



**ΠΡΟΓΡΑΜΜΑ ΜΕΤΑΠΤΥΧΙΑΚΩΝ ΣΠΟΥΔΩΝ
«ΡΟΜΠΟΤΙΚΗ, STEAM ΚΑΙ ΝΕΕΣ ΤΕΧΝΟΛΟΓΙΕΣ
ΣΤΗΝ ΕΚΠΑΙΔΕΥΣΗ»**

Διπλωματική Εργασία

**«ΑΝΑΠΤΥΞΗ ΕΚΠΑΙΔΕΥΤΙΚΟΥ ΥΛΙΚΟΥ ΜΕ ARDUINO
ΓΙΑ ΤΟ ΜΑΘΗΜΑ ΤΗΣ ΤΕΧΝΟΛΟΓΙΑΣ
ΣΤΗ ΔΕΥΤΕΡΟΒΑΘΜΙΑ ΕΚΠΑΙΔΕΥΣΗ»**

της

ΟΛΓΑΣ ΜΑΛΑΜΟΥΔΗ

Επιβλέπων Καθηγητής
Χρήστος Υφούλης

Υποβλήθηκε ως απαιτούμενο για την απόκτηση του μεταπτυχιακού διπλώματος
ειδίκευσης Ρομποτική, STEAM και Νέες Τεχνολογίες στην Εκπαίδευση
Θεσσαλονίκη, Ιανουάριος 2024



Η παρούσα Διπλωματική Εργασία καλύπτεται στο σύνολό της νομικά από δημόσια άδεια πνευματικών δικαιωμάτων Creative Commons:

Αναφορά Δημιουργού - Μη Εμπορική Χρήση - Παρόμοια Διανομή



Μπορείτε να:

- Μοιραστείτε: αντιγράψετε και αναδιανέμετε το παρόν υλικό με κάθε μέσο και τρόπο
- Προσαρμόστε: αναμείξτε, τροποποιήστε και δημιουργήστε πάνω στο παρόν υλικό

Υπό τους ακόλουθους όρους:

- Αναφορά Δημιουργού: Θα πρέπει να καταχωρίσετε αναφορά στη 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 η δημιουργός αποδέχεται το έργο σας ή τη χρήση που εσείς κάνετε.
- Μη Εμπορική Χρήση: Δε μπορείτε να χρησιμοποιήσετε το υλικό για εμπορικούς σκοπούς.
- Παρόμοια Διανομή: Αν αναμείξετε, τροποποιήσετε, ή δημιουργήσετε πάνω στο παρόν υλικό, πρέπει να διανείμετε τις δικές σας συνεισφορές υπό την ίδια άδεια Creative Commons όπως και το πρωτότυπο.

Αναλυτικές πληροφορίες νομικού κώδικα στην ηλεκτρονική διεύθυνση:

<https://creativecommons.org/licenses/by-nc-sa/4.0/legalcode>

Υπεύθυνη Δήλωση Φοιτήτριας

Η παρούσα Διπλωματική Εργασία και τα συμπεράσματά της, σε οποιαδήποτε μορφή, αποτελούν συνιδιοκτησία του Τμήματος Μηχανικών Παραγωγής και Διοίκησης του Διεθνούς Πανεπιστημίου Ελλάδος και της φοιτήτριας. Οι προαναφερόμενοι διατηρούν το δικαίωμα ανεξάρτητης χρήσης και αναπαραγωγής (τμηματικά ή συνολικά) για διδακτικούς και ερευνητικούς σκοπούς. Σε κάθε περίπτωση πρέπει να αναφέρεται ο τίτλος, η συγγραφέας, ο επιβλέπων και το τμήμα του ΔιΠαΕ.

Η έγκριση της παρούσας Διπλωματικής Εργασίας από το Τμήμα Μηχανικών Παραγωγής και Διοίκησης δεν υποδηλώνει απαραίτητως και αποδοχή των απόψεων της συγγραφέως εκ μέρους του Τμήματος.

Δηλώνω υπεύθυνα ότι η παρούσα Διπλωματική Εργασία είναι εξ ολοκλήρου δικό μου έργο και συγγράφηκε ειδικά για τις απαιτήσεις του μεταπτυχιακού προγράμματος σπουδών του Τμήματος Μηχανικών Παραγωγής και Διοίκησης με τίτλο «Ρομποτική, STEAM και νέες Τεχνολογίες στην Εκπαίδευση».

Κατά τη συγγραφή ακολούθησα την πρόπουσα ακαδημαϊκή δεοντολογία αποφυγής λογοκλοπής και έχω αποφύγει οποιαδήποτε ενέργεια που συνιστά παράπτωμα λογοκλοπής. Δεν χρησιμοποίησα ολόκληρο ή μέρος έργου άλλου δημιουργού ή τις ιδέες και αντιλήψεις άλλου δημιουργού χωρίς να γίνεται σαφής αναφορά στην πηγή προέλευσης (βιβλίο, άρθρο από επιστημονικό περιοδικό, ιστοσελίδα κλπ.).

Θεσσαλονίκη, 2 Ιανουαρίου 2024

Η Δηλούσα: Όλγα Μαλαμούδη

ΠΕΡΙΛΗΨΗ

Τα ακόλουθα διδακτικά σενάρια με τη χρήση του Arduino στα πλαίσια του μαθήματος της Τεχνολογίας στη Δευτεροβάθμια Εκπαίδευση, έχουν ως σκοπό να έρθουν οι μαθητές και οι μαθήτριες σε μια πρώτη επαφή με τον Arduino, να γνωρίσουν ένα μέρος των δυνατοτήτων του, ώστε να τον εντάξουν σε κάποια κατασκευή τους. Το ζητούμενο δεν είναι να μάθουν να γράφουν σύνθετους και πολύπλοκους κώδικες, αλλά να είναι σε θέση να γράψουν έναν απλό κώδικα στο Arduino IDE, να καταλάβουν, με τα σενάρια που θα διδαχτούν, πώς προγραμματίζονται κάποιοι αισθητήρες, πώς εκτελούν τις εντολές οι ενεργοποιητές, και να είναι σε θέση κάποιον κώδικα που θα βρουν έτοιμο στο διαδίκτυο να τον τροποποιήσουν ώστε να εκτελέσει τις ενέργειες που επιθυμούν.

ABSTRACT

The following teaching scenarios aim to utilize the presence of an Arduino unit in teaching the subject of Technology in Secondary Education, so as for the students to make first contact with the Arduino, familiarize themselves with its core capabilities and eventually integrate it in a project of their own.

The aim is not to instruct the students on how to write elaborate and complex pieces of code, but instead to focus on enabling them to write simple programs on the Arduino IDE framework, understand – through the scenarios they will be shown – how to configure an array of sensors, how the commands given are executed by the actuators, and give students the tools necessary for them to modify sample code available online to fit to their needs and goals.

ΠΙΝΑΚΑΣ ΠΕΡΙΕΧΟΜΕΝΩΝ

STEAM	7
Διαθεματικότητα	7
Μαθητοκεντρική διδασκαλία	8
Διερευνητική μάθηση	8
Ομαδοσυνεργατική Διδασκαλία	9
Δημιουργικότητα	9
Arduino	11
Arduino UNO	11
Ψηφιακές εισόδου/έξοδοι	12
Ψευδοαναλογικές έξοδοι	13
PWM	13
Αναλογικές εισόδου	13
Βιβλιοθήκες	14
Breadboard	14
Καλώδια	15
LED	16
Αντιστάτες ή Αντιστάσεις	17
Αισθητήρες	19
Push Button ή Διακόπτης ή Κομβίο (κοινώς κουμπί)	20
Φωτοαντίσταση.....	20
Ποτενσιόμετρο.....	21
Αισθητήρας Υπερήχων – Ultrasonic Sensor HC-SR04.....	22
Αισθητήρας Υγρασίας Εδάφους (Χωρητικός).....	23
Ενεργοποιητές	24
Buzzer	24
Σερβοκινητήρας ή Servo.....	25
Αντλία Νερού.....	26
Ρελέ.....	27
Εκπαιδευτικό Υλικό με Arduino	28
1^ο Διδακτικό Σενάριο: Θα καεί το led ή δεν θα καεί;	31
2^ο Διδακτικό Σενάριο: Μα τι είναι αυτό το «breadboard»;	34
3^ο Διδακτικό Σενάριο: Χρήση Push Button και Buzzer	38
4^ο Διδακτικό Σενάριο: Γνωριμία με τον Arduino	44
5^ο Διδακτικό Σενάριο: Φανάρια Κυκλοφορίας	52

6° Διδακτικό Σενάριο: Εισαγωγή στους αισθητήρες - Push Button	57
7° Διδακτικό Σενάριο: Φωτοαντίσταση	63
8° Διδακτικό Σενάριο: Ποτενσιόμετρο.....	67
9° Διδακτικό Σενάριο: Αισθητήρας Υπερήχων με LED και Buzzer.....	71
10° Διδακτικό Σενάριο: Αισθητήρας Υπερήχων και Servo - Theremin	77
11° Διδακτικό Σενάριο: Αισθητήρας Υπερήχων και Αντλία Νερού με Ρελέ.....	82
12° Διδακτικό Σενάριο: Αισθητήρας Υγρασίας Εδάφους και Αντλία Νερού με Ρελέ.	87
Ολοκλήρωση διδακτικών σεναρίων	91
Κατασκευές.....	92
Φανάρια Κυκλοφορίας Οχημάτων και Πεζών.....	93
Theremin	99
Έξυπνος Κάδος	104
Αυτόματη Βρύση	108
Αυτόματο Πότισμα	112
Ολοκλήρωση Κατασκευών (ή αλλιώς Trial & Error).....	116
Εκπαίδευση STEAM και Arduino - Συμπεράσματα.....	117
Διαθεματικότητα STEAM.....	117
Διερευνητική μάθηση.....	117
Ομαδοσυνεργατική Διδασκαλία.....	117
Ενδιαφέρον για τα παιδιά	117
Προβληματισμοί.....	118
Πίνακας Κόστους Υλικών.....	119
Βιβλιογραφία.....	120
Πηγές	120
Πηγές Σύνταξης Κώδικα Αισθητήρων & Ενεργοποιητών	120
Πηγή κώδικα Theremin.....	120
Πηγή Εικόνας Χρωματικού Κώδικα	120
Λογισμικό	120
Κυκλώματα – Κατασκευές - Εικόνες - Φωτογραφίες.....	121
Ευχαριστίες.....	122

STEAM

Ο όρος STEAM αναφέρεται στη συνολική προσέγγιση ενός συνόλου γνωστικών πεδίων, τα οποία σχετίζονται με γνώσεις και δεξιότητες που θεωρούνται κλειδί για τον μετασχηματισμό της οικονομίας, την αύξηση της παραγωγικότητας και την ενίσχυση της κοινωνικής συνοχής. Τα STEAM απαρτίζονται από τις Φυσικές Επιστήμες (Science), την Τεχνολογία (Technology), τη Μηχανική (Engineering), τις Τέχνες (Arts) και τα Μαθηματικά (Mathematics). Τα STEAM ορίζονται ως ένα συνδυασμένο σύνολο γνώσεων και δεξιοτήτων σε όλα τα επιμέρους γνωστικά πεδία, που επιτρέπουν τη συνδυαστική χρήση της γνώσης και τη διεπιστημονική προσέγγιση για την αντιμετώπιση πραγματικών προβλημάτων και την ανάπτυξη κρίσιμων δεξιοτήτων.

Όσον αφορά στη σχολική εκπαίδευση των STEAM, αυτή δεν περιορίζεται στην απλή διδασκαλία των βασικών γνωστικών πεδίων που αποτελούν το ακρωνύμιο. Αντίθετα, επικεντρώνεται στην καλλιέργεια της κριτικής σκέψης και της ικανότητας αξιολόγησης και χρήσης της πληροφορίας, αντί για την απλή απομνημόνευση πληροφοριών, που εφαρμόζεται συχνά στη σχολική εκπαίδευση.

Τα κύρια συστατικά της εκπαίδευσης STEM είναι τα παρακάτω:

Διαθεματικότητα

Η προσέγγιση που συνδυάζει διεπιστημονικές και διαθεματικές πτυχές αποτελεί βασικό στοιχείο στην εκπαίδευση STEAM. Αντίθετα με την παλαιότερη εκπαιδευτική άποψη που επικεντρωνόταν στην απομνημόνευση ενός μεμονωμένου γνωστικού πεδίου, η διεπιστημονική προσέγγιση τοποθετεί κάθε γνωστικό θέμα σε ένα ευρύ διαθεματικό πλαίσιο, με έμφαση στην αριθμητικότητα ή αλλιώς τον "μαθηματικό εγγραμματισμό". Στόχος είναι η ενεργοποίηση του εγκεφάλου για τη δημιουργία νευρικών συνάψεων, που οδηγούν στην ενεργητική απόκτηση γνώσης.

Στην εκπαίδευση STEAM, η διδασκαλία επικεντρώνεται κυρίως στην υλοποίηση μαθητικών εργασιών, γνωστών και ως «projects». Αυτά τα projects εκτείνονται σε μεγαλύτερη χρονική διάρκεια, με διάρκεια πολλές φορές περισσότερες μέρες ή ακόμη και εβδομάδες, εμπλέκοντας παράλληλα περισσότερους εκπαιδευτικούς με διάφορες ειδικότητες. Σκοπός είναι να παρασχεθεί στους μαθητές και στις μαθήτριες μια

ολοκληρωμένη κατανόηση των αλληλεξαρτήσεων που διαμορφώνουν τη λειτουργία του φυσικού, αλλά και του ψηφιακού κόσμου.

Μαθητοκεντρική διδασκαλία

Η εκπαίδευση STEAM θέτει τον μαθητή και τη μαθήτριά στο επίκεντρο. Σε αντίθεση με την παραδοσιακή δασκαλοκεντρική προσέγγιση, όπου ο/η εκπαιδευτικός διαδραματίζει τον ενεργό ρόλο έναντι ενός σχεδόν παθητικού αποδέκτη, στην μαθητοκεντρική προσέγγιση ο μαθητής/ η μαθήτριά αναλαμβάνει έναν ενεργό και υπεύθυνο ρόλο στην ατομική του/της μαθησιακή πορεία. Η εκπαίδευση STEAM υιοθετεί εκπαιδευτικές πρακτικές, όπως η ανάθεση πρακτικών ασκήσεων ή ατομικών εργασιών, αντί της απομνημόνευσης, και η επίλυση προβλημάτων, αντί της απλής επαναληπτικής αναπαραγωγής ενός κειμένου. Αυτό ενισχύει την ικανότητα των μαθητών/τριών για συνεχή μάθηση και ανεξάρτητη επίλυση προβλημάτων, τα οποία αποτελούν κρίσιμες δεξιότητες για την επαγγελματική τους επιβίωση στο σύγχρονο παραγωγικό περιβάλλον. Σε αυτό το πλαίσιο, όπου οι επαγγελματικοί χώροι και ρόλοι αλλάζουν διαρκώς και η τεχνολογική πρόοδος απαιτεί συνεχή προσαρμογή, η εκπαίδευση STEAM ανταποκρίνεται στις αυξανόμενες απαιτήσεις.

Διερευνητική μάθηση

Στόχος της διερευνητικής μάθησης είναι να προωθήσει την ενεργό συμμετοχή των μαθητών/τριών στη διαδικασία μάθησης, να ενισχύσει τις δεξιότητές τους στον τομέα της κριτικής σκέψης και της επίλυσης προβλημάτων, και να τους παράσχει τη δυνατότητα να εφαρμόζουν τις γνώσεις τους σε πρακτικές καταστάσεις. Η διερευνητική μάθηση συνδέεται συχνά με πρακτικές όπως ο προβληματισμός, η έρευνα, ο σχεδιασμός έργων και η συνεργασία, προωθώντας την ανάπτυξη δεξιοτήτων που απαιτούνται στον σύγχρονο κόσμο, όπως η κριτική σκέψη, η αντιμετώπιση προβλημάτων και η συνεργατικότητα.

Ομαδοσυνεργατική Διδασκαλία

Σε κάθε φάση της ζωής και σε όλους τους τομείς της κοινωνικής και επαγγελματικής δραστηριότητάς του, ο άνθρωπος καλείται να αλληλεπιδρά με άλλα άτομα. Επομένως, οι κοινωνικές και επικοινωνιακές δεξιότητες αποτελούν κρίσιμο παράγοντα για τον σύγχρονο άνθρωπο, τόσο ως πολίτης όσο και ως εργαζόμενος. Ιδιαίτερα στους τομείς των STEAM, η συνεργασία σε ομάδες αναγνωρίζεται ως ουσιαστικό στοιχείο της αποτελεσματικής εκπαίδευσης. Με την ανταλλαγή απόψεων, εμπειριών και ιδεών, οι μαθητές/τριες αποκτούν πιο ευχάριστη και αποδοτική μάθηση. Έχει παρατηρηθεί ότι όταν ένα θέμα εξηγείται από ένα μέλος της ομάδας σε έναν συμμαθητή ή μια συμμαθήτρια, ο μαθητής ή η μαθήτρια κατανοεί καλύτερα την πληροφορία. Επίσης, ο μαθητής ή η μαθήτρια που ακούει επωφελείται περισσότερο όταν η εξήγηση προέρχεται από κάποιον/α με παρόμοιο επίπεδο γνώσεων. Πέρα από τα γνωστικά οφέλη, η ομαδική εργασία επηρεάζει θετικά και την κοινωνικοσυναισθηματική ανάπτυξη. Οι μαθητές/τριες αναπτύσσουν διαπροσωπικές σχέσεις, δημιουργούν φιλίες, επικοινωνούν εποικοδομητικά, δημιουργούν θετικά συναισθήματα, μαθαίνουν να εκτιμούν τη συνεισφορά των άλλων μελών της ομάδας και εκπαιδεύονται σε θέματα όπως η συνέπεια, η υπευθυνότητα και η σωστή κοινωνική συμπεριφορά. Ταυτόχρονα, αποκτούν μεγαλύτερη αυτοπεποίθηση, αυτοεκτίμηση και αυτογνωσία, χαρακτηριστικά που διευκολύνουν την προσωπική ανάπτυξη, τη διαρκή μάθηση και την αυτοπραγμάτωση.

Δημιουργικότητα

Η αυτοματοποίηση, η ρομποτική, η πληροφορική και οι τηλεπικοινωνίες επιφέρουν ουσιαστικές αλλαγές στον σύγχρονο εργασιακό χώρο. Οι καθιερωμένες, επαναλαμβανόμενες, χειρωνακτικές ή πνευματικές εργασίες σταδιακά υποκαθίστανται από πιο πολύπλοκες και δημιουργικές απαιτήσεις, οι οποίες μπορεί ακόμη και να εξελιχθούν σε νέες, πολλές φορές εντελώς διαφορετικές, κατά τη διάρκεια της εργασιακής ζωής. Αυτό υποδηλώνει ότι οι επόμενες γενιές εργαζομένων θα πρέπει να είναι εξοπλισμένες με υψηλό βαθμό ευελιξίας, δημιουργικότητας και καινοτομίας. Εξάλλου, η αναγνώριση της σημασίας της δημιουργικότητας οδήγησε πρόσφατα στην ενσωμάτωση των Τεχνών (Arts) στα βασικά πεδία STEM. Υπάρχουν

πολλοί τρόποι με τους οποίους η δημιουργικότητα μπορεί να αναπτυχθεί στο πλαίσιο του σχολείου. Είναι σημαντικό, πρωτίστως, να δίνεται στους μαθητές και στις μαθήτριες η δυνατότητα να αντιμετωπίζουν καταστάσεις και να ενθαρρύνονται στην δημιουργική σκέψη και τον πειραματισμό. Επομένως, είναι απαραίτητη η κατάλληλη κατάρτιση των εκπαιδευτικών και η ανάπτυξη αντίστοιχων εκπαιδευτικών εργαλείων και πόρων.

Arduino

Ο Arduino είναι ένας μικροελεγκτής, δηλαδή ένας αυτόνομος μικροεπεξεργαστής με έναν μικρό αριθμό ολοκληρωμένων κυκλωμάτων που μπορεί να λειτουργήσει με ελάχιστα εξωτερικά εξαρτήματα. Ο μικροελεγκτής επιτρέπει τον έλεγχο ηλεκτρονικών κυκλωμάτων, αισθητήρων και συστημάτων επέκτασης, μέσω ενός αποθηκευμένου προγράμματος που συντάσσεται σε γλώσσα Wiring C. Έτσι, με τον Arduino μπορούμε να κάνουμε τα πάντα, από λαμπάκια που αναβοσβήνουν μόνα τους μέχρι αυτόνομα ρομποτικά οχήματα.

Arduino UNO

Ο Arduino Uno και η έκδοση 1.0 του Arduino Software (IDE) ήταν οι εκδόσεις αναφοράς του Arduino, που τώρα εξελίχθηκαν σε νεότερες εκδόσεις. Η πλακέτα Uno είναι η πρώτη από μια σειρά πλακετών Arduino, και το μοντέλο αναφοράς για την πλατφόρμα Arduino. Εκτός από την πλακέτα της εταιρίας Arduino, στην αγορά κυκλοφορούν και κλώνοι.



Κλώνοι Arduino με διάφανη θήκη

Ο προγραμματισμός του Arduino μπορεί να γίνει είτε μέσα από το περιβάλλον του Arduino IDE, όπου συντάσσουμε τον κώδικα στη γλώσσα Wiring C, είτε με άλλα

προγράμματα οπτικού προγραμματισμού, όπως το S4A (Scratch For Arduino), στα οποία ο προγραμματισμός γίνεται με τη βοήθεια πλακιδίων. Το λογισμικό Arduino IDE είναι ένα εύχρηστο λογισμικό ανοικτού κώδικα, με πλούσιο διαδικτυακό υλικό υποστήριξης. Για όλους αυτούς τους λόγους, το Arduino είναι ένα εξαιρετικό περιβάλλον για τη γνωριμία των παιδιών με τον προγραμματισμό και τη δημιουργία εφαρμογών, κατασκευών και ρομποτικών συστημάτων.

Η τροφοδοσία της πλακέτας γίνεται είτε απευθείας από τον υπολογιστή, συνδέοντας το καλώδιο που χρησιμοποιούμε για την μεταφορά του κώδικα, είτε με μετασχηματιστή 9V 2A DC , είτε με μπαταρία 9V και κατάλληλο βύσμα. Θα πρέπει να προσέξουμε ότι, αν η τάση εισόδου είναι μεγαλύτερη από 12V, μπορεί να καταστραφεί η πλακέτα, ενώ αν είναι μικρότερη από 7V, τότε ίσως η λειτουργία της να είναι ασταθής.

Ο Arduino διαθέτει 14 ψηφιακές εισόδους/εξόδους (0 έως 13) και 6 αναλογικές εισόδους (A0 έως A5). Σε αυτές μπορούμε να συνδέσουμε αισθητήρες αλλά και ενεργοποιητές. Στη συνέχεια θα περιγράψουμε αυτούς που αναφέρονται στα διδακτικά σενάρια.

Ψηφιακές εισοδοί/έξοδοι

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

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

Ψευδοαναλογικές εξόδους

Είδαμε προηγουμένως ότι στις ψηφιακές εξόδους του Arduino η τάση εξόδου μπορεί να είναι 0 ή 5V. Κάποιες εξόδους όμως, που είναι σημειωμένες με το σύμβολο «~» (οι 3, 5, 6, 9, 10 και 11) μπορούν, με τη μέθοδο PWM, να δίνουν 256 διαφορετικά επίπεδα ψευδοαναλογικής τάσης, με το επίπεδο 0 να αντιστοιχεί στα 0V, ενώ το επίπεδο 255 αντιστοιχεί στα 5V. Ο όρος «ψευδοαναλογική» αφορά στο ότι οι ενδιάμεσες τιμές τάσης δεν είναι άπειρες, και συνεπώς δεν θα ήταν σωστό να πούμε ότι είναι αναλογική.

PWM

Η τεχνική PWM (Pulse Width Modulation - Διαμόρφωση Πλάτους) είναι 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.

Εκτός από τον Arduino και το πρόγραμμα Arduino IDE που χρειάζεται να έχουμε εγκατεστημένο στον υπολογιστή μας, για να μπορέσουμε να υλοποιήσουμε κάποιο κύκλωμα με αισθητήρες ή/και ενεργοποιητές, είναι απαραίτητα και μερικά ακόμη στοιχεία, τα οποία θα δούμε παρακάτω.

Αναλογικές εισόδους

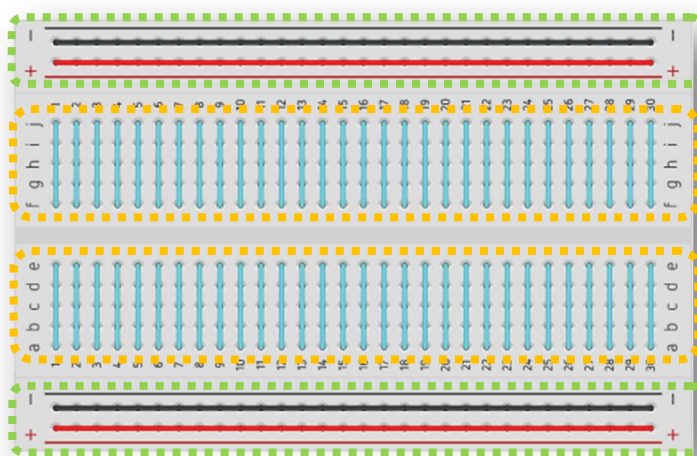
Οι αναλογικές εισόδους, που συμβολίζονται με A0 έως A5, μπορούν να αναγνωρίσουν και ενδιάμεσες τιμές, καθώς το Arduino έχει ενσωματωμένο έναν αναλογικοψηφιακό μετατροπέα εύρους 10-bit (analog to digital converter) ο οποίος αντιστοιχίζει την τάση εισόδου (εύρους 0V έως 5V εξ ορισμού) σε μια ακέραια τιμή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

Βιβλιοθήκες

Οι βιβλιοθήκες είναι συλλογές προ-γραμμένου κώδικα που απλοποιούν τη διαδικασία προγραμματισμού. Οι βιβλιοθήκες του Arduino είναι διαθέσιμες για διάφορους αισθητήρες, ενεργοποιητές και πρωτόκολλα επικοινωνίας, καθιστώντας ευκολότερη για τους προγραμματιστές τη χρήση αυτών των συστατικών στους κώδικές τους.

Breadboard

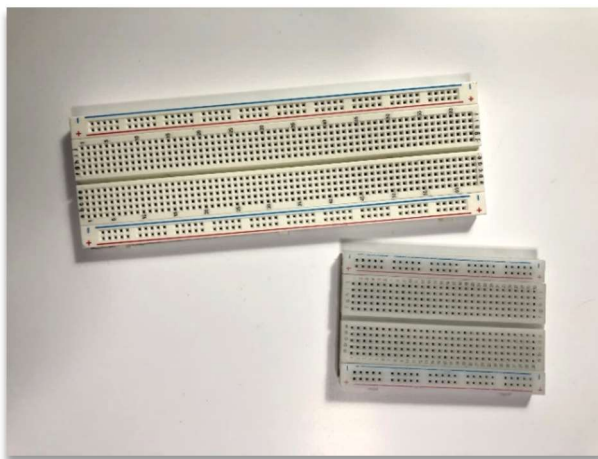
Για να αποφύγουμε τις κολλήσεις και να έχουμε τη δυνατότητα ανά πάσα στιγμή να τροποποιήσουμε κάποιο υπάρχον κύκλωμα, ή ακόμη να το διαλύσουμε και να κάνουμε από την αρχή κάποιο νέο, θα χρησιμοποιήσουμε το «Breadboard». Το breadboard είναι μια πλακέτα δοκιμών και έχει οπές οι οποίες είναι βραχυκυκλωμένες μεταξύ τους ανά ομάδες και με συγκεκριμένο τρόπο, όπως φαίνεται και στο παρακάτω σχήμα:



Breadboard

Στα πράσινα πλαίσια βλέπουμε τις γραμμές τροφοδοσίας, όπου οι οπές είναι βραχυκυκλωμένες οριζόντια (κόκκινες και μαύρες συνεχείς γραμμές), από τη μία άκρη του breadboard ως την άλλη. Σε όποια οπή μιας γραμμής και να βάλουμε ένα καλώδιο ή ένα εξάρτημα, το αποτέλεσμα είναι το ίδιο. Το ίδιο συμβαίνει και στις

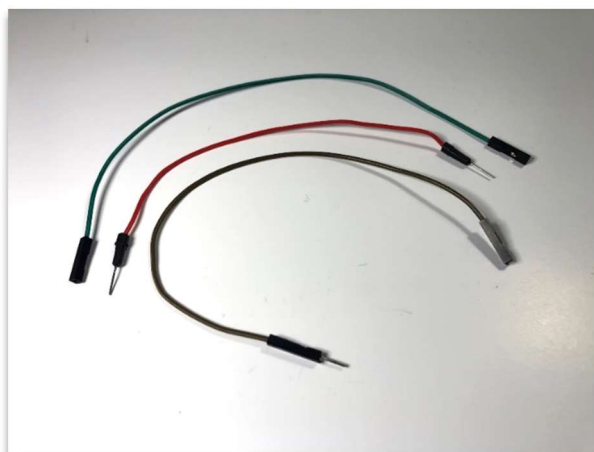
οπές που βρίσκονται στα κίτρινα πλαίσια, μόνο που οι βραχυκυκλωμένες οπές σχηματίζουν κάθετες γραμμές και παριστάνονται στο σχήμα με γαλάζιο χρώμα.



Breadboard μεγάλο και μεσαίο

Καλώδια

Στις οπές του breadboard μπαίνουν οι άκρες των καλωδίων και οι ακίδες των εξαρτημάτων. Υπάρχουν καλώδια M-M (Male - Male), δηλαδή αυτά που και τα δύο άκρα τους καταλήγουν σε ακίδες (αρσενικά), υπάρχουν καλώδια που και τα δύο άκρα τους καταλήγουν σε οπές (θηλυκά), τα F-F (Female - Female), καθώς επίσης και αυτά που το ένα άκρο τους είναι αρσενικό και το άλλο θηλυκό, τα M-F (Male - Female). Το καθένα από αυτά χρησιμοποιείται ανάλογα με τις απαιτήσεις του κυκλώματος.



Καλώδια M-M, F-F και M-F

Εκτός από τα καλώδια, τα δύο στοιχεία που χρησιμοποιούνται συχνότερα στα κυκλώματα είναι τα LED και οι αντιστάσεις.

LED

Η δίοδος LED (Light Emitted Diode – Δίοδος Εκπομπής Φωτός) είναι ένας ημιαγωγός (ένας αγωγός που επιτρέπει στο ρεύμα να περάσει μόνο προς τη μια κατεύθυνση), που όταν τον πολώσουμε ορθά, φωτοβολεί. Με απλά λόγια, πρόκειται για ένα ηλεκτρονικό στοιχείο με δύο ακροδέκτες, που όταν (οι ακροδέκτες) συνδεθούν σε μια πηγή, εκπέμπει φως.

Συναντάμε τα LED σε διάφορα μεγέθη και χρώματα, και χρησιμοποιούνται σε διάφορες εφαρμογές. Τα LED που βρίσκουν χρήση στις εφαρμογές με τον Arduino, έχουν διάμετρο 3mm ή 5mm.



LED 5mm

Οι ακροδέκτες των LED έχουν έναν μακρύ ακροδέκτη, την άνοδο «+», και έναν κοντύτερο, την κάθοδο «-». Για να εκπέμπει φως, να ανάψει το λαμπάκι δηλαδή, θα πρέπει η άνοδός του να συνδεθεί στον θετικό πόλο της πηγής μας, ενώ η κάθοδος στον αρνητικό πόλο. Τα LED για να φωτοβολήσουν χρειάζονται μία ελάχιστη τιμή τάσης, στα συγκεκριμένα που χρησιμοποιούμε στα διδακτικά σενάρια και έχουν διάμετρο 5mm η τιμή αυτή κυμαίνεται μεταξύ 2,2V και 3,2V. Επίσης η τιμή της

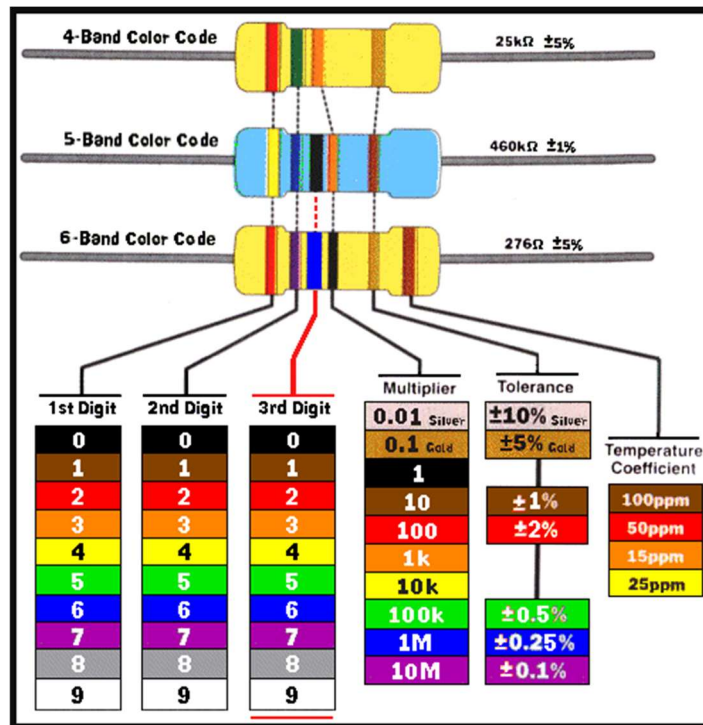
έντασης του ρεύματος δεν πρέπει να ξεπερνά μια μέγιστη τιμή, η οποία στην περίπτωση των δικών μας LED και σύμφωνα με τον κατασκευαστή, είναι τα 20 mA.

Για να μπορέσουμε αφενός να κάνουμε το LED να φωτοβολήσει, και αφετέρου να μην το καταστρέψουμε, να μην το «κάψουμε» δηλαδή, θα πρέπει να τηρήσουμε τις παραπάνω προϋποθέσεις. Τι σημαίνει αυτό; Αν συνδέσουμε ένα LED σε μια μπαταρία τάσης 1,5V, με τη σωστή πολικότητα εννοείται, η τάση αυτή δεν επαρκεί και το LED δεν θα φωτοβολήσει (δεν θα ανάψει). Αν σε μια μπαταρία των 9V συνδέσουμε ένα LED (με την ορθή πολικότητα), τότε το LED θα καταστραφεί (θα ανάψει στιγμιαία και θα μυρίσει καμένο). Σε αυτή την περίπτωση, έχουμε ξεπεράσει το μέγιστο όριο της έντασης του ρεύματος που θα έπρεπε να διαρρέει το LED. Άρα, πρέπει με κάποιον τρόπο να περιορίσουμε αυτό το ρεύμα, και αυτό θα το κάνουμε χρησιμοποιώντας τους «αντιστάτες», ή αλλιώς «αντιστάσεις». Στα διδακτικά σενάρια, όταν έχουμε μπαταρία 9V, ο αντιστάτης που χρησιμοποιούμε είναι 470Ω, ενώ στις εφαρμογές με τον Arduino χρησιμοποιούμε αντιστάτη 220Ω, ώστε να μην υπερβούμε τα 20 mA και καταστρέψουμε το LED.

