**Marco de Referencia**

1. **Referente al Hardware**
   1. **ESP32**

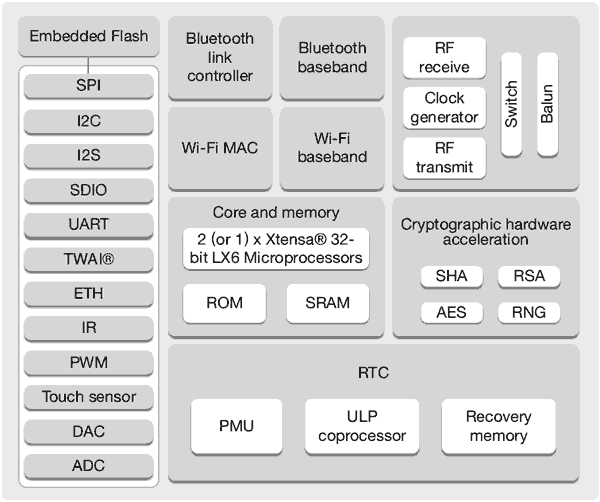
El ESP32 es una serie de SoC (por sus siglas en inglés, System on Chip) y módulos de bajo costo y bajo consumo de energía creado por Espressif Systems. Poseen un alto nivel de integración. En su pequeño encapsulado se incluyen:

* Interruptores de antena
* Balun de RF
* Amplificador de potencia
* Amplificador de recepción de bajo ruido
* Filtros y módulos de administración de energía

Además de todo eso, logra un consumo de energía muy bajo a través de funciones de ahorro de energía que incluyen sincronización de reloj y múltiples modos de operación. Todo esto lo convierte en la herramienta ideal para los proyectos energizados con baterías o aplicaciones IoT.

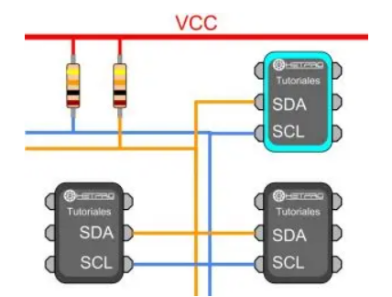
* 1. **Características y especificaciones del ESP32**

En la Figura 1.1, tomada de la hoja de datos, se muestran todos los bloques funcionales que conforman un SoC ESP32.



**Esquema de comunicación y elementos**

Siempre que hablamos de una comunicación oral, se entiende que es entre dos o más personas. Como consecuencia podemos también indicar que en una comunicación digital existen distintos dispositivos o elementos. En el caso de I2C se diferencian dos elementos básicos, un MAESTRO y un ESCLAVO. La Figura 1.2, muestra una conexión típica de tres dispositivos, el bus consiste en dos líneas llamadas, Serial DAta – SDA y Serial CLock – SCL. Es decir, Datos Seriales y Reloj Serial. En particular al bus se le conectan dos resistencias en arreglo pull-up, de entre 2.2K y 10K.

El MAESTRO I2C se encarga de controlar al cable de reloj, por sus siglas en inglés llamada SCL – Serial CLock. Además, el MAESTRO se encarga de iniciar y parar la comunicación. La información binaria serial se envía sólo por la línea o cable de datos seriales, en inglés se llama SDA – Serial DAta. Dos Maestros no pueden hacer uso de un mismo puerto I2C. Puede funcionar de dos maneras, como maestro-transmisor o maestro-receptor. Sus funciones principales son:

* Iniciar la comunicación – S
* Enviar 7 bits de dirección – ADDR
* Generar 1 bit de Lectura ó Escritura – R/W
* Enviar 8 bits de dirección de memoria
* Transmitir 8 bits de datos
* Confirmar la recepción de datos – ACK – ACKnowledged
* Generar confirmación de No-recepción, NACK – No-ACKnowledged
* Finalizar la comunicación

El ESCLAVO I2C, generalmente suele ser un sensor. Este elemento suministra de la información de interés al MAESTRO. Puede actuar de dos formas: esclavo-transmisor ó esclavo-receptor. Un dispositivo I2C esclavo, no puede generar a la señal SCL. Sus funciones principales son:

* Enviar información en paquetes de 8 bits
* Enviar confirmaciones de recepción, llamadas ACK

**1.3 XIAO ESP32S3**

Seeed Studio XIAO ESP32S3 Sense integra sensor de cámara, micrófono digital y soporte para tarjeta SD. Al combinar la potencia informática de ML integrada y la capacidad de fotografía, esta placa de desarrollo puede ser su gran herramienta para comenzar con la IA inteligente de voz y visión.

