

Νικολάου Α. Φανουράκη

Εκπαιδευτική Ρομποτική με τον μικροελεγκτή Arduino.

Ένας πρακτικός οδηγός κατασκευών για μαθητές και εκπαιδευτικούς.

ISBN : 978-618-00-1548-5

Εμπεριέχει παραδείγματα βήμα προς βήμα, σχήματα με τα πλήρη ηλεκτρικά τους κυκλώματα, φωτογραφίες από τις κατασκευές και τον κώδικα προγραμματισμού.

Ασήμι Ηρακλείου, 2019

Educational robotics with Arduino.

Nikolaos A. Fanourakis

Assimi Heraklion Crete Greece, 2019

ISBN: 978-618-00-1548-5

Εκπαιδευτική Ρομποτική με τον μικροελεγκτή Arduino.

Ένας πρακτικός οδηγός κατασκευών για μαθητές και εκπαιδευτικούς.

ISBN : 978-618-00-1548-5

Αυτό το υλικό διατίθεται με άδεια Creative Commons Αναφορά Δημιουργού - Παρόμοια Διανομή 4.0 Διεθνές (CC BY-SA 4.0) .

(<http://creativecommons.org/licenses/by-sa/4.0/>)

Όλες οι φωτογραφίες, εκτός από αυτές που αναφέρεται ρητά η πηγή, έχουν τραβηχτεί από τον Νικόλαο Α. Φανουράκη.

Οι εικόνες των εξαρτημάτων των κυκλωμάτων δημιουργήθηκαν με τη βοήθεια του λογισμικού ανοικτού κώδικα (open source) Fritzing (<http://www.fritzing.org>).

Πρώτη ψηφιακή έκδοση: Οκτώβριος 2019, Αθήμι Ηρακλείου Κρήτης Τ.Κ. 70016.

Το έγγραφο σε ψηφιακή μορφή PDF διατίθεται στην ιστοσελίδα :

<https://fanourakisnikos.weebly.com>

Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές ευχαριστίες μου προς: τον Τζωρτζάκη Ιωάννη, Συντονιστή Εκπαιδευτικού Έργου ΠΕ81 (Πολιτικών Μηχανικών και Αρχιτεκτόνων), για την τιμή που μου έκανε να προλογίσει το βιβλίο και για τις χρήσιμες επισημάνσεις του, τον Χανιωτάκη Ευάγγελο (Ηλεκτρολόγο Μηχ. M.Sc.) για τις εύστοχες παρατηρήσεις του, το Δασκαλάκη Ιωάννη (Ηλεκτρολόγο Μηχανικό και Τεχν. Υπολογιστών) για την ανατροφοδότηση του πριν την έκδοση, τη Δωράκη Χρυσάνθη (Φιλολόγο M.A.) για την επιμέλεια των κειμένων του βιβλίου και ιδιαιτέρως την οικογένεια μου για την υπομονή και την αμέριστη συμπαράστασή της σε όλες μου τις πρωτοβουλίες.

Νικόλαος Α. Φανουράκης, M.Sc.

Εκπαιδευτικός κλάδου Π.Ε.83 (Ηλεκτρολόγων)

Πίνακας Περιεχομένων

Πρόλογος.....	7
Εισαγωγή.....	8
Κεφάλαιο 1	10
Υλισμικό και Λογισμικό (Hardware - Software).	10
1.1 Μικροελεγκτής	11
1.2 Γέφυρα τύπου-Η	15
1.3 Αισθητήρας υπερήχων	18
1.4 Οθόνη Χαρακτήρων	20
1.5 Ηλεκτρονόμος	22
1.6 Μικρόφωνο.....	24
1.7 Αισθητήρας Θερμοκρασίας-Υγρασίας.	26
1.8 Αισθητήρας Υγρασίας Εδάφους	28
1.9 Αντλία	30
1.10 Εξάρτημα ρόδας – κινητήρα.	32
1.11 Άλλα εξαρτήματα.....	33
1.12 Λογισμικό.....	37
Κεφάλαιο 2 - Κατασκευές.....	41
2.1 Ηχοδιακόπτης	42
2.2 Φασαριόμετρο.	49
2.3 Έλεγχος Θερμοκρασίας – Υγρασίας Θερμοκηπίου.	58
2.4 Αυτόματο Πότισμα Γλάστρας.....	66
2.5 Ρομπότ Αποφυγής Εμποδίων.	75

Παράρτημα	88
Κώδικες προγραμματισμού των κατασκευών.....	88
1. Ηχοδιακόπτης.....	88
2. Φασαριόμετρο	90
3. Έλεγχος Θερμοκρασίας – Υγρασίας Θερμοκηπίου.....	94
4. Αυτόματο Πότισμα Γλάστρας.....	96
5. Ρομπότ Αποφυγής Εμποδίων.....	99
Βιβλιογραφία και Δικτυογραφία.....	105

Πρόλογος

Μια έρευνα στα σύγχρονα εκπαιδευτικά συστήματα, εύκολα αποκαλύπτει κοινά σημεία που σχετίζονται κυρίως με τις μαθητοκεντρικές προσεγγίσεις στη διδασκαλία και την προσπάθεια η μάθηση να έχει νόημα για τους μαθητές και συνάφεια με πρακτικά ζητήματα της ζωής. Η εισαγωγή των ερευνητικών εργασιών και άλλων συναφών αντικειμένων που αξιοποιούν τη διερευνητική μάθηση, την επίλυση προβλημάτων και τη μέθοδο project, ήταν ένα πολύ θετικό βήμα του ελληνικού εκπαιδευτικού συστήματος προς αυτή την κατεύθυνση. Παράλληλα, πρωτοπόροι εκπαιδευτικοί, με δική τους πρωτοβουλία και κάποιες φορές, επεκτείνοντας τα όρια των αναλυτικών προγραμμάτων και των σχολικών βιβλίων, προχωρούν περισσότερο, εισάγοντας μοντέλα όπως αυτό του STEAM και αντικείμενα που κάποτε αποτελούσαν αρμοδιότητα λίγων εξειδικευμένων «μυστών».

Ένας τέτοιος εκπαιδευτικός είναι ο Νίκος Φανουράκης, με τον οποίο είχα την τύχη να συνεργαστώ, ως παιδαγωγικά υπεύθυνος στο ΕΚ Ηρακλείου, όπου διδάσκει και καινοτομεί. Το βιβλίο αυτό, πετυχαίνει κάτι πολύ σημαντικό με απλό τρόπο, καταφέρνει να δείξει όλα τα βήματα για τη δημιουργία προχωρημένων ηλεκτρονικών κατασκευών, αξιοποιώντας το δημοφιλέστατο Arduino, χωρίς να απευθύνεται σε προχωρημένους μαθητές και χωρίς να χρειάζεται να το εφαρμόσουν εκπαιδευτικοί ειδικοτήτων πληροφορικής ή ηλεκτρονικών. Αυτό το πετυχαίνει έχοντας ουσιαστικά καλύψει όλες τις πιθανές ερωτήσεις που θα μπορούσε ένας αρχάριος να θέσει, από το «που θα τα βρω αυτά» και «πώς θα τα συναρμολογήσω», μέχρι το «πώς θα γράψω τον κώδικα». Επιπλέον, όχι μόνο δίνει έτοιμο τον κώδικα για εγκατάσταση, αλλά επεξηγεί και τα επιμέρους σημεία του, επιτρέποντας έτσι να ασχοληθούν δημιουργικά από τους πλέον αρχάριους έως τους πιο προχωρημένους μαθητές.

Συνοψίζοντας, πρόκειται για ένα πολύ χρήσιμο πρακτικό οδηγό δημιουργίας ρομποτικών ηλεκτρονικών κατασκευών που διανέμεται ελεύθερα και μπορεί να χρησιμοποιηθεί από μαθητές δημοτικού μέχρι και Επαγγελματικού Λυκείου και να διευκολύνει τους εκπαιδευτικούς χωρίς να απαιτεί από αυτούς εξειδικευμένες γνώσεις.

Γιάννης Τζωρτζάκης

Συντονιστής Εκπαιδευτικού Έργου ΠΕ81
(Πολιτικών Μηχανικών & Αρχιτεκτόνων)

Εισαγωγή

Το εγχειρίδιο αυτό απευθύνεται σε μαθητές και εκπαιδευτικούς όλων των βαθμίδων αλλά και σε αρχάριους, χομπίστες, λάτρεις των ηλεκτρικών – ηλεκτρονικών κατασκευών. Η ελληνική βιβλιογραφία στο συγκεκριμένο αντικείμενο της Εκπαιδευτικής Ρομποτικής (ιδιαίτερα αναφορικά με τον μικροελεγκτή Arduino) είναι ιδιαίτερα ελλιπής οπότε ο εμπλουτισμός της κρίνεται απαραίτητος. Δεν έχει σκοπό να εισάγει τον αναγνώστη στον Προγραμματισμό ή στη Βασική Ηλεκτροτεχνία, καθώς υπάρχουν αρκετά και αξιόλογα εγχειρίδια γι' αυτό το σκοπό, αλλά φιλοδοξεί να δώσει τη δυνατότητα στον κάθε ενδιαφερόμενο να νιώσει την ικανοποίηση της ολοκλήρωσης και της λειτουργίας μιας κατασκευής.

Οι κατασκευές οι οποίες παρουσιάζονται είναι από τη φύση τους διαθεματικές, οπότε στο πλαίσιο της εκπαιδευτικής διαδικασίας πιθανόν να αποτελεί ένα χρήσιμο εργαλείο στα χέρια δασκάλων και καθηγητών οι οποίοι διδάσκουν αντικείμενα “STEM” (Science, Technology, Engineering and Mathematics), ακρωνύμιο που χρησιμοποιείται για τις Φυσικές Επιστήμες, την Τεχνολογία, την Επιστήμη των Μηχανικών και τα Μαθηματικά. Ακόμη, χρήσιμο μπορεί να φανεί σε αυτούς που διδάσκουν στην Τεχνική Επαγγελματική εκπαίδευση και συγκεκριμένα σε τομείς Ηλεκτρολογίας, Ηλεκτρονικής, Μηχανολογίας, Πληροφορικής κ.α. και αντιμετωπίζουν αυτά τα πεδία με την απαιτούμενη πλέον διαθεματικότητα.

Το βιβλίο αποτελείται από δυο κεφάλαια και ένα παράρτημα. Στο πρώτο κεφάλαιο καταγράφονται όλα τα υλικά που χρησιμοποιήθηκαν, όπως το Arduino Uno, διάφορα περιφερειακά εξαρτήματα, αγωγοί κ.α. Παρουσιάζονται η λειτουργία, η συνδεσμολογία, τα τεχνικά τους χαρακτηριστικά καθώς και οι φωτογραφίες των πραγματικών υλικών. Επίσης, για κάθε υλικό υπάρχει ένας πίνακας με το ενδεικτικό κόστος αγοράς του από ένα ελληνικό κατάστημα (τιμή αγοράς με τους φόρους) και συγκριτικά το κόστος αγοράς από διαδικτυακά καταστήματα του εξωτερικού (τιμή αγοράς με τους φόρους και τα μεταφορικά).

Στο δεύτερο κεφάλαιο παρουσιάζονται πέντε ολοκληρωμένες κατασκευές. Σε καθεμία υπάρχει μια εισαγωγική ενότητα με μια συνοπτική παρουσίαση και φωτογραφίες από την πραγματική κατασκευή. Στη συνέχεια, στη δεύτερη ενότητα, που ονομάζεται “Υλικά Κατασκευής”, απαριθμούνται τα απαραίτητα για την κατασκευή υλικά σε έναν πίνακα. Στην τρίτη ενότητα, η οποία αναφέρεται ως “Περιγραφή Κατασκευής”, δίνονται οδηγίες υλοποίησης του κυκλώματος βήμα προς βήμα (αυτές επιπλέον μπορούν να δοθούν στους μαθητές και ως φύλλο εργασίας). Η τέταρτη ενότητα, η

οποία αναφέρεται ως “Περιγραφή Λειτουργίας”, περιλαμβάνει μια σύντομη περιγραφή του κώδικα προγραμματισμού και μερικές παρατηρήσεις – προτάσεις αφενός για αλλαγές στις σταθερές και μεταβλητές που χρησιμοποιούνται, προκειμένου να βελτιωθεί η λειτουργία της κατασκευής στις εκάστοτε συνθήκες, και αφετέρου για την καλύτερη κατανόηση της συμπεριφοράς των αυτοματισμών αυτών. Τέλος, στην πέμπτη ενότητα υπάρχει παραπομπή για τον κώδικα του προγράμματος κάθε κατασκευής (ο οποίος παρατίθεται στο Παράρτημα και έχει σχόλια στα ελληνικά ώστε να είναι εύκολα κατανοητός). Ο κώδικας καθεμίας κατασκευής μπορεί να γραφεί με πολλούς τρόπους αλλά εδώ δίνεται μόνο ένας εύκολος τρόπος γραφής του διότι δημιουργήθηκε για εκπαιδευτικούς λόγους. Φυσικά μπορεί να αποτυπωθεί με πιο πολύπλοκο ή συμπαγή τρόπο ανάλογα με τις εκάστοτε απαιτήσεις.

Οι κατασκευές αυτές πρέπει να υλοποιούνται πάντα υπό την επίβλεψη του υπεύθυνου εκπαιδευτικού και να τηρούνται όλα τα απαραίτητα μέτρα ασφαλείας. Η παρουσίαση των κατασκευών έγινε με τη λογική του “πρωτοτύπου” και στην περίπτωση που θα ήθελε κάποιος αναγνώστης να προχωρήσει σε μόνιμες ολοκληρωμένες κατασκευές, απαιτούνται θήκες, διακόπτες, ασφάλειες, ασφαλειοθήκες κτλ. τα οποία είναι εύκολο να προμηθευτεί μαζί με τα υπόλοιπα υλικά.

Κεφάλαιο 1

Υλισμικό και Λογισμικό (Hardware - Software).

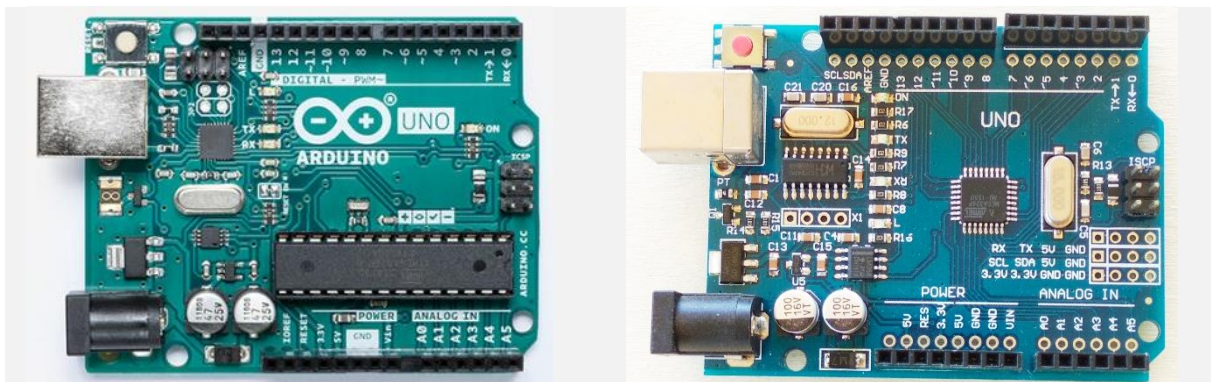
Η υλοποίηση των κατασκευών έγινε με όσο το δυνατόν πιο απλά υλικά, τα οποία μπορεί εύκολα να βρει ο κάθε ενδιαφερόμενος σε οποιοδήποτε κατάστημα ηλεκτρολογικού - ηλεκτρονικού υλικού ή σε κάποιο διαδικτυακό κατάστημα της Ελλάδας (τα οποία πλέον είναι αρκετά με ανταγωνιστικές τιμές και ικανοποιητικές διαθεσιμότητες) ή ακόμα και του εξωτερικού. Οι κατασκευές έχουν ως σκοπό την επίδειξη της συνδεσμολογίας του ηλεκτρικού κυκλώματος και της λειτουργίας τους και όχι την παρουσίαση ενός άρτιου οπτικά και λειτουργικά αποτελέσματος (π.χ. δε χρησιμοποιούνται κουτιά και διακόπτες on-off, τα οποία όμως μπορούν να προστεθούν εύκολα εκ των υστέρων). Έτσι τα υλικά των κατασκευών είναι κατά κανόνα τα φθηνότερα στην κατηγορία τους, χωρίς θήκες, βάσεις, τροφοδοτικά και περιττά εξαρτήματα, πράγμα που τα καθιστά πολύ προσιτά και οικονομικά μολονότι δεν υστερούν λειτουργικά για τις συγκεκριμένες εφαρμογές.

1.1 Μικροελεγκτής

Η αναπτυξιακή πλακέτα που χρησιμοποιείται είναι ένας κλώνος του Arduino Uno R3, ο οποίος διαθέτει 14 ψηφιακές εισόδους/εξόδους (6 από αυτές με δυνατότητα PWM), 6 αναλογικές εισόδους, ρολόι στα 16 MHz, σύνδεση USB, υποδοχή τροφοδοσίας, υποδοχή ICSP, ένα κρύσταλλο χαλαζία 16 MHz, μια σύνδεση USB, μια υποδοχή ρεύματος και ένα κουμπί επαναφοράς.

Περιέχει όλα όσα χρειάζονται για τη στήριξη του μικροελεγκτή, απλά θα πρέπει να συνδεθεί με έναν υπολογιστή με καλώδιο USB ή με ένα προσαρμογέα AC σε DC ή μπαταρία για να ξεκινήσει. Οι μικροελεγκτές Arduino έχουν γίνει διάσημοι για την ευκολία και την απλότητα χρήσης τους και σήμερα χρησιμοποιούνται ευρέως από απλές μαθητικές εφαρμογές μέχρι και εξελιγμένους ρομποτικούς μηχανισμούς σε όλο τον κόσμο.

Παρακάτω γίνεται μια σύντομη περιγραφή των Εισόδων και Εξόδων του Arduino Uno R3 πάνω στον οποίο στηρίχθηκαν οι κατασκευές.



Αριστερά : Η γνήσια αναπτυξιακή πλακέτα Arduino Uno r3.

Φωτογραφία : <https://store.arduino.cc/arduino-uno-rev3>

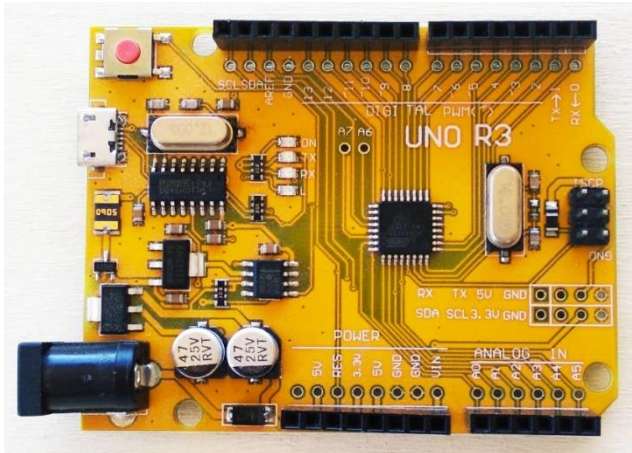
Δεξιά: Ένας από τους κλώνους του Arduino Uno r3.

Σχήμα 1.1.1

Ακροδέκτες Παροχής Ισχύος.

VIN. Η τάση εισόδου στην πλακέτα Arduino κατά τη χρήση μιας εξωτερικής πηγής ενέργειας (σε αντιδιαστολή των 5V από τη σύνδεση USB ή άλλες πηγή ενέργειας). Μπορεί να τροφοδοτηθεί με τάση μέσω αυτού του ακροδέκτη από 5V ως 20V.

5V. Αυτός ο ακροδέκτης εξάγει μια ρυθμιζόμενη 5V από τον ρυθμιστή της κεντρικής πλακέτας.



Ο κλώνος του Arduino Uno r3 που χρησιμοποιήθηκε στις κατασκευές.

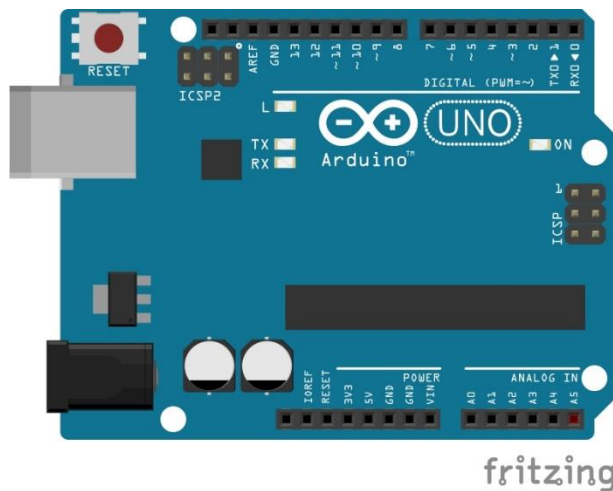
Σχήμα 1.1.2

3,3V. Αυτός ο ακροδέκτης εξάγει μια ρυθμιζόμενη 3,3V από τον ρυθμιστή της κεντρικής πλακέτας. Η μέγιστη κατανάλωση ρεύματος είναι 50 mA.

Είσοδοι και Έξοδοι.

Κάθε ένας από τους 14 ψηφιακούς ακροδέκτες και τους 6 αναλογικούς για το UNO μπορεί να χρησιμοποιηθεί ως είσοδος ή έξοδος εφαρμόζοντας τις εντολές `pinMode()`, `digitalWrite()`, και `digitalRead()`. Οι αναλογικοί ακροδέκτες δοκιμάζουν τα επίπεδα τάσης και αναφέρουν μια ψηφιακή τιμή χρησιμοποιώντας την εντολή `analogRead()`. Οι αναλογικοί ακροδέκτες και ένα υποσύνολο των ψηφιακών προσφέρουν Pulse Width Modulation (PWM) χρησιμοποιώντας την εντολή `analogWrite()`.

Κάθε ακροδέκτης λειτουργεί σε ηλεκτρική τάση 0-5V DC, μπορεί να παρέχει ή να λάβει ηλεκτρικό ρεύμα με ανώτατο όριο 40mA (συνιστάται 20mA) και έχει εσωτερική αντίσταση pull-up 20-50 kOhms.



Η αναπτυξιακή πλακέτα Arduino Uno R3 (σχήμα από το Fritzing).

Σχήμα 1.1.3

Εξειδικευμένες Λειτουργίες.

Κωδικός: Serial 0 (RX) και 1 (TX). Χρησιμοποιείται για τη λήψη (RX) και διαβίβαση σειριακών δεδομένων (TX) TTL.

Pins 0 και 1 είναι επίσης συνδεδεμένα με τα αντίστοιχα του Serial τσιπ USB-to-TTL.

Interrupt - Εξωτερικές διακοπές: Αυτοί οι ακροδέκτες μπορούν να ρυθμιστούν ώστε να προκαλέσουν μια διακοπή με τη λειτουργία `attachInterrupt`.

PWM: 3, 5, 6, 9, 10, 11 και A0 έως A5 παρέχουν 8-bit εξόδου PWM με τη λειτουργία `analogWrite()`.

LED: 13. Υπάρχει ένα ενσωματωμένο LED που συνδέεται με τον ψηφιακό ακροδέκτη 13. Το LED είναι αναμμένο με το 1, είναι κλειστό με το 0.

I2C/TWI: SDA και SCL, I2C υποστήριξη IIC, επικοινωνία TWI χρησιμοποιώντας τη βιβλιοθήκη Wire.

Αναλογικές Είσοδοι.

Το Uno έχει 6 αναλογικές εισόδους κάθε μία από τις οποίες παρέχει 10 bit (δηλαδή 1024 διαφορετικές τιμές). Εξ' ορισμού μετρούν από τη γείωση (0V) σε 5V, αλλά είναι δυνατόν να αλλάξει το άνω άκρο του εύρους τους χρησιμοποιώντας τον AREF και την εντολή `analogReference()`.

Όλοι οι αναλογικοί ακροδέκτες λειτουργούν και σαν αναλογικές έξοδοι χρησιμοποιώντας PWM. Επίσης, μπορούν να χρησιμοποιηθούν ως Digital εισοδοι ή έξοδοι.

Άλλες Είσοδοι και Έξοδοι.

AREF: Τάση αναφοράς για τις αναλογικές εισόδους. Χρησιμοποιείται με `analogReference ()`.

Reset: Με 0 επαναφέρει τον μικροελεγκτή.

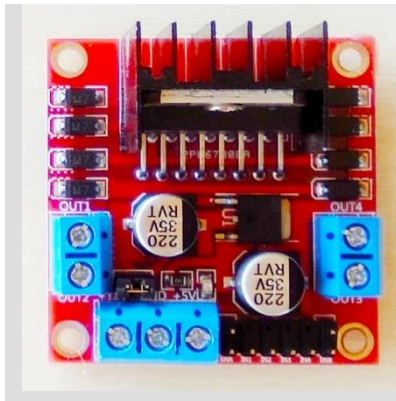
Τεχνικά Χαρακτηριστικά:

- Μικροελεγκτής: ATmega328P
- Τάση λειτουργίας: 5V
- Τάση εισόδου (συνιστώμενη): 7-12V
- Τάση εισόδου (όριο): 6-20V
- Ψηφιακοί I/O ακροδέκτες : 14 (6 εκ των οποίων παρέχουν έξοδο PWM)
- Αναλογικοί ακροδέκτες Εισόδου: 6
- DC ρεύμα ανά I/O ακροδέκτη: 20 mA
- DC ρεύμα για 3,3V ακροδέκτη: 50 mA
- Flash Memory: 32 KB εκ των οποίων 0.5 KB χρησιμοποιείται από τον bootloader
- SRAM: 2 KB
- EEPROM: 1 KB
- Clock Speed: 16 MHz
- Μήκος: 68,6 mm
- Πλάτος: 53,4 mm
- Βάρος: 25 g

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Γνήσια Πλακέτα Arduino Uno r3	21,50 € με Φ.Π.Α	€20.00 χωρίς φόρους και μεταφορικά.
Κλώνος Πλακέτα Arduino Uno r3	7,00 € με Φ.Π.Α	€ 4.00 με φόρους και μεταφορικά.
Πίνακας 1.1		

1.2 Γέφυρα τύπου-H

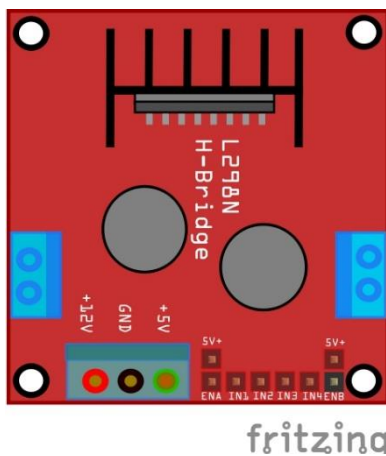
Προκειμένου να γίνει η οδήγηση των κινητήρων που δίνουν κίνηση στις ρόδες του ρομπότ της τελευταίας κατασκευής (βλ. Κεφ.2.5), χρησιμοποιείται το ολοκληρωμένο L298N Dual H-Bridge, που είναι μια διάταξη H-γέφυρας για έλεγχο ως 2 DC κινητήρων στέλνοντας PWM σήμα ταχύτητας και φοράς περιστροφής. Η γέφυρα λειτουργεί μέσω 6 σημάτων (IN1, IN2 και ENA για τον Α' κινητήρα και αντίστοιχα για τον Β') που παρέχονται από τον μικροελεγκτή. Η ενεργοποίηση/απενεργοποίηση της γέφυρας καθώς και η φορά περιστροφής για τον Α' κινητήρα καθορίζεται από τις τιμές των IN1 και IN2, ενώ η ταχύτητα ρυθμίζεται μέσω ενός PWM σήματος (duty cycle, δηλαδή ρύθμιση των κύκλων εργασίας οι οποίοι περιγράφουν το χρόνο επί τοις εκατό ενός ενεργού σήματος) στον ακροδέκτη ENA.



Η διάταξη H-γέφυρας που χρησιμοποιήθηκε, το ολοκληρωμένο L298N Dual H-Bridge.

Σχήμα 1.2.1

Αυτό σημαίνει πρακτικά ότι για 8 Bit έλεγχο έχουμε 256 επίπεδα ταχύτητας γράφοντας έναν αριθμό από 0 ως 255 στην εντολή `analogWrite(pin, value)`. Αν δοθεί τιμή 0, έχουμε 0VDC στους ακροδέκτες **OUT** της πλακέτας του, αν όμως δοθεί τιμή 255, έχουμε το μέγιστο της τάσης τροφοδοσίας η οποία εφαρμόζεται στον ακροδέκτη **+12V** (+7VDC ως +35VDC) να εφαρμόζεται ανάμεσα στους ακροδέκτες **OUT1 - OUT2**. Αντίστοιχα και ανεξάρτητα μπορεί να ρυθμιστεί η ταχύτητα και η φορά περιστροφής για τον Β' κινητήρα.



fritzing

Η γέφυρα που χρησιμοποιήθηκε, το ολοκληρωμένο L298N Dual H-Bridge. (σχήμα από το Fritzing) .

Σχήμα 1.2.2

Συνδεσμολογία:

- +12V - Τροφοδοσία των κινητήρων από +7VDC ως +35VDC.
- +5V - Τροφοδοσία 5V.
- IN1 - Είσοδος 1, σήμα ελέγχου Α' κινητήρα.
- IN2 - Είσοδος 2, σήμα ελέγχου Α' κινητήρα.
- IN3 - Είσοδος 3, σήμα ελέγχου Β' κινητήρα.

- IN4 - Είσοδος 3, σήμα ελέγχου Β' κινητήρα
- ENA - Είσοδος, ενεργοποίηση και έλεγχος ταχύτητας Α' κινητήρα.
- ENB - Είσοδος, ενεργοποίηση και έλεγχος ταχύτητας Β' κινητήρα.
- OUT1 - Έξοδος 1, πρώτο άκρο του Α' κινητήρα.
- OUT2 - Έξοδος 2, δεύτερο άκρο του Α' κινητήρα.
- OUT3 - Έξοδος 3, πρώτο άκρο του Β' κινητήρα.
- OUT4 - Έξοδος 4, δεύτερο άκρο του Β' κινητήρα.

Ο καθορισμός λειτουργίας του οδηγούμενου DC-κινητήρα με βάση τα σήματα ελέγχου της Η-γέφυρας L298N γίνονται βάσει των συνδυασμών των τιμών που φαίνονται στον παρακάτω πίνακα.

Καθορισμός λειτουργίας των οδηγούμενων DC κινητήρων, με βάση τα σήματα ελέγχου της Η-γέφυρας L298N.			
Λειτουργία	IN1	IN2	ENA
Ο κινητήρας περιστρέφεται δεξιά με ταχύτητα αυξανόμενη με το Duty Cycle του PWM.	1	0	PWM (8 bit)
Ο κινητήρας περιστρέφεται αριστερά με ταχύτητα αυξανόμενη με το Duty Cycle του PWM.	0	1	PWM (8 bit)
Δυναμική πέδηση του κινητήρα.	1	1	1
Δυναμική πέδηση του κινητήρα.	0	0	1
Απενεργοποίηση του κινητήρα.	---	---	0
Πίνακας 1.2.1			

Ιδιαίτερη προσοχή πρέπει να δοθεί στους βραχυκυκλωτήρες (jumpers) της πλακέτας καθώς μπορεί να λειτουργεί με διαφορετικές τάσεις τροφοδοσίας για τους κινητήρες καθώς και με δυο διαφορετικά είδη κινητήρων, δηλαδή με κινητήρες συνεχούς ρεύματος και βηματικούς κινητήρες.

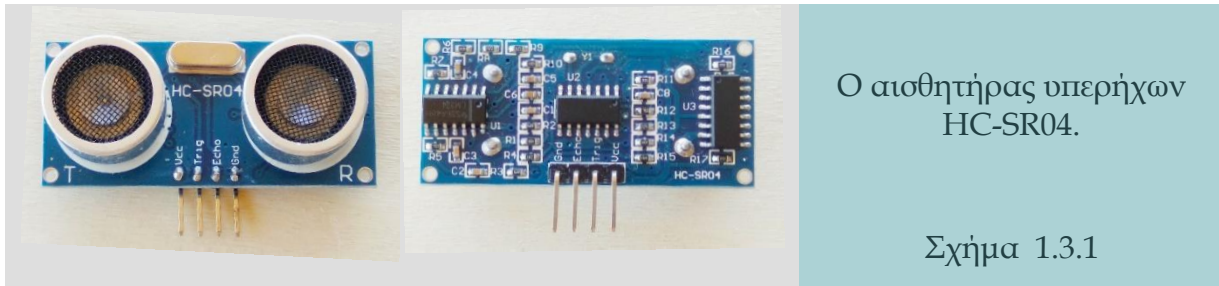
- Ο βραχυκυκλωτήρας στον ακροδέκτη **+12V** αφαιρείται, αν χρησιμοποιείται τάση μεγαλύτερη των +12V DC.
- Όταν είναι συνδεδεμένος ο βραχυκυκλωτήρας (jumper) **+12V**, ο ρυθμιστής τάσης της πλακέτας είναι ενεργοποιημένος και μπορεί να συνδεθεί τάση **+5V** DC στον αντίστοιχο ακροδέκτη.
- Οι δύο άλλοι βραχυκυκλωτήρες στους ακροδέκτες **ENA** και **ENB** αφαιρούνται, αν χρησιμοποιούνται κινητήρες συνεχούς ρεύματος και διατηρούνται για βηματικούς κινητήρες.

