```
/*
  Adapted From: Analog Input by David Cuartielles and Tom Igoe
  Author: Malcolm Knapp
  Project: Light Sensor to Servo
  Date: 4/10/14
  Version: 0.1
  Description: This code shows how to use a light sensor to control
               the "blink" rate of a servo. In this case "blink" means
               moving between two positions.
 */
// ---------- included libraries ------------
#include <Servo.h>

// ---------- hardware pin defines  -----------
~~int sensorPin = A0;    // select the input pin for the potentiometer~~

// ---------- variable initialization  -----------
~~int sensorValue = 0;  // variable to store the value coming from the sensor~~
int delayTime = 0; //variable that holds the delay time in milliseconds
int scaling = 1;
int maxValue = 750;
int minValue = 300;


// ---------- library initialization  -----------
Servo myservo;  // create servo object to control a servo a maximum of eight servo objects can be created

void setup() {
  Serial.begin(9600);
 // ---------- hardware connections ----------
   myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}


void loop() {
  // Input
  sensorValue = analogRead(sensorPin);
  // Debugging
  Serial.print("Sensor value: ");  Serial.println(sensorValue);


  // Processing
  //Scaling
  delayTime = map (sensorValue, minValue, maxValue, 200, 1023);
  Serial.print ("Delay in milliseconds: "); Serial.println (delayTime);


  // Output
  myservo.write(155);
  delay(delayTime);
  myservo.write(30);
  delay(delayTime);
}
```

```
/*
  Adapted From: Analog Input by David Cuartielles and Tom Igoe
  Author: Malcolm Knapp
  Project: Ultrasonic Sensor to Servo
  Date: 4/10/14
  Version: 0.1
  Description: This code shows how to use a ultrasonic distance sensor to
               control the "blink" rate of a servo. In this case "blink" means
               moving between two positions.
 */
// ---------- included libraries ------------
#include <Servo.h>
#include <NewPing.h>
// ---------- hardware pin defines  -----------
int triggerPin = 12;      // select the pin for ultrasonic trigger
int echoPin = 11;      // select the pin for echo

// ---------- variable initialization  -----------
int delayTime = 0;
unsigned int uS = 0;      // holds the time it took for the pulse to be received
int distance = 0; // holds the distance in centimeters
int scaling = 1;
int maxValue = 200;
int minValue = 5;
int maxDistance = 200;   // in centimeters

// ---------- library initialization  -----------
Servo myservo;  // create servo object to control a servo a maximum of eight servo objects can be created
NewPing sonar(triggerPin, echoPin, maxDistance);

void setup() {
  Serial.begin(9600);
 // ---------- hardware connections ----------
   myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  // Input
  delay(50);  // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.
  uS = sonar.ping(); // Send ping, get ping time in microseconds (uS).
  distance = uS / US_ROUNDTRIP_CM;  // convert time to distance
  // Debugging
  Serial.print("Sensor value: "); Serial.println(distance);

  // Processing
  //Scaling
  delayTime = map (distance, minValue, maxValue, 0, 1023);
  Serial.print ("Delay in milliseconds: "); Serial.println (delayTime);


  // Output
  myservo.write(155);
  delay(delayTime);
  myservo.write(30);
  delay(delayTime);
}
```