

Χρήσεις της πλατφόρμας ανοιχτού υλισμικού Arduino στις σύγχρονες διαδραστικές τέχνες και τη μουσική δημιουργία

Φίλιππος Θεοχαρίδης

Τι είναι το Arduino

- Μια πλατφόρμα 'φυσικής πληροφορικής' ανοιχτού κώδικα
 - Ανοιχτού κώδικα ως προς το λογισμικό και το υλικό του
 - 'φυσική πληροφορική' (physical computing)... χμμ
πανταχού παρούσα πληροφορική, αόρατη πληροφορική, νοήμοντα περιβάλλοντα, έξυπνα αντικείμενα, διαδραστικά συστήματα...
- Μια ηλεκτρονική πλακέτα, ένα προγραμματιστικό περιβάλλον, μια φιλοσοφία ανάπτυξης υλισμικού...
- Ένας μικρός υπολογιστής που μπορείτε να προγραμματίσετε
 - Δουλεύει αυτόνομα ή επικοινωνεί με άλλες συσκευές (υπολογιστές, κινητά τηλέφωνα...)

Τι είναι το Arduino

- 'Αισθάνεται' το περιβάλλον (φυσικός κόσμος) λαμβάνοντας σήματα εισόδου από διάφορους αισθητήρες
- και το 'επηρεάζει' ελέγχοντας πχ. ήχους, φώτα, μοτέρ και άλλες συσκευές

Τι είναι το Arduino

Τεχνικά χαρακτηριστικά (Arduino Uno)

- | | |
|---------------------------------|--|
| • Ψηφιακές είσοδοι/έξοδοι (I/O) | 14 (οι 6 μπορούν να χρησιμοποιηθούν σαν PWM* έξοδοι) |
| • Αναλογικές είσοδοι | 6 |
| • Μνήμη Flash | 32 KB |
| • Ταχύτητα επεξεργαστή | 16 MHz (... ≈ έναν Η/Υ του '90) |

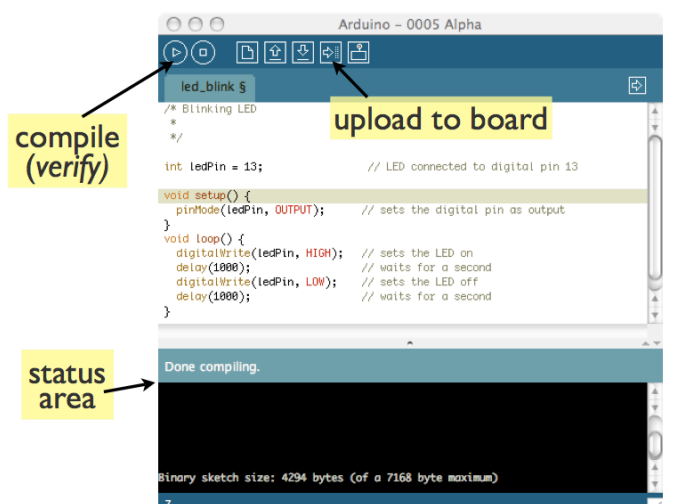
* = πρακτικά αναλογικές έξοδοι

Τι είναι το Arduino

Πως το προγραμματίζουμε;

- Γράφουμε το πρόγραμμα στον υπολογιστή μας
- Το φορτώνουμε στο Arduino
- Το Arduino μπορεί να χρησιμοποιηθεί αυτόνομα
ή σαν διεπαφή (interface) μεταξύ του υπολογιστή και του φυσικού κόσμου

Λογισμικό Arduino

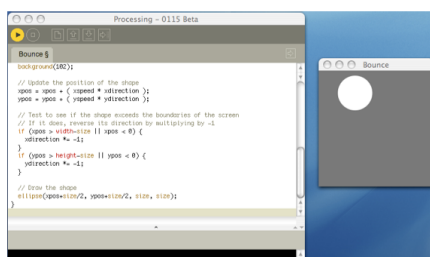


Arduino & Processing

<http://processing.org>

Ένα εύχρηστο προγραμματιστικό περιβάλλον για εικόνα και βίντεο

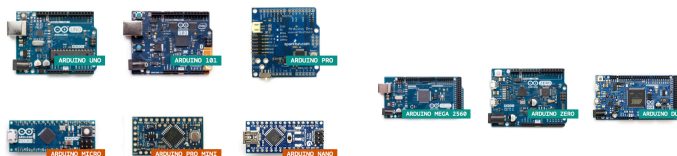
- Ίδιο γραφικό περιβάλλον και παρόμοια φιλοσοφία αλλά λίγο διαφορετική γλώσσα