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Πλακέτα Η-γέφυρας L298N	€ 4,00 με Φ.Π.Α	€ 1,20 με φόρους και μεταφορικά.

Πίνακας 1.2.2

1.3 Αισθητήρας υπερήχων

Ο αισθητήρας υπερήχων HC-SR04 για τον υπολογισμό απόστασης μπορεί να υπολογίζει αποστάσεις από 2εκ. έως 400εκ. με ακρίβεια ενός περίπου εκατοστού του μέτρου. Η συνδεσμολογία του γίνεται με 4 ακροδέκτες και ενδείκνυται για χρήση με πλακέτα δοκιμών (Breadboard). Για να λειτουργήσει απαιτείται η τροφοδοσία του με 5V και γείωση στους ακροδέκτες Vcc και Gnd αντίστοιχα. Επιπλέον, έχουμε μια έξοδο (από το Uno) Trigger για την παραγωγή παλμών υπερήχων και μια είσοδο Echo για την ανίχνευση των παλμών αυτών.



Ο αισθητήρας υπερήχων
HC-SR04.

Σχήμα 1.3.1

Συνδεσμολογία (Σύνδεση 4 αγωγών):

- Vcc - Τροφοδοσία (5V).
- Gnd - Γείωση (0V).
- Trig - Παραγωγή υπερήχων (έξοδος).
- Echo - Ανίχνευση υπερήχων (είσοδος).



Ο αισθητήρας υπερήχων HC-SR04,
(σχήμα από το Fritzing).

Σχήμα 1.3.2

Η λειτουργία του στηρίζεται σε αυτή του Ραντάρ, δηλαδή ο ακροδέκτης ενεργοποίησης Trigger χρησιμοποιείται για την παραγωγή ενός ηχητικού κύματος (πομπός) και ο ακροδέκτης Echo (δέκτης) παρατηρεί το ανακλώμενο κύμα. Η χρονική διάρκεια που ταξιδεύει το κύμα στον αέρα εξαρτάται από την απόσταση μεταξύ του αισθητήρα και του αντικειμένου. Με τη μέτρηση της χρονικής διάρκειας που ταξιδεύει ένα κύμα και δεδομένου ότι η ταχύτητα του ήχου σε ξηρό αέρα στους 20°C είναι 343 μέτρα ανά δευτερόλεπτο, η απόσταση υπολογίζεται με τον ακόλουθο τύπο:

$$\text{Απόσταση} = (\text{Ταχύτητα Ήχου} \times \text{Χρόνος}) / 2$$

Για τις εφαρμογές που χρησιμοποιείται αυτός ο αισθητήρας πρέπει να ληφθεί υπόψη ότι τα επίπεδα αντικείμενα μπορούν να ανακλούν καλύτερα τα ηχητικά κύματα και να ανιχνεύονται ευκολότερα ενώ τα αιχμηρά και μικρά αντικείμενα ή τα αντικείμενα που εκτρέπουν τα ηχητικά κύματα σε άλλη κατεύθυνση από εκείνη του αισθητήρα, δεν μπορούν να ανιχνευθούν επαρκώς.

Τεχνικά Χαρακτηριστικά:

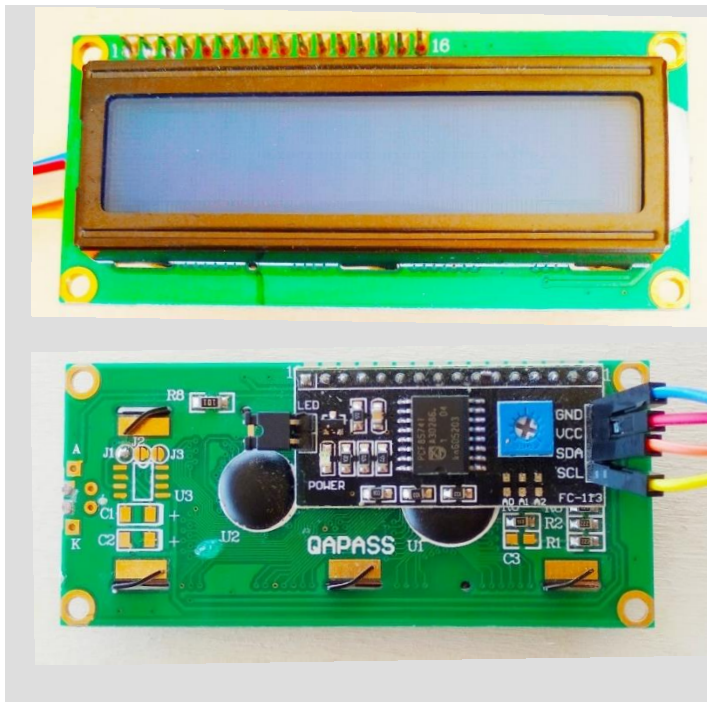
- Τύπος Αισθητήρα : Απόστασης
- Τάση εργασίας : 5V DC
- Ρεύμα Λειτουργίας : 15mA
- Ακτίνα : 2 - 400cm
- Φιλικό για Breadboard

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Αισθητήρας υπερήχων HC-SR04	€ 1,90 με Φ.Π.Α	€ 1.05 με φόρους και μεταφορικά.
Πίνακας 1.3.1		

1.4 Οθόνη Χαρακτήρων

Η οθόνη που χρησιμοποιείται στην τρίτη κατασκευή, προκειμένου να δίνει πληροφορίες για τη θερμοκρασία και την υγρασία στον χρήστη, είναι τύπου υγρών κρυστάλλων με οπίσθιο φωτισμό (backlight) και διαθέτει 2 γραμμές και 16 στήλες. Οι πληροφορίες που δίνονται στον χρήστη στη συγκεκριμένη κατασκευή είναι: ένα μήνυμα καλωσορίσματος και στη συνέχεια οι ενδιαφερόμενες τιμές των μεταβλητών με τις μονάδες μέτρησης.

Η σύνδεση της οθόνης γίνεται με τους ακροδέκτες του μικροελεγκτή SDA, SCL για την παροχή δεδομένων και 5V, GROUND για την παροχή ισχύος, όπως καταγράφεται παρακάτω:



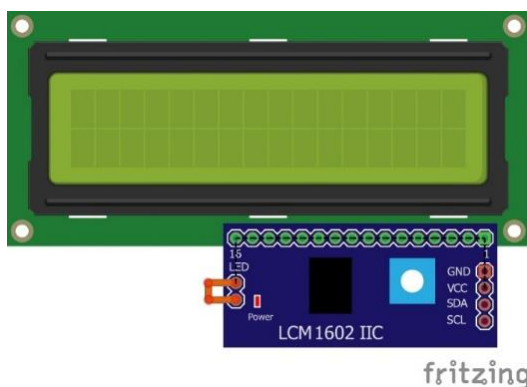
Οθόνη Χαρακτήρων
16 στηλών επί 2 γραμμών.

16x02 character lcd module.

Σχήμα 1.4.1

Συνδεσμολογία (Σύνδεση 4 αγωγών):

- Vcc - Τροφοδοσία 5V.
- Gnd - Γείωση (0V).
- Arduino SDA - Πλακέτα οθόνης SDA.
- Arduino SCL - Πλακέτα οθόνης SCL.



Οθόνη Χαρακτήρων 1602,
(σχήμα από το Fritzing).

Σχήμα 1.4.2

Τεχνικά Χαρακτηριστικά:

- Τύπος : Υγρών κρυστάλλων
- Μέγεθος : 2 γραμμές και 16 στήλες
- Τάση εργασίας : 5V DC
- Οπίσθιος φωτισμός : Γαλάζιος με δυνατότητα απενεργοποίησης
- Αντίθεση (Contrast) : Ρυθμιζόμενη από ποτενσιόμετρο
- I2C διεύθυνση : εύρος 0x20 ως 0x27 (προεπιλογή = 0x27)
- Διαστάσεις πλακέτας 80mm x 36mm x 20 mm
- Διαστάσεις οθόνης 66mm x 16mm

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Οθόνη Χαρακτήρων 1602	€ 5,40 με Φ.Π.Α	€ 3,20 με φόρους και μεταφορικά.
Πίνακας 1.4.1		

Η επιλογή ενεργοποίησης του οπίσθιου φωτισμού γίνεται με την προεπιλεγμένη διασύνδεση στο πίσω μέρος της πλακέτας και την εντολή `blacklight()` στον κώδικα του προγράμματος.

Επίσης, απαιτείται η χρήση της βιβλιοθήκης "LiquidCrystal_I2C" καθώς απλοποιεί σημαντικά τη διαδικασία προγραμματισμού.

1.5 Ηλεκτρονόμος

Η χρήση φορτίων με απαιτήσεις σε ηλεκτρικό ρεύμα και τάση μεγαλύτερων από τις δυνατότητες της πλακέτας του μικροελεγκτή, όπως ο ανεμιστήρας και η αντλία νερού στις κατασκευές που παρουσιάζονται, καθιστά αναγκαία τη χρήση ηλεκτρονόμων (ρελέ) για τον έλεγχο τους, καθώς το Arduino Uno μπορεί να παρέχει από τους

ακροδέκτες του έως 5V τάση και 20mA ρεύμα. Στην παρακάτω εικόνα είναι η πλακέτα με τον ηλεκτρονόμο που χρησιμοποιήθηκε, η οποία μπορεί να ελέγχει διάφορες συσκευές με μεγάλη τάση (έως AC250V/DC30V) και ισχυρό ρεύμα (έως AC 16A).



Η πλακέτα ηλεκτρονόμου που χρησιμοποιήθηκε, έως AC250V/DC30V και 16A.

Σχήμα 1.5.1

Η πλακέτα αυτή μπορεί να ελέγχεται άμεσα από κάποιον μικροελεγκτή, περιέχει 1 ηλεκτρονόμο (υπάρχουν και με περισσότερους όπως: 2, 4, 8, 16 κτλ. ώστε να καλύπτουν κάθε εφαρμογή) με πηνίο το οποίο ελέγχει μια μεταγωγική επαφή, δηλαδή μπορεί να έχει λειτουργία Κανονικά Κλειστή και Κανονικά Ανοιχτή (N.C./N.O.) για κάθε επαφή.

Συνδεσμολογία (Σύνδεση 3 αγωγών):

- VCC : 3,3V-5V (θετικός πόλος).
- GND : Γείωση (αρνητικός πόλος).
- IN1 : Έλεγχος του πηνίου (Οπλίζει με LOW).



Πλακέτα 1 ηλεκτρονόμου, (σχήμα από το Fritzing).

Σχήμα 1.5.2

Τεχνικά Χαρακτηριστικά:

- Ονομαστική Τάση Επαφής : έως AC 250V/DC 30V
- Πηνίο ισχύος : 0.1A - 0.5W
- Ονομαστικό Ρεύμα : έως 16A

- Ρεύμα Λειτουργίας : 0.1A
- Τάσης Λειτουργίας Πηγίου : 5V
- Λειτουργία Επαφής : NC/NO
- Οπλίζει με LOW (ένδειξη ON), αφοπλίζει με HIGH (ένδειξη OFF).
- Μέγιστη ισχύς : DC 30V - 10A , AC 250V - 10A
- Παρέχεται λυχνία LED για ένδειξη της κατάστασης εξόδου.

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Πλακέτα 1 ηλεκτρονόμου	€ 2,40 με Φ.Π.Α	€ 1,00 με φόρους και μεταφορικά.
Πίνακας 1.5.1		

1.6 Μικρόφωνο

Το μικρόφωνο που χρησιμοποιείται στις κατασκευές 1 και 2 είναι το MAX9812, το οποίο μπορεί να ανιχνεύει το επίπεδο θορύβου του περιβάλλοντος χώρου και να στέλνει μια αναλογική τάση στην έξοδό του.



Μικρόφωνο MAX9812.

Σχήμα 1.6.1

Το μικρόφωνο αυτό έχει τρεις ακροδέκτες: την τάση τροφοδοσίας, που μπορεί να είναι 3,3V και 5V, τη γείωση και την έξοδο (σήμα), η οποία είναι μια τάση έως 5V ανάλογα με το επίπεδο του θορύβου που ανιχνεύει.

Συνδεσμολογία (Σύνδεση 3 αγωγών):

- VCC : 3,3V-5V (θετικός πόλος).
- GND : Γείωση (αρνητικός πόλος).
- OUT : Σήμα εξόδου.



Μικρόφωνο MAX9812,
(σχήμα από το Fritzing).

Σχήμα 1.6.2

Τεχνικά Χαρακτηριστικά:

- Τάση λειτουργίας : 3,3 V και 5V
- Τάση εξόδου : 3-5V
- Υψηλή ευαισθησία.
- Κέρδος : 20dB
- Διαστάσεις : 20x11mm
- Φιλικό για χρήση με πλακέτα δοκιμών (Breadboard).

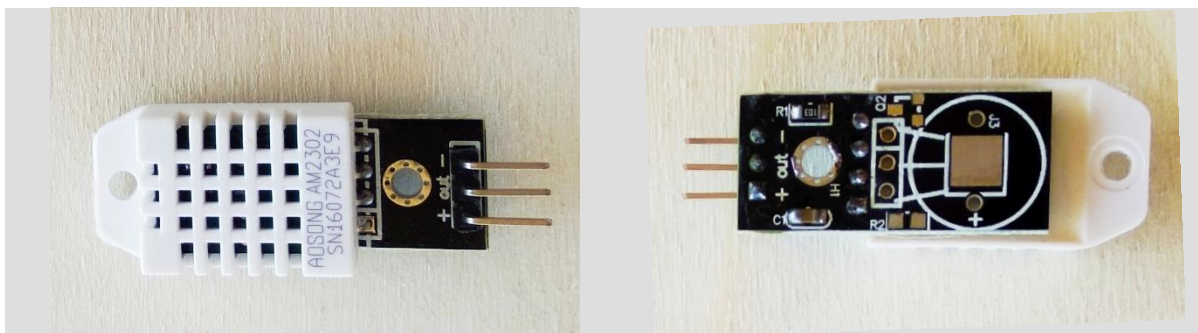
Ενδεικτικό Κόστος Υλικού.

Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Μικρόφωνο MAX9812	€ 3,30 με Φ.Π.Α	€ 1,70 με φόρους και μεταφορικά.

Πίνακας 1.6.1

1.7 Αισθητήρας Θερμοκρασίας-Υγρασίας.

Ο Αισθητήρας Θερμοκρασίας-Υγρασίας DHT22 (AM2302) είναι μια πλακέτα με ένα στοιχείο ανίχνευσης υγρασίας, έναν αισθητήρα θερμοκρασίας NTC (θερμίστορ) και ένα ολοκληρωμένο κύκλωμα (IC - Integrated Circuit) στο πίσω μέρος της. Για τη μέτρηση της υγρασίας χρησιμοποιείται ένα στοιχείο ανίχνευσης υγρασίας το οποίο έχει δύο ηλεκτρόδια με ένα υπόστρωμα συγκράτησης υγρασίας μεταξύ τους. Καθώς η υγρασία μεταβάλλεται, η αντίσταση μεταξύ αυτών των ηλεκτροδίων αλλάζει και η αλλαγή αυτή στην αντίσταση μετριέται και επεξεργάζεται από το ολοκληρωμένο κύκλωμα σε μορφή αναγνώσιμη από τον μικροελεγκτή. Για τη μέτρηση της θερμοκρασίας η πλακέτα χρησιμοποιεί έναν αισθητήρα θερμοκρασίας NTC (θερμίστορ) ο οποίος είναι ένας μεταβλητός αντιστάτης που αλλάζει την αντίσταση του όταν μεταβάλλεται η θερμοκρασία. Ο όρος "NTC" σημαίνει "αρνητικός συντελεστής θερμοκρασίας", ότι δηλαδή η αντίσταση μειώνεται με την αύξηση της θερμοκρασίας. Οι αισθητήρες τύπου DHTxx χρησιμοποιούν το δικό τους πρωτόκολλο για τη μεταφορά των δεδομένων και απαιτείται ακριβής χρονισμός, διαδικασία αρκετά πολύπλοκη η οποία όμως, όταν χρησιμοποιείται η βιβλιοθήκη "DHT", απλοποιείται.



Αισθητήρας Θερμοκρασίας-Υγρασίας DHT22.

Σχήμα 1.7.1

Συνδεσμολογία (Σύνδεση 3 αγωγών):

- VCC : 3,3V-5V (θετικός πόλος).
- GND : Γείωση (αρνητικός πόλος).
- OUT : Σήμα εξόδου.



Αισθητήρας Θερμοκρασίας -
Υγρασίας DHT22,
(σχήμα από το Fritzing).

Σχήμα 1.7.2

Τεχνικά Χαρακτηριστικά:

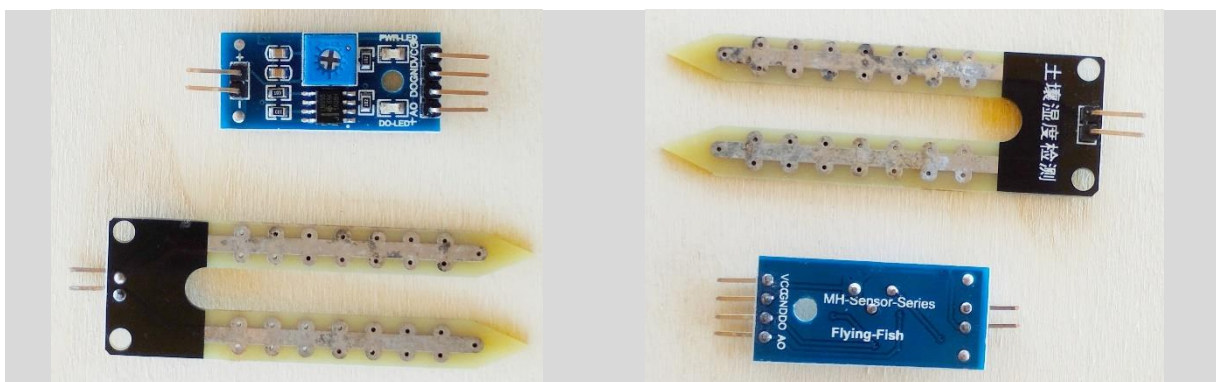
- Τάση λειτουργίας : 3,3 V - 5V
- Τάση εξόδου : 3-5V
- Ρεύμα λειτουργίας : 1,5mA
- Εύρος μέτρησης υγρασίας : 0-100% RH
- Εύρος μέτρησης θερμοκρασίας : -40 ως +80 °C
- Ακρίβεια μέτρησης υγρασίας : +-2% RH
- Ακρίβεια μέτρησης θερμοκρασίας : +-0,5 °C
- Φιλικό για χρήση με πλακέτα δοκιμών (Breadboard).

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Αισθητήρας Θερμοκρασίας - Υγρασίας DHT22	€ 6,50 με Φ.Π.Α	€ 0,85 με φόρους και μεταφορικά.
Πίνακας 1.7.1		

1.8 Αισθητήρας Υγρασίας Εδάφους

Ο αισθητήρας υγρασίας εδάφους FC-28 είναι μια πλακέτα η οποία μπορεί να χρησιμοποιηθεί για την ανίχνευση της υγρασίας του εδάφους και αποτελείται από δύο ανιχνευτές (ηλεκτρόδια) σε σχήμα δίσχολου. Αυτοί εφαρμόζουν μια τάση και επιτρέπουν σε ένα μικρό ρεύμα να διέλθει μέσω του εδάφους το οποίο δίνει την τιμή αντίστασης του. Με αυτόν τον τρόπο μετριέται έμμεσα η τιμή υγρασίας του εδάφους. Όταν υπάρχει νερό στο χώμα, το ρεύμα που διέρχεται ανάμεσα στους δυο ανιχνευτές (δίσχαλο) θα είναι περισσότερο, πράγμα που σημαίνει ότι υπάρχει λιγότερη αντίσταση. Το ξηρό έδαφος δεν επιτρέπει εύκολα στο ηλεκτρικό ρεύμα να περάσει, επομένως όταν η υγρασία είναι λιγότερη, το έδαφος έχει μεγαλύτερη αντίσταση. Όταν λοιπόν η τιμή υγρασίας εδάφους είναι μικρή, ο αισθητήρας στέλνει από την αναλογική έξοδο του (ΑΟ) μεγάλη τιμή σήματος (για τα 10Bit της αναλογικής εισόδου το μέγιστο είναι 1023) ενώ όταν η τιμή υγρασίας εδάφους είναι μεγάλη, ο αισθητήρας στέλνει μικρή τιμή σήματος με ελάχιστο το μηδέν.

Ο αισθητήρας υγρασίας εδάφους (FC-28) όμως παρέχει και μια δεύτερη ψηφιακή έξοδο (DO) η οποία δείχνει μια τιμή 0V ή 5V (LOW ή HIGH) και αλλάζει όταν η υγρασία του εδάφους υπερβαίνει μια προκαθορισμένη τιμή κατωφλίου. Η τιμή κατωφλίου ρυθμίζεται από το γαλάζιο ποτενσιόμετρο δίνοντας το πλεονέκτημα να οδηγήσει απευθείας ένα φορτίο, όπως έναν ηλεκτρονόμο, έναν βομβητή κ.α. Σίγουρα, η αναλογική (ΑΟ) έξοδος (ως 5V) σε συνδυασμό με μια αναλογική είσοδο του μικροελεγκτή μπορεί να δώσει περισσότερη ακρίβεια στις μετρήσεις από ότι η ψηφιακή έξοδος.



Ο Αισθητήρας Υγρασίας Εδάφους FC-28.

Σχήμα 1.8.1

Συνδεσμολογία (Σύνδεση 3 ή 4 αγωγών):

- VCC : 3,3V - 5V (θετικός πόλος).
- GND : Γείωση (αρνητικός πόλος).
- DO : ψηφιακή έξοδος (LOW ή HIGH).
- AO : αναλογική έξοδος (ως 5V).



Αισθητήρας Υγρασίας
Εδάφους FC-28,
(σχήμα από το Fritzing).

Σχήμα 1.8.2

Τεχνικά Χαρακτηριστικά:

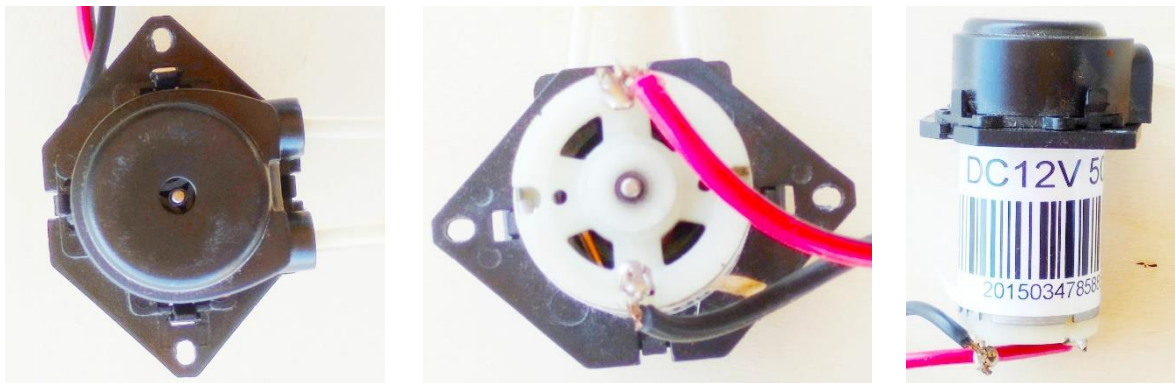
- Τάση λειτουργίας : 3,3 V - 5V
- Chip : LM393
- Μέγεθος πλακέτας (PCB) : 3,2cm x 1,4cm
- Ρυθμιζόμενη από ποτενσιόμετρο ευαισθησία της ψηφιακής εξόδου.
- Φιλικό για χρήση με πλακέτα δοκιμών (Breadboard).

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Αισθητήρας Υγρασίας Εδάφους FC-28.	€ 2,50 με Φ.Π.Α	€ 0,80 με φόρους και μεταφορικά.

Πίνακας 1.8.1

1.9 Αντλία

Η αντλία της κατασκευής είναι περισταλτική, γνωστή και ως αντλία ελαστικού σωλήνα. Αξιοποιεί την αρχή της περισταλτικής κίνησης για να αντλήσει ρευστά μέσω ενός ελαστικού σωλήνα, που εφαρμόζει γύρω από ένα δρομέα με ράουλα. Αυτός όταν περιστρέφεται, πιέζει ένα τμήμα του ελαστικού σωλήνα. Η πίεση στο συγκεκριμένο τμήμα απελευθερώνεται καθώς ο ρότορας περιστρέφεται και ο ελαστικός σωλήνας ανακτά το αρχικό του σχήμα. Έτσι, δημιουργείται στιγμιαία μια υποπίεση η οποία προκαλεί επαναπλήρωση του ελαστικού σωλήνα με υγρό.



Η περισταλτική αντλία που χρησιμοποιήθηκε στην εφαρμογή.

Σχήμα 1.9.1

Τα σημαντικότερα πλεονεκτήματα της είναι:

- ο Λειτουργεί στεγνή χωρίς να φθείρεται.
- ο Δεν χρειάζεται στεγανοποίηση.
- ο Αντιστρεφόμενη φορά περιστροφής.
- ο Είναι ανθεκτική στην φθορά.
- ο Έχει μεγάλη ακρίβεια δοσομέτρησης.
- ο Είναι ανθεκτική στην διάβρωση.
- ο Είναι αθόρυβη.
- ο Είναι εύκολη στον καθαρισμό.

Η αντλία αυτή χρησιμοποιείται ευρέως σε γαλακτοκομικά προϊόντα, κρασιά, σάλτσες, μαρμελάδες, κονιάματα, χρώματα, κόλλες, απορρυπαντικά, οξέα και άλλα.

Τεχνικά Χαρακτηριστικά:

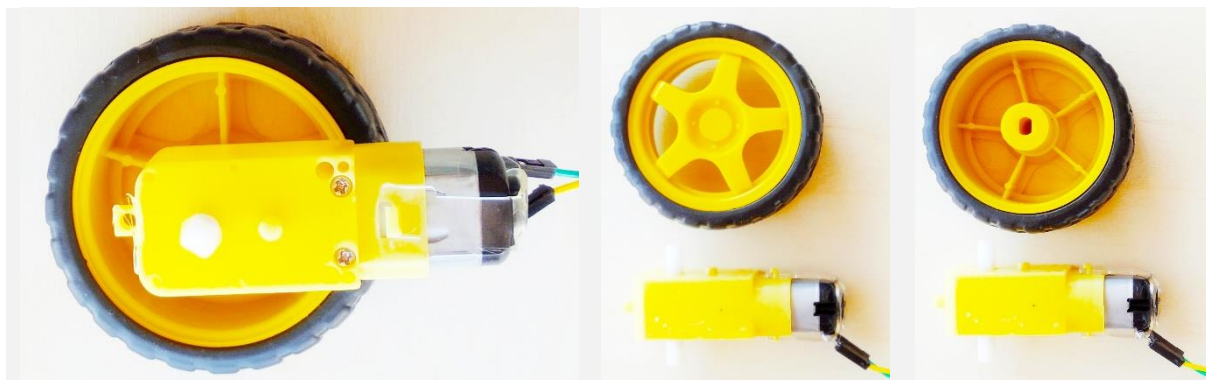
- ο Τάση τροφοδοσίας : έως 12V DC
- ο Ρεύμα λειτουργίας : 80mA
- ο Θερμοκρασία λειτουργίας : 0 - 40 °C
- ο Σχετική υγρασία < 80%
- ο Ρυθμός ροής : 20-60 ml / min
- ο Περιστροφική ταχύτητα αντλίας : 0,1 - 100 rpm
- ο Περιστροφική ταχύτητα κινητήρα : 5000 rpm
- ο Διαστάσεις του οδηγού (Φ x Η) : Διαμ. 27,6 x Ύψος 37,9 (mm)
- ο Διαστάσεις κεφαλιού της αντλίας (Φ x Η) : Διαμ. 31,7 x Ύψος 20,1 (mm)
- ο Διαστάσεις του σωλήνα της αντλίας (ΕΣ. Διαμ x ΕΞ. Διαμ): 2 x 4 χιλιοστά
- ο Βάρος : 200g
- ο Κατεύθυνση ροής : Ελεγχόμενη από τροφοδοσία.

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Περισταλτική αντλία	€ 9,50 με Φ.Π.Α	€ 3,40 με φόρους και μεταφορικά.

Πίνακας 1.9.1

1.10 Εξάρτημα ρόδας - κινητήρα.

Για τη κατασκευή του ρομπότ χρησιμοποιείται ένα εξάρτημα βασικής σημασίας το οποίο αποτελείται από ένα κινητήρα συνεχούς ρεύματος που κινεί μέσω ενός μειωτήρα με γρανάζια έναν άξονα πάνω στον οποίο εφαρμόζει μια πλαστική ρόδα.



Το εξάρτημα ρόδας -κινητήρα.

Σχήμα 1.10.1

Ενδεικτικό Κόστος Υλικού.

Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
Εξάρτημα ρόδας -κινητήρα.	€ 5,00 με Φ.Π.Α	€ 2,10 με φόρους και μεταφορικά.

Πίνακας 1.10.1

Τα τεχνικά χαρακτηριστικά του παρουσιάζονται παρακάτω.

Γενικά Τεχνικά Χαρακτηριστικά:

- ο Τάση τροφοδοσίας : 5 έως 10V DC
- ο Λόγος μετατροπής μειωτήρα : 48/1
- ο Θόρυβος : < 65 dB
- ο Βάρος κινητήρα : 50 gr

**Χαρακτηριστικά του εξαρτήματος ρόδας – κινητήρα
για 3 διαφορετικές τιμές τάσης τροφοδοσίας.**

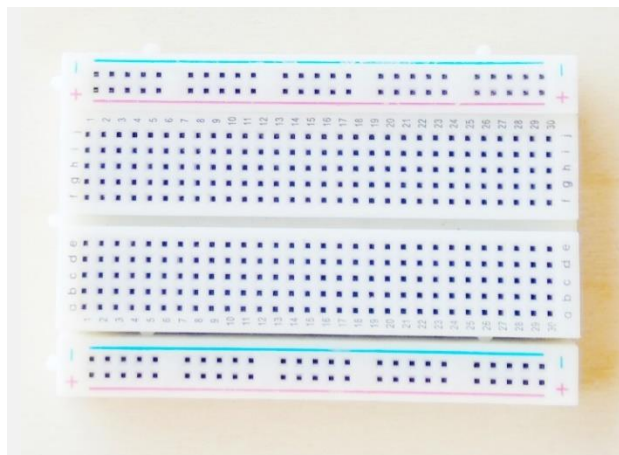
Για τάση τροφοδοσίας (DC V)	3	5	6
Ρεύμα λειτουργίας (mA)	100	100	120
Ταχύτητα περιστροφής με τη ρόδα (RPM)	100	190	240
Ταχύτητα Αυτοκινήτου (m/min)	20	39	48
Πίνακας 1.10.2			

1.11 Άλλα εξαρτήματα

Διάφορα μικρότερα εξαρτήματα που χρησιμοποιήθηκαν για τις ανάγκες των κατασκευών είναι:

1. Πλακέτα Δοκιμών (Breadboard).
2. Συσσωρευτής (Μπαταρία) 9V.
3. Συσσωρευτής (Μπαταρία) 1,5V.
4. Δίοδος Φωτοεκπομπής (Led), 5mm, 1,8V, 15mA.
5. Αντίσταση 220Ω, 0,25W, 5 %
6. Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1
7. Διάφοροι Αγωγοί (jumper wires, male to male, male to female).

8. Ξύλο κόντρα πλακέ πάχους 4mm (περίπου 15x20cm).
9. Ελεύθερα περιστρεφόμενη ρόδα.
10. Πιστόλι θερμόκολλας και Ράβδοι Σιλικόνης.



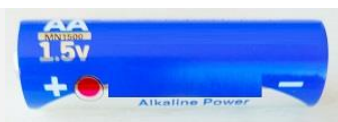
Πλακέτα Δοκιμών (Breadboard).

Σχήμα 1.11.1



Συσσωρευτής 9V

Σχήμα 1.11.2



Συσσωρευτής 1,5V

Σχήμα 1.11.3



Δίοδος Φωτοεκπομπής (Led), 5mm, 1,8V, 15mA

Σχήμα 1.11.4



Αντίσταση 220Ω, 0,25W, 5%

Σχήμα 1.11.5



Προσαρμογέας συσσωρευτή 9V
σε ακροδέκτη 5.5/2.1

Σχήμα 1.11.6



Αγωγοί (jumper wires).

Σχήμα 1.11.7



Ξύλο, κόντρα πλακέ πάχους 4mm
(περίπου 20x30cm).

Σχήμα 1.11.8



Ελεύθερα περιστρεφόμενη ρόδα.

Σχήμα 1.11.9



Πιστόλι θερμόκολλας και
Ράβδοι Σιλικόνης.

