

Η πληροφορική της  
Γ' Γυμνασίου με την χρήση του

**Arduino**

μέσα από την πλατφόρμα

**Tinkercad**



Τιμολέων Θεοφανέλλης

ISBN 978-618-00-1983-4



Οι άδειες και τα εργαλεία πνευματικών δικαιωμάτων Creative Commons φέρνουν μια ισορροπία στην παραδοσιακή ρύθμιση “διατήρησης πλήρους δικαιώματος” που δημιουργεί η νομοθεσία για τα πνευματικά δικαιώματα. Προσφέρουν στον καθένα, έναν απλό, προτυποποιημένο τρόπο για να χορηγεί άδειες επί των δικαιωμάτων πνευματικής ιδιοκτησίας των έργων του. Ο συνδυασμός των εργαλείων που παρέχονται έχει δημιουργήσει μια δεξαμενή περιεχομένου που μπορεί να αντιγραφεί, διανεμηθεί, τροποποιηθεί, αναδομηθεί και αξιοποιηθεί πάντοτε εντός των ορίων της νομοθεσίας για τα πνευματικά δικαιώματα.



**Attribution-NonCommercial 4.0  
International (CC BY-NC 4.0)**

## Περιεχόμενα

Σημείωμα για τον εκπαιδευτικό .....	4
Βασικές Έννοιες.....	5
Κύκλωμα 1ο.....	8
Κύκλωμα 2ο.....	11
Κύκλωμα 3ο.....	13
Κύκλωμα 4ο.....	15
Κύκλωμα 5ο.....	16
Κύκλωμα 6ο.....	18
Κύκλωμα 7ο.....	19
Κύκλωμα 8ο.....	21
Κύκλωμα 9ο.....	23
Κύκλωμα 10ο.....	25
Κύκλωμα 11ο.....	27
Κύκλωμα 12ο.....	28
Εργασία για το σπίτι.....	33

## Σημείωμα για τον εκπαιδευτικό

Οι δραστηριότητες, Κύκλωμα (1 – 12) αποτελούνται από:

- την εικόνα της διάταξης που πρέπει να υλοποιηθεί
- το σύνδεσμο το tinkercad, μέσα από τον οποίο μπορεί κανείς να δει το κύκλωμα, πρόγραμμα, να τρέξει τον κώδικα και να πειραματιστεί με τη λειτουργία του
- την περιγραφή όσων χρειάζεται να γνωρίζουν οι μαθητές

Το υλικό αυτό δημιουργήθηκε κατά τη διδασκαλία του μαθήματος Πληροφορικής κατά το σχολικό έτος 2019 -2020.

Οι σημειώσεις αυτές είναι μια πρόταση για διδασκαλία του προγραμματισμού στη Γ' Γυμνασίου μέσα από μια σειρά δραστηριοτήτων. Σε κάποιο σημείο της ύλης ο εκπαιδευτικός μπορεί να χρησιμοποιήσει Arduino για να αποκτήσουν οι μαθητές μια αίσθηση πως είναι στην πραγματικότητα το Arduino. Η πρόταση είναι αυτό να γίνει Στην περίπτωση που το σχολείο διαθέτει εξοπλισμό για να το υλοποιήσουν σε ομάδες 3-4 ατόμων μπορεί να υλοποιηθούν αυτές ή και εμπλουτισμένες δραστηριότητες, αλλά ο απαιτούμενος χρόνος για την υλοποίηση των ασκήσεων θα αυξηθεί σημαντικά.

Η μεθοδολογία που αξιοποιείται είναι αυτή των σχεδίων δράσης (project). Ουσιαστικά πρόκειται για κατευθυνόμενη έρευνα σε ένα συγκεκριμένο θέμα και αφορά γεγονότα ή φαινόμενα του άμεσου περιβάλλοντος. Με αυτό τον τρόπο το όλο εγχείρημα αφορά και εμπλέκει τους μαθητές ενεργά. Οι μαθητές εργάζονται σε ομάδες, οπότε συνεργάζονται και αλληλοεπιδρούν σε ομάδες.

## Ευχαριστίες

Ευχαριστώ το συνάδελφο Βασίλη Μπαρέκο που κοίταξε τις σημειώσεις και για τα σχόλια του, που θεωρώ ότι βελτίωσαν το τελικό αποτέλεσμα.

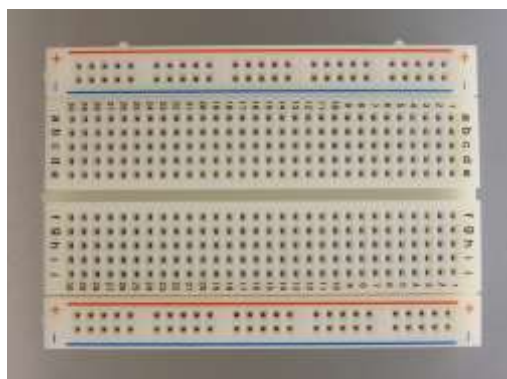
## Βασικές Έννοιες

Το Tinkercad είναι μια εφαρμογή που κάνει **προσομοίωση** για ηλεκτρονικά κυκλώματα και επιτρέπει την χρήση της πλακέτας **Arduino**. Πρόκειται για μία ολοκληρωμένη πλατφόρμα ανάπτυξης έργων ηλεκτρονικής, αυτοματισμού και ρομποτικής, η οποία παρέχει ελεύθερα το αναγκαίο λογισμικό και υλικό.

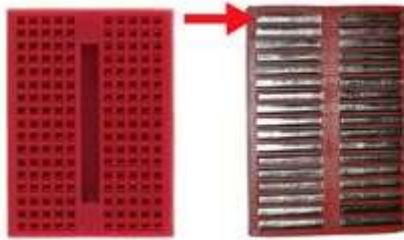


Το Arduino διαθέτει ψηφιακούς ακροδέκτες εισόδου/εξόδου (1-12) αναλογικής εισόδου, αν και πρακτικά χρησιμοποιούνται οι ψηφιακές έξοδοι 2 -13. Επιπλέον διαθέτει ακροδέκτες αναλογικής εισόδου (A0 -A5) και ακροδέκτες τροφοδοσίας (power). Στο πάνω μέρος υπάρχει σύνδεση USB που χρησιμοποιείται για φόρτωση προγράμματος, για σειριακή επικοινωνία με τον υπολογιστή αλλά και για τροφοδοσία. Στο κάτω αριστερά μέρος υπάρχει η υποδοχή για την εξωτερική τροφοδοσία 7 - 12 V.

Το **breadboard** είναι μία διάταξη που επιτρέπει την εύκολη και προσωρινή κατασκευή κυκλωμάτων χωρίς να απαιτούνται κολλήσεις. Διαθέτει οπές πάνω στις οποίες



μπορούν να συνδεθούν διάφορα ηλεκτρικά και ηλεκτρονικά στοιχεία. Εσωτερικά οι οπές αυτές συνδέονται μεταξύ τους κάθετα όπως φαίνεται στο σχήμα.



Οι 4 εξωτερικές οριζόντιες σειρές οπών (2 πάνω και 2 κάτω) είναι συνδεδεμένες οριζόντια και υπάρχει βοηθητικά η ένδειξη + και -. Στο - συνήθως συνδέουμε τη γείωση (ground -GND).

Στο κεντρικό τμήμα του breadboard οι συνδέσεις είναι κατακόρυφες και η κάθε στήλη χωρίζεται σε δύο ανεξάρτητα τμήματα των 5 οπών το κάθε ένα.

Ένα βίντεο που δείχνει τη λειτουργία του <https://www.youtube.com/watch?v=6WReFkfrUIk>

Το Arduino είναι μια μητρική πλακέτα ανοιχτού κώδικα η οποία περιέχει ένα ενσωματωμένο μικροελεγκτή και διαθέτει Εισόδους και εξόδους. Ο λόγος που υπάρχει μεγάλη χρήση του είναι τα πλεονεκτήματα που έχει σε σχέση με τον ανταγωνισμό:

- **Χαμηλό κόστος** αφού η αγορά ξεκινάει από 4 ευρώ και δωρεάν μεταφορικά από διαδικτυακό κατάστημα στην Κίνα. Ενώ το starter kit που διαθέτει πολλά εξαρτήματα (αισθητήρες, διακόπτες, οθόνες, καλώδια) κοστίζει περίπου 24 ευρώ. Επιπλέον το λογισμικό του είναι δωρεάν.
- Ως προς τη **συμβατότητα**, το λογισμικό λειτουργεί σε όλα τα γνωστά περιβάλλοντα Windows, Macintosh και Linux.
- Το περιβάλλον ανάπτυξης είναι **απλό** στην κατανόηση και τη χρήση ώστε μπορεί να χρησιμοποιείται εύκολα από αρχάριους χρήστες.
- Το λογισμικό είναι ανοικτού κώδικα και είναι **επεκτάσιμο/τροποποιήσιμο**.

Ο προγραμματισμός γίνεται μέσω της θύρας USB από την οποία τροφοδοτείται και με ρεύμα. Ωστόσο μπορεί να τροφοδοτηθεί και από εξωτερική τροφοδοσία που παρέχεται μέσω της υποδοχής 2.1mm (με το θετικό πόλο στο κέντρο) που υπάρχει στην κάτω-αριστερή γωνία. Η εξωτερική τροφοδοσία πρέπει να είναι από 7 ως 12V

και μπορεί να γίνει από κοινό μετασηματιστή του εμπορίου, από μπαταρίες ή οποιαδήποτε άλλη πηγή συνεχούς τάσης (DC).

The screenshot shows search results for "arduino uno starter kit" with 3,416 results. The interface includes filters for price, shipping, and sorting options. Four product listings are visible:

- Starter Kit for Arduino Uno R3 - Bu...**: Price € 6,63, Free Shipping, 4.8 rating, 683 Sold. Store: EGV Official Store.
- cnc shield V3 engraving machine 3...**: Price € 7,73, Free Shipping, 4.8 rating, 165 Sold. Store: T2T Official Store.
- 2019 The Most cost-effective DIY P...**: Price € 21,32, Free Shipping, 4.8 rating, 2859 Sold. Store: Robotlinking Store.
- NEWEST RFID Starter Kit for Ardui...**: Price € 19,95, Free Shipping, 4.9 rating, 7552 Sold. Store: SHENZHEN LIANSHENGDA Electro...

Στο παρακάτω βίντεο εξηγείται ο τρόπος λειτουργίας του Arduino <https://www.youtube.com/watch?v=nL34zDTPkcs>. Αναφέρονται παραπάνω

The screenshot shows search results for "arduino uno" with 2,972 results. The interface includes filters for price, shipping, and sorting options. Two product listings are visible:

