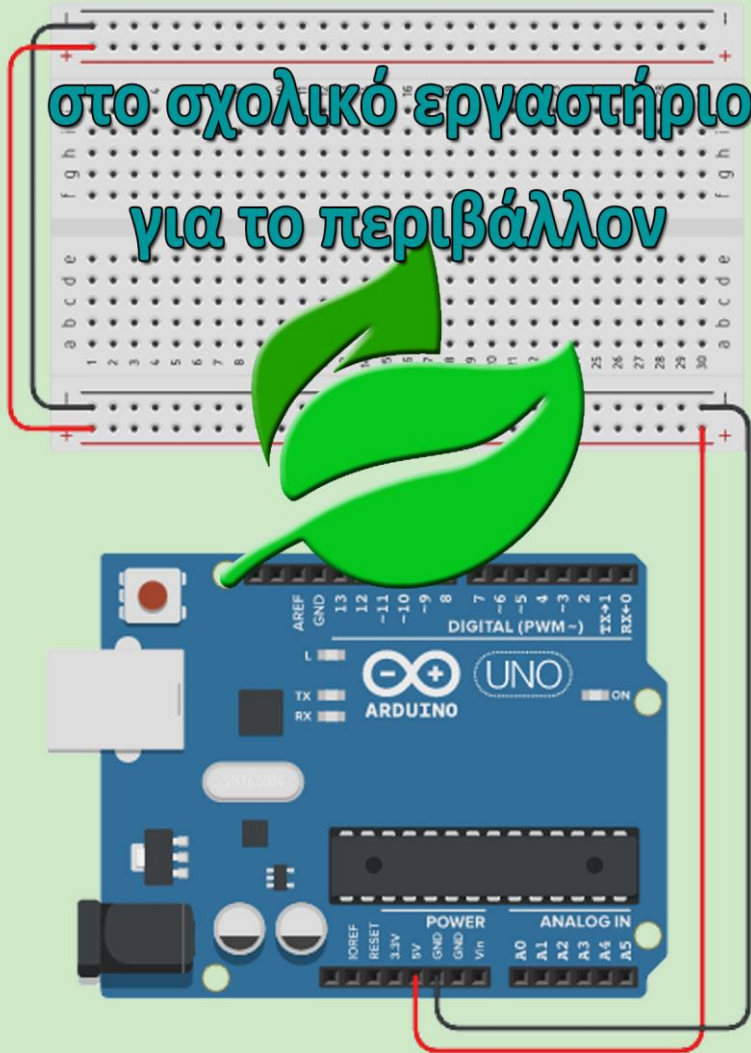


Γεώργιος Γίδας - Ευάγγελος Μαντζάνας

ΕΦΑΡΜΟΓΕΣ ARDUINO

στο σχολικό εργαστήριο
για το περιβάλλον



1ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ



Γ. Γίδας

Καθηγητής Πληροφορικής
1^ο ΕΠΑΛ Τρικάλων

Ε. Μαντζάνας

Καθηγητής Πληροφορικής
1^ο ΕΠΑΛ Τρικάλων

ΕΦΑΡΜΟΓΕΣ ARDUINO

στο σχολικό εργαστήριο
για το περιβάλλον

Τρίκαλα, 2021

ΣΥΓΓΡΑΦΙΚΗ ΟΜΑΔΑ:

Γεώργιος Γίδας, καθηγητής πληροφορικής

Ευάγγελος Μαντζάνας, καθηγητής πληροφορικής

ΕΠΙΜΕΛΕΙΑ:

Νικόλαος Ντόγκας, Διευθυντής 1^{ου} ΕΠΑΛ Τρικάλων

1^η ΕΚΔΟΣΗ:

1^ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ, Απρίλιος 2021

ISBN: 978-618-85383-0-6

Copyright[©]:

1^ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ, ΓΕΩΡΓΙΟΣ ΓΙΔΑΣ, ΕΥΑΓΓΕΛΟΣ ΜΑΝΤΖΑΝΑΣ



**Αναφορά Δημιουργού-Μη Εμπορική Χρήση 4.0 Διεθνές
(CC BY-NC 4.0)**

Μπορείτε να:

Μοιραστείτε — αντιγράψετε και αναδιανέμετε το υλικό με κάθε μέσο και τρόπο

Προσαρμόστε — αναμείξτε, τροποποιήστε και δημιουργήστε πάνω στο υλικό

Υπό τους ακόλουθους όρους:

Αναφορά Δημιουργού — Θα πρέπει να καταχωρίσετε αναφορά στο δημιουργό, με σύνδεσμο της άδειας, και με αναφορά αν έχουν γίνει αλλαγές. Μπορείτε να το κάνετε αυτό με οποιονδήποτε εύλογο τρόπο, αλλά όχι με τρόπο που να υπονοεί ότι ο δημιουργός αποδέχεται το έργο σας ή τη χρήση που εσείς κάνετε.

Μη Εμπορική Χρήση — Δε μπορείτε να χρησιμοποιήσετε το υλικό για εμπορικούς σκοπούς.

Περιεχόμενα

Πρόλογος.....	5
Εισαγωγή.....	9
Απαραίτητα Εξαρτήματα – Υλικά	10
1. Βασικά στοιχεία Arduino.....	11
1.1 Προγραμματισμός του Arduino	12
1.2 Θύρες εισόδου και αισθητήρες.....	13
1.3 Θύρες εξόδου και ενεργοποιητές	15
1.4 Κινητήρες.....	16
2. Αυτόματος Φωτισμός.....	19
2.1 Μέτρηση απόστασης.....	19
2.2 Ελεγχόμενος Φωτισμός.....	23
3. Ηλιακό πάνελ.....	27
4. Συνδυασμός ηλιακού πάνελ με αυτόματο φωτισμό.....	33
5. Έξυπνο Σκίαστρο.....	39
6. Πότισμα	47
6.1 Έξυπνο πότισμα	47
6.2 Έξυπνο πότισμα τη νύχτα	52
7. Διαχωριστής Μεταλλικών Αντικειμένων	57
7.1 Έλεγχος κινητήρα DC.....	58
7.2 Έλεγχος βηματικού κινητήρα	63
7.3 Επαγωγικός αισθητήρας	68
7.4 Μικροδιακόπτες και όρια κίνησης.....	73
7.5 Ολοκλήρωση του διαχωριστή μετάλλων.....	79

8. Παρακολούθηση ποιότητας αέρα 87
Βιβλιογραφία 102
Πηγές 102



Πρόλογος

Σε ένα διαρκώς μεταβαλλόμενο οικονομικό και τεχνολογικό περιβάλλον το 1^ο ΕΠΑΛ Τρικάλων επιχειρεί να κάνει κινήσεις εξωστρέφειας, να δημιουργήσει και να προσφέρει προστιθέμενη αξία στην επαγγελματική εκπαίδευση· μέσα από καινοτόμες ενέργειες στην εκπαιδευτική διαδικασία, με νέες μεθόδους και τεχνικές διδασκαλίας και με τη μεταφορά τεχνογνωσίας σε κάθε έναν από τους τομείς και τις ειδικότητες που λειτουργούν σε αυτό.

Το βιβλίο που κρατάτε στα χέρια σας ή ξεφυλλίζετε ηλεκτρονικά στον υπολογιστή σας, αποτελεί την πρώτη εκδοτική προσπάθεια στο 1^ο ΕΠΑΛ Τρικάλων. Προσφέρει τη δυνατότητα δημιουργίας ενός αμφίδρομου διαύλου επικοινωνίας με το εκπαιδευτικό προσωπικό, που θα προωθή την ανταλλαγή καλών πρακτικών, τη μεταφορά καινοτόμων πρακτικών διδασκαλίας και τον εμπλουτισμό της εκπαιδευτικής διαδικασίας.

Πρόκειται για έργο δύο εξάιρετων συναδέλφων καθηγητών, του κ. Γεώργιου Γίδα και του κ. Ευάγγελου Μαντζάνα, ειδικότητας Πληροφορικής, οι οποίοι τα τελευταία χρόνια εργάζονται στο σχολείο μας και οργάνωσαν το νέο εργαστήριο Ρομποτικής και Καινοτομίας, που παραλάβαμε αρχικά από το Υπουργείο Παιδείας και Θρησκευμάτων και το εμπλουτίσαμε στη συνέχεια με εξοπλισμό τον οποίο μας χορήγησε ο Δήμος Τρικκαίων και η Περιφέρεια Θεσσαλίας ή αποκτήσαμε από αγορές και δωρεές μέσω της συμμετοχής μας σε διάφορα εκπαιδευτικά προγράμματα.

Το βιβλίο αυτό δημιουργήθηκε από τις σημειώσεις, τις ασκήσεις, τις εργασίες και τα φύλλα έργου που εκπονήθηκαν στο 1^ο ΕΠΑΛ Τρικάλων για τις ανάγκες των μαθημάτων «Ερευνητική Εργασία», «Ζώνη Δημιουργικών Δραστηριοτήτων», «Πληροφορική», καθώς και των μαθημάτων ειδικότητας του Τομέα Πληροφορικής.

Η έμπνευση για τα έργα που αναφέρονται στο παρόν βιβλίο οφείλεται στους καθηγητές-συγγραφείς του βιβλίου και στους μαθητές του σχολείου, στο πλαίσιο της συμμετοχής των στον 2^ο Πανελλήνιο Διαγωνισμό Ανοιχτών Τεχνολογιών στην Εκπαίδευση για την σχολική περίοδο 2019-2020 της ΕΛΛΑΚ. Το έργο με το οποίο συμμετείχε το 1^ο ΕΠΑΛ Τρικάλων, «Το Αειφόρο Σχολείο», ξεχώρισε για την πρωτότυπη ιδέα του, την υλοποίησή του, καθώς και για τις αναλυτικές οδηγίες κατασκευής και χρήσης του, και έλαβε διάκριση μεταξύ των ΕΠΑΛ. Επίσης, αρκετές από τις επιμέρους εργασίες του «Αειφόρου Σχολείου» χρησιμοποιήθηκαν στο τεχνικό μέρος του προγράμματος «Μαθητική Εικονική Επιχείρηση 2020» του Junior Achievement Greece, στο οποίο το 1^ο ΕΠΑΛ Τρικάλων συμμετείχε δημιουργώντας το LifeDrop, ένα καινοτόμο προϊόν, το οποίο σχεδιάστηκε και κατασκευάστηκε στο Εργαστήριο Καινοτομίας και Ρομποτικής του σχολείου.

Αποτελεί έναν πρακτικό και κατανοητό οδηγό, χωρίς πολλή θεωρία ή δυσνόητες έννοιες, και ταυτόχρονα ένα χρηστικό εργαλείο για κάθε αρχάριο χρήστη.

Η δομή του είναι απλή και τα κεφάλαια αποτελούν ξεχωριστά φύλλα έργου, δίνοντας έτσι τη δυνατότητα, οι εργασίες που αναφέρονται να εφαρμόζονται με εύκολο τρόπο, τόσο από τους διδάσκοντες καθηγητές όσο και από τους μαθητές, μέσα σε ένα εργαστήριο.

Επιλέξαμε, το παρόν βιβλίο να κυκλοφορήσει παράλληλα και σε ηλεκτρονική μορφή, έτσι ώστε να είναι προσιτό σε όλη την εκπαιδευτική κοινότητα και να αποτελέσει εργαλείο διδασκαλίας σε μαθήματα που ενσωματώνουν τεχνολογίες Arduino.

Στο 1^ο ΕΠΑΛ Τρικάλων θέλουμε να είμαστε πάντα πρωτοποριακοί και να ανταποκρινόμαστε στις προκλήσεις των καιρών, για να διαμορφώσουμε ένα σχολείο που θα είναι καινοτόμο στην επαγγελματική εκπαίδευση και θα γίνει γνωστό όχι μόνο στην τοπική κοινωνία αλλά και σε εθνικό και ευρωπαϊκό επίπεδο.

Επενδύουμε στην ανάπτυξη του εκπαιδευτικού προσωπικού με την αξιοποίηση διαφόρων δραστηριοτήτων κατάρτισης και τη συμμετοχή του σε εκπαιδευτικούς διαγωνισμούς καθώς επίσης και σε κινητικότητες κατάρτισης στο εξωτερικό μέσω του προγράμματος Erasmus+.

Θέτουμε υψηλούς στόχους στην κατάρτιση των μαθητών και επιδιώκουμε την ανάπτυξή τους μέσα από ευρωπαϊκά και άλλα προγράμματα, τόσο σε προσωπικό όσο και σε επαγγελματικό επίπεδο, προσφέροντάς τους τις ευκαιρίες εκείνες που θα τους βοηθήσουν στην εύρεση εργασίας όταν αποφοιτήσουν από το σχολείο.

Στο 1^ο ΕΠΑΛ Τρικάλων, διεύθυνση και εκπαιδευτικό προσωπικό, θα συνεχίσουμε να καινοτομούμε και θα διευρύνουμε την εκδοτική μας προσπάθεια με πρωτότυπες εργασίες και άλλων καθηγητών.

ΝΙΚΟΛΑΟΣ Κ. ΝΤΟΓΚΑΣ

Διευθυντής 1^{ου} ΕΠΑΛ ΤΡΙΚΑΛΩΝ

Εισαγωγή

Χρονοδιακόπτες, θερμοστάτες, τηλεχειριστήρια, ρομπότ, συστήματα απομακρυσμένου ελέγχου και επικοινωνίας, διάφοροι αυτοματισμοί, συναγερμοί, έξυπνες συσκευές και πολλά άλλα συστήματα αυτομάτου ελέγχου είναι μόνο μερικά παραδείγματα από ένα πλήθος εφαρμογών μικροελεγκτών ενσωματωμένων σε συσκευές που οι άνθρωποι χρησιμοποιούν πλέον στην καθημερινότητά τους. Κοινό στοιχείο αυτών των μικροελεγκτών είναι η καταγραφή κάποιων φυσικών μεγεθών, η επεξεργασία τους με κάποιο πρόγραμμα-αλγόριθμο και η λήψη των κατάλληλων αποφάσεων μέσω των οποίων ενεργοποιούνται οι ανάλογες δράσεις, υλοποιώντας με αυτόν τον τρόπο διάφορους αυτοματισμούς. Πολύ σημαντική θέση στη μελέτη, ανάπτυξη, δοκιμή και προτυποποίηση συστημάτων αυτοματισμού έχει το Arduino.

Στο βιβλίο αυτό παρουσιάζονται τα έργα που έχουν υλοποιηθεί στο εργαστήριο Ρομποτικής και Καινοτομίας του 1^{ου} ΕΠΑΛ Τρικάλων από τους μαθητές της Α' τάξης και της Β' τάξης του τομέα Πληροφορικής.

Οι εφαρμογές εμφανίζονται μέσα στο βιβλίο σε ξεχωριστά κεφάλαια, πιο απλά στην αρχή και πιο σύνθετα στο τέλος, έτσι ώστε οι μαθητές να χτίσουν τις γνώσεις πάνω στο Arduino σταδιακά. Επίσης, η δομή τους είναι τέτοια, ώστε το κάθε κεφάλαιο να αποτελεί και ένα ξεχωριστό φύλλο έργου που θα μπορεί να χρησιμοποιηθεί στο εργαστήριο εύκολα τόσο από τους διδάσκοντες καθηγητές όσο και από τους μαθητές.

Απαραίτητα Εξαρτήματα – Υλικά

Ηλεκτρονικά

Arduino Uno	Πλακέτα ελέγχου κινητήρα L298N
Grove Base Shield	Πλακέτα ελέγχου κινητήρα L9110
Wi-Fi module ESP8266	Πλακέτα ελέγχου κινητήρα ULN2003
Οθόνη LCD 16x2	Φωτοβολταϊκή κυψέλη 2W 136x110mm
Λάμπες LED	Αντιστάσεις 220Ω, 1kΩ, 2.2kΩ, 10kΩ
Φωτοαντιστάσεις LDR	Ποτενσιόμετρα (ροοστάτες) 10kΩ
Πυκνωτές 0.1μF, 100μF	Σταθεροποιητής τάσης LD1117
Κουμπιά (push buttons)	Ηλεκτρικός διακόπτης (relay)
Μικροδιακόπτες	Καλώδια σύνδεσης (Jumper Wires)
Breadboard	

Αισθητήρες

Υγρασίας εδάφους YL-38	Θερμοκρασίας-υγρασίας DHT11
Ποιότητας αέρα CCS811	Επαγωγικός LJ18A3-8-Z/AX
Υπερήχων HC-SR04	

Κινητήρες

Κινητήρας DC 5V	Κινητήρας Servo Micro
Αντλία νερού 12V	Βηματικός κινητήρας 28BYJ-48

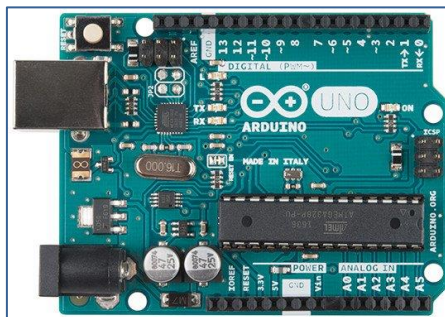
Πηγές τροφοδοσίας

Μπαταρίες 9V	Τροφοδοτικό 5V
Τροφοδοτικό 12V	

1. Βασικά στοιχεία Arduino

Το Arduino (Εικόνα 1.1) είναι μια ηλεκτρονική πλατφόρμα ανοικτού κώδικα. Αποτελείται από έναν προγραμματιζόμενο μικροελεγκτή ο οποίος είναι ενσωματωμένος πάνω σε μια ηλεκτρονική πλακέτα που περιλαμβάνει κυκλώματα τροφοδοσίας, κυκλώματα σύνδεσης με υπολογιστή, αναλογικοψηφιακούς μετατροπείς, θύρες εισόδου/εξόδου κλπ.

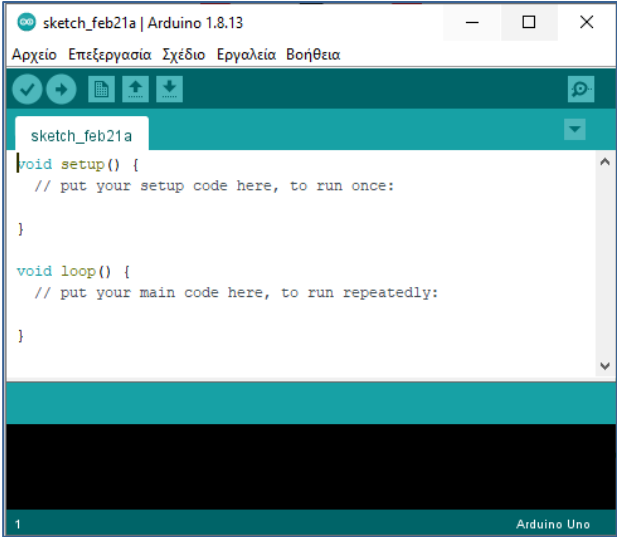
Το Arduino έχει υλοποιηθεί σε διάφορες μορφές και πλακέτες. Στις εφαρμογές που αναπτύσσονται παρακάτω χρησιμοποιείται το **Arduino Uno Rev. 3** το οποίο βασίζεται στον μικροελεγκτή ATmega328P. Έχει 14 ψηφιακές θύρες εισόδου/εξόδου (οι 6 μπορούν να λειτουργήσουν σαν PWM έξοδοι), 6 αναλογικές εισόδους, κεραμικό ταλαντωτή 16 MHz, σύνδεση USB, βύσμα τροφοδοσίας, κουμπί επαναφοράς (reset) κ.α. Η πλακέτα περιέχει όλα τα απαραίτητα υποστηρικτικά κυκλώματα και μπορεί να χρησιμοποιηθεί πολύ εύκολα, απλώς συνδέοντάς την στον υπολογιστή μέσω USB για τη μεταφόρτωση ενός προγράμματος. Στη συνέχεια, μπορεί να λειτουργήσει με την υποστήριξη ενός τροφοδοτικού ή μπαταρίας εκτελώντας το αποθηκευμένο πρόγραμμα.



