

## Ρομποτική

Η Ρομποτική είναι η επιστήμη και η τεχνολογία των Ρομπότ, περιλαμβάνοντας τον σχεδιασμό, την κατασκευή και τις εφαρμογές τους.

Το Ρομπότ είναι μία κατασκευή που μπορεί να εκτελεί προγραμματισμένες εργασίες, αλλά σε αντίθεση με τις απλές μηχανές ακολουθεί την μέθοδο «αντιλαμβάνομαι, σκέπτομαι, ενεργώ»



Το Ρομπότ καθοδηγείται από ένα ηλεκτρονικό κύκλωμα και από κάποιο πρόγραμμα υπολογιστή και αποτελείται από δομικά μέρη (δοκοί, γρανάζια, ρόδες, ...), ηλεκτρονικά (μονάδα εγκεφάλου, ...), αισθητήρες (χρώματος, φωτός, κίνησης, απόστασης, ...) και κινητήρες. Συνήθως χρησιμοποιείται για την εκτέλεση επικίνδυνων ή δύσκολων (για ανθρώπους) εργασιών, ή για να αυτοματοποιήσει μονότονες επαναληπτικές εργασίες, που ωστόσο πρέπει να γίνονται με μεγάλη ταχύτητα κι ακρίβεια.

Το Ρομπότ αποτελεί μία ιδανική τέτοια δημιουργία – κατασκευή, με στόχο την ανάπτυξη και εξέλιξη πρακτικών, διδακτικών, γνωσιακών και εφαρμοσμένων δεξιοτήτων από τους χρήστες.

### Οφέλη της Ρομποτικής

- Μαθησιακή «πειθαρχία» και οργάνωση σε πολλά επίπεδα: από το ερευνητικό ενδιαφέρον μέχρι την υλοποίηση.
- Κάλυψη με ιδανικό τρόπο των επιστημών και της τεχνολογίας (STEM)\* με τον συνδυασμό της μάθησης με το «παιχνίδι».
- Ανάπτυξη της ικανότητας να παρουσιάζεται σε κοινό μία δημιουργία καθώς και της περιγραφικής (γραπτής και προφορικής) δεξιότητας, καλλιέργεια της δημιουργικότητας των παιδιών.
- Επέκταση σε πολύτιμες δεξιότητες, όπως επικοινωνία, έρευνα, στρατηγική και αναλυτική σκέψη, διαχείριση χρόνου (project management), κατανομή έργου και πόρων.
- Ενδυνάμωση δεξιοτήτων έρευνας και επίλυσης προβλημάτων.
- Ανάπτυξη ενδιαφερόντων σε εφαρμοσμένες τεχνολογίες.
- Γίνεται συνήθεια η ανάλυση ο σχεδιασμός και η επίτευξη στόχων.

### Η Ιστορία των Ρομπότ, οι πρώτοι μύθοι.

Πολύ πριν να κατασκευαστεί το πρώτο ρομπότ, οι αρχαίοι Έλληνες είχαν οραματιστεί τα Αυτόματα (διεθνώς Automata). Ο Όμηρος περιγράφει δύο Αυτόματα-κοπέλες. Τις είχε φτιάξει ο Ήφαιστος από χρυσό. Είχαν ως εργασία τους να σηκώνουν και να μεταφέρουν τον Ήφαιστο μέσα στο εργαστήριό του στον Όλυμπο. Καθώς φαίνεται στον «προγραμματισμό» τους συνεισέφεραν όλοι οι θεοί: «... και τρέχαν δίπλα του να ανεβαστούν τον ρήγα χρυσές

δυο βάγιες, απaráλλαχτες με ζωντανές κοπέλες, ξυπνάδα και μιλιά και δύναμη, τα ἔχουν κι αυτές, κι ακόμα οι αθάνατοι θεοί τούς έμαθαν πάσα γυναίκεια τέχνη.....». Οι στίχοι αυτοί (417 – 420, Ραψωδία Σ, Ομήρου Ιλιάδα, μτφ. Ν. Καζαντζάκης – Ι.Θ. Κακριδής) σαφώς μιλάνε για αρχαία Αυτόματα. Αναφέρει επίσης ο Όμηρος μεταλλικά λεβέτια (λεκάνες με νερό), τα οποία περιπλανούνταν αυτοκινούμενα ανάμεσα στους θεούς, ώστε να έχουν να πλυθούν. Αυτά ήταν έργα του Ηφαίστου.

Σε όλους επίσης είναι γνωστός ο μύθος του Τάλω, μιας ακόμη κατασκευής του Ηφαίστου. Αυτό, που πρέπει να σημειωθεί για τον Τάλω, είναι οι εξαιρετικές του επιδόσεις (Για αυτό άλλωστε δεν χρειαζόμαστε τα ρομπότ;) και το γεγονός πως είχε όλη κι όλη μία φλέβα από την οποία εξαρτιόταν η λειτουργία του. Η λειτουργία αυτή έπαψε, μόλις η Μήδεια άνοιξε τη φλέβα και ο Τάλως άδειασε από ενέργεια. Ο Τάλως δέχτηκε την επέμβαση αυτή της Μήδειας, διότι του είχε υποσχεθεί πως με αυτόν τον τρόπο θα μεταλλασσόταν από μηχανή σε έμβιο ον.



Μεταλλικές μηχανικές κατασκευές, που δρουν αυτοβούλως, συναντάμε σε μύθους και παραμύθια πολλών λαών σε διάφορες εποχές. Αρκεί να ανατρέξει κανείς στη σχετική βιβλιογραφία.

#### **Από τον Μύθο στην κατασκευή.**

Ήδη λοιπόν από την Ελληνιστική Εποχή αναφέρονται κατασκευές, οι οποίες έχουν αυτοματισμό και μηχανικό προγραμματισμό. Μολαταύτα, η πρώτη πραγματική αρχή των ρομπότ τοποθετείται στο 17<sup>ο</sup> και στον 18<sup>ο</sup> αιώνα. Τότε άρχισαν να κατασκευάζονται τα Αυτόματα. Μα, πριν να πάμε σε αυτά, ας δούμε τι είχαν πετύχει οι αλεξανδρινοί μας πρόγονοι και μετά από αυτούς οι Βυζαντινοί.

Ο αρχαιότερος γνωστός αυτοματισμός είναι, φυσικά, ο Μηχανισμός των Αντικυθήρων. Ο Αρχύτας από τον Τάραντα, που συγκαταλέγεται στους πρώτους οραματιστές του κλάδου, κατασκεύασε (μέσα 4ου π.Χ. αιώνα) μια ιπτάμενη «περιστερά», που ήταν ατμοωθούμενη και η οποία κατάφερε να πετάξει μέχρι και 200 μέτρα, σύμφωνα με τις πηγές.

Ο Ήρωνας ο Αλεξανδρεύς κατασκεύασε και το πρώτο προγραμματιζόμενο Αυτόματο, που ήταν ένα αυτοκινούμενο τρίκυκλο. Στη θεωρία του αυτοματισμού συνεισέφερε με το κορυφαίο σύγγραμμα της αρχαίας μηχανικής, με τον ενδεικτικό τίτλο «Αυτοματοποιητική». Ο Ήρων έζησε το 1<sup>ο</sup> μ.Χ. αιώνα.

Καθώς τελείωνε η Αρχαιότητα και η Ευρώπη έμπαινε στον Μεσαίωνα, δεν ξεχάστηκαν και δε χάθηκαν ούτε τα επιτεύγματα των Ελλήνων, ούτε το όραμα τους αλλά ούτε και η διάθεση των Ρωμαίων (μετέπειτα Βυζαντινών) να επινοήσουν και να κατασκευάσουν Αυτόματα. Έτσι σε όλη την Αυτοκρατορία υπήρχαν διάσπαρτα αυτά τα υπέροχα δείγματα της μηχανικής και του αυτοματισμού.

Το υδραυλικό ρολόι της Γάζας (6ος μ.Χ. αιώνας). Σε αυτό ένας μηχανικός Ηρακλής ήταν προγραμματισμένος να σημαίνει τις ώρες με το ρόπαλό του. Περιώνυμος ήταν ο μηχανικός θρόνος του αυτοκράτορα Θεόφιλου από χρυσό (9ος μ.Χ. αιώνας). Τον είχε κατασκευάσει ο Λέων ο Μαθηματικός, ο μεγαλύτερος ίσως μαθηματικός και μηχανικός του Μεσαίωνα. Ο θρόνος είχε στη βάση του μηχανικά λιοντάρια που «βρυχούνταν» και μηχανικά πουλιά, στα ψεύτικα κλαδιά ενός τεχνητού πλατάνου, και τα οποία «κελαηδούσαν». Το κάθισμα μαζί με τον αυτοκράτορα, με κάποιο σύστημα, ανυψωνόταν πάνω από τα κεφάλια όλων.

Οι Άραβες ως γνωστόν μετέφρασαν και διέδωσαν σε όλον τον αραβικό κόσμο τα συγγράμματα του Αρχύτα του Ταραντίνου και του Ήωνα του Αλεξανδρέα. Πολύ γνωστό είναι πως επιδίδονταν και στην κατασκευή Αυτομάτων. Ο Αλ Γιαζάρι κατασκεύασε το πρώτο ανθρωποειδές Αυτόματο του Μεσαίωνα, έναν τυμπανιστή.

Η εξέλιξη των Αυτομάτων καταδεικνύει τον τρόπο με τον οποίο ο άνθρωπος αντελήφθη και διαμόρφωσε τη διαδικασία με την οποία προγραμματίζεται ένα μηχανισμός, δημιουργώντας την ψευδαίσθηση πραγματικής ανθρώπινης αντίδρασης και κίνησης. Αυτό επιτυγχάνεται με τη συνάρμωση και την κατασκευή συνδέσεων και μηχανικών αλληλουχιών κίνησης, οι οποίες αποτελούνται από απλούς στροφάλους, σχοινιά, γρανάζια, τροχαλίες, πλήκτρα, ελάσματα, κλειδιά και ελατήρια. Αποτέλεσμα είναι ένα λειτουργικό σύνολο – μηχανισμός, ο οποίος καταφέρνει να «χορογραφεί» τα «άκρα» του, τα «μάτια» του, ακόμη και τα «φρύδια» του!

Πέραν όλων των παραπάνω κατασκευάστηκαν και πιο περίπλοκα Αυτόματα, τα οποία μπορούσαν να παίζουν μουσικά όργανα, να σχεδιάζουν, να χειρίζονται τόξο και βέλη, να παίζουν σκάκι. Όχι και τόσο μακρινοί συγγενείς των Αυτομάτων είναι τα κουρδιστά ρολόγια καθώς και οι λατέρνες και οι ρομβίες.

Επίσης, το παλαιότερο σωζόμενο σχέδιο ανθρωποειδούς Αυτομάτου είναι αυτό ενός πολεμιστή με πανοπλία. Τα σχέδια ανήκουν στον πνευματικό γίγαντα της Αναγέννησης Λεονάρντο Ντα Βίντσι (Leonardo Da Vinci). Το συγκεκριμένο Αυτόματο μπορούσε να ανασηκώνεται και να κινεί τα χέρια του και το κεφάλι του με περιορισμένες όμως κινήσεις. Ο Λεονάρντο το σχεδίασε για μάχη, όπως και πολλές από τις μηχανές του

Ο Γάλλος Ζακ Ντε Βωκανσόν ( Jacques De Vaucanson), κατασκεύασε μία πάπια – Αυτόματο. Η πάπια μπορούσε να «τρώνει» σπόρους και να κουνάει τα φτερά της. Στην εποχή της έκανε τεράστια εντύπωση.

Τον 19<sup>ο</sup> αιώνα ο Ιάπωνας Χισασίγκε Τανάκα (Hisashige Tanaka) δημιούργησε ρομποτικούς μηχανισμούς που είχαν την «ικανότητα» γραφής ιαπωνικών ιδεογραμμάτων ή σερβιρίσματος τσαγιού.

Και σιγά – σιγά όλο και πιο πολλές κατασκευές πραγματοποιούνταν και με αυτόν τον τρόπο εξελισσόταν η τεχνογνωσία και η πολυπλοκότητά τους. Στα τέλη του 19ου αιώνα ο Σέρβος Νίκολα Τέσλα (Nicola Tesla) παρουσίασε το πρώτο τηλεχειριζόμενο πλοίο.

## Arduino UNO

Το Arduino είναι μία ολοκληρωμένη πλατφόρμα ανάπτυξης έργων ηλεκτρονικής, αυτοματισμού και ρομποτικής, η οποία περιλαμβάνει το αναγκαίο λογισμικό και υλικό. Το λογισμικό της πλατφόρμας είναι το Arduino IDE. Το υλικό περιλαμβάνει μία σειρά από πλακέτες μικροελεγκτή με διαφορετικά χαρακτηριστικά, ώστε ο χρήστης να μπορεί να επιλέξει την έκδοση που ταιριάζει καλύτερα στην εκάστοτε εφαρμογή του.

Το UNO είναι η βασική έκδοση πλακέτας της πλατφόρμας Arduino. Στον Πίνακα 1 παρατίθενται τα τεχνικά χαρακτηριστικά του.

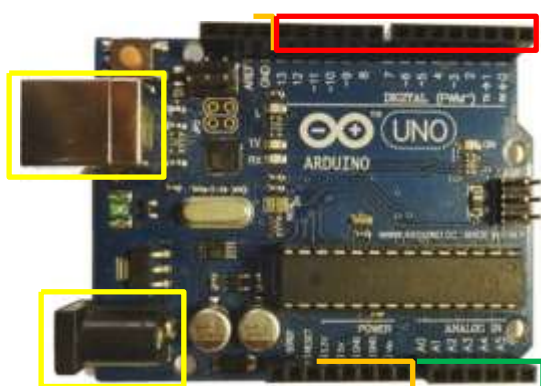
Πίνακας 1

Μικροελεγκτής	ATmega328P
Τάση λειτουργίας	5V
Τάση εξωτερικής τροφοδοσίας	7-12V
Ψηφιακά pin I/O	14 (0-13)
PWM pin	6 (3, 5, 6, 9, 10, 11)
Pin αναλογικής εισόδου	6 (A0-A5)
Ρεύμα ανά I/O pin	20 mA
Μνήμη Flash	32 KB
SRAM	2 KB
EEPROM	1 KB

Συχνότητα ρολογιού	16 MHz
--------------------	--------

Στην Εικόνα 1, στο κόκκινο πλαίσιο περικλείονται οι ψηφιακοί ακροδέκτες εισόδου/εξόδου, στο πράσινο οι ακροδέκτες αναλογικής εισόδου και στα πορτοκαλί οι ακροδέκτες τροφοδοσίας.

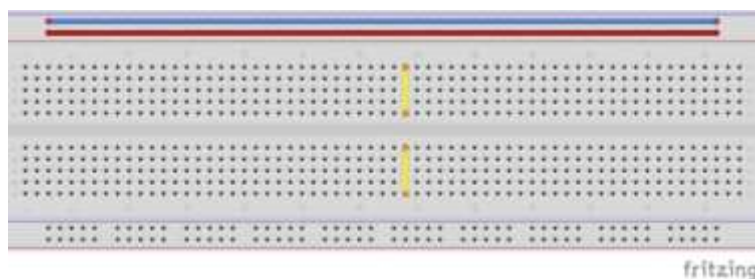
Στα αριστερά της εικόνας μπορούμε να διακρίνουμε στο πάνω μέρος την υποδοχή USB που χρησιμοποιείται για φόρτωση προγράμματος, για σειριακή επικοινωνία με τον υπολογιστή αλλά και για τροφοδοσία. Στο κάτω μέρος υπάρχει η υποδοχή για την εξωτερική τροφοδοσία 7 - 12 V.



Εικόνα 1 Arduino UNO

## Breadboard

Το breadboard είναι μία διάταξη που επιτρέπει την εύκολη κατασκευή κυκλωμάτων χωρίς να απαιτούνται κολλήσεις. Συγκεκριμένα, το breadboard διαθέτει οπές πάνω στις οποίες μπορούν να συνδεθούν διάφορα ηλεκτρικά και ηλεκτρονικά στοιχεία. Εσωτερικά οι οπές αυτές διασυνδέονται μεταξύ τους όπως φαίνεται στην Εικόνα 2. Οι 4 εξωτερικές σειρές (2 πάνω και 2 κάτω) είναι συνδεδεμένες οριζόντια, ενώ στο κεντρικό του τμήμα οι διασυνδέσεις είναι κατακόρυφες, με κάθε στήλη να χωρίζεται σε δύο ανεξάρτητα τμήματα.

