

5^ο Τρίωρο

Arduino Αισθητήρες και Περιφερειακά

Το Σειριακό Μόνιτορ (Serial Monitor) του Arduino

Το σειριακό μόνιτορ ή σειριακή οθόνη παρακολούθησης είναι μια επιπλέον δυνατότητα του Arduino IDE. Ανοίγει με το κουμπί που υπάρχει επάνω δεξιά στο IDE. Σε αυτό μπορούμε να λάβουμε και να δούμε γράμματα (χαρακτήρες) μέσω της θύρας USB (από το Arduino στον υπολογιστή), όπως επίσης και να στείλουμε γράμματα μέσω της θύρας USB (από τον υπολογιστή στο Arduino). Ένα απλό πρόγραμμα θα μας εξοικειώσει με τη χρήση του. Το πρόγραμμα απλά εκτυπώνει το μήνυμα Hello World ! στο σειριακό μόνιτορ και κάποιο άλλο μήνυμα που θέλουμε.

Οι εντολές που χρειαζόμαστε για να στείλουμε κάτι από το Arduino και να απεικονιστεί στο σειριακό μόνιτορ είναι βασικά δύο:

- `Serial.print(...)` ; Αυτή τυπώνει (ότι βάλουμε στην παρένθεση) αμέσως δίπλα σε ότι είχαμε εκτυπώσει πριν (δηλαδή ΚΟΛΛΗΜΕΝΟ).
- `Serial.println(...)` ; Αυτή τυπώνει δίπλα σε ότι είχαμε εκτυπώσει πριν και ΜΕΤΑ αλλάζει γραμμή (στέλνοντας αόρατους χαρακτήρες «αλλαγής γραμμής»)

Το πρόγραμμα για να δοκιμάσουμε τις εντολές είναι:

```
void setup(){
  Serial.begin(9600); // ξεκίνησε το σειριακό μόνιτορ με ταχύτητα 9600 bps
  Serial.print("Hello World !"); // τύπωσε το μήνυμα στην οθόνη
  Serial.print("123456"); // τύπωσε κι αυτό (θα βγει ΚΟΛΛΗΜΕΝΟ με το προηγούμενο)
  Serial.println("Bye"); // τύπωσε αυτό και άλλαξε γραμμή μετά (πάλι ΚΟΛΛΗΜΕΝΟ)
  Serial.print("Hello again"); // τυπώνεται σε νέα γραμμή (προηγήθηκε: Serial.println)
}

void loop(){
}
```

ΠΑΡΑΤΗΡΗΣΗ: Η loop σε αυτό το πρόγραμμα δεν κάνει τίποτε. Είναι άδεια.

Ο φωτοαντιστάτης: Ένας αισθητήρας φωτός



Φωτοαντιστάτης

Ο φωτοαντιστάτης (Light Dependent Resistor, LDR) είναι ένας αντιστάτης, που η τιμή της αντίστασής του εξαρτάται από το φως που πέφτει πάνω του. Όσο πιο έντονο το φως, τόσο μικρότερη η αντίσταση. Ο φωτοαντιστάτης κατασκευάζεται από ειδικό φωτοευαίσθητο υλικό.

Εφαρμογή 1: LED που ανάβει όταν πέφτει το σκοτάδι

<https://www.vodafonegenerationnext.gr/lessons/fota-poy-anaboyn-sto-skotadi>

Στα πλαίσια της εφαρμογής αυτής οι μαθητές θα έχουν την ευκαιρία να κατασκευάσουν έναν αισθητήρα φωτός με τη χρήση ενός φωτοαντιστάτη (ή αλλιώς: φωτοαντίστασης). Στη συνέχεια, θα χρησιμοποιήσουν μία από τις αναλογικές εισόδους του Arduino για την ανάγνωση της τιμής του αισθητήρα. Η εφαρμογή που θα υλοποιηθεί, θα ανάβει αυτόματα ένα LED, όταν θα μειώνεται ο φωτισμός του χώρου.

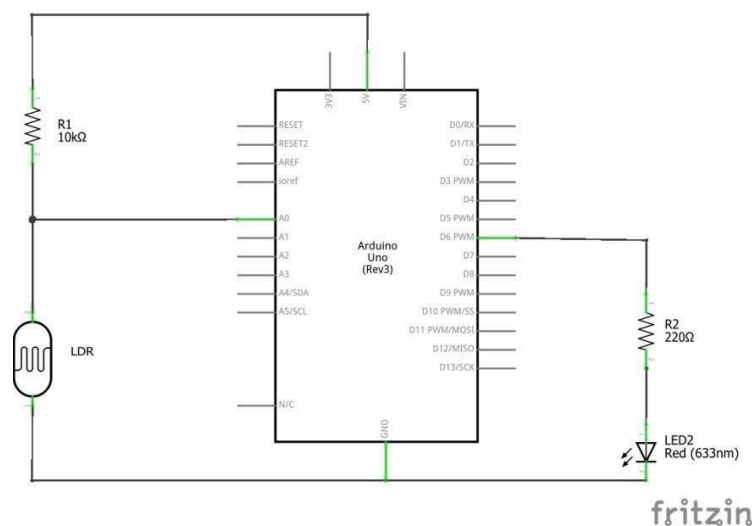
Υλικά

Τα νέα υλικά που θα εισάγουμε σε αυτή την εφαρμογή είναι ένας φωτοαντιστάτης και μία αντίσταση 10 kΩ. Επίσης χρησιμοποιούνται τα ήδη γνωστά: Κόκκινο LED, αντίσταση 220Ω, καλώδια.