Εικόνα 1.1: Πλακέτα Arduino Uno

1.1 Προγραμματισμός του Arduino

Ο προγραμματισμός του Arduino μπορεί να γίνει τόσο με κώδικα όσο και με πλακίδια (blocks) μέσω διάφορων εφαρμογών είτε εγκατεστημένων είτε online. Στο βιβλίο αυτό χρησιμοποιήθηκε το ολοκληρωμένο περιβάλλον ανάπτυξης (IDE) του Arduino που είναι μια εφαρμογή η οποία λειτουργεί σε πολλές πλατφόρμες (Windows, Linux κλπ.) και είναι διαθέσιμο για μεταφόρτωση από τον επίσημο ιστότοπο του Arduino (<https://www.arduino.cc/>). Προσφέρει πολλές ευκολίες για τη συγγραφή του προγράμματος σε γλώσσα όμοια με C++, την εγκατάσταση και συμπερίληψη διάφορων βιβλιοθηκών, τη μεταγλώττιση και, τέλος, τη μεταφόρτωση του εκτελέσιμου κώδικα στο ίδιο το Arduino. Το πρόγραμμα που γράφεται στο Arduino IDE ονομάζεται **Sketch** (Εικόνα 1.2).



```
sketch_feb21a | Arduino 1.8.13
Αρχείο Επεξεργασία Σχέδιο Εργαλεία Βοήθεια

sketch_feb21a
void setup() {
  // put your setup code here, to run once:
}

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

1 Arduino Uno
```

Εικόνα 1.2: Πρόγραμμα Sketch για Arduino

Στο Arduino μπορεί να υπάρχει μόνο ένα αποθηκευμένο πρόγραμμα κάθε φορά. Η εκτέλεση του προγράμματος ξεκινά, αμέσως μόλις το Arduino δεχθεί τροφοδοσία.

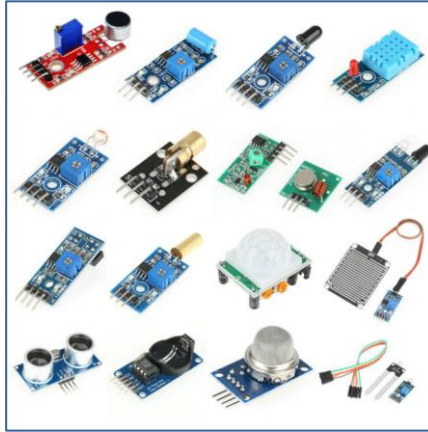
Κάθε πρόγραμμα του Arduino αποτελείται από δύο μέρη. Το ένα μέρος εκτελείται μόνο μία φορά στην αρχή (συνάρτηση **setup()** του Sketch που χρησιμοποιείται για αρχικοποιήσεις), ενώ το άλλο μέρος εκτελείται επαναληπτικά, δηλαδή ξαναρχίζει αμέσως μόλις τελειώσει (συνάρτηση **loop()** του Sketch).

Το πρόγραμμα δεν τελειώνει παρά μόνο με τη διακοπή της τροφοδοσίας. Η εκτέλεση του προγράμματος μπορεί να ξεκινήσει πάλι από την αρχή, από τη συνάρτηση **setup()**, πατώντας το κουμπί **reset** του Arduino.

Οι **θύρες** (pins) του Arduino μπορούν να διαμορφωθούν είτε σαν θύρες εισόδου είτε σαν θύρες εξόδου και να λειτουργήσουν με ψηφιακό ή μερικές από αυτές με αναλογικό τρόπο.

1.2 Θύρες εισόδου και αισθητήρες

Οι θύρες εισόδου του Arduino χρησιμοποιούνται για τη σύνδεση αισθητήρων. Οι **αισθητήρες** (Εικόνα 1.3) είναι συσκευές που ανιχνεύουν διάφορα φυσικά μεγέθη και δημιουργούν τα κατάλληλα ηλεκτρικά σήματα τα οποία μπορούν να “διαβαστούν” από το Arduino. Υπάρχει μια πληθώρα αισθητήρων: θερμοκρασίας, υγρασίας, πίεσης, αερίων, κίνησης, απόστασης, κλίσης, μαγνητικού πεδίου, μετάλλων, υπερύθρων, φωτός, χρωμάτων, δύναμης, επιτάχυνσης, επαφής, βιομετρικοί και πολλοί άλλοι.



Εικόνα 1.3: Διάφοροι αισθητήρες για Arduino

Στο Arduino υπάρχουν και ψηφιακές και αναλογικές θύρες εισόδου-εξόδου (Εικόνα 1.4). Οι ψηφιακές θύρες εισόδου μπορούν να ανιχνεύσουν τάσεις σε δύο επίπεδα: HIGH (+5V) και LOW (0V). Για το “διάβασμα” μιας ψηφιακής εισόδου από το πρόγραμμα χρησιμοποιείται η συνάρτηση ***digitalRead()***.

Οι αναλογικές εισοδοί, που συμβολίζονται με A0 έως A5, μπορούν να αναγνωρίσουν και ενδιάμεσες τιμές, καθώς το Arduino έχει ενσωματωμένο έναν αναλογικοψηφιακό μετατροπέα εύρους 10-bit (analog to digital converter) ο οποίος αντιστοιχίζει την τάση εισόδου (εύρους 0 έως 5V εξ ορισμού) σε μια ακέραια τιμή από 0 έως 1023, δηλαδή διακρίνει συνολικά $2^{10}=1024$ διαφορετικές τιμές). Η αντίστοιχη συνάρτηση στο πρόγραμμα είναι η ***analogRead()***.



Εικόνα 1.4: Ψηφιακές και Αναλογικές θύρες Arduino

1.3 Θύρες εξόδου και ενεργοποιητές

Στις θύρες εξόδου συνδέονται συσκευές που ονομάζονται **ενεργοποιητές** (actuators), καθώς δρουν στον φυσικό κόσμο δημιουργώντας φως, κίνηση, ήχους, εικόνες και γράμματα σε οθόνες κλπ. Οι ψηφιακές θύρες εξόδου μπορούν να δώσουν είτε στάθμη 5V (τιμή HIGH) είτε 0 (τιμή LOW). Η συνάρτηση που χρησιμοποιούμε στο πρόγραμμα, για να στείλουμε ένα ψηφιακό σήμα εξόδου είναι η ***digitalWrite()***.

Το ρεύμα που δίνει το Arduino από τα κυκλώματα εξόδου μπορεί να έχει ένταση έως 40mA το οποίο είναι αρκετό για την τροφοδοσία μερικών ενεργοποιητών, όπως π.χ. LED ή οθόνες LCD. Κάποιες άλλες συσκευές όμως, όπως ρελέ, ηλεκτρομαγνήτες ή κινητήρες (μοτέρ), χρειάζονται πολύ περισσότερο ρεύμα για να λειτουργήσουν, οπότε η προσπάθεια να τροφοδοτηθούν από θύρες του Arduino μπορεί να καταστρέψει τα κυκλώματα (τρανζίστορ) της θύρας ή ακόμα και ολόκληρο τον μικροελεγκτή. Παρόμοια ζημιά μπορεί να γίνει και από βραχυκύκλωμα των θυρών εξόδου.

Οι θύρες που είναι σημειωμένες με το σύμβολο “~” μπορούν να δώσουν “ψευδοαναλογική” έξοδο με την τεχνική **PWM** (Pulse Width Modulation - Διαμόρφωση Πλάτους). Πρόκειται για προσομοίωση εξόδου αναλογικών σημάτων από μια ψηφιακή έξοδο. Πραγματοποιείται με γρήγορη εναλλαγή μεταξύ των δύο καταστάσεων HIGH (+5V) και LOW (0V) η οποία κατανέμει κατάλληλα τον χρόνο που το σήμα εξόδου παραμένει στη μία κατάσταση, με τον χρόνο που παραμένει στην άλλη κατάσταση. Για παράδειγμα, ένα σήμα που εναλλάσσεται μεταξύ των δύο καταστάσεων και παραμένει σε κάθε μια για 50% του χρόνου προσομοιάζει ένα σήμα 2.5V. Στον προγραμματισμό του Arduino η

Λειτουργία αυτή υλοποιείται με τη συνάρτηση ***analogWrite()*** στην οποία μπορούμε να δώσουμε τιμές από 0 έως 255. Η τιμή 255 αντιστοιχεί στο HIGH (+5V), η τιμή 0 στο LOW και οι υπόλοιπες τιμές ενδιάμεσα.

1.4 Κινητήρες

Οι κινητήρες είναι ενεργοποιητές που μετατρέπουν την ηλεκτρική ενέργεια σε μηχανική. Στο βιβλίο αυτό χρησιμοποιούνται και τα τρία πιο συνηθισμένα είδη κινητήρων: οι κινητήρες συνεχούς ρεύματος, οι βηματικοί κινητήρες και οι κινητήρες servo.

- Οι **κινητήρες συνεχούς ρεύματος** (DC motors) είναι οι πιο συνηθισμένοι κινητήρες (Εικόνα 1.5). Έχουν δύο ακροδέκτες από όπου τροφοδοτούνται με συνεχές ρεύμα και περιστρέφονται συνεχόμενα προς τη μία φορά. Η ταχύτητα περιστροφής ρυθμίζεται από την τάση τροφοδοσίας, ενώ η φορά περιστροφής καθορίζεται από την πολικότητα της τροφοδοσίας. Κοινές εφαρμογές είναι περιστροφή ανεμιστήρων, αντλιών, τροχών κλπ.



Εικόνα 1.5:
*Κινητήρας συνεχούς
ρεύματος*

- Οι **βηματικοί κινητήρες** (stepper motors) είναι κινητήρες που έχουν τη δυνατότητα να κινούνται σε μικρά διακριτά βήματα (Εικόνα 1.6). Μπορούν να ελεγχθούν με πολύ μεγάλη ακρίβεια από το Arduino, έτσι ώστε να περιστραφούν για ένα συγκεκριμένο αριθμό βημάτων είτε προς τη μία είτε προς την άλλη φορά. Μπορούν, επίσης, να μετακινηθούν σε μια συγκεκριμένη θέση ή ακόμα να κινήσουν με συγκεκριμένη ταχύτητα. Χρησιμοποιούνται σε εφαρμογές ελέγχου μεγάλης ακριβείας, όπως π.χ. 3D εκτυπωτές.



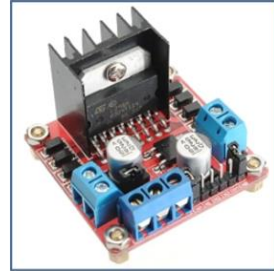
Εικόνα 1.6:
Βηματικός Κινητήρας

- Οι **κινητήρες servo** (Εικόνα 1.7) είναι κινητήρες που έχουν τη δυνατότητα να περιστρέφονται προς μια συγκεκριμένη θέση ανάλογα με το σήμα που δέχονται στην είσοδο και να παραμένουν σε αυτή τη θέση, μέχρι να αλλάξει το σήμα εισόδου. Έχουν πολύ μικρό μέγεθος αλλά συγκριτικά πολύ μεγάλη δύναμη (ροπή).



Εικόνα 1.7:
Κινητήρας Servo

- Για μεγαλύτερη ευκολία στη σύνδεση, στη ρύθμιση και στον έλεγχο των διαφόρων κινητήρων υπάρχουν ειδικές πλακέτες (Εικόνα 1.8) που ονομάζονται **οδηγοί κινητήρων** (motor drivers).



Εικόνα 1.8:
Οδηγός Κινητήρων

Λέξεις κλειδιά

<i>analogWrite()</i>	Ενεργοποιητές
<i>digitalWrite()</i>	Θύρες εισόδου-εξόδου
<i>analogRead()</i>	Κινητήρες servo
<i>digitalRead()</i>	Κινητήρες συνεχούς ρεύματος
IDE	Μικροελεγκτής
<i>loop()</i>	Προγραμματισμός Arduino
<i>reset</i>	Τεχνική PWM
<i>setup()</i>	Ψηφιακές και αναλογικές θύρες
Sketch	
Αισθητήρες	Οδηγός κινητήρων
Βηματικοί κινητήρες	

2. Αυτόματος Φωτισμός

Στο κεφάλαιο αυτό παρουσιάζεται βήμα-βήμα η δημιουργία ενός κυκλώματος αυτόματου φωτισμού που μπορεί να λειτουργήσει είτε ως ένα απλό φωτοκύτταρο στην είσοδο ενός σπιτιού είτε ως ένα ολοκληρωμένο σύστημα διαχείρισης έξυπνου φωτισμού σε μια πόλη ή εθνική οδό.

2.1 Μέτρηση απόστασης

Στην άσκηση αυτή θα συνδέσουμε αρχικά έναν **αισθητήρα υπερήχων** (Ultrasonic) με ένα Arduino Uno, προκειμένου να μετράμε την απόσταση από ένα αντικείμενο. Ένας αισθητήρας υπερήχων είναι μια ηλεκτρονική συσκευή που μετρά την απόσταση ενός αντικειμένου στόχου σε ένα εύρος 2-400 cm, εκπέμποντας υπερηχητικά κύματα, μετατρέποντας τον ανακλώμενο ήχο σε ηλεκτρικό σήμα. Οι αισθητήρες υπερήχων αποτελούνται από δύο (2) μέρη. Τον **πομπό-trigger** (ο οποίος εκπέμπει τον ήχο χρησιμοποιώντας πιεζοηλεκτρικούς κρυστάλλους) και τον **δέκτη-echo** (ο οποίος δέχεται τον ήχο, αφού έχει ταξιδέψει προς και από τον στόχο).

Για τον υπολογισμό της απόστασης μεταξύ του αισθητήρα και του αντικειμένου, ο αισθητήρας μετρά τον χρόνο που απαιτείται μεταξύ της εκπομπής του ήχου από τον πομπό στην επιφάνεια πρόσκρουσης ως την επιστροφή του στον δέκτη. Ο τύπος για αυτόν τον υπολογισμό είναι $D = \frac{1}{2} T \times C$ (όπου D είναι η απόσταση, T είναι ο χρόνος και C είναι η ταχύτητα του ήχου ~ 343 μέτρα/δευτερόλεπτο). Οι τιμές αυτές θα εμφανίζονται στη σειριακή οθόνη.

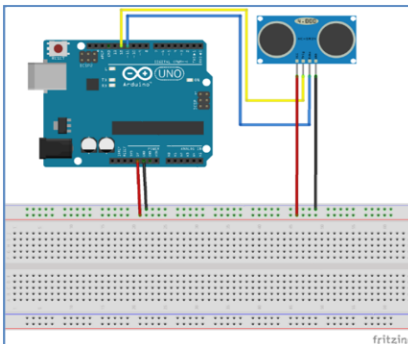
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε τα παρακάτω υλικά:

1 Arduino Uno	
1 Αισθητήρα Υπερήχων HC-SR04	
1 Breadboard	

Διάταξη κυκλώματος

Συνδέστε τα υλικά που συγκεντρώσατε χρησιμοποιώντας τη συνδεσμολογία της εικόνας 2.1.



Εικόνα 2.1:
Κύκλωμα Arduino
για μέτρηση της
απόστασης

Προγραμματίστε το κύκλωμα της εικόνας 2.1, έτσι ώστε ο αισθητήρας υπερήχων να μετράει την απόσταση σε εκατοστά κάθε 0,25 sec. Οι τιμές του αισθητήρα να εμφανίζονται στη σειριακή οθόνη του Arduino.

Κώδικας

```
/* Μέτρηση απόστασης */

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

void setup()
{
  // Αρχικοποίηση σειριακής
  Serial.begin(9600);
  // Ορισμός σημείων εισόδου-εξόδου
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop()
{
  // Ο αισθητήρας ενεργοποιείται από ψηλό παλμό 10 ή
  // περισσότερων μικροδευτερολέπτων.
  // Δίνει ένα σύντομο παλμό χαμηλού επιπέδου για να
  // εξασφαλισθεί καθαρός ψηλός παλμός
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  // Διαβάζει το σήμα από τον αισθητήρα:
  // ένας υψηλός παλμός του οποίου η διάρκεια
  // είναι ο χρόνος (σε μικροδευτερόλεπτα)
  // από την αποστολή του ping μέχρι τη λήψη της ηχούς
  // από ένα αντικείμενο-εμπόδιο
  pinMode(echoPin, INPUT);
```


2.2 Ελεγχόμενος Φωτισμός

Στην ενότητα αυτή θα εμπλουτίσουμε το κύκλωμα της προηγούμενης ενότητας, προκειμένου να ανάβει ένα LED, όταν πλησιάζει αντικείμενο στον αισθητήρα.

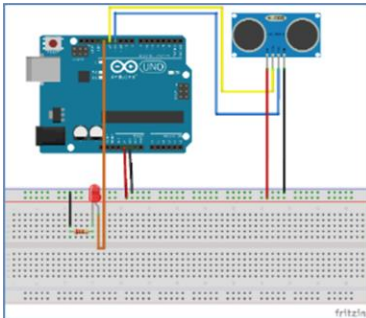
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε επιπλέον τα παρακάτω υλικά:

1 LED	
1 Αντίσταση 220Ω	

Διάταξη κυκλώματος

Προσθέστε το LED και την αντίσταση 220Ω στο υπάρχον κύκλωμα σύμφωνα με την εικόνα 2.2.



Εικόνα 2.2:
Κύκλωμα Arduino
για ελεγχόμενο
φωτισμό

Κώδικας

Τροποποιήστε το πρόγραμμα της ενότητας 2.1, έτσι ώστε να ανάβει το LED, όταν θα πλησιάζει αντικείμενο μπροστά από τον αισθητήρα υπερήχων σε απόσταση μικρότερη των 20cm.

Ο κώδικας μπορεί να διαμορφωθεί ως εξής:

```
/* Ελεγχόμενος φωτισμός */

int trigPin = 12; // Trigger
int echoPin = 11; // Echo
int led = 13; // Φωτισμός
long duration, cm, inches;

void setup()
{
  // Αρχικοποίηση σειριακής
  Serial.begin (9600);
  // Ορισμός σημείων εισόδου - εξόδου
  pinMode(trigPin, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop()
{
  // Ο αισθητήρας ενεργοποιείται από ψηλό παλμό 10
  // ή περισσότερων μικροδευτερολέπτων.
  // Δίνει ένα σύντομο παλμό χαμηλού επιπέδου για να
  // εξασφαλισθεί καθαρός ψηλός παλμός.
  digitalWrite(trigPin, LOW);
  delayMicroseconds(5);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);

  // Διαβάζει το σήμα από τον αισθητήρα: ένας ψηλός
  // παλμός του οποίου η διάρκεια είναι ο χρόνος (σε
  // μικροδευτερόλεπτα) από την αποστολή του ping μέχρι
  // τη λήψη της ηχώ από ένα αντικείμενο-εμπόδιο.
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
```

```
// Μετατροπή χρόνου σε απόσταση
cm = (duration/2) / 29.1;
// Διαίρεση με 29.1 ή πολλαπλασιασμός με 0.0343
// Ελέγχει αν διέρχεται αντικείμενο σε απόσταση
// μικρότερη από 20cm

if (cm<20)
{
  digitalWrite(led, HIGH); // Ανάβει ο φωτισμός
}
else
{
  digitalWrite(led, LOW); // Σβήνει ο φωτισμός
}

Serial.print("Απόσταση: ");
Serial.print(cm);
Serial.println("cm");
delay(250);
}
```

Ερωτήσεις

1. Τι θα συμβεί αν στον κώδικα δεν συμπεριλάβετε την εντολή
`"pinMode(led, OUTPUT);"`;
-
-

2. Τι θα συμβεί αν αντικαταστήσετε την αντίσταση 220Ω, που συνδέεται με το LED, με μια μεγαλύτερη αντίσταση των 330Ω ή του 1kΩ;
-
-

Δραστηριότητες

1. Προσθέστε άλλα 2 LED (πορτοκαλί και κίτρινο) στο κύκλωμα με τις αντίστοιχες αντιστάσεις. Προγραμματίστε τα LED να ανάβουν όλα μαζί.
2. Διαφοροποιήστε το προηγούμενο βήμα, έτσι ώστε:
 - Στα 20cm να ανάβει μόνο το κίτρινο
 - Στα 40cm να ανάβει και το πορτοκαλί
 - Στα 60cm να ανάβουν όλα μαζί.