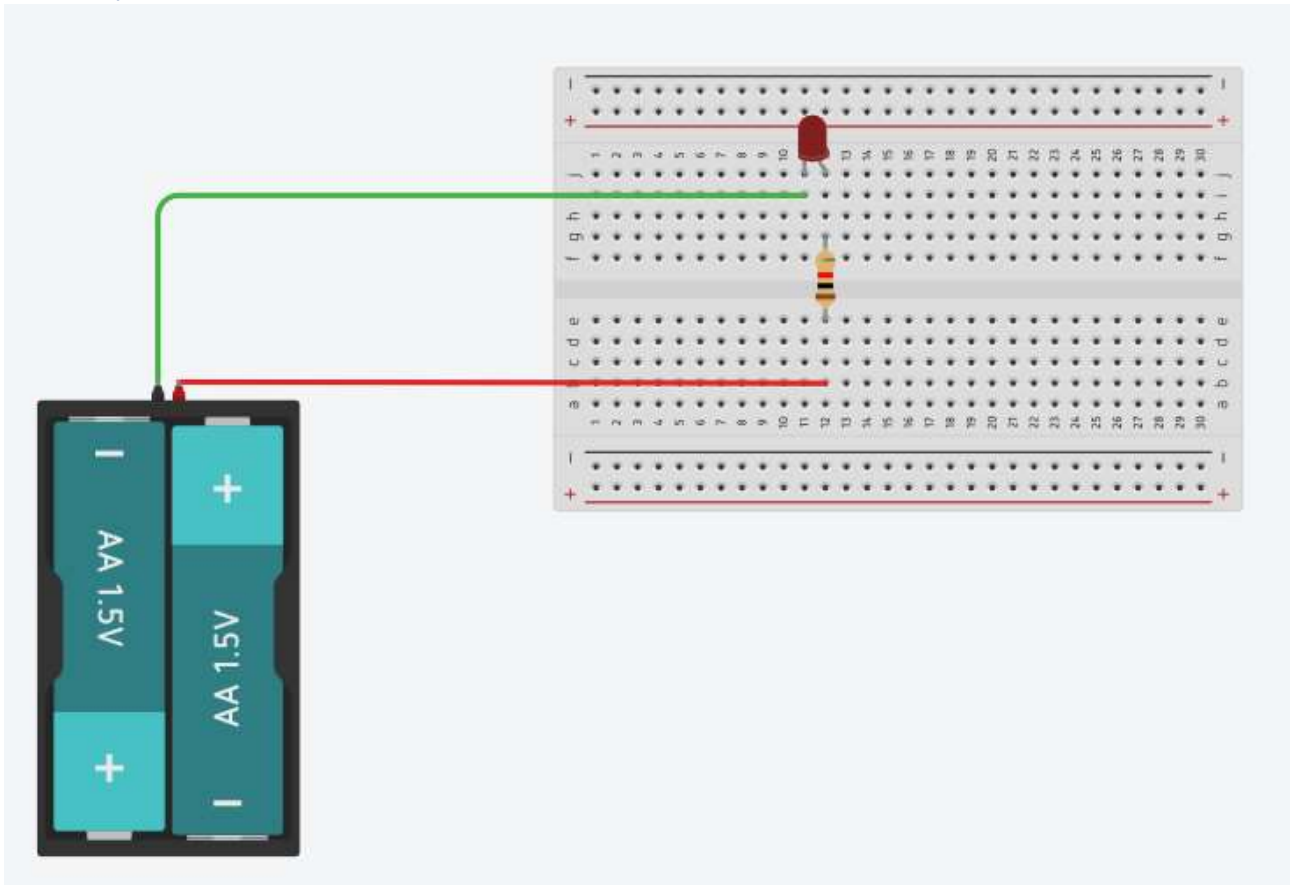
- UNO R3 Development Board ATmega328P CH340 CH340G For Arduino UNO R3 With Straight Pin...**: Price € 3,29 - 4, Free Shipping, 4.9 rating, 5600 Sold. Store: Robotlinking Store.
- 1 Set UNO R3 Official Box ATMEGA16U2+MEGA328P Chip For Arduino UNO R3 Development board + U...**: Price € 2,75 - 4,09, Free Shipping, 4.9 rating, 2072 Sold. Store: All goods are free shipping Store.

πράγματα από όσα χρειάζονται να ξέρουν οι μαθητές, αλλά είναι καλό να τα γνωρίζει ο εκπαιδευτικός. Στους μαθητές μπορούμε να δείξουμε τμήματα αυτού του βίντεο.

Η εφαρμογή μπορεί να χρησιμοποιηθεί για τη Φυσική και την Πληροφορική της Γ' Γυμνασίου ή ακόμα και στο μάθημα της Τεχνολογίας. Ενώ το βέλτιστο είναι η συνεργασία των εκπαιδευτικών που διδάσκουν αυτά τα τρία μαθήματα.



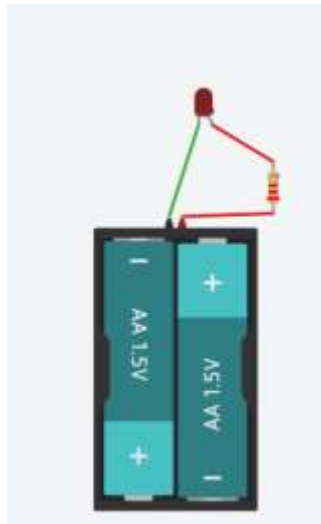
## Κύκλωμα 1ο



Στο **1ο** κύκλωμα (<https://www.tinkercad.com/things/6WftvQDXnC2>) οι μαθητές καλούνται να χρησιμοποιήσουν το breadboard για να συνδέσουν μια μπαταρία (που πρέπει να την κάνουν αργότερα 3V) με ένα LED μέσω μιας αντίστασης. Με τον τρόπο αυτό αντιλαμβάνονται το θετικό πόλο μπαταρίας και τον τρόπο που συνδέουμε το led σε ένα κύκλωμα. Κατόπιν τους ζητάμε να αλλάξουν την αντίσταση σε 220 Ohm και τη μπαταρία σε 3V και να δουν αν υπάρχει κάποια διαφορά στο φως που βγάζει το LED. Επίσης τους ζητάμε να συνδέσουν το LED ανάποδα για να διαπιστώσουν ότι δεν ανάβει.

Το κύκλωμα αυτό μπορεί αρχικά ή μετά να υλοποιηθεί χωρίς το breadboard και να συζητηθούν οι διαφορές των δυο υλοποιήσεων.





Το LED (Light Emitting Diode) είναι ένα στοιχείο, το οποίο όταν διαρρέεται από ρεύμα φωτοβολεί. Όσο μεγαλύτερη είναι η ένταση του ρεύματος που διαρρέει το LED, τόσο εντονότερο είναι το φως που παράγεται. Ως δίοδος, το LED επιτρέπει τη διέλευση του ρεύματος μόνο προς μία φορά (το στραβό ποδαράκι το συνδέουμε στο θετικό, ενώ την άλλη άκρη στον αρνητικό πόλο (- ή GND ή γείωση).

Προτείνεται να γίνει μια συζήτηση για το τι είναι και πως λειτουργεί ένα LED. Αυτό μπορεί να γίνει μέσα από το βίντεο [https://www.youtube.com/watch?v=Yo6Jl\\_bzUzo](https://www.youtube.com/watch?v=Yo6Jl_bzUzo) ή με άλλο τρόπο που θεωρεί κατάλληλο ο εκπαιδευτικός.

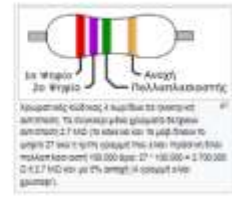
Επίσης είναι καλό να μιλήσουμε στους μαθητές για τις αντιστάσεις, αυτό μπορεί να γίνει και μέσα από αυτό το βίντεο <https://www.youtube.com/watch?v=NfcgA1axPLo> και μπορεί να γίνει και σε συνεργασία με τον συνάδελφο που διδάσκει Φυσική.

Στο σημείο αυτό μαθαίνουν και για το χρωματικό κώδικα των αντιστάσεων. Αυτό που χρειάζεται είναι να μπορούν να αναγνωρίζουν μια αντίσταση ανάλογα με τα χρώματα χρησιμοποιώντας τον πίνακα που έχει το χρωματικό κώδικα των αντιστάσεων. Θα μπορούσαμε να κάνουμε αλλαγές στην αντίσταση ώστε να μάθουν πως λειτουργεί ο παρακάτω πίνακας και με το πολύμετρο μπορούμε να βλέπουμε πως αλλάζει η ένταση του ρεύματος και η τάση ανάλογα με τις τιμές που παίρνει η αντίσταση.

**Ο Χρωματικός Κώδικας** | Εντάξεις | Στοιχεία κώδικα

Στην παρούσα σελίδα παρουσιάζονται οι ταινίες που χρησιμοποιούνται σε αλληλκίνητα σύρματα με το πρότυπο IEC 60062 από 0 της Διεθνούς Επιτροπής Ηλεκτροτεχνικών (IEC - International Electrotechnical Commission) [9] [10]

Χρώμα	1 <sup>η</sup> Λοβίδα	2 <sup>η</sup> Λοβίδα	3 <sup>η</sup> Λοβίδα (πολλαπλασιαστής)	4 <sup>η</sup> Λοβίδα (ακρίβεια)
Μαύρο	0	0	$\times 10^0$	
Κίτρινο	1	1	$\times 10^1$	$\pm 5\%$ (B)
Κόκκινο	2	2	$\times 10^2$	$\pm 5\%$ (B)
Πράσινο	3	3	$\times 10^3$	
Μπλε	4	4	$\times 10^4$	
Πορτοκάλι	5	5	$\times 10^5$	$\pm 0,5\%$ (A)
Μαύρο	6	6	$\times 10^6$	$\pm 0,25\%$ (C)
Κίτρινο	7	7	$\times 10^7$	$\pm 0,1\%$ (D)
Κόκκινο	8	8	$\times 10^8$	$\pm 0,05\%$ (E)
Μαύρο	9	9	$\times 10^9$	
Χρυσό			$\times 0,1$	$\pm 5\%$ (B)
Ασημί			$\times 0,01$	$\pm 10\%$ (F)
Καφέ				$\pm 20\%$ (G)



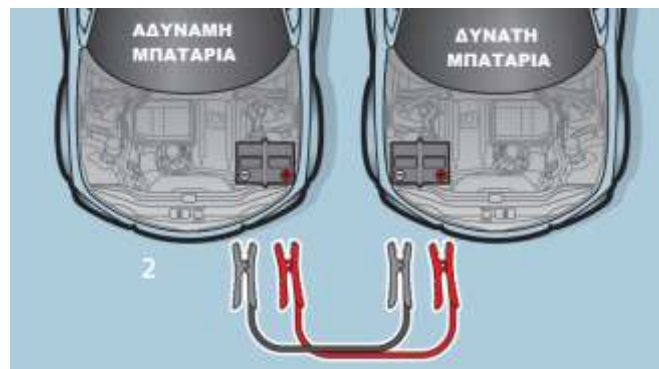
**Παραδείγματα** | Εντάξεις | Στοιχεία κώδικα

Στην εικόνα που βλέπουμε στο δεξί μέρος αλληλκίνητα σύρματα, πάνω στην οποία βρίσκονται 4 λοβίδες χρωμάτων. Οι πρώτες δύο λοβίδες συμβολίζουν αριθμό, η τρίτη λοβίδα καθορίζει τον πολλαπλασιαστή και η τέταρτη την ακρίβεια της αντίστασης. Χρησιμοποιώντας τον πίνακα του χρωματικού κώδικα μπορούμε να εξαγάγουμε την τιμή της αντίστασης με βάση τα χρώμα που συμβολίζουν την μονάδα, το πρώτο χρώμα συμβολίζει το πρώτο, το δεύτερο χρώμα συμβολίζει το 1000 και το χρυσό καθορίζει πως η ακρίβεια είναι 5%. Άρα η τιμή της αντίστασης θα είναι:  $15 \times 1000 = 15000$  με ακρίβεια 5%.

Σε δεύτερο παράδειγμα θα θεωρήσουμε την αντίσταση άγνωστη που φαίνεται στο δεξί και θα προσπαθήσουμε να διαβάσουμε την τιμή της. Αναστέλλεται το πρόβλημα διότι να χρησιμοποιήσουμε τον χρωματικό κώδικα έχουμε. Το πρώτο συμβολίζει το 1, το δεύτερο συμβολίζει το 1, το τρίτο συμβολίζει το 100 και το χρυσό καθορίζει την ακρίβεια 5%. Άρα η τελική τιμή της αντίστασης προκύπτει:  $15 \times 100 = 15000 = 1,5k\Omega$  με ακρίβεια 5%. Η ακρίβεια υπολογίζεται καθορίζοντας το όριο πριν που μπορεί να πάρει η αντίσταση. [9] Στο συγκεκριμένο παράδειγμα, η αντίσταση θα πρέπει να βρίσκεται ανάμεσα στις τιμές  $1500 - 0,05 \times 1500 = 1425\Omega$  και  $1500 + 0,05 \times 1500 = 1575\Omega$ .



Στο σημείο αυτό υπάρχει η δυνατότητα να μιλήσουμε στους μαθητές για τη χρήση των χρωμάτων που έχουν τα καλώδια ότι με το κόκκινο καλώδιο συνδέουμε το θετικό (+) πόλο και με το μαύρο καλώδιο με τον αρνητικό (-) πόλο.

