



Τμήμα Μηχανολογίας,  
Επιστήμη Ξύλου & Σχεδιασμού



MSc - ΠΡΟΗΓΜΕΝΕΣ ΜΕΘΟΔΟΙ  
ΣΧΕΔΙΑΣΜΟΥ, ΤΕΧΝΟΛΟΓΙΑΣ &  
MANAGEMENT ΠΡΟΪΟΝΤΩΝ  
ΑΠΟ ΞΥΛΟ

# ΠΡΟΣΟΜΟΙΩΣΗ ΕΦΑΡΜΟΓΩΝ ΑΙΣΘΗΤΗΡΩΝ

---

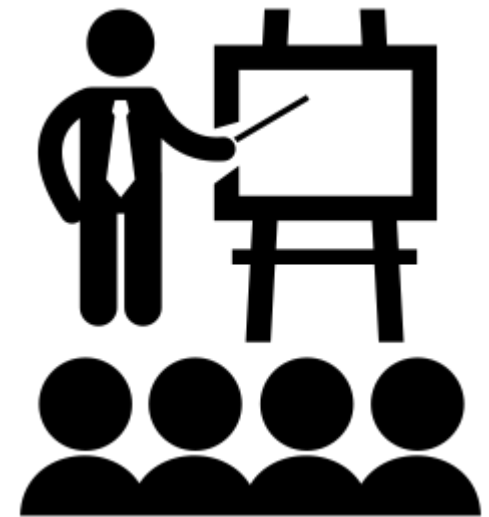
Αντώνιος Καραγεώργος

8<sup>η</sup> διάλεξη



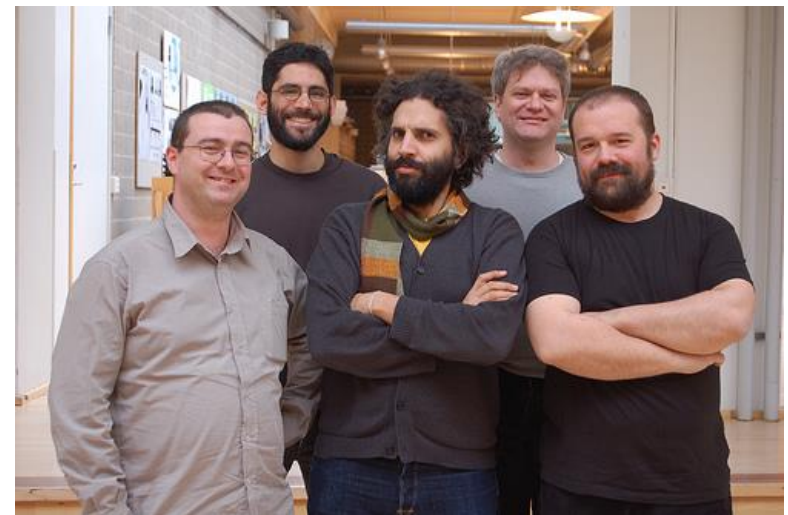
# Ενότητες Παρουσίασης

- Εισαγωγή στο Arduino Uno
- Βασικές έννοιες ηλεκτρονικής και ηλεκτρονικών κυκλωμάτων
- Προγραμματισμός Arduino Uno
- Παρουσίαση του περιβάλλοντος προσομοίωσης εφαρμογών αισθητήρων 123D Circuits με χρήση του Arduino Uno
- Παράδειγμα εφαρμογής: προγραμματισμός Arduino Uno στο 123D Circuits όπου θα αναβοσβήνει ένα λαμπάκι led.



# Εισαγωγή στο Arduino

- Δημιουργήθηκε στην Ivrea, Italy
- Παρουσιάστηκε το 2005 από τους Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, και David Mellis
- Open Source Hardware
- Open Source Development Environment





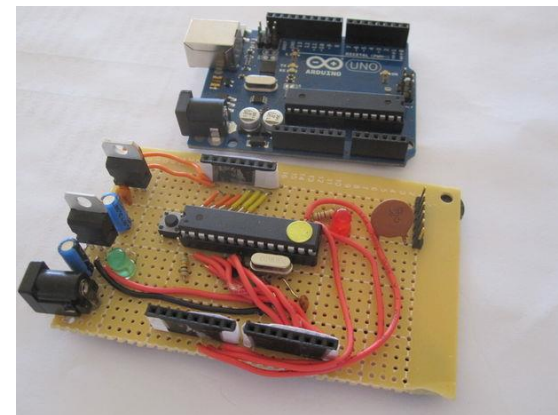
# Τι είναι το Arduino;

- Είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα και σχεδιασμού που βασίζεται σε
  - έναν απλό μικροελεγκτή (micro-controller)
  - ένα περιβάλλον ανάπτυξης (development environment, IDE) για τη συγγραφή λογισμικού.
- Μπορεί να χρησιμοποιηθεί για την ανάπτυξη διαδραστικών αντικειμένων.
- Είσοδος: ποικιλία διακοπών και sensors.
- Έξοδος: λαμπτήρες, μοτέρ, και άλλες φυσικές εξόδους.
- Ένα Arduino project μπορεί να είναι stand-alone, ή να επικοινωνεί με λογισμικό που τρέχει σε υπολογιστή.
- Οι πλακέτες μπορούν να αγοραστούν ή να φτιαχτούν.



# Γιατί Arduino;

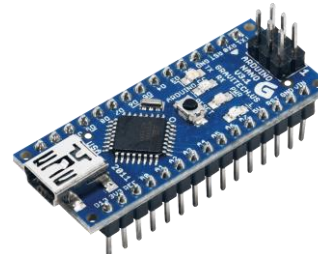
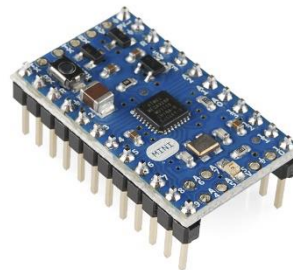
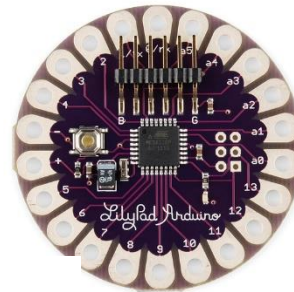
- Είναι φθηνό:
  - Μπορεί κανείς να το αγοράσει με λιγότερα από 100€
  - Μπορεί κανείς να το κατασκευάσει μόνος του
- Cross Platform IDE (Windows, MAC, Linux)
- Open source IDE και επεκτάσεις (shields)
- Δυνατότητα προγραμματισμού ακόμη και με Scratch (S4a, Snap4arduino)
- Χαμηλή κατανάλωση ενέργειας
- Κατάλληλο για αρχάριους χρήστες
- Κατάλληλο για εφαρμογές μικρού μεγέθους





# Πλακέτες Arduino

- Arduino Uno
- Arduino Leonardo
- Arduino LilyPad
- Arduino Mega
- Arduino Nano
- Arduino Mini
- Arduino Mini Pro
- Arduino Due



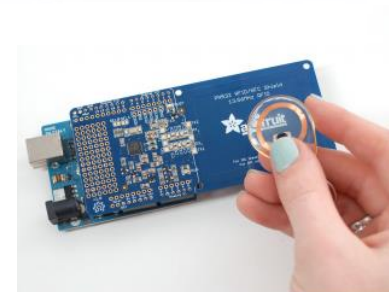
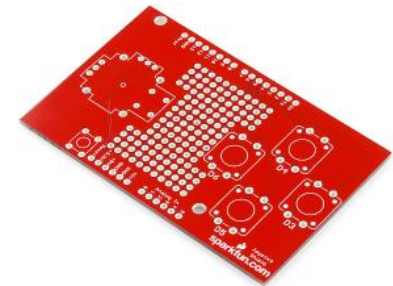
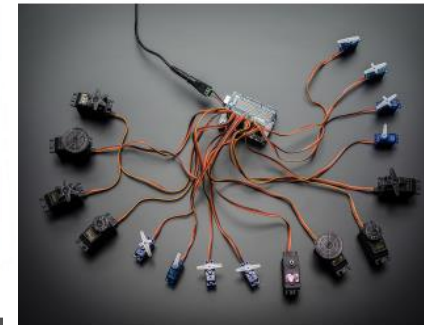


# Προδιαγραφές Arduino Uno

- Microcontroller: ATmega 328
  - 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
  - Flash Memory 32 KB (of which 0.5 KB used by bootloader)
  - SRAM 2 KB (ATmega328)
  - EEPROM 1 KB (ATmega328)
  - Clock Speed 16 MHz
-

# Πρόσθετα (Shields)

- Οθόνη LCD
- Ethernet
- GSM
- WiFi
- PWM/Servo Shield
- Joystick
- Wave Shield
- NFC/RFID Shield





# Η Πλατφόρμα

LED (Φόρτωση προγράμματος)

14 ψηφιακές είσοδοι/έξοδοι

Button (Reset)

LED (Power)

