

# ARDUINO SIMULATOR 1.8

<https://ceil.ir>

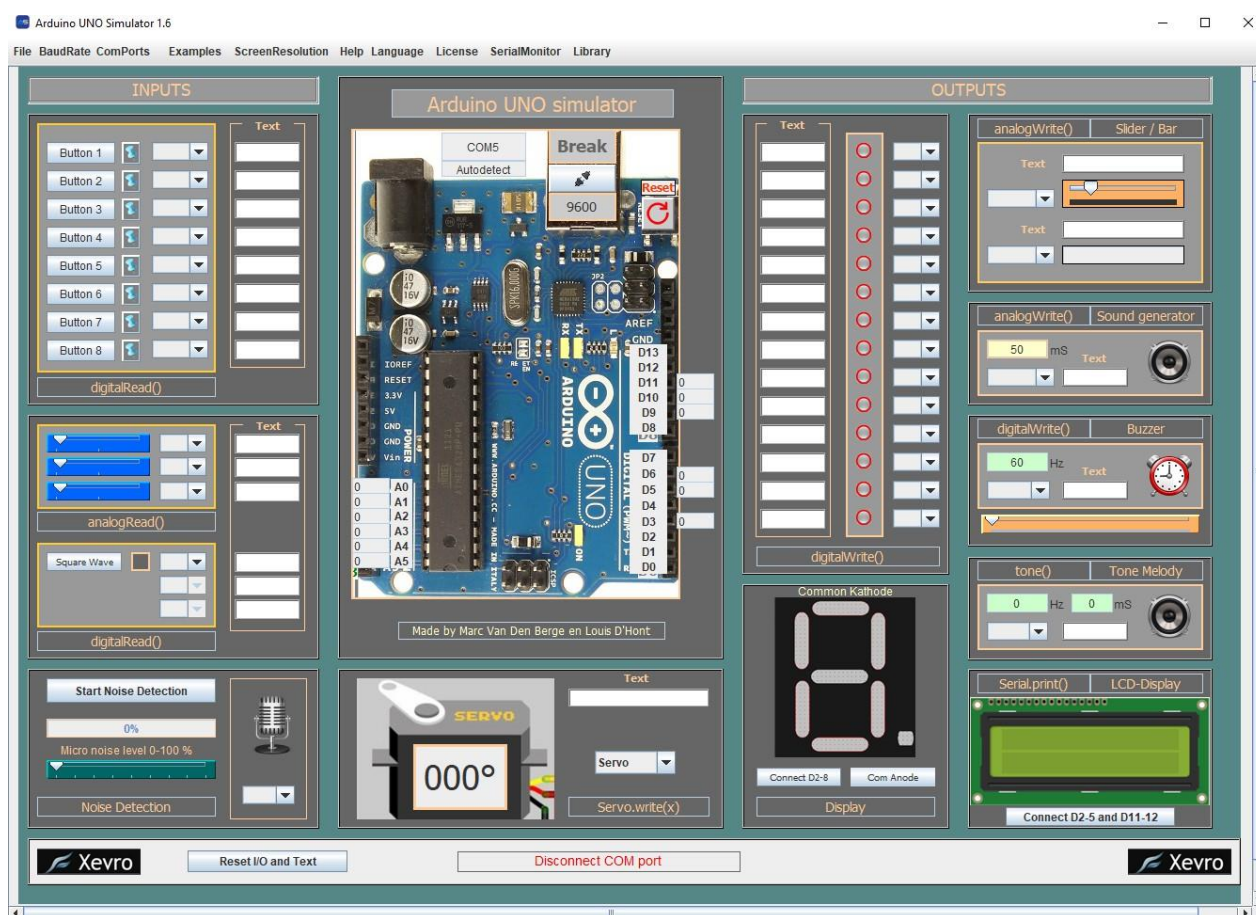


نسخه 1.8

Xevro© 2021

Louis D'Hont & Marc Van Den Berge

در این راهنما همه ویژگی‌ها و قابلیت‌های شبیه‌ساز آردوینو بیان می‌شود.



Arduino Simulator ابزارها و اجزای مورد نیاز برای شبیه سازی Arduino IO را در اختیار شما قرار می دهد. این شبیه ساز برای آزمایش های سریع و پروژه های کوچک ساخته شده است و هنوز هم برای دستیابی به گسترده ترین توابع ممکن در حال به روز رسانی است.

شبیه ساز آردوینو به این دلیل طراحی شده است تا برنامه آردوینو را به سرعت با برد آردوینو بدون اتصال به IO خارجی (دکمه ها، پتانسیومترها، LED ها، نمایشگرهای LCD و ...) آزمایش کند.

برای استفاده از شبیه ساز ما به ۳ برنامه احتیاج داریم:  
جاوا JRE (محیط اجرای جاوا)  
برنامه شبیه ساز آردوینو  
آردوینو IDE

برای استفاده از شبیه ساز باید Java JRE را در رایانه خود نصب کنیم، می توانید لینک دانلود را در وب سایت Xevro پیدا کنید یا عبارت 'دانلود جاوا JRE' را جستجو کنید.

## آردوینو IDE

قبل از شروع استفاده از شبیه ساز آردوینو، به نرم افزار توسعه آردوینو نیاز داریم، که در وب سایت آردوینو رایگان موجود است :

<https://arduino.cc/en/Main/Software>



### Download the Arduino Software

**ARDUINO 1.6.0**  
The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.  
This software can be used with any Arduino board. Refer to the Getting Started page for installation instructions.

**Windows Installer**  
Windows ZIP file for non-admin install

Mac OS X for Java 6 (recommended)  
Mac OS X for Java 7+ (experimental)

Linux 32 bits  
Linux 64 bits

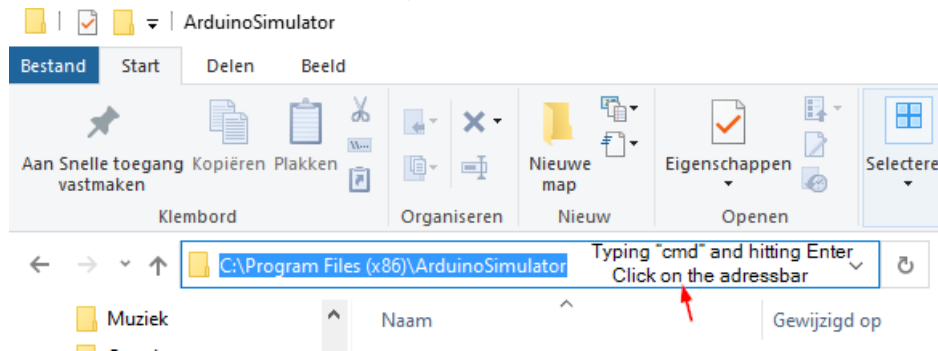
Release Notes Source Code

## چگونه شبیه ساز را از خط فرمان راه اندازی کنیم

می توانید پنجره Command Prompt را مستقیماً از داخل پنجره Windows Explorer باز کنید. شما را مستقیماً به محل پوشه می برد!

