

Προγραμματίζοντας με Arduino

1η δραστηριότητα: Διαδοχικό άναμμα των LEDs στις μονές και ζυγές θέσεις

Χρησιμοποιώντας κύκλωμα με 4 LEDs, γράψτε τον κώδικα προγράμματος που να επαναλαμβάνει συνεχώς τα ακόλουθα: πρώτα θα ανάβουν **ΟΛΑ ΜΑΖΙ τα LEDs που βρίσκονται στις ΜΟΝΕΣ** θέσεις με τα υπόλοιπα σβηστά και μετά θα ανάβουν όλα μαζί **ΟΛΑ ΜΑΖΙ τα LEDs που είναι στις ΖΥΓΕΣ** θέσεις με τα υπόλοιπα σβηστά.

Σύνδεσμος Tinkercad

<https://www.tinkercad.com/things/7AVRalsVaYP>

2η δραστηριότητα: Άναμμα 6 λυχνιών LEDs με διαδοχική σειρά

Φτιάξτε κύκλωμα με 6 LEDs στη σειρά (χρησιμοποιήστε όσα διαφορετικά χρώματα έχετε).

Γράψτε κώδικα ώστε να ανάβει κάθε LED με τη σειρά του. (Δηλαδή το ένα μετά το άλλο, σαν να αναπηδάει το φως από το ένα LED στο διπλανό του και πάλι από τη αρχή).

Στο πρόγραμμα θα πρέπει να χρησιμοποιηθεί ένας πίνακας 6 θέσεων που να περιέχει τα ανάλογα 6 pins που έχετε χρησιμοποιήσει για να συνδέσετε τα LEDs με το Arduino.

Σύνδεσμος tinkercad

https://www.tinkercad.com/things/gVgMSHm4QaF-copy-of-led-sequence-w-breadboard/editel?sharecode=opT-iVvFNp_4ADFcs8qeg4IzuJ5rb_jyMgCNfe_hLrU

3η δραστηριότητα: Ενδεικτική λυχνία LED με μεταβλητή φωτεινότητα

Σε κύκλωμα με ένα LED γράψτε πρόγραμμα που επαναλαμβάνει τα ακόλουθα:

Το LED αρχικά είναι αναμμένο. Σταδιακά μειώνεται η φωτεινότητα του μέχρι που σβήνει τελείως. Μένει σβηστό για 3 δευτερόλεπτα και στη συνέχεια κατευθείαν (και όχι σταδιακά) παίρνει τη μέγιστη φωτεινότητά του, τη μειώνει σταδιακά κτλ.

Η μεταβολή της φωτεινότητας θα πρέπει να γίνεται με βήμα 3.

Σύνδεσμος tinkercad

<https://www.tinkercad.com/things/41dH9WaNN8J>

4η δραστηριότητα: Οπτικός Κώδικας Mors με LED

Χρησιμοποιώντας κύκλωμα με ένα LED, γράψτε πρόγραμμα που να υλοποιεί τον οπτικό κώδικα Morse για τα 3 πρώτα γράμματα από το επώνυμό σας. Θα χρησιμοποιήσετε ένα συνδυασμό μακρών και σύντομων καθυστερήσεων ώστε να κάνετε το LED να αναβοσβήσει το επώνυμό σας.

Βάλτε μια μεγαλύτερη καθυστέρηση στο τέλος του μηνύματος και στη συνέχεια αφήστε τον κώδικα να επαναλαμβάνει το επώνυμό σας ξανά και ξανά.

Παρακάτω βλέπετε την αντιστοιχία κάθε γράμματος με τον κώδικα Morse.

