

ΚΩΔΙΚΑΣ ΕΛΕΓΚΤΗ Νο1

```
const int pinHalftime = A1;      // Κουμπί αδειάσματος μισού καζανιού
const int pinFulltime = A2;      // Κουμπί αδειάσματος όλου του καζανιού
const int pinSinkDrainTank = A3;  // Ένδειξη δεξαμενής νερού αποχέτευσης νιπτήρα
const int pinCisternTank = A4;    // Ένδειξη νερού απο το καζανάκι

const int pinPumpFaucet = 22;     // Διαχείριση αντλίας νερού βρύσης
const int pinPumpBowl = 24;       // Διαχείριση αντλίας νερού λεκάνης
const int pinPumpCisternTank = 26; // Διαχείριση αντλίας νερού καζανάκι τουαλέτας
const int pinPumpSinkDrainTank = 28; // Διαχείριση αντλίας νερού αποχέτευσης νιπτήρα

int echoPin = 8;                  // Ορισμός μονάδα υπερήχων ECHO στο D8
int triggerPin = 9;              // Ορισμός μονάδα υπερήχων TRIG στο D9

void setup () {
  Serial.begin(9600);            // Καθορισμός ταχύτητας σειριακής επικοινωνίας

  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW); // σβήσιμο εσωτερικού led

  pinMode (pinFulltime, INPUT);
  pinMode (pinHalftime, INPUT);
  pinMode (pinSinkDrainTank, INPUT);
  pinMode (pinCisternTank, INPUT);
  pinMode (echoPin, INPUT);

  pinMode (pinPumpFaucet, OUTPUT);
  pinMode (pinPumpBowl, OUTPUT);
  pinMode (pinPumpCisternTank, OUTPUT);
  pinMode (pinPumpSinkDrainTank, OUTPUT);
  pinMode (triggerPin, OUTPUT);

}

void loop() {
  int x1 =analogRead(pinFulltime); //Αδειασμα ολόκληρου του καζανιού
  if (x1 < 100) {
    digitalWrite(pinPumpBowl, HIGH);
    delay(10000);
    digitalWrite(pinPumpBowl, LOW);
  }

  int x2 =analogRead(pinHalftime); //Αδειασμα μισού καζανιού
  if (x2 < 100) {
    digitalWrite(pinPumpBowl, HIGH);
    delay(5000);
    digitalWrite(pinPumpBowl, LOW);
  }

  //float x3 = 0.01723 * readUltrasonicDistance(9, 8); //Ανίχνευση χεριών στο νιπτήρα
```

```

float x3= Ultrasonic_Ranging();
if (x3 > 0 && x3 < 10) {
    digitalWrite(pinPumpFaucet, HIGH);
}
else {
    delay(3000);
    digitalWrite(pinPumpFaucet, LOW);
}

float x4 = analogRead(pinSinkDrainTank);  //Άδειασμα της δεξαμενής αποχέτευσης του
νυπτήρα
if (x4 > 400) {
    digitalWrite(pinPumpSinkDrainTank, HIGH);
}
else {
    digitalWrite(pinPumpSinkDrainTank, LOW);
}

float x5 = analogRead(pinCisternTank);  //Γέμισμα στο καζανάκι
if (x5 < 500) {
    digitalWrite(pinPumpCisternTank, HIGH);
}
else {
    digitalWrite(pinPumpCisternTank, LOW);
}

//Serial.print(x1);
//Serial.print(" ");
Serial.println(x5);

delay(1000);
}

long readUltrasonicDistance(int triggerPin, int echoPin) {
    pinMode(triggerPin, OUTPUT); // Clear the trigger
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Sets the trigger pin to HIGH state for 10 microseconds
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    // Reads the echo pin, and returns the sound wave travel time in microseconds
    return pulseIn(echoPin, HIGH);
}

int Ultrasonic_Ranging() {
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);

```

```
digitalWrite(triggerPin, LOW);  
int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων  
distance = distance / 58;           // Μετατροπή του χρόνου παλμού σε απόσταση  
delay(50);  
return distance;  
}
```