اگر روی این نوار آدرس کلیک کنید، می توانید متن را تایپ کنید. با تایپ «cmd» و زدن Enter، خط فرمان را در آن مکان باز می کنید.

به دایرکتوری C:\Program Files (x86)\ArduinoUNOsimulator بروید.



نسخه جاوا را بررسی کنید **java -version + ENTER** :

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. Alle rechten voorbehouden.

C:\Program Files (x86)\ArduinoSimulator>java -version
java version "1.8.0_281"
Java(TM) SE Runtime Environment (build 1.8.0_281-b09)
Java HotSpot(TM) Client VM (build 25.281-b09, mixed mode)

C:\Program Files (x86)\ArduinoSimulator>
```

**java -jar ArduinoSimulator.jar + ENTER** : شبیه ساز را باز کنید.

```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19041.985]
(c) Microsoft Corporation. Alle rechten voorbehouden.

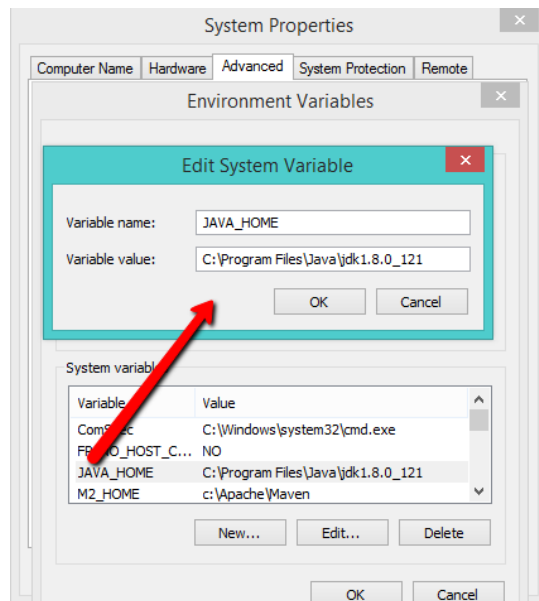
C:\Program Files (x86)\ArduinoSimulator>java -jar ArduinoSimulator.jar
```

## اگر جاوا در خط فرمان کار نمی کند

دکمه‌های « windows » و « pause » را همزمان روی صفحه کلید خود فشار دهید، با این کار کادر گفتگوی تنظیمات سیستم ظاهر می‌شود. در تب Advanced ، Environment Variables را پیدا کنید.

- در بخش User (بالا)، دو متغیر زیر را ایجاد یا به روز کنید :
- محل نصب جاوا = JAVA\_HOME (به عنوان مثال ، C:/Java/SDK)
- PATH = %JAVA\_HOME%/bin

دیالوگ ها را ببندید

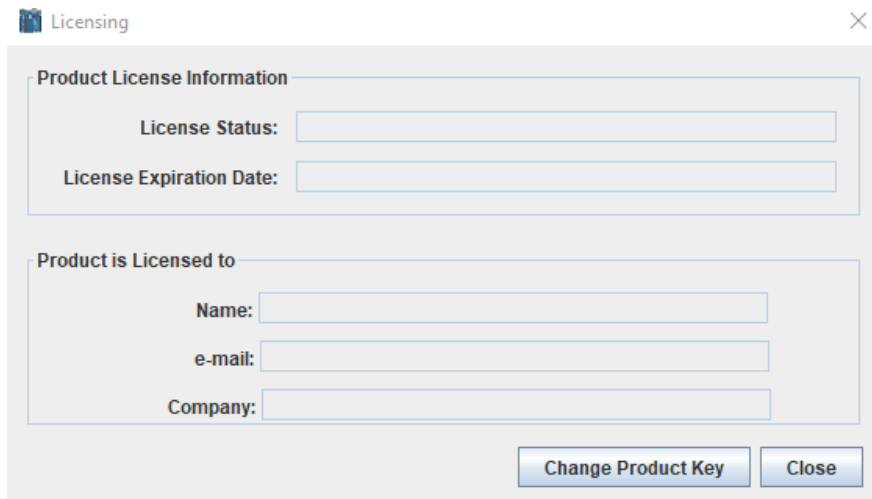


را امتحان کنید و ببینید آیا شناسایی شده است یا خیر. "java -version" سپس، در یک کنسول خط فرمان جدید، مهم است که از یک کنسول جدید استفاده کنید (پنجره کامند قبلی را ببندید)، زیرا متغیرهای محیط تنها زمانی که کنسول راه اندازی می شود خوانده می شوند.

## مجوز ثبت نام

Arduino Simulator به صورت رایگان در دسترس است اما ما آن را با یک کلید مجوز ایمن کردیم. اولین باری که برنامه را باز می کنید یک صفحه فعال سازی ظاهر می شود که می توانید مجوز (لایسنس) را در آن قرار داده و آن را فعال کنید.

برای فعال سازی کلید مجوز را در وب سایت کپی کنید سپس روی «change product key» کلیک کنید، پس از وارد کردن آن باید روی «ok» کلیک کنید.



Licensing

Product License Information

License Status:

License Expiration Date:

Product is Licensed to

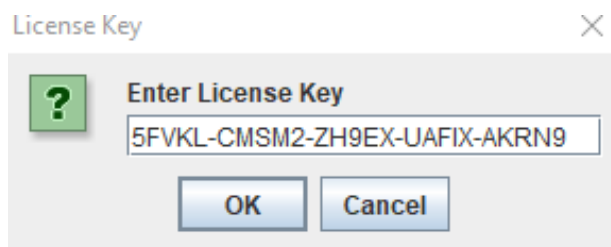
Name:

e-mail:

Company:

Change Product Key Close

مجوز کلید ورودی زمینه



License Key

Enter License Key

5FVKL-CMSM2-ZH9EX-UAFIX-AKRN9

OK Cancel

## تغییرات کد

با **Arduino IDE** دستورالعمل‌هایی کار می‌کند که **IO** می‌خواند و می‌نویسد، با افزودن کتابخانه‌ها به پروژه شما می‌توانید پروژه را شبیه‌سازی کنید.

ما نمی‌خواهیم دستورالعمل‌های واقعی را تغییر دهیم، تصمیم گرفتیم کتابخانه‌ها را ویرایش کنیم تا با نرم‌افزار ما سازگار باشند. کتابخانه هسته **SimulatorProgram** در اولین باری که شبیه‌ساز را باز می‌کنید به پوشه **Arduino/libraries** در پوشه اسناد اضافه می‌شود. در نرم‌افزار **Simulator** تابعی برای افزودن کتابخانه برای افزودن کتابخانه‌های بیشتر به **Arduino IDE** موجود است.

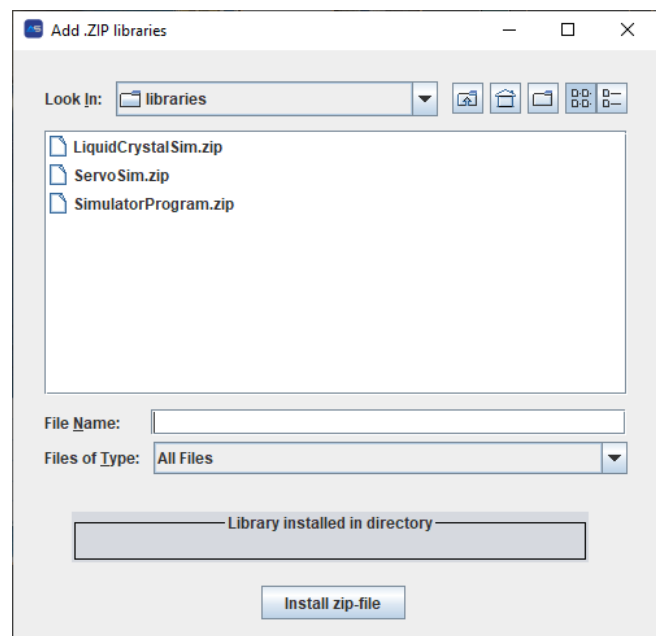
در برنامه شبیه‌ساز در منو **Library** زیرمنوی **install libraries** را انتخاب کنید. سپس همه کتابخانه‌ها را نصب کنید.

### کتابخانه‌های شبیه‌ساز :

\\.\ > Program Files (x86) > ArduinoUNC

Naam

- LiquidCrystalSim
- ServoSim
- SimulatorProgram



به داخل یک پوشه بروید و فایل **zip** را انتخاب کنید، بر روی دکمه "install zip-file" کلیک کنید تا کتابخانه نصب شود. کتابخانه به طور خودکار توسط **Arduino IDE** شناسایی می‌شود. با افزودن عبارت **Arduino include** به پروژه خود، از کتابخانه در پروژه شما استفاده می‌شود و می‌توانید با برد آردوینو شبیه‌سازی کنید.

از **SerialInput ("x")** برای خواندن یک مقدار از خط ارتباط سریال استفاده کنید.

