Controlling of SERVO DH-03X using Arduino UNO

The Wiring Diagram

Servo Motor (DH-03X) -------------- Arduino Uno

Signal Pin ----------------------- Digital Pin 10

Power Pin (+) -------------------- 5V

Ground Pin (-) ------------------- GND

Diagram

Description automatically generated

The Code

Code1: With Potentio meter

#include <Servo.h>

Servo myservo; // create servo object to control a servo

int potpin = 0; // analog pin used to connect the potentiometer

int val; // variable to read the value from the analog pin

void setup() {

myservo.attach(9); // attaches the servo on pin 9 to the servo object

}

void loop() {

val = analogRead(potpin); // reads the value of the potentiometer (value between 0 and 1023)

val = map(val, 0, 1023, 0, 180); // scale it to use it with the servo (value between 0 and 180)

myservo.write(val); // sets the servo position according to the scaled value

delay(15); // waits for the servo to get there

}

Explanation of the code:

This code is an Arduino sketch that uses the Servo library to control a servo motor based on the value read from a potentiometer (a variable resistor). Let's go through the code step by step:

The code starts by including the Servo library, which provides functions to control servo motors.

It declares a Servo object called myservo, which will be used to control the servo motor.

Two integer variables are declared: potpin and val.

potpin is assigned the value 0, indicating that the potentiometer is connected to analog pin 0 on the Arduino board.

val will be used to store the value read from the potentiometer.

In the setup() function, the myservo.attach(9) statement attaches the servo motor to pin 9 on the Arduino board.

The loop() function is where the main code execution takes place. It runs repeatedly in a loop.

Inside the loop() function:

The analogRead(potpin) function reads the voltage value from the potentiometer, which is connected to pin 0.

The val variable is then assigned the value read from the potentiometer. The analogRead() function returns a value between 0 and 1023.

The map() function is used to scale the val variable from the range of 0-1023 to the range of 0-180. This is done because servos typically have a range of motion from 0 to 180 degrees.

The myservo.write(val) function sets the position of the servo motor based on the scaled val value. It moves the servo to the specified angle.

The delay(15) function introduces a small delay of 15 milliseconds. This delay gives the servo motor time to reach its target position before the next iteration of the loop.

The loop() function continues running indefinitely, repeatedly reading the potentiometer value, scaling it, and adjusting the servo position accordingly.

In summary, this code reads the value from a potentiometer, maps it to the range suitable for a servo motor, and controls the position of the servo motor based on the mapped value. The servo motor moves in response to the changing position of the potentiometer.

Code 2: Without potentiometer continuous run

#include <Servo.h>

Servo myservo; // create servo object to control a servo

void setup()

{

myservo.attach(9); // attaches the servo on pin 9 to the servo object

}

void loop()

{

for(int i=0; i<180; i++) // 0 to 180 degrees

{

myservo.write(i); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

for(int i=180; i>=0; i--) // 180 to 0 degrees

{

myservo.write(i); // tell servo to go to position in variable 'pos'

delay(15); // waits 15ms for the servo to reach the position

}

}

Explanation of the code:

This code is another Arduino sketch that uses the Servo library to control a servo motor. It moves the servo motor in a sweeping motion from 0 to 180 degrees and then back to 0 degrees repeatedly. Let's break down the code:

The code includes the Servo library, which provides functions to control servo motors.

It declares a Servo object called myservo, which will be used to control the servo motor.

In the setup() function, the myservo.attach(9) statement attaches the servo motor to pin 9 on the Arduino board.

The loop() function is where the main code execution takes place. It runs repeatedly in a loop.

Inside the loop() function:

There is a for loop that starts with int i=0 and runs as long as i<180. This loop increments the variable i from 0 to 180.

Inside the loop, myservo.write(i) sets the position of the servo motor to the value of i. As i increases, the servo motor moves from 0 to 180 degrees.

The delay(15) function introduces a small delay of 15 milliseconds to allow the servo motor time to reach the desired position.

After the first for loop, there is another for loop that starts with int i=180 and runs as long as i>=0. This loop decrements the variable i from 180 to 0.

Inside this loop, myservo.write(i) sets the position of the servo motor to the value of i. As i decreases, the servo motor moves from 180 degrees back to 0 degrees.

Again, the delay(15) function provides a small delay between each position change.

The loop() function continues running indefinitely, repeatedly sweeping the servo motor back and forth from 0 to 180 degrees.

In summary, this code creates a servo object, attaches the servo motor to pin 9, and then continuously moves the servo back and forth in a sweeping motion from 0 to 180 degrees. The servo motor's position is controlled using the myservo.write() function, and delays are used to ensure smooth movement.

Reference

<https://docs.arduino.cc/learn/electronics/servo-motors>

https://playground.arduino.cc/Learning/SingleServoExample/