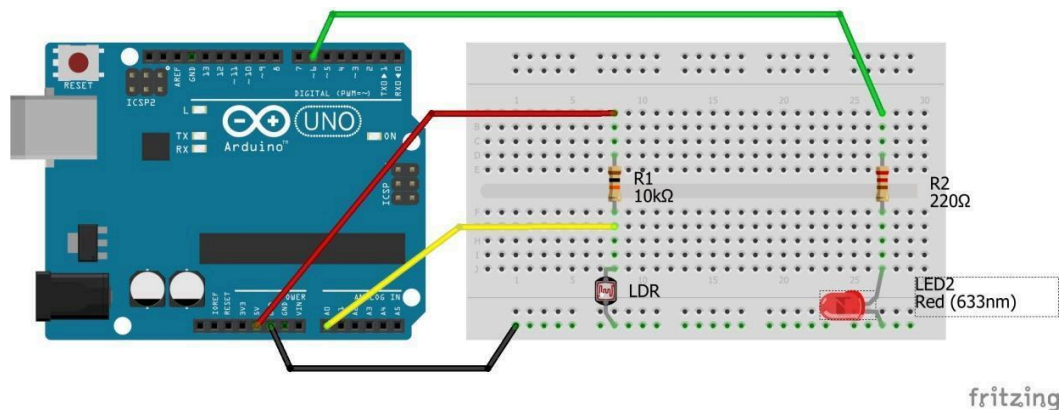
Κύκλωμα

Κατασκευάζουμε το κύκλωμα που δείχνουν οι επόμενες δύο εικόνες:

Σχηματικό διάγραμμα:



Πραγματική υλοποίηση:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

// Πρόγραμμα για απλό πειραματισμό με φωτοαντιστάτη

```
int a; // εδώ θα κρατάμε την τιμή που διαβάζουμε από τον φωτοαντιστάτη
float b; // εδώ θα αποθηκεύουμε την τάση που έχει ο φωτοαντιστάτης στα άκρα του
float c; // εδώ θα αποθηκεύουμε την τιμή της αντίστασης του φωτοαντιστάτη
int d; // εδώ θα αποθηκεύσουμε την τιμή-όριο πάνω από την οποία θα ανάβουμε το LED
```

```
void setup() {
  Serial.begin(9600); // ξεκινάμε το σειριακό μόνιτορ του Arduino IDE
  d=500; // τιμή πάνω από την οποία θα ανάβουμε το LED
  pinMode(6, OUTPUT); // το πιν 6 θα είναι ή ΕΞΟΔΟΣ που θα συνδέσουμε το LED
}
```

```
void loop() {
  a=analogRead(A0); // διάβασε την αναλογική είσοδο A0 και βάλε την τιμή στην a
  Serial.print("Τιμή: "); // απεικόνισε την τιμή της a στο σειριακό μόνιτορ του Arduino IDE
  Serial.print(a);
  Serial.print("\t");

  b=a*5.0/1023; // υπολόγισε την τιμή της τάσης στα άκρα του φωτοαντιστάτη
  (αντιστοιχία: 1023 --> 5V)
```

```

Serial.print("Τάση: ");
Serial.print(b); // απεικόνισε την τιμή της τάσης
Serial.print(" V \t");

c=b*10.0/(5-b); // υπολόγισε την αντίσταση του φωτοαντιστάτη σε kΩ

Serial.print(c);
Serial.println(" kΩ");
If(a>d) // αν η τιμή του a είναι μεγαλύτερη από το d (όριο)
    digitalWrite(6, HIGH); // άναψε το LED
else // αλλιώς
    digitalWrite(6, LOW); // σβήσε το LED
delay(500); // περίμενε εδώ 0,5 δευτερόλεπτο
}

```

Εντολές που χρησιμοποιήθηκαν:

Χρησιμοποιήσαμε μια εντολή ανάγνωσης για αναλογική είσοδο:

- analogRead(A0): Αυτή διαβάζει την αναλογική είσοδο A0 και δίνει μια τιμή μεταξύ 0 και 1023, που αντιστοιχεί σε τάση από 0 ως 5V.

Χρησιμοποιήσαμε κάποιες εντολές για το Serial Monitor του Arduino IDE.

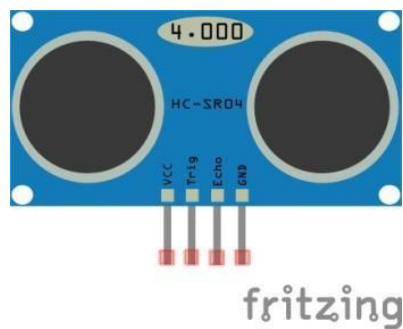
- Serial.begin(9600); Αυτή ξεκινάει την επικοινωνία μεταξύ Arduino και Serial Monitor στο IDE. Το «9600» είναι η ταχύτητα επικοινωνίας (9600 bps).
- Serial.print(...); Αυτή τυπώνει ότι έχουμε στην παρένθεση.
- Serial.println(...); Αυτή τυπώνει ότι έχουμε στην παρένθεση και ΜΕΤΑ αλλάζει γραμμή (στέλνοντας έναν αόρατο χαρακτήρα «αλλαγής γραμμής»)

ΠΑΡΑΤΗΡΗΣΗ: Το κύκλωμα χρησιμοποιεί έναν διαιρέτη τάσης αποτελούμενο από την αντίσταση 10kΩ και τη φωτοαντίσταση (βρίσκεται στην αριστερή άκρη του σχηματικού διαγράμματος). Χρησιμοποιώντας τον διαιρέτη τάσης πετυχαίνουμε να πάρουμε μια τάση που αλλάζει, εξαρτούμενη από το φως που πέφτει στη φωτοαντίσταση. Αυτή την τάση μετράει το Arduino μέσω της αναλογικής εισόδου A0.

Ο αισθητήρας απόστασης HC-SR04

Τι είναι

Ο HC-SR04 είναι ένας αισθητήρας απόστασης υπερήχων. Όπως φαίνεται και στην Εικόνα, διαθέτει έναν πομπό και ένα δέκτη υπερήχων, καθώς και 4 ακροδέκτες (επαφές σύνδεσης). Οι δύο ακριανοί ακροδέκτες VCC και GND, χρησιμοποιούνται για την τροφοδοσία του αισθητήρα και συνδέονται στην τάση (5V) και τη γείωση αντίστοιχα. Ο ακροδέκτης Trig χρησιμοποιείται για την εκκίνηση της διαδικασίας μέτρησης και ο ακροδέκτης Echo χρησιμοποιείται για την έξοδο του αποτελέσματος. Οι δύο αυτοί ακροδέκτες συνδέονται σε δύο ψηφιακές ακίδες (πιν) του Arduino.



