

Programming a web page to control the arm using the web serial API?

THE ARCHITECTURE AND DESIGN

The proposed block diagram of this robotic system is shown in Figure 1 which consists of an Arduino EPS33 board microcontroller and four digital servo motors driven by driver integrated circuit (IC), in addition to a WIFI router and volts-direct current (5 V-DC) power supply. The pick and place robotic arm has a robotic arm put on a fixed base. The pick and place robot utilize 4-motors for the operation of the structure. also, it contains an arm assembly with a selectable jaw or with a screwdriver, which can move upend down direction only. For, the motor controlling is using motor driver IC and Arduino EPS33 microcontroller. The processing signal is provided by a personal computer (PC) or mobile phone via html designed web page to the wireless router. The signal is wireless router. The signal is decoded in the controller when it is sent from the wireless router and suitable controlling signal is sent to actuators (servo engine) in the system.

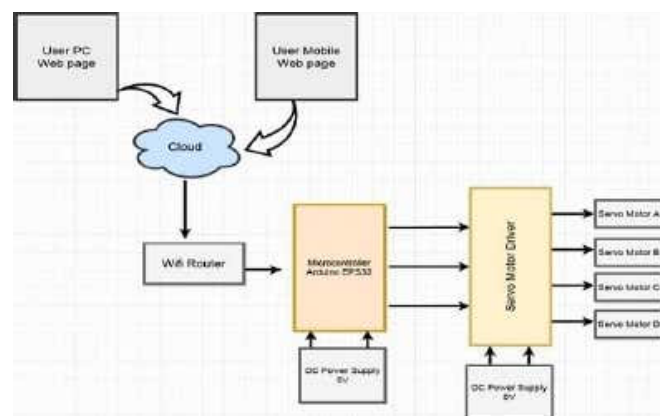


Figure 1. Block diagram of the proposed project

FLOWCHART OF THE IMPLEMENTED CODE

The Figure 2 describe the methodology for programming of the system. First, we are arising a server instance that obey for hypertext transfer protocol (HTTP) commands on port 80. This port is represented the default port for servers of web. In the platform of Arduino, the signal is transmitted from the router to be read on Arduino utilized the command (if client available). The receiver flag of the Arduino controller is monitored to discover every order is supplied to the controller next mission is decoded the instruct the flag of receiver is high.

