



1^ο Επαγγελματικό Λύκειο Συκεών

Σχέδια Δράσης 2022-2023



Πινακίδες και Εκπαιδευτικό Υλικό για Δραστηριότητες STEAM

Συκιές 2023



Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού,
Εκπαίδευση και Διά Βίου Μάθηση
Ειδική Υπηρεσία Διαχείρισης
Με την συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





Η εικόνα του εξωφύλλου αντλήθηκε από τη διεύθυνση:

<https://s3-us-west-2.amazonaws.com/robogarden-new/Articles/upload/blogs/lg-steam-for-kids-integrating-steam-activities-in-education.jpg>



Επιχειρησιακό Πρόγραμμα
Ανάπτυξη Ανθρώπινου Δυναμικού,
Εκπαίδευση και Διά Βίου Μάθηση
Ειδική Υπηρεσία Διαχείρισης
Με την συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





ΠΕΡΙΕΧΟΜΕΝΑ

1.	Πρόταση Εκπαιδευτικού σεναρίου για δραστηριότητες STEAM	5
1.1	Εξοικείωση με το δικτυακό τόπο tinkercad της AUTODESK.....	5
1.2	Βοήθεια για τη γλώσσα προγραμματισμού Wiring C και έτοιμα παραδείγματα κυκλωμάτων και κώδικα.....	7
1.3	Πρόταση εκπαιδευτικού σεναρίου.	8
1.4	Οδηγίες Εγκατάστασης του Ολοκληρωμένου Προγραμματιστικού Περιβάλλοντος IDE και σύνδεσης της πλακέτας Arduino στον ΗΥ.....	9
2.	Δραστηριότητες εξοικείωσης με το Arduino	11
2.1	Ψηφιακό ALARM με LED.....	11
2.2	Ψηφιακός Φακός με LED	14
2.3	Ανάγνωση αναλογικού σήματος με το Arduino.....	19
2.4	Αναλογική έξοδος με το Arduino – Οδήγηση κινητήρα DC	23
2.4.1	Η Τεχνική σήματος PWM	24
2.4.2	Το ηλεκτρονικό κύκλωμα στο TINKERCAD	27
2.4.3	Ο κώδικας στο TINKERCAD και στο ARDUINO IDE	28
2.5	Αναλογική έξοδος με το Arduino – Οδήγηση πιεζοηλεκτρικού βομβητή	30
2.5.1	Ο Ενεργός και ο παθητικός πιεζοηλεκτρικός βομβητής.....	30
2.5.2	Δραστηριότητα με ενεργό πιεζοηλεκτρικό βομβητή.....	32
2.5.3	Δραστηριότητα με ενεργό πιεζοηλεκτρικό βομβητή.....	34
2.6	ΕΛΕΓΧΟΣ ΚΙΝΗΤΗΡΑ micro SERVO	36
2.6.1	Έλεγχος micro SERVO MOTOR με την κλάση Servo της Arduino	37





2.6.2 Πληροφορίες για τον κινητήρα micro Servo της Arduino αλλά και άλλων εταιριών.....	37
2.6.3 Ανάλυση του κώδικα.....	39
2.7 Οδήγηση Μονοπολικού Βηματικού Κινητήρα (Unipolar Stepper Motor) με Arduino	41
2.7.1 Κατηγορίες των Βηματικών κινητήρων.....	42
2.7.2 Μονοπολικός βηματικός κινητήρας.....	43
2.7.3 Οδήγηση μονοπολικού βηματικού κινητήρα με την πλακέτα οδηγού ULN2003	43
2.7.4 Το κύκλωμα σύνδεσης του οδηγού ULN2003 με το ARDUINO UNO και το μοτέρ 28BYJ-48	46
2.7.5 Κώδικας Οδήγησης του βηματικού κινητήρα 28BYJ-48 με τον οδηγό ULN2003 μέσω ενσωματωμένων βιβλιοθηκών στο Arduino IDE.....	48
3. Δραστηριότητες STEAM	51
3.2.1 Οδηγίες ενσωμάτωσης βιβλιοθηκών.....	51
3.2.2 Σύνδεσμοι των δραστηριοτήτων STEAM στο TINKERCAD.....	54
4. Βιβλιογραφία	56
5. Ποιοι πήραν μέρος σε αυτή την εργασία.....	57





1. Πρόταση Εκπαιδευτικού σεναρίου για δραστηριότητες STEAM

1.1 Εξοικείωση με το δικτυακό τόπο tinkercad της AUTODESK

Θέτουμε σε ένα φυλλομετρητή τη διεύθυνση: <https://www.tinkercad.com/>

Αποκτούμε πρόσβαση με έναν από τους διαθέσιμους τρόπους πατώντας

Sign In

Την πρώτη φορά δημιουργούμε Λογαριασμό Σύνδεσης επιλέγοντας **Join now** ή **Sign In/ New to Autodesk ? Create Account** οπότε συμπληρώνουμε τα στοιχεία μας στα πτυσσόμενα μενού που εμφανίζονται και δημιουργούμε λογαριασμό ο οποίος θα περιλαμβάνει:

Το πραγματικό e-mail μας για επικοινωνία και επιβεβαίωση/έλεγχο του λογαριασμού.

Ένα password που θα ορίσουμε εμείς για να συνδεόμαστε στον ιστότοπο tinkercad (προσοχή : ορίζουμε διαφορετικό από εκείνον που αντιστοιχεί στο λογαριασμό ηλεκτρονικού ταχυδρομείου που έχουμε δηλώσει).

ή **Εναλλακτικά** όπως και σε πολλούς άλλους διαδικτυακούς τόπους, μπορούμε να χρησιμοποιήσουμε απευθείας **το λογαριασμό μας στη Google**, για να συνδεθούμε στο TINKERCAD

Με την ολοκλήρωση της δημιουργίας του λογαριασμού μεταφερόμαστε στο χώρο εργασίας που βλέπουμε και τη δυνατότητα δημιουργία 3D γραφικών και εκτυπώσεων σε 3D εκτυπωτή. Εμείς όμως δουλεύουμε με τις καρτέλες **Circuits**.

Στην πάνω δεξιά μεριά της εφαρμογής βλέπουμε το εικονίδιο (πρόσωπο) του λογαριασμού μας και επιλέγουμε το σύνδεσμο **Learn** και στη συνέχεια στην αριστερή πλευρά αντί για στο πτυσσόμενο μενού **3D** αριστερά, επιλέγουμε διαδοχικά **Circuits/Projects/Show All Arduino**.





Επιλέξτε ένα οποιοδήποτε project (προτείνουμε το Blink a LED with Digital Output) και στη συνέχεια επιλέξτε **Tinker This** ώστε να έρθετε στο περιβάλλον του εικονικού εργαστηρίου και της προσομοίωσης.

Εξοικειωθείτε με τα μενού εργαλείων και τον τρόπο του σχεδιασμού του κυκλώματος και της προσομοίωσης ακολουθώντας τις οδηγίες από τα παράθυρο **Βοήθειας** που εμφανίζεται στο αριστερό τμήμα της εφαρμογής.

Εξοικειωθείτε με τους ακροδέκτες του Arduino UNO (βλέποντας και την πραγματική πλακέτα) και τη δομή του προγράμματος όπως αυτό φαίνεται στα παράθυρα Βοήθειας.

Εξοικειωθείτε με το παράθυρο κώδικα Code συζητώντας τον κώδικα επιλέγοντας τη μορφή κώδικα **Text** ώστε να είμαστε σε περιβάλλον Wiring C και στο Serial Monitor.

Πατήστε το τετράγωνο κουμπί με τα πολύχρωμα γράμματα TINKERCAD στην αριστερή πλευρά της οθόνης και στην συνέχεια το κουμπί **Create New Circuit** για να σχεδιάσετε κυκλώματα. Κάθε φορά που δημιουργείτε ένα project είναι καλό να το μετονομάζετε κάνοντας διπλό κλικ πάνω στο τυχαίο όνομα που βάζει η εφαρμογή.

Η Εφαρμογή αυτόματα αποθηκεύει όλες τις τελευταίες αλλαγές που κάνετε και κάθε φορά που θα εισέρχεστε στην καρτέλα **Circuits** βλέπετε τα κυκλώματα που έχετε δημιουργήσει και τα οποία μπορείτε να τα μοιραστείτε και με άλλους χρήστες.

Για να προσομοιώσετε ένα προηγούμενο project, κάνετε κλικ πάνω του και στη συνέχεια επιλέγετε **tinker this**.





Μπορείτε επίσης να μετονομάσετε ή να διαγράψετε ένα προηγούμενο project κάνοντας κλικ στο **πορτοκαλί γρανάζι** στο παραθυράκι του κάθε project.

Κάθε φορά που θέλω να επιστρέψω στο παράθυρο με τα projects που έχω ήδη δημιουργήσει πατώ στο **τετράγωνο κουμπί με τα πολύχρωμα γράμματα TINKERCAD** στην αριστερή πλευρά της οθόνης .

1.2 Βοήθεια για τη γλώσσα προγραμματισμού Wiring C και έτοιμα παραδείγματα κυκλωμάτων και κώδικα

Ανοίξτε μια επιπλέον καρτέλα στο φυλλομετρητή σας και πληκτρολογήστε www.arduino.cc

Κατεβάστε εάν δεν το έχετε ήδη, το περιβάλλον προγραμματισμού της Arduino

Όλες τις εντολές, τις συναρτήσεις και τις βιβλιοθήκες μπορούμε να τις δούμε επιλέγοντας **Resources/Reference** μαζί με τα προτεινόμενα παραδείγματα, **από εδώ** (<https://www.arduino.cc/reference/en/>)

Επιλέξτε μέσα από τα έτοιμα παραδείγματα που υπάρχουν και στο περιβάλλον προγραμματισμού IDE - διαδρομή **Resources/Tutorials/Built In Examples** ή **Examples from Libraries** (<https://www.arduino.cc/en/Tutorial/HomePage>), για παράδειγμα: το Blink a Led και δείτε την πρόταση για τα απαραίτητα εξαρτήματα.

Δείτε τους ακροδέκτες της πλακέτας Arduino Uno και συγκρίνετε με την πραγματική πλακέτα που έχετε μπροστά σας

Δείτε το προτεινόμενο σχηματικό και τις επεξηγήσεις του κώδικα





Πατήστε το κουμπί **Open Code** ώστε να μπορέσετε να αντιγράψετε τον κώδικα που επιθυμείτε

Επικολλείτε ελεύθερα τον κώδικα είτε στο περιβάλλον προγραμματισμού του Tinkercad για προσομοίωση είτε στο προγραμματιστικό περιβάλλον IDE της Arduino ώστε να το κατεβάσετε στην πραγματική πλακέτα και στο κύκλωμα που φτιάξατε στο εργαστήριο

1.3 Πρόταση εκπαιδευτικού σεναρίου.

Οι μαθητές:

Δημιουργούν προσωπικούς λογαριασμούς στο tinkercad (για ευκολία μπορούν να χρησιμοποιήσουν τους λογαριασμούς Google που έχουν σχεδόν όλοι ..)

Ενημερώνονται από τον ιστότοπο της Arduino για τα απαραίτητα στοιχεία συγγραφής κώδικα στη Wiring C είτε αντιγράφουν ολόκληρο τον κώδικα

Φορτώνουν σε επιπλέον καρτέλα του φυλλομετρητή το tinkercad (www.tinkercad.com), σχεδιάζουν το κύκλωμα και μεταφέρουν τον προηγούμενο κώδικα στο παράθυρο **Code (Text)**.

Κάνουν τις απαραίτητες τροποποιήσεις ενδεχομένως στους ακροδέκτες που θα χρησιμοποιήσουν και κάνουν έλεγχο σφαλμάτων πατώντας το κουμπί **Debugging** (εικονίδιο με τη **Μέλισσα**)

Πατούν **Start Simulation** και ανοίγουν και το παράθυρο **Serial Monitor** εάν χρειάζεται.

Φορτώνουν ταυτόχρονα στον Η/Υ το περιβάλλον προγραμματισμού της Arduino





Αντιγράφουν τον κώδικα από το tinkercad στο περιβάλλον προγραμματισμού της Arduino

Συνδέουν την πλακέτα Arduino Uno, επιλέγουν από το μενού **Εργαλεία/Πλακέτα** τη σωστή πλακέτα και ελέγχουν από τη διαδρομή **Εργαλεία/Θύρα** για τη αναγνώριση της πλακέτας Arduino Uno στη σωστή θύρα COM του Η/Υ. Για περισσότερες λεπτομέρειες ανατρέξτε στην ιστοσελίδα του 1^{ου} ΕΠΑΛ Συκεών (<http://1epal-sykeon.thess.sch.gr>) στο σύνδεσμο **Εκπαιδευτικό Υλικό/Arduino** και στο φύλλο έργου: *Οδηγίες εγκατάστασης και χρήσης του Arduino Software (IDE) ...*

Εκτελούν **Debugging** και **μεταφόρτωση** του κώδικα μηχανής στην πλακέτα Arduino Uno και ελέγχουν τη λειτουργία του πραγματικού κυκλώματος.

1.4 Οδηγίες Εγκατάστασης του Ολοκληρωμένου Προγραμματιστικού Περιβάλλοντος IDE και σύνδεσης της πλακέτας Arduino στον ΗΥ

Βήμα 1^ο: (μόνο την πρώτη φορά της εγκατάστασης του λογισμικού)

Κατεβάζουμε από τον ιστότοπο της εταιρείας:

(<https://www.arduino.cc/en/Main/Software>)

και εγκαθιστούμε στον Η/Υ μας, μια έκδοση του **Arduino Software IDE**. Για τις ανάγκες των συγκεκριμένων δραστηριοτήτων μπορείτε να επιλέξετε μια έκδοση της σειράς **1** π.χ. 1.8.19 σε συνδυασμό και με την έκδοση του λειτουργικού συστήματος που διαθέτει το εργαστήριο (32 ή 64 bits). Το περιβάλλον προγραμματισμού εγκαθίσταται κανονικά στην ομάδα **Program Files(x86)** και μαζί με αυτό εγκαθίστανται και οι οδηγοί για τις διάφορες πλακέτες Arduino. Έτσι όταν συνδέσουμε την πλακέτα Arduino UNO σε μια θύρα USB του Η/Υ μας, η αναγνώριση της πλακέτας Arduino UNO γίνεται αυτόματα. Στην





περίπτωση που δεν γίνει σωστή αναγνώριση (π.χ. Windows XP) τότε στον οδηγό εγκατάστασης που εμφανίζεται μπορούμε να επιλέξουμε χειροκίνητη εγκατάσταση (**όχι Windows Update**) και στη συνέχεια **Αυτόματη Εγκατάσταση** (από οδηγούς που ήδη έχουν εγκατασταθεί στον Η/Υ μας).

Βήμα 2^ο: (όταν συνδέσω την πλακέτα Arduino σε μια θύρα USB του Η/Υ και ολοκληρωθεί η αναγνώριση)

Στη **Διαχείριση Συσκευών** (Η/Υ μου / δεξί κλικ /ιδιότητες/ Διαχείριση Συσκευών/Θύρες COM&LPT) βλέπω το όνομα των καταχωρητών COM της σειριακής θύρας (το βύσμα USB του Η/Υ στο οποίο συνδέσαμε την πλακέτα Arduino UNO) πχ. COM4 ή COM7 ή COM11.

Βήμα 3^ο: (όταν φορτώνω τη γλώσσα προγραμματισμού)

Φορτώνουμε το περιβάλλον (**IDE**) της Wiring C με διπλό κλικ στη συντόμευση **Arduino** που έχει εγκατασταθεί από το Βήμα 1 στην επιφάνεια εργασίας και

α) στο Μενού **Εργαλεία /Πλακέτα** δηλώνω την πλακέτα **Arduino UNO** ενώ

β) στο Μενού **Εργαλεία /Σειριακή θύρα** κάνω κλικ στη θύρα **COMn** που βρήκα στο Βήμα (2) ακόμα και αν αυτή φαίνεται επιλεγμένη, ώστε να γίνει σωστά την πρώτη φορά η μεταφόρτωση του κώδικα στην πλακέτα Arduino UNO. Επίσης μόνο τότε θα λειτουργεί και η ενσωματωμένη **Σειριακή Οθόνη** (επιλέγοντας **Εργαλεία /Σειριακή Οθόνη**) για την ανταλλαγή μηνυμάτων μεταξύ του Η/Υ μας και της πλακέτας Arduino UNO.

