

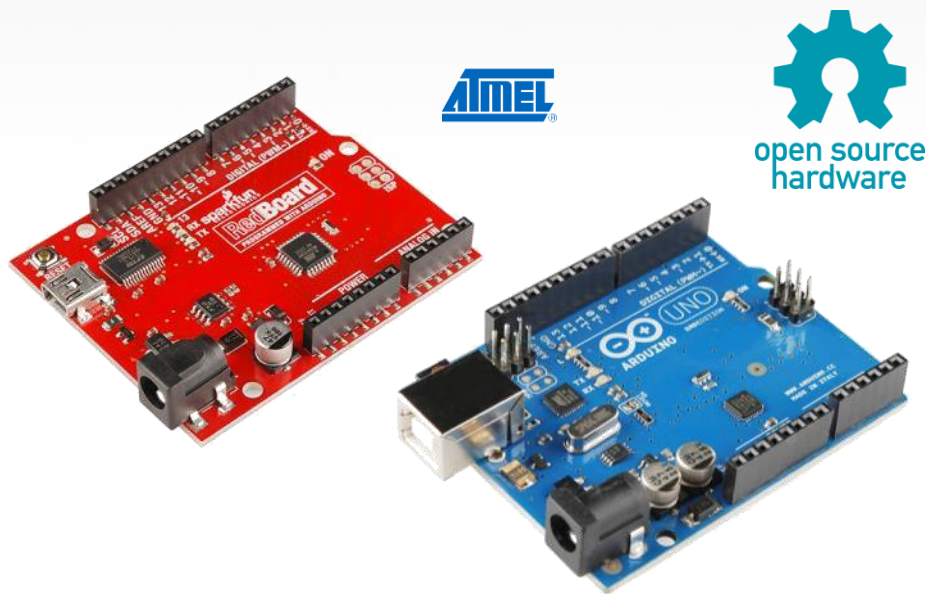
Εφαρμογές Arduino

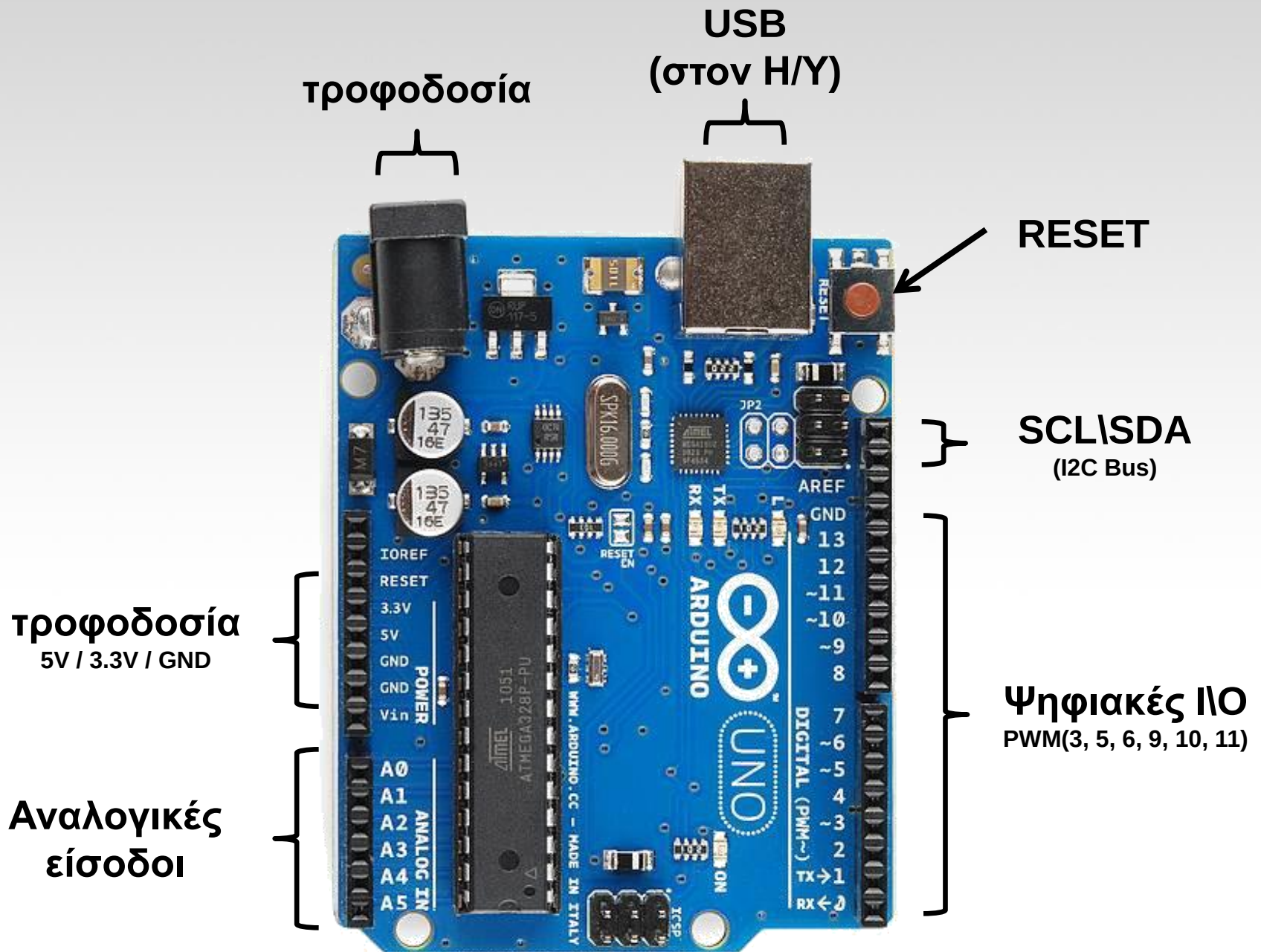
Σεμινάριο Ηλεκτρονικού Τομέα



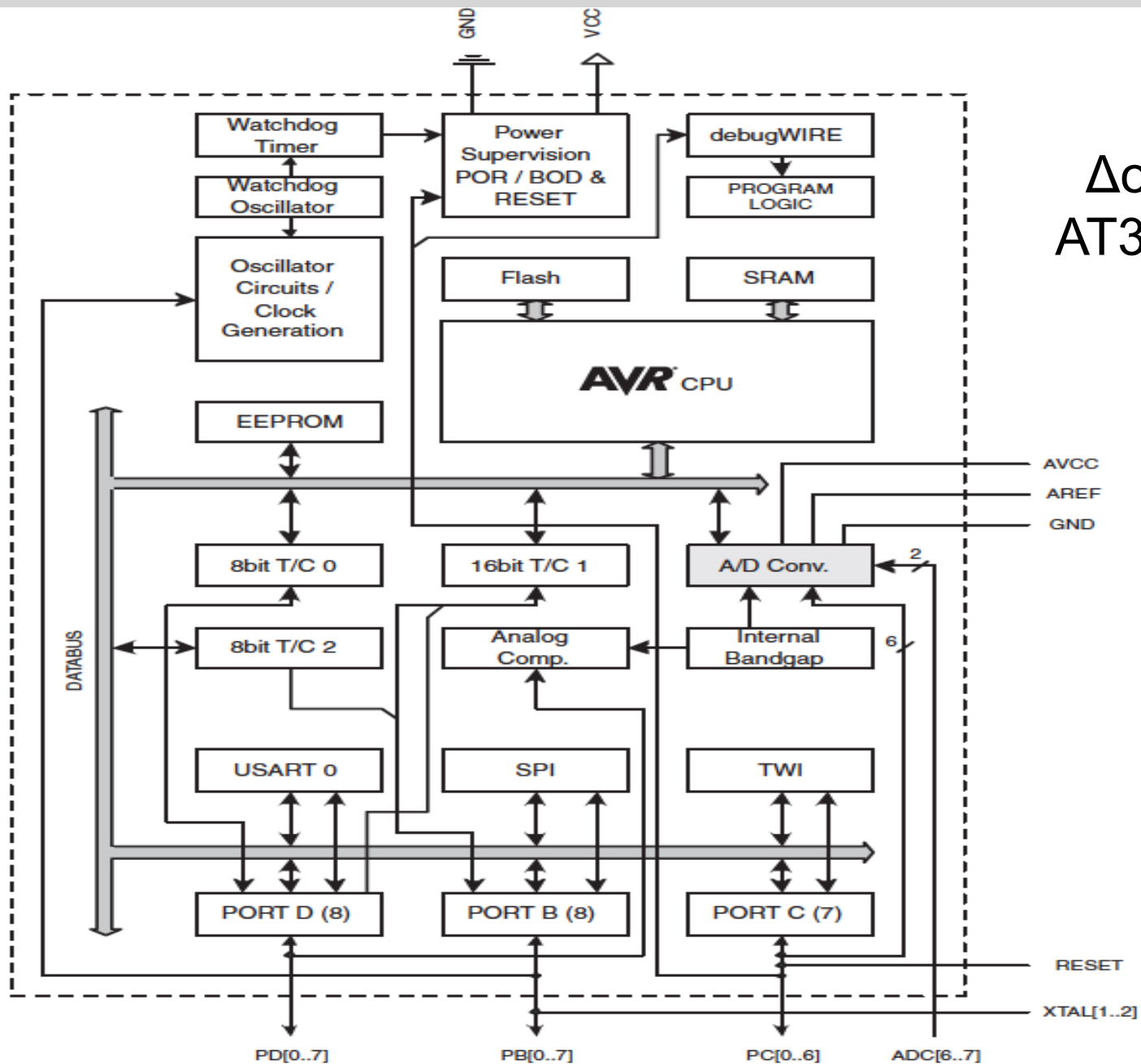
Πλακέτα Arduino

- Το 2005 οι Massimo Banzi και David Cueartielles στο Ivrea Δημιουργούν την υπολογιστική πλατφόρμα Arduino.
- Το Arduino είναι βασισμένο σε μια απλή μητρική πλακέτα ανοικτού κώδικα, με ενσωματωμένο μικροελεγκτή και εισόδους-εξόδους, και η οποία μπορεί να προγραμματιστεί με τη γλώσσα wiring





Δομή AT328P



Ακροδέκτες ATmega328P

Arduino function

reset

digital pin 0 (RX)

digital pin 1 (TX)

digital pin 2

digital pin 3 (PWM)

digital pin 4

VCC

GND

crystal

crystal

digital pin 5 (PWM)

digital pin 6 (PWM)

digital pin 7

digital pin 8

(PCINT14/RESET) PC6

(PCINT16/RXD) PD0

(PCINT17/TXD) PD1

(PCINT18/INT0) PD2

(PCINT19/OC2B/INT1) PD3

(PCINT20/XCK/T0) PD4

VCC

GND

(PCINT6/XTAL1/TOSC1) PB6

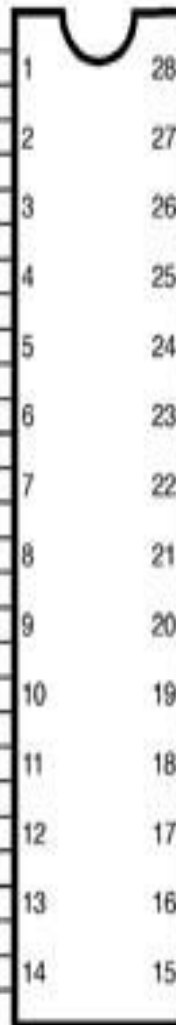
(PCINT7/XTAL2/TOSC2) PB7

(PCINT21/OC0B/T1) PD5

(PCINT22/OC0A/AIN0) PD6

(PCINT23/AIN1) PD7

(PCINT0/CLKO/ICP1) PB0



PC5 (ADC5/SCL/PCINT13)