Λέξεις κλειδιά

Αισθητήρας υπερήχων

Μέτρηση απόστασης

Αυτόματος φωτισμός

Πομπός

Δέκτης

Σειριακή οθόνη

3. Ηλιακό πάνελ

Ένα ηλιακό πάνελ έχει μεγαλύτερη απόδοση, όταν επιτυγχάνεται η μέγιστη πρόσπτωση ηλιακής ακτινοβολίας στην επιφάνειά του. Κάτι τέτοιο μπορεί να επιτευχθεί σε μεγάλο βαθμό, όταν το ηλιακό πάνελ έχει τη δυνατότητα να «ακολουθεί» τον ήλιο. Στην άσκηση αυτή θα τοποθετήσουμε ένα ηλιακό πάνελ σε μια βάση και το ηλιακό πάνελ θα περιστρέφεται με έναν κινητήρα servo σύμφωνα με τις τιμές που θα δίνουν δύο φωτοαντιστάσεις.



Για να δείτε τον τρόπο λειτουργίας του ηλιακού πάνελ, μπορείτε να παρακολουθήσετε το βίντεο της ιστοσελίδας <https://youtu.be/DYyxl5reaRc>.

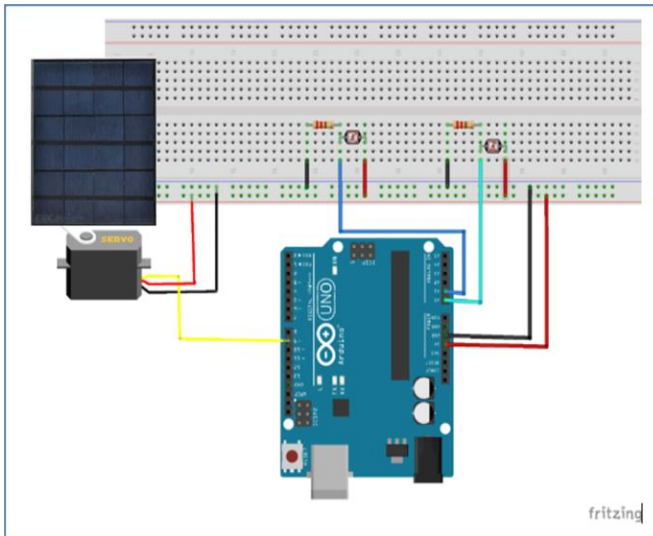
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε τα παρακάτω υλικά:

1 Arduino Uno		1 Breadboard	
1 Φωτοβολταϊκή Κυψέλη 2W 136x110mm		1 Servo Micro κινητήρας	
2 Φωτοαντιστάσεις		2 Αντιστάσεις 10kΩ	

Διάταξη κυκλώματος

Συνδέστε τα υλικά που συγκεντρώσατε, χρησιμοποιώντας τη συνδεσμολογία σύμφωνα με την εικόνα 3.1. Οι φωτοαντιστάσεις τοποθετούνται στην ανατολική και στη δυτική πλευρά της φωτοβολταϊκής κυψέλης αντίστοιχα.



Εικόνα 3.1: Κύκλωμα Arduino για ηλιακό πάνελ



Για την καλύτερη στήριξη και περιστροφή του ηλιακού πάνελ μπορείτε να χρησιμοποιήσετε μια βάση την οποία μπορείτε:

- α) να τη κατασκευάσετε μόνοι σας (οδηγίες στο βιβλίο «Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο»), ή
- β) να την εκτυπώσετε σε 3D εκτυπωτή (<https://github.com/1epaltrik/climabots/blob/master/code/solarsystem/Solarsystem%20A3.stl>)

Κώδικας

Προγραμματίστε το παραπάνω κύκλωμα, έτσι ώστε το φωτοβολταϊκό πάνελ να κινείται με την βοήθεια ενός κινητήρα servo από ανατολή προς δύση και αντίστροφα, για την επίτευξη της μέγιστης πρόσπτωσης ηλιακής ακτινοβολίας.

```

/* Ηλιακό πάνελ */

#include <Servo.h>
// Δημιουργία αντικειμένου servo για έλεγχο του
// κινητήρα servo
Servo servo;
int eLDRPin = A0; // Εκχώρηση pins στις φωτοαντιστάσεις
int wLDRPin = A1;
// Δημιουργία μεταβλητών για αποθήκευση τιμών
// φωτοαντιστάσεων
int eastLDR = 0;
int westLDR = 0;
// Δημιουργία μεταβλητής για σύγκριση τιμών των
// φωτοαντιστάσεων
int difference = 0;
// Μεταβλητή για εάν υπάρχει αξιοσημείωτη διαφορά μεταξύ
// των δύο φωτοαντιστάσεων
int error = 10;
// Μεταβλητή για έλεγχο θέσης του κινητήρα servo
// (διαφοροποιείται σε κάθε τύπο κινητήρα).
int servoSet = 130;

void setup()
{
  // Σύνδεση αντικειμένου servo στον ακροδέκτη 9
  servo.attach(9);
  Serial.begin(9600);
}

void loop()
{
  // Διάβασμα τιμών των φωτοαντιστάσεων
  eastLDR = analogRead(eLDRPin);
  westLDR = analogRead(wLDRPin);
  // Έλεγχος αν υπάρχει χαμηλός φωτισμός
  // στις 2 φωτοαντιστάσεις
  if (eastLDR < 400 && westLDR < 400)

```

```
{
  // το πάνελ κινείται ανατολικά
  while (servoSet <= 140 && servoSet >= 15)
  {
    servoSet++;
    servo.write(servoSet);
    delay(100);
  }
}
difference = eastLDR - westLDR ;

// Έλεγχος της διαφοράς τιμών των φωτοαντιστάσεων
if (difference > 10)
{
  // Μετακίνηση του πάνελ προς τη φωτοαντίσταση
  // με τη μεγαλύτερη ένδειξη φωτεινότητας
  if (servoSet <= 140)
  {
    servoSet++;
    servo.write(servoSet);
  }
}
else if (difference < -10)
{
  if (servoSet >= 15)
  {
    servoSet--;
    servo.write(servoSet);
  }
}
Serial.print(eastLDR);
// Η σειριακή οθόνη δεν είναι υποχρεωτικό να
// χρησιμοποιηθεί, αλλά μπορεί να φανεί χρήσιμη
// για εντοπισμό σφαλμάτων
Serial.print(" - ");
Serial.print(westLDR);
// Έλεγχος αν οι φωτοαντιστάσεις έχουν μεγάλη διαφορά
// στην τιμή τους. Εάν ναι, διορθώνετε την τιμή σφάλματος
Serial.print(" - ");
Serial.print(difference);
Serial.print(" - ");
Serial.print(servoSet);
// Προσαρμογή ρυθμίσεων servo, για μεγιστοποίηση της
// διαθέσιμης ταλάντευσης
Serial.print(" - ");
Serial.println(".");
delay(100);
}
```




Για τη σωστή περιστροφή του κινητήρα πρέπει να βρείτε το σημείο που θα λαμβάνεται ως σημείο εκκίνησης. Αυτό μπορεί να μεταβάλλεται ανάλογα με τον τρόπο που θα συνδεθεί το πάνελ με τον κινητήρα. Επίσης για τη δοκιμή της σωστής λειτουργίας των φωτοαντιστάσεων χρησιμοποιήστε τη σειριακή οθόνη.

Ερωτήσεις

1. Πότε θεωρείτε ότι υπάρχει χαμηλός φωτισμός στον παραπάνω κώδικα;

2. Καταγράψτε το εύρος τιμών που παρατηρείτε στις φωτοαντιστάσεις.

eastLDR	westLDR

3. Η χρήση της σειριακής οθόνη είναι υποχρεωτική σε αυτή την εφαρμογή.

Σωστό Λάθος

Δραστηριότητες

1. Συνδέστε στο κύκλωμα ένα LED με την κατάλληλη αντίσταση, έτσι ώστε να ανάβει με την ενέργεια που παράγεται από το ηλιακό πάνελ.
2. Διερευνήστε και καταγράψτε τον τρόπο με τον οποίο μπορεί να αποθηκεύεται η προσλαμβανόμενη ηλιακή ακτινοβολία σε μια συστοιχία μπαταριών.

Λέξεις κλειδιά

Ηλιακό πάνελ

Φωτοβολταϊκό πάνελ

Κινητήρας servo

Περιστροφή

Φωτοαντιστάσεις

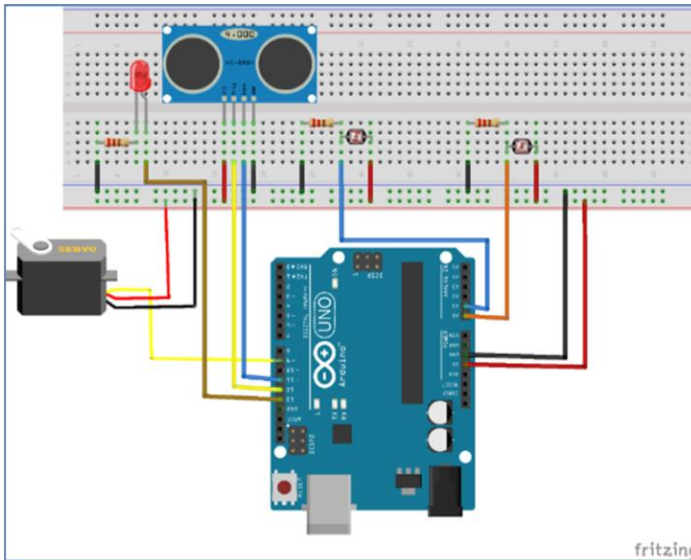
4. Συνδυασμός ηλιακού πάνελ με αυτόματο φωτισμό

Στο κεφάλαιο αυτό θα συνδυασθούν τα κυκλώματα και ο κώδικας που χρησιμοποιήθηκαν στα κεφάλαια 2 και 3.

Ο περιορισμός πόρων είναι πάντα ένα ζητούμενο σε παρόμοιες κατασκευές, αρκεί, βέβαια, να μην υπάρχει επιβάρυνση στη λειτουργία ενός συστήματος.

Διάταξη κυκλώματος

Συμπύξτε σε ένα κύκλωμα τις δύο προηγούμενες εργασίες του ηλιακού πάνελ και του αυτόματου φωτισμού σύμφωνα με την εικόνα 4.1.



Εικόνα 4.1: Κύκλωμα Arduino για συνδυασμό ηλιακού πάνελ με αυτόματο φωτισμό

Κώδικας

Ο κώδικας μπορεί να διαμορφωθεί όπως φαίνεται παρακάτω, χωρίς να είναι δεσμευτική η σειρά των εντολών.

```
/* Συνδυασμός ηλιακού πάνελ με αυτόματο φωτισμό απόστασης */

#include <Servo.h>
// Δημιουργία αντικειμένου servo για έλεγχο του
// κινητήρα servo
Servo servo;
int eLDRPin = A0; // Εκχώρηση pins στις φωτοαντιστάσεις
int wLDRPin = A1;
// Δημιουργία μεταβλητών για αποθήκευση τιμών
// φωτοαντιστάσεων
int eastLDR = 0;
int westLDR = 0;
// Δημιουργία μεταβλητής για σύγκριση τιμών των
// φωτοαντιστάσεων
int difference = 0;
// Μεταβλητή για εάν υπάρχει αξιοσημείωτη διαφορά
// μεταξύ των δύο φωτοαντιστάσεων
int error = 10;

// Μεταβλητή για έλεγχο θέσης του κινητήρα servo
// (διαφοροποιείται σε κάθε τύπο κινητήρα)
int servoSet = 130;

int trigPin = 12; // Trigger
int echoPin = 11; // Echo
int led = 13; // Φωτισμός
long duration, cm, inches;

void setup()
{
  // Σύνδεση του αντικειμένου servo στον ακροδέκτη 9
  servo.attach(9);
  // Αρχικοποίηση σειριακής
  Serial.begin(9600);
  // Ορισμός σημείων εισόδου - εξόδου
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}
```

```

void loop()
{
  // Διάβασμα τιμών των φωτοαντιστάσεων
  eastLDR = analogRead(eLDRPin);
  westLDR = analogRead(wLDRPin);
  // Ελέγχει αν υπάρχει χαμηλός φωτισμός στις
  // 2 φωτοαντιστάσεις
  if (eastLDR < 400 && westLDR < 400)
  {
    // το πάνελ κινείται ανατολικά
    while (servoSet <= 140 && servoSet >= 15)
    {
      servoSet++;
      servo.write(servoSet);
      delay(100);
    }
  }

  difference = eastLDR - westLDR ;
  // Έλεγχος της διαφοράς τιμών των φωτοαντιστάσεων
  if (difference > 10)
  {
    // Μετακινεί το πάνελ προς τη φωτοαντίσταση
    // με τη μεγαλύτερη ένδειξη
    if (servoSet <= 140)
    {
      servoSet++;
      servo.write(servoSet);
    }
  }
  else if (difference < -10)
  {
    if (servoSet >= 15)
    {
      servoSet--;
      servo.write(servoSet);
    }
  }
}

Serial.print(eastLDR);

// Η σειριακή οθόνη δεν είναι υποχρεωτικό να
// χρησιμοποιηθεί, αλλά μπορεί να φανεί χρήσιμη
// για εντοπισμό σφαλμάτων
Serial.print(" - ");
Serial.print(westLDR);

```

```
// Έλεγχος αν οι φωτοαντιστάσεις έχουν μεγάλη διαφορά
// στην τιμή τους. Εάν ναι, διορθώνετε την τιμή
// σφάλματος
Serial.print(" - ");
Serial.print(difference);
Serial.print(" - ");
Serial.print(servoSet);
// Προσαρμογή ρυθμίσεων servo, για μεγιστοποίηση της
// διαθέσιμης ταλάντευσης
Serial.print(" - ");
Serial.println(".");
delay(100);

// Ο αισθητήρας ενεργοποιείται από ψηλό παλμό 10
// ή περισσότερων μικροδευτερολέπτων
// Δίνει ένα σύντομο παλμό χαμηλού επιπέδου για να
// εξασφαλισθεί καθαρός ψηλός παλμός.
digitalWrite(trigPin, LOW);
delayMicroseconds(5);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
// Διαβάζει το σήμα από τον αισθητήρα: ένας υψηλός
// παλμός του οποίου η διάρκεια είναι ο χρόνος (σε
// μικροδευτερόλεπτα) από την αποστολή του ping μέχρι
// τη λήψη της ηχώ από ένα αντικείμενο-εμπόδιο
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);
// Μετατροπή χρόνου σε απόσταση
cm = (duration/2) / 29.1;
// Διαίρεση με 29.1 ή πολλαπλασιασμός με 0.0343
// Έλεγχος αν διέρχεται αντικείμενο σε απόσταση
// μικρότερη από 20cm
if (cm < 20)
{
    digitalWrite(led, HIGH); // Ανάβει ο φωτισμός
}
else
{
    digitalWrite(led, LOW); // Σβήνει ο φωτισμός
}

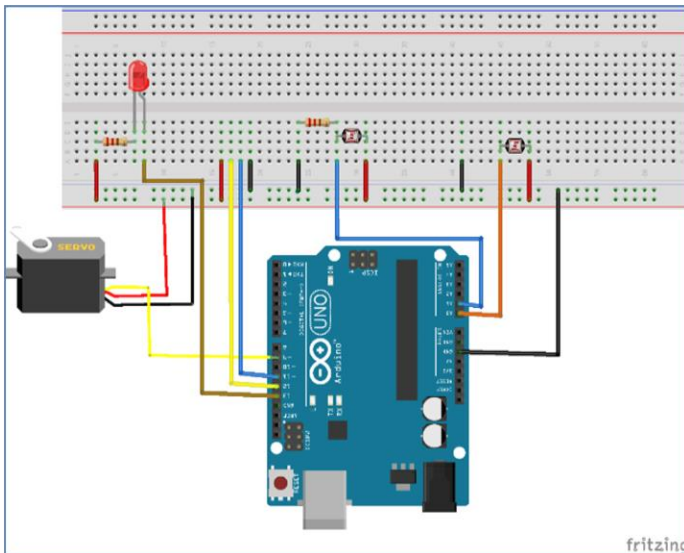
Serial.print("Απόσταση: ");
Serial.print(cm);
Serial.println("cm");
delay(250);
}
```

Δραστηριότητες

1. Καταγράψτε τα 4 υλικά που εξοικονομήσατε συνδυάζοντας σε ένα κύκλωμα τον ελεγχόμενο φωτισμό και το ηλιακό πάνελ.

1.	2.
3.	4.

2. Εντοπίστε τα 4 λάθη που υπάρχουν στη διάταξη του κυκλώματος της εικόνας 4.2 ώστε να συμφωνεί με αυτό της εικόνας 4.1.



Εικόνα 4.2: Λανθασμένο κύκλωμα Arduino

Λέξεις κλειδιά

Αισθητήρας υπερήχων

Αυτόματος φωτισμός

Δέκτης

Ηλιακό πάνελ

Κινητήρας servo

Μέτρηση απόστασης

Περιστροφή

Πομπός

Σειριακή οθόνη

Φωτοαντιστάσεις

5. Έξυπνο Σκίαστρο

Στο κεφάλαιο αυτό θα κατασκευάσουμε ένα έξυπνο σκίαστρο, δηλαδή μια τέντα που θα ανοιγοκλείνει αυτόματα ανάλογα με τον περιβάλλοντα φωτισμό. Η τέντα θα είναι τυλιγμένη σε έναν άξονα ο οποίος θα μπορεί να περιστρέφεται με τη βοήθεια ενός βηματικού κινητήρα (stepper motor). Ένας αισθητήρας φωτός θα ανιχνεύει την ένταση του περιβάλλοντος φωτισμού και το Arduino θα ενεργοποιεί ανάλογα τον κινητήρα για το άνοιγμα ή κλείσιμο της τέντας.



Για τη στήριξη όλου του μηχανισμού σχεδιάσαμε μια βάση στο Tinkercad και την εκτυπώσαμε στον 3D εκτυπωτή (οδηγίες στο βιβλίο «Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο»). Στη σχεδίαση λάβαμε υπόψη και τις διαστάσεις των εξαρτημάτων που περιγράφονται στην παρακάτω λίστα υλικών.

Μπορείτε να κατεβάσετε τα μέρη της βάσης από τις ιστοσελίδες:

- <https://github.com/1epaltrik/climabots/blob/master/code/solarsystem/Σκίαστρο%20-%20αριστερή%20πλευρά.stl>
- <https://github.com/1epaltrik/climabots/blob/master/code/solarsystem/Σκίαστρο%20-%20δεξιά%20πλευρά.stl>
- <https://github.com/1epaltrik/climabots/blob/master/code/solarsystem/Σκίαστρο-Δοκός.stl>

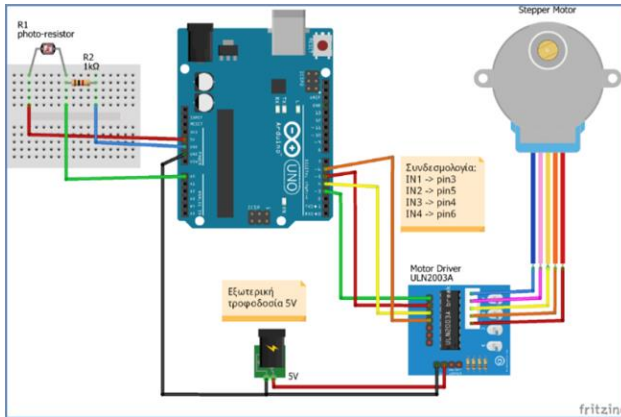
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε τα παρακάτω υλικά:

1 Arduino Uno		1 Breadboard	
1. Βηματικός κινητήρας π.χ. 28BYJ-48		1 Πλακέτα οδήγησης του stepper motor π.χ. Stepper Driver Breakout - ULN2003	
1 Φωτοαντίσταση LDR (photoresistor)		1 Αντιστάση 1kΩ	
1 Τροφοδοτικό 5V, π.χ. από USB		Καλώδια σύνδεσης (Jumper Wires)	
1 Άξονας γύρω από τον οποίο θα τυλίγεται η τέντα (20cm μήκος, 5mm διάμετρος)		1 Εξάρτημα σύνδεσης του άξονα με τον άξονα του Stepper Motor (5mm εσωτερική διάμετρος)	
1 Κομμάτι υφάσματος τέντας διαστάσεων 17,50cm x 12,00cm περίπου			

Διάταξη κυκλώματος

Συνδέστε τα υλικά που συγκεντρώσατε χρησιμοποιώντας τη συνδεσμολογία σύμφωνα με την εικόνα 5.1.