Βήμα 4^ο: (όταν γράφω τον κώδικα και μεταφέρω το αρχείο στον μικροελεγκτή)

α) Γράφουμε τον κώδικα με τις εντολές της Wiring C.

β) Διορθώνουμε τα συντακτικά λάθη κάνοντας **Επαλήθευση/Μεταγλώττιση** (μενού **Σχέδιο**)





γ) Μεταφέρω τον κώδικα στην πλακέτα Arduino UNO κάνοντας κλικ στο δεξί βελάκι **Φόρτωση** και παρατηρώ τη λειτουργία του προγράμματος που κατέβασα στον μικροελεγκτή.

2. Δραστηριότητες εξοικείωσης με το Arduino

Σε αυτή την ενότητα, παρουσιάζουμε ορισμένες δραστηριότητες για την εξοικείωση με το Arduino και το περιβάλλον προγραμματισμού του για όσους έρχονται σε επαφή για πρώτη φορά με αυτή την ομάδα των ηλεκτρονικών πλακετών

2.1 Ψηφιακό ALARM με LED

Στόχοι: Η εξοικείωση με

α) τη δήλωση σταθερών και μεταβλητών με τις εντολές `const` και `int`

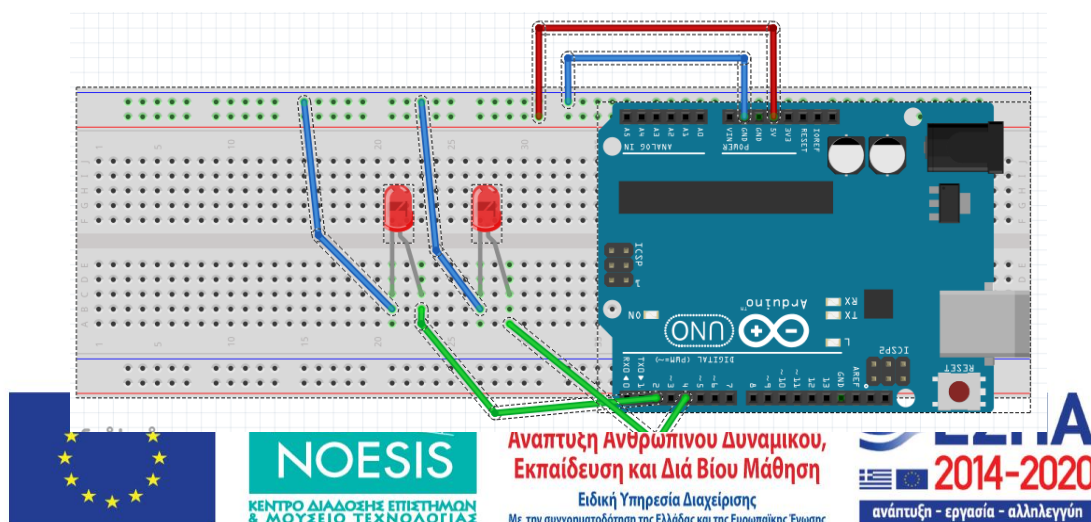
β) τον ορισμό ακροδεκτών ως ψηφιακών εξόδων με την εντολή **`pinmode(pin,mode)`**, τη χρονοκαθυστέρηση με την εντολή **`delay(ms)`**, και την οδήγηση των εξόδων με την εντολή **`digitalWrite(pin, mode)`**

Τα Υλικά που θα χρειαστούμε:

- 1 Πλακέτα Arduino Uno και ένα breadboard
- 2 2x Led

Το κύκλωμα που θα σχεδιάσουμε :

α) Σχεδιάζω στο tinkercad ή συνθέτω απευθείας σε ένα breadboard το κύκλωμα που φαίνεται το παρακάτω σχήμα





Το σχέδιο του κυκλώματος έγινε με beta-έκδοση του fritzing (www.fritzing.org)

β) Τα δύο LEDs τα συνδέω στους ψηφιακούς ακροδέκτες 2 και 4 της πλακέτας του Arduino και τέλος,

γ) μεταφέρω την τροφοδοσία 5V και GND από τη πλακέτα του Arduino UNO στο breadboard για την τροφοδοσία του κυκλώματος.

Επιπλέον στην περίπτωση που κατασκευάσω το πραγματικό κύκλωμα στο breadboard, συνδέω την πλακέτα του Arduino UNO με μια θύρα USB του ΗΥ χρησιμοποιώντας ένα καλώδιο τύπου **USB Cable A Male to B Male** (ίδιο με εκείνο που χρησιμοποιείται για τη σύνδεση εκτυπωτών σε ΗΥ). Με τον τρόπο αυτό

α) θα γίνει η μεταφορά του κώδικα που θα γράψουμε στο προγραμματιστικό περιβάλλον (IDE) από τον ΗΥ στο τσιπάκι του μικροελεγκτή ATmega328P της πλακέτας Arduino UNO ενώ,

β) εξασφαλίζω μια στοιχειώδη ηλεκτρική τροφοδοσία για χαμηλά ηλεκτρικά φορτία στο το πραγματικό κύκλωμα χωρίς να απαιτείται προς το παρόν κάποια μπαταρία ή σύνδεση με τροφοδοτικό συνεχούς τάσης.

Ο κώδικας:

Στη συνέχεια δίνεται ο κώδικας με τα απαραίτητα σχόλια. Οι γραμμές κώδικα με τις δύο πλάγιες γραμμές // ή ανάμεσα στα σύμβολα /*.....*/ είναι για τα επεξηγηματικά σχόλια του προγραμματιστή. Μπορούμε να δούμε ότι η δομή ενός τυπικού κώδικα περιλαμβάνει τρεις ενότητες :

/* Α. Τις δηλώσεις σταθερών και μεταβλητών */

```
//θα χρησιμοποιήσουμε τους 2 : led1 4 :led2  
int led1 = 2;  
int led2 = 4;
```





/* Β. Τις αρχικοποιήσεις που αφορούν τους ακροδέκτες και τις ρυθμίσεις επικοινωνίας με τον ΗΥ ή άλλες συσκευές */

```
// η ρουτίνα setup() εκτελείται μόνο μια φορά και
// περιλαμβάνει τον καθορισμό των ψηφιακών εξόδων
void setup() {
  pinMode(led1, OUTPUT); // ορίζουμε το ψηφιακό
                          // ακροδέκτη 2 ως ψηφιακή έξοδο
  pinMode(led2, OUTPUT); // ορίζουμε το ψηφιακό
                          // ακροδέκτη 4 ως ψηφιακή έξοδο
}
```

/* Γ. Τις εντολές που θα επαναλαμβάνονται συνεχώς όσο υπάρχει τροφοδοσία στο κύκλωμα */

```
// η ρουτίνα αυτή επαναλαμβάνεται συνεχώς
void loop() {
  digitalWrite(led1, HIGH); // βάζουμε στον ακροδέκτη 2
                            // σε κατάσταση HIGH (5 Volts)
                            // και το LED ανάβει
  digitalWrite(led2, LOW); // βάζουμε στον ακροδέκτη 4
                            // σε κατάσταση LOW(0 Volts)
                            // και το LED σβήνει
  delay(1000); // χρονοκαθυστέρηση 1000 ms
  digitalWrite(led1, LOW); // βάζουμε στον ακροδέκτη 2
                            // σε κατάσταση LOW(0 Volts)
                            // και το LED σβήνει
  digitalWrite(led2, HIGH); // βάζουμε στον ακροδέκτη 4
                             // σε κατάσταση HIGH (5 Volts)
                             // και το LED ανάβει
}
```





```
delay(1000); // χρονοκαθυστέρηση 1000 ms
```

```
}
```

Μεταφορά του κώδικα στην πλακέτα :

α) Διορθώνουμε τα συντακτικά λάθη επιλέγοντας στο προγραμματιστικό περιβάλλον

Επαλήθευση/Μεταγλώττιση (από το μενού **Σχέδιο**) ή κάνω κλικ στο εικονίδιο **V**

β) μεταφέρω τον κώδικα στην πλακέτα Arduino κάνοντας κλικ στο δεξί βελάκι **Φόρτωση**

Όταν ολοκληρωθεί η μεταφορά του κώδικα στο τσιπ του μικροελεγκτή, παρατηρώ τη λειτουργία του πραγματικού κυκλώματος. Εναλλακτικά στην περίπτωση που εκτελώ προσομοίωση στο tinkercad, η λειτουργία φαίνεται στην οθόνη του ΗΥ κάνοντας κλικ στο κουμπί **Simulation**.

2.2 Ψηφιακός Φακός με LED

Στόχοι: Η εξοικείωση με

α) τον ορισμό ακροδεκτών ως ψηφιακών εισόδων και ψηφιακών εξόδων με την εντολή

pinmode(pin,mode) όπου **mode=INPUT/OUTPUT**

β) τις εντολές επιλογής **if Elseif....**, για επιλογή υπό συνθήκη των εντολών που

εκτελούνται κάθε φορά και με την εντολή **digitalRead(pin, mode)** για την ανάγνωση ψηφιακών εισόδων

γ) τη σύνδεση διακόπτη-μπουτόν μέσω αντίστασης pull-down

Τα Υλικά που θα χρειαστούμε:

- 1 Πλακέτα Arduino Uno και ένα breadboard
- 2 1x Led
- 3 Αντιστάσεις : 1x220 Ω και 1x10KΩ
- 4 1x Διακόπτης-push button

Λειτουργία του διακόπτη-μπουτόν :





Ο διακόπτης (button) έχει τέσσερις ακροδέκτες που ανά δύο είναι βραχυκυκλωμένοι. Επομένως για να λειτουργήσει ως διακόπτης σε σειρά πρέπει το κύκλωμα να συνδεθεί μεταξύ ακροδεκτών που δεν είναι μεταξύ τους βραχυκυκλωμένοι. Για να εντοπίσω τη λειτουργία των ακροδεκτών του button θα χρησιμοποιήσουμε πολύμετρο. Στο παρακάτω σχήμα βλέπουμε τους δύο τρόπους με τους οποίους μπορούμε να συνδέσουμε το μπουτόν σε ένα κύκλωμα ώστε όταν πατάμε το μπουτόν να ανάβει ένα LED. Για την προστασία του LED χρησιμοποιούμε σε σειρά μια αντίσταση 220Ω.

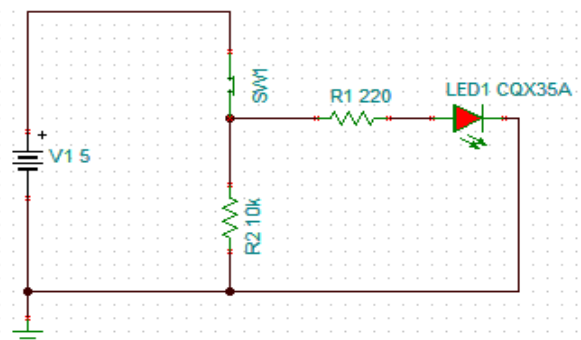
button μέσω pull down αντίστασης

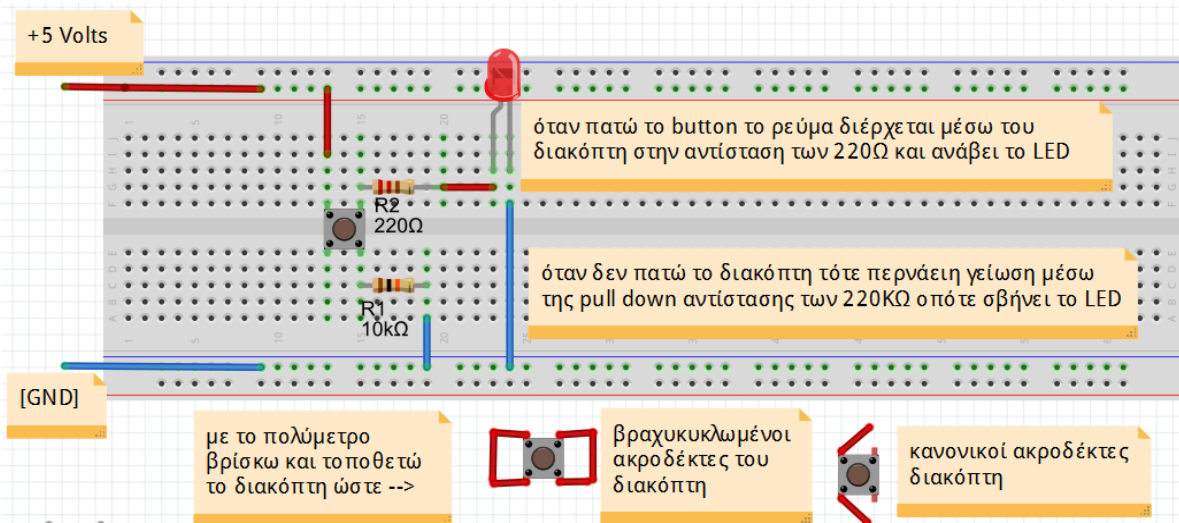
Στη σύνδεση αυτή, ο ένας ακροδέκτης του button (SW1) συνδέεται στην τροφοδοσία των 5 Volts ενώ ο άλλος ακροδέκτης συνδέεται σε μια αντίσταση 10 KΩ η οποία από την άλλη μεριά καταλήγει στη γείωση. Η αντίσταση τότε χαρακτηρίζεται ως pull down.

Στο σημείο που συνδέεται το button (SW1) με την pull down αντίσταση, συνδέουμε την αντίσταση προστασίας 220 Ω με τον θετικό ακροδέκτη (άνοδος -μεγαλύτερο μήκος) του LED ενώ ο αρνητικός ακροδέκτης του LED (κάθοδος-μικρότερος μήκος) συνδέεται στη γείωση.

Με αυτή τη σύνδεση όταν πατήσουμε το button, το ρεύμα που διέρχεται επιλέγει τον κλάδο με τη μικρότερη αντίσταση οπότε ανάβει το LED. Έτσι όταν πατάμε το button μεταφέρουμε σε ένα κύκλωμα το λογικό '1'.

button μέσω pull down αντίστασης



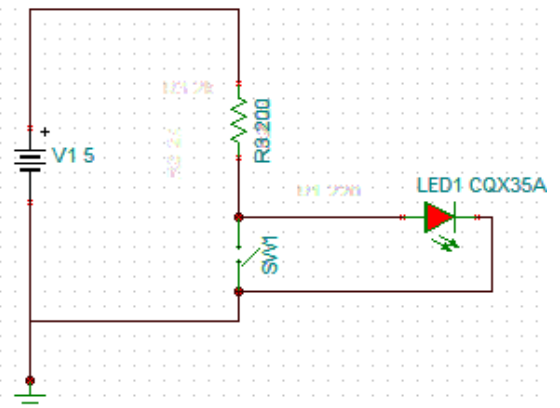


Το σχέδιο του κυκλώματος έγινε με beta-έκδοση του fritzing (www.fritzing.org)

button μέσω pull up αντίστασης

Στη σύνδεση αυτή, ο ένας ακροδέκτης του button (SW1) συνδέεται στον κλάδο του LED και ο άλλος στη γείωση. Η αντίσταση συνδέεται τώρα από τη μεριά της τροφοδοσίας των 5 Volts (10 KΩ ή και μικρότερη) και τότε χαρακτηρίζεται ως pull up. Τώρα το, το LED είναι μόνιμα αναμμένο και όταν πατήσουμε το button, μεταφέρεται η γείωση (0 Volts) στο LED και σβήνει. Έτσι όταν πατάμε το button μεταφέρουμε σε ένα κύκλωμα το λογικό '0'.

button μέσω pull up αντίστασης

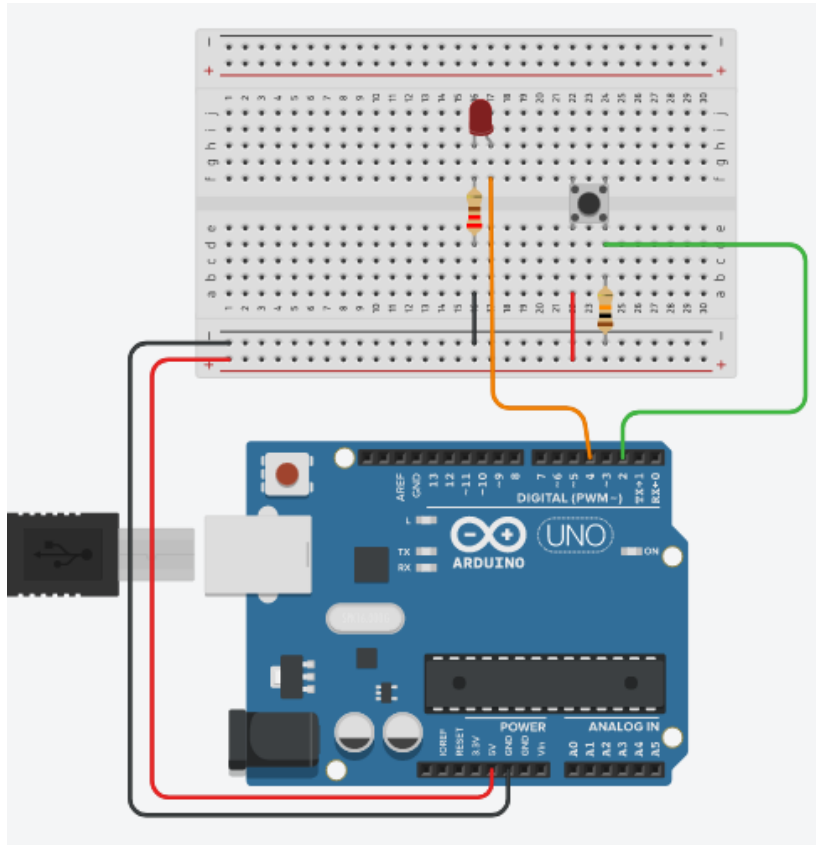


Το κύκλωμα που θα σχεδιάσουμε :





α) Σχεδιάζω στο tinkercad ή συνθέτω απευθείας σε ένα breadboard το κύκλωμα που φαίνεται στην παρακάτω εικόνα (διακόπτης με αντίσταση στη γείωση):



Το LED συνδέεται μέσω της αντίστασης **220 Ω** στο ψηφιακό ακροδέκτη του Arduino **D4** (οπότε αυτός λειτουργεί σαν ψηφιακή έξοδος **OUTPUT**) ενώ ο διακόπτης στο ψηφιακό ακροδέκτη του Arduino **D2** (οπότε αυτός λειτουργεί σαν ψηφιακή είσοδος **INPUT**).