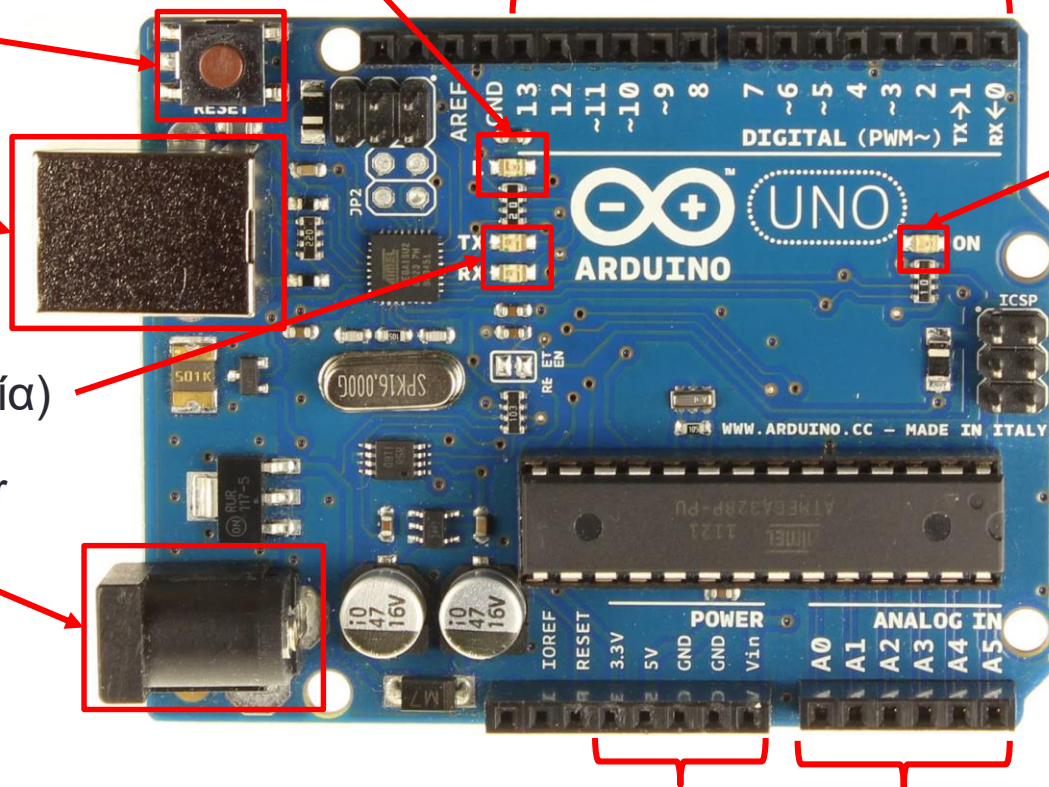
USB

LED (Επικοινωνία)

AC/DC Adapter

Είσοδος DC

Αναλογικές είσοδοι





# Περιβάλλον Προγραμματισμού Arduino IDE: Καταβίβαση και Εγκατάσταση

- Διαθέσιμο στον ιστότοπο:  
<https://www.arduino.cc/en/Main/Software>
  - Τελευταία έκδοση 1.6.8
  - Διαθέσιμο για τα εξής λειτουργικά:
    - Windows
    - Mac OS X
    - Linux
  - Πριν την εκτέλεση του προγράμματος συνδέουμε την πλακέτα μέσω USB στον υπολογιστή (δεν είναι απαραίτητη η σύνδεση εξωτερική πηγής ρεύματος)
-



