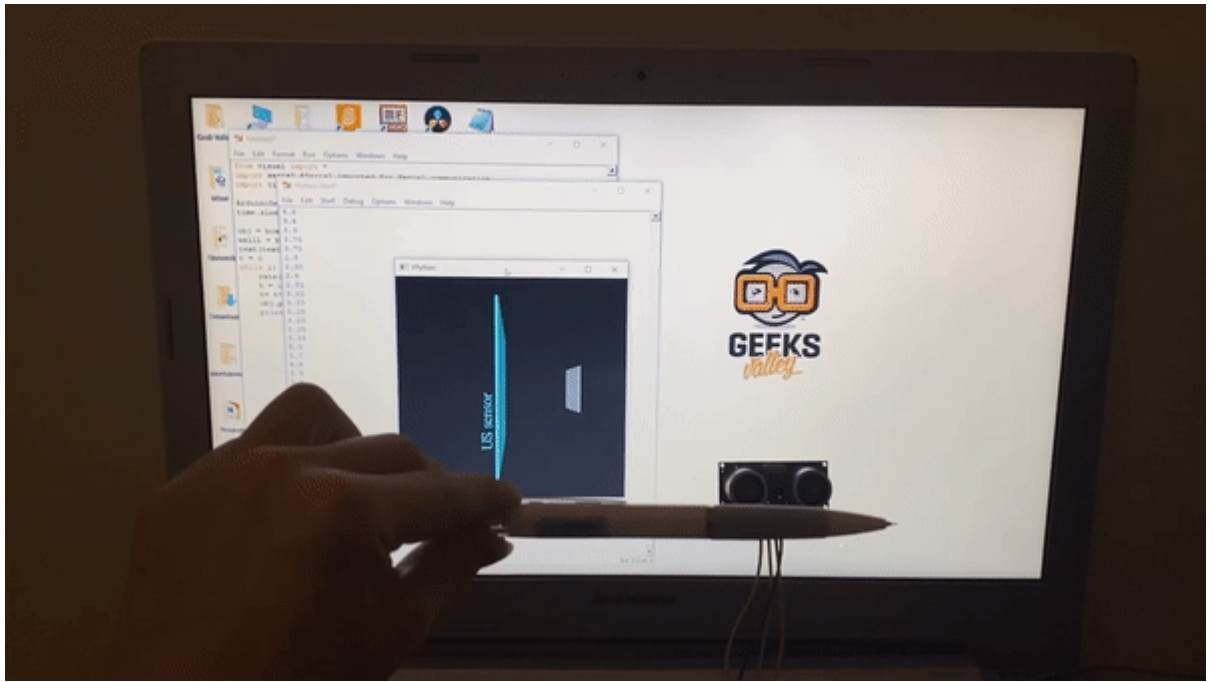


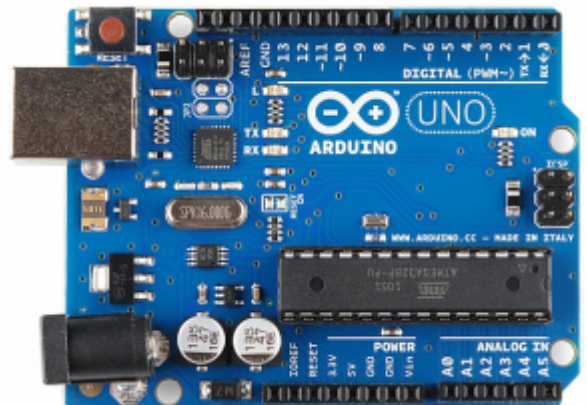
# تمثيل المسافة باستخدام الاردوينو وواجهة المستخدم الرسومية

## مقدمة

تستخدم واجهة المستخدم الرسومية (GUI) في مجالات كثيرة منها الطب لتمثيل تخطيط قلب المريض أو في الرياضيات لتمثيل الإحصائيات والبيانات وغيرها من المجالات في هذا الدرس سنتعلم كيفية تمثيل المسافة باستخدام الاردوينو وواجهة المستخدم الرسومية.