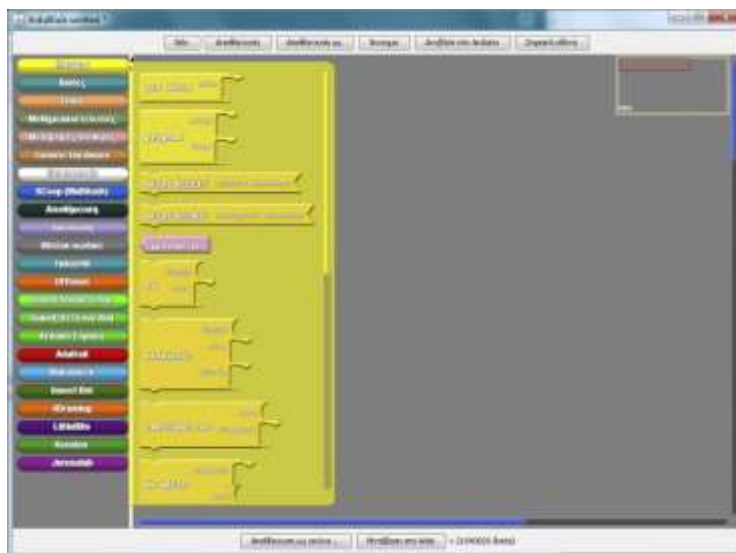


Εικόνα 2 Breadboard

Έτσι, αν θέλουμε για παράδειγμα να ενώσουμε δύο καλώδια, αντί να τα κολλήσουμε με το κολλητήρι, αρκεί να τοποθετήσουμε από ένα άκρο τους σε δύο συνδεδεμένες οπές (π.χ. στην ίδια μισή στήλη). Οι οριζόντιες εξωτερικές σειρές, χρησιμοποιούνται συνήθως σε σύνθετες εφαρμογές, για να παρέχουμε τροφοδοσία και γείωση σε πολλά εξαρτήματα ταυτόχρονα.

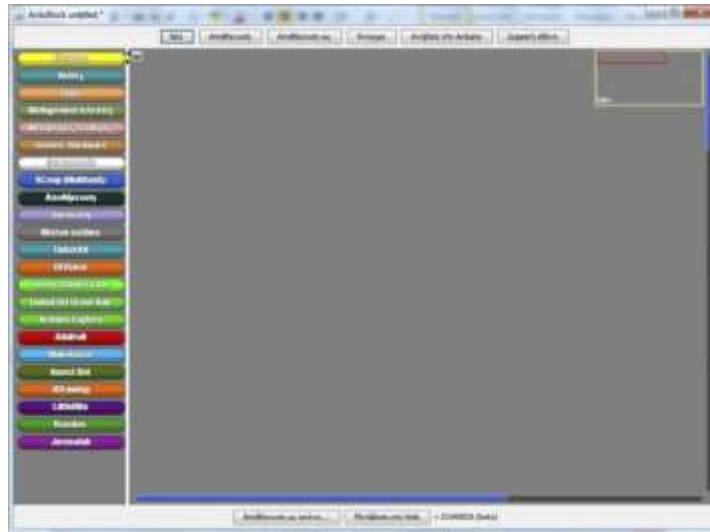
## ArduBlock

Το ArduBlock είναι ένα εργαλείο λογισμικού που εκτελείται μέσα από το Arduino IDE και μας παρέχει μία διεπαφή για τον οπτικό προγραμματισμό του Arduino. Πατώντας την επιλογή *ArduBlock*, μέσα από το μενού *Εργαλεία* του IDE, ανοίγει το παράθυρο που παρουσιάζεται στην Εικόνα 3.



Εικόνα 3

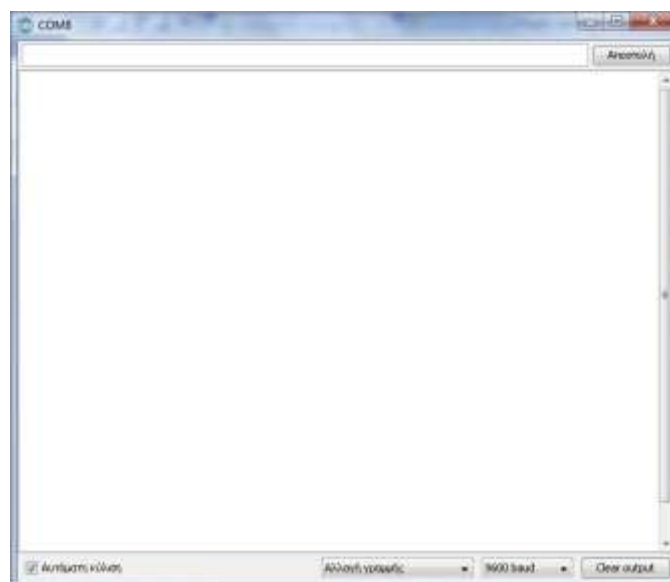
Στο αριστερό μέρος του παραθύρου υπάρχουν κουμπιά με τις κατηγορίες των εντολών. Όταν επιλέγουμε μία κατηγορία αναδύεται η λίστα με τα πλακίδια των αντίστοιχων εντολών (Εικόνα 4). Σύροντας μία εντολή από τη λίστα στο κεντρικό τμήμα του παραθύρου και ενώνοντάς την με τις ήδη υπάρχουσες, την εισάγουμε στο πρόγραμμα. Για να αφαιρέσουμε μία ή περισσότερες εντολές από το πρόγραμμα, αρκεί να τις σύρουμε αριστερά στην περιοχή όπου υπάρχουν οι κατηγορίες των εντολών.



Εικόνα 4

Με τα κουμπιά στο επάνω μέρος του παραθύρου μπορούμε να δημιουργήσουμε ένα νέο πρόγραμμα (*Νέο*), να αποθηκεύσουμε το πρόγραμμα το οποίο δουλεύουμε αυτή τη στιγμή σε μορφή *.abr* (*Αποθήκευση* και *Αποθήκευση ως*) και να ανοίξουμε ένα αρχείο *.abr* (*Άνοιγμα*). Το κουμπί *Αποθήκευση ως εικόνα*, στο κάτω μέρος, μας επιτρέπει να αποθηκεύσουμε το πρόγραμμα του ArduinoBlock με τη μορφή μίας εικόνας, που μπορεί να ενσωματωθεί π.χ. σε μία παρουσίαση.

Όταν πατηθεί το κουμπί *Ανέβασε στο Arduino*, το πρόγραμμα που έχουμε φτιάξει στο ArduinoBlock, μεταφράζεται σε Wiring C μέσα στο ανοικτό παράθυρο του Arduino IDE και στη συνέχεια φορτώνεται από το IDE στην πλακέτα. Κατά την πρώτη φόρτωση, μας δίνεται η επιλογή να αποθηκεύσουμε τη μορφή Wiring C του προγράμματος, αν δεν το έχουμε κάνει ήδη. Το κουμπί *Σειριακή οθόνη* εμφανίζει ένα παράθυρο (Εικόνα 5), μέσα



Εικόνα 5

στο οποίο μπορούμε να λάβουμε και να στείλουμε δεδομένα από και προς το Arduino, μέσω της σύνδεσης USB με τον υπολογιστή. Απαραίτητη προϋπόθεση για τη λειτουργία των δύο τελευταίων κουμπιών, είναι να υπάρχει συνδεδεμένη πλακέτα και να έχουν οριστεί σωστά οι παράμετροι για τη σύνδεση (τύπος πλακέτας και θύρα) μέσα από το περιβάλλον του IDE (μενού Εργαλεία).

Προσοχή: Αν κλείσουμε το παράθυρο του Arduino IDE, μέσα από το οποίο καλέσαμε το ArduBlock, τότε κλείνει και το παράθυρο του ArduBlock.



## Εφαρμογή 1: Έλεγχος LED

Στα πλαίσια της εφαρμογής θα παρουσιαστεί η χρήση των ψηφιακών εξόδων της πλακέτας καθώς και η βασική δομή ενός προγράμματος Arduino. Η τελική κατασκευή θα είναι ένα LED που θα αναβοσβήνει περιοδικά.

### 1. Υλικά

Στη συνέχεια παρουσιάζονται τα υλικά που θα χρησιμοποιήσουμε για την εφαρμογή μας.

#### LED

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.



Εικόνα 6 LED

Ως δίοδος, το LED επιτρέπει τη διέλευση του ρεύματος μόνο προς μία φορά. Όπως φαίνεται και στην Εικόνα 6 οι δύο ακροδέκτες του LED έχουν διαφορετικό μήκος. Ο πιο μακρύς ονομάζεται άνοδος και συνδέεται στο θετικό πόλο της πηγής, ενώ ο πιο κοντός ονομάζεται κάθοδος και συνδέεται στον αρνητικό πόλο (γείωση).

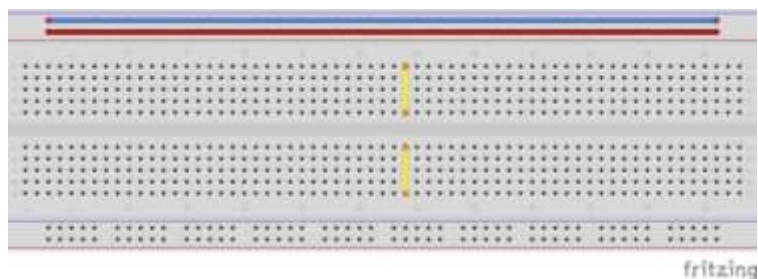
#### Αντίσταση

Ανάμεσα στο LED και την πηγή πρέπει να παρεμβάλλεται μία αντίσταση (Εικόνα 7), προκειμένου να περιοριστεί το ρεύμα που θα διαρρεύσει το κύκλωμα και να προστατευτεί το LED. Η πιο συνηθισμένη τιμή αντίστασης, που χρησιμοποιείται με το LED στις εφαρμογές Arduino είναι 220 Ω.



Εικόνα 7 Αντίσταση

## Breadboard



Εικόνα 8 Breadboard

## Arduino UNO

Το Arduino UNO (Εικόνα 9) διαθέτει 14 ψηφιακούς ακροδέκτες εισόδου/εξόδου (0 - 13). Όταν οι ψηφιακοί ακροδέκτες χρησιμοποιούνται ως έξοδοι, μπορούν να τεθούν σε μία από δύο καταστάσεις: HIGH (5V) και LOW (0V).

Στα πλαίσια της εφαρμογής μας θα χρησιμοποιήσουμε το Arduino ως μία προγραμματιζόμενη πηγή για την τροφοδοσία του κυκλώματος. Συγκεκριμένα θα αξιοποιήσουμε έναν από τους ψηφιακούς ακροδέκτες, ο οποίος όταν τίθεται σε κατάσταση HIGH το LED θα ανάβει, ενώ όταν τίθεται σε κατάσταση LOW το LED θα σβήνει. Ο ακροδέκτης γείωσης (GND) θα παίζει το ρόλο του αρνητικού πόλου της πηγής.



Εικόνα 9 Arduino UNO

## Καλώδια

Για τη διασύνδεση των διαφόρων στοιχείων των κυκλωμάτων μας, θα χρησιμοποιηθούν ειδικά καλώδια, που είναι κατάλληλα για χρήση με breadboard και ονομάζονται jumper cables.

## Κατασκευή κυκλώματος

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος.

## Βήμα 1

Συνδέουμε ένα καλώδιο από το pin 5 του Arduino σε μία ελεύθερη στήλη του breadboard.

## Βήμα 2

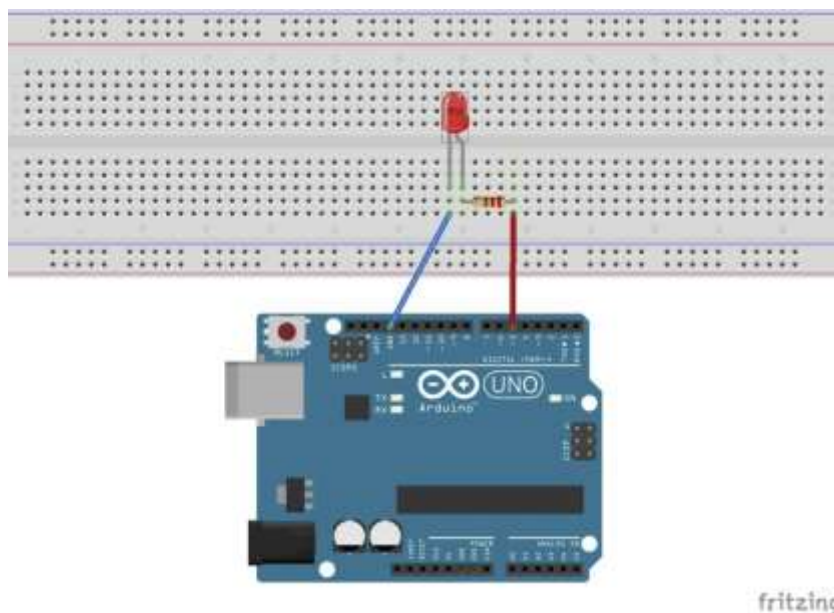
Συνδέουμε το ένα άκρο της αντίστασης στη στήλη όπου καταλήγει το καλώδιο και το άλλο σε μία κενή στήλη του breadboard.

## Βήμα 3

Συνδέουμε την άνοδο του LED (μακρύ ποδαράκι) στην ίδια στήλη με το ελεύθερο άκρο της αντίστασης και την κάθοδο του LED σε μία ελεύθερη στήλη του breadboard.

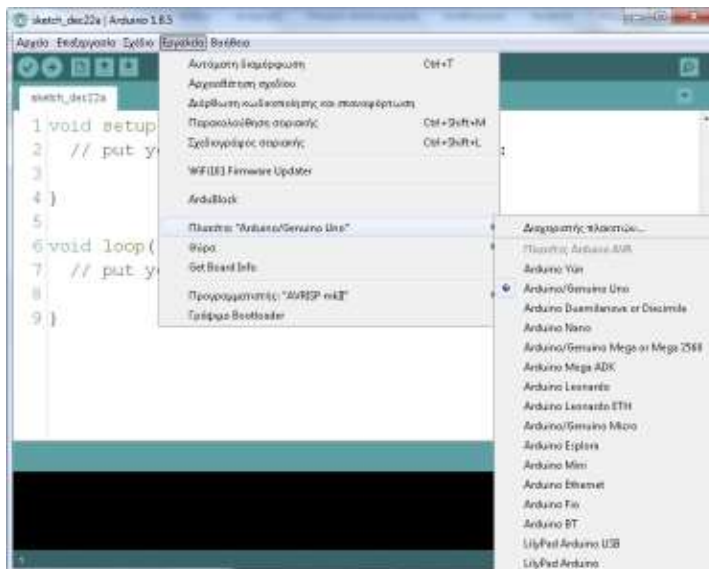
## Βήμα 4

Συνδέουμε ένα καλώδιο από τον ακροδέκτη GND, στη στήλη του breadboard όπου καταλήγει η κάθοδος του LED.



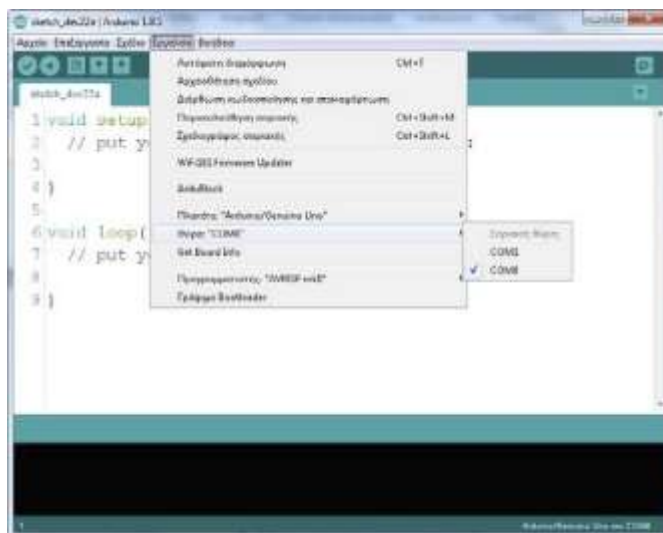
## Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα σε μία από τις USB θύρες του υπολογιστή. Στη συνέχεια, εκτελούμε το Arduino IDE. Στο παράθυρο που θα ανοίξει πηγαίνουμε στο μενού *Εργαλεία*, στην εγγραφή *Πλακέτα* και επιλέγουμε *Arduino/Genuino UNO* (Εικόνα 10).



