

# Τρίτη Εργαστηριακή Συνάντηση

## εμπέδωση βασικών στοιχείων και διαδικασιών

*Άνθιμος Χαλκίδης, Αρτεμησία Στούμπα,  
Ηλίας Μπόικος, Αριστοτέλης Γκιόλμας*

ΠΜΣ Εκπαίδευση STEM και Συστήματα Εκπαιδευτικών Ρομποτικών Διατάξεων / ΠΤΔΕ ΕΚΠΑ

**Εργαστήριο (Εκπαιδευτικής) Ρομποτικής I**

Νοέμβριος 2022

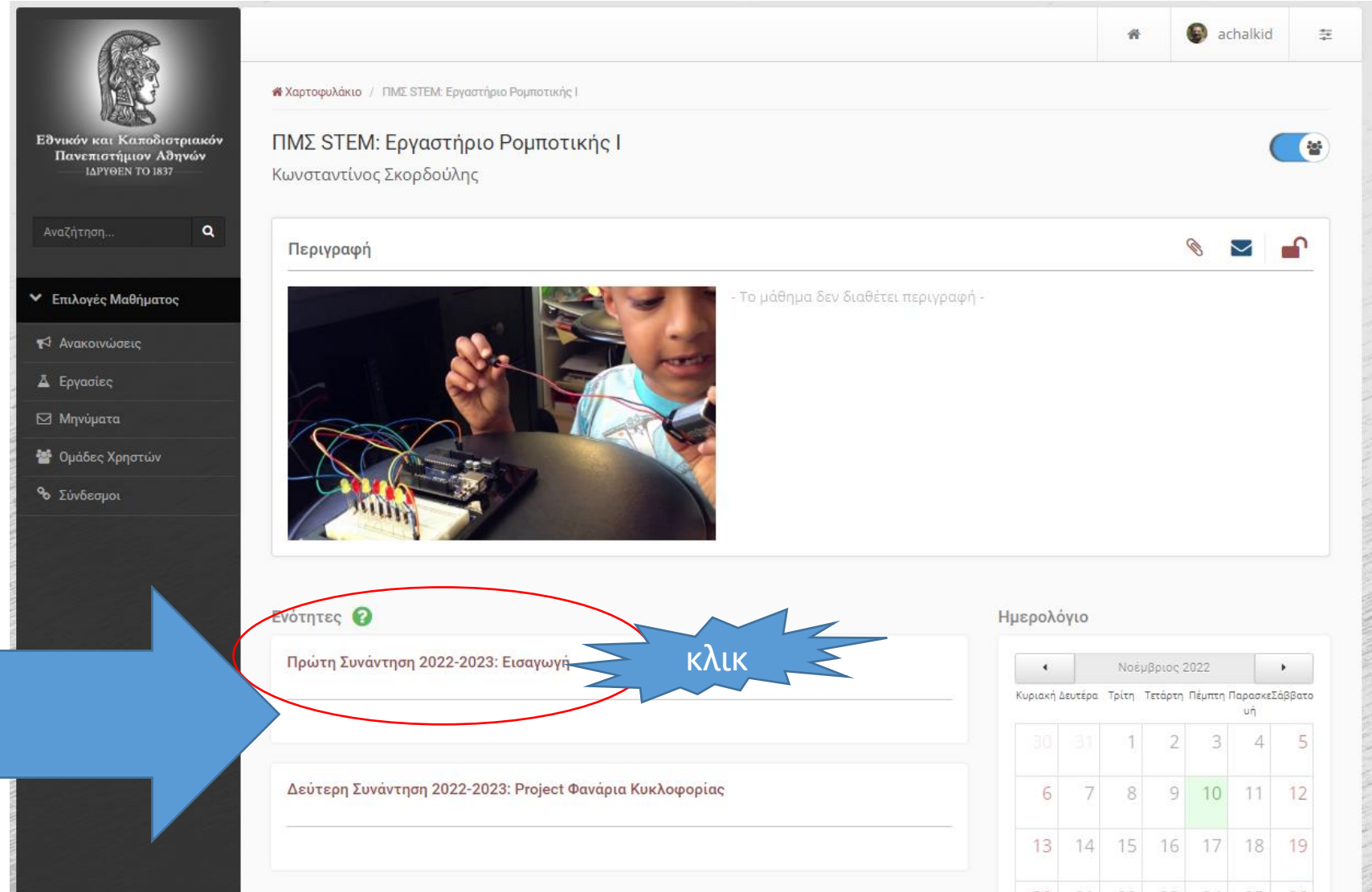
# Οργανωτικά ξανά

Γιατί τα κάνουμε έτσι

- Φύλλο Εργαστηριακής άσκησης  
(το ακολουθούμε, το συμπληρώνουμε, το παραδίνουμε)
- E-class  
(θα δούμε λίγο την δομή της)
- Οριστικοποίηση ομάδων
- Τάξεις *(και ηθική)*  
(βοηθάμε στην τακτοποίηση των υλικών που χρησιμοποιήσαμε)

# E-class

- Η βασική οργάνωση είναι οι «θεματικές ενότητες»
- Θα υπάρχει μία για κάθε συνάντηση



The screenshot shows the E-class interface for the course "ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι" by Κωνσταντίνος Σκορδούλης. The interface includes a sidebar with navigation options like "Αναζητήση...", "Επιλογές Μαθήματος", "Ανακοινώσεις", "Εργασίες", "Μηνύματα", "Ομάδες Χρηστών", and "Σύνδεσμοι". The main content area features a video titled "Περιγραφή" showing a child working on a circuit board. Below the video, there is a section for "Ενότητες" (Units) with a red circle highlighting the first unit, "Πρώτη Συνάντηση 2022-2023: Εισαγωγή", and a blue starburst graphic with the word "κλικ" (click) pointing to it. A large blue arrow points from the text on the left towards this unit. To the right, there is a "Ημερολόγιο" (Calendar) for November 2022.

Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών  
ΙΔΡΥΘΕΝ ΤΟ 1837

Αναζητήση...

Επιλογές Μαθήματος

Ανακοινώσεις

Εργασίες

Μηνύματα

Ομάδες Χρηστών

Σύνδεσμοι

Χαρτοφυλάκιο / ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι

ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι  
Κωνσταντίνος Σκορδούλης

Περιγραφή

- Το μάθημα δεν διαθέτει περιγραφή -

Ενότητες ?

Πρώτη Συνάντηση 2022-2023: Εισαγωγή


Δεύτερη Συνάντηση 2022-2023: Project Φανάρια Κυκλοφορίας

Ημερολόγιο

Νοέμβριος 2022

Κυριακή	Δευτέρα	Τρίτη	Τετάρτη	Πέμπτη	Παρασκευή	Σάββατο
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26

# E-class

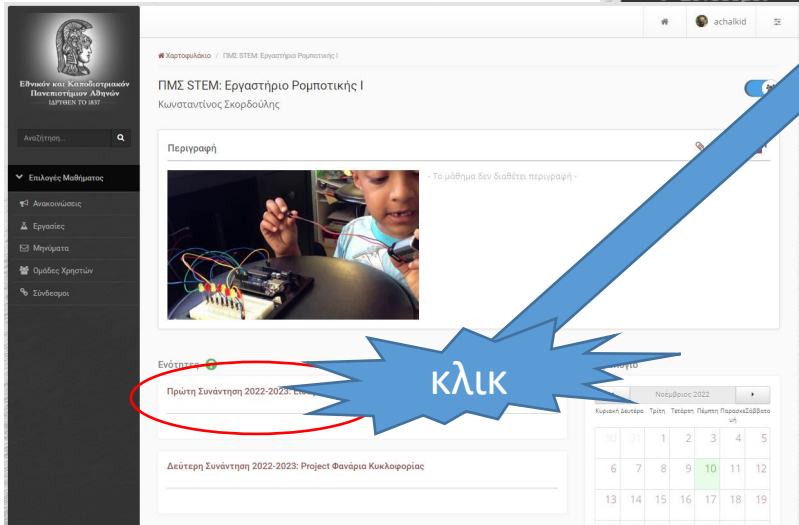


Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών  
ΙΔΡΥΘΕΝ ΤΟ 1837

Αναζήτηση...

Επιλογές Μαθήματος

- Ανακοινώσεις
- Εργασίες
- Μηνύματα
- Ομάδες Χρηστών
- Σύνδεσμοι



Εθνικόν και Καποδιστριακόν  
Πανεπιστήμιον Αθηνών  
ΙΔΡΥΘΕΝ ΤΟ 1837

Αναζήτηση...

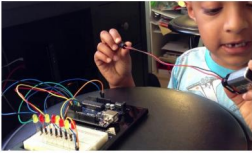
Επιλογές Μαθήματος

- Ανακοινώσεις
- Εργασίες
- Μηνύματα
- Ομάδες Χρηστών
- Σύνδεσμοι

Χαρτοφυλάκιο / ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι

ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι  
Κωνσταντίνος Σκορδούλης

Περιγραφή



Ενότητες

Πρώτη Συνάντηση 2022-2023: Εισαγωγή

Δεύτερη Συνάντηση 2022-2023: Project Φανάρια Κυκλοφορίας

ΚΛΙΚ

Χαρτοφυλάκιο / ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι / Πρώτη Συνάντηση 2022-2023: Εισαγωγή

## ΠΜΣ STEM: Εργαστήριο Ρομποτικής Ι

### Ενότητες

Δεύτερη Συνάντηση 2022-2023: P... →

#### Πρώτη Συνάντηση 2022-2023: Εισαγωγή

Σε κάθε συνάντηση θα βρίσκετε ότι αρχεία χρειάζονται και κάποιες σύντομες οδηγίες, επεξηγήσεις κλπ.

Στην πρώτη συνάντηση, ουσιαστικά ερχόμαστε αντιμέτωποι με πολλά πράγματα, τον μικροελεγκτή arduino uno, την πλακέτα δοκιμών/breadboard, λίγα ηλεκτρονικά εξαρτήματα, το περιβάλλον προγραμματισμού IDE και την γλώσσα προγραμματισμού που υποστηρίζει τις διαδικασίες που πρέπει να τηρούνται και γενικά ένα τρόπο σκέψης που ίσως δεν ήταν γνωστός σε όλες και όλους σας.

Όμως ερχόμαστε σε επαφή με όλα αυτά με ένα ελεγχόμενο τρόπο και με σταδιακή εμφάνιση νέων στοιχείων, χτίζοντας κάθε φορά στις προηγούμενες εμπειρίες.

Σε κάθε συνάντησή μας θα δείτε πως θα αποκτήσετε περισσ'οτερη εμπιστοσυνη στις δυνάμεις σας και σιγά-σιγά θα χρειάζεστεόλο και λιγότερη υποστήριξη.

ΚΑΛΗ ΑΡΧΗ!!!

Ακολουθεί το υλικό που είδαμε στην πρώτη συνάντηση (η παρουσίαση και το φύλλο εργασίας)

- Εισαγωγική Παρουσίαση Πρώτης Συνάντησης
- 2022-2023 Arduino Φύλλο Εργασίας 1.pdf

Ακολουθεί ένα κόμικ για το arduino, που ίσως σας διευκολύνει να καταλάβετε καποια σημεία. Μοιάζει εισαγωγικό αλλά σίγουρα ξεπερνάει την πρώτη επαφή που ειχαμε στην πρώτη συνάντηση. Δίνονται δυο εκδοχές και η πρωτότυπη και η εξελληνισμένη.

- arduino\_comic\_v0004\_GREEK.pdf
- arduino\_comic\_v0004.pdf

Ενότητες

Πρώτη Συνάντηση 2022-2023: Εισαγω

# Σήμερα στην συνάντησή μας

- **Επανάληψη, εμπέδωση**
  - Breadboard, βασικά Arduino pins
  - Διαδικασία «ανεβάσματος» του προγράμματος στο Arduino
  - Βασικό σχήμα: Κύκλωμα και Πρόγραμμα ΣΥΝΕΡΓΑΖΟΝΤΑΙ
  - Φανάρι κυκλοφορίας (βασική εκδοχή)
  - Έλεγχος με διακόπτη (βασική ιδέα)
- **ΒΑΣΙΚΕΣ ΙΔΕΕΣ**
  - Είσοδος/έξοδος, αναλογική/ψηφιακή πληροφορία
  - Αξιοποίηση της «**λογικής**» τους σε **κύκλωμα** και **πρόγραμμα**
- **Νέο υλικό:** μια μεταβλητή αντίσταση (ποτενσιόμετρο), μια φωτοαντίσταση
- **Πως πήγε ???** Νέα Διαδικασία: Πώς θα φτιάξω (θα σχεδιάσω) παιδαγωγικό υλικό? (π.χ. ένα σύντομο φύλλο εργασίας)
  - Αναστοχασμός στον τρόπο δουλειάς στο Εργαστήριό μας



# Ρομποτική Κατασκευή / Physical Computing

Κατασκευή (Ηλεκτρο-Μηχανική)



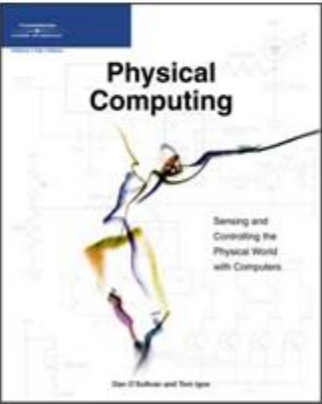
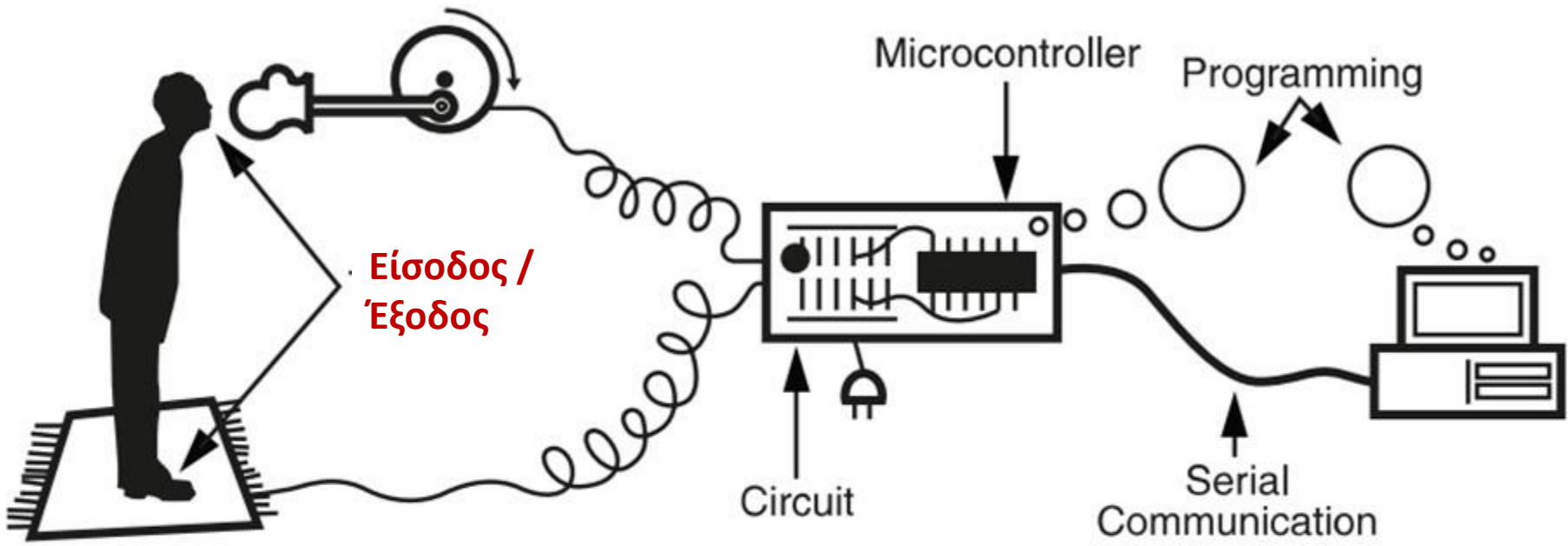
Ηλεκτρονικό κύκλωμα & Αισθητήρες

