

Σύστημα αυτόματου ποτίσματος τροφοδοτούμενου από δεξαμενή συλλογής νερού



Φύλλα εργασίας

2ο Γυμνάσιο Μοσχάτου

Σχολικό έτος 2022 - 23

Εισαγωγή

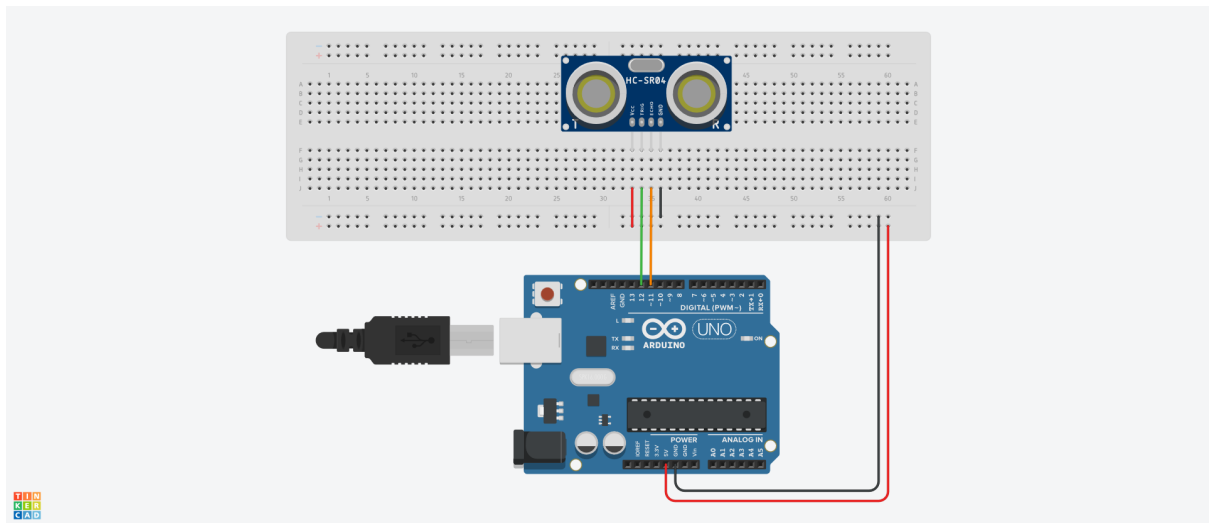
Τα φύλλα εργασίας που περιλαμβάνονται στις παρούσες σημειώσεις εξηγούν τη λειτουργία και τη χρήση των επιμέρους στοιχείων που χρησιμοποιούμε στο σύστημα αυτόματου ποτίσματος. Η εκτέλεση των δραστηριοτήτων μπορεί να γίνει εξ ολοκλήρου στο ελεύθερο λογισμικό TinkerCAD και δεν απαιτείται η αγορά εξοπλισμού.

Φύλλο εργασίας 1 - Μελέτη αισθητήρα υπερήχων

Ο αισθητήρας υπερήχων χρησιμοποιεί ήχο πάνω από την μέγιστη συχνότητα ακοής για να ανιχνεύσει απόσταση. Ο αισθητήρας εκπέμπει ένα σύντομο παλμό ήχου και περιμένει να επιστρέψει. Μετρά τον χρόνο που χρειάζεται για να ταξιδέψει ο ήχος, να ανακλαστεί σε ένα αντικείμενο και μετά να επιστρέψει στον αισθητήρα. Επειδή η ταχύτητα του ήχου είναι γνωστή, ο συνολικός χρόνος που απαιτείται για την επιστροφή του ήχου εξαρτάται από την απόσταση του αντικειμένου.

A. Μέτρηση απόστασης με τον αισθητήρα υπερήχων

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

```
const int echoPin=11;
const int trigPin=12;
long duration;
long distance;

void setup() {
  Serial.begin(9600);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
```

```

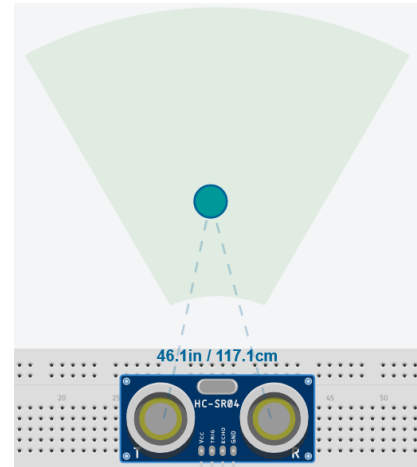
Serial.print(distance);
Serial.println(" cm");
delay(1000);
}

```

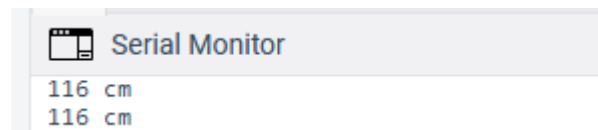
Το πρόγραμμα διαβάζει την απόσταση ενός εμποδίου από τον αισθητήρα υπερήχων και καταγράφει την ένδειξη σε εκατοστά στη σειριακή.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD.