Εικόνα 10

Ακολούθως, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino (Εικόνα 11).



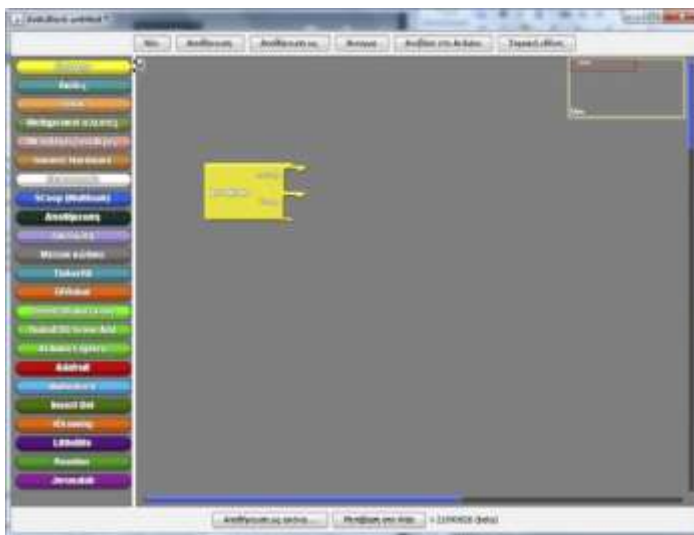
Εικόνα 11

## Ανάπτυξη προγράμματος

Από το μενού εργαλεία του Arduino IDE επιλέγουμε *ArduBlock*. Στη συνέχεια ακολουθούμε τα παρακάτω βήματα για να αναπτύξουμε το πρόγραμμα. (Εναλλακτικά μπορείτε μέσα από το ArduBlock να ανοίξετε το αρχείο με όνομα *LED\_Blink.abp*, που υπάρχει στο συνοδευτικό υλικό και περιέχει έτοιμο το πρόγραμμα. Για το σκοπό αυτό πατήστε το κουμπί **Άνοιγμα** και εντοπίστε το αρχείο στη θέση όπου το έχετε κατεβάσει. Στη συνέχεια προχωρήστε απευθείας στο 7<sup>ο</sup> βήμα.)

### Βήμα 1<sup>ο</sup>

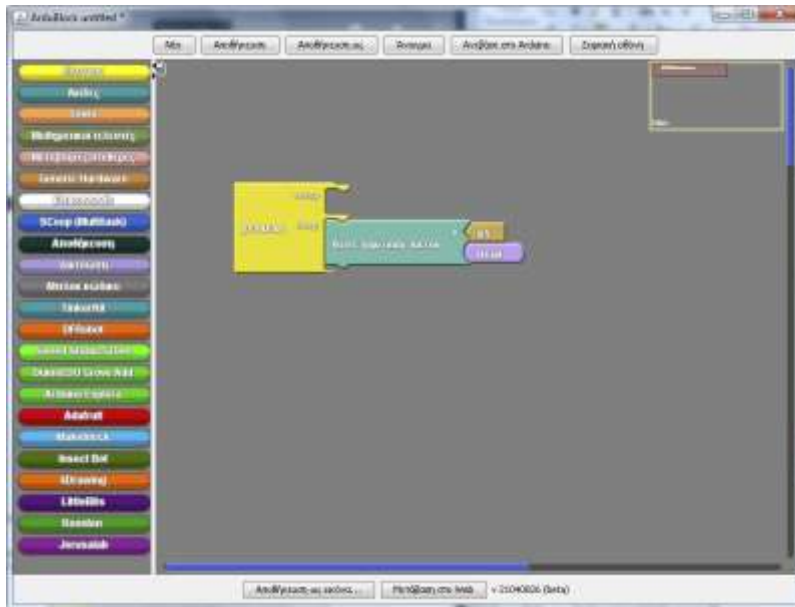
Αφαιρούμε το πλακίδιο *για πάντα κάνε* και στη συνέχεια επιλέγουμε το κουμπί **Έλεγχος** και σύρουμε το πλακίδιο *program* στο κεντρικό παράθυρο (Εικόνα 12). Το πλακίδιο αυτό αντιπροσωπεύει τη βασική δομή ενός προγράμματος Arduino. Οι εντολές που τοποθετούνται μέσα στο *setup* εκτελούνται μόνο μία φορά όταν ξεκινάει η εκτέλεση του προγράμματος, ενώ η συνάρτηση *loop* εκτελείται συνέχεια (επαναλαμβανόμενα) όσο υπάρχει τροφοδοσία.



Εικόνα 12

### Βήμα 2<sup>ο</sup>

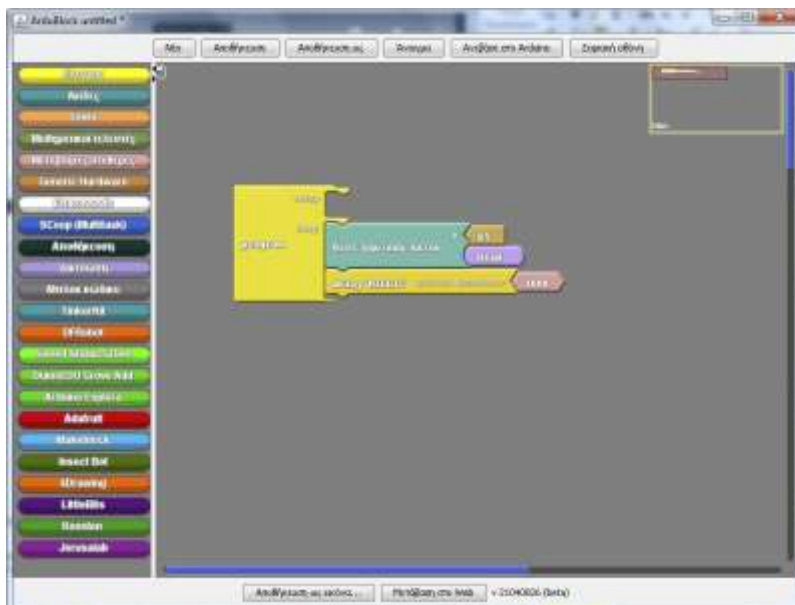
Από την καρτέλα *Ακίδες* σύρουμε το πλακίδιο *θέσε ψηφιακή ακίδα* μέσα στο *loop*, ώστε να κουμπώσει. Πηγαίνουμε το ποντίκι πάνω από τον αριθμό της ακίδας, πατάμε στο βελάκι που εμφανίζεται και επιλέγουμε από τη λίστα τον ακροδέκτη D5 για να αντιστοιχεί στην ακίδα όπου έχει συνδεθεί το LED (Εικόνα 13).



Εικόνα 13

### Βήμα 3<sup>ο</sup>

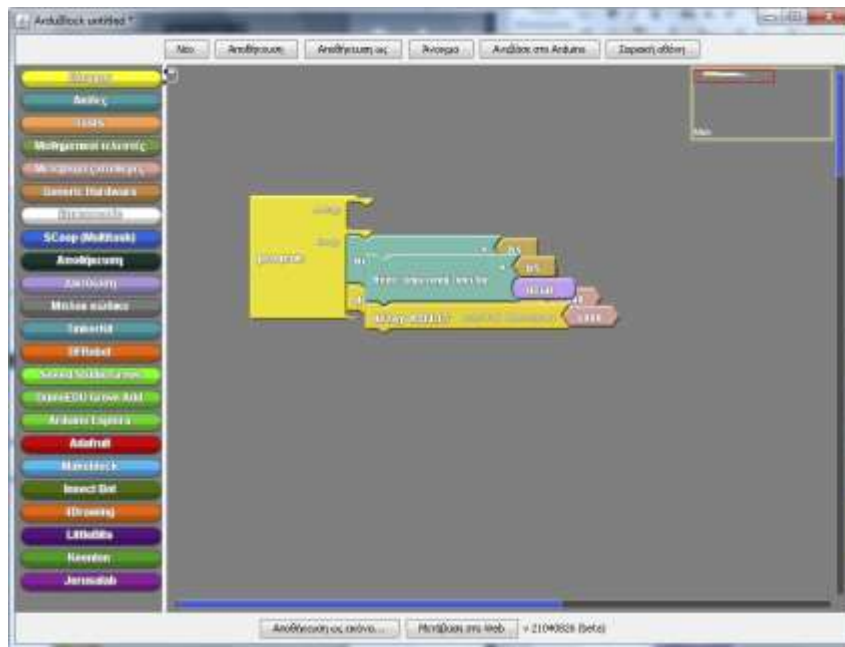
Από την καρτέλα *Έλεγχος*, προσθέτουμε την εντολή *delay MILLIS* κάτω από την εντολή *θέσε ψηφιακή ακίδα* και αφήνουμε την τιμή καθυστέρησης 1000 (Εικόνα 14). Με τον τρόπο αυτό το LED θα μένει αναμμένο για 1 δευτερόλεπτο.



Εικόνα 14

### Βήμα 4°

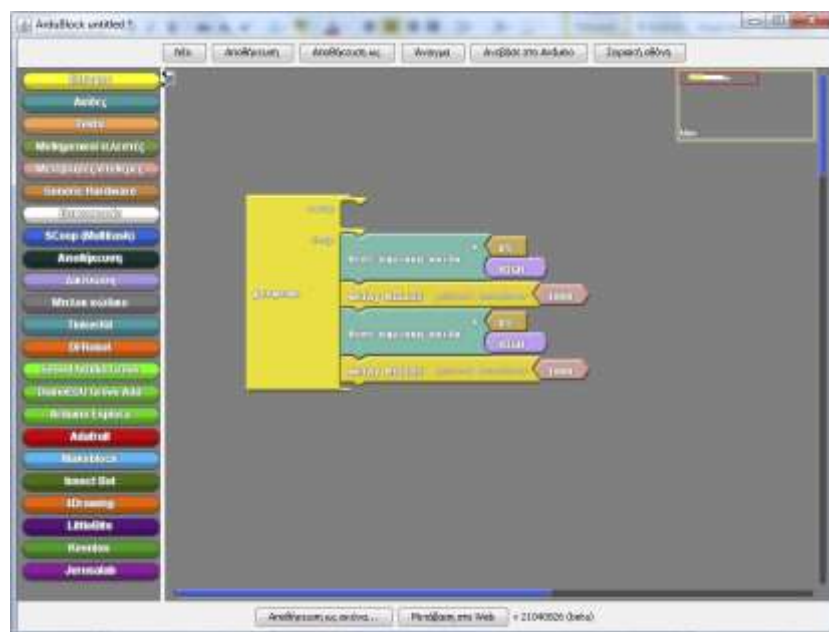
Κάνουμε δεξί κλικ πάνω στην εντολή *θέσε ψηφιακή ακίδα* που προσθήσαμε στο 1° βήμα και επιλέγουμε *Αντέγραψε*. (Εικόνα 15)



Εικόνα 15

### Βήμα 5°

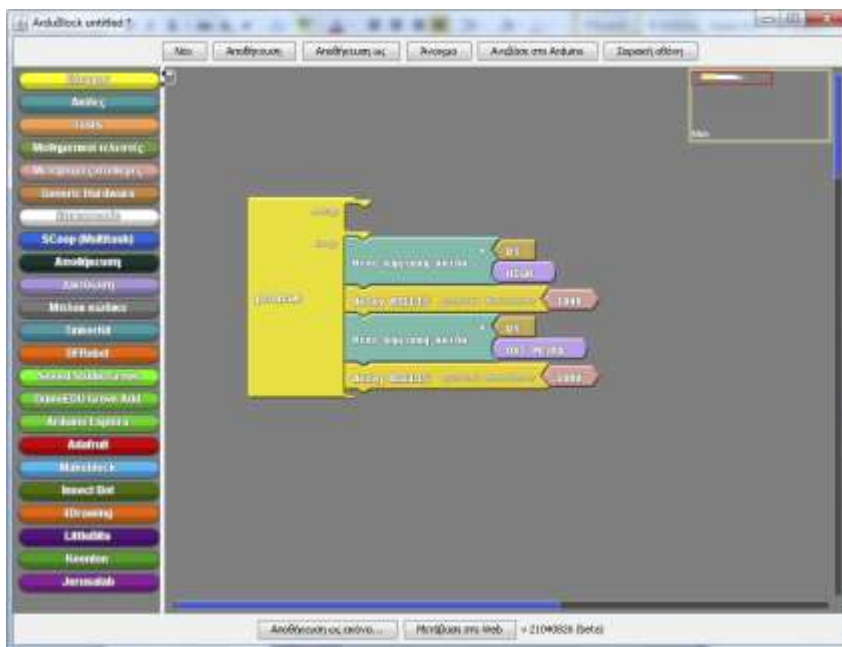
Σύρουμε τις εντολές που εμφανίστηκαν με την αντιγραφή, ώστε να κουμπώσουν κάτω από την εντολή *delay MILLIS* του βήματος 3. (Εικόνα 16)



Εικόνα 16

## Βήμα 6°

Αλλάζουμε την τιμή της νέας εντολής **θέσε ψηφιακή ακίδα** σε **OXI PEYMA** κάνοντας κλικ στο βελάκι που εμφανίζεται όταν ο δείκτης περνάει πάνω από το πλακίδιο **HIGH** (Εικόνα 17). Οι εντολές αυτές κάνουν το LED να παραμένει σβηστό για ένα δευτερόλεπτο.



Εικόνα 17

## Βήμα 7°

### Φόρτωση του προγράμματος στο Arduino

Από το περιβάλλον του ArduBlock πατάμε το κουμπί **Ανέβασε στο Arduino**. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα και αρχίζει να εκτελείται.



## Εφαρμογή 2: Φανάρι Κυκλοφορίας

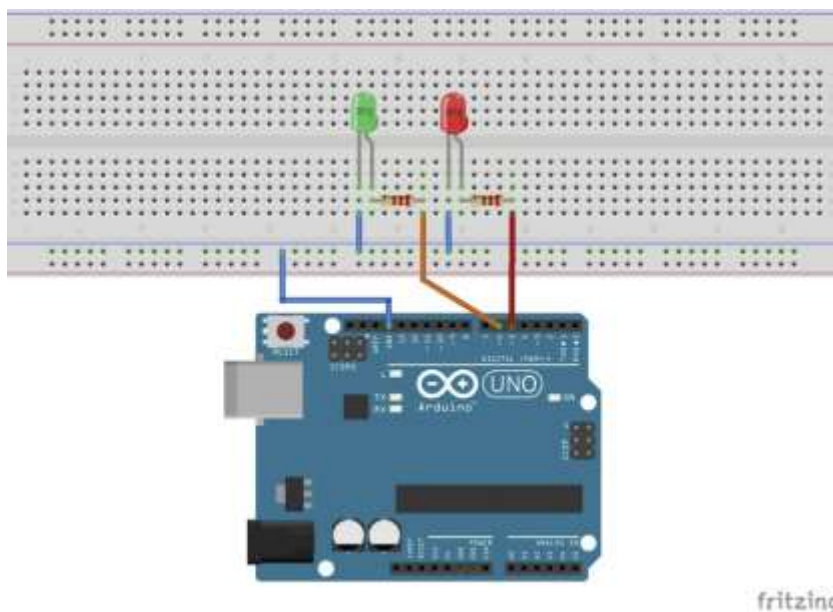
Η εφαρμογή που θα αναπτυχθεί θα προσομοιώνει τη λειτουργία ενός φαναριού κυκλοφορίας. Συγκεκριμένα, θα χρησιμοποιηθούν δύο LED (ένα κόκκινο και ένα πράσινο), τα οποία θα ανάβουν εναλλάξ με προκαθορισμένες διάρκειες. Στα πλαίσια της εφαρμογής, παρουσιάζεται η αξιοποίηση του breadboard σε πιο σύνθετα κυκλώματα και η έννοια του συγχρονισμού ενεργειών στο πρόγραμμα.

### Υλικά

Σε σχέση με τα υλικά της Εφαρμογής 1, θα χρησιμοποιηθούν επιπλέον ένα LED, μία αντίσταση 220 Ω και 3 καλώδια.