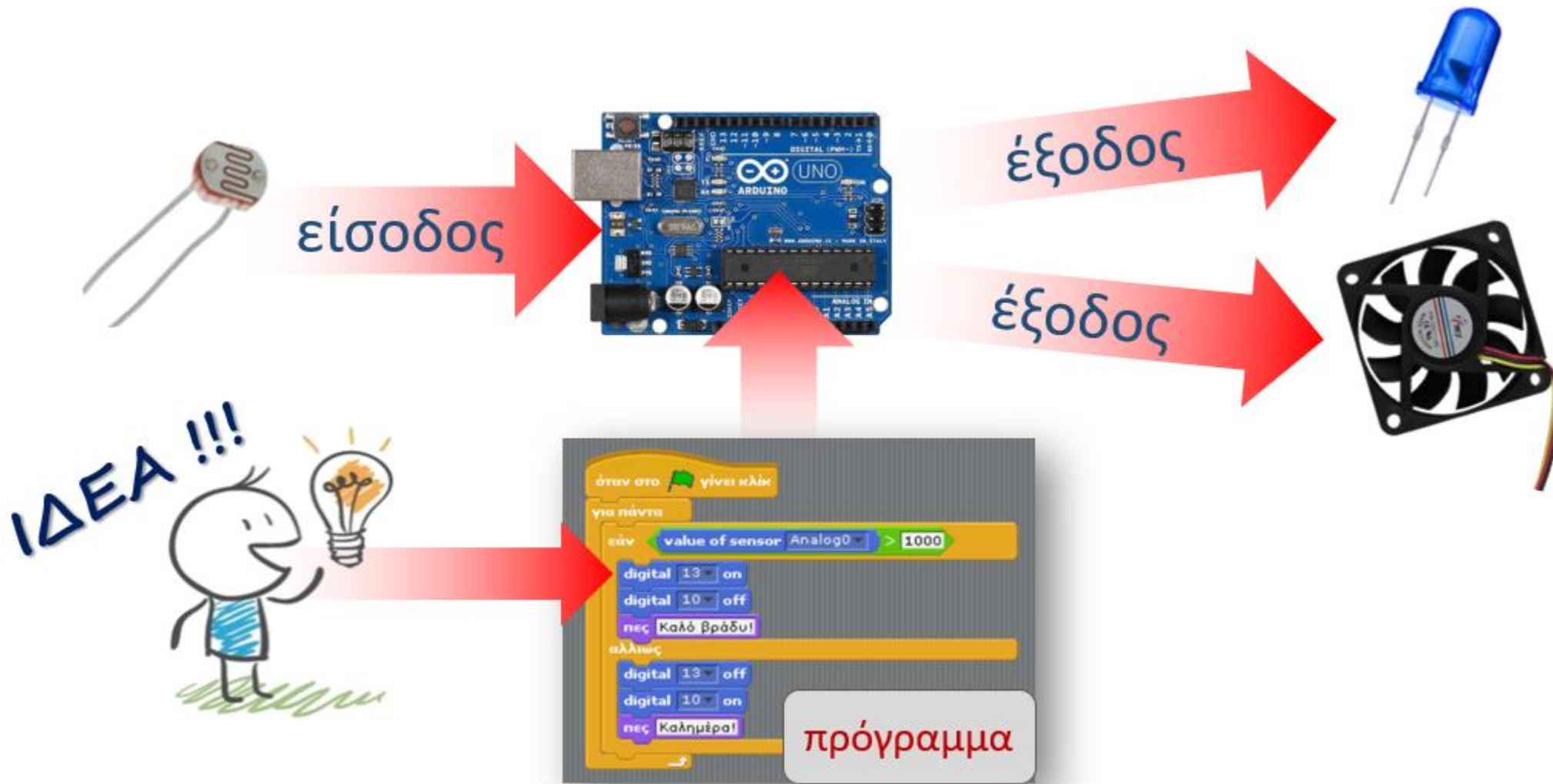


Προγραμματισμός

Σήμερα εστιάζουμε στα δυο από τα τρία μέρη

**Figure 1.4**  
The parts of a physical computing system.







## Ρομποτική Κατασκευή / Physical Computing

Κατασκευή (Ηλεκτρο-Μηχανική)




Ηλεκτρονικό κύκλωμα & Αισθητήρες



**Προγραμματισμός**

Σήμερα εστιάζουμε στα δυο από τα τρία μέρη

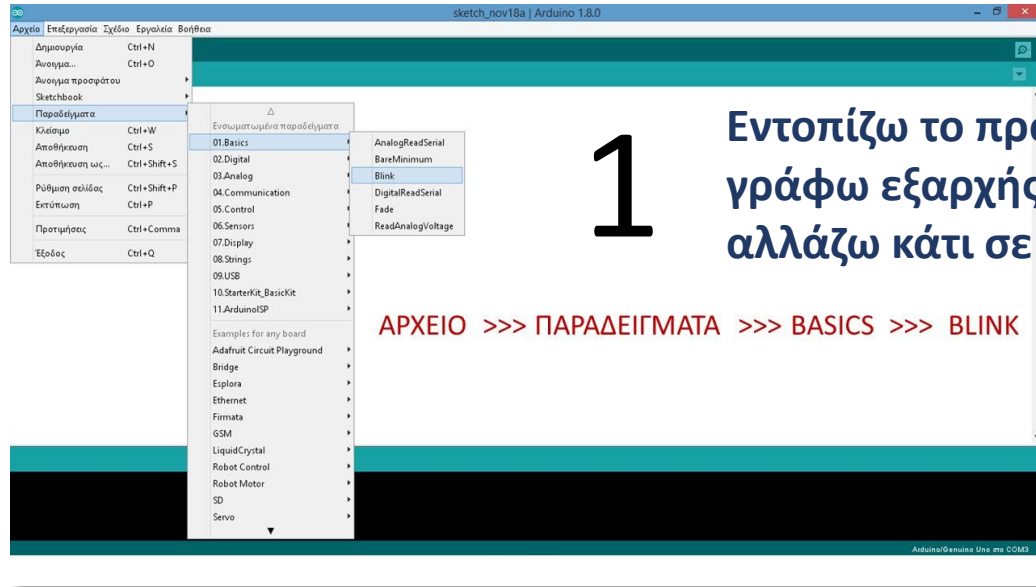
# IDE / κώδικας



ΤΙ ΕΙΝΑΙ ΤΟ  
ΟΛΟΚΛΗΡΩΜΕΝΟ  
ΠΕΡΙΒΑΛΛΟΝ  
ΑΝΑΠΤΥΞΗΣ  
(INTEGRATED DEVELOPMENT ENVIRONMENT / IDE);

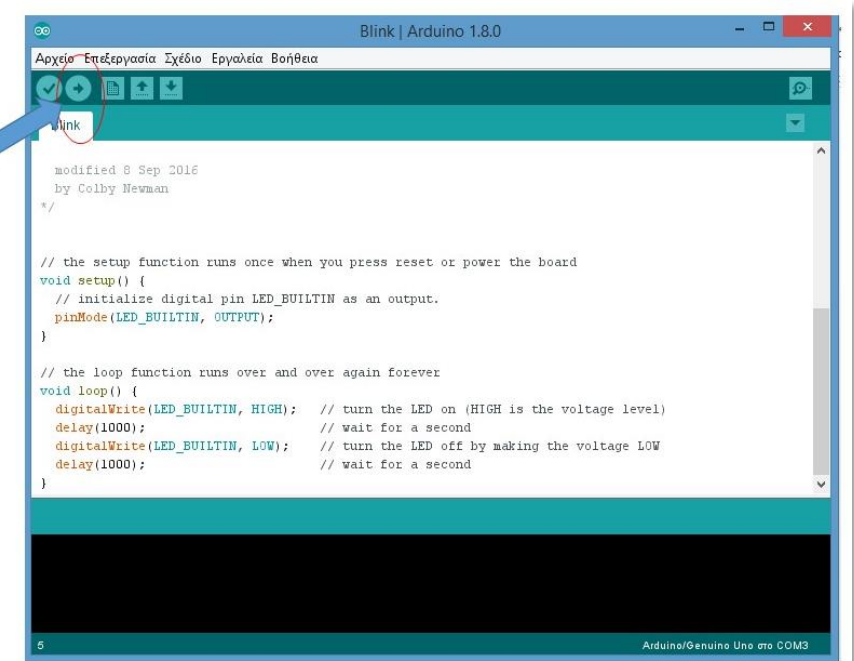
ΟΤΑΝ ΚΑΤΕΒΑΖΕΙΣ ΤΟ ΛΟΓΙΣΜΙΚΟ ΤΟΥ ARDUINO ΚΑΤΕΒΑΖΕΙΣ ΕΝΑ IDE. ΑΥΤΟ ΣΥΝΔΥΑΖΕΙ ΕΝΑ ΕΠΕΞΕΡΓΑΣΤΗ ΠΗΓΑΙΟΥ ΚΩΔΙΚΑ (TEXT EDITOR), ΕΝΑΝ ΜΕΤΑΓΛΩΤΤΙΣΤΗ (COMPILER) ΚΑΙ ΆΛΛΑ ΕΡΓΑΛΕΙΑ ΠΟΥ ΒΟΗΘΑΝΕ ΤΟΥΣ ΠΡΟΓΡΑΜΜΑΤΙΣΤΕΣ ΣΤΗΝ ΑΝΑΠΤΥΞΗ ΠΡΟΓΡΑΜΜΑΤΩΝ.

# Διαδικασία Προγραμματισμού (για κάθε αλλαγή που θέλω να κάνω)



2

Το ΑΝΕΒΑΖΩ στην πλακέτα του Arduino



3

Παρατηρώ την επίδρασή του, την λειτουργία του, στο κύκλωμα, με το οποίο συνεργάζεται

# Προγραμματισμός με εικονίδια

Ardublock



Scratch for Arduino



M-block



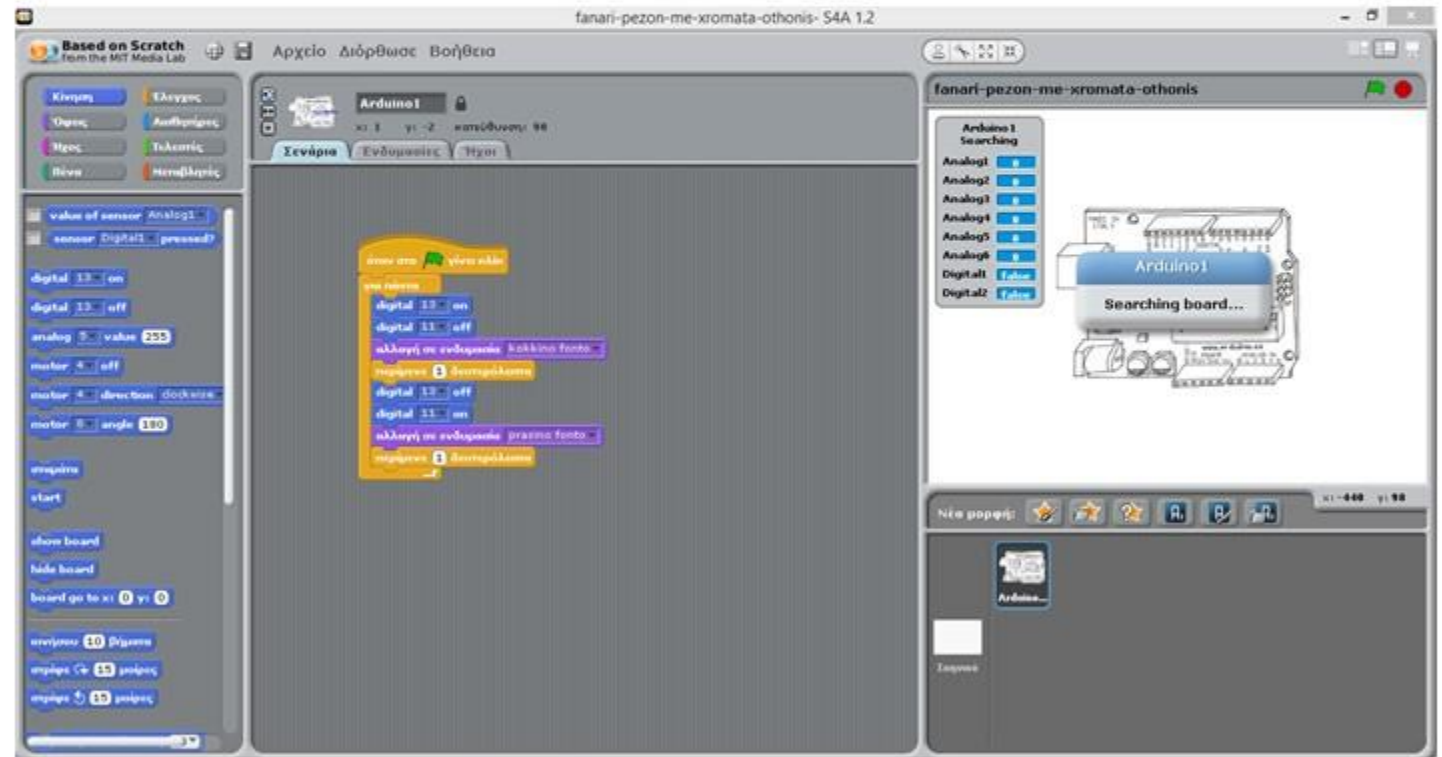
PictoBlox



Από την επόμενη εβδομάδα



Scratch for Arduino

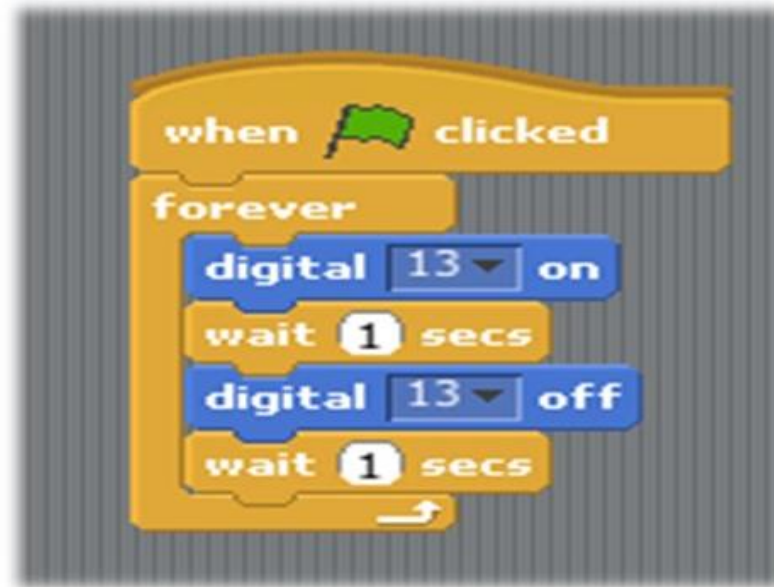




Σε αυτό το γενικό πλαίσιο, η χρήση του S4A ως εργαλείο οπτικού προγραμματισμού, μεταξύ των άλλων πλεονεκτημάτων του, απαλλάσσει από τον φόρτο του ελέγχου και της αποκατάστασης των συντακτικών λαθών καθώς και από την ανάγκη για μια σειρά δηλώσεων.



