

ΔΗΜΙΟΥΡΓΩ ΜΕ ΤΟ ARDUINO ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΖΩ ΜΕ ΤΟ ARDUBLOCK



Δραστηριότητες για το Δημοτικό

2^η Έκδοση

Γεωργιτζίκη Ναταλία Θεόκλεια
Εκπαιδευτικός Πληροφορικής, ΠΕ86

ISBN:978-618-00-1324-5

Λίγα λόγια για το βιβλίο:

Σε αυτό το βιβλίο θα βρείτε πληροφορίες για τον μικροελεγκτή arduino και τα εξαρτήματα που χρειαζόμαστε για να μπορέσουμε να υλοποιήσουμε δραστηριότητες αυτοματισμού και ρομποτικής.

Μέσα από απλά παραδείγματα θα γνωρίσουμε τις εντολές του προγραμματιστικού περιβάλλοντος Ardublock με το οποίο προγραμματίζουμε το Arduino και θα μάθουμε να δημιουργούμε εύκολα τα δικά μας προγράμματα.

Τα υλικά που θα χρειαστούμε, μπορούμε να τα προμηθευτούμε από πολλά καταστήματα στην Ελλάδα. Υπάρχουν έτοιμα πακέτα (Kit) αλλά μπορούμε να προμηθευτούμε και μεμονωμένα τα υλικά μας.

Επειδή υπάρχει πληθώρα υλικών, αισθητήρων, εξαρτημάτων κτλ, καλό είναι να κάνετε πρώτα μια λίστα με αυτά που θα χρειαστείτε δεδομένου ότι γνωρίζετε τις δραστηριότητες που θα υλοποιήσετε. Στο βιβλίο αυτό υπάρχουν μερικές δραστηριότητες που έχω υλοποιήσει με τους μαθητές μου αλλά μπορείτε να χρησιμοποιήσετε και εσείς άλλους αισθητήρες και να δημιουργήσετε τις δικές σας δραστηριότητες προσαρμοσμένες στο σχολείο σας, στις ανάγκες των μαθητών σας και την ηλικιακή τους ομάδα και στην οικονομική δυνατότητά σας.

Εξερευνώντας το διαδίκτυο μπορείτε να βρείτε πάρα πολλές δραστηριότητες και projects που μπορείτε να δημιουργήσετε ή να πάρετε ιδέες για να δημιουργήσετε ένα δικό σας.

Μέσα από την εμπειρία μου, ορισμένες από τις παρακάτω δραστηριότητες μπορούν να υλοποιηθούν από μαθητές Γ' και Δ' Δημοτικού. Η τοποθέτηση και η σύνδεση των εξαρτημάτων είναι σχετικά απλή ιδιαίτερα στις πρώτες δραστηριότητες και μπορεί στη συνέχεια να γίνει έλεγχος από τον εκπαιδευτικό για τυχόν λάθος σύνδεση. Ο προγραμματισμός τους επίσης είναι σχετικά απλός ή μπορείτε να δώσετε έτοιμο τον κώδικα και οι μικροί μαθητές να τον αντιγράψουν.

Οι πρώτες δραστηριότητες μπορούν να υλοποιηθούν σε μια διδακτική ώρα αρκεί οι μαθητές να έχουν εξοικειωθεί με τα υλικά και τα εξαρτήματα αλλά και το προγραμματιστικό περιβάλλον ενώ για τις επόμενες απαιτούνται περισσότερες από μια διδακτικές ώρες.

Για τις συγκεκριμένες δραστηριότητες χρησιμοποιείται η έκδοση 1.6.0 του arduino και η έκδοση 201310310 του ardublock.

Για να κατεβάσετε τα προγράμματα επισκεφτείτε τις παρακάτω διευθύνσεις.

<https://www.arduino.cc/en/main/software> (arduino)

<https://sourceforge.net/projects/ardublock/> (ardublock)

2^η έκδοση:

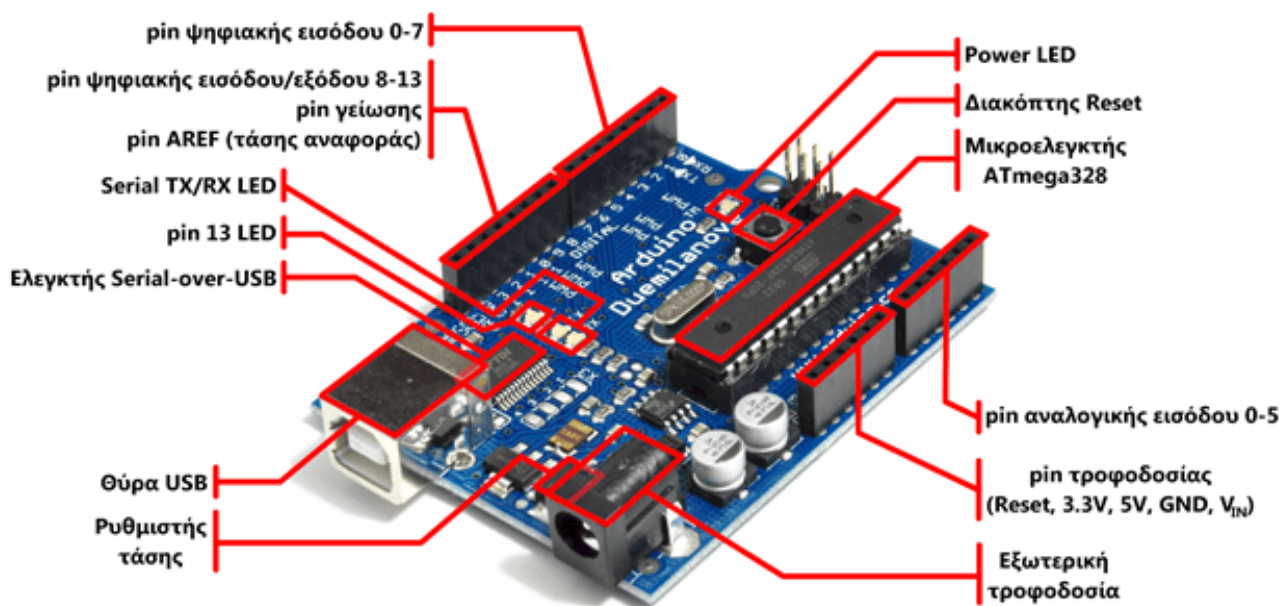
Σε αυτή την έκδοση θα βρείτε περισσότερες δραστηριότητες και έχουν γίνει ορισμένες διορθώσεις στον κώδικα. Επιπλέον, υπάρχουν περισσότερες εικόνες από τις δραστηριότητες.

Περιεχόμενα

Τι είναι το Arduino;	4
Αναλογικές είσοδοι.....	5
Εξαρτήματα – υλικά που χρειαζόμαστε	6
Τι είναι το Ardublock.....	11
Περιβάλλον εργασίας Ardublock- ελληνικά	13
1 ^η Δραστηριότητα: Φωτάκια- Leds που αναβοσβήνουν	17
2 ^η Δραστηριότητα: Αυτόματο φωτιστικό	20
3 ^η Δραστηριότητα: Αλλαγή φωτεινότητας Led.....	22
4 ^η Δραστηριότητα: Χρησιμοποιώντας σερβοκινητήρες- Χαιρετώντας τον κόσμο	23
5 ^η Δραστηριότητα: Σύστημα συναγερμού ελέγχου απόστασης	26
6 ^η Δραστηριότητα: Εμφάνιση μηνυμάτων- απόσταση από αντικείμενο.....	28
7 ^η Δραστηριότητα: Ανιχνευτής κίνησης- σύστημα συναγερμού.....	30
8 ^η Δραστηριότητα: Αυτόματη ράμπα σε διάβαση τρένου	32
9 ^η Δραστηριότητα: Αυτόματη τραμπάλα σε παιδική χαρά που λειτουργεί με το άνοιγμα και το κλείσιμο ενός διακόπτη On-Off	34
10 ^η Δραστηριότητα: Διακόπτης πίεσης (Push button) για τη λειτουργία led	36
11 ^η Δραστηριότητα: Διακόπτης λειτουργίας led (Push button) σαν On/Off διακόπτης.....	37
12 ^η Δραστηριότητα: Σύνδεση κινητήρα- αυξομείωση έντασης περιστροφής.....	39
13 ^η Δραστηριότητα: Αυτόματο ρομποτικό όχημα που κινείται ευθεία και σταματά όταν βρει εμπόδιο.	43
14 ^η Δραστηριότητα: Αυτόματο ρομποτικό όχημα που αλλάζει πορεία όταν συναντήσει εμπόδιο.....	47
15 ^η Δραστηριότητα: Αυτόνομο όχημα που κινείται συνεχώς πάνω σε μια μαύρη γραμμή.....	51
16 ^η Δραστηριότητα: Όχημα που δεν πέφτει από το τραπέζι ή παραμένει μέσα σε μια κλειστή περιοχή.....	55
Επίλογος.....	57

Τι είναι το Arduino;

Ο Arduino είναι μία πλακέτα “ανοικτού κώδικα” την οποία κάποιος, χωρίς ιδιαίτερες γνώσεις προγραμματιστή, μπορεί εύκολα να χρησιμοποιήσει και να κατασκευάσει εφαρμογές ρομποτικής και συστήματα αυτοματισμού.



Ο Arduino αποτελείται από έναν μικροεπεξεργαστή, τον ATmega της Atmel και έχει την δυνατότητα να δεχθεί μονάδες εισόδου / εξόδου. Οι μονάδες εισόδου / εξόδου χωρίζονται σε Ψηφιακές & Αναλογικές. Υπάρχει μεγάλη πληθώρα συσκευών, συμβατές με τις πλακέτες Arduino. Κάποιες από αυτές είναι: Αισθητήρες θερμοκρασίας, υγρασίας, δύναμης, απόστασης, γυροσκόπια, αεζερόμετρα, πίεσης κ.α. Επίσης με τον Arduino μπορείτε να ελέγχετε μοτοράκια DC, βηματικά (stepper) και servo (κινητήρες που περιστρέφονται) ακόμη και leds, φώτα (220v), ρελέ, κινητήρες και ό,τι άλλο μπορείτε να σκεφτείτε ώστε να δημιουργήσετε το δικό σας project.

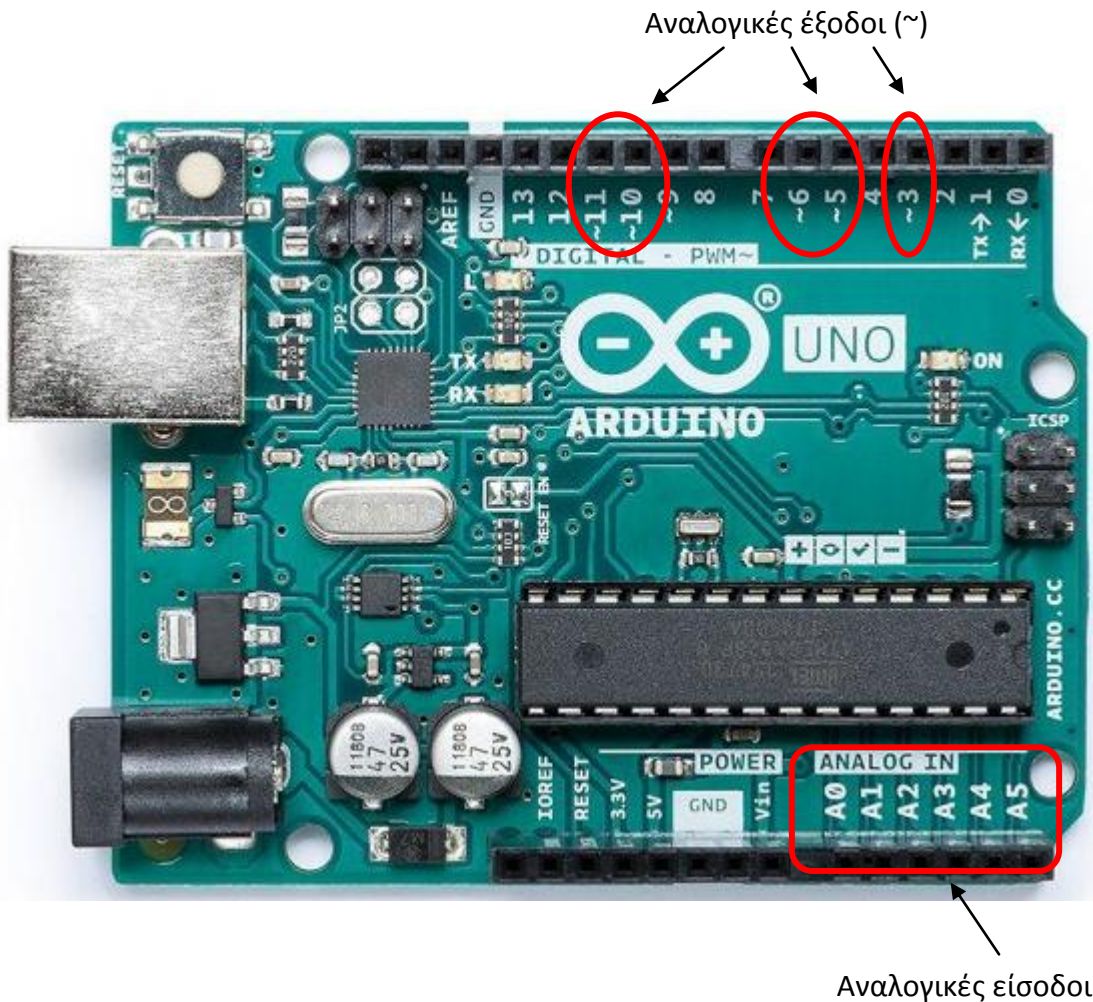
Το Arduino μπορείτε να το προγραμματίσετε από τον υπολογιστή σας μέσω της σειριακής θύρας που υποστηρίζει ο μικροεπεξεργαστής ATmega από την θύρα USB του υπολογιστή. Η σειριακή αυτή σύνδεση (Serial over Usb) χρησιμοποιείται για την μεταφορά προγραμμάτων από τον υπολογιστή προς την πλακέτα Arduino αλλά και το αντίστροφο για την μεταφορά των δεδομένων, που λαμβάνει ο Arduino από τις συσκευές, προς τον υπολογιστή.

Στην αριστερή πλευρά του Arduino είναι τοποθετημένες 14 θύρες (Pins) αριθμημένες από το 0 ως το 13 που μπορούν να λειτουργούν ως ψηφιακές είσοδοι και εξοδοι και ορισμένες από αυτές (έχουν το σήμα ∞ πάνω τους) μπορούν να χρησιμοποιηθούν και ως αναλογικές εξοδοι. Ο μικροελεγκτής δίνει ρεύμα 5V σε κάθε pin.

Οι ψηφιακές εξοδοι δίνουν 2 διακριτές τιμές, τη HIGH (έχει ρεύμα) και τη LOW (δεν έχει ρεύμα). Έτσι, για παράδειγμα αν θέλουμε ένα led να έχει ρεύμα θα πρέπει να το συνδέσουμε σε ένα Pin πχ στο 13 και να θέσουμε αυτό το pin σε τιμή HIGH με την κατάλληλη εντολή της γλώσσας προγραμματισμού που θα χρησιμοποιήσουμε. Υπάρχουν διάφορες γλώσσες προγραμματισμού με τις οποίες μπορούμε να προγραμματίσουμε το arduino αλλά όλες μετατρέπουν οποιοδήποτε πρόγραμμα που δημιουργούμε σε

γλώσσα που καταλαβαίνει το arduino. Εμείς, στις δραστηριότητές μας, θα δουλέψουμε με το προγραμματιστικό περιβάλλον Ardublock το οποίο θα παρουσιάσω στη συνέχεια.

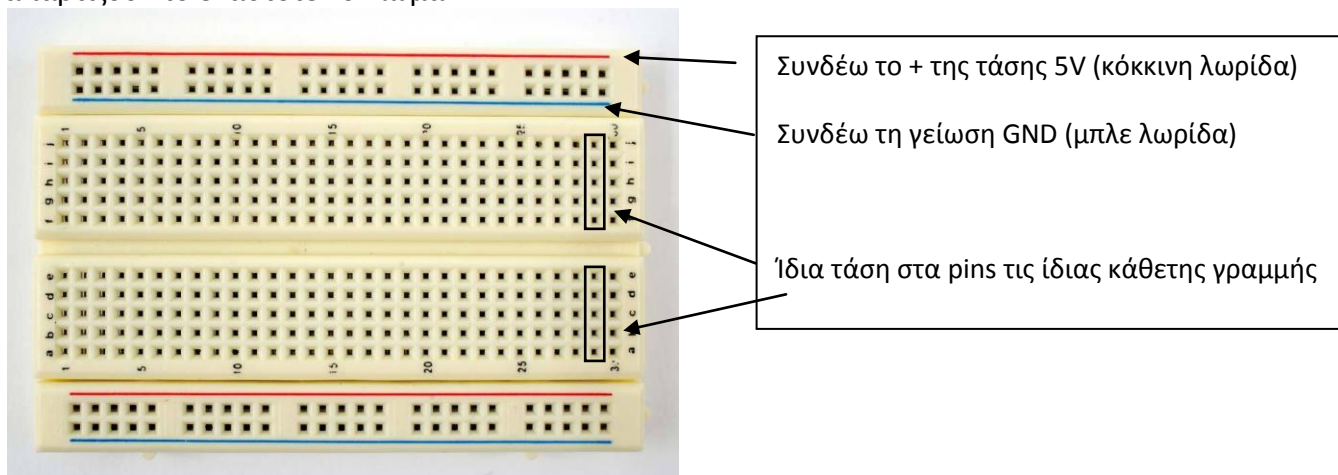
Στη δεξιά πλευρά του Arduino, με τη σήμανση ANALOG IN, βρίσκονται ακόμα 6 ακίδες- pins αριθμημένα από το 0 έως το 5. Αυτά λειτουργούν ως αναλογικές εισοδοι και έτσι μπορούμε πχ να τροφοδοτούμε ένα από αυτά τα pins με μια τάση που να αυξομειώνεται από τα 0V ως τα 5V, με τη χρήση ενός ποτενσιόμετρου.



Εξαρτήματα – υλικά που χρειαζόμαστε

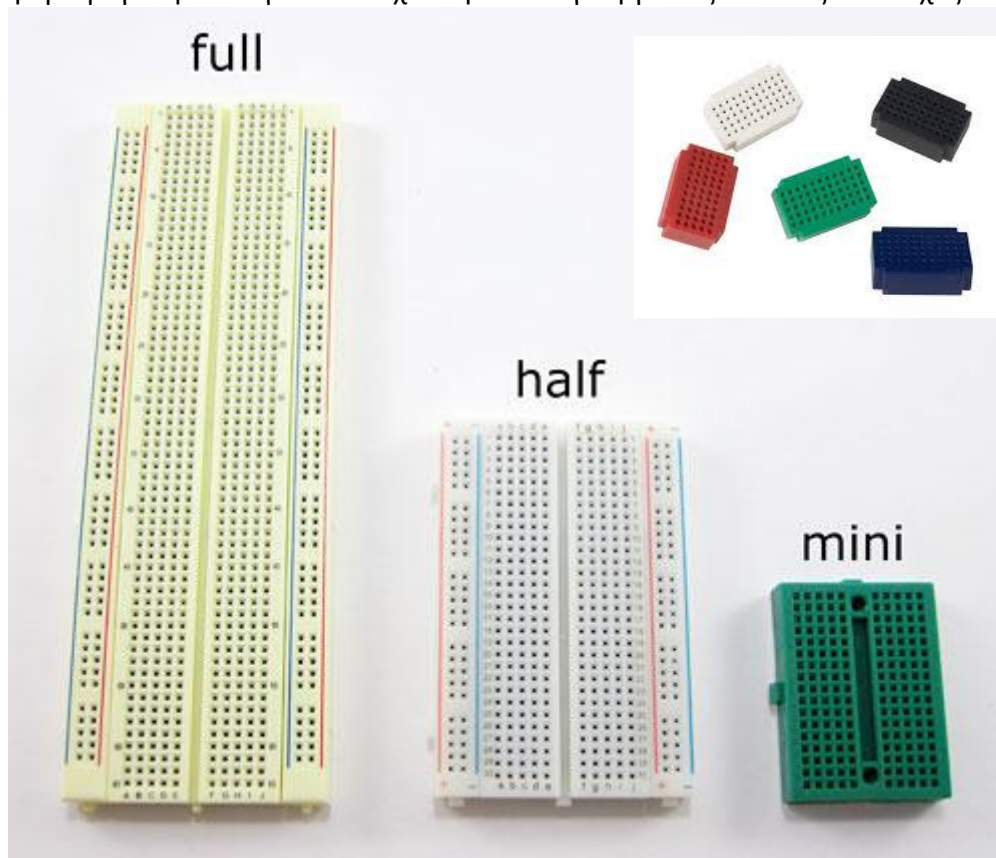
Το breadboard

Το breadboard είναι μια μονάδα για την κατασκευή προσωρινών κυκλωμάτων (πρωτοτύπων). Το breadboard αποτελείται από μια επιφάνεια γεμάτη με τρύπες οι οποίες είναι μεταξύ τους συνδεδεμένες ανά πέντε. Στις τρύπες αυτές τοποθετούνται οι ακροδέκτες των εξαρτημάτων που θα απαρτίζουν το εκάστοτε κύκλωμα.



Όταν τοποθετηθεί ένα εξάρτημα σε μία υποδοχή-τρύπα, τότε είναι ηλεκτρικά συνδεδεμένο σε σειρά με οτιδήποτε άλλο τοποθετηθεί στην ίδια **κάθετη** λωρίδα. Αυτό οφείλεται στις μεταλλικές γραμμές που είναι αγωγίμες και επιτρέπουν στο ρεύμα να ρέει σε κάθε σημείο τους.

Όπως μπορείτε να παρατηρήσετε σε κάθε μεταλλική λωρίδα υπάρχουν μόνο 5 κλιπς. Αυτό ισχύει σε κάθε breadboard. Αυτό σημαίνει ότι μπορείτε να συνδέσετε μέχρι 5 εξαρτήματα σε κάθε λωρίδα. Υπάρχουν διάφορα μεγέθη αλλά για όλα ισχύει η ίδια λογική για τις κάθετες υποδοχές.



Μεγέθη breadboard

Τα leds- φωτοδιόδοι

Τα led- φωτοδιόδοι είναι τα μικρά φωτάκια του κυκλώματός μας! Επειδή στα δικά μας κυκλώματα η τάση τροφοδοσίας είναι 5 Volt, δεν μπορούμε να συνδέσουμε κατ' ευθείαν ένα LED στην τάση αυτή προκειμένου να το ανάψουμε, διότι αυτό θα υπερφορτωθεί και θα καεί (αμέσως ή σιγά-σιγά) (αν δείτε ένα κίτρινο LED να βγάζει κόκκινο φως, και προλάβετε, σβήστε αμέσως το διακόπτη του τροφοδοτικού και ελέγξτε το κύκλωμά σας: μάλλον το έχετε υπερφορτώσει). Για να ανάψουμε σωστά λοιπόν ένα LED, συνδέουμε σε αυτό μια κατάλληλη αντίσταση.

Προσέξτε το μακρύ πόδι του LED να είναι από την θετική πλευρά (5V), και το μικρό πόδι του LED να συνδεθεί με μια αντίσταση, να είναι δηλαδή στην ίδια κατακόρυφη πεντάδα τρυπών και να συνδεθούν τέλος με τη γείωση (GND).

Τα LED's ανάβουν μόνον όταν τα συνδέουμε με τη σωστή πολικότητα, και όχι ανάποδα! Την πολικότητα ενός LED την αναγνωρίζουμε από το μήκος των ακροδεκτών της. Ο μακρύτερος ακροδέκτης πρέπει να είναι σε θετική τάση ως προς τον κοντύτερο που πρέπει να συνδέεται με τη γείωση (GND).



Μεγάλο ποδαράκι στην τάση 5V ή σε μια ψηφιακή έξοδο
Μικρό ποδαράκι στη γείωση GND

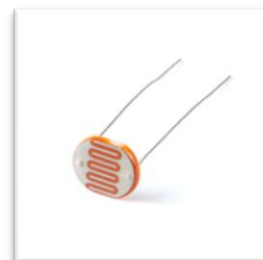
Οι αντιστάσεις

Οι αντιστάσεις χρησιμοποιούνται για να μειώσουν την ποσότητα ρεύματος που διαρρέει ένα υλικό, όπως για παράδειγμα ένα led, ώστε να μην καεί. Υπάρχουν πολλές αντιστάσεις στο εμπόριο, η μονάδα μέτρησης μιας αντίστασης είναι Ω hm.



Αισθητήρες φωτός (φωτοαντιστάσεις)

Η φωτοαντίσταση είναι μια μεταβλητή αντίσταση η τιμή της οποίας αλλάζει ανάλογα με το φως που πέφτει πάνω σε αυτή. Τη φωτοαντίσταση τη συναντάμε σε κυκλώματα τα οποία θέλουν με κάποιο τρόπο να αναγνωρίσουν αν το περιβάλλον είναι φωτεινό ή σκοτεινό. Για παράδειγμα λάμπες που ανάβουν αυτόματα σε χαμηλό φωτισμό. Συνδέεται με μια αναλογική είσοδο (A0-A5).



Αισθητήρες Κίνησης

Ο αισθητήρας αυτός χει την δυνατότητα να ανιχνεύσει την κίνηση μέσα σε έναν χώρο, σε εμβέλεια έξι μέτρων. Ο αισθητήρας διαθέτει δύο μεταβλητές αντιστάσεις (trimmer) όπου μπορείτε να ρυθμίσετε την ευαισθησία και τον χρόνο ενεργοποίησης του από την στιγμή που θα ανιχνεύσει την κίνηση.



Αισθητήρες απόστασης- υπερηχητικός αισθητήρας

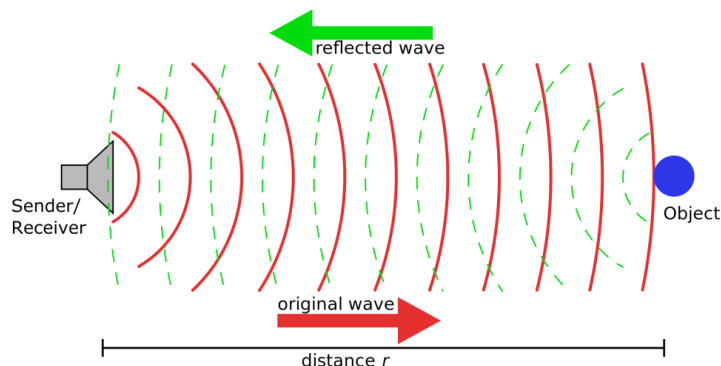
Ο υπερηχητικός αισθητήρας χρησιμοποιεί σόναρ για τον προσδιορισμό της απόστασης σε ένα αντικείμενο όπως οι νυχτερίδες ή τα δελφίνια. Προσφέρει εξαιρετική ανίχνευση εύρους με υψηλή ακρίβεια και σταθερές ενδείξεις. Μπορεί να ανιχνεύσει αποστάσεις από 2cm έως 400 cm.