Πώς λειτουργεί

Για να ξεκινήσει η διαδικασία της μέτρησης, πρέπει να στείλουμε στον ακροδέκτη Trig έναν παλμό High με διάρκεια τουλάχιστον 10 μsec. Μόλις ο αισθητήρας λάβει το σήμα ενεργοποίησης, στέλνει από τον πομπό μια ακολουθία υπερήχων. Όταν οι υπέρηχοι συναντήσουν κάποιο εμπόδιο αντανακλώνται και επιστρέφουν προς τον αισθητήρα, όπου και ανιχνεύονται από το δέκτη. Στη συνέχεια, ο αισθητήρας βγάζει ως έξοδο στον ακροδέκτη Echo έναν παλμό HIGH. Η διάρκεια του παλμού είναι ίση με το χρόνο που πέρασε από τη στιγμή της εκπομπής των υπερήχων, μέχρι τη λήψη της αντανάκλασης.

Πώς υπολογίζεται η απόσταση από το εμπόδιο

Το Arduino με κατάλληλες εντολές μετράει τη διάρκεια του παλμού που βγάζει ως έξοδο ο αισθητήρας, έστω *duration*. Με δεδομένο ότι οι υπέρηχοι ταξιδεύουν με την ταχύτητα του ήχου ($340\text{m/s} = 0,034\text{cm}/\mu\text{s}$) και με βάση τον τύπο της ταχύτητας ($u=s/t$), αν *distance* είναι η απόσταση από το εμπόδιο έχουμε:

$$0,034 = \frac{\text{duration} \times 2}{\text{distance}} \quad \text{duration} = \frac{0,034 \times \text{distance}}{2} \approx \frac{\text{distance}}{59}$$

Η διαίρεση με το 2, προκύπτει από το γεγονός ότι η διάρκεια του παλμού αντιστοιχεί στο χρόνο που έκαναν οι υπέρηχοι να πάνε μέχρι το εμπόδιο και να γυρίσουν πίσω στον αισθητήρα. Άρα η απόσταση που καλύπτουν οι υπέρηχοι σε αυτό το χρόνο, είναι η διπλάσια από αυτήν που θέλουμε να υπολογίσουμε.

Εφαρμογή 2: Μέτρηση απόστασης και απεικόνιση στο Serial Monitor

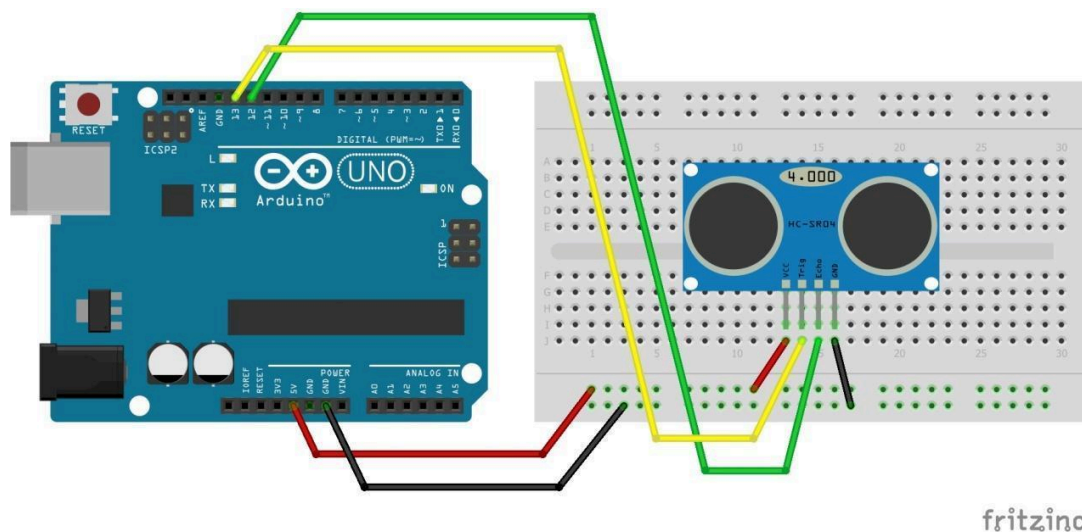
Στην εφαρμογή αυτή θα μετράμε απόσταση από κάποιο αντικείμενο με τον αισθητήρα υπερήχων και θα την εμφανίζουμε στο Serial Monitor του Arduino IDE.

Υλικά

Χρειαζόμαστε: την πλακέτα HC-SR04 (αισθητήρας απόστασης με υπερήχους), Arduino UNO, breadboard, Καλώδια jumper

Κύκλωμα

Κατασκευάζουμε το παρακάτω κύκλωμα:



fritzing

Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα σε κώδικα C του Arduino:

```
int echoPin = 12; // Echo Pin
int trigPin = 13; // Trigger 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);
  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("Εκτός ορίων");
delay(100);
}
else {
Serial.println(distance);
delay(50);
};

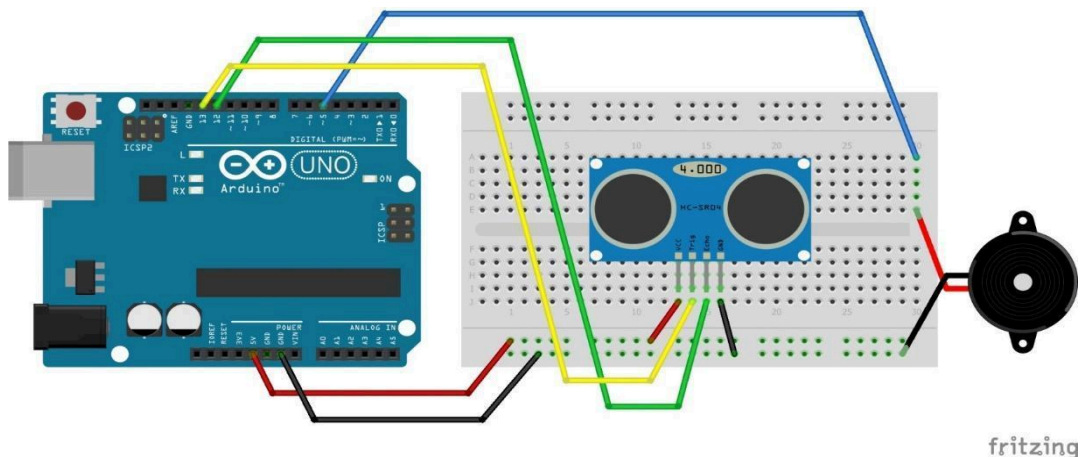
//Καθυστέρηση 50ms πριν την επόμενη ανάγνωση
delay(50);
}

```

Optional Εφαρμογή 1: Συναγερμός προσέγγισης με αισθητήρα υπερήχων και βομβητή (buzzer)

Στην εφαρμογή αυτή ο αισθητήρας υπερήχων ανιχνεύει την απόσταση από κάποιο αντικείμενο που πλησιάζει και ηχεί συναγερμός με τον βομβητή όταν το αντικείμενο πλησιάσει πολύ κοντά. Μπορεί να χρησιμοποιηθεί ως αισθητήρας προσέγγισης π.χ. σε κάποιο όχημα.