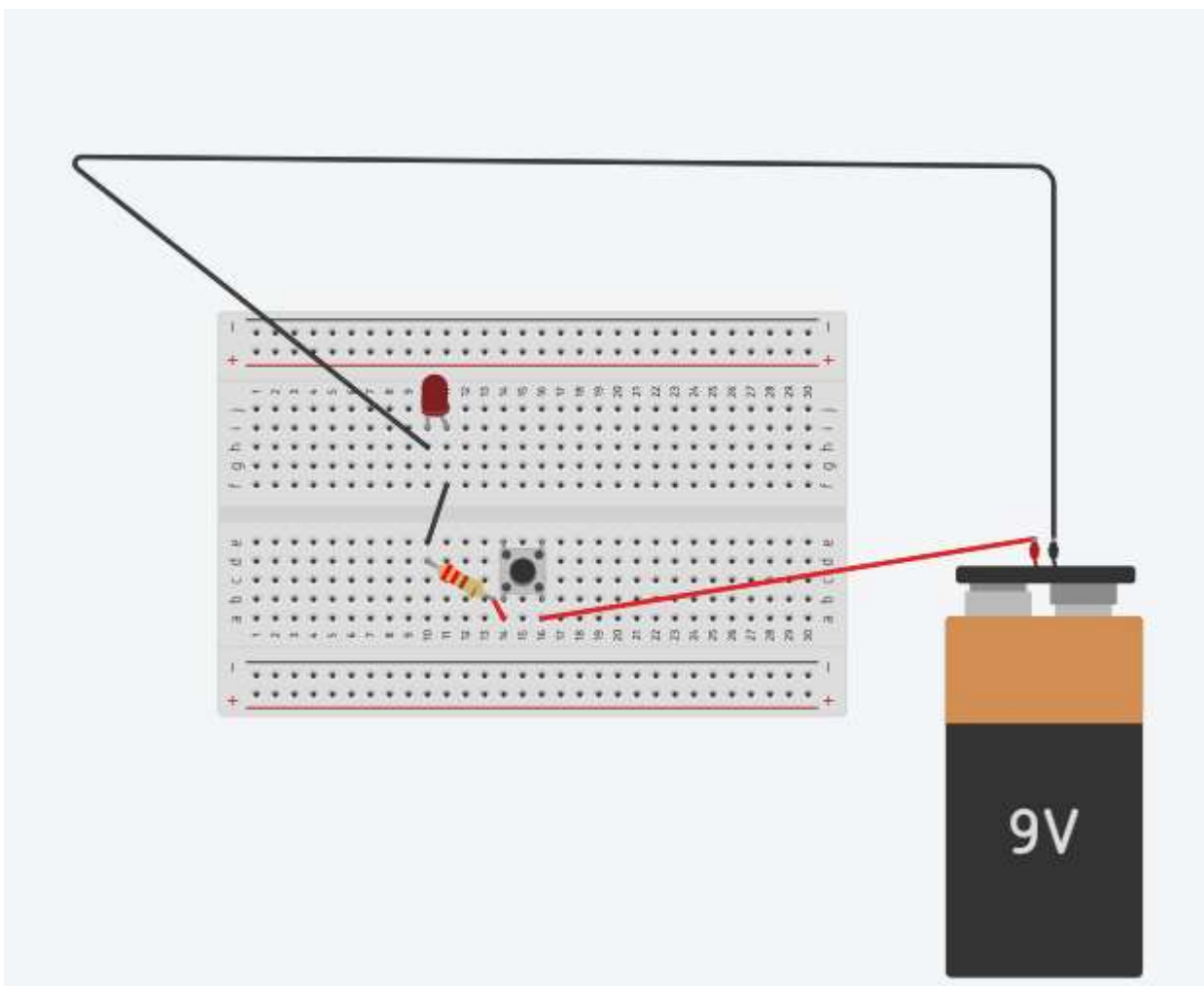


## Κύκλωμα 2ο

Στο **2ο κύκλωμα** συνδέουμε στο κύκλωμα ένα διακόπτη και μια μεγαλύτερη μπαταρία (9V)

<https://www.tinkercad.com/things/jBu24FIVids>

Αυτό θα μπορούσε να είναι ένα σύστημα που στέλνουμε σήματα morse. Στην θέση από το λαμπάκι θα μπορούσε να μπει ένα buzzer ή το buzzer να μπει παράλληλα με το LED. Επίσης θα μπορούσε να μπει και σε σειρά ώστε να συζητηθεί η φωτεινότητα και ο ήχος στα διαφορετικά είδη συνδεσμολογίας. Η όλη συνδεσμολογία θα μπορούσε να έχει ένα “άρωμα” STEM αν έχουμε μαθητές που γνωρίζουν τον κώδικα morse πχ από τους προσκόπους ή με τη χρήση του ιστοτόπου <https://morsecode.world/international/translator.html>.



Στο σημείο αυτό θα μπορούσαν να μπούνε δραστηριότητες που θα αλλάζαμε την τάση της μπαταρίας, τις αντιστάσεις, θα δοκιμάζαμε την παράλληλη και σε σειρά σύνδεση και θα βλέπαμε την τάση και την ένταση του ρεύματος με το πολύμετρο. Κάθε γραμμή του breadboard έχει 10 τρύπες, που χωρίζονται από ένα αυλάκι στο μέσο. Το αυλάκι αυτό χωρίζει και απομονώνει τις 2 λωρίδες της γραμμής, έτσι το breadboard είναι χωρισμένο σε δυο τμήματα το πάνω και το κάτω.

## Κύκλωμα 3ο

Στο **3ο κύκλωμα** χρησιμοποιούμε το Arduino για να αναβοσβήνουμε ένα LED. Επίσης μαθαίνουμε να γράφουμε κώδικα και να τον παραμετροποιούμε

Αρχικά θα πρέπει να γίνει επιλογή της πλακέτας που θα χρησιμοποιηθεί μέσα από το περιβάλλον προγραμματισμού. Tools -> Board -> Arduino/Genuino Uno ή όποια άλλη έκδοση του Arduino χρησιμοποιούμε.

Κατόπιν πρέπει να επιλεγεί η θύρα επικοινωνίας Tools -> Port -> π.χ. COM3

Ο προγραμματισμός του Arduino περιλαμβάνει ένα πρόγραμμα κυκλικής εκτέλεσης που διαθέτει δυο βασικές λειτουργίες:

- `setup()`: συνάρτηση που εκτελείται μία φορά στην αρχή του προγράμματος και αρχικοποιεί τις ρυθμίσεις
- `loop()`: συνάρτηση που εκτελείται συνέχεια μέχρι η πλακέτα να απενεργοποιηθεί

Θα χρησιμοποιηθούν οι εντολές:

**pinMode** (... αριθμός του ακροδέκτη pin που θα ρυθμιστεί , ... INPUT ή OUTPUT);  
ορίζει το συγκεκριμένο pin του Arduino που θα χρησιμοποιηθεί είτε ως ΕΙΣΟΔΟ (INPUT) είτε ως ΕΞΟΔΟ (OUTPUT). Κάθε εντολή τελειώνει με το ελληνικό ερωτηματικό και αυτό σηματοδοτεί το τέλος της. Π.χ `pinMode(5, OUTPUT)`;

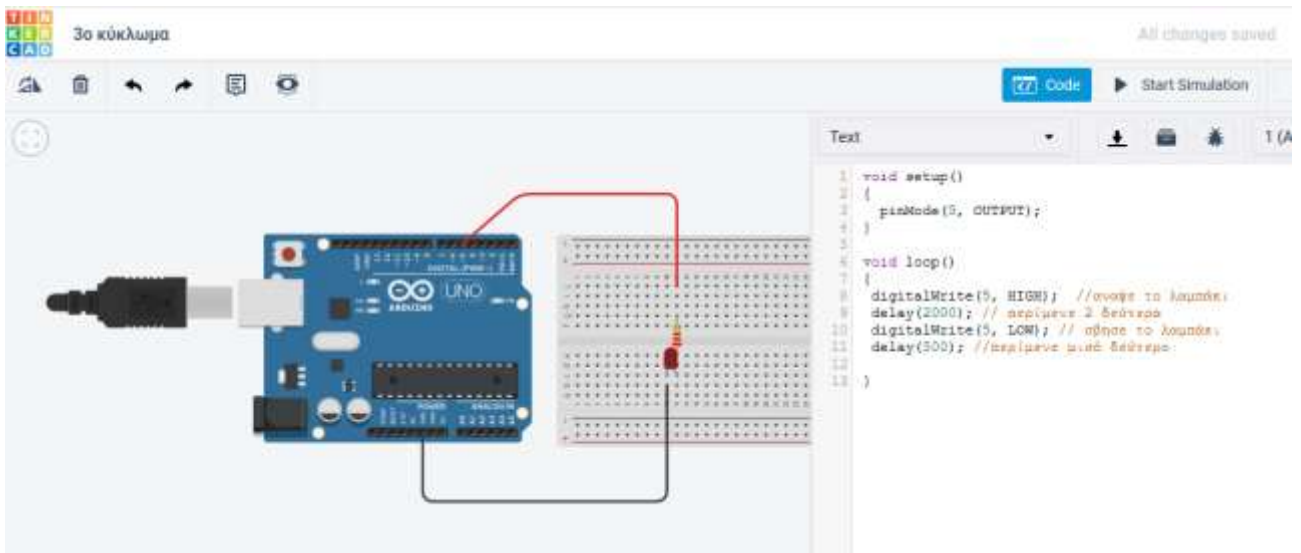
**digitalWrite**( ... αριθμός της ακίδας, ... HIGH ή LOW);  
που κάνει την αντίστοιχη ακίδα να βγάζει + (5V) ή - (GND). Ο αριθμός της ακίδας μπορεί να είναι από 0 ως 13 (αναφερόμαστε στις ψηφιακές ακίδες). Π.χ. `digitalWrite(5, HIGH)`;

**delay**(... χρόνος σε ms);  
που σταματάει τη ροή του προγράμματος για ορισμένο χρόνο, τον οποίο δίνουμε σε χιλιοστά του δευτερολέπτου (millisecond ή ms). Π.χ. `delay(2000)`; για καθυστέρηση 2 δευτερόλεπτα.

### // Σχόλια

Οι δύο κάθετες τη μια δίπλα στην άλλη δείχνουν στον μεταγλωττιστή ότι πρέπει να αγνοήσει ότι ακολουθεί σε εκείνη τη σειρά. Ουσιαστικά είναι σημείωση (σχόλιο) του προγραμματιστή, που εξηγεί ή θυμίζει τι κάνει κάθε εντολή.

<https://www.tinkercad.com/things/koYZtkrQaY1>



Το Arduino χρησιμοποιείται ως μία προγραμματιζόμενη πηγή αλλά και για την τροφοδοσία του κυκλώματος. Συγκεκριμένα όταν ο ψηφιακός ακροδέκτης 5 είναι σε κατάσταση HIGH (το LED ανάβει), ενώ όταν είναι σε κατάσταση LOW το LED σβήνει. Ο ακροδέκτης γείωσης (GND) παίζει το ρόλο του αρνητικού πόλου της πηγής.

Τα δυο τμήματα του προγράμματος για την ακρίβεια είναι δύο διαδικασίες (υποπρογράμματα) που υπάρχουν σε κάθε πρόγραμμα και είναι η `setup()` και η `loop()`.

