



**Καινοτομία
στην τοπική
κοινωνία**

Οδηγός Arduino για το μάθημα της Πληροφορικής

Περιεχόμενα

Εισαγωγικό Σημείωμα προς τους Εκπαιδευτικούς.....	6
Λίστα Υλικών που περιέχει το κάθε Πακέτο	12
Εισαγωγή: Οδηγίες εγκατάστασης Arduino IDE και ArduBlock	13
1 Arduino IDE	13
1.1 Κατέβασμα του Arduino IDE.....	13
1.2 Εγκατάσταση του Arduino IDE	15
1.2.1 Για τους χρήστες Windows	15
1.2.2 Για τους χρήστες Linux	18
1.3 Φόρτωση προγράμματος με το Arduino IDE.....	18
2 ArduBlock	21
2.1 Ενσωμάτωση του ArduBlock στο IDE.....	21
A' Γυμνασίου	23
3 Εισαγωγή.....	24
3.1 Arduino UNO	24
3.2 Breadboard	25
3.3 Χρήση του ArduBlock	26
4 Εφαρμογή 1: Έλεγχος LED.....	29
4.1 Υλικά.....	29
LED	29
Αντίσταση	29
Breadboard	30
Arduino UNO	30
Καλώδια	30
4.2 Κατασκευή κυκλώματος	30
4.3 Σύνδεση πλακέτας με τον υπολογιστή	33
4.4 Ανάπτυξη προγράμματος	34
5 Εφαρμογή 2: Φανάρι Κυκλοφορίας	39
5.1 Υλικά.....	39
5.2 Κύκλωμα	39
5.3 Σύνδεση πλακέτας με τον υπολογιστή	40
5.4 Πρόγραμμα.....	40

6 Εφαρμογή 3: LED με κουμπί	42
6.1 Υλικά.....	42
Κουμπί.....	42
Αντίσταση 10kΩ.....	42
6.2 Κύκλωμα	43
6.3 Σύνδεση πλακέτας με τον υπολογιστή	46
6.4 Πρόγραμμα.....	46
7 Εφαρμογή 4: LED που αναβοσβήνει και LED με κουμπί.....	48
7.1 Υλικά.....	48
7.2 Κύκλωμα	48
7.3 Σύνδεση πλακέτας με τον υπολογιστή	49
7.4 Πρόγραμμα.....	49
B' Γυμνασίου	53
8 Εφαρμογή 5: LED με μεταβαλλόμενη ένταση.....	54
8.1 Υλικά.....	54
LED	54
Αντίσταση 220Ω.....	54
Breadboard	54
Arduino UNO	55
Καλώδια	56
8.2 Κύκλωμα	56
8.3 Σύνδεση πλακέτας με τον υπολογιστή	58
8.4 Πρόγραμμα.....	59
9 Εφαρμογή 6: LED ρυθμιζόμενο με ποτενσιόμετρο	61
9.1 Υλικά.....	61
Περιστροφικό ποτενσιόμετρο	61
9.2 Κύκλωμα	62
9.3 Σύνδεση πλακέτας με τον υπολογιστή	65
9.4 Πρόγραμμα.....	65
10 Εφαρμογή 7: Αισθητήρας φωτός	67
10.1 Υλικά.....	67
Φωτοαντίσταση.....	67
Αντίσταση 10 kΩ.....	67
10.2 Κύκλωμα	68

10.3 Σύνδεση πλακέτας με τον υπολογιστή	72
10.4 Πρόγραμμα.....	72
11 Εφαρμογή 8: Αισθητήρας υπερήχων HC-SR04.....	74
11.1 Υλικά.....	74
Αισθητήρας HC-SR04.....	74
Active Buzzer	75
Αντίσταση 100 Ω.....	76
11.2 Κύκλωμα	76
11.3 Σύνδεση πλακέτας με τον υπολογιστή	83
11.4 Πρόγραμμα.....	83
Γ' Γυμνασίου	85
12 Σχεδιασμός κατασκευής ρομποτικού οχήματος.....	86
12.1 Εξαρτήματα	86
Σκελετός.....	86
Μοτέρ dc	86
Κύκλωμα οδήγησης μοτέρ	87
L293D chip.....	87
L293D motor shield.....	89
Αισθητήρας HC-SR04.....	90
Arduino UNO	91
Breadboard	91
12.2 Τρόπος κίνησης του οχήματος	91
12.3 Αρχική περιγραφή λειτουργίας προγράμματος	92
13. Συναρμολόγηση οχήματος	92
13.1 Προετοιμασία εξαρτημάτων.....	92
13.2 Τοποθέτηση μοτέρ.....	94
13.3 Τοποθέτηση μπαταριοθήκης	95
13.4 Τοποθέτηση πίσω ρόδας.....	96
13.5 Τοποθέτηση και σύνδεση διακόπτη	97
13.6 Τοποθέτηση Arduino.....	98
13.7 Τοποθέτηση τροχών	99
13.8 Τοποθέτηση breadboard και αισθητήρα HC-SR04	100
13.9 Τοποθέτηση κυκλώματος οδήγησης	101
14 Σύνδεση εξαρτημάτων.....	103

14.1 Συνδεσμολογία με χρήση του motor shield.....	103
14.1.1 Σύνδεση πηγής τροφοδοσίας.....	103
14.1.2 Σύνδεση μοτέρ.....	104
14.1.3 Σύνδεση αισθητήρα απόστασης	105
14.2 Συνδεσμολογία με χρήση του chip L293D.....	106
14.2.1 Συνδέσεις τροφοδοσίας	107
14.2.2 Συνδέσεις μοτέρ	110
14.2.3 Συνδέσεις σημάτων ελέγχου.....	111
14.2.4 Σύνδεση αισθητήρα απόστασης	115
15 Προγραμματισμός για την κίνηση του οχήματος.....	118
15.1 Σύνδεση Arduino με τον υπολογιστή.....	118
15.2 Πρόγραμμα για υλοποίηση με motor shield.....	119
15.3 Πρόγραμμα για υλοποίηση με L293D chip.....	123
16 Προγραμματισμός για την αποφυγή εμποδίων	128
16.1 Σύνδεση Arduino με τον υπολογιστή.....	128
16.2 Διαδικασία που θα ακολουθηθεί	128
16.3 Πρόγραμμα.....	129
Πηγές.....	131

Εισαγωγικό Σημείωμα προς τους Εκπαιδευτικούς

Το σχολείο αποτελεί ένα ανοιχτό κοινωνικό σύστημα, με το μέλλον του να είναι στενά συνδεδεμένο με το μέλλον της ευρύτερης κοινωνίας, μέσα στην οποία λειτουργεί και πραγματοποιεί τους στόχους του. Μέσα σε συνθήκες συνεχούς αλλαγής, η Ελληνική πολιτεία οραματίζεται ένα σύγχρονο, δημοκρατικό και ανθρώπινο σχολείο, το οποίο θα λειτουργεί ως μανθάνουσα κοινότητα (Schools as Focused Learning Organizations, OECD 2004) με στόχο τη διαρκή προσπάθεια για βελτίωση της ποιότητας του επιτελούμενου εκπαιδευτικού έργου. Στο πλαίσιο αυτό το πρόγραμμα STEMpowering Youth φιλοδοξεί να ανοίξει το σχολείο στην τοπική κοινωνία και να το μετατρέψει σε εκκολαπτήριο καινοτόμων ιδεών που βασίζονται στις νέες τεχνολογίες αλλά και στη δημιουργική σκέψη των μικρών μαθητών.

Το εκπαιδευτικό πρόγραμμα «STEMpowering Youth» σχεδιάστηκε και υλοποιείται από το μη κερδοσκοπικό εκπαιδευτικό οργανισμό [«Επιστήμη Επικοινωνία - SciCo»](#), με την [Ελληνογερμανική Αγωγή](#) ως Ακαδημαϊκό Σύμβουλο και υποστήριξη από το [Ίδρυμα Vodafone](#). Από τον Φεβρουάριο του 2017 το «STEMpowering Youth» λειτουργεί ως εβδομαδιαίο εξωσχολικό πρόγραμμα με θεματικές που προάγουν την φιλοσοφία της STEM εκπαίδευσης και βιωματικές δραστηριότητες βασισμένες στο ομαδοσυνεργατικό πνεύμα. Το εξωσχολικό STEM πρόγραμμα διάρκειας 10 εβδομάδων, λαμβάνει χώρα σε παραμεθόριες περιοχές της Ελληνικής Επαρχίας και μέχρι στιγμής έχει υλοποιηθεί στη Θράκη και στα Δωδεκάνησα. Τον Φεβρουάριο του 2018 ο οργανισμός SciCo στα πλαίσια του προγράμματος «STEMpowering Youth» παρέχει πιλοτικά σε 12 γυμνάσια της περιφέρειας, Πακέτα Υλικών για να διευκολύνει το έργο του εκπαιδευτικού ενισχύοντας το μάθημα της Φυσικής με εργαστηριακό εξοπλισμό και το μάθημα της Πληροφορικής με εξοπλισμό Arduino. Η παροχή πακέτων θα συνεχιστεί και σε άλλα σχολεία της περιφέρειας για το Ακαδημαϊκό έτος 2018-19.

Το πρόγραμμα «STEMpowering Youth» θα υλοποιείται στα πλαίσια του ευρωπαϊκού έργου H2020 [«Open Schools for Open Societies - OSOS»](#) (Ένα Ανοιχτό σχολείο σε μια Ανοιχτή κοινωνία) από τον Σεπτέμβριο του 2018. Εθνικός συντονιστής του έργου OSOS για την Ελλάδα είναι το [Ινστιτούτο Εκπαιδευτικής Πολιτικής \(ΙΕΠ\)](#). Αντικείμενο του έργου H2020 είναι η προετοιμασία της εισαγωγής της καινοτομίας του «Ανοιχτού σχολείου», με βασικό άξονα τις φυσικές επιστήμες και τα αντικείμενα του STEM

(Science Technology Engineering Mathematics) σε θεματικές που συνδέονται με σύγχρονες κοινωνικές προκλήσεις, σε όλες τις βαθμίδες της Εκπαίδευσης. Το «Ανοικτό Σχολείο» έχει ως στόχο να αξιοποιήσει την εκπαιδευτική καινοτομία, τα εργαλεία της ανακαλυπτικής μάθησης, την τεχνολογία, τη διδασκαλία των φυσικών, ανθρωπιστικών και κοινωνικών επιστημών για τη μελέτη και ψηλάφηση τοπικών ζητημάτων. Οι Διευθυντές και οι Εκπαιδευτικοί των σχολείων που θα συμμετέχουν, θα θέσουν και το δικό τους όραμα για το ανοικτό σχολείο, το οποίο θα κληθούν να υποστηρίξουν με τη δημιουργία πρωτότυπων μαθησιακών εμπειριών υποστηριζόμενων με τα κατάλληλα εκπαιδευτικά εργαλεία που θα διευκολύνουν τους μαθητές να συμμετέχουν με ενεργό ρόλο στη μαθησιακή διαδικασία.

Πακέτα Υλικών Πληροφορικής

Ο οργανισμός SciCo σε μια προσπάθεια να ενισχύσει την βιωματική εκπαίδευση και την καλλιέργεια της ερευνητικής και επινοητικής ευφυΐας, παρέχει πιλοτικά σε Γυμνάσια παραμεθόριων περιοχών της χώρας, ειδικά σχεδιασμένα Πακέτα Υλικών με εξοπλισμό Arduino για να ενισχύσει το μάθημα της Πληροφορικής και να διευκολύνει το έργο του εκπαιδευτικού, κάνοντας το περισσότερο βιωματικό.

Οι γνωσιακές επιστήμες μας δείχνουν ότι όταν η γνώση κατακτάται μέσω της εμπειρίας, ενσώματα και βιωματικά, τότε η πληροφορία που λαμβάνουμε αποτυπώνεται μακροπρόθεσμα στη μνήμη και γίνεται ευκολότερα κατανοητή. Αυτό μας οδηγεί στο συμπέρασμα πως για να είναι η μάθηση αποτελεσματική πρέπει το θεωρητικό κομμάτι της ύλης να συνοδεύεται από πρακτικές εφαρμογές. Με τη διεξαγωγή πρακτικών δραστηριοτήτων δίνεται επίσης η δυνατότητα στον μαθητή να συμμετέχει σε ομαδικές εργασίες αναπτύσσοντας πολύπλευρες δεξιότητες όπως η επικοινωνία, η κριτική και η λογική σκέψη, η έρευνα, η ανταλλαγή ιδεών, ο πειραματισμός και ο σχεδιασμός επίλυσης του προβλήματος.

Το Υπουργείο Παιδείας, Έρευνας και Θρησκευμάτων σε μια προσπάθεια να εντάξει τον πληροφορικό γραμματισμό (ICT Literacy) στο Αναλυτικό Πρόγραμμα και να ενισχύσει την χρήση σύγχρονων ψηφιακών τεχνολογιών και εργαλείων επικοινωνίας στην τάξη, προτείνει στους εκπαιδευτικούς να διερευνήσουν τις προϋπάρχουσες γνώσεις των μαθητών τους και να σχεδιάσουν δραστηριότητες πιο κοντά στην καθημερινή τους ζωή.

Με αυτό τον τρόπο επιδιώκει το Εργαστήριο Πληροφορικής να αποτελεί για τους μαθητές χώρο μελέτης, έρευνας, ενεργητικής συμμετοχής και συνεργασίας, ώστε να ενθαρρύνεται και να ευνοείται η διερευνητική προσέγγιση της γνώσης, η συνεργατική μάθηση, η αυτενέργεια και η δημιουργικότητα.

Ειδικότερα:

Σύμφωνα με το Αναλυτικό Πρόγραμμα Σπουδών και τις οδηγίες που παρέχονται από το Ινστιτούτο Εκπαιδευτικής Πολιτικής, ο εκπαιδευτικός έχει την δυνατότητα να σχεδιάσει τις δικές του δραστηριότητες, να αξιοποιήσει αξιόλογο και κατάλληλο εκπαιδευτικό υλικό που αυτός θα βρει αλλά και να δημιουργήσει νέο δικό του. Η διδασκαλία της Πληροφορικής στο Γυμνάσιο έχει σαφή εργαστηριακό προσανατολισμό με βασικό παράγοντα την ενεργό συμμετοχή κάθε μαθητή, την συνεχή αλληλεπίδραση και συνεργασία με τον διδάσκοντα και τους συμμαθητές του. Βασική τεχνική διδασκαλίας, κυρίως στο εργαστηριακό μέρος του μαθήματος καθίστανται τα σχέδια εργασίας/έρευνας (projects). Οι εκπαιδευτικοί ενθαρρύνονται να χρησιμοποιούν Ελεύθερο και Ανοιχτό Λογισμικό και Υλικό και έχουν τη δυνατότητα να επιλέξουν οποιοδήποτε προγραμματιστικό περιβάλλον οπτικού προγραμματισμού με πλακίδια για την ενότητα «Προγραμματισμό Υπολογιστικών Συσκευών και Ρομποτικών Συστημάτων» όπως προτείνεται στις Αναλυτικές Οδηγίες του ΙΕΠ.

Λαμβάνοντας υπόψιν τα παραπάνω, παρέχουμε σε κάθε σχολείο που θα συμμετέχει στο πρόγραμμα **εξοπλισμό Arduino** (δεκατέσσερα σετ Arduino με βασικούς αισθητήρες, υλικά για πέντε ρομποτικά οχήματα) για να χρησιμοποιηθεί σε ομάδες των δύο ή περισσότερων μαθητών. Ο παρών οδηγός αποτελείται από διδακτικές προτάσεις που μπορούν να αξιοποιηθούν από τον εκπαιδευτικό για την διδασκαλία της ενότητας «Προγραμματισμός Υπολογιστικών Συσκευών και Ρομποτικών Συστημάτων» η οποία περιλαμβάνεται στη διδακτέα ύλη της Α, Β, Γ Γυμνασίου με προτεινόμενο διδακτικό χρόνο 5 ώρες ανά σχολικό έτος.

Οι δραστηριότητες που προτείνουμε είναι διαβαθμισμένης δυσκολίας και λειτουργούν ως επίπεδα στην Α, Β Γυμνασίου για να εισάγουν τον μαθητή στον προγραμματισμό του τελικού σχεδίου εργασίας (project) που είναι απαιτούμενο στο Αναλυτικό Πρόγραμμα της Γ Γυμνασίου. Το προγραμματιστικό περιβάλλον που προτείνουμε με οδηγίες στον

παρόν οδηγό είναι το Ardublock, μια γλώσσα οπτικού προγραμματισμού με πλακίδια. Εάν ο εκπαιδευτικός θέλει να χρησιμοποιήσει ένα προγραμματιστικό περιβάλλον παρόμοιο με την Scratch, έχει την δυνατότητα να χρησιμοποιήσει το S4A (Scratch for Arduino). Με τον εξοπλισμό που παρέχουμε στο Πακέτο Υλικών, ο εκπαιδευτικός θα μπορεί να δημιουργήσει ομάδες των δύο με τριών ατόμων στην Α, Β γυμνασίου και ομάδες των τεσσάρων με πέντε ατόμων στη Γ γυμνασίου.

Επιλέξαμε την πλακέτα Arduino για το πρόγραμμα μας διότι μέσω αυτής παρέχουμε τη δυνατότητα στους μαθητές να δουν την εφαρμογή του προγραμματισμού σε ρεαλιστικές καταστάσεις. Με την χρήση του, οι μαθητές κατανοούν ότι μέσω του προγραμματισμού έχουν τον κύριο έλεγχο της λειτουργίας ρομποτικών συστημάτων ή κατασκευών. Μετά από κάθε κώδικα που σχεδιάζουν, οι μαθητές βλέπουν άμεσα αν εφαρμόζεται στην πράξη, εντοπίζοντας εύκολα τα λάθη που έκαναν για να τα διορθώσουν.

Το Arduino ενδείκνυται για εκπαιδευτικές εφαρμογές για τους παρακάτω λόγους:

- Έχει Χαμηλό Κόστος.
- Είναι απλό. Μέσα σε λίγες ώρες ο άπειρος χρήστης μπορεί να δημιουργήσει την πρώτη του κατασκευή.
- Είναι Ανοικτού κώδικα (υλικό και λογισμικό).
- Ιδιοκατασκευή εναλλακτικών, καινοτόμων πειραματικών διατάξεων σε σχέση με αυτές του εμπορίου.
- Έχει πλούσιο διαδικτυακό υλικό υποστήριξης.
- Ειδικά forum όπου μοιράζονται την εμπειρία και την εργασία τους με άλλους εκπαιδευτικούς, ερευνητές και χομπίστες.
- Αισθητήρες με χαμηλό κόστος.
- Εύκολοι τρόποι απεικόνισης των δεδομένων από αισθητήρες.
- Πολύ οικονομική λύση για Εκπαιδευτική Ρομποτική.
- Πολλές εκδόσεις (κάθε έκδοση καλύπτει διαφορετικές ανάγκες).
- Πλακέτες επέκτασης (shield) που δίνουν νέες δυνατότητες στις πλατφόρμες του Arduino.
- Το Arduino IDE εκτελείται σε linux, windows, mac.

Λίγα λόγια για τους φορείς του προγράμματος

Ο Μη Κερδοσκοπικός εκπαιδευτικός Οργανισμός [«Επιστήμη Επικοινωνία - SciCo»](#) (από το Science Communication) ιδρύθηκε με σκοπό την καταπολέμηση του επιστημονικού αναλφαριθμητισμού στην Ελλάδα και στόχος της είναι να ενισχύσει την εικόνα του ερευνητή/επιστήμονα στην ελληνική κοινωνία και να ενημερώσει το κοινό για θέματα που αφορούν τις επιστήμες στην καθημερινότητα. Η SciCo αποτελείται από μια ομάδα επιστημόνων, ακαδημαϊκών, καλλιτεχνών, εκπαιδευτικών και τα τελευταία χρόνια πραγματοποιεί μια ποικιλία δράσεων για την επίτευξη των στόχων της (Εκπαιδευτικές Ημερίδες και Εκδηλώσεις, Φεστιβάλ Επιστήμης, Εκπαιδευτικά Εργαστήρια μεγάλης ή μικρής διάρκειας σε Αθήνα και επαρχία).

Με έδρα τη Μεγάλη Βρετανία και τη φράση «Connecting for Good» να συνοψίζει τη φιλοσοφία του, το [Ίδρυμα Vodafone](#) υλοποιεί προγράμματα σε 27 διαφορετικές χώρες εδώ και 25 χρόνια. Στην Ελλάδα, το Ίδρυμα Vodafone δραστηριοποιείται από το 2002, συνεργαζόμενο με περισσότερους από 70 οργανισμούς και φορείς της χώρας. Έχει υποστηρίξει δράσεις και προγράμματα για την κάλυψη σημαντικών, πάγιων και διαρκών αναγκών στον χώρο της υγείας και της εκπαίδευσης με στόχο την επιμόρφωση, την κοινωνική επανένταξη και τη βελτίωση των συνθηκών ζωής ευάλωτων κοινωνικά ομάδων.

Η [Ελληνογερμανική Αγωγή](#) έχει συμμετάσχει ως συντονιστής ή εταίρος σε δεκάδες επιδοτούμενα από την Ευρωπαϊκή Ένωση προγράμματα συνεργαζόμενη με σχολεία, πανεπιστημιακά ιδρύματα και ερευνητικά κέντρα της Ελλάδας και του εξωτερικού. Τα αποτελέσματα της έρευνας αξιολογούνται, παρουσιάζονται σε συνέδρια και δημοσιεύονται σε διεθνή επιστημονικά περιοδικά. Τα εκπαιδευτικά εργαλεία εφαρμόζονται στη σχολική πράξη, συμπληρώνοντας και εμπλουτίζοντας την παραδοσιακή διδασκαλία. Η Ελληνογερμανική Αγωγή προσφέρει το κατάλληλο περιβάλλον στους μαθητές της, έτσι ώστε οι απόφοιτοι του σχολείου να έχουν τα απαραίτητα εφόδια για τη ζωή τους, να εργάζονται ομαδικά, να ανακαλύπτουν και να δημιουργούν μέσα από δραστηριότητες που οδηγούν παράλληλα με την απόκτηση γνώσεων και στην ανάπτυξη κριτικής σκέψης

Λίστα Υλικών που περιέχει το κάθε Πακέτο

10 τεμάχια πλακέτες Arduino Uno

14 τεμάχια breadboard 400 tie points

100 καλώδια male to male

60 resistors 220 Ohm

60 resistors 10K Ohm (ή 100 Ohm)

60 τεμάχια λαμπάκια άσπρα

40 τεμάχια λαμπάκια κόκκινα

40 τεμάχια λαμπάκια πράσινα

12 τεμάχια button 4pin

12 ποτενσιόμετρα

12 buzzer

12 φωτοαντιστάσεις

12 αισθητήρας υπερήχων

4 βάσεις ρομποτικού οχήματος (μοτέρ, ρόδες, μπαταριοθήκη)

32 μπαταρίες AAA

4 motor driver shield

4 push pull motor driver

12 female header (3 σε κάθε ρομποτικό όχημα)

Εισαγωγή: Οδηγίες εγκατάστασης Arduino IDE και ArduBlock

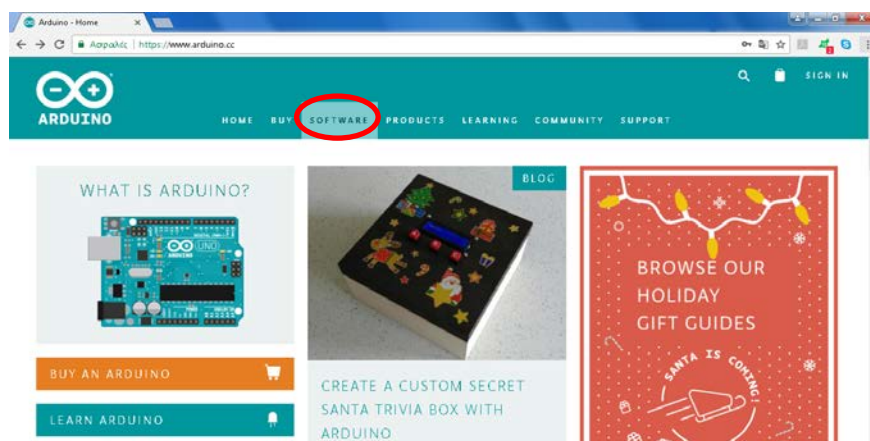
Στην ενότητα αυτή περιγράφεται η διαδικασία κατεβάσματος και εγκατάστασης του Arduino IDE, καθώς και η διαδικασία ενσωμάτωσης του εργαλείου ArduBlock στο περιβάλλον του IDE. Επίσης περιλαμβάνονται οδηγίες για τη φόρτωση προγραμμάτων στην πλακέτα Arduino.

1 Arduino IDE

Το Arduino IDE είναι το λογισμικό της αναπτυξιακής πλατφόρμας Arduino. Πρόκειται για ένα ολοκληρωμένο προγραμματιστικό περιβάλλον, μέσα στο οποίο μπορούμε να συντάξουμε τα προγράμματά μας σε γλώσσα Wiring C και στη συνέχεια να τα φορτώσουμε στην εκάστοτε πλακέτα.

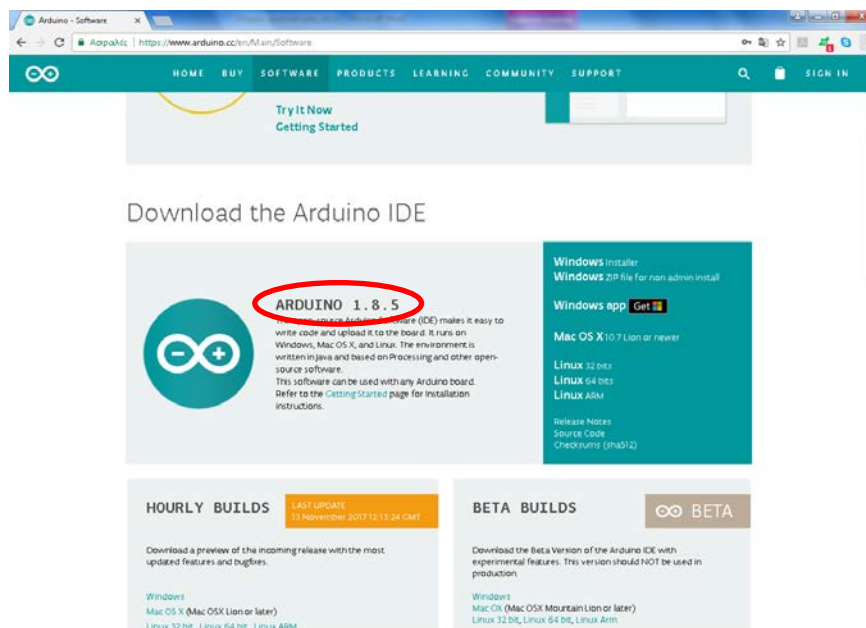
1.1 Κατέβασμα του Arduino IDE

Για να κατεβάσουμε το πρόγραμμα επισκεπτόμαστε την επίσημη ιστοσελίδα του Arduino (<https://www.arduino.cc>) και επιλέγουμε την καρτέλα *Software* (Εικόνα 1).



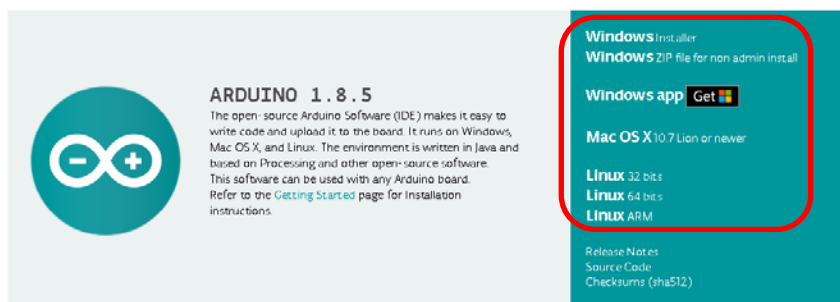
Εικόνα 1

Στη σελίδα του λογισμικού κατεβαίνουμε προς τα κάτω και βρίσκουμε την τελευταία έκδοση του IDE (Εικόνα 2).



Εικόνα 2

Download the Arduino IDE

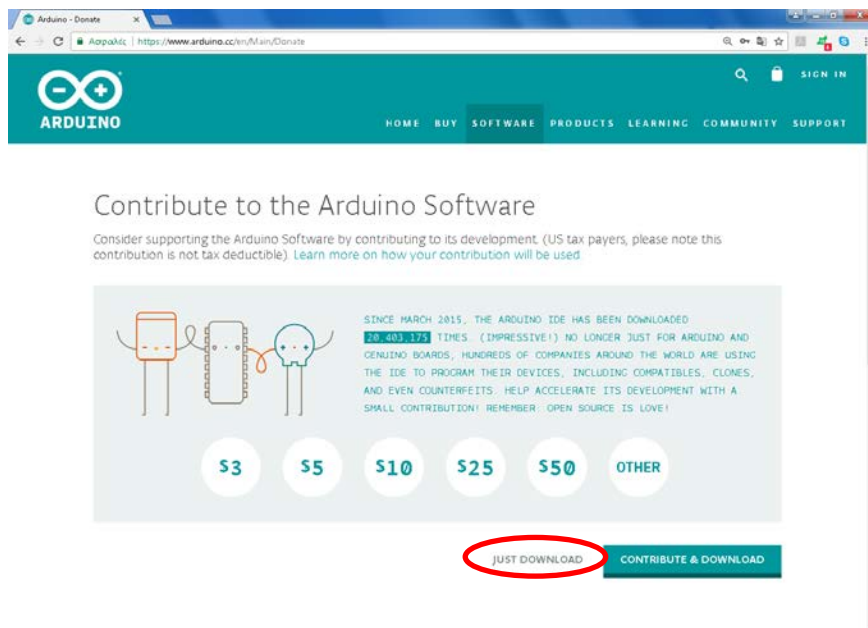


Εικόνα 3

Όπως φαίνεται στην Εικόνα 3 υπάρχουν εκδόσεις του προγράμματος για όλα τα λειτουργικά συστήματα. Για τους χρήστες των Windows υπάρχουν 3 σύνδεσμοι. Ο πρώτος κατεβάζει στον υπολογιστή μας ένα εκτελέσιμο αρχείο για αυτοματοποιημένη εγκατάστασή (προτεινόμενος τρόπος), ενώ ο δεύτερος ένα συμπιεσμένο αρχείο που περιέχει τον κατάλογο του προγράμματος. Ο τρίτος σύνδεσμος απευθύνεται σε χρήστες με Windows 10 και επιτρέπει το κατέβασμα του Arduino IDE ως app από το Windows Store. Ωστόσο, στο Διαδίκτυο υπάρχουν διάφορες αναφορές για προβλήματα και

περιορισμούς στη λειτουργία του app, οπότε προτείνεται και για τα Windows 10 η επιλογή ενός από τους δύο προηγούμενους τρόπους εγκατάστασης.

Αφού επιλέξουμε την έκδοση που επιθυμούμε, οδηγούμαστε στην επόμενη σελίδα, όπου πατάμε *JUST DOWNLOAD* (Εικόνα 4) και ξεκινάει το κατέβασμα του αρχείου.



Εικόνα 4

1.2 Εγκατάσταση του Arduino IDE

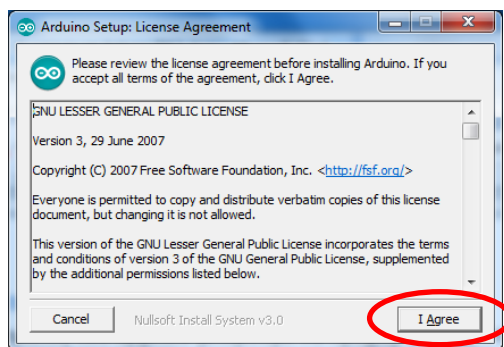
Με την ολοκλήρωση του κατεβάσματος του αρχείου, μπορούμε να προχωρήσουμε στην εγκατάσταση και την εκτέλεσή του IDE.

1.2.1 Για τους χρήστες Windows

Περίπτωση 1^η:

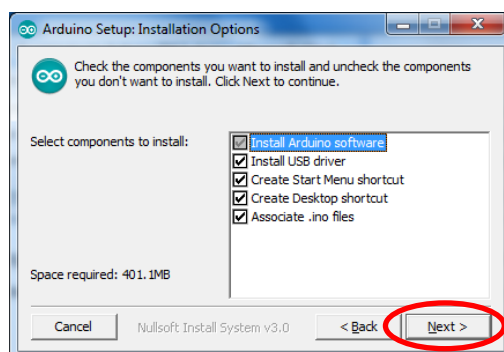
Αν έχουμε επιλέξει να κατεβάσουμε το εκτελέσιμο αρχείο, το βρίσκουμε στο φάκελο των λήψεων και το εκτελούμε. Η ενέργεια αυτή μπορεί να απαιτεί δικαιώματα Administrator. Στη συνέχεια ακολουθούμε τα βήματα που μας εμφανίζει ο οδηγός εγκατάστασης.

Αρχικά ανοίγει ένα παράθυρο με την Άδεια Χρήσης, όπου επιλέγουμε *I Agree* (Εικόνα 5).



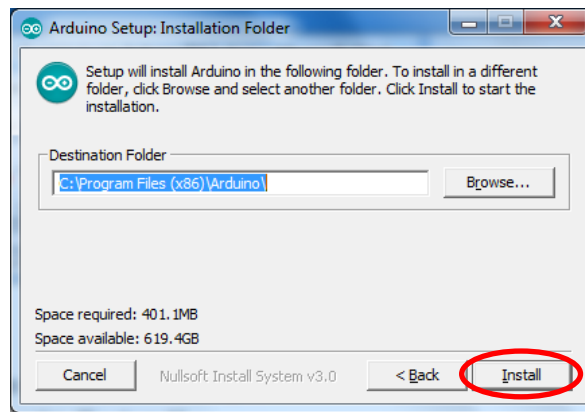
Εικόνα 5

Στη συνέχεια εμφανίζεται η λίστα επιλογής των στοιχείων που θα εγκατασταθούν και πατάμε *Next* χωρίς να αλλάξουμε κάτι (Εικόνα 6).



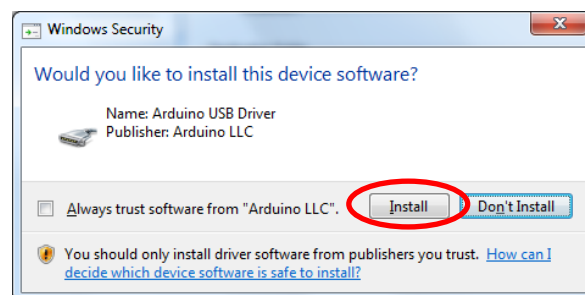
Εικόνα 6

Στο επόμενο παράθυρο μπορούμε να επιλέξουμε το φάκελο εγκατάστασης του προγράμματος. Αν επιθυμούμε αλλάζουμε τον προκαθορισμένο προορισμό και σε κάθε περίπτωση επιλέγουμε *Install* (Εικόνα 7).



Εικόνα 7

Μετά τη βασική εγκατάσταση της εφαρμογής, εγκαθίστανται μία σειρά από drivers. Ανάλογα με τις ρυθμίσεις ασφαλείας του συστήματός μας, μπορεί να μας εμφανιστούν ένα ή περισσότερα παράθυρα που θα ζητούν σχετική άδεια, οπότε και επιλέγουμε *Install* (Εικόνα 8).



Εικόνα 8

Με την ολοκλήρωση της εγκατάστασης και εφόσον έχουμε κάνει τη σχετική επιλογή (Εικόνα 6), εμφανίζεται στην επιφάνεια εργασίας μία συντόμευση για την εκτέλεση του προγράμματος.

Περίπτωση 2^η:

Αν έχουμε επιλέξει να κατεβάσουμε το συμπίεμένο αρχείο, το μόνο που χρειάζεται είναι να το αποσυμπιέσουμε σε μία θέση μέσα στο δίσκο μας (π.χ. στο C:\Program Files\). Η διαδικασία αυτή μπορεί να εκτελεστεί χωρίς δικαιώματα διαχειριστή (ανάλογα βέβαια και με τη θέση όπου θα επιλέξουμε να γίνει η αποσυμπίεση) και μας επιτρέπει να

έχουμε ταυτόχρονα πολλές διαφορετικές εκδόσεις της εφαρμογής μέσα στο σύστημά μας. Με τον τρόπο αυτό ωστόσο, δεν δημιουργείται αυτόματα συντόμευση, ούτε εγκαθίστανται οι απαραίτητοι drivers.