Οι υπέρηχοι βρίσκονται πάνω απ' τις ακουστικές συχνότητες, έτσι δεν μπορεί να τους ακούσει το ανθρώπινο αυτί. Παρόλο πάντως που εμείς δεν τους ακούμε κάποια ζώα μπορούν και να τους ακούν αλλά και να τους χρησιμοποιούν. Χαρακτηριστικά παραδείγματα που μας το δείχνουν αυτό είναι η κίνηση των νυχτερίδων και η σφυρίχτρα που χρησιμοποιείται για τους σκύλους. Αισθητήρες υπερήχων συναντάμε σε πολλές



εφαρμογές στην ιατρική, στην πλοήγηση σκαφών/πλοίων ακόμα και στα αυτοκίνητα μας στα γνωστά park sensors.

Οι αισθητήρες υπερήχων λειτουργούν με την ίδια αρχή που λειτουργούν τα ραντάρ και τα σόναρ. Εκτιμούν την απόσταση ενός στόχου λαμβάνοντας υπόψη τους την αντανάκλαση ενός ραδιοκύματος ή ενός ηχητικού σήματος πάνω στο στόχο.



Δημιουργούν υψηλής συχνότητας κύματα και χρησιμοποιώντας το επιστρεφόμενο σήμα καθορίζουν την απόσταση ή ακόμα και την ταχύτητα του στόχου. Για να το επιτύχουν αυτό χρησιμοποιούν τον χρόνο που έκανε το σήμα για να καλύψει την απόσταση από τον αισθητήρα στο αντικείμενο και πίσω.

Ο αισθητήρας έχει τέσσερις (4) υποδοχές, η πρώτη είναι η Vcc που συνδέεται με το ρεύμα (5V), η τελευταία συνδέεται με τη γείωση (GND), η δεύτερη υποδοχή (trig) συνδέεται μια σειριακή θύρα και η τρίτη υποδοχή (Echo) με άλλη σειριακή θύρα.

Σερβοκινητήρας

Ο σερβοκινητήρας είναι ένας κινητήρας που μπορεί να περιστρέφει έναν άξονα από 0° μοίρες ως 180° μοίρες. Έχει 3 καλώδια, μαύρο ή καφέ, κόκκινο και πορτοκαλί. Το κόκκινο καλώδιο συνδέεται με την ακίδα 5V, το μαύρο/ καφέ με τη γείωση GND και το πορτοκαλί με μία ψηφιακή ακίδα ώστε να μπορούμε να τον ελέγξουμε.



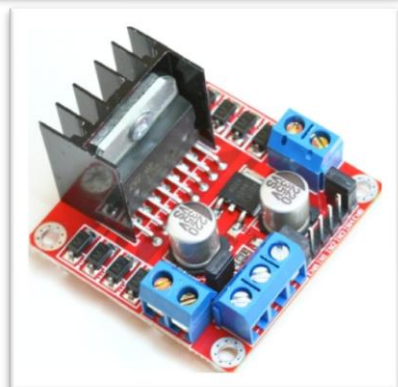
Buzzer

Είναι ένα μικρό μονοτονικό ηχείο. Τέτοια ηχεία συναντάμε σε υπολογιστές, ξυπνητήρια, σε συστήματα συναγερμού ακόμα και στα πληκτρολόγια των ATMs.



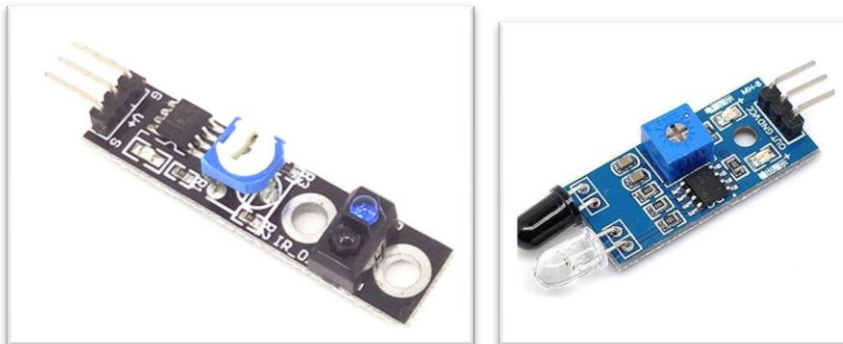
Γέφυρα- Bridge

Είναι το εξάρτημα που βοηθάει στην σύνδεση και τη λειτουργία κινητήρων. Θα το δούμε αναλυτικά αργότερα.



Αισθητήρες υπέρυθρων IR sensor

Οι αισθητήρες υπέρυθρων είναι ηλεκτρονικές συσκευές οι οποίες χρησιμοποιούνται για να ανιχνεύσουν τις αλλαγές που συμβαίνουν στο περιβάλλον τους. Η αλλαγή μπορεί να είναι χρώματος, θερμοκρασίας, υγρασίας, ήχου, θερμότητας κλπ. Αισθάνονται την αλλαγή και λειτουργούν αναλόγως.



Στους αισθητήρες υπέρυθρων υπάρχει ο πομπός και ανιχνευτής. Ο πομπός εκπέμπει τις υπέρυθρες ακτίνες και ο ανιχνευτής τις ανιχνεύει. Για παράδειγμα, ένας αισθητήρας φωτός, αν επιστρέψει μεγάλη ποσότητα φωτός τότε σημαίνει ότι υπάρχει κάτι κοντά ενώ όταν δεν επιστρέψει μεγάλη ποσότητα φωτός τότε δεν υπάρχει κάτι να ανιχνεύσει (κενό ή μαύρη γραμμή). Έτσι, μπορούμε να χρησιμοποιήσουμε τον αισθητήρα για να ανιχνεύει μια μαύρη γραμμή στο δάπεδο ώστε να μπορεί να κινείται πάνω της, να ελέγχει αν υπάρχει κενό, να ανιχνεύει εμπόδια κτλ.

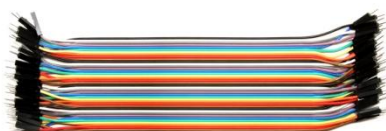
Καλώδια

Τα καλώδια τα χρησιμοποιούμε για να συνδέσουμε τα εξαρτήματά μας είτε με το breadboard είτε με το arduino. Υπάρχουν καλώδια που έχουν ακίδες από τη μια μεριά και τρύπες από την άλλη (Male-Female), άλλα που έχουν ακίδες και από τις δύο πλευρές (Male- Male) και άλλα που έχουν τρύπες και από τις δύο μεριές (Female- Female).

Τα καλώδια έχουν χρώματα. Μπορείτε να χρησιμοποιήσετε όποιο χρώμα θέλετε αλλά γενικά καλό είναι να φροντίζετε όπου υπάρχει γείωση να χρησιμοποιείτε μαύρο καλώδιο και όπου θέλετε να συνδέσετε τάση 5V τότε να χρησιμοποιείτε κόκκινο καλώδιο. Πολύ σημαντικό είναι όταν συνδέεται καλώδια αρσενικά με θηλυκά ώστε να έχετε μια προέκταση, να φροντίζετε να συνδέετε τα ίδια χρώματα.



M/F



M/M



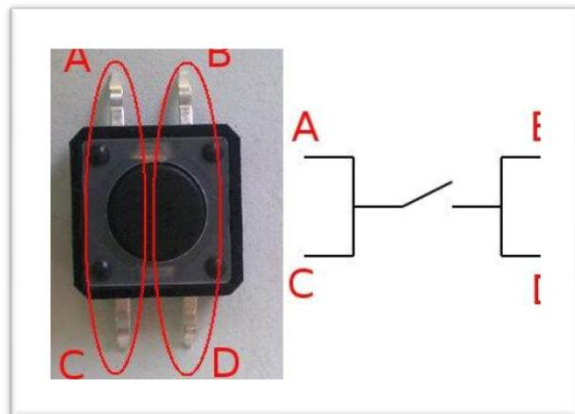
F/F

Διακόπτες

Διακόπτης on/off : Όταν είναι στο 1 τότε είναι κλειστό το κύκλωμα και όταν είναι στο 0 είναι ανοιχτό το κύκλωμα. Λειτουργούν όπως οι διακόπτες φωτός που έχουμε σπίτι μας.



Push button: Το κύκλωμα κλείνει όταν είναι πατημένος ο διακόπτης. Για να είναι πατημένος πρέπει να ασκηθεί συνεχώς πίεση σε αυτόν. Όπως όταν πατάμε έναν διακόπτη κουδουνιού μόνο τότε χτυπάει το κουδούνι.



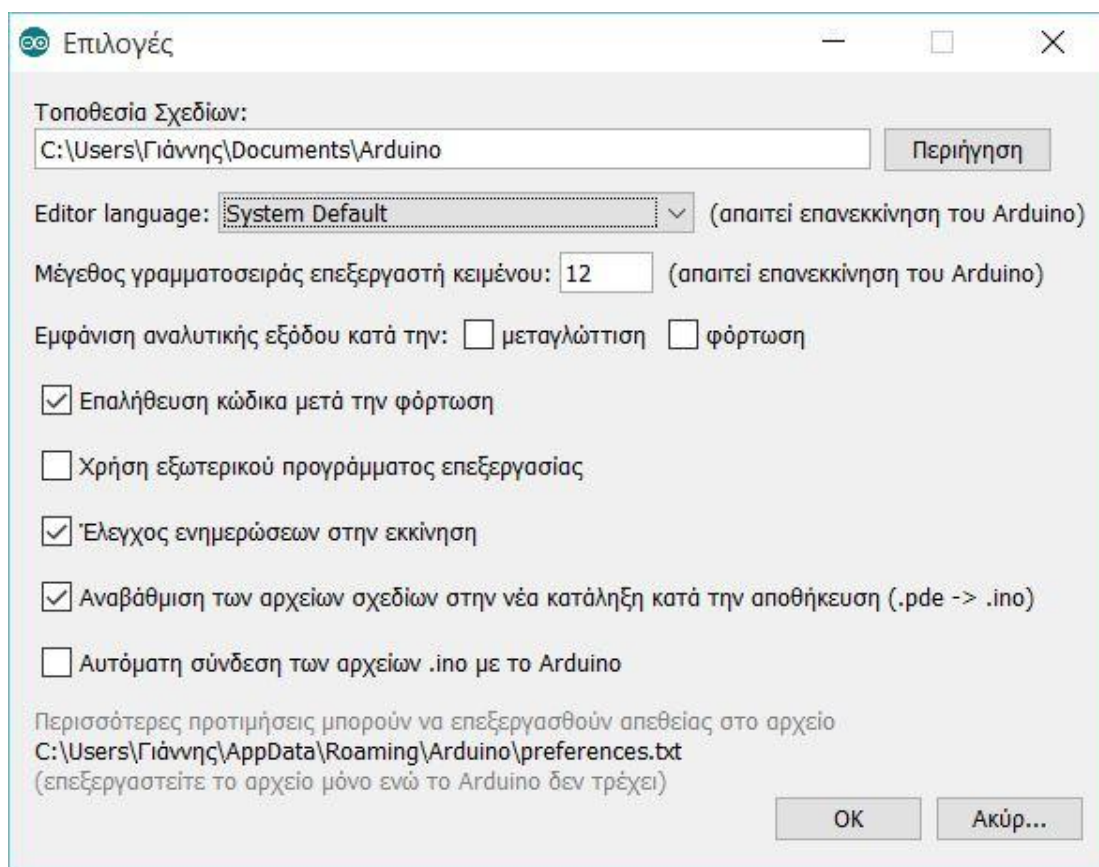
Τι είναι το Ardublock

Το ArduBlock είναι ένα περιβάλλον προγραμματισμού που έχει σχεδιαστεί για να κάνει τον προγραμματισμό του Arduino πολύ εύκολο χρησιμοποιώντας την τεχνική "σύρε και άσε" (drag-and-drop). Αντί να γράφουμε κώδικα και να ανησυχούμε με τη σύνταξη των εντολών στο περιβάλλον του Arduino IDE, το ArduBlock επιτρέπει τον οπτικό προγραμματισμό συνδυάζοντας έτοιμα μπλοκ κώδικα.

Εγκατάσταση του ArduBlock

(DOWNLOAD Ardublock)

1. Κατεβάστε το αρχείο: ardublock-all.jar ArduBlock από <https://sourceforge.net/projects/ardublock/>
2. Στο περιβάλλον του Arduino IDE, πηγαίνετε από το μενού "Αρχείο" -> "Επιλογές"
3. Βρείτε και σημειώστε κάπου τη θέση που βλέπετε στο "Τοποθεσία Σχεδίων:" ή επιλέξτε την και αντιγράψτε την.



Σε Mac, είναι εξ' ορισμού "Documents/Arduino" κάτω από το home directory του χρήστη.

Σε Linux, είναι εξ' ορισμού "sketchbook" κάτω από το home directory του χρήστη.

Σε Windows, είναι εξ' ορισμού "Documents\Arduino" κάτω από το home directory του χρήστη.

4. Αντιγράψτε το αρχείο που κατεβάσατε: ardublock-all.jar στο φάκελο tools/ArduBlockTool/tool κάτω από το φάκελο τοποθεσίας σχεδίων. Άν π.χ. ο χρήστης λέγεται "Γιάννης",

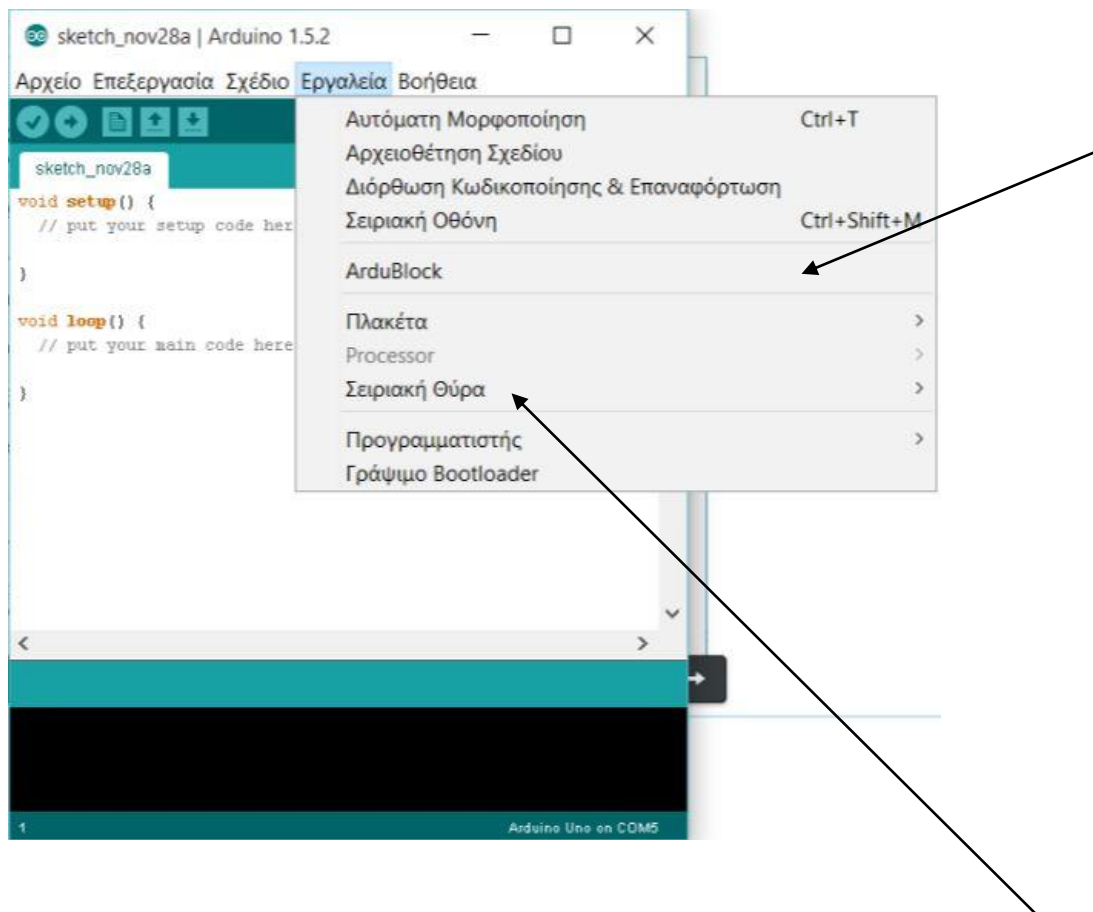
Σε Mac, /Users/Γιάννης/Documents/Arduino/tools/ArduBlockTool/tool/ardublock-all.jar

Σε Linux, /home/Γιάννης/sketchbook/tools/ArduBlockTool/tool/ardublock-all.jar

Σε Windows, C:\Users\Γιάννης\Documents\Arduino\tools\ArduBlockTool\tool\ardublock-all.jar

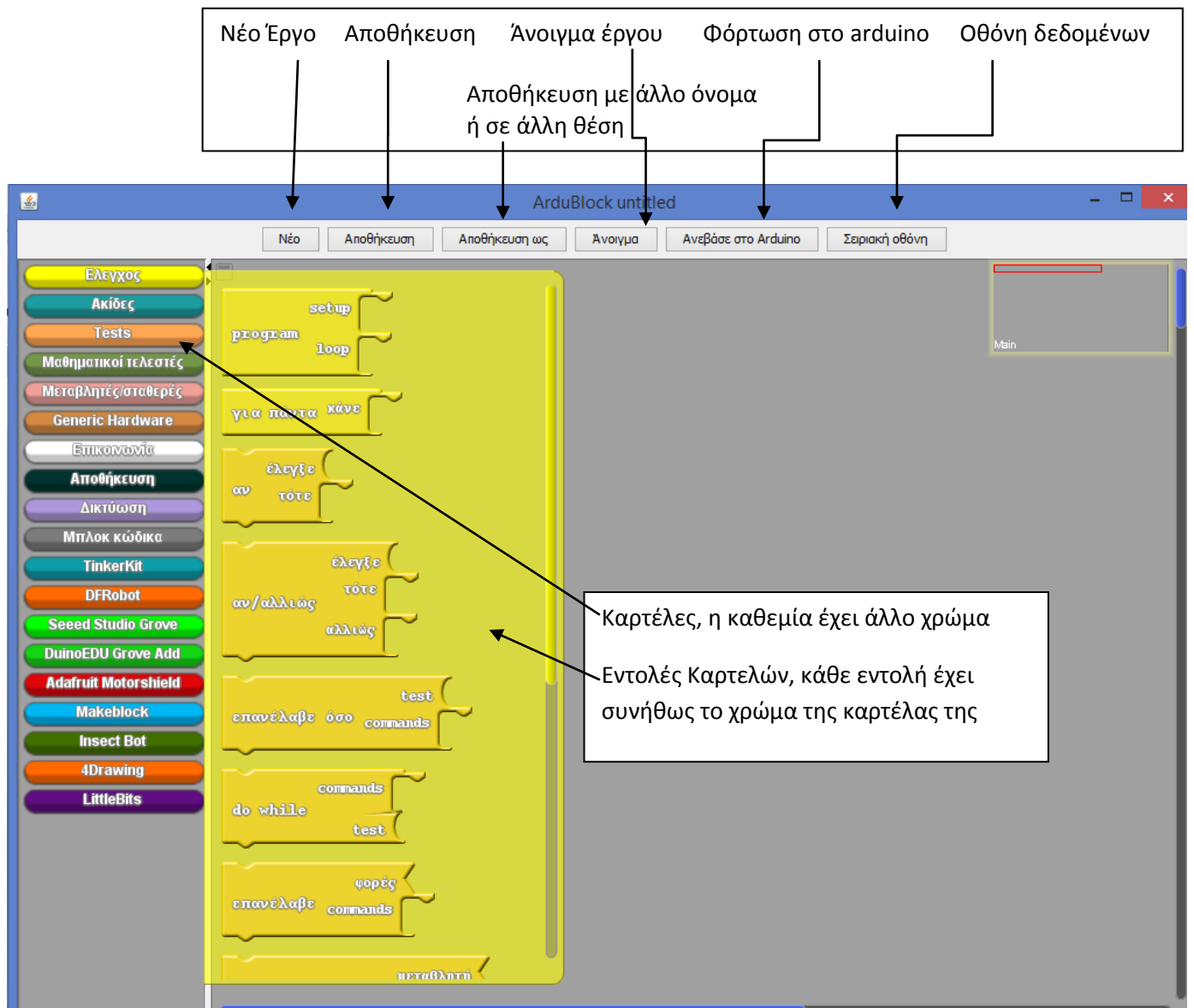
* Προσοχή υπάρχει διάκριση κεφαλαίων-μικρών για το όνομα του φακέλου "ArduBlockTool" κάτω από τον φάκελο tools.

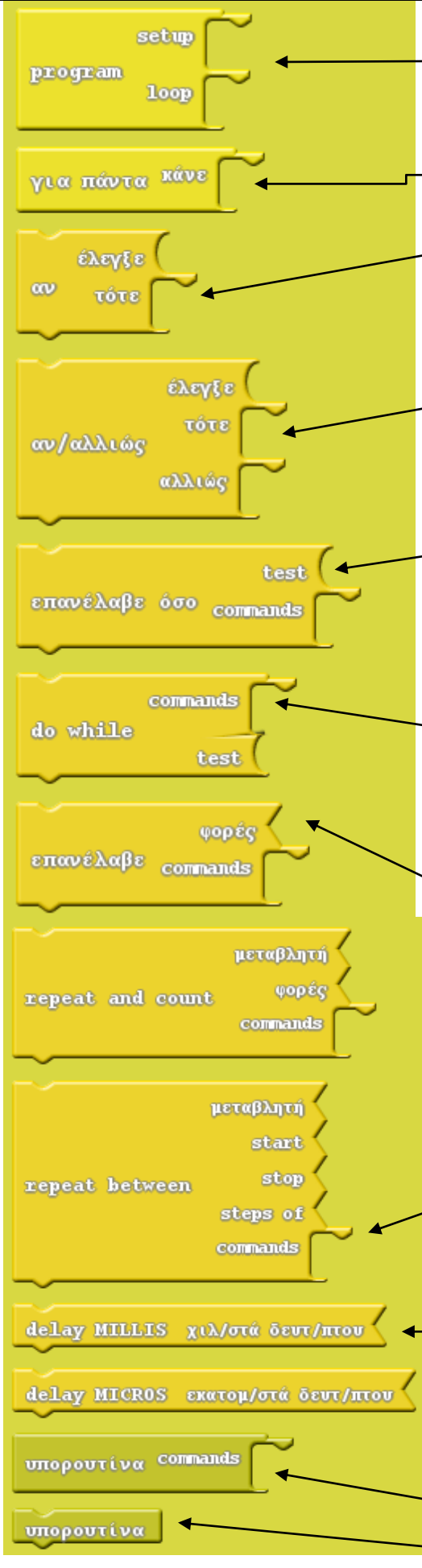
5. Ξεκινήστε το περιβάλλον του Arduino IDE και θα βρείτε την επιλογή ArduBlock στο μενού Εργαλεία.

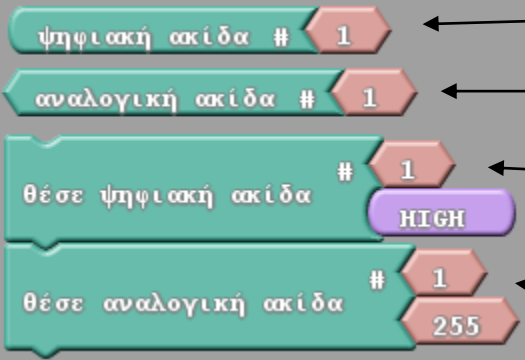
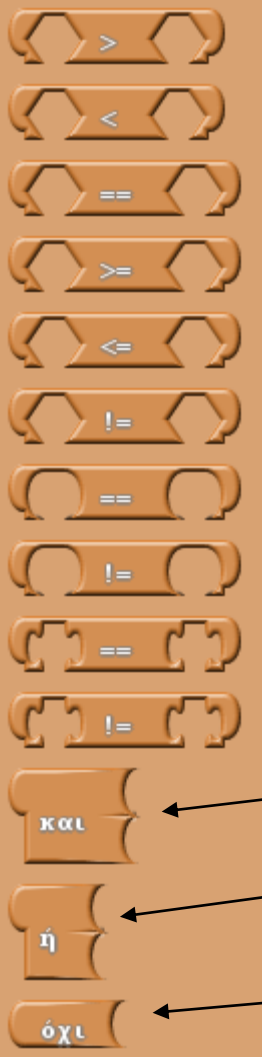


Θυμηθείτε ότι πρέπει να συνδέσετε την πλακέτα arduino σας μέσα από την επιλογή Εργαλεία-Σειριακή θύρα, επιλέγοντας τη θύρα στην οποία συνδέσατε την πλακέτα arduino.

Περιβάλλον εργασίας Ardublock- ελληνικά



Εντολές καρτέλας «Έλεγχος»	Λειτουργία
	Όταν έχω μια αρχική συνθήκη, πχ το λαμπάκι μου αρχικά να είναι σβηστό, αυτή η εντολή μπαίνει στο “setup” ενώ το κυρίως πρόγραμμα που εκτελείται συνέχεια μπαίνει στο “loop” και εκτελείται συνεχώς. Εντολές που θέλω να εκτελούνται συνέχεια. Δομή επιλογής, στο «έλεγξε» βάζω τη λογική συνθήκη που ελέγχει ενώ στο «τότε» τις εντολές που θα εκτελεστούν αν ισχύει η συνθήκη. Δομή πολλαπλής επιλογής, στο «έλεγξε» βάζω τη λογική συνθήκη που ελέγχει, στο «τότε» τις εντολές που θα εκτελεστούν αν ισχύει η συνθήκη και στο «αλλιώς» βάζω τις εντολές που θα εκτελεστούν αν δεν ισχύει η συνθήκη ελέγχου. Δομή επανάληψης με βάση κάποια συνθήκη, οι εντολές στο “commands” εκτελούνται όσο η συνθήκη που βάζω στο “test” ισχύει, όταν δεν ισχύει το πρόγραμμα συνεχίζει με τις εντολές μετά την επανάληψη. Δομή επανάληψης μέχρι κάποια συνθήκη, οι εντολές στο “commands” εκτελούνται όσο η συνθήκη που βάζω στο “test” δεν ισχύει, όταν ισχύει το πρόγραμμα συνεχίζει με τις εντολές μετά την επανάληψη. Εκτελείται τουλάχιστον μια φορά. Δομή επανάληψης για συγκεκριμένο αριθμό επαναλήψεων που ορίζω στο «φορές». Επανάληψη με έλεγχο μιας μεταβλητής η οποία αλλάζει με συγκεκριμένο αριθμό βημάτων, διαφορετικών του 1. Το “start” δίνουμε την αρχική τιμή της μεταβλητής, στο “stop” την τελική της τιμή και στο “steps of” το βήμα αλλαγής της μεταβλητής. Εντολές καθυστέρησης ώστε να κάνω μια εντολή να εκτελείται για συγκεκριμένο χρόνο, είναι σε χιλιοστά δευτερολέπτου, άρα αν θέλω 1 sec θα γράψω 1000 MILLIS. Ομοίως και με την καθυστέρηση σε MICROS. Υπορουτίνα με τις εντολές της. Εκτέλεση υπορουτίνας.