Γράμμα	Κώδικας Morse	Γράμμα	Κώδικας Morse
A	· —	N	— ·
B	— · · ·	Ξ	— · · —
Γ	— — ·	O	— — —
Δ	— · ·	Π	· — — ·
E	·	P	· — ·
Z	— — · ·	Σ	· · ·
H	· · · ·	T	—

Θ	— · — ·	Υ	— · — —
Ι	· ·	Φ	· · — ·
Κ	— · —	Χ	— — — —
Λ	· — · ·	Ψ	· · —
Μ	— —	Ω	· — —

Βήματα και παραδοχές για την υλοποίηση της δραστηριότητας

- Τελεία (**dot**) θεωρούμε το «*μικρό χρόνο*» που παραμένει αναμμένο το LED. Ορίστε το χρόνο που θα αφήσετε αναμμένη την ενδεικτική λυχνία LED για την τελεία (**dot**). Αυτό είναι το μήκος μίας τελείας (**dot length**).
- Παύλα (**dash**) θεωρούμε τον «*μεγάλο χρόνο*» που παραμένει αναμμένο το LED. Το μήκος της παύλας (**dash length**) πρέπει να είναι 3 φορές το μήκος της τελείας.
- Ανάμεσα σε κάθε τελεία ή παύλα πρέπει να υπάρχει ένα διάκενο για να ξεχωρίζουμε το ένα στοιχείο από το άλλο (δηλαδή διάκριση μεταξύ τελείας και παύλας) με ένα μήκος χρόνου (όπου η ενδεικτική λυχνία θα είναι σβηστή).
- Ανάμεσα σε κάθε πλήρες γράμμα, (δηλαδή διάκριση μεταξύ γραμμάτων) πρέπει να υπάρχει κενό ίσο με το μήκος μίας παύλας (όπου η ενδεικτική λυχνία θα είναι σβηστή).
- Μεταξύ κάθε πλήρους λέξης, θα πρέπει να υπάρχει ένα κενό ίσο με το 7 φορές το μήκος της μίας τελείας (όπου η ενδεικτική λυχνία θα είναι σβηστή).

Για το πρόγραμμα αυτό πρέπει να χρησιμοποιηθούν συναρτήσεις.

Η παρακάτω συνάρτηση dot υλοποιεί την τελεία.

```
void dot() {
  digitalWrite(pin, HIGH);
  delay(400); // 400 θα μπορούσε να είναι ο χρόνος για την τελεία.
  digitalWrite(pin, LOW);
  delay(400);
}
```

Όμοια θα φτιάξετε τη συνάρτηση dash() για την παύλα, καθώς και συναρτήσεις για τα γράμματα που σας ενδιαφέρουν. Π.χ. κάποιος που λέγεται Χ. ΑΡΝΤΟΥΙΝΟΠΟΥΛΟΣ πρέπει να υλοποιήσει συναρτήσεις για το Α, το Ρ και το Ν.

Π.χ. η συνάρτηση για το Ω θα ήταν:

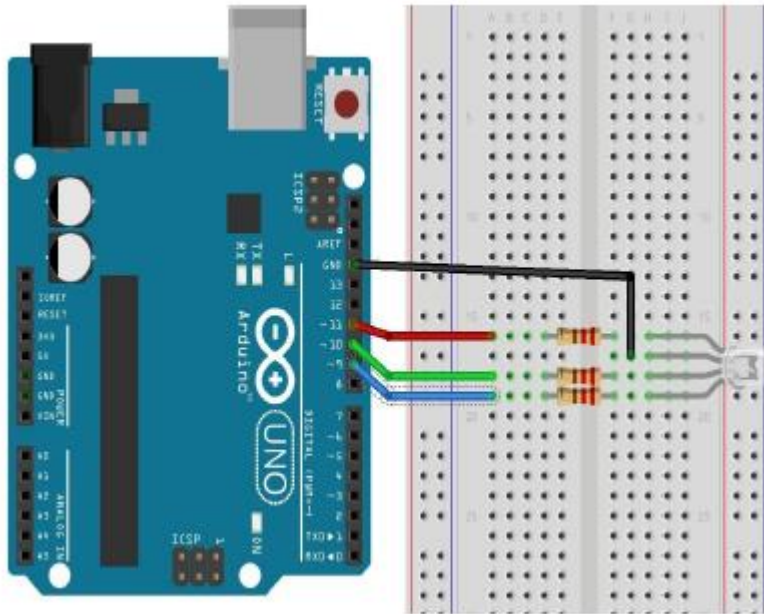
```
void omega () {
  dot();
  dash();
  dash();
}
```