Για να εκτελέσουμε το πρόγραμμα πρέπει να μεταβούμε με τον Windows Explorer μέσα στο φάκελο που δημιουργείται με την αποσυμπίεση και να κάνουμε διπλό κλικ στο αρχείο με όνομα `arduino.exe`. Με δεξί κλικ πάνω του μπορούμε να δημιουργήσουμε μία συντόμευση και να την μεταφέρουμε στην επιφάνεια εργασίας. Τέλος, μπαίνοντας στο φάκελο `drivers` μπορούμε να εγκαταστήσουμε τους απαραίτητους οδηγούς κάνοντας διπλό κλικ στο κατάλληλο αρχείο (`dpinst-amd64.exe` για Windows 64-bit και `dpinst-x86.exe` για Windows 32-bit).

1.2.2 Για τους χρήστες Linux

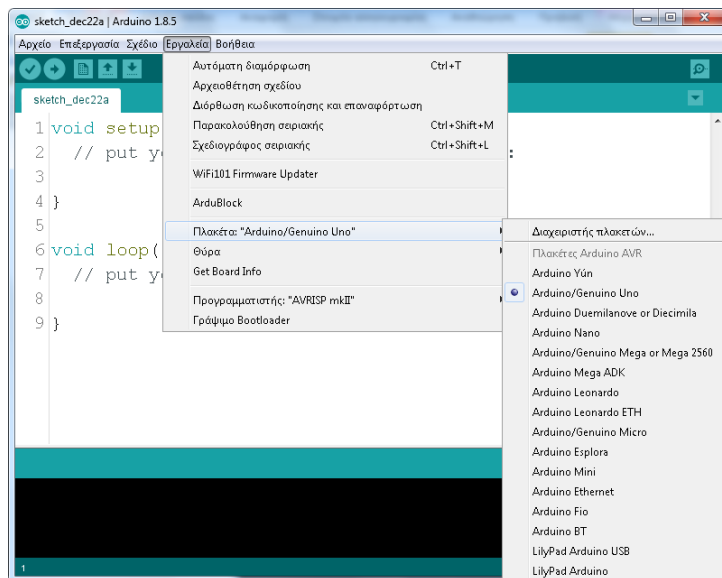
Βρίσκουμε το συμπιεσμένο αρχείο που κατεβάσαμε και το αποσυμπιέζουμε σε μία θέση της επιλογής μας. Στο σημείο αυτό το πρόγραμμα μπορεί να εκτελεστεί μεταβαίνοντας στο φάκελο που δημιουργήθηκε από την αποσυμπίεση και επιλέγοντας το αρχείο με όνομα `arduino`.

Για ακόμα καλύτερη χρηστικότητα, ανοίγουμε ένα τερματικό, μεταβαίνουμε στο φάκελο του προγράμματος και δίνουμε την εντολή `./install.sh`. Το script που εκτελείται με την εντολή αυτή, δημιουργεί μία συντόμευση στην επιφάνεια εργασίας, εισάγει το Arduino IDE στο μενού των εφαρμογών και συσχετίζει τα αρχεία τύπου `.ino` με την εφαρμογή.

1.3 Φόρτωση προγράμματος με το Arduino IDE

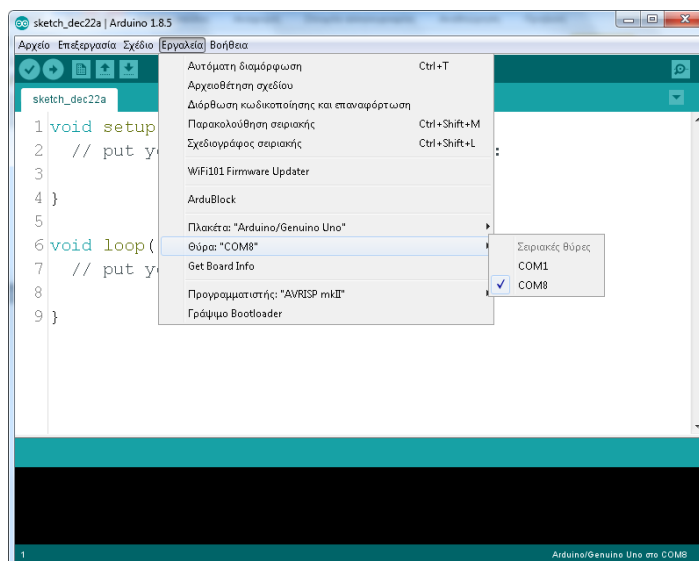
Για να μπορέσουμε να φορτώσουμε ένα πρόγραμμα από το Arduino IDE σε μία πλακέτα, πρέπει πρώτα να συνδέσουμε την πλακέτα με τον υπολογιστή μέσω ενός κατάλληλου καλωδίου USB. Αν έχουν φορτωθεί σωστά οι drivers ο υπολογιστής βλέπει την πλακέτα Arduino ως μία εικονική σειριακή θύρα (στα Windows COMx).

Το επόμενο βήμα είναι να ορίσουμε στο IDE τον τύπο της πλακέτας που έχουμε συνδέσει. Για το σκοπό αυτό, από το μενού πατάμε *Εργαλεία* και στη συνέχεια βάζουμε το ποντίκι πάνω στην επιλογή *Πλακέτα*. Από τη λίστα που εμφανίζεται κάνουμε κλικ στον κατάλληλο τύπο πλακέτας (στην περίπτωσή μας *Arduino/Genuino Uno*) (Εικόνα 9).



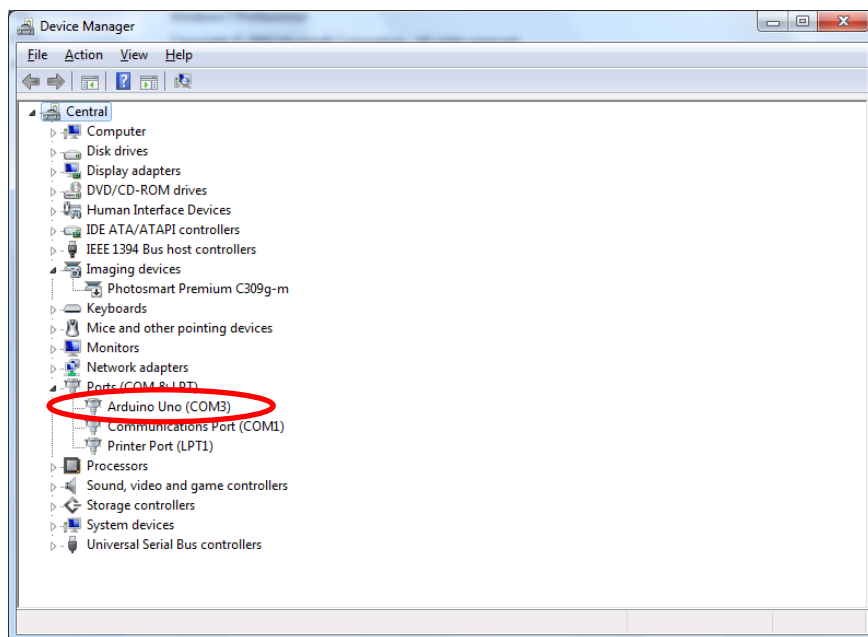
Εικόνα 9

Στη συνέχεια ορίζουμε τη σειριακή θύρα στην οποία βλέπει ο υπολογιστής μας την πλακέτα. Από το μενού πατάμε *Εργαλεία* και στη συνέχεια βάζουμε το ποντίκι πάνω στην επιλογή *Θύρα*. Εμφανίζεται μία λίστα με τις διαθέσιμες σειριακές θύρες του υπολογιστή και επιλέγουμε την κατάλληλη (Εικόνα 10).



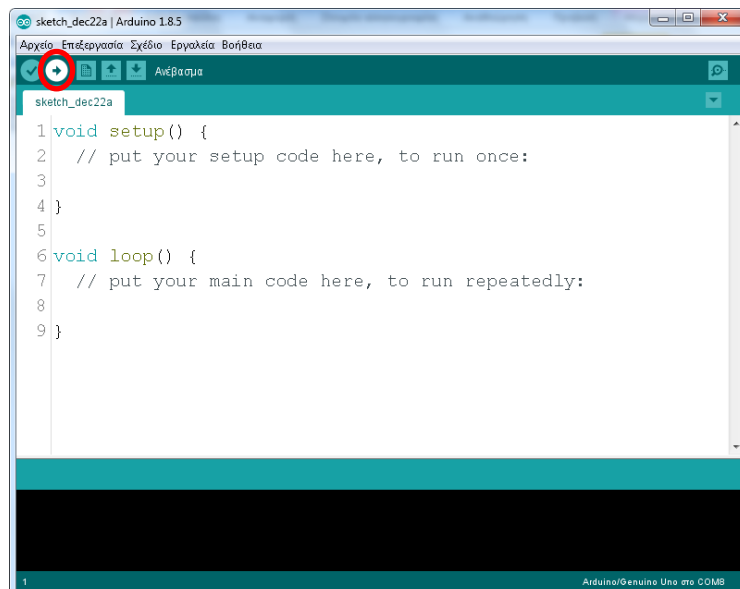
Εικόνα 10

Επειδή πρόκειται για εικονική θύρα, ο αριθμός της είναι συνήθως μεγάλος (> 2 . Στην Εικόνα 10 είναι η COM8). Αν δεν είμαστε βέβαιοι μπορούμε να το ελέγξουμε στη διαχείριση συσκευών του συστήματός μας (Εικόνα 11).



Εικόνα 11

Αφού ορίσουμε και τη σειριακή θύρα, μπορούμε να φορτώσουμε το πρόγραμμα στην πλακέτα πατώντας το σχετικό κουμπί στο περιβάλλον Arduino IDE (Εικόνα 12).



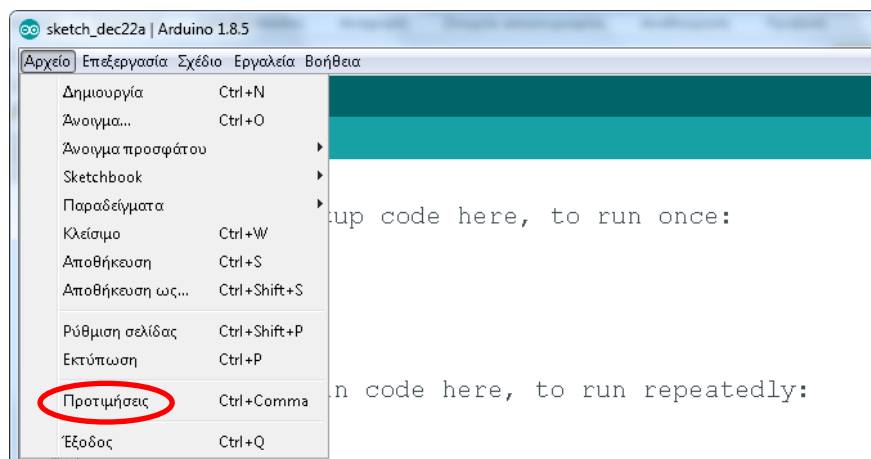
Εικόνα 12

2 ArduBlock

Το ArduBlock είναι ένα εργαλείο που εκτελείται μέσα από το IDE και μας παρέχει μία διεπαφή για τον οπτικό προγραμματισμό του Arduino. Κυκλοφορούν πολλές διαφορετικές εκδόσεις του ArduBlock. Αυτή που θα χρησιμοποιήσουμε είναι η 20171028 (διεύθυνση: <https://github.com/taweili/ardublock/releases> αρχείο: ardublock-all-master-beta-20171028.jar), η οποία είναι συμβατή με τις πιο πρόσφατες εκδόσεις του Arduino IDE.

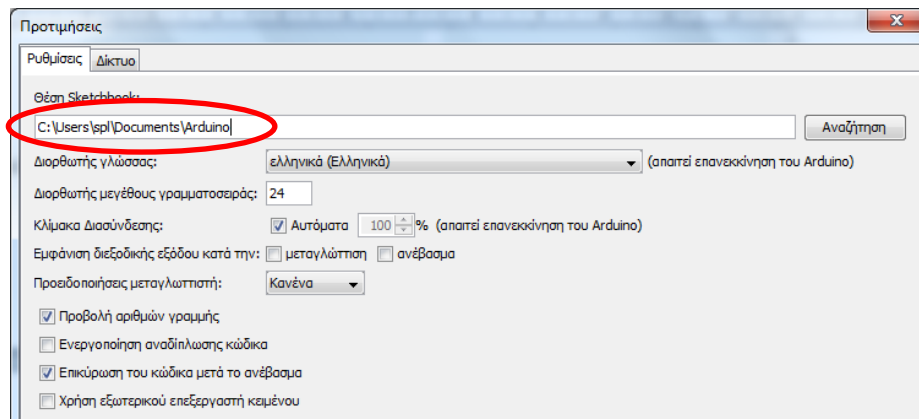
2.1 Ενσωμάτωση του ArduBlock στο IDE

Εκτελούμε το Arduino IDE και από το μενού *Αρχείο* επιλέγουμε *Προτιμήσεις* (Εικόνα 13).



Εικόνα 13

Στο παράθυρο που ανοίγει σημειώνουμε τη θέση του Sketchbook, δηλαδή του καταλόγου όπου αποθηκεύονται τα προγράμματά που φτιάχνουμε με το IDE (Εικόνα 14).

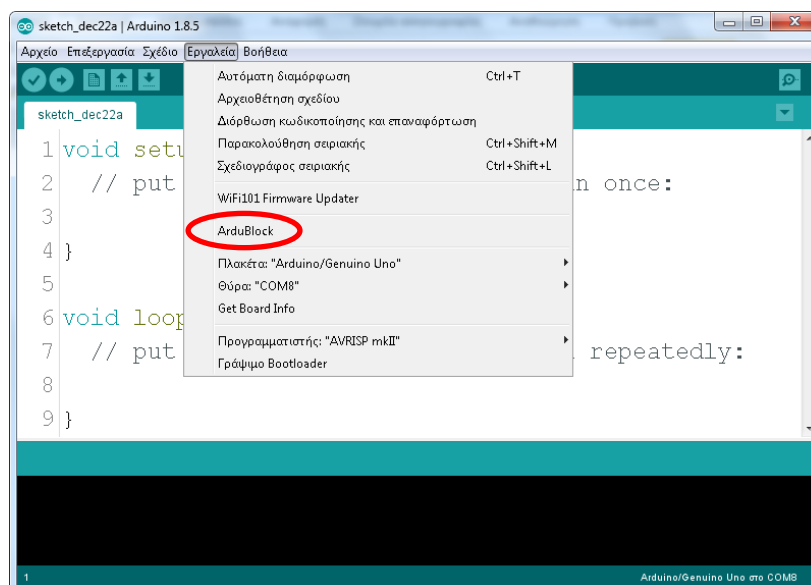


Εικόνα 14

Για την ενσωμάτωση του ArduBlock μέσα στο IDE, πρέπει κάτω από τον κατάλογο του Sketchbook να δημιουργηθεί η δομή καταλόγων `tools\ArduBlockTool\tool` και στη συνέχεια να τοποθετηθεί μέσα στο φάκελο `tool` το σχετικό αρχείο jar (για παράδειγμα: `C:\Users\spl\Documents\Arduino\tools\ArduBlockTool\tool\ardublock-all-master-beta-20171028.jar`).

Για λόγους ευκολίας, μαζί με τις οδηγίες, θα σας δοθεί συμπιεσμένος ο φάκελος `tools` με όλα τα απαιτούμενα περιεχόμενα. Συνεπώς, το μοναδικό που χρειάζεται να γίνει, είναι να αποσυμπίεστεί το αρχείο `tools.zip` μέσα στον κατάλογο του Sketchbook.

Στην επόμενη εκκίνηση του IDE, το ArduBlock θα εμφανιστεί ως επιλογή μέσα στο μενού *Εργαλεία* (Εικόνα 15).



Εικόνα 15

Α΄ Γυμνασίου

3 Εισαγωγή

Στην ενότητα αυτή θα παρουσιαστούν το Arduino UNO και το breadboard, που αποτελούν τα βασικά εξαρτήματα για όλες τις εφαρμογές που θα αναπτυχθούν στη συνέχεια. Επιπλέον, θα γίνει μία σύντομη εισαγωγή στη χρήση του ArduBlock, που θα χρησιμοποιηθεί για την ανάπτυξη των προγραμμάτων των εφαρμογών.

3.1 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.

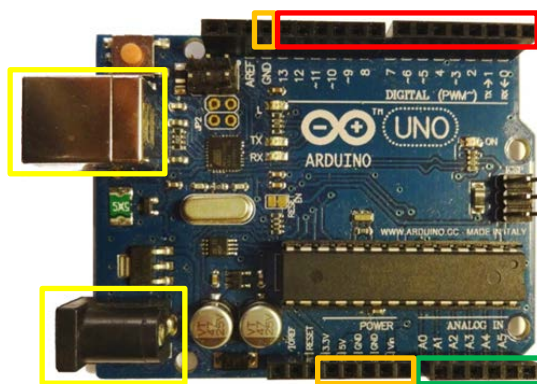
Πίνακας 1

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

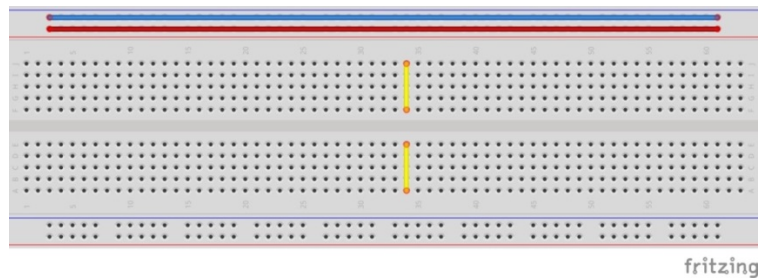
Στα αριστερά της εικόνας μπορούμε να διακρίνουμε στο πάνω μέρος την υποδοχή USB που χρησιμοποιείται για φόρτωση προγράμματος, για σειριακή επικοινωνία με τον υπολογιστή αλλά και για τροφοδοσία. Στο κάτω μέρος υπάρχει η υποδοχή για την εξωτερική τροφοδοσία 7 - 12 V.



Εικόνα 16 Arduino UNO

3.2 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 17. 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.

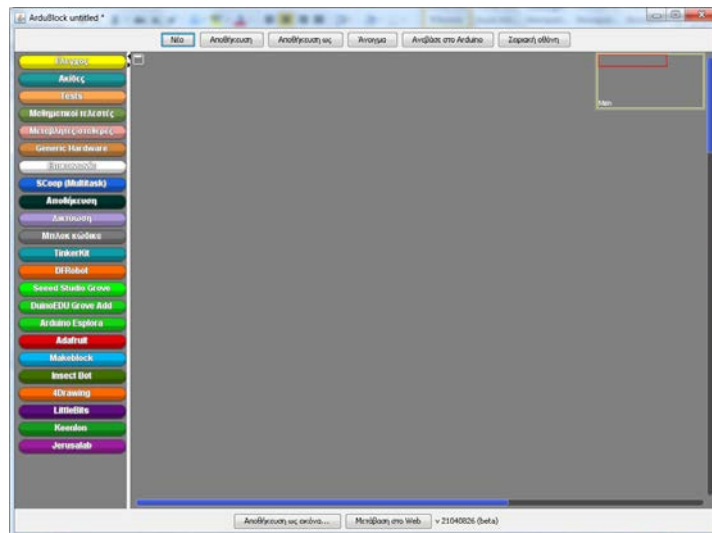


Εικόνα 17 Breadboard

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.

3.3 Χρήση του ArduBlock

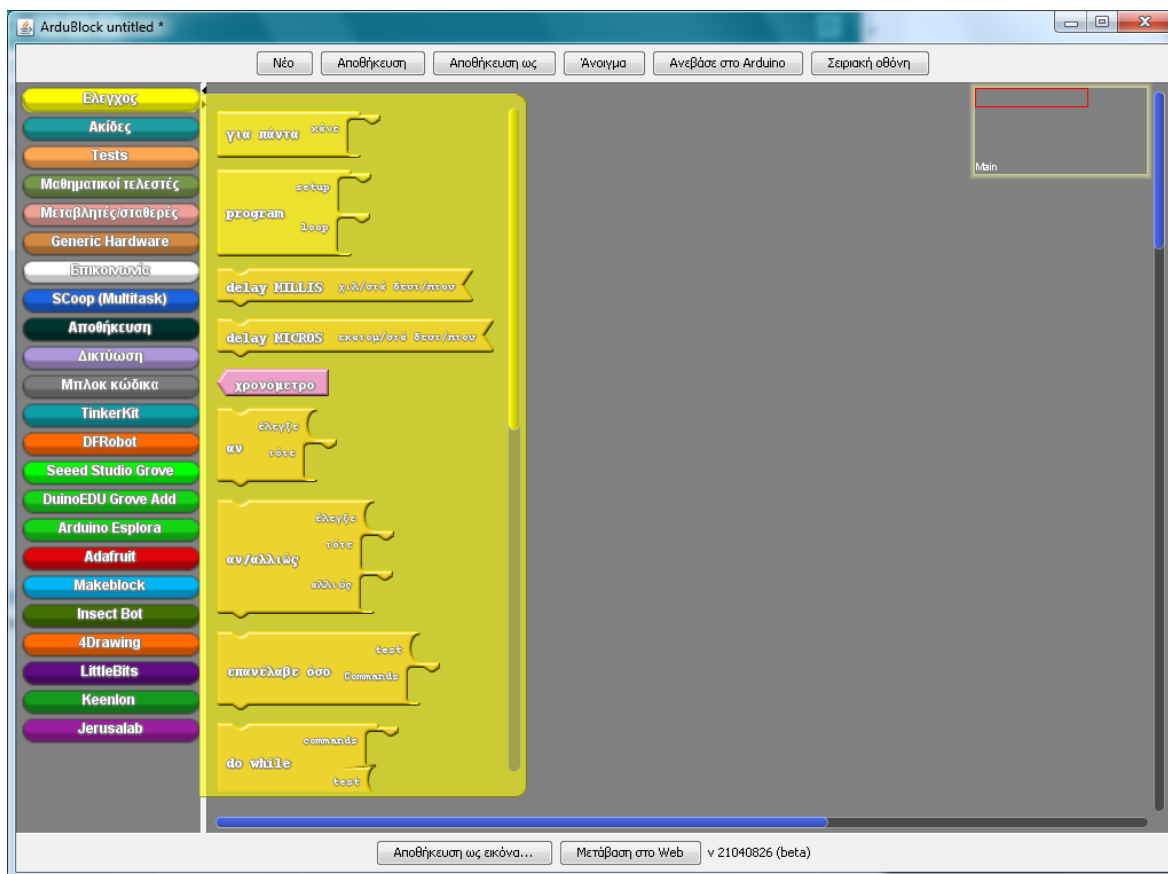
Το ArduBlock είναι ένα εργαλείο λογισμικού που εκτελείται μέσα από το Arduino IDE και μας παρέχει μία διεπαφή για τον οπτικό προγραμματισμό του Arduino. Πατώντας την επιλογή *ArduBlock*, μέσα από το μενού *Εργαλεία* του IDE, ανοίγει το παράθυρο που παρουσιάζεται στην Εικόνα 18.



Εικόνα 18

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 19). 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.

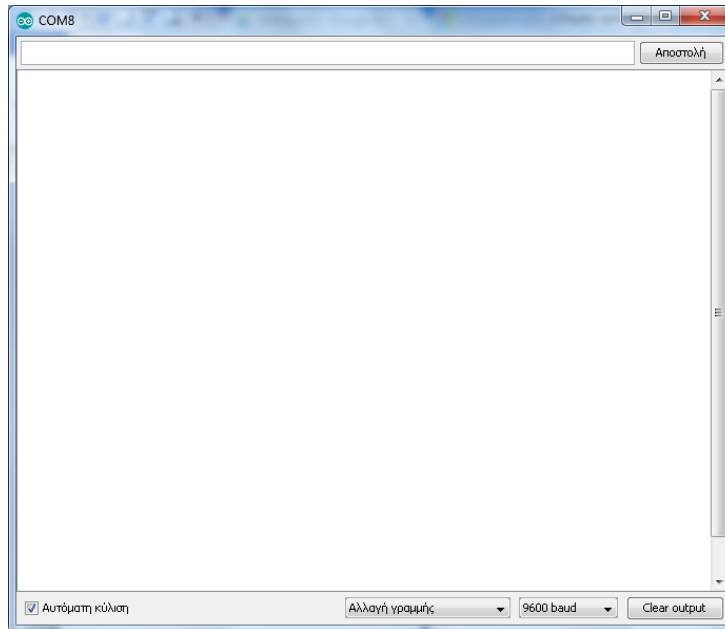


Εικόνα 19

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 ArduBlock 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 ArduBlock, 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 20), μέσα

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).



Εικόνα 20

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

4 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.

4.1 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 21 LED

Ως δίοδος, το LED επιτρέπει τη διέλευση του ρεύματος μόνο προς μία φορά. Όπως φαίνεται και στην Εικόνα 21 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).

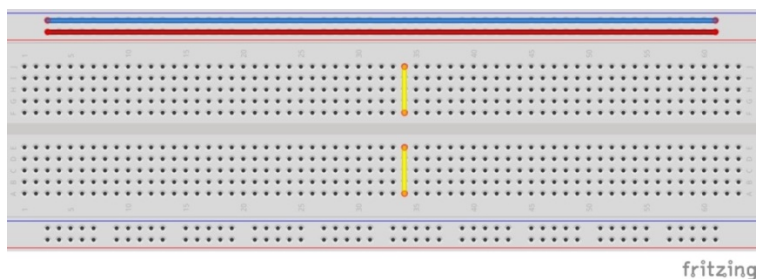
Αντίσταση

Ανάμεσα στο LED και την πηγή πρέπει να παρεμβάλλεται μία αντίσταση (Εικόνα 22), 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 22 Αντίσταση

Breadboard

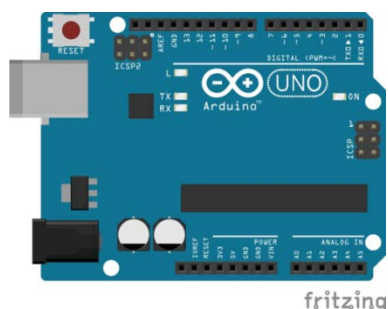


Εικόνα 23 Breadboard

Arduino UNO

Το Arduino UNO (Εικόνα 24) διαθέτει 14 ψηφιακούς ακροδέκτες εισόδου/εξόδου (0 – 13). Όταν οι ψηφιακοί ακροδέκτες χρησιμοποιούνται ως εξόδοι, 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.



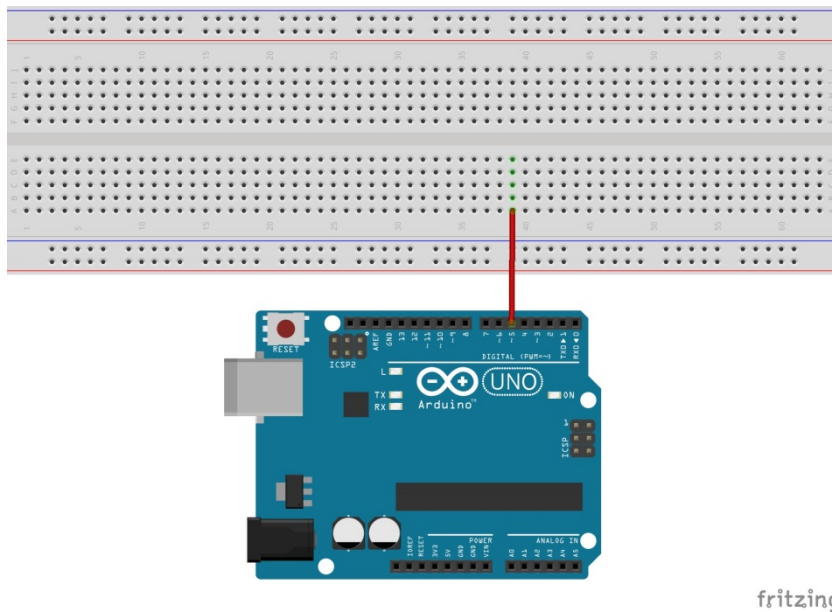
Εικόνα 24 Arduino UNO

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.

4.2 Κατασκευή κυκλώματος

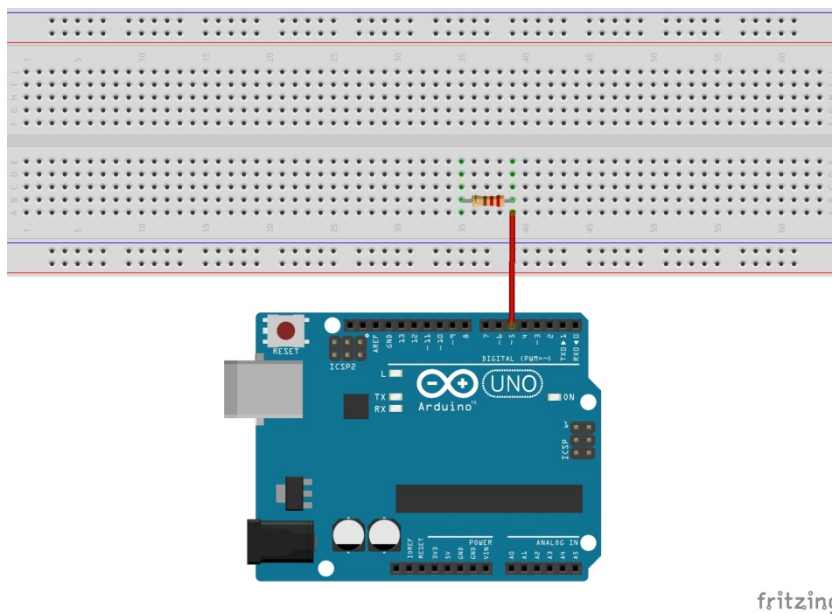
Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος.



fritzing

Βήμα 1

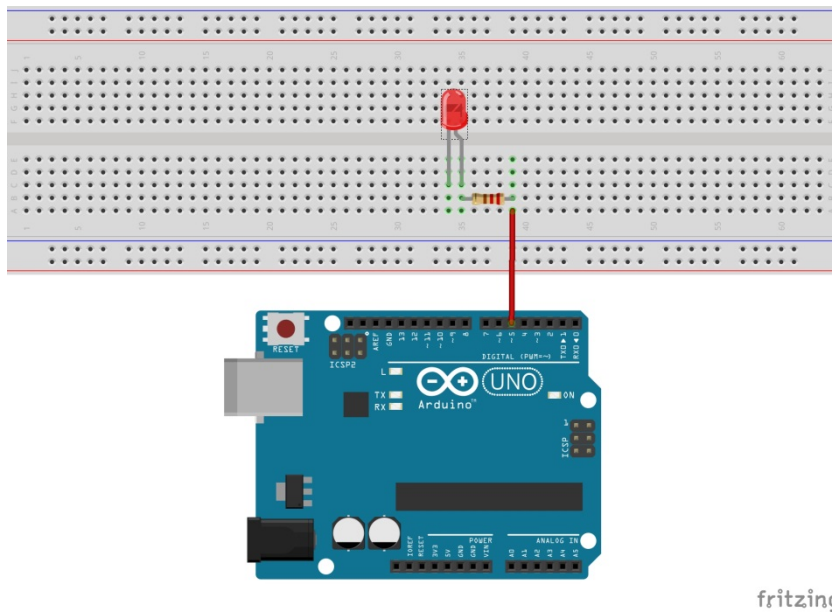
Συνδέουμε ένα καλώδιο από το pin 5 του Arduino σε μία ελεύθερη στήλη του breadboard.



fritzing

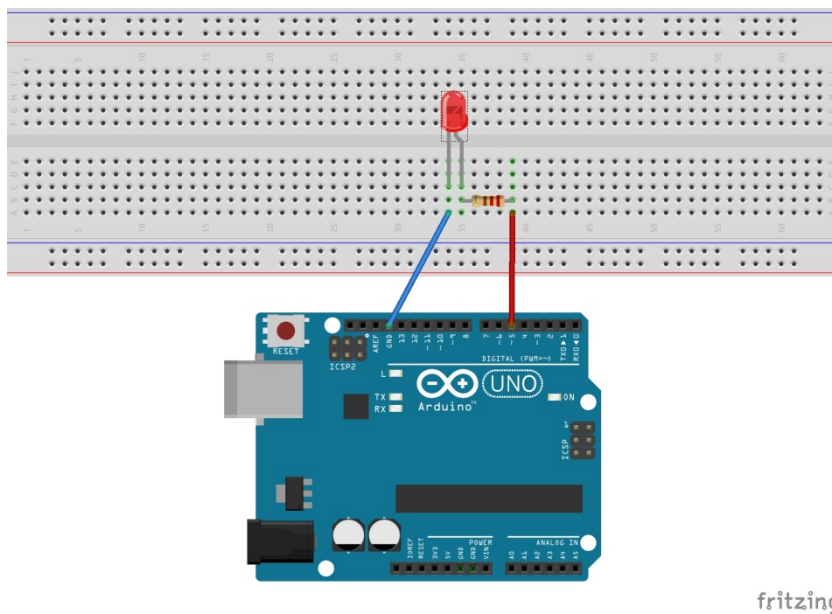
Βήμα 2

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.

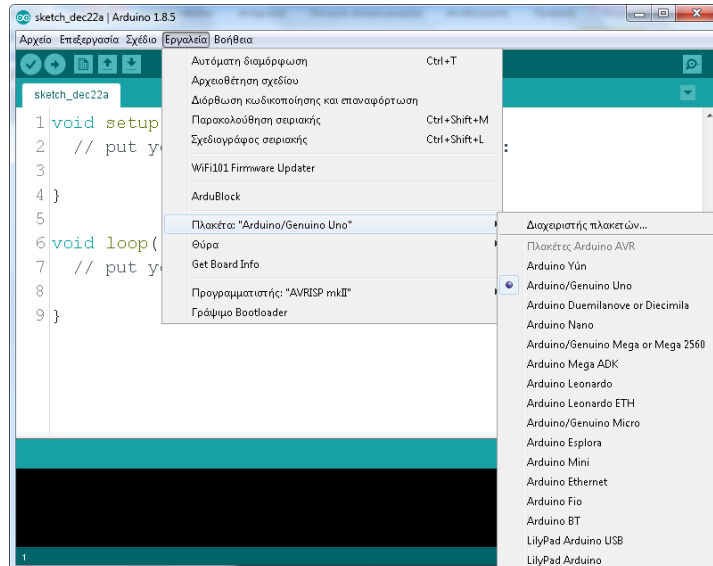


Βήμα 4

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

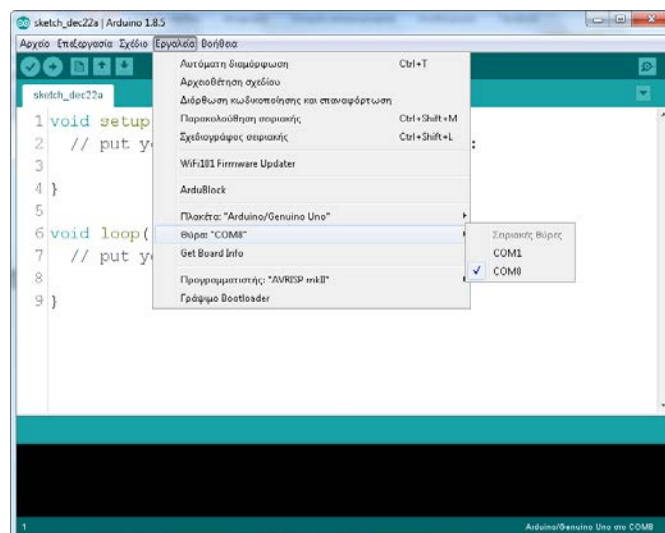
4.3 Σύνδεση πλακέτας με τον υπολογιστή

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 25).



Εικόνα 25

Ακολούθως, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino (Εικόνα 26).