Κατασκευάζουμε το παρακάτω κύκλωμα με βάση το προηγούμενο. Προσθέτουμε μόνο τον βομβητή και κάποια καλώδια.



Κώδικας σε γλώσσα C του Arduino:

```
int echoPin = 12; // Πιν Echo της πλακέτας υπερήχων
int trigPin = 13; // Πιν Trigger της πλακέτας υπερήχων
int alarmPin = 5; // Βομβητής
int maximumRange = 200; // Μέγιστη επιτρεπόμενη εμβέλεια
int minimumRange = 0; // Ελάχιστη επιτρεπόμενη εμβέλεια
int alarmRange=10; // Η απόσταση στην οποία θα χτυπήσει συναγερμός
float duration, distance;
void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(alarmPin, OUTPUT);
  Serial.begin(9600);
}

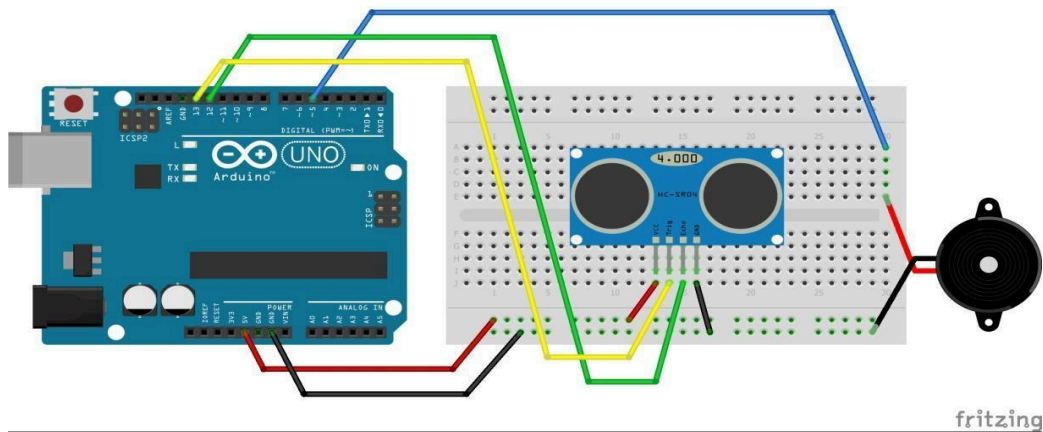
void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  //Υπολογισμός απόστασης (σε cm) βασιζόμενοι στην ταχύτητα του ήχου.
  distance = duration/58.2;
  if (distance >= maximumRange || distance <= minimumRange){
    Serial.println("Εκτός ορίων");
    delay(100);
  }
  else {
    Serial.println(distance,2);
    if(distance<alarmRange)
      digitalWrite(5,HIGH); // Άναψε το βομβητή
    else
      digitalWrite(5,LOW); // Σβήσε το βομβητή
  }
  delay(50); //Καθυστέρηση 50ms πριν την επόμενη ανάγνωση
```


}

Optional Εφαρμογή 2: Βελτιωμένος συναγερμός προσέγγισης με αισθητήρα υπερήχων και βομβητή (buzzer)

Στην εφαρμογή αυτή χρησιμοποιούμε το ίδιο κύκλωμα με την Εφαρμογή 6^α, αλλά βελτιωμένο λογισμικό για να αποφύγουμε περιστασιακούς παρασιτικούς συναγερμούς. Το λογισμικό παίρνει αρκετές μετρήσεις και εξάγει μέσο όρο πριν αποφασίσει να χτυπήσει συναγερμό.

Κατασκευάζουμε το παρακάτω κύκλωμα, ίδιο με το προηγούμενο.



Κώδικας σε γλώσσα C του Arduino:

```
// Πρόγραμμα που ηχεί βομβητή  
// όταν πλησιάσει κάποιος κοντά  
// στον αισθητήρα απόστασης με υπερήχους.  
// Βελτίωση: Απορρίπτει περιστασιακές παρασιτικές τιμές.
```

```
int echoPin = 12; // Πιν Echo της πλακέτας υπερήχων  
int trigPin = 13; // Πιν Trigger της πλακέτας υπερήχων  
int alarmPin = 5; // Βομβητής int a[20]; // Πίνακας-αποθήκη 20 τελευταίων τιμών απόστασης  
int alarmRange=10; // Η απόσταση στην οποία θα χτυπήσει συναγερμός σε cm  
unsigned long duration;  
bool ignore; // σημαία για το αν θα αγνοηθεί μια τιμή  
int i, sum, counter=0;  
int mo, maxRange=500, minRange=6; // μέσος όρος, όρια παρασιτικών (παράλογων) τιμών  
int distance; // απόσταση σε cm (ακέραιος. τα δεκαδικά δε μας ενδιαφέρουν)
```

```

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(alarmPin, 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;
  sum=0;
  for(i=0; i<19; i++) // ολίσθησε αριστερά τις 19 τιμές στον πίνακα
  {
    a[i]=a[i+1];
    sum=sum+a[i]; // πρόσθεσε την τιμή στο άθροισμα τιμών
  }
  if(distance>maxRange || distance<minRange) // αν η τιμή είναι εκτός λογικών ορίων...
    distance=mo; // βάλε το μέσο όρο στη θέση της

  sum=sum+distance; // τώρα το άθροισμα έχει όλες τις τιμές του πίνακα ενημερωμένου

  a[19]=distance; // αντικατάστησε την τελευταία θέση του a[] με την πρόσφατη τιμή
  απόστασης

  mo=sum/20; // μέσος όρος των 20 τιμών απόστασης
  if(distance>mo+mo/4 || distance<mo-mo/4) // αν η πρόσφατη τιμή απέχει πολύ από το
  μέσο όρο...
    ignore=true; // σημείωσε να μην ενεργοποιήσει τον συναγερμό
  else
    ignore=false;