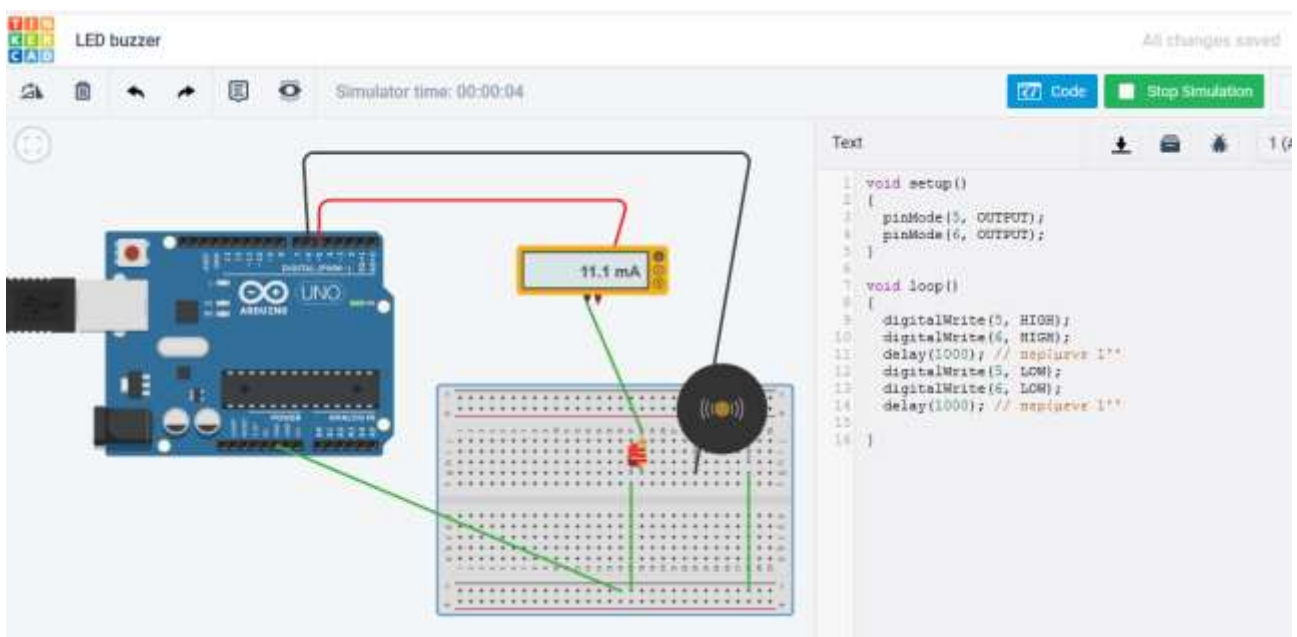
- `setup` σε αυτή μπαίνουν οι εντολές που θα εκτελεστούν μία φορά μόνο στην αρχή του προγράμματος.
- `loop` σε αυτή μπαίνουν οι εντολές που θα επαναλαμβάνονται μέχρι να αποσυνδέσουμε το Arduino από την τάση τροφοδοσίας ή να πατήσουμε το κουμπί Reset.

## Κύκλωμα 4ο

Στο κύκλωμα 4 χρησιμοποιούμε ένα LED και ένα βομβητή (buzzer) που είναι ένα μικρό μονοτονικό ηχείο. Στην καθημερινή ζωή μας συναντάμε τέτοια ηχεία σε υπολογιστές, ξυπνητήρια, σε συστήματα συναγερμού ή και στα πληκτρολόγια των συναλλαγών που διαθέτουν οι τράπεζες.

Το κύκλωμα αυτό το pin 6 δίνει στο βομβητή και το pin 5 στο LED. Στο LED έχει προστεθεί και ένα πολύμετρο που χρησιμοποιείται ως αμπερόμετρο και μετράει την ένταση του ρεύματος που περνάει από το LED. Όπως είναι η συνδεσμολογία όταν ανάβει το LED δουλεύει και ο βομβητής. Θα μπορούσε να δοθεί στους μαθητές για εξάσκηση να το κάνουν εναλλάξ και μετά με χρόνους π.χ. για 2'' ο βομβητής και σβηστό το LED και μετά για 3'' να ανάβει το LED αλλά να μην ακούγεται ο βομβητής.

<https://www.tinkercad.com/things/cosVsU4udUE>





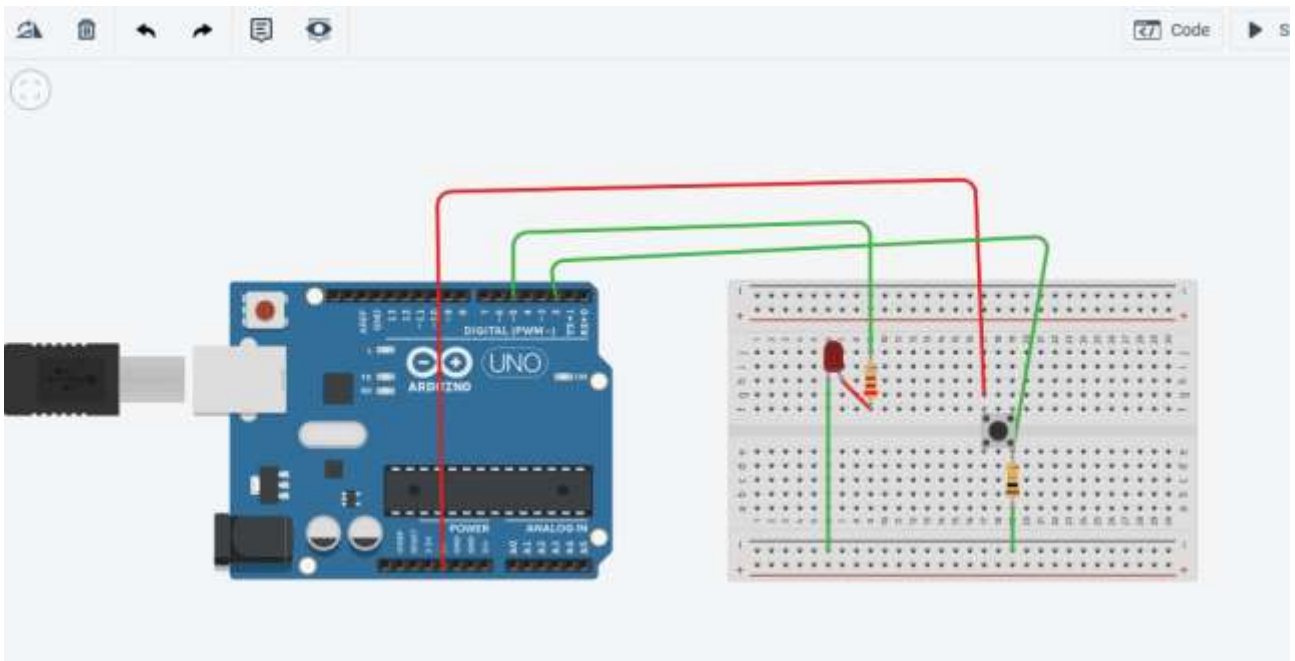
## Κύκλωμα 5ο

Στο 5ο κύκλωμα θα κάνουμε ένα LED να αναβοσβήνει με το πάτημα ενός κουμπιού. Όσο πατάμε το κουμπί το LED θα ανάβει και όταν το αφήνουμε θα σβήνει.

Θα χρησιμοποιηθεί μία εντολή ελέγχου: `if(...) ... else ...`

**Αν**(ισχύει κάποια συνθήκη) θα γίνεται κάτι **αλλιώς** θα γίνεται κάτι άλλο. Μέσα στη συνθήκη ελέγχου θα ελέγχουμε την κατάσταση της ακίδας ΕΙΣΟΔΟΥ (pin 2) και θα ενεργοποιούμε κατάλληλα την ακίδα ΕΞΟΔΟΥ (pin 5).

```
if (digitalRead(2) == HIGH) // Αν ο διακόπτης είναι πατημένος...
{
  digitalWrite(5, HIGH); // Άναψε το LED
}
else {
  // ... αλλιώς ...
  digitalWrite(5, LOW); // Σβήσε το LED
}
```



Το αριστερό τμήμα του κουμπιού συνδέεται στην τροφοδοσία (pin 5 V) και το δεξί στη γείωση (GND) μέσω της αντίστασης 10 kΩ, για αποφυγή βραχυκυκλώματος όταν πατάμε το κουμπί. Για τον έλεγχο της κατάστασης του κουμπιού, συνδέουμε το δεξί του τμήμα στον ψηφιακό ακροδέκτη 2 του Arduino, που θα χρησιμοποιηθεί ως είσοδος. Όταν το κουμπί δεν είναι πατημένο, το κύκλωμα είναι ανοικτό, δεν υπάρχει

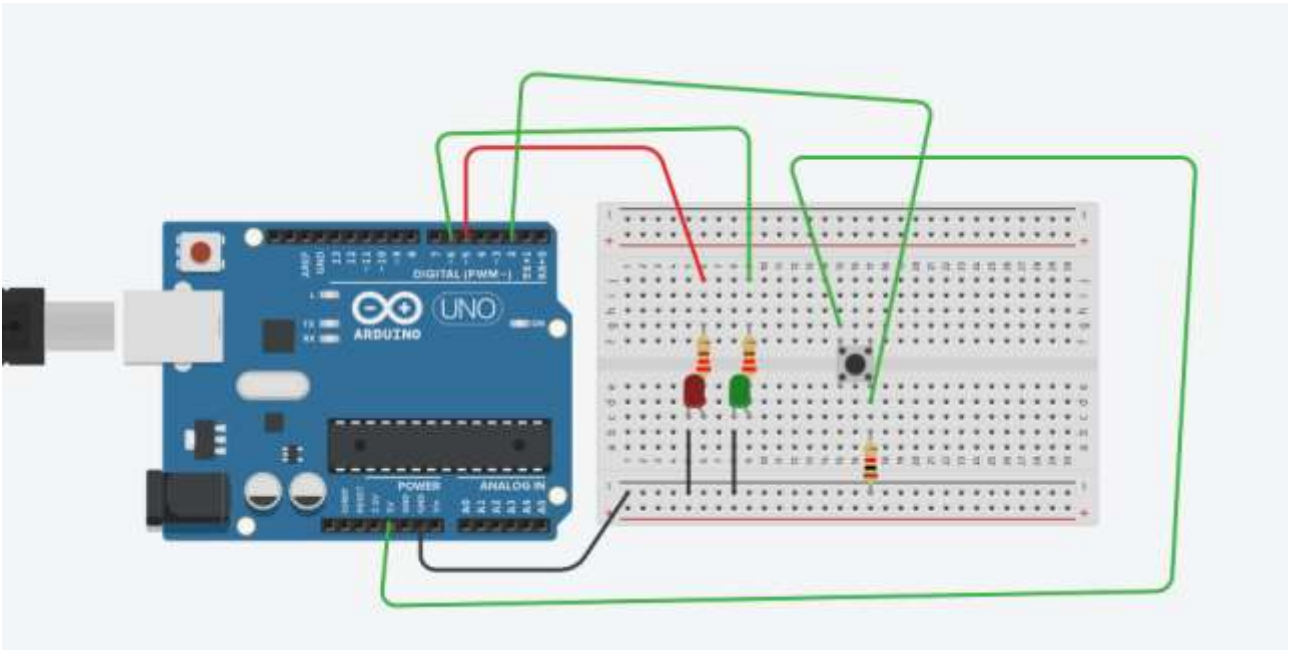
ρεύμα, ούτε τάση στην αντίσταση των 10 kΩ, άρα, ο ακροδέκτης 2 είναι συνδεδεμένος, μέσω της R2, στη γείωση και είναι σε δυναμικό 0V (κατάσταση LOW). Όταν πατηθεί το κουμπί, κλείνει το κύκλωμα και ο ακροδέκτης 2 βρίσκεται συνδεδεμένος σε δυναμικό 5 V, δηλαδή στον θετικό πόλο της τάσης τροφοδοσίας (κατάσταση HIGH).

`digitalRead(... ακίδα);`

Διαβάζει την κατάσταση της ακίδας που αναφέρουμε στην παρένθεση. Π.χ. η εντολή: `a=digitalRead(2);` διαβάζει την κατάσταση της ακίδας 2 και την ελέγχει αν είναι ίση με HIGH. Αν είναι εκτελείται το πρώτο τμήμα της `if`, αν όχι το δεύτερο τμήμα.

## Κύκλωμα 6ο.

Στο 6ο κύκλωμα θα γίνει ένας συνδυασμός του κυκλώματος 3 και 4. Δηλαδή θα υπάρχει ένας διακόπτης που θα ανάβει το κόκκινο LED, ενώ το πράσινο LED θα αναβοσβήνει μόνο του.

