

Η κατασκευή ενός μοντέλου που προσομοιώνει τη λειτουργία ενός φράγματος όπως το MOSE της Βενετίας

Το σύστημα MOSE χρησιμοποιεί φουσκωτούς πλωτήρες για την ανύψωση φραγμάτων που προστατεύουν τη Βενετία από την παλίρροια.

Υλικά που θα χρειαστείς

1. Arduino Uno (ή συμβατή πλακέτα)
2. Αισθητήρας υπερήχων (HC-SR04) για μέτρηση της στάθμης του νερού.
3. Σερβομηχανισμοί (Servo Motors) για την ανύψωση των "φραγμάτων".
4. Μικρή αντλία νερού (προαιρετική, για προσομοίωση της πλήρωσης).
5. LEDs (για ενδείξεις στάθμης νερού).
6. Πλαστικά ή ξύλινα υλικά για τη δημιουργία του φράγματος.
7. Προγραμματιζόμενος τροφοδότης αέρα ή μπαλόνια για να προσομοιωθεί η άνωση (προαιρετικό).
8. Κατασκευαστική βάση (όπως ένα διαφανές κουτί για τη "δεξαμενή νερού").

Λειτουργικότητα του συστήματος

1. Ανίχνευση στάθμης νερού:

- Ο αισθητήρας υπερήχων μετρά την απόσταση από την επιφάνεια του νερού.
- Αν η στάθμη ξεπεράσει ένα καθορισμένο όριο, το Arduino ενεργοποιεί το φράγμα.

2. Ανύψωση φραγμάτων:

- Οι σερβομηχανισμοί ανυψώνουν τα "φράγματα" από την επιφάνεια της δεξαμενής για να σταματήσουν την υποτιθέμενη εισροή νερού.

3. Οπτική ειδοποίηση:

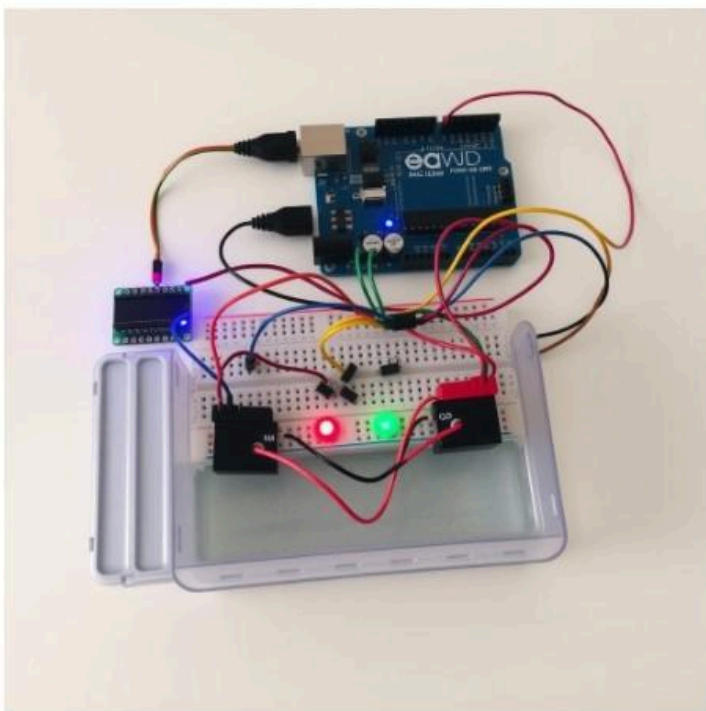
- Τα LEDs αλλάζουν χρώμα ανάλογα με τη στάθμη του νερού (π.χ., πράσινο για φυσιολογική, κόκκινο για υψηλή στάθμη).

4. Προαιρετική αντλία ή φούσκωμα μπαλονιών:

- Μια μικρή αντλία μπορεί να γεμίζει το "σύστημα φραγμάτων" για να δείξει πώς γεμίζουν τα MOSE με αέρα.