	Ορίζω μια ψηφιακή ακίδα- τη χρησιμοποιώ μέσα σε μπλοκ εντολών. Ορίζω μια αναλογική ακίδα τη χρησιμοποιώ μέσα σε μπλοκ εντολών. Θέτω σε μια ψηφιακή ακίδα μια συγκεκριμένη τιμή που μπορεί να είναι HIGH ή LOW. Θέτω σε μια αναλογική ακίδα μια οποιαδήποτε τιμή από -255 έως 255.
	Οι λογικές συνθήκες ελέγχου δίνουν ως αποτέλεσμα δύο τιμές, την τιμή ΑΛΗΘΗΣ-TRUE, ΙΣΧΥΕΙ και την τιμή ΨΕΥΔΗΣ, FALSE, ΔΕΝ ΙΣΧΥΕΙ. Χωρίζονται σε λογικές και αριθμητικές: >Μεγαλύτερο < μικρότερο >= μεγαλύτερο ή ίσο <= μικρότερο ή ίσο == ίσο != όχι ίσο, διάφορο του.. Μέσα κουμπώνουν τιμές ακίδων, μεταβλητές, αριθμοί κτλ. Και Ή Όχι, δίνει το αντίθετο αποτέλεσμα (αν η αρχική είναι αληθής τότε επιστρέφει ψευδής και το αντίστροφο).
	Μαθηματικοί τελεστές που εκτελούν μια πράξη μεταξύ αριθμών ή αριθμητικών μεταβλητών και επιστρέφουν μια αριθμητική τιμή. Το % επιστρέφει το ακέραιο υπόλοιπο μιας διαίρεσης ακέραιων αριθμών.

	Μεταβλητές και σταθερές Θέτω σε μια ακέραια μεταβλητή μια τιμή. Ονομάζω μια ακέραια μεταβλητή. Αριθμητική τιμή. Θέτω μια ψηφιακή μεταβλητή, παίρνει 2 τιμές. Ονομάζω μια ψηφιακή μεταβλητή. Έχει ρεύμα. Δεν έχει ρεύμα :για έλεγχο αν διαρρέεται από ρεύμα ή όχι ένα εξάρτημα. Αληθής ή ψευδής σε εντολές ελέγχου- Ισχύει ή όχι μια συνθήκη ελέγχου.
	Generic Hardware- ελέγχει κινητήρες και αισθητήρες Θέτει έναν σερβοκινητήρα που έχω βάλει σε μια ακίδα σε μια συγκεκριμένη γωνία. Έλεγχος του αισθητήρα υπερηχητικών, δίνω τις ακίδες που έχω βάλει το trigger και echo του αισθητήρα στο κύκλωμά μου.

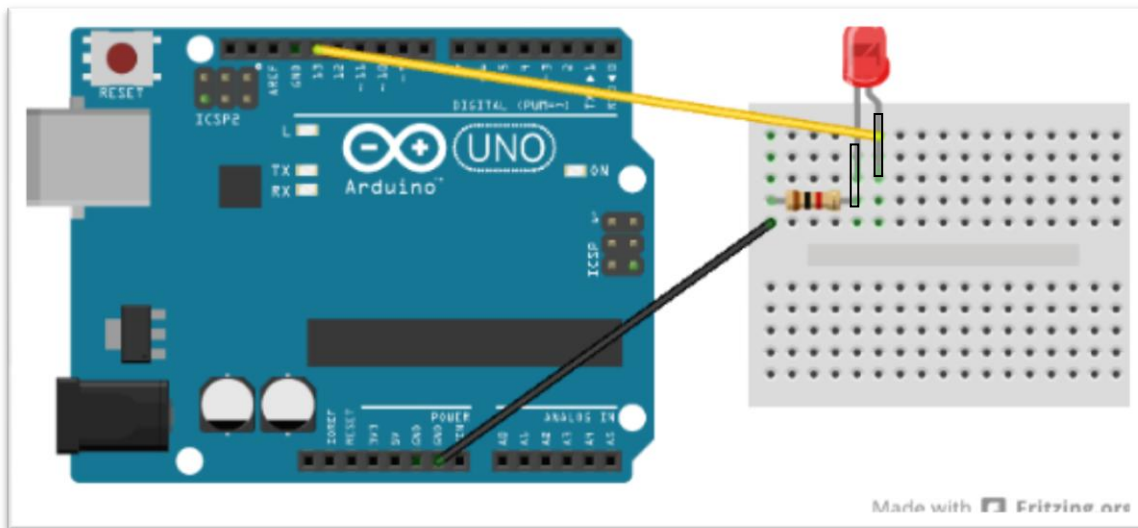
1^η Δραστηριότητα: Φωτάκια- Leds που αναβοσβήνουν

Στη πρώτη μας δραστηριότητα θα συνδέσουμε ένα led στο arduino χρησιμοποιώντας το breadboard και αντιστάσεις και στη συνέχεια θα το προγραμματίσουμε ώστε να αναβοσβήνει συνέχεια ανά 1 δευτερόλεπτο, δηλαδή για 1 sec να είναι αναμμένο και για 1 sec να είναι σβηστό κ.ο.κ.

Για τη συνδεσμολογία μας θα χρειαστούμε:

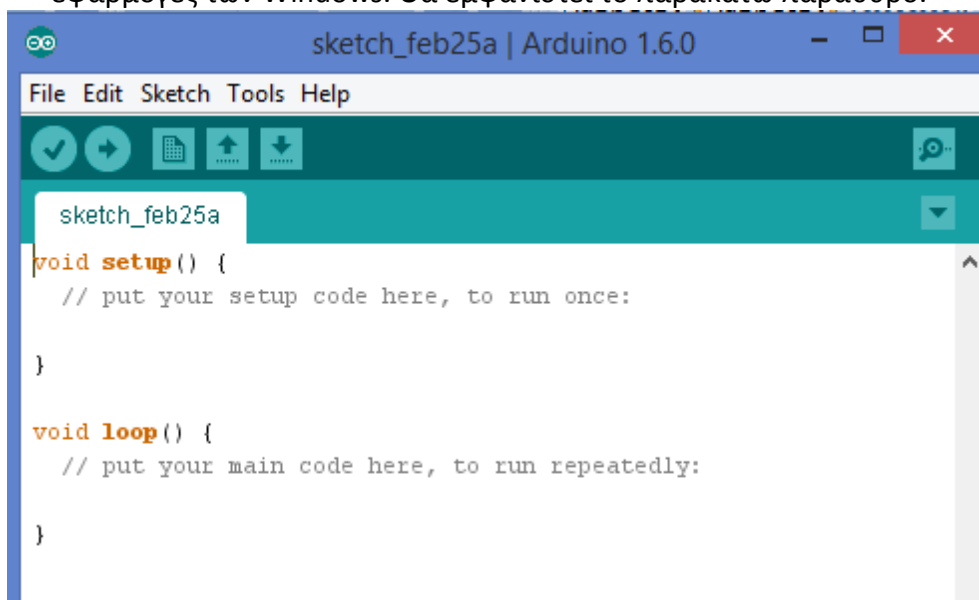
- Μικροελεγκτή Arduino
- 1 breadboard
- 1 αντίσταση
- 1 led κόκκινου χρώματος
- 1 καλώδιο κίτρινο και 1 μαύρο

Συνδέστε τα εξαρτήματα σύμφωνα με την παρακάτω συνδεσμολογία. Το κοντό ποδαράκι του led συνδέεται με μια αντίσταση και στη συνέχεια με τη γείωση ενώ το μεγάλο ποδαράκι με την ακίδα 13 από την οποία θα παίρνει ρεύμα και θα ελέγχεται. Προσέξτε το ποδαράκι της αντίστασης που είναι στην ίδια κάθετη σειρά με το μικρό ποδαράκι της αντίστασης (κυκλωμένο) καθώς και το μεγάλο ποδαράκι της αντίστασης με το κίτρινο καλώδιο που πηγαίνει στην ακίδα 13.

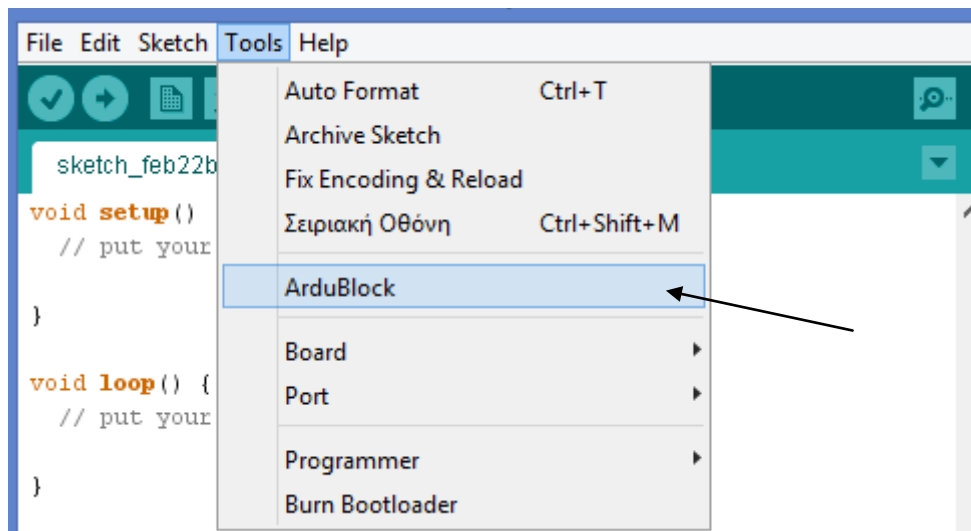


Οδηγίες προγραμματισμού

1. Ανοίξτε το προγραμματιστικό περιβάλλον του Arduino από την επιφάνεια εργασίας ή από τις εφαρμογές των Windows. Θα εμφανιστεί το παρακάτω παράθυρο.



2. Από το μενού Tools αρχικά βεβαιωθείτε ότι είναι συνδεδεμένο το arduino με τον υπολογιστή ελέγχοντας τη θύρα «Port»
3. Από το μενού Tools επιλέξτε «Ardublock» ώστε να ανοίξει το προγραμματιστικό περιβάλλον του Ardublock

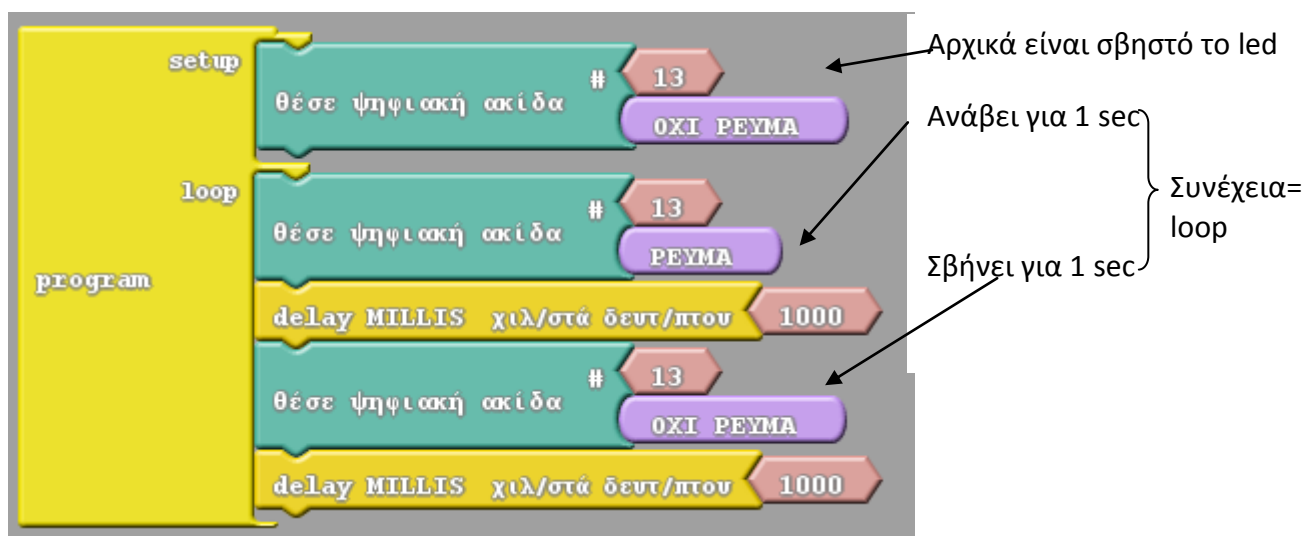


Για να ανάψει το led μας χρειάζεται ρεύμα, δηλαδή, είτε να συνδεθεί με την τροφοδοσία 5V είτε να πάρει ρεύμα μέσα από μία ψηφιακή ή αναλογική έξοδο.

Θα προτιμήσουμε το δεύτερο όπως είπαμε παραπάνω, καθώς θέλουμε να ελέγχουμε παράλληλα το led, να του δίνουμε ρεύμα για συγκεκριμένο χρόνο ή να το σβήνουμε και όχι να το έχουμε αναμμένο συνεχώς.

Οι εντολές που θα εισάγουμε βρίσκονται στο προγραμματιστικό περιβάλλον ardublock, στην κίτρινη καρτέλα «Έλεγχος» και στην μπλε καρτέλα «Ακίδες». Χρειαζόμαστε και την εντολή «για πάντα» γιατί θέλουμε κάτι να γίνεται συνέχεια και όχι για μια μόνο φορά (να αναβοσβήνει συνεχώς).

Χρησιμοποιείτε τις παρακάτω εντολές και προσπαθήστε να τις βάλετε σε σειρά ώστε να ανάβει το λαμπάκι για 1 sec και στη συνέχεια να σβήνει για 1 sec. Να θυμίσουμε ότι θα βάλετε το led στη θύρα 13 αλλά μπορείτε και σε οποιαδήποτε άλλη θύρα. Μη ξεχάσετε όλες τις εντολές σας να τις βάλετε μέσα στο κίτρινο πλακίδιο «για πάντα κάνε» ώστε να εκτελούνται συνεχώς.

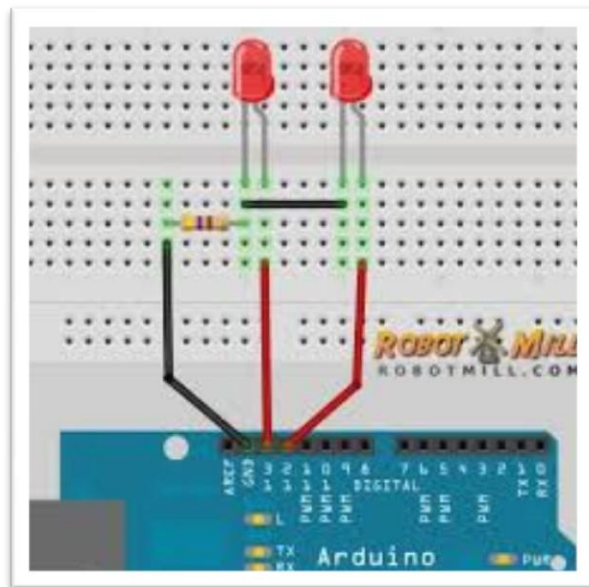


4. Πατήστε το κουμπί «Ανέβασε στο Arduino» ώστε να φορτωθεί το πρόγραμμα στον μικροελεγκτή και να εκτελεστεί. Παρατηρείστε τι γίνεται στην κατασκευή σας.

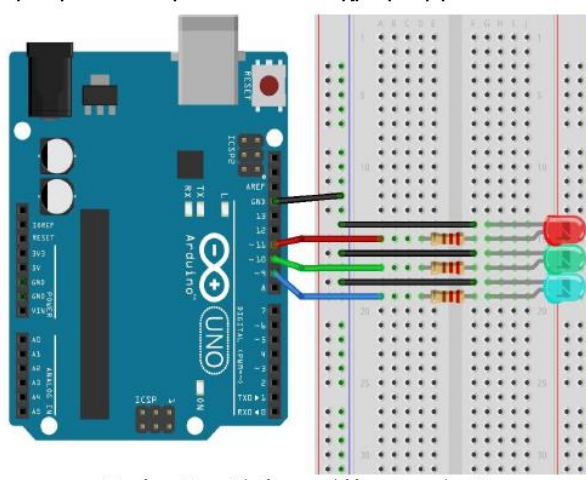
Δουλεύουν όλα σωστά;.....

Δοκιμάστε να αλλάξετε τους χρόνους ώστε να αναβοσβήνει πιο γρήγορα ή πιο αργά.

5. Δοκιμάστε να συνδέσετε και δεύτερο λαμπάκι, διαφορετικού ή ίδιου χρώματος, σε άλλη ψηφιακή έξοδο και δοκιμάστε όταν ανάβει το ένα να σβήνει το άλλο κοκ.



6. Εισάγετε 3 λαμπάκια χρώματος πράσινο, κίτρινο και κόκκινο. Προσομοιώστε το κύκλωμα να λειτουργεί ως το φανάρι των αυτοκινήτων. Σκεφτείτε τους χρόνους που θα είναι ανοιχτά ή σβηστά τα λαμπάκια



7. Τι άλλο θα χρειαζόταν να έχει η κατασκευή μας ώστε να μπορούμε να προσομοιάζουμε ένα φανάρι αυτοκινήτων και ένα φανάρι πεζών ταυτόχρονα;

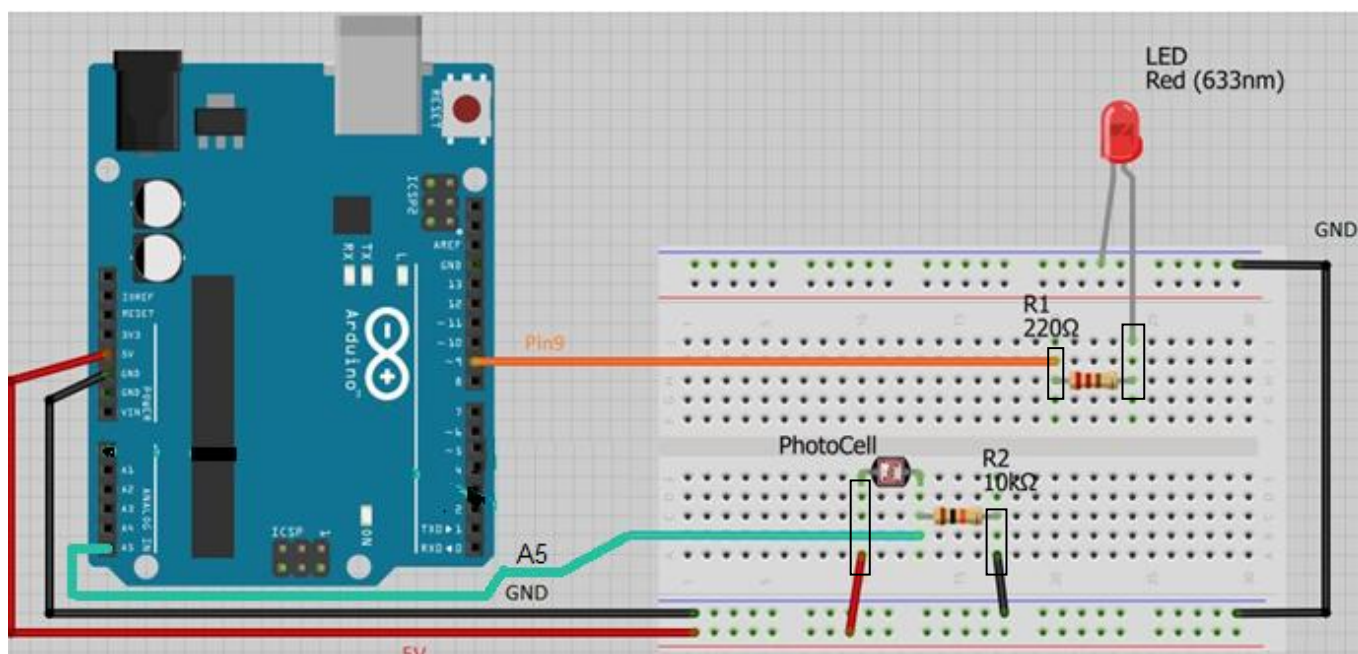
2^η Δραστηριότητα: Αυτόματο φωτιστικό

Σε αυτή τη δραστηριότητα θα δημιουργήσουμε ένα αυτόματο σύστημα φωτός στο οποίο θα ανάβει ένα ή περισσότερα φώτα (leds) αν η ποσότητα του φωτός είναι κάτω από μια τιμή. Κάτι τέτοιο γίνεται στα αυτόματα φώτα που ανάβουν όταν νυχτώνει και ελαττωθεί η ποσότητα του φωτός.

Για τη δραστηριότητα αυτή θα χρειαστούμε:

- Arduino uno
- Έναν αισθητήρα φωτός- φωτοαντίσταση
- Ένα led
- Δύο αντιστάσεις- 220Ω (για το led) και 10KΩ (για τη φωτοαντίσταση)
- Καλώδια
- Breadboard

Δημιουργήστε το παρακάτω κύκλωμα



Την φωτοαντίσταση τη συνδέουμε με την αναλογική είσοδο A5 γιατί, οι τιμές που δίνει δεν είναι διακριτές (0/1, ΡΕΥΜΑ/ΟΧΙ ΡΕΥΜΑ, ΑΛΗΘΕΣ/ΨΕΥΔΕΣ) αλλά παίρνει πολλές τιμές, από 0 έως 255. Άρα, αν η τιμή της φωτοαντίστασης είναι πάνω από μια τιμή, πχ 100 τότε να ανάβει το led, αλλιώς να είναι σβηστό.

Θα χρησιμοποιήσουμε την εντολή **Αν (συνθήκη)...τότε...αλλιώς...**

Η συνθήκη είναι κάποιες παραστάσεις, ή λογικές τιμές που μπορούν να έχουν δύο πιθανές απαντήσεις, σωστό ή λάθος, ναι ή όχι, αληθές ή ψευδές.

Για παράδειγμα η παράσταση $\alpha > \beta$ μπορεί να είναι αλήθεια ή ψέματα, σωστό ή λάθος ανάλογα τις τιμές που θα δώσουμε.

Έτσι και εδώ, η συνθήκη μας θα είναι:

Αν η τιμή της αναλογικής ακίδας 5 είναι μικρότερη ή ίση από μια τιμή
τότε θα ανάβει το led
αλλιώς θα είναι σβηστό.

Θα χρησιμοποιήσετε τις εντολές της προηγούμενης δραστηριότητας αλλά στην τιμή ελέγχου θα χρησιμοποιήσετε την εντολή για την αναλογική ακίδα  και στην συνθήκη ελέγχου το καφέ μπλοκ  ώστε να ελέγχουμε αν η τιμή της αναλογικής ακίδας 0 είναι μικρότερη ή ίση από 100.



Το πρόγραμμα σε κώδικα arduino

Με την ίδια λογική μπορείτε να υλοποιήσετε όλα τα συστήματα ελέγχου όπως για παράδειγμα, αν σε ένα θερμοκήπιο είναι μικρή η τιμή υγρασίας του εδάφους τότε να ανοίγει η παροχή του νερού, αν ένας αισθητήρας δόνησης ανιχνεύσει κάποια δόνηση (πχ από σεισμό) να ενεργοποιεί ένα ηχείο και να ανάβουν led κτλ.

Σκεφτείτε κάποιο άλλο παράδειγμα από το σπίτι σας, το περιβάλλον σας και καταγράψτε το:

.....

.....

.....

.....

.....

.....

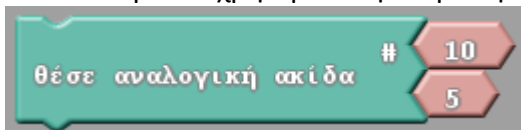
.....

.....

3^η Δραστηριότητα: Αλλαγή φωτεινότητας Led

Σε αυτή τη δραστηριότητα θα βάλουμε ένα ή περισσότερα leds σε ένα breadboard και θα αυξάνουμε και θα μειώνουμε σταδιακά την ένταση του φωτός τους. Θα δημιουργήσουμε ένα κύκλωμα με led όπως στην 1^η δραστηριότητα και θα συνδέσουμε το led στην ψηφιακή ακίδα 10. Οι ψηφιακές τιμές είπαμε ότι είναι δυαδικές δηλαδή παίρνουν μια από δύο διαθέσιμες τιμές, δηλαδή 1 ή 0, ΝΑΙ ή ΟΧΙ, ΣΩΣΤΟ ή ΛΑΘΟΣ, ΡΕΥΜΑ ή ΟΧΙ ΡΕΥΜΑ κτλ. Εμείς όμως θέλουμε να διαρρέει ρεύμα το led αλλά σε διάφορες τιμές ώστε να αλλάζει η φωτεινότητά του, να φέγγει λίγο ή περισσότερο. Άρα θα πρέπει να λειτουργήσουμε με αναλογική έξοδο (αυτή που έχει το ~ πάνω της), πχ την ακίδα 10.

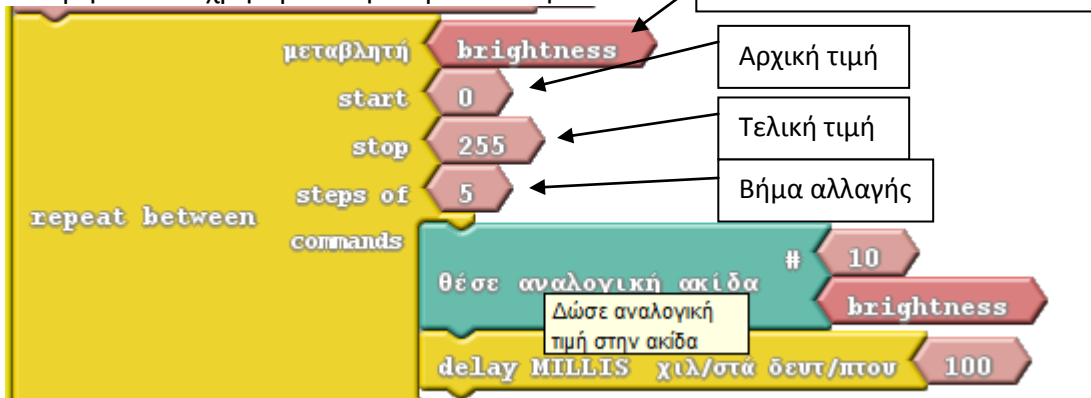
Μια μεταβλητή λέγεται αναλογική όταν μπορεί να πάρει πολλές διαφορετικές τιμές μέσα από ένα εύρος τιμών. Για παράδειγμα, εδώ τα led μπορούν να έχουν φωτεινότητα από 0 έως 255 ενώ στην ψηφιακή τιμή μπορούν να έχουν 0 ή 255. Έτσι αν θέλω να βγαίνει λίγο φως μπορώ να έχω μια τιμή πχ 100 κτλ. Άρα θα χρησιμοποιήσουμε την εντολή



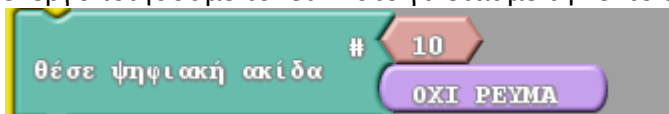
για κάποιο χρόνο (εντολή delay)

Στη συνέχεια αν θέλουμε να αυξήσουμε τη φωτεινότητα, θα αλλάξουμε την τιμή 5 σε 10, 15, 20,255. Για να το πετύχουμε αυτή θέλουμε 255/5 φορές, κάτι που είναι πολύ χρονοβόρο. Έτσι, χρειαζόμαστε μια δομή επανάληψης για να το πετύχουμε αυτό.

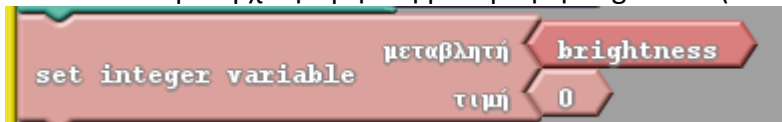
Η δομή που θα χρησιμοποιήσουμε είναι η



Αρχικά όμως θα πρέπει να δηλώσουμε τι τιμή θα έχει η ένταση φωτός του led μας και να ενεργοποιήσουμε το led. Αυτό γίνεται με την εντολή που μάθαμε στην 1^η δραστηριότητα



Και να θέσουμε αρχική τιμή στη μεταβλητή brightness (ένταση φωτός)



που βρίσκεται στην καρτέλα «μεταβλητές».