PC4 (ADC4/SDA/PCINT12)

PC3 (ADC3/PCINT11)

PC2 (ADC2/PCINT10)

PC1 (ADC1/PCINT9)

PC0 (ADC0/PCINT8)

GND

AREF

AVCC

PB5 (SCK/PCINT5)

PB4 (MISO/PCINT4)

PB3 (MOSI/OC2A/PCINT3)

PB2 (SS/OC1B/PCINT2)

PB1 (OC1A/PCINT1)

Arduino function

analog input 5

analog input 4

analog input 3

analog input 2

analog input 1

analog input 0

GND

analog reference

VCC

digital pin 13

digital pin 12

digital pin 11(PWM)

digital pin 10 (PWM)

digital pin 9 (PWM)

ΜΝΗΜΕΣ ATMEGA328

Flash (32K) (15-bit addresses) τοποθετείται κάθε φορά το πρόγραμμα που πρόκειται να εκτελεστεί καθώς και ο φορτωτής εκκίνησης που διευκολύνει την διαδικασία του προγραμματισμού της πλατφόρμας. (μη πτητική)

SRAM (2K) χρησιμοποιείται για την προσωρινή αποθήκευση των στατικών και των μεταβλητών δεδομένων του προγράμματος που εκτελείται. Χάνει τα δεδομένα της όταν η παροχή ρεύματος στο Arduino σταματήσει ή αν γίνει reset. (πτητική)

EEPROM (1K) αποθηκεύονται οι τιμές των μεταβλητών όταν η πλατφόρμα σβήσει (OFF). Χρησιμοποιείται για την αποθήκευση ρυθμίσεων και άλλων παραμέτρων ανάμεσα στα Reset του Arduino. (μη πτητική)

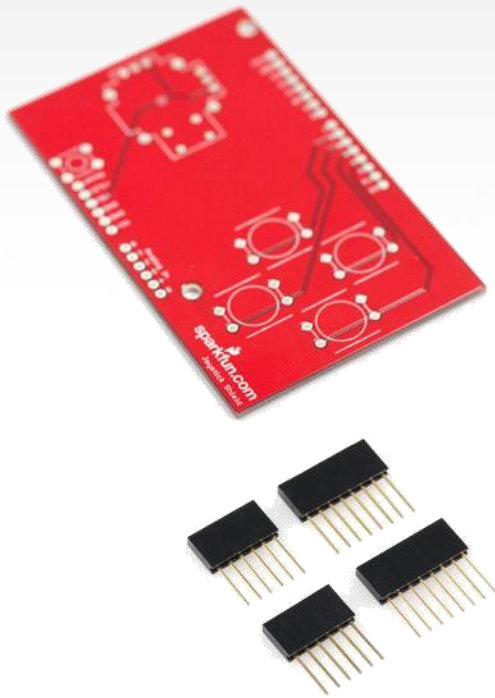
Arduino Shields

- Τα shield είναι ολοκληρωμένες πλακέτες που είναι σχεδιασμένες ώστε να κουμπώνουν πάνω στο Arduino προεκτείνοντας την λειτουργικότητά του.



Arduino Shields

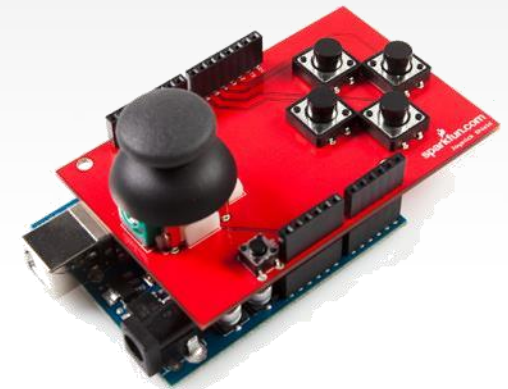
PCB



Built Shield



Inserted Shield

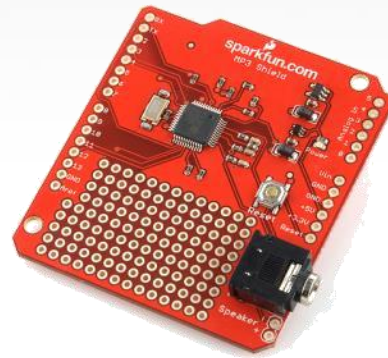


Arduino Shields

Micro SD

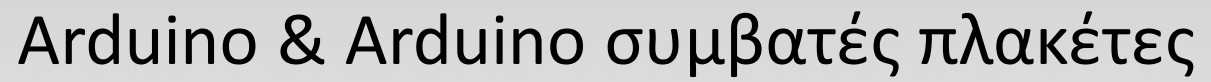


MP3 Trigger

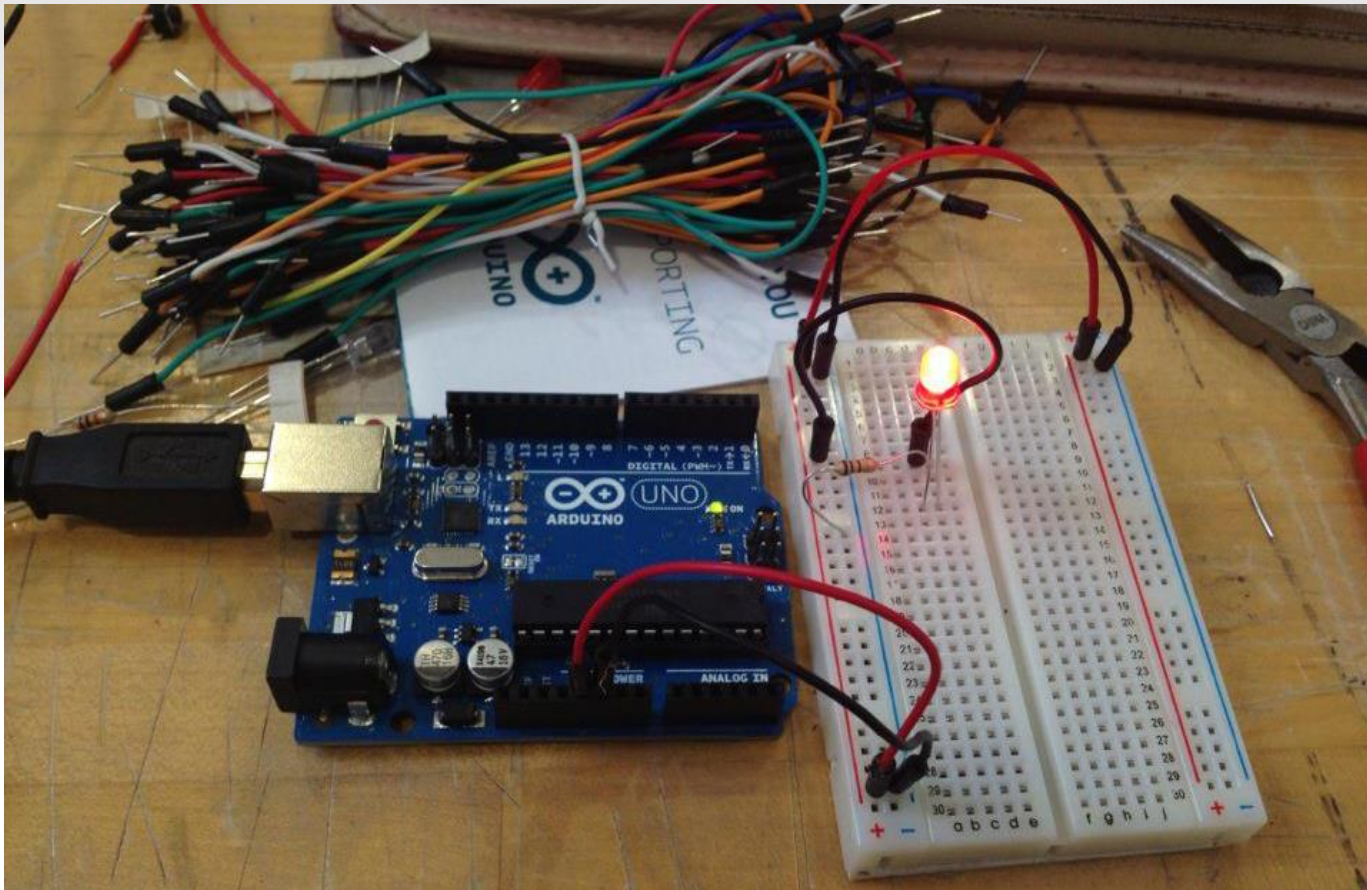


LCD



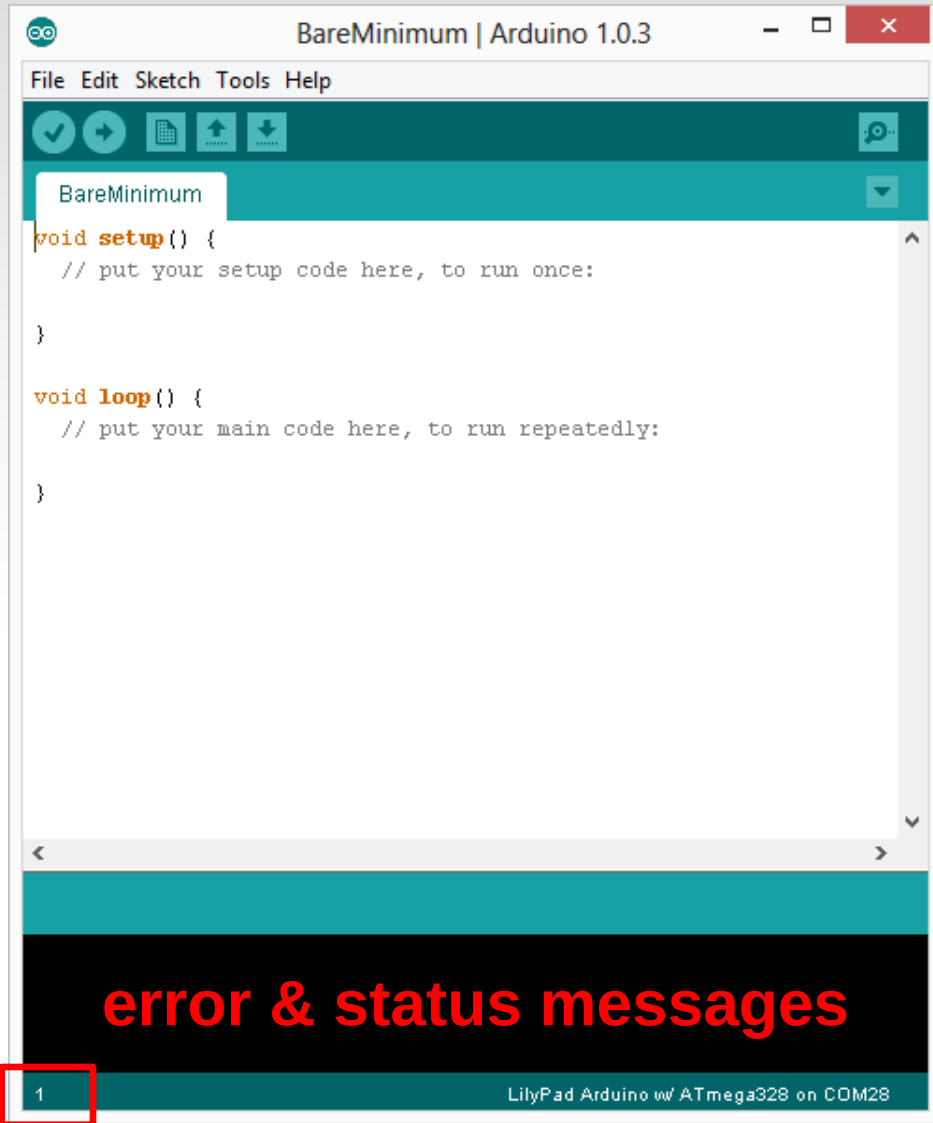


Ας αρχίσουμε να χρησιμοποιούμε
λοιπόν τον δικό μας Arduino!!!



