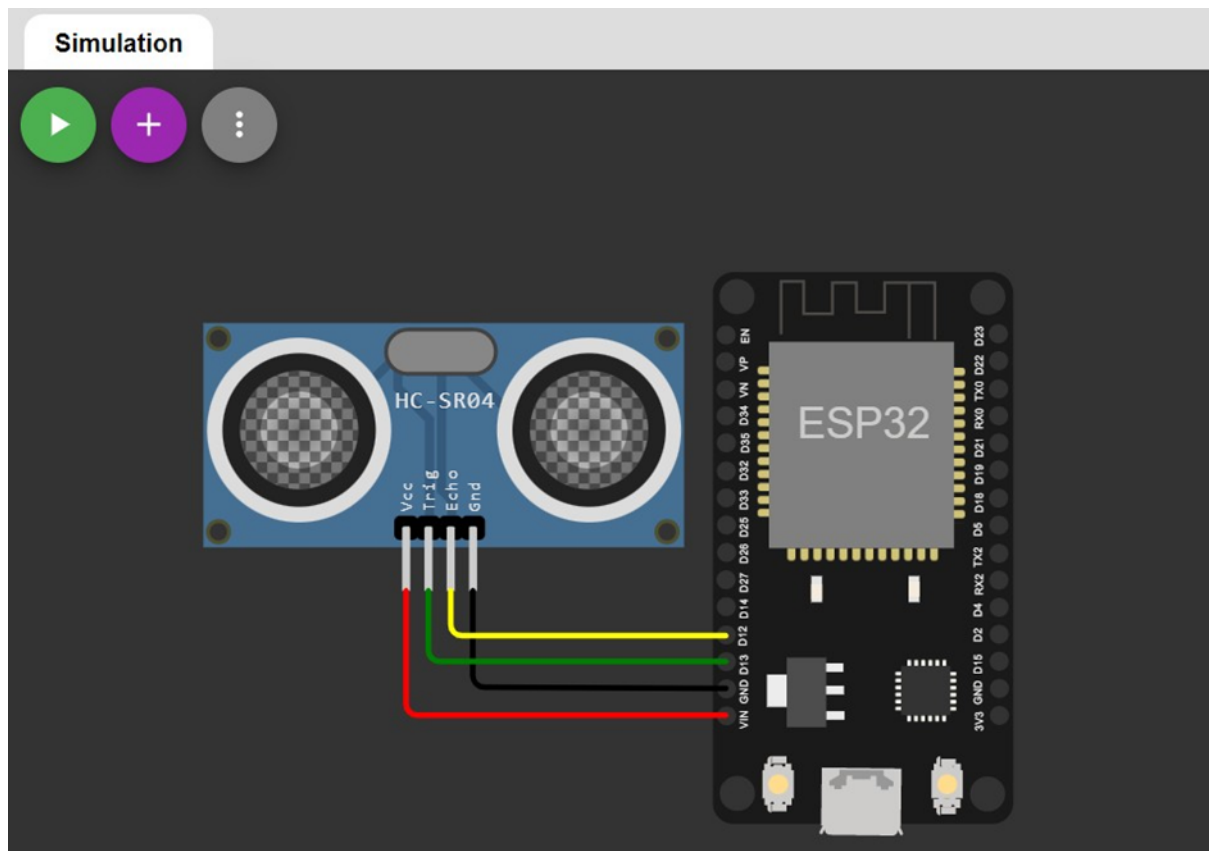


# ASSIGNMENT 4

INPUT:



## CODE

```
#include <stdio.h>

#include <stdlib.h>

#include <freertos/FreeRTOS.h>

#include <freertos/task.h>

#include <esp_err.h>

#include "ultrasonic.h"

#define ECHO_GPIO 12

#define TRIGGER_GPIO 13
```

```
#define MAX_DISTANCE_CM 500 // Maximum of 5 meters
```

```
void ultrasonic_test(void *pvParameters)
```

```
{    float  
distance;
```

```
    ultrasonic_sensor_t sensor = {  
.trigger_pin = TRIGGER_GPIO,  
  
    .echo_pin = ECHO_GPIO  
};
```

```
    ultrasonic_init(&sensor);
```

```
    while (true) {        esp_err_t    res        =        ultrasonic_measure(&sensor,  
MAX_DISTANCE_CM, &distance);
```

```
        if (res == ESP_OK) {            printf("Distance:  
%0.04f m\n", distance);  
        } // Print error            else {  
printf("Error %d: ", res);            switch (res) {  
        case ESP_ERR_ULTRASONIC_PING:  
            printf("Cannot ping (device is in invalid state)\n");            break;  
        case ESP_ERR_ULTRASONIC_PING_TIMEOUT:  
            printf("Ping timeout (no device found)\n");  
break;  
  
        case ESP_ERR_ULTRASONIC_ECHO_TIMEOUT:
```

```

        printf("Echo timeout (i.e. distance too big)\n");
break;                default:

        printf("%s\n", esp_err_to_name(res));

    }

}

vTaskDelay(pdMS_TO_TICKS(500));

}

}