Αντιστάτες ή Αντιστάσεις

Με τον όρο «ηλεκτρική αντίσταση» εκφράζουμε το φυσικό μέγεθος που καθορίζει την αντίσταση που προβάλλει ένα στοιχείο του ηλεκτρικού κυκλώματος στη ροή του ηλεκτρικού ρεύματος, ενώ με τον όρο «αντιστάτης» εννοούμε το ηλεκτρονικό εξάρτημα που χρησιμοποιούμε στα ηλεκτρονικά κυκλώματα για να περιορίσουμε την ένταση του ρεύματος. Παρόλα αυτά, έχει επικρατήσει τους «αντιστάτες» να τους ονομάζουμε «αντιστάσεις». Μονάδα μέτρησης των αντιστατών είναι το Ω (Ohm) και τον συναντάμε σε μεγάλη ποικιλία μεγεθών, συνήθως από 10Ω έως 10M.

Οι αντιστάτες στην επιφάνειά τους έχουν χρωματιστούς δακτυλίους, οι οποίοι μας βοηθούν να βρούμε το μέγεθός τους με βάση τον χρωματικό κώδικα, έναν πίνακα που αντιστοιχίζει το χρώμα του πρώτου δακτυλίου με το ψηφίο μιας δεκάδας, το χρώμα του δεύτερου δακτυλίου με το ψηφίο μιας μονάδας, και το χρώμα του τρίτου δακτυλίου με έναν πολλαπλασιαστή. Το μέγεθος του αντιστάτη είναι το γινόμενο του αθροίσματος των δεκάδων και των μονάδων που διαβάσαμε, με τον πολλαπλασιαστή.



Ο Χρωματικός Κώδικας

Ένας άλλος τρόπος να βρούμε το μέγεθος ενός αντιστάτη είναι με τη χρήση του πολυμέτρου, και αυτό θα χρησιμοποιήσουμε στα πρώτα διδακτικά σεναρία.

Στα ηλεκτρονικά κυκλώματα συναντούμε αντιστάτες μεταλλικής μεμβράνης - Metal Film Resistors, που έχουν μπλε χρώμα, είτε αντιστάτες μεμβράνης άνθρακα - Carbon Film Resistors, που έχουν κίτρινο/καφέ ή ροζ χρώμα.



Αντίσταση μεταλλικής μεμβράνης (αριστερά) και μεμβράνης άνθρακα

Αισθητήρες

Αναφέρθηκε προηγουμένως ότι στον Arduino μπορούμε να συνδέσουμε αισθητήρες (sensors) ή και ενεργοποιητές (actuators). Οι αισθητήρες είναι ηλεκτρονικά εξαρτήματα τα οποία είναι για τον Arduino ό,τι και οι αισθήσεις για τον άνθρωπο.

Οι αισθητήρες είναι συσκευές που ανιχνεύουν διάφορα φυσικά μεγέθη και δημιουργούν τα κατάλληλα ηλεκτρικά σήματα τα οποία μπορούν να “διαβαστούν” από τον Arduino. Υπάρχει μια πληθώρα αισθητήρων: θερμοκρασίας, υγρασίας, πίεσης, αερίων, κίνησης, απόστασης, κλίσης, μαγνητικού πεδίου, μετάλλων, υπερύθρων, φωτός, δύναμης, επιτάχυνσης, επαφής, βιομετρικοί και πολλοί άλλοι.

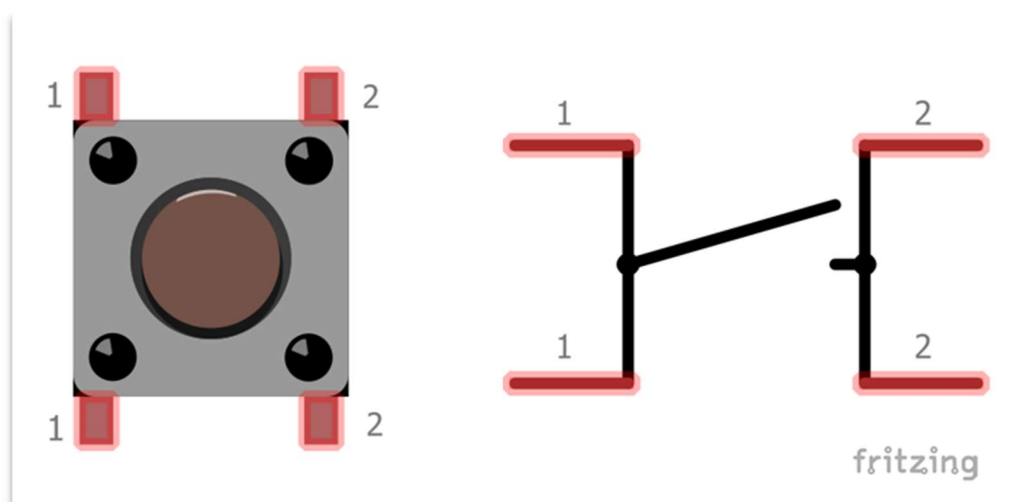


Αισθητήρες

Παρακάτω θα περιγράψουμε τους αισθητήρες και τους ενεργοποιητές που χρησιμοποιούμε στα κυκλώματα των διδακτικών σεναρίων.

Push Button ή Διακόπτης ή Κουμπίο (κοινώς κουμπί)

Το push button δεν είναι παρά ένας διακόπτης, με τον οποίο ανοίγουμε ή κλείνουμε ένα κύκλωμα, και διαθέτει 4 ακροδέκτες. Οι ακροδέκτες αυτοί είναι ανά 2 συνδεδεμένοι μεταξύ τους (οι αριστεροί ακροδέκτες μεταξύ τους, και οι δεξιοί μεταξύ τους), ενώ η διάταξη χωρίζεται σε δύο ανεξάρτητα κομμάτια (αριστερό - δεξί). Όταν πιέζουμε το κουμπί, κλείνει ο διακόπτης και συνδέεται το δεξί με το αριστερό του μέρος.

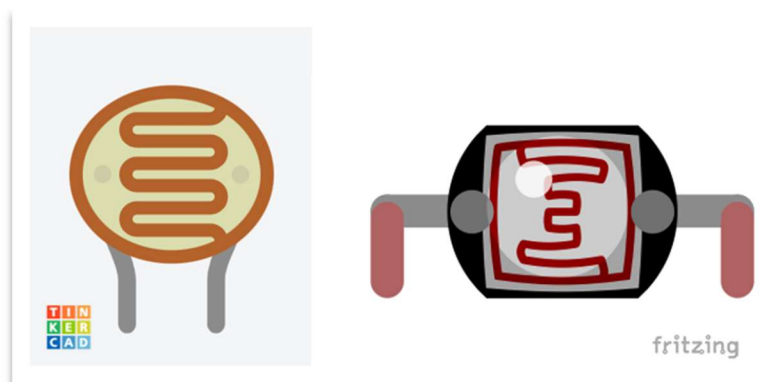


Push Button

Φωτοαντίσταση

Η φωτοευαίσθητη αντίσταση ή φωτοαντίσταση είναι μια αντίσταση το μέτρο της οποίας μεταβάλλεται ανάλογα με την ένταση του φωτός που προσπίπτει σε αυτή. Όταν το φως γίνεται εντονότερο, το μέτρο της αντίστασης γίνεται μικρότερο. Η φωτοαντίσταση κατασκευάζεται από φωτοευαίσθητο υλικό. Αυτό σημαίνει ότι όταν πέφτει φως πάνω στα άτομα αυτού του υλικού, η φωτεινή ενέργεια διεγείρει τα ηλεκτρόνια και κάποια από αυτά διαφεύγουν. Με τον τρόπο αυτό αυξάνεται ο αριθμός των ελεύθερων ηλεκτρονίων, άρα γίνεται ευκολότερη η διέλευση του ρεύματος, δηλαδή μικραίνει η τιμή της αντίστασης. Εντονότερο φως σημαίνει περισσότερη προσφερόμενη ενέργεια, περισσότερα ηλεκτρόνια που διαφεύγουν, ευκολότερη διέλευση ρεύματος και άρα μικρότερη αντίσταση. Αντίστροφα, η

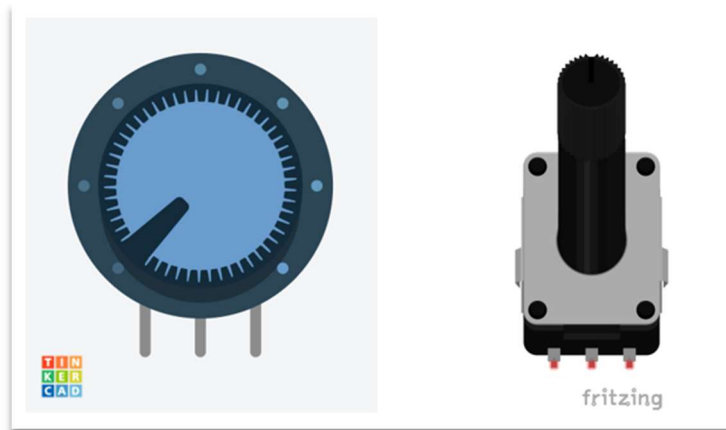
αντίσταση παίρνει τη μέγιστη τιμή της στο σκοτάδι. Στις εφαρμογές με τον Arduino συνδυάζεται σε κύκλωμα με αντίσταση 10KΩ.



Φωτοαντίσταση

Ποτενσιόμετρο

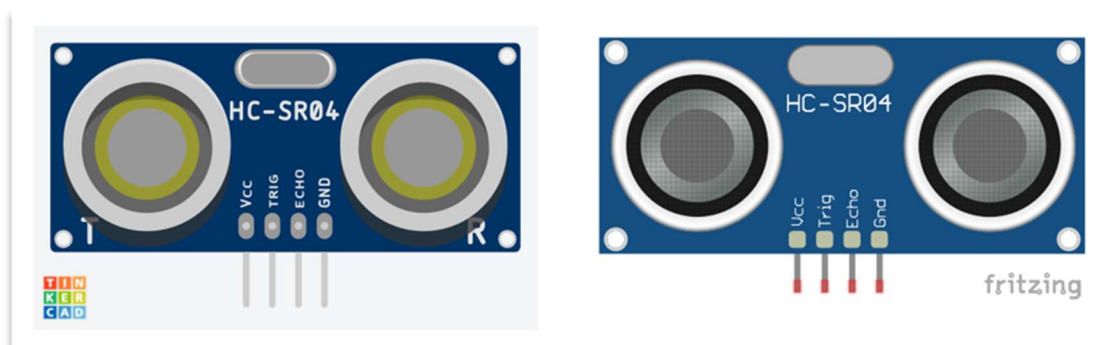
Το ποτενσιόμετρο είναι μια μεταβλητή αντίσταση, ή για την ακρίβεια ένας διαιρέτης τάσης, με τον οποίο από μια σταθερή τάση μπορούμε να πάρουμε ενδιάμεσες τιμές από 0V μέχρι την τάση εισόδου. Όπως φαίνεται στην εικόνα, το περιστροφικό ποτενσιόμετρο διαθέτει έναν περιστρεφόμενο ρυθμιστή και 3 ακροδέκτες. Εσωτερικά, οι 2 ακραίοι ακροδέκτες συνδέονται στα άκρα μίας ωμικής αντίστασης. Ο μεσαίος ακροδέκτης συνδέεται με μία κινητή επαφή, η οποία «τρέχει» πάνω στην αντίσταση και η θέση της καθορίζεται από τον περιστρεφόμενο ρυθμιστή. Όταν συνδέουμε τους ακριανούς ακροδέκτες, τον ένα στη γείωση GND και τον άλλο στην τροφοδοσία 5V (η πολικότητα δεν είναι σημαντική), το ποτενσιόμετρο μπορεί να λειτουργήσει ως ένας μεταβλητός διαιρέτης τάσης, βγάζοντας στο μεσαίο ακροδέκτη τάση από 0V έως 5V, ανάλογα με τη θέση του ρυθμιστή. Για να διαβάσουμε την τάση από το ποτενσιόμετρο χρησιμοποιούμε μια από τις 6 αναλογικές εισόδους του Arduino, δηλαδή τα pin A0 ως A5. Αυτά μπορούν να δεχθούν τάση από 0 ως 5V και να δώσουν ως αποτέλεσμα τους αριθμούς 0 ως 1023 (το 1023 αντιστοιχεί σε τάση 5V που εφαρμόζουμε στο pin).



Ποτενσιόμετρο

Αισθητήρας Υπερήχων – Ultrasonic Sensor HC-SR04

Πρόκειται για έναν αισθητήρα ο οποίος χρησιμοποιείται για τη μέτρηση της απόστασης. Ο αισθητήρας υπερήχων διαθέτει έναν πομπό και έναν δέκτη υπερήχων, καθώς και τέσσερις ακροδέκτες. Η αρχή λειτουργίας του αισθητήρα αυτού είναι ίδια με αυτή που έχουν οι νυχτερίδες: ο πομπός στέλνει υπερήχους, και εάν αυτοί συναντήσουν κάποιο εμπόδιο, τότε επιστρέφουν πίσω και τους λαμβάνει ο δέκτης. Ο χρόνος που διανύουν οι υπέρηχοι μέχρι να συναντήσουν το εμπόδιο και να επιστρέψουν πίσω στον δέκτη, μέσα από εντολές του Arduino μετατρέπονται σε μονάδες μήκους, και έτσι μπορούμε να έχουμε την απόσταση του εμποδίου από τον αισθητήρα. Η απόσταση που μπορεί να μετρηθεί κυμαίνεται μεταξύ 2 cm και 400 cm.



Αισθητήρας Υπερήχων – Ultrasonic Sensor

Αισθητήρας Υγρασίας Εδάφους (Χωρητικός)

Ο χωρητικός αισθητήρας υγρασίας εδάφους (Capacitive Soil Moisture) λειτουργεί σαν πυκνωτής, και μετρώντας πόσο γρήγορα (ή αργά) φορτίζει, μπορούμε να προσδιορίσουμε πόσο υγρό είναι το έδαφος. Το νερό στο έδαφος αλλάζει την ικανότητα του πυκνωτή. Η χωρητικότητά του, και επομένως ο ρυθμός φόρτισής του, αλλάζει ανάλογα με το πόσο νερό υπάρχει γύρω του.

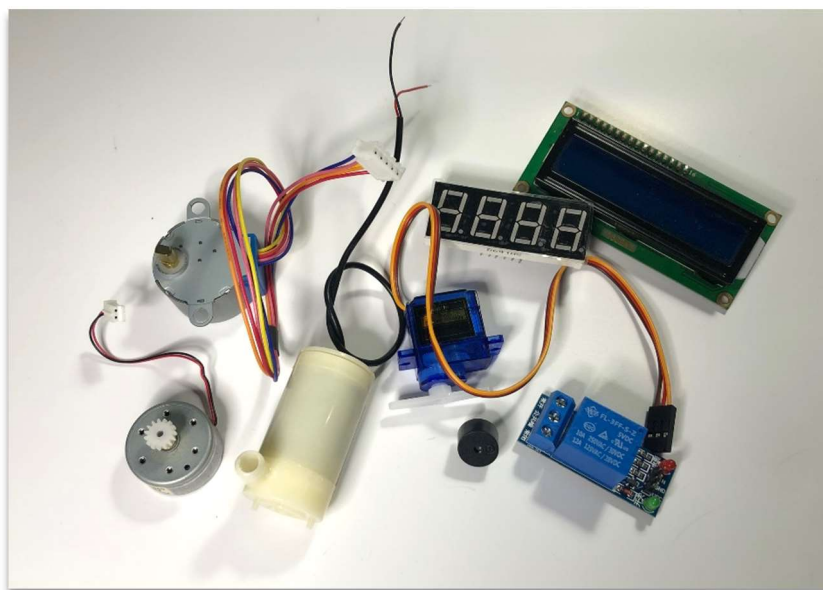
Καθώς το έδαφος γίνεται πιο υγρό, η τιμή εξόδου μειώνεται και καθώς γίνεται ξηρότερο, η τιμή εξόδου αυξάνεται. Όταν τροφοδοτείται στα 5V, η έξοδος κυμαίνεται από περίπου 1,5V (για υγρό έδαφος) έως 3V (για στεγνό χώμα). Αυτό μεταφράζεται ως εξής: η τιμή του αισθητήρα στην αναλογική είσοδο για στεγνό χώμα είναι 380, για έδαφος που μόλις ποτίστηκε η τιμή του είναι περίπου 280, ενώ οι ενδιάμεσες τιμές αφορούν στην ιδανική υγρασία του εδάφους.



Αισθητήρας Υγρασίας Εδάφους

Ενεργοποιητές

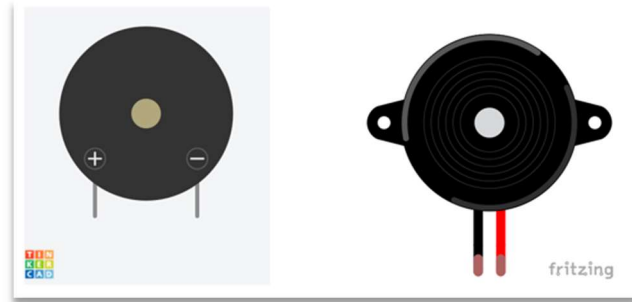
Οι ενεργοποιητές είναι συσκευές που μετατρέπουν ένα σήμα εντολής ελεγκτή σε μια αλλαγή μιας φυσικής παραμέτρου. Η αλλαγή είναι συνήθως μηχανική (π.χ. θέση ή ταχύτητα).



Συσκευές εξόδου / ενεργοποιητές

Buzzer

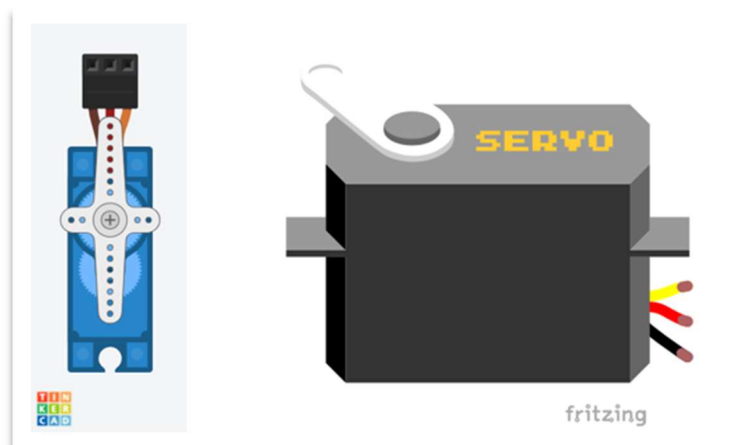
Τα buzzer είναι συσκευές που παράγουν ηχητικά σήματα. Υπάρχουν δύο κατηγορίες buzzer, τα active και τα passive. Τα active buzzer διαθέτουν εσωτερικό ταλαντωτή και όταν τροφοδοτούνται με συνεχή τάση, παρέχουν ένα ηχητικό σήμα συγκεκριμένης συχνότητας. Αντίθετα, τα passive buzzer δεν διαθέτουν εσωτερικό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



Buzzer

Σερβοκινητήρας ή Servo

Ο σερβοκινητήρας ή Servo είναι μια συσκευή που περιλαμβάνει έναν κινητήρα DC, ένα σύστημα γραναζιών, ένα ποτενσιόμετρο, ένα κύκλωμα ελέγχου και έναν άξονα εξόδου. Η χρήση του ποτενσιόμετρου και του κυκλώματος ελέγχου, επιτρέπουν την περιστροφή του άξονα σε συγκεκριμένη γωνία. Ο ορισμός της γωνιακής θέσης του άξονα γίνεται με ένα σήμα PWM, το οποίο δίνεται ως είσοδος στο servo. Όταν ολοκληρωθεί η περιστροφή του άξονα, το σήμα PWM παραμένει ενεργό, διατηρώντας τον στην επιθυμητή γωνία. Τα τυπικά servo υποστηρίζουν γωνία περιστροφής 180°, αλλά υπάρχουν και servo με γωνία περιστροφής 360°.



Servo

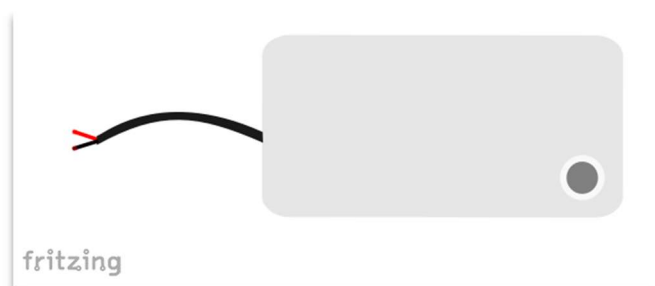
Αντλία Νερού

Στα διδακτικά σενάρια χρησιμοποιούμε εμβαπτίζόμενη αντλία νερού χαμηλής τάσης 3V-5V, η οποία τροφοδοτείται απευθείας από τον Arduino. Ο Arduino ελέγχει την αντλία νερού χρησιμοποιώντας ρελέ, οδηγούς κινητήρων ή άλλα κατάλληλα εξαρτήματα. Όταν ο κώδικάς μας υποδεικνύει μέσω κάποιου αισθητήρα ότι η αντλία πρέπει να ενεργοποιηθεί, ο Arduino στέλνει ένα σήμα στο ρελέ για να ενεργοποιηθεί την αντλία. Ομοίως, όταν η αντλία πρέπει να απενεργοποιηθεί, το Arduino στέλνει σήμα στο ρελέ για απενεργοποίηση της αντλίας.

Η αντλία αποτελείται από έναν κινητήρα DC, κλεισμένο στεγανά σε ένα κέλυφος, ο άξονας του οποίου, όταν τροφοδοτείται με τάση, περιστρέφει μια φτερωτή. Στον θάλαμο που βρίσκεται η φτερωτή, εισέρχεται νερό από μια είσοδο, και η περιστρεφόμενη φτερωτή το στέλνει με πίεση στην έξοδο της αντλίας, όπου συνδέουμε έναν εύκαμπτο σωλήνα.

Ο κινητήρας DC έχει έναν άξονα, και όσο τροφοδοτείται με τάση ο κινητήρας, ο άξονας περιστρέφεται σε μια φορά περιστροφής. Αν αλλάξουμε την πολικότητα στους ακροδέκτες του κινητήρα, η φορά περιστροφής του άξονα αντιστρέφεται. Επίσης, όταν μεταβάλλουμε την τάση τροφοδοσίας, μεταβάλλεται και η ταχύτητα περιστροφής του άξονα.

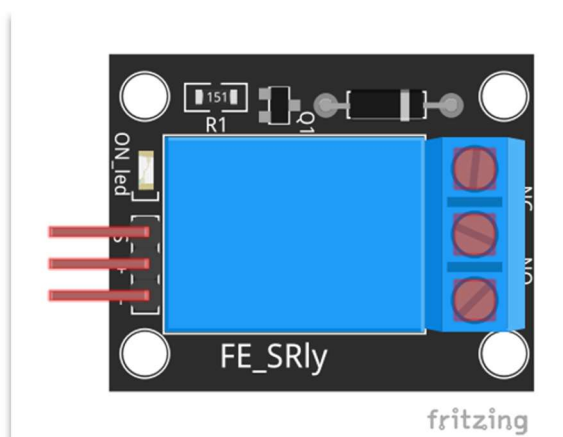
Η εμβαπτίζόμενη αντλία είναι βυθισμένη στο νερό, και είναι σχεδιασμένη για να λειτουργεί σε αυτή τη συνθήκη, μέσα στο νερό, και να ψύχεται από αυτό. Αν τη λειτουργήσουμε σε στεγνό περιβάλλον, είναι πολύ πιθανό να υπερθερμανθεί και να καταστραφεί.



Αντλία Νερού

Ρελέ

Οι ηλεκτρομηχανικοί διακόπτες, γνωστοί σαν ηλεκτρονόμοι ή ρελέ, χρησιμοποιούνται ευρύτατα σε όλες τις ηλεκτρικές και βιομηχανικές εγκαταστάσεις. Η αρχή λειτουργίας του ηλεκτρονόμου είναι απλή: ένας ηλεκτρομαγνήτης, δηλαδή ένας σιδερένιος πυρήνας με ένα ηλεκτρικό πηνίο τυλιγμένο γύρω του, έλκει τον οπλισμό του, δηλαδή ένα κινητό σιδερένιο μοχλό, όταν περνάει ηλεκτρικό ρεύμα μέσα από το πηνίο. Μόλις διακοπεί το ηλεκτρικό ρεύμα, ο μαγνήτης παύει να έλκει τον οπλισμό του, και ένα ελατήριο επαναφέρει τον τελευταίο στην αρχική του θέση, μακριά από τον σιδερένιο πυρήνα. Συνδέουμε τον οπλισμό μ' ένα διακόπτη, και καθώς ο οπλισμός κινείται υπό την επίδραση του ηλεκτρομαγνήτη, αυτός κάνει το διακόπτη να ανοιγοκλείνει. Επίσης, έχει τη δυνατότητα να ελέγχει συσκευές οι οποίες έχουν ανάγκη υψηλότερης τάσης ή έντασης ρεύματος από αυτά που παρέχει ο Arduino.



Ηλεκτρονόμος (Ρελέ)

Εκπαιδευτικό Υλικό με Arduino

Βασικός σκοπός εξ αρχής ήταν να ξεφύγει η διδασκαλία του Arduino από τα όρια του κλασικού μαθήματος, και να γίνει η μάθηση διασκεδαστική, οι μαθητές/τριες να έχουν το περιθώριο να παίζουν με τα υλικά, αφού όμως πρώτα μάθουν τη σωστή χρήση και συνδεσμολογία τους. Και κυρίως να βγάλουν μόνοι/ες τους κάποια συμπεράσματα με βιωματικό τρόπο, να κατακτήσουν τη γνώση, και όχι να τους δοθεί έτοιμη, απλά και μόνο για να εξεταστούν σε αυτή. Άλλωστε, το μάθημα της Τεχνολογίας αυτό έχει ως σκοπό, τα παιδιά να έρθουν σε επαφή με την τεχνολογία, να τη γνωρίσουν, να την κατανοήσουν. Και η διδασκαλία STEAM αποτελεί την κορυφαία επιλογή για την γνωριμία και επαφή με την τεχνολογία, συνδυάζοντας τους τομείς της επιστήμης, της μηχανικής, των μαθηματικών. Οι εφαρμογές με τον Arduino είναι αμέτρητες, έχει γίνει επιλογή κάποιων απλών και αντιπροσωπευτικών, ώστε να έρθουν σε επαφή τα παιδιά με τον προγραμματισμό και την υπολογιστική σκέψη, και μετά το πέρας των διδακτικών σεναρίων, οι ομάδες να υλοποιήσουν μια κατασκευή στην οποία θα γίνεται χρήση του Arduino με τους αντίστοιχους αισθητήρες και ενεργοποιητές.

Η μάθηση είναι μια δυναμική διαδικασία, δεν μπορούμε να είμαστε σίγουροι εκ των προτέρων ότι μέσα σε 2 διδακτικές ώρες θα διδάξουμε ένα διδακτικό σενάριο. Μπορεί να είναι εφικτό, μπορεί και όχι. Κάποιες ομάδες θα προχωρούν με το ρυθμό που έχουμε υπολογίσει, κάποιες άλλες όμως ίσως να χρειάζονται περισσότερο χρόνο. Αν θέλουμε να κερδίσουμε όλα τα παιδιά, ανεξάρτητα από τις δυνατότητές τους, πρέπει να περιμένουμε όλες τις ομάδες να ολοκληρώσουν. Και οι ομάδες που είναι μπροστά, θα έχουν χρόνο να πειραματιστούν με τα υλικά τους στο τέλος του διδακτικού δίωρου, δίνοντάς τους επιπλέον λαμπάκια και αντιστάσεις, ή υλικά για τα οποία έχουν μάθει πώς χρησιμοποιούνται.

Τα σενάρια που ακολουθούν διδάχτηκαν στην Α' τάξη ΕΠΑΛ, στα πλαίσια του μαθήματος «Ερευνητική Εργασία στην Τεχνολογία (EET)», σε ένα τμήμα των 20 μαθητών/τριών. Σύμφωνα με το αναλυτικό πρόγραμμα, το τμήμα χωρίζεται στη μέση και το πρώτο μισό τμήμα διδάσκεται EET, ενώ στο δεύτερο μισό διδάσκεται η Πληροφορική, για ένα συνεχόμενο δίωρο κάθε εβδομάδα. Το αντίστροφο συμβαίνει για ένα ακόμη δίωρο.

Η κάθε ομάδα των 10 μαθητών/τριών χωρίστηκε σε 5 ομάδες των 2 ατόμων, όπως και ιδανικά επιθυμούσα. Κατά τη διδασκαλία της πρώτης δεκάδας γίνονταν κάποιες παρατηρήσεις, κυρίως σχετικά με τη διαχείριση του εξοπλισμού, οι οποίες και ενσωματώθηκαν εκ των υστέρων στα διδακτικά σενάρια που εφαρμόστηκαν στη δεύτερη δεκάδα, και είναι αυτά που αποτυπώνονται στην παρούσα εργασία. Ένα χαρακτηριστικό παράδειγμα της παραπάνω διατύπωσης είναι το εξής: ενώ είχα προγραμματίσει να κατασκευαστεί σε πρώτη φάση ένα κύκλωμα με μία αντίσταση και ένα led, να προγραμματίσουμε, να βγάλουμε συμπεράσματα και γενικότερα να γίνει μια συζήτηση σχετικά με τις δυνατότητες του Arduino και του προγραμματισμού του, και σε δεύτερη φάση να προστεθεί στο κύκλωμα το δεύτερο led με την αντίστασή του, πολλά από τα παιδιά, έχοντας στη διάθεσή τους τα υλικά και των δύο φάσεων, προτίμησαν να μην συμμετέχουν στην συζήτηση, αλλά να προσθέσουν στο κύκλωμα το επιπλέον led και την αντίσταση, και να εκφράσουν τις δικές τους απορίες, που όμως ούτως ή άλλως θα τις λύναμε στη δεύτερη φάση του σεναρίου. Για το λόγο αυτό αποφάσισα να δίνω στις ομάδες ένα αποθηκευτικό κουτάκι που περιέχει μόνο τα υλικά της πρώτης φάσης (υλικά – 1^ο στάδιο), ώστε και οι λιγότερο δυνατοί μαθητές να έχουν ολοκληρώσει την πρώτη φάση, και αν χρειάζεται να προχωρήσουμε σε επόμενη φάση, τότε να δίνω τα απαραίτητα επιπλέον υλικά.

Κι επειδή υπάρχουν παιδιά που θέλουν να κάνουν κάτι παραπάνω στο τέλος του μαθήματος, και εφόσον υπάρχει διαθέσιμος χρόνος - φροντίζω πάντα να υπάρχει ελεύθερος χρόνος στο τέλος, γι' αυτό τα διδακτικά σενάρια προχωρούν με μικρή ύλη – έχω επιπλέον υλικά στο δικό μου κουτάκι και αφήνω ελεύθερα να πάρουν ό,τι τους χρειάζεται, σχετικά πάντα με όσα έχουν ήδη διδαχθεί. Θεωρώ ότι αυτό το τελευταίο μέρος του μαθήματος, η ελεύθερη ανάπτυξη της περιέργειας των παιδιών, είναι και το πιο δημιουργικό για αυτά. Άλλωστε, η ικανοποίηση που παίρνουν καθρεφτίζεται στα πρόσωπά τους.

Ιδανικά, θα ήθελα τα διδακτικά σενάρια να υλοποιούνται σε ένα εργαστήριο πληροφορικής, να υπάρχουν ομάδες των 2 ατόμων σε έναν υπολογιστή, με εγκατεστημένο το Arduino IDE, να έχω τη δυνατότητα να προβάλλω σε μια οθόνη προβολής τα σχέδια που θέλω να υλοποιήσουν τα παιδιά. Επίσης η κάθε ομάδα να έχει έναν Arduino με τα απαραίτητα υλικά για να μπορούν να εξασκηθούν τα μέλη της στην κατασκευή των κυκλωμάτων.

Επειδή, όμως, αντικειμενικοί λόγοι δεν επέτρεψαν να έχω αυτή τη δυνατότητα, το μάθημα γίνεται στην αίθουσα, χωρίς υπολογιστή, και χωρίς άλλον εξοπλισμό. Επέλεξα, πριν ζητήσω την ανάθεση του μαθήματος, να προμηθευτώ τα υλικά που ήθελα να χρησιμοποιήσω, οπότε έχουμε 5 ομάδες των 2 ατόμων, 5 Arduino, τα υλικά που θα χρειαστεί η κάθε ομάδα για την υλοποίηση των εφαρμογών που περιγράφονται στα διδακτικά σενάρια, και να διαθέσω το προσωπικό μου laptop για να φορτώνουμε τον κώδικα στον Arduino. Η μη δυνατότητα χρήσης του εργαστηρίου πληροφορικής του σχολείου άλλαξε τον προγραμματισμό σχετικά με το να γράφει η κάθε ομάδα τον δικό της κώδικα (όπως θα ήθελα να γίνεται), και κατέληξα στο να δίνω κάθε φορά τον Arduino με τον κώδικα φορτωμένο από πριν, να τον μοιράζω εκτυπωμένο στις ομάδες, να τον γράφω στον πίνακα και να τον αναλύουμε, και αν κάποια ομάδα θέλει να διαφοροποιήσει κάτι, θα το τροποποιούμε, θα το φορτώνουμε στην πλακέτα και θα βλέπουμε αν δουλεύει όπως επιθυμούμε.

1^ο Διδακτικό Σενάριο: Θα καεί το led ή δεν θα καεί;

Στόχοι:

Οι μαθητές/τριες

- Να κατανοήσουν την αναγκαιότητα των αντιστάσεων
- Να γνωρίσουν τη χρήση του πολυμέτρου
- Να είναι σε θέση να μετρούν την τάση και την αντίσταση
- Να γνωρίσουν τη σωστή συνδεσμολογία ενός led σε ένα ηλεκτρικό κύκλωμα

Ρωτάμε τα παιδιά αν έχουν ακούσει για τον Arduino.

