

Arduino*

www.arduino.cc

*after Arduin of Ivrea (955-1015), King of Italy

Σπύρος Βερονίκης, Ph.D.

Ηλεκ/γος Μηχ/κός & Μηχ/κός Η/Υ

spyros.veronikis@gmail.com

<http://www.linkedin.com/in/veronikis>



Arduino???

Τι είναι το Arduino?

- Είναι ένα εργαλείο για να δημιουργήσουμε υπολογιστές που μπορούν να αλληλεπιδράσουν με το περιβάλλον τους. Είναι βασισμένο σε μια πλατφόρμα ανοικτού κώδικα. Μπορεί να ανιχνεύσει σήματα από το περιβάλλον και να χρησιμοποιήσει ενεργοποιητές για να ελέγξει φώτα, κινητήρες, κ.α. Μπορεί επίσης να επικοινωνήσει με άλλους υπολογιστές (ή Arduinos), τοπικούς ή απομακρυσμένους.

Τι είναι το πιο cool πράγμα που θα μάθω/κάνω σήμερα;

- Θα δούμε **ΤΙ** μπορούμε να κάνουμε με ένα Arduino
(θα φτιάξουμε ένα video-game, το ArduPong)
- Θα δούμε **ΠΩΣ** μπορούμε να το κάνουμε

Περιεχόμενο παρουσίασης

- Εισαγωγή στο Arduino
- Ρύθμιση του περιβάλλοντος ανάπτυξης
- Το πρώτο σας πρόγραμμα
- Δημιουργία κυκλωμάτων
- Είσοδοι, έξοδοι, αισθητήρες και ενεργοποιητές
- Arduino + Processing Language
- Επεκτείνοντας τις δυνατότητες του Arduino
- Πηγές πληροφόρησης

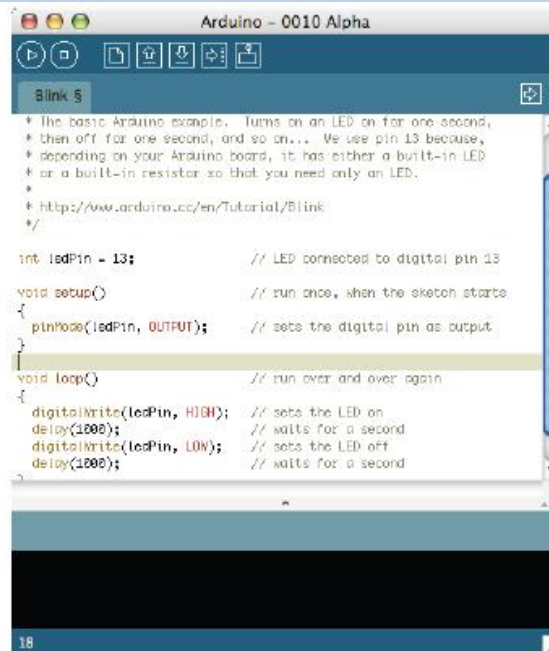
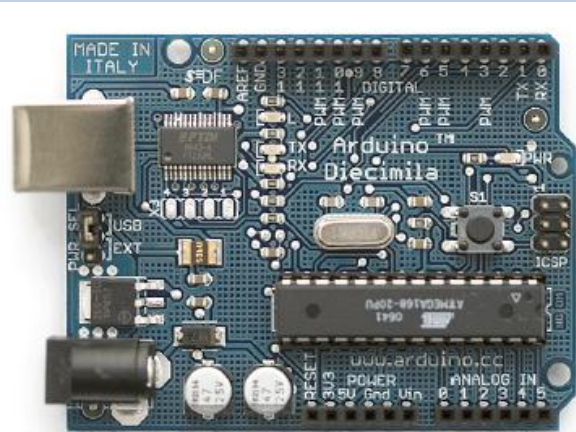
Τι είναι το Arduino ;

Το Arduino μπορεί να αναφέρεται σε ένα από τα εξής 3 πράγματα

Μια Ηλεκτρονική πλακέτα

Ένα Προγραμματιστικό περιβάλλον

Μια κοινότητα και μια φιλοσοφία



Ποιος το χρησιμοποιεί; Για ποιο σκοπό;

- Ερευνητές (CERN, MIT, ...)
 - Συλλογή μετρήσεων, γάντι νοηματικής γλώσσας (HMI/HCI), παρακολούθηση σεισμικής δραστηριότητας, επεξεργασία DNA
- Εταιρείες & Start-ups
 - Arducopter, interface design, Pebble Prototype, ArduSat
- Hobbyists
 - Αυτοσχέδια παιχνίδια, τάισμα κατοικίδιων, TV mute (Enough already),
- Μαθητές
 - Μουσικές συσκευές, φροντίδα φυτών, Robots

Δείτε την παρουσίαση του Massimo Banzi, “How Arduino is open-sourcing imagination”

<http://www.youtube.com/watch?v=UoBUXOOdLXY>

Βρείτε τις ομοιότητες



* Marty Feldman (1933-1982)
http://en.wikipedia.org/wiki/Marty_Feldman



* Arduino Duemilanove
<http://en.wikipedia.org/wiki/Arduino>

Κατ' εικόνα και καθ' ομοίωσιν ...

- Οι μικροεπεξεργαστές σχεδιάστηκαν έχοντας κατά νου το ανθρώπινο μοντέλο
- Διαθέτουν εγκέφαλο, δηλαδή ένα κέντρο ελέγχου και λήψεως αποφάσεων
- Διαθέτουν αισθητήρια για να αντιλαμβάνονται το περιβάλλον τους

- Μικρόφωνα για τη σύλληψη ήχων
- Κάμερες για τη σύλληψη οπτικών ερεθισμάτων
- Διακόπτες και αισθητήρες πίεσης για την αφή αντικειμένων

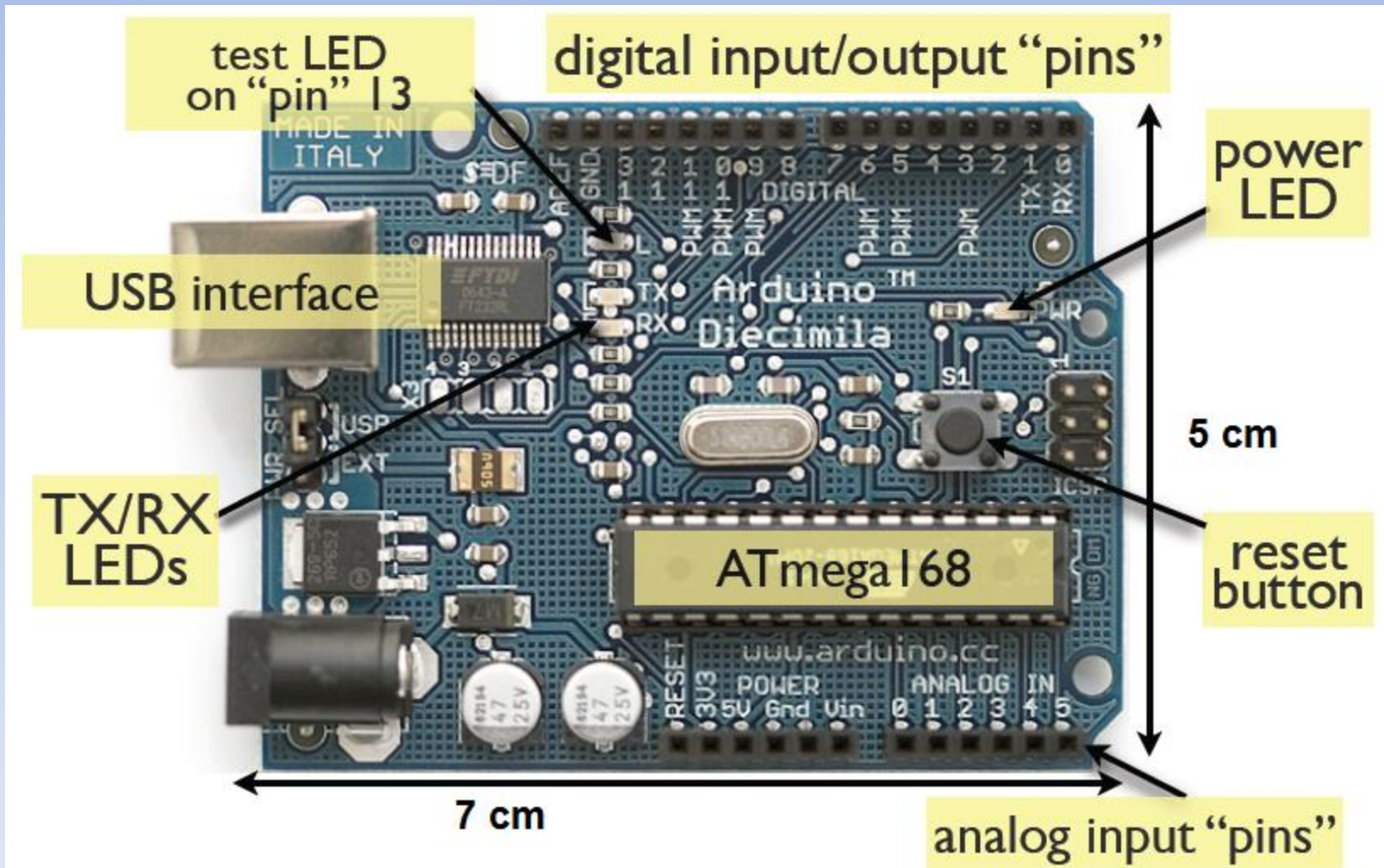
Επιπλέον μπορούν να αντιληφθούν μεταβολές στην ένταση φωτός, στην βαρομετρική πίεση, στην βαρύτητα (accelerometers), στη συχνότητα οπτικών, ακουστικών και ηλεκτρικών σημάτων, κ.α.

- Διαθέτουν ενεργοποιητές για να αλληλεπιδράσουν με το περιβάλλον τους
- Χρησιμοποιούν λαμπτήρες για να υποδηλώσουν μια κατάσταση (π.χ. Morse Code)
- Χρησιμοποιούν μεγάφωνα για να δημιουργήσουν ήχους (τόνους ή και σύνθεση φωνής)
- Χρησιμοποιούν κινητήρες για να μετακινήσουν αντικείμενα

Επιπλέον μπορούν να επικοινωνήσουν ενσύρματα ή ασύρματα με άλλες συσκευές ανταλλάσσοντας δεδομένα

- Διαθέτουν μνήμη για να καταγράψουν και να ανασύρουν δεδομένα

Arduino: η πλακέτα



Arduino: η πλακέτα

- 16 kBytes of Flash program memory
- 1 kByte of RAM
- 16 MHz clock
- Είσοδοι & Έξοδοι
 - 13 ακροδέκτες ψηφιακών εισόδων/εξόδων
 - 5 ακροδέκτες αναλογικής εισόδου
 - 6 ακροδέκτες αναλογικής εξόδου
- Stand-alone (no computer needed)

Arduino: Ορολογία

- **“sketch”:**

το πρόγραμμα που γράφουμε για να εκτελέσει το Arduino

- **“pin”:**

ο ακροδέκτης που συνδέουμε σε κάτι (π.χ. σε ένα LED ή έναν αισθητήρα)

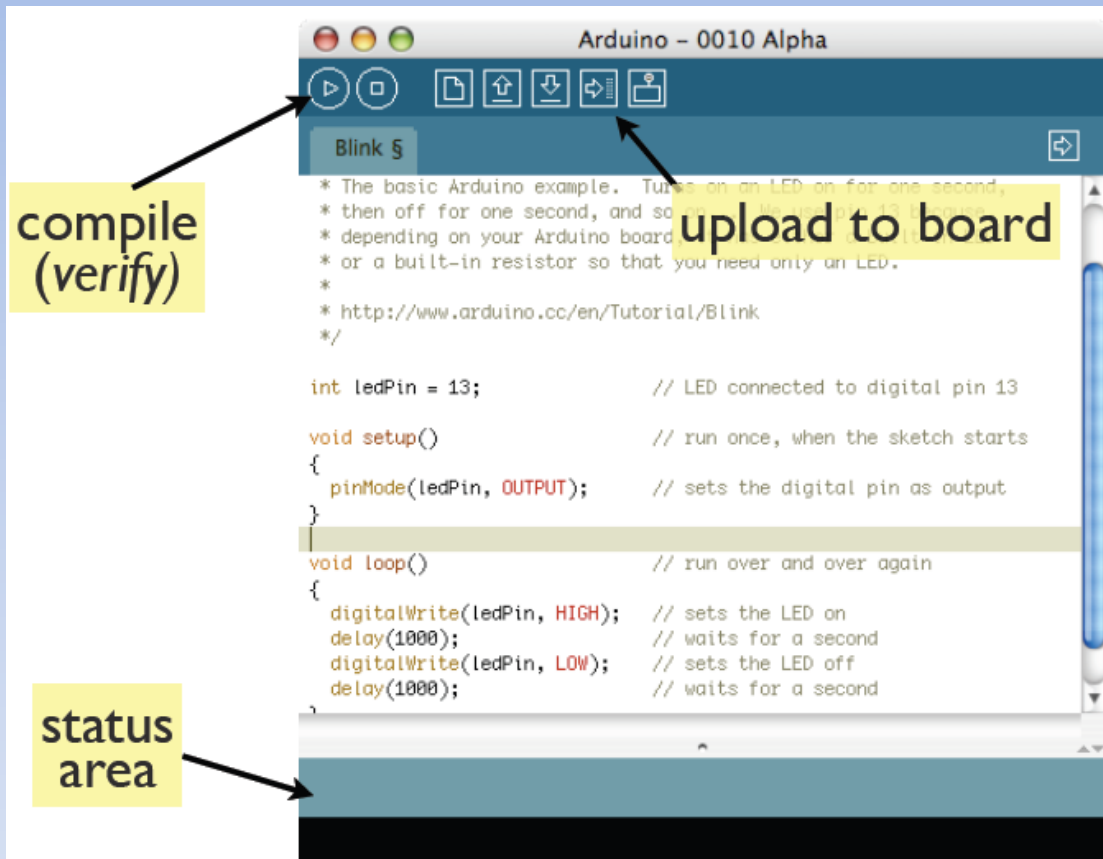
- **“digital”:**

μια διακριτή τιμή τάσης ή έντασης ρεύματος (π.χ., LOW/HIGH ή FALSE/TRUE)

