



ΕΙΣΑΓΩΓΗ ΣΤΟ ARDUINO



Δημιουργός:
Επιβλέπων:

Μπακάλη Ιωάννα
Πετεινάτος Ηλίας
Υποψήφιος Διδάκτωρ

ΤΙ ΕΙΝΑΙ ΤΟ ARDUINO;

- Είναι ένας συνδυασμός
 - **Υλικού** (πλακέτας, μικροελεγκτή, αντιστάσεων κτλ) και
 - **Λογισμικού** (με την βοήθεια του οποίου προγραμματίζουμε τον μικροελεγκτή)όπου με την βοήθεια του μπορούμε να δημιουργήσουμε διάφορες κατασκευές του πραγματικού κόσμου.
- Χαρακτηριστικά παραδείγματα:
 - Τα φανάρια στους δρόμους
 - Ένα τηλεκατευθυνόμενο αυτοκινητάκι
 - Συστήματα που ελέγχουν κίνηση, θερμοκρασία, υγρασία κτλ
 - Drones
 - Οθόνες με κυλιόμενα μηνύματα που βλέπετε σε μαγαζιά.

Η ΑΝΑΠΤΥΞΙΑΚΗ ΠΛΑΤΦΟΡΜΑ ARDUINO

Ο προγραμματισμός του μικροελεγκτή στην πλατφόρμα **Arduino** γίνεται με τη γλώσσα **Wiring C** που αποτελεί μια παραλλαγή της γνωστής C++.

Έτσι, ο προγραμματισμός μιας τέτοιας πλατφόρμας είναι φιλικότερος και επιτρέπει τη γρηγορότερη ανάπτυξη εφαρμογών χωρίς να προϋποθέτει τη λεπτομερή γνώση της αρχιτεκτονικής του μικροελεγκτή.

Η ΔΙΑΦΟΡΕΤΙΚΗ ΠΡΟΣΕΓΓΙΣΗ ΜΕ ΤΟ ARDUINO

Επίπεδο 3

Επίπεδο 2

Επίπεδο 1

Επίπεδο 0

Assembly

Γλώσσα
Μηχανής

Μικροελεγκτής
(Υλικό)

Κλασικός
προγραμματισμός
Μικροελεγκτή

Wiring C

Assembly

Γλώσσα
Μηχανής

Μικροελεγκτής
(Υλικό)

Προγραμματισμός
μικροελεγκτή με
το Arduino

Ανεξαρτησία
από το υλικό

Εξάρτηση από
το υλικό

- Ο προγραμματισμός οποιουδήποτε μικροελεγκτή προϋποθέτει την ύπαρξη είτε στοιχειωδών κυκλωμάτων, είτε συγκεκριμένων διασυνδέσεων (μέσω κατάλληλων αντιστάσεων) με ένα υπολογιστικό σύστημα (π.χ. μέσω σειριακής ή παράλληλης θύρας), είτε ειδικού προγραμματιστή.
- Στην πλατφόρμα Arduino η συγκεκριμένη τεχνολογία **απαλλάσσει** το σχεδιαστή ή τον προγραμματιστή από όλες αυτές τις λεπτομέρειες και τα κυκλώματα.
- Οτιδήποτε είναι απαραίτητο για τον προγραμματισμό του μικροελεγκτή, βρίσκεται πάνω στην κάρτα στην οποία είναι τοποθετημένος.

- Το πρόγραμμα αναπτύσσεται στον υπολογιστή, ενώ η μεταφόρτωση του εκτελέσιμου κώδικα γίνεται μέσω της θύρας USB.
- Επιπλέον, οι μικροελεγκτές που τοποθετούνται στα Arduino είναι εφοδιασμένοι με ένα ειδικό λογισμικό που ονομάζεται bootloader και δίνει τη δυνατότητα του επαναπρογραμματισμού.
- Οι πιο προχωρημένοι έχουν πάντα τη δυνατότητα να προγραμματίσουν τον μικροελεγκτή του Arduino, όπως και κάθε άλλο μικροελεγκτή, αφαιρώντας τον bootloader ή και χρησιμοποιώντας άλλες διασυνδέσεις και κυκλώματα.

ΚΥΚΛΩΜΑΤΑ
ΥΠΟΣΤΗΡΙΞΗΣ
, BOARD &
ΥΠΟΛΟΓΙΣΤΕΣ



Μικροελεγκτής

Συνιστώσες
αρχιτεκτονικής

Κλασικός
μικροελεγκτής

Υπολογιστής



Μικροελεγκτής

Συνιστώσες
αρχιτεκτονικής

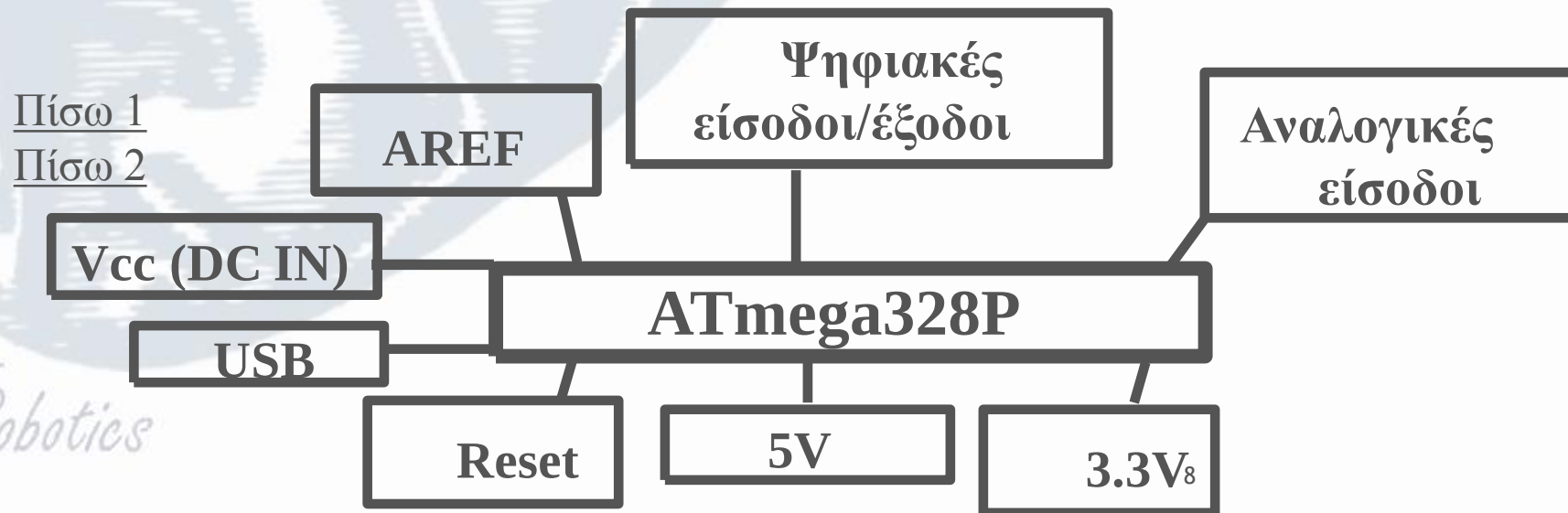
bootloader

Μικροελεγκτής
στην πλατφόρμα
Arduino

Το σχήμα δείχνει τη διαφορά του κλασικού μικροελεγκτή από το «πακέτο» Arduino.

Το **Arduino UNO** που χρησιμοποιεί τον μικροελεγκτή **ATmega328P**, είναι η πιο δημοφιλής έκδοση Arduino. Σχεδόν όλες οι υπόλοιπες εκδόσεις βασίζονται στην ίδια φιλοσοφία.

Το παρακάτω σχήμα παρουσιάζει σε διάγραμμα βαθμίδας τα βασικά χαρακτηριστικά και δυνατότητες που προσφέρονται μέσω της κάρτας.



Vcc (DC IN) Εξωτερική τροφοδοσία για αυτόνομη λειτουργία χωρίς υπολογιστή. Η τροφοδοσία αυτή μπορεί να προέρχεται από τροφοδοτικό, μπαταρία ή άλλο παρόμοιο στοιχείο.

USB Σύνδεση με υπολογιστή μέσω USB για μεταφόρτωση κώδικα και τροφοδοσία. Η επικοινωνία USB υποστηρίζεται από ειδικό ολοκληρωμένο κύκλωμα (IC), το οποίο λειτουργεί συμπληρωματικά με τον μικροελεγκτή.

AREF Τάση αναφοράς, που καθορίζει την ακρίβεια της μέτρησης όταν γίνεται ανάγνωση από τις αναλογικές εισόδους.

Reset Υποστήριξη κουμπιού αλλά και εισόδου reset που ελέγχεται από το πρόγραμμα

ΣΧΗΜΑ

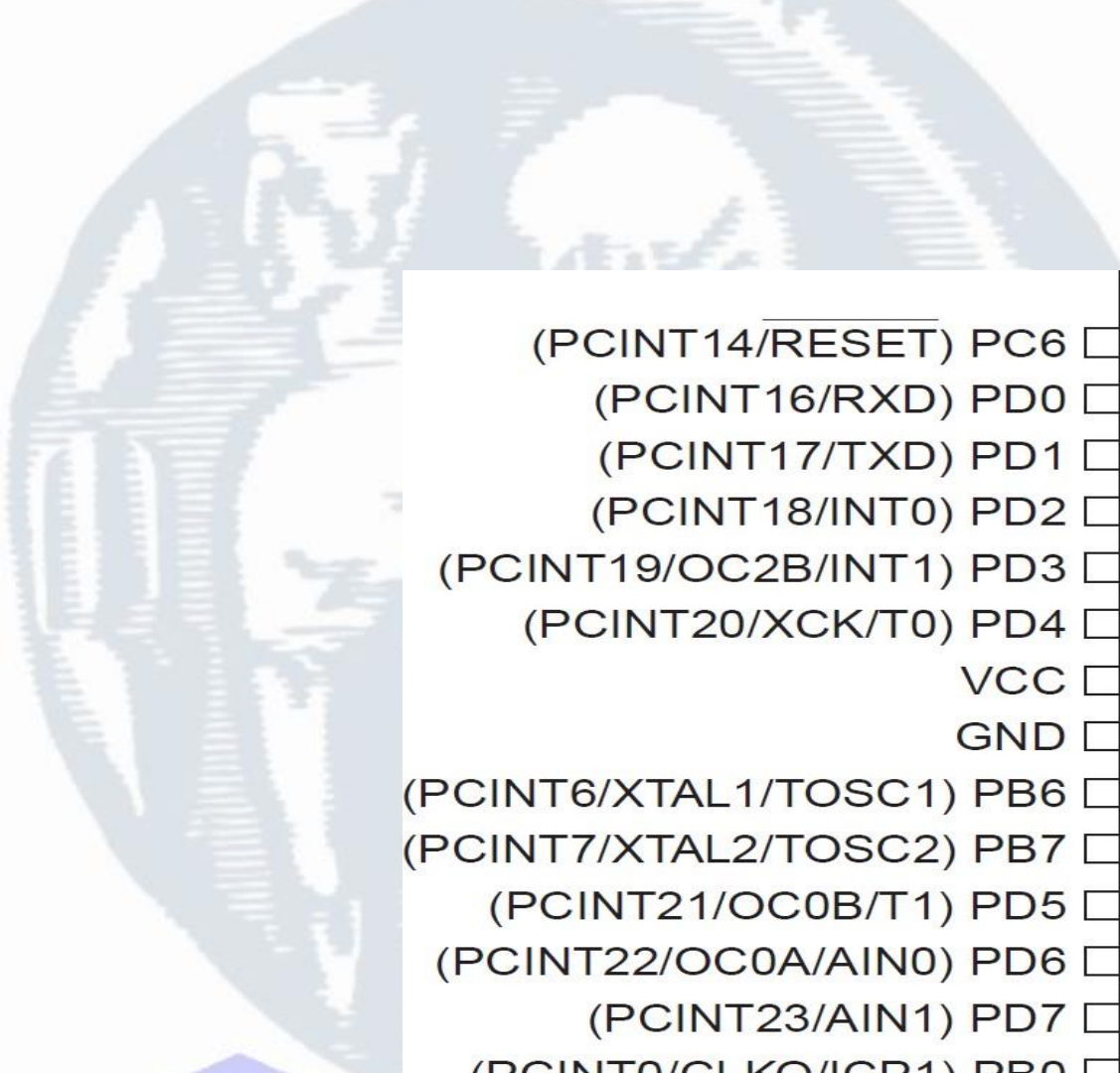
Digital I/O Ψηφιακές είσοδοι και έξοδοι, που αναγνωρίζουν τις στάθμες **0** ή **5V**. Επιπλέον, με την τεχνική διαμόρφωσης εύρους παλμού **PWM (Pulse Width Modulation)** μπορούν να παραχθούν και ψευδοαναλογικά σήματα.

Analog Inputs Ανάγνωση αναλογικών σημάτων 0 έως 5V με δυνατότητα αποτύπωσης σε 1024 στάθμες.

3.3V Τροφοδοσία εξωτερικών κυκλωμάτων στα 3.3V (π.χ. συνδεδεμένων αισθητήρων).

5V Τροφοδοσία εξωτερικών κυκλωμάτων στα 5V (π.χ. συνδεδεμένων αισθητήρων).