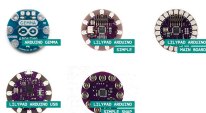
Η πλακέτα Arduino

Υπάρχουν πολλές παραλλαγές της βασικής πλακέτας Arduino με διαφορές στο μέγεθος, τον αριθμό εισόδων/εξόδων, με ασύρματη συνδεσιμότητα (wifi, bluetooth), μερικές με ενσωματωμένους αισθητήρες...

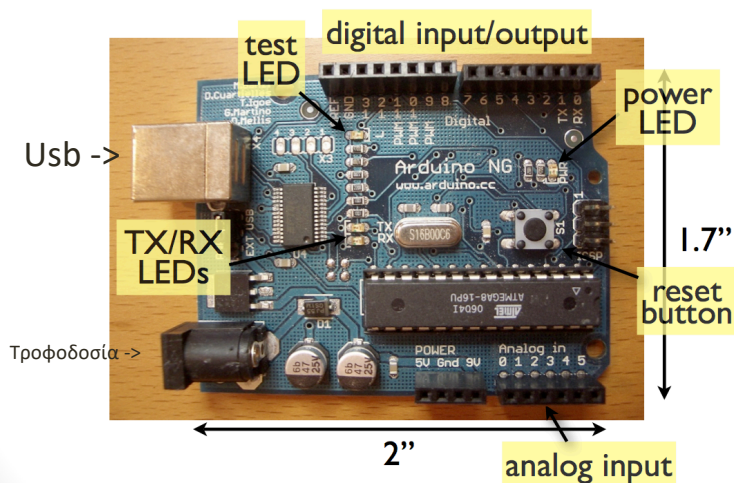
- ... Πολλά 'μοντέλα' (και πολλές μάρκες)



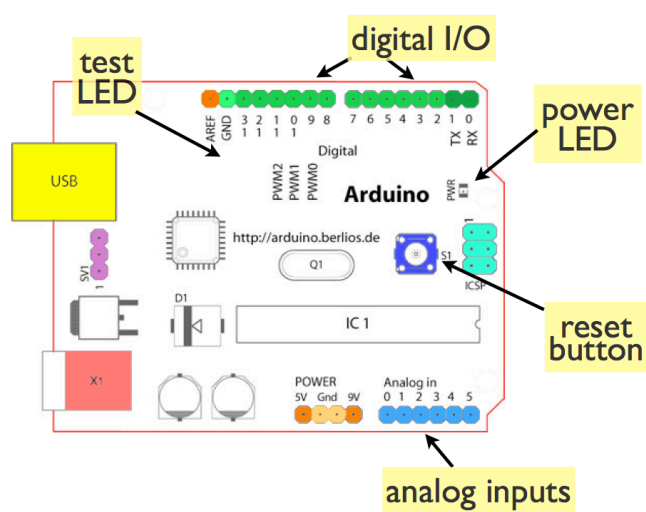
- ... Κάποια μπορείτε να τα ράψετε και στα ρούχα σας (η σειρά Lilypad)