Σχήμα 1.11.10

Ενδεικτικό Κόστος Υλικού.		
Υλικό	Τιμή Ελληνικού καταστήματος.	Τιμή Διαδικτυακού καταστήματος εξωτερικού.
1. Πλακέτα Δοκιμών (Breadboard).	€ 2,50 με Φ.Π.Α	€ 0,90 με φόρους και μεταφορικά.
2. Συσσωρευτής (Μπαταρία) 9V.	€ 1,40 με Φ.Π.Α	€ 2,00 με φόρους και μεταφορικά.
3. Συσσωρευτής (Μπαταρία) 1,5V	€ 0,50 με Φ.Π.Α	€ 0,60 με φόρους και μεταφορικά.
4. Δίοδος Φωτοεκπομπής (Led), 5mm, 1,8V, 15mA	€ 0,10 με Φ.Π.Α	€ 0,015 με φόρους και μεταφορικά.
5. Αντίσταση 220Ω, 0,25W, 5%	€ 0,01 με Φ.Π.Α	€ 0,0073 με φόρους και μεταφορικά.
6. Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1	€ 0,50 με Φ.Π.Α	€ 0.35 με φόρους και μεταφορικά.
7. Διάφοροι αγωγοί (jumper wires, male to male, male to female).	€ 0,06 με Φ.Π.Α	€ 0,015 με φόρους και μεταφορικά.
8. Ξύλο κόντρα πλακέ πάχους 4mm (περίπου 15x20cm).	€ 2,00 με Φ.Π.Α	€ 4,00 με φόρους και μεταφορικά.
9. Ελεύθερα περιστρεφόμενη ρόδα.	€ 1,80 με Φ.Π.Α	€ 1,40 με φόρους και μεταφορικά.
10. Πιστόλι θερμόκολλας και ράβδοι σιλικόνης.	€ 7,00 με Φ.Π.Α	€ 10,00 με φόρους και μεταφορικά.
Πίνακας 1.11.1		

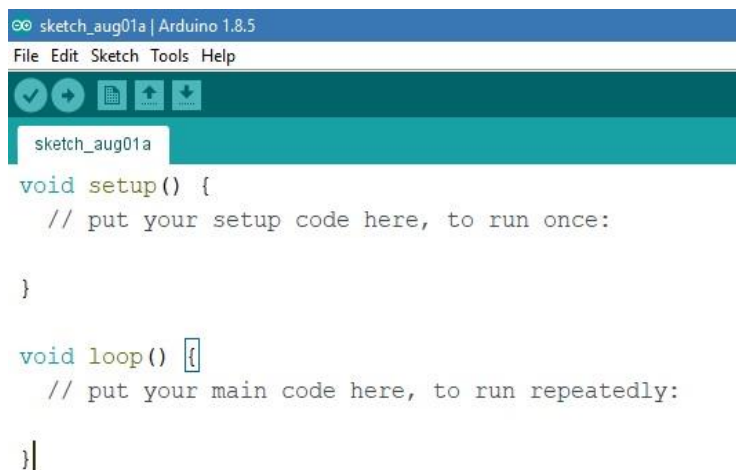
1.12 Λογισμικό.

Ο προγραμματισμός του μικροελεγκτή έγινε μέσω της γλώσσας *Arduino* (με βάση τη Wiring και τη C++) και του λογισμικού Arduino IDE, το οποίο παρέχεται από την Arduino.cc για τον σκοπό αυτό. Ο κώδικας αναπτύχθηκε με τέτοιο τρόπο ώστε να είναι όσο το δυνατόν πιο κατανοητός από αρχάριους (μαθητές, χομπίστες κτλ.) και να έχει εκπαιδευτικό χαρακτήρα. Δεν είναι ο πιο συμπαγής ούτε ο πιο αποδοτικός αλλά έχει τη λογική του «μαθαίνω να χτίζω πάνω στα απλά παραδείγματα περισσότερο σύνθετα πράγματα».

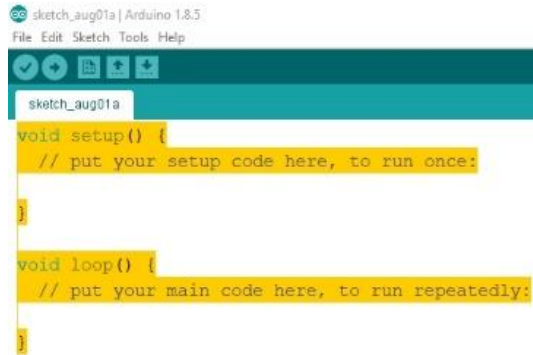
1.12.1 Μεταφόρτωση προγράμματος στο Arduino.

Προκειμένου να μεταφορτωθεί ένα πρόγραμμα (σκίτσο ή sketch, όπως το ονομάζει το Arduino IDE) σε μια πλακέτα Arduino UNO (όπως αυτή που χρησιμοποιείται στις κατασκευές) μέσω λειτουργικού συστήματος windows, πρέπει να ακολουθηθεί μια απλή διαδικασία βήμα προς βήμα:

1. Συνδέστε το Arduino με το καλώδιο USB.
2. Το τετράγωνο άκρο του καλωδίου USB συνδέστε το με το Arduino και την επίπεδη άκρη συνδέστε τη με μια θύρα USB στον υπολογιστή σας.
3. Εκκινήστε την εφαρμογή Arduino IDE από τον Η/Υ σας.
4. Δημιουργήστε ένα νέο σκίτσο.



5. Επιλέξτε ο,τιδήποτε περιέχει αυτόματα το νέο σκίτσο (δημιουργούνται οι δυο βασικές ρουτίνες **setup** και **loop**).



```

sketch_aug01a | Arduino 1.8.5
File Edit Sketch Tools Help

sketch_aug01a

void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}

```

6. Διαγράψτε ο,τιδήποτε περιέχει.



```

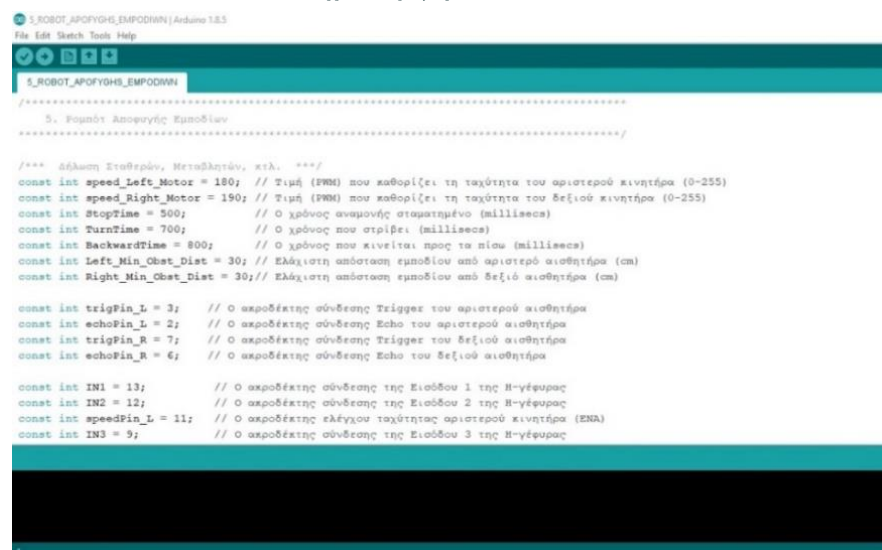
sketch_aug01a | Arduino 1.8.5
File Edit Sketch Tools Help

sketch_aug01a$

```



7. Αντιγράψτε από το Παράτημα τον κώδικα της κατασκευής που υλοποιήσατε και επικολλήστε στο νέο σκίτσο που δημιουργήσατε.



```

5_ROBOT_APOFYGH6_EMPODWN | Arduino 1.8.5
File Edit Sketch Tools Help

5_ROBOT_APOFYGH6_EMPODWN

5. Ρομπότ Αποφυγής Εμποδίων
*****

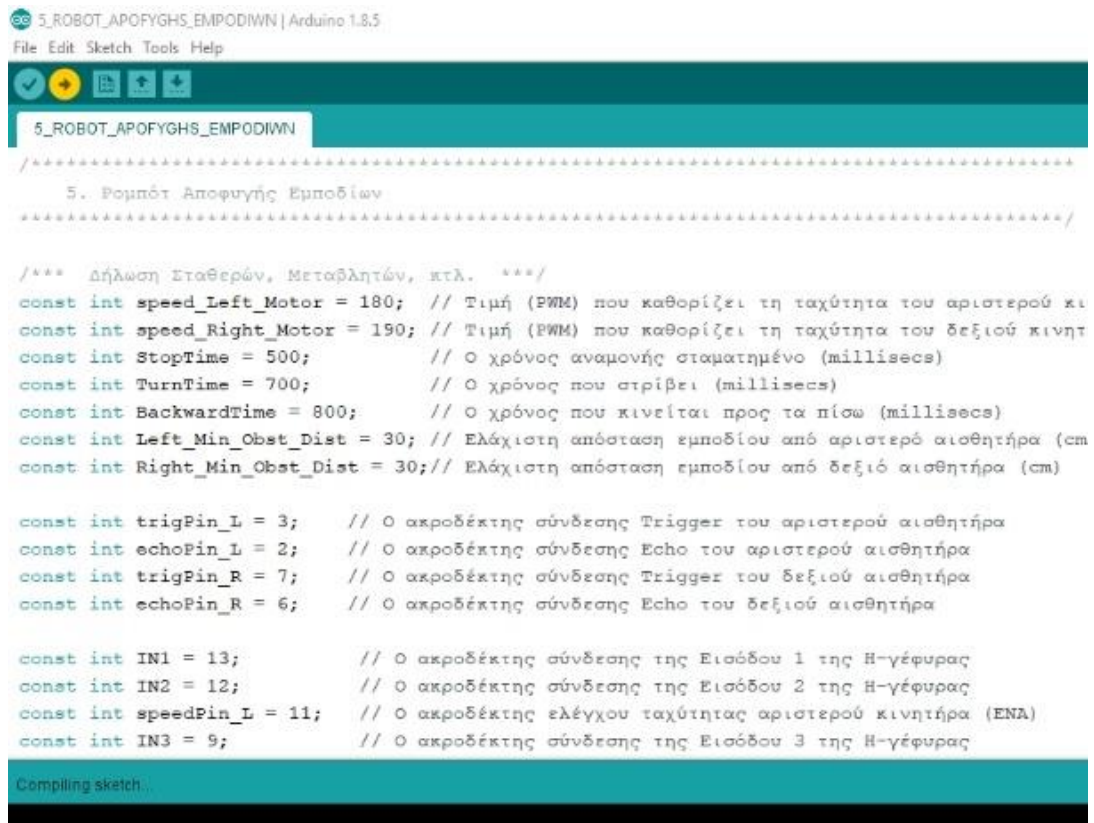
/**
 * Δόκιμο Σταθιρόν, Νεταδελιόν, κτλ.
 */
const int speed_Left_Motor = 180; // Τιμή (PWM) που καθορίζει τη ταχύτητα του αριστερού κινητήρα (0-255)
const int speed_Right_Motor = 190; // Τιμή (PWM) που καθορίζει τη ταχύτητα του δεξιού κινητήρα (0-255)
const int StopTime = 500; // Ο χρόνος αναμονής σταματημένο (milliseconds)
const int TurnTime = 700; // Ο χρόνος που στρίβει (milliseconds)
const int BackwardTime = 800; // Ο χρόνος που κινείται προς τα πίσω (milliseconds)
const int Left_Min_Obst_Dist = 30; // Ελάχιστη απόσταση εμποδίου από αριστερό αισθητήρα (cm)
const int Right_Min_Obst_Dist = 30; // Ελάχιστη απόσταση εμποδίου από δεξιό αισθητήρα (cm)

const int trigPin_L = 3; // Ο ακροδέκτης σύνδεσης Trig του αριστερού αισθητήρα
const int echoPin_L = 2; // Ο ακροδέκτης σύνδεσης Echo του αριστερού αισθητήρα
const int trigPin_R = 7; // Ο ακροδέκτης σύνδεσης Trig του δεξιού αισθητήρα
const int echoPin_R = 6; // Ο ακροδέκτης σύνδεσης Echo του δεξιού αισθητήρα

const int IN1 = 13; // Ο ακροδέκτης σύνδεσης της Εισόδου 1 της Η-γέφυρας
const int IN2 = 12; // Ο ακροδέκτης σύνδεσης της Εισόδου 2 της Η-γέφυρας
const int speedPin_L = 11; // Ο ακροδέκτης ελέγχου ταχύτητας αριστερού κινητήρα (ENA)
const int IN3 = 9; // Ο ακροδέκτης σύνδεσης της Εισόδου 3 της Η-γέφυρας

```


8. Επιλέξτε Εργαλεία → Πίνακας → Arduino Uno για να βρείτε την πλακέτα σας στο μενού Arduino.
9. Βρείτε μια λίστα με όλες τις διαθέσιμες σειριακές θύρες επιλέγοντας Εργαλεία → Σειριακή θύρα → comX. Το X σηματοδοτεί έναν διαδοχικό ή τυχαίο αριθμό.
10. Πολλές συσκευές μπορούν να καταχωρηθούν στη λίστα θύρας COM και αν συνδέσετε πολλές πλακέτες, θα τους δοθεί ένας νέος αριθμός. Αν δεν έχετε άλλο Arduino συνδεδεμένο, θα πρέπει να είναι ο μόνος ορατός.
11. Επιλέξτε τη σωστή σειριακή θύρα για την πλακέτα σας.
12. Πατήστε Μεταφόρτωση - UPLOAD (το οριζόντιο βελάκι πάνω αριστερά) και περιμένετε μερικά δευτερόλεπτα μέχρι να ολοκληρωθεί η διαδικασία.



```

5_ROBOT_APOFYGHIS_EMPODIWN | Arduino 1.8.5
File Edit Sketch Tools Help

5_ROBOT_APOFYGHIS_EMPODIWN

/*****
5. Ρομπότι Αποφυγής Εμποδίων
*****/

/**** Δήλωση Σταθερών, Μεταβλητών, κτλ. ****/
const int speed_Left_Motor = 180; // Τιμή (PWM) που καθορίζει τη ταχύτητα του αριστερού κι
const int speed_Right_Motor = 190; // Τιμή (PWM) που καθορίζει τη ταχύτητα του δεξιού κινητή
const int StopTime = 500; // Ο χρόνος αναμονής σταματημένο (milliseconds)
const int TurnTime = 700; // Ο χρόνος που στρίβει (milliseconds)
const int BackwardTime = 800; // Ο χρόνος που κινείται προς τα πίσω (milliseconds)
const int Left_Min_Obst_Dist = 30; // Ελάχιστη απόσταση εμποδίου από αριστερό αισθητήρα (cm)
const int Right_Min_Obst_Dist = 30; // Ελάχιστη απόσταση εμποδίου από δεξιό αισθητήρα (cm)

const int trigPin_L = 3; // Ο ακροδέκτης σύνδεσης Trigger του αριστερού αισθητήρα
const int echoPin_L = 2; // Ο ακροδέκτης σύνδεσης Echo του αριστερού αισθητήρα
const int trigPin_R = 7; // Ο ακροδέκτης σύνδεσης Trigger του δεξιού αισθητήρα
const int echoPin_R = 6; // Ο ακροδέκτης σύνδεσης Echo του δεξιού αισθητήρα

const int IN1 = 13; // Ο ακροδέκτης σύνδεσης της Εισόδου 1 της H-γέφυρας
const int IN2 = 12; // Ο ακροδέκτης σύνδεσης της Εισόδου 2 της H-γέφυρας
const int speedPin_L = 11; // Ο ακροδέκτης ελέγχου ταχύτητας αριστερού κινητήρα (ENA)
const int IN3 = 9; // Ο ακροδέκτης σύνδεσης της Εισόδου 3 της H-γέφυρας

Compiling sketch...

```


Εναλλακτικά:

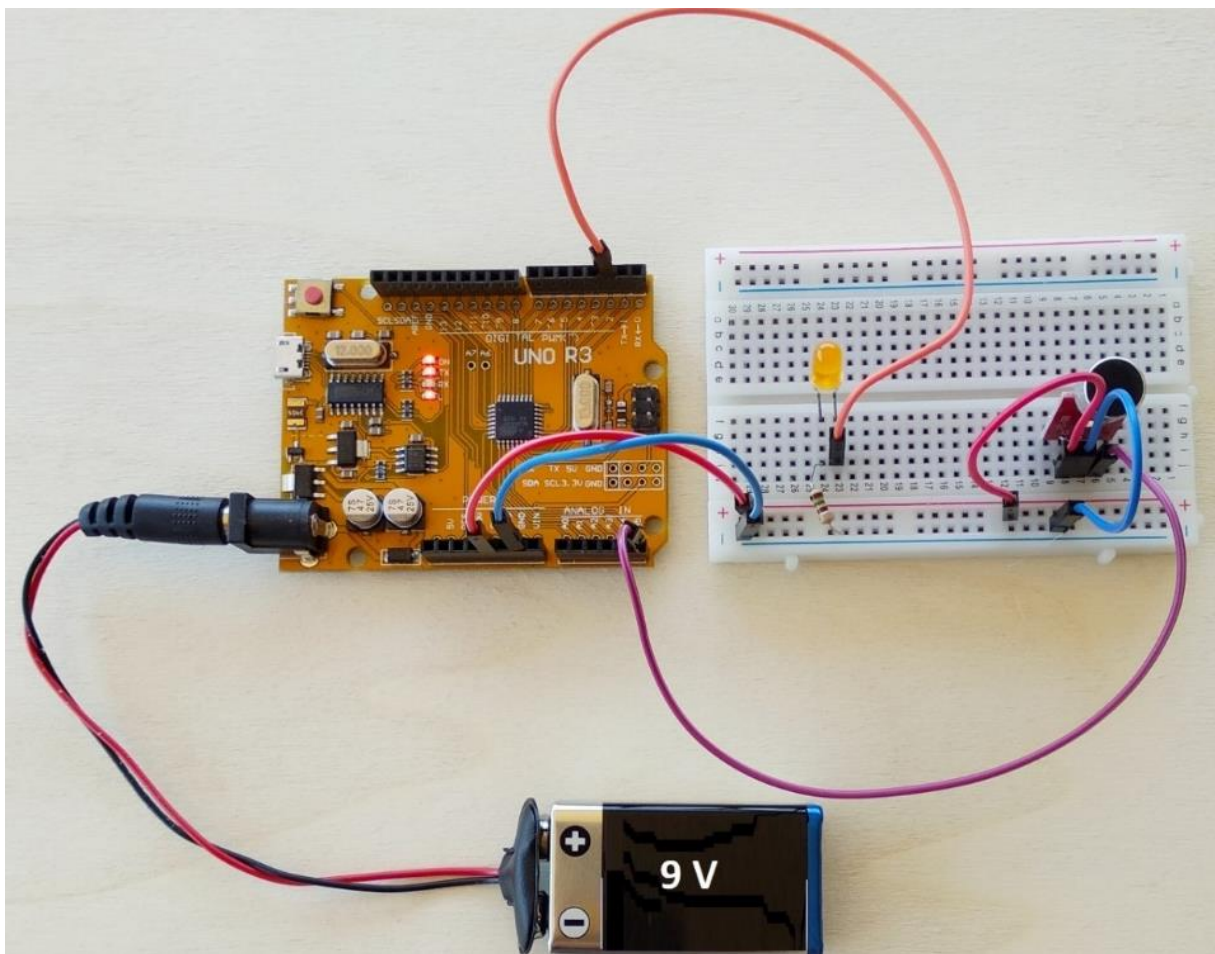
1. Πατήστε την υπερσύνδεση που υπάρχει στο τέλος κάθε κατασκευής στο κεφάλαιο 2.
2. Αποθηκεύστε το κατάλληλο αρχείο (τύπου **.ino**) που περιέχει τον κώδικα.
3. Πατήστε μεταφόρτωση (**UPLOAD**).

Κεφάλαιο 2 - Κατασκευές

2.1 Ηχοδιακόπτης

2.1.1 Εισαγωγή

Η κατασκευή του ηχοδιακόπτη λειτουργεί ανιχνεύοντας το επίπεδο του θορύβου στον περιβάλλοντα χώρο και αλλάζει την κατάσταση μιας λυχνίας (Led). Αν είναι σβηστή και ο θόρυβος ξεπεράσει ένα προκαθορισμένο όριο κατωφλίου που έχει δηλωθεί στον κώδικα προγραμματισμού με ένα παλαμάκι, μια φωνή ή κάποιον άλλον ήχο, τότε η λυχνία θα ανάψει.

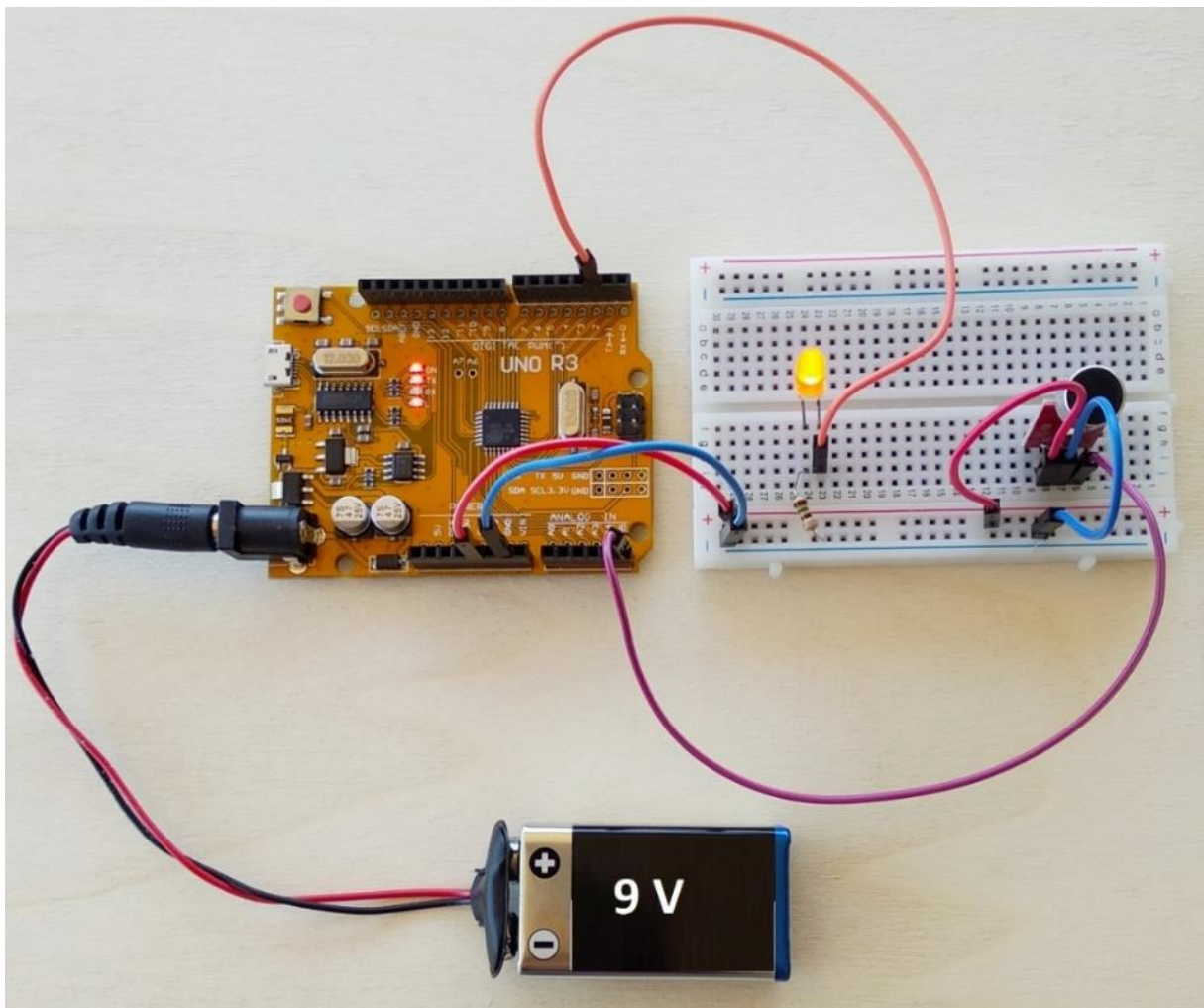


Η κατασκευή του Ηχοδιακόπτη.
Σε κατάσταση ηρεμίας, η λυχνία σβηστή.

Σχήμα 2.1.1.α

Αν είναι αναμμένη με ένα παλαμάκι θα σβήσει και ούτω καθεξής. Αυτή η κατασκευή είναι αρκετά απλή και μπορεί να έχει χρηστική αξία αν στη θέση της λυχνίας τοποθετηθεί ένας ηλεκτρονόμος (ρελέ). Αυτός μπορεί να ελέγξει μεγαλύτερα φορτία και φορτία διαφορετικής ονομαστικής τάσης από την τροφοδοσία του μικροελεγκτή, όπως ένα φωτιστικό οικίας, έναν ανεμιστήρα κ.α. Ένα τέτοιο ηχοδιακόπτη μπορούμε να τοποθετήσουμε π.χ. στα φώτα ενός σαλονιού και με ένα παλαμάκι να τα ελέγξουμε.

Η κατασκευή μπορεί να δουλέψει με μια απλή μπαταρία 9V ή 12V ή με ένα τροφοδοτικό DCV ενώ η ευαισθησία του ηχοδιακόπτη ρυθμίζεται εύκολα από το πρόγραμμα που φορτώνεται στο μικροελεγκτή.











Η κατασκευή του Ηχοδιακόπτη.
Η λυχνία αναμμένη.

Σχήμα 2.1.1.β

2.1.2 Υλικά Κατασκευής

Τα απαραίτητα υλικά για την κατασκευή καταγράφονται στον παρακάτω πίνακα:

Μικροελεγκτής Arduino Uno		1 τεμάχιο
Μικρόφωνο Max9812		1 τεμάχιο
Πλακέτα Δοκιμών (Breadboard).		1 τεμάχιο
Λυχνία (Led).		1 τεμάχιο
Συσσωρευτής (Μπαταρία) 9V		1 τεμάχιο
Αντίσταση 220Ω.		1 τεμάχιο
Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1		1 τεμάχιο
Διάφοροι Αγωγοί (jumper wires- male to male).		6 τεμάχια

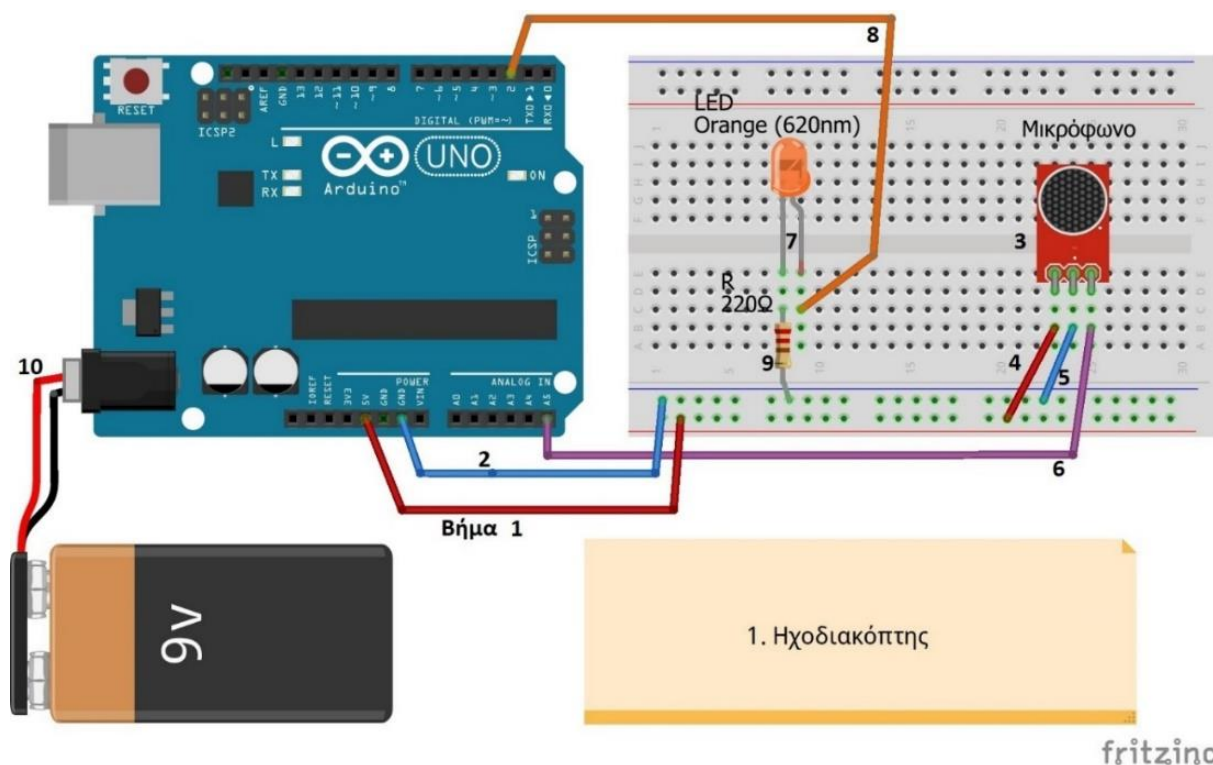
Πίνακας 2.1.2

2.1.3 Περιγραφή Κατασκευής

Αρχικά, μεταφορτώστε το πρόγραμμα στον μικροελεγκτή από το περιβάλλον προγραμματισμού Arduino IDE.

Στη συνέχεια για την κατασκευή του κυκλώματος, εφαρμόστε τα παρακάτω βήματα όπως καταγράφονται και πάνω στο σχήμα 2.1.3:

1. Συνδέστε το ένα άκρο ενός αγωγού (jumper wire – male to male) κόκκινου χρώματος με τον ακροδέκτη **5V** του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** (οριζόντια κόκκινη γραμμή) της πλακέτας δοκιμών (Breadboard).
2. Συνδέστε το ένα άκρο ενός αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** (γείωση) του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** (γαλάζια γραμμή) της πλακέτας δοκιμών (Breadboard).



Το κύκλωμα του Ηχοδιακόπτη.
Οι αριθμοί είναι τα βήματα για τη κατασκευή του.

Σχήμα 2.1.3

3. Καρφώστε το μικρόφωνο στην πλακέτα δοκιμών (Breadboard), οι τρεις ακροδέκτες να είναι σε τρεις διαδοχικές **πεντάδες οπών** (κάθετης γραμμής), όπως φαίνεται στο σχήμα 2.1.3.
4. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με μια οπή από την **πεντάδα οπών** (κάθετης γραμμής) με την οποία συνδέθηκε ο ακροδέκτης **Vcc** του μικροφώνου και το άλλο άκρο με μια από τις οπές της κάτω **θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
5. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε ο ακροδέκτης **GND** του μικροφώνου και το άλλο άκρο με μια από τις οπές της κάτω **αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
6. Συνδέστε το ένα άκρο ενός αγωγού π.χ. ιώδους (μωβ) χρώματος με τον ακροδέκτη **A5** του μικροελεγκτή και το άλλο άκρο με την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε ο ακροδέκτης **OUT** του μικροφώνου.

7. Καρφώστε τη λυχνία (Led) στη πλακέτα δοκιμών, με τους δυο ακροδέκτες σε δύο διαδοχικές **πεντάδες οπών** κάθετης γραμμής.
8. Συνδέστε το ένα άκρο ενός αγωγού π.χ. πορτοκαλί χρώματος με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε το μακρύτερο άκρο της λυχνίας (**Άνοδος +**) και το άλλο άκρο με τον ακροδέκτη **2** του μικροελεγκτή.
9. Συνδέστε το ένα άκρο μιας αντίστασης 220Ω με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε το πιο κοντό άκρο της λυχνίας (**Κάθοδος -**) και το άλλο άκρο της αντίστασης με μια από τις οπές της κάτω **αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
10. Τέλος, συνδέστε τον προσαρμογέα (συσσωρευτή 9V με ακροδέκτη 5.5/2.1) με το συσσωρευτή και με την κατάλληλη υποδοχή του μικροελεγκτή για να τον τροφοδοτήσετε.

Ακολουθώντας τις παραπάνω οδηγίες ο ηχοδιακόπτης θα πρέπει να λειτουργεί.

2.1.4 Περιγραφή Λειτουργίας

Ένας συνήθης κι απλός κώδικας, όπως ο παρακάτω, αποτελείται από τρία μέρη: στο πρώτο ορίζονται οι σταθερές, έπειτα οι μεταβλητές που θα χρησιμοποιηθούν και τέλος οι αρχικές τιμές τους.

Εδώ έχουμε τρεις σταθερές:

- **PinLed**
Ο ακροδέκτης που συνδέθηκε η λυχνία (2).
- **PinMic**
Ο ακροδέκτης που συνδέθηκε το μικρόφωνο (A5).
- **ThresholdValue**
Το όριο (κατώφλι) του σήματος πάνω από το οποίο θα μεταβάλλεται η κατάσταση της λυχνίας, με αρχική τιμή 470 βάσει των δοκιμών που έγιναν (ThresholdValue = 470).

και δυο μεταβλητές:

- **SignalMic**
Η τιμή σήματος του μικροφώνου.
- **LedState**
Η κατάσταση της λυχνίας η οποία καθορίζεται αρχικά ως σβηστή (LedState = LOW).

Στη συνέχεια, στη ρουτίνα Setup που «τρέχει» μόνο μια φορά, γίνεται η απαραίτητη αρχικοποίηση, δηλαδή δηλώνεται ποιοι από τους ακροδέκτες θα λειτουργούν ως εισόδοι και ποιοι ως εξόδοι. Εδώ ο ακροδέκτης 2 της λυχνίας θα δηλωθεί ως εξόδος ενώ ο A5 θα δηλωθεί ως είσοδος. Επίσης, ξεκίνα η σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.

