



Universidad autónoma de baja california

Ingeniería en computación

Internet de las cosas

Lab-taller: envió de alerta por correo electrónico

Aguilar Noriega Leocundo

Garcia Chaves erik

17 de febrero del 2026

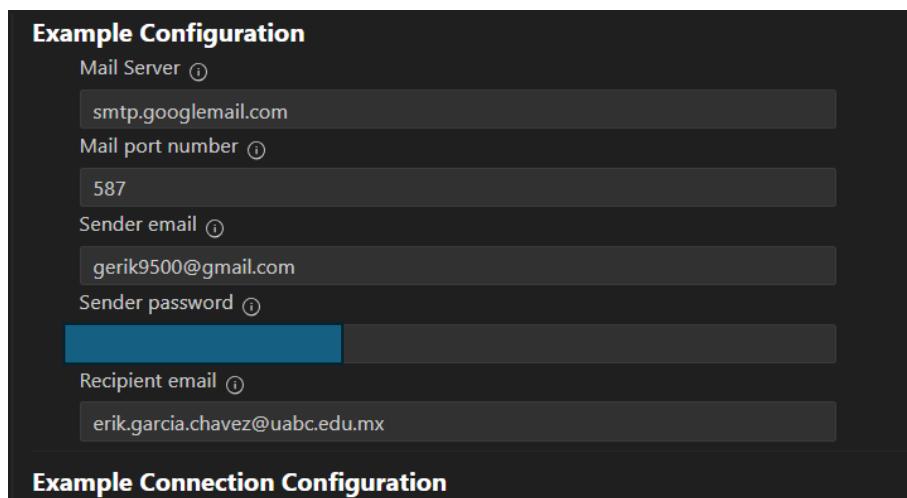
Descripción:

- Realizar un programa que envió alertas por medio de correo electrónico de Gmail, haciendo uso del ESP32, el protocolo SMTP, el programa deberá de enviar un mensaje al usuario cuando detecte un **short press** en la placa.

Desarrollo:

Para el desarrollo de esta practica se utilizo el ejemplo por defecto del protocolo SMTP para el envio de correo electrónico por medio del ESP32, solo que se tuvieron que cambiar algunas configuraciones, empezando con las credenciales de la cuenta Gmail de donde se enviara el correo, para eso necesitamos crear una **contraseña de aplicación** en la cuenta Google de donde se estarán enviando los correos, no es recomendable usar la contraseña real de nuestra cuenta Google.

Después ingresamos nuestras credenciales, dentro de **SDK configuration**:



En donde detallamos quien envía el correo, quien lo recibe y la contraseña del que lo envía.

Después necesitamos escribir el código que configura el GPIO junto con su tarea que está en espera que se presione el botón en un periodo de tiempo no mayor a 50 mS, esto le dice al programa que es un **short press** por lo que por medio de un **grupo de eventos** podemos decirle a la tarea encargada de enviar el mensaje que puede hacerlo para después “dormirse” la tarea por 1 minuto ya que entre mensaje a mensaje debe haber 1 minuto de diferencia.

```

void GPIO_INIT(void){
    gpio_reset_pin(BUTTON);
    gpio_set_direction(BUTTON, GPIO_MODE_INPUT);
    // con pull-down, siempre estara leyendo 0, cuando presione el btn entonces leerá 1
    gpio_pullup_dis(BUTTON);
    gpio_pulldown_en(BUTTON);
    //en este punto el GPIO 18 está configurado como entrada, habilitando pull-down, cuando se presione
}

// tarea que mapea el GPIO
void task_input(void *params){

    uint64_t current_time;

    while(1){
        current_time = esp_timer_get_time();
        int current_level = gpio_get_level(BUTTON);

        if(current_level == 1){

            vTaskDelay(pdMS_TO_TICKS(50)); // esperamos 50 ms para ver si es un botón presionado

            ESP_LOGI(TAG, "BOTÓN PRESIONADO - Envío señal");
            // ocurrió un evento que presionó el botón
            // este solo detecta que se presionó el botón.
            LAST_PRESS = current_time; // verificamos que se haya presionado el botón.
            xEventGroupSetBits(event_button, presionado);
            vTaskDelay(pdMS_TO_TICKS(200)); // para un antirebote
        }

        vTaskDelay(pdMS_TO_TICKS(10));
    }
}