```
if (SerialInput.equalsIgnoreCase("START")){  
    digitalWrite(relais,HIGH);  
}
```

## توجه

به یاد داشته باشید تنظیمات اولیه را برای مقداردهی اولیه IO واقعی انجام دهید.

مثال :

```
pinMode (0, OUTPUT);
```

```
pinMode (1, INPUT);
```

در هر اسکچ (کد) دستورات عمل ها تغییر می کند، بنابراین شما فقط باید ورودی ها و خروجی ها را از طریق صفحه نمایش متصل کنید و اسکچ باید کار کند.

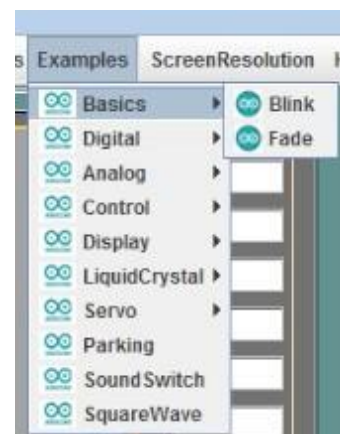
### نحوه استفاده از مثال :

۱. یک اسکچ نمونه را باز کنید و آن را در برد آردوینو آپلود کنید.
۲. اجزای استفاده شده (I/O) را در کاربرگ اضافه کنید. (نمونه هایی وجود دارد که قبلاً در تنظیمات بازیابی ذخیره شده اند)
۳. Arduino IO Simulator را به برد وصل کنید
۴. آردوینو خود را در داخل و خروجی های شبیه ساز شبیه سازی کنید.

## نحوه استفاده از آن

درک شبیه ساز آردوینو بسیار آسان است. شبیه ساز به ۵ چیز ساده نیاز دارد تا به درستی کار کند.

۱. برد آردوینو را وصل کنید.
۲. کد آردوینو سفارشی خود را با فایل کتابخانه مربوطه آپلود کنید.
۳. کد اصلی آردوینو را تغییر دهید.
۴. ورودی های استفاده شده را در شبیه ساز آردوینو انتخاب کنید.



## 1. برد آردوینو را وصل کنید

درک و کار با آردوینو IO Simulator بسیار آسان است. شبیه ساز به ۵ مرحله ساده برای شبیه سازی یک پروژه نیاز دارد.

۱. برد آردوینو را وصل کنید.

۲. کد آردوینو سفارشی خود را با فایل کتابخانه مربوطه آپلود کنید

۳. کتابخانه های استفاده شده را اضافه کنید

۴. خروجی های استفاده شده در Arduino IO Simulator را انتخاب کنید

۵. Arduino IO Simulator را با پورت سریال مناسب به برد Arduino متصل کنید.

## ۱. برد آردوینو را وصل کنید

Arduino IO Simulator با تعداد زیادی برد آردوینو کار می کند:

- Arduino UNO
- Arduino Mega
- Arduino Leonardo
- Arduino...

توجه :

فقط از پین های دیجیتال و آنالوگ موجود در شبیه ساز می توان استفاده کرد! قبل از آپلود کد آردوینو با IDE، Arduino IO Simulator را قطع کنید.

## ۲. کد آردوینو سفارشی خود را با فایل کتابخانه مربوطه آپلود کنید

شبیه ساز را باز کنید و به

"Arduino UNO programming code (ino) -> Arduino UNO programming code -> Help" بروید.

با این کار یک فایل آردوینو (ino) با کتابخانه مربوطه و کدهای مهم در آن باز می شود.

## ۳. کتابخانه های استفاده شده را اضافه کنید

برای اینکه شبیه ساز کد را درک کند، کتابخانه های خود را ایجاد کرده ایم. برای حفظ قابلیت استفاده، تصمیم گرفته ایم دستورالعمل ها را همانطور که هست نگه داریم، اما کتابخانه ها را کمی تغییر دادیم تا با نرم افزار ما سازگار باشند.

چند کتابخانه برای استفاده وجود دارد. کتابخانه برنامه شبیه ساز برای دستورات DigitalWrite... ضروری است. برای استفاده از نمایشگر LCD 16x2 باید کتابخانه liquidCrystalSim ما را اضافه کنید تا از آن با شبیه ساز استفاده کنید. همه دستورالعمل ها یکسان است.

## ۴. خروجی های استفاده شده را در Arduino IO Simulator انتخاب کنید

هر ورودی و خروجی در Simulator یک جعبه انتخاب دارد که می توان پین دیجیتال یا آنالوگ استفاده شده را در آن وصل کرد.

## ۵. Arduino IO Simulator را با پورت سریال مناسب به برد Arduino متصل کنید

Arduino IO Simulator می داند که برد آردوینو کدام پورت است.

هنگام آپلود کد آردوینو، مطمئن شوید که آردوینو قطع شده است.



## برنامه خود را با متغیرهای CHECKVAR بررسی کنید

با Checkvars امکان بررسی وضعیت متغیرهای خود در کد آردوینو وجود دارد. می توانید یک متغیر را روی یک خط در کد خود وارد کنید و از طریق مانیتور سریال می توانید مقادیر متغیرها را دنبال کنید. می توانید متغیرهای مختلفی را به صورت زیر وارد کنید:

Int, long int, long unsigned int, word, double, float, char, string, Boolean

دستورالعمل:

**CheckVar(num, var);**

num: عدد صحیح از ۰ تا ۳۲۷۶۸

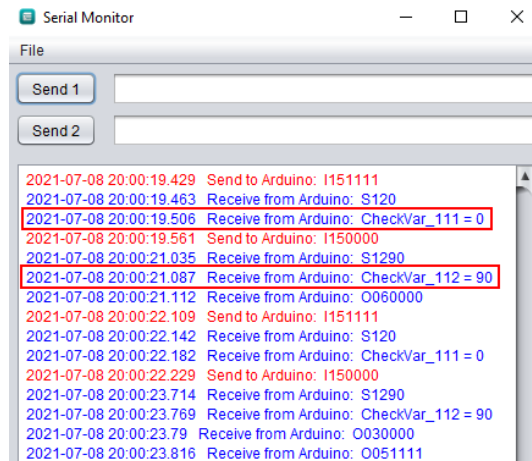
var: Int, long int, long unsigned int, word, double, float, char, string, Boolean

### یک نمونه کد:

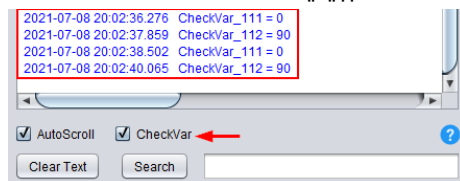
من می خواهم متغیرهای "BarUp" و "BarLow" را در کد بررسی کنم و به CheckVars اعداد ۱۱۱ و ۱۱۲ را برای BarUp و BarLow می دهم.

در مانیتور سریال می توانید متغیرهای CheckVar\_111 و CheckVar\_112 و مقادیر آنها را ببینید.