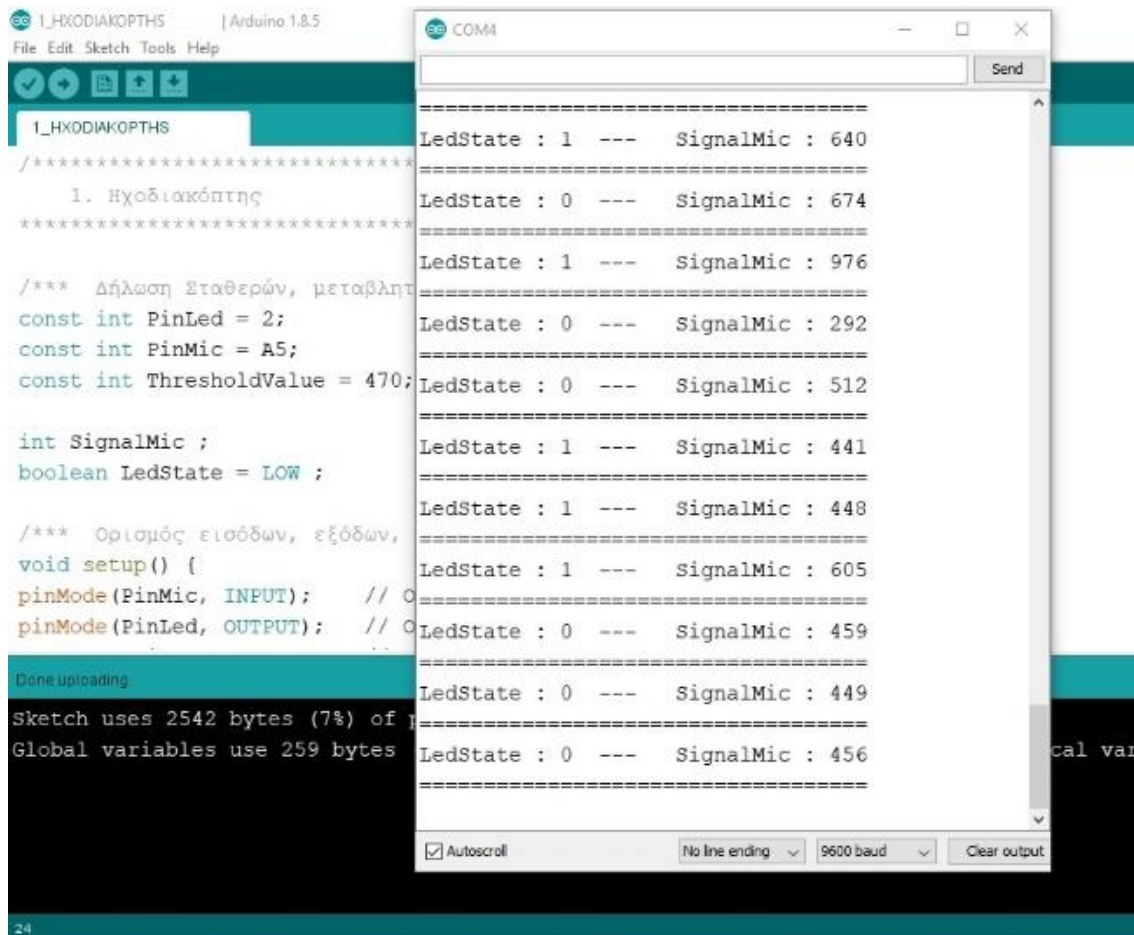
Τέλος, έχουμε τη ρουτίνα Loop σαν ένα ατέρμονα βρόχο, η οποία εκτελείται ξανά και ξανά για όσο λειτουργεί το πρόγραμμα. Εδώ κάθε φορά «διαβάζεται» η τρέχουσα τιμή της εισόδου και αποδίδεται στη μεταβλητή **SignalMic**. Έπειτα, η τιμή (LOW ή HIGH) της μεταβλητής **LedState** (κατάσταση της λυχνίας) δίνεται στη σταθερά **PinLed** (ακροδέκτης της λυχνίας, δηλαδή ο 2), αυτό σημαίνει ότι αν η μεταβλητή **LedState** είναι LOW τότε ο ακροδέκτης 2, με τον οποίο είναι συνδεδεμένη η λυχνία, θα πάρει 0V και η λυχνία θα είναι σβηστή. Αν όμως η μεταβλητή **LedState** είναι HIGH τότε ο ακροδέκτης 2, με τον οποίο είναι συνδεδεμένη η λυχνία, θα πάρει 5V και η λυχνία θα ανάβει. Ακολούθως, εκτυπώνεται στην οθόνη του Η/Υ η τρέχουσα κατάσταση της λυχνίας και η τρέχουσα τιμή του σήματος μικροφώνου.

Ύστερα από αυτά, ο κώδικας έχει μια IF η οποία ελέγχει την τρέχουσα τιμή του σήματος **SignalMic** με το όριο κατωφλίου **ThresholdValue** του σήματος πάνω από το οποίο θα αλλάζει η κατάσταση της λυχνίας. Αν η τιμή είναι μεγαλύτερη, τότε αλλάζει την τιμή της μεταβλητής “κατάσταση” της λυχνίας **LedState** και περιμένει 200 millisecs για να αποφύγει ψευδή ανάμματα και σβησίματα. Τέλος, περιμένει 1 millisec βελτιώνοντας την ευστάθεια του βρόχου πριν αρχίσει ξανά από την αρχή να εκτελεί τις εντολές της Loop.

Παρατηρήσεις :

- Προτείνεται να δοκιμάσετε το όριο κατωφλίου **ThresholdValue** στο δικό σας χώρο με τις ιδιαίτερες συνθήκες που επικρατούν και να δώσετε τη δική σας τιμή στη μεταβλητή ώστε να έχετε την επιθυμητή ευαισθησία του ηχοδιακόπτη σας, π.χ. αν η λυχνία αναβοσβήνει συνέχεια, ανεβάστε λίγο το όριο από το 470 που έχει προεπιλεγεί δοκιμάζοντας ανά 10.
- Οι εντολές που αφορούν τη σειριακή επικοινωνία και τα σχόλια μπορούν να παραληφθούν χωρίς να επηρεάζεται η λειτουργία της κατασκευής.
- Τα χρώματα των αγωγών δεν είναι προφανώς δεσμευτικά, είθισται πάντως σε κυκλώματα συνεχούς ρεύματος η συνδεσμολογία της τροφοδοσίας να γίνεται για τον θετικό πόλο με κόκκινο χρώμα και για τον αρνητικό (γείωση) με μαύρο

ή γαλάζιο. Οι αγωγοί που συνδέονται σε λυχνίες έχουν το ίδιο χρώμα με αυτές. Για τους υπόλοιπους αγωγούς δεν έχει ακολουθηθεί κάποια τυποποιημένη διαδικασία αλλά χρησιμοποιήθηκαν επίτηδες αρκετά διαφορετικά χρώματα προκειμένου να είναι ευδιάκριτο για τον κάθε αναγνώστη από πού ξεκινά και πού καταλήγει κάθε αγωγός.



Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.

Σχήμα 2.1.4

2.1.5 Κώδικας Προγραμματισμού

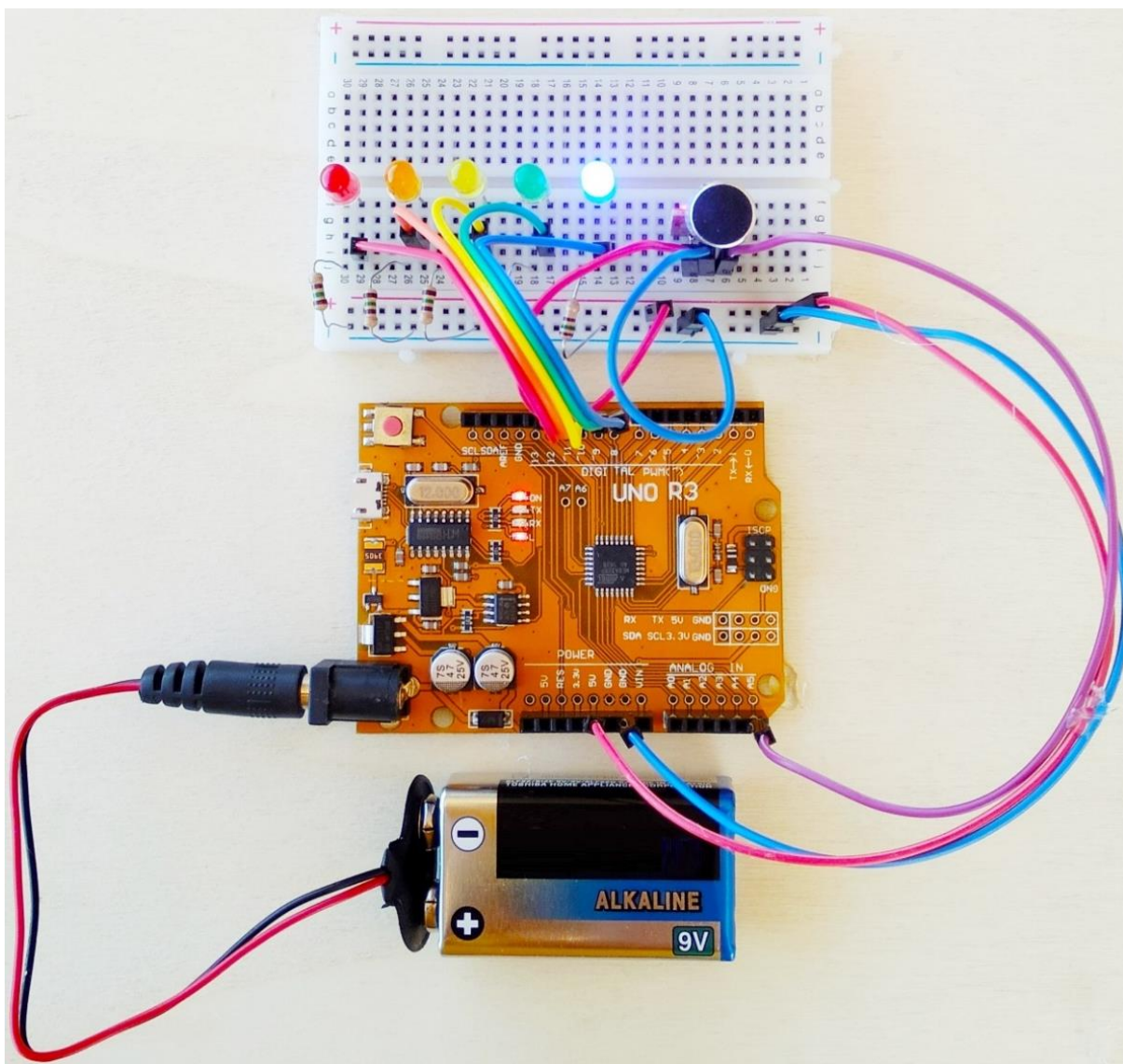
Ο κώδικας παρατίθεται στο Παράρτημα.

Επιπλέον, μπορείτε να τον κατεβάσετε σε αρχείο πατώντας [εδώ](#).

2.2 Φασαριόμετρο.

2.2.1 Εισαγωγή

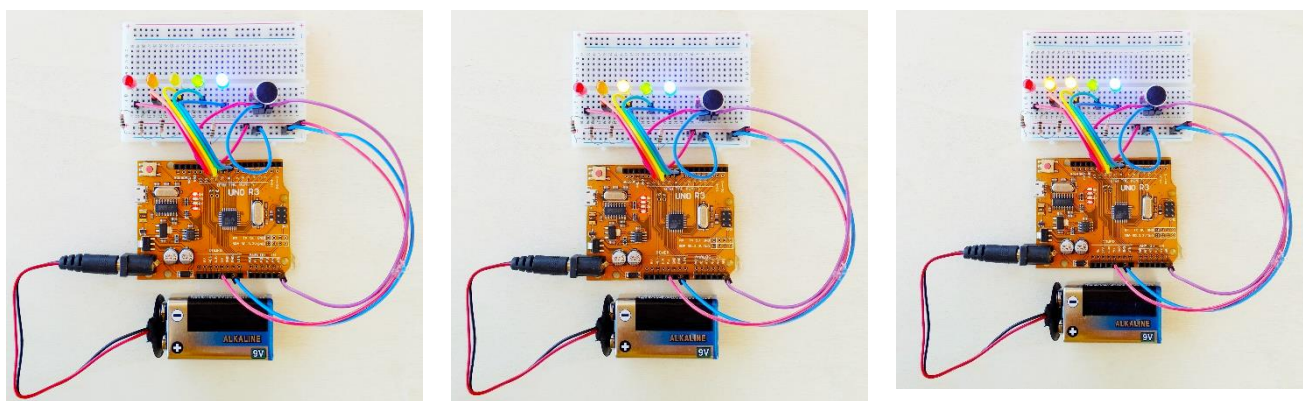
Η κατασκευή του Φασαριόμετρου είναι αρκετά διασκεδαστική και έχει χρηστική αξία μέσα στη σχολική τάξη όταν ολοκληρωθεί.



Η κατασκευή του Φασαριόμετρου.
Το επίπεδο του θορύβου (0) είναι πολύ χαμηλό, ανάβει μόνο η γαλάζια λυχνία..

Σχήμα 2.2.1.α

Όμοια με τον ηχοδιακόπτη, το φασαριόμετρο λειτουργεί ανιχνεύοντας το επίπεδο του θορύβου στον περιβάλλοντα χώρο και οπτικοποιεί το επίπεδο αυτό (ταξινόμηση σε πέντε επίπεδα από 0 ως 4) σε μια πεντάδα λυχνιών. Αν το επίπεδο του θορύβου είναι πολύ χαμηλό, ανάβει μόνο η γαλάζια λυχνία. Αν το επίπεδο ανεβεί λίγο, ανάβουν δυο λυχνίες, η γαλάζια και η πράσινη. Η διαδικασία συνεχίζεται ως το μέγιστο που μπορεί να μετρήσει το μικρόφωνο στο οποίο ανάβουν και οι πέντε λυχνίες, δηλαδή ως την κόκκινη.



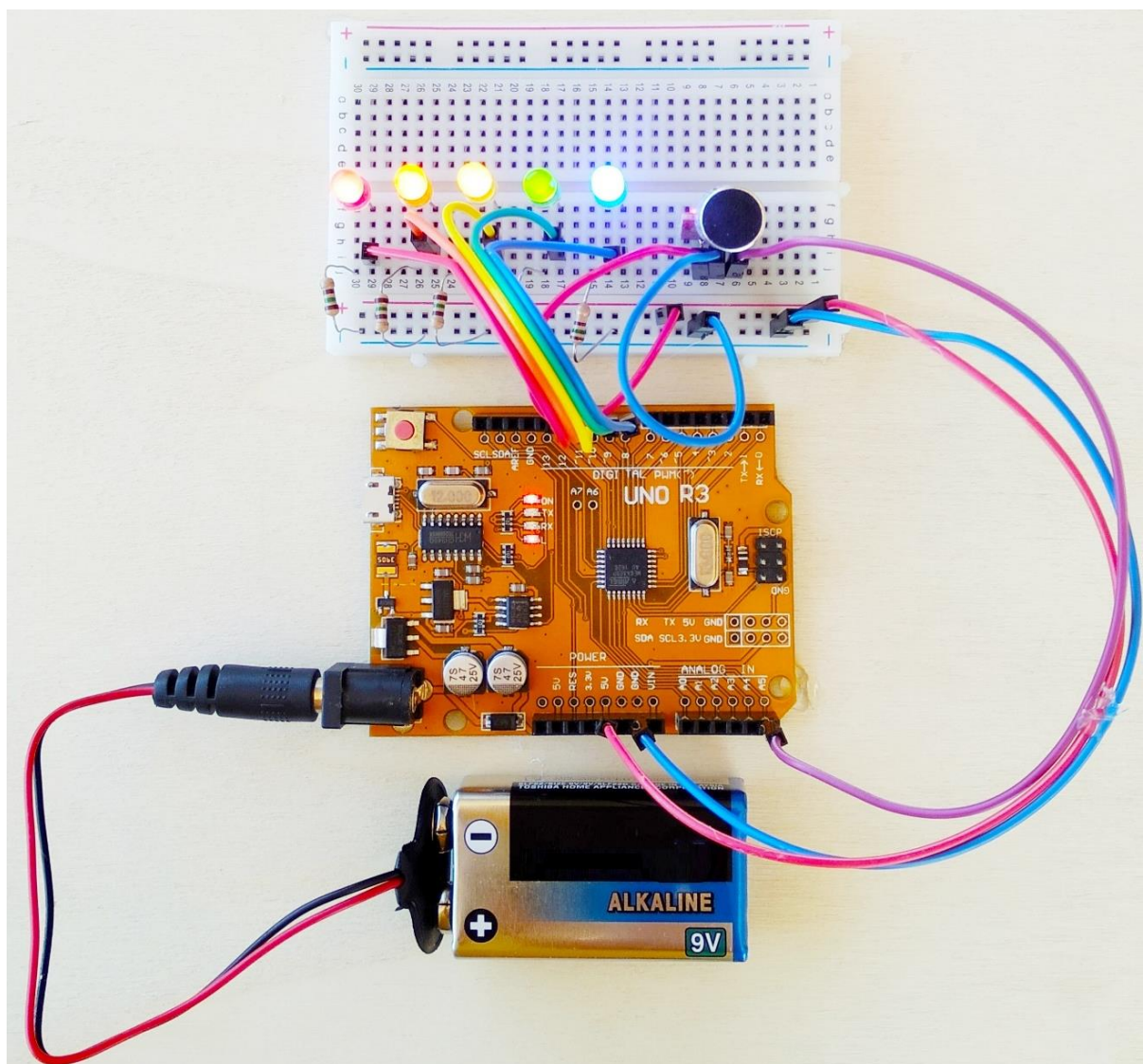
Τα μεσαία επίπεδα θορύβου που μπορεί να μετρήσει η κατασκευή, από χαμηλό (1) ως υψηλό (3).

Σχήμα 2.2.1.β

Με την κατασκευή και τη χρήση του «φασαριόμετρου» στην εκπαιδευτική πράξη επιτυγχάνεται η σύνδεση της τεχνολογίας και της επιστήμης με την καθημερινότητα των μαθητών, προάγεται η συνεργασία και δημιουργείται ένα κλίμα συλλογικότητας. Οι μαθητές διασκεδάζουν και η διαδικασία αυτή φαίνεται να λειτουργεί ως όργανο διαχείρισης της τάξης ακόμα και σε δύσκολες περιπτώσεις. Επίσης, όλα τα μέλη της ομάδας συμφωνούν για την ένταση του ήχου που είναι ανεκτός, ανάλογα με τη δραστηριότητα που καλούνται να περατώσουν οι μαθητές (ατομική, ομαδική ή συλλογική εργασία), δημιουργώντας έτσι ευνοϊκές συνθήκες εργασίας στην τάξη (βλ. β.11).

Ένα πρόσθετο πλεονέκτημα είναι ότι επειδή η κατασκευή μπορεί να δουλέψει με μια απλή μπαταρία 9V, 12V ή με ένα τροφοδοτικό DCV, μπορεί να μεταφερθεί και να δανειστεί από τάξη σε τάξη και να διασκεδάσουν περισσότεροι μαθητές. Από διαπιστώσεις εκπαιδευτικών που έχουν χρησιμοποιήσει τη συγκεκριμένη εφαρμογή

και άλλες παρόμοιες, σε διεθνές επίπεδο και σε διαφορετικές ηλικίες, φαίνεται να επιβεβαιώνουν τα ικανοποιητικά αυτά αποτελέσματα.











Πολύ υψηλό επίπεδο (4) θορύβου. Όλες οι λυχνίες ανάβουν.

Σχήμα 2.2.1.γ

2.2.2 Υλικά Κατασκευής

Τα απαραίτητα υλικά για την κατασκευή καταγράφονται στον παρακάτω πίνακα:

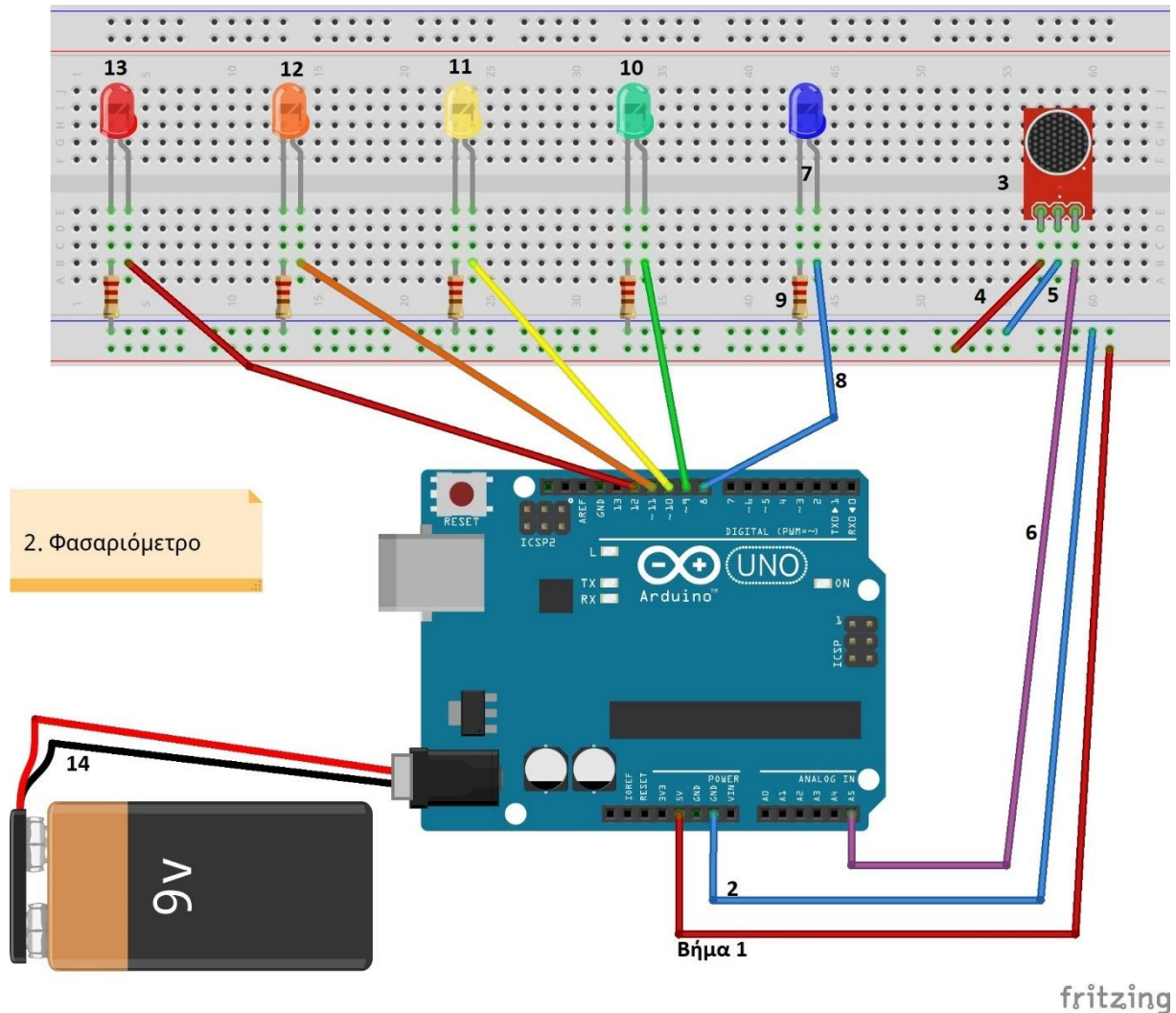
Μικροελεγκτής Arduino Uno.		1 τεμάχιο
Μικρόφωνο Max9812.		1 τεμάχιο
Πλακέτα Δοκιμών (Breadboard).		1 τεμάχιο
Λυχνία (Led) , διάφορα χρώματα.		5 τεμάχια
Συσσωρευτής (Μπαταρία) 9V.		1 τεμάχιο
Αντίσταση 220Ω.		5 τεμάχια
Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1		1 τεμάχιο
Διάφοροι Αγωγοί (jumper wires- male to male).		10 τεμάχια
Πίνακας 2.2.2		

2.2.3 Περιγραφή Κατασκευής

Αρχικά, μεταφορτώστε το πρόγραμμα στον μικροελεγκτή από το περιβάλλον προγραμματισμού Arduino IDE.

Στη συνέχεια για την κατασκευή του κυκλώματος, όπως απεικονίζεται στο Σχήμα 2.2.3, υλοποιήστε τα παρακάτω βήματα:

1. Συνδέστε το ένα άκρο ενός αγωγού (jumper wire – male to male) κόκκινου χρώματος με τον ακροδέκτη **5V** του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** (οριζόντια κόκκινη γραμμή) της πλακέτας δοκιμών (Breadboard).
2. Συνδέστε το ένα άκρο ενός αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** (γείωση) του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** (γαλάζια γραμμή) της πλακέτας δοκιμών (Breadboard).



Το κύκλωμα του Φασαριόμετρου.
Οι αριθμοί είναι τα βήματα για τη κατασκευή του.

Σχήμα 2.2.3

3. Καρφώστε το μικρόφωνο στην πλακέτα δοκιμών (Breadboard), οι τρεις ακροδέκτες σε τρεις διαδοχικές **πεντάδες οπών** (κάθετης γραμμής), όπως φαίνεται στο σχήμα 2.2.3.
4. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με μια οπή από την **πεντάδα οπών** (κάθετης γραμμής) με την οποία συνδέθηκε ο ακροδέκτης **Vcc** του μικροφώνου και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.

5. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε ο ακροδέκτης **GND** του μικροφώνου και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
6. Συνδέστε το ένα άκρο ενός αγωγού π.χ. ιώδους (μωβ) χρώματος με τον ακροδέκτη **A5** του μικροελεγκτή και το άλλο άκρο με την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε ο ακροδέκτης **OUT** του μικροφώνου.
7. Καρφώστε την πρώτη λυχνία (Led) γαλάζιου χρώματος στην πλακέτα δοκιμών με τους δυο ακροδέκτες σε δύο διαδοχικές **πεντάδες οπών** κάθετης γραμμής.
8. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε το μακρύτερο άκρο της λυχνίας **Άνοδος (+)** και το άλλο άκρο με τον ακροδέκτη **8** του μικροελεγκτή.
9. Συνδέστε το ένα άκρο μιας αντίστασης 220Ω με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε το πιο κοντό άκρο της λυχνίας **Κάθοδος (-)** και το άλλο άκρο της αντίστασης με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
10. Επαναλάβετε τα βήματα 7, 8 και 9 για τη δεύτερη λυχνία πράσινου χρώματος χρησιμοποιώντας τον ακροδέκτη **9** του μικροελεγκτή.
11. Επαναλάβετε τα βήματα 7, 8 και 9 για την τρίτη λυχνία κίτρινου χρώματος χρησιμοποιώντας τον ακροδέκτη **10** του μικροελεγκτή.
12. Επαναλάβετε τα βήματα 7, 8 και 9 για την τέταρτη λυχνία πορτοκαλί χρώματος χρησιμοποιώντας τον ακροδέκτη **11** του μικροελεγκτή.
13. Επαναλάβετε τα βήματα 7, 8 και 9 για την πέμπτη και τελευταία λυχνία κόκκινου χρώματος χρησιμοποιώντας τον ακροδέκτη **12** του μικροελεγκτή.
14. Τέλος, συνδέστε τον προσαρμογέα (συσσωρευτή 9V με ακροδέκτη 5.5/2.1) με το συσσωρευτή και με την κατάλληλη υποδοχή του μικροελεγκτή για να τον τροφοδοτήσετε.

Ακολουθώντας τα παραπάνω βήματα το φασαριόμετρο θα πρέπει να λειτουργεί.

2.2.4 Περιγραφή Λειτουργίας

Στο πρώτο μέρος του κώδικα ορίζονται οι σταθερές, οι μεταβλητές που θα χρησιμοποιηθούν και δίνονται οι αρχικές τιμές τους. Εδώ έχουμε περισσότερες σταθερές σε σχέση με τη προηγούμενη κατασκευή του ηχοδιακόπτη.

Οι σταθερές είναι:

- **SampleTime**
Η διάρκεια χρονικού διαστήματος Μέτρησης με αρχική τιμή 100 millisecs.
- **LightsOnTime**
Η διάρκεια χρονικού διαστήματος που παραμένουν αναμμένες οι λυχνίες με αρχική τιμή 100 millisecs.
- **Gain**
Προσαύξηση τιμής στάθμης από Κορυφή σε Κορυφή (PeakToPeak) με αρχική τιμή 200 μονάδες.
- **MicPin**
Ο ακροδέκτης σύνδεσης του μικροφώνου με αρχική τιμή A5.
- **BluePin**
Ο ακροδέκτης σύνδεσης της γαλάζιας λυχνίας LED με αρχική τιμή 8.
- **GreenPin**
Ο ακροδέκτης σύνδεσης της πράσινης λυχνίας LED με αρχική τιμή 9.
- **YellowPin**
Ο ακροδέκτης σύνδεσης της κίτρινης λυχνίας LED με αρχική τιμή 10.
- **OrangePin**
Ο ακροδέκτης σύνδεσης της πορτοκαλί λυχνίας LED με αρχική τιμή 11.
- **RedPin**
Ο ακροδέκτης σύνδεσης της κόκκινης λυχνίας LED με αρχική τιμή 12.

Οι μεταβλητές (μόνο δυο) είναι:

- **SignalMic**
Το σήμα από το μικρόφωνο το οποίο είναι τιμές από 0 - 1023.
- **NoiseValue**
Η στάθμη του θορύβου η οποία έχει πέντε επίπεδα και παίρνει τιμές από 0 - 4.

Στη ρουτίνα Setup, που τρέχει μόνο μια φορά, γίνεται η απαραίτητη αρχικοποίηση, δηλαδή δηλώνεται ποιοι από τους ακροδέκτες θα λειτουργούν ως είσοδοι και ποιοι ως έξοδοι. Εδώ οι ακροδέκτες **8** ως **12** θα δηλωθούν ως έξοδοι ενώ ο **A5** θα δηλωθεί ως είσοδος. Επίσης, ξεκίνα η σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ και ως οπτική ένδειξη εκκίνησης του προγράμματος με χρήση της εντολής FOR ανάβουν οι λυχνίες 2 φορές πάνω – κάτω.

Η ρουτίνα **Loop**, σαν ένας ατέρμονας βρόχος, εκτελείται ξανά και ξανά για όσο χρόνο λειτουργεί το πρόγραμμα και σε αυτήν γράφεται ο κύριος κώδικας κάθε προγράμματος.

Στη συγκεκριμένη περίπτωση, με το ξεκίνημα ορίζονται τέσσερις επιπλέον τοπικές μεταβλητές:

- **startMillis**
Αρχή χρονικού διαστήματος μέτρησης με αρχική τιμή την τρέχουσα τιμή της **millis()**.
- **PeakToPeak**
Έυρος σήματος μικροφώνου από Κορυφή σε Κορυφή με αρχική τιμή 0.
- **signalMax**
Η αρχική μέγιστη τιμή σήματος του μικροφώνου με αρχική τιμή 0.
- **signalMin**
Η αρχική ελάχιστη τιμή σήματος του μικροφώνου με αρχική τιμή 1024.

Στη συνέχεια, με μια εντολή **while** ορίζεται ένα χρονικό διάστημα ίσο με τη σταθερά **sampleTime** στη διάρκεια του οποίου καταμετράται το σήμα του μικροφώνου, αποθηκεύεται η Μέγιστη και η Ελάχιστη τιμή του και υπολογίζεται από τα προηγούμενα το Εύρος από Κορυφή σε Κορυφή **PeakToPeak**. Επίσης, προσαυξάνεται η τιμή του, αθροίζοντας τη σταθερά **Gain**. Η σταθερά αυτή είναι ίση με διακόσιες μονάδες καθώς αντιστοιχεί αποτελεσματικότερα στο εύρος των 1023 του σήματος του μικροφώνου και αποκτά μεγαλύτερη ευαισθησία. Ακολουθώς, μετατρέπεται το εύρος στη μεταβλητή **NoiseValue** (Στάθμη Θορύβου) αντιστοιχώντας τα 1023 σε 5 επίπεδα.

Τέλος, εκτυπώνονται οι μεταβλητές **signalMin**, **signalMax**, **PeakToPeak**, **NoiseValue** στην οθόνη του Η/Υ, ταξινομείται η μεταβλητή Στάθμη Θορύβου (**NoiseValue**) με την εντολή **Switch (case)** ανάμεσα στα πέντε όρια που ορίστηκαν και με αποτέλεσμα να ανάβει τις αντίστοιχες λυχνίες για όσο χρόνο ορίζεται στη μεταβλητή **LightsOnTime**.

Παρατηρήσεις:

- Προτείνεται να δοκιμάσετε διάφορες τιμές στους χρόνους **sampleTime**, **LightsOnTime** και στη προσαύξηση τιμής **Gain** στο δικό σας χώρο και με τις αντίστοιχες συνθήκες προκειμένου να πειραματιστείτε. Με αυτόν τον τρόπο, θα μελετήσετε τη μεταβολή της συμπεριφοράς της κατασκευής και θα έχετε στο τέλος την επιθυμητή ευαισθησία του φασαριόμετρου σας. Γι' αυτό τον λόγο άλλωστε αυτές οι σταθερές ορίζονται στην αρχή του κώδικα προγραμματισμού.

- Μπορείτε να προσθέσετε περισσότερες λυχνίες (π.χ. άλλες πέντε) κάνοντας και τις απαραίτητες μετατροπές στον κώδικα, έτσι ώστε να αυξήσετε τα επίπεδα της στάθμης θορύβου που μετρά το Φασαριόμετρο και να έχετε μεγαλύτερη ακρίβεια στις μετρήσεις σας.