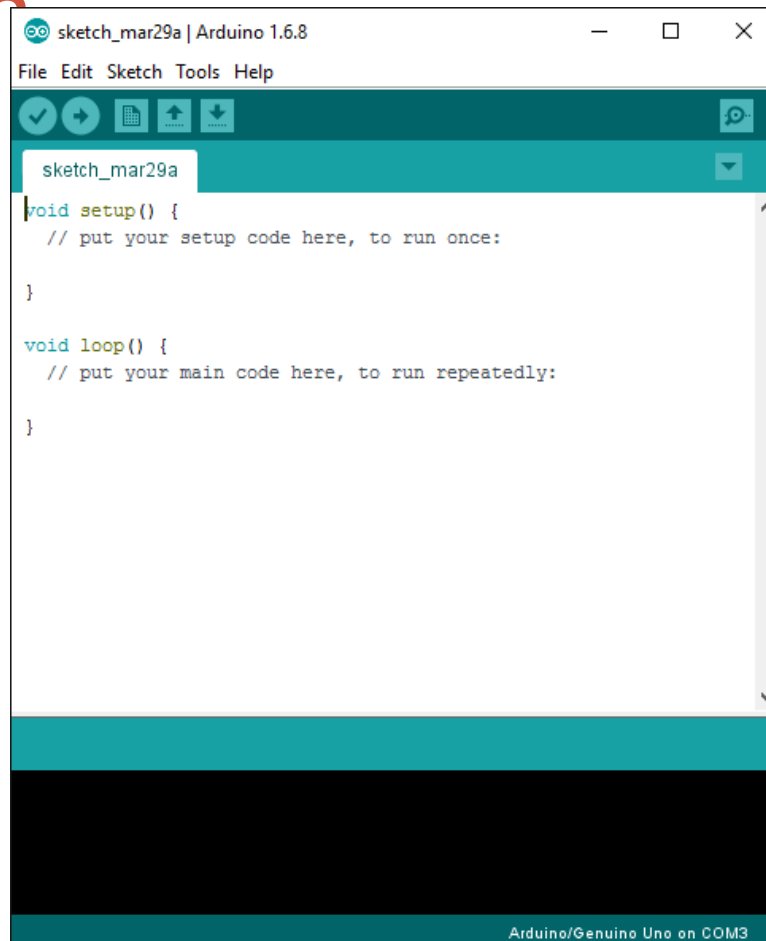
# Περιβάλλον Προγραμματισμού Arduino

## IDE: Δουή

Περιοχή  
Εντολών

Περιοχή  
Κειμένου

Περιοχή  
Μηνυμάτων



Βασικό Μενού

Εικονίδια Συντομεύσεων

# Περιβάλλον Προγραμματισμού Arduino

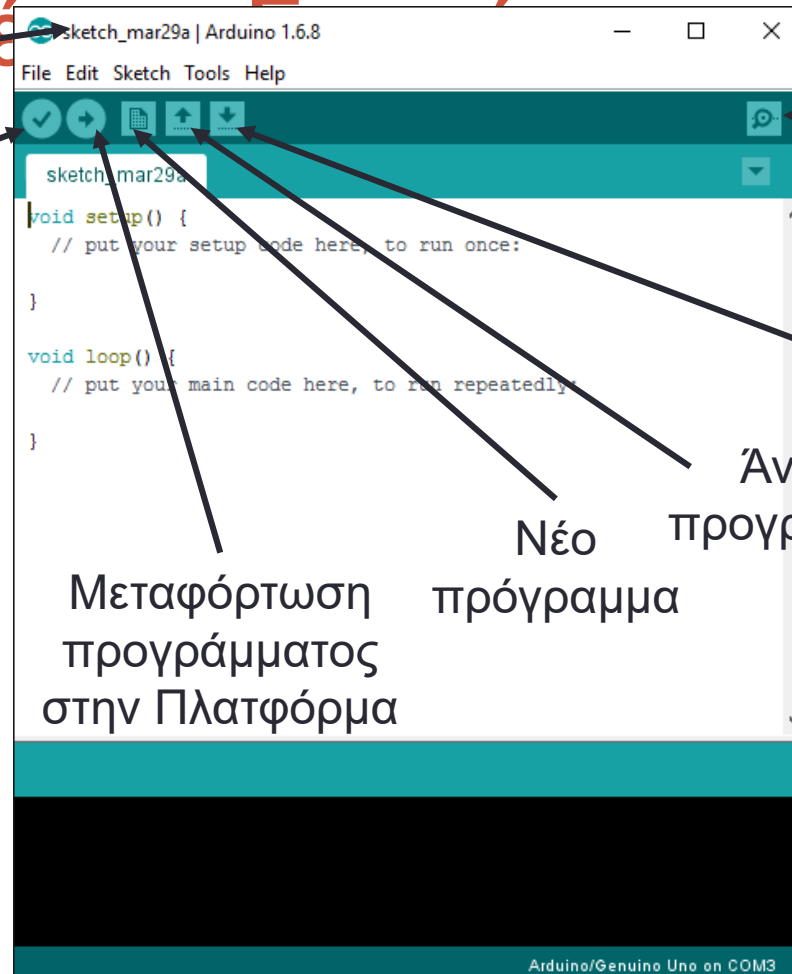
## IDE: Επιμέλεια

Όνομα  
προγράμματος

Μεταγλώττιση  
προγράμματος

Περιοχή σύνταξης  
προγράμματος

Μηνύματα/Λάθη



Serial Monitor

Αποθήκευση  
προγράμματος

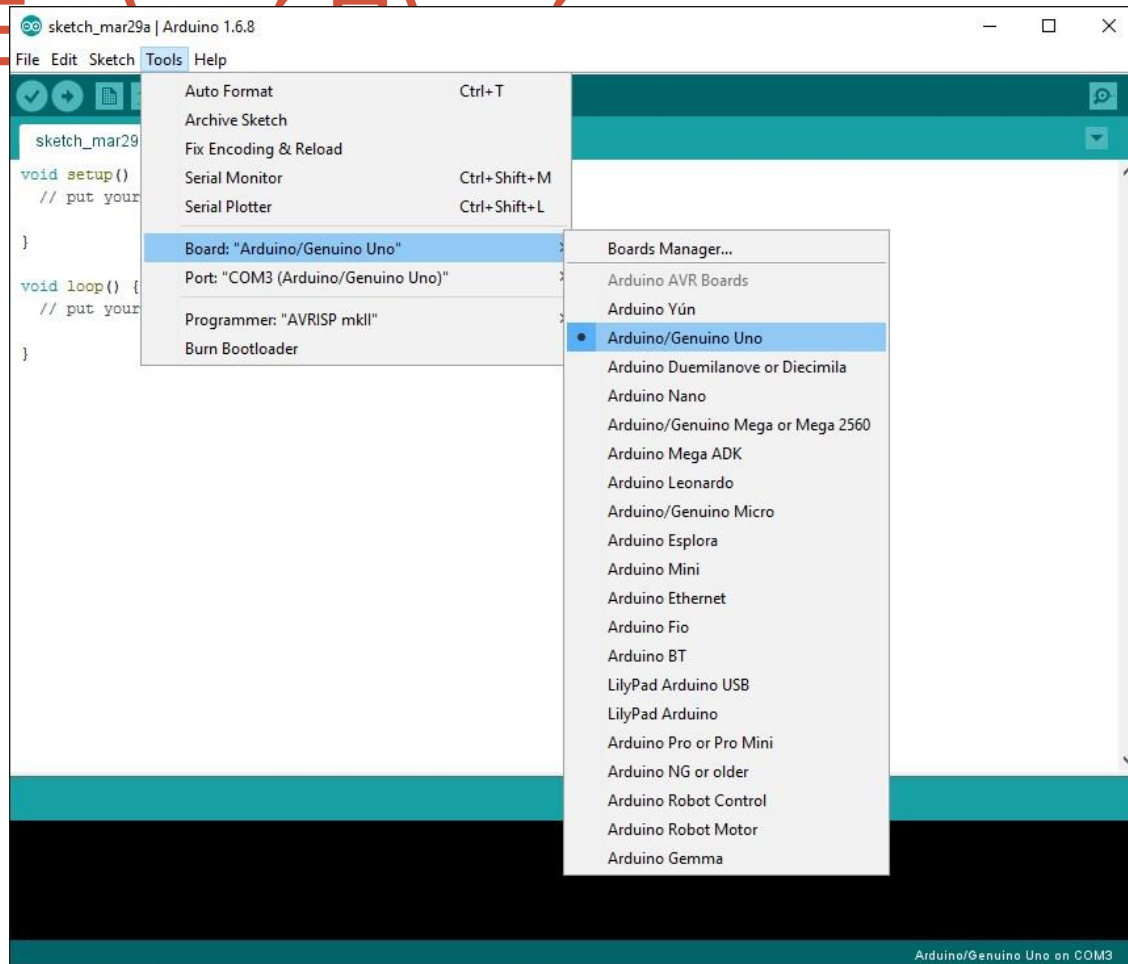
Άνοιγμα  
προγράμματος

Νέο  
πρόγραμμα

Μεταφόρτωση  
προγράμματος  
στην Πλατφόρμα



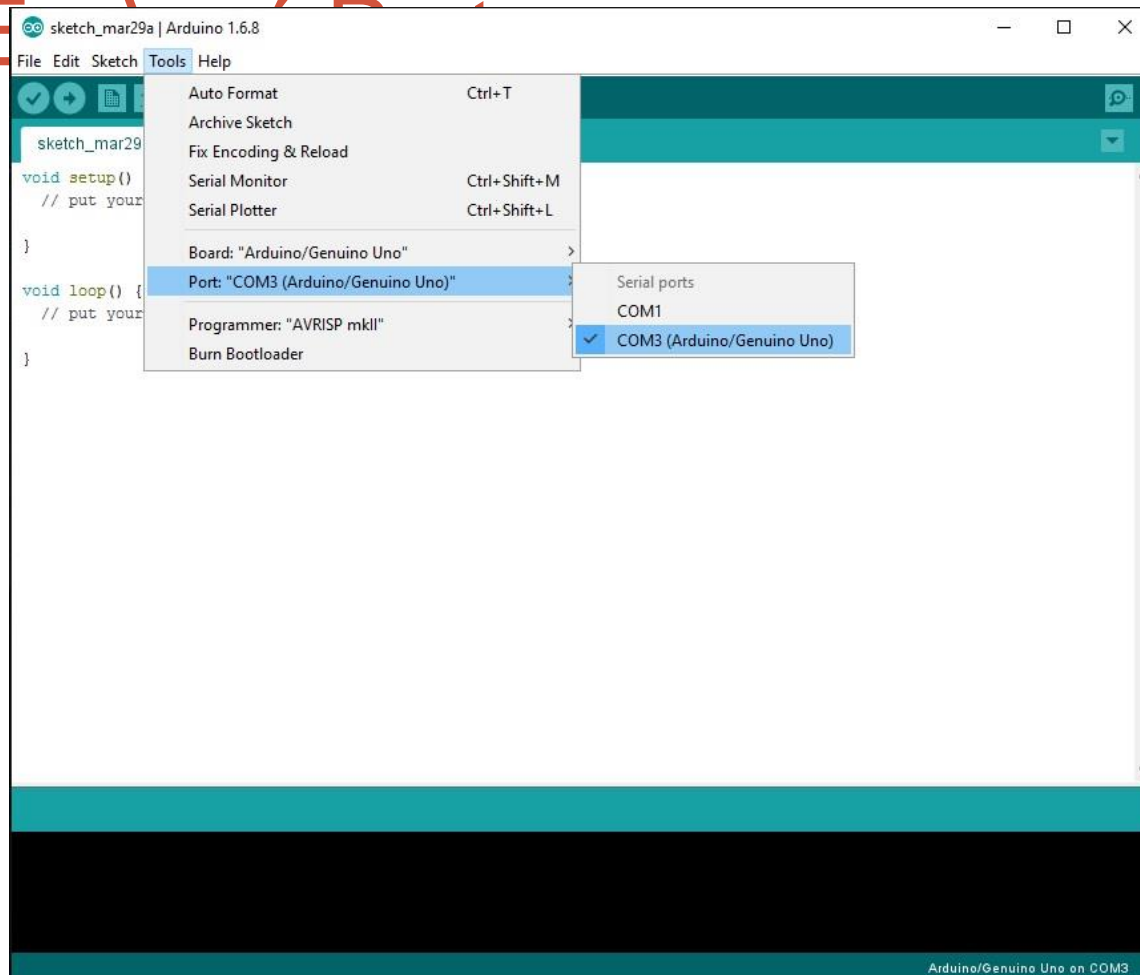
# Περιβάλλον Προγραμματισμού Arduino IDE: Εγκατάσταση







# Περιβάλλον Προγραμματισμού Arduino IDE: Εγκατάσταση



# Δομή Προγράμματος

- void setup()
  - Εκτελείται μόνο μια φορά κατά την έναρξη του προγράμματος (ή όταν πιέζεται το κουμπί reset)
- void loop()
  - Εκτελείται επανειλημμένα
- Το κείμενο που ακολουθεί τις δύο καθέτους “//” αποτελεί ένα σχόλιο που αγνοείται από τον μεταγλωττιστή

```
void setup() {  
    // put your setup code here, to run once:  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```



# Βασικά Στοιχεία της Γλώσσας Προγραμματισμού

- Constants:
  - HIGH/LOW – Ορίζουν το επίπεδο της τάσης εισόδου/εξόδου ενός pin
  - INPUT/OUTPUT – Ορίζουν την κατάσταση ενός pin
- Συναρτήσεις:
  - Επικοινωνία:
    - Serial – Χρησιμοποιείται για επικοινωνία μεταξύ του Arduino και του υπολογιστή
      - begin(speed) – Καθορίζει το ρυθμό δεδομένων σε bps για σειριακή μετάδοση δεδομένων (300, 600, 1200, 2400, 4800, 9600, 14400, 19200, 28800, 38400, 57600, ή 115200)
      - print("text"/value)/println("text"/value) – Τυπώνει δεδομένα στη σειριακή πύλη σε μορφή κειμένου
  - Χρόνο:
    - delay(ms) – Κάνει παύση της εκτέλεσης ενός προγράμματος για το προκαθορισμένο χρονικό διάστημα (millisecond, 1000 ms = 1 s)



# Βασικά Στοιχεία της Γλώσσας Προγραμματισμού

- Συναρτήσεις:
  - Ψηφιακή I/O:
    - `pinMode(pin, mode)` – Καθορίζει αν ένα pin είναι είσοδος ή έξοδος
    - `digitalWrite(pin, value)` – Αναθέτει μια τιμή (HIGH/LOW) σε ένα pin
    - `digitalRead(pin)` – Διαβάζει την τιμή ενός pin (HIGH/LOW)
  - Αναλογική I/O:
    - `analogReference(type)` – Καθορίζει την τάση αναφοράς (μέγιστη τιμή) που χρησιμοποιείται για αναλογική είσοδος (DEFAULT, INTERNAL, INTERNAL1V1, INTERNAL2V56, ή EXTERNAL)
    - `analogRead(pin)` – Διαβάζει την τιμή ενός αναλογικού pin (τιμές μεταξύ 0 και 5 volt που μετασχηματίζονται σε τιμές μεταξύ 0 και 1023)
    - `analogWrite(pin, value)` – Αναθέτει μια αναλογική τιμή σε ένα pin (μεταξύ 0 και 255)

\* Περισσότερες πληροφορίες: <http://www.arduino.cc/en/Reference/HomePage>

# Παράδειγμα 1: LED Blink (On board)

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin 13 as an output.
  pinMode(13, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(13, HIGH); // turn the LED on (HIGH is the voltage level)
  delay(1000);            // wait for a second
  digitalWrite(13, LOW);  // turn the LED off by making the voltage LOW
  delay(1000);            // wait for a second
}
```