Ο κώδικας:

Στη συνέχεια δίνεται ο κώδικας με τα απαραίτητα σχόλια. Οι γραμμές κώδικα με τις δύο πλάγιες γραμμές `//` ή ανάμεσα στα σύμβολα `/*.....*/` είναι για τα επεξηγηματικά σχόλια του προγραμματιστή.





// δήλωση μεταβλητών:

```
int button = 0; // μεταβλητή που διαβάζει το button
                // θέτοντας αρχική τιμή 0 (δεν πατήθηκε)
```

```
/* αρχικοποίηση παραμέτρων λειτουργίας των ακροδεκτών του
Arduino UNO */
```

```
void setup() {
    pinMode(4, OUTPUT); //ορίζουμε τον ακροδέκτη 4
                        // ως έξοδο
    pinMode(2, INPUT); //ορίζουμε τον ακροδέκτη 2
                        // ως είσοδο
```

```
}
```

```
/* τμήμα εντολών που επαναλαμβάνονται όσο υπάρχει τροφοδοσία
στο Arduino (συνήθως μέσω του καλωδίου USB από τον ΗΥ) */
```

```
void loop() {
    button = digitalRead(2); // διαβάζουμε 0 ή 1 από τον
    /* ακροδέκτη 2 για να δούμε αν πατήθηκε ο διακόπτης
    και θέτουμε το αποτέλεσμα στην τιμή της μεταβλητής
    button */
    if (button == HIGH) // αν πατήθηκε το button
    {
        digitalWrite(4, HIGH); // άναψε το LED στον
                                // ακροδέκτη 13
    }
    else // αλλιώς
    {
        digitalWrite(4, LOW); // σβήσε το LED στον
                                // ακροδέκτη 13
    }
}
```





Μεταφορά του κώδικα στην πλακέτα :

α) Διορθώνουμε τα συντακτικά λάθη επιλέγοντας στο προγραμματιστικό περιβάλλον **Επαλήθευση/Μεταγλώττιση** (από το μενού **Σχέδιο**) ή κάνω κλικ στο εικονίδιο **V**

β) μεταφέρω τον κώδικα στην πλακέτα Arduino κάνοντας κλικ στο δεξί βελάκι **Φόρτωση**

Όταν ολοκληρωθεί η μεταφορά του κώδικα στο τσιπ του μικροελεγκτή , παρατηρώ τη λειτουργία του πραγματικού κυκλώματος. Εναλλακτικά στην περίπτωση που εκτελώ προσομοίωση στο tinkercad, η λειτουργία φαίνεται στην οθόνη του ΗΥ κάνοντας κλικ στο κουμπί **Simulation**.

2.3 Ανάγνωση αναλογικού σήματος με το Arduino

Στόχοι: α) κατανόηση λειτουργίας A/D μετατροπέα 10 bit

β) να γίνει προσομοίωση ενός εξόδου ενός αναλογικού αισθητηρίου

Τα Υλικά που θα χρειαστούμε:

1. Πλακέτα Arduino Uno
2. 1x ποτενσιόμετρο π.χ. 10KΩ, 22KΩ

Το κύκλωμα που θα σχεδιάσουμε :

Συνδέουμε το ποτενσιόμετρο ενός αναλογικούς ακροδέκτες ενός πλακέτας Arduino UNO ενός στο παρακάτω σχήμα. Οι δύο εξωτερικοί ακροδέκτες του ποτενσιόμετρου ενός εξόδους **5V** και **0 Volts** ενώ ο μεσαίος ακροδέκτης στην αναλογική είσοδο **A0** (ANALOG INPUT).

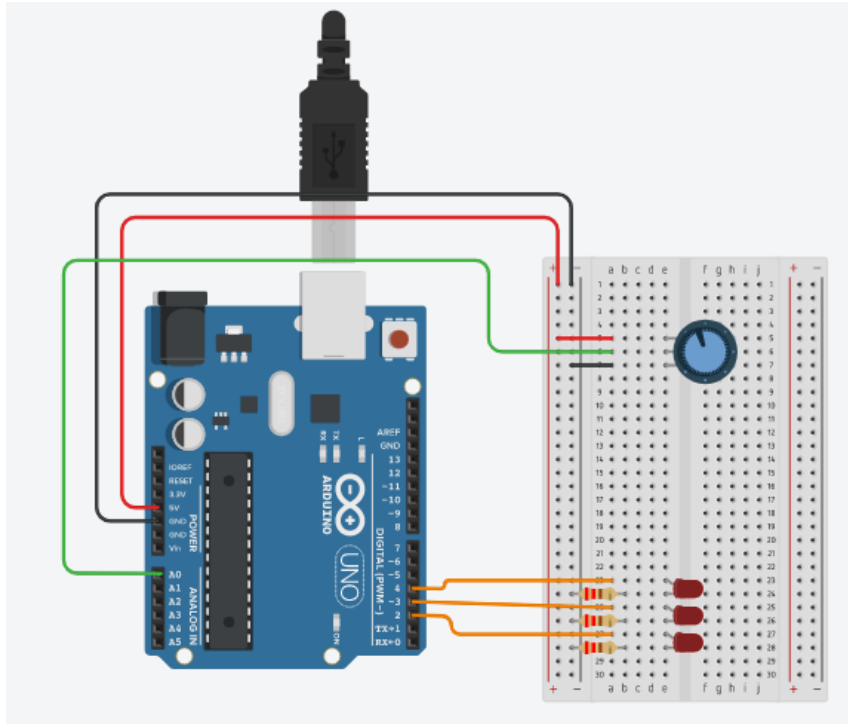
Τι πρέπει να γνωρίζουμε για την ανάγνωση αναλογικών σημάτων

μικροελεγκτής ενός διαθέτει ένα μετατροπέα A/D των 10 bits (1023 επίπεδα κβάντισης) ενώ με τη συνάρτηση **analogReference(DEFAULT)** ορίζουμε ότι η τάση αναφοράς του μετατροπέα είναι η προκαθορισμένη 5V ή 3.3 Volt ανάλογα την πλακέτα που θα





χρησιμοποιήσουμε. Η τάση αναφοράς πρέπει να είναι πάντα μεγαλύτερη από τη μέγιστη τάση που ενός δίνει το αναλογικό αισθητήριο.



Η αναλογική τάση εισόδου δίνεται από τον παρακάτω τύπο:

$$V_{in} = \frac{\Delta V}{2^{10} - 1} \times digitalWord$$

$$digitalWord = b_9 \times 2^9 + b_8 \times 2^8 + b_7 \times 2^7 + b_6 \times 2^6 + b_5 \times 2^5 + b_4 \times 2^4 + b_3 \times 2^3 + b_2 \times 2^2 + b_1 \times 2^1 + b_0 \times 2^0$$

Στο πρόγραμμα που γράφουμε παρακάτω, η εντολή

```
int sensorVal=analogRead(sensorPin);
```

Μεταφέρει τη **δεκαδική τιμή** ενός ψηφιακής λέξης (*digitalWord*) στη μεταβλητή *sensorVal*. Η Δεκαδική τιμή προκύπτει από την μετατροπή που κάνει ο μετατροπέας A/D στο αναλογικό σήμα που έρχεται από το ποτενσιόμετρο στον αναλογικό ακροδέκτη A0 ενός πλακέτας ArduinoUNO.

Στη συνέχεια εφαρμόζουμε ενός παραπάνω τύπους για να υπολογίσουμε την αναλογική τάση που μετράμε θέτοντας :





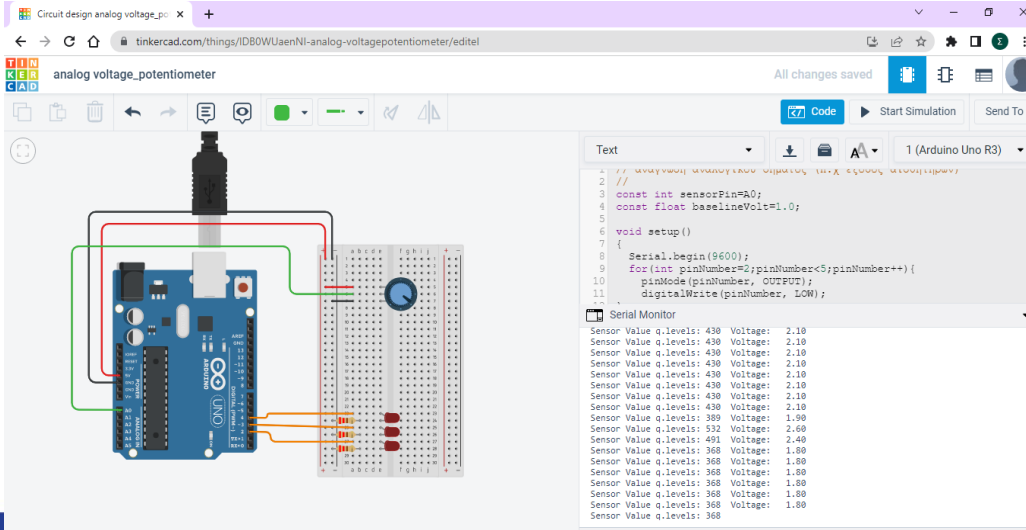
$$\Delta V = 5 \text{volts}$$

$$2^{10} = 1024$$

Με τη συνάρτηση `Serial.begin(9600)` ρυθμίζουμε την ταχύτητα σειριακής επικοινωνίας μεταξύ ενός πλακέτας Arduino UNO και του υπολογιστή ενός να είναι 9600 bps.

Με την εντολή `Serial.print(...)` απεικονίζουμε στην ενσωματωμένη **Σειριακή Οθόνη** ενός μεταβλητές των μετρήσεων ενός. Η **Σειριακή Οθόνη** ενεργοποιείται στο περιβάλλον ενός Wiring C από το μενού **Εργαλεία / Σειριακή Οθόνη**. Προσοχή θα πρέπει να έχει δηλωθεί σωστά η θύρα COM που συνδέσαμε τη USB πλακέτα Arduino UNO σε προηγούμενες παραγράφους. Αντίστοιχα, στο περιβάλλον tinkercad, ενεργοποιείται στο στάδιο ενός προσομοίωσης (**Simulation**), κάνοντας κλικ στο πεδίο Serial Monitor που βρίσκεται κάτω από το παράθυρο του κώδικα (**Code**).

Στη συνέχεια γράφουμε τον παρακάτω κώδικα με τον οποίο διαβάζουμε την τιμή τάσης από το μεταβλητό ακροδέκτη ενός ποτενσιόμετρου και χρησιμοποιούμε την εντολή επιλογής **if...else if..** για να έχουμε οπτική ένδειξη των τιμών ενός αναλογικής τάσης μέσω τριών LEDs. Ταυτόχρονα μπορούμε να βλέπουμε και τις τιμές στο παράθυρο της σειριακής οθόνης. Έτσι βλέπουμε πως, χρησιμοποιώντας τη δομή επανάληψης **for..** ορίζουμε με μόνο δύο εντολές ταυτόχρονα την κατάσταση λειτουργίας πολλών ψηφιακών ακροδεκτών:



The screenshot shows the Tinkercad environment with a project titled 'analog voltage_potentiometer'. The circuit includes an Arduino Uno, a potentiometer, and three LEDs. The Serial Monitor window displays the following output:

Sensor Value	q.levels	Voltage
430	430	2.10
430	430	2.10
430	430	2.10
430	430	2.10
430	430	2.10
430	430	2.10
430	430	2.10
430	430	2.10
389	389	1.90
532	532	2.60
491	491	2.40
368	368	1.80
368	368	1.80
368	368	1.80
368	368	1.80



Ο κώδικας

```
// ανάγνωση αναλογικού σήματος (π.χ έξοδος αισθητήρων)
//
const int sensorPin=A0; // δήλωσης σταθεράς ακροδέκτη
                        // αναλογικής εισόδου A0
const float baselineVolt=1.0;// δήλωση ενός ορίου τάσης

void setup()
{
  Serial.begin(9600);
  for(int pinNumber=2;pinNumber<5;pinNumber++){
    pinMode(pinNumber, OUTPUT); // δήλωση ακροδεκτών
                                // ψηφιακών εξόδων
    digitalWrite(pinNumber, LOW); // αρχικές τιμές LOW
                                // ώστε LEDs σβηστά
  }
}

void loop()
{
  int sensorVal=analogRead(sensorPin); //ανάγνωση A/D
  Serial.print("\n Sensor Value q.levels:\t");
  Serial.print(sensorVal);

  float voltage=(sensorVal/1024.0)*5.0; //μετατροπή σε
                                // τιμές αναλογικής τάσης
```





```
Serial.print (« Voltage:\t»);  
Serial.print (voltage);  
  
/* άναμμα LEDS για περιοχές αναλογικών τάσεων που  
   σχετίζονται με το όριο */  
if (voltage<baselineVolt){  
    digitalWrite(2, HIGH);  
    digitalWrite(3, LOW);  
    digitalWrite(4, LOW);  
}  
else if (voltage>=baselineVolt &&  
        voltage<baselineVolt+2) {  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, LOW);  
}  
else if (voltage>=baselineVolt+2) {  
    digitalWrite(2, HIGH);  
    digitalWrite(3, HIGH);  
    digitalWrite(4, HIGH);  
}  
delay(1);  
  
}
```

2.4 Αναλογική έξοδος με το Arduino – Οδήγηση κινητήρα DC

- Στόχοι:** α) η γνωριμία με την Τεχνική PWM και η εφαρμογή της για οδήγηση των αναλογικών εξόδων του Arduino,
β) η εξοικείωση με τις συναρτήσεις **analogWrite()** και **analogRead()** .



