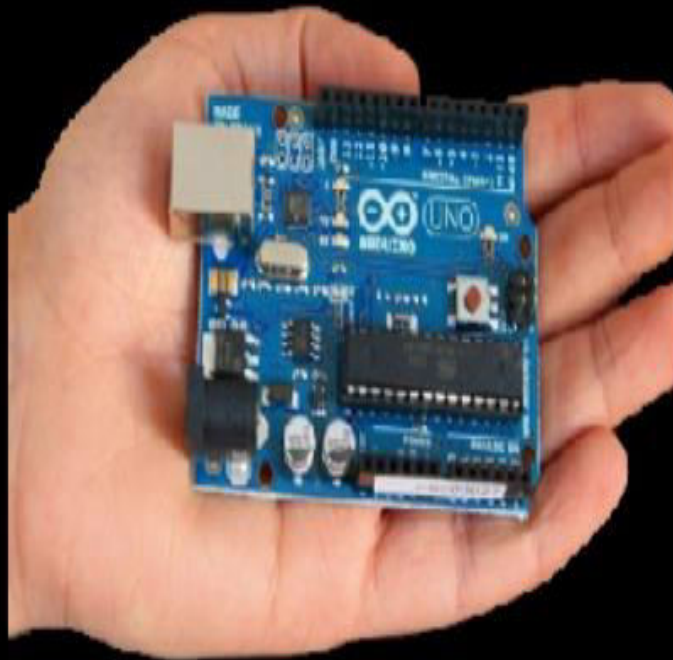


# ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO

Βλάχος Ιωάννης  
Ηλ/γος Μηχανικός (Τ.Ε.Ι) , MSc in Informatics

Ionian University  
8/3/2019

# Arduino Platform



Μια ελεύθερη υπολογιστική  
πλατφόρμα με  
απεριόριστες δυνατότητες

# Arduino – Τι είναι;

Οι δημιουργοί (Massimo Banzi και David Cuartielles)

είναι μια «ανοικτού κώδικα» πλατφόρμα «πρωτοτυποποίησης» ηλεκτρονικών βασισμένη σε ευέλικτο hardware και software που προορίζεται για οποιονδήποτε έχει λίγη προγραμματιστική εμπειρία, στοιχειώδεις γνώσεις ηλεκτρονικών και ενδιαφέρεται να δημιουργήσει διαδραστικά αντικείμενα ή περιβάλλοντα.

**[el.wikipedia.org](http://el.wikipedia.org)**

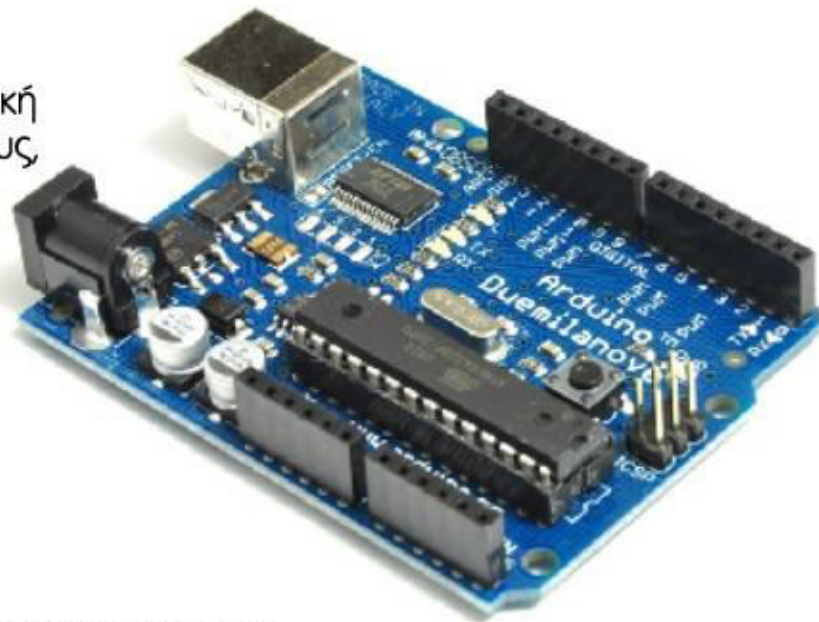
είναι μια υπολογιστική πλατφόρμα βασισμένη σε μια απλή μητρική πλακέτα με ενσωματωμένο μικροελεγκτή και εισόδους/εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα Wiring.

**[Microplanet.gr](http://Microplanet.gr)**

είναι ένα εργαλείο για να κατασκευάσουμε ένα υπολογιστικό σύστημα με την έννοια ότι αυτό θα ελέγχει συσκευές του φυσικού κόσμου, σε αντίθεση με τον κοινό σας Ηλεκτρονικό Υπολογιστή.

**Εγώ, ένας απλός χρήστης**

Είναι ένα ολοκληρωμένο ηλεκτρονικό κύκλωμα με ενσωματωμένο μικροελεγκτή, αναλογικές και ψηφιακές εισόδους και εξόδους, του οποίου τα σχέδια και το software διανέμονται ελεύθερα και δωρεάν,



# Arduino – Ιστορικό

2005

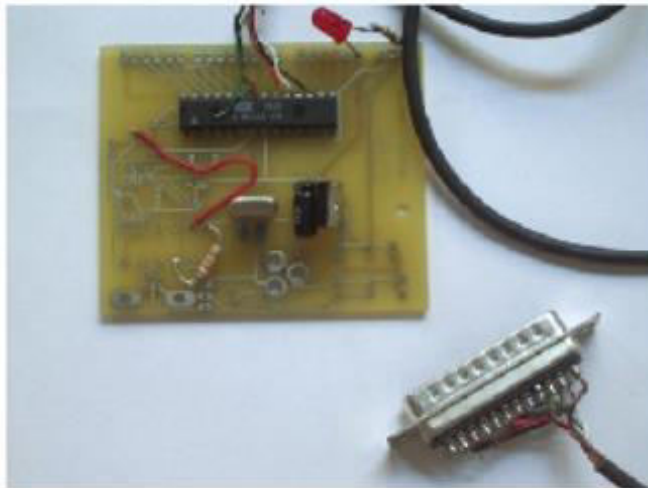
Δημιουργήθηκε στην πόλη Ιβρέα, όπου είναι η έδρα της εταιρίας Olivetti, από τους Massimo Banzi και David Cuartielles.

2011

Πάνω από 300.000 arduino boards έχουν πουληθεί σε όλο το κόσμο

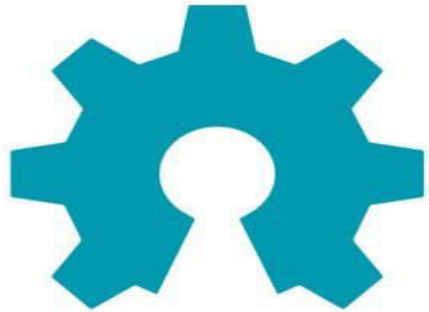


Η ομάδα ανάπτυξης του Arduino:  
David Cuartielles, Gianluca Martino,  
Tom Igoe, David Mellis, και Massimo  
Banzi



Το πρωτότυπο που κατασκευάστηκε το 2005.

# Open Hardware – Open Source Πλατφόρμα Ελεύθερου Υλικού - Λογισμικού



open source  
hardware

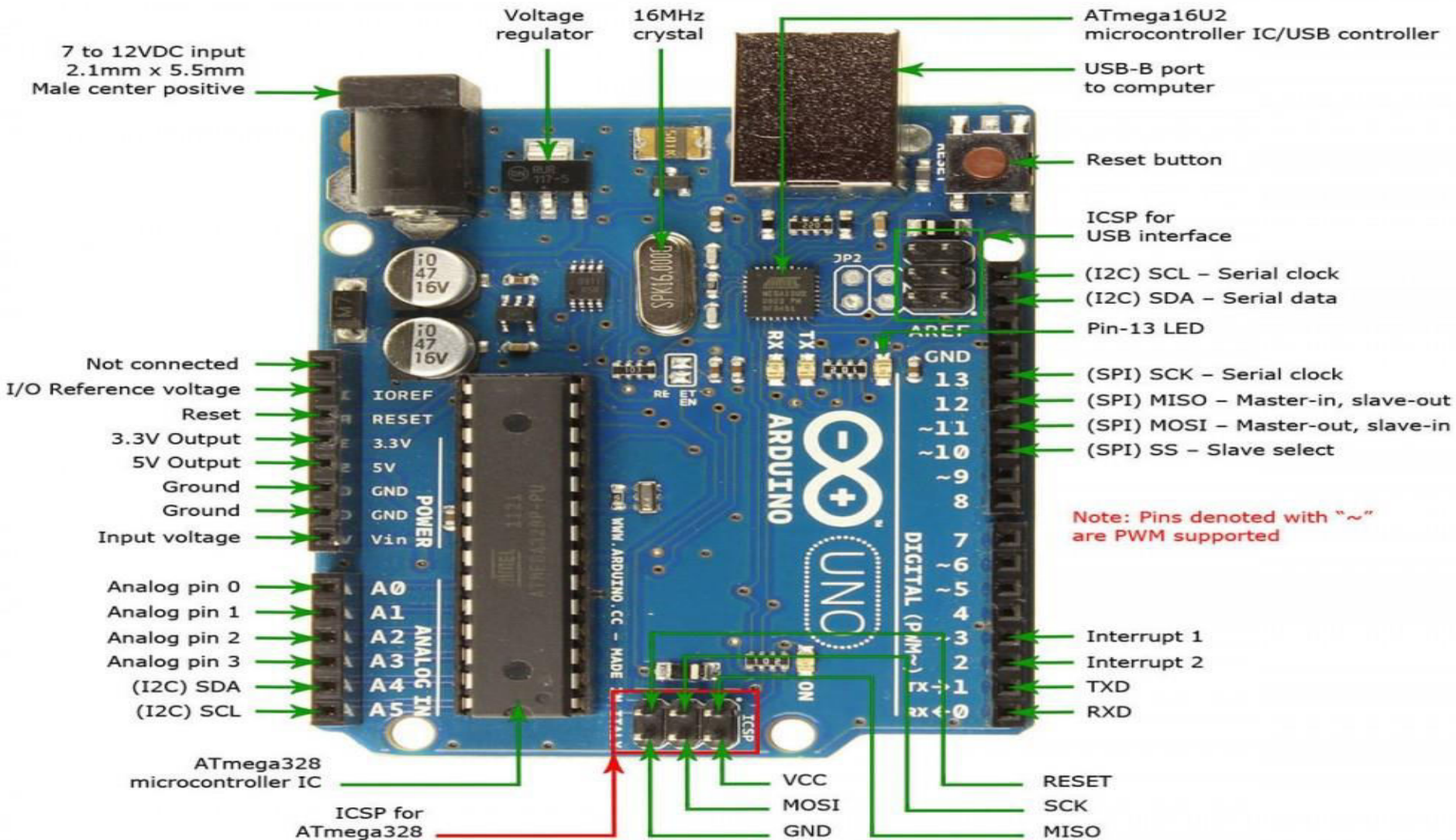


open source





# Arduino Uno – Η πιο συνηθισμένη Έκδοση



Digital output  
~: PWM.  
0,1: Serial port.

In circuit Serial  
programming

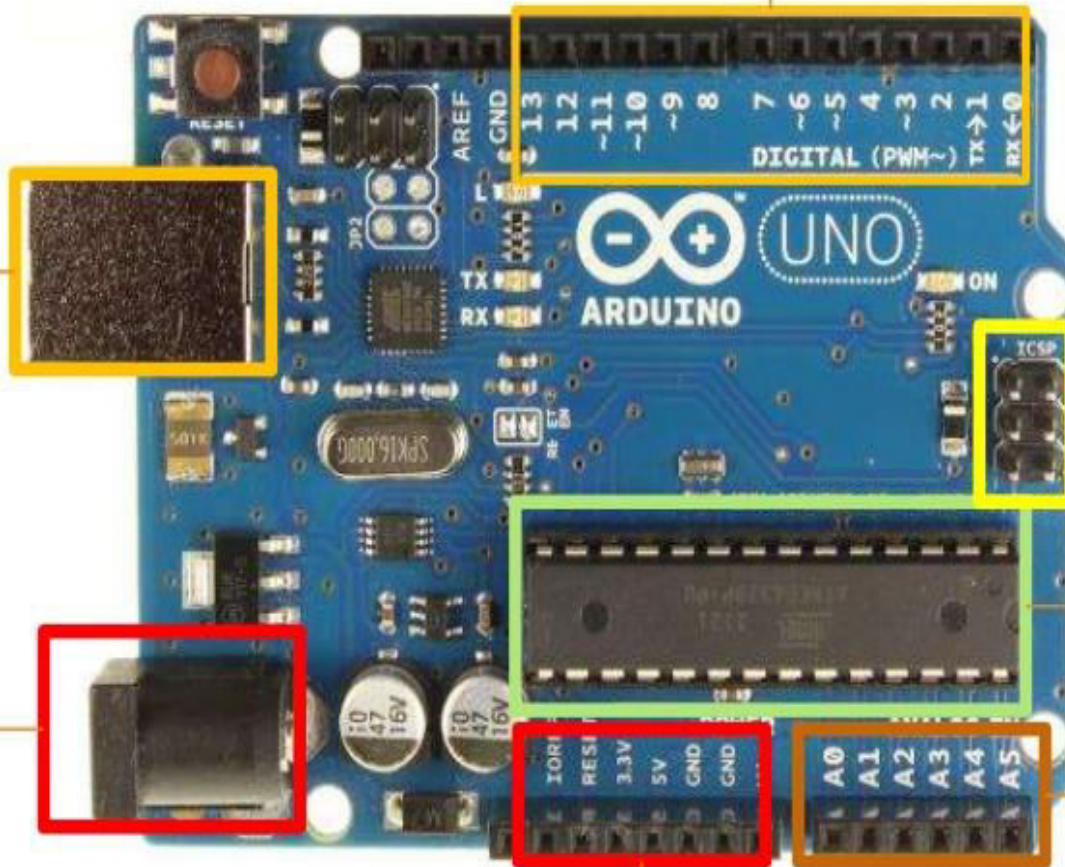
USB port

Atmel  
MicroController

Power input

Power Supply

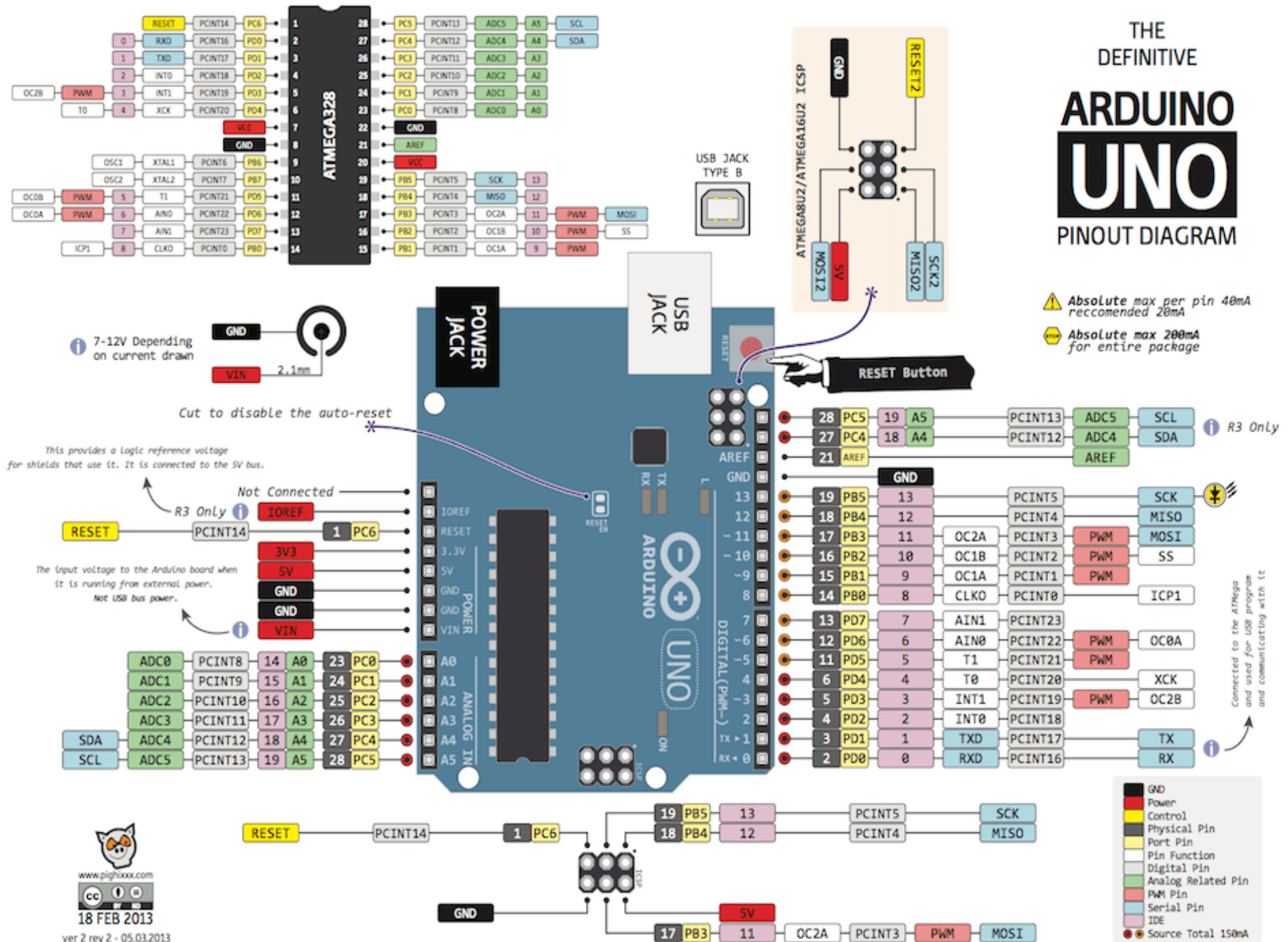
Analog input.





# THE DEFINITIVE ARDUINO UNO PINOUT DIAGRAM

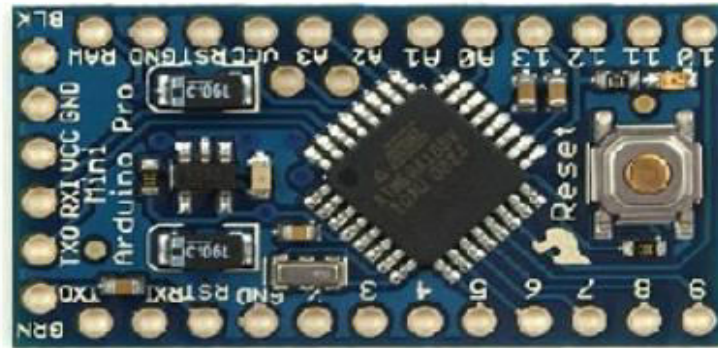
- ⚠ Absolute max per pin 40mA recommended 20mA
- ⚡ Absolute max 200mA for entire package





**Άλλα μοντέλα**

# Arduino Pro Mini



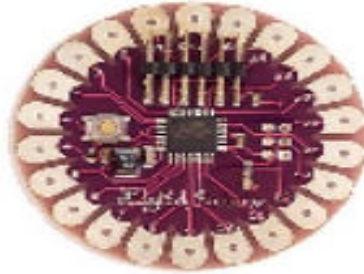
Microcontroller:	ATmega168
Operating Voltage:	3.3V or 5V (depending on model)
Input Voltage :	3.35 - 12 V or 5 - 12 V
Digital I/O Pins:	14 (of which 6 provide PWM output)
Analog Input Pins :	6
DC Current:	40 mA
Flash Memory:	16 KB
SRAM:	1 KB
EEPROM:	512 bytes
Clock Speed:	8 MHz or 16 MHz

# Arduino Mega



Microcontroller:	ATmega1280
Operating Voltage:	5V
Input Voltage:	6-20V
Digital I/O Pins:	54 (of which 15 provide PWM output)
Analog Input Pins:	16
DC Current:	40 mA
Flash Memory:	128 KB
SRAM:	8 KB
EEPROM:	4 KB
Clock Speed:	16 MHz

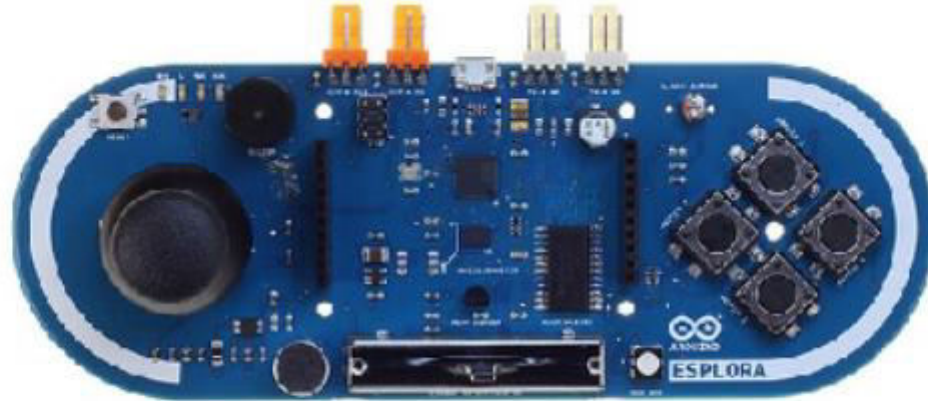
# Arduino Lilypad



Microcontroller:	ATmega168V or ATmega328V
Operating Voltage:	2.7-5.5 V
Input Voltage:	2.7-5.5 V
Digital I/O Pins:	14 (of which 6 provide PWM output)
Analog Input Pins:	6
DC Current:	40 mA
Flash Memory:	16 KB
SRAM:	1 KB
EEPROM:	512 bytes
Clock Speed:	8 MHz



# Arduino Esplora



Analog joystick

4 push-buttons

Linear potentiometer slider

Microphone

Light sensor

Temperature sensor

Three-axis accelerometer

Buzzer can produce square-waves.

RGB led bright LED

2 TinkerKit Inputs to connect the TinkerKit sensor modules with the 3-pin connectors.

2 TinkerKit Outputs to connect the TinkerKit actuator modules with the 3-pin connectors.

TFT display connector connector for an optional color LCD screen, SD card

Boards	Microcontroller	Operating Voltage/s (V)	Digital I/O Pins	PWM Enabled Pins	Analog I/O Pins	DC per I/O (mA)	Flash Memory (KB)	SRAM (KB)	EEPROM (KB)	Clock (MHz)	Length (mm)	Width (mm)	Cable	Native Network Support
Uno	ATmega328	5	14	6	6	20	32	2	1	16	68.6	53.4	USB A-B	None
Leonardo	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	micro-USB	None
Micro	ATmega32u4	5	20	7	12	40	32	2.5	1	16	48	18	micro-USB	None
Nano	ATmega328	5	22	6	8	40	32	2	0.51	16	45	18	mini-B USB	None
Mini	ATmega328	5	14		6	20	32	2	1	16	30	18	USB-Serial	None
Due	Atmel SAM3X8E ARM Cortex-M3 CPU	3.3	54	12	12	800	512	96	✕	84	102	53.3	micro-USB	None
Mega	ATmega2560	5	54	15	16	20	256	8	4	16	102	53.3	USB A-B	None
M0	Atmel SAMD21	3.3	20	12	6	7	256	32	✕	48	68.6	53.3	micro-USB	None
Yun Mini	ATmega32u4	3.3	20	7	12	40	32	2.5	1	400	71.1	23	micro-USB	Ethernet/Wifi
Uno Ethernet	ATmega328p	5	20	4	6	20	32	2	1	16	68.6	53.4	Ethernet	Ethernet
Tian	Atmel SAMD21	5	20	12	0	7	16000	64000	✕	560	68.5	53	micro-USB	Ethernet/Wifi
Mega ADK	ATmega2560	5	54	15	16	40	256	8	4	16	102	53.3	USB A-B	None
M0 Pro	Atmel SAMD21	3.3	20	12	6	7	256	32	✕	48	68.6	53.3	micro-USB	None
Industrial 101	ATmega32u4	5	7	2	4	40	16000	64000	1	400	51	42	micro-USB	Ethernet/Wifi
Uno Wifi	ATmega328	5	20	6	6	20	32	2	1	16	68.6	53.4	USB A-B	Wifi
Leonardo Ethernet	ATmega32u4	5	20	7	12	40	32	2.5	1	16	68.6	53.3	USB A-B	Ethernet
MKR1000	Atmel SAMD21	3.3	8	12	7	7	256	32	✕	48	64.6	25	micro-USB	Wifi