- **“analog”:**

μια συνεχής τιμή τάσης ή έντασης ρεύματος (π.χ., 0-255 για τάση ή ένταση φωτεινότητας ή ταχύτητα περιστροφής κινητήρα)

Arduino: Το λογισμικό



Το Ολοκληρωμένο Σχεδιαστικό Περιβάλλον (Integrated Design Environment) του Arduino

- Όπως ένας κειμενογράφος
- Προβολή/ δημιουργία/ διόρθωση προγραμμάτων
- Μεταφόρτωση στον επεξεργαστή

Arduino: Φιλοσοφία και Κοινότητα

- Υπολογιστική πλατφόρμα Ανοικτού Λογισμικού
 - Open source hardware (schematics)
 - Ανοικτό λογισμικό: ελεύθερο προς ανάγνωση και τροποποίηση
- Χτισμένη από την Κοινότητα, για την Κοινότητα
 - **Arduino Playground**: πλήθος παραδειγμάτων (συνεισφέρει και τροποποιεί ο καθένας)
<http://playground.arduino.cc/>
 - **Forums** από πολλούς enthusiasts, πρόθυμους να βοηθήσουν
<http://arduino.cc/forum/>

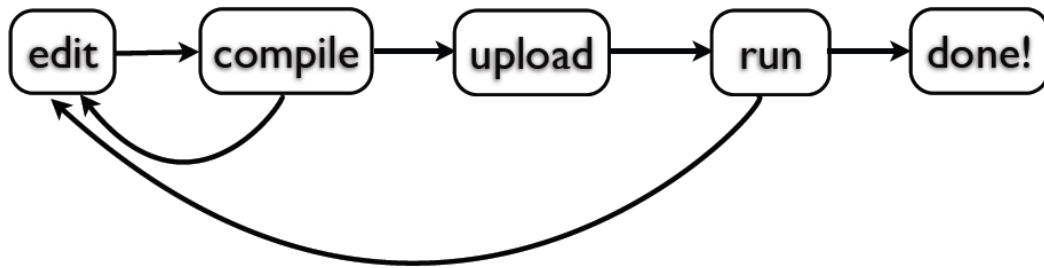
Χρησιμοποιώντας το Arduino

- Δεν απαιτείται εγκατάσταση του λογισμικού (Windows)
 - Μεταφόρτωση από: <http://arduino.cc/en/Main/Software>
- Εγκατάσταση των drivers
- Επιλογή της πλακέτας
- Επιλογή της σειριακής θύρας προγραμματισμού
- Μεταφόρτωση του προγράμματός σας

Το παράδειγμα του Blinking L.E.D.

1. Γράφουμε το πρόγραμμά μας (sketch)
2. Ελέγχουμε το sketch για σφάλματα (compile)
3. Μεταφορτώνουμε το sketch
4. Εκτελούμε το πρόγραμμα

Για να διαφοροποιήσουμε το αποτέλεσμα, επαναλαμβάνουμε τα 4 βήματα



```
void setup() {  
  pinMode(ledPin, OUTPUT); // sets t  
}  
void loop() {  
  digitalWrite(ledPin, HIGH); // sets t  
  delay(1000); // waits  
  digitalWrite(ledPin, LOW); // sets t  
  delay(1000); // waits  
}
```



compile

Done compiling.



upload

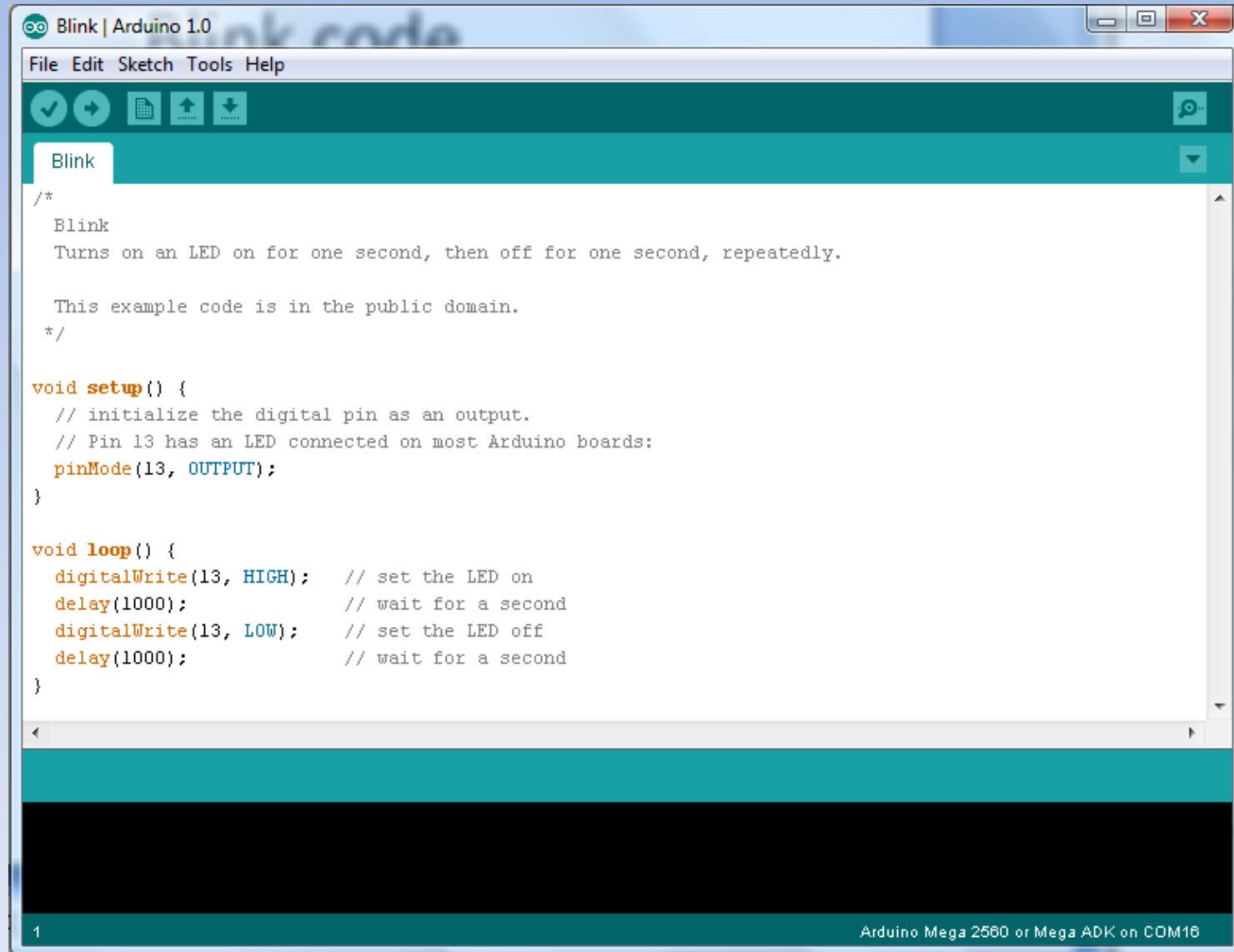


TX/RX flash



blink blink
sketch runs

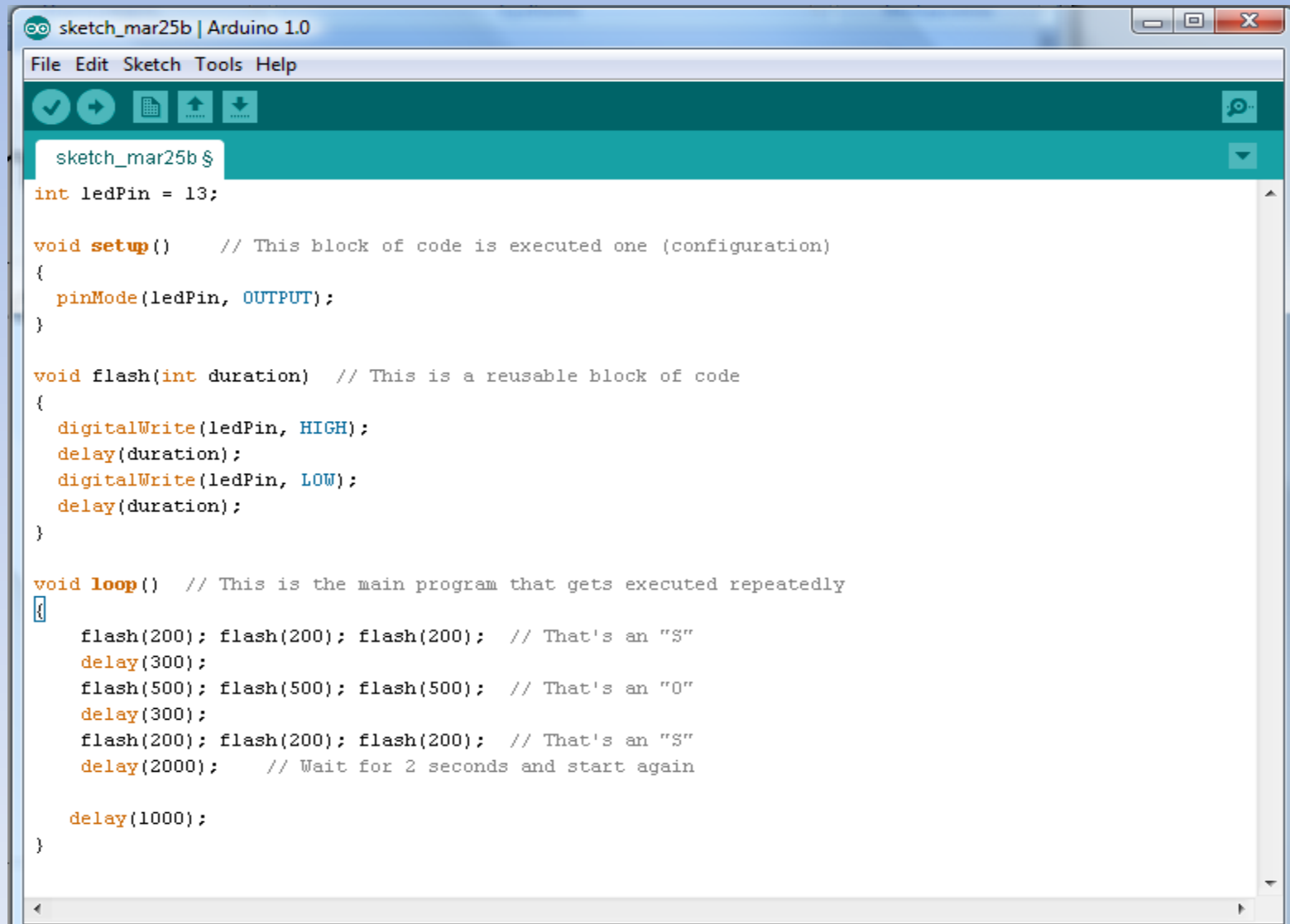
Blink code

A screenshot of the Arduino IDE interface. The title bar reads "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for checking, running, saving, and uploading. A tab labeled "Blink" is active. The main text area contains the following code:

```
/*  
  Blink  
  Turns on an LED on for one second, then off for one second, repeatedly.  
  
  This example code is in the public domain.  
  */  
  
void setup() {  
  // initialize the digital pin as an output.  
  // Pin 13 has an LED connected on most Arduino boards:  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  // set the LED on  
  delay(1000);             // wait for a second  
  digitalWrite(13, LOW);   // set the LED off  
  delay(1000);             // wait for a second  
}
```

The bottom status bar shows "1" on the left and "Arduino Mega 2560 or Mega ADK on COM16" on the right.