Επιλέγουμε τον αισθητήρα υπερήχων στον χώρο σχεδίασης και μεταβάλλουμε τη θέση του αντικειμένου:

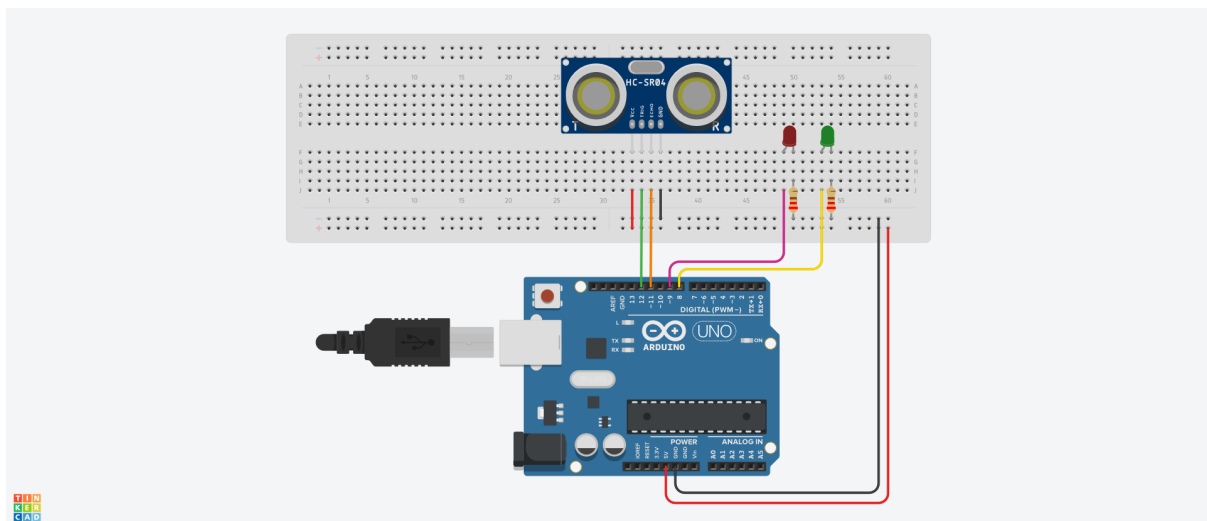


Παρατηρούμε την ένδειξη στη σειριακή:



B. Λήψη απόφασης ανάλογα με την απόσταση

Τροποποιούμε το προηγούμενο κύκλωμα στο TINKERCAD, όπως φαίνεται παρακάτω:



Τροποποιούμε το πρόγραμμα όπως φαίνεται παρακάτω:

```

const int ledGreen=8;
const int ledRed=9;
const int echoPin=11;
const int trigPin=12;
long duration;
long distance;

void setup() {
  Serial.begin(9600);
  pinMode(ledGreen, OUTPUT);
  pinMode(ledRed, OUTPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;

  if (distance<=30){
    digitalWrite(ledRed, HIGH);
    digitalWrite(ledGreen, LOW);
    delay(100);
  }
  else{
    digitalWrite(ledGreen, HIGH);
    digitalWrite(ledRed, LOW);
    delay(100);
  }

  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);
}

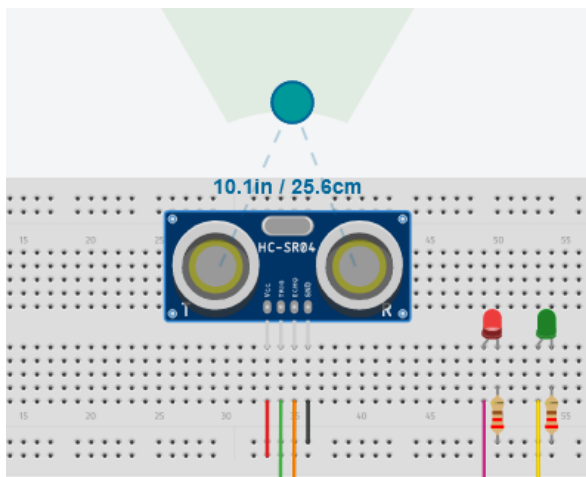
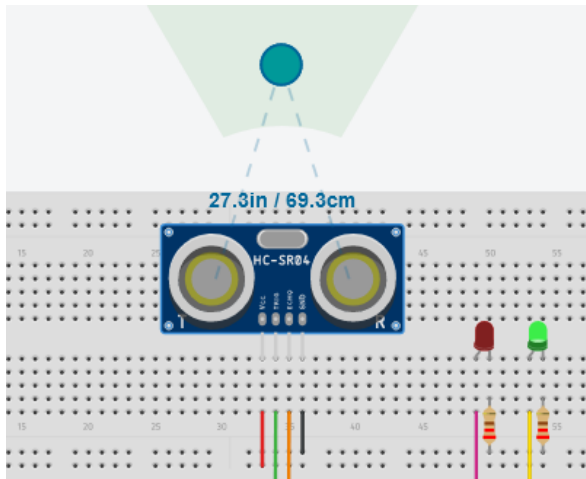
```

