

# UNIVERSIDAD TÉCNICA DE AMBATO



## FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL MAESTRÍA EN TELECOMUNICACIONES

---

**TEMA:** SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE  
LOS CULTIVOS DE LA PARROQUIA IZAMBA.

---

Trabajo de Titulación previo a la obtención del Grado Académico de  
Magíster en Telecomunicaciones.

Modalidad de titulación "Proyecto de Desarrollo"

**Autor:** Ingeniero Gonzalo Guillermo Velasco Arcos

**Director:** Ingeniero Juan Pablo Pallo Noroña, Magíster.

Ambato - Ecuador

2021

## **APROBACIÓN DEL TRABAJO DE TITULACIÓN**

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas Electrónica e Industrial.

El Tribunal receptor del Trabajo de Titulación presidido por la Ingeniera Elsa Pilar Urrutia Urrutia, Magíster, e integrado por los señores Ingeniero Víctor Santiago Manzano Villafuerte Magíster e Ingeniero Julio Enrique Cuji Rodríguez Magíster, designados por la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial de la Universidad Técnica de Ambato, para receptor el Trabajo de Titulación con el tema: “SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE LOS CULTIVOS DE LA PARROQUIA IZAMBA”, elaborado y presentado por el señor Ingeniero Gonzalo Guillermo Velasco Arcos, para optar por el Grado Académico de Magíster en Telecomunicaciones; una vez escuchada la defensa oral del Trabajo de Titulación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la Universidad Técnica de Ambato.

-----  
Ing. Elsa Pilar Urrutia Urrutia, Mg.

**Presidenta y Miembro del Tribunal de Defensa**

-----  
Ing. Víctor Santiago Manzano Villafuerte, Mg.

**Miembro del Tribunal de Defensa**

-----  
Ing. Julio Enrique Cuji Rodríguez, Mg.

**Miembro del Tribunal de Defensa**

## **AUTORÍA DEL TRABAJO DE TITULACIÓN**

La responsabilidad de las opiniones, comentarios y críticas emitidas en el Trabajo de Titulación presentado con el tema: “Sistema de Información Geográfica Para la Gestión de los Cultivos de la Parroquia Izamba”, le corresponde exclusivamente al: Ingeniero Gonzalo Guillermo Velasco Arcos, Autor bajo la Dirección del Ingeniero, Juan Pablo Pallo Noroña, Magíster, director del Trabajo de Titulación; y el patrimonio intelectual a la Universidad Técnica de Ambato.

-----  
Ing. Gonzalo Guillermo Velasco Arcos

**AUTOR**

-----  
Ing. Juan Pablo Pallo Noroña, Mg.

**DIRECTOR**

## **DERECHOS DE AUTOR**

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Titulación, sirva como un documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad.

-----  
Ing. Gonzalo Guillermo Velasco Arcos

C.C.: 1804032538

# ÍNDICE GENERAL

<b>CONTENIDO</b>	<b>i</b>
PORTADA . . . . .	i
APROBACIÓN DEL TRABAJO DE TITULACIÓN . . . . .	ii
AUTORÍA DEL TRABAJO DE TITULACIÓN . . . . .	iii
DERECHOS DE AUTOR . . . . .	iv
ÍNDICE GENERAL . . . . .	v
ÍNDICE DE TABLAS . . . . .	viii
ÍNDICE DE FIGURAS . . . . .	ix
AGRADECIMIENTO . . . . .	x
DEDICATORIA . . . . .	xi
RESUMEN EJECUTIVO . . . . .	xii
EXECUTIVE SUMMARY . . . . .	xiv
<b>1 CAPÍTULO I</b>	
<b>EL PROBLEMA DE INVESTIGACIÓN</b>	<b>1</b>
1.1 Introducción . . . . .	1
1.2 Justificación . . . . .	2
1.3 Objetivos . . . . .	3
1.3.1 General . . . . .	3
1.3.2 Específicos . . . . .	3
<b>2 CAPÍTULO II</b>	
<b>ANTECEDENTES INVESTIGATIVOS</b>	<b>4</b>
2.1 Estado del Arte . . . . .	4
2.2 Marco Teórico . . . . .	7
2.2.1 Agricultura Actual . . . . .	7
2.2.2 Requerimientos Edafoclimáticos . . . . .	7
2.2.3 Sistema de Comunicación Electrónica . . . . .	7
2.2.4 Sistema de Comunicaciones en la Agricultura . . . . .	8
2.2.5 Sensores en la agricultura . . . . .	8
2.2.6 Redes de Sensores Inalámbricos (WSN) . . . . .	9
2.2.7 Componentes y arquitectura de una WSN . . . . .	9
2.2.8 Topologías WSN . . . . .	10
2.2.9 Sistema electrónico . . . . .	12

2.2.10	Dispositivos de Entrada . . . . .	12
2.2.11	Dispositivos de proceso . . . . .	16
2.2.12	Dispositivos de Salida . . . . .	17
2.2.13	Sistema de Información Geográfica . . . . .	18
2.2.14	GPS Essentials . . . . .	18
<b>3</b>	<b>CAPÍTULO III</b>	
	<b>METODOLOGÍA</b>	<b>19</b>
3.1	Ubicación . . . . .	19
3.2	Equipos y materiales . . . . .	19
3.3	Tipo de investigación . . . . .	19
3.3.1	Investigación Bibliográfica . . . . .	19
3.3.2	Investigación de Campo . . . . .	19
<b>4</b>	<b>CAPÍTULO IV</b>	
	<b>ANÁLISIS E INTERPRETACIÓN DE RESULTADOS</b>	<b>20</b>
4.1	Parroquia Izamba . . . . .	20
4.1.1	Ubicación Geográfica y Límites . . . . .	20
4.2	Factores edafoclimáticos . . . . .	22
4.2.1	pH . . . . .	23
4.2.2	Humedad del Suelo . . . . .	23
4.2.3	Temperatura Ambiental o Factor Climático . . . . .	24
4.3	Tipos de cultivos . . . . .	24
4.4	Requerimientos Técnicos . . . . .	24
4.4.1	Diagrama General del Sistema . . . . .	26
4.5	Análisis Técnico del Hardware Requerido . . . . .	27
4.5.1	Microcontrolador . . . . .	27
4.5.2	Sensor de Temperatura y pH . . . . .	28
4.5.3	Sensor de Humedad del Suelo . . . . .	29
4.5.4	Dispositivo de Comunicación . . . . .	30
4.5.5	Análisis del Software . . . . .	30
4.6	Adquisición de Señales . . . . .	30
4.6.1	Sensor PH-4502 . . . . .	31
4.6.2	Sensor de Humedad del Suelo . . . . .	36
4.7	Procesamiento y Transmisión . . . . .	37
4.8	Presentación de Señales . . . . .	39

4.9	Diseño Electrónico del Prototipo . . . . .	39
4.9.1	Circuito Esquemático de los Nodos Sensores . . . . .	39
4.10	Topología física y lógica del sistema . . . . .	43
4.11	Pruebas de Funcionamiento . . . . .	43
4.12	Análisis presupuestario . . . . .	50
<b>5</b>	<b>CAPÍTULO V</b>	
	<b>CONCLUSIONES Y RECOMENDACIONES</b>	<b>51</b>
5.1	Conclusiones . . . . .	51
5.2	Recomendaciones . . . . .	52
	<b>BIBLIOGRAFÍA</b> . . . . .	<b>53</b>
	<b>ANEXOS</b> . . . . .	<b>55</b>

## ÍNDICE DE TABLAS

2.1	Componentes WSN . . . . .	9
2.2	Tipos de sensores más utilizados . . . . .	16
3.1	Equipos y materiales . . . . .	19
4.1	Tipos de Suelo de la parroquia Izamba . . . . .	21
4.2	Uso actual del suelo parroquia de Izamba . . . . .	22
4.3	Rango de pH . . . . .	23
4.4	Tipos de cultivos según requerimientos edafoclimáticos . . . . .	24
4.5	Comparación Técnica de los Microcontroladores . . . . .	28
4.6	Características de los sensores de temperatura y pH . . . . .	29
4.7	Características del Sensor de Humedad del Suelo . . . . .	29
4.8	Tabla de presupuesto económico de nodo sensor . . . . .	50



## ÍNDICE DE FIGURAS

2.1	Diagrama de un sistema de comunicación electrónica . . . . .	8
2.2	Arquitectura básica de un nodo sensor . . . . .	10
2.3	Topologías de una Red . . . . .	10
2.4	Diagrama de bloques de sistema electrónico . . . . .	12
2.5	Sensores según su funcionamiento . . . . .	13
2.6	Sensores según las señales generadas . . . . .	14
2.7	Sensores según variable física medida . . . . .	14
4.1	Ubicación de la parroquia Izamba . . . . .	20
4.2	Cobertura y Usos del Suelo de la Parroquia de Izamba . . . . .	22
4.3	Diagrama de Bloques . . . . .	25
4.4	Etapa de Adquisición de Información . . . . .	25
4.5	Diagrama General del Sistema . . . . .	27
4.6	Pinout Sensor PH-4502 . . . . .	32
4.7	Algoritmo de Valor Sensor en Voltaje . . . . .	33
4.8	Algoritmo de Calibración de pH . . . . .	35
4.9	Diagrama de Pines FC-28 . . . . .	36
4.10	Diagrama de Flujo Transmisión de Datos . . . . .	38
4.11	Circuito Esquemático Nodo Sensor . . . . .	40
4.12	Nodo Sensor . . . . .	42
4.13	Topologías del sistema . . . . .	43
4.14	Baterías de LiPo . . . . .	44
4.15	Instalación prototipo sectores agrícolas . . . . .	44
4.16	Prueba de Conexión de los Nodos Sensores . . . . .	45
4.17	Resultados de Servidor Web . . . . .	46
4.18	Puntos de Referencia GPS Essentials . . . . .	47
4.19	Ubicación de nodos sensores en Google Maps . . . . .	49

## AGRADECIMIENTO

*A mis amados padres, por su apoyo incondicional, sus consejos constantes,  
por ser los pilares fundamentales de mi vida y ayudarme en esta nueva  
etapa con los principios y valores que me han inculcado  
para ser una mejor persona.*

*Al amor de mi vida, Liss, por ser el complemento perfecto, mil gracias por tu  
tiempo, por ser mi compañera en cada paso y llenar de alegrías cada día  
que compartimos juntos, te amo.*

*A mi hija, Keyla, que es la luz de mis ojos, mi fortaleza un motivo más por  
el cual luchar y levantarme cada día. Conjuntamente con Liss  
son mis mágicas princesas.*

*A mi familia, que día tras día han estado siempre pendiente y apoyándose  
en todos mis pasos por falsos que hayan estado.*

*A mi tutor, Ing. Juan Pablo Pallo, una excelente persona en todo el sentido  
de la palabra. Gracias por apoyarme en esta etapa para culminar  
con éxito y salir hacia adelante.*

*A Cristina y Franklin una mención especial ya que sin su ímpetu por el  
estudio no hubiera seguido la maestría, gracias por acompañarme  
en este camino de principio a fin.*

*A mis compañeros y amigos que semestre a semestre hemos superado  
cada obstáculo y poniendo como reto la culminación de esta  
grandiosa etapa.*

**Gonzalo Velasco**

## **DEDICATORIA**

*A mis padres, por estar siempre pendiente y apoyándome con sus palabras y consejos.*

*A mi linda esposa Liss, por su apoyo incondicional, amor, motivación y darme ánimos en todo momento. Sin duda la persona que me ha hecho crecer como ser humano y así mismo mostrarme siempre el lado bueno de las cosas.*

*Te amo mi amor.*

*A mi princesita Keyla, el mejor regalo que la vida me pudo dar, gracias por sacarme una sonrisa con tus divertidas ocurrencias.*

*A mis amigos Franklin, Cristina, Anita, Sabrina y Kevin por sus palabras de aliento y siempre estar ahí cuando mas los necesite en su debido momento y ayudar que mi meta se haga realidad.*

**Gonzalo Velasco**

**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**

**MAESTRÍA EN TELECOMUNICACIONES**

**TEMA:**

SISTEMA DE INFORMACIÓN GEOGRÁFICA PARA LA GESTIÓN DE LOS CULTIVOS DE LA  
PARROQUIA IZAMBA.

**AUTOR:** Ingeniero Gonzalo Guillermo Velasco Arcos

**DIRECTOR:** Ingeniero Juan Pablo Pallo Noroña, Magíster.

**LÍNEA DE INVESTIGACIÓN:**

- TECNOLOGÍAS, SEGURIDAD Y GESTIÓN DE REDES DE COMUNICACIONES

**FECHA:** 27 de octubre de 2021

**RESUMEN EJECUTIVO**

La agricultura es la columna vertebral del Ecuador y nuestra economía, que representa casi el 9 por ciento del producto interno bruto (PIB) y genera fuentes de empleo por encima del 30 por ciento. Debido al aumento de la demanda de alimentos, las personas están tratando de hacer esfuerzos adicionales y técnicas especiales para multiplicar la producción de alimentos, es decir, sacar el mayor rendimiento a sus plantaciones. El uso de diferentes tecnologías para el sector agrícola es uno de esos esfuerzos. Aparte del uso de tecnologías científicas en la agricultura, la tecnología de la información se está ejerciendo mucho en esta área. Tecnologías como la navegación por satélite, la red de sensores, la computación en cuadrícula, la computación ubicua y la computación consciente del contexto están respaldando dicho dominio para mejorar las capacidades de monitoreo y toma de decisiones.

El agricultor de hoy en día se preocupa debido a que esta labor cada vez se hace más costosa por los insumos, el consumo de agua, esto solo en la agricultura tradicional y si a esto se le aumenta la tecnología ya sea para ambientes no controlados (aire libre) o controlados (invernadero) considerarían que es una inversión que no lo requieran como necesaria, pero debido al avance de otros países ya se empieza a considerar a la tecnología dentro de la agricultura como una técnica fundamental. Sin embargo, el fuerte de los ciudadanos que llevan las labores agrícolas no es la tecnología, gracias al Ministerio de Agricultura y Ganadería (MAG) se ha realizado esfuerzos para que este sector conste con algunas capacitaciones por parte del gobierno ecuatoriano y así brindar al agricultor el conocimiento necesario no solo en la parte de requerimiento edafoclimáticos sino también en técnicas que pueden utilizar dentro del periodo de labranza ayudados de la tecnología. En términos generales, esto genera un descontento a gran parte del sector agrícola debido que solo queda en palabras y ellos al no tener gran

conocimiento en esta parte no lo pueden llegar a implementar en sus cultivos y siendo el caso contrario accederían a una persona externa para que realice los procesos de automatización por así llamarlo y que puedan acceder a la denominada agricultura de precisión como se la conoce hoy.

Por otro lado, la mayoría de técnicas en las que se utiliza la denominada agricultura de precisión se da cuando los cultivos ya han sido seleccionados y sembrados, mas no antes del proceso de siembra, de esta manera se quiere dar un aporte más significativo al sistema que se va a desarrollar, teniendo en cuenta desde el minuto cero las condiciones edafoclimáticas y de acuerdo a ello ir por una toma de decisiones menos complicada a la hora de que cultivo plantar para sacar el mayor rendimiento posible.

La investigación propone un sistema de fácil manejo para las personas dedicadas a la agricultura y de bajo costo. En primera instancia se utiliza la tecnología basada en redes de sensores inalámbricos (WSN) para acceder a la información edafoclimática que se pueda adquirir mediante algunos sensores. En segundo lugar, para el envío de información se configura una red de malla ya que se pueden adicionar múltiples dispositivos a la red y en el caso de que exista algún problema en un nodo, el sistema siga funcionando con normalidad.

Para finalizar, se presentan los resultados al análisis realizado mediante el sistema de redes de sensores inalámbricos desde el minuto cero como se lo ha denominado al proceso de tomar una decisión antes de plantar e indicando el lugar mediante un sistema de información geográfica.

Adicionalmente, se evalúa las ventajas de utilizar una red de sensores inalámbricos en la agricultura tradicional y resultados comparecientes de acuerdo a los obtenidos mediante el sistema. Además de proporcionar facilidad de manejo con un mayor beneficio para todas las personas que realizan esta labor diaria dentro del país.

**PALABRAS CLAVE:** Red de Sensores Inalámbricos, Agricultura de Precisión, Agricultura Tradicional, Sistemas de Información Geográfica, Módulo ESP32, Topología tipo Malla, Sensor de pH E201 BNC Electrodo, Humedad del Suelo, GPS Essentials.

**UNIVERSIDAD TÉCNICA DE AMBATO**

**FACULTAD DE INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**

**MAESTRÍA EN TELECOMUNICACIONES**

**THEME:**

GEOGRAPHICAL INFORMATION SYSTEM FOR CROP MANAGEMENT IN IZAMBA PARISH.

**AUTHOR:** Engineer Gonzalo Guillermo Velasco Arcos

**DIRECTED BY:** Engineer Juan Pablo Pallo Noroña, Magister.

**LINE OF RESEARCH:**

- TECHNOLOGIES, SECURITY AND MANAGEMENT OF COMMUNICATIONS NETWORKS

**DATE:** October 27th, 2021

**EXECUTIVE SUMMARY**

Agriculture is the backbone of Ecuador and our economy, which accounts for almost 9 per cent of gross domestic product (GDP) and generates sources of employment above 30 per cent. Because of the increased demand for food, people are trying to make extra efforts and special techniques to multiply food production, that is, to get the most out of their plantations. The use of different technologies for the agricultural sector is one such effort. Apart from the use of scientific technologies in agriculture, information technology is being used extensively in this area. Technologies such as satellite navigation, sensor network, grid computing, ubiquitous computing, and context-conscious computing are supporting that domain to improve monitoring and decision-making capabilities.

Today's farmer is concerned because this work is becoming more and more expensive by inputs, water consumption, this only in traditional agriculture and if this is increased technology either for uncontrolled (outdoor) or controlled environments (greenhouse) would consider it an investment that they do not require as necessary, but due to the progress of other countries technology is already beginning to be considered within agriculture as a fundamental technique. However, the strength of the citizens who carry out the agricultural work is not technology, thanks to the Ministry of Agriculture and Livestock (MAG) efforts have been made to ensure that this sector receives some training from the Ecuadorian government and thus provide the farmer with the necessary knowledge not only in the area of soil and climate requirements but also in techniques that can be used within the Technology assisted tillage period. In general terms, this generates a discontent to much of the agricultural sector because it only remains in words and they do not have much knowledge in this part can not get to implement it in their crops and otherwise they would access an external person to carry out the processes I would like

to make it clear to the House that the Committee on Agriculture, Fisheries and Rural Development, Fisheries and Food Safety, has decided to set up a Committee of Inquiry into the Committee on the Environment, Public Health and Consumer Protection.

On the other hand, most of the techniques in which the so-called precision agriculture is used occur when the crops have already been selected and sown, but not before the sowing process, in this way a more significant contribution is intended to be made to the system to be developed, taking into account from the zero minute the soil and climatic conditions and according to it go for a less complicated decision-making when it comes to planting to get the best possible yield.

The research proposes a user-friendly system for people engaged in agriculture and low cost. In the first instance, the technology based on wireless sensor networks (WSN) is used to access the soil and climate information that can be acquired through some sensors. Second, a mesh network is configured for the sending of information as multiple devices can be added to the network and in the event that there is a problem in a node, the system continues to function normally.

Finally, the results are presented to the analysis carried out through the wireless sensor network system from the zero minute as it has been called to the process of making a decision before planting and indicating the location through a geographic information system.

In addition, the advantages of using a wireless sensor network in traditional agriculture are evaluated and the results compared according to those obtained through the system. In addition to providing ease of handling with greater benefit to all people who perform this daily work within the country.

**KEYWORDS:** Wireless Sensor Network, Precision Agriculture, Traditional Agriculture, Geographic Information Systems, ESP32 Module, Mesh Type Topology, pH Sensor E201 BNC Electrode, Soil Moisture, GPS Essentials.

# 1 CAPÍTULO I

## EL PROBLEMA DE INVESTIGACIÓN

### 1.1 Introducción

Desde antes y hasta la actualidad, la agricultura ha jugado un papel clave para el desarrollo de la civilización humana especialmente en latinoamérica y el caribe responsable del 16% de exportaciones a nivel mundial, una región líder con potencial para aumentar su producción agrícola y seguir por delante de otros países hasta el 2025. Siendo reconocido por expertos en la Conferencia de Ministros de Agricultura de América en noviembre de 2019, latinoamérica necesita políticas de largo plazo para el crecimiento y progreso de la agricultura mediante el uso de tecnologías como agricultura de precisión, vigilancia con drones, planificación y almacenamiento de cosechas, entre otros. De hecho la aplicación de tecnología ha contribuido de manera eficiente en las últimas dos décadas con aplicaciones de técnicas de irrigación mas eficientes, variedades perfeccionadas de los cultivos y mejores prácticas productivas como automatización e informatización, trazabilidad, el uso de sistemas GPS, fertilización y siembra [4, 15].

En Ecuador, el ente regulador del agro es el Ministerio de Agricultura y Ganadería (MAG), que cerro el año 2020 con saldo positivo pese a la pandemia, incrementando sus exportaciones con relación al 2019 en un 12%. Ecuador es una potencia agrícola, pero con la ayuda de la tecnología y el internet puede llegar a liderar el mercado de exportaciones en la región, múltiples beneficios son los que brindan las nuevas tecnologías dentro de la agricultura a fin de lograr un agro de precisión amigable con el medio ambiente. Entre las actividades que lleva a cabo el MAG con la ayuda de la tecnología es la semilla certificada que conjuntamente con el Instituto Nacional de Investigaciones Agropecuarias (INIAP), le dan al agricultor diferentes técnicas y uso para que se obtenga el más alto rendimiento en sistemas de producción ecológicos, implementación de micro nivelación láser para los cultivos y la reducción de costos operativos; todo gracias a modelos matemáticos y estadísticos, mediciones a través de sensores y drones de variables importantes de cada cultivo que permiten la creación de bases de datos con información especializada y georeferenciada [6].

El uso de la red de sensores inalámbricos en el manejo de la agricultura es la técnica más eficiente en este caso, de igual manera la aplicación, características y contribuciones de esta técnica es significativa, algunos se ponen como objetivos la optimización del agua debido a los temporales como sequía que ocurren en ese lugar, otros proponen la elaboración de algoritmos para el ahorro de energía dentro de la red, en sí el principal problema que se enfrenta con este tipo de sistemas es el ahorro de energía de



acuerdo a las fuentes de alimentación que se vayan a utilizar, el cual puede resolverse realizando un buen diseño tomando en cuenta características, limitaciones, ventajas, desventajas y seguridades de la red WSN.

Por lo tanto, se ha considerado necesario la implementación de un sistema enfocado a la adquisición de parámetros ambientales, recopilando información meteorológica y del suelo, utilizando una red de sensores inalámbricos. De acuerdo a ellos determinar su posterior uso o a su vez implementar mejoras en el sector en donde se este realizando las mediciones con el fin de obtener el mayor rendimiento del suelo en el sitio; a través del uso de nuevas tecnologías para la adquisición de datos, análisis y toma de decisiones.

## **1.2 Justificación**

La agricultura es una labor que acompaña al hombre hace miles de años y sus procesos han evolucionado con el transcurso del tiempo. Según la FAO, un tercio de la población global deriva su sustento de la agricultura, y en economías emergentes esta puede representar hasta el 30 % del PIB. La agricultura es una de las principales fuentes de ingreso en el Ecuador, por ende, son esenciales para el desarrollo agrícola del país. Además, las tecnologías de la información y comunicación (TIC) se han empezado a utilizar e implementar en todo el ciclo de vida del cultivo aplicados a la agricultura de precisión, capaces de incrementar el rendimiento del cultivo y optimizar el uso de recursos como el agua, fertilizantes, pesticidas entre otras funciones, sin embargo la implementación del sistema se realiza una vez que los cultivos ya han sido sembrados y no antes.

La mayoría de los sistemas están enfocados hacia la agricultura de precisión y optimizar el uso de recursos como el agua, fertilizantes, pesticidas mas no para determinar qué tipo de cultivo se puede sembrar en las diferentes partes de un terreno, por consiguiente el proyecto de investigación planteado aportará a ese nivel anterior en cual se determinará el tipo de cultivo a sembrar y en que parte del terreno se lo puede hacer para garantizar desde un comienzo el desarrollo óptimo del mismo y un gran aporte para los agricultores de nuestra zona.

Por último se analizó si el proyecto es factible en todas sus etapas por lo cual, se realizó un estudio financiero, que determinó el costo - beneficio del proyecto, el estudio del arte realizado se basó en los estudios acerca de la agricultura de precisión en sus diferentes etapas: antes, durante y después. Consecuentemente la metodología a utilizar se basó en los dos parámetros antes descritos y por cual se obtuvo el tiempo estimado que lleva tomar a cabo el proyecto final cumpliendo con todos los objetivos planteados.

## **1.3 Objetivos**

### **1.3.1 General**

- Diseñar un sistema de información geográfica para la gestión de los cultivos de la parroquia Izamba.

### **1.3.2 Específicos**

- Analizar las propiedades del suelo que inciden en la siembra de un cultivo.
- Desarrollar una red de sensores inalámbricos (WSN) para la obtención de los datos en la parroquia Izamba.
- Implementar el sistema de información geográfica para la gestión de los cultivos de la parroquia Izamba.

## 2 CAPÍTULO II

### ANTECEDENTES INVESTIGATIVOS

#### 2.1 Estado del Arte

Durante la última década, la agricultura ha ido en constante evolución gracias a los avances tecnológicos y la manera en que se recolectan los datos, se vuelve inevitable cambiar el sistema agrícola tradicional y que se de uso a diferentes técnicas utilizando nuevas tecnologías conjuntamente con la electrónica para garantizar la máxima producción agrícola para cada uno de los agricultores.

El término agricultura de precisión (AP) también conocido como agricultura satélite o manejo de cultivos específico del sitio (SSCM) es un concepto interdisciplinario que básicamente propone añadir las herramientas tecnológicas a la tarea diaria del agricultor. En este sentido la tecnología en conjunto con la agricultura han dado magníficos resultados durante todo el proceso de labranza, no solo en el momento de optimizar recursos como agua, pesticidas, etc. sino también mejorar la productividad preservando el medio ambiente.

Hoy en día, la red de sensores inalámbricos es el sistema que más se utiliza por la mayoría de investigadores, en el apartado en curso se analiza el estado del arte sobre investigaciones que buscan cuidar las plantaciones tanto en sus invernaderos como al aire libre, optimizar los recursos que utilizan a diario y todo esto gracias a la implementación de WSN en la agricultura.

La investigación realizada por Sherine El-kader y Basma Mohammad evalúa la importancia de utilizar la red de sensores inalámbricos de acuerdo a una estrategia de enrutamiento adecuada utilizada llamada APTEEN, analizando diferentes factores dentro de un cultivo entre los cuales se destaca el número de nodos de red, el despliegue de los nodos en los sectores estudiados y los parámetros de detección para aumentar el desempeño en la agricultura, cabe destacar de igual manera que el tamaño del despliegue de la red es directamente proporcional con la calidad del rendimiento, es decir, mientras mas nodos se utilicen mejor sera la calidad del producto, sin embargo el estudio plantea un número adecuado de nodos de red para que el costo/beneficio de la implementación sea en menor tiempo [7].

A diferencia de la anterior investigación basada en el monitoreo y utilización de red de sensores inalámbricos para entornos al aire libre, los investigadores Mare Srbinovska y Cvetan Gavrovski proponen una arquitectura WSN para ambientes controlados como los invernaderos. Como resultado de este análisis tomando en cuenta tres parámetros fundamentales dentro de la agricultura como temperatura, humedad e iluminación y utilizando una topología en estrella concluyeron que el consumo de energía de los nodos sensores depende de la frecuencia con que se obtiene los datos y la cantidad de variables a medir, es

decir, la cantidad de variables debe elegirse adecuadamente para crear un sistema de monitoreo ambiental óptimo, también sugiere que es considerable utilizar protocolos de comunicación con velocidades de comunicación bajas, lo que hace referencia a la vida útil del sistema, es decir mientras mayor sea el número de comunicaciones dentro de la implementación, menor será la vida útil del mismo [21].

Dentro de la agricultura de precisión se pueden realizar diversas técnicas (control de riego, optimización de recursos, evaluadores de plaga, etc.), así como también el tipo de tecnologías a utilizar, como menciona Raghunandan en su análisis comparativo, si bien es cierto que la mejor forma para la obtención de datos en la agricultura es a través de una red de sensores inalámbricos, se puede utilizar diferentes tecnologías asociadas a ella entre las cuales se tiene Zigbee, RF, RFID, GSM, GPRS para el envío de datos, cabe recalcar que no todas las tecnologías van a ser utilizadas dentro de un mismo sistema ya que depende de las necesidades y técnicas que se requieran utilizar, pero todas conducen al mismo objetivo de obtener un mayor rendimiento y calidad dentro de la producción agrícola. En base a todo esto el investigador concluye que dependiendo de las técnicas y tecnologías que se vayan a utilizar se debe buscar dispositivos con el menor coste posible y con una buena calidad de medición asegurando la eficiencia dentro del sistema que se vaya a proponer ya que en el mercado se puede encontrar un sinnúmero de sensores y dispositivos a utilizar [19].

El uso de tecnologías dentro de la labor agrícola de cada país es sinónimo de cambio y de un futuro netamente sustentable es lo que plantea Parveen Kumari y Amit Mahal gracias a la incorporación de WSN dentro del proceso de labranza se puede optimizar el rendimiento de los insumos utilizados y preservar los recursos naturales con el fin de ayudar al medio ambiente, para ello plantean utilizar topologías que ayuden a la causa teniendo en cuenta algunas de ellas como las de tipo anillo, malla, estrella, bus y árbol. Dentro de la investigación analizan los parámetros utilizados para la medición de efectividad de cada topología y utilizarla de acuerdo a las necesidades planteadas, sin duda el definir una topología adecuada para el sistema definirá la ruta de la red, algoritmos de enrutamiento y por ende la eficiencia del sistema [11].

De acuerdo a los resultados publicados por Subramania Ananda Kumar y Paramasivam Ilango demuestran que los parámetros más relevantes para realizar un estudio mediante red de sensores inalámbricos en el campo agrícola mediante el suelo son variables como la temperatura, nivel de agua, humedad y la conductividad, a través de ellos se puede determinar la cantidad y calidad de un cultivo, así como también algunos sensores ayudan a conocer la fuerza y calidad de la planta a través de mediciones y control de fotosíntesis, hidrógeno y CO<sub>2</sub>. En base a esto la agricultura de precisión (AP) comprende la integración de tecnología de la información en la agricultura para el aumento de producción y la

calidad de los cultivos desde un control de plagas hasta la optimización de recursos como el agua que son elementos claves para el cuidado del medio ambiente. Los sensores utilizados en el campo poseen muchas características relevantes para la agricultura como por ejemplo, el sensor Hydra Probe II puede medir una amplia gama de fuentes de información que permiten compactar el diseño de un sistema, pero la realidad es que un solo sensor no va a medir todos los parámetros del suelo requeridos por lo que la mejor opción sería utilizar diferentes sensores ya que si alguno falla no perjudicaría en la toma de decisiones [10].

El sistema de monitoreo utilizado para la agricultura de precisión basado en la utilización WSN más completo se lo denomina iFarm, así lo han llamado Yassine Jiber, Hamid Harroud y Ahmed Karmouch al sistema creado por ellos donde analizan desde la perspectiva de la productividad agrícola, arquitectura, características y principales funcionalidades donde se ofrece un conjunto de servicios para los agricultores que incluyen el control de plagas y enfermedades, donde a través de datos recopilados mediante los nodos sensores realizan la correlación de modelos estadísticos para la predicción de las plagas; el riego y la gestión del agua realizado mediante el estudio de las propiedades del suelo, el tipo de terreno, clima y el tipo de cultivo son algunos de los parámetros escogidos para este fin y la optimización de recursos es el más grande desafío que presentan hoy en día los agricultores saber que equipos agrícolas utilizar y cuanta mano de obra, de igual manera cantidad de agua y pesticidas, que cultivo, cuando cultivar o cosechar ayudaran al agricultor a tomar mejores decisiones sin perder tiempo, dinero o recursos. Los sensores utilizados dentro de la investigación son eKo ES1100, ES1110 y ES1201 que a través del sistema iFarm proporciona notificaciones utilizando el sitio web y SMS únicamente [9].

Si bien es cierto que la mejor tecnología a utilizar para el campo de la agricultura son la red de sensores inalámbricos (WSN), no todo es ideal cuando se comienza hablar de la eficiencia energética, así lo describen Haider Mahmood Jawad y Rosdiadee Nordin en el estudio realizado donde la única limitante de las redes WSN es la energía, donde destacan varias técnicas de reducción de energía para el monitoreo a largo plazo. Se realizó una comparación entre diferentes protocolos o tecnologías inalámbricas, como WiFi, Bluetooth, ZigBee, GPRS / 3G / 4G, LoRa y SigFox. Los resultados de la comparación revelaron que los protocolos inalámbricos ZigBee y LoRa son más convenientes para aplicaciones agrícolas que los demás debido a su bajo consumo de energía y rango de comunicación adecuado corto para ZigBee y largo para LoRa [8].

Por lo tanto, de las investigaciones analizadas previamente y con el afán de generar una investigación destacada en la que se utilice la tecnología dentro de la agricultura. Para el proyecto investigativo se propone analizar los parámetros que inciden en un cultivo con el fin de obtener las variables que permitan

realizar un sistema WSN óptimo para la obtención de datos edafoclimáticos y con ellos procesar la información y tener un producto final disponible para los agricultores con toda la información necesaria.

## **2.2 Marco Teórico**

### **2.2.1 Agricultura Actual**

La agricultura juega un papel de vital importancia en el desarrollo de la humanidad. Debido al aumento de la demanda de alimentos, las personas están tratando de realizar esfuerzos para generar el mayor número de producción agrícola en cada plantación. En Ecuador existen 4 regiones bien delimitadas costa, sierra, amazonía y región insular. En la mayor parte del territorio nacional según datos del Instituto Nacional de Estadísticas y Censos (INEC) existen productores agrícolas ocupando 12.3 millones de hectáreas (ha) para la producción agropecuaria continua [22].

De esta cifra, en la región central del Ecuador, específicamente en la zona 3 comprendida por cuatro provincias de la Sierra fueron intervenidas 73.193 ha de terreno lo que generó alimentos para la población como lechuga, papas, maíz, quinua, etc. El agro genera fuentes de trabajo directa e indirectamente, además el sector agrícola no solo es una fuente de alimentación para el sector donde se lo realiza, también existen productos que pueden ser exportados con el procesamiento adecuado y cumpliendo los estándares de calidad [5].

### **2.2.2 Requerimientos Edafoclimáticos**

Una zonificación edáfica y climática (edafoclimática) corresponde a las características tanto del suelo y del clima de una área geográfica determinada respectivamente [14]. De acuerdo al ministerio de agricultura y ganadería (MAG) los requerimientos edafoclimáticos para cada cultivo son diferentes por las condiciones de cada plantación y las características que debe tener cada semilla en los periodos de plantación, formación y cosecha [6].

### **2.2.3 Sistema de Comunicación Electrónica**

El objetivo primordial de un sistema de comunicación electrónica, es el traspaso de información de un lugar hacia otro distinto. En consecuencia, se puede indicar que las comunicaciones electrónicas son la transmisión, recepción y procesamiento de información entre dos o más lugares, a través de circuitos electrónicos. La fuente original de información puede estar en forma analógica (voz), o en forma digital (códigos alfanuméricos). Sin embargo, todas las fuentes de información se deben convertir a energía electromagnética antes de ser enviadas a través de un sistema de comunicación electrónica. Aunque los principios y conceptos no han cambiado a lo largo del tiempo, los circuitos y métodos con que se realizan han sufrido un gran cambio debido a la constante actualización tecnológica permitiendo

así mejor desempeño, mayor eficiencia y generar costos relativamente bajos al consumidor [23]. El diagrama de un sistema de comunicación electrónica es simple como se visualiza en la Figura 2.1.

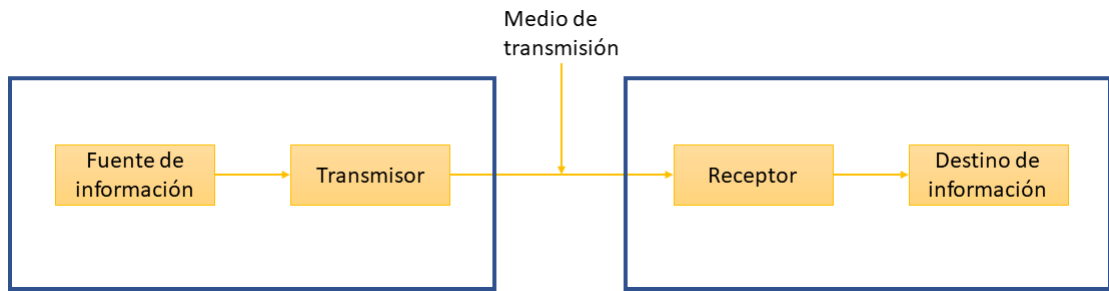


Figura 2.1: Diagrama de un sistema de comunicación electrónica  
Elaborado por: Investigador basado en [23]

#### 2.2.4 Sistema de Comunicaciones en la Agricultura

El uso de nuevas tecnologías para la agricultura es el gran esfuerzo que realizan las personas dedicadas a esta labor, que se la denomina agricultura 4.0.

La agricultura 4.0 aparece a principios del siglo XXI, a través de mecanismos tecnológicos que ayudan a la digitalización en el sector industrial, se las conoce también como agricultura inteligente o agricultura digital. La cuarta revolución industrial se caracteriza por la introducción de servicios digitales basado en procesos, combinando redes de sensores inalámbricos (WSN), agricultura de precisión y Big Data [16]. Además del uso de tecnologías científicas en la agricultura, la tecnología de la información se está ejerciendo en gran medida en esta área. Tecnologías como la navegación por satélite, la red de sensores, la computación en cuadrícula, la computación ubicua y la computación consciente del contexto están apoyando dicho dominio para mejorar las capacidades de monitoreo y toma de decisiones [1].

#### 2.2.5 Sensores en la agricultura

Los avances de las tecnologías y el uso de redes de sensores apoyando las prácticas agrícolas conllevaron a reducir el tamaño cada vez más de los dispositivos, lo que permite utilizar una variedad de modelos con diferentes parámetros. Los sensores se utilizan para obtener información sobre los atributos físicos y ambientales, todos los datos acumulados gracias a ellos se los denomina contexto; aportando en la toma de decisiones y modelado de situaciones que tienen una variedad de atributos como en la parte agrícola.

El dominio de la agricultura plantea varios requisitos como:

1. Recopilación de información acerca del clima, cultivos y el suelo
2. Seguimiento de la tierra
3. Cultivos múltiples en un solo terreno

4. Uso de fertilizantes y agua
5. Condiciones ambientales según el cultivo
6. Soluciones pro activas

Los requisitos anteriormente descritos implican la obtención de información para un posterior procesamiento. Se requiere el uso de sensores inalámbricos en algunos casos para recopilar la información y hacer frente a la toma de decisiones, proporcionando así un entorno controlado basado en un sistema [1].

### 2.2.6 Redes de Sensores Inalámbricos (WSN)

Una red de sensores inalámbricos es un sistema formado por nodos, dispositivos inteligentes livianos y relativamente pequeños que se utilizan para recopilar atributos, incorporan su capacidad de comunicación utilizando ondas de radio, para enviar los valores medidos a una central, para procesarla y convertirla en información útil. Las redes inalámbricas de sensores permiten la comunicación en un espacio y tiempo flexible y de forma autónoma [18]. La restricción energética, el poder de cómputo limitado, la memoria pequeña y la seguridad de los datos son algunos de los problemas sustanciales de las redes de sensores.

Una red de sensores realiza tres funciones básicas: detección, comunicación y computación mediante el uso de hardware, software y algoritmos. Los nodos que se encargan de obtener la información se les denomina nodo de origen, mientras que los nodos que se conectan a otros nodos y sirven de puente para el paso de información se los denomina nodo receptor o nodo puerta de enlace. Cuando en una red de sensores se incorpora el uso de actuadores se la conoce como WSAN ya que aumenta la capacidad de WSN desde el monitoreo hasta el control [1].

### 2.2.7 Componentes y arquitectura de una WSN

Como se explico anteriormente un nodo es el componente básico de una red de sensores inalámbricos (WSN). Un nodo común se compone de cuatro elementos como: sensor, transceptor, memoria externa y fuente de alimentación.

Los componentes para formar una red de sensores se describen en la Tabla 2.1.

Tabla 2.1: Componentes WSN

Componentes	Descripción
Sensores	Distintos tipos y tecnologías convierten la información a señales eléctricas
Nodo Sensor	Capacidad de procesamiento y almacenamiento
Gateway	Elemento para la interconexión entre red de sensores y TCP/IP.
Estación Base	Recolección de datos de los nodos
Red Inalámbrica	Se utiliza el estándar IEEE 802.x

Elaborado por: Investigador basado en [18]



En la Figura 2.2 se muestra la arquitectura básica de los nodos, que tienen la capacidad de detectar, procesar y comunicar variables de interés para acoplarse a un sistema o solamente realizar un monitoreo ambiental.

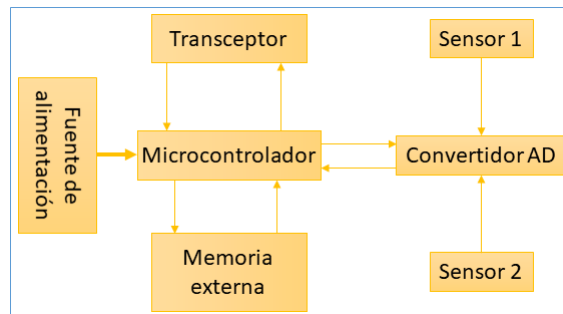


Figura 2.2: Arquitectura básica de un nodo sensor  
Elaborado por: Investigador basado en [2].

### 2.2.8 Topologías WSN

En las redes de comunicación, una topología es una descripción generalmente esquemática de la disposición de una red, incluidos sus nodos y líneas de conexión como se ve en la Figura 2.3.