The S.O.S. example

The image shows a screenshot of the Arduino IDE interface. The title bar at the top reads "sketch_mar25b | Arduino 1.0". Below the title bar is a menu bar with "File", "Edit", "Sketch", "Tools", and "Help". Underneath the menu bar is a toolbar with icons for checking, running, saving, and uploading. The main text area contains the following C++ code:

```
sketch_mar25b $
int ledPin = 13;

void setup()    // This block of code is executed one (configuration)
{
  pinMode(ledPin, OUTPUT);
}

void flash(int duration)  // This is a reusable block of code
{
  digitalWrite(ledPin, HIGH);
  delay(duration);
  digitalWrite(ledPin, LOW);
  delay(duration);
}

void loop()    // This is the main program that gets executed repeatedly
{
  flash(200); flash(200); flash(200);  // That's an "S"
  delay(300);
  flash(500); flash(500); flash(500);  // That's an "O"
  delay(300);
  flash(200); flash(200); flash(200);  // That's an "S"
  delay(2000);    // Wait for 2 seconds and start again

  delay(1000);
}
```

Κάναμε ένα LED να αναβοσβήνει.

So what??

- Χρησιμοποιήσαμε μια ψηφιακή έξοδο (ενεργοποίηση / απενεργοποίηση). Με τον ίδιο τρόπο μπορούμε να ελέγξουμε άλλες συσκευές (π.χ. κινητήρες)
- Η εντολή `digitalWrite()` είναι το τμήμα του προγράμματος που μας επιτρέπει να ελέγξουμε σχεδόν οτιδήποτε
- Το Arduino μπορεί να διαθέτει πάρα πολλούς ψηφιακούς ακροδέκτες εξόδου

Πλήθος έτοιμων παραδειγμάτων

File → Examples:

Ψηφιακά & αναλογικά σήματα, επικοινωνία, έλεγχος, αισθητήρες, οθόνες, κινητήρες, δίκτυο, κ.α.

Επιπλέον:

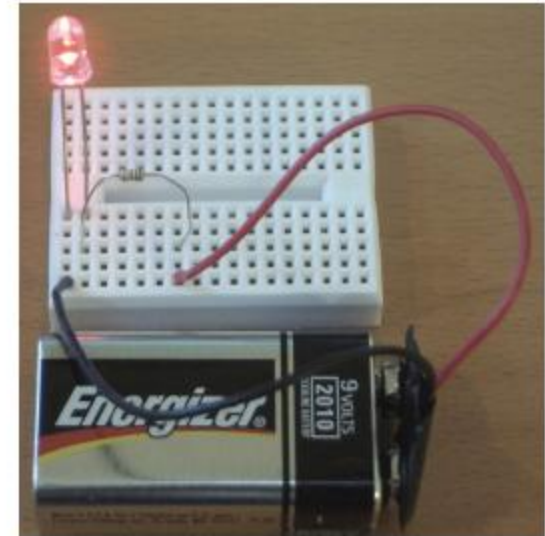
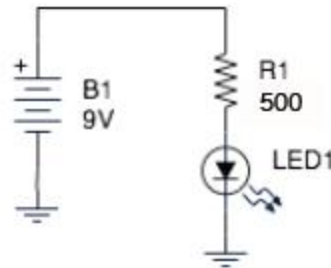
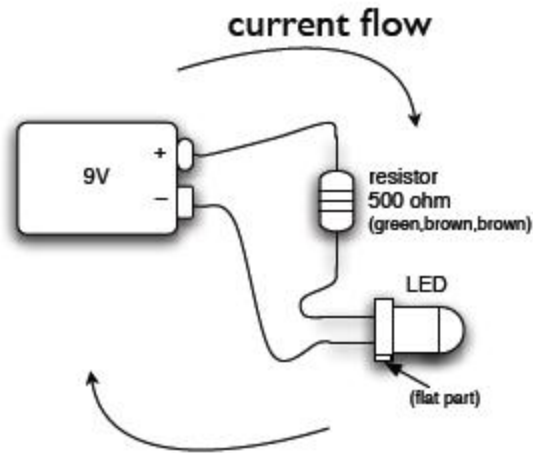
<http://www.arduino.cc/en/Tutorial/HomePage>

Google keywords:

- “Arduino tutorials”,
- “Arduino projects”,
- “Arduino examples”

Γρήγορη δημιουργία ηλεκτρικών κυκλωμάτων

- Αντιστάτες, LEDs, διακόπτες και breadboards

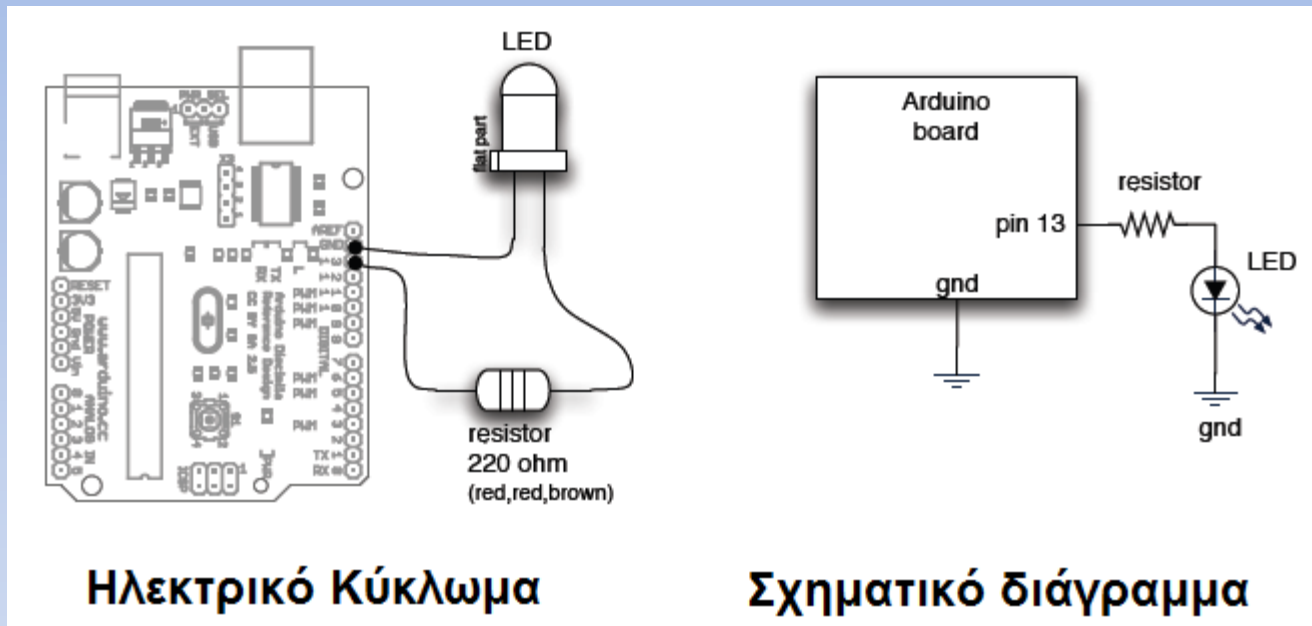


Ηλεκτρικό κύκλωμα

Σχηματικό Διάγραμμα

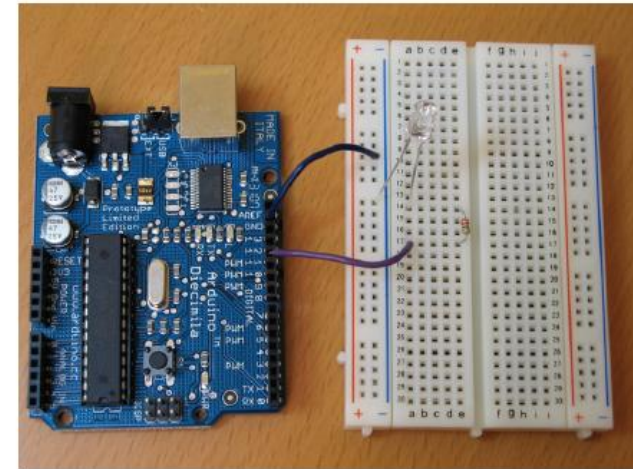
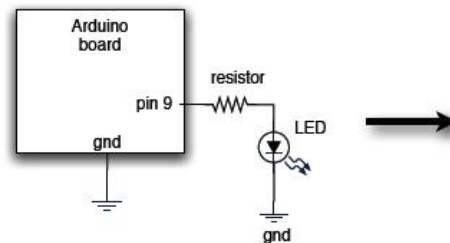
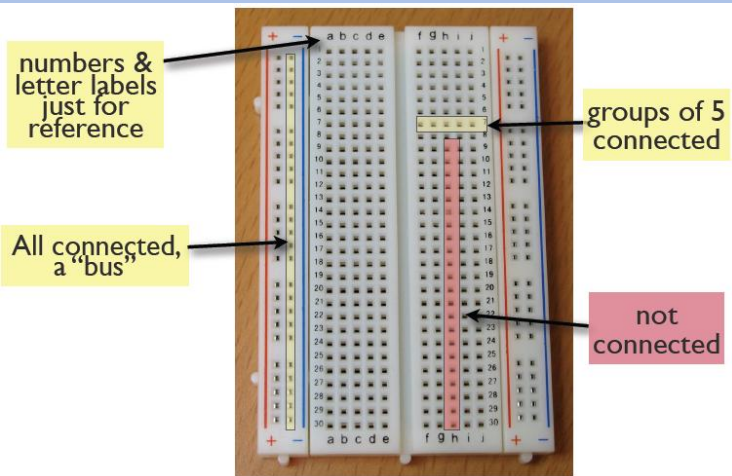
Υλοποίηση

Παρεμβάλλοντας το Arduino



- Το Arduino χρησιμοποιείται:
 - Για την παροχή ηλεκτρικής ισχύος
 - Για τη διακοπή λειτουργίας του LED
- Οι σχηματικές αναπαραστάσεις χρησιμοποιούνται περισσότερο από τα ηλεκτρικά κυκλώματα

Breadboards για γρήγορες δοκιμές

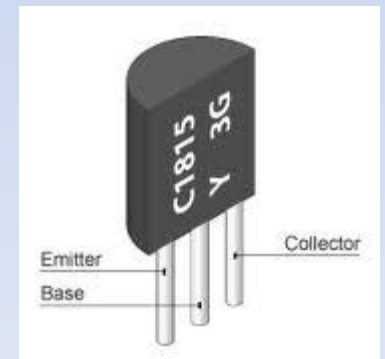
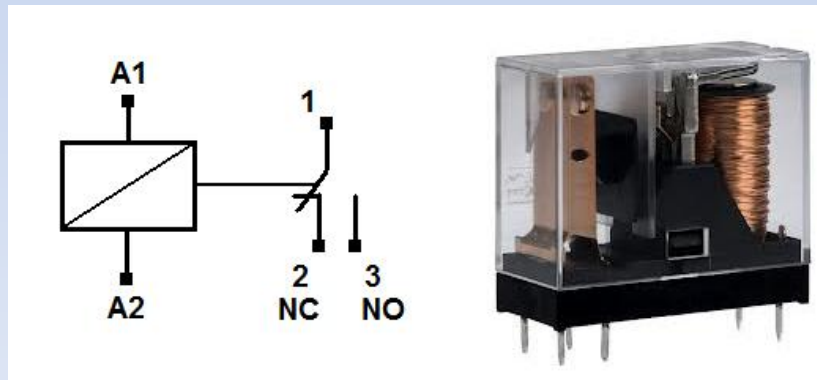


Από το σχηματικό διάγραμμα στην υλοποίηση του κυκλώματος

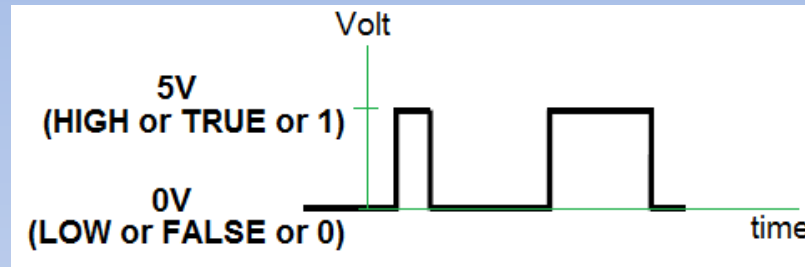
- Ταχύτατη δημιουργία ηλεκτρικών κυκλωμάτων
- Δεν χρειάζονται κολλήσεις
- Πάρα πολλά ηλεκτρονικά υλικά μπορούν να τοποθετηθούν σε breadboards