Σύνδεσμος tinkercard

<https://www.tinkercad.com/things/f0fWfWfWypk>

5η δραστηριότητα: Παίζοντας με τα χρώματα (RGB Led)

Για την εργασία αυτή θα χρειαστούμε ένα RGB led και τρεις αντιστάσεις, μία για κάθε χρώμα. Στην εικόνα 26 βλέπετε τη συνδεσμολογία. Προσέξτε τις τιμές των αντιστάσεων, η τιμή για την κόκκινη είναι λίγο διαφορετική από αυτές για το πράσινο και μπλε. Επίσης, προσέξτε ποιο pin θα συνδέσετε με ποιο χρώμα. Το μακρύτερο (2ο) πόδι του led πάει στη γείωση (GND), το δίπλα του εξωτερικό (1ο) στο κόκκινο, το δίπλα του εσωτερικό (3ο) στο πράσινο και το τελευταίο εξωτερικό (4ο) στο μπλε. Επίσης, για όλα τα χρώματα θα χρειαστούμε PWM θύρες κι όχι απλές ψηφιακές.



Για την εργασία αυτή θα χρειαστεί να συνδέσουμε το led μας σε κάποια από τις PWM θύρες του

Arduino, οι οποίες δουλεύουν όπως έχουμε δει και ως αναλογικές έξοδοι.

Συγκεκριμένα:

1. Κάνετε όποιες αρχικοποιήσεις χρειάζονται (μεταβλητές, τιμές, θύρες).
2. Με μια επαναληπτική διαδικασία θα δίνετε σταδιακά τιμές σε κάθε led.

π.χ.

```
for(i=0;i<=255;i=i+5) {  
  analogWrite(redPin, i);  
  analogWrite(greenPin, i);  
  analogWrite(bluePin, i);  
  delay(200);  
}
```

3. Προσπαθήστε να παραλλάξετε τον κώδικά σας και να πετύχετε όλους τους δυνατούς συνδυασμούς χρωμάτων!

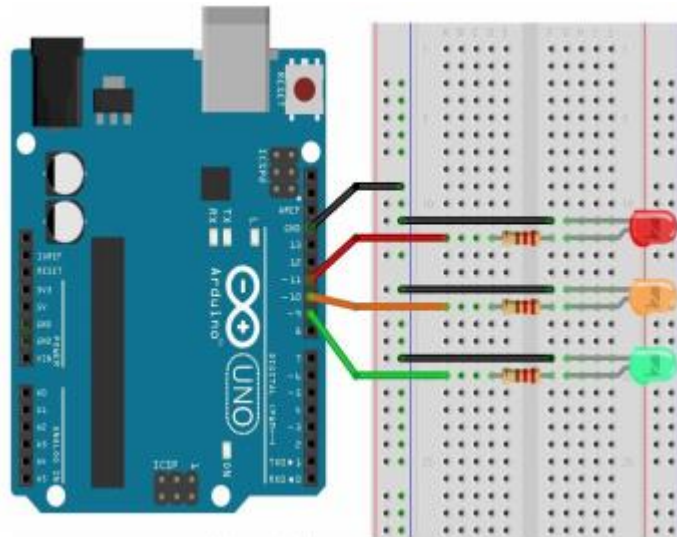
Σύνδεσμος Tinkercad

<https://www.tinkercad.com/things/0vIsspgxk8E-copy-of-rgb-analog/edite1?sharecode=0-5GWhDfYGq-lvLFvGTYjf-neRK9Mi2QLcNCYLjVUhM>

6η δραστηριότητα: Φανάρια Κυκλοφορίας

Στο φύλλο εργασίας αυτό θα προσομοιώσουμε τη λειτουργία των φαναριών. Έτσι, δίνοντας κατάλληλα χρονικά διαστήματα θα κρατήσουμε ανοιχτό το κάθε χρώμα για κάποιο χρόνο και ακολούθως η διαδικασία θα επαναλαμβάνεται από την αρχή. Τα φανάρια θα αναβοσβήνουν όπως στην Ελλάδα*, δηλαδή Πράσινο, Πορτοκαλί, Κόκκινο, Πράσινο, κ.ό.κ.