Arduino

Ολοκληρωμένο Περιβάλλον Ανάπτυξης (IDE)



δυσ ειδικές συναρτήσεις
που είναι μέρος του κάθε
sketch του Arduino

//δήλωση μεταβλητών

```
void setup ()
```

```
{
```

```
//αρχικοποιήσεις
```

```
}
```

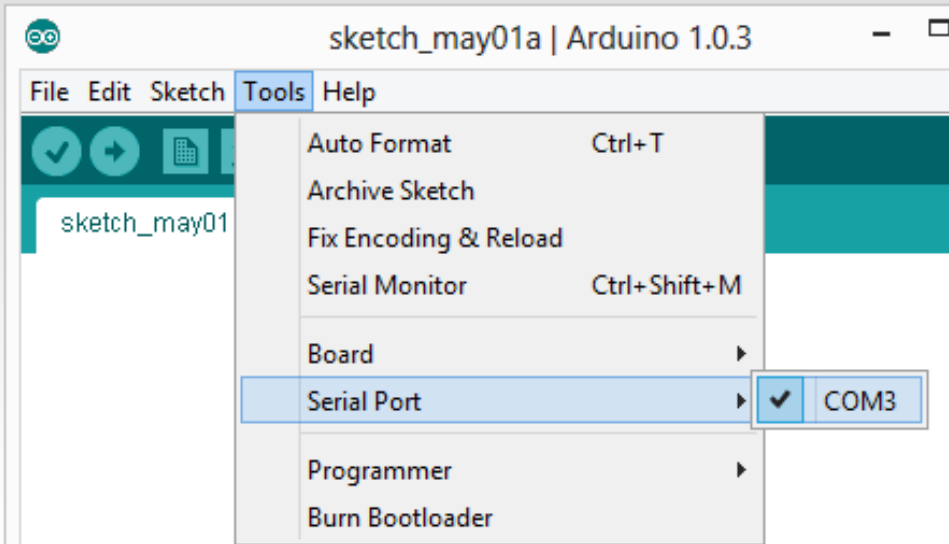
```
void loop ()
```

```
{
```

```
//Κώδικας
```

```
}
```

Settings: Tools → Serial Port

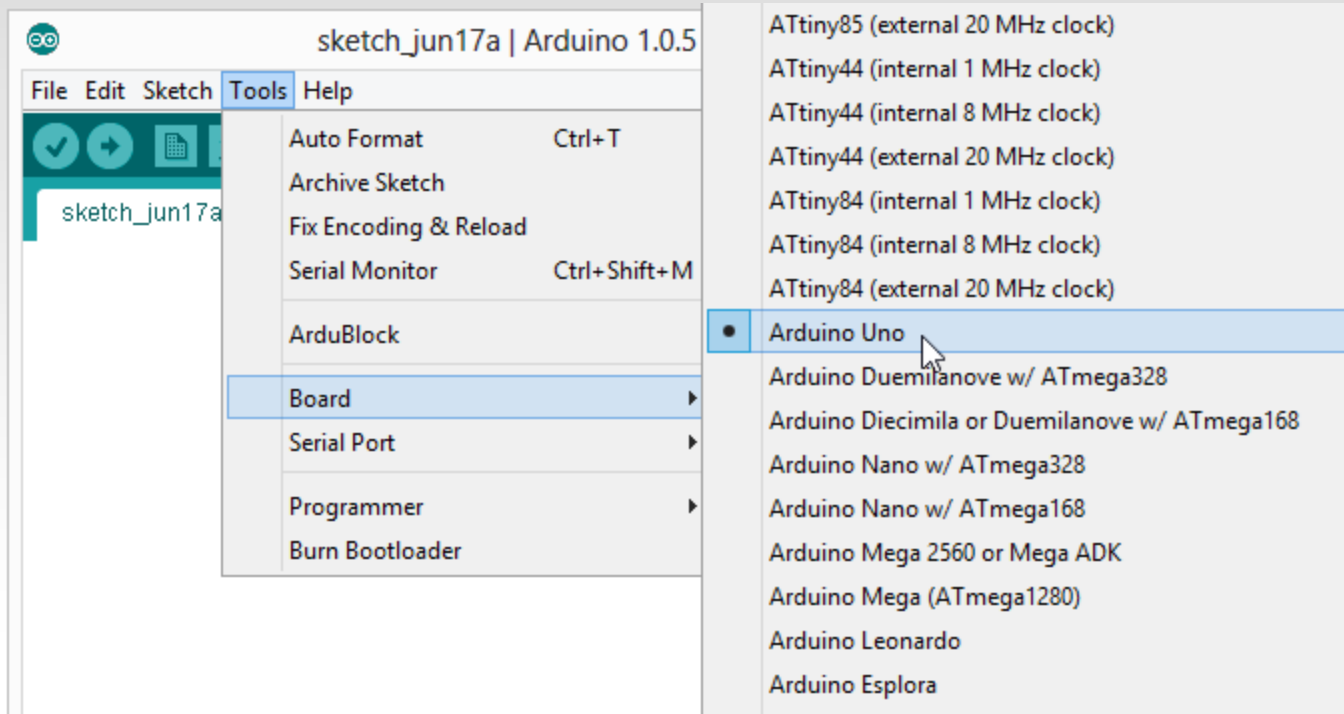


- Ο υπολογιστής επικοινωνεί με το Arduino microcontroller με την Serial port → COM3 μέσω ενός υποδοχέα USB.

- Ελέγξτε για να σιγουρευτείτε ότι έχουν εγκατασταθεί οι drivers.

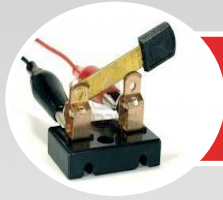
Στο δικό σας Arduino η θύρα COM μπορεί φυσικά να είναι διαφορετική

Επιλογή: Tools → Board



- Επιλέγουμε την πλακέτα με το μενού :Tools→Board→Arduino Uno

6 ΣΗΜΑΝΤΙΚΕΣ ΕΝΝΟΙΕΣ



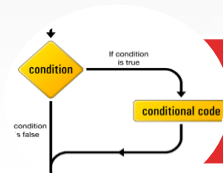
digitalWrite()



analogWrite()



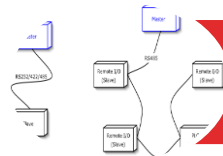
digitalRead()



if() statements / Boolean



analogRead()



Serial communication

Ψηφιακές και Αναλογικές I/O

Ψηφιακά pins:

- Pins 0 – 7: PORT D [0:7]
- Pins 8 – 13: PORT B [0:5]
- Pins 14 – 19: PORT C [0:5] (Arduino αναλογικά pins 0 – 5)
- Τα pins 0 και 1 είναι RX και TX για serial επικοινωνία
- Στο ψηφιακό pin 13 είναι συνδεδεμένο ένα LED στην βασική πλακέτα

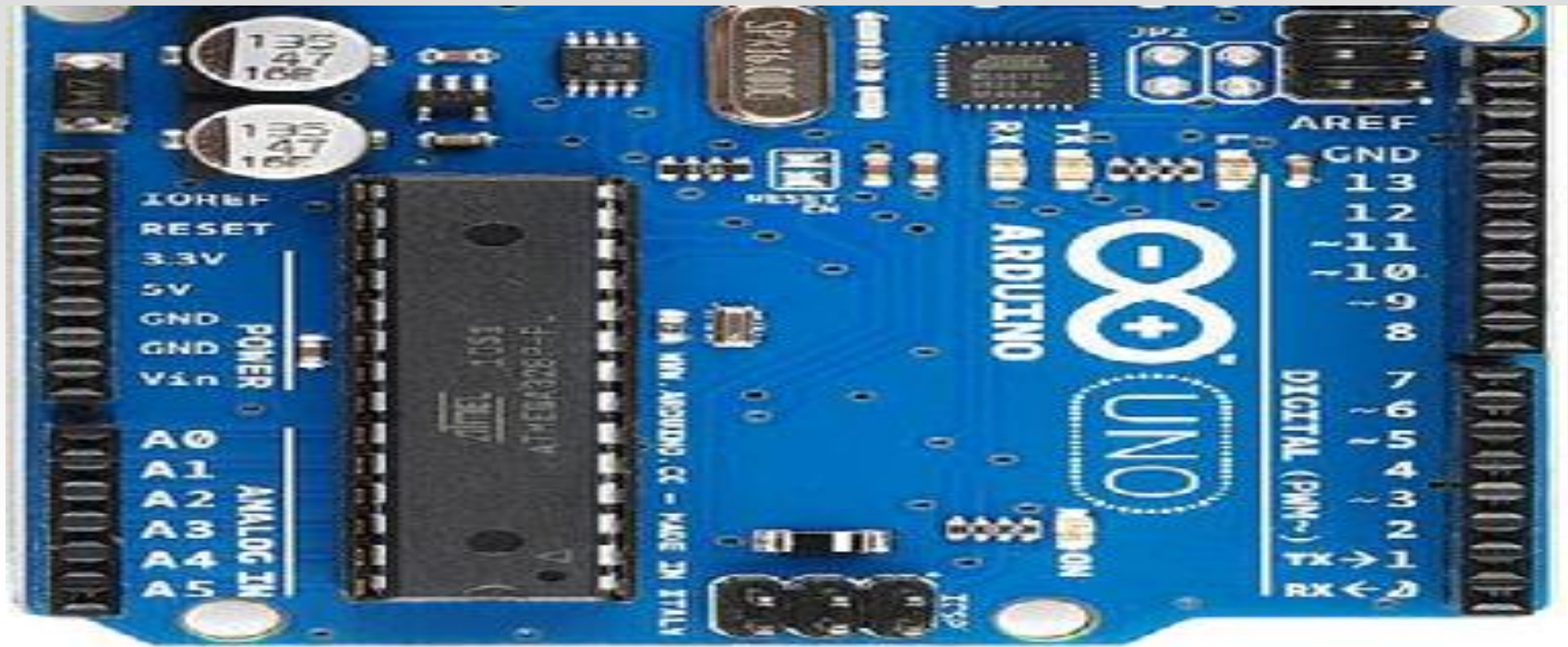


Ψηφιακές και Αναλογικές I/O

Ψηφιακά Pin I/O – εντολές, λειτουργίες

- *pinMode(pin, mode)*
- Θέτει το pin σε κατάσταση INPUT ή OUTPUT
- (ορίζει μια επαφή ως είσοδο ή έξοδο)
- *digitalWrite(pin, value)*
- Θέτει το pin value σε LOW ή HIGH (0 ή 1)
- (γράφει σε μία ψηφιακή επαφή εξόδου)
- *int value = digitalRead(pin)*
- Επιστρέφει την τιμή του pin value (0 ή 1)

Arduino Αναλογικές είσοδοι/έξοδοι (I/O)



- Αναλογικές είσοδοι pins: 0 – 5
- Αναλογικές έξοδοι pins: 3, 5, 6, 9, 10, 11 (ψηφιακά pins)

Arduino Αναλογικές είσοδοι/έξοδοι (I/O)

- **Αναλογικές είσοδοι** – εντολές, λειτουργίες
- `int val = analogRead(pin)`
- Μετατρέπει την τάση 0 – 5V σε έναν 10-bit αριθμό (0 – 1023)
- **Analog output**
- `analogWrite(pin, value)`
- value είναι 0 – 255
- Δημιουργεί PWM έξοδο στα ψηφιακά pin (3, 5, 6, 9, 10, 11)



Let's get to coding...