Διακόπτες και Ψηφιακές Είσοδοι/ Έξοδοι

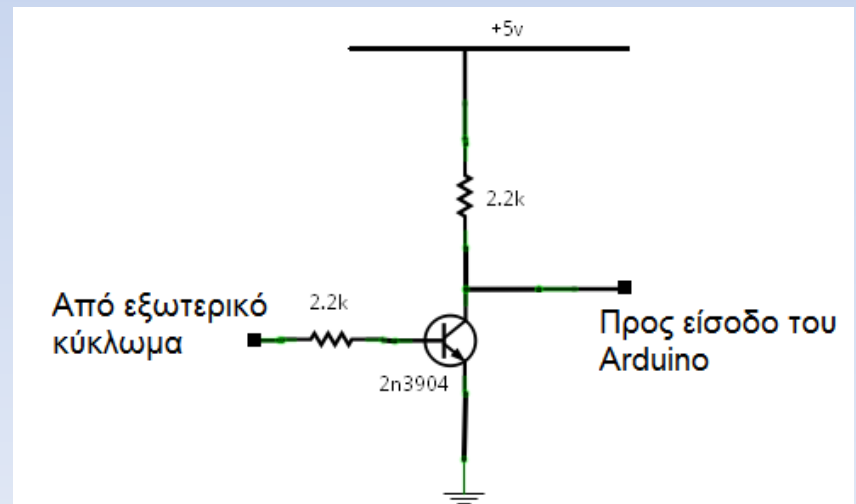
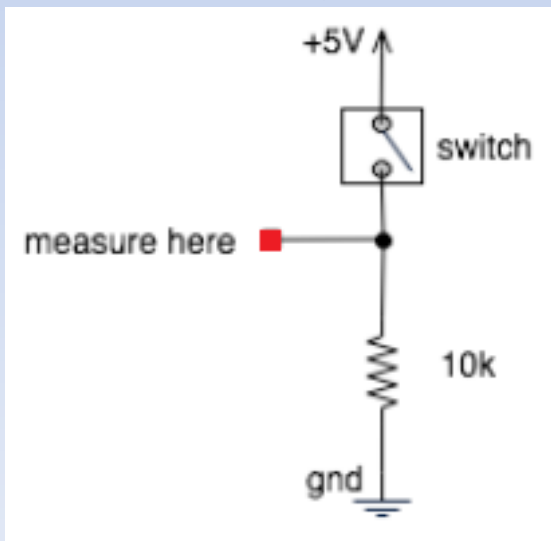
- Οι διακόπτες χρησιμοποιούνται για να επιτρέψουν ή να παρεμποδίσουν τη ροή ηλεκτρικών σημάτων (τάσης ή ρεύματος) από/προς τους ακροδέκτες του Arduino
- Μπορούν να είναι ελεγχόμενοι (από άνθρωπο ή από άλλο ηλεκτρικό κύκλωμα, π.χ. relays)
- Στις ψηφιακές εισόδους του, το Arduino περιμένει να δει υψηλή (5V) ή χαμηλή (0V) τάση.
- Στις ψηφιακές εξόδους του, το Arduino θέτει τους ακροδέκτες σε υψηλή (5V) ή χαμηλή (0V) τάση.



Ψηφιακές είσοδοι/Digital Inputs



- Στο αριστερό σχήμα, όταν ο διακόπτης είναι ανοικτός, η ψηφιακή είσοδος δένεται μέσω της αντίστασης στο μηδέν (LOW)
- Όταν κλείνει ο διακόπτης, τα +5V «σπρώχνονται» στην ψηφιακή είσοδο (HIGH)
- Ένα transistor μπορεί να χρησιμοποιηθεί ως ηλεκτρονικός διακόπτης
- Στο δεξί σχήμα το κύκλωμα λειτουργεί στην ακριβώς αντίστροφη λογική



Διάλειμα – Ελεύθερη συζήτηση

Μέρος 2ο:

The Arduino language

- Η γλώσσα που χρησιμοποιείται είναι standard C
- Διατίθενται πολλές χρήσιμες συναρτήσεις
 - `pinMode()`: θέτει ένα pin ως είσοδο ή έξοδο
 - `digitalWrite()`: θέτει την τιμή ενός pin σε “high” ή “low”
 - `digitalRead()`: διαβάζει την τιμή μιας ψηφιακής εισόδου
 - `analogRead()`: διαβάζει την τιμή μιας αναλογικής εισόδου
 - `analogWrite()`: θέτει την τιμή σε μια αναλογική έξοδο
 - `delay()`: καθυστερεί την εκτέλεση του προγράμματος για ορισμένο χρόνο
 - `millis()`: επιστρέφει τον χρόνο που πέρασε από την έναρξη του προγράμματος
- Υπάρχουν βιβλιοθήκες (συλλογές προγραμμάτων) που παρέχουν πολλές ακόμα συναρτήσεις

Η δομή ενός προγράμματος

- Οι (γενικές) μεταβλητές δηλώνονται στην αρχή
- Αρχικοποιούμε το πρόγραμμα
 - `setup()`: εκτελείται 1 φορά, πριν το κυρίως πρόγραμμα
- Εκτελούμε το πρόγραμμα, επαναληπτικά
 - `loop()`: όλες οι εντολές που υπάρχουν σ' αυτό το τμήμα εκτελούνται διαδοχικά. Όταν εκτελεστεί η και η τελευταία, ο κύκλος επαναλαμβάνεται.

Πρόγραμμα Ανάγνωσης Ψηφιακής Εισόδου

Χρησιμοποιούμε τις εντολές

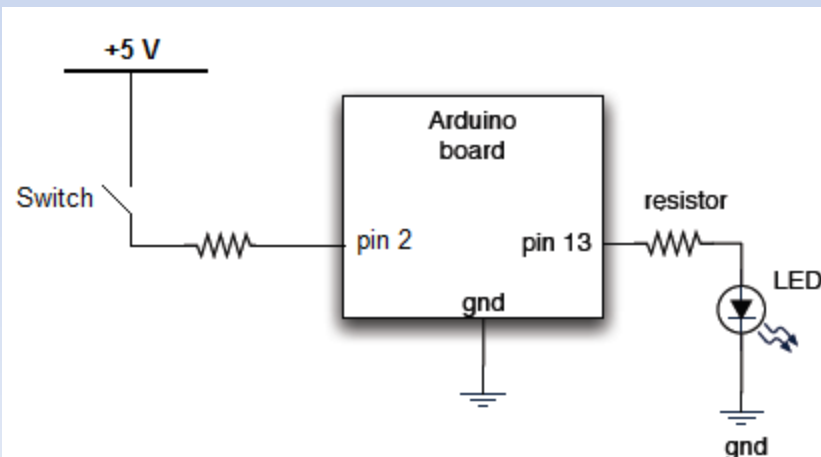
- pinMode()
- digitalRead()
- digitalWrite()

Τώρα μπορούμε να ελέγχουμε πότε ανάβει το LED

```
int ledPin = 13;           // choose the pin for the LED
int inputPin = 2;          // choose the input pin (for a pushbutton)
int val = 0;               // variable for reading the pin status

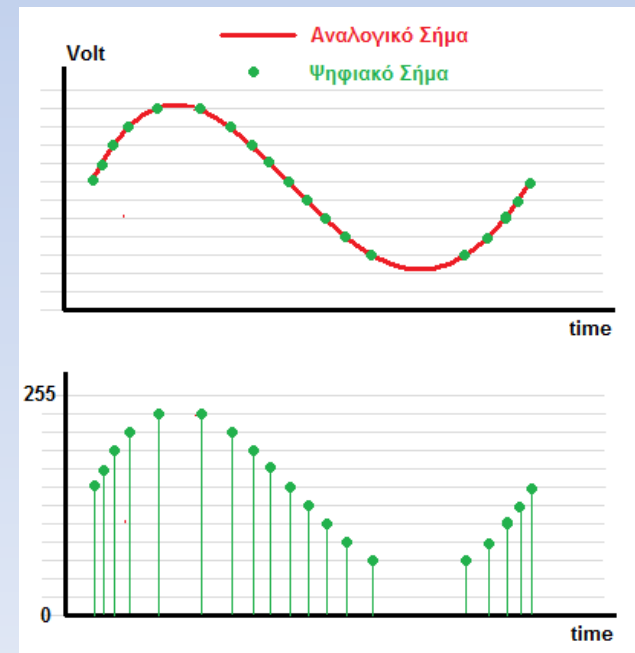
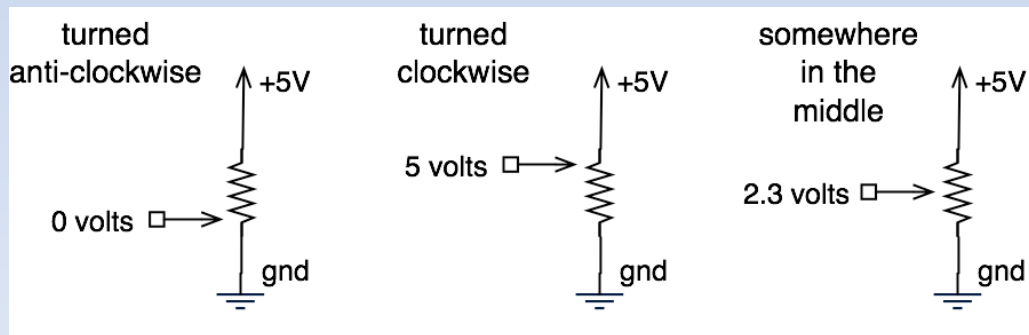
void setup() {
  pinMode(ledPin, OUTPUT); // declare LED as output
  pinMode(inputPin, INPUT); // declare pushbutton as input
}

void loop(){
  val = digitalRead(inputPin); // read input value
  if (val == HIGH) {           // check if the input is HIGH
    digitalWrite(ledPin, LOW); // turn LED OFF
  } else {
    digitalWrite(ledPin, HIGH); // turn LED ON
  }
}
```



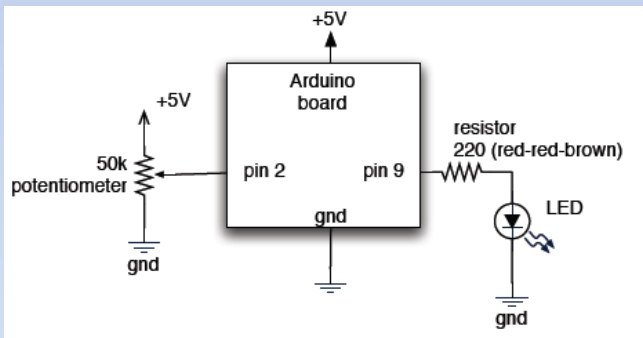
Αναλογικά σήματα εισόδου

- Πολλές διακριτές καταστάσεις (όχι μόνον δύο, π.χ. LOW/HIGH)
- Διακριτική ικανότητα (resolution) είναι το πλήθος των διακριτών καταστάσεων (π.χ. $2^8 = 256$)
- Το Arduino «διαβάζει» εξωτερικές τάσεις από 0 έως +5V
- Η διακριτική του ικανότητα είναι 1024 στάθμες (10bit resolution), δηλαδή αντιλαμβάνεται μεταβολές της τάξεως των $(5-0)/1024 = 4.8\text{mV}$



Πρόγραμμα ανάγνωσης αναλογικής εισόδου

Ρυθμίζουμε το χρόνο κατά τον οποίο το LED παραμένει αναμμένο



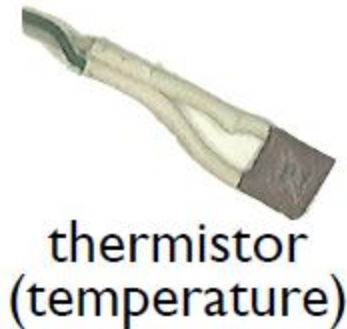
```
/*
  Analog Input
  Demonstrates analog input by reading an analog sensor on analog pin 2 and
  turning on and off a light emitting diode(LED) connected to digital pin 9.
  The amount of time the LED will be on and off depends on the value obtained
  by analogRead().
  */

int sensorPin = A2;    // select the input pin for the potentiometer
int ledPin = 9;        // select the pin for the LED
int sensorValue = 0;    // variable to store the value coming from the sensor

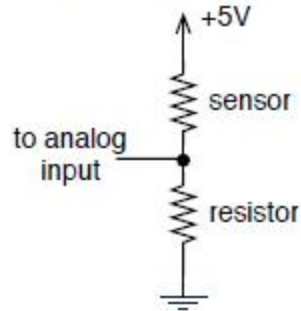
void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  sensorValue = analogRead(sensorPin); // read the value from the sensor
  digitalWrite(ledPin, HIGH);          // turn the ledPin on
  delay(sensorValue);                  // stop the program for <sensorValue> milliseconds
  digitalWrite(ledPin, LOW);           // turn the ledPin off
  delay(sensorValue);                  // stop the program for for <sensorValue> milliseconds
}
```

Άλλοι ωμικοί (παθητικοί) αισθητήρες



circuit is the same
for all these



photocell
(light)



force sensors
(pressure)