Το πρόγραμμα διαβάζει την απόσταση ενός εμποδίου από τον αισθητήρα υπερήχων και ανάβει το κόκκινο LED αν η απόσταση είναι μικρότερη από 30 cm, ενώ ανάβει το πράσινο LED αν είναι μεγαλύτερη. Η απόσταση καταγράφεται σε εκατοστά και στη σειριακή.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD.

Επιλέγουμε τον αισθητήρα υπερήχων στον χώρο σχεδίασης και μεταβάλλουμε τη θέση του αντικειμένου.

Ελέγχουμε τη σωστή εκτέλεση του προγράμματος παρατηρώντας τις ενδείξεις των LEDs και της σειριακής.

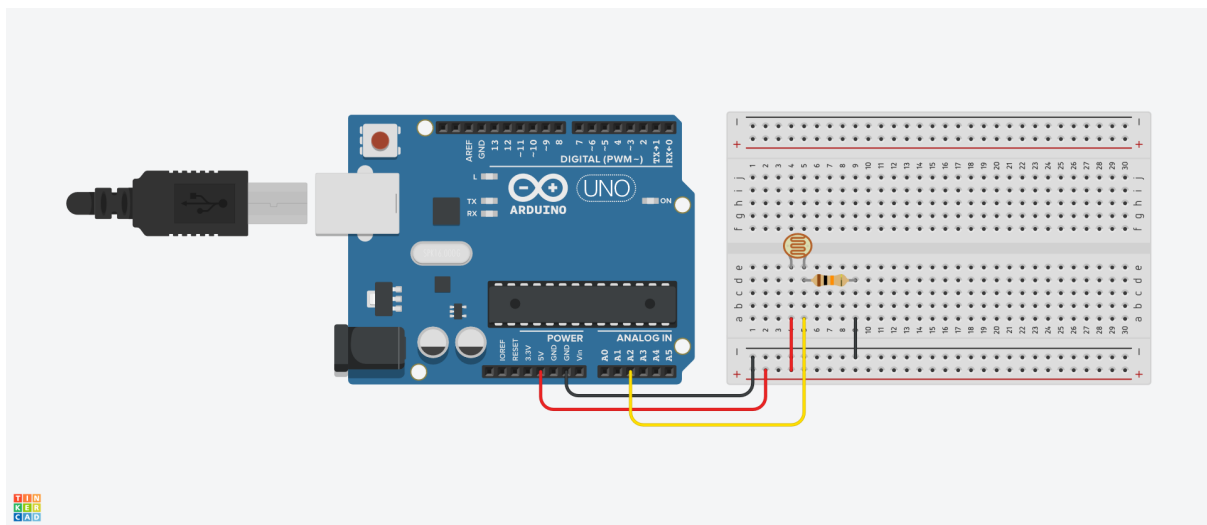


Φύλλο εργασίας 2 - Μέτρηση αναλογικών μεγεθών

Για να μετρήσουμε αναλογικά μεγέθη με το Arduino Uno χρησιμοποιούμε τις αναλογικές θύρες A0 μέχρι A5. Οι αναλογικές θύρες είναι συνδεδεμένες, μέσω ενός πολυπλέκτη, σε ένα μετατροπέα αναλογικού σε ψηφιακό (ADC). Ο μετατροπέας αναλογικού σε ψηφιακό μπορεί να μετατρέψει μια τιμή αναλογικού σήματος σε ψηφιακή αναπαράσταση. Στο Arduino Uno ο μετατροπέας αναλογικού σε ψηφιακό έχει ανάλυση 10-bit. Αυτό σημαίνει ότι μια τάση μεταξύ 0 και 5V σε μια αναλογική θύρα του Arduino Uno θα αντιπροσωπεύεται από μια αντίστοιχη τιμή μεταξύ 0 και 1023. Το αποτέλεσμα μιας αναλογικο-ψηφιακής μετατροπής σε κάποια αναλογική είσοδο του Arduino μπορούμε να "διαβάσουμε" με την εντολή `analogRead()`.