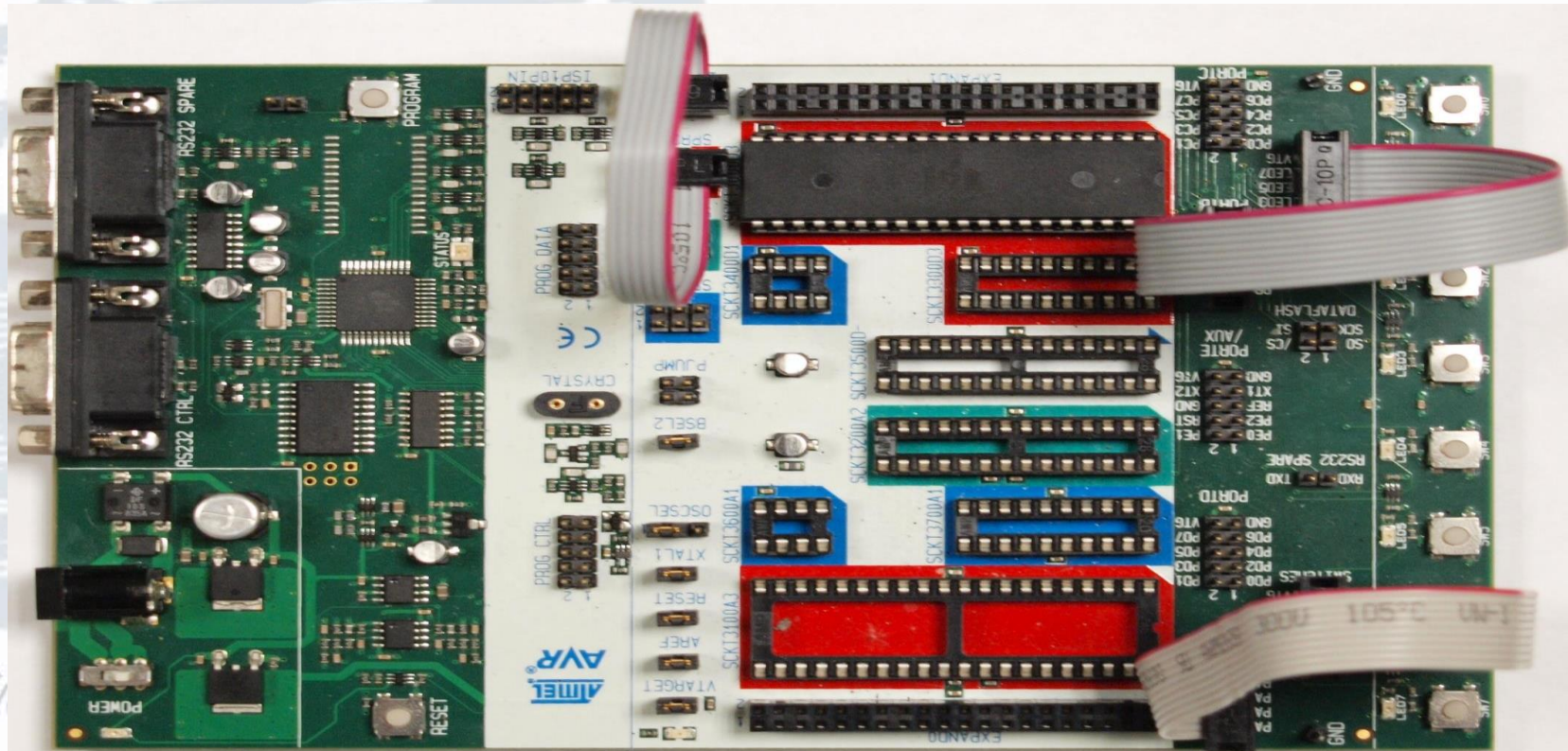
ΣΧΗΜΑ



(PCINT14/RESET) PC6	☐	1	28	☐	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	☐	2	27	☐	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	☐	3	26	☐	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	☐	4	25	☐	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	☐	5	24	☐	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	☐	6	23	☐	PC0 (ADC0/PCINT8)
VCC	☐	7	22	☐	GND
GND	☐	8	21	☐	AREF
(PCINT6/XTAL1/TOSC1) PB6	☐	9	20	☐	AVCC
(PCINT7/XTAL2/TOSC2) PB7	☐	10	19	☐	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	☐	11	18	☐	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	☐	12	17	☐	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	☐	13	16	☐	PB2 ($\overline{\text{SS}}$ /OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	☐	14	15	☐	PB1 (OC1A/PCINT1)

- Όλες οι ακίδες αξιοποιούνται μέσω της κάρτας που φιλοξενεί τον μικροελεγκτή.
- Ο μικροελεγκτής μετά τον προγραμματισμό του μπορεί ακόμα και να αφαιρεθεί από την κάρτα και να τοποθετηθεί για αυτόνομη λειτουργία αφού πλαισιωθεί από τα κατάλληλα κυκλώματα.
- Η κάρτα δηλαδή μπορεί να χρησιμοποιείται μόνο για τον προγραμματισμό.
- Φυσικά υπάρχει και η δυνατότητα τοποθέτησης ολόκληρου του Arduino στο σημείο ενδιαφέροντος για την ολοκλήρωση της εφαρμογής.

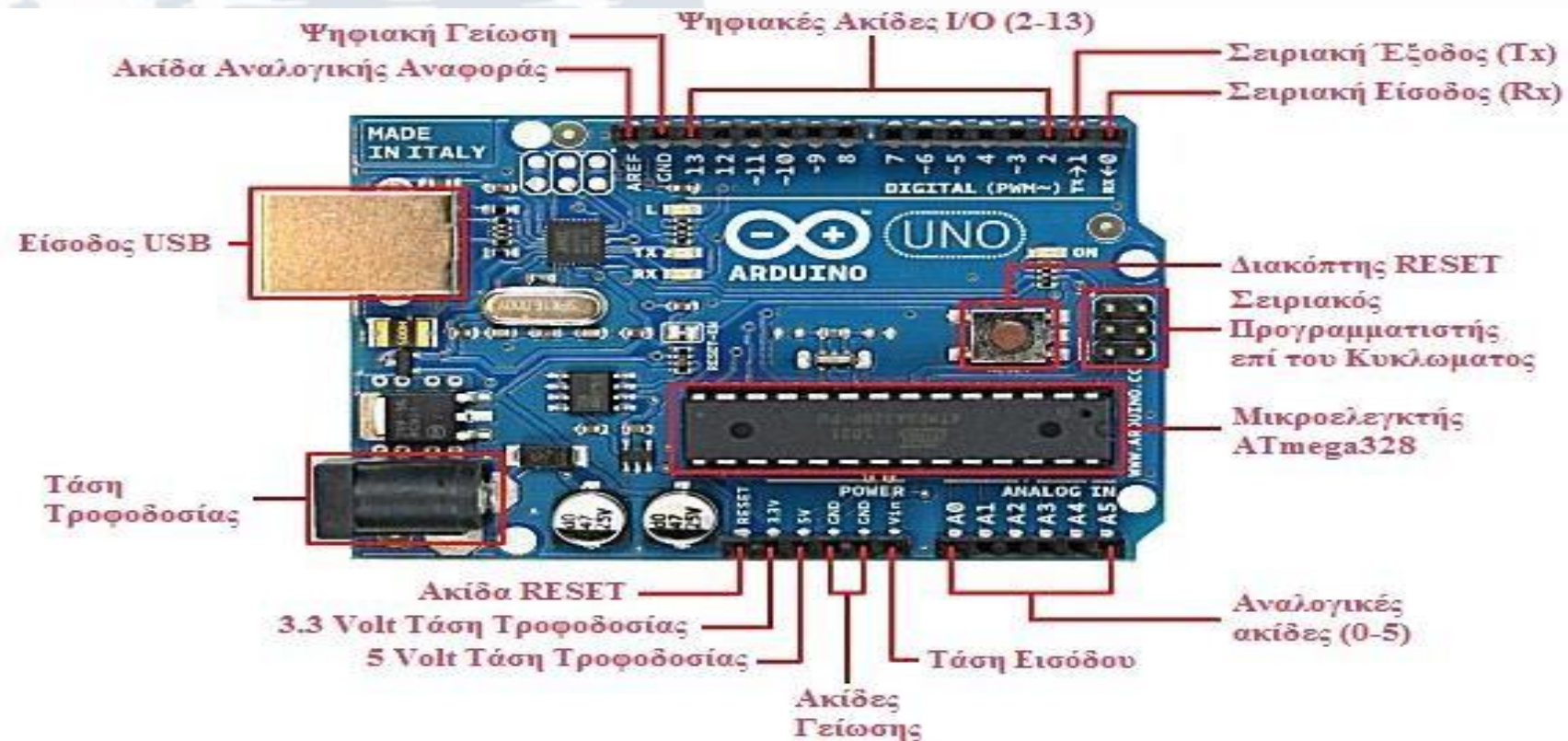
ΣΥΣΤΗΜΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ ΜΕ ΜΙΚΡΟΕΛΕΓΚΤΕΣ ATMEΛ



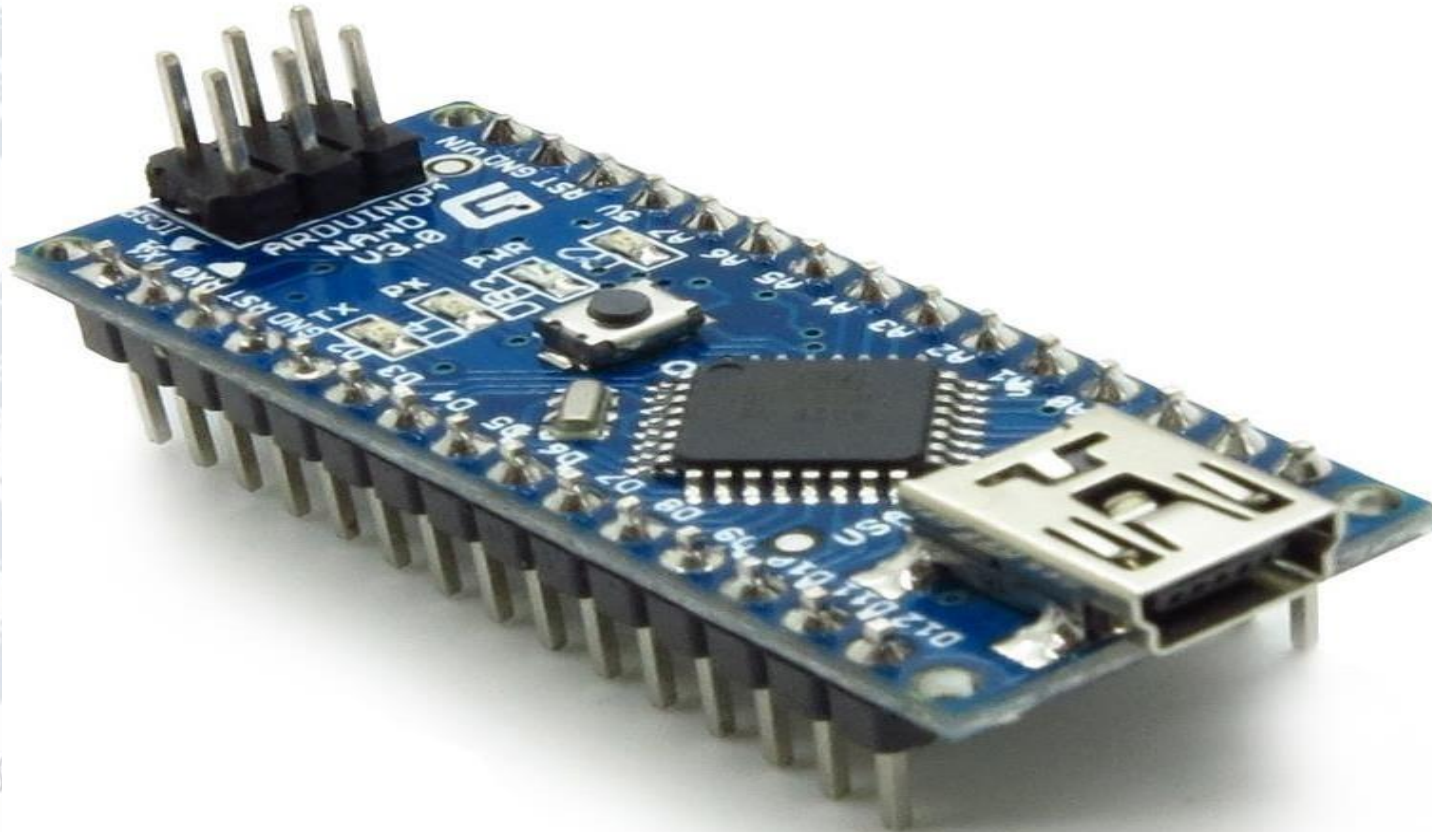
ΜΙΚΡΟΕΛΕΓΚΤΕΣ ΤΗΣ ΑΤΜΕΛ ΠΟΥ ΠΡΟΓΡΑΜΜΑΤΙΖΟΝΤΑΙ ΜΕ ΤΗΝ ΚΑΡΤΑ ΑΝΑΠΤΥΞΗΣ ΕΦΑΡΜΟΓΩΝ SDK 500

AT90S1200, AT90S2313, ATtiny2313,
AT90S2323, AT90S2343, ATtiny12,
ATtiny13, ATtiny22, ATtiny11, ATtiny28,
AT90S4414, AT90S8515, ATmega161,
ATmega8515, ATmega162, AT89S51, AT89S52,
AT90S4434, AT90S8535, ATmega16,
ATmega163, ATmega165, ATmega169,
ATmega32, ATmega323, ATmega8535, AT90S2333,
AT90S4433, ATmega48, ATmega8, ATmega8,
ATmega88, ATmega168, ATtiny26, ATmega103,
ATmega128, AT90CAN128, ATmega64

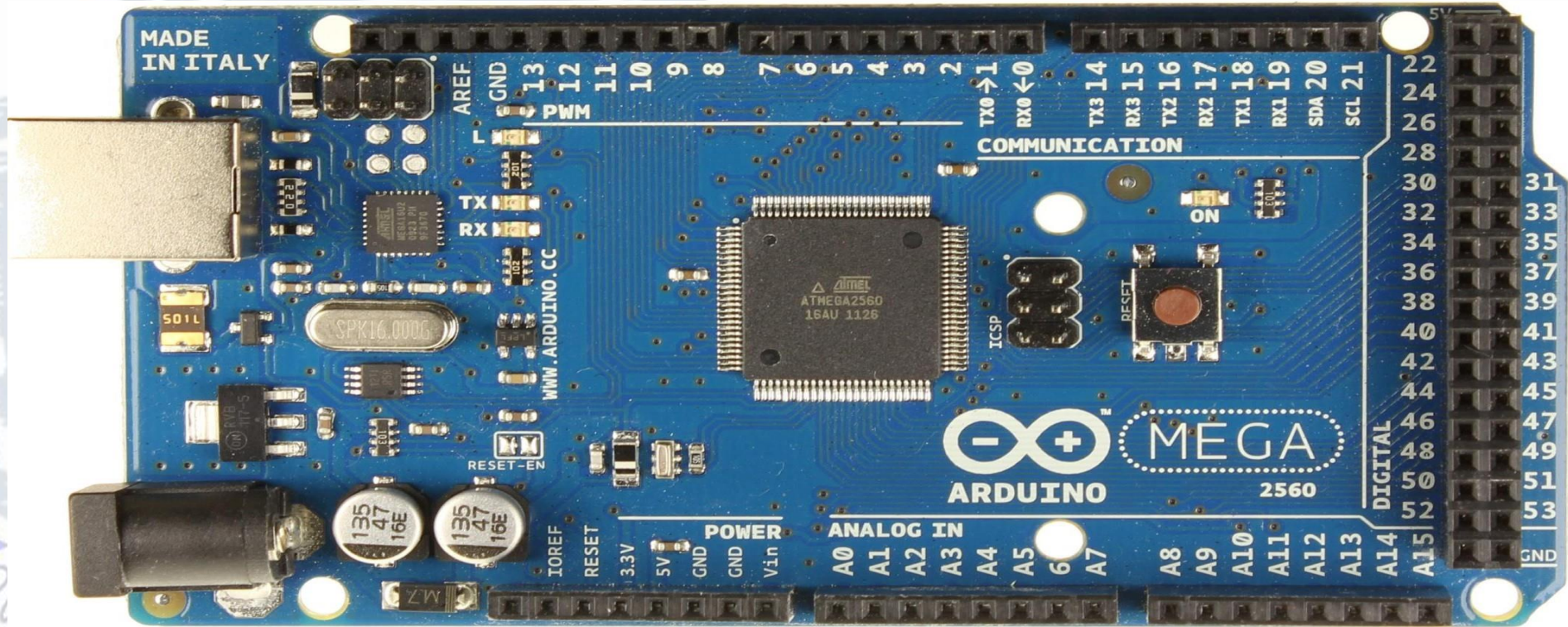
ARDUINO UNO



ARDUINO NANO



ARDUINO MEGA (ATMEGA 1280)