```
if (digitalRead(Exit) == 1)
{
  if (Available != CAPACITY) {
    Available++;
    myservo.write(BarUp);
    CheckVar(111, BarUp);
    delay(1500);
    myservo.write(BarLow);
    CheckVar(112, BarLow);
  }
}
```



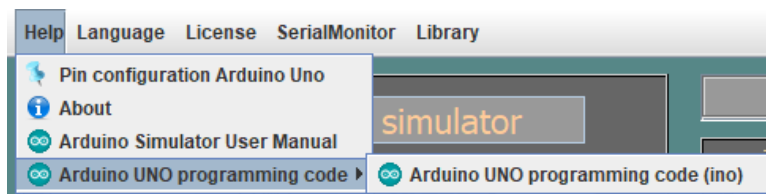
همچنین می توانید فقط CheckVars را با علامت زدن کادر «CheckVar» ببینید.



## آماده سازی برنامه ARDUINO UNO

یک اسکچ جدید باز کنید: (xx.ino)

برنامه (.ino) Simulator UNO و کتابخانه "SimulatorProgram.h" را می توانید در زیر منو Help پیدا کنید:



برنامه "Arduino UNO programming code" را راه اندازی کنید اکنون می توانید کد خود را در آردوینو وارد کنید، اگر در آردوینو آپلود شد، می توانید آن را با شبیه ساز تست کنید.

توجه : کتابخانه **SimulatorProgram.h** فقط با شبیه ساز هست.

یک برنامه جدید در برد آردوینو آپلود کنید.

- برنامه آردوینو را راه اندازی کنید
- اسکچ را باز کنید
- اتصال آردوینو UNO با کامپیوتر:

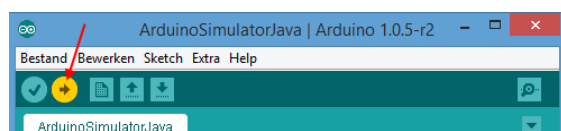


• برد «Arduino UNO» را انتخاب کنید

• پورت سریال را انتخاب کنید

• برنامه را در Arduino UNO آپلود کنید

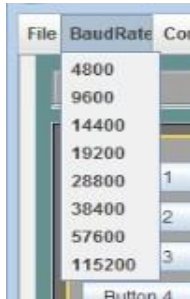
توجه BaudRate در شبیه ساز ۹۶۰۰ است.



## پورت کام را پیکربندی کنید

### تنظیم BaudRate

Baud Rate به طور پیش فرض روی ۹۶۰۰ تنظیم شده است یا نرخ Baud را در کد آردوینو و شبیه ساز تغییر دهید.



### تنظیم پورت کام

ابتدا باید پورت COM را انتخاب کنید، پورت USB که توسط آردوینو استفاده می شود. شبیه ساز به طور خودکار آردوینو را شناسایی می کند و "قرمز" می شود.

#### Before the selection



#### After the selection



### توجه:

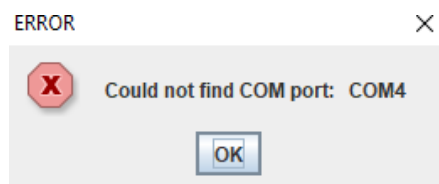
هنگام راه اندازی، وضعیت شبیه ساز را در پایین پورت COM نیز مشاهده می کنیم:

Disconnect from COM Port

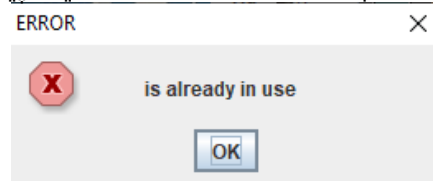
هنگامی که پورت COM صحیح را انتخاب کردید به این متن تغییر می کند:

Connect to COM PORT

اگر پورت COM اشتباهی را انتخاب کنید یا آردوینو متصل نیست، این پیام را دریافت می کنید:



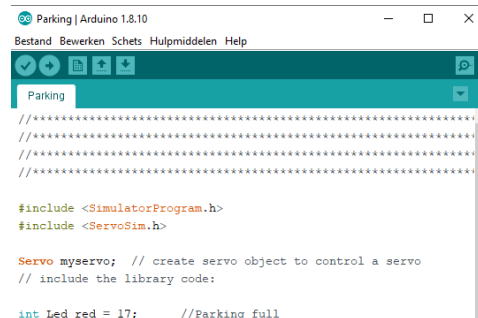
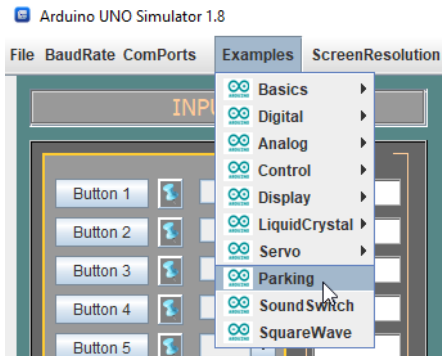
اگر یک ورودی یا خروجی را انتخاب کنید که قبلاً استفاده شده است، یک پیام خطا دریافت خواهید کرد:



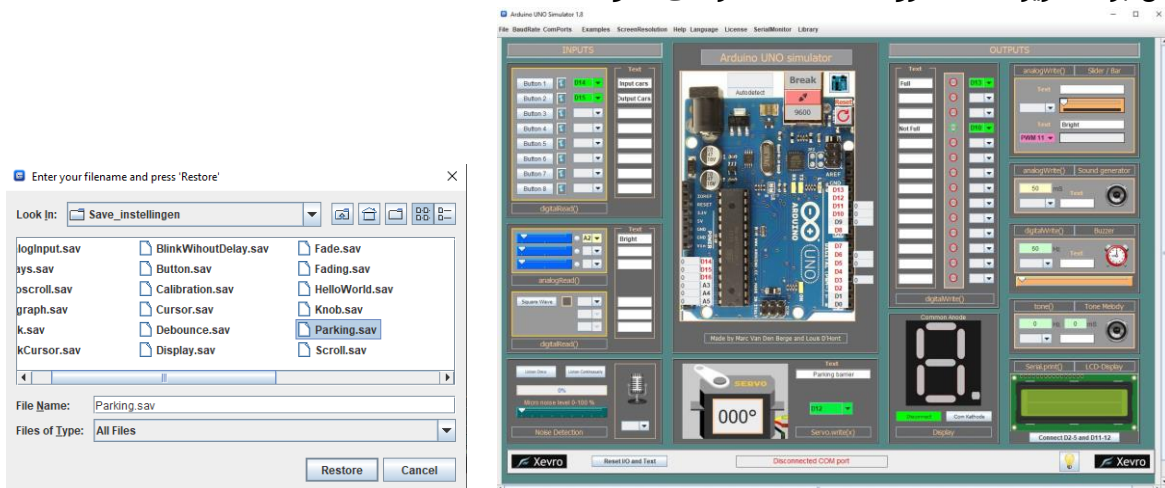
# چگونه یک طرح از پیش برنامه ریزی شده را اجرا کنیم

## مثال: Parking

"Examples" را انتخاب کنید و "Parking" را انتخاب کنید، طرح (اسکچ) آردوینو شروع به کار می کند.



به منوی "File" بروید و "Restore Settings" را انتخاب کنید، فایل "parking.sav" را انتخاب کنید و ابزارهای از پیش برنامه ریزی شده روی صفحه ظاهر می شوند.



BaudRate معمولاً ۹۶۰۰ است، این را در طرح آردوینو بررسی کنید.

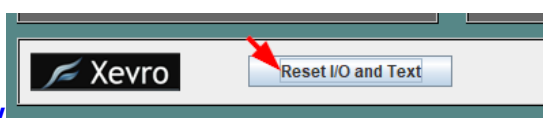
`Serial.begin(9600);`

پورت COM را انتخاب کنید، ارتباط بین آردوینو و شبیه ساز برقرار شد (دکمه سبز) و برنامه در حال اجرا است.



اگر می خواهید یک طرح جدید شروع کنید:

دکمه «Reset I/O and Text» را فشار دهید.



## نمایشگر برد آردوینو

با کلیک بر روی دکمه برد آردوینو پنجره ای باز می شود که در آن یک برد آردوینو UNO با تمام پین های IO استفاده شده روی آن مشاهده می کنید. دکمه Reset همه پین های IO را پاک می کند تا دوباره IO را انتخاب کنید LED. های RX/TX نیز به صورت بصری شبیه سازی شده اند.

پنجره برد آردوینو UNO به عنوان یک پنجره سطح بالا تنظیم شده است تا اطمینان حاصل شود که همیشه می توانید برد آردوینو را در حین شبیه سازی پروژه ها ببینید.

روش اتصال واقعی IO در تنظیمات به کتابخانه شبیه ساز اجازه می دهد تا از IO شبیه سازی شده همراه با پین های IO واقعی استفاده کند، اگر می خواهید از اجزای متصل در آن استفاده کنید. وقتی یک ریست سخت افزاری روی برد آردوینو واقعی یا ریست از برد آردوینو شبیه ساز انجام می دهید، تمام اتصالات IO واقعی را از دست خواهید داد، برای بازگرداندن این اتصالات باید دستورالعمل زیر را در تنظیمات وارد کنید.

هنگام کار با TCPL، باید دستور "RealIO\_Connect()" را بعد از تاخیر (5000) قرار دهید.