The screenshot shows the Arduino IDE with the sketch '2_FASARIOMETRO' open. The sketch defines pin constants for various LEDs (Blue, Green, Yellow, Orange, Red) and a microphone pin (A5). It also defines variables for SignalMin, SignalMax, PeakToPeak, and NoiseValue. The setup function initializes the pins. The serial monitor (COM4) displays the output of the sketch, showing the signal levels for each LED and the noise value. The output is as follows:

```
SignalMin : 292 SignalMax : 328
PeakToPeak : 236 NoiseValue : 0 Άναψε μόνο τη γαλάζια λυχνία.
=====
SignalMin : 258 SignalMax : 384
PeakToPeak : 326 NoiseValue : 1 Άναψε ως την πράσινη λυχνία.
=====
SignalMin : 35 SignalMax : 605
PeakToPeak : 770 NoiseValue : 3 Άναψε ως την πορτοκαλί λυχνία.
=====
SignalMin : 114 SignalMax : 516
PeakToPeak : 602 NoiseValue : 2 Άναψε ως την κίτρινη λυχνία.
=====
SignalMin : 184 SignalMax : 417
PeakToPeak : 433 NoiseValue : 1 Άναψε ως την πράσινη λυχνία.
=====
SignalMin : 76 SignalMax : 655
PeakToPeak : 779 NoiseValue : 3 Άναψε ως την πορτοκαλί λυχνία.
=====
SignalMin : 0 SignalMax : 662
PeakToPeak : 862 NoiseValue : 3 Άναψε ως την πορτοκαλί λυχνία.
=====
SignalMin : 0 SignalMax : 620
PeakToPeak : 820 NoiseValue : 3 Άναψε ως την πορτοκαλί λυχνία.
=====
SignalMin : 0 SignalMax : 619
PeakToPeak : 819 NoiseValue : 3 Άναψε ως την πορτοκαλί λυχνία.
=====
```

Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.

Σχήμα 2.2.4

2.2.5 Κώδικας Προγραμματισμού

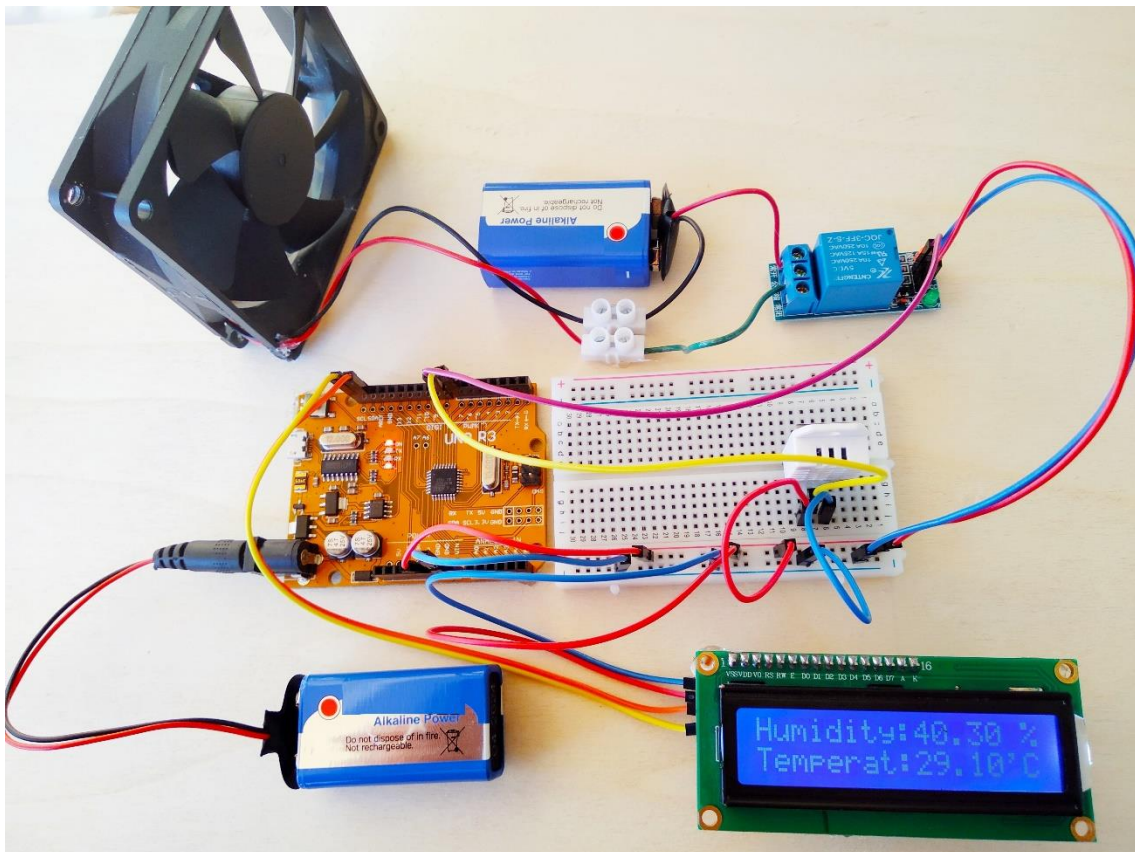
Ο κώδικας παρατίθεται στο Παράρτημα.

Επιπλέον, μπορείτε να τον κατεβάσετε σε αρχείο πατώντας [εδώ](#).

2.3 Έλεγχος Θερμοκρασίας – Υγρασίας Θερμοκηπίου.

2.3.1 Εισαγωγή

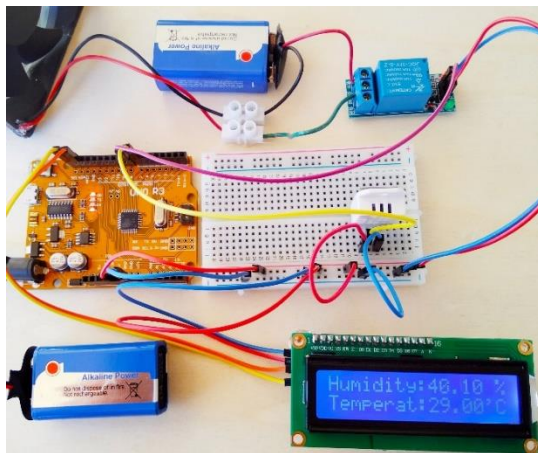
Η τρίτη κατά σειρά κατασκευή είναι μια προσομοίωση λειτουργίας ενός θερμοκηπίου και πιο συγκεκριμένα ο έλεγχος της θερμοκρασίας και της σχετικής υγρασίας στο εσωτερικό του. Οι θερμοκηπιακές καλλιέργειες είναι πολύ διαδεδομένες στο τόπο μας και πολλοί ντόπιοι έχουν αρκετές αντίστοιχες εμπειρίες, οπότε η κατασκευή αυτή πολύ πιθανόν να κεντρίσει το ενδιαφέρον των μαθητών.



Η κατασκευή της προσομοίωσης ελέγχου
θερμοκρασίας και σχετικής υγρασίας ενός θερμοκηπίου.

Σχήμα 2.3.1.α

Η κατασκευή στηρίζεται στον αισθητήρα DHT-22, ο οποίος μετρά τη θερμοκρασία σε βαθμούς Κελσίου και τη σχετική υγρασία της ατμόσφαιρας επί τοις εκατό. Τα δεδομένα στέλνονται σε ένα μικροελεγκτή Arduino Uno και απεικονίζονται σε μια οθόνη χαρακτήρων 16 στηλών και 2 γραμμών και στην οθόνη του Η/Υ ταυτόχρονα μέσω σειριακής επικοινωνίας. Στον μικροελεγκτή επίσης είναι συνδεδεμένος ένας ανεμιστήρας μέσω μιας πλακέτας ηλεκτρονόμου ο οποίος ψύχει το θερμοκήπιο προστατεύοντας τα φυτά.



Η κατασκευή της προσομοίωσης
ελέγχου θερμοκρασίας και σχετικής
υγρασίας ενός θερμοκηπίου.

Σχήμα 2.3.1.β

Η κατασκευή αυτή έχει ως σκοπό να προσομοιώσει τους αυτοματισμούς που προστατεύουν τα φυτά στις θερμοκηπιακές καλλιέργειες, καθώς αυτά είναι ιδιαίτερα ευαίσθητα σε ακραίες τιμές θερμοκρασίας και υγρασίας και κάθε είδος πρέπει να βρίσκεται σε ένα ιδανικό ή τουλάχιστον αποδεκτό εύρος τιμών για να επιβιώσει.










Με βάση τα παραπάνω η κατασκευή λειτουργεί ως εξής:

Έστω ότι έχουμε ένα θερμοκήπιο όπου τα φυτά που καλλιεργούνται πρέπει να βρίσκονται συνεχώς κάτω από 35 βαθμούς Κελσίου και κάτω από το 80% σχετικής υγρασίας της ατμόσφαιρας (οι τιμές αυτές μπορούν να αλλάξουν πολύ εύκολα από τον κώδικα προγραμματισμού). Το σύστημα λοιπόν μετρά αυτά τα δύο μεγέθη ανά τακτά χρονικά διαστήματα (τα οποία έχουν οριστεί στον κώδικα), τα τυπώνει στην οθόνη και ελέγχει αν έχει ξεπεράσει η θερμοκρασία ή η υγρασία τα προκαθορισμένα μέγιστα όρια. Αν υπερβεί τα όρια έστω και ένα από αυτά τα δύο, τότε ενεργοποιείται ο ανεμιστήρας ο οποίος με τον φρέσκο αέρα ψύχει την ατμόσφαιρα και μειώνει την

υγρασία στο προσομοιωμένο θερμοκήπιο. Η κατασκευή αυτή μπορεί να υλοποιηθεί στη σχολική τάξη σε διάφορες παραλλαγές, π.χ. να κατασκευάσουν ένα ψυγείο ή κάποιο κύκλωμα ελέγχου της κεντρικής θέρμανσης μιας οικίας.

2.3.2 Υλικά Κατασκευής

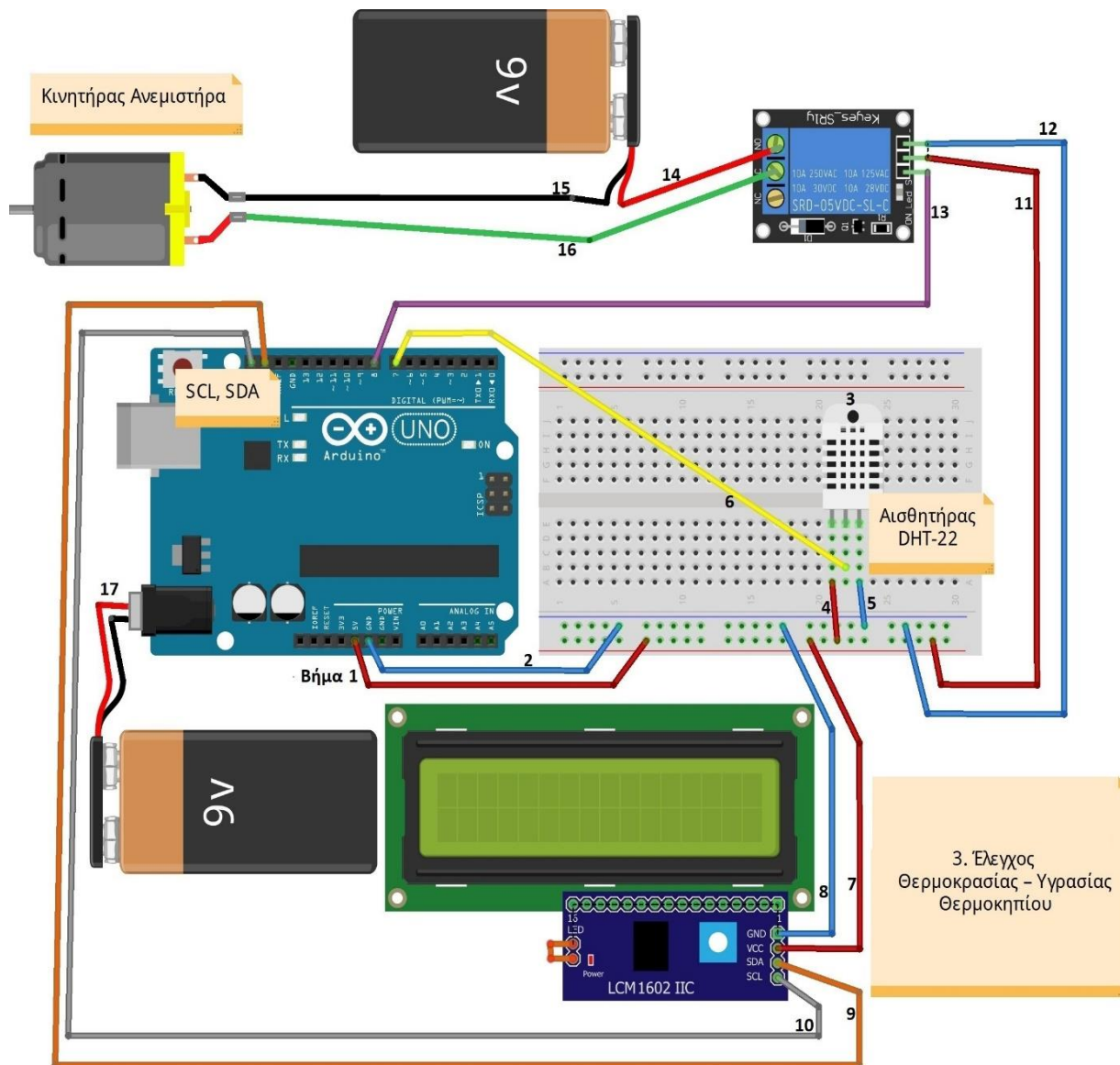
Τα απαραίτητα υλικά για την κατασκευή καταγράφονται στον παρακάτω πίνακα:

Μικροελεγκτής Arduino Uno.		1 τεμάχιο
Αισθητήρας Υγρασίας – Θερμοκρασίας DHT22.		1 τεμάχιο
Ηλεκτρονόμος (1 μεταγωγική επαφή).		1 τεμάχιο
Ανεμιστήρας		1 τεμάχιο
Οθόνη χαρακτήρων (I2C, 16x2).		1 τεμάχιο
Συσσωρευτής (Μπαταρία) 9V.		2 τεμάχια
Πλακέτα Δοκιμών (Breadboard).		1 τεμάχιο
Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1		2 τεμάχια
Διάφοροι Αγωγοί (jumper wires – male to male).		7 τεμάχια
Πίνακας 2.3.2		

2.3.3 Περιγραφή Κατασκευής

Αρχικά, εγκαταστήστε τις βιβλιοθήκες "DHT", "LiquidCrystal_I2C" (Θα τις εντοπίσετε κάνοντας μια απλή αναζήτηση στο διαδίκτυο. Μεταφορτώστε τες ως συμπιεσμένα αρχεία τύπου zip, εγκαταστήστε τες μέσω της σχετικής εντολής από το μενού "Σχέδιο/Συμπερίληψη βιβλιοθήκης/Προσθήκη βιβλιοθήκης ZIP" του Arduino IDE). Μεταφορτώστε το πρόγραμμα στον μικροελεγκτή από το περιβάλλον προγραμματισμού Arduino IDE.

Στη συνέχεια, για την κατασκευή του κυκλώματος, όπως απεικονίζεται στο Σχήμα 2.3.3, υλοποιήστε τα παρακάτω βήματα:



fritzing

Το κύκλωμα της προσομοίωσης ελέγχου θερμοκρασίας - υγρασίας θερμοκηπίου.
Οι αριθμοί είναι τα βήματα για τη κατασκευή του.

Σχήμα 2.3.3

1. Συνδέστε το ένα άκρο ενός αγωγού (jumper wire - male to male) κόκκινου χρώματος με τον ακροδέκτη 5V του μικροελεγκτή και το άλλο άκρο με μια από

- τις οπές της **κάτω θετικής ράγας τροφοδοσίας** (οριζόντια κόκκινη γραμμή) της πλακέτας δοκιμών (Breadboard).
2. Συνδέστε το ένα άκρο ενός αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** (γείωση) του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** (γαλάζια γραμμή) της πλακέτας δοκιμών (Breadboard).
 3. Καρφώστε τον αισθητήρα DHT-22 στην πλακέτα δοκιμών (Breadboard), οι τρεις ακροδέκτες σε τρεις διαδοχικές **πεντάδες οπών** (κάθετης γραμμής), όπως φαίνεται στο σχήμα 2.3.3.
 4. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με μια οπή από την **πεντάδα οπών** (κάθετης γραμμής) με την οποία συνδέθηκε ο ακροδέκτης **Vcc** του αισθητήρα και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
 5. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε ο ακροδέκτης **GND** του αισθητήρα και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
 6. Συνδέστε το ένα άκρο ενός αγωγού π.χ. κίτρινου χρώματος με τον ακροδέκτη **7** του μικροελεγκτή και το άλλο άκρο με την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε ο ακροδέκτης **OUT** του αισθητήρα.
 7. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **Vcc** της οθόνης και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
 8. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** της οθόνης και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
 9. Συνδέστε το ένα άκρο ενός αγωγού π.χ. πορτοκαλί χρώματος με τον ακροδέκτη **SDA** του μικροελεγκτή και το άλλο άκρο με τον ακροδέκτη **SDA** της οθόνης.
 10. Συνδέστε το ένα άκρο ενός αγωγού π.χ. γκρι χρώματος με τον ακροδέκτη **SCL** του μικροελεγκτή και το άλλο άκρο με τον ακροδέκτη **SCL** της οθόνης.
 11. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **Vcc** του ηλεκτρονόμου (ρελέ) και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.

12. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** του ηλεκτρονόμου και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
13. Συνδέστε το ένα άκρο ενός αγωγού π.χ. ιώδους χρώματος με τον ακροδέκτη **IN1** του ηλεκτρονόμου και το άλλο άκρο με τον ακροδέκτη **8** του μικροελεγκτή.
14. Συνδέστε τον προσαρμογέα συσσωρευτή 9V και βιδώστε το κόκκινο άκρο του στον ακροδέκτη **N.O.** (Normally Open) του ηλεκτρονόμου με κατσαβίδι
15. Συνδέστε το μαύρο άκρο του προσαρμογέα με το **πρώτο άκρο** του κινητήρα του ανεμιστήρα (μαύρο άκρο στο σχήμα).
16. Βιδώστε το ένα άκρο ενός αγωγού π.χ. πράσινου χρώματος στον ακροδέκτη **C** (Common) του ηλεκτρονόμου με κατσαβίδι ενώ το άλλο άκρο του συνδέστε το με το **δεύτερο άκρο** του κινητήρα του ανεμιστήρα (κόκκινο άκρο στο σχήμα).
17. Τέλος, συνδέστε τον προσαρμογέα (συσσωρευτή 9V με ακροδέκτη 5.5/2.1) με το συσσωρευτή και με την κατάλληλη υποδοχή του μικροελεγκτή για να τον τροφοδοτήσετε.

Ακολουθώντας τα παραπάνω βήματα η κατασκευή θα πρέπει να λειτουργεί.

2.3.4 Περιγραφή Λειτουργίας

Στην αρχή του κώδικα έχουν ενσωματωθεί τρεις βιβλιοθήκες **Wire.h**, **LiquidCrystal_I2C.h**, **DHT.h**, απαραίτητες για την εύκολη διαχείριση του αισθητήρα και της οθόνης.

Έχουμε τις εξής σταθερές:

- **MaxHumidity**
Το όριο τιμής Μέγιστης Σχετικής Υγρασίας το οποίο καθορίστηκε στο 80%.
- **MaxTemperature**
Το όριο τιμής Μέγιστης Θερμοκρασίας το οποίο καθορίστηκε στο 35 °C .
- **FanPin**
Ο ακροδέκτης σύνδεσης του ανεμιστήρα ο οποίος είναι ο 8.
- **DhtPin**
Ο ακροδέκτης σύνδεσης του αισθητήρα ο οποίος είναι ο 7.
- **DHTType**
Καθορισμός τύπου αισθητήρα σε DHT22 (AM2302).

Οι μεταβλητές είναι μόνο δυο:

- **Hum**
Σχετική Υγρασία επί τοις εκατό.

- **Temp**

Θερμοκρασία σε βαθμούς Κελσίου.

Οι μεταβλητές αυτές δηλώνονται ως τύπου **float**, ώστε να υποστηρίζονται οι δεκαδικοί αριθμοί στις τιμές τους. Ακολουθώς, έχουμε την αρχικοποίηση της οθόνης με τον καθορισμό των ακροδεκτών σύνδεσης, διεύθυνσης κτλ.

The screenshot shows the Arduino IDE with the sketch '3_THERMOKHPIO' and the serial monitor window open. The code in the sketch includes headers for Wire, LiquidCrystal_I2C, and DHT, and defines constants for pin numbers and DHT type. The serial monitor displays a series of humidity and temperature readings.

```

3_THERMOKHPIO | Arduino 1.8.5
File Edit Sketch Tools Help

3_THERMOKHPIO
/*****
3. Έλεγχος Θερμοκρασίας - Υγρασίας Θερμοκηπίου.
*****/

/**** Ενσωμάτωση βιβλιοθηκών. ****/
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"

/**** Δήλωση Σταθερών, Μεταβλητών, κτλ. ****/
const int MaxHumidity = 80 ; // Καθορισμός ορίου υ
const int MaxTemperature = 35 ; // Καθορισμός ορίου τ
const int FanPin = 8 ; // Ο ακροδέκτης σύνδε
const int DhtPin = 7 ; // Ο ακροδέκτης σύνδε
#define DHTType DHT22 // Καθορισμός τύπου ο
DHT dht(DhtPin, DHTType); // Καθορισμός τύπου κ
float Hum ; // Σχετική Υγρασία απ
float Temp ; // Θερμοκρασία από α
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3,POSITIVE);

/**** Ορισμός εισόδων, εξόδων, αρχικοποιήσεις κτλ. (τ
Done uploading.
Sketch uses 7674 bytes (23%) of program storage space.
Global variables use 615 bytes (30%) of dynamic memory,

```

Serial Monitor Output:

```

Humidity: 65.80 % Temperature: 27.50 *C
Humidity: 66.10 % Temperature: 27.50 *C
Humidity: 66.10 % Temperature: 27.50 *C
Humidity: 66.20 % Temperature: 27.50 *C
Humidity: 66.20 % Temperature: 27.50 *C
Humidity: 65.90 % Temperature: 27.50 *C
Humidity: 65.90 % Temperature: 27.50 *C
Humidity: 65.50 % Temperature: 27.60 *C
Humidity: 65.50 % Temperature: 27.60 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 64.70 % Temperature: 27.50 *C
Humidity: 64.70 % Temperature: 27.50 *C
Humidity: 64.40 % Temperature: 27.50 *C
Humidity: 64.40 % Temperature: 27.50 *C
Humidity: 64.20 % Temperature: 27.50 *C
Humidity: 64.20 % Temperature: 27.50 *C
Humidity: 64.40 % Temperature: 27.50 *C
Humidity: 64.80 % Temperature: 27.50 *C
Humidity: 64.80 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 65.20 % Temperature: 27.50 *C
Humidity: 65.20 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C
Humidity: 65.10 % Temperature: 27.50 *C

```

Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.

Σχήμα 2.3.4

Στη ρουτίνα Setup γίνεται η απαραίτητη αρχικοποίηση. Εδώ ο ακροδέκτης του ανεμιστήρα **FanPin**, δηλαδή ο 8, ορίζεται ως έξοδος και δίνεται στην επόμενη γραμμή τιμή HIGH ώστε να έχουμε σίγουρα εκκίνηση του προγράμματος με απενεργοποιημένο ανεμιστήρα. Ξεκίνα η σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ, έχουμε εκκίνηση και αρχικοποίηση της βιβλιοθήκης του αισθητήρα, εκκίνηση και αρχικοποίηση της βιβλιοθήκης της οθόνης χαρακτήρων Lcd για 16

στήλες 2 γραμμές και ενεργοποίηση του οπίσθιου φωτισμού της οθόνης. Στη συνέχεια έχουμε την εκτύπωση του αρχικού μηνύματος το οποίο είναι:

**-*- ELEGXOS -*-
THERMOKHPIOY**

Το πρόγραμμα καθυστερεί δυο δευτερόλεπτα ώστε ο χρήστης να προλάβει να διαβάσει το μήνυμα πριν καθαρίσει η οθόνη.

Η ρουτίνα Loop ξεκινά με την μέτρηση από τον αισθητήρα των τιμών των δυο μεταβλητών δηλαδή της σχετικής υγρασίας επί τοις εκατό (**Hum**) και της θερμοκρασίας σε βαθμούς Κελσίου (**Temp**). Στη συνέχεια με μια εντολή IF γίνεται έλεγχος για το αν η μέτρηση από τον αισθητήρα είναι επιτυχής. Αν δεν είναι, επιστρέφει και προσπαθεί ξανά.

Αν η ανάγνωση είναι επιτυχής, συνεχίζει το πρόγραμμα με δεύτερη IF, η οποία ελέγχει αν η Μέγιστη Υγρασία ή η Μέγιστη Θερμοκρασία είναι πάνω από τα όρια που έχουν οριστεί ως σταθερές στην αρχή (**MaxHumidity, MaxTemperature**). Αν ένα από τα δυο έχει υπερβεί το όριο, τότε ο ανεμιστήρας ενεργοποιείται, διαφορετικά απενεργοποιείται. Ακολούθως εκτυπώνεται η υγρασία και η θερμοκρασία στην οθόνη του H/Y και στην οθόνη χαρακτήρων lcd. Τέλος, το πρόγραμμα αναμένει 1000 millisecs πριν ξαναμετρήσει.

Παρατηρήσεις:

- Προτείνεται να δοκιμάσετε διαφορετικά όρια για τη Μέγιστη Υγρασία και τη Μέγιστη Θερμοκρασία (**MaxHumidity, MaxTemperature**), οι οποίες ορίζονται ως σταθερές στην αρχή του κώδικα, ώστε να προσαρμοστεί η κατασκευή στις καιρικές συνθήκες κάτω από τις οποίες υλοποιείται.
- Ένας απλός τρόπος για να αυξήσετε γρήγορα την υγρασία και τη θερμοκρασία, προκειμένου να παρατηρήσετε τη συμπεριφορά της κατασκευής, είναι να κλείσετε τον αισθητήρα ανάμεσα στις παλάμες σας (χούφτα) για μερικά δευτερόλεπτα.

2.3.5 Κώδικας Προγραμματισμού

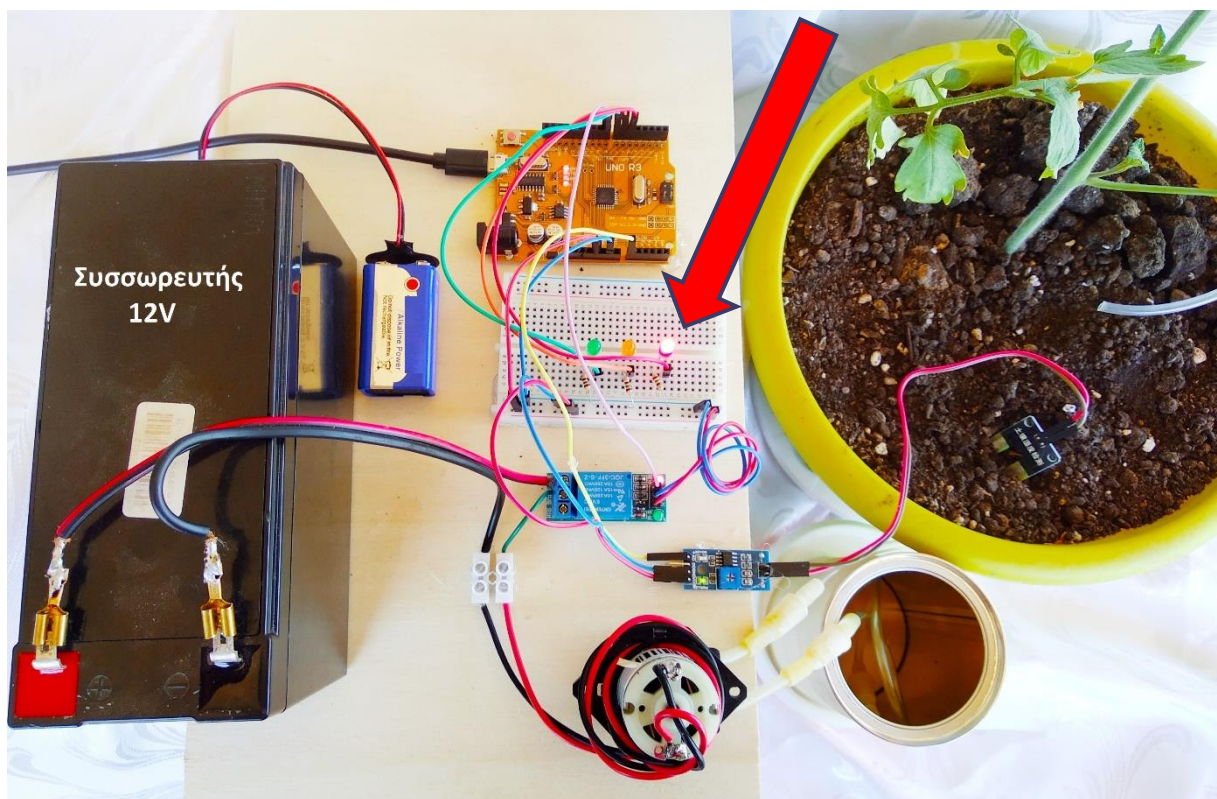
Ο κώδικας παρατίθεται στο Παράρτημα.

Επιπλέον, μπορείτε να τον κατεβάσετε ως αρχείο πατώντας [εδώ](#).

2.4 Αυτόματο Πότισμα Γλάστρας.

2.4.1 Εισαγωγή

Ένα πολύ συνηθισμένο πρόβλημα για όσους έχουν γλάστρες με λουλούδια, λαχανικά, βότανα κ.α. είναι αυτό του ποτίσματος. Άλλες φορές τα φυτά ποτίζονται πολύ συχνά, άλλες λιγότερο από αυτό που χρειάζονται ενώ άλλες φορές ο ιδιοκτήτης λείπει σε ταξίδι και τα φυτά μένουν άνοδρα για μέρες. Η λύση σε αυτό το πρόβλημα είναι μια κατασκευή που ποτίζει όσο πρέπει τα φυτά αυτόματα

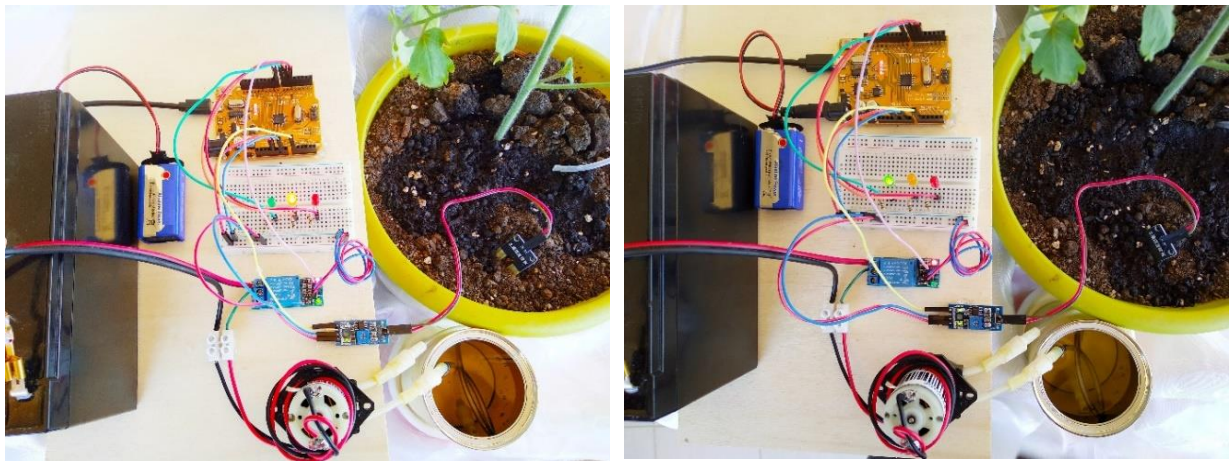


Η κατασκευή για το αυτόματο πότισμα γλάστρας.
Το χώμα θεωρείται "Ξηρό" οπότε ανάβει η κόκκινη λυχνία.

Σχήμα 2.4.1.α

Η συγκεκριμένη κατασκευή εστιάζει σε μια γλάστρα μόνο, λειτουργεί με έναν αισθητήρα υγρασίας εδάφους και μια μικρή αντλία που συνδέονται (όπως και οι προηγούμενες κατασκευές) με ένα Arduino Uno. Αν η γλάστρα δεν έχει αρκετή

υγρασία στο χώμα της, ενεργοποιείται η αντλία για μερικά δευτερόλεπτα έτσι ώστε να στείλει νερό από ένα δοχείο για να ποτιστεί το χώμα. Στη συνέχεια, το πρόγραμμα περιμένει για ένα χρονικό διάστημα μερικών δευτερολέπτων ανάμεσα στις μετρήσεις, καθώς το νερό που παρέχεται σε κάθε πότισμα χρειάζεται λίγο χρόνο για να διαχυθεί στο χώμα και να ανεβάσει τη μετρήσιμη σχετική υγρασία. Στην επόμενη μέτρηση αν η υγρασία παραμένει χαμηλή, το πρόγραμμα επαναλαμβάνει τη διαδικασία ποτίζοντας τη γλάστρα και περιμένοντας για την επόμενη μέτρηση. Η διαδικασία επαναλαμβάνεται μέχρι να αλλάξουν τα επίπεδα υγρασίας του χώματος. Αν η μέτρηση δείξει ότι η γλάστρα έχει ποτιστεί επαρκώς, τότε το πρόγραμμα μπαίνει σε μια μεγαλύτερη αναμονή πριν μετρήσει ξανά. Για εκπαιδευτικούς λόγους εδώ ο χρόνος αναμονής έχει οριστεί ένα λεπτό, όμως σε πραγματικές συνθήκες μπορεί να οριστεί π.χ. μερικές ώρες ή μέχρι και μια μέρα.