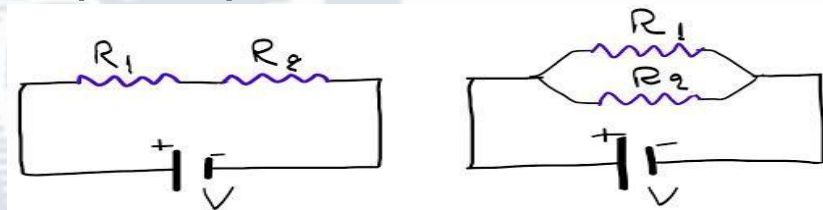


ARDUINO MEGA

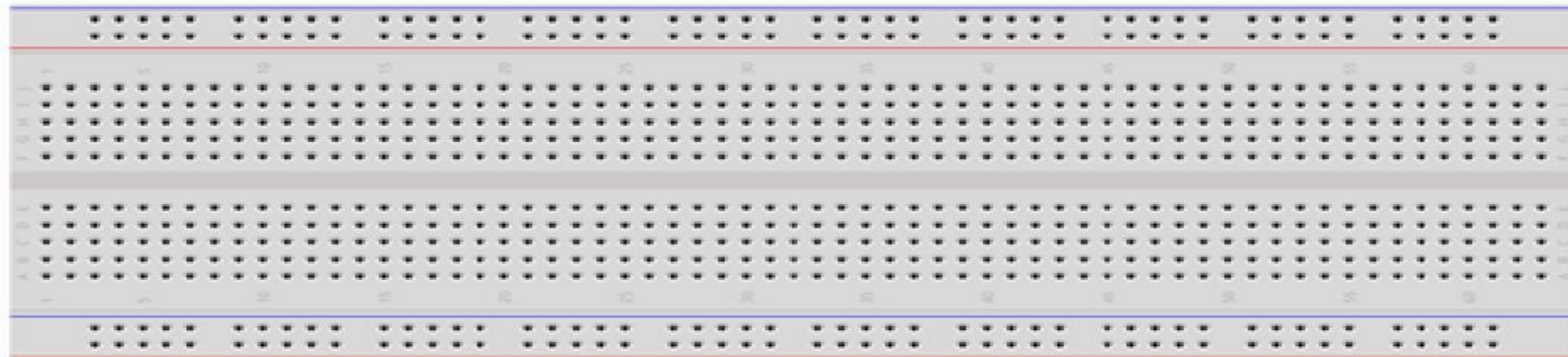
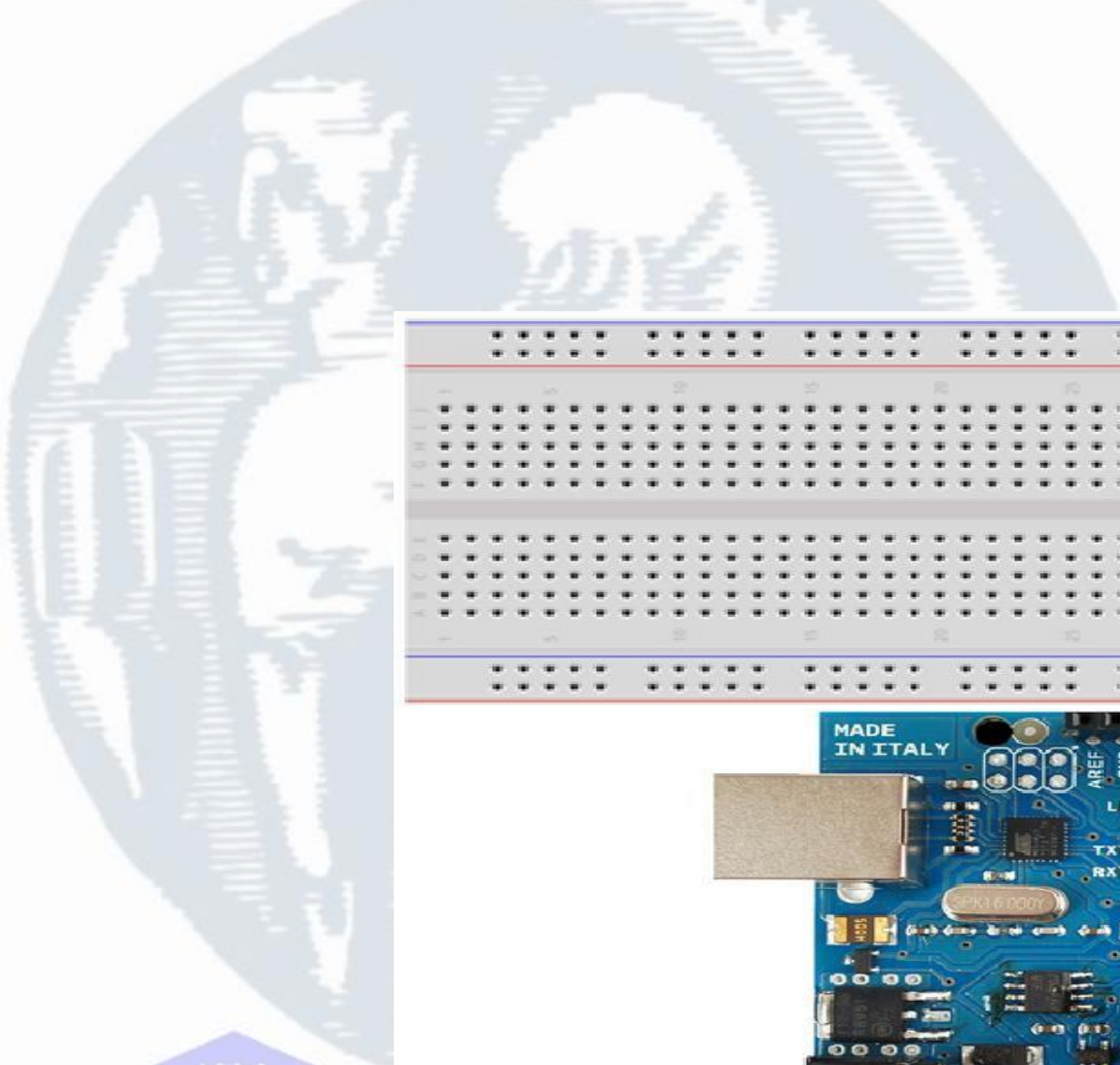
- 54 ψηφιακές εισόδους/εξόδους
(από τις οποίες 14 έξοδοι PWM),
- 16 Αναλογικές εισόδους,
- 4 UARTs (hardware σειριακές πόρτες),
- 16 MHz κρυσταλλικό ταλαντωτή,
- Σύνδεση USB, σύνδεση τροφοδοσίας,
- Σύνδεση ICSP h, reset button.

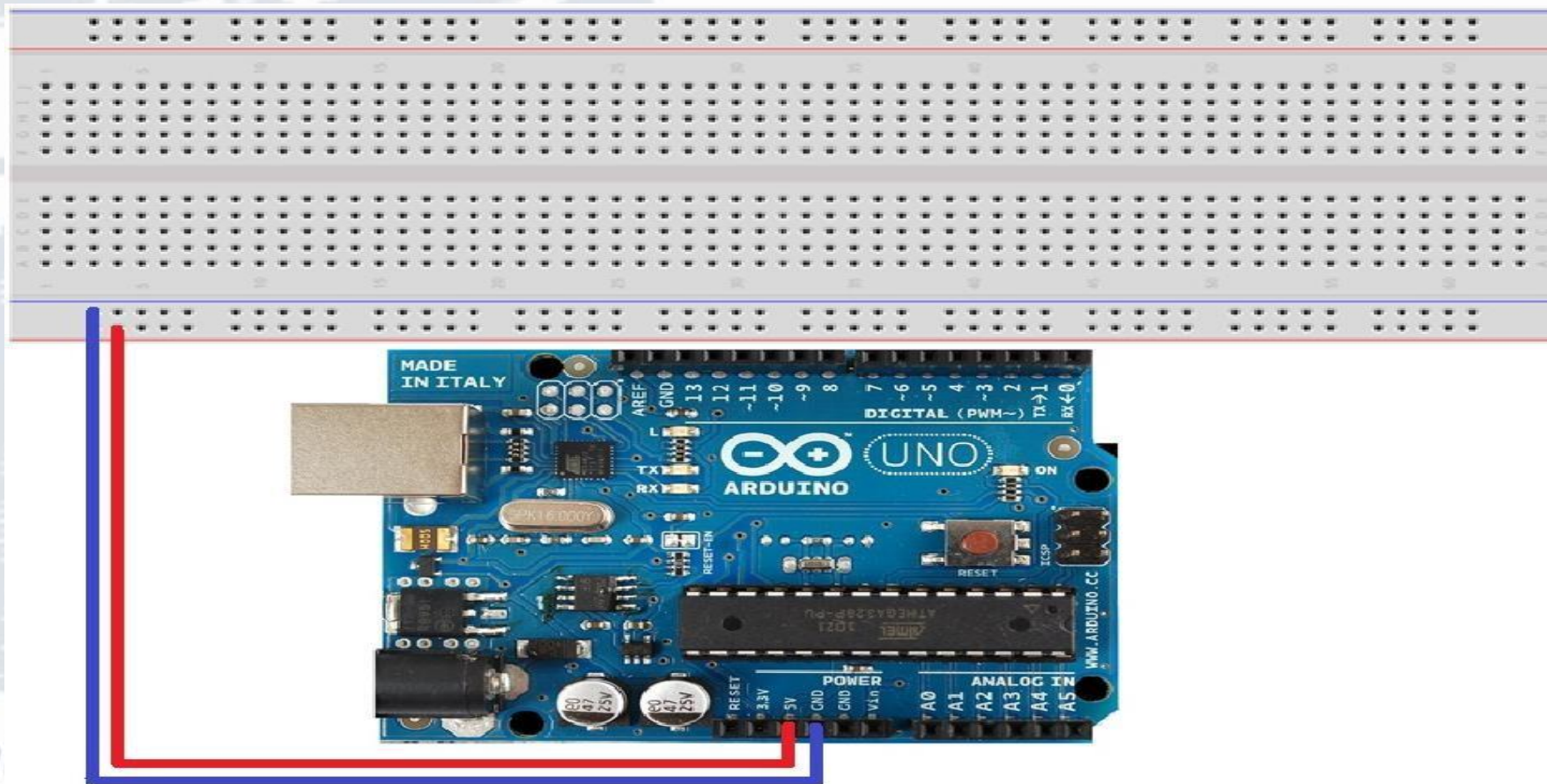
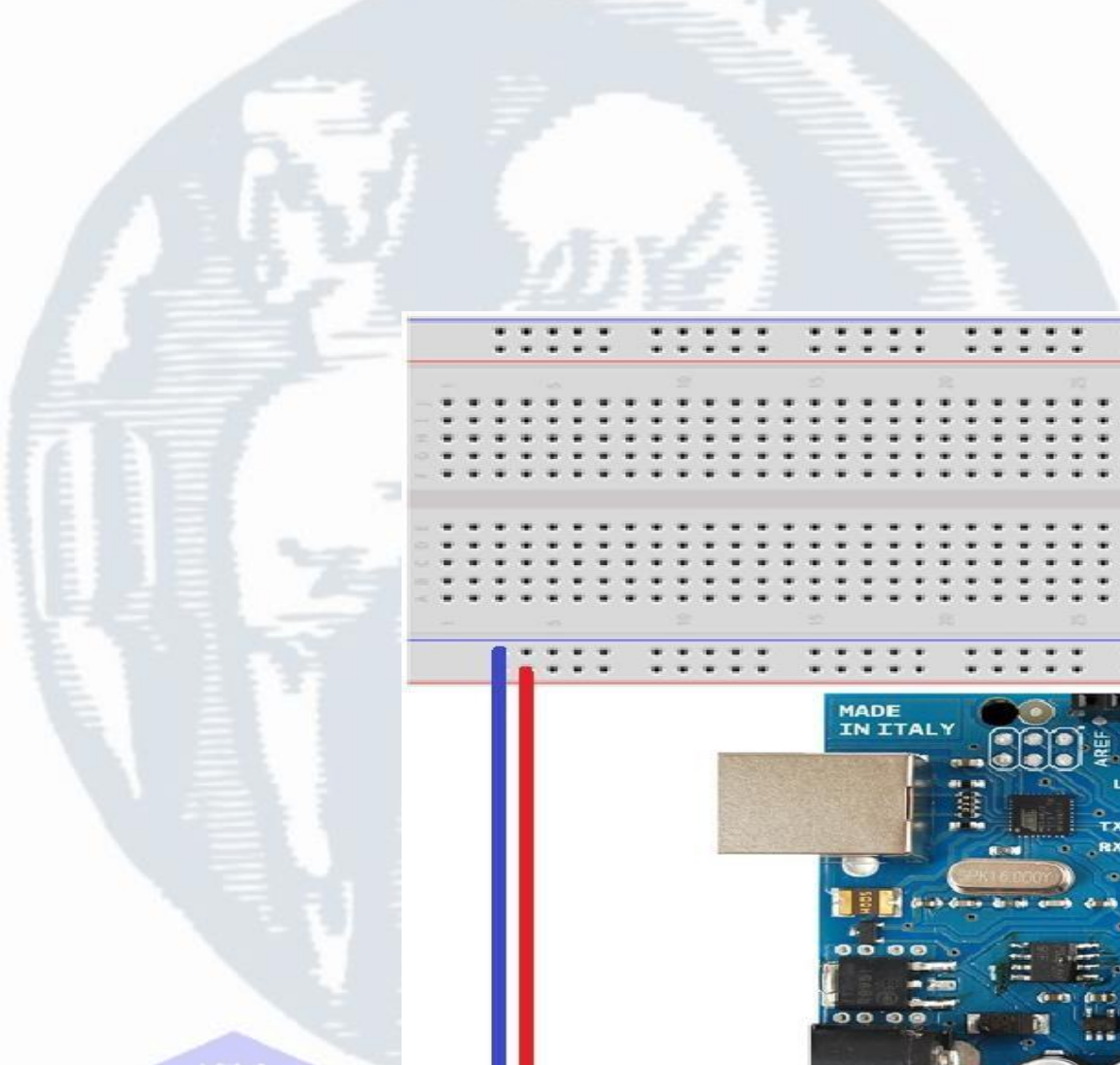
ΡΑΣΤΕΡ, ΓΕΙΩΣΗ ΚΑΙ ΤΡΟΦΟΔΟΣΙΑ