Χρησιμοποίησε το  
pin 13 ως ψηφιακή  
έξοδο

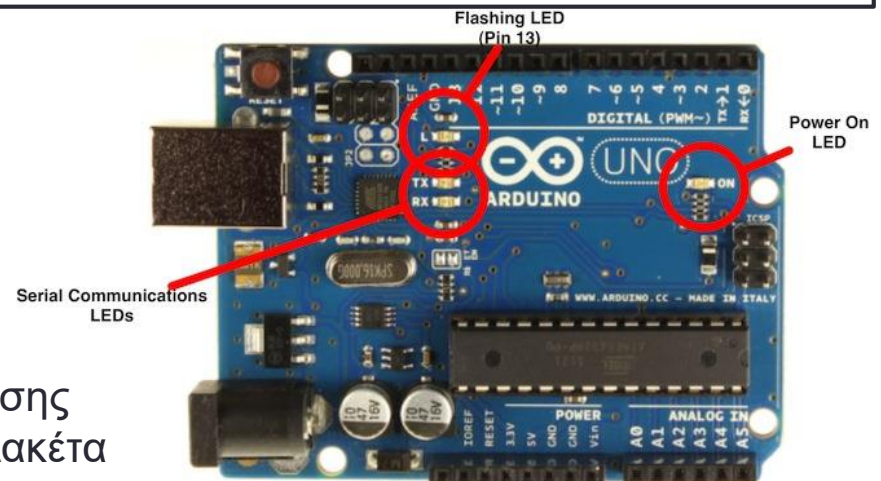
Θέσε την τιμή  
HIGH (+5V) στο  
pin 13

Περίμενε 1000 ms

Θέσε την τιμή LOW  
(0V) στο pin 13

Περίμενε 1000 ms

\* Το pin 13 συνδέεται με το LED ένδειξης φόρτωσης  
προγράμματος (L) που βρίσκεται πάνω στην πλακέτα





# Arduino και Java

- Το περιβάλλον προγραμματισμού Arduino IDE είναι προγραμματισμένο σε Java και μπορεί να επικοινωνήσει μέσω αυτού με τη σειριακή θύρα (serial port)
- Για να επιτευχθεί η επικοινωνία χρησιμοποιείται η βιβλιοθήκη java RXTX (διαθέσιμη για καταβίβαση στον σύνδεσμο <http://rxtx.qbang.org/wiki/index.php/Download>).
- Ο φάκελος περιλαμβάνει:
  - 4 φακέλους ανάλογα με το λειτουργικό του Η/Υ (Linux, Mac OS X, Solaris, Windows)
  - Βιβλιοθήκη RXTXcomm.jar

↑ > rxtx-2.1-7-bins-r2			
Name	Date modified	Type	Size
Linux	04-Feb-06 6:59 PM	File folder	
Mac_OS_X	01-Mar-06 12:06 PM	File folder	
Solaris	04-Feb-06 5:36 PM	File folder	
Windows	04-Feb-06 5:36 PM	File folder	
INSTALL	04-Feb-06 5:58 PM	File	2 KB
RXTXcomm.jar	30-Jan-06 12:23 AM	Executable Jar File	59 KB



# Επικοινωνία Arduino με Java:

## Προγραμματισμός Arduino

- Πριν ξεκινήσουμε την επικοινωνία της πλακέτας Arduino με τη Java θα πρέπει:
  - Να έχουμε εγκαταστήσει το Arduino IDE
  - Να έχουμε φορτώσει στην πλακέτα ένα πρόγραμμα το οποίο να στέλνει δεδομένα στη σειριακή θύρα
    - Ενδεικτικά, ένα πρόγραμμα που στέλνει δεδομένα στη σειριακή θύρα είναι διαθέσιμο στο Arduino IDE (Αρχείο → Παραδείγματα → 01.Basics → AnalogReadSerial) το οποίο διαβάζει το pin A0 και στέλνει τις μετρήσεις στην επιλεγμένη θύρα.

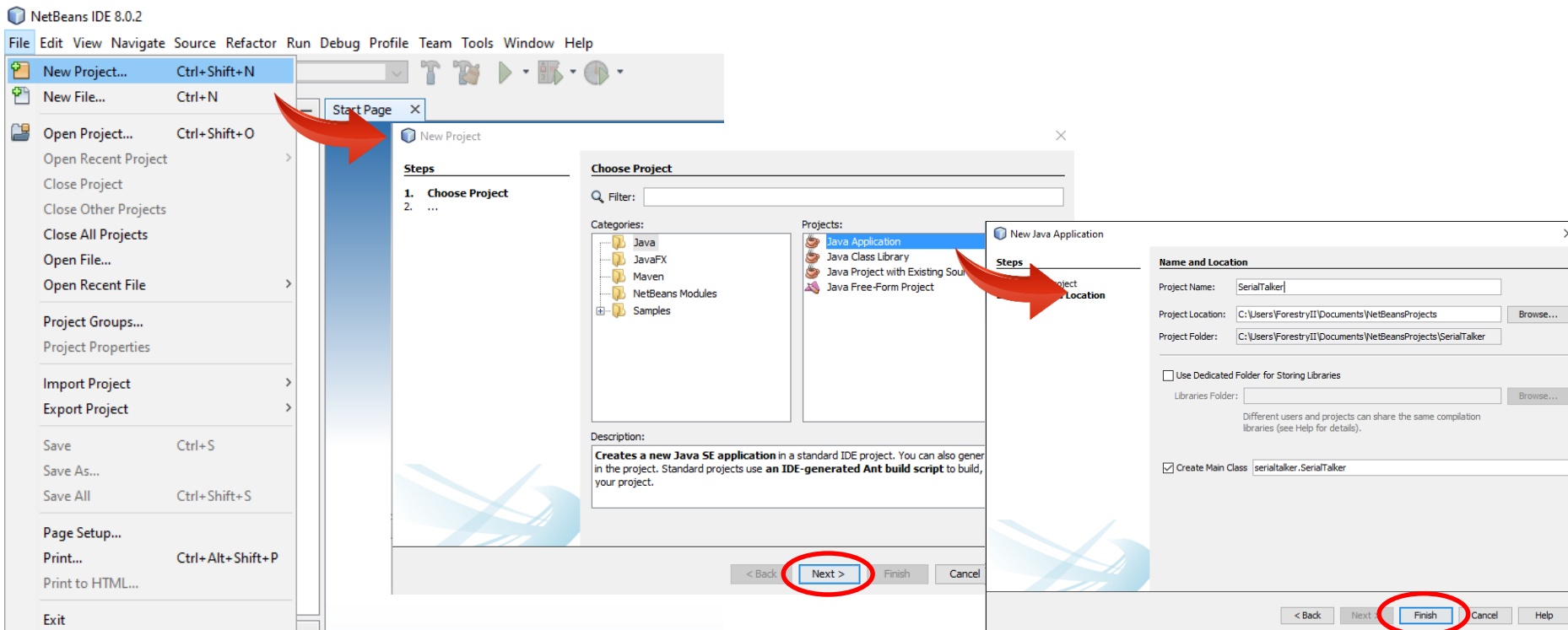
```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1000);          // delay in between reads for stability
}
```



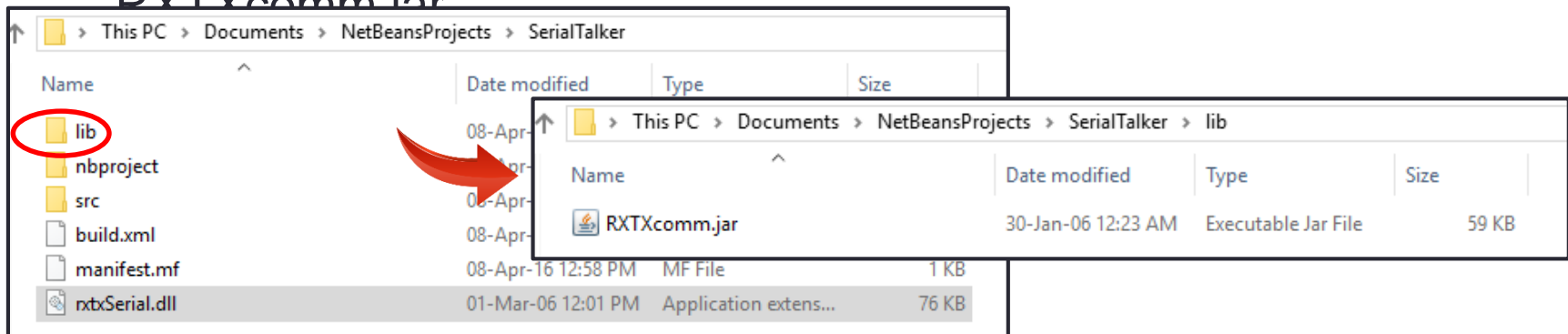
# Επικοινωνία Arduino με Java: Set Up Βιβλιοθήκης RXTX (1/3)

- Δημιουργία project στο NetBeans IDE
  - File → New Project, Επιλέγουμε Java Application → Next, Δίνουμε το όνομα του project (π.χ. SerialTalker) → Finish