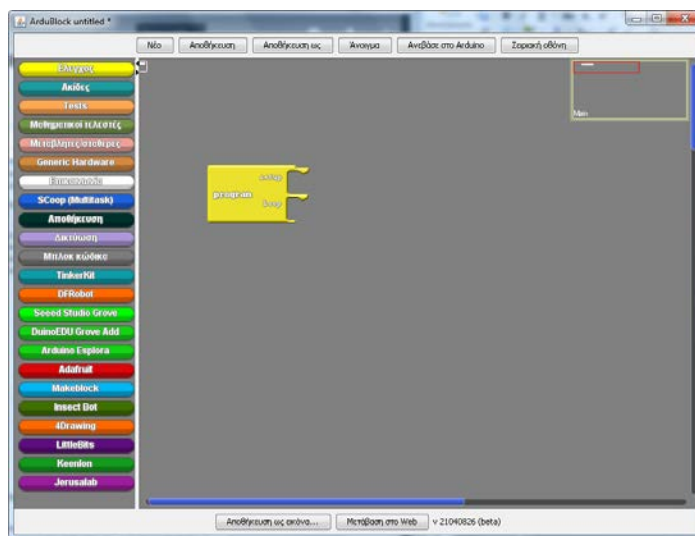
Εικόνα 26

4.4 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^ο βήμα.)

Βήμα 1^ο

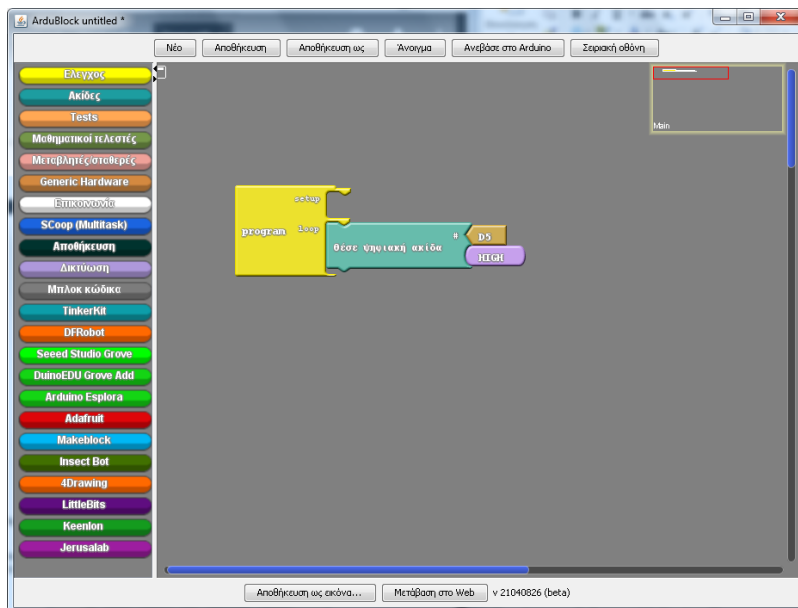
Αφαιρούμε το πλακίδιο *για πάντα κάνε* και στη συνέχεια επιλέγουμε το κουμπί *Έλεγχος* και σύρουμε το πλακίδιο *program* στο κεντρικό παράθυρο (Εικόνα 27). Το πλακίδιο αυτό αντιπροσωπεύει τη βασική δομή ενός προγράμματος Arduino. Οι εντολές που τοποθετούνται μέσα στο *setup* εκτελούνται μόνο μία φορά όταν ξεκινάει η εκτέλεση του προγράμματος, ενώ η συνάρτηση *loop* εκτελείται συνέχεια (επαναλαμβανόμενα) όσο υπάρχει τροφοδοσία.



Εικόνα 27

Βήμα 2^ο

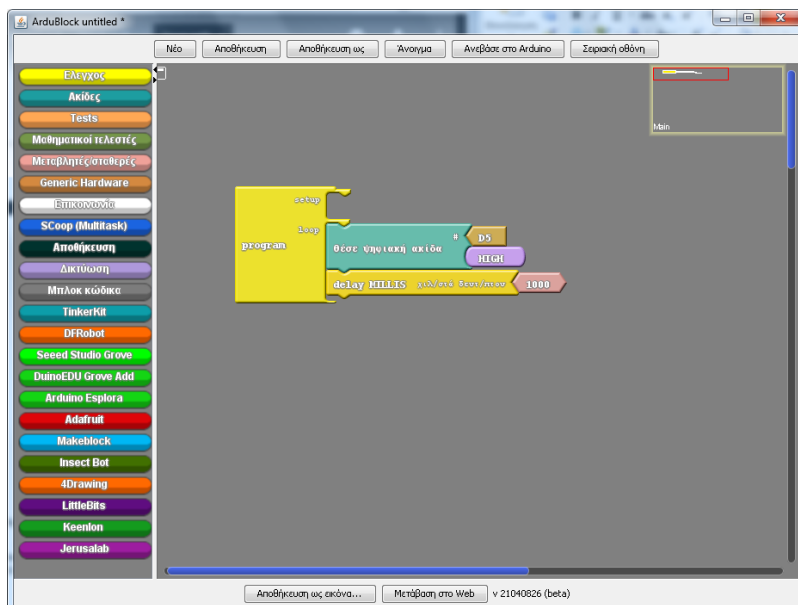
Από την καρτέλα *Ακίδες* σύρουμε το πλακίδιο *θέσε ψηφιακή ακίδα* μέσα στο *loop*, ώστε να κουμπώσει. Πηγαίνουμε το ποντίκι πάνω από τον αριθμό της ακίδας, πατάμε στο βελάκι που εμφανίζεται και επιλέγουμε από τη λίστα τον ακροδέκτη D5 για να αντιστοιχεί στην ακίδα όπου έχει συνδεθεί το LED (Εικόνα 28).



Εικόνα 28

Βήμα 3°

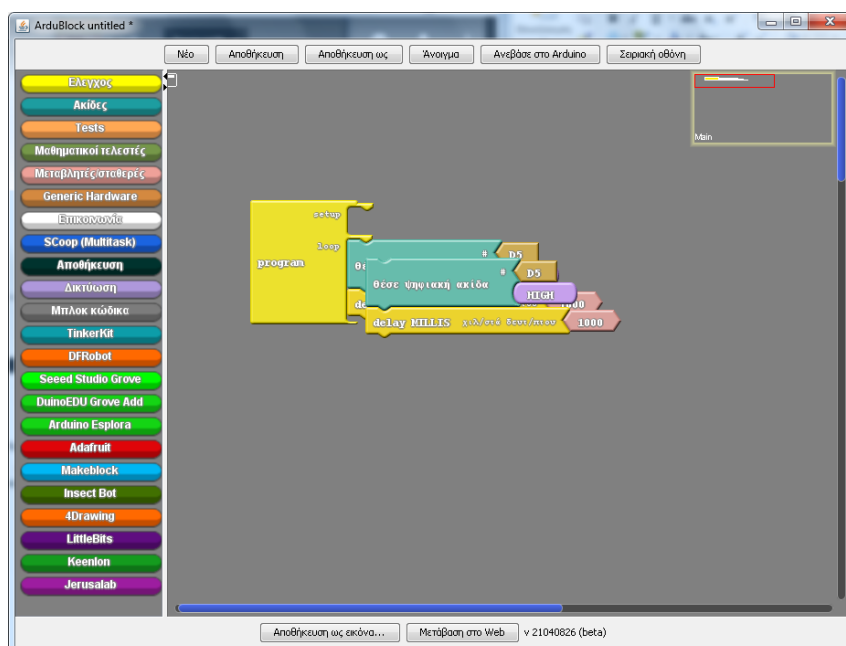
Από την καρτέλα *Έλεγχος*, προσθέτουμε την εντολή *delay* *MILLIS* κάτω από την εντολή *θέσε ψηφιακή ακίδα* και αφήνουμε την τιμή καθυστέρησης 1000 (Εικόνα 29). Με τον τρόπο αυτό το LED θα μένει αναμμένο για 1 δευτερόλεπτο.



Εικόνα 29

Βήμα 4°

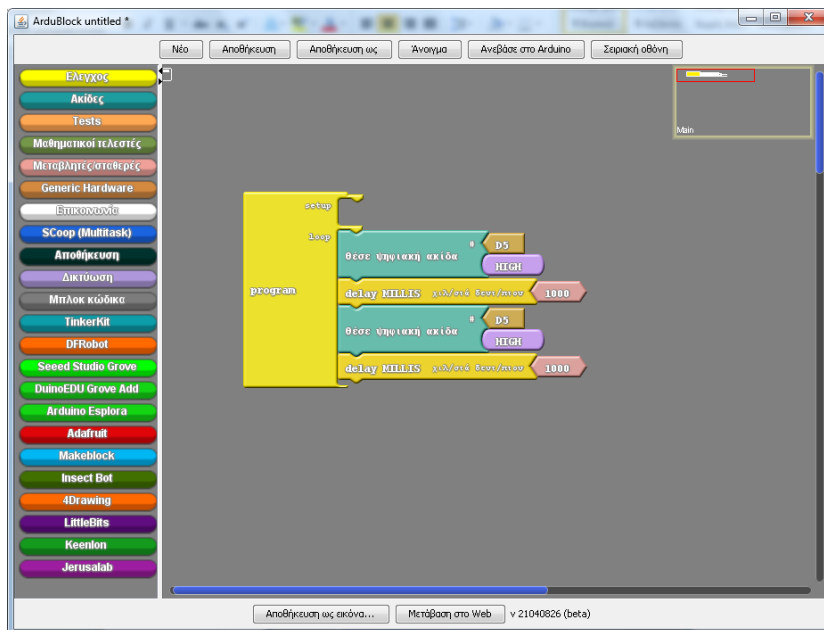
Κάνουμε δεξί κλικ πάνω στην εντολή *θέσε ψηφιακή ακίδα* που προσθέσαμε στο 1° βήμα και επιλέγουμε *Αντέγραψε*. (Εικόνα 30)



Εικόνα 30

Βήμα 5°

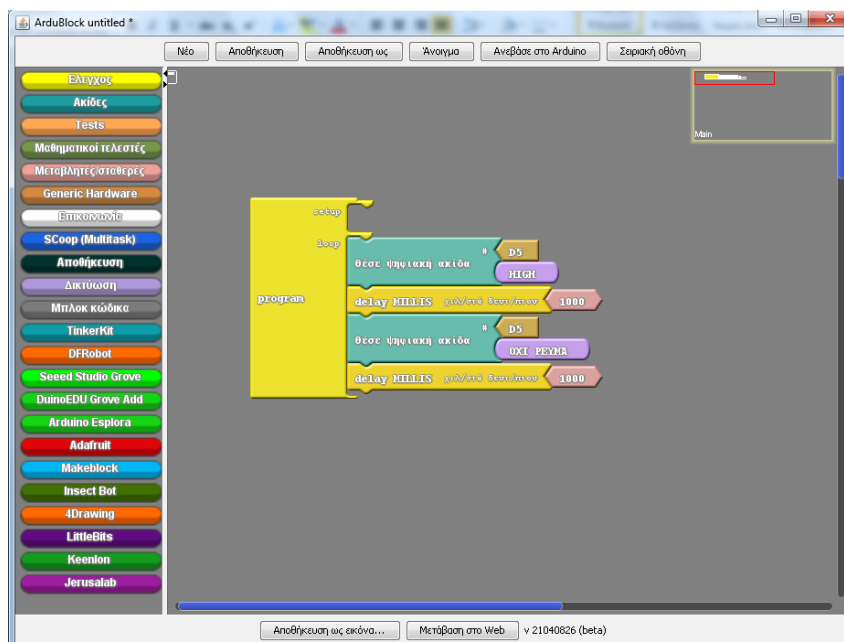
Σύρουμε τις εντολές που εμφανίστηκαν με την αντιγραφή, ώστε να κουμπώσουν κάτω από την εντολή *delay MILLIS* του βήματος 3. (Εικόνα 31)



Εικόνα 31

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 32). Οι εντολές αυτές κάνουν το LED να παραμένει σβηστό για ένα δευτερόλεπτο.



Εικόνα 32

Βήμα 7^ο

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.

5 Εφαρμογή 2: Φανάρι Κυκλοφορίας

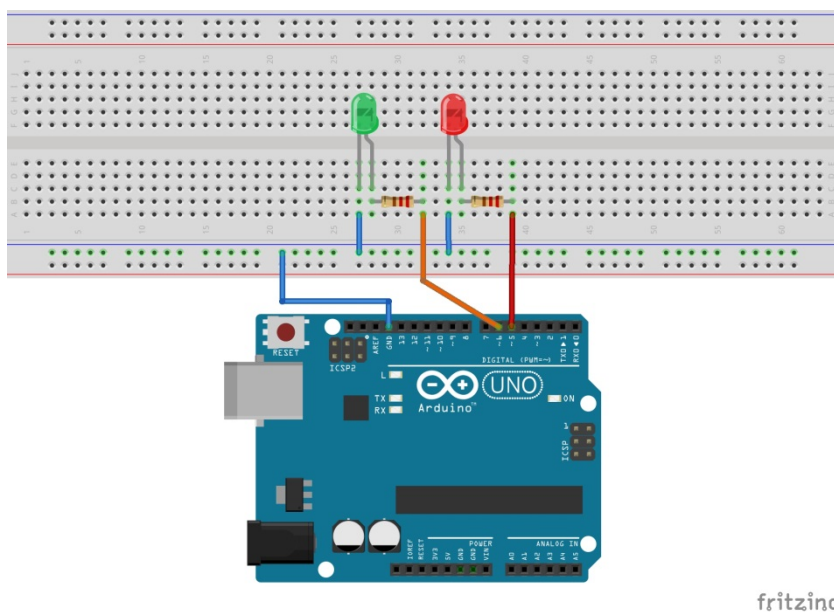
Η εφαρμογή που θα αναπτυχθεί θα προσομοιώνει τη λειτουργία ενός φαναριού κυκλοφορίας. Συγκεκριμένα, θα χρησιμοποιηθούν δύο LED (ένα κόκκινο και ένα πράσινο), τα οποία θα ανάβουν εναλλάξ με προκαθορισμένες διάρκειες. Στα πλαίσια της εφαρμογής, παρουσιάζεται η αξιοποίηση του breadboard σε πιο σύνθετα κυκλώματα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

5.1 Υλικά

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

5.2 Κύκλωμα

Το κύκλωμα της εφαρμογής παρουσιάζεται στην Εικόνα 33. 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^η εφαρμογή), ενώ η άνοδος του πράσινου LED συνδέεται μέσω της 2^{ης} αντίστασης στον ψηφιακό ακροδέκτη 6.



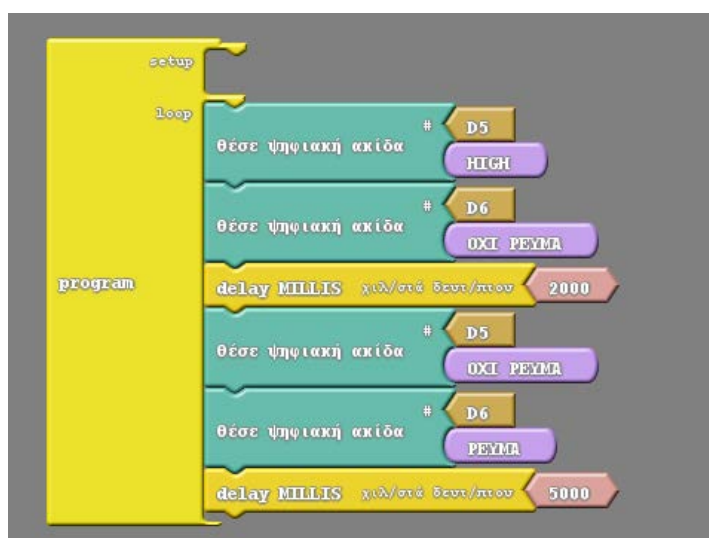
Εικόνα 33

5.3 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.

5.4 Πρόγραμμα

Από το μενού *Εργαλεία* του IDE επιλέγουμε *ArduBlock*. Στην Εικόνα 34 παρουσιάζεται ο κώδικας της εφαρμογής.



Εικόνα 34

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 εντολής *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*).

Μπορείτε να βρείτε το πρόγραμμα έτοιμο, στο συνοδευτικό αρχείο με όνομα *Traffic_Light.abp* και να το ανοίξετε στο περιβάλλον του ArduBlock με το κουμπί *Άνοιγμα*.

Ανεβάστε το πρόγραμμα στο Arduino σύμφωνα με τις οδηγίες της πρώτης εφαρμογής και δοκιμάστε τη λειτουργία του.

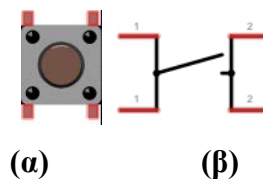
6 Εφαρμογή 3: LED με κουμπί

Οι ψηφιακοί ακροδέκτες του Arduino (ακροδέκτες 0 - 13 στο UNO) μπορούν να χρησιμοποιηθούν είτε ως έξοδοι είτε ως εισόδοι. Προκειμένου να γνωρίσουμε τη χρήση των ψηφιακών εισόδων του Arduino, θα προσθέσουμε στην 1^η 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.

6.1 Υλικά

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

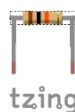
Κουμπί



Εικόνα 35

Όπως φαίνεται στην Εικόνα 35(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 35(β)). Όταν πιέζουμε το κουμπί, κλείνει ο διακόπτης και συνδέεται το δεξί με το αριστερό του μέρος.

Αντίσταση 10kΩ

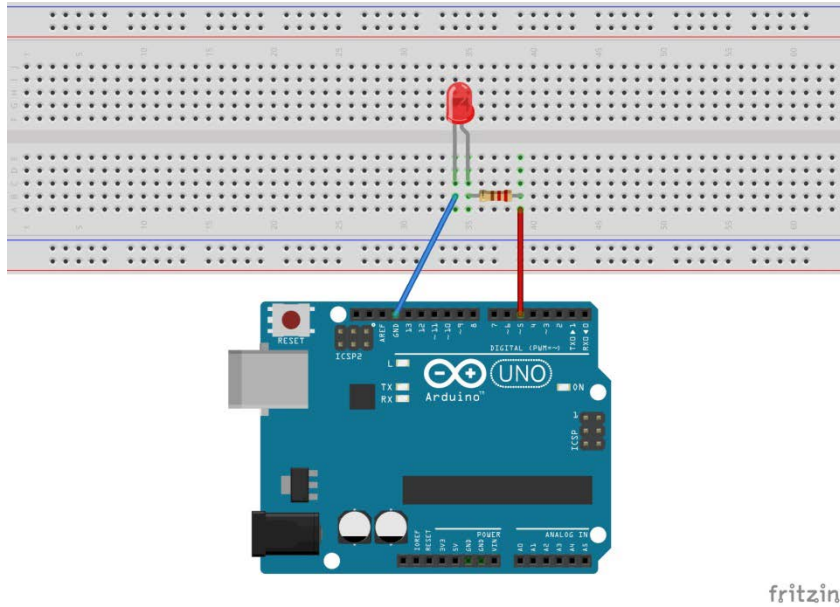


Εικόνα 36

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

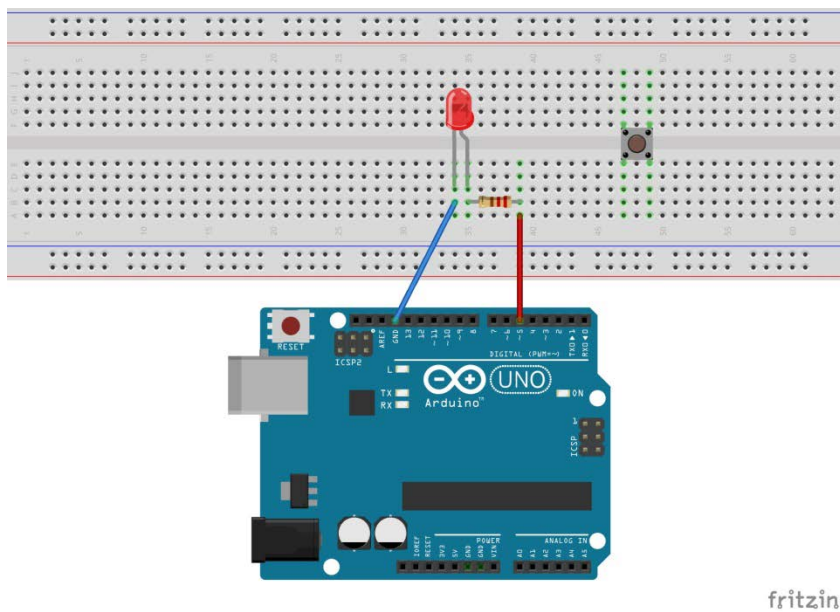
6.2 Κύκλωμα

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^{ης} εφαρμογής (Εικόνα 37).



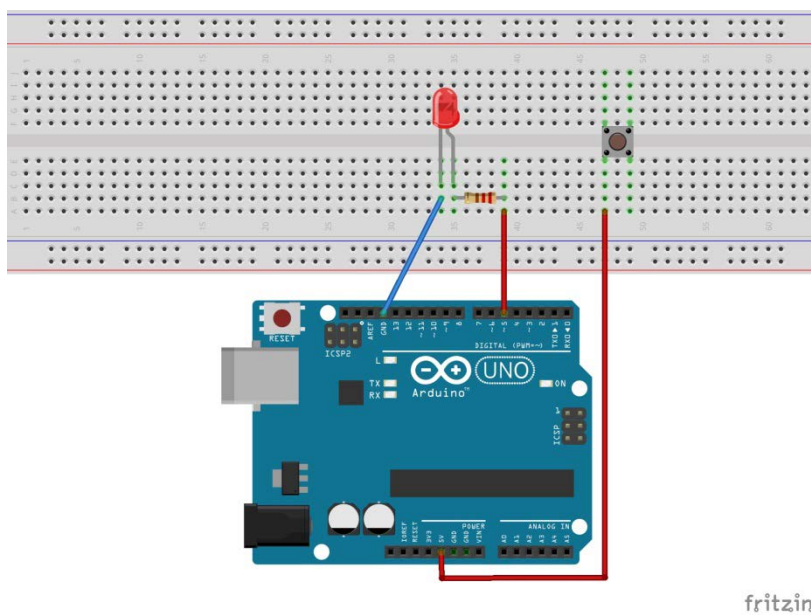
Εικόνα 37

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



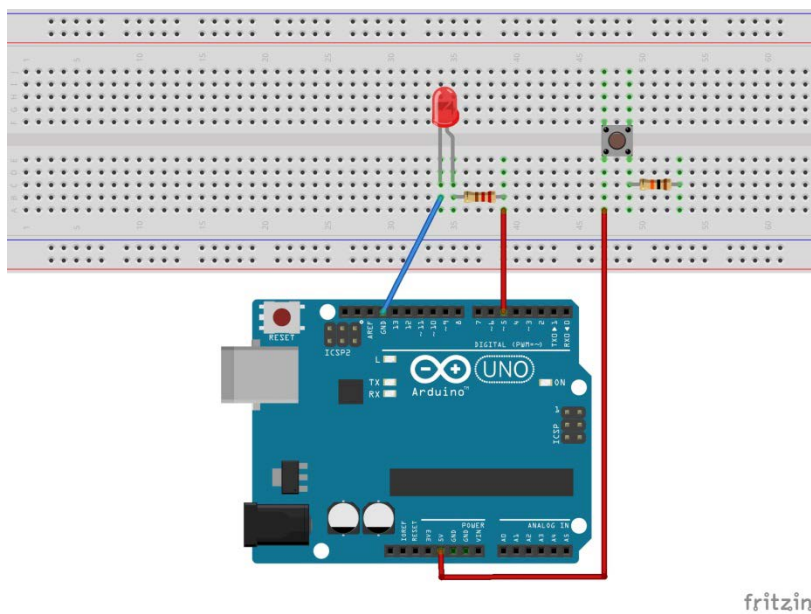
Βήμα 1

Τοποθετούμε το κουμπί πάνω στο breadboard, όπως φαίνεται στην εικόνα. Με τον τρόπο αυτό, οι συνδεδεμένοι ακροδέκτες (Εικόνα 35(β)) γεφυρώνουν τα 2 ανεξάρτητα τμήματα των στηλών στις οποίες πατάνε.



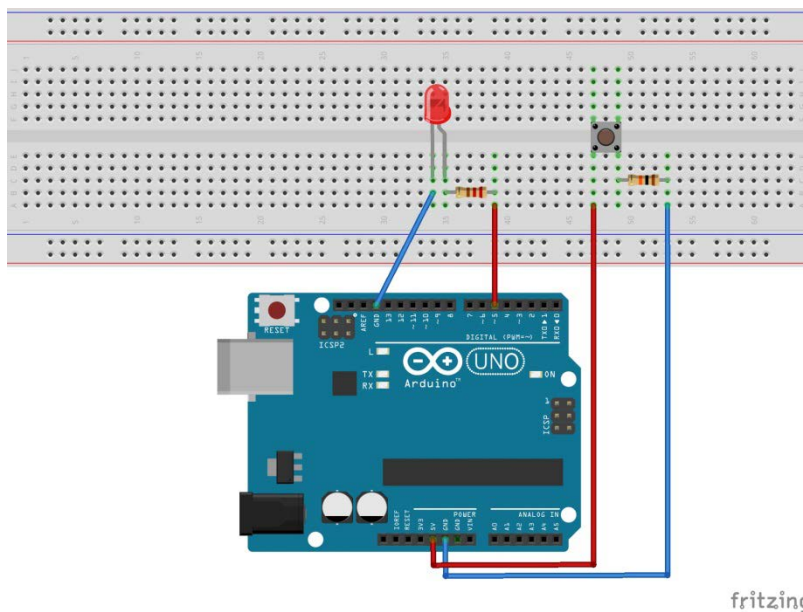
Βήμα 2

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.



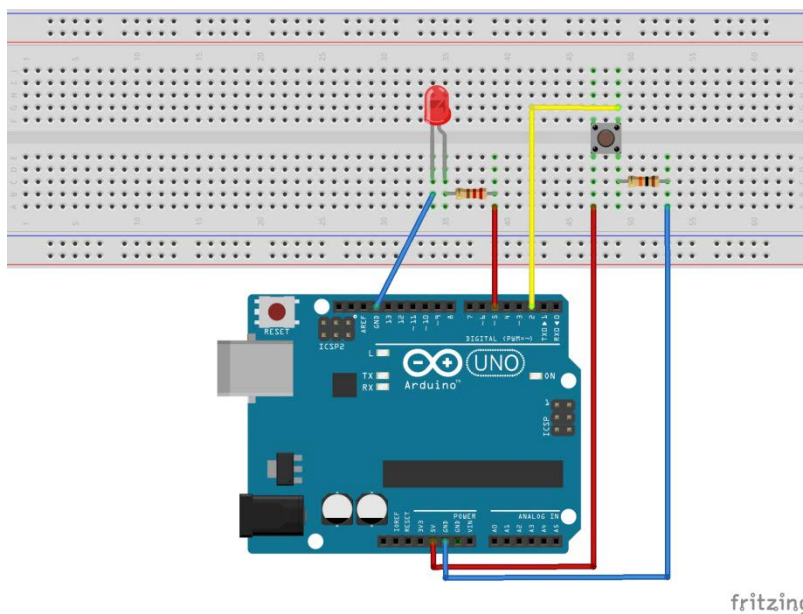
Βήμα 3

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 επεξηγείται η συνδεσμολογία και η λειτουργία του κουμπιού.

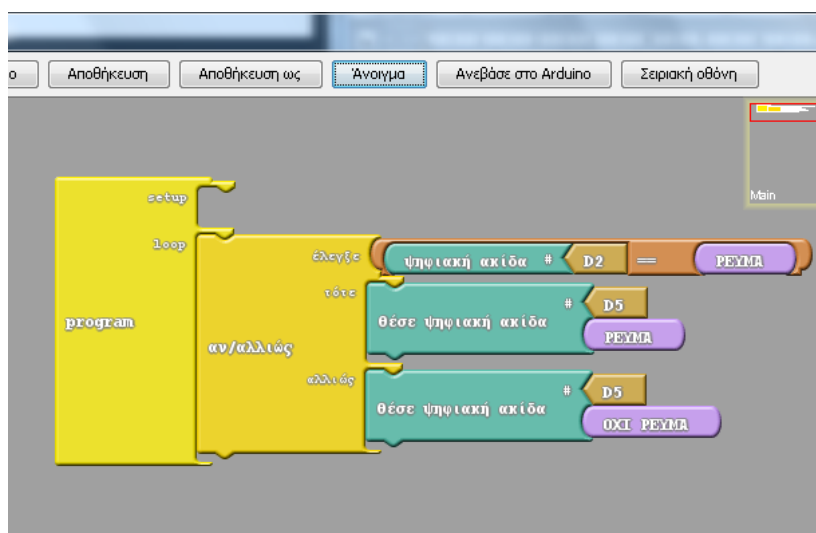
Το αριστερό τμήμα του κουμπιού συνδέεται στην τροφοδοσία (pin 5 V) και το δεξί στη γείωση (GND) μέσω της αντίστασης 10 kΩ, για αποφυγή βραχυκυκλώματος. Για τον έλεγχο της κατάστασης του κουμπιού, συνδέουμε το δεξί του τμήμα στον ψηφιακό ακροδέκτη 2 του Arduino, που θα χρησιμοποιηθεί ως είσοδος. Όταν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

6.3 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.

6.4 Πρόγραμμα

Από το μενού *Εργαλεία* του IDE επιλέγουμε *ArduBlock* και προχωράμε στην ανάπτυξη του προγράμματος της εφαρμογής, όπως παρουσιάζεται στην Εικόνα 38. 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 38

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.

Μπορείτε να βρείτε το έτοιμο πρόγραμμα στο συνοδευτικό αρχείο με όνομα *LED_w_Button.abp* και να το ανοίξετε στο περιβάλλον του ArduBlock με το κουμπί *Άνοιγμα*.

Ανεβάστε το πρόγραμμα στο Arduino και δοκιμάστε τη λειτουργία του.

7 Εφαρμογή 4: LED που αναβοσβήνει και LED με κουμπί

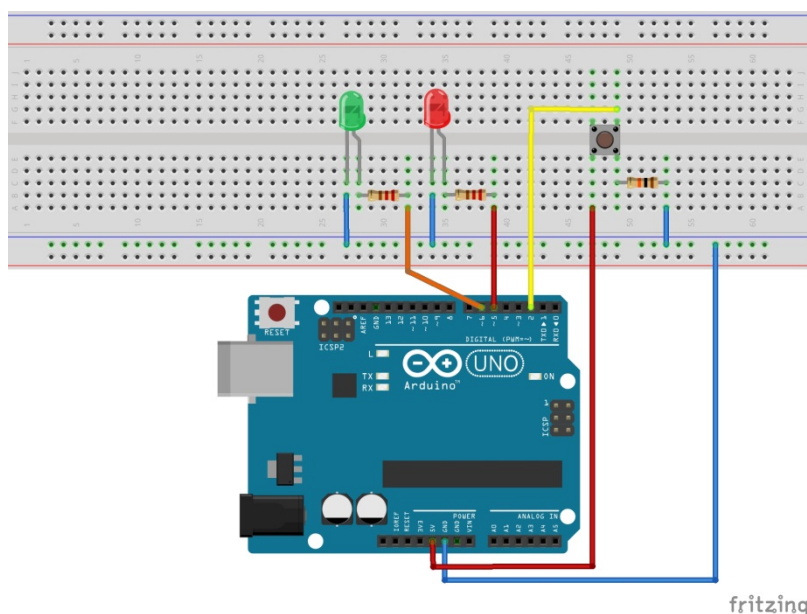
Αυτή η εφαρμογή αποτελεί έναν συνδυασμό των Εφαρμογών 1 και 3. Συγκεκριμένα, θα υπάρχουν δύο LED και ένα κουμπί. Το ένα LED θα αναβοσβήνει περιοδικά (όπως στην Εφαρμογή 1), ενώ το δεύτερο θα ανάβει όσο το κουμπί είναι πατημένο (Εφαρμογή 3). Στα πλαίσια της ανάπτυξης του προγράμματος, θα εντοπίσουμε ένα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

7.1 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.

7.2 Κύκλωμα

Όπως αναφέρθηκε και προηγούμενα, 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 39.



Εικόνα 39

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).

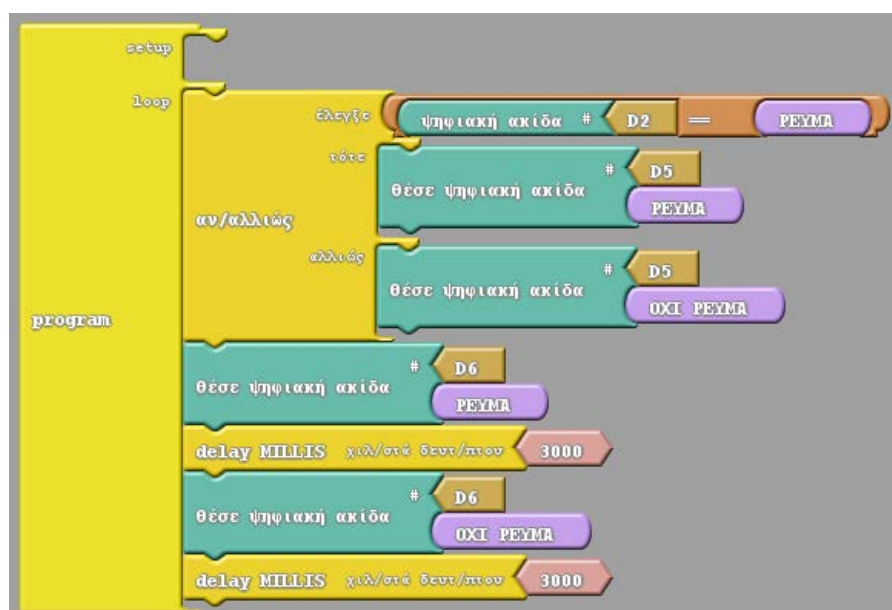
7.3 Σύνδεση πλακέτας με τον υπολογιστή

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

7.4 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 40 παρουσιάζεται ο αρχικός κώδικας της εφαρμογής.



Εικόνα 40

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

παραμένει αναμμένο για 3 δευτερόλεπτα, να σβήσει και να παραμένει σβηστό για 3 δευτερόλεπτα.

Μπορείτε να βρείτε το έτοιμο πρόγραμμα στο συνοδευτικό αρχείο με όνομα *LED_Button_Blink_A.abp* και να το ανοίξετε στο περιβάλλον του ArduBlock με το κουμπί *Άνοιγμα*.

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

Β' Γυμνασίου

8 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.

8.1 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 42 LED

Ως δίοδος, το LED επιτρέπει τη διέλευση του ρεύματος μόνο προς μία κατεύθυνση. Όπως φαίνεται και στην Εικόνα 42 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 43), 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.

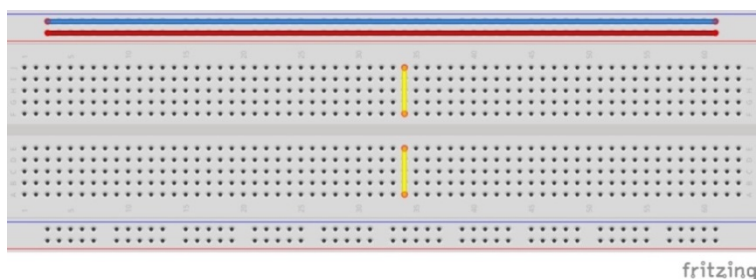


Εικόνα 43 Αντίσταση

Breadboard

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

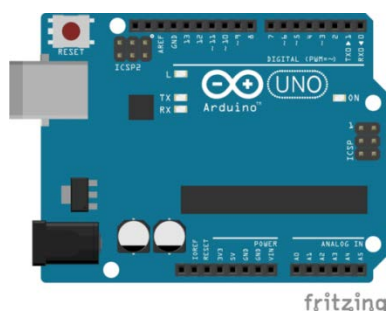
οποίες μπορούν να συνδεθούν διάφορα ηλεκτρικά και ηλεκτρονικά στοιχεία. Εσωτερικά οι οπές αυτές διασυνδέονται μεταξύ τους όπως φαίνεται στην Εικόνα 44. Οι 4 εξωτερικές σειρές (2 πάνω και 2 κάτω) είναι συνδεδεμένες οριζόντια, ενώ στο κεντρικό του τμήμα οι διασυνδέσεις είναι κατακόρυφες, με κάθε στήλη να χωρίζεται σε δύο ανεξάρτητα τμήματα.