Δοκιμάστε το πρόγραμμά σας για ένα led. Στη συνέχεια τροποποιήστε κατάλληλα το πρόγραμμα ώστε να είναι αρχικά αναμμένο (high) και να έχει τη μέγιστη τιμή (255) και να μειώνεται αυτή μέχρι το 0. Σκεφτείτε και συμπληρώστε τα παρακάτω. Μη ξεχάσετε να αποθηκεύσετε το πρόγραμμά σας!

Αρχική τιμή:

Τελική τιμή:

Βήμα:

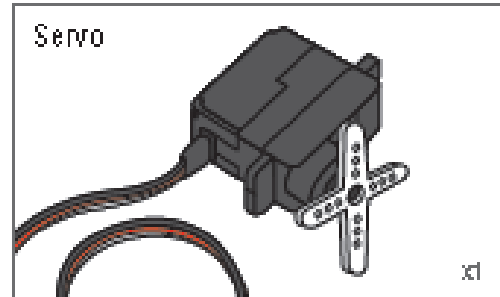
4^η Δραστηριότητα: Χρησιμοποιώντας σερβοκινητήρες- Χαιρετώντας τον κόσμο

Σε αυτή τη δραστηριότητα θα σχεδιάσουμε και θα κατασκευάσουμε μια παλάμη από χαρτόνι και καλαμάκι. Στη συνέχεια, με τη βοήθεια του σερβοκινητήρα θα περιστρέφουμε την παλάμη με τέτοιο τρόπο ώστε να φαίνεται ότι χαιρετάει τον κόσμο!

Ο σερβοκινητήρας μπορεί να περιστρέψει έναν άξονα από 0 έως 180 μοίρες (όχι ολόκληρο κύκλο)

Για το σκοπό αυτό θα χρειαστούμε:

- Χαρτόνι
- Ψαλίδι
- Καλαμάκι
- Σελοτέιπ- κόλλα
- Καλώδια
- Arduino Uno



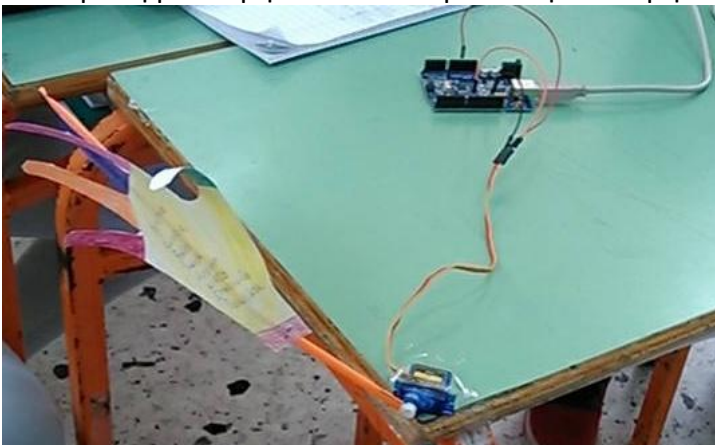
Βήμα1- Κατασκευή παλάμης

- Σχεδιάστε μια παλάμη πάνω στο χαρτόνι που έχετε και χρωματίστε την κατάλληλα. Μπορείτε να σχεδιάσετε νύχια, κοσμήματα και ό,τι άλλο θέλετε. Γράψτε στο πίσω μέρος τα ονόματά σας
- Συρράψτε ή κολλήστε στο καλαμάκι σας στο κάτω μέρος της παλάμης.
- Την άλλη άκρη (άξονα) από το καλαμάκι περάστε την στον σερβοκινητήρα, στο άσπρο κινητό μέρος και αν χρειαστεί στερεώστε την με σελοτέιπ
- Κολλήστε τον σερβοκινητήρα με σελοτέιπ στην άκρη στο θρανίο σας ώστε το μέρος που περιστρέφεται να βρίσκεται στο κενό για να μπορεί να περιστραφεί

Σχεδιάζουμε σε χαρτόνι μια παλάμη και τη χρωματίζουμε όπως επιθυμούμε.

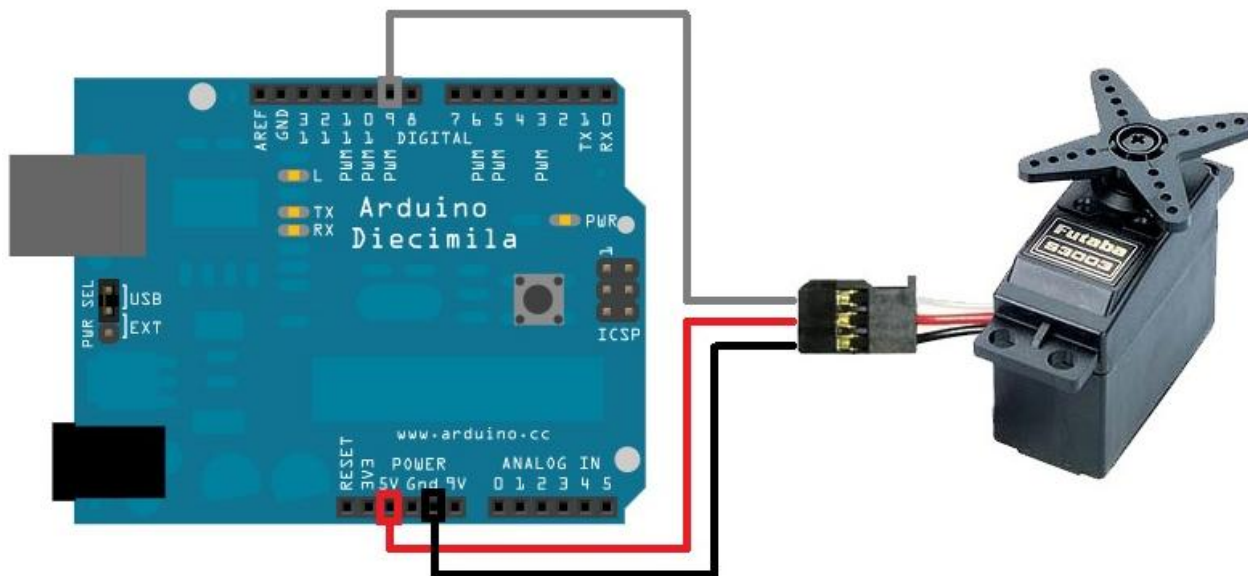


Κολλάμε τη μια άκρη από το καλαμάκι στην παλάμη και την άλλη στον άξονα του σερβοκινητήρα.



Βήμα2- Προγραμματισμός- Σύνδεση του σερβοκινητήρα με το arduino

Ο σερβοκινητήρας είναι ένας κινητήρας που μπορεί να περιστρέφει έναν άξονα από 0° μοίρες ως 180° μοίρες. Στην άκρη του έχει 3 καλώδια, μαύρο, κόκκινο και πορτοκαλί. Το κόκκινο καλώδιο συνδέεται με την ακίδα 5V, το μαύρο με τη γείωση GND και το πορτοκαλί με μια από τις ψηφιακές ακίδες.



Συνδέστε τον σερβοκινητήρα όπως φαίνεται στο παραπάνω σχεδιάγραμμα.

Το κόκκινο καλώδιο στα 5V, το μαύρο στη γείωση GND και το πορτοκαλί με την ψηφιακή ακίδα 9. Εδώ δεν χρειαζόμαστε Breadboard, θα το χρειαστούμε όταν θα βάλουμε και άλλους σερβοκινητήρες. Ανοίξτε το arduino και στη συνέχεια το ardublock σύμφωνα με τις οδηγίες που είδαμε στην 1^η δραστηριότητα.

Θα χρησιμοποιήσετε την εντολή που βρίσκεται στην καφέ καρτέλα «Generic Hardware» για να ελέγξετε τον σερβοκινητήρα. Δηλώνετε την ακίδα που έχετε συνδέσει τον σερβοκινητήρα πατώντας την μπλε καρτέλα «ακίδες» και τη γωνία που θέλετε να έχει ο σερβοκινητήρας σας πατώντας τη ροζ καρτέλα «μεταβλητές/σταθερές».



Μπορείτε να το κάνετε αυτό για κάποιο χρόνο χρησιμοποιώντας την εντολή delay που έχουμε δει και στη συνέχεια να αλλάξετε τη γωνία.

Αρχικά θέλω να βρίσκεται στη γωνία 0 μοίρες. Αυτό θα πρέπει να το ελέγξετε τοποθετώντας κατάλληλα τον άξονα στον σερβοκινητήρα και γυρνώντας το με το χέρι σας να καταλάβετε την έναρξη και το τέλος της περιστροφής του.

Ξεκινήστε με το παρακάτω απλό παράδειγμα:

- Να ξεκινά στις 90 μοίρες για 1 δευτερόλεπτο
- Να στρίβει στις 180 μοίρες για 1 δευτερόλεπτο
- Να επιστρέφει στις 90 μοίρες για 1 δευτερόλεπτο
- Να στρίβει στις 0 μοίρες για 1 δευτερόλεπτο

Γράψτε εδώ το πρόγραμμά σας και στη συνέχεια εισάγετε τον κώδικα στο ardublock

Δοκιμάστε να αλλάζει η γωνία ανά 2 δευτερόλεπτα.

Συνεχίστε ως εξής:

Σκεφτείτε η γωνία να ξεκινάει από 0 μοίρες και ανά 0,2 δευτερόλεπτα να αλλάζει 10 μοίρες, μέχρι να φτάσει στις 180 μοίρες.

Για να πάει από 0 μοίρες σε 180 μοίρες ανά δέκα μοίρες, πόσες φορές θα αλλάξει γωνία;.....

Για να πάει από 0 μοίρες σε 180 μοίρες ανά μία μοίρα, πόσες φορές θα αλλάξει γωνία;.....

Είναι εύκολο να γράψουμε αυτόν τον κώδικα στο ardublock; Δικαιολογήστε την άποψή σας.

Υπάρχει κάποια εντολή επανάληψης που μπορούμε να χρησιμοποιήσουμε δεδομένου ότι γνωρίζουμε πόσες φορές θα αλλάξει γωνία ο σερβοκινητήρας μας;.....

Αν ναι συμπληρώστε την εδώ με τις κατάλληλες εντολές:

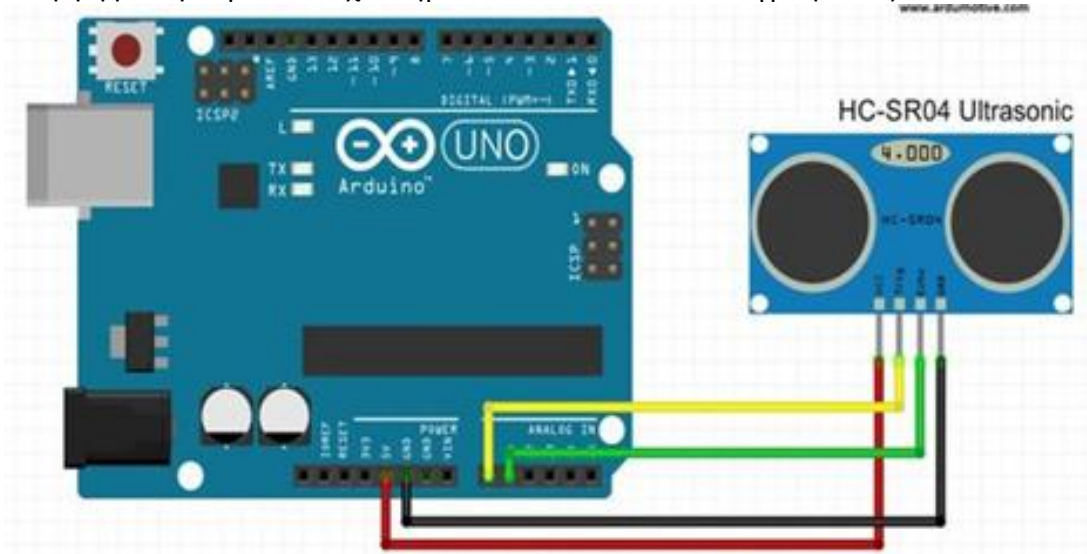
5^η Δραστηριότητα: Σύστημα συναγερμού ελέγχου απόστασης

Ο υπερηχητικός αισθητήρας χρησιμοποιεί σόναρ για τον προσδιορισμό της απόστασης σε ένα αντικείμενο όπως οι νυχτερίδες ή τα δελφίνια. Προσφέρει εξαιρετική ανίχνευση εύρους χωρίς επαφή με υψηλή ακρίβεια και σταθερές ενδείξεις σε μια εύχρηστη συσκευασία. Μπορεί να ανιχνεύσει αποστάσεις από 2cm έως 400cm.



Σε αυτή τη δραστηριότητα θα δούμε πώς να χρησιμοποιούμε αυτόν τον αισθητήρα με το Arduino uno και να ανάβουμε και σβήνουμε ένα led ανάλογα με την απόσταση. Παρακάτω φαίνεται η σύνδεση του αισθητήρα υπερήχων στο arduino.

Ο αισθητήρας έχει 4 υποδοχές, η πρώτη είναι η Vcc που συνδέεται με το ρεύμα (5V), η τελευταία (GND) συνδέεται με τη γείωση (GND), η 2^η υποδοχή (trig) συνδέεται σε μια ψηφιακή θύρα και η 3^η υποδοχή (Echo) σε άλλη ψηφιακή θύρα. Δεν έχει σημασία σε ποια αλλά καταγράψτε τις.

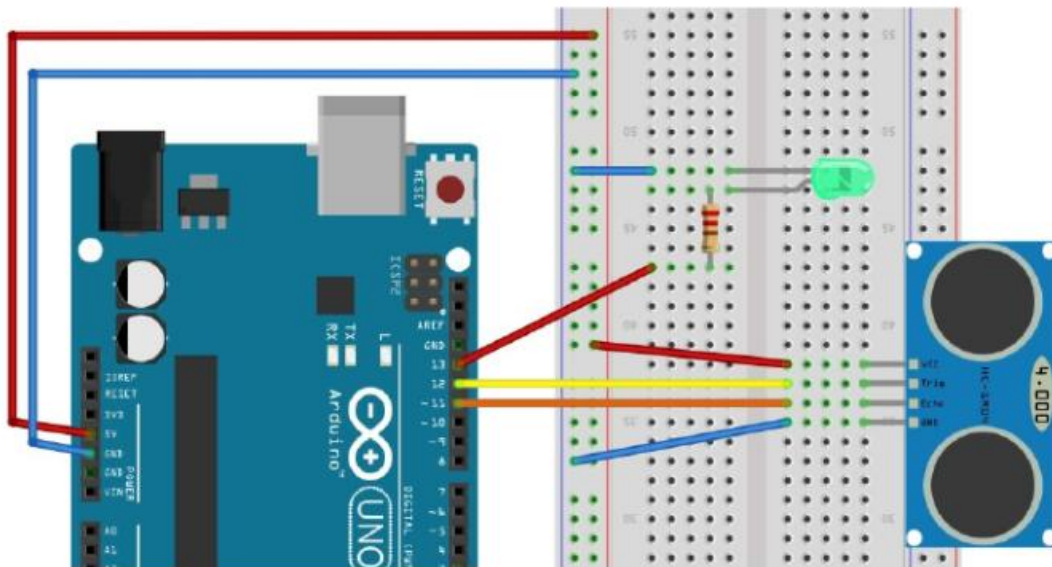


Στη δραστηριότητα αυτή θα ελέγχουμε την απόσταση από ένα αντικείμενο. Αν έρθει κοντά ένα αντικείμενο, κάτω από μια συγκεκριμένη απόσταση, τότε θα ανάβει ένα λαμπάκι. Σκεφτείτε τα καινούρια αυτοκίνητα που έχουν αισθητήρες παρκαρίσματος και όταν είναι κοντά σε εμπόδιο χτυπάει ένα buzzer.

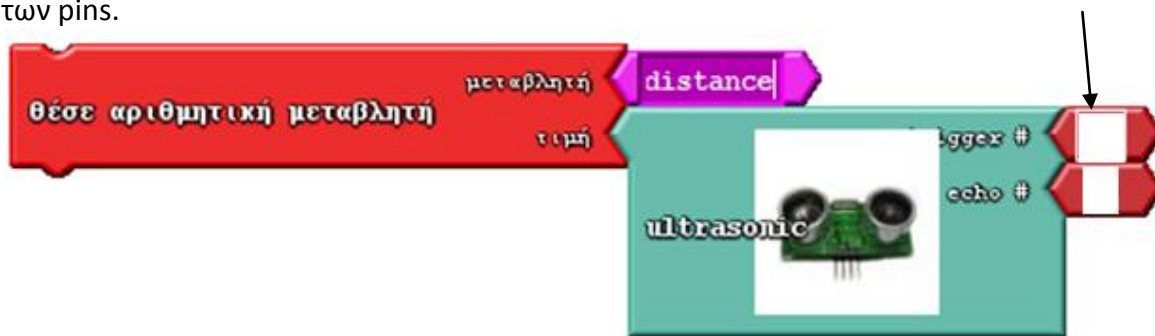
Για τη δραστηριότητά μας θα χρειαστούμε τα παρακάτω:

- 1 Arduino Uno
- 1 Breadboard
- 1 Αισθητήρα απόστασης Ultrasonic Sensor
- 1 αντίσταση 1 Ωhm
- 1 led

Δημιουργούμε το κύκλωμα συνδέοντας το trigger στο 12 και το echo στο 11.



Συνδέουμε τον αισθητήρα σύμφωνα με την παραπάνω συνδεσμολογία. Μέσα από το Ardublock ας προσπαθήσουμε να χρησιμοποιήσουμε την απόσταση αυτή θέτοντας αρχικά την τιμή που επιστρέφει ο αισθητήρας σε μια μεταβλητή πχ distance. Σημειώστε δίπλα από το trigger και echo τις σωστές τιμές των pins.



Στη συνέχεια θα ελέγχουμε την τιμή αυτή και αν είναι κάτω από μια τιμή πχ κάτω από 20 εκ τότε να ανάβει το led μας, διαφορετικά (αλλιώς) να είναι σβηστό το led.

Θα χρησιμοποιήσουμε την εντολή Αν (συνθήκη)...τότε...αλλιώς...

Η συνθήκη όπως είδαμε, είναι κάποιες παραστάσεις, ή λογικές τιμές που μπορούν να έχουν δύο πιθανές απαντήσεις, σωστό ή λάθος, ναι ή όχι, αληθές ή ψευδές. Για παράδειγμα η παράσταση $a > b$ μπορεί να είναι αλήθεια ή ψέματα, σωστό ή λάθος ανάλογα τις τιμές που θα δώσουμε. Έτσι και εδώ, η συνθήκη μας θα είναι **distance < 20** και αν αυτή είναι αληθής (αν ισχύει δηλαδή) θα ανάβει το led αλλιώς θα είναι σβηστό.

Συνδέουμε ένα led όπως δείξαμε στην 1^η δραστηριότητα και συνδέουμε και τον αισθητήρα απόστασης όπως παραπάνω. Η εντολή ελέγχου βρίσκεται στην κίτρινη καρτέλα και οι εντολές για τον σχηματισμό της συνθήκης στην καφέ (tests)



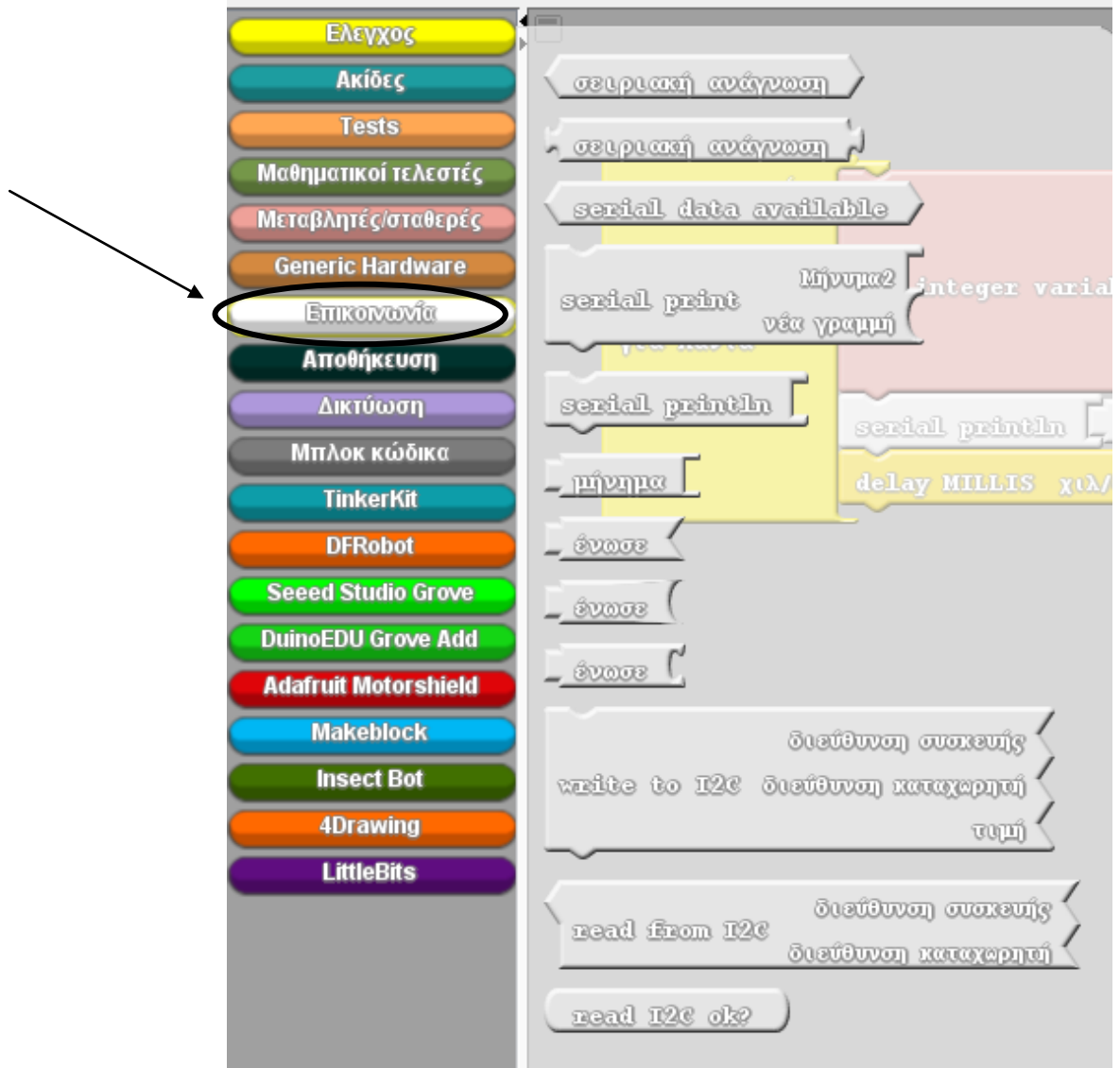
Το πρόγραμμα στο ardublock

Αποθηκεύστε το πρόγραμμά σας και φορτώστε το στο arduino σας

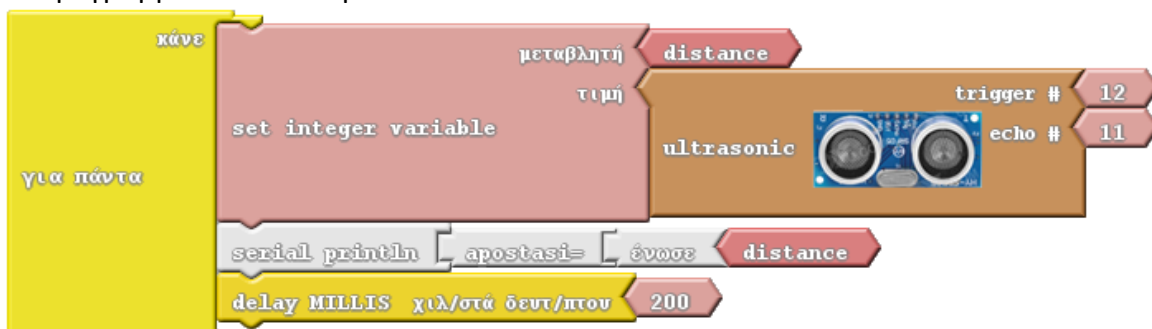
6^η Δραστηριότητα: Εμφάνιση μηνυμάτων- απόσταση από αντικείμενο

Στη συγκεκριμένη δραστηριότητα θα χρησιμοποιήσουμε το κύκλωμα της προηγούμενης δραστηριότητας και θα εμφανίζουμε στην οθόνη την απόσταση που επιστρέφει ο αισθητήρας υπερηχητικών σημάτων. Θα την εμφανίσουμε στη σειριακή οθόνη μέσα από το πρόγραμμα arduino. Χρησιμοποιούμε τη συνδεσμολογία του προηγούμενου κυκλώματος. Αν μας ενδιαφέρει απλά να εμφανίσουμε την απόσταση που επιστρέφει ο αισθητήρας μπορούμε να κρατήσουμε μόνο τον αισθητήρα και να αφαιρέσουμε το led.

Θα χρησιμοποιήσουμε την άσπρη καρτέλα «Επικοινωνία»



Το πρόγραμμα είναι το παρακάτω:





Η εντολή μας εμφανίζει στην οθόνη «apostasi=» και δίπλα την τιμή distance που επιστρέφει ο αισθητήρας. Κάθε φορά που εμφανίζεται μια τιμή αλλάζει και γραμμή λόγω της εντολής “println” που εισάγαμε. Χρησιμοποιήσαμε την εντολή καθυστέρησης για 200 χιλιοστά του δευτερολέπτου (δηλαδή ανά 0,2 sec) ώστε να μην εμφανίζονται συνέχεια οι τιμές, αλλά ανά 0,2 sec ώστε να προλαβαίνουμε να τα δούμε στην οθόνη. Μπορείτε να αλλάξετε την τιμή, να τη μεγαλώσετε, να την μικρύνετε και να συγκρίνετε σε ποια τιμή έχετε το αποτέλεσμα που θέλετε. Να θυμίσουμε ότι για να δείτε την σειριακή οθόνη πρέπει να πάτε από το

μενού του Arduino, Tools>>Σειριακή οθόνη ή να πατήσετε το



7^η Δραστηριότητα: Ανιχνευτής κίνησης- σύστημα συναγερμού

Σε αυτή τη δραστηριότητα θα χρησιμοποιήσουμε έναν αισθητήρα ανίχνευσης κίνησης ο οποίος όταν θα ανιχνεύει κίνηση τότε θα ενεργοποιεί έναν συναγερμό- buzzer. Ταυτόχρονα μπορούμε να ενεργοποιήσουμε και φωτάκια- leds που θα αναβοσβήνουν.