Η πλακέτα Arduino

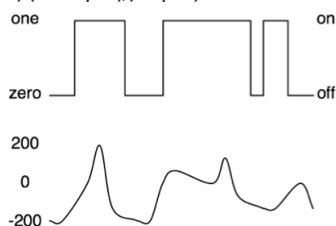


Η πλακέτα Arduino



Ψηφιακό και Αναλογικό σήμα

- Ψηφιακό – μόνο δύο τιμές: On/Off, 0/1
- Αναλογικό – Πολλές (άπειρες;) τιμές



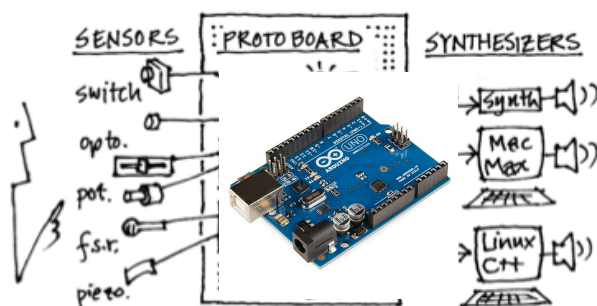
- Οι υπολογιστές δεν το “έχουν” πραγματικά με τον αναλογικό κόσμο
- ... Τα (ψευτο*)καταφέρνουν όμως με ψηφιοποίηση

* κβαντισμός = δείγματα του αναλογικού σήματος σε τακτά χρονικά διαστήματα

Το Arduino στις διαδραστικές τέχνες, τη μουσική και τα μουσικά όργανα

Σαν διεπαφή για αισθητήρες (sensor interface)

- Λαμβάνει σήματα από τους αισθητήρες και τα στέλνει στον υπολογιστή ώστε να τα χρησιμοποιήσουμε σαν σήματα ελέγχου για παραμέτρους του ήχου πχ.



Αισθητήρες

Μετατρέπουν κάποιο φυσικό μέγεθος σε ηλεκτρισμό

- Πχ. ένα μικρόφωνο είναι ένας 'αισθητήρας' ηχητικών κυμάτων
- Μερικοί συνηθισμένοι αισθητήρες
 - Πίεση (Force Sensitive Resistor - FSR)
 - κρούση (piezo)
 - Κάμψη
 - Απόσταση (υπέρυθρες, ultrasound)
 - Θερμοκρασία
 - φωτεινότητα (φωτοαντίσταση)
 - Επιτάχυνση, προσανατολισμός (accelerometer, gyroscope)
 - Βιομετρικοί, πχ. εγκεφαλικά κύματα (EEG)



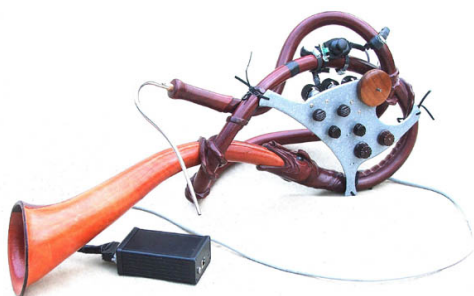
Επαυξημένα μουσικά όργανα (augmented instruments)

- 'Παραδοσιακά' όργανα με αυξημένες δυνατότητες μέσω της χρήσης αισθητήρων



Matthew Burtner: "Metasax", Neal Farwell: "Adapted Trombone"

Υβριδικά, Ηλεκτρο-ακουστικά όργανα



Bent Leather Band: Serpentine Basoon

Ελεγκτές (controllers)

- ... Που μοιάζουν με 'παραδοσικά' μουσικά όργανα



Dan Overholt: "Overtone Violin"

Ελεγκτές (controllers)

- ...Ή εμπνευσμένοι από μουσικά όργανα



Dan Trueman: "BoSSA - bowed-sensored-speaker array"

Ελεγκτές (controllers)

- Έναλλακτικά μουσικά όργανα



Seaboard Alphashpere



Gil Weinberg: "Beatbugs"

Το Arduino στις διαδραστικές τέχνες, τη μουσική και τα μουσικά όργανα

- Σαν διεπαφή για ενεργοποιητές (actuator interface)

Ρομποτικοί 'ερμηνευτές'