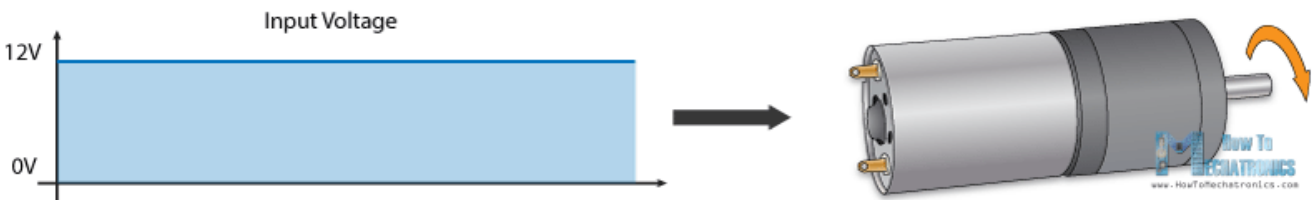


Τα Υλικά που θα χρειαστούμε:

- 1 Πλακέτα Arduino Uno και ένα breadboard
- 2 1x ποτενσιόμετρο
- 3 1x Κινητήρα Συνεχούς τάσης 5 Volts

2.4.1 Η Τεχνική σήματος PWM

Μπορούμε να ελέγξουμε την ταχύτητα ενός κινητήρα συνεχούς ρεύματος απλά μεταβάλλοντας την τάση εισόδου στον κινητήρα.



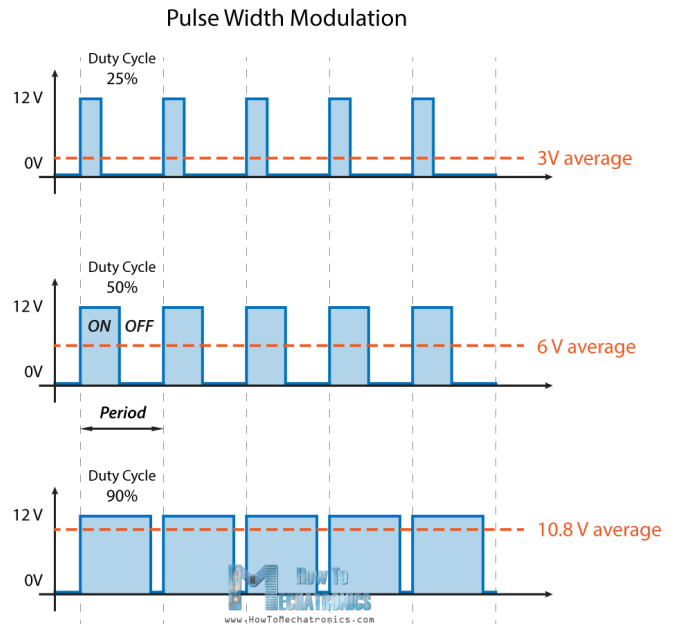
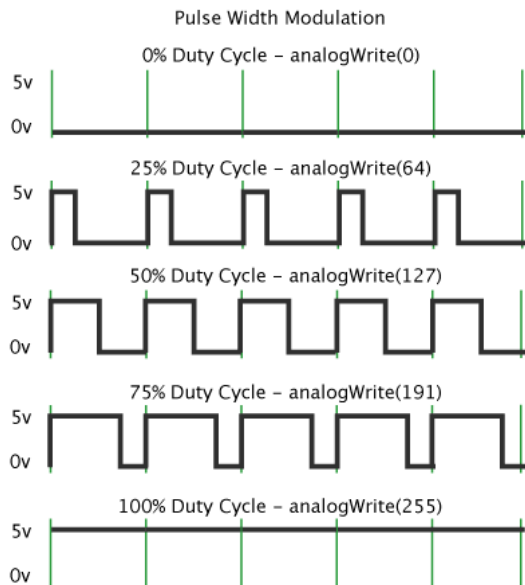
<https://howtomechatronics.com/tutorials/arduino/arduino-dc-motor-control-tutorial-l298n-pwm-h-bridge/>

Η πλακέτα όμως ARDUINO UNO R3 δεν έχει ακροδέκτες αναλογικών εξόδων τάσης. Χρησιμοποιούμε όμως την πλέον διαδεδομένη τεχνική σήματος PWM έτσι ώστε από τους ακροδέκτες ψηφιακών εξόδων του Arduino να δημιουργούμε ένα σήμα αναλογικής τάσης μεταβαλλόμενης τιμής με το οποίο τροφοδοτούμε τον DC κινητήρα ελέγχοντας έτσι και την ταχύτητα περιστροφής του.





Με την τεχνική **Pulse Width Modulation (PWM)** δημιουργούμε αναλογικά σήματα σε ακροδέκτες που λειτουργούν ως ψηφιακές έξοδοι. Στην πραγματικότητα μέσω της εντολής **analogWrite()** παράγουμε τετραγωνικούς παλμούς με διαφορετικούς κύκλους εργασίας (duty cycles %).



<https://www.arduino.cc/en/Tutorial/PWM>

Εφόσον η συχνότητα της τετραγωνικής κυματομορφής είναι αρκετά μεγάλη, μια συσκευή που συνδέεται σε έναν ακροδέκτη PWM αντιλαμβάνεται την κυματομορφή σαν μια DC τάση με τιμή ίσο με το μέσο όρο της τάσης της κυματομορφής. Οι ακροδέκτες εξόδου PWM σε μια πλακέτα ARDUINO UNO είναι εκείνοι οι ακροδέκτες ψηφιακής εξόδου που έχουν επιπλέον το σύμβολο (\sim). Μέσω της τεχνικής PWM οι ψηφιακοί ακροδέκτες εξόδου εμφανίζονται να λειτουργούν ως ακροδέκτες εξόδου αναλογικής τάσης μεταβάλλοντας με τον τρόπο αυτό την ταχύτητα περιστροφής του κινητήρα.

Η δομή της εντολής είναι:

`analogWrite(ακροδέκτης, περιεχόμενο 0-255);`





Όταν ο duty cycle=0% τότε το παραγόμενο σήμα έχει τιμή 0 Volts.

Περιεχόμενο εντολής: 0

Όταν ο duty cycle=50% τότε το παραγόμενο σήμα έχει τιμή 2.5 Volts.

Περιεχόμενο εντολής: 127

Όταν ο duty cycle=100% τότε το παραγόμενο σήμα έχει τιμή 5Volts.

Περιεχόμενο εντολής: 255

Όταν ο duty cycle=0-100% τότε το παραγόμενο σήμα έχει τιμή 0-5Volts.

Περιεχόμενο : 0-255

Η συχνότητα του τετραγωνικού παλμού για την πλακέτα μας είναι περίπου 500Hz δηλαδή έχει περίοδο 2ms οπότε για να προλαβαίνουμε να βλέπουμε το αποτέλεσμα της εντολής `analogWrite()` θα πρέπει να έχουμε ένα `delay(ms)>2ms`.

ΠΡΟΣΞΕΤΕ:

A) οι αναλογικές είσοδοι (A0-A5) που διαβάζονται με την εντολή `analogRead(...)` προφανώς δεν ορίζονται ως είσοδοι με εντολές `pinmode`. Το ίδιο ισχύει και για τους ψηφιακούς ακροδέκτες που έχουν την ένδειξη με το σύμβολο \sim όταν χρησιμοποιούνται ως έξοδοι PWM με την εντολή `analogWrite(...)` (δεν χρειάζεται να τους ορίσουμε ως εξόδους με εντολές `pinmode`).

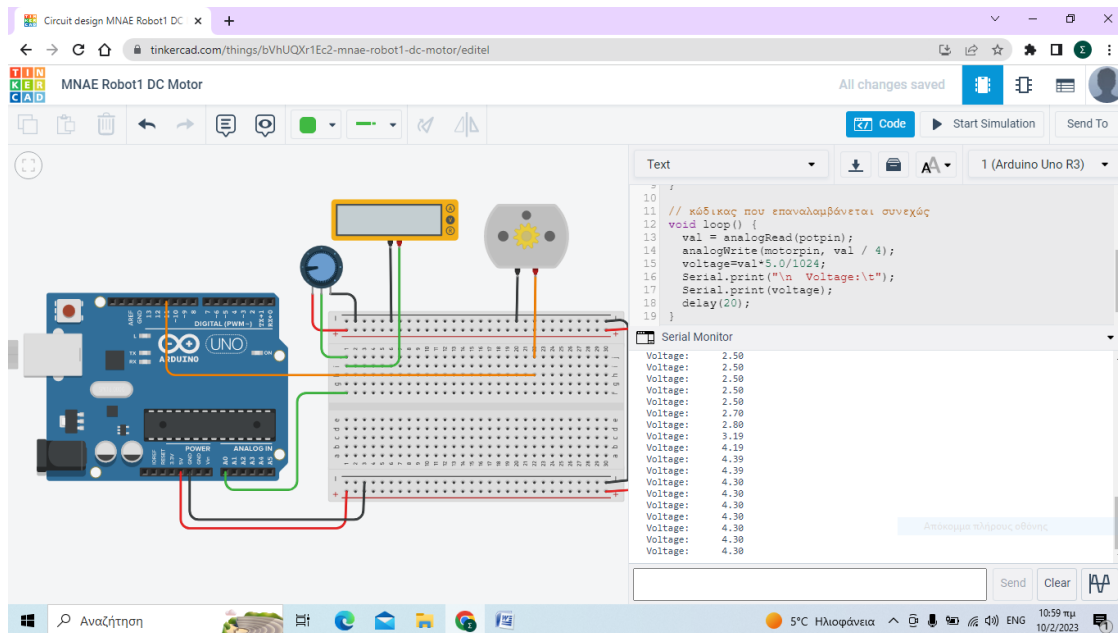
B) οι ακροδέκτες γείωσης της πλακέτας ARDUINO και της τροφοδοσίας του κινητήρα πρέπει να είναι συνδεδεμένοι μεταξύ τους. Αυτό θέλει ιδιαίτερη προσοχή στην περίπτωση που θα χρησιμοποιήσουμε διαφορετική τροφοδοσία για τον κινητήρα από εκείνη που μας παρέχει η πλακέτα ARDUINO UNO. Στη συγκεκριμένη δραστηριότητα η τροφοδοσία του κινητήρα γίνεται με το σήμα PWM και η γείωση έρχονται απευθείας από την πλακέτα του ARDUINO UNO.





2.4.2 Το ηλεκτρονικό κύκλωμα στο TINKERCAD

Το κύκλωμα που σχεδιάσαμε στο TINKERCAD φαίνεται στη συνέχεια και στο οποίο:



ως **Αναλογική Είσοδο**: χρησιμοποιούμε μια μεταβαλλόμενη αναλογική τάση από το ποτενσιόμετρο στην αναλογική είσοδο **A0** του Arduino UNO και την εντολή:

```
val = analogRead(analogPin);
```

Η μεταβλητή **val** θα παίρνει τιμές από **0-1023** (επίπεδα κβάντισης του 12 bits A/D μετατροπέα) και τα οποία στη συνέχεια μετατρέπονται σε τιμές αναλογικής τάσης **0-5 Volts** με την εντολή:

```
voltage=val*5.0/1024;
```

ενώ ως **Έξοδο**: χρησιμοποιούμε έναν ακροδέκτη **PWM** (ψηφιακές έξοδοι με το σύμβολο ~) που στην περίπτωση μας είναι ο ψηφιακός ακροδέκτης 11 στέλνοντας έναν τετραγωνικό παλμό με διάρκεια **0-255** με την εντολή:





```
analogWrite(ledPin, duration);
```

Επειδή η παράμετρος διάρκειας είναι το 1/4 των επιπέδων κβάντισης του A/D μετατροπέα, μπορούμε να τροποποιήσουμε την προηγούμενη εντολή όπως παρακάτω, ώστε να έχουμε την αντιστοίχιση σε τιμές τάσης **0-5 Volts**:

```
analogWrite(ledPin, val/4);
```

Η τιμή της Αναλογική τάσης εμφανίζεται στη σειριακή οθόνη του Arduino IDE αλλά και του TINKERCAD με την εντολή:

```
Serial.println(voltage);
```

Επιπλέον στο περιβάλλον tinkercad έχουμε προσθέσει και ένα εικονικό πολύμετρο για λόγους εξοικείωσης με τα ηλεκτρονικά κυκλώματα.

2.4.3 Ο κώδικας στο TINKERCAD και στο ARDUINO IDE

Ο κώδικας σε Wiring C στο Arduino IDE αλλά και στο περιβάλλον TIMKERCAD είναι ο παρακάτω όπου δίπλα σε κάθε εντολή υπάρχει και σχόλιο για τη λειτουργία της:

```
/* ΜΙΑ ΝΕΑ ΑΡΧΗ ΓΙΑ ΤΑ ΕΠΑΛ- ΕΛΕΓΧΟΣ DC-κινητήρα  
*/
```

```
/*δήλωση σταθεράς potpin με την οποία αναφερόμαστε στον  
ακροδέκτη A0 */
```

```
const int potpin = A0;
```

```
/*δήλωση ακέραιας σταθεράς motorpin με την οποία  
αναφερόμαστε στον ακροδέκτη 11 */
```

```
const int motorpin = 11;
```





/* δήλωση **ακέραιας μεταβλητής** val που θα έχει τις τιμές **0-1023** από τον A/D μετατροπέα της εισόδου A0 στην οποία συνδέεται ο μεσαίος ακροδέκτης του ποτενσιόμετρου */

```
int val =0;
```

/* δήλωση **μεταβλητής** voltage με **δεκαδικό μέρος** για τιμές αναλογικής τάσης μετά τη μετατροπή από τις ακέραιες τιμές της μεταβλητής val */

```
float voltage = 0.0;
```

```
void setup() {
```

```
/* δήλωση του ακροδέκτη motorpin (11) ως έξοδος-ωστόσο δεν είναι απαραίτητο για ακροδέκτες PWM */
```

```
pinMode(motorpin, OUTPUT);
```

```
/* ορισμός της ταχύτητας σειριακής επικοινωνίας της πλακέτας Arduino UNO με το Serial Monitor του Arduino IDE και του TINKERCAD */
```

```
Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
/* ανάγνωση τιμών ψηφιακών επιπέδων (0-1023) από το αναλογικό σήμα εισόδου που παράγεται από το ποτενσιόμετρο στον ακροδέκτη αναλογικής εισόδου */
```

```
val = analogRead(potpin);
```





/* μεταφορά αντίστοιχων τιμών στον ακροδέκτη εξόδου σε τιμές 0-255 για την παραγωγή αναλογικού σήματος εξόδου*/

```
analogWrite(motorpin, val/4);
```

/* μετατροπή των τιμών ψηφιακών επιπέδων (0-1023) από το αναλογικό σήμα εισόδου σε τιμές αντίστοιχης αναλογικής τάσης σε Volts */

```
voltage=val*5.0/1024;
```

/* αποστολή των τιμών της αναλογικής τάσης στη Σειριακή Οθόνη είτε του Arduino IDE είτε του tinkercad

```
Serial.print("\n Voltage:\t");
```

```
Serial.print(voltage);
```

```
delay(20);
```

```
}
```

2.5 Αναλογική έξοδος με το Arduino – Οδήγηση πιεζοηλεκτρικού βομβητή

Στόχοι: α) η εξοικείωση με τα δύο είδη πιεζοηλεκτρικών βομβητών

β) η εξοικείωση με τις συναρτήσεις **analogWrite()** και **analogRead()** καθώς και τη συνάρτηση **map()**.

2.5.1 Ο Ενεργός και ο παθητικός πιεζοηλεκτρικός βομβητής

Ο **ενεργός (active) βομβητής (buzzer)** παράγει έναν ήχο, απλά στέλνοντας ένα λογικό '1' (+5 V) στο θετικό ακροδέκτη του ενώ ο αρνητικός είναι στη γείωση. Βασικά τον ενεργοποιείς ή τον απενεργοποιείς.

Ο **παθητικός (passive) βομβητής (buzzer)** χρειάζεται μια πηγή σήματος που παρέχει το ηχητικό σήμα στον θετικό ακροδέκτη ενώ ο αρνητικός είναι στη γείωση. Ανάλογα με τα χαρακτηριστικά του σήματος (πλάτος, συχνότητα) μπορείς να έχεις και διαφορετικό ήχο.





Για να καταλάβουμε την κατηγορία του βομβητή μετράμε αντίσταση μεταξύ των δύο ακροδεκτών του. Εάν είναι λίγα Ω, τότε είναι παθητικός, ενώ οι υψηλότερες τιμές υποδεικνύουν ενεργό βομβητή. Επίσης ο ενεργός βομβητής θα έχει το δικό του κύκλωμα (το pcb που βλέπουμε στο κάτω μέρος του βομβητή).