```

```
Serial.print("counter=");
Serial.println(counter);
Serial.print("mo=");
Serial.println(mo);
Serial.println(distance); // εκτύπωσε απόσταση στο μόνιτορ του IDE

if(distance<alarmRange && counter>=20 && ignore==false)
// δε χτυπάμε συναγερμό για τις 20 πρώτες μετρήσεις
// ή αν η ignore είναι true
    digitalWrite(5,HIGH); // Άναψε το βομβητή
else
    digitalWrite(5,LOW); // Σβήσε το βομβητή

counter=counter+1; // αύξησε το μετρητή
if(counter>20) counter=20; // αν πέρασε το 20, κράτα τον στο 20
delay(20); //Καθυστέρηση πριν την επόμενη επανάληψη
}
```

Η οθόνη LCD

Η οθόνη LCD 16x2 είναι μια οθόνη που απεικονίζει 2 σειρές των 16 χαρακτήρων. Εμείς χρησιμοποιούμε μια οθόνη LCD με επικοινωνία IIC ή I2C. Η οθόνη μας να έχει πίσω από την κεντρική πλακέτα της και μία μικρότερη πλακέτα που είναι το κύκλωμα για την επικοινωνία I2C μεταξύ οθόνης και Arduino. Η σύνδεση με αυτό το πρωτόκολλο επικοινωνίας απαιτεί μόνο 4 καλώδια μεταξύ των δύο συσκευών που



αναφέραμε: +5V, GND, SDA, SCL.

Εφαρμογή 3 : Οθόνη LCD με σύνδεση I2C και “Hello world!”

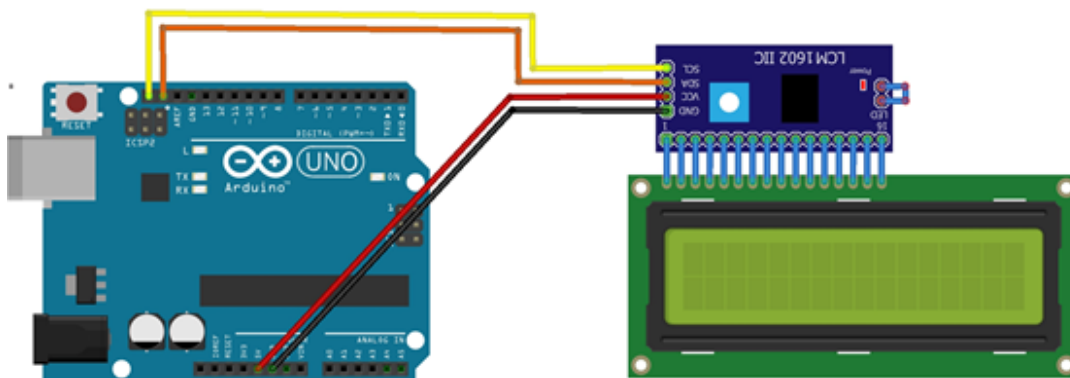
Στην εφαρμογή αυτή εμφανίζουμε στην οθόνη το γνωστό μήνυμα πρώτου πειραματισμού «Hello world!».

Υλικά

Οθόνη LCD I2C, Arduino UNO, Καλώδια jumper

Κύκλωμα

Κατασκευάζουμε το κύκλωμα που δείχνει η παρακάτω εικόνα:



Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα “Hello World” για δοκιμή της οθόνης I2C LCD είναι:

```
// I2C LCD screen demo
//Compatible with the Arduino IDE 1.0
//Library version:1.1
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

LiquidCrystal_I2C lcd(0x27,16 ,2); // set the LCD address to 0x27, set 16 chars and 2 line
display

void setup()
{
  lcd.init();           // initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(1,0);
  lcd.print(" Hello, world! ");
  lcd.setCursor(0,1);
  lcd.print(" I2C LCD screen.");
}

void loop()
{
}
```

Αντιμετώπιση προβλημάτων

No1: Η οθόνη LCD δεν φωτίζεται

Δεν ανάβει η οθόνη LCD; (εννοείται ότι κάνατε σωστά τις 4 συνδέσεις με το Arduino). Γυρίστε την οθόνη σας από πίσω. Έχει ένα μαύρο πλαστικό βραχυκυκλωτήρα (jumper) στην άκρη της μικρής πλακέτας; Ή βλέπετε δύο γυμνά πιν στον αέρα; Στη δεύτερη περίπτωση πρέπει να τα ενώσετε με ένα κομμάτι καλώδιο ή και να τα κολλήσετε με κολλητήρι (ή να βάλετε ένα βραχυκυκλωτήρα, φυσικά). Αυτά τα πιν πρέπει να είναι ενωμένα για να ανάβει ο φωτισμός της οθόνης.

Παρατηρήθηκε από εκπαιδευτή σε προηγούμενο κύκλο να ΛΕΙΠΕΙ ο βραχυκυκλωτήρας από τη μικρή πλακέτα! Κανονικά πρέπει να υπάρχει στη θέση του από το εργοστάσιο. Μπορείτε να βρείτε βραχυκυκλωτήρες (πέρα από καταστήματα ηλεκτρονικών) από κάποια παλιά μητρική πλακέτα υπολογιστή (στα εργαστήρια πληροφορικής έχει κάποιες χαλασμένες που υπάρχουν για επίδειξη).