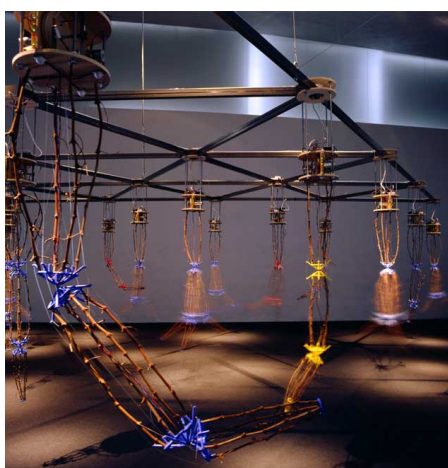


Magnetic Resonator Piano



Το Arduino στις διαδραστικές τέχνες, τη μουσική και τα μουσικά όργανα

- Σε διαδραστικές εγκαταστάσεις



Autopoiesis 2000 by Ken Rinaldo at the Kiasma Museum in Helsinki Finland

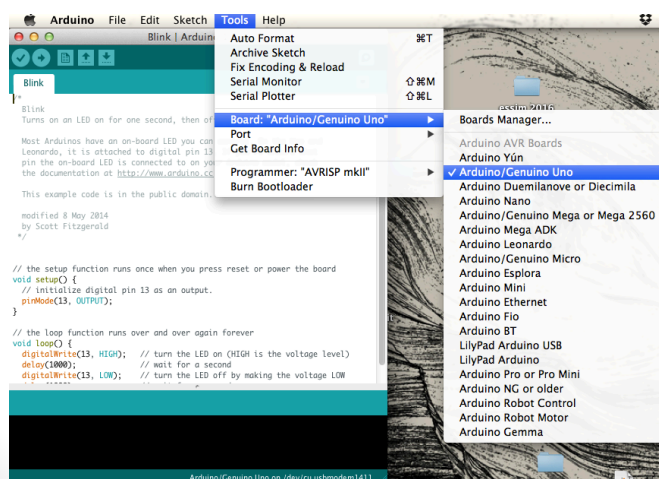
ΜΙΑ ΠΡΑΚΤΙΚΗ ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO

Μεταφόρτωση (upload) λογισμικού στο Arduino

- Κατεβάξετε το **Arduino Desktop IDE** δωρεάν από τη σελίδα [arduino.cc](https://www.arduino.cc) και το εγκαθιστάτε
- <https://www.arduino.cc/en/Guide>
 - Στη δεξιά πλευρά της σελίδας, δείτε τις οδηγίες για το μοντέλο σας

Μεταφόρτωση (upload) λογισμικού στο Arduino

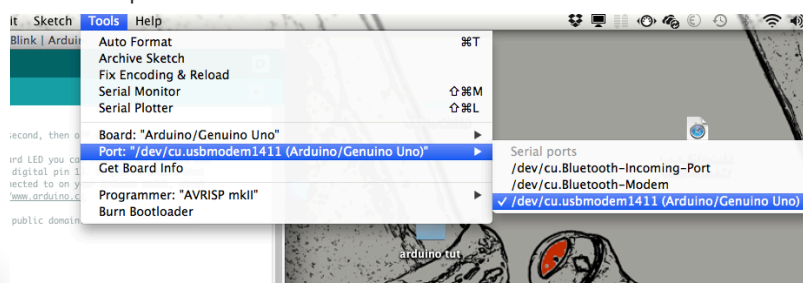
- Επιλέγετε το μοντέλο του Arduino σας



Μεταφόρτωση (upload) λογισμικού στο Arduino

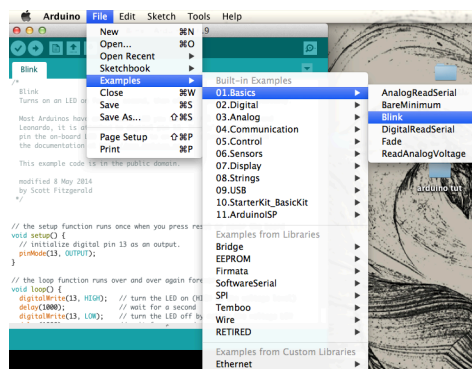
- Συνδέετε το arduino στον υπολογιστή με usb
- Στο λογισμικό του επιλέγετε τη θύρα (port) που είναι συνδεδεμένο το arduino

Η φυσική σύνδεση μπορεί να είναι usb αλλά στην πραγματικότητα το arduino επικοινωνεί με τον υπολογιστή με σειριακό (serial) πρωτόκολλο



Μεταφόρτωση (upload) λογισμικού στο Arduino

- Τα προγράμματα ονομάζονται *sketches*
- Φορτώστε το παράδειγμα 'blink'