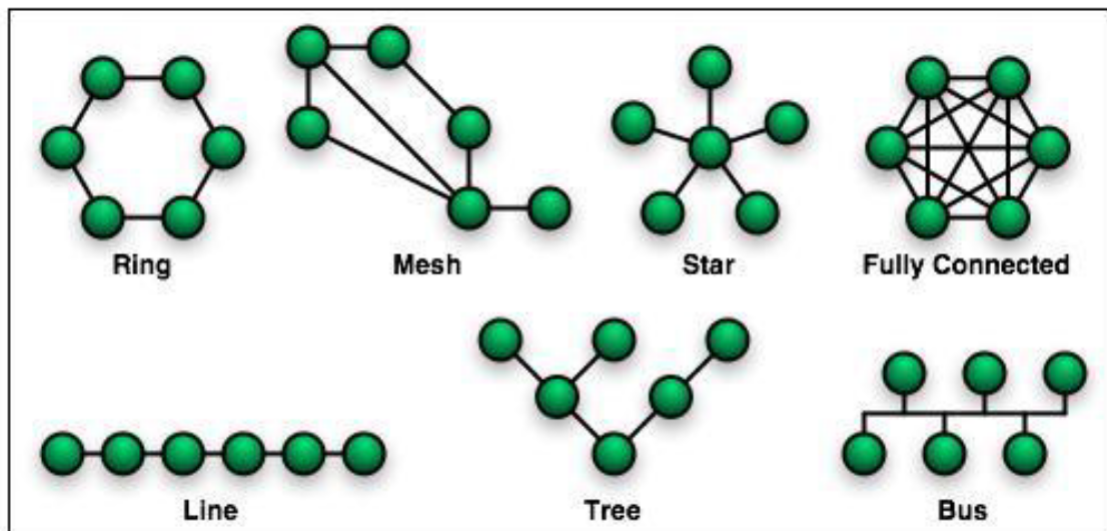


Figura 2.3: Topologías de una Red  
Elaborado por Investigador basado en [11]

Tomando en cuenta la Figura 2.3 donde es relativamente sencillo escoger una de las topologías y empezar a utilizar, se debería tomar en cuenta algunos parámetros para medir la efectividad de cada una de ellas.

#### Parámetros para medir la efectividad de una topología

- **Escalabilidad:** Es la propiedad de poder hacer frente a celdas de red tan pequeñas como unos pocos nodos a celdas de miles o incluso decenas de miles de nodos, así como aumentar el tamaño

de la red existente en un orden de magnitud sin emplear costosas comunicaciones celulares u otras soluciones de largo alcance.

- **Alcance y cobertura:** Estos parámetros son probablemente los requisitos más obvios para una red de sensores inalámbricos (WSN) comenzando por la distancia entre nodo y nodo a una ganancia de antena, potencia de transmisión y velocidad de datos determinadas. Los principales factores que afectan el alcance de una red inalámbrica son la calidad de la capa física y la eficiencia de la transmisión de datos a través de la red. Los requisitos de cobertura son la eliminación de puntos muertos en la red y la extensión del área de cobertura en el rango, los cuales están estrechamente relacionados con el rango.
- **Número de saltos:** El recuento de saltos es la métrica más utilizada en las redes inalámbricas de saltos múltiples. Se selecciona la ruta que tiene el número mínimo de enlaces entre una fuente determinada y el nodo de destino.
- **Consumo de energía:** En la mayoría de los protocolos de comunicación publicados para redes de sensores inalámbricos, la extensión de la vida útil de la red se ha descrito como un importante objetivo. La topología de los nodos puede afectar significativamente la vida útil de la red. Por otro lado, la distribución uniforme de los nodos en una red puede resultar en el agotamiento de la energía de los nodos que están cerca de la estación base a una tasa mayor que la de los otros nodos, lo que a su vez acortará la vida útil de la red [11].
- **Recuento de transmisión esperado (ETX):** Considera la pérdida de datos debido a la contención de acceso al medio y los peligros ambientales y considera el número de transmisiones necesarias para transmitir con éxito un paquete a través de un enlace.

Según el estado del arte, las principales topologías utilizadas en una red de sensores inalámbricos son la de estrella, malla y topología basada en árbol de clústeres tomando en cuenta los parámetros anteriores.

- **Topología tipo estrella**

La topología en estrella de un solo salto es el mas simple, consiste en tener varios nodos (cliente) que se comunican con un nodo concentrador (servidor). Los nodos clientes no tienen retransmisión de datos a través de otros nodos cliente, solo se comunica con el nodo central para el envío de paquetes.

- **Topología basada en árbol de clústeres**

En este tipo de topología, dos o mas redes tipo estrella se comunican mediante un dispositivo central llamado nodo raíz. Cada nodo transmite datos dentro de su clúster para llegar a un nodo concentrador. El nodo concentrador analiza la información enviada por los nodos sensores, la depura y se la envía al nodo raíz. Este tipo de topologías se utiliza para reducir el consumo de energía en un sistema WSN.

- **Topología de malla**

En la topología tipo malla todos los nodos de la red se comunican entre si, es decir actúan como emisores y receptores al mismo tiempo [2].

### 2.2.9 Sistema electrónico

Los sistemas electrónicos se basan en el paso de información desde un lugar hacia otro empleando un conjunto de circuitos que operan con señales eléctricas y se las procesa para realizar una función determinada. Basados en esto el sistema electrónico consta de tres bloques bien definidos que son: bloques de entrada, bloques de proceso y bloques de salida.

- Bloque de Entrada.- El bloque cumple la función de recibir las señales físicas del exterior a través de elementos como sensores o un accionador.
- Bloque de Proceso.- El bloque tiene por objetivo transformar las señales físicas del bloque de entrada en señales eléctricas a partir de ello se lo interpreta, gestiona y elaboran resultados.
- Bloque de Salida.- El bloque se encarga de recibir la señal del bloque anterior para enviarlo hacia un elemento actuador al que este conectado y de ser el caso activarlo o no.

El diagrama del sistema electrónico mediante bloques se puede ver en la Figura 2.4:

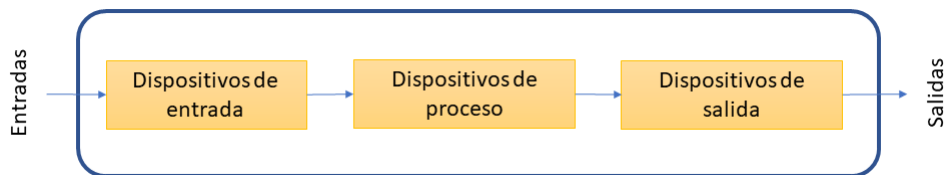


Figura 2.4: Diagrama de bloques de sistema electrónico  
Elaborado por: Investigador

Como ya se lo describió anteriormente el bloque de entrada cumple la función de recibir las señales provenientes del exterior a través de dispositivos (sensores, interruptores, etc.) a los que se les denomina dispositivos de entrada.

### 2.2.10 Dispositivos de Entrada

Los dispositivos de entrada son los encargados de recopilar la información proveniente del exterior a través de sensores, interruptores o algún dispositivo diseñado para obtener señales físicas. De esta manera se adquiere la información para luego ser procesada.

## Sensores

Los sensores son dispositivos con una característica bastante especial que permite imitar la capacidad de percepción de los seres humanos. Dicho esto, se puede encontrar sensores relacionados con cada uno de los sentidos de los seres humanos como son: tacto, gusto, oído y vista, es decir, que reaccionan con el contacto, sustancias, sonidos y objetos a su alrededor. Los sensores son dispositivos electrónicos que permiten interactuar con el entorno, de manera que proveen de información útil de las variables que se encuentran en el medio ambiente para luego ser procesadas y así mismo tomar decisiones como activar procesos, generar ordenes, entre otros.

### Clasificación de los sensores

Dado el gran número de sensores que coexisten en el mercado, se hace necesario clasificarlos para poder entender de mejor manera su funcionamiento y naturaleza. Debido a esto existen varias clasificaciones las que se detalla a continuación:

Según el principio físico de funcionamiento como se observa en la Figura 2.5, se clasifican en:

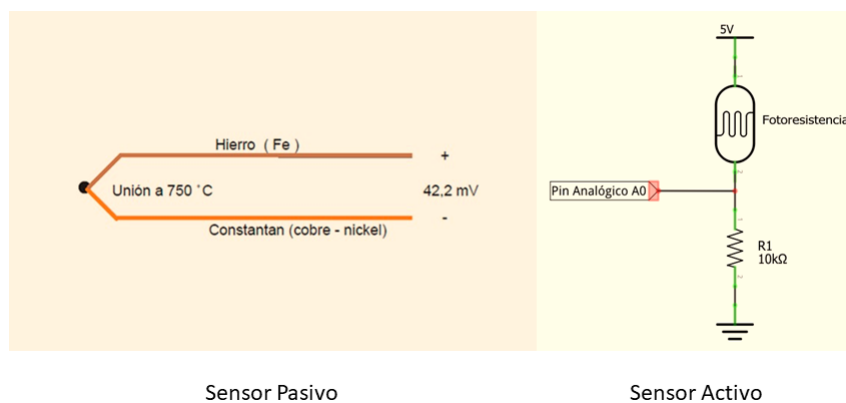


Figura 2.5: Sensores según su funcionamiento  
Elaborado por: Investigador basado en [20].

- **Pasivos.-** El sensor es pasivo cuando la intensidad física a detectar proporciona la energía necesaria para la generación de la señal eléctrica, es decir, no requiere fuente externa de alimentación.
- **Activos.-** El sensor es activo cuando la magnitud a detectar se limita a modificar alguno de las referencias eléctricas características del sensor, adicionalmente necesita de una fuente externa de energía.

Según las señales que generan los sensores como se observa en la Figura 2.6, se clasifican en:

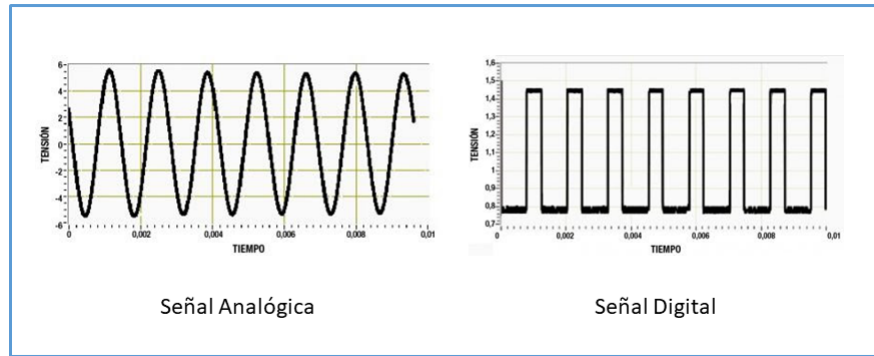


Figura 2.6: Sensores según las señales generadas  
Elaborado por: Investigador basado en [20].

- **Analógicas.-** Se los llama así a los sensores cuya información adquirida es mediante una señal analógica, es decir, que puede tomar infinidad de valores entre mínimos y máximos.
- **Digitales.-** Los sensores digitales brindan la información mediante una señal que puede ser cero “0” o uno “1” lógico, o puede ser un código de bits.

Según el tipo de variable física medida como se observa en la Figura 2.7, se clasifican en:

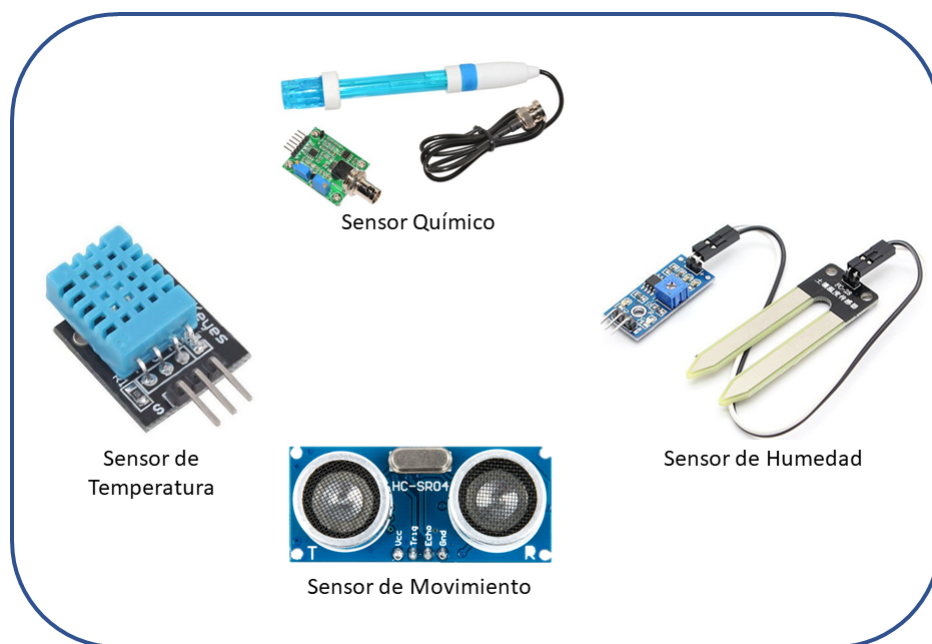


Figura 2.7: Sensores según variable física medida  
Elaborado por: Investigador

- **Movimiento.-** Son sensores que comprueban variaciones en función de los movimientos a los que son sometidos.
- **Posición.-** Son sensores que experimentan variaciones en función de la posición que ocupan en

cada instante de acuerdo a una referencia.

- **Temperatura.-** Son sensores que aprecian variaciones en función de la temperatura del lugar donde se encuentran ubicados.
- **Magnéticos.-** Son sensores que perciben variaciones en función del campo magnético que les atraviesa.
- **Humedad.-** Son sensores que comprueban variaciones en función del nivel de humedad existente en el medio en que se están ubicados.
- **Presión.-** Son sensores que experimentan variaciones en función de la presión a la que son sometidos.
- **Fotoeléctricos.-** Son sensores que aprecian variaciones en función de la luz que incide sobre ellos.
- **Químicos.-** Son sensores que perciben variaciones en función de los agentes químicos externos que pudieran incidir sobre ellos.

#### **Características de los sensores**

Debido a todos los dispositivos que se encuentran hoy en día en el mercado, se debe elegir un sensor para la aplicación que se va a desarrollar, es necesario tener en cuenta determinados parámetros y aspectos para obtener los resultados requeridos.

- Rango de operatividad
- Fiabilidad
- Rapidez de respuesta
- Consumo de corriente
- Tensiones de alimentación
- Radio de acción
- Situación donde van a ser utilizados
- Márgenes de temperatura
- Vulnerabilidad al ruido
- Relación costo/beneficio

De igual manera es preciso conocer características técnicas sobre los dispositivos que se van a utilizar.

De acuerdo a los parámetros siguientes se puede obtener la calidad del sensor:

- **Error.-** Es la desviación de la medida proporcionada por el sensor respecto de la real. Se lo puede expresar en %.
- **Precisión.-** Es el error de medida máximo esperado.
- **Repetitividad.-** Es el error esperado al repetir varias veces la misma medida.
- **Resolución.-** Es la mínima variación de la magnitud de entrada que puede apreciarse a la salida.
- **Sensibilidad.-** Es la relación entre la variación de la magnitud de salida y la variación de la magnitud de entrada.

La Tabla 2.2 muestra el tipo de sensores que habitualmente más se utiliza en función de la aplicación deseada.

Tabla 2.2: Tipos de sensores más utilizados

Aplicación	Sensores
Humedad	Capacitivos, Resistivos
Presión	Piezoeléctricos, Resistivos
Iluminación	Fotoeléctricos, Fotorresistivos (LDR)
Temperatura	Termistores (PTC, NTC, Semiconductores)
Distancia	Infrarrojos, Ultrasonido
Químicos	Detectores de gas y humo
Caudal	Magnetorresistivos, Piezoeléctricos
Frío/Calor	Células termoeléctricas (Peltier)
Presencia	Ultrasonidos, Magnéticos, Infrarrojos
Posición/Inclinación	Mecánicos, Resistivos, Acelerómetros, Magnéticos

Elaborado por: Investigador basado en [20].

### 2.2.11 Dispositivos de proceso

Los dispositivos de proceso se encargan de manipular, interpretar y transformar las señales provenientes de los elementos de entrada como se vio anteriormente. Estos pueden ser dispositivos electrónicos de fuente abierta, es decir, accesibles públicamente a sus diseños.

- **Hardware de Fuente Abierta**

Según la Asociación de Hardware de Fuentes Abiertas (OSHW por sus siglas en inglés) es aquel hardware cuyo diseño se lo hace público para que cualquier individuo lo pueda analizar, modificar, mejorar y vender si así lo requiere, tanto el hardware original como uno modificado por la persona. El hardware también conocido como open source (fuentes abiertas) utiliza componentes estandarizados, de alta disponibilidad, dando libertad a las personas de controlar esa tecnología y poder compartir conocimientos; elevando al siguiente nivel la comercialización de dichos productos[3].

### 2.2.12 Dispositivos de Salida

Los dispositivos de salida son los que se encargan de convertir señales de voltaje o corriente en señales físicas para abrir, mover o realizar una acción determinada. Por lo general los actuadores son dispositivos de salida.

#### **Actuadores**

Los actuadores son dispositivos que ofrecen la oportunidad de transformar diferentes energías recibidas para gestar el funcionamiento dentro de un sistema automatizado determinado. Estos generan una fuerza mecánica a partir de distintos elementos como: energía hidráulica, eléctrica o neumática, las cuales son su fuente de energía. Se puede clasificar a los actuadores en diferentes clases:

- Hidráulicos
- Electrónicos
- Eléctricos

El primero de la lista, el actuador hidráulico su funcionamiento se basa en utilizar líquidos a presión para generar fuerza; un elemento líquido que puede ser aceite y es usado para todo tipo de maquinarias de gran tamaño y peso.

Esta clase de actuadores, generalmente, se caracterizan por ser utilizados en varios tipos de maquinarias de gran tamaño, siendo característica relevante en este tipo de dispositivos una mayor velocidad y resistencia mecánica. Es bastante fundamental en esta clase de sistema que el líquido no tenga posibilidad de filtrarse por ningún lugar, pues podría ser altamente nocivo. Para este tipo de problemas que puedan llegar a surgir es recomendable que se ejecuten mantenimientos periódicos de las maquinarias utilizadas para evitar un mayor riesgo y que las mismas podrían generar si no se toman las medidas y precauciones necesarias.

Los actuadores electrónicos como su nombre lo indica están basados en mecanismos electrónicos para su funcionamiento. En el mercado hay actuadores electrónicos que disipan una gran cantidad de energía, para este tipo de casos se utilizan controladores.

Los actuadores eléctricos son empleados en diferentes artefactos mecatrónicos, entre los cuales están los robots. Algunos tipos de actuadores eléctricos son [17]:

- Electro válvula
- Relé
- Motor de corriente continua



- Nitinol
- Leds

### **2.2.13 Sistema de Información Geográfica**

Los Sistemas de Información Geográfica poseen varias definiciones dependiendo de los técnicos y los usos que se les den, la concepción básica de un Sistema de Información Geográfica es un sistema basado en computadoras que permite la interrelación y manejo de información geográfica, a través de su captura y preparación; manejo, manipulación y análisis y presentación de resultados. La definición expuesta incluye sin decirlo ciertos componentes que conforman a un Sistema de Información Geográfica y que la definen como son los datos, el software, el hardware, los procesos, las metodologías y las personas que permiten la integración y análisis de la información geoespacial [13].

### **2.2.14 GPS Essentials**

Un sistema de información geográfico completo y al alcance de todos con un teléfono celular se llama GPS Essentials. Más que una aplicación, es una herramienta completa de GPS para dispositivos Android que permite navegar, administrar y la creación de puntos de referencia (waypoints), uso de mapas como google maps y geoetiquetado de fotos [12].

### 3 CAPÍTULO III

## METODOLOGÍA

### 3.1 Ubicación

El trabajo de investigación fue desarrollado en la ciudad de Ambato, específicamente en la parroquia de Izamba en las zonas agrícolas. Para efectos de investigación, se seleccionaron tres puntos para la obtención de datos, debido a la colaboración de moradores con el terreno agrícola y posibilidad de acceso para la investigación.

### 3.2 Equipos y materiales

Los equipos y materiales utilizados para el desarrollo de la investigación se presentan en la Tabla 3.1.

Tabla 3.1: Equipos y materiales

Descripción	Valor (USD)
Equipos	1125
Materiales	150
Transporte	100
Varios	150
<b>Total</b>	<b>1525</b>

Fuente: Elaborado por el investigador.

### 3.3 Tipo de investigación

#### 3.3.1 Investigación Bibliográfica

La presente investigación mantiene un enfoque cuantitativo, llegando a lo correlacional entre las distintas variables de su estudio, lo cual permitió buscar la solución al problema planteado. El análisis de la información permitió realizar el diseño de la investigación de la propuesta planteada. Se ha seleccionado la metodología de investigación bibliográfica debido a que gran cantidad de información requerida para la implementación del estudio, pudo hallarse de forma dispersa en artículos científicos, publicaciones de estudios de maestría o doctorado y libros o manuales provistos por desarrolladores y fabricantes.

#### 3.3.2 Investigación de Campo

La investigación de campo se la realizó debido a la recolección de datos en sectores dedicados a la agricultura, en la zona rural de la ciudad de Ambato, específicamente en la parroquia Izamba.

## 4 CAPÍTULO IV

### ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

#### 4.1 Parroquia Izamba

Debido a la naturaleza de una investigación de campo, se requiere conocer algunos detalles de la parroquia donde se va a desarrollar la investigación, en este apartado se analizará la geografía de la zona, el número de hectáreas (ha) dedicadas a la agricultura y demás características que sirvan como referencia para el desarrollo de la investigación.

##### 4.1.1 Ubicación Geográfica y Límites

Izamba con una superficie de 29,04 Km<sup>2</sup> es una de las 18 parroquias rurales del cantón Ambato, se encuentra ubicada en la parte norte del cantón, provincia de Tungurahua. Se encuentra ubicada a 5 km de la urbe central del cantón Ambato.

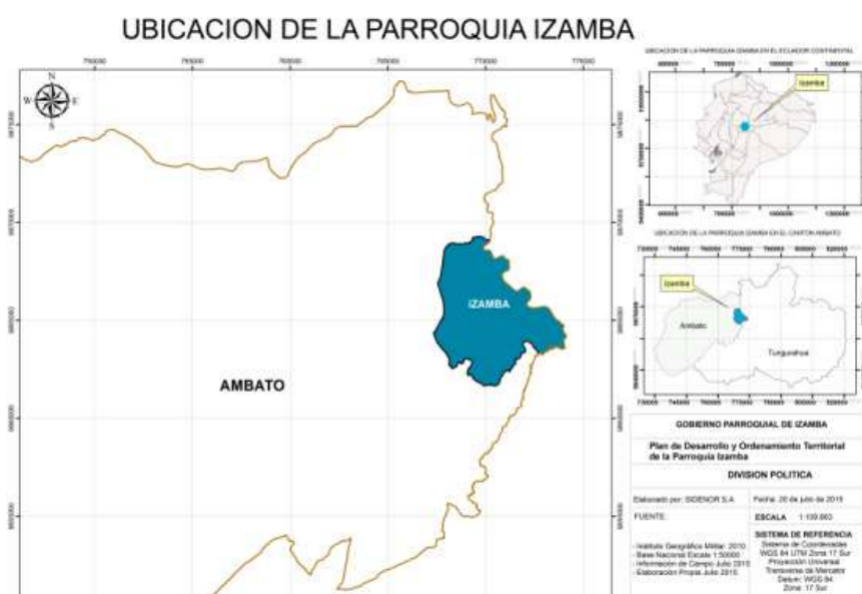


Figura 4.1: Ubicación de la parroquia Izamba

Fuente: Cartografía base 1:50000 IGM

Elaborado por: Equipo Técnico PDyOT 2015

Según datos del Instituto Nacional de Estadísticas y Censos (INEC) la parroquia de Izamba tiene una población de 14.543 habitantes. Limita con las parroquias: Unamuncho, río Ambato, Atahualpa y el río Culapachán.

- **Relieve**

**Altitud:** La parroquia Izamba revela altitudes que van desde los 2239 hasta los 2670 msnm. El terreno es de una condición bastante irregular marcado en su mayoría por pendientes medias y altas y en algunos

sectores es de tipo accidentado en su superficie. La parroquia se encuentra en medio de quebradas como: Chasinato, Quindulli, Huagra Corral, Pisocucho y Quebrada Seca.

- **Suelo**

Casi el 40% de los suelos corresponden al Orden de Inceptisoles, cuyo suelo se caracteriza por ser de tipo color negro, aspecto arenoso, tiene arena fina a media, profundo, con manchas de color rojizo y claro, por lo general saturado de agua a menos de 1 m de profundidad. El porcentaje restante corresponde a suelos de Orden Mollisoles con otras características similares a ecosistemas de pastizales como se lo describe en la Tabla 4.1.

Tabla 4.1: Tipos de Suelo de la parroquia Izamba

<b>Orden</b>	<b>Suborden</b>	<b>Descripción</b>	<b>Superficie (ha)</b>	<b>Porcentaje (%)</b>
INCEPTISOL	ANDEPTS	Cangahua a 30/40 cm. de profundidad. Cangahua dura a 20cm de profundidad con costra de carbonato. Suelo arenoso, negro, arena fina a media, profundo, con manchas de color rojizo y claro.	1120.85	38.59
MOLLISOL	USTOLL	Son los sedimentos de los ecosistemas de pastizales. Se caracterizan por un horizonte superficial de espesor, superficie oscura. Este horizonte fértil, conocido como un epipedón mólico, son consecuencia de la adición a largo plazo de ingredientes orgánicos derivados de raíces de las plantas.	1783.44	61.41
<b>TOTAL</b>			<b>2904.29</b>	<b>100</b>

Elaborado por: Investigador

- **Cobertura del Suelo**

La mayoría de los suelos casi el 80% son de uso agrícola. Posee uno de los suelos mejores en su tipo y apto para todo tipo de cultivo ya sea de ciclo cortos o largos, en algunas zonas crece cangagua donde ha dominado la erosión. La parroquia posee una gran variedad de suelos que son considerablemente fértiles y entre estos se tiene a los suelos rocosos, suelos arcillosos, suelos arenosos y zonas donde los suelos no son suficiente para el cultivo como las laderas que la mayoría de las veces son muy empinadas y de difícil acceso para las población que allí habita y por lo tanto no son aptas para el cultivo, solamente han crecido plantas propias de la zona, que crecen con ayuda de las lluvias y desaparecen cuando el calor es extremadamente intenso. Los principales cultivos agrícolas son: coliflor, cebolla, cilantro, lechuga, nabo, acelga, col, perejil, apio, remolacha, tomate de árbol y maíz.

En la Tabla 4.2 se describe el uso que tiene el suelo y la superficie utilizada tanto en hectáreas de terreno como en porcentajes.

Tabla 4.2: Uso actual del suelo parroquia de Izamba

Uso de Suelo	Superficie (ha)	Porcentaje (%)
Cultivos	1163.07	40
Pastos y cultivos	1163.07	40
Pastos	289	10
Zonas Erosionadas	279	9.7
Zonas extractivas e industriales	10	0.3
<b>TOTAL</b>	<b>2904.29</b>	<b>100</b>

Elaborado por: Investigador

Como se observa en el cuadro la mayoría de zonas están destinadas para el uso de la agricultura, mediante la Figura 4.2 se aprecia con mayor detalle este aspecto de la parroquia. Tomando en cuenta que para la investigación se requiere de suelos agrícolas fértiles, la selección de la parroquia Izamba como punto estratégico esta totalmente justificada, por así decirlo.

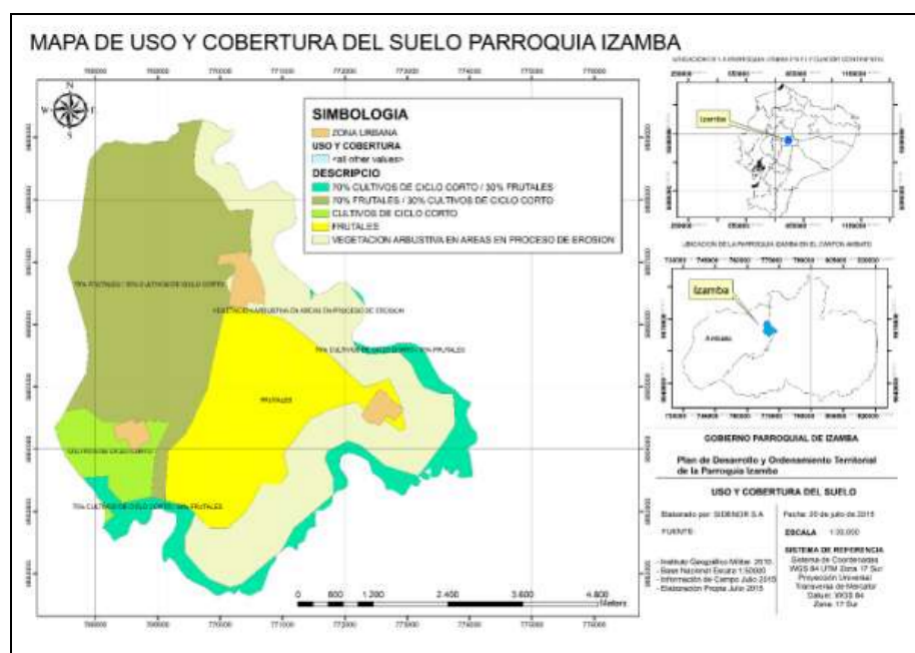


Figura 4.2: Cobertura y Usos del Suelo de la Parroquia de Izamba

Fuente: Cartografía base 1:50000 IGM

Elaborado por: Equipo Técnico PDyOT 2015

## 4.2 Factores edafoclimáticos

Como se vio en el marco teórico, los factores edafoclimáticos corresponde a una selección de parámetros ambientales y de suelo que afectan al momento de realizar el proceso de labranza desde su plantación hasta su posterior cosecha. La mayor parte de agricultores toman decisiones empíricamente, es decir, no tienen un fundamento teórico científico en el cual se basan para determinar que cultivo van a plantar,

si se lo realiza de manera adecuada o peor aún considerando los factores edafoclimáticos. El proceso de cultivo desde el minuto cero debe tomar en cuenta muchas consideraciones como la fertilidad del suelo, condiciones climáticas, entre otros. Como se revisó anteriormente la mayoría del terreno ubicado en la parroquia de Izamba es fértil gracias a la ubicación geográfica de este sector por lo que no se necesitará realizar una evaluación de fertilidad, pero existen factores químicos y físicos que afectan directamente al crecimiento de la planta como son el pH, la humedad del suelo y la temperatura. Los tres factores descritos forman parte de los requerimientos edafoclimáticos a evaluar al momento de realizar una siembra.

#### 4.2.1 pH

El primer factor químico relevante dentro del proceso de siembra es el pH. Para comprender el concepto de acidez del suelo primero se analiza el concepto de un ácido que es un compuesto químico que transfiere protones, lo cual la acidez es la actividad de iones de hidrógeno ( $H^+$ ) de una solución. Para determinar la acidez del suelo se realiza por medio de la medición, para comparar la actividad de  $H^+$  presente en la materia del terreno y se expresa en la unidad pH (potencial hidrógeno). De acuerdo al pH obtenido en el suelo se clasifican en: Ácido, Neutro y Alcalino. En la Tabla 4.3 se muestra el rango de pH y sus características de acuerdo a su clasificación mencionada por la Organización de las Naciones para la Alimentación y Agricultura (FAO) .

Tabla 4.3: Rango de pH

pH en el suelo	Rango	Valor	Descripción
Alcalino	Muy Alcalino	$pH > 8,5$	No asimilan nutrientes. Dificultad para el desarrollo de cultivos exigentes.
	Ligeramente Alcalino	$7,5 < pH \leq 8,5$	
Neutro	Neutro	$6,5 < pH \leq 7,5$	Este es el rango más óptimo, indica buena fertilidad de suelo por ende condiciones óptimas para el desarrollo de cultivos y asimilación de nutrientes
Ácido	Ligeramente Ácido	$5,5 < pH \leq 6,5$	No son favorables para el crecimiento de raíces. Reduce actividad microbiana
	Muy Ácido	$pH \leq 5,5$	

Elaborado por: Investigador

#### 4.2.2 Humedad del Suelo

La humedad del suelo es un factor fundamental que permite determinar la disponibilidad y cantidad de nutrientes que existen en un terreno, puesto que las raíces de las plantas se abastecen de nutrientes que están presentes en forma iónica en la solución del terreno. Si no hay solución, no hay posibilidad de que haya nutrientes para el cultivo disponibles. Si en el suelo no hay humedad, los nutrientes no se solubilizan ( minerales, fertilizantes), de tal motivo no hay reacción de equilibrio entre la fase sólida y la solución, debido a esto no ocurriría la mineralización de la materia orgánica.

La humedad del suelo se la considera a la cantidad de agua por volumen de tierra que contiene un

terreno, sin esta capacidad la semilla recién sembrada no fecundaría debido a que si existe poca cantidad de agua la planta se marchita y al contrario si abunda la cantidad de agua en un terreno la planta tiende a no desarrollar como debería por lo cual este segundo factor es de similar importancia.

#### 4.2.3 Temperatura Ambiental o Factor Climático

Si bien es cierto el pH y la humedad del suelo son factores de vital importancia a la hora de plantar una semilla, así mismo el factor climático cumple una importante función dentro de la labor agrícola. Debido a la temperatura ambiental del sector en el que se realiza la actividad agrícola se puede determinar el tipo de cultivo apto para su germinación. Debido a este gran factor se puede clasificar el tipo de plantaciones por regiones en el territorio nacional, es decir, los cultivos que se dan en la costa no pueden hacerlo en la Sierra debido al clima que impera en estas dos regiones tan diversas en cuanto a propiedades agrícolas. Es por ello que la temperatura juega un papel estratégico a la hora de la toma de decisiones.

### 4.3 Tipos de cultivos

De acuerdo a la información obtenida mediante la investigación bibliográfica se realizó una Tabla 4.4 con los productos agrícolas que más acogida tienen dentro de la región sierra y tomando en cuenta los factores edafoclimáticos anteriormente vistos dio como resultado lo siguiente:

Tabla 4.4: Tipos de cultivos según requerimientos edafoclimáticos

Cultivo	Temperatura Mínima (oC)	Temperatura Máxima (oC)	Humedad Mínima (%)	Humedad Máxima (%)	pH Mínimo	pH Máximo
Papa	13	18	45	55	5.5	6
Fresa	15	20	65	70	6	7
Trigo	10	24	50	60	7	8
Cebada	19	26	35	50	6.1	7.2
Zanahoria	15	25	50	70	5.8	6.8
Cebolla Paiteña	13	24	60	70	6	7
Café	18	21	70	95	5.6	6.5
Manzana	18	24	40	60	5.5	7
Habas	15	20	45	50	5.7	6.8
Quinua	10	17	30	45	5.5	8
Maíz	19	24	30	40	5.5	7.8

Elaborado por: Investigador

### 4.4 Requerimientos Técnicos

Una vez analizado el estado del arte y seleccionado las variables edafoclimáticas a evaluar, se ha llegado a determinar el diagrama de bloques en la Figura 4.3 para el sistema de información geográfica que tendrá como ubicación la parroquia Izamba.



Figura 4.3: Diagrama de Bloques  
Elaborado por: Investigador

En el primer bloque se va a realizar la adquisición de información mediante sensores específicos para el sistema. La segunda etapa se realiza el procesamiento en la cual se lleva a cabo el acondicionamiento de las señales para poder interpretarlas y enviarlas. El tercer bloque es el de comunicación en la que se realiza el traslado de información de un lugar hacia otro inalámbricamente, como último punto se facilita la parte de presentación donde van a llegar la información una vez completado el proceso.

- **Adquisición de Información**

Una de las partes primordiales en un sistema de recolección de información es la adquisición de datos a través de algún tipo de sensor o transductor, el cual es de gran beneficio a la hora de obtener datos de las variables físicas desconocidas por el usuario. Se puede realizar de dos formas ya sea de manera analógica o digital.

Las principales variables a tomar en cuenta dentro de la agricultura es la humedad del suelo, pH y la temperatura, estas dos últimas se las puede realizar con un mismo sensor como se lo indica en la Figura 4.4. Los sensores deben ser de alta precisión para tener un margen de error mínimo, en cuanto al sensor de pH se debe tomar en cuenta la compatibilidad, rapidez en la respuesta y el rango de operación tanto del pH como de la temperatura.

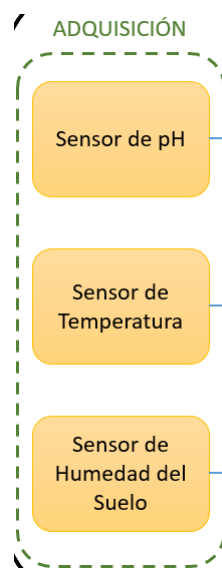


Figura 4.4: Etapa de Adquisición de Información  
Elaborado por: Investigador

En cuanto a la recolección de información de los datos se verificará el tipo de topología adecuada.



- **Procesamiento**

A través del uso de dispositivos electrónicos de open source o fuente libre, se puede realizar el procesamiento de la información obtenida a través de los nodos sensores, estos dispositivos tiene el compromiso de recibir las magnitudes provenientes de los sensores y transformarlos en valores numéricos de fácil interpretación para el usuario final. De tal manera cada vez que el sensor obtenga la información del exterior, se realiza el procesado de dicha información por el dispositivo y se da paso hacia el dispositivo de comunicación.

- **Comunicación**

El objetivo fundamental de un sistema electrónico de comunicaciones, es el envío de información de un lugar hacia otro. La comunicación es la transmisión, recepción y procesamiento de información a través de circuitos electrónicos.

El dispositivo encargado de enviar los datos desde la fuente hacia el destino, deberá tener amplio rango de alcance, un consumo energético bajo y además transmitir un ancho de banda adecuado, es decir que abarquen los datos a transmitir.

- **Presentación**

Una vez realizada la comunicación del sistema, se procede a realizar la presentación de la información, la misma que se visualizará de manera ordenada, ya que el sistema tendrá que identificar y realizar las debidas comparaciones con la información obtenida de los sensores y la información de la Tabla 4.4. De tal manera, que pueda ser interpretada de la mejor forma por el usuario a la que va dirigido la presentación, en este caso a las personas dedicadas a la labor agrícola dentro de la parroquia.

#### **4.4.1 Diagrama General del Sistema**

Una vez realizado el análisis en cada uno de los bloques a implementar, se obtiene una visión mas detallada del sistema, graficando un diagrama específico que servirá para el análisis técnico posterior y valoración tomando como referencia la Figura 4.5.

## NODO SENSOR

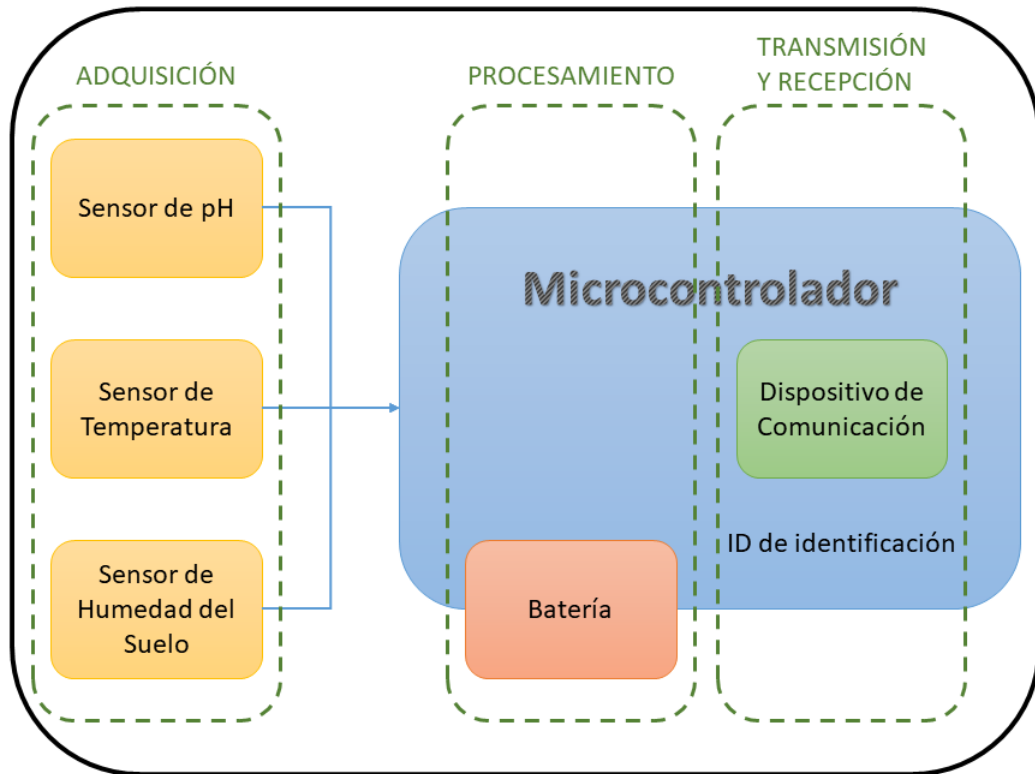


Figura 4.5: Diagrama General del Sistema  
Elaborado por: Investigador

En la red de comunicación que se va a diseñar se debe tomar en cuenta la topología que se va a utilizar con el fin de sacar el mayor beneficio de los dispositivos. Es por ello que se realiza el análisis técnico del hardware y software que se utilizará para realizar el diseño del prototipo. Se tendrá en consideración algunas recomendaciones del estado del arte y revisión bibliográfica que se ha realizado durante la investigación.

### 4.5 Análisis Técnico del Hardware Requerido

#### 4.5.1 Microcontrolador

Hoy en el mercado se puede encontrar gran variedad de placas de fuente libre destinadas al procesamiento de señales, en la Tabla 4.5 se realiza una comparativa de los principales microcontroladores, tomando como referencia la accesibilidad, parámetros técnicos, y atributos que brinden al sistema el mejor desempeño posible.