### Κύκλωμα

Το κύκλωμα της εφαρμογής παρουσιάζεται στην Εικόνα 18. Μία πρώτη διαφορά σε σχέση με το κύκλωμα της Εφαρμογής 1, είναι ότι ο ακροδέκτης GND του Arduino έχει συνδεθεί σε μία οπή στην κάτω οριζόντια μπλε σειρά του breadboard. Με τον τρόπο αυτό, όλη η σειρά έχει μετατραπεί σε γείωση, στην οποία συνδέονται οι κάθοδοι των δυο LED. Από εκεί και πέρα, η άνοδος του κόκκινου LED συνδέεται μέσω της μίας αντίστασης στον ψηφιακό ακροδέκτη 5 του Arduino (όπως και στην 1<sup>η</sup> εφαρμογή), ενώ η άνοδος του πράσινου LED συνδέεται μέσω της 2<sup>ης</sup> αντίστασης στον ψηφιακό ακροδέκτη 6.



Εικόνα 18

## Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα σε μία ελεύθερη θύρα USB του υπολογιστή. Στη συνέχεια, εκτελούμε το Arduino IDE και μέσα από το μενού *Εργαλεία* κάνουμε κλικ στην εγγραφή *Πλακέτα* και επιλέγουμε *Arduino/Genuino UNO*. Ακολούθως, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino.

## Πρόγραμμα

Από το μενού *Εργαλεία* του IDE επιλέγουμε *ArduBlock*. Στην Εικόνα 19 παρουσιάζεται ο κώδικας της εφαρμογής.



Εικόνα 19

Σε κάθε επανάληψη της *loop*, αρχικά ο ακροδέκτης D5 τίθεται σε κατάσταση HIGH, ενώ ο D6 σε κατάσταση LOW. Σαν αποτέλεσμα, το κόκκινο LED ανάβει και το πράσινο σβήνει. Με την εντολή *delay MILLIS* που ακολουθεί, η κατάσταση των LED διατηρείται για 2 δευτερόλεπτα. Στη συνέχεια, ο ακροδέκτης D5 τίθεται σε κατάσταση LOW, ενώ ο D6 σε κατάσταση HIGH. Πλέον το κόκκινο LED είναι σβηστό και το πράσινο αναμμένο. Η νέα κατάσταση του φαναριού διαρκεί για 5 δευτερόλεπτα (δεύτερη εντολή *delay MILLIS*).

Όπως γίνεται αντιληπτό, ο συγχρονισμός στην αλλαγή των καταστάσεων των δύο LED, επιτυγχάνεται εκτελώντας τις αντίστοιχες εντολές μαζί, πριν από την κοινή εντολή καθυστέρησης (*delay MILLIS*).

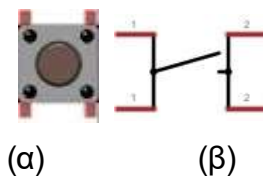
## Εφαρμογή 3: LED με κουμπί

Οι ψηφιακοί ακροδέκτες του Arduino (ακροδέκτες 0 - 13 στο UNO) μπορούν να χρησιμοποιηθούν είτε ως έξοδοι είτε ως είσοδοι. Προκειμένου να γνωρίσουμε τη χρήση των ψηφιακών εισόδων του Arduino, θα προσθέσουμε στην 1<sup>η</sup> εφαρμογή ένα κουμπί. Όσο το κουμπί παραμένει πατημένο, το LED θα είναι αναμμένο. Μόλις αφήνουμε το κουμπί το LED θα σβήνει. Για το σκοπό αυτό, θα χρησιμοποιήσουμε μέσα στο σχετικό πρόγραμμα μία εντολή ελέγχου *αν/αλλιώς*, μέσα στην οποία θα ελέγχουμε την κατάσταση της ακίδας.

### Υλικά

Στη συνέχεια παρουσιάζονται τα επιπλέον υλικά που θα χρησιμοποιηθούν σε αυτή τη δεύτερη εφαρμογή.

#### Κουμπί



Εικόνα 20

Όπως φαίνεται στην Εικόνα 20(α), το κουμπί πίεσης διαθέτει 4 ακροδέκτες. Οι ακροδέκτες αυτοί είναι ανά 2 συνδεδεμένοι μεταξύ τους (πάνω-κάτω), ενώ η διάταξη χωρίζεται σε δύο ανεξάρτητα κομμάτια (δεξί - αριστερό) (Εικόνα 20(β)). Όταν πιέζουμε το κουμπί, κλείνει ο διακόπτης και συνδέεται το δεξί με το αριστερό του μέρος.

#### Αντίσταση 10kΩ

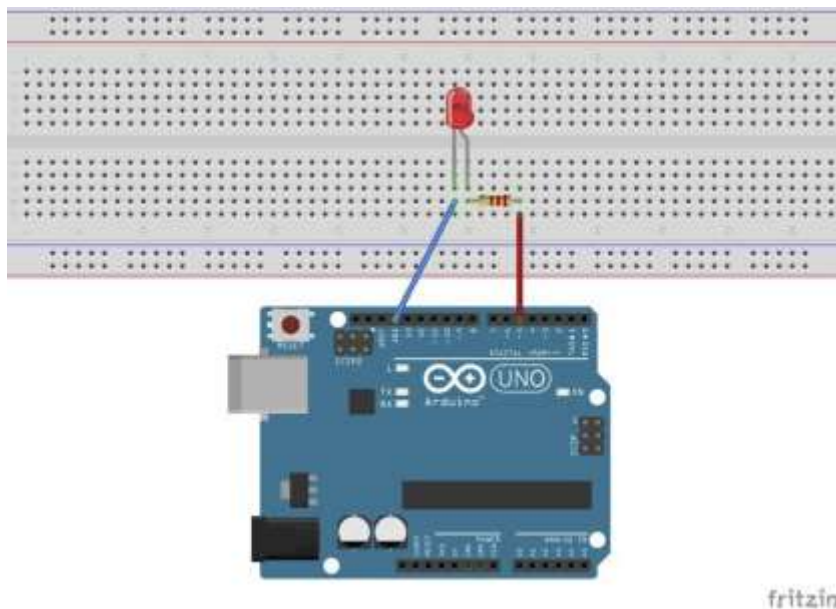


Εικόνα 21

Για τη σύνδεση του κουμπιού με το Arduino θα χρησιμοποιήσουμε μία αντίσταση 10 kΩ.

## Κύκλωμα

Το κύκλωμα θα περιλαμβάνει ένα LED και ένα κουμπί. Όσον αφορά στη συνδεσμολογία του LED, δεν υπάρχει κάποια διαφοροποίηση σε σχέση με το κύκλωμα της 1<sup>ης</sup> εφαρμογής (Εικόνα 22).



Εικόνα 22

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η προσθήκη του κουμπιού στο κύκλωμα.

### Βήμα 1

Τοποθετούμε το κουμπί πάνω στο breadboard, όπως φαίνεται στην εικόνα. Με τον τρόπο αυτό, οι συνδεδεμένοι ακροδέκτες γεφυρώνουν τα 2 ανεξάρτητα τμήματα των στηλών στις οποίες πατάνε.

### Βήμα 2

Παίρνουμε ένα καλώδιο και συνδέουμε τη μία άκρη του στον ακροδέκτη 5V του Arduino και την άλλη στην ίδια στήλη με τον κάτω αριστερό ακροδέκτη του κουμπιού.

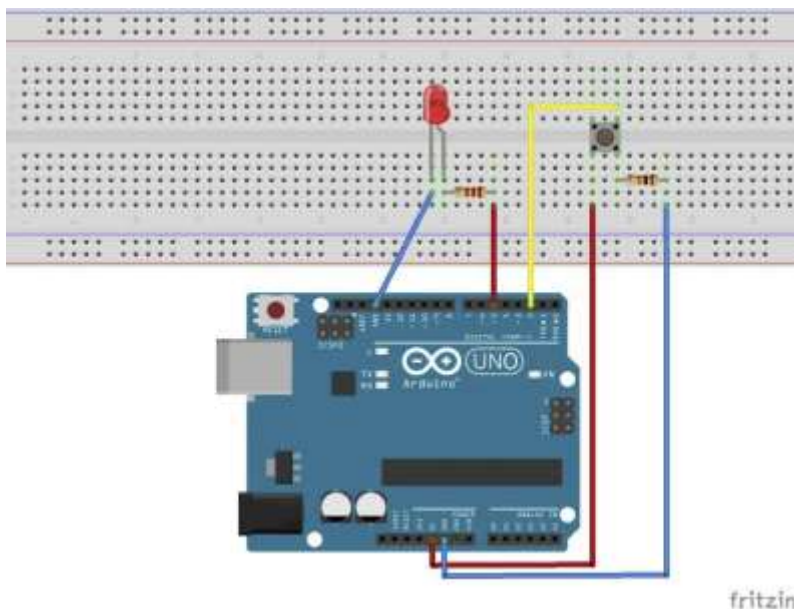
### Βήμα 3

Συνδέουμε το ένα άκρο της αντίστασης 10 kΩ στην ίδια στήλη με τον κάτω δεξιό ακροδέκτη του κουμπιού και το άλλο άκρο σε μία ελεύθερη στήλη του breadboard.

## Βήμα 4

Συνδέουμε ένα καλώδιο από τον ακροδέκτη GND του Arduino στη στήλη όπου καταλήγει το ελεύθερο άκρο της αντίστασης 10kΩ.

## Βήμα 5



Παίρνουμε ένα ακόμα καλώδιο και συνδέουμε το ένα του άκρο στον ψηφιακό ακροδέκτη 2 του Arduino και το άλλο στην ίδια στήλη με τον πάνω δεξιό ακροδέκτη του κουμπιού. Στη συνέχεια εξηγείται η συνδεσμολογία και η λειτουργία του κουμπιού.

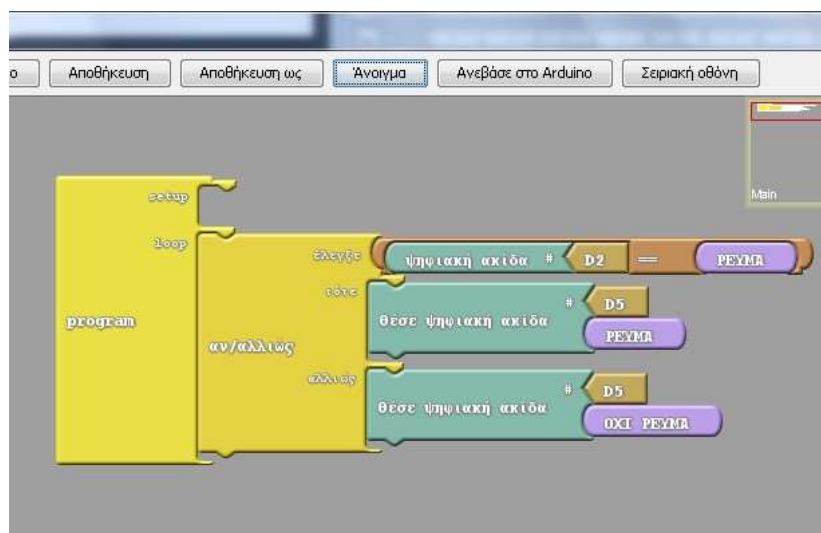
Το αριστερό τμήμα του κουμπιού συνδέεται στην τροφοδοσία (pin 5 V) και το δεξί στη γείωση (GND) μέσω της αντίστασης 10 kΩ, για αποφυγή βραχυκυκλώματος. Για τον έλεγχο της κατάστασης του κουμπιού, συνδέουμε το δεξί του τμήμα στον ψηφιακό ακροδέκτη 2 του Arduino, που θα χρησιμοποιηθεί ως είσοδος. Όταν το κουμπί δεν είναι πατημένο, το κύκλωμα είναι ανοικτό, δεν υπάρχει ρεύμα, ούτε και πτώση τάσης στην αντίσταση. Άρα, ο ακροδέκτης 2 είναι απευθείας συνδεδεμένος στη γείωση (κατάσταση LOW). Όταν πατηθεί το κουμπί, κλείνει το κύκλωμα και ο ακροδέκτης 2 συνδέεται απευθείας στα 5 V (κατάσταση HIGH).

## Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα στον υπολογιστή και μέσα από το μενού *Εργαλεία* του Arduino IDE ορίζουμε τον τύπο της πλακέτας και τη θύρα σύνδεσης.

## Πρόγραμμα

Από το μενού *Εργαλεία* του IDE επιλέγουμε *ArduBlock* και προχωράμε στην ανάπτυξη του προγράμματος της εφαρμογής, όπως παρουσιάζεται στην Εικόνα 23. Η ανάγνωση της κατάστασης μίας ψηφιακής ακίδας υλοποιείται στο ArduBlock με την εντολή *ψηφιακή ακίδα*. Η εντολή αυτή υπάρχει μέσα στην ομάδα εντολών *Ακίδες*.



Εικόνα 23

Σε κάθε επανάληψη της *loop*, το πρόγραμμα διαβάζει και ελέγχει την κατάσταση της ακίδας D2. Ο έλεγχος υλοποιείται με την εντολή *αν/αλλιώς* που υπάρχει μέσα στην ομάδα εντολών *Έλεγχος*, ενώ ο τελεστής σύγκρισης *==* βρίσκεται μέσα στην ομάδα *Tests*.

Αν η κατάσταση της ακίδας εισόδου είναι HIGH, τότε η ακίδα εξόδου D5 τίθεται σε κατάσταση HIGH, κάνοντας το LED να ανάβει. Σε διαφορετική περίπτωση, η ακίδα D5 τίθεται σε κατάσταση LOW και το LED σβήνει.

## Εφαρμογή 4: LED που αναβοσβήνει και LED με κουμπί

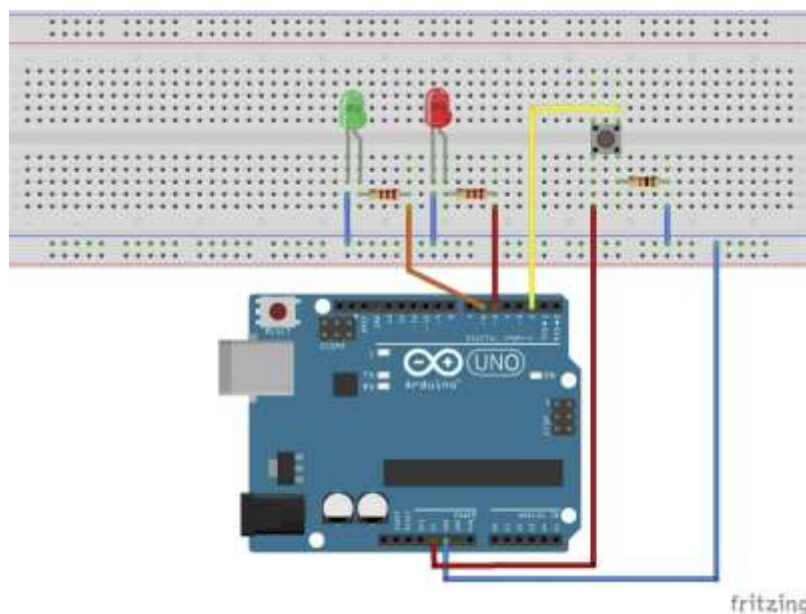
Αυτή η εφαρμογή αποτελεί έναν συνδυασμό των Εφαρμογών 1 και 3. Συγκεκριμένα, θα υπάρχουν δύο LED και ένα κουμπί. Το ένα LED θα αναβοσβήνει περιοδικά (όπως στην Εφαρμογή 1), ενώ το δεύτερο θα ανάβει όσο το κουμπί είναι πατημένο (Εφαρμογή 3). Στα πλαίσια της ανάπτυξης του προγράμματος, θα εντοπίσουμε ένα μειονέκτημα της εντολής *delay* *MILLIS* και θα προτείνουμε μία εναλλακτική υλοποίηση χρονοκαθυστέρησης.

### Υλικά

Τα υλικά που θα χρησιμοποιηθούν για την εφαρμογή περιλαμβάνουν 2 LED, 2 αντιστάσεις 220Ω (για τα LED), ένα κουμπί, 1 αντίσταση 10kΩ (για το κουμπί) και 8 καλώδια.

### Κύκλωμα

Όπως αναφέρθηκε και προηγουμένα, η εφαρμογή θα περιλαμβάνει δύο LED και ένα κουμπί. Ουσιαστικά, το κύκλωμα θα είναι αυτό της εφαρμογής 3, με μία αλλαγή στη συνδεσμολογία της γείωσης και με ένα επιπλέον LED. Το τελικό κύκλωμα φαίνεται στην Εικόνα 24.