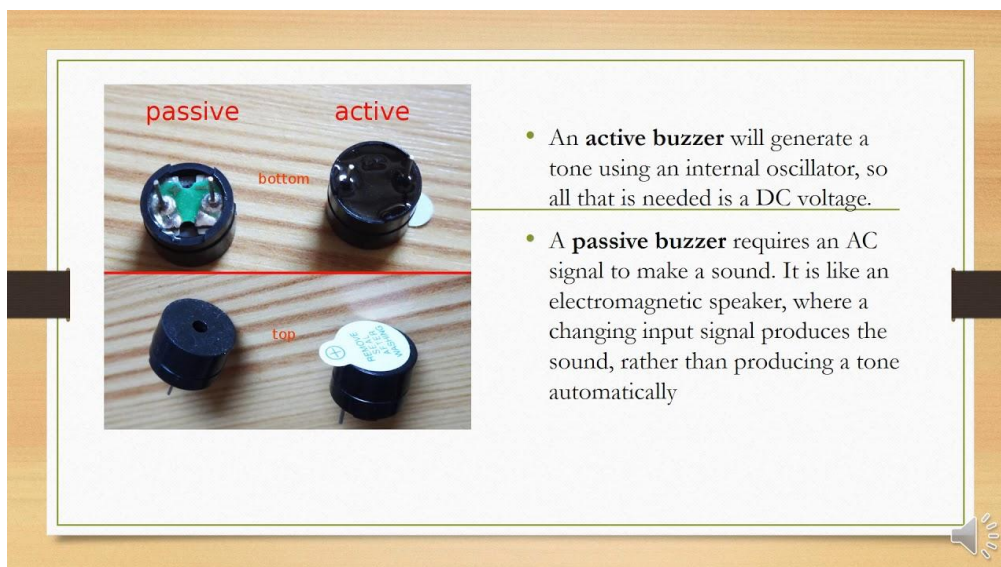
Για τις εντολές προγραμματισμού μπορούμε να χρησιμοποιήσουμε τις παρακάτω

Ενεργός βομβητής: χρησιμοποιώντας ένα απλό **digitalWrite (buzzerPin, HIGH)** θα ενεργοποιηθεί το ηχητικό σήμα, αφού έχει έναν εσωτερικό ταλαντωτή.

Παθητικός βομβητής: πρέπει να χρησιμοποιήσουμε τη συνάρτηση **Tone()** για να παραχθεί. Επειδή δεν έχει εσωτερικό ταλαντωτή, πρέπει να χρησιμοποιήσουμε τη λειτουργία **Tone()** για να δημιουργήσετε τη συχνότητα που θα ταλαντωθεί. Ενδεικτικά με η εντολή **Tone(3, 440)**, θα δημιουργήσει ένα ήχο 440Hz σε παθητικό βομβητή συνδεδεμένο στον ακροδέκτη 3 του Arduino Uno.

Για να σταματήσουμε τον ήχο:

σε έναν **ενεργό βομβητή** χρησιμοποιούμε την **digitalWrite (buzzerPin, LOW)**, ενώ σε έναν **παθητικό βομβητή** χρησιμοποιούμε την **noTone (passiveBuzzerPin)**.



<https://www.youtube.com/watch?v=1XFhu05ap3c>





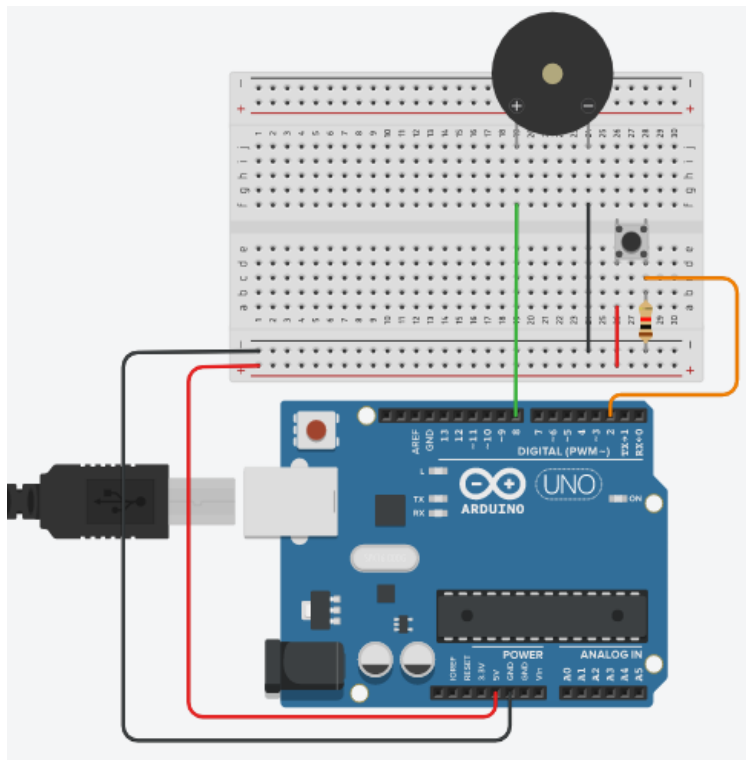
2.5.2 Δραστηριότητα με ενεργό πιεζοηλεκτρικό βομβητή

Τα Υλικά που θα χρειαστούμε:

- 1 Πλακέτα Arduino Uno και ένα breadboard
- 2 Ένας ενεργός πιεζοηλεκτρικός βομβητής
- 3 1x button
- 4 1x220 Ω

Τα κύκλωμα που σχεδιάσαμε:

Κάθε φορά που πατάμε το διακόπτη ενεργοποιείται ο βομβητής. Όταν δεν είναι πατημένος ο διακόπτης ο βομβητής δεν λειτουργεί



Ο κώδικας που χρησιμοποιήσαμε:





// δήλωση μεταβλητών:

```
int button = 0; /* μεταβλητή που διαβάζει το button  
                  θέτοντας αρχική τιμή 0 (δεν πατήθηκε)*/
```

```
void setup() {
```

```
  pinMode(8, OUTPUT); //ορίζουμε τον ακροδέκτη  
                          // ως ψηφιακή έξοδο
```

```
  pinMode(2, INPUT); //ορίζουμε τον ακροδέκτη 2  
                          // ως ψηφιακή είσοδο
```

```
}
```

```
void loop() {
```

```
  button = digitalRead(2); /* διαβάζουμε 0 ή 1 από τον  
                          ακροδέκτη 2 για να δούμε αν πατήθηκε ο διακόπτης  
                          και θέτουμε το αποτέλεσμα στην τιμή της μεταβλητής  
                          button */
```

```
  if (button == HIGH)                   // αν πατήθηκε το button  
  {
```

```
    digitalWrite(8, HIGH);           // ήχος στο βομβητή
```

```
  }
```

```
  else                                 // αλλιώς
```

```
  {
```

```
    digitalWrite(8, LOW);           // σβήσε το βομβητή
```





}

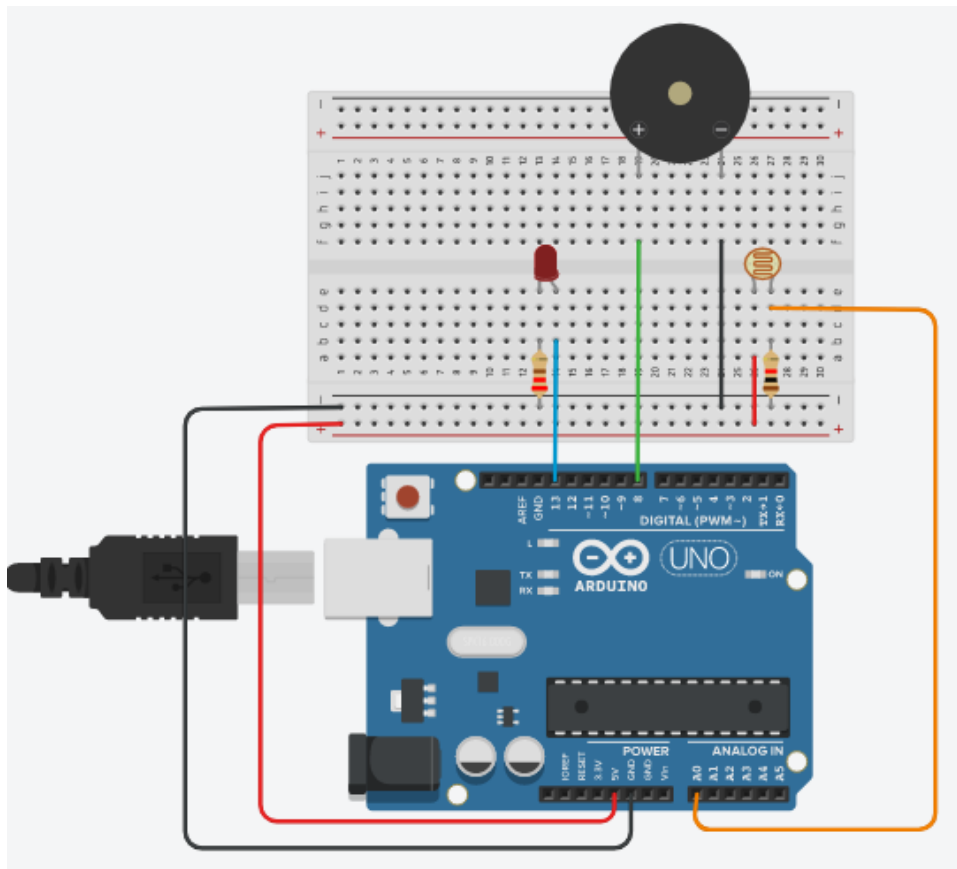
}

2.5.3 Δραστηριότητα με ενεργό πιεζοηλεκτρικό βομβητή

Τα Υλικά που θα χρειαστούμε:

- 5 Πλακέτα Arduino Uno και ένα breadboard
- 6 Ένας παθητικός πιεζοηλεκτρικός βομβητής
- 7 1x LED
- 8 1x φωτοαντίσταση
- 9 1x220 Ω και 1x1ΚΩ

Τα κύκλωμα που σχεδιάσαμε:





Ο κώδικας που θα χρησιμοποιήσουμε:

Ο κώδικας αυτός προέρχεται από τα σεμινάρια **etwinning Arduino Intermediate: Εμβάθυνση στην εκπαιδευτική ρομποτική** με χρήση του Arduino με επιμορφωτή τον εκπαιδευτικό – επιμορφωτή **Παναγιώτη Βεργεράκη ΠΕ86** στο 1^ο ΕΠΑΛ Ρεθύμνου

```
int sensorValue;  
int sensorLow=1023;  
int sensorHigh=0;  
const int ledPin=13;
```

```
/* εντός του χρονικού διαστήματος των 5 sec, σκεπάζεις  
τη φωτοαντίσταση για να βρείς το low limit και την  
ξεσκεπάζεις για να βρείς το High limit  
όταν σβήσει το LED μπορείς να παίξεις με το φωτισμό στη  
φωτοαντίσταση */
```

```
void setup()  
{  
  pinMode(ledPin, OUTPUT);  
  digitalWrite(ledPin, HIGH);  
  while (millis()<5000) {  
    sensorValue=analogRead(A0);  
    if (sensorValue>sensorHigh) {  
      sensorHigh=sensorValue;
```





```
}  
if (sensorValue<sensorLow) {  
    sensorLow=sensorValue;  
}  
}  
digitalWrite(ledPin, LOW); //τέλος του calibration  
}  
  
void loop()  
{  
sensorValue=analogRead(A0) ;  
/* με τη συνάρτηση map προσαρμόζουμε τη μέτρηση της  
έντασης του φωτός από τη φωτοαντίσταση από την παραπάνω  
εντολή σε τιμές που προσαρμόζονται ανάμεσα στις  
συχνότητες των 50-4000 HZ λαμβάνοντας υπόψη το φωτισμό  
περιβάλλοντος όταν υπάρχει πλήρες σκοτάδι ή μέγιστο φώς.  
Η παραγόμενη τιμή της συχνότητας γίνεται όρισμα στη  
συνάρτηση tone μέσω της μεταβλητής pitch*/  
int pitch=map(sensorValue,sensorLow,sensorHigh,50,4000) ;  
/* παραγωγή σήματος ήχου στον ακροδέκτη 8, με συχνότητα  
pitch για 20 msec */  
tone(8,pitch,20) ;  
delay(10) ;  
}
```

2.6 ΕΛΕΓΧΟΣ ΚΙΝΗΤΗΡΑ micro SERVO





Στις επόμενες δύο παραγράφους παραθέτουμε οδηγίες εξοικείωσης για την οδήγηση ενός πολύ απλού σερβοκινητήρα που συναντάμε σε δραστηριότητες μοντελισμού. Πληρέστερες οδηγίες για έλεγχο κινητήρων μπορείτε να βρείτε σε παραδοτέο υλικό από προηγούμενα Σχέδια Δράσης του 1^{ου} ΕΠΑΛ Συκεών στα πλαίσια του προγράμματος ΜΙΑ ΝΕΑΣ ΑΡΧΗ ΓΙΑ ΤΑ ΕΠΑΛ στα σχολικά έτη του σχολικού έτους 2019-2020 και 2020-2021.

2.6.1 Έλεγχος micro SERVO MOTOR με την κλάση Servo της Arduino

Η βιβλιοθήκη Servo μας βοηθάει στον έλεγχο μέσω του Arduino, σερβοκινητήρων (Servo Motor) αντίστοιχων με αυτούς που χρησιμοποιούνται στο ραδιομοντελισμό (RC-Hobby servo motor).

Η ενσωμάτωση της βιβλιοθήκης **Servo** της **Arduino** στον κώδικά μας γίνεται με δύο τρόπους:

α) απευθείας μέσω της εντολής: **#include <Servo.h>** ή

β) από τα μενού: **Σχέδιο (Sketch)/Συμπερίληψη Βιβλιοθήκης/Servo** οπότε αυτόματα εισάγεται η παραπάνω εντολή στο τμήμα δηλώσεων του κώδικα.

Αξιοποιώντας τη βιβλιοθήκη Servo:

δεν θα κάνουμε χρήση των εντολών `pinmode(pin, input/output)` και `analogwrite(pin, value)` αλλά,

θα χρησιμοποιήσουμε ενσωματωμένες συναρτήσεις - εντολές της βιβλιοθήκης Servo που τις βρίσκουμε στη διεύθυνση <https://www.arduino.cc/en/Reference/Servo>

2.6.2 Πληροφορίες για τον κινητήρα micro Servo της Arduino αλλά και άλλων εταιριών

Ο κινητήρας micro Servo διαθέτει σύστημα από γρανάζια (integrated gears) και έναν άξονα (shaft) και μπορεί να ελεγχθεί με ακρίβεια. Τα πιο συνηθισμένα servo motors





επιτρέπουν ο άξονας να τοποθετείται σε διάφορες γωνίες, συνήθως μεταξύ 0 και 180 μοιρών. Το servo motor που χρησιμοποιούμε έχει τρεις ακροδέκτες:



Κόκκινο :για +5 Volts

Μαύρο : για γείωση

Πορτοκαλί : για σύνδεση του σήματος PWM από έναν αντίστοιχο ψηφιακό ακροδέκτη της πλακέτας Arduino UNO με την ένδειξη ~

Με ένα ποτενσιόμετρο δημιουργούμε μια μεταβαλλόμενη αναλογική τάση εισόδου, όπως και στην περίπτωση του DC κινητήρα, η οποία διαβάζεται από έναν αναλογικό ακροδέκτη του μικροελεγκτή. Στη συνέχεια ο μικροελεγκτής δημιουργεί ένα σήμα εξόδου PWM με το οποίο καθορίζεται η γωνία στροφής του servo motor. Την τεχνική PWM την αναλύουμε σε προηγούμενη δραστηριότητα.