Εικόνα 44

Arduino UNO

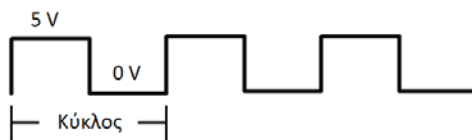
Το Arduino UNO (Εικόνα 45) διαθέτει 14 ψηφιακούς ακροδέκτες. Κάποιοι από αυτούς (οι 3, 5, 6, 9, 10 και 11), μπορούν μέσω μίας τεχνικής που ονομάζεται PWM να δίνουν ως έξοδο και ενδιάμεσες τιμές τάσης (μεταξύ 0V και 5V). Συγκεκριμένα, μπορούν να δίνουν 256 διαφορετικά επίπεδα τάσης, με το επίπεδο 0 να αντιστοιχεί στα 0 V, το επίπεδο 255 στα 5 V, το επίπεδο 127 στα 2,5 V κ.ο.κ.



Εικόνα 45

Με την τεχνική PWM οι στάθμες 0V και 5V εναλλάσσονται περιοδικά, παράγοντας έναν τετραγωνικό παλμό (Εικόνα 46). 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.



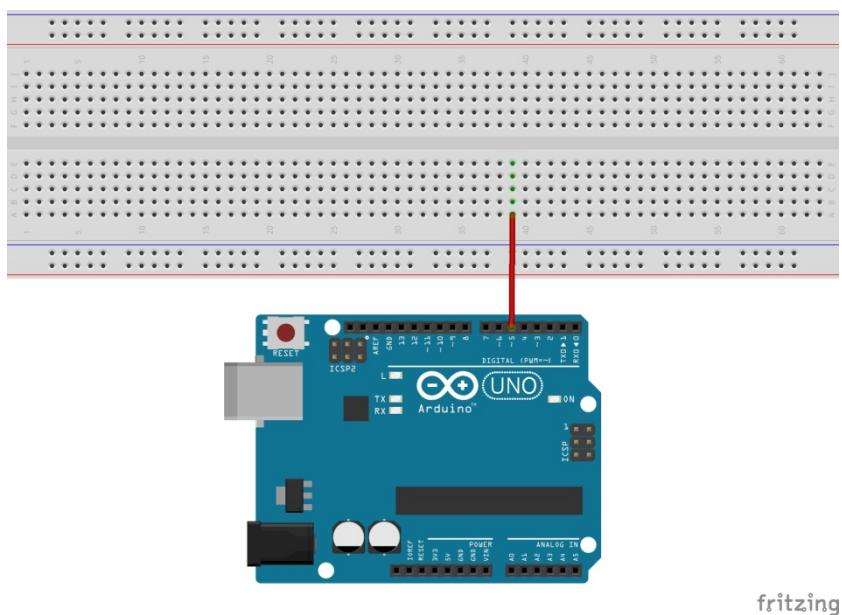
Εικόνα 46

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.

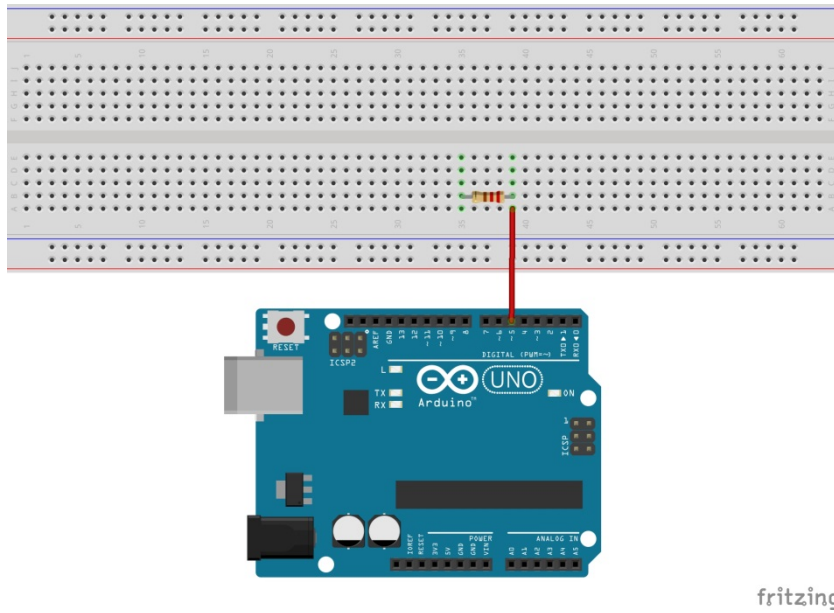
8.2 Κύκλωμα

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος.



Βήμα 1

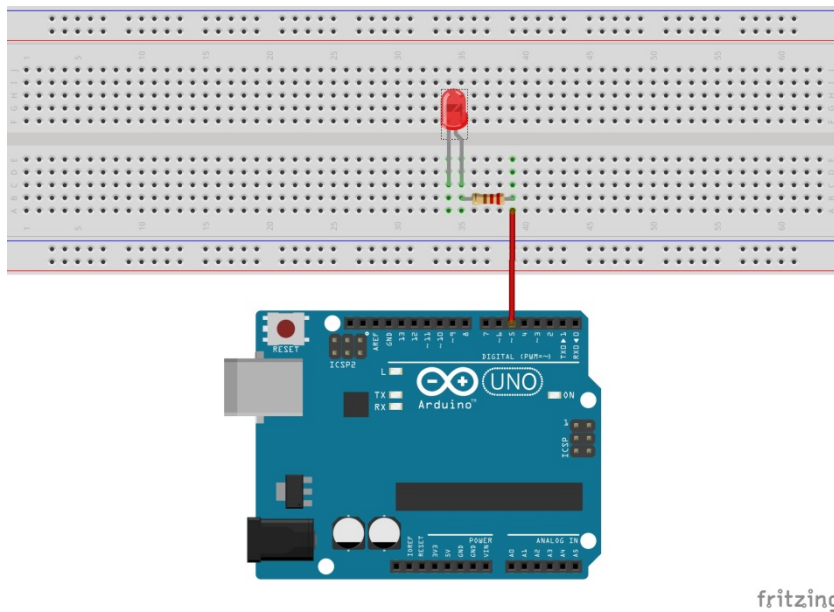
Συνδέουμε ένα καλώδιο από το pin 5 του Arduino σε μία ελεύθερη στήλη του breadboard.



fritzing

Βήμα 2

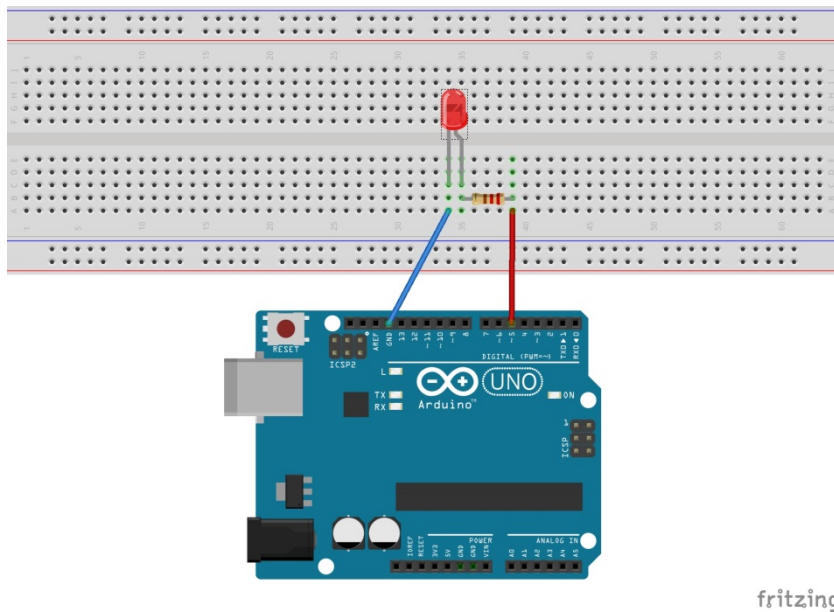
Συνδέουμε το ένα άκρο της αντίστασης στη στήλη όπου καταλήγει το καλώδιο και το άλλο σε μία κενή στήλη του breadboard.



fritzing

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.



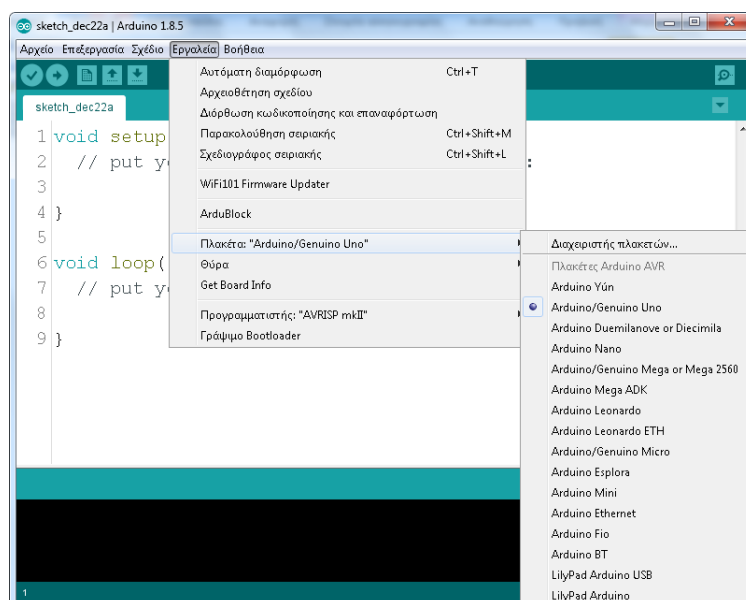
fritzing

Βήμα 4

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

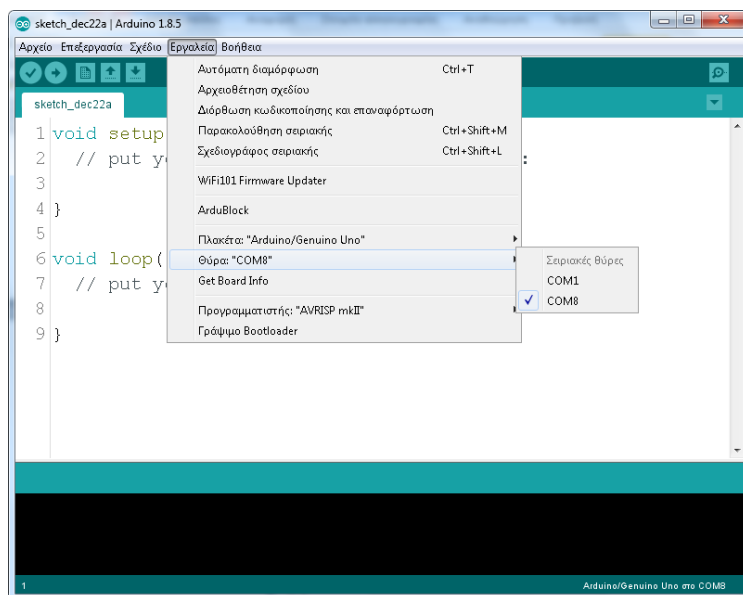
8.3 Σύνδεση πλακέτας με τον υπολογιστή

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 47).



Εικόνα 47

Ακολουθως, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino (Εικόνα 48).



Εικόνα 48

8.4 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 49

Στην Εικόνα 49 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 ArduBlock με το κουμπί Άνοιγμα.

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

9 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.

9.1 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

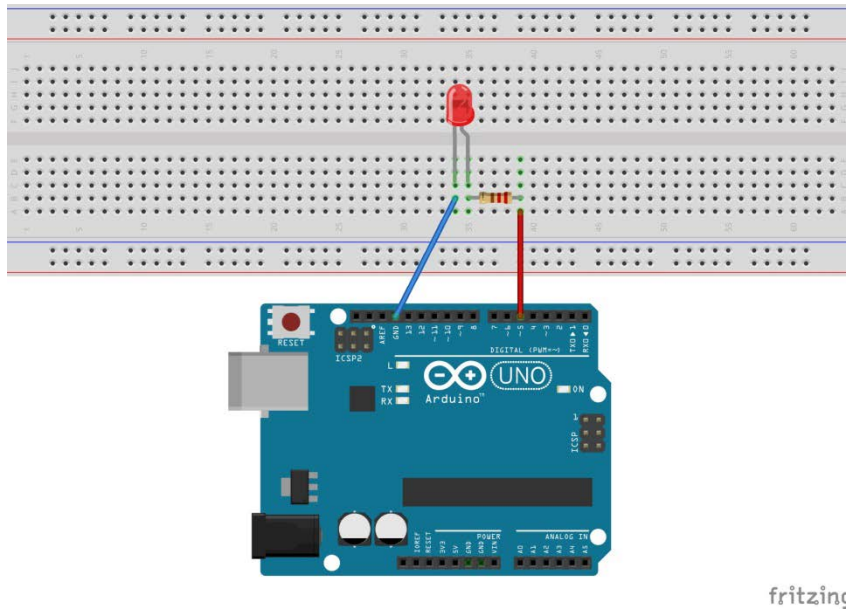
Για τη ρύθμιση της έντασης του LED θα χρησιμοποιήσουμε ένα ποτενσιόμετρο. Όπως φαίνεται στην Εικόνα 50, το περιστροφικό ποτενσιόμετρο διαθέτει έναν περιστρεφόμενο ρυθμιστή και 3 ακροδέκτες. Εσωτερικά, οι 2 ακραίοι ακροδέκτες συνδέονται στα άκρα μίας ωμικής αντίστασης. Ο μεσαίος ακροδέκτης συνδέεται με 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 50

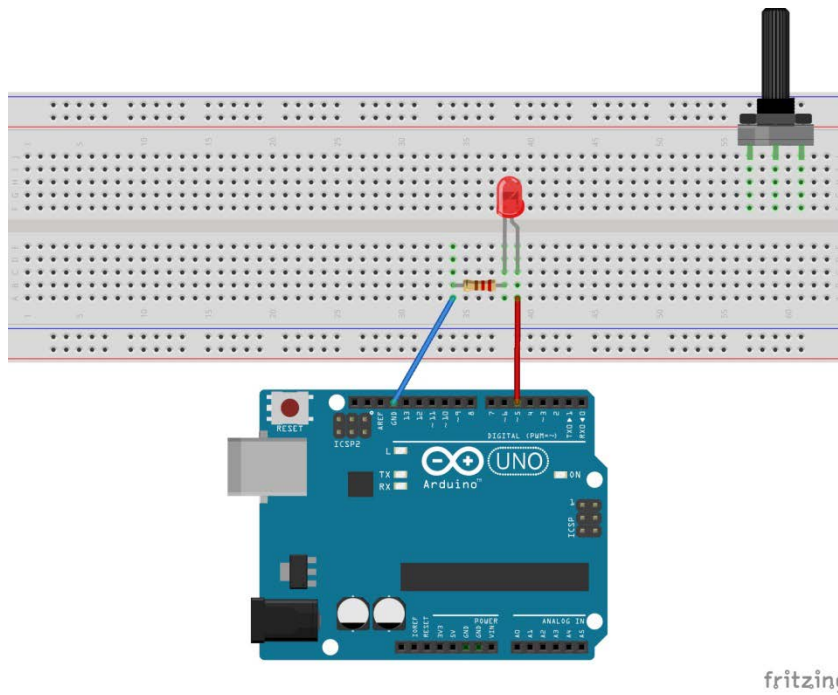
9.2 Κύκλωμα

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 51).



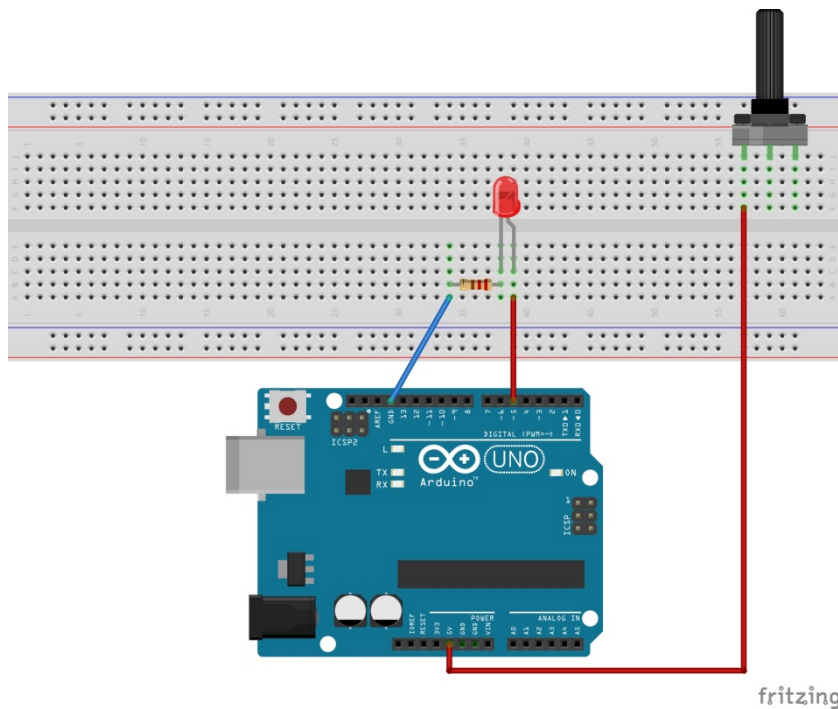
Εικόνα 51

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



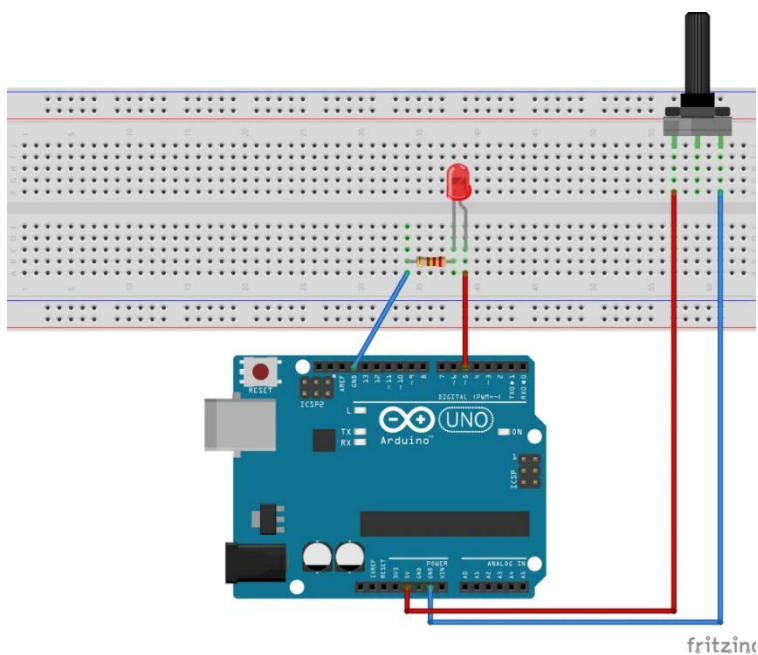
Βήμα 1

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



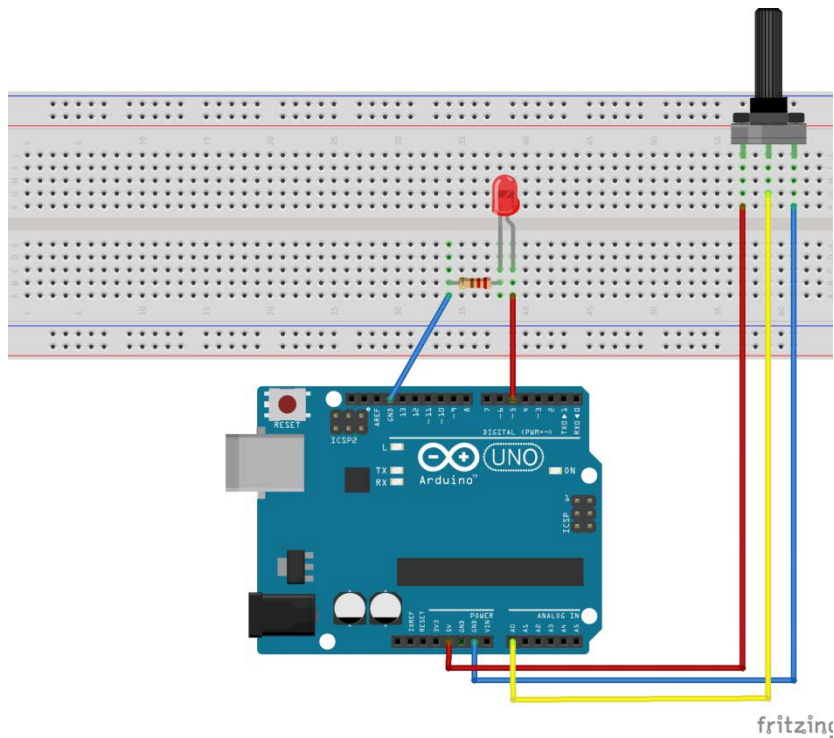
Βήμα 2

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

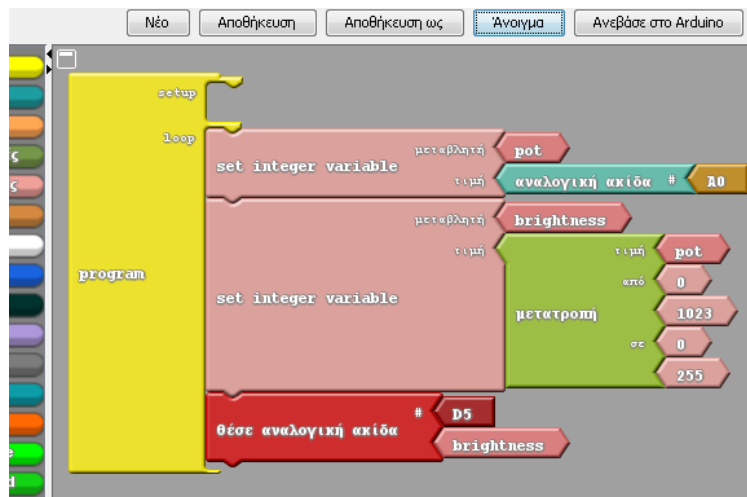
Τέλος, προκειμένου να διαβάσουμε την έξοδο του ποτενσιόμετρου, συνδέουμε με ένα καλώδιο τη στήλη του μεσαίου ακροδέκτη στην αναλογική είσοδο A0 του Arduino.

9.3 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.

9.4 Πρόγραμμα

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



Εικόνα 52

Στην Εικόνα 52 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.

Μπορείτε να βρείτε έτοιμο το πρόγραμμα στο συνοδευτικό αρχείο με όνομα *LED_w_Pot.abp* και να το ανοίξετε στο περιβάλλον του ArduBlock με το κουμπί *Άνοιγμα*.

Από το περιβάλλον του ArduBlock πατάμε το κουμπί *Ανέβασε στο Arduino*. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα και αρχίζει να εκτελείται. Περιστρέψτε το ποτενσιόμετρο για να αλλάξετε τη φωτεινότητα του LED.

10 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.

10.1 Υλικά

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

Φωτοαντίσταση

Η φωτοαντίσταση (Εικόνα 53) 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.



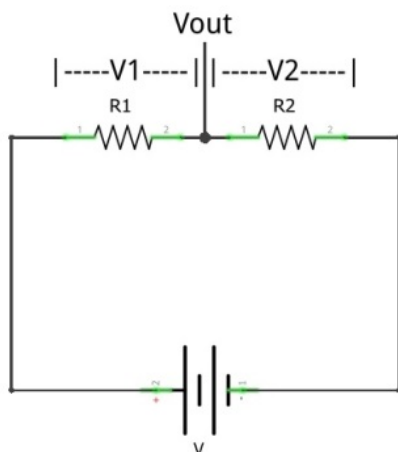
Εικόνα 53 Φωτοαντίσταση

Αντίσταση 10 kΩ

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

10.2 Κύκλωμα

Ένας διαιρέτης τάσης (Εικόνα 54) είναι μία διάταξη που αποτελείται από δύο αντιστάσεις R1 και R2, συνδεδεμένες σε σειρά.

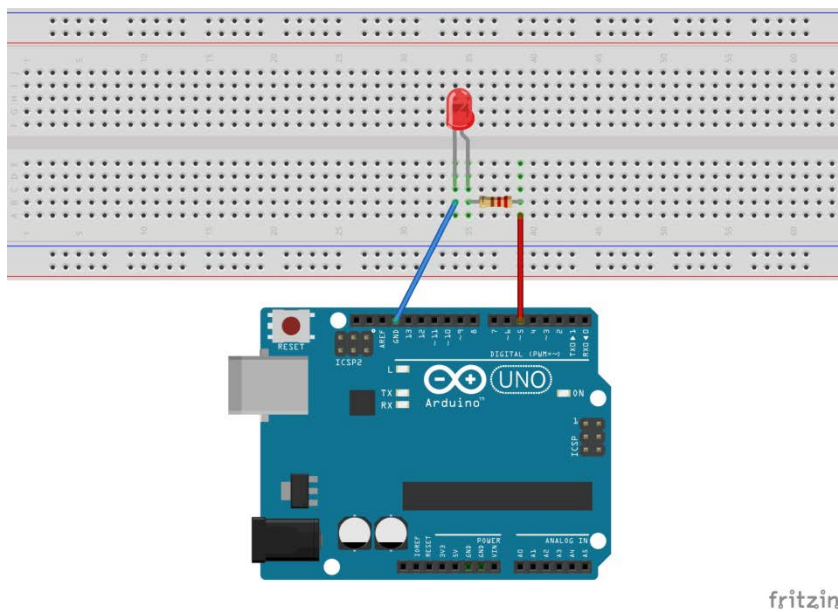


Εικόνα 54

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.

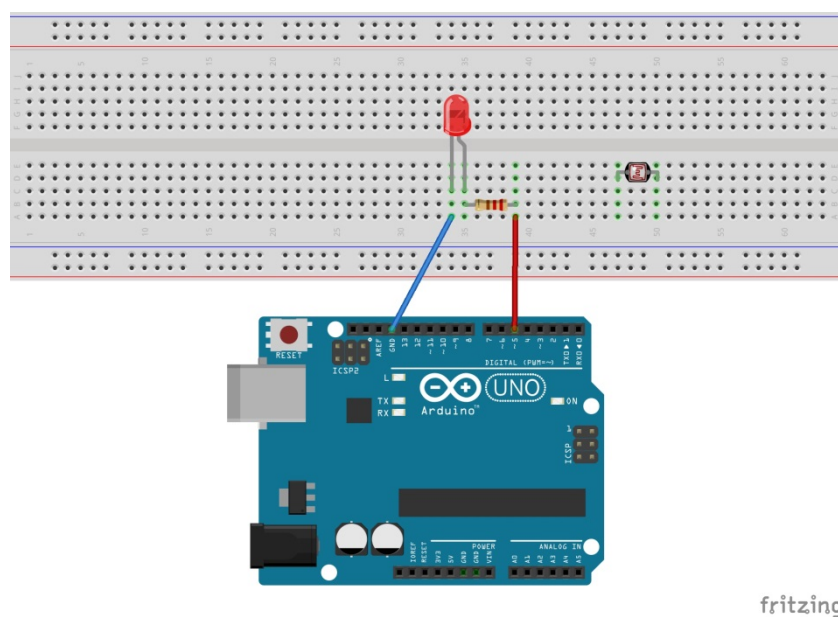
Για να φτιάξουμε έναν αισθητήρα φωτός, κατασκευάζουμε ένα διαιρέτη τάσης με μία φωτοαντίσταση και μία αντίσταση 10kΩ. Όταν αλλάζει η ένταση του φωτός, μεταβάλλεται η τιμή της φωτοαντίστασης, άρα αλλάζει ο λόγος των τιμών των δύο αντιστάσεων, αλλάζει η πτώση τάσης που συμβαίνει σε καθεμία από αυτές και 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 55).



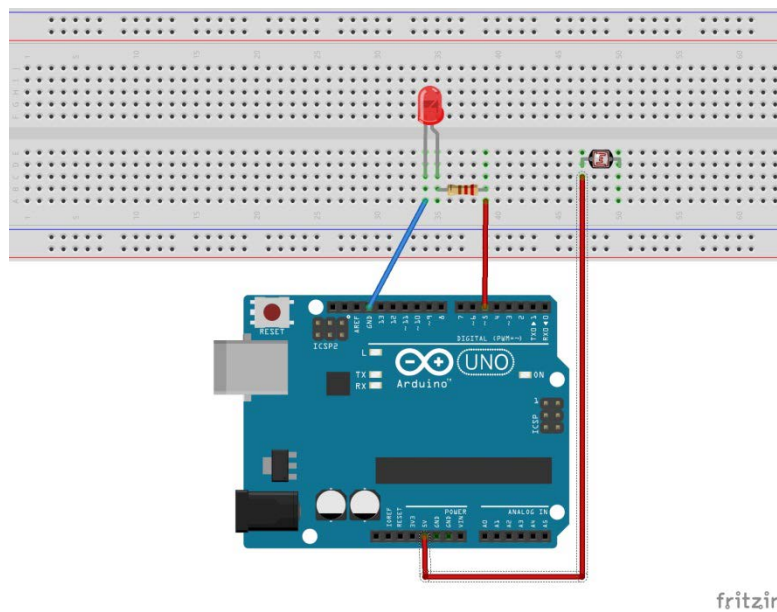
Εικόνα 55

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



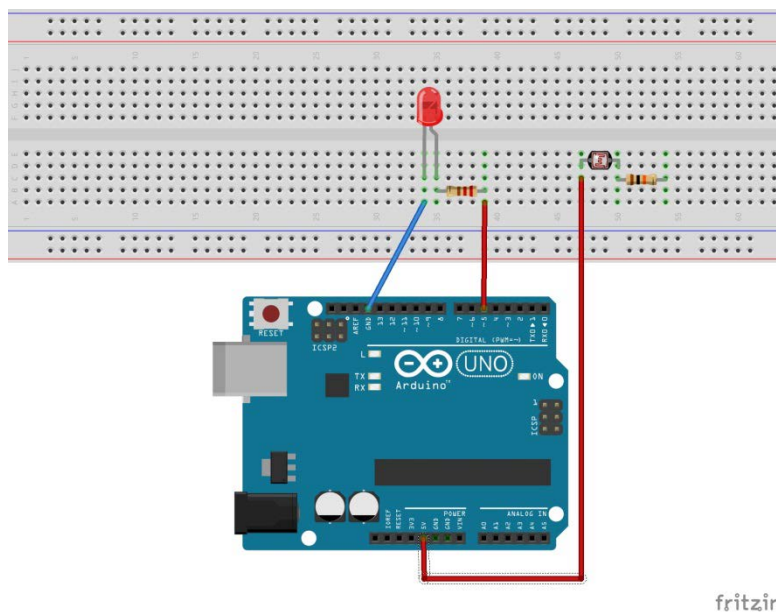
Βήμα 1

Τοποθετούμε τη φωτοαντίσταση πάνω στο breadboard.



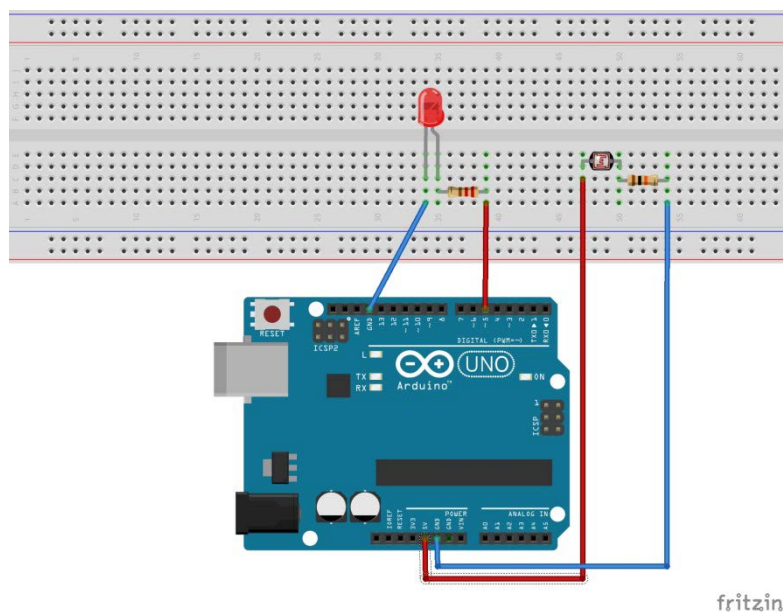
Βήμα 2

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



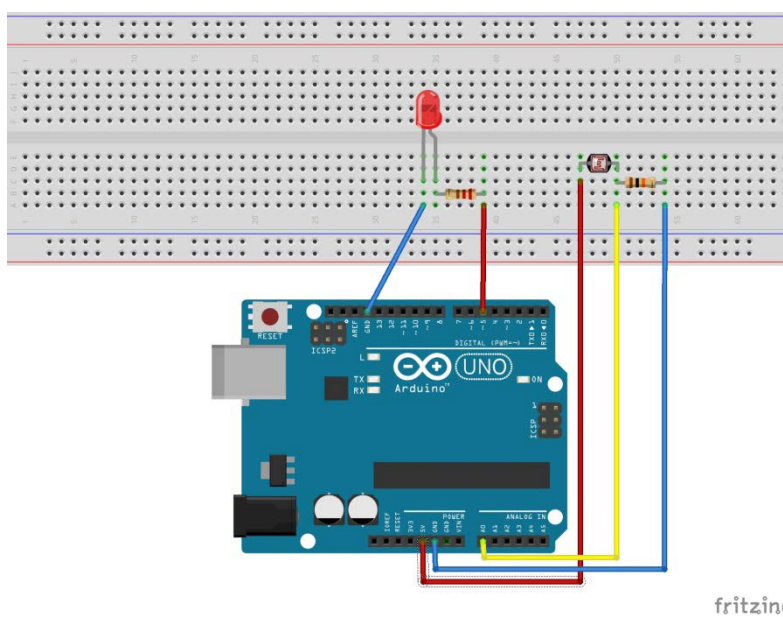
Βήμα 3

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

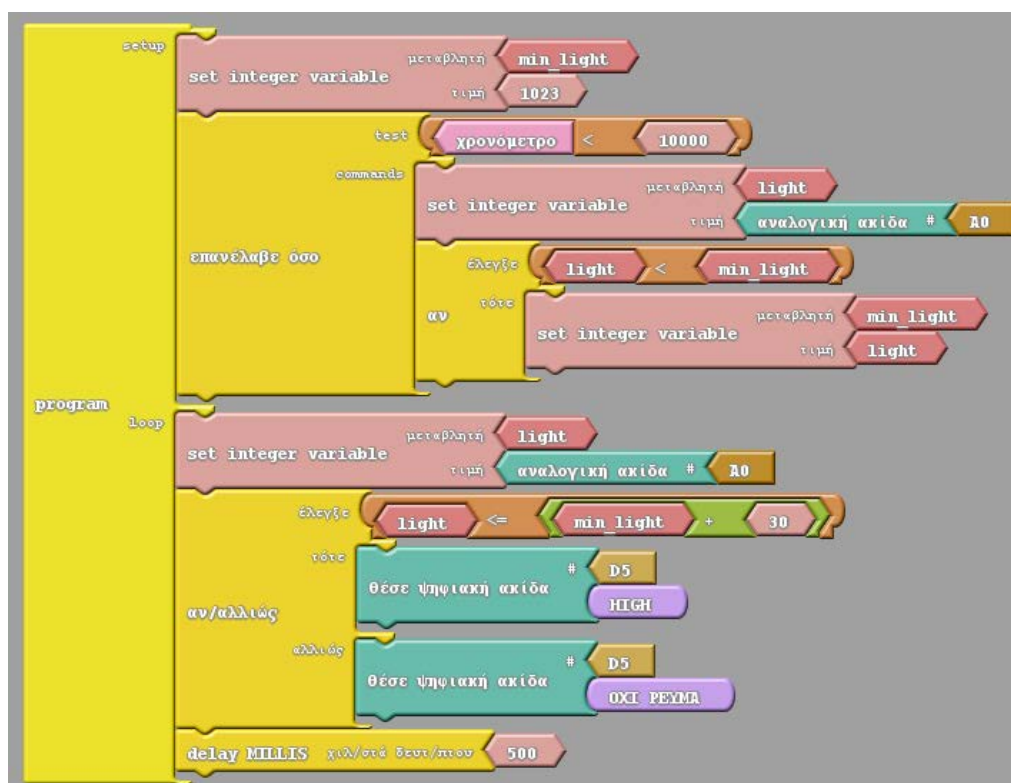
Τέλος, με ένα καλώδιο, συνδέουμε τη στήλη στην οποία ενώνονται οι δύο αντιστάσεις, με τον ακροδέκτη A0 του Arduino, προκειμένου να διαβάσουμε την τιμή του αισθητήρα φωτός.