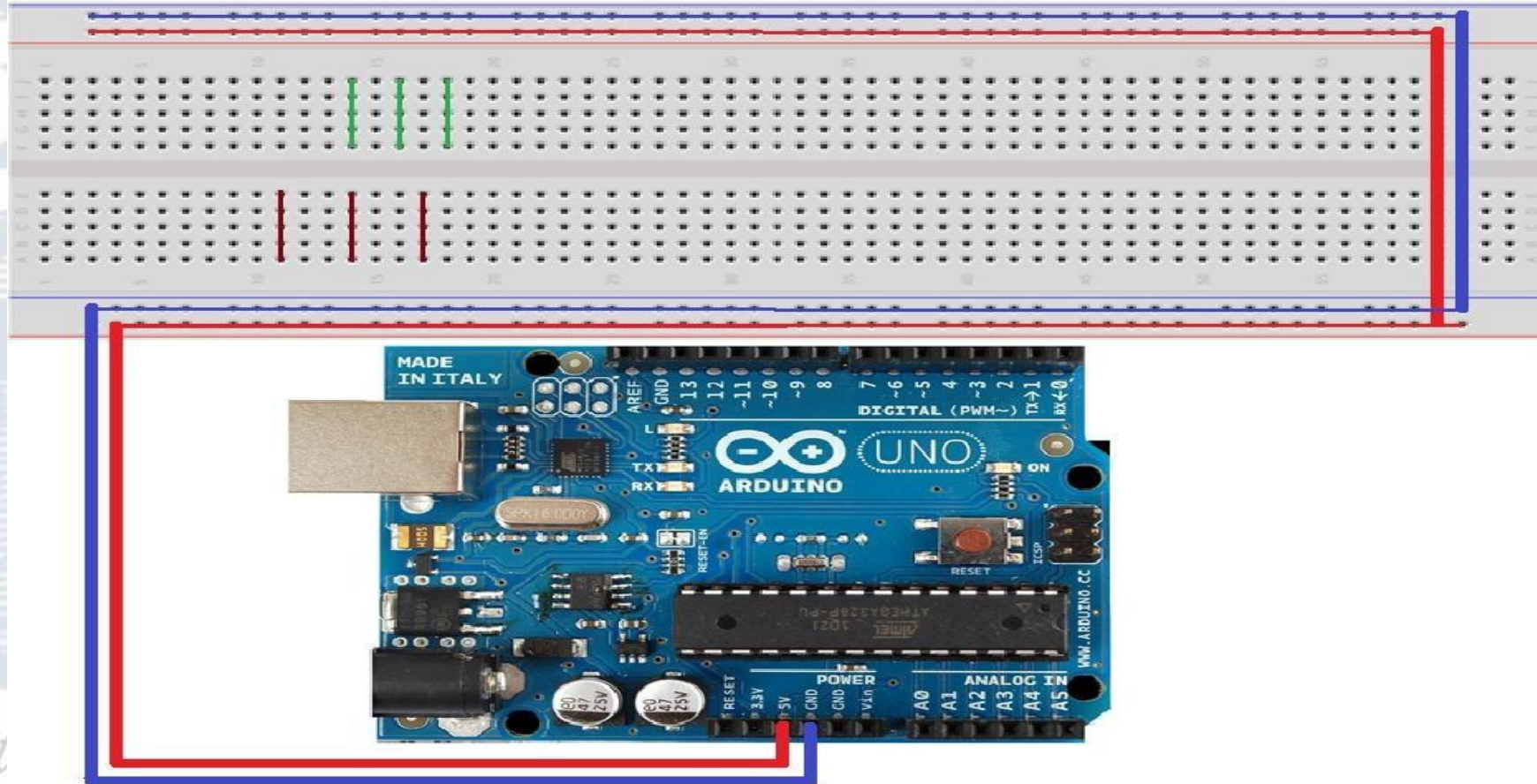
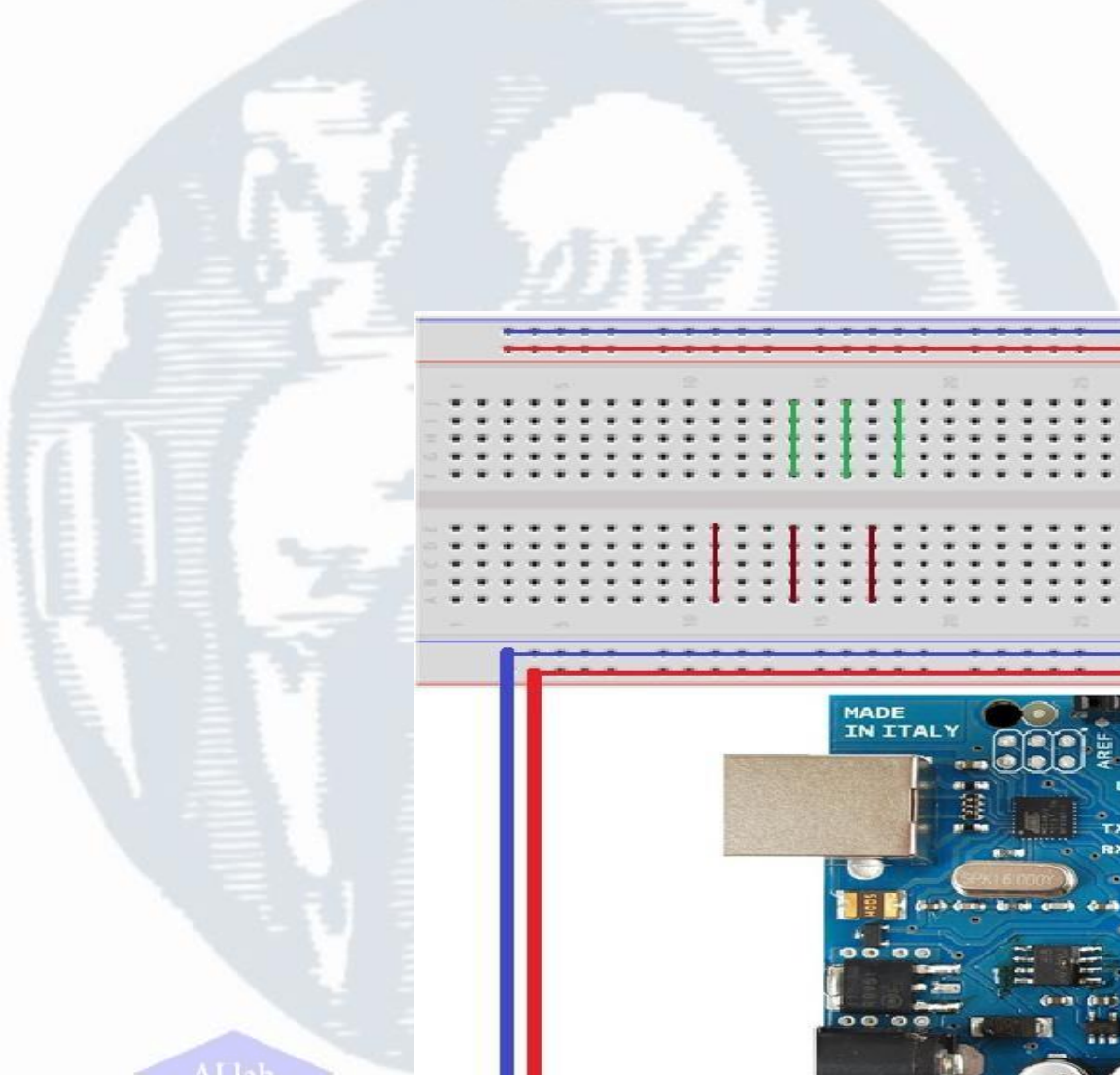
- Το **ράστερ** αποτελείται από πολλές μικρές τρυπούλες, στις οποίες συνδέονται τα **ηλεκτρονικά** στοιχεία. Υπάρχουν δύο τρόποι σύνδεσης των επιμέρους στοιχείων μεταξύ τους, σε σειρά και παράλληλα.



- Οι μεγάλες οριζόντιες γραμμές (κόκκινη και μπλε) δείχνουν τη σύνδεση των τρυπών κάτω από την επιφάνεια του ράστερ σε εκείνη την περιοχή.
- Όλες οι υπόλοιπες τρυπούλες ενώνονται ανά κάθετες πεντάδες. Αυτές αποτελούνε τους κόμβους ουσιαστικά του κυκλώματος μας και με αυτές υλοποιείται το κυρίως κύκλωμα που θέλουμε να κατασκευάσουμε.







ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

Όταν ανοίξουμε το πρόγραμμα βλέπουμε την ακόλουθη εικόνα:



ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

```
//Εδώ δηλώνουμε τυχόν  
βιβλιοθήκες void setup()  
{  
  // put your setup code here, to run once:  
  //Τμήμα 1  
}  
void loop() {  
  // put your main code here, to run repeatedly:  
  //Τμήμα 2  
}
```

Στο τμήμα 1 γράφουμε τα ονόματα των μεταβλητών καθώς και αρχικοποίηση αυτών.

Στο τμήμα 2 βρίσκονται οι εντολές οι οποίες λένε στην κατασκευή μας πώς να λειτουργήσει.

Ένα πρόγραμμα αποτελείται πάντα από δυο βασικά τμήματα (συναρτήσεις):

Τμήμα 1 με ονομασία `setup()`

Κάθε φορά που εκτελείται το πρόγραμμα το τμήμα αυτό εκτελείται μόνο μια φορά.

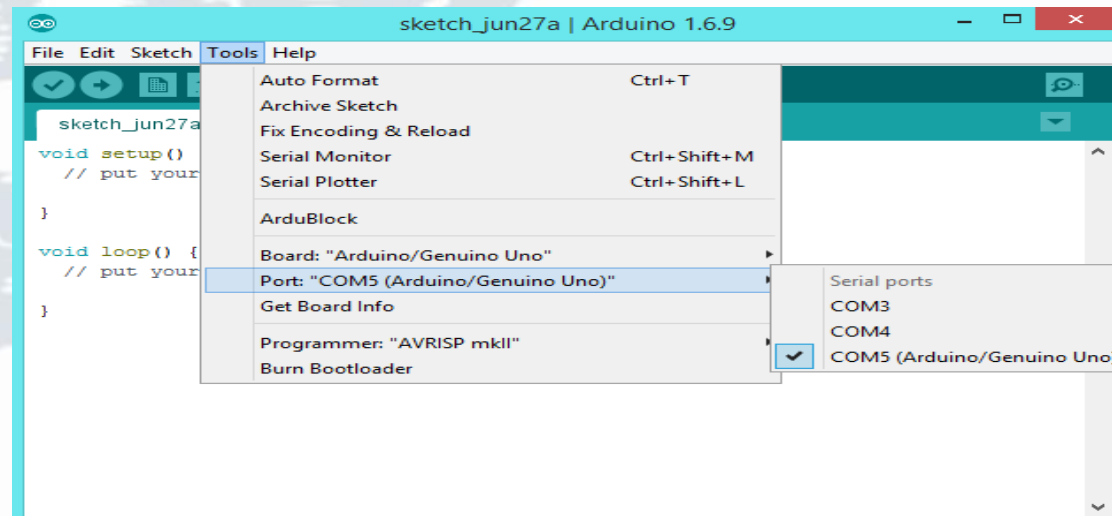
Τμήμα 2 με ονομασία `loop()`:

Κάθε φορά που εκτελείται το πρόγραμμα το τμήμα αυτό εκτελείται συνέχεια.

ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

Πριν γράψουμε οποιοδήποτε πρόγραμμα πρέπει να κάνουμε δύο βασικά βήματα:

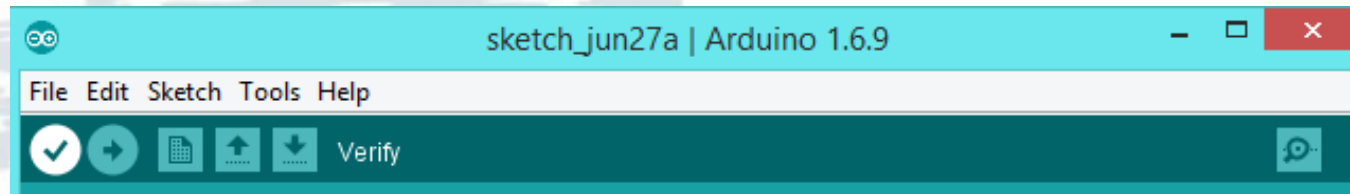
- (α) Να επιλέξουμε την έκδοση του arduino και
- (β) Να συνδέσουμε, δηλαδή να επιλέξουμε την σωστή θύρα επικοινωνίας, το arduino με τον υπολογιστή μας.



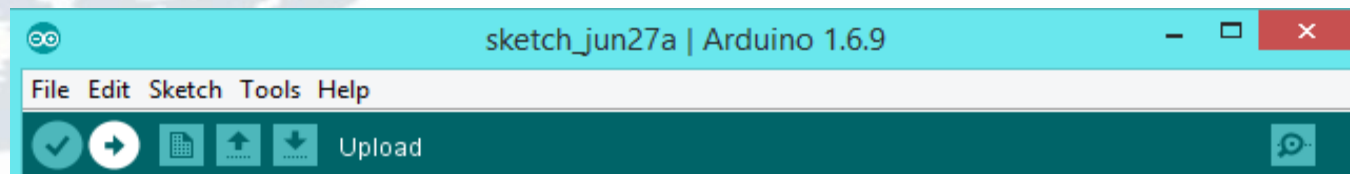
ΛΟΓΙΣΜΙΚΟ (SOFTWARE)

Αφού έχουμε γράψει το πρόγραμμα, πρέπει να δούμε αν έχει συντακτικά λάθη.

Γι' αυτό πατάμε το πλήκτρο της επόμενης εικόνας



Αν δεν υπάρχουν συντακτικά λάθη, τότε φορτώνουμε το πρόγραμμα στο arduino, πατώντας το ακόλουθο πλήκτρο. Αν όλα πάνε καλά, και το πρόγραμμα είναι σωστό, τότε η κατασκευή μας θα κάνει αυτά που τις είπαμε μέσω του προγράμματός μας.



ΠΡΟΣΟΧΗ: κάθε φορά που θέλετε να κάνετε αλλαγή στο κύκλωμα σας, θα πρέπει να αποσυνδέετε το καλώδιο USB από τον υπολογιστή σας.

ΣΥΜΠΕΡΑΣΜΑΤΙΚΑ:

1. Το Arduino είναι ένα «πακέτο» που προσφέρει άμεση αξιοποίηση και προγραμματισμό ενός μικροελεγκτή.
2. Οι πόρτες επικοινωνίας και οι ακίδες του μικροελεγκτή προσφέρονται μέσω της κάρτας για εύκολες συνδέσεις.
3. Ο προγραμματισμός γίνεται μέσω της πόρτας USB.
4. Η ανάπτυξη του κώδικα γίνεται σε γλώσσα τύπου C (Wiring C).