No2: Η οθόνη LCD δεν εμφανίζει κείμενο

1. Φορτώσατε την εφαρμογή στο Arduino αλλά δεν εμφανίζετε τίποτε στην οθόνη; Το πρώτο που πρέπει να κάνετε είναι να αποσυνδέσετε το Arduino από την τάση και να ελέγξετε ξανά τις συνδέσεις σας.
2. Αν οι συνδέσεις σας είναι εντάξει και δεν εμφανίζεται τίποτε στην οθόνη: Δοκιμάστε να στρέψετε το μικρό ποτενσιόμετρο τρίμερ που υπάρχει στην πίσω πλευρά της οθόνης, επάνω στη μικρή πλακέτα. Ταυτόχρονα ελέγχετε αν η εικόνα εμφανίζεται. Αυτό το τρίμερ ρυθμίζει το κοντράστ (αντίθεση) της εικόνας και σε κάποιες οθόνες δεν είναι σωστά ρυθμισμένο από το εργοστάσιο.
3. Αν και πάλι δεν μπορείτε να εμφανίσετε κείμενο στην οθόνη ενώ τρέξατε την εφαρμογή (και δοκιμάσατε να στρέψετε το τρίμερ ποτενσιόμετρο στην πίσω πλευρά και πάλι τίποτε), μπορεί η διεύθυνση επικοινωνίας οθόνης -Arduino μέσω I2C να είναι άλλη και όχι η 0x27 (που έχουν οι περισσότερες που στάλθηκαν από την SciCo). Τότε, απλά ανιχνεύουμε τη διεύθυνση που έχει η οθόνη με ένα πρόγραμμα I2C scanner. Υπάρχει το λινκ παρακάτω. Αντιγραφή - επικόλληση στο IDE και τρέξτε το. Θα σας πει ποια διεύθυνση πρέπει να χρησιμοποιείτε στα προγράμματά σας με την "δύσκολη" οθόνη LCD:

https://playground.arduino.cc/Main/I2cScanner?fbclid=IwAR2o_itSEALt0bk8iT9vDq-lnm9HXk5z7pUmWY-rHZrBdFq_VqNJcZnz_c

ή απλά κάντε έρευνα στο διαδίκτυο για ένα i2c scanner για Arduino.

Εφαρμογή 4: Γράφω κείμενο στο Serial monitor και εμφανίζεται την οθόνη LCD

Ένα ακόμη πρόγραμμα δοκιμής είναι αυτό, που μας επιτρέπει να γράφουμε στην οθόνη LCD ότι μήνυμα θέλουμε, πληκτρολογώντας το στο παράθυρο εισόδου του σειριακού μόνιτορ:

```
#include <Wire.h>
```

```

#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
void setup()
{
  lcd.init();           // initialize the lcd
  lcd.backlight();
  Serial.begin(9600);
}

void loop()
{
  // when characters arrive over the serial port...
  if (Serial.available()) {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0) {
      // display each character to the LCD
      lcd.write(Serial.read());
    }
  }
}

```

Εφαρμογή 5: Εμφανίζω στην οθόνη LCD ένα μήνυμα για TO Generation Next

Ένα τρίτο πρόγραμμα δοκιμής είναι αυτό, ειδικά για το «**Generation Next**»:

```

#include <Wire.h>
#include <LiquidCrystal_I2C.h>
//I2C pins declaration
LiquidCrystal_I2C lcd(0x3F, 16, 2);

void setup()
{

```

```

lcd.init();//Defining 16 columns and 2 rows of lcd display
lcd.backlight();//To Power ON the back light
//lcd.backlight();// To Power OFF the back light
}
void loop()
{
lcd.setCursor(0,0); //Defining positon to write from first row, first column .
lcd.print(" Generation Next"); // You can write 16 Characters per line .
delay(1000);//Delay used to give a dynamic effect
lcd.setCursor(0,1); //Defining positon to write from second row, first column .
lcd.print(" Youth ");
delay(8000);
lcd.clear();//Clean the screen
lcd.setCursor(0,0);
lcd.print(" I2C LCD Screen ");
lcd.setCursor(0,1);
lcd.print(" DEMO ");
delay(8000);
}

```

Σερβομηχανισμός (servo): Ένας μηχανισμός που κινεί

Ο σερβομηχανισμός (servo) είναι μια μηχανική διάταξη που περιστρέφει ένα μικρό πλαστικό βραχίονα στην επιθυμητή από εμάς θέση. Η συνήθης χρήση σερβομηχανισμών είναι σε τηλεκατευθυνόμενα μοντέλα. Πλέον χρησιμοποιούνται και σε κατασκευές με Arduino και η τιμή τους είναι πολύ χαμηλή. Είναι ο πιο εύκολος τρόπος να κινήσουμε «εμπρός πίσω» κάποιο μηχανικό μέρος ή να περιστρέψουμε.



Σερβομηχανισμός

(servo)

Ο σερβομηχανισμός έχει 3 ακροδέκτες για σύνδεση. Συνήθως είναι χρωματισμένα τα αντίστοιχα καλώδια ως εξής: καφέ (GND), κόκκινο (+5V), πορτοκαλί (σήμα χειρισμού, είσοδος του σερβομηχανισμού).

Ένα τυπικό servo συνοδεύεται από βραχίονες (λευκούς πλαστικούς συνήθως, σε διάφορα σχήματα: σταυροειδή, ραβδόμορφο με δύο σκέλη, ραβδόμορφο με ένα σκέλος), βίδες.



Για τον έλεγχο του σερβομηχανισμού πρέπει να στέλνουμε ένα σήμα με παλμούς μεταβαλλόμενου χρονικού πλάτους (PWM Pulse Width Modulation). Αυτό είναι αρκετά εύκολο να γίνει με το Arduino με τη βοήθεια της αντίστοιχης βιβλιοθήκης Servo.

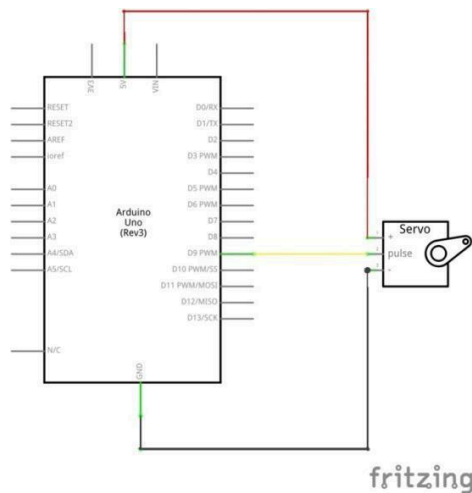
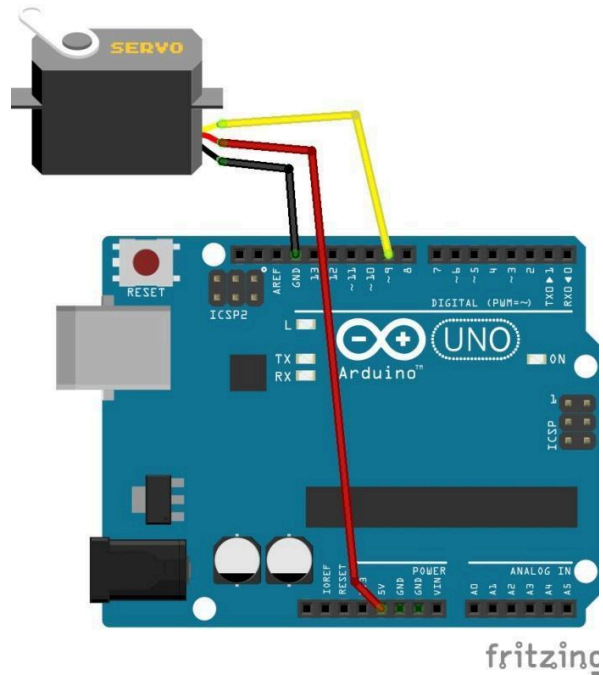
Εφαρμογή 6: Σερβομηχανισμός που περιστρέφεται δεξιά αριστερά

Υλικά

Χρειαζόμαστε: Arduino UNO, servo, καλώδια jumper.

