

ASSIGNMENT – 4

NAME	SHYLENDRAN R
PROJECT NAME	IoT Based Smart Crop Protection System For Agriculture
TEAM ID	PNT2022TMID26521

1. Write code and connections in wokwi for the ultrasonic sensor. Whenever the distance is less than 100 cms send an "alert" to the IBM cloud and display in the device recent events. Upload document with wokwi share link and images of IBM cloud.

```
#define TRIGGER_GPI
#include <stdio.h>
#include
<stdbool.h>
#include
<freertos/FreeRTOS.h>
#include <freertos/task.h>
#include <esp_err.h>
#include "ultrasonic.h"#define ECHO_GPIO O 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);
}

```

```
        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:

WOKWI SAVE SHARE esp-idf-ultrasonic by arcostasi Docs SIGN UP

main.c diagram.json ultrasonic.h ultrasonic.c esp_idf_helpers.h Library Manager

```
1 #include <stdio.h>
2 #include <stdbool.h>
3 #include <freertos/FreeRTOS.h>
4 #include <freertos/task.h>
5 #include <esp_err.h>
6
7 #include "ultrasonic.h"
8
9 #define ECHO_GPIO 12
10 #define TRIGGER_GPIO 13
11 #define MAX_DISTANCE_CM 500 // Maximum of 5 meters
12
13 void ultrasonic_test(void *pvParameters)
14 {
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
```

Simulation 00:04.964 25%



Distance: 4.0564 m
Distance: 4.0564 m
Distance: 4.0566 m
Distance: 4.0566 m
Distance: 4.0564 m
Distance: 4.0564 m
Distance: 4.0564 m

🔍 ⏸ 🗑