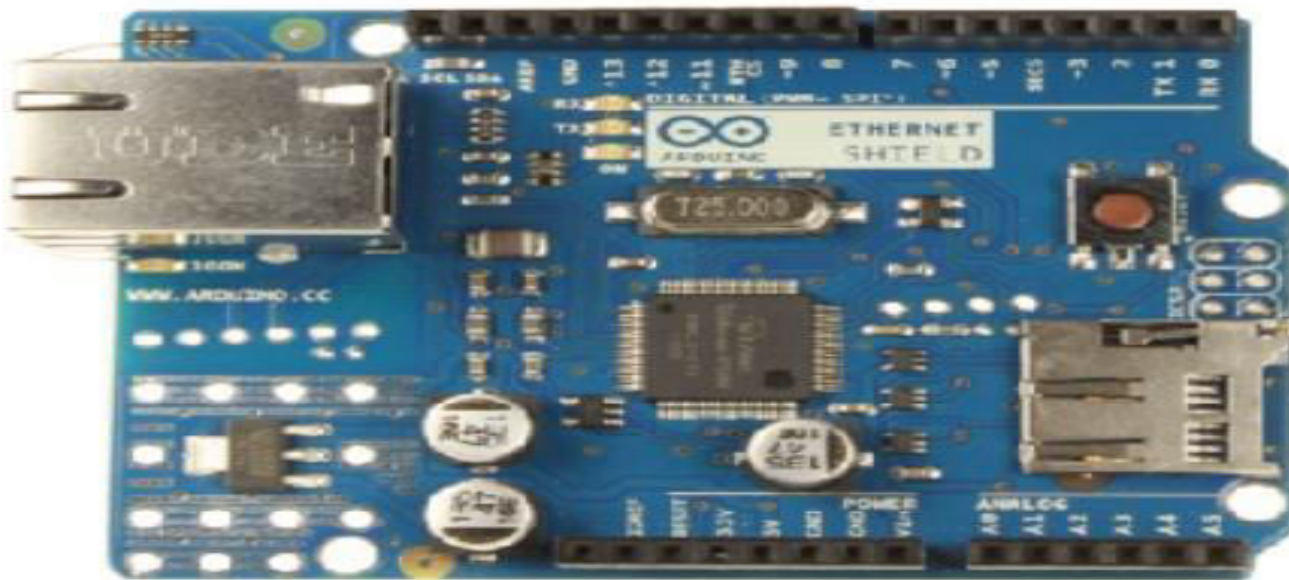
# Λίστα Επίσημων Μοντέλων

<http://arduino.cc/en/Main/Products?from=Main.Hardware>

ΕΠΕΚΤΑΣΕΙΣ

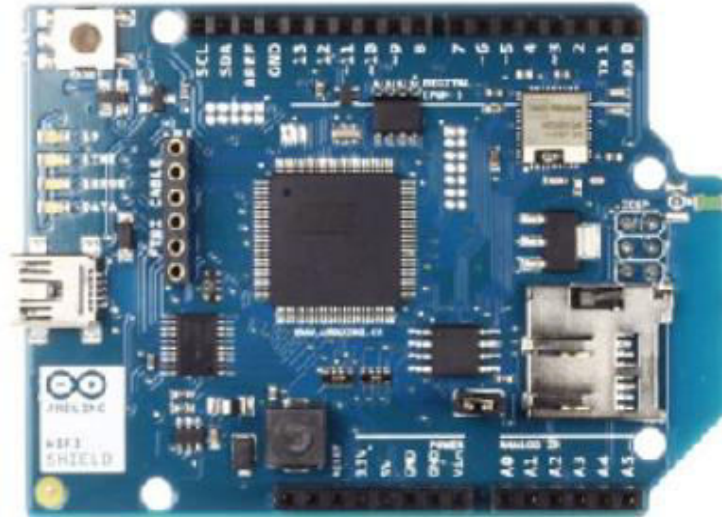


# Arduino Ethernet Shield



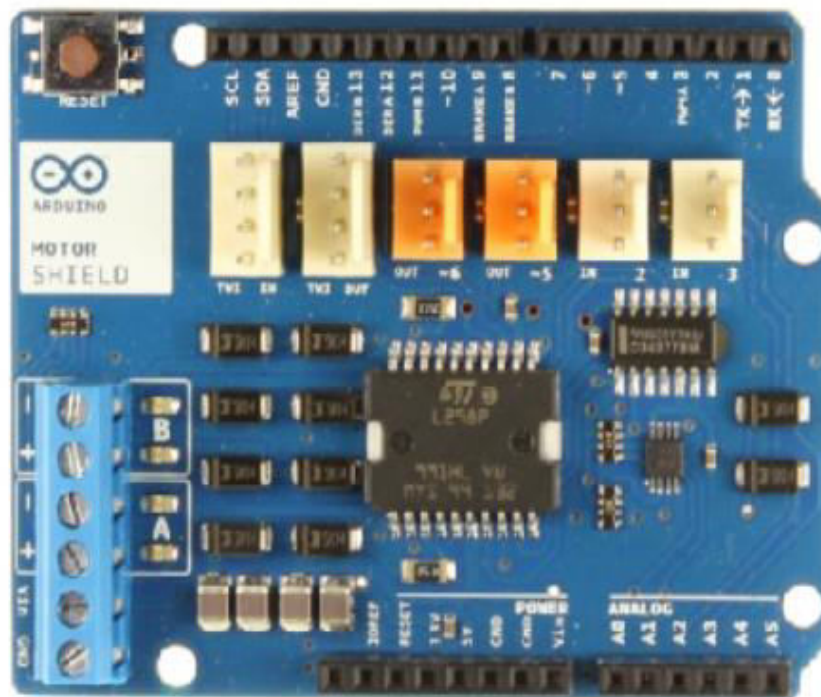
Επιτρέπει τη σύνδεση του Arduino με δίκτυο

# Arduino Wi-Fi Shield



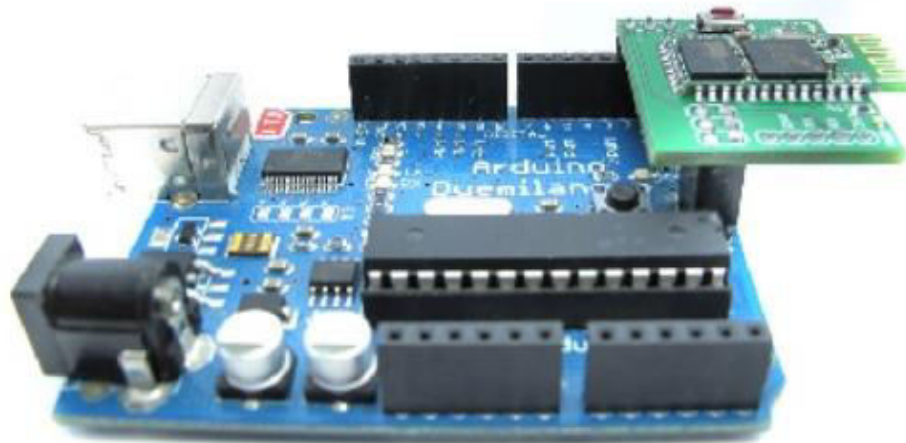
Επιτρέπει τη σύνδεση του Arduino με ασύρματα δίκτυα