Tabla 4.5: Comparación Técnica de los Microcontroladores

<b>Microcontrolador</b>				
<b>Parámetros Técnicos</b>	<b>Raspberry Pi Pico</b>	<b>Arduino Nano</b>	<b>ESP32</b>	<b>STM32 Blue Pill</b>
Controlador	RP2040 Dual Core	ATMEGA328 Single Core	D0WD-V3 Dual Core	STM32 Single Core
Arquitectura	32 bit	8 bit	32 bit	32 bit
Reloj	48 MHz	16 MHz	80 - 240 MHz	72 MHz
RAM (Kilobytes)	264	2	520	20 - 128
Flash	2 MB	32 KB	4 MB	64 - 524 KB
EEPROM	No	1 KB	512 B	No
Lenguajes de Programación	Micro Python C/C++	Arduino IDE	C/C++, Python, Arduino IDE	C/C++
Voltaje (V)	5	5	5	5
GPIO	26	22	36	37
Pins ADC	3x12 bit	6x10 bit	18x12 bit	10x12 bit
Pins SPI	2	1	4	2
Pins PWM	16	6	16	12
Conectividad Inalámbrica	-	-	WIFI y Bluetooth	-
Consumo Energía (mA)	600	15	80	0.04
Precio (\$)	12	8	10	6

Elaborado por: Investigador

Analizando los parámetros técnicos de la Tabla 4.5 y los beneficios de cada una de las placas electrónicas se ha optado por elegir el microcontrolador ESP32 debido a su rápida respuesta, el número de entradas y salidas que posee ya que ayudará a la escalabilidad de la red, la resolución debido a los pins ADC por lo que se garantizará mediciones muy cercanas al valor real de las variables a medir y como punto decisivo sobre los otros microcontroladores es la conectividad inalámbrica, lo cual no se tiene que comprar un dispositivo de comunicación o módulo adicional para establecer la comunicación entre dispositivos de iguales características. Cabe destacar que el dispositivo al tener este módulo incluido y realizando las configuraciones necesarias a nivel de software, ayudaría en gran medida a la comunicación de la WSN.

#### 4.5.2 Sensor de Temperatura y pH

En el caso de los sensores de temperatura y pH, está claro que se puede conseguir los dos sensores por separado, sin embargo, las variables físicas a medir se lo puede realizar con un solo dispositivo, de esta manera se reduce el costo, además el principal requerimiento para este sensor es su compatibilidad con el microcontrolador, rango de operación y la exactitud que nos pueda brindar. Es por ello que se presenta la Tabla 4.6.

Tabla 4.6: Características de los sensores de temperatura y pH

<b>Sensores de Temperatura y pH</b>			
<b>Parámetros Técnicos</b>	<b>pHure</b>	<b>CPS31D</b>	<b>PH 4502</b>
Precio (\$)	60	215	46
Temperatura de Funcionamiento (°C)	0 - 80	0 - 80	0 - 80
Sensor de Temperatura	Pt 1000	NTC 30K	DS18B20
Rango pH	1 - 11	1 -12	0 -14
Presión (bar)	0 - 7	1 - 4	-
Accesibilidad	Determinadas Zonas Geográficas	Cualquier parte	Cualquier parte
Compatibilidad con microcontrolador	No	No	Si
Información	Escasa	Escasa	Abundante

Elaborado por: Investigador

Una vez analizado en la Tabla 4.6 las características de los sensores puestos a consideración se ha determinado al Sensor PH-4502 como el único compatible con el microcontrolador a utilizar y debido a la escasa información provenientes de los otros rivales será el encargado de realizar el funcionamiento de sensado tanto de la temperatura como de pH lo que garantiza la compatibilidad y por ende un fácil tratamiento de los datos relativamente.

#### 4.5.3 Sensor de Humedad del Suelo

Para la medición de la humedad del suelo se requiere un dispositivo de bajo costo con una alta sensibilidad y una respuesta rápida para que el sistema sea altamente fiable y preciso. En la Tabla 4.7 se considera algunos parámetros técnicos según las necesidades de la investigación.

Tabla 4.7: Características del Sensor de Humedad del Suelo

<b>Sensor de Humedad del Suelo</b>			
<b>Parámetros Técnicos</b>	<b>SHT10</b>	<b>Capacitivo Humedad</b>	<b>FC-28</b>
Voltaje Alimentación (V)	5	3.3 - 5	3.3 - 5
Corriente Operación (mA)	10	5	35
Rango de Humedad (%)	0 - 100	0 y 100	0 - 100
Voltaje de Salida (V)	0 - 5	0 - 3	0 - 5
Tipo de Salida	Analógico	Analógico	Analógico y Digital
Compatibilidad con microcontrolador	Si	Si	Si
Tipo de Sensor	Capacitivo	Capacitivo	Resistivo
Resolución	Alta	Baja	Alta
Precio (\$)	40	4	2.5
Respuesta Rápida	Si	No	Si

Elaborado por: Investigador

Después de realizar el análisis de los parámetros técnicos de los sensores detallados en la Tabla 4.7 se ha decidido optar por el FC-28, debido a su respuesta rápida, el tipo de salida que genera y sobretudo el parámetro que influyo en gran medida es el precio por lo que se requiere de un sistema de bajo costo pero con grandes beneficios como lo es de este sensor.

#### **4.5.4 Dispositivo de Comunicación**

Al momento de realizar el diagrama de bloques de los requerimientos técnicos se evidenció que los datos obtenidos por los sensores deben enviarse hacia otro lugar para realizar el procesamiento de la información, con este antecedente se delineó un diagrama general del sistema en la Figura 4.5, una vez realizado el análisis técnico del hardware requerido y en su primera sección donde se seleccionó el microcontrolador que se va a utilizar dentro del proyecto de investigación, se tomó el ESP32 por un parámetro esencial que es el dispositivo de comunicación que ya viene incluido dentro del dispositivo, obteniendo así un mayor beneficio por parte del microcontrolador y a un costo relativamente bajo ya que los demás no contaban con conectividad inalámbrica propia. Es por esto que en este apartado no se considera necesario una comparativa de parámetros técnicos a evaluar.

Haciendo un breve hincapié al microcontrolador seleccionado y tomando en cuenta el estado del arte previamente estudiado se requiere optar por una topología adecuada para la adquisición de señales. Una topología es una descripción generalmente esquemática de la disposición de una red, es decir, como estarán ubicados los nodos sensores para obtener la información deseada. Como ya se vio anteriormente existen diferentes topologías que se utilizan mediante la red de sensores inalámbricos (WSN) entre las cuales destacan una red tipo malla tomando en cuenta los parámetros para medir la efectividad de la topología. Por lo tanto se seleccionará la topología tipo malla para la investigación presente.

#### **4.5.5 Análisis del Software**

El software que se seleccione deberá ser compatible con el hardware open source que se ha elegido previamente, destacando el manejo para el usuario de una forma sencilla, además el costo debe ser mínimo por lo que la mejor opción sería un software de código abierto y gratuito.

En este caso se requiere visualizar el contenido de cada una de las variables, después del respectivo proceso y desarrollando un algoritmo para la comparación con los tipos de cultivos para así obtener una interfaz de fácil interpretación por el usuario final al que se quiere llegar, debido a que la mayoría, es gente que trabaja en la labor agrícola.

#### **4.6 Adquisición de Señales**

La sección más relevante del sistema es el apartado de adquisición de las señales, el sistema tiene un alto porcentaje de dependencia de las variables con las que se va a trabajar que va a ayudar a la confiabilidad del mismo. Las señales del estudio deben ser tratadas de forma responsable e íntegra para no tener ninguna lectura errónea que posteriormente pueda alterar su procesamiento y por ende su presentación final. Cabe tomar en cuenta que las señales involucradas dentro de la investigación son altamente valiosas y cruciales.

#### **4.6.1 Sensor PH-4502**

Como se vio en el apartado de factores edafoclimáticos acerca del pH, los valores van desde 0 hasta 14. Donde el PH 0 será muy ácido, pH 7 será neutro y el pH 14 muy alcalino. El agua está cerca de un pH 7 y esto es generalmente por aquí que se va a necesitar para tomar valores de pH dentro del sistema. Primero se examinará la placa del módulo de sonda de pH y luego la sonda de PH porque tanto la sonda de PH como el sensor tienen que ajustarse correctamente:

- Ajuste de compensación límite.
- Bosquejo para probar el rango analógico del tablero.
- Bosquejo para lectura de PH y calibración.
- Calibración de la sonda PH.
- Uso de la sonda de PH.

Para ello se debe tomar en cuenta cada parte del módulo del sensor como se lo detalla en la Figura 4.6.

## PINOUT

### Sensor de PH Liquido PH-4502C con electrodo E201-BNC

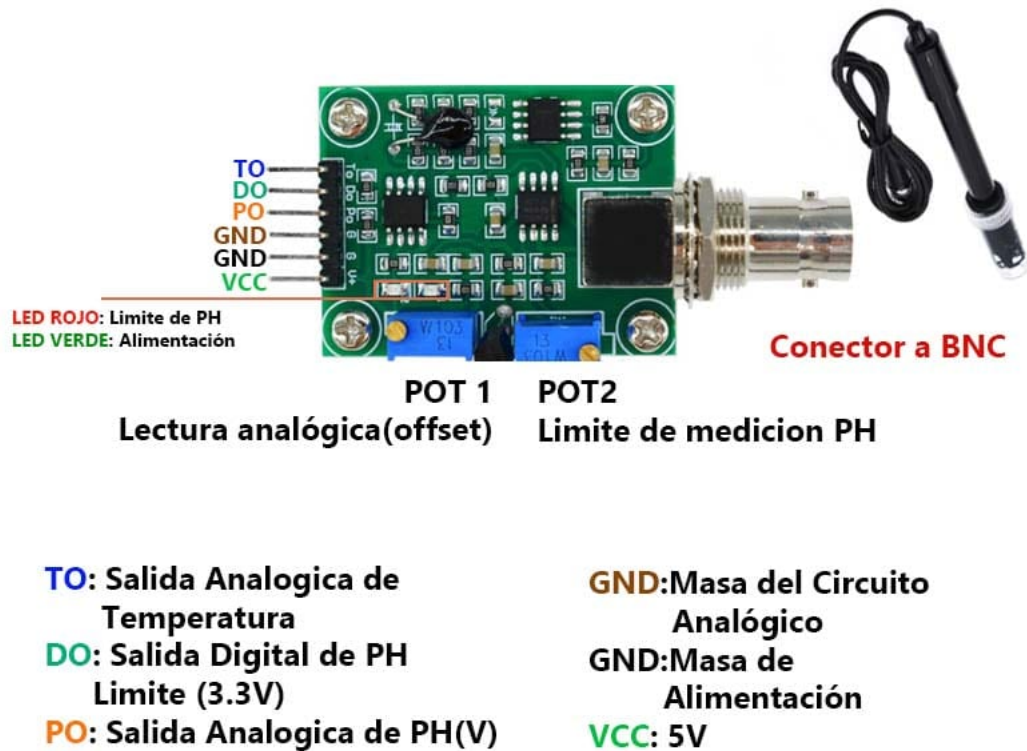


Figura 4.6: Pinout Sensor PH-4502  
Elaborado por: Investigador

Esta placa que viene incluida con el electrodo tiene la capacidad de suministrar una salida de tensión a la placa analógica que representa un valor de PH al igual que cualquier otro sensor que se conecta a un pin analógico. Idealmente, se quiere que un pH 14 represente 5V y un pH 0 represente en este caso 0V. Este dispositivo por defecto tiene un pH neutro fijado a 0V (o cerca de él, se diferencia de otros electrodos de pH, es por eso que se tiene que calibrar la sonda como se verá más adelante), esto significa que la tensión que se va a leer los valores ácidos de pH y que no puede ser leídos por el puerto analógico del microcontrolador. El potenciómetro del desplazamiento se utiliza para cambiar esto de modo que un pH neutro igual a 7 leerá la mitad del voltaje referencial, es decir, 2.5V esperado en la entrada analógica del microcontrolador, la entrada análoga puede leer voltajes entre 0V y 5V, por lo tanto el 2.5V que está a medio camino entre 0V y 5V, como un pH neutro de valor 7 esta a medio camino entre pH 0 y pH 14, se tendrá que girar el potenciómetro de desplazamiento para obtener el desplazamiento derecho, el potenciómetro de desplazamiento es el potenciómetro azul más cercano al conector BNC que limita la

medición de pH.

Configurar el desplazamiento es fácil. En primer lugar, se debe desconectar la sonda del circuito y cortocircuitar el interior del conector BNC con el exterior para generar un pH neutro, en este caso (pH = 7). Para realizar este procedimiento con un trozo de alambre, se envuelve el un lado alrededor del exterior del conector BNC o lo que podría llamarse tierra y se empuja el otro lado en el agujero BNC de donde estaría la sonda conectada. Este cortocircuito representa aproximadamente una lectura de pH neutral con un valor de 7.

Hay dos maneras de realizar el ajuste, una de ellas es a nivel de hardware y la otra a nivel de software. Para un ajuste a nivel de hardware se deberá realizar lo siguiente. Si tiene un multímetro a mano se puede medir el valor del pin P0 y ajustar el potenciómetro de desplazamiento hasta que P0 mida la mitad de la tensión 2.5V.

Por otro lado para realizar una calibración a nivel de software se deberá escribir un código básico de lectura analógica como se observa en la Figura 4.7, es decir, verificar mediante el programa los voltios que se tiene en la salida analógica mediante el monitor serie del Arduino IDE, posterior a ello se ajustará el potenciómetro para que el valor de voltaje llegue a los 2.5V exactamente debido a que el pH neutral tiene un valor 7 y esta en medio de la escala.

```
void setup() {  
  // initialize serial communication at 9600 bits per second:  
  Serial.begin(9600);  
}  
  
// the loop routine runs over and over showing the voltage on A0  
void loop() {  
  // read the input on analog pin 0:  
  int sensorValue = analogRead(A0);  
  // Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):  
  float voltage = sensorValue * (5.0 / 1023.0);  
  // print out the value you read:  
  Serial.println(voltage);  
  delay(300);  
}
```

Figura 4.7: Algoritmo de Valor Sensor en Voltaje  
Elaborado por: Investigador

- **Ajuste límite de pH**

El otro potenciómetro actúa como un interruptor del límite. Básicamente, el pin D0 en la placa de sensores suministrará 3.3V al pin hasta que se alcance un valor PH preestablecido, en este punto un LED de color rojo se encenderá y el pin se reducirá a aproximadamente 0V.



Para la investigación en curso no será necesario este tipo de ajuste debido a que no se tiene actuadores de salida, pero sería de gran ayuda si se requiera activar un zumbador o cualquier otro actuador dependiendo del límite de pH que alcance cierto valor.

- **Conexión y calibración de la sonda pH.**

1. Las lecturas de las sondas cambian con el tiempo y deben calibrarse de vez en cuando para asegurarse de que el valor sigue siendo el mismo y se ajusta si llega a cambiar en algún momento.
2. Necesita al menos una solución tampón pH para calibrar su sonda pH. Están disponibles en muchos valores de pH diferentes, una solución de búfer de 7.76 y 5.01 es más común, ya que cubre la gama de la mayoría de las aplicaciones. Si sólo va a utilizar una solución de búfer, asegúrese de que su valor esté cerca del rango de valores que utilizará en sus pruebas normales - si es agua para la labor agrícola, una solución de búfer cercana a los 7 suele estar lo suficientemente cerca.
3. Una sonda de pH se toma algún tiempo para llegar a la magnitud correcta, permitir que esté en el líquido que desea medir durante al menos un par de minutos o más, no significa que será estable en un valor de pH, saltará alrededor de un bit entre 4 o 5 valores, pero en el último dígito, por ejemplo, entre 5.83 - 5.88.
4. Los valores de pH difieren en diferentes temperaturas, aunque eso podría sonar un poco engorroso, en el rango de temperatura entre 10 - 30 grados Celsius el pH no difiere y de 30 grados Celsius en adelante sube con un pH de aproximadamente 0.01, a 50 grados Celsius es de aproximadamente 0.06 el rango de error.

Para determinar una calibración del pH de la sonda se realiza el siguiente algoritmo que se visualiza en la Figura 4.8.

```

float calibration = 0.00; //change this value to calibrate
const int analogInPin = A0;
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;
void setup() {
  Serial.begin(9600);
}

void loop() {
  for(int i=0;i<10;i++)
  {
    buf[i]=analogRead(analogInPin);
    delay(30);
  }
  for(int i=0;i<9;i++)
  {
    for(int j=i+1;j<10;j++)
    {
      if(buf[i]>buf[j])
      {
        temp=buf[i];
        buf[i]=buf[j];
        buf[j]=temp;
      }
    }
  }
  avgValue=0;
  for(int i=2;i<8;i++)
  avgValue+=buf[i];
  float pHVol=(float)avgValue*5.0/1024/6;
  float pHValue = -5.70 * pHVol + calibration;
  Serial.print("sensor = ");
  Serial.println(pHValue);
}

```

Figura 4.8: Algoritmo de Calibración de pH  
Elaborado por: Investigador

En el caso del sensor PH-4502, la calibración cambio por un valor de 25 según las muestras de diferentes

pH y para el agua dando como resultado pH igual a 7, llegando a tener una buena calibración; debido a que los valores a medir en el entorno agrícola deberían estar cerca del pH neutro por lo cual se realizó la calibración cercana a este.

#### 4.6.2 Sensor de Humedad del Suelo

Un higrómetro de suelo FC-28 es un sensor que mide la humedad del suelo. El FC-28 es un sensor sencillo que mide la humedad del suelo por la alteración de su conductividad. El FC-28 se distribuye con una placa de medición estándar que permite obtener la medición como valor analógico o como una salida digital, activada cuando la humedad supera un cierto umbral.

La salida digital dispara cuando el valor de humedad supera un cierto umbral, que se ajusta mediante el potenciómetro. Por tanto, obtendremos una señal LOW cuando el suelo no está húmedo, y HIGH cuando la humedad supera el valor de consigna.

El valor concreto dependerá del tipo de suelo y la presencia de elementos químicos, como fertilizantes. Además, no todas las plantas requieren la misma humedad, por lo que lo mejor es que se realice una pequeña calibración en el terreno real.

De igual manera el diagrama de conexión del sensor FC-28 es bastante sencillo. Se tiene cuatro pines distribuidos según la Figura 4.9.

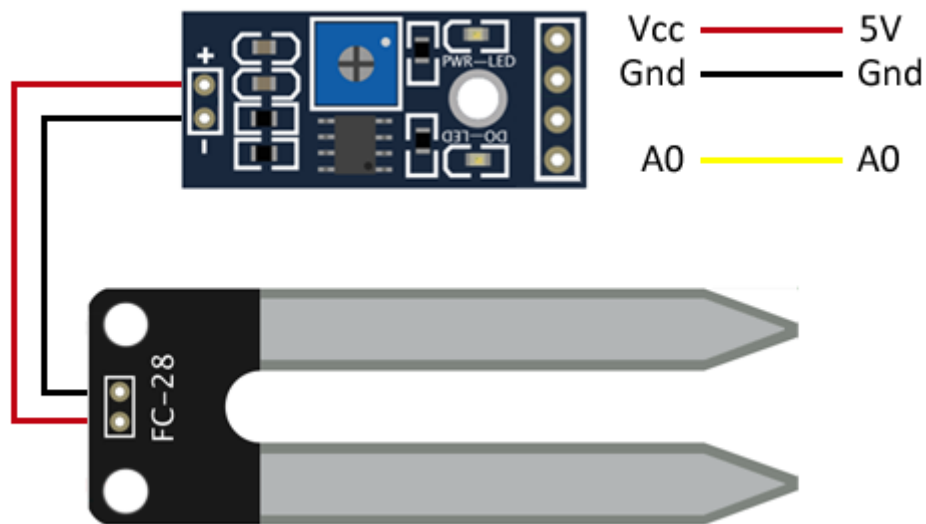


Figura 4.9: Diagrama de Pines FC-28  
Elaborado por: Investigador

Cabe recalcar que el pin D0 que esta entre los pines GND y A0, es decir, la salida digital del sensor no va a ser utilizada dentro de la investigación por lo que no se visualiza en la Figura 4.9.

## **4.7 Procesamiento y Transmisión**

La etapa de procesamiento de las señales físicas obtenidas mediante los sensores se lleva a cabo en el microcontrolador ESP32 en los cuatro nodos sensores que se va a utilizar para la captación de señales, el cual es el encargado de recibir las señales enviadas por los sensores y convertirlos en valores indicadores de cada una de las variables establecidas. Por consiguiente se envían dichas señales en un formato Json, es decir, es un lenguaje de mediano nivel para no sobrecargar de información, a través del módulo integrado de comunicación WIFI, es necesario recalcar que para poder utilizar dichos módulos se debe realizar un proceso de nombramiento a través del iD de los microcontroladores para que puedan comunicarse sin ningún problema. Dicha parte del procesamiento y la conversión de señales eléctricas a valores indicadores de las variables anteriormente mencionadas se describen en el diagrama de flujo de la Figura 4.10, además a través de los anexos correspondientes se presentan los algoritmos acordes al diagrama de la parte de transmisión de datos.

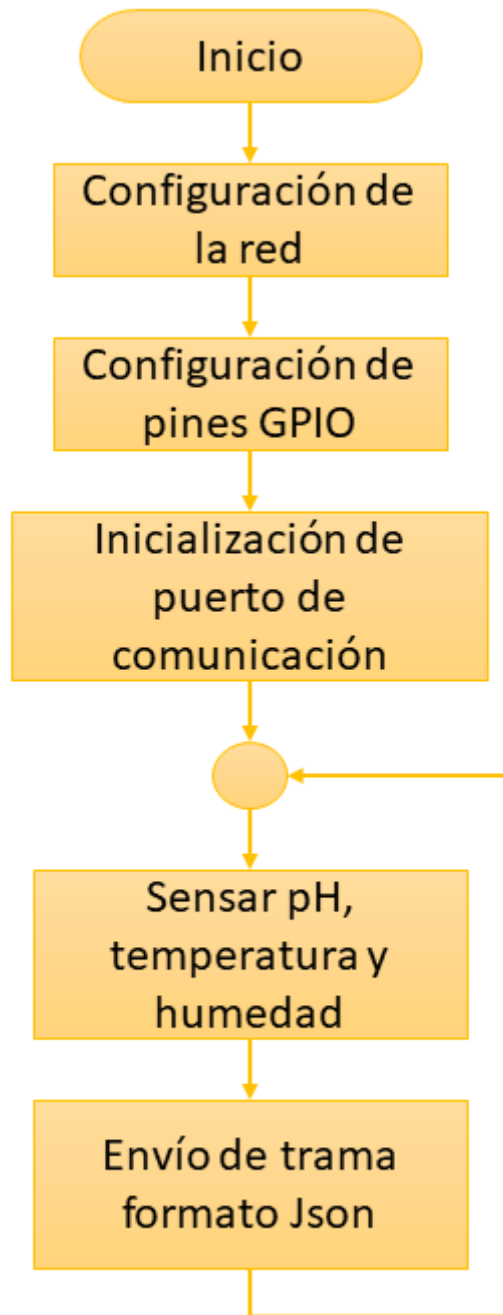


Figura 4.10: Diagrama de Flujo Transmisión de Datos  
Elaborado por: Investigador

Los datos de envío representados en la Figura 4.10 tanto de pH, temperatura y humedad son enviados mediante el formato Json, el cual es un formato de texto ligero de intercambio de datos, que resulta sencillo de leer y escribir para los programadores y de la misma manera simple a la hora de interpretar y generar para los dispositivos, cuenta con una estructura ligera y compacta.

Cabe recalcar que debido a la importancia que tiene el sistema no se lo ha provisto de retardos con el fin

de optimizar y disminuir los tiempos de envío. Finalmente una vez que los datos son enviados a través de los módulos de comunicación se descompacta la trama utilizando un método de deserialización simple de utilizar debido a que se está utilizando el formato Json.

## **4.8 Presentación de Señales**

Para la presentación de los datos se utiliza las ventajas del microcontrolador seleccionado previamente, el ESP32 mediante el programa Arduino IDE presenta la posibilidad de crear un servidor Web. Debido a que el prototipo que se va a diseñar es de bajo costo tanto a hardware, el software tendría el mismo objetivo por lo que se utilizaría el software libre y gratuito de Arduino IDE para crear el servidor dentro del microcontrolador, en donde se visualizará los datos recibidos mediante una página Web con una interfaz gráfica de usuario sencilla y simplificada con el único objetivo de verificar los resultados.

El servidor Web a través de un requerimiento POST acogerá el envío de datos de los sensores conectados al microcontrolador, permitiendo visualizar en la página los resultados de una manera sencilla.

## **4.9 Diseño Electrónico del Prototipo**

Del mismo modo todo software necesita al hardware para poder funcionar, los diagramas del circuito esquemático de los dispositivos utilizados para el sensado de las tres variables: pH, temperatura y humedad del suelo que se recoge de los terrenos ubicados en la parroquia de Izamba se los detalla a continuación.

### **4.9.1 Circuito Esquemático de los Nodos Sensores**

Después de la preselección de los dispositivos que se van a utilizar en la parte de los nodos, es decir, los sensores, se realiza un diagrama llamado circuito esquemático, con el fin de saber la interconexión de los dispositivos a interactuar.

En el diagrama esquemático se visualiza la configuración de pines de cada dispositivo, la función que cumple y dependiendo a ello la conexión que se necesita realizar, según el datasheet de cada uno. Los dispositivos utilizados en los nodos sensores se componen del microcontrolador, el sensor de pH y temperatura (PH-4502) y el sensor de humedad (FC-28) como se ilustra en la Figura 4.11.

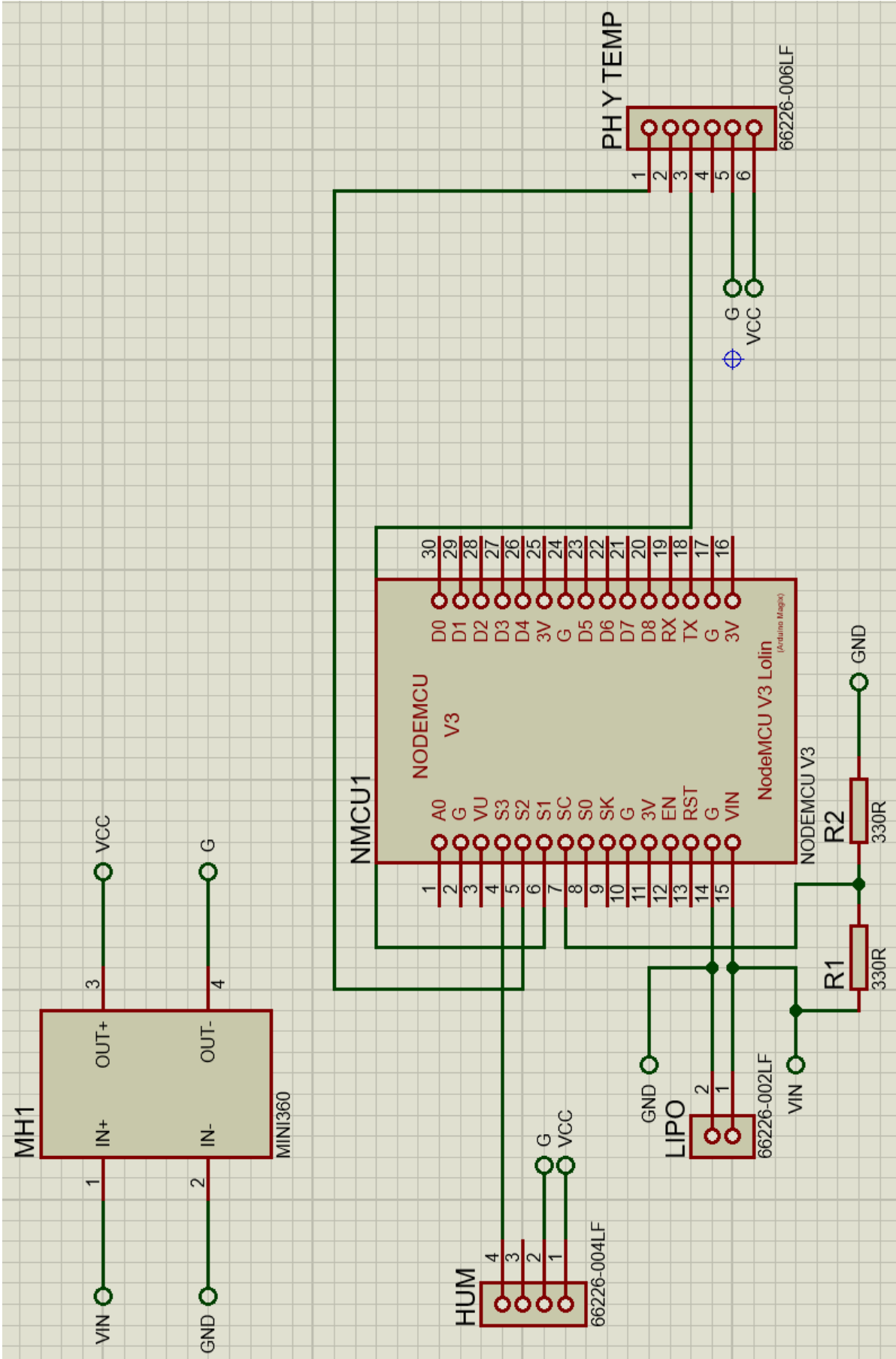


Figura 4.11: Circuito Esquemático Nodo Sensor  
Elaborado por: Investigador

Gracias al diseño del circuito donde interactúan todos los dispositivos tanto la parte de adquisición de información por los sensores, como la parte de procesamiento y transmisión de datos gracias al microcontrolador se toma en cuenta los datasheet de cada uno de ellos para saber voltajes de alimentación, y rango de operación en los que se pueda operar.

Dando como resultado, el diseño final del prototipo de la Figura 4.12, donde se observa todos los dispositivos conectados mediante una placa electrónica diseñada en base al circuito esquemático anterior.



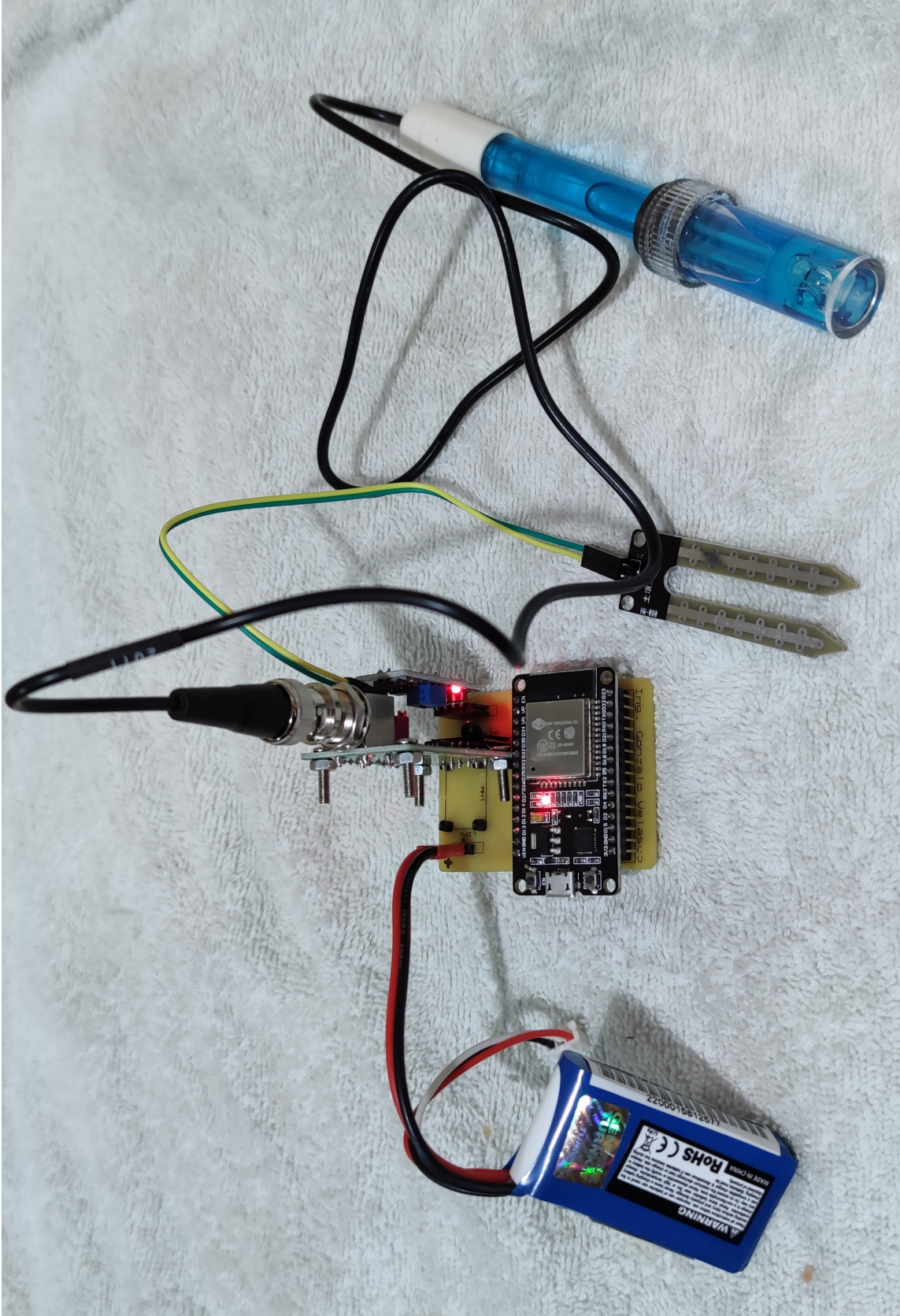


Figura 4.12: Nodo Sensor  
Elaborado por: Investigador

De acuerdo a las necesidades del prototipo no se requiere ninguna modificación adicional, por cuanto se utilizará para medir las variables correspondientes, el prototipo considera dispositivos invasivos para la recolección de datos de tal manera que van a estar en contacto físicamente con el terreno lo que garantiza el sensado de las variables. Debido a que se va a utilizar una topología en malla los nodos sensores serán iguales para cada nodo sensor que se requiera adicionar, en este caso se realizó cuatro nodos sensores.

#### 4.10 Topología física y lógica del sistema

Tomando en cuenta los parámetros para definir la efectividad de una topología y el estudio del arte visto previamente se eligió una topología de tipo malla, debido a la alta confiabilidad de la red y la escalabilidad que tiene para el presente proyecto como se observa en la Figura 4.13.

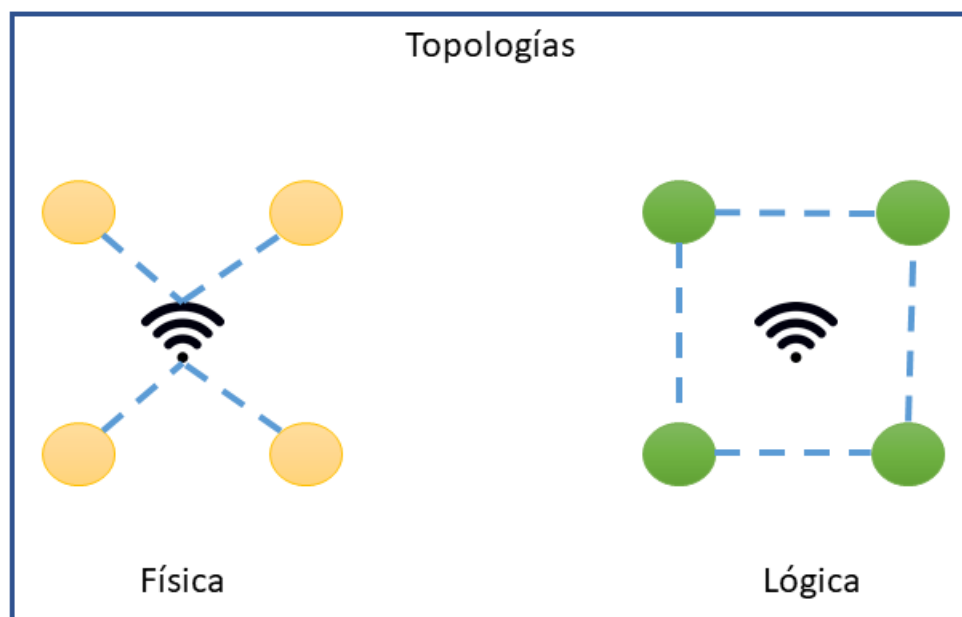


Figura 4.13: Topologías del sistema  
Elaborado por: Investigador

#### 4.11 Pruebas de Funcionamiento

Una vez realizado el prototipo, se inicia con las pruebas de funcionamiento en el sector seleccionado, en este caso en la parroquia Izamba. Para realizar las respectivas pruebas de funcionamiento y comprobar que el prototipo funcione adecuadamente se recolectó durante una semana en varios puntos de la parroquia, debido a que la zona delimitada para la recolección de información se realiza en sectores rurales y tomando en cuenta que los dispositivos necesitan de una fuente de energía, se añadió al prototipo baterías de LiPo muy utilizadas y con la finalidad de proveer de una fuente de energía, de fácil manejo y relativamente pequeña para transporte de un lugar a otro para realizar las pruebas sin necesidad de estar conectadas al computador. En la Figura 4.14 se visualiza las baterías que se utilizó como fuente de alimentación del prototipo.



Figura 4.14: Baterías de LiPo  
Elaborado por: Investigador

Se seleccionó lugares aleatoriamente de preferencia donde no haya cultivos sembrados previamente, para la implementación del sistema dentro de la parroquia Izamba como se muestra en la Figura 4.15.



Figura 4.15: Instalación prototipo sectores agrícolas  
Elaborado por: Investigador

Sin duda las pruebas realizadas en los sectores agrícolas fueron exitosas como se muestra en la Figura 4.16, donde se verifica que existe comunicación entre los nodos sensores mediante la red de tipo malla.

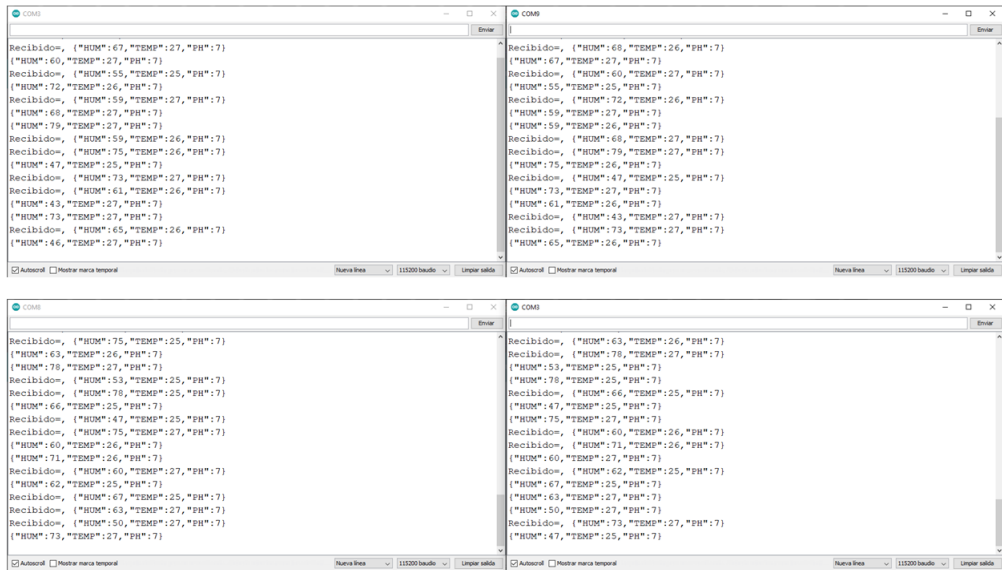


Figura 4.16: Prueba de Conexión de los Nodos Sensores  
Elaborado por: Investigador

Posterior a ello, se tenía que enlazar la información obtenida hacia el servidor Web montado en el mismo ESP32, de ese modo poder analizar la información y generar un resultado final, mediante un algoritmo desarrollado se realizó la comparación de los datos con la Tabla 4.4, lo que da como resultado el tipo de cultivo que mejor se adaptará a las condiciones edafoclimáticas medidas por el prototipo diseñado, generando así una mayor productividad de los cultivos, sabiendo mediante el análisis total, que es el mejor producto para el terreno seleccionado como se muestra en la Figura 4.17.



Figura 4.17: Resultados de Servidor Web  
Elaborado por: Investigador

Un punto importante es el almacenamiento de información de los datos, como se vio en la investigación las variables tanto de temperatura como de pH, no tienden a cambiar de manera abrupta por lo que únicamente se almacenada dentro del microcontrolador el último dato de los nodos sensores, es decir, con la última toma de muestras se determina mediante el algoritmo el cultivo recomendado.

Cabe recalcar que el prototipo esta diseñado para trabajar como un instrumento de medición con el fin de tomar una decisión con los últimos datos obtenidos por lo que se indico anteriormente.

En base a los resultados obtenidos del prototipo en diferentes sectores de la parroquia de Izamba se visualiza los resultados mediante la página de la Figura 4.17, donde se colocó tres resultados diferentes para su respectivo detalle, como se observa en uno de los resultados el terreno donde se realizó las mediciones no cumplía con las condiciones necesarias para la siembra de algún cultivo evaluado dentro de la investigación, mientras que en otros lugares el terreno estaba adecuado para los cultivos de cebolla paitaña y en el otro caso maíz.

Finalmente, se adquirió mediante la aplicación de GPS Essentials instalada en un celular los puntos de referencia donde estaban ubicados los nodos sensores para una mejor visualización del terreno donde se debería sembrar el cultivo recomendado mediante el prototipo diseñado como se mira en la Figura

4.18. Es así que mediante la aplicación y de manera muy sencilla se logra determinar un sistema de información geográfica basados en la ubicación de los nodos sensores.