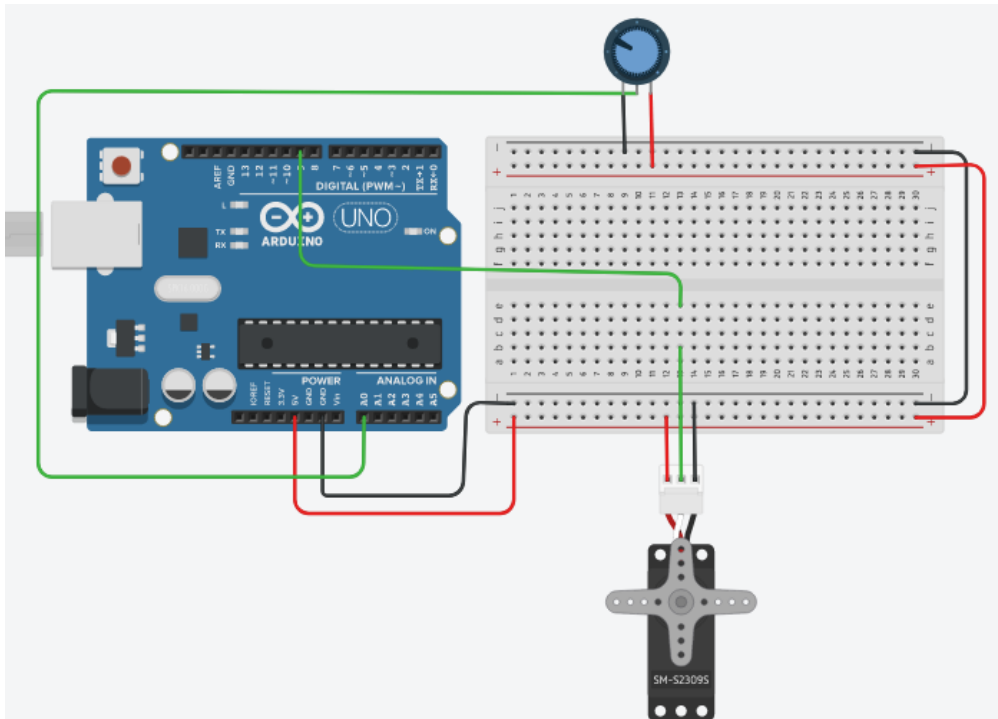
Στο κύκλωμα στο περιβάλλον TINKERCAD που δίνεται στην παρακάτω εικόνα:

οι εξωτερικοί ακροδέκτες του ποτενσιόμετρου και οι ακροδέκτες τροφοδοσίας του κινητήρα Servo τροφοδοτούνται από τα 5 V της πλακέτας Arduino UNO R3.

ο μεσαίος ακροδέκτης του ποτενσιόμετρου συνδέεται στον ακροδέκτη αναλογικής εισόδου A0 της πλακέτας Arduino UNO R3

ο μεσαίος ακροδέκτης του κινητήρα Servo συνδέεται στον ακροδέκτη αναλογικής εξόδου(PWM) 9 της πλακέτας Arduino UNO R3





2.6.3 Ανάλυση του κώδικα.

Διαβάζουμε μια μεταβαλλόμενη αναλογική τάση στην αναλογική είσοδο **A0** του Arduino UNO από το ποτενσιόμετρο που αντιστοιχεί σε επίπεδα κβάντισης 0-1023 (0-5 Volts) με την εντολή `val = analogRead(analogPin);`

Μετατρέπουμε την τιμή `val` με εύρος τιμών από 0-1023 σε τιμές με εύρος τιμών 0-255 με την εντολή `val = map(val, 0, 1023, 0, 180);` ώστε να είναι συμβατή με τις τιμές με τις οποίες λειτουργεί το servo motor.

Στέλνουμε στην έξοδο PWM (ψηφιακές έξοδοι με το σύμβολο ~) που στην περίπτωσή μας είναι ο ψηφιακός ακροδέκτης **9**, ένα τετραγωνικό παλμό με διάρκεια 0-180 (0-5 Volts) με την εντολή `myservo.write(val);`





Ο κώδικας για την εφαρμογή

/* φορτώνουμε τη βιβλιοθήκη Servo για τον έλεγχο του **servo motor** */

```
#include <Servo.h>
```

/* δημιουργούμε ένα αντικείμενο της κλάσης Servo με το όνομα `srvmotor1`. Από εδώ και πέρα το αντικείμενο `srvmotor1` θα λειτουργεί με τις ίδιες εντολές που συμπεριλαμβάνονται στη βιβλιοθήκη κλάση Servo. Αν έχω και δεύτερο σερβοκινητήρα δημιουργώ και δεύτερο αντικείμενο `srvmotor2` με την εντολή `Servo srvmotor2`; και στη συνέχεια αντιγράφω τις ίδιες εντολές ελέγχου για το αντικείμενο `srvmotor2` */

```
Servo srvmotor1;
```

/* ορίζω με τη μεταβλητή `potpin`, σε ποια αναλογική είσοδο (`pin`) θα συνδέσω το potentiometer, δηλαδή την είσοδο `A0` */

```
int potpin = 0;
```

/* ορίζω τη μεταβλητή `val` που θα παίρνει τις τιμές από το potentiometer. Οι τιμές αυτές θα έχουν εύρος από 0-1023 λόγω του 10 bit A/D μετατροπέα του Arduino που εφαρμόζεται στον αναλογικό ακροδέκτη `A0` */

```
int val;
```

```
void setup() {
```

/* με την εντολή `attach` της κλάσης Servo δηλώνω ότι το αντικείμενο `srvmotor1` που αντιστοιχεί στο σερβοκινητήρα μου θα παίρνει σήμα από τον





ακροδέκτη 9 της πλακέτας Arduino Uno. Συνδέω δηλαδή το πορτοκαλί καλώδιο του σερβοκινητήρα στην υποδοχή του Arduino Uno ~9 */

```
srvmotor1.attach(9);  
}
```

```
void loop() {
```

```
/* διαβάζω την αναλογική τάση από το potentiometer σε κλίμακα από 0  
έως 1023(A/D) */
```

```
val = analogRead(protin);
```

```
/* αλλάζω την κλίμακα της μεταβλητής val από 0-1023 σε κλίμακα 0-180  
στην οποία είναι η γωνία περιστροφής του servo motor */
```

```
val = map(val, 0, 1023, 0, 180);
```

```
/* με την εντολή write της κλάσης Servo στρέφω το αντικείμενο srvmotor1  
(δηλαδή servo motor) σε γωνίες 0-180 */
```

```
srvmotor1.write(val);
```

```
/* καθυστέρηση 15 ms ώστε να φτάσει στη θέση του, το servo motor */
```

```
delay(15);
```

```
}
```

2.7 Οδήγηση Μονοπολικού Βηματικού Κινητήρα (Unipolar Stepper Motor) με Arduino





Στη συνέχεια δίνονται οδηγίες μόνο για το μονοπολικό βηματικό κινητήρα χαμηλού κόστους με τον οποίο μπορούμε να αναπτύξουμε για απλές σχολικές δραστηριότητες. Πληρέστερες οδηγίες για έλεγχο κινητήρων μπορείτε να βρείτε σε παραδοτέο υλικό από προηγούμενα Σχέδια Δράσης του 1^{ου} ΕΠΑΛ Συκεών στα πλαίσια του προγράμματος ΜΙΑ ΝΕΑΣ ΑΡΧΗ ΓΙΑ ΤΑ ΕΠΑΛ στα σχολικά έτη του σχολικού έτους 2019-2020 και 2020-2021.

2.7.1 Κατηγορίες των Βηματικών κινητήρων.

Οι βηματικοί κινητήρες διαφέρουν από τους άλλους τύπους κινητήρων συνεχούς και εναλλασσομένου ρεύματος στο ότι τροφοδοτούνται με παλμούς και παράγουν ηλεκτρική κίνηση. Ο άξονας τους δεν έχει μια συνεχή περιστροφική κίνηση, αλλά περιστρέφεται κατά μία γωνία κάθε φορά που δέχεται ένα παλμό. Ένας βηματικός κινητήρας είναι ένας κινητήρας ελεγχόμενος από μια σειρά ηλεκτρομαγνητικών σπειρών. Ο κεντρικός άξονας έχει μια σειρά από μαγνήτες προσαρμοσμένους πάνω του και πηνία που τον περιβάλλουν. Στα πηνία δίδεται διαδοχικά ηλεκτρικό ρεύμα ή όχι, δημιουργώντας μαγνητικά πεδία τα οποία απωθούν ή έλκουν τους μαγνήτες του άξονα, προκαλώντας την περιστροφή του κινητήρα.

Αυτός ο σχεδιασμός επιτρέπει τον πολύ ακριβή έλεγχο του κινητήρα: με κατάλληλη παλμική κίνηση, μπορεί να γυρίσει σε πολύ ακριβή βήματα. Χρησιμοποιούνται σε εκτυπωτές, δίσκους και άλλες συσκευές όπου απαιτείται ακριβής μετακίνηση του κινητήρα. Υπάρχουν δύο βασικοί τύποι βηματικών κινητήρων, οι **μονοπολικοί βηματικοί κινητήρες** και οι **διπολικοί βηματικοί κινητήρες**. Στο εγχειρίδιο αυτό θα ασχοληθούμε μόνο με τον μονοπολικό βηματικό κινητήρα που είναι πολύ χαμηλού κόστους και βρίσκεται πολύ εύκολα για αγορά από το διαδίκτυο.

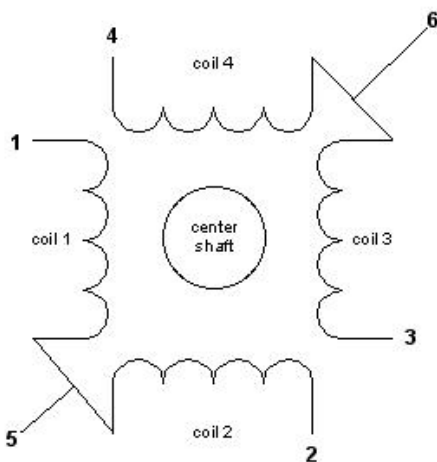




2.7.2 Μονοπολικός βηματικός κινητήρας

Ο μονοπολικός βηματικός κινητήρας έχει πέντε ή έξι σύρματα και τέσσερα πηνία (στην πραγματικότητα δύο πηνία διαιρούμενα με κεντρικές συνδέσεις σε κάθε πηνίο). Οι κεντρικές συνδέσεις των πηνίων συνδέονται μεταξύ τους και χρησιμοποιούνται ως σύνδεση ρεύματος. Ονομάζονται μονοπολικοί, επειδή η δύναμη πάντα έρχεται σε αυτόν τον ένα πόλο.

Όπως και οι άλλοι κινητήρες, οι βηματικοί κινητήρες απαιτούν περισσότερη ενέργεια από ό, τι μπορεί να τους δώσει ένας μικροελεγκτής, οπότε θα χρειαστούμε ξεχωριστό τροφοδοτικό. Στην ιδανική περίπτωση θα γνωρίζουμε την τάση από τον κατασκευαστή. Για την οδήγηση-έλεγχο των βηματικών κινητήρων υπάρχουν πλακέτες-οδηγοί (drivers) ανάλογα με τον κινητήρα και τα χαρακτηριστικά λειτουργίας του όπως αυτή που θα δούμε στη συνέχεια.



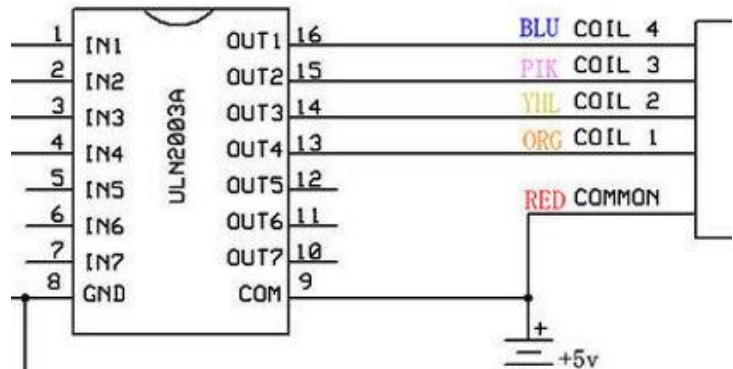
2.7.3 Οδήγηση μονοπολικού βηματικού κινητήρα με την πλακέτα οδηγού ULN2003

Στο διαδίκτυο υπάρχει μεγάλη ποικιλία οδηγών και βηματικών κινητήρων που συνεργάζονται με τις πλακέτες Arduino UNO όπως ενδεικτικά φαίνεται παρακάτω:



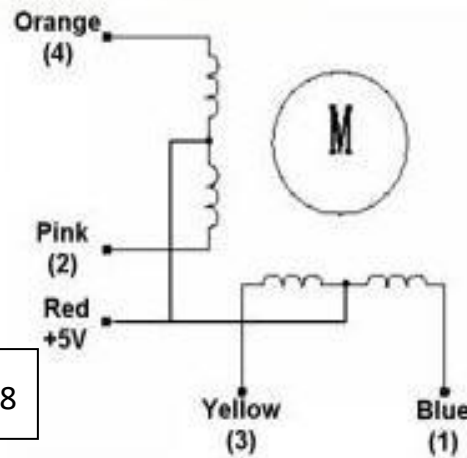


Jumper στην πλακέτα Οδηγού ULN2003:
για να τροφοδοτείται ο κινητήρας θα
πρέπει να έχουμε συνδέσει το jumper
που βρίσκεται δεξιά από τους
ακροδέκτες τροφοδοσίας (-, + 5-12 V).

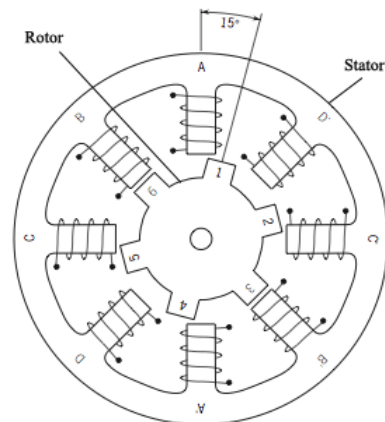
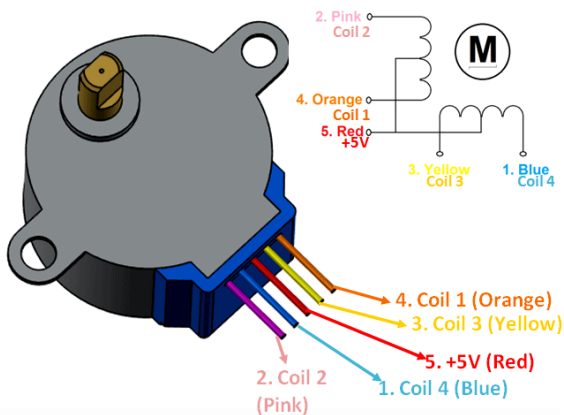


Πλακέτα Οδηγού ULN2003

Μοτέρ 28-ΥJ48



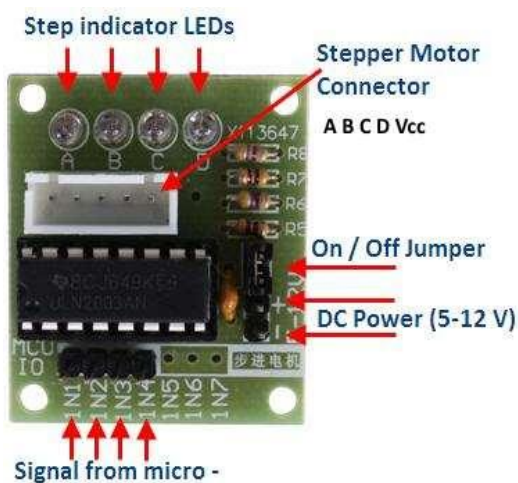
Ο κινητήρας 28-BYJ48 είναι μονοπολικός (4 φάσεις) και συνδέεται με τον οδηγό του (ULN2003) με 5 καλώδια τα οποία τροφοδοτούν 2 ζεύγη πηνίων (πορτοκαλί-ροζ), (κίτρινο-μπλε). Το κόκκινο καλώδιο (+5V) συνδέεται στη μεσαία λήψη του κάθε ζεύγους.





Ο άξονας του κινητήρα δεν περιστρέφεται συνεχώς αλλά περιστρέφεται κατά μία γωνία κάθε φορά που δέχεται ένα παλμό. Ο κεντρικός άξονας έχει μια σειρά από μαγνήτες προσαρμοσμένους πάνω του ενώ ο ρότορας του κινητήρα αποτελείται από 2 ζεύγη πηνίων όπως αναφέραμε προηγουμένως. Στα πηνία διοχετεύουμε διαδοχικά παλμούς και τα μαγνητικά πεδία που δημιουργούνται απωθούν ή έλκουν τους μαγνήτες του άξονα με αποτέλεσμα τη περιστροφή του κινητήρα προς τη μία ή την άλλη κατεύθυνση. Με αυτό τον τρόπο λοιπόν μπορούμε να ελέγξουμε τη φορά περιστροφής του κινητήρα ή την ταχύτητα της περιστροφής ανάλογα με τη συχνότητα του παλμού.