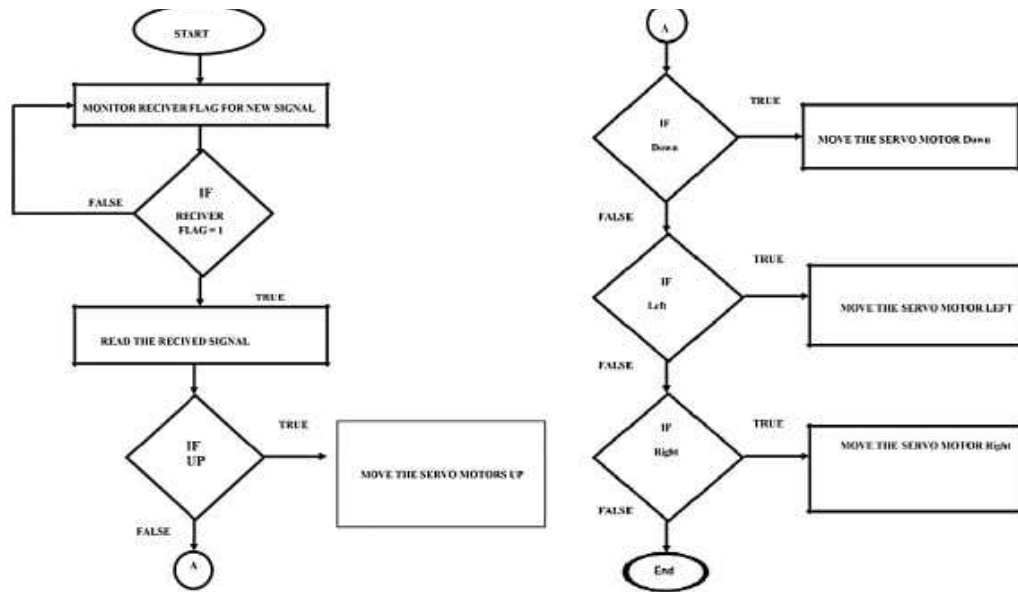


Figure 3. Flowchart of the Arduino source code

SIMULATION AND RESULTS DISCUSSION

The robotic arm is tested in Proteus software before the construction as shown in Figure 4. The arm is constructed using 4-servo motors with Arduino. After the circuit diagram, the simulation shows a good operation by the servo motors. Then the arm is ready to be installed for running and performs the duty which is for pick and place or to be used as screw driver for tightening the screws in the appropriate positions. Now we will be talking about the code of the Arduino which let us to control the servo movements by just sending its position in degree value via the web page designed in html language. The servos control lines are:

```

const int Servo A=13;
const int Servo B=14;
const int Servo C=15;
const int Servo D=16;

```

Here we are used to specify the four servos to the pins (13, 14, 15, 16) respectively. You can use other pins as its required. Practically we used ESP32 Arduino for wirelessly, connecting the robotic arm and control on it and movement the dashboard in the web page as shown in the Figure 5. The command which is the: ledcSetup (1, 90, 8) in Arduino, in the setup function, is going to setup the PWM with the ledcSetup function, which receives as input the PWM channel, the final servo the increment in each command time of position and the duty cycle resolution, i.e., the slider movement.

The function ledcAttachPin (Servo A, pin) is to attach the servo (A) to the first specified pin (channel) which was the pin (13). This command is repeated to the other servos as being as:

```

ledcSetup (1, 90, 8); for servos B
ledcSetup (2, 90, 8); for servos C

```

ledcSetup (3, 9, 8); for servos D

The number (8) indicate increment in the servo movement in each step the slider of the dashboards ends. The command serial begins (115200) doesn't actually print anything but It is utilized to set up the communication speed in bits/ second instead of it initializes the connection of serial at 115200 bits per second. To obtain any type of clear data, it is using the same order of serial connection speed by adjust both sides of the serial connection (i.e., the computer and the Arduino). The data are distorting if there doesn't have symmetric in the speed for both the two systems. When the number is making smaller (e.g., serial. begin (300)) the Arduino will be transmitting the data in slow. While rising it, say to 115200 will transmit the data fast. Both the transmitting and the receiving systems need to agree on the speed must to use: the serial program for your computer's, as the window of Arduino serial monitor will let you adjust the speed at that the data is received by your computer but the speed is selected from the common speeds only: 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600 and 115200 bit per sec. Now we have to enter the web page environment which begins with the log in gate. This can be done using the command (if (header.indexOf ("") >= 0) {}), this will match the user name and password you have specified to the web page authentication which is shown in Figure 6. If the user's name and password is wrong the web page in the Figure 5 will pop up.