Μεταφορτώστε το sketch στο Arduino

Compile (έλεγχος για λάθη στον κώδικα)

Τα λαμπάκια Tx (transmit) και Rx (receive) πάνω στο arduino αναβοσβήνουν κατά τη διάρκεια της μεταφόρτωσης

κώδικας

Πιθανά Errors εμφανίζονται με πορτοκαλί χαρακτήρες εδώ

```

// 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
  // ...
}
  
```

Copy error messages

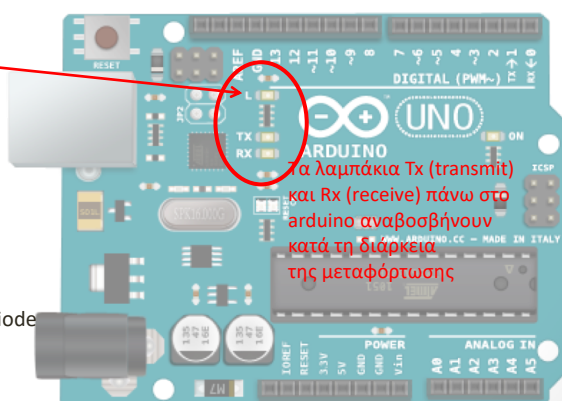
error: 'pinMode' was not declared in this scope
 pinMode(13, OUTPUT);
 ^
 exit status 1

Blinky - blinky

Το sketch αυτό (blink) αναβοσβήνει ένα λαμπάκι LED* στο pin 13 κάθε ένα δευτερόλεπτο

- Ελέγξτε στην πλακέτα σας ότι αυτό πραγματικά συμβαίνει

Pin 13 led



Τα λαμπάκια Tx (transmit) και Rx (receive) πάνω στο arduino αναβοσβήνουν κατά τη διάρκεια της μεταφόρτωσης

* Light Emitting Diode

- Οκ...
 - Το δοκιμάσαμε...
 - Δουλεύει....
-
- Μπορούμε να χρησιμοποιήσουμε Arduino χωρίς να ασχοληθούμε στα σοβαρά με τη γλώσσα του...

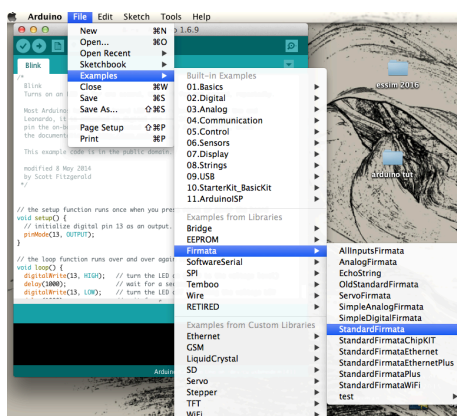
1. Maxuino (σχετικά παλιό πλέον, αλλά δουλεύει ... ειδικά με παλιότερα boards)

- ... **Max** (της cycling74) – (ard)uino
- Ένας τρόπος να χρησιμοποιήσουμε δεδομένα από/προς το arduino (... και τον έξω κόσμο) μέσα στο Max
- ... χωρίς να χρειάζεται να προγραμματίσουμε τίποτα στο arduino
- ... και άρα χωρίς να ξέρουμε τη γλώσσα του
- Διαθέσιμο δωρεάν στο Maxuino.org

... Αν δε μας αρέσει ούτε το Max, υπάρχει έτοιμη συσκευή (device) maxuino για max4live (στο Ableton Live) (ή στο *m4l-connection-kit* και αλλού)

Maxuino (ή για το m4l-connection-kit)