Δείχνουμε μια εφαρμογή με Arduino που έχουμε ήδη υλοποιήσει, την αυτόματη βρύση, που θα τη δούμε στο κεφάλαιο με τα ολοκληρωμένα projects. Κάνουμε επίδειξη της λειτουργίας της και ζητάμε να πει ο καθένας και η καθεμία τη γνώμη του/της σχετικά με το πώς φαντάζονται πως λειτουργεί. Μετά αναλύουμε τη λειτουργία της. Αναφερόμαστε στον αισθητήρα υπερήχων, και δείχνουμε την καρδιά του συστήματος, τον Arduino. Αναφέρουμε ότι στο τέλος των μαθημάτων θα μπορούν και τα παιδιά να κάνουν το ίδιο ή κάτι αντίστοιχο.

Υλικά ανά θρανίο

- 1 μπαταρία 9V
- 2 μπαταρίες AAA
- 2 led
- 2 καλώδια
- 3 αντιστάσεις διαφορετικές μεταξύ τους, η μία 470 Ω
- 1 πολύμετρο

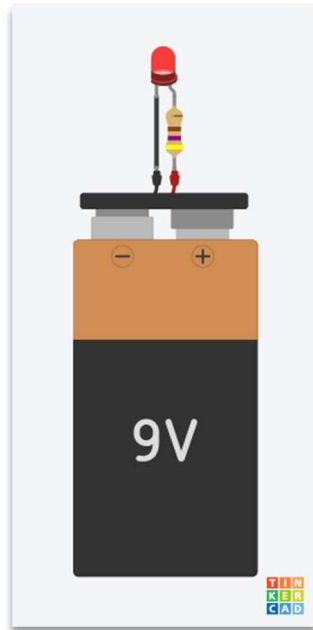


Εικ. 1.1 Υλικά

Δίνουμε σε κάθε θρανίο μία μπαταρία των 9V και δύο μπαταρίες AAA, καθώς και δύο LED και δύο καλώδια. Ζητάμε με αυτά τα υλικά να ανάψουν το LED αλλά να μην το κάψουν. Όσοι χρησιμοποιήσουν μία μπαταρία AAA δεν θα καταφέρουν να ανάψουν το LED, ενώ όσοι χρησιμοποιήσουν την 9V θα το κάψουν. Καταλήγουμε στο συμπέρασμα ότι το 1,5V είναι λίγο, ενώ τα 9V είναι πολλά. Χρησιμοποιούμε και το πολύμετρο για να μετρήσουμε την τάση της κάθε μπαταρίας. Προτείνουμε να μετρήσουν και τις δύο AAA σε σειριακή σύνδεση. Το πολύμετρο δείχνει 3V. Μετά από αυτό, δοκιμάζουν να συνδέσουν το LED σε αυτή την διάταξη των μπαταριών. Αν συνδέσουν το μακρύ ποδαράκι στο θετικό πόλο και το κοντό ποδαράκι στον αρνητικό πόλο, το LED ανάβει. Στην αντίθετη περίπτωση το LED δεν ανάβει.

Βγάζουμε το συμπέρασμα για τη σωστή συνδεσμολογία του LED. Ονομάζουμε τα ποδαράκια «ακροδέκτες», ο μακρύς ακροδέκτης είναι η «άνοδος», και συνδέεται στο «+» της μπαταρίας, ο κοντός ακροδέκτης είναι η «κάθοδος» και συνδέεται στο «-» της μπαταρίας. Μαζεύουμε τις μπαταρίες AAA. Σε αυτή τη φάση ζητάμε από τα παιδιά να σκεφτούν με ποιον τρόπο διαθέτοντας μόνο την μπαταρία των 9V, θα μπορούσαμε να ανάψουμε το LED. Μετά από συζήτηση καταλήγουμε στο συμπέρασμα ότι αυτό που χρειάζεται είναι να μειωθεί η τάση της μπαταρίας (στην πραγματικότητα αυτό που θέλουμε είναι να μειωθεί η ένταση του ρεύματος που διαρρέει το led, αλλά αρκούμαστε στην, έστω και όχι σωστά διατυπωμένη, αιτιολόγηση των παιδιών).

Αυτή είναι μια καλή ευκαιρία να εισάγουμε την έννοια του αντιστάτη, ή αλλιώς, της αντίστασης. Δίνουμε στα παιδιά να πιάσουν και να περιεργαστούν μερικές αντιστάσεις, διαφόρων μεγεθών. Αναρωτιόμαστε αν φαίνονται στα παιδιά ίδιες ή διαφορετικές μεταξύ τους. Αναφέρουμε ότι μπορούμε να διαβάσουμε το μέγεθός τους με τον χρωματικό κώδικα, δίνοντας και μια σύντομη περιγραφή του, και συνεχίζουμε με την δεύτερη επιλογή που έχουμε, να τις μετρήσουμε με το πολύμετρο. Εξηγούμε τον τρόπο με τον οποίο μετράμε το μέγεθος των αντιστάσεων με το πολύμετρο, και αφήνουμε τα παιδιά να μετρήσουν τις αντιστάσεις που τους δώσαμε.



Σχ.1.1

Σχεδιάζουμε στον πίνακα ένα απλό κύκλωμα με την μπαταρία, μια αντίσταση 470 Ω και το LED (Σχ. 1.1) και αφήνουμε να το υλοποιήσουν. Με αυτόν τον τρόπο το LED ανάβει χωρίς να υπάρχει ο κίνδυνος να καταστραφεί. Αν μείνει ελεύθερος χρόνος, διαθέτουμε επιπλέον αντιστάσεις, led και καλώδια και αφήνουμε τα παιδιά να πειραματιστούν. Τέλος, η κάθε δυάδα μαζεύει τα υλικά της στο κουτάκι της.

2^ο Διδακτικό Σενάριο: Μα τι είναι αυτό το «breadboard»;

Στόχοι:

Οι μαθητές/τριες

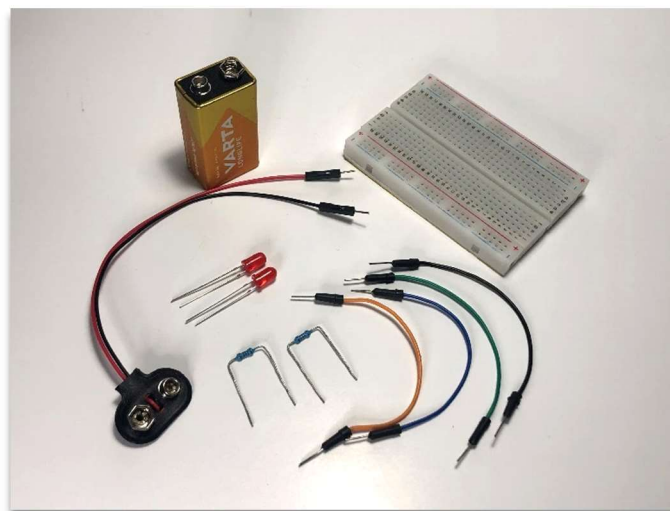
- Να γνωρίσουν τη χρήση του breadboard
- Να είναι σε θέση να διαβάσουν το σχέδιο ενός κυκλώματος και να το κατασκευάσουν
- Να κατανοήσουν τα αποτελέσματα της χρήσης διαφορετικών αντιστάσεων
- Να εντοπίσουν τα λάθη τους και να τα διορθώσουν

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 μπαταρία 9V και το αντίστοιχο κλιπ με 2 ακροδέκτες
- 1 breadboard
- 1 κόκκινο led
- 2 καλώδια
- 1 αντίσταση 470 Ω
- 1 πολύμετρο

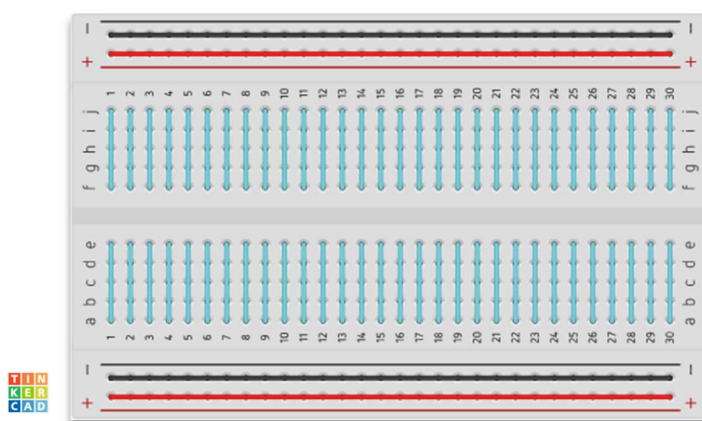
Υλικά ανά ομάδα – 2^ο στάδιο

- 1 κόκκινο led
- 2 καλώδια
- 1 αντίσταση 2 ΚΩ ή μεγαλύτερη



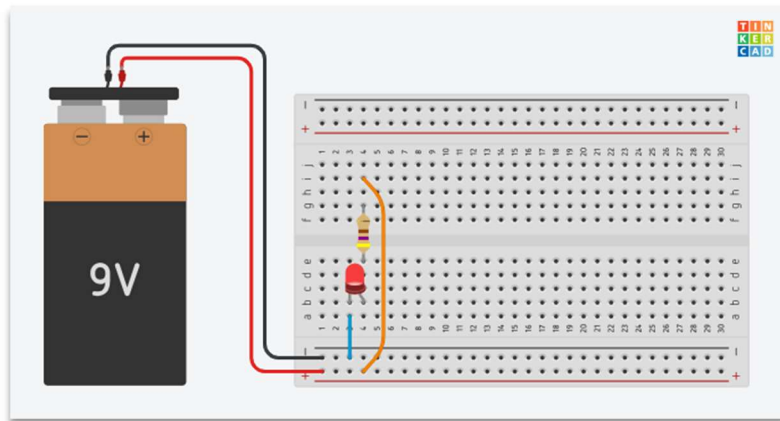
Εικ. 2.1 Υλικά

Ζητάμε από τα παιδιά να χωριστούν σε ομάδες των 2 ατόμων. Σε κάθε ομάδα δίνουμε ένα breadboard. Επισημαίνουμε ότι δεν χρειάζεται να δοκιμάσουμε την αντοχή τους, και ότι θα τα ξαναχρησιμοποιήσουμε. Εξηγούμε την χρηστικότητα του breadboard, ότι μας απαλλάσσει από την ανάγκη κολλήσεων για την κατασκευή ενός κυκλώματος, ότι η συνδεσμολογία μας δεν έχει μόνιμο και σταθερό χαρακτήρα, αλλά μπορούμε οποιαδήποτε στιγμή ή στο τέλος του μαθήματος να την αποσυναρμολογήσουμε. Για να μπορέσουμε όμως να το χρησιμοποιήσουμε, πρέπει πρώτα να καταλάβουν τα παιδιά την αόρατη σύνδεση μεταξύ των μικρών οπών. Σχεδιάζουμε στον πίνακα ένα breadboard, και ενώνουμε τις οπές που είναι συνδεδεμένες μεταξύ τους (Σχ.2.1). Δίνουμε και κάποια παραδείγματα σωστής και λανθασμένης χρήσης.



Σχ.2.1

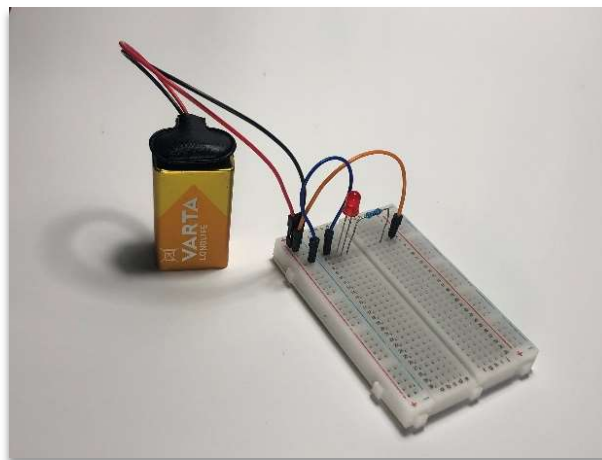
Κατόπιν μοιράζουμε σε κάθε ομάδα 2 καλώδια, ένα LED, μία μπαταρία των 9V και μία αντίσταση 470 Ω, καθώς και ένα σχέδιο (Σχ 2.2) της συνδεσμολογίας που θέλουμε να υλοποιήσουν. Δίνουμε αρκετό χρόνο να ολοκληρώσουν τα παιδιά την προσπάθειά τους, και ελέγχουμε αν έχουν φτάσει στο επιθυμητό αποτέλεσμα, να ανάψει δηλαδή το LED (Εικ.2.1).



Σχ. 2.2

Στην περίπτωση που κάποια ομάδα δεν πέτυχε να έχει αναμμένο LED, αναζητούμε την αιτία αυτού του αποτελέσματος. Τρία είναι τα επικρατέστερα σενάρια:

- έχει μπει ανάποδα το LED, στη θέση της καθόδου έχει μπει η άνοδος.
- κάποιο από τα καλώδια έχει μπει σε διπλανή οπή.
- έχει μπει ανάποδα η τροφοδοσία από την μπαταρία στο breadboard.

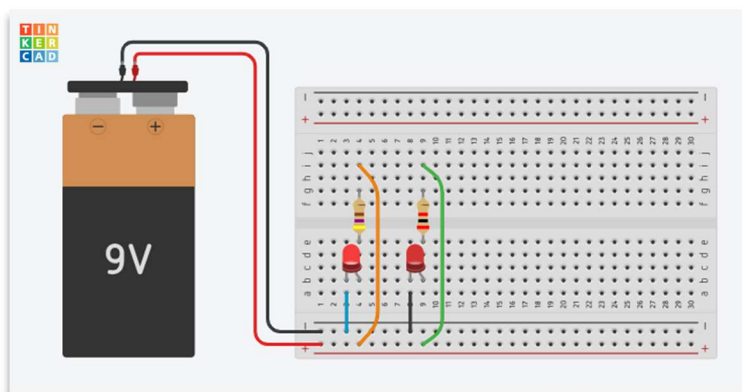


Εικ. 2.2 Υλοποίηση 1^{ου} σταδίου

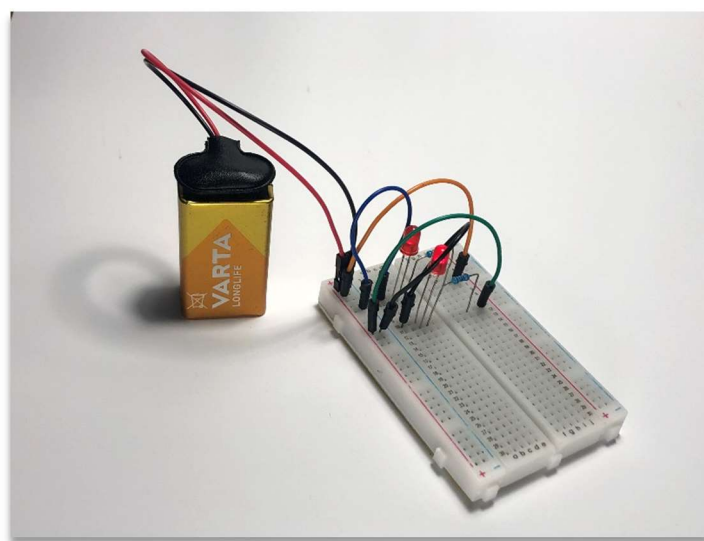
Αφήνουμε τα παιδιά να βρουν μόνα τους το λάθος, και να το διορθώσουν.

Δίνουμε σε κάθε ομάδα ακόμη μία αντίσταση, 2KΩ ή μεγαλύτερη, ένα LED και 2 καλώδια, και ζητάμε να τα προσθέσουν στο breadboard (Σχ.2.3) ώστε αυτή τη φορά να έχουν δύο αναμμένα LED. Στην περίπτωση που δεν ανάψει το δεύτερο λαμπάκι, αναζητούν μόνοι τους το αίτιο, και όταν το βρουν, κάνουν την απαραίτητη διόρθωση. Συγκρίνουμε την φωτεινότητα μεταξύ των δύο LED, και συζητάμε όλοι μαζί για να

εξετάσουμε το λόγο για τον οποίο συμβαίνει αυτό. Ακούγονται όλες οι γνώμες, και αφού μετρήσουν τα παιδιά τις αντιστάσεις με το πολύμετρο, καταλήγουν στο συμπέρασμα ότι οι διαφορετικές αντιστάσεις ευθύνονται για την διαφορετική φωτεινότητα, καθώς και ότι όσο μεγαλύτερη είναι η αντίσταση, τόσο λιγότερο φωτίζει το LED.



Σχ. 2.3



Εικ. 2.3 Υλοποίηση 2^{ου} σταδίου

Στην περίπτωση που μείνει ελεύθερος χρόνος, ενθαρρύνουμε τα παιδιά να πειραματιστούν και δίνουμε επιπλέον υλικά σε όσες ομάδες το επιθυμούν.

Τέλος, η κάθε ομάδα αποσυναρμολογεί το κύκλωμά της και αποθηκεύει τα υλικά στο κουτί της.

3^ο Διδακτικό Σενάριο: Χρήση Push Button και Buzzer

Στόχοι:

Οι μαθητές/τριες

- Να γνωρίσουν τη χρήση του push button
- Να αντιληφθούν την χρησιμότητα του προγραμματισμού
- Να γνωρίσουν το buzzer

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 μπαταρία 9V και το αντίστοιχο κλιπ με 2 ακροδέκτες
- 1 breadboard
- 2 κόκκινα LED
- 4 καλώδια
- 1 αντίσταση 470 Ω
- 1 αντίσταση 2 ΚΩ

Υλικά ανά ομάδα – 2^ο στάδιο

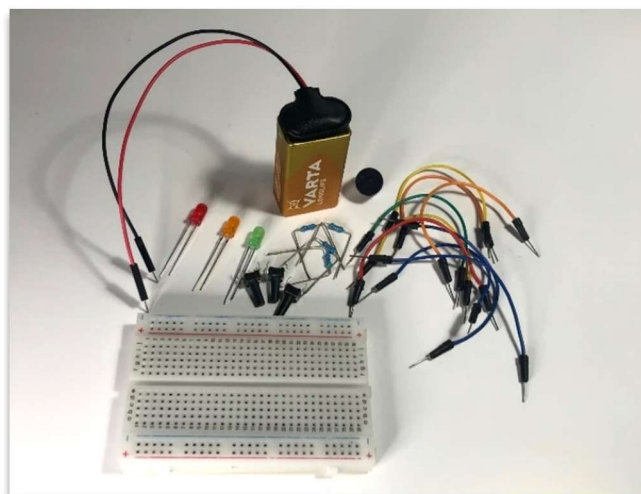
- 1 push button

Υλικά ανά ομάδα – 3^ο στάδιο

- 2 push button
- 5 καλώδια
- 2 αντιστάσεις 470 Ω

Υλικά ανά ομάδα – 4^ο στάδιο

- 1 buzzer
- 1 αντίσταση 220 Ω

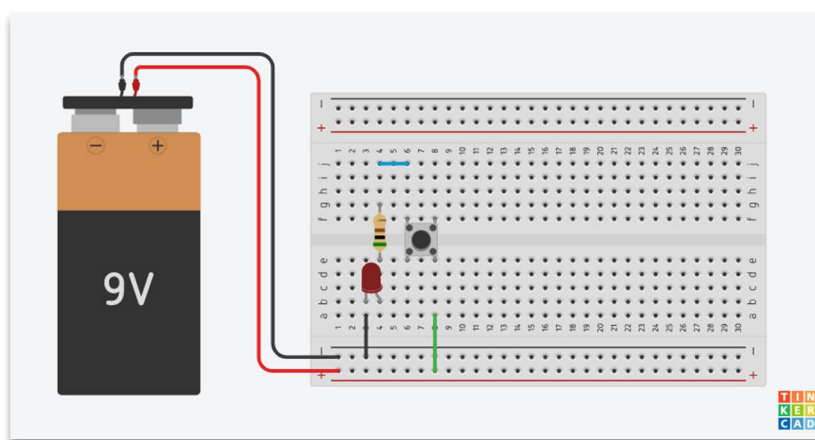


Εικ. 3.1 Υλικά

Μοιράζουμε από ένα κουτί σε κάθε ομάδα, το οποίο περιέχει τα υλικά του 1^{ου} σταδίου.

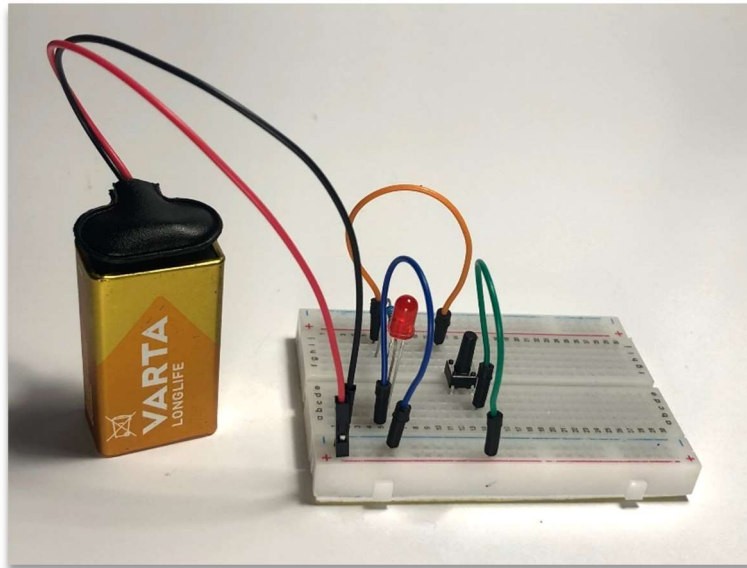
Δίνουμε στα παιδιά το σχέδιο του προηγούμενου μαθήματος και ζητάμε να κατασκευάσουν το κύκλωμα από την αρχή, κυρίως για να θυμηθούν τα συμπεράσματα στα οποία καταλήξαμε, αλλά και για να εξοικειωθούν με τα υλικά που διαθέτουμε.

Κατόπιν δίνουμε σε κάθε ομάδα ένα push button, ένα κουμπί δηλαδή, που λειτουργεί σαν διακόπτης. Εξηγούμε τον τρόπο με τον οποίο το συνδέουμε στο κύκλωμά μας, μοιράζουμε και από ένα σχέδιο της συνδεσμολογίας (Σχ. 3.1), και τα παιδιά προσθέτουν το κουμπί στο breadboard κάνοντας τις απαραίτητες τροποποιήσεις στο κύκλωμα που έχουν ήδη κατασκευάσει (ζητάμε να μας επιστρέψουν την αντίσταση των 2 ΚΩ). Ενδεχομένως η τοποθέτηση του push button στο breadboard την πρώτη φορά να δυσκολέψει τα παιδιά, οπότε δείχνουμε πώς το τοποθετούμε και βοηθάμε όπου χρειαστεί, για να μην καταστραφούν οι ακίδες του.



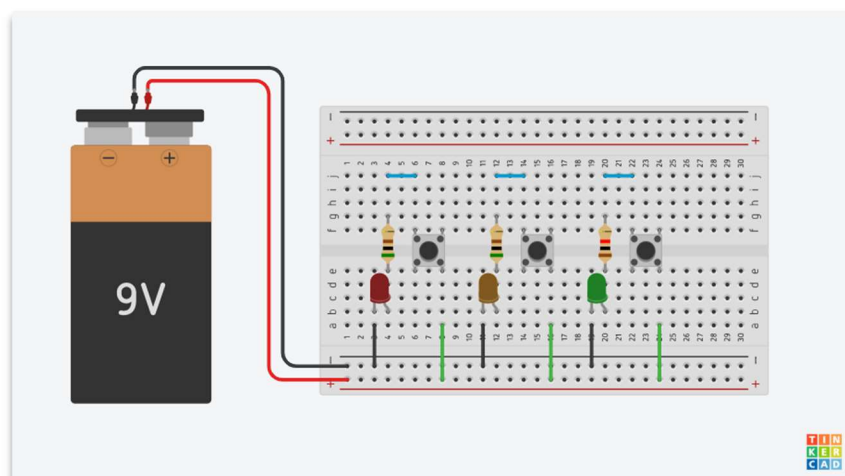
Σχ. 3.1

Αφού ολοκληρωθεί από όλες τις ομάδες οι συνδεσμολογία, βλέπουμε στην πράξη τη λειτουργία του κουμπιού, πατώντας το. Όπως κάθε φορά, σε περίπτωση που κάποιο λαμπάκι δεν ανάψει με το πάτημα του κουμπιού, τα παιδιά αναζητούν το σφάλμα, και προχωρούν στη διόρθωσή του.



Εικ. 3.2 Υλοποίηση 2^{ου} σταδίου

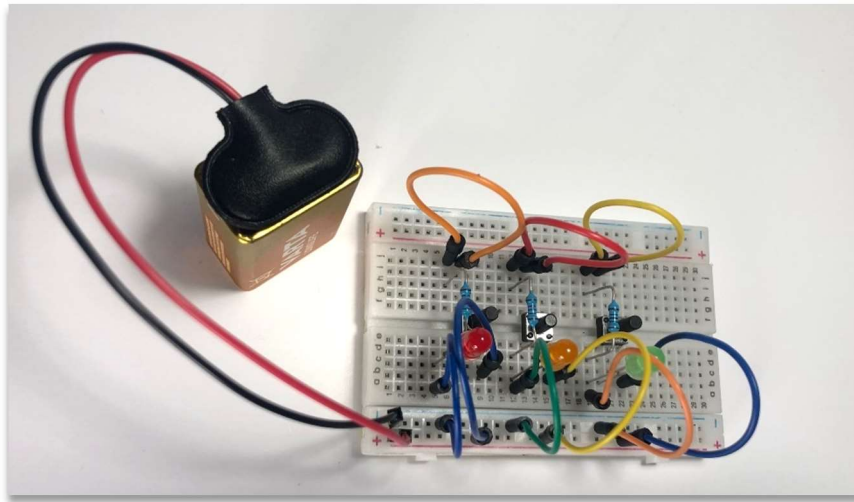
Συνεχίζουμε με την προσθήκη 2 ακόμη κουμπιών και ισάριθμων led και αντιστάσεων, σύμφωνα με το παρακάτω σχέδιο (Σχ. 3.2):



Σχ. 3.2

Τα LED έχουν χρώμα κόκκινο, πορτοκαλί και πράσινο. Το πιθανότερο είναι τα παιδιά να παρομοιάσουν αυτά τα χρώματα με τα φανάρια κυκλοφορίας. Σε αντίθετη περίπτωση, συζητάμε μαζί τους, αν τους θυμίζει κάτι αυτή η σειρά των χρωμάτων, και καταλήγουμε στο συμπέρασμα ότι πρόκειται για μικρογραφία ενός φαναριού κυκλοφορίας. Αφήνουμε χρόνο στις ομάδες να παίξουν με τα κουμπιά, και

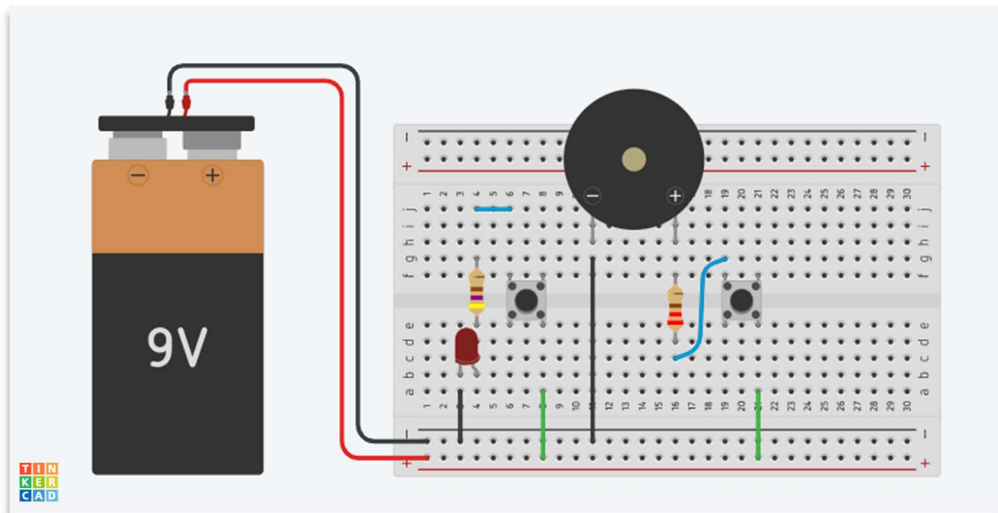
χειροκίνητα προφανώς, να προσομοιάσουν τη λειτουργία ενός φαναριού. Συζητάμε για τη λειτουργία των φαναριών κυκλοφορίας, αναρωτιόμαστε αν θα ήταν εφικτό να γίνεται χειροκίνητα το άναμμα και το σβήσιμο της κάθε λάμπας σε ένα πραγματικό φανάρι, και πόσο ασφαλές θα ήταν αυτό, και καταλήγουμε στο συμπέρασμα ότι είναι αδύνατο να λειτουργήσουν τα φανάρια χειροκίνητα, και ότι χρειάζεται κάποιος τρόπος ώστε να λειτουργούν προγραμματισμένα. Επειδή ήδη έχουμε δείξει την εφαρμογή με τον Arduino, τα παιδιά είναι αναμενόμενο να υποθέσουν ότι αυτόν τον προγραμματισμό μπορούμε να τον κάνουμε με την πλακέτα.



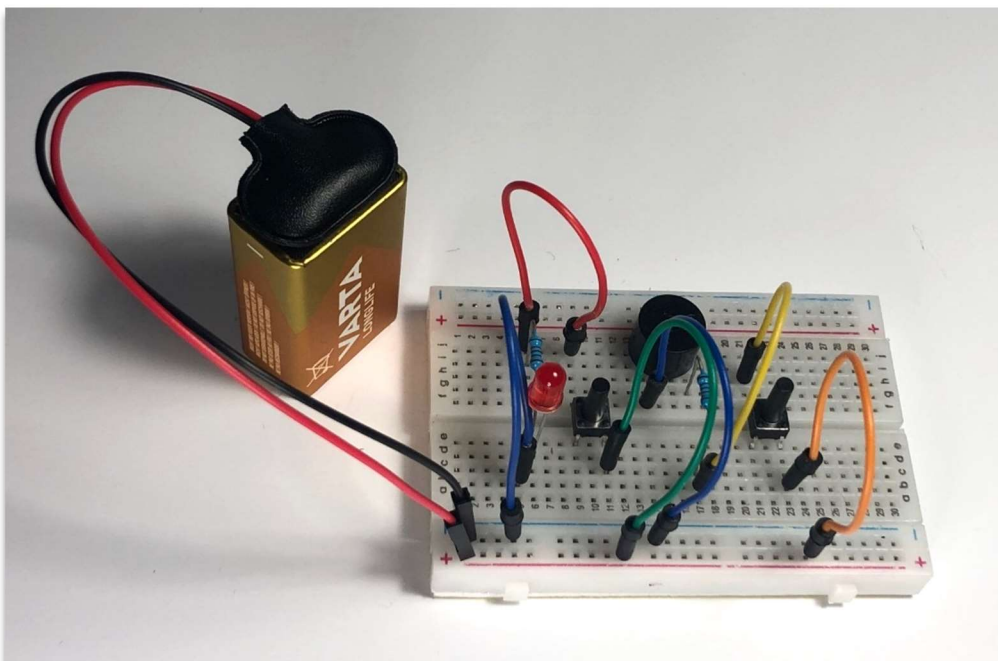
Εικ. 3.3 Υλοποίηση 3^{ου} σταδίου

Ενημερώνουμε ότι στο επόμενο μάθημα θα ασχοληθούμε με αυτό το αντικείμενο, πώς δηλαδή με τη χρήση του Arduino μπορούμε να προγραμματίσουμε το άναμμα και το σβήσιμο ενός ή περισσότερων LED.

Στη συνέχεια μοιράζουμε στις ομάδες από ένα buzzer, μια αντίσταση 470 Ω και ένα σχέδιο, και περιμένουμε να το υλοποιήσουν. Δεν δίνουμε καμία πληροφορία για το buzzer, παρά μόνο ότι έχει «+» και «-», και ότι το αντιμετωπίζουμε σαν ένα LED (Σχ. 3.3). Το τι κάνει, θα το ανακαλύψουν μόνοι τους.



Σχ. 3.3



Εικ. 3.4 Υλοποίηση 4^{ου} σταδίου

Αφού μετά τη συνδεσμολογία και το πάτημα του κουμπιού καταλάβουν τί είναι το buzzer, συζητάμε για τη χρήση του, σε ποιες εφαρμογές το συναντούμε, καθώς και σε ποιες εφαρμογές θα μπορούσαμε να το χρησιμοποιήσουμε. Αφήνουμε τον χρόνο που έχει μείνει ώστε να πειραματιστούν οι ομάδες με τα υλικά τους, δίνουμε επιπλέον λεντάκια, κουμπιά και αντιστάσεις σε όποια ομάδα επιθυμεί να κάνει κάτι πιο

περίπλοκο και στο τέλος γίνεται η αποσυναρμολόγηση των κυκλωμάτων και η αποθήκευση των υλικών στο κουτί της κάθε ομάδας.

