

Introduction to Arduino



IEEE Student Branch
University of Thessaly

Γιώργος Γκουγκούδης
Βασίλης Σαμαράς

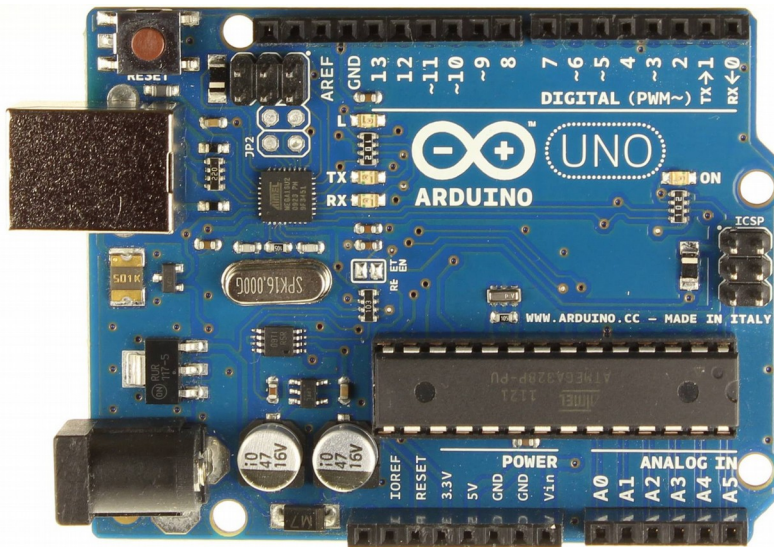
Τι είναι το Arduino;

Το Arduino είναι ένα εργαλείο για κατασκευή κυκλωμάτων που μπορούν να αισθανθούν και να ελέγχουν το φυσικό κόσμο έξω από την επιφάνεια εργασίας του υπολογιστή σας (physical computing).

Είναι μία πλατφόρμα ανοιχτού κώδικα(open-source) που βασίζεται σε μια απλή πλακέτα με μικροελεγκτή(AVR), και ένα περιβάλλον ανάπτυξης μέσω του οποίου αυτός προγραμματίζεται.

Μια πρώτη ματιά

- Arduino UNO Rev3 η προτεινόμενη έκδοση για αρχάριους.



Γιατί Arduino;

- Υπάρχουν πολλές άλλες πλατφόρμες με μικρο-ελεγκτή που διατίθενται για physical computing. Το Arduino απλοποιεί τη διαδικασία της υλοποίησης ενός Project με μικρο-ελεγκτές, ενώ παράλληλα προσφέρει αρκετά πλεονεκτήματα για τους εκπαιδευτικούς, τους μαθητές, και τους ενδιαφερόμενους ερασιτέχνες, σε σχέση με άλλα συστήματα.
- Σχετικά φθηνό σε σύγκριση με άλλες πλατφόρμες μικρο-ελεγκτή. Οι φθηνότερες εκδόσεις του Arduino μπορούν να συναρμολογηθούν με το χέρι, και ακόμη και οι προ-συναρμολογημένες πλακέτες Arduino κοστίζουν περίπου 30€
- Cross-platform - Το Arduino λογισμικό τρέχει σε Windows, Macintosh OSX και Linux λειτουργικά συστήματα. Τα περισσότερα συστήματα μικρο-ελεγκτή περιορίζονται στα Windows.
- Απλό, καθαρό περιβάλλον προγραμματισμού - Το περιβάλλον προγραμματισμού Arduino είναι εύκολο στη χρήση για αρχάριους, αλλά είναι και αρκετά ευέλικτο για τους προχωρημένους χρήστες

Που μπορεί να χρησιμοποιηθεί;

- Οι πλακέτες Arduino μπορούν να χρησιμοποιηθούν για την ανάπτυξη διαδραστικών συστημάτων, τα οποία λαμβάνουν εισόδους από μια ποικιλία από διακόπτες ή αισθητήρες, και ελέγχουν φώτα, κινητήρες, και άλλες φυσικές εξόδους.
- Τα σχέδια(sketches) του Arduino μπορούν να είναι αυτόνομα, ή μπορούν να επικοινωνούν με λογισμικό που τρέχει στον υπολογιστή σας (π.χ. Python, C, Processing).
- Οι πλακέτες μπορούν να συναρμολογηθούν από εσάς τους ίδιους ή να αγοραστούν προ-συναρμολογημένες. Το περιβάλλον προγραμματισμού μπορείτε να το κατεβάσετε δωρεάν και είναι ανοιχτού κώδικα.
<https://www.arduino.cc/en/Main/Software>