Μονάδα οδηγού: **ULN2003 Stepper Motor Driver** (αν χρησιμοποιήσουμε τον L293 driver θα αφήσουμε το κόκκινο καλώδιο ασύνδετο)



Τεχνικά Χαρακτηριστικά:

Rated voltage 5V (DC), Number of Phase 4, Speed Variation Ratio 1/64, Stride Angle 5.625°/64, Frequency 100Hz, DC resistance 50Ω±7%, Idle In-traction Frequency > 600Hz, Idle Out-traction Frequency > 1000Hz, In-traction Torque >34.3mN.m(120Hz), Self-positioning Torque >34.3mN.m, Friction torque 600-1200 gf.cm, Pull in torque 300 gf.cm

Ο οδηγός ULN2003 αποτελείται από τέσσερα NPN τρανζίστορ σε συνδεσμολογία κοινού εκπομπού (γέφυρα Darlington) ικανά να δώσουν 500 mA στην έξοδο, ενισχύοντας το σήμα ελέγχου από τις ψηφιακές εξόδους του Arduino

Υπολογισμός βημάτων για την περιστροφή





Το μοτέρ έχει ένα λόγο γραναζιών : Gear ratio=64

ενώ η γωνία που κάνει σε κάθε βήμα (Stride Angle)= 5.625°

ARDUINO UNO

οδηγός ULN2003

Επομένως για να κάνει

το μοτέρ 28-BYJ48 μια

πλήρη περιστροφή

χρειαζόμαστε:

Steps_ in_ One

Revolution = 360° /

5.625°= 64

steps = Steps_ in_ One

Revolution * Gear ratio

= 64 * 64 =4096 steps

11

IN4

10

IN3

9

IN2

8

IN1

5 V

+ (5-12 V)

GND

(5-12 V)

Jumper ON ώστε να παίρνει

τροφοδοσία το μοτέρ

Οι παραπάνω υπολογισμοί είναι διαφορετικοί αν έχουμε έναν άλλο βηματικό κινητήρα όπως για παράδειγμα το **adafruit Stepper Motor** :

Το μοτέρ αυτό έχει ένα λόγο γραναζιών : Gear ratio=16

ενώ η γωνία που κάνει σε κάθε βήμα (Stride Angle)= 7.5°

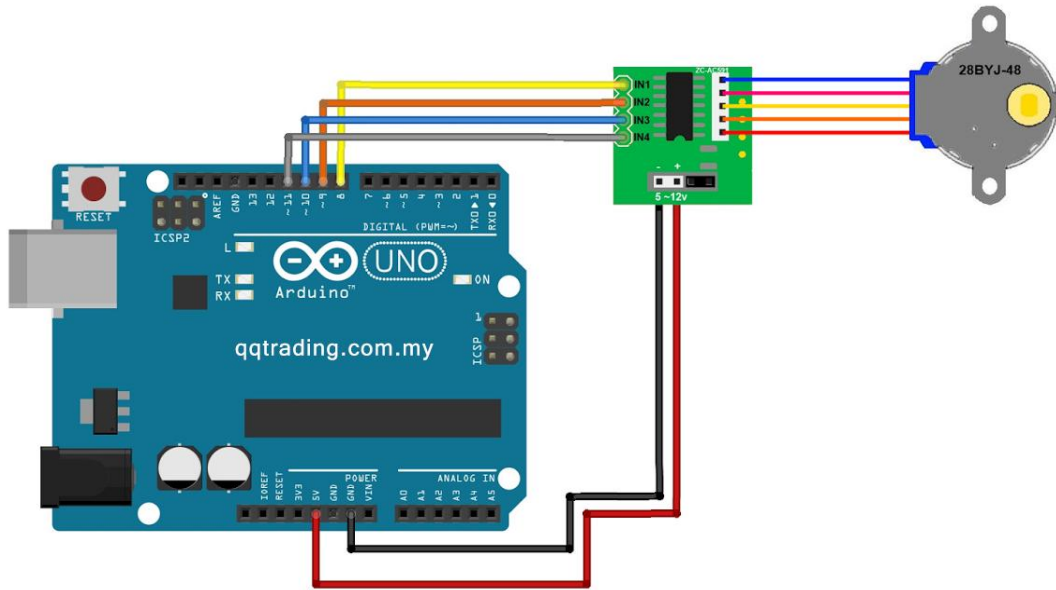
Επομένως για να κάνει το **adafruit Stepper Motor** μια πλήρη περιστροφή χρειαζόμαστε:

Number of steps in One Revolution = 360° / 7.5°= 48

steps = Number of steps in One Revolution * Gear ratio = 48 * 16 =768 steps

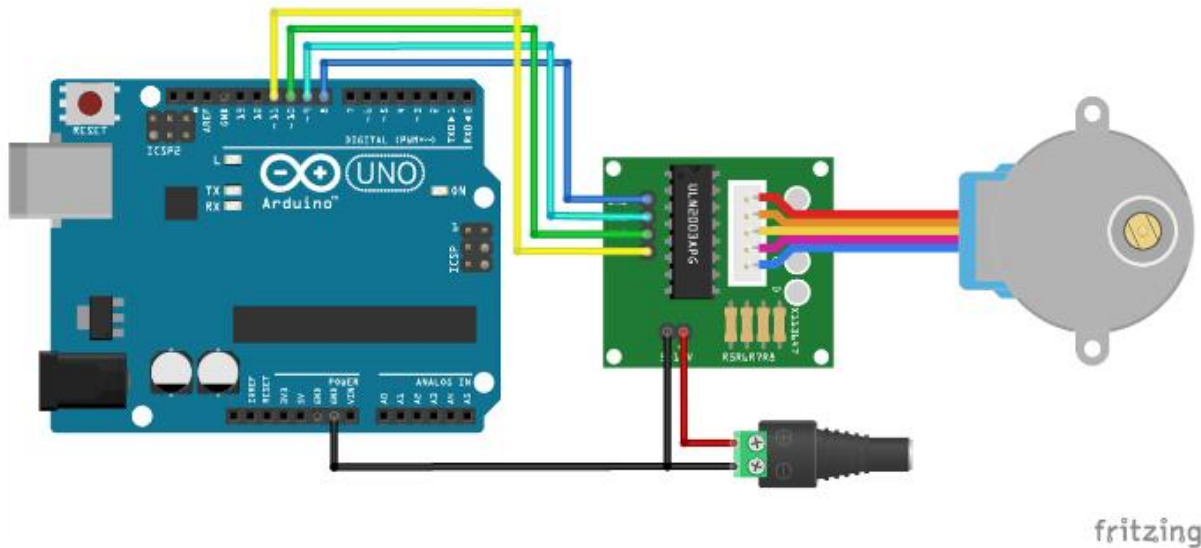
Στις παρακάτω εικόνες βλέπουμε τη σύνδεση της πλακέτας οδηγού ULN2003 με το ARDUINO UNO και το μοτέρ 28BYJ-48 με τροφοδοσία του οδηγού είτε από την πλακέτα Arduino είτε από εξωτερικό τροφοδοτικό (συνιστάται).





<https://www.makerguides.com/28byj-48-stepper-motor-arduino-tutorial/>

Για τη σύνδεση της πλακέτας οδηγού με το μοτέρ χρησιμοποιούμε το καλώδιο του μοτέρ. Δείτε σε προηγούμενο σχήματα οδηγίες για τη σύνδεση.



Wiring diagram for ULN2003 driver with 28BYJ-48 stepper motor and Arduino.





2.7.5 Κώδικας Οδήγησης του βηματικού κινητήρα 28BYJ-48 με τον οδηγό ULN2003 μέσω ενσωματωμένων βιβλιοθηκών στο Arduino IDE

Ο κώδικας οδήγησης του βηματικού κινητήρα απλουστεύεται πάρα πολύ όταν χρησιμοποιούμε τις ενσωματωμένες βιβλιοθήκες του περιβάλλοντος Arduino IDE.

Η ενσωμάτωση της βιβλιοθήκης **Stepper** της **Arduino** στον κώδικά μας γίνεται με δύο τρόπους:

α) απευθείας μέσω της εντολής **#include <Stepper.h>** ή

β) από τα μενού: **Σχέδιο (Sketch)/Συμπερίληψη Βιβλιοθήκης/Stepper** οπότε αυτόματα εισάγεται η παραπάνω εντολή στο τμήμα δηλώσεων του κώδικα. Αξιοποιώντας τη βιβλιοθήκη **Stepper** θα χρησιμοποιήσουμε ενσωματωμένες συναρτήσεις - εντολές της βιβλιοθήκης που τις βρίσκουμε στη διεύθυνση:

<https://www.arduino.cc/en/Reference/Stepper>

Στην περίπτωση που θέλουμε να χρησιμοποιήσουμε μια βιβλιοθήκη όπως η **AccelStepper.h** που φτιάχτηκε από τον **Mike McCauley** και η οποία δεν υπάρχει εγκατεστημένη στο περιβάλλον IDE της Arduino τότε κάνουμε τα παρακάτω βήματα:

Αρχικά από τα μενού: **Εργαλεία/Διαχειριστής Βιβλιοθήκης** περιμένουμε λίγα sec και μετά βλέπουμε ποιες βιβλιοθήκες είναι εγκατεστημένες και αν βρούμε αυτή που μας ενδιαφέρει απλά ζητάμε **update** (εάν υπάρχει αυτή η επιλογή).

Αν δεν υπάρχει αυτή η βιβλιοθήκη, τότε μετά από αναζήτηση στο διαδίκτυο, κατεβάζουμε τη βιβλιοθήκη σε μορφή συνήθως ZIP και κάνουμε **unzip** σε ένα δικό μας φάκελο.

Στη συνέχεια από τα μενού: **Σχέδιο (Sketch)/Συμπερίληψη Βιβλιοθήκης/Προσθήκη Βιβλιοθήκης** περιμένουμε λίγα sec και μετά δηλώνουμε τον φάκελο στον οποίο βρίσκεται η βιβλιοθήκη. Περιμένουμε πάλι λίγα sec μέχρι





το IDE να μας ενημερώσει για την ολοκλήρωση της εγκατάστασης της βιβλιοθήκης.

Τέλος η ενσωμάτωση βιβλιοθήκης στον κώδικά μας γίνεται από τα μενού **Σχέδιο (Sketch)/Συμπερίληψη Βιβλιοθήκης/AccelStepper** οπότε αυτόματα εισάγονται οι δύο παρακάτω εντολές στο τμήμα δηλώσεων του κώδικα.

```
#include <AccelStepper.h>
```

```
#include <MultiStepper.h>
```

Με τον παρακάτω κώδικα, θέτουμε σε περιστροφή με **μικρή ταχύτητα** το βηματικό κινητήρα διαδοχικά δεξιόστροφα για μία στροφή και μετά αριστερόστροφα πάλι για μία στροφή αξιοποιώντας τη **βιβλιοθήκη Stepper**.

```
/* Συμπερίληψη της βιβλιοθήκης Stepper.h */
```

```
#include <Stepper.h>
```

```
/* Καθορισμός αριθμού παλμών ανά περιστροφή κινητήρα */
```

```
const int stepsPerRevolution = 2048;
```

```
/*Για τη σύνδεση του βηματικού κινητήρα στο κύκλωμα:
```

```
Arduino Pin 8 στο IN1 του ULN2003 driver
```

```
Arduino Pin 9 στο IN2 του ULN2003 driver
```

```
Arduino Pin 10 στο IN3 του ULN2003 driver
```

```
Arduino Pin 11 στο IN4 του ULN2003 driver */
```





```
/* Δημιουργία του προγραμματιστικού αντικειμένου  
'myStepper', με τις ιδιότητες-συναρτήσεις της  
βιβλιοθήκης Stepper */
```

```
Stepper myStepper = Stepper(stepsPerRevolution, 8, 10,  
9, 11);
```

```
void setup() {
```

```
/* Καθορισμός ταχύτητας περιστροφής σε 5 rpm */  
myStepper.setSpeed(5);
```

```
/* Καθορισμός ταχύτητας σειριακής επικοινωνίας με τη  
σειριακή οθόνη του IDE στα 9600 baud rate */
```

```
Serial.begin(9600);
```

```
}
```

```
void loop() {
```

```
Serial.println("clockwise");
```

```
/* Εντολή για μια περιστροφή σε clockwise */
```

```
myStepper.step(stepsPerRevolution);
```

```
delay(500);
```

```
Serial.println("counterclockwise");
```

```
/* Εντολή για μια περιστροφή σε counterclockwise */
```

```
myStepper.step(-stepsPerRevolution);
```

```
delay(500);
```





}

3. Δραστηριότητες STEAM

Στη συνοδευτικό εγχειρίδιο παραθέτουμε φωτογραφίες των κυκλωμάτων με τις προτεινόμενες δραστηριότητες STEAM ενώ στη συνέχεια δίνουμε γενικές οδηγίες ενσωμάτωσης στον κώδικα των απαραίτητων βιβλιοθηκών για τον έλεγχο αισθητήρων, κινητήρων, οθονών κ.λ.π

3.2.1 Οδηγίες ενσωμάτωσης βιβλιοθηκών

Στην ενότητα αυτή παραθέτουμε οδηγίες ενσωμάτωσης των βιβλιοθηκών (εντολές `#include library`) που απαιτούνται στους προτεινόμενους κώδικες των δραστηριοτήτων STEAM χρησιμοποιώντας ως παράδειγμα τη βιβλιοθήκη DHT για τον αισθητήρα μέτρησης θερμοκρασίας και υγρασίας **DHT11**. Αντίστοιχα βήματα πρέπει να ακολουθήσουμε για οποιαδήποτε βιβλιοθήκη απαιτείται στον κώδικα που χρησιμοποιούμε.

A. Αρχικά ενσωματώνουμε τη βιβλιοθήκη **DHTSensorLibrary** της Arduino στον κώδικά μας. Αυτό γίνεται με τα παρακάτω βήματα:

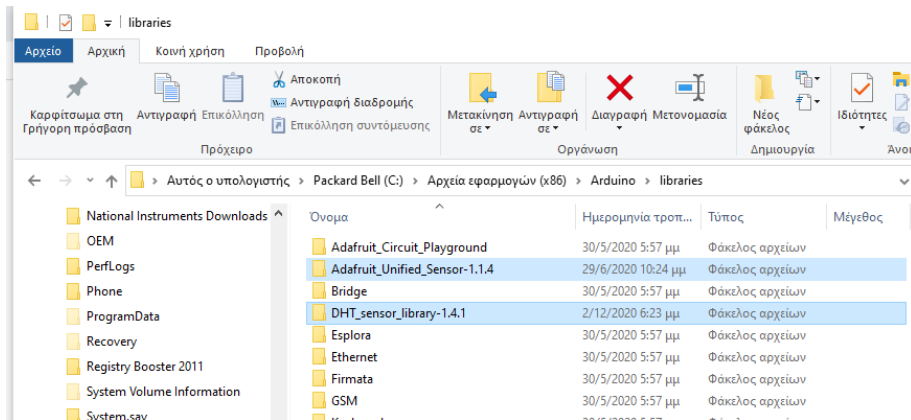
1. Πρώτα κατεβάζουμε τη βιβλιοθήκη στον υπολογιστή μας σε μορφή ZIP από κάποια σχετική ιστοσελίδα κάνοντας αναζήτηση στο διαδίκτυο,

Filename	Release Date	File Size
DHT_sensor_library-1.4.1.zip	2020-12-02	18.31 KIB
DHT_sensor_library-1.4.0.zip	2020-09-16	18.30 KIB
DHT_sensor_library-1.3.10.zip	2020-05-26	18.21 KIB
DHT_sensor_library-1.3.9.zip	2020-04-19	18.11 KIB
DHT_sensor_library-1.3.8.zip	2019-11-02	15.83 KIB
DHT_sensor_library-1.3.7.zip	2019-07-19	13.49 KIB



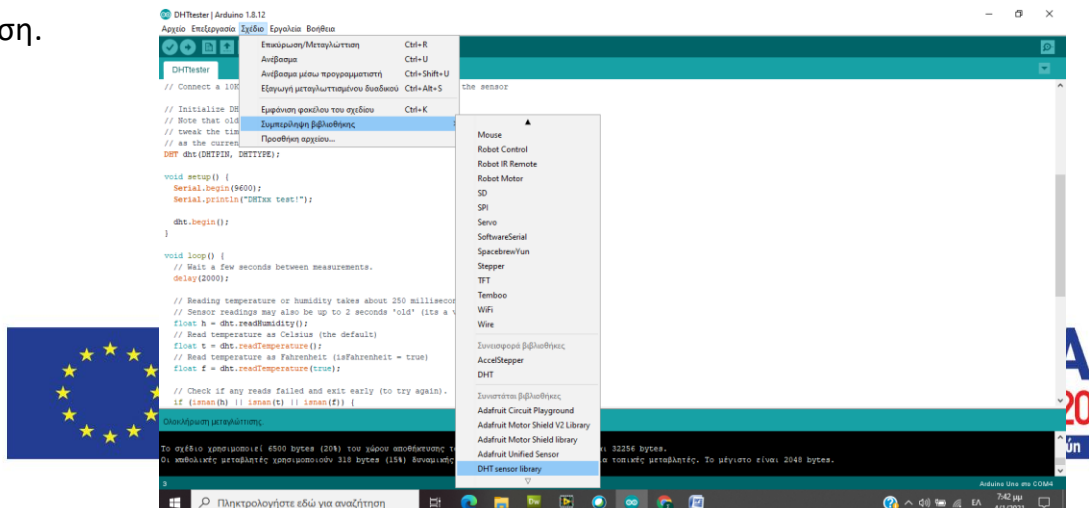


2. Στη συνέχεια αντιγράφουμε τις βιβλιοθήκες στο υποφάκελο **Libraries** του φακέλου **Arduino** στον ΗΥ μας που περιέχει στην εφαρμογή ArduinoIDE



3. Από το περιβάλλον IDE του Arduino και το μενού επιλογών **Σχέδιο** επιλέγουμε **Συμπερίληψη βιβλιοθήκης** και στη συνέχεια **Προσθήκη βιβλιοθήκης ZIP...**, επιλέγουμε το συμπιεσμένο αρχείο της βιβλιοθήκης ή το φάκελο που περιέχει τη βιβλιοθήκη από τον υποφάκελο **C:\ProgramFiles (x86)\Arduino\libraries** και πατάμε το κουμπί **Open**.