4^ο Διδακτικό Σενάριο: Γνωριμία με τον Arduino

Στόχοι:

Οι μαθητές/τριες

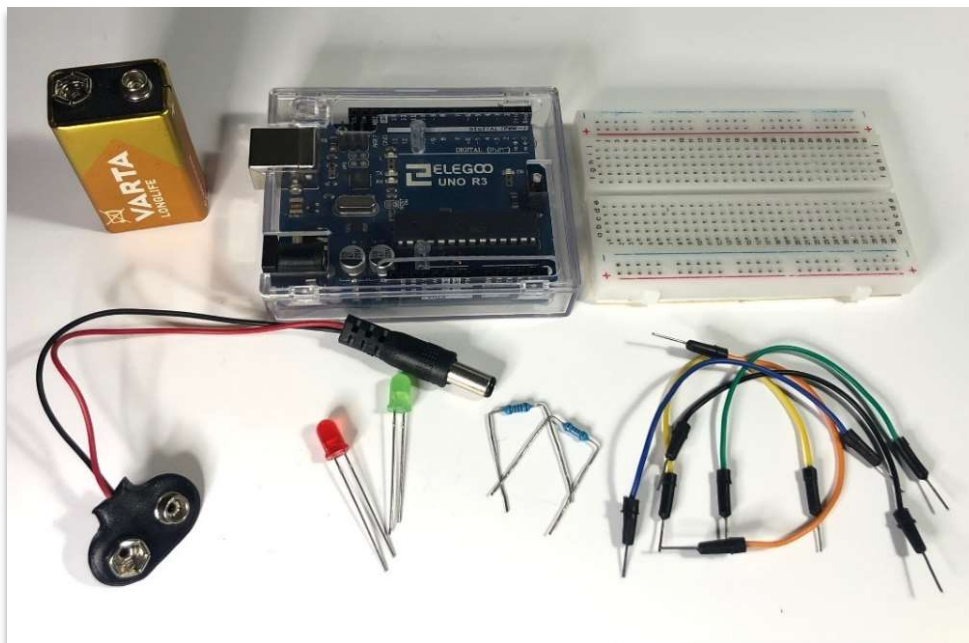
- Να γνωρίσουν τον Arduino
- Να κατανοήσουν τη χρησιμότητα των εισόδων και εξόδων του Arduino
- Να γνωρίσουν τις βασικές εντολές στον προγραμματισμό στο Arduino IDE
- Να είναι σε θέση να τροποποιήσουν έναν κώδικα

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 πράσινο led
- 3 καλώδια
- 1 αντίσταση 220 Ω

Υλικά ανά ομάδα – 2^ο στάδιο

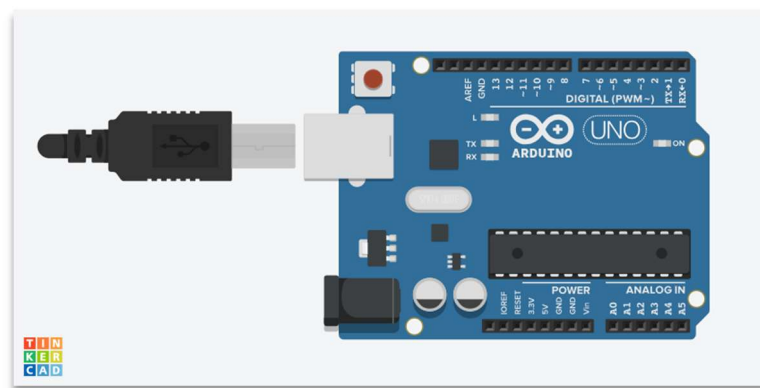
- 1 κόκκινο led
- 2 καλώδια
- 1 αντίσταση 220 Ω



Εικ. 4.1 Υλικά

Κάνουμε μια αναδρομή στο τελευταίο μάθημα, στο οποίο κάναμε χρήση push button, και επαναφέρουμε την συζήτηση στο συμπέρασμα στο οποίο είχαμε καταλήξει, ότι θα ήταν αδύνατη η σωστή κυκλοφορία εάν τα φανάρια τα άναβαν και τα έσβηναν άνθρωποι με τη χρήση διακοπών, και για το σκοπό αυτό χρειαζόμαστε κάτι που να προγραμματίζεται ώστε να εκτελεί αυτή την εργασία, επαναλαμβανόμενα και χωρίς λάθη.

Μοιράζουμε στην κάθε ομάδα μια φωτοτυπία (Σχ. 4.1) που απεικονίζει έναν Arduino σε μεγάλο μέγεθος, ώστε να δούμε τα στοιχεία που χρειάζεται να γνωρίζουμε και που θα χρησιμοποιήσουμε.



Σχ. 4.1 Arduino

Επισημαίνουμε ότι ο Arduino είναι μια πλακέτα που είναι σαν ένα λευκό άγραφο χαρτί. Αν δεν γράψουμε εμείς κάτι, θα παραμείνει λευκό. Και για να μπορεί κάποιος να διαβάσει αυτό που γράφουμε, θα πρέπει να γράψουμε με σωστή σύνταξη και ορθογραφία. Αντίστοιχα στον Arduino αυτό που γράφουμε το ονομάζουμε «κώδικα» και για να το εκτελέσει ο Arduino θα πρέπει να είναι σωστά γραμμένος, στην γλώσσα που «καταλαβαίνει» ο Arduino, και αυτή η γλώσσα ονομάζεται Wiring C. Αφού συντάξουμε τον κώδικα, με ένα κατάλληλο καλώδιο (στην περίπτωσή μας είναι μπλε) και μέσα από το πρόγραμμα Arduino IDE το φορτώνουμε στον Arduino.

Αναφερόμαστε στην τροφοδοσία του Arduino, ότι μπορεί να γίνει απευθείας από τον υπολογιστή, όταν έχουμε συνδεδεμένο το μπλε καλώδιο, όμως όταν θέλουμε να είμαστε ανεξάρτητοι από τον υπολογιστή μπορούμε να χρησιμοποιήσουμε ένα μετασχηματιστή 9V 2A DC, ή μία μπαταρία των 9V με το κατάλληλο βύσμα.

Προχωράμε με την τροφοδοσία του breadboard. Όπως στα προηγούμενα μαθήματα είχαμε μια μπαταρία των 9V, τώρα την τροφοδοσία αναλαμβάνει η πλακέτα μας, από τις θύρες 5V και GND που αντιστοιχούν στο «+» και στο «-» της μπαταρίας, με τη διαφορά ότι η τάση που δίνει ο Arduino είναι 5V, άρα οι αντιστάσεις που θα χρησιμοποιούμε για τα LED θα είναι 220 Ω.

Αναφερόμαστε στην περιοχή των αναλογικών εισόδων ANALOG IN, από A0 έως A5, τις οποίες θα χρησιμοποιήσουμε σε επόμενα μαθήματα, όταν θα κάνουμε χρήση αισθητήρων, και με απλό τρόπο δίνουμε να καταλάβουν τα παιδιά ότι σε αυτές τις εισόδους συνδέουμε αισθητήρες, οι οποίοι στον Arduino λειτουργούν όπως οι αισθήσεις στον άνθρωπο.

Συνεχίζουμε με τις εξόδους της πλακέτας, στην περιοχή DIGITAL (PWM-), όπου εκεί συνδέονται τα LED, οι κινητήρες, τα buzzer, και οτιδήποτε άλλο θελήσουμε να ενεργοποιήσουμε κατά την εκτέλεση του κώδικα.

Τα παραπάνω αρκούν για μια πρώτη γνωριμία με τον Arduino, οτιδήποτε επιπλέον χρειαστεί να γνωρίζουν τα παιδιά, θα αναφέρεται όταν είναι απαραίτητο.

Μετά την θεωρία, περνάμε στην πράξη: μοιράζουμε στις ομάδες τα κουτάκια με τα απαραίτητα υλικά.

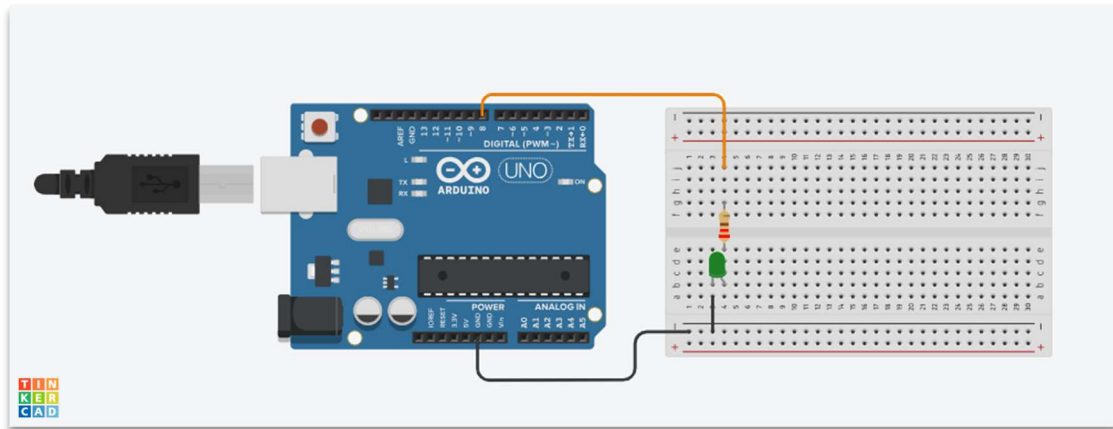
Όπως έχει αναφερθεί στην εισαγωγή, τα διδακτικά σενάρια προσαρμόζονται ανάλογα με τα μέσα που διαθέτουμε. Κατά συνέπεια, στους Arduino που μοιράζουμε, υπάρχει ήδη φορτωμένος ο παρακάτω κώδικας:

```
int ledGreen = 8; // πράσινο led στο pin 8

void setup() {
  pinMode(ledGreen, OUTPUT); // πράσινο led, ΕΞΟΔΟΣ
}

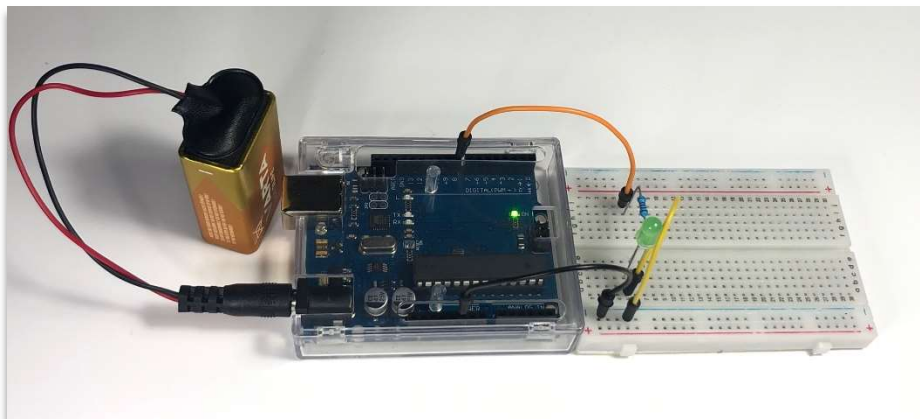
void loop() {
  digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
  delay(2000); // διάρκεια 2 sec
  digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
  delay(1000); //διάρκεια 1 sec
}
```

Δίνουμε στις ομάδες το κύκλωμα που θα υλοποιήσουν (Σχ. 4.2):



Σχ. 4.2

Επισημαίνουμε ότι σε αυτό το κύκλωμα δεν χρειάζεται να τροφοδοτήσουμε το breadboard με τάση 5 V από τον Arduino, επειδή αυτή τη δουλειά την κάνει το pin στο οποίο έχουμε τοποθετήσει το «+» του LED.



Εικ. 4.2 Υλοποίηση 1^{ου} σταδίου

Όταν φέρουν όλες οι ομάδες σε πέρας την αποστολή τους, γράφουμε τον κώδικα στον πίνακα, και τον εξηγούμε:

```
int ledGreen = 8; // πράσινο led στο pin 8
```

Int Εδώ αντιστοιχίζουμε ένα εξάρτημα (πχ. led) με ένα pin (πχ 8) στον Arduino, και ταυτόχρονα «βαφτίζουμε» με ένα μοναδικό και χαρακτηριστικό όνομα (ledGreen) το εξάρτημά μας

ledGreen Το μοναδικό όνομα που δίνουμε στο εξάρτημα, για να είναι ευκολότερο να γνωρίζουμε σε τι αναφερόμαστε όταν γράφουμε τον κώδικα. Εύκολα καταλαβαίνουμε ότι το όνομα ledGreen αναφέρεται σε ένα πράσινο led. Στην περίπτωση που έχουμε σε ένα κύκλωμα 2 πράσινα led, τότε μπορούμε το δεύτερο να το ονομάσουμε ledGreen2.

8 Σε αυτό το pin του Arduino έχουμε βάλει το led. Από εδώ και στο εξής, για το υπόλοιπο πρόγραμμα, όπου εμείς αναφέρουμε «ledGreen», ο Arduino «καταλαβαίνει» ότι εννοούμε το pin 8.

; Στο τέλος κάθε εντολής πρέπει να βάζουμε το ελληνικό ερωτηματικό «;» που σημαίνει ότι μετά από αυτό ο Arduino μπορεί να περάσει στην εκτέλεση της επόμενης εντολής. Αν παραλείψουμε να το βάλουμε, τότε φαίνεται πως στον κώδικα έχουμε συντακτικό σφάλμα, και δεν γίνεται να φορτωθεί στην πλακέτα μας.

// πράσινο led στο pin 8 Μπορούμε να γράφουμε σχόλια στον κώδικά μας, εισάγοντάς τα με διπλή κάθετη γραμμή «//», ο Arduino απλώς τα προσπερνάει

```
void setup() {  
  pinMode(ledGreen, OUTPUT); // πράσινο led, ΕΞΟΔΟΣ  
}
```

void setup() { Εδώ γίνεται η αρχικοποίηση (σετάρισμα) των δεδομένων, δηλώνουμε δηλαδή αν το εξάρτημα είναι είσοδος ή έξοδος.

{ Η αρχικοποίηση ακολουθείται από άγκιστρο

pinMode(ledGreen, OUTPUT); Η κατάσταση του ledGreen είναι ΕΞΟΔΟΣ (OUTPUT)

} Η αρχικοποίηση ολοκληρώνεται με το άγκιστρο

```

void loop() {
    digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
    delay(2000); // διάρκεια 2 sec
    digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
    delay(1000); //διάρκεια 1 sec
}

```

`void loop() {` Μετά την αρχικοποίηση, ακολουθεί ο βρόχος επανάληψης. Οποιαδήποτε εντολή γράψουμε μέσα στα άγκιστρα, εκτελείται από την αρχή έως το τέλος, και επαναλαμβάνεται συνέχεια, όσο τροφοδοτείται με ρεύμα ο Arduino

`digitalWrite(ledGreen, HIGH);` Ορίσαμε στην αρχικοποίηση πως το ledGreen είναι έξοδος (ψηφιακή έξοδος, για την ακρίβεια). Οπότε με αυτή την εντολή, θέλουμε ο Arduino να «γράψει» - ορίσει ότι το pin 8 θα διαρρέεται από ρεύμα (HIGH), και άρα το ledGreen θα είναι αναμμένο

`delay(2000);` Αν θέλουμε η ενέργεια του ledGreen να μην συμβαίνει για πάντα, αλλά να έχει μια ορισμένη διάρκεια, τότε χρησιμοποιούμε την εντολή «delay» και μέσα στην παρένθεση ορίζουμε τη διάρκεια αυτής της ενέργειας, μετρούμενη σε χιλιοστά του δευτερολέπτου (msec), οπότε αν θέλουμε το led να είναι αναμμένο για 2 δευτερόλεπτα, γράφουμε μέσα στην παρένθεση 2000.

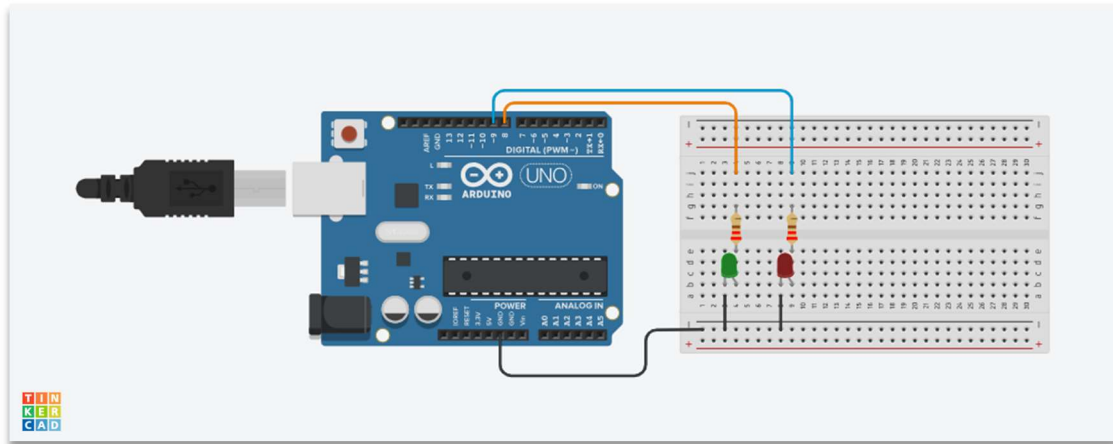
`digitalWrite(ledGreen, LOW);` Με αυτή την εντολή ζητάμε το pin 8 να μην διαρρέεται από ρεύμα, άρα το ledGreen να είναι σβηστό.

`delay(1000);` Όπως και προηγουμένως, θέλουμε η τελευταία ενέργεια να έχει και αυτή ορισμένη διάρκεια, έστω 1 δευτερόλεπτο.

`}` Φτάσαμε στο τέλος του βρόχου επανάληψης, οπότε χρειάζεται να το ορίσουμε στον κώδικά μας κλείνοντας το άγκιστρο. Τώρα το πρόγραμμα θα τρέξει από την αρχή.

Μετά από την επεξήγηση του κώδικα, δίνουμε σε κάθε ομάδα ένα ακόμη λαμπάκι, μία αντίσταση των 220 Ω και δύο καλώδια. Ζητάμε από τους μαθητές/τριες να τα τοποθετήσουν δίπλα από το 1ο λαμπάκι με την ίδια ακριβώς διάταξη (Σχ. 4.3) και τέλος, όλοι μαζί επιχειρούμε να κάνουμε κάποια τροποποίηση στον κώδικα, ώστε το δεύτερο λαμπάκι να ανάβει όταν σβήνει το πρώτο, και να σβήνει όταν ανάβει το

πρώτο. Στην πράξη, όπως έχουμε γραμμένο τον κώδικα στον πίνακα, προσθέτουμε καινούργιες γραμμές που αφορούν στο δεύτερο λαμπάκι.



Σχ. 4.3

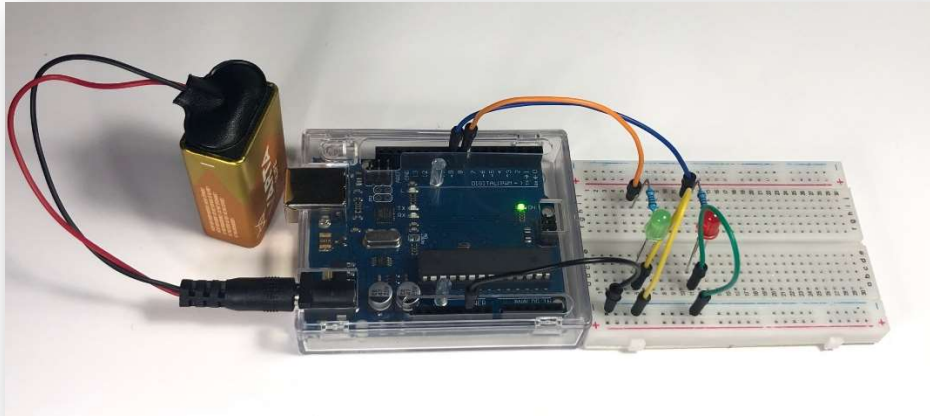
Ο νέος κώδικας είναι ο παρακάτω:

```
int ledGreen = 8; // πράσινο led στο pin 8
int ledRed = 9; // κόκκινο led στο pin 9

void setup() {
  pinMode(ledGreen, OUTPUT); // πράσινο led, ΕΞΟΔΟΣ
  pinMode(ledRed, OUTPUT); // κόκκινο led, ΕΞΟΔΟΣ
}

void loop() {
  digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
  digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
  delay(2000); // διάρκεια 2 sec

  digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
  digitalWrite(ledRed, HIGH); // κόκκινο led αναμμένο
  delay(1000); //διάρκεια 1 sec
}
```



Εικ. 4.3 Υλοποίηση 2^{ου} σταδίου

Φορτώνουμε τον τροποποιημένο κώδικα στο κάθε Arduino και βλέπουμε στην πράξη το αποτέλεσμα. Συζητάμε με τα παιδιά, διαπιστώνουμε ότι δεν είναι και τόσο δύσκολο να προγραμματίσουμε έτσι ώστε τα δύο λαμπάκια να ανάβουν και να σβήνουν με τον τρόπο που εμείς θέλουμε, ενημερώνουμε ότι στο επόμενο μάθημα θα ασχοληθούμε με τον προγραμματισμό φαναριών κυκλοφορίας και λύνουμε τυχόν απορίες που υπάρχουν. Αν υπάρχει ελεύθερος χρόνος ζητάμε από τα παιδιά να προτείνουν κάποιο δικό τους σενάριο και να σκεφτούν πώς θα τροποποιηθεί ο υπάρχων κώδικας. Γράφουμε στο IDE τον νέο κώδικα, τον φορτώνουμε στις πλακέτες και παρατηρούμε το αποτέλεσμα. Τέλος, τα παιδιά αποσυναρμολογούν τις κατασκευές τους και αποθηκεύουν τα υλικά τους.

5^ο Διδακτικό Σενάριο: Φανάρια Κυκλοφορίας

Στόχοι:

Οι μαθητές/τριες

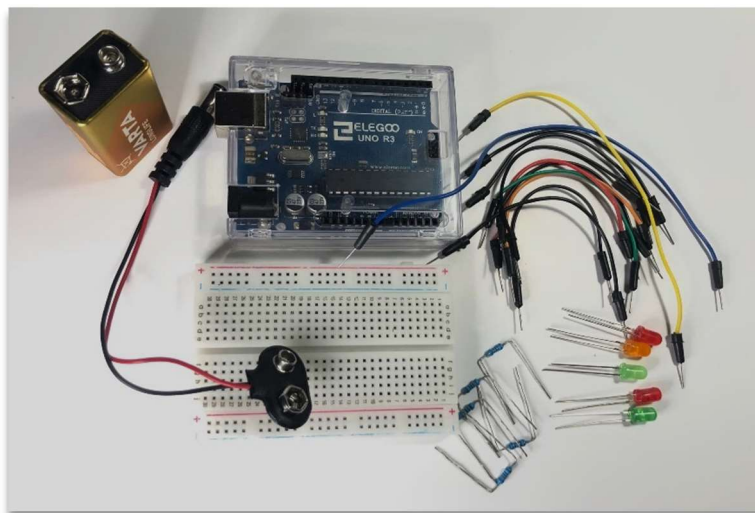
- Να είναι σε θέση να χρησιμοποιήσουν τις βασικές εντολές στον προγραμματισμό του Arduino
- Να μπορούν να συμπληρώσουν η/και να τροποποιήσουν έναν κώδικα σύμφωνα με τις απαιτήσεις ενός project
- Να προσομοιάσουν τα φανάρια κυκλοφορίας πεζών και οχημάτων

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 κόκκινο led
- 1 πορτοκαλί led
- 1 πράσινο led
- 7 καλώδια
- 3 αντιστάσεις 220 Ω

Υλικά ανά ομάδα – 2^ο στάδιο

- 1 κόκκινο led
- 1 πράσινο led
- 4 καλώδια
- 2 αντιστάσεις 220 Ω



Εικ. 5.1 Υλικά

Μοιράζουμε τα υλικά του 1^{ου} σταδίου στις ομάδες, και αρχίζουμε μια συζήτηση για τα φανάρια κυκλοφορίας, πώς είναι η λειτουργία τους, με ποια σειρά ανάβουν τα τρία χρώματα, ποιο έχει μεγαλύτερη διάρκεια σε έναν κεντρικό δρόμο. Αρχικά ασχολούμαστε μόνο με τα φανάρια κυκλοφορίας των οχημάτων. Δίνουμε στα παιδιά από μια εκτύπωση του τελευταίου κώδικα, ώστε όλοι μαζί να τον τροποποιήσουμε και να γράψουμε στον πίνακα τον κώδικα των φαναριών των οχημάτων, με τους χρόνους που θα είναι αναμμένο το κάθε χρώμα να τους αποφασίζουν τα παιδιά. Έχουμε ήδη έτοιμο ένα αρχείο στο Arduino IDE που αφορά στα φανάρια των οχημάτων, και το οποίο θα διορθώσουμε σύμφωνα με την επιθυμία των παιδιών σχετικά με τους χρόνους που έχουν προαναφερθεί.

Ο κώδικας για τα φανάρια κυκλοφορίας των οχημάτων:

```
int ledRed= 8; // κόκκινο led στο pin 8
int ledOrange = 9; // πορτοκαλί led στο pin 9
int ledGreen = 10; // πράσινο led στο pin 10

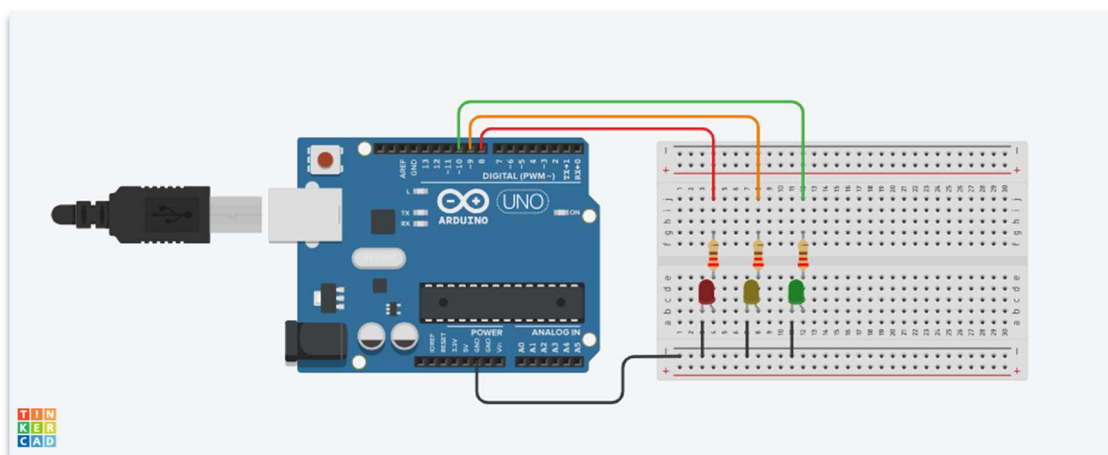
void setup() {
  pinMode(ledRed, OUTPUT); // κόκκινο led, ΕΞΟΔΟΣ
  pinMode(ledGreen, OUTPUT); // πράσινο led, ΕΞΟΔΟΣ
  pinMode(ledOrange, OUTPUT); // πορτοκαλί led, ΕΞΟΔΟΣ
}

void loop() {
  digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
  digitalWrite(ledOrange, LOW); // πορτοκαλί led σβηστό
  digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
  delay(6000); // διάρκεια 6 sec

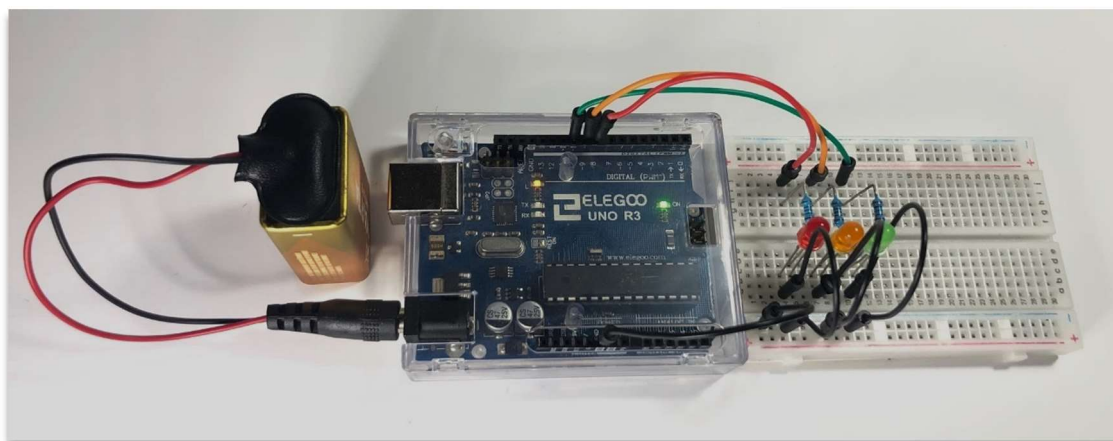
  digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
  digitalWrite(ledOrange, HIGH); // πορτοκαλί led αναμμένο
  digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
  delay(1000); //διάρκεια 1 sec

  digitalWrite(ledRed, HIGH); // κόκκινο led αναμμένο
  digitalWrite(ledOrange, LOW); // πορτοκαλί led σβηστό
  digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
  delay(3000); // διάρκεια 3 sec
}
```

Ζητάμε από τις ομάδες να ετοιμάσουν το breadboard με τα τρία led και με την παρακάτω σειρά: κόκκινο, πορτοκαλί και πράσινο (Σχ. 5.1). Έχουμε ήδη φορτωμένο τον κώδικα στις πλακέτες, οπότε παρατηρούμε το αποτέλεσμα. Αν αυτό ικανοποιεί τις προσδοκίες των παιδιών, προχωράμε στο επόμενο βήμα. Αν όχι, τότε κάνουμε στους χρόνους τις διορθώσεις που προτείνουν τα παιδιά, και φορτώνουμε τον διορθωμένο κώδικα.



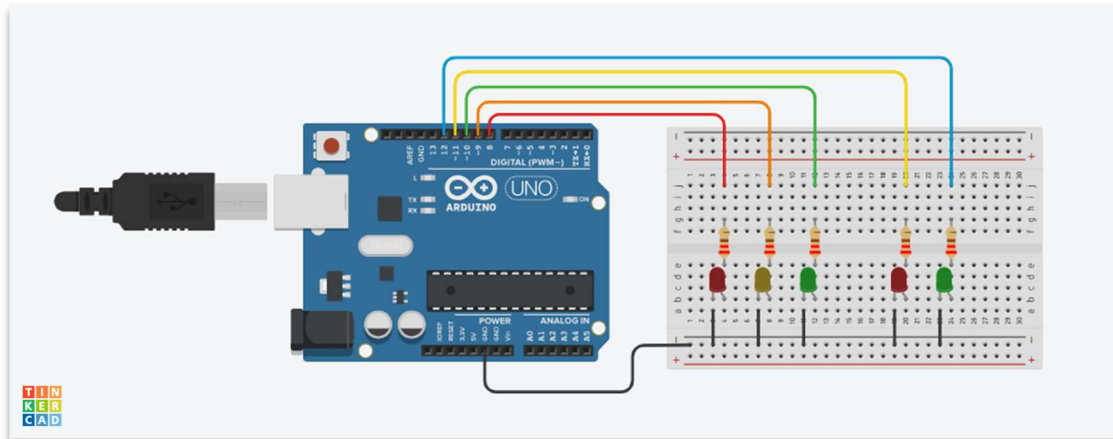
Σχ. 5.1



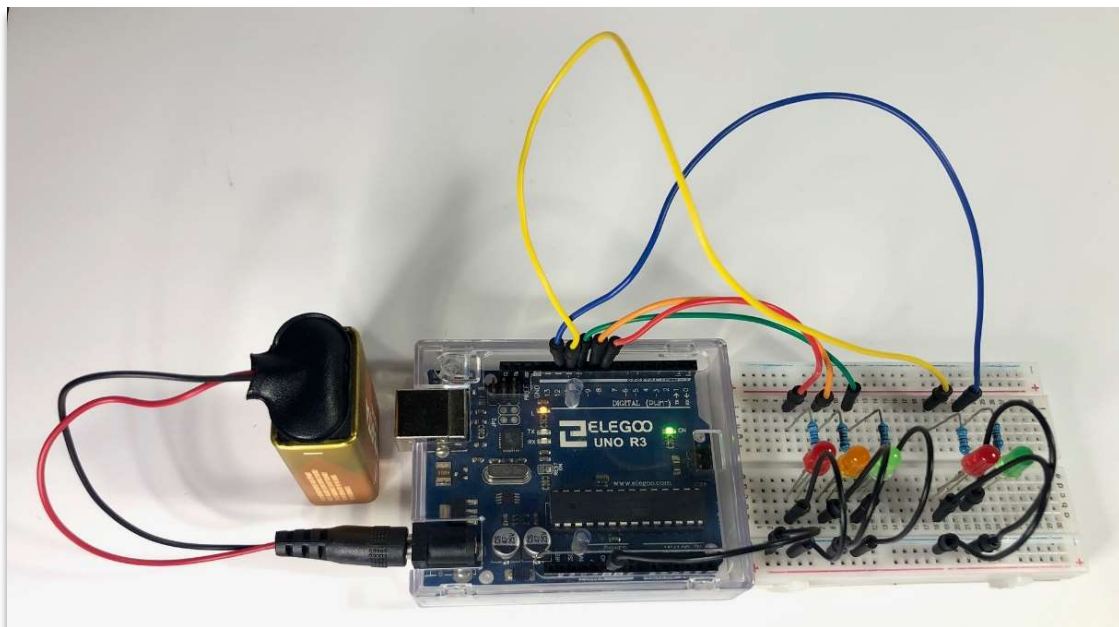
Εικ. 5.2 Υλοποίηση 1^{ου} σταδίου

Για να προχωρήσουμε στο επόμενο βήμα, δίνουμε στην κάθε ομάδα ένα κόκκινο και ένα πράσινο led, 2 αντιστάσεις των 220 Ω και τα απαραίτητα καλώδια, ώστε να προσομοιάσουν και τα φανάρια των πεζών (Σχ. 5.2). Τα προσθέτουν στο breadboard, και αφήνουμε στους μαθητές/τριες να προτείνουν τις απαραίτητες προσθήκες στον

κώδικα ώστε να έχουμε μια ολοκληρωμένη εικόνα των φαναριών κυκλοφορίας, με φανάρια για τα οχήματα αλλά και για τους πεζούς.



Σχ. 5.2



Εικ. 5.3 Υλοποίηση 2^{ου} σταδίου

Ο νέος κώδικας:

```
int ledRed= 8; // κόκκινο led στο pin 8
int ledOrange = 9; // πορτοκαλί led στο pin 9
int ledGreen = 10; // πράσινο led στο pin 10
int ledRed2= 11; // κόκκινο led στο pin 11
int ledGreen2 = 12; // πράσινο led στο pin 12

void setup() {
pinMode(ledRed, OUTPUT); // Κόκκινο led, ΕΞΟΔΟΣ
pinMode(ledOrange, OUTPUT); // Πορτοκαλί led, ΕΞΟΔΟΣ
pinMode(ledGreen, OUTPUT); // Πράσινο led, ΕΞΟΔΟΣ
pinMode(ledRed2, OUTPUT); // Κόκκινο led, ΕΞΟΔΟΣ
pinMode(ledGreen2, OUTPUT); // Πράσινο led, ΕΞΟΔΟΣ
}

void loop() {
digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
digitalWrite(ledOrange, LOW); // πορτοκαλί led σβηστό
digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
digitalWrite(ledRed2, HIGH); // κόκκινο led αναμμένο
digitalWrite(ledGreen2, LOW); // πράσινο led σβηστό
delay(6000); // διάρκεια 6 sec

digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
digitalWrite(ledOrange, HIGH); // πορτοκαλί led αναμμένο
digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
digitalWrite(ledRed2, HIGH); // κόκκινο led αναμμένο
digitalWrite(ledGreen2, LOW); // πράσινο led σβηστό
delay(1000); //διάρκεια 1 sec

digitalWrite(ledRed, HIGH); // κόκκινο led αναμμένο
digitalWrite(ledOrange, LOW); // πορτοκαλί led σβηστό
digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
digitalWrite(ledRed2, LOW); // κόκκινο led σβηστό
digitalWrite(ledGreen2, HIGH); // πράσινο led αναμμένο
delay(3000); // διάρκεια 3 sec
}
```

Αφού ολοκληρώσουν όλες οι ομάδες, συζητάμε όλοι μαζί για τις εντυπώσεις από το αποτέλεσμα, και κάνουμε μία σύγκριση με την εφαρμογή του τρίτου μαθήματος, όπου την ίδια εργασία έπρεπε να την κάνουμε χειροκίνητα με τη χρήση των κουμπιών (push button). Στον χρόνο που απομένει, οι ομάδες πειραματίζονται με περισσότερα λαμπάκια, προτείνουν αλλαγές στον κώδικα και φορτώνοντάς τον στον Arduino παρατηρούν το αποτέλεσμα. Τέλος, αποθηκεύονται τα υλικά.