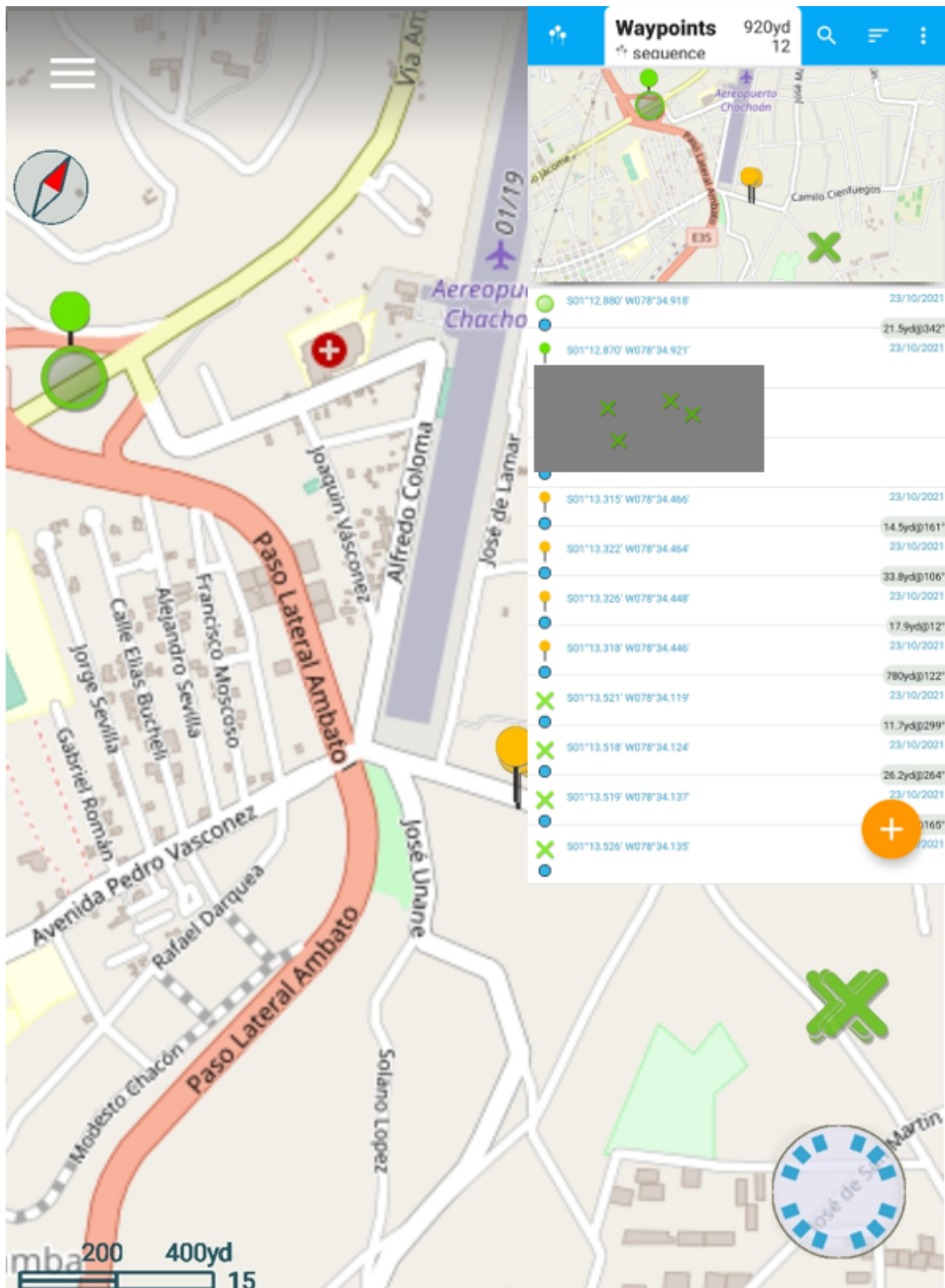


Figura 4.18: Puntos de Referencia GPS Essentials  
Elaborado por: Investigador

El sistema de información geográfica muestra los puntos donde se localiza cada nodo sensor mediante datos como latitud y longitud y para una mejor visualización dentro de la misma aplicación se puede

exportar los datos a Google Maps y elegir las vistas que sirva para poder determinar el sector donde fueron ubicados los nodos sensores como se ve en la Figura 4.19.



Figura 4.19: Ubicación de nodos sensores en Google Maps  
Elaborado por: Investigador



Un punto bastante importante dentro de la investigación fue obtener resultados precisos, tanto así que en algunos lugares de la parroquia de Izamba coincidentalmente los resultados que se mostraba luego del análisis con el prototipo daban como resultado el mismo cultivo que ya se había sembrado previamente por el dueño del terreno, esto generó mayor niveles de confianza.

Con base a lo expuesto anteriormente, se ha demostrado que el prototipo cumple con el objetivo deseado, es decir, contribuye en gran medida a sacar el mejor rendimiento del terreno y aumentando la compatibilidad con el cultivo, brindando excelentes resultados y garantizando el desarrollo de la planta en su etapa inicial o etapa cero.

#### 4.12 Análisis presupuestario

En concordancia con el análisis técnico expuesto en los anteriores apartados, en esta sección se presenta un análisis económico con el afán de otorgar una perspectiva realista a tomar en cuenta al momento de integrar la tecnología a la agricultura y por ende al desarrollo de la agricultura de precisión como se denomina a esta práctica. En la Tabla 4.8 se presenta el costo referencial de cada nodo sensor que componga la red WSN con N nodos separados geográficamente.

Tabla 4.8: Tabla de presupuesto económico de nodo sensor

<b>Materiales</b>	<b>Cantidad</b>	<b>Costo Unitario (\$)</b>	<b>Costo Total (\$)</b>
Microcontrolador ESP32	1	10.00	10.00
Sensor PH-4502	1	46.00	46.00
Sensor FC-28	1	2.50	2.50
Baterías de LiPo 500 mA	1	16.00	16.00
Materiales y Cableado	1	30.00	30.00
		<b>TOTAL</b>	<b>104.5</b>

Elaborado por: Investigador

El costo total de ciento cuatro dólares americanos con cincuenta centavos no representa un costo excesivo si se considera los beneficios del sistema, de pasar de una agricultura tradicional a una agricultura de precisión, adicionalmente se optimiza los recursos como el cultivo a plantar generando mayor rendimiento del mismo, y sacando el mayor provecho del terreno de cada agricultor, por lo que incrementará favorablemente la producción y con ello las ventas con un mejor producto.

## 5 CAPÍTULO V

### CONCLUSIONES Y RECOMENDACIONES

#### 5.1 Conclusiones

- Durante la implementación de las redes de sensores inalámbricos, en base a los datos de comunicación obtenidos se concluyó que la distancia adecuada de operación entre los nodos sensores es de 20 m debido que a mayor distancia la transmisión de datos entre los nodos se perdía casi en 20% por cada metro de distancia adicional. Garantizando la comunicación entre todos los sensores de la red en una topología tipo malla con cuatro sensores ubicados en diferentes puntos de referencia de acuerdo a la topología utilizada en esta investigación.
- En base a los datos obtenidos de los sensores y tomando como evidencia los puntos donde se realizó las mediciones, se llegó a la conclusión que tanto el pH y la temperatura en diferentes zonas de la parroquia no cambian, pero si tomamos en cuenta la variable de humedad del suelo puede ser crucial para la toma de decisiones de que cultivo sembrar, ya que muchos de los sectores tenían un sistema de riego manual por lo que afecta directamente a la humedad en el terreno.
- Gracias a las ventajas del microcontrolador, uno de los puntos fundamentales fue el dispositivo de comunicación, aparte de tener un costo relativamente accesible para cualquier persona ofrece una relación calidad precio demasiado alta para el sistema de redes de sensores inalámbricos, ayudando de esta manera a la optimización de recursos y obteniendo una forma de presentar los datos obtenidos sin ningún dispositivo adicional.
- Para el uso del sensor de pH y temperatura (PH-4502) se debe realizar una calibración previa ya sea por el método a nivel de hardware o de software, con una sustancia de pH cercana a 7, esto hará que el sensor actúe de manera eficaz en las lecturas, es decir, aumentando la sensibilidad del sensor una vez expuesto a otra sustancia de pH relativamente igual. En todos los sectores agrícolas donde se utilizó el sensor el pH tomaba valores relativamente cercanos a 7, por lo que se concluye que el terreno era adecuado para los cultivos estudiados.

## 5.2 Recomendaciones

- Para terrenos agrícolas de áreas superiores a los 400 metros cuadrados se recomienda utilizar nodos sensores adicionales para cubrir todo el terreno a ser estudiado y sacar de mejor manera las mediciones, garantizando un estudio de manera general y no solo en una parcela, o a su vez realizar las mediciones cada 400 metros cuadrados utilizando los mismos nodos sensores.
- Es necesario tomar en cuenta para el desarrollo del sistema propuesto en el estudio, realizar la actualización del Arduino IDE a su última versión, instalar los complementos necesarios para no tener conflictos en la programación, de igual manera al momento de cargar el programa al microcontrolador se debe apretar el botón de BOOT para que no haya conflictos con el programa previamente cargado.
- Después del estudio planteado se requiere algunas modificaciones para realizar mejoras sustanciales utilizando actuadores de por medio para diferentes técnicas de la agricultura de precisión como puede ser la optimización de recursos como el agua, añadiendo electro válvulas o realizando un estudio mas detallado con el sensor de pH, para optimizar recursos como fertilizantes y pesticidas que se usan a diario dentro de la agricultura.
- En el caso que se requiera utilizar mas sensores dentro del sistema se debe realizar un estudio de la vida útil de la batería debido a que los dispositivos electrónicos mientras mayor sea el número de sensores y de comunicaciones entre ellos, mayor será el consumo energético.

## BIBLIOGRAFÍA

- [1] Abu Zafar Abbasi, Noman Islam, Zubair Ahmed Shaikh, et al. A review of wireless sensors and networks' applications in agriculture. *Computer Standards & Interfaces*, 36(2):263–270, 2014.
- [2] Mohammad Hossein Anisi, Gaddafi Abdul-Salaam, and Abdul Hanan Abdullah. A survey of wireless sensor network approaches and their energy consumption for monitoring farm fields in precision agriculture. *Precision Agriculture*, 16(2):216–238, 2015.
- [3] Open Source Hardware Association. Declaracion de principios 1.0. 2021.
- [4] Red de Especialistas en Agricultura. Agricultura latinoamericana, gran potencial y grandes desafios. 2017.
- [5] Instituto Nacional de Estadística y Censos. Productores agrícolas. October 2021.
- [6] Ecuador. Ministerio de agricultura y ganaderia (mag).
- [7] Sherine M Abd El-Kader and Basma M Mohammad El-Basioni. Precision farming solution in egypt using the wireless sensor network technology. *Egyptian Informatics Journal*, 14(3):221–233, 2013.
- [8] Haider Mahmood Jawad, Rosdiadee Nordin, Sadik Kamel Gharghan, Aqeel Mahmood Jawad, and Mahamod Ismail. Energy-efficient wireless sensor networks for precision agriculture: A review. *Sensors*, 17(8):1781, 2017.
- [9] Yassine Jiber, Hamid Harroud, and Ahmed Karmouch. Precision agriculture monitoring framework based on wsn. In *2011 7th International Wireless Communications and Mobile Computing Conference*, pages 2015–2020. IEEE, 2011.
- [10] Subramania Ananda Kumar and Paramasivam Ilango. The impact of wireless sensor network in the field of precision agriculture: A review. *Wireless Personal Communications*, 98(1):685–698, 2018.
- [11] Parveen Kumari and Amit Mahal. A study on modern farming using wireless sensor networks. *International Journal of Advances in Scientific Research and Engineering*, December 2017.
- [12] LearnOsm. Gps essentials, 2015.
- [13] Neptali Armando Echeverria Llumipanta. Desarrollo de un sistema de informacion geografica que permita la gestion integral de los cultivos de palma aceitera a traves de la aplicacion de sistemas de agricultura de precision. Master's thesis, Universidad San Francisco de Quito, June 2015.

- [14] Cristian Mattar Luis Morales, Fabricio Canessa. Characterization and edafic and climatic zonification in theregion of coquimbo. 2006.
- [15] Douglas Marin. La agricultura en latinoamerica y el caribe y las claves para su futuro. 2019.
- [16] Sergio Monteleone, Edmilson Alves de Moraes, and Rodrigo Filev Maia. Analysis of the variables that affect the intention to adopt precision agriculture for smart water management in agriculture 4.0 context. In *2019 Global IoT Summit (GIoTS)*, pages 1–6. IEEE, 2019.
- [17] 330 ohms. ¿que son los actuadores electronicos? October 2013.
- [18] Edgar Alberto Maya Olalla and Gustavo Xavier Chafra Altamirano. Red inalámbrica de sensores a través de 6lowpan para una agricultura de precisión. *SATHIRI*, (6):22–32, 2018.
- [19] GH Raghunandan, SY Namratha, SY Nanditha, and G Swathi. Comparative analysis of different precision agriculture techniques using wireless sensor networks. In *2017 4th International Conference on Electronics and Communication Systems (ICECS)*, pages 129–133. IEEE, 2017.
- [20] Antonio Ruiz. *Guia practica de sensores*. Creaciones Copyright, S.l, 2010.
- [21] Mare Srbinovska, Cvetan Gavrovski, Vladimir Dimcev, Aleksandra Krkoleva, and Vesna Borozan. Environmental parameters monitoring in precision agriculture using wireless sensor networks. *Journal of cleaner production*, 88:297–307, 2015.
- [22] Telegrafo. En ecuador, 12,3 millones de hectareas estan a cargo de los productores agropecuarios. Diciembre 2020.
- [23] Wayne Tomasi. *Sistemas de comunicaciones electronicas*. Pearson Educacion, Mexico, 2003.

## ANEXOS

### Anexo 1. Código de Programación para crear una red tipo malla entre los nodos sensores.

```
#include "painlessMesh.h"

#define MESH_PREFIX "NombredelaredWIFI"

#define MESH_PASSWORD "ContrasenadelaredWIFI"

#define MESH_PORT 5555

Scheduler userScheduler; // to control your personal task

painlessMesh mesh;

void sendMessage() ;

Task taskSendMessage( TASK_SECOND * 1 , TASK_FOREVER, &sendMessage );

void sendMessage() {

String msg = "Hello from node ";

msg += mesh.getNodeId();

mesh.sendBroadcast( msg );

taskSendMessage.setInterval( random( TASK_SECOND * 1, TASK_SECOND * 5 ));

}

// Needed for painless library

void receivedCallback( uint32_t from, String &msg ) {

Serial.printf("startHere: Received from %u msg=%s\n", from, msg.c_str());

}

void newConnectionCallback(uint32_t nodeId) {

Serial.printf("--> startHere: New Connection, nodeId = %u\n", nodeId);

}

void changedConnectionCallback() {

Serial.printf("Changed connections\n");

}

void nodeTimeAdjustedCallback(int32_t offset) {

Serial.printf("Adjusted time %u. Offset = %d\n", mesh.getNodeTime(),offset);

}

void setup() {

Serial.begin(115200);

//mesh.setDebugMsgTypes( ERROR | MESH_STATUS | CONNECTION | SYNC | COMMUNICA-
TION | GENERAL | MSG_TYPES | REMOTE ); // all types on

mesh.setDebugMsgTypes( ERROR | STARTUP ); // set before init() so that you can see startup messages

mesh.init( MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT );
```

```
mesh.onReceive(&receivedCallback);
mesh.onNewConnection(&newConnectionCallback);
mesh.onChangedConnections(&changedConnectionCallback);
mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);
userScheduler.addTask( taskSendMessage );
taskSendMessage.enable();
}
void loop() {
// it will run the user scheduler as well
mesh.update();
}
```

## **Anexo 2. Código de Programación para obtener los datos de las variables como humedad del suelo, temperatura y pH.**

```
const int hum_pin = 34; // se define el pin de ESP32 GPIO del pin de humedad
const int temp_pin = 35; // se define el pin de ESP32 GPIO del pin de temperatura

int h = 0;

float t = 0.0;

float ph = 0;

//*****

float calibration = 25.00;//change this value to calibrate

const int analogInPin = 32; //pin GPIO del sensor de pH

int sensorValue = 0;

unsigned long int avgValue;

float b;

int buf[10],temp;

//*****

void setup() {
  Serial.begin(115200);

  //define el modo de trabajo del pin (Entrada o Salida)

  pinMode (hum_pin, INPUT);
  pinMode (temp_pin, INPUT);
}

void loop() {
  //convertir datos del sensor a porcentaje

  h = map(analogRead(hum_pin),0,4095,100,0);

  t = analogRead (temp_pin);

  phsensor();

  Serial.println(h);

  Serial.println(t);

  delay(1000);
}

// debido a que el sensor necesita calibracion se realiza un metodo aparte

void phsensor() {
  for(int i=0;i<10;i++)

  {

    buf[i]=analogRead(analogInPin);
```



```
delay(30);
}
for(int i=0;i<9;i++)
{
for(int j=i+1;j<10;j++) {
if(buf[i]>buf[j])
{
temp=buf[i];
buf[i]=buf[j];
buf[j]=temp;
}
}
}
avgValue=0;
for(int i=2;i<8;i++)
avgValue+=buf[i];
float pHVol=(float)avgValue*3.6/4095/6;
ph = -5.70 * pHVol + calibration;
}
```

**Anexo 3. Código de programación para enviar y recibir el valor de las variables a través de la red tipo malla en formato Json en los nodos sensores.**

```
#include "painlessMesh.h"
#include <ArduinoJson.h>
#define MESH_PREFIX "Loquesea"
#define MESH_PASSWORD "nomelase"
#define MESH_PORT 5555
//*****
float calibration = 25.00;//change this value to calibrate
const int analogInPin = 32; //pin GPIO del sensor de pH
int sensorValue = 0;
unsigned long int avgValue;
float b;
int buf[10],temp;
//*****
Scheduler plan; // to control your personal task
painlessMesh mesh;
//variables
const int hum_pin = 34; // se define el pin de ESP32 GPIO del pin de humedad
const int temp_pin = 35; // se define el pin de ESP32 GPIO del pin de temperatura
int h = 0;
float t = 0.0;
float ph = 0;
void sendMessage() ;
Task taskSendMessage( TASK_SECOND * 1 , TASK_FOREVER, &sendMessage );
void sendMessage() {
h = map(analogRead(hum_pin),0,4095,100,0);
t = analogRead (temp_pin);
phsensor();
String msg;
//*****
//Guardar en Json
//Proceso para el envío de las variables mediante el formato Json
StaticJsonDocument <512> doc;
doc["HUM"] = h;
```

```

doc["TEMP"] = t;
doc["PH"] = ph;
serializeJson(doc, msg);
Serial.println(msg);
//*****

mesh.sendBroadcast( msg );
taskSendMessage.setInterval( random( TASK_SECOND * 1, TASK_SECOND * 5 ));
}

void receivedCallback( uint32_t from, String &msg ) {
Serial.printf("Recibido, %s\n", msg.c_str());
//*****

//Proceso para la recepción de datos a través del formato Json
String json;
StaticJsonDocument <512> doc;
json = msg.c_str();
DeserializationError error = deserializeJson (doc, json);
if (error) {
Serial.print(F("Error deserializado: "));
Serial.println(error.c_str());
}
hum = doc["HUM"];
float temp = doc["TEMP"];
float pah = doc["PH"];
//*****

}

void newConnectionCallback(uint32_t nodeId) {
}

void changedConnectionCallback() {
}

void nodeTimeAdjustedCallback(int32_t offset) {
}

void setup() {
Serial.begin(115200);
pinMode (hum_pin, INPUT);
pinMode (temp_pin, INPUT);

```

```

mesh.setDebugMsgTypes( ERROR | STARTUP ); // set before init() so that you can see startup messages
mesh.init( MESH_PREFIX, MESH_PASSWORD, &userScheduler, MESH_PORT );
mesh.onReceive(&receivedCallback);
mesh.onNewConnection(&newConnectionCallback);
mesh.onChangedConnections(&changedConnectionCallback);
mesh.onNodeTimeAdjusted(&nodeTimeAdjustedCallback);
plan.addTask( taskSendMessage );
taskSendMessage.enable();
}
void loop() {
// it will run the user scheduler as well
mesh.update();
}
void phsensor() {
for(int i=0;i<10;i++)
{
buf[i]=analogRead(analogInPin);
delay(30);
}
for(int i=0;i<9;i++)
{
for(int j=i+1;j<10;j++) {
if(buf[i]>buf[j])
{
temp=buf[i];
buf[i]=buf[j];
buf[j]=temp;
}
}
}
avgValue=0;
for(int i=2;i<8;i++)
avgValue+=buf[i];
float pHVol=(float)avgValue*3.6/4095/6;
ph = -5.70 * pHVol + calibration;

```



#### Anexo 4. Código para la creación del servidor Web y configuración de la página para presentación de datos.

```
#include <WiFi.h>

//-----Servidor Web en puerto 80-----

WiFiServer server(80);

//-----Credenciales de WiFi-----

const char* ssid = "RedWIFIdelacasa";

const char* password = "Contrasena";

//-----VARIABLES GLOBALES-----

int contconexion = 0;

String header; // Variable para guardar el HTTP request

String estadoSalida = "off";

const int salida = 2;

//-----CODIGO HTML-----

String pagina1 = "<!DOCTYPE html>"

"<html>"

"<head>"

"<meta charset='utf-8' />"

"<title>SENSORES</title>"

"</head>"

"<body>"

"<center>"

"<h1>SISTEMA DE INFORMACION GEOGRAFICA PARA LA GESTION DE LOS CULTIVOS DE LA PARROQUIA IZAMBA</h1>"

"<p><H2>HUMEDAD DEL SUELO: </H2></p>";

String humedad_str = "10.0";

String pagina2 = "<p><H2>PH DEL SUELO: </H2></p>";

String ph_str = "20.0";

String pagina3 = "<p><H2>TEMPERATURA DEL ENTORNO: </H2></p>";

String temp_str = "30.0";

String pagina4 = "<BR><BR>"

"<b><h2>CULTIVO RECOMENDADO: <h2><b>";

String cultivo = "cultivo";

String cola = "</center>"

"</body>"
```

```

"</html>";

//-----SETUP-----

void setup() {
  Serial.begin(115200);
  Serial.println("");
  pinMode(salida, OUTPUT);
  digitalWrite(salida, LOW);
  // Conexión WIFI
  WiFi.begin(ssid, password);
  //Cuenta hasta 50 si no se puede conectar lo cancela
  while (WiFi.status() != WL_CONNECTED and contconexion <50) {
    ++contconexion;
    delay(500);
    Serial.print(".");
  }
  if (contconexion <50) {
    //para usar con ip fija
    //IPAddress ip(192,168,1,180);
    //IPAddress gateway(192,168,1,1);
    //IPAddress subnet(255,255,255,0);
    //WiFi.config(ip, gateway, subnet);
    Serial.println("");
    Serial.println("WiFi conectado");
    Serial.println(WiFi.localIP());
    server.begin(); // iniciamos el servidor
  }
  else {
    Serial.println("");
    Serial.println("Error de conexion");
  }
}

//-----LOOP-----

void loop(){
  WiFiClient client = server.available(); // Escucha a los clientes entrantes

```

```

String pagina = pagina1 + humedad_str + pagina2 + ph_str + pagina3 + temp_str + pagina4 + cultivo +
cola;
if (client) { // Si se conecta un nuevo cliente }
Serial.println("New Client."); //
String currentLine = ""; //
while (client.connected()) { // loop mientras el cliente está conectado
if (client.available()) { // si hay bytes para leer desde el cliente
char c = client.read(); // lee un byte
Serial.write(c); // imprime ese byte en el monitor serial
header += c;
if (c == '\n') { // si el byte es un caracter de salto de linea
// si la nueva linea está en blanco significa que es el fin del
// HTTP request del cliente, entonces respondemos:
if (currentLine.length() == 0) {
client.println("HTTP/1.1 200 OK");
client.println("Content-type:text/html");
client.println("Connection: close");
client.println();
// Muestra la página web client.println(pagina);
// la respuesta HTTP termina con una linea en blanco
client.println();
break;
} else { // si tenemos una nueva linea limpiamos currentLine
currentLine = "";
}
} else if (c != '\r') { // si C es distinto al caracter de retorno de carro
currentLine += c; // lo agrega al final de currentLine
}
}
}
// Limpiamos la variable header
header = "";
// Cerramos la conexión
client.stop();
Serial.println("Client disconnected.");

```



```
Serial.println("");
```

```
}
```

```
}
```

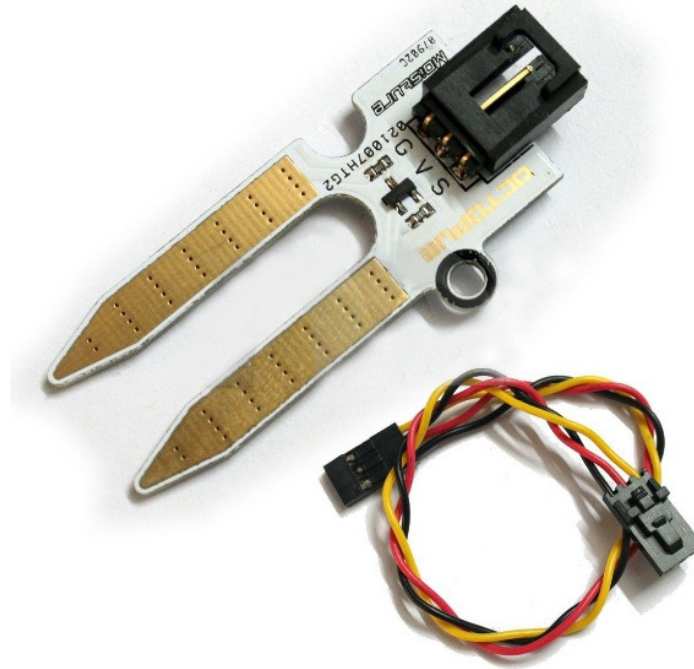
**Anexo 5. Método realizado con el algoritmo para determinar el tipo de cultivo recomendado para el terreno.**

```
void tipoCultivo()
{
if((h >= 45 && h <= 55)&&(t >=13 && t <=18 )&&(ph >=5.5 && ph <= 6.0 ))
{
cultivo = "Papa";
}
else if((h >= 65 && h <= 70)&&(t >=15 && t <=20 )&&(ph >=6.0 && ph <= 7.0 ))
{
cultivo = "Fresa";
}
else if((h >= 50 && h <= 60)&&(t >=10 && t <=24 )&&(ph >=7.0 && ph <= 8.0 ))
{
cultivo = "Trigo";
}
else if((h >= 35 && h <= 50)&&(t >=19 && t <=26 )&&(ph >=6.1 && ph <= 7.2 ))
{
cultivo = "Cebada";
}
else if((h >= 50 && h <= 70)&&(t >=15 && t <=25 )&&(ph >=5.8 && ph <= 6.8 ))
{
cultivo = "Zanahoria";
}
else if((h >= 60 && h <= 70)&&(t >=13 && t <=24 )&&(ph >=6.0 && ph <= 7.0 ))
{
cultivo = "Cebolla Paitaña";
}
else if((h >= 70 && h <= 95)&&(t >=18 && t <=21 )&&(ph >=5.6 && ph <= 6.5 ))
{
cultivo = "Café";
}
else if((h >= 40 && h <= 60)&&(t >=18 && t <=24 )&&(ph >=5.5 && ph <= 7.0 ))
{
cultivo = "Manzana";
}
```

```
}  
else if((h >= 45 && h <= 50)&&(t >=15 && t <=20 )&&(ph >=5.7 && ph <= 6.8 ))  
{  
cultivo = "Habas";  
}  
else if((h >= 30 && h <= 45)&&(t >=10 && t <=17 )&&(ph >=5.5 && ph <= 8.0 ))  
{  
cultivo = "Quinoa";  
}  
else if((h >= 30 && h <= 40)&&(t >=19 && t <=24 )&&(ph >=5.5 && ph <= 7.8 ))  
{  
cultivo = "Maíz";  
}  
else  
{  
cultivo = "El terreno no cumple con las condiciones para sembrar los cultivos preseleccionados";  
}  
}
```



## OBSoil-01 Octopus Soil Moisture Sensor Brick



Octopus Soil Moisture Sensor Brick can read the amount of moisture present in the soil surrounding it. Ideal for monitoring an urban garden, or your pet plant's water level. This is a must have tool for a connected garden!

This sensor uses the two probes to pass current through the soil, and then it reads that resistance to get the moisture level. More water makes the soil conduct electricity more easily (less resistance), while dry soil conducts electricity poorly (more resistance).

This sensor isn't hardened against contamination or exposure of the control circuitry to water and may be prone to electrolytic corrosion across the probes (Also it can be switched on, take the reading and switched off to minimize electrolytic corrosion), so it isn't well suited to being left in place or used outdoors.

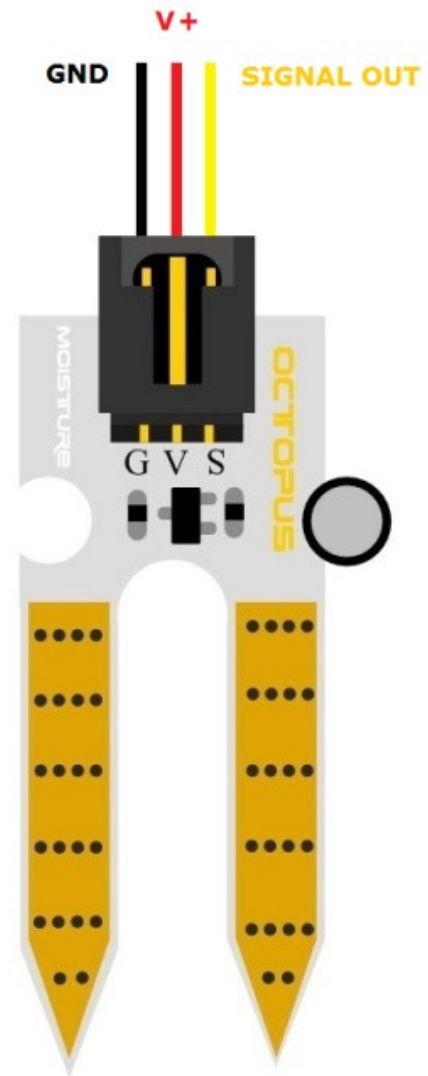
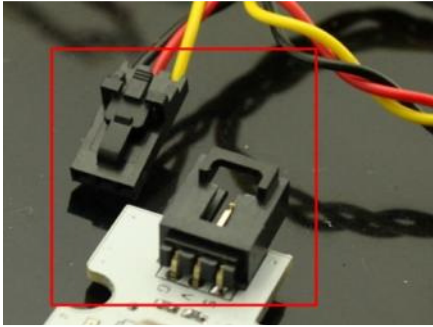
### Specification

Item	Condition	Min	Typical	Max	Unit
Voltage	-	3.3	/	5	V
Current	-	0	/	35	mA
Output Voltage	Supply Voltage 5 V	0	~	4.2	V
Output Value	Sensor in dry soil	0	~	300	/
	Sensor in humid soil	300	~	700	/
	Sensor in water	700	~	950	/

## Pinout:

- **G: GND (Black wire)**
- **V: Voltage Supply (+) (Red wire)**
- **S: Signal out (Yellow wire)**

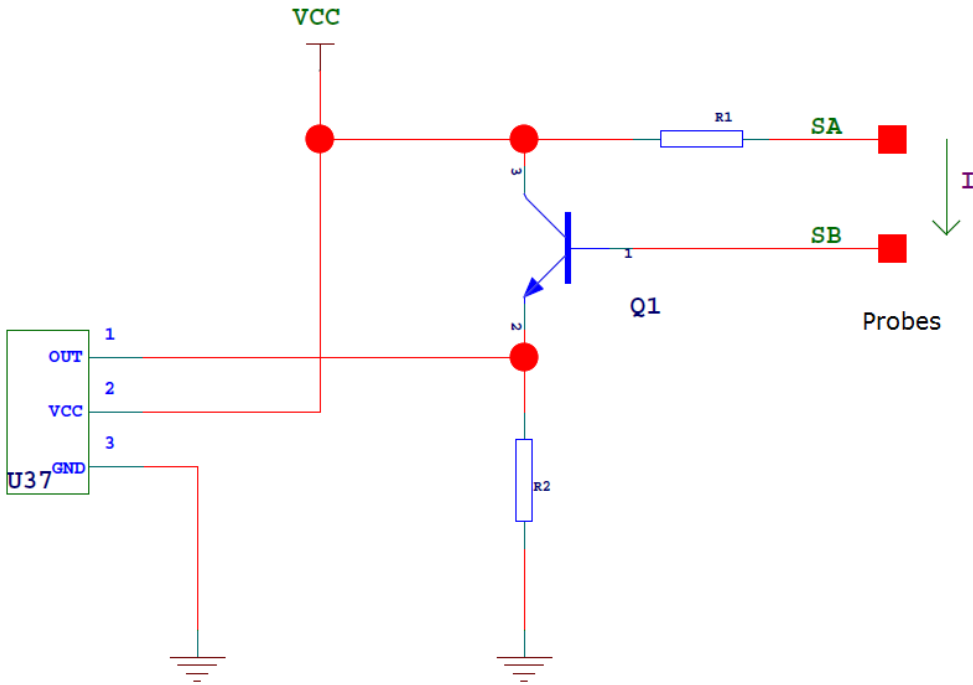
3P buckled wires connector and cable included:



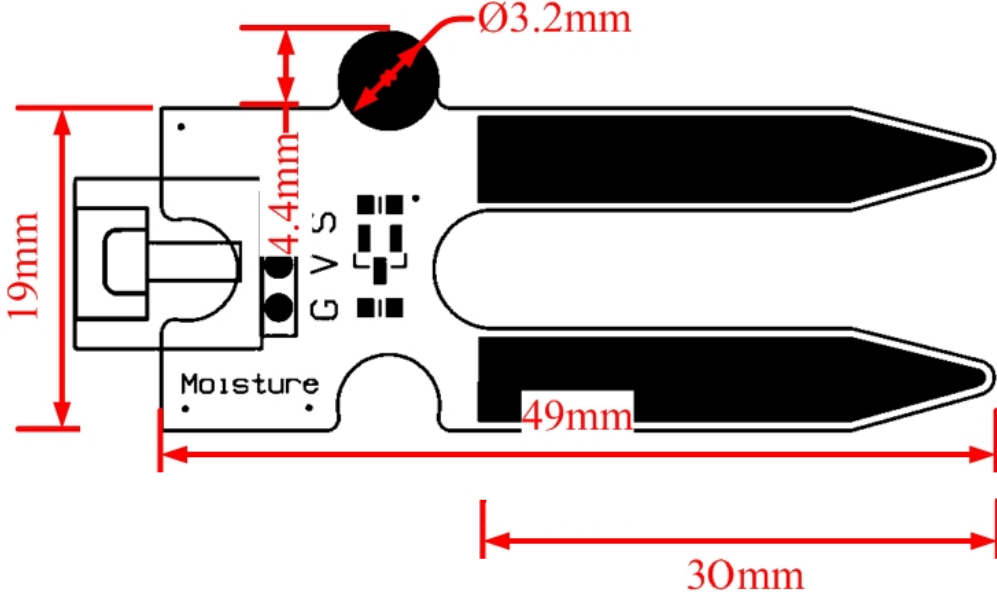
## Applications

- Moisture sensing
- Botanical gardening
- Flood detection
- Liquid level detection

# Internal Schematic

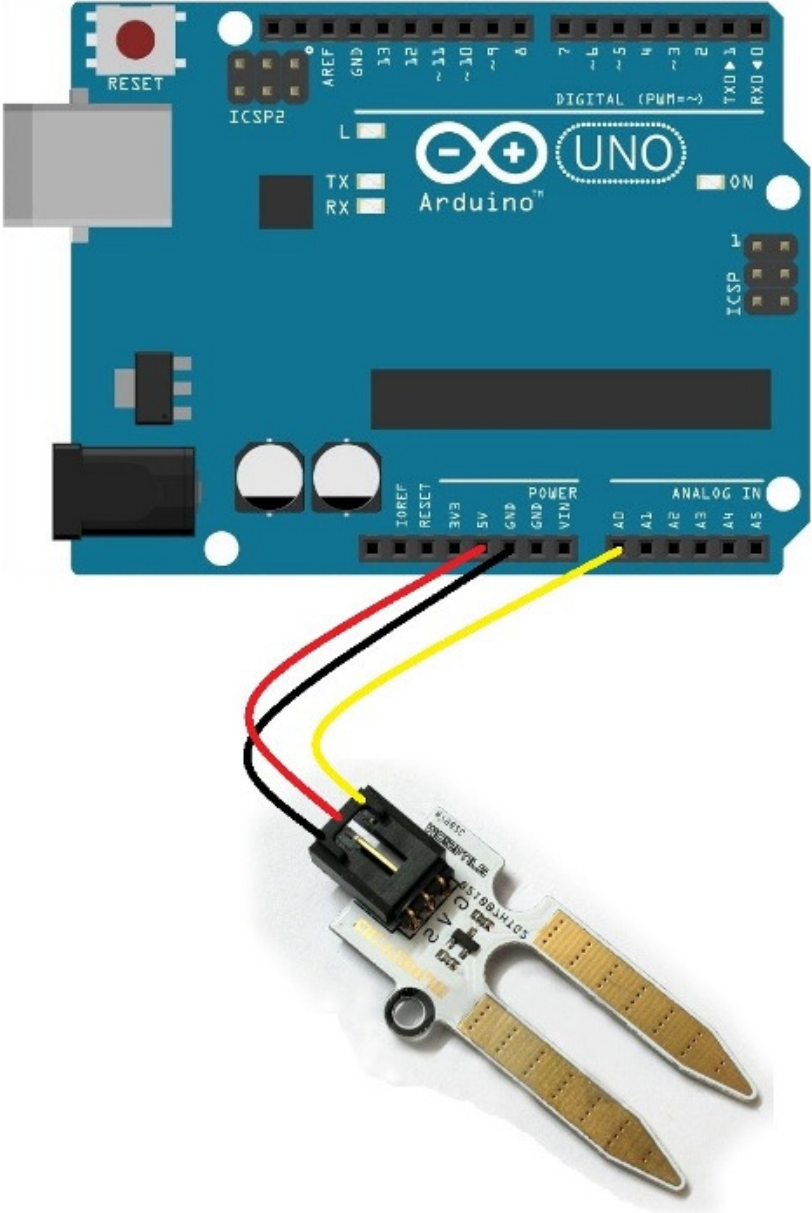


# Mechanic Dimensions



# Arduino Example

## Connection Diagram



## Sample Code

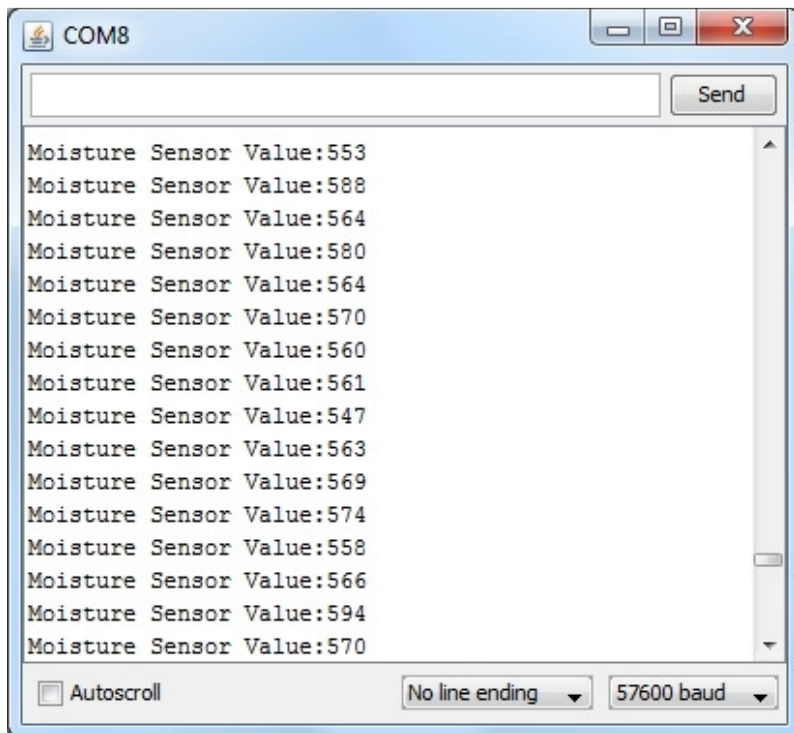
```
/*
 # Example code for the moisture sensor
 # Editor      : Lauren
 # Date       : 13.01.2012
 # Version    : 1.0
 # Connect the sensor to the A0 (Analog 0) pin on the Arduino board

 # the sensor value description
 # 0 ~300     dry soil
 # 300~700    humid soil
 # 700~950    in water
 */

void setup(){
  Serial.begin(57600);
}

void loop(){
  Serial.print("Moisture Sensor Value:");
  Serial.println(analogRead(0));
  delay(100);
}
```

The Result in different condition after open the “serial monitor” of Arduino IDE (Under tools menu):





# PH Sensor E-201-C



Technical Manual Rev 1r0



**The pH sensor Module** consist of PH sensor also called as PH probe and a Signal Conditioning Board which gives an output which is proportional to the PH Value and can be interfaced directly to any Micro-controller.

The pH sensor components are usually combined into one device called a combination pH electrode. The measuring electrode is usually glass and quite fragile. Recent developments have replaced the glass with more durable solid-state sensors. The preamplifier is a signal-conditioning device. It takes the high-impedance pH electrode signal and changes it into a low impedance signal which the analyzer or transmitter can accept. The preamplifier also strengthens and stabilizes the signal, making it less susceptible to electrical noise.

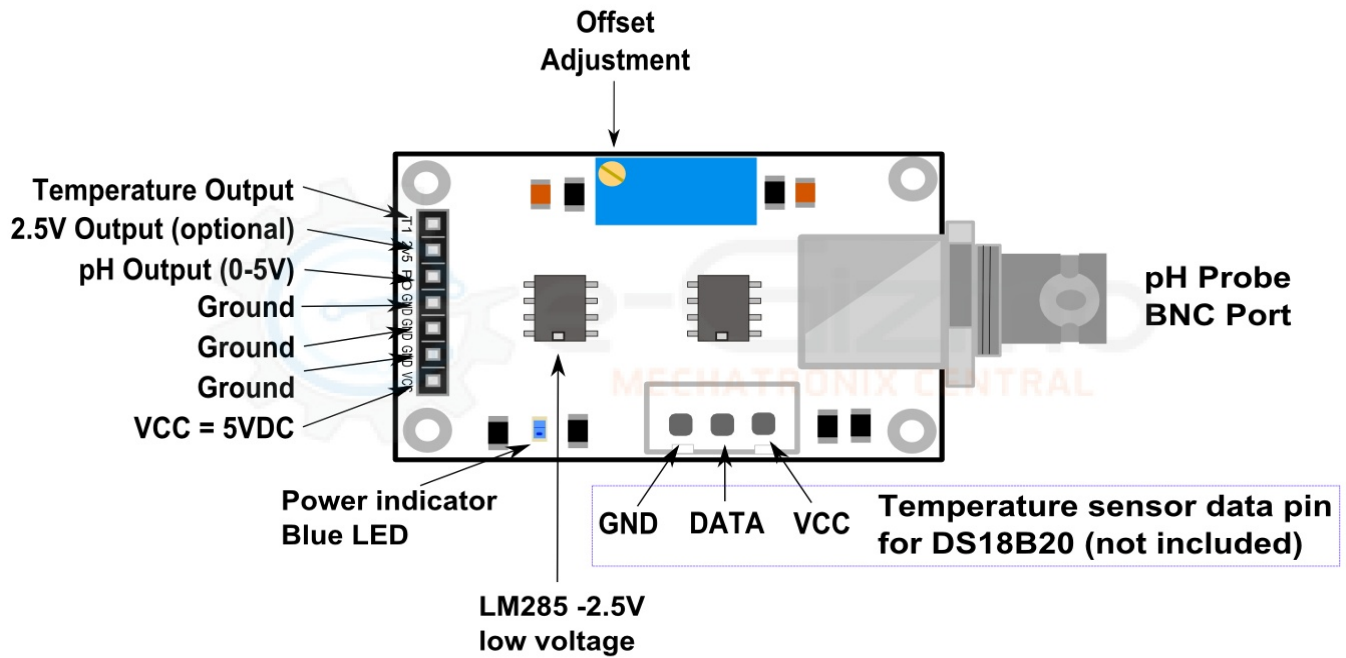
pH and ORP probes are both used for measuring the acidic intensity of liquid solutions. A pH probe measures acidity on a scale from 0 to 14, with 0 being the most acidic and 14 being the most basic. Similarly, an Oxidation-Reduction Potential (ORP) probe returns a voltage proportional to the tendency of the solution to gain or lose electrons from other substances (which is linked directly to the pH a substance).