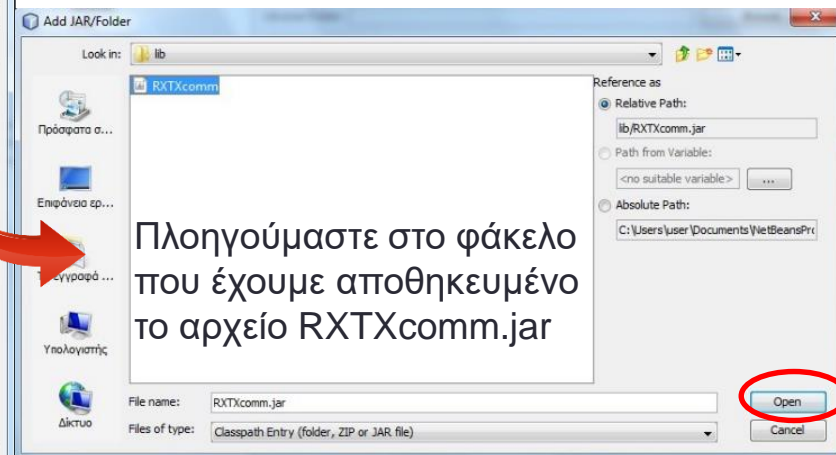
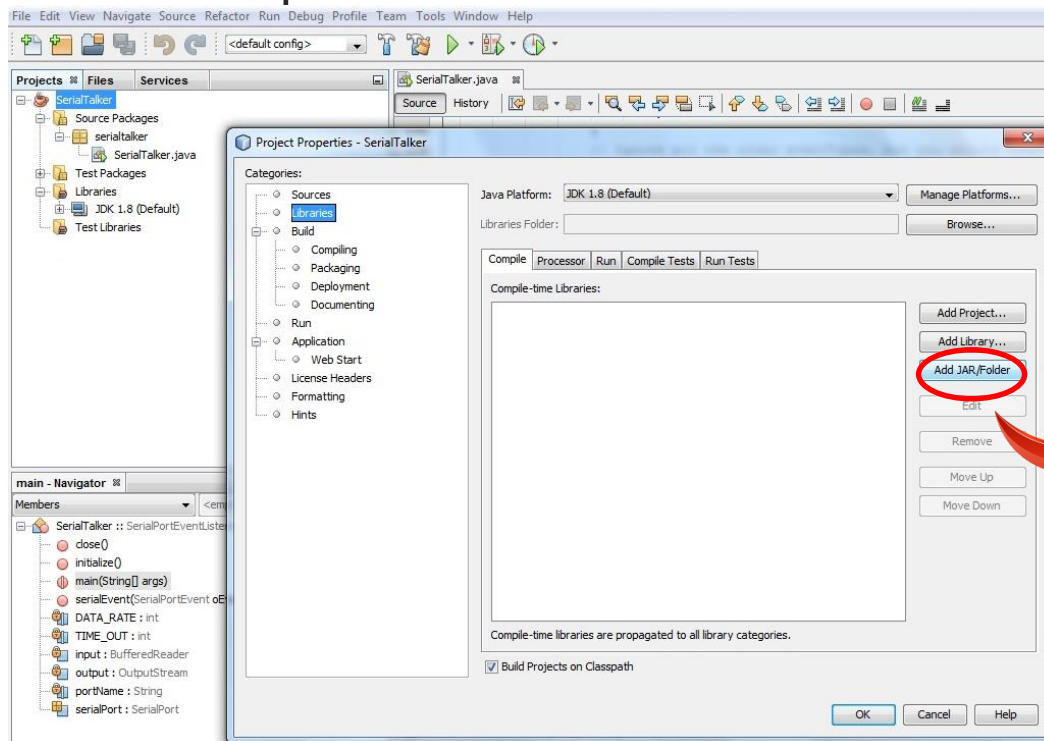
# Επικοινωνία Arduino με Java: Set Up Βιβλιοθήκης RXTX (2/3)

- Στο φάκελο του project  
(Documents\NetBeansProjects\SerialTalker)
  - Αντιγράφουμε το αρχείο rxtxSerial.dll (βρίσκεται στο φάκελο windows\i368-mingw32 του zip)
  - Δημιουργούμε το φάκελο lib και μέσα αντιγράφουμε το αρχείο **RXTXcomm.jar**



# Επικοινωνία Arduino με Java: Set Up Βιβλιοθήκης RXTX (3/3)

- Προσθέτουμε τη βιβλιοθήκη RXTXcomm.jar στο project
  - Επιλέγουμε το project και κάνουμε δεξί κλικ
  - Properties → Libraries → Add JAR/Folder → RXTXcomm.jar



Πλοηγούμαστε στο φάκελο  
που έχουμε αποθηκευμένο  
το αρχείο RXTXcomm.jar

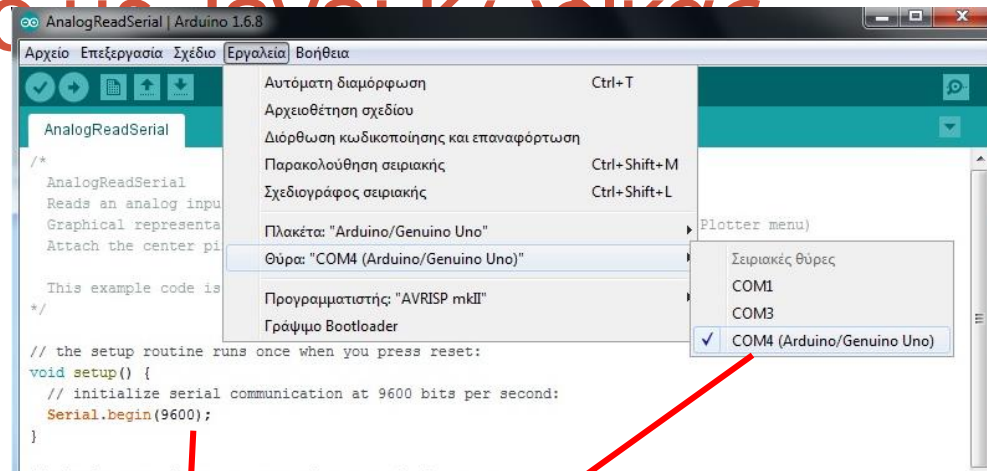




# Επικοινωνία Arduino με Java Κώδικας

(1/1)

```
1 package serialtalker;
2
3 import java.io.BufferedReader;
4 import java.io.InputStreamReader;
5 import java.io.OutputStream;
6 import java.util.Enumuration;
7
8 import gnu.io.CommPortIdentifier;
9 import gnu.io.SerialPort;
10 import gnu.io.SerialPortEvent;
11 import gnu.io.SerialPortEventListener;
12
13 public class SerialTalker implements SerialPortEventListener {
14     SerialPort serialPort;
15     /** Η θύρα που θα χρησιμοποιήσουμε για επικοινωνία
16      * (θα πρέπει να είναι αυτή που μας δίνει το Arduino IDE) */
17     private static final String portName = "COM4";
18
19     /**Ορίζουμε τον BufferedReader ο οποίος θα τροφοδοτείται με δεδομένα από τον
20      * InputStreamReader μετατρέποντας τα bytes σε χαρακτήρες */
21     private BufferedReader input;
22     /** Το stream εξόδου προς τη θύρα */
23     private OutputStream output;
24     /** Milliseconds αναμονής για το άνοιγμα της θύρας */
25     private static final int TIME_OUT = 2000;
26     /** Ο ρυθμός ροής δεδομένων από τη θύρα όπως το έχουμε ορίσει στο arduino */
27     private static final int DATA_RATE = 9600;
```



# Επικοινωνία Arduino με Java: Κώδικας (2/4)

```
29 public void initialize() {  
30     //Ορίζουμε μια μεταβλητή τύπου CommPortIdentifier όπου αποθηκεύεται η θύρα  
31     CommPortIdentifier portId = null;  
32     //Ορίζουμε μια μεταβλητή τύπου enum όπου αποθηκεύονται οι διαθέσιμες θύρες  
33     //για επικοινωνία του H/Y  
34     Enumeration portEnum = CommPortIdentifier.getPortIdentifiers();  
35  
36     //Για κάθε διαθέσιμη θύρα ελέγχουμε αν είναι η θύρα του arduino  
37     while (portEnum.hasMoreElements()) {  
38         CommPortIdentifier currPortId = (CommPortIdentifier) portEnum.nextElement();  
39         System.out.println("port"+currPortId.getName());  
40  
41         if (currPortId.getName().equals(portName)) {  
42             //Θέτουμε την θύρα του arduino στη μεταβλητή portId  
43             portId = currPortId;  
44             break;  
45         }  
46     }  
47  
48  
49     //Αν δεν βρεθεί η θύρα στις διαθέσιμες εκτυπώνεται αντίστοιχο μήνυμα στην οθόνη  
50     if (portId == null) {  
51         System.out.println("Could not find COM port.");  
52         return;  
53     }
```

# Επικοινωνία Arduino με Java: Κώδικας

```
55 try {
56     // Ανοίγουμε τη σειριακή θύρα
57     serialPort = (SerialPort) portId.open(this.getClass().getName(), TIME_OUT);
58
59     // Θέτουμε τις παραμέτρους της θύρας
60     serialPort.setSerialPortParams(DATA_RATE, SerialPort.DATABITS_8,
61                                     SerialPort.STOPBITS_1, SerialPort.PARITY_NONE);
62
63     // Ανοίγουμε τα streams ροής δεδομένων
64     input = new BufferedReader(new InputStreamReader(serialPort.getInputStream()));
65     output = serialPort.getOutputStream();
66
67     // Προσθέτουμε τους event listeners
68     serialPort.addEventListener(this);
69     serialPort.notifyOnDataAvailable(true);
70
71     } catch (Exception e) {
72         System.err.println(e.toString());
73     }
74 }
75
76 /**Συνάρτηση που αναλαμβάνει το κλείσιμο της θύρας όταν σταματήσει η επικοινωνία */
77 public synchronized void close() {
78     if (serialPort != null) {
79         serialPort.removeEventListener();
80         serialPort.close();
81     }
82 }
```