A. Ανάγνωση αναλογικής εισόδου

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

```
const int lightPin=A2;
int light;

void setup() {
  Serial.begin(9600);
}

void loop() {
  light = analogRead(lightPin);
  delay(100);

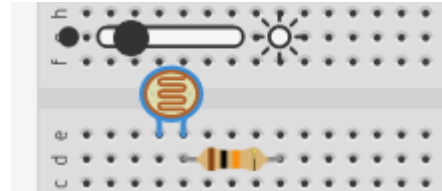
  Serial.print("Light: ");
  Serial.println(light);
}
```

Στο κύκλωμα χρησιμοποιούμε μια φωτοαντίσταση σε συνδεσμολογία διαιρέτη τάσης με μια αντίσταση σταθερής τιμής 10kΩ. Το πρόγραμμα διαβάζει την τιμή της τάσης στα άκρα της

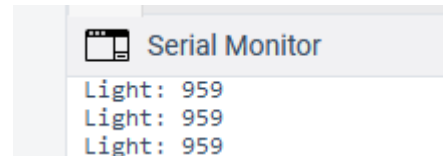
αντίστασης των 10kΩ και καταγράφει την ένδειξη του μετατροπέα αναλογικού σε ψηφιακό στη σειριακή.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD.

Επιλέγουμε τη φωτοαντίσταση στον χώρο σχεδίασης και μεταβάλλουμε τη θέση του δρομέα:

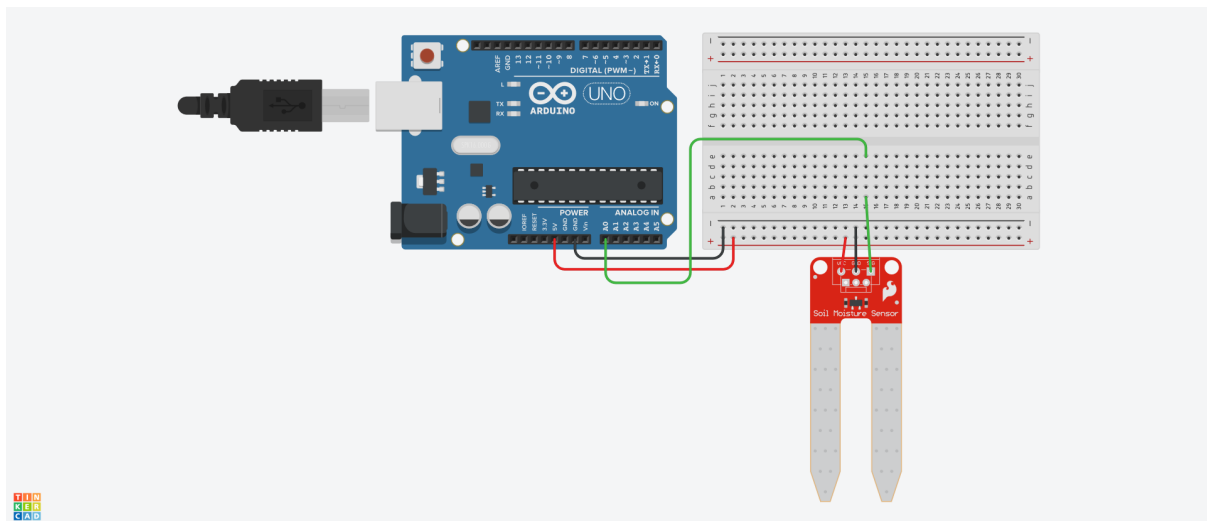


Παρατηρούμε την ένδειξη στη σειριακή:



B. Αναγωγή αναλογικής εισόδου σε ποσοστό %

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

```
const int hygro0Pin=A0;  
int hygro0;
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  hygro0 = analogRead(hygro0Pin);  
  hygro0 = constrain(hygro0,400,1023);  
  hygro0 = map(hygro0,400,1023,100,0);  
  delay(100);
```



```

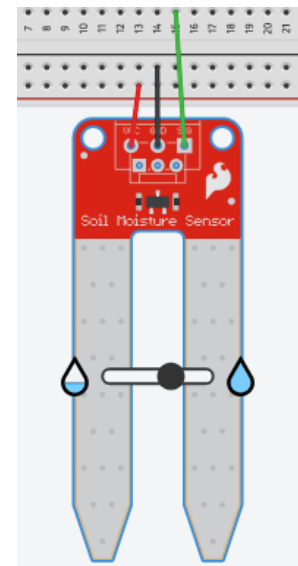
Serial.print("Soil humidity 0: ");
Serial.print(hygro0);
Serial.println("%");
}

