

```
#include <Servo.h>

Servo myservo1;
Servo myservo2;

int servopin1 = 9;
int servopin2 = 10;

int buzzerpin = 4;
int trigpin = 5;
int echopin = 6;
int buttonpin = 7;

int greenled = 13;
int redled = 12;

float distance;
float threshold = 15; // adjust based on your box's closed distance

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

  pinMode(trigpin, OUTPUT);
  pinMode(echopin, INPUT);
  pinMode(buttonpin, INPUT_PULLUP);
  pinMode(buzzerpin, OUTPUT);
  pinMode(greenled, OUTPUT);
  pinMode(redled, OUTPUT);

  myservo1.attach(servopin1);
  myservo2.attach(servopin2);

  closeBox(); // start closed
}

void loop() {

  distance = getDistance(trigpin, echopin);
  bool buttonPressed = (digitalRead(buttonpin) == LOW);

  Serial.print("Distance: ");
  Serial.println(distance);
```

```

//AUTHORIZED OPEN)

if (buttonPressed) {
  openBoxAuthorized();
  return; // skip everything else while button is held
}

// UNAUTHORIZED OPEN

if (distance > threshold) { //CHANGED THIS TO REVERSE ALARM
  openBoxUnauthorized();
  return; // skip reset while motion is happening
}

// NO BUTTON + NO MOTION > CLOSE BOX

closeBox();
}

float getDistance(int trig, int echo) {
  digitalWrite(trig, LOW);
  delayMicroseconds(2);
  digitalWrite(trig, HIGH);
  delayMicroseconds(10);
  digitalWrite(trig, LOW);

  long duration = pulseIn(echo, HIGH);
  return duration / 148.0; // convert to inches
}

void openBoxAuthorized() {
  myservo1.write(100); // SAME ANGLE AS UNAUTHORIZED
  myservo2.write(100);

  digitalWrite(greenled, HIGH);
  digitalWrite(redled, LOW);

  noTone(buzzerpin);
}

void openBoxUnauthorized() {
  //myservo1.write(100); // SAME ANGLE AS AUTHORIZED
  //myservo2.write(100); turned into comment to remove from working code

```

```
digitalWrite(redled, HIGH);  
digitalWrite(greenled, LOW);  
  
tone(buzzerpin, 1000);  
}  
  
void closeBox() {  
myservo1.write(10);  
myservo2.write(10);  
  
digitalWrite(greenled, LOW);  
digitalWrite(redled, LOW);  
  
noTone(buzzerpin);  
}
```