- Project #1 – Blink

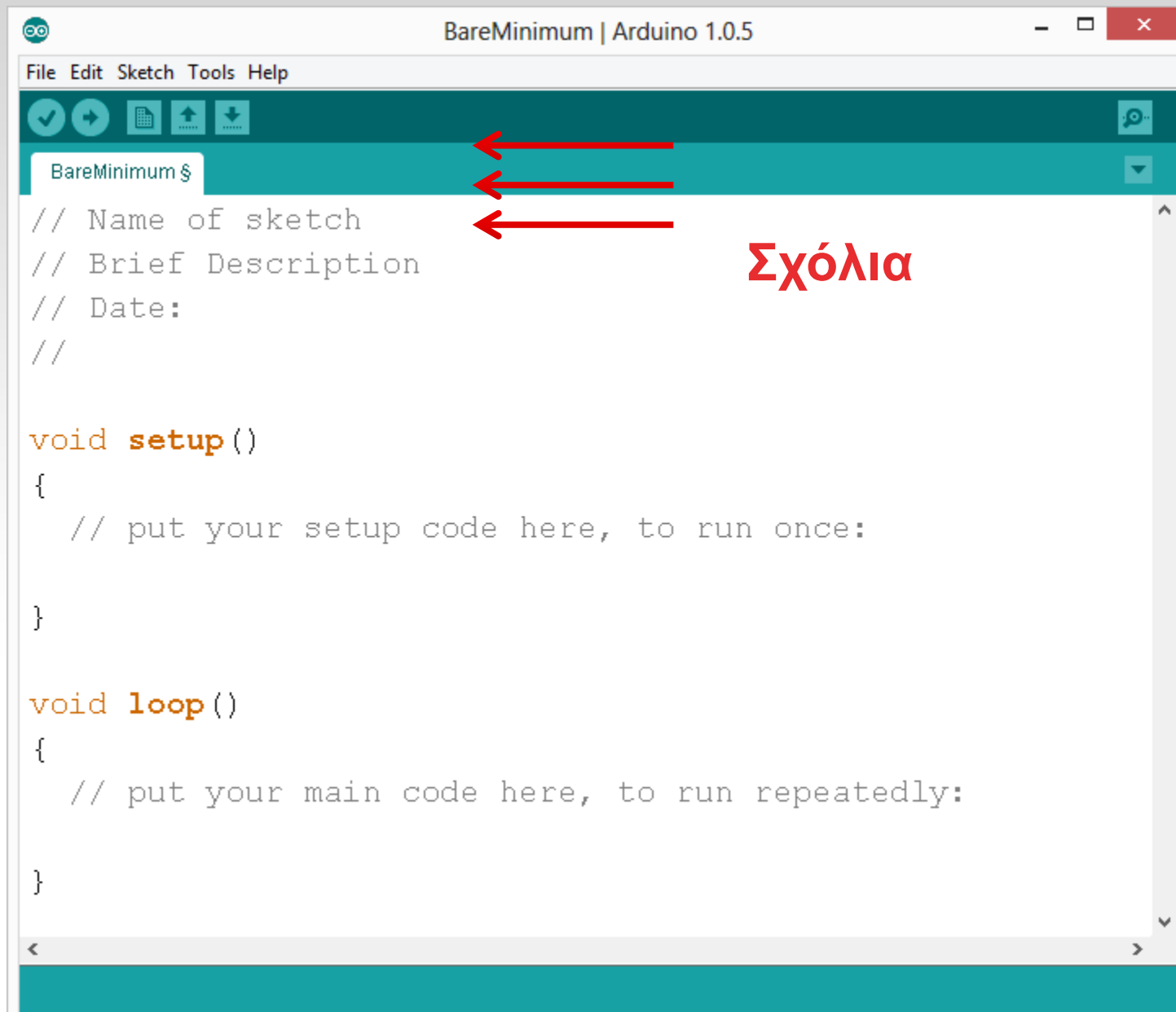
- “Hello World” of Physical Computing

- *Pseudo-code – how should this work?*



Σχόλια, Σχόλια, Σχόλια

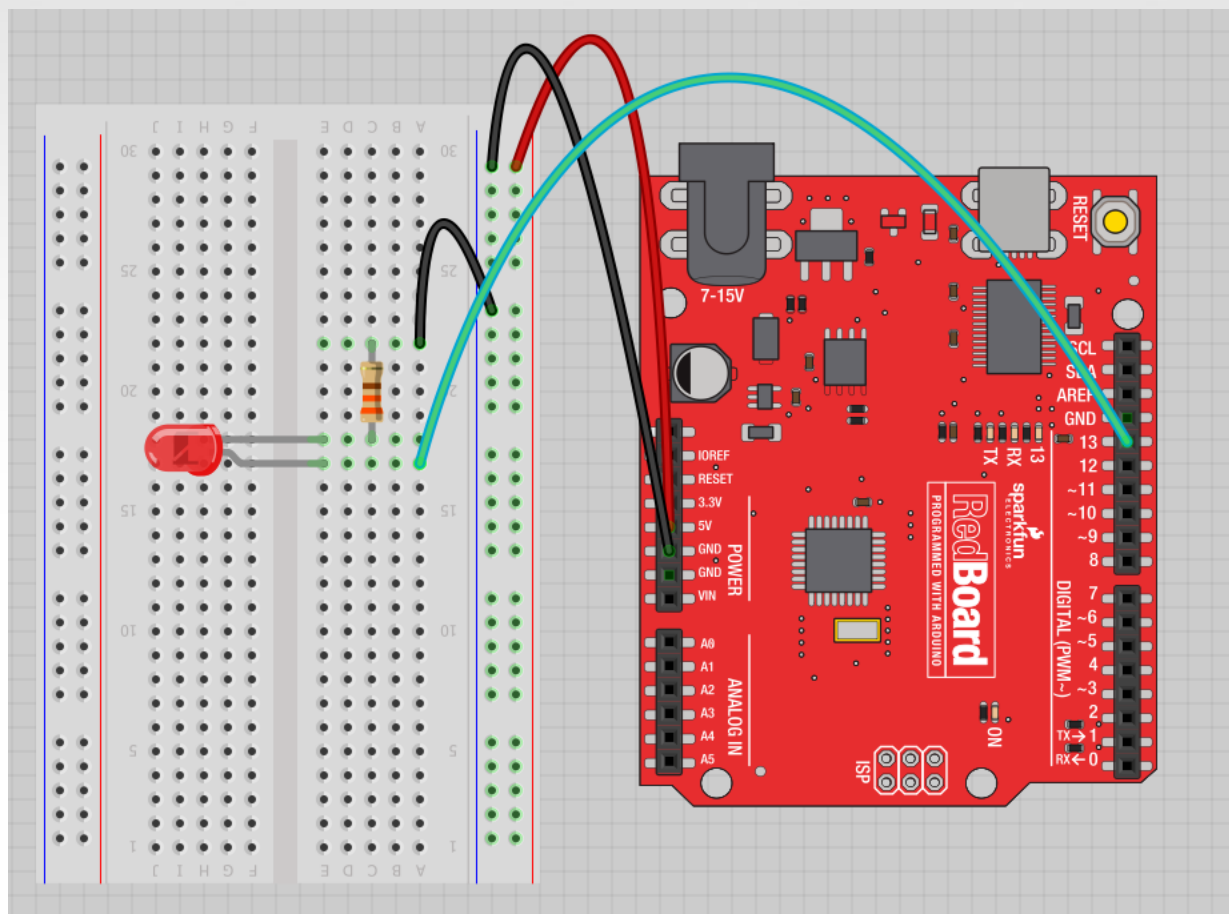
- Τα σχόλια είναι για σας– τον προγραμματιστή και τους φίλους του...ή οποιονδήποτε θελήσει να διαβάσει τον κώδικα.
- `//` αυτό είναι για σχόλια μιας γραμμής
- `//` Είναι καλό να βάζουμε μια περιγραφή πριν από κάποιο 'κόλπο'
- `/*` αυτό είναι για σχόλια πολλών γραμμών
- Σαν και αυτό...
- ή αυτό...
- `*/`



Τρεις εντολές που πρέπει να γνωρίζουμε...

- `pinMode (pin, INPUT/OUTPUT) ;`
- `πχ: pinMode (13, OUTPUT) ;`
- `digitalWrite (pin, HIGH/LOW) ;`
- `πχ: digitalWrite (13, HIGH) ;`
- `delay (time_ms) ;`
- `πχ: delay (2500) ; // καθυστέρηση 2.5 sec.`
- `// NOTE: -> commands are CASE-sensitive`
(Κεφαλαία μικρά έχουν σημασία)