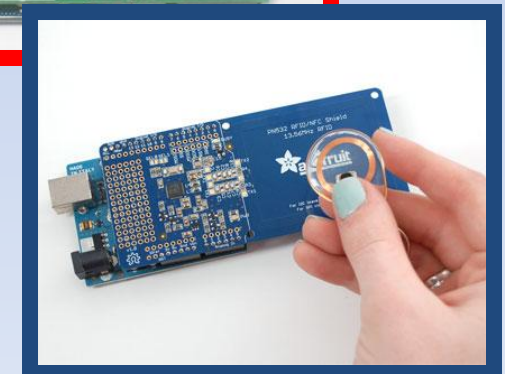
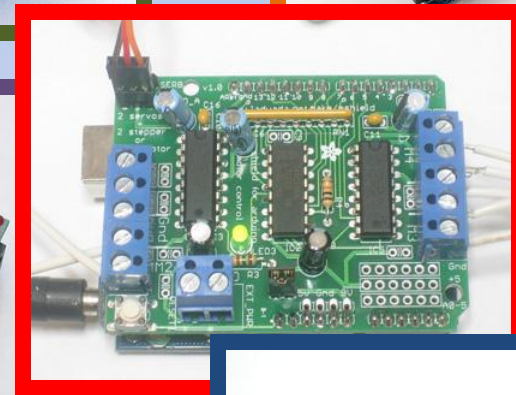
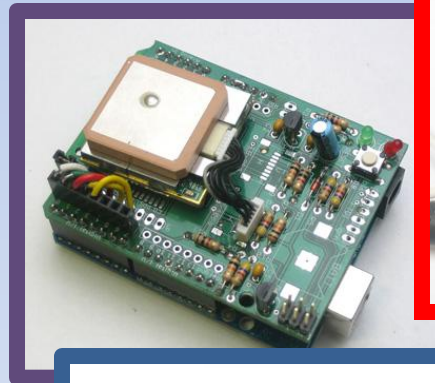
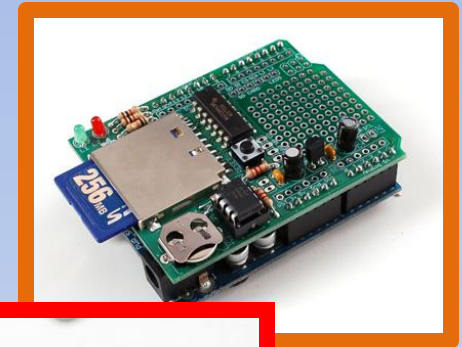


flex sensor
(bend, deflection)

also air pressure
and others

Arduino: Επεκτάσεις & Βιβλιοθήκες

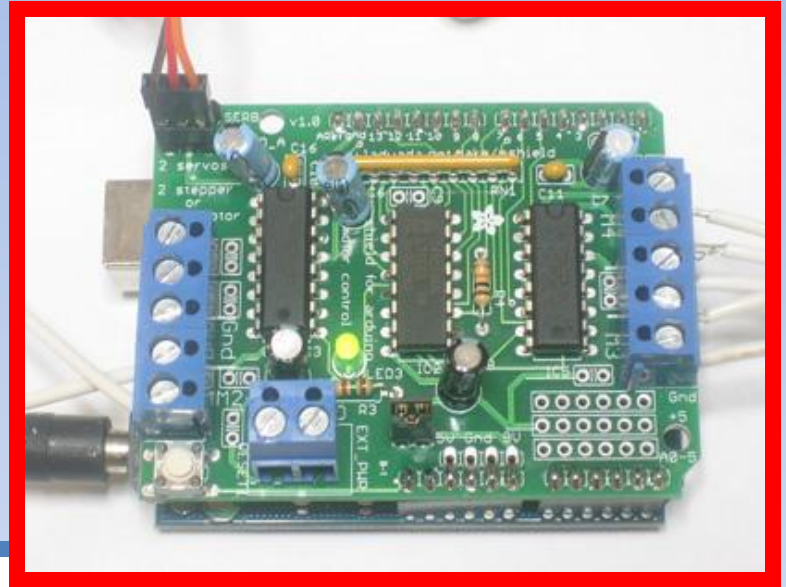
- Κινητήρες
- Οθόνες
- Δίκτυο Ethernet
- WiFi
- RFID
- Κινητό GSM
- Κάρτα μνήμης
- ...



Κάρτα οδήγησης κινητήρων

Επιτρέπει την οδήγηση εξειδικευμένων κινητήρων που μας επιτρέπουν να ορίσουμε την ταχύτητα αλλά και την θέση περιστροφής. Τέτοια παραδείγματα κινητήρων είναι οι:

- βηματικοί κινητήρες (steppers) που χρησιμοποιούνται σε ρομποτικές εφαρμογές και εργαλειομηχανές. Περιστρέφονται σε βήματα με ρυθμιζόμενη ταχύτητα
- σερβο-κινητήρες (servos) που χρησιμοποιούνται σε μοντελισμό. Περιστρέφονται με σταθερή ταχύτητα μέχρι ο άξονάς τους να διαγράψει καθορισμένο τόξο.



Η γλώσσα Processing

<http://www.processing.org/>

- Η γλώσσα προγραμματισμού **Processing** χρησιμοποιείται για τη δημιουργία εικόνων, animations και αλληλεπιδράσεων ανθρώπου-υπολογιστή.
- Προσφέρει ευελιξία, καθώς παρέχει πολλά εργαλεία τα οποία μπορούν να συνεργαστούν για να δημιουργήσουν μια νέα εμπειρία
- Μπορεί να συνδυάσει εικόνες, κείμενο, ήχους, animations, γραφικά. Επίσης, μπορεί να αντλήσει πληροφορίες από το πληκτρολόγιο, το ποντίκι, άλλους Η/Υ.
- Το ολοκληρωμένο περιβάλλον σχεδίασης (**P**rocessing **D**esign **E**nvironment) χρησιμοποιείται για τη δημιουργία νέων προγραμμάτων (sketches) και μοιάζει με το περιβάλλον του Arduino.
- Θα χρησιμοποιήσουμε το Arduino για να στείλουμε δεδομένα στο PDE

Το ολοκληρωμένο περιβάλλον σχεδίασης (PDE)

Το περιβάλλον είναι παρόμοιο με αυτό του Arduino

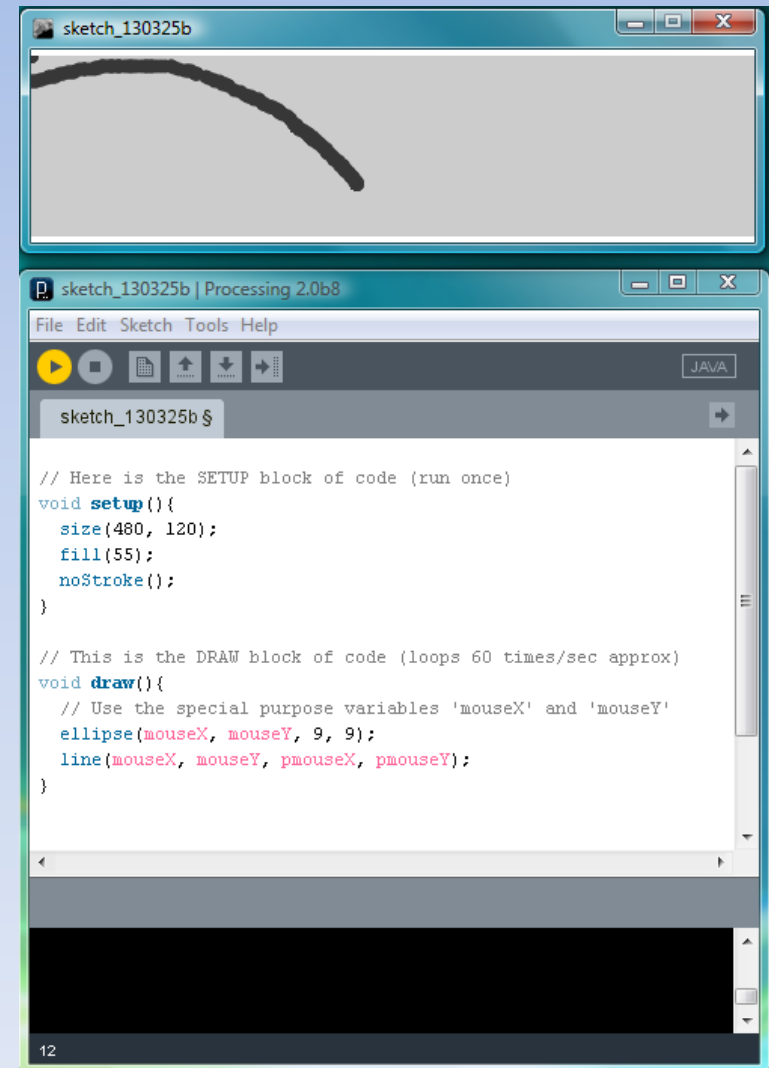
1. Γράφουμε τον κώδικα
2. Ελέγχουμε για σφάλματα
3. Εκτελούμε τον κώδικα

Δείτε το Wiki:

http://wiki.processing.org/w/Main_Page

και το forum:

<http://forum.processing.org/>

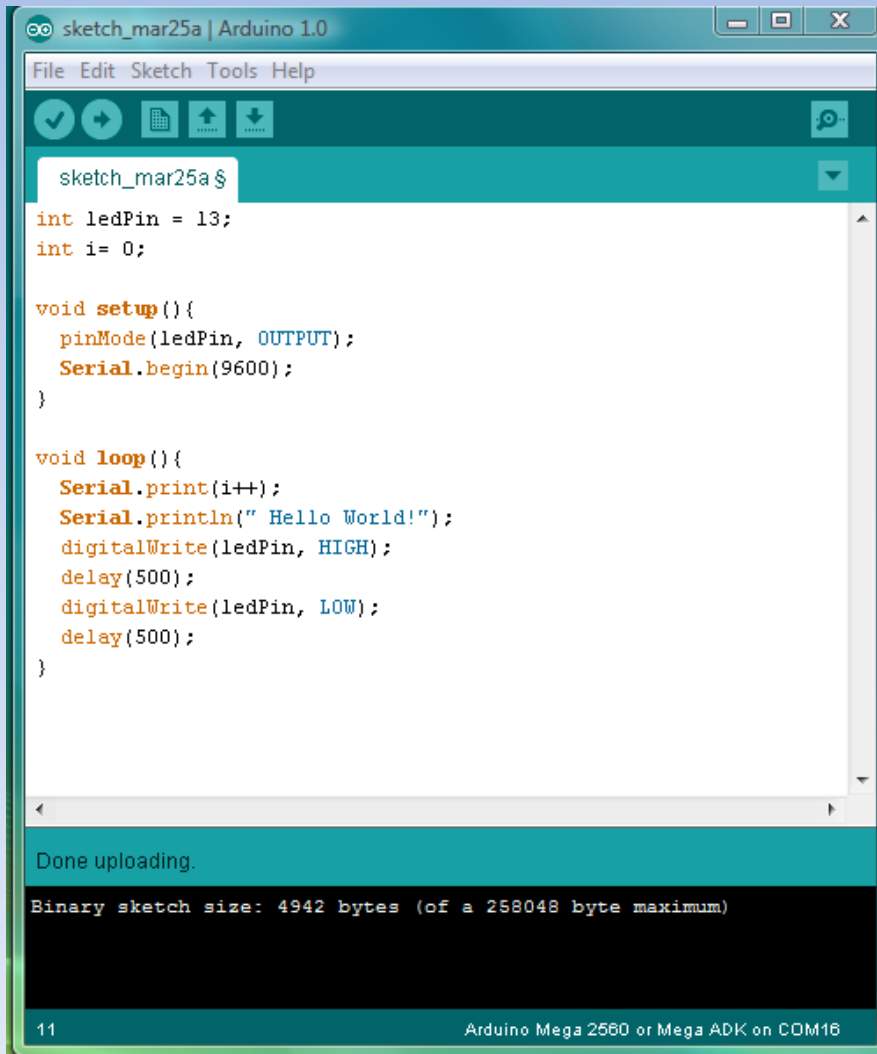


Επικοινωνία με το Arduino

- Μπορούμε να χρησιμοποιήσουμε τη θύρα USB για ενσύρματη, σειριακή επικοινωνία
- Χρησιμοποιούμε την ομάδα εντολών 'Serial'
(<http://arduino.cc/en/Reference/Serial>):
 - Serial.available(...) : έλεγχος για διαθεσιμότητα σειριακής θύρας
 - Serial.begin(...) : προετοιμασία της θύρας για χρήση
 - Serial.end(...) : αποδέσμευση σειριακής θύρας

 - Serial.println(...) : αποστολή δεδομένων στη θύρα
 - Serial.read(...) : παραλαβή δεδομένων απ' τη θύρα