Περιγραφή κυκλώματος



Για την εργασία αυτή θα χρειαστούμε τρία led, ένα κόκκινο, ένα πράσινο κι ένα πορτοκαλί (ή κίτρινο) – ένα για κάθε φανάρι, όπως επίσης και τις αντίστοιχες αντιστάσεις. Μπορείτε να δείτε το κύκλωμα στην εικόνα 28 που ακολουθεί. Προσέξτε το γεγονός ότι επειδή δεν υπάρχουν πολλές υποδοχές GND, όλες βραχυκυκλώνονται πάνω στο breadboard (-) και εκεί συνδέουμε το GND του Arduino.

Προγραμματισμός κυκλώματος

Το μόνο που μένει είναι να προγραμματίσουμε το κύκλωμά μας. Δηλαδή, θα κάνετε τις απαραίτητες αρχικοποιήσεις και ακολούθως:

1. Θα πρέπει να δίνετε τάση 5V στο led που θέλετε να ανάβει και 0V στα υπόλοιπα.
2. Θα πρέπει να περιμένουμε 1 δευτερόλεπτο κατά τη διάρκεια του οποίου προφανώς ανάβει κάποιο led και είναι σβηστά τα υπόλοιπα.
3. Τα βήματα 1, 2 θα τα κάνετε εναλλάξ για το κόκκινο, πορτοκαλί και πράσινο Led.
4. Δοκιμάστε να προσαρμόσετε τους χρόνους. Το πορτοκαλί συνήθως ανάβει για πολύ λιγότερο χρόνο από ότι τα άλλα δύο χρώματα.

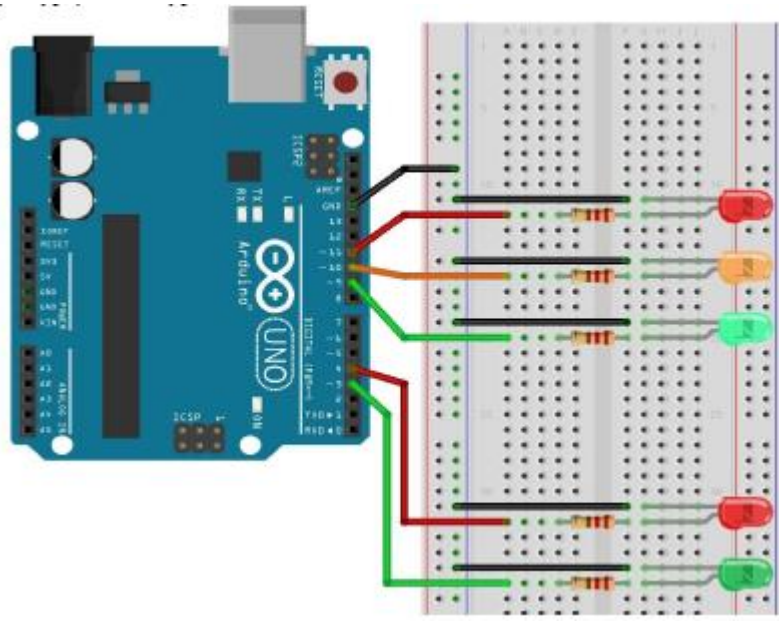
<https://www.tinkercad.com/things/cSdvKwgsZQI-copy-of-traffic-light/editel?sharecode=RxpPGYk1RdHI0weBoyTDoBWXKdI3SMBxYxFXZeG-44>

7η δραστηριότητα: Φανάρια Κυκλοφορίας με φανάρια πεζών

Στο φύλλο εργασίας αυτό θα εμπλουτίσουμε τη λειτουργία των φαναριών, όπως υλοποιήθηκε στο φύλλο εργασίας 8, με δύο ακόμα φανάρια για τους πεζούς. Έτσι, δίνοντας κατάλληλα χρονικά διαστήματα θα κρατήσουμε ανοιχτό το κάθε χρώμα κυκλοφορίας για κάποιο χρόνο, ενώ όταν υπάρχει κίνηση θα ανάβει το κόκκινο των πεζών, ενώ όταν υπάρχει κόκκινο στην κυκλοφορία θα ανάβει το πράσινο των πεζών. Ακολούθως η διαδικασία θα επαναλαμβάνεται από την αρχή.