ΒΑΣΙΚΑ ΗΛΕΚΤΡΟΝΙΚΑ ΕΞΑΡΤΗΜΑΤΑ

ΑΝΤΙΣΤΑΣΗ

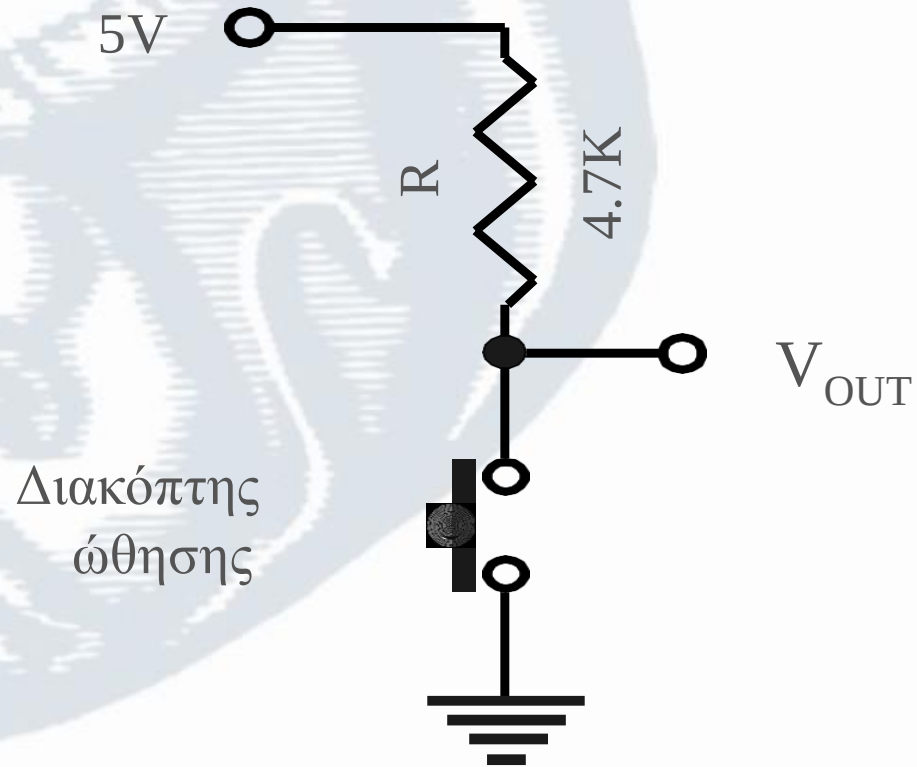


R
27
K

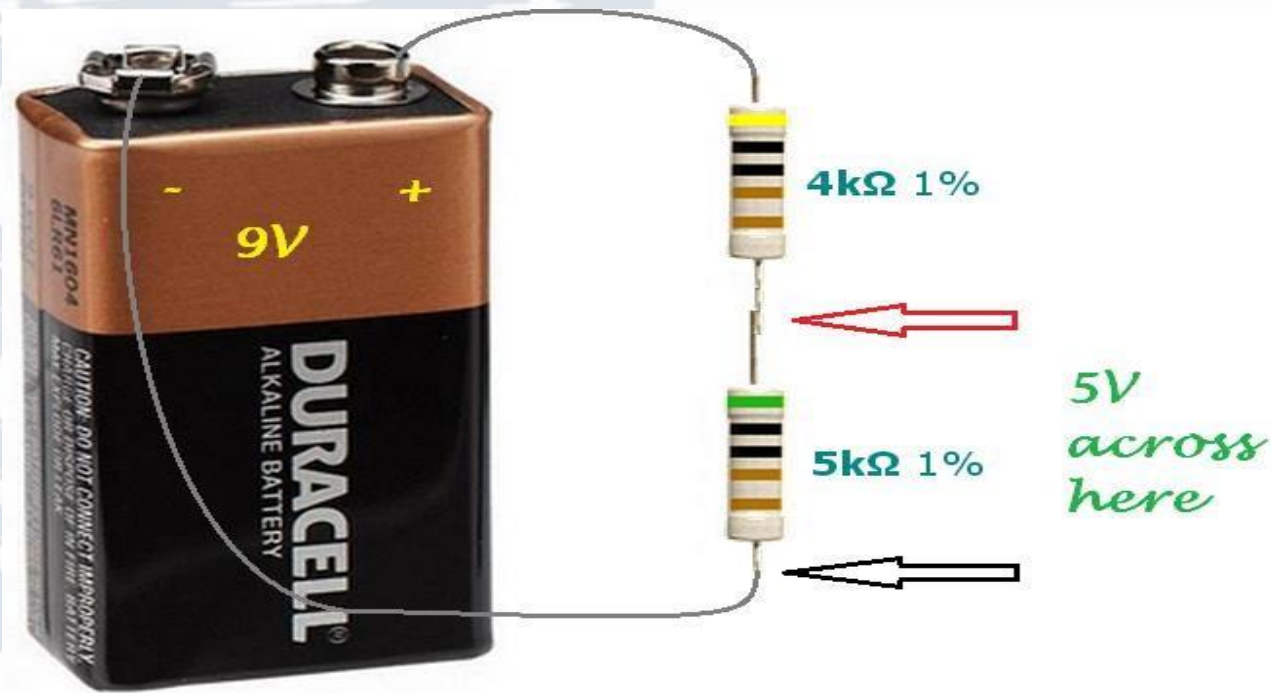


R
27
K

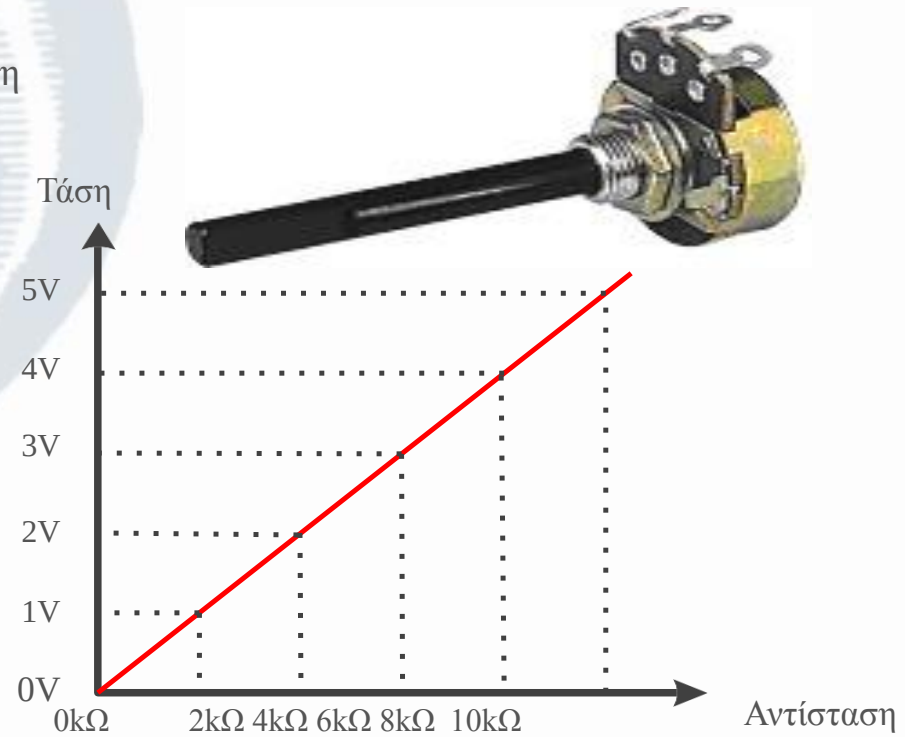
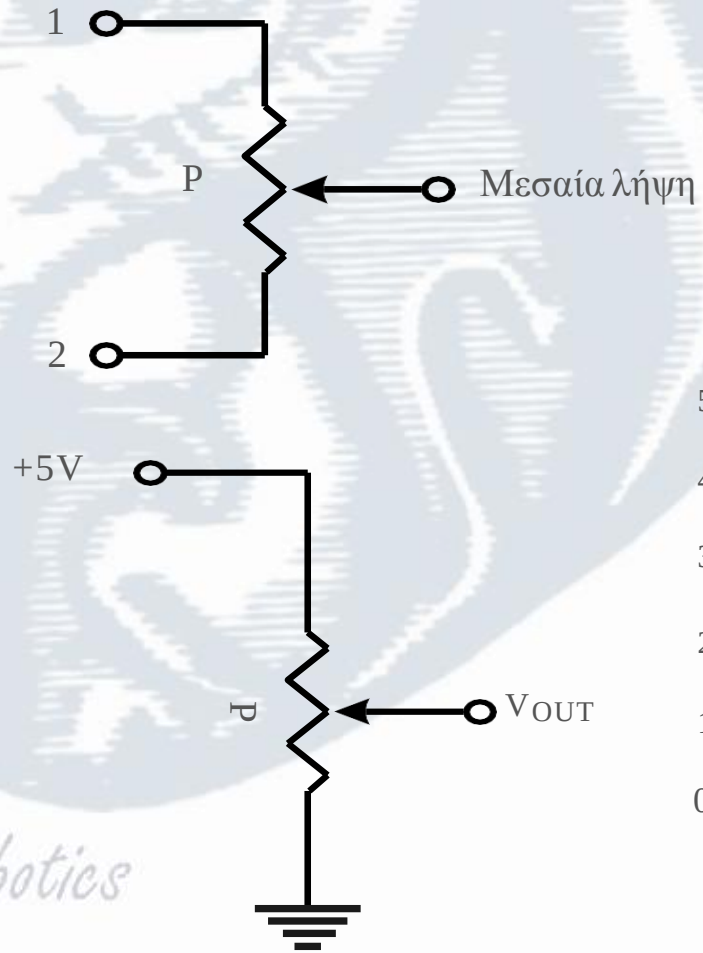
ΔΙΑΚΟΠΤΗΣ ΩΘΗΣΗΣ (PUSH BUTTON)



ΜΠΑΤΑΡΙΑ



ΠΟΤΕΝΣΙΟΜΕΤΡΟ

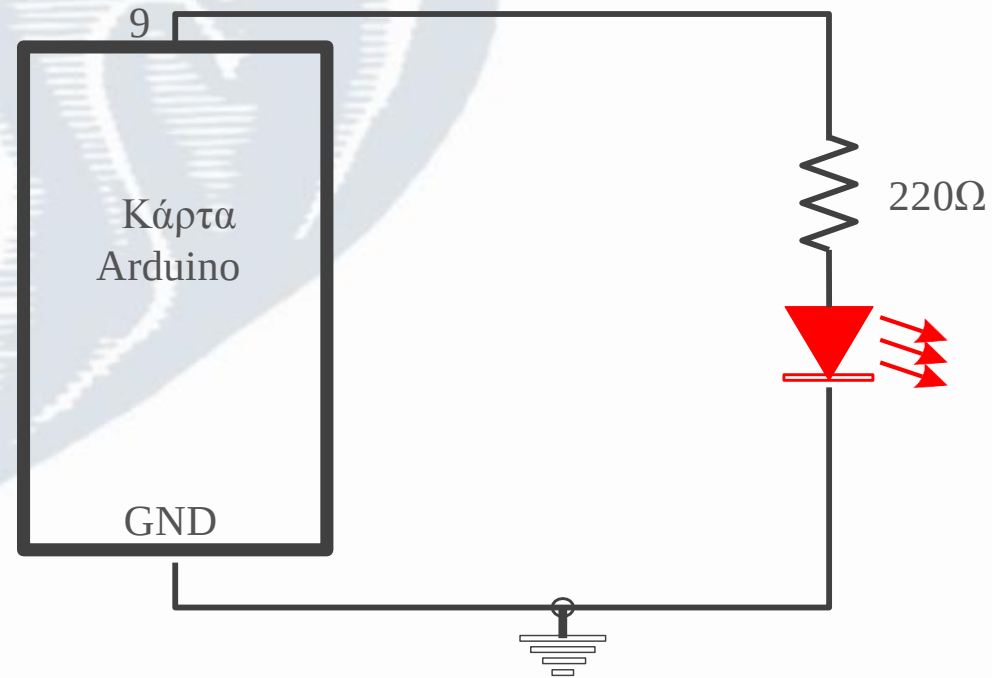




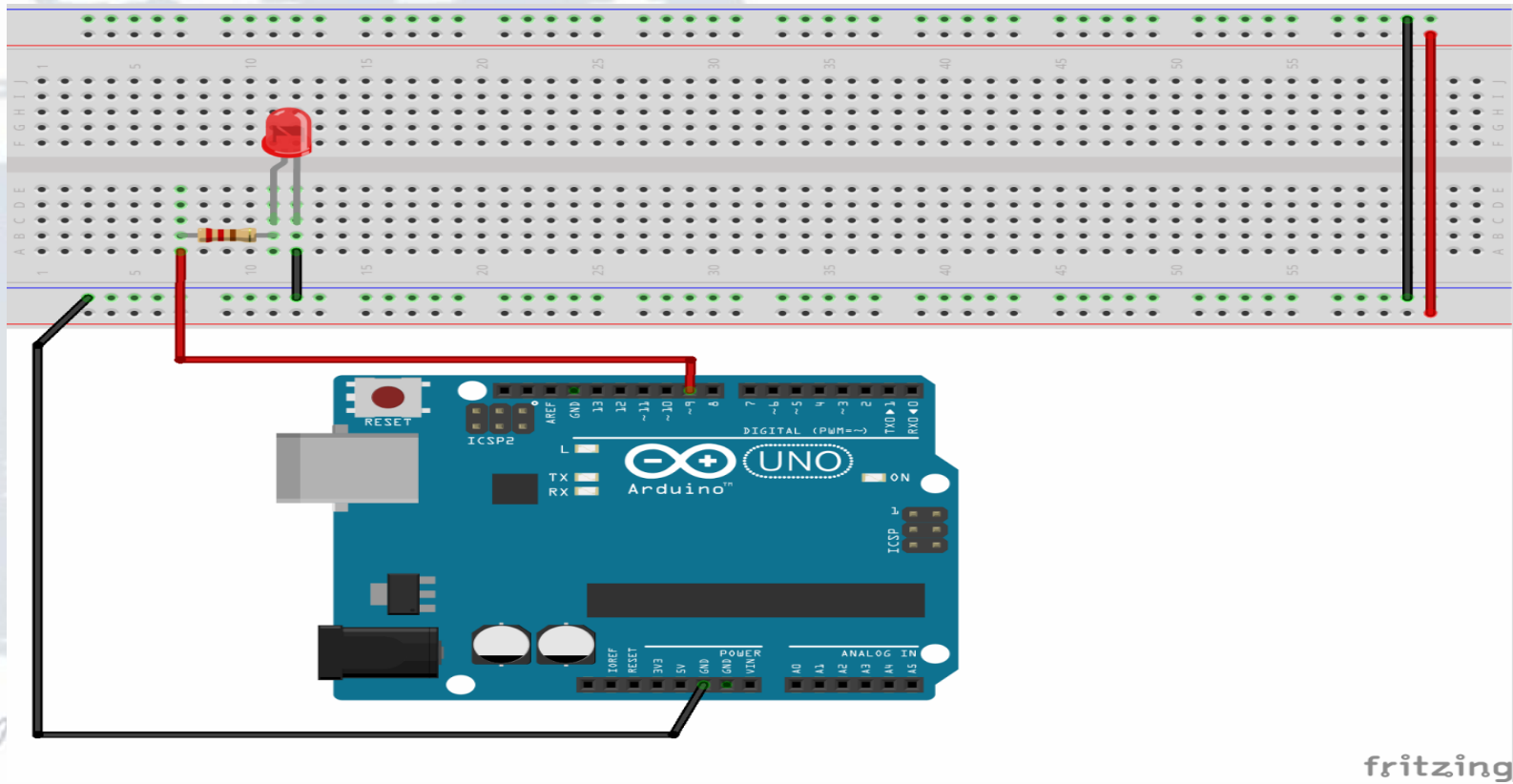
ΠΑΡΑΔΕΙΓΜΑΤΑ ΜΕ ΒΑΣΙΚΑ ΗΛΕΚΤΡΟΝΙΚΑ ΕΞΑΡΤΗΜΑΤΑ

ΠΑΡΑΔΕΙΓΜΑ 1:

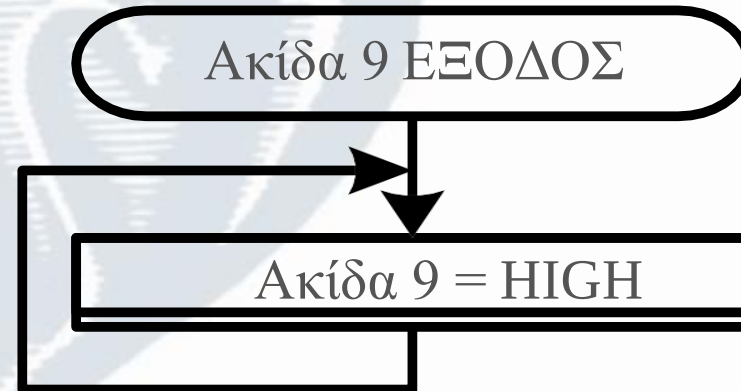
Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί Arduino και θα ανάβει ένα LED



ΥΛΟΠΟΙΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΥΚΛΩΜΑΤΟΣ



ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ



Το πρόγραμμα σε γλώσσα “wiring C”

```
void setup()  
{  
  pinMode( 9 , OUTPUT);  
}
```

```
void loop()  
{  
  digitalWrite( 9 , HIGH );  
}
```


Η χρήση της αντίστασης **περιορισμού ρεύματος R** είναι απαραίτητη προκειμένου να προφυλαχθεί η δίοδος LED από τη διαρροή υψηλού ρεύματος το οποίο θα μπορούσε να την καταστρέψει.