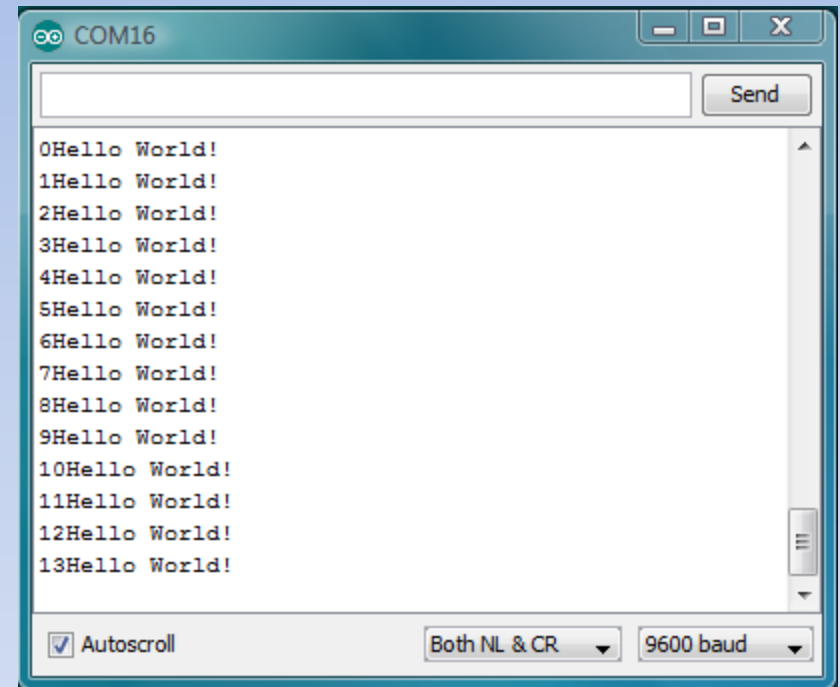
Το Arduino λέει: «Γεια!»



```
sketch_mar25a$  
int ledPin = 13;  
int i= 0;  
  
void setup(){  
  pinMode(ledPin, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop(){  
  Serial.print(i++);  
  Serial.println(" Hello World!");  
  digitalWrite(ledPin, HIGH);  
  delay(500);  
  digitalWrite(ledPin, LOW);  
  delay(500);  
}
```

Done uploading.
Binary sketch size: 4942 bytes (of a 258048 byte maximum)

11 Arduino Mega 2560 or Mega ADK on COM16



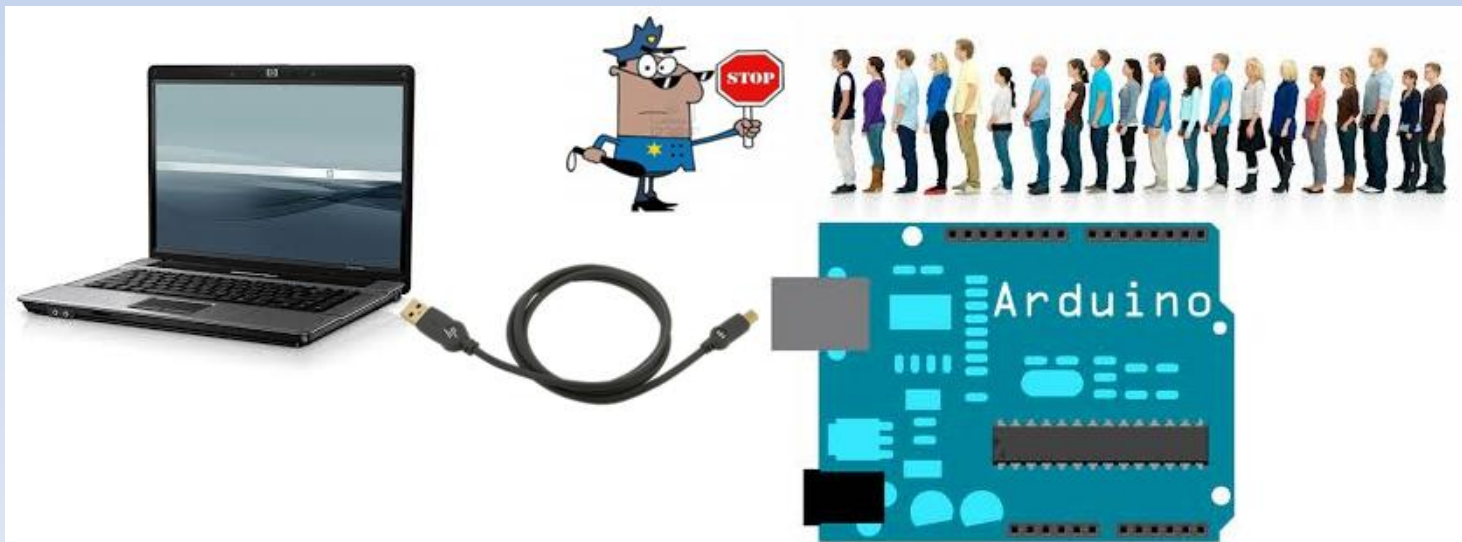
Send

0Hello World!
1Hello World!
2Hello World!
3Hello World!
4Hello World!
5Hello World!
6Hello World!
7Hello World!
8Hello World!
9Hello World!
10Hello World!
11Hello World!
12Hello World!
13Hello World!

☒ Autoscroll Both NL & CR 9600 baud

Η σειριακή επικοινωνία (Rx/Tx)

- Κατά τη σειριακή επικοινωνία, κάθε πληροφορία «σπάει» σε bits τα οποία στέλνονται διαδοχικά από 1 και μόνον καλώδιο.
- Το μήνυμα SOS «σπάει» σε: . . . - - - . . .
- Η επικοινωνία είναι αμφίδρομη. Σε άλλη χρονική στιγμή, το ίδιο καλώδιο χρησιμοποιείται για λήψη.
- Για την αποστολή των bits χρησιμοποιείται η αρχή First In / First Out (FIFO)
- Το Arduino ελέγχει τη ροή των δεδομένων στη σειριακή θύρα
- Απαιτείται προετοιμασία και από την πλευρά του Η/Υ



Ο κώδικας ASCII

American Standard Code for Information Interchange

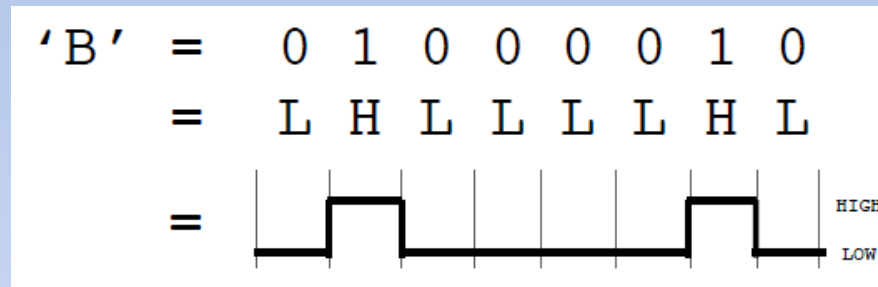
Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	 	Space	64	40	100	@	@	96	60	140	`	`
1	1	001	SOH (start of heading)	33	21	041	!	!	65	41	101	A	A	97	61	141	a	a
2	2	002	STX (start of text)	34	22	042	"	"	66	42	102	B	B	98	62	142	b	b
3	3	003	ETX (end of text)	35	23	043	#	#	67	43	103	C	C	99	63	143	c	c
4	4	004	EOT (end of transmission)	36	24	044	$	\$	68	44	104	D	D	100	64	144	d	d
5	5	005	ENQ (enquiry)	37	25	045	%	%	69	45	105	E	E	101	65	145	e	e
6	6	006	ACK (acknowledge)	38	26	046	&	&	70	46	106	F	F	102	66	146	f	f
7	7	007	BEL (bell)	39	27	047	'	'	71	47	107	G	G	103	67	147	g	g
8	8	010	BS (backspace)	40	28	050	(	(	72	48	110	H	H	104	68	150	h	h
9	9	011	TAB (horizontal tab)	41	29	051	)	)	73	49	111	I	I	105	69	151	i	i
10	A	012	LF (NL line feed, new line)	42	2A	052	*	*	74	4A	112	J	J	106	6A	152	j	j
11	B	013	VT (vertical tab)	43	2B	053	+	+	75	4B	113	K	K	107	6B	153	k	k
12	C	014	FF (NP form feed, new page)	44	2C	054	,	,	76	4C	114	L	L	108	6C	154	l	l
13	D	015	CR (carriage return)	45	2D	055	-	-	77	4D	115	M	M	109	6D	155	m	m
14	E	016	SO (shift out)	46	2E	056	.	.	78	4E	116	N	N	110	6E	156	n	n
15	F	017	SI (shift in)	47	2F	057	/	/	79	4F	117	O	O	111	6F	157	o	o
16	10	020	DLE (data link escape)	48	30	060	0	0	80	50	120	P	P	112	70	160	p	p
17	11	021	DC1 (device control 1)	49	31	061	1	1	81	51	121	Q	Q	113	71	161	q	q
18	12	022	DC2 (device control 2)	50	32	062	2	2	82	52	122	R	R	114	72	162	r	r
19	13	023	DC3 (device control 3)	51	33	063	3	3	83	53	123	S	S	115	73	163	s	s
20	14	024	DC4 (device control 4)	52	34	064	4	4	84	54	124	T	T	116	74	164	t	t
21	15	025	NAK (negative acknowledge)	53	35	065	5	5	85	55	125	U	U	117	75	165	u	u
22	16	026	SYN (synchronous idle)	54	36	066	6	6	86	56	126	V	V	118	76	166	v	v
23	17	027	ETB (end of trans. block)	55	37	067	7	7	87	57	127	W	W	119	77	167	w	w
24	18	030	CAN (cancel)	56	38	070	8	8	88	58	130	X	X	120	78	170	x	x
25	19	031	EM (end of medium)	57	39	071	9	9	89	59	131	Y	Y	121	79	171	y	y
26	1A	032	SUB (substitute)	58	3A	072	:	:	90	5A	132	Z	Z	122	7A	172	z	z
27	1B	033	ESC (escape)	59	3B	073	;	;	91	5B	133	[	[	123	7B	173	{	{
28	1C	034	FS (file separator)	60	3C	074	<	<	92	5C	134	\	\	124	7C	174	|	
29	1D	035	GS (group separator)	61	3D	075	=	=	93	5D	135	]	]	125	7D	175	}	}
30	1E	036	RS (record separator)	62	3E	076	>	>	94	5E	136	^	^	126	7E	176	~	~
31	1F	037	US (unit separator)	63	3F	077	?	?	95	5F	137	_	_	127	7F	177		DEL

Source: www.LookupTables.com

- Το γράμμα 'B' αντιστοιχεί στον δεκαδικό αριθμό 66 = 64 + 2 = 2⁶ + 2¹, δηλαδή σε δυαδική μορφή 8bits γράφεται ως: 0100.0010
- Χρειαζόμαστε 8 bits για την αποστολή ενός μόνον γράμματος

Η σειριακή επικοινωνία

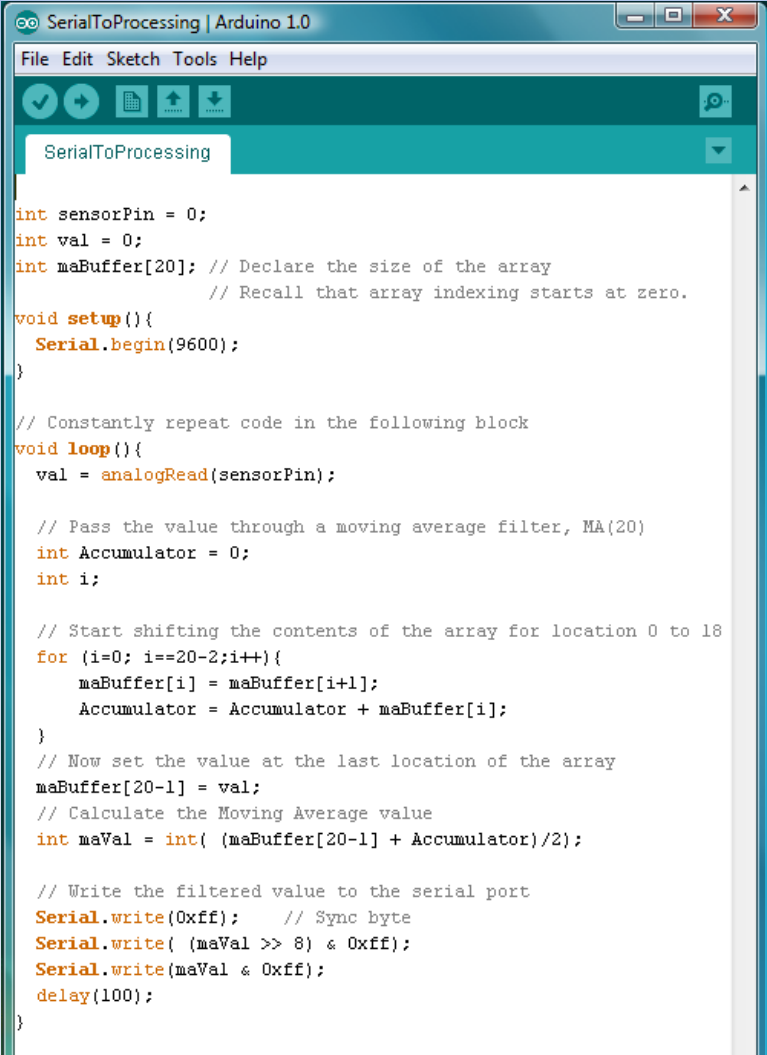
- Κατά τη σειριακή επικοινωνία, μόνον δύο τύποι bits μεταδίδονται: 0 και 1
- Το απλό γράμμα 'B' (βλ. πίνακα ASCII) στέλνεται ως:



- Ο αριθμός 142 ή '1' '4' '2' αντιστοιχεί στους δεκαδικούς '49', '52', '50' και κατ' επέκταση στους δυαδικούς: '0011.0001' '0011.0100' και '0011.0010'
- Η αποστολή μπορεί να ξεκινήσει αυτόματα ή με το πάτημα ενός διακόπτη
- Μπορούμε να υλοποιήσουμε τη σειριακή επικοινωνία με τις εντολές
 - digitalWrite()
 - delay()
- Για λόγους ευκολίας το Arduino το κάνει για εμάς. Εμείς παρέχουμε μόνον το μήνυμα προς μετάδοση

Αποστολή δεδομένων στον Η/Υ

- Ένα πρόγραμμα γραμμένο για το Arduino αναλαμβάνει να στείλει τις πληροφορίες στον υπολογιστή.
- Το πρόγραμμα αυτό εκτελείται διαρκώς (επαναληπτικά)
- Χρησιμοποιούμε ένα φίλτρο απαλοιφής των απότομων μεταβολών (jerky movement), τύπου κινητής μέσης τιμής (**Moving Average**)
- Χρησιμοποιούμε επίσης ένα χαρακτηριστικό σήμα για συγχρονισμό (0xff) της ασύγχρονης μετάδοσης δεδομένων

A screenshot of the Arduino IDE interface. The title bar reads "SerialToProcessing | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening, saving, and running sketches. The main text area displays the code for the "SerialToProcessing" sketch. The code includes variable declarations for sensorPin, val, maBuffer, accumulator, and i. It features a setup function for serial communication and a loop function that implements a moving average filter. The loop function reads an analog value, shifts the contents of the maBuffer array, calculates the moving average, and writes the result to the serial port along with a sync byte (0xff).

```
SerialToProcessing | Arduino 1.0
File Edit Sketch Tools Help

SerialToProcessing

int sensorPin = 0;
int val = 0;
int maBuffer[20]; // Declare the size of the array
                  // Recall that array indexing starts at zero.

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

// Constantly repeat code in the following block
void loop(){
  val = analogRead(sensorPin);

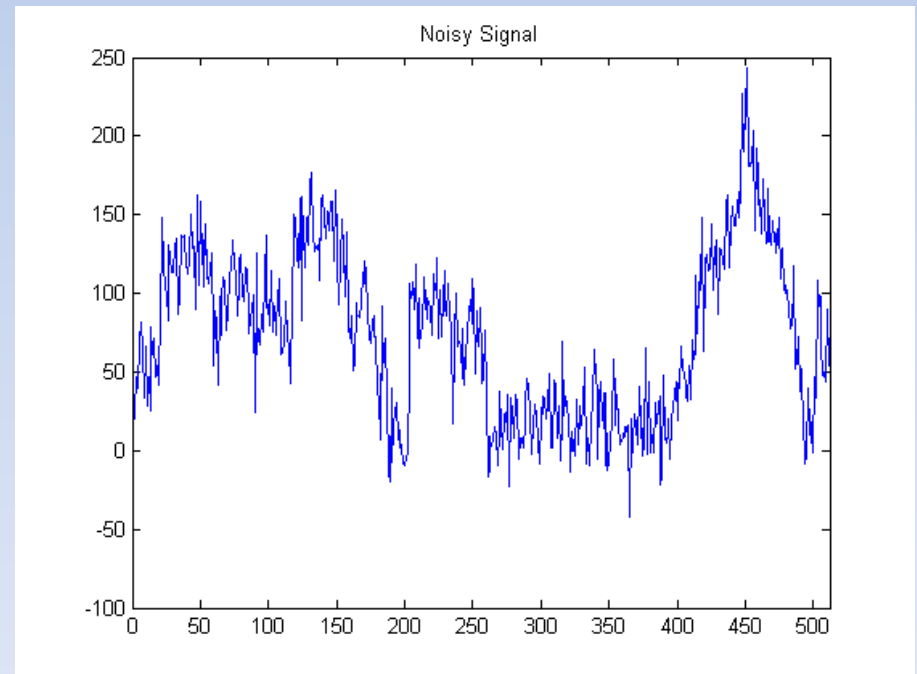
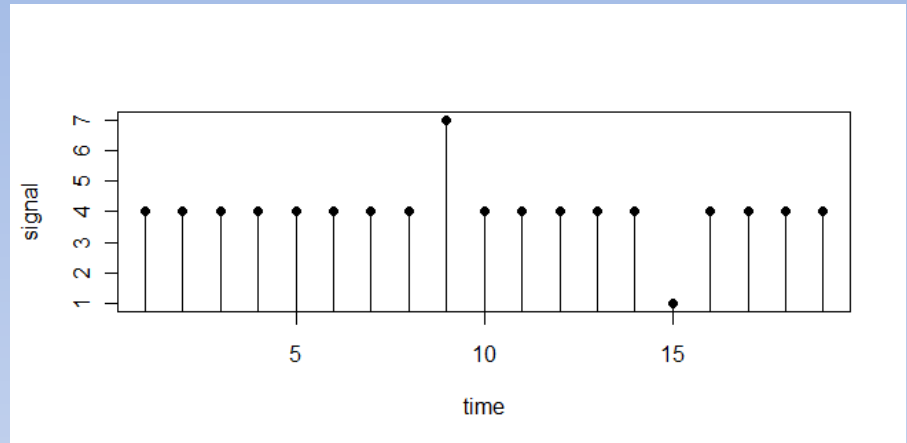
  // Pass the value through a moving average filter, MA(20)
  int Accumulator = 0;
  int i;

  // Start shifting the contents of the array for location 0 to 18
  for (i=0; i==20-2;i++){
    maBuffer[i] = maBuffer[i+1];
    Accumulator = Accumulator + maBuffer[i];
  }
  // Now set the value at the last location of the array
  maBuffer[20-1] = val;
  // Calculate the Moving Average value
  int maVal = int( (maBuffer[20-1] + Accumulator)/2);

  // Write the filtered value to the serial port
  Serial.write(0xff); // Sync byte
  Serial.write( (maVal >> 8) & 0xff);
  Serial.write(maVal & 0xff);
  delay(100);
}
```

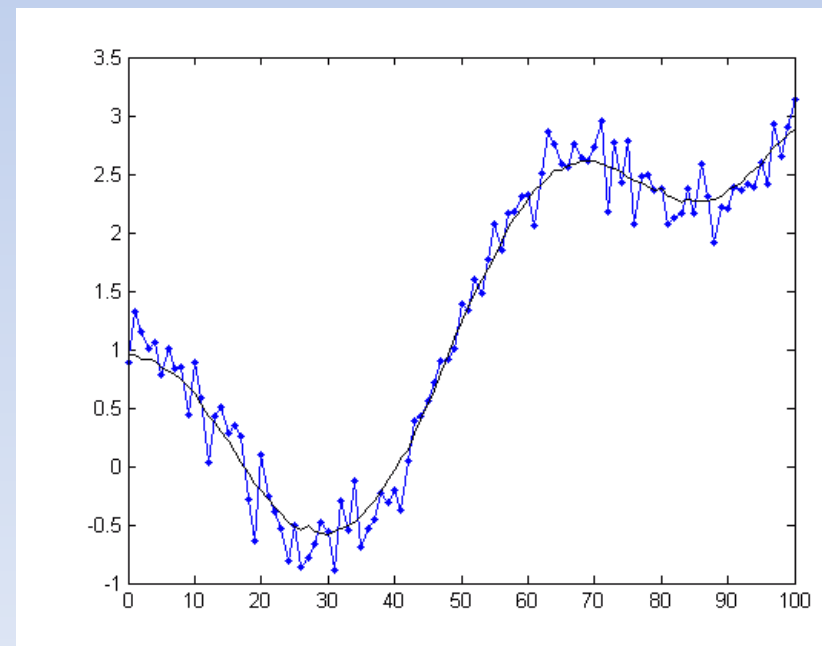
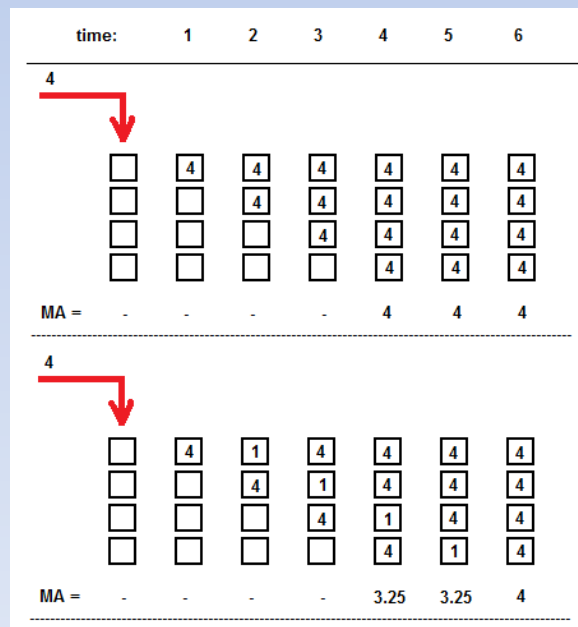
Θορυβώδη σήματα

- **Θόρυβος** είναι κάθε σήμα το οποίο επικάθεται στο αρχικό σήμα και το διαφοροποιεί κατά τρόπο ανεπιθύμητο
- Η παρουσία του είναι αναπόφευκτη. Η ένταση του προβλήματος εξαρτάται από τις συνθήκες.
- Τα θορυβώδη δεδομένα χαρακτηρίζονται συνήθως ως **RAW** data
- Τα απαλλαγμένα από θόρυβο σήματα χαρακτηρίζονται ως **FILTERED** data

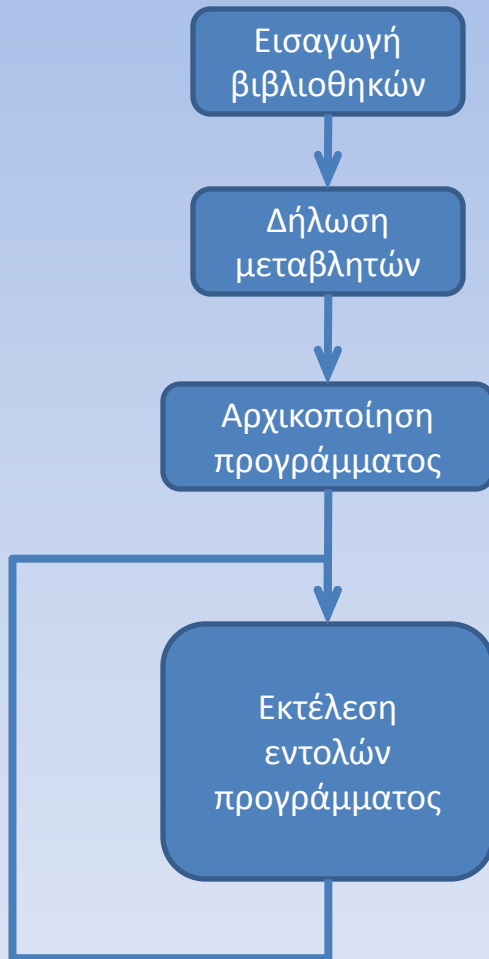


Φίλτρα κινητής μέσης τιμής

- Ένα είδος χαμηλοπερατού φίλτρου το οποίο «φιμώνει» τις απότομες μεταβολές σήματος
- Εξίσωση μοντέλου: $SMA[t] = (x[t] + x[t-1] + \dots + x[t-(K-1)]) / K$
- Μπορούμε να καθορίσουμε το μέγεθος του buffer, K
- Δείτε επίσης: http://en.wikipedia.org/wiki/Moving_average