- Φορτώνουμε το sketch 'standard firmata' στο λογισμικό του arduino



- και πατάμε upload για να το METAφορτώσουμε στην πλακέτα arduino

2. Το Arduino σαν διεπαφή MIDI

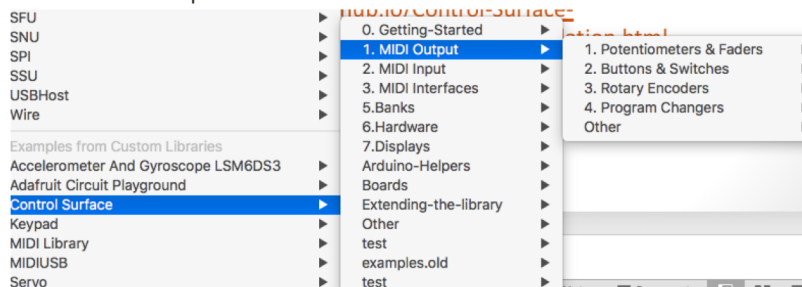
- Τα νεότερα μοντέλα (Uno, Leonardo, due, nano 33 iot... teensy...) μπορούν να εμφανιστούν σε υπολογιστές / φορητές συσκευές σαν *class compliant MIDI devices* (χρειάζονται τη βιβλιοθήκη *MIDIUSB* από το μενού *tools->manage libraries*)
- Δουλεύουν όμως και παλιότερα μοντέλα, μέσω κάποιου λογισμικού *serial>MIDI* που θα πρέπει να τρέχει στον υπολογιστή σας (πχ. <https://github.com/projectgus/hairless-midiserial>)
- Control Surface
 - <https://ttapa.github.io/Control-Surface-doc/Doxygen/index.html>
 - <https://github.com/ttapa/Control-Surface>
 - https://ttapa.github.io/Control-Surface-doc/Doxygen/d8/da8/md_pages_Installation.html

Control Surface

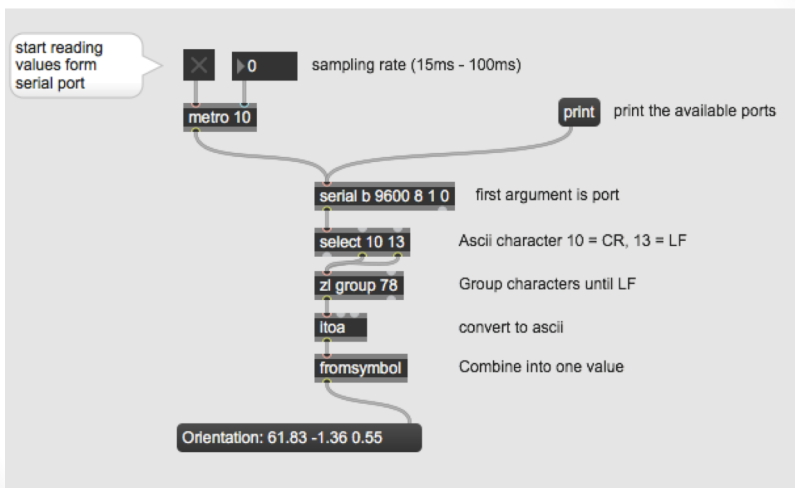
Αφού το εγκαταστήσετε σαν βιβλιοθήκη στο Arduino IDE...

Μπορείτε να συνδυάσετε κώδικα από τα παραδείγματά της, ανάλογα με τις ανάγκες σας

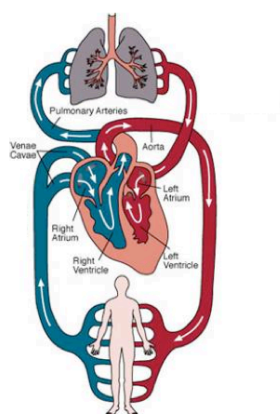
Μενού File->Examples->



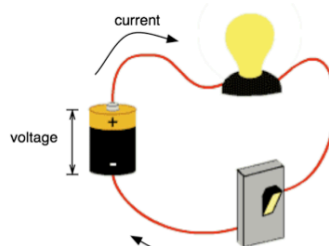
3. Ένα απλό patch στο Max που διαβάζει τις πληροφορίες από το *serial port*