```

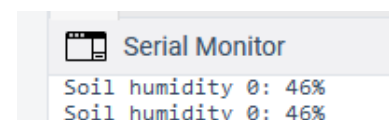
Τα δύο άκρα του αισθητήρα υγρασίας εδάφους λειτουργούν ως μεταβλητή αντίσταση. Περισσότερο νερό στο έδαφος σημαίνει καλύτερη αγωγιμότητα, επομένως μικρότερη αντίσταση. Όταν η υγρασία είναι μέγιστη, ο αισθητήρας δίνει μια τιμή μεταξύ 300 και 400. Με την εντολή constrain() θέτουμε το ελάχιστο όριο της μέτρησης, ενώ με την εντολή map() εκφράζουμε τις μετρήσεις σε ποσοστό με την μικρότερη μέτρηση να αντιστοιχεί στο μέγιστο ποσοστό υγρασίας. Το ποσοστό της υγρασίας καταγράφεται στη σειριακή.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD.

Επιλέγουμε τον αισθητήρα υγρασίας στο χώρο σχεδίασης και μεταβάλλουμε τη θέση του δρομέα:



Παρατηρούμε την ένδειξη στη σειριακή:



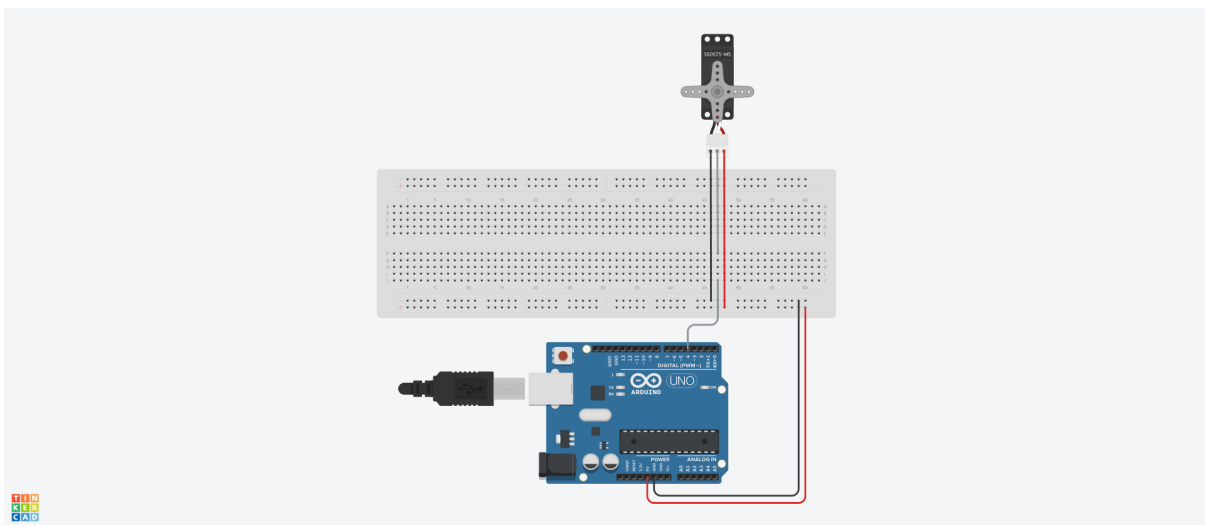
Φύλλο εργασίας 3 - Χειρισμός servo

Ένα servo είναι μια διάταξη που αποτελείται από ένα μικρό ηλεκτρικό κινητήρα, ένα συνδυασμό γραναζιών και ένα σύστημα για τον ακριβή έλεγχο της θέσης του άξονά του. Ο άξονας μπορεί να περιστρέφεται με μικρή ταχύτητα αλλά με μεγάλη ροπή με το εύρος περιστροφής να περιορίζεται μεταξύ 0 και 180 μοιρών (μισή περιστροφή).