«Διαβάζοντας» το ποτενσιόμετρο



```
import processing.serial.*;
Serial port;

// Declare GLOBAL variables
int padW; int padH; // set the paddle Width and Height
float ypos; // store the y-coordinate of the Upper-Left corner of the paddle
int gap; // define the distance of paddle from the left wall
float pot; // Use a variable to store the potentiometer value

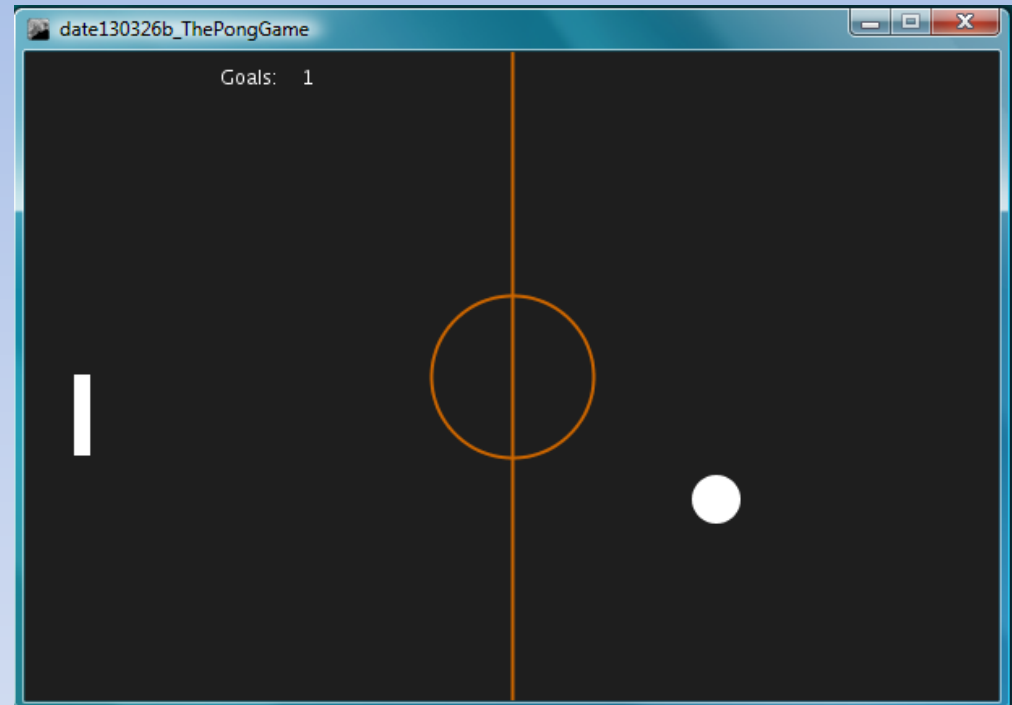
// Initialization
void setup(){
  size(600, 400); // Set the frame size
  ellipseMode(CENTER); smooth(); // Initialize graphics settings
  // Initialize Serial Communication interface (set port number and data rate)
  String arduinoPort = Serial.list()[7];
  port = new Serial(this, arduinoPort, 9600);
  gap = 50; // Give the gap a value
  padW = 10; padH = 50; // set the paddle size
}

// Main program
void draw(){
  background(30);
  // Get the paddle position from potentiometer
  if(port.available() > 0){
    if (port.read() == 0xff){pot = (port.read() << 8) | (port.read());}
  }
  if (pot != -1) { // We have valid data from potentiometer
    ypos = int(pot); // Read-in the data
    ypos = map(ypos, 0, 51, 0, height-padH); // Map the data to a corresponding scale
  }

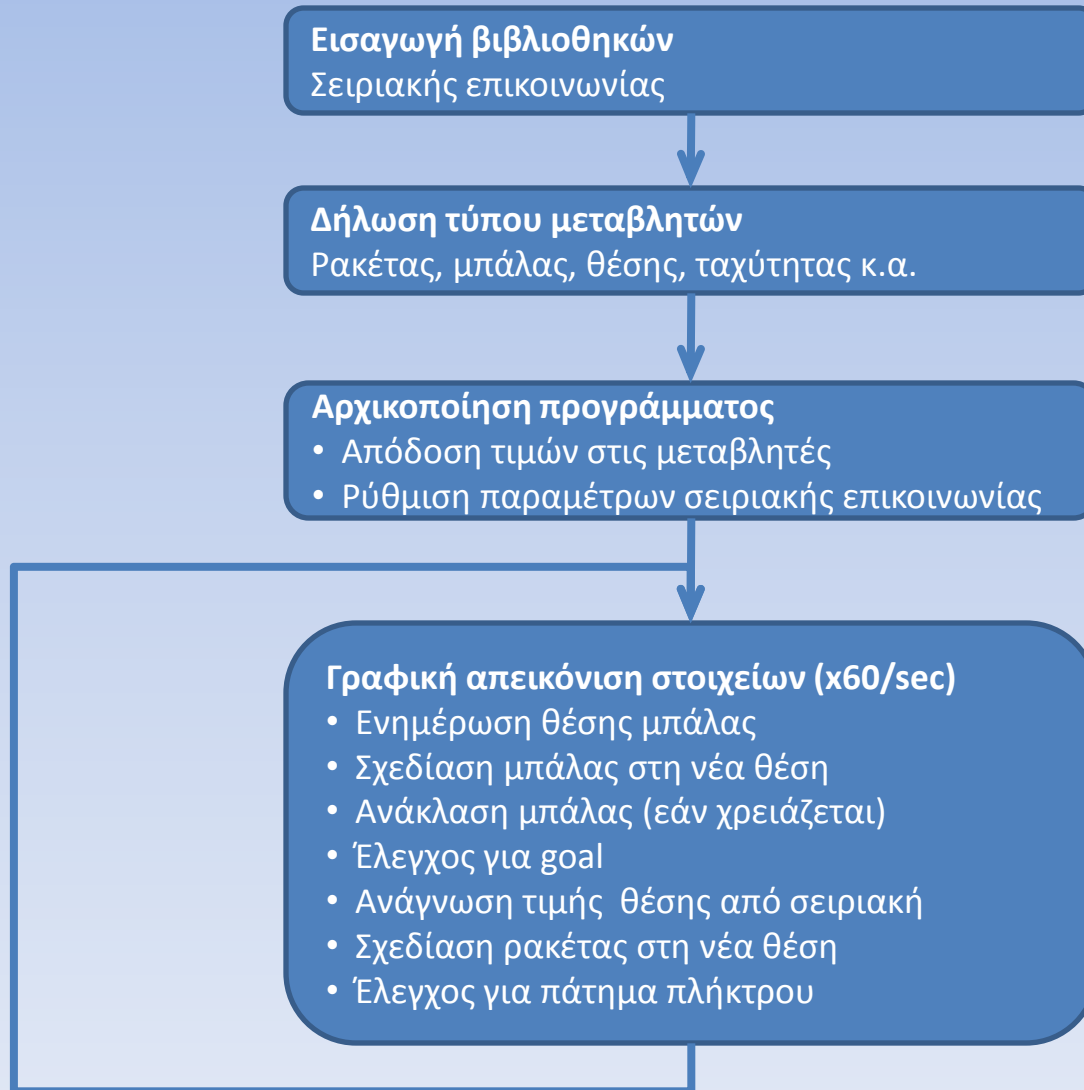
  // Finally, draw the paddle
  rect(gap, ypos, 10, 50);
}
```

Ardu-Pong: the game

- Αντικείμενα:
 - Μπάλα
 - Ρακέτα
- Η μπάλα δεν πρέπει να περάσει την ρακέτα
- Η μπάλα ανακλάται στα τοιχώματα και στη ρακέτα
- Η ρακέτα ελέγχεται από τον χρήστη με το ποτενσιόμετρο
- Το Arduino στέλνει τις πληροφορίες θέσης στο περιβάλλον της Processing, για απεικόνιση



ArduPong: ο κώδικας



ArduPong: ο κώδικας

```
/**
 * A pong game using Arduino and Processing
 *
 * This example demonstrates the use of Processing to create animations as well as the
 * use of the 'serial' package to read-in values from a serial port. In particular,
 * these values come from an Arduino board and correspond to the position (resistance value)
 * of a potentiometer. We use this value (pot) to update the position of a paddle on our
 * screen.
 *
 * Spyros Veronikis (spyros.veronikis@gmail.com), March 2013
 */

import processing.serial.*;
Serial port;

// Declare GLOBAL variables
int padW; int padH;
float ypos;
float ball_x; float ball_y; float ballvel_x; float ballvel_y; int diameter;

boolean play;
int goals; int gap;
int pot; // Use a variable to store the potentiometer value

// User-defined functions
void ball_init(){
  ball_x = int(width/2);  ball_y = int(height/2);  // Set the ball position at start
  // Set the initial ball velocity
  ballvel_x = int(random(0.2, 0.5)*10);  ballvel_y = int(random(0.2, 0.3)*10);
}
```


ArduPong: ο κώδικας

```
// Initialization
void setup(){
  size(600, 400);           // Set the frame size
  ellipseMode(CENTER); smooth(); // Initialize graphics settings
  // Initialize Serial Communication interface (set port number and data rate)
  String arduinoPort = Serial.list()[7];
  port = new Serial(this, arduinoPort, 9600);
  goals = 0; play = true;   // Set the initial game values
  gap = 50;
  padW = 10; padH = 50;    // Initialize paddle values
  ball_init(); diameter = 30; // Initialize the ball position and diameter
}

// Main program
void draw(){
  background(30);
  textSize(12);
  text("Goals: ", 120, 20); text(goals, 170, 20);
  stroke(204, 102, 0); line(width/2, 0, width/2, height);
  strokeWeight(2); noFill(); ellipse(width/2, height/2, 100, 100);
  fill(255); noStroke();

  if(play){
    // Update ball position
    ball_x = ball_x + ballvel_x;
    ball_y = ball_y + ballvel_y;

    // Reflect off bottom and top walls
    if ((ball_y >= height - diameter/2) || (ball_y < 0+diameter/2)){
      ballvel_y = -ballvel_y; }

    if (ball_x >= width - diameter/2){
      ballvel_x = -ballvel_x; }
  }
}
```

ArduPong: ο κώδικας

```
if (ball_x < gap + padW + diameter/2){
  if ((abs((ypos+padH/2)-ball_y) < 0.75*padH) && (ball_x > gap+padW-3) ){
    ballvel_x = -1.20 * ballvel_x;    // That's a physics law violation! (increase speed 20%)
  }
}
if (ball_x < 0 ){  play = false;    // Stop game
                  goals++;         // Increment goals taken
}

ellipse(ball_x, ball_y, diameter, diameter); // Draw the ball

// Get the paddle position from potentiometer
if(port.available() > 0){
  if (port.read() == 0xff){pot = (port.read() << 8) | (port.read());}
}
if (pot != -1) {          // We have valid data from potentiometer
  ypos = pot;             // Read-in the data
  ypos = map(ypos, 0, 511, 0, height-padH); // Map the data to a corresponding scale
}

} else {  textSize(36);          // if game is not in play...
          text("Press 'c' to continue", 120, 200);
          text("or 'r' to restart!", 120, 240);
        }
}

if (keyPressed){
  switch(key){
    case 'c':
      play = true; ball_init(); break;
    case 'r':
      play = true; ball_init(); goals = 0; break;
  }
}
// Finally, draw the paddle
rect(gap, ypos, 10, 50);
}
```

Περίληψη

- Το Arduino είναι μια ανοικτή πλατφόρμα προγραμματισμού δηλαδή μπορούμε ελεύθερα να το αντιγράψουμε, τροποποιήσουμε και να το διανέμουμε
- Ο προγραμματισμός γίνεται με τυπική C, από ολοκληρωμένο περιβάλλον
- Διαθέτει μικροεπεξεργαστή ο οποίος έχει ψηφιακές και αναλογικές εισόδους/εξόδους
- Ο μικροεπεξεργαστής μπορεί να προγραμματιστεί για να εκτελεί μια ορισμένη λογική ελέγχου (εάν συμβεί αυτό τότε κάνε αυτό...)
- Διατίθενται κάρτες επέκτασης (shields), αισθητήρες και ενεργοποιητές για να δώσουν νέες δυνατότητες στο Arduino
- Μπορεί να επικοινωνεί σειριακά (ενσύρματα ή ασύρματα) με άλλες συσκευές
- Σε συνδυασμό με τη γλώσσα Processing δημιουργούν ένα δυνατό περιβάλλον ανάπτυξης εφαρμογών για απεικόνιση και αλληλεπίδραση με τον χρήστη
- Υπάρχει μια μεγάλη κοινότητα υποστήριξης και πληθώρα πληροφοριών & παραδειγμάτων σχετικά με το Arduino

Πηγές Πληροφόρησης

Ιστοσελίδες

- Arduino GR: <http://www.arduino.gr/>
- Arduino Homepage: <http://www.arduino.cc>
- Arduino Playground: <http://playground.arduino.cc> and <http://playground.arduino.cc/Interfacing/Processing>
- MAKE magazine: <http://makezine.com>
- Instructables: <http://www.instructables.com>
- Sparkfun: <http://www.sparkfun.com/>

Παραδείγματα

- Adafruit: <http://ladyada.net/learn/arduino/>
- Tronixstuff: <http://tronixstuff.wordpress.com/>

Βιβλία

- ***“Getting Started With Arduino”***, by Massimo Banzi. O’ Reilly Press, 2nd ed., 2011
- ***“Arduino Cookbook”***, by Michael Margolis. O’ Reilly Press, 2nd ed., 2011.
- ***“30 Arduino Projects for the evil genius”***, by Simon Monk. McGraw Hill, 2010.
- ***“Getting Started With Processing”***, by Casey Reas & Ben Fry. O’Reilly, 2010.