Documentación Técnica ESP32-Flasher

Version: 1.0

Date: 2024-06-10

Índice

1 - Control de Versiones

Versión	Fecha	Responsable	Detalles
1.0	2024-06-10	Salvador López Criado	Documento público. Operativo con formato

2 – Introducción

En esta documentación técnica se recogerán los pasos importantes a tener en cuenta de cara a modificar internamente el código que se publica en este repositorio.

Este documento tiene previsión de seguir actualizando constantemente, sin vista de cesar a corto plazo.

Para más información acerca de futuras actualizaciones o dudas, contactar con Salvador López Criado

3 – Componentes y Herramientas necesarias.

Para la utilización de está aplicación será necesario lo siguiente.

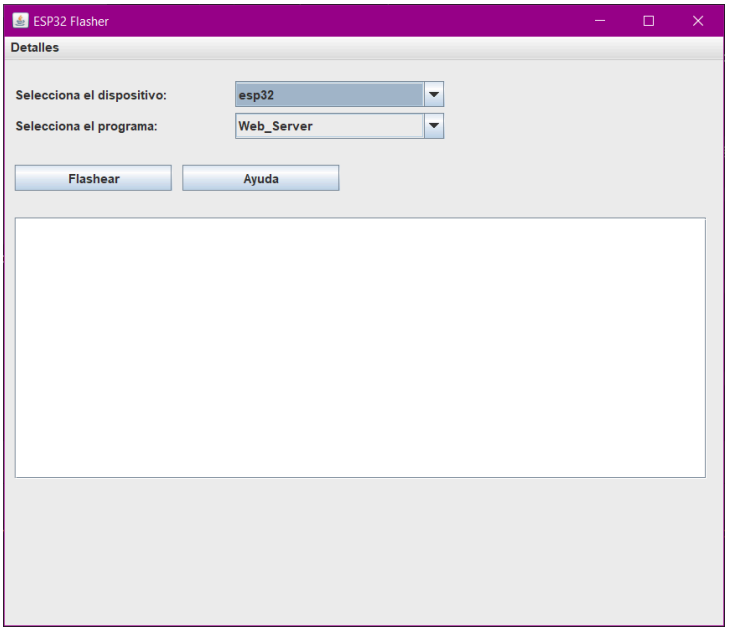
- Java
- Microcontrolador ESP32 (Wroover o Wroom)
- Cable MicroUSB
- Python

Para la modificación y compilación de los programas, será necesario las siguientes herramientas:

- Espressif-IDF
- Editor de textos cualquiera (Recomendado Visual Studio Code).
- Arduino-IDE
- JLink

4 – Primeros pasos

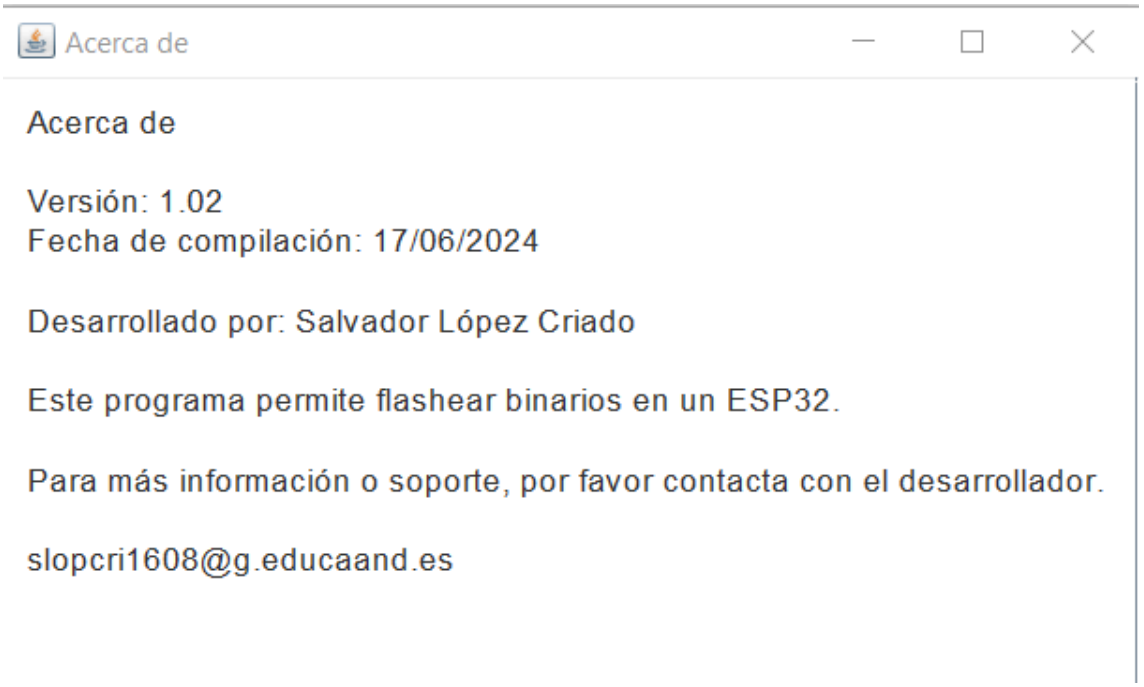
Una vez ejecutamos la aplicación veremos la primera ventana y la cual será la principal durante todo nuestro uso de esta.