Για να δουλέψει ένα servo, πρέπει να εισαχθεί η αντίστοιχη βιβλιοθήκη στο πρόγραμμα (`#include <Servo.h>`), να οριστεί μια μεταβλητή τύπου servo (`servo myservo;`) και ο ακροδέκτης μέσω του οποίου θα το χειριστούμε (`int servoPin = 4;`). Στη συνάρτηση `setup()` πρέπει να συσχετίσουμε τον ακροδέκτη με το σερβοκινητήρα (`myservo.attach(servoPin);`).

A. Περιστροφή του servo

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

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

```
Servo servo2;  
int servo2Pos = 0;  
const int servo2Pin=4;
```

```
void setup() {  
  Serial.begin(9600);  
  servo2.attach(servo2Pin);  
}
```

```
void loop() {  
  for (servo2Pos = 0; servo2Pos <= 180; servo2Pos += 30) {  
    servo2.write(servo2Pos);  
    Serial.println(servo2Pos);  
    delay(500);  
  }  
  for (servo2Pos = 180; servo2Pos >= 0; servo2Pos -= 30) {
```

```

servo2.write(servo2Pos);
Serial.println(servo2Pos);
delay(500);
}
}

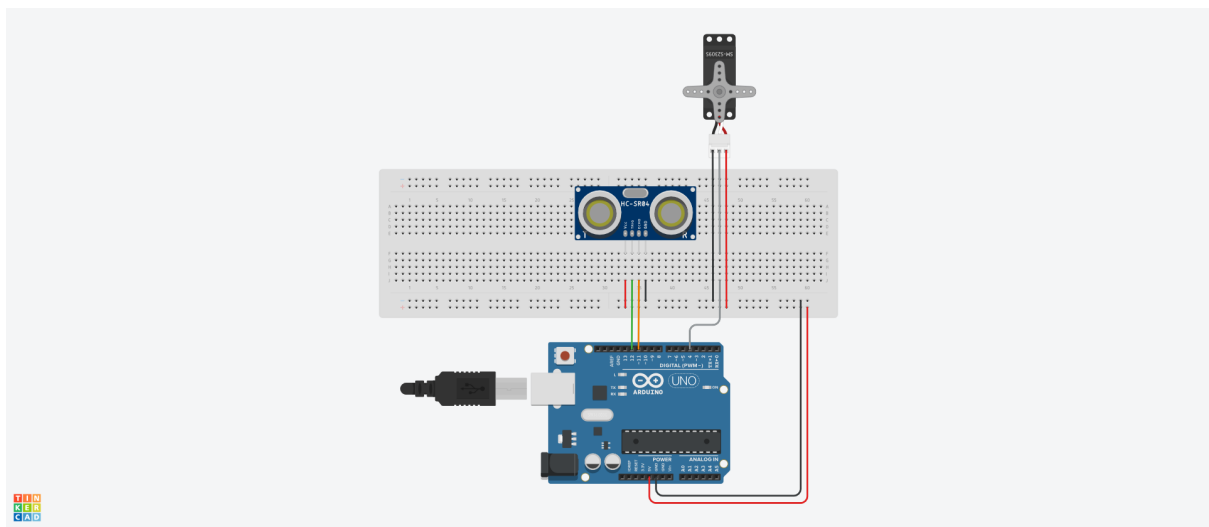
```

Το πρόγραμμα περιστρέφει τον άξονα του servo από 0 μέχρι 180 μοίρες και αντίστροφα σε βήματα των 30 μοιρών.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD και παρατηρούμε την περιστροφή του άξονα.

B. Χειρισμός του servo με τον αισθητήρα υπερήχων.

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

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

```
Servo servo2;
int servo2Pos = 0;
```

```
const int servo2Pin=4;
const int echoPin=11;
const int trigPin=12;
long duration;
long distance;
```

```
const int minDistance=30;
```

```
void setup() {
  Serial.begin(9600);
  servo2.attach(servo2Pin);
  servo2.write(servo2Pos);
  pinMode(trigPin, OUTPUT);
}
```

```

    pinMode(echoPin, INPUT);
}

void loop() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    duration = pulseIn(echoPin, HIGH);
    distance = (duration/2) / 29.1;

    if (distance <= minDistance){
        servo2Pos=90;
        servo2.write(servo2Pos);
        delay(100);
    }
    else{
        servo2Pos=0;
        servo2.write(servo2Pos);
        delay(100);
    }

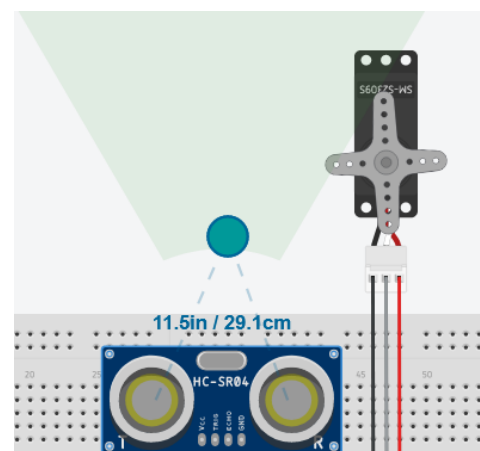
    Serial.print(distance);
    Serial.println(" cm");
    delay(1000);
}