Εικόνα 24

Όπως φαίνεται στην εικόνα, η άνοδος του δεύτερου LED έχει συνδεθεί, μέσω της αντίστασης 220Ω, στον ψηφιακό ακροδέκτη 6 του Arduino. Η διαφοροποίηση όσον αφορά στη συνδεσμολογία της γείωσης είναι ότι, ο ακροδέκτης GND του Arduino έχει

συνδεθεί με την κάτω μπλε οριζόντια σειρά του breadboard και στη συνέχεια, όλα τα εξαρτήματα παίρνουν γείωση από το breadboard (όπως είδαμε και στην Εφαρμογή 2).

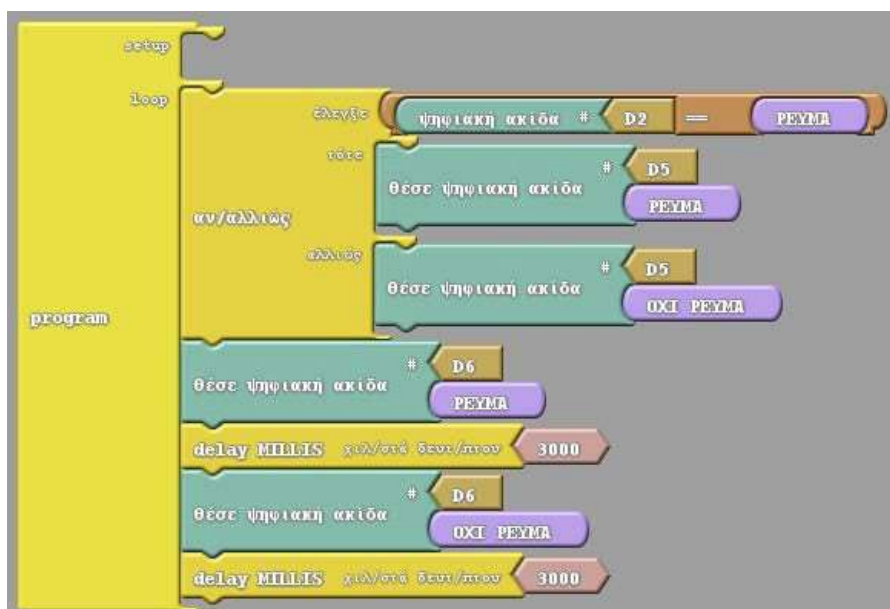
## Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα στον υπολογιστή και μέσα από το μενού *Εργαλεία* του Arduino IDE ορίζουμε τον τύπο της πλακέτας και τη θύρα σύνδεσης.

## Πρόγραμμα

Από το μενού *Εργαλεία* του IDE επιλέγουμε *ArduBlock* και ανοίγει το σχετικό παράθυρο. Το πρόγραμμα που θα αναπτυχθεί, θα πρέπει να συνδυάζει τις λειτουργίες των Εφαρμογών 3 και 1. Συγκεκριμένα το LED που συνδέεται στον ακροδέκτη D5 θα ανάβει μόνο όσο το κουμπί παραμένει πατημένο, ενώ το LED στον ακροδέκτη D6 θα αναβοσβήνει περιοδικά αλλάζοντας κατάσταση π.χ. κάθε 3 δευτερόλεπτα.

Μία πρώτη σκέψη για την υλοποίηση του προγράμματος, είναι να συνδυάσουμε τα αντίστοιχα προγράμματα των εφαρμογών 3 και 1. Στην Εικόνα 25 παρουσιάζεται ο αρχικός κώδικας της εφαρμογής.



Εικόνα 25

Σε κάθε επανάληψη της *loop*, το πρόγραμμα αρχικά ελέγχει το κουμπί και με βάση αυτό τον έλεγχο ορίζει την κατάσταση του LED που συνδέεται στον ακροδέκτη D5. Στη συνέχεια, εκτελούνται οι εντολές που κάνουν το LED στον ακροδέκτη D6 να ανάψει, να παραμείνει αναμμένο για 3 δευτερόλεπτα, να σβήσει και να παραμείνει σβηστό για 3 δευτερόλεπτα.



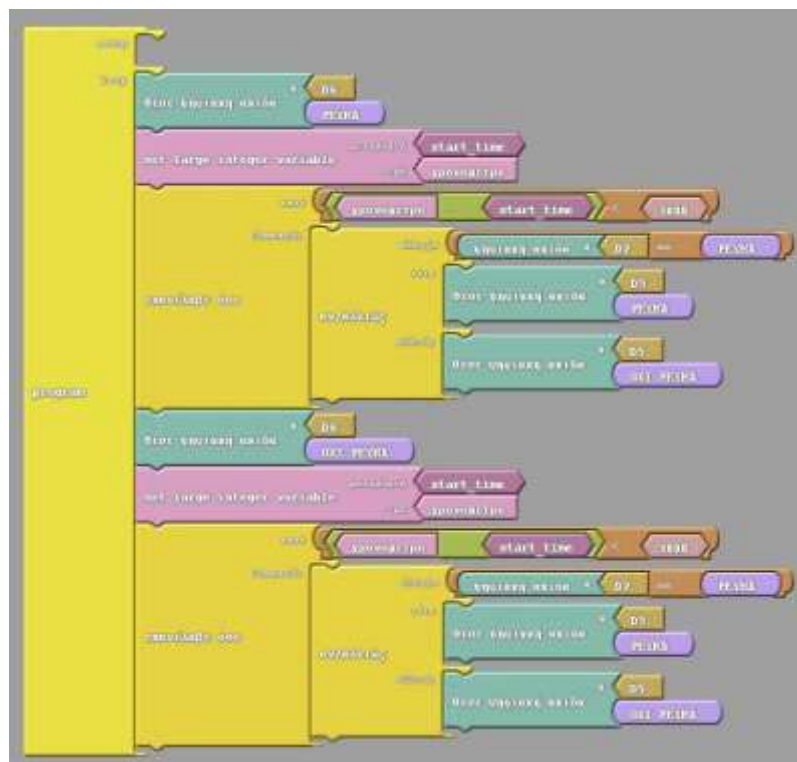
Ανεβάστε το πρόγραμμα στο Arduino και δοκιμάστε τη λειτουργία του πατώντας το κουμπί αρκετές φορές. Όπως μπορείτε να παρατηρήσετε, ενώ το LED που αναβοσβήνει λειτουργεί κανονικά, το LED που ελέγχεται από το κουμπί δεν έχει την αναμενόμενη συμπεριφορά.

Η δυσλειτουργία που παρατηρείται, οφείλεται στο γεγονός ότι η κατάσταση του κουμπιού ελέγχεται στο πρόγραμμα μόνο μία στιγμή στην αρχή (εντολή *αν/αλλιώς*), ενώ στα 6 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 πρέπει η κατάστασή του να ελέγχεται συνέχεια κατά τη διάρκεια των δύο χρονοκαθυστερήσεων (εντολές *delay MILLIS*).

Ωστόσο, η εντολή *delay MILLIS* είναι υλοποιημένη με τέτοιο τρόπο, που ο μικροελεγκτής του Arduino να παραμένει άεργος (δεν εκτελεί καμία εντολή) κατά τη διάρκεια της καθυστέρησης. Έτσι, χρειάζεται να υπάρχει μία εναλλακτική υλοποίηση χρονοκαθυστερήσης, που θα επιτρέπει την εκτέλεση εντολών κατά τη διάρκειά της.

Για το σκοπό αυτό, χρησιμοποιούμε τη συνάρτηση *χρονόμετρο* (ομάδα εντολών *Έλεγχος*), σε συνδυασμό με μία εντολή επανάληψης *επανάλαβε όσο* (ομάδα εντολών *Έλεγχος*) και μία μεταβλητή. Η συνάρτηση *χρονόμετρο* επιστρέφει κάθε στιγμή το χρόνο που έχει περάσει από την έναρξη εκτέλεσης του προγράμματος (σε χιλιοστά του δευτερολέπτου).

Στη συνέχεια εξηγείται η υλοποίηση μίας χρονοκαθυστερήσης με συγκεκριμένη διάρκεια (έστω *duration*). Αρχικά, αποθηκεύουμε την τιμή της συνάρτησης *χρονόμετρο* (χρονική στιγμή έναρξης της καθυστέρησης) σε μία μεταβλητή (έστω *start\_time*). Στη συνέχεια προσθέτουμε μία εντολή *επανάλαβε όσο* με συνθήκη *χρονόμετρο - start\_time < duration*. Με τον τρόπο αυτό, η εντολή επανάληψης εκτελείται συνέχεια, μέχρι να παρέλθει ο επιθυμητός χρόνος. Μέσα στο σώμα της *επανάλαβε όσο* μπορούμε να προσθέσουμε εντολές, οι οποίες θα εκτελούνται συνέχεια κατά τη διάρκεια της χρονοκαθυστερήσης.



Εικόνα 26

Στην Εικόνα 26 παρουσιάζεται η τελική μορφή του κώδικα της εφαρμογής. Οι δύο εντολές *delay MILLIS* έχουν αντικατασταθεί σύμφωνα με όσα αναφέρονται παραπάνω, ενώ η εντολή *αν/αλλιώς* που ελέγχει το κουμπί και ορίζει την κατάσταση του LED του ακροδέκτη D5, έχει μεταφερθεί μέσα στις δύο εντολές *επανάλαβε όσο*. Η εντολή *set large integer variable*, καθώς και το πλακίδιο για τη σταθερή τιμή 3000, βρίσκονται μέσα στην ομάδα εντολών *Μεταβλητές/σταθερές*. Οι τελεστές σύγκρισης < και = βρίσκονται στην ομάδα εντολών *Tests*. Ο τελεστής αφαίρεσης - βρίσκεται στην ομάδα εντολών *Μαθηματικοί τελεστές*.

Ανεβάστε το πρόγραμμα στο Arduino και επαληθεύστε την ορθή λειτουργία του, πατώντας το κουμπί.

## Εφαρμογή 5: LED με μεταβαλλόμενη ένταση

Στα πλαίσια της εφαρμογής θα παρουσιαστεί η χρήση των ψευδοαναλογικών εξόδων του Arduino. Η τελική κατασκευή θα είναι ένα LED που θα αναβοσβήνει σταδιακά. Για το σκοπό αυτό, στο πρόγραμμα θα χρησιμοποιηθεί μία εντολή επανάληψης που εκτελείται για συγκεκριμένο πλήθος φορών, μεταβάλλοντας την τιμή μίας μεταβλητής.

### Υλικά

#### LED

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.



Εικόνα 27 LED

Ως δίοδος, το LED επιτρέπει τη διέλευση του ρεύματος μόνο προς μία κατεύθυνση. Όπως φαίνεται και στην Εικόνα 27 οι δύο ακροδέκτες του LED έχουν διαφορετικό μήκος. Ο πιο μακρύς ονομάζεται άνοδος και συνδέεται στο θετικό πόλο της πηγής, ενώ ο πιο κοντός ονομάζεται κάθοδος και συνδέεται στον αρνητικό πόλο (γείωση).

#### Αντίσταση 220Ω

Ανάμεσα στο LED και την πηγή πρέπει να παρεμβάλλεται μία αντίσταση (Εικόνα 28), προκειμένου να περιοριστεί το ρεύμα που θα διαρρεύσει το κύκλωμα και να προστατευτεί το LED. Η πιο συνηθισμένη τιμή αντίστασης, που χρησιμοποιείται με το LED στις εφαρμογές Arduino είναι 220 Ω.

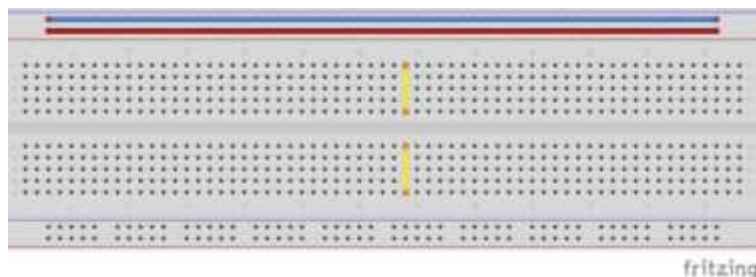


Εικόνα 28 Αντίσταση

#### Breadboard

Το breadboard είναι μία διάταξη που επιτρέπει την εύκολη κατασκευή κυκλωμάτων χωρίς να απαιτούνται κολλήσεις. Συγκεκριμένα, το breadboard διαθέτει οπές πάνω στις

οποίες μπορούν να συνδεθούν διάφορα ηλεκτρικά και ηλεκτρονικά στοιχεία. Εσωτερικά οι οπές αυτές διασυνδέονται μεταξύ τους όπως φαίνεται στην Εικόνα 29. Οι 4 εξωτερικές σειρές (2 πάνω και 2 κάτω) είναι συνδεδεμένες οριζόντια, ενώ στο κεντρικό του τμήμα οι διασυνδέσεις είναι κατακόρυφες, με κάθε στήλη να χωρίζεται σε δύο ανεξάρτητα τμήματα.



Εικόνα 29

### Arduino UNO

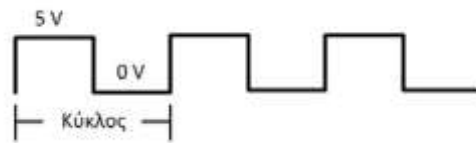
Το Arduino UNO (Εικόνα 30) διαθέτει 14 ψηφιακούς ακροδέκτες. Κάποιοι από αυτούς (οι 3, 5, 6, 9, 10 και 11), μπορούν μέσω μίας τεχνικής που ονομάζεται PWM να δίνουν ως έξοδο και ενδιάμεσες τιμές τάσης (μεταξύ 0V και 5V). Συγκεκριμένα, μπορούν να δίνουν 256 διαφορετικά επίπεδα τάσης, με το επίπεδο 0 να αντιστοιχεί στα 0 V, το επίπεδο 255 στα 5 V, το επίπεδο 127 στα 2,5 V κ.ο.κ.



Εικόνα 30

Με την τεχνική PWM οι στάθμες 0V και 5V εναλλάσσονται περιοδικά, παράγοντας έναν τετραγωνικό παλμό (Εικόνα 31). 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 μπορούμε να μεταβάλλουμε τη διάρκεια του θετικού μετώπου του παλμού.



Εικόνα 31

### Καλώδια

Για τη διασύνδεση των διαφόρων στοιχείων των κυκλωμάτων μας, θα χρησιμοποιηθούν ειδικά καλώδια, που είναι κατάλληλα για χρήση με breadboard και ονομάζονται jumper cables.

### Κύκλωμα

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος.

#### Βήμα 1

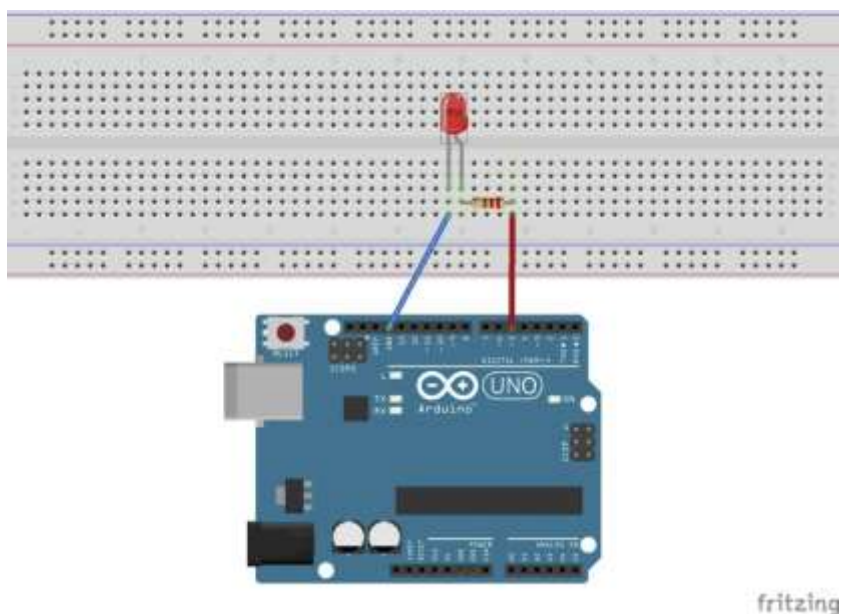
Συνδέουμε ένα καλώδιο από το pin 5 του Arduino σε μία ελεύθερη στήλη του breadboard.

#### Βήμα 2

Συνδέουμε το ένα άκρο της αντίστασης στη στήλη όπου καταλήγει το καλώδιο και το άλλο σε μία κενή στήλη του breadboard.

#### Βήμα 3

Συνδέουμε την άνοδο του LED (μακρύ ποδαράκι) στην ίδια στήλη με το ελεύθερο άκρο της αντίστασης και την κάθοδο του LED σε μία ελεύθερη στήλη του breadboard.