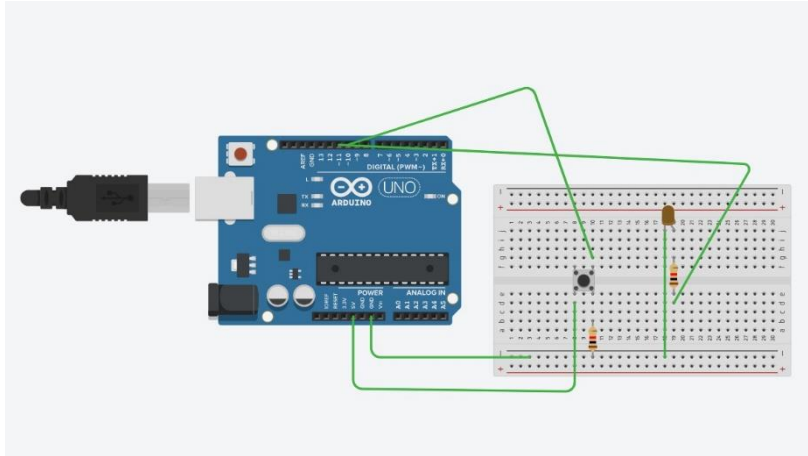
Περιγραφή κυκλώματος:

Για την εργασία αυτή θα χρειαστούμε, επιπλέον από αυτά του φύλλου εργασίας 8, δύο led, ένα κόκκινο κι ένα πράσινο για να προσομοιώσουμε τη λειτουργία των φαναριών των πεζών και τις αντίστοιχες αντιστάσεις.



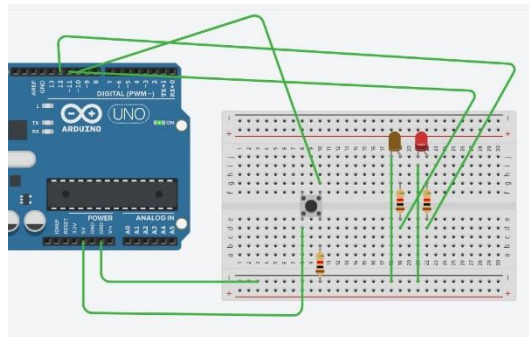
8η δραστηριότητα: Άσκηση με button

Σχεδιάστε το παρακάτω κύκλωμα στο Tinkercad και τρέξτε τον κώδικα που σας δίνεται. Τι κάνει το κύκλωμα?



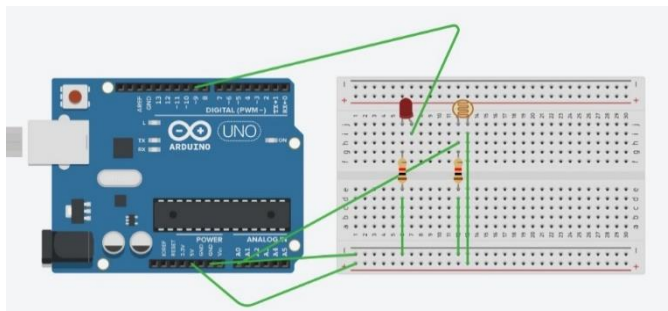
```
// C++ code
//
void setup()
{
  pinMode(11, OUTPUT);
  pinMode(10, INPUT);
}
void loop()
{
  digitalWrite(11, LOW);
  if (digitalRead(10) == HIGH)
  {
    digitalWrite(11, HIGH);
  }
}
```

1. Τροποποιήστε το παραπάνω πρόγραμμα έτσι ώστε το λαμπάκι να μένει αναμμένο για 2 δευτερόλεπτα όταν πατάμε το κουμπί.
2. Σχεδιάστε το παρακάτω κύκλωμα, και γράψτε κώδικα έτσι ώστε να είναι αρχικά αναμμένο το κόκκινο λαμπάκι και σβηστό το πορτοκαλί και όταν πατάμε το κουμπί να σβήνει το κόκκινο και να ανάβει το πορτοκαλί για 2 δευτερόλεπτα