Scratch for Arduino



```
void setup() {  
  pinMode(13, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  delay(1000);  
  digitalWrite(13, LOW);  
  delay(1000);  
}
```

### Πηγαίος κώδικας (χωρίς σχόλια)

```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  digitalWrite(12, LOW);  
  delay(1000);  
  
  digitalWrite(13, LOW);  
  digitalWrite(12, HIGH);  
  delay(1000);  
}
```

### Βελτίωση στον κώδικα (καλύτερη προγραμματιστική τεχνική)

```
int redledpin = 13;           // Δηλώνω δύο ΜΕΤΑΒΛΗΤΕΣ ακέραιου τύπου  
int greenledpin = 12;       // και τους δίνω αρχικές τιμές (Με το = δίνω τιμή)  
                               // Έτσι αν αλλάξω pins στο κύκλωμα αλλάζω το πρόγραμμα  
                               // μόνο σε ένα σημείο  
  
void setup() {  
  pinMode(redledpin, OUTPUT);  
  pinMode(greenledpin, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(redledpin, HIGH);  
  digitalWrite(greenledpin, LOW);  
  delay(1000);  
  
  digitalWrite(redledpin, LOW);  
  digitalWrite(greenledpin, HIGH);  
  delay(1000);  
}  
  
// Εντολή εκχώρησης τιμής: Με το = δίνω τιμή, δεν είναι το = των μαθηματικών
```

# Ρομποτική Κατασκευή / Physical Computing

Κατασκευή (Ηλεκτρο-Μηχανική)



**Ηλεκτρονικό κύκλωμα & Αισθητήρες**



Προγραμματισμός

Σήμερα εστιάζουμε στα δυο από τα τρία μέρη

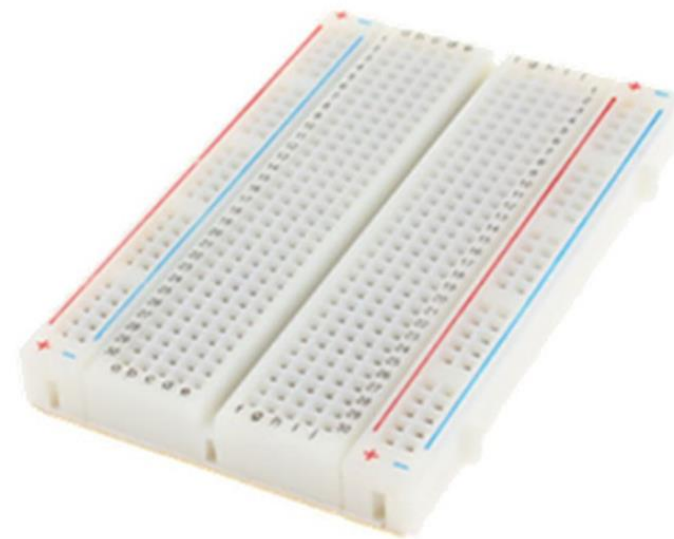
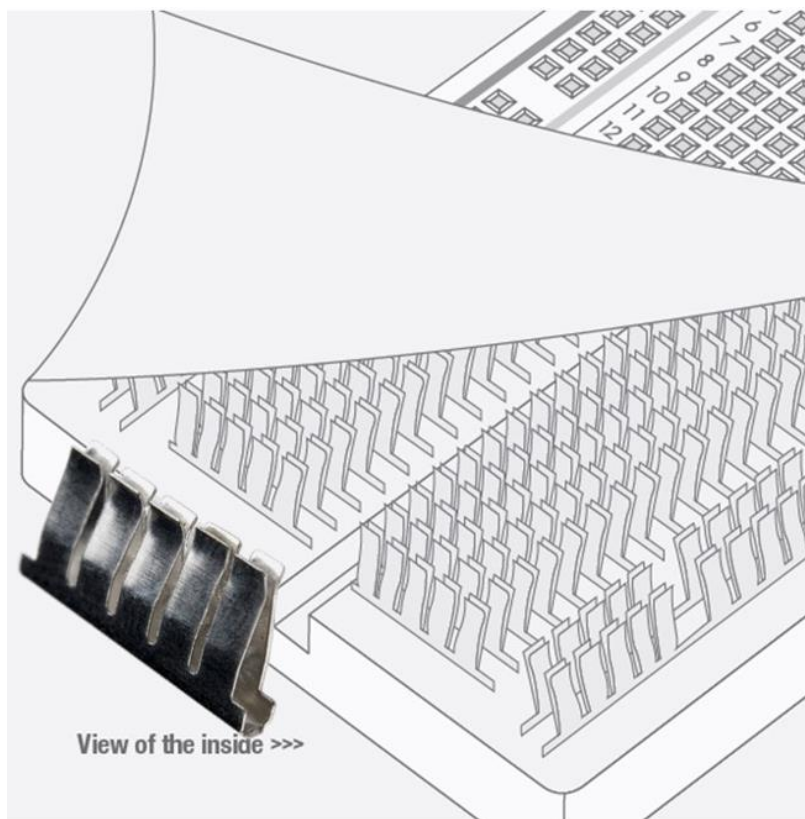
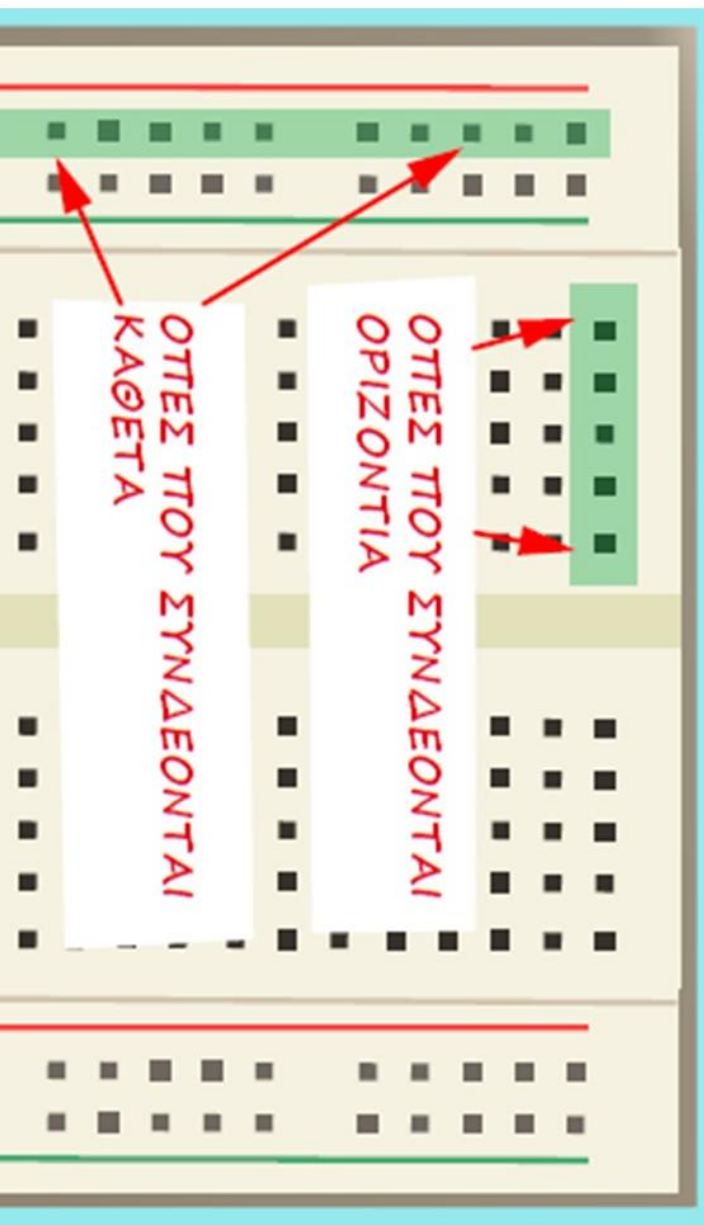
# Οι υποδοχές (τα pin), του Arduino που μας ενδιαφέρουν

- Αναλογικές και Ψηφιακές Είσοδοι και έξοδοι
- 5 Volt και Γείωση (GND)





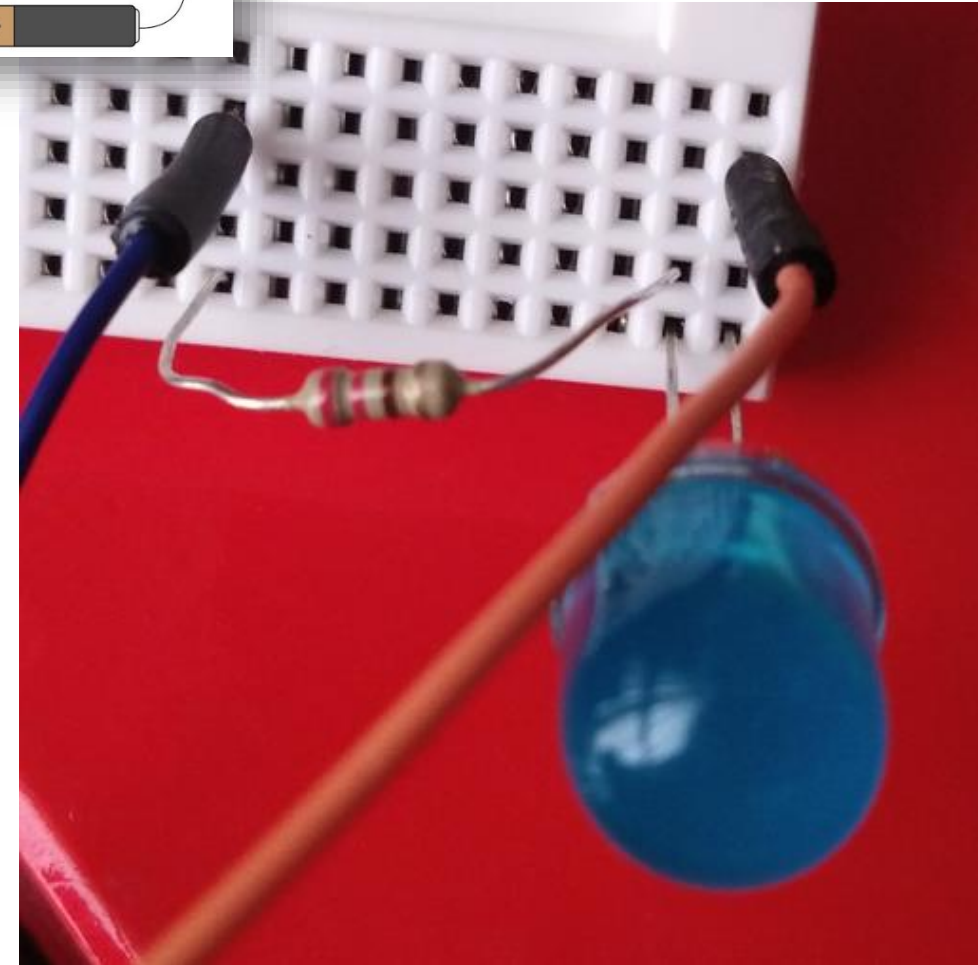
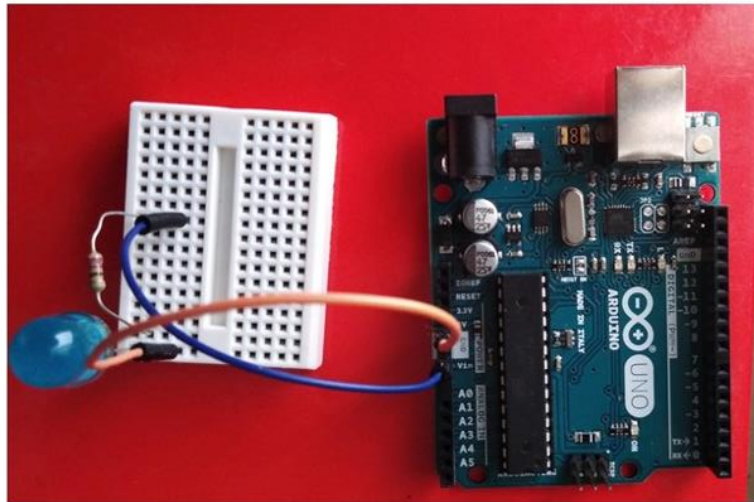
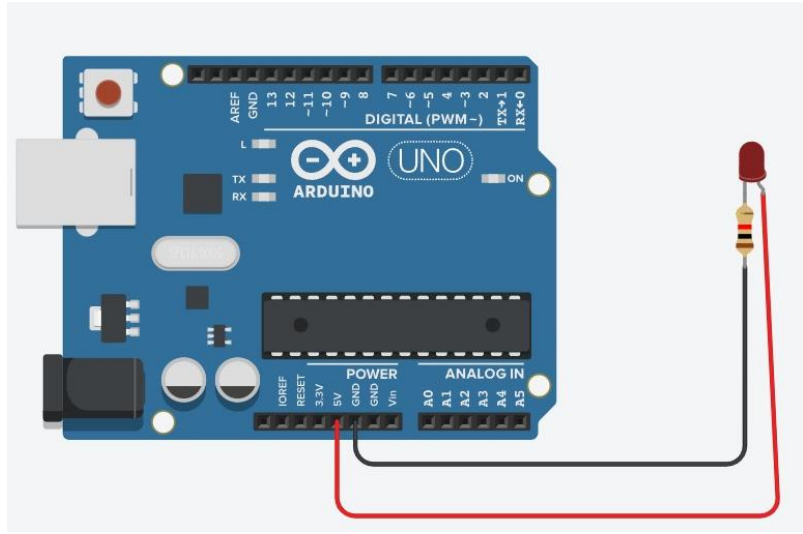
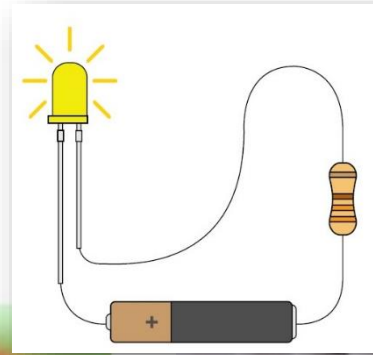
# Breadboard



Κάθε πεντάδα από υποδοχές (τρυπίτσες) είναι συνδεμένη μεταξύ της και δημιουργεί ένωση ανάμεσα στα «ποδαράκια» των ηλεκτρονικών εξαρτημάτων.