Project #1: Καλωδίωση



Ένα χρήσιμο
προγραμμα για τις
καλωδιώσεις των
κατασκευών μας
είναι το Fritzing

Η εικόνα φτιάχτηκε με το πρόγραμμα Fritzing

Έννοιες προγραμματισμού : Μεταβλητές

ProtosnapProMiniExample2 \$

```
// Comments go here
// Written by:  Joesephine Jones
// Date:  April 12, 2013

int sensorValue;
int ledPin;

void setup()
{
  // put your setup code here, to run once:
  int setupVariable;

}

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

πεδίο
μεταβλητών

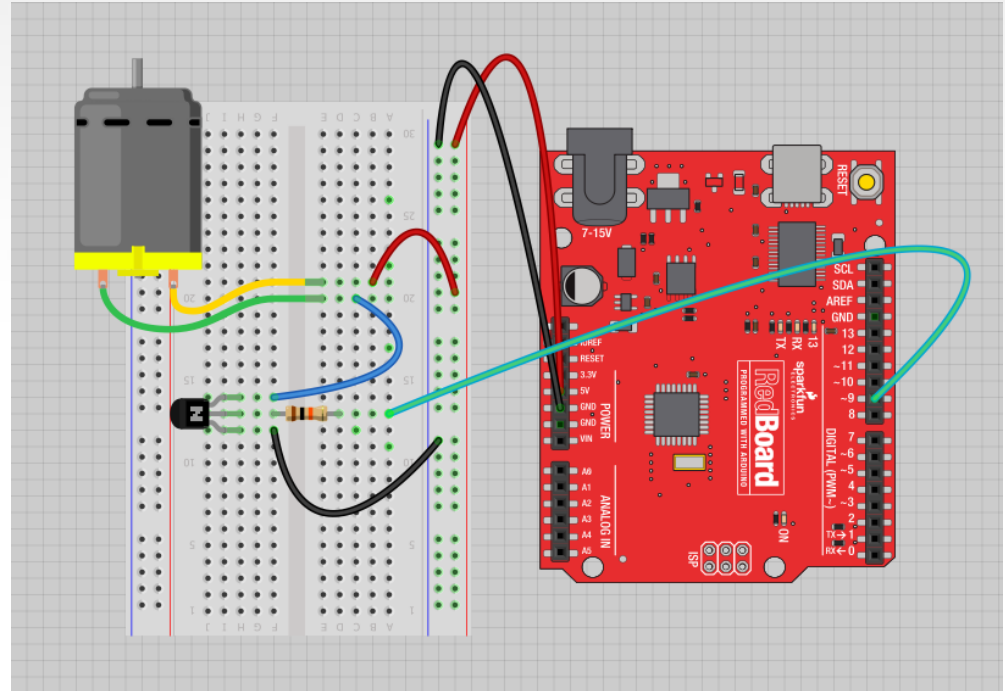
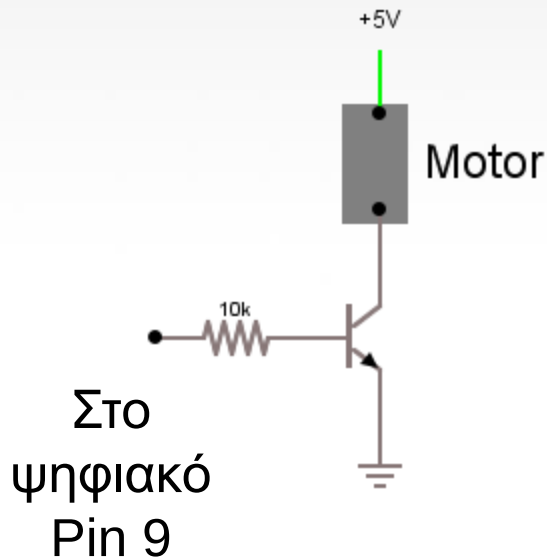
• *Global*

• ---

• *Function-level*

“Οδηγώντας” κινητήρες ή άλλα φορτία με μεγάλο ρεύμα

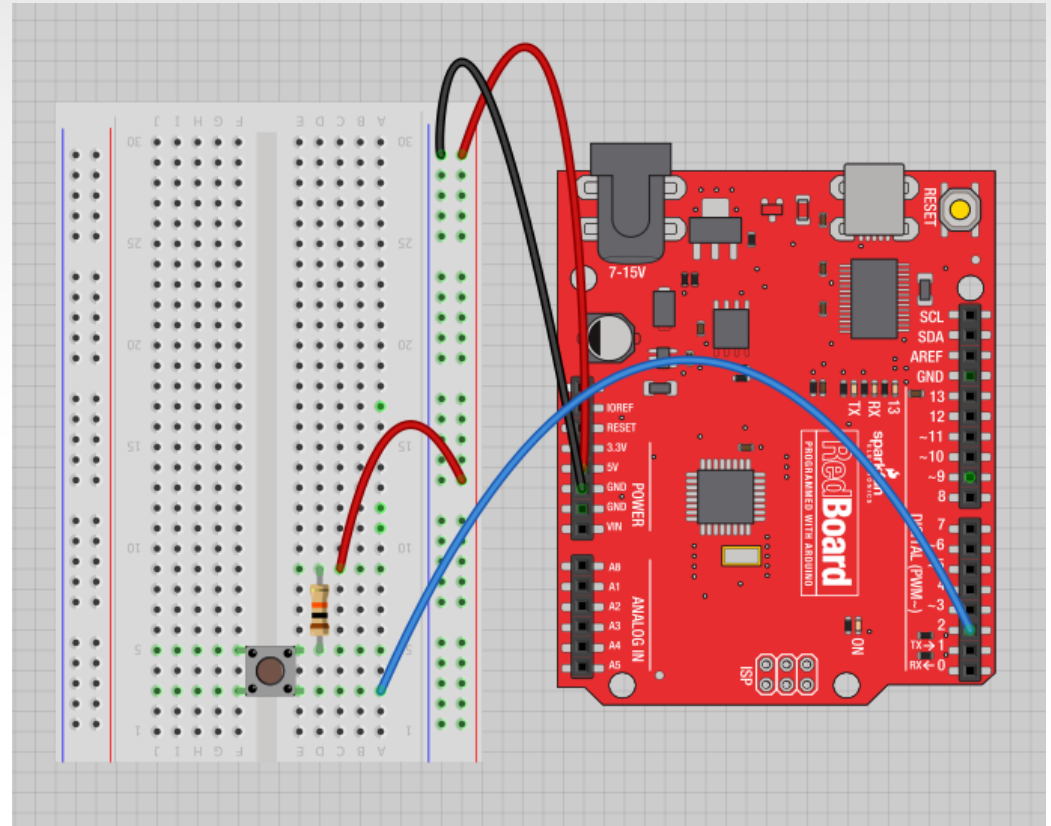
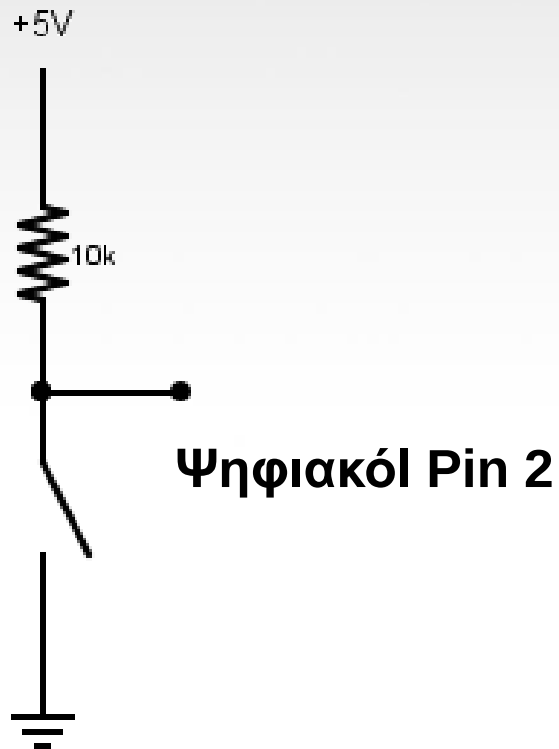
- **NPN Transistor (Common Emitter “Amplifier” Circuit)**



Project #2 – Ψηφιακή είσοδος

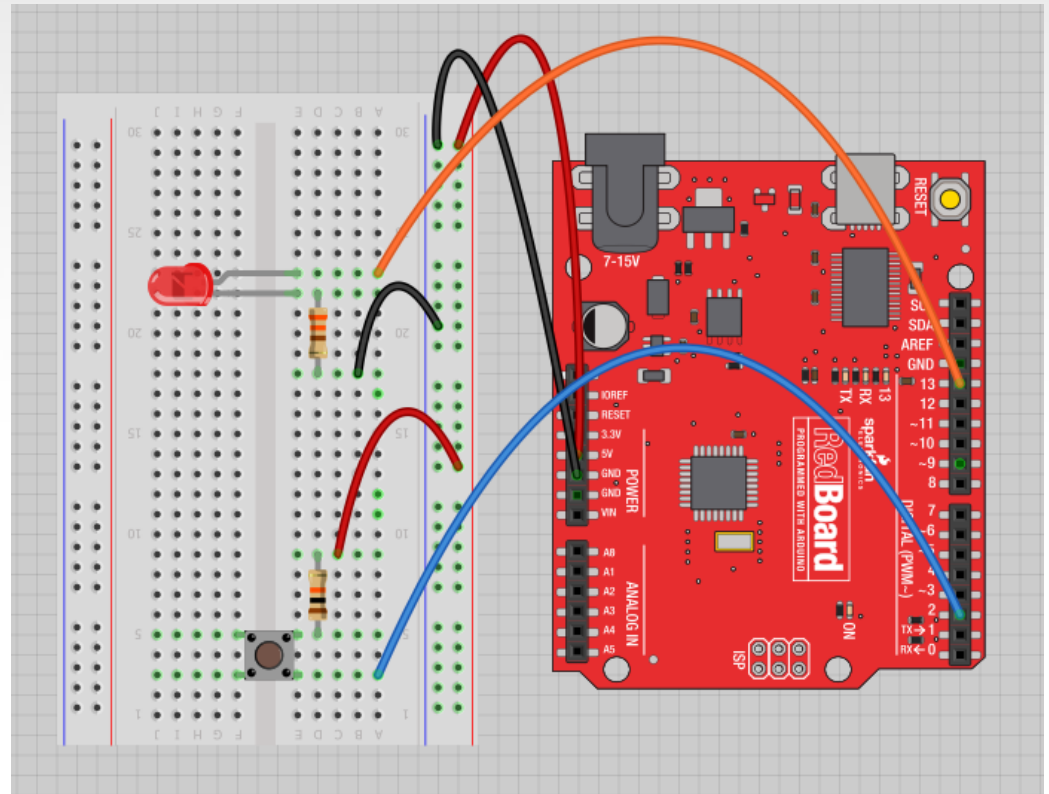
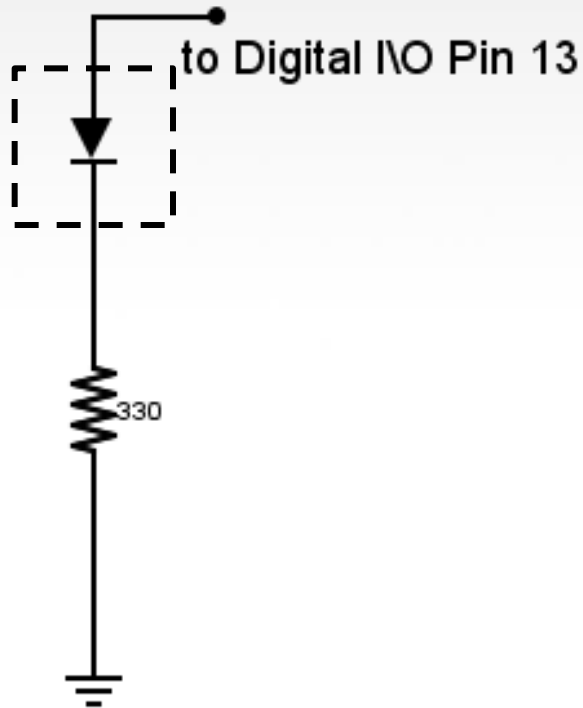
- In Arduino, open up:
- File → Examples → 02.Digital → Button

Ψηφιακοί αισθητήρες(διακόπτες) αντίσταση Pull-up (κύκλωμα)



Ψηφιακοί αισθητήρες(διακόπτες) προσθέσαμε ενδεικτικό LED στο Pin 13

Είναι σαν το πρώτο
μας κύκλωμα!



Ψηφιακή είσοδος

- Συνδέστε την ψηφιακή είσοδο του Arduino με τα Pins # 0 – 13 (Although pins # 0 & 1 are also used for programming)
- Χρειαζόμαστε την εντολή `pinMode` :
- `pinMode (pinNumber, INPUT) ;`
- *Make sure to use ALL CAPS for **INPUT***
- Για να διαβάσουμε την ψηφιακή είσοδο:
- `int buttonState = digitalRead (pinNumber) ;`
- Digital Input values are only **HIGH** (On) or **LOW** (Off)

`int buttonState = digitalRead(pushButton);`

We declare a variable as an integer.

We set it equal to the function `digitalRead(pushButton)`

The function `digitalRead()` will return the value 1 or 0, depending on whether the button is being pressed or not being pressed.

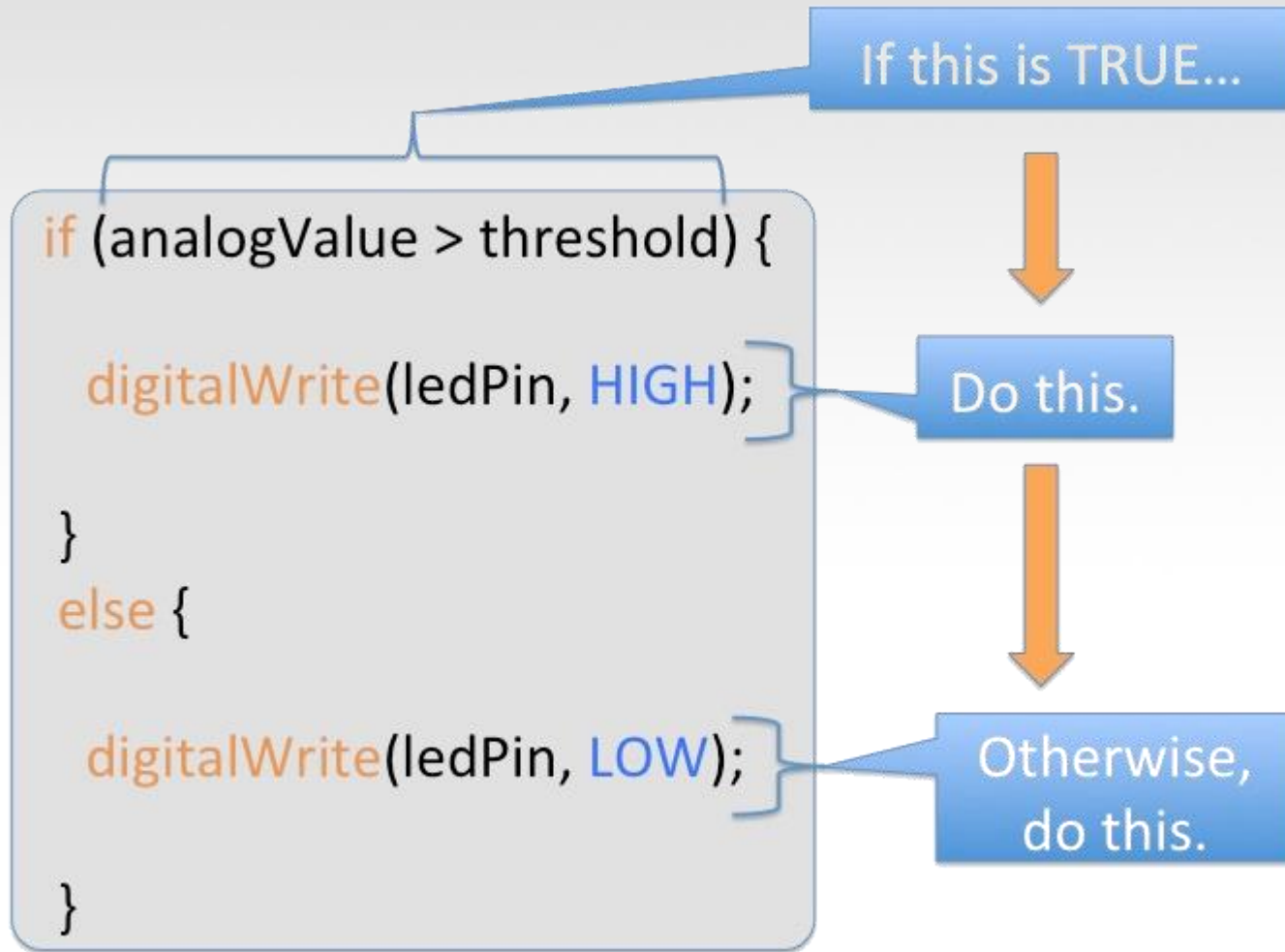
We name it `buttonState`

Recall that the `pushButton` variable stores the number 2

The value 1 or 0 will be saved in the variable `buttonState`.

Προγραμματίζοντας: Δομές ελέγχου ροής

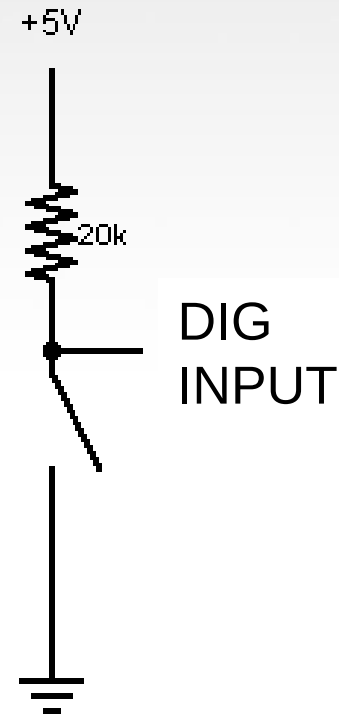
`if ( )`



Προγραμματίζοντας: Δομές ελέγχου ροής

`if ( )`