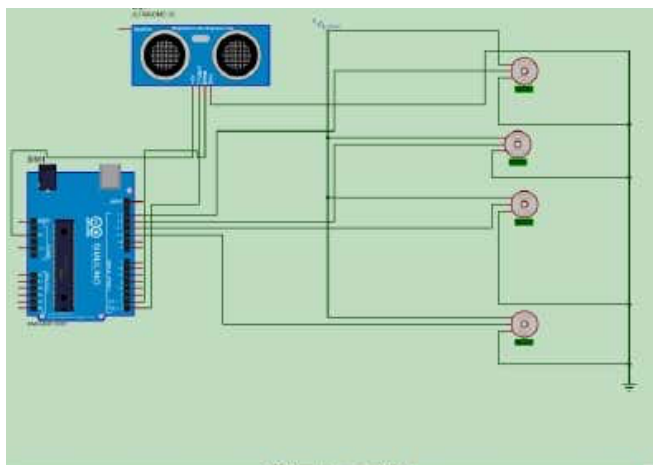


Figure 4. proteus simulation for robotic arm

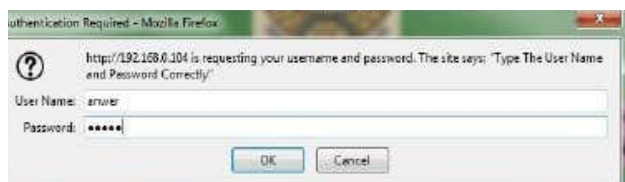


Figure 6. Web page authentication



Figure 9. web page dashboard



Figure 10. Web page for wrong log in

After the entering the web page the control section will be in the slider bar, and each servo motor has its own slider bar to be controlled separately by the user. After the entering the web page the control section will be in the slider bar, and each servo motor has its own slider bar to be controlled separately by the user as shown below for servo A and is repeated for the other three ones:

```
client. println("<p>Servo A: <span id=\"servoPosA\"></span></p>");
client. println ("<input type=\"range\" min=\"0\" max=\"180\" class=\"slider\"
id=\"servoSliderA\"onchange=\"servo (this.value,'A
')\" value=\"\" + valueString + "\"/>");
client. println("<script>"); //send A valueclient. println ("var sliderA = document.
getElementById(\"servoSliderA\");");
client. println ("var servoPA = document. getElementById(\"servoPosA\");
servoPA.innerHTML = sliderA.value;");
```

Our servo motors are installed by the screws on board shown in the Figure 11. They are tightened depending on the angular rotation of servo motors

The rotation degree is to be from 0 to 180 degrees for the base servo which is to rotate the arm. This base handle the remaining three servos to make the bending and the pick and place of things. The

`println()` method writes data to the serial port. This is often helpful for looking at the data a program is producing or to write data to other devices connected to the serial port. The

`println` method works like `print`, but sends a new line character for each call to the function. Data can either be a single

`int`, `float`, `byte`, `long`, `char`, `char[]`

, String or a number in decimal (DEC), hexadecimal (HEX), octal (OCT), or binary (BIN) base [24]. The authentication for the log in page can be done using the line command: client.

```
println
("HTTP/1.1 401 Unauthorized").
```

The 401 Unauthorized error is an HTTP status code that means the page you were trying to access cannot be loaded until you first log in with a valid user identification (ID) and password

If you've just logged in and received the 401 Unauthorized error, it means that the credentials you entered were invalid for some reason

401 Unauthorized error messages are often customized by each website, especially very large ones, so keep in mind that this error may present itself in more. The 401 Unauthorized error displays inside the web browser window, just as web pages do. Like most errors like these, you can find them in all browsers that run on any operating system [25].

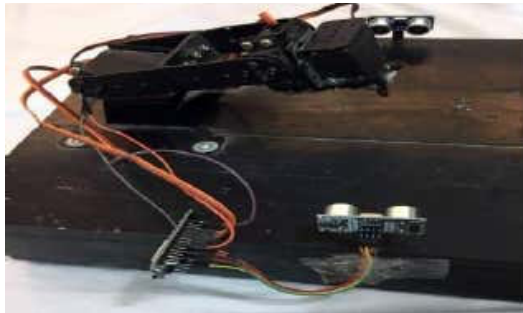


Figure 8. robotic arm servos construction

Ultrasonic sensor

Pulse-width modulation (PWM) applications can be found in a number of applications such as communication, servo motor control, voltage regulation, and power conductivity [26], [27]. This paper gives in details the way to measure the PWM display by get the benefit of Arduino. Moreover, how the ultrasonic sensor (which can be used for telemetry) that can make its job with the help of the PWM is an ultrasound sensor that can remotely measure the physical quantities by using the ultrasound waves [28], [29]. The human ear has not the ability to detect ultrasound. This unit has the ability to correctly measure the distance from 2 to 400 cm (1 to 157 inches) and easily be connected together with the Arduino microcontroller. HC-SR-04 consumes less energy that makes it a good choice to work in automatic and automatic control systems [30]. The principle of operation of this ultrasonic sensor (HC-SR-04) is similar to the of object detection system of a bat which is shown in Figure 9. Ultrasound sensor sends a 40 kHz wireless frequency to the air at 343 meters per second and the receiver get the reflected signal arrived from the reflector body. Then this distance between the received (Rx) and the transmit (Tx) can be simply calculated by the looking at the time taken by the ultrasound to send from the transmitter and reflect back to the receiver which is reflected by the receiver. If we be in the interested calculation to this separating distance between

the object and the transmitter, the sensor module is adjusted to do its job. The result of these calculations is depending on the pulse which has a width that is related to the separating measured distance.