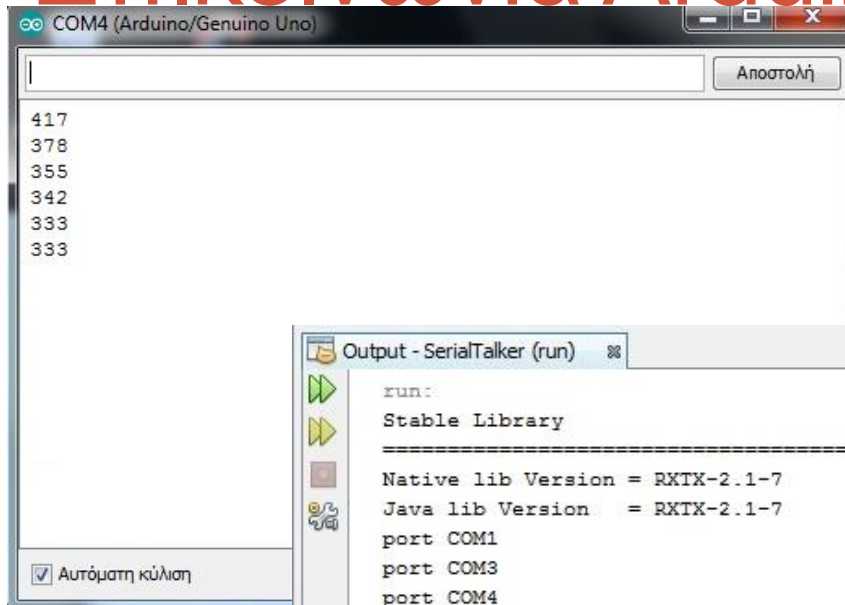
# Επικοινωνία Arduino με Java: Κώδικας (4/4)

```
84 |  
85 |  
86 |  
87 |  
88 |  
89 |  
90 |  
91 |  
92 |  
93 |  
94 |  
95 |  
96 |  
97 |  
98 |  
99 |  
100 |  
101 |  
102 |  
103 |  
104 |  
105 |  
106 |
```

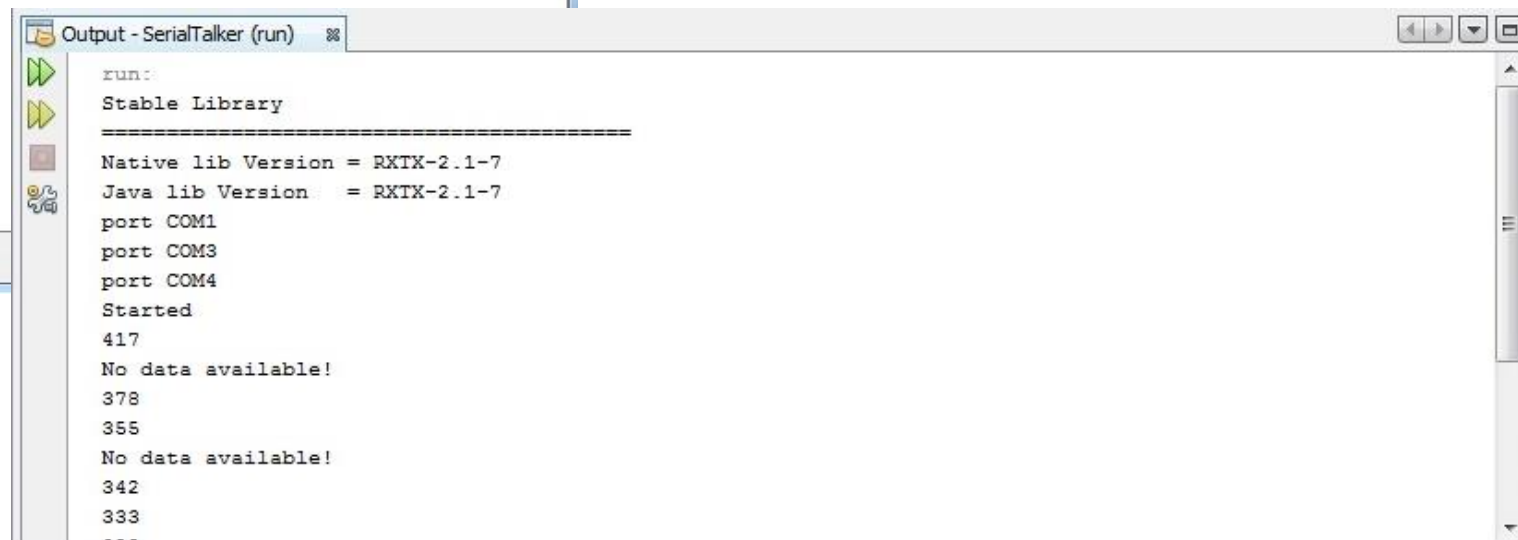
```
/**Συνάρτηση που διαχειρίζεται τα event στη σειριακή θύρα  
 * (διαβάζει δεδομένα και τα εκτυπώνει) */  
public synchronized void serialEvent(SerialPortEvent oEvent) {  
    if (oEvent.getEventType() == SerialPortEvent.DATA_AVAILABLE) {  
        try {  
            String inputLine=input.readLine();  
            System.out.println(inputLine);  
        } catch (Exception e) {  
            System.err.println(e.toString());  
        }  
    }  
}  
  
public static void main(String[] args) throws Exception {  
    //Δημιουργούμε ένα instance της κλάσης  
    SerialTalker main = new SerialTalker();  
  
    //Καλούμε τη συνάρτηση αρχικοποίησης της κλάσης  
    main.initialize();  
  
    System.out.println("Started");  
}
```



# Επικοινωνία Arduino με Java: Έξοδος



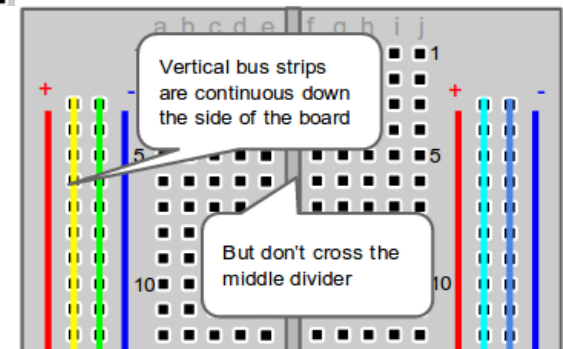
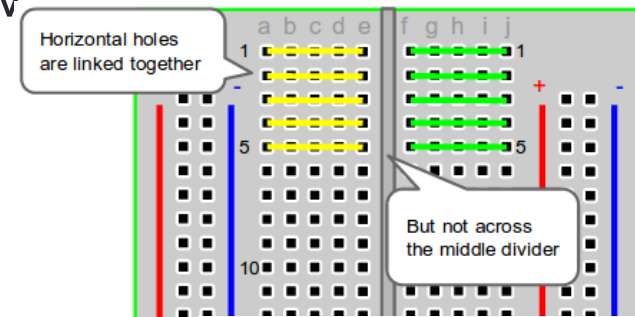
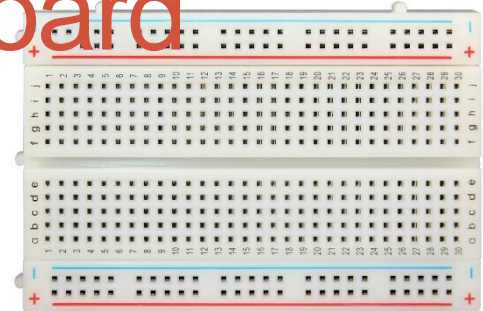
Netbeans Output



\* Περισσότερες πληροφορίες σχετικά με την επικοινωνία Arduino με χρήση Java στο σύνδεσμο <http://playground.arduino.cc/Interfacing/Java>

# Απαραίτητα Υλικά: Breadboard

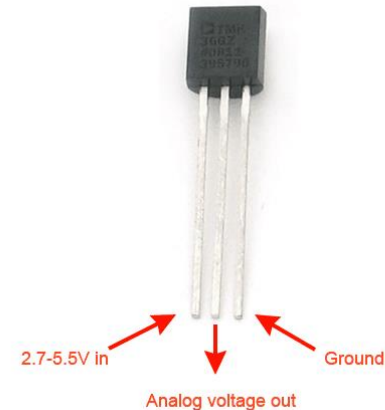
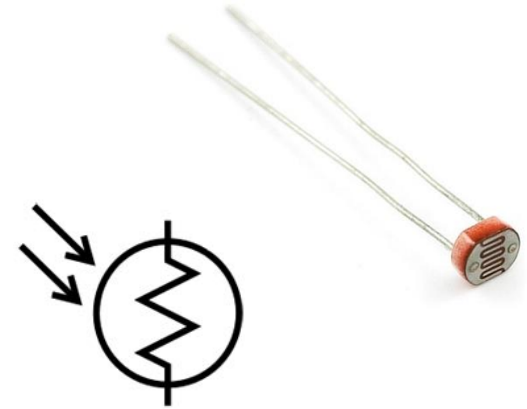
- Απαραίτητο για την ανάπτυξη κατασκευών.
- Σημαντικά σημεία:
  - Η χρήση του είναι πιο εύκολη από την κόλληση
  - Έχει πολλές μικρές τρύπες όπου συνδέονται τα διάφορα στοιχεία (π.χ. αισθητήρες, αντιστάσεις)
- Οι οριζόντιες ομάδες τρυπών είναι συνδεδεμένες.
- Οι στήλες της τάσης και της γείωσης συνδέονται κάθετα