Κύκλωμα

Το κύκλωμα που θα κατασκευάσουμε είναι το παρακάτω:



Το πρόγραμμα σε Arduino C για πειραματισμό με το servo είναι:

```
// Πειραματισμός με servo No1
```

```
// το servo περιστρέφεται από τη μία ακραία θέση του στην άλλη και πάλι πίσω
```

```
#include <Servo.h> // Συμπερίλαβε τη βιβλιοθήκη του σέρβο
```

```
Servo myservo; // Δημιούργησε ένα αντικείμενο τύπου Servo
```

```
int s=1; // Το βήμα (μοίρες) που θα αυξάνουμε τη θέση του σέρβο σε κάθε επανάληψη
```

```
void setup()
```


Ανάπτυξη προγράμματος σε Arduino IDE

Το πρόγραμμα που θα χρησιμοποιήσουμε είναι:

```
int a; // Εδώ αποθηκεύουμε την τιμή που διαβάσαμε από την αναλογική είσοδο A0
(από 0 ως 1023)
float temp; // Εδώ αποθηκεύουμε την τιμή της θερμοκρασίας

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

void loop() {

  a=analogRead(A0); // Διάβασε την τιμή από την A0

  temp = (5.0 * analogRead(A0) * 100.0) / 1023; // Υπολόγισε την θερμοκρασία σε
βαθμούς Κελσίου

  Serial.print(temp); // Απεικόνισε τη θερμοκρασία στο σειριακό μόνιτορ του Arduino
IDE
  Serial.println(" C");

  delay(1000); // Περίμενε εδώ 1 δευτερόλεπτο

}
```

Εφαρμογή έμπνευσης από τους μαθητές των Χανίων

Σύστημα καταγραφής πελατών Safe Entry από τους Upcoming Scientists

Περιγραφή προβλήματος:

Το πρόβλημα το οποίο προσπαθούμε να επιλύσουμε είναι η μη ασφαλής είσοδος στα super markets λόγω συνωστισμού και παραβίασης των μέτρων που καθορίζουν τον αριθμό των πελατών που επιτρέπονται σε ένα κατάστημα ανάλογα με τα τετραγωνικά του. Ελπίζουμε ότι με μια συσκευή σαν τη δική μας θα περιορίσουμε την μετάδοση του COVID-19 σε πολυσύχναστους χώρους όπως τα super markets.

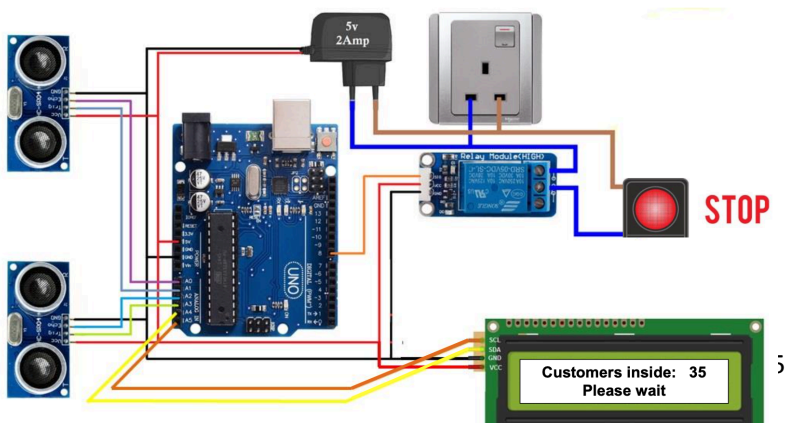
Περιγραφή προτεινόμενης λύσης:

Η πρόταση μας αποτελείται από δύο μέρη. Το 1ο και σημαντικότερο κομμάτι της είναι η συσκευή μας "Safe Entry" η οποία λειτουργεί σαν μετρητής των πελατών που βρίσκονται ανά πάσα στιγμή μέσα στο κατάστημα. Μετράει αυτούς που εισέρχονται και εξέρχονται κρατώντας πάντα λογαριασμό για το πόσοι βρίσκονται κάθε χρονική στιγμή μέσα στο κατάστημα. Το 2ο κομμάτι είναι ο αυτόματος μηχανισμός παροχής αντισηπτικού με το οποίο οι πελάτες θα βάζουν αντισηπτικό στα χέρια τους ανέπαφα, αποφεύγοντας έτσι τον κίνδυνο μετάδοσης μέσω των επαφών.