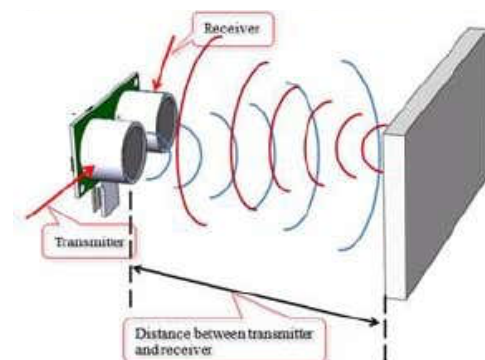


Figure 9. HC-SR04 module operation

The HC-SR04 unit diagram has four pins using them in its operation as shown in Figure 10. At the first, it is very useful to study the functionality of the pins of this module sensor.

- Vcc can be supplied up to 5V.
- Trig pin input is for getting a pulse with a time width of 10 micro seconds to generate the 40 KHz ultrasonic signals to be transmitted into the air from the Tx.
- Pin Echo is the pin for receiving the Echo pulses arrived to the module from the objects to calculate the width of the pulse and then calculate the distance.
- GND is the ground pin.

The characteristics curve of this module is that it generates a series of pulses signals with at least 10 micro second then it will be fed to the Trig pin.

After that by about 1.5 milliseconds, it will receive a pulses signal which arrive to the echo pin which has signal width ranges start from 100 microseconds to 200 milliseconds. If the width of the signal is greater than the range above, the sensor module will not be able to detect the object. Therefore, it must wait for about 10 milliseconds and retransmit another Trig signal again as shown in Figure 11.

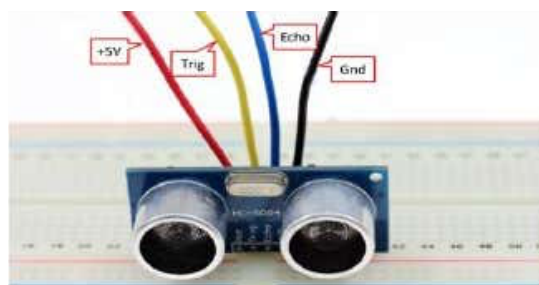


Figure 10. HC-SR04 module pin out

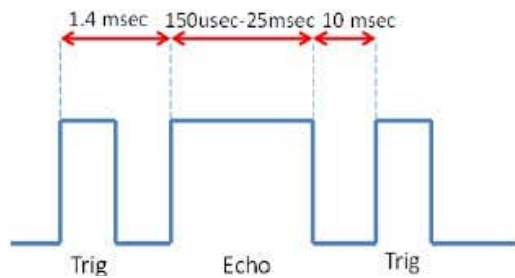


Figure 11. HC-SR04 Trig and Echo signal

In the Arduino program, we can calculate the object distance which stand in the front of the sensor. At first, the sensor will send a signal at an instance and staying waiting for the answer (reflecting) signal from the standing object or body, i.e. the echo again at another particular instance. The Pinging or transmit and receive is about a voice wave that a human can't hear for this sensor is called "ultrasound". In mathematical details, the sensor programmed to send a signal at time t_1 and receives a reflected one at time t_2 .

For that, if we know the speed of the sound wave, then the time difference $\Delta t = t_2 - t_1$

will give the idea of the distance between the sensor and the object.