Πώς μπορώ να το προγραμματίσω;

Η γλώσσα προγραμματισμού του Arduino είναι μια υλοποίηση της Wiring, η οποία αποτελεί μια παραλλαγή-μίξη C και C++ που απευθύνεται σε μικροελεγκτές αρχιτεκτονικής AVR όπως ο ATmega η οποία βασίζεται στο περιβάλλον προγραμματισμού πολυμέσων Processing.

<https://www.arduino.cc/en/Reference/HomePage>

Συνήθως χρησιμοποιούμε C++ όταν θέλουμε να φτιάξουμε βιβλιοθήκες επέκτασης κάποιας λειτουργικότητας.

Αυτές οι βιβλιοθήκες μας βοηθούν να υλοποιήσουμε μια λειτουργία με μόνο λίγες γραμμές κώδικα χωρίς να χρειάζεται να γνωρίζουμε λεπτομέρειες χαμηλού επιπέδου.

Μπορούμε να βρούμε τέτοιες βιβλιοθήκες στο ArduinoPlayground, ή στο Github.

<https://github.com/search?utf8=%E2%9C%93&q=arduino>

Βασικά μέρη

Παραθέτουμε ορισμένα από τα βασικά στοιχεία του υλικού και του λογισμικού Arduino και σύντομες περιγραφές τους:

- Sketch λέγεται το αρχείο που αποθηκεύεται ο κώδικας που γράφουμε για κάθε project.
- Digital Pins λέγονται οι ακίδες που έχει κάθε πλακέτα και χρησιμοποιούνται για ψηφιακή λογική (έξοδο/είσοδο).
- Analog Input Pins είναι οι ακίδες που χρησιμοποιούνται για την ανάγνωση αναλογικών σημάτων, τα οποία μετριοούνται ως ψηφιακές ποσότητες(Analog to Digital Converter).
- Memory: Προσωρινή και διαρκής μνήμη(SRAM, EEPROM, SD)
- Variables-Μεταβλητές
- Functions-Συναρτήσεις
- Libraries-Βιβλιοθήκες
- Shields λέγονται οι πλακέτες που <<κουμπώνουν>> πάνω στο Arduino και επεκτείνουν τις διαθέσιμες λειτουργίες του, ή προσθέτουν καινούριες.