# Ένα απλό ηλεκτρικό κύκλωμα



# Η πιο απλή κατασκευή

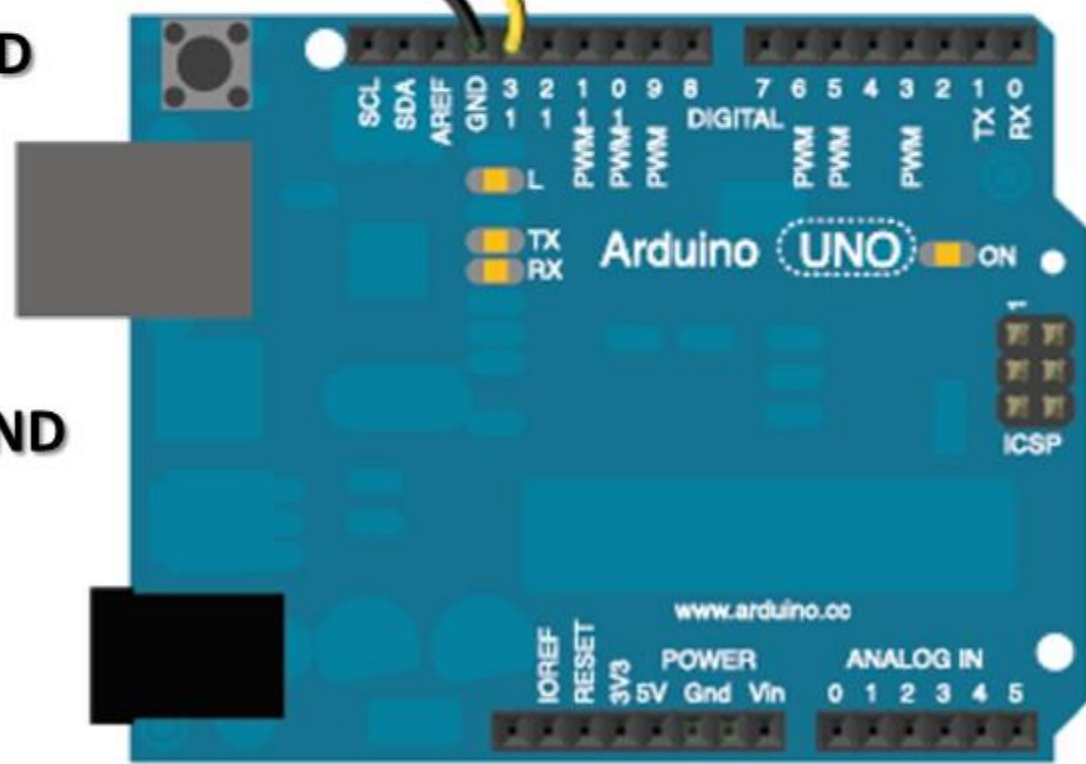
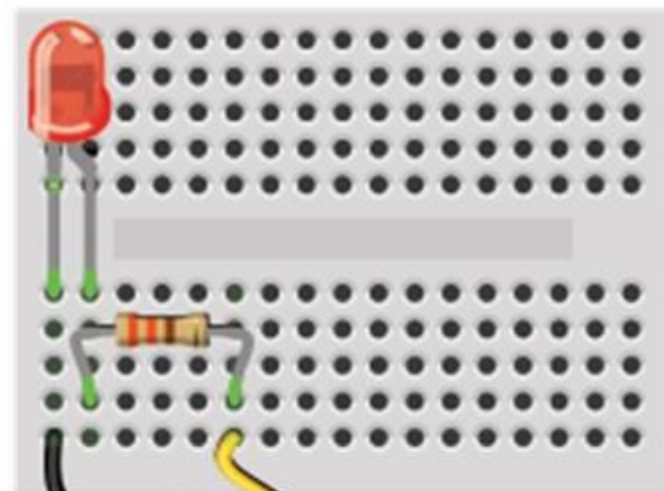
Θέλουμε να αναβοσβήνει ένα led

13 ----- + Led - ----- αντίσταση ----- Γείωση/GND

ή

13 ----- αντίσταση ----- + Led - ----- Γείωση/GND

## ΚΥΚΛΩΜΑ





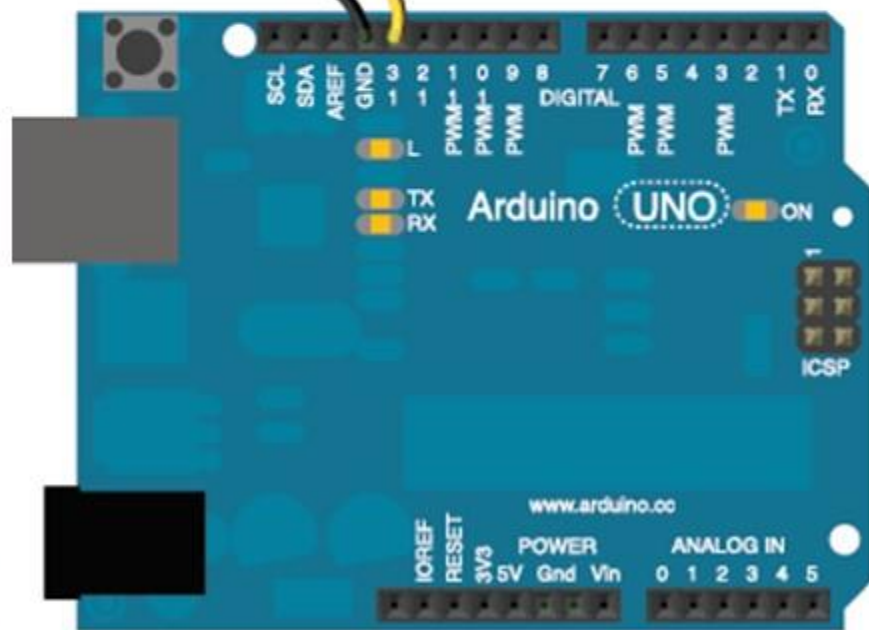
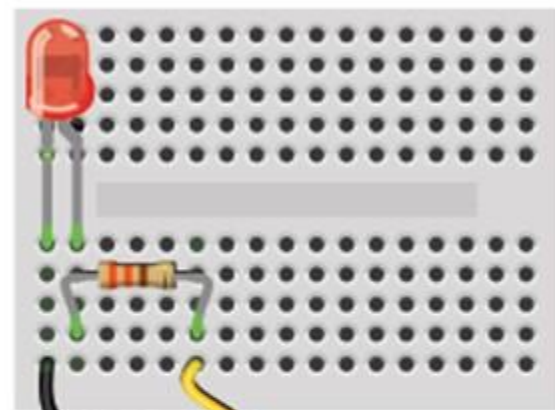
# Κώδικας και κύκλωμα συνεργάζονται

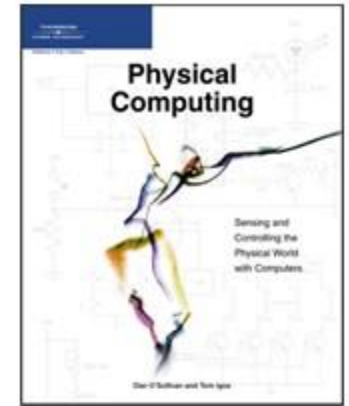
```
void setup() { //δηλώνει την αρχή του
                συγκεκριμένου τμήματος του κώδικα
  pinMode(13, OUTPUT); //ορίζει το pin 13 ως έξοδο
} //τέλος τμήματος κώδικα

void loop() { //δηλώνει την αρχή του
              συγκεκριμένου τμήματος του κώδικα
  digitalWrite(13, HIGH); //δίνει στο pin 13 την τιμή high
  delay(1000); //χρονοκαθυστέρηση ενός
              δευτερόλεπτου

  digitalWrite(13, LOW); //δίνει στο pin 13 την τιμή low
  delay(1000); //χρονοκαθυστέρηση ενός
              δευτερόλεπτου

} //τέλος τμήματος κώδικα
```

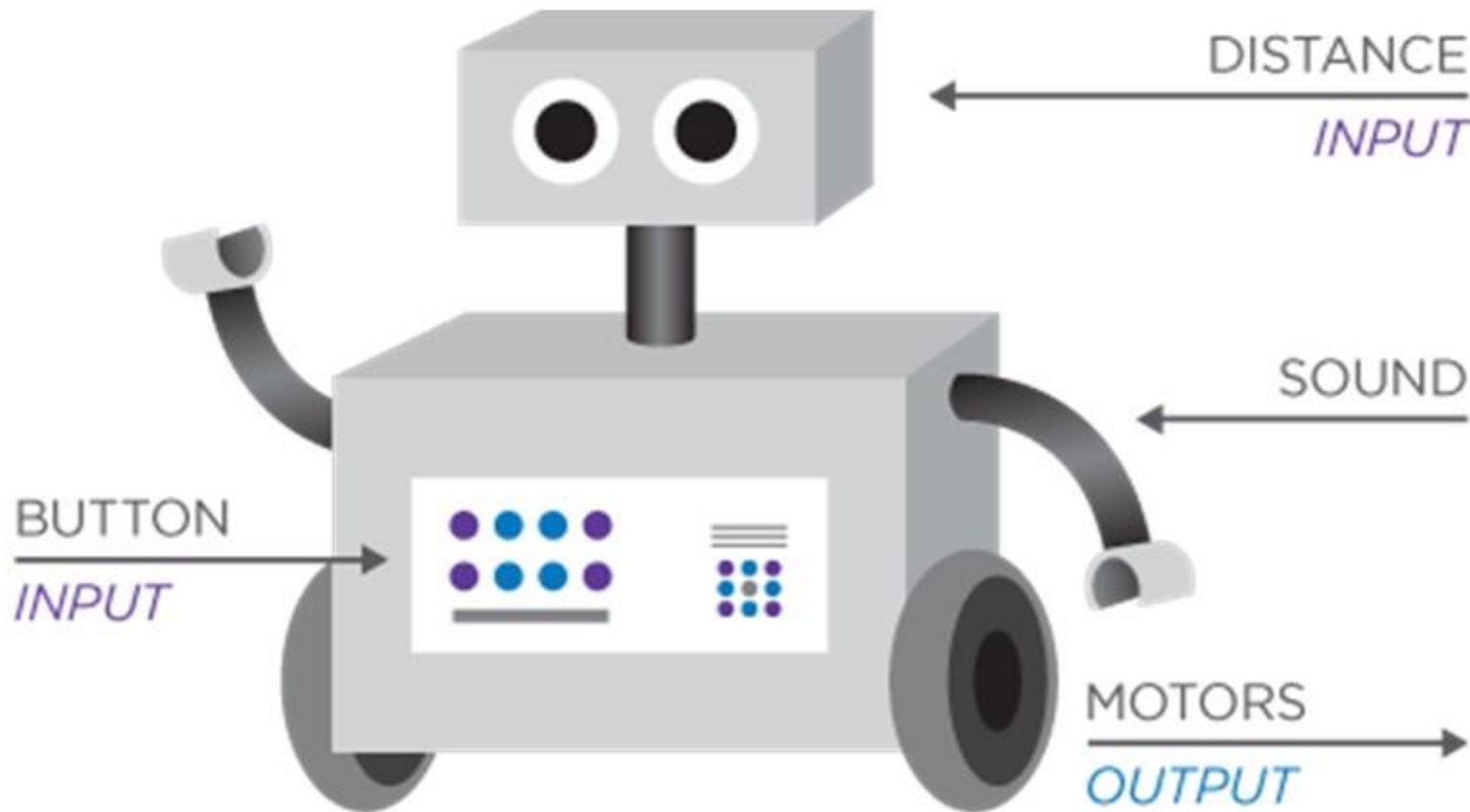
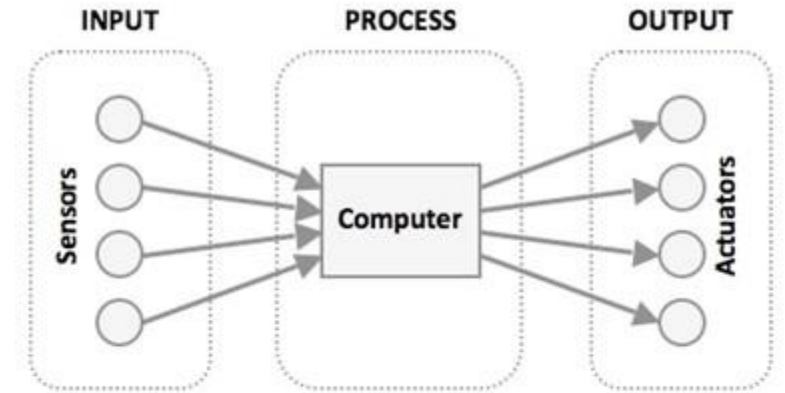




**Figure 1.2**  
Categorize your physical computing challenges.

DIGITAL INPUT	ANALOG INPUT	PROCESSING	DIGITAL OUTPUT	ANALOG OUTPUT

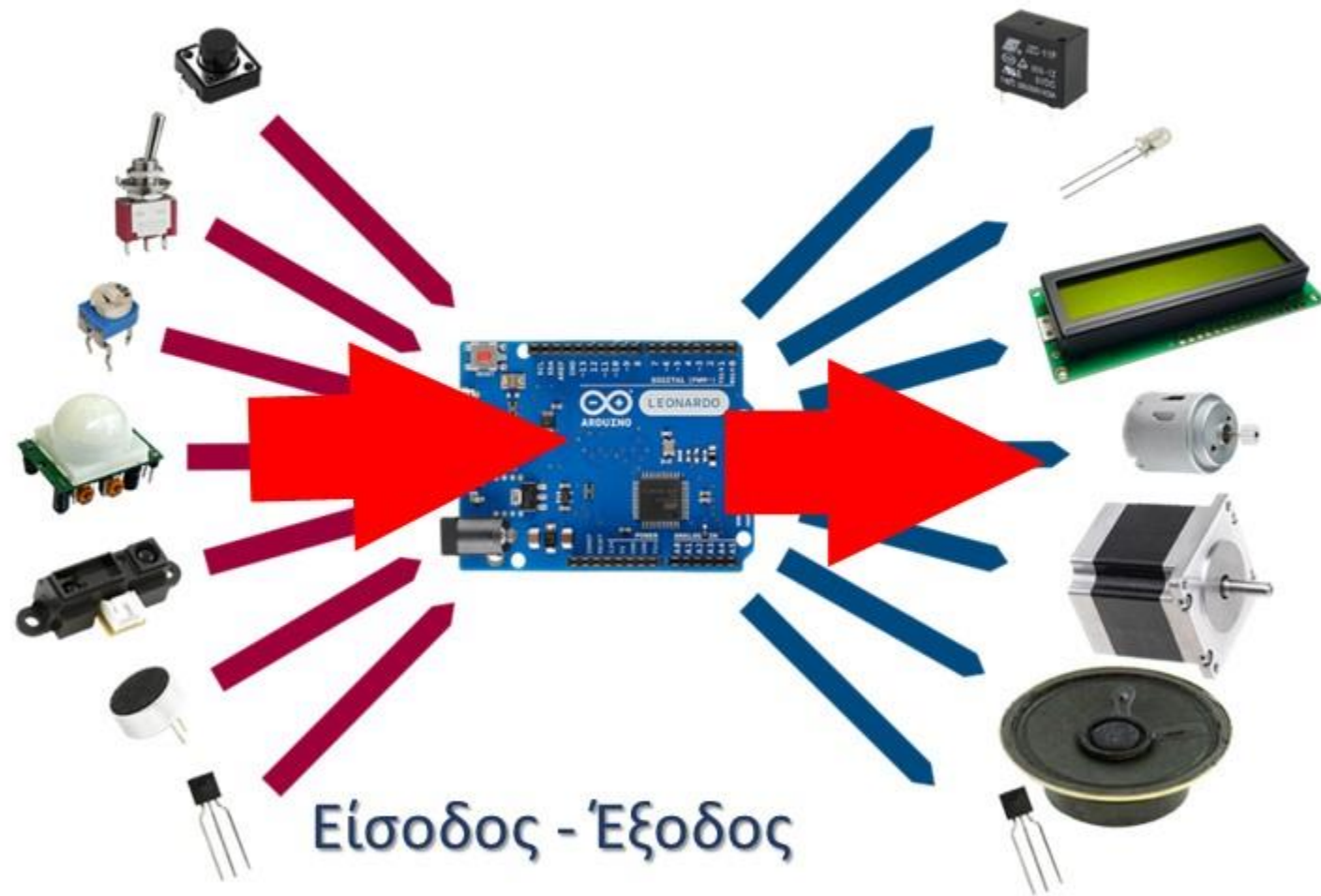
# Είσοδος – Έξοδος (Input / Output)



? !!! ???