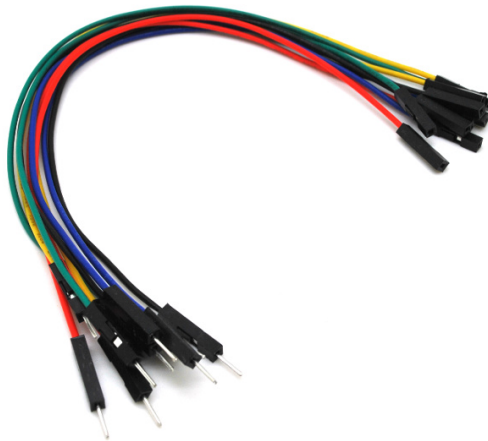
## المواد والأدوات



×1 اردوينو اونو



×1 سلك الاردوينو



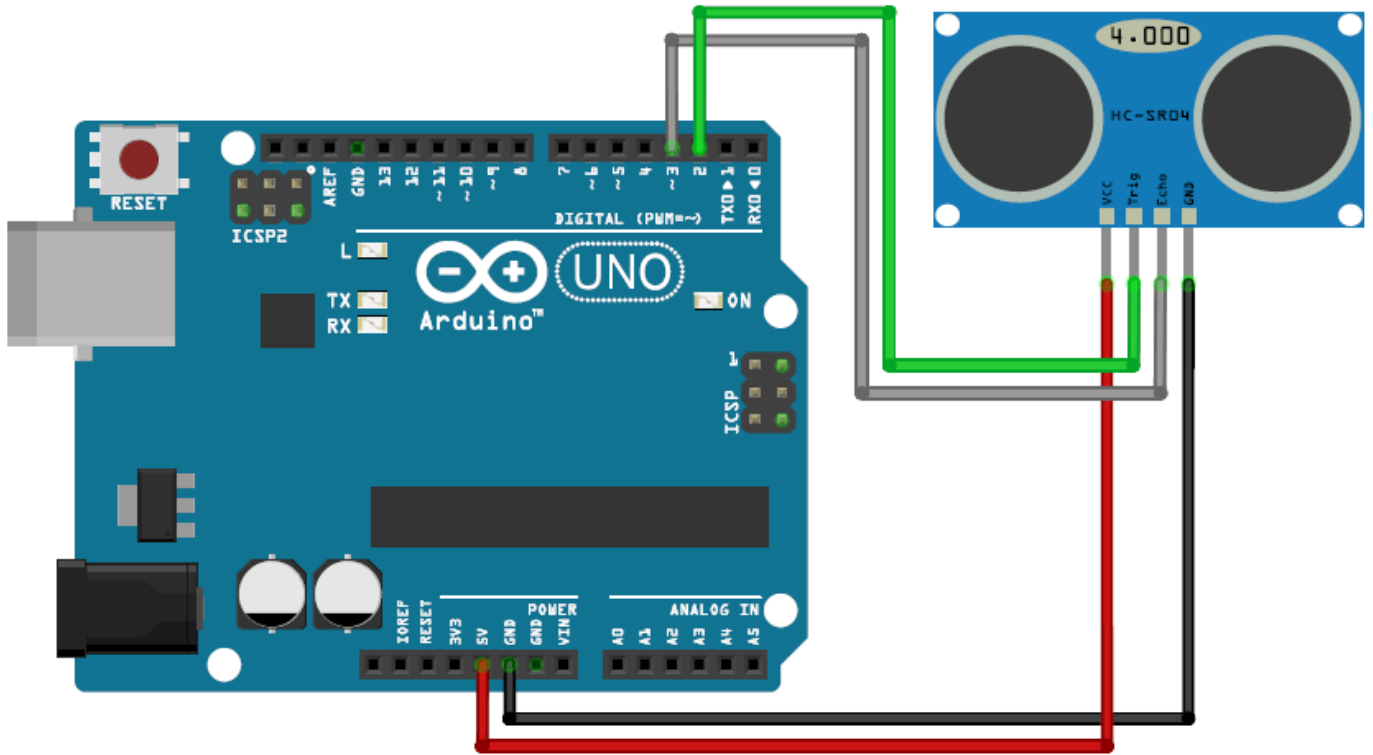
حزمة أسلاك توصيل (ذكر - أنثى)



×1 حساس المسافة (HC-SR04)

## توصيل الدائرة

للمزيد حول حساس الموجات فوق الصوتية يمكنك الرجوع للدرس التالي حساب المسافة باستخدام حساس الموجات فوق صوتية



## الكود البرمجي للتحقق من قراءة المدخلات

في البداية سنقوم برفع كود برمجي لمحاكاة مشروع تمثيل المسافة باستخدام الاردوينو وواجهة المستخدم الرسومية والتحقق من أن الحساس يعطي قراءات صحيحة، ويتم طباعة الأوامر على شاشة الاتصال التسلسلي.

ارفع الكود التالي إلى لوحة الاردوينو عن طريق (Arduino DE) وافتح شاشة الاتصال التسلسلي.

```
#define Trigger 2
#define Echo 3
int timetaken, dist;
int sendv;

void setup() {
  Serial.begin (9600);
  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);
}
void loop() {
  timetaken=dist=0; //initialize the variable to zero before calculation

  //request the US to send a wave
  digitalWrite(Trigger, HIGH);
  digitalWrite(Trigger, LOW);

  timetaken = pulseIn(Echo, HIGH); //calculate the time taken for the wave to return
  dist = (timetaken/2) / 2.91; //formulae to calculate the distance using time taken
  if (dist <= 200 && dist > 0)//send the value to python only if it ranhes from 0-20
  cm
  sendv = dist;
```

```
Serial.println(sendv);
delay(200);

}
```

## شرح الكود البرمجي

هذه الأسطر توضح منافذ الـ Arduino التي ستستخدمها لربط منافذ حساس الموجات فوق الصوتية Trigger مع المنفذ الرقمي 2 و Echo مع المنفذ الرقمي 3.

```
#define Trigger 2
#define Echo 3
```

هنا عرفنا المتغير `timetaken` ويقوم بحساب الوقت اللازم لرجوع الموجه لحساس الموجات فوق البنفسجية والمتغير `dist` يقوم بحساب المسافة وآخر متغير هو `sendv` يقوم بإرسال القيمة للبايثون.

```
int timetaken, dist;
int sendv;
```

في الدالة `setup()` ستتم قراءة البيانات بواسطة حساس الموجات فوق الصوتية منفذ Trigger للمدخلات ومنفذ Echo للمخرجات وستطبع القراءة على شاشة الاتصال التسلسلي.

```
void setup()
{
  Serial.begin (9600);
  pinMode(Trigger, OUTPUT);
  pinMode(Echo, INPUT);
}
```

في الدالة `loop()` ستكون القيمة الافتراضية صفر لمتغير المسافة ومتغير الوقت اللازم. (ما لم يقوم المستخدم بتنفيذ إجراء على حساس الموجات فوق الصوتية)

```
void loop() {
  timetaken=dist=0; //initialize the variable to zero before calculation
```

بعد ذلك سيتم تحديث البيانات المقروءة من حساس الموجات فوق الصوتية عن طريق المتغير `Trigger`.

```
digitalWrite(Trigger, HIGH);
digitalWrite(Trigger, LOW);
```