Διάγραμμα Σύνδεσης

- Αισθητήρας HC-SR04:
 - Trig -> Pin 7
 - Echo -> Pin 6
 - Vcc και GND ΣΤΟ Arduino.
- Servo Motors:
 - Σήμα -> Pins 9 και 10
 - Vcc και GND ΣΤΟ Arduino ή σε εξωτερική παροχή.
- LEDs:
 - Συνδέονται σε Pins 11-13 μέσω αντιστάσεων.
- Αντλία νερού (προαιρετική):
 - Συνδέεται μέσω τρανζίστορ ή ρελέ.



An Arduino Uno circuit simulating a flood barrier system. It includes an ultrasonic sensor (HC-SR04) connected to the Arduino for water level detection, two servo motors operating small flood barriers, three LEDs (green, yellow, red) with resistors on a breadboard for status indication, and all components connected with organized colored wires. The setup features a small water tank to illustrate barrier functionality, placed on a clean workspace with a white or neutral background.

Κώδικας Arduino

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

```
Servo barrier1;
```

```
Servo barrier2;
```

```
const int trigPin = 7;
```

```
const int echoPin = 6;
```

```
const int ledGreen = 11;
```

```
const int ledYellow = 12;
```

```
const int ledRed = 13;
```

```
void setup() {
```

```
  pinMode(trigPin, OUTPUT);
```

```
  pinMode(echoPin, INPUT);
```

```
  pinMode(ledGreen, OUTPUT);
```

```
  pinMode(ledYellow, OUTPUT);
```

```
  pinMode(ledRed, OUTPUT);
```

```
  barrier1.attach(9);
```

```
  barrier2.attach(10);
```

```
  barrier1.write(0); // Φράγμα κατεβασμένο
```

```
  barrier2.write(0); // Φράγμα κατεβασμένο
```

```
}
```

```
void loop() {
```

```
  long duration, distance;
```

```
  digitalWrite(trigPin, LOW);
```

```
  delayMicroseconds(2);
```

```
  digitalWrite(trigPin, HIGH);
```

```
  delayMicroseconds(10);
```

```
digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);
distance = duration * 0.034 / 2; // Μετατροπή σε εκατοστά

if (distance < 10) {
    // Υψηλή στάθμη - ενεργοποίηση φράγματος
    digitalWrite(ledRed, HIGH);
    digitalWrite(ledGreen, LOW);
    digitalWrite(ledYellow, LOW);
    barrier1.write(90); // Ανύψωση φραγμάτων
    barrier2.write(90);
} else if (distance < 20) {
    // Μέτρια στάθμη - προειδοποίηση
    digitalWrite(ledYellow, HIGH);
    digitalWrite(ledGreen, LOW);
    digitalWrite(ledRed, LOW);
} else {
    // Φυσιολογική στάθμη
    digitalWrite(ledGreen, HIGH);
    digitalWrite(ledYellow, LOW);
    digitalWrite(ledRed, LOW);
    barrier1.write(0); // Κατέβασμα φραγμάτων
    barrier2.write(0);
}

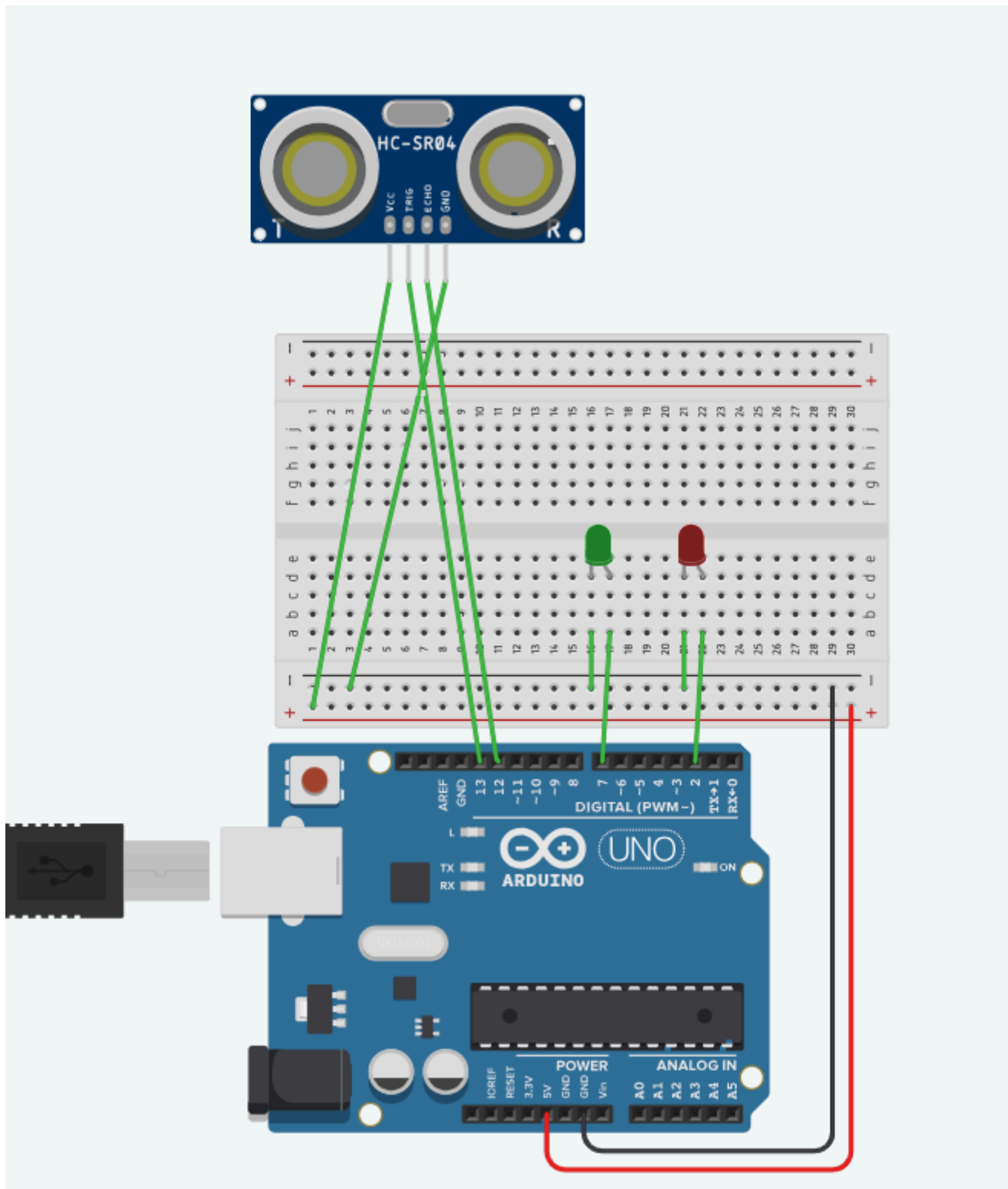
delay(500);
}
...
```