```

```

void app_main()

{
    xTaskCreate(ultrasonic_test, "ultrasonic_test",

configMINIMAL_STACK_SIZE * 3, NULL, 5, NULL);

}

```

## OUTPUT:

The image shows a development environment with two main windows. The left window displays the source code for an ESP32 application using the Arduino IDE. The code defines an ultrasonic sensor, initializes it, and enters a loop that measures distance and prints the result or an error message. The right window shows a simulation of the hardware. It includes a visual representation of the ESP32 microcontroller and an HC-SR04 ultrasonic sensor connected by wires. A 'Simulation' window is open, showing a slider for the sensor's distance, currently set to 151cm. Below the simulation, a list of distance measurements is displayed, showing values like 4.0564 m, 2.1295 m, and 1.5312 m.

**main.c**

```

15 float distance;
16
17 ultrasonic_sensor_t sensor = {
18     .trigger_pin = TRIGGER_GPIO,
19     .echo_pin = ECHO_GPIO
20 };
21
22 ultrasonic_init(&sensor);
23
24 while (true) {
25     esp_err_t res = ultrasonic_measure(&sensor, MAX_DISTANCE_CM, &distance);
26
27     if (res == ESP_OK) {
28         printf("Distance: %0.04f m\n", distance);
29     } // Print error
30     else {
31         printf("Error %d: ", res);
32         switch (res) {
33             case ESP_ERR_ULTRASONIC_PING:
34                 printf("Cannot ping (device is in invalid state)\n");
35                 break;
36             case ESP_ERR_ULTRASONIC_PING_TIMEOUT:
37                 printf("Ping timeout (no device found)\n");
38                 break;
39             case ESP_ERR_ULTRASONIC_ECHO_TIMEOUT:
40                 printf("Echo timeout (i.e. distance too big)\n");

```

**Simulation**

Editing Ultrasonic Distance Sensor  
Distance: 151cm

Distance: 4.0564 m  
Distance: 4.0564 m  
Distance: 2.1295 m  
Distance: 1.7036 m  
Distance: 1.5310 m  
Distance: 1.5312 m  
Distance: 1.5312 m

main.cdiagram.jsonultrasonic.hultrasonic.cesp\_idf\_lib\_helpers.h

Library Manager

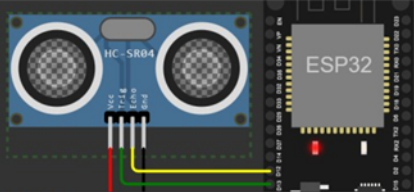
```
15 float distance;
16
17 ultrasonic_sensor_t sensor = {
18     .trigger_pin = TRIGGER_GPIO,
19     .echo_pin = ECHO_GPIO
20 };
21
22 ultrasonic_init(&sensor);
23
24 while (true) {
25     esp_err_t res = ultrasonic_measure(&sensor, MAX_DISTANCE_CM, &distance);
26
27     if (res == ESP_OK) {
28         printf("Distance: %0.04f m\n", distance);
29     } // Print error
30     else {
31         printf("Error %d: ", res);
32         switch (res) {
33             case ESP_ERR_ULTRASONIC_PING:
34                 printf("Cannot ping (device is in invalid state)\n");
35                 break;
36             case ESP_ERR_ULTRASONIC_PING_TIMEOUT:
37                 printf("Ping timeout (no device found)\n");
38                 break;
39             case ESP_ERR_ULTRASONIC_ECHO_TIMEOUT:
40                 printf("Echo timeout (i.e. distance too big)\n");
```

Simulation

00:10:517 76%

Editing Ultrasonic Distance Sensor

Distance: 337cm



Distance: 3.4176 m  
Distance: 3.4176 m  
Distance: 3.4176 m  
Distance: 3.4174 m  
Distance: 3.4174 m  
Distance: 3.4174 m  
Distance: 3.4174 m