```
• void loop()  
• {  
•   int buttonState =  
digitalRead(5);  
•   if(buttonState == LOW)  
•   { // do something  
•   }  
•   else  
•   { // do something else  
•   }  
• }
```



Τελεστές σύγκρισης

<Boolean>	Description
() == ()	is equal?
() != ()	is not equal?
() > ()	greater than
() >= ()	greater than or equal
() < ()	less than
() <= ()	less than or equal

analogRead()

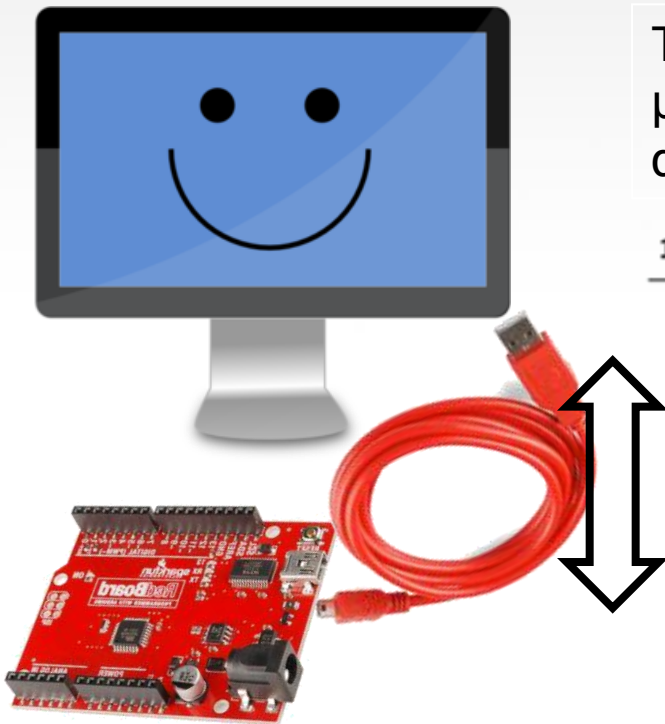
- Ο Arduino χρησιμοποιεί 10-bit A/D Converter:
- Αυτό σημαίνει τιμές εισόδου από 0 έως 1023
 - Για 0 V → δίνει 0
 - Για 5 V → δίνει 1023

πχ:

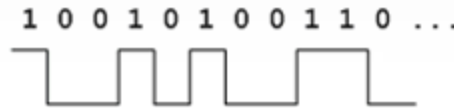
- `int sensorValue = analogRead(A0) ;`

Σειριακή επικοινωνία

Όταν έχουμε ανταλλαγή δεδομένων μεταξύ δύο συσκευών.

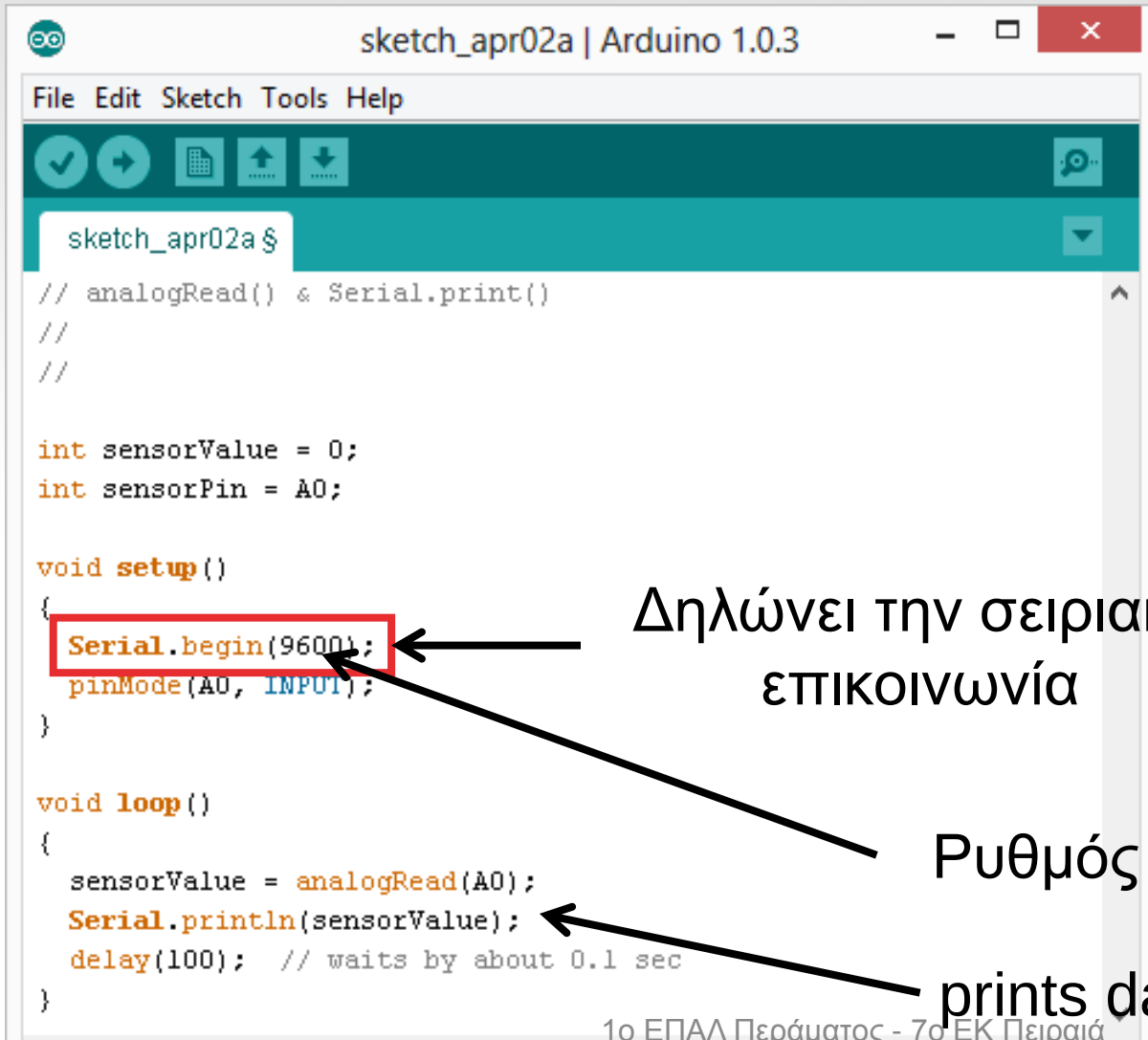


Τα δεδομένα περνούν μεταξύ του Η/Υ και του Arduino με ένα καλώδιο USB. Εκπέμπονται σαν μια συνεχή ακολουθία από μηδέν ('0') και ένα ('1').



Ο Arduino χρησιμοποιεί το ψηφιακό pin # 0 για λήψη και το ψηφιακό pin #1 για εκπομπή.

Serial Monitor & analogRead()