Αριστερά: Η τιμή της υγρασίας στο χώμα δεν είναι η επιθυμητή, βρίσκεται σε μέτριο επίπεδο, τοποθετείται ανάμεσα στο "Ξηρό" και στο "Ποτισμένο", οπότε ανάβει η πορτοκαλί λυχνία.

Δεξιά: Το χώμα θεωρείται "Ποτισμένο" οπότε ανάβει η πράσινη λυχνία.

Σχήμα 2.4.1.β

Η συγκεκριμένη εκπαιδευτική διάταξη μπορεί να αποτελέσει τη βάση για να υλοποιηθεί κάτι μεγαλύτερο και πολυπλοκότερο χρησιμοποιώντας πολλαπλούς αισθητήρες, ηλεκτρονόμους οι οποίοι ελέγχουν ηλεκτροβάνες και αντλίες. Έτσι, μπορεί να εφαρμοστεί για το αυτόματο πότισμα ακόμη και ενός χωραφιού, ενός θερμοκηπίου κ.α.

2.4.2 Υλικά Κατασκευής

Τα απαραίτητα υλικά για την κατασκευή καταγράφονται στον παρακάτω πίνακα:

Μικροελεγκτής Arduino Uno.		1 τεμάχιο
Αισθητήρας Υγρασίας Εδάφους FC-28.		1 τεμάχιο
Ηλεκτρονόμος (1 μεταγωγική επαφή).		1 τεμάχιο
Αντλία.		1 τεμάχιο
Πλακέτα Δοκιμών (Breadboard).		1 τεμάχιο
Λυχνία (Led), 1 Πράσινη, 1 Πορτοκαλί , 1 Κόκκινη.		3 τεμάχια
Συσσωρευτής (Μπαταρία) 9V.		2 τεμάχια
Συσσωρευτής (Μπαταρία) 1,5V.		8 τεμάχια
Αντίσταση 220Ω.		3 τεμάχια
Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1		1 τεμάχιο
Διάφοροι Αγωγοί (jumper wires – male to male).		16 τεμάχια
Δοχείο νερού.		1 τεμάχιο

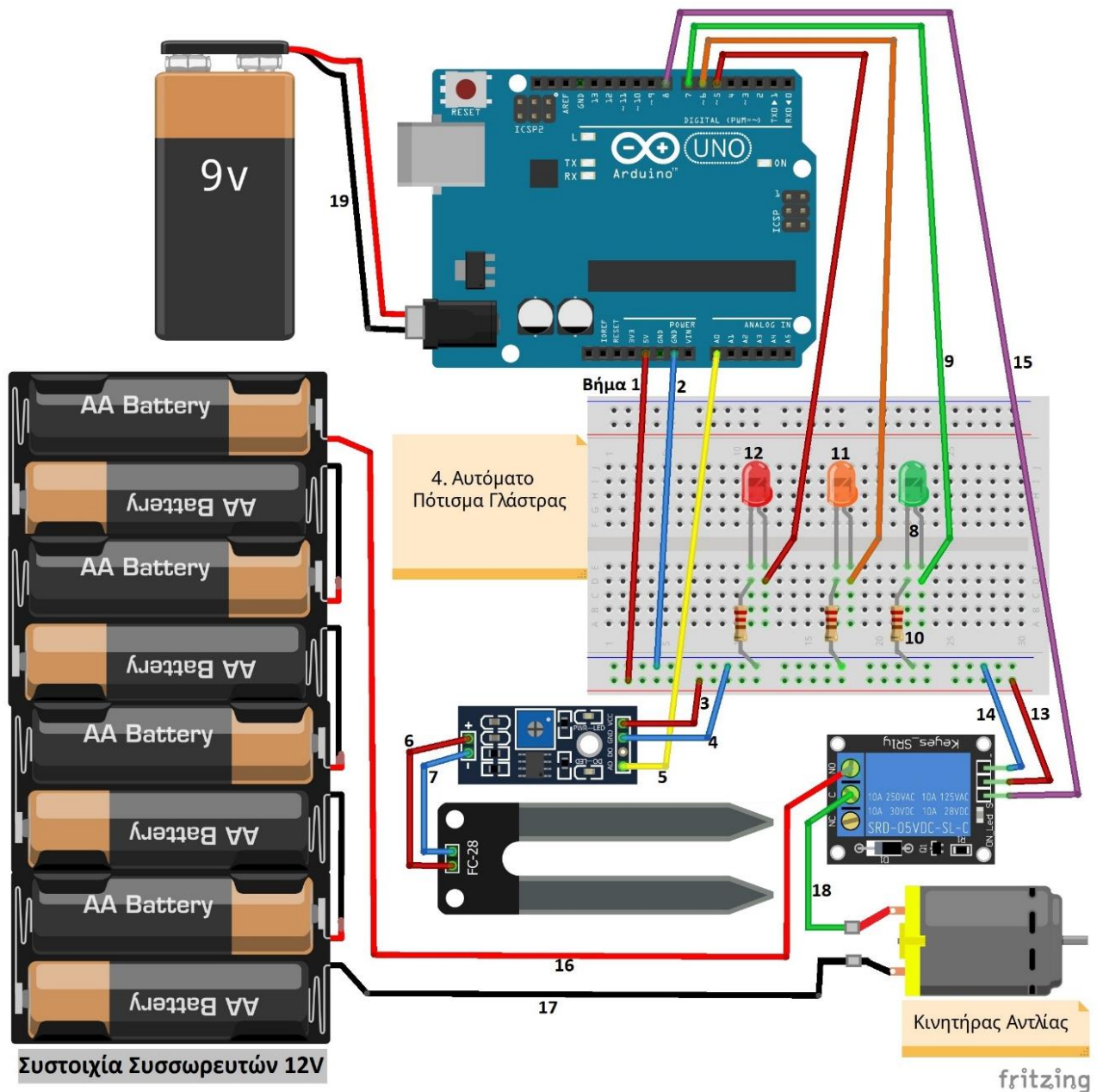
Πίνακας 2.4.2

2.4.3 Περιγραφή Κατασκευής

Για την κατασκευή του κυκλώματος, όπως απεικονίζεται στο Σχήμα 2.4.3 αρχικά μεταφορτώστε το πρόγραμμα στον μικροελεγκτή από το περιβάλλον προγραμματισμού Arduino IDE και στη συνέχεια υλοποιήστε τα παρακάτω βήματα:

1. Συνδέστε το ένα άκρο ενός αγωγού (jumper wire – male to male) κόκκινου χρώματος με τον ακροδέκτη **5V** του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** (οριζόντια κόκκινη γραμμή) της πλακέτας δοκιμών (Breadboard).
2. Συνδέστε το ένα άκρο ενός αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** (γείωση) του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **κάτω**

αρνητικής ράγας τροφοδοσίας (οριζόντια γαλάζια γραμμή της πλακέτας δοκιμών (Breadboard)).



Το κύκλωμα του Αυτόματου Ποτίσματος Γλάστρας.
Οι αριθμοί είναι τα βήματα για τη κατασκευή του.

Σχήμα 2.4.3

3. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **Vcc** της πλακέτας του αισθητήρα υγρασίας εδάφους FC-28 και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
4. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** της πλακέτας του αισθητήρα υγρασίας εδάφους FC-28 και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
5. Συνδέστε το ένα άκρο ενός αγωγού π.χ. κίτρινου χρώματος με τον ακροδέκτη **A0** του μικροελεγκτή και το άλλο άκρο με την Αναλογική Έξοδο **Α0** της πλακέτας του αισθητήρα.
6. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **(+)** της πλακέτας του αισθητήρα υγρασίας εδάφους FC-28 και το άλλο άκρο με τον ακροδέκτη **(+)** του δίχαλου του αισθητήρα.
7. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **(-)** της πλακέτας του αισθητήρα υγρασίας εδάφους FC-28 και το άλλο άκρο με τον ακροδέκτη **(-)** του δίχαλου του αισθητήρα.
8. Καρφώστε την πρώτη λυχνία πράσινου χρώματος στην πλακέτα δοκιμών, με τους δυο ακροδέκτες σε δύο διαδοχικές **πεντάδες οπών** κάθετης γραμμής.
9. Συνδέστε το ένα άκρο ενός άλλου αγωγού πράσινου χρώματος με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε το μακρύτερο άκρο της λυχνίας (**Άνοδος +**) και το άλλο άκρο με τον ακροδέκτη **7** του μικροελεγκτή.
10. Συνδέστε το ένα άκρο μιας αντίστασης 220Ω με μια οπή από την **πεντάδα οπών** της πλακέτας δοκιμών με την οποία συνδέθηκε το πιο κοντό άκρο της λυχνίας (**Κάθοδος -**) και το άλλο άκρο της αντίστασης με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
11. Επαναλάβετε τα βήματα 8, 9 και 10 για τη δεύτερη λυχνία πορτοκαλί χρώματος χρησιμοποιώντας τον ακροδέκτη **6** του μικροελεγκτή.
12. Επαναλάβετε τα βήματα 8, 9 και 10 για την τρίτη λυχνία κόκκινου χρώματος χρησιμοποιώντας τον ακροδέκτη **5** του μικροελεγκτή.
13. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **Vcc** του ηλεκτρονόμου (ρελέ) και το άλλο άκρο με μια από τις οπές της **κάτω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.

14. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** του ηλεκτρονόμου και το άλλο άκρο με μια από τις οπές της **κάτω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
15. Συνδέστε το ένα άκρο ενός αγωγού π.χ. ιώδους χρώματος με τον ακροδέκτη **IN1** του ηλεκτρονόμου και το άλλο άκρο με τον ακροδέκτη **8** του μικροελεγκτή.
16. Βιδώστε το κόκκινο άκρο της συστοιχίας συσσωρευτών 12V (η συστοιχία δημιουργείται συνδέοντας τους οκτώ συσσωρευτές των 1,5V μεταξύ τους **σε σειρά**) στον ακροδέκτη **N.O.** (Normally Open) του ηλεκτρονόμου με κατσαβίδι.
17. Συνδέστε το μαύρο άκρο της συστοιχίας συσσωρευτών 12V με το ένα **άκρο** (μαύρο χρώμα) του κινητήρα της αντλίας.
18. Βιδώστε το ένα άκρο ενός αγωγού π.χ. πράσινου χρώματος στον ακροδέκτη **C** (Common) του ηλεκτρονόμου με κατσαβίδι ενώ το άλλο άκρο του συνδέστε το με το δεύτερο **άκρο** (κόκκινο χρώμα) του κινητήρα της αντλίας.
19. Τέλος, συνδέστε τον προσαρμογέα (συσσωρευτή 9V με ακροδέκτη 5.5/2.1) με το συσσωρευτή και με την κατάλληλη υποδοχή του μικροελεγκτή για να τον τροφοδοτήσετε.

Ακολουθώντας τα παραπάνω βήματα η κατασκευή θα πρέπει να λειτουργεί.

2.4.4 Περιγραφή Λειτουργίας.

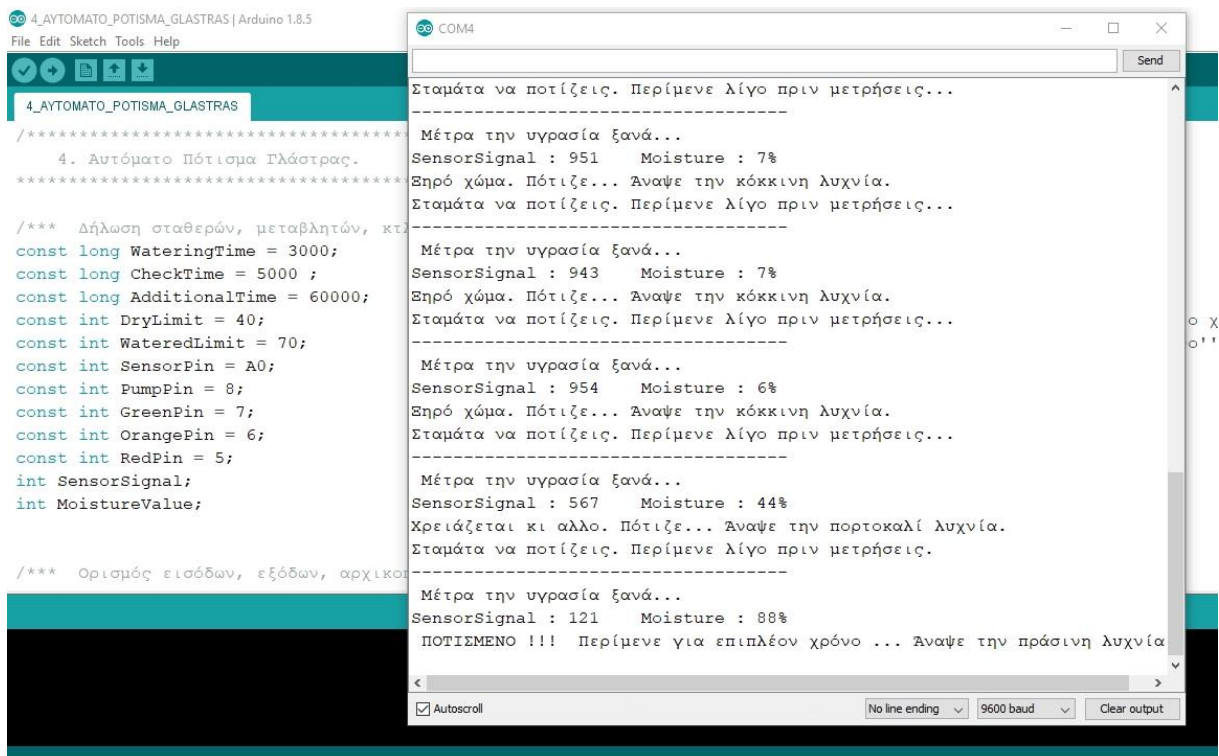
Αρχικά στον κώδικα προγραμματισμού έχουν δηλωθεί οι παρακάτω σταθερές:

- **WateringTime**
Ο χρόνος ποτίσματος με αρχική τιμή 3000 (milliseconds).
- **CheckTime**
Ο χρόνος αναμονής πριν μετρήσει ξανά σε κάθε κύκλο προγράμματος με αρχική τιμή 5000 (milliseconds).
- **AdditionalTime**
Ο επιπλέον χρόνος αναμονής όταν είναι "Ποτισμένο" το χώμα με αρχική τιμή 60000 (milliseconds), δηλαδή ένα λεπτό της ώρας.
- **DryLimit**
Η τιμή ορίου Σχετικής Υγρασίας κάτω από το οποίο θεωρείται "Ξηρό" το χώμα με αρχική τιμή 40.
- **WateredLimit**
Η τιμή ορίου Σχετικής Υγρασίας πάνω από το οποίο θεωρείται "Ποτισμένο" το χώμα με αρχική τιμή 70.

- **SensorPin**
Ο ακροδέκτης σύνδεσης του αισθητήρα ο οποίος είναι ο A0.
- **PumpPin**
Ο ακροδέκτης σύνδεσης της αντλίας ο οποίος είναι ο 8.
- **GreenPin**
Ο ακροδέκτης σύνδεσης της πράσινης λυχνίας LED ο οποίος είναι ο 7.
- **OrangePin**
Ο ακροδέκτης σύνδεσης της πορτοκαλί λυχνίας LED ο οποίος είναι ο 6.
- **RedPin**
Ο ακροδέκτης σύνδεσης της κόκκινης λυχνίας LED ο οποίος είναι ο 5.

Οι δυο μεταβλητές είναι:

- **SensorSignal**
Η τιμή σήματος από τον αισθητήρα.
- **MoistureValue**
Η τιμή Σχετικής Υγρασίας επί τοις εκατό (%).



Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.

Σχήμα 2.4.4

Στη ρουτίνα Setup γίνεται η απαραίτητη αρχικοποίηση. Ο ακροδέκτης του αισθητήρα ορίζεται ως είσοδος ενώ ο ακροδέκτης της αντλίας ορίζεται ως έξοδος και δίνεται στην επόμενη γραμμή τιμή HIGH (με HIGH ο ηλεκτρονόμος αυτός παραμένει αφοπλισμένος) ώστε η αντλία να είναι σίγουρα απενεργοποιημένη στην εκκίνηση. Οι ακροδέκτες σύνδεσης της πράσινης, πορτοκαλί και κόκκινης λυχνίας ορίζονται ως έξοδοι. Ξεκινά η σειριακή επικοινωνία και εκτυπώνεται το μήνυμα εκκίνησης στην οθόνη του Η/Υ. Επίσης, ως ένδειξη εκκίνησης του προγράμματος (όταν δεν υπάρχει Η/Υ) αποτελεί το άναμμα των λυχνιών 2 φορές πάνω - κάτω.

Η ρουτίνα Loop ξεκινά με την ανάγνωση του σήματος από τον αισθητήρα, την αποθήκευση της τρέχουσας τιμής του στη μεταβλητή **SignalValue** και τη μετατροπή του σε Σχετική Υγρασία επί τοις εκατό (**MoistureValue**). Στη συνέχεια, εκτυπώνονται οι τιμές του σήματος του αισθητήρα και της Σχετικής Υγρασίας (%) στην οθόνη του Η/Υ. Ακολουθεί η κατάταξη της τιμής υγρασίας σε τρεις διαφορετικές περιπτώσεις και η εκτέλεση των ανάλογων εντολών.

Στην πρώτη περίπτωση **αν** η υγρασία είναι μικρότερη ή ίση από το όριο “Ξηρό”, ενεργοποιείται η αντλία (οπλίζει ο ηλεκτρονόμος με LOW), εκτυπώνεται ένα μήνυμα στην οθόνη του Η/Υ, ανάβει μόνο η κόκκινη λυχνία, περιμένει (ποτίζοντας ωστόσο) για χρόνο **WateringTime** και στη συνέχεια σταματά να ποτίζει εκτυπώνοντας το ανάλογο μήνυμα στην οθόνη του Η/Υ.

Διαφορετικά, **αν** (δεύτερη περίπτωση) η Σχετική Υγρασία (%) είναι μεγαλύτερη από το όριο “Ξηρό” **και** ταυτόχρονα μικρότερη από το όριο “Ποτισμένο”, ενεργοποιείται η αντλία, εκτυπώνεται ένα μήνυμα στην οθόνη του Η/Υ και ανάβει μόνο η πορτοκαλί λυχνία. Στη συνέχεια, ποτίζει για χρόνο **WateringTime** και έπειτα σταματά εκτυπώνοντας το ανάλογο μήνυμα στην οθόνη του Η/Υ.

Αν το χώμα θεωρείται “ποτισμένο” (η Σχετική Υγρασία είναι μεγαλύτερη ή ίση με το 70%), απενεργοποιείται η αντλία, ανάβει μόνο η πράσινη λυχνία και το πρόγραμμα περιμένει για επιπλέον χρόνο (**AdditionalTime**).

Τέλος, το πρόγραμμα περιμένει για χρόνο **CheckTime**, ο οποίος είναι ο χρόνος αναμονής πριν μετρήσει ξανά την υγρασία του εδάφους σε κάθε κύκλο προγράμματος και εκτυπώνει το κατάλληλο μήνυμα στην οθόνη του Η/Υ.

Παρατηρήσεις:

- Προτείνεται να δοκιμάσετε διάφορες τιμές για τις μεταβλητές **WateringTime**, **CheckTime** και **AdditionalTime** οι οποίες αρχικά έχουν τιμές 3 sec, 5 sec και 1 min αντίστοιχα. Αυτές οι τιμές για μια μικρή γλάστρα είναι ικανοποιητικές κατά την εκπαιδευτική δραστηριότητα, καθώς μέσα σε μερικά λεπτά φαίνεται η διαφορά από μια άνυδρη γλάστρα σε μια επαρκώς ποτισμένη. Όπως

προαναφέρθηκε, προσοχή πρέπει να δοθεί στο ότι η γλάστρα πρέπει να ποτίζεται λίγο και το χρονικό διάστημα ανάμεσα στις μετρήσεις πρέπει να είναι ικανοποιητικό, διότι το νερό που παρέχεται σε κάθε πότισμα χρειάζεται λίγο χρόνο για να διαχυθεί στο χώμα και να ανεβάσει τη μετρήσιμη σχετική υγρασία.

- Αν χρησιμοποιήσετε κάποια παραλλαγή αυτής της κατασκευής σε πραγματικές συνθήκες, πρέπει να φροντίσετε πάντα να έχετε χαμηλή τάση από συσσωρευτές (προτείνονται έως 12VDC) και το ηλεκτρικό κύκλωμα (με την προσθήκη διακοπών, ασφαλειών κτλ.) να είναι στεγανό προκειμένου να αποφευχθούν βραχυκυκλώματα και ατυχήματα. Καλό είναι να συμβουλευτείτε έναν γεωπόνο για την τιμή της ιδανικής υγρασίας του χώματος ανάλογα με το φυτό και τις επιθυμητές συνθήκες στο χώρο σας.

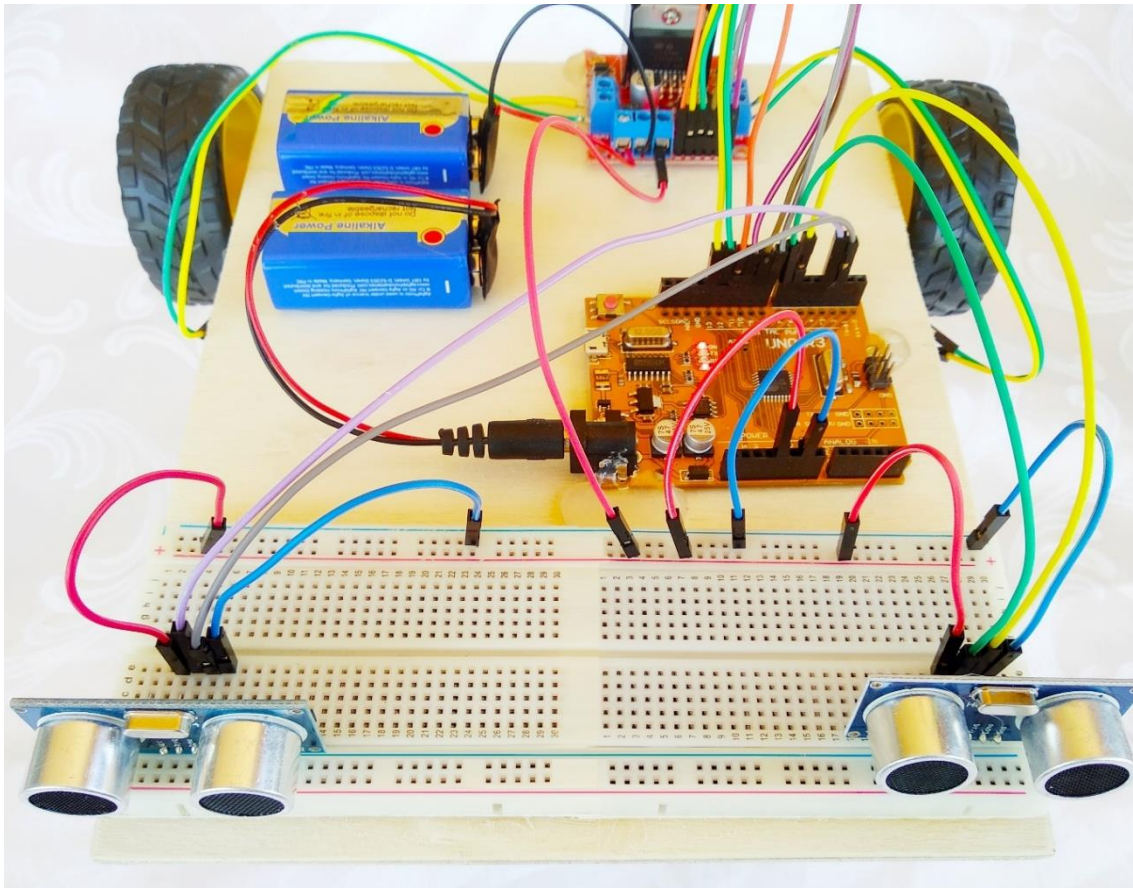
2.4.5 Κώδικας Προγραμματισμού.

Ο κώδικας παρατίθεται στο Παράρτημα.

Επιπλέον, μπορείτε να τον κατεβάσετε ως αρχείο πατώντας [εδώ](#).

2.5 Ρομπότ Αποφυγής Εμποδίων.

Η τελευταία κατασκευή που παρουσιάζεται σε αυτό το εγχειρίδιο είναι η κατασκευή ενός Ρομπότ το οποίο κινείται αυτόνομα στο χώρο αποφεύγοντας οποιαδήποτε εμποδία μπορούν να εμφανιστούν μπροστά του.

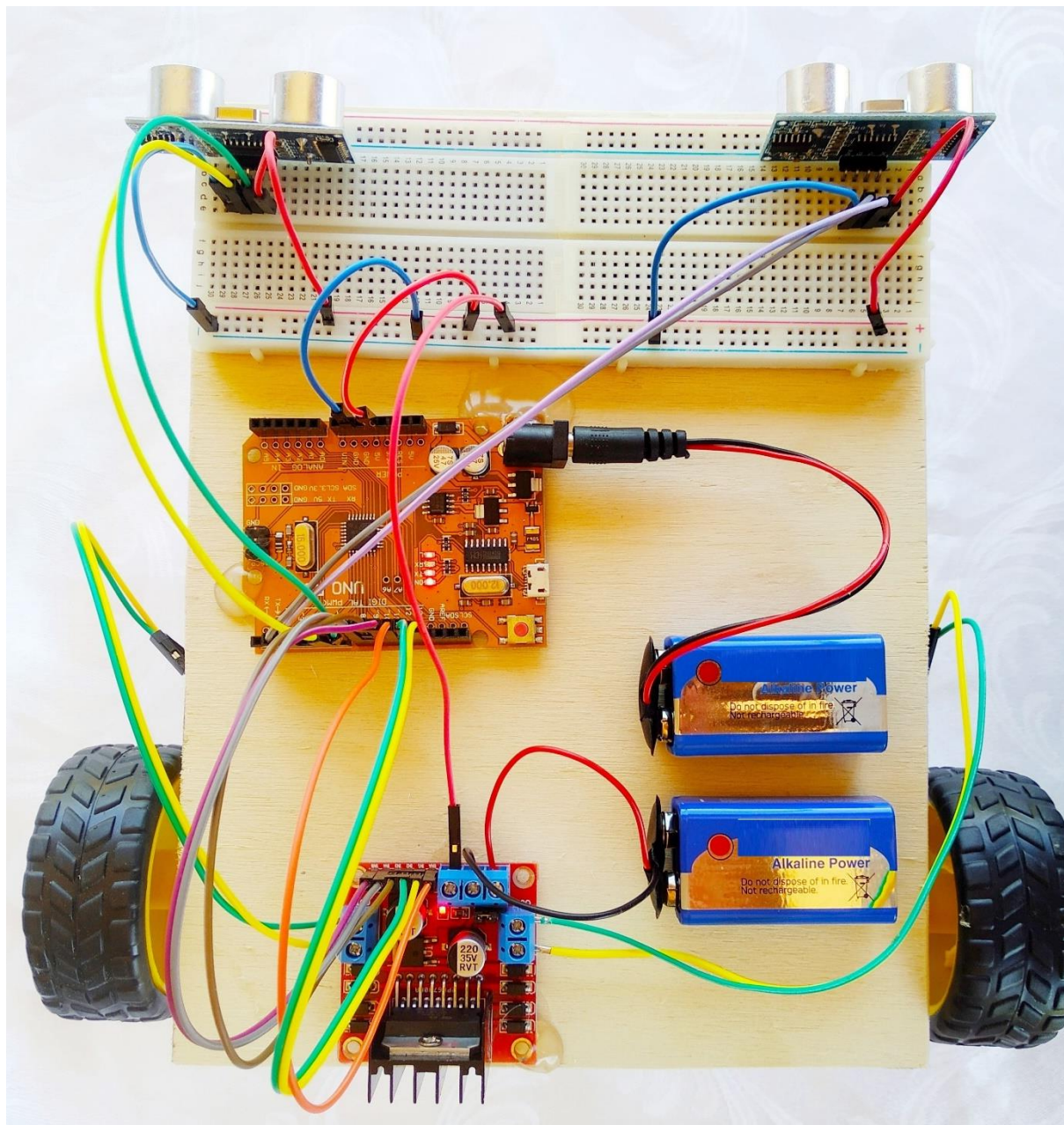


Η κατασκευή του Ρομπότ.

Σχήμα 2.5.1.α

Χρησιμοποιείται, όπως και στις προηγούμενες κατασκευές, η πλακέτα Arduino Uno R3 (κλώνος) με δυο αισθητήρες υπερήχων για την ανίχνευση των εμποδίων αριστερά και δεξιά. Το Ρομπότ οδηγείται από δυο τροχούς με ενσωματωμένους κινητήρες που ελέγχονται μέσω μιας Η-γέφυρας από το Arduino Uno. Όλα τα εξαρτήματα μαζί με μια ελεύθερα περιστρεφόμενη ρόδα τοποθετούνται και κολλιούνται με θερμόκολλα σε ένα κομμάτι κόντρα πλακέ (θα μπορούσε να είναι μακετόχαρτο, χαρτόκουτα

συσκευασίας ή κάτι ανάλογο) πάχους 4mm και διαστάσεων περίπου 20x30cm, το οποίο παίρνει το ρόλο του αμαξώματος για να διαμορφώσει το τρίκυκλο που επιθυμούμε.

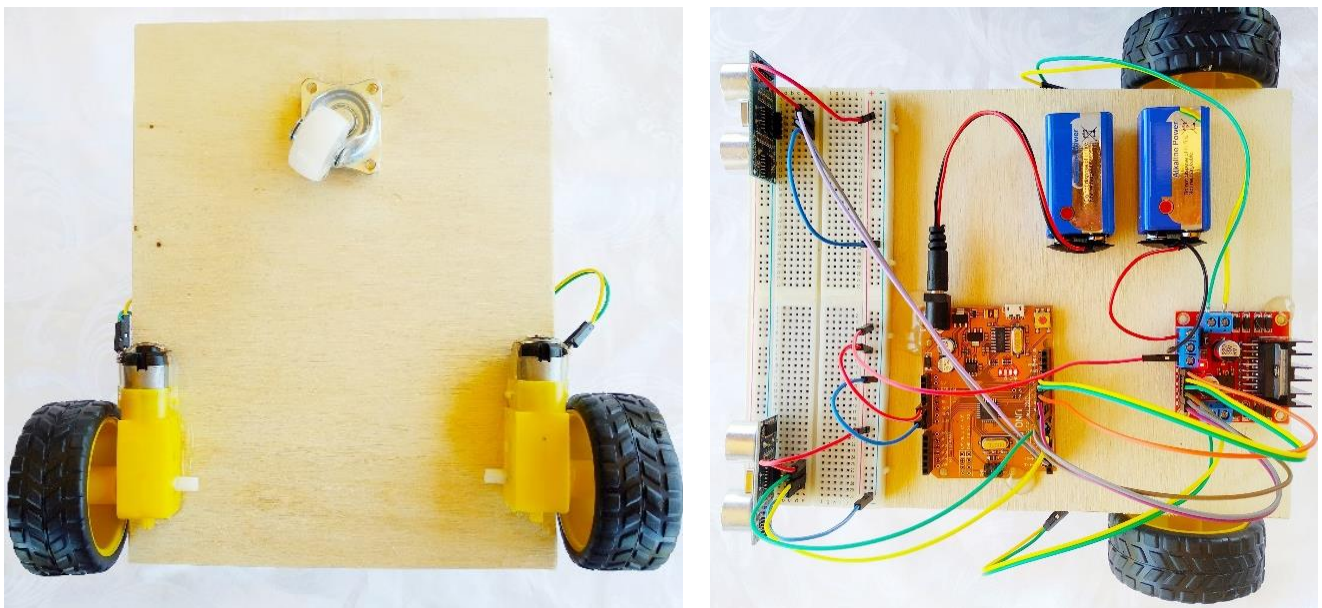


Η κατασκευή του Ρομπότ.

Σχήμα 2.5.1.β

Οι αισθητήρες τοποθετούνται στο μπροστινό δεξιό και αριστερό μέρος ελέγχοντας συνεχώς την απόσταση των εμποδίων. Αν το Ρομπότ εντοπίσει ένα αντικείμενο μπροστά του σε απόσταση μικρότερη από ένα ελάχιστο όριο, το οποίο έχει προκαθοριστεί στο κώδικα προγραμματισμού, σταματά και κατευθύνεται λίγο προς τα πίσω ώστε να απομακρυνθεί από το εμπόδιο.

Στη συνέχεια, ελέγχει τις δυο αποστάσεις και ψάχνει να βρει τον καλύτερο δυνατό τρόπο για να γυρίσει. Αν έχει περισσότερο χώρο αριστερά, τότε στρίβει προς τα αριστερά, αν έχει περισσότερο χώρο δεξιά, τότε στρίβει προς τα δεξιά, αλλάζοντας έτσι συνεχώς την πορεία του αριστερά ή δεξιά ανάλογα με τα εμπόδια που θα βρεθούν στο δρόμο του.














Η κατασκευή του Ρομπότ.