κυκλώματα

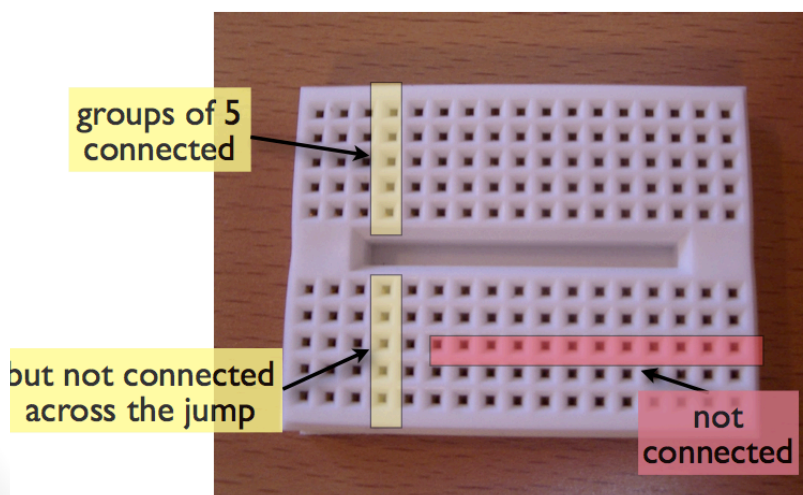


Η καρδιά χτυπάει, το αίμα κυλάει

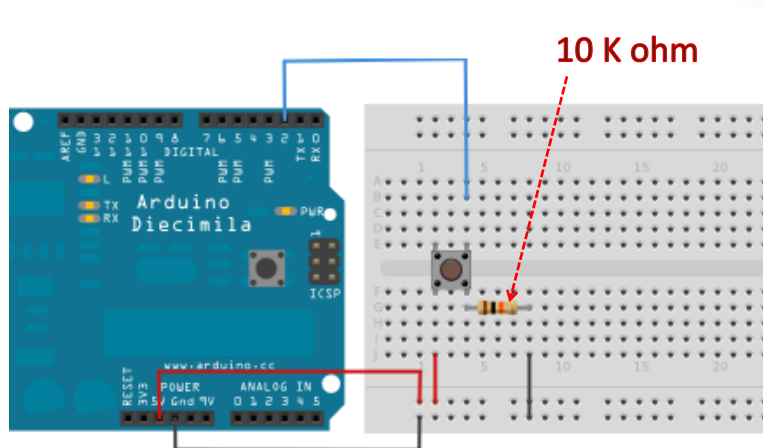


Η τάση 'σπρώχνει', το ρεύμα κυλάει

Solderless breadboard



Διαβάζουμε τις τιμές ενός κουμπιού (button)

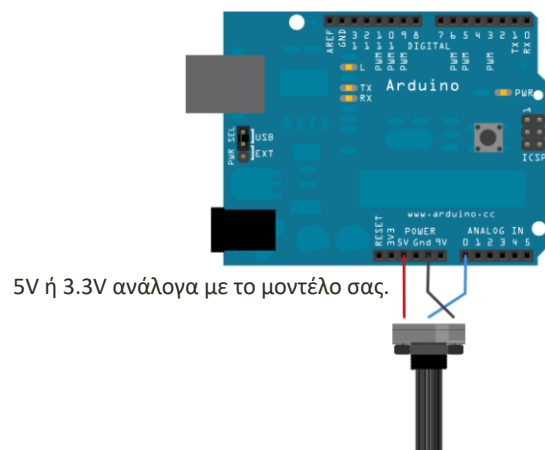


ΠΡΟΣΟΧΗ!!! 5V ή 3.3V ανάλογα με το μοντέλο σας.

Πχ. τα nano 33 χρησιμοποιούν τα 3.3V σαν reference. Δεν πρέπει ποτέ να δίνετε παραπάνω από τη reference τάση στις αναλογικές εισόδους ενός arduino

Color	1 st band	2 nd band	3 rd band (multiplier)	4 th band (tolerance)	Temp. Coefficient
Black	0	0	$\times 10^0$		
Brown	1	1	$\times 10^1$	$\pm 1\%$ (F)	100 ppm
Red	2	2	$\times 10^2$	$\pm 2\%$ (G)	50 ppm
Orange	3	3	$\times 10^3$		15 ppm
Yellow	4	4	$\times 10^4$		25 ppm
Green	5	5	$\times 10^5$	$\pm 0.5\%$ (D)	
Blue	6	6	$\times 10^6$	$\pm 0.25\%$ (C)	
Violet	7	7	$\times 10^7$	$\pm 0.1\%$ (B)	
Gray	8	8	$\times 10^8$	$\pm 0.05\%$ (A)	
White	9	9	$\times 10^9$		
Gold			$\times 10^{-1}$	$\pm 5\%$ (J)	
Silver			$\times 10^{-2}$	$\pm 10\%$ (K)	
None				$\pm 20\%$ (M)	