9η δραστηριότητα: Άσκηση με αισθητήρα φωτός

Σχεδιάστε το παρακάτω κύκλωμα με το πρόγραμμα που σας δίνεται. Τι παρατηρείτε?



```
int sensorValue = 0;

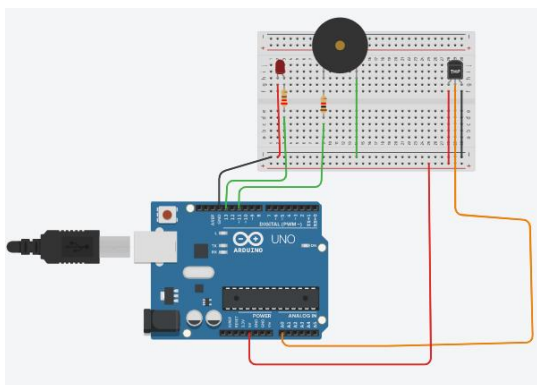
void setup()
{
  pinMode(A0, INPUT);
  Serial.begin(9600);
  pinMode(9, OUTPUT);
}

void loop()
{
  // read the value from the sensor
  sensorValue = analogRead(A0);
  // print the sensor reading so you know its range
  Serial.println(sensorValue);
  delay(100); // Wait for 100 millisecond(s)
}
```

1. Προσθέστε την εντολή `analogWrite(9, map(sensorValue, 0, 1023, 0, 255));`
Πριν την εντολή `delay(100);`
Τι παρατηρείτε?
2. Αλλάξτε τον παρακάτω κώδικα έτσι ώστε όταν η έξοδος της φωτοαντίστασης είναι πάνω από 300 να ανάβει το λαμπάκι αλλιώς να σβήνει.

10η δραστηριότητα: Αισθητήρας θερμότητας

Άσκηση 6: Σχεδιάστε το παρακάτω κύκλωμα και τρέξτε τον κώδικα που σας δίνεται. Τι παρατηρείτε?



```
int a;
void setup()
```

```
{  
  pinMode(A0, INPUT);  
  pinMode(11, OUTPUT);  
  Serial.begin(9600);  
  pinMode(13, OUTPUT);  
}
```

```
void loop()  
{  
  a=analogRead(A0);  
  Serial.println(a);  
}
```

Οι εντολές που μετατρέπουν την έξοδο του αισθητήρα σε βαθμούς κελσίου και Φαρενάιτ είναι η εξής

```
celsius = map(((analogRead(A0) - 20) * 3.04), 0, 1023, -40, 125);
```

```
fahrenheit = ((celsius * 9) / 5 + 32);
```


1. Τροποποιήστε το πρόγραμμα ώστε στην σειριακή έξοδο να εμφανίζει την θερμοκρασία ως εξής Celsius=23 Fahrenheit=73.4
2. Τροποποιήστε το πρόγραμμα έτσι ώστε όταν η θερμοκρασία είναι πάνω από 40 βαθμούς κελσίου να ανάβει το λαμπάκι και να χτυπάει το buzzer αλλιώς να σβήνει και να σταματάει.