المتغير `timetaken` سيقوم بحساب الوقت اللازم لرجوع الموجه المأخوذة من الحساس عن طريق المتغير `Echo`.

```
timetaken = pulseIn(Echo, HIGH); //calculate the time taken for the wave to return
```

المتغير `dist` سيقوم بتحديث المسافة المقروءة من حساس الموجات فوق الصوتية بالاعتماد على متغير `timetaken`.

```
dist = (timetaken/2) / 2.91; //formulae to calculate the distance using time taken
```

إذا كانت المسافة المقروءة أصغر من أو تساوي 200 وأكبر من 0 سيتم إرسال القيمة للبايثون. (أي بين 0-20 سم)

```
if (dist <= 200 && dist > 0)//send the value to python only if it ranhes from 0-20  
cm
```

أخيراً ستتم طباعة المسافة على شاشة الاتصال التسلسلي.

```
Serial.println(sendv);
```

## تنصيب (python) و (Vpython)

1- تنصيب (Python IDLE)

في البداية عليك تنصيب برنامج البايثون بنسخته القديمة عليك تحميل الإصدار القديم؛ لأن برنامج Vpython لا يدعم الإصدارات الجديدة من البايثون.

انقر على الرابط التالي Python 2.7.0 وثبت البرنامج على جهاز الحاسوب.

حمل نسخة 32 بت دائماً مهما كان نظام التشغيل.

### Download

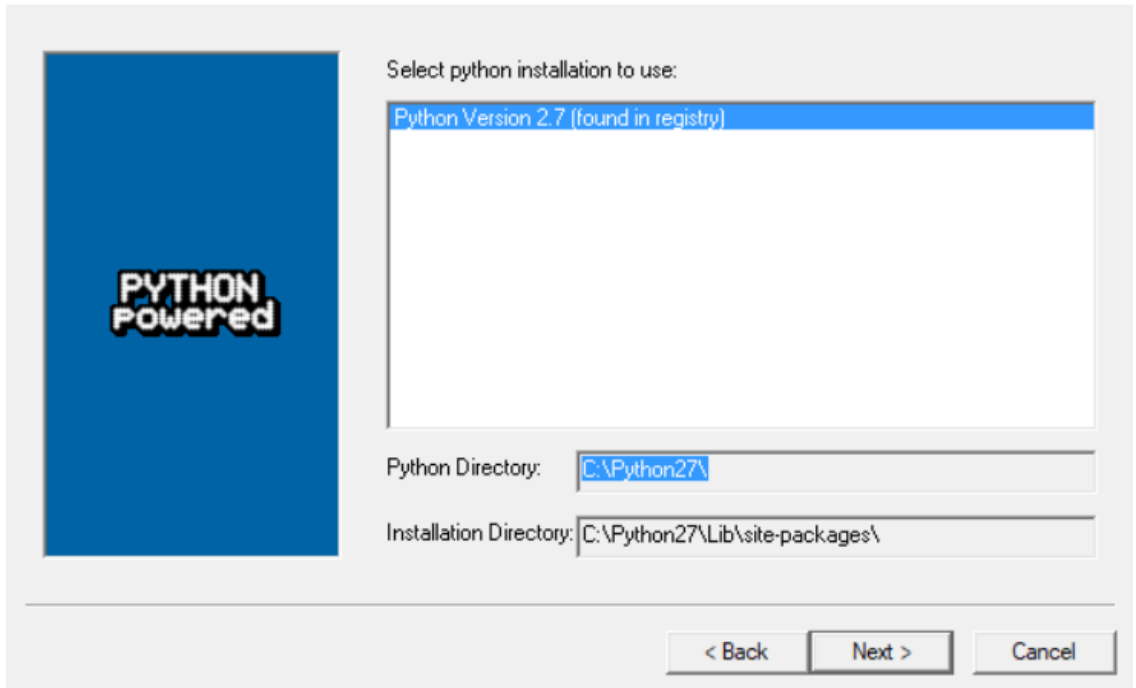
This is a production release. Please [report any bugs](#) you encounter.

We currently support these formats for download:

- [Gzipped source tar ball \(2.7.0\) \(sig\)](#)
- [Bzipped source tar ball \(2.7.0\) \(sig\)](#)
- [Windows x86 MSI Installer \(2.7.0\) \(sig\)](#)
- [Windows X86-64 MSI Installer \(2.7.0\) \[1\] \(sig\)](#)
- [Mac Installer disk image \(2.7.0\) for OS X 10.5 and later \(sig\)](#). It contains code for PPC, i386, and x86-64.
- [32-bit Mac Installer disk image \(2.7.0\) for OS X 10.3 and later \(sig\)](#).
- [Windows help file \(sig\)](#)

افتح ملف exe الذي تم تنزيله واتبع التعليمات لا تقم بتغيير المسار الذي يتم فيه تثبيت Python.

سيكون المسار C:\Python27 بشكل افتراضي اتركه كما هو عليه.