# Arduino Motor Shield



Επιτρέπει τον έλεγχο 2 DC κινητήρων

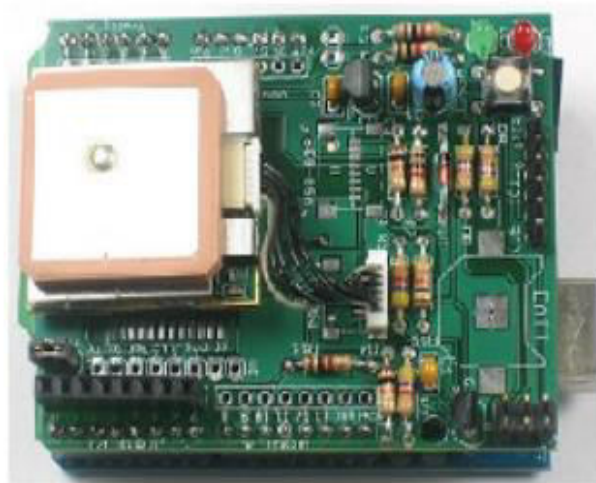
# Arduino Bluetooth Shield



Δυνατότητα επικοινωνίας μέσω bluetooth



# Arduino GPS Shield



Δυνατότητα γεωεντοπισμού

# Arduino LCD Shield



Εύκολη πρόσβαση σε οθόνη LCD

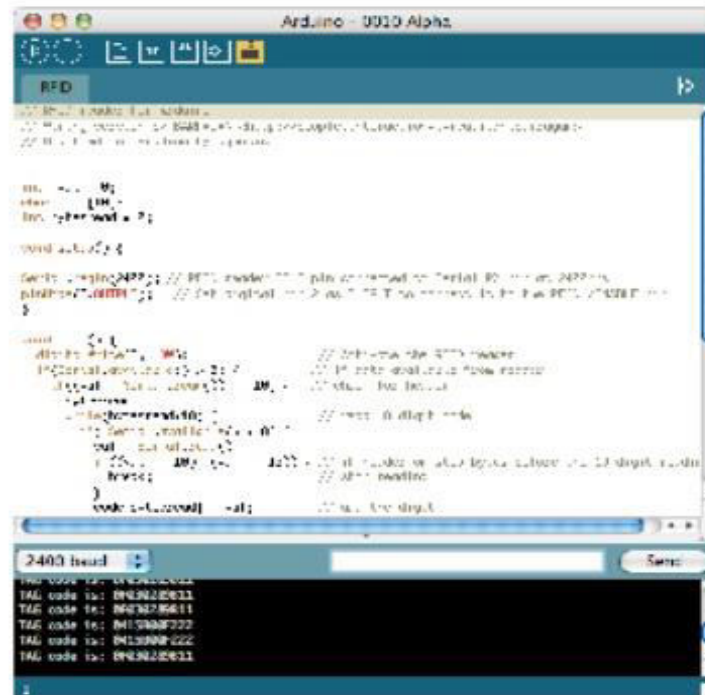
# Λίστα Επεκτάσεων

<http://shieldlist.org/>

Προγραμματιστικά  
περιβάλλοντα



# Arduino Software



Βασίζεται στην C  
Windows/Linux/Mac OS  
Ελεύθερο – Ανοικτού κώδικα

<http://arduino.cc/en/main/software>

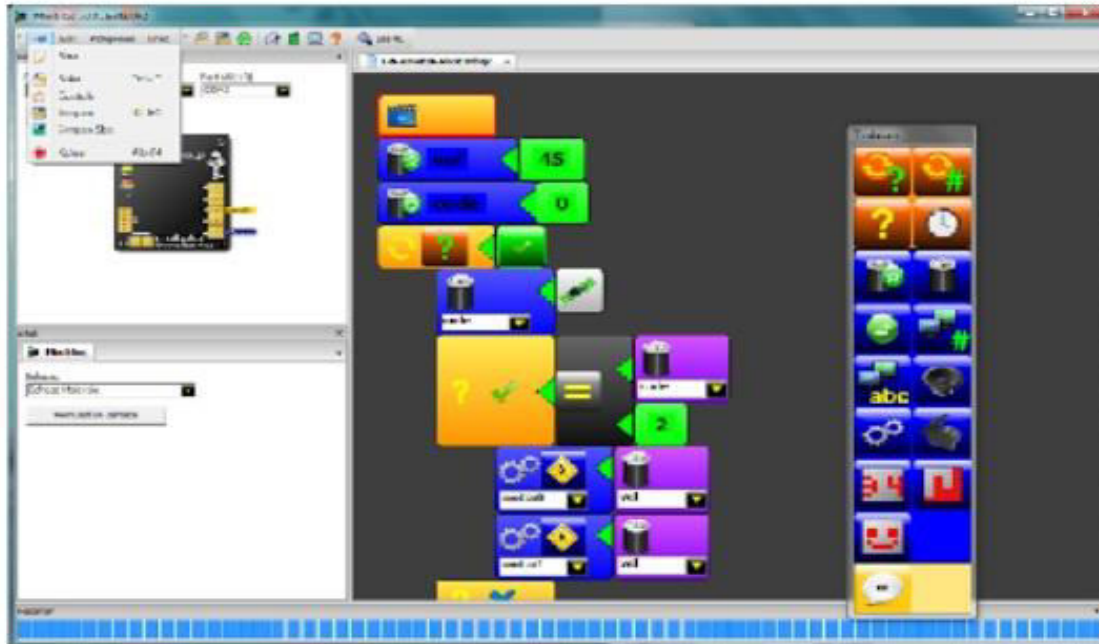
# Modkit



Θυμίζει scratch  
Windows/Linux/Mac OS  
Free online – pay to download

<http://www.modk.it>

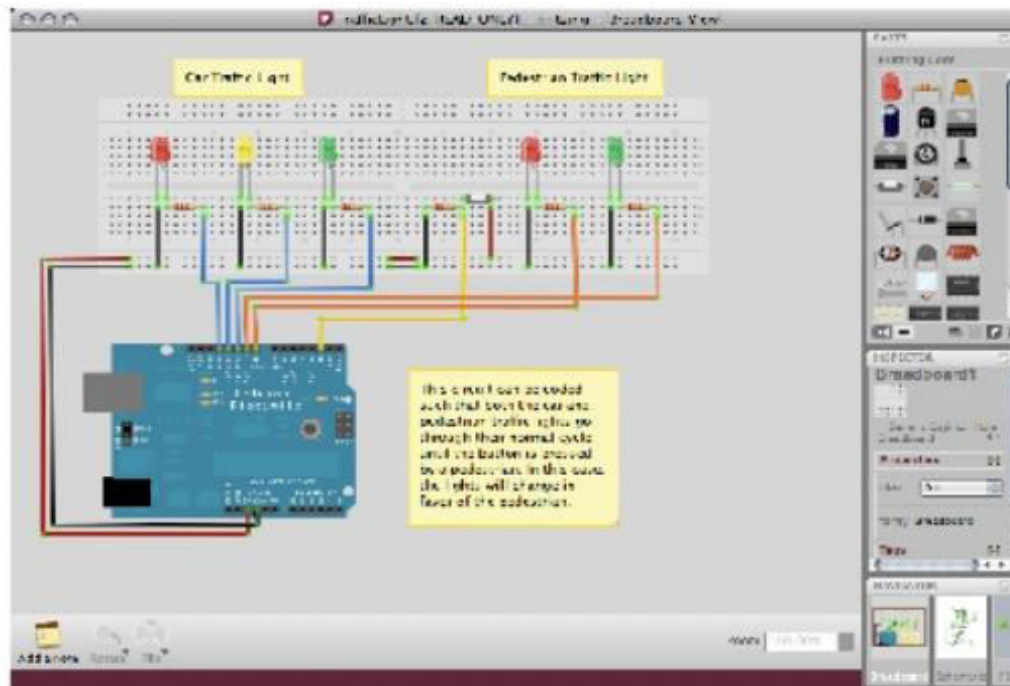
# MiniBlog



Γραφικό περιβάλλον  
Windows / Linux / OLPC / Mac OS

<http://blog.minibloq.org/>

# Fritzing Design Software



Σχεδίαση του κυκλώματος  
Εξαγωγή σχηματικού  
Εξαγωγή σχεδίου PCB  
Windows / Linux / Mac OS  
<http://fritzing.org/>

+

-





- Φθηνό σε σχέση με τους ανταγωνιστές
- Open source hardware
- Επεκτασιμότητα
- Πολυπληθής κοινότητα που το υποστηρίζει
- Πλήθος tutorials και βοηθημάτων



Χρειάζεται εξοικείωση με βασικές έννοιες ηλεκτρονικών κυκλωμάτων  
Έντονη hands-on εμπειρία

# Arduino Coding



**ARDUINO**

# Data Types and operators

**Integer:** used with integer variables with value between 2147483647 and -2147483647.

Ex: `int x=1200;`

**Character:** used with single character, represent value from -127 to 128.

Ex. `char c='r';`

**Long:** Long variables are extended size variables for number storage, and store 32 bits (4 bytes), from -2,147,483,648 to 2,147,483,647.

Ex. `long u=199203;`

**Floating-point** numbers can be as large as 3.4028235E+38 and as low as -3.4028235E+38. They are stored as 32 bits (4 bytes) of information.

Ex. `float num=1.291;`

[The same as **double** type]

# Statement and operators:

Statement represents a command, it ends with ;

Ex:

```
int x;
```

```
x=13;
```

Operators are symbols that used to indicate a specific function:

- Math operators: `+, -, *, /, %, ^`
- Logic operators: `==, !=, &&, ||`
- Comparison operators: `==, >, <, !=, <=, >=`

Syntax:

`;` Semicolon, `{ }` curly braces, `//` single line comment, `/* Multi-line comments */`

# Statement and operators:

## Compound Operators:

++ (increment)

-- (decrement)

+= (compound addition)

-= (compound subtraction)

\*= (compound multiplication)

/= (compound division)



# Control statements:

## If Conditioning:

```
if(condition)  
{  
  statements-1;  
  ...  
  Statement-N;  
}  
else if(condition2)  
{  
  Statements;  
}  
Else{statements;
```



# Control statements:

## Switch case:

```
switch (var) {  
    case 1:  
        //do something when var equals 1  
        break;  
    case 2:  
        //do something when var equals 2  
        break;  
    default:  
        // if nothing else matches, do the default  
        // default is optional  
}
```



# Loop statements:

## **Do... while:**

```
do
{
Statements;
}
while(condition);    // the statements are run at least once.
```

## **While:**

```
While(condition)
{statements;}
```

## **for**

```
for (int i=0; i <= val; i++){
    statements;
}
```

# Code structure:

`Void setup(){}`

Used to indicate the initial values of system on starting.

`Void loop(){}`

Contains the statements that will run whenever the system is powered after setup.



# Input and output:

## Led blinking example:

Used functions:

```
pinMode();  
digitalRead();  
digitalWrite();  
delay(time_ms);
```

other functions:

```
analogRead();  
analogWrite();//PWM.
```





# Programming Arduino

Here you can learn how to program Arduino and what each code means and do, from here:

<http://arduino.cc/en/Reference/HomePage>

## Language Reference

Arduino programs can be divided in three main parts: *structure*, *values* (variables and constants), and *functions*.

