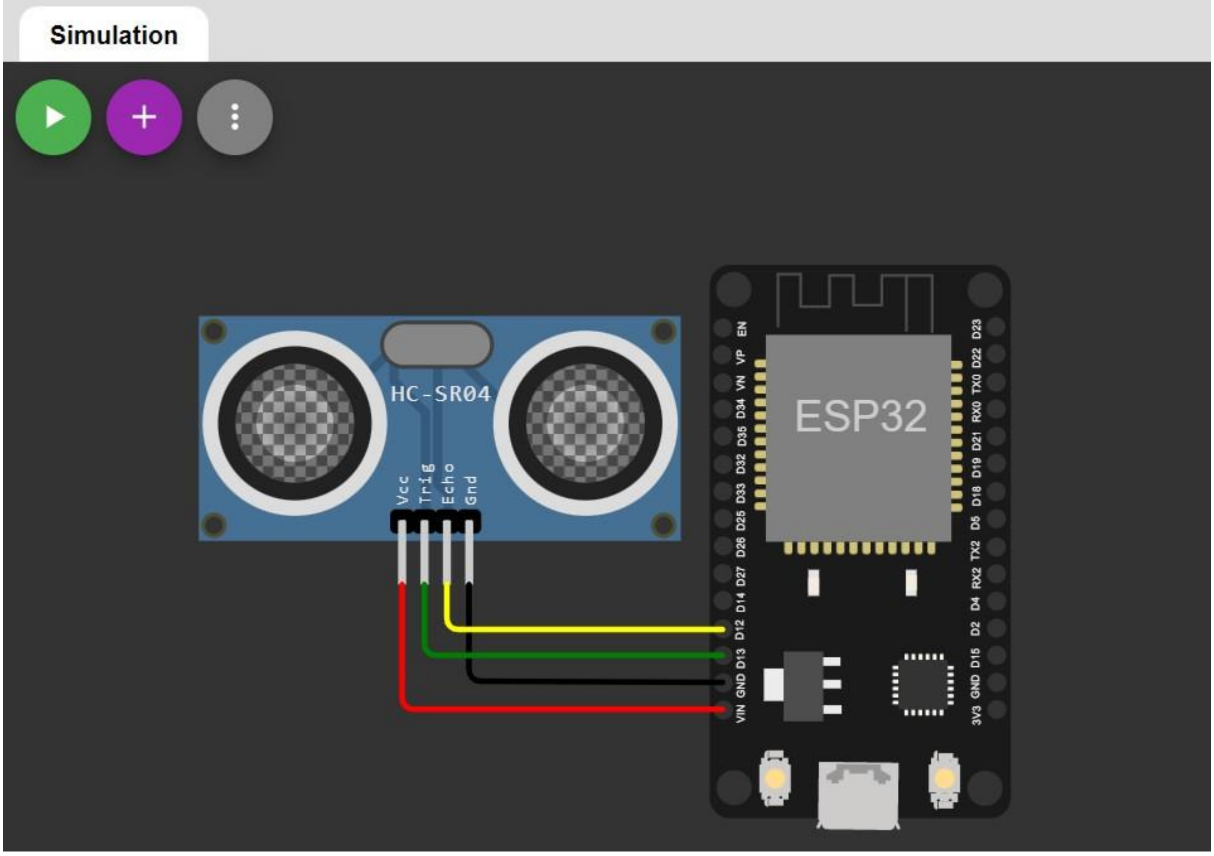
Assignment -4

Assignment Date	25 September 2022
Student Name	Mr. A.G.Abishek
Student Roll Number	910619104003
Maximum Marks	4 Marks

Input:

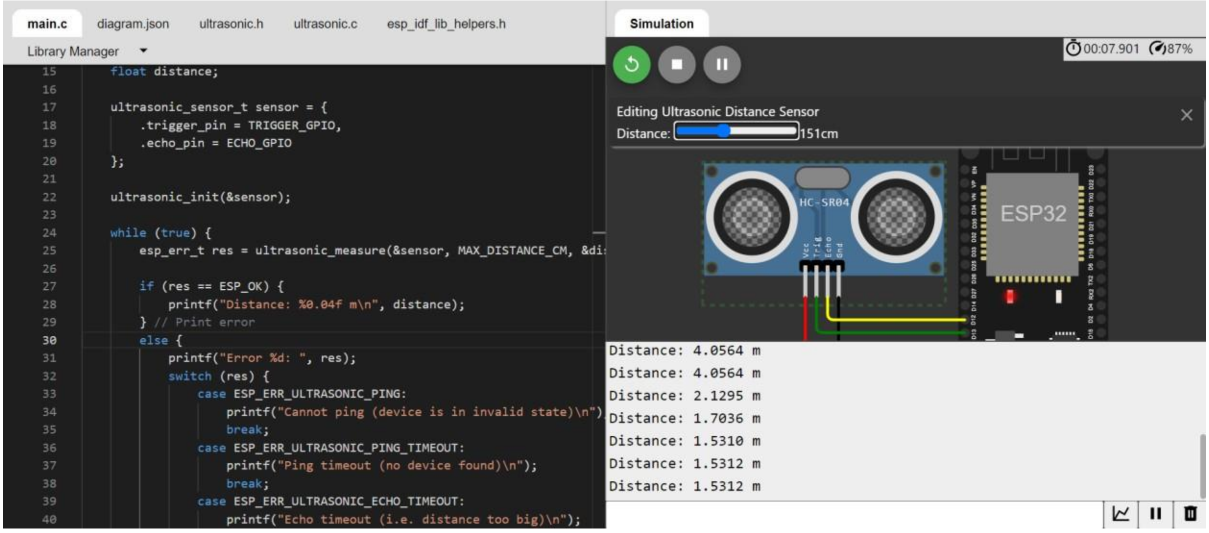


CODE:

```
#include <stdio.h>
#include <stdbool.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:

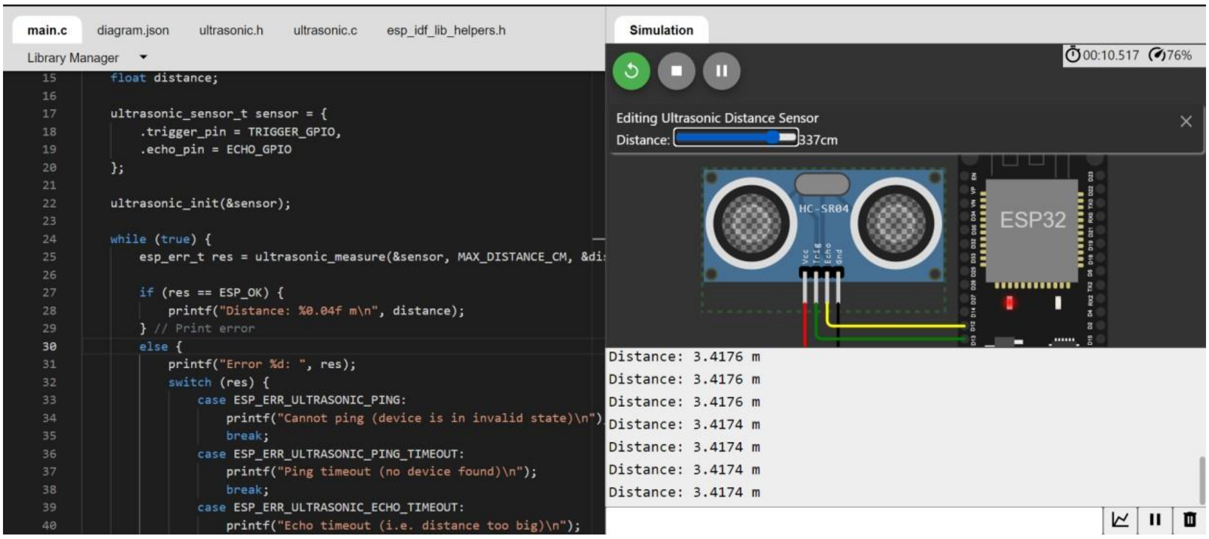


```
    float distance;

    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, &di

        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");
```

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
```



```c
    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, &di

        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");
```

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
```