```

Το πρόγραμμα περιστρέφει τον άξονα του servo από τις 0 στις 90 μοίρες, όταν η απόσταση που ανιχνεύει ο αισθητήρας υπερήχων είναι μικρότερη από 30 cm.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD.

Επιλέγουμε τον αισθητήρα υπερήχων στο χώρο σχεδίασης και μεταβάλλουμε τη θέση του αντικειμένου. Ελέγχουμε τη σωστή εκτέλεση του προγράμματος παρατηρώντας την περιστροφή του άξονα του servo.



Φύλλο εργασίας 4 - Χειρισμός κινητήρων με τη βοήθεια ρελέ

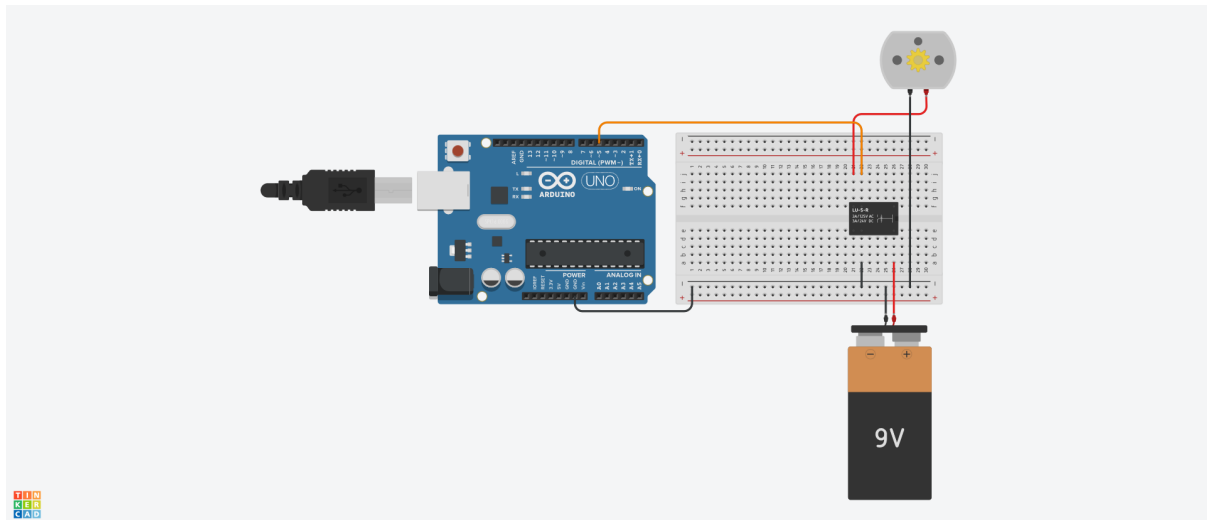
Ο ηλεκτρονόμος ή ρελέ (relay) είναι μια ηλεκτρομηχανική διάταξη που λειτουργεί ως διακόπτης ελεγχόμενος από τάση.

Στην απλούστερη μορφή του πρόκειται για ένα διακόπτη που ενεργοποιείται με τη βοήθεια ενός ηλεκτρομαγνήτη. Όταν ο ηλεκτρομαγνήτης δεν διαρρέεται από ρεύμα είναι βραχυκυκλωμένες οι επαφές C (COMMON) και NC (Normal Closed) του διακόπτη, ενώ όταν ο ηλεκτρομαγνήτης διαρρέεται από ρεύμα βραχυκυκλώνονται οι επαφές C και NO (Normal Opened). Αφήνοντας την επαφή NC χωρίς σύνδεση, μπορούμε να συνδέσουμε το ρελέ ως απλό (on-off) διακόπτη σε ένα ηλεκτρικό κύκλωμα, που μπορεί να λειτουργήσει σε τάσεις μέχρι 250 V AC και ρεύματα μέχρι και 10 A.

Σε μονάδες (module) ρελέ η ενεργοποίηση του ηλεκτρομαγνήτη γίνεται μέσω μιας ακίδας του (Signal ή Input), που συνδέεται σε μια ψηφιακή έξοδο του Arduino. Λογικό 1 στην ακίδα Signal ενεργοποιεί τον ηλεκτρομαγνήτη, και λογικό μηδέν τον απενεργοποιεί.