Είσοδος - Έξοδος

# Ψηφιακό – Αναλογικό (Digital / Analog)





ΤΠΟΙΑ ΕΙΝΑΙ Η ΔΙΑΦΟΡΑ  
ΑΝΑΜΕΣΑ ΣΤΙΣ  
ΑΝΑΛΟΓΙΚΕΣ ΚΑΙ ΨΗΦΙΑΚΕΣ  
ΕΙΣΟΔΟΥΣ ΚΑΙ ΕΞΟΔΟΥΣ;

ΟΙ ΕΙΣΟΔΟΙ ΚΑΙ ΕΞΟΔΟΙ ΜΤΟΡΟΥΝ ΝΑ ΕΙΝΑΙ ΕΙΤΕ ΨΗΦΙΑΚΕΣ  
ΕΙΤΕ ΑΝΑΛΟΓΙΚΕΣ. Η ΨΗΦΙΑΚΗ ΠΛΗΡΟΦΟΡΙΑ ΕΙΝΑΙ ΔΥΑΔΙΚΗ  
ΕΙΝΑΙ ΔΗΛΑΔΗ ΑΛΗΘΗΣ Ή ΨΕΥΔΗΣ. Η ΑΝΑΛΟΓΙΚΗ ΕΙΝΑΙ  
ΣΥΝΕΧΗΣ, ΜΤΟΡΕΙ ΔΗΛΑΔΗ ΝΑ ΠΕΡΙΕΧΕΙ ΕΝΑ ΣΥΝΟΛΟ  
ΤΙΜΩΝ.

Η ΨΗΦΙΑΚΗ ΠΛΗΡΟΦΟΡΙΑ ΕΙΝΑΙ ΔΙΑΚΡΙΤΗ ΚΑΙ ΠΤΕΤΕΡΑΣΜΕΝΗ. ΠΕΡΙΓΡΑΦΕΤΑΙ ΜΕ ΔΥΟ ΤΡΟΤΠΟΥΣ, 1 ΚΑΙ 0 Η ΕΝΕΡΓΟΤΠΟΙΗΜΕΝΗ ΚΑΙ ΑΤΤΕΝΕΡΓΟΤΠΟΙΗΜΕΝΗ (ON - OFF).

Η ΑΝΑΛΟΓΙΚΗ ΠΛΗΡΟΦΟΡΙΑ ΧΑΡΑΚΤΗΡΙΖΕΤΑΙ ΑΤΠΟ ΜΙΑ ΣΥΝΕΧΟΜΕΝΗ ΡΟΗ. ΜΤΟΡΕΙ ΝΑ ΛΑΒΕΙ ΑΤΤΕΡΙΟΡΙΣΤΟ ΑΡΙΘΜΟ ΤΙΜΩΝ.

ΕΝΑΣ ΔΙΑΚΟΤΠΤΗΣ ΕΙΝΑΙ ΜΙΑ ΨΗΦΙΑΚΗ ΕΙΣΟΔΟΣ ΕΝΩ ΕΝΑΣ ΣΕΝΣΟΡΑΣ ΜΙΑ ΑΝΑΛΟΓΙΚΗ. ΤΟ ΕΥΡΟΣ ΤΟΥ ΑΝΑΛΟΓΙΚΟΥ ΣΕΝΣΟΡΑ ΠΕΡΙΟΡΙΖΕΤΑΙ ΑΤΠΟ ΤΗΝ ΜΕΤΑΤΡΟΠΗ ΤΩΝ ΔΕΔΟΜΕΝΩΝ ΤΠΟΥ ΠΑΡΑΓΕΙ ΣΕ ΨΗΦΙΑΚΑ.

Είσοδος	Έξοδος	Σχόλιο / Παράδειγμα ...
-	-	Ασχολίαστο
-	Digital	Blink, ...
-	Analog	
Digital	-	Χωρίς νόημα(*)
Digital	Digital	
Digital	Analog	
Analog	-	Χωρίς νόημα(*)
Analog	Digital	
Analog	Analog	

(\*) Το ρομπότ είναι εργάτης. Πρέπει να κάνει κάτι



Είσοδος	Έξοδος	Παράδειγμα (ενδεικτικό / προσωρινό)
-	Digital	Λειτουργία ενός «Φάρου» (Blink)
-	Analog	?
Digital	Digital	?
Digital	Analog	?
Analog	Digital	?
Analog	Analog	?



Είσοδος	Έξοδος	Σχόλιο / Παράδειγμα (ενδεικτικό / προσωρινό)
-	Digital	Blink, φανάρι τροχαίας, SOS, ...
-	Analog	Led με μεταβαλλόμενη φωτεινότητα
Digital	Digital	Έλεγχος led με διακόπτη, ...
Digital	Analog	Έλεγχος με διακόπτη της λειτουργίας (πότε θα ξεκινάει) του led με μεταβαλλόμενη φωτεινότητα
Analog	Digital	Έλεγχος led (άναψε/σβήσε) με φωτοαντίσταση
Analog	Analog	Έλεγχος φωτεινότητας led με ποτενσιόμετρο (μεταβλητή αντίσταση)

# Το φύλλο εργασίας

## Περιγραφή, εξηγήσεις

Στις ακόλουθες περιπτώσεις, συμπληρώστε το πλήθος (0, 1, 2, 3, ...) και το είδος (Analog/Digital / - ) των κυκλωμάτων εισόδου και εξόδου σε έναν «αυτοματισμό».

Παρατηρήστε πως ενώ μπορεί να μην υπάρχει «είσοδος», δεν έχει νόημα να μην υπάρχει «έξοδος».

Περιγραφή περίπτωσης	Είσοδος		Έξοδος	
	Πλήθος (-, 1, 2, ...)	Είδος (- /A / D)	Πλήθος (1, 2, ...)	Είδος (A / D)
Ένα κόκκινο led ανάβει και σβήνει κάθε ένα δευτερόλεπτο	-	-	1	D
Ένα κόκκινο led και ένα πράσινο led, ανάβουν και σβήνουν εναλλάξ, κάθε ένα δευτερόλεπτο				
Δυο κίτρινα led συνδεδεμένα σε σειρά, ανάβουν και σβήνουν (προφανώς μαζί) κάθε ένα δευτερόλεπτο				
Με τον κατάλληλο προγραμματισμό ελέγχω την λειτουργία ενός φωτεινού σηματοδότη (οχημάτων και πεζών), σε μια διασταύρωση				
Όσο πατάω έναν διακόπτη πίεσης (pushbutton), ανάβει ένα led, και όταν δεν τον πατάω το led παραμένει σβηστό				
Ένα led ανάβει αυξάνοντας σταδιακά την φωτεινότητα του και μετά την μειώνει σταδιακά. Αυτό επαναλαμβάνεται συνέχεια.				
Όταν με έναν αισθητήρα φωτός, θεωρώ πως είναι μέρα, σβήνουν τα φώτα (ένα led), και όταν θεωρώ πως είναι νύχτα (είναι δηλ. αρκετά σκοτεινά) τότε ανάβουν τα φώτα (ένα led)				
Με έναν αισθητήρα φωτός, ελέγχω το επίπεδο φωτισμού του χώρου και ανάλογα ανάβουν με μεταβαλλόμενη ένταση φωτισμού τα φώτα (ένα led)				
Με έναν αισθητήρα θερμοκρασίας μετρώ την θερμοκρασία του χώρου και αν είναι πάνω από 20°C αρχίζει να λειτουργεί ένας ανεμιστήρας (αλλιώς σβήνει)				
Με έναν αισθητήρα θερμοκρασίας μετρώ την θερμοκρασία του χώρου και αν είναι πάνω από 20°C αρχίζει να λειτουργεί ένας ανεμιστήρας, αν είναι πάνω από 30 °C αρχίζει να λειτουργεί και ένας δεύτερος ανεμιστήρας				
Με ένα ποτενσιόμετρο * (μεταβλητή αντίσταση) ελέγχω πόσο πολύ ή λίγο θα φωτοβολεί ένα led				
Με τρία ποτενσιόμετρα (μεταβλητές αντιστάσεις) ελέγχω πόσο πολύ ή λίγο θα φωτοβολούν τρία led (κόκκινο, πράσινο, μπλε) με σκοπό να φτιάξω πολλές διαφορετικές αποχρώσεις (χρωματικό μοντέλο RGB/Red-Green-Blue)				
Όταν πλησιάζω σε μια απόσταση από την πόρτα του super market, αυτή ανοίγει αυτόματα				
Όταν περάσω μπροστά από έναν ανιχνευτή, λίγο πριν την πόρτα του super market, η πόρτα αυτή ανοίγει αυτόματα				
Με ένα σύστημα που μετρώ την υγρασία στο έδαφος, ανάβει ένα πράσινο led αν είναι ποτισμένο ή ένα κόκκινο led αν χρειάζεται πότισμα				
Όταν πλησιάσω το χέρι μου κοντά σε έναν «έξυπνο» κάδο ανακύκλωσης, ανοίγει το καπάκι του αυτόματα				

παρακάτω περιγράψτε δυο δικές σας ιδέες αυτοματισμών

Εστίαση σε βασικές παραμέτρους :

Είσοδος/έξοδος,

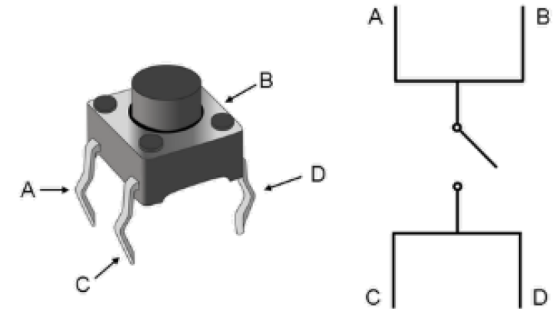
Αναλογικό/Ψηφιακό,

Μηδέν/Ένα/Πολλά

Ερεθίσματα/Αποτελέσματα

# Επανάληψη

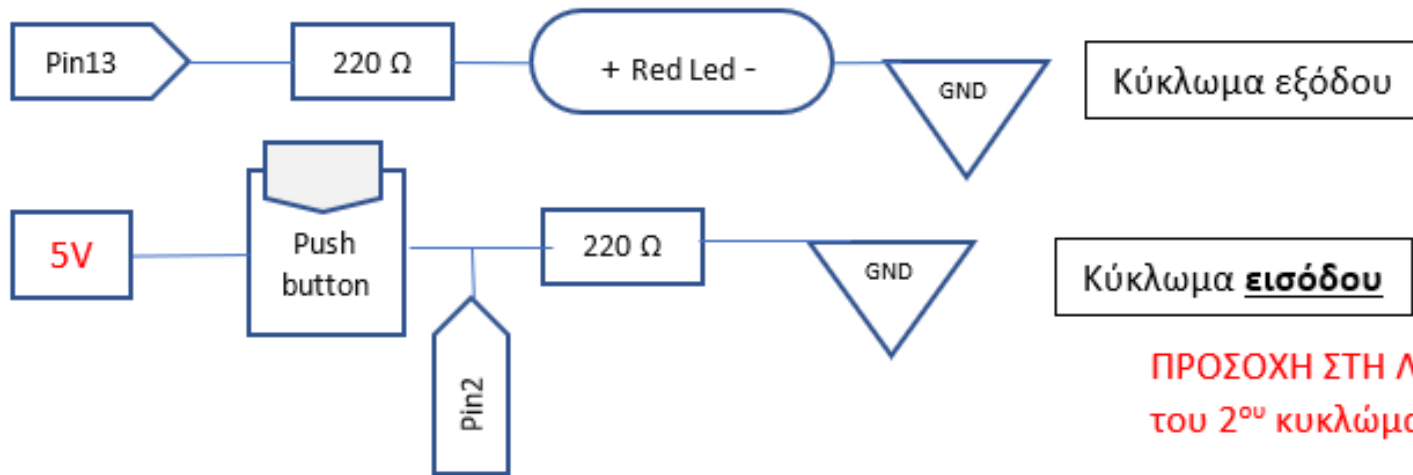
- Φανάρι οχημάτων
- Έλεγχος με διακόπτη pushbutton



Όσο πιο απλό πρόγραμμα θέλετε και καταλαβαίνετε

# Ο διακόπτης push button ως ψηφιακή είσοδος

Πρόχειρο σχεδίασμα κυκλώματος



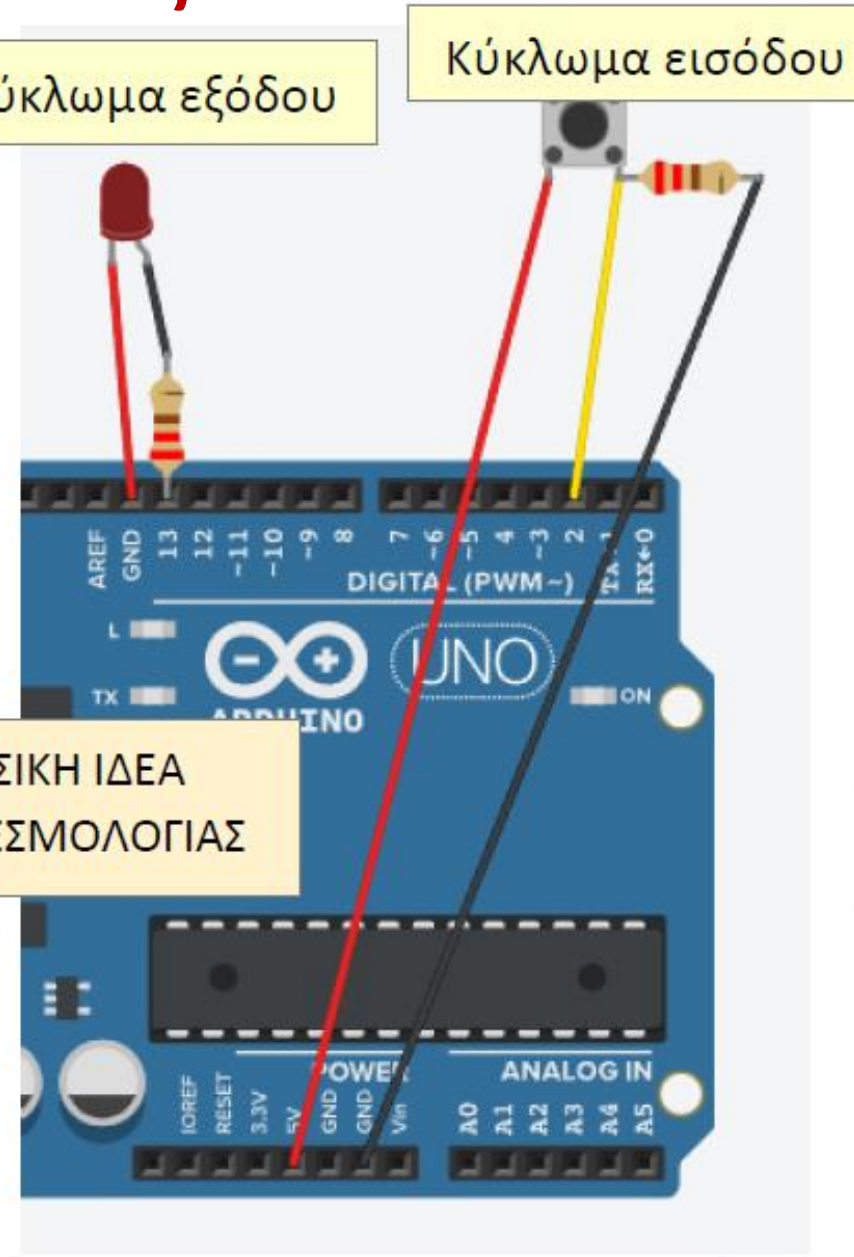
ΠΡΟΣΟΧΗ ΣΤΗ ΛΟΓΙΚΗ του 2<sup>ου</sup> κυκλώματος!!!