ΚΩΔΙΚΑΣ ΕΛΕΓΚΤΗ Νο2

```
const int pinHalftime = A1;      // Κουμπί αδειάσματος μισού καζανιού
const int pinFulltime = A2;      // Κουμπί αδειάσματος όλου του καζανιού
const int pinSinkDrainTank = A3;  // Ένδειξη δεξαμενής νερού αποχέτευσης νιπτήρα
const int pinCisternTank = A4;    // Ένδειξη νερού απο το καζανάκι

const int pinPumpFaucet = 22;     // Διαχείριση αντλίας νερού βρύσης
const int pinPumpBowl = 24;       // Διαχείριση αντλίας νερού λεκάνης
const int pinPumpCisternTank = 26; // Διαχείριση αντλίας νερού καζανάκι τουαλέτας
const int pinPumpSinkDrainTank = 28; // Διαχείριση αντλίας νερού αποχέτευσης νιπτήρα

int echoPin = 8;                  // Ορισμός μονάδα υπερήχων ECHO στο D8
int triggerPin = 9;              // Ορισμός μονάδα υπερήχων TRIG στο D9

void setup () {
  Serial.begin(9600);            // Καθορισμός ταχύτητας σειριακής επικοινωνίας

  pinMode(LED_BUILTIN, OUTPUT);
  digitalWrite(LED_BUILTIN, LOW); // σβήσιμο εσωτερικού led

  pinMode (pinFulltime, INPUT);
  pinMode (pinHalftime, INPUT);
  pinMode (pinSinkDrainTank, INPUT);
  pinMode (pinCisternTank, INPUT);
  pinMode (echoPin, INPUT);

  pinMode (pinPumpFaucet, OUTPUT);
  pinMode (pinPumpBowl, OUTPUT);
  pinMode (pinPumpCisternTank, OUTPUT);
  pinMode (pinPumpSinkDrainTank, OUTPUT);
  pinMode (triggerPin, OUTPUT);

}

void loop() {
  int x1 =analogRead(pinFulltime); //Αδειασμα ολόκληρου του καζανιού
  if (x1 < 100) {
    digitalWrite(pinPumpBowl, HIGH);
    delay(10000);
    digitalWrite(pinPumpBowl, LOW);
  }
  int x2 =analogRead(pinHalftime); //Αδειασμα μισού καζανιού
  if (x2 < 100) {
    digitalWrite(pinPumpBowl, HIGH);
    delay(5000);
    digitalWrite(pinPumpBowl, LOW);
  }

  //float x3 = 0.01723 * readUltrasonicDistance(9, 8); //Ανίχνευση χεριών στο νιπτήρα
```

```

float x3= Ultrasonic_Ranging();
if (x3 > 0 && x3 < 10) {
    digitalWrite(pinPumpFaucet, HIGH);
}
else {
    delay(3000);
    digitalWrite(pinPumpFaucet, LOW);
}

float x4 = analogRead(pinSinkDrainTank);  //Άδειασμα της δεξαμενής αποχέτευσης του
νυπτήρα
if (x4 > 400) {
    digitalWrite(pinPumpSinkDrainTank, HIGH);
}
else {
    digitalWrite(pinPumpSinkDrainTank, LOW);
}

float x5 = analogRead(pinCisternTank);  //Γέμισμα στο καζανάκι
if (x5 < 500) {
    digitalWrite(pinPumpCisternTank, HIGH);
}
else {
    digitalWrite(pinPumpCisternTank, LOW);
}

//Serial.print(x1);
//Serial.print(" ");
Serial.println(x5);

delay(1000);
}

long readUltrasonicDistance(int triggerPin, int echoPin) {
    pinMode(triggerPin, OUTPUT); // Clear the trigger
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Sets the trigger pin to HIGH state for 10 microseconds
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);
    pinMode(echoPin, INPUT);
    // Reads the echo pin, and returns the sound wave travel time in microseconds
    return pulseIn(echoPin, HIGH);
}

int Ultrasonic_Ranging() {
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);

```

```
digitalWrite(triggerPin, LOW);  
int distance = pulseIn(echoPin, HIGH); // Ανάγνωση της διάρκειας των υψηλών επιπέδων  
distance = distance / 58;           // Μετατροπή του χρόνου παλμού σε απόσταση  
delay(50);  
return distance;  
}
```