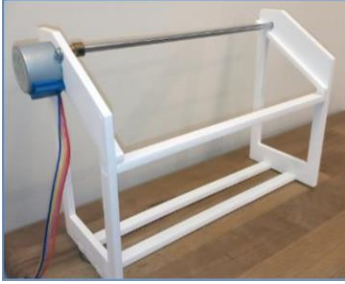
Εικόνα 5.1: Κύκλωμα Arduino για έξυπνο σκίαστρο

Συναρμολόγηση

Στην κατασκευή θα χρησιμοποιηθούν τα αντικείμενα που δημιουργήθηκαν και εκτυπώθηκαν στον 3D εκτυπωτή (οδηγίες στο βιβλίο «Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο»).

- Συναρμολογήστε τα κομμάτια της βάσης όπως φαίνεται στην εικόνα 5.1.
- Βιδώστε τον βηματικό κινητήρα στη βάση (εικόνα 5.2).
- Συνδέστε τον άξονα του βηματικού κινητήρα με τον άξονα της τέντας βιδώνοντας το αντίστοιχο εξάρτημα σύνδεσης και στους 2 άξονες (Εικόνες 5.3 και 5.4)
- Τέλος, προσαρμόστε και στερεώστε το πανί της τέντας στον άξονα, όπως φαίνεται στην εικόνα 5.5. Ένας επιπλέον

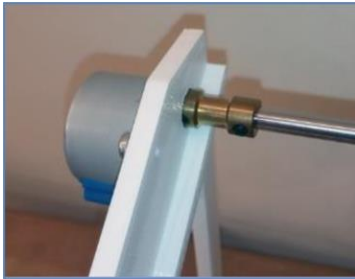
άξονας στο κάτω μέρος της τέντας χρησιμοποιείται ως αντίβαρο, έτσι ώστε η τέντα να παραμένει τεντωμένη.



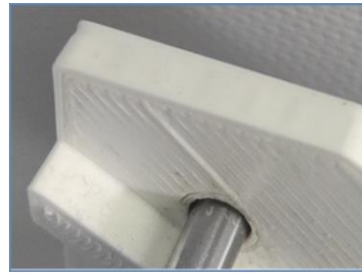
Εικόνα 5.1: Ο σκελετός της βάσης της τέντας



Εικόνα 5.2: Ο βηματικός κινητήρας βιδωμένος στο πλαϊνό στήριγμα



Εικόνα 5.3: Ο βηματικός κινητήρας συνδέεται με τον άξονα με το ειδικό εξάρτημα



Εικόνα 5.4: Σημείο στερέωσης του άξονα της τέντας



Εικόνα 5.5:
Τελική μορφή
έξυπνου σκίαστρου

Κώδικας

Προγραμματίστε το Arduino, έτσι ώστε να κινείται ο βηματικός κινητήρας (άρα και η τέντα) ανάλογα με την τιμή του περιβάλλοντος φωτισμού που διαβάζει ο αισθητήρας στην είσοδο A0. Αρχικά, η τέντα θα είναι τελείως κλειστή (δηλ. μαζεμένη-τυλιγμένη στον άξονα) και ανάλογα με το φως είτε θα ανοίγει μέχρι τη μέση είτε θα ανοίγει εντελώς. Όταν ο φωτισμός είναι μέγιστος, η τέντα θα είναι πλήρως ανοιχτή.

```

/*
 * Έξυπνο Σκίαστρο
 * Το πρόγραμμα ρυθμίζει το άνοιγμα της τέντας
 * ανάλογα με το φωτισμό
 * σε μια από 3 θέσεις: κλειστή, μισάνοιχτη, ανοιχτή.
 * Υποθέτουμε ότι η αρχική θέση της τέντας είναι
 * πλήρως τυλιγμένη (κλειστή).
 */

#include <Stepper.h>

// Το stepper motor 28BYJ-48 κάνει 2048 βήματα σε μια
// περιστροφή (επειδή σε Full Step mode κάνει
// 32 steps per revolution * 64 Gear Ratio = 2048)
const int stepsPerRevolution = 2048;

// Ο βηματικός κινητήρας συνδέεται στα pins 3 έως 6
// προσοχή στα μεσαία pins: συνδέονται ανάποδα
Stepper myStepper(stepsPerRevolution, 3, 4, 5, 6);

// Ο αριθμός βημάτων που πρέπει να κάνει ο κινητήρας για να
// ανοίξει πλήρως η τέντα (εξαρτάται από το μήκος της)
const int maxSteps = 3500;

// Η τρέχουσα θέση της τέντας: 0 κλειστή, 1 μισάνοιχτη,
// 2 ανοιχτή
int pos = 0;

// Τα βήματα της περιστροφής του μοτέρ
int steps = 0;

```

```
// Η τιμή του αισθητήρα φωτός
int lightSensor = 0;

void setup()
{
  // Ορισμός της ταχύτητας του μοτέρ σε rpm
  // (περιστροφές ανά λεπτό)
  myStepper.setSpeed(14);

  // Αρχικοποίηση σειριακής θύρας για monitor
  Serial.begin(9600);

  delay(500);
}

void loop()
{
  // Διάβασμα της αναλογικής εισόδου (τιμής του αισθητήρα)
  lightSensor = analogRead(A0);

  // Εμφάνιση της τιμής στο monitor
  Serial.println(lightSensor);

  steps = 0;
  if( pos == 0 ) // Αν η τέντα είναι κλειστή.
  {
    if( lightSensor > 600 ) // και αν υπάρχει φως,
    {
      pos = 1; // τότε η τέντα ανοίγει ως τη μέση.
      steps = maxSteps/2;
    }
  }
  else if( pos == 1 ) // Αν η τέντα είναι μισάνοιχτη
  {
    if( lightSensor > 850 ) // και αν υπάρχει πολύ φως,
    {
      pos = 2; // τότε η τέντα ανοίγει πλήρως,
      steps = maxSteps/2;
    }
    else if( lightSensor < 600 ) // αλλά αν δεν έχει φως,
    {
      pos = 0; // τότε η τέντα κλείνει.
      steps = -maxSteps/2;
    }
  }
  else if( pos == 2 ) // Αν η τέντα είναι τελείως ανοιχτή...
  {
```

```

if( lightSensor < 850 ) // και αν δεν έχει πολύ φως,
{
  pos = 1;           // τότε η τέντα κλείνει μέχρι τη μέση
  steps = -maxSteps/2;
}
}

// Εμφάνιση στη σειριακή οθόνη για έλεγχο
Serial.print("steps=");
Serial.println(steps);
Serial.print("pos=");
Serial.println(pos);

// Ενεργοποίηση του μοτέρ
myStepper.step(steps);

// Μια καθυστέρηση 0.5 sec
delay(500);
}

```

Δραστηριότητες

1. Στο προηγούμενο πρόγραμμα η τέντα βρίσκεται σε μία από τις 3 πιθανές θέσεις: κλειστή, μισάνοιχτη, ανοιχτή. Βελτιώστε το πρόγραμμα, έτσι ώστε η τέντα να μπορεί να βρίσκεται σε 4 θέσεις. Προσθέστε, δηλαδή, επιπλέον μια ενδιάμεση θέση.
2. Αλλάξτε τη λογική του προγράμματος: αντί για 3 ή 4 πιθανές θέσεις της τέντας, η τέντα να ανοίγει σύμφωνα με την ένταση του περιβάλλοντος φωτισμού με αναλογικό τρόπο. Δηλαδή, όσο πιο πολύ το φως, τόσο πιο πολύ να ανοίγει η τέντα.

Λέξεις κλειδιά

Αισθητήρας φωτός

Σκίαστρο

Βηματικός κινητήρας

Φωτοαντίσταση

Πλακέτα οδήγησης

Περιβάλλον φωτισμός



Εικόνα 5.6: Το έξυπνο σκίαστρο

6. Πότισμα

Η κατανάλωση τεράστιων ποσοτήτων νερού για το πότισμα φυτών ή την άρδευση μεγάλων καλλιεργειών, με την παράλληλη δραματική μείωση των αποθεμάτων, λόγω και των κλιματικών αλλαγών, αποτελούν απειλή για την ανθρωπότητα. Η εγκατάσταση αυτοματοποιημένων συστημάτων ποτίσματος, όταν και μόνο υπάρχει ανάγκη, τόσο σε οικίες όσο και σε καλλιέργειες, μπορεί να συμβάλλει στη μείωση αυτού του φαινομένου.

6.1 Έξυπνο πότισμα

Στην υπο-ενότητα αυτή θα δημιουργήσουμε ένα έξυπνο σύστημα αυτόματου ποτίσματος, όπου:

- Η αντλία νερού θα ενεργοποιείται μόνο στην περίπτωση που η υγρασία του εδάφους θα είναι μικρότερη από ένα επίπεδο υγρασίας που θα ορίσουμε.
- Μια οθόνη θα εμφανίζει ως ένδειξη την υγρασία του εδάφους και αν απαιτείται πότισμα.
- Ένας ροοστάτης θα ρυθμίζει τη φωτεινότητα της οθόνης.
- Δύο LED, πράσινο και κόκκινο θα ανάβουν σε περίπτωση που η αντλία νερού είναι ενεργή ή όχι (πράσινο ανοιχτή – κόκκινο κλειστή).
- Η τροφοδοσία αυτού του συστήματος μπορεί να γίνει μέσω κατάλληλης μπαταρίας ή μετασχηματιστή.

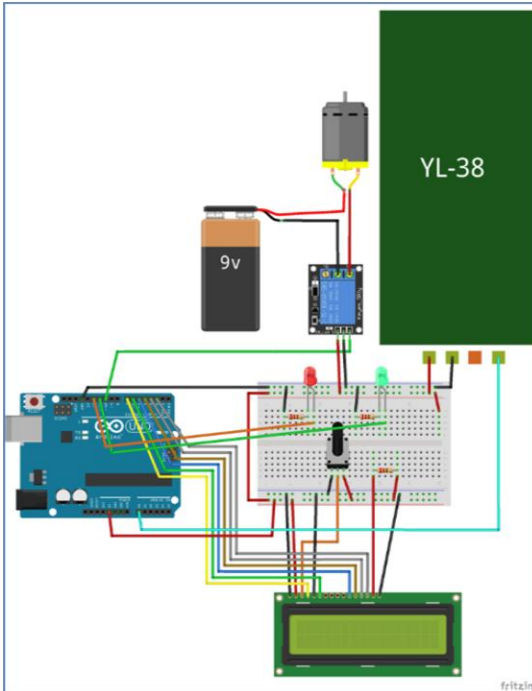
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε τα παρακάτω υλικά:

1 Arduino Uno και 1 Grove Base shield		1 Ροοστάτης (μεταβλητή αντίσταση)	
1 Relay (ηλεκτρικός διακόπτης)		1 Αισθητήρας Υγρασίας Εδάφους YL-38	
2 Λάμπες LED		2 Αντιστάσεις 220Ω	
1 Τροφοδοτικό 5V, π.χ. από USB		1 Πηγή τροφοδοσίας αντλίας νερού (τροφοδοτικό ή μπαταρία 9-12V)	
1 Οθόνη LCD με οπίσθιο φωτισμό RGB		1 Αντλία νερού 12V (Liquid Pump Motor)	

Διάταξη κυκλώματος

Συνδέστε τα υλικά που συγκεντρώσατε και τοποθετήστε τα χρησιμοποιώντας τη συνδεσμολογία της εικόνας 6.1.



Εικόνα 6.1:
*Κύκλωμα Arduino
για έξυπνο πότισμα*



Για να πετύχετε μια λειτουργική, φορητή και όμορφη κατασκευή μπορείτε να εκτυπώσετε σε έναν 3D εκτυπωτή ένα κουτί και το καπάκι του (οδηγίες στο βιβλίο «Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο»).

Για το κουτί επισκεφθείτε την ιστοσελίδα:

<https://github.com/1epaltrik/climabots/blob/master/code/hygrometer/ΥΓΡΑΣΙΟΜΕΤΡΟ ΚΟΥΤΙ.stl>

Για το καπάκι του επισκεφθείτε την ιστοσελίδα:

<https://github.com/1epaltrik/climabots/blob/master/code/hygrometer/ΥΓΡΑΣΙΟΜΕΤΡΟ ΚΑΠΑΚΙ.stl>.

Κώδικας

Προγραμματίστε το παραπάνω κύκλωμα, έτσι ώστε να ενεργοποιείται η αντλία νερού, όσο η υγρασία του εδάφους είναι μικρότερη από 45%.

```
/* Πότισμα */

const int hygrometer = A0;
int value;
#include <LiquidCrystal.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);
const int motorPin = 10; // Θέση αντλίας
const int greenLed = 11; // Θέση πράσινου LED
const int redLed = 12; // Θέση κόκκινου LED
void setup()
{
  pinMode(motorPin, OUTPUT); // Ορισμός σημείων εξόδου
  pinMode(greenLed, OUTPUT);
  pinMode(redLed, OUTPUT);
  Serial.begin(9600); // Αρχικοποίηση σειριακής
  lcd.begin(16, 2); // Αρχικοποίηση οθόνης LCD
}

void loop()
{
  lcd.clear(); // Καθαρισμός οθόνης
  value = analogRead(hygrometer); // Διάβασμα υγρασίας
  value = constrain(value, 400, 1023);
  // Μετατροπή υγρασίας σε ποσοστό
  value = map(value, 400, 1023, 100, 0);
  // Εμφάνιση υγρασίας στη σειριακή οθόνη
  Serial.print("YGRASIA EDAFOUS: ");
  Serial.print(value);
  Serial.println("");
  lcd.setCursor(0, 0); // Ορισμός αρχικής θέσης οθόνης LCD
  lcd.print("YGRASIA:"); // Εμφάνιση υγρασίας στην οθόνη LCD
  lcd.setCursor(8, 0);
  lcd.print(value);
  lcd.print("%");
  if (value < 45)
  { // Έλεγχος αν η υγρασία είναι <45%
    lcd.setCursor(0, 1);
    lcd.print("THELEI POTISMA");
    digitalWrite(motorPin, HIGH); // Ανοίγει η αντλία
```

```

    digitalWrite(greenLed, HIGH); // Ανάβει το πράσινο LED
    digitalWrite(redLed, LOW);    // Σβήνει το κόκκινο LED
}
else
{
    lcd.setCursor(0, 1);
    lcd.print("EXEI ARKETO NERO");
    digitalWrite(motorPin, LOW); // Κλείνει η αντλία
    digitalWrite(greenLed, LOW); // Σβήνει το πράσινο LED
    digitalWrite(redLed, HIGH);  // Ανάβει το κόκκινο LED
}
delay(1000);
}

```

Ερωτήσεις

- Σε ποιο σημείο θα ορισθεί η αρχική θέση οθόνης LCD αν η εντολή `lcd.setCursor(0, 0)`; αντικατασταθεί με την `lcd.setCursor(1, 0)`;

α. 1 ^η γραμμή, 1 ^η στήλη	γ. 2 ^η γραμμή, 1 ^η στήλη
β. 1 ^η γραμμή, 2 ^η στήλη	δ. 2 ^η γραμμή, 2 ^η στήλη
- Ποια είναι η βιβλιοθήκη της οθόνης LCD 16x2;

α. LiquidCrystal.h	γ. LED.h
β. LCD.h	δ. Liquid.h
- Η εντολή `lcd.clear()`; χρησιμοποιείται για τον καθαρισμό της οθόνης.

Σωστό Λάθος

6.2 Έξυπνο πότισμα τη νύχτα

Ένα μεγάλο μέρος του νερού που χρησιμοποιείται για το πότισμα τους θερινούς μήνες χάνεται καθώς υγροποιείται. Θα τροποποιήσουμε το σύστημα ποτίσματος της ενότητας 6.1, να ανοίγει η αντλία νερού μόνο το βράδυ, προκειμένου να πετύχουμε μέγιστη απόδοση στο πότισμα.

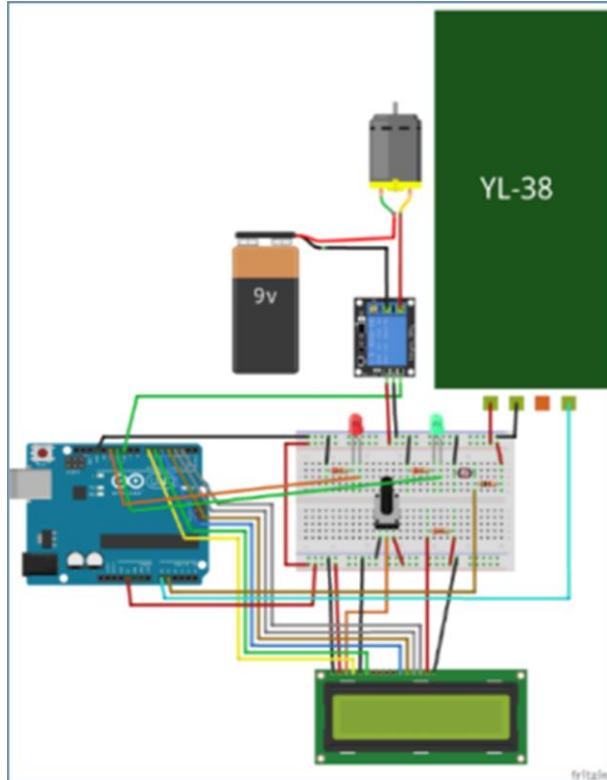
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε επιπλέον τα παρακάτω υλικά:

1 Φωτοαντίσταση	
1 Αντίσταση 10kΩ	

Διάταξη κυκλώματος

Συνδέστε τη φωτοαντίσταση και την αντίσταση, σύμφωνα με την εικόνα 6.2.