Για τη δραστηριότητα αυτή θα χρειαστούμε:

- Arduino uno
- Breadboard
- Αισθητήρα ανίχνευσης κίνησης
- Buzzer (12v Piezo Electronic Buzzer)
- Καλώδια
- Led- αντιστάσεις (αν θέλετε)

Λίγα λόγια για τον αισθητήρα ανίχνευσης κίνησης:

Ο αισθητήρας ανίχνευσης κίνησης έχει την δυνατότητα να ανιχνεύσει την κίνηση μέσα σε ένα δωμάτιο σε εμβέλεια έξι μέτρων. Ο αισθητήρας διαθέτει δύο μεταβλητές αντιστάσεις (trimmer) και μπορείτε να ρυθμίσετε την ευαισθησία και τον χρόνο ενεργοποίησης του από την στιγμή που θα ανιχνεύσει την κίνηση.

Χαρακτηριστικά:

- Τάση Εισόδου: 5 - 12 V DC
- Κατανάλωση: <60uA
- Σήμα Εξόδου: 3.3V TTL
- Απόσταση Ανίχνευσης: 3 - 6 μέτρα
- Γωνία Ανίχνευσης: <140°
- Χρόνος ενεργοποίησης: 5 - 200 sec
- Χρόνος καθυστέρησης: 2 - 4 sec

Ο ανιχνευτής κίνησης έχει 3 υποδοχές:

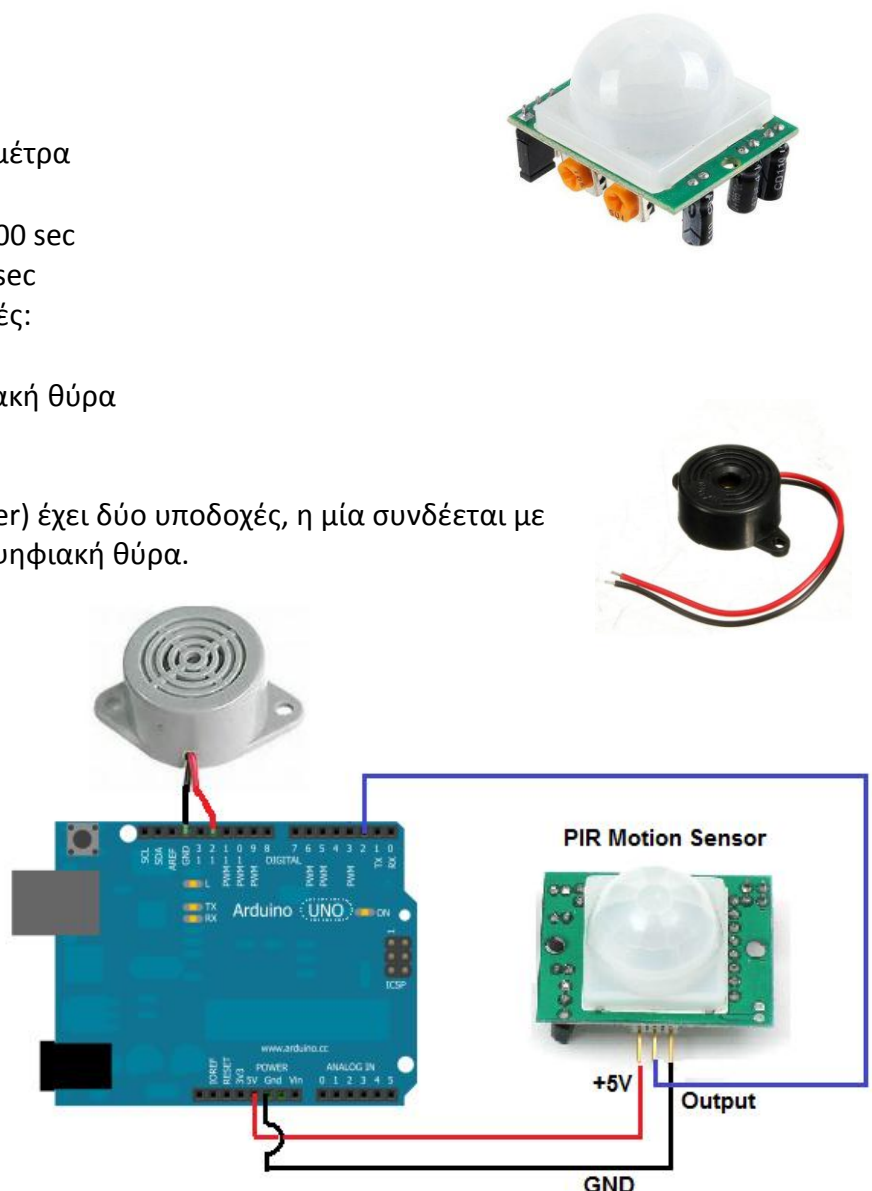
- GND – σύνδεση με γείωση
- OUT – σύνδεση με μια ψηφιακή θύρα
- 5V – σύνδεση με 5V

Το buzzer (12v Piezo Electronic Buzzer) έχει δύο υποδοχές, η μία συνδέεται με τη γείωση (GND) και η άλλη με μια ψηφιακή θύρα.

Η σύνδεση φαίνεται στην εικόνα δίπλα.

Ο αισθητήρας ανίχνευσης κίνησης συνδέεται με την pin2 και το Buzzer με την Pin12.

Ρυθμίζουμε την απόσταση που θέλουμε να ανιχνεύσει ο αισθητήρας ώστε να ενεργοποιηθεί.



Ας σκεφτούμε το πρόγραμμα:

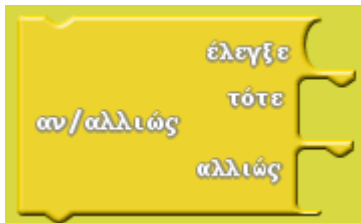
Αν η pin2 (που δίνει ο αισθητήρας κίνησης) έχει ΡΕΥΜΑ (ανιχνεύσει κίνηση)
τότε το pin12 ΡΕΥΜΑ (ενεργοποιούμε το buzzer)

Αλλιώς (δηλαδή pin2 δεν έχει ρεύμα)

τότε το pin12 ΟΧΙ ΡΕΥΜΑ (σταματά το buzzer)

Ανοίξτε το προγραμματιστικό περιβάλλον Ardublock και δημιουργήστε το πρόγραμμά σας. Στη συνέχεια ανεβάστε το στο arduino και ελέγξτε το.

Υλοποιείτε το πρόγραμμα στο Ardublock χρησιμοποιώντας την δομή επιλογής:



Μέσα στην «έλεγξε» πρέπει να χρησιμοποιήσουμε μια εντολή- ένα μπλοκ από τις καφέ εντολές- συνθήκες ελέγχου.



Μέσα θα πρέπει να τοποθετήσουμε την



δίνοντας τον κατάλληλο

αριθμό της ακίδας και



για να ελέγχουμε αν υπάρχει ρεύμα στην ψηφιακή ακίδα. Στο

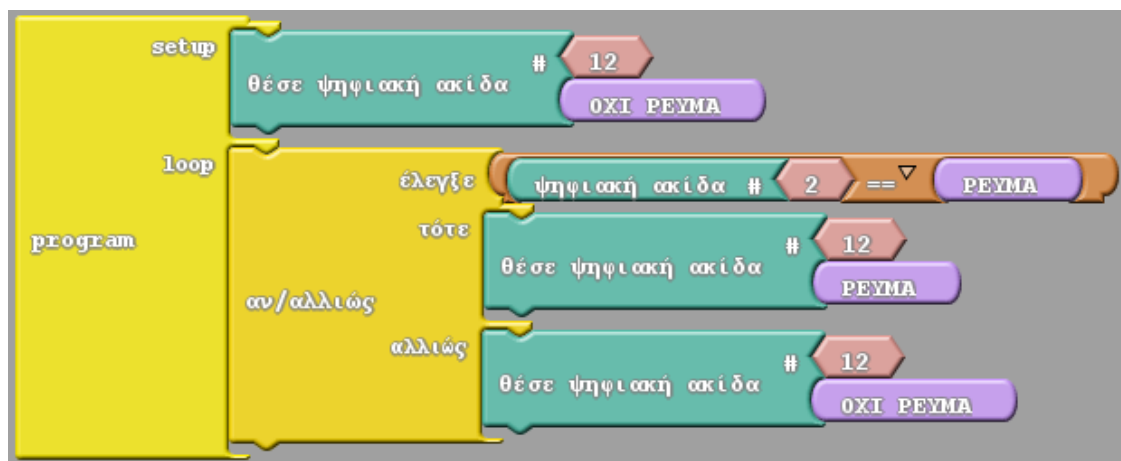


ΤΟΤΕ θα πρέπει να βάλουμε την εντολή

στην ψηφιακή έξοδο 12 την τιμή ΡΕΥΜΑ ή ΟΧΙ ΡΕΥΜΑ. Αυτές οι εντολές είναι στην καρτέλα με τις



μεταβλητές και τις σταθερές:



Το πρόγραμμα σε κώδικα ardublock

Δοκιμάστε να βάλετε στο κύκλωμά σας ένα ή περισσότερα led τα οποία να αναβοσβήνουν όταν λειτουργεί το buzzer. Αν τα συνδέσετε στα pin4 και pin5 συμπληρώστε κατάλληλα το παραπάνω πρόγραμμα.

8^η Δραστηριότητα: Αυτόματη ράμπα σε διάβαση τρένου

Στη δραστηριότητα αυτή θα κατασκευάσουμε μια αυτόματη ράμπα σε σιδηροδρομική διάβαση, που θα είναι όρθια για να περνάνε τα αυτοκίνητα και θα κατεβαίνει όταν θα έρχεται τρένο. Μπορούμε να προσθέσουμε στην προηγούμενη κατασκευή μας έναν σερβοκινητήρα ώστε όταν το τρένο έρθει σε κοντινή απόσταση (πχ 50 εκ) τότε να κατεβαίνει η ράμπα και να ανάβει το λαμπάκι.

Θα χρειαστούμε:

- 1 Arduino Uno
- 1 Breadboard
- 1 Αισθητήρα απόστασης Ultrasonic Sensor
- 1 σερβοκινητήρα
- 1 αντίσταση 1 Ωhm
- Ξυλάκια γλωσσοπίεστρες ή καλαμάκι για την μπάρα. Μπορείτε να κολλήσετε τη γλωσσοπίεστρα ή το καλαμάκι στην υποδοχή του σερβοκινητήρα.
- Ένα τρένο, ένα αυτοκινητάκι κτλ, διαφορετικά χρησιμοποιούμε κάποιο αντικείμενο

Στο κύκλωμα της προηγούμενης δραστηριότητας συνδέουμε και έναν σερβοκινητήρα.

Για τον αισθητήρα υπερηχητικών:

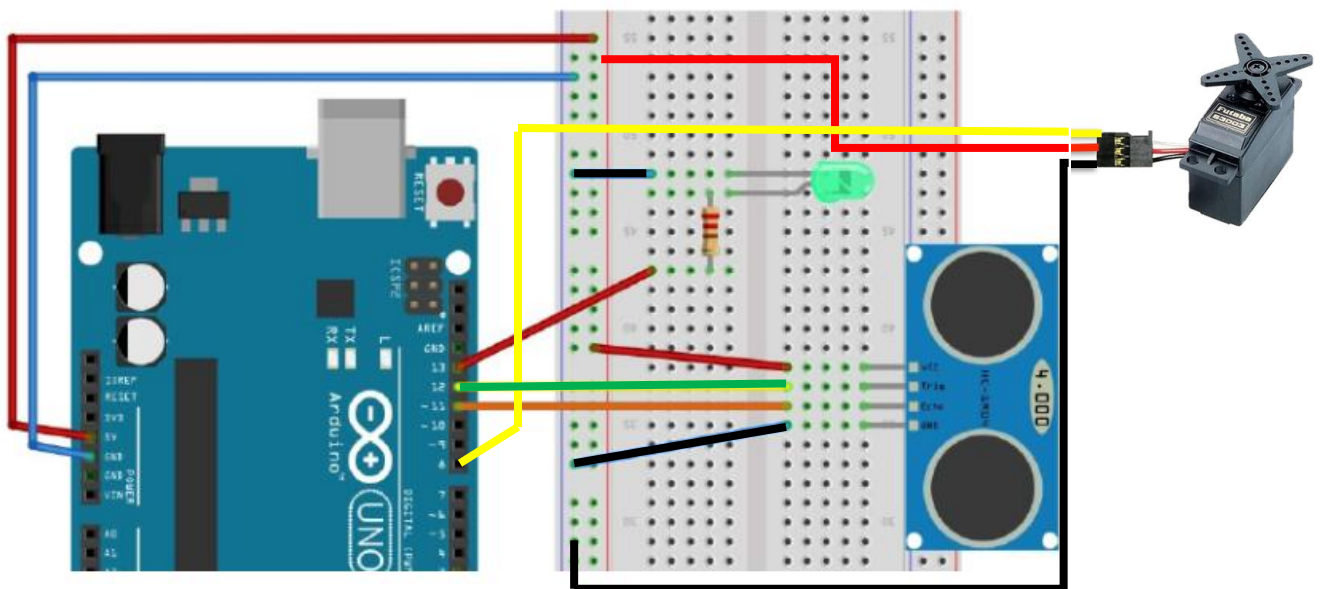
Trig->pin12

Eccho ->pin11

Led -> pin13

Για τον σερβοκινητήρα: Servo -> pin8

Σημείωση: Τα παραπάνω είναι ενδεικτικά, μπορείτε να συνδέσετε τα εξαρτήματα σε όποια ακίδα επιθυμείτε



Ας καταγράψουμε τι θέλουμε να κάνει το πρόγραμμά μας.

- Η ράμπα (θέση σερβοκινητήρα) είναι σε όρθια θέση (0 ή 180 μοίρες- θα πρέπει να κάνετε έλεγχο) και το λαμπάκι σβηστό.
- Θα ελέγχει συνεχώς την απόσταση από ένα σταθερό σημείο από όπου θα περάσει το τρένο.
- Αν η απόσταση αυτή γίνει μικρότερη από έναν αριθμό πχ από 20 εκ (μπορείτε να κάνετε δοκιμές) τότε η ράμπα (σερβοκινητήρας) να είναι σε οριζόντια θέση (θέση 90 μοιρών) και να ανάψει το λαμπάκι.

- Αν η απόσταση είναι μεγαλύτερη από 20 εκ τότε θα ανέβει η ράμπα πάλι και θα σβήνει το λαμπάκι.

Άρα:

Αρχικά:

Γωνία= 0 ή 180

Led σβηστό

Συνεχώς:

Αν απόσταση ≤ 20 τότε

Led ανοιχτό

Γωνία 90 μοίρες

Αλλιώς

Led σβηστό

Γωνία 0 ή 180 μοίρες

Χρησιμοποιώντας τις δομές επιλογής που γνωρίζεται και τις εντολές ελέγχου γράψτε το πρόγραμμα στο ardublock. Αποθηκεύστε το στο φάκελό σας.

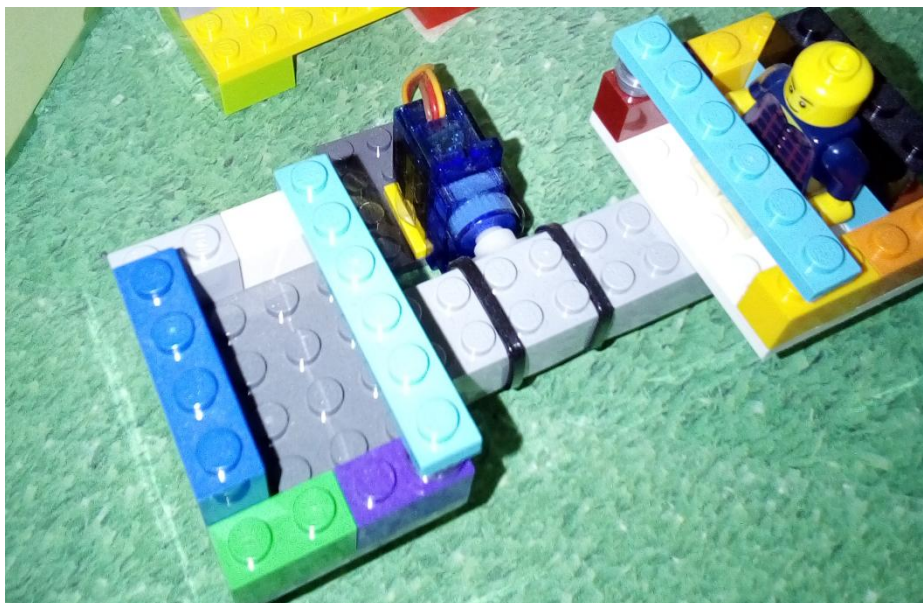
9^η Δραστηριότητα: Αυτόματη τραμπάλα σε παιδική χαρά που λειτουργεί με το άνοιγμα και το κλείσιμο ενός διακόπτη On-Off

Σε αυτή τη δραστηριότητα θα κατασκευάσουμε με διάφορα υλικά όπως τουβλάκια τύπου lego ή άλλα υλικά όπως χαρτόνι ή μακετόχαρτο, γλωσσοπιέστρες κτλ μια τραμπάλα. Θα τη συνδέσουμε με έναν σερβοκινητήρα ο οποίος θα περιστρέφει την κατασκευή μας πάνω κάτω ώστε να προσομοιάζει την κίνηση της τραμπάλας. Η τραμπάλα θα είναι αρχικά σε οριζόντια θέση ώστε να μπορεί κάποιος να ανέβει ή να κατέβει και μόλις πατήσουμε έναν διακόπτη On/Off τότε θα αρχίζει αυτή να κινείται πάνω κάτω. Όταν θα πατήσουμε το διακόπτη για να σταματήσει τότε θα επιστρέφει στην αρχική οριζόντια θέση της.

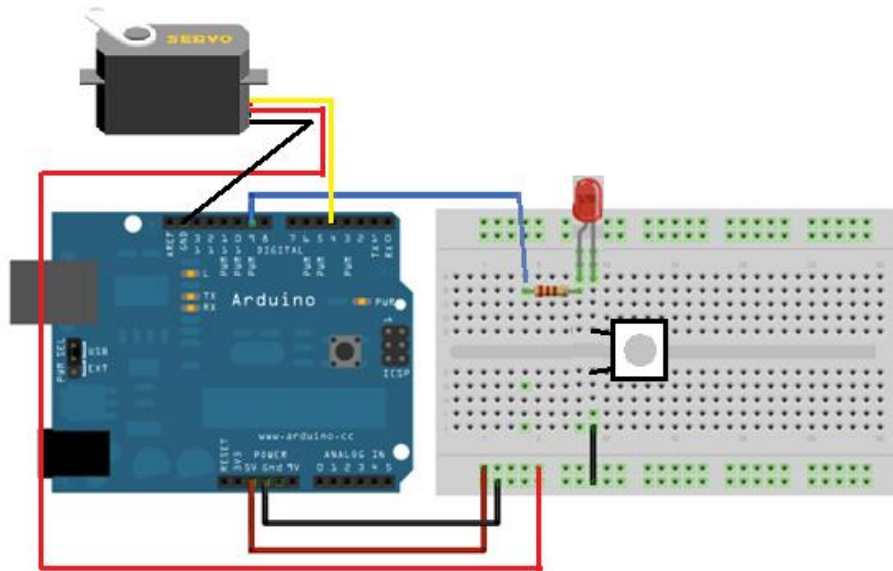
Για την κατασκευή αυτή εκτός από τα υλικά της τραμπάλας θα χρειαστούμε και τα παρακάτω:

- Arduino uno
- Διακόπτη on/off
- Led
- Αντίσταση 220-330 Ω m
- Breadboard
- Σερβοκινητήρα
- Καλώδια

Συνδέουμε τον σερβοκινητήρα με την τραμπάλα σε οριζόντια θέση. Δοκιμάζουμε να τον περιστρέψουμε ώστε να είμαστε σίγουροι ότι τον τοποθετήσαμε σωστά.



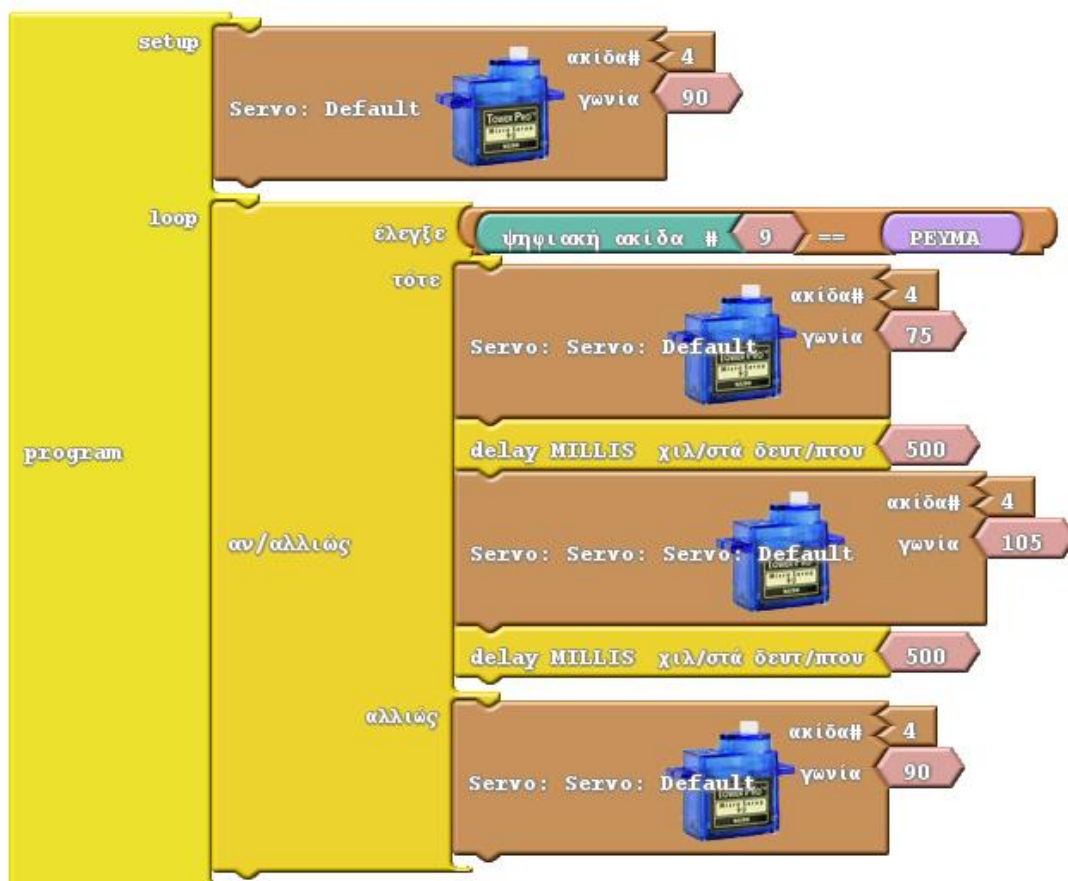
- Συνδέω τον σερβοκινητήρα με την pin4. Η οριζόντια θέση θα είναι στις 90° και θα ανεβοκατεβαίνει στις $\pm 15^\circ$ δηλαδή στις 75° και στις 105° . Βέβαια, αυτές οι τιμές εξαρτώνται από την κατασκευή σας ώστε να μη χτυπάει η τραμπάλα στο έδαφος. Θα κάνετε εσείς τον έλεγχο και θα δώσετε τις σωστές τιμές.
- Συνδέω την ακίδα pin9 στη μια άκρη του led ώστε να ελέγχω αν διαρρέεται από ρεύμα ή όχι.



Έτσι έχω το παρακάτω πρόγραμμα

Αρχικά Γωνία σερβοκινητήρα= 90 (οριζόντια θέση)
Αν pin9=PEYMA **τότε**
 Γωνία σερβοκινητήρα= 75
 Περίμενε 0,5 δευτερόλεπτα
 Γωνία σερβοκινητήρα= 105
 Περίμενε 0,5 δευτερόλεπτα
Αλλιώς
 Γωνία σερβοκινητήρα= 90 (οριζόντια θέση)

Παρακάτω είναι το πρόγραμμα στο Ardublock



10^η Δραστηριότητα: Διακόπτης πίεσης (Push button) για τη λειτουργία led

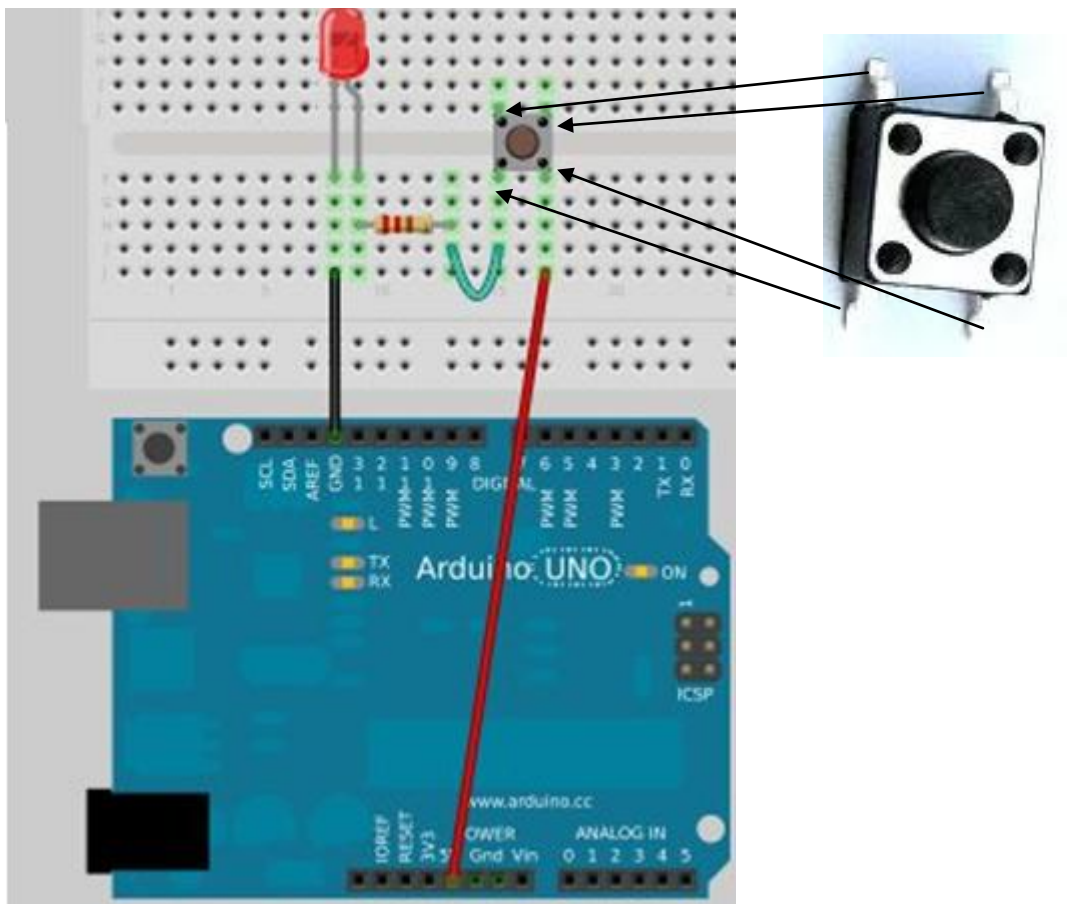
Σε αυτή τη δραστηριότητα θα συνδέσουμε έναν διακόπτη push button τον οποίο όταν τον κρατάμε πατημένο τότε θα ανάβει το λαμπάκι και όταν θα τον αφήνουμε τότε θα σβήνει.

Τα push button ενώνουν δύο σημεία σε ένα κύκλωμα όταν τα πατήσουμε. Έτσι «κλείνει» το κύκλωμά μας.