```
void setup() {
//***** code for SimulatorTCP *****
Serial.begin(9600); // Simulator Serial Con
inString.reserve(25);

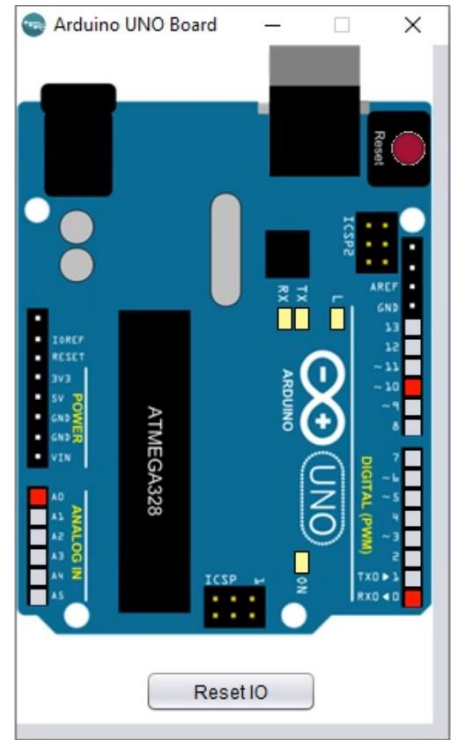
while (!Serial) {
; // wait for serial port to connect. Needed for native USB port only
}
Ethernet.begin(mac, ip, gateway, subnet);
server.begin(); // Start the server.
Serial.print("Init");
delay(5000);
RealIO_Connect(); // Used for connection between arduino and real IO . place
//lcd.port(portname); // The lcd library needs this portname
//myservo.port(portname); // The servo library needs this portname
//stepper.port(portname); // The stepper library needs this portname
//irrecv.port(portname); // The IRremote library needs this portname
//irsend.port(portname); // The IRremote library needs this portname
//***** and code SimulatorTCP *****
}