A. Ενεργοποίηση και απενεργοποίηση κινητήρα με τη βοήθεια ρελέ

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

```
const int relay0Pin=5;

void setup() {
  pinMode(relay0Pin, OUTPUT);
}

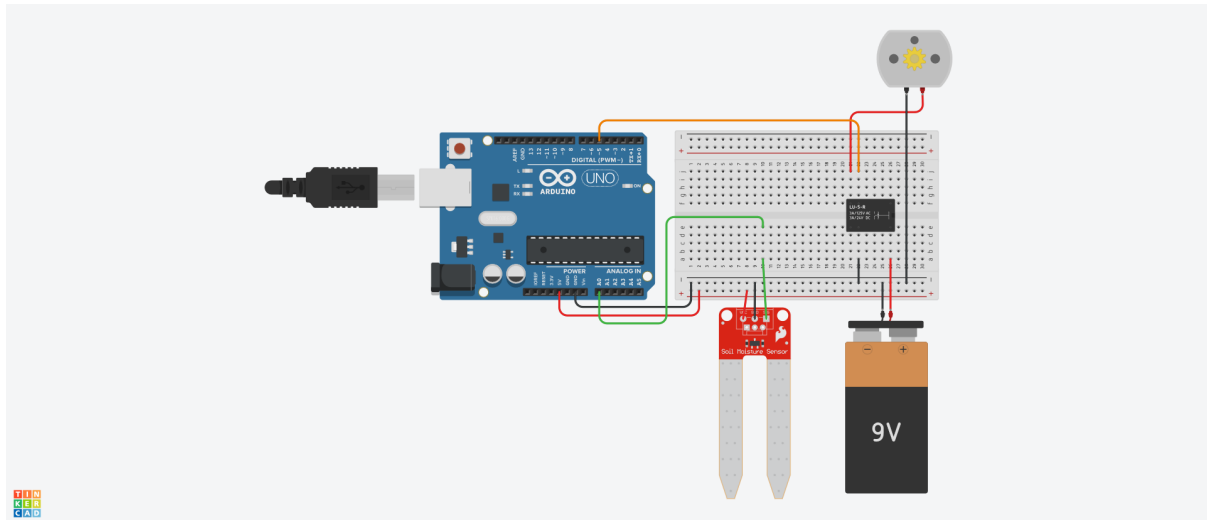
void loop() {
  digitalWrite(relay0Pin, HIGH);
  delay(3000);
  digitalWrite(relay0Pin, LOW);
  delay(3000);
}
```

Το πρόγραμμα ενεργοποιεί τον κινητήρα για 3 δευτερόλεπτα και στη συνέχεια τον απενεργοποιεί για άλλα 3.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD και παρατηρούμε τη λειτουργία του κινητήρα.

B. Ενεργοποίηση κινητήρα με τον αισθητήρα υγρασίας

Σχεδιάζουμε το παρακάτω κύκλωμα στο TINKERCAD:



Εισάγουμε τον παρακάτω κώδικα:

```
const int relay0Pin=5;
const int hygro0Pin=A0;
int hygro0;

void setup() {
  Serial.begin(9600);
  pinMode(relay0Pin, OUTPUT);
}

void loop() {
  hygro0 = analogRead(hygro0Pin);
  hygro0 = constrain(hygro0,400,1023);
  hygro0 = map(hygro0,400,1023,0,100);
  delay(100);

  if (hygro0<30){
    digitalWrite(relay0Pin, HIGH);
    delay(100);
  }

  if (hygro0>=70){
    digitalWrite(relay0Pin, LOW);
    delay(100);
  }
}
```

```

Serial.print("Soil humidity 0: ");
Serial.print(hygro0);
Serial.println("%");
}

```

Το πρόγραμμα ενεργοποιεί τον κινητήρα όταν η υγρασία του εδάφους είναι μικρότερη από 30%. Όταν η υγρασία γίνει μεγαλύτερη από 70%, ο κινητήρας απενεργοποιείται.

Ξεκινάμε την προσομοίωση του κυκλώματος στο TinkerCAD.

Επιλέγουμε τον αισθητήρα ανίχνευσης της υγρασίας του εδάφους στο χώρο σχεδίασης και μεταβάλλουμε τη θέση του δρομέα. Ελέγχουμε τη σωστή εκτέλεση του προγράμματος παρατηρώντας την ενεργοποίηση και την απενεργοποίηση του κινητήρα.