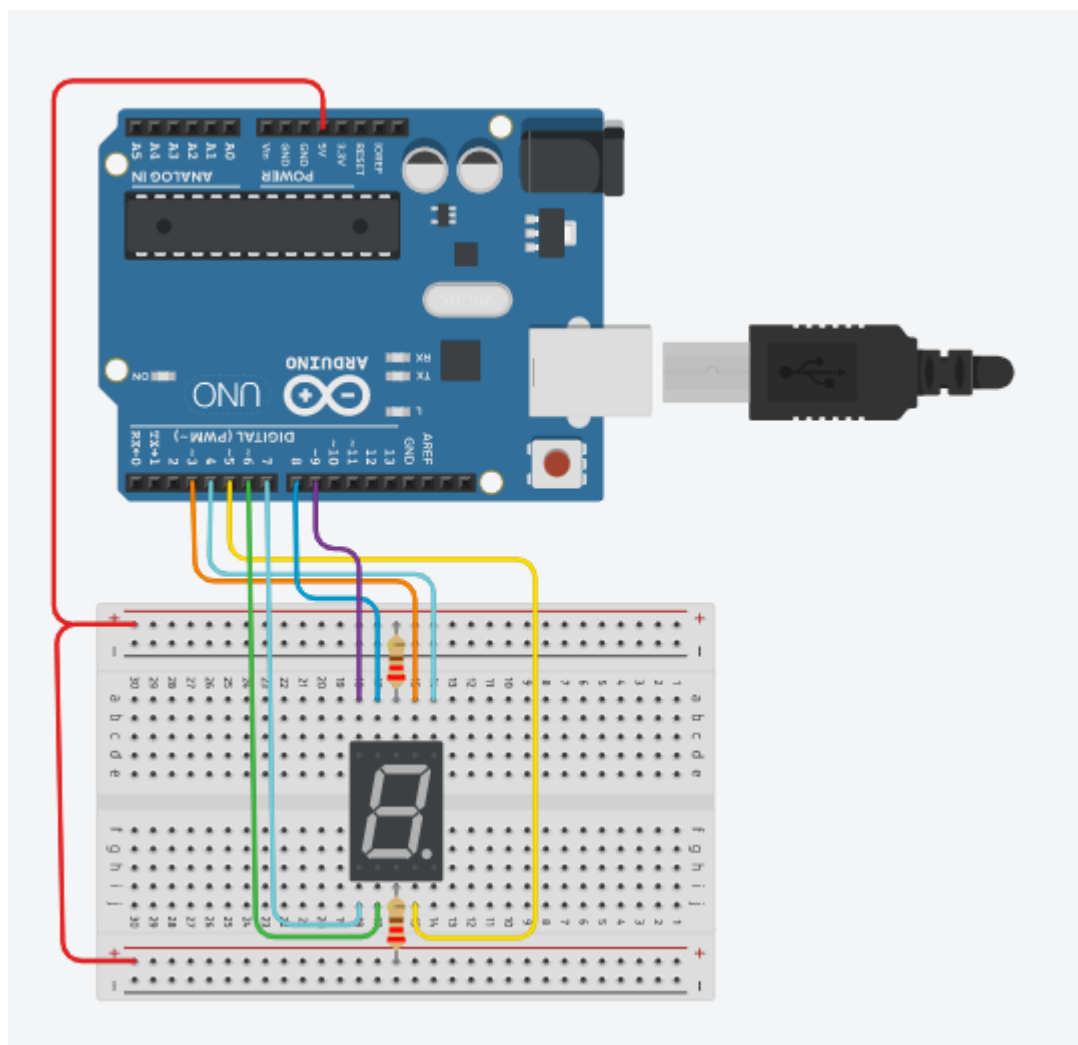
Σύνδεσμος Tinkercad

<https://www.tinkercad.com/things/7HX1IAIA3oW-copy-of-room-temperature-control-with-buzzer-and-led/editel?sharecode=oLDOi9PzLJz7SwwQ1RqmgTxRn46hEKneVGy9XwaqB5A>

11η δραστηριότητα: 7 Segment Display

Δίνεται το παρακάτω κύκλωμα το οποίο μετράει από το 0-9 εμφανίζοντας τους αριθμούς στο 7 Segment Display. Μπορείτε να βρείτε το κύκλωμα και τον κώδικα

https://www.tinkercad.com/things/5Clsc1jU7PF-segments-display-count-0-to-9/editel?sharecode=J0omhHN3x900_emKu7mnnHzCzwRc3mIx3tclsURoKPk