```
// analogRead() & Serial.print()
//
//

int sensorValue = 0;
int sensorPin = A0;

void setup()
{
  Serial.begin(9600);
  pinMode(A0, INPUT);
}

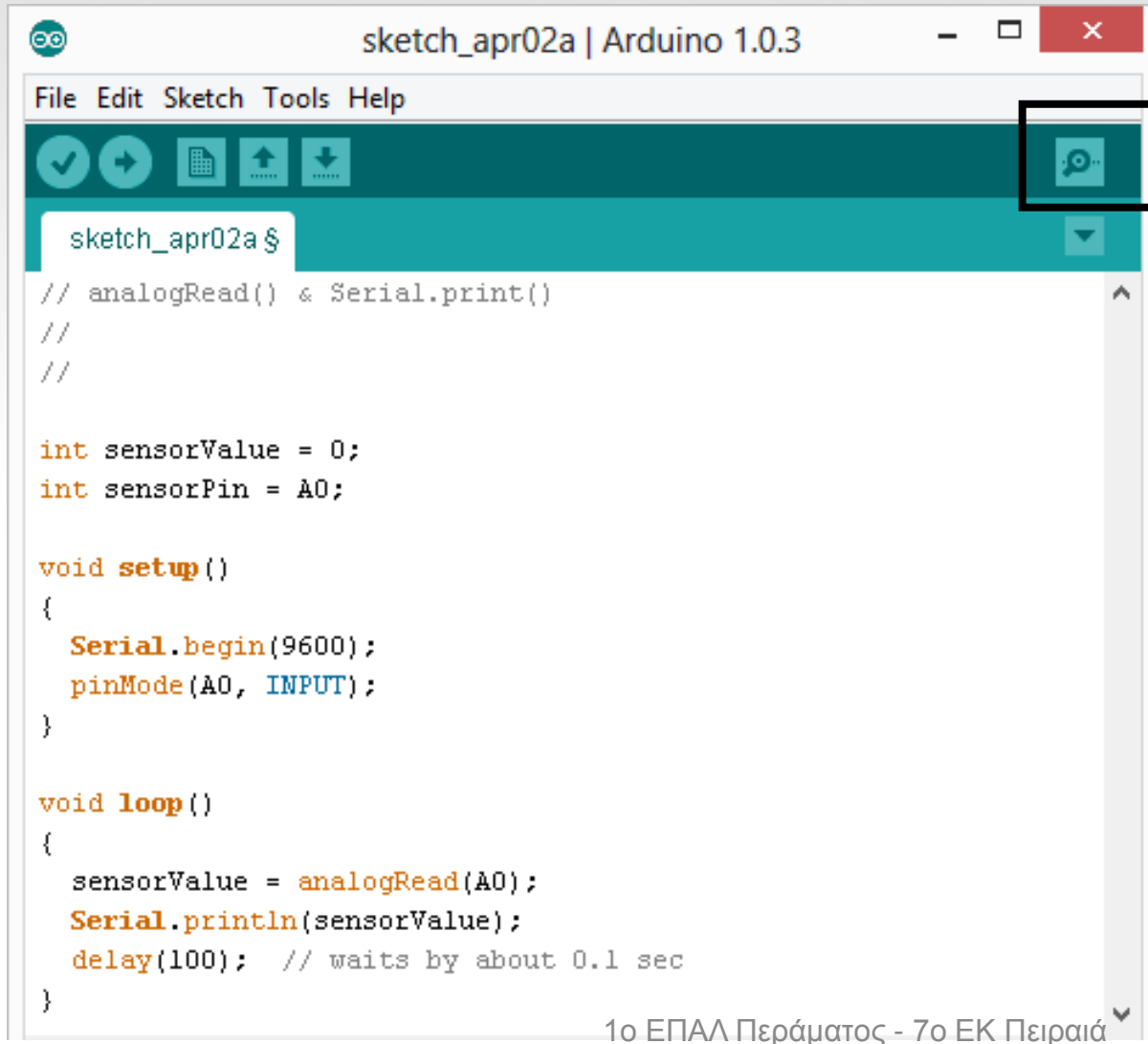
void loop()
{
  sensorValue = analogRead(A0);
  Serial.println(sensorValue);
  delay(100); // waits by about 0.1 sec
}
```

Δηλώνει την σειριακή
επικοινωνία

Ρυθμός 9600 baud

prints data to serial bus

Serial Monitor & analogRead()

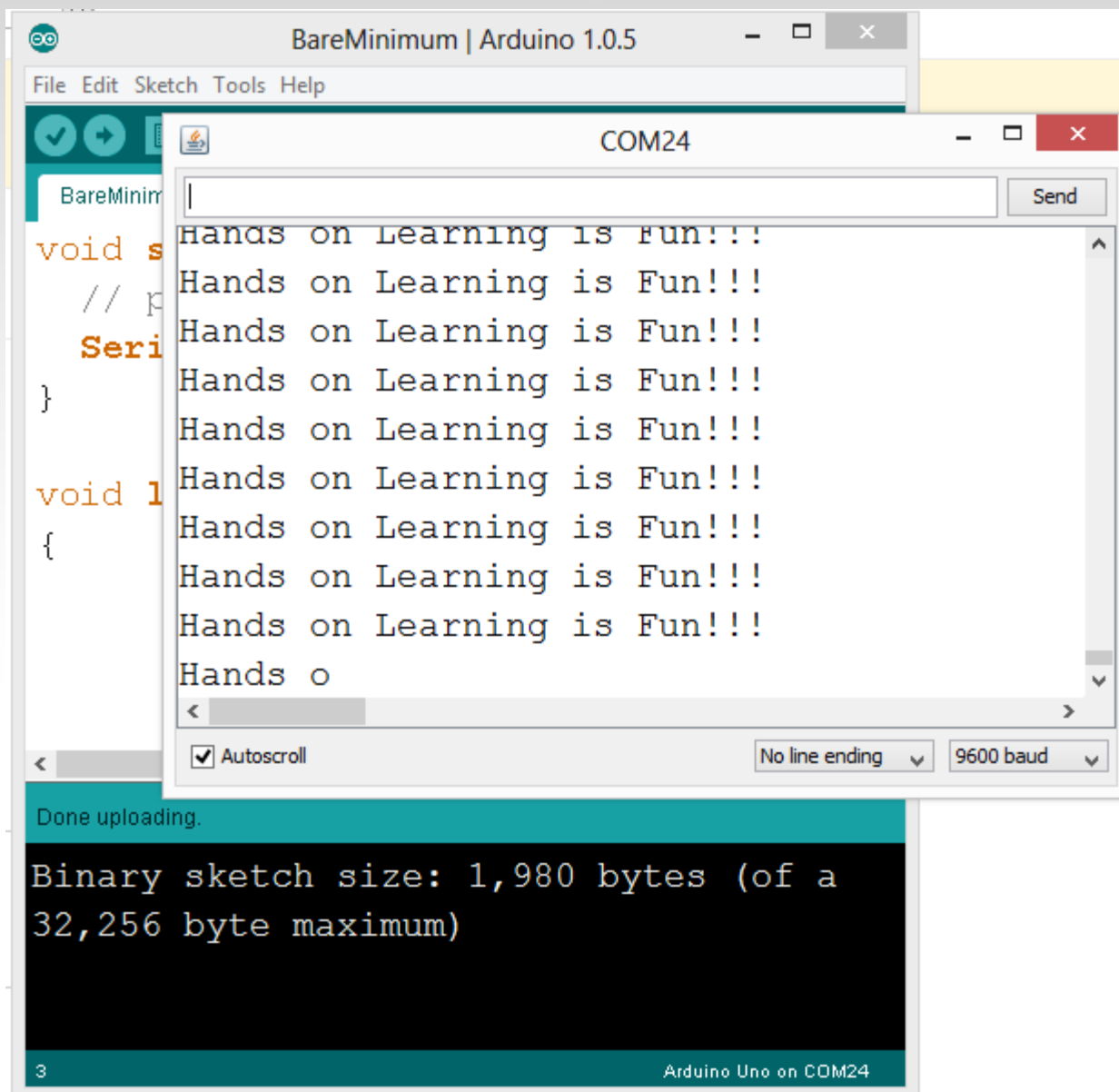


Ανοίγει το
παράθυρο
Serial Terminal

σειριακής επικοινωνίας συνέχεια

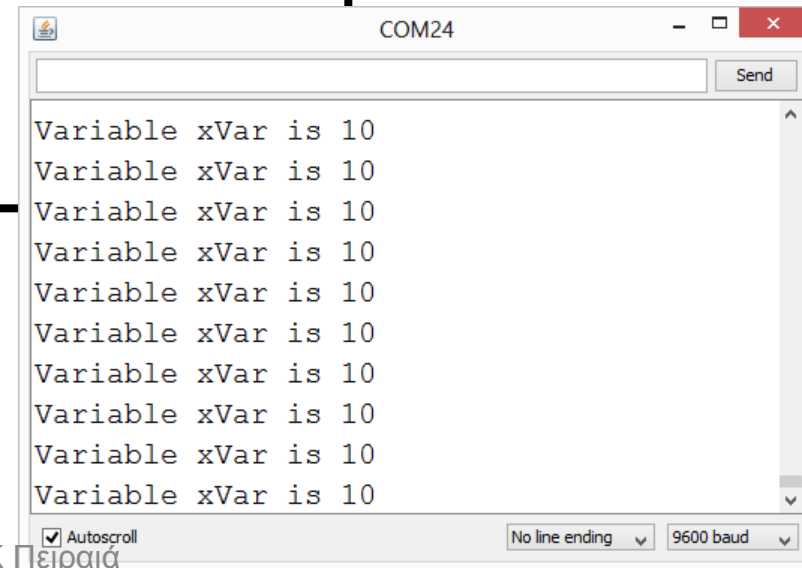
Στέλνοντας ένα μήνυμα

```
void loop ( )  
{  
  Serial.print("Hands on ") ;  
  Serial.print("Learning ") ;  
  •   Serial.println("is Fun!!!") ;  
  
}
```



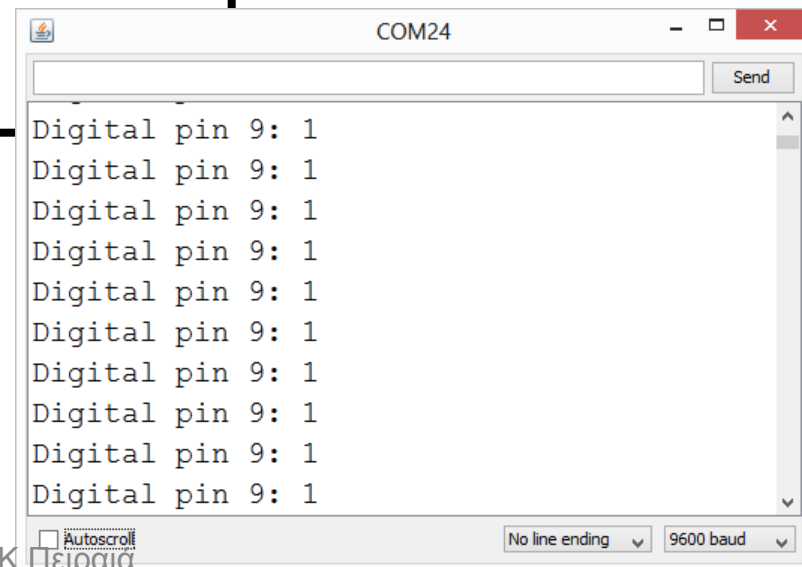
Σειριακή επικοινωνία: Serial Debugging