Για τη δραστηριότητα αυτή θα χρειαστούμε

- Arduino uno
- Breadboard
- Led
- Push button
- Αντίσταση 10KΩ

Το push button έχει 4 ποδαράκια, φροντίστε να τα τοποθετήσετε όπως παρακάτω:



Έτσι, το led θα είναι αναμμένο όταν πατήσουμε το push button αφού έτσι θα κλείσει το κύκλωμα και θα διαρρέει ρεύμα το led και το led θα είναι σβηστό όταν αφήνω το push button.

11^η Δραστηριότητα: Διακόπτης λειτουργίας led (Push button) σαν On/Off διακόπτης

Στην προηγούμενη δραστηριότητα χρειαζόταν να πατάμε συνεχώς το διακόπτη ώστε να ανάβει το led καθώς είχαμε συνδέσει το διακόπτη και το led σε σειρά και διαρρέονταν από το ίδιο ρεύμα.

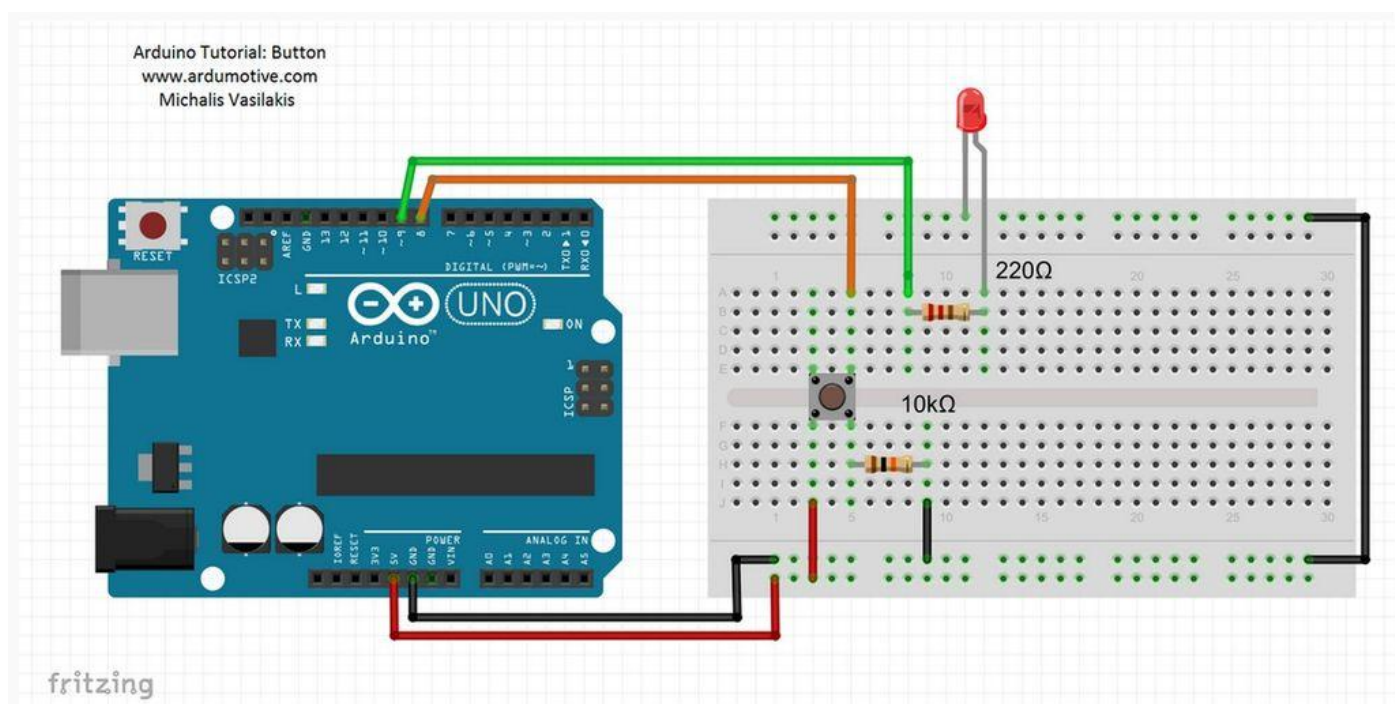
Τώρα θα δοκιμάσουμε κάτι άλλο, που συναντάμε συχνά και στο σπίτι μας. Φανταστείτε τον διακόπτη λειτουργίας της τηλεόρασης από το τηλεχειριστήριο. Είναι ένας διακόπτης πίεσης όπου τον πατάμε μια φορά και είτε ανάβει είτε κλείνει η τηλεόραση και στη συνέχεια αυτός έρχεται στην αρχική του θέση. Φανταστείτε ότι θα έπρεπε να τον πατούσαμε συνεχώς για να είναι ανοιχτή η τηλεόραση. Το ίδιο ισχύει για τον διακόπτη του πλυντηρίου κτλ. Θέλουμε λοιπόν να δημιουργήσουμε ένα κύκλωμα όπου όταν πατάμε τον διακόπτη πίεσης τότε να αλλάζει κατάσταση το led μας, δηλαδή αν είναι κλειστό τότε να ανάβει και το αντίστροφο.

Όταν πατάμε το Push button τότε κλείνει το κύκλωμα, περνάει ρεύμα από αυτόν ενώ όταν τον αφήνουμε τότε δεν περνάει ρεύμα από τον διακόπτη. Μπορούμε να χρησιμοποιήσουμε αυτές τις «αλλαγές» στην κατάσταση ρεύματος του διακόπτη και έτσι να μπορέσουμε να ανάψουμε και να σβήσουμε αντίστοιχα ένα led.

Για τη δραστηριότητα αυτή θα χρειαστούμε

- Arduino uno
- Breadboard
- Led
- Push button
- Αντίστασεις 10KΩ και 220 Ωm
- καλώδια

Συνδέστε τα εξαρτήματα σύμφωνα με την παρακάτω συνδεσμολογία



Ας σκεφτούμε τι θέλουμε να κάνουμε:

- Αρχικά το led θα είναι κλειστό.
- Όταν τα πατήσω μια φορά το διακόπτη τότε θα ανάψει το led και θα μείνει αναμμένο.
- Όταν πατήσω εκ νέου το διακόπτη τότε θα σβήσει το led και θα παραμείνει σβηστό.
- Κ.ο.κ

Το led δηλαδή θα ανάβει και θα σβήνει κάθε φορά που θα πατάμε τον διακόπτη, θα λειτουργούμε πλέον τον διακόπτη σαν on/off.

Για να το κάνουμε αυτό δεν θα χρησιμοποιήσουμε την κατάσταση του ρεύματος του διακόπτη αλλά θα χρησιμοποιήσουμε τις αλλαγές της κατάστασής του. Δηλαδή κάθε φορά που θα αλλάζει κατάσταση η ακίδα που είναι συνδεδεμένη με τον διακόπτη (από όχι ρεύμα σε ρεύμα και αντίστροφα) τότε θα αλλάζει και κατάσταση το led (από όχι ρεύμα σε ρεύμα και αντίστροφα).

Όταν το push button δεν είναι πατημένο (ανοιχτό κύκλωμα) τότε η ακίδα στην οποία συνδέουμε το push button έχει την τιμή low- όχι ρεύμα. Όταν το push button πατηθεί (κλειστό κύκλωμα) τότε η ακίδα στην οποία συνδέουμε το Push button έχει την τιμή high-ρεύμα.

Όταν Pin8= ρεύμα (πατήσω τον διακόπτη) τότε

Αν pin9= ρεύμα τότε

Pin9= όχι ρεύμα (άλλαξε την κατάσταση και σταμάτα ρεύμα)

Αλλιώς (δηλαδή pin9= όχι ρεύμα)

Pin9= ρεύμα (άλλαξε την κατάσταση και δώσε ρεύμα)

Σημειώστε εδώ το πρόγραμμα σε ardublock:

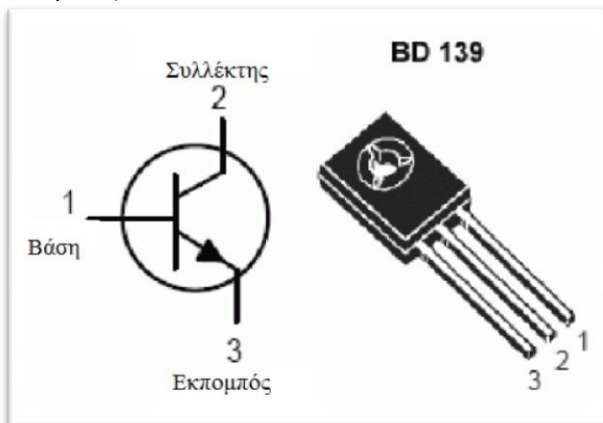
12^η Δραστηριότητα: Σύνδεση κινητήρα- αυξομείωση έντασης περιστροφής

Σε αυτή τη δραστηριότητα θα βάλουμε έναν κινητήρα σε ένα breadboard και θα αυξάνουμε και θα μειώνουμε σταδιακά την ένταση περιστροφής του. Φανταστείτε έναν ανεμιστήρα που περιστρέφεται με διάφορες ταχύτητες, μια ρόδα αυτοκινήτου που γυρίζει με διάφορες ταχύτητες κτλ.

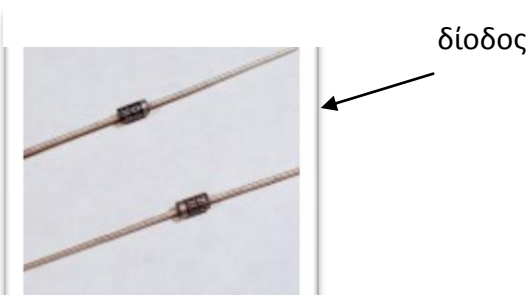
Για να υλοποιήσουμε το κύκλωμα θα χρειαστούμε

- ένα μοτοράκι (dc motor) ή έναν κινητήρα (stepper motor)
- μια αντίσταση
- μια δίοδο
- ένα τρανζίστορ.
- γέφυρα L298N
- ένα breadboard μικρό ή μεσαίο
- καλώδια

Το βασικό χαρακτηριστικό του τρανζίστορ είναι ότι μας επιτρέπει να μεταβάλλουμε την αντίσταση σε ένα κύκλωμα, καθώς μπορεί να ρυθμίζει την ροή του ρεύματος που απορροφά, ανάλογα με την τάση που δέχεται. Αυτό το χαρακτηριστικό του, το καθιστά ιδανικό για ενίσχυση ή σταθεροποίηση τάσης, για διαμόρφωση συχνοτήτων, ενώ έχει και αρκετές εφαρμογές ως διακόπτης ή ως μεταβλητή αντίσταση. Το arduino μπορεί να παρέχει μόνο 40mA ρεύμα στα ψηφιακά του Pins. Οι περισσότεροι κινητήρες χρειάζονται περισσότερο ρεύμα ή και τάση για να λειτουργήσουν. Ένα τρανζίστορ μπορεί να λειτουργήσει σαν ψηφιακός διακόπτης αφήνοντας το arduino να ελέγχει τη ροή σε εξαρτήματα που θέλουν περισσότερο ρεύμα. Τα τρανζίστορ έχουν 3 ακίδες που ονομάζονται βάση, συλλέκτης και εκπομπός.



PN2222 Transistor

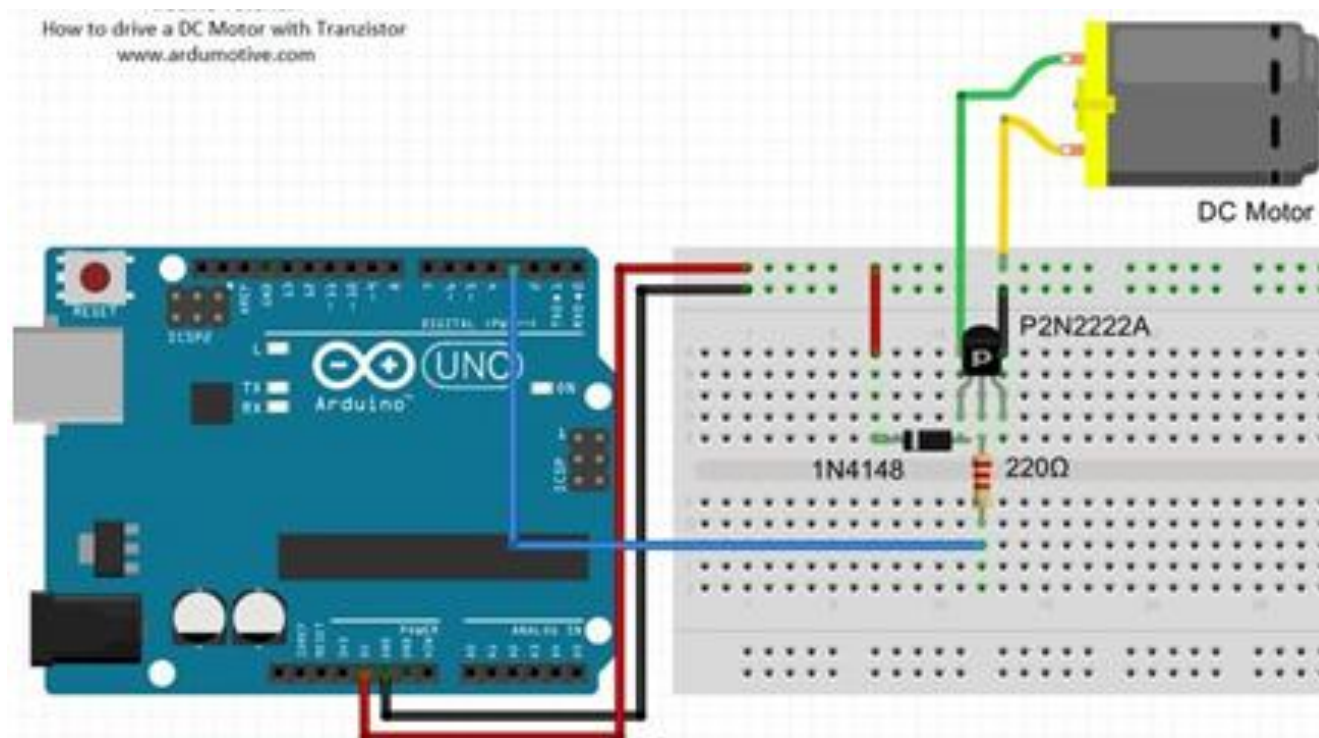


Επίσης, θα χρειαστούμε μια δίοδο. Η δίοδος προστατεύει από απότομες αλλαγές τάσης ώστε να μην καταστραφεί το τρανζίστορ.



220 Ω Resistor

Το κύκλωμα φαίνεται παρακάτω:



Το τρανζίστορ έχει ένα καμπυλωτό σχήμα, η μια μεριά είναι ίσια. Όπως κοιτάτε την ίσια μεριά έχει δυο άκρες, την αριστερή και τη δεξιά. Επιπλέον έχει ένα μεσαίο ποδαράκι.

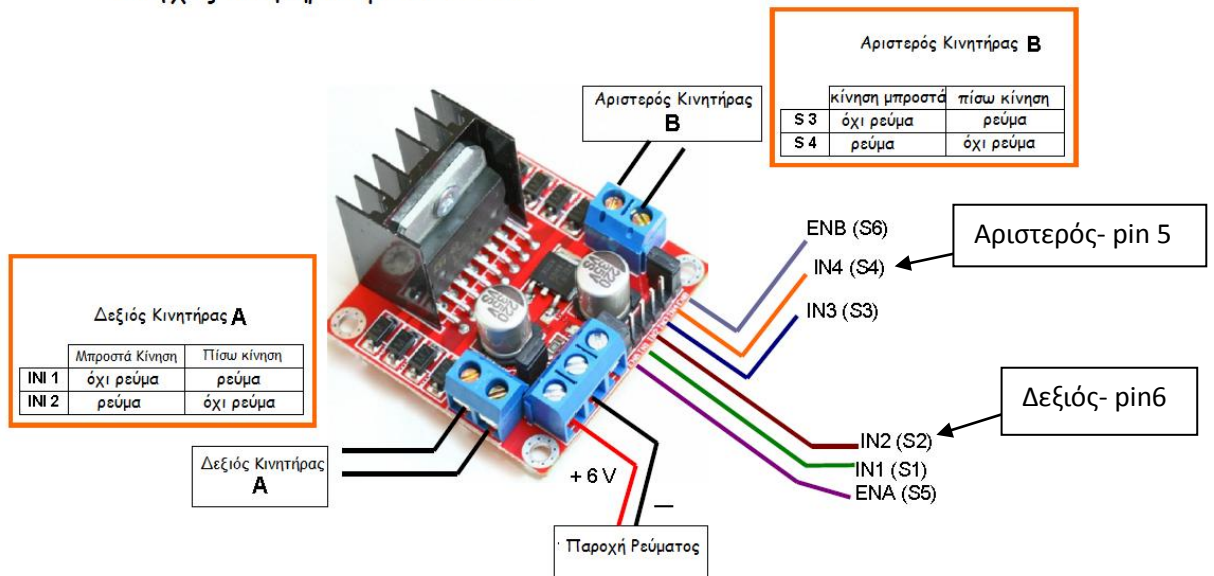
Οι δυο άκρες του κινητήρα μπαίνουν στα ακριανά πόδια του τρανζίστορ. Την αριστερή θα τη συνδέσετε με τη δίοδο και στη συνέχεια με την ακίδα 5V. Τη δεξιά, θα τη συνδέσετε με τη γείωση. Στο μεσαίο πόδι μπαίνει η αντίσταση και στη συνέχεια συνδέουμε με αναλογική/ψηφιακή έξοδο. (προσέχουμε να έχει το σήμα ~ πάνω της, δηλαδή, την 5, 6, 10 ή 11.

Σύνδεση κινητήρα με γέφυρα L298N

Για να ελέγξουμε έναν, 2 κινητήρες μπορούμε να χρησιμοποιούμε μια γέφυρα L298N. Αυτό γίνεται για να προστατεύσουμε τους κινητήρες από καταστροφή. Η γέφυρα στα πλάγια έχει από 2 μπλε υποδοχές για να μπουν τα καλώδια από τους κινητήρες. Στο μπροστινό μέρος έχει 3 μπλε υποδοχές, οι δύο ακριανές είναι αυτές που μπαίνουν στο ρεύμα, η πρώτη στα 5V ή στο θετικό πόλο μιας μπαταρίας και η δεύτερη στη γείωση ή στον αρνητικό πόλο μιας μπαταρίας.

Επιπλέον, στο μπροστινό μέρος υπάρχουν 4 μεταλλικές ακίδες οι οποίες ελέγχουν 4 ψηφιακές εξόδους. Μπορούμε να συνδέσουμε 2 κινητήρες και να τους κινήσουμε μπρος και πίσω. Βλέπετε στο παρακάτω σχεδιάγραμμα ποιες ακίδες δίνουν κίνηση μπρος και ποιες πίσω.

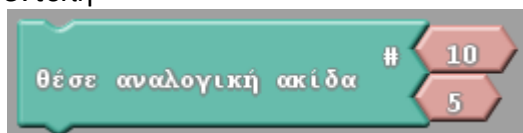
Έλεγχος Κινητήρων με το L298N



Εμείς θέλουμε να κινήσουμε προς μια κατεύθυνση τον κινητήρα ώστε να περιστρέφει την κατασκευή μας. Έτσι, θα συνδέσουμε τον κινητήρα στο δεξί μέρος και θα χρησιμοποιήσουμε την 1^η προς τα δεξιά ακίδα, δίπλα στον κινητήρα, ώστε να τον κινούμε. Θυμόμαστε να το συνδέσουμε σε μια αναλογική έξοδο (ακίδα 3).

Αργότερα που θα συνδέσουμε και δεύτερο κινητήρα, θα τον βάλουμε στο αριστερό μέρος και θα χρησιμοποιήσουμε την 3 από δεξιά προς τα αριστερά ακίδα από τη γέφυρα για να τον γυρίσουμε.

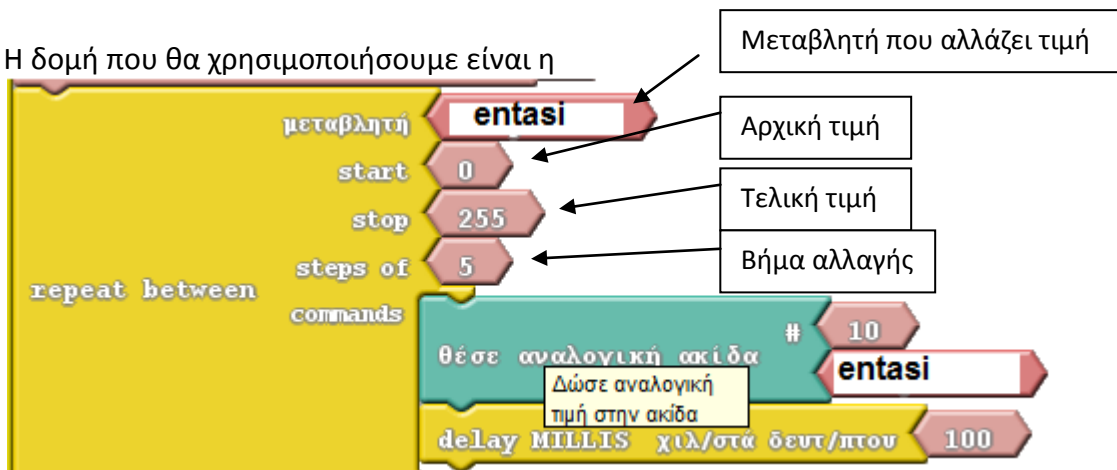
Όπως είπαμε, μια ψηφιακή μεταβλητή είναι μια μεταβλητή που μπορεί να πάρει μόνο δύο τιμές ενώ αναλογική είναι αυτή που μπορεί να πάρει πολλές τιμές, μέσα από ένα εύρος τιμών. Για παράδειγμα, ένας κινητήρας μπορεί να περιστραφεί με δύναμη από 0 έως 255 ενώ στην ψηφιακή τιμή μπορεί να είναι 0 (δεν περιστρέφεται ο κινητήρας) ή 255 (περιστρέφεται με τη μέγιστη ταχύτητα περιστροφής). Έτσι αν θέλω να περιστρέφεται αργά μπορώ να έχω μια τιμή πχ 100 κτλ. Άρα θα χρησιμοποιήσουμε την εντολή



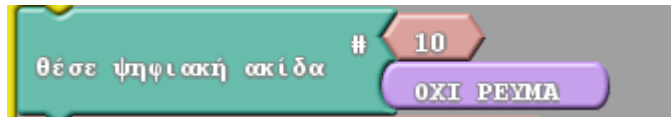
για κάποιο χρόνο (εντολή delay)

Στη συνέχεια αν θέλουμε να αυξήσουμε την ταχύτητα, θα αλλάξουμε την τιμή 5 σε 10, 15, 20,255. Για να το πετύχουμε αυτή θέλουμε 255/5 φορές, κάτι που είναι πολύ χρονοβόρο. Έτσι, χρειαζόμαστε μια δομή επανάληψης για να το πετύχουμε αυτό.

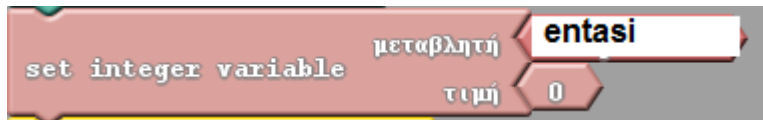
Η δομή που θα χρησιμοποιήσουμε είναι η



Αρχικά όμως θα πρέπει να δηλώσουμε τι τιμή θα έχει η ένταση περιστροφής του κινητήρα μας και να τον ενεργοποιήσουμε. Αυτό γίνεται με την εντολή που μάθαμε στην 1^η δραστηριότητα

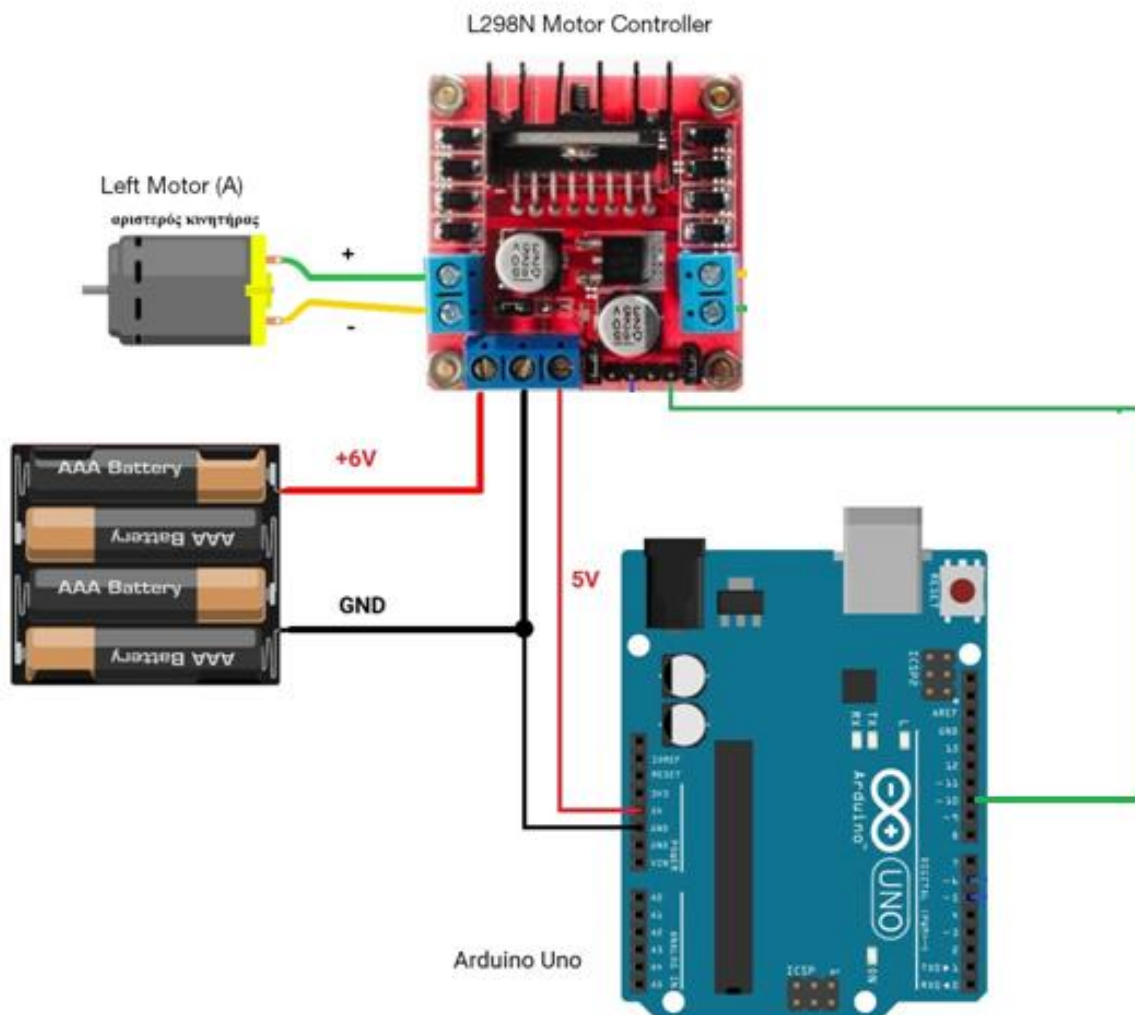


Και να θέσουμε αρχική τιμή 0 στη μεταβλητή entasi (ένταση περιστροφής κινητήρα)



που βρίσκεται στην καρτέλα «μεταβλητές».