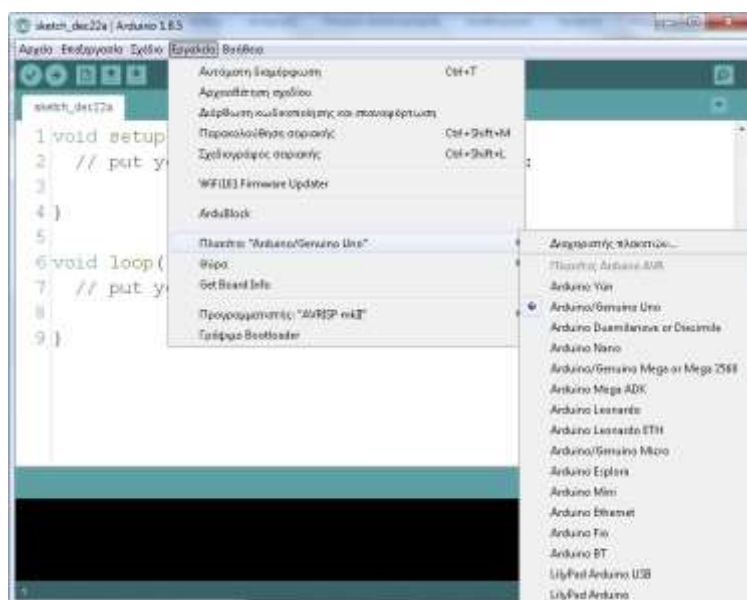


#### Βήμα 4

Συνδέουμε ένα καλώδιο από τον ακροδέκτη GND, στη στήλη του breadboard όπου καταλήγει η κάθοδος του LED.

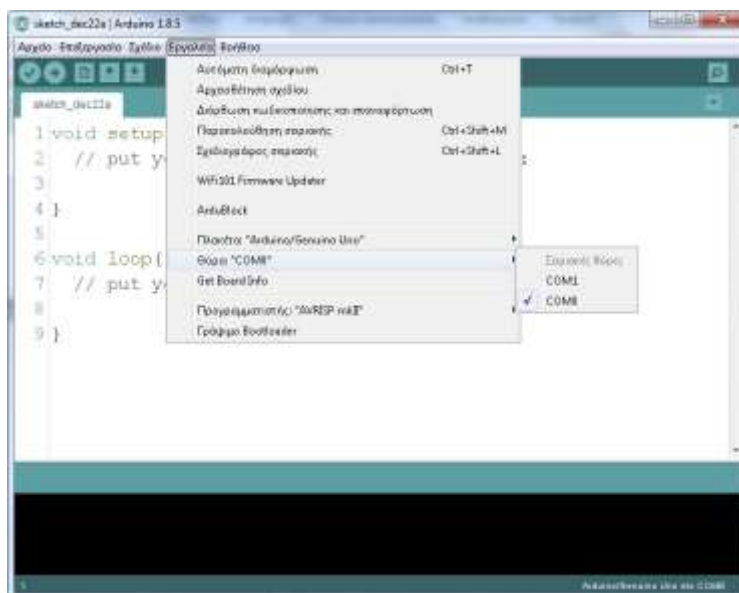
#### Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα σε μία από τις USB θύρες του υπολογιστή. Στη συνέχεια, εκτελούμε το Arduino IDE. Στο παράθυρο που θα ανοίξει πηγαίνουμε στο μενού *Εργαλεία*, στην εγγραφή *Πλακέτα* και επιλέγουμε *Arduino/Genuino Uno* (Εικόνα 32).



Εικόνα 32

Ακολουθως, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino (Εικόνα 33).



Εικόνα 33

## Πρόγραμμα

Από το μενού εργαλεία του Arduino IDE επιλέγουμε *ArduBlock* και ανοίγει το σχετικό περιβάλλον οπτικού προγραμματισμού, στο οποίο μπορούμε να αναπτύξουμε το πρόγραμμά μας με χρήση πλακιδίων. Ο ορισμός της τιμής μίας ψευδοαναλογικής εξόδου γίνεται στο ArduBlock με την εντολή *θέσε αναλογική ακίδα*, που βρίσκεται στην ομάδα εντολών *Ακίδες*.



Εικόνα 34

Στην Εικόνα 34 φαίνεται ο κώδικας της εφαρμογής. Μέσα στη *loop* υπάρχουν δύο εντολές επανάληψης *repeat between* (ομάδα εντολών Έλεγχος). Η πρώτη επαναλαμβάνεται για όλες τις τιμές της μεταβλητής *brightness* από 0 μέχρι 255, με βήμα μεταβολής 1. Σε κάθε επανάληψή της, θέτουμε στην ψευδοαναλογική ακίδα *D5* την τιμή που περιέχεται στη μεταβλητή *brightness*. Με τον τρόπο αυτό, τροφοδοτούμε διαδοχικά το LED με τα 256 διαφορετικά επίπεδα τάσης που υποστηρίζει η τεχνική PWM, ξεκινώντας από τα 0V και καταλήγοντας στα 5V (σταδιακό άναμμα). Για να είναι πιο ομαλή η μεταβολή, σε κάθε επανάληψη έχει προστεθεί και μία εντολή *delayMILLIS* (ομάδα εντολών Έλεγχος) που εισάγει μία καθυστέρηση 10ms.

Η δεύτερη *repeat between* αποτελεί αντίγραφο της πρώτης, με αντιστραμμένα τα όρια των τιμών της μεταβλητής *brightness* (από 255 μέχρι 0 με βήμα 1). Με τον τρόπο αυτό, επιτυγχάνεται το σταδιακό σβήσιμο του LED.

Μπορείτε να βρείτε έτοιμο το πρόγραμμα στο συνοδευτικό αρχείο με όνομα *LED\_Fading.abp* και να το ανοίξετε στο περιβάλλον του ArduinoBlock με το κουμπί Άνοιγμα.

Από το περιβάλλον του ArduinoBlock πατάμε το κουμπί *Ανέβασε στο Arduino*. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα και αρχίζει να εκτελείται.



## Εφαρμογή 6: LED ρυθμιζόμενο με ποτενσιόμετρο

Το Arduino UNO διαθέτει 6 ακροδέκτες αναλογικής εισόδου (A0 - A5), στους οποίους μπορούν να συνδέονται διατάξεις που δίνουν ως έξοδο τάσεις από 0 έως 5V. Οι εισοδοί αυτές συνδέονται με έναν μετατροπέα αναλογικού σε ψηφιακό 10 bit, ο οποίος επιστρέφει στο πρόγραμμα μία τιμή από 0 - 1023. Η τιμή 0 αντιστοιχεί σε τάση εισόδου 0V, ενώ η τιμή 1023 αντιστοιχεί σε μία μέγιστη τάση αναφοράς, που μπορεί να ρυθμίζεται ανάλογα με τη συνδεδεμένη διάταξη, αλλά η προκαθορισμένη τιμή της για το Arduino UNO είναι 5V.

Σε αυτή την εφαρμογή, θα αξιοποιήσουμε μία αναλογική είσοδο και μία ψευδοαναλογική έξοδο του Arduino. Η τελική κατασκευή θα περιλαμβάνει ένα περιστροφικό ποτενσιόμετρο και ένα LED. Η φωτεινότητα του LED θα μεταβάλλεται ανάλογα με τη θέση του ποτενσιόμετρου.

### Υλικά

Σε σχέση με τα υλικά της Εφαρμογής 5, θα χρησιμοποιήσουμε επιπλέον ένα περιστροφικό ποτενσιόμετρο και τρία καλώδια. Στη συνέχεια παρουσιάζεται το περιστροφικό ποτενσιόμετρο.

#### Περιστροφικό ποτενσιόμετρο

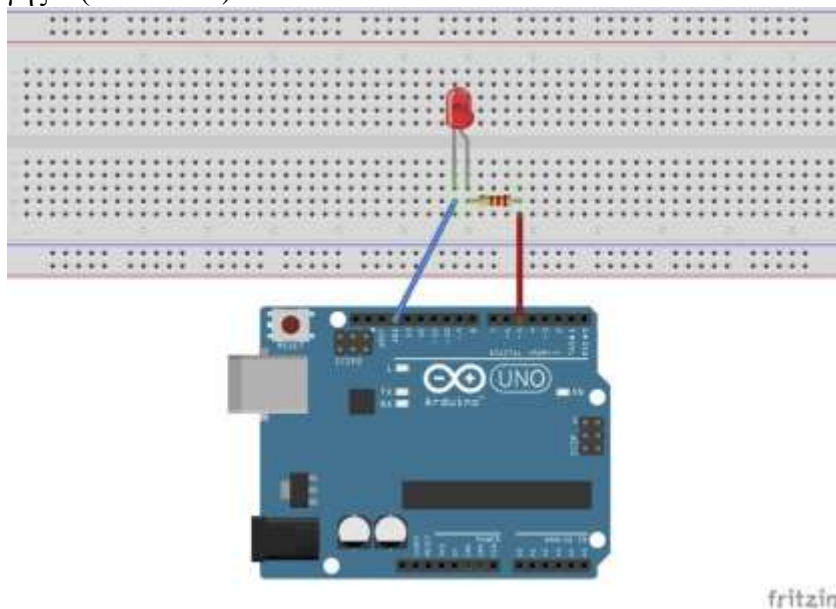
Για τη ρύθμιση της έντασης του LED θα χρησιμοποιήσουμε ένα ποτενσιόμετρο. Όπως φαίνεται στην Εικόνα 35, το περιστροφικό ποτενσιόμετρο διαθέτει έναν περιστρεφόμενο ρυθμιστή και 3 ακροδέκτες. Εσωτερικά, οι 2 ακραίοι ακροδέκτες συνδέονται στα άκρα μίας ωμικής αντίστασης. Ο μεσαίος ακροδέκτης συνδέεται με μία κινητή επαφή, η οποία τρέχει πάνω στην αντίσταση και η θέση της καθορίζεται από τον περιστρεφόμενο ρυθμιστή. Όταν συνδέουμε τους ακριανούς ακροδέκτες τον ένα στη γείωση (GND) και τον άλλο στην τροφοδοσία (5V) (η πολικότητα δεν είναι σημαντική), το ποτενσιόμετρο μπορεί να λειτουργήσει ως ένας μεταβλητός διαιρέτης τάσης, βγάζοντας στο μεσαίο ακροδέκτη τάση από 0 έως 5 Volt, ανάλογα με τη θέση του ρυθμιστή.



Εικόνα 35

## Κύκλωμα

Το κύκλωμα θα περιλαμβάνει ένα LED και ένα ποτενσιόμετρο. Όσον αφορά στη συνδεσμολογία του LED, δεν υπάρχει κάποια διαφοροποίηση σε σχέση με το κύκλωμα της Εφαρμογής 5 (Εικόνα 36).



Εικόνα 36

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η προσθήκη του ποτενσιόμετρου στο κύκλωμα.

### Βήμα 1

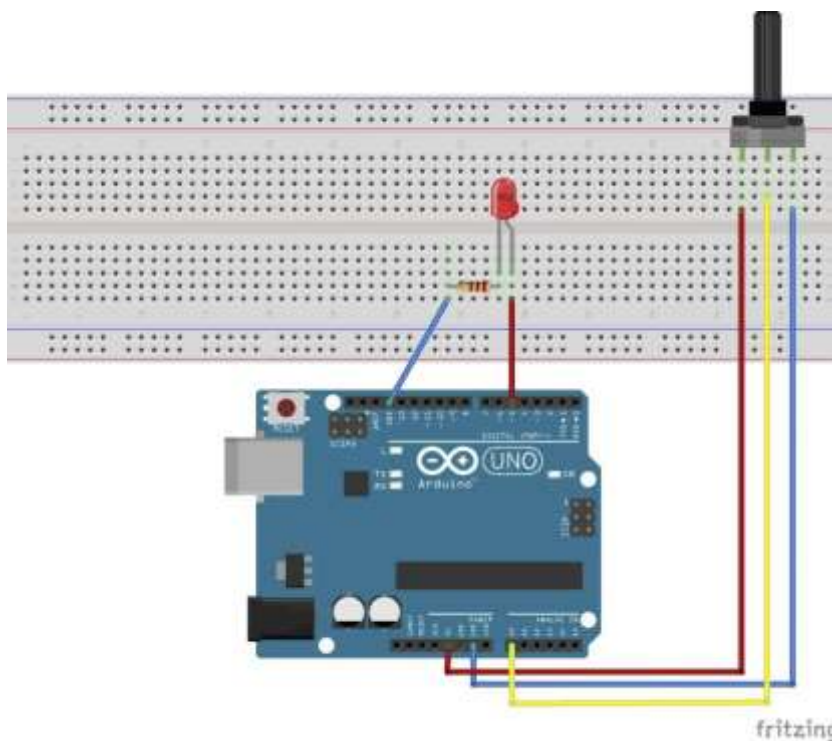
Καρφώνουμε το ποτενσιόμετρο πάνω στο breadboard, με κάθε ακροδέκτη να συνδέεται σε μία ξεχωριστή στήλη.

### Βήμα 2

Παίρνουμε ένα καλώδιο και συνδέουμε το ένα άκρο του στον ακροδέκτη 5V του Arduino και το άλλο στην ίδια μισή στήλη με τον αριστερό ακροδέκτη του ποτενσιόμετρου.

### Βήμα 3

Με ένα ακόμα καλώδιο, συνδέουμε τον ακροδέκτη γείωσης (GND) του Arduino, με τη στήλη στην οποία έχει τοποθετηθεί ο δεξιός ακροδέκτης του ποτενσιόμετρου.



#### Βήμα 4

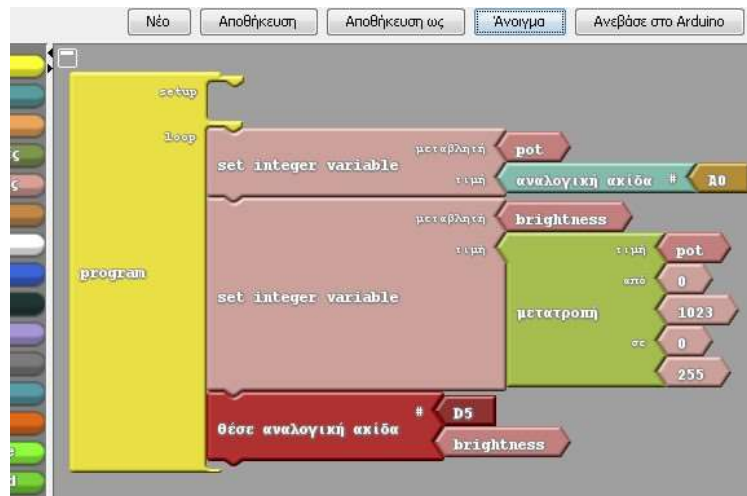
Τέλος, προκειμένου να διαβάσουμε την έξοδο του ποτενσιόμετρου, συνδέουμε με ένα καλώδιο τη στήλη του μεσαίου ακροδέκτη στην αναλογική είσοδο A0 του Arduino.

#### Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα με τον υπολογιστή και εκτελούμε το Arduino IDE. Μέσα από το μενού *Εργαλεία*, ορίζουμε τον τύπο της πλακέτας και τη θύρα σύνδεσης.

#### Πρόγραμμα

Από το μενού εργαλεία του Arduino IDE επιλέγουμε *ArduBlock* και ανοίγει το σχετικό παράθυρο.



Εικόνα 27

Στην Εικόνα 27 φαίνεται ο κώδικας της εφαρμογής. Η ανάγνωση της αναλογικής εισόδου γίνεται στο ArduBlock με την εντολή *αναλογική ακίδα* και η έξοδος PWM με την εντολή *θέσε αναλογική ακίδα*. Και οι δύο αυτές εντολές βρίσκονται στην ομάδα εντολών *Ακίδες*.

Σε κάθε επανάληψη της loop, το πρόγραμμα διαβάζει την τιμή της αναλογικής εισόδου *A0* (0 - 1023) και την αποθηκεύει σε μία μεταβλητή με όνομα *pot* (η σχετική εντολή βρίσκεται στην ομάδα εντολών *Μεταβλητές/σταθερές*). Στη συνέχεια, με την εντολή *μετατροπή* (ομάδα *Μαθηματικοί τελεστές*), απεικονίζει την τιμή της *pot* από το διάστημα 0 - 1023 στο διάστημα 0 - 255, που είναι κατάλληλο για την ψευδοαναλογική έξοδο. Τη νέα τιμή την αποθηκεύει σε μία μεταβλητή με όνομα *brightness*. Τέλος, γράφει αυτή την τιμή της *brightness* στην PWM ακίδα *D5*, μεταβάλλοντας τη φωτεινότητα του LED.