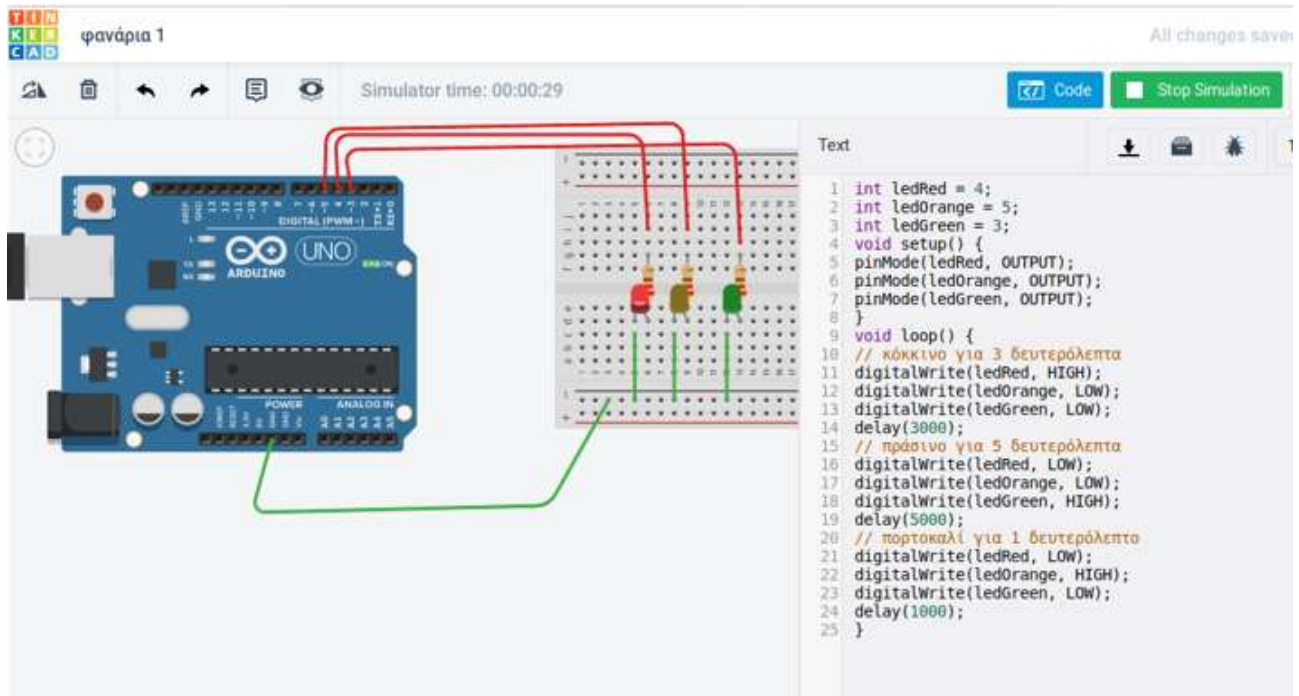


Παρατηρήστε ότι το κόκκινο λαμπάκι δεν ανάβει αμέσως μόλις πατηθεί το button αλλά μόνο όταν το πράσινο λαμπάκι σβήνει μετά το delay(1000) γιατί οι εντολές εκτελούνται σε σειρά. Επίσης δεν σβήνει αμέσως μόλις το αφήσουμε αλλά μόλις σβήσει το πράσινο λαμπάκι.

Η δυσλειτουργία που παρατηρείται οφείλεται στο γεγονός ότι η κατάσταση του κουμπιού ελέγχεται στο πρόγραμμα μόνο μία στιγμή στην αρχή (εντολή if...else...), ενώ στα 3 δευτερόλεπτα που διαρκεί το αναβόσβημα του δεύτερου LED, το κουμπί δεν ελέγχεται και άρα οποιαδήποτε αλλαγή στην κατάστασή του αγνοείται. Για να μπορέσει λοιπόν το κουμπί να λειτουργήσει σωστά, θα έπρεπε η κατάστασή του να ελέγχεται συνέχεια κατά τη διάρκεια των δύο χρονοκαθυστερήσεων (εντολές delay). Αυτό δε μπορεί να γίνει αν χρησιμοποιήσουμε τις “delay” γιατί πολύ απλά το πρόγραμμα ΣΤΑΜΑΤΑΕΙ όταν εκτελείται μια εντολή delay.

## Κύκλωμα 7ο

Στο 7ο κύκλωμα θα κατασκευαστεί ένα φανάρι κυκλοφορίας για αυτοκίνητα μόνο

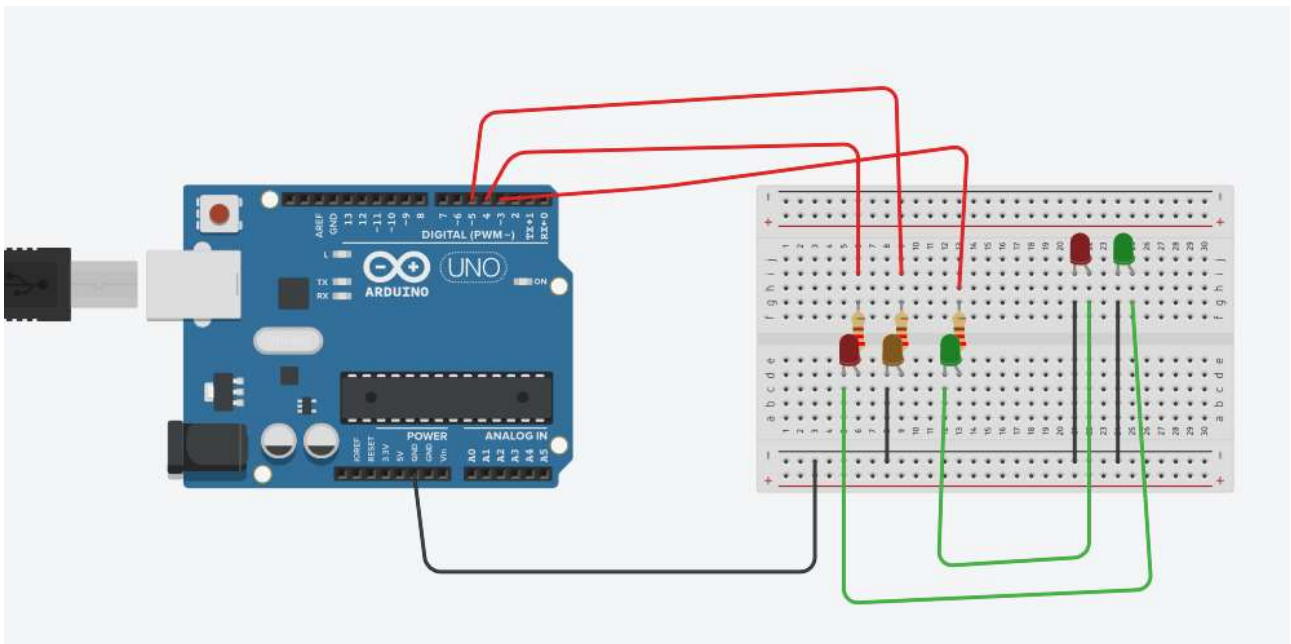


int ...; Από το integer, ορίζει μια ακέραιη μεταβλητή πχ ledRed=4;

Σε κάποιες χώρες μετά το κόκκινο ανάβει πορτοκαλί πριν ανάψει το πράσινο. Τι αλλαγές θα κάνατε στο πρόγραμμα ώστε να γίνει αυτό;

Μπορεί να γίνουν **τροποποιήσεις** ώστε το φανάρι πέρα από τα αυτοκίνητα θα έχει και για πεζούς. Αυτό προτείνεται να γίνει με πειραματισμό. Παρακάτω δίνουμε μια απλή εκδοχή του.

<https://www.tinkercad.com/things/9af1LLjCdw1w>



Και ο κώδικας μένει ο ίδιος, απλά έγιναν αλλαγές στη συνδεσμολογία, ώστε όταν ανάβει το κόκκινο για τα αυτοκίνητα ανάβει ταυτόχρονα το πράσινο για τους πεζούς συνδέοντας σε σειρά μετά το κόκκινο των αυτοκινήτων το πράσινο των πεζών. Το αντίστοιχο γίνεται με το πράσινο των αυτοκινήτων και το κόκκινο των πεζών.

Μια απλή αναζήτηση θα μας δείξει πιο προχωρημένες υλοποιήσεις της εφαρμογής των φαναριών κυκλοφορίας. Οι λέξεις κλειδιά που χρησιμοποίησαν οι μαθητές για να εντοπίσουν και να «τρέξουν» τις εφαρμογές ήταν *arduino traffic light code*. Κάποιες ομάδες προχώρησαν παρακάτω και έβαλαν και κουμπί για τους πεζούς *arduino traffic light code with button*. Αυτές οι υλοποιήσεις έγιναν εκτός σχολικού ωραρίου και σε περίοδο διακοπών γιατί ήθελα να διερευνήσω αν οι μαθητές μπορούν να αναζητήσουν και να πειραματιστούν με κώδικα και με κατασκευές χωρίς την διαθεσιμότητα του εκπαιδευτικού μετά από την έως τώρα εμπειρία τους. Σημαντικό ότι τους είπα να δουν τις συζητήσεις πάνω στον κώδικα και τα σχόλια και να κάνουν ερωτήσεις και εκείνοι, είτε μεταξύ τους είτε στο forum. Με αυτό τον τρόπο ήθελα να κάνουν ένα βήμα προς την ανεξαρτητοποίηση τους. Τα αποτελέσματα ήταν από πολύ καλά σε κάποιες ομάδες μέχρι απογοητευτικά σε άλλες. Από συζητήσεις που έκανα μαζί τους συμπέρανα ότι σε κάποιους ταιριάζει αυτός ο τρόπος εργασίας και διερεύνησης, ενώ σε άλλους είτε δεν ταιριάζει είτε δεν είναι έτοιμοι για αυτόν ακόμη.

## Κύκλωμα 8ο

Στο κύκλωμα 8 θα έχουμε ένα LED που θα αυξάνει την ένταση του ρεύματος που το διαρρέει.

<https://www.tinkercad.com/things/d1eL4bfaW4j>

Με την τεχνική PWM (Pulse with Modulation) οι στάθμες 0V και 5V εναλλάσσονται περιοδικά. Σε κάθε κύκλο του παλμού, το σήμα παίρνει τις τιμές 5V (θετικό μέτωπο) και 0V. Ο χρόνος που το σήμα παραμένει στην τιμή 5V, σε σχέση με τη συνολική διάρκεια του κύκλου, καθορίζει τη μέση τιμή της τάσης. Όσο περισσότερο διαρκεί η στάθμη 5V, τόσο μεγαλύτερη ή μέση τιμή τάσης. Προφανώς, η μέγιστη μέση τιμή τάσης είναι τα 5V και προκύπτει όταν το σήμα παραμένει HIGH καθ' όλη τη διάρκεια του κύκλου. Με την κατάλληλη εντολή μπορεί να μεταβάλλουμε τη διάρκεια του θετικού μετώπου του παλμού.

```
analogWrite(... ψηφιακή ακίδα, ... τιμή από 0 ως 255);
```