10.3 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.

10.4 Πρόγραμμα

Από το μενού *Εργαλεία* του Arduino IDE επιλέγουμε *ArduBlock*. Στην Εικόνα 56 παρουσιάζεται ο κώδικας του προγράμματος και στη συνέχεια εξηγείται η λειτουργία του.



Εικόνα 56

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.

Μπορείτε να βρείτε έτοιμο το πρόγραμμα στο συνοδευτικό αρχείο με όνομα *NightLight.abp* και να το ανοίξετε στο ArduBlock με το κουμπί *Άνοιγμα*.

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.

11 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.

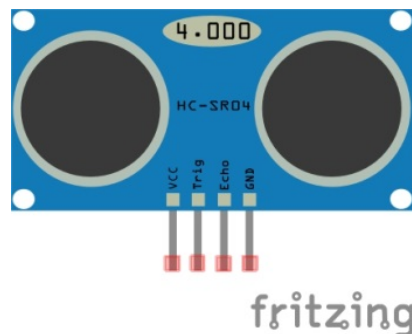
11.1 Υλικά

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 57, 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 57

Πώς λειτουργεί

Για να ξεκινήσει η διαδικασία της μέτρησης, πρέπει να στείλουμε στον ακροδέκτη Trig έναν παλμό High με διάρκεια τουλάχιστον 10 μsec . Μόλις ο αισθητήρας λάβει το σήμα ενεργοποίησης, στέλνει από τον πομπό μια ακολουθία υπερήχων. Όταν οι υπερήχοι συναντήσουν κάποιο εμπόδιο αντανακλώνονται 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 υπερήχοι 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:

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

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

Active Buzzer

Τα buzzer (Εικόνα 58) 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 58 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

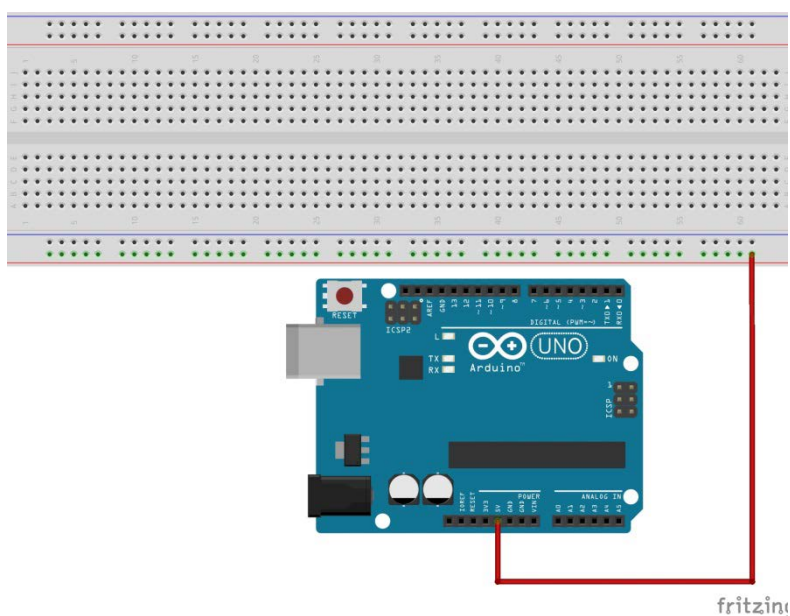
Η αντίσταση των 100 Ω θα συνδεθεί σε σειρά με το buzzer για τον περιορισμό του ρεύματος.



Εικόνα 59 Αντίσταση 100Ω

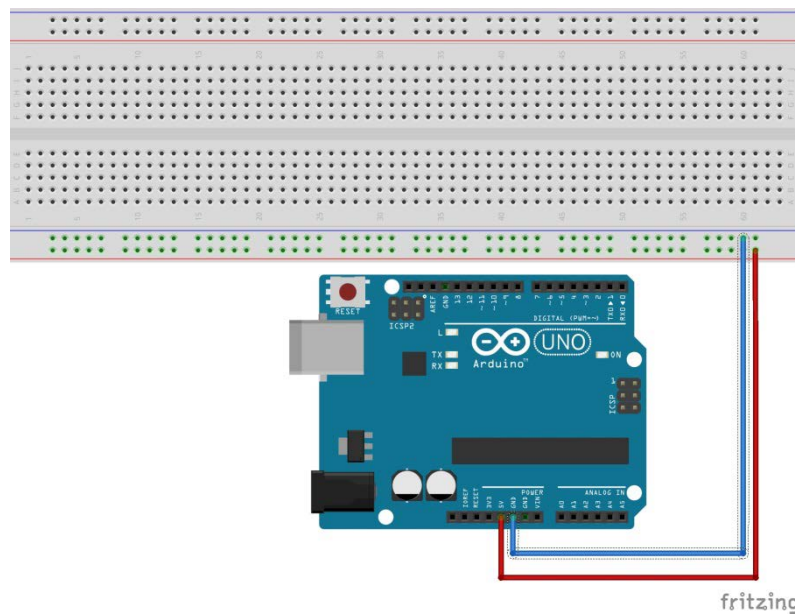
11.2 Κύκλωμα

Στη συνέχεια παρουσιάζεται βήμα προς βήμα η κατασκευή του κυκλώματος.



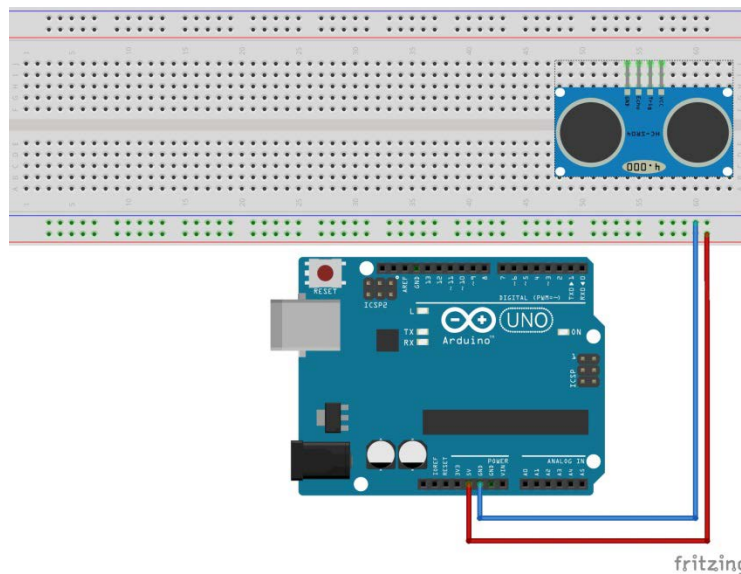
Βήμα 1

Συνδέουμε ένα καλώδιο από το pin 5V του Arduino στην πρώτη σπή 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 στήλη της οριζόντιας σειράς του breadboard με την μπλε γραμμή, ώστε να μπορούμε να πάρουμε γείωση από όλη τη σειρά.

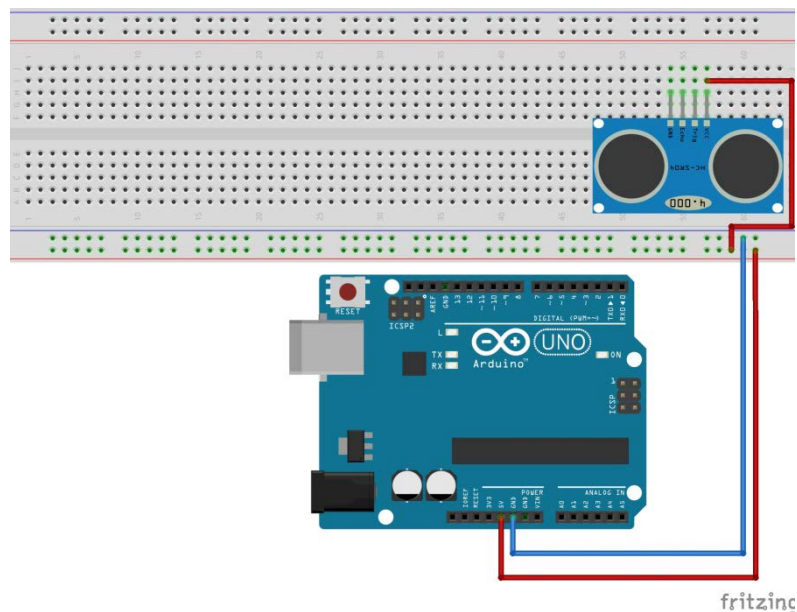


Βήμα 3

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

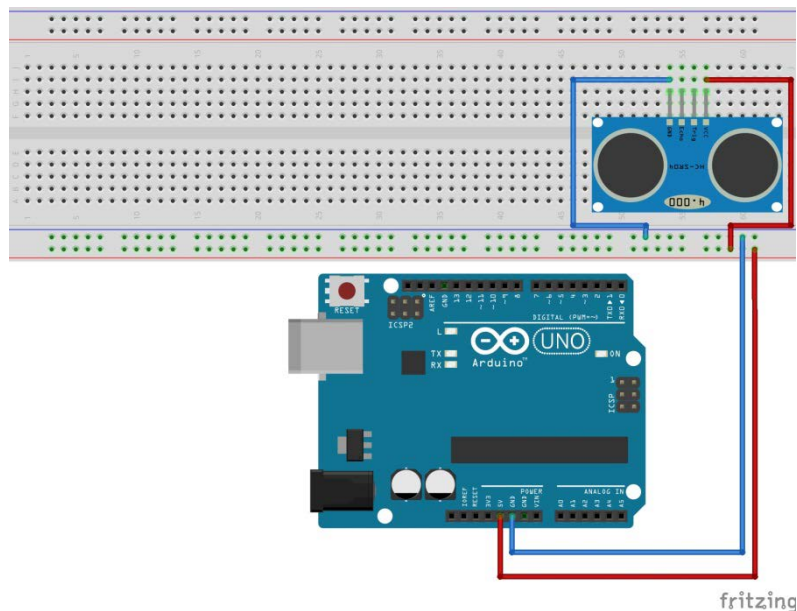
Προσοχή: Προκειμένου να είναι πιο ευδιάκριτες οι συνδέσεις των ακροδεκτών του αισθητήρα, στα επόμενα βήματα ο HC-SR04 παρουσιάζεται τοποθετημένος στη μέση της άνω πλευράς του breadboard και οι συνδέσεις γίνονται μπροστά από τον αισθητήρα.

Στο πραγματικό κύκλωμα τοποθετήστε τον HC-SR04 όπως φαίνεται στο Βήμα 3 και κάντε τις συνδέσεις πίσω από τους αντίστοιχους ακροδέκτες.



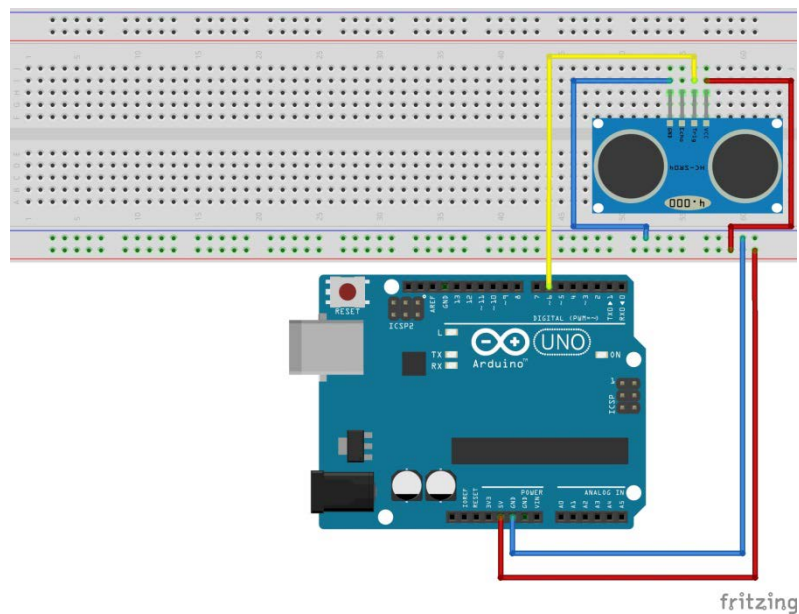
Βήμα 4

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



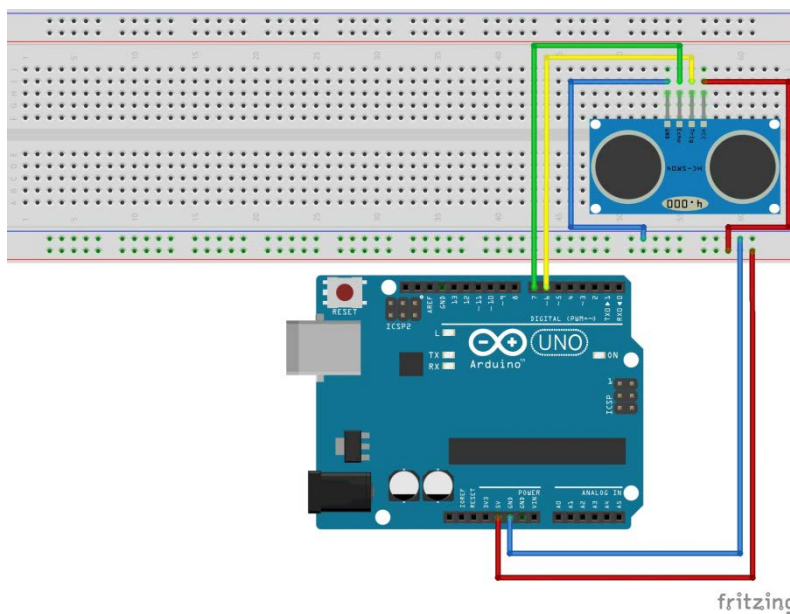
Βήμα 5

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



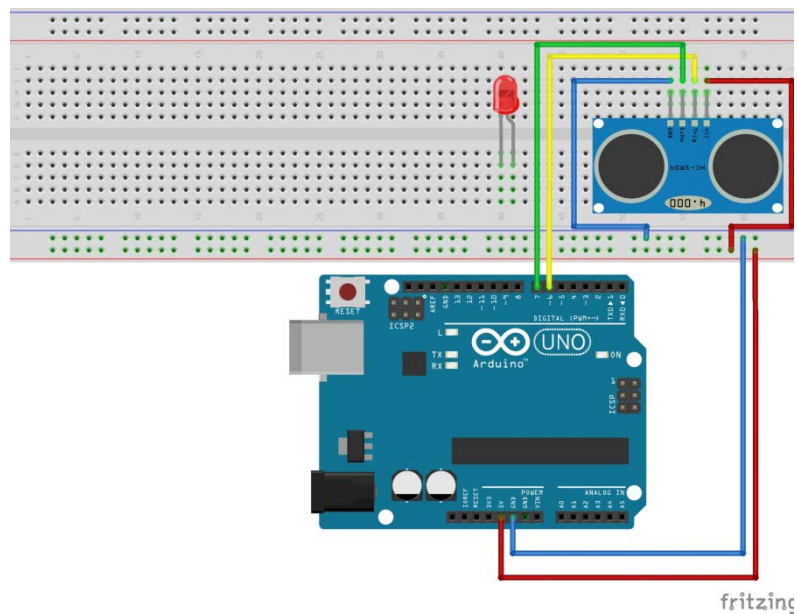
Βήμα 6

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



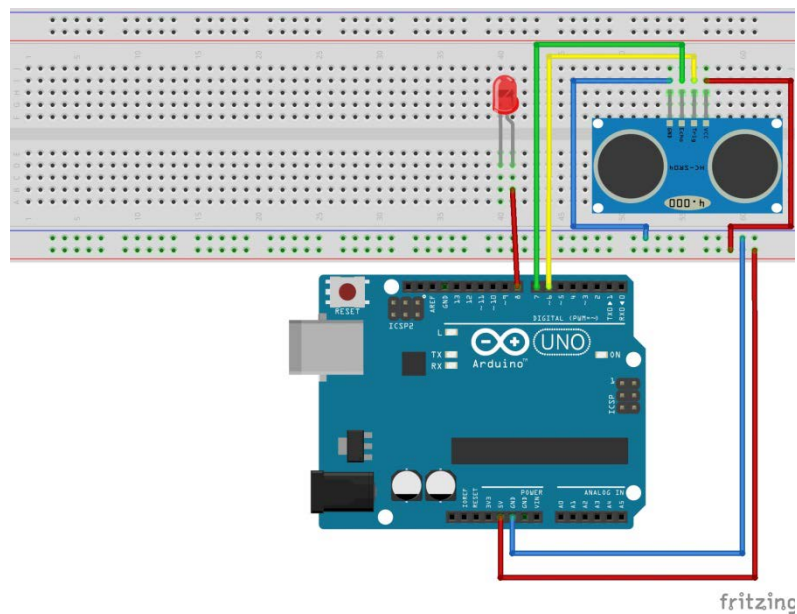
Βήμα 7

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



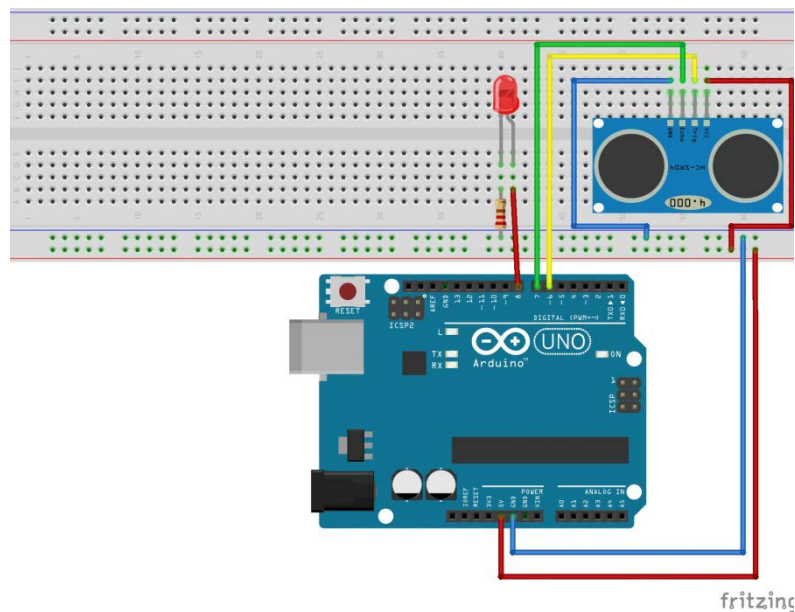
Βήμα 8

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



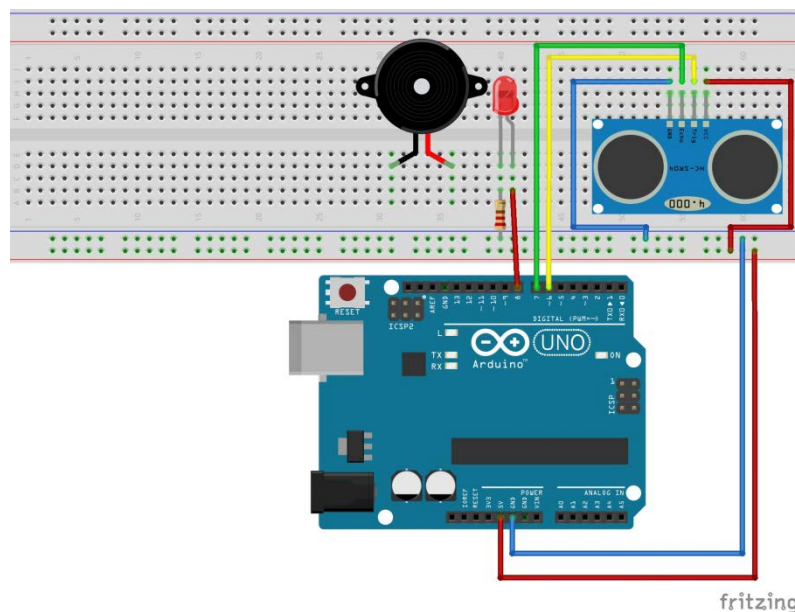
Βήμα 9

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



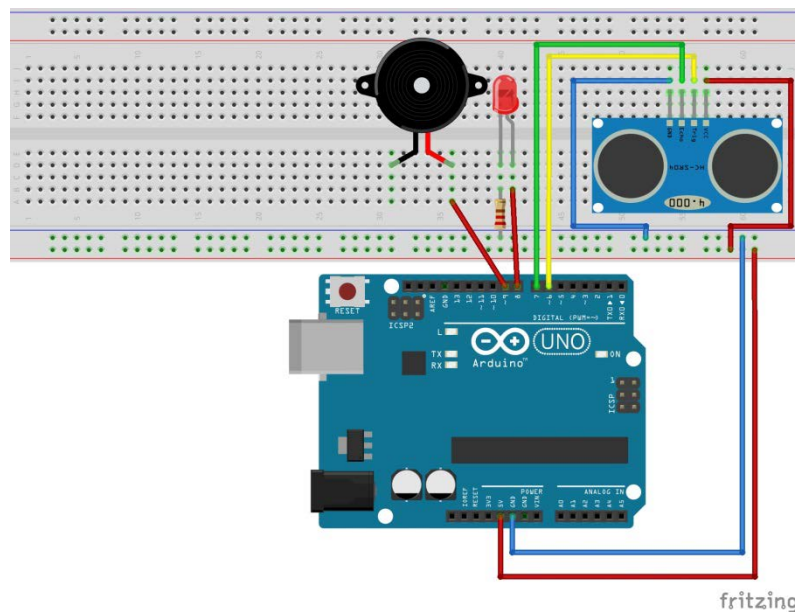
Βήμα 10

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.



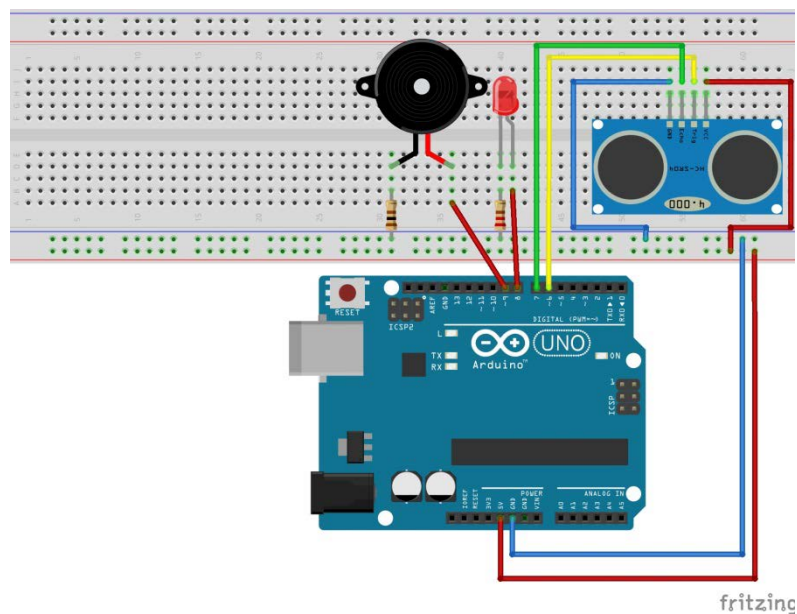
Βήμα 11

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

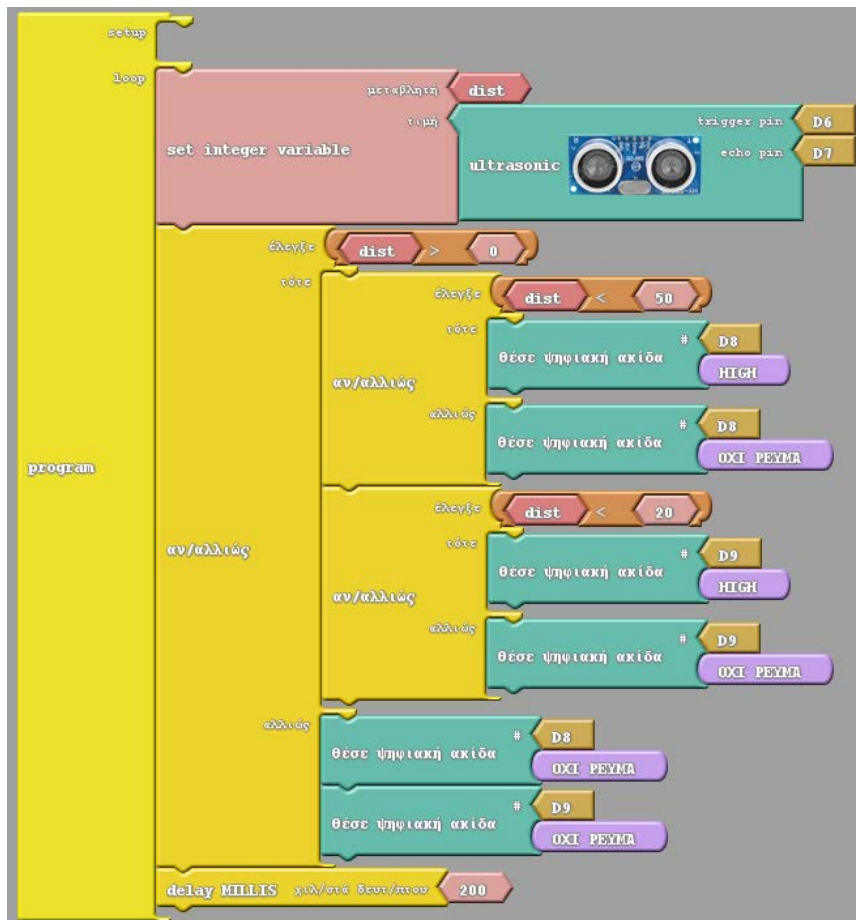
Συνδέουμε το ένα άκρο της αντίστασης των 100 Ω στην ίδια στήλη με τον αρνητικό ακροδέκτη του buzzer και το άλλο της άκρο στη σειρά του breadboard όπου συνδέεται το GND του Arduino. Αν δεν υπάρχει διαθέσιμη αντίσταση 100 Ω, χρησιμοποιήστε ένα καλώδιο για να συνδέσετε τον αρνητικό ακροδέκτη του buzzer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

11.3 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.

11.4 Πρόγραμμα

Από το μενού *Εργαλεία* του Arduino IDE επιλέγουμε *ArduBlock*. Στην Εικόνα 60 παρουσιάζεται ο κώδικας του προγράμματος και στη συνέχεια εξηγείται η λειτουργία του.



Εικόνα 60

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.

Το πρόγραμμα για τη λειτουργία της εφαρμογής υπάρχει έτοιμο στο συνοδευτικό αρχείο με όνομα *DistanceSensor.abp*. Για να το ανοίξετε, μέσα από το ArduBlock πατήστε το κουμπί *Άνοιγμα*, εντοπίστε το αρχείο στη θέση όπου το έχετε αποθηκεύσει και πατήστε *Open*.

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 απομακρύνετε το από τον αισθητήρα, για να προκαλέσετε την ενεργοποίηση και απενεργοποίηση του LED και του buzzer.

Γ' Γυμνασίου

12 Σχεδιασμός κατασκευής ρομποτικού οχήματος

Στο τελευταίο κομμάτι του οδηγού, θα κατασκευάσουμε ένα ρομποτικό όχημα, το οποίο θα κινείται αυτόνομα και θα έχει τη δυνατότητα να ανιχνεύει εμπόδια.

Σε αυτή την ενότητα, παρουσιάζονται τα διάφορα υλικά που θα χρησιμοποιηθούν για την κατασκευή του οχήματος και γίνεται μία αρχική ανάλυση του τρόπου λειτουργίας του.

12.1 Εξαρτήματα

Σκελετός

Ο σκελετός του οχήματος περιλαμβάνει:

- μία βάση πάνω στην οποία θα τοποθετηθούν όλα τα εξαρτήματα
- στηρίγματα για την προσάρτηση 2 μοτέρ (ένα δεξιά και ένα αριστερά στη βάση)
- 2 ρόδες, που θα τοποθετηθούν πάνω στα μοτέρ και θα αποτελούν τους κινητήριους τροχούς του οχήματος
- μία ρόδα ελεύθερης περιστροφής, που θα αποτελεί την πίσω ρόδα του οχήματος
- μία μπαταριοθήκη για 4 μπαταρίες AA 1.5V, που θα παρέχει την απαιτούμενη τροφοδοσία
- έναν διακόπτη on/off για έλεγχο της τροφοδοσίας
- βίδες, παξιμάδια και αποστάτες για τη συναρμολόγηση

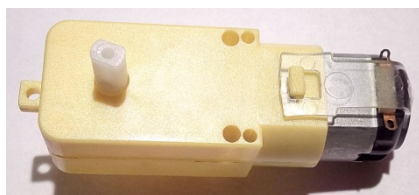
Μοτέρ dc

Ένα μοτέρ dc είναι ένα μοτέρ, το οποίο όταν τροφοδοτείται με συνεχή τάση, περιστρέφεται συνέχεια με μία συγκεκριμένη φορά. Για να αλλάξει η φορά περιστροφής του μοτέρ, πρέπει να αλλάξει η πολικότητα της τάσης με την οποία τροφοδοτείται.

Κάθε dc μοτέρ έχει ένα συγκεκριμένο εύρος τάσης λειτουργίας. Η ταχύτητα περιστροφής του μοτέρ μπορεί να μεταβάλλεται, αυξομειώνοντας την τάση τροφοδοσίας του (υψηλότερη τάση → μεγαλύτερη ταχύτητα περιστροφής), πάντα μέσα στο αποδεκτό εύρος τιμών.

Τα μοτέρ dc έχουν υψηλές απαιτήσεις σε ρεύμα, που δεν μπορούν να καλυφθούν από τους ακροδέκτες του Arduino. Για το λόγο αυτό είναι απαραίτητη η ύπαρξη εξωτερικής τροφοδοσίας.

Για την κίνηση του οχήματος που θα κατασκευάσουμε, θα χρησιμοποιηθούν 2 μοτέρ dc (Εικόνα 61) με ρόδες, τα οποία θα τοποθετηθούν στις δύο πλευρές της βάσης. Μέσα στο πλαστικό περίβλημα των μοτέρ, υπάρχουν γρανάζια για τη μετάδοση της κίνησης στον πλαστικό άξονα στο πλάι.



Εικόνα 61

Κύκλωμα οδήγησης μοτέρ

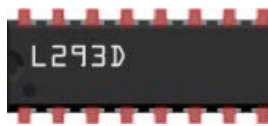
Όπως αναφέρθηκε προηγούμενα, για τη λειτουργία ενός dc μοτέρ απαιτείται η χρήση εξωτερικής τροφοδοσίας. Συνεπώς, για να ελέγξουμε τη λειτουργία του μοτέρ με το Arduino, θα πρέπει με κάποιο τρόπο να ρυθμίζουμε την παροχή τάσης από την εξωτερική πηγή προς το μοτέρ. Αυτό επιτυγχάνεται με τη χρήση ειδικών κυκλωμάτων που ονομάζονται κυκλώματα οδήγησης μοτέρ.

Το H-Bridge είναι ένας τύπος κυκλώματος οδήγησης, που μας επιτρέπει να ελέγχουμε με λογικά σήματα (π.χ. ακροδέκτες εξόδου του Arduino) τόσο την ταχύτητα, όσο και τη φορά περιστροφής ενός DC μοτέρ.

Στη συνέχεια παρουσιάζονται δύο εναλλακτικές λύσεις, για την οδήγηση των μοτέρ του ρομποτικού μας οχήματος .

L293D chip

Το L293D (Εικόνα 62) είναι ένα ολοκληρωμένο κύκλωμα που επιτρέπει τον ταυτόχρονο έλεγχο 2 μοτέρ DC.



Εικόνα 62

Στην Εικόνα 63 παρουσιάζονται οι ακροδέκτες του L293D και στη συνέχεια εξηγείται η λειτουργία τους.



Εικόνα 63

Στους ακροδέκτες Enable 1, In 1 και In 2 (1,2 και 7) συνδέονται τα λογικά σήματα για τον έλεγχο του πρώτου μοτέρ (έστω M1). Συγκεκριμένα, με το σήμα που συνδέεται στον ακροδέκτη Enable 1, μπορούμε να ενεργοποιούμε (κατάσταση HIGH) και να απενεργοποιούμε (κατάσταση LOW) το M1. Αν στον Enable 1 συνδέσουμε ένα σήμα PWM (βλέπε 8 Εφαρμογή 5: LED με μεταβαλλόμενη ένταση) τότε μπορούμε να ελέγχουμε την ταχύτητα περιστροφής του μοτέρ.

Τα σήματα στους ακροδέκτες In 1 και In 2 ελέγχουν την περιστροφή του M1, σύμφωνα με τον παρακάτω πίνακα

In 1	In 2	M1
HIGH	LOW	Περисτροφή με τη φορά του ρολογιού
LOW	HIGH	Περисτροφή αντίθετα από τη φορά του ρολογιού
LOW	LOW	Σταμάτημα

HIGH	HIGH	Σταμάτημα
------	------	-----------

Στους ακροδέκτες Out 1 και Out 2 (3 και 6) συνδέονται οι 2 επαφές του μοτέρ M1.

Παρόμοια, στους ακροδέκτες Enable 2, In 3 και In 4 (9, 10 και 15) συνδέονται τα σήματα για τον έλεγχο του δεύτερου μοτέρ (έστω M2), ενώ στους ακροδέκτες Out 3 και Out 4 (11 και 14) συνδέεται το μοτέρ (M2). Ο σχετικός πίνακας για τη λειτουργία του M2 είναι ο εξής:

In 3	In 4	M2
HIGH	LOW	Περιστροφή με τη φορά του ρολογιού
LOW	HIGH	Περιστροφή αντίθετα από τη φορά του ρολογιού
LOW	LOW	Σταμάτημα
HIGH	HIGH	Σταμάτημα

Στον ακροδέκτη +Vmotor (8), συνδέεται ο θετικός πόλος της εξωτερικής πηγής τροφοδοσίας για την τροφοδοσία των μοτέρ.

Στον ακροδέκτη +V (16) συνδέεται τάση 5V (pin 5V του Arduino) για τη μετάφραση των λογικών σημάτων.

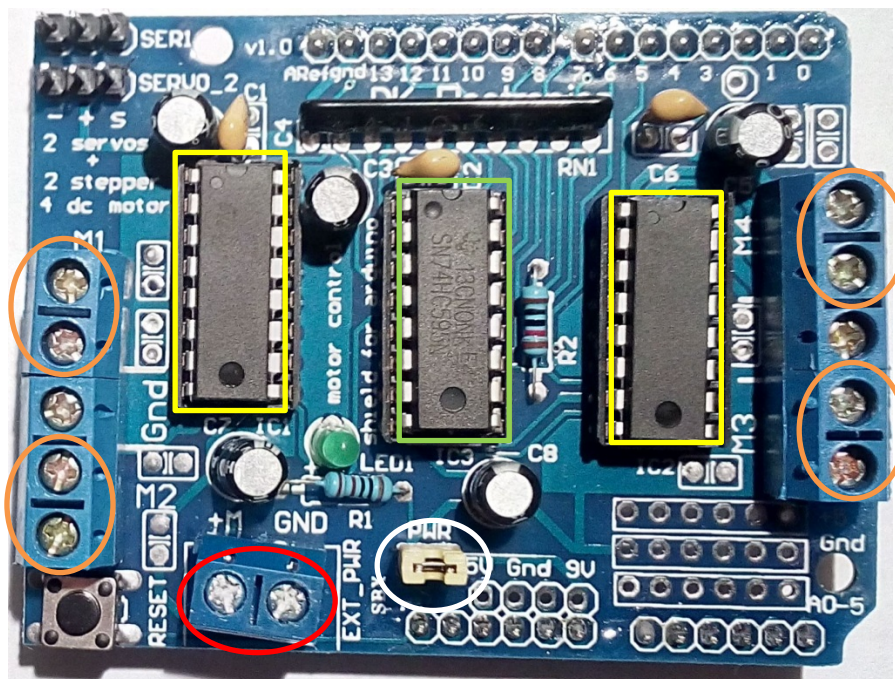
Οι 4 ακροδέκτες 0V (4, 5, 12 και 13) είναι συνδεδεμένοι μεταξύ τους και σε έναν από όλους συνδέεται η γείωση της πηγής τροφοδοσίας. Επειδή η γείωση της τροφοδοσία των μοτέρ και η γείωση των λογικών σημάτων ελέγχου πρέπει να είναι κοινή, αν το Arduino τροφοδοτείται από διαφορετική πηγή, τότε πρέπει να συνδέσουμε και το pin GND του Arduino σε έναν από αυτούς τους ακροδέκτες.