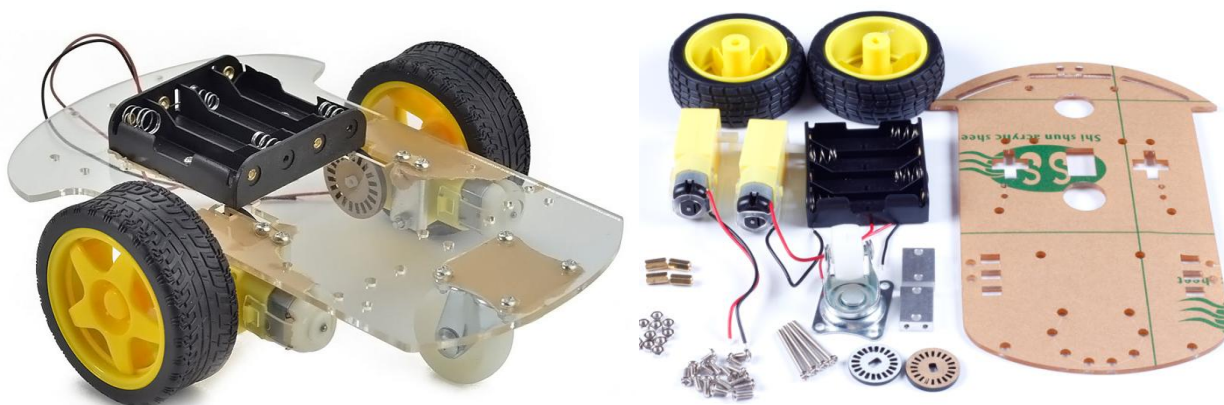
Σημείωση: Καλό είναι η ταχύτητα να μη πέφτει κάτω από μια τιμή πχ 70 γιατί δυσκολεύεται ο κινητήρας. Δοκιμάστε το για να δείτε σε ποια τιμή και κάτω δεν περιστρέφεται.



Φροντίστε να δώσετε ρεύμα στο arduino και στη γέφυρα με δύο ξεχωριστές παροχές. Το arduino μπορείτε να το συνδέσετε με τον υπολογιστή ή με μια άλλη παροχή ρεύματος.

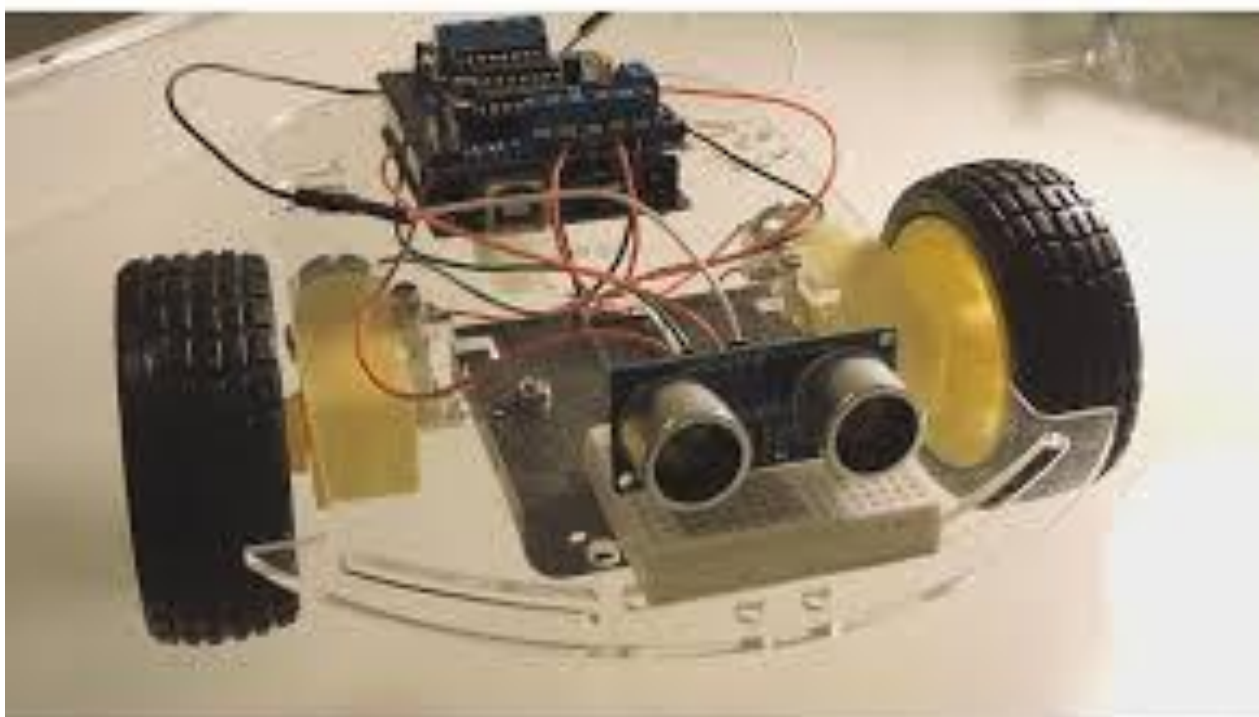
13^η Δραστηριότητα: Αυτόματο ρομποτικό όχημα που κινείται ευθεία και σταματά όταν βρει εμπόδιο.

Σε αυτή τη δραστηριότητα θα κατασκευάσουμε ένα όχημα που θα κινείται σε 3 ρόδες, οι 2 θα κινούνται με κινητήρες. Μπορούμε να αγοράσουμε και να τοποθετήσουμε 2 κινητήρες και 2 ρόδες σε ένα χαρτόνι, κουτί, να εκτυπώσουμε ένα σασί σε έναν τρισδιάστατο εκτυπωτή, να χρησιμοποιήσουμε ένα έτοιμο αυτοκίνητο που έχουμε σπίτι ή να αγοράσουμε ένα σασί όπου κουμπώνουν εύκολα τα εξαρτήματα του arduino. Ένα τέτοιο είναι το παρακάτω:



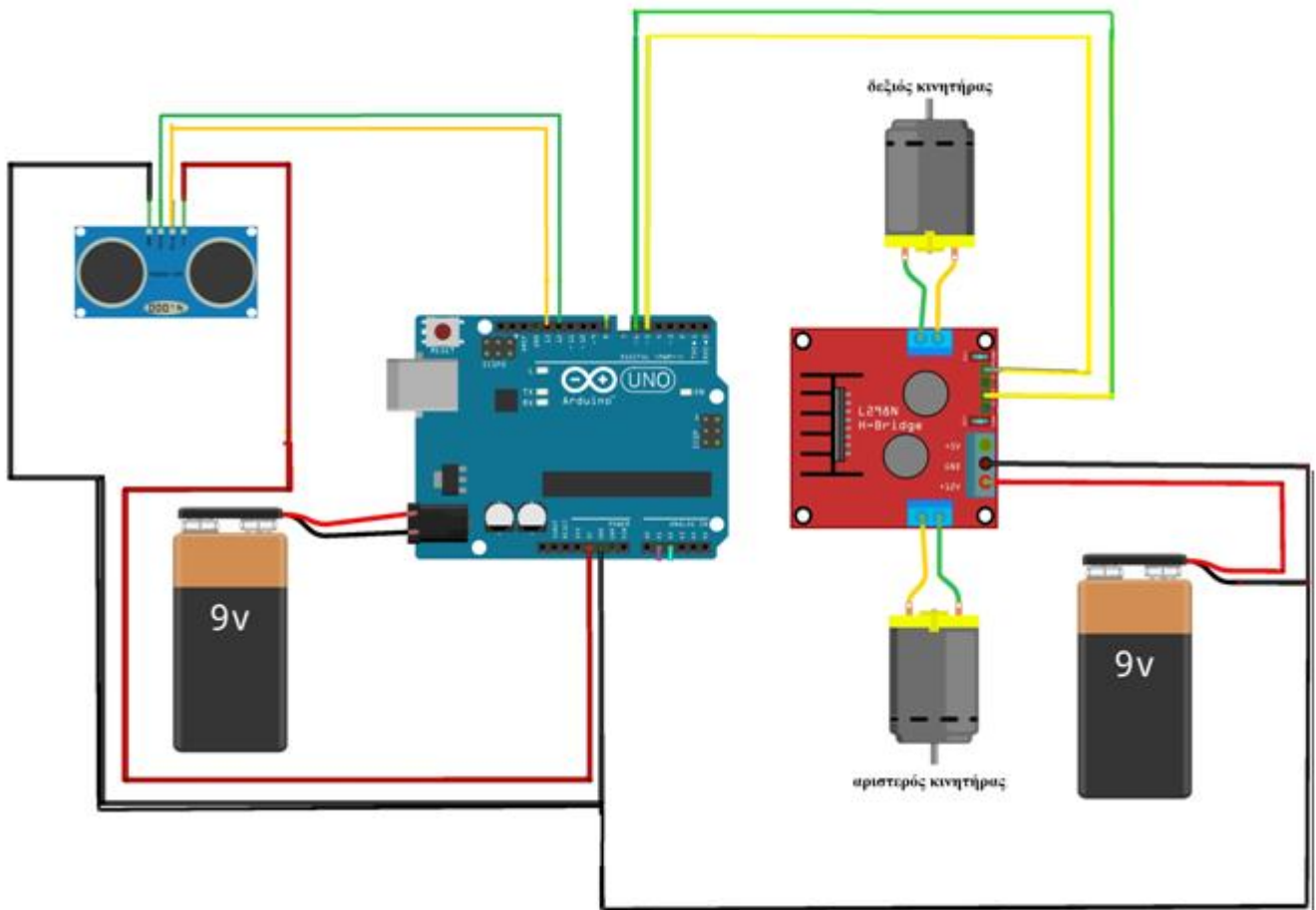
Το πακέτο διαθέτει 2 κινητήρες, βοηθητική ρόδα, σασί, βάση 4ων μπαταριών, βίδες κτλ.

Αφού κατασκευάσουμε και συναρμολογήσουμε το αυτοκίνητο (μπορείτε να δείτε πολλά βίντεο στο youtube κάνοντας αναζήτηση πχ «assemble 2WD Smart Robot Car Chassis Kit for Arduino») μένει να τοποθετήσουμε τη γέφυρα που θα ελέγχει τους κινητήρες και το arduino και να συνδέσουμε τον αισθητήρα υπερηχητικών σημάτων που θα ελέγχει την απόσταση. Το αυτοκινητάκι μας φαίνεται παρακάτω.



Ας σκεφτούμε τι θέλουμε να κάνει το αυτοκίνητό μας.

- Να κινείται συνέχεια ευθεία δηλαδή οι δύο κινητήρες να κινούνται με ίδια ταχύτητα.
- Όταν ο αισθητήρας υπερηχητικών που υπάρχει στο μπροστινό του μέρος (μάτια) ανιχνεύσει ένα εμπόδιο τότε το αυτοκίνητο σταματά δηλαδή οι κινητήρες έχουν μηδενική ταχύτητα.
- Μπορούμε τους κινητήρες όπως είδαμε να τους συνδέσουμε σε δύο αναλογικές εισόδους ώστε να κινούνται με μικρότερη ταχύτητα (ακίδες 5 και 6).

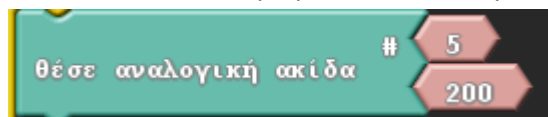


Θα χρησιμοποιήσουμε και ένα breadboard ώστε να μπορούμε εύκολα να συνδέσουμε γειώσεις GND και 5V.

Ας προσπαθήσουμε να μεταφράσουμε τα παραπάνω σε εντολές:

Ο αριστερός κινητήρας είναι ο 5 και ο δεξιός ο 6. Η εντολή που θα χρησιμοποιήσουμε για να τους

δώσουμε κίνηση είναι η

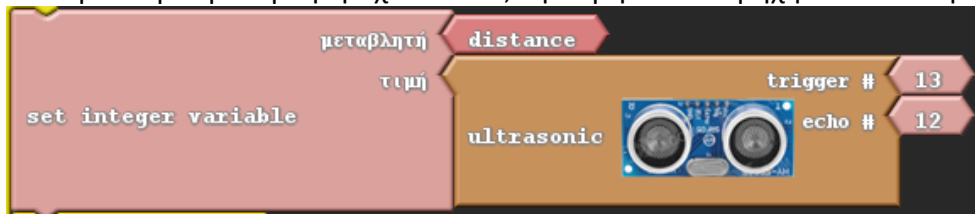


Όπως βλέπετε χρησιμοποιήσαμε την εντολή «θέσε αναλογική ακίδα» και όχι «θέσε ψηφιακή ακίδα» γιατί θέλουμε το όχημά μας να κινείται με λιγότερη ταχύτητα από τη μέγιστη (255). Και για το λόγο αυτό συνδέσαμε με τις ακίδες 5 και 6 που λειτουργούν και ως αναλογικές. Μπορούμε να βάλουμε και μικρότερη ταχύτητα και να κάνουμε ελέγχους.

Αν θέλουμε να σταματήσει ένας τροχός τότε συμπληρώστε την παρακάτω εντολή με τιμή 0 έως 255:



Θέτουμε σε μια μεταβλητή πχ distance, την τιμή του υπερηχητικού αισθητήρα



και την ελέγχουμε αν είναι μικρότερη ή μεγαλύτερη από μια τιμή πχ 25εκ.



Αν είναι μεγαλύτερη τι θα κάνει το αυτοκινητάκι μας;.....

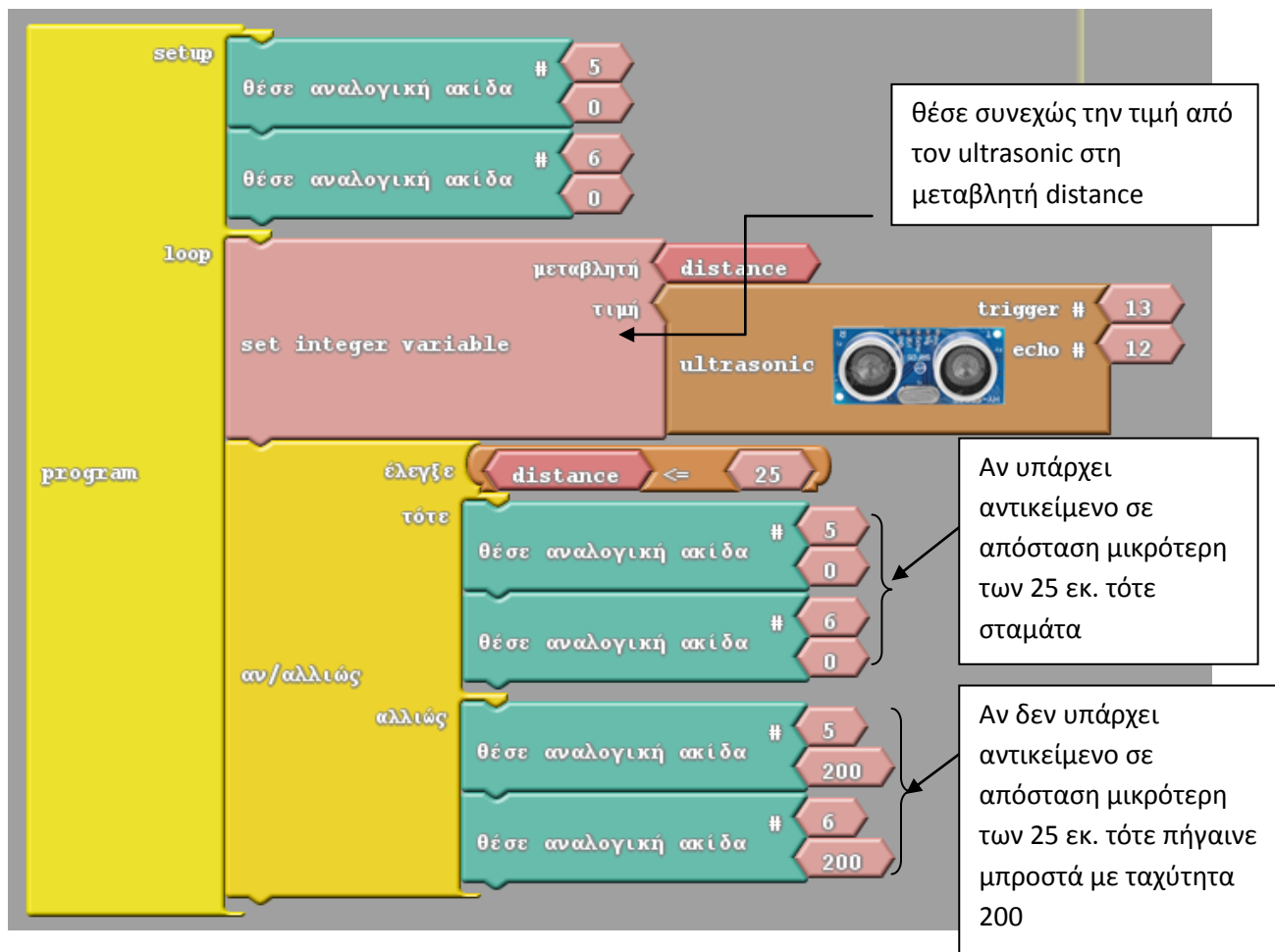
.....

Αν είναι μικρότερη;.....

.....

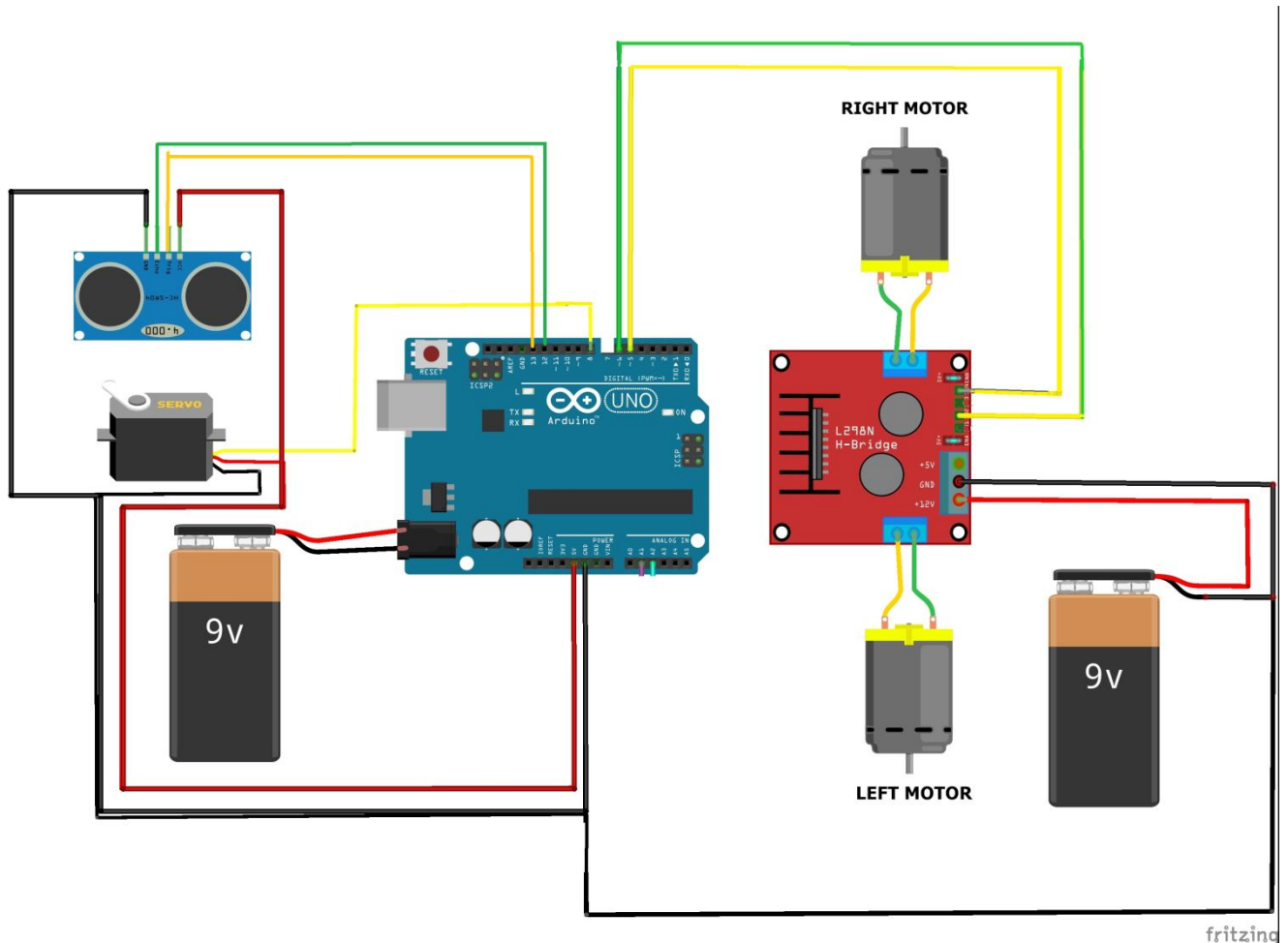
Γράψτε τα παραπάνω στο πλαίσιο και τοποθετήστε τα σε μια δομή ελέγχου Αν...Αλλιώς.

Το πρόγραμμα φαίνεται παρακάτω:

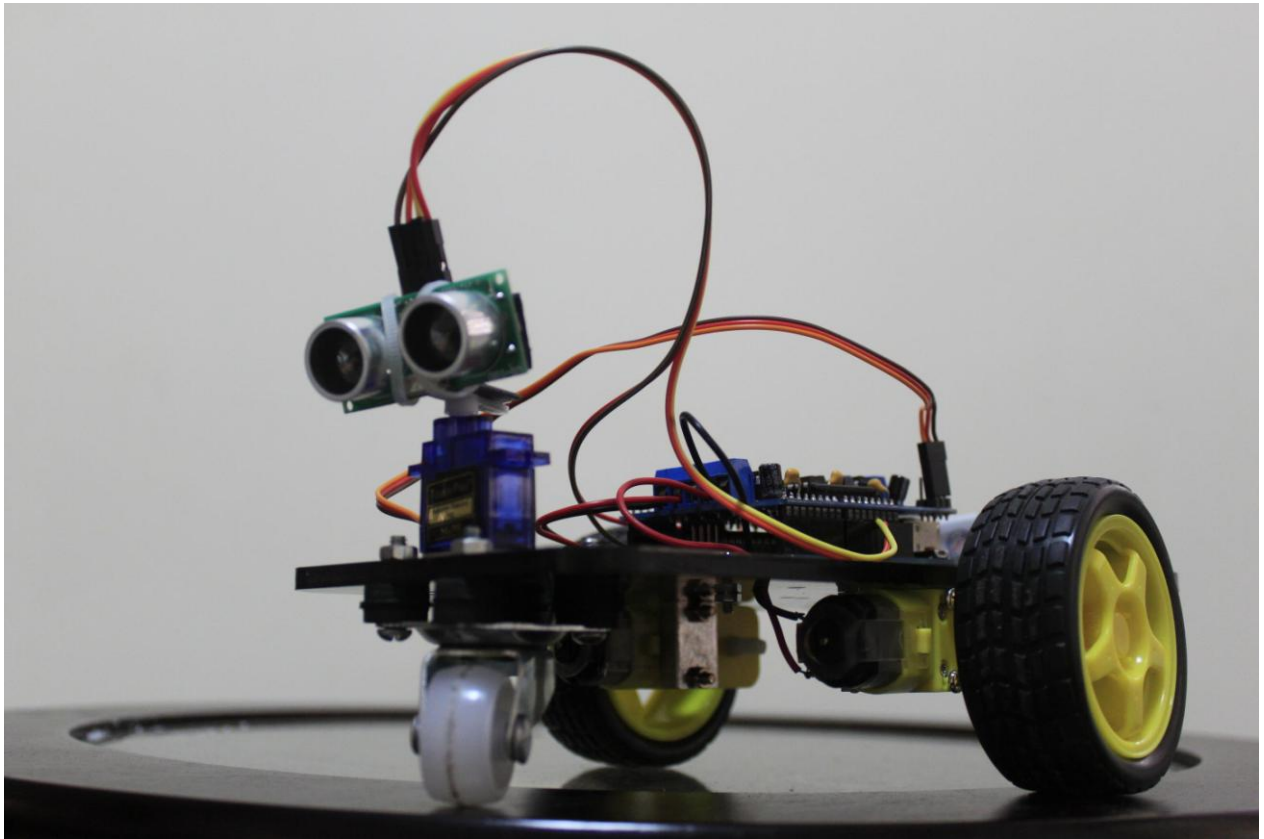


14^η Δραστηριότητα: Αυτόματο ρομποτικό όχημα που αλλάζει πορεία όταν συναντήσει εμπόδιο

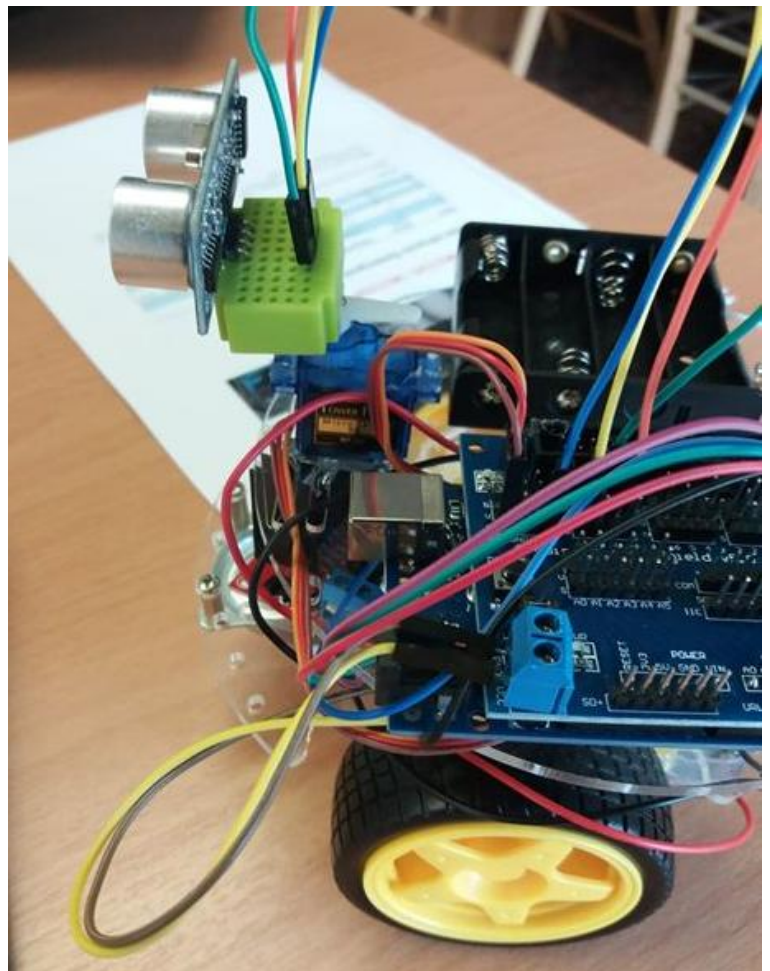
Η προηγούμενη προσπάθεια μας ήταν να κάνουμε το ρομπότ μας να σταματάει κάθε φορά που ανιχνεύει εμπόδιο σε απόσταση μικρότερη των 25εκ και όταν το εμπόδιο αυτό δεν υπάρχει να κινείται ευθεία. Σήμερα θα ολοκληρώσουμε το ρομπότ μας και θα το κάνουμε πιο έξυπνο. Θα το προγραμματίσουμε να κινείται ευθεία και όταν θα συναντήσει ένα εμπόδιο σε απόσταση μικρότερη των 25 εκ να μπορεί να αλλάξει πορεία και να το αποφύγει. Για το σκοπό αυτό θα χρησιμοποιήσουμε έναν σερβοκινητήρα ο οποίος θα περιστρέφει τον υπερηχητικό αισθητήρα δεξιά και αριστερά, ώστε να φαίνεται τελικά ότι τα μάτια του αυτοκινήτου μπορούν να ελέγχουν και αριστερά και δεξιά τι υπάρχει.