### Structure

- `setup()`
- `loop()`

Control Structures

### Variables

#### Constants

- `HIGH` | `LOW`
- `INPUT` | `OUTPUT` | `INPUT_PULLUP`

### Functions

#### Digital I/O

- `pinMode()`
- `digitalWrite()`

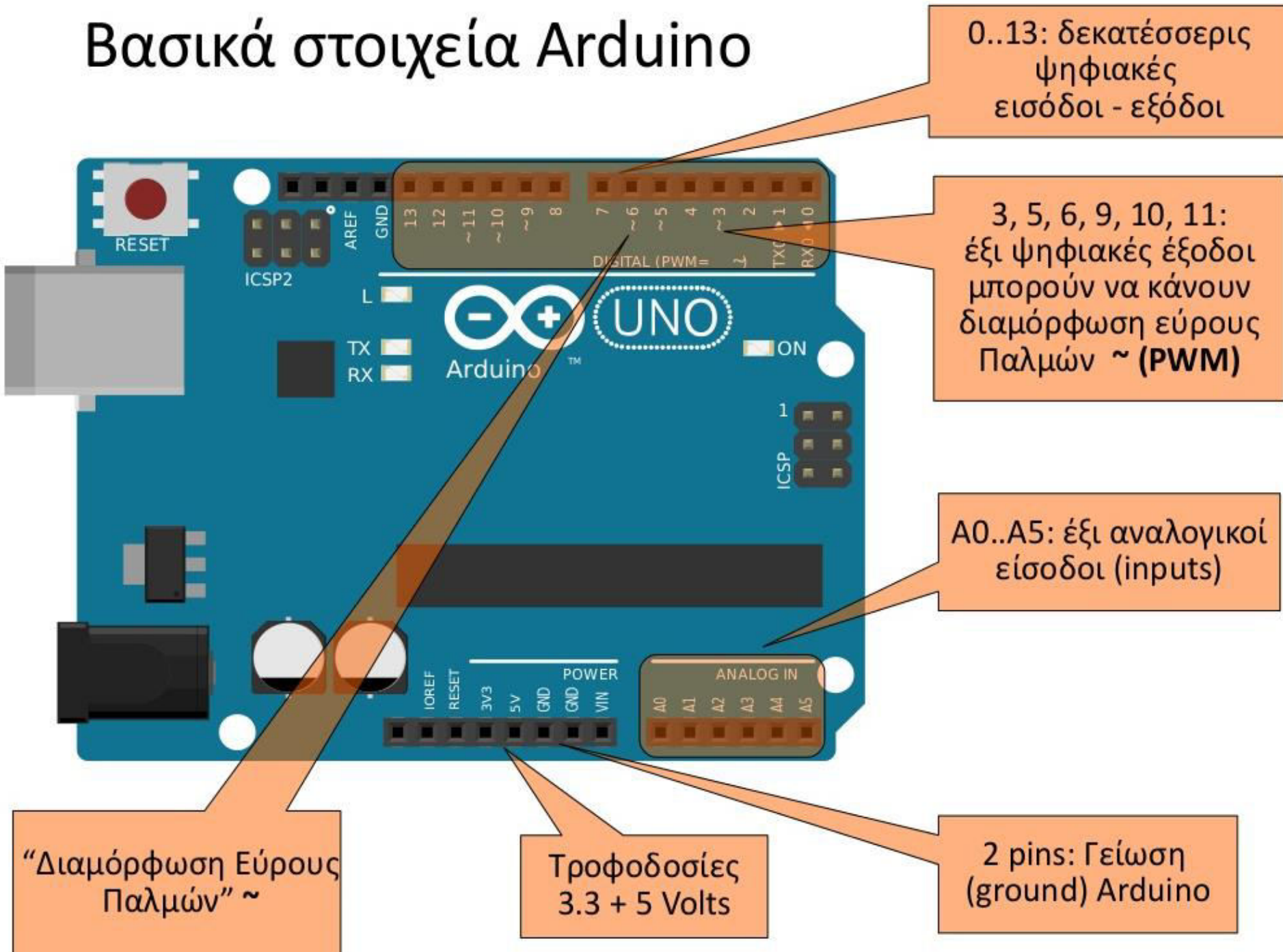
# Arduino Starter Workshop



# Example #1 – Blink a LED

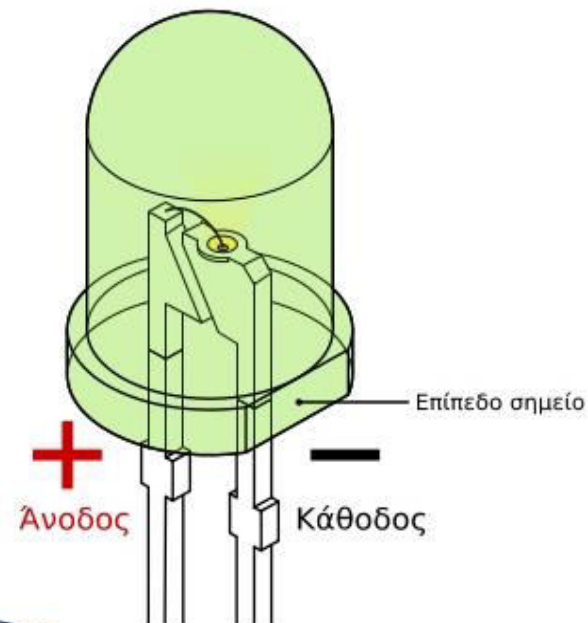
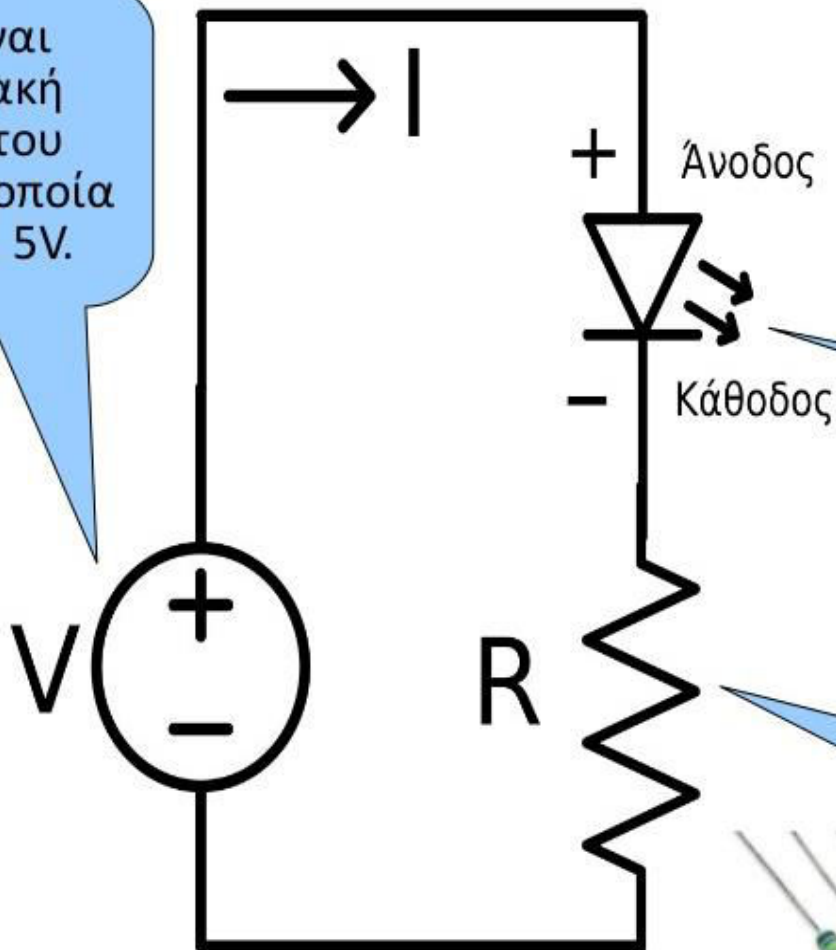
- Υλικά που θα χρειασθούν :
- Arduino Uno Board
- Breadboard
- Led diode
- Αντίσταση 220ohm  $\frac{1}{4}$  watt
- Καλώδια σύνδεσης
- Καλώδιο USB σύνδεσης arduino με τον Η/Υ
- Εγκατάσταση software IDE στον Η/Υ

# Βασικά στοιχεία Arduino



# Βασικό κύκλωμα LED

Εδώ είναι η ψηφιακή έξοδος του Arduino η οποία δίνει 0 ή 5V.



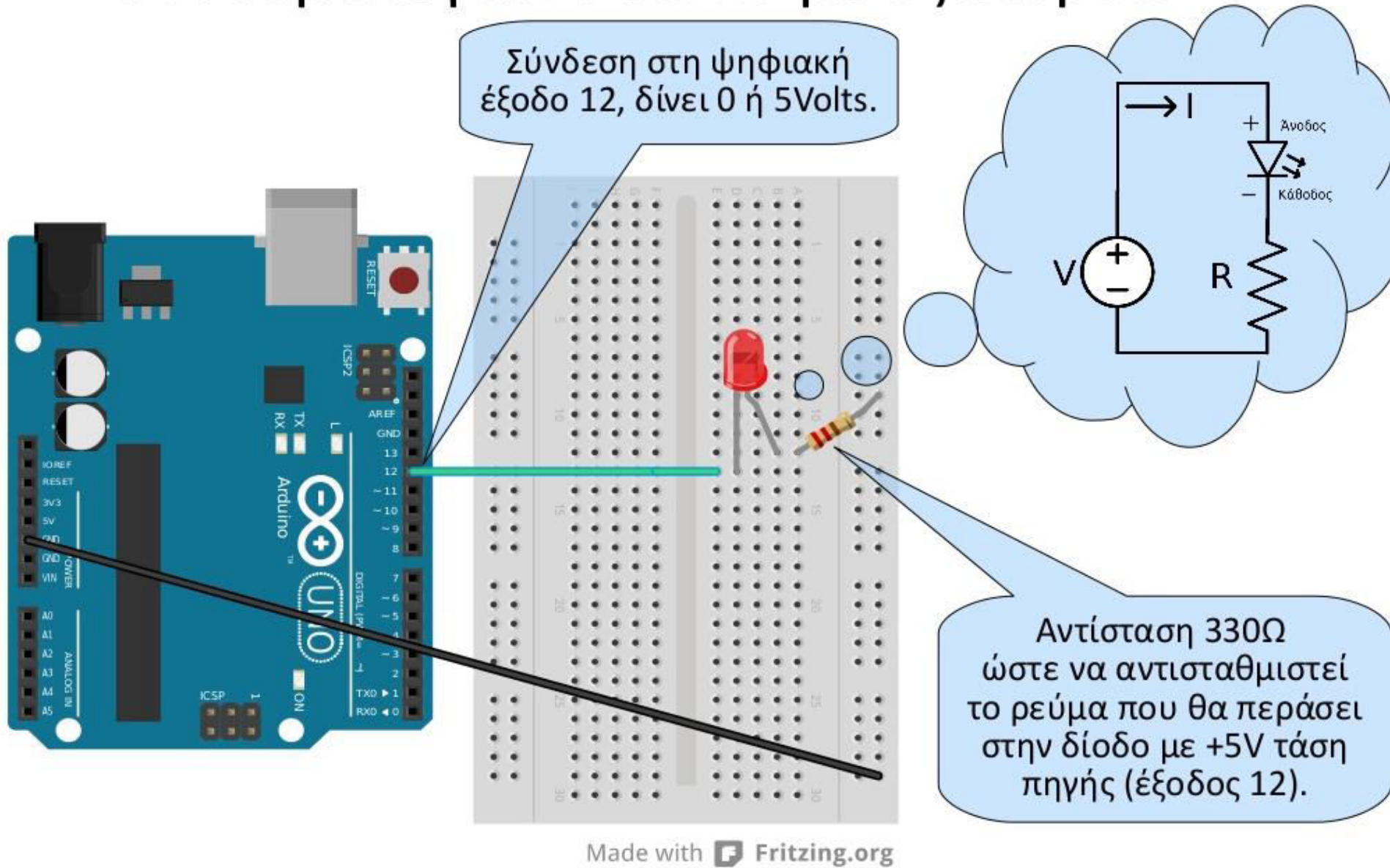
Δίοδος Εκπομπής Φωτός (LED)



Η ωμική αντίσταση  $R$  χρησιμοποιείται ώστε να μειώσει το ρεύμα που θα τροφοδοτήσει το LED ώστε να μην καεί.