```

El programa una vez cargado espera a que el botón sea presionado:

```

D I (3200) wifi:state: assoc -> init (0x2a0)
I (3200) wifi:new:<6,0>, old:<6,0>, ap:<255,255>, sta:<6,0>, prof:1, snd_ch_cfg:0x0
P I (3200) example_connect: Wi-Fi disconnected 203, trying to reconnect...
I (5620) example_connect: Wi-Fi disconnected 205, trying to reconnect...
E I (8040) wifi:new:<6,0>, old:<6,0>, ap:<255,255>, sta:<6,0>, prof:1, snd_ch_cfg:0x0
C I (8040) wifi:state: init -> auth (0xb0)
I (8040) wifi:state: auth -> assoc (0x0)
M I (8050) wifi:state: assoc -> run (0x10)
I (8070) wifi:connected with INFINITUMF4AF, aid = 15, channel 6, BW20, bssid = d8:6d:17:90:8f:48
I (8070) wifi:security: WPA2-PSK, phy: bgn, rssi: -63
I (8070) wifi:pm start, type: 1
b
I (8070) wifi:dp: 1, bit: 102400, li: 3, scale listen interval from 307200 us to 307200 us
c I (8130) wifi:AP's beacon interval = 102400 us, DTIM period = 1
I (9580) example_connect: Got IPv6 event: Interface "example_netif_sta" address: fe80:0000:0000:0000:024b:12ff:fe8e:b988
, type: ESP_IP6_ADDR_IS_LINK_LOCAL
I (11220) esp_netif_handlers: example_netif_sta ip: 192.168.1.93, mask: 255.255.255.0, gw: 192.168.1.254
I (11220) example_connect: Got IPv4 event: Interface "example_netif_sta" address: 192.168.1.93
I (11220) example_common: Connected to example_netif_sta
I (11230) example_common: - IPv4 address: 192.168.1.93,
I (11230) example_common: - IPv6 address: fe80:0000:0000:0000:024b:12ff:fe8e:b988, type: ESP_IP6_ADDR_IS_LINK_LOCAL
I (11240) main_task: Returned from app_main()
I (11580) example_connect: Got IPv6 event: Interface "example_netif_sta" address: 2806:0290:8800:cc1b:024b:12ff:fe8e:b98
8, type: ESP_IP6_ADDR_IS_GLOBAL