6^ο Διδακτικό Σενάριο: Εισαγωγή στους αισθητήρες - Push Button

Στόχοι:

Οι μαθητές/τριες-

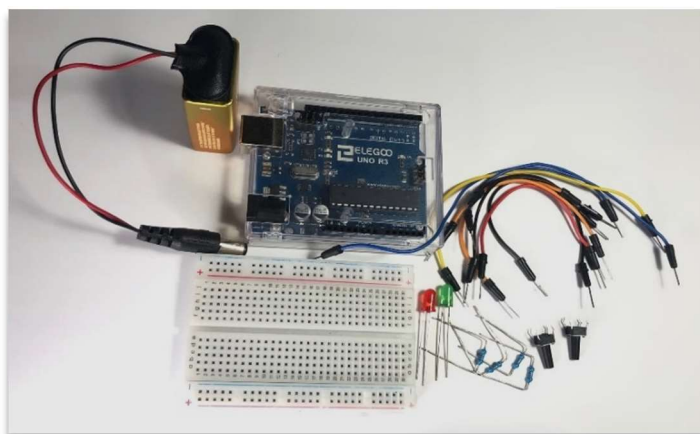
- Να κατανοήσουν οι μαθητές/τριες ότι ένα κουμπί (push button) είναι αισθητήρας για τον Arduino
- Να ξεχωρίσουν τη διαφορά ανάμεσα στην είσοδο – INPUT και στην έξοδο – OUTPUT
- Να γνωρίσουν τη χρήση της συνθήκης «if... else...»
- Να μπορούν να συμπληρώσουν η/και να τροποποιήσουν έναν κώδικα σύμφωνα με τις απαιτήσεις ενός project

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 κόκκινο led
- 1 πράσινο led
- 1 push button
- 8 καλώδια
- 2 αντιστάσεις 220 Ω
- 1 αντίσταση 10 KΩ

Υλικά ανά ομάδα – 2^ο στάδιο

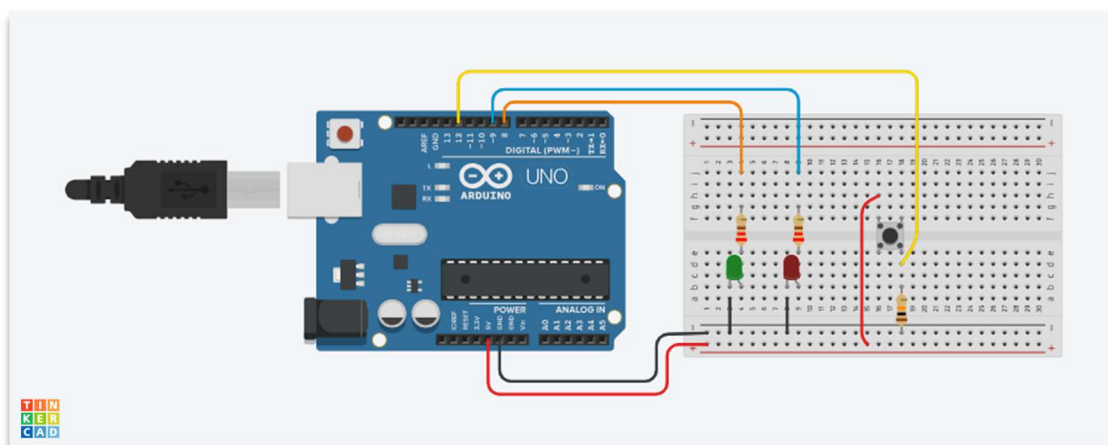
- 1 push button
- 2 καλώδια
- 1 αντίσταση 10 KΩ



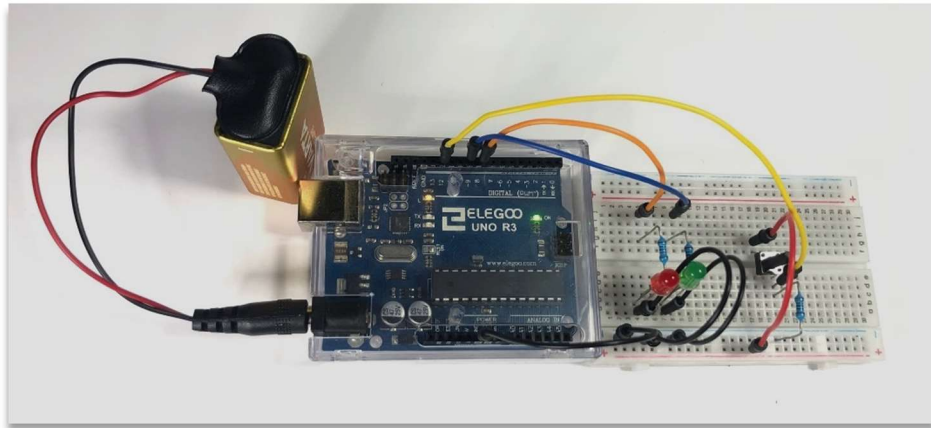
Εικ. 6.1 Υλικά

Μέχρι στιγμής είδαμε στα εισαγωγικά μαθήματα χωρίς τον Arduino, τη χρήση ενός κουμπιού, όπου λειτουργούσε σαν ένας απλός διακόπτης και είτε άνοιγε είτε έκλεινε ένα κύκλωμα. Ήρθε η ώρα να δούμε το κουμπί σε συνδυασμό με τον Arduino, ότι έχουμε τη δυνατότητα να προγραμματίσουμε ανάλογα με το αν είναι πατημένο ή όχι ποιες ενέργειες θα κάνουν οι ενεργοποιητές: να το πατάμε και να ανάβει ένα led, ή να το πατάμε και να σβήνει ένα αναμμένο led, ή ακόμη η ενέργεια αυτή να έχει διάρκεια, να πατάμε δηλαδή στιγμιαία το κουμπί και να έχουμε προγραμματίσει έτσι ώστε αυτό το άναμμα του led να διαρκεί πχ ένα λεπτό, όπως στο διάδρομο μιας πολυκατοικίας πατάμε ένα διακόπτη και η λάμπα ανάβει για κάποιο χρονικό διάστημα και μετά σβήνει μόνη της.

Μοιράζουμε τα υλικά στις ομάδες, και αυτές ετοιμάζουν το παρακάτω κύκλωμα, με την επισήμανση ότι η αντίσταση των 10 KΩ τοποθετείται στο push button. Για να μην χάνουμε χρόνο με το ποια αντίσταση είναι 220 Ω και ποια είναι 10 KΩ μετρώντας τες με το πολύμετρο, φροντίζουμε ώστε οι αντιστάσεις των 10 KΩ να είναι μεμβράνης άνθρακα, ώστε το χρώμα τους να είναι καφέ, σε αντίθεση με αυτές των 220 Ω, που είναι μεταλλικής μεμβράνης και χρώματος μπλε.



Σχ. 6.1



Εικ. 6.2 Υλοποίηση 1^{ου} σταδίου

Γράφουμε τον κώδικα στον πίνακα, και τον επεξηγούμε.

Κώδικας

```
int ledGreen = 8; // πράσινο led στο pin 8
int ledRed = 9; // κόκκινο led στο pin 9
int button = 12; // κουμπί στο pin 12

void setup() {
  pinMode(ledGreen, OUTPUT); // πράσινο led, ΕΞΟΔΟΣ
  pinMode(ledRed, OUTPUT); // κόκκινο led, ΕΞΟΔΟΣ
  pinMode(button, INPUT); // Κουμπί, ΕΙΣΟΔΟΣ
}

void loop() {
  if (digitalRead(button) == HIGH){ // Αν το κουμπί είναι πατημένο
    digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
    digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
  }

  else{
    digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
    digitalWrite(ledRed, HIGH); // κόκκινο led αναμμένο
  }
}
```

Επεξήγηση νέων εντολών

`int button = 12;` Όπως δίνουμε ένα όνομα στο led και το αντιστοιχίζουμε με κάποιο pin του Arduino, έτσι κάνουμε και με το push button, το ονομάζουμε «button» και δηλώνουμε ότι το συνδέουμε στο pin 12.

INPUT Το κουμπί είναι αισθητήρας, και ως αισθητήρας δίνει κάποια δεδομένα εισόδου στον Arduino.

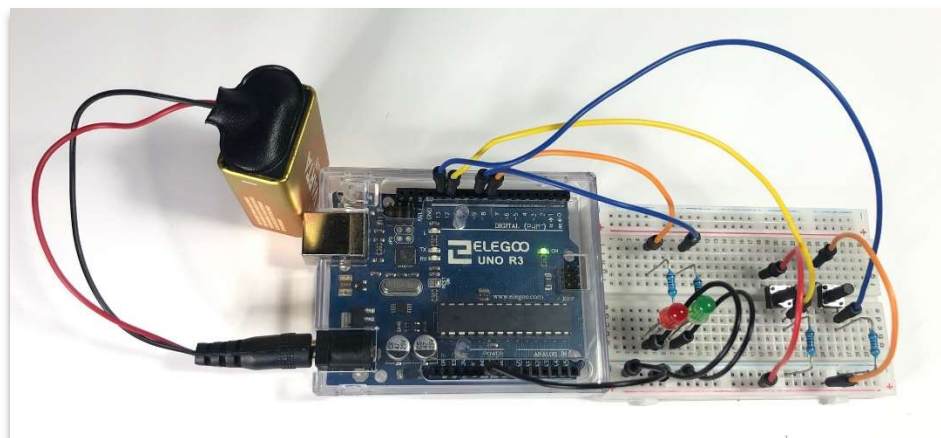
`digitalRead(button)` Όπως χρησιμοποιούμε την `digitalWrite` για μια έξοδο και την ενέργεια που θα κάνει ο Arduino σύμφωνα με τον κώδικά μας, αντίστοιχα χρησιμοποιούμε την `digitalRead`, ώστε να «διαβάσει» ο Arduino τι καταγράφει ο αισθητήρας μας, που σε αυτή την περίπτωση είναι το κουμπί.

`if (digitalRead(button) == HIGH){` «if» δηλαδή «εάν». Εάν ο Arduino διαβάσει την τιμή εισόδου στο pin 12 να είναι «HIGH», δηλαδή το κουμπί να είναι πατημένο, τότε θα κάνει την ενέργεια που ακολουθεί στην επόμενη γραμμή. Αυτή η ενέργεια είναι να ανάψει το πράσινο led και να σβήσει το κόκκινο, και συντακτικά, κλείνεται σε ένα ζευγάρι αγκίστρων `}`.

`else{` Είπαμε στον Arduino τι να κάνει όταν θα είναι πατημένο το κουμπί. Πρέπει όμως να του πούμε τι να κάνει και όταν δεν είναι πατημένο. Το μόνο που χρειάζεται η εντολή «else» είναι το τι θα γίνει όταν δεν ισχύει η συνθήκη της «if», άρα μέσα στα άγκιστρα της else γράφουμε τις ενέργειες που θα γίνουν όταν δεν είναι πατημένο το κουμπί.

Κάθε φορά μετά την επεξήγηση του κώδικα ζητάμε από τα παιδιά να εκφράσουν τυχόν απορίες, τις συζητάμε και τις λύνουμε όλοι μαζί.

Στη συνέχεια θα ανεβάσουμε λιγάκι το επίπεδο δυσκολίας, και θα δώσουμε ακόμη ένα κουμπί σε κάθε ομάδα, ζητώντας να το συνδέσουν δίπλα από το πρώτο. Θα συζητήσουμε και θα αποφασίσουμε όλοι μαζί σχετικά με το τί θα θέλαμε να κάνει ο Arduino τώρα που έχουμε δύο κουμπιά και δύο led.



Εικ. 6.3 Υλοποίηση 2^{ου} σταδίου

Αφού καταλήξουμε στον νέο κώδικα, τροποποιούμε αυτόν που έχουμε γραμμένο στον πίνακα. Μια τροποποίηση θα μπορούσε να είναι η εξής: όταν και για όσο πατάμε το πρώτο κουμπί, να ανάβει το πράσινο led, όταν και για όσο πατάμε το δεύτερο κουμπί να ανάβει το κόκκινο led, διαφορετικά (όταν κανένα κουμπί δεν είναι πατημένο), θα είναι και τα δύο led σβηστά.

Ο νέος κώδικας

```
int ledGreen = 8; // πράσινο led στο pin 8
int ledRed = 9; // κόκκινο led στο pin 8
int button1 = 12; // κουμπί 1 στο pin 12
int button2 = 13; // κουμπί 2 στο pin 13

void setup() {
  pinMode(ledGreen, OUTPUT); // πράσινο led, ΕΞΟΔΟΣ
  pinMode(ledRed, OUTPUT); // κόκκινο led, ΕΞΟΔΟΣ
  pinMode(button1, INPUT); // Κουμπί 1, ΕΙΣΟΔΟΣ
  pinMode(button2, INPUT); // Κουμπί 2, ΕΙΣΟΔΟΣ
}
void loop() {
  if (digitalRead(button1) == HIGH) { // Αν το κουμπί 1 είναι πατημένο
    digitalWrite(ledGreen, HIGH); // πράσινο led αναμμένο
    digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
  }
  if (digitalRead(button2) == HIGH) { // Αν το κουμπί 2 είναι πατημένο
    digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
    digitalWrite(ledRed, HIGH); // κόκκινο led αναμμένο
  }
  else{
    digitalWrite(ledGreen, LOW); // πράσινο led σβηστό
    digitalWrite(ledRed, LOW); // κόκκινο led σβηστό
  }
}
```

Φορτώνουμε τον κώδικα στις πλακέτες, και διαπιστώνουμε ότι δουλεύει. Σε περίπτωση που κάναμε κάποιο συντακτικό λάθος ούτως ή άλλως θα μας ενημερώσει το πρόγραμμα IDE ότι υπάρχει σφάλμα και δεν θα προχωρήσει στη μεταφόρτωση.

Συνεχίζουμε ζητώντας από τα παιδιά να σκεφτούν με ποιον τρόπο θα μπορούσαμε να έχουμε μια διάρκεια στο άναμμα των led, όταν δηλαδή πατήσουμε στιγμιαία ένα κουμπί, να ανάβει το αντίστοιχο led για 5 δευτερόλεπτα. Περιμένουμε πως θα δώσουν την σωστή απάντηση, ότι για να γίνει αυτό, απλά προσθέτουμε την εντολή «delay(5000);» πριν από το άγκιστρο κλεισίματος της κάθε «if».

Αφήνουμε τα παιδιά να παίζουν με τα κουμπιά, μπορούν, ανάλογα με τον χρόνο που απομένει, να προτείνουν και άλλες τροποποιήσεις στον κώδικα και να τις δοκιμάσουμε.

Στο τέλος αποθηκεύονται τα υλικά.

7^ο Διδακτικό Σενάριο: Φωτοαντίσταση

Στόχοι:

Οι μαθητές/τριες

- Να κατανοήσουν τη λειτουργία και τη χρήση της φωτοαντίστασης
- Να μάθουν σχετικά με τις αναλογικές εισόδους του Arduino
- Να κατανοήσουν πως λειτουργεί το φωτάκι νυκτός
- Να μάθουν πώς «φορτώνουμε» τον κώδικα στον Arduino

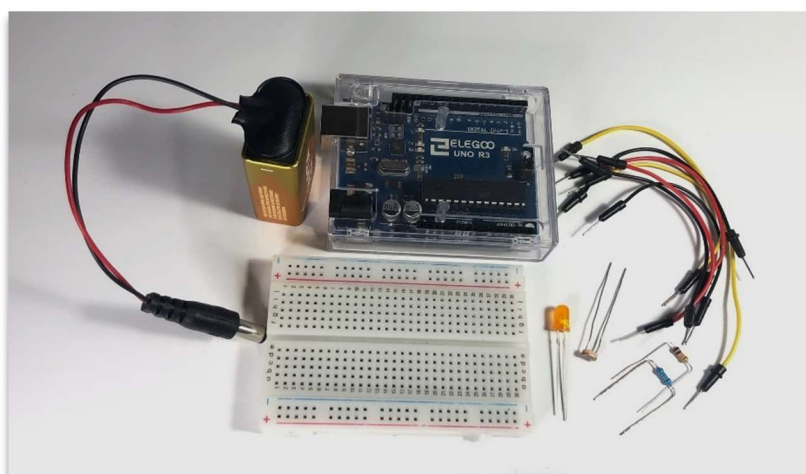
Σε αυτό το διδακτικό σενάριο οι μαθητές/τριες θα μάθουν πώς λειτουργεί το φωτάκι νυκτός, το οποίο ανάβει όταν νυχτώσει, και σβήνει όταν ξημερώσει.

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 φωτοαντίσταση
- 1 πορτοκαλί led
- 7 καλώδια
- 1 αντίσταση 220 Ω
- 1 αντίσταση 10 ΚΩ

Υλικά ανά ομάδα – 2^ο στάδιο

- 1 πράσινο led
- 2 καλώδια
- 1 αντίσταση 220 Ω

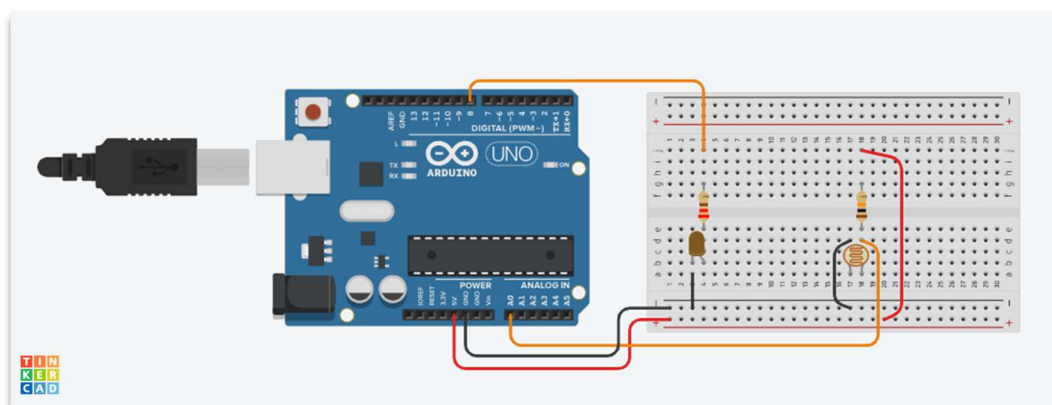


Εικ. 7.1 Υλικά

Δίνουμε στις ομάδες τα υλικά του 1ου σταδίου. Ρωτάμε τους/τις μαθητές/τριες αν αναγνωρίζουν ποιο είναι αυτό το καινούργιο εξάρτημα που έχουν μπροστά τους. Το πιθανότερο είναι να μην το έχουν ξαναδεί, οπότε αναφέρουμε το όνομά του, ότι πρόκειται δηλαδή για φωτοαντίσταση, και ρωτάμε αν γνωρίζουν ποια είναι η λειτουργία της. Αφού απαντήσουν τα παιδιά σχετικά με το τι υποθέτουν ότι κάνει, εξηγούμε ότι είναι ένας αισθητήρας, του οποίου η αντίσταση μεταβάλλεται ανάλογα με την ποσότητα του φωτός που πέφτει πάνω του. Όσο περισσότερο είναι το φως που πέφτει πάνω στην φωτοαντίσταση, τόσο μικρότερη είναι η αντίσταση, και, όσο μειώνεται το φως που πέφτει πάνω της, τόσο αυξάνεται η τιμή της.

Η φωτοαντίσταση συνδέεται σε μια από τις αναλογικές εισόδους του Arduino A0 έως A5, και οι τιμές που διαβάζει ο Arduino σε αυτές τις εισόδους κυμαίνεται μεταξύ 0 και 1023. Αν εμείς θέλουμε να ανάβει ένα led όταν αρχίσει να νυχτώνει, μπορούμε να χρησιμοποιήσουμε μια τιμή που είναι πιο κοντά στο 1023, άρα θα μπορούσαμε να θεωρήσουμε ότι η τιμή 700 προσομοιάζει στο σούρουπο.

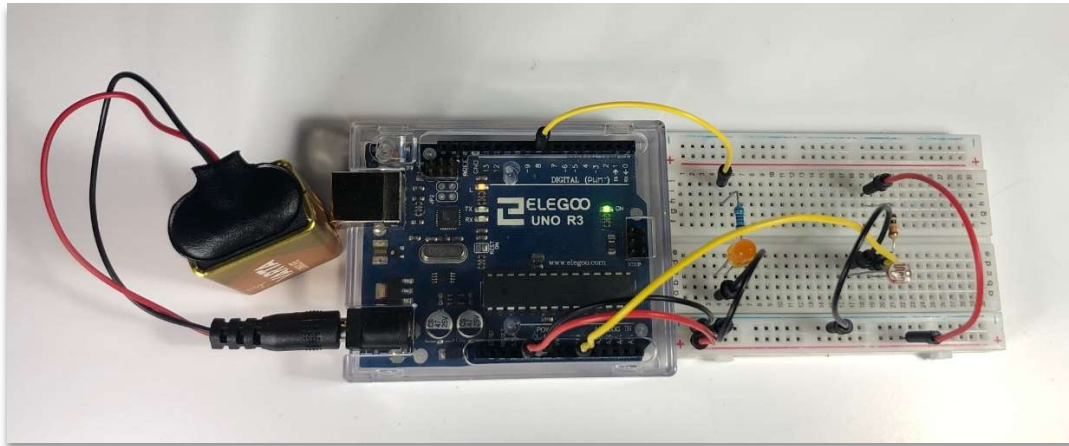
Μοιράζουμε σε κάθε ομάδα ένα σχέδιο του κυκλώματος με την φωτοαντίσταση, και ζητάμε να το κατασκευάσουν. Παράλληλα, γράφουμε στον πίνακα τον κώδικα που είναι φορτωμένος στον Arduino.



Σχ. 7.1

Αφήνουμε τα παιδιά να δουν το αποτέλεσμα, βάζοντας το χέρι τους κοντά στην φωτοαντίσταση. Το led ανάβει. Αυτό που συμβαίνει είναι το εξής: αφού μειώνεται η τιμή του φωτός που πέφτει πάνω στην φωτοαντίσταση, από μια μικρή, αρχικά, τιμή που ήταν η τιμή που διαβάζει ο Arduino, όσο πλησιάζουμε το χέρι μας, τόσο

μειώνεται το φως και άρα αυξάνεται σταδιακά η αντίσταση. Μόλις η τιμή αυτή που διαβάζει ο Arduino γίνει 700, τότε το led ανάβει.



Εικ. 7.2 Υλοποίηση 1^{ου} σταδίου

Αυτό άλλωστε έχουμε γράψει και στον κώδικα, τον οποίο αναλύουμε.

Κώδικας

```
int ledOrange=8;
int a=A0;

void setup() {
  pinMode(ledOrange, OUTPUT);
}

void loop() {
  a=analogRead(A0);
  if(a>700){
    digitalWrite(ledOrange, HIGH);
  }

  else{
    digitalWrite(ledOrange, LOW);
  }
}
```

Επεξήγηση νέων εντολών:

`int a=A0;` Όπως δίνουμε ένα όνομα στο led και το αντιστοιχίζουμε με κάποιο pin του Arduino, έτσι κάνουμε και με την φωτοαντίσταση, την ονομάζουμε «a» και δηλώνουμε ότι την συνδέουμε στο αναλογικό pin A0.

`a=analogRead(A0);` Ορίζουμε ότι, εφόσον το «a» είναι είσοδος, ο Arduino θα «διαβάσει» την τιμή στο αναλογικό pin A0.

`if(a>700){` Ορίσαμε αυθαίρετα την τιμή 700, ως την τιμή που θα διαβάσει ο Arduino στο A0 όταν αρχίσει να νυχτώνει. Αυτό που ζητάμε να γίνει, είναι: όταν το φως λιγοστεύει, άρα η τιμή στο A0 γίνει μεγαλύτερη από 700, να γίνει κάποια ενέργεια. Αυτή η ενέργεια είναι να ανάψει το led, και συντακτικά, κλείνεται σε ένα ζευγάρι αγκίστρων {}.

`else{` Είπαμε στον Arduino τι να κάνει όταν θα είναι νύχτα, όταν δηλαδή είναι $a > 700$. Πρέπει όμως να του πούμε τι να κάνει σε διαφορετική περίπτωση, όταν δηλαδή δεν είναι νύχτα, όταν το a πάρει τιμές μικρότερες από 700. Συντακτικά, και αυτή η ενέργεια κλείνεται σε ένα ζευγάρι αγκίστρων {}.

Ζητάμε από τους/τις μαθητές/τριες να εκφράσουν τυχόν απορίες, και αφού τις λύσουμε, περνάμε στο επόμενο στάδιο. Μοιράζουμε τα επιπλέον υλικά, και αυτό που θέλουμε είναι το εξής: να προσθέσουν ένα δεύτερο led στο υπάρχον κύκλωμα, και να τροποποιήσουν τον κώδικα έτσι, ώστε όταν είναι μέρα να είναι αναμμένο το πορτοκαλί led, ενώ όταν είναι νύχτα να ανάβει το πράσινο. Δεν είναι δύσκολο αυτό που ζητάμε, είναι όμως μια καλή ευκαιρία - επειδή έχουμε αρκετό χρόνο στη διάθεσή μας - να καθίσουν τα παιδιά στον υπολογιστή, να τροποποιήσουν τον κώδικα, και να τους δείξουμε πώς γίνεται να «περάσουμε» τον κώδικα, από τον υπολογιστή στον Arduino. Και αν κάτι δεν πάει όπως το περιμέναμε, να βρούμε το λάθος και να το διορθώσουμε. Φροντίζουμε να κάνουν όλες οι ομάδες την παραπάνω ενέργεια.

Τέλος, τα παιδιά αποσυναρμολογούν τα κυκλώματά τους και αποθηκεύουν τα υλικά.

8^ο Διδακτικό Σενάριο: Ποτενσιόμετρο

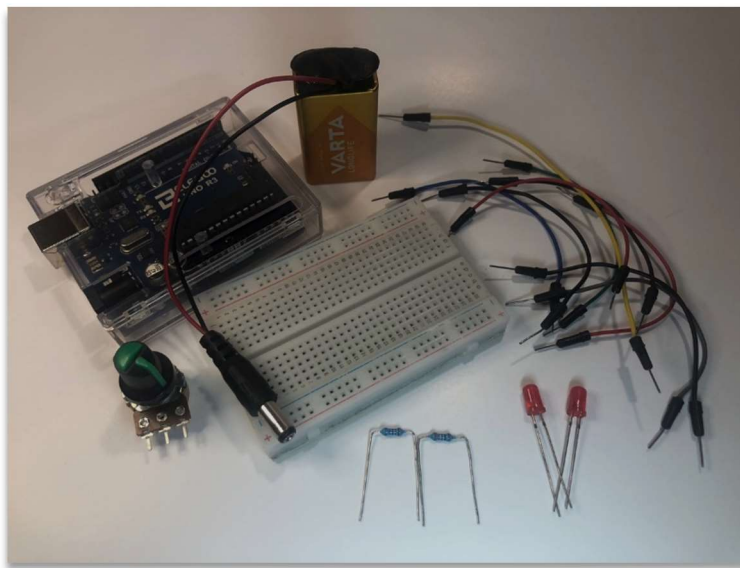
Στόχοι:

Οι μαθητές/τριες

- Να κατανοήσουν τη λειτουργία και τη χρήση του ποτενσιόμετρου
- Να κατανοήσουν την εντολή map
- Να κατανοήσουν πως λειτουργεί το dimmer στον φωτισμό ενός χώρου

Υλικά ανά ομάδα

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 ποτενσιόμετρο
- 2 κόκκινα led
- 9 καλώδια
- 2 αντιστάσεις 220 Ω



Εικ. 8.1 Υλικά

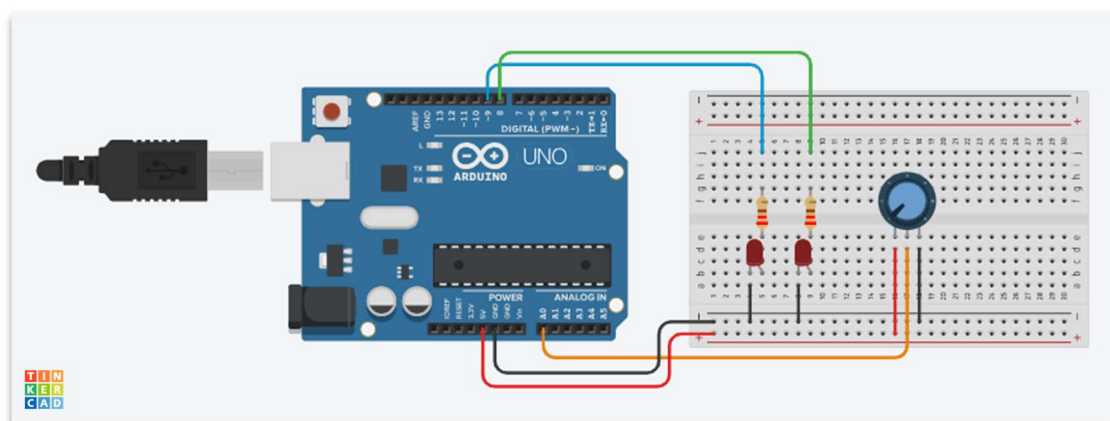
Συζητάμε με τα παιδιά σχετικά με τις εφαρμογές που είδαμε στα προηγούμενα μαθήματα, και οι οποίες έχουν σχετίζονται με τον φωτισμό ενός χώρου. Μέχρι στιγμής είδαμε πώς μπορούμε να έχουμε αναμμένο ένα led με τη χρήση του push button, ή να προγραμματίσουμε ώστε να μείνει αναμμένο για κάποιο χρονικό διάστημα, και μετά να σβήνει μόνο του. Είδαμε επίσης πώς μπορούμε να προγραμματίσουμε τον Arduino ώστε, με τη χρήση μιας φωτοαντίστασης, να ανάβει ένα led όταν νυχτώσει, και να σβήνει όταν ξημερώσει. Τώρα θα δούμε με ποιον

τρόπο μπορούμε να αυξομειώσουμε την ένταση ενός led, όπως γίνεται και στον φωτισμό ενός χώρου με τη χρήση ενός dimmer.

Ρωτάμε τα παιδιά αν έχουν βρεθεί σε κάποιον χώρο, στον οποίο να έχουν παρατηρήσει ότι υπάρχει η δυνατότητα να ρυθμίζεται η ένταση του φωτισμού με κάποιον περιστρεφόμενο διακόπτη. Ίσως κάποια από αυτά να έχουν σε κάποιον χώρο του σπιτιού τους αυτή τη δυνατότητα, συχνότερα όμως το συναντούμε σε χώρους διασκέδασης, καφετέριες κλπ. Η συνηθέστερη ονομασία αυτού του διακόπτη είναι «dimmer», και δεν είναι παρά ένα ποτενσιόμετρο.

Τι είναι όμως το ποτενσιόμετρο; Με απλά λόγια, είναι μία μεταβλητή αντίσταση. Ανάλογα με τη θέση του περιστρεφόμενου διακόπτη, η αντίσταση αυξάνεται ή μειώνεται. Χρησιμοποιείται σε διάφορες εφαρμογές, η πιο γνωστή στα παιδιά είναι ο διακόπτης που χρησιμοποιείται στα εξωτερικά ηχεία του υπολογιστή, όπου περιστρέφοντάς τον αυξομειώνουν την ένταση του ήχου.

Αντίστοιχα, με τον ίδιο τρόπο, μπορούμε στις εφαρμογές με τον Arduino να αυξομειώσουμε την φωτεινότητα ενός led. Δίνουμε στους μαθητές και στις μαθήτριες το σχέδιο της συνδεσμολογίας, και το υλοποιούν, ενώ εμείς γράφουμε στον πίνακα τον κώδικα.



Σχ. 8.1

Κώδικας

```
int ledRed1 = 8;
int ledRed2 = 9;
int potPin = A0;
int potValue;

void setup() {
  pinMode(ledRed1, OUTPUT);
  pinMode(ledRed2, OUTPUT);
}

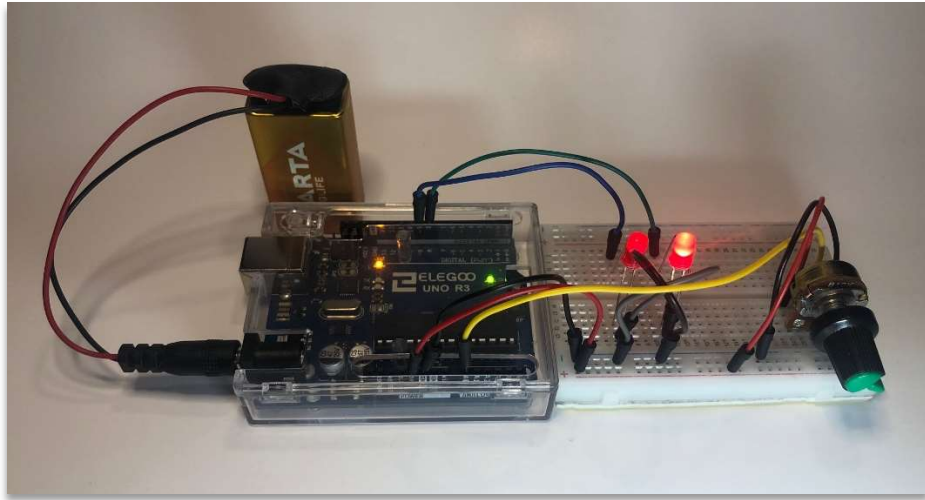
void loop() {
  potPin=analogRead(A0);
  potValue=map(potPin, 0, 1023, 0, 255);
  digitalWrite(ledRed1, HIGH);
  analogWrite(ledRed2, potValue);
}
```

Επεξήγηση νέων εντολών:

`potValue=map(potPin, 0, 1023, 0, 255);` Αντιστοιχίζουμε το εύρος τιμών (0 έως 1023) της αναλογικής εισόδου του ποτενσιόμετρου, στο εύρος (0 έως 255) της ψευδοαναλογικής εξόδου του LED.