# Συνδεσμολογία Arduino με εξωτερικό LED



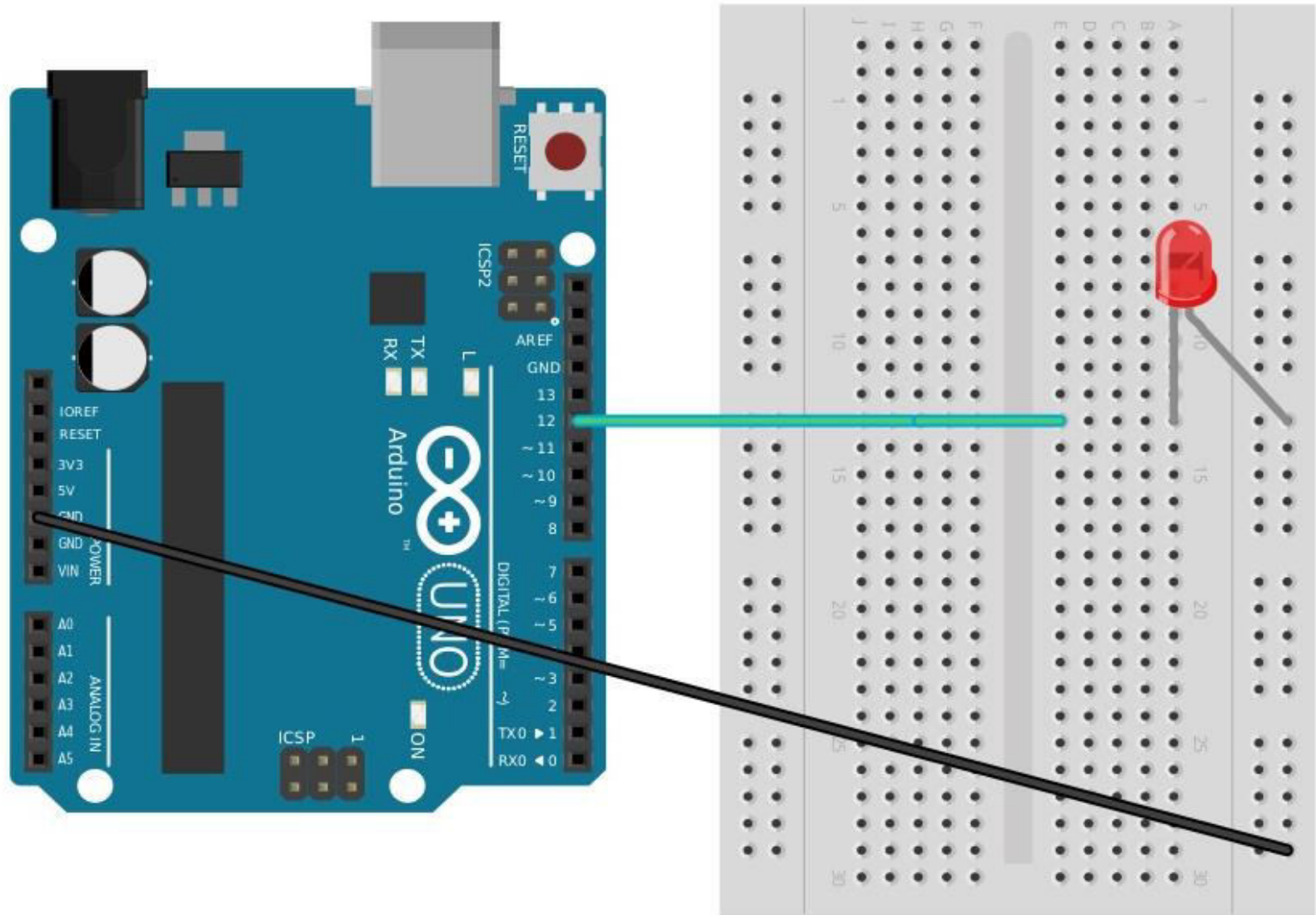
Στη παραπάνω συνδεσμολογία χρησιμοποιείται η ψηφιακή **έξοδος 12** (πράσινο καλώδιο) η οποία τροφοδοτεί με ρεύμα το κύκλωμα με το λαμπάκι LED.

# Προγραμματισμός LED κυκλώματος

```
/* Το πρόγραμμα αυτό αναβοσβήνει το LED που είναι συνδεδεμένο στην έξοδο 12
   κάθε 1 δευτερόλεπτο.
*/
void setup() {
  pinMode(12, OUTPUT); // ενεργοποίηση ψηφιακής εξόδου 12 (για έξοδο/output)
}
void loop() {
  digitalWrite(12, HIGH); // κάνε την έξοδο 12 υψηλή (δηλαδή στείλε +5 volts)
  delay(1000);           // περίμενε 1000ms = 1 δευτερόλεπτο
  digitalWrite(12, LOW); // κάνε την έξοδο 12 χαμηλή (δηλαδή στείλε 0 volts)
  delay(1000);           // περίμενε 1000ms = 1 δευτερόλεπτο
}
```

Ό,τι ξεκινάει από // ή βρίσκεται ανάμεσα σε /\* ... \*/ είναι σχόλια που τα αγνοεί ο μεταγλωττιστής του κώδικα.

# Τι λάθος υπάρχει παρακάτω;



Made with  Fritzing.org

Είναι σίγουρο ότι θα βραχυκυκλώσει το LED ή το Arduino;

# Arduino Workshop



**ARDUINO**

# Example #2 – Analog Read

- ⦿ Υλικά που θα χρειασθούν :
- ⦿ Arduino Uno Board
- ⦿ Breadboard
- ⦿ Μεταβλητή Αντίσταση 10K
- ⦿ Καλώδια σύνδεσης
- ⦿ Καλώδιο USB σύνδεσης arduino με τον Η/Υ
- ⦿ Εγκατάσταση software IDE στον Η/Υ