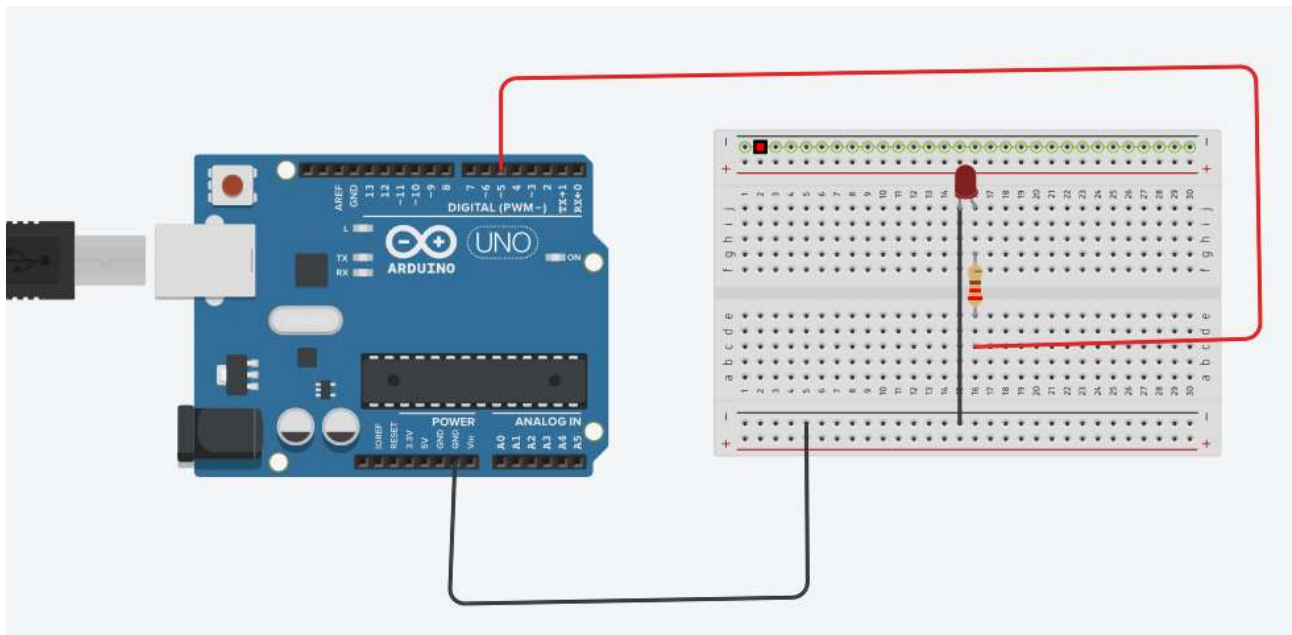
Αυτή η εντολή παράγει συνεχή παλμοσειρά PWM (Pulse Width Modulation) με μεταβαλλόμενο λόγο χρόνων On και Off. Αυτό κάνει το LED να αναβοσβήνει αλλά στα μάτια μας φαίνεται σαν μια ενδιάμεση τιμή φωτεινότητας ανάλογα με το πόσο χρόνο είναι αναμμένο και πόσο χρόνο είναι σβηστό. Η συχνότητα των παλμών είναι περίπου 490 Hz και είναι σταθερή. Αυτό που μεταβάλλεται είναι ο χρόνος που είναι On και ο χρόνος που είναι Off κάθε παλμός. Τιμή 0 θα κάνει τους παλμούς να είναι μηδενικής χρονικής διάρκειας στο On (άρα μόνιμα Off) άρα το LED θα φαίνεται σβηστό, ενώ τιμή 255 θα κάνει τους παλμούς να είναι συνέχεια On (και καθόλου Off) άρα το LED θα φαίνεται αναμμένο πλήρως.

Ενδιάμεσες τιμές δίνουν αίσθηση ενδιάμεσης φωτεινότητας.

- 

```
for (brightness=0; brightness<255; brightness++){  
.....  
.....  
}
```





Αυτή η εντολή είναι **δομή επανάληψης**. Μέσα στην παρένθεση δίνουμε μια τιμή στη μεταβλητή brightness, μετά γράφουμε υπο ποια συνθήκη θα συνεχίζει να κάνει επανάληψη (σε ότι εντολές υπάρχουν ανάμεσα στα άγκιστρα), και τέλος αυξάνουμε τη μεταβλητή brightness κατά 1 με την "brightness++"

int brightness; // Όρισε ακέραια μεταβλητή με όνομα brightness (φωτεινότητα του LED)

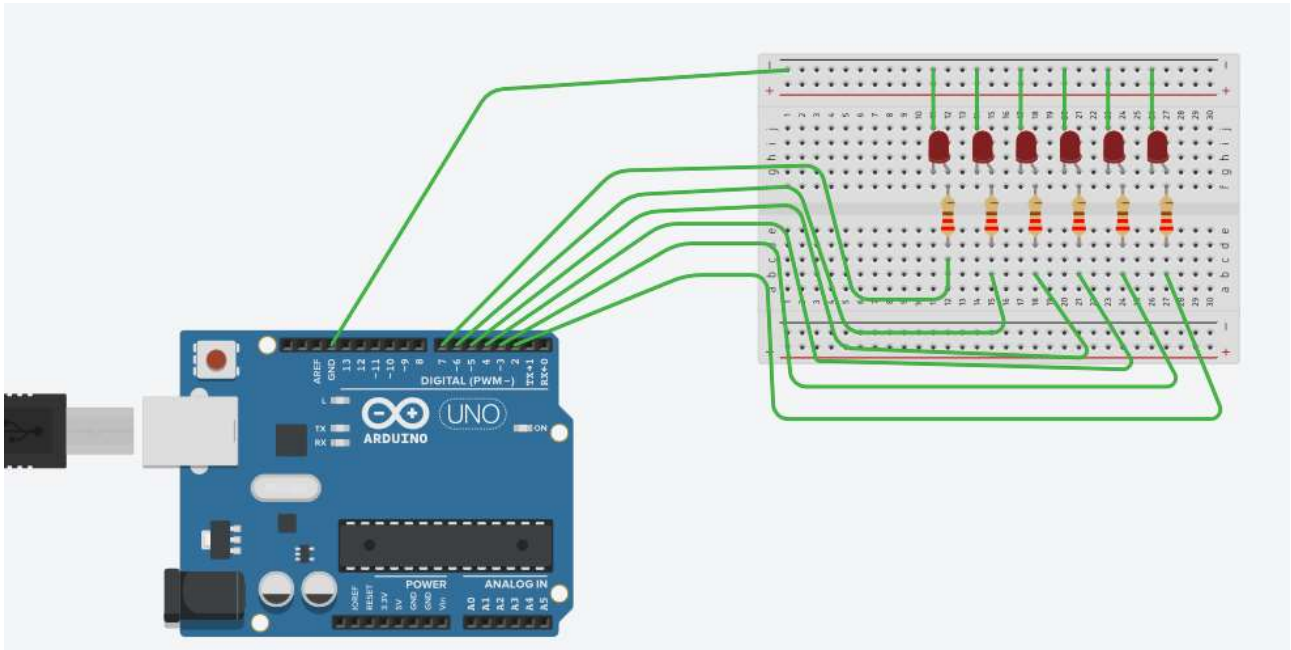
```
1 int brightness; // Όρισε ακέραια μεταβλητή με όνομα brightness (φωτεινότητα του LED)
2 void setup()
3 {
4   pinMode(5, OUTPUT);
5 }
6
7 void loop() {
8   for (brightness=0; brightness<255; brightness++) {
9     analogWrite(5, brightness);
10    //delay(10); // περίμενε εδώ για 10ms
11  } // (τέλος της for)
12  for (brightness=255; brightness>0; brightness--) {
13    analogWrite(5, brightness);
14    //delay(10); // περίμενε εδώ για 10ms
15  } // (τέλος της for)
16 } // (τέλος της loop)
```



## Κύκλωμα 9ο

Στο κύκλωμα 9 υλοποιείται ακόμα ένα παράδειγμα αξιοποίησης της `for` ώστε να βεβαιωθούμε ότι οι μαθητές κατανόησαν

<https://www.tinkercad.com/things/ljY6AYL1G2P>



Υπάρχουν 6 LED που τροφοδοτούνται από τα pin 2 έως 7. Θα χρησιμοποιηθεί μια `for` για να ανάβει και αφού το αφήσει αναμμένο για όσο χρόνο “λέει” ο timer το σβήνει και συνεχίζει διαδοχικά στο επόμενο LED από το χαμηλότερο (2) προς το υψηλότερο (7)

```
for (int thisPin = 2; thisPin < 8; thisPin ++) {  
  // ανάψε το pin  
  digitalWrite(thisPin, HIGH);  
  delay(timer);  
  // σβήσε το pin  
  digitalWrite(thisPin, LOW);  
}
```

Αφού γίνει αυτό ανάβει τα LED ανάποδα από το μεγαλύτερο (7) στο μικρότερο (2). Με αυτό τον τρόπο όμως το 7 θα μείνει το διπλάσιο χρόνο αναμμένο. Αν ζητήσουμε από τους μαθητές να δούν αν υπάρχει κάποια “παραφωνία” οι μαθητές το εντοπίζουν και στο “ανάποδο ανάμμα” αντί για 7 μέχρι 2 γράφουν από 6 μέχρι 3 ώστε τα ακριανά να μην ανάβουν για διπλάσιο χρόνο.

```
for (int thisPin = 6; thisPin >= 3; thisPin --) {
```

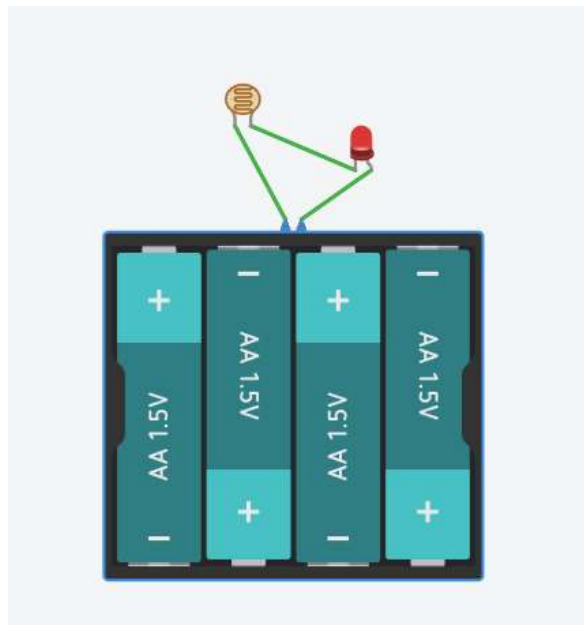
```
// άναψε pin
digitalWrite(thisPin, HIGH);
delay(timer);
// σβήσε το pin
digitalWrite(thisPin, LOW);
}
```

```
1  int timer = 200; // The higher the number, the slower the timing.
2
3  void setup() {
4      // use a for loop to initialize each pin as an output:
5      for (int thisPin = 2; thisPin < 8; thisPin++) {
6          pinMode(thisPin, OUTPUT);
7      }
8  }
9
10 void loop() {
11     // loop from the lowest pin to the highest:
12     for (int thisPin = 2; thisPin < 8; thisPin ++) {
13         // turn the pin on:
14         digitalWrite(thisPin, HIGH);
15         delay(timer);
16         // turn the pin off:
17         digitalWrite(thisPin, LOW);
18     }
19
20     // loop from the highest pin to the lowest:
21     for (int thisPin = 6; thisPin >= 3; thisPin --) {
22         // turn the pin on:
23         digitalWrite(thisPin, HIGH);
24         delay(timer);
25         // turn the pin off:
26         digitalWrite(thisPin, LOW);
27     }
28 }
```

Όπως βλέπετε στον κώδικα χρησιμοποιείται ακόμα μια δομή επανάληψης στην αρχή ώστε να αρχικοποιήσουμε όλα τα pin ως έξοδοι.

## Κύκλωμα 10ο

Στο 10ο κύκλωμα θα χρησιμοποιήσουμε μια φωτοαντίσταση. Η φωτοαντίσταση είναι μια μεταβλητή αντίσταση η τιμή της οποίας αλλάζει ανάλογα με το φως που πέφτει πάνω της. Η φωτοαντίσταση χρησιμοποιείται σε κυκλώματα που θέλουν να αναγνωρίσουν αν το περιβάλλον είναι φωτεινό ή σκοτεινό. Στην πράξη το συναντάμε στις λάμπες που ανάβουν αυτόματα σε χαμηλό φωτισμό



Στο κύκλωμα αυτό μπορούμε να μεταβάλλουμε τη φωτεινότητα του LED (χρησιμοποιώντας τη μπάρα που βρίσκεται πάνω από τη φωτοαντίσταση). Η φωτεινότητα του LED κανονικά αλλάζει ανάλογα με το φως που δέχεται. Η τιμή της αντίστασης μικραίνει όταν στον χώρο υπάρχει πολύς φωτισμός, με αποτέλεσμα το LED να είναι πολύ φωτεινό, αφού είναι συνδεδεμένα σε σειρά. Αντίθετα, η τιμή της αντίστασης μεγαλώνει όσο μικραίνει ο φωτισμός, με αποτέλεσμα το LED να σβήνει στο σκοτάδι.