`analogWrite(ledRed, potValue);` Το κόκκινο LED στο pin 9, που είναι η ψευδοαναλογική έξοδος του LED, η ένταση με την οποία φωτοβολεί το LED θα είναι ανάλογη με την τιμή του ποτενσιόμετρου.

Έχουμε επίτηδες τοποθετήσει 2 led στο κύκλωμά μας. Το 1ο είναι πάντα αναμμένο στην μέγιστη ένταση, αφού είναι σε ψηφιακή έξοδο και το σήμα του θα μπορούσε να είναι μόνο LOW ή HIGH (εμείς το έχουμε HIGH), ενώ το 2ο ανάβει με ένταση αντιστρόφως ανάλογη της αντίστασης του ποτενσιόμετρου, έτσι ώστε να υπάρχει σύγκριση με τη μέγιστη φωτεινότητα.



Εικ. 8.2 Υλοποίηση Σεναρίου

Αφού οι μαθητές και οι μαθήτριες κατασκευάσουν το κύκλωμα, τροφοδοτούν με τάση και «παίζουν» με το ποτενσιόμετρο περιστρέφοντάς το. Συζητάμε όλοι μαζί τις παρατηρήσεις τους, και, αν κάποια ομάδα θέλει, μπορεί να πάρει επιπλέον υλικά και να εξαντλήσει τον υπόλοιπο χρόνο για να τροποποιήσει τον κώδικα και το κύκλωμά της.

Τέλος, οι ομάδες αποθηκεύουν τα υλικά τους.

9^ο Διδακτικό Σενάριο: Αισθητήρας Υπερήχων με LED και Buzzer

Στόχοι:

Οι μαθητές/τριες

- Να κατανοήσουν τη λειτουργία και τη χρήση του αισθητήρα υπερήχων
- Να αντιληφθούν την αναγκαιότητα της χρήσης βιβλιοθηκών
- Να καταλάβουν πως λειτουργεί ένας συναγερμός προσέγγισης
- Να μάθουν πώς να ελέγξουν και να μεταφορτώσουν έναν κώδικα στον Arduino

Υλικά ανά ομάδα – 1^ο στάδιο

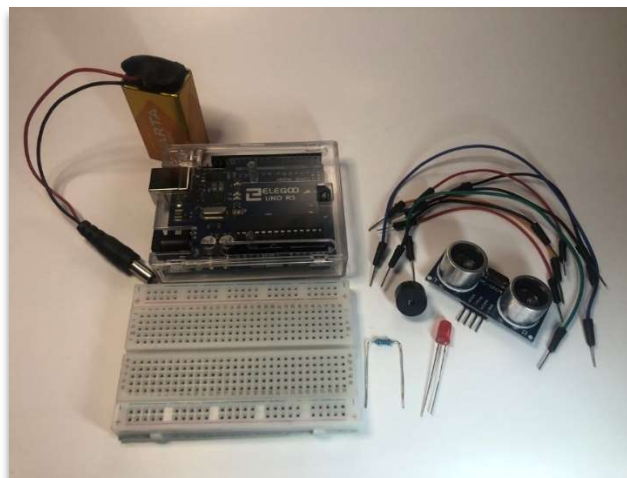
- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 αισθητήρας υπερήχων
- 1 κόκκινο LED
- 8 καλώδια
- 1 αντίσταση 220 Ω

Υλικά ανά ομάδα – 2^ο στάδιο

- 1 buzzer
- 1 αντίσταση 100 Ω

Υλικά ανά ομάδα – 3^ο στάδιο

- led
- αντιστάσεις 220 Ω
- καλώδια



Εικ. 9.1 Υλικά

Αναρωτιόμαστε αν γνωρίζει κάποιος/α τον τρόπο με τον οποίο μια νυχτερίδα εντοπίζει την τροφή της, ή αποφεύγει εμπόδια, από την στιγμή που δεν βλέπει. Είναι κάτι που οι περισσότεροι/ες έχουν ακούσει σε κάποια μικρότερη τάξη. Ακούγονται όλες οι απόψεις, και καταλήγουμε στην σωστή, ότι στέλνει, δηλαδή, η νυχτερίδα κάποια σήματα, και όταν αυτά συναντήσουν κάποιο εμπόδιο, τότε τα σήματα χτυπάνε πάνω σε αυτό και επιστρέφουν πίσω στη νυχτερίδα. Το κάνουμε και λίγο πιο συγκεκριμένο: τα σήματα αυτά ονομάζονται υπέρηχοι, και υπάρχουν αισθητήρες που δουλεύουν με τον ίδιο τρόπο.

Ο αισθητήρας υπερήχων (Ultrasonic Sensor) έχει δύο «ματάκια»: τον Πομπό (Transmitter) και τον Δέκτη (Receiver) υπερήχων. Ο πομπός εκπέμπει υπερήχους, οι οποίοι όταν συναντήσουν κάποιο εμπόδιο, επιστρέφουν στον δέκτη. Γνωρίζοντας την ταχύτητα του ήχου και τον χρόνο που κάνει ο υπέρηχος να διανύσει την απόσταση μέχρι το εμπόδιο και να επιστρέψει στον δέκτη, μπορούμε να υπολογίσουμε την απόσταση του εμποδίου από τον αισθητήρα. Δεν χρειάζεται να ασχοληθούμε περαιτέρω με τον υπολογισμό της απόστασης, αυτό το κάνει για εμάς ο Arduino, με τις κατάλληλες εντολές.

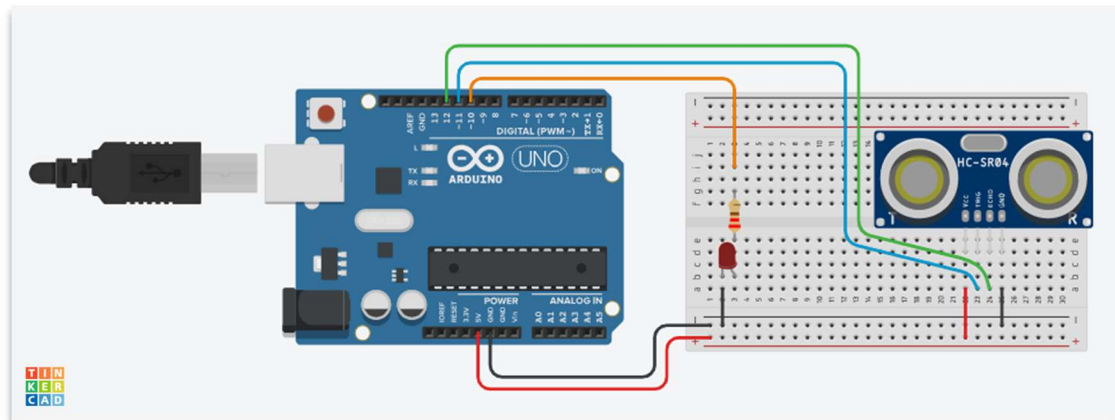
Υπάρχουν κάποιοι περιορισμοί κατά τη χρήση του αισθητήρα υπερήχων, που καλό είναι να τους γνωρίζουμε. Αν το εμπόδιο είναι πολύ μικρό ή έχει χνουδωτή υφή, ο υπέρηχος δεν μπορεί να επιστρέψει, και έτσι ο δέκτης δεν θα τον λάβει (θα θεωρήσει ότι ο υπέρηχος δεν συνάντησε εμπόδιο). Επίσης, το εμπόδιό μας θα πρέπει να βρίσκεται σε απόσταση από 2cm έως 4m, αυτή είναι η εμβέλεια του αισθητήρα υπερήχων που έχουμε στη διάθεσή μας.

Μοιράζουμε στις ομάδες τα υλικά του 1^{ου} σταδίου και το σχέδιο του κυκλώματος (Σχ. 8.1), και καθώς υλοποιούν το κύκλωμα εμείς γράφουμε τον κώδικα στον πίνακα. Ζητάμε από τα παιδιά να τοποθετήσουν τον αισθητήρα σε τέτοια θέση στο breadboard, πχ στις θέσεις a22 έως a25, ώστε τα καλώδια να τοποθετηθούν πίσω από αυτόν, και όχι όπως ακριβώς φαίνεται στο σχέδιο του κυκλώματος (απλά δεν γίνεται, αν θέλουμε τον αισθητήρα «καρφωμένο» στο breadboard, να τον παρουσιάσουμε διαφορετικά).

Ο αισθητήρας έχει 4 ακίδες. Η ακίδα VCC συνδέεται στα 5V του breadboard, οι ακίδες TRIG και ECHO συνδέονται στην ψηφιακή έξοδο και στην ψηφιακή είσοδο

αντίστοιχα του Arduino που ορίζουμε στον κώδικα, ενώ η ακίδα GND συνδέεται στη γείωση του breadboard.

Αυτόν τον κώδικα τον έχουμε ήδη φορτωμένο στις πλακέτες.



Σχ. 9.1

Τα παιδιά παρατηρούν το αποτέλεσμα, πλησιάζουν το χέρι τους στον αισθητήρα υπερήχων και όταν αυτό φτάσει σε απόσταση 10cm, τότε ανάβει το κόκκινο LED, και παραμένει αναμμένο όσο αυτή η απόσταση είναι μικρότερη από 10cm. Όταν το χέρι απομακρυνθεί από τον αισθητήρα και η απόσταση αυτή γίνει μεγαλύτερη των 10cm, το LED σβήνει. Κατόπιν, επεξηγούμε τις εντολές του κώδικα, ώστε να δούμε ότι αυτό είναι που ορίσαμε να γίνει.

Κώδικας

```
#include <SR04.h>
int ledRed=10;
int trigPin=11;
int echoPin=12;
long a;
SR04 sr04 = SR04(echoPin,trigPin);

void setup() {
```

```

pinMode(ledRed,OUTPUT);
pinMode(trigPin,OUTPUT);
pinMode(echoPin,INPUT);
}

void loop() {
a=sr04.Distance();

if(a<10){
digitalWrite(ledRed,HIGH);
}
else{
digitalWrite(ledRed,LOW);
}
}

```

Επεξήγηση νέων εντολών

`#include <SR04.h>` Με αυτή την εντολή συμπεριλαμβάνουμε την βιβλιοθήκη του αισθητήρα υπερήχων, απλουστεύοντας τον κώδικά μας. Η βιβλιοθήκη που θέλουμε να συμπεριλάβουμε κλείνεται μέσα σε <> ή “ ”.

`int echoPin=11;` Συνδέουμε την ακίδα «echo» στο pin 11

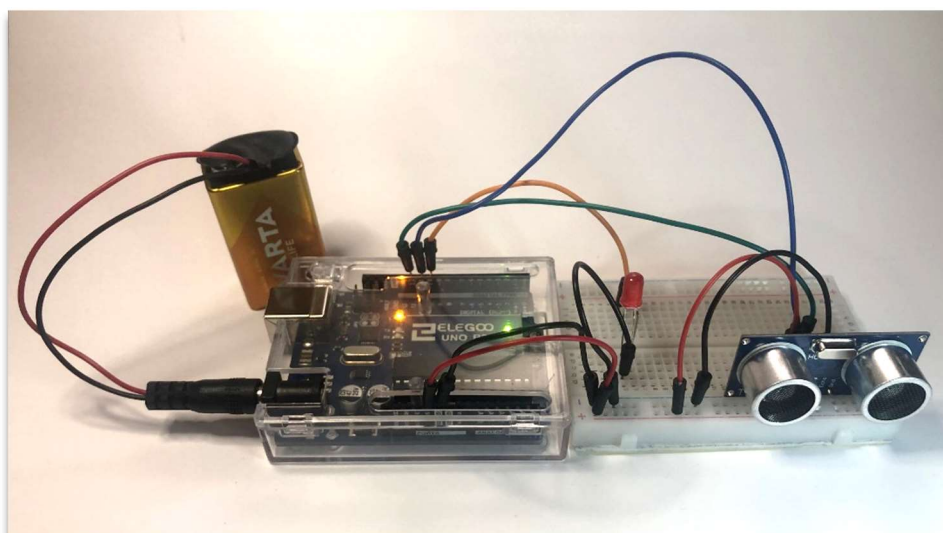
`int trigPin=12;` Συνδέουμε την ακίδα «trig» στο pin 12

`long a;` Είναι η αντίστοιχη εντολή της int στο LED, τη χρησιμοποιούμε επειδή αφορά σε διάρκεια (χρόνο)

`SR04 sr04 = SR04(echoPin,trigPin);` Με αυτή την εντολή ορίζουμε την χρήση δεδομένων από τον αισθητήρα υπερήχων

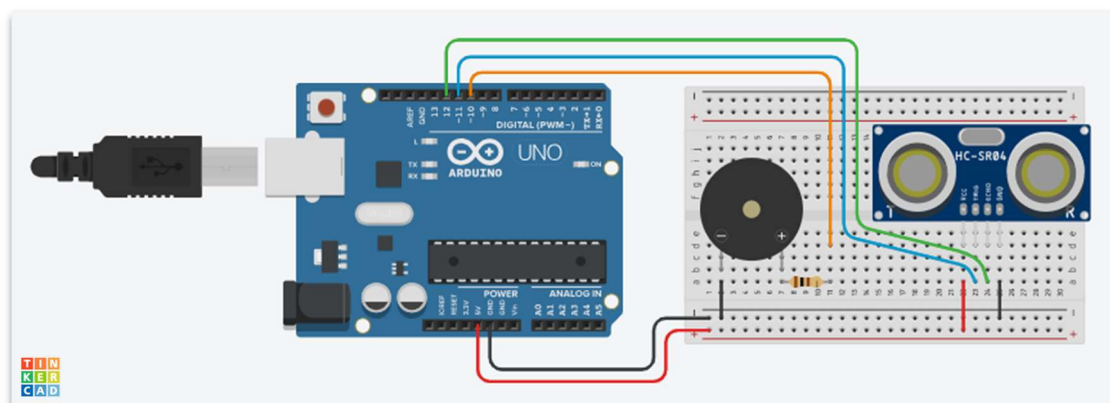
`a=sr04.Distance();` Μετατρέπει τη διάρκεια εκπομπής και λήψης υπερήχου σε cm

`if(a<10){` Συνθήκη if: αν ο αισθητήρας ανιχνεύσει εμπόδιο σε απόσταση μικρότερη των 10 cm, τότε....



Εικ. 9.2 Υλοποίηση Σεναρίου

Αφού εξηγήσουμε τον κώδικα και λύσουμε απορίες των παιδιών, προχωράμε στο 2ο στάδιο. Μοιράζουμε τα επιπλέον υλικά και το κύκλωμα (Σχ. 9.2) και περιμένουμε να ακούσουμε το αποτέλεσμα.





Σχ. 9.2

Σε αυτή την υλοποίηση, όταν ένα αντικείμενο πλησιάζει τον αισθητήρα μας σε απόσταση μικρότερη των 10 cm, το αποτέλεσμα είναι να ακούγεται από το buzzer ένας συνεχής ήχος, όπως και στην υλοποίηση του 1^{ου} σταδίου ανάβε το κόκκινο LED.

Ο κώδικας που εκτελεί ο Arduino είναι ο ίδιος, δεν αλλάξαμε κάτι, σε σχέση με το προηγούμενο στάδιο. Δεν είναι πρόβλημα το ότι έχουμε buzzer αλλά στον κώδικα αναφερόμαστε σε ledRed. Εμείς γράφουμε τον κώδικα, και δίνουμε όποιο όνομα θέλουμε. Το σημαντικό είναι να χρησιμοποιήσουμε το ίδιο pin.

Ας επιστρέψουμε στον συνεχή (και ενοχλητικό) ήχο του buzzer. Είναι η ιδανική αφορμή να θέσουμε κάποιες προκλήσεις στις ομάδες μας, και να περάσουμε στο 3^ο στάδιο. Αναθέτουμε να προτείνει η κάθε ομάδα όποιες αλλαγές θέλει στον κώδικα, ώστε να έχουμε κάποιο αποτέλεσμα που να προσομοιάζει σε συναγερμό, είτε με κόκκινο led που αναβοσβήνει, είτε με buzzer που ηχεί διακοπτόμενα, ή και τα δύο. Ή, ακόμη, μπορεί κάποια ομάδα να τροποποιήσει τον κώδικα για να ανάβει κόκκινο LED όταν έχει πλησιάσει κάποιο αντικείμενο, και πράσινο όταν δεν υπάρχει αντικείμενο κοντά, ή το αντίστροφο. Μπορούν επίσης να αλλάξουν την απόσταση στην οποία ενεργοποιείται ο «συναγερμός».

Μας δίνεται επίσης η ευκαιρία να εξηγήσουμε τη διαδικασία με την οποία φορτώνουμε έναν κώδικα στον Arduino. Αφού γράψουμε ή τροποποιήσουμε έναν κώδικα, στο πάνω αριστερό μέρος του περιβάλλοντος του Arduino IDE υπάρχει το εικονίδιο «verify ». Είτε έχουμε συνδέσει τον Arduino με το καλώδιο, είτε όχι, πατώντας στο εικονίδιο αυτό, γίνεται έλεγχος σχετικά με το αν η σύνταξη του κώδικα είναι σωστή. Αν λείπει κάποιο «;», κάποιο «{ }», αν έχουμε κάνει κάποιο τυπογραφικό λάθος (γράφουμε ledRed σε μια γραμμή, ενώ σε άλλη leddRed), τότε το ίδιο το πρόγραμμα μας ενημερώνει για το λάθος, και φυσικά πρέπει να το διορθώσουμε.

Αν δεν έχουμε κάποιο λάθος, μπορούμε να περάσουμε τον κώδικα στον Arduino, αφού πρώτα τον συνδέσουμε στον υπολογιστή με το καλώδιό του. Επιλέγουμε από την διαδρομή «tools», «port» την θύρα στην οποία είναι συνδεδεμένη η πλακέτα. Πατάμε το εικονίδιο «upload » και ο κώδικας μεταφορτώνεται στον Arduino.

Αφήνουμε να πάρουν όσα LED, καλώδια και αντιστάσεις χρειάζονται, και επιβλέπουμε, ώστε να τροποποιήσουν τον κώδικα και να περάσουν τον τροποποιημένο κώδικα στις πλακέτες.

Αφού ολοκληρώσουν οι ομάδες τους πειραματισμούς τους και ευχαριστηθούν το αποτέλεσμα, αποθηκεύουν τα υλικά τους.

10^ο Διδακτικό Σενάριο: Αισθητήρας Υπερήχων και Servo - Theremin

Στόχοι:

Οι μαθητές/τριες

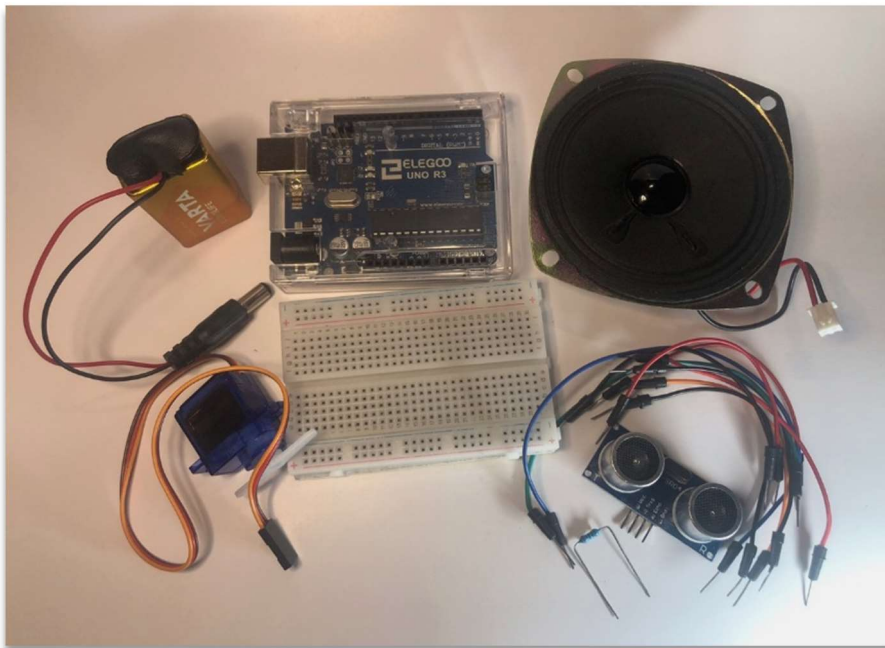
- Να κατανοήσουν τη λειτουργία και τη χρήση του Servo
- Να εμποδίσουν τη χρήση βιβλιοθηκών
- Να είναι σε θέση να χρησιμοποιήσουν έτοιμους κώδικες

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 αισθητήρας υπερήχων
- 1 σερβοκινητήρας
- 9 καλώδια

Υλικά ανά ομάδα – 2^ο στάδιο

- 1 buzzer
- 1 αντίσταση 100 Ω



Εικ. 10.1 Υλικά

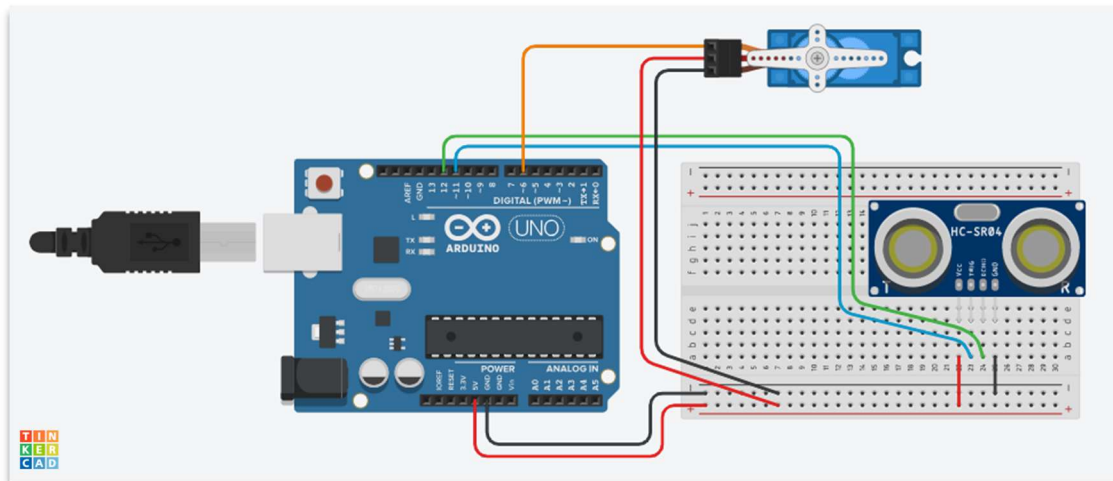
Μοιράζουμε τα υλικά στα παιδιά και συστήνουμε το νέο εξάρτημα, τον σερβοκινητήρα, ή αλλιώς, Servo. Εξηγούμε ποια είναι η λειτουργία και ποια η χρήση του.

Διευκρινίζουμε ότι δεν είναι ένας απλός κινητήρας, του οποίου ο άξονας περιστρέφεται για όσο τροφοδοτείται με τάση. Είναι ένας κινητήρας, με ένα σύστημα γραναζιών που μειώνει τις στροφές του άξονά του, και που η δυνατότητα που έχουμε προγραμματίζοντάς τον είναι να περιστρέφεται για την γωνία που ορίζουμε στον κώδικα, μετρούμενη σε μοίρες. Έχουμε έναν περιορισμό όσον αφορά στη γωνία περιστροφής, αυτή συνήθως δεν μπορεί να είναι μεγαλύτερη από 180° , αλλά υπάρχουν και σερβοκινητήρες που μπορούν να περιστρέφουν τον άξονά τους μέχρι 360° .

Ένα χαρακτηριστικό παράδειγμα της χρήσης του είναι η μπάρα σε ένα πάρκινγκ αυτοκινήτων ή στα διόδια. Κάποιος αισθητήρας αντιλαμβάνεται ότι ισχύει κάποια συνθήκη, και ο σερβοκινητήρας περιστρέφεται για κάποιες μοίρες. Ο κώδικας μπορεί να ορίζει ότι όταν παύει να ισχύει η προαναφερόμενη συνθήκη, ο άξονας του σερβοκινητήρα να επιστρέφει στην αρχική του θέση.

Αυτό θα κάνουμε και εμείς με τον σερβοκινητήρα και τον αισθητήρα υπερήχων. Όταν πλησιάσει ένα αντικείμενο στον αισθητήρα, η μπάρα θα περιστραφεί από τη θέση 0 στις 90° . Όταν το αντικείμενο απομακρυνθεί, η μπάρα θα επιστρέψει στη θέση 0. Για να μπορέσουμε να τα κάνουμε όμως όλα αυτά, θα πρέπει να συνδέσουμε τον σερβοκινητήρα σε μια ψευδοαναλογική έξοδο (PWM).

Τα παιδιά ετοιμάζουν το κύκλωμα (Σχ. 9.1), κι εμείς γράφουμε τον κώδικα στον πίνακα και τον εξηγούμε.



Σχ. 10.1

Η υλοποίηση του κυκλώματος έχει το εξής αποτέλεσμα: όταν πλησιάζει στον αισθητήρα ένα αντικείμενο, η «εικονική» μπάρα ανεβαίνει. Βάλαμε και μια καθυστέρηση 3 δευτερολέπτων, για να προλάβει να περάσει από την σηκωμένη μπάρα το όχημα, και μετά η μπάρα επιστρέφει στην οριζόντια θέση. Στην πραγματικότητα αυτά τα 3 δευτερόλεπτα καθυστέρησης τα βάλαμε στον κώδικα για να βοηθήσουμε τον σερβοκινητήρα να επιβιώσει από την παιχνιδιάρικη διάθεση των παιδιών, που θα έκαναν δοκιμές με το χέρι τους ως αντικείμενο σε απόσταση 9 έως 11 cm από τον αισθητήρα.

Κώδικας

```
#include <SR04.h>
#include <Servo.h>
int trigPin=11;
int echoPin=12;
SR04 sr04 = SR04(echoPin, trigPin);
long a;
Servo myServo;
int i;

void setup() {
  myServo.attach(6);
}

void loop() {
  a=sr04.Distance();
```

```

if(a<10){
myServo.write(90);
delay(3000);
}
else{
myServo.write(0);
}
}

```

Επεξήγηση κώδικα

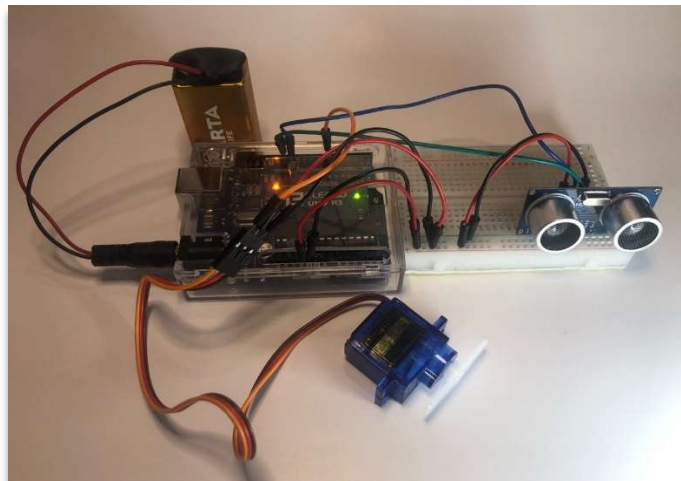
`Servo myServo;` Δημιουργούμε ένα αντικείμενο τύπου Servo.

`int i;` Δημιουργούμε την μεταβλητή *i* για την γωνία περιστροφής του άξονα.

`myServo.attach(6);` Προσαρτούμε το αντικείμενο myServo στο pin 6 (PWM).

`myServo.write(90);` Περιστρέφεται ο άξονας του servo στην τιμή $i=90^\circ$.

`myServo.write(0);` Περιστρέφεται ο άξονας του servo στην τιμή $i=0^\circ$.



Εικ. 10.2 Υλοποίηση Σεναρίου

Αυτή η δραστηριότητα διαρκεί μία διδακτική ώρα. Για την επόμενη ώρα αφαιρούμε τον σερβοκινητήρα και επιστρέφουμε στο προηγούμενο διδακτικό σενάριο. Δίνουμε στα παιδιά από μια αντίσταση $100\ \Omega$ και ένα buzzer. Αν έχουμε στη διάθεσή μας ηχεία (το εσωτερικό των ηχείων υπολογιστή για την ακρίβεια, και κάποιον τρόπο να τα βάλουμε στο breadboard) αντί για buzzer, το αποτέλεσμα θα είναι πιο εύηχο.

Θα φτιάξουμε Theremin, ένα μουσικό όργανο. Υπάρχουν έτοιμοι κώδικες στο διαδίκτυο τόσο για Theremin με αισθητήρα υπερήχων, όσο και για Theremin με φωτοαντίσταση. Εμείς φορτώνουμε στις πλακέτες έναν έτοιμο κώδικα (ο κώδικας αυτός υπάρχει στο κεφάλαιο των κατασκευών) και τα παιδιά υλοποιούν το κύκλωμα. Δεν ασχολούμαστε με την επεξήγηση του κώδικα, δεν είναι αυτός ο σκοπός αυτής της δραστηριότητας. Θα μπορούσαμε να πούμε ότι ο σκοπός μας είναι να δείξουμε στα παιδιά ότι υπάρχουν αμέτρητοι κώδικες σε διάφορες σελίδες στο διαδίκτυο, του οποίους και μπορούμε να χρησιμοποιήσουμε.

Ένα είναι σίγουρο: αυτή η δεύτερη ώρα θα είναι αρκετά διασκεδαστική για τα παιδιά. Στο τέλος, συζητάμε μαζί τους για το πώς τους φάνηκε αυτή η δραστηριότητα, και όπως πάντα, οι ομάδες αποθηκεύουν τα υλικά.

11^ο Διδακτικό Σενάριο: Αισθητήρας Υπερήχων και Αντλία Νερού με Ρελέ

Στόχοι:

Οι μαθητές/τριες

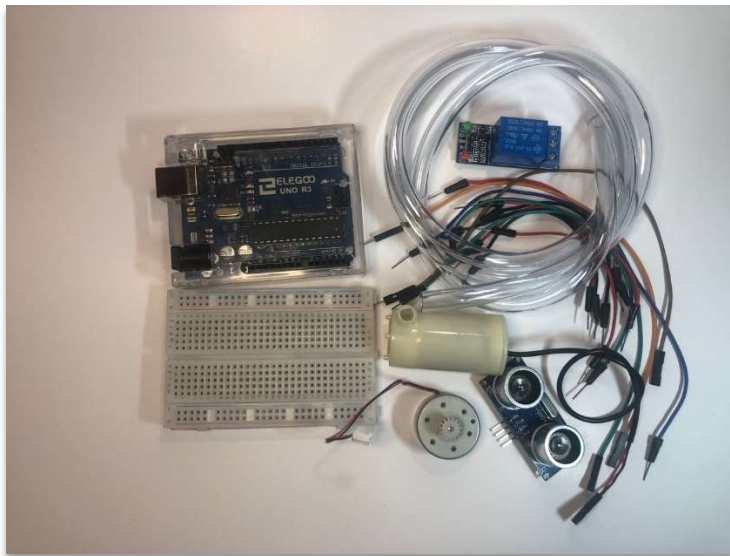
- Να κατανοήσουν τη λειτουργία και τη χρήση του ρελέ
- Να κατανοήσουν τη λειτουργία της αντλίας νερού
- Να συνδυάσουν τον αισθητήρα υπερήχων και την αντλία νερού για να φτιάξουν μελλοντικά μια χρηστική κατασκευή

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 αισθητήρας υπερήχων
- 1 ρελέ
- 1 κινητήρα DC για τις δοκιμές λειτουργίας του κυκλώματος
- 12 καλώδια (τα 3 M-F)

Υλικά ανά ομάδα – 2^ο στάδιο

- 1 αντλία νερού
- 1 δοχείο με νερό



Εικ. 11.1 Υλικά

Πριν αρχίσουμε τα διδακτικά σενάρια με Arduino, κάναμε επίδειξη λειτουργίας μιας κατασκευής που είχαμε ετοιμάσει και λειτούργησε ως δόλωμα στα παιδιά, την αυτόματη βρύση. Ήρθε η ώρα που θα υλοποιήσουν και τα παιδιά το κύκλωμα λειτουργίας της.

Φέρνουμε την κατασκευή, και την θέτουμε σε λειτουργία. Αφήνουμε τα παιδιά να την περιεργαστούν, να δουν το κύκλωμα και να πει το καθένα τη γνώμη του σχετικά με την λειτουργία της. Στο τέλος εξηγούμε πώς ακριβώς δουλεύει: όταν ο αισθητήρας υπερήχων ανιχνεύσει κάποιο αντικείμενο στα 15 cm, και πιο συγκεκριμένα τα χέρια μας, τότε μέσω του ρελέ ενεργοποιεί την εμβαπτισμένη αντλία νερού, η οποία με τη σειρά της στέλνει νερό από το δοχείο μέσα στο οποίο βρίσκεται, στον ελαστικό σωλήνα. Έχουμε δώσει ένα καμπύλο σχήμα στον σωλήνα, και έτσι μοιάζει με βρύση. Άρα, όταν πλησιάζουμε τα χέρια μας στον αισθητήρα υπερήχων, το ρελέ κλείνει το κύκλωμα, ενεργοποιείται η αντλία και από το σωλήνα τρέχει νερό. Όταν απομακρύνουμε τα χέρια μας, το ρελέ ανοίγει το κύκλωμα, και έτσι η αντλία παύει να λειτουργεί, οπότε από το σωλήνα δεν τρέχει νερό.