Ποτενσιόμετρο (μεταβλητή αντίσταση)



Control Surface

```
Control-Change-Potentiometer | Arduino 1.8.13

Control-Change-Potentiometer
* Mapping
* -----
* Select the Arduino as a custom MIDI controller in your DAW, and use the
* MIDI learn option to assign the potentiometer to a function.
* It will send the MIDI Control Change Channel Volume parameter for channel 1.
*
* Written by PieterP, 2019-08-13
* https://github.com/ttappa/Control-Surface
*/

#include <Control_Surface.h> // Include the Control Surface library

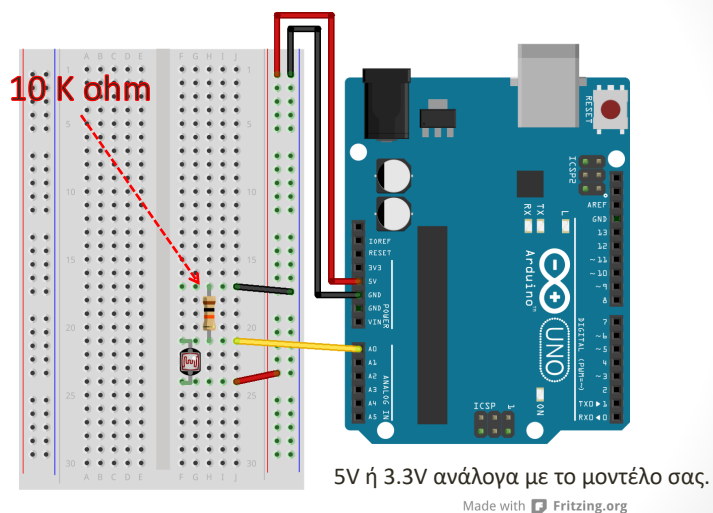
// Instantiate a MIDI over USB interface.
USBMIDI_Interface midi;

// Instantiate a CPotentiometer object
CPotentiometer potentiometer = {
  A0, // Analog pin connected to potentiometer
  {MIDI_CC::Channel_Volume, CHANNEL_1}, // Channel volume of channel 1
};

void setup() {
  Control_Surface.begin(); // Initialize Control Surface
}

void loop() {
  Control_Surface.loop(); // Update the Control Surface
}
```

Φωτοαντίσταση



Control Surface – 2cc pots, 1 note button



```

example-2CC-Potentiometers-1note-Button | Arduino 1.8.13
example-2CC-Potentiometers-1note-Button
#include <Control_Surface.h> // Include the Control Surface library

// Instantiate a MIDI over USB interface.
USBMIDI_Interface midi;

using namespace MIDI_Notes;

// Instantiate a NoteButton object
NoteButton button = {
  5, // Push button on pin 5
  {note(C, 4), CHANNEL_1}, // Note C4 on MIDI channel 1
};

// Instantiate an array of CCPotentiometer objects
CCPotentiometer potentiometers[] = {
  {A0, // Analog pin connected to potentiometer 1
   0x10}, // Controller number of the first potentiometer
  {A1, // Analog pin connected to potentiometer 2
   0x11}, // Controller number of the second potentiometer
};

void setup() {
  Control_Surface.begin(); // Initialize Control Surface
}

void loop() {
  Control_Surface.loop(); // Update the Control Surface
}

Done compiling.
Sketch uses 20544 bytes (78%) of program storage space. Maximum is 262144 bytes.
Global variables use 3384 bytes (10%) of dynamic memory, leaving 29384 bytes for local variables.
Arduino NANO 33 IoT on /dev/cu.usbmodem1423
  
```