Πειραματισμοί

- Προσάρμοσε τον κώδικα για περισσότερα φράγματα.

- Πρόσθεσε οθόνη LCD για εμφάνιση των δεδομένων.
- Χρησιμοποίησε αντλία ή μπαλόνια για πιο ρεαλιστική προσομοίωση.

Παράδειγμα 1



```
int trigPin = 13; // Trigger Pin
```

```
int echoPin = 12; // Echo Pin
```

```
int maximumRange = 200; // Maximum range needed

int minimumRange = 0; // Minimum range needed


long duration, distance; // Duration used to calculate distance


void setup() {

  pinMode(trigPin, OUTPUT);

  pinMode(echoPin, INPUT);

  pinMode(7, OUTPUT);


  pinMode(2, OUTPUT);

  Serial.begin(9600);

}


void loop() {


  digitalWrite(trigPin, LOW);

  delayMicroseconds(2);


  digitalWrite(trigPin, HIGH);

  delayMicroseconds(10);


  digitalWrite(trigPin, LOW);


  duration = pulseIn(echoPin, HIGH);


  //Υπολογισμός απόστασης (σε cm) βασιζόμενοι στην ταχύτητα του ήχου.
  distance = duration/58.2;


  if (distance >= maximumRange || distance <= minimumRange){

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

```
delay(100);  
  
}  
  
else {  
  
Serial.println(distance);  
  
delay(50);  
  
};
```

```
if (distance<=100)  
{digitalWrite(7, HIGH);  
  
}  
  
else  
  
{ digitalWrite(7, LOW);  
  
}
```

```
if (distance<=50)  
{digitalWrite(2, HIGH);  
  
}  
  
else  
  
{ digitalWrite(2, LOW);  
  
}
```

```
delay(50);  
  
}
```