to 180 degrees
  {
    // in steps of 1 degree
    servo2.write(pos2); // tell servo to go to position in variable 'pos'
    delay(5); // waits 15ms for the servo to reach the position
  }
  for(pos2 = 180; pos2 >= 0; pos2 -= 1) // goes from 180 degrees to 0 degrees
  {
    servo2.write(pos2); // tell servo to go to position in variable 'pos'
    delay(5); // waits 15ms for the servo to reach the position
  }
}
```

پایان