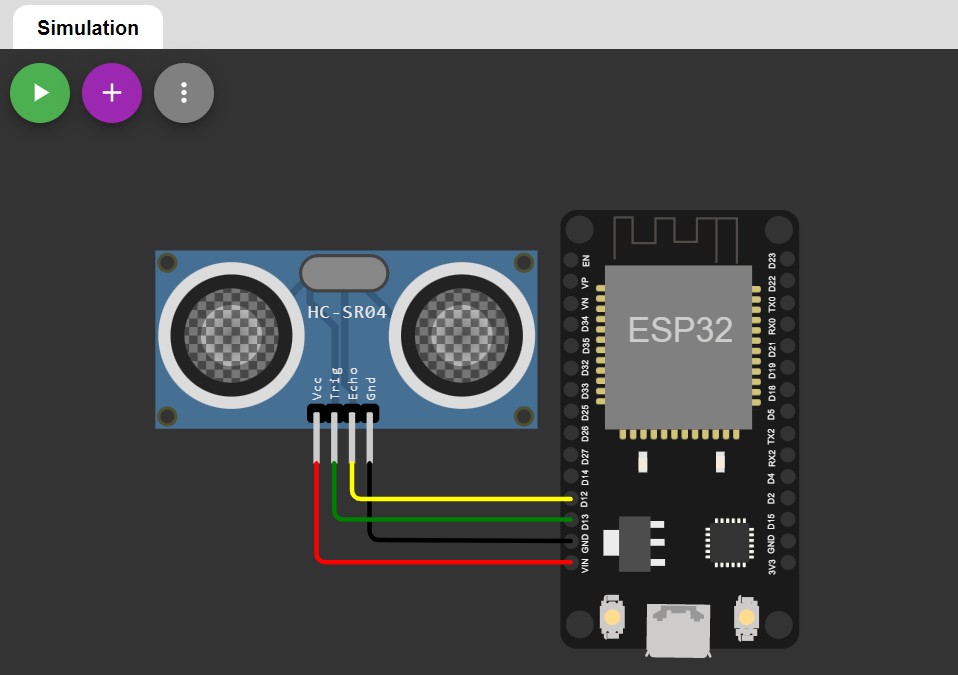
**Assignment -4**

|  |  |
| --- | --- |
| 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:

