

Laboratorio 2

TouchPad, WiFi y Webserver

Objetivo

Ya con algunas nociones básicas del IDE y la placa de desarrollo del curso, se seguirá avanzando en el control del hardware, se conectará el dispositivo a una red WiFi y generará su propia red, así como también se montará un pequeño servidor web en el dispositivo.

Se comenzará a utilizar la herramienta brindada por Espressif para la configuración de registros de periféricos a través de sus APIs (Interfaz de Programación de Aplicaciones). Con esta herramienta, se configurarán puertos de entrada para el uso de botones, comunicación WiFi y servidor web.

Evaluación y metodología de trabajo

Al igual que en el resto de los laboratorios y el proyecto final, se trabajará en grupos (siempre el mismo). Los docentes asistirán a los estudiantes constantemente realizando sugerencias, devoluciones y aclarando conceptos cada vez que se necesite.

Se evaluará el trabajo de cada grupo durante el laboratorio, y se considerarán para dicha evaluación los siguientes criterios:

- A. Organización del grupo, involucramiento de cada uno de los integrantes.
- B. Completitud y correctitud de las soluciones a las tareas indicadas en el laboratorio en el tiempo de clase.
- C. Respuestas a las preguntas planteadas en el laboratorio y/o preguntas que los docentes puedan realizar.
- D. Utilización de buenas prácticas de programación.
- E. Entrega en fecha de las soluciones propuestas.

Ejercicios

Primera Parte

En esta primera parte haremos uso de la placa de extensión [ESP-LyraP-TouchA v1.1](#) que cuenta con 6 botones capacitivos y un “anillo de guarda” para aplicaciones “resistentes al agua”.

1. **Rehacer la configuración del LED del laboratorio anterior.**
 - a. En VSCode crear un proyecto nuevo y llamarlo ‘Laboratorio_2a’.
 - b. Agregar al proyecto las librerías del LED RGB creadas en el ‘Laboratorio_1’.
 - c. Crear un programa sencillo y probar que el laboratorio 1 sigue funcionando correctamente (Probar que las librerías están bien incluidas y que estas funcionan).
2. **Lectura de botones mediante polling**
 - a. Lea las [referencias](#) brindadas por Espressif para botones touch y haga una breve descripción de su funcionamiento.
 - b. Revise el [esquemático de la placa base](#) y el [esquemático del TouchPad](#) ¿Hay que modificar algo en la placa base para poder utilizar la placa de expansión? ¿Qué problemas (known issues) existen para esta placa de expansión?
 - c. Diseñe un algoritmo (en pseudocódigo) indicando cómo utilizar el TouchPad y cómo detectar el estado de sus botones (presionado/no presionado).
 - d. Cree una librería nueva inicializando el TouchPad e implementando la detección de los botones y modifique el código anterior para que al presionar un botón cambie el estado del LED (color, brillo, on/off, etc).

Segunda Parte

En esta oportunidad se creará otro proyecto (Laboratorio_2b, se puede mantener todo en un mismo código pero recuerde mantener la prolijidad del mismo) y se implementará un Access Point (AP) y una estación (STA) WiFi y se debe poder conectar tanto al equipo como el equipo a una red ya existente.

1. Interfaz de Red y Loop de eventos

- a. Para poder llevar a cabo la implementación tanto del punto de acceso Wifi como la estación WiFi es necesario comprender que es el [ESP-NETIF](#) y que es el [Event Loop](#).
- b. Comente que hace la función `'esp_netif_init()'` y si cree que es necesario para la implementación de redes WiFi.
- c. Comente que hace la función `'esp_event_loop_create_default()'` y qué utilidad podría tener para la implementación de redes WiFi.

2. implementación AP

- a. Comente con qué opciones de configuración cuenta el API de WiFi del equipo basado en la [guía de referencia](#).
- b. implemente en pseudocódigo, y utilizando las funciones que aparecen en la referencia (se recomienda investigar bien los distintos [ejemplos](#) para llevar a cabo dicha implementación), como implementar un AP con el ESP32-S2.
- c. Lleve a cabo dicha implementación e intente conectar un dispositivo al ESP32-S2. ¿Qué datos del equipo que se conecta muestra el equipo? ¿Qué datos de red relevantes puede obtener en el monitor del idf?
- d. Pruebe distintas configuraciones (WiFi abierto/con contraseña, SSID oculto, etc.) y comente su resultado en la implementación.

3. implementación STA

- a. Recordando las funciones provistas en la [guía de referencia](#), implemente, en pseudocódigo (se recomienda investigar bien los distintos [ejemplos](#) para llevar a cabo dicha implementación), una STA con el ESP32-S2 y conectarlo a alguna red WiFi que tenga disponible.
- b. Lleve a cabo dicha implementación e intente conectar el equipo a una red. ¿Qué datos de red relevantes puede obtener en el monitor del idf? ¿Qué diferencias hay comparado con la implementación del AP?
- c. Pruebe distintas configuraciones y comente su efecto en la implementación.

Tercera Parte

En esta parte se hará énfasis en la creación de un servidor web en el ESP-KALUGA. En él almacenaremos una página web, a la cual accederemos para configurar el equipo. Para ello será necesario crear una página web sencilla, esta incluye un archivo de hipertexto (*.html), un archivo de “estilo” (*.css, opcional) y un archivo de ejecución de script javascript (*.js). Para comprender más el funcionamiento de estos archivos se recomienda la página W3School y los tutoriales y referencias que este brinda para crear páginas web.

- a. Recordando las funciones provistas en la [guía de referencia](#) implemente en pseudocódigo (se recomienda investigar bien los distintos [ejemplos](#) para llevar a cabo dicha implementación) un servidor web y que al acceder despliegue el mensaje de “Hola mundo” en el navegador.
- b. Usando las librerías ya implementadas genere una red WiFi (que el equipo funcione como AP) o conecte el mismo a una red ya presente (que el equipo funcione como STA).
- c. Monte un servidor web basándose en los ejemplos brindados previamente. Se debe poder conectar/acceder a este desde una PC como de un dispositivo móvil (por ej. Celular).
- d. ¿Cómo haría para enviar información al equipo? Implemente dicha solución.

Nota: Puede serle de utilidad saber que puede incluir archivos en la compilación. Para ello debe agregar la siguiente línea de comando en el correspondiente archivo de CMake:

```
idf_component_register(SRCS "${srcs}"  
                      INCLUDE_DIRS "${include_dirs}"  
                      REQUIRES "${requires}"  
                      PRIV_REQUIRES "${priv_requires}"  
                      EMBED_FILES "<path-to>/index.html")
```

Y puede acceder a dicho archivo desde el ESP-IDF con las siguientes declaraciones:

```
extern const uint8_t index_html_start[] asm("_binary_index_html_start");  
extern const uint8_t index_html_end[] asm("_binary_index_html_end");
```

- e. ¿Qué datos de un sistema de IoT cree relevantes para persistir en el equipo?.