2- ضبط إعدادات لوحة الأوامر

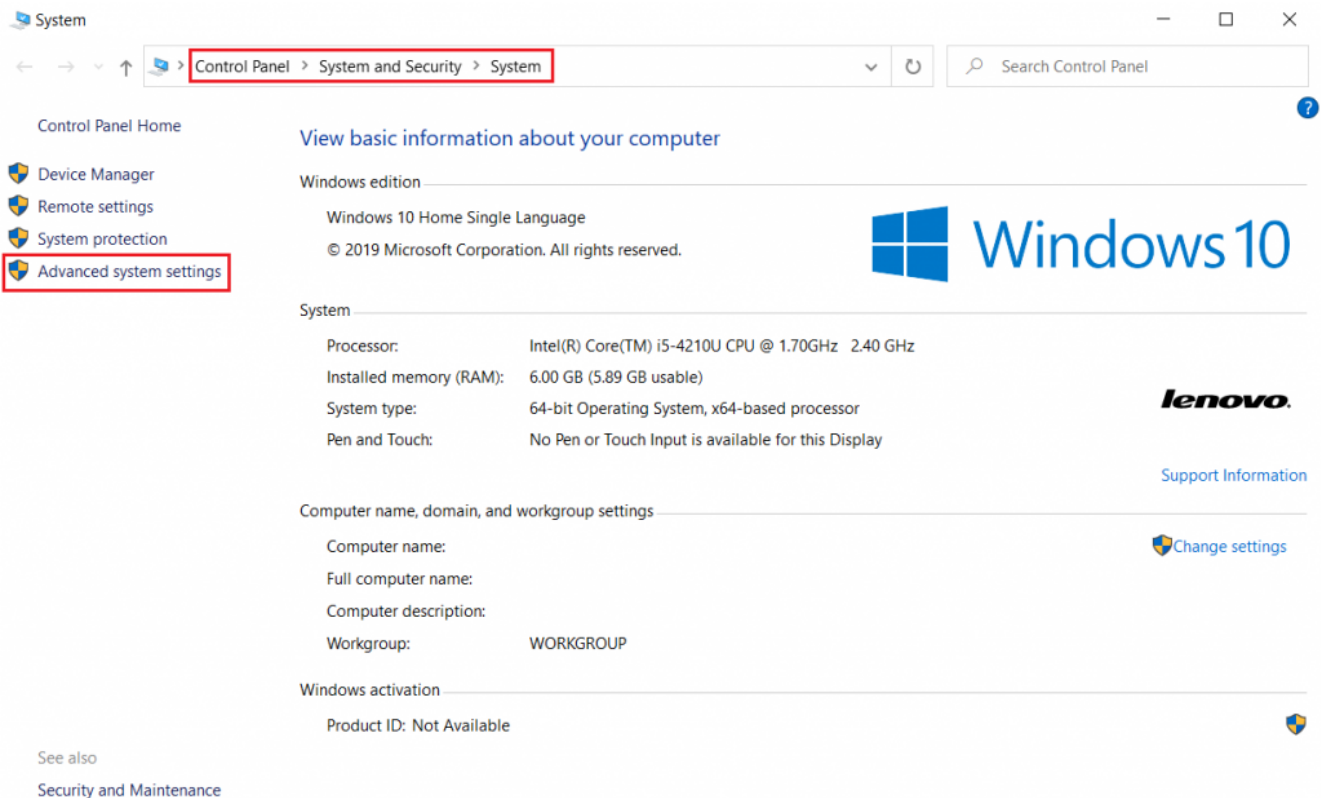
بعد تنصيب البايثون على جهازك، علينا اجراء التعديلات ليتم السماح بتنفيذ أوامره في شاشة الأوامر Command Prompt.

علينا اضافة مترجم اوامر البايثون إلى امتدادات الأوامر - PATH environment variable.

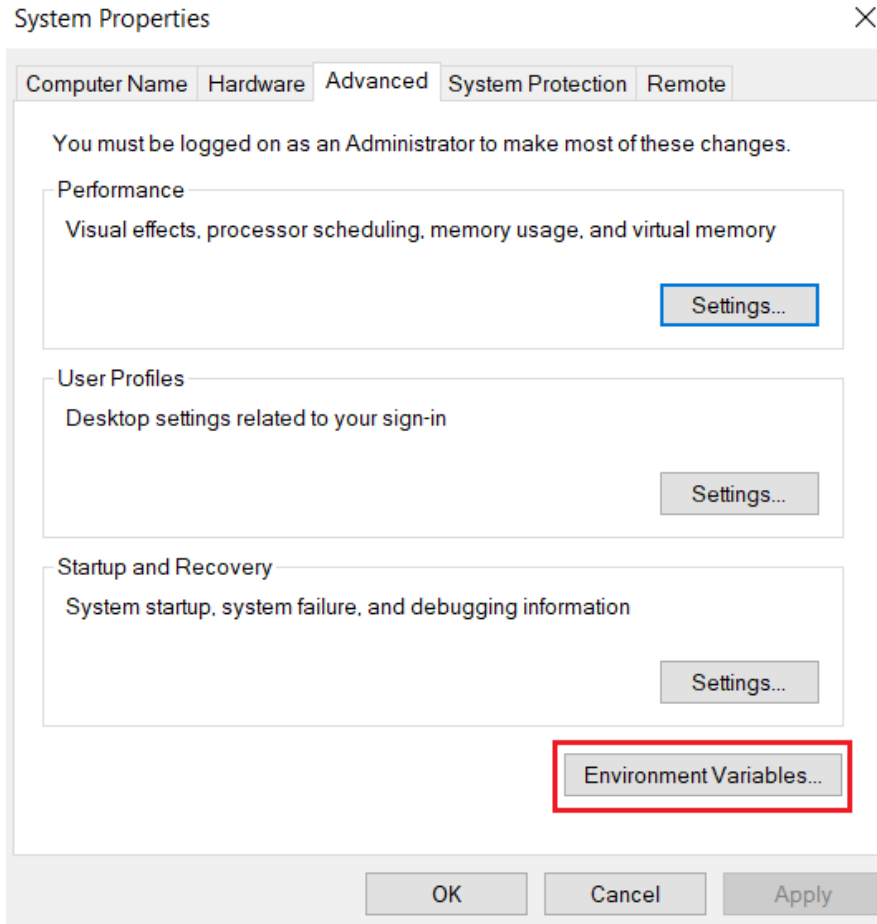
عليك الذهاب للوحة التحكم بويندوز واتباع المسار التالي:

Control Panel\System and Security\System

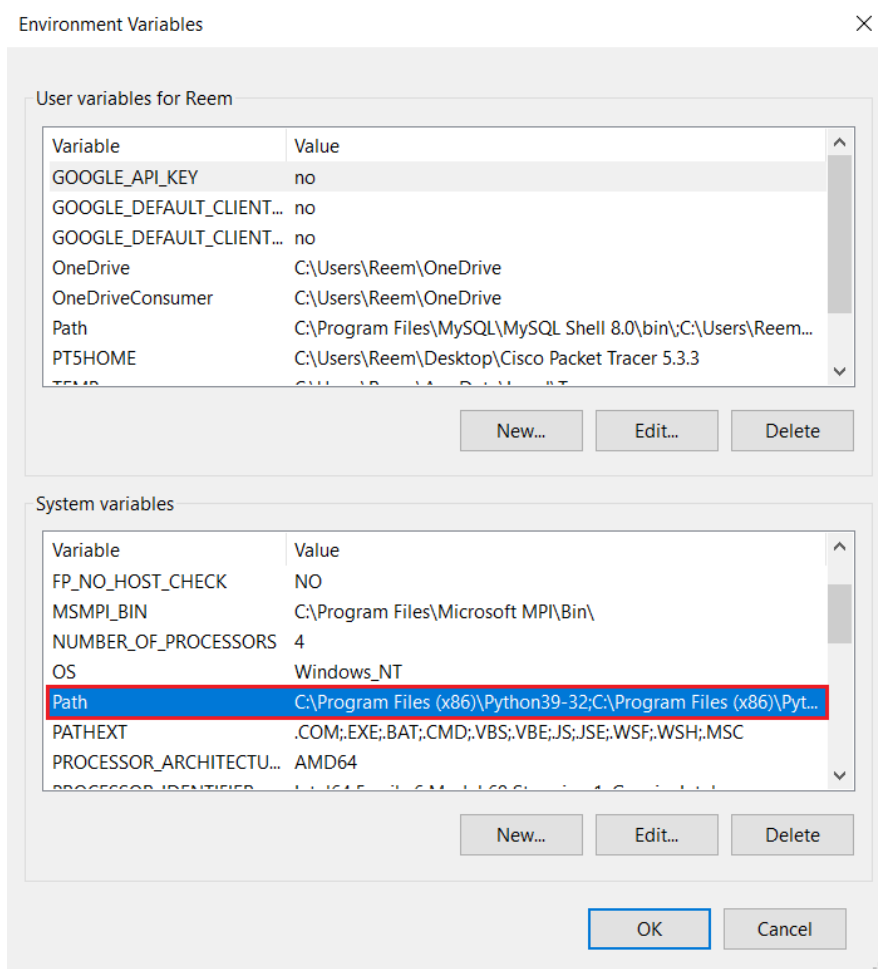
:Advanced system setting القائمة



انقر على Environment variable:



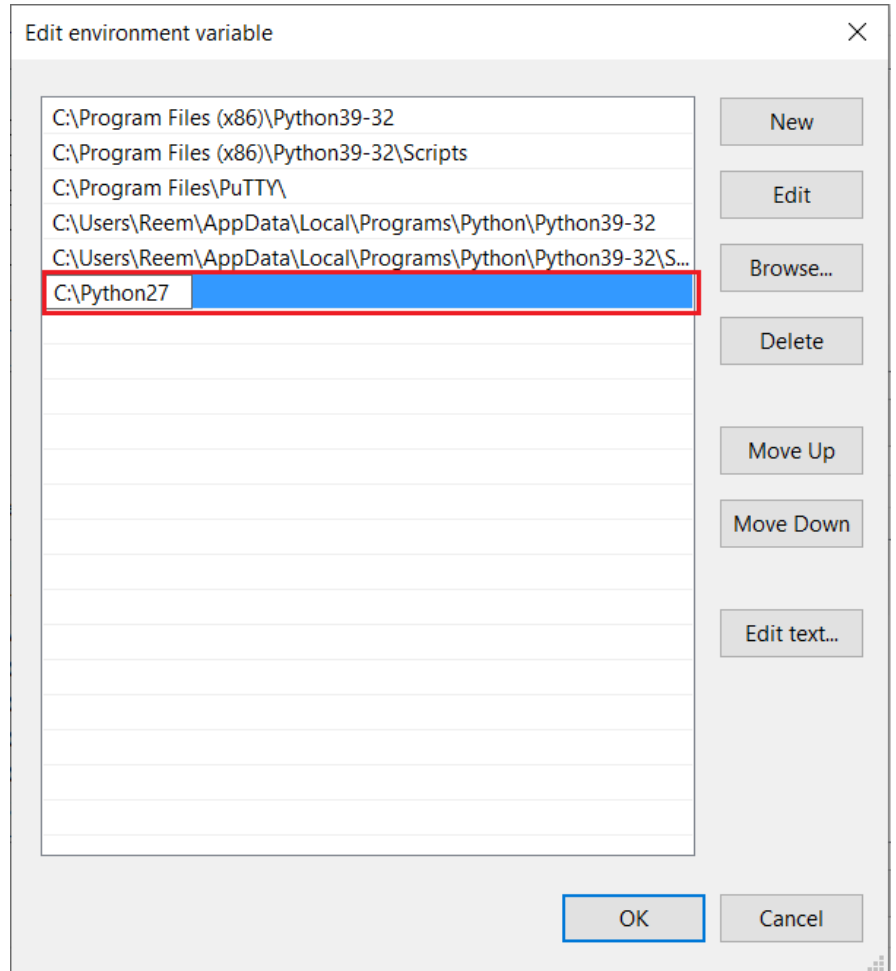
انقر على Path من قائمة System Variables.