void setup() {
// Simulator Serial Connection code
Serial.begin(9600);
inString.reserve(10);
RealIO_Connect(); // Used for connection between
```

وقتی روی یک پین کلیک می کنید، می توانید وضعیت پین را تغییر دهید. برای استفاده از یک پین آنالوگ یا دیجیتال به عنوان ورودی یا خروجی واقعی، باید روی آن ها کلیک کنید و علامت D یا A در آن نشان داده می شود. یک پین "P" PWM را نشان می دهد. هر بار که روی یک پین کلیک می کنید، وضعیت به کتابخانه Simulator در برد آردوینو ارسال می شود.



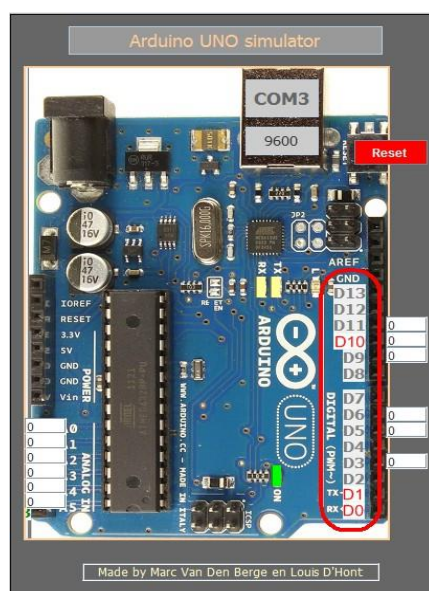
- Real analog input
- Real digital in – or output
- Real PWM output
- Select by simulator
- No selection



## از IO آنالوگ و دیجیتال استفاده کنید

### ورودی های دیجیتال

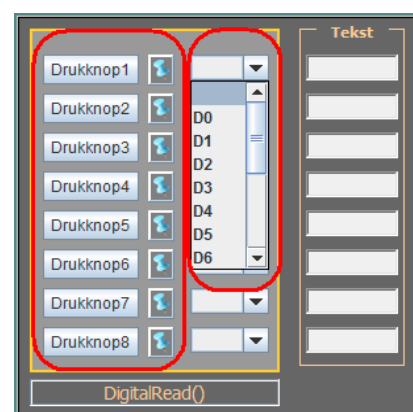
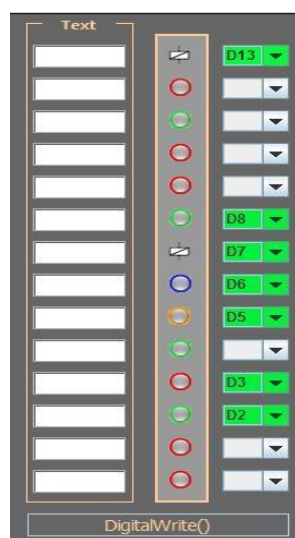
Arduino UNO دارای ۱۴ پین دیجیتال و ۶ پین آنالوگ IO است که می توانیم آنها را در ورودی یا خروجی (IO) پیکربندی کنیم. این پین ها تصاویر نمادین را از D0 تا D13 و A0 تا A5 دریافت می کنند. در صورت انتخاب، برچسب های IO به فرمز تغییر می کنند.



### دگمه ها

8 دکمه موجود است. جعبه ترکیبی برای اتصال دکمه به یکی از ۱۴ پین IO استفاده می شود.

از پین آبی روشن می توان برای نگه داشتن دکمه در حین انجام کارهای دیگر استفاده کرد، با فشار دادن آن، حاشیه به رنگ فرمز تغییر می کند. دکمه ها را می توان با تابع `digitalRead()` کنترل کرد.



### LED

۱۴ led موجود است. به ازای هر پین آردوینو یک led. برای اتصال آن به آردوینو از کمپوآکس استفاده کنید. با کلیک بر روی led می توانید رنگ را تغییر دهید.

LED ها را می توان با تابع `digitalWrite()` کنترل کرد.

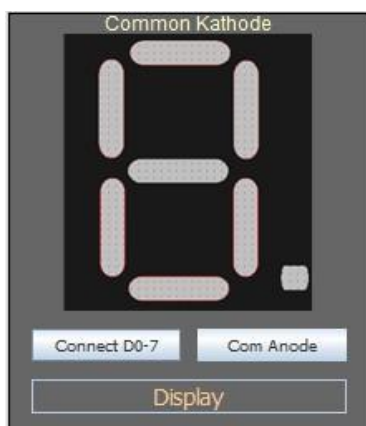
از زنگ برای ایجاد نویز با فرکانس سفارشی استفاده می شود. از کمپوباکس برای اتصال زنگ با آردوینو استفاده می شود.

زنگ را می توان با `digitalWrite()` کنترل کرد. تابع با ارسال دیجیتال- `Write (Pin, HIGH)`. در کد آردوینو، زنگ صدا با فرکانس قابل تنظیم صدا ایجاد می کند (از نوار لغزنده برای تغییر فرکانس استفاده کنید).



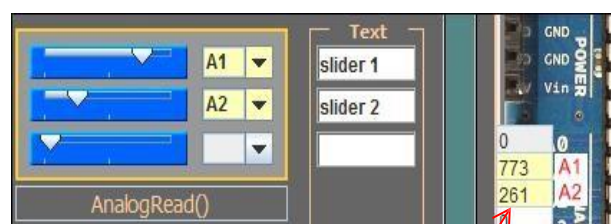
## سون سگمنت

صفحه نمایش ۷ سگمنت دارای ۶ پین دیجیتال است که می تواند به D2-8 در آردوینو متصل شود. صفحه نمایش را می توان در آند مشترک یا کاتد مشترک متصل کرد.



برای روشن کردن صفحه نمایش از `DigitalWrite` استفاده کنید. مثال را ببینید: پارکینگ.

## SLIDERS



3 لغزنده برای اتصال با یکی از ۶ پین آنالوگ (A0-A5) وجود دارد. لغزنده ها را می توان توسط آردوینو با تابع `analogRead()` خواند. در آردوینو یک کادر سفید دارید که مقدار لغزنده نشان داده شده است.

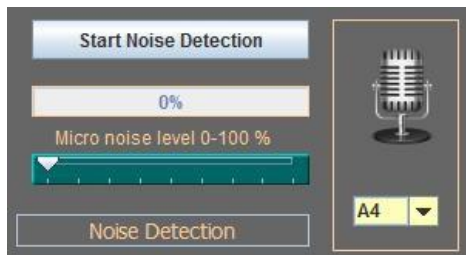