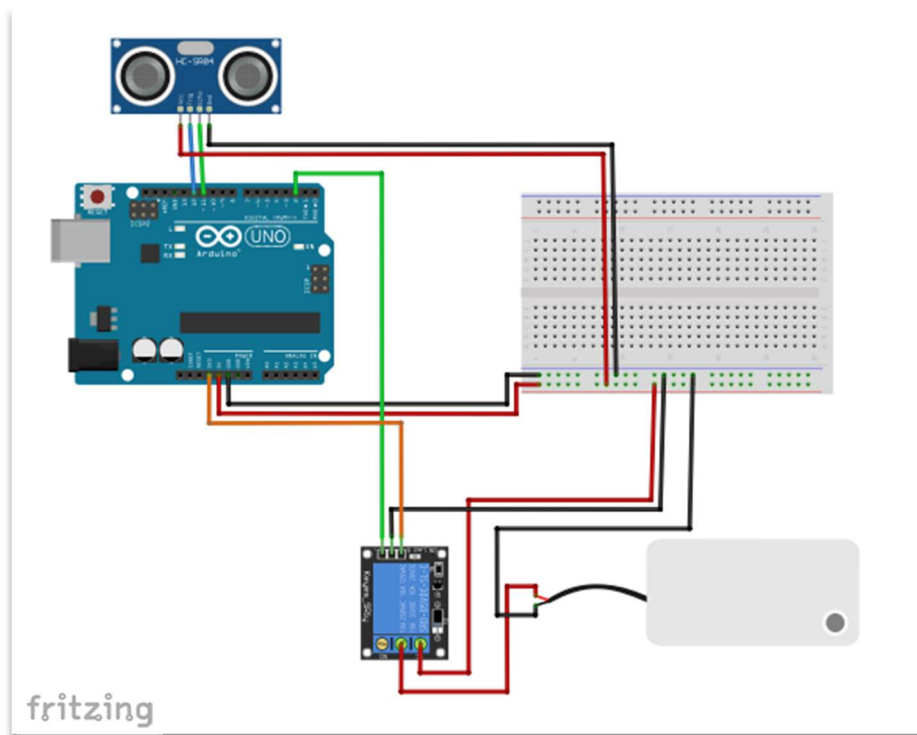
Τι είναι όμως το ρελέ; Είναι ένας διακόπτης, ο οποίος παίρνει σήμα από τον Arduino για το πότε θα ανοίξει ή θα κλείσει ένα κύκλωμα. Έχει επίσης τη δυνατότητα να ελέγχει συσκευές οι οποίες έχουν ανάγκη υψηλότερης τάσης ή έντασης ρεύματος από αυτά που παρέχει ο Arduino, αλλά στη δική μας αντλία νερού που λειτουργεί με 3V-5V, το ρελέ λειτουργεί καθαρά ως διακόπτης.

Η αντλία αποτελείται από έναν κινητήρα DC, κλεισμένο στεγανά σε ένα κέλυφος, ο άξονας του οποίου, όταν τροφοδοτείται με τάση, περιστρέφει μια φτερωτή. Στον θάλαμο που βρίσκεται η φτερωτή, μπαίνει νερό από μια τρύπα, και η περιστρεφόμενη φτερωτή το στέλνει με πίεση στην έξοδο της αντλίας, όπου συνδέουμε έναν εύκαμπτο σωλήνα.

Ο κινητήρας DC έχει έναν άξονα, και όσο τροφοδοτείται με τάση ο κινητήρας, ο άξονας περιστρέφεται σε μια φορά περιστροφής. Αν αλλάξουμε την πολικότητα στους ακροδέκτες του κινητήρα, η φορά περιστροφής του άξονα αντιστρέφεται. Επίσης, όταν μεταβάλλουμε την τάση τροφοδοσίας, μεταβάλλεται και η ταχύτητα περιστροφής του άξονα. Δεν θα επεκταθούμε περισσότερο για το πώς τα επιτυγχάνουμε αυτά και με ποιον εξοπλισμό, εφόσον δεν χρειάζεται να αλλάξουμε ταχύτητα ή φορά περιστροφής στην φτερωτή της αντλίας.

Αναφέραμε την αντλία ως εμβραπιζόμενη. Αυτό σημαίνει ότι είναι βυθισμένη στο νερό, και είναι σχεδιασμένη για να λειτουργεί σε αυτή τη συνθήκη, μέσα στο νερό, και να ψύχεται από αυτό. Αν τη λειτουργήσουμε σε στεγνό περιβάλλον, είναι πολύ πιθανό να υπερθερμανθεί και να καταστραφεί. Για το λόγο αυτό, καλύτερα να ετοιμάσουν τα παιδιά τα κυκλώματά τους, και αντί για αντλία να χρησιμοποιήσουν έναν κινητήρα DC, ή έναν μικρό ανεμιστήρα από παλιό υπολογιστή.

Μοιράζουμε τα υλικά και το κύκλωμα στις ομάδες, και γράφουμε τον κώδικα στον πίνακα. Πιθανόν να χρειαστεί να βοηθήσουμε τα παιδιά με τη σύνδεση των καλωδίων στο ρελέ.



Σχ. 11.1

Ο κώδικας

```
int relayPin = 2;  
#include <SR04.h>  
int trigPin=12;  
int echoPin=11;  
SR04 sr04 = SR04(echoPin, trigPin);  
long a;
```

```

void setup() {
pinMode(relayPin, OUTPUT);
}

void loop() {
a=sr04.Distance();

if(a<15){
digitalWrite(relayPin, HIGH);
}
else{
digitalWrite(relayPin, LOW);
}
}

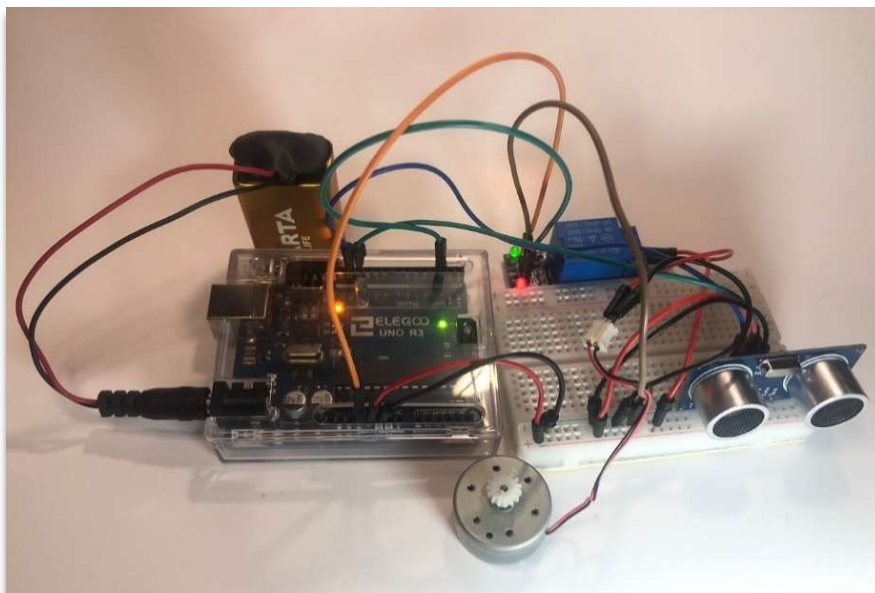
```

Επεξήγηση του κώδικα

`int relayPin = 2;` Συνδέουμε το ρελέ στο pin 2 του Arduino

`digitalWrite(relayPin, HIGH);` Ο Arduino μέσω του ρελέ ενεργοποιεί την αντλία

`digitalWrite(relayPin, LOW);` Ο Arduino μέσω του ρελέ απενεργοποιεί την αντλία



Εικ. 11.2 Υλοποίηση Σεναρίου

Αφού ολοκληρώσουν όλες οι ομάδες τη συνδεσμολογία και διορθωθούν τυχόν σφάλματα, τους δίνουμε από μία αντλία με τον εύκαμπτο σωλήνα της, η οποία αντικαθιστά τον κινητήρα, και από ένα δοχείο με νερό. Φροντίζουμε ώστε τα καλώδια που συνδέουν την αντλία με το breadboard να έχουν μεγάλο μήκος, ή κάνουμε συνδυασμό καλωδίων, ώστε το συνολικό τους μήκος να είναι τέτοιο που η αντλία να είναι μακριά από το υπόλοιπο κύκλωμα, για ευνόητους λόγους. Ζητάμε από τα παιδιά να φροντίσουν ώστε ο εύκαμπτος σωλήνας να επιστρέφει το νερό, που θα στέλνει η αντλία, πίσω στο δοχείο. Έτσι η αίθουσα θα μείνει στεγνή, ενώ ταυτόχρονα το δοχείο θα είναι συνεχώς γεμάτο. Επίσης, ενδεχομένως η μπαταρία να μην μπορεί να καλύψει την εφαρμογή με την αντλία νερού, οπότε καλό θα ήταν να έχουμε και έναν μετασχηματιστή 9V 2A.

Στο τέλος μαζεύουμε τις αντλίες και τα παιδιά αποθηκεύουν τα υπόλοιπα υλικά.

12^ο Διδακτικό Σενάριο: Αισθητήρας Υγρασίας Εδάφους και Αντλία Νερού με Ρελέ

Στόχοι:

Οι μαθητές/τριες

- Να κατανοήσουν τη λειτουργία και τη χρήση του αισθητήρα υγρασίας εδάφους
- Να συνδυάσουν τον αισθητήρα υγρασίας και την αντλία νερού για να φτιάξουν μελλοντικά μια χρηστική κατασκευή

Υλικά ανά ομάδα – 1^ο στάδιο

- 1 Arduino
- 1 μπαταρία 9V και το αντίστοιχο κλιπ με βύσμα
- 1 breadboard
- 1 αισθητήρας υγρασίας εδάφους
- 1 ρελέ
- 1 κινητήρα DC για τις δοκιμές λειτουργίας του κυκλώματος
- 11 καλώδια (τα 3 M-F)
- 1 μικρή γλάστρα (με στεγνό χώμα)

Υλικά ανά ομάδα – 2^ο στάδιο

- 1 αντλία νερού
- 1 δοχείο με νερό

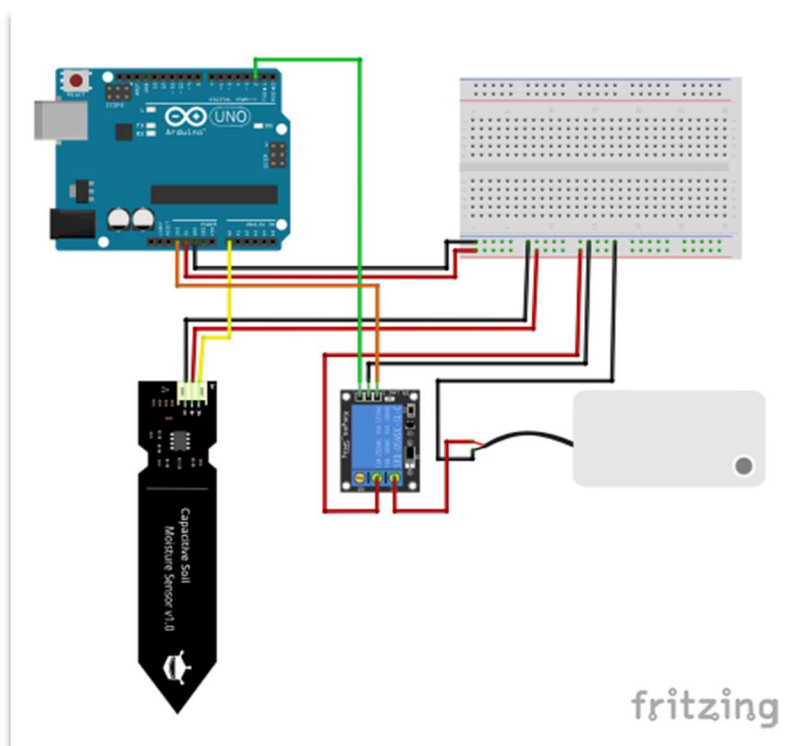


Εικ. 12.1 Υλικά

Είδαμε στο προηγούμενο διδακτικό σενάριο την εφαρμογή με τον αισθητήρα υπερήχων και την αντλία νερού. Θα συνεχίσουμε με μία ακόμη εφαρμογή που χρησιμοποιεί την αντλία νερού, και σίγουρα θα τραβήξει την προσοχή των παιδιών: το αυτόματο πότισμα.

Ο αισθητήρας που θα χρησιμοποιήσουμε είναι ο αισθητήρας υγρασίας εδάφους. Τον χρησιμοποιούμε σε εφαρμογές αυτόματου ποτίσματος. Τον συνδέουμε σε μια αναλογική είσοδο του Arduino, και αυτός για διαφορετικά ποσοστά υγρασίας του εδάφους δίνει διαφορετικές τιμές. Όσο περισσότερη είναι η υγρασία, τόσο πιο μικρή είναι η τιμή που δίνει στον Arduino, και το αντίστροφο. Οι τιμές της ιδανικής υγρασίας κυμαίνονται μεταξύ 280 και 380, εξαρτάται βέβαια και από το φυτό που έχουμε.

Η τιμή που θα θέσουμε εμείς στον κώδικά μας είναι 370. Μοιράζουμε τα υλικά και το σχέδιο του κυκλώματος, και γράφουμε τον κώδικα στον πίνακα. Όπως και στην προηγούμενη εφαρμογή, και για τους λόγους που ήδη εξηγήσαμε, περιμένουμε να ολοκληρώσουν όλες οι ομάδες την κατασκευή του κυκλώματος με τον κινητήρα και όχι με την αντλία.



Σχ. 12.1

Τα παιδιά τοποθετούν κάθετα τον αισθητήρα μέσα στη γλάστρα, μέχρι το επίπεδο που ορίζει η λευκή οριζόντια γραμμή. Το χρώμα της γλάστρας είναι στεγνό, οπότε περιμένουμε ότι ο Arduino θα δώσει εντολή στο ρελέ να κλείσει το κύκλωμα λειτουργίας του κινητήρα, με αποτέλεσμα την περιστροφή του άξονα του κινητήρα. Αν δεν συμβεί αυτό, σημαίνει ότι κάτι έχουν γίνει λάθος στη συνδεσμολογία του κυκλώματος, και αυτό το λάθος το εντοπίζουν τα παιδιά και το διορθώνουν.

Κώδικας

```
int relayPin = 2;
int sensorValue = analogRead(A0);

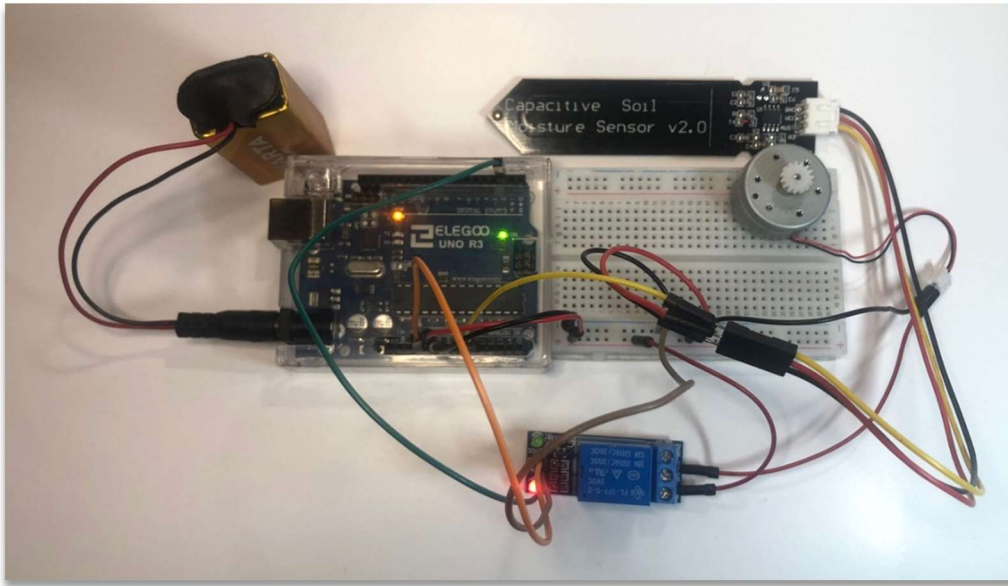
void setup() {
  pinMode(relayPin, OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  if(sensorValue>370){
    digitalWrite(relayPin, HIGH);
  }
  else{
    digitalWrite(relayPin, LOW);
  }
}
```

Επεγήριση κώδικα

`int sensorValue = analogRead(A0);` Τοποθετούμε τον αισθητήρα στην αναλογική είσοδο A0 του Arduino

`if(sensorValue>370){` Αν η τιμή που δίνει ο αισθητήρας είναι μεγαλύτερη από 370, τότε το χρώμα της γλάστρας είναι στεγνό και θέλει πότισμα



Εικ. 12.2 Υλοποίηση Σεναρίου

Αφού τεθούν σε λειτουργία όλοι οι κινητήρες των παιδιών, που σημαίνει ότι η συνδεσμολογία των κυκλωμάτων είναι σωστή, τους δίνουμε και τις αντλίες. Τις εμβαπτίζουμε στο δοχείο με το νερό και ο εύκαμπτος σωλήνας καταλήγει στη γλάστρα. Τότε, σύμφωνα με τον κώδικα, το ρελέ θα ενεργοποιηθεί την αντλία, η οποία μέσω του εύκαμπτου σωλήνα θα στείλει νερό στην γλάστρα. Όταν το χώμα υγρανθεί, ο αισθητήρας θα δώσει μία χαμηλότερη τιμή, ο Arduino θα δώσει εντολή στο ρελέ να ανοίξει το κύκλωμα λειτουργίας της αντλίας, και η αντλία θα απενεργοποιηθεί. Όταν στεγνώσει το χώμα, οι παραπάνω ενέργειες θα επαναληφθούν.

Ιδανικά, κατά την περίοδο των διακοπών των Χριστουγέννων, μπορούμε να αφήσουμε στην αίθουσα δύο ίδιες γλάστρες με το ίδιο φυτό, ποτισμένες, και στη μια από αυτές να προσαρμόσουμε το αυτόματο πότισμα. Η άλλη αναγκαστικά θα αφεθεί απότιστη. Και να παρατηρήσουμε το αποτέλεσμα όταν επιστρέψουμε. Σε αυτή την περίπτωση, η μπαταρία των 9V δεν θα μπορέσει να ανταποκριθεί επαρκώς στην απαίτηση του Arduino για τάση, οπότε η καλύτερη μέθοδος τροφοδοσίας είναι ο μετασχηματιστής 9V 2A DC.

Ολοκλήρωση διδακτικών σεναρίων

Για την πλατφόρμα Arduino υπάρχει πληθώρα αισθητήρων και ενεργοποιητών. Το ζητούμενο στην παρούσα εργασία είναι να μνηθούν οι μαθητές και οι μαθήτριες σε απλές εφαρμογές, με την απλούστερη σύνταξη κώδικα, ώστε να μπορέσουν να κατανοήσουν τις αρχές προγραμματισμού, και ταυτόχρονα, μέσα από αυτή την κατανόηση, να αποκτήσουν την πεποίθηση ότι είναι στο μέτρο του δυνατού να υλοποιήσουν και μόνοι/ες τους κάποια projects, στα οποία θα ενσωματώνουν κάποια στοιχεία από όσα διδάχτηκαν σχετικά με τον Arduino.

Θα ήταν περισσότερο εντυπωσιακό αν παρουσιάζαμε διδακτικά σενάρια με την χρήση πχ οθόνης που θα εμφανίζει κάποιο μήνυμα ή ακόμη και με την κατασκευή οχήματος που αποφεύγει εμπόδια κλπ. Κατά πόσο όμως αυτό θα εξυπηρετούσε τον σκοπό μας, την κατανόηση δηλαδή του κώδικα, και την προσέλκυση των παιδιών στον μαγικό κόσμο του Arduino; Μήπως περίπλοκοι κώδικες, δυσνόητοι στα παιδιά, θα έφερναν το αντίθετο αποτέλεσμα;

Μπορούμε, βέβαια, να ασχοληθούμε και με κάποιο project (όπως κάναμε με το Theremin) που θα έχει κάποιες εντυπωσιακές εφαρμογές, αλλά με την επίγνωση ότι δεν είναι διδακτικό σενάριο, αλλά χρήση ενός έτοιμου κώδικα, τον οποίο χρησιμοποιούμε για ένα επιθυμητό αποτέλεσμα, χωρίς απαραίτητα να μπορούν να τον κατανοήσουν τα παιδιά.

Κατασκευές

Μετά τη θεωρία, έρχεται η πράξη. Και όσα δείξαμε στα διδακτικά σενάρια, θα εφαρμοστούν στην πράξη, ζητώντας από τα παιδιά να υλοποιήσουν κάποια κατασκευή, η οποία θα περιέχει κάποιον από τους αυτοματισμούς που είδαμε.

Αν κάποιος/α δεν προσπαθήσει να κατασκευάσει κάτι αντίστοιχο, δεν θα μάθει ποτέ πόσο απέχει η θεωρία από την πράξη. Και καλό θα ήταν, όταν ζητάμε κάτι από τους μαθητές μας και τις μαθήτριές μας, να το έχουμε υλοποιήσει πρώτα εμείς, ώστε να γνωρίζουμε ποιες δυσκολίες θα συναντήσουν τα παιδιά, και να μπορέσουμε να τους δώσουμε τα εφόδια να τις ξεπεράσουν.

Τι είδους κατασκευές θα μπορούσαν να υλοποιήσουν τα παιδιά;

Ενδεικτικά, ακολουθούν κατασκευές που χρησιμοποιούν τον ίδιο κώδικα ή βασίζονται σε κάποιον από αυτούς που συναντήσαμε στα διδακτικά σενάρια.

Πρόκειται για 5 κατασκευές, και για την καθεμία από αυτές, εκτός από μια σύντομη περιγραφή της λειτουργίας της και φωτογραφίες, αναφέρονται τα υλικά που χρησιμοποιήθηκαν, το κύκλωμα λειτουργίας και ο κώδικας.

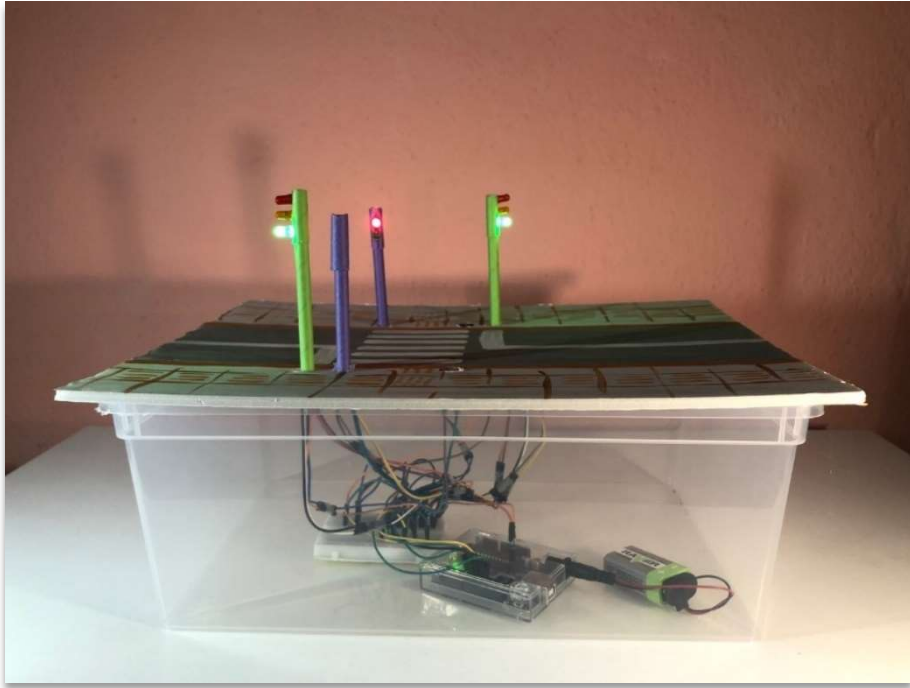
Φανάρια Κυκλοφορίας Οχημάτων και Πεζών



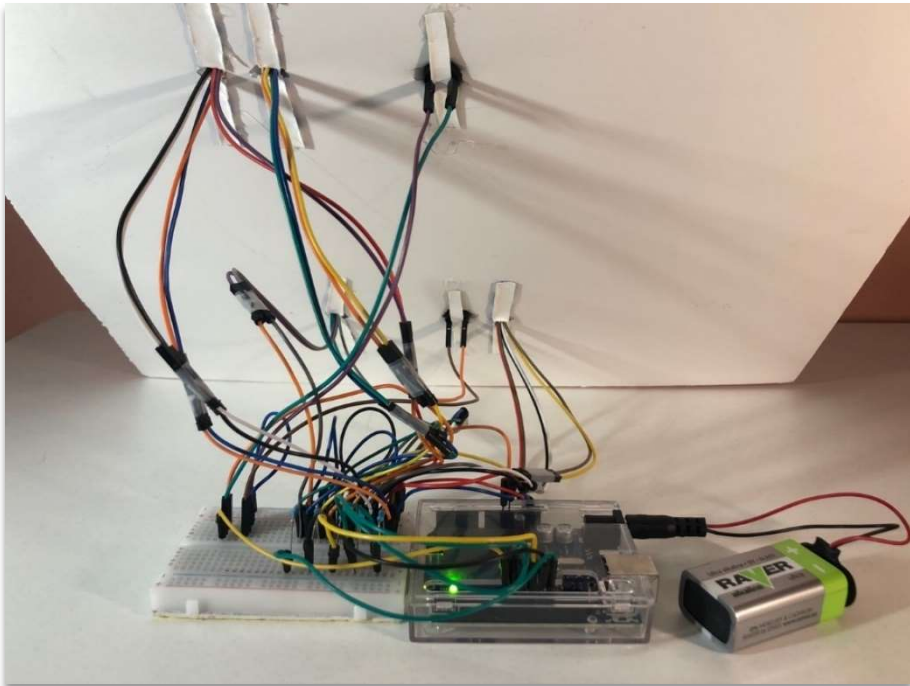
Τα φανάρια σε λειτουργία



Από ψηλά



Από χαμηλά



Το κύκλωμα

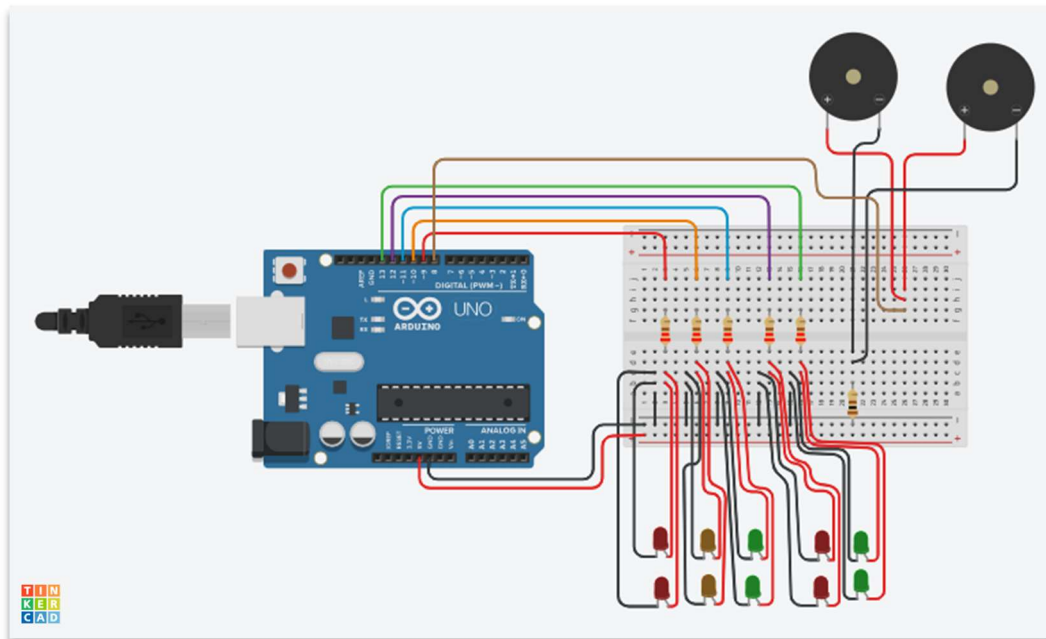
Περιγραφή

Πρόκειται για τα φανάρια κυκλοφορίας οχημάτων και πεζών, σε μια διάβαση, όπου όταν γίνεται πράσινο το φανάρι των πεζών, ακούγεται διακεκομμένα ένας ήχος για τους ανθρώπους με προβλήματα όρασης.

Γενικά στις κατασκευές που χρησιμοποιούμε τον Arduino, πρέπει να έχουμε υπόψη μας ότι χρειάζεται χώρος κάτω ή πίσω από την κατασκευή μας που θα είναι ο χώρος του Arduino, του breadboard και των καλωδίων. Στη συγκεκριμένη υλοποίηση, το πιο δύσκολο σημείο είναι το πέρασμα των καλωδίων μέσα από το καλαμάκι. Για την τροφοδοσία αρκεί μία μπαταρία των 9V. Το κύκλωμα που ακολουθεί φαίνεται περίπλοκο, αλλά δεν είναι.

Υλικά

- 1 Arduino
- 1 breadboard
- 1 μπαταρία 9V
- 4 κόκκινα LED
- 4 πράσινα LED
- 2 πορτοκαλί LED
- 2 buzzer
- 5 αντιστάσεις 220 Ω
- 1 αντίσταση 100 Ω
- 37 καλώδια
- Μακετόχαρτο 30cm x 40cm
- Χάρτινα Καλαμάκια για Γρανίτα
- Χρώματα για το βάψιμο
- Θερμόκολλα



Κύκλωμα Φαναριών

Ο κώδικας

```
int ledRed1 = 9; // κόκκινο φανάρι οχημάτων στο pin 9
int ledOrange1 = 10; // πορτοκαλί φανάρι οχημάτων στο pin 10
int ledGreen1 = 11; // πράσινο φανάρι οχημάτων στο pin 11
int ledRed2 = 12; // κόκκινο φανάρι πεζών στο pin 12
int ledGreen2 = 13; // πράσινο φανάρι πεζών στο pin 13
int buzzer = 8; // buzzer στο pin 8

void setup() {
  pinMode(ledRed1, OUTPUT); // Κόκκινο οχημάτων, ΕΞΟΔΟΣ
  pinMode(ledOrange1, OUTPUT); // Πορτοκαλί οχημάτων, ΕΞΟΔΟΣ
  pinMode(ledGreen1, OUTPUT); // Πράσινο οχημάτων, ΕΞΟΔΟΣ
  pinMode(ledRed2, OUTPUT); // Κόκκινο πεζών, ΕΞΟΔΟΣ
  pinMode(ledGreen2, OUTPUT); // Πράσινο πεζών, ΕΞΟΔΟΣ
  pinMode(buzzer, OUTPUT); // Buzzer, ΕΞΟΔΟΣ
}

void loop() {
  // πράσινο οχημάτων για 8 δευτερόλεπτα
  digitalWrite(ledRed1, LOW);
  digitalWrite(ledOrange1, LOW);
  digitalWrite(ledGreen1, HIGH);
  digitalWrite(ledRed2, HIGH);
  digitalWrite(ledGreen2, LOW);
  digitalWrite(buzzer, LOW);
  delay(8000);
}
```

```

// πορτοκαλί οχημάτων για 2 δευτερόλεπτα
digitalWrite(ledRed1, LOW);
digitalWrite(ledOrange1, HIGH);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, HIGH);
digitalWrite(ledGreen2, LOW);
digitalWrite(buzzer, LOW);
delay(2000);
// κόκκινο οχημάτων και κόκκινο πεζών για 0,7 δευτερόλεπτα
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, HIGH);
digitalWrite(ledGreen2, LOW);
digitalWrite(buzzer, LOW);
delay(700);
// κόκκινο οχημάτων και πράσινο πεζών για 2 δευτερόλεπτα
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, HIGH);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, LOW);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, HIGH);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, LOW);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);

```

```

digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, HIGH);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, LOW);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, HIGH);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, LOW);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, HIGH);
delay(200);
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, LOW);
digitalWrite(ledGreen2, HIGH);
digitalWrite(buzzer, LOW);
delay(200);
// κόκκινο οχημάτων και κόκκινο πεζών για 0,7 δευτερόλεπτα
digitalWrite(ledRed1, HIGH);
digitalWrite(ledOrange1, LOW);
digitalWrite(ledGreen1, LOW);
digitalWrite(ledRed2, HIGH);
digitalWrite(ledGreen2, LOW);
digitalWrite(buzzer, LOW);
delay(700);
}

```


Theremin



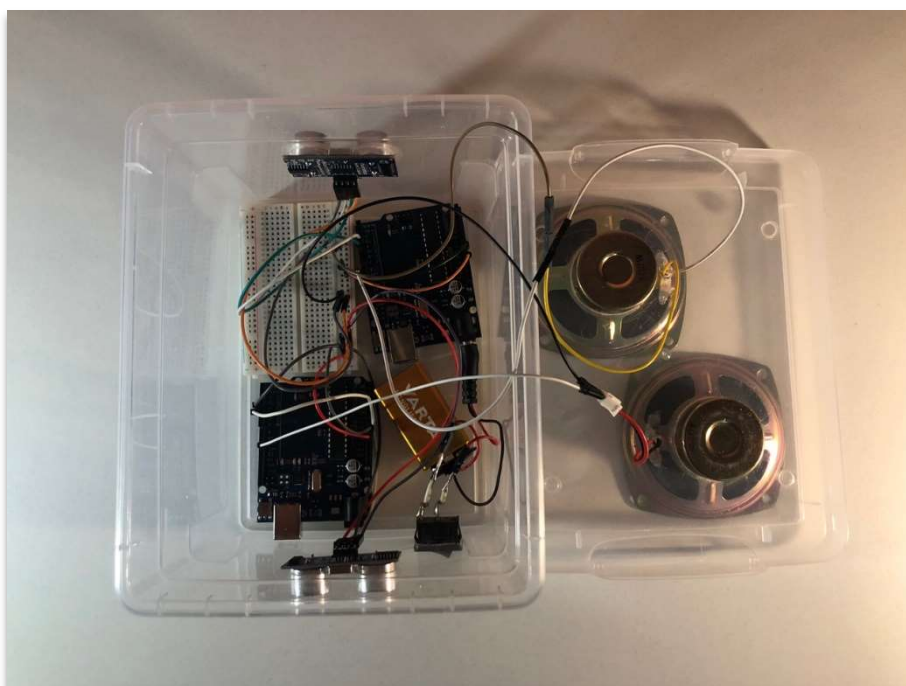
Theremin με διακόπτη On/Off



Εξωτερικά



Τα ηχεία



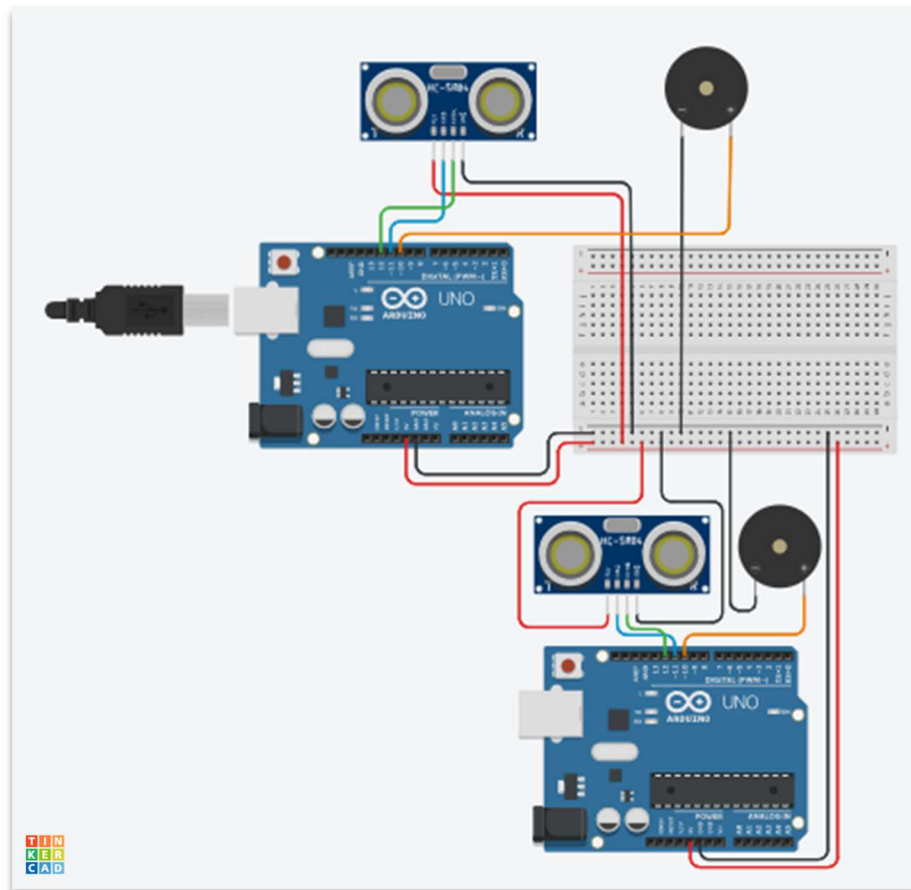
Το εσωτερικό

Περιγραφή

Το Theremin (αρχικά αιθερόφωνο) είναι ένα ηλεκτρονικό μουσικό όργανο που εφευρέθηκε το 1920. Είναι το μόνο μουσικό όργανο το οποίο μπορεί να παιχτεί μακρόθεν, δηλ. «στον αέρα», χωρίς να αγγίζεται από τον ερμηνευτή (Wikipedia). Η κατασκευή μας «φιλοξενείται» σε ένα διάφανο κουτί, στο οποίο έχουμε δύο Arduino, δύο αισθητήρες υπερήχων και ένα breadboard. Η τροφοδοσία γίνεται από μία μπαταρία των 9V, με την εξής ιδιαιτερότητα: για λόγους οικονομίας χώρου, κρίθηκε σκόπιμο να χρησιμοποιηθεί ένα breadboard, οπότε ο ένας Arduino τροφοδοτείται από την μπαταρία, και ο δεύτερος Arduino «τροφοδοτείται» από το breadboard. Η μόνη δυσκολία ήταν η διάνοιξη οπών τόσο για τα ηχεία όσο και για τους αισθητήρες στο πλαστικό κουτί. Επειδή μόλις τροφοδοτηθεί το Theremin, αρχίζει να ηχεί, κρίθηκε απαραίτητο να μπει και ένας διακόπτης on/off.