Το κύκλωμα θα διαρρέεται από το ρεύμα I_K .

$$V_{out} = V_R + V_{LED} \text{ ή } V_R = V_{out} - V_{LED}$$

Επομένως:

$$I_K = \frac{V_{out} - V_{LED}}{R}$$

$$I_K = \frac{5V - 2V}{150\Omega} = \frac{3V}{150\Omega} = 0.02A = 20mA$$

Ο ΕΛΕΓΧΟΣ ΤΟΥ ΡΕΥΜΑΤΟΣ ΣΤΟ ΚΥΚΛΩΜΑ ΓΙΝΕΤΑΙ ΜΕΣΩ ΚΑΤΑΛΛΗΛΩΝ ΕΝΤΟΛΩΝ ΚΑΙ ΠΡΟΥΠΟΘΕΤΕΙ ΔΥΟ ΕΝΕΡΓΕΙΕΣ:

- (α) καθορισμό ψηφιακού ακροδέκτη ως εξόδου
- (β) καθορισμό στάθμης σήματος (**0** ή **5V**) στον επιλεγμένο ακροδέκτη

Η εντολή που ρυθμίζει τη μορφή του ακροδέκτη είναι **pinMode** που συντάσσεται ως: **pinMode (pin, mode);**

pin είναι ο ψηφιακός ακροδέκτης (οι διαθέσιμοι ψηφιακοί ακροδέκτες μπορεί να διαφέρουν ανάλογα με την έκδοση του Arduino) και **mode** ο τρόπος που θα χρησιμοποιηθεί (οι βασικότερες επιλογές είναι, **INPUT** για είσοδο και **OUTPUT** για έξοδο).

Στο συγκεκριμένο παράδειγμα, επιλέχθηκε ο ψηφιακός ακροδέκτης 9 και ο καθορισμός του ως εξόδου θα γίνει ως εξής:
pinMode (9,OUTPUT);

Η εντολή που ρυθμίζει το αν θα ανάβει ή θα είναι σβηστό το LED συντάσσεται ως: **digitalWrite(pin, status);**

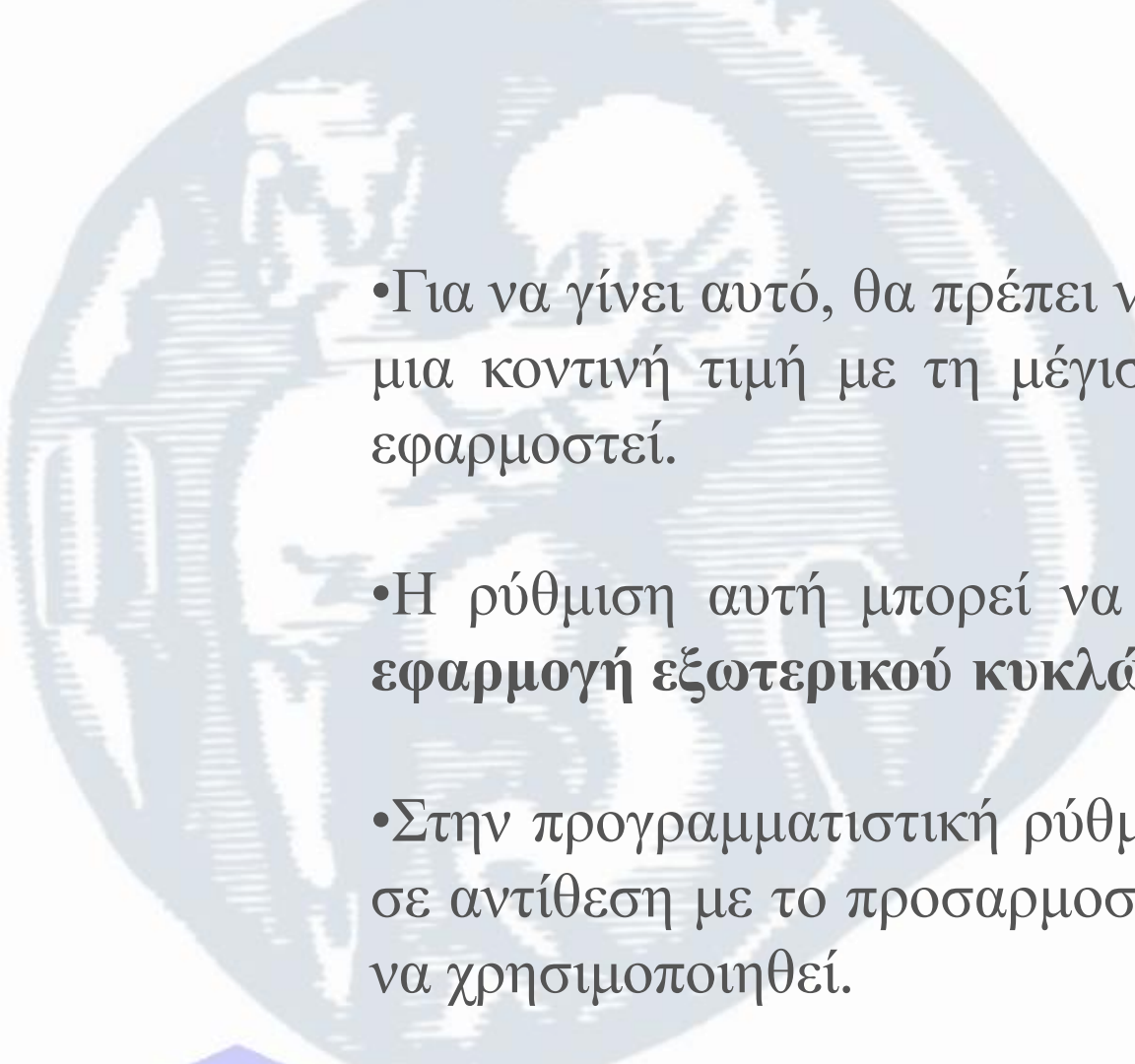
όπου **pin** είναι ο ψηφιακός ακροδέκτης (οι διαθέσιμοι ψηφιακοί ακροδέκτες μπορεί να διαφέρουν ανάλογα με την έκδοση του Arduino) και **status** είναι η στάθμη του σήματος στην ακίδα (**0V για LOW και 5V για HIGH**).

Επομένως η εντολή που θα ανάψει το LED θα είναι η:
digitalWrite(9, HIGH);

Και η εντολή που θα σβήσει το LED θα είναι η:
digitalWrite(9, LOW);

- Το Arduino προσφέρει ένα σύνολο ακροδεκτών που συνδέονται απευθείας με τις αναλογικές εισόδους του μικροελεγκτή.
- Το αναλογικό σήμα εισόδου μπορεί να βρίσκεται στο εύρος 0-5V.
- Ο μικροελεγκτής μετατρέπει τις αναλογικές τάσεις έτσι ώστε να αντιστοιχούν σε 1024 διαφορετικές ψηφιακές στάθμες.
- Άρα διαθέτει ακρίβεια μετατροπής $5V/1024=0.0049V$ ή 49mV.
- Η ακρίβεια αυτή είναι ικανοποιητική για τις περισσότερες εφαρμογές.

- Η ανάγνωση αναλογικού σήματος γίνεται με την εντολή **analogRead(pin)**, όπου pin το αναγνωριστικό της αναλογικής εισόδου.
- Οι δύο επόμενες εντολές είναι ισοδύναμες και αναφέρονται στην ανάγνωση από την αναλογική είσοδο 0: **analogRead(0)**
analogRead(A0)
- Το πρόβλημα που προκύπτει είναι όταν το εύρος των αναλογικών τάσεων είναι περιορισμένο, π.χ. 0-3V.
- Σε μια τέτοια περίπτωση η ανάλυση που προκύπτει θα είναι πάλι η ίδια!, δηλαδή 0.0049.
- Αυτό συμβαίνει διότι ο μικροελεγκτής χρησιμοποιεί εξ' ορισμού ως τάση αναφοράς τα 5V.
- Το ζητούμενο όμως είναι η βελτίωση της ανάλυσης.



• Για να γίνει αυτό, θα πρέπει να προσαρμοστεί η τάση αναφοράς σε μια κοντινή τιμή με τη μέγιστη της αναλογικής εισόδου που θα εφαρμοστεί.

• Η ρύθμιση αυτή μπορεί να γίνει **είτε προγραμματιστικά είτε με εφαρμογή εξωτερικού κυκλώματος.**

• Στην προγραμματιστική ρύθμιση, οι επιλογές είναι περιορισμένες σε αντίθεση με το προσαρμοσμένο εξωτερικό κύκλωμα που μπορεί να χρησιμοποιηθεί.

Η ρύθμιση της τάσης αναφοράς γίνεται με την εντολή **analogReference(Ref_type)**, όπου η παράμετρος **Ref_type** μπορεί να είναι:

DEFAULT

Αντιστοιχεί σε τάση **5V** ή **3.3V** ανάλογα με την έκδοση του Arduino.

INTERNAL

Στις περισσότερες περιπτώσεις αντιστοιχεί στην τιμή **1.1V** (**2.56V** στον μικροελεγκτή ATmega8).

Στην έκδοση Mega χρησιμοποιούνται οι **INTERNAL1V1** και **INTERNAL2V56** αντί της **INTERNAL** για τάσεις αναφοράς **1.1V** και **2.56V** αντίστοιχα.

EXTERNAL

Με αυτή την **παράμετρο**, η τάση αναφοράς ρυθμίζεται από την τάση που εφαρμόζεται στον ακροδέκτη **AREF**. Πάντως, αυτή η τάση δεν μπορεί να είναι μικρότερη από 0V ή μεγαλύτερη από 5V.

Η τάση αυτή μπορεί να προσαρμοστεί εύκολα στο επιθυμητό επίπεδο χρησιμοποιώντας ένα **διαιρέτη τάσης**.

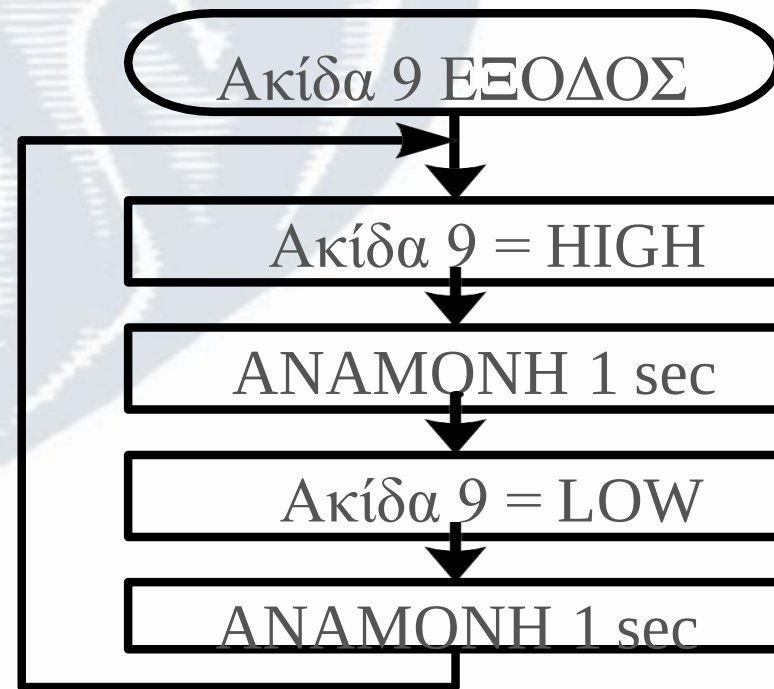
Έτσι, με τάση αναφοράς τα 3V, η αντίστοιχη ανάλυση θα είναι $3/1024=0.003$, δηλαδή πολύ καλύτερη από την 0.0049 στην περίπτωση της τάσης αναφοράς των 5V.

Με αυτό τον τρόπο επιτυγχάνεται μεγαλύτερη ευχέρεια διαχείρισης ενός αναλογικού σήματος που κινείται στην περιοχή 0 έως 3V.

ΠΑΡΑΔΕΙΓΜΑ 2:

Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί Arduino και θα αναβοσβύνει ένα LED

Λογικό Διάγραμμα Προγράμματος



Το πρόγραμμα σε γλώσσα “wiring C”

```
void setup()
{
  pinMode( 9 , OUTPUT);
}
void loop()
{
  digitalWrite( 9 , HIGH );
  delay( 1000 );
  digitalWrite( 9 , LOW );
  delay( 1000 );
}
```

ΠΑΡΑΔΕΙΓΜΑ 3:

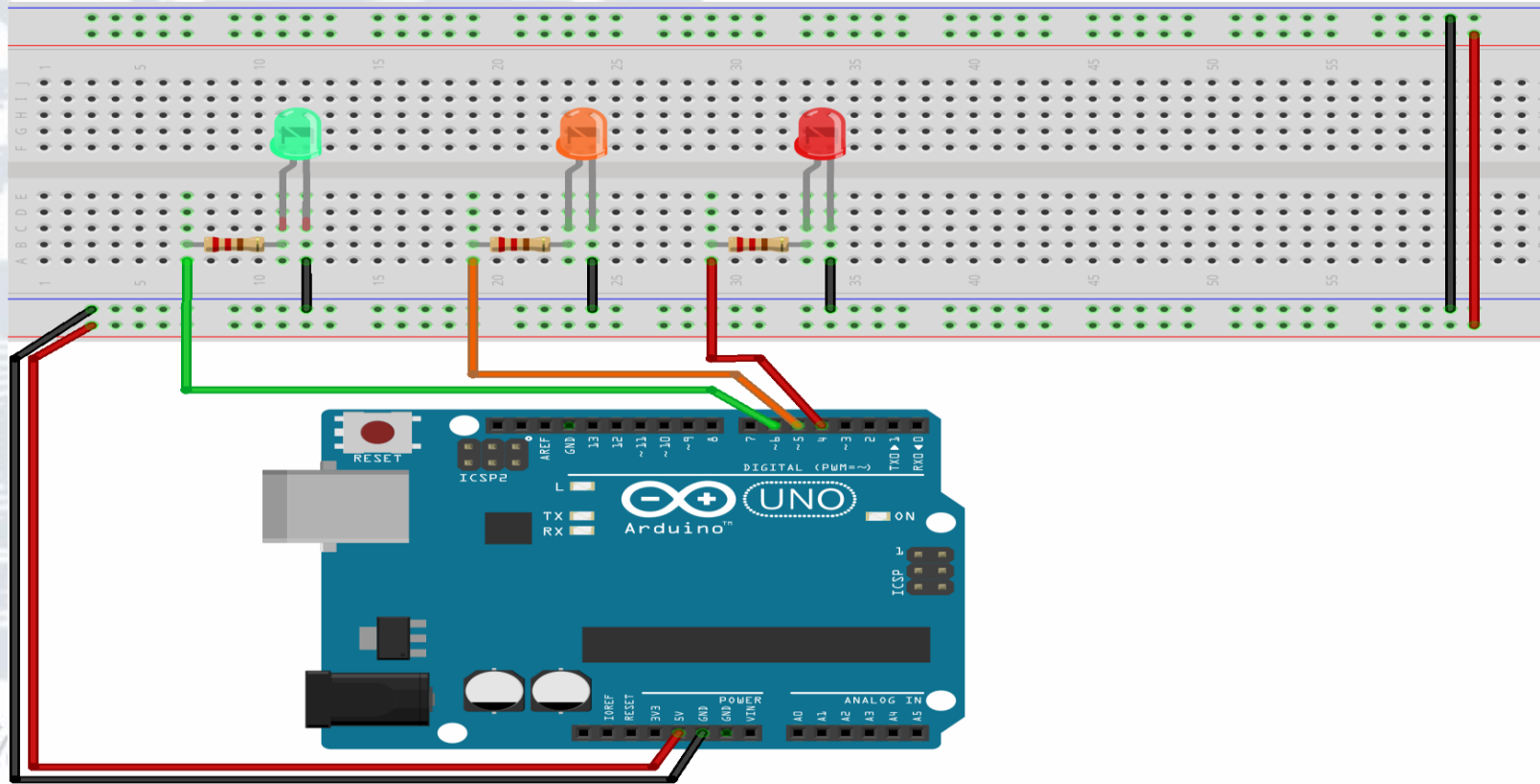
Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί Arduino και θα ανάβει τα LED ενός φωτεινού σηματοδότη με βάση τους παρακάτω χρόνους.