DigitalRead()

Κύκλωμα εξόδου

Κύκλωμα εισόδου

ΒΑΣΙΚΗ ΙΔΕΑ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ



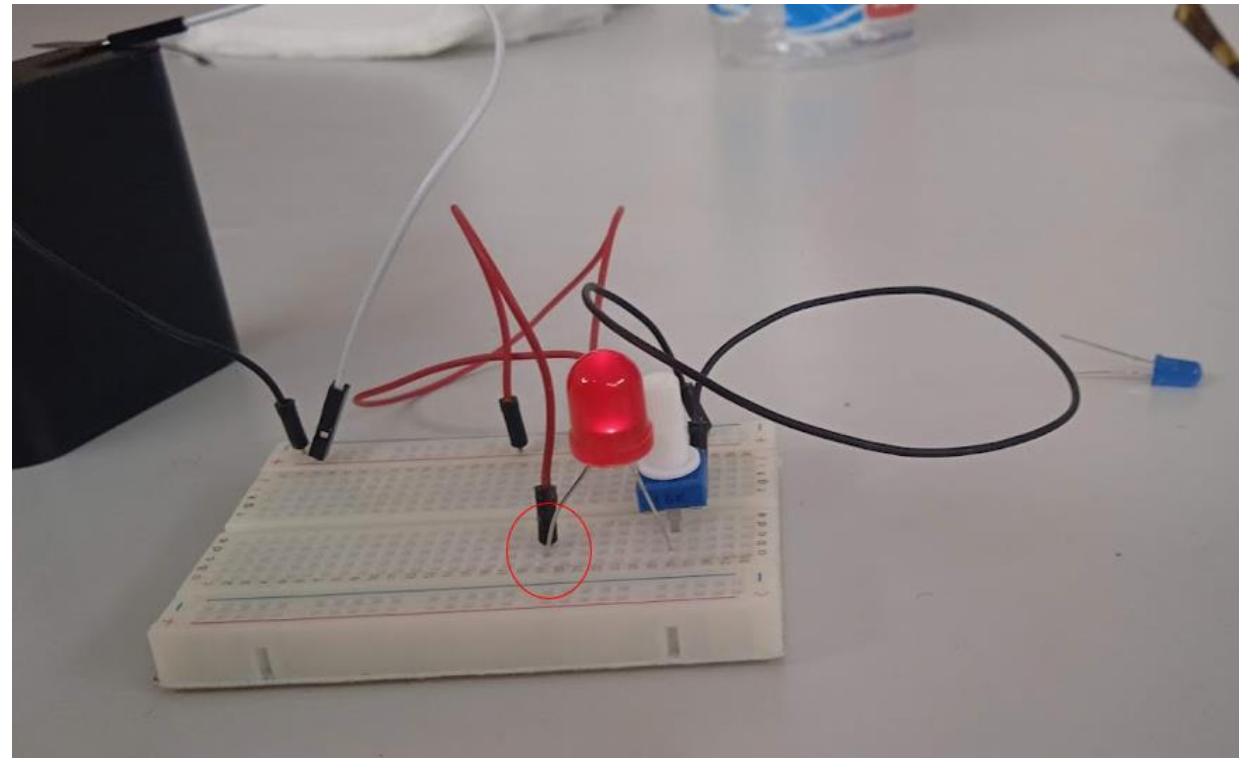


# Επέκταση: Φανάρι κυκλοφορίας οχημάτων

- ΣΧΕΔΙΑΣΤΕ κύκλωμα και πρόγραμμα
- Κατασκευάστε και ελέγξτε τη λειτουργία του «Φαναριού σας»
- Φωτογραφίστε, βιντεοσκοπήστε, αποθηκεύτε τον πηγαίο σας κώδικα, οργανώστε το υλικό σας.
- ~~Καταγράψτε τις δυσκολίες που συναντήσατε και τον τρόπο που τις αντιμετωπίσατε.~~
  - ~~Χαρακτηρίστε τις δυσκολίες ως «τεχνικές» (κύκλωμα κλπ), «λογικές» (σχεδίασης αλγόριθμου), «γλώσσας προγραμματισμού» (συντακτικά λάθη), «συνεργασίας» (σχέσεις στην ομάδα) ή «άλλο» (Περιγράψτε)~~

# Νέα υλικά

- Φωτοαντίσταση
- Μεταβλητή αντίσταση (ποτενσιόμετρο)



Πειραματιστείτε για το πως λειτουργούν (ΧΩΡΙΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟ)

# Η ιδέα του «διαίρεση τάσης»

&

## Serial monitor (παρατηρώ την αναλογικότητα)

Συνδέω μια  
αναλογική Είσοδο

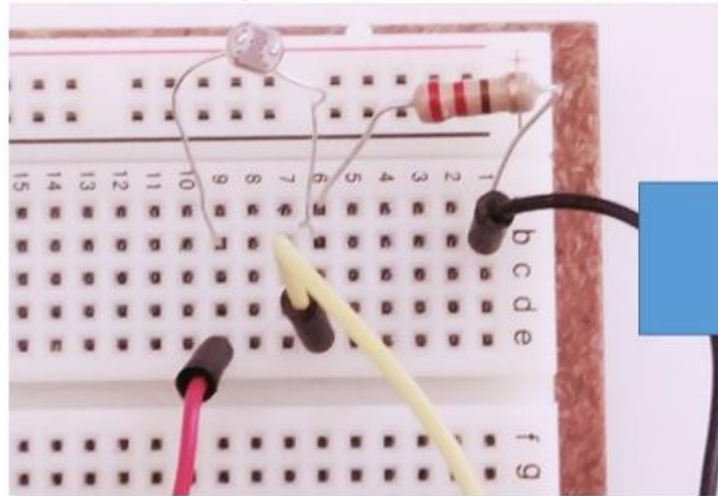
```
void setup() {  
  Serial.begin(9600);  
}  
  
void loop() {  
  int sensorValue = analogRead(A0);  
  float voltage = sensorValue * (5.0 / 1023.0);  
  Serial.println(voltage);  
}
```

AnalogRead()

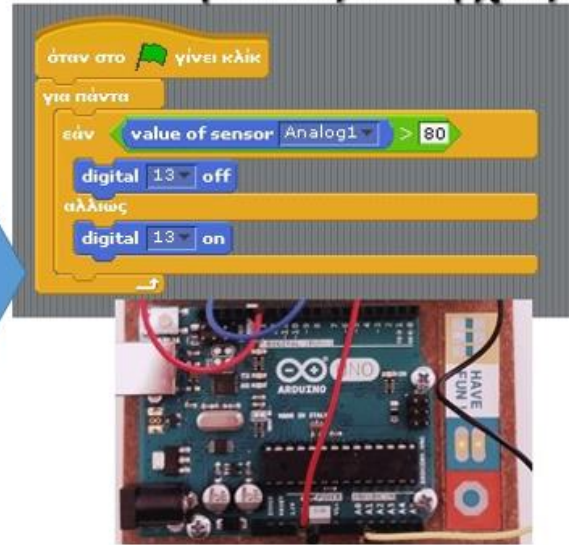
ΠΡΟΣΕΧΩΣ

Ελέγχουμε τον φωτισμό του χώρου με μια φωτοαντίσταση και ανάλογα έχουμε προγραμματίσει τότε θα ανάβει ένα led (φως).

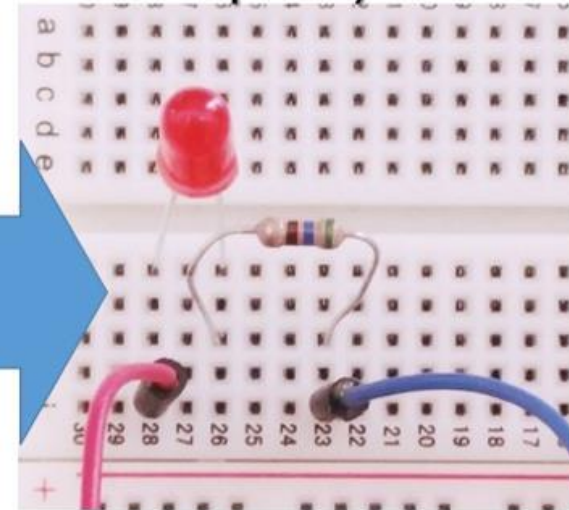
Κύκλωμα Εισόδου



Αυτόματος Έλεγχος



Κύκλωμα Εξόδου



Δείτε τα επιμέρους τμήματα του κυκλώματος και το μικρό πρόγραμμα που τα ελέγχει



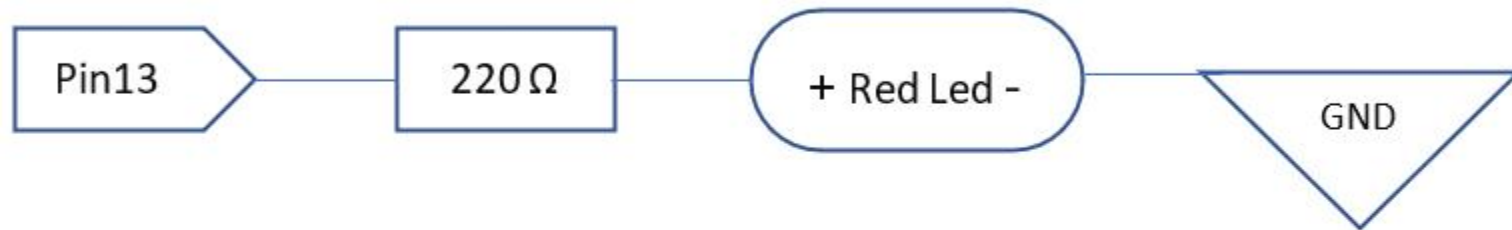


**ΑΠΟΘΗΚΗ ΠΑΛΑΙΟΤΕΡΩΝ «ΥΛΙΚΩΝ»**

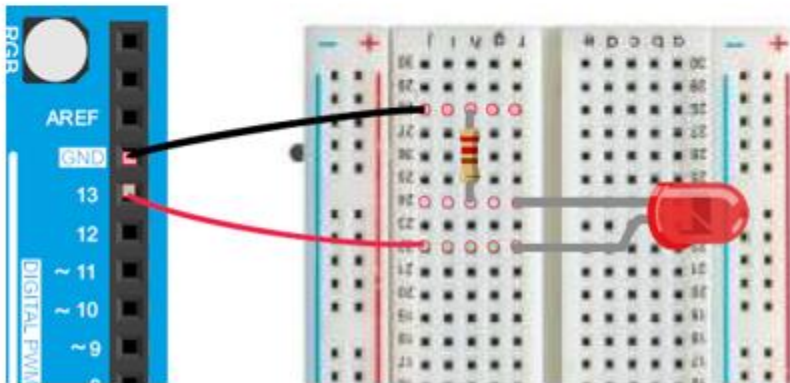
# Σύνδεση με τα προηγούμενα

Έλεγχος ενός led (αναβοσβήνει ανά 1 sec)

*Πρόχειρο σχεδίασμα κυκλώματος/*



**Σχέδιο κυκλώματος**

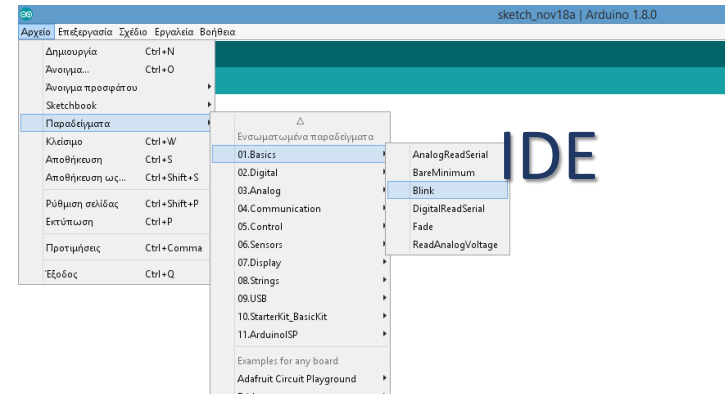


Εδώ βλέπετε το ισοδύναμο κύκλωμα

pin13 ---- +led- --- 220Ω --- GND

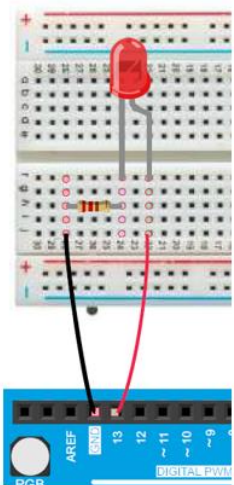
(η σειρά led/αντίσταση ή αντίσταση/led ΔΕΝ παίζει ρόλο)