Στο παρακάτω κύκλωμα θα χρησιμοποιηθεί η φωτοαντίσταση για να ανάψουμε το LED στο σκοτάδι. Ουσιαστικά θα χρησιμοποιήσουμε το Arduino για να "αντιστρέψουμε" την πραγματική λειτουργία της φωτοαντίστασης ώστε να ανάβει το LED στο σκοτάδι

Ο φωτοαντιστάτης επιστρέφει μια τιμή ανάλογα με το φως που φτάνει σε αυτόν. Επίσης χρησιμοποιούμε μια τιμή-όριο πάνω από την οποία θα ανάβει το LED. Η τιμή 25 είναι ενδεικτική και μπορεί να αποτελέσει αντικείμενο πειραματισμού. Με τον τρόπο αυτό εισάγεται και η μεταβλητή και η αξιοποίηση της.

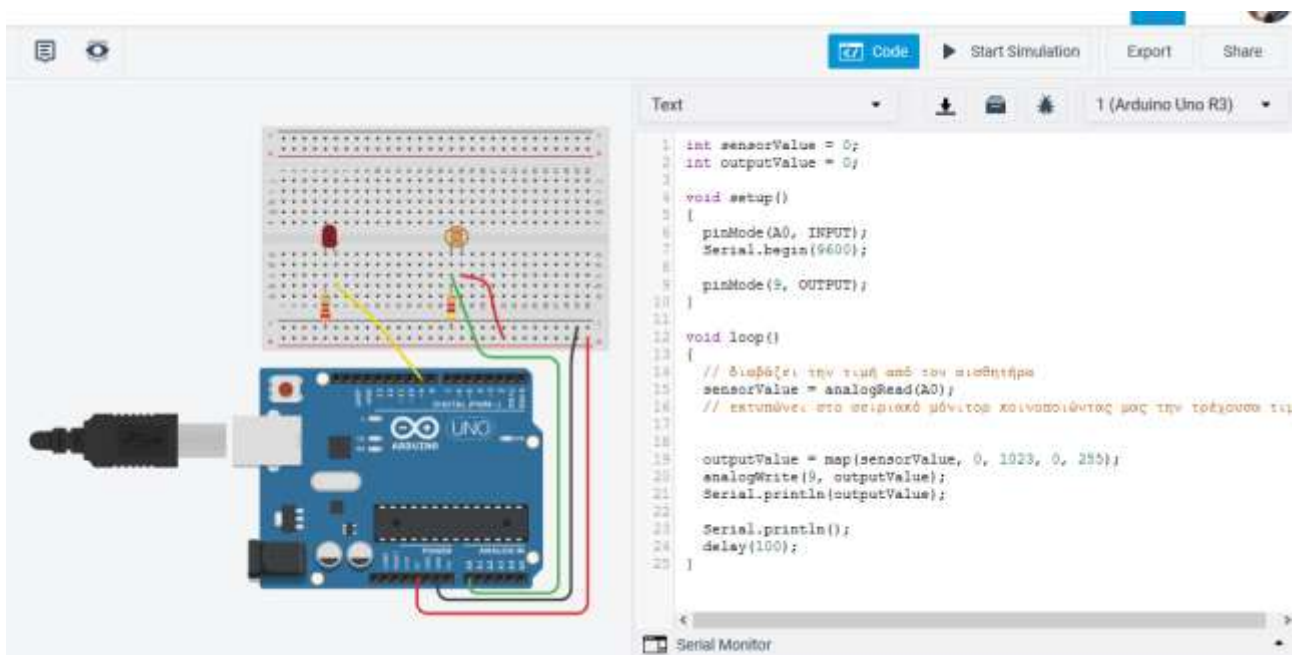
The image shows a simulation of an Arduino Uno R3 board connected to a breadboard circuit. The breadboard contains a resistor and an LED. The circuit is connected to the Arduino's power and ground pins. The code editor on the right contains the following C++ code:

```
1 //σταθερές
2 const int pResistor = A0; //φωτοαντίσταση στην αναλογική A0
3
4 //μεταβλητές
5 int value; // αποθηκεύεται η τιμή φωτοαντίστασης(0-1023)
6
7 void setup(){
8   pinMode(9, OUTPUT); // 9 pin ως output
9   pinMode(pResistor, INPUT); // pResistor - A0 pin ως input
10 }
11
12 void loop(){
13   value = analogRead(pResistor);
14
15   //Αν η value γίνει μεγαλύτερη από "25"
16   if (value > 25){
17     digitalWrite(9, LOW); //σβήνει το LED
18   }
19   else{
20     digitalWrite(9, HIGH); //ανάβει το LED
21   }
22   delay(500); //καθυστέρηση
23 }
24 }
```

## Κύκλωμα 11ο

Στο κύκλωμα 11 συνεχίζουμε με τη φωτοαντίσταση αλλά εισάγουμε και τη χρήση του σειριακού μόνιτορ.

Η συνδεσμολογία του παρακάτω σχεδίου είναι μια νέα καλή πρακτική αφού δίνει 5V στη φωτοαντίσταση και στο LED ταυτόχρονα μέσω της κόκκινης γραμμής (+) του breadboard. Ενώ στη μαύρη γραμμή (-) συνδέονται ο αρνητικός πόλος του LED μέσω αντίστασης αλλά και της φωτοδιόδου.



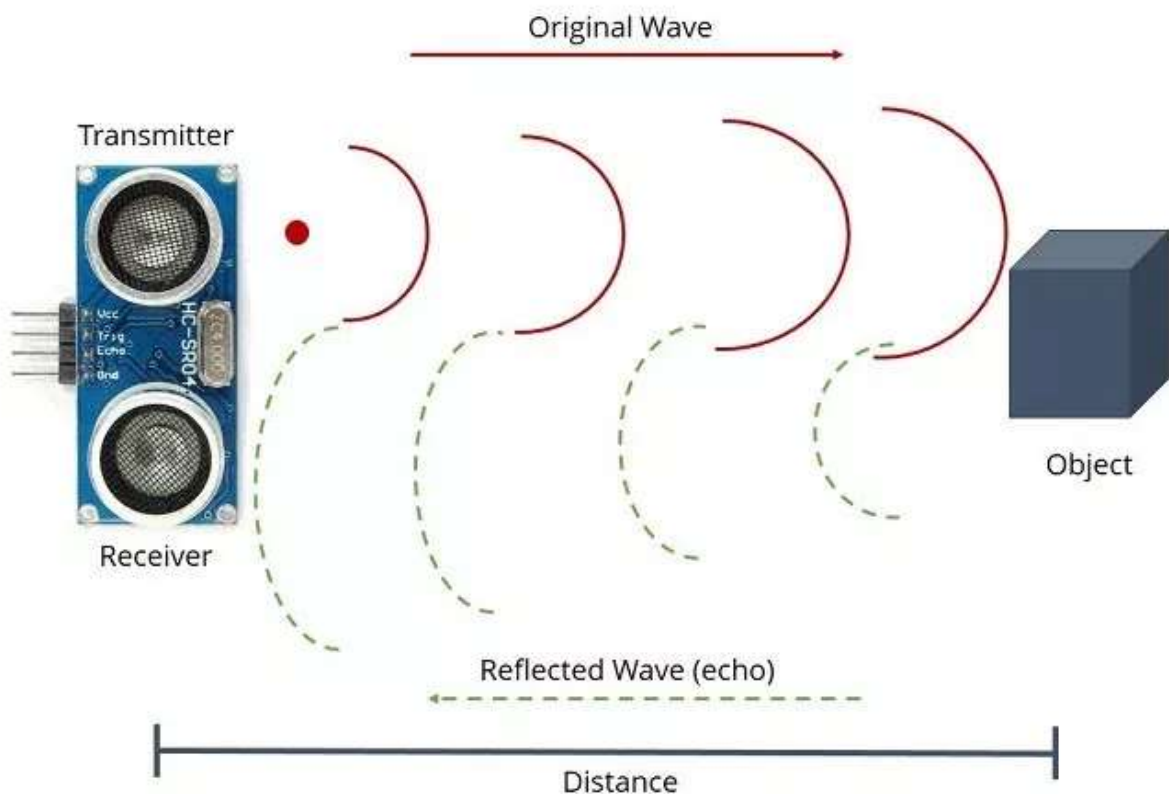
Μια άλλη νέα λειτουργία που υπάρχει σε αυτή την εφαρμογή είναι η Σειριακή οθόνη (Serial Monitor). Σε αυτή εμφανίζονται τα σειριακά δεδομένα που στέλνονται από το Arduino. Η αποστολή δεδομένων στο Arduino γίνεται με τη ρύθμιση της ταχύτητας (baud rate) `Serial.begin(9600)`. Ορίζει την ταχύτητα δεδομένων σε bits ανά δευτερόλεπτο (baud) για τη μετάδοση σειριακών δεδομένων για παράδειγμα `Serial.begin(9600); // ανοίγει την σειριακή θύρα και ορίζει το ρυθμό σε 9600 bps (bits per second)`

Μια άλλη εντολή είναι η `map` που πραγματοποιεί γραμμικό μετασχηματισμό ορίων. Πρακτικά είναι μια συνάρτηση αντιστοίχισης (μετατροπής) των τιμών της αναλογικής τάσης από 0 – 1023 σε εύρος τιμών από 0-255 και απόδοση των νέων τιμών `analogWrite(9, map(sensorValue, 0, 1023, 0, 255))`. Εκτυπώνει στο σειριακό μόνιτορ κοινοποιώντας την τρέχουσα τιμή.

## Κύκλωμα 12ο

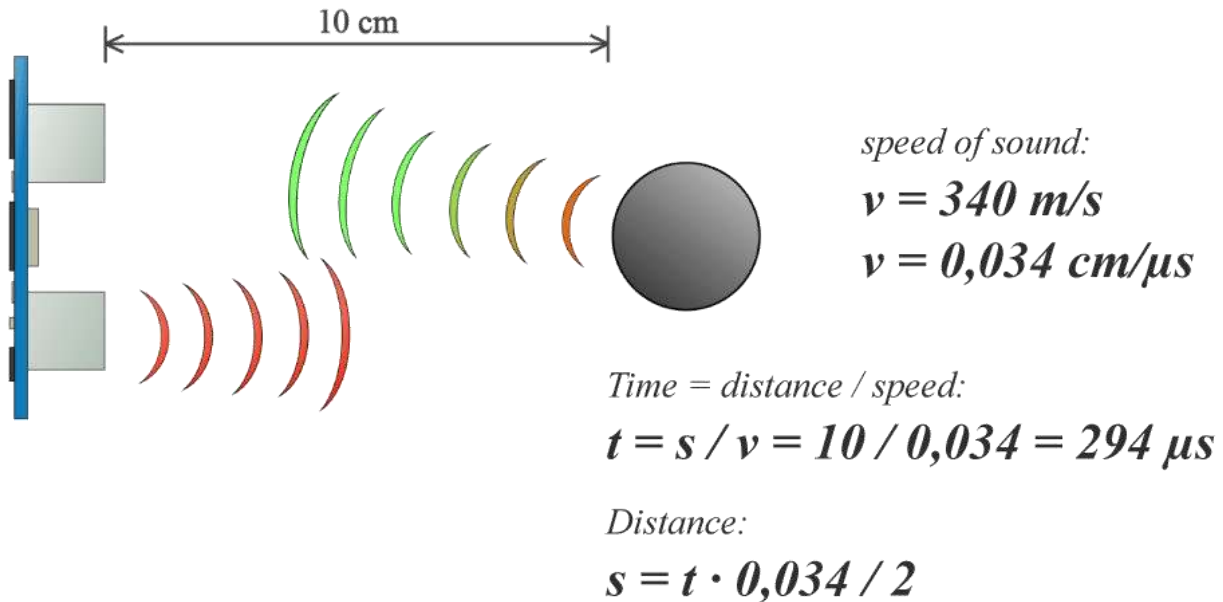
Στο κύκλωμα 12 θα χρησιμοποιηθεί ο αισθητήρας υπερήχων Ultrasonic Sensor HC-SR04. Ο αισθητήρας αυτός χρησιμοποιεί σόναρ για τον προσδιορισμό της απόστασης από ένα αντικείμενο όπως οι νυχτερίδες. Προσφέρει πολύ καλή ανίχνευση απόστασης, χωρίς επαφή, με υψηλή ακρίβεια και σταθερές ενδείξεις, ενώ η χρήση του είναι απλή. Περιέχει τον πομπό και τον δέκτη υπερήχων. Με τη χρήση του μπορεί να μετρηθούν αποστάσεις από 2 εκατοστά μέχρι 4 μέτρα με σχετικά μεγάλη ακρίβεια, ιδανικά τεσσάρων χιλιοστών. Ο αισθητήρας εκπέμπει ένα υπέρηχο των 40000Hz στον αέρα και αν υπάρχει κάποιο αντικείμενο ή εμπόδιο μπροστά του επιστρέφει πίσω στον αισθητήρα.

