# **[Demo 20: How to control a Servo via Arduino ESP32 Web Server](http://www.iotsharing.com/2017/06/how-to-control-servo-via-arduino-esp32-webserver.html)**

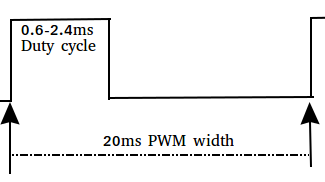
1. **Introduction**  
   - In this tutorial, I will show you how to control a Servo motor via Arduino ESP32 Web Server.



- This servo:

+ Has 3 wires with colors: PWM, Vcc, Ground

+ Can rotate 180 degrees (90 degrees for each direction).

[](https://1.bp.blogspot.com/-eiBCs2MAMEU/WTuhOOQ2yxI/AAAAAAAAECg/4CVIi8utKkoYvOQCTm3jzoTh7A0VDHdigCLcB/s1600/servo2.png)

**Figure: Pulse wave of trigger**

- In order to control the rotation angle of servo we need to trigger pulses to Servo PWM pin. The pulses has pulse width between 0.6–2.4 ms and PWM signal period is 20 ms (50Hz). For example, if we trigger pulse with width is 0.6 ms  then servo is at 0 degrees angle, 1.5 ms  is 90 degrees angle and 2.4 ms is 180 degrees angle.

- We can use the **map(angleDegrees, 0,180,600,2400)** function to map the rotation angle to pulse width. This function will return the width of pulse in **microseconds**. Here 0 is minimum angle with pulse width 600 microseconds, 180 is maximum angle with pulse width 2400 microseconds.

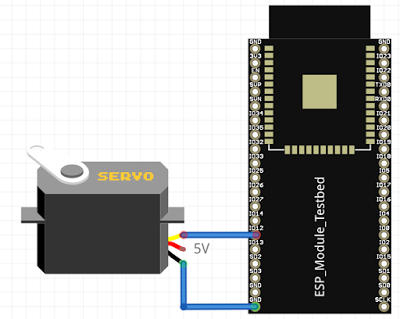
1. **Hardware**

We connect:

[Servo PWM - ESP32 GPIO 12]

[Server VCC - ESP32 5V]

[Server GND - ESP32 GND]



1. **Software**

**3.1 Web server**

You can refer [Demo 12: How to turn the Arduino ESP32 into a Web Server](http://www.iotsharing.com/2017/05/how-to-turn-esp32-into-web-server.html" \t "http://www.iotsharing.com/2017/06/_blank)

**3.2 Servo**

Create an Arduino project with code:

|  |
| --- |
| #include <WiFiClient.h>  #include <ESP32WebServer.h>  #include <WiFi.h>  #include <ESPmDNS.h>  /\* change your ssid and password here \*/  const char\* ssid = "Coffee Amy";  const char\* password = "amy12345";  ESP32WebServer server(80);  const int servoPin = 12;  int oldAngle = 0;  int PWM\_WIDTH = 20000;  /\* this array contains the web will be responded to client  it uses jquery for making GET request and processing slider UI control \*/  char res[900]=  "<!DOCTYPE html>\  <html>\  <head>\  <meta charset='utf-8'>\  <H1>iotsharing.com Servo</H1>\  <link href='https://code.jquery.com/ui/1.10.4/themes/ui-lightness/jquery-ui.css' rel='stylesheet'>\  <script src='https://code.jquery.com/jquery-1.10.2.js'></script>\  <script src='https://code.jquery.com/ui/1.10.4/jquery-ui.js'></script>\  <script>\  $(function() {\  $('#sliVal').html('Angle: 0');\  $('#slider').slider({\  orientation:'vertical',value:0,min: 0,max: 180,step: 5\  });\  $('#slider').slider().bind({\  slidestop: function(e,ui){\  $('#res').css('background','red');\  $('#sliVal').html('Angle: '+ui.value);\  $.get('/ang?val=' + ui.value, function(d, s){\  $('#res').css('background','green');\  $('#res').html(s);\  }); \  }\  });\  });\  </script>\  </head>\  <body>\  <div id='slider'></div></br>\  <div id='sliVal'></div>\  <div id='res'></div>\  </body>\  </html>";  void handleRoot() {  server.send(200, "text/html", res);  }  /\* this function map from angle to pulse width \*/  int servoPulse(int angleDegrees)  {  int pulseWidth = map(angleDegrees, 0,180,600,2400);  return pulseWidth;  }  /\* this function check the rotation angle  and trigger pulse accordingly\*/  void servoGo(int oldAngle, int newAngle)  {  int pulseWidth;  if(oldAngle == newAngle){  return;  }else if(oldAngle < newAngle){  /\* clockwise processing \*/  for (int i=oldAngle; i<=newAngle; i++){  /\* convert angle to pulse width us\*/  pulseWidth = servoPulse(i);  /\* trigger HIGH pulse \*/  digitalWrite(servoPin, HIGH);  /\* use delayMicroseconds to delay for pulseWidth \*/  delayMicroseconds(pulseWidth);  /\* trigger LOW pulse \*/  digitalWrite(servoPin, LOW);  /\* use delayMicroseconds to delay  for rest time (20000 - pulseWidth) \*/  delayMicroseconds(PWM\_WIDTH - pulseWidth);  }  }else if(oldAngle > newAngle){  /\* anti-clockwise processing \*/  for (int i=oldAngle; i>=newAngle; i--){  pulseWidth = servoPulse(i);  digitalWrite(servoPin, HIGH);  delayMicroseconds(pulseWidth);  digitalWrite(servoPin, LOW);  delayMicroseconds(PWM\_WIDTH - pulseWidth);  }  }  }  /\* this callback will be invoked when get servo rotation request \*/  void handleServo() {  //Serial.println(server.argName(0));  int newAngle = server.arg(0).toInt();  servoGo(oldAngle, newAngle);  oldAngle = newAngle;  server.send(200, "text/html", "ok");  }  void handleNotFound(){  String message = "File Not Found\n\n";  server.send(404, "text/plain", message);  }  void setup(void){    Serial.begin(115200);  WiFi.begin(ssid, password);  Serial.println("");  // Wait for connection  while (WiFi.status() != WL\_CONNECTED) {  delay(500);  Serial.print(".");  }  Serial.println("");  Serial.print("Connected to ");  Serial.println(ssid);  Serial.print("IP address: ");  Serial.println(WiFi.localIP());  if (MDNS.begin("esp32")) {  Serial.println("MDNS responder started");  }    server.on("/", handleRoot);  server.on("/ang", handleServo);  server.onNotFound(handleNotFound);  server.begin();  pinMode(servoPin, OUTPUT);  Serial.println("HTTP server started");  }  void loop(void){  server.handleClient();  } |