## Applications:

- Can be used as different tester, pH meter or controller
- Suitable for most aquariums, hydroponics, laboratory etc

## General Specifications:

- Input supply voltage:** 5V
- Working current:** 5 – 10mA
- Detection concentration range:** PH 0 – 14
- Detection range of temperature:** 0 – 80 degC
- Response Time::** ≤ 5S
- Stability Time:** ≤ 60S
- Output:** Analog
- Power Consumption:** ≤ 0.5W
- Working Temperature:** -10 to +50 deg C
- Working Humidity:** 95%RH (nominal humidity 65%RH)
- Weight:** 25g
- PCB Dimension:** 42mm x 32mm x 20mm



**Figure 1: pH Sensor module kit v2 pinouts**



**Figure 2: pH Substance (solutions for testing reference)**

**IMPORTANT TIPS:**

**On How to Clean pH Electrodes:**

1. Do not "wipe" or rub the electrode.
2. Swirl the electrode gently in the cleaning solution.
3. Gently rinse with deionized or distilled water.
4. Store in a storage solution.
5. When possible, use a specialized electrode.

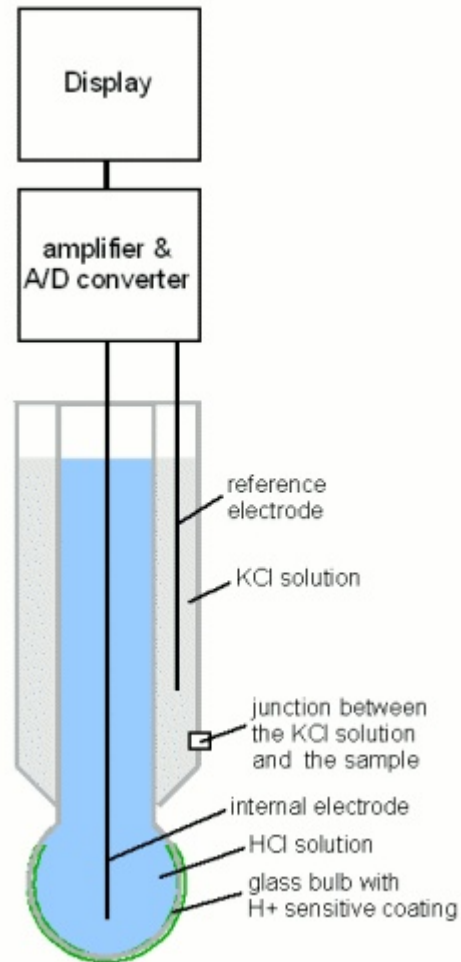
**Which Cleaning Solution to use?**

The cleaning solution you use will depend upon your particular process and the residues you are trying to remove. There is a wide range of pre-mixed cleaning solutions available online or you can make your own. Make sure you take care when handling any cleaning solution – some can be hazardous so make sure you follow all safety instructions and wear appropriate protection equipment!

(For more information - Please go to reference link)

**Reference:**

**<https://www.southforkinst.com/ph-electrodes-clean-repair/>**



**Figure 3: pH Electrode parts**

**Figure 1: Reference pH Value and Output voltage.**

pH Value	Output
4	3.071
7	2.535
10	2.066

**Table 2. pH Value of the known substances.**

0	5% Sulphuric acid, H <sub>2</sub> SO <sub>4</sub> , battery acid.
1	0.1 N HCl, hydrochloric acid (1.1)
2	Lemon juice. Vinegar (2.4-3.4)
3	wine (3.5-3.7)
4	Orange juice. Apple juice (3.8). Beer. Tomatoes.
5	Cottage cheese. Black coffee. Rain water 5.6.
6	Milk. Fish (6.7-7). chicken (6.4-6.6).
7	Neutral: equal numbers of hydrogen and hydroxyl ions. Blood (7.3-7.4). Distilled water without CO <sub>2</sub> , after boiling.
8	Sea water (8.1). Egg white.
9	Borax. baking soda.
10	Milk of magnesia, Magnesium hydroxide Mg (OH) <sub>2</sub> .
11	Household ammonia
12	Photographic developer, household bleach
13	Oven cleaner
14	Sodium lye NaOH, 1 mol/litre.

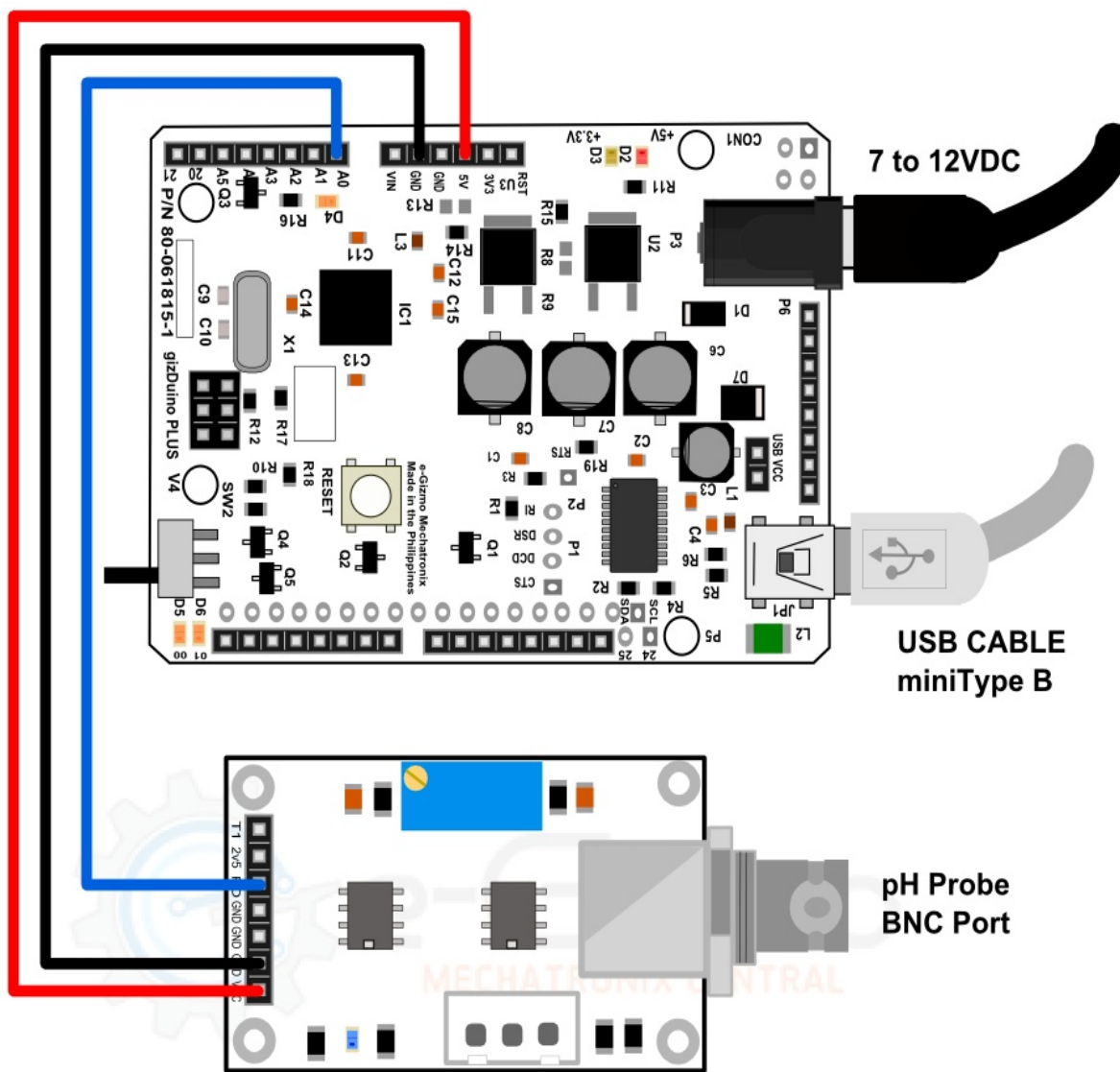


Figure 4: Sample connections.

```

/*
  e-Gizmo pH Sensor kit module v2.0

  This sample sketch is for pH sensor module
  an easy output voltage to compare it to pH value.
  See the table below:
  pH Value | Output (V)
  =====
    4      3.071
    7      2.535
   10      2.066

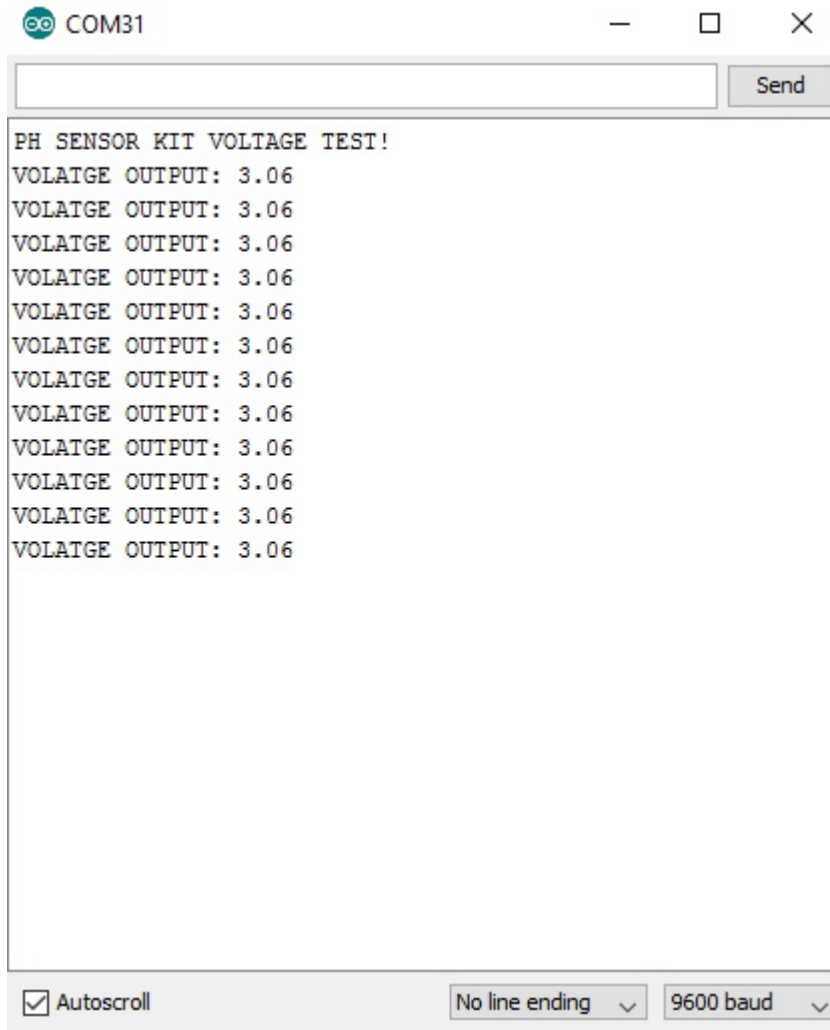
  For pH scale table for reference of the known substance.
  Here's the link: http://www.seafriends.org.nz/dda/ph.htm
  (See the pH scale)

  Modified by e-Gizmo Mechatronix Central
  http://www.e-gizmo.com
  July 18,2017

*/
#define SENSOR A0          //pH meter Analog output to Arduino Analog Input 0
#define OFFSET 0.00       //deviation compensate
#define LED 13
#define SAMPLING_INTERVAL 20
#define PRINT_INTERVAL 800
#define ARRAY_LENGTH 40 //times of collection
int PH_ARRAY[ARRAY_LENGTH]; //Store the average value of the sensor feedback
int PH_ARRAY_INDEX=0;
void setup(void)
{
  pinMode(LED,OUTPUT);
  Serial.begin(9600);
  Serial.println("PH SENSOR KIT VOLTAGE TEST!"); //Test the serial monitor
}
void loop(void)
{
  static unsigned long SAMPLING_TIME = millis();
  static unsigned long PRINT_TIME = millis();
  static float VOLTAGE;
  if(millis()-SAMPLING_TIME > SAMPLING_INTERVAL)
  {
    PH_ARRAY[PH_ARRAY_INDEX++]=analogRead(SENSOR);
    if(PH_ARRAY_INDEX==ARRAY_LENGTH)PH_ARRAY_INDEX=0;
    VOLTAGE = AVERAGE_ARRAY(PH_ARRAY, ARRAY_LENGTH)*5.0/1024;
    SAMPLING_TIME=millis();
  }
  if(millis() - PRINT_TIME > PRINT_INTERVAL) //Every 800 milliseconds, print a numerical, convert
  the state of the LED indicator
  {
    Serial.print("VOLATGE OUTPUT: ");
  }
}

```

```
        Serial.println(VOLTAGE,2);
        digitalWrite(LED,digitalRead(LED)^1);
        PRINT_TIME=millis();
    }
}
double AVERAGE_ARRAY(int* ARR, int NUMBER){
    int i;
    int max,min;
    double AVG;
    long AMOUNT=0;
    if(NUMBER<=0){
        Serial.println("ERROR!/n");
        return 0;
    }
    if(NUMBER<5){ //less than 5, calculated directly statistics
        for(i=0;i<NUMBER;i++){
            AMOUNT+=ARR[i];
        }
        AVG = AMOUNT/NUMBER;
        return AVG;
    }else{
        if(ARR[0]<ARR[1]){
            min = ARR[0];max=ARR[1];
        }
        else{
            min=ARR[1];max=ARR[0];
        }
        for(i=2;i<NUMBER;i++){
            if(ARR[i]<min){
                AMOUNT+=min; //arr<min
                min=ARR[i];
            }else {
                if(ARR[i]>max){
                    AMOUNT+=max; //arr>max
                    max= AMOUNT+=ARR[i];
                }else{
                    AMOUNT+=ARR[i]; //min<=arr<=max
                }
            }
        }
    }
    AVG = (double)AMOUNT/(NUMBER-2);
    return AVG;
}
```



**Figure 5: Serial print output from pH sensor.**



# ESP32 Series

## Datasheet

### Including:

ESP32-D0WD-V3

ESP32-U4WDH

ESP32-S0WD

ESP32-D0WD – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6 – [Not Recommended for New Designs \(NRND\)](#)

ESP32-D0WDQ6-V3 – [Not Recommended for New Designs \(NRND\)](#)



Version 3.8  
Espressif Systems  
Copyright © 2021

## About This Document

This document provides the specifications of ESP32 series of chips.

## Document Updates

Please always refer to the latest version on <https://www.espressif.com/en/support/download/documents>.

## Revision History

For any changes to this document over time, please refer to the [last page](#).

## Documentation Change Notification

Espressif provides email notifications to keep customers updated on changes to technical documentation.

Please subscribe at [www.espressif.com/en/subscribe](http://www.espressif.com/en/subscribe). Note that you need to update your subscription to receive notifications of new products you are not currently subscribed to.

## Certification

Download certificates for Espressif products from [www.espressif.com/en/certificates](http://www.espressif.com/en/certificates).

# Contents

<b>1</b>	<b>Overview</b>	<b>8</b>
1.1	Featured Solutions	8
1.1.1	Ultra-Low-Power Solution	8
1.1.2	Complete Integration Solution	8
1.2	Wi-Fi Key Features	8
1.3	Bluetooth Key Features	9
1.4	MCU and Advanced Features	9
1.4.1	CPU and Memory	9
1.4.2	Clocks and Timers	10
1.4.3	Advanced Peripheral Interfaces	10
1.4.4	Security	10
1.5	Applications (A Non-exhaustive List)	11
1.6	Block Diagram	12
<b>2</b>	<b>Pin Definitions</b>	<b>13</b>
2.1	Pin Layout	13
2.2	Pin Description	15
2.3	Power Scheme	18
2.4	Strapping Pins	19
<b>3</b>	<b>Functional Description</b>	<b>22</b>
3.1	CPU and Memory	22
3.1.1	CPU	22
3.1.2	Internal Memory	22
3.1.3	External Flash and SRAM	23
3.1.4	Memory Map	23
3.2	Timers and Watchdogs	25
3.2.1	64-bit Timers	25
3.2.2	Watchdog Timers	25
3.3	System Clocks	26
3.3.1	CPU Clock	26
3.3.2	RTC Clock	26
3.3.3	Audio PLL Clock	26
3.4	Radio	26
3.4.1	2.4 GHz Receiver	27
3.4.2	2.4 GHz Transmitter	27
3.4.3	Clock Generator	27
3.5	Wi-Fi	27
3.5.1	Wi-Fi Radio and Baseband	27
3.5.2	Wi-Fi MAC	28
3.6	Bluetooth	28
3.6.1	Bluetooth Radio and Baseband	28

3.6.2	Bluetooth Interface	29
3.6.3	Bluetooth Stack	29
3.6.4	Bluetooth Link Controller	29
3.7	RTC and Low-Power Management	30
<b>4</b>	<b>Peripherals and Sensors</b>	<b>32</b>
4.1	Descriptions of Peripherals and Sensors	32
4.1.1	General Purpose Input/Output Interface (GPIO)	32
4.1.2	Analog-to-Digital Converter (ADC)	32
4.1.3	Hall Sensor	33
4.1.4	Digital-to-Analog Converter (DAC)	33
4.1.5	Touch Sensor	33
4.1.6	Ultra-Low-Power (ULP) Coprocessor	33
4.1.7	Ethernet MAC Interface	34
4.1.8	SD/SDIO/MMC Host Controller	34
4.1.9	SDIO/SPI Slave Controller	34
4.1.10	Universal Asynchronous Receiver Transmitter (UART)	35
4.1.11	I2C Interface	35
4.1.12	I2S Interface	35
4.1.13	Infrared Remote Controller (RMT)	35
4.1.14	Pulse Counter	35
4.1.15	Pulse Width Modulation (PWM)	36
4.1.16	LED PWM	36
4.1.17	Serial Peripheral Interface (SPI)	36
4.1.18	TWAI Controller	36
4.1.19	Accelerator	37
4.2	Peripheral Pin Configurations	37
<b>5</b>	<b>Electrical Characteristics</b>	<b>42</b>
5.1	Absolute Maximum Ratings	42
5.2	Recommended Operating Conditions	42
5.3	DC Characteristics (3.3 V, 25 °C)	43
5.4	Reliability Qualifications	43
5.5	RF Power-Consumption Specifications	44
5.6	Wi-Fi Radio	44
5.7	Bluetooth Radio	45
5.7.1	Receiver – Basic Data Rate	45
5.7.2	Transmitter – Basic Data Rate	45
5.7.3	Receiver – Enhanced Data Rate	46
5.7.4	Transmitter – Enhanced Data Rate	46
5.8	Bluetooth LE Radio	47
5.8.1	Receiver	47
5.8.2	Transmitter	48
<b>6</b>	<b>Package Information</b>	<b>49</b>

<b>7</b>	<b>Part Number and Ordering Information</b>	50
<b>8</b>	<b>Related Documentation and Resources</b>	51
	<b>Appendix A – ESP32 Pin Lists</b>	52
	A.1. Notes on ESP32 Pin Lists	52
	A.2. GPIO_Matrix	54
	A.3. Ethernet_MAC	59
	A.4. IO_MUX	59
	<b>Revision History</b>	61

## List of Tables

1	Pin Description	15
2	Description of ESP32 Power-up and Reset Timing Parameters	19
3	Strapping Pins	20
4	Parameter Descriptions of Setup and Hold Times for the Strapping Pins	21
5	Memory and Peripheral Mapping	24
6	Power Consumption by Power Modes	30
7	ADC Characteristics	32
8	ADC Calibration Results	33
9	Capacitive-Sensing GPIOs Available on ESP32	33
10	Peripheral Pin Configurations	37
11	Absolute Maximum Ratings	42
12	Recommended Operating Conditions	42
13	DC Characteristics (3.3 V, 25 °C)	43
14	Reliability Qualifications	43
15	RF Power-Consumption Specifications	44
16	Wi-Fi Radio Characteristics	44
17	Receiver Characteristics – Basic Data Rate	45
18	Transmitter Characteristics – Basic Data Rate	45
19	Receiver Characteristics – Enhanced Data Rate	46
20	Transmitter Characteristics – Enhanced Data Rate	47
21	Receiver Characteristics – Bluetooth LE	47
22	Transmitter Characteristics – Bluetooth LE	48
23	ESP32 Ordering Information	50
24	Notes on ESP32 Pin Lists	52
25	GPIO_Matrix	54
26	Ethernet_MAC	59

## List of Figures

1	Functional Block Diagram	12
2	ESP32 Pin Layout (QFN 6*6, Top View)	13
3	ESP32 Pin Layout (QFN 5*5, Top View)	14
4	ESP32 Power Scheme	18
5	ESP32 Power-up and Reset Timing	19
6	Setup and Hold Times for the Strapping Pins	21
7	Address Mapping Structure	23
8	QFN48 (6x6 mm) Package	49
9	QFN48 (5x5 mm) Package	49
10	ESP32 Part Number	50

# 1 Overview

ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with the TSMC ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios.

The ESP32 series of chips includes ESP32-D0WD-V3, ESP32-U4WDH, ESP32-S0WD, ESP32-D0WDQ6-V3 ([NRND](#)), ESP32-D0WD ([NRND](#)), and ESP32-D0WDQ6 ([NRND](#)), among which, ESP32-D0WD-V3, ESP32-D0WDQ6-V3 ([NRND](#)), and ESP32-U4WDH are based on ECO V3 wafer.

For details on part numbers and ordering information, please refer to Section 7.

For details on ECO V3 instructions, please refer to [ESP32 ECO V3 User Guide](#).

## 1.1 Featured Solutions

### 1.1.1 Ultra-Low-Power Solution

ESP32 is designed for mobile, wearable electronics, and Internet-of-Things (IoT) applications. It features all the state-of-the-art characteristics of low-power chips, including fine-grained clock gating, multiple power modes, and dynamic power scaling. For instance, in a low-power IoT sensor hub application scenario, ESP32 is woken up periodically only when a specified condition is detected. Low-duty cycle is used to minimize the amount of energy that the chip expends. The output of the power amplifier is also adjustable, thus contributing to an optimal trade-off between communication range, data rate and power consumption.

**Note:**

For more information, refer to Section 3.7 *RTC and Low-Power Management*.

### 1.1.2 Complete Integration Solution

ESP32 is a highly-integrated solution for Wi-Fi-and-Bluetooth IoT applications, with around 20 external components. ESP32 integrates an antenna switch, RF balun, power amplifier, low-noise receive amplifier, filters, and power management modules. As such, the entire solution occupies minimal Printed Circuit Board (PCB) area.

ESP32 uses CMOS for single-chip fully-integrated radio and baseband, while also integrating advanced calibration circuitries that allow the solution to remove external circuit imperfections or adjust to changes in external conditions. As such, the mass production of ESP32 solutions does not require expensive and specialized Wi-Fi testing equipment.

## 1.2 Wi-Fi Key Features

- 802.11 b/g/n
- 802.11 n (2.4 GHz), up to 150 Mbps
- WMM
- TX/RX A-MPDU, RX A-MSDU
- Immediate Block ACK



- Defragmentation
- Automatic Beacon monitoring (hardware TSF)
- 4 × virtual Wi-Fi interfaces
- Simultaneous support for Infrastructure Station, SoftAP, and Promiscuous modes  
Note that when ESP32 is in Station mode, performing a scan, the SoftAP channel will be changed.
- Antenna diversity

**Note:**

For more information, please refer to Section [3.5 Wi-Fi](#).

## 1.3 Bluetooth Key Features

- Compliant with Bluetooth v4.2 BR/EDR and Bluetooth LE specifications
- Class-1, class-2 and class-3 transmitter without external power amplifier
- Enhanced Power Control
- +9 dBm transmitting power
- NZIF receiver with -94 dBm Bluetooth LE sensitivity
- Adaptive Frequency Hopping (AFH)
- Standard HCI based on SDIO/SPI/UART
- High-speed UART HCI, up to 4 Mbps
- Bluetooth 4.2 BR/EDR Bluetooth LE dual mode controller
- Synchronous Connection-Oriented/Extended (SCO/eSCO)
- CVSD and SBC for audio codec
- Bluetooth Piconet and Scatternet
- Multi-connections in Classic Bluetooth and Bluetooth LE
- Simultaneous advertising and scanning

## 1.4 MCU and Advanced Features

### 1.4.1 CPU and Memory

- Xtensa® single-/dual-core 32-bit LX6 microprocessor(s)
- CoreMark® score:
  - 1 core at 240 MHz: 504.85 CoreMark; 2.10 CoreMark/MHz
  - 2 cores at 240 MHz: 994.26 CoreMark; 4.14 CoreMark/MHz
- 448 KB ROM
- 520 KB SRAM

- 16 KB SRAM in RTC
- QSPI supports multiple flash/SRAM chips

### 1.4.2 Clocks and Timers

- Internal 8 MHz oscillator with calibration
- Internal RC oscillator with calibration
- External 2 MHz ~ 60 MHz crystal oscillator (40 MHz only for Wi-Fi/Bluetooth functionality)
- External 32 kHz crystal oscillator for RTC with calibration
- Two timer groups, including 2 × 64-bit timers and 1 × main watchdog in each group
- One RTC timer
- RTC watchdog

### 1.4.3 Advanced Peripheral Interfaces

- 34 × programmable GPIOs
- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit DAC
- 10 × touch sensors
- 4 × SPI
- 2 × I2S
- 2 × I2C
- 3 × UART
- 1 host (SD/eMMC/SDIO)
- 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- TWAI<sup>®</sup>, compatible with ISO 11898-1 (CAN Specification 2.0)
- RMT (TX/RX)
- Motor PWM
- LED PWM up to 16 channels
- Hall sensor

### 1.4.4 Security

- Secure boot
- Flash encryption
- 1024-bit OTP, up to 768-bit for customers
- Cryptographic hardware acceleration:

- AES
- Hash (SHA-2)
- RSA
- ECC
- Random Number Generator (RNG)

## 1.5 Applications (A Non-exhaustive List)

- Generic Low-power IoT Sensor Hub
  - Agriculture robotics
- Generic Low-power IoT Data Loggers
- Cameras for Video Streaming
- Over-the-top (OTT) Devices
- Speech Recognition
- Image Recognition
- Mesh Network
- Home Automation
  - Light control
  - Smart plugs
  - Smart door locks
- Smart Building
  - Smart lighting
  - Energy monitoring
- Industrial Automation
  - Industrial wireless control
  - Industrial robotics
- Smart Agriculture
  - Smart greenhouses
  - Smart irrigation
- Audio Applications
  - Internet music players
  - Live streaming devices
  - Internet radio players
  - Audio headsets
- Health Care Applications
  - Health monitoring
  - Baby monitors
- Wi-Fi-enabled Toys
  - Remote control toys
  - Proximity sensing toys
  - Educational toys
- Wearable Electronics
  - Smart watches
  - Smart bracelets
- Retail & Catering Applications
  - POS machines
  - Service robots

## 1.6 Block Diagram

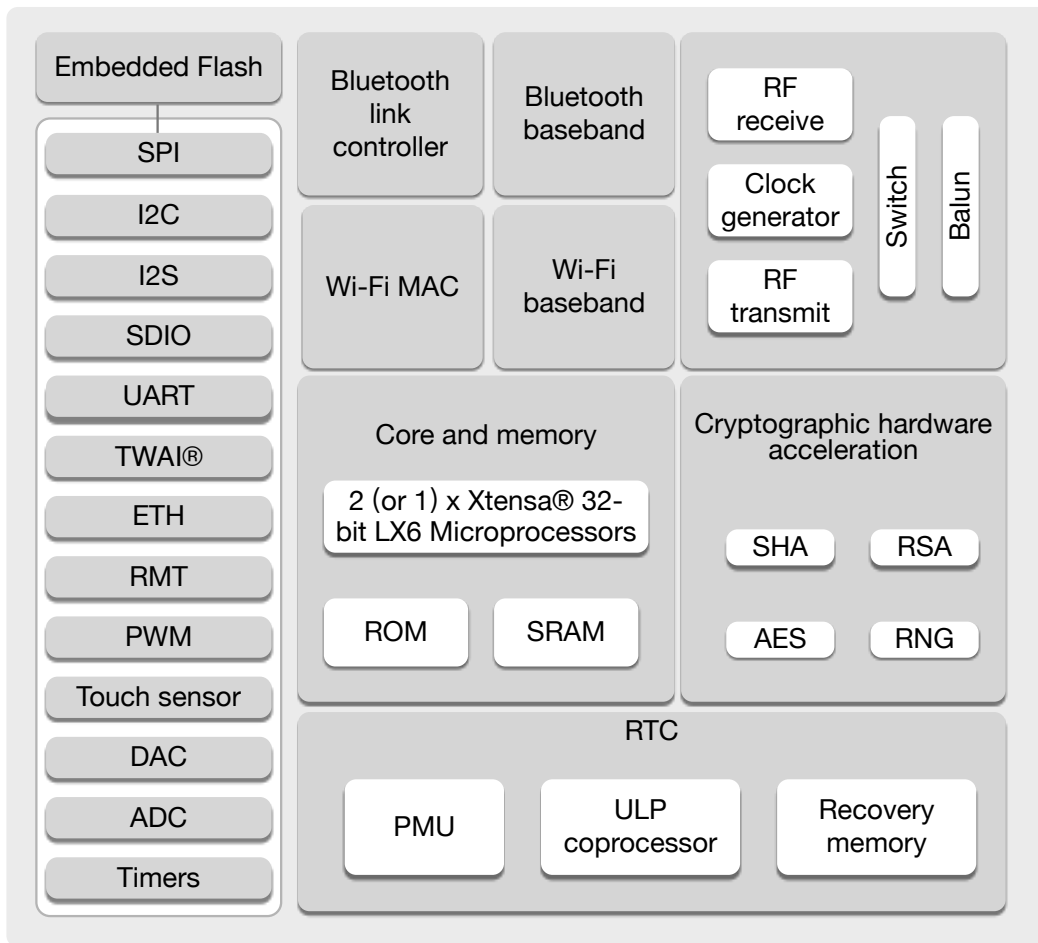


Figure 1: Functional Block Diagram

**Note:**

Products in the ESP32 series differ from each other in terms of their support for embedded flash and the number of CPUs they have. For details, please refer to Section 7 *Part Number and Ordering Information*.

## 2 Pin Definitions

### 2.1 Pin Layout

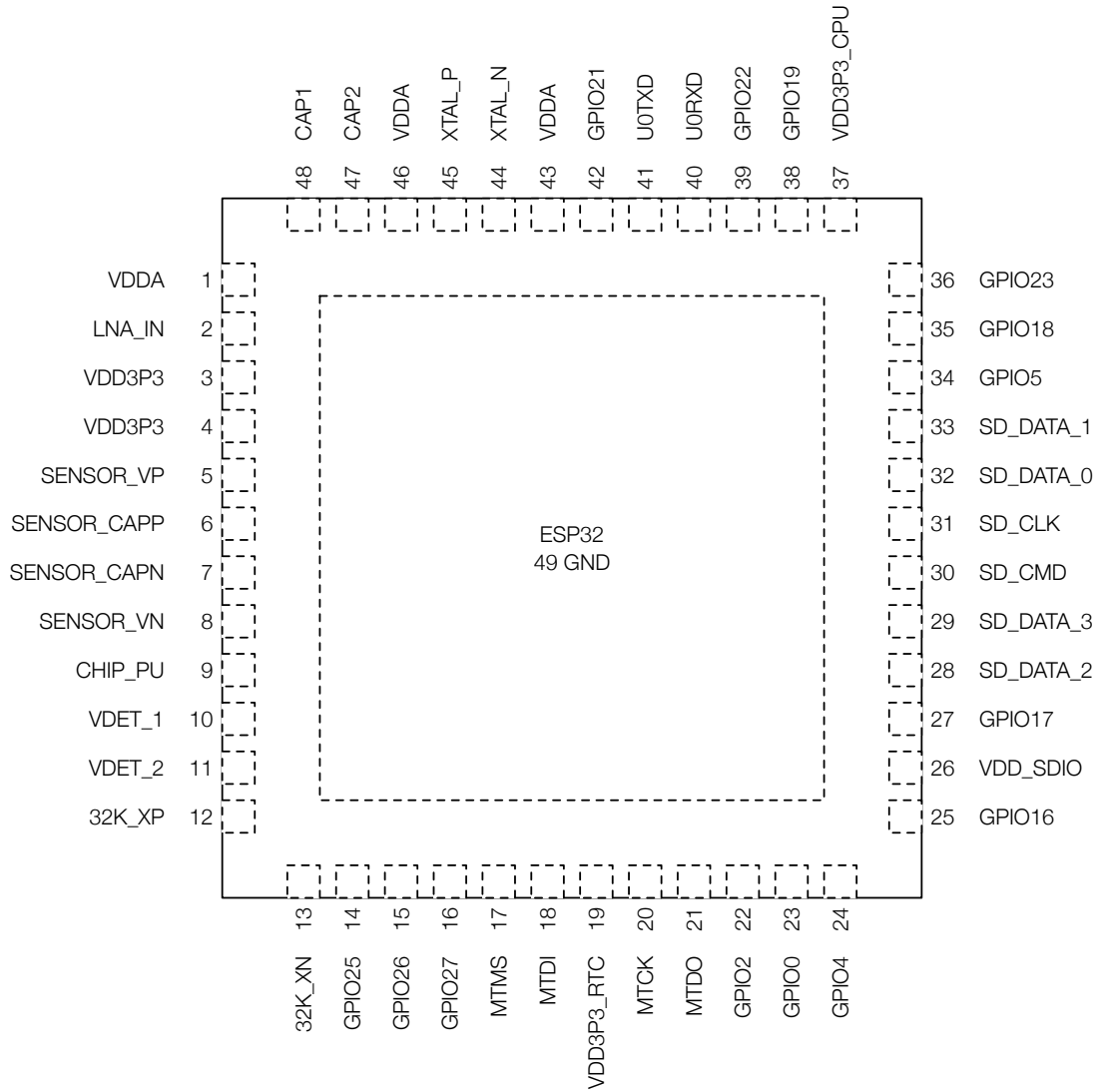


Figure 2: ESP32 Pin Layout (QFN 6\*6, Top View)

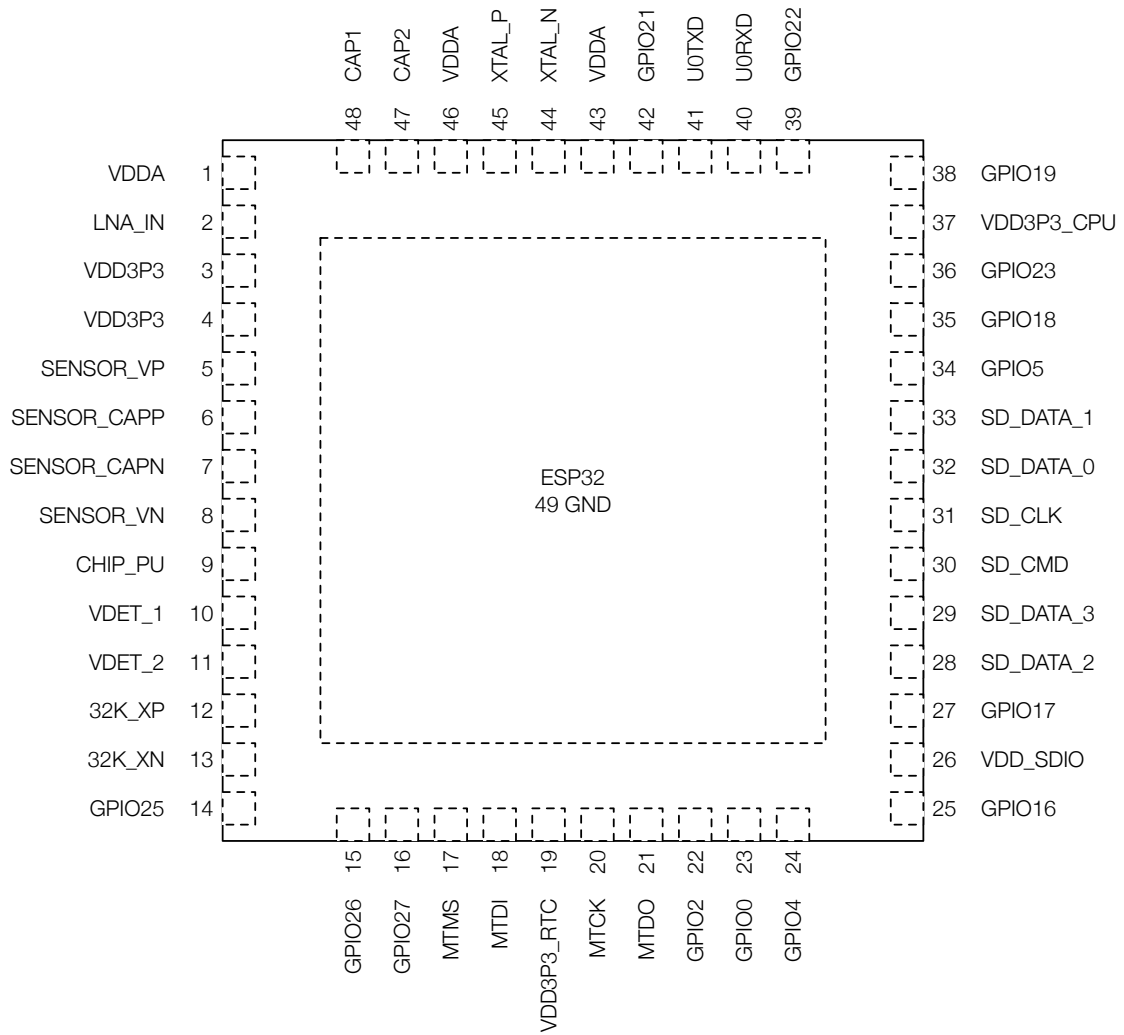


Figure 3: ESP32 Pin Layout (QFN 5\*5, Top View)

**Note:**

For details on ESP32's part numbers and the corresponding packaging, please refer to Section 7 *Part Number and Ordering Information*.

## 2.2 Pin Description

Table 1: Pin Description

Name	No.	Type	Function
Analog			
VDDA	1	P	Analog power supply (2.3 V ~ 3.6 V)
LNA_IN	2	I/O	RF input and output
VDD3P3	3	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3	4	P	Analog power supply (2.3 V ~ 3.6 V)
VDD3P3_RTC			
SENSOR_VP	5	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_CAPP	6	I	GPIO37, ADC1_CH1, RTC_GPIO1
SENSOR_CAPN	7	I	GPIO38, ADC1_CH2, RTC_GPIO2
SENSOR_VN	8	I	GPIO39, ADC1_CH3, RTC_GPIO3
CHIP_PU	9	I	High: On; enables the chip Low: Off; the chip powers off Note: Do not leave the CHIP_PU pin floating.
VDET_1	10	I	GPIO34, ADC1_CH6, RTC_GPIO4
VDET_2	11	I	GPIO35, ADC1_CH7, RTC_GPIO5
32K_XP	12	I/O	GPIO32, ADC1_CH4, RTC_GPIO9, TOUCH9, 32K_XP (32.768 kHz crystal oscillator input)
32K_XN	13	I/O	GPIO33, ADC1_CH5, RTC_GPIO8, TOUCH8, 32K_XN (32.768 kHz crystal oscillator output)
GPIO25	14	I/O	GPIO25, ADC2_CH8, RTC_GPIO6, DAC_1, EMAC_RXD0
GPIO26	15	I/O	GPIO26, ADC2_CH9, RTC_GPIO7, DAC_2, EMAC_RXD1
GPIO27	16	I/O	GPIO27, ADC2_CH7, RTC_GPIO17, TOUCH7, EMAC_RX_DV
MTMS	17	I/O	GPIO14, ADC2_CH6, RTC_GPIO16, TOUCH6, EMAC_TXD2, HSPICLK, HS2_CLK, SD_CLK, MTMS
MTDI	18	I/O	GPIO12, ADC2_CH5, RTC_GPIO15, TOUCH5, EMAC_TXD3, HSPIQ, HS2_DATA2, SD_DATA2, MTDI
VDD3P3_RTC	19	P	Input power supply for RTC IO (2.3 V ~ 3.6 V)
MTCK	20	I/O	GPIO13, ADC2_CH4, RTC_GPIO14, TOUCH4, EMAC_RX_ER, HSPID, HS2_DATA3, SD_DATA3, MTCK
MTDO	21	I/O	GPIO15, ADC2_CH3, RTC_GPIO13, TOUCH3, EMAC_RXD3, HSPICS0, HS2_CMD, SD_CMD, MTDO

Name	No.	Type	Function
GPIO2	22	I/O	GPIO2, ADC2_CH2, RTC_GPIO12, TOUCH2, HSPiWP, HS2_DATA0, SD_DATA0
GPIO0	23	I/O	GPIO0, ADC2_CH1, RTC_GPIO11, TOUCH1, EMAC_TX_CLK, CLK_OUT1,
GPIO4	24	I/O	GPIO4, ADC2_CH0, RTC_GPIO10, TOUCH0, EMAC_TX_ER, HSPiHD, HS2_DATA1, SD_DATA1
VDD_SDIO			
GPIO16	25	I/O	GPIO16, HS1_DATA4, U2RXD, EMAC_CLK_OUT
VDD_SDIO	26	P	Output power supply: 1.8 V or the same voltage as VDD3P3_RTC
GPIO17	27	I/O	GPIO17, HS1_DATA5, U2TXD, EMAC_CLK_OUT_180
SD_DATA_2	28	I/O	GPIO9, HS1_DATA2, U1RXD, SD_DATA2, SPiHD
SD_DATA_3	29	I/O	GPIO10, HS1_DATA3, U1TXD, SD_DATA3, SPiWP
SD_CMD	30	I/O	GPIO11, HS1_CMD, U1RTS, SD_CMD, SPiCS0
SD_CLK	31	I/O	GPIO6, HS1_CLK, U1CTS, SD_CLK, SPiCLK
SD_DATA_0	32	I/O	GPIO7, HS1_DATA0, U2RTS, SD_DATA0, SPiQ
SD_DATA_1	33	I/O	GPIO8, HS1_DATA1, U2CTS, SD_DATA1, SPiD
VDD3P3_CPU			
GPIO5	34	I/O	GPIO5, HS1_DATA6, VSPiCS0, EMAC_RX_CLK
GPIO18	35	I/O	GPIO18, HS1_DATA7, VSPiCLK
GPIO23	36	I/O	GPIO23, HS1_STROBE, VSPiD
VDD3P3_CPU	37	P	Input power supply for CPU IO (1.8 V ~ 3.6 V)
GPIO19	38	I/O	GPIO19, U0CTS, VSPIQ, EMAC_TXD0
GPIO22	39	I/O	GPIO22, U0RTS, VSPiWP, EMAC_TXD1
U0RXD	40	I/O	GPIO3, U0RXD, CLK_OUT2
U0TXD	41	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
GPIO21	42	I/O	GPIO21, VSPiHD, EMAC_TX_EN
Analog			
VDDA	43	P	Analog power supply (2.3 V ~ 3.6 V)
XTAL_N	44	O	External crystal output
XTAL_P	45	I	External crystal input
VDDA	46	P	Analog power supply (2.3 V ~ 3.6 V)
CAP2	47	I	Connects to a 3.3 nF (10%) capacitor and 20 k $\Omega$ resistor in parallel to CAP1



Name	No.	Type	Function
CAP1	48	I	Connects to a 10 nF series capacitor to ground
GND	49	P	Ground

**Note:**

- The pin-pin mapping between ESP32-U4WDH and the embedded flash is as follows: GPIO16 = CS#, GPIO17 = IO1/DO, SD\_CMD = IO3/HOLD#, SD\_CLK = CLK, SD\_DATA\_0 = IO2/WP#, SD\_DATA\_1 = IO0/DI. The pins used for embedded flash are not recommended for other uses.
- In most cases, the data port connection between ESP32 series of chips other than ESP32-U4WDH and external flash is as follows: SD\_DATA0/SPIQ = IO1/DO, SD\_DATA1/SPID = IO0/DI, SD\_DATA2/SPIHD = IO3/HOLD#, SD\_DATA3/SPIWP = IO2/WP#.
- For a quick reference guide to using the IO\_MUX, Ethernet MAC, and GPIO Matrix pins of ESP32, please refer to Appendix [ESP32 Pin Lists](#).