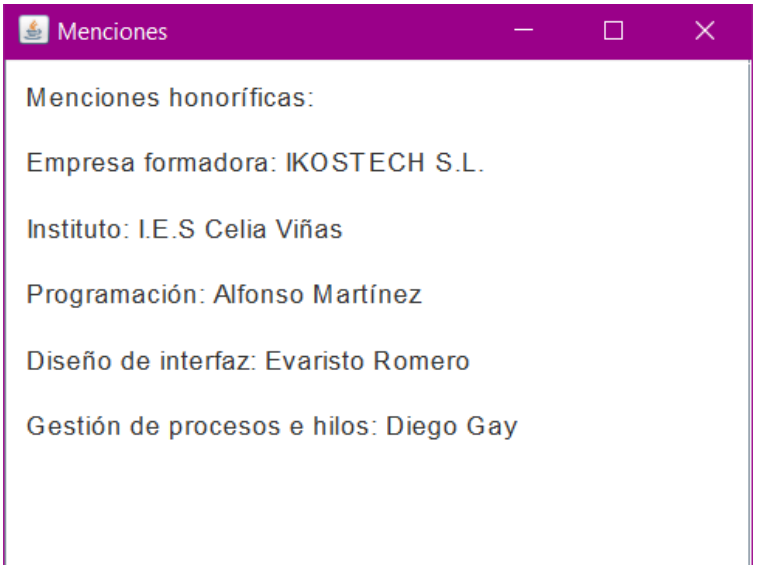
Como se puede apreciar, es bastante básica e intuitiva, facilitando así su uso para un usuario no tan experto.

Arriba vemos el menú **“Detalles”,** del cual se despliegan dos apartados, **“Acerca de”** y **“Menciones honoríficas”.**

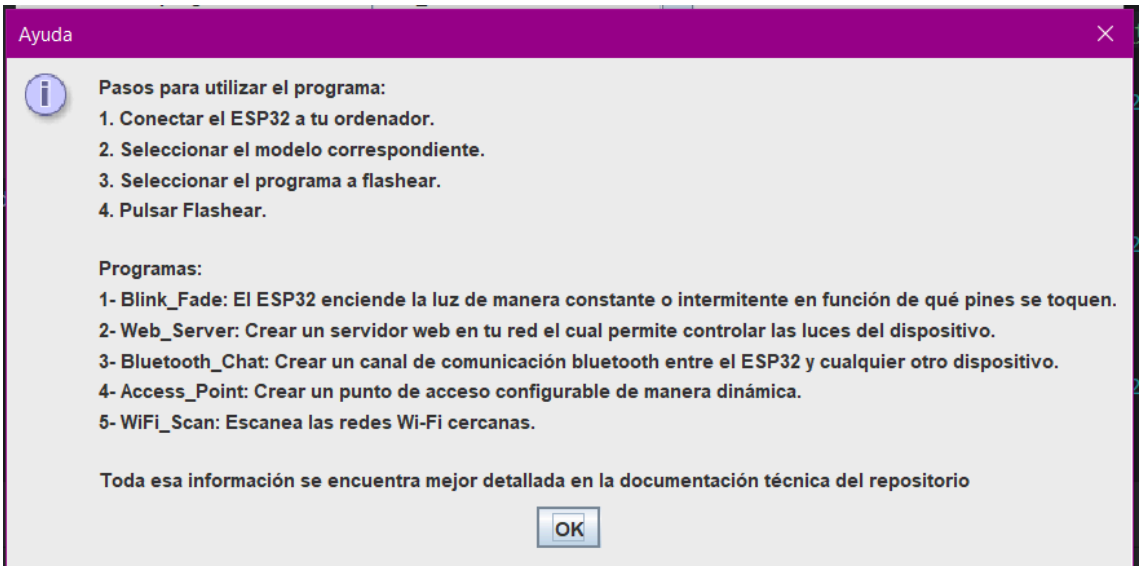
En **“Acerca de”** encontraremos información básica del programa como la fecha de compilación, el uso esperado, la versión de la aplicación y la información del desarrollador, en este caso, yo mismo.