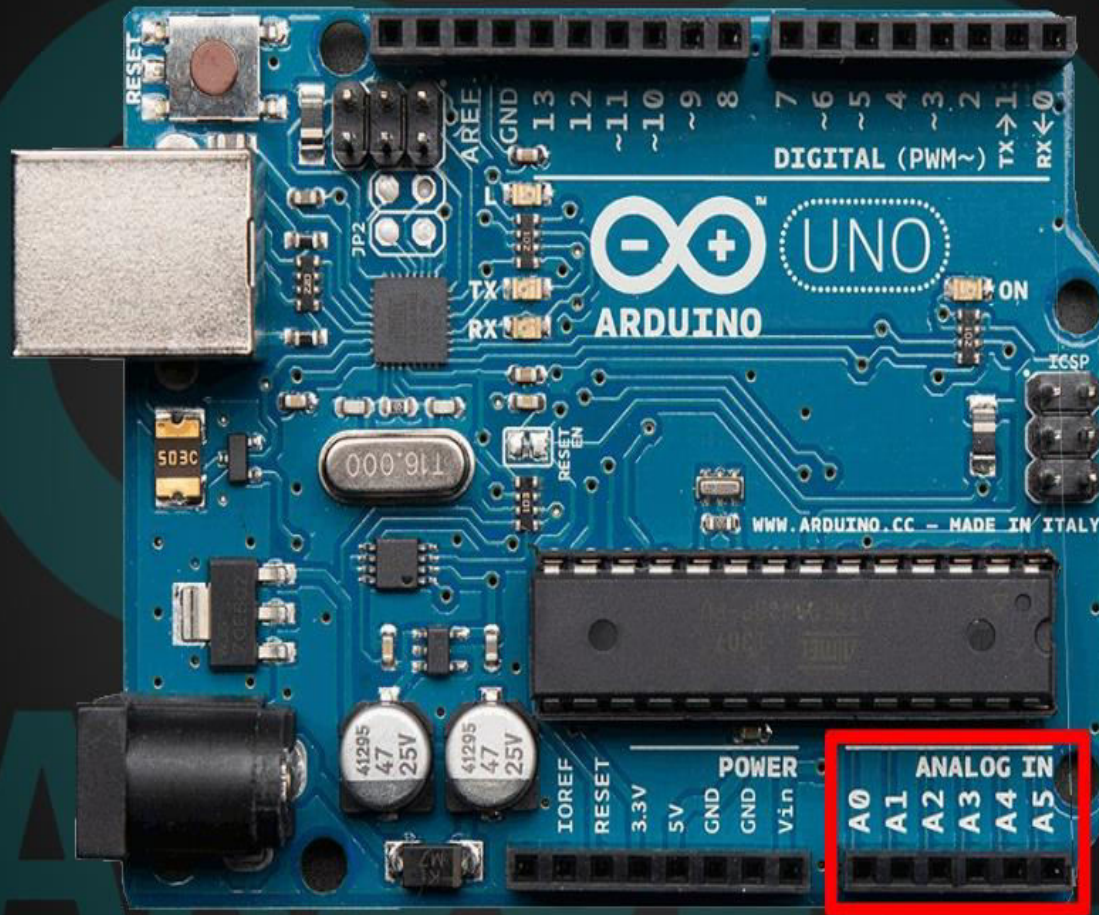
# Η συνάρτηση `analogRead()`

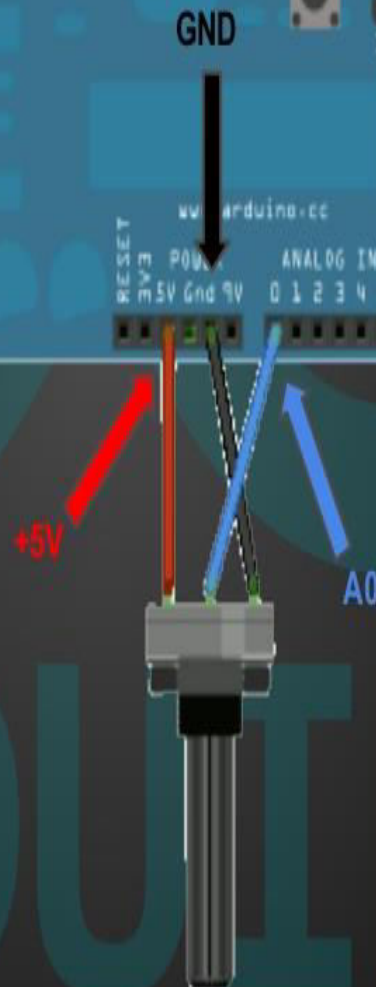
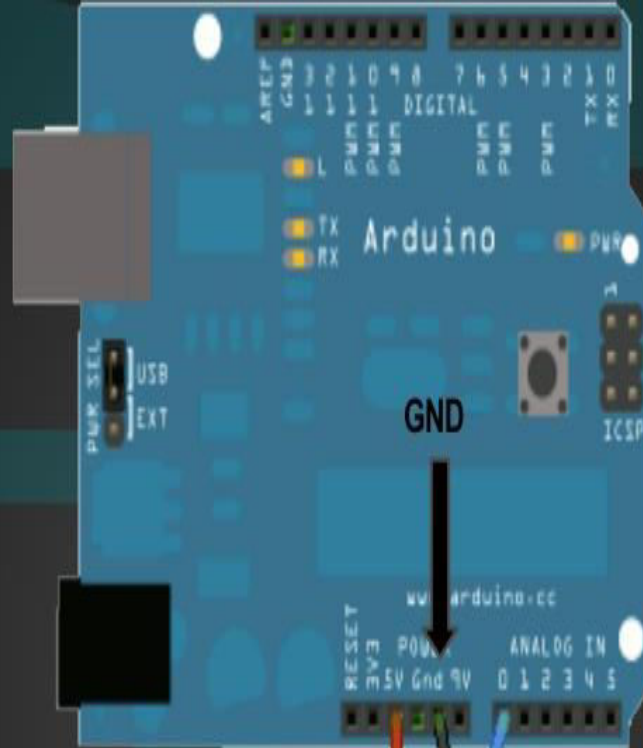
- Σύνταξη: `analogRead("Analog In")`
- Λειτουργία: Επιστρέφει μία **ακέραια** τιμή από το **0 έως 1023** που αντιπροσωπεύει την διαφορά δυναμικού στην θύρα που έχουμε επιλέξει. ( $1023 = 5V$ )

# ARDUINO



# Arduino Uno – Analog Input Pins





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

Εκκίνηση σειριακής επικοινωνίας  
με τον υπολογιστή.

```
void loop() {  
  int sensorValue = analogRead(A0);  
  Serial.println(sensorValue);  
  delay(1);  
}
```

Εδώ διαβάζουμε την τιμή στην  
θύρα **A0**. Άρα η sensorValue  
κυμαίνεται από 0 έως 1023!

Καθυστέρηση 1ms

# ARDUINO

# Από αριθμούς σε Volt

Με απλή μέθοδο των τριών:

Όταν η συνάρτηση `analogRead()` δίνει **1023**, δίνουμε στην θύρα **5.0 Volt**.

Όταν η συνάρτηση `analogRead()` δίνει **α**, δίνουμε στην θύρα **x (?) Volt**.

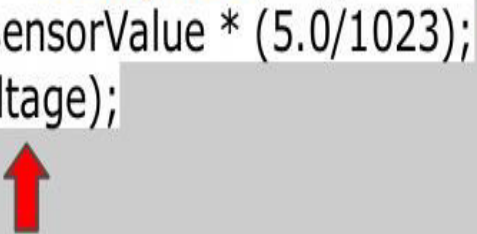
Άρα καταλήγουμε στον τύπο (σε γλώσσα C):

$$x = (5.0/1023)*\alpha$$

# ARDUINO

Το αποτέλεσμα της προηγούμενης παράστασης όμως, είναι δεκαδικός αριθμός.  
Άρα θα χρησιμοποιήσουμε τον τύπο μεταβλητής **float**.

```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  float voltage = sensorValue * (5.0/1023);  
  Serial.println(voltage);  
  delay(1);  
}
```

Two red arrows are present. One arrow points from the right towards the 'float' keyword in the line 'float voltage = sensorValue \* (5.0/1023);'. The other arrow points upwards from below towards the '1' in the 'delay(1);' line.



# Η συνάρτηση **map()**

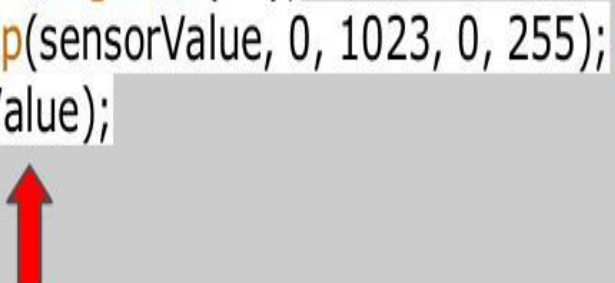
- Σύνταξη:  
`map("value", "fromLow", "fromHigh", "toLow", "toHigh")`
- Λειτουργία: Δέχεται στο πεδίο **value** μια μεταβλητή, στο πεδίο **fromLow** το **κάτω άκρο** των τιμών της, στο **fromHigh** το **άνω άκρο** των τιμών της και την μετατρέπει ώστε να περιορίζεται μεταξύ δύο νέων άνω και κάτω άκρων που δίνονται στα πεδία **toLow** και **toHigh** αντίστοιχα. Οι ενδιάμεσες τιμές της, πάνε αναλογικά.



# Παράδειγμα - Εφαρμογή

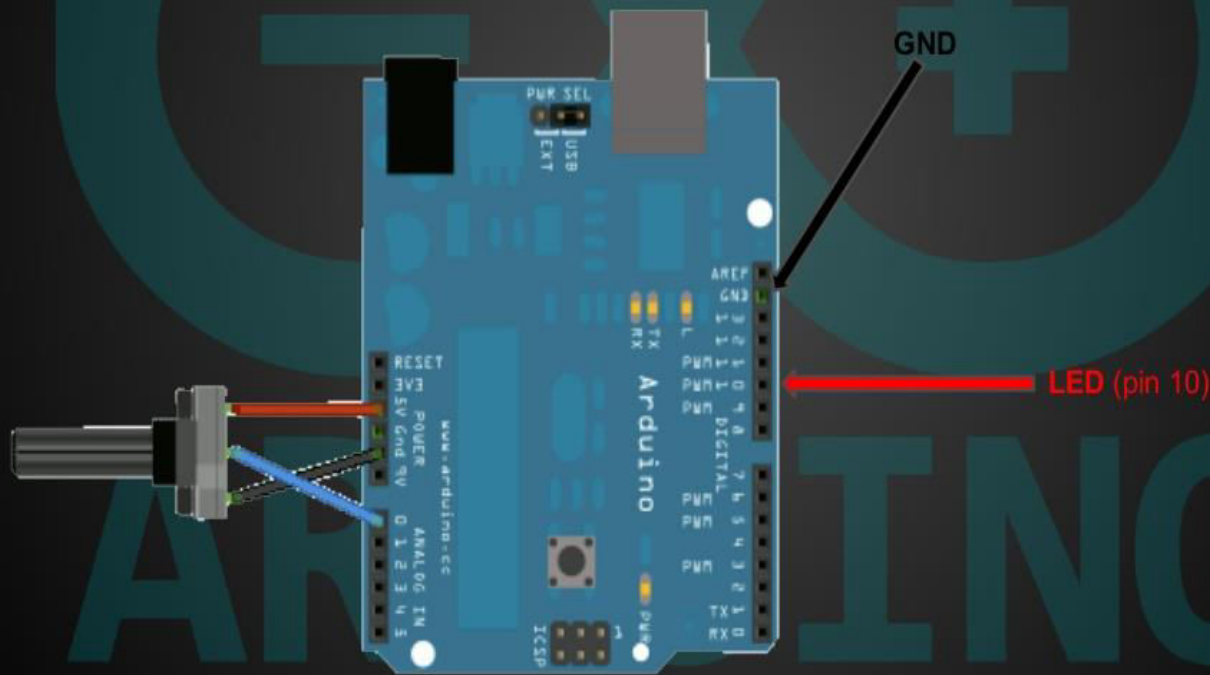
```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int sensorValue = analogRead(A0);  
  int newValue = map(sensorValue, 0, 1023, 0, 255);  
  Serial.println(newValue);  
  delay(1);  
}
```

Two red arrows are present. One arrow points from the right towards the 'map' function in the 'loop' function. The other arrow points upwards towards the '1' in the 'delay(1)' function.

# Ρύθμιση φωτεινότητας του Led με την χρήση μεταβλητής αντίστασης

Fade A Led: vol. 2



Θα χρησιμοποιήσουμε την συνάρτηση `analogWrite` για να ανάβουμε το LED σε διαφορετικές εντάσεις. Όμως **δεν** μπορούμε να βάλουμε κατευθείαν την τιμή που λαμβάνουμε από το ποτενσιόμετρο στην συνάρτηση `analogWrite` διότι ενώ η τελευταία δέχεται τιμές από **0 έως 255**, η συνάρτηση `analogRead` δίνει από **0 έως 1023**.

Εδώ χρειαζόμαστε την συνάρτηση `map()`.

```
void setup() {  
  pinMode(10, OUTPUT);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  int ledValue = map(sensorValue, 0, 1023, 0, 255);  
  analogWrite(10, ledValue);  
  delay(1);  
}
```

“Γράφουμε” στο LED την τιμή που μας δίνει η `ledValue`.

Σας Ευχαριστώ για τον Χρόνο σας.....

Βλάχος Ιωάννης

Ηλ/γος Μηχανικός (T.E.I) , MSc in Informatics

Ionian University

8/3/2019