انقر على New وأضف المسار التالي (أو المسار الذي قمت بتحميل البرنامج عليه):

C:\Python27





بعد اتمامك للخطوات سيكتمل تنصيب البرنامج، يمكنك التحقق من وجوده وذلك بالبحث عنه مع قائمة البرامج Python IDLE.  
 3- تنصيب (Pyserial)

بعد ذلك علينا تنصيب Pyserial بنسخته القديمة انقر على الرابط التالي وحمله على حاسوبك Pyserial 2.7

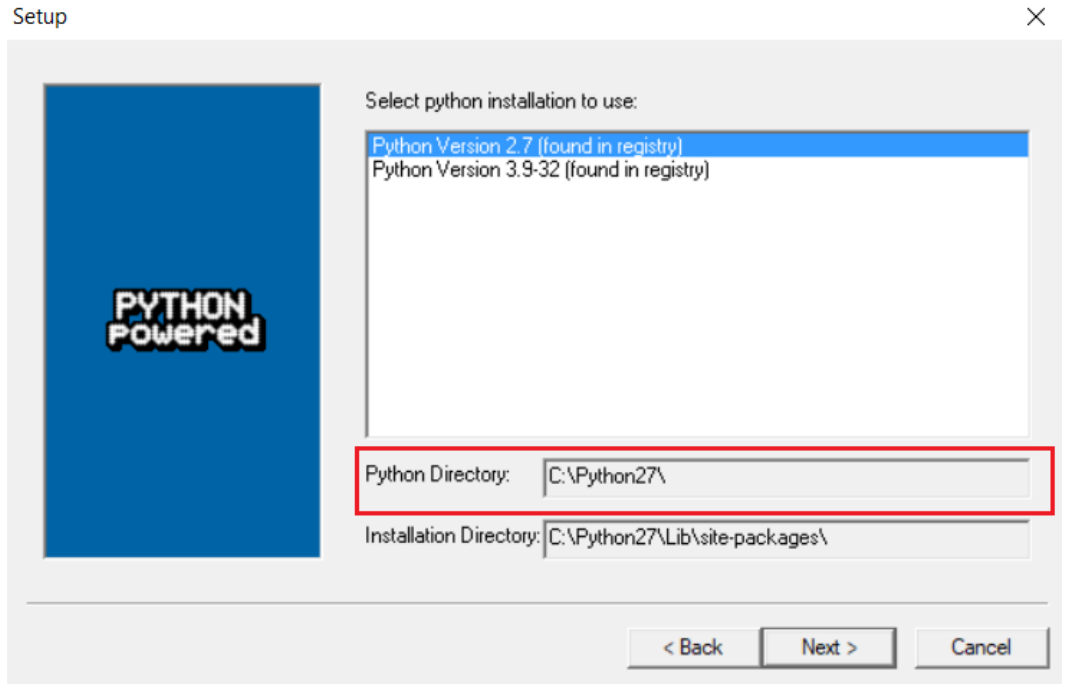
## Download files

Download the file for your platform. If you're not sure which to choose, learn more about [installing packages](#).

Filename, size	File type	Python version	Upload date	Hashes
<a href="#">pyserial-2.7.tar.gz</a> (122.1 kB)	Source	None	Oct 17, 2013	<a href="#">View</a>
<a href="#">pyserial-2.7.win32.exe</a> (270.2 kB)	Windows Installer	any	Oct 17, 2013	<a href="#">View</a>
<a href="#">pyserial-2.7.win32_py3k.exe</a> (265.4 kB)	Windows Installer	any	Oct 17, 2013	<a href="#">View</a>

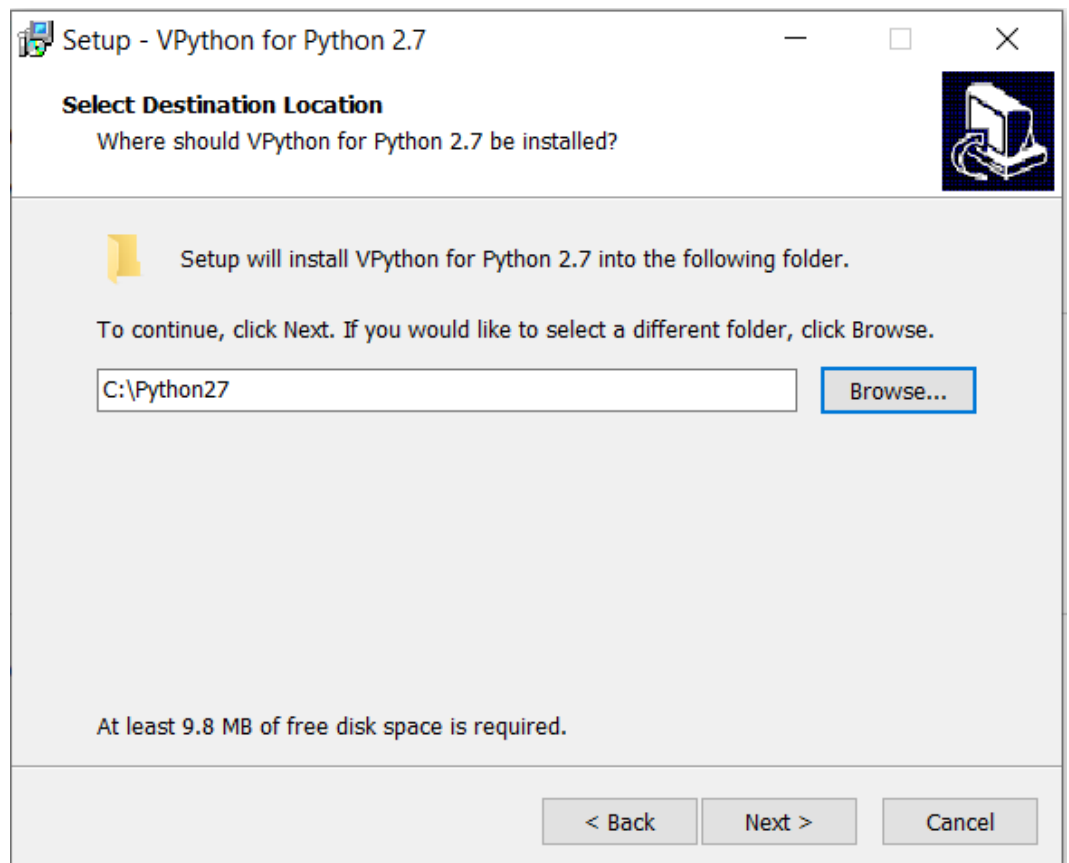
بعد تحميله اضغط على أيقونة البرنامج؛ لاستكمال اجراءات التثبيت.