## Εφαρμογή 7: Αισθητήρας φωτός

Στα πλαίσια της εφαρμογής οι μαθητές θα έχουν την ευκαιρία να κατασκευάσουν έναν αισθητήρα φωτός με τη χρήση μίας φωτοαντίστασης. Στη συνέχεια, θα χρησιμοποιήσουν μία από τις αναλογικές εισόδους του Arduino για την ανάγνωση της τιμής του αισθητήρα. Η εφαρμογή που θα υλοποιηθεί, θα ανάβει αυτόματα ένα LED, όταν θα μειώνεται ο φωτισμός του χώρου.

### Υλικά

Τα νέα υλικά που θα εισάγουμε σε αυτή την εφαρμογή είναι μία φωτοαντίσταση και μία αντίσταση 10 kΩ.

### Φωτοαντίσταση

Η φωτοαντίσταση (Εικόνα 28) είναι μία αντίσταση, που η τιμή της αλλάζει ανάλογα με το φως που πέφτει πάνω της. Όσο πιο έντονο το φως, τόσο μικρότερη η τιμή της αντίστασης. Η φωτοαντίσταση κατασκευάζεται από φωτοευαίσθητο υλικό. Αυτό σημαίνει ότι όταν πέφτει φως πάνω στα άτομα αυτού του υλικού, η φωτεινή ενέργεια διεγείρει τα ηλεκτρόνια και κάποια από αυτά διαφεύγουν. Με τον τρόπο αυτό αυξάνεται ο αριθμός των ελεύθερων ηλεκτρονίων, άρα γίνεται ευκολότερη η διέλευση του ρεύματος, δηλαδή μικραίνει η τιμή της αντίστασης. Εντονότερο φως σημαίνει περισσότερη προσφερόμενη ενέργεια, περισσότερα ηλεκτρόνια που διαφεύγουν, ευκολότερη διέλευση ρεύματος και άρα μικρότερη αντίσταση.



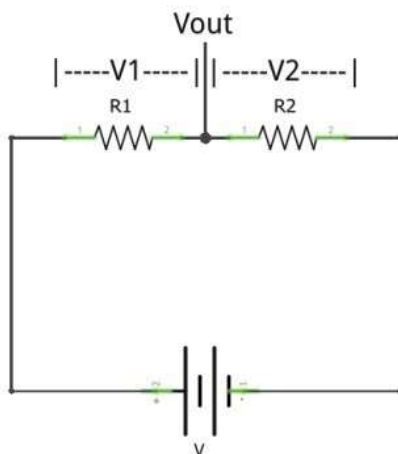
Εικόνα 28 Φωτοαντίσταση

### Αντίσταση 10 kΩ

Η αντίσταση των 10 kΩ θα συνδεθεί σε σειρά με τη φωτοαντίσταση, για την κατασκευή του αισθητήρα φωτός.

## Κύκλωμα

Ένας διαιρέτης τάσης (Εικόνα 29) είναι μία διάταξη που αποτελείται από δύο αντιστάσεις R1 και R2, συνδεδεμένες σε σειρά.

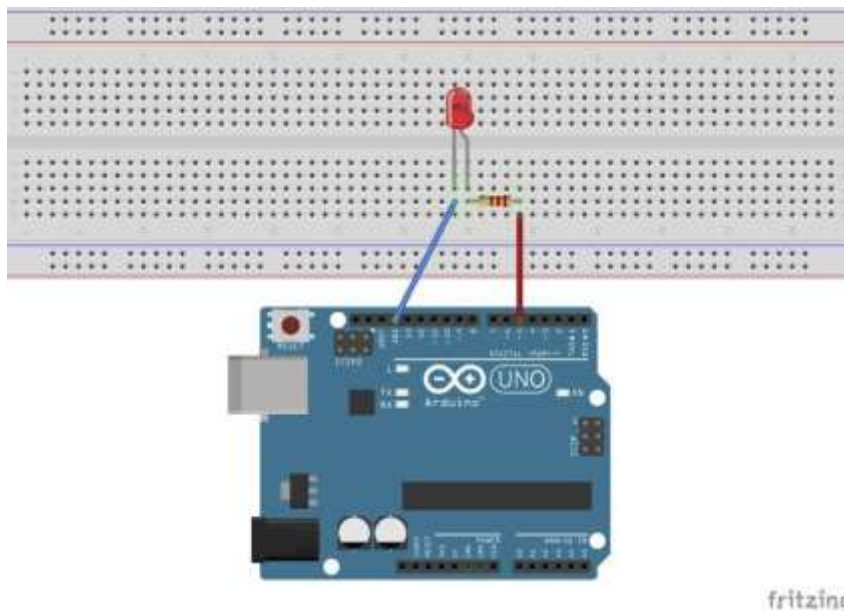


Εικόνα 29

Όταν εφαρμόζουμε μία τάση V στο διαιρέτη τάσης, η πτώση τάσης που παρατηρείται σε κάθε αντίσταση, είναι ανάλογη της τιμής της (νόμος του Ohm). Έτσι έχουμε  $V=V1+V2$  και  $V1/V2 = R1/R2$ . Άρα, η τάση Vout που παίρνουμε ως έξοδο ανάμεσα στις δύο αντιστάσεις, εξαρτάται από τις τιμές τους και είναι μεγαλύτερη από 0V και μικρότερη από την τάση V της πηγής.

Για να φτιάξουμε έναν αισθητήρα φωτός, κατασκευάζουμε ένα διαιρέτη τάσης με μία φωτοαντίσταση και μία αντίσταση 10kΩ. Όταν αλλάζει η ένταση του φωτός, μεταβάλλεται η τιμή της φωτοαντίστασης, άρα αλλάζει ο λόγος των τιμών των δύο αντιστάσεων, αλλάζει η πτώση τάσης που συμβαίνει σε καθεμία από αυτές και άρα μεταβάλλεται η τιμή της τάσης στο σημείο ανάμεσά τους.

Το κύκλωμα θα περιλαμβάνει ένα LED και έναν αισθητήρα φωτός. Όσον αφορά στη συνδεσμολογία του LED, δεν υπάρχει κάποια διαφοροποίηση σε σχέση με το κύκλωμα της Εφαρμογής 5 (Εικόνα 30).



Εικόνα 30

Στη συνέχεια, παρουσιάζεται βήμα προς βήμα η κατασκευή του αισθητήρα φωτός με τη φωτοαντίσταση και την αντίσταση 10kΩ.

### Βήμα 1

Τοποθετούμε τη φωτοαντίσταση πάνω στο breadboard.

### Βήμα 2

Με ένα καλώδιο συνδέουμε το αριστερό ποδαράκι της φωτοαντίστασης με τον ακροδέκτη 5V του Arduino.

### Βήμα 3

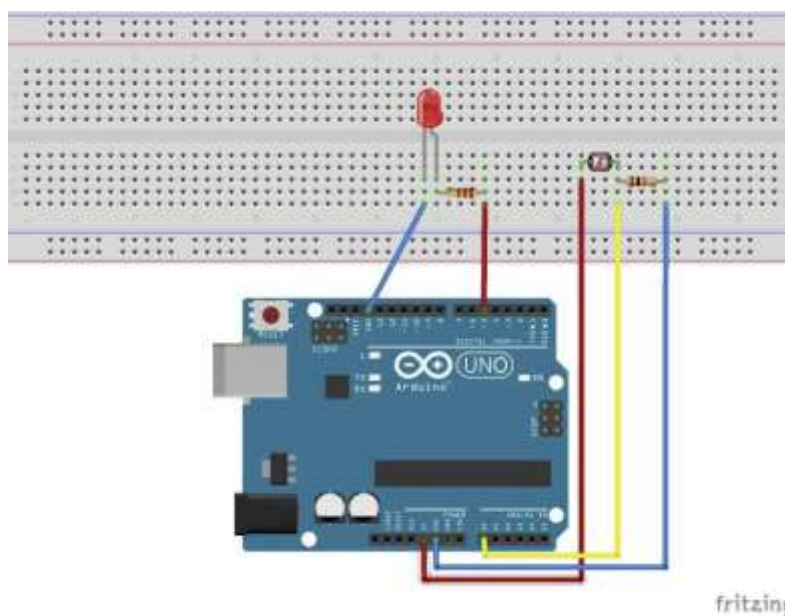
Συνδέουμε το ένα άκρο της αντίστασης 10 kΩ στην ίδια στήλη με το δεξί πόδι της φωτοαντίστασης και το άλλο σε μία κενή στήλη του breadboard.

## Βήμα 4

Με ένα καλώδιο συνδέουμε το ελεύθερο πόδι της αντίστασης 10 kΩ με τον ακροδέκτη GND του Arduino.

## Βήμα 5

Τέλος, με ένα καλώδιο, συνδέουμε τη στήλη στην οποία ενώνονται οι δύο αντιστάσεις, με τον ακροδέκτη A0 του Arduino, προκειμένου να διαβάζουμε την τιμή του αισθητήρα φωτός.



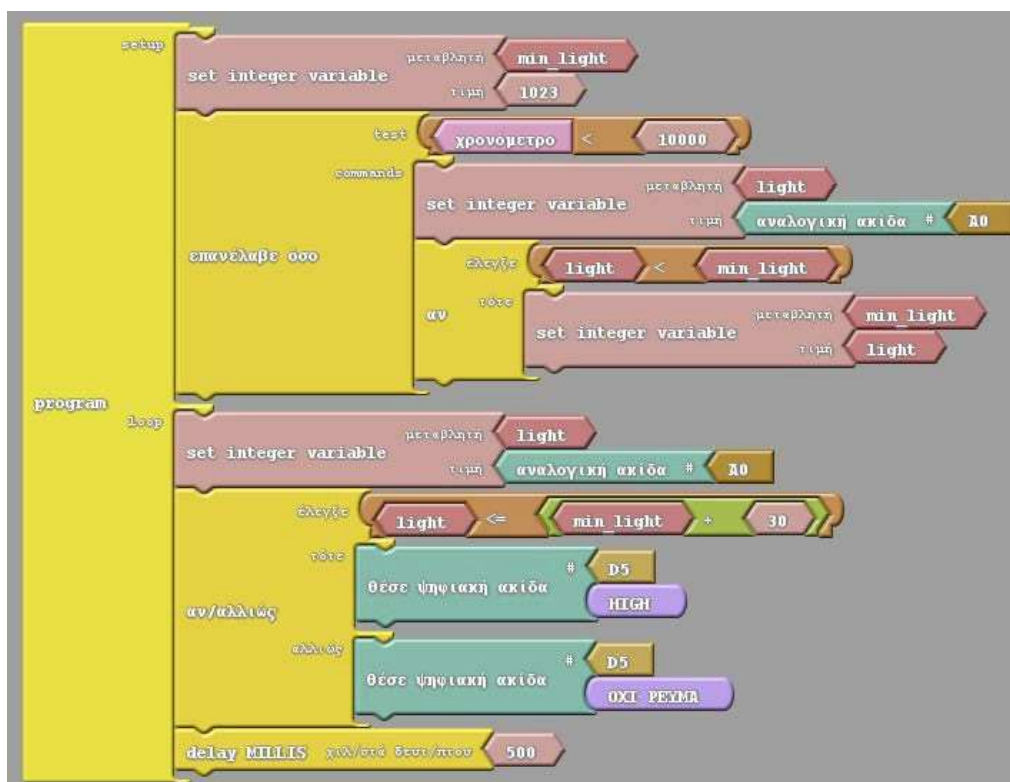


## Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα στον υπολογιστή και εκτελούμε το Arduino IDE. Από το μενού *Εργαλεία*, ορίζουμε τον τύπο της πλακέτας και τη θύρα COM στην οποία έχει συνδεθεί το Arduino.

## Πρόγραμμα

Από το μενού *Εργαλεία* του Arduino IDE επιλέγουμε *ArduBlock*. Στην Εικόνα 41 παρουσιάζεται ο κώδικας του προγράμματος και στη συνέχεια εξηγείται η λειτουργία του.



Εικόνα 41

Η τιμή του αισθητήρα φωτός για την οποία θα θέλουμε να ενεργοποιείται το LED δεν είναι συγκεκριμένη και προκαθορισμένη, αλλά εξαρτάται από πολλούς παράγοντες. Για το λόγο αυτό, το πρόγραμμά, προσπαθεί να καθορίσει με αυτόματο τρόπο το κατώφλι ενεργοποίησης του LED. Αυτό υλοποιείται μέσα στη *setup* με ένα βρόχο (εντολή *επανάλαβε όσο* στην ομάδα εντολών *Έλεγχος*) που εκτελείται για τα πρώτα 10 δευτερόλεπτα της λειτουργίας του προγράμματος. Το πλακίδιο *χρονόμετρο* που χρησιμοποιείται στη συνθήκη του βρόχου, υπάρχει μέσα στην ομάδα *Έλεγχος* και επιστρέφει το χρόνο που έχει περάσει από την εκκίνηση εκτέλεσης του προγράμματος σε milliseconds. Ο τελεστής σύγκρισης *<* υπάρχει μέσα στην ομάδα *Tests*.

Μέσα σε αυτό το βρόχο, με μία εντολή *αν* (ομάδα εντολών *Έλεγχος*), το Arduino ελέγχει συνέχεια την τιμή του αισθητήρα και κρατάει τη χαμηλότερη τιμή φωτισμού που θα ανιχνεύσει, για να υπολογίσει το κατώφλι για το άναμμα του LED. Άρα αρκεί ο χρήστης μέσα σε αυτά τα πρώτα 10 δευτερόλεπτα, να αναπαράγει τις συνθήκες φωτισμού για τις οποίες επιθυμεί να ενεργοποιείται το LED (π.χ. να σβήσει τα φώτα ή να σκεπάσει τη φωτοαντίσταση με το χέρι του).

Από εκεί και πέρα, μέσα στη *loop* διαβάζουμε κάθε φορά την τιμή από τον αισθητήρα και αν είναι χαμηλότερη από το κατώφλι ανάβουμε το LED, σε διαφορετική περίπτωση το σβήνουμε (εντολή *θέσε ψηφιακή ακίδα* από την ομάδα εντολών *Ακίδες*). Ο έλεγχος της τιμής του αισθητήρα και ο ορισμός της κατάστασης του LED, υλοποιούνται με μία εντολή *αν/αλλιώς* (ομάδα εντολών *Έλεγχος*). Το κατώφλι ενεργοποίησης του LED υπολογίζεται, προσθέτοντας (τελεστής + από την ομάδα εντολών *Μαθηματικοί Τελεστές*) στην ελάχιστη τιμή φωτισμού που ανιχνεύθηκε κατά την πρώτη φάση, ένα σταθερό αριθμό (εδώ 30, ομάδα εντολών *Μεταβλητές/σταθερές*). Η ανοχή που παρέχει αυτός ο υπολογισμός είναι απαραίτητη, καθώς η τιμή του αισθητήρα επηρεάζεται ακόμα και από το φως του LED, το οποίο στα πρώτα 10 δευτερόλεπτα είναι σβηστό.

Από το περιβάλλον του ArduBlock πατάμε το κουμπί *Ανέβασε στο Arduino*. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα και αρχίζει να εκτελείται. Μέσα στα πρώτα 10 δευτερόλεπτα χαμηλώστε το φωτισμό (π.χ. σβήστε τα φώτα), ώστε να υπολογιστεί σωστά η τιμή του αισθητήρα για το κατώφλι. Μετά τα 10 δευτερόλεπτα αλλάξτε τις συνθήκες φωτισμού, για να κάνετε το LED να ανάψει και να σβήσει.

Αν η εφαρμογή δεν λειτουργεί σωστά, προσπαθήστε να αυξήσετε την ανοχή, αυξάνοντας τη σταθερά που προστίθεται στην ελάχιστη τιμή κατά τον υπολογισμό του κατωφλίου. Αν για οποιοδήποτε λόγο θέλετε να υπολογιστεί από την αρχή το κατώφλι ενεργοποίησης, διακόψτε προσωρινά την τροφοδοσία του Arduino ή πατήστε το κουμπί *reset*. Το πρόγραμμα θα εκτελεστεί από την αρχή, ξεκινώντας και πάλι με τη φάση ανίχνευσης της ελάχιστης τιμής φωτισμού.