## تشخیص نویز

تشخیص نویز برای ارسال سیگنال آنالوگ (۰-۱۰۲۳) به آردوینو استفاده می شود که بستگی به سطح نویز دارد. کمبوباکس برای اتصال ردیاب نویز به یکی از ۶ پین آنالوگ (A0-A5) استفاده می شود.

هنگامی که روی 'Start Noise Detection' شروع تشخیص نویز" کلیک می کنید، تشخیص شروع به گوش دادن به سطح نویز میکروفون می کند. هنگامی که سطح نویز از مقدار لغزنده فراتر رفت، سیگنال (۰-۱۰۲۳) را به آردوینو ارسال می کند. مقدار حد در کد آردوینو باید کمتر از نوار لغزنده تشخیص نویز باشد زیرا هنگامی که نویز تشخیص داده شود سیگنال ارسال می شود.

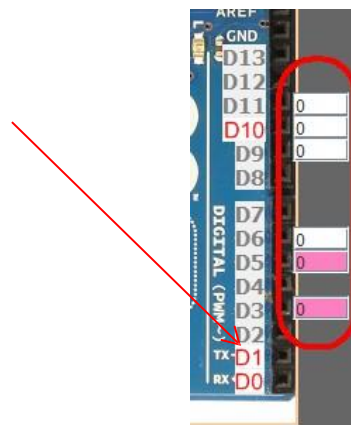
تشخیص نویز را می توان با تابع `analogRead()` کنترل کرد.



## بارگراف

بارگراف را می توان به یکی از ۶ پین دیجیتال PWM آردوینو متصل کرد. نمودار درصد مقدار شما (۰-۱۰۲۳) را نشان می دهد، این می تواند برای شبیه سازی سیگنال PWM به عنوان یک سیگنال استفاده شود.

از `analogWrite(pin, value)`; برای کنترل بارگراف استفاده کنید. (برای مثال نگاه کنید به: sound switch).



## مولد صدا

The sound generator can be connected to one of the 6 digital PWM pins of the Arduino.  
By changing the time (ms) you change the duration that the sound goes off (1ms – 10 000ms). The frequency can go from 10hz to 10Khz.  
Use analogWrite(pin, value); to control the sound generator.

مولد صدا را می توان به یکی از ۶ پین دیجیتال PWM آردوینو متصل کرد.  
با تغییر زمان (ms) مدت زمان خاموش شدن صدا را تغییر می دهید (۱ میلی ثانیه تا ۱۰۰۰۰ میلی ثانیه).  
فرکانس می تواند از ۱۰ هرتز تا ۱۰ کیلوهرتز باشد.  
برای کنترل مولد صدا از analogWrite (پین، مقدار) استفاده کنید.

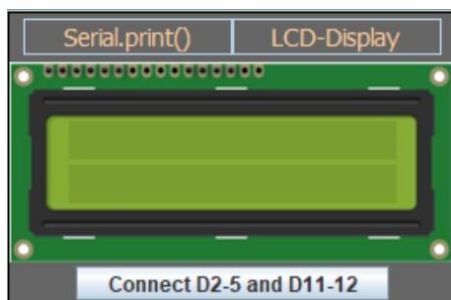


## SERVO

سرو را می توان به یک پین دیجیتال (D2-13) آردوینو متصل کرد. تعداد درجات (°) در سرو قابل مشاهده است. روی سرو کلیک کنید تا با حذف پس زمینه و جعبه ترکیبی، سرو کوچک تر شود. از servo.write() و servo.attach() استفاده کنید. کتابخانه شبیه ساز سرو را برای استفاده از آن اضافه کنید.



## نمایشگر LCD



صفحه نمایش LCD را می توان با اتصال D2-5 ، D11 و D12 به آردوینو متصل کرد. کتابخانه LiquidCrystalSim شبیه ساز را اضافه کنید تا با شبیه ساز کار کند.

## ملودی تن

ملودی تن را می توان به پین دیجیتال D8 آردوینو متصل کرد. فرکانس و زمان صدا (میلی ثانیه) در کادرهای سبز روشن وجود دارد.  
از tone(8, f, d); and noTone(8); Use استفاده کنید. (نگاه کنید به مثال: Tone Melody)

برای توقف: noTone()

f = frequency

d = duration

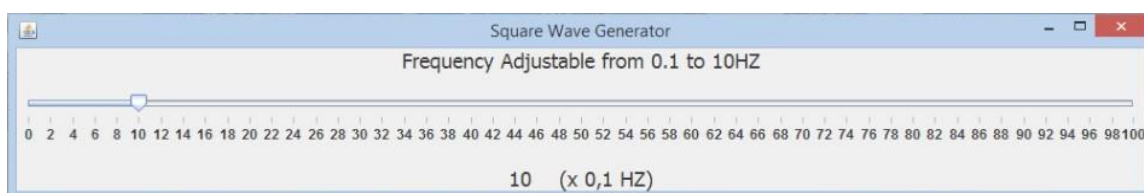
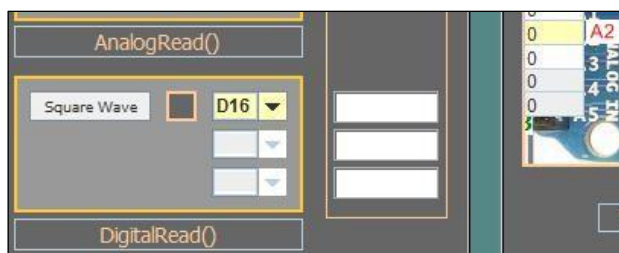


## مولد موج مربعی

موج مربعی سیگنال‌های کششی را به آردوینو می‌فرستد، هنگامی که سیگنال بالا است، مربع خاکستری به رنگ قرمز روشن می‌شود. جعبه ترکیبی برای اتصال موج مربعی به یکی از ۶ پین آنالوگ (A0-A5 = D14-19) استفاده می‌شود.