```

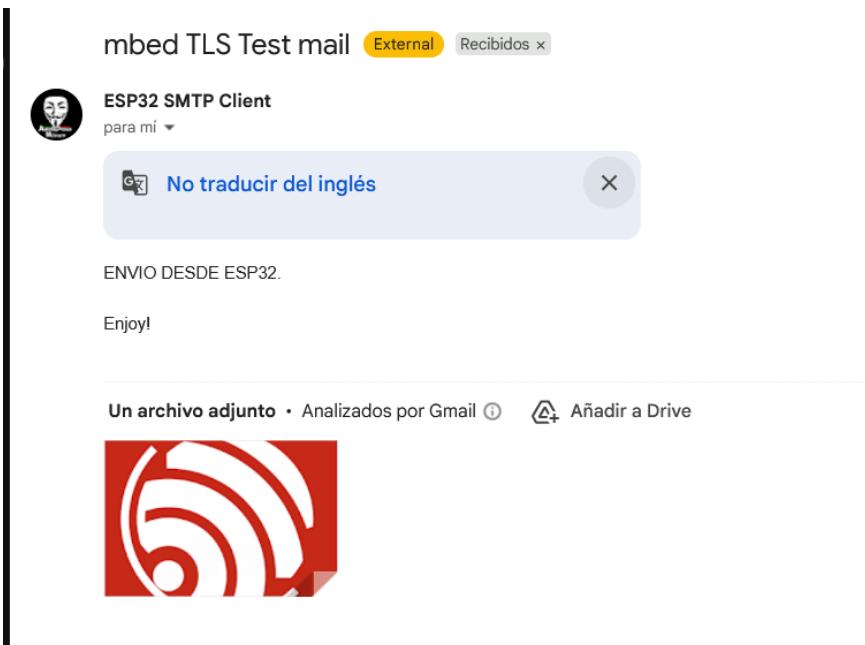
Una vez presionado enviará el mensaje e indica que se presionó

```
250=smtp.googlemail.com at your service, [107.230.232.150]
250-SIZE 35882577
250-8BITMIME
250-STARTTLS
250-ENHANCEDSTATUSCODES
250-DES
250-PIPELINING
250-CHUNKING
250 SMTPUTF8
I (37790) smtp_example: Writing STARTTLS to server...

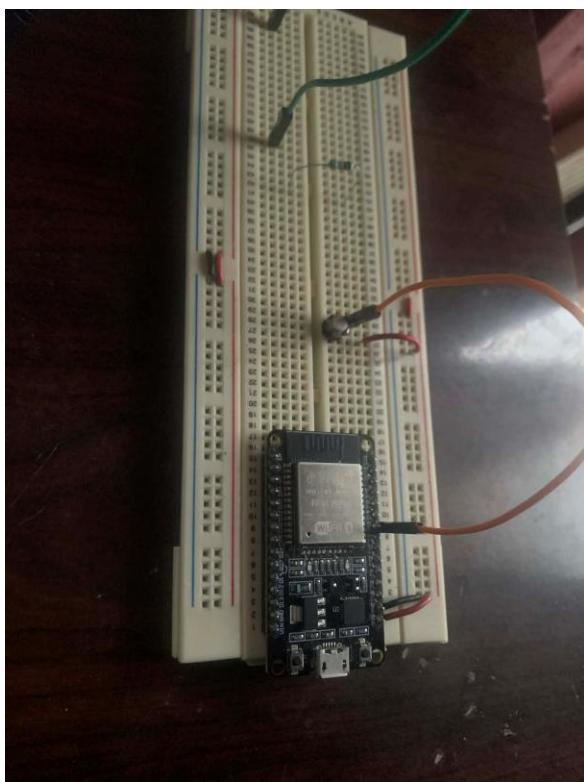
220 2.0.0 Ready to start TLS
I (37840) smtp_example: Performing the SSL/TLS handshake...
I (39150) esp-x509-crt-bundle: Certificate validated
I (40050) smtp_example: Verifying peer X.509 certificate...
I (40050) smtp_example: Certificate verified.
I (40050) smtp_example: Cipher suite is TLS-ECDHE-ECDSA-WITH-AES-128-GCM-SHA256
I (40050) smtp_example: Authentication...
I (40060) smtp_example: Write AUTH LOGIN
I (40120) smtp_example: Write USER NAME
I (40170) smtp_example: Write PASSWORD
I (40390) smtp_example: Write MAIL FROM
I (40440) smtp_example: Write RCPT
I (40500) smtp_example: Write DATA
I (40690) smtp_example: Write Content
I (41920) smtp_example: Email sent!

I (46330) wifi:<ba-add>idx:1 (ifx:0, d8:6d:17:90:8f:48), tid:6, ssn:5, winSize:64
I (50880) smtp_example: BOTON PRESIONADO - Enviando señal
```

El mensaje que envía es el por default solo le cambie que digiera **ENVIADO DESDE ESP32**



Círculo armado:



Conclusión:

En la practica se pudo ver algo que se acerca mucho a lo que ocurriría con algún dispositivo real, algo que este interactuando con el entorno y envía notificaciones, “interactúa” con el usuario, un poco a medias ya que solo hay un solo canal, pero esto puede dar mas contexto de lo que esta pasando con el micro o algún artefacto comercial.