# Απαραίτητα Υλικά: Αισθητήρες

- Φωτοαντίσταση:
  - Η αρχική τιμή της αντίστασης είναι αρκετά μεγάλη (ΜΩ) και μειώνεται με την ένταση του φωτός που προσπίπτει στην επιφάνεια τους
  - Χρησιμοποιείται κυρίως για διακόπτες On/Off.
  - Δεν έχει πολικότητα
- Αισθητήρας θερμοκρασίας:
  - Παρέχει μια τάση εξόδου που είναι γραμμικά ανάλογη με την αλλαγή της θερμοκρασίας



# Απαραίτητα Υλικά: LED και Αντιστάσεις

## • LED:

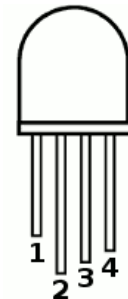
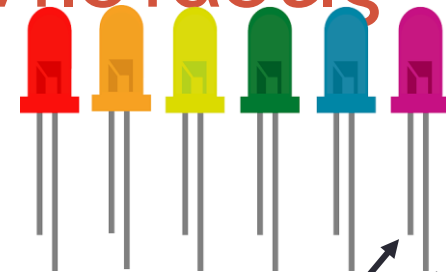
- Δύο ακροδεκτών (μονοχρωματικά)
- Τεσσάρων ακροδεκτών (RGB, 16.777.216 χρώματα)

## • Αντίσταση:

- Περιορίζει τη ροή ρεύματος στο κύκλωμα
- Η τιμή τους μετράται σε Ohm ( $\Omega$ )
- Οι χρωματιστές λωρίδες πάνω τους υποδεικνύουν την τιμή τους
- Μια αντίσταση τουλάχιστον 220  $\Omega$  πρέπει να συνδέεται σε σειρά με ένα LED ώστε να μην καταστρέφονται

## • Κουμπιά:

- Δίνουν τη δυνατότητα στο χρήστη να παρεμβαίνει στο κύκλωμα διακόπτοντας ή παρέχοντας ρεύμα σε ένα κύκλωμα.



**RGB LED**

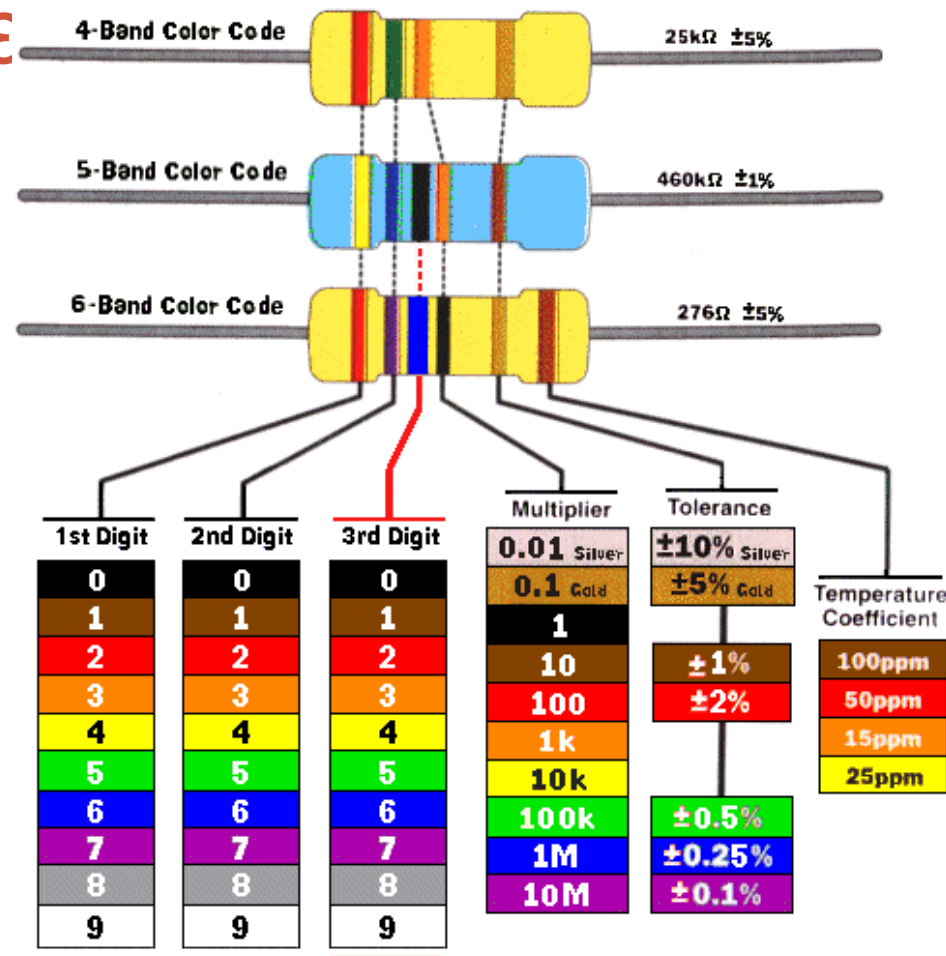
- 1: Green (+)
- 2: Ground (-)
- 3: Blue (+)
- 4: Red (+)

Κάθοδος (-)

Άνοδος (+)



# Απαραίτητα Υλικά: Χρωματικός Κώδικας Αντιστάσε



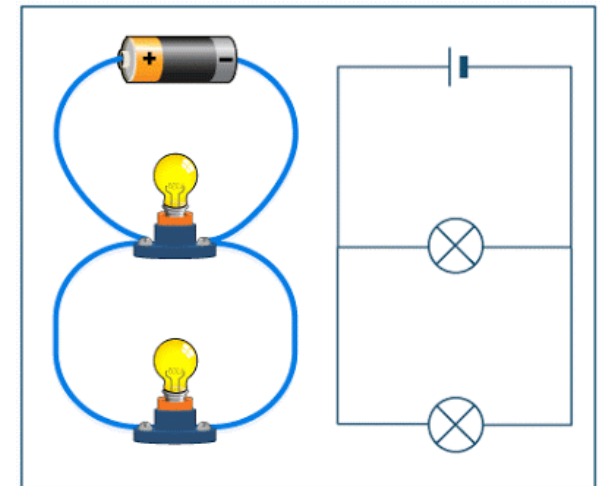
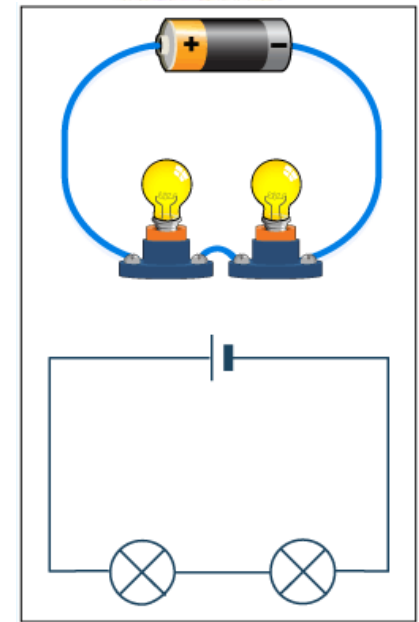
# Σύνδεση Στοιχείων: Σε Σειρά και Παράλληλα

- Σύνδεση σε σειρά:

- Δύο στοιχεία λέμε ότι συνδέονται σε σειρά όταν διαρρέονται από το ίδιο ρεύμα. Όποια ηλεκτρόνια περάσουν από τον έναν θα περάσουν και από τον άλλον.