L293D motor shield

Το L293D motor shield (Εικόνα 64) είναι μία πλακέτα επέκτασης, που κουμπώνει πάνω στο Arduino και επιτρέπει μεταξύ άλλων τον ταυτόχρονο έλεγχο 4 μοτέρ dc. Όπως φαίνεται και στην εικόνα, διαθέτει ενσωματωμένα 2 chip L293D (κίτρινο πλαίσιο) και ένα SN74HC595N (πράσινο πλαίσιο).

Οι ακροδέκτες του Arduino που δεσμεύονται με τη χρήση του motor shield είναι οι 3, 4, 5, 6, 7, 8, 9, 10, 11 και 12. Με δεδομένο ότι οι ακροδέκτες 0 και 1 χρησιμοποιούνται

για τη σειριακή επικοινωνία, ουσιαστικά οι ακροδέκτες που μένουν διαθέσιμοι είναι ο 2, ο 13 και οι A0 έως A5.



Εικόνα 64

Τα καλώδια των τεσσάρων μοτέρ συνδέονται στα ζευγάρια επαφών M1, M2, M3 και M4 (πορτοκαλί πλαίσια στην Εικόνα 64), ενώ η εξωτερική πηγή τροφοδοσίας συνδέεται στις επαφές με τη σήμανση EXT_PWR (κόκκινο πλαίσιο). Ιδιαίτερη προσοχή πρέπει να δίνεται στην πολικότητα της τροφοδοσίας. Συγκεκριμένα, ο θετικός πόλος της πηγής συνδέεται στην επαφή +M (αριστερή στην εικόνα) και ο αρνητικό πόλος της πηγής στην επαφή GND (δεξιά στην εικόνα). Όταν το jumper PWR (λευκό πλαίσιο) είναι τοποθετημένο, η σύνδεση εξωτερικής τροφοδοσίας στις επαφές EXT_PWR, τροφοδοτεί και το Arduino.

Αισθητήρας HC-SR04

Ο HC-SR04 είναι ένας αισθητήρας υπερήχων, που χρησιμοποιείται για τη μέτρηση της απόστασης από ένα αντικείμενο. Στο όχημά μας, θα χρησιμοποιήσουμε τον αισθητήρα για να ανιχνεύουμε πιθανά εμπόδια στην πορεία. Για περισσότερες πληροφορίες σχετικά με τον αισθητήρα HC-SR04 και τη λειτουργία του, ανατρέξτε στο 11 Εφαρμογή 8: Αισθητήρας υπερήχων HC-SR04.

Arduino UNO

Το Arduino UNO, με το πρόγραμμα που θα εκτελείται σε αυτό, θα αποτελέσει τον εγκέφαλο του ρομποτικού μας οχήματος. Παίρνοντας μετρήσεις από τον αισθητήρα υπερήχων και υπολογίζοντας την απόσταση από πιθανά εμπόδια, θα στέλνει τα κατάλληλα σήματα ελέγχου στο κύκλωμα οδήγησης των 2 μοτέρ, καθορίζοντας την πορεία του οχήματος.

Breadboard

Το breadboard θα στερεωθεί στην μπροστά πλευρά του οχήματος και πάνω του θα τοποθετηθεί ο αισθητήρας απόστασης HC-SR04, ο οποίος θα συνδεθεί με το Arduino. Αν ως κύκλωμα για την οδήγηση των μοτέρ επιλεγεί το chip L293D, τότε θα τοποθετηθεί και αυτό πάνω στο breadboard, προκειμένου να γίνουν οι απαιτούμενες συνδέσεις του με τα μοτέρ, την πηγή και το Arduino.

12.2 Τρόπος κίνησης του οχήματος

Όπως αναφέρθηκε και προηγούμενα, για την κίνηση του οχήματος θα χρησιμοποιηθούν 2 ανεξάρτητα μοτέρ dc, ένα στη δεξιά πλευρά και ένα στην αριστερή. Είναι προφανές ότι, για να κινηθεί το όχημα προς τα εμπρός ή προς τα πίσω, θα πρέπει οι 2 κινητήριои τροχοί (τα αντίστοιχα μοτέρ) να κινηθούν ταυτόχρονα προς τα εμπρός ή προς τα πίσω.

Ωστόσο, από τη στιγμή που οι τροχοί είναι πάντα παράλληλοι στο σώμα του οχήματος, το θέμα που προκύπτει είναι το πώς θα μπορεί το όχημα να αλλάζει κατεύθυνση, δηλαδή να στρίβει δεξιά και αριστερά. Ο μοναδικός τρόπος για να γίνει αυτό, είναι να περιστρέφονται οι δύο τροχοί με διαφορετική ταχύτητα μεταξύ τους. Αν για παράδειγμα ο δεξιός τροχός κινηθεί γρηγορότερα από τον αριστερό, το αποτέλεσμα θα είναι το όχημα να στρίψει προς τα αριστερά. Μάλιστα, όσο μεγαλύτερη είναι η διαφορά στις ταχύτητες περιστροφής των δύο τροχών, τόσο πιο κλειστή θα είναι η στροφή. Η πιο απότομη στροφή που μπορεί να πραγματοποιηθεί με αυτόν τον τρόπο, είναι η επιτόπια περιστροφή και επιτυγχάνεται όταν ο ένας τροχός κινείται προς τα εμπρός και ο άλλος προς τα πίσω.

Επειδή η επιτόπια περιστροφή δίνει τη βέλτιστη δυνατότητα ελιγμού σε περιορισμένο χώρο, είναι η καταλληλότερη για το ρομποτικό μας όχημα. Συνεπώς, η αλλαγή κατεύθυνσης θα υλοποιηθεί ως εξής

- **στροφή αριστερά:** περιστροφή δεξιού τροχού προς τα εμπρός και αριστερού προς τα πίσω
- **στροφή δεξιά:** περιστροφή αριστερού τροχού προς τα εμπρός και δεξιού προς τα πίσω

12.3 Αρχική περιγραφή λειτουργίας προγράμματος

Όπως γίνεται αντιληπτό, η λειτουργία του ρομποτικού οχήματος θα καθορίζεται πλήρως από τις εντολές που θα εκτελούνται στο Arduino. Το πρώτο λοιπόν βήμα για την ανάπτυξη του αντίστοιχου προγράμματος, είναι να περιγράψουμε με απλά λόγια την επιθυμητή συμπεριφορά του οχήματος:

Το όχημα θα ελέγχει αν υπάρχει εμπόδιο μπροστά του. Αν η απόσταση από το κοντινότερο εμπόδιο είναι μεγαλύτερη από 25 εκατοστά, θα κινείται προς τα εμπρός. Σε διαφορετική περίπτωση θα εκτελεί έναν ελιγμό για να αποφύγει το εμπόδιο. Η διαδικασία αυτή θα επαναλαμβάνεται όσο υπάρχει τροφοδοσία.

Στη συνέχεια, επιχειρούμε να διατυπώσουμε αυτή την περιγραφή με τη μορφή ενός αλγορίθμου σε υψηλό επίπεδο, χωρίς να αναλύουμε τις επιμέρους λειτουργίες:

Για πάντα

Αν απόσταση_από_εμπόδιο > 25 **τότε**

 κινήσου_μπροστά

Αλλιώς

 εκτέλεσε_ελιγμό_αποφυγής

Τέλος_αν

Τέλος_επανάληψης

13. Συναρμολόγηση οχήματος

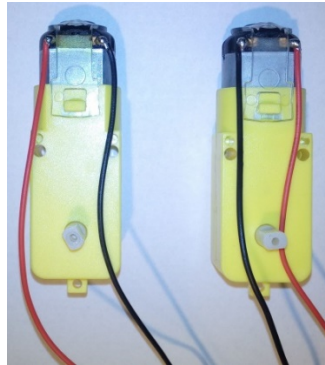
Στη συνέχεια περιγράφεται η διαδικασία της συναρμολόγησης του ρομποτικού οχήματος.

13.1 Προετοιμασία εξαρτημάτων

Πριν να ξεκινήσουμε τη συναρμολόγηση του σκελετού του οχήματος, θα πρέπει να γίνουν προεργασίες σε κάποια από τα εξαρτήματα.

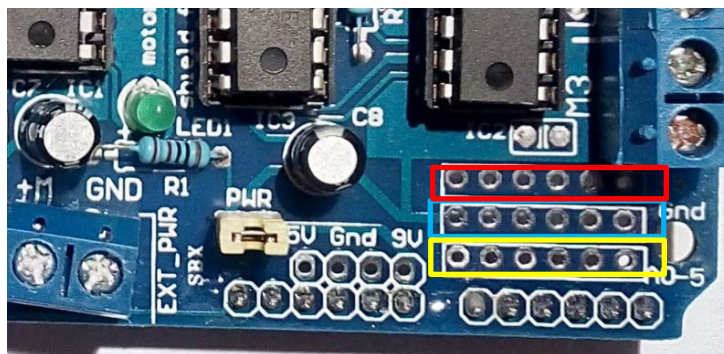
Η πρώτη εργασία που θα πρέπει να γίνει είναι η κόλληση των καλωδίων που παρέχονται, στις επαφές των 2 μοτέρ dc. Όπως φαίνεται και στην Εικόνα 65, με τις

επαφές των μοτέρ στραμμένες προς το μέρος μας, τα καλώδια (κόκκινο και μαύρο) έχουν κολληθεί σε διαφορετική πλευρά σε κάθε μοτέρ.



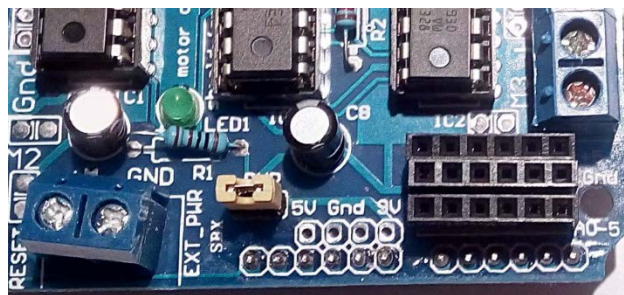
Εικόνα 65

Επιπλέον, αν για την οδήγηση των μοτέρ επιλεγθεί να χρησιμοποιηθεί το motor shield (προτεινόμενος τρόπος), θα πρέπει να κολληθούν σε αυτό τα 3 female headers. Επειδή, όπως αναφέρθηκε παραπάνω, οι περισσότεροι ακροδέκτες του Arduino δεσμεύονται όταν τοποθετούμε πάνω του το motor shield, θα πρέπει η σύνδεση του αισθητήρα HC-SR04 να γίνει σε δύο από τους ακροδέκτες αναλογικής εισόδου (A0-A5).



Εικόνα 66

Στην Εικόνα 66, στο κίτρινο πλαίσιο μπορείτε να δείτε τις αναμονές μέσω των οποίων είναι διαθέσιμοι οι ακροδέκτες A0-A5. Οι δύο άλλες σειρές αναμονών χρησιμοποιούνται για γείωση (γαλάζιο πλαίσιο) και για τροφοδοσία 5V (κόκκινο πλαίσιο). Σε αυτές τις 3 σειρές κολλάμε τα female header, όπως φαίνεται στην εικόνα 67.



Εικόνα 67

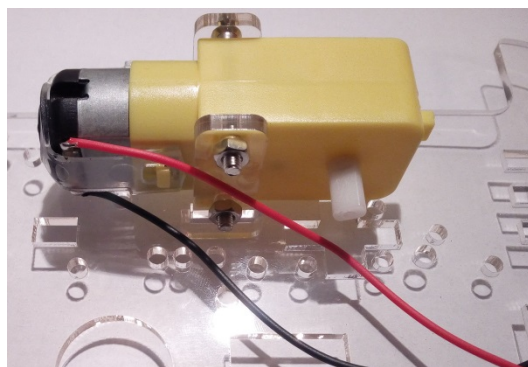
13.2 Τοποθέτηση μοτέρ

Περνάμε τα 2 πρώτα στηρίγματα μέσα από τις εγκοπές που υπάρχουν στη βάση και στη συνέχεια γυρνάμε τη βάση ανάποδα, όπως φαίνεται στην Εικόνα 68. Όπως μπορείτε να δείτε, έχουμε αφαιρέσει τα προστατευτικά αυτοκόλλητα τόσο από τη βάση όσο και από τα στηρίγματα.



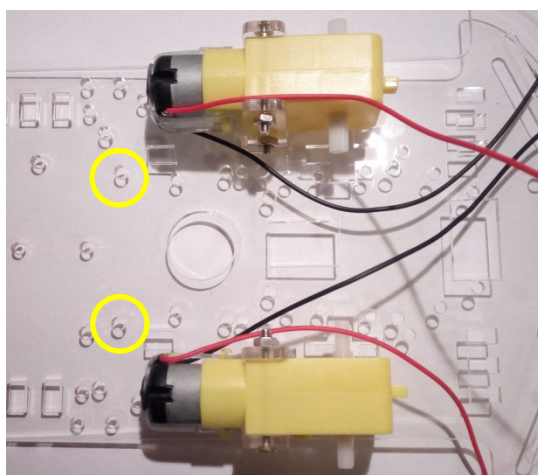
Εικόνα 68

Στη συνέχεια, τοποθετούμε το ένα μοτέρ ανάμεσα στα δύο στηρίγματα, με τις επαφές στραμμένες προς την εσωτερική και την πίσω πλευρά του οχήματος, και με το μαύρο καλώδιο στην πλευρά της βάσης. Ευθυγραμμίζουμε τις οπές που υπάρχουν στο περίβλημα των μοτέρ, με τις οπές στα στηρίγματα και χρησιμοποιώντας 2 από τις βίδες με το μεγαλύτερο μήκος και 2 παξιμάδια, βιδώνουμε το μοτέρ πάνω στα στηρίγματα (Εικόνα 69). Τα παξιμάδια τοποθετούνται στην εσωτερική πλευρά. Προσέξτε να μη σφίξετε πολύ τις βίδες, γιατί μπορεί να δημιουργηθεί πρόβλημα στην περιστροφή του μοτέρ.



Εικόνα 69

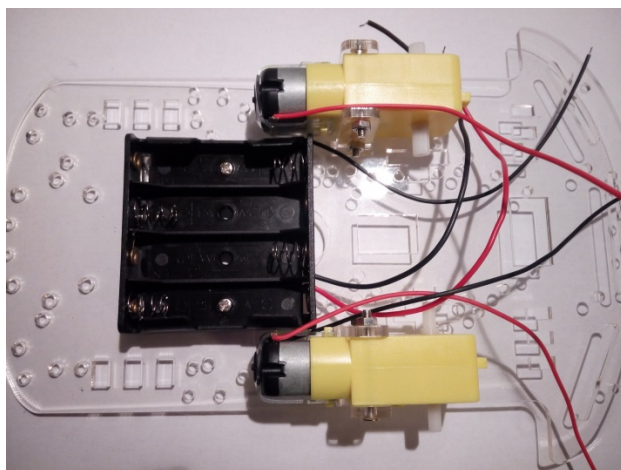
Με παρόμοιο τρόπο τοποθετούμε το δεύτερο μοτέρ στην άλλη πλευρά του οχήματος (Εικόνα 70).



Εικόνα 70

13.3 Τοποθέτηση μπαταριοθήκης

Η μπαταριοθήκη θα τοποθετηθεί στην κάτω πλευρά της βάσης (μαζί με τα μοτέρ). Για το σκοπό αυτό, ευθυγραμμίζουμε τις δύο εξωτερικές οπές της με αυτές που επισημαίνονται στην Εικόνα 70 (κίτρινα πλαίσια), φροντίζοντας τα καλώδιά της να είναι στραμμένα προς την μπροστινή πλευρά του οχήματος. Χρησιμοποιώντας 2 μικρές βίδες και 2 παξιμάδια, βιδώνουμε την μπαταριοθήκη πάνω στη βάση (Εικόνα 71).



Εικόνα 71

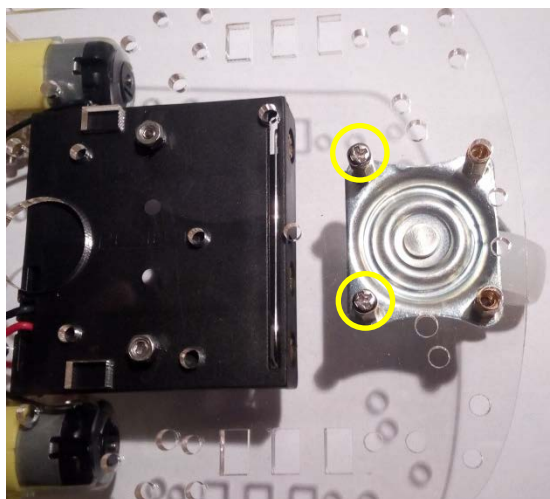
13.4 Τοποθέτηση πίσω ρόδας

Αρχικά, χρησιμοποιώντας 4 μικρές βίδες, βιδώνουμε τους αποστάτες πάνω στη μεταλλική βάση της πίσω ρόδας (Εικόνα 72).



Εικόνα 72

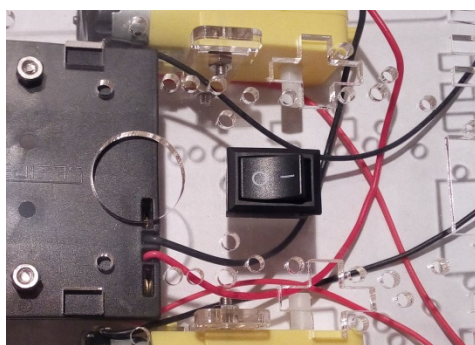
Στη συνέχεια, τοποθετούμε τη ρόδα στην κάτω πλευρά της βάσης, ευθυγραμμίζοντας τους αποστάτες με τις τέσσερις συμμετρικές οπές, στο πίσω μέρος του οχήματος. Χρησιμοποιώντας 2 ακόμα μικρές βίδες, βιδώνουμε τους δύο μπροστινούς αποστάτες πάνω στη βάση (Εικόνα 73).



Εικόνα 73

13.5 Τοποθέτηση και σύνδεση διακόπτη

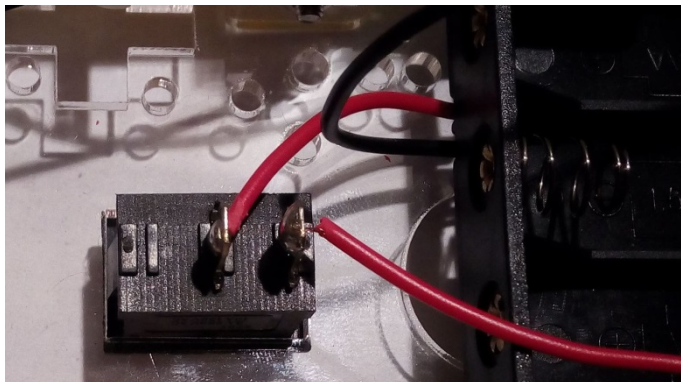
Αρχικά, τοποθετούμε το διακόπτη στην επάνω πλευρά της βάσης, στο ορθογώνιο άνοιγμα ανάμεσα στα δύο μοτέρ, με τη θέση Off (σύμβολο O) προς την πλευρά της μπαταριοθήκης (Εικόνα 74).



Εικόνα 74

Για να μπορούμε να ελέγχουμε με το διακόπτη την παροχή τροφοδοσίας προς το όχημα, θα πρέπει να τον παρεμβάλουμε ανάμεσα στην μπαταριοθήκη και στα

τροφοδοτούμενα κυκλώματα. Για το σκοπό αυτό, στη μία επαφή του διακόπτη θα πρέπει να συνδεθεί το κόκκινο καλώδιο που έρχεται από την μπαταριοθήκη και στη δεύτερη επαφή, θα συνδεθεί ένα καλώδιο που θα μεταφέρει την τροφοδοσία στα εξαρτήματα του οχήματος.

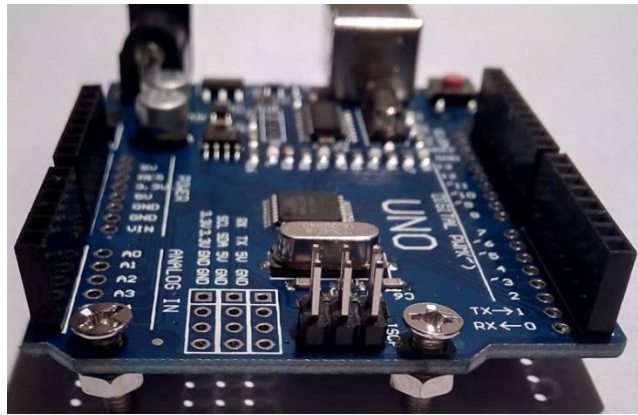


Εικόνα 75

Στην Εικόνα 75 φαίνεται η σχετική συνδεσμολογία. Το καλώδιο που έρχεται από την μπαταριοθήκη έχει κολληθεί στην επαφή στη μέση του διακόπτη, ενώ το καλώδιο που φεύγει προς τα υπόλοιπα εξαρτήματα στην ακριανή. Στην υλοποίηση που παρουσιάζεται, για να δημιουργήσουμε το επιπλέον καλώδιο (αυτό που φεύγει από το διακόπτη), έχουμε κόψει το καλώδιο της μπαταριοθήκης. Επειδή όμως το καλώδιο αυτό είναι ευπαθές και το μήκος του μπορεί να μην επαρκεί, προτείνεται να μην κοπεί, αλλά να συνδεθεί η υπάρχουσα άκρη του στην επαφή του διακόπτη και να χρησιμοποιηθεί ένα άλλο κομμάτι καλωδίου (εφόσον υπάρχει διαθέσιμο) για τη μεταφορά της τροφοδοσίας στα κυκλώματα του οχήματος.

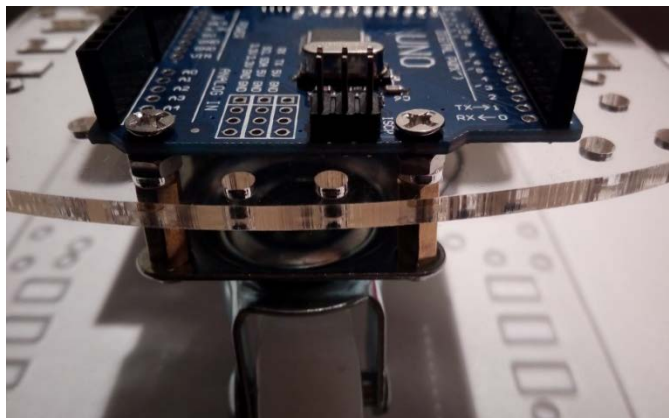
13.6 Τοποθέτηση Arduino

Για την τοποθέτηση του Arduino θα χρησιμοποιήσουμε τις 2 οπές που υπάρχουν στη δεξιά πλευρά της πλακέτας. Αρχικά, περνάμε τις 2 βίδες μεσαίου μήκους μέσα από τις οπές και κάτω από την πλακέτα περνάμε στις βίδες δύο παξιμάδια χωρίς να τα σφίξουμε (Εικόνα 76).



Εικόνα 76

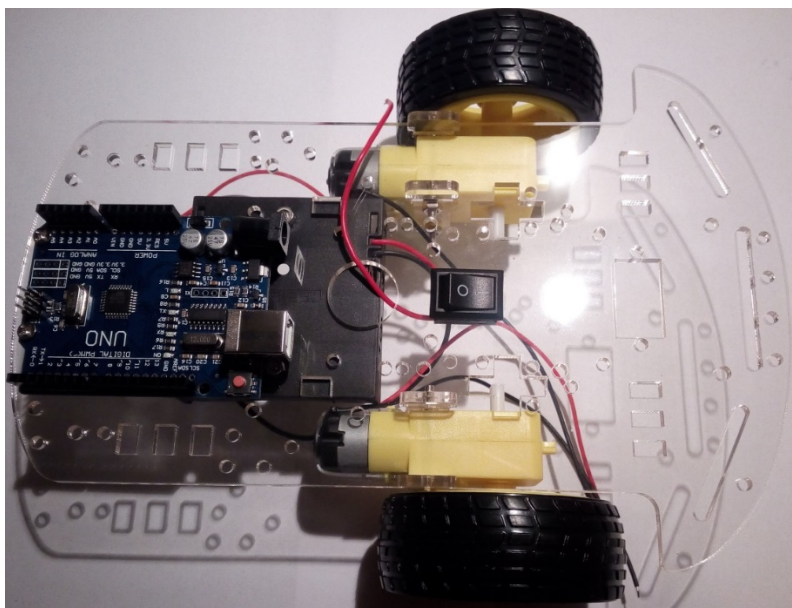
Στη συνέχεια περνάμε τις βίδες μέσα από τη βάση και τις βιδώνουμε στους ελεύθερους αποστάτες της πίσω ρόδας (βλέπε Εικόνα 73). Επειδή η ευθυγράμμιση δεν είναι τέλεια, καλό είναι να βιδώνεται τις βίδες λίγο-λίγο εναλλάξ. Στην Εικόνα 77 φαίνεται το Arduino τοποθετημένο πάνω στη βάση.



Εικόνα 77

13.7 Τοποθέτηση τροχών

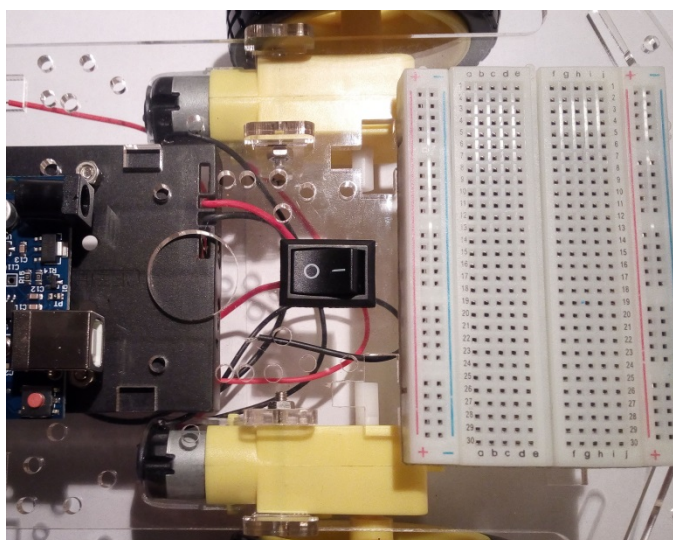
Τοποθετούμε τους τροχούς στους άξονες των 2 μοτέρ, αφήνοντας λίγη απόσταση από τη βάση, ώστε να μην ακουμπάνε κατά την περιστροφή τους (Εικόνα 78).



Εικόνα 78

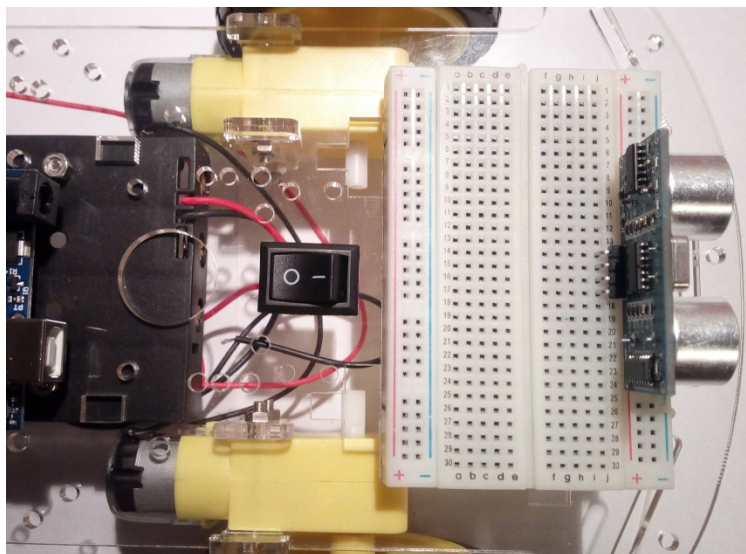
13.8 Τοποθέτηση breadboard και αισθητήρα HC-SR04

Τοποθετούμε το breadboard στην μπροστινή πλευρά του οχήματος, κοντά στο διακόπτη, όπως φαίνεται στην Εικόνα 79. Για τη σταθεροποίησή του μπορεί να χρησιμοποιηθεί είτε το αυτοκόλλητο που υπάρχει στην κάτω πλευρά του, είτε κάποιο άλλο μέσο (π.χ. κολλητική ταινία διπλής όψης), προκειμένου να υπάρχει δυνατότητα αφαίρεσης και επανατοποθέτησής του.



Εικόνα 79

Ακολουθώντας, τοποθετούμε τον αισθητήρα απόστασης στο κέντρο της εξωτερικής πλευράς του breadboard (ΠΡΟΣΟΧΗ όχι στις δύο εξωτερικές σειρές), όπως φαίνεται στην Εικόνα 80.

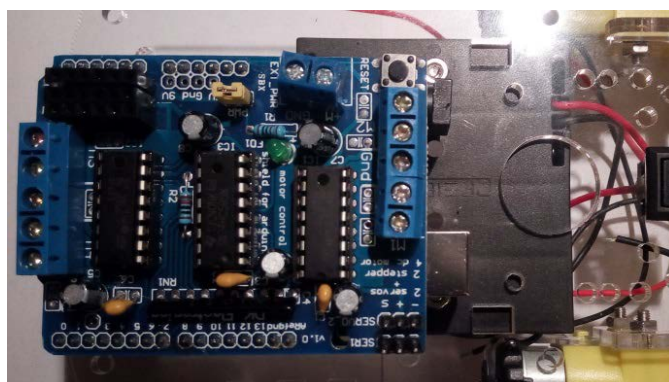


Εικόνα 80

13.9 Τοποθέτηση κυκλώματος οδήγησης

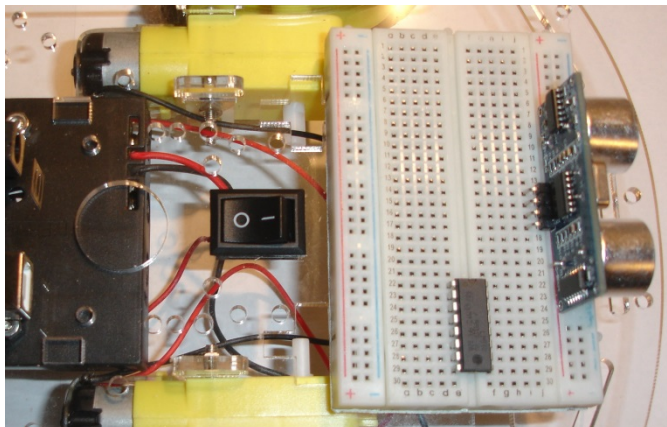
Όπως έχει ήδη αναφερθεί, στο πακέτο υλικών παρέχονται δύο εναλλακτικές λύσεις για την οδήγηση των μοτέρ, το motor shield και το chip L293D. Στο σημείο αυτό θα πρέπει να αναφερθεί ότι, η χρήση του motor shield καθιστά ευκολότερη την υλοποίηση τόσο του κυκλώματος, όσο και του προγράμματος για τη λειτουργία του ρομποτικού οχήματος.

Αν επιλέξετε να χρησιμοποιήσετε το motor shield, τότε θα χρειαστεί να το κουμπώσετε πάνω στο Arduino, που είναι ήδη τοποθετημένο πάνω στο όχημα (Εικόνα 81).



Εικόνα 81

Αν πάλι επιλέξετε να χρησιμοποιήσετε το chip L293D, θα πρέπει να το τοποθετήσετε πάνω στο breadboard, με κάθε ακροδέκτη του να συνδέεται σε μία ξεχωριστή ελεύθερη μισή στήλη του breadboard. Όπως φαίνεται και στην Εικόνα 82, τοποθετούμε το chip με την ημικυκλική εγκοπή στραμμένη προς το κέντρο του οχήματος.



Εικόνα 82

14 Σύνδεση εξαρτημάτων

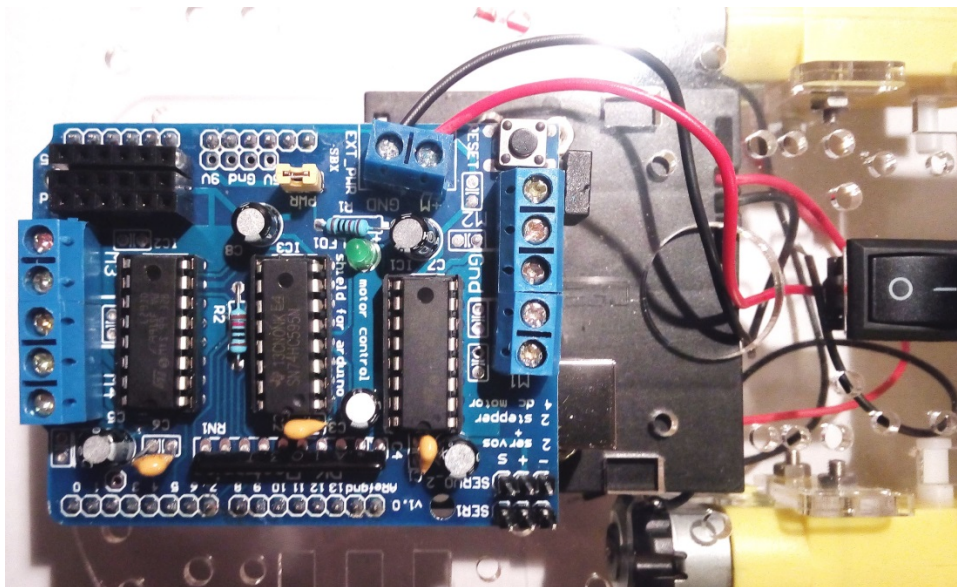
Στη συνέχεια αναλύεται ο τρόπος σύνδεσης των εξαρτημάτων, που περιλαμβάνονται στο ρομποτικό όχημα. Ουσιαστικά παρουσιάζονται δύο διαφορετικές συνδεσμολογίες, ανάλογα με το κύκλωμα οδήγησης που θα επιλεγεί.

14.1 Συνδεσμολογία με χρήση του motor shield

Η χρήση του motor shield απλοποιεί σημαντικά την υλοποίηση του κυκλώματος λειτουργίας και ελέγχου του ρομποτικού οχήματος.

14.1.1 Σύνδεση πηγής τροφοδοσίας

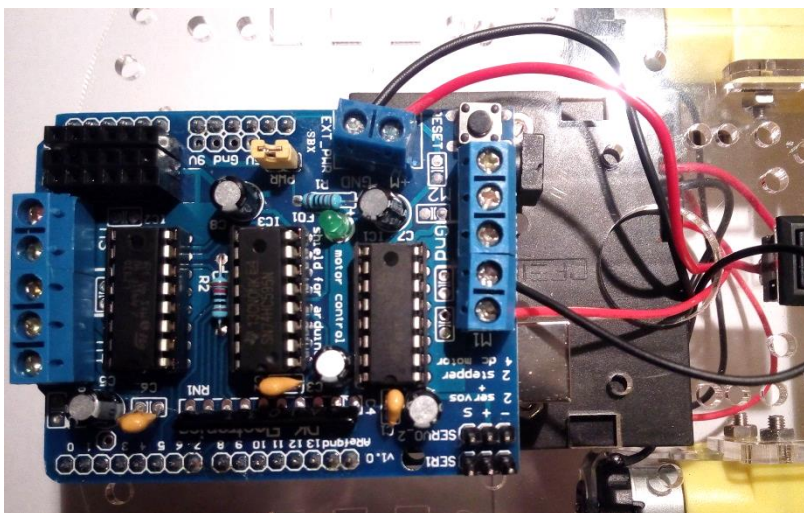
Τα δύο καλώδια τροφοδοσίας που έρχονται από την μπαταριοθήκη (και το διακόπτη) συνδέονται στις επαφές του shield με της σήμανση EXT_PWR. Ιδιαίτερη προσοχή πρέπει να δοθεί στην πολικότητα της σύνδεσης με την πηγή. Συγκεκριμένα, το κόκκινο καλώδιο που έρχεται μέσω του διακόπτη, συνδέεται στην επαφή +M, ενώ το μαύρο καλώδιο που έρχεται απευθείας από την μπαταριοθήκη συνδέεται στην επαφή GND (Εικόνα 83).