```
void loop()  
{  
    int xVar = 10;  
    Serial.print ( "Variable xVar is " ) ;  
    Serial.println ( xVar ) ;  
}
```



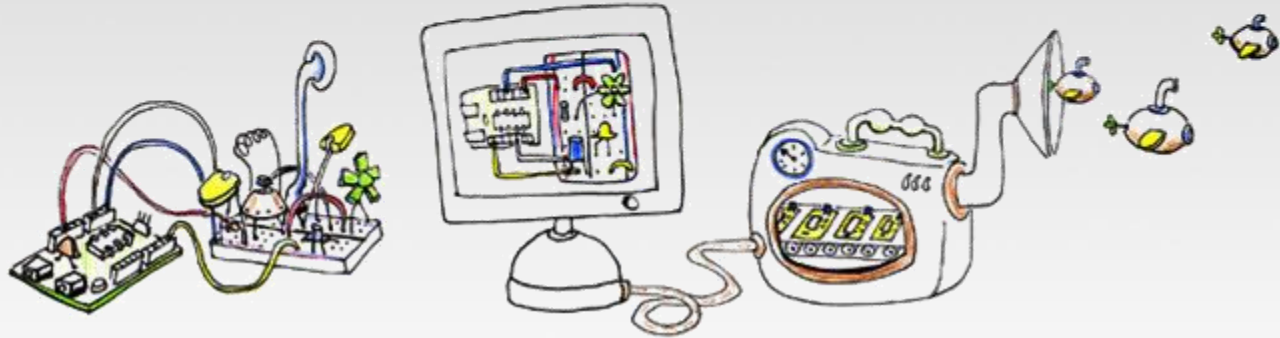
Σειριακή επικοινωνία : Serial Troubleshooting

```
void loop ( )  
{  
    Serial.print ("Digital pin 9: ");  
    Serial.println (digitalRead(9));  
}
```





FRIITZING



Virtual Electrical Prototyping Project
started in 2007 by the Interaction Design Lab
at the University of Applied Science Potsdam, Germany

Open Source

Prototypes: Document, Share, Teach, Manufacture

Εξομοίωση για τον Arduino

<http://virtronics.com.au/Simulator-for-Arduino.html>

The screenshot displays the Arduino IDE simulator interface. The main window is titled "Fade.ino" and contains a code editor with the following code:

```
0013 // declare pin 9 to be an output:
0014 pinMode(9, OUTPUT);
0015 }
0016
0017 void loop() {
0018 // set the brightness of pin 9:
0019 analogWrite(9, brightness);
0020
0021 // change the brightness for next time
0022 brightness = brightness + fadeAmount;
0023
0024 // reverse the direction of the fading
0025 if (brightness == 0 || brightness == 255) {
0026 fadeAmount = -fadeAmount ;
0027 }
0028 // wait for 30 milliseconds to see the
0029 delay(30);
0030 }
```

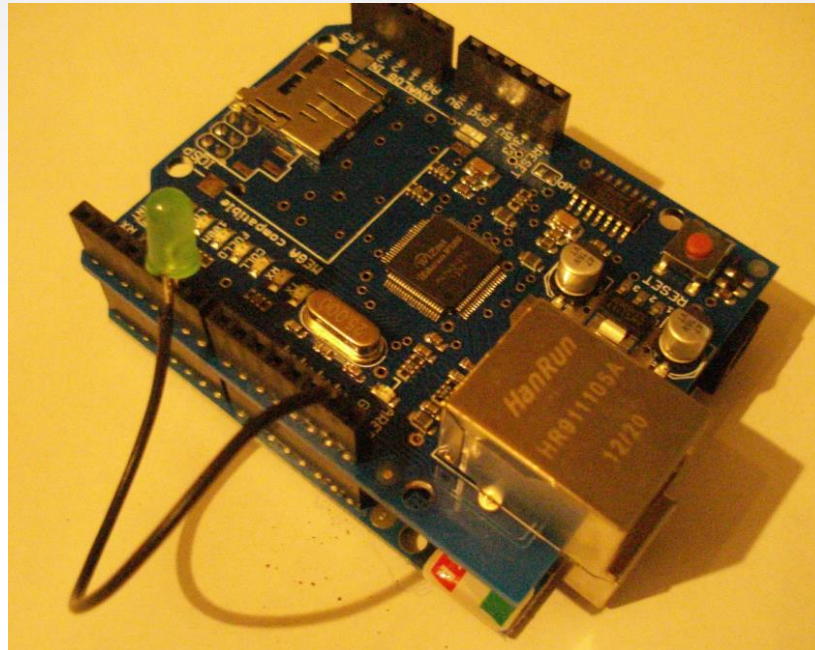
Below the code editor is a variable table:

No	Variable	Value
	int	Value
1	brightness	85
2	fadeAmount	5

On the right side of the simulator is a virtual representation of an Arduino Uno board. The board is blue and features various components including a USB Type-B port, a DC power jack, a reset button, and a microcontroller. The board is labeled "ARDUINO" and "UNO". The pin headers are visible, with pins 1 through 14 labeled. The status bar at the bottom indicates "Line: 0022/31", "brightness => 85", "7 days and 79 sketch simulations left", "Level1", and "F1=Help".

Άλλες Εφαρμογές

- **Control an LED over the Internet using the Arduino Ethernet Shield**

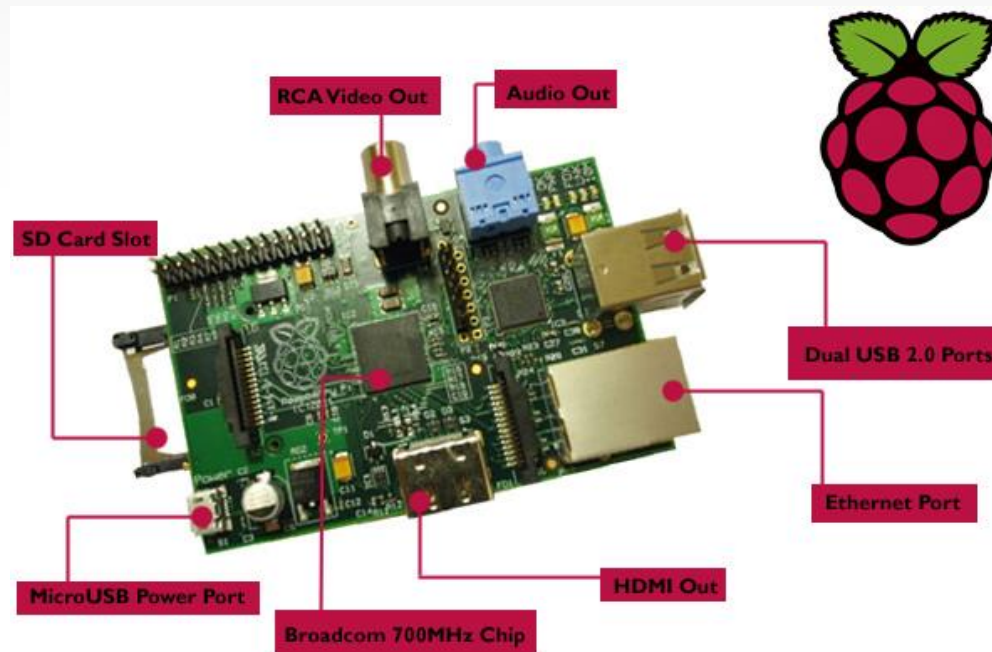


Άλλες Εφαρμογές

RASPBERRY PI

LINUX Single Board Computer

- Αναβόσβημα Led χρήση εντολών από την βιβλιοθήκη Wiring Pi (Arduino like Programming)



Άλλες Εφαρμογές

**Ανάπτυξη εφαρμογής με έλεγχο από PS2 Keyboard
και απεικόνιση σε VGA Monitor με Gameduino**



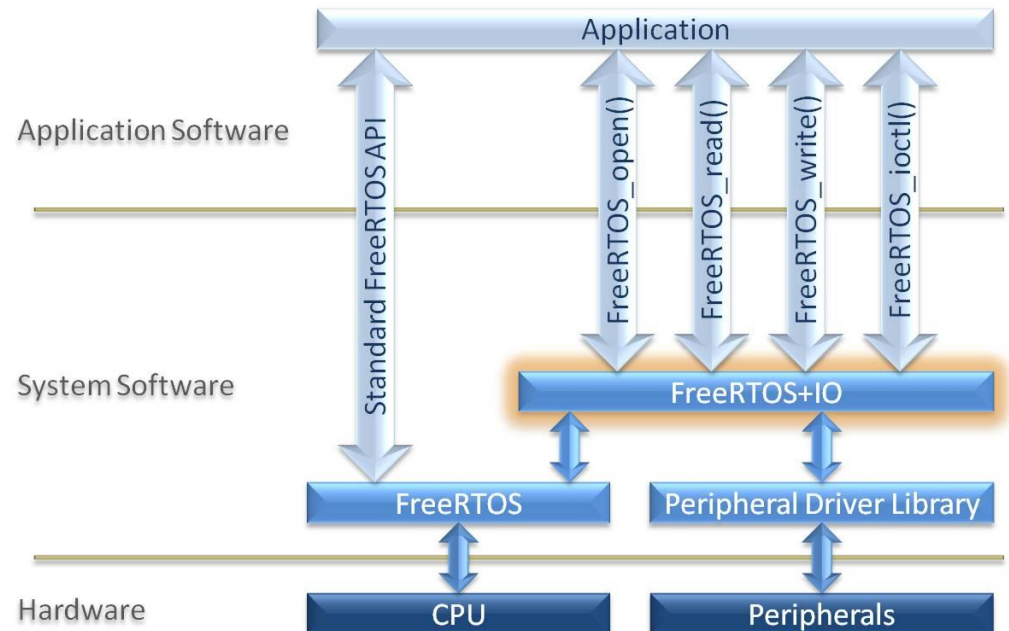
Άλλες Εφαρμογές

Ανάπτυξη εφαρμογής με χρήση του RTOS (Λειτουργικού Συστήματος Πραγματικού Χρόνου) **FreeRTOS**



Σύστημα μετρήσεων με 5 διεργασίες και FreeRTOS

- Luminosity and Temperature measurements - Task A & Task B
- Optical switch - Task C
- Sending information's to the personal computer - Task D
- Receiving information from the personal computer - Task E



Το μέλλον

AVR Arduino microcontroller ATmega32u4
Operating Digital I/O Pins 20

PWM Channels 7

Analog Input Channels 12

Flash Memory 32 KB (of which 4 KB used by bootloader)

SRAM 2.5 KB

EEPROM 1 KB

Clock Speed 16 MHz



Linux microprocessor Processor Atheros AR9331

Architecture MIPS @400MHz

Operating Voltage 3.3V

Ethernet IEEE 802.3 10/100Mbit/s

WiFi IEEE 802.11b/g/n

USB Type-A 2.0 Host/Device

Card Reader Micro-SD only

RAM 64 MB DDR2

Flash Memory 16 MB