Σχήμα 2.5.1.γ

2.5.2 Υλικά Κατασκευής

Τα απαραίτητα υλικά για την κατασκευή καταγράφονται στον παρακάτω πίνακα:

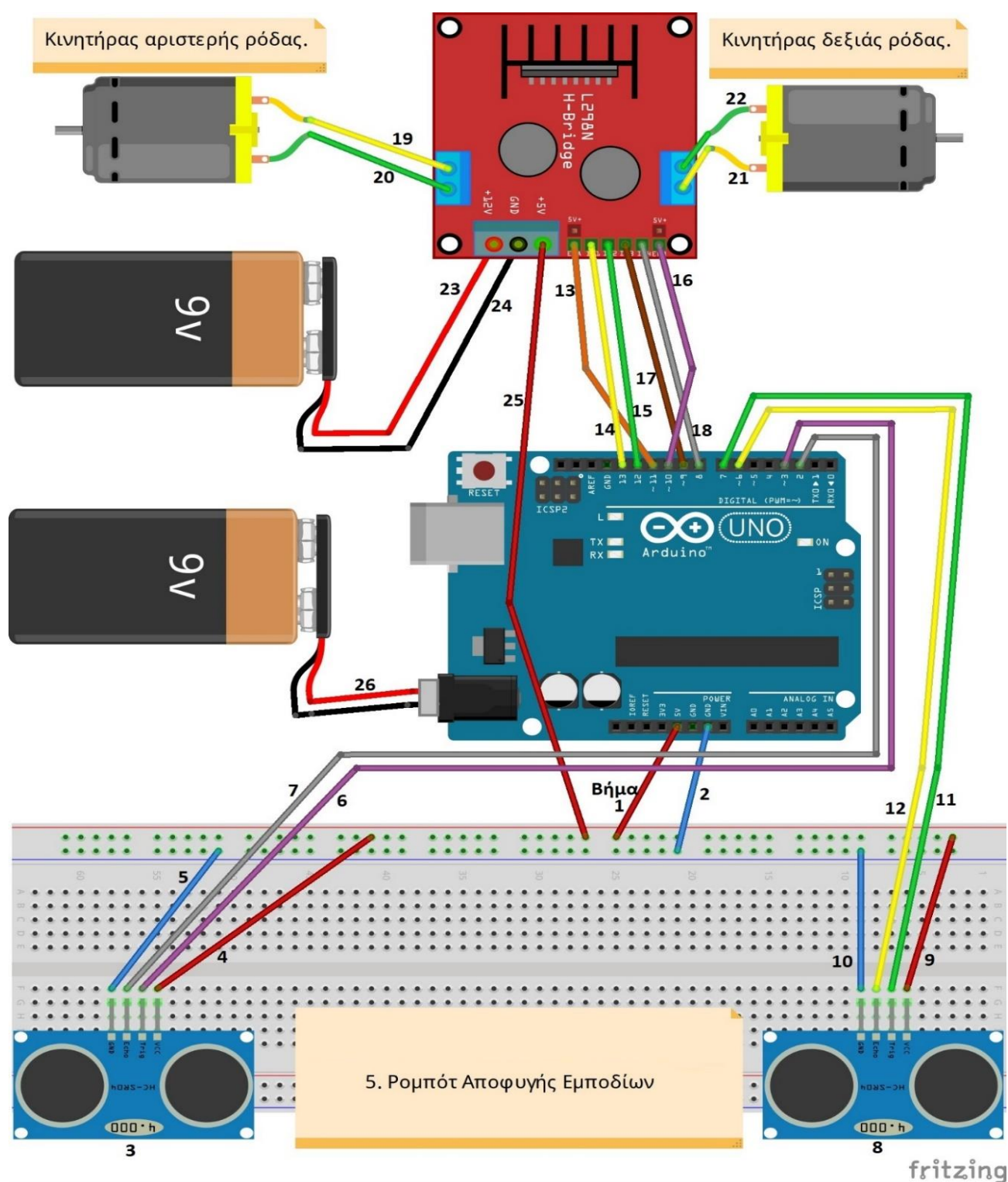
Μικροελεγκτής Arduino Uno.		1 τεμάχιο
Η-γέφυρα L298N		1 τεμάχιο
Εξάρτημα Ρόδας-Κινητήρα.		2 τεμάχια
Αισθητήρας Υπερήχων HC-SR04.		2 τεμάχια
Πλακέτα Δοκιμών.		1 τεμάχιο
Συσσωρευτής 9V.		2 τεμάχια
Προσαρμογέας συσσωρευτή 9V σε ακροδέκτη 5.5/2.1		2 τεμάχια
Διάφοροι Αγωγοί. (Jumper wires, male to male, male to female).		24 τεμάχια
Ξύλο, κόντρα πλακέ πάχους 4mm (περίπου 20x30cm).		1 τεμάχιο
Ελεύθερα περιστρεφόμενη ρόδα.		1 τεμάχιο
Πιστόλι Θερμόκολλας και Ράβδοι Σιλικόνης.		1 τεμάχιο

Πίνακας 2.5.2

2.5.3 Περιγραφή Κατασκευής

Για την κατασκευή του κυκλώματος, όπως απεικονίζεται στο Σχήμα 2.5.3 αρχικά μεταφορτώστε το πρόγραμμα στον μικροελεγκτή από το περιβάλλον προγραμματισμού Arduino IDE και στη συνέχεια υλοποιήστε τα παρακάτω βήματα:

1. Συνδέστε το ένα άκρο ενός αγωγού (jumper wire – male to male) κόκκινου χρώματος με τον ακροδέκτη **5V** του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **πάνω θετικής ράγας τροφοδοσίας** (οριζόντια κόκκινη γραμμή) της πλακέτας δοκιμών (Breadboard).



Το κύκλωμα του ρομπότ.
Οι αριθμοί είναι τα βήματα για τη κατασκευή του.

Σχήμα 2.5.3

2. Συνδέστε το ένα άκρο ενός αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** (γείωση) του μικροελεγκτή και το άλλο άκρο με μια από τις οπές της **πάνω αρνητικής ράγας τροφοδοσίας** (οριζόντια γαλάζια γραμμή της πλακέτας δοκιμών (Breadboard)).
3. Καρφώστε τον **αριστερό** αισθητήρα υπερήχων HC-SR04 στην πλακέτα δοκιμών (Breadboard), οι τέσσερις ακροδέκτες σε τέσσερις διαδοχικές **πεντάδες οπών** (κάθετης γραμμής), όπως φαίνεται στο σχήμα 2.5.3.
4. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **Vcc** της πλακέτας του **αριστερού** αισθητήρα και το άλλο άκρο με μια από τις οπές της **πάνω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
5. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** της πλακέτας του **αριστερού** αισθητήρα και το άλλο άκρο με μια από τις οπές της **πάνω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
6. Συνδέστε το ένα άκρο ενός αγωγού π.χ. ιώδους χρώματος με τον ακροδέκτη **Trigger** της πλακέτας του **αριστερού** αισθητήρα και το άλλο άκρο με τον ακροδέκτη **3** του μικροελεγκτή.
7. Συνδέστε το ένα άκρο ενός αγωγού π.χ. γκρι χρώματος με τον ακροδέκτη **Echo** της πλακέτας του **αριστερού** αισθητήρα και το άλλο άκρο με τον ακροδέκτη **2** του μικροελεγκτή.
8. Καρφώστε τον **δεξιό** αισθητήρα υπερήχων HC-SR04 στην πλακέτα δοκιμών (Breadboard), οι τέσσερις ακροδέκτες σε τέσσερις διαδοχικές **πεντάδες οπών** (κάθετης γραμμής), όπως φαίνεται στο σχήμα.
9. Συνδέστε το ένα άκρο ενός άλλου αγωγού κόκκινου χρώματος με τον ακροδέκτη **Vcc** της πλακέτας του **δεξιού** αισθητήρα και το άλλο άκρο με μια από τις οπές της **πάνω θετικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
10. Συνδέστε το ένα άκρο ενός άλλου αγωγού γαλάζιου χρώματος με τον ακροδέκτη **GND** της πλακέτας του **δεξιού** αισθητήρα και το άλλο άκρο με μια από τις οπές της **πάνω αρνητικής ράγας τροφοδοσίας** της πλακέτας δοκιμών.
11. Συνδέστε το ένα άκρο ενός αγωγού π.χ. πράσινου χρώματος με τον ακροδέκτη **Trigger** της πλακέτας του **δεξιού** αισθητήρα και το άλλο άκρο με τον ακροδέκτη **7** του μικροελεγκτή.
12. Συνδέστε το ένα άκρο ενός αγωγού π.χ. κίτρινου χρώματος με τον ακροδέκτη **Echo** της πλακέτας του **δεξιού** αισθητήρα και το άλλο άκρο με τον ακροδέκτη **6** του μικροελεγκτή.

13. Συνδέστε το ένα άκρο ενός αγωγού (jumper wire – male to female) π.χ. πορτοκαλί χρώματος με τον ακροδέκτη **ΕΝΑ** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον ακροδέκτη **11** του μικροελεγκτή.
14. Συνδέστε το ένα άκρο ενός αγωγού π.χ. κίτρινου χρώματος με τον ακροδέκτη **IN1** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον ακροδέκτη **13** του μικροελεγκτή.
15. Συνδέστε το ένα άκρο ενός αγωγού π.χ. πράσινου χρώματος με τον ακροδέκτη **IN2** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον ακροδέκτη **12** του μικροελεγκτή.
16. Συνδέστε το ένα άκρο ενός αγωγού π.χ. ιώδους χρώματος με τον ακροδέκτη **ΕΝΒ** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον ακροδέκτη **10** του μικροελεγκτή.
17. Συνδέστε το ένα άκρο ενός αγωγού π.χ. καφέ χρώματος με τον ακροδέκτη **IN3** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον ακροδέκτη **9** του μικροελεγκτή.
18. Συνδέστε το ένα άκρο ενός αγωγού π.χ. γκρι χρώματος με τον ακροδέκτη **IN4** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον ακροδέκτη **8** του μικροελεγκτή.
19. Βιδώστε (με κατσαβίδι) το ένα άκρο ενός αγωγού π.χ. κίτρινου χρώματος με τον ακροδέκτη **OUT1** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο συνδέστε το με τον έναν ακροδέκτη του κινητήρα της **αριστερής** ρόδας.
20. Βιδώστε (με κατσαβίδι) το ένα άκρο ενός αγωγού π.χ. πράσινου χρώματος με τον ακροδέκτη **OUT2** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο με τον άλλον **ακροδέκτη του κινητήρα** της αριστερής ρόδας.
21. Βιδώστε (με κατσαβίδι) το ένα άκρο ενός αγωγού π.χ. κίτρινου χρώματος με τον ακροδέκτη **OUT4** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο συνδέστε το με τον έναν **ακροδέκτη του κινητήρα** της δεξιάς ρόδας.
22. Βιδώστε (με κατσαβίδι) το ένα άκρο ενός αγωγού π.χ. πράσινου χρώματος με τον ακροδέκτη **OUT3** της πλακέτας της Η-γέφυρας L298N και το άλλο άκρο συνδέστε το με τον άλλον **ακροδέκτη του κινητήρα** της δεξιάς ρόδας.
23. Συνδέστε τον προσαρμογέα του πρώτου συσσωρευτή 9V με ακροδέκτη 2.2 και το **κόκκινο άκρο** του βιδώστε το στον ακροδέκτη **+12V** της πλακέτας της Η-γέφυρας L298N με κατσαβίδι.
24. Βιδώστε το **μαύρο άκρο** του προσαρμογέα στον ακροδέκτη **GND** της πλακέτας της Η-γέφυρας L298N με κατσαβίδι.
25. Βιδώστε στον ακροδέκτη **+5V** της πλακέτας της Η-γέφυρας L298N το ένα άκρο ενός αγωγού π.χ. κόκκινου χρώματος με κατσαβίδι και το άλλο άκρο συνδέστε

το με μια από τις οπές της **πάνω θετικής ράγας τροφοδοσίας** (οριζόντια κόκκινη γραμμή) της πλακέτας δοκιμών.

26. Τέλος, συνδέστε τον προσαρμογέα (συσσωρευτή 9V με ακροδέκτη 5.5/2.1) με το συσσωρευτή και με την κατάλληλη υποδοχή του μικροελεγκτή για να τον τροφοδοτήσετε.

Ακολουθώντας τα παραπάνω βήματα το Ρομπότ θα πρέπει να λειτουργεί.

2.5.4 Περιγραφή Λειτουργίας

Στην αρχή του κώδικα προγραμματισμού δηλώνονται οι σταθερές οι οποίες είναι οι παρακάτω:

- **SpeedLeftMotor**
Τιμή (PWM) που καθορίζει τη ταχύτητα του αριστερού κινητήρα (0 ως 255), με αρχική τιμή 120.
- **SpeedRightMotor**
Τιμή (PWM) που καθορίζει τη ταχύτητα του δεξιού κινητήρα (0 ως 255), με αρχική τιμή 120.
- **StopTime**
Ο χρόνος αναμονής του Ρομπότ σταματημένο (milliseconds), με αρχική τιμή 700.
- **TurnTime**
Ο χρόνος που το Ρομπότ στρίβει (milliseconds), με αρχική τιμή 500.
- **BackwardTime**
Ο χρόνος που το Ρομπότ κινείται προς τα πίσω (milliseconds), με αρχική τιμή 800.
- **LeftMinObstDist**
Ελάχιστη απόσταση εμποδίου από αριστερό αισθητήρα (cm), με αρχική τιμή 35.
- **RightMinObstDist**
Ελάχιστη απόσταση εμποδίου από δεξιό αισθητήρα (cm) με αρχική τιμή 35.
- **TrigPinL**
Ο ακροδέκτης σύνδεσης Trigger του αριστερού αισθητήρα, ο οποίος είναι ο 3.
- **EchoPinL**
Ο ακροδέκτης σύνδεσης Echo του αριστερού αισθητήρα, ο οποίος είναι ο 2.
- **TrigPinR**
Ο ακροδέκτης σύνδεσης Trigger του δεξιού αισθητήρα, ο οποίος είναι ο 7.
- **EchoPinR**
Ο ακροδέκτης σύνδεσης Echo του δεξιού αισθητήρα, ο οποίος είναι ο 6.

- **IN1**
Ο ακροδέκτης σύνδεσης της Εισόδου 1 της Η-γέφυρας ο οποίος είναι ο 13.
- **IN2**
Ο ακροδέκτης σύνδεσης της Εισόδου 2 της Η-γέφυρας ο οποίος είναι ο 12.
- **SpeedPinL**
Ο ακροδέκτης ελέγχου ταχύτητας αριστερού κινητήρα (ENA) ο οποίος είναι ο 11.
- **IN3**
Ο ακροδέκτης σύνδεσης της Εισόδου 3 της Η-γέφυρας ο οποίος είναι ο 9.
- **IN4**
Ο ακροδέκτης σύνδεσης της Εισόδου 4 της Η-γέφυρας ο οποίος είναι ο 8.
- **SpeedPinR**
Ο ακροδέκτης ελέγχου ταχύτητας δεξιού κινητήρα (ENB) ο οποίος είναι ο 10.

The screenshot shows the Arduino IDE interface with the sketch '5_ROBOT_APOFYGHIS_EMPODIWN' loaded. The code defines various motor speeds, times, and pin numbers. The serial monitor on the right shows the output of the program, which includes distance measurements and status messages.

```

5_ROBOT_APOFYGHIS_EMPODIWN | Arduino 1.8.5
File Edit Sketch Tools Help

5_ROBOT_APOFYGHIS_EMPODIWN

/***** 5. Ρομπότ Αποφυγής Εμποδίων. *****/

/**** Δήλωση σταθερών, μεταβλητών, κτλ. ****/
const int SpeedLeftMotor = 120; // Τιμή (PWM 0-255) που καθορίζει την ταχύτητα του αριστερού κινητήρα
const int SpeedRightMotor = 120; // Τιμή (PWM 0-255) που καθορίζει την ταχύτητα του δεξιού κινητήρα
const int StopTime = 700; // Ο χρόνος αναμονής σταματημού των κινητήρων
const int TurnTime = 500; // Ο χρόνος που στρίβει (milli)
const int BackwardTime = 800; // Ο χρόνος που κινείται προς τα πίσω
const int LeftMinObstDist = 35; // Ελάχιστη απόσταση εμποδίου προς τα αριστερά
const int RightMinObstDist = 35; // Ελάχιστη απόσταση εμποδίου προς τα δεξιά

const int TrigPinL = 3; // Ο ακροδέκτης σύνδεσης Trigger του αριστερού υπερηχητικού αισθητήρα
const int EchoPinL = 2; // Ο ακροδέκτης σύνδεσης Echo του αριστερού υπερηχητικού αισθητήρα
const int TrigPinR = 7; // Ο ακροδέκτης σύνδεσης Trigger του δεξιού υπερηχητικού αισθητήρα
const int EchoPinR = 6; // Ο ακροδέκτης σύνδεσης Echo του δεξιού υπερηχητικού αισθητήρα

const int IN1 = 13; // Ο ακροδέκτης σύνδεσης της Εισόδου 1 της Η-γέφυρας
const int IN2 = 12; // Ο ακροδέκτης σύνδεσης της Εισόδου 2 της Η-γέφυρας
const int SpeedPinL = 11; // Ο ακροδέκτης ελέγχου ταχύτητας αριστερού κινητήρα (ENA)
const int IN3 = 9; // Ο ακροδέκτης σύνδεσης της Εισόδου 3 της Η-γέφυρας

Done uploading.

Sketch uses 4720 bytes (14%) of program storage space. Maximum is 32768 bytes.
Global variables use 590 bytes (28%) of dynamic memory, leaving 2310 bytes for local variables. Maximum is 26384 bytes.

Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!

106cm , 111cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!

56cm , 80cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!

106cm , 111cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!

59cm , 77cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!

56cm , 80cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!
  
```

Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.
Το Ρομπότ δεν ανιχνεύει κανένα εμπόδιο και προχωράει μπροστά.

Σχήμα 2.5.4.α

Επίσης, δηλώνονται οι μεταβλητές οι οποίες εδώ είναι τέσσερις. Ειδικά, οι χρόνοι ορίζονται ως τύπου **long** καθώς είναι αρκετά μεγάλοι αριθμοί.

- **DurationL**
Ο χρόνος που ταξίδεψε ο παλμός από τον αριστερό αισθητήρα στο εμπόδιο και πίσω (microsecs).
- **DistanceL**
Απόσταση εμποδίου από τον αριστερό αισθητήρα (cm).
- **DurationR**
Ο χρόνος που ταξίδεψε ο παλμός από τον δεξιό αισθητήρα στο εμπόδιο και πίσω (microsecs).
- **DistanceR**
Απόσταση εμποδίου από τον δεξιό αισθητήρα (cm).

Στη ρουτίνα Setup, η οποία τρέχει μόνο μια φορά, γίνεται η απαραίτητη αρχικοποίηση, δηλαδή ο ορισμός των εισόδων, των εξόδων κτλ. Έτσι, οι ακροδέκτες **EchoPinL** και **EchoPinR** ορίζονται ως εισοδοί ενώ οι **TrigPinL**, **TrigPinR** των αισθητήρων απόστασης ορίζονται ως εξοδοί. Επίσης, οι ακροδέκτες του Υπο με τους οποίους συνδέονται οι **IN1**, **IN2**, **IN3**, **IN4**, **SpeedPinL**, **SpeedPinR** της Η-γέφυρας ορίζονται ως εξοδοί. Με αυτόν το τρόπο, ξεκινά η σειριακή επικοινωνία.

The screenshot shows the Arduino IDE interface. The sketch is named "5_ROBOT_APOFYGH5_EMPODIWN" and is being uploaded to an Arduino 1.8.5. The serial monitor is open, showing the output of the sketch. The output displays the robot's position and the distance to the obstacle on both the left and right sides. The robot is currently at 115cm, 152cm. The output also indicates that there is no obstacle ahead and that the robot is moving forward.

```

5_ROBOT_APOFYGH5_EMPODIWN
File Edit Sketch Tools Help

5. Ρομπότ Αποφυγής Εμποδίων.

*** Δήλωση σταθερών, μεταβλητών, κτλ. ***
const int SpeedLeftMotor = 120; // Τιμή (PWM 0-255) που καθο
const int SpeedRightMotor = 120; // Τιμή (PWM 0-255) που καθο
const int StopTime = 700; // Ο χρόνος αναμονής σταματη
const int TurnTime = 500; // Ο χρόνος που στρίβει (mil
const int BackwardTime = 800; // Ο χρόνος που κινείται προ
const int LeftMinObstDist = 35; // Ελάχιστη απόσταση εμποδίου
const int RightMinObstDist = 35; // Ελάχιστη απόσταση εμποδίου

const int TrigPinL = 3; // Ο ακροδέκτης σύνδεσης Trigger το
const int EchoPinL = 2; // Ο ακροδέκτης σύνδεσης Echo του α
const int TrigPinR = 7; // Ο ακροδέκτης σύνδεσης Trigger το
const int EchoPinR = 6; // Ο ακροδέκτης σύνδεσης Echo του δ

const int IN1 = 13; // Ο ακροδέκτης σύνδεσης της Εισό
const int IN2 = 12; // Ο ακροδέκτης σύνδεσης της Εισό
const int SpeedPinL = 11; // Ο ακροδέκτης ελέγχου ταχύτητας
const int IN3 = 9; // Ο ακροδέκτης σύνδεσης της Εισό

Done uploading.
Sketch uses 4720 bytes (14%) of program storage space. Max
Global variables use 590 bytes (28%) of dynamic memory,
1

COM4
115cm , 152cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!
=====
113cm , 10cm
Κίνδυνος, εμπόδιο μπροστά! Οι απόστασεις είναι :
Αριστερά: 113 cm --- Δεξιά: 10 cm
Πήγαινε Πίσω!
Stop !
Στρίψε Αριστερά!
Stop !
=====
115cm , 13cm
Κίνδυνος, εμπόδιο μπροστά! Οι απόστασεις είναι :
Αριστερά: 115 cm --- Δεξιά: 13 cm
Πήγαινε Πίσω!
Stop !
Στρίψε Αριστερά!
Stop !
=====
114cm , 154cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!
=====
115cm , 153cm
Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.
Πήγαινε Μπροστά!
=====
Autoscroll No line ending 9600 baud Clear output
204

```

Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.
Το Ρομπότ ανιχνεύει εμπόδιο δεξιά, οπισθοχωρεί και στρίβει αριστερά.

Σχήμα 2.5.4.β

Στην Ρουτίνα Loop, η οποία εκτελείται επαναλαμβανόμενα για όσο λειτουργεί η κατασκευή, καταγράφεται το κύριο πρόγραμμα το οποίο όμως έχει αυξημένη πολυπλοκότητα σε σχέση με τα προηγούμενα παραδείγματα. Γι' αυτόν τον λόγο υπάρχουν έξι υπορουτίνες οι οποίες ενεργοποιούνται όποτε είναι απαραίτητο με αποτέλεσμα ο κώδικας να γίνεται πιο εύκολα κατανοητός.

The screenshot shows the Arduino IDE interface. The main window displays the code for the sketch '5_ROBOT_APOFYGHIS_EMPODIWN'. The code includes comments in Greek and defines various constants for motor speeds, times, and distances. It also defines pin numbers for triggers and echoes. The serial monitor window on the right shows the output of the program, which includes distance readings and status messages in Greek.

```

5_ROBOT_APOFYGHIS_EMPODIWN
5. Ρομπότ Αποφυγής Εμποδίων.

*** Δήλωση σταθερών, μεταβλητών, κτλ. ***
const int SpeedLeftMotor = 120; // Τιμή (PWM 0-255) που καθο
const int SpeedRightMotor = 120; // Τιμή (PWM 0-255) που καθο
const int StopTime = 700; // Ο χρόνος αναμονής σταματη
const int TurnTime = 500; // Ο χρόνος που στρίβει (mil
const int BackwardTime = 800; // Ο χρόνος που κινείται προ
const int LeftMinObstDist = 35; // Ελάχιστη απόσταση εμποδίο
const int RightMinObstDist = 35; // Ελάχιστη απόσταση εμποδίο

const int TrigPinL = 3; // Ο ακροδέκτης σύνδεσης Trigger το
const int EchoPinL = 2; // Ο ακροδέκτης σύνδεσης Echo του α
const int TrigPinR = 7; // Ο ακροδέκτης σύνδεσης Trigger το
const int EchoPinR = 6; // Ο ακροδέκτης σύνδεσης Echo του δ

const int IN1 = 13; // Ο ακροδέκτης σύνδεσης της Εισό
const int IN2 = 12; // Ο ακροδέκτης σύνδεσης της Εισό
const int SpeedPinL = 11; // Ο ακροδέκτης ελέγχου ταχύτητας
const int IN3 = 9; // Ο ακροδέκτης σύνδεσης της Εισό

Done uploading.
Sketch uses 4720 bytes (14%) of program storage space. Memory
Global variables use 590 bytes (28%) of dynamic memory, 1

```

Serial Monitor Output:

```

Δεν υπάρχει εμπόδιο. Πήγαίνε προς τα εμπρός.
Πήγαίνε Μπροστά!
=====
113cm , 130cm
Δεν υπάρχει εμπόδιο. Πήγαίνε προς τα εμπρός.
Πήγαίνε Μπροστά!
=====
114cm , 130cm
Δεν υπάρχει εμπόδιο. Πήγαίνε προς τα εμπρός.
Πήγαίνε Μπροστά!
=====
16cm , 104cm
Κίνδυνος, εμπόδιο μπροστά! Οι απόστασεις είναι :
Αριστερά: 16 cm --- Δεξιά: 104 cm
Πήγαίνε Πίσω!
Stop !
Στρίψε Δεξιά! !
Stop !
=====
11cm , 134cm
Κίνδυνος, εμπόδιο μπροστά! Οι απόστασεις είναι :
Αριστερά: 11 cm --- Δεξιά: 134 cm
Πήγαίνε Πίσω!
Stop !
Στρίψε Δεξιά! !
Stop !
=====
114cm , 130cm
Δεν υπάρχει εμπόδ

```

Στιγμιότυπο από τη σειριακή επικοινωνία του μικροελεγκτή με τον Η/Υ.
Το Ρομπότ ανιχνεύει εμπόδιο αριστερά, οπισθοχωρεί και στρίβει δεξιά.

Σχήμα 2.5.4.γ

Υπάρχουν λοιπόν οι εξής υπορουτίνες:

- **Read_DistanceL**
Υπορουτίνα η οποία υπολογίζει την απόσταση των εμποδίων από τον αριστερό αισθητήρα.
- **Read_DistanceR**
Υπορουτίνα η οποία υπολογίζει την απόσταση των εμποδίων από τον δεξιό αισθητήρα.
- **Turn_Right**

- Υπορουτίνα η οποία στρίβει το Ρομπότ προς τα δεξιά.
- **Turn_Left**
Υπορουτίνα η οποία στρίβει το Ρομπότ προς τα αριστερά.
- **Go_Forward**
Υπορουτίνα η οποία οδηγεί το Ρομπότ προς τα εμπρός.
- **Go_Backward**
Υπορουτίνα η οποία οδηγεί το Ρομπότ προς τα πίσω.

Ξεκινώντας την εκτέλεση των εντολών στη Ρουτίνα Loop, αρχικά καλούνται οι δυο υπορουτίνες υπολογισμού των αποστάσεων των εμποδίων από τον αριστερό και δεξιό αισθητήρα (**Read_DistanceL**, **Read_DistanceR**). Μετά τον υπολογισμό εκτυπώνονται οι τιμές αυτές των αποστάσεων των εμποδίων στην οθόνη του Η/Υ σε εκατοστά του μέτρου (cm).

Στη συνέχεια, με μια εντολή IF ελέγχεται αν η απόσταση του εμποδίου από τον αριστερό **ή** τον δεξιό αισθητήρα είναι μικρότερη από το ελάχιστο όριο. Αν ισχύει αυτό, εκτυπώνεται ένα μήνυμα κινδύνου μαζί με τις αποστάσεις αριστερά - δεξιά στην οθόνη του Η/Υ σε cm και καλείται η υπορουτίνα να οδηγήσει το Ρομπότ λίγο πίσω (**Go_Backward**).

- Σε αυτό το σημείο με μια δεύτερη εντολή IF εντός της πρώτης (φωλιασμένη IF) ελέγχεται αν η απόσταση του εμποδίου αριστερά είναι μεγαλύτερη από αυτήν δεξιά. Αν ισχύει αυτή η συνθήκη, τότε καλείται η αντίστοιχη υπορουτίνα (**Turn_Left**) και το Ρομπότ στρίβει αριστερά (δηλαδή προς τα εκεί όπου υπάρχει χώρος), αλλιώς στρίβει δεξιά καλώντας την αντίστοιχη υπορουτίνα (**Turn_Right**).

Επανερχόμενη στην πρώτη εντολή IF, αν δεν ισχύει η συνθήκη η οποία ελέγχθηκε (αν δηλαδή η απόσταση του εμποδίου από τον αριστερό **ή** δεξιό αισθητήρα είναι μικρότερη από το ελάχιστο όριο) τότε προχωρά το πρόγραμμα στην εκτέλεση των επόμενων εντολών μετά το ELSE, δηλαδή τυπώνει ένα μήνυμα στην οθόνη του Η/Υ ότι δεν υπάρχει εμπόδιο και μπορεί να κινηθεί προς τα εμπρός. Επομένως, καλείται η αντίστοιχη υπορουτίνα (**Go_Forward**) και στην τελευταία γραμμή του κώδικα υπάρχει μια καθυστέρηση 1ms για την αύξηση της ευστάθειας του βρόχου.

Παρατηρήσεις:

- Προτείνεται να πειραματιστείτε αλλάζοντας τις τιμές των σταθερών που επηρεάζουν την ταχύτητα των τροχών (**SpeedLeftMotor**, **SpeedRightMotor**). Αν το Ρομπότ σας κινείται γρήγορα και συγκρούεται με τα εμπόδια μειώστε τις τιμές τους από τις αρχικές. Αντίθετα, αν πάει πολύ αργά ή δεν έχει καν την απαιτούμενη ροπή για να ξεκινήσει, αυξήστε τις τιμές (μέγιστο το 255). Πιθανόν το Ρομπότ να μην κινείται ευθεία όταν πρέπει να πάει μπροστά, τότε αν π.χ. στρίβει αριστερά, μπορείτε να αυξήσετε λίγο την ταχύτητα του αριστερού τροχού, αν πάλι στρίβει δεξιά, μπορείτε να αυξήσετε λίγο την ταχύτητα του δεξιού τροχού. Δώστε προσοχή ώστε οι αλλαγές να είναι μικρές (ανά 2 ή 5 μονάδες) για να μην έχετε αντίθετα αποτελέσματα από τα επιθυμητά.
- Τροποποιήστε τις τιμές των χρόνων που το Ρομπότ είναι σταματημένο (**StopTime**), στρίβει (**TurnTime**) και κινείται προς τα πίσω (**BackwardTime**) προκειμένου να κατανοήσετε καλύτερα πώς λειτουργεί και να πετύχετε την επιθυμητή για σας συμπεριφορά.
- Μεταβάλλετε τις τιμές των ελάχιστων αποστάσεων των εμποδίων από τους αισθητήρες (**LeftMinObstDist**, **RightMinObstDist**) προς τα πάνω αν το Ρομπότ δεν προλαβαίνει να φρενάρει και συγκρούεται με τα εμπόδια ή προς τα κάτω αν σταματά και στρίβει πολύ μακριά από αυτά.

2.5.5 Κώδικας Προγραμματισμού

Ο κώδικας παρατίθεται στο Παράρτημα.

Επιπλέον, μπορείτε να τον κατεβάσετε ως αρχείο πατώντας [εδώ](#).

Παράρτημα

Κώδικες προγραμματισμού των κατασκευών.