Εικόνα 6.2: Κύκλωμα Arduino για έξυπνο πότισμα την νύχτα

Κώδικας

Τροποποιήστε τον κώδικα της προηγούμενης ενότητας, έτσι ώστε η αντλία νερού να ενεργοποιείται μόνο όταν η τιμή της φωτεινότητας είναι μικρότερη από 500.

```
/* Εξυπνο πότισμα την ημέρα */

#include <LiquidCrystal.h>
const int hygrometer = A0;
const int pResistor = A1; // Θέση φωτοαντίστασης
const int motorPin = 10; // Θέση αντλίας
const int greenLed = 11; // Θέση πράσινου LED
const int redLed = 12; // Θέση κόκκινου LED
int value, valuep;
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup()
{
  Serial.begin(9600);
  lcd.begin(16, 2);
  pinMode(greenLed, OUTPUT);
  pinMode(redLed, OUTPUT);
  pinMode(motorPin, OUTPUT);
  // Ορισμός φωτοαντίστασης ως σημείο εισόδου
  pinMode(pResistor, INPUT);
}

void loop()
{
  lcd.clear();
  // Διάβασμα τιμής φωτοαντίστασης
  Valuep = analogRead(pResistor);
  value = analogRead(hygrometer);
  value = constrain(value, 400, 1023);
  value = map(value, 400, 1023, 100, 0);
  Serial.print("Soil humidity: ");
  Serial.print(value);

  Serial.println("%");
  lcd.print("YGRASIA:");
  lcd.setCursor(8, 0);
  lcd.print(value);
  lcd.print("%");
  if (value < 45) // Έλεγχος αν η υγρασία είναι <45%
  {
    lcd.setCursor(0, 1);
    lcd.print("ΑΠΑΙΤΕΙ ΠΟΤΙΣΜΑ");
    digitalWrite(greenLed, HIGH); // Ανάβει το πράσινο LED
    digitalWrite(redLed, LOW); // Σβήνει το κόκκινο LED
    if (valuep<500) // Σε περίπτωση χαμηλής φωτεινότητας
    {
      digitalWrite(motorPin, HIGH); // Ανοίγει η αντλία
    }
  }
}
```



```
    }  
    else  
    {  
        digitalWrite(motorPin, LOW); // Κλείνει η αντλία  
    }  
}  
else  
{  
    lcd.setCursor(0, 1);  
    lcd.print("EXEI ARKETO NERO");  
    digitalWrite(motorPin, LOW);  
    digitalWrite(greenLed, LOW); // Σβήνει το πράσινο LED  
    digitalWrite(redLed, HIGH); // Ανάβει το κόκκινο LED  
}  
delay(1000);  
}
```



Εικόνα 6.3: Τελική μορφή του κουτιού που περιέχει το σύστημα ποτίσματος

Δραστηριότητες

1. Τι πρέπει να προστεθεί στο κύκλωμά σας, αν πρόκειται να το χρησιμοποιήσετε σε μια μεγάλη έκταση, π.χ. 1 στρέμματος;

2. Προσαρμόστε στο κύκλωμά σας άλλους δύο αισθητήρες υγρασίας εδάφους YL-38.
3. Πώς θα τροποποιήσετε τον κώδικά σας, προκειμένου να ενεργοποιείται η αντλία, όταν ο μέσος όρος των αισθητήρων υγρασίας εδάφους YL-38, υποχωρεί κάτω από 40%;

Λέξεις κλειδιά

Αντλία νερού

Υγρασία εδάφους

Αισθητήρας Υγρασίας

Φωτεινότητα

Ροοστάτης

Φωτοαντίσταση

7. Διαχωριστής Μεταλλικών Αντικειμένων

Ο διαχωριστής μεταλλικών αντικειμένων είναι μια πρωτότυπη κατασκευή που μπορεί να διαχωρίζει μεταλλικά από μη μεταλλικά αντικείμενα. Η ιδέα είναι η εξής:

Το αντικείμενο θα τοποθετείται πάνω σε έναν κάδο, όπου θα ανιχνεύεται αν είναι μεταλλικό ή όχι με τη βοήθεια ενός αισθητήρα μετάλλων. Κατόπιν, ο κάδος θα κινείται πάνω σε μια διαδρομή στο τέλος της οποίας θα ανατρέπεται, για να αδειάσει το αντικείμενο είτε προς τη μια πλευρά, αν είναι μεταλλικό, είτε προς την άλλη, αν δεν είναι μεταλλικό.

Για τη μεταφορά του αντικειμένου θα αξιοποιηθεί ο μηχανισμός κίνησης της κεφαλής ενός παλιού εκτυπωτή. Πάνω στην κεφαλή εκτύπωσης θα στερεωθεί ο κάδος που σχεδιάσαμε στο Tinkercad και εκτυπώσαμε στον 3D εκτυπωτή. Ο κάδος θα στηρίζεται πάνω σε μια βάση, έτσι ώστε να μπορεί να περιστρέφεται, για να αδειάζει το περιεχόμενό του.

Ένας βηματικός κινητήρας θα αναλάβει την περιστροφή του κάδου. Δύο μικροδιακόπτες θα τοποθετηθούν στα όρια της διαδρομής του κάδου, για να σηματοδοτούν το τέλος της διαδρομής. Ένας αισθητήρας μετάλλων θα ανιχνεύει τα μεταλλικά αντικείμενα.

Για τη μείωση της πολυπλοκότητας και την ευκολότερη υλοποίηση αυτού του project, το χωρίσαμε σε 5 μικρότερες εργασίες, οι οποίες περιγράφονται αναλυτικά στις επόμενες ενότητες. Κάθε εργασία αποτελεί συνέχεια της προηγούμενης.

7.1 Έλεγχος κινητήρα DC

Στην εργασία αυτή θα χρησιμοποιήσουμε δύο κουμπιά για να ελέγξουμε τη φορά περιστροφής ενός κινητήρα: το ένα θα περιστρέφει τον κινητήρα προς τη μία κατεύθυνση και το άλλο προς την άλλη.

Λίστα Υλικών

1 Arduino Uno		2 Κουμπιά Push Button	
Καλώδια σύνδεσης Jumper Wires		2 Αντιστάσεις 10kΩ	
1 Πλακέτα ελέγχου του κινητήρα, όπως αυτή που χρησιμοποιεί το ολοκληρωμένο L298N τύπου H-Bridge.		1 Πηγή Τροφοδοσίας για το μοτέρ (μπαταρία 6V, 9V ή ένα τροφοδοτικό 12V, ανάλογα με το μοτέρ)	
Ένας μηχανισμός κίνησης της κεφαλής εκτύπωσης από έναν παλιό εκτυπωτή. Ο μηχανισμός αυτός αποτελείται από το φορείο, το οποίο παίρνει κίνηση από έναν κινητήρα συνεχούς ρεύματος (DC motor) μέσω ενός ελαστικού οδοντωτού ιμάντα και κινείται κατά μήκος ενός άξονα. Όλα τα εξαρτήματα είναι στερεωμένα πάνω σε μια μεταλλική βάση.			

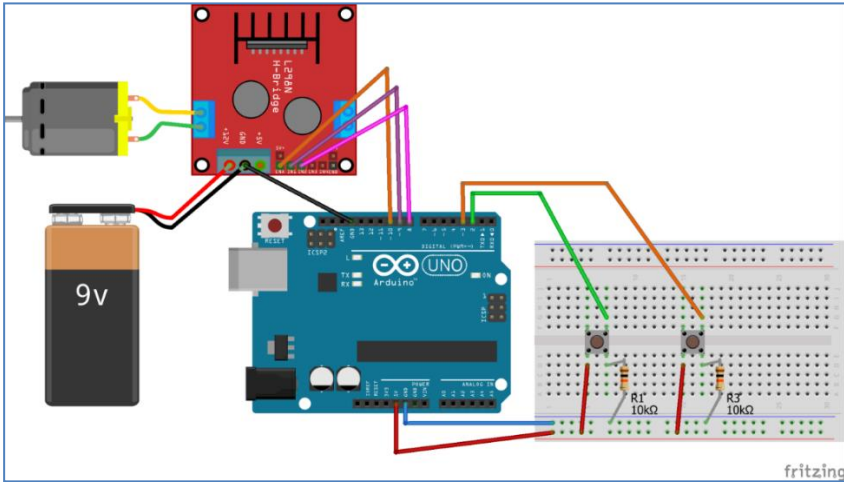
Πλακέτα ελέγχου κινητήρα DC (motor driver)

Η συνδεσμολογία της πλακέτας ελέγχου του κινητήρα αναλύεται παρακάτω:

Συνδέσεις			
1, 2	Εδώ συνδέεται το μοτέρ συνεχούς ρεύματος (DC)		
3	Βραχυκυκλωτήρας. Πρέπει να αφαιρεθεί αν στην επόμενη σύνδεση 4 εφαρμόζεται τάση πάνω από 12V DC		
4	Ο θετικός πόλος της μπαταρίας (τροφοδοσία μοτέρ)		
7	Ο βραχυκυκλωτήρας αφαιρείται και συνδέεται η έξοδος (PWM οπωσδήποτε) του Arduino που ελέγχει την ταχύτητα του μοτέρ	8, 9	IN1 και IN2: συνδέονται σε 2 ψηφιακές εξόδους του Arduino και καθορίζουν τη φορά περιστροφής του μοτέρ
5	Γείωση (GND). Ο αρνητικός πόλος της μπαταρίας, που συνδέεται οπωσδήποτε και με τη γείωση του Arduino	10 έως 14	Εδώ μπορεί να συνδεθεί και δεύτερο μοτέρ
6	Είσοδος 5V για την τροφοδοσία της πλακέτας		

Διάταξη κυκλώματος

Υλοποιήστε το παρακάτω κύκλωμα της εικόνας 7.1. Ο μηχανισμός του εκτυπωτή παριστάνεται ως ένα απλό μοτέρ DC.



Εικόνα 7.1: Κύκλωμα ελέγχου κινητήρα DC

Κώδικας

Προγραμματίστε το παραπάνω κύκλωμα έτσι ώστε να ελέγχετε τον κινητήρα.

```
/* Κινητήρας DC */  
  
// Χρησιμοποιούμε σταθερές για τους αριθμούς θυρών:  
const int buttonPin1 = 2;  
const int buttonPin2 = 3;  
  
// Χρησιμοποιούμε μεταβλητές που αποθηκεύουν την κατάσταση  
// των κουμπιών. LOW: όχι πατημένο, HIGH: πατημένο  
int buttonState1 = LOW;  
int buttonState2 = LOW;
```

```

// Οι συνδέσεις των pins του Arduino με τα pins
// της πλακέτας ελέγχου του μοτέρ (L298N H-Bridge)
const int enA = 10;
const int in1 = 9;
const int in2 = 8;

void setup()
{
  // Ορίζουμε ότι τα κουμπιά είναι σε θύρες εισόδου:
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);

  // Ορισμός όλων των pins που συνδέονται στον
  // ελεγκτή μοτέρ ως εξόδου
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
}

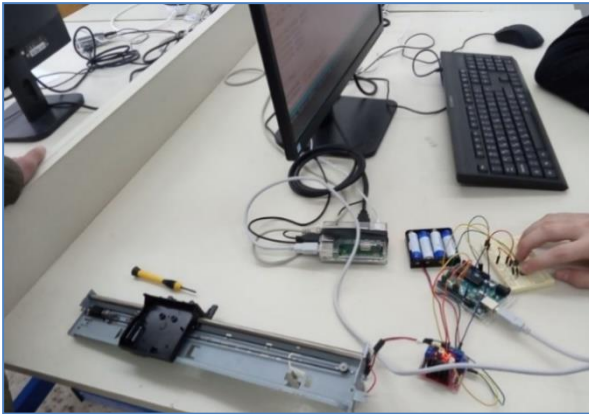
void loop()
{
  // Διαβάζουμε τη θύρα κάθε κουμπιού και αποθηκεύουμε
  // την κατάστασή του:
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);
  // Αν δεν είναι πατημένο κανένα κουμπί:
  if (buttonState1 == LOW && buttonState2 == LOW)
  {
    // Ορισμός της ταχύτητας στο 0
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    analogWrite(enA, 0);
  }
  else if (buttonState1 == HIGH)
  // Αλλιώς αν είναι πατημένο το 1ο κουμπί:
  {
    // Πήγαινε προς τη μία κατεύθυνση
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // Ορισμός της ταχύτητας στο εύρος 0~255
    analogWrite(enA, 240);
  }
  else if (buttonState2 == HIGH)
  // Αλλιώς αν είναι πατημένο το 2ο κουμπί:
  {
    // Πήγαινε προς την άλλη κατεύθυνση
    digitalWrite(in1, LOW);
  }
}

```

```
digitalWrite(in2, HIGH);  
// Ορισμός της ταχύτητας από το εύρος 0~255  
analogWrite(enA, 220);  
}  
}
```



Ελέγξτε τη λειτουργία του προγράμματος στο Arduino.



Εικόνα 7.2: Έλεγχος λειτουργίας κινητήρα DC

Δραστηριότητες

1. Κάντε τις απαραίτητες αλλαγές στο πρόγραμμα, έτσι ώστε τα κουμπιά να κινούν το φορείο προς τις αντίθετες κατευθύνσεις από ό,τι αρχικά.
 2. Δοκιμάστε να μειώσετε την ταχύτητα κίνησης του φορείου. Ποια είναι η ελάχιστη ταχύτητα που μπορείτε να φτάσετε;
-
-

7.2 Έλεγχος βηματικού κινητήρα

Στην εργασία αυτή θα επεκτείνετε το κύκλωμα της προηγούμενης εργασίας προσθέτοντας έναν βηματικό κινητήρα (Stepper Motor) ο οποίος θα ελέγχεται επίσης από δύο κουμπιά. Το ένα κουμπί θα κάνει μια πλήρη περιστροφή του βηματικού κινητήρα προς τα δεξιά, ενώ το άλλο μια πλήρη περιστροφή προς τα αριστερά.

Ταυτόχρονα, θα κινείται και το φορείο με διαφορετικό όμως τρόπο από την προηγούμενη εργασία, για λόγους καλύτερης επίδειξης. Πλέον, με το πάτημα του ενός κουμπιού, το φορείο θα κινείται προς την αντίστοιχη κατεύθυνση για ένα μικρό κλάσμα του δευτερολέπτου (π.χ. 0,7 sec), μετά θα σταματά και κατόπιν θα αρχίσει να περιστρέφεται ο βηματικός κινητήρας.

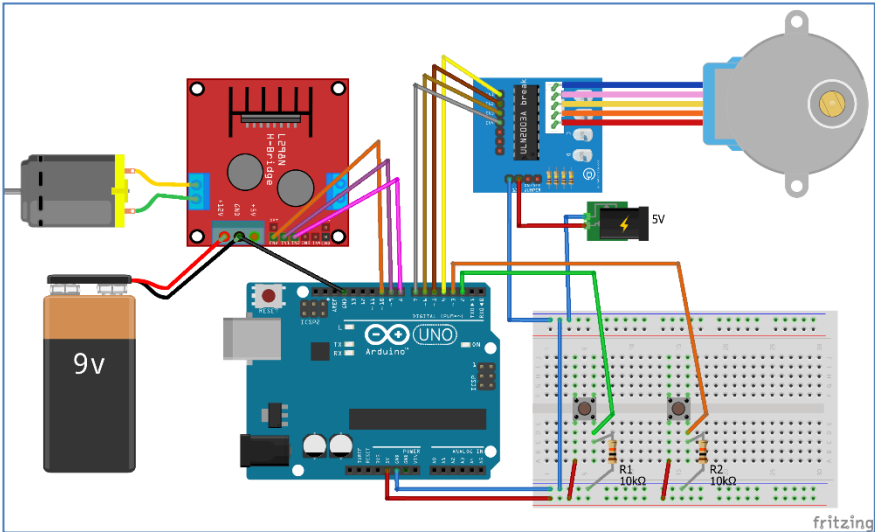
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε επιπλέον τα παρακάτω υλικά:

<p>1 Βηματικός κινητήρας (stepper motor, π.χ. 28BYJ-48)</p>	
<p>1 Πλακέτα οδήγησης του βηματικού κινητήρα (Stepper Driver Breakout - ULN2003)</p>	

Διάταξη κυκλώματος

Στο κύκλωμα της προηγούμενης εργασίας προσθέστε τον οδηγό του βηματικού κινητήρα σύμφωνα με την συνδεσμολογία της εικόνας 7.3.



Εικόνα 7.3: Κύκλωμα ελέγχου βηματικού κινητήρα

Pin Οδηγού	Σύνδεση
VDD	5V (από εξωτερική τροφοδοσία)
IN1	Θύρα 4 στο Arduino
IN2	Θύρα 5 στο Arduino
IN3	Θύρα 6 στο Arduino
IN4	Θύρα 7 στο Arduino
GND	Γείωση

Κώδικας

Συμπληρώστε το πρόγραμμα με τις νέες εντολές, για να προγραμματίσετε το παραπάνω κύκλωμα, έτσι ώστε να ελέγχετε τον βηματικό κινητήρα.

```

/* Βηματικός κινητήρας */

#include <Stepper.h>

// Το stepper motor 28BYJ-48 κάνει 2048 βήματα σε μια
// περιστροφή (επειδή σε Full Step mode κάνει
// 32 steps per revolution * 64 Gear Ratio = 2048)
const int stepsPerRevolution = 2048;

// Συνδεσμολογία Arduino->ULN2003 Board:
// Pin4->IN1, Pin5->IN2, Pin6->IN3, Pin7->IN4
Stepper myStepper = Stepper(stepsPerRevolution,4,6,5,7);
// Προσοχή στη σειρά!

// Χρησιμοποιούμε σταθερές για τους αριθμούς θυρών:
const int buttonPin1 = 2;
const int buttonPin2 = 3;

// Χρησιμοποιούμε μεταβλητές που αποθηκεύουν την κατάσταση
// των κουμπιών. LOW: όχι πατημένο, HIGH: πατημένο
int buttonState1 = LOW;
int buttonState2 = LOW;

// Οι συνδέσεις των pins του Arduino με τα pins
// της πλακέτας ελέγχου του μοτέρ (L298N H-Bridge)
const int enA = 10;
const int in1 = 9;
const int in2 = 8;

void setup()
{
  // Ορισμός της ταχύτητας του stepper σε 5 rpm
  myStepper.setSpeed(5);

  // Ορισμός pins εισόδου
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
}

```

```
// Ορισμός pins εξόδου
pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
}

void loop()
{
  // Διαβάζουμε τη θύρα κάθε κουμπιού και αποθηκεύουμε
  // την κατάσταση του:
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);

  // Αν δεν είναι πατημένο κανένα κουμπί:
  if (buttonState1 == LOW && buttonState2 == LOW)
  {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    analogWrite(enA, 0); // Ορισμός της ταχύτητας στο 0
  }
  else if (buttonState1 == HIGH)
  // Αλλιώς αν είναι πατημένο το 1ο κουμπί:
  {
    // Πήγαινε το φορείο προς τα δεξιά για 0.7 sec
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    // Ορισμός της ταχύτητας στο εύρος 0~255
    analogWrite(enA, 200);
    delay(700);
    // και μετά stop
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    analogWrite(enA, 0);

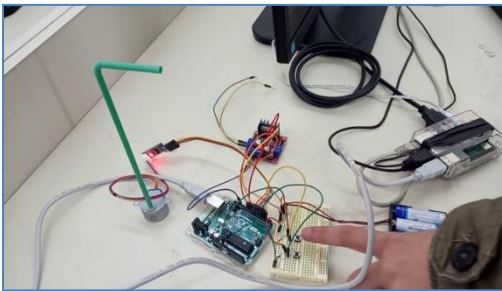
    // Κάνε μια πλήρη περιστροφή του stepper motor
    // προς τη μια κατεύθυνση
    myStepper.step(stepsPerRevolution);
    delay(500);
  }
  else if (buttonState2 == HIGH)
  // Αλλιώς αν είναι πατημένο το 2ο κουμπί:
  {
    // Πήγαινε το φορείο προς τα αριστερά για 0.7 sec
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    // Ορισμός της ταχύτητας στο εύρος 0~255
    analogWrite(enA, 200);
    delay(700);
  }
}
```

```
// και μετά stop
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
analogWrite(enA, 0);

// Κάνε μια πλήρη περιστροφή του stepper motor
// προς την άλλη κατεύθυνση
myStepper.step(-stepsPerRevolution);
delay(500);
}
}
```



Ελέγξτε τη λειτουργία του προγράμματος στο Arduino.



Εικόνα 7.4:
Έλεγχος
συναρμολόγησης
κυκλώματος
ελέγχου βηματικού
κινητήρα

Δραστηριότητες




1. Αυξήστε την ταχύτητα περιστροφής του βηματικού κινητήρα και παρατηρείστε τη συμπεριφορά του. Ποια είναι η μέγιστη ταχύτητα περιστροφής που μπορείτε να φτάσετε;
2. Προσθέστε στο breadboard ένα ποτενσιόμετρο (μεταβλητή αντίσταση) και συνδέστε το σε μια αναλογική είσοδο του Arduino. Προσθέστε στο πρόγραμμα τη δυνατότητα να διαβάζει την τιμή του ποτενσιόμετρου και να ρυθμίζει ανάλογα την ταχύτητα περιστροφής του βηματικού κινητήρα.

7.3 Επαγωγικός αισθητήρας

Στην εργασία αυτή θα προσθέσουμε έναν επαγωγικό αισθητήρα (inductive sensor), ο οποίος ανιχνεύει μεταλλικά αντικείμενα. Για τον έλεγχο της λειτουργίας του αισθητήρα θα χρησιμοποιήσουμε το ενσωματωμένο LED (Built-in LED) του Arduino (θύρα 13).