Ο αισθητήρας που χρησιμοποιήσαμε είναι αυτός με τα 4 pin και δεν υπάρχει στα βασικά (basic) εργαλεία (components), αλλά βρίσκεται στα συνολικά (all). Αυτή η



επιλογή έγινε για να είναι σε αντιστοιχία με τους φυσικούς αισθητήρες υπέρυθρων που διαθέτουμε στο εργαστήριο. Επειδή αυτό το κύκλωμα είναι ενδιαφέρον να υλοποιηθεί και φυσικά ώστε να καταλάβουν οι μαθητές πως είναι ένα πραγματικό project με arduino.

Ο πομπός (transmitter) μέσω του trig pin στέλνει έναν ήχο υψηλής συχνότητας. Όταν το σήμα βρίσκει ένα αντικείμενο, αντανακλάται και επιστρέφει στον υποδοχέα (receiver) μέσω του echo pin.

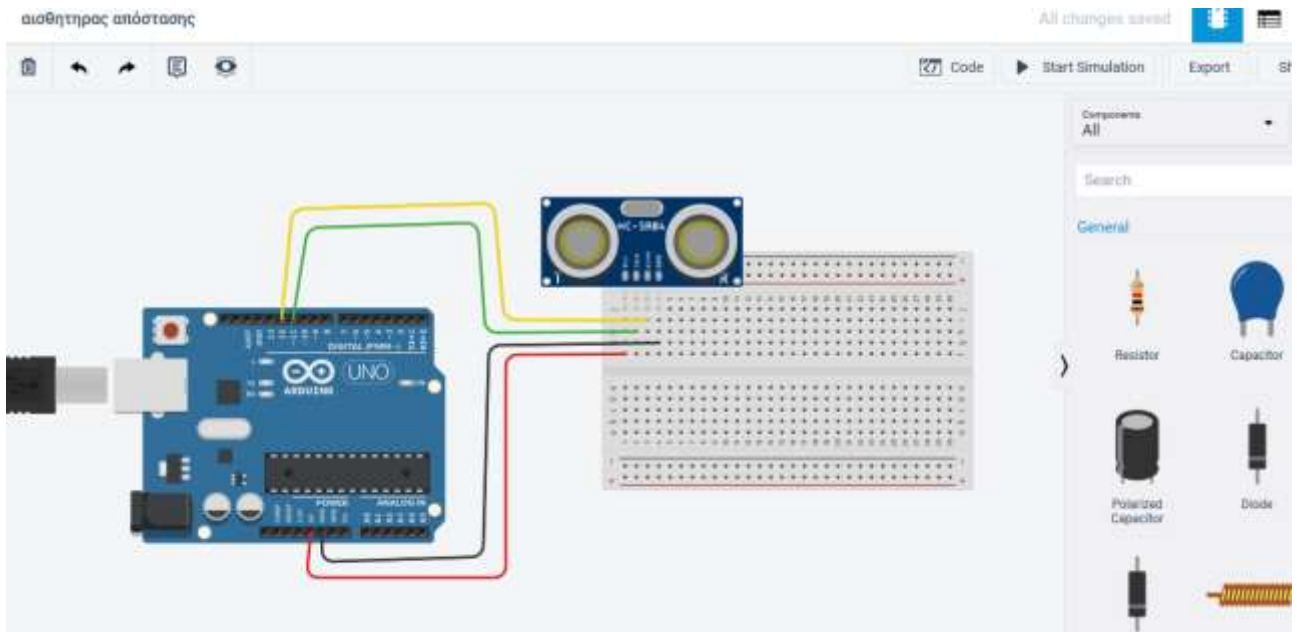


Ο χρόνος (time) που μεσολαβεί ανάμεσα στη μετάδοση και τη λήψη του σήματος επιτρέπει τον υπολογισμό της απόστασης (distance) από ένα αντικείμενο, επειδή γνωρίζουμε την ταχύτητα του ήχου στον αέρα (speed of sound). Γνωρίζοντας τη διάρκεια επιστροφής του υπέρηχου και την ταχύτητα του ήχου, μπορούμε να υπολογιστεί η απόσταση του αντικειμένου από τον αισθητήρα.

Στις υποδοχές συνδέουμε στο VCC: +5V DC, στο Trig από το Trigger την είσοδο (INPUT), στο Echo την έξοδο (OUTPUT) και στο GND την γείωση (Ground).

<https://www.tinkercad.com/things/fsibYES5Rp3>





Ο αισθητήρας διαθέτει 4 υποδοχές. Την γείωση (Ground) που αυτονόητα συνδέεται στο GND του Arduino, το VCC στην υποδοχή 5 Volts του Arduino. Η υποδοχή Trig και Echo σε οποιαδήποτε ψηφιακή υποδοχή (είτε εισαγωγή, είτε εξαγωγή), στην κάρτα Arduino, ανάλογα με τη χρήση.

Για να παραχθεί ο υπέρηχος, χρειάζεται να δοθεί στο Trig ρεύμα για 10 microSeconds ( $\mu\text{S}$ ), τότε στέλνονται υπέρηχοι που ταξιδεύουν με την ταχύτητα του ήχου και επιστρέφουν στην υποδοχή Echo. Η υποδοχή Echo θα δίνει τον χρόνο που έκανε να ταξιδεύσει ο ήχος σε microseconds ( $\mu\text{S}$ ).

Η καταγραφή που θα πάρουμε από την υποδοχή Echo είναι η διπλάσια λόγω του ότι ο υπέρηχος κάνει διπλάσιο ταξίδι (φεύγει από τον αισθητήρα προς το αντικείμενο και επιστρέφει από το αντικείμενο προς τον αισθητήρα).

Οι μεταβλητές trigPin και echoPin συνδέονται στο ψηφιακό Pin 11 και το echo συνδέεται στο ψηφιακό Pin 12.

Χρησιμοποιούνται 3 ακόμα μεταβλητές duration (διάρκεια), cm (εκατοστά) και inch (ίντσες). Στη μεταβλητή duration αποθηκεύεται ο χρόνος ανάμεσα στην εκπομπή και τη λήψη του σήματος. Στη μεταβλητή cm θα αποθηκευτεί η απόσταση σε εκατοστά και στη μεταβλητή inch θα αποθηκευθεί η απόσταση σε inches.

Η απόσταση (distance) = (χρόνος/2) x ταχύτητα του ήχου

Η ταχύτητα του ήχου είναι: 343m/s

Ο χρόνος διαιρείται με 2 γιατί μέσα σε αυτό το χρόνο το σήμα φεύγει από τον αισθητήρα και επιστρέφει σε αυτόν, άρα κάνει την συγκεκριμένη απόσταση 2 φορές.

```
int trigPin = 11; // Trigger
int echoPin = 12; // Echo
long duration, cm, inches;

void setup() {
  //ξεινάει την σειριακή έξοδο serial port
  Serial.begin (9600);
  //ορίζει εισόδους inputs και εξόδους outputs
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  //δίνουμε ρεύμα για 10μsec-στέλνουμε σήμα
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  duration = pulseIn(echoPin, HIGH);

  // μετατροπή του χρόνου σε απόσταση
  cm = (duration/2) / 29.1; // Divide by 29.1 or multiply by 0.0343
  inches = (duration/2) / 74; // Divide by 74 or multiply by 0.0135

  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();

  delay(250);
}
```

Με αυτές τις ασκήσεις ολοκληρώνονται 12 κυκλώματα με τα οποία καλύπτονται οι οδηγίες του ΙΕΠ που φτάνουν κάθε χρόνο στα σχολεία για το μάθημα της Πληροφορικής.

## **Ρομποτική**

Ωστόσο πέρα από τους βασικούς αισθητήρες είναι σημαντικό να δοθεί ένας «αέρας» ρομποτικής και η ρομποτική χρειάζεται κινητήρες. Οι κινητήρες (motors) που υποστηρίζει το Arduino χωρίζονται σε 2 κατηγορίες: servo motor (σερβοκινητήρας) και stepper motor (βηματικός κινητήρας).

Ο σερβοκινητήρας διαθέτει σύστημα από γρανάζια (integrated gears) και έναν άξονα (shaft) που μπορεί να ελεγχθεί με ακρίβεια. Οι Servo συνδέονται με 3 καλώδια απ' ευθείας στο Arduino και κινούνται από 0 μέχρι 180 μοίρες. Ελέγχονται δίνοντας τη γωνία και μπορεί να ρυθμιστεί η ταχύτητα περιστροφής που θα φτάσει τη επιθυμητή γωνία δίνοντας το χρόνο που θέλουμε να γίνει αυτό.

Αντίστροφα μπορεί να «διαβαστεί» η τιμή της γωνίας που στρίβει και αυτό μπορεί να είναι πολύ χρήσιμο έως απαραίτητο σε κάποιες εφαρμογές.

Ο βηματικός κινητήρας (stepper motor) έχει ως χαρακτηριστικό το ότι δεν κινείται ανεξέλεγκτα αλλά σε διακριτές θέσεις και η μονάδα μέτρησης των θέσεων είναι τα βήματα. Οι κινητήρες αυτοί χρησιμοποιούνται στα ρομπότ.

Περισσότερα για τη λειτουργία του βηματικού κινητήρα μπορείτε να βρείτε στη <https://www.getcert.gr/pos-leitourgei-to-stepper-motor/>

## Εργασία για το σπίτι

Μετά το τέλος των μαθημάτων μπορεί ο εκπαιδευτικός να προτείνει στους μαθητές να υλοποιήσουν μια εργασία στο σπίτι. Η εργασία αυτή είναι καλύτερο να γίνει σε ομάδες, αλλά αν υπάρχει πρόβλημα μπορεί να είναι και ατομική.

Η εργασία αυτή είναι αρκετά καθοδηγούμενη και μπορεί να θεωρηθεί ως ένα βήμα φθίνουσας καθοδήγησης που σταδιακά θα εφαρμόζει ο εκπαιδευτικός καθώς οι μαθητές του εξοικειώνονται με τον τρόπο λειτουργίας της εφαρμογής.

Ένα παράδειγμα θα μπορούσε να είναι η μέτρηση της θερμοκρασίας του περιβάλλοντος και όταν αυτή ξεπερνάει ένα όριο πχ 30C τότε θα ξεκινάει να δουλεύει ένας κινητήρας servo που θα ανοίγει ένα παράθυρο ή ένας ανεμιστήρας θα αρχίσει να λειτουργεί.

Με αυτό τον τρόπο οι μαθητές θα μάθουν μέσα από πηγές που θα βρουν οι ίδιοι πως λειτουργεί:

- ο αισθητήρας θερμοκρασίας -Temperature Sensor TMP (DS18B20 για πραγματικό κύκλωμα)
- ένα Micro Servo για την περίπτωση που θέλουμε να ανοίγει ένα παράθυρο
- ένα stepper motor στην περίπτωση του ανεμιστήρα

Αυτά τα υλικά είναι αξιοποιήσιμα σε πολλά project πχ οι κινητήρες είναι απαραίτητοι σε όλες τις εφαρμογές ρομποτικής <https://create.arduino.cc/projecthub/projects/tags/robot>, ενώ ο αισθητήρας θερμοκρασίας μαζί με άλλους παρόμοιας πρακτικής και λειτουργίας δίνουν πρόσβαση σε εφαρμογές κηπουρικής <https://create.arduino.cc/projecthub/projects/tags/garden>.

Η πρόταση είναι στο δεύτερο τετράμηνο μια φορά κάθε 15 μέρες να βλέπουμε και να συζητάμε την εξέλιξη των project που υλοποιούν οι μαθητές.

ISBN 978-618-00-1983-4