# Σύνδεση με τα προηγούμενα



## Πηγαίος κώδικας (με σχόλια)

Θυμηθείτε πως μια εκδοχή του βρίσκεται στο IDE: ΑΡΧΕΙΟ > ΠΑΡΑΔΕΙΓΜΑΤΑ > BASICS > BLINK



```
// Το setup τρέχει μια φορά
void setup() {
  pinMode(13, OUTPUT); // Αρχικοποιεί το pin 13 ως έξοδο/output
}

// Το loop εκτελείται συνέχεια σε διαρκή επανάληψη
void loop() {
  digitalWrite(13, HIGH); // Ανάβει ένα LED που συνδέσαμε (σωστά) στο pin 13
                          // δίνοντας τάση (high voltage)
  delay(1000);           // περιμένει για 1 sec

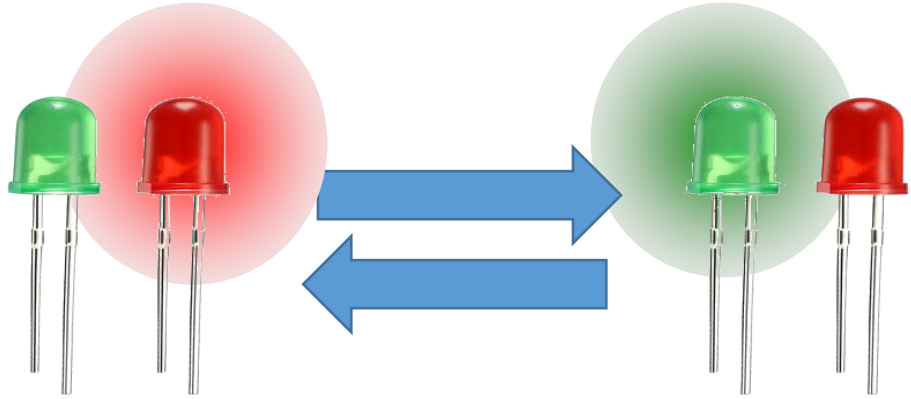
  digitalWrite(13, LOW); // Σβήνει το LED στο pin 13 κάνοντας το voltage LOW
  delay(1000);           // περιμένει για 1 sec
}
```

**Σχόλιο:** Έχω ΕΝΑ κύκλωμα που λειτουργεί σε ΔΥΟ καταστάσεις

Επιπλέον Σχόλιο: Δεν έχω ΕΙΣΟΔΟ, έχω Ψηφιακή Έξοδο

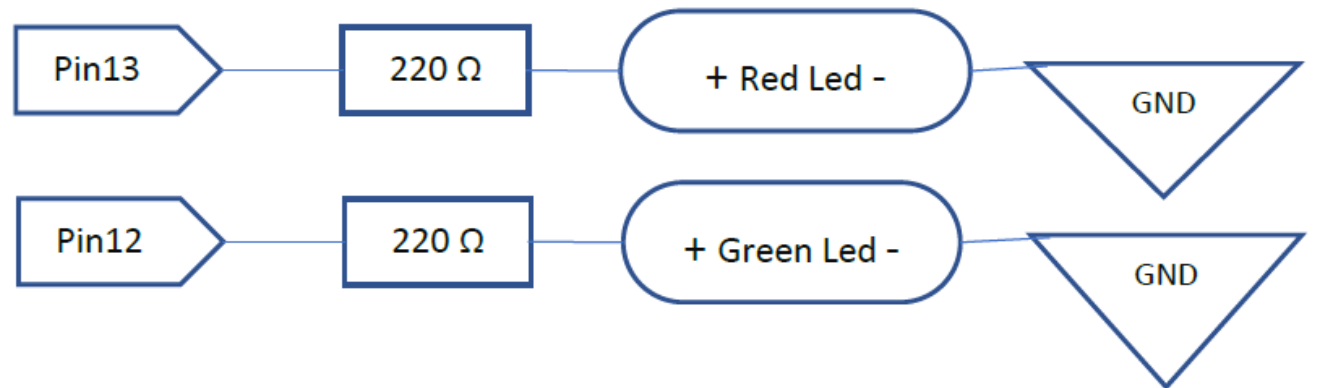


# Σύνδεση με τα προηγούμενα



Έλεγχος δύο led (αναβοσβήνουν εναλλάξ)

*Πρόχειρο σχεδιάσμα κυκλώματος*



### Πηγαίος κώδικας (χωρίς σχόλια)

```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(12, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(13, HIGH);  
  digitalWrite(12, LOW);  
  delay(1000);  
  
  digitalWrite(13, LOW);  
  digitalWrite(12, HIGH);  
  delay(1000);  
}
```

### Βελτίωση στον κώδικα (καλύτερη προγραμματιστική τεχνική)

```
int redledpin = 13;           // Δηλώνω δύο ΜΕΤΑΒΛΗΤΕΣ ακέραιου τύπου  
int greenledpin = 12;       // και τους δίνω αρχικές τιμές (Με το = δίνω τιμή)  
                               // Έτσι αν αλλάξω pins στο κύκλωμα αλλάζω το πρόγραμμα  
                               // μόνο σε ένα σημείο  
  
void setup() {  
  pinMode(redledpin, OUTPUT);  
  pinMode(greenledpin, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(redledpin, HIGH);  
  digitalWrite(greenledpin, LOW);  
  delay(1000);  
  
  digitalWrite(redledpin, LOW);  
  digitalWrite(greenledpin, HIGH);  
  delay(1000);  
}  
  
// Εντολή εκχώρησης τιμής: Με το = δίνω τιμή, δεν είναι το = των μαθηματικών
```

## Επέκταση: Φανάρι κυκλοφορίας οχημάτων (3 led red/yellow/green)

- Πόσα ανεξάρτητα κυκλώματα θα φτιάξουμε; .....
- Πόσες καταστάσεις λειτουργίας έχουμε; .....
- Σχεδιάστε πρόχειρα τα κυκλώματα
  
- Οργανώστε τη σκέψη σας

- Συμπληρώστε (στο χαρτί) τον πηγαίο κώδικα ακολουθώντας την (βελτιωμένη) τεχνική που μόλις χρησιμοποιήσαμε.

```
//Δηλώσεις μεταβλητών
```

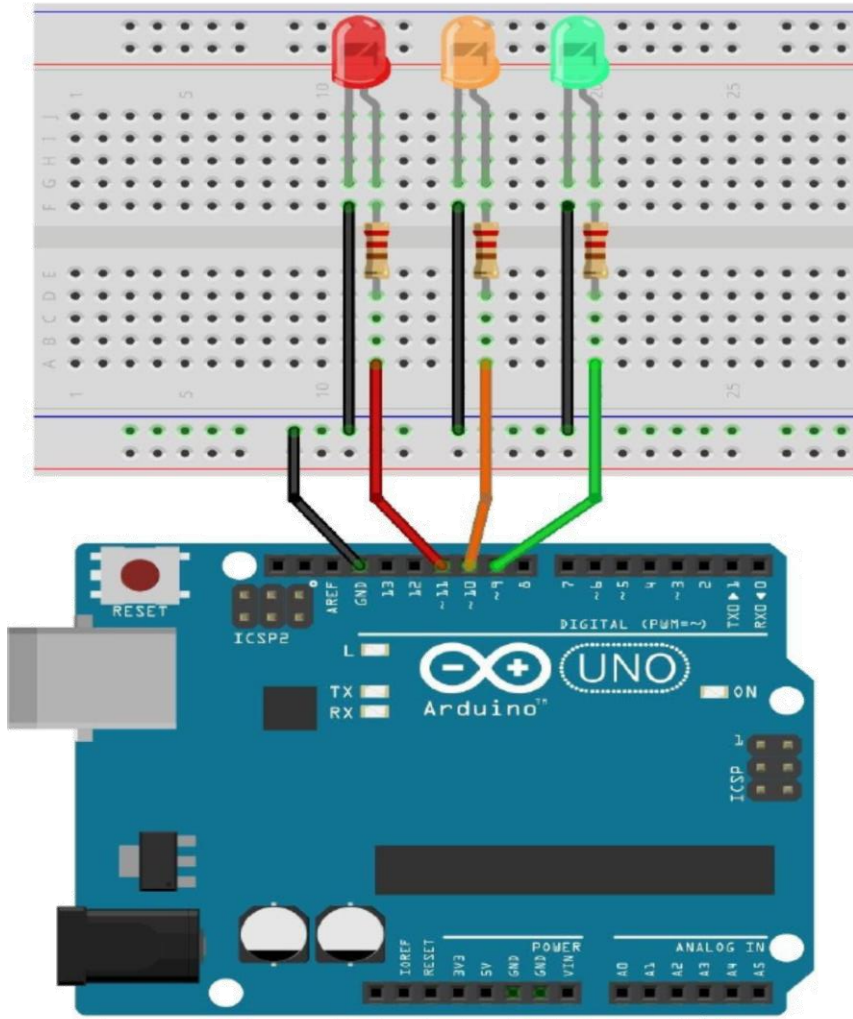
```
int
```

```
void setup() {
```

```
}
```

```
void loop() {
```

```
}
```



Κάτι σαν αυτό





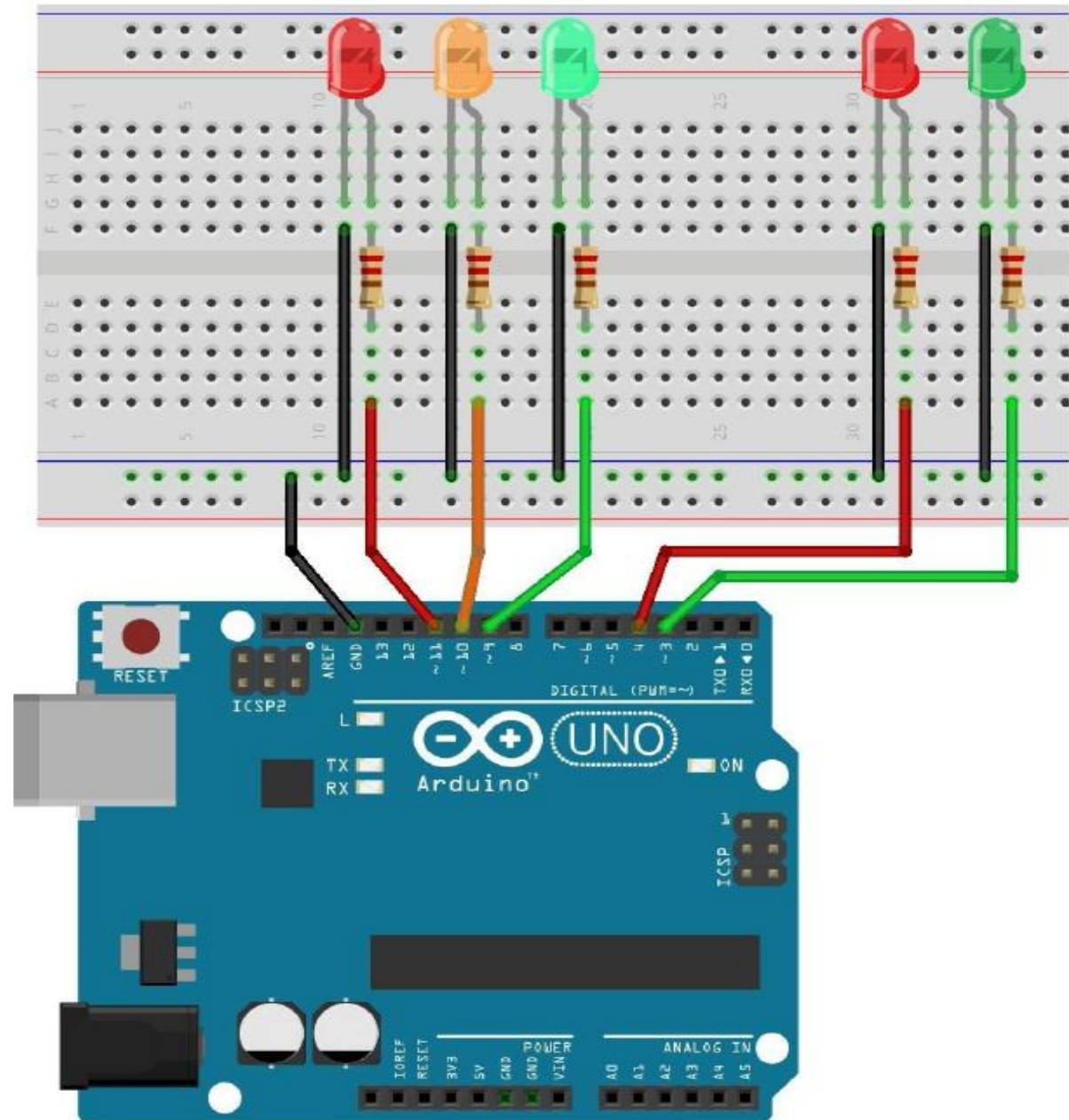
## PROJECT 1

### Φανάρι κυκλοφορίας για οχήματα με διάβαση πεζών σε διασταύρωση

Αντίστοιχη πορεία εργασίας

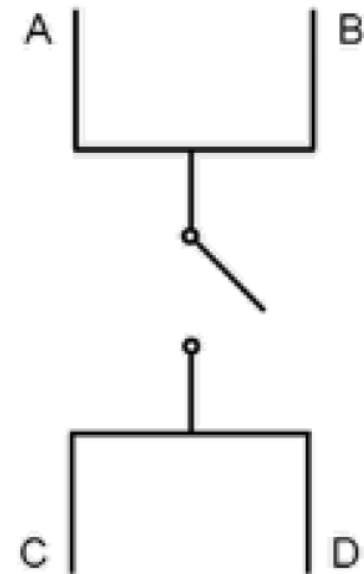
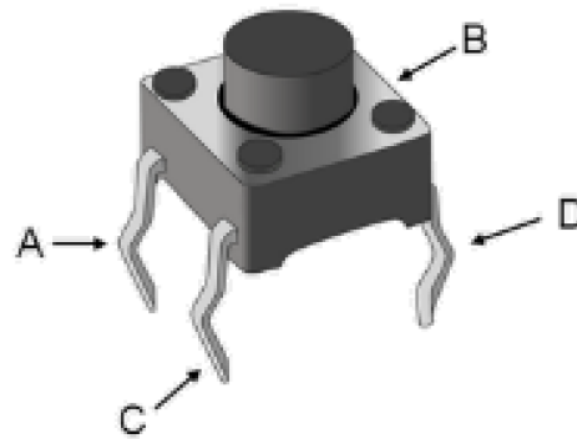


ΦΑΝΑΡΙ			ΦΑΝΑΡΙ ΠΕΖΩΝ		
Χρώμα	pin		Χρώμα	pin	
Πράσινο		ON	Πράσινο		OFF
Πορτοκαλί		OFF	Κόκκινο		ON
Κόκκινο		OFF			
Πράσινο			Πράσινο		
Πορτοκαλί			Κόκκινο		
Κόκκινο					
Πράσινο			Πράσινο		
Πορτοκαλί			Κόκκινο		
Κόκκινο					