Shields

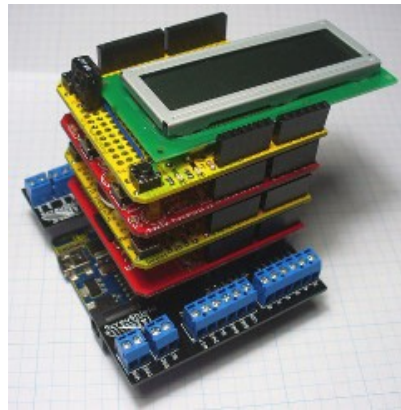
WiFi



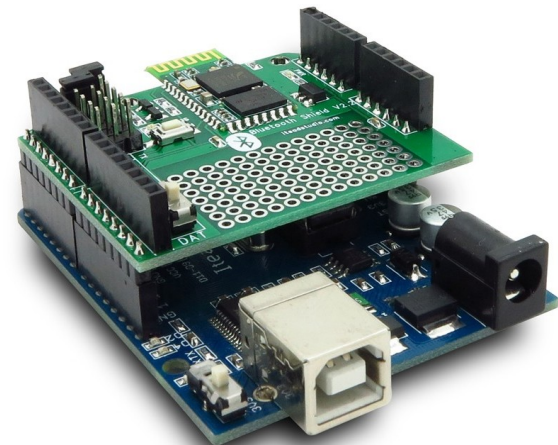
Servo



Stack

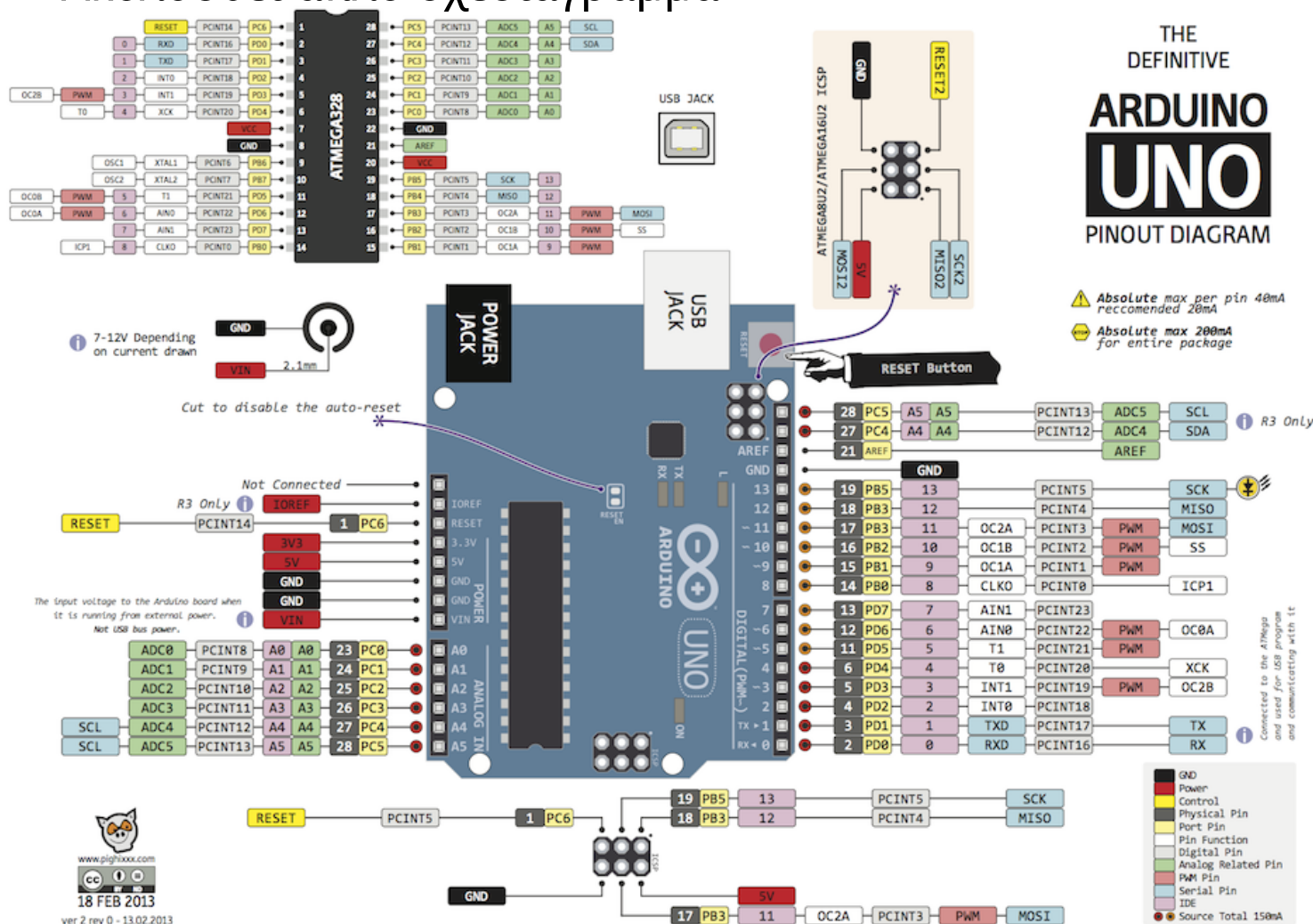


Bluetooth + Proto



Ανατομία

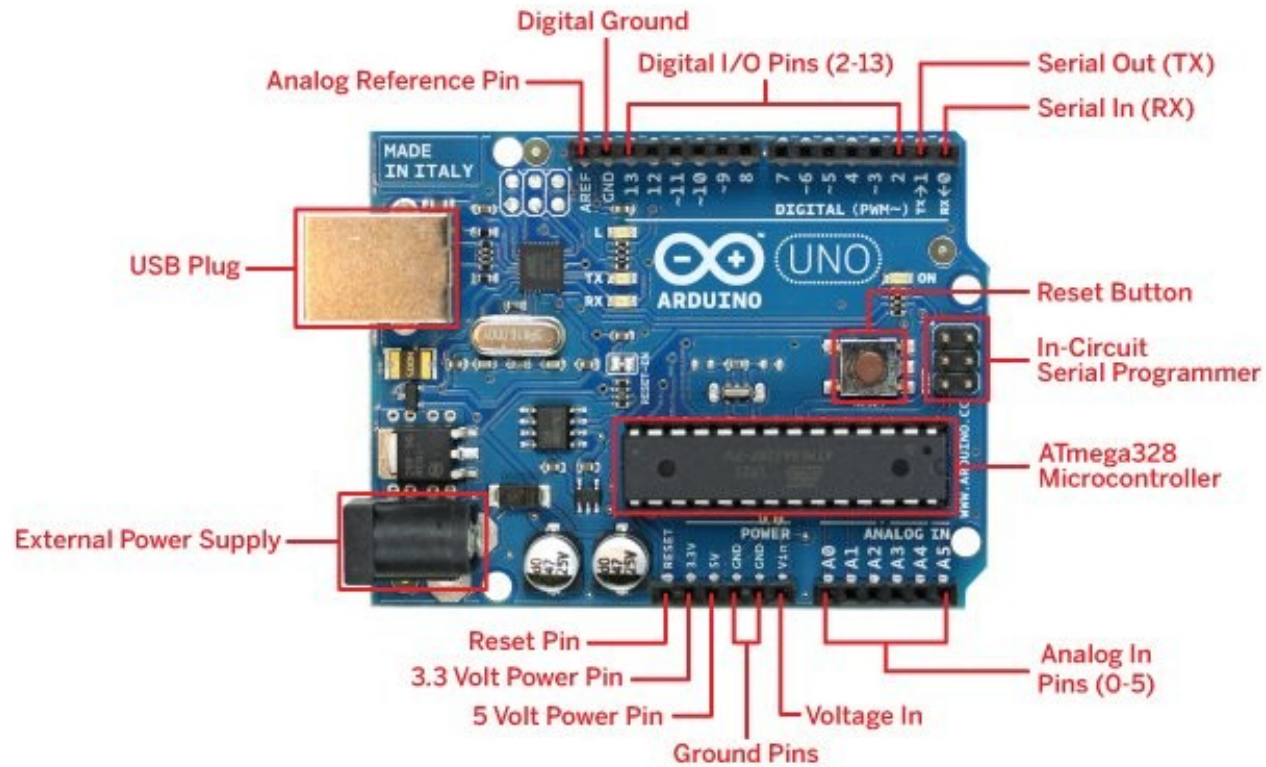
- Ακολουθεί απλό σχεδιάγραμμα



- ή και όχι...

Ανατομία

- Όντως απλό σχεδιάγραμμα:



Περιβάλλον Ανάπτυξης Λογισμικού Arduino

The image shows the Arduino IDE interface with several components labeled:

- verify sketch**: Points to the checkmark icon in the toolbar.
- compile and upload sketch to Arduino**: Points to the upload icon in the toolbar.
- new sketch**: Points to the document icon in the toolbar.
- save sketch**: Points to the floppy disk icon in the toolbar.
- open Serial Monitor**: Points to the magnifying glass icon in the toolbar.
- Sketch name**: Points to the text "Fading" in the title bar.
- Arduino software version**: Points to the text "Arduino 1.0" in the title bar.
- current tab**: Points to the "Fading" tab.
- open sketch**: Points to the document icon in the toolbar.
- Tab menu**: Points to the dropdown arrow in the tab bar.
- the sketch named Fading's source code**: Points to the code editor area.
- The Editor**: Points to the code editor area.
- current line number**: Points to the number "1" in the left margin.
- Error console**: Points to the bottom panel.
- current Arduino model**: Points to the text "Arduino Duemilanove w/ ATmega328" in the bottom panel.
- your system's name for current USB port**: Points to the text "/dev/tty.usbserial-A800f8gT" in the bottom panel.

```
Fading | Arduino 1.0
Fading LED
// by BARRAGAN <http://p...rea.it/h.barragan>

int value = 0; // variable to keep the
int ledpin = 9; // light connected to digital pin 9

void setup()
{
  // nothing for setup
}

void loop()
{
  for(value = 0 ; value <= 255; value+=5) // fade in (from min to max)
  {
    analogWrite(ledpin, value); // sets the value (range from 0 to 255)
    delay(70); // waits for 30 milli seconds to see the dim
  }
  for(value = 255; value >=0; value-=5) // fade out (from max to min)
  {
    analogWrite(ledpin, value);
    delay(70);
  }
}
```