**Características de la XIAO ESP32S3**

* **Potente placa MCU:** incorpora el chip procesador Xtensa ESP32S3 de 32 bits y doble núcleo que funciona hasta 240 MHz, múltiples puertos de desarrollo montados, compatible con Arduino/MicroPython.
* **Funcionalidad avanzada (para Sense)**: sensor de cámara OV2640 desmontable para resolución de 1600\*1200, compatible con sensor de cámara OV5640, micrófono digital adicional integrado.
* **Diseño de energía elaborado:** capacidad de administración de carga de batería de litio, ofrece 4 modelos de consumo de energía que permiten el modo de suspensión profunda con un consumo de energía tan bajo como 14 uA.
* **Gran memoria para más posibilidades:** ofrece 8 MB de PSRAM y 8 MB de FLASH, compatible con ranura para tarjeta SD para memoria FAT externa de 32 GB.
* **Excelente rendimiento de RF:** admite comunicación inalámbrica dual Wi-Fi de 2,4 GHz y BLE, admite comunicación remota a más de 100 m cuando se conecta con una antena U.FL.
* **Diseño compacto del tamaño de un pulgar:** 21 x 17,5 mm, que adopta el factor de forma clásico de XIAO, adecuado para proyectos con espacio limitado, como dispositivos portátiles.

1. **Bluetooth**

Para que un dispositivo digital sea compatible con Bluetooth, este debe contar con un software adecuado para controlar la transferencia de datos, así como con un chip Bluetooth especial que dispone de una unidad transmisora y otra receptora, y que está integrado en el hardware. Entre los fabricantes más conocidos de estos chips se encuentran Atheros, Nordic Semiconductor o Toshiba. También se puede conectar un adaptador Bluetooth en el puerto USB de un dispositivo y, con ello, añadir esta función.

La frecuencia dedicada a Bluetooth es una banda ISM sin licencia entre los 2,402 GHz y los 2,480 GHz. Los dispositivos compatibles que cumplen los estándares del Bluetooth SIG pueden, como dispositivos de corto alcance o Short Range Devices (SRD), enviar por este rango de frecuencias en todo el mundo y sin licencia. Para poder identificarlo sin ningún género de dudas, cada aparato está provisto de una dirección MAC de 48 bits individual.

1. **Auriculares Inalámbricos**

Los auriculares inalámbricos son auriculares que se conectan a un dispositivo, por ejemplo, un smartphone, un altavoz estéreo, un televisor, una consola de videojuegos, un ordenador u otro dispositivo electrónico, sin utilizar cables. Los auriculares inalámbricos transmiten la señal de audio por señales de radio o de infrarrojos (IR) dependiendo del dispositivo.

**Referente a la Programación**

1. **Librería (o Biblioteca)**

Una librería es uno o varios archivos escritos en un lenguaje de programación determinado, que proporcionan diversas funcionalidades. A diferencia de un framework, una librería no aporta la estructura sobre cómo realizar el desarrollo, sino que proporciona funcionalidades comunes, que ya han sido resueltas previamente por otros programadores y evitan la duplicidad de código. Además, reducen el tiempo de desarrollo y aumentan la calidad de este.

1. **Framework**

Un framework es un conjunto de archivos y pautas que definen la estructura y metodología, sobre cómo hacer el desarrollo de un proyecto software. Se podría decir que es una guía o esquema que nos ayuda a programar de forma sencilla y rápida.

Su objetivo es el desarrollo ágil de aplicaciones mediante la aportación de librerías y/o funcionalidades ya desarrolladas. Principalmente, nos permite centrarnos en el problema, en vez de preocuparnos por implementar funcionalidades que son de uso común en muchas aplicaciones. Generalmente los frameworks están basados en un lenguaje de programación, aunque no siempre es así.

1. **Aplicación**

Una aplicación (también llamada app) es simplemente un programa informático creado para llevar a cabo o facilitar una tarea en un dispositivo informático. Cabe destacar que, aunque todas las aplicaciones son programas, no todos los programas son aplicaciones. Existe multitud de software en el mercado, pero sólo se denomina así a aquel que ha sido creado con un fin determinado, para realizar tareas concretas. No se consideraría una aplicación, por ejemplo, un sistema operativo, ni una suite, pues su propósito es general.