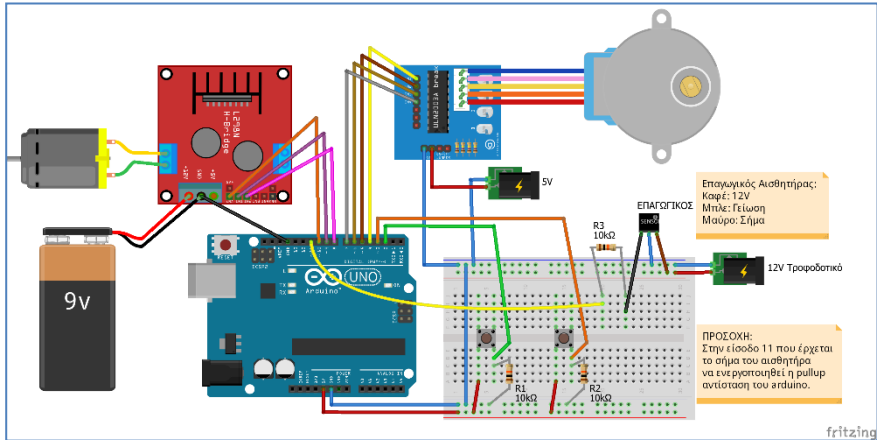
Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε επιπλέον τα παρακάτω υλικά:

1 Επαγωγικός αισθητήρας (LJ18A3-8-Z/AX)	
1 Αντίσταση 10kΩ	
1 Τροφοδοτικό 12V για τον αισθητήρα	

Διάταξη κυκλώματος

Στο κύκλωμα της προηγούμενης εργασίας προσθέστε τον επαγωγικό αισθητήρα σύμφωνα με τη συνδεσμολογία της εικόνας 7.5.



Εικόνα 7.5: Κύκλωμα ελέγχου επαγωγικού αισθητήρα

Καλώδιο αισθητήρα	Σύνδεση
Μπλε	12V (από εξωτερική τροφοδοσία)
Καφέ	Γείωση
Μαύρο	Σήμα (στη θύρα εισόδου 11 στο Arduino, με ενεργοποιημένη την εσωτερική pullup αντίσταση)

Κώδικας

Συμπληρώστε το πρόγραμμα με τις νέες εντολές, για να προγραμματίσετε το παραπάνω κύκλωμα, έτσι ώστε να ελέγχεται ο επαγωγικός αισθητήρας.

```
/* Επαγωγικός αισθητήρας */
#include <Stepper.h>

// Το stepper motor 28BYJ-48 κάνει 2048 βήματα σε μια
// περιστροφή (επειδή σε Full Step mode κάνει
// 32 steps per revolution * 64 Gear Ratio = 2048)
const int stepsPerRevolution = 2048;

// Συνδεσμολογία Arduino->ULN2003 Board:
// Pin4->IN1, Pin5->IN2, Pin6->IN3, Pin7->IN4
Stepper myStepper = Stepper(stepsPerRevolution, 4, 6, 5, 7);
// Προσοχή στη σειρά!

// Χρησιμοποιούμε σταθερές για τους αριθμούς θυρών:
const int buttonPin1 = 2;
const int buttonPin2 = 3;

// Χρησιμοποιούμε μεταβλητές που αποθηκεύουν την
// κατάσταση των κουμπιών:
// LOW: όχι πατημένο, HIGH: πατημένο
int buttonState1 = LOW;
int buttonState2 = LOW;

// Οι συνδέσεις των pins του Arduino με τα pins
// της πλακέτας ελέγχου του μοτέρ (L298N H-Bridge)
const int enA = 10;
const int in1 = 9;
const int in2 = 8;

const int ledPin = 13; // Το ενσωματωμένο led
const int sensorPin = 11; // Η θύρα του αισθητήρα
int sensorVal; // Το σήμα του αισθητήρα
void setup()
{
  // Ορισμός της ταχύτητας του stepper σε 5 rpm
  myStepper.setSpeed(5);
  // Ορισμός pins εισόδου
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);
```



```

// Ορισμός pins εξόδου
pinMode(enA, OUTPUT);
pinMode(in1, OUTPUT);
pinMode(in2, OUTPUT);
pinMode(ledPin, OUTPUT);

// Pin εισόδου με ενεργοποιημένη την εσωτερική
// pullup αντίσταση
pinMode(sensorPin, INPUT_PULLUP);
}

void loop()
{
// Διαβάζουμε τον επαγωγικό αισθητήρα
sensorVal = digitalRead(sensorPin);
// Ο επαγωγικός αισθητήρας δίνει LOW αν
// ανιχνεύσει μέταλλο, αλλιώς HIGH
if( sensorVal == HIGH )
{
digitalWrite(ledPin, LOW);
}
else
{
digitalWrite(ledPin, HIGH);
}

// Διαβάζουμε τη θύρα κάθε κουμπιού και αποθηκεύουμε
// την κατάσταση του:
buttonState1 = digitalRead(buttonPin1);
buttonState2 = digitalRead(buttonPin2);
// Αν δεν είναι πατημένο κανένα κουμπί:
if (buttonState1 == LOW && buttonState2 == LOW)
{
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
analogWrite(enA, 0); // Ορισμός της ταχύτητας στο 0
}
else if (buttonState1 == HIGH)
// Αλλιώς αν είναι πατημένο το 1ο κουμπί:
{
// Πήγαινε το φορείο προς τα δεξιά για 0.7 sec
digitalWrite(in1, HIGH);
digitalWrite(in2, LOW);
// Ορισμός της ταχύτητας από το εύρος 0~255
analogWrite(enA, 200);
delay(700);
// και μετά stop
digitalWrite(in1, LOW);

```

```
digitalWrite(in2, LOW);
analogWrite(enA, 0);

// Κάνε μια πλήρη περιστροφή του stepper motor
// προς τη μια κατεύθυνση
myStepper.step(stepsPerRevolution);
delay(500);
}
else if (buttonState2 == HIGH)
// Αλλιώς αν είναι πατημένο το 2ο κουμπί:
{
// Πήγαινε το φορείο προς τα αριστερά για 0.7 sec
digitalWrite(in1, LOW);
digitalWrite(in2, HIGH);
// Ορισμός της ταχύτητας από το εύρος 0~255
analogWrite(enA, 200);
delay(700);
// και μετά stop
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
analogWrite(enA, 0);

// Κάνε μια πλήρη περιστροφή του stepper motor
// προς την άλλη κατεύθυνση
myStepper.step(-stepsPerRevolution);
delay(500);
}
delay(50);
}
```



Ελέγξτε τη λειτουργία του προγράμματος στο Arduino. Το ενσωματωμένο LED του Arduino πρέπει να ανάβει όταν πλησιάζουμε ένα μεταλλικό αντικείμενο.

Δραστηριότητα

Δοκιμάστε τον επαγωγικό αισθητήρα με διάφορα μέταλλα (αλουμίνιο, σίδηρο, ατσάλι κλπ.) και σημειώστε την απόσταση στην οποία ενεργοποιείται ο αισθητήρας σε κάθε περίπτωση.

7.4 Μικροδιακόπτες και όρια κίνησης

Στην εργασία αυτή θα προσθέσουμε δύο μικροδιακόπτες στα άκρα της διαδρομής του φορείου, έτσι ώστε να μπορούμε να ελέγχουμε την κίνησή του. Οι μικροδιακόπτες αυτοί πρέπει να στερεωθούν σε σημεία όπου θα πιέζονται από το ίδιο το φορείο, όταν φτάνει στο τέλος της διαδρομής του. Με τον τρόπο αυτό, το πρόγραμμα θα καταλαβαίνει ότι το φορείο έφτασε στο τέλος της διαδρομής, καθώς θα έχει πατηθεί ο αντίστοιχος μικροδιακόπτης.

Λίστα Υλικών

Για την ολοκλήρωση της εργασίας θα χρειαστείτε επιπλέον τα παρακάτω υλικά:

2 Μικροδιακόπτες	
Υλικά κόλλησης και κολλητήρι	
Καλώδια σύνδεσης (Jumper Wires)	

Συνδεσμολογία

Κολλήστε τα καλώδια στις κατάλληλες επαφές του μικροδιακόπτη. Προσέξτε το μήκος των καλωδίων βάζοντας τους μικροδιακόπτες στις κανονικές τους θέσεις, δηλαδή στα όρια κίνησης του φορείου. Ο μηχανισμός του εκτυπωτή και οι μικροδιακόπτες πρέπει να είναι σταθερά στερεωμένοι σε μια βάση (π.χ. ξύλινη), όπως φαίνεται στην εικόνα 7.6.



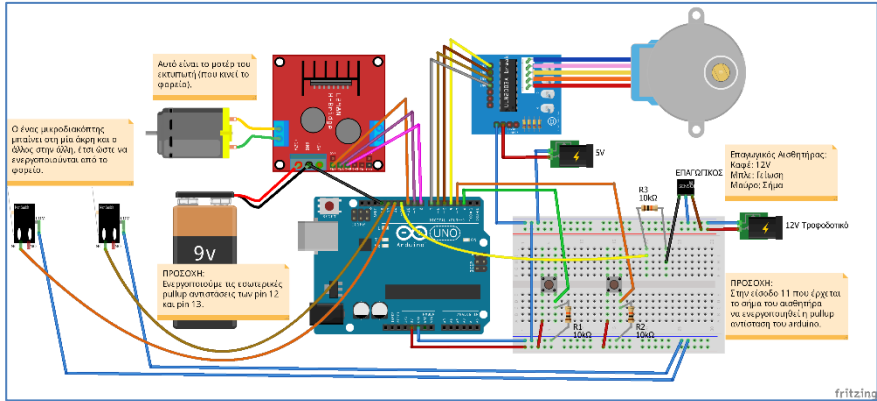
Εικόνα 7.6: Τοποθέτηση μηχανισμού εκτυπωτή και μικροδιακοπών

Επαφή Μικροδιακόπτη	Σύνδεση
Common	GND (Γείωση)
Normally Open (NO)	Ασύνδετο
Normally Closed (NC)	Είσοδος του Arduino (με ενεργοποιημένη την εσωτερική pullup αντίσταση)

Σημείωση: Στο κύκλωμα αυτό θα χρησιμοποιήσουμε τις επαφές COM και NC με την ορθή λογική, έτσι ώστε με πατημένο το διακόπτη να διαβάζουμε HIGH στην είσοδο του Arduino. Υπάρχει και η κλασική συνδεσμολογία που χρησιμοποιεί το COM και το NO με αντίστροφη λογική.

Διάταξη κυκλώματος

Στο προηγούμενο κύκλωμα προσθέστε τους 2 μικροδιακόπτες υλοποιώντας το παρακάτω κύκλωμα:



Εικόνα 7.7: Κύκλωμα με την πρόσθεση μικροδιακοπών

Κώδικας

Για να ελέγξετε τη σωστή λειτουργία των μικροδιακοπών, υλοποιείτε πρόγραμμα, το οποίο θα πρέπει να κάνει τα εξής:

- Αν πατηθεί το ένα κουμπί (push button), το φορέα θα πρέπει να μετακινείται μέχρι τη μία άκρη (δηλαδή, μέχρι να πατήσει τον αντίστοιχο μικροδιακόπτη που βρίσκεται εκεί) και κατόπιν θα σταματάει.
- Παρομοίως, αν πατηθεί το άλλο κουμπί (push button), θα μετακινείται στην άλλη άκρη, μέχρι να πατήσει τον άλλο μικροδιακόπτη, και μετά θα σταματάει.

Σε αυτή την εφαρμογή χρησιμοποιούμε τα δύο push buttons, τους δύο μικροδιακόπτες καθώς και το μηχανισμό κίνησης του φορείου μαζί με την πλακέτα ελέγχου του μοτέρ του εκτυπωτή και αγνοούμε τον βηματικό κινητήρα και τον αισθητήρα μετάλλων. Όμοια, έχει αφαιρεθεί και ο αντίστοιχος κώδικας.

Προσοχή πρέπει να δοθεί στην κατεύθυνση κίνησης του φορείου, η οποία εξαρτάται από τον τρόπο σύνδεσης των δύο καλωδίων του κινητήρα στην πλακέτα ελέγχου. Αντιστρέφοντας τα δύο αυτά καλώδια, το φορείο θα κινείται προς την αντίθετη κατεύθυνση. Στην περίπτωση αυτή, χρειάζεται ανάλογη τροποποίηση του προγράμματος, έτσι ώστε για το σταμάτημα του φορείου να ελέγχεται ο κατάλληλος μικροδιακόπτης που βρίσκεται στην κατεύθυνση κίνησής του. Διαφορετικά, αν δηλαδή το πρόγραμμα περιμένει να ενεργοποιηθεί ο μικροδιακόπτης που βρίσκεται στην αντίθετη άκρη από αυτή που πηγαίνει το φορείο, τότε θα πέσει σε ατέρμονα βρόχο.

```
/* Μικροδιακόπτες */  
  
// Χρησιμοποιούμε σταθερές για τους αριθμούς θυρών:  
const int buttonPin1 = 2;  
const int buttonPin2 = 3;  
  
// Χρησιμοποιούμε μεταβλητές που αποθηκεύουν την κατάσταση  
// των κουμπιών. LOW: όχι πατημένο, HIGH: πατημένο  
int buttonState1 = LOW;  
int buttonState2 = LOW;  
  
// Οι συνδέσεις των pins του Arduino με τα pins  
// της πλακέτας ελέγχου του μοτέρ (L298N H-Bridge)  
const int enA = 10;  
const int in1 = 9;  
const int in2 = 8;  
  
// Τα pins που συνδέονται οι 2 μικροδιακόπτες  
const int switch1Pin = 12;
```

```

const int switch2Pin = 13;

void setup()
{
  // Ορισμός pins εισόδου
  pinMode(buttonPin1, INPUT);
  pinMode(buttonPin2, INPUT);

  // Ορισμός pins εξόδου
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);

  // Ορισμός pins εισόδου με ενεργοποιημένη
  // την εσωτερική pullup αντίσταση
  pinMode(switch1Pin, INPUT_PULLUP);
  pinMode(switch2Pin, INPUT_PULLUP);
}

void loop()
{
  // Διαβάζουμε τη θύρα του κουμπιού και αποθηκεύουμε
  // την κατάσταση του:
  buttonState1 = digitalRead(buttonPin1);
  buttonState2 = digitalRead(buttonPin2);

  // Αν είναι πατημένο το 1ο κουμπί:
  if( buttonState1 == HIGH )
  {
    // Εκκίνα το φορείο προς τη μία πλευρά...
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    // Ορισμός της ταχύτητας από 0~255
    analogWrite(enA,200);
    // Περίμενε μέχρι να φτάσει στην άκρη...
    while( digitalRead(switch2Pin) == LOW )
      delay(1);

    // και μετά stop
    digitalWrite(in1,LOW);
    digitalWrite(in2,LOW);
    analogWrite(enA,0);
  }
  else if( buttonState2 == HIGH )
  // Αλλιώς αν είναι πατημένο το 2ο κουμπί:
  {
    // Εκκίνα το φορείο προς την άλλη πλευρά...
    digitalWrite(in1,LOW);
  }
}

```

```
digitalWrite(in2,HIGH);  
// Ορισμός της ταχύτητας από 0~255  
analogWrite(enA,200);  
// Περίμενε μέχρι να φτάσει στην άκρη...  
while( digitalRead(switch1Pin) == LOW )  
    delay(1);  
  
    // και μετά stop  
    digitalWrite(in1,LOW);  
    digitalWrite(in2,LOW);  
    analogWrite(enA,0);  
}  
delay(10);  
}
```



Ελέγξτε τη λειτουργία του προγράμματος στο Arduino.

Δραστηριότητες

1. Γράψτε μια δική σας συνάρτηση που να σταματά το φορείο. Ονομάστε την **“stop”** και ορίστε την έτσι ώστε να μη δέχεται παραμέτρους και να μην επιστρέφει καμία τιμή. Χρησιμοποιήστε την στο πρόγραμμα, καλώντας την από τα κατάλληλα σημεία.
2. Γράψτε μια νέα συνάρτηση **“move”** που να χρησιμοποιείται για την κίνηση του φορείου. Να δέχεται 2 παραμέτρους: την κατεύθυνση κίνησης και την ταχύτητα.

7.5 Ολοκλήρωση του διαχωριστή μετάλλων

Η τελευταία φάση του έργου περιλαμβάνει την κατασκευή, τη συναρμολόγηση και τον τελικό προγραμματισμό του διαχωριστή μεταλλικών αντικειμένων.

Το σενάριο του project είναι η κατασκευή μηχανισμού που θα διαχωρίζει μεταλλικά από μη μεταλλικά αντικείμενα. Πάνω στο κινούμενο φορείο πρέπει να βρίσκεται ένας ανατρεπόμενος κάδος στηριγμένος σε κατάλληλη βάση, τον οποίο μπορείτε να εκτυπώσετε σε έναν 3D εκτυπωτή (οδηγίες στο βιβλίο «Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο»). Ο κάδος θα μπορεί να περιστρέφεται από έναν βηματικό κινητήρα είτε προς τη μία είτε προς την άλλη φορά, ανατρέποντας το περιεχόμενό του στη μία ή στην άλλη πλευρά του φορείου.

Η απόφαση της κατεύθυνσης περιστροφής πρέπει να λαμβάνεται ανάλογα με το περιεχόμενο του κάδου, αφού έχει εξεταστεί από τον αισθητήρα μετάλλων. Ο αισθητήρας μετάλλων πρέπει να είναι στερεωμένος σε μικρή απόσταση από την αρχική θέση του φορείου.

Η διαδικασία θα είναι η εξής:

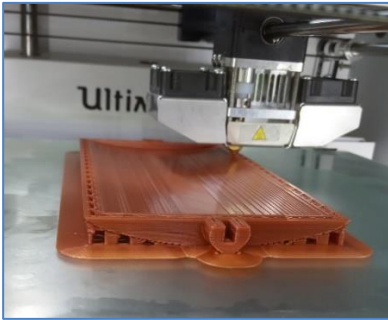
- Αρχικά, το φορείο θα μετακινείται στη μία άκρη της βάσης και θα περιμένει.
- Ο χειριστής θα φορτώνει ένα αντικείμενο στον κάδο και κατόπιν θα πατάει το κουμπί έναρξης.
- Ο αισθητήρας θα ανιχνεύει το αντικείμενο και κατόπιν, το φορείο θα μετακινείται μέχρι την άλλη άκρη, όπου θα γίνεται η ανατροπή είτε προς τα δεξιά είτε προς τα αριστερά, ανάλογα με το είδος του αντικειμένου (μεταλλικό ή όχι).

- Τέλος, το φορείο επιστρέφει στην αρχική του θέση, για να επαναληφθεί η διαδικασία.

Συναρμολόγηση

Στην προηγούμενη εργασία ο μηχανισμός του εκτυπωτή είχε στερεωθεί πάνω σε μια ξύλινη βάση και οι δύο μικροδιακόπτες είχαν τοποθετηθεί σταθερά στις κατάλληλες θέσεις.

Για την εργασία αυτή θα πρέπει να κατασκευαστούν ο κάδος και η βάση του σε έναν 3D εκτυπωτή. Στην εικόνα 7.8 φαίνεται η εκτύπωση του κάδου.



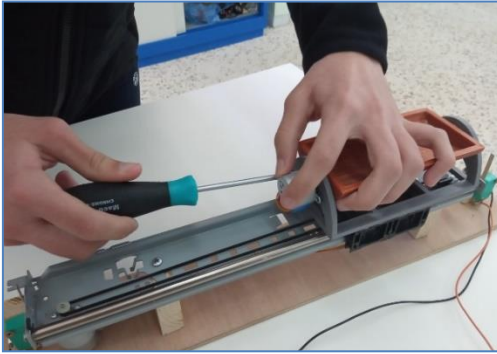
Εικόνα 7.8:
Εκτύπωση κάδου
στον 3D εκτυπωτή

Η βάση του κάδου πρέπει να κολληθεί πάνω στην κεφαλή εκτύπωσης, όπως φαίνεται στην εικόνα 7.9.