The Arduino IDE

Άλλα περιβάλλοντα

Atom + PlatformIO

The screenshot displays the Atom IDE interface with the PlatformIO menu open. The menu items are as follows:

- Initialize new PlatformIO Project or update existing...
- Import Arduino IDE Project...
- Open Project folder... (Alt+Ctrl+O)
- Build (Alt+Ctrl+B)
- Upload (Alt+Ctrl+U)
- Clean (Alt+Ctrl+C)
- Run other target... (F7)
- Update targets
- Toggle Build Panel (F8)
- Terminal
- Library Manager CLI (GUI soon)
- Serial Monitor (Alt+Shift+M)
- List Serial Ports
- Rebuild C/C++ Project Index (Autocomplete, Linter)
- Install Shell Commands
- Update Platforms and Libraries
- Upgrade PlatformIO CLI
- Update Atom packages
- Settings
- Documentation
- Help

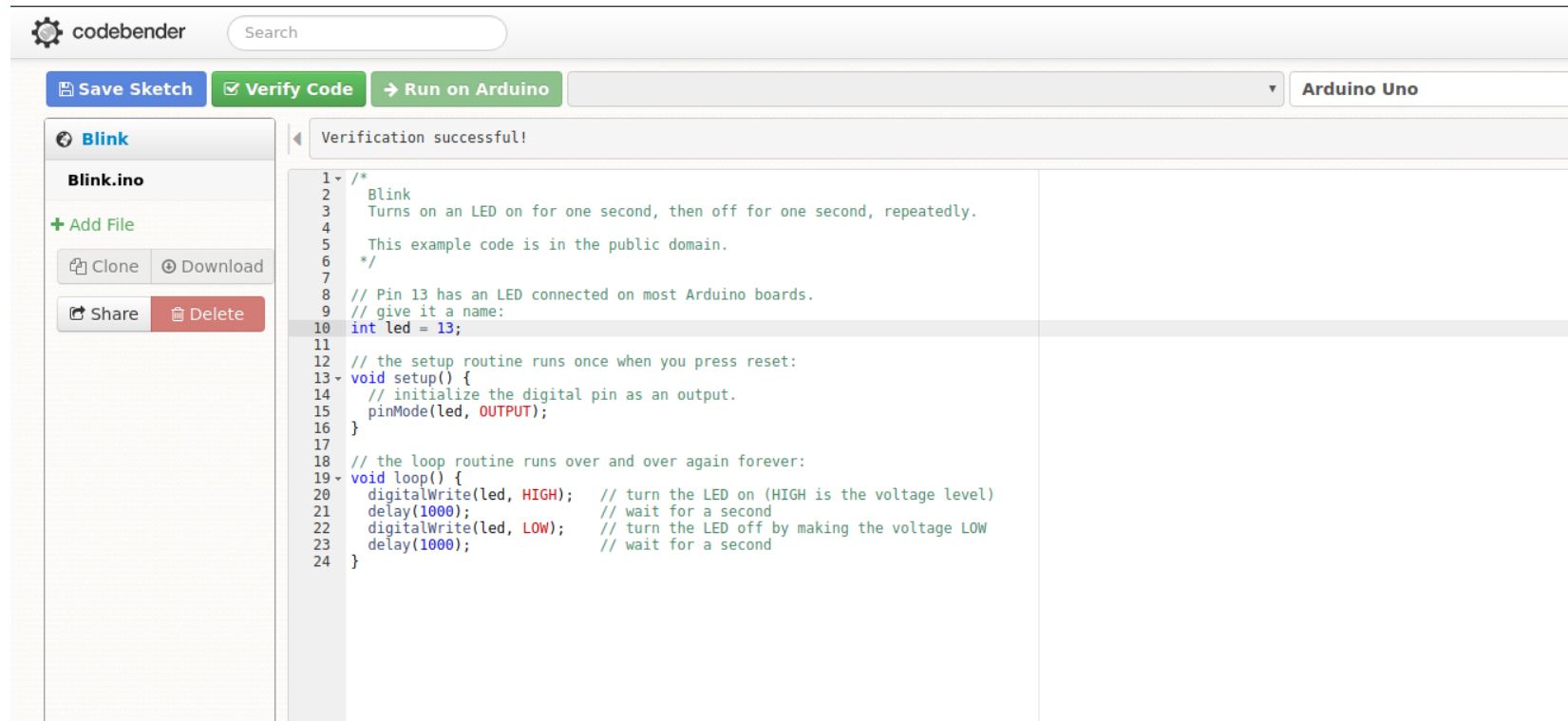
The background code in the editor is as follows:

```
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55 // The uart is the standard output device STDOUT.
56 stdout = &uartout ;
57 // t.every(200, debug);
58 #endif
59 // feed back led
60 pinMode(led_pin, OUTPUT);
61 blink(2);
62
63 // init our adaptive min/max
64 sensor_reading = analogRead(sensor_pin);
65 min_light = sensor_reading - 20;
66 max_light = sensor_reading;
```