4. Κατόπιν, πάλι από το περιβάλλον IDE του Arduino και το μενού επιλογών **Σχέδιο** επιλέγουμε **Συμπερίληψη βιβλιοθήκης** και στη συνέχεια **Προσθήκη βιβλιοθήκης ZIP...**, επιλέγουμε τη βιβλιοθήκη όπως φαίνεται στην παρακάτω εικόνα και πλέον η βιβλιοθήκη έχει ενσωματωθεί στο περιβάλλον εργασίας του Arduino και είναι έτοιμη για χρήση.





5. Κατεβάζουμε στον ΗΥ μας και τη βιβλιοθήκη **Adafruit_Sensor.h** κάνοντας αναζήτηση στο διαδίκτυο

The screenshot shows the Arduino Library List page for the 'Adafruit Unified Sensor' library. The page includes the following information:

- Author:** Adafruit
- Website:** https://github.com/adafruit/Adafruit_Sensor
- Category:** Sensors
- License:** Apache 2.0
- Library Type:** Recommended
- Architectures:** Any

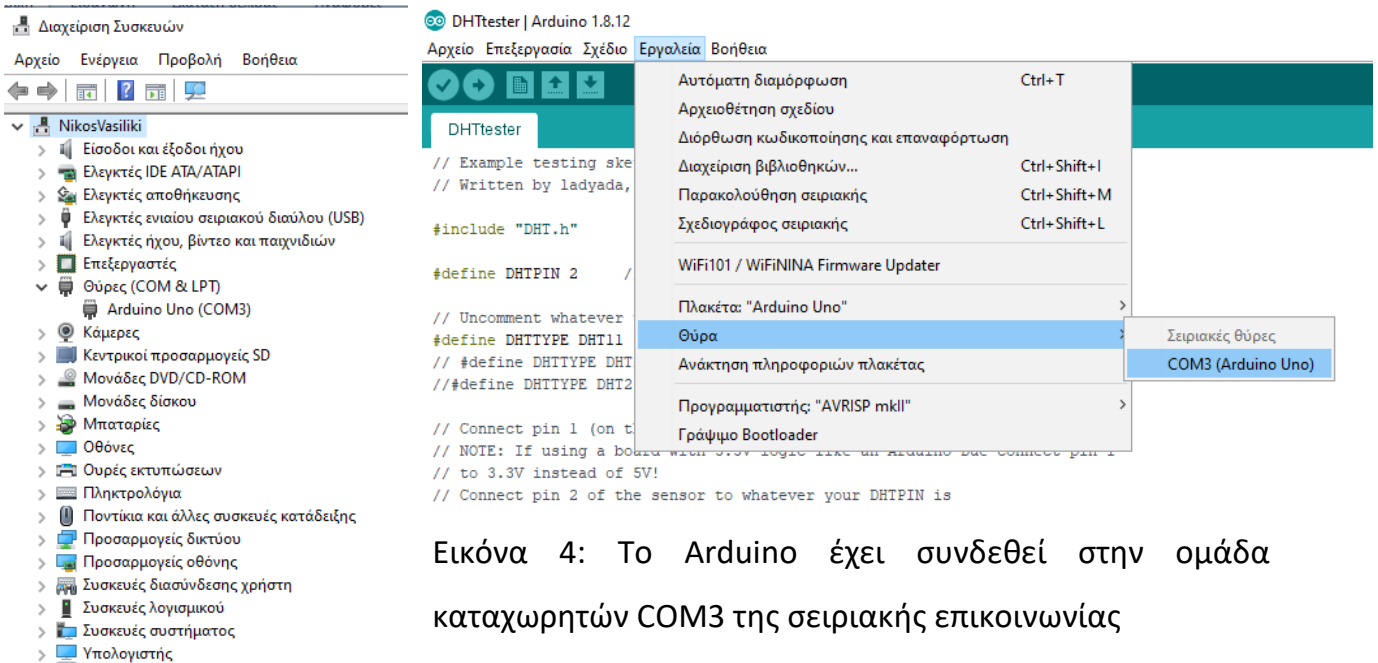
Below the metadata, there is a 'Downloads' section with a table listing various versions of the library:

Filename	Release Date	File Size
Adafruit_Unified_Sensor-1.1.4.zip	2020-06-29	13.93 KiB
Adafruit_Unified_Sensor-1.1.3.zip	2020-05-26	13.94 KiB
Adafruit_Unified_Sensor-1.1.2.zip	2020-02-02	13.94 KiB
Adafruit_Unified_Sensor-1.1.1.zip	2019-12-29	13.91 KiB
Adafruit_Unified_Sensor-1.1.0.zip	2019-12-29	13.90 KiB

και επαναλαμβάνουμε τα βήματα 2,3 και 4 για την ενσωμάτωση και της βιβλιοθήκης **Adafruit_Sensor.h** στον κώδικα μας που συνεργάζεται με την βιβλιοθήκη **DHTSensorLibrary**.

Β. Συνδέουμε το Arduino σε μία θύρα USB του ΗΥ μας και επιβεβαιώνουμε την σύνδεση από τη **Διαχείριση Συσκευών** των Windows ενώ σημειώνουμε σε ποια **θύρα COM?** έχει συνδεθεί ώστε να δηλώσουμε και την αντίστοιχη ρύθμιση στο περιβάλλον προγραμματισμού του Arduino





Εικόνα 4: Το Arduino έχει συνδεθεί στην ομάδα καταχωρητών COM3 της σειριακής επικοινωνίας

3.2.2 Σύνδεσμοι των δραστηριοτήτων STEAM στο TINKERCAD

Στον κατάλογο που ακολουθεί δίνονται οι σύνδεσμοι στην ελεύθερη διαδικτυακή εφαρμογή TINKERCAD για κάθε μια από τις προτεινόμενες δραστηριότητες STEAM. Από εκεί μπορεί ο κάθε ενδιαφερόμενος να δει το κύκλωμα, να αντιγράψει τον κώδικα και κάνει προσομοίωση της λειτουργίας του κυκλώματος κάνοντας κλικ στο εικονίδιο του αντίστοιχου αισθητήρα. Προϋπόθεση να πατήσουμε το κουμπί Simulation.

Αριθμός Δραστηριότητας	Τίτλος Δραστηριότητας	Link στο TINKERCAD
PROJECT 01	Εφέ Φωτιάς με LED	https://www.tinkercad.com/things/6uBTYfueE79?sharecode=ho5Pi3sdjiSr4zZ5Ay40TaaU5H8KP4qvkdsChrVvLMg
PROJECT 02	Λαμπάκι νυκτός με φωτοαντίσταση	https://www.tinkercad.com/things/0EWOifUEqd2?sharecode=hL5f8oEaFnoio0tMHFw2piuDATz5bi4aP0ZiPlodEPY





PROJECT 03	Κυνηγητό LED με ρυθμιστή	https://www.tinkercad.com/things/gOjSC05SjIN?sharecode=Jug78jxi_zBl_qUOf9wexlZMVAB3nQMhJVOZrtdRhgEQ
PROJECT 04	Μείκτης χρωμάτων με RGB LED	https://www.tinkercad.com/things/9tYgiVGx0ZE?sharecode=y1x4NmdxJQiPL6ldVmiSxNj5_ZA51qbkwS0vZq1PcwU
PROJECT 05	Αναπαράσταση Δυαδικού συστήματος	https://www.tinkercad.com/things/4bIPqHkbK0a?sharecode=nG1IA01EA7v5Cu5ymzXqzZ4zjat2-Js4B0mLW-13Nxl
PROJECT 06	Μετρητής πάνω-κάτω μονού ψηφίου	https://www.tinkercad.com/things/aPDdlvagX6e?sharecode=xVTeleZMBMjPgeAaA59PsXkX3CfmvUAsilrxWIZGjyY
PROJECT 07	Μουσική με Buzzer	https://www.tinkercad.com/things/1fYaSpkbsne?sharecode=bd4fem_ff90S003Ze41WfYFLBR8zM3AciZM4F266zFE
PROJECT 08	Πιάνο Buzzer με 7 νότες	https://www.tinkercad.com/things/hqP7G7zVg3K?sharecode=Pf4j0e82_yqixNdrzEDCeDG1tgjpn16ocPlfmGd6WjM
PROJECT 09	Βολτόμετρο με γραφικό μετρητή	https://www.tinkercad.com/things/4GANSww5vIU?sharecode=jfsHSjsMSuAcp1s7Zuwel6fstyn3Q8sicxd594W3W0o
PROJECT 10	Μετρητής χωρητικότητας Πυκνωτών	https://www.tinkercad.com/things/5itmT5xjEmW?sharecode=2rV5srWrVn26maeyskz0RM0b7JcMwXye11LGSAJ0SvE
PROJECT 11	Μηνύματα σε LCD οθόνη	https://www.tinkercad.com/things/kJvc4KJvrco?sharecode=frQiLGmDPIIqEEnUyESS2KqC7OWquUSNqWB0ffJ7FCc
PROJECT 12	Χρονοδιακόπτης ON/OFF με DC power supply	https://www.tinkercad.com/things/dOhiNvAUw5I?sharecode=i_LiH60Yw085VHOMNGkaTWOGrUXfgwrqG9yYV7nRk8A
PROJECT 13	Απεικόνιση απόστασης με τον SR04	https://www.tinkercad.com/things/IUxQI56dr8F?sharecode=1iVuZPhToR2XITkU7G8hShBCHxpPSORW1ZyFO-1UucE
PROJECT 14	Συναγερμός με αισθητήρα κίνησης	https://www.tinkercad.com/things/1atPgUzIJGk?sharecode=XBoiphEA_2oXlObcFCOs9xB91GpKS_9Peil3KVI8DwQ
PROJECT 15	Ανιχνευτής φωτιάς και καπνού	https://www.tinkercad.com/things/hj6i8BvmUjj?sharecode=1O_aA-l-BhRTspL5y_mfEo_w4dqaLJ-cbXZbnYMqAFg
PROJECT 16	Μετεωρολογικός σταθμός με LCD	https://www.tinkercad.com/things/a7yfBI1Wi5R?sharecode=Mvdr3KT9L3gaCOyR64Iq_hZxxbgtDts_y5qHE5zB6FY
PROJECT 17	Κλειδαριά με πληκτρολόγιο	https://www.tinkercad.com/things/0TGZFd8kvCt?sharecode=rWF40MbUlvXuK7BxtYYW93yGBF462EFsPkclCE2aaF4
PROJECT 18	Τηλεκοντρόλ με υπέρυθρες	https://www.tinkercad.com/things/4aiOcyW5whj?sharecode=0moB4v4xbv7Ugvfets3VrGbXOVXFEKIMokr9ufoHAGk





PROJECT 19	Έλεγχος βηματικού μοτέρ με εντολές	https://www.tinkercad.com/things/5mHrE0iZPnm?sharecode=Ep0MC6WFTQuFWz7gyInqxq7oY763YqXLRoZCerowxIU
PROJECT 20	Έλεγχος βηματικού μοτέρ με υπέρυθρες	https://www.tinkercad.com/things/82uzoxXdgeF?sharecode=crQneGBfzir86T4qVfUqOLCjirn0BpYHysJot2tQTpg

4. Βιβλιογραφία

- <https://create.arduino.cc/projecthub/wolkabout-writer/arduino-environmental-monitoring-a51e9d>
- <https://www.seeedstudio.com/blog/2019/07/19/arduino-environmental-monitoring-system-using-grove-sensors/>
- <https://alejandroquinteros.files.wordpress.com/2012/11/environmental-monitoring-with-arduino.pdf>
- <https://www.ni.com/getting-started/set-up-hardware/instrument-control/ethernet-connect>
- <https://knowledge.ni.com/KnowledgeArticleDetails?id=kA03q000000x3gXCAQ&l=en-GR>
- <https://www.allaboutcircuits.com/projects/using-an-arduino-as-a-web-server/>
- <https://pimylifeup.com/arduino-web-server/>
- <https://randomnerdtutorials.com/arduino-webserver-with-an-arduino-ethernet-shield/>
- <https://zugiduino.wordpress.com/2012/07/11/how-to-wire-and-test-your-arduino-ethernet-shield/>
- <https://www.arduino.cc/en/Tutorial/WebServer>
- <https://freeware.the-meiers.org/>





5. Ποιοι πήραν μέρος σε αυτή την εργασία

Για την υλοποίηση του σχεδίου δράσης πήραν μέρος οι παρακάτω μαθητές:

A) για τη σύνθεση των πινακίδων και την πιλοτική εφαρμογή των δραστηριοτήτων STEAM οι μαθητές της Α' τάξης:

Μπιμπίρης Παναγιώτης, Πιαστόπουλος Σωτήριος, Μουτάφης Ορέστης, Σπυρολλάρης Βασίλης, Χατζηχρήστου Δημήτριος, Σαρόγλου Σταύρος, Κωστανόγλου Κυριακή, Ξύντας Μιχάλης,

Αχπαράκης Σταύρος, Γάκης Μάριος, Κομίτης Αγγελος, Παντελάκου Ειρήνη, Πολυζούδης Χρήστος, Τσιφτελίδης Ορφέας, Σιμόπουλος Κωνσταντίνος-Ιάσων, Τσορλιανός Αλέξανδρος και Φιλιππίδης Σταύρος.

B) για την ανάπτυξη των δραστηριοτήτων STEAM στο TINKERCAD οι μαθητές της Γ' τάξης:

Καρέντζος Ευθύμιος, Κιλτσιδης Στυλιανός, Κοντοπαντελής Νικόλαος, Λύκου Μελίνα, Φωτιάδης Χαράλαμπος, Μακρής Δημήτριος, Βύζας Δημήτριος, Σαρηχανδάκης Αγγελος.

Η εργασία βασίστηκε σε μια ιδέα του εκπαιδευτικού **Μισαηλίδη Άνθιμου** και υλοποιήθηκε από τους εκπαιδευτικούς:

Μισαηλίδη Άνθιμο, ΠΕ86

Μαυρίδη Κωνσταντίνο, ΠΕ84,

Μπουλταδάκη Στυλιανό, ΠΕ84 και

Πανούσο Τρύφωνα ΔΕ0102