هنگامی که روی دکمه "SquareWave" کلیک می‌کنید، پنجره دوم با یک نوار لغزنده برای تغییر فرکانس باز می‌شود.

موج مربعی را می‌توان با تابع `digitalRead()` کنترل کرد.



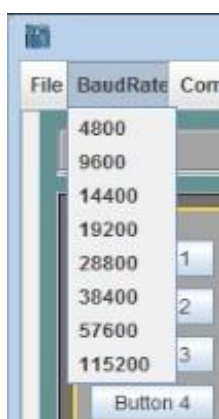
## چند نکته در حین آماده سازی برنامه ARDUINO همیشه طرح را در Arduino UNO آپلود کنید.

اگر شبیه ساز به آردوینو متصل باشد، نمی توانید برنامه آردوینو را آپلود کنید. ما یک ابزار "قطع اتصال" ایجاد کردیم که اتصال را با پورت COM شبیه ساز می بندد تا بتوانید طرح را در آردوینو آپلود کنید.

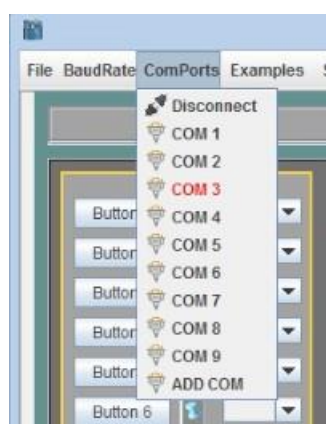
مزیت بزرگ این است که ما نیازی به خاموش کردن Simulator هر زمان که می خواهیم شبیه ساز اسکیچ را آپلود کنیم، نداریم.

پس از دانلود شبیه ساز، دوباره با پورت COM وصل شده و ورودی/خروجی را مقدار دهی اولیه می کنیم.

### Choose BaudRate



### Choose COM port



### Disconnect COM port

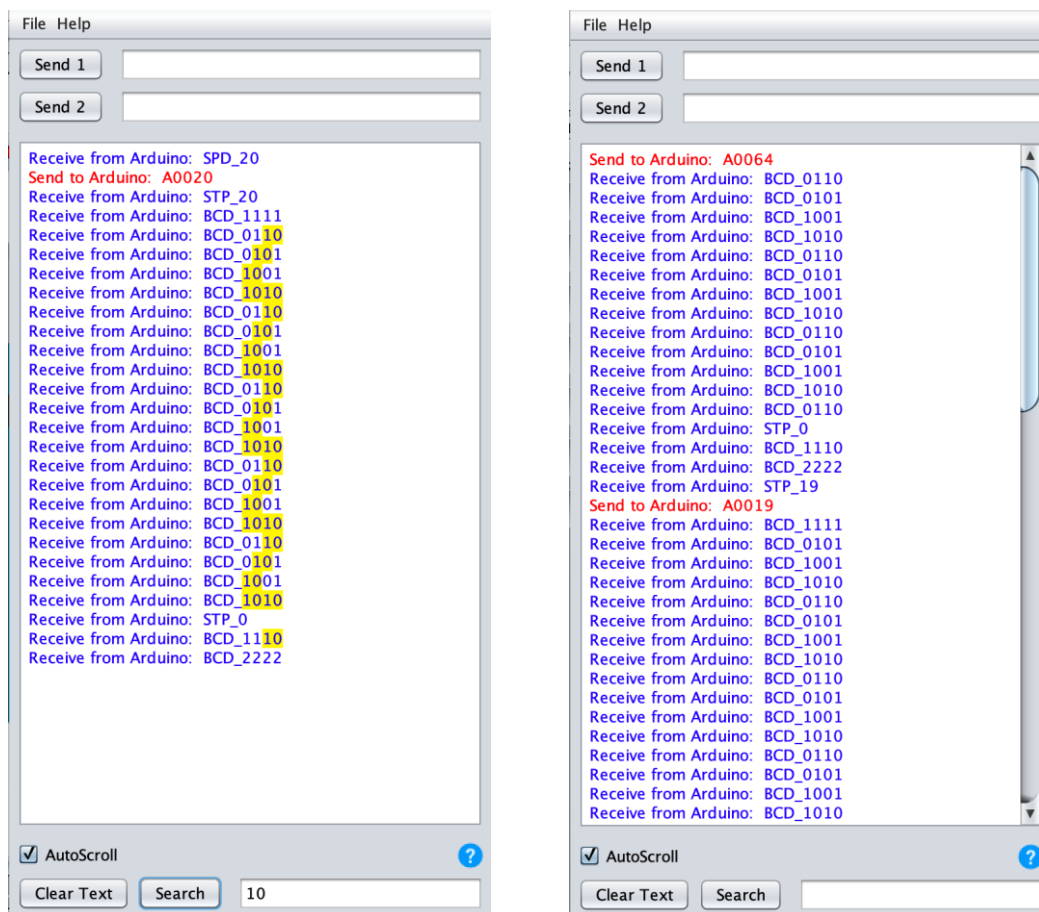


## سریال مانیتور

در شبیه ساز، اکنون امکان مشاهده داده های سریال نیز وجود دارد. ۲ دکمه وجود دارد که می توانید از آنها برای ارسال سیگنال به آردوینو استفاده کنید. وقتی آردوینو داده ها را به شبیه ساز ارسال می کند، مانیتور خط «دریافت از آردوینو» را به شما نشان می دهد و وقتی چیزی را با شبیه ساز به آردوینو ارسال می کنید، خط «ارسال به آردوینو» نشان داده می شود.

اگر می خواهید خروجی مانیتور سریال را ذخیره کنید، می توانید کل متن یا یک منطقه انتخاب شده را ذخیره کنید. با عملکرد جستجو، جستجو برای یک کلمه یا کاراکتر خاص امکان پذیر است. اگر کلمه یا کاراکتر پیدا شود، با رنگ زرد برجسته می شود. با کلیک بر روی علامت سوال آبی توضیحات تمامی کدهای شبیه ساز را دریافت می کنید.

با استفاده از Serialprint(); در Arduino IDE می توانید یک پیام سریال ارسال کنید. پیام سریال با " \_txt" در مانیتور سریال شروع می شود.

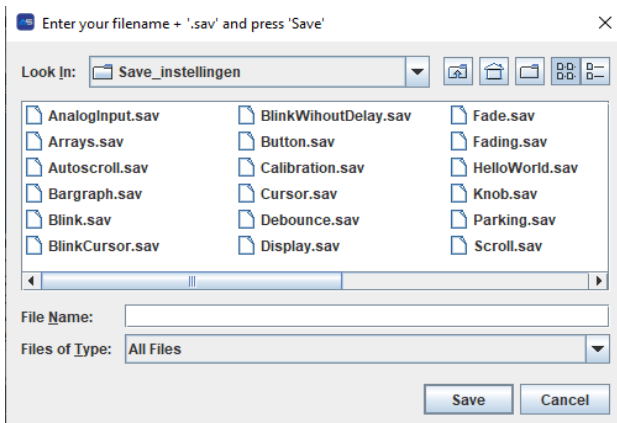


## ذخیره و بازیابی تنظیمات

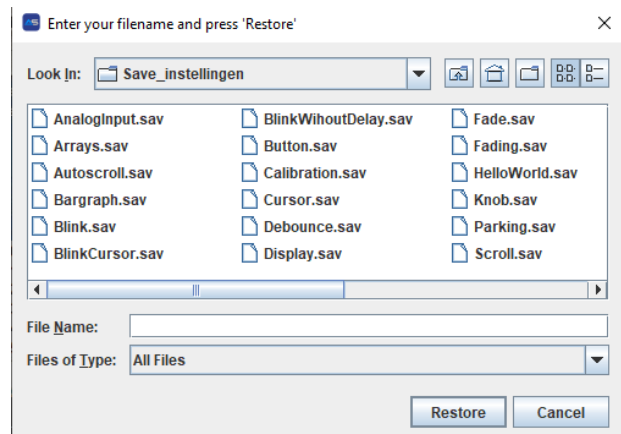
با "ذخیره" می‌توانید ورودی/خروجی انتخابی و متون دیکته شده خود را ذخیره کنید. دکمه "بازیابی" تنظیمات را بازیابی می‌کند تا استفاده از آن آسان شود. می‌توانیم نام فایل پسوند را با \*.sav یا \*.txt ذخیره کنیم.

تابع «ذخیره و بازیابی» را در زیر «فایل» پیدا می‌کنید.

### Save



### Restore



## زبان

5 زبان موجود است، می‌توانید زبان را در نوار ابزار منو تغییر دهید. ما از انگلیسی، آلمانی، فرانسوی، اسپانیایی و هلندی پشتیبانی می‌کنیم. نرم‌افزار همیشه با انگلیسی شروع می‌شود.



## وضوح صفحه نمایش

ما ۴ گزینه داریم:

-وضوح: 1024 x 768

-وضوح: 1336 x 768

-رزولوشن: 1920 x 1080

-وضوح متغیر (حداقل: ۵۰۰، حداکثر: ۲۰۰۰)