Εικόνα 7.9:
Συγκόλληση βάσης
κάδου στην κεφαλή
εκτύπωσης

Στη συνέχεια τοποθετείται ο κάδος πάνω στη βάση και τέλος, προσαρμόζεται και βιδώνεται ο βηματικός κινητήρας, όπως φαίνεται στην εικόνα 7.10. Ο βηματικός κινητήρας πρέπει να μπορεί να περιστρέφει τον κάδο δεξιόστροφα και αριστερόστροφα.



Εικόνα 7.10:
*Τοποθέτηση κάδου
στη βάση και
προσαρμογή
Stepper motor*

Ο αισθητήρας μετάλλων πρέπει να στερεωθεί πάνω σε μια βάση, έτσι ώστε να βρίσκεται σε αρκετά κοντινή απόσταση από τον κάδο στην αρχική του θέση. Η απόσταση πρέπει να επιτρέπει τη σωστή ανίχνευση του αντικειμένου που φορτώνεται στον κάδο. Στην εικόνα 7.11 φαίνεται ο αισθητήρας τοποθετημένος σε μια βάση που σχεδιάστηκε σε έναν 3D εκτυπωτή.



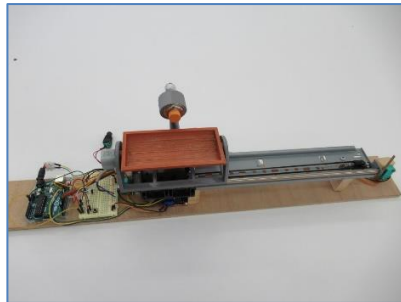
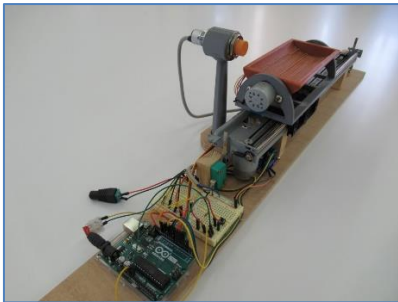
Εικόνα 7.11:
*Προσαρμογή
αισθητήρα κοντά
στον κάδο*

Στην εικόνα 7.12 φαίνεται το τέλος της διαδρομής του φορείου, όπου η βάση του κάδου πατάει το έλασμα του μικροδιακόπτη.



Εικόνα 7.12: Το τέλος της διαδρομής του φορείου

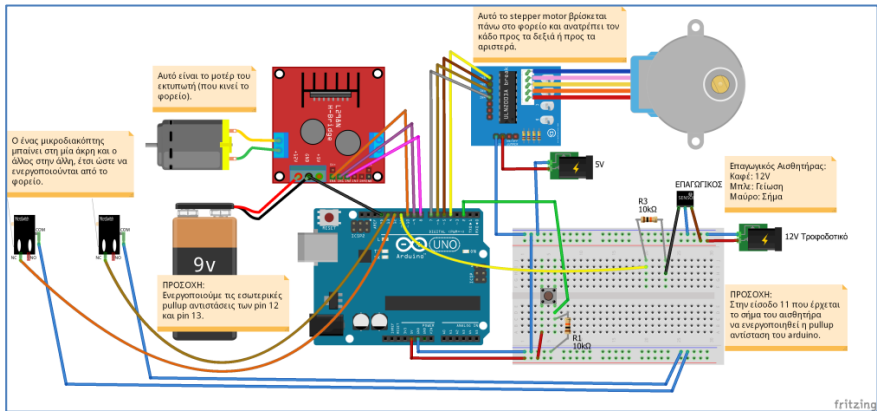
Η ολοκληρωμένη κατασκευή του διαχωριστή μετάλλων φαίνεται στις εικόνες 7.13 και 7.14.



Εικόνα 7.13 & Εικόνα 7.14:
Ολοκληρωμένη κατασκευή του διαχωριστή μετάλλων

Διάταξη κυκλώματος

Στο κύκλωμα της προηγούμενης εργασίας αφαιρέστε το ένα push button. Το άλλο push button που μένει θα λειτουργεί ως το κουμπί έναρξης.



Εικόνα 7.15. Ολοκληρωμένο κύκλωμα διαχωριστή μεταλλικών αντικειμένων

Κώδικας

Ολοκληρώστε το πρόγραμμα του διαχωριστή μετάλλων όπως φαίνεται παρακάτω.

```
/* Διαχωριστής Μετάλλων */
#include <Stepper.h>

// Τα βήματα που κάνει ο βηματικός κινητήρας σε μια πλήρη
// περιστροφή
const int stepsPerRevolution = 2048;

Stepper myStepper = Stepper(stepsPerRevolution,4,6,5,7);
```

ΕΦΑΡΜΟΓΕΣ ARDUINO στο σχολικό εργαστήριο για το περιβάλλον

```
// Τα βήματα περιστροφής του βηματικού κινητήρα που
// απαιτούνται για την ανατροπή του κάδου
// Η γωνία περιστροφής θα είναι  $350 \cdot 360 / 2048 = 61.5$  μοίρες
const int anatropi = 350;

// Οι συνδέσεις του motor driver
const int enA = 10;
const int in1 = 9;
const int in2 = 8;

// Η ταχύτητα του DC motor (από 0 ως 255)
const int motorSpeed = 140;
const int buttonPin = 2; // Το κουμπί έναρξης
const int sensorPin = 11; // Ο αισθητήρας μετάλλων
const int switch1Pin = 12; // Οι 2 μικροδιακόπτες
const int switch2Pin = 13;

void setup()
{
  // Η ταχύτητα του stepper σε περιστροφές ανά λεπτό (rpm)
  myStepper.setSpeed(5);

  // Pins εισόδου
  pinMode(buttonPin, INPUT);
  pinMode(sensorPin, INPUT_PULLUP);
  pinMode(switch1Pin, INPUT_PULLUP);
  pinMode(switch2Pin, INPUT_PULLUP);

  // Pins εξόδου
  pinMode(enA, OUTPUT);
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);

  // Μετακίνηση του φορείου στην αρχική θέση
  digitalWrite(in1, HIGH);
  digitalWrite(in2, LOW);
  analogWrite(enA, motorSpeed);

  // Αναμονή μέχρι να φτάσει...
  while( digitalRead(switch1Pin) == LOW )
    delay(1);

  // stop
  digitalWrite(in1, LOW);
  digitalWrite(in2, LOW);
  analogWrite(enA, 0);
}
```

```

void loop()
{
    // Έλεγχος του κουμπιού έναρξης
    int buttonState = digitalRead(buttonPin);

    // Αν πατήθηκε...
    if( buttonState == HIGH )
    {
        // Διάβασμα του αισθητήρα μετάλλων και αποθήκευση
        // της τιμής του
        int sensorVal = digitalRead(sensorPin);
        delay(200);

        // Πήγαινε το φορείο μέχρι την άλλη άκρη
        digitalWrite(in1, LOW);
        digitalWrite(in2, HIGH);
        analogWrite(enA, motorSpeed);

        // Αναμονή μέχρι να φτάσει...
        while( digitalRead(switch2Pin) == LOW )
            delay(1);

        // stop
        digitalWrite(in1, LOW);
        digitalWrite(in2, LOW);
        analogWrite(enA, 0);

        if( sensorVal == HIGH ) // Αν βρέθηκε μέταλλο...
        {
            // Ανατροπή δεξιά και επαναφορά
            myStepper.step(anatropi);
            delay(500);
            myStepper.step(-anatropi);
        }

        else // Αν το αντικείμενο δεν είναι μέταλλο...
        {
            // Ανατροπή αριστερά και επαναφορά
            myStepper.step(-anatropi);
            delay(500);
            myStepper.step(anatropi);
        }

        // Επιστροφή του φορείου στην αρχική θέση
        digitalWrite(in1, HIGH);
        digitalWrite(in2, LOW);
        analogWrite(enA, motorSpeed);
    }
}

```

```
// Αναμονή μέχρι να φτάσει...
while( digitalRead(switch1Pin) == LOW )
    delay(1);

// stop
digitalWrite(in1, LOW);
digitalWrite(in2, LOW);
analogWrite(enA, 0);
}

delay(50);
}
```



Ελέγξτε τη λειτουργία του προγράμματος στο Arduino.

Δραστηριότητα

1. Αλλάξτε το σημείο όπου αδειάζει ο κάδος τα μέταλλα: καθορίστε την αντίθετη πλευρά του κάδου. Κάντε το ίδιο για τα μη μεταλλικά αντικείμενα.
2. Γράψτε μια νέα συνάρτηση “**rotate**” για την ανατροπή του κάδου. Να δέχεται μία παράμετρο που να καθορίζει τη φορά της περιστροφής (αριστερά ή δεξιά).

Λέξεις κλειδιά

Μικροδιακόπτης

Βηματικός κινητήρας

Κινητήρας DC

Πλακέτα οδήγησης

Πλακέτα ελέγχου

Ποτενσιόμετρο

Ελεγκτής μοτέρ

Επαγωγικός αισθητήρας

Ταχύτητα κίνησης

Ανίχνευση μετάλλων

8. Παρακολούθηση ποιότητας αέρα

Στην εργασία αυτή θα υλοποιήσουμε ουσιαστικά δύο συστήματα. Ένα για την ανίχνευση της ποιότητας του αέρα με τη μέτρηση βασικών περιβαλλοντικών στοιχείων και ένα για την αποστολή των δεδομένων της μέτρησης στο διαδίκτυο, σε έναν ιστότοπο συλλογής δεδομένων, όπου τα στοιχεία θα είναι προσβάσιμα από οποιονδήποτε.

Για τη μέτρηση της ποιότητας του αέρα θα χρησιμοποιήσουμε έναν αισθητήρα αερίων, όπως είναι ο SparkFun CCS811, ο οποίος σχεδιάστηκε για μετρήσεις σε εσωτερικούς χώρους και έχει τη δυνατότητα να ανιχνεύει έναν μεγάλο αριθμό πτητικών οργανικών ουσιών (Total Volatile Organic Compounds - tVOCs), καθώς και το ισοδύναμο του CO₂ (eCO₂).

Για τη μέτρηση θερμοκρασίας και υγρασίας θα χρησιμοποιήσουμε έναν αισθητήρα DHT11.

Οι μετρήσεις των αισθητήρων θα εμφανίζονται τοπικά σε μια οθόνη LCD και παράλληλα, θα στέλνονται στο νέφος (cloud), υλοποιώντας την τεχνολογία **IoT (Internet of Things)**.

Για τη σύνδεση με το διαδίκτυο θα συνδέσουμε στο κύκλωμά μας ένα Wi-Fi module (κύκλωμα σύνδεσης σε ασύρματο δίκτυο), όπως είναι το ESP8266.

Επειδή το ESP8266 απαιτεί σταθερή τροφοδοσία με 3.3V, την οποία δεν μπορεί να προσφέρει το Arduino, θα χρησιμοποιήσουμε έναν ρυθμιστή τάσης (voltage regulator) 3.3V, όπως είναι ο LD1117.

Επίσης, θα προσθέσουμε έναν ανεμιστήρα που θα λειτουργεί αυτόματα, όταν η θερμοκρασία ξεπεράσει ένα όριο (π.χ. 26°C).

Τέλος, θα δημιουργήσουμε έναν δωρεάν λογαριασμό στον ιστότοπο ThingSpeak για τη συλλογή και επεξεργασία δεδομένων, που εξυπηρετεί την τεχνολογία IoT, δηλαδή τη δυνατότητα διαφόρων συσκευών να συνδέονται στο διαδίκτυο και να αποστέλλουν δεδομένα που διαβάζουν μέσω αισθητήρων.

Δημιουργία λογαριασμού στο ThingSpeak

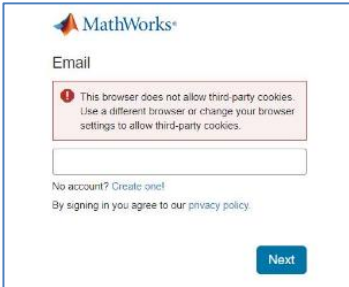
Ακολουθείστε τα παρακάτω βήματα για να δημιουργήσετε λογαριασμό στο ThingSpeak και στο Mathworks:

1. Μπείτε στο <https://thingspeak.com/> και πατήστε το κουμπί:



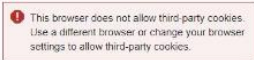
Get Started For Free

2. Για να προχωρήσετε, απαιτείται να δημιουργήσετε λογαριασμό και στο MathWorks, αν δεν έχετε ήδη κάποιον. Για να δημιουργήσετε ένα λογαριασμό, πατήστε στο "Create One".
ΠΡΟΣΟΧΗ: Ο φυλλομετρητής που χρησιμοποιείτε πρέπει να επιτρέπει τα cookies τρίτων, διαφορετικά θα σας βγάλει το παρακάτω μήνυμα:



MathWorks®

Email



No account? [Create one!](#)

By signing in you agree to our [privacy policy](#).

Next

3. Στο επόμενο βήμα συμπληρώστε την παρακάτω φόρμα, που απαιτεί διεύθυνση e-mail:

Create MathWorks Account

Email Address

i To access your organization's MATLAB license, use your school or work email.

Location

First Name

Last Name

Continue

Cancel

4. Στη συνέχεια, θα σας ζητηθεί να επιβεβαιώσετε το e-mail που δηλώσατε, έτσι ώστε να μπορέσετε να ακολουθήσετε τα επόμενα βήματα που απαιτούνται για τη δημιουργία του λογαριασμού στο MathWorks.
5. Όταν ολοκληρωθεί η δημιουργία του λογαριασμού στο MathWorks, μπορείτε πλέον να μείετε στο ThingSpeak και να κάνετε σύνδεση (Sign In) χρησιμοποιώντας το e-mail που δηλώσατε για τη δημιουργία του λογαριασμού MathWorks, όπως φαίνεται στην παρακάτω εικόνα. Αμέσως μετά, θα σας ζητηθεί και ο αντίστοιχος κωδικός πρόσβασης (password).

MathWorks

Email

No account? [Create one!](#)

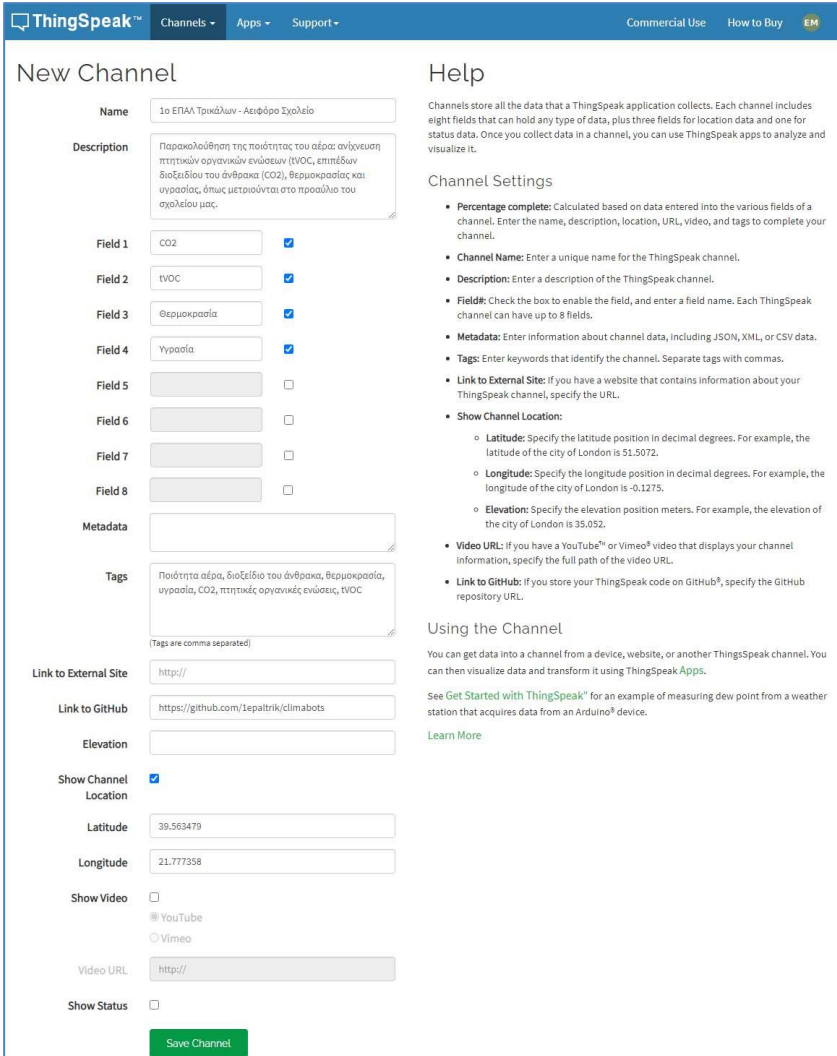
By signing in you agree to our [privacy policy](#).

Next

6. Στο ThingSpeak πρέπει να δημιουργήσετε ένα νέο κανάλι που θα λαμβάνει τα δεδομένα που στέλνετε. Πατήστε το κουμπί:

A green rectangular button with the text "New Channel" in white, centered within the button.

7. Συμπληρώστε τη φόρμα, όπως στην παρακάτω εικόνα:

A screenshot of the ThingSpeak 'New Channel' form. The form is titled 'New Channel' and is set against a blue header with 'ThingSpeak™' and navigation links. The form fields include: Name (1ο ΕΠΑΛ Τρικάλων - Αειφόρο Σχολείο), Description (Παρακολούθηση της ποιότητας του αέρα: ανίχνευση πτητικών οργανικών ενώσεων (tVOC, επιπέδων διοξειδίου του άνθρακα (CO2), θερμοκρασίας και υγρασίας, όπως μετρούνται στο πρασάλιο του σχολείου μας), Field 1 (CO2), Field 2 (tVOC), Field 3 (θερμοκρασία), Field 4 (Ύγρασία), Fields 5-8 (empty), Metadata (empty), Tags (Ποιότητα αέρα, διοξείδιο του άνθρακα, θερμοκρασία, υγρασία, CO2, πτητικές οργανικές ενώσεις, tVOC), Link to External Site (http://), Link to GitHub (https://github.com/lepatrik/climabots), Elevation (empty), Show Channel Location (checked), Latitude (39.563479), Longitude (21.777358), Show Video (radio buttons for YouTube and Vimeo), Video URL (http://), and Show Status (unchecked). A 'Save Channel' button is at the bottom. To the right, there is a 'Help' section with 'Channel Settings' and 'Using the Channel' subsections, each with a list of instructions.

8. Ορίστε 4 πεδία για να φαίνονται οι τιμές CO₂, tVOC, θερμοκρασίας και υγρασίας.

Field 1	<input type="text" value="CO2"/>	<input checked="" type="checkbox"/>
Field 2	<input type="text" value="tVOC"/>	<input checked="" type="checkbox"/>
Field 3	<input type="text" value="Θερμοκρασία"/>	<input checked="" type="checkbox"/>
Field 4	<input type="text" value="Υγρασία"/>	<input checked="" type="checkbox"/>

9. Καθορίστε τον τρόπο πρόσβασης στη σελίδα σας, όπως φαίνεται στην παρακάτω σελίδα που φτιάξαμε «1^ο ΕΠΑΛ Τρικάλων – Αειφόρο Σχολείο».

1ο ΕΠΑΛ Τρικάλων - Αειφόρο Σχολείο

Channel ID: 1003587
 Author: mwa0000017527459
 Access: Public

Παρακολούθηση της ποιότητας του αέρα:
 Ανίχνευση πτητικών οργανικών ενώσεων (tVOC) και επιπέδων διοξειδίου του άνθρακα (CO₂), όπως επιρούνται στο προαύλιο του σχολείου μας.
 ποιότητα αέρα, διοξείδιο του άνθρακα, co₂, πτητικές οργανικές ενώσεις, tvoc

Private View Public View Channel Settings **Sharing** API Keys Data Import / Export

Channel Sharing Settings

Keep channel view private
 Share channel view with everyone
 Share channel view only with the following users:

Email Address

Help

ThingSpeak allows you to control who can view the data in your channel. Irrespective of the settings on this tab, reading data from or writing data to the fields of a channel requires the appropriate API key for the channel.