The status bar at the bottom shows: PlatformIO: Build | File 0 | Project 0 | No Issues | src/main.cpp | 123:15 | LF | Normal | UTF-8 | C++ | master | +27, -35 | 4 updates

Άλλα περιβάλλοντα

Codebender

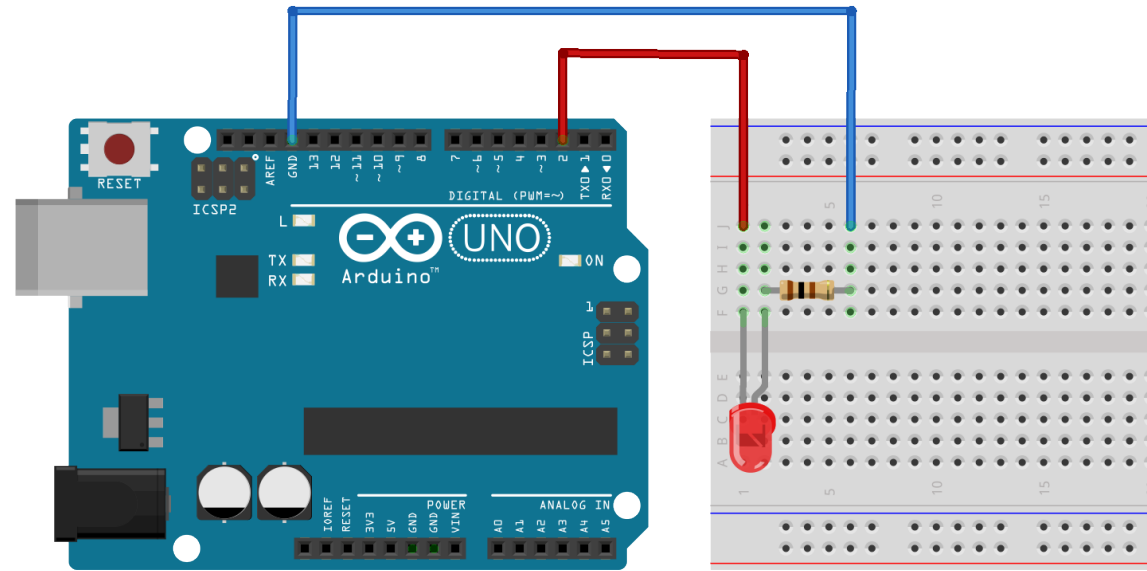


The screenshot displays the Codebender web interface. At the top, there is a search bar and the Codebender logo. Below this, a navigation bar contains buttons for 'Save Sketch', 'Verify Code', and 'Run on Arduino', along with a dropdown menu set to 'Arduino Uno'. On the left side, a sidebar shows the file 'Blink.ino' with options to 'Add File', 'Clone', 'Download', 'Share', and 'Delete'. The main area shows a 'Verification successful!' message and the source code for the Blink sketch. The code is as follows:

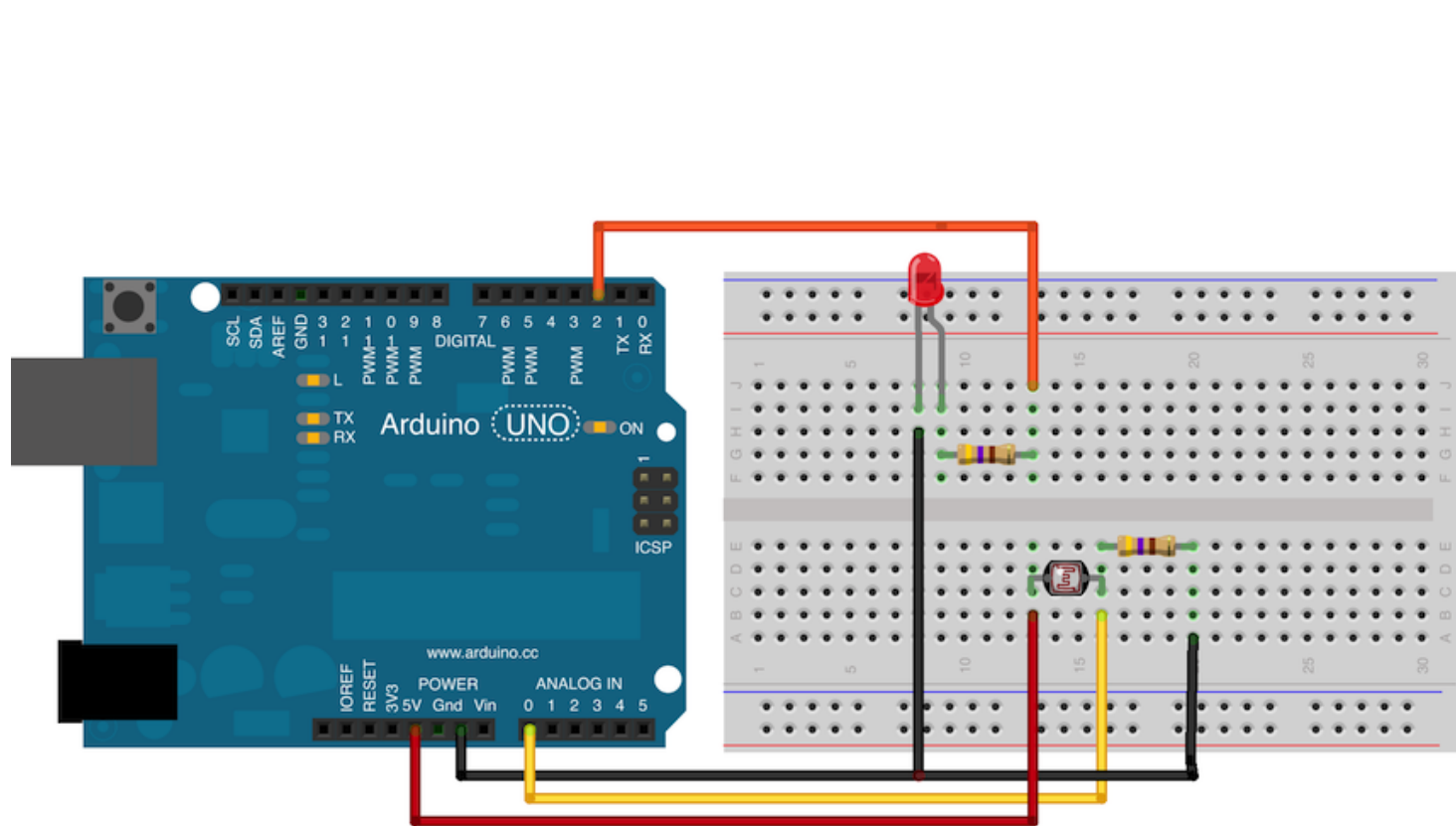
```
1- /*
2-  * Blink
3-  * Turns on an LED on for one second, then off for one second, repeatedly.
4-  *
5-  * This example code is in the public domain.
6-  */
7-
8- // Pin 13 has an LED connected on most Arduino boards.
9- // give it a name:
10 int led = 13;
11
12 // the setup routine runs once when you press reset:
13 void setup() {
14   // initialize the digital pin as an output.
15   pinMode(led, OUTPUT);
16 }
17
18 // the loop routine runs over and over again forever:
19 void loop() {
20   digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level)
21   delay(1000); // wait for a second
22   digitalWrite(led, LOW); // turn the LED off by making the voltage LOW
23   delay(1000); // wait for a second
24 }
```

Online πλατφόρμα ανάπτυξης λογισμικού για Arduino

Παράδειγμα Blink



Παράδειγμα Smart LED



Made with  Fritzing.org

Παράδειγμα IR

<https://github.com/z3t0/Arduino-IRremote>