اختر المسار الذي قمت بتحميل برنامج البايثون عليه بعد ذلك اضغط على next.

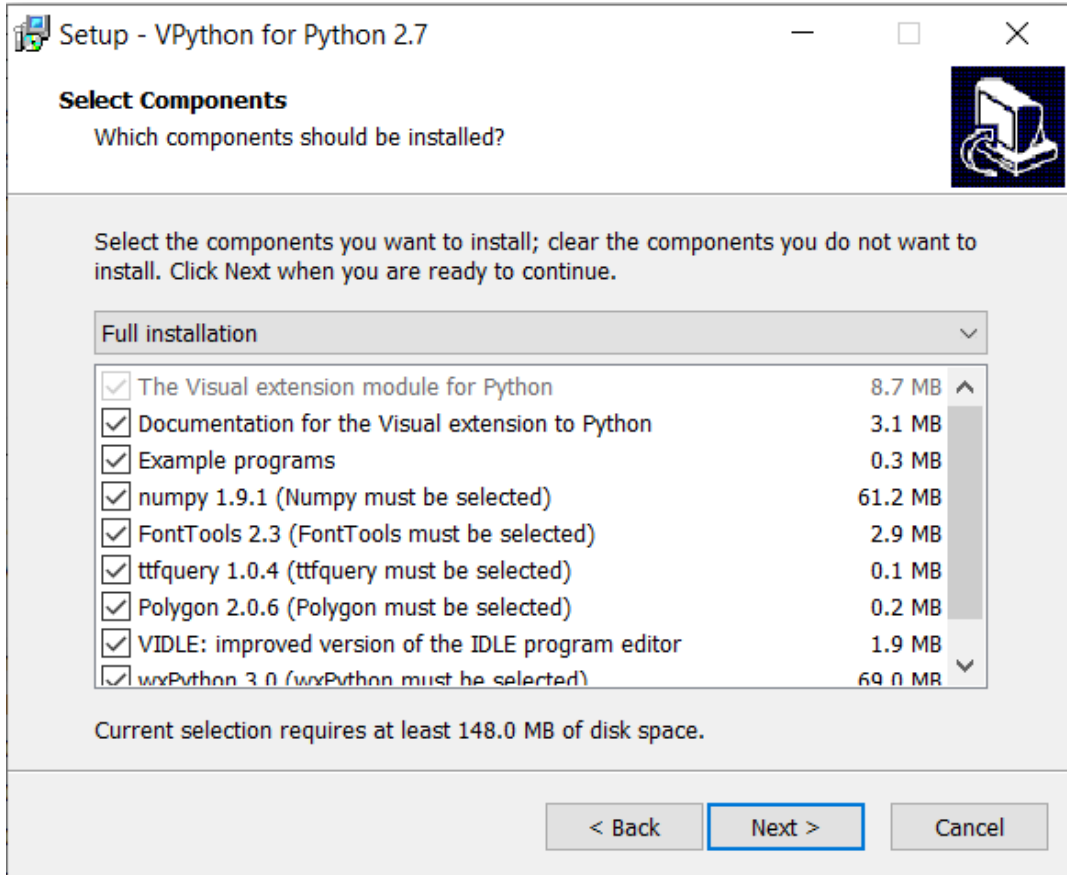


4- تنصيب (Vpython)

بعد ذلك حمّل VPython إصدار 32 بت بنفس مسار برنامج البايثون.



تأكد من تحميل الحزمة الكاملة Full installation.



بعد اكتمال التحميل يمكنك أن تجد نسخة من البرنامج على سطح المكتب أو ابحث عنه في مربع البحث.

افتح VIDLE for VPython.