## Εφαρμογή 8: Αισθητήρας υπερήχων HC-SR04

Μέσα από την εφαρμογή θα παρουσιαστεί ο αισθητήρας απόστασης HC-SR04 και το active buzzer. Η τελική εφαρμογή θα είναι ένα σύστημα ανίχνευσης εμποδίων. Όταν η απόσταση από το κοντινότερο εμπόδιο είναι μικρότερη από 50 cm τότε θα ανάβει ένα LED. Αν η απόσταση είναι μικρότερη από 20 cm, τότε μαζί με το άναμμα του LED θα παράγεται και ένας ήχος από το active buzzer.

### Υλικά

Στην συνέχεια παρουσιάζονται τα υλικά που θα χρησιμοποιήσουμε για πρώτη φορά σε αυτή την εφαρμογή.

### Αισθητήρας HC-SR04

Τι είναι

Ο HC-SR04 είναι ένας αισθητήρας απόστασης υπερήχων. Όπως φαίνεται και στην Εικόνα 42, διαθέτει έναν πομπό και ένα δέκτη υπερήχων, καθώς και 4 ακροδέκτες. Οι δύο ακριανοί ακροδέκτες VCC και GND, χρησιμοποιούνται για την τροφοδοσία του αισθητήρα και συνδέονται στην τάση (5V) και τη γείωση αντίστοιχα. Ο ακροδέκτης Trig χρησιμοποιείται για την εκκίνηση της διαδικασίας μέτρησης και ο ακροδέκτης Echo χρησιμοποιείται για την έξοδο του αποτελέσματος. Οι δύο αυτοί ακροδέκτες συνδέονται σε δύο ψηφιακές ακίδες του Arduino.



Εικόνα 42

### Πώς λειτουργεί

Για να ξεκινήσει η διαδικασία της μέτρησης, πρέπει να στείλουμε στον ακροδέκτη Trig έναν παλμό High με διάρκεια τουλάχιστον 10  $\mu$ sec. Μόλις ο αισθητήρας λάβει το σήμα ενεργοποίησης, στέλνει από τον πομπό μια ακολουθία υπερήχων. Όταν οι υπερήχοι συναντήσουν κάποιο εμπόδιο αντανακλώνται και επιστρέφουν προς τον αισθητήρα, όπου και ανιχνεύονται από το δέκτη. Στη συνέχεια, ο αισθητήρας βγάζει ως έξοδο στον

ακροδέκτη Echo έναν παλμό HIGH. Η διάρκεια του παλμού είναι ίση με το χρόνο που πέρασε από τη στιγμή της εκπομπής των υπερήχων, μέχρι τη λήψη της αντανάκλασης.

Πως υπολογίζεται η απόσταση από το εμπόδιο

Το Arduino με μία κατάλληλη συνάρτηση μετράει τη διάρκεια του παλμού που βγάζει ως έξοδο ο αισθητήρας, έστω *duration*. Με δεδομένο ότι οι υπέρηχοι ταξιδεύουν με την ταχύτητα του ήχου ( $340\text{m/s} = 0,034\text{cm}/\mu\text{s}$ ) και με βάση τον τύπο της ταχύτητας ( $u=s/t$ ), αν *distance* είναι η απόσταση από το εμπόδιο έχουμε:

$$0,034 = \frac{\text{distance} * 2}{\text{duration}} \Leftrightarrow = \frac{0,034}{2} * \approx \frac{\quad}{59}$$

Η διαίρεση με το 2, προκύπτει από το γεγονός ότι η διάρκεια του παλμού αντιστοιχεί στο χρόνο που έκαναν οι υπέρηχοι να πάνε μέχρι το εμπόδιο και να γυρίσουν πίσω στον αισθητήρα. Άρα η απόσταση που καλύπτουν οι υπέρηχοι σε αυτό το χρόνο, είναι η διπλάσια από αυτήν που θέλουμε να υπολογίσουμε.

### Active Buzzer

Τα buzzer (Εικόνα 43) είναι συσκευές που παράγουν ηχητικά σήματα. Υπάρχουν δύο κατηγορίες buzzer, τα active και τα passive. Τα active buzzer διαθέτουν εσωτερικό ταλαντωτή και όταν τροφοδοτούνται με συνεχή τάση, παρέχουν ένα ηχητικό σήμα συγκεκριμένης συχνότητας. Αντίθετα, τα passive buzzer δεν διαθέτουν εσωτερικό ταλαντωτή και για να παράγουν ήχο, πρέπει η τάση τροφοδοσίας τους να μεταβάλλεται (π.χ. HIGH· LOW· HIGH· LOW ...). Η συχνότητα του ήχου που παράγει ένα passive buzzer είναι ίση με τη συχνότητα με την οποία μεταβάλλεται η τάση τροφοδοσίας του και άρα μπορεί να αλλάξει.



Εικόνα 43 Buzzer

Στην εφαρμογή μας θα χρησιμοποιήσουμε ένα active buzzer, το οποίο θα παράγει μία ηχητική ειδοποίηση όταν η απόσταση από το κοντινότερο εμπόδιο θα είναι μικρότερη από 20 cm.

## Αντίσταση 100 Ω

Η αντίσταση των 100 Ω θα συνδεθεί σε σειρά με το buzzer για τον περιορισμό του ρεύματος.



Εικόνα 44 Αντίσταση 100Ω

## Κύκλωμα

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος.

### Βήμα 1

Συνδέουμε ένα καλώδιο από το pin 5V του Arduino στην πρώτη οπή της οριζόντιας σειράς του breadboard με την κόκκινη γραμμή, ώστε να δώσουμε 5V σε όλη τη σειρά.

### Βήμα 2

Συνδέουμε ένα καλώδιο από το pin GND στη δεύτερη οπή της οριζόντιας σειράς του breadboard με την μπλε γραμμή, ώστε να μπορούμε να πάρουμε γείωση από όλη τη σειρά.

### Βήμα 3

Καρφώνουμε τον HC-SR04 στην πάνω σειρά του άνω τμήματος του breadboard και στραμμένο προς τα έξω, όπως φαίνεται στο σχήμα.

**Προσοχή:** Προκειμένου να είναι πιο ευδιάκριτες οι συνδέσεις των ακροδεκτών του αισθητήρα, στα επόμενα βήματα ο HC-SR04 παρουσιάζεται τοποθετημένος στη μέση της άνω πλευράς του breadboard και οι συνδέσεις γίνονται μπροστά από τον αισθητήρα.

### Βήμα 4

Με ένα καλώδιο συνδέουμε τον ακροδέκτη VCC του αισθητήρα στη σειρά του breadboard όπου έχουμε οδηγήσει την τροφοδοσία 5V από το Arduino.

## Βήμα 5

Με παρόμοιο τρόπο συνδέουμε τον ακροδέκτη GND του αισθητήρα στη σειρά όπου έχουμε οδηγήσει τη γείωση από το Arduino.

## Βήμα 6

Συνδέουμε τον ακροδέκτη Trig με τον ψηφιακό ακροδέκτη 6 του Arduino.

## Βήμα 7

Συνδέουμε τον ακροδέκτη Echo του αισθητήρα με τον ψηφιακό ακροδέκτη 7 του Arduino.

## Βήμα 8

Τοποθετούμε το LED πάνω στο breadboard, με την άνοδο (μακρύ ποδαράκι) στραμμένη προς τα δεξιά.

## Βήμα 9

Συνδέουμε την άνοδο του LED στον ψηφιακό ακροδέκτη 8 του Arduino.

## Βήμα 10

Συνδέουμε το ένα άκρο της αντίστασης των 220 Ω στην ίδια στήλη με την κάθοδο του LED και το άλλο της άκρο στη σειρά του breadboard όπου συνδέεται το GND του Arduino. Με τον τρόπο αυτό το LED συνδέεται στη γείωση μέσω της αντίστασης.

## Βήμα 11

Τοποθετούμε το buzzer πάνω στο breadboard με το θετικό ακροδέκτη στη δεξιά πλευρά. Το buzzer που διαθέτετε δεν έχει συνδεδεμένα καλώδια και θα πρέπει να το καρφώσετε επάνω στο breadboard. Ο θετικός ακροδέκτης του active buzzer, είναι στην πλευρά που υπάρχει το + στην επιφάνειά του.

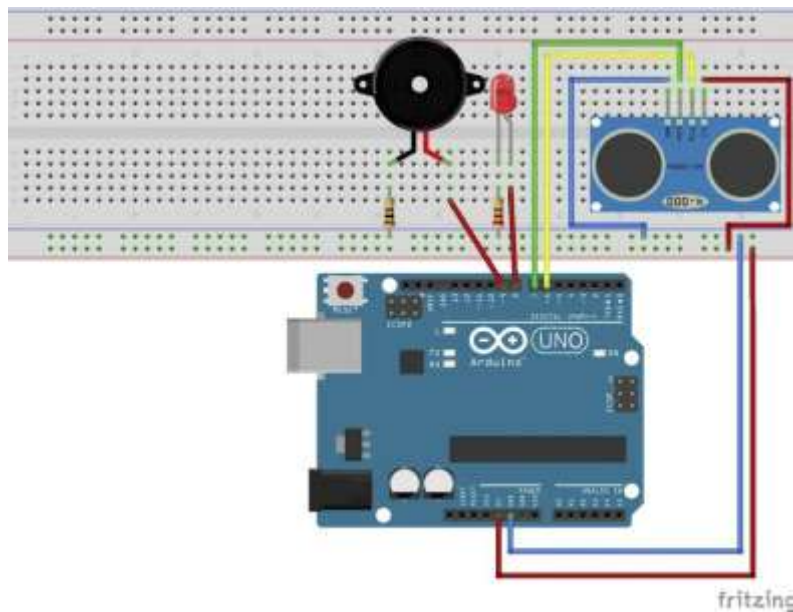
## Βήμα 12

Καρφώνουμε το ένα άκρο ενός καλωδίου στη στήλη όπου έχει συνδεθεί ο θετικός

ακροδέκτης του buzzer και το άλλο στον ψηφιακό ακροδέκτη 9 του Arduino.

### Βήμα 13

Συνδέουμε το ένα άκρο της αντίστασης των 100 Ω στην ίδια στήλη με τον αρνητικό ακροδέκτη του buzzer και το άλλο της άκρο στη σειρά του breadboard όπου συνδέεται το GND του Arduino. Αν δεν υπάρχει διαθέσιμη αντίσταση 100 Ω, χρησιμοποιήστε ένα καλώδιο για να συνδέσετε τον αρνητικό ακροδέκτη του buzzer απευθείας στη γείωση.

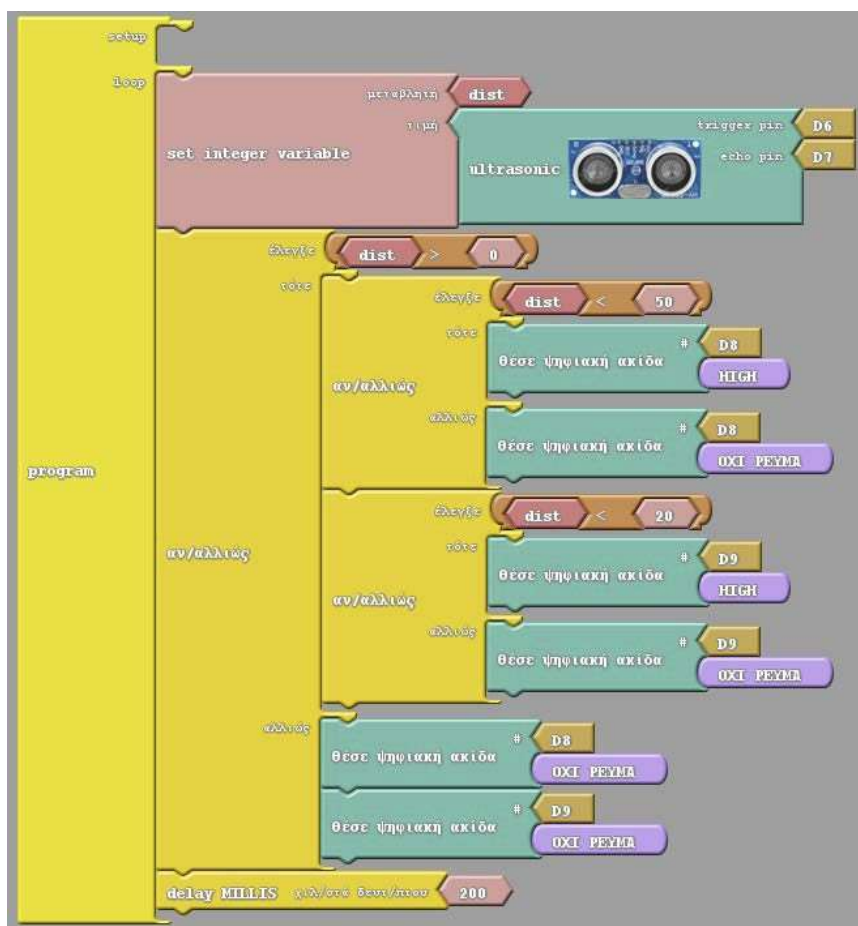


## Σύνδεση πλακέτας με τον υπολογιστή

Συνδέουμε την πλακέτα με τον υπολογιστή. Στη συνέχεια, τρέχουμε το Arduino IDE και από το μενού *Εργαλεία* ορίζουμε τον τύπο της πλακέτας και τη θύρα στην οποία έχει συνδεθεί το Arduino.

## Πρόγραμμα

Από το μενού *Εργαλεία* του Arduino IDE επιλέγουμε *ArduBlock*. Στην Εικόνα 45 παρουσιάζεται ο κώδικας του προγράμματος και στη συνέχεια εξηγείται η λειτουργία του.



Εικόνα 45

Σε κάθε επανάληψη της *loop* παίρνουμε μία μέτρηση από τον αισθητήρα απόστασης και την αποθηκεύουμε σε μία μεταβλητή με όνομα *dist*. Ολόκληρη η διαδικασία της αποστολής του παλμού ενεργοποίησης στον ακροδέκτη Trig, της μέτρησης της διάρκειας του παλμού εξόδου από τον ακροδέκτη Echo και του υπολογισμού της απόστασης σε εκατοστά, γίνεται στο ArduBlock με το πλακίδιο *ultrasonic*, το οποίο υπάρχει μέσα στην ομάδα πλακιδίων *Generic Hardware*.



Επειδή ο αισθητήρας HC-SR04 δεν είναι ιδιαίτερα αξιόπιστος, το πλακίδιο *ultrasonic* επιστρέφει την τιμή 0 αν η μέτρηση που θα διαβάσει από τον αισθητήρα είναι εκτός ορίων. Για το λόγο αυτό, μετά τη λήψη της μέτρησης ελέγχουμε αν η τιμή είναι αποδεκτή ( $>0$ ).

Αν η τιμή είναι αποδεκτή, προχωρούμε σε περισσότερους ελέγχους. Αρχικά συγκρίνουμε την απόσταση που μετρήθηκε με τα 50 cm. Αν είναι μικρότερη, τότε ανάβουμε το LED γράφοντας HIGH στην ακίδα 8 όπου το έχουμε συνδέσει, αλλιώς το σβήνουμε. Στη συνέχεια συγκρίνουμε και πάλι την τιμή της απόστασης με το 20 αυτή τη φορά. Αν η απόσταση είναι μικρότερη από 20 cm, δίνουμε HIGH στον ακροδέκτη 9 ενεργοποιώντας το buzzer. Σε αντίθετη περίπτωση σταματάμε την τροφοδοσία του buzzer κάνοντάς το να σιγήσει.

Αν η τιμή που διαβάσαμε είναι 0, τότε είτε η απόσταση από το κοντινότερο εμπόδιο είναι πολύ μεγάλη, είτε ο αισθητήρας για κάποιο λόγο πήρε μία εσφαλμένη μέτρηση. Σε αυτή την περίπτωση απενεργοποιούμε τόσο το LED όσο και το buzzer.

Στο τέλος κάθε επανάληψης υπάρχει μία χρονοκαθυστέρηση 200ms, προκειμένου να δώσουμε λίγο χρόνο στον αισθητήρα, πριν από την επόμενη μέτρηση.

Από το περιβάλλον του ArduBlock πατάμε το κουμπί **Ανέβασε στο Arduino**. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα και αρχίζει να εκτελείται. Πάρτε ένα βιβλίο και τοποθετήστε το μπροστά από τον αισθητήρα. Πλησιάστε το και απομακρύνετέ το από τον αισθητήρα, για να προκαλέσετε την ενεργοποίηση και απενεργοποίηση του LED και του buzzer.