En la siguiente pestaña, en **“Menciones honoríficas”** podremos ver una pequeña dedicatoria a las personas que me han acompañado hasta la realización de este proyecto y los que me han ayudado e inspirado.



Y por último, el botón de **“Ayuda”,** el cual nos va a ofrecer una pequeña ventana con información básica sobre la funcionalidad de la aplicación y de los programas que maneja.



5 – Programas para el ESP32

Dentro de este repositorio podemos encontrar una cantidad de 4 programas distintos.

(Se irán añadiendo más en futuras actualizaciones.)

Vamos a proceder a revisarlos y ver que parámetros se pueden cambiar de manera interna.

5.1 Blink_Fade

La función de este programa es activar los pines sensitivos del ESP32 (en este caso el T0 y el T6) y se programan para que al notar un cambio de voltaje y corriente, se encienda una luz de una manera o de otra, en función de que pin se haya tocado.

El pin T0 por defecto tiene un valor de 150 y el T6 de 120. Estos valores se pueden cambiar para ajustar la sensibilidad de los pines.

```
void comprobarPines() {
    uint16_t toqueIzq = 0, toqueDer = 0;

    //Leer los valores de los pines táctiles. Esto es corriente eléctrica. El valor nulo es
    //150 para el izquierdo y 120 para el derecho
    touch_pad_read(TOUCH_PAD_LEFT_PIN, &toqueIzq);
    touch_pad_read(TOUCH_PAD_RIGHT_PIN, &toqueDer);

    //Si baja de esos valores, significa que se están tocando.
    if (toqueIzq < 120 && toqueDer > 100) {
        pinIzq = true;
        pinDer = false;
    } else if (toqueDer < 100 && toqueIzq > 120) {
        pinDer = true;
        pinIzq = false;
    } else {
        pinIzq = false;
        pinDer = false;
    }
}
```

Dentro de la función comprobarPines, tendríamos que cambiar los valores 120 y 100. (Se prevé cambiar estos valores por unas constantes, para hacer más fácil su edición).

```
void encenderLuces() {
    if (pinIzq) {
        ledc_set_duty(LEDC_HS_MODE, LEDC_HS_CH0_CHANNEL, brillo);
        ledc_update_duty(LEDC_HS_MODE, LEDC_HS_CH0_CHANNEL);
        brillo += prueba;
        if (brillo <= 0 || brillo >= 255) {
            prueba = -prueba;
            vTaskDelay(10 / portTICK_PERIOD_MS); // Cambiar este retraso para la velocidad del fade
        }
    } else if (pinDer) {
        ledc_set_duty(LEDC_HS_MODE, LEDC_HS_CH0_CHANNEL, LEDC_TEST_DUTY);
        ledc_update_duty(LEDC_HS_MODE, LEDC_HS_CH0_CHANNEL);
    } else {
        ledc_set_duty(LEDC_HS_MODE, LEDC_HS_CH0_CHANNEL, 0);
        ledc_update_duty(LEDC_HS_MODE, LEDC_HS_CH0_CHANNEL);
    }
}
```

En función del estado de los pines (verdadero si es está tocando, falso si no), el programa realizará una función u otra. En el caso del derecho, simplemente dará el pin 2 como activo y encenderá la luz led y en el caso del izquierdo, hará un gradiente con la luz simulando así un parpadeo constante.

5.2 Web_Server

Este programa genera un servidor web en la red wifi que nosotros le especifiquemos y nos permite encender y apagar las luces del ESP32 de manera remota pulsando unos botones.