## 2.3 Power Scheme

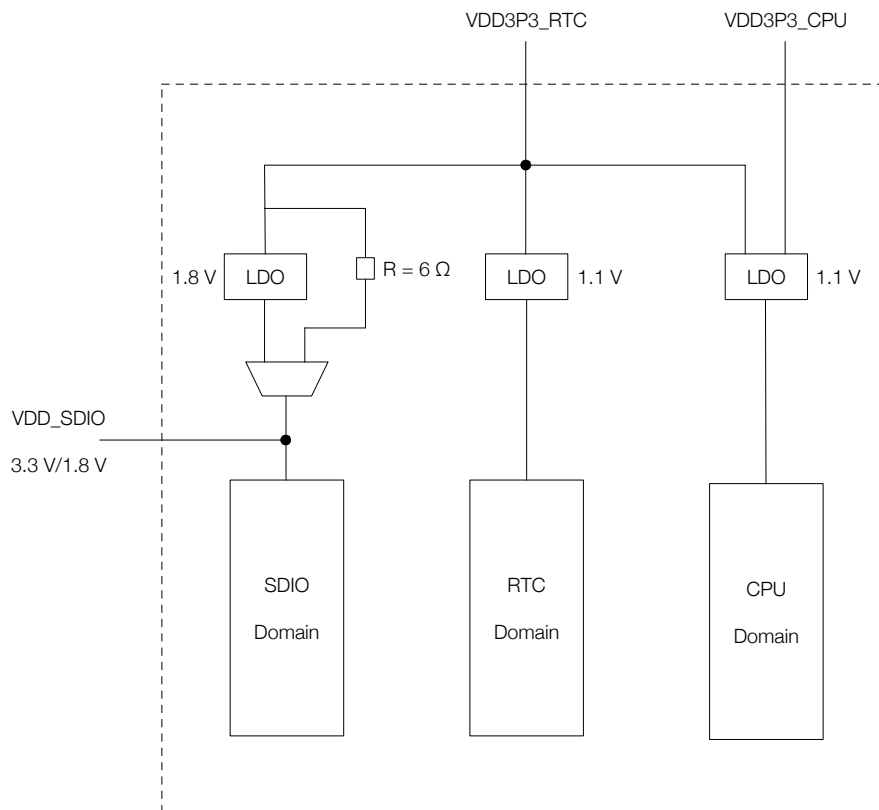
ESP32's digital pins are divided into three different power domains:

- VDD3P3\_RTC
- VDD3P3\_CPU
- VDD\_SDIO

VDD3P3\_RTC is also the input power supply for RTC and CPU.

VDD3P3\_CPU is also the input power supply for CPU.

VDD\_SDIO connects to the output of an internal LDO whose input is VDD3P3\_RTC. When VDD\_SDIO is connected to the same PCB net together with VDD3P3\_RTC, the internal LDO is disabled automatically. The power scheme diagram is shown below:



**Figure 4: ESP32 Power Scheme**

The internal LDO can be configured as having 1.8 V, or the same voltage as VDD3P3\_RTC. It can be powered off via software to minimize the current of flash/SRAM during the Deep-sleep mode.

### Notes on CHIP\_PU:

- The illustration below shows the ESP32 power-up and reset timing. Details about the parameters are listed in Table 2.

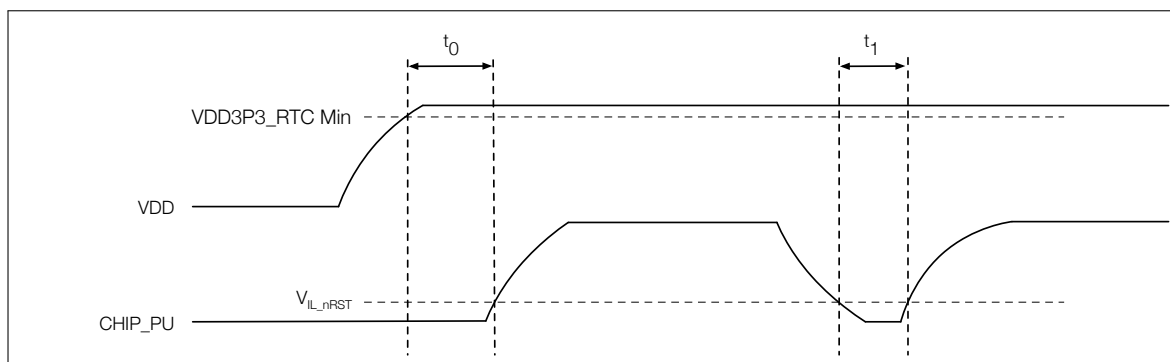


Figure 5: ESP32 Power-up and Reset Timing

Table 2: Description of ESP32 Power-up and Reset Timing Parameters

Parameters	Description	Min.	Unit
$t_0$	Time between the 3.3 V rails being brought up and CHIP_PU being activated	50	$\mu\text{s}$
$t_1$	Duration of CHIP_PU signal level $< V_{IL\_nRST}$ (refer to its value in Table 13 DC Characteristics) to reset the chip	50	$\mu\text{s}$

- In scenarios where ESP32 is powered on and off repeatedly by switching the power rails, while there is a large capacitor on the VDD33 rail and CHIP\_PU and VDD33 are connected, simply switching off the CHIP\_PU power rail and immediately switching it back on may cause an incomplete power discharge cycle and failure to reset the chip adequately.  
An additional discharge circuit may be required to accelerate the discharge of the large capacitor on rail VDD33, which will ensure proper power-on-reset when the ESP32 is powered up again.
- When a battery is used as the power supply for the ESP32 series of chips and modules, a supply voltage supervisor is recommended, so that a boot failure due to low voltage is avoided. Users are recommended to pull CHIP\_PU low if the power supply for ESP32 is below 2.3 V.

#### Notes on power supply:

- The operating voltage of ESP32 ranges from 2.3 V to 3.6 V. When using a single-power supply, the recommended voltage of the power supply is 3.3 V, and its recommended output current is 500 mA or more.
- When VDD\_SDIO 1.8 V is used as the power supply for external flash/PSRAM, a 2 k $\Omega$  grounding resistor should be added to VDD\_SDIO. For the circuit design, please refer to Figure **ESP32-WROVER Schematics**, in [ESP32-WROVER Datasheet](#).
- When the three digital power supplies are used to drive peripherals, e.g., 3.3 V flash, they should comply with the peripherals' specifications.

## 2.4 Strapping Pins

There are five strapping pins:

- MTDI
- GPIO0
- GPIO2

- MTDO
- GPIO5

Software can read the values of these five bits from register "GPIO\_STRAPPING".

During the chip's system reset release (power-on-reset, RTC watchdog reset and brownout reset), the latches of the strapping pins sample the voltage level as strapping bits of "0" or "1", and hold these bits until the chip is powered down or shut down. The strapping bits configure the device's boot mode, the operating voltage of VDD\_SDIO and other initial system settings.

Each strapping pin is connected to its internal pull-up/pull-down during the chip reset. Consequently, if a strapping pin is unconnected or the connected external circuit is high-impedance, the internal weak pull-up/pull-down will determine the default input level of the strapping pins.

To change the strapping bit values, users can apply the external pull-down/pull-up resistances, or use the host MCU's GPIOs to control the voltage level of these pins when powering on the chip.

After reset release, the strapping pins work as normal-function pins.

Refer to Table 3 for a detailed boot-mode configuration by strapping pins.

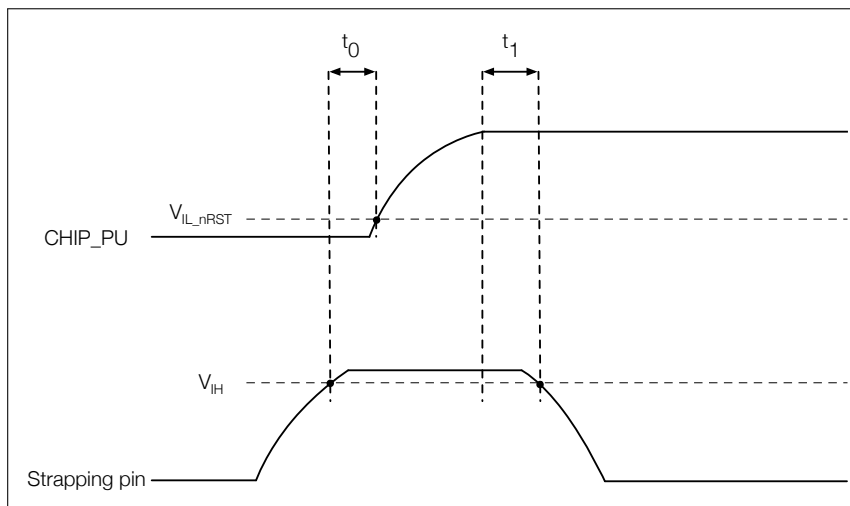
**Table 3: Strapping Pins**

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Bootling Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Bootling					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	FE Sampling FE Output	FE Sampling RE Output	RE Sampling FE Output	RE Sampling RE Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

**Note:**

- FE: falling-edge, RE: rising-edge.
- Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD\_SDIO)" and "Timing of SDIO Slave", after bootling.
- For ESP32 chips that contain an embedded flash, users need to note the logic level of MTDI. For example, ESP32-U4WDH contains an embedded flash that operates at 3.3 V, therefore, the MTDI should be low.

The illustration below shows the setup and hold times for the strapping pins before and after the CHIP\_PU signal goes high. Details about the parameters are listed in Table 4.



**Figure 6: Setup and Hold Times for the Strapping Pins**

**Table 4: Parameter Descriptions of Setup and Hold Times for the Strapping Pins**

Parameters	Description	Min.	Unit
$t_0$	Setup time before CHIP_PU goes from low to high	0	ms
$t_1$	Hold time after CHIP_PU goes high	1	ms

## 3 Functional Description

This chapter describes the functions integrated in ESP32.

### 3.1 CPU and Memory

#### 3.1.1 CPU

ESP32 contains one or two low-power Xtensa® 32-bit LX6 microprocessor(s) with the following features:

- 7-stage pipeline to support the clock frequency of up to 240 MHz (160 MHz for ESP32-S0WD)
- 16/24-bit Instruction Set provides high code-density
- Support for Floating Point Unit
- Support for DSP instructions, such as a 32-bit multiplier, a 32-bit divider, and a 40-bit MAC
- Support for 32 interrupt vectors from about 70 interrupt sources

The single-/dual-CPU interfaces include:

- Xtensa RAM/ROM Interface for instructions and data
- Xtensa Local Memory Interface for fast peripheral register access
- External and internal interrupt sources
- JTAG for debugging

#### 3.1.2 Internal Memory

ESP32's internal memory includes:

- 448 KB of ROM for booting and core functions
- 520 KB of on-chip SRAM for data and instructions
- 8 KB of SRAM in RTC, which is called RTC FAST Memory and can be used for data storage; it is accessed by the main CPU during RTC Boot from the Deep-sleep mode.
- 8 KB of SRAM in RTC, which is called RTC SLOW Memory and can be accessed by the ULP coprocessor during the Deep-sleep mode.
- 1 Kbit of eFuse: 256 bits are used for the system (MAC address and chip configuration) and the remaining 768 bits are reserved for customer applications, including flash-encryption and chip-ID.
- Embedded flash

**Note:**

Products in the ESP32 series differ from each other, in terms of their support for embedded flash and the size of it. For details, please refer to Section [7 Part Number and Ordering Information](#).

### 3.1.3 External Flash and SRAM

ESP32 supports multiple external QSPI flash and SRAM chips. More details can be found in Chapter SPI in the [ESP32 Technical Reference Manual](#). ESP32 also supports hardware encryption/decryption based on AES to protect developers' programs and data in flash.

ESP32 can access the external QSPI flash and SRAM through high-speed caches.

- Up to 16 MB of external flash can be mapped into CPU instruction memory space and read-only memory space simultaneously.
  - When external flash is mapped into CPU instruction memory space, up to 11 MB + 248 KB can be mapped at a time. Note that if more than 3 MB + 248 KB are mapped, cache performance will be reduced due to speculative reads by the CPU.
  - When external flash is mapped into read-only data memory space, up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads are supported.
- External SRAM can be mapped into CPU data memory space. SRAM up to 8 MB is supported and up to 4 MB can be mapped at a time. 8-bit, 16-bit and 32-bit reads and writes are supported.

**Note:**  
After ESP32 is initialized, firmware can customize the mapping of external SRAM or flash into the CPU address space.

### 3.1.4 Memory Map

The structure of address mapping is shown in Figure 7. The memory and peripheral mapping of ESP32 is shown in Table 5.

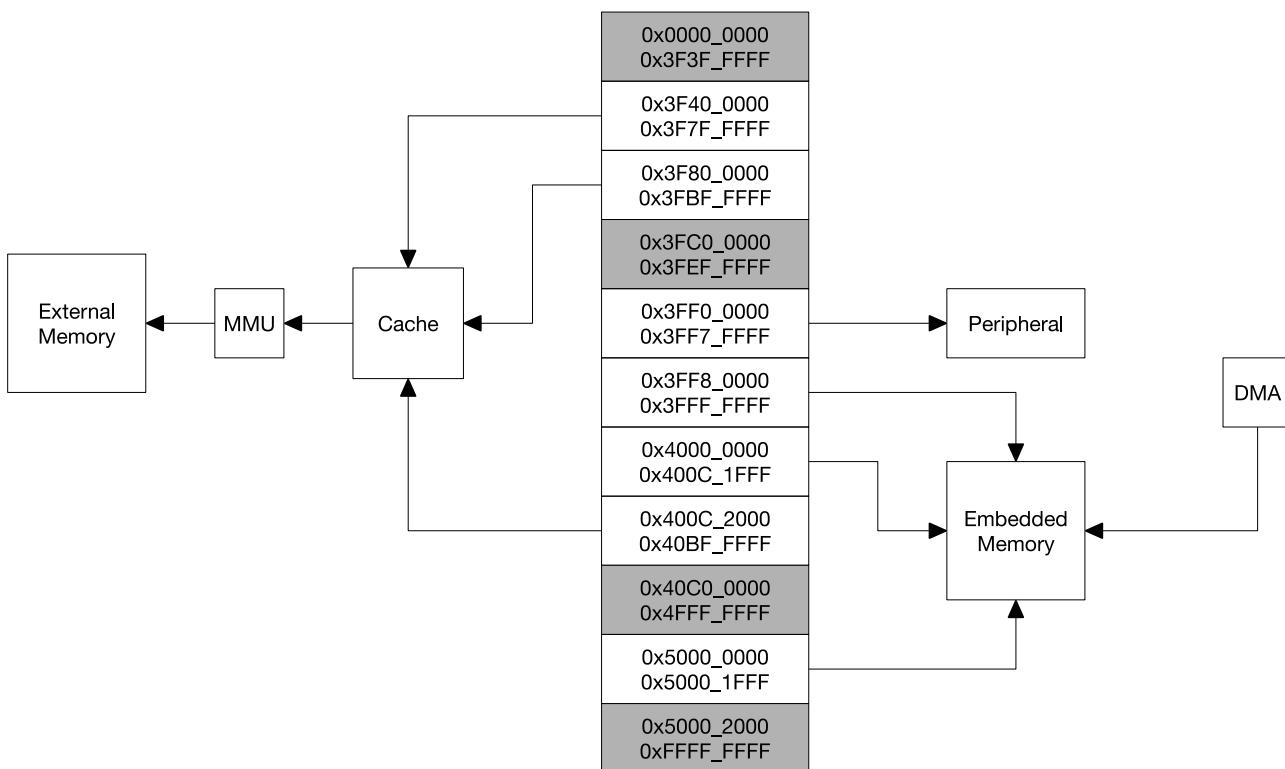


Figure 7: Address Mapping Structure

Table 5: Memory and Peripheral Mapping

Category	Target	Start Address	End Address	Size
Embedded Memory	Internal ROM 0	0x4000_0000	0x4005_FFFF	384 KB
	Internal ROM 1	0x3FF9_0000	0x3FF9_FFFF	64 KB
	Internal SRAM 0	0x4007_0000	0x4009_FFFF	192 KB
	Internal SRAM 1	0x3FFE_0000	0x3FFF_FFFF	128 KB
		0x400A_0000	0x400B_FFFF	
	Internal SRAM 2	0x3FFA_E000	0x3FFD_FFFF	200 KB
	RTC FAST Memory	0x3FF8_0000	0x3FF8_1FFF	8 KB
0x400C_0000		0x400C_1FFF		
RTC SLOW Memory	0x5000_0000	0x5000_1FFF	8 KB	
External Memory	External Flash	0x3F40_0000	0x3F7F_FFFF	4 MB
		0x400C_2000	0x40BF_FFFF	11 MB+248 KB
	External RAM	0x3F80_0000	0x3FBF_FFFF	4 MB
Peripheral	DPort Register	0x3FF0_0000	0x3FF0_0FFF	4 KB
	AES Accelerator	0x3FF0_1000	0x3FF0_1FFF	4 KB
	RSA Accelerator	0x3FF0_2000	0x3FF0_2FFF	4 KB
	SHA Accelerator	0x3FF0_3000	0x3FF0_3FFF	4 KB
	Secure Boot	0x3FF0_4000	0x3FF0_4FFF	4 KB
	Cache MMU Table	0x3FF1_0000	0x3FF1_3FFF	16 KB
	PID Controller	0x3FF1_F000	0x3FF1_FFFF	4 KB
	UART0	0x3FF4_0000	0x3FF4_0FFF	4 KB
	SPI1	0x3FF4_2000	0x3FF4_2FFF	4 KB
	SPI0	0x3FF4_3000	0x3FF4_3FFF	4 KB
	GPIO	0x3FF4_4000	0x3FF4_4FFF	4 KB
	RTC	0x3FF4_8000	0x3FF4_8FFF	4 KB
	IO MUX	0x3FF4_9000	0x3FF4_9FFF	4 KB
	SDIO Slave	0x3FF4_B000	0x3FF4_BFFF	4 KB
	UDMA1	0x3FF4_C000	0x3FF4_CFFF	4 KB
	I2S0	0x3FF4_F000	0x3FF4_FFFF	4 KB
	UART1	0x3FF5_0000	0x3FF5_0FFF	4 KB
	I2C0	0x3FF5_3000	0x3FF5_3FFF	4 KB
	UDMA0	0x3FF5_4000	0x3FF5_4FFF	4 KB
	SDIO Slave	0x3FF5_5000	0x3FF5_5FFF	4 KB
	RMT	0x3FF5_6000	0x3FF5_6FFF	4 KB
	PCNT	0x3FF5_7000	0x3FF5_7FFF	4 KB
	SDIO Slave	0x3FF5_8000	0x3FF5_8FFF	4 KB
	LED PWM	0x3FF5_9000	0x3FF5_9FFF	4 KB
	eFuse Controller	0x3FF5_A000	0x3FF5_AFFF	4 KB
	Flash Encryption	0x3FF5_B000	0x3FF5_BFFF	4 KB
	PWM0	0x3FF5_E000	0x3FF5_EFFF	4 KB
	TIMG0	0x3FF5_F000	0x3FF5_FFFF	4 KB
	TIMG1	0x3FF6_0000	0x3FF6_0FFF	4 KB
	SPI2	0x3FF6_4000	0x3FF6_4FFF	4 KB
	SPI3	0x3FF6_5000	0x3FF6_5FFF	4 KB



Category	Target	Start Address	End Address	Size
Peripheral	SYSCON	0x3FF6_6000	0x3FF6_6FFF	4 KB
	I2C1	0x3FF6_7000	0x3FF6_7FFF	4 KB
	SDMMC	0x3FF6_8000	0x3FF6_8FFF	4 KB
	EMAC	0x3FF6_9000	0x3FF6_AFFF	8 KB
	TWAI	0x3FF6_B000	0x3FF6_BFFF	4 KB
	PWM1	0x3FF6_C000	0x3FF6_CFFF	4 KB
	I2S1	0x3FF6_D000	0x3FF6_DFFF	4 KB
	UART2	0x3FF6_E000	0x3FF6_EFFF	4 KB
	PWM2	0x3FF6_F000	0x3FF6_FFFF	4 KB
	PWM3	0x3FF7_0000	0x3FF7_0FFF	4 KB
	RNG	0x3FF7_5000	0x3FF7_5FFF	4 KB

## 3.2 Timers and Watchdogs

### 3.2.1 64-bit Timers

There are four general-purpose timers embedded in the chip. They are all 64-bit generic timers which are based on 16-bit prescalers and 64-bit auto-reload-capable up/down-timers.

The timers feature:

- A 16-bit clock prescaler, from 2 to 65536
- A 64-bit timer
- Configurable up/down timer: incrementing or decrementing
- Halt and resume of time-base counter
- Auto-reload at alarming
- Software-controlled instant reload
- Level and edge interrupt generation

### 3.2.2 Watchdog Timers

The chip has three watchdog timers: one in each of the two timer modules (called the Main Watchdog Timer, or MWDT) and one in the RTC module (called the RTC Watchdog Timer, or RWDT). These watchdog timers are intended to recover from an unforeseen fault causing the application program to abandon its normal sequence. A watchdog timer has four stages. Each stage may trigger one of three or four possible actions upon the expiry of its programmed time period, unless the watchdog is fed or disabled. The actions are: interrupt, CPU reset, core reset, and system reset. Only the RWDT can trigger the system reset, and is able to reset the entire chip, including the RTC itself. A timeout value can be set for each stage individually.

During flash boot the RWDT and the first MWDT start automatically in order to detect, and recover from, booting problems.

The watchdogs have the following features:

- Four stages, each of which can be configured or disabled separately
- A programmable time period for each stage

- One of three or four possible actions (interrupt, CPU reset, core reset, and system reset) upon the expiry of each stage
- 32-bit expiry counter
- Write protection that prevents the RWDT and MWDT configuration from being inadvertently altered
- SPI flash boot protection  
If the boot process from an SPI flash does not complete within a predetermined time period, the watchdog will reboot the entire system.

## 3.3 System Clocks

### 3.3.1 CPU Clock

Upon reset, an external crystal clock source is selected as the default CPU clock. The external crystal clock source also connects to a PLL to generate a high-frequency clock (typically 160 MHz).

In addition, ESP32 has an internal 8 MHz oscillator. The application can select the clock source from the external crystal clock source, the PLL clock or the internal 8 MHz oscillator. The selected clock source drives the CPU clock directly, or after division, depending on the application.

### 3.3.2 RTC Clock

The RTC clock has five possible sources:

- external low-speed (32 kHz) crystal clock
- external crystal clock divided by 4
- internal RC oscillator (typically about 150 kHz, and adjustable)
- internal 8 MHz oscillator
- internal 31.25 kHz clock (derived from the internal 8 MHz oscillator divided by 256)

When the chip is in the normal power mode and needs faster CPU accessing, the application can choose the external high-speed crystal clock divided by 4 or the internal 8 MHz oscillator. When the chip operates in the low-power mode, the application chooses the external low-speed (32 kHz) crystal clock, the internal RC clock or the internal 31.25 kHz clock.

### 3.3.3 Audio PLL Clock

The audio clock is generated by the ultra-low-noise fractional-N PLL. More details can be found in Chapter Reset and Clock in the [ESP32 Technical Reference Manual](#).

## 3.4 Radio

The radio module consists of the following blocks:

- 2.4 GHz receiver
- 2.4 GHz transmitter
- bias and regulators

- balun and transmit-receive switch
- clock generator

### 3.4.1 2.4 GHz Receiver

The 2.4 GHz receiver demodulates the 2.4 GHz RF signal to quadrature baseband signals and converts them to the digital domain with two high-resolution, high-speed ADCs. To adapt to varying signal channel conditions, RF filters, Automatic Gain Control (AGC), DC offset cancelation circuits and baseband filters are integrated in the chip.

### 3.4.2 2.4 GHz Transmitter

The 2.4 GHz transmitter modulates the quadrature baseband signals to the 2.4 GHz RF signal, and drives the antenna with a high-powered Complementary Metal Oxide Semiconductor (CMOS) power amplifier. The use of digital calibration further improves the linearity of the power amplifier, enabling state-of-the-art performance in delivering up to +20.5 dBm of power for an 802.11b transmission and +18 dBm for an 802.11n transmission.

Additional calibrations are integrated to cancel any radio imperfections, such as:

- Carrier leakage
- I/Q phase matching
- Baseband nonlinearities
- RF nonlinearities
- Antenna matching

These built-in calibration routines reduce the amount of time required for product testing, and render the testing equipment unnecessary.

### 3.4.3 Clock Generator

The clock generator produces quadrature clock signals of 2.4 GHz for both the receiver and the transmitter. All components of the clock generator are integrated into the chip, including all inductors, varactors, filters, regulators and dividers.

The clock generator has built-in calibration and self-test circuits. Quadrature clock phases and phase noise are optimized on-chip with patented calibration algorithms which ensure the best performance of the receiver and the transmitter.

## 3.5 Wi-Fi

ESP32 implements a TCP/IP and full 802.11 b/g/n Wi-Fi MAC protocol. It supports the Basic Service Set (BSS) STA and SoftAP operations under the Distributed Control Function (DCF). Power management is handled with minimal host interaction to minimize the active-duty period.

### 3.5.1 Wi-Fi Radio and Baseband

The ESP32 Wi-Fi Radio and Baseband support the following features:

- 802.11 b/g/n
- 802.11 n MCS0-7 in both 20 MHz and 40 MHz bandwidth
- 802.11 n MCS32 (RX)
- 802.11 n 0.4  $\mu$ s guard-interval
- up to 150 Mbps of data rate
- Receiving STBC 2x1
- Up to 20.5 dBm of transmitting power
- Adjustable transmitting power
- Antenna diversity

ESP32 supports antenna diversity with an external RF switch. One or more GPIOs control the RF switch and selects the best antenna to minimize the effects of channel fading.

### 3.5.2 Wi-Fi MAC

The ESP32 Wi-Fi MAC applies low-level protocol functions automatically. They are as follows:

- 4  $\times$  virtual Wi-Fi interfaces
- Simultaneous Infrastructure BSS Station mode/SoftAP mode/Promiscuous mode
- RTS protection, CTS protection, Immediate Block ACK
- Defragmentation
- TX/RX A-MPDU, RX A-MSDU
- TXOP
- WMM
- CCMP (CBC-MAC, counter mode), TKIP (MIC, RC4), WAPI (SMS4), WEP (RC4) and CRC
- Automatic beacon monitoring (hardware TSF)

## 3.6 Bluetooth

The chip integrates a Bluetooth link controller and Bluetooth baseband, which carry out the baseband protocols and other low-level link routines, such as modulation/demodulation, packet processing, bit stream processing, frequency hopping, etc.

### 3.6.1 Bluetooth Radio and Baseband

The Bluetooth Radio and Baseband support the following features:

- Class-1, class-2 and class-3 transmit output powers, and a dynamic control range of up to 21 dB
- $\pi/4$  DQPSK and 8 DPSK modulation
- High performance in NZIF receiver sensitivity with a minimum sensitivity of -94 dBm
- Class-1 operation without external PA
- Internal SRAM allows full-speed data-transfer, mixed voice and data, and full piconet operation

- Logic for forward error correction, header error control, access code correlation, CRC, demodulation, encryption bit stream generation, whitening and transmit pulse shaping
- ACL, SCO, eSCO, and AFH
- A-law,  $\mu$ -law, and CVSD digital audio CODEC in PCM interface
- SBC audio CODEC
- Power management for low-power applications
- SMP with 128-bit AES

### 3.6.2 Bluetooth Interface

- Provides UART HCI interface, up to 4 Mbps
- Provides SDIO/SPI HCI interface
- Provides PCM/I2S audio interface

### 3.6.3 Bluetooth Stack

The Bluetooth stack of the chip is compliant with the Bluetooth v4.2 BR/EDR and Bluetooth LE specifications.

### 3.6.4 Bluetooth Link Controller

The link controller operates in three major states: standby, connection and sniff. It enables multiple connections, and other operations, such as inquiry, page, and secure simple-pairing, and therefore enables Piconet and Scatternet. Below are the features:

- Classic Bluetooth
  - Device Discovery (inquiry, and inquiry scan)
  - Connection establishment (page, and page scan)
  - Multi-connections
  - Asynchronous data reception and transmission
  - Synchronous links (SCO/eSCO)
  - Master/Slave Switch
  - Adaptive Frequency Hopping and Channel assessment
  - Broadcast encryption
  - Authentication and encryption
  - Secure Simple-Pairing
  - Multi-point and scatternet management
  - Sniff mode
  - Connectionless Slave Broadcast (transmitter and receiver)
  - Enhanced power control

- Ping
- Bluetooth Low Energy
  - Advertising
  - Scanning
  - Simultaneous advertising and scanning
  - Multiple connections
  - Asynchronous data reception and transmission
  - Adaptive Frequency Hopping and Channel assessment
  - Connection parameter update
  - Data Length Extension
  - Link Layer Encryption
  - LE Ping

### 3.7 RTC and Low-Power Management

With the use of advanced power-management technologies, ESP32 can switch between different power modes.

- Power modes
  - **Active mode:** The chip radio is powered on. The chip can receive, transmit, or listen.
  - **Modem-sleep mode:** The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth baseband and radio are disabled.
  - **Light-sleep mode:** The CPU is paused. The RTC memory and RTC peripherals, as well as the ULP coprocessor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
  - **Deep-sleep mode:** Only the RTC memory and RTC peripherals are powered on. Wi-Fi and Bluetooth connection data are stored in the RTC memory. The ULP coprocessor is functional.
  - **Hibernation mode:** The internal 8 MHz oscillator and ULP coprocessor are disabled. The RTC recovery memory is powered down. Only one RTC timer on the slow clock and certain RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

**Table 6: Power Consumption by Power Modes**

Power mode	Description		Power consumption
Active (RF working)	Wi-Fi Tx packet		Please refer to Table 15 for details.
	Wi-Fi/BT Tx packet		
	Wi-Fi/BT Rx and listening		
Modem-sleep	The CPU is powered on.	240 MHz *	Dual-core chip(s) 30 mA ~ 68 mA
			Single-core chip(s) N/A
		160 MHz *	Dual-core chip(s) 27 mA ~ 44 mA
			Single-core chip(s) 27 mA ~ 34 mA

Power mode	Description		Power consumption
	Normal speed: 80 MHz	Dual-core chip(s)	20 mA ~ 31 mA
		Single-core chip(s)	20 mA ~ 25 mA
Light-sleep	-		0.8 mA
Deep-sleep	The ULP coprocessor is powered on.		150 $\mu$ A
	ULP sensor-monitored pattern		100 $\mu$ A @1% duty
	RTC timer + RTC memory		10 $\mu$ A
Hibernation	RTC timer only		5 $\mu$ A
Power off	CHIP_PU is set to low level, the chip is powered off.		1 $\mu$ A

**Note:**

- \* Among the ESP32 series of SoCs, ESP32-D0WD-V3, ESP32-U4WDH, ESP32-D0WD ([NRND](#)), ESP32-D0WDQ6 ([NRND](#)), and ESP32-D0WDQ6-V3 ([NRND](#)) have a maximum CPU frequency of 240 MHz, ESP32-S0WD has a maximum CPU frequency of 160 MHz.
- When Wi-Fi is enabled, the chip switches between Active and Modem-sleep modes. Therefore, power consumption changes accordingly.
- In Modem-sleep mode, the CPU frequency changes automatically. The frequency depends on the CPU load and the peripherals used.
- During Deep-sleep, when the ULP coprocessor is powered on, peripherals such as GPIO and RTC I2C are able to operate.
- When the system works in the ULP sensor-monitored pattern, the ULP coprocessor works with the ULP sensor periodically and the ADC works with a duty cycle of 1%, so the power consumption is 100  $\mu$ A.

## 4 Peripherals and Sensors

### 4.1 Descriptions of Peripherals and Sensors

#### 4.1.1 General Purpose Input/Output Interface (GPIO)

ESP32 has 34 GPIO pins which can be assigned various functions by programming the appropriate registers. There are several kinds of GPIOs: digital-only, analog-enabled, capacitive-touch-enabled, etc. Analog-enabled GPIOs and Capacitive-touch-enabled GPIOs can be configured as digital GPIOs.

Most of the digital GPIOs can be configured as internal pull-up or pull-down, or set to high impedance. When configured as an input, the input value can be read through the register. The input can also be set to edge-trigger or level-trigger to generate CPU interrupts. Most of the digital IO pins are bi-directional, non-inverting and tristate, including input and output buffers with tristate control. These pins can be multiplexed with other functions, such as the SDIO, UART, SPI, etc. (More details can be found in the Appendix, Table [IO\\_MUX](#).) For low-power operations, the GPIOs can be set to hold their states.

#### 4.1.2 Analog-to-Digital Converter (ADC)

ESP32 integrates 12-bit SAR ADCs and supports measurements on 18 channels (analog-enabled pins). The ULP coprocessor in ESP32 is also designed to measure voltage, while operating in the sleep mode, which enables low-power consumption. The CPU can be woken up by a threshold setting and/or via other triggers.

With appropriate settings, the ADCs can be configured to measure voltage on 18 pins maximum.

Table 7 describes the ADC characteristics.

**Table 7: ADC Characteristics**

Parameter	Description	Min	Max	Unit
DNL (Differential nonlinearity)	RTC controller; ADC connected to an external 100 nF capacitor; DC signal input; ambient temperature at 25 °C; Wi-Fi&Bluetooth off	-7	7	LSB
INL (Integral nonlinearity)		-12	12	LSB
Sampling rate	RTC controller	-	200	ksps
	DIG controller	-	2	Msps

#### Notes:

- When atten = 3 and the measurement result is above 3000 (voltage at approx. 2450 mV), the ADC accuracy will be worse than described in the table above.
- To get better DNL results, users can take multiple sampling tests with a filter, or calculate the average value.
- The input voltage range of GPIO pins within VDD3P3\_RTC domain should strictly follow the DC characteristics provided in Table 13. Otherwise, measurement errors may be introduced, and chip performance may be affected.

By default, there are  $\pm 6\%$  differences in measured results between chips. ESP-IDF provides couple of [calibration methods](#) for ADC1. Results after calibration using eFuse Vref value are shown in Table 8. For higher accuracy, users may apply other calibration methods provided in ESP-IDF, or implement their own.



**Table 8: ADC Calibration Results**

Parameter	Description	Min	Max	Unit
Total error	Atten = 0, effective measurement range of 100 ~ 950 mV	-23	23	mV
	Atten = 1, effective measurement range of 100 ~ 1250 mV	-30	30	mV
	Atten = 2, effective measurement range of 150 ~ 1750 mV	-40	40	mV
	Atten = 3, effective measurement range of 150 ~ 2450 mV	-60	60	mV

### 4.1.3 Hall Sensor

ESP32 integrates a Hall sensor based on an N-carrier resistor. When the chip is in the magnetic field, the Hall sensor develops a small voltage laterally on the resistor, which can be directly measured by the ADC.

### 4.1.4 Digital-to-Analog Converter (DAC)

Two 8-bit DAC channels can be used to convert two digital signals into two analog voltage signal outputs. The design structure is composed of integrated resistor strings and a buffer. This dual DAC supports power supply as input voltage reference. The two DAC channels can also support independent conversions.

### 4.1.5 Touch Sensor

ESP32 has 10 capacitive-sensing GPIOs, which detect variations induced by touching or approaching the GPIOs with a finger or other objects. The low-noise nature of the design and the high sensitivity of the circuit allow relatively small pads to be used. Arrays of pads can also be used, so that a larger area or more points can be detected. The 10 capacitive-sensing GPIOs are listed in Table 9.

**Table 9: Capacitive-Sensing GPIOs Available on ESP32**

Capacitive-sensing signal name	Pin name
T0	GPIO4
T1	GPIO0
T2	GPIO2
T3	MTDO
T4	MTCK
T5	MTDI
T6	MTMS
T7	GPIO27
T8	32K_XN
T9	32K_XP

### 4.1.6 Ultra-Low-Power (ULP) Coprocessor

The ULP coprocessor and RTC memory remain powered on during the Deep-sleep mode. Hence, the developer can store a program for the ULP coprocessor in the RTC slow memory to access the peripheral devices, internal timers and internal sensors during the Deep-sleep mode. This is useful for designing applications where the CPU needs to be woken up by an external event, or a timer, or a combination of the two, while maintaining minimal power consumption.

### 4.1.7 Ethernet MAC Interface

An IEEE-802.3-2008-compliant Media Access Controller (MAC) is provided for Ethernet LAN communications. ESP32 requires an external physical interface device (PHY) to connect to the physical LAN bus (twisted-pair, fiber, etc.). The PHY is connected to ESP32 through 17 signals of MII or nine signals of RMI. The following features are supported on the Ethernet MAC (EMAC) interface:

- 10 Mbps and 100 Mbps rates
- Dedicated DMA controller allowing high-speed transfer between the dedicated SRAM and Ethernet MAC
- Tagged MAC frame (VLAN support)
- Half-duplex (CSMA/CD) and full-duplex operation
- MAC control sublayer (control frames)
- 32-bit CRC generation and removal
- Several address-filtering modes for physical and multicast address (multicast and group addresses)
- 32-bit status code for each transmitted or received frame
- Internal FIFOs to buffer transmit and receive frames. The transmit FIFO and the receive FIFO are both 512 words (32-bit)
- Hardware PTP (Precision Time Protocol) in accordance with IEEE 1588 2008 (PTP V2)
- 25 MHz/50 MHz clock output

### 4.1.8 SD/SDIO/MMC Host Controller

An SD/SDIO/MMC host controller is available on ESP32, which supports the following features:

- Secure Digital memory (SD mem Version 3.0 and Version 3.01)
- Secure Digital I/O (SDIO Version 3.0)
- Consumer Electronics Advanced Transport Architecture (CE-ATA Version 1.1)
- Multimedia Cards (MMC Version 4.41, eMMC Version 4.5 and Version 4.51)

The controller allows up to 80 MHz clock output in three different data-bus modes: 1-bit, 4-bit, and 8-bit. It supports two SD/SDIO/MMC4.41 cards in a 4-bit data-bus mode. It also supports one SD card operating at 1.8 V.

### 4.1.9 SDIO/SPI Slave Controller

ESP32 integrates an SD device interface that conforms to the industry-standard SDIO Card Specification Version 2.0, and allows a host controller to access the SoC, using the SDIO bus interface and protocol. ESP32 acts as the slave on the SDIO bus. The host can access the SDIO-interface registers directly and can access shared memory via a DMA engine, thus maximizing performance without engaging the processor cores.

The SDIO/SPI slave controller supports the following features:

- SPI, 1-bit SDIO, and 4-bit SDIO transfer modes over the full clock range from 0 to 50 MHz
- Configurable sampling and driving clock edge
- Special registers for direct access by host

- Interrupts to host for initiating data transfer
- Automatic loading of SDIO bus data and automatic discarding of padding data
- Block size of up to 512 bytes
- Interrupt vectors between the host and the slave, allowing both to interrupt each other
- Supports DMA for data transfer

#### 4.1.10 Universal Asynchronous Receiver Transmitter (UART)

ESP32 has three UART interfaces, i.e., UART0, UART1, and UART2, which provide asynchronous communication (RS232 and RS485) and IrDA support, communicating at a speed of up to 5 Mbps. UART provides hardware management of the CTS and RTS signals and software flow control (XON and XOFF). All of the interfaces can be accessed by the DMA controller or directly by the CPU.

#### 4.1.11 I2C Interface

ESP32 has two I2C bus interfaces which can serve as I2C master or slave, depending on the user's configuration. The I2C interfaces support:

- Standard mode (100 Kbit/s)
- Fast mode (400 Kbit/s)
- Up to 5 MHz, yet constrained by SDA pull-up strength
- 7-bit/10-bit addressing mode
- Dual addressing mode

Users can program command registers to control I2C interfaces, so that they have more flexibility.

#### 4.1.12 I2S Interface

Two standard I2S interfaces are available in ESP32. They can be operated in master or slave mode, in full duplex and half-duplex communication modes, and can be configured to operate with an 8-/16-/32-/48-/64-bit resolution as input or output channels. BCK clock frequency, from 10 kHz up to 40 MHz, is supported. When one or both of the I2S interfaces are configured in the master mode, the master clock can be output to the external DAC/CODEC.

Both of the I2S interfaces have dedicated DMA controllers. PDM and BT PCM interfaces are supported.

