



Δοκιμάζω με κώδικα το Servo

1. Στον κώδικα της Z2 που χρησιμοποιήσαμε συμπληρώνω τα σημεία που είναι απαραίτητα για να λειτουργήσει το servo σύμφωνα με τις προδιαγραφές που βρίσκονται στα σχόλια (//).

```
#include <_____>
const int UsonicTrig = 9; // Ακίδα σκανδαλισμού
const int UsonicEcho = 8; // Ακίδα απόκρισης
const int FlashLED = 13; // Ενσωματωμένο LED
const int HookSrv = ____; // Ακίδα Σερβομηχανισμού

Servo myservo; // Ορίζω το myservo ως σερβομηχανισμό

void setup()
{
  pinMode(UsonicTrig, OUTPUT);
  pinMode(UsonicEcho, INPUT);
  pinMode(FlashLED, OUTPUT);
  myservo.attach(_____, 500, 2500);
}

void loop()
{
  float cm = 0;

  // Ενεργοποίηση αισθητήρα
  long react = readUsonicDist (UsonicTrig, UsonicEcho);
  // Καταμέτρηση της απόκρισης σε εκατοστά
  cm = 0.01723 * react;

  // Όταν η απόσταση είναι μεγαλύτερη του ενός μέτρου
  // να μην ανάβει το LED και η δαγκάνα είναι 180 μοίρες
  if (cm > 100) {
    myservo.write(____);
    digitalWrite (FlashLED, LOW);
  }
}
```

```

    delay(500);
}
// Όταν η απόσταση είναι 20-100 εκατοστά το LED
// να αναβοσβήνει με ρυθμό μισού δευτερολέπτου
// και η δαγκάνα κλείνει στις 90 μοίρες
else if (cm >= 20) {
    myservo.write(_____);
    digitalWrite (FlashLED, HIGH);
    delay (500);
    digitalWrite (FlashLED, LOW);
    delay (500);
}
// Όταν η απόσταση είναι μέχρι 20 εκατοστά
// το LED να αναβοσβήνει με ρυθμό 200 msec
// και η δαγκάνα κλείνει εντελώς
else {
    myservo.write(_____);
    digitalWrite (FlashLED, HIGH);
    delay (200);
    digitalWrite (FlashLED, LOW);
    delay (200);
}
}

// Λειτουργία σκανδαλισμού και μέτρησης χρόνου (απόστασης)
long readUsonicDist(int triggerPin, int echoPin)
{
    // Καθαρισμός σκανδάλης
    digitalWrite(triggerPin, LOW);
    delayMicroseconds(2);
    // Σκανδαλισμός αισθητήρα
    digitalWrite(triggerPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(triggerPin, LOW);

    // Επιστροφή τιμής απόκρισης αισθητήρα
    // (χρόνος σε msec που ταξίδεψε ο υπέρηχος)
    return pulseIn(echoPin, HIGH);
}

```