For example,

if the time difference was $\Delta t = 500$ microseconds, then it will be known that soundwave takes 250 microseconds to receive and hit the object and another 250 microseconds to be returned. Therefore, for accurate operation, the robotic arm equipped with ultra-sonic sensor (HC-SR04) to be used in the accurate approximation between this arm and the aimed things that to be picked and placed, or to be tightened by the screw driver installed at the top which is rotate automatically when the splitting space is zero according to the ultra-sonic sensor reading. Ultra-sonic indicate to a voice wave with a frequency is over 20 kHz, which is greater than the upper human hearing level. Since the (HC-SR04) is depends on the ultra-sonic waves. Then the movement of this wave will be affected by the temperature of the environment it works. Theoretical calculation of the distance the ultrasonic wave travel is by the (1):

$$\text{Distance} = v \times t$$

where v , is the wave velocity (about 343 m/s) in the air at (20°C). Since the ultra-sonic wave is affected by the temperature of the ambient of working, there must be a modification on the (1) to get the accurate position of the object the robotic arm deals with. Depending on the system specification below, we will develop a modification on the equation (1) to get the expected results:

Frequency of the oscillator = 20 Mhz, Cycle = 50 ns, 20 Mhz = $20,000,000$ Hz and timer count = $1,000,000$ At 20°C sound speed = 34300 cm/sec so within $1,000,000$ the distance will be

$(1,000,000 \times 34300 \text{ cm/sec}) = 34,300,000,000 \text{ cm}$ per count, but sound distance is twice (because the come and go back), so relationship becomes $34,300,000,000 / 2 = 17,150,000,000 \text{ cm}$ per count.

The sound speed within $1,6 \mu s$ the distance will be $(1,6 * 0,000001 * 34000 = 0,0544 \text{ cm}$ per count), but sound distance is twice (because the come and go back), so relationship becomes $0,0544/2 = 0,0272 \text{ cm}$ per count. To start the measure, the device needs a pulse of $1 \mu s$ on trigger input then send (itself) a burst of 8 periods of 40 KHz (so during $200 \mu s$) then echo output signal goes to 1 status and returns to 0 status when echoes is back timer measures this duration. To avoid to ear the receiver when emitter send the burst salve of 40 kHz , timer start must begin to count after $(1 + 210 = 211 \mu s)$. So minimum distance (theatrical) is $211 * 0,0272 = 5,7392 \text{ cm}$ ($0,0272$ because only one way), and add a little offset for calibration here offset is $1,093 \text{ cm}$. Then the final equation will be. Figure 12 shows the simulation of the effect of the temperature on the ultra-sonic speed which results in drift in the actual distance (Y).

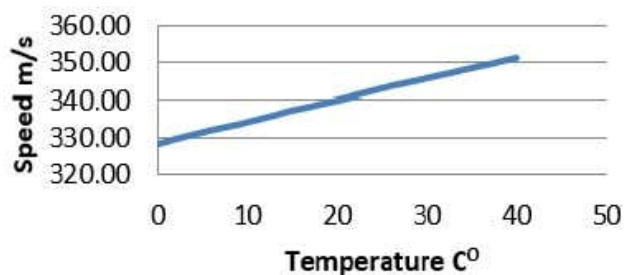
$$\text{distance} = \text{time} * 0,0272 + 1,093$$


Figure 12. Effect of the temperature on the ultrasonic speed

CONCLUSION

The robotic arm is designed and simulated using Proteus software by take the advantages of the Arduino model ESP32 which can be connected to the wireless router. The command is sent from the designed web page to this arm so as to pick or place objects or for tightening the screws of the devices or objects. We can use the speed of sound also (pace of sound) for this purpose. The pace of sound can be equal to $(1/\text{Speed of Sound} = 1/0,000001 = 29,1 \text{ s/cm})$. Therefore, the equation which is used to calculate the distance will become: $D = (\Delta t / 2) / \text{of sound wave pace}$.

This robotic arm here takes the benefits of the ultra-sonic sensor model (HC-SR-04) for the best accuracy of working for good proximity between the arm and the objects and to specify the operation area of the arm. Since the sensor depends on the ultra-sonic waves, there will be an effect on its speed by the ambient working temperature, which made the need to the modification of the model distance equation. Finally, the goal of the IOT is can be done using the real internet protocol (IP) to connect to the external world of networks (INTERNET).