Πράσινο 10sec

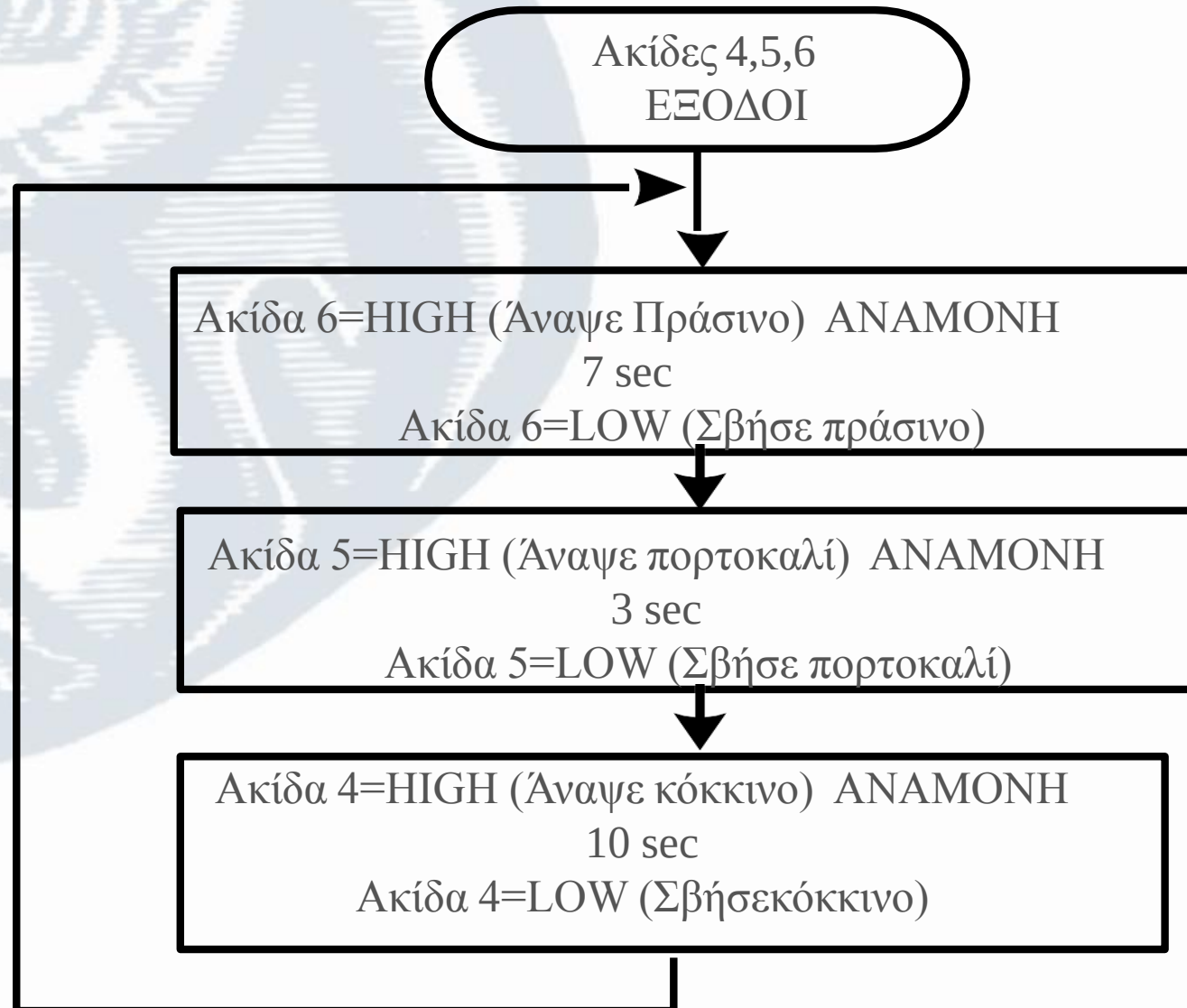
Πορτοκαλί 3 sec

Κόκκινο 7 sec

ΥΛΟΠΟΙΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΥΚΛΩΜΑΤΟΣ



ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΤΟ ΠΡΟΓΡΑΜΜΑ ΣΕ ΓΛΩΣΣΑ “WIRING C”

- **int RledPin = 4;**
int OledPin = 5; int
GledPin = 6;

- **unsigned int RDelay = 10000;**
unsigned int ODelay= 3000;
unsigned int GDelay = 7000;

- **void setup()**

- **{**

- **pinMode(RledPin, OUTPUT);** //Κάνε τις ακίδες

- **pinMode(OledPin, OUTPUT);** //4, 5 και 6

- **pinMode(GledPin, OUTPUT);** //εξόδους

- **}**

// Ονόμασε ακίδα 4 RledPin

// Ονόμασε ακίδα 5 OledPin

// Ονόμασε ακίδα 6 GledPin

//Χρονική καθυστέρηση για το κόκκινο

//Χρονική καθυστέρηση για το πορτοκαλί

//Χρονική καθυστέρηση για το πράσινο

//Συνάρτηση αρχικοποίησης



```
void loop()
{
  digitalWrite( GledPin, HIGH); delay(
  GDelay); digitalWrite(GledPin,
  LOW); digitalWrite( OledPin,
  HIGH); delay( ODelay );
  digitalWrite(    OledPin,    LOW);
  digitalWrite(    RledPin,    HIGH);
  delay(RDelay);
  digitalWrite( RledPin, LOW);
}
```

//Κύριος βρόχος

//Αναψε το πράσινο LED

//για 7sec και

//σβήσε το.

//Αναψε το πορτοκαλί LED

//για 3sec και

//σβήσε το.

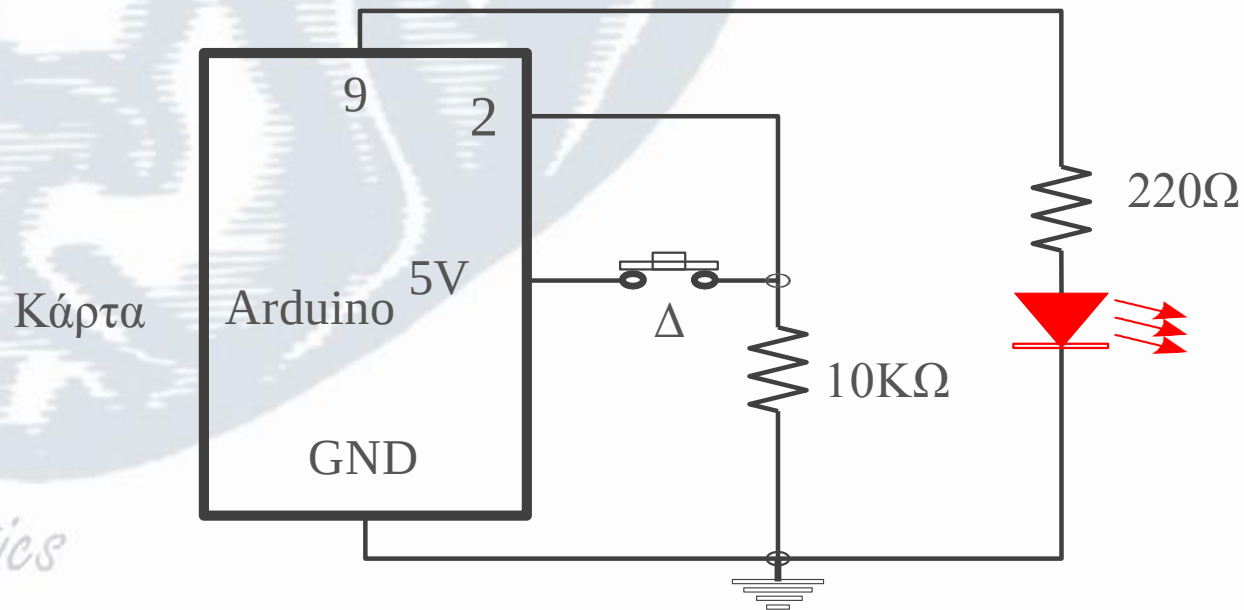
//Αναψε το πράσινο LED

//για 10sec και

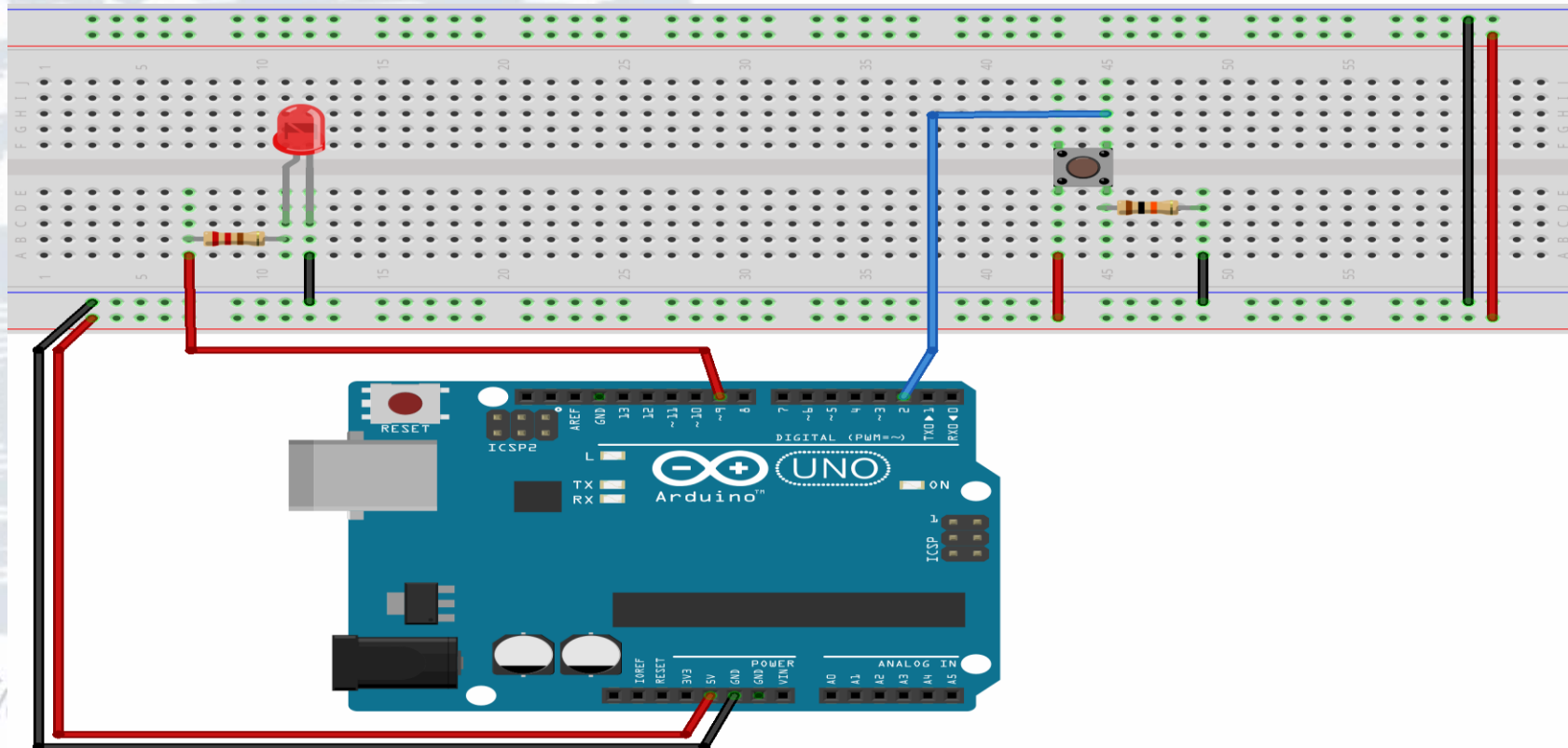
//σβήσε το.

ΠΑΡΑΔΕΙΓΜΑ 4:

Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί arduino και θα αναβοσβήνει ή θα κρατά σβηστό ένα LED κάτω από τον έλεγχο ενός διακόπτη ώθησης (push button).