#### 4.1.13 Infrared Remote Controller (RMT)

The infrared remote controller supports eight channels of infrared remote transmission and receiving. By programming the pulse waveform, it supports various infrared protocols. Eight channels share a 512 x 32-bit block of memory to store the transmitting or receiving waveform.

#### 4.1.14 Pulse Counter

The pulse counter captures pulse and counts pulse edges through seven modes. It has eight channels, each of which captures four signals at a time. The four input signals include two pulse signals and two control signals. When the counter reaches a defined threshold, an interrupt is generated.

### 4.1.15 Pulse Width Modulation (PWM)

The Pulse Width Modulation (PWM) controller can be used for driving digital motors and smart lights. The controller consists of PWM timers, the PWM operator and a dedicated capture sub-module. Each timer provides timing in synchronous or independent form, and each PWM operator generates a waveform for one PWM channel. The dedicated capture sub-module can accurately capture events with external timing.

### 4.1.16 LED PWM

The LED PWM controller can generate 16 independent channels of digital waveforms with configurable periods and duties.

The 16 channels of digital waveforms operate with an APB clock of 80 MHz. Eight of these channels have the option of using the 8 MHz oscillator clock. Each channel can select a 20-bit timer with configurable counting range, while its accuracy of duty can be up to 16 bits within a 1 ms period.

The software can change the duty immediately. Moreover, each channel automatically supports step-by-step duty increase or decrease, which is useful for the LED RGB color-gradient generator.

### 4.1.17 Serial Peripheral Interface (SPI)

ESP32 features three SPIs (SPI, HSPI and VSPI) in slave and master modes in 1-line full-duplex and 1/2/4-line half-duplex communication modes. These SPIs also support the following general-purpose SPI features:

- Four modes of SPI transfer format, which depend on the polarity (CPOL) and the phase (CPHA) of the SPI clock
- Up to 80 MHz (The actual speed it can reach depends on the selected pads, PCB tracing, peripheral characteristics, etc.)
- up to 64-byte FIFO

All SPIs can also be connected to the external flash/SRAM and LCD. Each SPI can be served by DMA controllers.

### 4.1.18 TWAI Controller

ESP32 family has a TWAI<sup>®</sup> controller with the following features:

- compatible with ISO 11898-1 protocol (CAN Specification 2.0)
- standard frame format (11-bit ID) and extended frame format (29-bit ID)
- bit rates from 1 Kbit/s to 1 Mbit/s
- multiple modes of operation: Normal, Listen Only, and Self-Test
- 64-byte receive FIFO
- special transmissions: single-shot transmissions and self reception
- acceptance filter (single and dual filter modes)
- error detection and handling: error counters, configurable error interrupt threshold, error code capture, arbitration lost capture

### 4.1.19 Accelerator

ESP32 is equipped with hardware accelerators of general algorithms, such as AES (FIPS PUB 197), SHA (FIPS PUB 180-4), RSA, and ECC, which support independent arithmetic, such as Big Integer Multiplication and Big Integer Modular Multiplication. The maximum operation length for RSA, ECC, Big Integer Multiply and Big Integer Modular Multiplication is 4096 bits.

The hardware accelerators greatly improve operation speed and reduce software complexity. They also support code encryption and dynamic decryption, which ensures that code in the flash will not be hacked.

## 4.2 Peripheral Pin Configurations

**Table 10: Peripheral Pin Configurations**

Interface	Signal	Pin	Function
ADC	ADC1_CH0	SENSOR_VP	Two 12-bit SAR ADCs
	ADC1_CH1	SENSOR_CAPP	
	ADC1_CH2	SENSOR_CAPN	
	ADC1_CH3	SENSOR_VN	
	ADC1_CH4	32K_XP	
	ADC1_CH5	32K_XN	
	ADC1_CH6	VDET_1	
	ADC1_CH7	VDET_2	
	ADC2_CH0	GPIO4	
	ADC2_CH1	GPIO0	
	ADC2_CH2	GPIO2	
	ADC2_CH3	MTDO	
	ADC2_CH4	MTCK	
	ADC2_CH5	MTDI	
	ADC2_CH6	MTMS	
	ADC2_CH7	GPIO27	
	ADC2_CH8	GPIO25	
ADC2_CH9	GPIO26		
DAC	DAC_1	GPIO25	Two 8-bit DACs
	DAC_2	GPIO26	
Touch Sensor	TOUCH0	GPIO4	Capacitive touch sensors
	TOUCH1	GPIO0	
	TOUCH2	GPIO2	
	TOUCH3	MTDO	
	TOUCH4	MTCK	
	TOUCH5	MTDI	
	TOUCH6	MTMS	
	TOUCH7	GPIO27	
	TOUCH8	32K_XN	
	TOUCH9	32K_XP	

Interface	Signal	Pin	Function
JTAG	MTDI	MTDI	JTAG for software debugging
	MTCK	MTCK	
	MTMS	MTMS	
	MTDO	MTDO	
SD/SDIO/MMC Host Controller	HS2_CLK	MTMS	Supports SD memory card V3.01 standard
	HS2_CMD	MTDO	
	HS2_DATA0	GPIO2	
	HS2_DATA1	GPIO4	
	HS2_DATA2	MTDI	
Motor PWM	HS2_DATA3	MTCK	Three channels of 16-bit timers generate PWM waveforms. Each channel has a pair of output signals, three fault detection signals, three event-capture signals, and three sync signals.
	PWM0_OUT0~2	Any GPIO Pins	
	PWM1_OUT_IN0~2		
	PWM0_FLT_IN0~2		
	PWM1_FLT_IN0~2		
	PWM0_CAP_IN0~2		
	PWM1_CAP_IN0~2		
	PWM0_SYNC_IN0~2		
PWM1_SYNC_IN0~2			
SDIO/SPI Slave Controller	SD_CLK	MTMS	SDIO interface that conforms to the industry standard SDIO 2.0 card specification
	SD_CMD	MTDO	
	SD_DATA0	GPIO2	
	SD_DATA1	GPIO4	
	SD_DATA2	MTDI	
	SD_DATA3	MTCK	
UART	U0RXD_in	Any GPIO Pins	Three UART devices with hardware flow-control and DMA
	U0CTS_in		
	U0DSR_in		
	U0TXD_out		
	U0RTS_out		
	U0DTR_out		
	U1RXD_in		
	U1CTS_in		
	U1TXD_out		
	U1RTS_out		
	U2RXD_in		
	U2CTS_in		
	U2TXD_out		
	U2RTS_out		
I2C	I2CEXT0_SCL_in	Any GPIO Pins	Two I2C devices in slave or master mode
	I2CEXT0_SDA_in		
	I2CEXT1_SCL_in		
	I2CEXT1_SDA_in		
	I2CEXT0_SCL_out		
	I2CEXT0_SDA_out		
	I2CEXT1_SCL_out		

Interface	Signal	Pin	Function
	I2CEXT1_SDA_out		
LED PWM	ledc_hs_sig_out0~7	Any GPIO Pins	16 independent channels @80 MHz clock/RTC CLK. Duty accuracy: 16 bits.
	ledc_ls_sig_out0~7		
I2S	I2S0I_DATA_in0~15	Any GPIO Pins	Stereo input and output from/to the audio codec; parallel LCD data output; parallel camera data input
	I2S0O_BCK_in		
	I2S0O_WS_in		
	I2S0I_BCK_in		
	I2S0I_WS_in		
	I2S0I_H_SYNC		
	I2S0I_V_SYNC		
	I2S0I_H_ENABLE		
	I2S0O_BCK_out		
	I2S0O_WS_out		
	I2S0I_BCK_out		
	I2S0I_WS_out		
	I2S0O_DATA_out0~23		
	I2S1I_DATA_in0~15		
	I2S1O_BCK_in		
	I2S1O_WS_in		
	I2S1I_BCK_in		
	I2S1I_WS_in		
	I2S1I_H_SYNC		
	I2S1I_V_SYNC		
	I2S1I_H_ENABLE		
	I2S1O_BCK_out		
I2S1O_WS_out			
I2S1I_BCK_out			
I2S1I_WS_out			
I2S1O_DATA_out0~23			
RMT	RMT_SIG_IN0~7	Any GPIO Pins	Eight channels for an IR transmitter and receiver of various waveforms
	RMT_SIG_OUT0~7		
General Purpose SPI	HSPIQ_in/_out	Any GPIO Pins	Standard SPI consists of clock, chip-select, MOSI and MISO. These SPIs can be connected to LCD and other external devices. They support the following features: <ul style="list-style-type: none"> <li>• Both master and slave modes;</li> <li>• Four sub-modes of the SPI transfer format;</li> <li>• Configurable SPI frequency;</li> <li>• Up to 64 bytes of FIFO and DMA.</li> </ul>
	HSPID_in/_out		
	HSPICLK_in/_out		
	HSPI_CS0_in/_out		
	HSPI_CS1_out		
	HSPI_CS2_out		
	VSPIQ_in/_out		
	VSPID_in/_out		
	VSPICLK_in/_out		
	VSPI_CS0_in/_out		
	VSPI_CS1_out		
	VSPI_CS2_out		

Interface	Signal	Pin	Function
Parallel QSPI	SPIHD	SD_DATA_2	Supports Standard SPI, Dual SPI, and Quad SPI that can be connected to the external flash and SRAM
	SPIWP	SD_DATA_3	
	SPICS0	SD_CMD	
	SPICLK	SD_CLK	
	SPIQ	SD_DATA_0	
	SPID	SD_DATA_1	
	HSPICLK	MTMS	
	HSPICS0	MTDO	
	HSPIQ	MTDI	
	HSPID	MTCK	
	HSPiHD	GPIO4	
	HSPiWP	GPIO2	
	VSPICLK	GPIO18	
	VSPICS0	GPIO5	
	VSPiQ	GPIO19	
	VSPID	GPIO23	
	VSPiHD	GPIO21	
VSPiWP	GPIO22		
EMAC	EMAC_TX_CLK	GPIO0	Ethernet MAC with MII/RMII interface
	EMAC_RX_CLK	GPIO5	
	EMAC_TX_EN	GPIO21	
	EMAC_TXD0	GPIO19	
	EMAC_TXD1	GPIO22	
	EMAC_TXD2	MTMS	
	EMAC_TXD3	MTDI	
	EMAC_RX_ER	MTCK	
	EMAC_RX_DV	GPIO27	
	EMAC_RXD0	GPIO25	
	EMAC_RXD1	GPIO26	
	EMAC_RXD2	U0TXD	
	EMAC_RXD3	MTDO	
	EMAC_CLK_OUT	GPIO16	
	EMAC_CLK_OUT_180	GPIO17	
	EMAC_TX_ER	GPIO4	
	EMAC_MDC_out	Any GPIO Pins	
	EMAC_MDI_in	Any GPIO Pins	
	EMAC_MDO_out	Any GPIO Pins	
	EMAC_CRS_out	Any GPIO Pins	
EMAC_COL_out	Any GPIO Pins		



Interface	Signal	Pin	Function
Pulse Counter	pcnt_sig_ch0_in0	Any GPIO Pins	Operating in seven different modes, the pulse counter captures pulse and counts pulse edges.
	pcnt_sig_ch1_in0		
	pcnt_ctrl_ch0_in0		
	pcnt_ctrl_ch1_in0		
	pcnt_sig_ch0_in1		
	pcnt_sig_ch1_in1		
	pcnt_ctrl_ch0_in1		
	pcnt_ctrl_ch1_in1		
	pcnt_sig_ch0_in2		
	pcnt_sig_ch1_in2		
	pcnt_ctrl_ch0_in2		
	pcnt_ctrl_ch1_in2		
	pcnt_sig_ch0_in3		
	pcnt_sig_ch1_in3		
	pcnt_ctrl_ch0_in3		
	pcnt_ctrl_ch1_in3		
	pcnt_sig_ch0_in4		
	pcnt_sig_ch1_in4		
	pcnt_ctrl_ch0_in4		
	pcnt_ctrl_ch1_in4		
	pcnt_sig_ch0_in5		
	pcnt_sig_ch1_in5		
	pcnt_ctrl_ch0_in5		
	pcnt_ctrl_ch1_in5		
	pcnt_sig_ch0_in6		
	pcnt_sig_ch1_in6		
	pcnt_ctrl_ch0_in6		
	pcnt_ctrl_ch1_in6		
pcnt_sig_ch0_in7			
pcnt_sig_ch1_in7			
pcnt_ctrl_ch0_in7			
pcnt_ctrl_ch1_in7			
TWAI	twai_rx	Any GPIO Pins	Compatible with ISO 11898-1 protocol (CAN Specification 2.0)
	twai_tx		
	twai_bus_off_on		
	twai_clkout		

## 5 Electrical Characteristics

### 5.1 Absolute Maximum Ratings

Stresses beyond the absolute maximum ratings listed in the table below may cause permanent damage to the device. These are stress ratings only, and do not refer to the functional operation of the device that should follow the [recommended operating conditions](#).

**Table 11: Absolute Maximum Ratings**

Symbol	Parameter	Min	Max	Unit
VDDA, VDD3P3, VDD3P3_RTC, VDD3P3_CPU, VDD_SDIO	Voltage applied to power supply pins per power domain	-0.3	3.6	V
$I_{output}^*$	Cumulative IO output current	-	1200	mA
$T_{store}$	Storage temperature	-40	150	°C

\* The chip worked properly after a 24-hour test in ambient temperature at 25 °C, and the IOs in three domains (VDD3P3\_RTC, VDD3P3\_CPU, VDD\_SDIO) output high logic level to ground.

### 5.2 Recommended Operating Conditions

**Table 12: Recommended Operating Conditions**

Symbol	Parameter	Min	Typ	Max	Unit
VDDA, VDD3P3_RTC, <sup>note 1</sup> VDD3P3, VDD_SDIO (3.3 V mode) <sup>note 2</sup>	Voltage applied to power supply pins per power domain	2.3/3.0 <sup>note 3</sup>	3.3	3.6	V
VDD3P3_CPU	Voltage applied to power supply pin	1.8	3.3	3.6	V
$I_{VDD}$	Current delivered by external power supply	0.5	-	-	A
$T$ <sup>note 4</sup>	Operating temperature	-40	-	125	°C

- When writing eFuse, VDD3P3\_RTC should be at least 3.3 V.
- VDD\_SDIO works as the power supply for the related IO, and also for an external device. Please refer to the Appendix [IO\\_MUX](#) of this datasheet for more details.
  - VDD\_SDIO can be sourced internally by the ESP32 from the VDD3P3\_RTC power domain:
    - When VDD\_SDIO operates at 3.3 V, it is driven directly by VDD3P3\_RTC through a 6 Ω resistor, therefore, there will be some voltage drop from VDD3P3\_RTC.
    - When VDD\_SDIO operates at 1.8 V, it can be generated from ESP32's internal LDO. The maximum current this LDO can offer is 40 mA, and the output voltage range is 1.65 V ~ 2.0 V.
  - VDD\_SDIO can also be driven by an external power supply.
  - Please refer to Power Scheme, section [2.3](#), for more information.
- Chips with a 3.3 V flash embedded: this minimum voltage is 3.0 V;
  - Chips with no flash: this minimum voltage is 2.3 V;
  - For more information, see Table [23 ESP32 Ordering Information](#).
- The operating temperature of ESP32-U4WDH ranges from -40 °C to 105 °C, due to the flash embedded in it. The other chips in this series have no embedded flash, so their range of operating temperatures is -40 °C ~ 125 °C.

## 5.3 DC Characteristics (3.3 V, 25 °C)

Table 13: DC Characteristics (3.3 V, 25 °C)

Symbol	Parameter	Min	Typ	Max	Unit	
$C_{IN}$	Pin capacitance	-	2	-	pF	
$V_{IH}$	High-level input voltage	$0.75 \times VDD^1$	-	$VDD^1 + 0.3$	V	
$V_{IL}$	Low-level input voltage	-0.3	-	$0.25 \times VDD^1$	V	
$I_{IH}$	High-level input current	-	-	50	nA	
$I_{IL}$	Low-level input current	-	-	50	nA	
$V_{OH}$	High-level output voltage	$0.8 \times VDD^1$	-	-	V	
$V_{OL}$	Low-level output voltage	-	-	$0.1 \times VDD^1$	V	
$I_{OH}$	High-level source current ( $VDD^1 = 3.3$ V, $V_{OH} \geq 2.64$ V, output drive strength set to the maximum)	VDD3P3_CPU power domain <sup>1, 2</sup>	-	40	-	mA
		VDD3P3_RTC power domain <sup>1, 2</sup>	-	40	-	mA
		VDD_SDIO power domain <sup>1, 3</sup>	-	20	-	mA
$I_{OL}$	Low-level sink current ( $VDD^1 = 3.3$ V, $V_{OL} = 0.495$ V, output drive strength set to the maximum)	-	28	-	mA	
$R_{PU}$	Resistance of internal pull-up resistor	-	45	-	k $\Omega$	
$R_{PD}$	Resistance of internal pull-down resistor	-	45	-	k $\Omega$	
$V_{IL\_nRST}$	Low-level input voltage of CHIP_PU to power off the chip	-	-	0.6	V	

### Notes:

1. Please see Table IO\_MUX for IO's power domain. VDD is the I/O voltage for a particular power domain of pins.
2. For VDD3P3\_CPU and VDD3P3\_RTC power domain, per-pin current sourced in the same domain is gradually reduced from around 40 mA to around 29 mA,  $V_{OH} \geq 2.64$  V, as the number of current-source pins increases.
3. For VDD\_SDIO power domain, per-pin current sourced in the same domain is gradually reduced from around 30 mA to around 10 mA,  $V_{OH} \geq 2.64$  V, as the number of current-source pins increases.

## 5.4 Reliability Qualifications

ESP32 chip series passed all reliability qualifications listed in Table 14.

Table 14: Reliability Qualifications

Test Item	Test Condition	Test Standard
HTOL (High Temperature Operating Life)	125 °C, 1000 hours	JESD22-A108
ESD (Electro-Static Discharge Sensitivity)	HBM (Human Body Mode) <sup>1</sup> $\pm 2000$ V	JESD22-A114
	CDM (Charge Device Mode) <sup>2</sup> $\pm 500$ V	JESD22-C101F
Latch up	Current trigger $\pm 200$ mA	JESD78
	Voltage trigger $1.5 \times VDD_{max}$	
Preconditioning	Bake 24 hours @125 °C	J-STD-020,
	Moisture soak (level 3: 192 hours @30 °C, 60% RH)	JESD47,
	IR reflow solder: 260 + 0 °C, 20 seconds, three times	JESD22-A113

Test Item	Test Condition	Test Standard
TCT (Temperature Cycling Test)	-65 °C / 150 °C, 500 cycles	JESD22-A104
Autoclave Test	121 °C, 100% RH, 96 hours	JESD22-A102
uHAST (Highly Accelerated Stress Test, unbiased)	130 °C, 85% RH, 96 hours	JESD22-A118
HTSL (High Temperature Storage Life)	150 °C, 1000 hours	JESD22-A103

1. JEDEC document JEP155 states that 500 V HBM allows safe manufacturing with a standard ESD control process.
2. JEDEC document JEP157 states that 250 V CDM allows safe manufacturing with a standard ESD control process.

## 5.5 RF Power-Consumption Specifications

The power consumption measurements are taken with a 3.3 V supply at 25 °C of ambient temperature at the RF port. All transmitters' measurements are based on a 50% duty cycle.

**Table 15: RF Power-Consumption Specifications**

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	240	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	190	-	mA
Transmit 802.11n, OFDM MCS7, POUT = +14 dBm	-	180	-	mA
Receive 802.11b/g/n	-	95 ~ 100	-	mA
Transmit BT/BLE, POUT = 0 dBm	-	130	-	mA
Receive BT/BLE	-	95 ~ 100	-	mA

## 5.6 Wi-Fi Radio

**Table 16: Wi-Fi Radio Characteristics**

Parameter	Condition	Min	Typ	Max	Unit
Operating frequency range <sup>note1</sup>	-	2412	-	2484	MHz
Output impedance <sup>note2</sup>	-	-	<i>note 2</i>	-	Ω
TX power <sup>note3</sup>	11n, MCS7	12	13	14	dBm
	11b mode	18.5	19.5	20.5	dBm
Sensitivity	11b, 1 Mbps	-	-98	-	dBm
	11b, 11 Mbps	-	-88	-	dBm
	11g, 6 Mbps	-	-93	-	dBm
	11g, 54 Mbps	-	-75	-	dBm
	11n, HT20, MCS0	-	-93	-	dBm
	11n, HT20, MCS7	-	-73	-	dBm
	11n, HT40, MCS0	-	-90	-	dBm
11n, HT40, MCS7	-	-70	-	dBm	

Parameter	Condition	Min	Typ	Max	Unit
Adjacent channel rejection	11g, 6 Mbps	-	27	-	dB
	11g, 54 Mbps	-	13	-	dB
	11n, HT20, MCS0	-	27	-	dB
	11n, HT20, MCS7	-	12	-	dB

1. Device should operate in the frequency range allocated by regional regulatory authorities. Target operating frequency range is configurable by software.
2. The typical value of ESP32's Wi-Fi radio output impedance is different between chips in different QFN packages. For ESP32 chips with a QFN 6x6 package, the value is  $30+j10 \Omega$ . For ESP32 chips with a QFN 5x5 package, the value is  $35+j10 \Omega$ .
3. Target TX power is configurable based on device or certification requirements.

## 5.7 Bluetooth Radio

### 5.7.1 Receiver – Basic Data Rate

Table 17: Receiver Characteristics – Basic Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @0.1% BER	-	-90	-89	-88	dBm
Maximum received signal @0.1% BER	-	0	-	-	dBm
Co-channel C/I	-	-	+7	-	dB
Adjacent channel selectivity C/I	F = F <sub>0</sub> + 1 MHz	-	-	-6	dB
	F = F <sub>0</sub> - 1 MHz	-	-	-6	dB
	F = F <sub>0</sub> + 2 MHz	-	-	-25	dB
	F = F <sub>0</sub> - 2 MHz	-	-	-33	dB
	F = F <sub>0</sub> + 3 MHz	-	-	-25	dB
	F = F <sub>0</sub> - 3 MHz	-	-	-45	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
Intermodulation	-	-36	-	-	dBm

### 5.7.2 Transmitter – Basic Data Rate

Table 18: Transmitter Characteristics – Basic Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see <a href="#">note</a> under Table 18)	-	-	0	-	dBm
Gain control step	-	-	3	-	dB
RF power control range	-	-12	-	+9	dBm
+20 dB bandwidth	-	-	0.9	-	MHz
Adjacent channel transmit power	F = F <sub>0</sub> ± 2 MHz	-	-47	-	dBm
	F = F <sub>0</sub> ± 3 MHz	-	-55	-	dBm

Parameter	Conditions	Min	Typ	Max	Unit
	$F = F_0 \pm > 3 \text{ MHz}$	-	-60	-	dBm
$\Delta f_{1\text{avg}}$	-	-	-	155	kHz
$\Delta f_{2\text{max}}$	-	133.7	-	-	kHz
$\Delta f_{2\text{avg}}/\Delta f_{1\text{avg}}$	-	-	0.92	-	-
ICFT	-	-	-7	-	kHz
Drift rate	-	-	0.7	-	kHz/50 $\mu\text{s}$
Drift (DH1)	-	-	6	-	kHz
Drift (DH5)	-	-	6	-	kHz

**Note:**

There are a total of eight power levels from 0 to 7, and the transmit power ranges from -12 dBm to 9 dBm. When the power level rises by 1, the transmit power increases by 3 dB. Power level 4 is used by default and the corresponding transmit power is 0 dBm.

### 5.7.3 Receiver – Enhanced Data Rate

Table 19: Receiver Characteristics – Enhanced Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
$\pi/4$ DQPSK					
Sensitivity @0.01% BER	-	-90	-89	-88	dBm
Maximum received signal @0.01% BER	-	-	0	-	dBm
Co-channel C/I	-	-	11	-	dB
Adjacent channel selectivity C/I	$F = F_0 + 1 \text{ MHz}$	-	-7	-	dB
	$F = F_0 - 1 \text{ MHz}$	-	-7	-	dB
	$F = F_0 + 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 2 \text{ MHz}$	-	-35	-	dB
	$F = F_0 + 3 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 3 \text{ MHz}$	-	-45	-	dB
8DPSK					
Sensitivity @0.01% BER	-	-84	-83	-82	dBm
Maximum received signal @0.01% BER	-	-	-5	-	dBm
C/I c-channel	-	-	18	-	dB
Adjacent channel selectivity C/I	$F = F_0 + 1 \text{ MHz}$	-	2	-	dB
	$F = F_0 - 1 \text{ MHz}$	-	2	-	dB
	$F = F_0 + 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 2 \text{ MHz}$	-	-25	-	dB
	$F = F_0 + 3 \text{ MHz}$	-	-25	-	dB
	$F = F_0 - 3 \text{ MHz}$	-	-38	-	dB

### 5.7.4 Transmitter – Enhanced Data Rate

Table 20: Transmitter Characteristics – Enhanced Data Rate

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 18)	-	-	0	-	dBm
Gain control step	-	-	3	-	dB
RF power control range	-	-12	-	+9	dBm
$\pi/4$ DQPSK max w0	-	-	-0.72	-	kHz
$\pi/4$ DQPSK max wi	-	-	-6	-	kHz
$\pi/4$ DQPSK max lwi + w0l	-	-	-7.42	-	kHz
8DPSK max w0	-	-	0.7	-	kHz
8DPSK max wi	-	-	-9.6	-	kHz
8DPSK max lwi + w0l	-	-	-10	-	kHz
$\pi/4$ DQPSK modulation accuracy	RMS DEVM	-	4.28	-	%
	99% DEVM	-	100	-	%
	Peak DEVM	-	13.3	-	%
8 DPSK modulation accuracy	RMS DEVM	-	5.8	-	%
	99% DEVM	-	100	-	%
	Peak DEVM	-	14	-	%
In-band spurious emissions	F = F0 $\pm$ 1 MHz	-	-46	-	dBm
	F = F0 $\pm$ 2 MHz	-	-40	-	dBm
	F = F0 $\pm$ 3 MHz	-	-46	-	dBm
	F = F0 +/- > 3 MHz	-	-	-53	dBm
EDR differential phase coding	-	-	100	-	%

## 5.8 Bluetooth LE Radio

### 5.8.1 Receiver

Table 21: Receiver Characteristics – Bluetooth LE

Parameter	Conditions	Min	Typ	Max	Unit
Sensitivity @30.8% PER	-	-94	-93	-92	dBm
Maximum received signal @30.8% PER	-	0	-	-	dBm
Co-channel C/I	-	-	+10	-	dB
Adjacent channel selectivity C/I	F = F0 + 1 MHz	-	-5	-	dB
	F = F0 - 1 MHz	-	-5	-	dB
	F = F0 + 2 MHz	-	-25	-	dB
	F = F0 - 2 MHz	-	-35	-	dB
	F = F0 + 3 MHz	-	-25	-	dB
	F = F0 - 3 MHz	-	-45	-	dB
Out-of-band blocking performance	30 MHz ~ 2000 MHz	-10	-	-	dBm
	2000 MHz ~ 2400 MHz	-27	-	-	dBm
	2500 MHz ~ 3000 MHz	-27	-	-	dBm
	3000 MHz ~ 12.5 GHz	-10	-	-	dBm
Intermodulation	-	-36	-	-	dBm

## 5.8.2 Transmitter

**Table 22: Transmitter Characteristics – Bluetooth LE**

Parameter	Conditions	Min	Typ	Max	Unit
RF transmit power (see note under Table 18)	-	-	0	-	dBm
Gain control step	-	-	3	-	dB
RF power control range	-	-12	-	+9	dBm
Adjacent channel transmit power	$F = F_0 \pm 2 \text{ MHz}$	-	-52	-	dBm
	$F = F_0 \pm 3 \text{ MHz}$	-	-58	-	dBm
	$F = F_0 \pm > 3 \text{ MHz}$	-	-60	-	dBm
$\Delta f_{1\text{avg}}$	-	-	-	265	kHz
$\Delta f_{2\text{max}}$	-	247	-	-	kHz
$\Delta f_{2\text{avg}}/\Delta f_{1\text{avg}}$	-	-	0.92	-	-
ICFT	-	-	-10	-	kHz
Drift rate	-	-	0.7	-	kHz/50 $\mu\text{s}$
Drift	-	-	2	-	kHz



## 6 Package Information

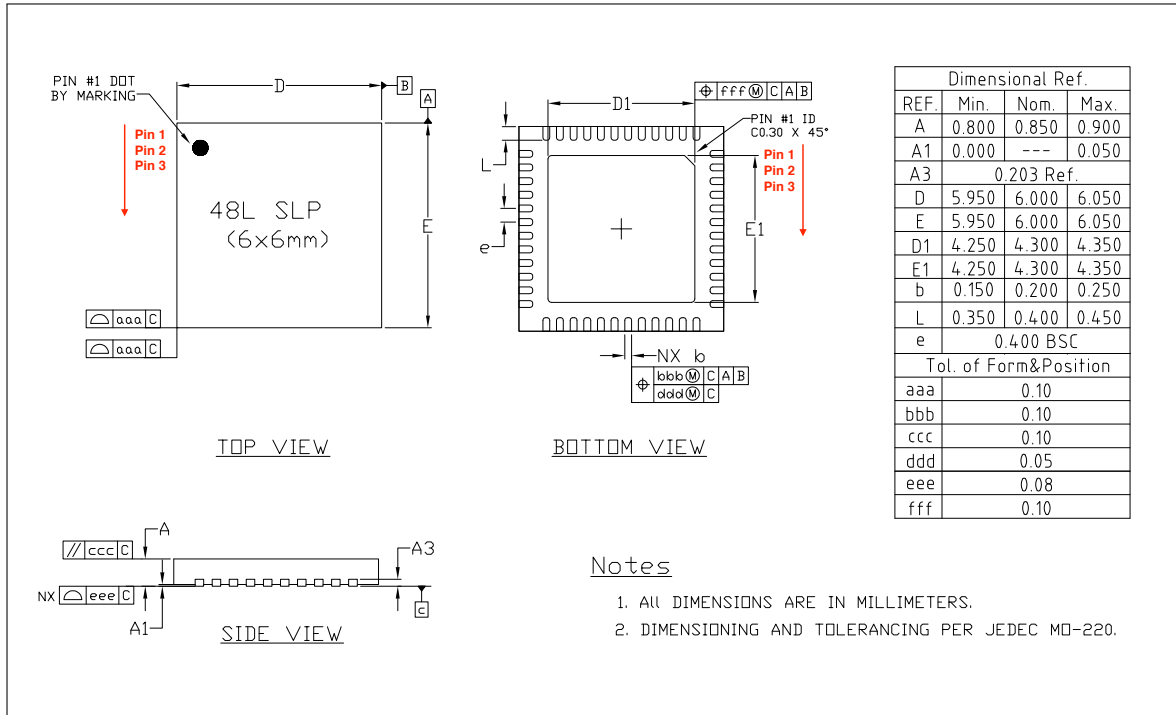


Figure 8: QFN48 (6x6 mm) Package

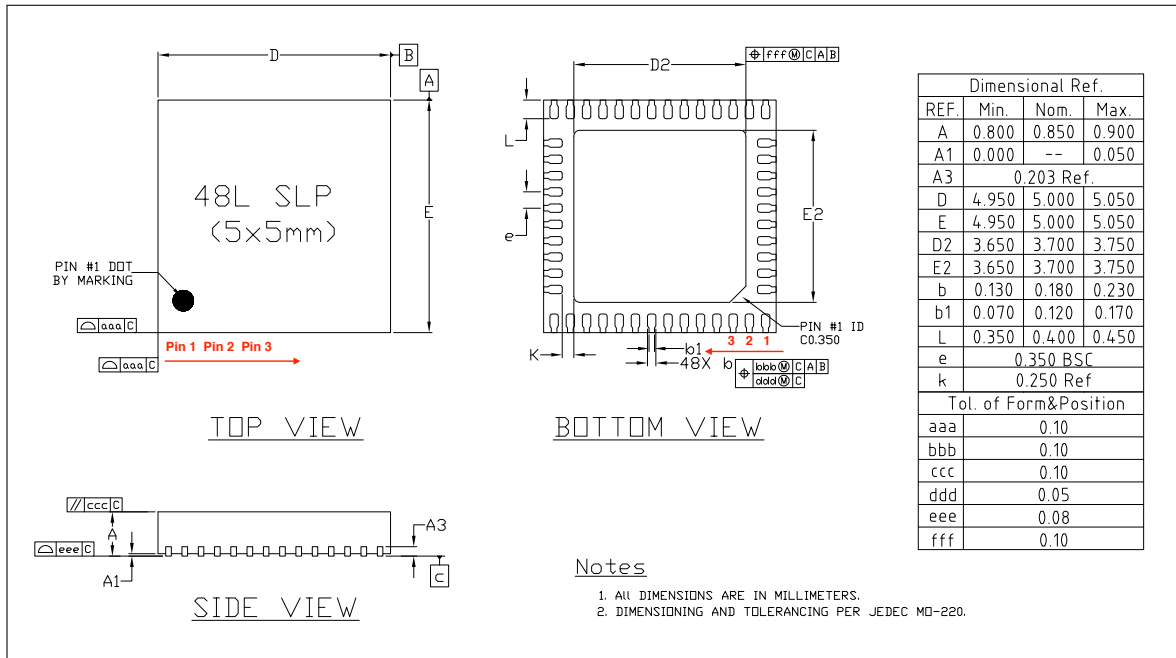
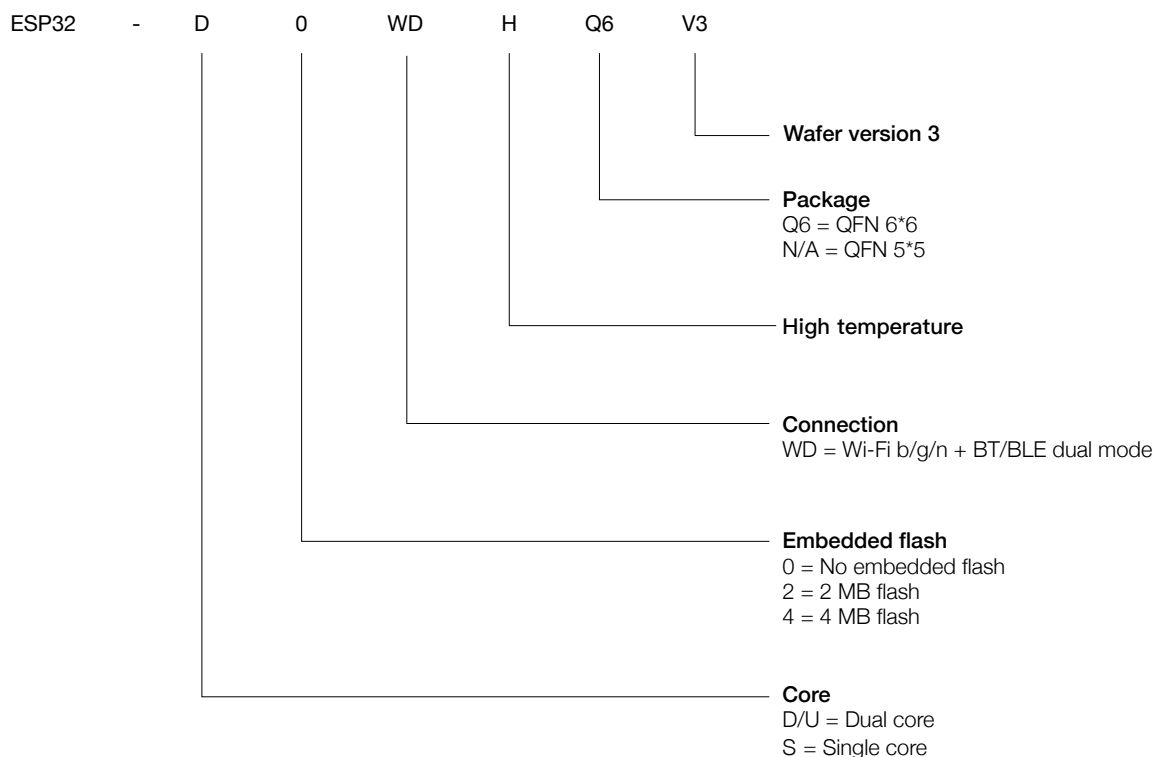


Figure 9: QFN48 (5x5 mm) Package

**Note:**

The pins of the chip are numbered in an anti-clockwise direction from Pin 1 in the top view. For information about tape, reel, and product marking, please refer to [Espressif Chip Package Information](#).

## 7 Part Number and Ordering Information



**Figure 10: ESP32 Part Number**

The table below provides the ordering information of the ESP32 series of chips.

**Table 23: ESP32 Ordering Information**

Ordering code	Core	Embedded flash	Package
ESP32-D0WD-V3	Dual core	No embedded flash	QFN 5*5
ESP32-D0WDQ6-V3 <a href="#">(NRND)</a>	Dual core	No embedded flash	QFN 6*6
ESP32-D0WD <a href="#">(NRND)</a>	Dual core	No embedded flash	QFN 5*5
ESP32-D0WDQ6 <a href="#">(NRND)</a>	Dual core	No embedded flash	QFN 6*6
ESP32-U4WDH	Dual core <sup>1</sup>	4 MB embedded flash (80 MHz)	QFN 5*5
ESP32-S0WD	Single core	No embedded flash	QFN 5*5

Note: All above chips support Wi-Fi b/g/n + Bluetooth/Bluetooth LE Dual Mode connection.

<sup>1</sup>ESP32-U4WDH is upgraded from single-core to dual-core, see [PCN-2021-021](#) (estimated effective date: December 2, 2021). The single-core version coexists with the new dual-core version around December 2, 2021. The physical product is subject to batch tracking.

## 8 Related Documentation and Resources

### Related Documentation

- [ESP32 Technical Reference Manual](#) – Detailed information on how to use the ESP32 memory and peripherals.
- [ESP32 Hardware Design Guidelines](#) – Guidelines on how to integrate the ESP32 into your hardware product.
- [ESP32 ECO and Workarounds for Bugs](#) – Correction of ESP32 design errors.
- *Certificates*  
<http://espressif.com/en/support/documents/certificates>
- *ESP32 Product/Process Change Notifications (PCN)*  
<http://espressif.com/en/support/documents/pcns>
- *ESP32 Advisories* – Information on security, bugs, compatibility, component reliability.  
<http://espressif.com/en/support/documents/advisories>
- *Documentation Updates and Update Notification Subscription*  
<http://espressif.com/en/support/download/documents>

### Developer Zone

- [ESP-IDF Programming Guide for ESP32](#) – Extensive documentation for the ESP-IDF development framework.
- *ESP-IDF* and other development frameworks on GitHub.  
<http://github.com/espressif>
- *ESP32 BBS Forum* – Engineer-to-Engineer (E2E) Community for Espressif products where you can post questions, share knowledge, explore ideas, and help solve problems with fellow engineers.  
<http://esp32.com/>
- *The ESP Journal* – Best Practices, Articles, and Notes from Espressif folks.  
<http://blog.espressif.com/>
- See the tabs *SDKs and Demos, Apps, Tools, AT Firmware*.  
<http://espressif.com/en/support/download/sdks-demos>

### Products

- *ESP32 Series SoCs* – Browse through all ESP32 SoCs.  
<http://espressif.com/en/products/socs?id=ESP32>
- *ESP32 Series Modules* – Browse through all ESP32-based modules.  
<http://espressif.com/en/products/modules?id=ESP32>
- *ESP32 Series DevKits* – Browse through all ESP32-based devkits.  
<http://espressif.com/en/products/devkits?id=ESP32>
- *ESP Product Selector* – Find an Espressif hardware product suitable for your needs by comparing or applying filters.  
<http://products.espressif.com/#/product-selector?language=en>

### Contact Us

- See the tabs *Sales Questions, Technical Enquiries, Circuit Schematic & PCB Design Review, Get Samples* (Online stores), *Become Our Supplier, Comments & Suggestions*.  
<http://espressif.com/en/contact-us/sales-questions>

## Appendix A – ESP32 Pin Lists

### A.1. Notes on ESP32 Pin Lists

Table 24: Notes on ESP32 Pin Lists

No.	Description
1	In Table <a href="#">IO_MUX</a> , the boxes highlighted in yellow indicate the GPIO pins that are input-only. Please see the following note for further details.
2	GPIO pins 34-39 are input-only. These pins do not feature an output driver or internal pull-up/pull-down circuitry. The pin names are: SENSOR_VP (GPIO36), SENSOR_CAPP (GPIO37), SENSOR_CAPN (GPIO38), SENSOR_VN (GPIO39), VDET_1 (GPIO34), VDET_2 (GPIO35).
3	The pins are grouped into four power domains: VDDA (analog power supply), VDD3P3_RTC (RTC power supply), VDD3P3_CPU (power supply of digital IOs and CPU cores), VDD_SDIO (power supply of SDIO IOs). VDD_SDIO is the output of the internal SDIO-LDO. The voltage of SDIO-LDO can be configured at 1.8 V or be the same as that of VDD3P3_RTC. The strapping pin and eFuse bits determine the default voltage of the SDIO-LDO. Software can change the voltage of the SDIO-LDO by configuring register bits. For details, please see the column “Power Domain” in Table <a href="#">IO_MUX</a> .
4	The functional pins in the VDD3P3_RTC domain are those with analog functions, including the 32 kHz crystal oscillator, ADC, DAC, and the capacitive touch sensor. Please see columns “Analog Function 0 ~ 2” in Table <a href="#">IO_MUX</a> .
5	These VDD3P3_RTC pins support the RTC function, and can work during Deep-sleep. For example, an RTC-GPIO can be used for waking up the chip from Deep-sleep.
6	The GPIO pins support up to six digital functions, as shown in columns “Function 0 ~ 5” In Table <a href="#">IO_MUX</a> . The function selection registers will be set as “ <i>N</i> ”, where <i>N</i> is the function number. Below are some definitions: <ul style="list-style-type: none"> <li>• SD_* is for signals of the SDIO slave.</li> <li>• HS1_* is for Port 1 signals of the SDIO host.</li> <li>• HS2_* is for Port 2 signals of the SDIO host.</li> <li>• MT* is for signals of the JTAG.</li> <li>• U0* is for signals of the UART0 module.</li> <li>• U1* is for signals of the UART1 module.</li> <li>• U2* is for signals of the UART2 module.</li> <li>• SPI* is for signals of the SPI01 module.</li> <li>• HSPI* is for signals of the SPI2 module.</li> <li>• VSPI* is for signals of the SPI3 module.</li> </ul>

No.	Description
7	<p>Each column about digital “Function” is accompanied by a column about “Type”. Please see the following explanations for the meanings of “type” with respect to each “function” they are associated with. For each “Function-<i>N</i>”, “type” signifies:</p> <ul style="list-style-type: none"> <li>• I: input only. If a function other than “Function-<i>N</i>” is assigned, the input signal of “Function-<i>N</i>” is still from this pin.</li> <li>• I1: input only. If a function other than “Function-<i>N</i>” is assigned, the input signal of “Function-<i>N</i>” is always “1”.</li> <li>• IO: input only. If a function other than “Function-<i>N</i>” is assigned, the input signal of “Function-<i>N</i>” is always “0”.</li> <li>• O: output only.</li> <li>• T: high-impedance.</li> <li>• I/O/T: combinations of input, output, and high-impedance according to the function signal.</li> <li>• I1/O/T: combinations of input, output, and high-impedance, according to the function signal. If a function is not selected, the input signal of the function is “1”.</li> </ul> <p>For example, pin 30 can function as HS1_CMD or SD_CMD, where HS1_CMD is of an “I1/O/T” type. If pin 30 is selected as HS1_CMD, this pin’s input and output are controlled by the SDIO host. If pin 30 is not selected as HS1_CMD, the input signal of the SDIO host is always “1”.</p>
8	<p>Each digital output pin is associated with its configurable drive strength. Column “Drive Strength” in Table <a href="#">IO_MUX</a> lists the default values. The drive strength of the digital output pins can be configured into one of the following four options:</p> <ul style="list-style-type: none"> <li>• 0: ~5 mA</li> <li>• 1: ~10 mA</li> <li>• 2: ~20 mA</li> <li>• 3: ~40 mA</li> </ul> <p>The default value is 2. The drive strength of the internal pull-up (wpu) and pull-down (wpd) is ~75 <math>\mu</math>A.</p>
9	<p>Column “At Reset” in Table <a href="#">IO_MUX</a> lists the status of each pin during reset, including input-enable (ie=1), internal pull-up (wpu) and internal pull-down (wpd). During reset, all pins are output-disabled.</p>
10	<p>Column “After Reset” in Table <a href="#">IO_MUX</a> lists the status of each pin immediately after reset, including input-enable (ie=1), internal pull-up (wpu) and internal pull-down (wpd). After reset, each pin is set to “Function 0”. The output-enable is controlled by digital Function 0.</p>
11	<p>Table <a href="#">Ethernet_MAC</a> is about the signal mapping inside Ethernet MAC. The Ethernet MAC supports MII and RMII interfaces, and supports both the internal PLL clock and the external clock source. For the MII interface, the Ethernet MAC is with/without the TX_ERR signal. MDC, MDIO, CRS and COL are slow signals, and can be mapped onto any GPIO pin through the GPIO-Matrix.</p>
12	<p>Table <a href="#">GPIO Matrix</a> is for the GPIO-Matrix. The signals of the on-chip functional modules can be mapped onto any GPIO pin. Some signals can be mapped onto a pin by both IO-MUX and GPIO-Matrix, as shown in the column tagged as “Same input signal from IO_MUX core” in Table <a href="#">GPIO Matrix</a>.</p>

No.	Description
13	*In Table <a href="#">GPIO_Matrix</a> the column “Default Value if unassigned” records the default value of the an input signal if no GPIO is assigned to it. The actual value is determined by register GPIO_FUNC <i>m</i> _IN_INV_SEL and GPIO_FUNC <i>m</i> _IN_SEL. (The value of <i>m</i> ranges from 1 to 255.)