Channel Sharing Settings

- **Keep channel view private:** Selecting this option keeps your channel private. Only you will be able to see the channel view.
- **Share channel view with everyone:** Selecting this option makes the public view of your channel viewable by anyone browsing the ThingSpeak website.
- **Share channel view only with the following users:** Selecting this option shares the private view of your channel only with specific ThingSpeak users.







Εικόνα 8.1: Σελίδα «1^ο ΕΠΑΛ Τρικάλων – Αειφόρο Σχολείο» στο ThingSpeak

10. Στην καρτέλα “API Keys” φαίνονται οι κωδικοί πρόσβασης. Ο κωδικός “Write API Key” είναι απαραίτητος και θα χρησιμοποιηθεί για την αποστολή των δεδομένων σας.
11. Στην καρτέλα “Public View” καθορίζεται ο τρόπος εμφάνισης των δεδομένων. Για κάθε πεδίο πατήστε στο εικονίδιο με το μολύβι για να δείτε τις επιλογές εμφάνισης των γραφημάτων.

Λίστα Υλικών

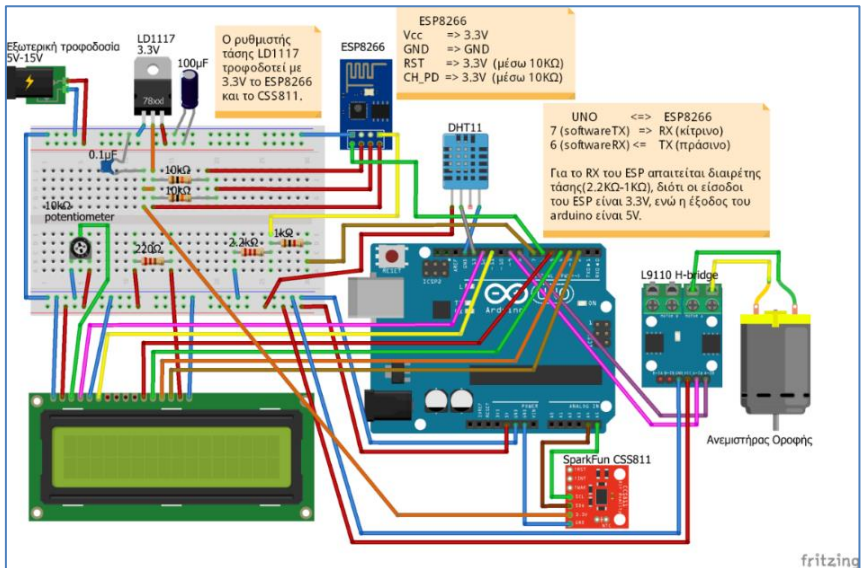
Για την ολοκλήρωση της εργασίας θα χρειαστείτε τα παρακάτω υλικά:

1 Arduino Uno		1 Breadboard	
1 Πλακέτα ESP8266 Wi-Fi Module		1 Σταθεροποιητής τάσης LD1117 – 3.3V 800mA	
1 Αισθητήρας ποιότητας αέρα CCS811		1 Οθόνη LCD 16x2 χαρακτήρων	
1 Αισθητήρας θερμοκρασίας-υγρασίας DHT11		Καλώδια σύνδεσης (Jumper Wires)	
1 Οδηγός κινητήρα L9110 (Dual Motor Driver Module)		1 Κινητήρας DC 5V για τον ανεμιστήρα	
1 Προπέλα για τον ανεμιστήρα		1 Μεταβλητή αντίσταση 10kΩ (ποτενσιόμετρο)	

2 Αντιστάσεις 10kΩ		1 Αντίσταση 2.2kΩ	
1 Αντίσταση 1kΩ		1 Αντίσταση 220Ω	
1 Πυκνωτής 100μF (προαιρετικά)		1 Πυκνωτής 0.1μF (προαιρετικά)	

Διάταξη κυκλώματος

Συνδέστε τα υλικά που συγκεντρώσατε, υλοποιώντας το κύκλωμα της Εικόνας 8.2.



Εικόνα 8.2: Κύκλωμα για την παρακολούθηση ποιότητας αέρα

Κώδικας

Στο Arduino IDE επιλέξτε «Εργαλεία-Διαχείριση Βιβλιοθηκών» για να εγκαταστήσετε τις απαραίτητες βιβλιοθήκες, αν δεν είναι ήδη εγκατεστημένες.

Βιβλιοθήκη	Ονομασία
SparkFunCCS811.h	SparkFun CCS811 Arduino Library
DHT.h	DHT sensor library (by Adafruit)
WiFiEsp.h	WiFiEsp (by bportaluri)
SoftwareSerial.h	Είναι προεγκατεστημένη

Προγραμματίστε το Arduino, έτσι ώστε να συνδέεται στο δίκτυο Wi-Fi της περιοχής εγκατάστασης, να διαβάζει από τους αισθητήρες τις τιμές CO₂, tVOC, θερμοκρασία και υγρασία, και κατόπιν να στέλνει αυτές τις τιμές στην ιστοσελίδα <https://thingspeak.com/>.

```
/* Arduino IoT
 * 1) Μέτρηση τιμών θερμοκρασίας-υγρασίας και
 *    συγκέντρωσης αερίων CO2 - tVOC
 * 2) Εμφάνιση των τιμών σε οθόνη LCD
 * 3) Σύνδεση στο Wi-Fi και αποστολή των τιμών στο
 *    website https://thingspeak.com/
 * 4) Λειτουργία ανεμιστήρα όταν κάνει ζέστη.

 * Επειδή η μοναδική σειριακή θύρα του UNO
 * χρησιμοποιείται από το serial monitor, για τη σύνδεση
 * του Wi-Fi module με το Arduino απαιτείται προσομοίωση
 * σειριακής θύρας στα pin 6,7
 */

#ifndef HAVE_HWSERIAL1
#include "SoftwareSerial.h"
SoftwareSerial Serial1(6, 7);
```



```
// Σύνδεση ESP8266 με Arduino: TX->6, RX->7
#endif

// Η βιβλιοθήκη για το ESP8266
#include "WiFiEsp.h"
// Δημιουργία ενός αντικειμένου Ethernet client
WiFiEspClient client;

// Κωδικοί Wi-Fi
char ssid[] = "XXXXXXXXXX";
// Στη θέση των X γράψτε το Wi-Fi SSID (name)
char pass[] = "XXXXXXXXXX";
// Στη θέση των X γράψτε το Wi-Fi password
int status = WL_IDLE_STATUS; // Η κατάσταση του Wi-Fi

// IP του thingspeak.com
char server[] = "184.106.153.149";
// Το "Write API Key" του λογαριασμού σας στο thingspeak
char writeKey[] = "XXXXXXXXXXXXXXXXXXXX";
// Η βιβλιοθήκη για το δίαυλο I2C bus, ο οποίος
// χρησιμοποιείται από τον αισθητήρα αερίων
#include <Wire.h>

// Η βιβλιοθήκη για την οθόνη LCD
#include <LiquidCrystal.h>

// Η βιβλιοθήκη για τον αισθητήρα αερίων
#include "SparkFunCCS811.h"
#define CCS811_ADDR 0x5B // Η default διεύθυνση I2C
// #define CCS811_ADDR 0x5A // Εναλλακτική διεύθυνση I2C
// (αν η default είναι δεσμευμένη)

// Ο αισθητήρας αερίων
CCS811 mySensor(CCS811_ADDR);
// Η οθόνη LCD
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
// Οι συνδέσεις της οθόνης LCD

// Η βιβλιοθήκη για τον αισθητήρα θερμοκρασίας-υγρασίας
#include "DHT.h"
#define DHTPIN 13
// Το pin στο οποίο συνδέεται ο αισθητήρας DHT
#define DHTTYPE DHT11 // Ο τύπος του αισθητήρα
DHT dht(DHTPIN, DHTTYPE);
// Ο αισθητήρας θερμοκρασίας-υγρασίας

// Τα pin όπου συνδέεται το μοτέρ του ανεμιστήρα οροφής
int INA = 9;
```

```
int INB = 8;

// Μεταβλητές για θερμοκρασία και υγρασία
float tem;
float hum;
// Μεταβλητές για διοξείδιο του άνθρακα και πιητικές
// οργανικές ενώσεις
int co2;
int tvoc;

void setup()
{
  // Αρχικοποίηση σειριακής θύρας για monitor
  Serial.begin(9600);
  // Αρχικοποίηση σειριακής θύρας για ESP module
  // (software serial)
  Serial1.begin(9600);
  // Αρχικοποίηση ESP module
  WiFi.init(&Serial1);

  // Έλεγχος για την ύπαρξη του ESP module (shield)
  if( WiFi.status() == WL_NO_SHIELD )
  {
    Serial.println("Wi-Fi shield not present");
    // τέλος...
    while(true);
  }

  // Προσπάθεια σύνδεσης στο δίκτυο Wi-Fi
  while( WiFi.status() != WL_CONNECTED )
  {
    Serial.print("Attempting to connect to WPA SSID: ");
    Serial.println(ssid);
    // Σύνδεση με WPA/WPA2
    status = WiFi.begin(ssid, pass);
  }
  // Επιτυχής σύνδεση
  Serial.println("You're connected to the Wi-Fi");
  Serial.println();

  // Εμφάνιση μηνυμάτων στην οθόνη LCD
  lcd.begin(16,2);
  lcd.setCursor(0, 0);
  lcd.print("1o EPAL TRIKALON");
  lcd.setCursor(0, 1);
  lcd.println("Gas Sensor & LCD");
}
```

```

// Αρχικοποίηση I2C hardware
Wire.begin();

// Αρχικοποίηση αισθητήρα θερμοκρασίας-υγρασίας
dht.begin();

// Αρχικοποίηση αισθητήρα αερίων
if( mySensor.begin() == false )
{
    lcd.println("CCS811 Error...");
    while (true);    // Αν εμφανιστεί πρόβλημα, τέλος...
}

// Ορισμός pin εξόδου για τον ανεμιστήρα
pinMode(INA, OUTPUT);
pinMode(INB, OUTPUT);
}

void loop()
{
    // Αν υπάρχουν εισερχόμενα μηνύματα από τον server,
    // εμφάνισέ τα
    while( client.available() )
    {
        char c = client.read();
        Serial.write(c);
    }

    // Διάβασε θερμοκρασία (Celsius) και υγρασία
    tem = dht.readTemperature();
    hum = dht.readHumidity();

    /* Εμφάνιση μηνυμάτων στο serial monitor (για debugging)
    Serial.print(F("Humidity: "));
    Serial.print(h);
    Serial.print(F("% Temperature: "));
    Serial.print(t);
    Serial.println(F("°C "));
    */

    // Εμφάνισε θερμοκρασία-υγρασία στην LCD οθόνη
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Hum:");
    lcd.print((int)hum);
    lcd.print(F("%-"));
    lcd.print(tem);
    lcd.print(F("o C"));
}

```

```
// Έλεγχξε αν ο αισθητήρας αερίων έχει δεδομένα
if( mySensor.dataAvailable() )
{
    // Διάβασε και υπολόγισε τα αποτελέσματα από τον
    // αισθητήρα
    mySensor.readAlgorithmResults();
    co2 = mySensor.getCO2();
    tvoc = mySensor.getTVOC();

    // Εμφάνισε το υπολογισμένο ως ισοδύναμο του CO2 στην
    // οθόνη LCD
    lcd.setCursor(0, 1);
    lcd.print("CO2=");
    lcd.print(co2);
    // Εμφάνισε την υπολογισμένη ποσότητα tVOC στην οθόνη
    // LCD
    lcd.print(" ");
    lcd.print("tVOC=");
    lcd.print(tvoc);
    lcd.print("          ");
}

// Συνδέσου με το thingspeak για αποστολή των δεδομένων
Serial.println("Starting connection to thingspeak...");
// Αν η σύνδεση ήταν επιτυχής,
if( client.connect(server, 80) )
{
    Serial.println("Connected to thingspeak.com");
    // Δημιουργία του αιτήματος HTTP, όπως απαιτεί το
    // thingspeak.com
    String request= "GET /update?key=";
    request += writeKey;      // Γράψε API key
    request += "&field1=";    // field1 για CO2
    request += String(co2);
    request += "&field2=";    // field2 για TVOC
    request += String(tvoc);
    request += "&field3=";    // field3 για θερμοκρασία
    request += String(tem);
    request += "&field4=";    // field4 για υγρασία
    request += String(hum);

    Serial.println(request);
    // Εμφάνιση στο serial monitor για debugging
    // Αποστολή του αιτήματος
    client.println(request);
    // client.println("GET /asciiologo.txt HTTP/1.1");
    client.println("Host: 184.106.153.149");
}
```

```

client.println("Connection: close");
client.println();
}

// Αν η θερμοκρασία είναι πάνω από 26 βαθμούς C, τότε
// άναψε τον ανεμιστήρα
if( tem > 26.0 )
{
    digitalWrite(INA, HIGH);
    digitalWrite(INB, LOW);
}
else // Αλλιώς σβήσε τον
{
    digitalWrite(INA, LOW);
    digitalWrite(INB, LOW);
}

// Μια καθυστέρηση 5 sec
delay(5000);
}

```



Η τοποθέτηση του συστήματος παρακολούθησης ποιότητας αέρα μπορεί να γίνει στο εσωτερικό ενός κτιρίου, που μπορείτε να εκτυπώσετε σε έναν 3D εκτυπωτή (οδηγίες στο βιβλίο «Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο»). Το κτίριο αυτό αποτελείται από πέντε (5) μέρη: α) Βιοκλιματικό κτίριο σχολείο (<http://tiny.cc/1lyutz>), β) Δεξαμενή όμβριων υδάτων (<http://tiny.cc/9lyutz>), γ) Σκεπή κτιρίου (<http://tiny.cc/alyutz>), δ) Στέγη γκαζόν κάτω (<http://tiny.cc/4lyutz>) και ε) Στέγη γκαζόν πάνω (<http://tiny.cc/6lyutz>).

Ερωτήσεις

1. Τα δεδομένα που ανεβάζετε στο ThingSpeak είναι προσβάσιμα σε όλους.
 Σωστό Λάθος
2. Στην καρτέλα “API Keys” φαίνονται...
 - α. Το δίκτυο Wi-Fi
 - β. Τα πεδία
 - γ. Οι κωδικοί πρόσβασης
 - δ. Το κανάλι σας
3. Ένα Arduino δεν μπορεί να συνδεθεί σε δίκτυο Wi-Fi.
 Σωστό Λάθος

Δραστηριότητες

Τροποποιήστε το κύκλωμά σας, προκειμένου να εμφανίζεται επιπλέον η υγρασία εδάφους στο κανάλι σας, στο ThingSpeak, ακολουθώντας τα παρακάτω βήματα:

1. Προσθέστε έναν αισθητήρα υγρασίας εδάφους YL-38 σύμφωνα με τις οδηγίες της ενότητας 6.1.
2. Προσθέστε τον κατάλληλο κώδικα στο Arduino, ώστε να εμφανίζεται στην οθόνη σας η υγρασία εδάφους ως 3η οθόνη σε σειρά.
3. Προσθέστε ένα πεδίο στο ThingSpeak για την υγρασία του εδάφους.
4. Αποστείλτε τις τιμές ως 5^ο πεδίο, έτσι ώστε να εμφανίζονται και αυτές στο ThingSpeak.

Λέξεις κλειδιά

CO₂

Αποστολή δεδομένων

Mathworks

Βιβλιοθήκες

ThingSpeak

Θερμοκρασία

tVOC

Νέφος

Ποιότητα αέρα

Υγρασία

Βιβλιογραφία

- Bansod, S. Jaiswal, R. , Sargam, P., Survase, P., & , Sawant, S. (2018). Arduino Based Water Irrigation System. *International Journal of Advance Research in Science and Engineering*, 7(3), 903-910. Retrieved March 20, 2021, from https://www.researchgate.net/publication/338824067_ARDUINO_BASED_WATER_IRRIGATION_SYSTEM
- Γίδας, Γ., & Μαντζάνας, Ε. (2021). *Τρισδιάστατη Σχεδίαση στο σχολικό εργαστήριο*. 1^ο ΕΠΑΛ Τρικάλων: Τρίκαλα.
- Greengard, S. (2015). *The Internet of Things*. MIT Press Ltd: Cambridge, Mass.
- McRoberts, M. (2009). *Earthshine Design Arduino Starters Kit Manual - A Complete Beginners Guide to the Arduino*. Earthshine Design.
- Norris, D. (2015). *The Internet of Things: Do-It-Yourself at Home Projects for Arduino, Raspberry Pi and BeagleBone Black*. McGraw-Hill Education: New York.
- Νούσης, Β. (2019). *Arduino για αρχάριους, με έμφαση στη χρήση του στο εργαστήριο Φυσικών Επιστημών*. Ηγουμενίτσα.
- Πουλάκης, Ε. (2015). *Προγραμματίζοντας με τον μικροελεγκτή Arduino*. Αυτοέκδοση: Ηράκλειο.

Πηγές

- <https://create.Arduino.cc>
Επίσημη ιστοσελίδα ανάπτυξης κώδικα Arduino
- <https://github.com/adafruit/DHT-sensor-library>
Βιβλιοθήκες αισθητήρα καταγραφής θερμοκρασίας-υγρασίας DHT
- <https://github.com/bportaluri/WiFiEsp>
Βιβλιοθήκες πλακέτας διασύνδεσης ESP8266 Arduino με Wi-Fi

-
- https://github.com/sparkfun/SparkFun_CCS811_Arduino_Library
Βιβλιοθήκες – παραδείγματα λειτουργίας συσκευής καταγραφής ποιότητας αέρα
 - <https://grobotronics.com/dual-motor-driver-module-l298n.html>
Οδηγίες χρήσης οδηγού για κινητήρες συνεχούς ρεύματος
 - <https://grobotronics.com/inductive-proximity-sensor-18mm-lj18a3-8-z-ax.html>
Οδηγίες χρήσης επαγωγικού αισθητήρα απόστασης LJ18A3-8-Z/AX
 - <https://grobotronics.com/microswitch-spdt-on-on-with-big-lever.html>
Οδηγίες χρήσης μικροδιακόπτη SPDT
 - <https://grobotronics.com/stepper-driver-breakout-uln2003.html>
Οδηγίες χρήσης πλακέτας οδήγησης βηματικού κινητήρα ULN2003
 - <https://howtomechatronics.com>
Ιστότοπος εκπαίδευσης στον τομέα της Μηχανολογίας, Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
 - <https://howtomechatronics.com/tutorials/Arduino/ultrasonic-sensor-hc-sr04/>
Τρόπος λειτουργίας αισθητήρα υπερήχων HC-SR04
 - <https://www.instructables.com/>
Κοινότητα ανταλλαγής καλών πρακτικών



1^ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ

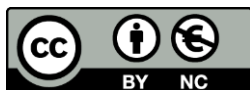
Διεύθυνση: ΒΑΘΥΚΛΕΟΥΣ 1, 42100-ΤΡΙΚΑΛΑ
Τηλέφωνο: 243-102-8977, **FAX:** 243-102-8964
Email: mail@1epal-trikal.tri.sch.gr
Ιστοσελίδα: <http://1epal-trikal.tri.sch.gr/>

ΣΥΓΓΡΑΦΙΚΗ ΟΜΑΔΑ:

Γεώργιος Γίδας (ggidas@gmail.com)
Ευάγγελος Μαντζάνας (emantzanas@sch.gr)

ΕΠΙΜΕΛΕΙΑ:

Νικόλαος Ντόγκας (ntogas@yahoo.gr)



ISBN: 978-618-85383-0-6

Copyright©:

1^ο ΕΠΑΛ ΤΡΙΚΑΛΩΝ, ΓΕΩΡΓΙΟΣ ΓΙΔΑΣ, ΕΥΑΓΓΕΛΟΣ ΜΑΝΤΖΑΝΑΣ

Το παρόν βιβλίο τυπώθηκε με δαπάνη του 1^{ου} ΕΠΑΛ ΤΡΙΚΑΛΩΝ και διανέμεται δωρεάν στην έντυπη και ηλεκτρονική του μορφή.