Εικόνα 83

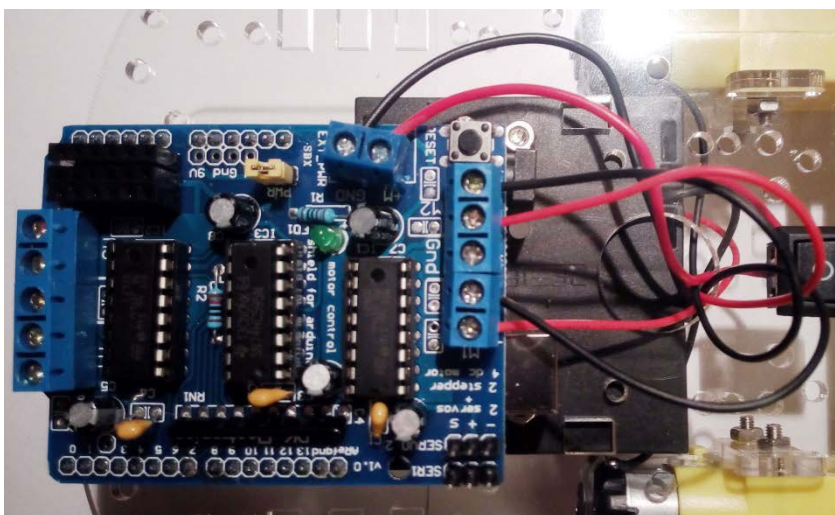
14.1.2 Σύνδεση μοτέρ

Το μοτέρ που βρίσκεται στη δεξιά πλευρά του οχήματος, συνδέεται στις δύο επαφές με τη σήμανση M1, με την πολικότητα που φαίνεται στην Εικόνα 84. Συγκεκριμένα, το κόκκινο καλώδιο συνδέεται στην εξωτερική επαφή (κάτω στην Εικόνα) και το μαύρο στη διπλανή της.



Εικόνα 84

Τα καλώδια από το αριστερό μοτέρ συνδέονται στις επαφές με τη σήμανση M2, με την ίδια σειρά (κόκκινο, μαύρο) (Εικόνα 85).



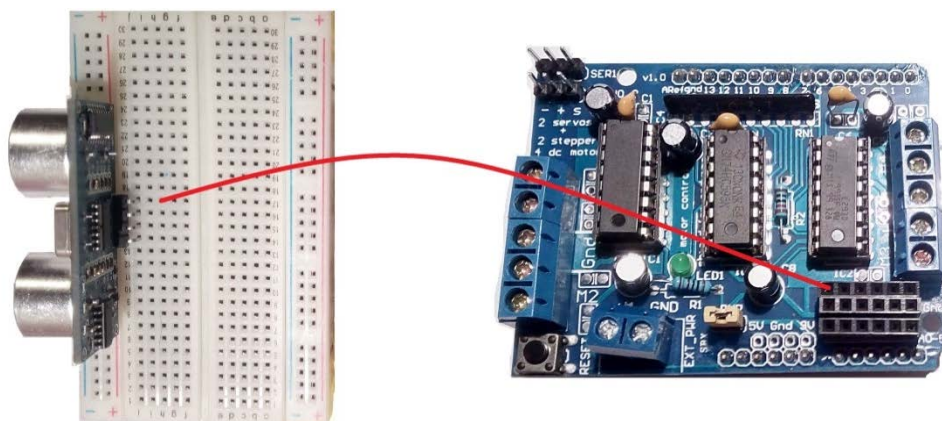
Εικόνα 85

Όπως φαίνεται και στην Εικόνα 85, όλα τα καλώδια (από την μπαταριοθήκη, το διακόπτη και τα μοτέρ) έχουν περαστεί μέσα από το στρογγυλό άνοιγμα που υπάρχει στην άκρη της μπαταριοθήκης.

14.1.3 Σύνδεση αισθητήρα απόστασης

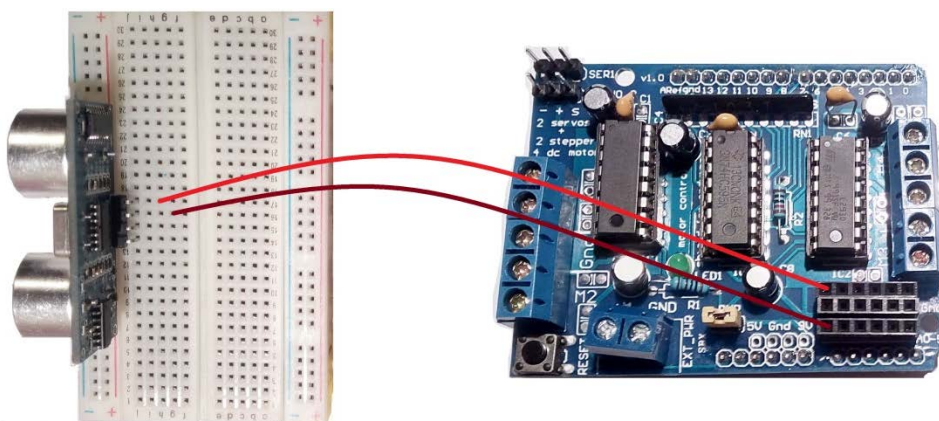
Όπως αναφέρθηκε και σε προηγούμενη ενότητα, η πλειοψηφία των ψηφιακών ακροδεκτών του Arduino δεσμεύονται από το motor shield. Για το λόγο αυτό, η σύνδεση του αισθητήρα HC-SR04, θα γίνει στους ακροδέκτες A0 – A5, οι οποίοι λειτουργούν και ως ψηφιακοί ακροδέκτες εισόδου/εξόδου.

Για τη σύνδεση του αισθητήρα στο motor shield, θα αξιοποιήσουμε τα female headers που κολλήσαμε κατά την προετοιμασία των εξαρτημάτων. Αρχικά, συνδέουμε με ένα καλώδιο (μέσω του breadboard) τον ακροδέκτη Vcc του αισθητήρα, σε μία από τις οπές του header με τη σήμανση +5 (εσωτερικό) (Εικόνα 86).



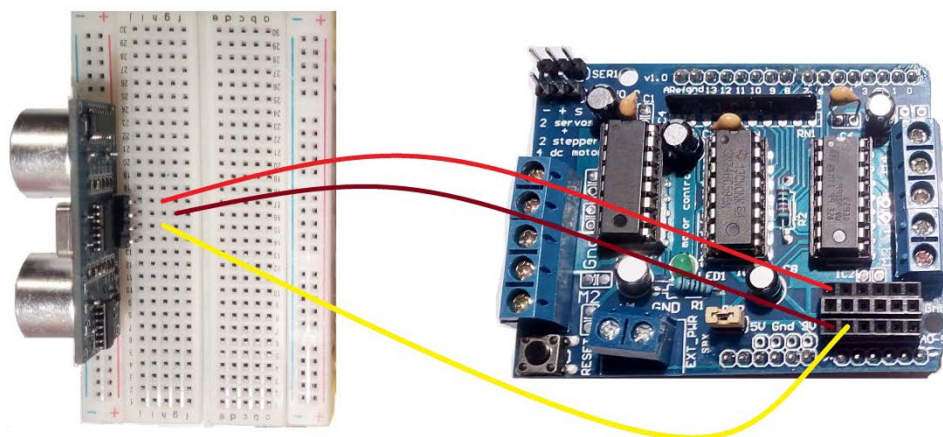
Εικόνα 86

Στη συνέχεια, όπως φαίνεται στην Εικόνα 87, συνδέουμε τον ακροδέκτη Trig του αισθητήρα, στον ακροδέκτη A0 του shield (πρώτη οπή στο εξωτερικό header, με τη σήμανση A0-A5).



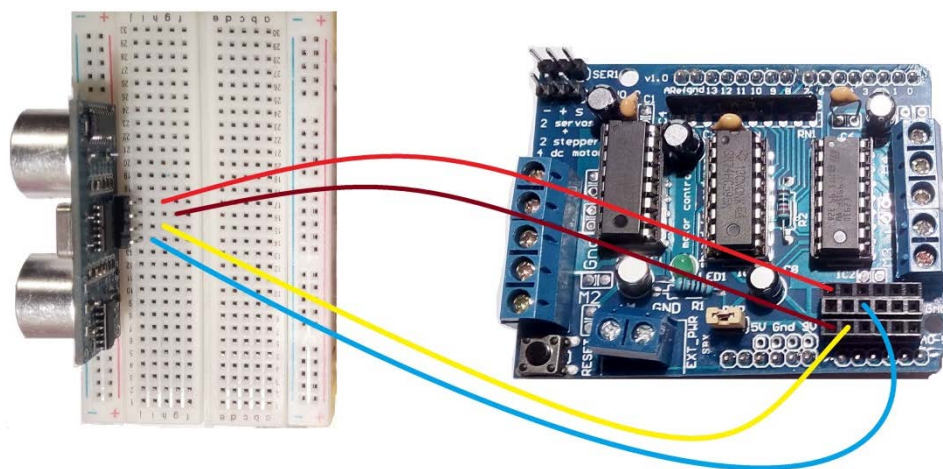
Εικόνα 87

Το επόμενο βήμα, είναι η σύνδεση του ακροδέκτη Echo του αισθητήρα, στον ακροδέκτη A1 του shield (δεύτερη οπή στο εξωτερικό header, με τη σήμανση A0-A5). Η σχετική σύνδεση παρουσιάζεται στην Εικόνα 88.



Εικόνα 88

Τέλος, συνδέουμε τον ακροδέκτη Gnd του αισθητήρα, σε μία από τις οπές του header με τη σήμανση GND (μεσαίο) (Εικόνα 89).



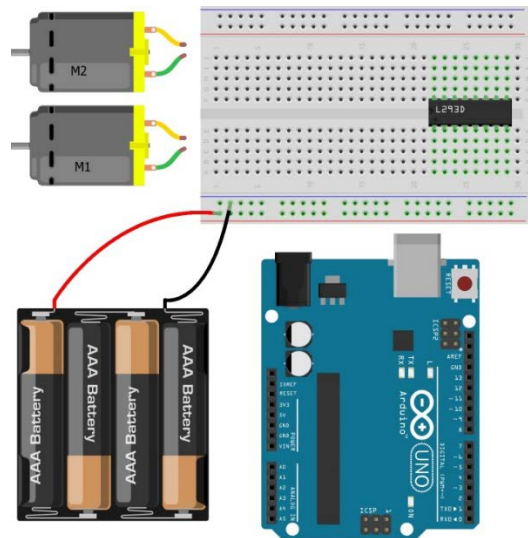
Εικόνα 89

14.2 Συνδεσμολογία με χρήση του chip L293D

Όπως είδαμε στην ενότητα της συναρμολόγησης, το L293D τοποθετείται πάνω στο breadboard, μαζί με τον αισθητήρα απόστασης. Σαν αποτέλεσμα, όλες οι συνδέσεις υλοποιούνται μέσω του breadboard. Στις συνδέσεις που παρουσιάζονται στη συνέχεια, θεωρούμε ότι το chip έχει τοποθετηθεί με την ημικυκλική εγκοπή του, προς τα αριστερά.

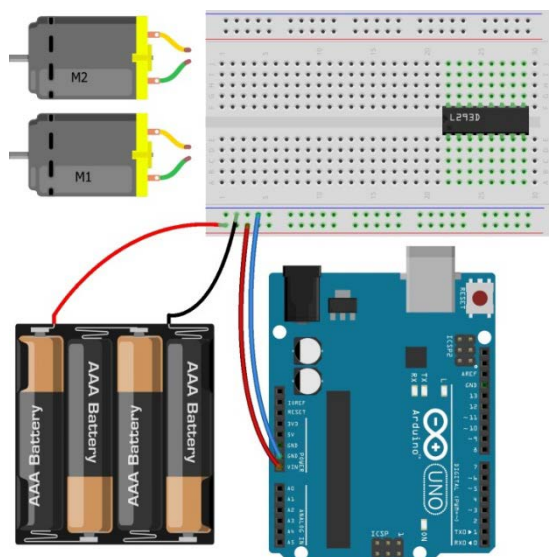
14.2.1 Συνδέσεις τροφοδοσίας

Αρχικά, συνδέουμε τα καλώδια που έρχονται από τη μπαταριοθήκη (το κόκκινο μέσω του διακόπτη), στις σειρές τροφοδοσίας του breadboard, προκειμένου να μπορούμε να τροφοδοτήσουμε από αυτές όλα τα εξαρτήματα. Συγκεκριμένα, το κόκκινο καλώδιο συνδέεται στη σειρά + (κόκκινη) και το μαύρο στη σειρά – (μπλε) του breadboard (Εικόνα 90). Για να φέρουμε τα καλώδια στην πάνω πλευρά της βάσης, τα περνάμε μέσα από το στρογγυλό άνοιγμα μπροστά από την μπαταριοθήκη.



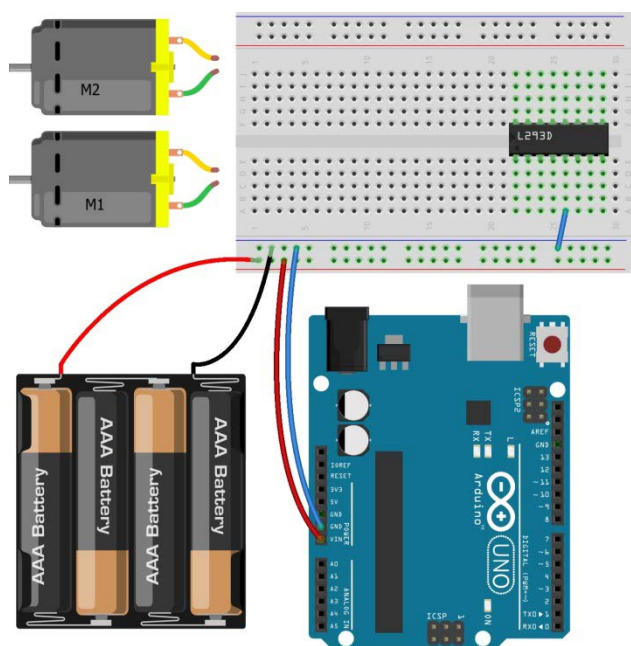
Εικόνα 90

Για την τροφοδοσία του Arduino, θα χρησιμοποιήσουμε τους ακροδέκτες GND και Vin. Με δύο καλώδια, συνδέουμε το pin Vin στην κόκκινη σειρά του breadboard και το GND στην μπλε (Εικόνα 91).



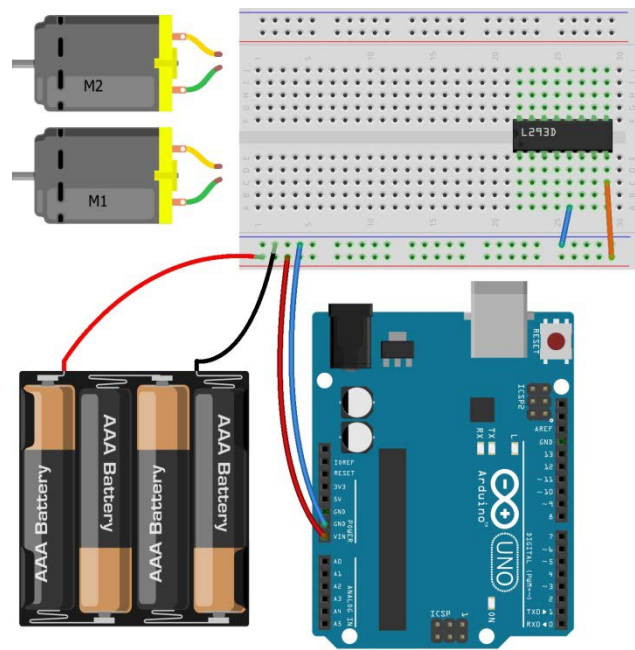
Εικόνα 91

Στη συνέχεια, κάνουμε τις συνδέσεις τροφοδοσίας για το L293D. Αρχικά, συνδέουμε έναν από τους ακροδέκτες γείωσης στην κάτω πλευρά του chip (4^{ος} και 5^{ος} από αριστερά στην Εικόνα) στη σειρά γείωσης του breadboard (Εικόνα 92).



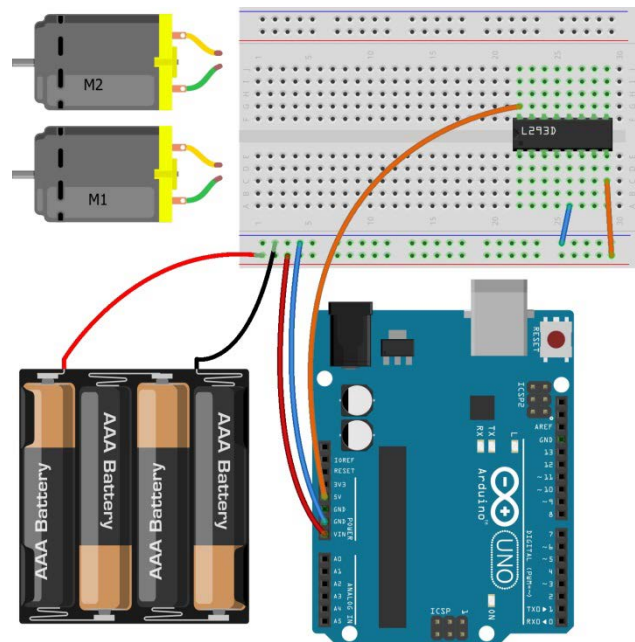
Εικόνα 92

Συνδέουμε τον ακροδέκτη +V Motor για την τροφοδοσία των μοτέρ (δεξιότερος ακροδέκτης στη κάτω πλευρά του chip), στην κόκκινη σειρά του breadboard (Εικόνα 93).



Εικόνα 93

Τέλος, συνδέουμε τον ακροδέκτη +V (1^{ος} από αριστερά στην πάνω πλευρά του chip) στον ακροδέκτη 5V του Arduino (Εικόνα 94). Η πλακέτα UNO που περιλαμβάνεται στον εξοπλισμό διαθέτει 2 ακροδέκτες 5V. Επιλέξτε οποιονδήποτε από τους δύο. Ο ελεύθερος θα χρησιμοποιηθεί αργότερα για την τροφοδοσία του HC-SR04.

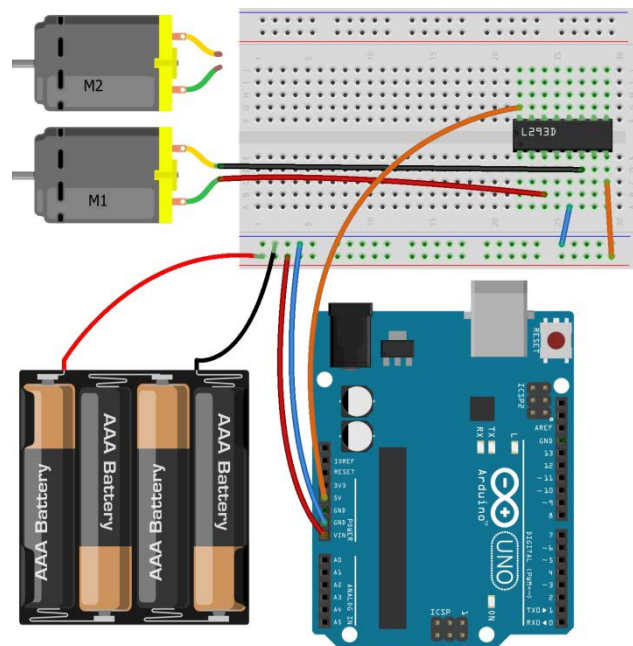


Εικόνα 94

14.2.2 Συνδέσεις μοτέρ

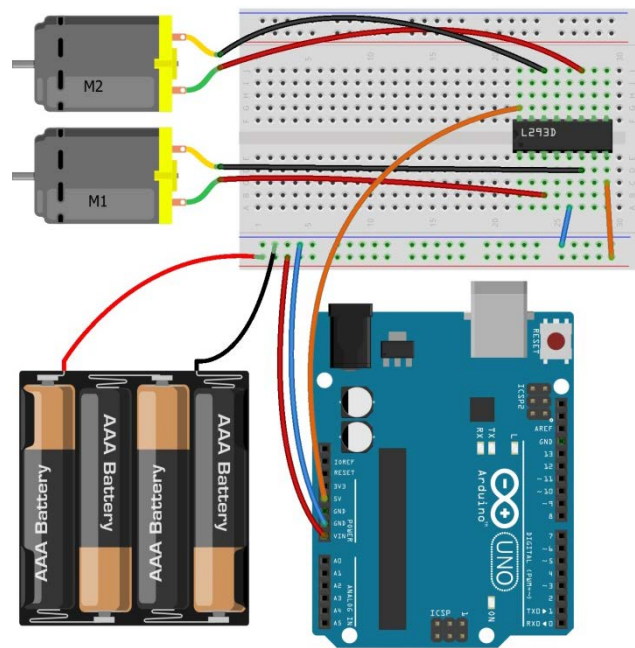
Το επόμενο βήμα είναι η σύνδεση των μοτέρ στο L293D. Φέρνουμε τα καλώδια των μοτέρ στην επάνω πλευρά της βάσης, περνώντας τα μέσα από τα ανοίγματα σε σχήμα σταυρού, που υπάρχουν δίπλα σε κάθε μοτέρ.

Συνδέουμε το κόκκινο καλώδιο του δεξιού μοτέρ (M1 στην Εικόνα) στον ακροδέκτη Out 1 (3^{ος} από αριστερά, στην κάτω πλευρά του chip) και το μαύρο στον ακροδέκτη Out 2 (6^{ος} από αριστερά, στην κάτω πλευρά του chip) (Εικόνα 95).



Εικόνα 95

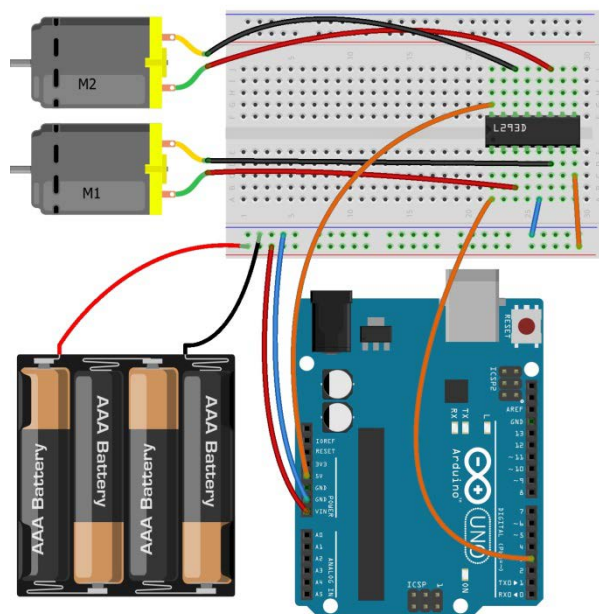
Παρόμοια, συνδέουμε το κόκκινο καλώδιο του αριστερού μοτέρ (M2 στην Εικόνα), στον ακροδέκτη Out 3 (6^{ος} από αριστερά στην πάνω πλευρά του chip) και το μαύρο στον ακροδέκτη Out 4 (3^{ος} από αριστερά στην πάνω πλευρά του chip) (Εικόνα 96).



Εικόνα 96

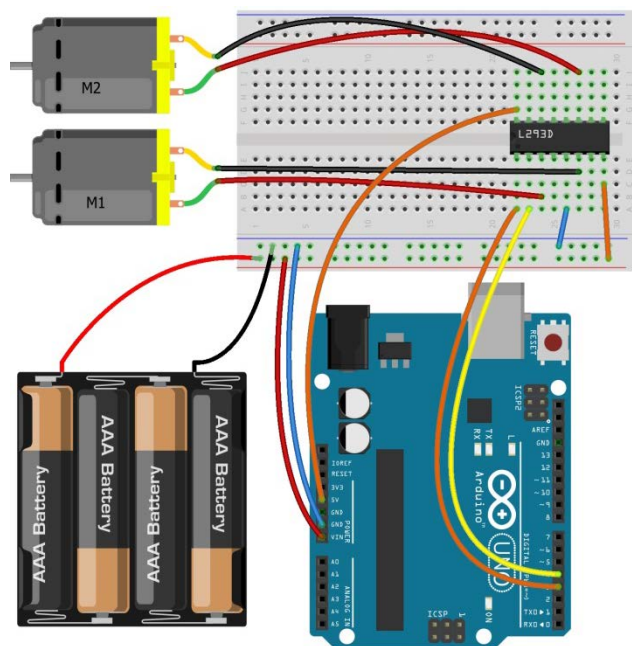
14.2.3 Συνδέσεις σημάτων ελέγχου

Ακολούθως, συνδέουμε στο L293D, τους ακροδέκτες του Arduino που θα χρησιμοποιηθούν για τον έλεγχο των μοτέρ.

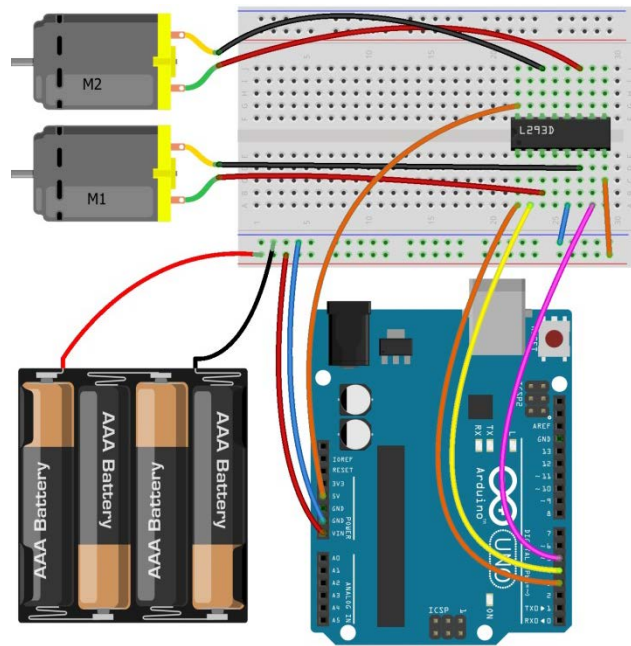


Εικόνα 97

Για τον έλεγχο του δεξιού μοτέρ (M1), θα χρησιμοποιηθούν οι ψηφιακοί ακροδέκτες 3, 4 και 5 του Arduino. Πιο συγκεκριμένα, ο ψηφιακός ακροδέκτης 3 συνδέεται στον ακροδέκτη Enable 1 (πρώτος από αριστερά στην κάτω πλευρά του L293D), για την ενεργοποίηση και απενεργοποίηση του μοτέρ (Εικόνα 97). Ο ψηφιακός ακροδέκτης 4, συνδέεται στον ακροδέκτη In 1 (2^{ος} από αριστερά στην κάτω πλευρά του chip) (Εικόνα 98) και ο ακροδέκτης 5 στον ακροδέκτη In 2 (7^{ος} από αριστερά στην κάτω πλευρά του chip) (Εικόνα 99).



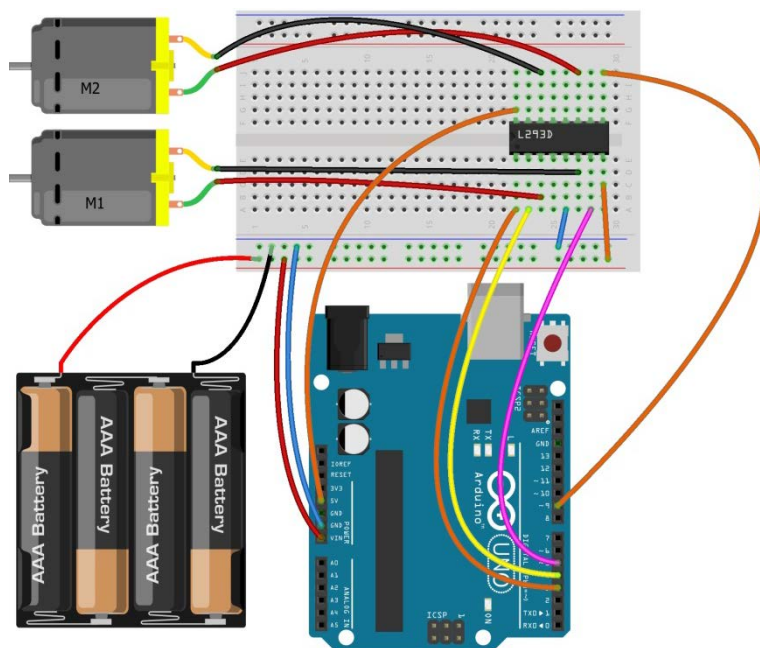
Εικόνα 98



Εικόνα 99

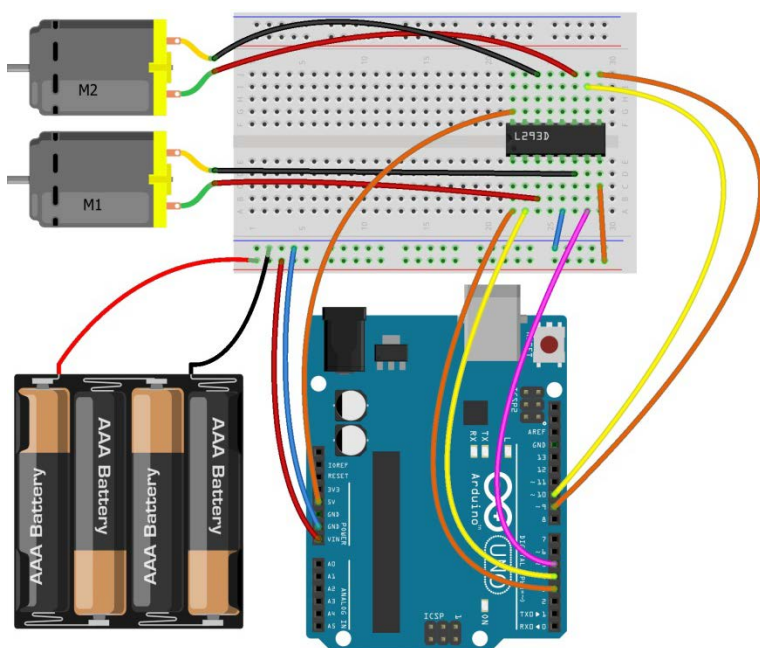
Παρόμοια, ο έλεγχος του αριστερού μοτέρ (M2), θα γίνεται με τους ακροδέκτες 9, 10 και 11 του Arduino.

Ο ψηφιακός ακροδέκτης 9, θα συνδεθεί με τον ακροδέκτη Enable 2 του L293D (1^{ος} από δεξιά στην πάνω σειρά του chip), για την ενεργοποίηση και απενεργοποίηση του μοτέρ.



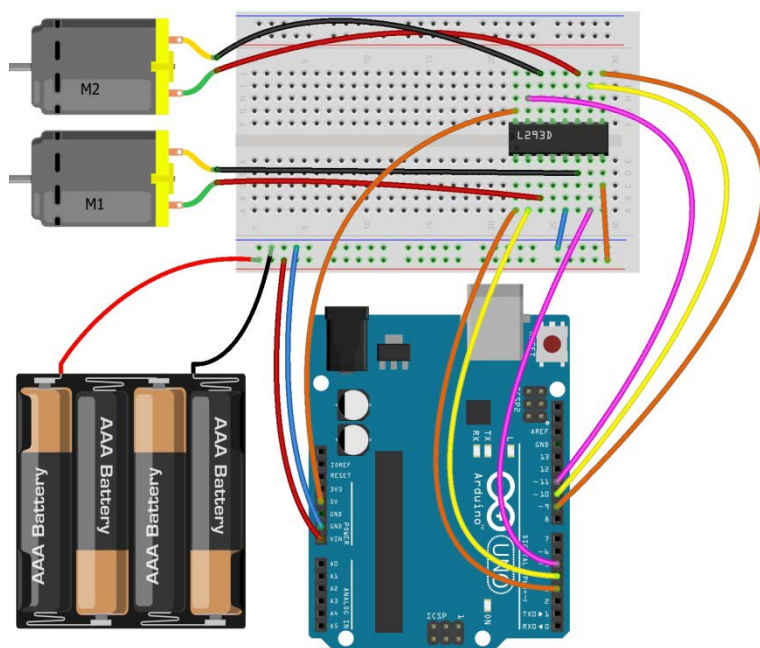
Εικόνα 100

Ο ακροδέκτης 10 θα συνδεθεί με τον ακροδέκτη In 3 του L293D (2^{ος} από δεξιά στην πάνω πλευρά του chip) (Εικόνα 101).



Εικόνα 101

Ο ακροδέκτης 11 του Arduino, θα συνδεθεί με τον ακροδέκτη In 4 (2^{ος} από αριστερά στην πάνω πλευρά του L293D)(Εικόνα 102).

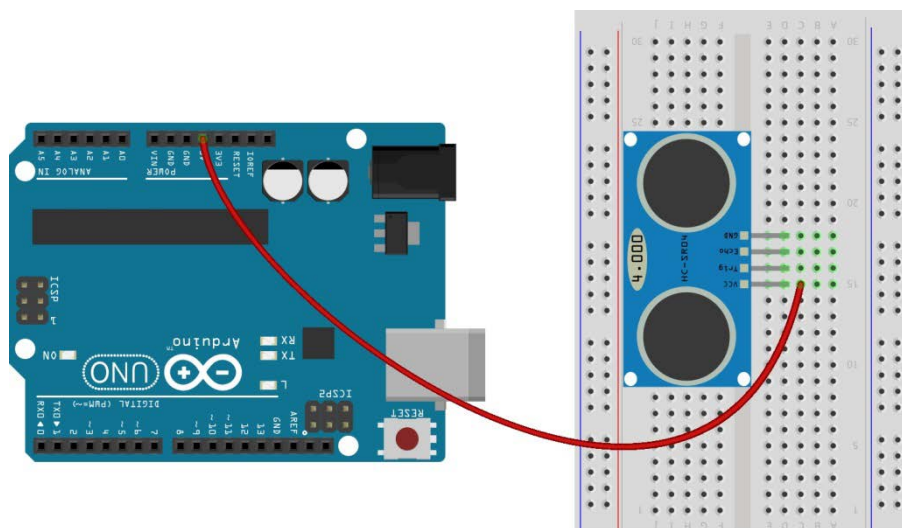


Εικόνα 102

14.2.4 Σύνδεση αισθητήρα απόστασης

Έχοντας ολοκληρώσει τις συνδέσεις του L293D, το μοναδικό που απομένει είναι η σύνδεση του HC-SR04 με το Arduino. Στις εικόνες που ακολουθούν, εμφανίζονται μόνο ο αισθητήρας απόστασης, το breadboard και το Arduino UNO, προκειμένου τα σχετικά κυκλώματα να είναι πιο κατανοητά.

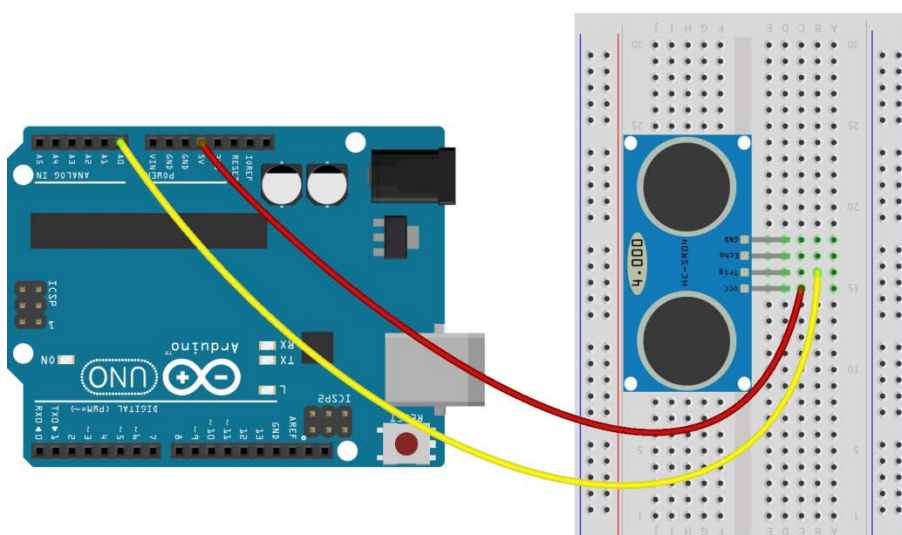
Επιπλέον, για να είναι πιο ευδιάκριτες οι συνδέσεις, ο HC-SR04 παρουσιάζεται τοποθετημένος στη μέση της μπροστινής πλευράς του breadboard και οι συνδέσεις γίνονται μπροστά από τον αισθητήρα. Στο πραγματικό κύκλωμα, τοποθετήστε τον HC-SR04 όπως περιγράφεται στις οδηγίες συναρμολόγησης (Εικόνα 80) και κάντε τις συνδέσεις πίσω από τους αντίστοιχους ακροδέκτες.