## A.2. GPIO\_Matrix

Table 25: GPIO\_Matrix

Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
0	SPICLK_in	0	yes	SPICLK_out	SPICLK_oe
1	SPIQ_in	0	yes	SPIQ_out	SPIQ_oe
2	SPID_in	0	yes	SPID_out	SPID_oe
3	SPIHD_in	0	yes	SPIHD_out	SPIHD_oe
4	SPIWP_in	0	yes	SPIWP_out	SPIWP_oe
5	SPICS0_in	0	yes	SPICS0_out	SPICS0_oe
6	SPICS1_in	0	no	SPICS1_out	SPICS1_oe
7	SPICS2_in	0	no	SPICS2_out	SPICS2_oe
8	HSPICLK_in	0	yes	HSPICLK_out	HSPICLK_oe
9	HSPIQ_in	0	yes	HSPIQ_out	HSPIQ_oe
10	HSPID_in	0	yes	HSPID_out	HSPID_oe
11	HSPICS0_in	0	yes	HSPICS0_out	HSPICS0_oe
12	HSPIHD_in	0	yes	HSPIHD_out	HSPIHD_oe
13	HSPIWP_in	0	yes	HSPIWP_out	HSPIWP_oe
14	U0RXD_in	0	yes	U0TXD_out	1'd1
15	U0CTS_in	0	yes	U0RTS_out	1'd1
16	U0DSR_in	0	no	U0DTR_out	1'd1
17	U1RXD_in	0	yes	U1TXD_out	1'd1
18	U1CTS_in	0	yes	U1RTS_out	1'd1
23	I2S0O_BCK_in	0	no	I2S0O_BCK_out	1'd1
24	I2S1O_BCK_in	0	no	I2S1O_BCK_out	1'd1
25	I2S0O_WS_in	0	no	I2S0O_WS_out	1'd1
26	I2S1O_WS_in	0	no	I2S1O_WS_out	1'd1
27	I2S0I_BCK_in	0	no	I2S0I_BCK_out	1'd1
28	I2S0I_WS_in	0	no	I2S0I_WS_out	1'd1
29	I2CEXT0_SCL_in	1	no	I2CEXT0_SCL_out	1'd1
30	I2CEXT0_SDA_in	1	no	I2CEXT0_SDA_out	1'd1
31	pwm0_sync0_in	0	no	sdio_tohost_int_out	1'd1
32	pwm0_sync1_in	0	no	pwm0_out0a	1'd1
33	pwm0_sync2_in	0	no	pwm0_out0b	1'd1
34	pwm0_f0_in	0	no	pwm0_out1a	1'd1

Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
35	pwm0_f1_in	0	no	pwm0_out1b	1'd1
36	pwm0_f2_in	0	no	pwm0_out2a	1'd1
37	-	0	no	pwm0_out2b	1'd1
39	pcnt_sig_ch0_in0	0	no	-	1'd1
40	pcnt_sig_ch1_in0	0	no	-	1'd1
41	pcnt_ctrl_ch0_in0	0	no	-	1'd1
42	pcnt_ctrl_ch1_in0	0	no	-	1'd1
43	pcnt_sig_ch0_in1	0	no	-	1'd1
44	pcnt_sig_ch1_in1	0	no	-	1'd1
45	pcnt_ctrl_ch0_in1	0	no	-	1'd1
46	pcnt_ctrl_ch1_in1	0	no	-	1'd1
47	pcnt_sig_ch0_in2	0	no	-	1'd1
48	pcnt_sig_ch1_in2	0	no	-	1'd1
49	pcnt_ctrl_ch0_in2	0	no	-	1'd1
50	pcnt_ctrl_ch1_in2	0	no	-	1'd1
51	pcnt_sig_ch0_in3	0	no	-	1'd1
52	pcnt_sig_ch1_in3	0	no	-	1'd1
53	pcnt_ctrl_ch0_in3	0	no	-	1'd1
54	pcnt_ctrl_ch1_in3	0	no	-	1'd1
55	pcnt_sig_ch0_in4	0	no	-	1'd1
56	pcnt_sig_ch1_in4	0	no	-	1'd1
57	pcnt_ctrl_ch0_in4	0	no	-	1'd1
58	pcnt_ctrl_ch1_in4	0	no	-	1'd1
61	HSPICS1_in	0	no	HSPICS1_out	HSPICS1_oe
62	HSPICS2_in	0	no	HSPICS2_out	HSPICS2_oe
63	VSPICLK_in	0	yes	VSPICLK_out_mux	VSPICLK_oe
64	VSPIQ_in	0	yes	VSPIQ_out	VSPIQ_oe
65	VSPID_in	0	yes	VSPID_out	VSPID_oe
66	VSPIHD_in	0	yes	VSPIHD_out	VSPIHD_oe
67	VSPIWP_in	0	yes	VSPIWP_out	VSPIWP_oe
68	VSPICS0_in	0	yes	VSPICS0_out	VSPICS0_oe
69	VSPICS1_in	0	no	VSPICS1_out	VSPICS1_oe
70	VSPICS2_in	0	no	VSPICS2_out	VSPICS2_oe
71	pcnt_sig_ch0_in5	0	no	ledc_hs_sig_out0	1'd1
72	pcnt_sig_ch1_in5	0	no	ledc_hs_sig_out1	1'd1
73	pcnt_ctrl_ch0_in5	0	no	ledc_hs_sig_out2	1'd1
74	pcnt_ctrl_ch1_in5	0	no	ledc_hs_sig_out3	1'd1
75	pcnt_sig_ch0_in6	0	no	ledc_hs_sig_out4	1'd1
76	pcnt_sig_ch1_in6	0	no	ledc_hs_sig_out5	1'd1
77	pcnt_ctrl_ch0_in6	0	no	ledc_hs_sig_out6	1'd1
78	pcnt_ctrl_ch1_in6	0	no	ledc_hs_sig_out7	1'd1

Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
79	pcnt_sig_ch0_in7	0	no	ledc_ls_sig_out0	1'd1
80	pcnt_sig_ch1_in7	0	no	ledc_ls_sig_out1	1'd1
81	pcnt_ctrl_ch0_in7	0	no	ledc_ls_sig_out2	1'd1
82	pcnt_ctrl_ch1_in7	0	no	ledc_ls_sig_out3	1'd1
83	rmt_sig_in0	0	no	ledc_ls_sig_out4	1'd1
84	rmt_sig_in1	0	no	ledc_ls_sig_out5	1'd1
85	rmt_sig_in2	0	no	ledc_ls_sig_out6	1'd1
86	rmt_sig_in3	0	no	ledc_ls_sig_out7	1'd1
87	rmt_sig_in4	0	no	rmt_sig_out0	1'd1
88	rmt_sig_in5	0	no	rmt_sig_out1	1'd1
89	rmt_sig_in6	0	no	rmt_sig_out2	1'd1
90	rmt_sig_in7	0	no	rmt_sig_out3	1'd1
91	-	-	-	rmt_sig_out4	1'd1
92	-	-	-	rmt_sig_out6	1'd1
94	twai_rx	1	no	rmt_sig_out7	1'd1
95	I2CEXT1_SCL_in	1	no	I2CEXT1_SCL_out	1'd1
96	I2CEXT1_SDA_in	1	no	I2CEXT1_SDA_out	1'd1
97	host_card_detect_n_1	0	no	host_ccmd_od_pullup_en_n	1'd1
98	host_card_detect_n_2	0	no	host_rst_n_1	1'd1
99	host_card_write_prt_1	0	no	host_rst_n_2	1'd1
100	host_card_write_prt_2	0	no	gpio_sd0_out	1'd1
101	host_card_int_n_1	0	no	gpio_sd1_out	1'd1
102	host_card_int_n_2	0	no	gpio_sd2_out	1'd1
103	pwm1_sync0_in	0	no	gpio_sd3_out	1'd1
104	pwm1_sync1_in	0	no	gpio_sd4_out	1'd1
105	pwm1_sync2_in	0	no	gpio_sd5_out	1'd1
106	pwm1_f0_in	0	no	gpio_sd6_out	1'd1
107	pwm1_f1_in	0	no	gpio_sd7_out	1'd1
108	pwm1_f2_in	0	no	pwm1_out0a	1'd1
109	pwm0_cap0_in	0	no	pwm1_out0b	1'd1
110	pwm0_cap1_in	0	no	pwm1_out1a	1'd1
111	pwm0_cap2_in	0	no	pwm1_out1b	1'd1
112	pwm1_cap0_in	0	no	pwm1_out2a	1'd1
113	pwm1_cap1_in	0	no	pwm1_out2b	1'd1
114	pwm1_cap2_in	0	no	pwm2_out1h	1'd1
115	pwm2_fta	1	no	pwm2_out1l	1'd1
116	pwm2_ftb	1	no	pwm2_out2h	1'd1
117	pwm2_cap1_in	0	no	pwm2_out2l	1'd1
118	pwm2_cap2_in	0	no	pwm2_out3h	1'd1
119	pwm2_cap3_in	0	no	pwm2_out3l	1'd1
120	pwm3_fta	1	no	pwm2_out4h	1'd1



Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
121	pwm3_fltb	1	no	pwm2_out4l	1'd1
122	pwm3_cap1_in	0	no	-	1'd1
123	pwm3_cap2_in	0	no	twai_tx	1'd1
124	pwm3_cap3_in	0	no	twai_bus_off_on	1'd1
125	-	-	-	twai_clkout	1'd1
140	I2S0I_DATA_in0	0	no	I2S0O_DATA_out0	1'd1
141	I2S0I_DATA_in1	0	no	I2S0O_DATA_out1	1'd1
142	I2S0I_DATA_in2	0	no	I2S0O_DATA_out2	1'd1
143	I2S0I_DATA_in3	0	no	I2S0O_DATA_out3	1'd1
144	I2S0I_DATA_in4	0	no	I2S0O_DATA_out4	1'd1
145	I2S0I_DATA_in5	0	no	I2S0O_DATA_out5	1'd1
146	I2S0I_DATA_in6	0	no	I2S0O_DATA_out6	1'd1
147	I2S0I_DATA_in7	0	no	I2S0O_DATA_out7	1'd1
148	I2S0I_DATA_in8	0	no	I2S0O_DATA_out8	1'd1
149	I2S0I_DATA_in9	0	no	I2S0O_DATA_out9	1'd1
150	I2S0I_DATA_in10	0	no	I2S0O_DATA_out10	1'd1
151	I2S0I_DATA_in11	0	no	I2S0O_DATA_out11	1'd1
152	I2S0I_DATA_in12	0	no	I2S0O_DATA_out12	1'd1
153	I2S0I_DATA_in13	0	no	I2S0O_DATA_out13	1'd1
154	I2S0I_DATA_in14	0	no	I2S0O_DATA_out14	1'd1
155	I2S0I_DATA_in15	0	no	I2S0O_DATA_out15	1'd1
156	-	-	-	I2S0O_DATA_out16	1'd1
157	-	-	-	I2S0O_DATA_out17	1'd1
158	-	-	-	I2S0O_DATA_out18	1'd1
159	-	-	-	I2S0O_DATA_out19	1'd1
160	-	-	-	I2S0O_DATA_out20	1'd1
161	-	-	-	I2S0O_DATA_out21	1'd1
162	-	-	-	I2S0O_DATA_out22	1'd1
163	-	-	-	I2S0O_DATA_out23	1'd1
164	I2S1I_BCK_in	0	no	I2S1I_BCK_out	1'd1
165	I2S1I_WS_in	0	no	I2S1I_WS_out	1'd1
166	I2S1I_DATA_in0	0	no	I2S1O_DATA_out0	1'd1
167	I2S1I_DATA_in1	0	no	I2S1O_DATA_out1	1'd1
168	I2S1I_DATA_in2	0	no	I2S1O_DATA_out2	1'd1
169	I2S1I_DATA_in3	0	no	I2S1O_DATA_out3	1'd1
170	I2S1I_DATA_in4	0	no	I2S1O_DATA_out4	1'd1
171	I2S1I_DATA_in5	0	no	I2S1O_DATA_out5	1'd1
172	I2S1I_DATA_in6	0	no	I2S1O_DATA_out6	1'd1
173	I2S1I_DATA_in7	0	no	I2S1O_DATA_out7	1'd1
174	I2S1I_DATA_in8	0	no	I2S1O_DATA_out8	1'd1
175	I2S1I_DATA_in9	0	no	I2S1O_DATA_out9	1'd1

Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
176	I2S1I_DATA_in10	0	no	I2S1O_DATA_out10	1'd1
177	I2S1I_DATA_in11	0	no	I2S1O_DATA_out11	1'd1
178	I2S1I_DATA_in12	0	no	I2S1O_DATA_out12	1'd1
179	I2S1I_DATA_in13	0	no	I2S1O_DATA_out13	1'd1
180	I2S1I_DATA_in14	0	no	I2S1O_DATA_out14	1'd1
181	I2S1I_DATA_in15	0	no	I2S1O_DATA_out15	1'd1
182	-	-	-	I2S1O_DATA_out16	1'd1
183	-	-	-	I2S1O_DATA_out17	1'd1
184	-	-	-	I2S1O_DATA_out18	1'd1
185	-	-	-	I2S1O_DATA_out19	1'd1
186	-	-	-	I2S1O_DATA_out20	1'd1
187	-	-	-	I2S1O_DATA_out21	1'd1
188	-	-	-	I2S1O_DATA_out22	1'd1
189	-	-	-	I2S1O_DATA_out23	1'd1
190	I2S0I_H_SYNC	0	no	pwm3_out1h	1'd1
191	I2S0I_V_SYNC	0	no	pwm3_out1l	1'd1
192	I2S0I_H_ENABLE	0	no	pwm3_out2h	1'd1
193	I2S1I_H_SYNC	0	no	pwm3_out2l	1'd1
194	I2S1I_V_SYNC	0	no	pwm3_out3h	1'd1
195	I2S1I_H_ENABLE	0	no	pwm3_out3l	1'd1
196	-	-	-	pwm3_out4h	1'd1
197	-	-	-	pwm3_out4l	1'd1
198	U2RXD_in	0	yes	U2TXD_out	1'd1
199	U2CTS_in	0	yes	U2RTS_out	1'd1
200	<a href="#">emac_mdc_i</a>	0	no	<a href="#">emac_mdc_o</a>	<a href="#">emac_mdc_oe</a>
201	<a href="#">emac_mdi_i</a>	0	no	<a href="#">emac_mdo_o</a>	<a href="#">emac_mdo_o_e</a>
202	<a href="#">emac_crs_i</a>	0	no	<a href="#">emac_crs_o</a>	<a href="#">emac_crs_oe</a>
203	<a href="#">emac_col_i</a>	0	no	<a href="#">emac_col_o</a>	<a href="#">emac_col_oe</a>
204	pcmfsync_in	0	no	bt_audio0_irq	1'd1
205	pcmclk_in	0	no	bt_audio1_irq	1'd1
206	pcmdin	0	no	bt_audio2_irq	1'd1
207	-	-	-	ble_audio0_irq	1'd1
208	-	-	-	ble_audio1_irq	1'd1
209	-	-	-	ble_audio2_irq	1'd1
210	-	-	-	pcmfsync_out	pcmfsync_en
211	-	-	-	pcmclk_out	pcmclk_en
212	-	-	-	pcmdout	pcmdout_en
213	-	-	-	ble_audio_sync0_p	1'd1
214	-	-	-	ble_audio_sync1_p	1'd1
215	-	-	-	ble_audio_sync2_p	1'd1
224	-	-	-	sig_in_func224	1'd1

Signal No.	Input signals	Default value if unassigned*	Same input signal from IO_MUX core	Output signals	Output enable of output signals
225	-	-	-	sig_in_func225	1'd1
226	-	-	-	sig_in_func226	1'd1
227	-	-	-	sig_in_func227	1'd1
228	-	-	-	sig_in_func228	1'd1

### A.3. Ethernet\_MAC

Table 26: Ethernet\_MAC

PIN Name	Function6	MII (int_osc)	MII (ext_osc)	RMII (int_osc)	RMII (ext_osc)
GPIO0	EMAC_TX_CLK	TX_CLK (I)	TX_CLK (I)	CLK_OUT(O)	EXT_OSC_CLK(I)
GPIO5	EMAC_RX_CLK	RX_CLK (I)	RX_CLK (I)	-	-
GPIO21	EMAC_TX_EN	TX_EN(O)	TX_EN(O)	TX_EN(O)	TX_EN(O)
GPIO19	EMAC_TXD0	TXD[0](O)	TXD[0](O)	TXD[0](O)	TXD[0](O)
GPIO22	EMAC_TXD1	TXD[1](O)	TXD[1](O)	TXD[1](O)	TXD[1](O)
MTMS	EMAC_TXD2	TXD[2](O)	TXD[2](O)	-	-
MTDI	EMAC_TXD3	TXD[3](O)	TXD[3](O)	-	-
MTCK	EMAC_RX_ER	RX_ER(I)	RX_ER(I)	-	-
GPIO27	EMAC_RX_DV	RX_DV(I)	RX_DV(I)	CRS_DV(I)	CRS_DV(I)
GPIO25	EMAC_RXD0	RXD[0](I)	RXD[0](I)	RXD[0](I)	RXD[0](I)
GPIO26	EMAC_RXD1	RXD[1](I)	RXD[1](I)	RXD[1](I)	RXD[1](I)
U0TXD	EMAC_RXD2	RXD[2](I)	RXD[2](I)	-	-
MTDO	EMAC_RXD3	RXD[3](I)	RXD[3](I)	-	-
GPIO16	EMAC_CLK_OUT	CLK_OUT(O)	-	CLK_OUT(O)	-
GPIO17	EMAC_CLK_OUT_180	CLK_OUT_180(O)	-	CLK_OUT_180(O)	-
GPIO4	EMAC_TX_ER	TX_ERR(O)*	TX_ERR(O)*	-	-
In GPIO Matrix*	-	MDC(O)	MDC(O)	MDC(O)	MDC(O)
In GPIO Matrix*	-	MDIO(IO)	MDIO(IO)	MDIO(IO)	MDIO(IO)
In GPIO Matrix*	-	CRS(I)	CRS(I)	-	-
In GPIO Matrix*	-	COL(I)	COL(I)	-	-

\*Notes: 1. The GPIO Matrix can be any GPIO. 2. The TX\_ERR (O) is optional.

### A.4. IO\_MUX

For the list of IO\_MUX pins, please see the next page.

IO\_MUX

Pin No.	Power Supply Pin	Analog Pin	Digital Pin	Power Domain	Analog Function0	Analog Function1	Analog Function2	RTC Function0	RTC Function1	Function0	Type	Function1	Type	Function2	Type	Function3	Type	Function4	Type	Function5	Type	Drive Strength (2'd2: 20 mA)	At Reset	After Reset			
1	VDDA			VDDA supply in																							
2		LNA_IN		VDD3P3																							
3	VDD3P3			VDD3P3 supply in																							
4	VDD3P3			VDD3P3 supply in																							
5		SENSOR_VP		VDD3P3_RTC	ADC_H	ADC1_CH0		RTC_GPIO0		GPIO36	I			GPIO36	I									oe=0, ie=0			
6		SENSOR_CAPP		VDD3P3_RTC	ADC_H	ADC1_CH1		RTC_GPIO1		GPIO37	I			GPIO37	I									oe=0, ie=0			
7		SENSOR_CAPN		VDD3P3_RTC	ADC_H	ADC1_CH2		RTC_GPIO2		GPIO38	I			GPIO38	I									oe=0, ie=0			
8		SENSOR_VN		VDD3P3_RTC	ADC_H	ADC1_CH3		RTC_GPIO3		GPIO39	I			GPIO39	I									oe=0, ie=0			
9		CHIP_PU		VDD3P3_RTC																							
10		VDET_1		VDD3P3_RTC		ADC1_CH6		RTC_GPIO4		GPIO34	I			GPIO34	I									oe=0, ie=0			
11		VDET_2		VDD3P3_RTC		ADC1_CH7		RTC_GPIO5		GPIO35	I			GPIO35	I									oe=0, ie=0			
12		32K_XP		VDD3P3_RTC	XTAL_32K_P	ADC1_CH4	TOUCH9	RTC_GPIO9		GPIO32	I/O/T			GPIO32	I/O/T							2'd2	oe=0, ie=0	oe=0, ie=0			
13		32K_XN		VDD3P3_RTC	XTAL_32K_N	ADC1_CH5	TOUCH8	RTC_GPIO8		GPIO33	I/O/T			GPIO33	I/O/T							2'd2	oe=0, ie=0	oe=0, ie=0			
14			GPIO25	VDD3P3_RTC	DAC_1	ADC2_CH8		RTC_GPIO6		GPIO25	I/O/T			GPIO25	I/O/T							I	2'd2	oe=0, ie=0	oe=0, ie=0		
15			GPIO26	VDD3P3_RTC	DAC_2	ADC2_CH9		RTC_GPIO7		GPIO26	I/O/T			GPIO26	I/O/T								I	2'd2	oe=0, ie=0	oe=0, ie=0	
16			GPIO27	VDD3P3_RTC		ADC2_CH7	TOUCH7	RTC_GPIO17		GPIO27	I/O/T			GPIO27	I/O/T								I	2'd2	oe=0, ie=0	oe=0, ie=0	
17			MTMS	VDD3P3_RTC		ADC2_CH6	TOUCH6	RTC_GPIO16		MTMS	I0	HSPICLK	I/O/T	GPIO14	I/O/T	HS2_CLK	O	SD_CLK	I0				O	2'd2	oe=0, ie=0	oe=0, ie=1, wpu	
18			MTDI	VDD3P3_RTC		ADC2_CH5	TOUCH5	RTC_GPIO15		MTDI	I1	HSPIQ	I/O/T	GPIO12	I/O/T	HS2_DATA2	I1/O/T	SD_DATA2	I1/O/T				O	2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd	
19	VDD3P3_RTC			VDD3P3_RTC supply in																							
20			MTCK	VDD3P3_RTC		ADC2_CH4	TOUCH4	RTC_GPIO14		MTCK	I1	HSPID	I/O/T	GPIO13	I/O/T	HS2_DATA3	I1/O/T	SD_DATA3	I1/O/T				I	2'd2	oe=0, ie=0	oe=0, ie=1, wpd	
21			MTDO	VDD3P3_RTC		ADC2_CH3	TOUCH3	RTC_GPIO13	I2C_SDA	MTDO	O/T	HSPICS0	I/O/T	GPIO15	I/O/T	HS2_CMD	I1/O/T	SD_CMD	I1/O/T				I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpd	
22			GPIO2	VDD3P3_RTC		ADC2_CH2	TOUCH2	RTC_GPIO12	I2C_SCL	GPIO2	I/O/T	HSPiWP	I/O/T	GPIO2	I/O/T	HS2_DATA0	I1/O/T	SD_DATA0	I1/O/T					2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd	
23			GPIO0	VDD3P3_RTC		ADC2_CH1	TOUCH1	RTC_GPIO11	I2C_SDA	GPIO0	I/O/T	CLK_OUT1	O	GPIO0	I/O/T								I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
24			GPIO4	VDD3P3_RTC		ADC2_CH0	TOUCH0	RTC_GPIO10	I2C_SCL	GPIO4	I/O/T	HSPiHD	I/O/T	GPIO4	I/O/T	HS2_DATA1	I1/O/T	SD_DATA1	I1/O/T				O	2'd2	oe=0, ie=1, wpd	oe=0, ie=1, wpd	
25			GPIO16	VDD_SDIO						GPIO16	I/O/T			GPIO16	I/O/T	HS1_DATA4	I1/O/T	U2RXD	I1				O	2'd2	oe=0, ie=0	oe=0, ie=1	
26				VDD_SDIO																							
27			GPIO17	VDD_SDIO						GPIO17	I/O/T			GPIO17	I/O/T	HS1_DATA5	I1/O/T	U2TXD	O				O	2'd2	oe=0, ie=0	oe=0, ie=1	
28			SD_DATA_2	VDD_SDIO						SD_DATA2	I1/O/T	SPiHD	I/O/T	GPIO9	I/O/T	HS1_DATA2	I1/O/T	U1RXD	I1					2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
29			SD_DATA_3	VDD_SDIO						SD_DATA3	I0/O/T	SPiWP	I/O/T	GPIO10	I/O/T	HS1_DATA3	I1/O/T	U1TXD	O					2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
30			SD_CMD	VDD_SDIO						SD_CMD	I1/O/T	SPiCS0	I/O/T	GPIO11	I/O/T	HS1_CMD	I1/O/T	U1RTS	O					2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
31			SD_CLK	VDD_SDIO						SD_CLK	I0	SPiCLK	I/O/T	GPIO6	I/O/T	HS1_CLK	O	U1CTS	I1					2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
32			SD_DATA_0	VDD_SDIO						SD_DATA0	I1/O/T	SPiD	I/O/T	GPIO7	I/O/T	HS1_DATA0	I1/O/T	U2RTS	O					2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
33			SD_DATA_1	VDD_SDIO						SD_DATA1	I1/O/T	SPiD	I/O/T	GPIO8	I/O/T	HS1_DATA1	I1/O/T	U2CTS	I1					2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
34			GPIO5	VDD3P3_CPU						GPIO5	I/O/T	VSPiCS0	I/O/T	GPIO5	I/O/T	HS1_DATA6	I1/O/T						I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu	
35			GPIO18	VDD3P3_CPU						GPIO18	I/O/T	VSPiCLK	I/O/T	GPIO18	I/O/T	HS1_DATA7	I1/O/T							2'd2	oe=0, ie=0	oe=0, ie=1	
36			GPIO23	VDD3P3_CPU						GPIO23	I/O/T	VSPiD	I/O/T	GPIO23	I/O/T	HS1_STROBE	I0							2'd2	oe=0, ie=0	oe=0, ie=1	
37	VDD3P3_CPU			VDD3P3_CPU supply in																							
38			GPIO19	VDD3P3_CPU						GPIO19	I/O/T	VSPiQ	I/O/T	GPIO19	I/O/T	U0CTS	I1						O	2'd2	oe=0, ie=0	oe=0, ie=1	
39			GPIO22	VDD3P3_CPU						GPIO22	I/O/T	VSPiWP	I/O/T	GPIO22	I/O/T	U0RTS	O							O	2'd2	oe=0, ie=0	oe=0, ie=1
40			U0RXD	VDD3P3_CPU						U0RXD	I1	CLK_OUT2	O	GPIO3	I/O/T										2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
41			U0TXD	VDD3P3_CPU						U0TXD	O	CLK_OUT3	O	GPIO1	I/O/T									I	2'd2	oe=0, ie=1, wpu	oe=0, ie=1, wpu
42			GPIO21	VDD3P3_CPU						GPIO21	I/O/T	VSPiHD	I/O/T	GPIO21	I/O/T									O	2'd2	oe=0, ie=0	oe=0, ie=1
43	VDDA			VDDA supply in																							
44		XTAL_N		VDDA																							
45		XTAL_P		VDDA																							
46	VDDA			VDDA supply in																							
47		CAP2		VDDA																							
48		CAP1		VDDA																							
<b>Total Number</b>	<b>8</b>	<b>14</b>	<b>26</b>																								

**Notes:**

- wpu: weak pull-up;
- wpd: weak pull-down;
- ie: input enable;
- oe: output enable;

\* Please see Table: Notes on ESP32 Pin Lists for more information. (请参考表: 管脚清单说明。)

## Revision History

Date	Version	Release notes
2021.10	v3.8	<p>Upgraded ESP32-U4WDH variant from single-core to dual-core, see <a href="#">PCN-2021-021</a> (estimated effective date: December 2, 2021). The single-core version coexists with the new dual-core version around December 2, 2021. The physical product is subject to batch tracking.</p> <p>Added CoreMark<sup>®</sup> score in Section 1.4: <i>MCU and Advanced Features</i></p> <p>Updated the description to TWAI in Section 4.1.18: <i>TWAI Controller</i></p> <p>Added <a href="#">Not Recommended for New Designs (NRND)</a> label to the ESP32-D0WDQ6-V3 variant</p> <p>Provided a link to <a href="#">Espressif Chip Package Information</a> in Section 6: <i>Package Information</i></p> <p>Updated Section 1.3: <i>Bluetooth Key Features</i></p>
2021.07	v3.7	<p>Removed ESP32-D2WD variant</p> <p>Updated the wording in Section 3.6.1</p> <p>Updated pin function numbers starting from Function0</p> <p>Added <a href="#">Not Recommended for New Designs (NRND)</a> label to ESP32-D0WD and ESP32-D0WDQ6 variants</p>
2021.03	V3.6	<p>Updated Figure 1: <i>Functional Block Diagram</i></p> <p>Updated Table 14: <i>Reliability Qualifications</i></p> <p>Updated Figure 4: <i>ESP32 Power Scheme</i></p> <p>Updated Table 12: <i>Recommended Operating Conditions</i></p> <p>Updated the notes below Table 2: <i>Description of ESP32 Power-up and Reset Timing Parameters</i></p> <p>Provided more information about TWAI<sup>®</sup> in Table 5, Table 10, Table 8, and Section 4.1</p>
2021.01	V3.5	<p>Updated the description for CAP2 from 3 nF to 3.3 nF</p> <p>Added TWAI<sup>®</sup> in Section 1.4.3: <i>Advanced Peripheral Interfaces</i></p> <p>Updated Figure 1: <i>Functional Block Diagram</i></p> <p>Updated the reset values for MTCK, MTMS, GPIO27 in Appendix IO_MUX</p>
2020.04	V3.4	<p>Added one chip variant: ESP32-U4WDH</p> <p>Updated some figures in Table 6, 16, 17, 19, 21, 22</p> <p>Added a note under Table 18</p>
2020.01	V3.3	<p>Added two chip variants: ESP32-D0WD-V3 and ESP32-D0WDQ6-V3.</p> <p>Added a note under Table 7.</p>
2019.10	V3.2	<p>Updated Figure 5: <i>ESP32 Power-up and Reset Timing.</i></p>
2019.07	V3.1	<p>Added pin-pin mapping between ESP32-D2WD and the embedded flash under Table 1 <i>Pin Description</i>;</p> <p>Updated Figure 10 <i>ESP32 Part Number.</i></p>
2019.04	V3.0	<p>Added information about the setup and hold times for the strapping pins in Section 2.4: <i>Strapping Pins.</i></p>
2019.02	V2.9	<p>Applied new formatting to Table 1: <i>Pin Description</i>;</p> <p>Fixed typos with respect to the ADC1 channel mappings in Table 10: <i>Peripheral Pin Configurations.</i></p>

Date	Version	Release notes
2019.01	V2.8	Changed the RF power control range in Table 18, Table 20 and Table 22 from -12 ~ +12 to -12 ~ +9 dBm; Small text changes.
2018.11	V2.7	Updated Section 1.5; Updated pin statuses at reset and after reset in Table IO_MUX.
2018.10	V2.6	Updated QFN package drawings in Chapter 6: Package Information.
2018.08	V2.5	<ul style="list-style-type: none"> <li>Added "Cumulative IO output current" entry to Table 11: Absolute Maximum Ratings;</li> <li>Added more parameters to Table 13: DC Characteristics;</li> <li>Changed the power domain names in Table IO_MUX to be consistent with the pin names.</li> </ul>
2018.07	V2.4	<ul style="list-style-type: none"> <li>Deleted information on Packet Traffic Arbitration (PTA);</li> <li>Added Figure 5: ESP32 Power-up and Reset Timing in Section 2.3: Power Scheme;</li> <li>Added the power consumption of dual-core SoCs in Table 6: Power Consumption by Power Modes;</li> <li>Updated section 4.1.2: Analog-to-Digital Converter (ADC).</li> </ul>
2018.06	V2.3	Added the power consumption at CPU frequency of 160 MHz in Table 6: Power Consumption by Power Modes.
2018.05	V2.2	<ul style="list-style-type: none"> <li>Changed the voltage range of VDD3P3_RTC from 1.8-3.6V to 2.3-3.6V in Table 1: Pin Description;</li> <li>Updated Section 2.3: Power Scheme;</li> <li>Updated Section 3.1.3: External Flash and SRAM;</li> <li>Updated Table 6: Power Consumption by Power Modes;</li> <li>Deleted content about temperature sensor;</li> </ul> <p>Changes to electrical characteristics:</p> <ul style="list-style-type: none"> <li>Updated Table 11: Absolute Maximum Ratings;</li> <li>Added Table 12: Recommended Operating Conditions;</li> <li>Added Table 13: DC Characteristics;</li> <li>Added Table 14: Reliability Qualifications;</li> <li>Updated the values of "Gain control step" and "Adjacent channel transmit power" in Table 18: Transmitter Characteristics - Basic Data Rate;</li> <li>Updated the values of "Gain control step", "<math>\pi/4</math> DQPSK modulation accuracy", "8 DPSK modulation accuracy" and "In-band spurious emissions" in Table 20: Transmitter Characteristics – Enhanced Data Rate;</li> <li>Updated the values of "Gain control step", "Adjacent channel transmit power" in Table 22: Transmitter Characteristics - BLE.</li> </ul>
2018.01	V2.1	<ul style="list-style-type: none"> <li>Deleted software-specific features;</li> <li>Deleted information on LNA pre-amplifier;</li> <li>Specified the CPU speed and flash speed of ESP32-D2WD;</li> <li>Added notes to Section 2.3: Power Scheme.</li> </ul>
2017.12	V2.0	Added a note on the sequence of pin number in Chapter 6.

Date	Version	Release notes
2017.10	V1.9	<ul style="list-style-type: none"> <li>• Updated the description of the pin CHIP_PU in Table 1;</li> <li>• Added a note to Section 2.3: Power Scheme;</li> <li>• Updated the description of the chip's system reset in Section 2.4: Strapping Pins;</li> <li>• Added a description of antenna diversity and selection to Section 3.5.1;</li> <li>• Deleted "Association sleep pattern" in Table 6 and added notes to Active sleep and Modem-sleep.</li> </ul>
2017.08	V1.8	<ul style="list-style-type: none"> <li>• Added Table 4.2 in Section 4;</li> <li>• Corrected a typo in Figure 1.</li> </ul>
2017.08	V1.7	<ul style="list-style-type: none"> <li>• Changed the transmitting power to +12 dBm; the sensitivity of NZIF receiver to -97 dBm in Section 1.3;</li> <li>• Added a note to Table 1 Pin Description;</li> <li>• Added 160 MHz clock frequency in section 3.1.1;</li> <li>• Changed the transmitting power from 21 dBm to 20.5 dBm in Section 3.5.1;</li> <li>• Changed the dynamic control range of class-1, class-2 and class-3 transmit output powers to "up to 24 dBm"; and changed the dynamic range of NZIF receiver sensitivity to "over 97 dB" in Section 3.6.1;</li> <li>• Updated Table 6: Power Consumption by Power Modes, and added two notes to it;</li> <li>• Updated sections 4.1.1, 4.1.9;</li> <li>• Updated Table 11: Absolute Maximum Ratings;</li> <li>• Updated Table 15: RF Power Consumption Specifications, and changed the duty cycle on which the transmitters' measurements are based by 50%.</li> <li>• Updated Table 16: Wi-Fi Radio Characteristics and added a note on "Output impedance" to it;</li> <li>• Updated parameter "Sensitivity" in Table 17, 19, 21;</li> <li>• Updated parameters "RF transmit power" and "RF power control range", and added parameter "Gain control step" in Table 18, 20, 22;</li> <li>• Deleted Chapters: "Touch Sensor" and "Code Examples";</li> <li>• Added a link to <a href="#">certification download</a>.</li> </ul>
2017.06	V1.6	<p>Corrected two typos:</p> <ul style="list-style-type: none"> <li>• Changed the number of external components to 20 in Section 1.1.2;</li> <li>• Changed the number of GPIO pins to 34 in Section 4.1.1.</li> </ul>
2017.06	V1.5	<ul style="list-style-type: none"> <li>• Changed the power supply range in Section: 1.4.1 CPU and Memory;</li> <li>• Updated the note in Section 2.3: Power Scheme;</li> <li>• Updated Table 11: Absolute Maximum Ratings;</li> <li>• Changed the drive strength values of the digital output pins in Note 8, in Table 24: Notes on ESP32 Pin Lists;</li> <li>• Added the option to subscribe for notifications of documentation changes.</li> </ul>

Date	Version	Release notes
2017.05	V1.4	<ul style="list-style-type: none"> <li>• Added a note to the frequency of the external crystal oscillator in Section <a href="#">1.4.2: Clocks and Timers</a>;</li> <li>• Added a note to Section <a href="#">2.4: Strapping Pins</a>;</li> <li>• Updated Section <a href="#">3.7: RTC and Low-Power Management</a>;</li> <li>• Changed the maximum driving capability from 12 mA to 80 mA, in Table <a href="#">11: Absolute Maximum Ratings</a>;</li> <li>• Changed the input impedance value of 50Ω, in Table <a href="#">16: Wi-Fi Radio Characteristics</a>, to output impedance value of 30+j10 Ω;</li> <li>• Added a note to No.8 in Table <a href="#">24: Notes on ESP32 Pin Lists</a>;</li> <li>• Deleted GPIO20 in Table <a href="#">IO_MUX</a>.</li> </ul>
2017.04	V1.3	<ul style="list-style-type: none"> <li>• Added Appendix: <a href="#">ESP32 Pin Lists</a>;</li> <li>• Updated Table: <a href="#">Wi-Fi Radio Characteristics</a>;</li> <li>• Updated Figure: <a href="#">ESP32 Pin Layout (for QFN 5*5)</a>.</li> </ul>
2017.03	V1.2	<ul style="list-style-type: none"> <li>• Added a note to Table: <a href="#">Pin Description</a>;</li> <li>• Updated the note in Section: <a href="#">Internal Memory</a>.</li> </ul>
2017.02	V1.1	<ul style="list-style-type: none"> <li>• Added Chapter: <a href="#">Part Number and Ordering Information</a>;</li> <li>• Updated Section: <a href="#">MCU and Advanced Features</a>;</li> <li>• Updated Section: <a href="#">Block Diagram</a>;</li> <li>• Updated Chapter: <a href="#">Pin Definitions</a>;</li> <li>• Updated Section: <a href="#">CPU and Memory</a>;</li> <li>• Updated Section: <a href="#">Audio PLL Clock</a>;</li> <li>• Updated Section: <a href="#">Absolute Maximum Ratings</a>;</li> <li>• Updated Chapter: <a href="#">Package Information</a>;</li> <li>• Updated Chapter: <a href="#">Learning Resources</a>.</li> </ul>
2016.08	V1.0	First release.





[www.espressif.com](http://www.espressif.com)

## Disclaimer and Copyright Notice

Information in this document, including URL references, is subject to change without notice.

ALL THIRD PARTY'S INFORMATION IN THIS DOCUMENT IS PROVIDED AS IS WITH NO WARRANTIES TO ITS AUTHENTICITY AND ACCURACY.

NO WARRANTY IS PROVIDED TO THIS DOCUMENT FOR ITS MERCHANTABILITY, NON-INFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, NOR DOES ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE.

All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No licenses express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

The Wi-Fi Alliance Member logo is a trademark of the Wi-Fi Alliance. The Bluetooth logo is a registered trademark of Bluetooth SIG.

All trade names, trademarks and registered trademarks mentioned in this document are property of their respective owners, and are hereby acknowledged.

**Copyright © 2021 Espressif Systems (Shanghai) Co., Ltd. All rights reserved.**