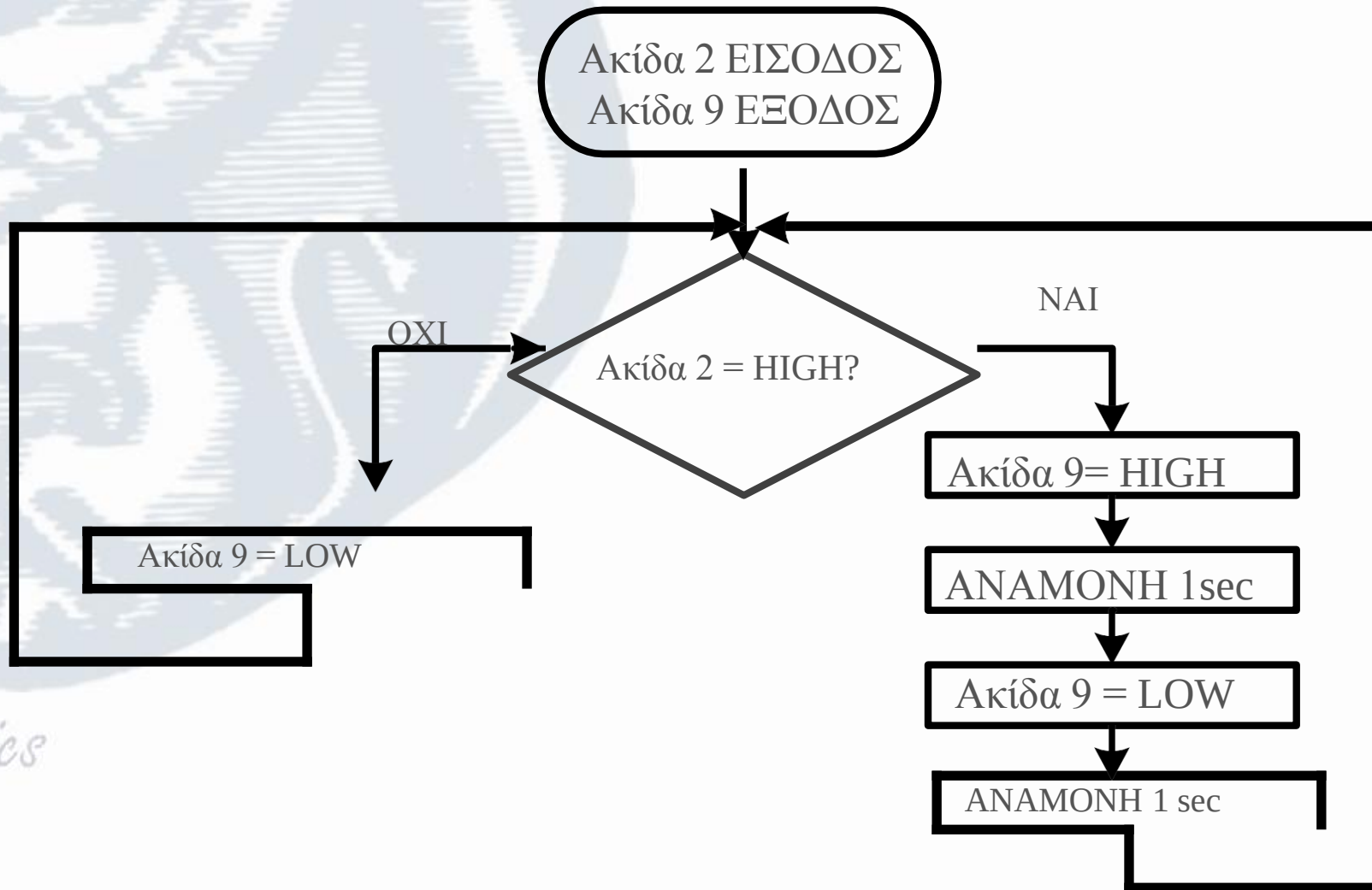


ΥΛΟΠΟΙΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΥΚΛΩΜΑΤΟΣ



fritzing

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΤΟ ΠΡΟΓΡΑΜΜΑ ΣΕ ΓΛΩΣΣΑ “WIRING C”

```
int pinButton = 2;  
int ledPin = 9;
```

```
void setup()  
{  
  pinMode( pinButton, INPUT );  
  pinMode( ledPin, OUTPUT );  
}
```

```
//Ονόμασε ακίδα 2 pinButton  
//Ονόμασε ακίδα 9 ledPin
```

```
//Συνάρτηση αρχικοποίησης
```

```
//Κάνε ακίδα 2 είσοδο  
//και ακίδα 9 έξοδο
```



```
void loop()  
{  
stateButton = digitalRead( pinButton );  
if ( stateButton == 1 )  
{  
digitalWrite( ledPin , HIGH ); delay(  
1000 ); digitalWrite(ledPin , LOW );  
delay( 1000 );  
}  
else  
digitalWrite( ledPin, LOW );
```

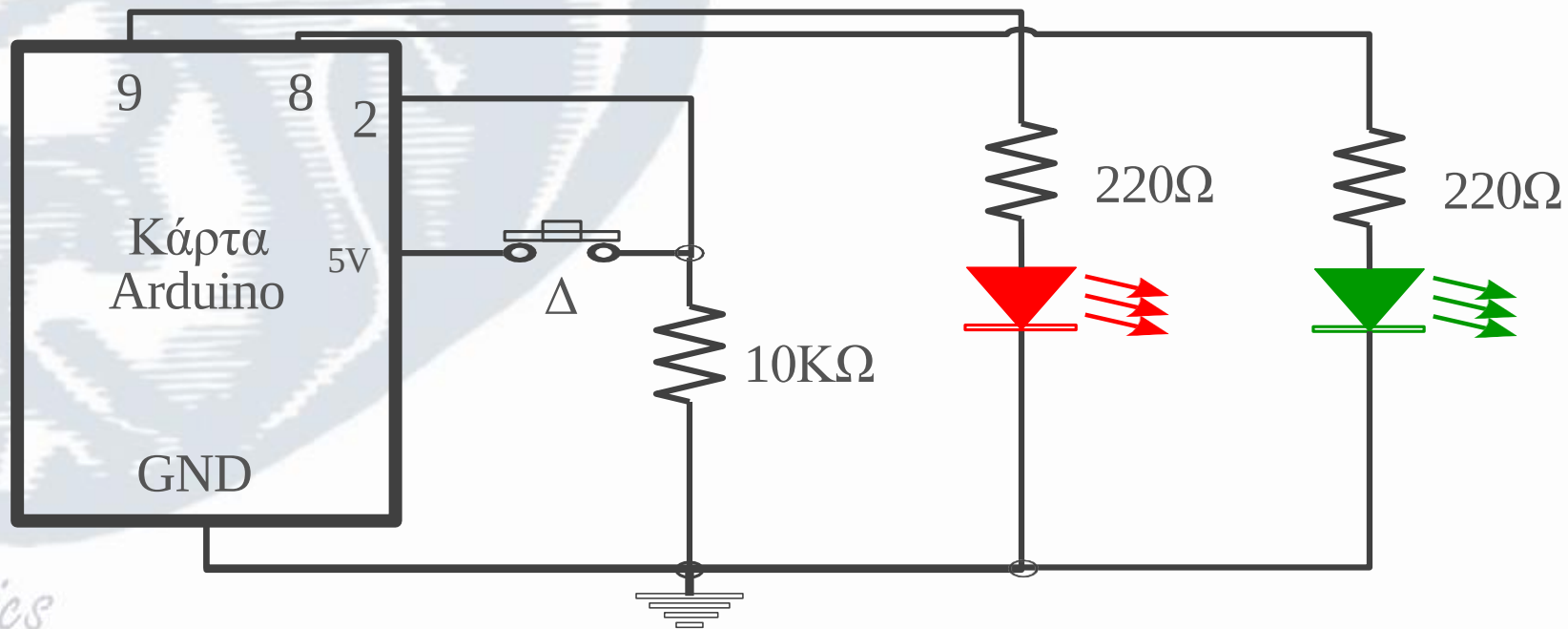
```
//Κύριος βρόχος  
//Διάβασε την κατάσταση  
//του διακόπτη και  
//αν έχει κλείσει ο διακόπτης
```

```
//αναβόσβησε το LED
```

```
//διαφορετικά  
//σβήσε το LED
```

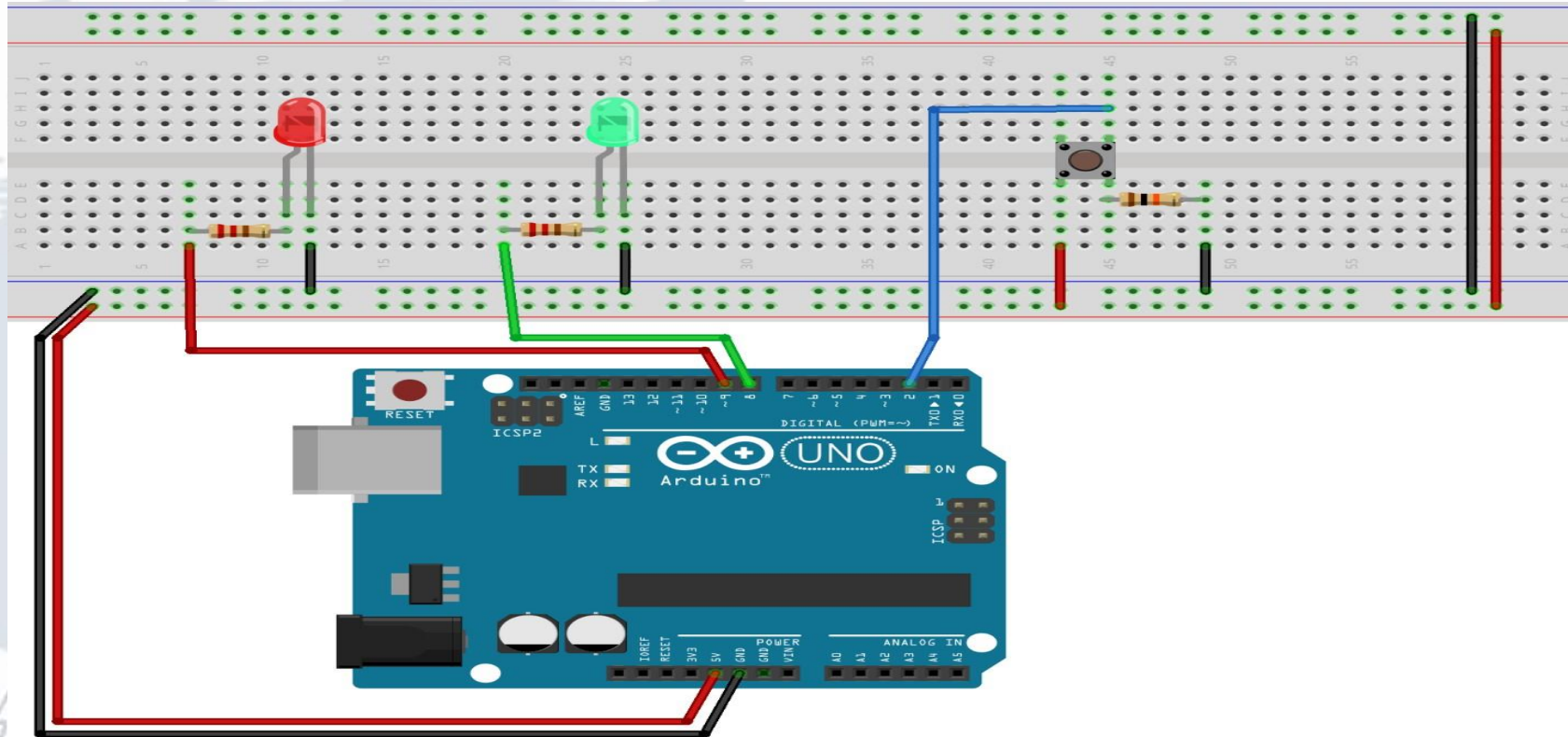

ΠΑΡΑΔΕΙΓΜΑ 6:

Να δημιουργηθεί ένα κύκλωμα που θα χρησιμοποιεί Arduino και θα αναβοσβήνει είτε ένα κόκκινο είτε ένα πράσινο LED κάτω από τον έλεγχο ενός διακόπτη ώθησης (push button)



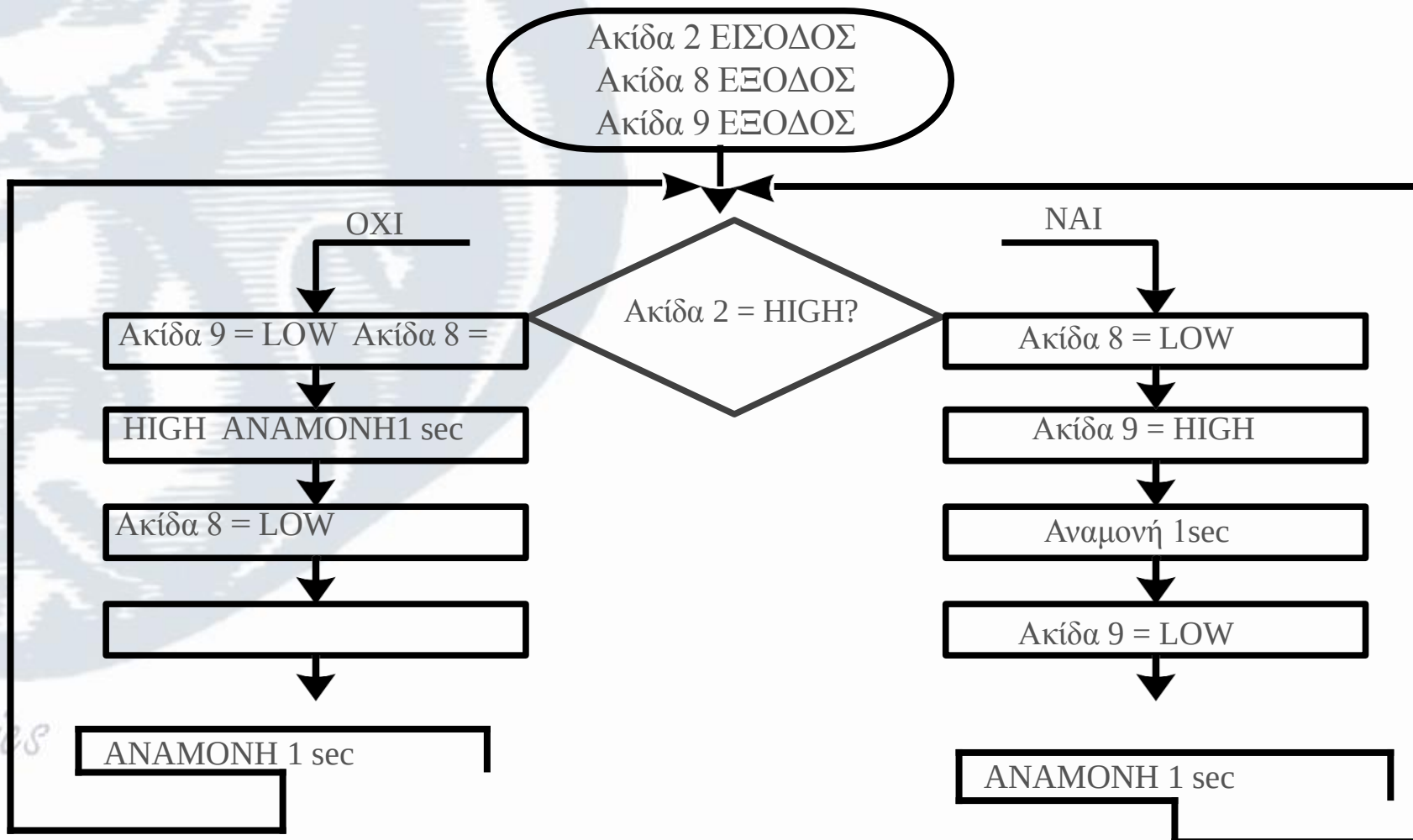
Ηλεκτρονικό Κύκλωμα

ΥΛΟΠΟΙΗΣΗ ΗΛΕΚΤΡΟΝΙΚΟΥ ΚΥΚΛΩΜΑΤΟΣ



fritzing

ΛΟΓΙΚΟ ΔΙΑΓΡΑΜΜΑ ΠΡΟΓΡΑΜΜΑΤΟΣ



ΤΟ ΠΡΟΓΡΑΜΜΑ ΣΕ ΓΛΩΣΣΑ “WIRING C”

```
int buttonPin = 2;  
int GledPin = 8;  
int RledPin = 9;
```

```
void setup()  
{  
  pinMode( buttonpin, INPUT);  
  pinMode( GledPin , OUTPUT);  
  pinMode( RledPin , OUTPUT);  
}
```

```
//Ονόμασε ακίδα 2 buttonPin  
//Ονόμασε ακίδα 8 GledPin  
//Ονόμασε ακίδα 9 RledPin
```

```
//Κάνε την ακίδα 2 είσοδο  
//και τις ακίδες 8 και 9  
// εξόδους
```




```
void loop()
{
  if (digitalRead(buttonPin))
  {
    digitalWrite( RledPin, LOW );
    digitalWrite( GledPin , HIGH );
    delay( 1000 );
    digitalWrite( GledPin , LOW );
    delay( 1000 );
  }
  else
  {
    digitalWrite(  GledPin  , LOW  );
    digitalWrite( RledPin , HIGH );
    delay( 1000 );
    digitalWrite( RledPin , LOW );
    delay( 1000 );
  }
}
```

//Αν ή ακίδα 2 είναι HIGH

//Αναβόσβησε πράσινο LED

//Διαφορετικά

//Αναβόσβησε κόκκινο LED



ΕΥΧΑΡΙΣΤΟΥΜΕ ΠΟΛΥ!!!