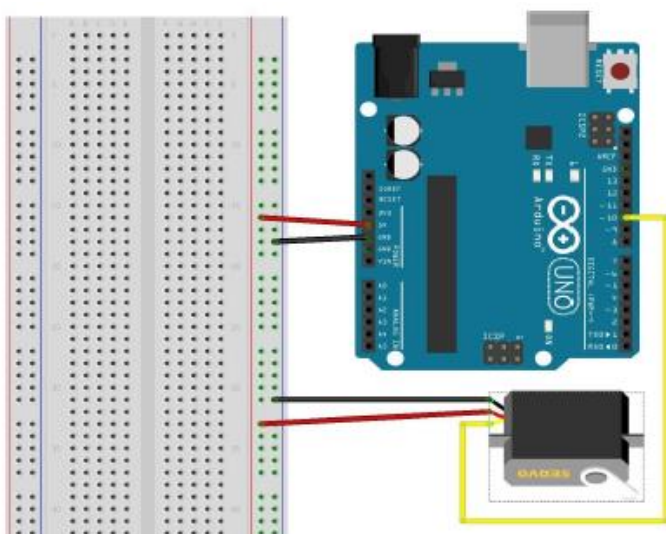


Τροποποιήστε το πρόγραμμα έτσι ώστε το κύκλωμα να εμφανίζει διαδοχικά τα γράμματα του ονόματός σας.

12η δραστηριότητα: Servo

Για την εργασία αυτή θα χρειαστούμε ένα σέρβο. Το σέρβο αποτελείται από τρία καλώδια: ένα για την πηγή (5V), ένα για τη γείωση κι ένα για τον έλεγχο του σέρβο μέσω ενός pin του Arduino. Τα χρώματα θα μας καθοδηγήσουν για τη συνδεσμολογία. Στο TowerPro SG90 έχουμε τρία καλώδια – ένα (πορτοκαλί) συνδέεται στην πηγή (5V), ένα (καφέ) στη γείωση (GND) κι ένα (κίτρινο) στο pin

ελέγχου του.



Για να μπορέσουμε να χρησιμοποιήσουμε ένα σέρβο, θα πρέπει να συμπεριλάβουμε στο πρόγραμμά μας την αντίστοιχη βιβλιοθήκη <Servo.h>. Επίσης, χρειάζεται να χρησιμοποιήσουμε συγκεκριμένες συναρτήσεις από τη βιβλιοθήκη αυτή. Αναζητήστε παρόμοιες πληροφορίες στο διαδίκτυο.

1. Δηλώστε μια μεταβλητή για το pin που συνδέσατε το servo
2. Συμπεριλάβετε στο πρόγραμμά σας την κατάλληλη βιβλιοθήκη για το σέρβο.
3. Δηλώστε μια μεταβλητή τύπου servo ώστε να αποκτήσετε πρόσβαση στο μηχανισμό του.
4. Κάνετε την κατάλληλη αρχικοποίηση για τη θύρα που χρησιμοποιείτε για το σέρβο. Για αυτή τη θύρα δεν δηλώνουμε στη συνάρτηση setup() ότι είναι έξοδος, αλλά την δεσμεύουμε με χρήση της εντολής attach(Pin).
5. Χρησιμοποιώντας της εντολή write() δώστε τις κατάλληλες εντολές στο servo σας, ώστε:
 - να ξεκινά στις 90 μοίρες
 - να στρίβει στις 180 μοίρες
 - να επιστρέφει στις 90 μοίρες
 - να στρίβει στις 0 μοίρες,λαμβάνοντας υπόψη και τους χρόνους που θα χρειαστούν για τις στροφές.
6. Τροποποιήστε τον κώδικα έτσι ώστε να περιστρέφεται στρίβοντας μια μοίρα την φορά και όταν ολοκληρώσει περιστροφή 180 μοιρών να επανέρχεται στην αρχική θέση στρίβοντας 10 μοίρες την φορά.

https://www.tinkercad.com/things/hGc5biQdLbM-copy-of-servo-moving/editel?sharecode=gqxl8d8_BQZ3F5TArnin9rAu3Z2l3Xqo1nWm8E-cNbU

7. Τροποποιήστε το κύκλωμα και προσθέστε ένα ποτενσιόμετρο. Το ποτενσιόμετρο θα καθορίζει την γωνία περιστροφής και θα την εμφανίζει στην σειριακή οθόνη.

<https://www.tinkercad.com/things/aWSZYBBWVXZ-copy-of-servopoten/editel?sharecode=lmx6T7KrRR2x9tMRhqlpiSgG5HdovBP-C7iBXzfjnFY>