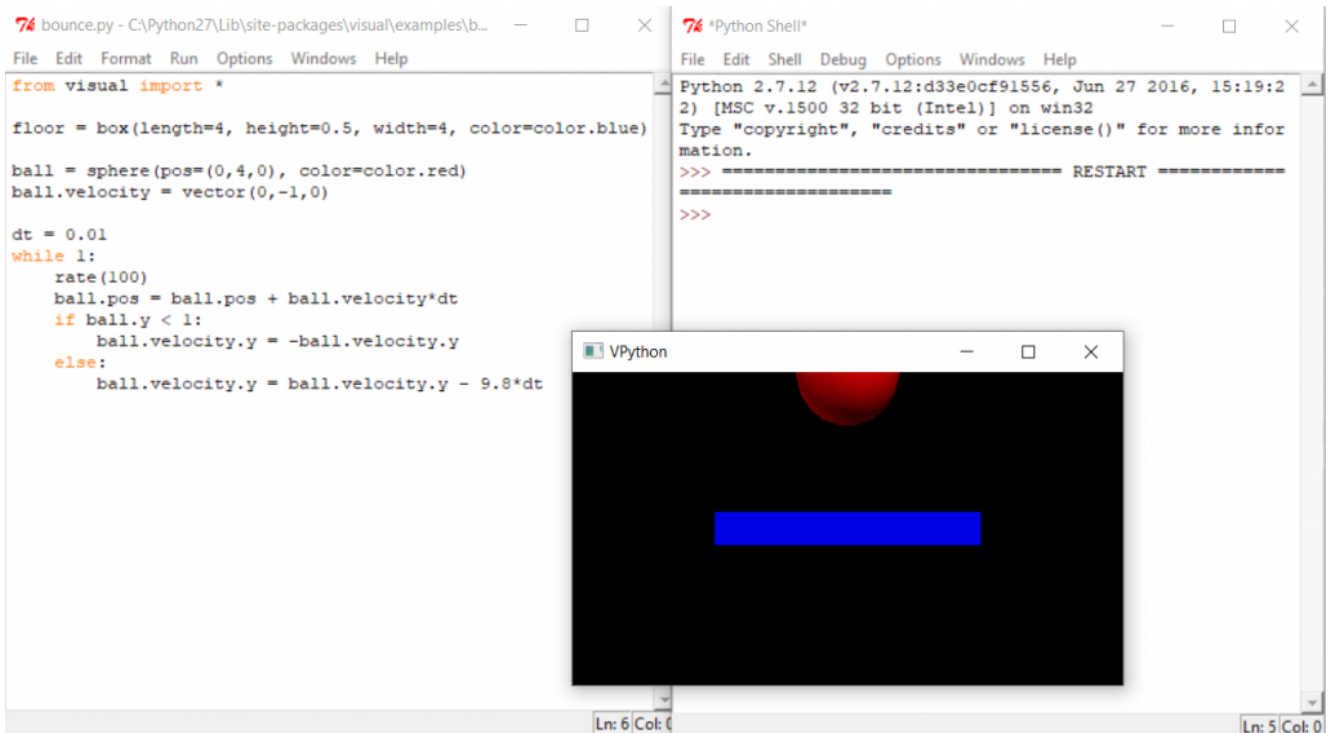
اتبع المسار التالي حتى تتمكن من اختبار صحة تحميلك للبرنامج:

File ->Open ->Bounce

ثم

Run -> Run Module

إذا ظهر لك التمثيل الثلاثي الأبعاد التالي فتنصيبك للبرنامج صحيح يمكنك الذهاب لخطوة الكود البرمجي.



## الكود البرمجي

افتح برنامج VIDLE for VPython والصق الكود البرمجي التالي وارفعه على لوحة الاردوينو عن طريق النقر على:

Run -> Run Module

```

from visual import *
import serial #Serial imported for Serial communication
import time #Required to use delay functions

ArduinoSerial = serial.Serial('com18',9600) #Create Serial port object called
arduinoSerialData
time.sleep(2) #wait for 2 seconds for the communication to get established

obj = box(pos=(-5,0,0), size=(0.1,4,4), color=color.white)
wallL = box(pos=(-1,0,0), size=(0.2,12,12), color=color.cyan)
text(text='US sensor', axis=(0,1,0) , pos=(-2,-6,0), depth=-0.3, color=color.cyan)
t = 0
while 1:
rate(100)
t = int (ArduinoSerial.readline()) #read the serial data and print it as line
t= t* 0.05
obj.pos.x = t
print(t)

```

## شرح الكود البرمجي

افتح صفحة جديدة في VIDLE for VPython وقم باستدعاء المكتبات المطلوبة visual تساعدك في انشاء نماذج ثلاثية الأبعاد, serial python لقرءة القيم من لوحة الاردوينو عن طريق المنفذ التسلسلي و time تساعدك في استخدام دوال معينة مثل delay.

```
from visual import *
import serial #Serial imported for Serial communication
import time #Required to use delay functions
```

حدد المنفذ COM المستخدم في الاتصال مع الاردوينو.

(هذا سطر قابل للتغيير بناء على نوع المنفذ الذي ستستخدمه في المشروع يمكنك تعيينه عن طريق برنامج اردوينو IDE من قائمة Port).

```
ArduinoSerial = serial.Serial('com18',9600) #Create Serial port object called
arduinoSerialData
```

بعد ذلك ستتم قراءة المسافة من حساس الموجات فوق البنفسجية وسيتم تمثيل المسافة على الواجهة الرسومية بنماذج ثلاثية الأبعاد على شكل مستطيلين بالألوان والأبعاد المذكورة بالكود البرمجي.

```
obj = box(pos=(-5,0,0), size=(0.1,4,4), color=color.white)
wallL = box(pos=(-1,0,0), size=(0.2,12,12), color=color.cyan)
text(text='US sensor', axis=(0,1,0) , pos=(-2,-6,0), depth=-0.3, color=color.cyan)
t = 0
while 1:
rate(100)
t = int (ArduinoSerial.readline()) #read the serial data and print it as line
t= t* 0.05
obj.pos.x = t
print(t)
```

بعد رفع الكود البرمجي سيظهر تمثيل المسافة على شكل مستطيل ثلاثي الأبعاد سماوي ثابت ومستطيل ثلاثي الأبعاد متحرك رمادي.

سيتحرك المستطيل الرمادي بناء على المسافة التي يقرأها من حساس الموجات فوق البنفسجية.