Στο παραπάνω σχέδιο, θα χρησιμοποιήσουμε ένα breadboard για να συνδέσουμε πιο εύκολα τις γειώσεις GND και τα 5V που χρειάζονται τα διάφορα εξαρτήματα. Το αυτοκίνητό μας θα φαίνεται περίπου ως εξής:



Μπορείτε πάνω στον σερβοκινητήρα να κολλήσετε ένα μικρό breadboard με ταινία διπλής όψεως ή με σιλικόνη ώστε να είναι σταθερή και να τοποθετήσετε εκεί τον ultrasonic sensor. Υπάρχουν και έτοιμες υποδοχές για τον ultrasonic sensor ώστε να συνδεθεί εύκολα με τον σερβοκινητήρα αντί για τον άξονά του.



Ας δούμε τι θα κάνει λοιπόν το αυτοκινητάκι μας, με την ακόλουθη σειρά:

Ο λαιμός (σερβοκινητήρας) θα κρατάει τα μάτια (υπερηχητικός αισθητήρας) σε ευθεία θέση (γωνία 90°). Αν ανιχνεύσει εμπόδιο (απόσταση μικρότερη από μια τιμή) τότε:

- Σταματάει (κινητήρες σε 0 ταχύτητα)
- Στρίβει το κεφάλι αριστερά (γωνία 0)



- Θέτει την τιμή του αισθητήρα απόστασης σε μια μεταβλητή «αριστερά- aristera» ώστε να μετρήσει την απόσταση από τα αριστερά



- Για 1 δευτερόλεπτο



- Στρίβει το κεφάλι δεξιά (γωνία 120ο)



- Θέτει την τιμή του αισθητήρα απόστασης σε μια μεταβλητή «δεξιά- deksia» ώστε να μετρήσει την απόσταση από τα αριστερά



- Για 1 δευτερόλεπτο



- Στρίβει το κεφάλι ευθεία για να αποφασίσει τι θα κάνει



Στις παραπάνω εντολές συμπληρώστε τις κατάλληλες τιμές στα κουτάκια που λείπουν.

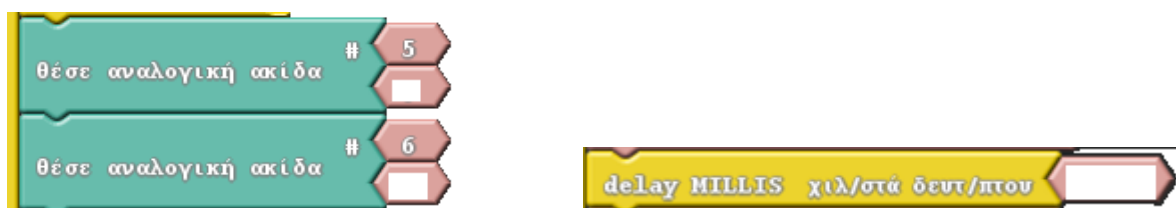
Μπορείτε να γράψετε με λόγια τι έκανε το ρομπότ μας;

Στη συνέχεια, θα συγκρίνει τις τιμές που πήρε από τον αισθητήρα δεξιά και αριστερά και ανάλογα θα στρίβει. Για να στρίψει έχουμε πει ότι θα πρέπει να λειτουργεί ένας από τους δύο κινητήρες. Ο 5 είναι ο δεξιός κινητήρας και ο 6 ο αριστερός. Μετά τη στροφή για μισό δευτερόλεπτο, θα σταματάει για 1 λεπτό και θα συνεχίζει το πρόγραμμα από την αρχή

Κοιτάξτε το παρακάτω πρόγραμμα και συμπληρώστε κατάλληλα τις εντολές του. Συμπληρώστε με ΡΕΥΜΑ/ ΟΧΙ ΡΕΥΜΑ ανάλογα αν θα λειτουργεί ή όχι η ψηφιακή ακίδα για τον κάθε κινητήρα. Μπορούμε να χρησιμοποιήσουμε και την ψηφιακή τιμή της ακίδας και την αναλογική όπως είδαμε στην προηγούμενη δραστηριότητα.



Τα παραπάνω γίνονται όταν ελέγχουμε αν αριστερά ή δεξιά υπάρχει μεγαλύτερη απόσταση από εμπόδιο ώστε να στρίψει εκ νέου. Στη συνέχεια θέλουμε να σταματήσει για 1 δευτερόλεπτο πριν αρχίσει να εκτελεί από την αρχή το πρόγραμμα. Συμπληρώστε με τιμές τις αναλογικές ακίδες 0-255 καθώς και τον αριθμό των χιλιοστών των δευτερολέπτων.

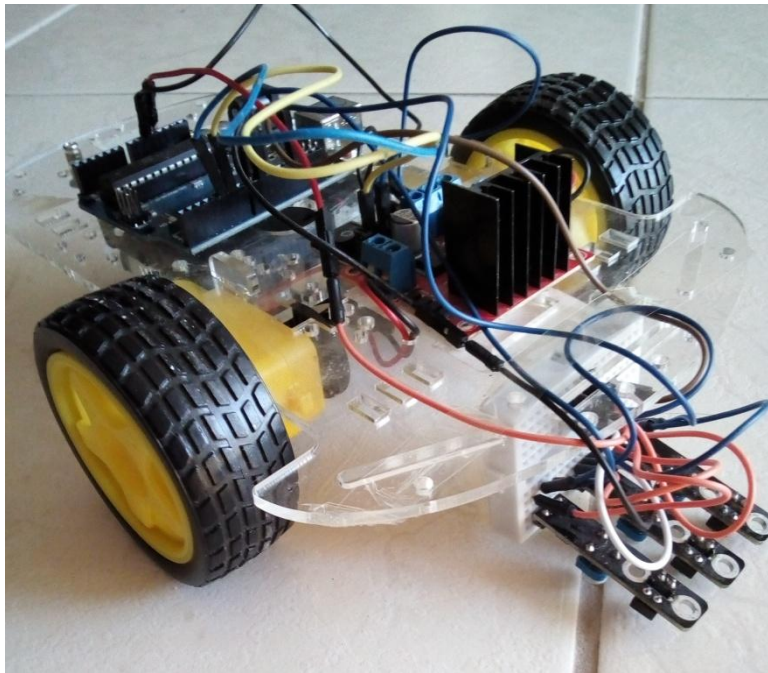
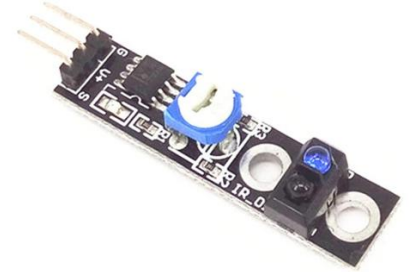


Ενώστε τις παραπάνω εντολές και δημιουργήστε το πρόγραμμα στο Ardublock.

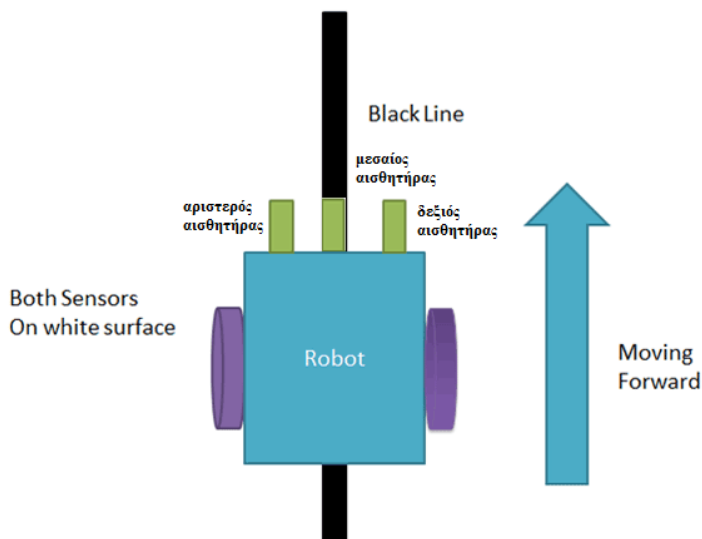
15^η Δραστηριότητα: Αυτόνομο όχημα που κινείται συνεχώς πάνω σε μια μαύρη γραμμή

Θα δημιουργήσουμε ένα όχημα το οποίο θα κινείται συνεχώς σε μια μαύρη γραμμή, θα την ακολουθεί. Θα χρειαστούμε:

- Ένα σασί
- 2 κινητήρες με ρόδες
- Μια γέφυρα για τον έλεγχο των κινητήρων
- 2 ή 3 αισθητήρες υπέρυθρων που ελέγχει αν βλέπει μαύρο ή όχι στο δάπεδο.
- Καλώδια
- Breadboard μεσαίο



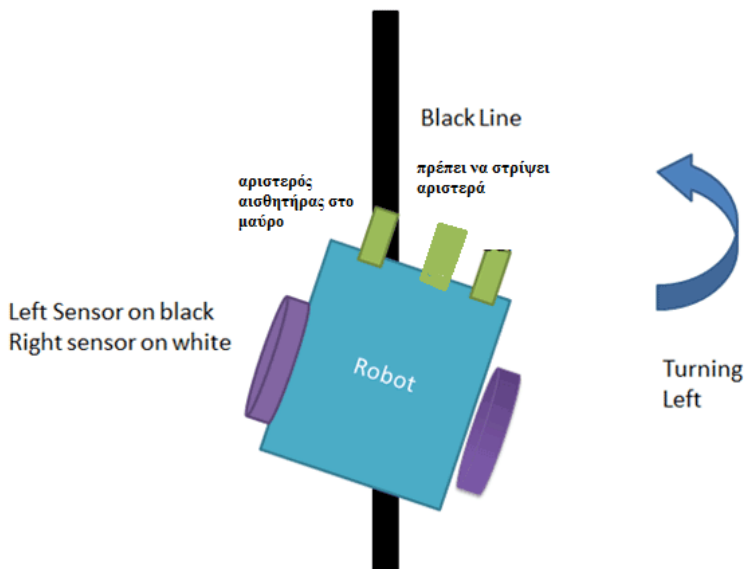
Οι αισθητήρες δίνουν δύο τιμές. Το μαύρο είναι 1 και το άσπρο 0 για τον συγκεκριμένο αισθητήρα.



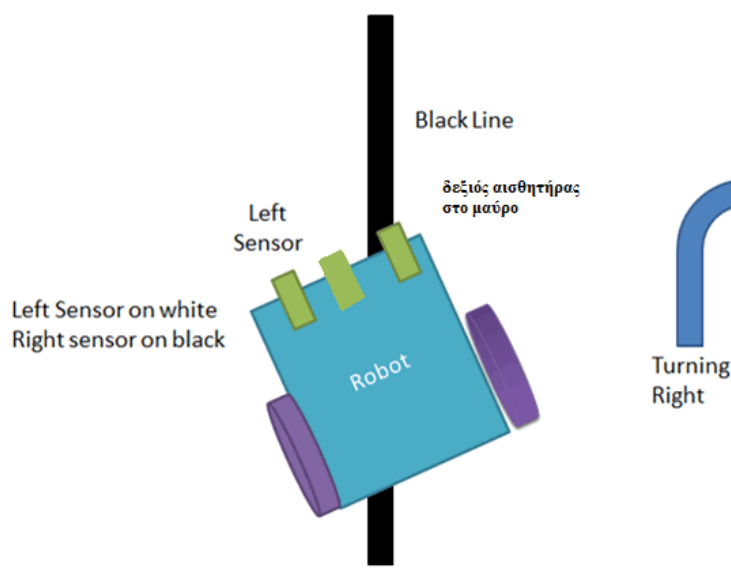
Αν ο μεσαίος επιστρέφει 1 και οι άλλοι 0 τότε οι κινητήρες κινούνται μπροστά, ίδια ταχύτητα και οι δύο. (μπροστά κίνηση)

Αν ο αριστερός είναι 1 τότε πρέπει να στρίψει προς τα αριστερά μέχρι να βρει ο μεσαίος μαύρο. Δηλαδή, ο αριστερός κινητήρας να είναι 0 ταχύτητα και ο δεξιός να λειτουργεί.

Αν ο δεξιός είναι 1 τότε πρέπει να στρίψει προς τα δεξιά μέχρι να βρει ο μεσαίος μαύρο. Δηλαδή ο δεξιός να είναι 0 ταχύτητα και ο αριστερός να λειτουργεί.

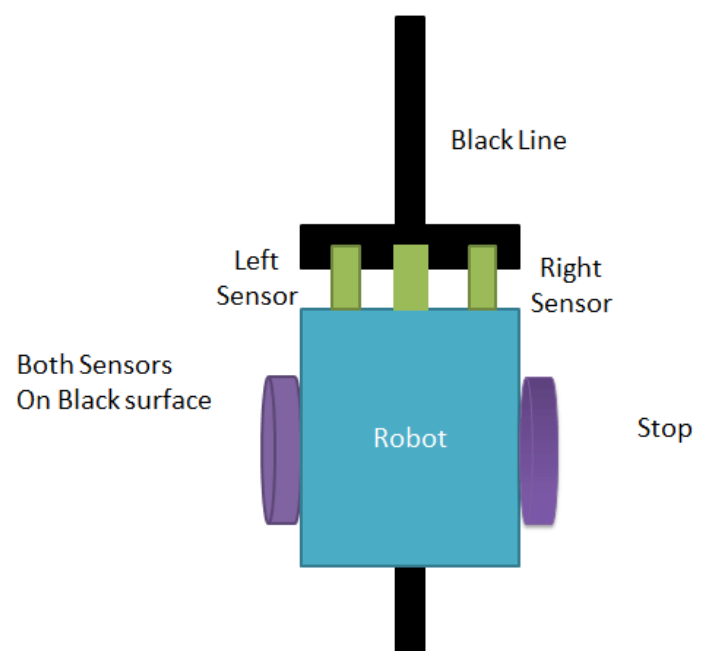


Ο αριστερός αισθητήρας είναι στη μαύρη γραμμή άρα πρέπει να στρίψει προς τα αριστερά, δηλαδή να δουλέψει ο δεξιός κινητήρας

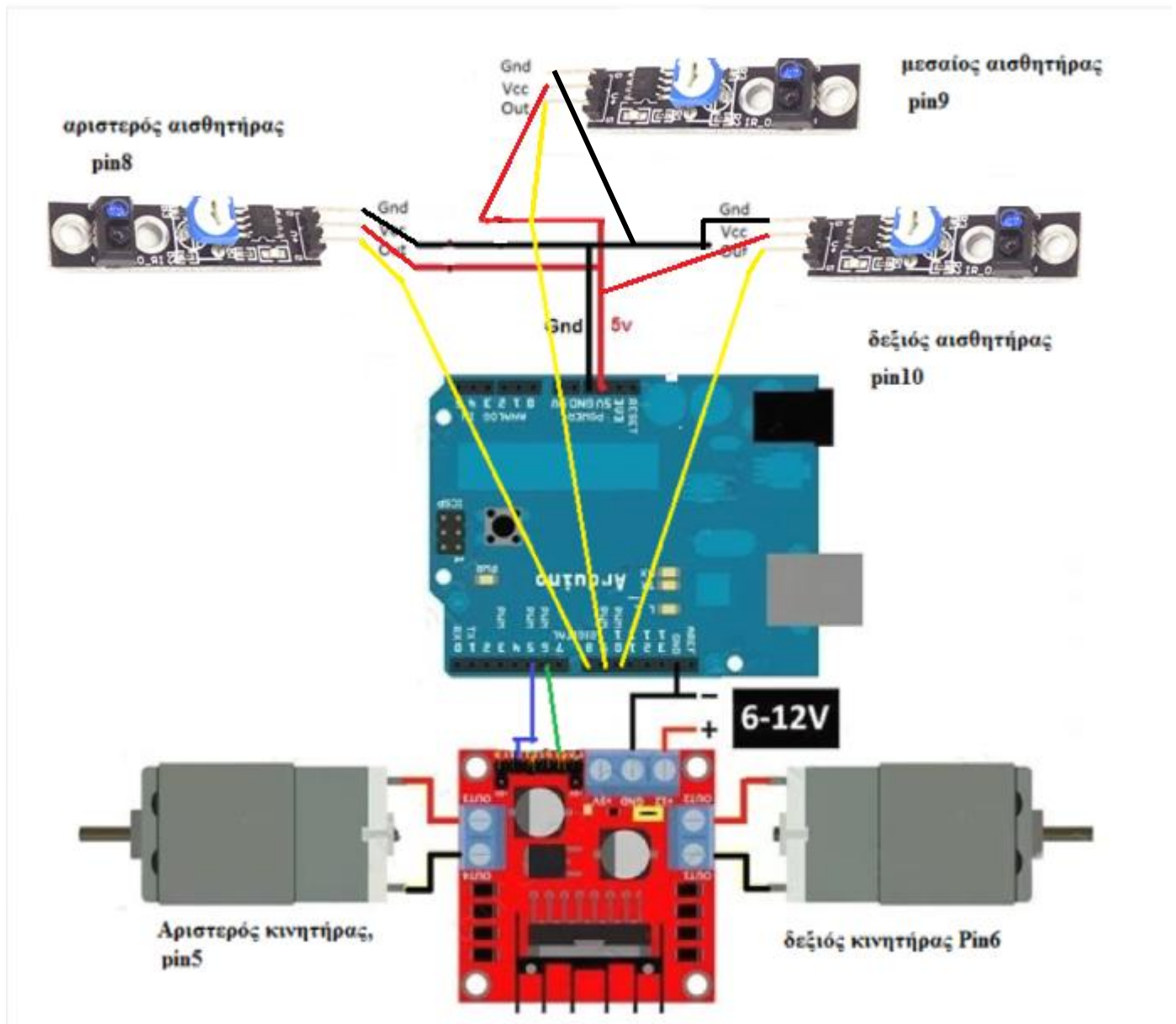


Ο δεξιός αισθητήρας είναι στη μαύρη γραμμή άρα πρέπει να στρίψει προς τα δεξιά, δηλαδή να δουλέψει ο αριστερός κινητήρας

Και οι τρεις μπροστινοί αισθητήρες βλέπουν τη μαύρη γραμμή, τότε σταματάει το αυτοκίνητο



Εδώ βλέπουμε μια εικόνα της συνδεσμολογίας. Χρησιμοποιούμε ένα breadboard ώστε να συνδέσουμε γειώσεις GND και τάση 5V καθώς και για να τοποθετήσουμε τους αισθητήρες IR, όπως στην εικόνα με το αυτοκινητάκι.



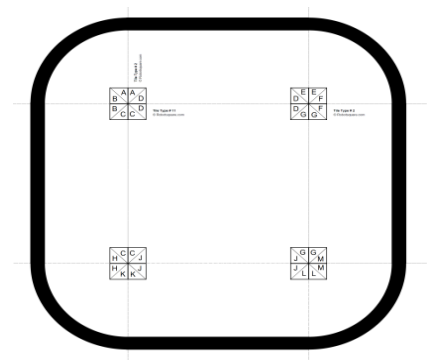
Τα pin στα οποία έχουν τοποθετηθεί τα εξαρτήματα είναι ενδεικτικά και μπορείτε να τα τοποθετήσετε όπου θέλετε. Αν θέλουμε να έχουμε μικρότερη ταχύτητα στους κινητήρες τότε θα πρέπει να τους συνδέσουμε με μια ακίδα που είναι και αναλογική, πχ 5 και 6.

Συμπληρώστε με τις κατάλληλες ταχύτητες ανάλογα με τις τιμές των αισθητήρων. Να θυμίσουμε ότι το μαύρο είναι η τιμή 1= ΑΛΗΘΕΣ.

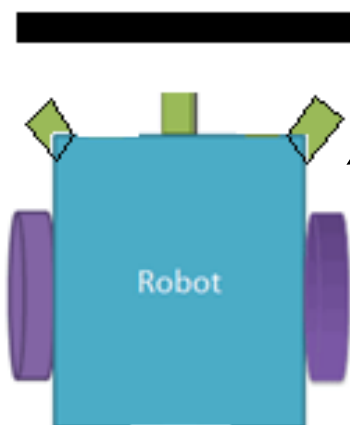
- Ο αριστερός αισθητήρας είναι 8
- Ο μεσαίος αισθητήρας 9
- Ο δεξιός αισθητήρας 10
- Ο αριστερός κινητήρας 5
- Ο δεξιός κινητήρας 6

16^η Δραστηριότητα: Όχημα που δεν πέφτει από το τραπέζι ή παραμένει μέσα σε μια κλειστή περιοχή

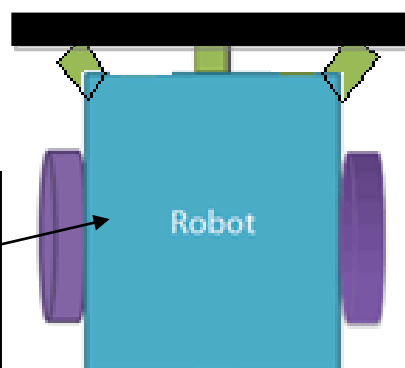
Παρόμοια δραστηριότητα μπορούμε να υλοποιήσουμε με τους αισθητήρες υπέρυθρων για να κατασκευάσουμε ένα όχημα και να το προγραμματίσουμε ώστε να μη πέφτει από το τραπέζι ή να μένει μέσα σε ένα πλαίσιο που έχουμε δημιουργήσει με μια μαύρη ταινία, όπως στη διπλανή εικόνα.



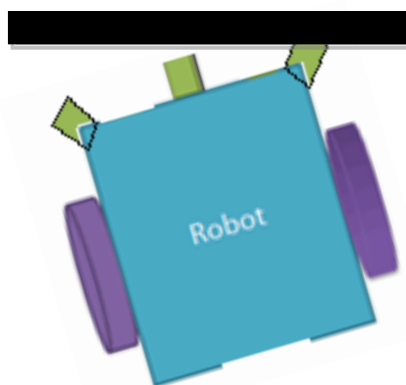
Μπορούμε να χρησιμοποιήσουμε έναν αισθητήρα στο μπροστινό μέρος ή έναν στο μπροστινό και δύο μπροστά και πλάγια ώστε να ελέγχουμε καλύτερα την απόσταση από τη γωνία του τραπεζιού και έτσι να είμαστε περισσότερο σίγουροι ότι δεν θα βρεθεί καμία ρόδα στον αέρα.



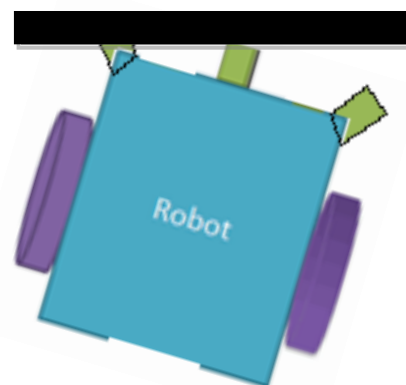
Κίνηση εμπρός, δεν
ανιχνεύει μαύρη
γραμμή ή κενό



Σταματάει και κινείται
προς τα πίσω αν
ανιχνεύσει μαύρη γραμμή
ή κενό



Οι πλάγιοι μπροστινοί
αισθητήρες ανιχνεύουν
μαύρη γραμμή ή κενό και
έτσι πρέπει να κάνουν
όπισθεν και στη συνέχεια να
στρίψουν αριστερά ή δεξιά



Με αυτή τη λογική κατασκευάστε το παραπάνω όχημα, εδώ δεν χρειάζεται να χρησιμοποιήσετε τον υπερηχητικό αισθητήρα (μάτια). Αποσυνδέστε στο προηγούμενο όχημα τα μάτια και προγραμματίστε το σύμφωνα με τα παρακάτω

Ας καταγράψουμε τη λογική του προγράμματος:

Αν ο μπροστινός αισθητήρας δει μαύρο **τότε**

Πήγαινε πίσω για κάποιο χρόνο και στρίψε αριστερά ή δεξιά για κάποιο χρόνο

Αλλιώς Αν ο δεξιός δει μαύρο **τότε**

κάνε πίσω για κάποιο χρόνο και στρίψε δεξιά για κάποιο χρόνο

Αλλιώς Αν ο αριστερός δει μαύρο **τότε**

κάνε πίσω για κάποιο χρόνο και στρίψε αριστερά για κάποιο χρόνο

Αλλιώς

Πήγαινε μπροστά

Εδώ θα πρέπει να κάνετε δοκιμές με τις ταχύτητες των κινητήρων και τους χρόνους που θα πηγαίνουν πίσω και θα στρίβουν αριστερά ή δεξιά. Καλό είναι να χρησιμοποιήσετε πάλι τις αναλογικές ακίδες και να χρησιμοποιήσετε μια χαμηλή ταχύτητα. Πάντα να είστε έτοιμοι να πιάσετε το όχημά σας όταν πλησιάζει στο τέλος του τραπεζιού. Για αρχή μπορείτε να το λειτουργήσετε μέσα σε ένα χώρο που είναι κλειστός από μια μαύρη γραμμή ώστε να διασφαλίσετε ότι δεν θα πέσει κάτω και στη συνέχεια να το τοποθετήσετε πάνω στο τραπέζι.

Επίλογος

Όπως είδαμε στις παραπάνω δραστηριότητες, οι εφαρμογές και οι δυνατότητες του μικροελεγκτή Arduino και των εξαρτημάτων του και του προγραμματιστικού περιβάλλοντος Ardublock είναι απεριόριστες. Υπάρχει πληθώρα και ποικιλία αισθητήρων και εξαρτημάτων που μπορούμε να χρησιμοποιήσουμε, να κατασκευάσουμε οτιδήποτε με απλά υλικά και να τα συνδέσουμε ώστε να δημιουργήσουμε λειτουργικές και διασκεδαστικές εφαρμογές. Λίγη φαντασία χρειάζεται και μπορούμε να δημιουργήσουμε τα πάντα!

Ας δώσουμε τη δυνατότητα στους μαθητές μας, στα παιδιά μας, να πάρουν μια γεύση από τον κόσμο του προγραμματισμού, του αυτοματισμού και της ρομποτικής και είμαι σίγουρη ότι θα μας καταπλήξουν με τις δυνατότητές τους και τη φαντασία τους.

Ελπίζω τα παραπάνω να ενθαρρύνουν τους συναδέλφους μου ώστε να εντάξουν τη ρομποτική και το arduino στην εκπαιδευτική τους διαδικασία.

Ναταλία Θεόκλεια Γεωργιτζίκη
Ηλεκτρολόγος Μηχανικός & Μηχανικός Υπολογιστών ΑΠΘ, ΜΕeng
Εκπαιδευτικός Πληροφορικής, ΠΕ86