- Παράλληλη σύνδεση:

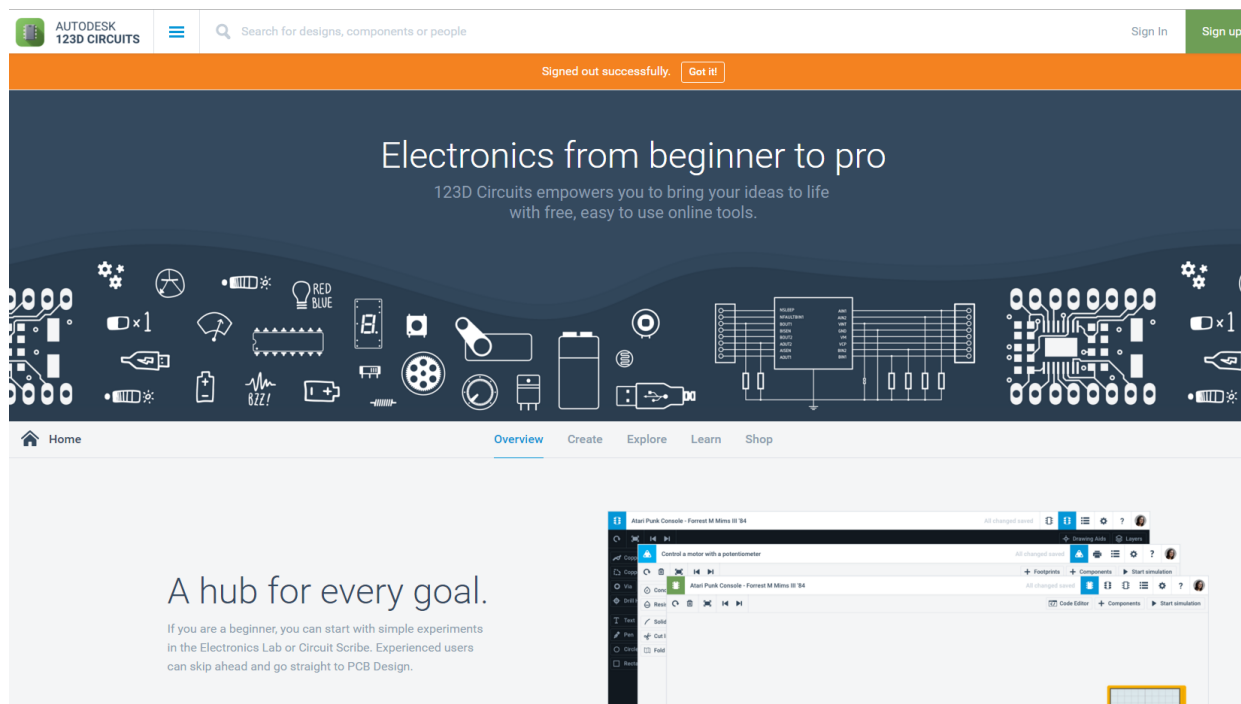
- Δύο στοιχεία λέμε ότι συνδέονται παράλληλα όταν έχουν ίδια τάση ή πιο απλά όταν οι δύο αντιστάτες έχουν κοινά άκρα.





# Προσομοίωση Ηλεκτρονικών Κυκλωμάτων

- Autodesk 123D Circuits
- Δωρεάν προσβάσιμο online στον ιστότοπο:  
<https://123d.circuits.io/>





# Autodesk 123D Circuits: Δημιουργία Λογαριασμού

**AUTODESK 123D CIRCUITS**

Search for designs, components or people

Sign In **Sign up**

**AUTODESK 123D CIRCUITS**

Username

like SuperPerson

Password

Password

Your Parent's Email

Your Parent's Email

By clicking on Create Account, you agree to the [Terms](#) and [Privacy Statement](#).

**Create Account**

Already a user? [Sign In](#)

**AUTODESK 123D CIRCUITS**

Country

United States

Birthday

Month Day Year

**Next**

Already a user? [Sign In](#)





https://123d.circuits.io

AUTODESK 123D CIRCUITS

Search for designs, components or people

+ New

Home

Overview Create Explore Learn Shop

Your recent designs

LED Blink Mar 29, 2016 Edit

The LOVE-O-METER Mar 29, 2016 Edit

Show all my designs

**NEW: Check out bendable wires**

Bendable wires allow you to route your wires any way you like to make complex circuits much easier to read. Also more colors and an easy way to change them.

Learn More

What others are doing

Γενική  
επισκόπηση

Υλικό  
εκμάθησης

Δημιουργία  
νέων  
κυκλωμάτων

Περιήγηση στα  
κυκλώματα  
άλλων  
χρηστών



Τμήμα Δασολογίας,  
Επιστημών Ξύλου & Σχεδιασμού



MSc - ΠΡΟΗΓΜΕΝΕΣ ΜΕΘΟΔΟΙ  
ΣΧΕΔΙΑΣΜΟΥ, ΤΕΧΝΟΛΟΓΙΑΣ &  
MANAGEMENT ΠΡΟΪΟΝΤΩΝ  
ΑΠΟ ΞΥΛΟ

# Autodesk 123D Circuits: Δημιουργία Νέου Κυκλώματος

Autodesk 123D Circuits

Search for designs, components or people

Home Overview Create Explore Learn Shop

Your recent designs

- LED Blink Mar 29, 2016 Edit
- The LOVE-O-METER Mar 29, 2016 Edit

Show all my designs

**NEW: Check out bendable wires**

Bendable wires allow you to route your wires any way you like to make complex circuits much easier to read. Also more colors and an easy way to change them.

[Learn More](#)

What others are doing

All Hubs

- New Electronics Lab**
- New PCB Design
- New PCB Component
- Import Eagle Project
- New Circuitscribe
- New MESH GPIO PCB

# Autodesk 123D Circuits: Περιβάλλον

https://123d.circuits.io/circuits/1847729-the-unnamed-circuit/edit

The Unnamed Circuit

Code Editor

Components

Start Simulation

Όνομα κυκλώματος

Επιστροφή στην Αρχική σελίδα

Εκκίνηση προσομοίωσης

Ηλεκτρονικά στοιχεία

Επεξεργασία κώδικα

Breadboard

All Components Grid All Components List Arduino Basic Kit DFRobot Beginner Kit

Resistor

LED

LED RGB

Light bulb

Power Supply

9V Battery



# Πρόγραμμα 2: LED Blink (Γραμμοκώδη)

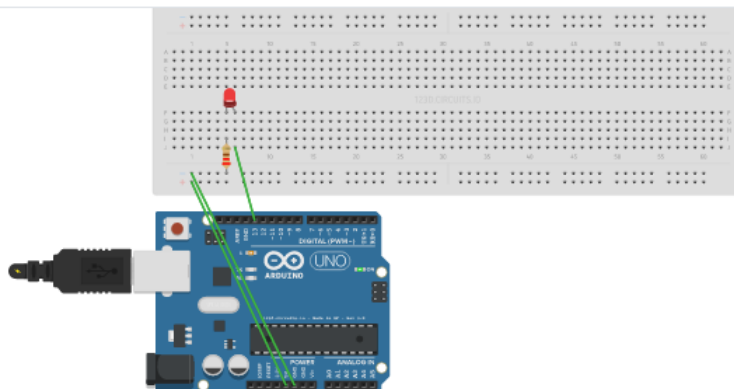
LED Blink | Electronics Lab

Edit

Show more

Stop Simulation

Code Editor



1 (Arduino Uno R3) ▾

Upload & Run

Download Code

Debugger

Serial Monitor

```
1 // Pin 13 has an LED connected on most Arduino boards.
2 // give it a name:
3 int led = 13;
4
5 // the setup routine runs once when you press reset:
6 void setup() {
7   // initialize the digital pin as an output.
8   pinMode(led, OUTPUT);
9 }
10
11 // the loop routine runs over and over again forever:
12 void loop() {
13   digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
14   delay(1000);               // wait for a second
15   digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
16   delay(1000);               // wait for a second
17 }
```



# Ερωτήσεις Αξιολόγησης

1. Τι είναι το Arduino και σε τι βασίζεται;
2. Ποια τα πλεονεκτήματα του Arduino;
3. Ποια είναι η βασική δομή ενός προγράμματος Arduino;
4. Τι ορίζουν οι σταθερές HIGH και LOW;
5. Τι καθορίζει η συνάρτηση `pinMode(pin, mode)`; Δώστε ένα παράδειγμα χρήσης της.
6. Γράψτε τον κώδικα που απαιτείται για να αναβοσβήνει ένα LED που είναι συνδεδεμένο στο pin 13 του Arduino.
7. Πως συνδέονται οι τρύπες ενός breadboard;
8. Η φωτοαντίσταση έχει πολικότητα; Αν ναι, πως την αναγνωρίζουμε;
9. Τα LED έχουν πολικότητα; Αν ναι, πως την αναγνωρίζουμε;
10. Ποιος ο ρόλος της αντίστασης;
11. Πως βρίσκουμε το μέγεθος μιας αντίστασης;

# Συμπληρωματική Βιβλιογραφία

- «*Introduction to Arduino: A piece of cake!*», A. G. Smith, Εκδόσεις Πανεπιστημίου Πατρών, 2011  
(<http://www.introtoarduino.com/downloads/IntroArduinoBook.pdf>).
- «*Προγραμματίζοντας με τον μικροελεγκτή Arduino*», Ε. Πουλάκης, Ηράκλειο, 2015  
(<http://users.sch.gr/manpoul/docs/arduino/ProgrammingArduino.pdf>).
- «*Ανάπτυξη Εφαρμογών με το Arduino: Ένας πλήρης οδηγός για αρχάριους και προχωρημένους*», Π. Παπάζογλου, Σ. Π. Λιώνης, Εκδόσεις Τζιόλα, 2015.
- «*Ηλεκτροτεχνία Ι: Εισαγωγικές Έννοιες*»  
(<http://inflab.kard.teilar.gr/elli/docs/Electrical%20Circuits.pdf>).
- «*Εισαγωγικά Στοιχεία Ηλεκτρονικών*», Ι. Σ. Χαλκιάδης  
([http://www.ee.teihal.gr/labs/electronics/web/downloads/theory/01\\_principles.pdf](http://www.ee.teihal.gr/labs/electronics/web/downloads/theory/01_principles.pdf)).





Τμήμα Αρχιτεκτονικής,  
Επιστήμη Ξύλου & Σχεδιασμού



MSc - ΠΡΟΗΓΜΕΝΕΣ ΜΕΘΟΔΟΙ  
ΣΧΕΔΙΑΣΜΟΥ, ΤΕΧΝΟΛΟΓΙΑΣ &  
MANAGEMENT ΠΡΟΪΟΝΤΩΝ  
ΑΠΟ ΞΥΛΟ

# Ερωτήσεις;

---