Υλικά

- 2 Arduino
- 1 breadboard
- 1 μπαταρία 9V
- 2 ηχεία
- 16 καλώδια
- 1 διακόπτης on/off
- 1 κουτί



Το κύκλωμα

Ο κώδικας

```

const int trigger = 5;
const int echo = 4;
const int piezo = 10;
int distance = 0;
int distanceHigh = 0;
int lengthOfScale = 0;
int note = 0;

//C Major scale
int scale[] = {
  131, 147, 165, 175, 196, 220, 247, 262, 294,
  330, 349, 392, 440, 494, 523, 587, 659, 698,
  784, 880, 988, 1047
};

void setup() {
  pinMode(trigger, OUTPUT);
  pinMode(echo, INPUT);

```

```

while (millis() < 5000) {
    digitalWrite(trigger, HIGH);
    digitalWrite(trigger, LOW);
    distance = pulseIn(echo, HIGH);
    if (distance > distanceHigh) {
        distanceHigh = distance;
    }
}
for (byte i = 0; i < (sizeof(scale) / sizeof(scale[0])); i++) {
    lengthOfScale += 1;
}
}

void loop() {
    digitalWrite(trigger, HIGH);
    digitalWrite(trigger, LOW);
    distance = pulseIn(echo, HIGH);
    note = map(distance, 250, distanceHigh, scale[0], scale[lengthOfScale
- 1]);
    for (byte j = 0; j < (lengthOfScale); j++) {
        if (note == scale[j]) {
            tone(piezo, note);
            break;
        }
        else if (note > scale[j] && note < scale[j + 1]) {
            note = scale[j];
            tone(piezo, note);
            break;
        }
    }
}
delay(30);
}

```

Έξυπνος Κάδος



Ο Έξυπνος Κάδος



Το εσωτερικό



Το κύκλωμα κάτω από το καπάκι



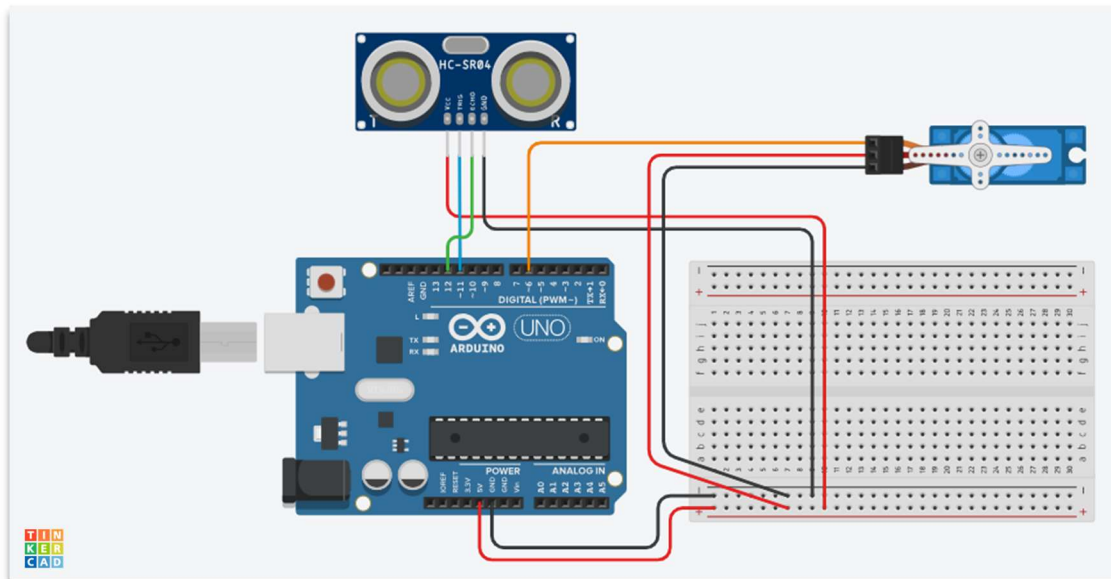
Ο σερβοκινητήρας

Περιγραφή

Μία ιδέα δανεισμένη από το διαδίκτυο, που δουλεύει με τον κώδικα του 10^{ου} διδακτικού σεναρίου. Η λειτουργία είναι απλή: όταν ο αισθητήρας υπερήχων αντιληφθεί κάποιο αντικείμενο σε απόσταση μικρότερη των 10 εκατοστών, περιστρέφει δεξιόστροφα τον άξονα του σερβοκινητήρα κατά 70°. Αυτή η περιστροφική κίνηση «τραβά» το καπάκι του κάδου προς τα κάτω, και μετά από 3 δευτερόλεπτα το καπάκι επιστρέφει στην θέση του. Η μεγαλύτερη δυσκολία ήταν να βρεθεί η ακριβής θέση τοποθέτησης του servo. Επίσης, αν η μπαταρία δεν είναι «γεμάτη», προκαλούνται αρρυθμίες στην λειτουργία του κάδου, κατά κάποιον τρόπο λειτουργεί αυτόνομα, ανοίγει και κλείνει το καπάκι του ακόμη και αν δεν βρεθεί αντικείμενο μπροστά στον αισθητήρα.

Υλικά

- 1 Arduino
- 1 breadboard
- 1 μπαταρία 9V
- 1 σερβοκινητήρας
- 1 αισθητήρας υπερήχων
- 9 καλώδια
- 1 κάδος
- Θερμόκολλα
- 1 μικρό κουτί για τον Arduino, το breadboard και την μπαταρία



Το κύκλωμα του έξυπνου κάδου

Ο κώδικας

```

#include <SR04.h>
#include <Servo.h>
int trigPin=11;
int echoPin=12;
SR04 sr04 = SR04(echoPin, trigPin);
long a;
Servo myServo;
int i;

void setup() {
  myServo.attach(6);
}

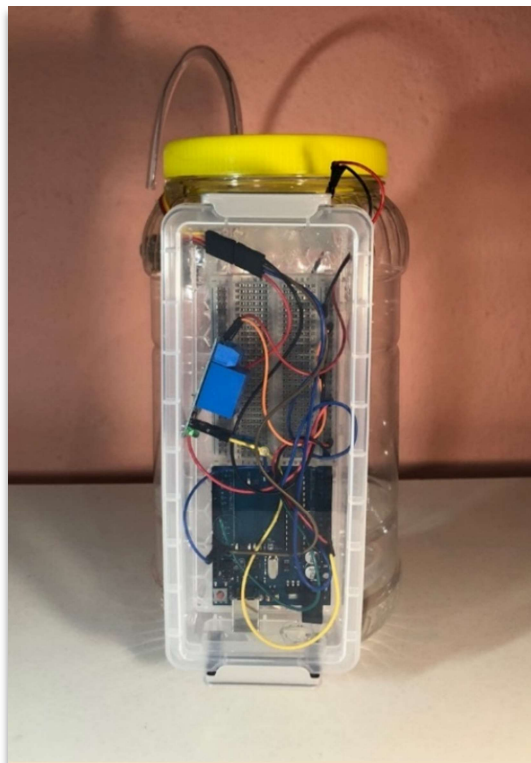
void loop() {
  a=sr04.Distance();
  if(a<10){
    myServo.write(0);
    delay(3000);
  }
  else{
    myServo.write(70);
  }
}

```


Αυτόματη Βρύση



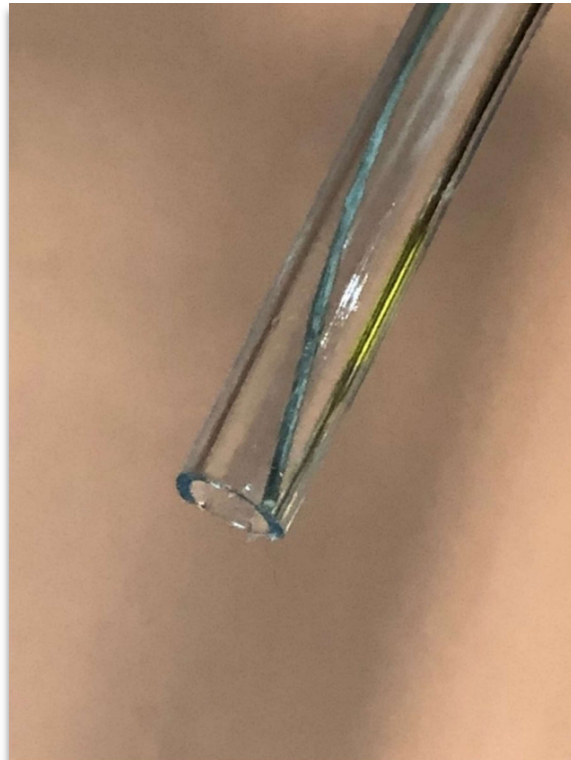
Η Αυτόματη Βρύση



Το Κύκλωμα



Κατασκευαστική Λεπτομέρεια



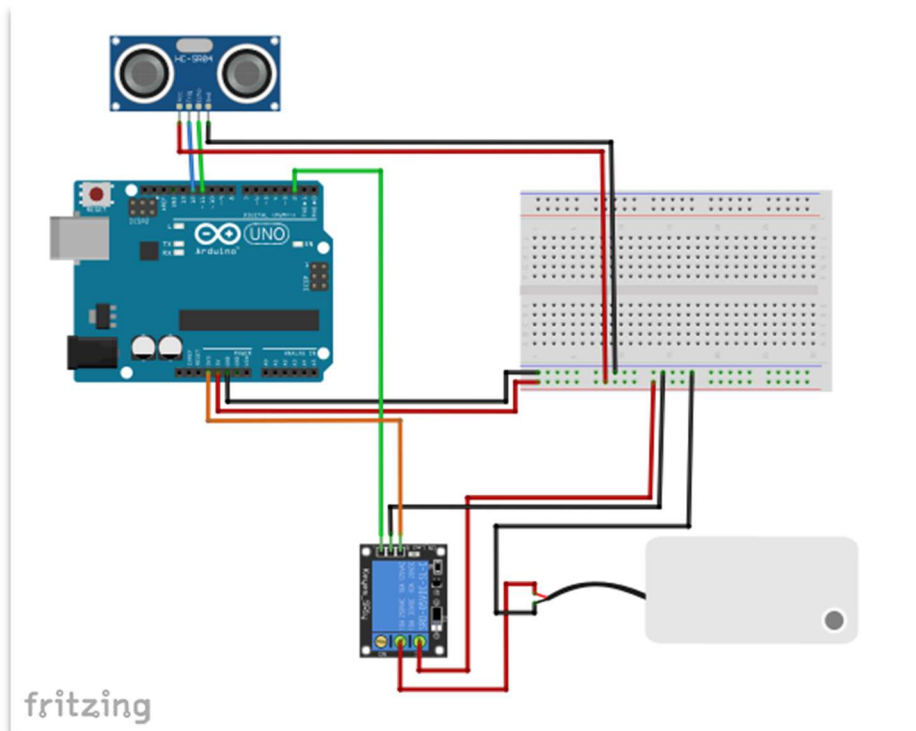
Το σύρμα μέσα στον σωλήνα

Περιγραφή

Μια εφαρμογή που την συναντούμε συνήθως σε χώρους εστίασης, καφετέριες κλπ. Δεν χρειάζεται να ανοίξουμε την βρύση χειροκίνητα, απλά βάζουμε τα χέρια μας κάτω από τη βρύση και τρέχει το νερό. Χρησιμότετη κατασκευή για το σπίτι, όταν υπάρχει διακοπή νερού. Χρησιμοποιούμε το κύκλωμα του 11^{ου} διδακτικού σεναρίου, με τον αισθητήρα υπερήχων και την αντλία νερού. Η μεγαλύτερη δυσκολία αφορούσε στην λειτουργία: όλα έγιναν σωστά, τόσο η συνδεσμολογία όσο και ο κώδικας. Όταν ο αισθητήρας αντιλαμβανόταν κάποιο αντικείμενο μπροστά του, έτρεχε το νερό, αλλά το ρελέ δεν άνοιγε το κύκλωμα της αντλίας όταν το αντικείμενο απομακρυνόταν. Το σφάλμα ήταν η χρήση μπαταρίας για την τροφοδοσία. Ενώ όλα λειτουργούσαν τέλεια στις άνυδρες δοκιμές, δεν συνέβαινε το ίδιο όταν γέμιζε το δοχείο με νερό. Τελικά το πρόβλημα λύθηκε όταν την τροφοδοσία ανέλαβε ο μετασχηματιστής 9V 2A DC. Χρήσιμη λεπτομέρεια για την κλίση του εύκαμπτου σωλήνα, δανεισμένη από το διαδίκτυο: βάζουμε σύρμα μέσα στον σωλήνα, και με αυτόν τον τρόπο σχηματίζουμε την καμπύλη.

Υλικά

- 1 Arduino
- 1 μετασχηματιστής 9V 2A DC
- 1 breadboard
- 1 αισθητήρας υπερήχων
- 1 ρελέ
- 1 αντλία νερού
- 12 καλώδια
- 1 δοχείο
- 1 μικρό κουτί για τον Arduino και το breadboard



Το κύκλωμα

Ο κώδικας

```

int relayPin = 2;
#include <SR04.h>
int trigPin=12;
int echoPin=11;
SR04 sr04 = SR04(echoPin, trigPin);
long a;
void setup() {
  pinMode(relayPin, OUTPUT);
}

void loop() {
  a=sr04.Distance();

  if(a<8){
    digitalWrite(relayPin, HIGH);
  }
  else{
    digitalWrite(relayPin, LOW);
  }
}

```

Αυτόματο Πότισμα



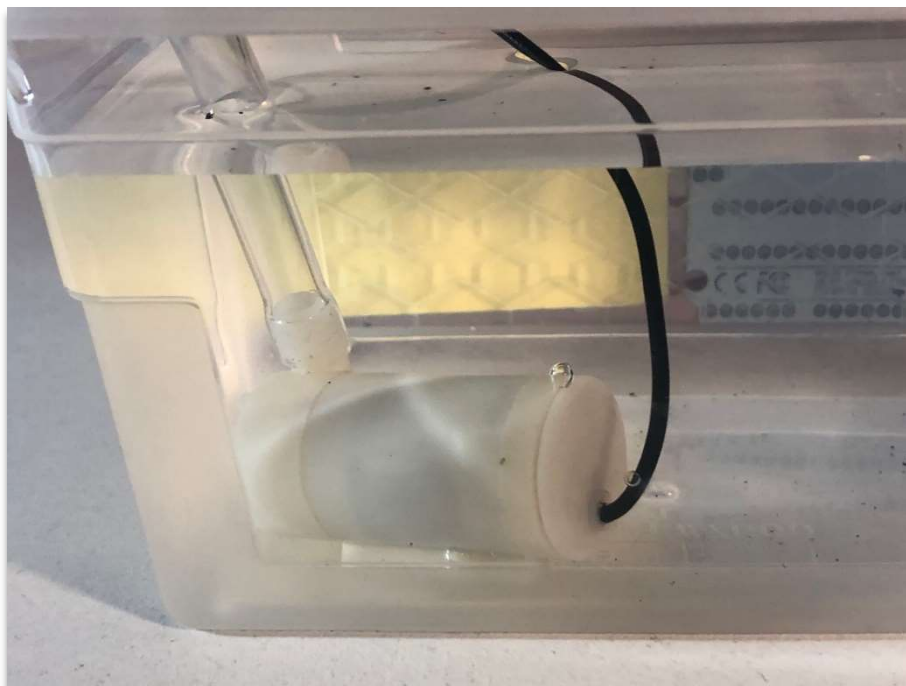
Το Αυτόματο Πότισμα και η Γλάστρα



Το κύκλωμα



Ο Αισθητήρας Υγρασίας Εδάφους



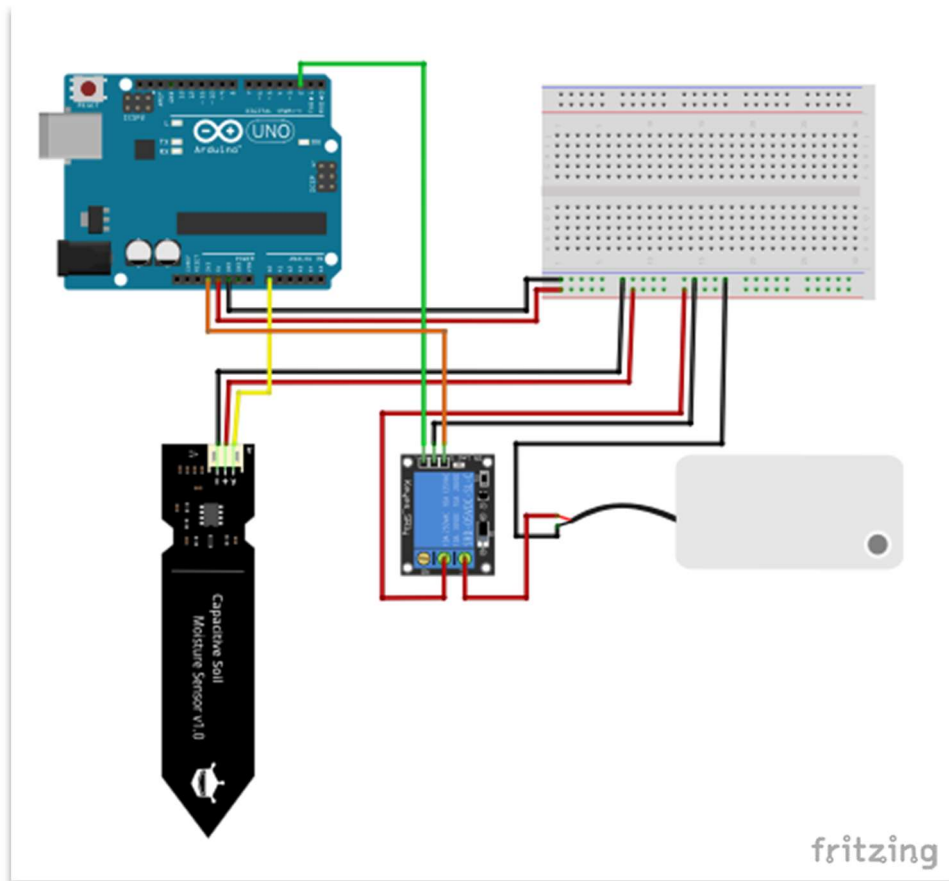
Η Αντλία Νερού

Περιγραφή

Μία εφαρμογή του 12^ο διδακτικού σεναρίου με σχετικά εύκολο κατασκευαστικό μέρος, ιδανικό για γλάστρες. Βάζουμε τον αισθητήρα υγρασίας στο χώμα, και όταν η τιμή που στέλνει στον Arduino πέσει κάτω από ένα όριο, τότε το ρελέ ενεργοποιεί την αντλία και αρχίζει το πότισμα. Όταν η υγρασία φτάσει στο όριο που προαναφέρθηκε, τότε το ρελέ ανοίγει το κύκλωμα της αντλίας, και το πότισμα σταματάει. Σε πραγματική και λειτουργική κατασκευή, σίγουρα χρειαζόμαστε ένα μεγαλύτερο δοχείο νερού. Ο μετασχηματιστής 9V 2A DC είναι απαραίτητος για την τροφοδοσία.

Υλικά

- 1 Arduino
- 1 μετασχηματιστής 9V 2A DC
- 1 breadboard
- 1 αισθητήρας υγρασίας εδάφους
- 1 ρελέ
- 1 αντλία νερού
- 11 καλώδια
- 1 μικρή γλάστρα (με στεγνό χώμα)
- 1 δοχείο
- 1 μικρό κουτί για τον Arduino και το breadboard



Το Κόκλωμα

Ο Κώδικας

```
int relayPin = 2;
int sensorValue = analogRead(A0);

void setup() {
  pinMode(relayPin, OUTPUT);
}

void loop() {
  sensorValue = analogRead(A0);
  if(sensorValue>370){
    digitalWrite(relayPin, HIGH);
  }
  else{
    digitalWrite(relayPin, LOW);
  }
}
```


Ολοκλήρωση Κατασκευών (ή αλλιώς Trial & Error)

Για να υλοποιηθούν οι παραπάνω κατασκευές χρειάστηκαν τα υλικά που αναφέρονται σε καθεμία από αυτές, με την επισήμανση ότι ο αριθμός των καλωδίων είναι ο ελάχιστος απαιτούμενος. Σε κάποιες περιπτώσεις που θα χρειαστεί κάποιο εξάρτημα να είναι μακριά από τον Arduino και το breadboard, ή θα έχουμε καλώδια μεγάλου μήκους, ή θα ενώσουμε μικρότερα καλώδια.

Εκτός όμως από τα υλικά, χρειάστηκε αρκετός χρόνος, και πολλή όρεξη. Αρκετή σκέψη για το πώς θα αρχίσει μια κατασκευή, πώς θα προχωρήσει, ποιο βήμα θα γίνει πρώτο, ποιο θα ακολουθήσει κλπ. Και πολλές δοκιμές, πχ για το πώς θα γίνουν τα φανάρια, πώς θα είναι σταθερά πάνω στο μακετόχαρτο, πού ακριβώς θα μπει ο servo στο καπάκι του κάδου, αν δουλεύει η κάθε κατασκευή...

Κυρίως όμως, για να φτάσουμε σε ένα επιτυχημένο αποτέλεσμα, είναι σίγουρο πως θα χρειαστεί να περάσουμε από πολλές αποτυχημένες προσπάθειες. Ενώ τα έχουμε κάνει όλα τέλεια και βάσει σχεδίου, τελικά δεν λειτουργεί όπως θα θέλαμε. Και ψάχνουμε να βρούμε τι φταίει: βάλουμε κάποιο καλώδιο σε λάθος pin του Arduino, έχουμε κάνει λάθος στον κώδικα, δεν είναι γεμάτη η μπαταρία μας; Το καλό είναι ότι υπάρχει λύση για όλα, αργά ή γρήγορα βρίσκουμε το λάθος, το διορθώνουμε και προχωράμε παρακάτω.

Αν υπήρχε κάποια παλιά κινέζικη παροιμία, που να εκφράζει σε δύο γραμμές τα παραπάνω, θα έλεγε το εξής: «για να φτάσεις στο μονοπάτι της επιτυχίας, πρέπει να διασχίσεις τις λεωφόρους της αποτυχίας».

Εκπαίδευση STEAM και Arduino - Συμπεράσματα

Η διδασκαλία και υλοποίηση των διδακτικών σεναρίων με Arduino έδειξαν στην πράξη ότι ο Arduino είναι μια πολύ επιτυχημένη επιλογή για την διδασκαλία των STEAM στη δευτεροβάθμια εκπαίδευση, για τους παρακάτω λόγους:

Διαθεματικότητα STEAM

Η δημιουργία εφαρμογών με Arduino μπορεί να συνδυάζει πολλά γνωστικά αντικείμενα που σχετίζονται με τα αναλυτικά προγράμματα σπουδών στο γυμνάσιο και στο λύκειο. Έτσι, εκτός από τη διδασκαλία της Πληροφορικής και της Ηλεκτρονικής, θα μπορούσε εύκολα να χρησιμοποιηθεί στη διδασκαλία της Φυσικής, της Ηλεκτρολογίας, της Μηχανολογίας, των Μαθηματικών και ιδιαίτερα στη διδασκαλία της Τεχνολογίας.

Διερευνητική μάθηση

Ένα χαρακτηριστικό προτέρημα του Arduino και των εφαρμογών του είναι ότι ευνοεί τη διαδικασία της Διερευνητικής Μάθησης. Τα παιδιά εφαρμόζουν στην πράξη θεωρητικές γνώσεις που έχουν ήδη αποκτήσει, και με βάση αυτές ανακαλύπτουν και εφαρμόζουν νέες, αποκτώντας ταυτόχρονα δεξιότητες που θα τα βοηθήσουν και στην μετέπειτα ζωή τους.

Ομαδοσυνεργατική Διδασκαλία

Η εκπαίδευση των μαθητών/τριών με τον Arduino ευνοεί την ομαδοσυνεργατική διδασκαλία αφού εκ των πραγμάτων τα παιδιά δουλεύουν σε ομάδες. Μπορούμε να αναθέτουμε ρόλους στα μέλη των ομάδων, και οι ρόλοι αυτοί θα μπορούν να εναλλάσσονται στα διάφορα projects, ώστε όλοι και όλες να αποκτούν τις γνώσεις και τις δεξιότητες του κάθε ρόλου.

Ενδιαφέρον για τα παιδιά

Οι εφαρμογές με Arduino, ειδικά αυτές που αναπτύσσονται στα διδακτικά σενάρια και είναι εσκεμμένα απλουστευμένες και απόλυτα κατανοητές από τα παιδιά, ενεργοποιούν την περιέργεια των παιδιών, τους δίνουν την ευκαιρία να μπορέσουν να τροποποιήσουν έναν κώδικα ώστε να επιτύχουν ένα διαφορετικό αποτέλεσμα,

ενισχύουν την αυτοπεποίθησή τους και ανοίγουν νέους ορίζοντες στην επιλογή επαγγελματικού προσανατολισμού.

Προβληματισμοί

Για να διδάξω τα σενάρια αυτά στους μαθητές και στις μαθήτριές μου, χρειάστηκε ένας ελάχιστος εξοπλισμός αποτελούμενος από πλακέτες Arduino, καλώδια, LED, αντιστάσεις, αισθητήρες και ενεργοποιητές. Το κόστος όλων αυτών το επωμίστηκα προσωπικά, επειδή ήθελα τον δικό μου ενθουσιασμό να τον μεταφέρω και στα παιδιά. Η άλλη επιλογή ήταν να ζητήσω από το σχολείο να προμηθευτεί και να μου διαθέσει αυτόν τον εξοπλισμό. Δεν γνωρίζω αν αυτό θα ήταν δυνατό, ή, αν σε όλα τα σχολεία θα μπορούσε να γίνει κάτι τέτοιο. Μια πρόταση ήταν να βρεθεί ιδιώτης χορηγός. Από ποιον; Και γιατί;

Αυτό είναι και το πρώτο ερώτημα: με ποιον τρόπο θα μπορούσε να χρηματοδοτηθεί η αγορά εξοπλισμού, όταν κάποιος/α εκπαιδευτικός έχει τις γνώσεις και τη διάθεση να διδάξει STEAM με τη χρήση Arduino;

Σε αυτό το σημείο τίθεται και το δεύτερο ερώτημα: Πώς αποκτά τις απαραίτητες γνώσεις ο/η εκπαιδευτικός που έχει τη διάθεση να μάθει και με τη σειρά του να προσφέρει αυτές τις γνώσεις στους μαθητές και στις μαθήτριές του; Με προσωπικό κόστος και στον ελεύθερο χρόνο του, παρακολουθώντας ένα μεταπτυχιακό; Παρακολουθώντας κάποιο σεμινάριο (ενδεχομένως χωρίς κόστος, υπάρχουν και δωρεάν σεμινάρια) στον ελεύθερο χρόνο του, με προσωπική του πρωτοβουλία; Ή μήπως οφείλει το Υπουργείο Παιδείας να εκπαιδεύσει τους/τις εκπαιδευτικούς και να εξοπλίσει όλα τα σχολεία, ώστε όλοι οι μαθητές και όλες οι μαθήτριες να έχουν τη δυνατότητα να εκπαιδευτούν στα STEAM με τη χρήση Arduino;

Πίνακας Κόστους Υλικών

Στον παρακάτω πίνακα αναγράφονται οι τιμές των υλικών που χρησιμοποιήθηκαν στα διδακτικά σενάρια και τις κατασκευές. Αγοράστηκαν από καταστήματα της Ελλάδας, της Γερμανίας και της Κίνας.

Υλικό	Κόστος / Ποσότητα Υλικών	Τιμή μονάδας
Arduino UNO (κλώνος)	42,72 € / 5 τεμ.	8,55 €
Θήκη Arduino	7,14 € / 6 τεμ.	1,19 €
Πολύμετρο	12,80 € / 1 τεμ.	12,80 €
Breadboard (μεσαίο)	8,84 € / 5 τεμ.	1,77 €
Μπαταρία 9V	2,80 € / 1 τεμ.	2,80 €
Κλιπ μπαταρίας με ακροδέκτες	1,20 € / 6 τεμ.	0,20 €
Κλιπ μπαταρίας με βύσμα	2,40 € / 6 τεμ.	0,40 €
Αντιστάσεις	0,85 € / 100 τεμ.	0,0085 €
LED	5,18 € / 200 τεμ.	0,0259 €
Καλώδια μικρά (65 τεμ)	5,01 € / 4 σετ	1,25 € (65 τεμ)
Καλώδια 20cm	1,76 € / 20 τεμ	0,088 €
Καλώδια 30cm	2,39 € / 20 τεμ.	0,1195 €
Ποτενσιόμετρο με κουμπί	2,34 € / 5 τεμ.	0,468 €
Push Button	0,59 € / 20 τεμ.	0,0295 €
Buzzer	1,80 € / 6 τεμ.	0,30 €
Φωτοαντίσταση	1 € / 20 τεμ.	0,05 €
Αισθητήρας Υπερήχων	10,50 € / 6 τεμ.	1,75 €
Σερβοκινητήρας	8,47 € / 5 τεμ.	1,69 €
Αντλία + Ρελέ + Αισθ. Υγρασίας	22,37 € / 8 σετ	2,80 € / σετ

Βιβλιογραφία

- Παπάζογλου, Π - Λιωνής, Σ-Π. (2021). *Ανάπτυξη Εφαρμογών με το Arduino*. Θεσσαλονίκη: Τζιόλα.
- Γίδας Ε. - Μαντζάνας, Γ. (2021). *Εφαρμογές Arduino στο σχολικό εργαστήριο για το περιβάλλον*. Τρίκαλα: 1ο ΕΠΑΛ Τρικάλων.
- ΣΕΒ, Τ. Α. (2021, Μάρτιος 16). Παιδεία STEM για καινοτομία και ευημερία. *Special Report Εκπαίδευση STEM*, σσ. 1-28.

Πηγές

Πηγές Σύνταξης Κώδικα Αισθητήρων & Ενεργοποιητών

elegoo.com

lastminuteengineers.com

www.vodafonegenerationnext.gr

Πηγή κώδικα Theremin

www.hackster.io/pollux-labs/arduino-theremin-with-a-minor-pentatonic-b56de4

Πηγή Εικόνας Χρωματικού Κώδικα

Resistor Color Codes Guide | Simply Smarter Circuitry Blog (circuitspecialists.com)

Λογισμικό

Η ανάπτυξη των εφαρμογών με Arduino έγινε στο Arduino IDE (www.arduino.cc), με τη γλώσσα Wiring C.



Κυκλώματα – Κατασκευές - Εικόνες - Φωτογραφίες

Τα κυκλώματα που χρησιμοποιήθηκαν στα διδακτικά σενάρια σχεδιάστηκαν στο Tinkercad (www.tinkercad.com), ενώ αυτά για τα οποία δεν υπήρχαν διαθέσιμα εξαρτήματα στο Tinkercad, δημιουργήθηκαν με το fritzing (fritzing.org).



Τα κυκλώματα, οι κατασκευές, οι εικόνες και οι φωτογραφίες έχουν παραχθεί από την υπογράφουσα.

Ευχαριστίες

Ευχαριστώ τον Επιβλέποντα Καθηγητή μου, κ. Χρήστο Υφούλη, για την εμπιστοσύνη που μου έδειξε και για την καθοδήγησή του, όλους τους Καθηγητές του Μεταπτυχιακού για τις γνώσεις που μου μετέδωσαν, και τον Σύζυγο και τα Παιδιά μου, χωρίς την συμπαράσταση των οποίων δεν θα έφτανα μέχρι εδώ.

Όλγα Μαλαμούδη

Εκπαιδευτικός

Οχημάτων Τ.Ε.