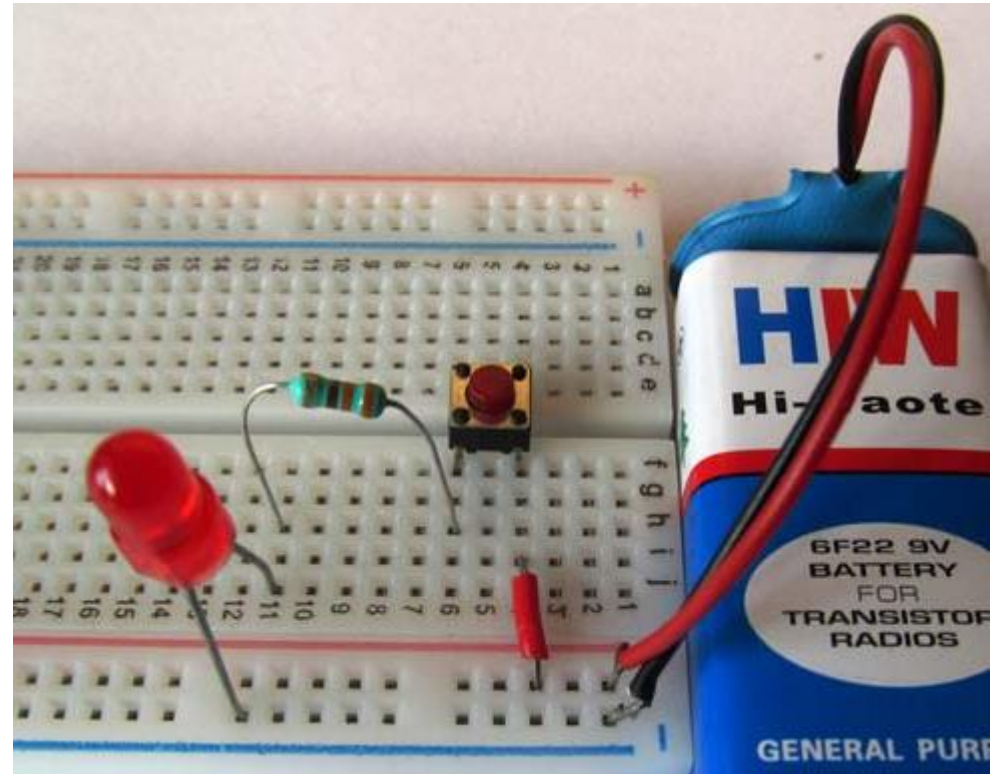
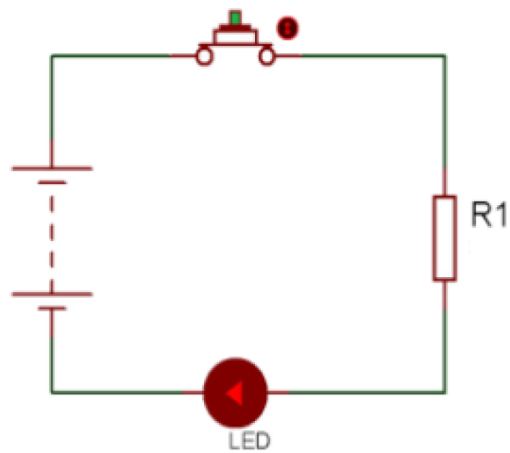
Κάτι σαν αυτό

# Έλεγχος led με διακόπτη (push button)



# Έλεγχος λειτουργίας διακόπτη

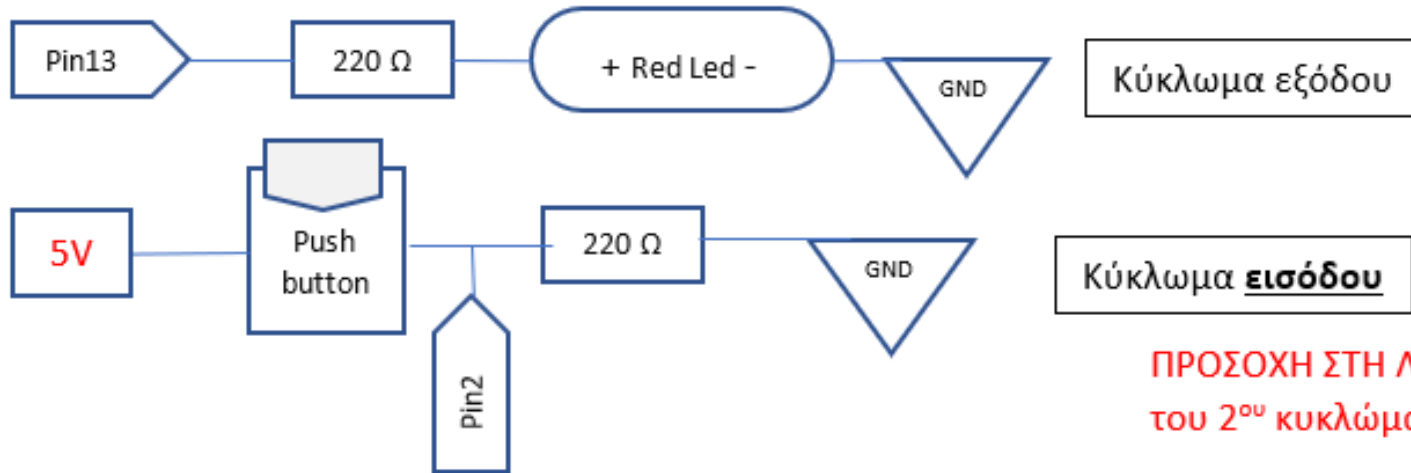
ενδεικτικά





# Ο διακόπτης push button ως ψηφιακή είσοδος

Πρόχειρο σχεδιάσμα κυκλώματος



ΠΡΟΣΟΧΗ ΣΤΗ ΛΟΓΙΚΗ του 2<sup>ου</sup> κυκλώματος!!!

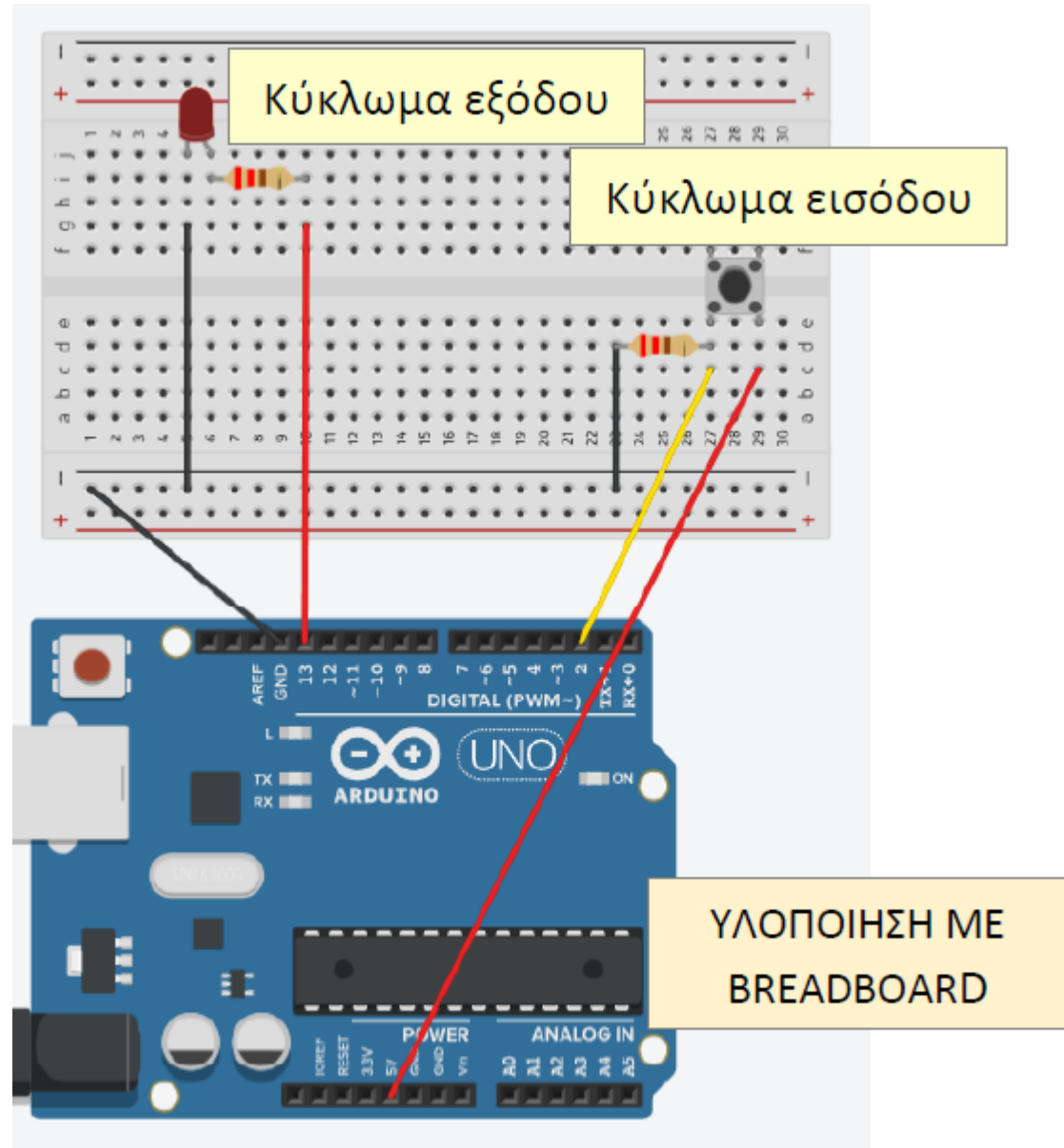
Κύκλωμα εξόδου

Κύκλωμα εισόδου

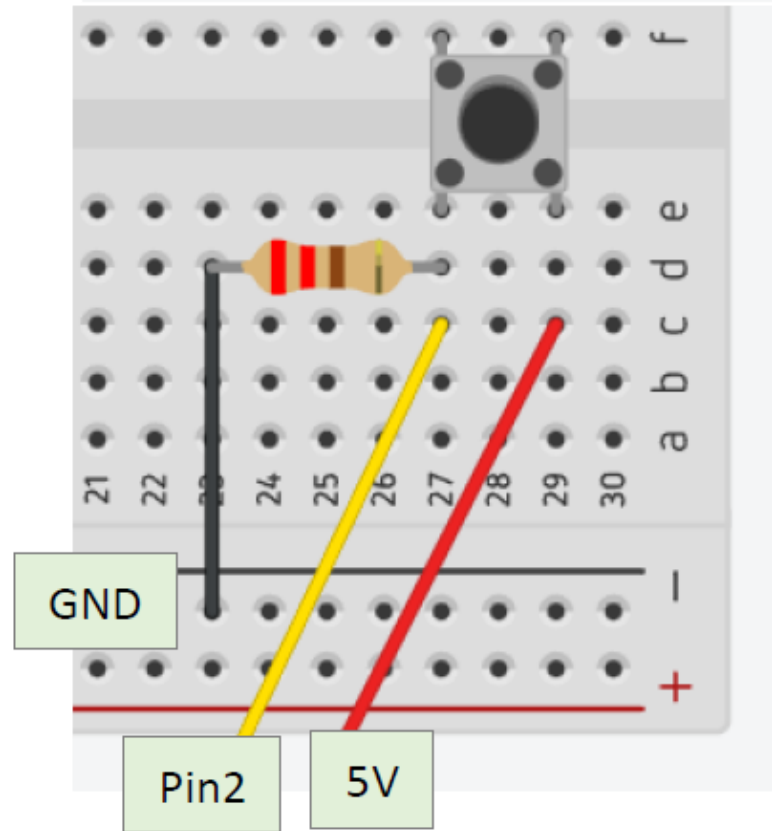
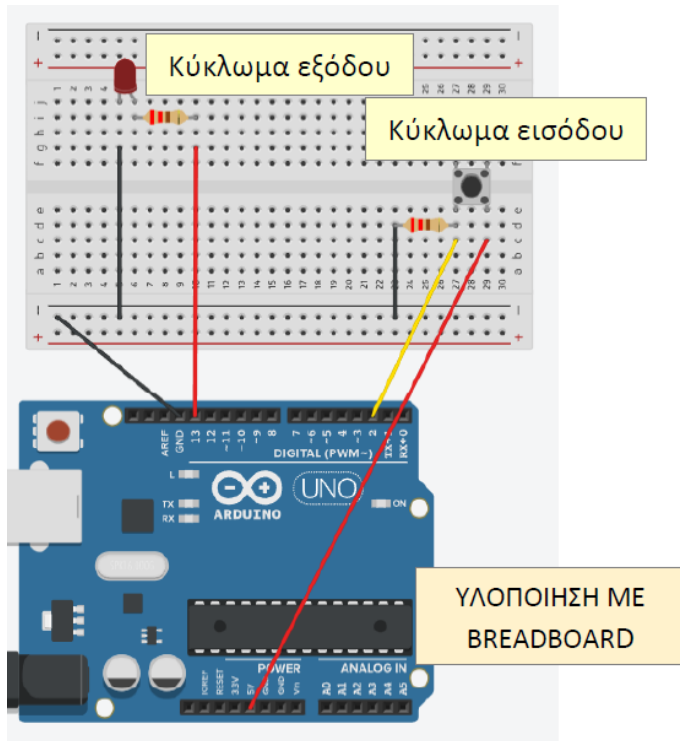
ΒΑΣΙΚΗ ΙΔΕΑ ΣΥΝΔΕΣΜΟΛΟΓΙΑΣ



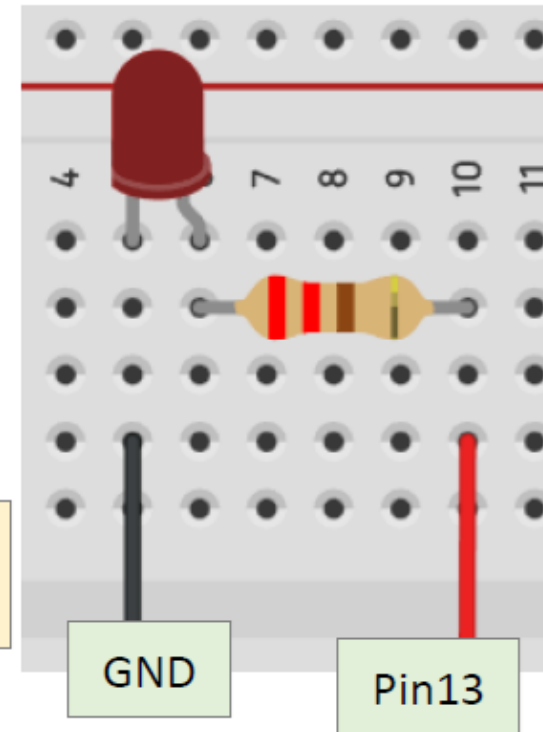
# Ο διακόπτης push button ως ψηφιακή είσοδος



# Ο διακόπτης push button ως ψηφιακή είσοδος



ΛΕΠΤΟΜΕΡΕΙΕΣ  
ΥΛΟΠΟΙΗΣΗΣ



## Ο διακόπτης push button ως ψηφιακή είσοδος

```
// Οι σταθερές (constants) ΔΕΝ μπορούν να αλλάξουν τιμή μέσα στο πρόγραμμα
const int redledPin = 13;    // LED pin (έστω κόκκινο)
const int buttonPin = 2;     // pushbutton pin

int buttonStatus = 0;       // μεταβλητή (variable) για ανάγνωση (είσοδο)
                             // του pushbutton status

void setup() {
  pinMode(redledPin, OUTPUT); // LED pin output
  pinMode(buttonPin, INPUT);  // pushbutton pin input
}

void loop() {
  buttonStatus = digitalRead(buttonPin); // «διαβάζω» την κατάσταση του pushbutton

  // Ελέγχω εάν το pushbutton πατήθηκε, οπότε το buttonState γίνεται HIGH:
  if (buttonStatus == HIGH) {           // Το == είναι ο τελεστής ελέγχου ισότητας
    digitalWrite(redledPin, HIGH);     // LED on
  } else {
    digitalWrite(redledPin, LOW);      // LED off
  }
}
```

Πως νομίζετε πως θα λειτουργεί η κατασκευή;