1. Ηχοδιακόπτης

```
/* *****  
1. Ηχοδιακόπτης  
***** */  
  
/** Δήλωση σταθερών, μεταβλητών, κτλ. **/  
const int PinLed = 2;      // Ο ακροδέκτης που συνδέθηκε η λυχνία (2).  
const int PinMic = A5;     // Ο ακροδέκτης που συνδέθηκε το μικρόφωνο (A5).  
const int ThresholdValue = 470; // Το όριο πάνω από το οποίο ανάβει(ως 1023).  
  
int SignalMic ;           // Η τιμή σήματος του μικροφώνου.  
boolean LedState = LOW ;  // Η κατάσταση της λυχνίας (HIGH - LOW).  
  
/** Ορισμός εισόδων, εξόδων, αρχικοποιήσεις κτλ., (τρέχει μόνο μια φορά). **/  
void setup() {  
  pinMode(PinMic, INPUT);    // Ορισμός ακροδέκτη μικροφώνου ως είσοδος.  
  pinMode(PinLed, OUTPUT);   // Ορισμός ακροδέκτη λυχνίας ως έξοδος.  
  Serial.begin(9600);        // Έναρξη σειριακής επικοινωνίας.  
}  
  
/** Ατέρμων Βρόχος (Εκτελείται επαναλαμβανόμενα.) **/  
void loop()  
{  
  SignalMic = analogRead(PinMic); // Διάβασε αναλογικά το σήμα του μικροφώνου από τον  
  ακροδέκτη A5.  
  digitalWrite(PinLed, LedState); // Ο ακροδέκτης λυχνίας πρέπει να πάρει την τιμή της  
  κατάστασης της λυχνίας.  
  Serial.print("LedState : ");    // Εκτύπωσε στην οθόνη αυτούς τους χαρακτήρες.  
  Serial.print(LedState);        // Εκτύπωσε στην οθόνη τη τρέχουσα κατάσταση της λυχνίας.
```

```
Serial.print(" --- SignalMic : "); // Εκτύπωσε στην οθόνη αυτούς τους χαρακτήρες.  
Serial.println(SignalMic);        // Εκτύπωσε στην οθόνη τη τρέχουσα τιμή του σήματος.  
Serial.println("====="); // Εκτύπωσε στην οθόνη  
αυτούς τους χαρακτήρες.  
  
/* Αν η τρέχουσα τιμή του σήματος είναι μεγαλύτερη του ορίου πρέπει να αλλάξει η κατάσταση  
της λυχνίας. */  
if (SignalMic > ThresholdValue){  
    LedState = !LedState;  
    delay(200); // Καθυστέρηση 200 millisecs για αποφυγή ψευδοαλλαγών.  
}  
  
delay(1);    // Καθυστέρηση 1 millisec για αύξηση της ευστάθειας του βρόχου.  
}
```


2. Φασαριόμετρο

```

/ *****
2. Φασαριόμετρο
***** /

/** Δήλωση σταθερών, μεταβλητών, κτλ. **/
const int SampleTime = 100; // Διάρκεια χρονικού διαστήματος Μέτρησης (milliseconds).
const int LightsOnTime = 100; // Χρόνος που παραμένουν αναμμένες οι λυχνίες (milliseconds).
const int Gain = 200; // Προσαύξηση τιμής στάθμης από Κορυφή σε Κορυφή (PeakToPeak).
const int MicPin = A5; // Ο ακροδέκτης σύνδεσης του μικροφώνου.
const int BluePin = 8; // Ο ακροδέκτης σύνδεσης της γαλάζιας λυχνίας LED.
const int GreenPin = 9; // Ο ακροδέκτης σύνδεσης της πράσινης λυχνίας LED.
const int YellowPin = 10; // Ο ακροδέκτης σύνδεσης της κίτρινης λυχνίας LED.
const int OrangePin = 11; // Ο ακροδέκτης σύνδεσης της πορτοκαλί λυχνίας LED.
const int RedPin = 12; // Ο ακροδέκτης σύνδεσης της κόκκινης λυχνίας LED.
int SignalMic; // Το σήμα από το μικρόφωνο το οποίο έχει τιμές από 0-1023.
int NoiseValue; // Στάθμη θορύβου η οποία έχει πέντε επίπεδα, με τιμές από 0-4.

/** Ορισμός εισόδων, εξόδων, αρχικοποιήσεις κτλ., (τρέχει μόνο μια φορά). **/
void setup() {
  Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας.
  pinMode(MicPin, INPUT); // Ορισμός ακροδέκτη ως είσοδος.
  pinMode(GreenPin, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
  pinMode(BluePin, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
  pinMode(YellowPin, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
  pinMode(OrangePin, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
  pinMode(RedPin, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.

  /** Άναψε τις λυχνίες, πάνω - κάτω 2 φορές ως ένδειξη εκκίνησης του προγράμματος. **/
  for(int k = 0; k < 2; k++){
    for(int i = 8; i < 13; i++){
      digitalWrite(i, HIGH);
      delay(20);
      digitalWrite(i, LOW);
      delay(20);
    }
  }
}

```

```

    }
    for(int i = 13; i > 7; i--){
        digitalWrite(i, HIGH);
        delay(20);
        digitalWrite(i, LOW);
        delay(20);
    }
}

/**/ Ατέρμων Βρόχος (Εκτελείται επαναλαμβανόμενα). ***/
void loop() {
    unsigned long startMillis = millis(); // Αρχή Χρονικού Διαστήματος Μέτρησης.
    unsigned int PeakToPeak = 0;         // Στάθμη σήματος μικροφώνου από Κορυφή σε Κορυφή.
    unsigned int SignalMax = 0;          // Η αρχική μέγιστη τιμή σήματος του μικροφώνου.
    unsigned int SignalMin = 1024;       // Η αρχική ελάχιστη τιμή σήματος του μικροφώνου.

    while (millis() - startMillis < SampleTime)
    {
        SignalMic = analogRead(MicPin); // Διάβασε τα δεδομένα από μικρόφωνο.

        if (SignalMic > SignalMax)
        {
            SignalMax = SignalMic;      // Αποθήκευση της Μέγιστης τιμής.
        }
        else if (SignalMic < SignalMin)
        {
            SignalMin = SignalMic;      // Αποθήκευση της Ελάχιστης τιμής.
        }
    }

    PeakToPeak = SignalMax - SignalMin ;    // Στάθμη από Κορυφή σε Κορυφή.
    PeakToPeak = PeakToPeak + Gain ;        // Προσαύξηση τιμής στο + Gain (+200).
    NoiseValue = map(PeakToPeak, 0, 1023, 0, 4); // Μετατροπή εύρους σε 5 επίπεδα.

    Serial.println("=====
    =====");
    Serial.print("SignalMin : "); Serial.print(SignalMin);
    Serial.print(" SignalMax : "); Serial.println(SignalMax);
    Serial.print("PeakToPeak : "); Serial.print(PeakToPeak);
    Serial.print(" NoiseValue : "); Serial.print(NoiseValue);

    switch (NoiseValue) {

```

```
case 0: // Άναψε μόνο τη γαλάζια λυχνία.
  Serial.println(" Άναψε μόνο τη γαλάζια λυχνία. ");
  digitalWrite(BluePin, HIGH);
  digitalWrite(GreenPin, LOW);
  digitalWrite(YellowPin, LOW);
  digitalWrite(OrangePin, LOW);
  digitalWrite(RedPin, LOW);
  delay(LightsOnTime );
  break;
case 1: // Άναψε ως την πράσινη λυχνία.
  Serial.println(" Άναψε ως την πράσινη λυχνία. ");
  digitalWrite(BluePin, HIGH);
  digitalWrite(GreenPin, HIGH);
  digitalWrite(YellowPin, LOW);
  digitalWrite(OrangePin, LOW);
  digitalWrite(RedPin, LOW);
  delay(LightsOnTime);
  break;
case 2: // Άναψε ως την κίτρινη λυχνία.
  Serial.println(" Άναψε ως την κίτρινη λυχνία. ");
  digitalWrite(BluePin, HIGH);
  digitalWrite(GreenPin, HIGH);
  digitalWrite(YellowPin, HIGH);
  digitalWrite(OrangePin, LOW);
  digitalWrite(RedPin, LOW);
  delay(LightsOnTime);
  break;
case 3: // Άναψε ως την πορτοκαλί λυχνία.
  Serial.println(" Άναψε ως την πορτοκαλί λυχνία. ");
  digitalWrite(BluePin, HIGH);
  digitalWrite(GreenPin, HIGH);
  digitalWrite(YellowPin, HIGH);
  digitalWrite(OrangePin, HIGH);
  digitalWrite(RedPin, LOW);
  delay(LightsOnTime);
  break;
case 4: // Άναψε ως την κόκκινη λυχνία.
  Serial.println(" Άναψε ως την κόκκινη λυχνία. ");
  digitalWrite(BluePin, HIGH);
  digitalWrite(GreenPin, HIGH);
  digitalWrite(YellowPin, HIGH);
  digitalWrite(OrangePin, HIGH);
  digitalWrite(RedPin, HIGH);
```

```
    delay(LightsOnTime);  
    break;  
  }  
}
```

3. Έλεγχος Θερμοκρασίας – Υγρασίας Θερμοκηπίου.

```

/*****
3. Έλεγχος Θερμοκρασίας – Υγρασίας Θερμοκηπίου.
*****/

/** Ενσωμάτωση βιβλιοθηκών. */
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
#include "DHT.h"

/** Δήλωση Σταθερών, Μεταβλητών, κτλ. */
const int MaxHumidity = 80 ; // Καθορισμός ορίου τιμής της Μεγίστης Υγρασίας.
const int MaxTemperature = 35 ; // Καθορισμός ορίου τιμής της Μεγίστης Θερμοκρασίας.
const int FanPin = 8 ; // Ο ακροδέκτης σύνδεσης του ανεμιστήρα.
const int DhtPin = 7 ; // Ο ακροδέκτης σύνδεσης του αισθητήρα.
#define DHTType DHT22 // Καθορισμός τύπου αισθητήρα σε DHT22(AM2302).
DHT dht(DhtPin, DHTType); // Καθορισμός τύπου και ακροδέκτη σύνδεσης του
αισθητήρα.
float Hum ; // Σχετική Υγρασία από αισθητήρα.
float Temp ; // Θερμοκρασία από αισθητήρα.
LiquidCrystal_I2C lcd(0x27,2,1,0,4,5,6,7,3,POSITIVE); // Καθορισμός διεύθυνσης οθόνης (0x27),
αρχικοποίηση οθόνης.

/** Ορισμός εισόδων, εξόδων, αρχικοποιήσεις κτλ. (τρέχει μόνο μια φορά). */
void setup() {
  pinMode(FanPin, OUTPUT); // Ορισμός ακροδέκτη ανεμιστήρα ως έξοδος.
  digitalWrite(FanPin, HIGH); // Εκκίνηση με απενεργοποιημένο ανεμιστήρα.
  Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας.
  dht.begin(); // Εκκίνηση και Αρχικοποίηση βιβλιοθήκης του αισθητήρα.
  lcd.begin(16,2); // Εκκίνηση και Αρχικοποίηση οθόνης.
  lcd.backlight(); // Εκκίνηση οπίσθιου φωτισμού οθόνης.
  lcd.setCursor(1, 0);
  lcd.print("-*- ELEGXOS -*-"); // Εκτύπωση αρχικού μηνύματος.
  lcd.setCursor(0, 1);
  lcd.print(" THERMOKHPIOY "); // Στην οθόνη χαρακτήρων lcd.
  delay(2000); // Περίμενε για 2000 millisecs.
  lcd.clear(); // Καθαρισμός χαρακτήρων αρχικού μηνύματος.
}

```

```
/** Ατέρμων Βρόχος (Εκτελείται επαναλαμβανόμενα). */  
void loop() {  
  Hum = dht.readHumidity();      // Μέτρηση σχετικής Υγρασίας.  
  Temp = dht.readTemperature();  // Μέτρηση Θερμοκρασίας.  
  
  /* Αν η μέτρηση από τον αισθητήρα δεν είναι επιτυχής, προσπάθησε ξανά. */  
  if (isnan(Hum) || isnan(Temp)) {  
    Serial.println("Αποτυχία μέτρησης ..."); // Εκτύπωση μηνύματος.  
    return;  
  }  
  
  /* Αν η Μέγιστη Υγρασία ή η Μέγιστη Θερμοκρασία είναι πάνω από το όριο,  
  ενεργοποίησε τον ανεμιστήρα. */  
  if (Hum > MaxHumidity || Temp > MaxTemperature) {  
    digitalWrite(FanPin, LOW); // Οπλίζει με LOW.  
  }  
  /* Αλλιώς απενεργοποίησε τον ανεμιστήρα. */  
  else {  
    digitalWrite(FanPin, HIGH); // Αφοπλίζει με HIGH.  
  }  
  
  /* Εκτύπωση Υγρασίας και Θερμοκρασίας στην οθόνη του Η/Υ. */  
  Serial.print("Humidity: ");  
  Serial.print(Hum);  
  Serial.print(" %\t");  
  Serial.print("Temperature: ");  
  Serial.print(Temp);  
  Serial.println(" *C ");  
  
  /* Εκτύπωση Υγρασίας και Θερμοκρασίας στην οθόνη χαρακτήρων lcd. */  
  lcd.setCursor(0,0);  
  lcd.print("Humidity:");  
  lcd.print(Hum);  
  lcd.setCursor(14,0);  
  lcd.print(" %");  
  lcd.setCursor(0,1);  
  lcd.print("Temperat:");  
  lcd.print(Temp);  
  lcd.setCursor(14,1);  
  lcd.print(" *C");  
  
  delay(1000); // Περίμενε 1000 millisecs πριν ξαναμετρήσεις.  
}
```


4. Αυτόματο Πότισμα Γλάστρας.

```

/*****
4. Αυτόματο Πότισμα Γλάστρας.
*****/

/*** Δήλωση σταθερών, μεταβλητών, κτλ. ***/
const long WateringTime = 3000;    // Ο χρόνος ποτίσματος (millisecs).
const long CheckTime = 5000 ;     // Ο χρόνος αναμονής πριν μετρήσει ξανά σε κάθε κύκλο
προγράμματος.
const long AdditionalTime = 60000; // Ο επιπλέον χρόνος αναμονής όταν είναι "Ποτισμένο"
το χώμα.
const int DryLimit = 40;    // Τιμή ορίου της Σχετικής Υγρασίας κάτω από το οποίο θεωρείται
"Ξηρό" το χώμα.
const int WateredLimit = 70;    // Τιμή ορίου της Σχετικής Υγρασίας πάνω από το οποίο
θεωρείται "Ποτισμένο" το χώμα.
const int SensorPin = A0;    // Ο ακροδέκτης σύνδεσης του αισθητήρα.
const int PumpPin = 8;    // Ο ακροδέκτης σύνδεσης της αντλίας.
const int GreenPin = 7;    // Ο ακροδέκτης σύνδεσης της πράσινης λυχνίας LED.
const int OrangePin = 6;    // Ο ακροδέκτης σύνδεσης της πορτοκαλί λυχνίας LED.
const int RedPin = 5;    // Ο ακροδέκτης σύνδεσης της κόκκινης λυχνίας LED.
int SensorSignal;    // Τιμή σήματος από τον αισθητήρα.
int MoistureValue;    // Τιμή Σχετικής Υγρασίας επί τοις εκατό (%).

/*** Ορισμός εισόδων, εξόδων, αρχικοποιήσεις κτλ.,(τρέχει μόνο μια φορά). ***/
void setup() {
  pinMode(SensorPin,INPUT);    // Ορισμός ακροδέκτη ως είσοδος.
  pinMode(PumpPin, OUTPUT);    // Ορισμός ακροδέκτη ως έξοδος.
  digitalWrite(PumpPin, HIGH);    // Αρχικά αποπλισμένο.
  pinMode(GreenPin, OUTPUT);    // Ορισμός ακροδέκτη ως έξοδος.
  pinMode(OrangePin, OUTPUT);    // Ορισμός ακροδέκτη ως έξοδος.
  pinMode(RedPin, OUTPUT);    // Ορισμός ακροδέκτη ως έξοδος.
  Serial.begin(9600);    // Έναρξη σειριακής επικοινωνίας.
  Serial.println("Εκκίνηση...");    // Εκτόπωση μηνύματος στην οθόνη του Η/Υ.

  /* Άναψε τις λυχνίες, πάνω - κάτω 2 φορές ως ένδειξη εκκίνησης του προγράμματος. */
  for(int k = 0; k < 2; k++){
    for(int i = 5; i < 8; i++){
      digitalWrite(i, HIGH);

```

```

    delay(20);
    digitalWrite(i, LOW);
    delay(20);
  }
  for(int i = 8; i > 4; i--){
    digitalWrite(i, HIGH);
    delay(20);
    digitalWrite(i, LOW);
    delay(20);
  }
}

/** Ατέρμων Βρόχος (εκτελείται επαναλαμβανόμενα). ***/
void loop() {

  /* Διάβασε το σήμα από τον αισθητήρα και μετάτρεψε το σε Σχετική Υγρασία επί τοις εκατό
  (%). */
  SensorSignal = analogRead(SensorPin);
  MoistureValue = map(SensorSignal, 1023, 0, 0, 100);

  /* Εκτόπωση σήματος του αισθητήρα και της Σχετικής Υγρασίας (%) στην οθόνη του Η/Υ. */
  Serial.print("SensorSignal : ");
  Serial.print(SensorSignal);
  Serial.print("  Moisture : ");
  Serial.print(MoistureValue);
  Serial.println("%");

  /*  Αν η Σχετική Υγρασία(%) είναι μικρότερη ή ίση από το όριο "Ξηρό"
  *  (δηλαδή αν MoistureValue <= 40), ενεργοποίησε την αντλία και άναψε
  *  την κόκκινη λυχνία. *****/
  if ( MoistureValue <= DryLimit )
  {
    digitalWrite(PumpPin, LOW);  // Οπλίζει με LOW.
    Serial.println("Ξηρό χώμα. Πότιζε... Άναψε την κόκκινη λυχνία.");
    digitalWrite(GreenPin, LOW);  // Σβήσε την πράσινη λυχνία.
    digitalWrite(OrangePin, LOW);  // Σβήσε την πορτοκαλί λυχνία.
    digitalWrite(RedPin, HIGH);  // Άναψε την κόκκινη λυχνία.
    delay(WateringTime);  // Περίμενε για χρόνο Watering_Time.
    digitalWrite(PumpPin, HIGH);  // Αφοπλίζει με HIGH.
    Serial.println("Σταμάτα να ποτίζεις. Περίμενε λίγο πριν μετρήσεις... ");
  }
}

```

```

}

/* Αλλιώς αν η Σχετική Υγρασία (%) είναι μεγαλύτερη από το όριο "Ξηρό" ΚΑΙ ταυτόχρονα
 * μικρότερη από το όριο "Ποτισμένο" (αν 40 < MoistureValue < 70), ενεργοποίησε την
 * αντλία και άναψε την πορτοκαλί λυχνία. *****/
else if ((MoistureValue > DryLimit) && (MoistureValue < WateredLimit))
{
    digitalWrite(PumpPin, LOW); // Οπλίζει με LOW.
    Serial.println("Χρειάζεται κι άλλο. Πότιζε... Άναψε την πορτοκαλί λυχνία.");
    digitalWrite(GreenPin, LOW); // Σβήσε την πράσινη λυχνία.
    digitalWrite(OrangePin, HIGH); // Άναψε την πορτοκαλί λυχνία.
    digitalWrite(RedPin, LOW); // Σβήσε την κόκκινη λυχνία.
    delay(WateringTime); // Περίμενε για χρόνο Watering_Time.
    digitalWrite(PumpPin, HIGH); // Αφοπλίζει με HIGH.
    Serial.println("Σταμάτα να ποτίζεις. Περίμενε λίγο πριν μετρήσεις.");
}

/* Αλλιώς (δηλαδή αν MoistureValue >= 70) απενεργοποίησε την αντλία
 * και άναψε την πράσινη λυχνία. *****/
else
{
    digitalWrite(PumpPin, HIGH); // Αφοπλίζει με HIGH.
    Serial.println(" ΠΟΤΙΣΜΕΝΟ !!! Περίμενε για επιπλέον χρόνο ... Άναψε την πράσινη
λυχνία.");
    digitalWrite(GreenPin, HIGH); // Άναψε την πράσινη λυχνία.
    digitalWrite(OrangePin, LOW); // Σβήσε την πορτοκαλί λυχνία.
    digitalWrite(RedPin, LOW); // Σβήσε την κόκκινη λυχνία.
    delay(AdditionalTime); // Περίμενε για επιπλέον χρόνο (AdditionalTime).
}

delay(CheckTime); // Περίμενε για χρόνο Check_Time.
Serial.println("-----");
Serial.println(" Μέτρα την υγρασία ξανά... ");
}

```

5. Ρομπότ Αποφυγής Εμποδίων.

```

/*****
5. Ρομπότ Αποφυγής Εμποδίων.
*****/

/** Δήλωση σταθερών, μεταβλητών, κτλ. */
const int SpeedLeftMotor = 120; // Τιμή (PWM 0-255) που καθορίζει τη ταχύτητα του
αριστερού κινητήρα.
const int SpeedRightMotor = 120; // Τιμή (PWM 0-255) που καθορίζει τη ταχύτητα του δεξιού
κινητήρα.
const int StopTime = 700; // Ο χρόνος αναμονής όταν είναι σταματημένο (milliseconds).
const int TurnTime = 500; // Ο χρόνος που στρίβει (milliseconds).
const int BackwardTime = 800; // Ο χρόνος που κινείται προς τα πίσω (milliseconds).
const int LeftMinObstDist = 35; // Ελάχιστη απόσταση εμποδίου από αριστερό αισθητήρα
(cm).
const int RightMinObstDist = 35; // Ελάχιστη απόσταση εμποδίου από δεξιό αισθητήρα (cm).

const int TrigPinL = 3; // Ο ακροδέκτης σύνδεσης Trigger του αριστερού αισθητήρα.
const int EchoPinL = 2; // Ο ακροδέκτης σύνδεσης Echo του αριστερού αισθητήρα.
const int TrigPinR = 7; // Ο ακροδέκτης σύνδεσης Trigger του δεξιού αισθητήρα.
const int EchoPinR = 6; // Ο ακροδέκτης σύνδεσης Echo του δεξιού αισθητήρα.

const int IN1 = 13; // Ο ακροδέκτης σύνδεσης της Εισόδου 1 της Η-γέφυρας.
const int IN2 = 12; // Ο ακροδέκτης σύνδεσης της Εισόδου 2 της Η-γέφυρας.
const int SpeedPinL = 11; // Ο ακροδέκτης ελέγχου ταχύτητας αριστερού κινητήρα (ENA).
const int IN3 = 9; // Ο ακροδέκτης σύνδεσης της Εισόδου 3 της Η-γέφυρας.
const int IN4 = 8; // Ο ακροδέκτης σύνδεσης της Εισόδου 4 της Η-γέφυρας.
const int SpeedPinR = 10; // Ο ακροδέκτης ελέγχου ταχύτητας του δεξιού κινητήρα (ENB).

long DurationL ; // Ο χρόνος που ταξίδεψε ο παλμός από τον αριστερό αισθητήρα στο
εμπόδιο και πίσω (microsecs).
int DistanceL ; // Απόσταση εμποδίου από τον αριστερό αισθητήρα (cm).
long DurationR ; // Ο χρόνος που ταξίδεψε ο παλμός από τον δεξιό αισθητήρα στο εμπόδιο
και πίσω (microsecs).
int DistanceR ; // Απόσταση εμποδίου από τον δεξιό αισθητήρα (cm).

/** Ορισμός εισόδων, εξόδων, αρχικοποιήσεις κτλ.,(τρέχει μόνο μια φορά). */
void setup() {

```

```

pinMode(TrigPinL, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(EchoPinL, INPUT); // Ορισμός ακροδέκτη ως είσοδος.
pinMode(TrigPinR, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(EchoPinR, INPUT); // Ορισμός ακροδέκτη ως είσοδος.
pinMode(IN1, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(IN2, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(IN3, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(IN4, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(SpeedPinL, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.
pinMode(SpeedPinR, OUTPUT); // Ορισμός ακροδέκτη ως έξοδος.

Serial.begin(9600); // Έναρξη σειριακής επικοινωνίας.
}

/** Ατέρμων Βρόχος (Εκτελείται επαναλαμβανόμενα). **/
void loop() {

  /* Εκτέλεση Υπορουτινών που υπολογίζουν την απόσταση των εμποδίων
   * από τον αριστερό και τον δεξιό αισθητήρα. *****/
  Read_DistanceL();
  Read_DistanceR();

  /* Εκτύπωση τιμών αποστάσεων των εμποδίων στην οθόνη του Η/Υ */
  Serial.println("=====");
  Serial.print(DistanceL); Serial.print("cm , ");
  Serial.print(DistanceR); Serial.print("cm");
  Serial.println();

  /* Αν η απόσταση του εμποδίου από τον αριστερό ή τον δεξιό αισθητήρα
   * είναι μικρότερη από το ελάχιστο όριο. *****/
  if ((DistanceL < LeftMinObstDist) || (DistanceR < RightMinObstDist)) {

    /* Εκτύπωσε το μήνυμα κινδύνου και τις αποστάσεις αριστερά - δεξιά στην οθόνη του Η/Υ.
    */
    Serial.println ("Κίνδυνος, εμπόδιο μπροστά! Οι αποστάσεις είναι :");
    Serial.print (" Αριστερά: ");
    Serial.print ( DistanceL);
    Serial.print ( " cm ");
    Serial.print (" --- Δεξιά: ");
    Serial.print (DistanceR);
    Serial.println ( " cm ");

    /* Εκτέλεση Υπορουτίνας, πήγαινε προς τα πίσω για λίγο. */

```

```
Go_Backward();

/* Άν η απόσταση του εμποδίου αριστερά είναι μεγαλύτερη
 * από την απόσταση δεξιά τότε Στρίψε αριστερά. *****/
if (DistanceL > DistanceR) {
    Turn_Left();
}
/* Αλλιώς Στρίψε δεξιά. */
else {
    Turn_Right();
}
}

/* Αλλιώς (αν δεν υπάρχει εμπόδιο κοντά) εκτύπωσε το παρακάτω μήνυμα οθόνη του Η/Υ. */
else
{
    Serial.println (" Δεν υπάρχει εμπόδιο. Πήγαινε προς τα εμπρός.");
    /* Εκτέλεση Υπορουτίνας, πήγαινε προς τα εμπρός. */
    Go_Forward();
}

delay(1); // Καθυστέρηση 1ms για την αύξηση της ευστάθειας του βρόχου.
}

/** Υπορουτίνα η οποία υπολογίζει την απόσταση του εμποδίου από τον αριστερό αισθητήρα.
**/
void Read_DistanceL(){
    /* Εκπομπή παλμού ηχητικού κύματος από τον αριστερό αισθητήρα. */
    digitalWrite(TrigPinL, LOW);
    delayMicroseconds(2);
    digitalWrite(TrigPinL, HIGH);
    delayMicroseconds(10);
    digitalWrite(TrigPinL, LOW);

    /* Διάβασε το σήμα από τον αισθητήρα το οποίο είναι η διάρκεια (Microsecs) ενός HIGH
    παλμού. */
    DurationL = pulseIn(EchoPinL, HIGH);
    /* Μετατροπή της χρονικής διάρκειας σε απόσταση (cm). */
    DistanceL = (DurationL/2) / 29.1; // Διαίρεσε με 29.1 για cm
    return DistanceL;
}
```



```
/** Υπορουτίνα η οποία υπολογίζει την απόσταση του εμποδίου από τον δεξιό αισθητήρα.
***/
void Read_DistanceR(){
  /* Εκπομπή παλμού ηχητικού κύματος από τον δεξιό αισθητήρα. */
  digitalWrite(TrigPinR, LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPinR, HIGH);
  delayMicroseconds(10);
  digitalWrite(TrigPinR, LOW);
  /* Διάβασε το σήμα από τον αισθητήρα το οποίο είναι η διάρκεια (Microsecs) ενός HIGH
  παλμού. */
  DurationR = pulseIn(EchoPinR, HIGH);
  /* Μετατροπή της χρονικής διάρκειας σε απόσταση (cm). */
  DistanceR = (DurationR/2) / 29.1; // Διάρθρωσε με 29.1 για cm
  return DistanceR;
}

/** Υπορουτίνα η οποία στρίβει το Ρομπότ προς τα Δεξιά. ***/
void Turn_Right(){
  // Σταμάτα τον αριστερό τροχό.
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
  // Σταμάτα τον δεξιό τροχό.
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, HIGH);
  Serial.println (" Stop !");
  delay(StopTime);

  // Στρίψε το Ρομπότ Δεξιά.
  Serial.println (" Στρίψε Δεξιά! ");
  analogWrite(SpeedPinL, SpeedLeftMotor); // Ορισμός ταχύτητας μέσω PWM.
  analogWrite(SpeedPinR, SpeedRightMotor); // Ορισμός ταχύτητας μέσω PWM.
  // Στρίψε τον αριστερό τροχό σε αντίθετη ωρολογιακή κατεύθυνση.
  digitalWrite(IN1, LOW);
  digitalWrite(IN2, HIGH);
  // Στρίψε τον δεξιό τροχό σε ωρολογιακή κατεύθυνση.
  digitalWrite(IN3, HIGH);
  digitalWrite(IN4, LOW);
  delay(TurnTime); // Περίμενε για TurnTime (milliseconds).
  // Σταμάτα τον αριστερό τροχό.
  digitalWrite(IN1, HIGH);
  digitalWrite(IN2, HIGH);
```

```
// Σταμάτα τον δεξί τροχό
digitalWrite(IN3, HIGH);
digitalWrite(IN4, HIGH);
Serial.println (" Stop !");
delay(StopTime); // Περίμενε για StopTime (millisecs).
}

/** Υπορουτίνα η οποία στρίβει το Ρομπότ προς τα Αριστερά. */
void Turn_Left(){
    // Σταμάτα τον αριστερό τροχό.
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    // Σταμάτα τον δεξί τροχό
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
    Serial.println (" Stop !");
    delay(StopTime); // Περίμενε για StopTime (millisecs).

    // Στρίψε το Ρομπότ Αριστερά.
    Serial.println (" Στρίψε Αριστερά! ");
    analogWrite(SpeedPinL, SpeedLeftMotor); // Ορισμός ταχύτητας μέσω PWM.
    analogWrite(SpeedPinR, SpeedRightMotor); // Ορισμός ταχύτητας μέσω PWM.
    // Στρίψε τον αριστερό τροχό σε ωρολογιακή κατεύθυνση.
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    // Στρίψε τον δεξί τροχό σε αντίθετη ωρολογιακή κατεύθυνση.
    digitalWrite(IN3, LOW);
    digitalWrite(IN4, HIGH);
    delay(TurnTime); // Περίμενε για TurnTime (millisecs).
    // Σταμάτα τον αριστερό τροχό
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, HIGH);
    // Σταμάτα τον δεξί τροχό
    digitalWrite(IN3, HIGH);
    digitalWrite(IN4, HIGH);
    Serial.println (" Stop !");
    delay(StopTime); // Περίμενε για StopTime (millisecs).
}

/** Υπορουτίνα η οποία οδηγεί το Ρομπότ προς τα εμπρός. */
void Go_Forward(){
```

```
Serial.println (" Πήγαινε Μπροστά! ");
analogWrite(SpeedPinL, SpeedLeftMotor);
analogWrite(SpeedPinR, SpeedRightMotor);
// Στρίψε τον αριστερό τροχό σε ωρολογιακή κατεύθυνση.
digitalWrite(IN1, LOW );
digitalWrite(IN2, HIGH);
// Στρίψε τον δεξί τροχό σε ωρολογιακή κατεύθυνση.
digitalWrite(IN3, LOW);
digitalWrite(IN4, HIGH);
}

/** Υπορουτίνα η οποία οδηγεί το Ρομπότ προς τα πίσω. **/
void Go_Backward(){
Serial.println (" Πήγαινε Πίσω! ");
// Σταμάτα τον αριστερό τροχό.
digitalWrite(IN1, HIGH);
digitalWrite(IN2, HIGH);
// Σταμάτα τον δεξί τροχό
digitalWrite(IN3, HIGH);
digitalWrite(IN4, HIGH);
delay(StopTime); // Περίμενε για StopTime (millisecs).
analogWrite(SpeedPinL, SpeedLeftMotor);
analogWrite(SpeedPinR, SpeedRightMotor);
// Στρίψε τον αριστερό τροχό σε αντίθετη ωρολογιακή κατεύθυνση.
digitalWrite(IN1, HIGH );
digitalWrite(IN2, LOW);
// Στρίψε τον δεξί τροχό σε αντίθετη ωρολογιακή κατεύθυνση.
digitalWrite(IN3, HIGH);
digitalWrite(IN4, LOW);
delay(BackwardTime);
}
```

Βιβλιογραφία και Δικτυογραφία.

1. <https://store.arduino.cc/arduino-uno-rev3>
2. <https://www.arduino.cc/en/Main/Software>
3. <https://www.arduino.cc/en/Main/Education>
4. <https://en.wikipedia.org/wiki/Arduino>
5. Ζούλης Ν., Καφφετζάκης Π., Σούλης Γ., Συστήματα Αυτοματισμών, Τ.Ε.Ε. Β' Τάξη 1ου κύκλου, Α' τόμος, 2000, Εκδ. Λιβάνη
6. Μπιτζιώνη, Β. (1997). Ηλεκτρικές Μετρήσεις, Εκδ. Τζιόλα
7. Τοπαλής Φ., Χαραλαμπακίης Ν., Χριστοδούλου Θ., Ηλεκτρολογικό Εργαστήριο, Τ.Ε.Ε. Α' Τάξη 1ου κύκλου, 2007, Ο.Ε.Δ.Β.
8. Δημόπουλος Φ., Παγιάτης Χ., Πάγκαλος Στ., Στοιχεία Ηλεκτρολογίας, Τ.Ε.Ε. Α' Τάξη 1ου κύκλου, 2007, Ο.Ε.Δ.Β.
9. Πουλάκης, Ε. (2015). Προγραμματίζοντας με τον μικροελεγκτή Arduino. Ε. Πουλάκης : Ηράκλειο.
10. Σφακιωτάκης Μ. (2015). Σημειώσεις μαθήματος "Ενσωματωμένα Συστήματα Ελέγχου", Δ.Π.Μ.Σ. «Προηγμένα Συστήματα Παραγωγής, Αυτοματισμού και Ρομποτικής», Τ.Ε.Ι. Κρήτης, Ηράκλειο
11. Κατασκευή «Φασαριόμετρου» Ν. Φανουράκης, Ε. Φανουράκη, Σάββατο 8 Απριλίου 2017, 10ο Πανελλήνιο Συνέδριο Διδακτικής των Φυσικών Επιστημών και Νέων Τεχνολογιών στην Εκπαίδευση. «ΓΕΦΥΡΩΝΟΝΤΑΣ ΤΟ ΧΑΣΜΑ ΜΕΤΑΞΥ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ, ΚΟΙΝΩΝΙΑΣ ΚΑΙ ΕΚΠΑΙΔΕΥΤΙΚΗΣ ΠΡΑΞΗΣ» 7-9 Απριλίου 2017, Πανεπιστήμιο Κρήτης, Παιδαγωγικό Τμήμα Δημοτικής Εκπαίδευσης, Εργαστήριο Διδακτικής Θετικών Επιστημών, Πανεπιστημιούπολη Ρεθύμνου σε συνεργασία με την Ένωση για την Εκπαίδευση στις Φυσικές Επιστήμες και την Τεχνολογία (ΕΝΕΦΕΤ). Πρακτικά 10ου Πανελληνίου Συνεδρίου 2018, Τόμος 10, ISBN: 978-960-86978-3-6

[11]

Εκπαιδευτική Ρομποτική με τον μικροελεγκτή Arduino.

Ένας πρακτικός οδηγός κατασκευών για μαθητές και εκπαιδευτικούς.

Νικολάου Α. Φανουράκη

ISBN : 978-618-00-1548-5

Ασήμι Ηρακλείου Κρήτης

2019