Εικόνα 103

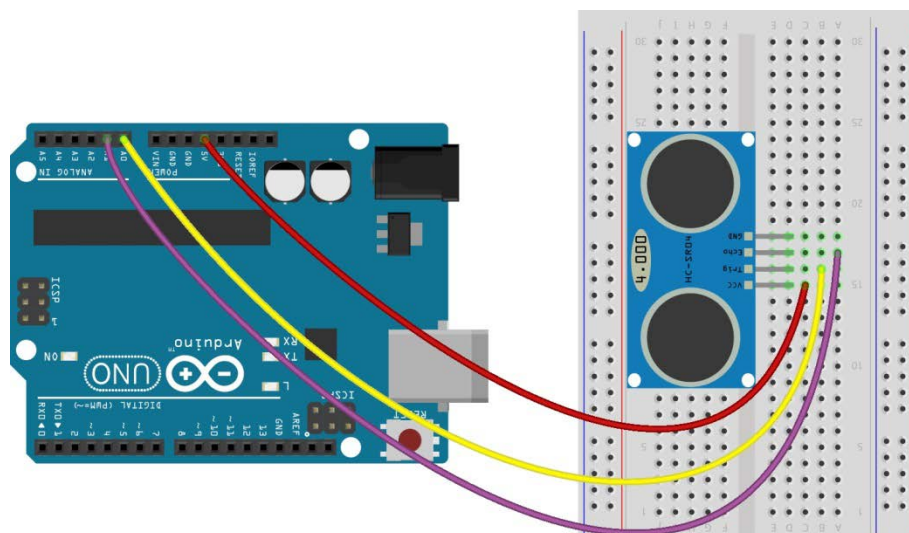
Αρχικά, συνδέουμε τον ακροδέκτη Vcc του HC-SR04 στον ακροδέκτη 5V του Arduino (Εικόνα 103). Όπως αναφέρθηκε και παραπάνω, το UNO που περιλαμβάνεται στον εξοπλισμό, διαθέτει δύο τέτοιους ακροδέκτες. Ο ένας χρησιμοποιείται ήδη από το L293D, οπότε εδώ θα χρησιμοποιήσετε τον ελεύθερο ακροδέκτη 5 V.

Στη συνέχεια, συνδέουμε τον ακροδέκτη Trig του αισθητήρα, στον ακροδέκτη A0 του Arduino (Εικόνα 104).



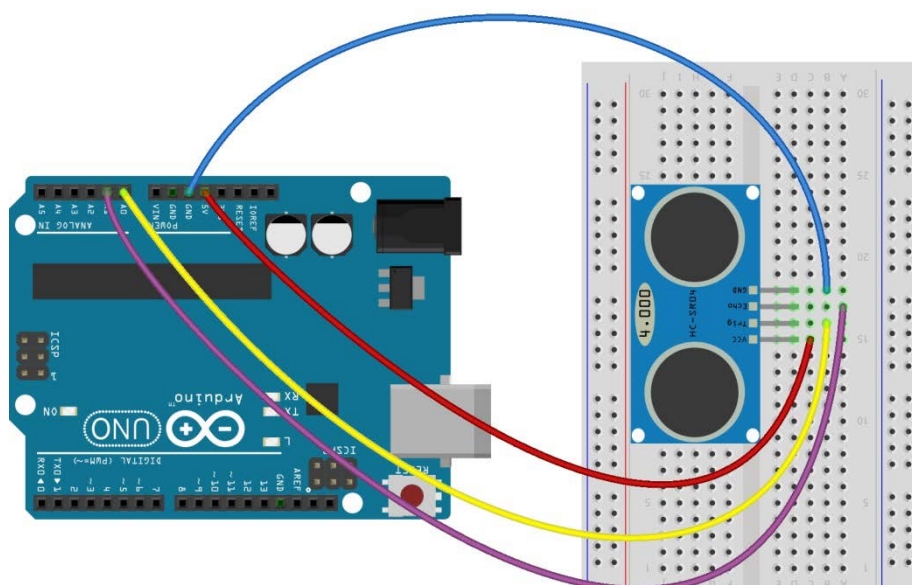
Εικόνα 104

Με ένα ακόμα καλώδιο, συνδέουμε τον ακροδέκτη Echo του αισθητήρα, στον ακροδέκτη A1 του Arduino (Εικόνα 105). Η επιλογή των ακροδεκτών A0 και A1, για τα σήματα Trig και Echo του αισθητήρα, έγινε προκειμένου να υπάρχει ομοιομορφία στα τελικά προγράμματα που θα αναπτυχθούν για τις δύο εναλλακτικές υλοποιήσεις του ρομποτικού οχήματος (motor shield και L293D chip).



Εικόνα 105

Τέλος, συνδέουμε τον ακροδέκτη Gnd του αισθητήρα, με έναν από τους ακροδέκτες GND του UNO (Εικόνα 106).



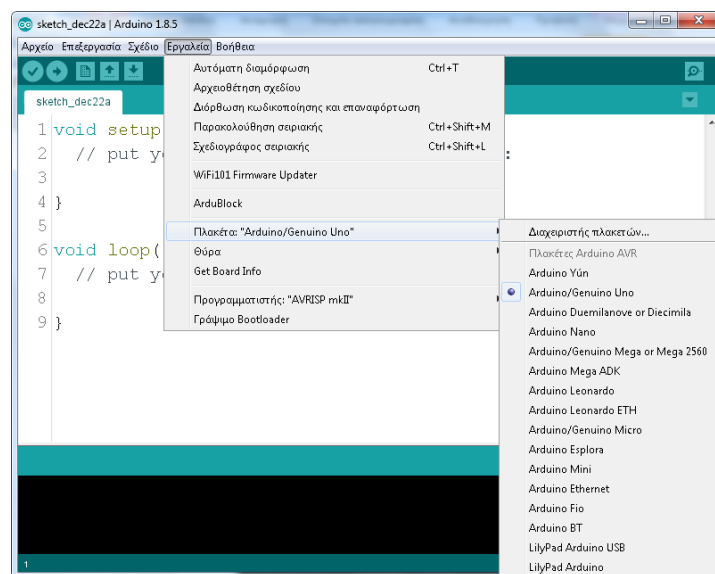
Εικόνα 106

15 Προγραμματισμός για την κίνηση του οχήματος

Η ανάπτυξη του προγράμματος λειτουργίας του ρομποτικού οχήματος, θα πραγματοποιηθεί σε δύο φάσεις. Στην πρώτη φάση, θα αναπτύξουμε μία σειρά από υπορουτίνες, για τις κινήσεις που θα μπορεί να εκτελεί το όχημα (μπροστά, πίσω, αριστερά, δεξιά και σταμάτημα). Με το κυρίως πρόγραμμα, το ρομποτικό όχημα θα εκτελεί μία επίδειξη των κινήσεων αυτών.

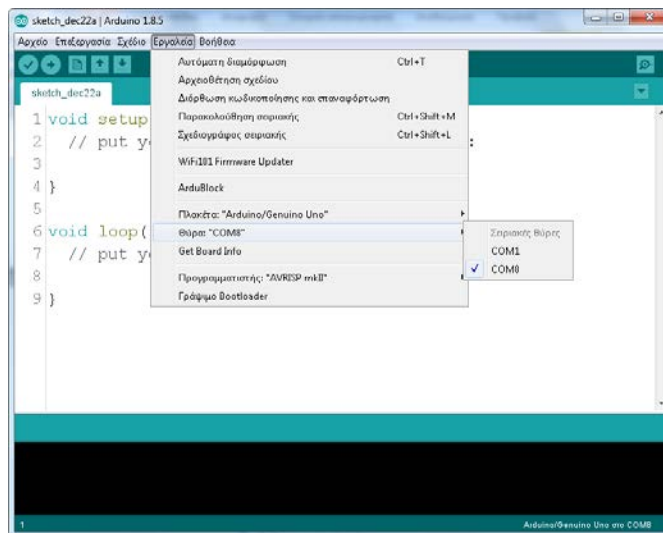
15.1 Σύνδεση Arduino με τον υπολογιστή

Συνδέουμε το Arduino UNO 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 107).



Εικόνα 107

Ακολουθώς, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino (Εικόνα 108).

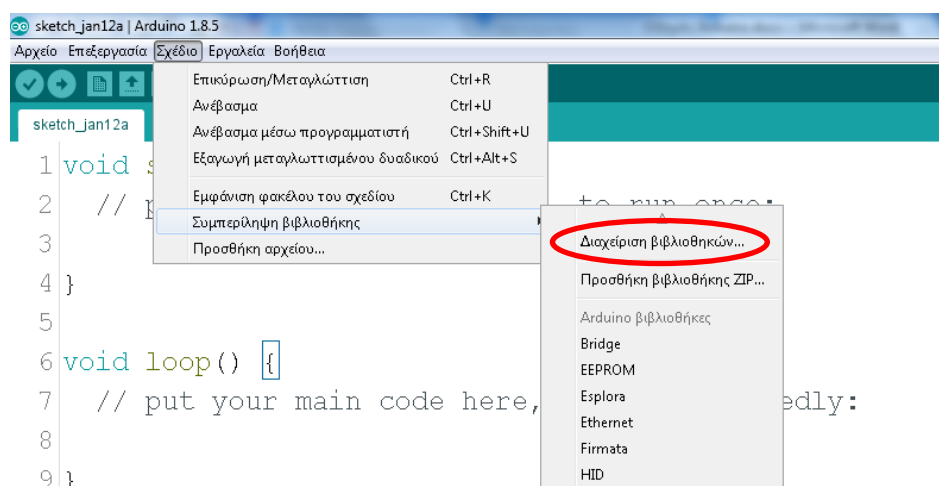


Εικόνα 108

15.2 Πρόγραμμα για υλοποίηση με motor shield

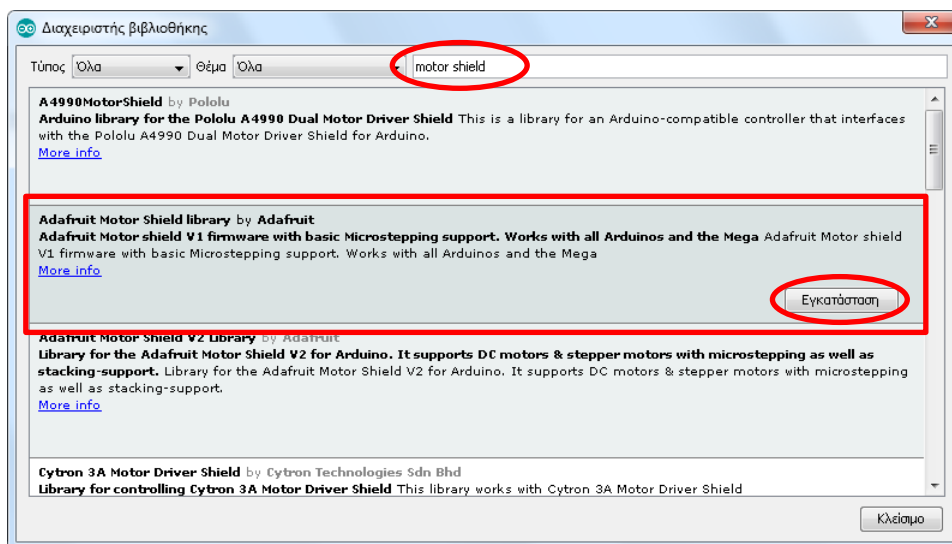
Για τον προγραμματισμό του motor shield, υπάρχουν στο ArduBlock έτοιμα πλακίδια, τα οποία χρησιμοποιούν τη βιβλιοθήκη *Adafruit Motor Shield library*. Πριν ξεκινήσουμε λοιπόν την ανάπτυξη του προγράμματός μας, θα πρέπει να εγκαταστήσουμε τη σχετική βιβλιοθήκη.

Από το μενού *Σχέδιο* του Arduino IDE, κάνουμε κλικ στην επιλογή *Συμπερίληψη βιβλιοθήκης*, και στο υπομενού που εμφανίζεται, πατάμε *Διαχείριση βιβλιοθηκών* (Εικόνα 109).



Εικόνα 109

Στο παράθυρο του *Διαχειριστή βιβλιοθήκης*, στο πλαίσιο κειμένου, πληκτρολογούμε τις λέξεις *motor shield*. Από τα αποτελέσματα που εμφανίζονται, κάνουμε κλικ στο *Adafruit Motor Shield library* (όχι στο *Adafruit Motor Shield V2 library*) και στη συνέχεια πατάμε το κουμπί *Εγκατάσταση* που εμφανίζεται (Εικόνα 110).



Εικόνα 110

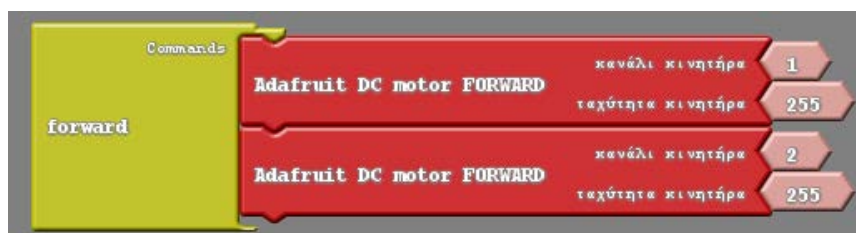
Αφού ολοκληρωθεί η εγκατάσταση, κλείνουμε το παράθυρο του *Διαχειριστή βιβλιοθήκης* και στη συνέχεια από το μενού *Εργαλεία* του Arduino IDE, επιλέγουμε *ArduBlock* και ανοίγει το περιβάλλον οπτικού προγραμματισμού.

Οι εντολές για τον έλεγχο των μοτέρ μέσω του motor shield, βρίσκονται στην ομάδα πλακιδίων *Adafruit*.

Για να δημιουργήσουμε τις υπορουτίνες που θέλουμε, από την ομάδα εντολών *Έλεγχος*, παίρνουμε το πλακίδιο *υπορουτίνα Commands*. Κάνοντας κλικ στη λέξη *υπορουτίνα*, την αντικαθιστούμε με το όνομα που θέλουμε να έχει η υπορουτίνα μας και στη συνέχεια, κουμπώνουμε στο πλακίδιο τις διάφορες εντολές που θέλουμε να εκτελούνται κατά την κλήση της.

Αρχικά, δημιουργούμε την υπορουτίνα με όνομα *forward* για την κίνηση του οχήματος προς τα εμπρός (Εικόνα 111). Η εντολή *Adafruit DC motor FORWARD* κάνει το μοτέρ να κινείται προς τα εμπρός. Η παράμετρος *κανάλι κινητήρα* καθορίζει σε ποιο από τα 4 μοτέρ που υποστηρίζει το shield θα εφαρμοστεί η εντολή (π.χ. το 1 αντιστοιχεί στο ζευγάρι επαφών M1 του shield, όπου έχουμε συνδέσει το δεξί μοτέρ του οχήματός μας). Η παράμετρος *ταχύτητα κινητήρα*, ορίζει την ταχύτητα περιστροφής του κινητήρα

(τεχνική PWM: 255 μέγιστη ταχύτητα, 0 μοτέρ σταματημένο). Η υπορουτίνα κάνει τα 2 μοτέρ μας να κινηθούν προς τα εμπρός.



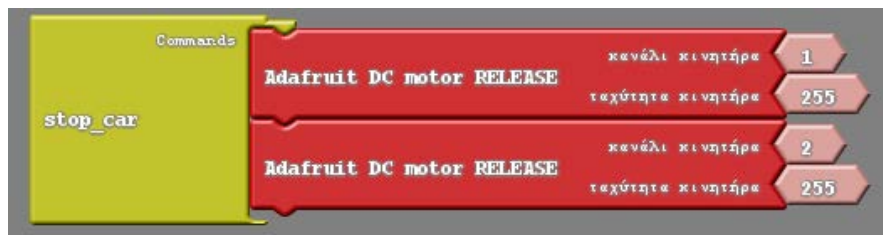
Εικόνα 111

Με παρόμοιο τρόπο, αναπτύσσουμε τις υπορουτίνες *backward*, *left* και *right*, που επιτρέπουν στο όχημά μας να κινείται προς τα πίσω, και να στρίβει δεξιά και αριστερά αντίστοιχα (Εικόνα 112).



Εικόνα 112

Η τελευταία υπορουτίνα που φτιάχνουμε, είναι η *stop_car*, για το σταμάτημα του οχήματος (Εικόνα 113). Η εντολή *Adafruit DC motor RELEASE*, κάνει το μοτέρ που καθορίζεται με την παράμετρο *κανάλι κινητήρα* να σταματήσει. Εδώ, η παράμετρος *ταχύτητα κινητήρα* δεν επηρεάζει ουσιαστικά τη λειτουργία του προγράμματος.



Εικόνα 113

Η κλήση μίας υπορουτίνας που έχουμε ορίσει νωρίτερα, γίνεται στο ArduBlock με το πλακίδιο *υπορουτίνα* (ομάδα εντολών *Έλεγχος*). Και εδώ, κάνοντας κλικ στη λέξη υπορουτίνα, την αντικαθιστούμε με το όνομα της συνάρτησης που θέλουμε να καλέσουμε.

Στο κυρίως πρόγραμμα, καλούνται όλες οι παραπάνω ρουτίνες, προκειμένου να εκτελέσει το όχημα τις αντίστοιχες κινήσεις. Οι κλήσεις πραγματοποιούνται μέσα στη *setup* ώστε η ακολουθία των κινήσεων να εκτελεστεί μόνο μία φορά (Εικόνα 114).



Εικόνα 114

Μελετώντας το πρόγραμμα, διαπιστώνουμε ότι μετά από την κλήση κάθε υπορουτίνας που υλοποιεί μία κίνηση, υπάρχει μία εντολή καθυστέρησης *delay_MILLIS*. Η καθυστέρηση αυτή, καθορίζει στην ουσία τη διάρκεια της κίνησης. Επιπλέον, παρατηρούμε ότι σε κάθε αλλαγή κίνησης (π.χ. από στροφή δεξιά σε στροφή αριστερά) παρεμβάλλεται ένα σύντομο σταμάτημα (κλήση της υπορουτίνας *stop_car*, με καθυστέρηση 100ms). Αυτό γίνεται, επειδή η άμεση αλλαγή της φοράς περιστροφής ενός μοτέρ, μπορεί να δημιουργήσει πρόβλημα τόσο στο ίδιο το μοτέρ, όσο και στη λειτουργία του οχήματος.

Το πρόγραμμα για τη λειτουργία της εφαρμογής υπάρχει έτοιμο στο συνοδευτικό αρχείο με όνομα *motor_shield_demo.abp*. Για να το ανοίξετε, μέσα από το ArduBlock πατήστε το κουμπί *Άνοιγμα*, εντοπίστε το αρχείο στη θέση όπου το έχετε αποθηκεύσει και πατήστε *Open*.

Από το περιβάλλον του ArduBlock πατάμε το κουμπί *Ανέβασε στο Arduino*. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα. Αφαιρούμε το καλώδιο USB, ενεργοποιούμε το όχημα πατώντας το διακόπτη και παρακολουθούμε την κίνησή του.

15.3 Πρόγραμμα για υλοποίηση με L293D chip

Από το μενού *Εργαλεία* του Arduino IDE, επιλέγουμε *ArduBlock* και ανοίγει το περιβάλλον οπτικού προγραμματισμού.

Για την κίνηση του οχήματος με τη χρήση του L293D chip, θα αναπτύξουμε δύο ομάδες υπορουτίνων. Η πρώτη ομάδα, θα περιλαμβάνει τις υπορουτίνες για την κίνηση του κάθε μοτέρ ξεχωριστά. Υπενθυμίζουμε ότι για τον έλεγχο του δεξιού μοτέρ, χρησιμοποιούνται οι ψηφιακοί ακροδέκτες 3 (Enable, ταχύτητα περιστροφής), 4 (σήμα ελέγχου In 1) και 5 (σήμα ελέγχου In 2) του Arduino, ενώ για τον έλεγχο του αριστερού, οι ακροδέκτες 9 (Enable, ταχύτητα περιστροφής), 10 (σήμα ελέγχου In 3) και 11 (σήμα ελέγχου In 4). Οι υπορουτίνες της δεύτερης ομάδας, θα αξιοποιούν αυτές της πρώτης, προκειμένου να υλοποιήσουν τις διάφορες κινήσεις του οχήματος.

Για να δημιουργήσουμε τις υπορουτίνες που θέλουμε, από την ομάδα εντολών *Έλεγχος*, παίρνουμε το πλακίδιο *υπορουτίνα Commands*. Κάνοντας κλικ στη λέξη *υπορουτίνα*, την αντικαθιστούμε με το όνομα που θέλουμε να έχει η υπορουτίνα μας και

στη συνέχεια, κουμπώνουμε στο πλακίδιο τις διάφορες εντολές που θέλουμε να εκτελούνται κατά την κλήση της.

Αρχικά, υλοποιούμε τις υπορουτίνες *right_forward*, *right_backward* και *right_stop*, οι οποίες επιτρέπουν στο δεξιό μοτέρ να κινηθεί προς τα εμπρός, προς τα πίσω και να σταματήσει (Εικόνα 115). Σε κάθε υπορουτίνα, θέτουμε στις ακίδες 4 και 5, που συνδέονται με τους ακροδέκτες In 1 και In 2 του L293D, τις τιμές που αναφέρονται στον πίνακα λειτουργίας του chip.



Εικόνα 115

Αντίστοιχα, αναπτύσσουμε και τις υπορουτίνες *left_forward*, *left_backward* και *left_stop*, για τον έλεγχο του αριστερού μοτέρ (Εικόνα 116).



Εικόνα 116

Επιπλέον, δημιουργούμε την υπορουτίνα *enable_motors*, η οποία ενεργοποιεί τα δύο μοτέρ, κάνοντας HIGH τις ψηφιακές ακίδες 3 και 9, που ελέγχουν την ενεργοποίηση και απενεργοποίηση των δύο μοτέρ (ακροδέκτες En 1 και En 2 του L293D)(Εικόνα 117).



Εικόνα 117

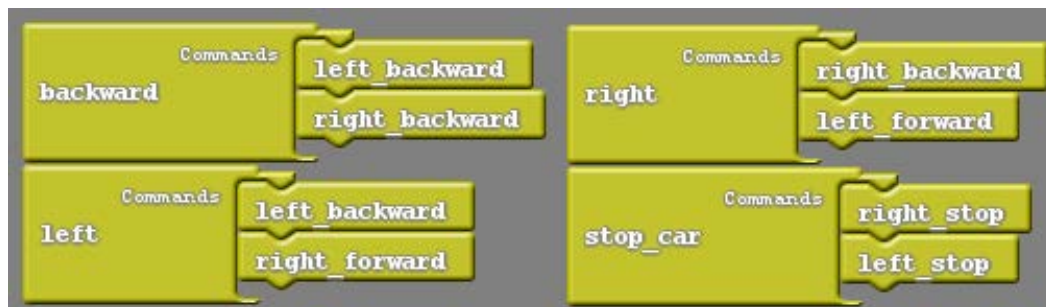
Έχοντας σαν βάση τις υπορουτίνες για την κίνηση των μοτέρ, προχωράμε σε εκείνες που θα υλοποιούν τη συνολική κίνηση του οχήματος. Η κλήση μίας υπορουτίνας που έχουμε ορίσει νωρίτερα, γίνεται στο ArduBlock με το πλακίδιο *υπορουτίνα* (ομάδα εντολών *Έλεγχος*). Κάνοντας κλικ στη λέξη υπορουτίνα, την αντικαθιστούμε με το όνομα της συνάρτησης που θέλουμε να καλέσουμε.

Πρώτα δημιουργούμε την υπορουτίνα *forward* που κινεί το όχημα προς τα εμπρός (Εικόνα 118).



Εικόνα 118

Στη συνέχεια, με ανάλογο τρόπο, δημιουργούμε τις υπορουτίνες *backward*, *left*, *right* και *stop_car*, που επιτρέπουν στο όχημα να κινείται προς τα πίσω, να στρίβει αριστερά, δεξιά και να σταματάει αντίστοιχα (Εικόνα 119).



Εικόνα 119

Στο κυρίως πρόγραμμα, αξιοποιούνται οι παραπάνω ρουτίνες, προκειμένου να εκτελέσει το όχημα τις αντίστοιχες κινήσεις. Οι αντίστοιχες κλήσεις, πραγματοποιούνται μέσα στη *setup*, ώστε η ακολουθία των κινήσεων να εκτελεστεί μόνο μία φορά (Εικόνα 120).



Εικόνα 120

Η πρώτη υπορουτίνα που καλείται είναι η *enable_motors*, η οποία ενεργοποιεί τα δύο μοτέρ. Μελετώντας το πρόγραμμα, διαπιστώνουμε ότι μετά από την κλήση κάθε υπορουτίνας που υλοποιεί μία κίνηση, υπάρχει μία εντολή καθυστέρησης *delay_MILLIS*. Η καθυστέρηση αυτή, καθορίζει στην ουσία τη διάρκεια της κίνησης. Επιπλέον, παρατηρούμε ότι σε κάθε αλλαγή κίνησης (π.χ. από στροφή δεξιά σε στροφή αριστερά) παρεμβάλλεται ένα σύντομο σταμάτημα (κλήση της υπορουτίνας *stop_car*, με καθυστέρηση 100ms). Αυτό γίνεται, επειδή η άμεση αλλαγή της φοράς περιστροφής ενός μοτέρ, μπορεί να δημιουργήσει πρόβλημα τόσο στο ίδιο το μοτέρ, όσο και στη λειτουργία του οχήματος.

Το πρόγραμμα για τη λειτουργία της εφαρμογής υπάρχει έτοιμο στο συνοδευτικό αρχείο με όνομα *L293D_demo.abp*. Για να το ανοίξετε, μέσα από το ArduBlock πατήστε το κουμπί *Ανοιγμα*, εντοπίστε το αρχείο στη θέση όπου το έχετε αποθηκεύσει και πατήστε *Open*.

Από το περιβάλλον του ArduBlock πατάμε το κουμπί *Ανέβασε στο Arduino*. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα. Αφαιρούμε το καλώδιο USB, ενεργοποιούμε το όχημα πατώντας το διακόπτη και παρακολουθούμε την κίνησή του.

Πριν κλείσουμε το ArduBlock, πατάμε το κουμπί *Αποθήκευση* ή *Αποθήκευσης* ως και αποθηκεύουμε το αρχείο .abr του προγράμματος στο δίσκο μας.

16 Προγραμματισμός για την αποφυγή εμποδίων

Σε αυτή τη δεύτερη φάση, επεκτείνουμε το πρόγραμμα που αναπτύξαμε νωρίτερα, έτσι ώστε να υλοποιήσουμε τη λειτουργία της αυτόνομης κίνησης του οχήματος, με δυνατότητα ανίχνευσης και αποφυγής εμποδίων.

16.1 Σύνδεση Arduino 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*. Ακολουθώντας, πάλι από το μενού *Εργαλεία*, πηγαίνουμε στο *Θύρα* και επιλέγουμε τη θύρα COM στην οποία έχει συνδεθεί το Arduino.

16.2 Διαδικασία που θα ακολουθηθεί

Από το μενού *Εργαλεία* του Arduino IDE, επιλέγουμε *ArduBlock* και ανοίγει το περιβάλλον οπτικού προγραμματισμού. Το πρόγραμμα που θα αναπτύξουμε, θα κάνει χρήση των συναρτήσεων που περιέχονται στο πρόγραμμα της ενότητας 15. Για το λόγο αυτό, θα ακολουθήσουμε την ακόλουθη διαδικασία.

Μέσα από το περιβάλλον του ArduBlock, πατάμε το κουμπί *Άνοιγμα*, εντοπίζουμε στο δίσκο το αρχείο με το πρόγραμμα που αναπτύχθηκε στην προηγούμενη ενότητα, το επιλέγουμε και πατάμε *Open*. Μόλις ανοίξει το πρόγραμμα, πατάμε το κουμπί *Αποθήκευση* ως και το αποθηκεύουμε με νέο όνομα, προκειμένου να μη χάσουμε την προηγούμενη έκδοση. Κάνουμε στο νέο αρχείο τις αλλαγές που περιγράφονται στη

συνέχεια, προκειμένου να υλοποιήσουμε τις λειτουργίες της ανίχνευσης και της αποφυγής εμποδίων.

16.3 Πρόγραμμα

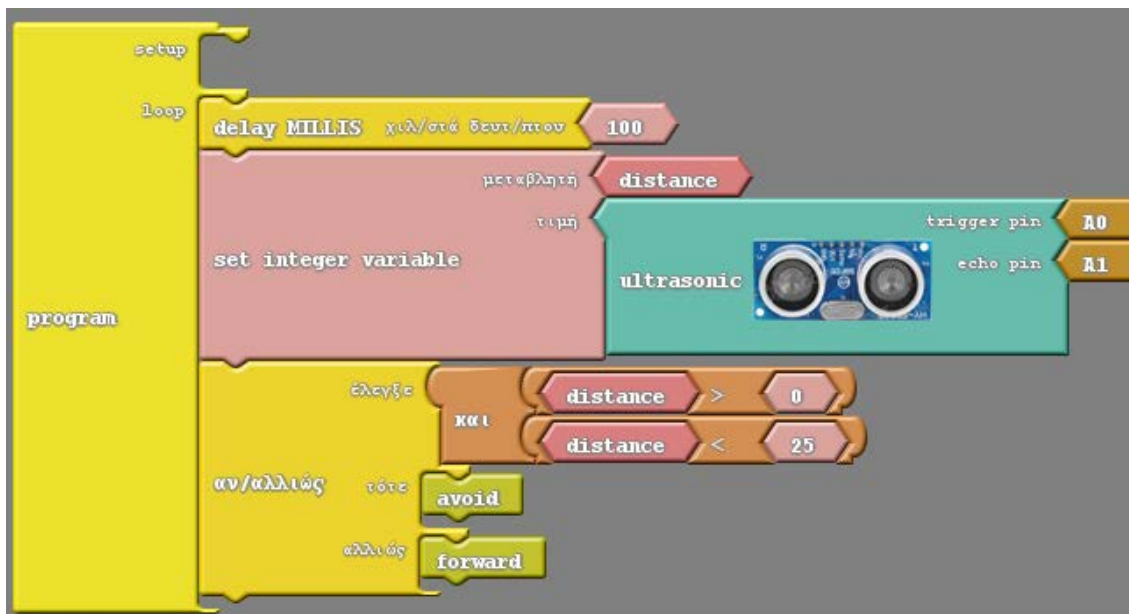
Στο υπάρχον πρόγραμμα, δημιουργούμε μία νέα υπορουτίνα με όνομα *avoid*, η οποία υλοποιεί έναν ελιγμό αποφυγής εμποδίου (Εικόνα 121).



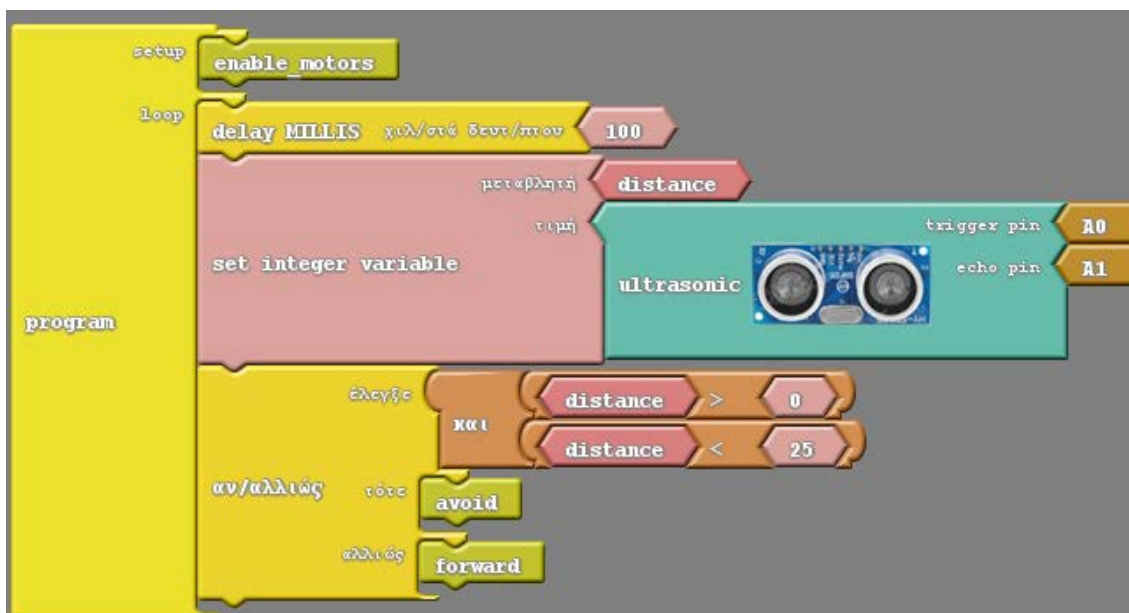
Εικόνα 121

Η *avoid*, χρησιμοποιεί τις προϋπάρχουσες υπορουτίνες, για να κάνει το όχημα να κινηθεί προς τα πίσω για μισό δευτερόλεπτο και στη συνέχεια να στρίψει προς τα αριστερά για 300 ms.

Στη συνέχεια, απομακρύνουμε τις εντολές που υπάρχουν στη setup (τις σύρουμε στην περιοχή με τις ομάδες πλακιδίων)



Εικόνα 122



Εικόνα 123

Αν έχετε κατασκευάσει το όχημα χρησιμοποιώντας το motor shield, μπορείτε να βρείτε το πρόγραμμα έτοιμο στο συνοδευτικό αρχείο με όνομα *motor_shield_avoid.abp*. Αν πάλι έχετε επιλέξει το L293D chip, τότε θα βρείτε το αντίστοιχο πρόγραμμα στο συνοδευτικό αρχείο *L293D_avoid.abp*. Σε κάθε περίπτωση, για να το ανοίξετε, μέσα από το ArduBlock πατήστε το κουμπί *Άνοιγμα*, εντοπίστε το αρχείο στη θέση όπου το έχετε αποθηκεύσει και πατήστε *Open*.

Από το περιβάλλον του ArduBlock πατάμε το κουμπί *Ανέβασε στο Arduino*. Με τον τρόπο αυτό το πρόγραμμα μεταφορτώνεται στην πλακέτα. Αφαιρούμε το καλώδιο USB, ενεργοποιούμε το όχημα πατώντας το διακόπτη και παρακολουθούμε την κίνησή του.

Πηγές

Κεντρικό Site Arduino (<https://www.arduino.cc/>). Ανακτήθηκε στις 10-1-2018

Site ArduBlock (<http://blog.ardublock.com/>). Ανακτήθηκε στις 10-1-2018

Κεντρικό Site Fritzing (<http://fritzing.org/home/>). Ανακτήθηκε στις 10-1-2018

<http://www.ti.com/lit/ds/symlink/l293.pdf> (Ανακτήθηκε στις 10-1-2018)

<https://playground.arduino.cc/Main/AdafruitMotorShield> (Ανακτήθηκε στις 10-1-2018)

Παπάζογλου, Π., Λιωνής Σ.Π. (2017). Ανάπτυξη Εφαρμογών με το Arduino, 2^η Έκδοση. Θεσσαλονίκη: Τζιόλας

TOOLS προγραμμάτων για τον παρόν Οδηγό Εκπαιδευτικού:

<https://drive.google.com/open?id=1JtvwW1OWTJ-s7ptpnNOCz5LGXhrYJKIL>

<http://www.ti.com/lit/ds/symlink/l293.pdf>

<https://playground.arduino.cc/Main/AdafruitMotorShield>

Η σύνταξη του οδηγού έγινε
από τον Σπύρο Πολυχρόνη Λιωνή
και η επιμέλεια του
από την Ανδρικοπούλου Μαριλένα

Εκ μέρους του μη κερδοσκοπικού οργανισμού

SciCo – Επιστήμη Επικοινωνία

Για το πρόγραμμα “STEMpowering Youth”

Αθήνα, Ιανουάριος 2017