5.2.1 HTML_TEMPLATE

En esta variable podemos cambiar el html correspondiente a nuestro antojo.

```
// HTML para el control de LEDs
const char *HTML_TEMPLATE =
    "<!DOCTYPE html>"
    "<html>"
    "<head><meta name=\"viewport\" content=\"width=device-width, initial-scale=1.0, user-scalable=no\">"
    "<title>LED Control</title>"
    "<style>html { font-family: Helvetica; display: inline-block; margin: 0px auto; text-align: center;}"
    "body{margin-top: 50px;} h1 {color: #444444;margin: 50px auto 30px;} h3 {color: #444444;margin-bottom: 50px;}"
    ".button {display: block;width: 80px;background-color: #3498db;border: none;color: white;padding: 13px 30px;text-decoration: none;font-size: 25px;margin: 0px auto 35px;cursor: pointer;border-radius: 4px;}"
    ".button-on {background-color: #3498db;}"
    ".button-on:active {background-color: #2980b9;}"
    ".button-off {background-color: #34495e;}"
    ".button-off:active {background-color: #2c3e50;}"
    "p {font-size: 14px;color: #888;margin-bottom: 10px;}"
    "</style>"
    "</head>"
    "<body>"
    "<h1>ESP32 Web Server</h1>"
    "<h3>Using Station(STA) Mode</h3>"
    "<p>LED1 Status: %s</p><a class=\"button %s\" href=\"/led1%s\">%s</a>"
    "<p>LED2 Status: %s</p><a class=\"button %s\" href=\"/led2%s\">%s</a>"
    "<p>LED3 Status: %s</p><a class=\"button %s\" href=\"/led3%s\">%s</a>"
    "</body>"
    "</html>";
```

5.2.2 init_wifi

Este método será el encargado de conectar nuestro ESP32 a la red wifi. Aqui tendremos que especificarle manualmente a que red queremos conectarnos y que contraseña tiene.

Ahora mismo se hace de manera “HardCoded” pero se prevé unir al programa Access_Point y hacerlo todo de manera dinámica.

```c
static void init_wifi(void)
{
    wifi_init_config_t cfg = WIFI_INIT_CONFIG_DEFAULT();
    ESP_ERROR_CHECK(esp_wifi_init(&cfg));
    ESP_ERROR_CHECK(esp_wifi_set_mode(WIFI_MODE_STA));
    wifi_config_t wifi_config = {
        .sta = {
            .ssid = "Nombre",
            .password = "contraseña",
        },
    };
    ESP_ERROR_CHECK(esp_wifi_set_config(ESP_IF_WIFI_STA, &wifi_config));
    ESP_ERROR_CHECK(esp_wifi_start());
    ESP_ERROR_CHECK(esp_wifi_connect());
}
```

5.3 Bluethoot_Chat

Este programa nos permite crear un canal de comunicación bluetooth entre un dispositivo y el ESP32. Para ello será necesario mínimo un PC y recomendable un teléfono móvil.

Una vez esté flasheado el programa, nos conectaremos al ESP32 con nuestro teléfono móvil y abriremos una terminal serie. Esto se puede hacer con apps como **Serial Bluetooth Terminal**.

A su vez, necesitaremos abrir una terminal serie desde el dispositivo al que esté conectado físicamente el ESP32, en este caso, nuestro PC.

Una vez abiertas ambas terminales, podemos comprobar que al escribir en una u en otra, aparecen los mensajes por pantalla.

En este programa no especifico ningún método ni variable a cambiar ya que es bastante cuadriculado y no permite modificación alguna.

5.4 Access_Point

Este programa nos permite crear un Punto de Acceso con nuestro ESP32. A su vez, nos permite, de manera dinámica, conectar ese punto de acceso a una red wifi cercana.

Una vez esté funcionando, nos conectamos desde cualquier dispositivo mediante Wi-Fi. Una vez conectado, podemos acceder a su dirección IP y desde ahí, encontraremos un pequeño servidor web el cual se encarga de gestionar la conexión a la red Wi-Fi deseada.

Está hecho y compilado en c + + (Arduino), a si que de cara a modificar y entender su funcionamiento, recomiendo leer la documentación de la librería correspondiente.