Κυκλωματική διάταξη

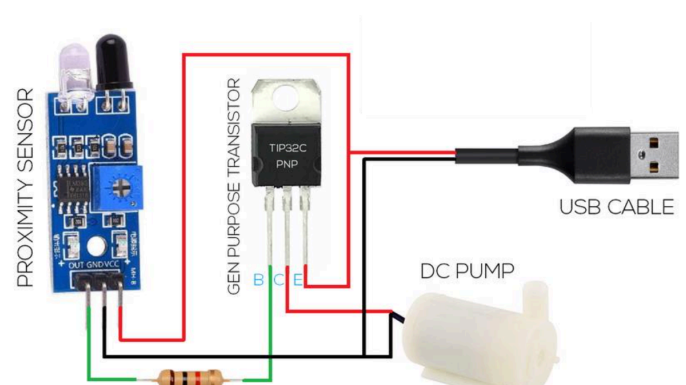
Safe Entry – 6ο Γυμνάσιο Χανίων



6ο Γυμνάσιο Χανίων

Upcoming Scientists

Safe Entry automatic alcohol dispenser



Κώδικας

```
// Safe Entry – 6ο Gymnasio – Σταύρος, Φάνης, Κωνσταντίνος, Γιάννης
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27, 20, 4);

#define e_s1 A0 //echo pin
#define t_s1 A1 //Trigger pin
#define e_s2 A2 //echo pin
#define t_s2 A3 //Trigger pin
int relay = 8; // Out for light
long dis_a=0,dis_b=0;
int flag1=0, flag2=0;
int person = 0;
//*****ultra_read*****
void ultra_read(int pin_t,int pin_e,long &ultra_time){
long time;
pinMode(pin_t,OUTPUT);
pinMode(pin_e,INPUT);
digitalWrite(pin_t,LOW);
delayMicroseconds(2);
digitalWrite(pin_t,HIGH);
delayMicroseconds(10);
time=pulseIn (pin_e,HIGH);
ultra_time = time / 29 / 2;
}
void setup(){
Serial.begin(9600);// initialize serial communication at 9600 bits per second:
pinMode(relay, OUTPUT);
lcd.init();
lcd.backlight();
lcd.begin(20, 4);
lcd.setCursor(0, 0);
lcd.print(" 6ο Gymnasio ");
lcd.setCursor(0, 1);
lcd.print(" Safe Entry Counter");
lcd.setCursor(0, 2);
lcd.print(" project by ");
lcd.setCursor(0, 3);
lcd.print("Upcoming Scientists");
delay(10000); // Waiting for a while
lcd.clear();
}
void loop(){
//*****
ultra_read(t_s1,e_s1,dis_a);delay(30);
ultra_read(t_s2,e_s2,dis_b);delay(30);
```

```

//*****
Serial.print("da:");Serial.println(dis_a);
Serial.print("db:");Serial.println(dis_b);
if(dis_a<90 && flag1==0){flag1=1;
if(flag2==0){person = person+1;}
}
if(dis_b<90 && flag2==0){flag2=1;
if(flag1==0){person = person-1;}
}
if(dis_a>90 && dis_b>90 && flag1==1 && flag2==1){
flag1=0, flag2=0;
delay(1000);
}
if(person<=0){
digitalWrite(relay,LOW);
lcd.setCursor(0, 0);
lcd.print(" 6o Gymnasio ");
lcd.setCursor(0, 1);
lcd.print("No Customers Inside");
lcd.setCursor(0,2);
lcd.print(" Welcome ");
lcd.setCursor(0,3);
lcd.print(" Please Enter ");
}
else if(person<10) {
digitalWrite(relay,LOW);
lcd.setCursor(0, 0);
lcd.print(" 6o Gymnasio ");
lcd.setCursor(0, 1);
lcd.print("Customers Inside: ");
lcd.print(person);
lcd.print(" ");
lcd.setCursor(0,2);
lcd.print(" Welcome ");
lcd.setCursor(0,3);
lcd.print(" Please Enter ");
}
else{digitalWrite(relay,HIGH);
lcd.setCursor(0, 0);
lcd.print("Customers Inside: ");
lcd.print(person);
lcd.print(" ");
lcd.setCursor(0,1);
lcd.print(" STOP ");
lcd.setCursor(0,2);
lcd.print(" PLEASE WAIT ");
lcd.setCursor(0,3);
lcd.print(" Too many customers");
}

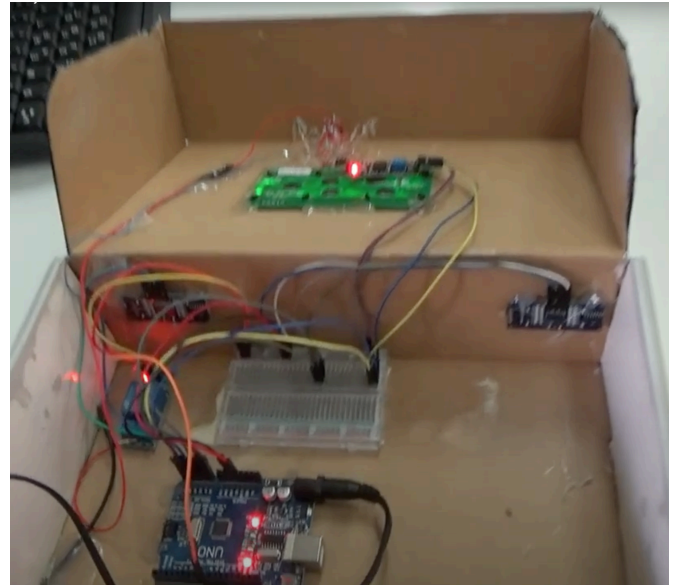
```


}

Περιγραφή τεχνολογίας που χρησιμοποιήθηκε:

Για την συσκευή "μετρητή" χρησιμοποιήθηκαν: 1 x Arduino UNO, 2 x HC-SR04 distance sensors, 1x LCD display 20x4, 1x 5V relay module, 1 x Breadboard και καλώδια

Για τον αυτόματο μηχανισμό παροχής αντισηπτικού χρησιμοποιήθηκαν: 1 x medical tube, 1 x βάζο γυάλινο, 1 x 5V dc water pump, 1 x Transistor PNP 3A - TIP32, καλώδια και 1 Powerbank.



Επικοινωνία με Τοπικούς Φορείς:

Μέσω Email κάναμε αίτημα στην αλυσίδα Super Market "SYNKA" να επισκεφτούμε ένα κατάστημα και να μελετήσουμε τον τρόπο που γίνεται η καταμέτρηση των πελατών και με ποιον τρόπο τηρούνται όλα τα μέτρα ασφαλείας για την αποφυγή της διάδοσης του COVID-19. Στη συνέχεια και αφού ολοκληρώσαμε την κατασκευή μας πήγαμε σε ένα κατάστημα των Super Markets και δοκιμάσαμε τη συσκευή στον φυσικό χώρο. Οι υπεύθυνοι του Super Market ήταν πολύ ευγενικοί και συνεργάσιμοι βοηθώντας μας να πραγματοποιήσουμε τη δοκιμή.

Περιγραφή κοινωνικού αντικτύπου:

Με τη συσκευή αυτή πιστεύουμε ότι συμβάλλουμε στην καταπολέμηση της διασποράς του ιού.