1. **Socket**

Un socket (enchufe), es un método para la comunicación entre un programa del cliente y un programa del servidor en una red. Un socket se define como el punto final en una conexión. Los sockets se crean y se utilizan con un sistema de peticiones o de llamadas de función a veces llamados interfaz de programación de aplicación de sockets (API, application programming interface).

Un socket es también una dirección de Internet, combinando una dirección IP (la dirección numérica única de cuatro partes que identifica a un ordenador particular en Internet) y un número de puerto (el número que identifica una aplicación de Internet particular, como FTP, Gopher, o WWW).

1. **Pipe**

Una tubería (pipe, cauce o ‘|’) consiste en una cadena de procesos conectados de forma tal que la salida de cada elemento de la cadena es la entrada del próximo. Permiten la comunicación y sincronización entre procesos. Es común el uso de buffer de datos entre elementos consecutivos.

**Referente al Software**

1. **Sistema Operativo**

Los equipos actuales de la gama media manejan un WI-FI 2.4ghz lo cual nos permite una transmisión de datos de 50 a 60 Mbps. Estos contienen un sistema operativo, los cuales vienen en variedad, los más populares y los que siguen abarcando el mercado solo son dos, Android e IOS.

1. **Android**

Android es una pila de software de código abierto basado en Linux creada para una variedad amplia de dispositivos y factores de forma que contiene componentes principales como son kernel, capa de abstracción de hardware, tiempos de ejecución bibliotecas en C/C++ nativas, marcos de trabajo API de java y Apps de trabajo.

1. La primera capa Kernel basado en Linux para funcionalidades subyacentes, como la generación de subprocesos y la administración de memoria de bajo nivel.
2. La segunda capa llamada capa de abstracción de hardware (HAL) consiste en varios módulos de bibliotecas y cada uno de estos implementa una interfaz para un tipo específico de componentes de hardware.
3. La tercera capa se compone de dos partes la Biblioteca C/C++ nativas y tiempos de ejecución de Android, la biblioteca C/C++ se basa en código nativo que requiere bibliotecas escritas en C y C++, Android proporciona API del marco de trabajo de java para exponer la funcionalidad de algunas de estas bibliotecas nativas, según cada app ejecuta sus propios procesos con sus propias instancias.
4. La cuarta capa consiste en el marco de trabajo de Java API, todas estas funciones del sistema operativo Android están escritas en el lenguaje java, son las bases para crear apps de Android simplificando la reutilización de componentes del sistema y servicios centrales y modulares, los desarrolladores cuentan con acceso a todas las API de trabajo
5. La quinta capa consiste en la APPS del sistema, contiene conjunto de apps para correo electrónico, mensajería SMS, calendarios, navegación en internet y contactos, las apps no contienen un estado principal esto da libertad que el usuario pueda instalar cualquier app, existen al gua excepciones como la “app Settings”.
6. **IOS**

El sistema operativo IOS se compone de cuatro capas principales, cada una de las cuales proporciona un conjunto de servicios y funcionalidades específicas. A continuación, se describen brevemente estas cuatro capas:

1. Capa de Cocoa Touch: esta es la capa superior de IOS y proporciona un conjunto de frameworks y herramientas para desarrollar aplicaciones IOS. Los frameworks incluyen UIKit para la interfaz de usuario, MapKit para la integración de mapas, y muchos más. También incluye el SDK de IOS para el desarrollo de aplicaciones.
2. Capa de Media: esta capa proporciona servicios para reproducir y grabar audio y video. Incluye los frameworks Core Audio y AVFoundation, que proporcionan servicios para la reproducción y grabación de audio y video, y el framework Core Animation, que se utiliza para animaciones y efectos visuales.
3. Capa de Core Services: esta capa proporciona servicios de bajo nivel, como la gestión de archivos, la red, la seguridad y la ubicación. Incluye los frameworks Core Location, para la obtención de la ubicación del dispositivo, y Core Data, para el almacenamiento de datos.
4. Capa de Kernel: esta es la capa inferior de IOS y proporciona una abstracción de hardware para las capas superiores del sistema operativo. Incluye el kernel de IOS, que es un kernel de tipo Unix que proporciona servicios básicos de gestión de procesos, gestión de memoria y gestión de dispositivos.

Que representa el accurancy y con que se mide